Max-i Specification

Version 15.1 May 28th 2025

Brief Overview

Max-i is a very cheap, but extremely powerful fieldbus, which has been designed to enable the lowest possible automation and lighting costs combined with a **very** high performance! It may be regarded as a combination between a highly improved CAN bus, a high-power 20-V power supply running on standard installation cables and some hardware-based I/O objects to handle the most commonly used functions like UART interface to microprocessors, SPI interface to for example sensors, displays and servo motor controllers, 4-bit Boolean I/O (on/off), A/D and D/A conversion, lock control with keypad scanning and very advanced LED driving. Max-i may be used for virtually all low to medium speed applications (up to 25,000 telegrams per second on short lines) like:

- Industrial, building and home automation including safety applications.
- Transportation applications such as automotive, special vehicles, railway, ships and aerospace.
- Traffic lights and supervision on bridges and tunnels.
- Intelligent LED lighting including demanding stage light and architectural lighting.
- Supplementary 20 V supply with control possibilities in the houses of the future.
- Military applications.

The name Max-i means **M**ultiple **A**ccess Cross (**X**) coupled **i**nterface. It refers to Max-i being an event driven multiple master bus (no dedicated master and slave units), which uses a cable with 4 conductors connected as a balanced 4-wire line (X-coupled) for all applications where mutual coupling to other cables are possible such as industrial applications and stage light. The name also refers to the Max-i-mum performance.

For applications where mutual coupling is not a problem and/or an unbalanced network is desirable such as a 20 V net in the house of the future or a charging net in busses, trains and aircrafts, a 3-wire line may be used, and for automotive application where a common chassis can be used for the negative supply, a simple 2-wire line may be enough.

Max-i has four basic properties:

- It is possible to make a complete bus interface in one integrated circuit (IC), which is small and cheap enough to be built into even the smallest and most price sensitive actuator, sensor or lamp, gets its supply voltage directly from the bus and only needs an absolute minimum amount of external components depending on the application – in some cases only a few small ceramic capacitor. This enables a new world of single-chip sensors and it not only saves the traditional distributed coupling boxes, components for transient protection and a lot of cabling, but also the expensive and time consuming conformance tests, which are required by most other fieldbus standards. It also makes it possible to do with only one bus in automotive applications except for infotainment systems because Max-i is as cheap as LIN, more powerful than CAN, as deterministic as TTCAN (Time Triggered CAN) and for a given amount of money, it offers the same speed and safety as FlexRay.
- It is possible to use cheap, standard, unshielded and un-terminated installation cables instead of special communication cables. This saves a lot of money and troubles and makes it possible to transfer much more power over the bus than any other fieldbus – in practice up to approximately 800 W on 4 mm^2 cables if there is a power supply in both ends. This not only makes it a true actuator bus, but also very suitable as a supplementary 20-V supply in the houses of the future for driving all kinds of battery chargers, computers and computer peripherals, alarm systems, energy management, intelligent LED lighting etc., so that all the clumsy converter boxes of often doubtful quality and cooling can be avoided. Because the cable is not shielded, the traditional dilemma with the shield connection is avoided. For the sake of noise immunity, a shield must be connected to ground/chassis in both ends, but it is simultaneously a **very** bad idea to establish ground loops and potential equalization between different parts of a plant through a thin fieldbus cable – especially in plants, which use a TN-C net with a common neutral and protective earth! Nevertheless, it is common practice with most other fieldbus systems!
- It is the first fieldbus designed directly for highly demanding safety applications according to IEC 61508 SIL 3. Many fieldbus systems are approved to this level, but they all obtain that by means of an additional safety

protocol on top of the fieldbus, separate safety monitors etc. The hardware based Max-i controller with its I/O objects cannot “go down” in the same way as software and is much more predictable and failure tolerant.

- Max-i is **very** simple, which makes it possible to learn in a few hours. There are just a few registers to setup and a new numbering system – PNS, which allows the various process values to be addressed directly as properties of the equipment to which they belong like for example HX127AT4 for Heat Exchanger 127A Temperature 4. The specification fills only 253 pages, but approximately half is used for background material and annexes and a lot is only necessary for the chip designer. As a comparison, most other fieldbus systems have specifications way over 1000 pages.

Document History and Revision Log

In this specification, a new version number X.Y (and date) indicates a change in the specification. A change in X indicates that devices made to this specification are not backward compatible. If only Y is changed, devices are compatible, but new features may have been added. If only the date is changed, there are only changes in the description, which do not affect the requirements of the specification.

Version Number	Date	Comments
1.0	May 21st 2010	First official release of a specification according to the first workable prototype.
1.1	July 16th 2010	Error in cable specification corrected. Error in formula for decoupling capacitor corrected.
1.1	July 28th 2010	Primary amendments in Annex A (Transmission Lines).
1.2	August 4th 2010	Error detection moved from Layer 4 to Layer 2, but incidentally unchanged. Retransmissions of Layer 5 moved to Layer 4, but incidentally unchanged. Type number format changed from AANNNNAA to AAANNNNA.
1.3	September 24th 2010	Amendments due to skin effect in lossy transmission lines. • Reduction factors for maximum line length. • Input hysteresis reduced from 4 V to 3 V at 12 V. • Description of skin effect in Annex A. Compensation capacitor for coupling capacitance changed to bootstrapped capacitor for high capacitance values. Specification of spur cables and maximum number of devices.
1.4	October 1st 2010	Formulas and tables added for capacitive and resistive load.
1.4	October 21st 2010 October 25th 2010	The terms “Attribute telegrams”, “value telegrams”, “attribute base” and “channels” changed to explicit messages, implicit messages, object base and I/O objects for better compatibility with the terms of CIP protocols such as DeviceNet, CompoNet and Ethernet/IP.
1.5	November 5th 2010	“Hardware interface selection attribute” changed to “I/O object specification attribute” and expanded with object size.

2.0	March 31st 2011	New optimized bit coding and timing – primary Layer 1. • ≈36% more telegrams per second for 32-bit ID and 2-bit data. ≈23% more telegrams for 32-bit ID and 18-bit data. ≈21% more telegrams for 13-bit ID and 2-bit data. ≈ 8% more telegrams for 13-bit ID and 18-bit data. • Bus arbitration now possible during the entire telegram. Parity check added to important setup attributes. Standardization of I/O object types in Layer 7.
3.0	August 2nd 2011	Optimized and changed UART speeds. Addition of 2 higher speeds – up to 3.556 Mbaud for Max-I (4.9152 Mbit/s for UART) at a line length of 15 m. Addition of 2 lower speeds and specification of reflected wave switching mode to allow much thinner cables if a reduced speed can be accepted. 16-bit coarse filtering added as a supplement to local/global in Modem object, and therefore Local and NoCount bits moved. Default telegram serial number changed from 00 to 01 to avoid confusion with break byte (00) in case use of the break flag is not supported. Default state of telegrams changed to global as it were in specification 1.x when the MSb of the ID (local) was 0 (local bit moved). Default state of output 2 in standard I/O objects changed to steady voltage to ease connection of optocouplers for Modem object with default settings. Possibility for inversion of serial I/O signals added in Modem object. Addition of optional PnP. • Always possible to store ID in RAM (not just EEPROM). • Automatic baud-rate detection.
4.0	December 29th 2011	Addition of flash synchronization to time telegrams. Signature of time telegrams and group telegrams changed to enable boolean devices to issue such telegrams and to give global messages and implicit messages the same telegram structure. Short (2-bit) flash synchronization telegram and group telegram (default group = 0) added to allow boolean devices to synchronize flash and switch a group of devices off. Addition of extra start bit to ensure that the idle bus state is always high and in this way ease live bus connections. Specification of bus connection, contact fritting and numbering for railway applications. (Modification of recommended output (three state) to allow a very simple and high efficiency drive (a single coil) of small and medium sized LED´s.)
4.0	January 4th 2012 January 13th 2012	Drawing and description of bus connection and fritting in railway applications enhanced. Minor bugs corrected.

4.0	February 6th 2012 February 7th 2012	Addition of optional filter hysteresis to input circuit. Enhancements in description.
4.0	February 9th 2012	Formula for load switching with limited di/dt added. Enhancements in description.
4.0	March 13th 2012 April 3rd 2012	PNS data types changed and minor changes in the interpretation of the two least significant bits.
5.0	July 5th 2013	Modification of output as per V4.0, December 29th 2011 removed again. Minimum line impedance reduced from 40 Ω to 30 Ω to allow more devices per meter. Expansion of reception delay from 255 16-bit words to 2047 bytes. Timer settings in signatures moved to make space for the necessary 11 bits. I/O settings changed to allow both password 2 and password 3 protected settings (previously only password 3). Addition of Lamp Controller object and Annex C belonging to it. Addition of D/A Converter object. Nominal supply voltage increased from 12V to 13V.
5.0	August 1st 2013	Specification of brownout (low voltage) behavior. Enhanced description.
5.1	August 16th 2013	Addition of 1.953, 3.906, … 125, 250, 500, 1000, 2000, 4000 kbit/s UART speed row.
5.2	September 6th 2013	Addition of trigger telegram for transmission of error/status telegram. Addition of safety output for green traffic light. Dimming counter changed from 7 to 6 bits.
5.2	September 9th 2013	Description of Fourier series for transmitted waveform.
5.3	September 20th 2013	Specification of high voltage power supply for high power applications.
5.3	September 23rd 2013	Color code for high voltage supply and enhancements in description. Nominal supply voltage increased to 14 V. Minimum supply voltage increased to 11 V to enable 3 LED’s in series.
5.3	September 25th 2013	Enhanced specification of brownout voltage.
5.3	October 2nd 2013	Enhanced specification of power supply and compensation for coupling capacitance on balanced lines.

6.0	January 22nd 2014	The high voltage power supply option added in V5.3 from September 23rd 2013 replaced by a general increase in supply voltage from 14 V to 20 V except for automotive and Ex applications. Line interface modified and reverse polarity and power-down protection added. Minimum line impedance increased from 30 Ω to 35 Ω to limit the increase in maximum transmitter and clamp current due to the higher voltage. Idle COM voltage changed from high to low. Addition of Annex D, which describes the background for the voltage increase and arc considerations.
6.0	January 28th 2014	Current values for 6 mm^2 cables added. Linguistic improvements.
6.0	January 30th 2014	Additional requirements for power supplies and improved description. Linguistic improvements.
6.0	January 31st 2014	Slightly changed fritting circuit (3-position switch).
6.0	February 5th 2014	Small changes in supply voltage specification. Improved specification and description of automotive applications.
6.1	February 18th 2014	Addition of decoded three-level boolean output for use for example for thermostats, heating, cooling, fan speed etc.
7.0	June 22nd 2014	Improvements of group control. • Addition of analog values including long telegrams with reception delay (no limitations compared to implicit messages). • Group addressing moved from network object to implicit messages and number of groups changed from 255 to 31 to be able to add analog values. • Addition of 100% command (2-bit message) for lamps. • Enhanced daylight control of lamps used for lighting (very long filter constant added). Addition of 18-bit light value telegrams to monochrome and dichromatic lamps and traffic lights and control lamps. A/D and D/A converter combined to one object to reduce the necessary numbers of controllers for many applications. LED object renumbered. Minimum value for lamps changed from 19 to 15. Addition of programmable minimum value for traffic light and control lamps. LED calibration factors shifted 1 bit to make room for 6-bit minimum value. Maximum reception delay changed from 11 bits to 10 bits (1022) to make room for 5-bit group numbers. Modem watchdog changed from 516 bytes to 1028 bytes (or infinite) to be compatible with maximum reception delay.

8.0	July 3rd 2014	Number of groups increased to 255 (8 bit). Previously unused attribute 8 used as a group register to make room for an 8-bit group number with 7-bit ID check and individual group delay. ID check and timers swapped in publisher signature and subscriber signature to make them equal to group register.
8.1	July 22nd 2014	Traffic light coding changed for better world-wide compatibility.
8.2	August 11th 2014 August 16th 2014	Selection between filter speed or gamma correction changed. Addition of programmable minimum value for one output on monochrome lamps. Addition of modem acceptance filter on group telegrams.
8.3	May 14th 2015	Removal of negative voltage levels in line interface to make the circuit compatible with a bulk CMOS or BiCMOS process (SoI not needed). Removal of 12-V option and temperature specification. Specification of maximum current from a Max-I outlet. Ex specification changed to HPTC. Simplified receiver circuit.
8.3	May 24th 2015 May 25th 2015 May 28th 2015	Brownout current changed to a percent value. Better specification for slew rate reduction. Better description with added links to various studies.
8.4	August 7th 2015	Simplified physical layer.
8.4	August 24th 2014 August 25th 2015 September 21st 2015	Improved description of hold circuit and transmitter with simplified schematic.

9.0	May 23rd 2016	Values changed from N × 16 bit + **2** bit to N × 16 bit + **4** bit. This has caused several side effects: • CRC polynomial changed from 22 to 20 bit. • Passwords changed to 6 decimal digits (20 bit). • Telegram serial number changed from 8 to 7 bit and combined with 1-bit data type. • Bit positions of serial number and ID check in signature interchanged. • FIX data type changed from 18+6 or 34+6 bit to 20+6 or 36+6 bit. • MSD of BCD data types changed from half digit to full digit. • Boolean object changed from 2 to 4 bit and previous 3-level coding removed (not necessary anymore). • A/D and D/A object changed from 18 to 20 bit and prescaler removed. • SPI object expanded with possibility for nibble and 9-nibble (36 bit) transfer. • 8 fixed levels and gamma control added to short (now 4-bit) lamp control. • Number of digital filter speeds for lamp control increased to 7 plus bypass. • 8 user defined data types with up to 32 bits added to lamp control for example for direction, focus, gobo, special effects etc. Timer resolution changed from centiseconds to milliseconds. 16-key keypad scanner object added. New data type (LAMPCTRL) added and coding of other data types changed. RGBAaC color system added (artificial cyan as a supplement to artificial amber). Object numbers changed for more logical order. Serial number slightly changed and used for identifier, which triggers an error message during PUR, if the identifier is not programmed. A few bits interchanged in the error word to make room for a parity test on the serial number (necessary for use as identifier). Number of I/O's fixed to 4 inputs and 5 outputs. Inputs expanded with possibility for use as error triggers. Optimized data coding for up to 16 % more efficiency on long telegrams with more than 20 bits of data. Addition of "mains plus dimming" cable for high power applications. Zener diode clamps changed to a row of diode-connected transistors. Number of communication speeds reduced from 24 to 9. Minimum UART speed increased to 9.6 kbit/s. Maximum speed reduced to 2 Mbit/s (1.4 Mbaud on bus) to reduce the necessary oscillator frequency to 32 MHz, which is more suitable for low-power microprocessors than 64 MHz.
9.0	May 26th 2016	Minor text changes.

9.1	March 9th 2017	Line interface changed to symmetrical, regulated transmitter and clamp to enhance the signal integrity and allow up to three times more voltage drop on L+ and L-. Possibility for closed ring topology on COM (not L+ and L-) removed due to high power loss and possible bad signal integrity during bus arbitration. Annex A. “Controller Implementation” removed to allow more flexibility.
10.0	February 17th 2018	Maximum timing inaccuracy reduced from ±8 % to ±6 % to increase the minimum safety margin from 0.7T to 1T (2T between telegrams) and make 9th bit communication possible. Bit timing changed from 7.5 or 6 samples per T to 8 samples per T to be able to: • double the maximum speed for the same clock frequency by utilizing both clock edges. • use the same digital receiver filter for all speeds. • use 9th bit UART communication as an alternative to Break. • ease future automatic baud-rate detection.
10.1	April 10th 2018 May 4th 2018	Addition of resistor terminated mode with or without line transformer. Enhanced description.
10.2	August 29th 2018	Voltage range changed from 18 V ±20 % (14.4 – 21.6 V) to 20 V ±23 % (15.4 – 24.6 V) to get the open circuit voltage (no charging) of a battery based power supply closer to 20 – 21 V for better compatibility with USB Power Delivery and Quick Charge, to allow traditional, industrial 24 V operation (24 V +2.5/-10 %) and to increase the power by 30 % (proportional to the supply voltage in the power of two). 3-level receiver hysteresis changed to 4-level in case of resistor termination.

11.0	March 31st 2020	Number of I/O changed to 8 inputs and 8 outputs. Soft start function (by means of gamma correction and smoothing filter) added to boolean outputs. Boost function added to increase the transmitter and clamp voltage in case of very thin and/or long transmission lines and in this way improve the signal integrity. High level hold current switched off from 0 – 1T. Flash function for lamps changed from short 4-bit boolean messages, which can only drive two lamps, to 20-bit implicit massages to utilize all 4 outputs and enable that any number of lamps can be driven synchronous from a common message. Added possibility for individual phase inversion of the two flash frequencies for all 4 outputs (8-bit programming). Gamma settings 2.00 and 2.23 removed as there will probably never be any need for them. This simplifies programming, frees some 4-bit message codes for other purposes and also saves some gates. Default time constant for lamp doubled to 160 ms. Extra lamp output with a gamma of 1.56 added to ease dim-to-warm. Programmable start level and emergency color added to lamps. Analog status signals added to monochrome lamps. X16 and X8 UART clock output added to MODEM object to enable clock synchronization if connected device has an external clock input. 2 boolean outputs added to Keypad scanner to drive a locking pawl. 4-bit boolean output possibility added to ADA object.
11.1	September 28th 2020 to November 17th 2020	Don’t care bits for MODEM object expanded and moved from subscriber ID and group ID to I/O attribute 1. AND function for 4-bit Group messages of the TIME data type replaced by OR function on output 0 and “old” AND on output 1 – 3 to enable “all-lock”. Possibility for bin-correction for LED’s removed from all other objects than LAMP’s as it makes no sense for control lamps etc., but occupies I/O attribute 1 and 2. Added expansion of 20-bit cold-white/warm-white group messages to RGBW or RGBA on 36-bit, full-color LAMP’s with or without automatic switch-off of artificial amber to make the RGB-part cold. Added recording of local time on control lamps. Lamp and TX-active outputs on KEYPAD object replaced by dual-phase beeper outputs to enable direct connection of a small loudspeaker. Data type for group message for light level control of MODEM and SPI object fixed to LAMPCTRL instead of subscriber data type. Boolean inputs for LAMPCTRL changed to also make it possible to publish 8 fixed light levels – even as group messages.

12.0	February 10th 2021 to March 10th 2021	Added possibility to control two lamps from one Max-i controller. Added AC/DC-converter enable output on LAMP. Coding of 4-bit LAMPCTRL commands modified. Minimum lamp level changed from 5.9 % linear to 15.2 % linear.
12.0	June 4th 2021	Requirement for inrush current limiting added to power outlets. Tolerances for mains driven power supplies reduced to enable parallel operation.
12.1	August 25th 2021	Added possibility for 2 × 6 keypad. Separate timeout added on group messages to enable them to be used for emergency stop. Simplified line interface added for applications like traffic lights where a regulated power supply is used and the devices have approximately the same temperature and are not powered down separately.
13.0	March 27th 2023	Added possibility for (rotary) quadrature encoder input to LAMP object. Gamma and filter bits changed and maximum smoothing filter time increased (4 bits instead of 3) for better usability for inrush current limiting. Serial number format modified and enabled to be usable as auxiliary identifier. Out-of-the-box UART speed selection added to password 3. 3 lower speeds added (UART 1200, 2400 and 4800 bps) to make Max-i suitable for long distance communication on thin cables for example for street lighting.
13.1	April 29th 2023	New coding for 4-bit LAMPCTRL to make off/0% = 0000B and compatible with boolean. Enable of “All-off” on group 255.
13.2	August 9th 2023 to October 10th 2023	Added programming bit in I/O setup attribute to select the data type of group messages to be equal to the subscriber or be LAMPCTRL, and changes made in many objects according to this. B20 bit, which specifies the expected length of implicit messages, moved from attribute 12 to attribute 7 to make it user programmable. Group timeout changed to exponential scale to make room for a DIS255 bit to enable All-off (and Back-on). Improved AND/OR coding for group messages with data type TIME. Error bit for group watchdog added and error bits reorganized. More logical coding of ID check for dual lamp publisher. Added possibility to subscribe to two implicit identifiers if the publisher data type is LAMPCTRL.
14.0	December 8th 2023	Light level register expanded to full 32 bit to enable reception of both light level and (mechanical) functions for stage lamps. Added SPI interface to LAMP object to transfer non-light functions like pan, tilt, zoom and focus to a connected microprocessor for handling. Maximum inrush current limiting time constant doubled to 5.4 seconds and number of (mechanical) lamp functions increased from 4 to 6.

14.1	January 8th 2024	Added SPI interface to BOOLEAN object to enable traffic lights with countdown display in a single Max-i controller. Added square root gamma output for artificial amber to enable dim-to-warm on RGB light (not just white light).
14.2	March 13th 2024	Changed debounce and error detection circuit to be able to detect (error)status of dimmed lamps.
14.3	May 31st 2024 to June 11th 2024	Addition of group 255 Panic light with same color as emergency light. >0.4 s sequence disabled for All-off, Back-on and Panic light (only 2 commands). Write protection of NVRAM simplified to 1111B (no input dependency) as identification by means of serial number has replaced the auto programming mode. Parity check removed from hour counter. Error bit for external oscillator failure added. Preparation for future 36-bit NVRAM attributes: • Bit 32 – 35 fixed to 0000B for Hour Counter and to 1111B for most other attributes. • Serial number format simplified to fixed 00000B for bit 13 – 17.
14.3	January 16th 2025	Maximum operating voltage increased from 24.6 V to 25.2 V to include 24-V applications and power supplies with a standard voltage tolerance of +5/-10 %.
15.0	January 31st 2025 to April 8th 2025	More detailed specification of the behavior in case of attempts to program an attribute by means of a wrong password. Write enable of **all** local and global attributes in NVRAM changed from bit 32 – 35 = 1111B to dual write with same 36-bit data to enable future full 36-bit parameters instead of only 32 bit. Dual write was previous only used for the 4 speed bits in global attribute 2. Data types and telegram signature optimized to enable: • Sign on all analog values (unsigned/signed or positive/negative). • An 8-bit FIX exponent instead of only 6 bit. • An 8-bit telegram serial number instead of only 7 bit. • Addition of an 8-bit code page specification for text strings. • Addition of an 8-bit specification for standard and used defined patterns.
15.1	May 20th 2025 to May 23rd 2025	Enhanced and simplified acceptance filter with individual enable bits for different message types and no dependency of subscriber data type (UNDEF). Hysteresis selection bit moved from local attribute 12 to global attribute 2 to make it possible to set this by means of password 2 instead of password 3. Fast poll disable bit added to make it possible to use master/slave mode on long distance and/or high speed communication where fast poll is not possible. Automatic transmission of time synchronization and flash bits with 2, 3 or 4 divider added by means of heartbeat timer.

Table of Contents

Notice of use and Disclosure

Max-i Association
c/o Innovatic

Boegebakken 3
Gjessoe
DK-8600 Silkeborg
Denmark

Build-Up, Glossary and Word Definitions

This specification/description is based on the OSI 7-layer reference model, which has the following definition:

Layer 1 is the Physical Layer, which is responsible for transmitting raw single bits without any particular meaning over the physical medium such as copper, optical cable, wireless communication etc. It describes the electrical interface, network topology, cable types with connectors, termination/clamp network, power supply, bit coding, baud rates etc. Typical protocols are RS-232 and RS-485. A hub is a layer 1 device.

Layer 2 is the Data Link Layer where small packages of data are converted to raw bits. This layer controls the bus access, creates and recognizes frame boundaries and value identifiers and performs error detection and possible error corrections. Typical protocols are LLC, MAC and HDLC. A switch is a layer 2 device.

Layer 3 is the Network Layer. The purpose of this is to select a route between nodes on the network, translate logical net names (IP addresses) to physical net addresses (MAC addresses) for layer 2 and filter telegrams according to specific rules including source and destination address, package type and port number. The layer takes segments from layer 4 and adds a header with the address and in this way converts a segment to a so called package. Typical protocols are IP, IPX and AppleTalk. A router and a firewall is a Layer 3 device. In Max-i, this layer is used to define repeaters, gateway functions and firewalls between internal, safe Max-i networks and external, unsafe Max-i networks and other networks like the internet.

Layer 4 is the Transport Layer. The purpose of this is to ensure end-to-end reliability by means of acknowledge telegrams and error recovery like retransmissions etc. This layer is also responsible for splitting big data files into smaller segments before transmission and recombining the segments again in the receiver. Typical protocols are TCP, UDP and SPX. In Max-i, this layer handles any automatic retransmissions in case of collisions or an unanswered poll.

Layer 5 is the Session Layer, which handles authentication, permissions, session restoration etc. It has three communication modes – simplex, half-duplex and full-duplex. A typical job for a session layer is to control the access to the internet from more browsers running simultaneously. In Max-i, this layer is empty.

Layer 6 is the Presentation Layer. The purpose of this layer is to ensure correct interpretation of the received data and perform data conversions like translation between different code pages, data compression such as JPEG, GIF and TIFF, data encryption end decryption etc. The layer is used to describe the standard formats for analog and boolean values and the configuration parameters (attributes). To enhance the overview, this layer is divided in two parts. Layer 6a describes the attributes, and Layer 6b describes the data types.

Layer 7 is the Application layer, which gives access to high-level net functions. Typical protocols are Telnet, WWW and e-mail. In Max-i, it defines standardized I/O object types.

This document contains a lot of information about why Max-i is designed and specified the way it is. This information is written in *blue italics* and may be skipped together with the annexes if you only want the pure specification.

The following definitions apply to all parts of the specification:

The word **shall** indicate a mandatory requirement to be strictly followed in order to conform to the standard. **Shall** equals **is required to**.

The word **should** indicate that, among several possibilities, one is recommended as being particularly suitable, without mentioning or excluding others; that a certain course of action is preferred, but not necessarily required; or, that in the negative form (should not) a certain course of action is deprecated, but not prohibited. **Should** equals **is recommended that**.

The word **may** indicate a suggestion, a possibility or a permissible action within the limits of the standard. **May** equals **is permitted to**.

The word combination **must only** is similar to **shall only**. **Must only** equals **is only allowed to**.

Hexadecimal numbers are indicated with a H-suffix like 3AH and binary numbers are indicated with a B-suffix like 1111,1100B. To ease the reading, bigger binary values are divided in groups of nibbles (4 bits) by means of commas and each comma is usually nibble aligned with the telegram. Numbers where the LSb is not nibble aligned may therefore have less bits than four to the right of the comma like 111,111B.

Acronyms and Abbreviations

A	Letter (A-Z). For example, AA is a two-letter code.
AC	Alternating Current
A/D	Analog to Digital
AFCB	Arc Fault Circuit Breaker
AMI	Alternating Mark Inversion
BCD	Binary Coded Digit
BiCMOS	Bipolar + CMOS
BOX	Buried Oxide
BWM	Bit Width Modulation
CAN	Controller Area Network (fieldbus from Bosch, Intel and Philips)
CIP	Common Industrial Protocol
CMOS	Complementary Metal Oxide Semiconductor
CMTI	Common-Mode Transient Immunity
COM	Communication
CPHA	Clock Phase (SPI)
CPOL	Clock Polarity (SPI)
CRC	Cyclic Redundancy Check
CRI	Color Rendering Index
CS	Chip Select (SPI)
CSMA-CD	Carrier Sense, Multiple Access with Collision Detection
CTS	Clear To Send
DALI	Digital Addressable Lighting Interface
D/A	Digital to Analog
DC	Direct Current
DQHI	Double Data rate Quad Hendshaked Interface
EEPROM	Electrically Erasable Programmable Read-Only Memory
EHS	European Home Systems
EIB	European Installation Bus (fieldbus)
EME	Electro-Magnetic Emission
EMF	Electro-Motive Force
EMI	Electro-Magnetic Interference
ESR	Equivalent Serial Resistance
Ex	Explosion Safe

FIFO	First In, First Out memory
FIR	Finite Impulse Response
FISCO	Fieldbus Intrinsically Safe COncept
FLL	Frequency Locked Loop
FM	Frequency Modulation
FPGA	Field Programmable Gate Array
FRPE	Flame Retardant Polyethylene
HDLC	High-level Data Link Control
HPTC	High Power Trunk Concept
I^2C	Inter-integrated Circuit
IC	Integrated Circuit
ID	Identifier
IDE	Identifier Extension
IIR	Infinite Impulse Response
IoT	Internet of Things
IP	Internet Protocol
IPX	Internetwork Packet eXchange
I/O	Input/Output
JSON	JavaScript Object Notation
LDO	Low Dropout
LED	Light Emitting Diode
LLC	Logical Link Control
LPS	Limited Power Source
LSb	Least Significant Bit
LSB	Least Significant Byte
MAC	Multiply and Accumulate
	Media Access Control (Ethernet MAC address = global unique address)
Max-i	Multiple Access Cross-coupled Interface
MISO	Master data In, Slave data Out (SPI)
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MOSI	Master data Out, Slave data In (SPI)
MOV	Metal Oxide Varistor
MSb	Most Significant Bit
MSB	Most Significant Byte
MSD	Most Significant Digit
MVB	Multifunction Vehicle Bus
N	Digit (0-9). For example, NNN is a three-digit number.
NAMUR	NormenArbeitsgemeinschaft für Mess- Und Regeltechnik
NC	Normally Closed

NMOT	Nominal Module Operating Temperature (standard for solar panels)
NO	Normally Open
NU	Not Used
NRZ	Not Return to Zero
NVRAM	Non Volatile RAM, that is, a memory, which does not lose its content in case of a power failure.
PCB	Printed Circuit Board
PCM	Pulse Code Modulation
PDR	Power Down Reset
PE	Polyethylene
PEX	Cross Linked Polyethylene
PLC	Programmable Logic Controller
PLL	Phase Locked Loop
PnP	Plug and Play
PNS	Plant Numbering System
PP	Polypropylene
PPM	Parts Per Million
PS	Power Supply
PV	Photovoltaic (array) = solar cells
PVC	Poly Vinyl Chloride
PWM	Pulse Width Modulator
PUR	Power-Up Reset
RAM	Random Access Memory
RC	Resistor and Capacitor
RDM	Remote Device Management
RGB(A/W)	Red, Green, Blue, (Amber/White)
ROM	Read-Only Memory
RTC	Real Time Clock
RTM	Released To Manufacturing
RTR	Remote Transmission Request (CAN)
RZ	Return to Zero
SCADA	Supervisory Control And Data Acquisition
SELV	Separated or Safety Extra Low Voltage
SFB	Selective Fuse Breaking
SMS	Short Message Service
S/N	Signal to Noise Ratio
SoI	Silicon on Insulator
SPI	Serial Peripheral Interface
SPSCLK	SPI Serial Clock
SPX	Sequenced Packet eXchange

SS	Slave Select (SPI)
STC	Standard Test Conditions (standard for solar panels)
TBD	To Be Determined
TCP	Transmission Control Protocol
TIG	Tungsten Inert Gas
TSPD	Thyristor Surge Protection Device
TSRE	Transmitter Shift Register Empty
TTL	Transistor, Transistor Logic
TVS	Transient Voltage Suppressor (Avalanche diode)
UART	Universal Asynchronous Receiver and Transmitter
UDP	User Datagram Protocol
Uoc	Open Circuit Voltage
UPS	Uninterruptible Power Supply
UVLO	UnderVoltage LockOut
VCO	Voltage Controlled Oscillator
VPN	Virtual Private Network
WTB	Wire Train Bus
XML	Extensible Markup Language
Z_0	Characteristic Impedance

Layer 1, Physical Layer

This layer is responsible for transmitting raw single bits without any particular meaning over the network. It describes the cable type with connectors, power supply, clamp networks and network topology. It also describes the electrical interface with bit coding, baud-rate etc.

As described previously, Max-i may be regarded as a combination between a fieldbus, a 20-V power supply and some standard hardware-based I/O objects. The reasons for choosing 20 V as the nominal voltage instead of the much more common 24 V are described in Annex D, but the most important are compatibility with the 20-V de-facto charging standard for mobile phones and laptop computers and **much** lower arc risk at high current levels. Due to a very big voltage range of +26/-23% as specified later, Max-i is however also useable for standard 24-V industrial and railway applications as long as the voltage is always below 24 V + 5% = 25.2 V.

1.1 Topology

1.1.1 Trunk Line

For maximum speed communication, Max-i uses incident wave switching (switching on first wave) on a trunk line where the devices and power supplies are placed as beads on a string – in practice a single cable with at least 3 conductors – two for power supply (red and blue on the drawing below) and one or two for communication (black):

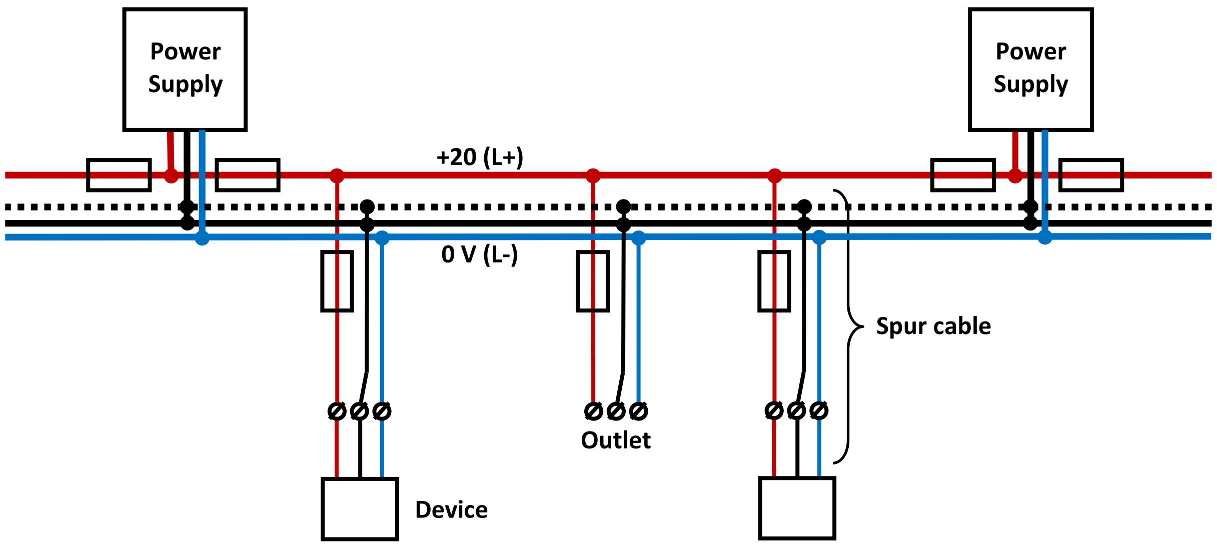


Fig. 1.1

Short spur cables are allowed, but the maximum length of the trunk line as specified later is reduced with the sum of the length of all spur cables so that the total capacitance of the line is not increased, and the length of each spur cable shall be limited to:

Length ≤ RiseTimeOfPulses / (6 × PropagationDelay)

Length ≤ 10 m

If for example the rise time of the communication pulses is the standard 120 ns as specified later and a PE cable with a propagation delay of 5 ns/m is used, the maximum length becomes 120 ns / (6 x 5 ns/m) = 4 m.

In practice, a few 10 m spur cables may not cause any problems even with a rise time of 120 ns, but as a rule of thumb, the maximum spur cable length is 1/3 of the line's critical length, which is the length where the down-and-back propagation delay equals the rise time of the pulse.

Note that devices between power supplies receive power from two sides so that the available current in the outlets are up to two times what the cable cross section and fuses would allow with a single power supply. This gives a very good utilization of the cable cross section and reduces the voltage drop considerably as long as the differences in power supply voltage is so small that they share the current fairly equally at maximum load.

⚠ **If a device needs a power switch or a switch is mounted in an outlet, this shall be placed in L+. L- shall never be disconnected alone.**

1.1.2 Railway Connection

If there are no devices on the line between the power supplies, they receive power from their own power supply or it is desirable to use separate power supplies for each line segment like for example connected trainsets, the positive conductor L+ may be more or less omitted. This makes it possible to use a simple 2-wire line and/or a 2-pin connector, **but to avoid a serious unbalance in the communication, the supply voltage on each side of a 2-wire line or a 2-pin connector shall match within ±2 % (±1 % tolerance for each power supply)**. This also makes it possible for more power supplies on the same segment to share the current equally at high loads. In practice, this can only be guaranteed with a power supply in each end of each segment as shown below because a single supply cannot guarantee low source impedance (no voltage drops) in both ends of a segment simultaneously.

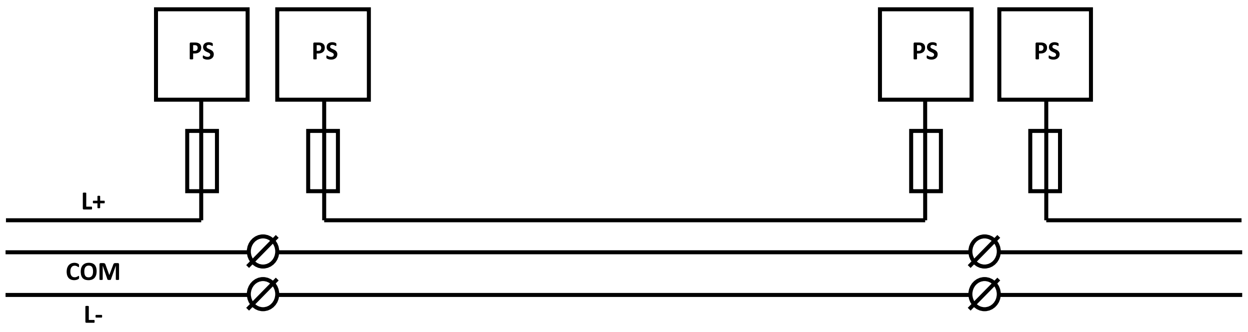


Fig. 1.2

Note that in case of a long interconnection, the line impedance must be maintained! This may for example be achieved by means 4-wire lines (described later) with two COM-conductors and either an L+ and L- connector or two L- connectors, which also creates redundancy.

A beneficial side effect of this connection is that the very low voltage drops and tight voltage range makes it possible with a very simple line interface with a saturated transmitter. This gives the maximum possible signal/noise ratio and may even save the traditional fritting generators in railway couplings as the special clamp termination (described later), which also saves the traditional termination resistors, generates a fritting voltage on an open circuit of 1.5 times the transmitter voltage swing so that the fritting voltage will be approximately 34 V for a standard 24-V railway system. This is close to the 48 V, which has proven to be enough even for harsh industrial environments with contact corrosion of up to 0.5 µm (fritting voltage 100 V/µm), and since the possible fritting current is **much** higher than from a fritting generator, the difference is probably not so big in practice.

1.1.3 Battery Operation

If the power comes directly from the poles of a battery, it is in practice impossible to use more power supplies due to the big voltage differences during charging and discharging, but in this situation, the trunk line may just be bent together in a "multi loop plus two tails" structure as shown below, so that the same power supply is reused several times:

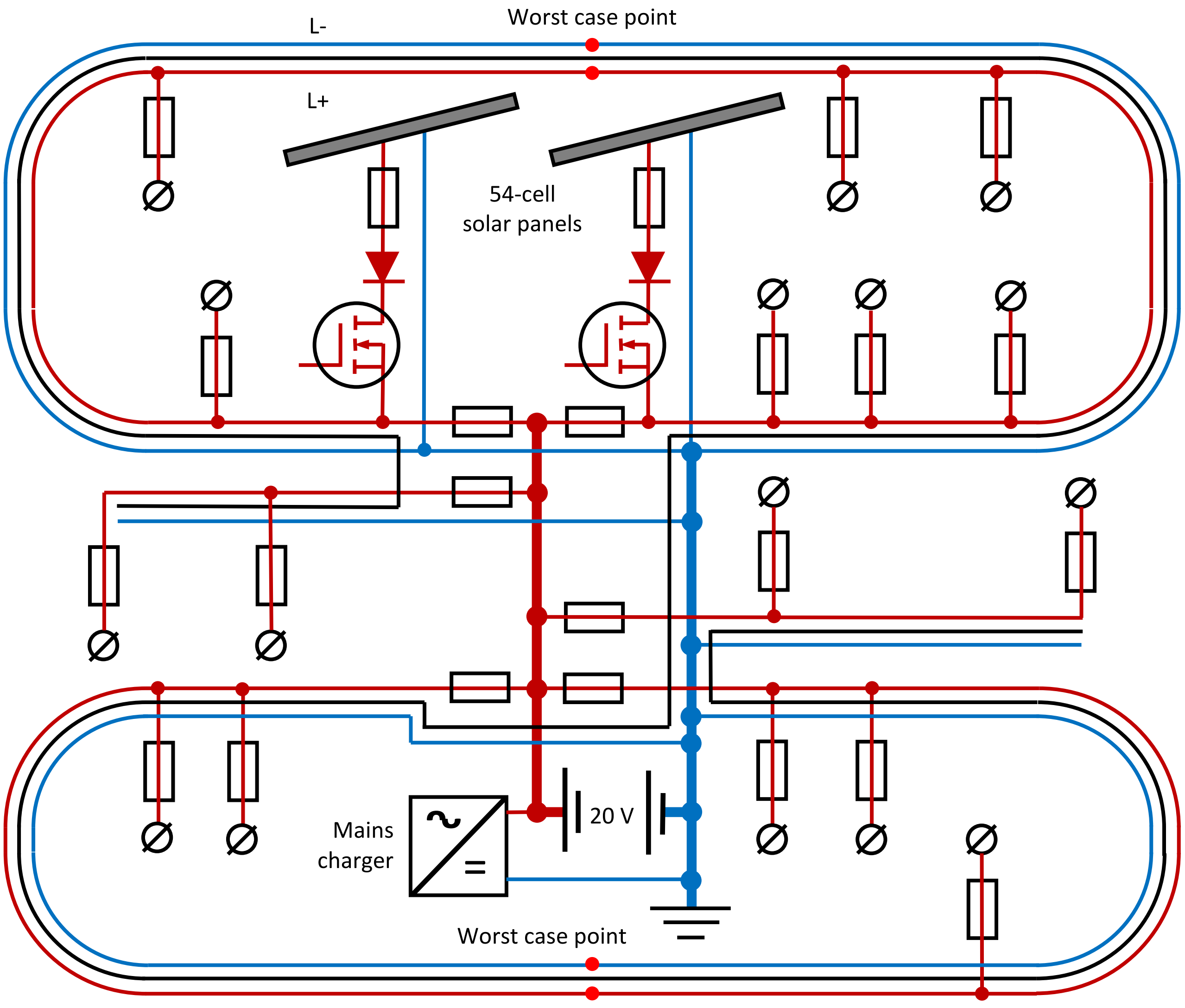


Fig. 1.3

The drawing shows a typical "green" smart-house application with two loops plus two tails, but the principle may be expanded to any number of loops for example in case of more floors. It is not necessary to use the same cable cross section for all loops and tails as long as the fuses are selected according to this. This may for example be practical for outdoor and garden lighting, where relatively long cables (tails) are needed, but low currents.

All loops and tails may be supplied from two high-current bus bars as shown, which are connected directly to the poles of a battery, which may be charged **directly** from solar panels and from a mains charger when there is not enough solar power and the electricity is cheap – typical during the early morning.

The communication (COM) conductor(s) shall be routed through from one end to the other so that the cable is still a trunk line from a communication point of view. If it is necessary to route the COM conductor(s) as single wire(s) over longer distances for example from one floor to another, it/they shall be drawn as close as possible to the negative bus bar (L-) as illustrated above to keep the characteristic impedance low – for example connected by means of strips, but it/they must not couple to other parts of the COM conductor(s). This may be prevented by using both sides of L- and in this way use it as a shield as also shown above.

1.1.4 Free Topology and Reflected Wave Switching

Because Max-i does not use any termination resistors, it may also use reflected wave switching where the signal may not pass the detection level at the first wave as it is the case with incident wave switching, but only after one or more reflections have arrived from open ends. This makes it possible to use cables with a very high DC-resistance and/or free topology. The speed must however be reduced corresponding to the distance the signal must travel back and forth until it is safely detected, which is at least two times in case of free topology without closed loops, but can be up to for example 16 times in case of a series resistance of 960 Ω on a 45-Ω line as shown and described in annex A. A traditional resistor terminated line would under these circumstances attenuate the signal 22 times no matter the speed and therefore hardly work.

1.2 Cable Types

Max-i does **not** need special communication cables, but may use standard, unshielded installation cables, which are available in very big cross-sections. The cables may be balanced, semi-balanced or unbalanced depending on the application and the cables do not need to be shielded, but unshielded cables, which are drawn in parallel with other cables or in a conductive cable duct, shall be balanced to prevent mutual coupling, conversion of common mode noise to differential mode noise and/or changes in line impedance!

For low demand applications or in cases where it is not possible to use a fully balanced cable or a good DC connection to ground is wanted, the cable may also be semi-balanced by means of a current balanced common-mode choke (balun) between the power supply and the line (on L+ and L-) as shown below:

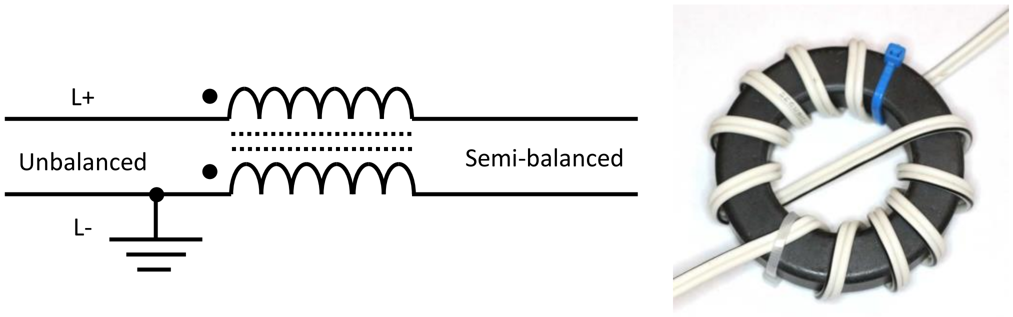


Fig. 1.4

This makes the cable balanced at high frequencies where mutual coupling between cables creates the biggest problems, but unbalanced at DC. It may be enough with a balun made by means of a 2-conductor cable wound on a high-permeability toroid core as shown. This winding method has the advantage that the input and output becomes opposite to each other, which reduces the coupling. It may even be enough with a number of ferrite beads with a small spacing between on the cable. To some extent, the line itself also works as a balun due to the self-inductance, so any mutual coupling between cables is higher close to a grounding point than further out.

Use of cables with 5 or more conductors or 4-wire cables with a yellow-green protective earth conductor (4Gnn) is not allowed for balanced cables, but may be used in case of for example unbalanced flat cables.

Note that in case of balanced cables and floating devices, nothing is gained by using a shielded cable! This saves a lot of trouble during mounting, and it avoids the usual dilemma with the cable shield. For the sake of noise immunity, a shield must be connected to ground/chassis in both ends, but it is simultaneously a ***very*** *bad idea to establish potential equalization (and ground loops) between different parts of a plant through a thin fieldbus cable – especially in plants, which use TN-C net with a common neutral and protective earth. Nevertheless, it is common practice with almost all RS-485 and CAN-based fieldbus systems!*

If mutual coupling is not a problem, the cable may be unbalanced **as long as the grounding is only done in one point or to a common power rail** so that the voltage difference between the grounding points is always negligible no matter the currents in the system including AC currents from motors etc. **Unbalanced** cables save one communication conductor, increases the line impedance, avoid high common mode signals on the power inputs of the connected devices during communication and makes it possible to connect a Max-i controller directly to most RS-232 ports without galvanic separation.

It is **not** allowed to mix balanced or semi-balanced cables with unbalanced cables except for short 3-wire spur cables on a balanced or semi-balanced 4-wire line.

Usually, balanced lines are implemented by means of twisted pair cables and H-bridge drivers, which drives both lines with opposite phase, and twisting may also be necessary in case of a 2-wire line without L+, but because twisted installation cables are not available, and a H-bridge would give a way too high communication voltage if used above 15 V, a balanced line is instead implemented by means of a cross-coupled 4-wire line where two conductors (COM) are connected directly together as shown to the left on the drawing below:

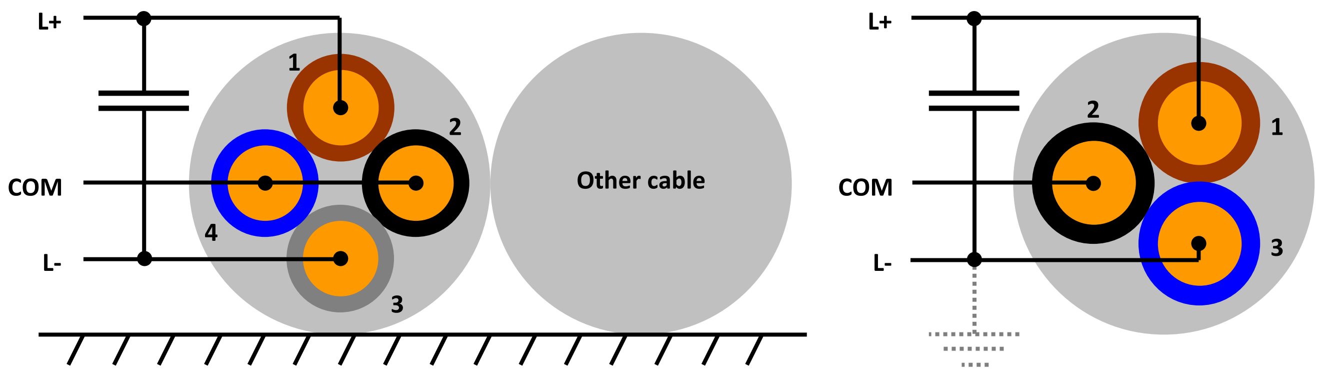


Fig. 1.5

A 4-wire balanced line has many important advantages compared to the usual twisted pair line or coax cable:

- *It does not need to be twisted to keep it balanced and prevent mutual coupling to other cables because the capacitance from each conductor pair to the mounting surface or other cables is almost constant no matter how the cable is rotated. This makes it possible to use standard installation cables, which are relatively cheap and available in very high cross-sections and in function safe and fireproof types if needed. This makes it possible to transfer an almost unlimited amount of power to the devices.*
- *Because the cable both carries communication and power, there is no need for a galvanic separation as it is often the case if one cable – typical a twisted pair is used for communication and another (thicker) cable is used for power.*
- *The power supply is separated from the communication so that it is not necessary to separate DC and communication by means of inductors or constant current generators and there is no load on the communication. In this way, it may be possible to connect several hundred devices to one transmission line, and it is not necessary with a serial inductor in the power supplies, which makes it possible to connect a battery directly and utilize its low AC and DC resistance.*
- *The relatively low impedance increases the transmitter current and with that the signal/noise (S/N) ratio and the number of possible devices on one line. The disadvantage is that bigger output transistors are needed, but this is a low price to pay for this and as shown later, approximately half of the drive power is not lost, but returned to the supply rails through the clamp diodes, and because Max-i does not use termination resistors, the energy loss due to communication may even be lower than many bus systems with **much** lower transmitter power.*
- *Only two output transistors are needed. A twisted pair line needs a 4-transistor H-bridge. If a 50 Ω, 4-wire line is compared to a 120 Ω, twisted pair line, the necessary chip area must be 1.44 times bigger for the H-bridge if the power dissipation in the output stage shall be the same because the voltage swing is twice as high and there are always two transistors in series.*
- *With a 4-wire line, there are always two conductors to carry the same communication signal. If redundant power supplies are used or the cable is supplied from both ends as shown previously, the bus can therefore survive a failure on one or even two conductors if the DC-resistance is sufficiently low and the two communication conductors are connected together in at least all outlets. This increases the failure tolerance considerably and makes Max-i very well suited for military applications. Note that the bus can survive a failure on any two neighbor conductors, which are much more likely than two opposite conductors!*
- *Due to the skin effect, the AC resistance of two conductors in parallel is approximately 0.7 times the AC resistance of a single conductor with the double cross section.*

Many properties of Max-i are due to the unique cross coupling of a 4-wire line. Therefore, the Max-i logo and the name (x = Cross-coupled) also refer to this.

For high current applications, it is a waste of cobber with 4 equal conductors and it increases the cable thickness and stiffness. Besides, a little resistance will damp unwanted reflections, so **very** thick communication conductors are in fact a slight disadvantage. It is therefore appropriate to make special Max-i cables with two thick and two thin conductors and enhanced color coding as shown below for HD-308 S2:

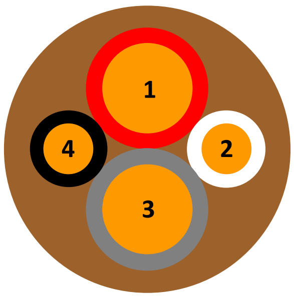


Fig. 1.6

The conductors in a Max-i cable are numbered 1 – 4 or 1 – 3 where conductor 1 always supplies the positive voltage (L+) and conductor 3 (L-) always supplies the negative voltage. The power supply and decoupling capacitors in each device ensure low AC and DC impedance between conductor 1 and 3. Conductor 2 (and 4) is used for communication and shall in case of a 4-wire line be connected together in all outlets and devices. In the following, the voltage level on these conductors is just referred to as the COM voltage, which is measured in relation to the midpoint between L+ and L- – even for unbalanced lines.

The color code for the conductors shall follow the national standards for DC systems. The codes for the Cenelec HD-308 S2 standard for DC systems (Annex 7), which is used in the EU and countries like United Kingdom, Israel, China, Russia, KSA, UAE and South Africa, and the National Electrical Code (NEC), which is used in the United States, India, Pakistan and some parts of Australia and New Zealand, are shown below:

Con-ductor	**HD-308 S2**		**Max-i**	**M12, M8**	**Balanced**			**Negative ground**			**(Positive gnd.)**	
	3-phase	**DC**			**HD-308 S2**		**NEC**	**HD-308 S2**		**NEC**	**HD-308**	**NEC**
1	**L1**	**L+ or M**	**L+**	**+**	**Red**	**Brown**	**Red**	**Red**	**Brown**	**Red**	**Blue**	**White**
2	**L2**		**COM**	**NC**	**White**	**Black**	**Gray**	**Black**		**Gray**	**Black**	**Gray**
3	**L3**	**M or L-**	**L-**	**-**	**Gray**		**Black**	**Blue**		**White**	**Gray**	**Black**
4	**N**		**COM**	**NO**	**Black**	**(Blue)**	**Gray**					

Fig. 1.7

Any **protective** ground, that is, a conductor that is grounded, but normally does **not** carry load currents, shall always be green/yellow or green.

If a conductor is grounded either directly or through a coil (balun), but carries load current, it is called “neutral” and shall be blue if HD-308 is used and white for NEC. **Note that blue does not mean negative, but (near) ground potential!** A supply conductor more positive than ground shall be brown or red for HD-308 countries and red in case of NEC and special Max-i cables, and a supply conductor more negative shall be gray for HD-308 and black for NEC. The communication (COM) conductor(s) should be white and black for HD-308, so that the cable may also be used for sensors with one normally closed (NC) contact and one normally open (NO) contact, but as blue represents a midpoint, one black and one blue conductor is acceptable if a balanced line must be made by means of a standard 4-wire HD-308 cable as shown previously. The COM conductor(s) should be gray for NEC. If special cables are manufactured for Max-i, the outer jacked should be ochre brown according to RAL 8001 (RGB 157-98-43, CMYK 30-60-90-10).

The colors of the Max-i logo refer to the preferred colors of the conductors for special Max-i cables – red, black on white background or white on black background and gray.

Note that the colors for an unbalanced cable with negative ground are the standard colors for a 3-wire cable without protective earth. This is very practical for smart-house applications. In case of for example automotive applications, the blue (or white) conductor may with some limitations be replaced with the chassis (ground).

The color code for positive ground is only shown for reference. Max-i **always** uses negative ground in case of unbalanced or semi-balanced cables.

The reason for this is a sequence of more points:

1. *Because electron mobility in silicon is approximately 2.4 times higher than hole mobility, it is most appropriate to do all power switching by means of N-channel MOSFET's or NPN-transistors as this reduces the necessary chip area (and price) approximately a factor 2.4 compared to a P-channel solution and gives much better data.*
2. *To avoid charge pumps, it is most appropriate to use low side switching (open drain or open collector). In a traditional system, this has the disadvantage that a short circuit between the output and ground will activate the actuator, but since it is intended to embed the Max-i controller directly in the actuators (and sensors) so that the output signal is not taken out, this is not a problem.*
3. *Low side switching makes it possible to use many different voltage levels and fuses in the same system like automotive 14 V (12 V), 21 V (18 V), 42 V (36 V) and 56 V (48 V) applications, but in this case, there is only one level, which all devices have in common and is never disconnected in case of a blown fuse, and this is L-, so the system must have negative ground.*
4. *To avoid level shifters on the output side, the low-voltage logic, which drives the output transistors, must have the same ground potential as the sources or emitters of the output transistors, that is, L-. In case of logic with more voltage levels like 5 V, 3.3 V, 1.5 V etc., they all share L-.*
5. *To avoid level shifters on logic signals and on communication channels without galvanic separation such as RS-232, RS-485, CAN, SPI etc., L- (logic ground) must have the same potential as ground/earth in connected systems and the de-facto standard for this is negative ground. This is especially important when a Max-i controller is embedded in other electronics for example together with a microprocessor or in case of an unbalanced DC net in smart-house applications where the net may be connected directly to PC's etc., and ground is the wire/shield, which is used to lead transients and noise to earth.*

The disadvantage of using negative ground is that power switching, switch mode buck converters and power limiters like electronic fuses need to use P-type transistors or use a charge pump. However, the Max-i controller may have a square-wave output, which makes it easy to make charge pumps by means of a few diodes and capacitors. Besides, ideal diodes made by means of MOSFET's require a P-channel transistor for positive ground and an N-channel transistor for negative ground, which is better, and in both cases, a charge pump is needed. With negative ground, a combined power switch and ideal diode for example for solar panels can be made with a dual N-channel transistor and a common charge pump.

The conductors shall be copper or aluminum. **Lead-free** tinned copper should not be used because of the risk of tin whiskers, which may cause a short circuit and maybe initiate an arc.

For low-power applications, it is also possible to use two pairs of a twisted pair cable. In this case, one of the conductors in each pair shall be used for power supply and the other for one of two connected communication conductors as shown below for pair 1 (blue) and pair 2 (orange) except that the major color is shown black and not white as it is according to the color standard:

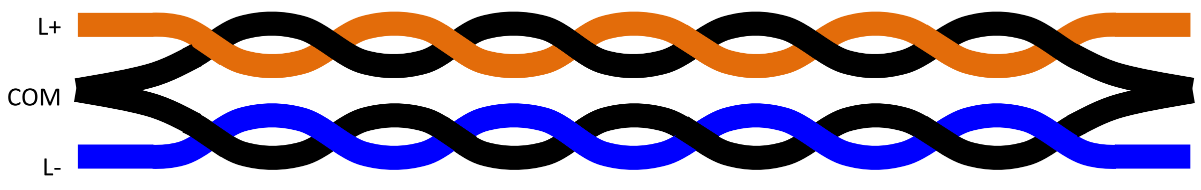


Fig. 1.8

It is not allowed to use one pair for power supply and the other pair for communication.

Since the characteristic impedance of each pair is approximately 120 Ω and there are two pairs in parallel and some mutual coupling, the characteristic impedance of such a line is close to 50 Ω and therefore very much the same as a balanced 4-wire line.

Due to the fairly high DC-resistance of thin twisted pair cables, it shall be verified that the voltage never falls below the minimum supply voltage **even with communication currents up to 300 mA**, but if the supply voltage is adjusted close to the maximum value, a very big voltage drop is allowed as shown later.

For smart-house and building applications where there is very little coupling to other cables, it is also possible and even recommended to use an unbalanced 5-conductor industrial flat cable, which may be used not only for 20-Vdc applications, but also for Max-i combined with 230 Vac or 115 Vac – the so called mains plus dimming applications. This is especially useful in big buildings, office buildings and halls where the necessary power for lighting, window openers, sun shielding etc. is too big for 20 V. In this case, the L+ conductor should be red, the phase conductor shall be brown and conductor 3 should be marked and may be used as protective ground (yellow-green) as shown below:

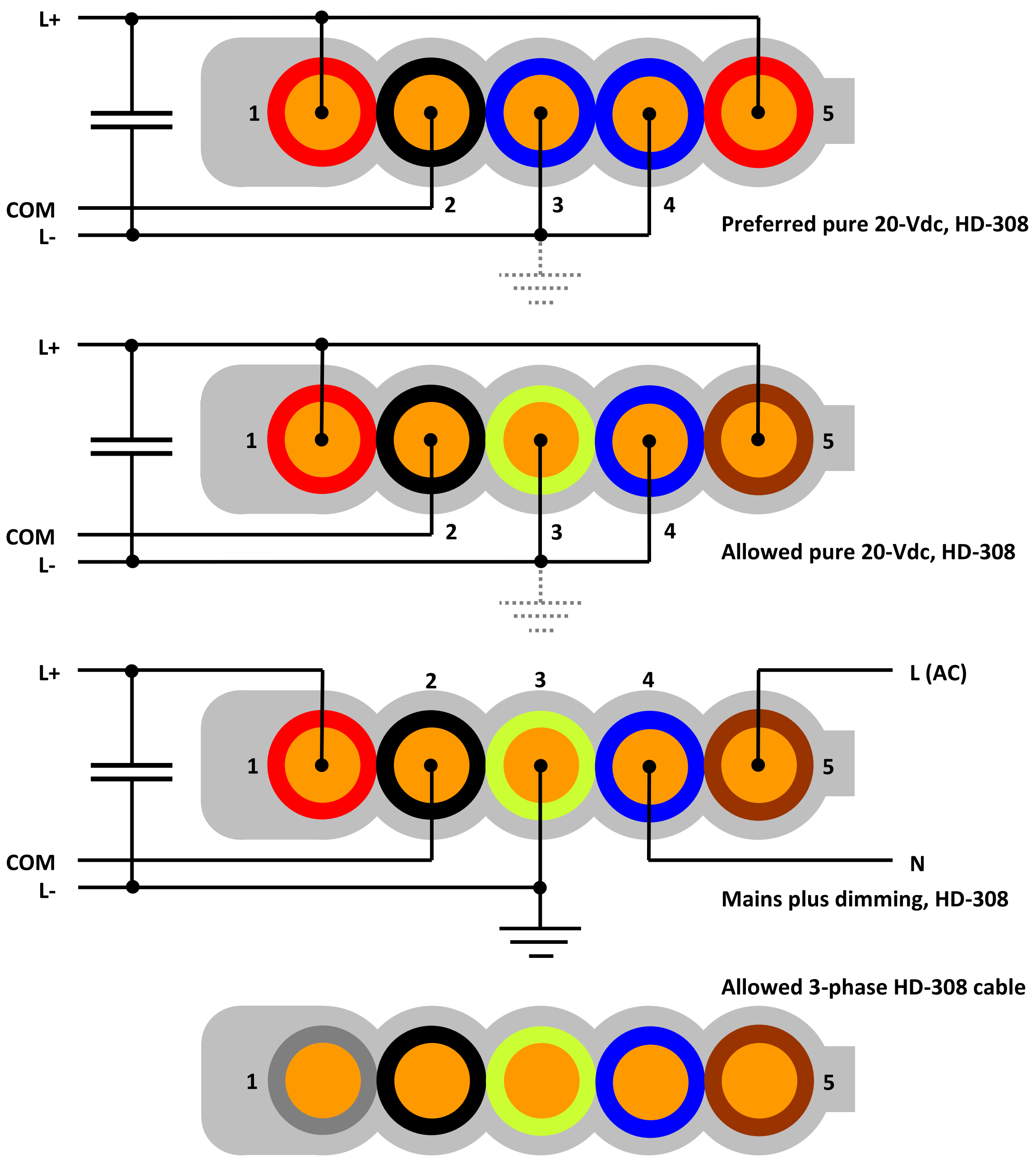

L+
1
5
COM
L-
2
3
4
Preferred pure 20-Vdc, HD-308
L+
1
5
COM
L-
2
3
4
Allowed pure 20-Vdc, HD-308
L+
2
3
4
L (AC)
1
5
COM
L-
N
Mains plus dimming, HD-308
Allowed 3-phase HD-308 cable
1
5

Fig. 1.9

Conductor 1 still carries the positive voltage for Max-i, conductor 2 carries the COM voltage and conductor 3 the negative voltage, but to be able to utilize all conductors in the cable equally in a pure 20-V system, conductor 1 and 5 and conductor 3 and 4 shall be connected together in all power supplies and in all outlets. This has the beneficial side effect that the coupling capacitance between L+ and L- is increased, which is important at high frequencies where the series inductance of the decoupling capacitors becomes significant.

In case of mains-plus-dimming, L- uses the protective ground. This is allowed because in this case the majority of load currents go through the (blue) neutral conductor on the mains side.

If a special cable is not available for pure 20-V applications (upper drawing), it is also allowed to use the conductor colors for mains plus dimming applications (lower drawing) or a standard 3-phase cable as shown, but a cable with two red and blue conductors for pure LV DC systems shall never be used for mains plus dimming.

A flat-cable solution has the great advantage that spur cables can be applied with tapping (insulation piercing) technique so that it is not necessary to cut the cable for every outlet, sensor or actuator. Suitable flat cables with tap-off adapters are for example available from Woertz, Wago (Winsta® IDC) and Wieland (gesis® NRG) in 5 × 2.5 mm^2 and 5 × 4 mm^2 with a cable dimension of only 24.1 × 6.3 mm. Unfortunately, none of these readily available cables have the specified conductor colors and order of signals – not even cables intended for mains plus dimming applications like KNX and DALI, but since the cable cannot be mounted wrong, this is not so important.

To be able to distinguish pure 20-V systems from mains plus dimming systems, the jacket color shall be different (jacket colors not shown on drawing). If special cables are manufactured for Max-i, the jacket shall be ochre brown for pure 20-V systems and black or gray for mains plus dimming.

The reason why the jacket shall have a different color for mains plus dimming systems is that the colors of the conductors are not visible when tapping/piercing technique is used.

In case of street lighting, it may be very desirable to use a single round cable for mains plus dimming, but it must then be ensured that transients on the high voltage lines due to changes in the load does not couple to the communication and 50 or 60 Hz couples as little as possible. If there is no mutual coupling to other cables so that a good balance is not needed, this may be done by means of a 5-wire line with center core and without earth conductor as shown below:

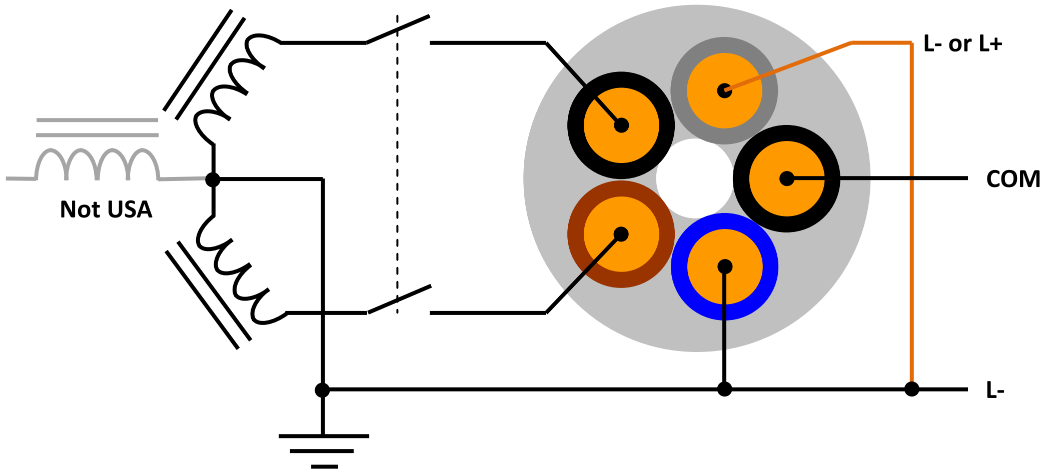


Fig. 1.10

Any earth connection shall instead be done directly from the light poles to ground and it shall be insured that the insulation between grounded parts and live parts is able to handle the maximum lightning transient voltage. A transient suppressor to ground from any of the cable conductors is **not** recommended as it will convert common mode transients to differential mode and in this way do much more harm than good!

In the USA where it is very common with a 2-phases distribution system with a phase difference of 180°, there is no 60 Hz coupling from the high voltage conductors as the voltage on the two lines cancel out 100 %. In Europe where the phase difference is 120°, the net voltage is reduced to the half of the voltage of a single phase system so that it is 115 V_{rms}, but this lower voltage is further shielded by the blue and gray conductors so that the net result is acceptable.

The connection to the gray conductor may either be L+ if the cable supplies the power to the Max-i controller or L- if a local galvanic separated power supply is used in each light fixture. If special cables are manufactured for Max-i, it is recommended to replace the gray conductor with either a blue for an L- connection or a red for an L+ connection, and the black phase conductor should then be gray to better distinguish it from the black COM conductor.

Note that the street lights are supplied from two phases and not one phase plus neutral as it is usually done! There are more important advantages of this:

- If a lightning transient couples to the cable, the induced voltage will be the same in all conductors so that there will neither be any differential mode transients on the power supply for the lighting source nor on the communication. In usual street lighting, the impedance from a phase conductor and the neutral conductor to earth is usually not the same, so relatively harmless common mode transients may be converted to differential mode, which may destroy the electronics in LED fixtures. There are numerous examples of that!

- When the load on the two high voltage lines changes, the voltage on one line will be temporary increased and the voltage on the other line will be correspondingly decreased. The net result is no coupling except for a reduced 50 Hz coupling in Europe.
- It is possible to use a cable with a smaller cross section due to the higher voltage. The saved copper may easily pay for the slightly more expensive power supply due to the higher voltage and galvanic separation to Max-i.

⚠ **No matter which cable is used, it shall be ensured that the voltage drop due to loading is (almost) the same in the L+ conductor(s) and the L- conductor(s).** This is very important for the communication. In practice this may be obtained in two ways:

- By means of the same cable cross section for L+ and L-.
- By means of a serial resistance in the power supply line, which has the lowest resistance. If this solution is chosen, the difference in resistance between the L+ conductor(s) and L- conductor(s) shall be less than ±10 % after compensation. If for example a number of LED lamps are connected to a number of serial connected, 1-phase (two cobber conductors) aluminum track rails, where the aluminum rails themselves are used as L-/ground, L- will usually have lower resistance than L+. In this case, small power resistors like a short piece of low-conductivity metal shall be added in series with L- at regular distances to keep the difference in resistance below the limit in any place. If a rail is designed especially for Max-i, the resistor(s) should be an integrated part of the rail so that it can be mounted and connected without any precautions.

It is **not** allowed to use a common chassis for L- except for line lengths at least 2 times shorter than the maximum length at the given speed and less than 1 V voltage drop in L+ at maximum load (automotive applications).

1.2.1 Explosive Areas

By means of High Power Trunk Concept – HPTC, Max-i may also be used in hazardous locations. These are classified into zones to facilitate the selection of appropriate electrical apparatus as well as the design of suitable electrical installations. Information and specifications for the classification into zones are included in for example IEC 60 079-10 and the European ATEX 118a directive. Max-i may be used for Zone 1 (previously category 2G/2D) operation, that is, a place in which an explosive atmosphere consisting of a mixture with air of flammable substances in the form of gas, vapor or mist is likely to occur in normal operation occasionally. This zone includes for example gas stations and tank trucks, but also some parts of feed mills where explosive dust may sometimes occur.

The equipment is classified into groups (and temperature classes) in accordance with the properties of the explosive atmosphere for which it is intended.

- Group I: Electrical apparatus for mining.
- Group II: Electrical apparatus for all remaining potentially explosive atmospheres.

Group II is further divided in three classes A, B and C according to the gas type (and temperature):

- Class A: Acetone, ethane, ethyl ethanoate, ammonia, benzol (pure), ethanoic acid, carbon oxide, methane, methanol, propane, toluene, ethanol, i-amyl acetate, n-butane, n-butyl alcohol, benzene, diesel fuel, aircraft fuel, heating oils, n-hexane, acetaldehyde and ethylether.
- Class B: Coal gas (lighting gas) and ethylene.
- Class C: Hydrogen and acetylene.

Max-i is primary usable for group IIA and IIB, but may under some circumstances also be used for IIC.

The type of protection is specified with a letter:

- d: Flameproof enclosure in accordance with IEC 60 079-1 or EN 50 018.
- e: Increased safety in accordance with IEC 60 079-7 or EN 50 019.
- p: Pressurized apparatus in accordance with IEC 60 079-3 or EN 50 016.

- i: Intrinsic safety in accordance with IEC 60 079-11 or EN 50 020, that is, apparatus contain intrinsically safe electric circuits only. An electric circuit is intrinsically safe if no sparks or thermal effects are produced under specified test conditions (which include normal operation and specific fault conditions) which might result in the ignition of a specified potentially explosive atmosphere.

  Intrinsically safe electrical apparatus and intrinsically safe components from related equipment are classified according to categories ia or ib. Equipment from category ”ia” is suitable for use in Zone 0, and equipment from category ”ib” for use in Zone 1.
- o: Oil immersion in accordance with IEC 60 079-6 or EN 50 015.
- q: Powder filling in accordance with IEC 60 079-5 or EN 50 017.
- m: Encapsulation in sealing compound in accordance with IEC 60 079-18 or EN 50 028.

HPTC is based on the fact that “intrinsic safety“ with its power limitations is really only needed at the device connection where actions such as maintenance or replacement has to be carried out during plant operation, but these actions are usually not required on Max-i. The HPTC thus utilizes a trunk, which is protected using increased safety (Ex e) ignition protection: Working on the trunk requires a hot work permit, which is the trade-off for increased power levels.

Max-i devices may be connected to the trunk with or without spur cable in three ways depending on the protection:

- m: Devices encapsulated in sealing compound may be connected directly without any limitations.
- m + i: In this case, the Max-i controller is located and sealed in the junction box, and the I/O connections are intrinsically safe, which is especially simple in case of inputs such as NAMUR where very little power is needed.
- i: Devices, which are not protected, must be connected to the trunk trough field barriers or other couplers, which provide the necessary power-limiting explosion protection at the spur connections. Such barriers may limit the current to for example 40 mA, but as described later, the transmitter current of Max-i is usually much higher. However, because Max-i is not terminated with resistors, the transmitter only draws current for the time it takes the signal to travel to the other end of the line and back again so even though the transmitter power is very high, the energy may be reduced by means of a high communication speed with low duty cycle and/or a short line.

The power supply is located in zone 2 and may supply the line with more amperes.

1.2.2 Characteristic Impedance

The characteristic impedance of the cable including load capacitance shall be ≥35 Ω. This is usually fulfilled for all cable types – even with PVC insulated conductors. However, if maximum line length is wanted and/or a high load capacitance is used such as LED ropes, it is necessary to use conductor insulation with a low relative dielectric constant/permittivity (ε_r) such as a polyolefin type like PE, PEX, FRPE or PP. The outer jacket of the cable may be based on any material – even PVC.

Various polyolefin types are widely used in many of the so-called “green” halogen-free installation cables. Because of the added flame retardant ceramics in FRPE like aluminum hydroxide ($2\ Al(OH)_3$ + heat → $Al_2O_3 + 3\ H_2O$) or magnesium hydroxide ($Mg(OH)_2$ + heat → $MgO + H_2O$), FRPE has a higher permittivity (ε_r approximately 2.6) than pure PE ($\varepsilon_r = 2.3$). With $\varepsilon_r = 2.6$, the propagation delay of the cable is approximately 5.4 µS/km, and with a typical capacitance of 110 nF/km for a 4-wire line, the characteristic impedance Z_0 is approximately 50 Ω.

PVC is a polarized material with a relatively high dissipation factor of approximately 0.016 compared to less than 0.0002 for PE and less than 0.0005 for PP. This causes both a power loss and an increased rise time.

The power loss may be calculated as:

$$P = 2\pi \times f \times V^2 \times C \times D$$

Where f is the frequency, V is the RMS voltage, C is the capacitance and D is the dissipation factor. For a maximum length cable, the power dissipation becomes approximately 3.2 D = 0.6 mW for PE, but 50 mW for PVC. 50 mW corresponds to the loss in a 2.2 kΩ resistor or 46 standard devices, which is acceptable.

1.2.3 Resistive and Capacitive Load

The maximum load on the COM conductor(s) due to input resistance, leakage currents and capacitance has not yet been finally determined, but **preliminary** data, which may change without notice in future versions of this specification, is shown below:

Line Type	Maximum Resistive Load			Maximum Capacitive Load			
	Cross Section			Randomly distributed Average	Equally distributed	Single Point	Power Supply
	Minimum	× 2	≥ × 4				
40 – 50 Ω, 4-wire and Coax	200 Ω	140 Ω	100 Ω	50 pF/m	100 pF/m	1 nF	10 nF
3-wire	TBD	TBD	TBD	TBD	TBD	TBD	TBD
120 Ω, 2-wire	500 Ω	350 Ω	250 Ω	20 pF/m	40 pF/m	400 pF	4 nF
5-wire flat cable	TBD	TBD	TBD	TBD	TBD	TND	TBD

Fig. 1.30

A standard DC load is defined as 100 kΩ, which corresponds to a current of approximately 0.13 mA at the nominal voltage swing of 13 V, which is less than the hold current. A standard AC load is defined as 25 pF.

The maximum resistive load depends on the conductor cross section, but is independent of the line length so that up to 1000 standard DC loads may be connected to a 4-wire line if the cross section is at least 4 times bigger than minimum.

Because of the skin effect, the maximum resistive load is inversed proportional to the circumference of the conductor and not the cross section.

The maximum capacitive load depends on the line length, as it is equivalent to an increased permittivity. A capacitive load has two important effects as described in Annex A – Transmission Lines:

- It lowers the line impedance as $Z_0 = \sqrt{(L / C)}$ and therefore causes more signal attenuation for a given serial resistance.
- The propagation delay is increased corresponding to a reduced characteristic impedance as $t = \sqrt{(L \times C)} = L / Z_0$. **This reduces the maximum line length, which must be taken into consideration!**

The maximum length due to capacitive loading may be calculated as:

$$K = K_0 \sqrt{\frac{2.25 \times C_0}{\varepsilon_r \times C_L}}$$

Where K_0 is the maximum length specified in table 1.21 later is the relative permeability of a reference cable with a propagation delay of 5 ns/m (3.333 × √2.25), C_0 is the capacitance of the line without any load and C_L is the capacitance of the line in loaded condition. If for example C_0 is 108 pF/m and the maximum load of 50 pF/m is applied to a polyolefin insulated cable with a relative permittivity of 2.7, the maximum length becomes 0.755 K_0. If the capacitive load is the same (50 pF/m), but a PVC insulated cable with a relative permittivity of approximately 4.7 – 5 is used instead, the maximum length becomes 0.56 K_0.

The reduction in line length is equivalent to the reduction in characteristic impedance. If for example the maximum randomly distributed capacitive load is applied to a 4-wire line, the impedance is typical reduced from approximately 48 Ω to: $48 \times \sqrt{(108 / (108 + 50))} = 40$ Ω. The maximum line length is therefore reduced with a factor 40 / 48 = 0.83. In the same way, the maximum equally distributed capacitive load reduces the impedance to the minimum 35 Ω and reduces the maximum line length a factor 0.73. If the maximum length of a 4-wire line with few devices is for example 1 km, it is therefore theoretically possible to connect up to 1660 randomly distributed standard AC loads to a 830-m line and up to 2920 loads to a 730-m line if they are equally distributed such as LED ropes and ladders.

To avoid a very low capacitance in one point, which may cause reflections of negative spikes, there is a limitation on the maximum capacitance in a single point. For this reason, no more than 2, 4-m spur cables (2 × 440 pF) shall be connected to the same coupling box on a 4-wire line and the distance between coupling-boxes shall be so big that the maximum average load is not exceeded. In many applications, it may be advantageous to use 3-wire spur cables on a 4-wire line to reduce the capacitance and decrease the thickness of the cable. Note that the maximum single point load of power supplies may not be possible on short lines as it may exceed the maximum random load for that line length.

Single point capacitive loads work as a low impedance (short circuit) to the high frequency components in the signal, which will be reflected corresponding to a short circuit. This is typical noticeable as a short voltage drop/notch in the signal just after the period where the transmitter is driving the line. However, a fairly big receiver hysteresis and high voltage levels prevent that this notch is detected as a new line state so in practice, Max-i can accept a fairly high single-point load capacitance – typical more than 47 nF, which gives a good safety margin even for power supplies where approximately 5 nF must be acceptable if it should be possible to design the supply in a reasonable way.

1.3 Cable and Outlet Protection

All line segments shall be protected in each end, which receives power, by means of fuses or non-adjustable, non-autoreset, electromechanical devices according to IEC/UL 60950 and according to the cross section of that particular segment as shown in the table below:

Cable type	**Cross section in mm^2 at fuse size.**				
	10 A	**15 A**	**20 A**	**25 A**	**32 A**
Unfused device cables from an outlet to a device	0.5	0.75		1.5	-
Fixed cables incl. spur cables in thermal uninsulated walls (normal cooling)	1.5	1.5	2.5	4	6
Fixed cables in walls and ceilings with thermal insulation on three sides	1.5	2.5	4	6	8

Fig. 1.11

The table is based on a maximum temperature raise of 40°C and copper cables (see Annex D for further information).

Fixed cables shall have a cross-section big enough to handle the maximum current under the expected duty cycle and cooling conditions.

Device cables are short spur cables from fixed installation outlets to devices like LED lamps, battery chargers, actuators, sensors etc. They shall fulfill the following requirements:

- Have a cross section big enough to blow the nearest fuse(s) in case of a **short circuit**, but it is not necessary that it is able to handle the maximum current continuously.
- Have a sufficiently low resistance to enable a minimum short circuit current of 2 times the maximum current of the nearest fuse at the lowest voltage so that it is possible to blow the fuse within a reasonable amount of time. If for example the supply voltage is at the minimum 15.4 V as specified later and a 5-A fuse is used, the loop resistance shall be less than 1.5 Ω **including all fixed cables**. This corresponds to the loop resistance of 10 m 0.25 mm^2 wire and since longer spur cables are not allowed, even fairly thin device cables may be used.

All fuses should be supervised and an alarm should be generated in case of a blown fuse.

IEC/UL 60950 Table 2C requires that a circuit breaker acts within 120 seconds at 210 % and that the short circuit current is limited to 1000/U_{ov} (50 A at 20 V) within 60 seconds after the application of the load, but such a slow breaking characteristic is not appropriate for semiconductor technology like Max-i and may also create an unnecessary big arc and fire risk. Therefore, it is **highly recommended** to use a **much** faster fuse or circuit breaker such as a standard automotive ATO blade fuse according to ISO 8820-3 or SAE J1284 that breaks within 5 seconds at 200 %, within 50 seconds at 160 % and within 10 minutes at 135%. An ATO fuse has a breaking capacity of 1 kA at 32 Vdc and more kA at 20 Vdc, which is enough even in case of a battery.

Note that it is necessary to fulfill some termination criterions to make the communication work **even in case of a blown fuse**. This is described in the chapter: “Clamp Termination”.

Electronic fuses (eFuse) based on semiconductor technology such as MOSFET's shall **not** be used for cable protection, but are allowed or even necessary for **secondary** protection of outlets.

There are three types of outlets:

1. Outlets in for example walls, which may only be **temporary** connected to for example battery chargers and computers. This type is further divided in two groups depending on the power level:
    a. Limited power source (LPS) outlets for devices up to 90 W and 7.5 A.
    b. Outlets for devices above 90 W or 7.5 A.
2. Outlets in for example the ceiling or a utility room, which are **permanently** connected to a devise, like lamps, alarm sensors, door locks, window openers, sun shielding and energy management systems.

1.3.1a LPS Outlets

These outlets shall be limited current sources or limited power **and** current sources (LPS) according to the IEC/UL 60950-1 safety standard. In this way, "*the maximum apparent power and fault current available on the outlet under any load conditions including a short-circuit are limited to magnitudes not likely to cause ignition under fault conditions in connected equipment mounted on, and constructed from, suitably rated materials*".

The limits are either:

- 5 A for $U_{oc} \leq 20$ V, and $\leq 100/V_{oc}$ A for 20 V < $U_{oc} \leq 30$ V. The overcurrent protective device shall be a fuse or a non-adjustable, non-autoreset, electromechanical device that breaks within 120 s at a load of 210 % of the specified, and the maximum apparent power shall be limited to 250 VA, 60 s after application of the load.
- ≤ 100 VA and ≤ 8 A up to 30 V as shown below. The limits shall be fulfilled 5 s after application of non-capacitive load if protection is by an electronic circuit or a positive temperature coefficient device, and 60 s in other cases.

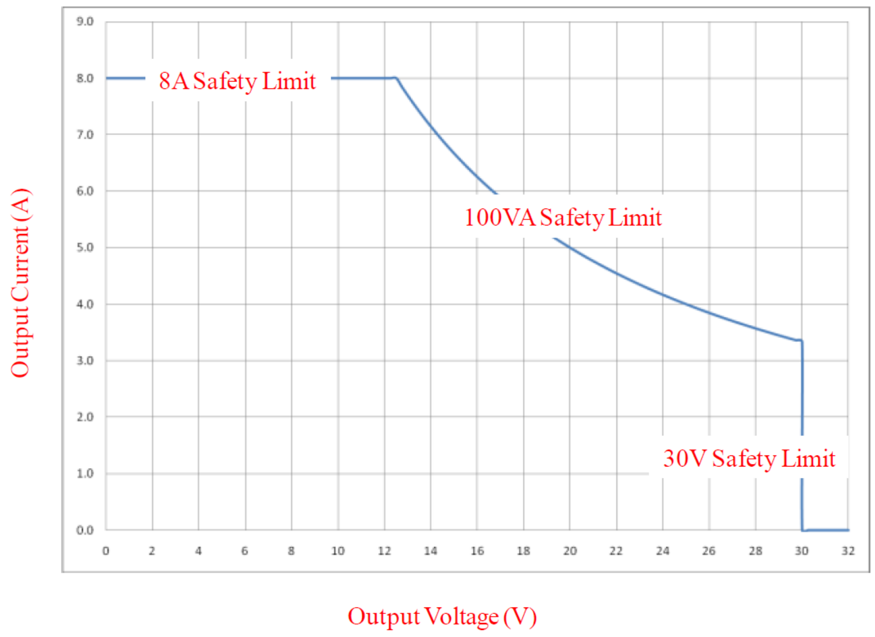


Fig. 1.12

In practice, the two methods are equal at 20 V and above, but the power limiter is preferred since it allows a much higher current at voltage levels below the maximum voltage of Max-i, which is 25.2 V, where a 4 A fuse must otherwise be used.

The 100 VA (W) is the reason why USB Power Delivery up to version 3.0 and Quick Charge up to version 4+, which both use 20 V, are limited to 5 A.

The maximum voltage (30 Vdc = SELV) makes it possible for the equipment to work even in bathroom zone 0 (shower etc.) except if big parts of a body is immersed into water. In that case, the maximum voltage shall be limited to 15 V, but otherwise the maximum voltage of Max-i, which is 25.2 V as described later, fulfills this criterion.

⚠ **If the maximum power or current limit is exceeded, the LPS shall disconnect the output within 5 second (electronic limitation) and shall not connect it again before the load has been disconnected!**

It is very important that the circuit is disconnected ***permanently*** *until a manual reset or a fuse replacement in case of an overload. If this is not the case, the current could overheat the cables, melt the insulation and even cause fire.*

Since this type of outlet includes an electronic fuse and some electronics to limit the power, it may be fairly easy to add inrush current limiting – often by means of a single capacitor. Therefore such an outlet shall also include an inrush current limiter, which can limit di/dt to maximum 5 A/ms **even in case of a short circuit** or a capacitive device.

As explained in chapter "Load Switching and Communication" in Annex A, the ***immediate*** *voltage drop or raise in voltage due to load switching is equal to the line impedance in the point of load multiplied with the change in current. If for example the point of load is located some distance from the ends of a 40-Ω line so that there are in effect two lines in parallel, the generator impedance is 20 Ω so an immediate load of just 1 A or bigger will cause the voltage to drop to zero temporary if the line voltage is 20 V! This is of course unacceptable as it may influence the communication and may even cause the devices on the line to reset.*

To reduce the consequence of a **reduction** in load current including an immediate cutoff, a serial connection of a capacitor and a small resistor to damp the ringing shall be added between L+ and L- as explained in Annex A. In case of a 5-A load, the capacitor should be at least 47 µF and the resistor should be in the order of 0.27 Ω. If a bigger capacitor is used, the resistor should be gradually reduced until the capacitor is ≥ 220 µF where the resistor can be entirely omitted.

1.3.1b Temporary Connected Outlets ≥90 W

Since it is not reasonable to dimension an outlet after unknown power and current requirements in the future, **devices** above 90 W shall include:

- A fuse dimensioned after the maximum current.
- An inrush current limiter, which limits di/dt to maximum 5 A/ms as for class 1a, but since it is anyway necessary with a lot of protection circuitry to reduce the fire risk as IEC 60850-1 is no longer fulfilled, this may not increase the price very much.

The outlet itself or the T-connection to the main bus shall include a fuse dimensioned after the cross section of the spur cable and the maximum current of the connector followed by at least a 220 µF capacitor between L+ and L- to reduce the line ringing when a device is disconnected or switched off.

Note that due to the increased fire risk, the connected device should not be left unattended.

1.3.2 Permanent Connected Outlets and Devices

In this case, any capacitance in the outlet and/or the connected device only generates current spikes during the initial power-up and therefore does not cause problems, so the outlet may just be a fuse protected clamp row as shown on fig. 1.1 and 1.2. The necessary capacitance to limit the voltage drops during current variations shall be included in the connected device, but it is often possible to reduce the value considerably by means of the output smoothing filter, which is integrated in the Max-i controller and may be used for example for lamps, motors and heaters.

Unless a higher current is needed, it is recommended that each outlet is protected by means of a 5 A fuse, which breaks within 60 seconds at a current of ≥8 A. This mimics a LPS quite well as neither the 8-A limit nor the 100-W limit is exceeded at 20 V or below after this time (an 8-A current limiter is not enough since it will pass 160 W through). As an automotive ATU blade fuse according to ISO 8820-3 / SAE J1284 breaks within 0,25 – 50 seconds at 160 % = 8 A for a 5-A fuse, such a fuse fulfils the requirements.

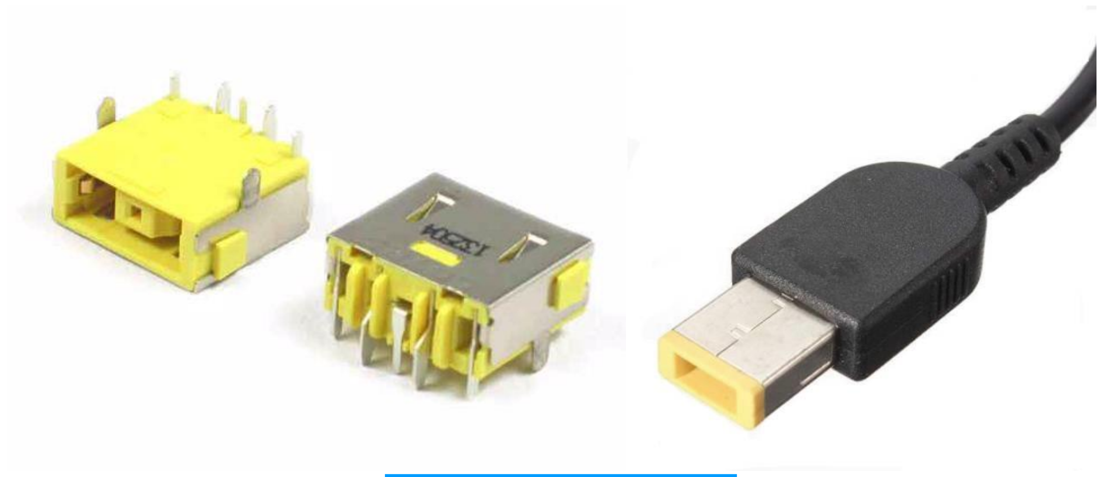
Note that 20 V is the lowest voltage where this is possible and therefore also a good choice for smart-house applications from this point of view. It is for example not possible to make a 100 W, 15-V ***fuse****-protected LPS for bathroom zone 0, as the maximum current would be 6.7 A and a fuse with that level uses too much time to break at 8 A or cannot break at all and would just dissipate power and heat up. Therefore, an electronic fuse must be used.*

1.4 Connectors

Any connectors on power supplies, outlets, hand terminals (with power supply) etc. shall be **female/socket** connectors. Any connectors on actuators, controllers, sensors, printers, SCADA systems and other device cables shall be **male/plug** connectors or coaxial female/socket connectors where L+ and COM are protected.

The preferred connector on industrial spur cables is M12 or M5. Because pin 2 and 4 shall always be connected together in all devices, conductor 2 and 4 may be interchanged so it is never necessary to twist the two conductors to connect a male or female connector.

For unbalanced or semi-balanced smart-house applications, the rectangular slim-tip Lenovo laptop connectors, which are shown below and for example available from TE Connectivity (jack part number 1-2129458-1, 1-2129458-2 or 1-2129567-1 and plug part number 1-2129334-1), are recommended:



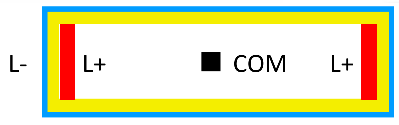


Fig. 1.13

These connectors are rated to 25 V, ≥7 A, which fits perfectly for Max-i, and use the outer part (shield) as L-, the two inner parts in the sides as L+, and in Max-i, the center pin is used for COM. Since L+ and COM are protected on the inside, the female part may be used both at the bus/power-outlet side and on connected devices so that interconnection cables have male connectors in both ends. These connectors further have the advantage that L- (ground) gets contact before L+, which gets contact before COM. This should also be the case if other connectors are used or manufactured for Max-i.

All outlets for temporary connected devices should contain a switch or a jumper, which can connect the center pin to either the COM line or to a resistor to ground, which indicates the maximum power of the outlet according to the Lenovo standard as shown below:

36 W	7300 Ω
45 W	120 Ω
65 W	280 Ω
90 W	550 Ω
135 W	1000 Ω
170 W	1900 Ω
230 W	4600 Ω

Fig. 1.14

This standard fits with Max-i, since a 100-W LPS should only be loaded with 90 W due to tolerances.

1.5 Power Supplies

The table below shows the voltage levels including any voltage drops on the line, which must be taken into consideration:

	Full range	Fully operational	Industrial and railway compatibility	PUR/PDR	Absolute maximum	
					Forward	Reverse
Voltage V	20 +26/-23% (15.4 – 25.2)	20 +26/-10% (18 – 25.2)	24 +5/-10% (21.6 – 25.2)	≈15.4/≤10.8	≥40	≤-30

Fig. 1.15

In case of battery operation, the battery shall be regarded as almost discharged, if the voltage falls below 18 V, and the load should therefore be reduced.

- All **temporary** connected devices like mobile phones, tablets, laptop computers and battery chargers, should reduce the maximum load current to 2 A (36 W) if the voltage is below 18 V when the device is connected or later goes below this level for more than 10 seconds. To avoid oscillation, the maximum current shall stay at this level until the device has been disconnected.

  *36 W is the same as the maximum **power** of USB Power Delivery profile 3 and Quick Charge Class A.*

  Many laptop computers have an 18 V UVLO. In this case, the battery charging should stops at 18 V, but the computer is still allowed to draw working current from the bus in such a way that the current in or out of the battery in the device is close to zero.

- All **permanent** connected devices, which may work at a lower power level such as LED lighting, should reduce their consumed current to less than 33 % of the nominal current at 20 V. For a typical LED lamp, it may be necessary to begin the dimming at approximately 19 V to reach this level at 18 V, where the eye will regard the light as having 63 % light intensity. The dimming shall be implemented by means of an analog current reduction in the LED driver.

  In this way, the Max-i controller does not need any brownout circuit (except for PUR) and the brownout dimming works no matter if the LED uses digital dimming or not or is driven directly from the bus voltage without a Max-i controller.

During power-up, a device shall reset all internal states and set all outputs low until the PUR voltage is reached, and during power-down, it shall reset again when the voltage falls below the PDR voltage. The PUR level shall be less than, but fairly close to the minimum voltage 15.4 V. In case of a switch-capacitor power supply, which divides the input voltage by 2, it is allowed that the PUR level is the level where the power supply is able to supply 5 V on an auxiliary power output so that a separate voltage monitor is not necessary, and it is also allowed that the PDR level is the level where the nominal voltage of the internal logic and all logic-level I/O's, which is typical 3.3 V, can no longer be maintained. If a Max-i controller includes an EEPROM, which can be used to save the hour counter during power down, the PUR level may be used to trigger the storage before the voltage reaches the PDR level where the EEPROM shall be write protected and therefore cannot be programmed.

The PDR level is set by the so-called cold crank voltage, which is the voltage, which may occur during engine start at low temperatures. Battery manufacturers define this to 7.2 V at 0 °C for a 12-V battery and 14.4 V for a 24-V battery, which corresponds to 10.8 V for an 18-V battery. However, at lower temperatures and/or old batteries, voltage levels down to 8 V may easily occur and since a new power-up reset cannot be expected after activating the starter, a separate, but small auxiliary battery may be necessary for automotive applications to supply the bus during engine start and ensure that any initialization information loaded into the device during power-up is maintained.

The absolute maximum voltage of Max-i is primary set by the automotive load dump transient and the clamping characteristic of MOV's, but the de-facto maximum voltage of 40 V for many semiconductor products is also very important.

The load dump transient occurs when a car battery is suddenly disconnected while being charged by the alternator. During such a load-dump condition, the voltage on the alternator terminals increases rapidly (1 – 10 ms) to the open circuit voltage. This voltage is ***much*** *higher than the battery voltage because a high voltage is necessary to drive a high AC charge current through the inductance of the alternator windings. The length of the pulse can be as long as several hundred milliseconds. The load-dump standard defines the maximum voltage as 87 V for 12-V systems and 174 V for 24-V systems and it will therefore be 145 V for a 20-V system, but in practice, it is completely unpredictable. Even voltage levels over 100 V can occur in a 12-V system. The series resistance of the alternator is only a fraction of an ohm, so the energy can reach as high as 50 J.*

High voltage and energy levels are not compatible with modern semiconductor technology for three reasons:

- *As the chip technology is scaled down to improve the speed and reduce the power and price, the maximum voltage is reduced correspondingly (the field strength [V/m] internal in the chip is almost constant).*
- *The speed of semiconductors is inversed proportional to the maximum voltage.*
- *The necessary chip area and therefore the price for a given current increases rapidly at higher voltages. The higher the operating voltage, the wider the depletion layer needs to be. Thus the diffusions not only need to be deeper, but also more widely spaced. If the necessary normalized area at 5 V is 1, it will be approximately 1.7 at 20 V, 5.7 at 40 V and 52 at 100 V for a typical process. For this reason, a maximum voltage of 36 – 40 V has become a de-facto standard for IC design, and some high-power buck converters in the 48 – 72 V range like LTC7821 use a switch-capacitor pre-stage to reduce the voltage to the half before the buck part.*

Heavy automotive loads like lamps and motors may easily have an inrush current 5 – 10 times the DC current. For example, a single 12-V, 55-W, filament headlamp may require a current of 70 A! If it shall be possible to switch such loads by means of reasonable priced semiconductor technology, the maximum voltage must be reduced to 40 V or lower. Therefore, most modern alternators are equipped with a special centralized clamping circuit, which clamps the voltage below 40 V, and 22 – 24 V is the maximum voltage where this is still possible by means of TVS diodes. For example, a 24-V, 15-kW, heavy-duty diode such as 15KPA24A has a minimum breakdown voltage of 26.8 V and a maximum clamp voltage of 40.7 V at 371 A. However, it is also very easy and cheap to prevent the load dump pulse ***entirely*** *just by using a battery with dual poles where one set is connected to the alternator and the other set to the car (output plus chassis). The saved chip area for the electronics can easily pay for the extra two poles on the battery and the extra connection to the alternator (connected to battery instead of ground) so there is no reason to design automotive electronics – including Max-i – for 87, 145 or even 174 V operation. Dual pole batteries also have the advantage that it is not necessary to supply two types with inversed pole configuration for different cars.*

During garage jump-start, it is not unusual to use a 24-V charger on a 12-V car. The absolute maximum voltage set by the load dump also covers this voltage.

All power supplies shall:

- Be galvanic insulated from any high-voltage (230 Vac) net by means of a transformer.
- Be able to run continuously with no load without any noticeable acoustical noise (no cyclic start-stop).
- Have an efficiency of at least 90 % from 5 % to 100 % load.

  This is easiest to fulfill with output directly from battery poles and more chargers, which are enabled and disabled according to the actual load.
- Be protected against overload **and shut off until a manual reset** if the situation at a given level of overload lasts longer than the time specified and recommended previously for the cable and outlet protection. If it can be guaranteed that the power supply is able to supply enough current for a sufficient amount of time to activate the cable protection and therefore has “Selective Fuse Breaking” (SFB) capability, additional current limiting is not necessary.
- Limit the line voltage to the absolute maximum voltage even under transient conditions. In case of battery operation, this is usually fulfilled without any precautions and transient suppressors, but if a charger or solar panels can supply more than the absolute maximum voltage, the load dump situation where the battery is disconnected shall be considered.
- It shall include a Max-i controller, which can provide a clamp and additional line hold as described later.

A power supply useable for parallel operation shall also:

- Go into constant current mode if the maximum output current is exceeded so that the current may be shared with other power supplies. To avoid overheating in this mode, the power supply shall be a switch-mode or be thermally protected.
- Be able to accept a voltage on the output even without any input voltage (no mains voltage, solar panel, wind turbine etc.).

A power supply for a **balanced** line shall also:

- Be galvanic separated, and the separation shall be able to withstand at least 2.5 kVdc.
- Have a grounded capacitive shield between the primary and secondary winding. The coupling capacitance between the shield (ground) and the secondary side (L- and L+) shall not exceed 5 nF and shall be compensated with an equally big capacitor between shield and COM, which is able to withstand the same voltage as the galvanic separation.

  *Ordinary power supplies for electronic process control may have a coupling capacitance in the order of 50 nF and may not have a capacitive shield between the primary and secondary side. Such power supplies may be used for unbalanced or semi-balanced applications, but are **not** recommended for balanced lines.*
- To protect a floating, balanced line against common mode transients as lightning transients, L- shall be grounded through a low-capacitance spark gap (gas discharge tube). The spark gap shall be able to handle the maximum failure voltage, which is quite common 600 Vac in an industrial environment so that a 1-kVdc type must be used. The impulse spark over voltage of such a tube is usually less than 2.5 kVpeak. Any insulation and galvanic separation shall therefore be able to withstand at least 2.5 kVdc. This corresponds to the test voltage of most optocouplers.
- To avoid electrostatic build-up and to be able to detect a short circuit, which can be dangerous and destroy the line balance, all four conductors shall be connected to ground through a resistor in the order of 1 MΩ (250 kΩ for the four resistors in parallel). The network shall be able to handle the maximum failure voltage, which is quite common 600 Vac in an industrial environment.

  The total current through the four resistors should be supervised so that if the line voltage exceeds for example 10 Vdc or 100-200 Vac or draws current during the communication pulses (sign of unbalance), an alarm is activated.

1.5.1 Battery Operation

Due to a very big voltage range of ±23 %, Max-i can be driven directly from the poles of a 20-V battery with charger, which is the preferred method for example for smart-house, IoT applications and traffic lights.

This has many important benefits:

- *It is possible to store energy from for example solar panels and wind turbines and utilize it for example in the evening. It is also possible to charge the battery when the electricity is cheapest, which may happen between midnight and early morning where there is very little power consumption from industry, homes end electrical trains.*
- *Operation is maintained in case of a mains voltage failure (UPS functionality).*
- *It is not necessary with a power supply/converter, which can handle the full power, which may be up to more kW. It is only necessary with a charger, which can handle the average power consumption, but it is an advantage if it is powerful enough to charge the battery while the electricity is cheapest.*
- *The very low internal resistance of a battery (<5 mΩ for a 20-V, 6-kWh lead-crystal battery) combined with a very high capacitance in the order of 1 F per 100 Ah for a lead-acid battery makes it an excellent transient protection – especially if a dual-pole battery is used.*
- *The battery is able to supply current pulses big enough to burn fuses or trigger circuit breakers. In this way, a shorted part can be disconnected without interrupting other parts (selective fuse breaking – SFB).*
- *The efficiency is very high at low-load conditions, which is very important for smart-house and IoT applications. If a constant voltage was required and it shall be possible to burn fuses (SFB), it would be necessary to have a power supply in maybe the 2 kW range running continuously even during night conditions where there is usually*

very little load. This would create very big stand-by losses and force the power supply into discontinues mode, which often generates a very irritating acoustical noise and wear down the electrolytic capacitors.

- *If the battery charger is driven from a three-phase mains voltage, there is almost no need for electrolytic capacitors, which have a fairly limited lifetime – not even in the input, which may be 6-pulse rectifier with power factor correction or a 12-pulse rectifier on the secondary side of a three-phase transformer (star plus triangle). The output from a switch mode buck converter may use the battery as output capacitor. In this case, a dual pole battery or more batteries in parallel is recommended for safety reasons.*
- *It is possible to use more chargers and only activate as many as necessary. In this way, the efficiency can be optimized for all load conditions without sacrificing the possibility to source very high load currents immediately.*
- *A low voltage can be used to indicate that it is necessary to reduce the power consumption before the battery runs flat (brownout) and/or lower the current drawn by temporary connected devices.*
- *It reduces the number of series connected power converters and in this way increases the efficiency and reliability.*
- *It makes it possible to connect solar panels directly (through a switch) without any converters.*

The disadvantage is the big voltage range that connected devices must be able to handle, but many devices already contain a switch-mode converter, which converts the supply voltage to the internal voltage levels, which are usually in the 0.9 – 5 V range, and in the future, mechanical contactors and traditional DC-motors etc., which require a fairly precise voltage, are replaced by electronic soft starters, variable frequency converters and brush-less electronic commutated and controlled motors (ECM), which have a much higher efficiency and can easily accept the voltage range. In case of low-cost LED lighting, which only use simple, linear current generators, it is fairly easy to by-pass one or more LED's at low voltage levels as shown in Annex C and in this way keep the efficiency high without the use of switch-mode converters.

For automotive applications where the battery is constantly charged, it may be advantageous to use an 18-V battery instead, which has a saturated charging voltage of 20.7 – 21 V, as this will bring the voltage within the voltage range of USB Power Delivery and Quick Charge when the motor is running. When the motor is **not** running, the voltage will fall to approximately 18 V and since for example a connected laptop computer usually has an 18-V UVLO, the charging will stop.

To be able to further protect against short circuits and arcs, the **change** in current from the battery output shall be supervised for example by means of a current transformer, which is only sensitive to AC and other changes in current. If a sudden and unusual strong increase in current is detected, which occurs during a short circuit or an arc, all chargers including solar panels shall turn off within 1 second. This will lower the bus voltage to the open circuit voltage of the battery (20 – 21 V or lower), which will reduce the arc risk considerably compared to the charging voltage, which may be as high as 25.2 V. If the maximum current of all chargers together is less than 5 A, this requirement may be omitted as 5 A is too little to pull the voltage above 20 – 21 V and there is very little arc risk at 25.2 V if the current is limited to 5 A.

The turn-off shall be done fail-safe – for example by means of group messages, which is send approximately 2 times per second and causes all chargers and solar panels to cut off immediately when it is received and cut off within 10 seconds if the messages are not received (group watchdog timeout).

1.5.2 Use of Solar Panels

In case of a battery driven system, 54-cell solar panels may be connected directly without any other electronics than a (self-protected) switch for each panel, which is used to connect the panels one-by-one and in this way control the charging current/power in steps of typical 50 – 170 W. For a 54-cell panel or for example 3, 8.4-V (9.2 V at STC) thin-film panels in series, the voltage at maximum power is approximately 25.1 V measured according to the NMOT standard where the panel temperature is 45°C. This fits perfectly with the maximum charging voltage of 25.2 V, and since the open circuit voltage of the panels is below 36 V and the absolute maximum voltage of Max-i is 40 V, there is no load-dump problem.

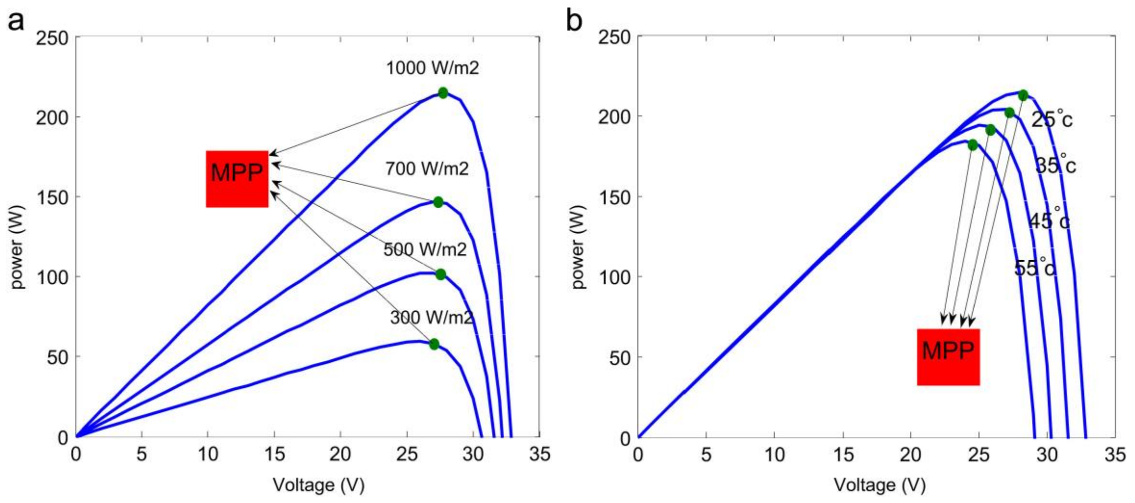


Fig. 1.16

Graph "a" shows the maximum power point (MPP) at a panel temperature of 25°C according to the STC standard, but such a low temperature is usually not realistic – especially at a solar power of 1 kW/m^2. Note however that the optimum load voltage is almost constant no matter the solar power except for a small drop at low levels, but if the lower (self)heating at these levels is taken into consideration, it is so constant that if the number of cells fits (54), even complicated switch-mode electronics cannot add much to the efficiency and will only lower the reliability, increase the price and may even create electrical noise.

All solar panel controllers shall be disconnected immediately in case of a short circuit or an arc. This may be implemented by means of fail-safe group messages as described in the previous chapter.

1.6 Creepage and Clearance

To avoid an arc risk at high current levels due to for example tin or silver whiskers, the creepage distance and air clearance between L+ and L- shall be at least 1 mm when the maximum power and current levels are limited to 100 VA and 8 A by means of a limited power source and 2 mm for higher currents.

1.7 Line Interface

As shown and described in annex A, communication is done by alternately driving the COM conductor(s) towards L+ and L- through a serial diode, which allows the voltage level on the line to become higher or lower than the transmitter levels when the reflections arrive. In case of a balanced line, each of the four conductors (two COM conductors) will pulse with the half pulse amplitude compared to ground. If for example the COM conductors are pulled from -6.5 V to +6.5 V – a voltage swing of 13 V – the voltage on L+ and L- will fall 6.5 V compared to ground and the voltage on the COM conductors will raise 6.5 V compared to ground. Therefore, it is very important that all devices connected to a balanced line are galvanic separated from ground and have a low coupling capacitance.

Max-i has two modes of operation depending on how the line is held in passive state:

1. Line held by means of a line hold circuit with the same voltage levels as the transmitter levels. **This is the usual operation mode**, which is special for Max-i and has a lot of important benefits:
    - It is the only method, which maintains good signal integrity between transmitting devices during bit-wise bus arbitration.
    - It is possible to use reflected wave switching in case of very long lines and/or free topology.
    - It has only two signal levels, which increases the transmitter power and signal/noise ratio and makes the signal much easier to detect in a reliable way.
    - By means of a 3-level hysteresis it is possible to use zero-crossing detection. This makes it possible to detect low voltage levels and do it without a big delay caused by the skin effect and/or reflected wave switching.
    - No power is lost in termination resistors.

2. Line held by means of ordinary termination resistors. This mode is only used in case of transformer coupling, if DC-free modulation is needed or if it is necessary with a higher communication speed than mode 1 allows for a given line length and it is acceptable that all devices, which may participate in bus arbitration, are close to each other.

Max-i uses BWM like a kind of Morse code where the dots corresponds to 0 and the dashes corresponds to 1. Each bit consists of a pause where the line is held followed by a standard pulse, which inverts the line state. The code is therefore self-clocking, which allows a fairly big clock tolerance and in this way enables the use of a rugged and reliable RC-oscillator.

The two transmitter levels shall be the same and except for the special railway applications as described in chapter 1.1.2, which allows voltage swings of over 22 V (±11 V), but requires more power supplies with ±1 % tolerance, the transmitter levels shall be ±6.5 V in relation to the midpoint between L+ and L-. To maintain the best possible signal integrity in case of mode 1 and voltage drops on the line, the levels shall be (almost) independent of the supply voltage and load current as long as the supply voltage is higher than 16 V and there is no current limitation. This allows a voltage drop up to 1.5 V in the transmitters including serial diode, but a lower voltage drop of maximum 1.2 V is recommended as this will allow the full voltage swing down to the minimum voltage (15.4 V) without limitations. The accuracy of the two levels shall be better than ±5 % over the entire temperature range with any resistive load in the range 33 – 100 Ω, and the numeric difference between the two levels shall not exceed 5 % either.

If the supply voltage is regulated, it is also possible to use a saturated transmitter and a clamp level, which is fixed according to the supply voltage as it is done in the special railway mode. This will however create some ringing in case of voltage drops, and just 10 % at 20 V degrades the signal integrity to approximately the same level as with a regulated transmitter as shown in Annex A, so it just uses more transmitter power without gaining anything except for a lower power loss in the transmitter due to the saturation, but simultaneously a much higher clamp loss.

The two communication modes are shown below:

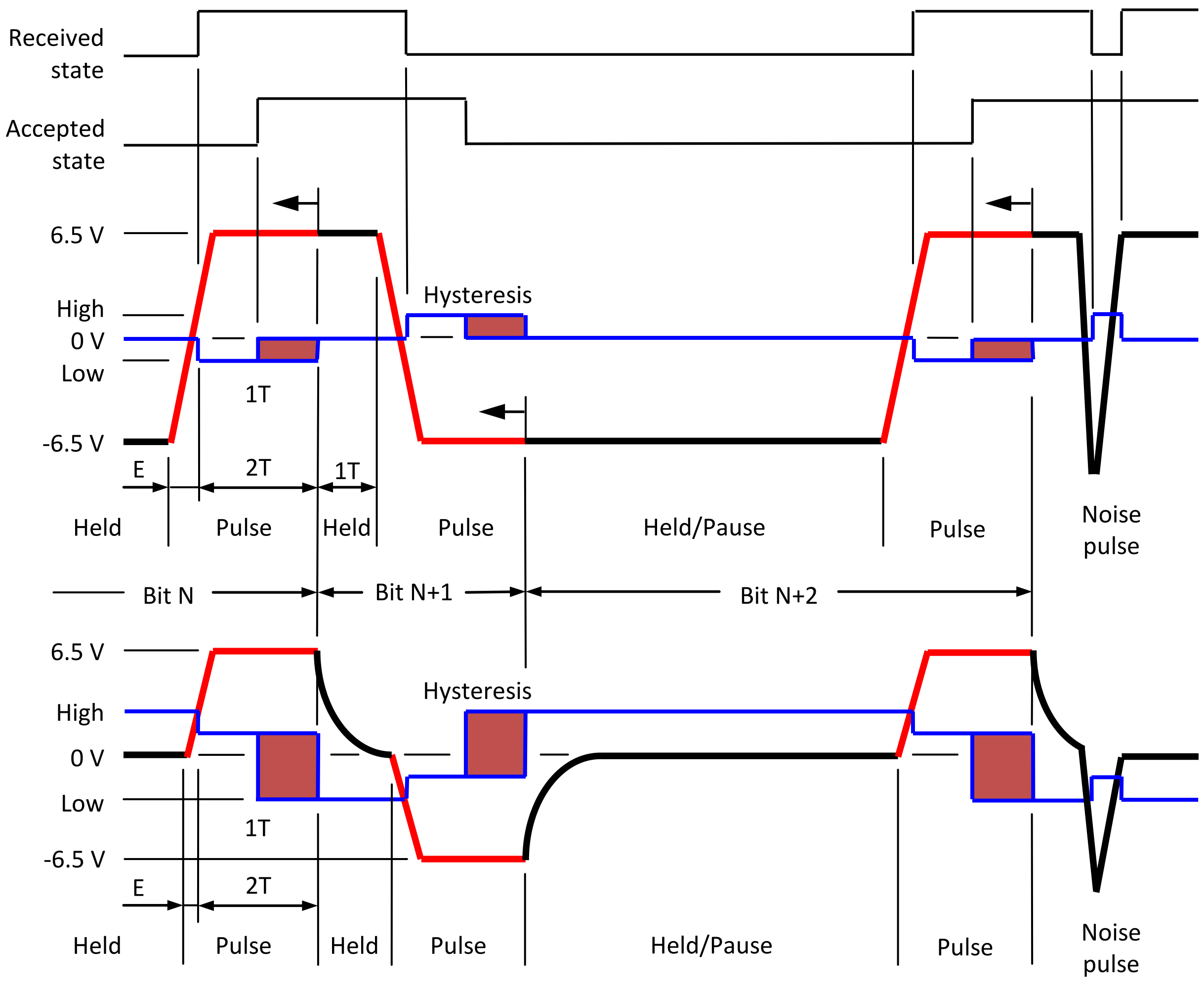


Fig. 1.17

The thick lines on the upper curve show a **simplified** waveform (without reflections) in mode 1 and the lower curve shows the corresponding waveform in mode 2.

In the red parts, the line is driven by a transmitter for a time of (up to) 2T after it has detected its own pulse. If the rise time (R) to the triggering level plus the detection delay is so long that the signal is not detected before the next sample, the total on-time of the pulse (P) is increased with one or more samples until this is the case as described later, so that P becomes approximately 2T + E. This may happen in case of high speeds and/or heavy capacitive loads. Note that in mode 1, the transmitter current is reduces to almost zero when the reflections from the cable ends arrive as shown with the 3 arrows so even though the transmitter **power** is very high, the transmitter **energy** may be very low (energy = power x time).

In the black parts, the line state is maintained by means of one or more hold circuits in mode 1 or by means of termination resistors in mode 2.

Even though the peak-peak voltage swing is the same for both hold methods, it is obvious that the voltage swing in the transmitter and with that the transmitter power is much higher in mode 1, and the difference between the signal and the detection limit, which is shown blue, is **much** smaller in mode 2, so the S/N ratio is reduced considerably in mode 2.

T is a standard time period, which depends on the communication speed. In mode 1, T must never be less than the propagation delay of the line, so the higher the speed, the shorter the line needs to be, but this is also necessary during the bit-wise bus arbitration so in practice it is not a limitation.

The accuracy of T shall be ±6% over the entire temperature range!

Because ±6 % may not be enough for example for UART communication and generation of time telegrams, it shall be possible to connect an external oscillator to a Max-i controller. An external oscillator may also be used to enable other UART speeds than the standard ones. If for example the very commonly used 115.2 kbit/s is needed, an internal 32 MHz oscillator may be replaced with an external 29.5 MHz oscillator. This will lower the standard speed 125 kbit/s to 115.2 kbit/s. Of course it will also lower the transmission speeds, but the decrease is only 8 %, so if all other devices anyway have an accurate clock with ±2 % accuracy to enable UART communication, the ±6 % tolerance between devices is still fulfilled.

The duty cycle of an external clock signal shall be 40 – 60 % and it should be verified that the external clock is not higher than approximately 80 % above the internal clock so that it is not running on harmonics.

*Note that unlike for example CAN, the used signal coding has **no** bias-distortion because the positive and negative going pulses, which charge the line, are always equal (no dominant and recessive bits). Due to the skin effect (see Annex A) and single point capacitive loads, a transmission line will always have a low-pass characteristic, which causes bias distortion in two situations:*

- *If the data coding contains DC like NRZ or 4-level codes (ISDN) and therefore is able to be charged by many 1's or 0's. Because of this, the simple NRZ coding, which is used by UART's and many other fieldbus systems, is really not suitable for long transmission lines without very advanced equalizers and/or preemphasers!*
- *If the drive impedance is different for 1's and 0's as in CAN as this creates a rectifier action.*

1.7.1 Receiver and Reception

To detect the signal in a reliable way, Max-i uses a combination of a simple comparator with a hysteresis of approximately 1 V plus a delayed offset circuit, which together with the hysteresis creates three threshold levels for voltage clamps and four threshold levels for resistor termination as shown with the blue lines on figure 1.17. The comparator hysteresis has the beneficial side effect that it increases the comparator gain to virtually infinite, which makes it possible to use a very simple circuit.

The delay of the comparator shall be below 70 nS for a 1 V overdrive.

In case of a voltage clamp, the line voltage must pass 0 V to change the line state, but as soon as this happens, the comparator level shall be changed to Low in case of rising edges and to High in case of falling edges.

If the new line state is accepted, which happens in the middle of the pulse at 1T, the following shall happen:

- The line state shall be latched (self-hold) and therefore be insensitive to further line changes (until 1T after the opposite polarity has been accepted).

- The receiver (and input filter) shall be dead until 2T where it shall open up for the opposite polarity (dead time shown with the brown field).
- The comparator level shall change back to 0-V somewhere between 1T and 2T to be ready for the next pulse.

There are more reasons why a 3-level hysteresis with 0 V as the nominal level is used instead of the much more common 2-level hysteresis:

- *Pulse detection becomes zero-crossing, which is the only level where the timing is not affected by the skin effect.*
- *The size of the hysteresis does not affect the initial threshold level, so:*
    - *more signal damping can be accepted and with that thinner cables.*
    - *a relatively big hysteresis can be used. This makes it easier to reject line oscillation/ringing during bus arbitration.*

In case of resistor termination, the offset circuit shall be changed in such a way that the upper threshold levels become approximately ±2.4 V, and the delayed low levels therefore approximately ±1.4 V (1 V hysteresis). This is necessary because the resistor terminated case has 0 V as resting voltage, and the threshold levels for the next pulse must therefore be in the order of the half pulse amplitude above and below that.

If the comparator draws any noticeable current when the voltage level on the line gets above or below the power supply rails, but below the clamp levels, this shall be subtracted from the hold current, so that the total input current is not affected.

As a supplement to the hysteresis, the receiver shall include a (digital) filter, which can reject oscillation pulses with low duty cycle and/or pulses narrower than 1/6 of the propagation delay (1/6 T) of the line as described in Annex A.

Max-i has been designed to be a sampled system (like a UART) **with 8 samples per T**.

The benefit of the sampling is that the circuit can be based on a master oscillator, which does not need to be synchronized. This simplifies the oscillator and makes it useable for other purposes like UART's and timers. It also makes it possible to use ring oscillators, which are difficult to synchronize. The disadvantage is that the sampling may delay the signal reception.

With 8 samples per T, it is very easy to implement the additional filter hysteresis.

In the following, R is the rise time from the pulse is initiated by the transmitter in a device until it passes the threshold level(s) of the receiver, which is always 0 V in case of clamp termination, but are approximately ±2.4 V in case of resistor termination.

The detection delay (D) is defined as the delay of the input comparator plus additional delay due to increased rise time as the signal propagates down the line for example caused by serial resistance, capacitive loads, skin effect and/or varying line impedance. The detection delay is usually quite small for clamp termination, but may be increased considerably if the skin effect knees get below the two threshold levels in case of resistor termination, which may happen in case of thin cables as shown in Annex A.

Note that when a signal is reflected from the end of a cable, the reflected wave superimposes the forward wave, so that the voltage is doubled, but may be limited by a clamp to a voltage level above or below the supply voltage rails. This in effect reduces the time it takes the signal to reach the detection level to the half so even if the amplitude is reduced and the rise time increased as the signal propagates down the line, the signal is to some extend reshaped by the reflection, which is a further advantage compared to termination resistors.

When a signal propagates down the line, it will be attenuated and distorted due to the skin effect, but as explained in Annex A, the signal will always pass 0 V at a time equal to the distance multiplied by the propagation delay of the line (nS/m) as long as the skin effect knees do not get below the threshold levels. Therefore, the detection delay due to the increased rise time is usually quite small for incident wave switching and clamp termination where the triggering level is always 0 V (3-level hysteresis), but it may be substantial in case of resistor termination, which is yet another reason not to use it unless it is strictly necessary.

Because of the possibility for increased detection delay in case of thin lines, Max-i has two operation modes – incident wave switching for high speed operation and reflected wave switching for applications where a thin cable and/or a more free topology than a trunk line is wanted and a reduced speed can be accepted. With incident wave switching, the receiver is triggered as soon as the signal arrives at the receiver. With reflected wave switching, the receiver may not be triggered before the reflection from the end of the line arrives or even not before many back and forth waves have been added. This creates some delay and jitter, which makes it necessary to reduce the speed. Note however that because

Max-i does not use any termination resistors, the line will eventually be charged to a voltage close to one of the two hold levels no matter how thin the cable is.

Fieldbus systems, which use termination resistors, cannot use reflected wave switching to increase the line length for a given cable cross section.

The reception sequence is described below and shown for mode 1 with clamp termination:

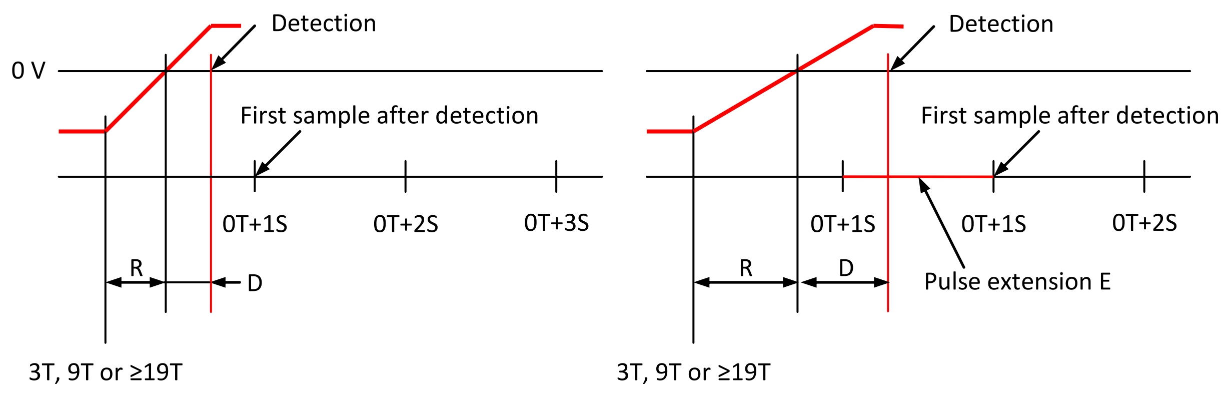


Fig. 1.18

1. A pulse is initiated by the device itself or by another device at 3T, 9T or ≥19T.
2. The signal ramps up and is detected (R + D) later plus any propagation delay between transmitter and receiver. (R + D) depends on the speed of the transceiver circuits, any slew rate limiting and any low pass characteristic on the line, which increases the rise time, but is independent of the communication speed and the clock accuracy.

   As soon as the signal is detected, the binary value corresponding to the time (0 before 7T and 1 after) shall be latched, and a new timer shall be released.
3. The new timer shall be counted forward to 1S at the first sample after the detection, which may occur immediately after. If R + D < S, which is usually the case, this keeps the timing accurate if the pulse is initiated by the device itself, since this is done at 0S. If the speed is however very high and/or there is heavy slew rate limiting, the detection may take place after the first sample, but even in this case, the new timer shall be set to 1S after the detection so that the pulse width is extended with one or more samples (E), as shown above. This ensures that the width of the pulse after it has reached maximum level in practice can be considered to be at least (2T - S)(1 - x) where x is the maximum clock inaccuracy. For S = T/8 and x = ±6 %, this is 1,76T.

   Since the propagation delay on the line from one end to another must never exceed ½ pulse width as explained and specified later, the absolute maximum propagation delay is 1,76T / 2 = 0.88T, but it is further reduced if there are not clamps in both ends.
4. If the line state has been steady above the threshold level(s) (not a noise pulse) when the new timer reaches 1T, that is 1T - S later, the new line state is accepted and two things shall happen:
   - The latched binary value shall be loaded into the receiver.
   - The new timer shall be used as the bit timer or the time shall be copied to the bit timer. This synchronizes the device.

   Note that a device is always synchronized by its receiver – not a direct connection from the transmitter!

   Also note that the synchronization may shorten a pulse. If for example a device has a 1-bit to transmit (at 9T) and a pulse is detected at 8.2T, the device cannot know whether the pulse is a noise pulse or a 1-bit from another device. Therefore, the device has no other choice than to starts its own transmitter at 9T (in case it is a noise pulse), but at 9.2T (1T - S after the detection) the new line state is accepted as a 1-bit from another device and when this happens, the time is set to 1T. The pulse width from this device is therefore reduced to 1.2T, but the pulse width on the line is still at least (2T - S)(1 - x). If a pulse from another device occurs more than 1T before the transmission time, the pulse will not be transmitted at all, which means that the minimum pulse width is 1T.

   When the pulse from another device is accepted at 1T, the bit timer is reset and after that a new pulse cannot be transmitted earlier than 3T. This is for example a very likely situation during the first bit of bus arbitration (start bit) in case of big clock inaccuracies, but since the bit (receiver) timing is independent of any transmission, all devices are just synchronized and may transmit the second bit. It is only necessary with a start-bit from one device.

The master/slave (latch) read system is very important for three reasons:

- *Because of the big clock inaccuracies and the sampling, which makes it appropriate to read the timer asynchronous as early as possible (not 1T later).*
- *Because the binary value for the bit may change from 1 to 0 when the counter is reset 1T after the edge. This occurs for all 1-bits. Without the latch system, there would be critical race between read and reset of the binary value.*
- *It makes it possible to compare the received bit with a reference bit as soon as the pulse is accepted at 1T. This is important for the ID acceptance filter.*

1.7.2 Transmitter

The transmitter shall be stable with any capacitive load and shall have a current limit of ±1.2 to ±1.6 A over the entire temperature range so that it is able to drive a line with a characteristic impedance Z_0 down to 35 Ω. The transmitter shall also be able to handle an **average** power dissipation of 0.75 W. To limit the power dissipation in case of a short circuit to V- or V+, it is recommended that the current limit depends on the voltage over the output stage. This is possible because the transmitter is intended to drive resistive loads (transmission line) where the current depends on the voltage.

As shown in chapter "A.10 Point-to-Point Communication and Reflected Wave Switching" in annex A, the maximum transmitter voltage swing is 3 × 6.5 V = 19.5 V, and with a minimum 17.5 Ω load (Z_0/2), the maximum current becomes 1.1 A.

With fixed transmitter, hold and clamp levels, the transmitter current only depends on the line impedance and is independent of the supply voltage, but the power loss in the transmitter and the clamp varies. At high voltage levels, the power loss in the transmitter is fairly high, but the loss in the clamp very low, and at low voltage levels, the loss in the transmitter is low, which may extend the battery life, but the loss in the clamp is higher.

If the transmitter is located in the end of the line, the initial load impedance corresponds to Z_0. With the maximum supply voltage of 25.2 V, V_{TX} = 13 V, V_{Clamp} = ±13 V and the minimum line impedance of 35 Ω, the TX current is 370 mA and the clamp current is 185 mA, so the power loss in the transmitter is ((25.2 V - 13 V) / 2) × 0.37 A = 2.26 W and the power loss in the clamp is (13 V - (25.2 V / 2)) × 0.185 A = 0.07 W. At the minimum supply voltage of 15.4 V, the power loss in the transmitter is reduced to only ((15.4V - 13 V) / 2) × 0.37 A = 0.44 W, but the power loss in the clamp is increased to (13 V - (15.4 V / 2)) × 0.185 A = 0.98 W. As shown in chapter A.10, the power dissipation in the transmitter is the same in case of a high loop resistance where the initial voltage swing is increased to 19.5 V and the ***maximum*** *current therefore to 557 mA since the* ***average*** *current is the same.*

If the transmitter is located in any other spot, the initial current will be doubled, as the transmitter looks into two lines in parallel. It must therefore be able to handle at least 1.1 A, but the current will be zero when the reflection from both ends have arrived. The average power loss will therefore be the same. A device in the middle of a line will just draw the double amount of current for half the time.

If more devices drive the line simultaneously, the current in a clamp between these may be doubled, so a clamp must be able to handle at least 370 mA, but as it is the case for the transmitter, the time is reduced correspondingly so the power loss is not increased either.

In all cases, the average power losses are less than ⅓ due to the bit coding and the scrambling, so it is only necessary with an IC package, which can handle 0.75 W over the entire temperature range.

The rise time measured from the transmitter is activated until the signal has passed the 90 % level shall be 70 ns – 150 ns, which allows spur cables of typical 4 m (120 ns) and may be used up to the maximum speed. Advanced transmitter may also be able to reduce the rise time to for example 600 ns, which allows spur cables up to 20 m and may be used at speeds up to 167 kBaud corresponding to a UART speed of 250 kbit/s. In case of more possibilities, the rise time should automatically be selected according to the selected speed – for example 600 ns at ≤167 kBaud, but may also be programmable.

During transmission of a high bit, the transmitter shall go off if the signal is not in high state the so-called self-hold time after the pulse is initiated. In the same way, the transmitter shall go off during transmission of a low bit if the signal is not in low state the self-hold time after the pulse is initiated. These situations are regarded as a short circuit. The self-hold time must of course always be longer than the maximum time it takes from the transmitter is activated until the signal has been detected by the device itself, but must not be so long that too much power is dissipated in the transmitter in case of a short circuit. The maximum detection time is typical ½ of the maximum rise time (zero crossing) plus the maximum receiver delay, that is, approximately 150 ns.

The self-hold time shall be ½ pulse width (1T) for all speeds except for the two maximum speeds. At maximum speed, 1T is only 125 ns, and with a maximum receiver delay of 150 ns and some transmitter delay including rise time, there is no time to establish self-hold. To ensure a reliable operation, but still a reasonable maximum 0-bit length, the self-hold time shall therefore be increased to 2T (250 ns) for the two highest speeds.

The response to a short circuit depends on the event that has caused it. In case of an **externally** triggered transmission like a UART transmission or a reply to a poll, the event shall be reset and blocked for the next 0.75 – 1 second. If the event is **internally** generated such as the transmission of an event driven implicit message, an error message including an error message for the short circuit itself, a serial number transmission or a pulse to drive the line low if it is high between telegrams, the event shall just be temporary suspended for the 0.75 – 1 second so that a steady short circuit condition causes the transmitter to retry at a rate of approximately 1 Hz.

The reason why the self-hold time is not always 2T is to limit the power dissipation at low speeds in case of a short circuit, which may dissipate up to 40 W (25.2 V x 1.6 A), and the reason why it is always at least 1T is to allow the line to charge in case of reflected wave switching, which may take up to 0.4T plus receiver delay. At the lowest speed, where T is 512 µs and the dissipated energy therefore up to 20.6 mJ, the suspend timer ensures that the average power dissipation is only 20 mW and at higher speeds it is reduced even further.

To avoid a big voltage drop on the supply lines during transmission, the transmitter shall be decoupled with at least a 2 µF decoupling capacitance between L+ and L-. With an average of 50 µF/km (25 devices per km), the characteristic impedance between L+ and L- is then reduced from approximately 150 Ω to approximately 3 Ω as explained in Annex A and since only the half transmitter current goes through the capacitors, the worst-case voltage drop (at 750 mA) with 2/3 on-time of the transmitter is reduced to approximately 0.8 V, which is acceptable.

The Fourier spectrum of a slew rate limited pulse is shown below:

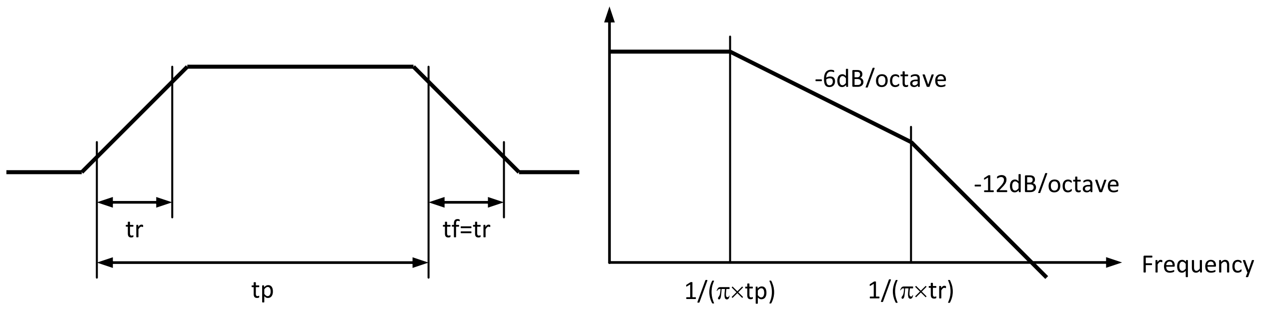


Fig. 1.19

tp is the symbol length, which may be as short as 3T (0-bit). The -6dB knee (0.106/T) is fixed by the transmission speed. The only way to limit the EME is therefore to control the slew rate. Above the -12dB cut off frequency, EME is reduced very fast so in practice the slew rate controls the maximum noise frequency. Note that slew rate limiting is not so important on low speeds because the symbol width is increased so the -6dB frequency falls. Therefore, it is not necessary to use a speed depending slew rate. This simplifies the transmitter considerably. Devices intended for applications where the maximum speed is not needed and/or the cable is unbalanced – for example intelligent house and automotive applications – may however benefit from a rise time in the order of 600 nS so that spur cables up to 20 m may be used, which gives virtually free topology and less high frequency EME.

It is not necessary to limit the slew rate of the falling edge as the transmitter anyway goes off when the reflection arrives.

Because of the pause between the pulses where no current is put into the line, a row of 0 or 1 bits becomes a quasi-square-wave with less harmonic content than a pure square-wave. The frequency spectrum of such a wave does not contain even harmonics (2nd, 4th etc.) and the magnitude of the odd harmonics may be calculated as:

$$V_{rms} = V_{TXPP} \times (2 \cos(n\,\alpha / 2)) / (n\,\pi\,\sqrt{2})$$

where n is the number of the harmonic and α is the width of the pause, which is 1T = π / 3 for 0-bits and 7T = 7 π / 9 for 1-bits in the worst case situation where the transmitter draws current for 2T. Note that the fundamental frequency of a row *of 1-bits is 1/3 the fundamental frequency of a row of 0-bits. If the line is shorter than maximum, the pulse width is reduced as the transmitter current goes to zero when the reflections arrive. This may reduce the magnitude of the harmonics considerably in for example automotive and Ex applications.*

Because $\cos(3^N \pi / 6) = 0$, a row of 0-bits with a full 2T pulse do not contain 3N harmonics like 3rd and 9th order. The other harmonics become a worst-case magnitude of:

$$V_{rms} = 0.39\ V_{TXPP} / n = 5.1 / n \text{ for } V_{TXPP} = 13\ V\text{, which is the nominal value.}$$

The fundamental frequency therefore becomes a magnitude of 5.1 V_{rms} and the 5th order harmonics a magnitude of 1 V_{rms}.

In case of 1-bits, the worst-case magnitude of the fundamental frequency and the first two harmonics becomes:

$$V_{rms}1 = 0.154\ V_{TXPP} = 2.0\ V_{rms}$$

$$V_{rms}3 = 0.130\ V_{TXPP} = 1.7\ V_{rms}$$

$$V_{rms}5 = 0.089\ V_{TXPP} = 1.2\ V_{rms}$$

If a balanced line is connected to some equipment, this equipment is exposed to a common mode voltage one half of these values (2.55 V_{rms} for the fundamental frequency at 0-bits). However, if the equipment is able to fulfill the requirement for the European CE marking, that is, is able to fulfill the requirements in EN 50082, it shall be able to withstand a common mode voltage of 3 V_{rms} from 150 kHz to 80 MHz with an 80 %, 1 kHz AM modulation so although it is difficult to compare the two waveforms there is a good chance that the circuit will work.

1.7.3 Line Transformers

If line transformers are used, they shall be carefully designed according to the wanted communication speed, the characteristic impedance of the line and the expected number of devices.

To avoid line ringing when a driver goes off, the transformers shall at least be critical damped ($\zeta \geq 1$) by the impedance of the line, which is equal to the characteristic impedance (Z_0) divided by two since each device looks into two lines (or termination resistors) in parallel. To avoid inductive loading and to filter the signal, each transformer shall be set in resonance corresponding to the synthetic sine wave, which is generated by a row of 0-bits. If for example the baud rate is 166.7 kbaud (UART speed 250 kbit/s), the sine wave becomes 83.33 kHz. Since the resonance frequency of an LC-circuit is $f = 1/(2\pi\sqrt{LC})$, the LC product may be calculated as:

$$LC = 1/(\pi \times \text{BaudRate})^2$$

For a parallel circuit, the damping factor is $\zeta = 1/(2R) \times \sqrt{(L/C)}$, and since $R = Z_0/2$ and $\zeta \geq 1$,

$$L/C \geq Z_0^2$$

When the two formulas are combined, the inductance for all transformers in parallel may be calculated as:

$$L \geq Z_0/(\pi \times \text{BaudRate})$$

If for example the baud rate is 166.7 kbaud and $Z_0 = 50\ \Omega$, the total inductance shall be at least 95.5 µH so in case of for example up to 10 devices on the line, each coupling transformer shall have a self-inductance of at least 955 µH, and if 1 mH is used, the parallel capacitance over each transformer including capacitance in the windings shall then be 3.6 nF.

⚠ **To reduce the noise sensitivity, some part of a line transformer shall always be grounded on the primary side (receiver side) – either directly (difficult with Max-i) or through a capacitor and a (resistor) circuit to hold the voltage at that point reasonable constant.**

1.7.4 Hold Circuit

In case of voltage clamps (mode 1), the COM conductors cannot float forever without a risk of changing the line state, and in case of low-loss lines, it is desirable to damp the reflections. Therefore, each device shall include a line hold circuit, which can draw the COM conductor(s) towards the high transmitter level if the line is in high state and towards the low transmitter level if the line is in low state as explained in Annex A – Transmission Lines.

To avoid self-hold on noise pulses, the state of the hold circuit shall not change before the new line state is accepted, which occurs in the middle of the period (at 1T) where the transmitter is active (in the middle of the transmitter pulse).

This is also true during transmission so that if the line is short circuited to some voltage level and therefore cannot change, the line state does not change either.

The hold levels shall be the same as the transmitter levels and therefore have the same tolerances, and it shall be possible to select between two different current levels:

1. Approximately ±0.2 mA to prevent that leakage currents in the device change the line state. This level shall always be active and shall be the default level.
2. Level 1 plus approximately ±5 mA as shown below to damp the reflections, compensate for small differences in clamp level and further hold the line to prevent that the line state is changed by noise pulses, mutual coupling from other cables and leakage currents in cables and junction boxes:

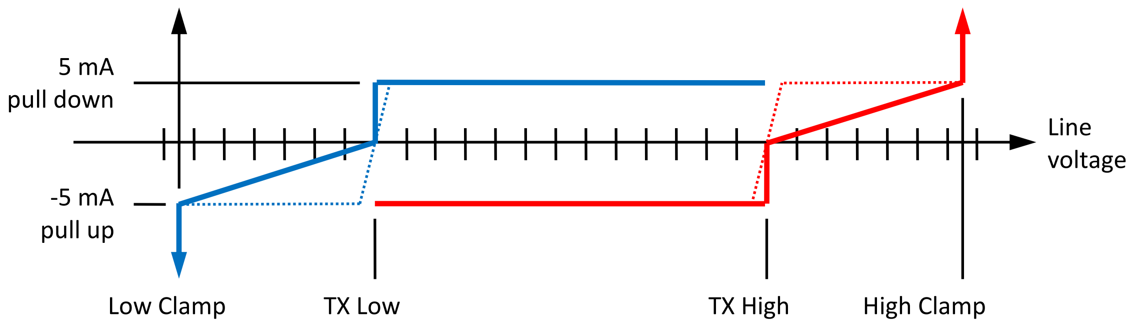


Fig. 1.20

The hold circuit shall try to keep the line voltage above the positive transmitter level (TX High) or below the negative transmitter level (TX Low), but not too much. The thick blue and red lines show the preferred characteristic, but a simplified characteristic as shown with the thin dotted lines is allowed.

The extra hold current shall by default be enabled in all channels on power supplies, and during commissioning it shall be ensured that it is enabled in 2 – 10 devices on the bus including devices in power supplies.

To prevent that the extra current in level 2 draws power out of transmitted pulses and in this way increases the reflections, which will cause the line to be charged to a lower level than optimal, the additional current (not the basic ±0.2 mA) shall be switched off any time there is a difference between the detected line state (after the receiver filter) and the accepted line state, which changes at 1T.

As shown, the hold circuit shall be able to source and sink the specified current levels if the COM voltage is both above and below the present hold level.

The default line state shall be low. This makes it possible to use a boot-strapped, N-channel MOSFET for the high side transmitter.

Devices where level 2 is **enabled** shall ensure a low line state by transmitting a single start bit (described later) if the line state is high when the line gets idle after a telegram. If there are other devices, which transmits telegrams at the same time, the start bit from the new telegram will overlap the extra start bit, which will therefore become invisible, so that the line state will not be changed before there are no telegrams to transmit. The extra start bit will therefore not reduce the efficiency.

During **power-up** and while the PUR pulse is active, the hold circuit shall **try** to pull the line low and therefore set the receiver threshold level to the high level.

1.7.5 Clamp Termination in Mode 1

As explained in Annex A, the clamp levels shall be ±13 V when the device is powered up. As with the transmitter and hold voltages, the accuracy shall be better than ±5% over the entire temperature range and the numeric difference between the two levels shall not exceed 5%. If the device is not powered up, the clamp shall be in high impedance state. In case of fritting – see next chapter, it may be necessary to add an additional negative clamp to make it work and prevent a too big negative voltage rise if the line is not powered up.

The clamp circuit shall be able to handle at least 370 mA and shall be able to handle voltage pulses up to the absolute maximum voltage without a big increase in current.

The clamp circuit shall act within 50 ns and have low capacitance, and below the clamp levels, it shall not add any additional line load current beyond the hold current.

To avoid negative going pulses due to high reverse recovery currents in the necessary blocking diodes when the clamp goes off, the diodes must be **extremely** fast. Junction diodes may be used, but since extremely fast diodes are usually made by means of gold doping or similar, which may not be available in an IC process and also increases the voltage drops, low-leakage schottky diodes, which do not have any reverse recovery current except for the one caused by the capacitance, may in practice be the only useable!

⚠ **As explained in Annex A, the line shall in mode 1 be terminated in at least one end by means of an active clamp network, that is, a network, which is powered up, and the absolute maximum length of the cable is determined by the time it takes a signal from the terminated end to be reflected from the other end and travel to the nearest clamp. This time must not exceed 1T!** If it does, negative pulses and even transmitter overload and reversed line state may occur as described in Annex A. It may therefore be necessary to shorten the line if there is no clamp in the far end of the cable as shown below!

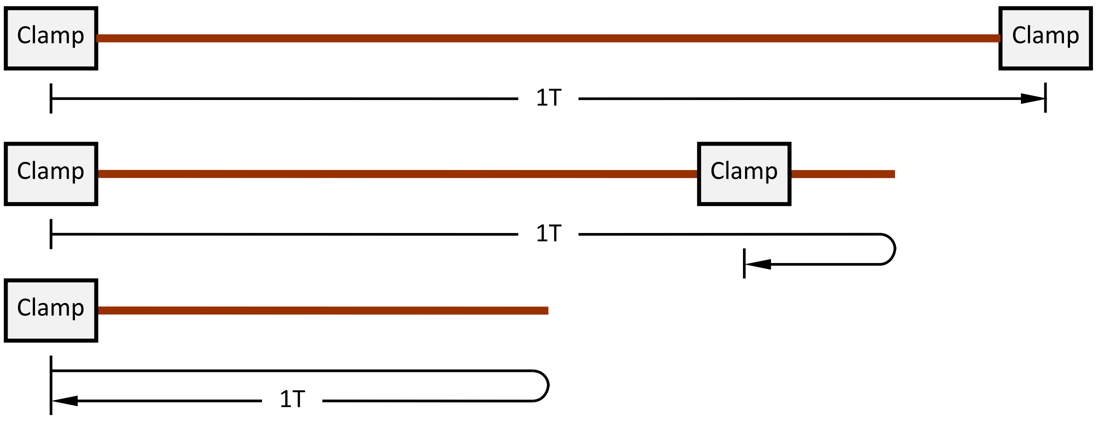


Fig. 1.21

Note that a clamp does not affect the signal unless two or more waves superimpose so in the middle situation, the clamp does not affect the signal until the reflection from the unterminated line arrives and double the voltage and therefore pass the clamp level, and if there is only a termination network in one end, the maximum length is reduced to the half.

Also note that a clamp must be powered up to work! This must be taken into consideration as it may change the topology if a fuse is blown or a device is switched off. If for example a centralized power supply with a fuse to each side of the cable is used, it may be necessary with a relay to disconnect the COM conductors to the unpowered side in case of a blown fuse, and a device, which may be switched off, shall not be regarded as a permanent clamp.

Fieldbus systems, which use termination resistors including Max-i in mode 2, have a similar problem. If the connection to just one of the resistor is lost, the communication may fail, and if the connection to both resistors is lost, it will almost always fail, so by means of a sufficient number of clamps, Max-i can in mode 1 be much more failure tolerant.

1.8 Line Length

The maximum length for incident, point-to-point and reflected wave switching on a 45-Ω line, **which is terminated with a clamp in each end**, is shown below. It is based on the simulations shown and described in chapter “A.10 Point-to-Point Communication and Reflected Wave Switching” in annex A:

Speed code	T	Max-i Speed	UART			Wave switching	Max. delay (0.88T)	Max. PE length (5 ns/m)	Maximum resistivity	
	µs	kBaud	Speed kbit/s	Clock divider	FIFO Size		µs	km	Ω	Ω/km
1111 1110	512	0.651	1.2	$13\cdot2^8$	1	Incident	450	90	20	0.18
						Reflected	450/N	90/N	60×N	$0.67N^2$
1101	256	1.302	2.4	$13\cdot2^7$	1	Incident	225	45	20	0.44
						Reflected	225/N	45/N	60×N	$1.33N^2$
1100	128	2.604	4.8	$13\cdot2^6$	1	Incident	112	22.5	20	0.78
						Reflected	112/N	22.5/N	60×N	$2.67N^2$
1011	64	5.208	9.6	$13\cdot2^5$	1	Incident	56	11	20	1.8
						Point-to-point	56	11	80	7.3
						Reflected	56/N	11/N	60×N	$5.4N^2$
1010	32	10.417	19.2	$13\cdot2^4$	1	Incident	28	5.6	20	3.1
						Point-to-point			80	12.5
						Reflected	28/N	5.6/N	60×N	$10.7N^2$
1001	16	20.83	38.4	$13\cdot2^3$	2	Incident	14	2.8	20	6.25
						Point-to-point			80	25
						Reflected	14/N	2.8/N	60×N	$21.4N^2$
1000	8	41.67	76.8	$13\cdot2^2$	4	Incident	7	1.4	20	12.5
						Point-to-point			80	50
						Reflected	7/N	1.4/N	60×N	$43N^2$
0111	4	83.33	125	2^5	8	Incident	3.5	0.7	20	25
						Point-to-point			80	100
						Reflected	3.5/N	0.7/N	60×N	$86N^2$
0110	2	166.7	250	2^4	16	Incident	1.75	0.35	20	50
						Point-to-point			80	200
						Reflected	1.75/N	0.35/N	60×N	$171N^2$
0101	1	333	500	2^3	32	Incident	0.88	0.18	20	100
0100	0.5	667	1000	2^2	64	Incident	0.44	0.088	20	200
0011	0.25	1333	2000	2^1	128	Incident	0.22	0.044	20	400
0010	0.125	2667	4000	2^0	256	Incident	0.11	0.022	20	800

Fig. 1.22

N is the number of back and forth waves for reflected wave switching and must therefore be an integer.

If a cable with lower characteristic impedance than 45 Ω is used, the allowed loop-resistance is correspondingly lower. Note that a capacitive load will lower the effective impedance as described previously in chapter "1.2.3 Resistive and Capacitive Load".

The resistance per km for various cross sections of copper cable is approximately:

Cross section	Ω/km
6.0	3.2
4.0	4.9
2.5	7.8
1.5	13
0.75	26
0.5	39
0.38	51
0.19	100

Fig. 1.23

At for example a UART speed of 1.2 kbit/s, a 4×4 mm^2 cable can be used down to N = √(4.9/0.67) = 2.7 → 3 times wave switching (round up to integer), so that the maximum cable length is 90/3 = 30 km.

The minimum size 0.19 mm^2 corresponds to a solid conductor diameter of 0.5 mm, which is a de facto standard for many communication cables. Thinner cables are not recommended. In this way, the reduced skin effect due to thicker cables than necessary at high speeds is used to partly compensate for the increased significance of the detection delay.

The specified cable cross-sections implies that a balanced 4-wire cable with 4 equal conductors and a characteristic impedance of at least 45 Ω is used and that the decoupling capacitance of the line is at least 25 µF/km (see Annex A). **If a cable with lower characteristic impedance is needed, the maximum DC-resistance must be reduced correspondingly and/or the cable length must be reduced and if a higher impedance is used for example in case of a 2-wire line, which typical has an impedance of 80 – 100 Ω, it is possible to increase the length for the same cable cross section because fewer reflections are needed to build-up the signal! T**he values only apply to the transmission – **not** any DC load. For a balanced 4-wire line with incident wave switching, the maximum DC resistance of **one** conductor is 20 Ω corresponding to 1.6 km of 1.5 mm^2 cable. Because all 4 conductors carry the communication, the maximum DC loop resistance for communication is also 20 Ω as there are two conductors in parallel in series with two conductors in parallel. As can be seen, incident wave switching is unrealistic (marked with red background) above 6.4 km due to the necessary cross section, but the cross section may be reduced a factor 8 if the half line length is acceptable for a given speed (2× Reflected). In this case, the maximum DC resistance is 80 Ω, so the cross section may be 8 times smaller. If Max-i is used for point-to-point communication between two groups of devices fairly close to each other, it is possible to use an 80 Ω cable without a reduction in speed and/or cable length!

A very simple rule of thumb is that the cross section in mm^2 shall at least correspond to the length in km for incident wave switching and one fourth for point-to-point communication and 2× reflected wave switching. If for example a 1.5 km line is used, the cross section must be 1.5 mm^2 for incident wave switching and 0.38 mm^2 for point-to-point communication and 2× reflected wave switching.

The UART field shows the communication speed in case of the Modem object.

The bus speeds and the UART speeds are locked to each other with the relationship shown in the table. Max-i uses the standard UART speeds 2^N × 1.2 kbit/s up to 76.8 kbit/s, which is the same as used by for example BACnet and P-net (only 76.8 kbit/s), but for higher speeds, it uses 2^N × 125 kbit/s, which is easier to generate without a fractional bit rate divider and is also used in many other fieldbus systems like DMX512 and CAN (DeviceNet, CANOpen, SDS etc.). Both rows may easily be generated from a 32 MHz clock.

The reason for having two codes for 1.2 kbit/s is to make it possible to use this speed even if the speed setting in password 3 is different. In this case, 1110B shall be specified for 1.2 kbit/s as 1111B indicates that the default speed in password 3 shall be used.

⚠ **Note that the UART speed does NOT represent the actual throughput!** Even though the scrambling (specified in layer 2) ensures approximately the same number of 0-bits and 1-bits in a long telegram, the UART speed needs to be fast enough to handle a row of 0-bits, which for example occur in time messages – and even if the internal oscillator runs at maximum speed. Since a long 1-bit is 3 times longer than a 0-bit so that the average bit time is twice the length of a 0-bit for the majority of messages and some time is needed for start and stop bits plus a break character to separate telegrams, the UART speed must be **much** higher than the actual throughput, which is therefore only 27 % of the UART speed for 1.2 – 76.8 kbit/s and 33 % for 125 – 4000 kbit/s! If for example a 750 m, 4×0.75 mm^2 line is used, it is only possible with an average throughput of approximately 40 kbit/s for incident wave switching even though the UART speed is 125 kbit/s!

This also reduces the necessary FIFO size for a given UART speed. The table shows the minimum FIFO for Windows operation (max. 1 ms interrupt latency or USB schedule). At for example 2 Mbit/s, each byte in average takes 12 µS to transmit (4, 0-bits and 4, 1-bits). This means that approximately 80 bytes may be received in 1 ms so that a 128 entry FIFO is enough and with that for example a 16C950 UART. At the recommended standard speed for smart-house applications 250 kbit/s, a 16-entry FIFO is enough, and this length is very common.

The default UART speed of an un-programmed Max-i controller (code 1111B) shall be 9.6 kbit/s, but as described later, it is possible for the product manufacturer to program another out-of-the-box speed in password 3. If for example the device is intended for smart-house applications, the default speed may be set to 250 kbit/s (code 0110B) to enable out-of-the-box compatibility with other devices.

At all speeds, Max-i uses 8 samples per T and a UART with 16 samples per bit. To generate the lower UART speeds, an additional divide-by-13 prescaler is used.

The speed of Max-i and the divider ratios of T and the UART have been selected in such a way that:

- *It is possible to use an integer, standard clock frequency, which is also easy to use for timers etc.*
- *The UART speeds are as low as possible.*

Except for the old teletype speed 110 bit/s (with two stop bits), there are two standard rows of UART speeds (with one stop bit) – 2^N × 300 bit/s and 2^N × 450 bit/s (1.5 times higher). Unfortunately, the PC world uses the first row up to 38.4 kbit/s, but then changes to the second row, that is, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600 etc. This causes a very un-logical discontinuity with more disadvantages:

- *It makes it difficult to generate all speeds from the same clock without some inaccuracies or a 1.5 times higher clock frequency, which may cause problems (48 MHz clock instead of "only" 32 MHz).*
- *It is necessary with a higher UART speed than necessary at the highest speeds. This may cause problems for UART's with a limited clock frequency and when galvanic separation and/or RS-232 are used. Most UART's have 3-6 Mbit/s as maximum speed.*

Besides, these speeds are fairly difficult to generate from an integer frequency in the MHz range – especially if a standard UART with 16 samples per bit is used. Therefore, most high speed systems use 2^N × 125 kbit/s, so there are in practice 3 rows. Since the standard UART speeds anyway switches row, and a row shift is desirable for high speed operation, it is reasonable to switch row somewhere between 38.4 kbit/s and 125 kbit/s. 57.6 kbit/s is a standard, but is impractical as this would cause 2 row shifts, so there are only two possibilities left – 76.8 kbit/s from the low row although it is not a standard in the PC world and 62.5 kbit/s from the high row. 62.5 kbit/s is used in some CANopen implementations ***although the standard is 50 kbit/s****, but because 9.6 – 76.8 kbit/s is used in BACnet for building automation, and 76.8 kbit/s is used in P-net for industrial process control, 76.8 kbit/s has been selected. Building automation and industrial process control are very important applications of Max-i.*

With a prescaler, which divide by 13, the clock frequency for 100% accurate low-row speeds (1.2 – 76.8 kbit/s) is 31.9488 MHz, but 32 MHz is used in Max-i, as this frequency is also very well suitable for the high speeds (125 – 4000 kbit/s), which become 100% accurate, and for timers and other purposes, and 32 MHz may even be a standard frequency in many microprocessor systems. The slight inaccuracy of 0.16% it creates at low speeds actually increases the timing margin by making the UART bits slightly shorter compared to the Max-i pulses. The minimum time for transmitting one byte on Max-i is 8, 0-bits equal to 24T – 8% = 22.08T. To keep the maximum time of one UART byte with start and stop bits (10 bits) below this, the maximum bit length is 2.208T. However, the UART bit length is only 13/6T = 2.167T, so there is a margin of 2.208/2.167 = 1.019 and bigger inaccuracies are anyway not allowed for UART communication because if one device is 1.9% too high and the other 1.9% too low, the total drift of the sample point of the stop bit is 36% (9.5 bit), which is just at the limit when the sampling jitter is taken into consideration. At the high speeds, the UART bit length is 16/7.5T = 2.133T, so the margin is 1.035, that is, approximately 1.5 % bigger than for the low speeds.

The maximum bus length depends on the pulse width of the drive pulse, which is (2T - S)(1 - x). For x = 8% (old margin) and S = T/8, the maximum round trip propagation delay becomes 1.725T. With a propagation velocity of 5.06 µS/km for a PE insulated cable (ε_r = 2.3), the maximum net length in km for a network in trunk line topology is: 1.725T / (2 × 5.06) = 0.17T *km/µS. With $T = (6 \times 2^N) / f_{clock}$ µS where 2^N is the clock prescaler, the network length becomes $2^N / f_{clock}$ km. This is the same as the standard lengths of a CAN network if a clock frequency of 2^M MHz is used like for example 32 MHz.*

If a smooth lighting control is wanted, the baud rate shall not be lower than 1302 baud corresponding to a UART speed of 2.4 kbit/s. At this speed, it is in average possible to transmit 11, 7-byte messages (358T) per second.

1.9 Bit Coding

Max-i uses three different bit symbols called 0-, 1- and Start-bit. The table below shows the length of the pause before the drive pulse and shows important time limits.

Bit	Limit	Pause	Pulse	Symbol Time
0		1T	2T+E	3T+E
	Short 0-1			5T
Short-1		5T	2T+E	7T+E
	Long 0-1			7T
Long-1		7T	2T+E	9T+E
	Save			15T
	Reset			16T
Start		≥17T	2T+E	≥19T+E

Fig. 1.24

E is shown gray because it is usually 0 except for very high speeds and/or heavy slew rate limiting.

Since the pulse width of Max-i is usually 2T and the length of a 0-bit is 3 T, the baud rate becomes 1/(3T).

Because the Baud rate is defined as the reciprocal of the shortest symbol width – not the reciprocal of the number of bits per second, Max-i has a fixed Baud rate = 1/(3T), but not a fixed bit rate! However, due to a scrambling, which ensures approximately the same number of 0- and 1-bits, the average bit rate is 2/(3T+9T) = 1/(6T), which is equal to the half Baud rate. This is the same as Manchester coding where the bit rate is also the half of the Baud rate so the two codes are equally efficient, but unlike Manchester coding, the used BWM cannot get out of phase synchronization and is useable for bus arbitration – even with a big propagation delay. Other codes such as NRZ and QAM may have a higher efficiency, but are not self-clocking and therefore require much higher clock accuracy.

The 0-bit and the long 1-bit are also used to implement nondestructive, bit-wise, bus arbitration. If one or more devices have a 0-bit to transmit and one or more devices have a (long) 1-bit to transmit, the result will be a 0-bit because of the shorter pause. The devices, which have a 1-bit to transmit, shall detect this as a collision and shall stop their transmitters when the 0-bit is accepted (in the middle of the pulse) so that the pulse belonging to their own 1-bit is never transmitted. However, the devices shall continue to listen to the line! Maybe the colliding telegram contains wanted information. Note that the collision detection also verifies the transmitted data. If the received data is not equal to the transmitted, the telegram is aborted.

When it is not possible with any more collisions, a switch to short 1-bits and a new detection limit shall be done to increase the efficiency of long telegrams. This increases the average bit rate to 2/(3T+7T) = 1/(5T), which is 20 % more efficient than Manchester coding. As described in layer 2, this shall happen 40 bits after the beginning of the data field if the message is long enough (messages with 4-bit or 20-bit data never reach the limit).

The Save limit (15T) is used as a telegram delimiter. If the bit timer reaches this limit, the telegram shall be regarded as terminated and may be saved. 1T later (16T) the receiver and scrambler (described later) shall be reset, but the decode timer shall continue to run. The device is allowed to start a new pulse (new telegram) if the timer ≥19T or up to 1T after a start-bit from another device is **accepted** (at 2T). This ensures that the bus arbitration works no matter if the device with the highest priority has the slowest clock.

The 17T start-bit pause is used as a telegram delimiter, which synchronizes all devices for each telegram.

The bit timing for 0-bits and long 1-bits and the waveform of the short 1-bit (gray) is shown graphically below:

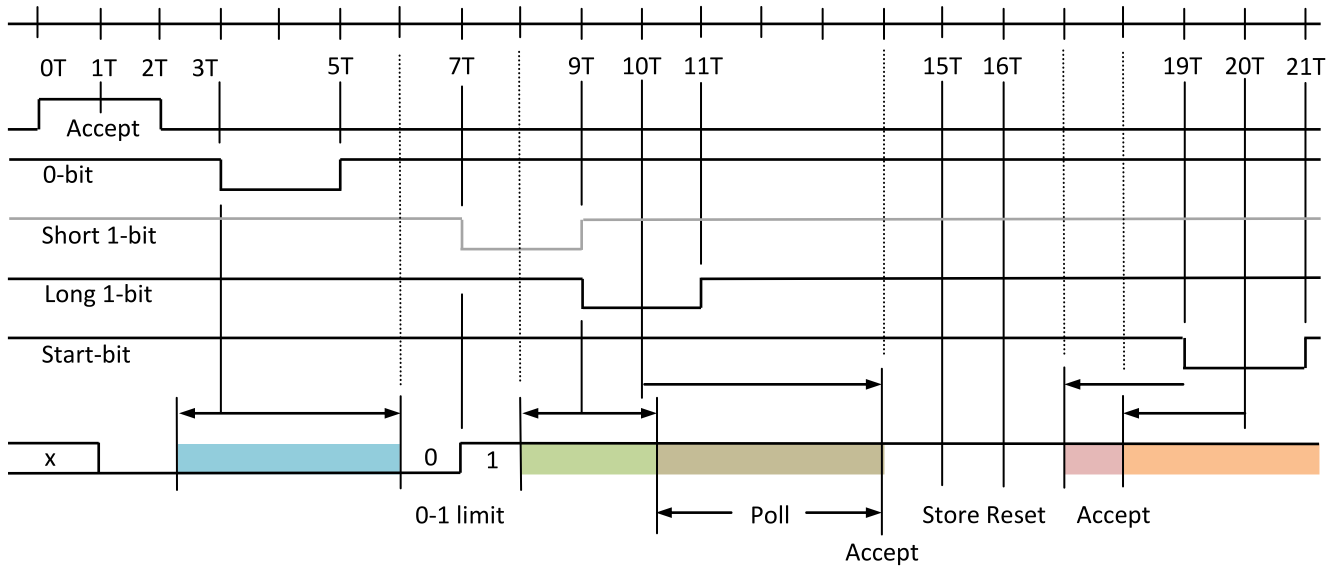


Fig. 1.25

Max-i has been designed for a worst-case timing margin of 1T (½ pulse width) for bit timing and 2T for reset timing (distance between telegrams) as shown with the thin dotted lines**.** The colored fields show the possible deviations due to propagation delay and differences in clock frequency as explained in further details in the following.

*At a speed/throughput of 125 kbit/s, CAN has a maximum bus length of 500 m. This may at first glance seem to be twice as fast as Max-i, which only has a throughput of 63 kbit/s at the same bus length, but if the timing margin is taken into consideration, CAN is only slightly faster than Max-i and only during bus arbitration and the first 40 data bits, but simultaneously **much** more critical.*

*In Max-i, the maximum back and forth propagation delay of the line is contained in the long 1-bit, but in CAN, it must be a part of **all** CAN bits as shown below compared to a **simplified** wave form for Max-i:*

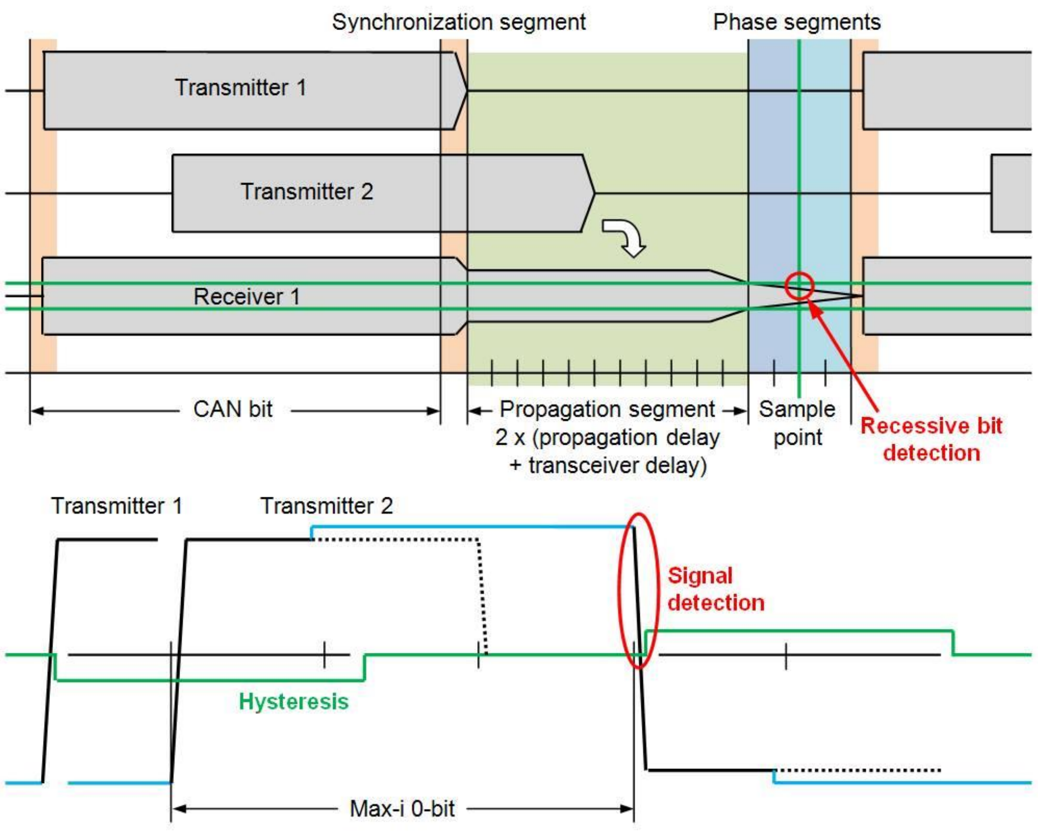


Fig. 1.26

For a given bit length, a big propagation delay will therefore reduce the length of the two phase segments and therefore make the system more critical.

If a timing margin of 1 µs is wanted in Max-i, the maximum back and forth propagation delay is 2 µs, and with a 50 % mix between 0- and 1-bits, each bit will in average take 6 µs up to a data length of 40 bit (long 1-bits) and 5 µs after that (short 1-bits).

If the same timing margin is wanted in CAN, each phase segment needs to be 1 µs long so with the same line length, each CAN bit must be 2+1+1 µs = 4 µs plus synchronization segment and transceiver delay and therefore typical in the order of 5 µs. This is still a little faster than Max-i ***during bus arbitration****, but the timing in CAN is usually not optimized after the line length, but just a standard setup, and since the bit detection is completely uncritical in Max-i, but* ***very*** *difficult in CAN as shown with the red circles on the figure, it may be necessary to reduce the CAN speed. For applications where reliability is of utmost importance and especially on long lines, Max-i is therefore at least as fast as CAN.*

1.9.1 Timing for Counteracting

The purpose of the 1T pause in the 0-bit is:

- *To ensure that more transmitters* ***never*** *counteract each other during bus arbitration (a slow device still has its transmitter on when the next pulse from a faster device is initiated).*

  Suppose a fast device (red waveform below) and a slow device (orange waveform) are connected together and the fast device starts to transmit a pulse with such a rise time that R + D is slightly less than S. Because R + D < S, E = 0 so the next pulse from the fast device may be transmitted at 3T(1 - x) in case of a 0-bit. The worst-case situation occurs if the slow device sample just before the pulse from the fast device is detected as shown below. The voltage offset on the right side of the figure is a typical offset caused by the reflections, which must be able to increase the line voltage slightly to be able to turn all transmitters off by means of their serial diode.

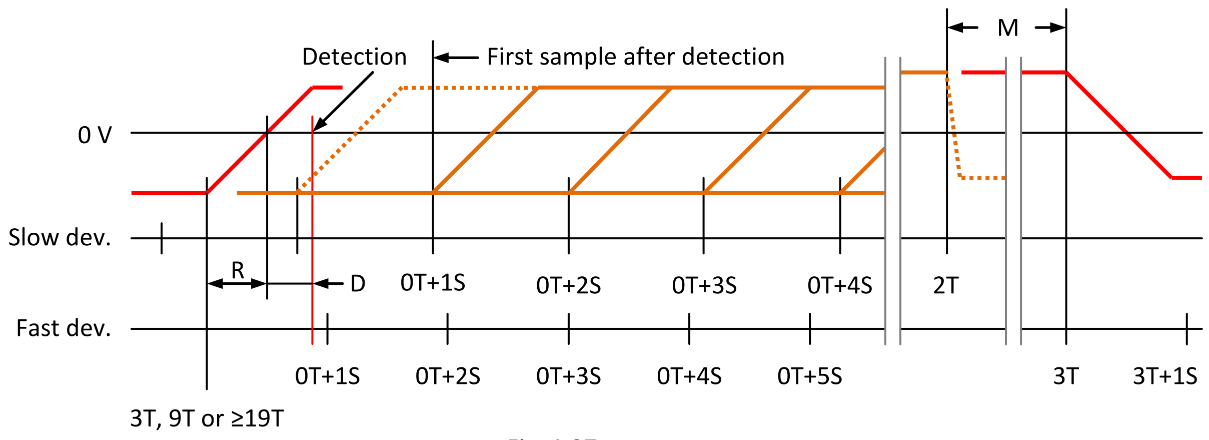


Fig. 1.27

The slow device may generate a pulse, which starts at any sample up to 1T – even the sample before the signal is detected as shown with the dotted line, but since the ***new*** *timer is counted forward to 0T+1S at the first sample after the detection, the pulse is always terminated before 2T(1 + x) + R + D so the worst-case safety margin is:*

$$M = 3T(1 - x) - (2T(1 + x) + R + D) = T(1 - 5x) - (R + D)$$

Since x = 6% and R + D ≈ S(1 - x) where S = T/8, the safety margin becomes 0.58T.

Any propagation delay between the two devices does not matter except for an increased R, and the power is shared between the devices. If for example two devices in each end of a maximum length line start to transmit simultaneously, the transmitter current will just be reduced to zero or almost zero when the signal from the other device arrives 0.88T later. Each device will therefore contribute with the half power. If a transmitter in one end is delayed so much that it starts to transmit when a signal from the other end arrives, the transmitter current will be zero due to the reflections and because of the serial diode, the reflected signal will not be pulled down so the transmitter will be invisible. The transmitter will therefore not counteract a new 0-bit from the other end even though a 0.88T delay (line length) of the period where the transmitter was activated (but didn't draw any current) overlaps the new 0-bit.

In case of reflected wave switching, the speed is reduced N times compared to the maximum speed for a given line length to be sure that the signal is detected. In this case, the reception delay may vary between (R + D) for devices close to the synchronizing device to approximately (R + D + 0.4T) as described in chapter "A.10.1 Detection Delay with Reflected Wave Switching" in Annex A. The worst case safety margin is therefore:

$$M = 0.58T - 0.4T = 0.18T$$

This is still positive and because the serial resistance of the cable is >1 kΩ in this case as shown on the simulation, a counter action would not matter.

- *To avoid that the incident voltage levels get lower and lower on consecutive 0-bits due to the skin effect (see Annex A). Without this pause, the incident voltage level far from the transmitter may for example be ±3V for 1-bits at the minimum cable cross section, but only ±1V for 0-bits. To reduce this effect to an insignificant level, the pause must at least be in the order of the propagation delay of the line, that is, approximately 50% of the pulse width.*
- *To make it possible to sample and integrate the voltage level after the reflections, but before the next pulse so that the clamp level may be automatically optimized in the future.*
- *To reduce the amount of harmonics in the drive signal. With a 50% pause, the drive power in a row of 0-bits becomes a synthetic sine wave, which does not contain 2nd, 3rd or 4th order harmonics.*
- *The pause has the further benefit that it enables a much optimized UART timing as explained later.*

1.9.2 0-bit Timing for Incident Wave Switching

The worst-case delay for 0-bit detection occurs during bus arbitration where the synchronization is handled over from a fast device A_F in one end of the line to a slow device B_S in the other end so that the next 0-bit is delayed ***when seen from A_F and other fast devices close to it*** *as shown and described below:*

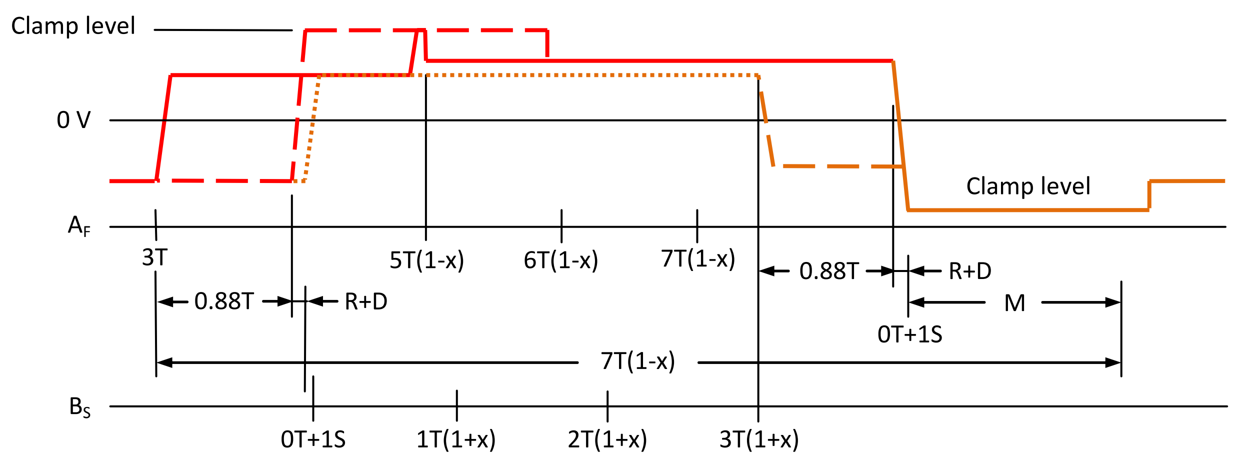


Fig. 1.28

The red waveforms are driven by A_F, and the orange waveforms by B_S.

- *A slow device B_S in one end of a line is synchronized by a faster device A_F in the other end, but due to the propagation delay, B_S may be up to 0.88T + R + D behind A_F in time if it samples just before the signal is detected and since $R + D \le S(1 + x)$, it may therefore be delayed up to almost 0.88T + S(1 + x).*
- *Both devices try to transmit a 0-bit. The waveform from A_F at B_S is shown with the red dashed line, but the transmitter in B_S does not draw any current as the reflection of the wave from A_F turns it off. This is shown with the dotted orange line.*
- *3T(1 + x) later, B_S transmits another 0-bit shown with the dashed orange line, which will be received at A_F 0.88T + R + D ≈ 0.88T + S(1 - x) later if A_F is going to transmit a 1-bit or its previous 0-bit was the last bit in a poll. This turns the synchronization over to B_S if there are no other devices, which want to transmit.*

The worst-case safety margin for detection of the 0-bit as shown on the figure is therefore:

$$M_F = 7T(1 - x) - (0.88T + S(1 + x)) - 3T(1 + x) - (0.88T + S(1 - x)) = T(3.75 - 10x) - 1.76T$$

For x = 6% and S = T/8, the safety margin becomes 1.39T, which must be divided in three parts. Two parts for any increased rise time due to the skin effect and one part (MB) to ensure a reliable bit decoding even under worst-case conditions. Since Max-i is designed to have a minimum safety margin for bit decoding of 1T, the maximum rise time delay becomes 0.2T, which is acceptable because in this case, the devices are located in each end of the line and will therefore benefit from reflected wave switching where the signal is amplified by the reflections and therefore has a faster rise time and does not suffer from skin effect knees.

The maximum rise time delay (RTD) for counteracting is 0.58T as described previously, but such a big delay usually does not occur and cannot occur if the devices are located closer than a distance corresponding to the half of the rise time delay (RTD/2) from the end. If they are, the reflected wave will limit the delay.

The safety margin for bit detection may therefore be calculated as:

$$M = M_F - 2 \times RTD + 2 \times RTD/2 = M_F - RTD$$

For M_F = 1.39 as calculated above, x = 6% and S = T/8 and M = 1T (minimum requirement), the maximum rise time delay therefore becomes 0.39T. The edge of a 0-bit will therefore always fall in the blue area of fig. 1.25.

1.9.3 0-bit Timing for Reflected Wave Switching

With reflected wave switching, T is at least doubled (N = 2) compared to the propagation delay of the line, but at the same time it is possible for two devices to be up to 0.4T out of synchronization so the formula becomes:

$$M_F = T(3.35 - 10x) - 1.76T/N$$

For x = 6% and N = 2 (two waves), the safety margin is increased from 1.39T to 1.87T and with more waves (N >2), it is increased even further.

1.9.4 1-bit Timing

If one or more devices are transmitting long 1-bits (synchronized to the fastest one) and a slow device is listening, the safety margin for correct detection of the 1-bit is:

$$M = 9T(1 - x) - 7T(1 + x) = 2T(1 - 8x)$$

For x = 6%, M becomes 1.04T, which is just above the requirement 1T.

For short 1-bits, where the sample point is 5T, the safety margin is:

$$M = 7T(1 - x) - 5T(1 + x) = 2T(1 - 6x)$$

For x = 6%, M becomes 1.28T, which is even better.

The propagation delay and the detection delay do not matter and are therefore not included in the formulas so the safety margin is the same for incident and reflected wave switching. Since the sample points 5T and 7T cannot be increased with S (only decreased), the worst-case situation occurs when the slow device is not ahead of the transmitting devices.

Seen from a fast device, which does not transmit and therefore is synchronized by a slow device (a fast device, which transmits, is never synchronized), the latest possible arrival of the edge of a long 1-bit is:

$$9T(1 + x)/(1 - x)$$

For x = 6%, this is 10.15T. The edge of a 1-bit will therefore always be detected in the green area of fig. 1.25 except if the slow device is polled and therefore no longer synchronized.

1.9.5 Save and Poll Timing for Incident Wave Switching

If the first bit of a poll reply is a 0-bit, the situation is exactly the same as the worst-case detection of a 0-bit as described previously. However, if a slow device has a 1-bit to transmit, the safety margin for the ***acceptance*** *of this 1-bit before acceptance of the telegram may take place at 15T is:*

$$M = 15T(1 - x) - (0.88T + S(1 + x)) - 9T(1 + x) - (0.88T + S(1 - x)) - T(1 - x) = T(4.75 - 23x) - 1.76T$$

Where T(1 - x) is the time it takes the fast device to accept the pulse after it is detected. For x = 6% and S = T/8, M becomes 1.61T. For the required safety margin of 1T, the margin for rise time become 0.6T, which is more than for 0 bit detection and approximately the same as for counter acting (0.58T).

In case of fast poll, the edge of a 1-bit may also fall in the dark green area of fig. 1.19. Note that since storage takes place if the timer reaches 15T, it is necessary that the pulse is accepted before that time – not just detected.

The purpose of the 1T delay between save and reset is:

- *To simplify the receiver as it avoids critical race between save and reset.*
- *To increase the safety margin to reset in case off a too slow reply to a poll (wide save pulse).*
- *To delay reset so much that a serial communication channel may immediately go into Break condition at the time of reset as described later.*

1.9.6 Save Timing for Reflected Wave Switching

In this case, the formula becomes:

$$M = T(4.35 - 23x) - 1.76T/N$$

For x = 6% and N = 2 this is 2.09T.

1.9.7 Reset Timing

The long pause in the start bit ensures that all devices are reset before a new telegram can be received (accepted). The worst-case situation occurs when a fast device wants to transmit at 19T. In this case, the safety margin for reset of a slow device is:

$$M = (19T(1 - x) + T(1 + x)) - 16T(1 + x) = T(4 - 34x)$$

The first part is the time it takes before the pulse from the fast device can be accepted at the slow device. For x = 6%, M becomes 2.0T, which is the requirement for reset timing. This is shown with the orange field on fig. 1.25. The edge of the pulse may occur in the pink field, but since the pulse is not accepted until 1T later, this does not reduce the safety margin. The propagation delay and the detection delay do not matter and are therefore not included in the formula, so the safety margin is the same for incident and reflected wave switching.

1.10 Priority Bit

Each telegram consists of a start-bit, a priority bit and N × 8 data bits. The priority bit determines the master priority of the telegram. It is used for a so-called "babbling-idiot" protection, which ensures that **no** device or devices are able to saturate the bus except for special devices dedicated to transmission of time telegrams or emergency telegrams (must not be used for any other purpose). Unlike many other CSMA-CD busses as for example CAN, it can therefore be guaranteed that **all** telegrams come through – even if all devices want to transmit.

The priority bit is initially set to zero (high priority) except for devices with a fixed priority. When the **full arbitration part** of a telegram, that is, the first part of the telegram with the identifier (see layer 2) has been transmitted, the priority bit shall be set to one in all following telegrams from that device until the device detects a telegram from **any** device – including itself, which also has the priority bit set to one. When this happens, the priority bit shall be set to zero again until the device has transmitted a new full arbitration field. In this way, all devices get a chance of transmitting. Each time a device transmits; one more device will have its priority bit set to one. When there are no more telegrams with priority zero, one of the telegrams with priority one comes through – the one with the highest priority. When this happens, all devices shall reset their priority bit to zero, and the process starts all over again. Note that if a device transmits a telegram with priority bit = 1, it will be upgraded to priority zero when it receives its own priority bit, but if it wins the bus arbitration, it will be downgraded again when the last bit of the ID has been transmitted. In this way, all devices on the bus will get priority zero except for the device, which has just transmitted. Also, note that the content of the rest of the telegram is completely unimportant for this mechanism.

It is only the transmitter in the device, which transmits a full arbitration field, which is down classified to a lower priority. The priority bit shall **not** be changed if the telegram is just an answer to a **fast** poll (described later).

Like the start bit, the priority bit is entirely added, removed and controlled by the Max-i controller. Bit 0 of the telegram is the first bit after the priority bit.

A SCADA system may have many data to transmit, but nevertheless it is just handled as any other device, that is, downgraded after just one telegram with the priority bit equal to zero. The majority of data from a SCADA system are anyway polls, which should have a lower priority than event driven data from the process.

The "babbling-idiot" protection is especial important for boolean telegrams, which are usually transmitted each time a bit changes state (event driven) and may even be transmitted at a high heartbeat rate.

The "babbling-idiot" protection uses the normal bit-by-bit based bus arbitration. With the big clock inaccuracy of Max-i, this is much more efficient than a system based on an extra time slot, where worst-case inaccuracies and propagation delays may cause very long idle times (>60T).

1.11 Telegram Separation

In case of the Modem object (UART communication), the various telegrams through that channel shall be separated by means of a break character, that is, a low line state for at least 10 bits (start bit, 8 data bits **and stop bit**). The worst-case timing for UART speeds ≥125 kbit/s (worst case) is shown below (not quite to scale):

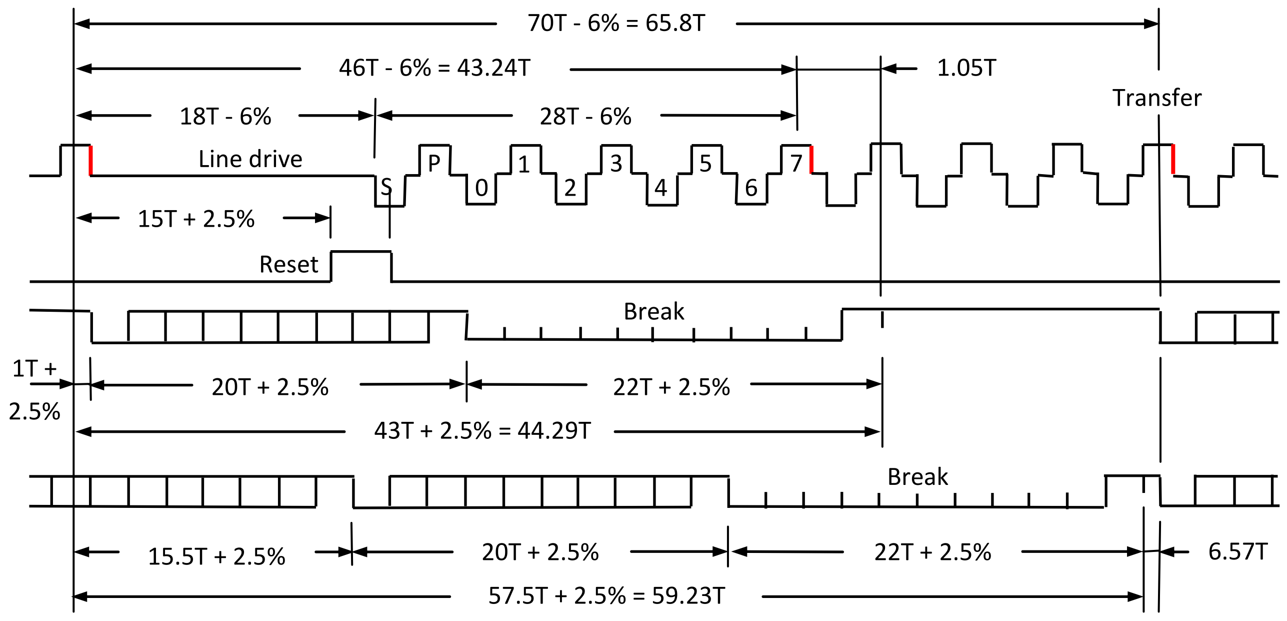


Fig. 1.29

The drawing illustrates the pause and the drive pulse of each bit. In practice, the signal will not fall back to zero (midpoint), but change polarity on the rising edge of each pulse as shown previously and in Annex A.

A device, which uses the UART, shall have a maximum inaccuracy of ±2.5 % for a reliable UART operation even though the Max-i communication accepts ±6 %.

Each UART bit has a width of 16/8T = 2T for 125 – 4000 kbit/s and 13/8T = 1.625T for 1.2 – 76.8 kbit/s.

Because it may take 1.05T less time before the first byte (plus start bit and priority bit) of a new telegram has been received than to transfer the last byte of the previous telegram plus a break character, it is necessary that all telegrams contain at least one 1-bit (extra 6T) to absorb this delay, but this is always the case even for network messages. It is also necessary that the UART is double buffered, but this is anyway necessary to be able to compare 16 bits before the telegram is transferred as explained in Layer 6.

It may be convenient to transfer the byte in the buffer to the serial output register when the pulse is accepted and have been compared at 1T, and then transfer the received byte to the buffer at the red edges in the drawing, that is, 1T later. In this way, only a single buffer is needed in spite of the 16-bit comparison. Because a byte can never be shifted out before it is available in the buffer, all bytes including the last one are always delayed at least 1T.

When a reset is generated, the UART shall generate a break character as soon as the stop bit of the last byte has been transferred. This will usually occur 21T after the last bit has been accepted as shown on the third curve on the figure, but this is not guaranteed. There is a small chance that the byte is delayed. This may happen if there are less than four 1-bits in the telegram after the first two bytes. The worst-case situation with only one 1-bit is shown on the lower curve, but as can be seen, the margin is still 6.57T, so break characters up to 14 bit can be accepted.

If a programmed delay between the telegram identifier and the data (described later) is longer than the telegram, the message shall be terminated with the last received character plus the break as described later. In this case, transmission of the last received character may begin at the storage time, which is 1T before reset. The safety margin in this case is therefore 64,4T - ((14T + 20T + 22T) + 2.5%) = 7T.

If the speed is changed by means of a network message as described in layer 6a, the speed may change at the storage time 1T before reset. This is the right time for all I/O objects except for the modem object where it will cause the break character and perhaps also more or less of the last character to be transferred at a wrong speed. To prevent this, the speed shall be latched **in case of the Modem object** until at least 3/4 of the stop bit of the break character has been transmitted.

On the **transmitter** side, all telegrams are also separated by means of a break character.

1.12 Optional 9-bit Communication

While many UART’s are able to detect a break condition, very few are able to generate it in a simple manner. A solution may be 9-bit communication. In this case, all telegram bytes should have the 9th bit set to 1 so that it works like an extra stop bit (a UART transmits the least significant bit first) and a telegram delimiter may then be generated by means of the 9th bit set to 0 in the last received byte so that it works like a framing error (Break). Since the UART and the **transmitter** use the same clock in the Max-i controller, 11 UART bits use 22T at 125 – 4000 kbit/s, but 8 0-bits use 24 T so there is a good safety margin. For 1.2 – 76.8 kbit/s, 11 UART bits only use up to 18.3T, so the safety margin is at least 4.2T. This makes it easier to use very simple single-buffered UART’s without automatic flow control.

On the **receiver** side, 11 UART bits at 125 – 4000 kbit/s may use up to 22T + 2.5% = 22.55T and 8 received 0-bits may use 24T - 6% = 22.56T, which just fits.

9-bit communication however has a lot of compatibility problems, which is the reason why it is not used as standard:

- *It is not supported by SerialPort of Microsoft .Net and the 16C550 - 16C750 series of UART's. This makes it useless in the PC-world and this alone makes it out of the question as the only possibility.*
- *It is not supported by DMX512 lighting controllers. Such controllers can drive Max-i directly at 250 kbps if automatic flow control is added, which may be very simple to do in hardware.*
- *Most other communication systems such as Ethernet use a stream of 8-bit data, which is then divided into frames. Such a stream can easily be transferred to Max-i by means of an 8-bit wide FIFO (or DMA), and the framing can then be replaced with an added Break when the FIFO is empty and there are no more data to load. With 9-bit communication, a 16 bit FIFO is needed and every byte has to be expanded and the last one marked as a frame delimited. This makes it impossible to use the content of a FIFO (empty it) before the entire message has been received and the last byte marked as delimiter, which completely destroys the purpose of a FIFO*

Layer 2, Data Link Layer

The Data Link Layer is responsible for converting small packages of data to raw bits for Layer 1. It controls the bus access, it creates and recognizes frame boundaries and value identifiers and it performs error detection.

2.1 Chr5 Format

For some applications, Max-i uses a special 5-bit text format called Chr5. The format is a simplified (truncated) ASCII code consisting of only English capital letters as shown below:

Code	Letter
00000	Serial number
00001	A
00010	B
.	•
.	•
11001	Y
11010	Z
11011	No letter
11100	NU
11101	NU
11110	NU
11111	NU

Fig. 2.1

The "Serial number" code 00000B is used to specify that a value identifier is based on a serial number as specified later.

The "No-letter" code is used in case of a letter field, where all letters are not used. Unlike a space, the no-letter code is not written out.

The reason for not using the last four characters (NU) is to enable the code to switch between long and short identifiers as described later.

2.2 Attributes and Object Model

In Max-i, all values like the published and consumed values, hour counter, error word, all configuration and calibration parameters etc. are stored in memory locations called attributes, which are grouped together in objects similar to the object model of CIP, CANopen etc. Max-i uses a very simple object model with only two types of objects – network object and I/O objects. Except for SCADA systems and other controllers, each device contains one network object and at least one I/O object.

The **network** object consists of 8, 36-bit, **write-only** attributes, which are used for values, which are common to all devices on the network, such as system time, passwords (note, write-only) and transmission speed. The network object does not need an object identifier as it is not necessary to be able to address individual network objects or read the values (it has no meaning to read/poll a value, which is located in all devices).

The **I/O objects**, which define the function of a device such as MODEM, SPI, BOOLEAN, KEYPAD, LAMP and ADDA (AD and DA converter), consist of 16 – 1024 attributes where the last 16 are standardized as specified in "Layer 6a, Presentation Layer. Attributes".

To simplify the addressing, the 10-bit (0 – 1023) attribute range is divided into 64 blocks of 16 attributes and all I/O objects use an integer number of blocks. The most significant 6 bits of the attribute number is called the object base and is used together with an object identifier to address an I/O object. In this way, a simple Max-i controller with only one I/O object with the 16 standard attributes may simply use the object base together with the identifier as an acceptance filter and use the least significant 4 bits to address the attributes so that no adders are necessary.

The default object base shall be 111111B = 63. If an object needs more blocks, the attribute address shall wrap around. For an object with the default object base, the first attribute of the first extension block will therefore get the address 0,0000,0000,0B = 0. This is important if the controller is used as a modem and connected to a complex device like a variable speed drive or mass-flow gauge, which uses the attribute range 1 – 999.

Complex devices typical use 200-300 attributes located in logical groups in the 3-digit range from 1 to 999 where the most significant digit indicate the group. To be able to use Max-i without changing such numbering systems, the default object base is 63. This also fits with default "1's", which is the standard of Max-i.

If all 1024 attributes are not used, which is usually the case, the object base is also used to enable more devices to broadcast the same implicit messages and in this way control the same devices. This may for example be used for more control panels and for multi-way landing switching of lamps. In this case, the available 64 attribute blocks are shared between all control devices, so if they all only use the 16 standard attributes, up to 64 controllers are possible plus any number of computer-based systems like SCADA systems and debuggers, where the Max-i controller just works as a modem and implicit control messages are generated in software.

Most other fieldbus systems, which use the publisher/subscriber model such as the CIP family and CANopen, have a mechanism to ***prevent*** *multiple use of the same identifier. This protects against errors, but it is simultaneously a* ***very*** *serious limitation for many applications. In Max-i, unintended multiple use of the same identifier is avoided by means of numbering systems like PNS and railway numbering, which makes it possible to use the device numbers as identifiers instead of meaningless numbers.*

Unlike many other object oriented fieldbus systems, Max-i has no separate identity object. This is not possible for three reasons:

- More devices from different vendors shall be able to share the same identifier to be able to control the same devices and therefore cannot have a common identity object connected to this identifier.
- Process equipment may consist of many different devices from different vendors located in different locations.

  The basic idea of Max-i is to use one small IC for each value instead of a common control device with more functions, so by far the majority of Max-i controllers only have a single I/O object. If process equipment needs more devices, they are typical not located in the same spot and therefore cannot belong to a common controller and have a common identity object. For example, the device, which controls a big motor contactor and generates an acknowledge, when the motor is running (draws current), is typical located in the terminal board with the heavy copper bus bars, but devices, which for example measure the motor temperature, speed and vibrations, must be located close to the motor.
- There is no bit to select between I/O objects and identity object and extra identifier bytes for object type and instance would lower the efficiency too much.

Instead, the object type is a part of the I/O object specification attribute, and the identity attributes are common to all I/O objects in one device so that changing these attributes in one I/O object shall affect all other I/O objects in that device.

It is obvious that any publisher must have a unique identifier and/or a unique object base, but this is also the case for subscribers, which also have attributes to setup and must be able to generate error telegrams. However, since there is at least one publisher and may be more subscribers for a given value (same identifier) there are only three possibilities for an individual subscriber addressing:

1. *It must use its own object base. In practice, this may be difficult or impossible since in that case, the attribute range must be shared not only between the subscribers, but also with the publisher. If the publisher is a complex device, there may be no attributes to share.*
2. *It must have its own unique identifier like a publisher. This solution makes it possible to use any number of subscribers and it makes it possible to use the same attribute number for different things in a publisher and a subscriber and in this way reduce the necessary number of standard attributes for each identifier. However, in practice, this is very impractical because it makes it impossible to interpret a telegram with an attribute value if it*

is not known whether an identifier belongs to a publisher or a subscriber, and the solution is very impractical if the device includes a publisher since it is then necessary with two identifiers for the same device.

3. *It must share the attributes with a publisher. This solution also makes it possible to use any number of subscribers and it fits very well with most practical devices, which have both inputs and outputs like push buttons (publisher) with lamps (subscriber), motor contactors (subscriber) with acknowledge (publisher) or controllers used as a modem with two-way communication. A big advantage of this solution is that it is easy to program a subscriber even if the subscriber identifier is not known. You only have to trigger a publisher telegram and can then read the publisher identifier and use this for subscriber programming. Besides, a publisher can also benefit from a subscriber to make it possible to do simple logical functions like OR and AND. For example, if it shall be possible to switch a lamp (A) on from two switches, but at the same time have another lamp (B), which is only activated by one of the switches, it is necessary with an OR function for lamp A. If a publisher also includes a subscriber, it is easy to implement such a function by connecting the output of the subscriber back to one side of a switch and the other side to a fixed On/high. Lamp A will then turn on if the switch is connected to the fixed On/high OR if the output from the other switch – and therefore the output from the subscriber – is activated. In the same way, it is easy to implement an AND function by connecting the input switch in series with the output from the subscriber.*

Because solution 3 is the most appropriate, 1 input, 1 output and attributes for identity is combined in one I/O object.

The identifier, which is used as publisher ID and to select an I/O object, is called the **Value ID** and the identifier, which is used to select the wanted telegrams for the output, is called the **Subscriber ID**.

2.3 Telegram Frame and Bus Arbitration

Max-i uses synchronous communication. The telegram frame consist of a start bit, a priority bit, a local or read bit, an optional attribute number, a value identifier (address), a data field and a signature as shown below:

Start bit	Priority	Local/Read	(Any attribute)		Identifier	Data	Signature
1 bit	1 bit	1 bit	111,111	10 bit	15 or 31 bit	N × 16 + 4 bit	20 bit

Fig. 2.2

The priority bit, the local/read bit, any attribute bits and the identifier bits are used for bus arbitration. During this part, it is common to switch over from one device to another. During the data part, there is usually no bus arbitration except for three situations:

- Special protocols, which use a common identifier during power-up, but transmits a unique vendor ID and serial number as data to make the telegram unique.
- A collision between a 20-bit message for example to set the light intensity of a display and a longer telegram for display data such as text and graphics. In this case, collisions may take place up to 40 bits after the beginning of the data field.
- Multiple use of the same value ID for example for multiple control panels and multiway landing switching for lamps where the data may or may not be different. If the data is the same, the signature will also be the same, and since the maximum data length for a Max-i controller is 36 bits, it is usually only necessary with bus arbitration up to 36 bits after the beginning of the data field. In case of longer data for example for a display or a common telegram for more Max-i controllers, a dummy byte, which is different for all devices with the same ID, may be inserted within the first 5 bytes of the data field to ensure bus arbitration before 40 (or 36) bits.

Since all collisions after 40 bits after the beginning of the data field are very seldom and can easily be avoided, it is possible to increase the efficiency of longer messages by means of a shorter 1-bit in the last part of the telegram as specified in Layer 1, chapter "Bit timing". This is however primarily important for long, joint telegrams to for example stage lamps, CNC machines, robots and cobots. 4-bit and 20-bit messages never reach the limit and there is only a minor increase in efficiency for 36-bit messages.

If a device wants to participate in the bus arbitration, this shall be done before the priority bit, but it is allowed that it doesn't transmit the start-bit itself (done by another device). The latest possible time is therefore 1T after the start-bit from another device has been accepted, that is, at 2T. Because of the big clock inaccuracies, it is very important that reception of a start-bit is not considered as a busy line and therefore blocks the device. This may happen in other fieldbus systems using bit-wise bus arbitration!

One of the unique things about Max-i is that the device, which has won the bus arbitration, not always transmits the last part of the telegram with the data and signature. This is used for a unique fast polling system. Instead of the usual poll telegram, which is used on probably all other fieldbus systems, the polling device just transmits the first part of the telegram with the identifier. The device, which has the wanted data, may then complete the telegram by **immediately** adding the data part and signature (synchronous communication) as if it had won a bus arbitration. This is described in details later.

The first bit of the data field is included in the arbitration field for three reasons:

- *It is utilized for the fast polling system.*
- *It makes it possible for a device to resynchronize in case of a collision between a poll and a broadcast of the same value. In this case, the arbitration part of the telegram is the same, but transmitted from two different devices, so if a device has synchronized itself to the polling device it needs to resynchronize.*
- *It makes it possible for two devices to work together on the same boolean value for example in case of two limit switches on the same valve. One device sends 0xxxB and the other device sends 1xxxB.*

If the line is resistor terminated (mode 2) as described in chapter 1.7 and a master/slave protocol is used to enable a longer line than possible with bit wise bus arbitration, fast poll cannot be used either. Instead it shall be possible to program the device to trigger a normal implicit transmission instead so that the data on the line becomes the ID from the poll followed by a new implicit message with the same ID. This is done in the Password 2 attributes as specified later.

2.4 Identifier and Publisher/Subscriber model

Max-i uses the very efficient publisher/subscriber (producer/consumer) model known from CAN. With this model, it is in principle **not** the various devices/bus-nodes, which are addressed and has an address, but the various values like commands, process status, voltages, currents, temperatures, pressures, levels, speeds etc.! These addresses are called identifiers (ID) and the ID for the published **main** value is called the Publisher ID.

Most messages like I/O data from the process may be regarded as broadcast information no matter if they are initiated by an event or polled! Many devices may subscribe to the same message and benefit from data requested by other devices. There is no difference between input and output messages. If for example a SCADA system wants to set an output, it just broadcast/publish the output value. Any device may then pick up this telegram and set its outputs according to this.

The publisher/subscriber model is very useful in for example redundant SCADA systems where the method efficiently reduces the necessary number of telegrams and guarantees data consistency – even without a common database – so that all devices show exactly the same values. It is also very efficient for synchronizing for example clock, program execution, set points etc. in more devices, and it is very efficient for information systems where the same information shall be shown on many displays. As every single value – even boolean values – has its own identifier, there is no need for AND and OR functions, and the publisher/subscriber model fits very well with XML, where each value also has a name.

The publisher/subscriber model has the further advantage that there is no need for address stacking and address conversion in case of gateways between different bus systems. This reduces the length of the telegrams and makes the gateway function simple and fully transparent so that a gateway may be added or removed at any time without any changes in the bus nodes.

To be able to program and read the setup attributes in the devices, which produce and consume these values, it is however also necessary to be able to address the devices themselves. This shall **also** be possible by means of the publisher ID, but to be able to (re)program an un-programmed device or a device with an unknown or faulty publisher ID, it shall **also** be possible by means of a modified serial number where bit 13 – 17 and maybe bit 1 are replaced with zeroes as specified in chapter “Attribute 13 – Serial Number” in layer 6a. Therefore, all numbering systems, which use the 31-bit identifier, shall be made in such a way that bit 13 – 17 = 000,00B is never used! An easy way to ensure this is to use a Chr5 character for these bits as shown in the examples below (Chr5 = 00000B indicates a serial number).

31-bit Industrial PNS identifier						
Equipment					Function	
Number (1 – 3999)		2-letter Code			Number	1-letter Code
Number (1 – 999)	A,B,C Suffix					
10 bit	2 bit	3 bit	2 bit	5 bit	4 bit	5 bit

31-bit Railway identifier			
MVB address (1 – 3999)	Vehicle/trainset Identifier		
	1-letter Code		4-digit Number
12 bit	3 bit	2 bit	14 bit

31-bit Manufacturer Serial Number for ASIC								
0	Least significant 11 bits of number of devices this week	Serial number Code		Production year	Production week	Production location	
0	11 bit	000	00	7 bit	6 bit	1 bit	4 bit

31-bit Manufacturer Serial Number for FPGA					
Major revision	Minor revision and test versions	Serial number Code		Parity bit 18 to 31	13-bit running number
5 bit	7 bit	000	00	1 bit	13 bit

15-bit Value Identifier	
Number (1 – 3999)	Short
12 bit	111B

15-bit Network Identifier	
000,0000,0000,0B	Att. #
12 bit	3 bit

15-bit Group Identifier	
Prefix	Group number
111,1101B	8 bit

Fig. 2.3

To make the serial numbers unique when they are used as identifiers, bit 13 – 17 are fixed to 00000B as shown. All numbering systems shall accept this, and the Publisher ID shall therefore always have bit 13 – 17 ≠ 00000B.

To be able to recognize and (re)program new, un-programmed devices and devices with a parity error on the Publisher ID, they shall transmit a serial number (attribute 13) message with an ID equal to the serial number during power-up (together with any error messages). If it is desirable to use the serial number permanently as ID, it should be copied to the publisher ID attribute **without** modification. This shall remove the log-on message during power-up and indicate that the number is no longer just an out-of-the-box number.

The identifier shall be regarded as short in three cases:

- If bit 13, 14 and 15, which are marked with blue, are 1.

  Since this is the case for the four Chr5 letter codes 11100B – 11111B, they cannot be used.

- If all 12 ID number bits are 0...0B as this is used to publish network messages as described later.

- If the most significant 7 bits are 111,1101B as this is used to publish group messages.

In **all** other cases, the identifier is long. In case of other numbering systems, the same letter position may be used or at least one of the three bits must be 0 to select the long identifier. An example of this is the use of standard CAN-A and CAN-B identifiers where bit 13 (IDE inverted) is used to select between long and short identifiers (0xxx...xxB or 111B as described later).

The advantage of this selection method is that the length of the long identifier is not reduced by a long/short selection bit. This is necessary with PNS and railway applications, which require all the bits they can get to squeeze a machine, MVB address, function or wagon number into a 31-bit field. The disadvantage is that the short identifier is reduced by two bits, but since it is anyway not realistic with more than 4000 devices on one bus, this is not a practical limitation, and compared to CAN, Max-i has one more bit. Besides, the three unused bits of the short identifier is utilized for 8 network object attributes without disturbing the 1 – 3999 number range (described later).

For industrial applications, it is recommended to use the long 31-bit identifier together with PNS, as shown. PNS is a universal numbering system based on alternating numerical digits and Chr5 letters like for example BC361AS1 for Belt Conveyor 361A, Speed 1 or HX127T2 for Heat Exchanger 127, Temperature 2. PNS uses the single-letter code Q (quick) for short 12-bit identifiers like for example Q0001 or Q3999 so that short identifiers may be mixed with long identifiers in the same numbering system. PNS may also be used to ensure correct data interpretation if the same data type like TIME and PATTERN has different interpretation depending on whether it is used for process signals or for control lamps (data type LH = **H**uman interface control **L**amp). A detailed specification of the various PNS codes can be found in the PNS specification. Other numbering systems may be used for other applications like for example railway applications, home automation and traffic lights.

To avoid that a PNS identifier is mistaken for an attribute or a group number (000000000000B, 1111101xxxxx111B or 111111xxxxxxB as described later), the range of the PNS device numbers shall be limited to 1 – 999 (1111100111B) and placed in front of the A,B,C suffix (a 11B suffix before 111110xxxxB would create 11111110xxxxB, which would be regarded as an attribute), and the MVB address shall be limited to 1 – 3999 (111110011111B). If the ABC suffix is not needed, the left shifting has the further advantage that the number range may be expanded to 12 bits in the same range (1 – 3999) as the MVB address.

The reason why the function code of PNS is the least significant bits, is to easy acceptance filtering in case of devices, which generate more values, but only 8 values of each type is needed (only the 3 LSb's of the number is used). In that case, the device can just use the three most significant bytes to determine whether it shall process the message or not. The coarse filtering of the two first bytes may be done by means of the acceptance filter in the Max-i controller so that the connected device only needs to check a single byte to determine if the telegram shall be processed. Besides, the position fits with the equipment number, which also has the code as its least significant bits.

Other fieldbus systems use either a special bus-address – typical 0 – 255 – and/or a unique 48 – 127 bit serial number, or they use an automatic numbering based on the order of which the devices are placed on the bus. However, there are many advantages of using the equipment numbers directly:

- *There are no cross-reference tables and special bus addresses to manage.*
- *There is no risk for any confusion if the equipment and function number is programmed into the device during mounting.*
- *It is very easy to connect a Web-interface or a debugger and communicate with equipment, because no other knowledge than the device and function number is necessary.*
- *It is easy to debug a communication because the device numbers are transmitted directly.*
- *The safety is increased as less than 1 ppm of the possible numbers is utilized. It is therefore extremely unlikely that a small error creates a used address, which also passes the ID check, so there is virtually no risk of masquerading.*

- *The address does not change in case of rebuilding.*
- *It is not necessary to record long serial numbers during the mounting of the equipment. They should however be printed on a label on the device to be able to program it out-of-the-box or if the Publisher ID is not known or faulty.*

The disadvantage of using the equipment number is that the long identifier lowers the efficiency (two more bytes). If this cannot be accepted, the 12-bit format must be used.

The railway identifier is based on the Litra numbers except that only the first Litra letter, which defines the main group, is used followed by the four digits, and it is compatible with the 12-bit MVB address so that each signal can be uniquely identified. In this way, the short identifier may be used to transmit local MVB signals, and this identifier may then be expanded to 31 bit in a gateway by adding the vehicle identifier so that the signals cannot be confused with signals from other vehicles when trainsets are connected.

2.5 Data

Max-i is entirely based on the so-called big-endian or "Motorola" model. In this model, the **most** significant byte or *word is located at the **lowest** address. For a 32-bit system it looks like this:*

LongWord, Addr. = 7			
Word, Addr. = 7			
Byte, Addr. = 8	Byte, Addr. = 9	Byte, Addr. = 10	Byte, Addr. = 11
			LongWord, Addr. = 7
	Word, Addr. = 5		Word, Addr. = 7
Byte, Addr. = 4	Byte, Addr. = 5	Byte, Addr. = 6	Byte, Addr. = 7
MS LongWord, Addr. = 0			
MS Word, Addr. = 0		Word, Addr. = 2	
MS Byte, Addr. = 0	Byte, Addr. = 1	Byte, Addr. = 2	Byte, Addr. = 3

Fig. 2.4

Note that all data are **left** shifted, so that the **most** significant bit (MSb) always has a fixed position no matter if the data length is byte, word or longword! **Therefore, the MSb is always called bit 0**. This is quite uncommon today except for the PowerPC architecture from IBM, NXP (previous Freescale) and STMicroelectronics, but it is the full and natural consequence of the big-endian model!

There are more important reasons for using the big-endian model:

- *It is natural to transmit the most significant bit of the most significant word first because the transmission then becomes a continuous stream of bits which may be divided into bytes, words or longwords as one pleases. This order of transmission is necessary for the bit-wise bus arbitration.*
- *The first bit of any telegram is bit 0 **no matter how long the telegram is**.*
- *The fixed part of many parameters and data with variable length becomes fixed bit numbers like the local bit of device identifiers.*
- *Left shifted data as 1.N two's-complement fractions and data for A/D and D/A converters can have a variable resolution, and can be truncated with just a slight loss of accuracy.*
- *It fits with microprocessors and A/D and D/A converters with SPI interface where the data is also left shifted, and the MSb is transmitted first. In this way, the same sequence and data can be used regardless of the resolution (number of bits).*
- *It is possible for a subscriber to accept telegrams with different resolutions for the same value and it is possible for a publisher to change the resolution without informing the subscribers.*

In principle, Max-i is able to transmit any number of data bits, but to simplify the communication with other devices, increase the safety and make it possible to receive and verify telegrams with an unknown length, all telegrams except for group polls, which only contain 2 bytes, shall have a length of an odd number of bytes (5, 7, 9 etc.) plus the start bit and the priority bit. Telegrams with a different length shall be rejected!

Max-i has no data length information, but because it uses synchronous communication, has 20 check bits and all telegrams with an unacceptable length are rejected; the data length may safely be taken from the telegram length. This saves 1-2 bytes.

The data length for values processed by the Max-i controller itself is 4 bit, 20 bit or 36 bit (N = 0, 1 or 2). This corresponds to the standard value-sizes word (16 bit) and long (32 bit) except that 4 more bits have been added.

8-bit data have too low resolution for most practical applications and 24-bit data are anyway not supported by most microprocessors so the lack of these data types is no big limitation in practice, but makes it much easier to receive telegrams of an unknown length.

The various data types are specified in layer 6b. Note that Boolean 4-bit values may be transmitter with or without supplementary 16-bit or 32-bit data like timestamp or group specification, and analog values may be transmitted with 20-bit or 36-bit resolution so that both lengths shall be accepted, but telegrams with any other length shall not be used by the controller.

Messages, which shall **not** be processed by the Max-i controller itself, but just transferred to other devices, can have a data length of any number of bytes (N × 8 bit) plus 4 bits, **but the telegram length must still be an odd number of bytes so addition of a dummy byte may be necessary**. Such messages may for example contain bit patterns, text strings, memory dumps, log files, downloads, 64-bit double precision floating point values, DMX512 data etc.

In these cases, the Max-i controller just works as a modem and the telegrams are transferred to/from the controller through a secondary serial channel – for example, an RS-232 or RS-485 interface. The controller shall by default perform automatic scrambling and de-scrambling of telegrams through this interface. However, it shall be possible to switch the scrambling off, so that the serial interface can also be supervised by the CRC polynomial if the connected device is able to perform the scrambling and de-scrambling.

It is in principle possible to transmit an infinite long telegram, which could saturate the bus and block all other communication. To prevent this, the maximum **data** length shall by default be limited to 1028 bytes (plus 4 bits). When Max-i is used to transfer data to more devices simultaneously in one telegram as explained later, this is enough even for 32-bit data to a device with a 1022 bytes (maximum) reception delay. It also leaves 32 bits free for offset etc. when for example big log files or software updates are transferred as more blocks of 1024 bytes. However, there may be applications where this limitation is undesirable for example in case of display systems or stage lamps. Therefore, it shall be possible to switch the watchdog off by means of a programming bit in the MODEM object as described in layer 7, but the default setting shall be "On".

If it is possible for a Max-i controller to run out of data for example in case of a serial interface without a sufficiently big FIFO buffer, each telegram between the controller and the connected device shall be separated with a telegram separator like a break condition (all bits including the stop bit equal to zero). The telegram shall be terminated when the FIFO runs empty – either because the telegram is finished (normal condition), the load of the FIFO is delayed, or if an error is detected like a framing error or an overrun condition. The telegram must not be restarted or a new telegram transmitted before the telegram separator has been received. This ensures that delayed data or data after an error condition are not regarded as the start of a new telegram.

2.5.1 Data Timeout

To prevent that no devices uses/displays old obsolete information, each Max-i controller or connected device, which consumes values, shall include two watchdog timers – one for implicit and explicit messages and one for group messages. The watchdog timers shall be restarted each time a valid message (no errors), which matches the identifier, is received, and if just one of the watchdogs runs out, the data shall be reset including bit 32 – 35 except for lamps used for lighting, where the data shall instead be temporary replaced by a programmed color.

If the timer runs out for analog values, which are displayed, the device should mark them as illegal for example by means of display flashing or another color, but the value itself shall not be changed or removed from the display. To be able to do this, the used object must have an output, which goes high if a watchdog runs out.

The safety in Max-i depends on this timeout system, and it is vital for safety systems. If a safety system or device has not received data for a given amount of time, it shall shut down the system. This mechanism also protects against delayed safety telegrams.

There are three ways to prevent the watchdog timer in the subscribers from running out.

- The data can be polled if the data has not been updated within a period, which must be shorter than the timeout time to allow some time for the answer. This is the preferred method for data consumed by SCADA systems because it makes it possible to adjust the poll speed according to the actual needs and only poll values, which are actually used for the moment. If more devices pool the same value, the poll speed will be set by the device with the fastest poll rate and the other devices will just utilize the polled values. Note that it may not take longer time to poll a value than to transmit it event driven.
- The data can be transmitted at regular time intervals in periods where there are no event driven transmissions – the so-called heartbeat.
- The watchdogs can be disabled. This is not allowed for safety systems.

The watchdog timers may be used to turn the outputs into pulse outputs for example in case of buzzers or to create stroboscopic lamps. In this case, the wanted pulse width is set by means of the watchdog timer, and the high timer resolution of 1 ms makes it fairly accurate and reproducible.

*For more reasons, a telegram-error shall **not** cause the outputs to be reset or a value to be marked as illegal:*

- *An error may cause an ID to be changed in such a way that a telegram intended for other devices may be erroneously received.*
- *A telegram may be aborted if the transmitting device is not able to supply the bytes fast enough for the synchronous communication.*
- *To be able to relax the error consequence by means of more telegrams within the timeout time.*

2.5.2 Heartbeat

Heartbeat is used to publish a value at a specified minimum rate.

⚠ **To reduce the system load, heartbeat of boolean values must only take place for active signals, that is, values, which are different from 0000B.** The watchdog timer shall therefore not run out if the consumed value is 0000B and shall therefore not transmit an error telegram in this case.

Heartbeat is the preferred method to transfer repetitive data for Max-i controllers because of two reasons:

- *It is much easier to implement in hardware than polling. Polling requires much more logic and two timers – one for poll and one for answer. Besides, the poll timer shall be increased in case of an unanswered poll as described in layer 5, which further complicates the logic.*
- *The system load may be reduced considerably for local Boolean values, which must be transmitted at a very high rate like safety signals or interlocking signals between conveyors. Because the heartbeat shall stop if the value is 0000B, passive devices do not load the bus, as they would do if the same signals were polled. When the heartbeat stops, one or more watchdog timers may run out, but the outputs have usually already been set low when the Boolean value changes to 0000B (event driven transmission).*

Heartbeat may also be used to repeat on-only telegrams at a fast rate for example to step a lamp dimmer up or down.

2.6 Message Types

Attribute 1 of an object is used to hold the main value to be published and attribute 2 is used to hold the consumed value to which the device subscribes. The other attributes are only transmitted during commissioning or in case of errors so to increase the efficiency, Max-i is able to skip the attribute specification for attribute 1 and in this way save two bytes. This is called an **implicit** message because it is implied that the value in the message originates from attribute 1 and shall be stored in attribute 2 if a device subscribes to it. A telegram, which includes the full 10-bit attribute number including the base address, which is necessary if more devices publish data with the same identifier, is called an **explicit** message. Note that attribute 1 may be read and attribute 2 may be written by means of both an implicit message and an explicit message, and that no read/write bit is necessary for implicit messages because they are always broadcast/read-only messages.

Invalid values for example caused by an input out-of-range or a too low supply voltage must **not** be transmitted in an implicit message, but it is allowed to poll the value by means of an explicit message. This is one of the reasons why an explicit message must never be regarded as containing valid process data as described later!

To be able to determine whether a message is implicit or explicit and to be able to distinguish between values to the network object and from an I/O object, the first telegram bits just after the priority bit are divided in four number ranges as shown below (yellow and white):

Bit Number	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17…
Network attr.	0	0	0	0	0	0	0	0	0	0	0	0	0	Attribute #				
Implicit ID	Local	0 . 1	0 . 1	0 . 1	0 . 1	0 . 1	0 . 0	0 . 0	0 . 1	0 . 1	0 . 1	0 . 1	1 . 1	1	1	1		
														ID extension (long)				
Group ID		1	1	1	1	1	0	1	8-bit group number									
Explicit attr. ID	Read	1	1	1	1	1	1	10-bit attribute number										ID
11-bit CAN-A	RTR	MSb	$\overline{\text{MSb}}$	Least significant 10 bits of 11-bit ID										$\overline{\text{IDE}}$	1	1		
29-bit CAN-B															18-bit identifier			

Fig. 2.5

The yellow part is the network object. These telegrams shall always be transmitted with maximum priority and therefore always have bit 0 = 0 and the priority bit set to 0. This is especially important for time telegrams, which then consist of a start-bit followed by 17 0-bits, but it may also be important for setting the speed and for future reset and/or reload telegrams.

This may in the future be used for automatic baud-rate detection, but it cannot be used for oscillator synchronization for two reasons:

- *Each bit may be extended with up to 1T = 33 % for all UART speeds up to 1 Mbps and up to 2T = 66 % for 2 Mbps and 4 Mbps.*
- *In case of a fast, high priority, global message (implicit or poll) with ID = 000,0000,0000,1…B, the time message will not win the bus arbitration before bit 12 and therefore consist of 13 bits from another device, which are up to 6 % too short, followed by 4 bits with correct length – in total up to 4.8 % too short.*

The white part is the I/O object. The I/O object may be addressed either as an individual object with a 12-bit or a 31-bit address or by means of an 8-bit group address.

The number range for 12-bit addressing is 0 – 4095, but only the range 1 – 3999 is used for practical process control. This makes it possible to use number 0 for the network object, the range 4000 – 4031 for group addressing and the range 4032 – 4095 for I/O attributes. Because of the three, long/short selection bits (bit 13 – 15), it is possible with 8 attributes in the network object even though only a single number (0) is used.

If more than 8 is needed in the future or two groups of network attributes are wanted, bit 0 may be used for that since it is not a part of the message type code.

The three types of messages are shown below:

Network message						
Start bit	Priority	0	Code	Address	Data	Signature
1 bit	1 bit	0	000,0000,0000,0	3 bit	4 bit boolean	20 bit
					36 bit other	

Implicit message and group message					
Start bit	Priority	Local	Identifier	Data	Signature
1 bit	1 bit	1 bit	15 or 31 bit for implicit 111,1101,xxxx,xxxx for group	4 bit boolean	20 bit
				N×8 + 4 bit after M×8 bit delay	

Explicit message								
Start bit	Priority	Read	Code	Attribute address		Identifier	Data	Signature
				Base	Number			
1 bit	1 bit	1 bit	111,111	6 bit	4 bit	15 or 31 bit for implicit 111,1101,xxxx,xxxx for group	36 bit or N×16 + 4 bit	20 bit

Fig. 2.6

Note that the build-up and length of network messages and implicit messages is the same if N = 0 or 4. This enables devices where the publisher or subscriber identifier is programmed to 000,0000,0000,0xxxB and the telegram type is the same as the network attribute to issue or receive network messages in the same way as normal implicit messages and group messages. This may for example be used for flash synchronization and time telegrams (in case of a device with an accurate clock). Because network messages and implicit messages use the same telegram type, the telegram type is called an implicit telegram in both cases.

Max-i utilizes the telegram length to distinguish between short 4-bit messages and other telegrams. It is therefore possible to address four different functions for each identifier:

- Short 4-bit implicit messages.

  These may for example be used to read and control Boolean process values and to turn a lamp on and off, step the light intensity up and down and select a few fixed values.

- Short 4-bit group messages.

  These may either be used to enable and disable Boolean outputs (GRPTYPE = 1) or for control of a light level (GRPTYPE = 0).

- Longer implicit messages to one or more devices in the same telegram.

  These may for example be used for process values and to set a specific color of one or more lamps.

- Longer group messages to one or more devices in the same telegram.

  These may for example be used to set a specific color for a group of lamps.

Note that the short and long functions shall always belong to the same device! It is for example **not** allowed to use the 4-bit boolean function for a motor starter and the long function to control the opening angle of a valve. This would also make it impossible to use a numbering or identification system like PNS, XML or MQTT where each device function has its own identifier.

2.6.1 Messages to more devices in the same telegram

To increase the efficiency for example for stage light and fast positioning systems in CNC-machines, robots and cobots, it shall be possible to transmit the same **type** of data, but not necessary the same number of bytes to more devices simultaneously in a common telegram. Here and in the following, a “message” is a message to a single device and a “telegram” is the sum of more messages.

The data length for each device shall be N × 8 bit, and the location of the data in the data field of the telegram shall be set by means of a 10-bit, DMX512 like byte delay specification for implicit messages and one for group messages so that it is possible to delay the reception 0 – 1022 bytes. In this case, the remaining 4-bit field just before the signature becomes common to all devices and therefore cannot contain individual data. Instead, it is used to select the speed of an (8-bit) filter for smoothing of light and inrush current limiting on a per-telegram basis and to select between different data types for the same device like lamp direction (pan and tilt), zoom, focus, control of gobo wheels etc. as specified later.

The reason for choosing a byte delay instead of a word delay, which would fits better with Max-i, is to enable 3-byte RGB data to lamps without any loss of efficiency and make the system compatible with DMX512. Note however, that the telegram shall always contain an odd number of bytes, so it may be necessary to add a dummy byte.

If a short 4-bit message is received, it will also be common to all subscribers.

This is used in many objects for All-off and Back-on and to change the light intensity by means of a common group message, but it may also be useful for more special applications like for example to implement interlace scanning on video walls where a short 4-bit off-telegram between update telegrams can switch an entire line off and in this way ensures that odd and even lines are not visible at the same time as described in Annex C – LED Drive.

The various values may overlap if only 8, 16 or 24 (+ 4) bits are needed. In this case, only the most significant bits plus the common 4 bits are valid as shown below with the light blue field for Value 4.

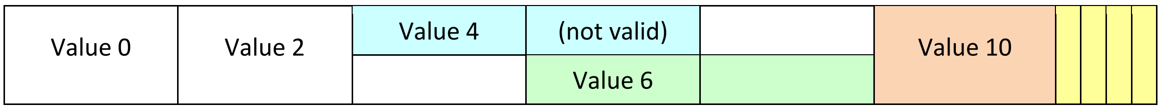


Fig. 2.7

Note that in case of messages longer than 20 bit, the data length cannot be derived from the telegram length except for the last value. **Therefore, all devices must have a data length specification.** For data used by the Max-i controller, there are only two possibilities – 16+4 bits or 32+4 bits, but connected devices like stage lamps may use many more bytes.

No matter the telegram length, Max-i **always** capture 32 data bits starting with the data field – plus the four common bits, which follow immediately after the 16-bit or 32-bit data field if the byte delay is 0. In this way, data is automatically left shifted according to the big endian model. This is particular useful in case of single FIX values where it makes it possible to accept data with both 20-bit and 36-bit resolution (same exponent) no matter if 20-bit or 36-bit data is wanted. A 20-bit message is readable in bit 0 – 19 and bit 20 – 31 then becomes the most significant 3 bytes of the signature and may therefore be regarded as part of a 36-bit value, but to avoid this, these bits shall be filled with 0’s as shown below if it is possible to determine the length of the message – either because the telegram ends or it is specified that only 20 bit is wanted.

36-bit consumed value (attribute 2)			
8 bit data	4 bit	Signature replaced by 0000,0000,0000,0000,0000	4 bit
16 bit data	4 bit	Signa. replaced by 0000,0000,0000	4 bit
24 bit data	4 bit	0000	4 bit
16 bit data	16 not valid bits		4 bit
32 bit data			4 bit

Fig. 2.8

The colors refer to the various messages shown in fig. 2.7. Since Max-i rejects telegrams with an odd number of bytes except for group polls, 8-bit and 24-bit data are only possible if the message is delayed an odd number of bytes.

The number of received bits is completely independent of the number of used bits. Due to the big-endian model, the data may be transferred to a connected device in **any** resolution (number of bits) starting with the MSb.

As it is the case for all data types, no data must be accepted unless the signature is verified and OK. In practice, this has the desirable side effect that all data in a common telegram are synchronized 100 % to the end of the telegram no matter if the data is located in the beginning or in the end of the telegram.

*For stage light control, Max-i may with advantage replace DMX512 and RDM, which can only transfer 512 bytes. Because of the **much** higher safety, Max-i may even be used to control pyrotechnics and laser lighting where performer or audience safety is at risk. Without special precautions, this is not allowed with DMX512, which has no error detection at all.*

DMX 512 has a specified maximum length of 1.2 km, but according to the RS-485 specification, on which it is based, it is only possible with a cable length of 850 m at 250 kbit/s and many manufacturers of DMX 512 equipment specify the maximum length to only 300 m or less for reliable operation. At that length, Max-i is only a little slower than DMX512, but for the reasons listed below, the efficiency is much higher, which at least compensates for the difference, and on short lines where Max-i may use a higher speed, it may be much faster.

- *Because there is no fixed telegram length, it is not necessary to send data to all devices/groups in every telegram. This makes it possible to update the lower groups more frequent than the others. This is in theory also possible with DMX 512, but since DMX 512 has no error detection, it needs to sends all data to all devices continuously so that an error is corrected fast by the next telegram and therefore may hardly be visible.*
- *Each identifier corresponds to one DMX512 universe, so Max-i can handle any number of universes (3999 or approximately 2 billion) on the same bus and update each universe separately.*
- *The LED controller is able to perform gamma correction and generate artificial amber and cyan, which makes it possible to use only 3 bytes for an RGBA value instead of 8 bytes (a linear 8-bit DMX 512 code is not enough).*
- *By means of the extra 4 bits, it is possible to select between lamp color with filter speed (36 bit) or up to 6 other 16 or 32 bits values (same width as color) like lamp direction (tilt and pan), beam width and focus, gobo rotation, special effects etc. It is therefore possible to send for example color data in some telegrams and direction or gobo data in others. In DMX512, all data must be transmitted in every telegram, which may cause a very low efficiency if some data are not transmitted very often or not need to be transmitted at the same time as other data. It is for example very seldom to change the gobo wheel while the light is on. If the system is fully utilized, up to 28 bytes may be received with only 4 "channels" where DMX512 needs all 28 channels and therefore would only be able to handle 18 devices with this amount of data and only 11 lamps if the flicker frequency shall be above 70 Hz.*

*It is often claimed that a big advantage of DMX512 is that all data are transmitted all the time so that a few errors are hardly detected by the audience. However, the error probability is typical in the order of $e^{-k(S/N)}$ so that if the transmission speed can be reduced to the half due to higher efficiency, the power in each pulse is doubled, the skin effect causes √2 times less signal attenuation and the noise bandwidth is decreased, so the S/N ratio is increased at least 3 times. The error probability is therefore put in the power of 3 so that for example an error probability of 0.1% is reduced to an error probability of 0.0000001%. **Efficiency is much better than speed!***

If the Max-i controller is used as a modem (MODEM object), the 10-bit byte delay may be used to exclude a number of data bytes from data telegrams (not attribute telegrams) if automatic scrambling is selected (default). In that case, the full identifier is transferred, but after that the data transfer shall be held until the specified byte where it is resumed. This is especially useful for stage light applications where it enables microprocessor controlled devices, which use the modem connection, to be programmed and behave exactly as devices, which only use the Max-i controller.

2.6.2 Local and Global Messages

The Local bit is used to divide the telegrams in two main groups – local and global – where local telegrams are intended for fast communication between a local group of devices, and global telegrams are telegrams, which may be used by any device and are also processed by SCADA systems, web interfaces, error handlers, gateways etc. A Max-i controller may be used either as a stand-alone I/O device or as a modem, which transfers telegrams on the Max-i network to more complex devices and SCADA systems over a secondary communication channel. If it is used as a modem, the local bit is used together with a 16-bit acceptance filter to determine whether a telegram shall be transmitted over the secondary channel. If the bit is 1 or the telegram is a network time message, the telegram shall be regarded as local communication and shall be blocked unless local telegram transfer is enabled in the MODEM object (described later). **The local bit has no influence on the subscriber or the I/O of the Max-i controller itself.** Devices with an **average** transmission rate higher than approximately one telegram per second should have the local bit set except for SCADA systems and other global control systems.

When a signal is polled, the local bit shall be returned in the answer. In this way, it is possible for a slow device like a SCADA system to poll the network time and samples from a very fast local communication such as a control loop without being overloaded – just by setting the local bit low in the acceptance filter (global telegram).

Because it shall be possible to control the local bit through a secondary communication channel, it is a part of the telegram and not an extra bit like the start bit and the priority bit.

Note that even though a signal is polled as global, all local subscribers shall still utilize it (local bit has no influence on the subscriber), so polling a fast signal in a control loop does not degenerate the loop performance. Also note that global telegrams have higher priority than local telegrams. This is important because global telegrams are usually transmitted from/to SCADA systems, web interfaces etc., which are used to control the process.

The local bit is used as a supplement to a traditional 16-bit CAN-like acceptance filter based on the identifier and some masks. There are three advantages of this:

- *It allows picking out individual signals even with a very big addressing range. The identifier-based filter is efficient for coarse filtering – especially in case of the short identifier where there is more freedom to group the various signals together, but it cannot select some signals from a device, but block other signals from the same device.*
- *It is possible to pick out samples from a fast data stream, which would overload the receiver (a SCADA system) if it were constantly enabled.*
- *It makes it possible to block long group messages, which are virtually impossible to block by means of the acceptance filter. Such messages are typically used to control more lamps or CNC machine axis simultaneously in the same telegram and are usually transmitted at a very high data rate.*

A typical automation system of today consists of a PLC, which controls the process, and a SCADA system, which handles the user interface. In such a system, the PLC will typical use local telegrams and the SCADA system will use global telegrams. In the automation system of the future, the PLC and SCADA system may be combined to one unit, which uses global telegrams. Local telegrams may then be used for high-speed inter-device communication such as control loops.

Network messages are always regarded as global, which also fits with bit 0 = 0. However, since time messages are used for repetitive transmission of time and/or flash synchronization, they may flush a debugger. They shall therefore be blocked as local messages unless the local block bit is set to 0 in the acceptance filter.

Telegrams containing explicit messages shall always be regarded as global, so they do not need the local bit and the following identifier doesn’t need it either. This makes it possible with a 10-bit attribute number, and it makes it possible to use the first bit as a Read bit, which is used to determine whether the attributes shall be polled or broadcasted (Read = 1), or written (Read = 0). The read bit is also used to solve the bus arbitration between a broadcast of an attribute and a write telegram to the same attribute.

All I/O attributes stored in NVRAM are programmed by transmitting an explicit message on the bus – **even if the explicit message is intended for the device itself!** This simplifies the controller and it has the further advantage that it is possible with a central logging of all attribute programming. If a device connected as a modem (see Modem object in Layer 7) wants to read-back the attribute, this must be done with a new explicit message (poll). This is necessary because there is only one serial output channel, and this channel shall always be ready for fieldbus telegrams and may therefore not be occupied by local communication to the controller.

If a value, which is transmitted by more publishers, shall be polled, only the device with base address 63 is allowed to answer the poll. For **all** the other devices, it is necessary to use an explicit message with the right base address for the published value (attribute 1) to avoid a collision with an unpredictable result, **but such an explicit message must never be regarded as containing valid process data. Only implicit messages contain valid process data!**

There are three reasons for this:

- *Another device may be the one, which has transmitted the last value, so the polled device may contain an obsolete value.*
- *Explicit messages may be used to change a process value temporary during commissioning – the so-called forcing. This is described later.*

- *Invalid values for example caused by an input out-of-range or a too low supply voltage must **not** be transmitted in an implicit message, but for debugging purpose it is allowed to poll the value by means of an explicit message.*

If more devices transmit the same value, there is a possibility for a collision between implicit messages. Although Max-i can handle bus arbitration in any part of the telegram, the final result after retransmission will be the value from the device with the lowest priority (delayed due to collision) and therefore not what would be expected, that is, the last updated value or the one with the highest priority. However, only one device should produce a value at a time. Besides, the time slot for collisions is very short – approximately 1 ms for a 500 m bus if the bus is busy and only 10 µS if the bus is idle, so even if more control panels are used simultaneously, the probability for an undesirable value is extremely small.

2.6.3 CAN and MVB Compatibility

Max-i is 100 % CAN compatible and can run any CAN protocol like DeviceNet and CANopen if the CAN ID and control bits are located as shown with the light blue background on fig. 2.5, and the number of data bytes are taken from the telegram length. Max-i may even run a CAN protocol simultaneously with the Max-i protocol. Note that the most significant bit of the CAN ID is duplicated so that 0 is converted to 01 and 1 is converted to 10 (inversion is shown as MSb). This preserves the most significant part of the mutual CAN priority (Max-i only has 64 priority levels), but ensures that the CAN identifier (11 or 29 bits) will never be confused with a network message, a group message or an explicit message.

Like the Local bit or Read bit, the CAN RTR bit is transferred to the receiver. It is then up to the receiver to send the requested reply. The RTR bit is only included for compatibility with some CAN protocols in case the fast poll system of Max-i (described later) is not utilized. If the bit is used, any device, which is used as a modem, must be programmed to accept both global (RTR = 0) and local (RTR = 1) telegrams.

The CAN IDE bit is inverted (shown as IDE) so that bit 13 becomes 1 for 11-bit CAN-A identifiers and 0 for 29-bit CAN-B identifiers. This is necessary to select between long and short identifier, but it has the side effect that CAN-B identifiers may become higher priority than CAN-A identifiers in case of the same 11-bit part depending on the scrambling. This is the opposite of CAN and because the RTR bit is placed before the identifier, which is also unlike CAN, the mutual priority between telegrams is not entirely preserved, but in Max-i, the main priority is not determined by the identifier, but by the Priority bit.

2.7 Polling

As described previously, a polling device just sends the first part of the telegram with the identifier. The device, which has the wanted data, may then complete the telegram by immediately adding the data part and signature.

In case of an implicit I/O message (**not** network message or group message), it may do so because there is usually only one device on the bus, which **transmits** a value (read-only) with a given identifier (many devices may poll it), so if this device **receives** this identifier, it must be a poll for this value or an error. If there are more devices, which publish the same value for example in case of more control panels, it is necessary to switch polling off to prevent that a value from one device causes all other devices with the same publisher ID to transmit their version of the value, which may even cause a "doorbell" with continuous transmissions at a very high rate. This is done by selecting a different base address than the default 111111B = 63, and polling must then be done in an **explicit** message for attribute 1 instead (see layer 6a), but it is anyway necessary to use the base address to set the attributes in a specific device in this case. Devices, which are programmed to generate network messages or group messages, shall not answer implicit polls at all, but can and shall still answer explicit polls as these have a different base address.

In case of an explicit message, which may be used for both read and write, it is necessary with a read-bit (R) to determine whether the device is allowed to add the data and the signature. If this bit is one, it is either a poll or a telegram from the device, which owns the attribute, so a poll may be answered. If the read bit is zero, it is instead a command for writing data to the attribute.

Although there is no error check on the arbitration field itself, the Hamming check on the identifier, which is included in the signature of the answer, ensures that if the wrong device answers the poll due to an error, the answer will not be accepted.

There are more important benefits of the fast poll system:

- *It usually does not take longer time to poll a value than to transmit it event driven. This saves a lot of time.*

- *It is not necessary with a question buffer if a device has more values or attributes to transmit. This simplifies the software.*
- *It makes it possible to exclude all other messages in for example a debugger than one's own where the transmitter is on. Without fast poll, the reply would be excluded too.*

Fast poll **by means of a group identifier** is of course impossible since this would return a mix of values from all devices in that group and there would be no valid publisher identifier. Instead, a poll on a group address shall trigger a transmission of the serial number (attribute 13) of all devices in that group. Unlike implicit messages, the poll itself shall not be extended with an answer, which could anyway only be one out of maybe many, but all answers shall be separate messages. This is used to distinguish between group polls, which therefore have only two bytes, and group messages/commands, which have more and may even be a common telegram to more devices.

*Group poll may be very useful to get a list of all devices in a group – or all devices on the bus if the group number is specified to 255 (unused group number). The list must be made by means of **explicit read** telegrams because implicit telegrams may cause action and may be impossible to distinguish from ordinary communication, and implicit telegrams from more devices may even contain the same identifier as the base address is not included. The serial numbers is used, because they may be used to address the devices without knowledge about the base address and any reception delay. This is very useful if the delay is not known as it may be the case for stage light.*

Note that the value ID and the group ID is not the same, so even though a device is programmed to generate group values by means of the value ID (111,1101,xxxx,xxxxB), it can also have a group ID and therefore answer group polls and respond to group messages – even its own.

The priority of a serial number telegram **triggered by means of a group poll** shall be set to low, and to reduce the probability of an overload of a common error handler, triggering of the telegram shall be delayed a random time from 0 – 0.25 seconds to approximately 0.5 – 0.75 seconds.

The bus arbitration of Max-i can easily handle a collision between any number of telegrams, but a typical SCADA system based on Windows or Linux can usually only handle in the order of 300 telegrams per second. With a random delay in the order of 0.5 s, a typical SCADA system can handle up to 150 devices without being overloaded and although more devices are possible if a list of all devices on the bus is wanted (group = 255), it is presumed that this time delay will usually be enough.

To be able to respond immediately, the published value or the attribute must be present in the Max-i controller. If the device for some reason is not able to reply immediately, it shall just let the telegram timeout and then sends the data in a full telegram later. This is faster than replying with a "data-later" telegram and there is no risk that such a telegram should be mistaken for the real data.

There are four situations where it is impossible to answer immediately:

- *If the Max-i controller just works as a modem for a device. This is usually the case for more complex devices with microprocessors. In this case, the published and consumed values and all user defined attributes are **not** located in the Max-i controller, but in the device itself.*
- *In case of a gateway/firewall. Note that due to the fully transparent gateway function (described in layer 3), which makes it possible to add or remove a gateway at any time without notifying the devices, a device shall never expect to receive an immediate answer!*
- *If the poll is done by means of a group address. In this case, more devices must answer the same poll (with a serial number message delayed 0 – 0.5 s).*
- *If the line is longer than possible for the bit wise bus arbitration. This is possible in Mode 2 (see chapter 1.7) if resistor termination and a master-slave protocol is used.*

In case of a fast poll, the local bit is of course returned in the answer, **but except for group messages, which shall return an explicit read message, which has no local bit, this shall also be the case if the device is not able to reply immediately!** Therefore, the local bit is a part of the telegram and not an extra bit like the start bit and the priority bit as described previously.

2.8 Scrambling and Signature

To be able to detect errors in the bit stream, Max-i uses a rather special and unique way of CRC-check invented by Innovatic and developed in collaboration with the Institute for Photonics, Coding and Visual Communication at the

Technical University of Denmark – DTU. Normally all data are just transmitted as they are and then terminated with one or more check bytes. The check bytes are actually known information – usually 0...0, which is scrambled with the CRC-generator-polynomial in the transmitter together with the rest of the telegram, and then de-scrambled in the receiver. If the known information is recognizable, the whole telegram is OK.

In Max-i, **all** bits are scrambled – not just the check bits – and the check bits are called the "signature", because the system works like a digital signature at the end of an encrypted message. If the signature is recognizable after the whole telegram has been decrypted, the message is OK.

There are many important benefits of this unique method:

- *It removes the necessity for data length information. With the traditional method, such information is necessary or else the receiver will not know when it should switch over and take the output data from the CRC generator instead of the data input. Data length information reduces the efficiency by increasing the telegram length and it is very questionable to use, because it must be used* ***before*** *it is verified by the CRC-check.*
- *The transceiver is highly simplified. It does not need to have any logic for switching between data and check-word and the only thing, which is needed for error detection, is three XOR gates and an increased length of the receiver shift register.*
- ***All*** *known information in the telegram such as the data type will contribute to the safety as this information works as extra signature bits.*
- *Any errors will affect the following bits so the rest of the telegram will be completely unreadable. This is especially useful for text where it for example makes it possible to strip the signature off before the data part of a telegram is transferred to a printer. It is much better with a "garbage" telegram than an erroneous telegram, which looks OK.*
- *The Max-i controller needs not to supply the known information, which is used as the signature. It is easily supplied by the used as any other data word.*
- *By using more different signatures, it is possible to transmit telegram type, value exponent, passwords, digital signatures and telegram serial numbers* ***without further data bytes*** *and in this way increase the efficiency a lot.*

Max-i uses a 20-bit signature, which is not only used for error detection, but also for holding the data type and any additional data type, exponent or serial number plus a Hamming code for the identifier. The signature shall be scrambled like any other bits.

The start bit and the priority bit shall **not** be included in the scrambling! These bits are added and removed by the Max-i controller and are not regarded as a part of the telegram.

Max-i uses the 20-bit CRC-polynomial:

$$G(x) = x^{20} + x^{14} + x^{13} + x^{12} + x^{10} + x^{7} + x^{5} + x^{3} + x + 1$$

This polynomial is especially selected for Max-i. It has a Hamming distance of 8 up to 56 bits (is able to find up to 7 errors), which is enough for all boolean telegrams (5 or 7 bytes), and a Hamming distance of 4 up to 152 bits, which is enough for all telegrams with 6, 16-bit data words plus 4 bits. It is also able to detect burst errors up to 20 bit and all odd number of errors, and it keeps the first 6 bits free of scrambling so the ID number range is divided in 64 priority levels, where the one with explicit messages becomes the lowest priority. Since the CRC calculation is initiated by the first "1" in the telegram, time telegrams (described later), which have an identifier consisting of pure 0's, will have higher priority than all other telegrams even though there are only 64 main priority levels.

Although the polynomial makes it possible to do error correction, this possibility is **not** utilized in Max-i as it may compromise the safety. See annex B for further information about this polynomial.

2.8.1 Signature

Max-i uses two basic types of signatures – one for write of data, which needs password protection, and one for all other telegrams as shown below:

	8 bit	4 bit	1 bit	7 bit
Implicit Boolean (read/broadcast) message	Any safety serial number	1100	1	ID Check
Implicit non-boolean non-safety (read/broadcast) message **Group message** **Explicit read message** **Time network message (write-only) to attribute in RAM**	FIX exponent	0000	Sign	
	Text code page	1011	8/16 bit	
	Any supplementary data type or radix point	Data type	Sign or 1	
Explicit write message **Network message (write-only) to attribute in RAM/NVRAM**	6-digit password (20 bit)			

Fig. 2.9

Note that an implicit message is the only message type, which shall update the telegram serial number if it is enabled, as it is also the only message, which is allowed to update the telegram watchdog!

The ID check of an explicit **read** message shall be the one for the publisher ID (attribute 4) no matter if the read command is issued by means of the Publisher ID or the Serial Number so in this case, the Hamming code is most likely not correct. An explicit **write** message shall use the password for the attribute to write as specified later.

In the following, the signature, which includes a password, is called the password signature, and the other signature is just called the signature.

A write to attributes, which may be stored in NVRAM, is always regarded as a restricted write as such attributes are used for setup parameters, which shall need a password to change. All writes by means of an explicit message is also considered a restricted write even though a few of the attributes may be located in RAM instead of NVRAM, so the only I/O attribute, which can be changed without a password is attribute 2 (consumed value – see layer 6a), which will be set when an implicit telegram (with attribute 1) is consumed.

The purpose of the optional serial number for Boolean safety values is to ensure that no telegrams are:

- lost
- appear repeatedly
- are inserted additionally, for example by hackers
- appear in the wrong order
- appear as another telegram (further safety against masquerading)

The receivers shall check that a Boolean safety telegram has the expected serial number. In this way, all 20 bits of the signature are known information as with a signature without counter.

Boolean group messages, which may be received by a group of devices with different subscriber ID and serial number, shall use the serial number 0000,0000B and shall **not** affect the serial number counter so that group messages does not destroy the sequence.

Note that if the receiver loses a telegram or a telegram is inserted by hackers, it will regain synchronization after 255 telegrams. The timeout time (watchdog timer) should therefore be shorter than the time it takes a hacker to transmit this number of messages so that the outputs get reset and the process stops if he tries to control a plant by inserting messages. This is the reason for using an 8-bit counter, which allows a reasonable long time.

The data types (4 + 1 bit) are defined in layer 6b.

The ID check is a 7-bit Hamming code based on the 15 or 31-bit identifier. This is true for all messages including group messages, where the first 7 bits of the identifier (111,1101B) are used to specify that it is a group message. The purpose of the Hamming code is to protect against masquerading, that is, a situation where the identifier is changed so that a device behaves like another one. The Hamming code ensures that no devices where the difference in identifiers is less than four bits can have the same signature. In practice, the remaining 13 bits increases the safety even further because they may make it impossible to compensate for three or more errors in the identifier by means of new errors in the data field, which is the only one, which is unknown.

To maintain the safety in case of an error on the identifier, the ID check shall never be generated automatically in the Max-i controller, but shall be calculated and loaded by a programming device!

Suppose there is an error on the attribute, which holds the publisher identifier in device 8 so that the identifier erroneously becomes 15 (3 bits difference). All devices with the subscriber identifier set to 15 will now receive implicit messages from device 8. However, these devices have the ID check in the subscriber signature set to the Hamming code for device 15, and because the difference in identifiers is only three bits, the received device-8 signature will not be the expected one, and the telegram will be rejected. In the same way, if there is an error on the attribute, which holds the subscriber identifier, the device will receive the wrong telegrams. However, if less than four bits of the subscriber identifier are wrong, the received telegrams will not have the expected signature and will therefore be rejected. The ID check also protects against masquerading or a receiver error in case of fast polls where the identifier is transmitted by one device and the rest of the telegram by another device. If the wrong device answers the poll, the ID check will prevent the answer from being accepted. For example, if device 9 answers a poll for value 8, the telegram identifier (first part of telegram) is 8, but the ID check in the answer is the Hamming code for ID 9 so the telegram will be rejected.

The selected Hamming code is shown below:

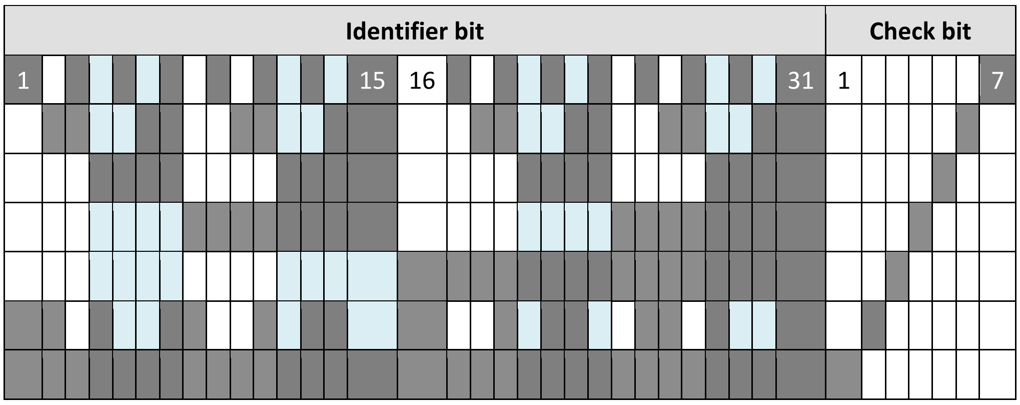


Fig. 2.10

All check bits use **even** parity, that is, the number of 1's including the check bit shall be even. This corresponds to a non-inverted XOR function so that if all identifier bits are either 0 or 1, all check bits shall be 0. In case of the short identifier (12 bit plus 3 1-bits), it is only necessary to use the first 15 bits. Even if bit 16 to 31 are filled with 0…0 or 1…1, the check bits will be the same. Because only 31 bits shall be checked, only the even Hamming codes (plus parity) have been used. The advantage of this is that in case of a one-bit error, the least significant 5 check bits points directly to the error position by means of the very simple formula:

ErrorBitNumber = (CheckWord XOR Expected-ID-check) AND 11111B

If for example there is an error on identifier bit 3, this will be indicated by check bit 1 (parity, which is not used in the formula) and check bit 6 and 7, which will have the opposite polarity of the expected ID check. When bit 3 to 7 of the Check-word and the expected ID check is XOR'ed together, the result will be the binary value 11B, which points directly to the error position (bit 3).

The publisher and subscriber identifiers are usually stored in NVRAM with typical data retention of 10 – 100 years. Therefore, it is extremely seldom that more than a single error occurs before the error is detected. Because the Hamming code is able to correct one error, it is therefore easy to find the faulty device with great probability. In this example, it is only necessary to invert bit 3 of the received identifier.

Note that the Max-i controller does **not** have to generate or check the Hamming codes except for network messages 0 and 4 – 7. The ID check is just supplied by the programming device like any other bits in the signature. The subscriber identifier works as an acceptance filter, which only passes telegrams through with the wanted identifier. If the signature is also the expected one, the ID check will always match the identifier so it is not necessary to verify the Hamming code and nothing would be gained by that. The only device, which needs to check the Hamming codes, is a common error handler, which must be able to receive explicit messages from all devices and therefore must be able to accept many different signatures. The data type for the error attribute is PATTERN (see layer 6b) and the ID check can be verified by comparing with the ID so all signature bits can be verified.

As defined in layer 6b, the data type for global attribute 0 is TIME (1100B), and the data type for global attribute 4 – 7 is PATTERN (1101B). The signatures for the global attributes therefore become:

Attribute 0:	*0000,0000, 1100,1 000,0000B*	*= 0 0C 80H*
Attribute 4:	*0000,0000, 1101,1 110,1101B*	*= 0 0D EDH*
Attribute 5:	*0000,0000, 1101,1 010,0010B*	*= 0 0D A2H*
Attribute 6:	*0000,0000, 1101,1 000,0011B*	*= 0 0D 83H*
Attribute 7:	*0000,0000, 1101,1 100,1100B*	*= 0 0D CCH*

Attribute 1 – 3 are used to hold the 3 passwords in NVRAM and therefore uses password 1 – 3 as signatures.

2.8.2 Password Signature

Telegrams, which are transmitted/broadcasted automatically from a Max-i controller, must have protection against masquerading as the publisher and subscriber identifier may be copied to rather unsafe RAM (only protected by a single parity bit) from NVRAM during power-up. Such telegrams must therefore be terminated with a signature with ID check. However, there is no risk of masquerading on telegrams, which are built in a high level control device such as a SCADA system and transmitted through a Max-i controller connected as a modem. If such telegrams are used to write data to attributes stored in NVRAM there is usually no need for a data type either, as almost all setup parameters have the data format PATTERN (no attributes stored in NVRAM contain process data, which may have a different data type). As no telegram counter is necessary on explicit messages, this leaves all 20 bits free for using the signature as a password so that it is only possible to change the content of an attribute if the password is known. The purpose of this is to ensure that any changes are only done by personal authorized to do so, and that no attribute is changed by accident. 20 bits makes it possible to make a 6-digit password, which is enough to protect the attributes if the user has to type the password (6 digits is used in many alarm systems). However, to prevent a password from being cracked or changed by scanners, a Max-i controller shall disable all further attribute writes to a device for approximately 5 minutes if an illegal password is received, and an error message shall be issued each time as described later. In this way, it will take an average of 5 years to crack a password.

Note that it is possible to poll the error word to find out when there is open for new attempts again and in this way avoid retriggering the timer.

Max-i uses three passwords:

Password 1 is the super user password. This password only allows access to attributes, which may be changed by trained users like process simulation (forcing) and change of hour counters.

Password 2 is the system integrator password. It is intended for all attributes, which are set during commissioning, like for example the identifiers, the signatures and the transmission speed.

Password 3 is the vendor password. It is intended for attributes, which may affect the measuring accuracy or validity of the data or affect the safety. It is used for attributes, which are programmed by the vendor like device and data type, gain and offset for measuring values, LED calibration factors, vendor ID, type number and serial number. In practice, there may be many different Password-3's in a plant if there are more vendors, but no vendor can change the attributes written by another vendor with a different password.

The passwords shall (normally) be typed in as 6 decimal digits, which shall then be converted to a 20-bit binary number and inverted (all 0's becomes 1 and vice versa). In this way, the default decimal password becomes 000000 since all un-programmed bits are 1. As the maximum 6-digit decimal number is 999999 = F423FH, which is inverted to 0BDC0H, the binary passwords 00000H – 0BDBFH are not possible to set in by means of a decimal keypad. This may be utilized to choose a password, which cannot be changed or used by ordinary users, but needs a special programming tool, which is able to program in a binary manner like the Max-i debugger (In Max-i, all user interface including numbering systems are decimal).

Max-i does not use any password hierarchy. No password can program an attribute, with is accessible with another password! This improves the safety and simplifies the Max-i controller, and it has the advantage that the user is forced to use the right password instead of just logging in as a vendor or system integrator. In this way, the higher-level passwords are not disclosed so often.

2.9 Telegram Speed

Because the 0- and 1-bits have different length, it is impossible to calculate a precise number of telegrams per second, but because of the scrambling, which generates approximately an equal number of 0's and 1's on a long telegram, an ***average*** *rate may be calculated.*

The priority bit is usually 0 so a telegram with a 31-bit identifier and a 4-bit or 20-bit value will therefore have the average length:

31-bit Identifier	
1 start-bit 29 0-bits + 28 1-bits	19T 339T
Total time for a 4-bit value	**358T**
8 0-bits + 8 1-bits	96T
Total time for a 20-bit value	**454T**

Fig. 2.11

If T = 6 µS (1 km line), Max-i is therefore able to transmit approximately 470 values of 4 bits or 370 values of 20 bits per second with the long identifier.

With a 12-bit identifier the average length becomes:

12-bit Identifier	
1 start-bit 21 0-bits + 20 1-bits	19T 243T
Total time for a 4-bit value	**262T**
8 0-bits + 8 1-bits	96T
Total time for an 20-bit value	**358T**

Fig. 2.12

If T = 6 µS, Max-i is therefore able to transmit approximately 640 values of 4 bits or 470 values of 20 bits per second with the short identifier.

Longer telegrams than 20 bits will benefit from the reduced 1-bit length during the last part.

Layer 3, Network Layer

This layer is used to define the transparent gateway function of Max-i and firewalls between safe and unsafe networks.

3.1 Gateways

A great advantage of the publisher/subscriber model is that there is no need for any address conversion during gateway functions. With other systems, it is necessary to have a stack of destination and source addresses and to perform a lot of source to destination and destination to source conversions. This is **not** necessary with Max-i.

The only thing the gateway should do is to setup an acceptance filter for the identifiers of the group of telegrams, which should be passes from the primary bus to the secondary bus. Because this will typical be global values, the local bit may be used in combination with the 16-bit acceptance filter for coarse filtering. Each time a telegram is transmitted to the secondary bus, the identifier should be saved so that the gateway knows that when the information becomes present on the secondary bus it is of interest for a device on the primary bus and shall therefore be passed to this bus. This mechanism may also be used as a firewall, which can block telegrams initiated from one side.

3.2 Firewalls

The safety model of Max-i is based on safe and unsafe networks connected by means of firewalls.

A wired network has the great advantage that it cannot be accessed unless you have physical access to the network or a wireless interface is connected. In this way, a network inside for example a house or a process plant is inherently safe and does not need encryption. If you are already inside, there are usually much easier and faster ways to control devices than to try to hack into a network and control the devices that way. In many cases, you just need to operate a wall switch or a control panel.

Security is only needed when an internal network is connected to the outside world. In this case, it should be done by means of a firewall so that telegrams from the internal network is allowed to be transmitted to the external network, and the reply is allowed to be transmitted back, but telegrams initiated from the outside network shall be blocked except perhaps for fire alarms, error messages etc., and any (other) data from this side must therefore be polled from the inside.

All devices and networks, which may be accessed unseen by the public, such as networks and devices in hotel rooms, exterior lamps and sensors, electronic lock panels etc. should either be driven from a controller on the inside, be tamper protected or be connected to any safe network through a firewall.

In some cases such as alarm systems and SCADA systems, it is desirable to be able to access a network over the internet from the outside. This may for example be done by initiating a telegram from the inside, which by means of an external server sets up a VPN tunnel between the network and the client. This requires strong encryption and passwords and is out of the scope of this specification.

Layer 4, Transport Layer

The purpose of this layer is to ensure end-to-end reliability by means of acknowledge telegrams and error recovery like retransmissions etc. In Max-i, this layer handles any automatic retransmissions in case of for example collisions or an unanswered poll.

4.1 Collisions

If a telegram is aborted during the bus arbitration, the transmitter shall retry as soon as the line gets idle.

A Max-i controller, which is used as a modem, shall be able to remember at least the first 13 bits of a telegram. In this way, it is always possible to retransmit a telegram with a short identifier because the last three bits are known. Collisions and errors after the 13th bit are extremely seldom so it is OK if the SCADA system should handle these. If a Max-i controller cannot retransmit a telegram or an error occurs, it is permissible that the transmission just stops without any further activity. The transport layer therefore cannot guarantee a full retransmission of a telegram.

4.2 Unanswered Poll

If a poll is not answered within a reasonable time, the device is allowed to try once more. If that poll is not answered either, the device shall wait at least 10 times longer before it tries to poll that value again. If there is still no reply, the poll rate shall be further reduced at least 10 times until there is a reply so that the poll rate for inactive devices are reduced at least a factor 100. Because of the heartbeat defined in layer 6a, a Max-i controller does not need to poll, so only high level devices need to implement this time increase.

4.3 Telegram Acknowledge

Because Max-i is a pure publisher/subscriber system, it has no acknowledge telegrams. It does not make sense to acknowledge a telegram, which can be received and utilized by more devices or no devices at all. If the published telegram is not received, but transmitted correctly, the transmitter anyway cannot do anything, even if there were an acknowledge telegram – except for logging the loss of telegram for later use when the communication channel (the bus) becomes functional again. However, due to the possibility for a telegram serial number (see Layer 2) the loss of a telegram may be detected the next time the data is polled or received.

Layer 5, Session Layer

The purpose of the Session Layer is to set up a communication channel between devices. As this is not necessary in Max-i, this layer is empty.

Layer 6a, Presentation Layer. Attributes

6a.1 Standard Attributes

To limit the necessary buffer size for temporary storage and simplify the Max-i controller, **all** attributes **within the Max-i controller** shall have a length of 36 bit, but in the present version, only the most significant 32 bits are used for values stored in NVRAM. The remaining 4 bits shall be filled with 1111B, 0000B or a ROM content defined during chip manufacturing or factory programming. Values stored in RAM use all 36 bits and in the future, this may also be the case for some NVRAM attributes. Any attributes **outside** the controller including a published value may have any length.

All vital 32-bit NVRAM attributes in the Max-i controller use the most significant bit (bit 0) as a parity check (error detection). This is especially important if the attributes are stored in a serial EEPROM and loaded to RAM during power up. The parity shall be even, that is, the number of 1's and the number of 0's shall always be an even number so that for example 00…00B and the default 11…11B do not cause a parity error. To save gates, a Max-i controller may use a serial parity checker, which only checks the parity when there is a telegram on the line (no continuous check). In case of a serious error, no implicit messages must be generated or accepted.

Attributes in the Max-i controller are read and written by means of a 36-bit explicit message, but Boolean global attributes in the network object (system time and group control) and a Boolean consumed value may also be written by means of a 4-bit implicit message. In this case, only bit 32 – 35 (least significant bits) shall be written, but if the consumed value has the type TIME, the local time shall be loaded in bit 0 – 31.

A Max-i controller may use NVRAM only to store the attributes, but it is recommended with a combination between NVRAM and RAM, which is always the case if the attributes are stored in for example a serial NVRAM and then loaded to RAM during power up. In this case, the NVRAM shall only be updated if the 36-bit data in the RAM is the same as the data in the write message **including any fixed bits** so that it is necessary to write the data twice to program the NVRAM.

This makes it possible to test a setup before the NVRAM is programmed, which is very convenient during development and production test (RAM write is much faster than NVRAM programming). If the setup doesn't work, the device can just be reset by removing the power temporary and is then back to normal. This is especially important with the speed setting in attribute 2 as a too high speed may otherwise disable all further programming.

Note that any fixed bits may be used for further write protection. If for example bits 32 – 35 are fixed to 1111B, as it is the case for many attributes, writing an attribute with bit 32 – 35 ≠ 1111B will only write to RAM, but not to NVRAM no matter how many times it is tried. Also note that bit 1 – 12 in the serial numbers are fixed during programming of the firmware version so it is also necessary to include these bits to change the serial number in NVRAM.

The only attributes, which shall always have a combination between NVRAM and RAM, are network attribute 2, which contains the speed (and password 2), and attribute 4, which contains the Value Identifier. In this way, the speed has to be written twice before it is stored permanently, which prevents a loss of control. It also makes it possible to assign a temporary identifier to a device, which may be convenient for plug-and-play and for rented equipment like stage lamps, which must have a known identifier for the next customer.

Attributes 0 – 3, **which are stored in RAM only,** and particular Attribute 1, which holds the published value, and attribute 2, which holds the consumed value, shall use all bit 0 – 35 as data bits (no parity and write protection) so that it is possible to use an explicit message to load an entire 36-bit value. This is for example used for analog values and data patterns and for stage light where bit 32 – 35 specifies whether the data contains a light value, which shall be stored in the 32-bit light register of attribute 2, or contains a 32-bit (mechanical) function, which shall be stored in the main register of attribute 2 together with bit 32 – 35, which specifies the type of function such as pan and tilt.

A simply Max-i controller implemented in a single IC may only have 16 x 36 = 576 bits of NVRAM to store attributes, which shall preserve their state after a power loss. For such a device, the minimum number of objects – one network object and one I/O object – shares the NVRAM as shown below:

Attribute No.	Rel. addr.	Content		Data Type	Write Password	Storage
0000000000000000B	0	System time (RTC)		TIME	Signature	RAM
0000000000000001B	1	Password 1		-	1	NVRAM
0000000000000010B	2	Password 2 and speed selection		-	2	NVRAM (or RAM)
0000000000000011B	3	Password 3 and default speed selection		-	3	NVRAM
0000000000000100B	4	(Free for future purpose)			Signature	RAM
0000000000000101B	5	(Free for future purpose)			Signature	RAM
0000000000000110B	6	(Free for future purpose)			Signature	RAM
0000000000000111B	7	(Free for future purpose)			Signature	RAM
111111xxxxxx0000B	0	Hour counter for subscriber		PATTERN	1	RAM / NVRAM
111111xxxxxx0001B	1	Published value		Publisher	1	RAM
111111xxxxxx0010B	2	Consumed value		Subscriber	1 or Signature	RAM
111111xxxxxx0011B	3	Error word		PATTERN	1	RAM
111111xxxxxx0100B	4	Publisher and attributes ID Main lamp publisher, subscriber and attributes ID		PATTERN	2	NVRAM or RAM
111111xxxxxx0101B	5	Publisher and attributes ID check Main lamp publisher, subscriber and attributes ID check	Publisher setup	PATTERN	2	NVRAM
111111xxxxxx0110B	6	Subscriber ID Aux. lamp publisher and subscriber ID		PATTERN	2	NVRAM
111111xxxxxx0111B	7	Subscriber ID check Aux. lamp publisher and subscriber ID check	Subscriber setup	PATTERN	2	NVRAM
111111xxxxxx1000B	8	Group ID, ID check and setup		PATTERN	2	NVRAM
111111xxxxxx1001B	9	I/O 1		PATTERN	2	NVRAM
111111xxxxxx1010B	10	I/O 2		PATTERN	3	NVRAM
111111xxxxxx1011B	11	I/O 3		PATTERN	3	NVRAM
111111xxxxxx1100B	12	I/O object and data types specification		PATTERN	3	NVRAM
111111xxxxxx1101B	13	Serial number and any Firmware Version		PATTERN	3	NVRAM + ROM
111111xxxxxx1110B	14	Vendor ID		PATTERN	3	NVRAM
111111xxxxxx1111B	15	Product Code and options		PATTERN	3	NVRAM

Fig. 6a.1

The yellow attributes are the Network object (see layer 2). The rest is the standard attributes of an I/O object. Note that unlike for example CIP, Max-i has no separate identity object as described in layer 2. Vendor ID, type number and serial number are a part of the I/O object. If attribute 13 – 15 (blue) is changed in one I/O objects of a device with more I/O objects, the change shall also affect attribute 13 – 15 in the other I/O objects. Any changes made to the network object will of course also affect all devices.

Because I/O attributes 0 – 3 are stored in RAM, these addresses may be reused for NVRAM addresses for the hour counters and the three passwords so that a 576 bit NVRAM is enough (hour counter, password 1, password 2, password 3, value identifier, publisher signature, subscriber identifier, subscriber signature etc.).

The data type for the published and the consumed value is defined in the I/O object specification (attribute 12). The other attributes are regarded as a string of bits and therefore have the data type PATTERN. The data type is only used in an attribute read telegram when an attribute is polled or is transmitted event driven (only the error attribute).

The consumed value may be written in two ways – by means of an explicit telegram with password 1 or by means of an implicit telegram with a signature. A read/broadcast telegram always uses the signature no matter if it is implicit or explicit.

No attributes must be programmed before the password in the programming telegram has been verified.

All attributes, which are stored in RAM-only, shall have the default 36-bit value 000000000H. Attributes, which are stored in non-volatile memory (NVRAM), shall have the default 32-bit or 36-bit value FFFFFFFF(F)H, and all unused (NU) bits shall be set to one (1). This corresponds to the de-facto un-programmed state of all types of NVRAM like EPROM, EEPROM and Flash memory. In this way, the default state of the internal attributes in the Max-i controller (1008 – 1023) becomes the same as any external attributes (0 – 1007) stored in NVRAM.

Usually, there is one publisher and one subscriber in a Max-i controller, but if the publisher data type is LAMPCTRL, the synchronized multi-way switching makes it necessary that the publisher ID and the subscriber ID are the same so they may share the same ID attribute and signature **including ID check**. This makes it possible to use a single Max-i controller to make wall switches similar to ordinary wall switches, which can control two lamps (or windows). In this case, the two publishers and the belonging subscribers use the attributes as shown with the light green fields above.

The LAMPCTRL data type is made in such a way that the lamp status can be determined without the need to be able to decode long messages and keep track of the lamp intensity. This makes it possible to make a simple subscriber for the auxiliary lamp, which is able to drive a simple on/off status signal.

The subscriber for the main lamp shall be able to receive both long and short messages and shall either be able to drive a full color lamp or a two-color lamp together with a simple on/off status signal similar to the one for the auxiliary lamp – see LAMP object in layer 7.

6a.2 NETWORK OBJECT

6a.2.1 System Time and Synchronization

The content of the system time attribute is equal to the two least significant 16-bit words of a TIMESTAMP and the two most significant Boolean bits are used for two-speed flash synchronization. In this way, a time telegram becomes exactly the same build-up as a Boolean value with timestamp. This makes it possible to issue time and/or flash synchronization telegrams from a Max-i controller and not just by a connected computer. To do this:

- *In attribute 4 (publisher ID):*
  - *Set the publisher identifier to the hexadecimal value 00 00 FF FF F (network message 0). This simultaneously makes it possible to address the device as unit 00 00 hex and use this to program the other attributes even though the device generates a network message.*
- *In attribute 5 (publisher ID check and setup):*
  - *Set the publisher signature (ID check) to 000,0000B (00 FF FF FF F).*
  - *Program the heartbeat timer to the wanted telegram repetition frequency – for example to 166 ms if you want 6 telegrams per second and therefore 3 Hz. Bit 0 will then automatically get the half, one third or one quarter frequency depending on the programming in attribute 9. You can also connect an external waveform generator to input 0 and 1 and transmit the telegrams as ordinary event driven Boolean values.*

  - *Set the XDATA bits 22 and 23 to 00B (00 tt tC FF F) or any of the other possibilities if you want the timestamp to be transmitted together with the Boolean flash bits as specified for attribute 5.*
- *In attribute 9 (Hardware setup 1):*
  - *Set bit 1 and 2 according to the divider specification (2, 3 or 4) in the application layer if automatic flash bit generation by means of the heartbeat timer is wanted.*
- *In attribute 12 (I/O object and data types specification):*
  - *Set the publisher data type to TIME (1100,0000,1B), which together with the ID-check generates the signature 0000,0000,1100,1000,0000B.*
  - *Set the priority bit (bit 1) to 0 if you want the telegram to be transmitted with high priority, which is recommended for time telegrams, but may not be necessary for flash bits only.*

To avoid that network time messages overflow the debugger, switch these messages off in the acceptance filter by means of bit 16 = 0 in attribute 9 (Hardware setup 1).

No matter the data length (4 or 36 bits) the flash bits shall always by updated even if they are generated by the device itself. **The time shall only be updated if the day is different from 0 and shall never be updated by a short telegram or by a telegram from the device itself!**

If the device does not contain a real time clock with battery backup, the attribute shall be reset to zero during power-up. This will set the day to zero, which indicates an illegal time. **The day shall stay at zero until a valid time (day ≠ 0) has been loaded, but the other fields shall be counted up!**

To ensure the maximum possible time accuracy, time telegrams with valid time shall always:

- Be transmitted with high priority, that is, with the priority bit set to 0 in attribute 12. In this way, time telegrams get higher priority than all other telegrams and will get through as soon as the bus gets idle.
- Use its own Max-i controller.

  *The reason for this is that a fixed high priority will affect **all** telegrams from that controller so that it must only be used for time telegrams and for emergency telegrams. If for example a SCADA system sends both time telegrams and normal telegrams, it shall be connected to the bus by means of at least two Max-i controllers – one for time telegrams and emergency stop and one for the rest. The use of a separate controller also guarantees that there are no waiting telegrams in queue, which must be transmitted before the time telegram.*
- Be transmitted by a dedicated device or by interrupt driven software with a guaranteed time delay less than 1 mS.
- Be transmitted by a crystal-controlled device, which should be synchronized to the GPS time.

A device, which transmits time telegrams, shall read any 4-bit flash synchronization telegrams from other devices and set the flash bits in the same way in its time telegrams. Bit 0 (32) shall follow the slowest flash frequency and bit 1 (33) shall follow the higher flash frequency. If only a single flash frequency is used, it shall be bit 0, and if the frequency of this is below 0.5 Hz, 1 Hz shall be used instead for slow flash and 3 Hz for fast flash.

The flash bits may be regarded as binary counting where the most significant bit (0) changes slower than the next most significant (1) and so on. Bit 2 and 3 may be used for other purposes, but if they are used for further frequencies, they should follow this scheme.

Because of the ≥17 0-bits in the beginning of a time telegram, time telegrams may also be used for oscillator synchronization and for automatic baud rate detection. For this reason, these telegrams shall only be transmitted from devices with a reasonable accurate clock – even if the day is 0 so that the time is not loaded.

6a.2.2 Passwords and Speed Selection

The three passwords have the following content:

Parity	NU	Password 1	NU		Fixed
1 bit	111	20 bit	1111	1111	1111

Parity	NU	Password 2	Hyst	FPoolEn	Boost	Speed	Fixed
1 bit	111	20 bit	1 bit	1 bit	2 bit	4 bit	1111

Parity	NU	Password 3	NU	Default speed	Fixed
1 bit	111	20 bit	1111	4 bit	1111

Fig. 6a.2

In a new Max-i controller, the three passwords attributes shall be set to FFFFFFFFH as any other attributes as shown for Password 1. This corresponds to the decimal password “0” (0 – 999999) as specified in chapter “Password Signature” in Layer 2 (all bits are inverted). The 20 password bits of Password 2 should not be changed simultaneously with the 4 speed bits as the speed will not be updated.

The **Hyst** bit is uses to select the receiver hysteresis according to the termination mode of the bus and must therefore be the same for all devices. The bit shall be left in the default 1 state if the ordinary voltage clams are used, but in case of resistor termination with or without line transformer, the bit shall be set to 0 to generate two threshold levels.

The **FPollEn** bit is used to disable the fast poll system. This may be necessary if resistor termination is used to increase the line length and/or the speed above what is possible with clamp termination. In that case, fast poll and more masters are not possible and a poll shall trigger an event driven transmission instead.

The **Boost** bits may be used to increase the transmitter (TX) and clamp level simultaneously (with the same voltage level) and in this way create a better signal integrity in case of a thin and/or long transmission line with high resistance. The levels depend on the actual controller implementation, but may be as follows:

Boost	**Level**	**Voltage increase on TX and Clamp**	**Minimum Supply Voltage**
00	High (high line resistance)	≈±3 V	≈23 V
01	Medium	≈±2 V	≈21 V
10	Low (low line resistance)	≈±1 V	≈19 V
11	No boost (default)	0	≈17 V

Fig. 6a.3

Use of the Boost bits requires that the supply voltage is high enough as shown in the table. If this is not the case, the transmitter and clamp levels shall just be as high as they can be. Even with no boost, transmitter saturation may take place below 17 V and therefore also at the minimum level 15.4 V.

The **Speed** bits are used to hold the speed selection as specified in chapter “Cable Length and Bus Speed” in Layer 1. If the speed setting in **both** password 2 and 3 is 1111B, the UART speed shall be 9.6 kbit/s, which can be handled by even a microprocessor without a hardware UART and hardware flow control, and this speed may then be used to program the device including the out-of-the-box speed in password 3. The speed setting in password 3 makes it possible for a device to work out-of-the-box if the intended application for the device has a standard speed like for example 250 kbit/s for smart-house and stage light applications. Because both 1111B and 1110B are used to select a speed of 1.2 kbit/s, password 3 can be set to 1110B, if an out-of-the-box speed of 1.2 kbit/s is wanted for example for long distance street lighting. As soon as the speed in password 2 is programmed to anything else than 1111B, this speed shall be used instead.

Since it is only possible to write to NVRAM if the data are the same as a previous write to RAM including any fixed bits, this procedure protects against a permanent loss of control due to a wrong speed setting. If the network does not work after a speed change, it is not possible to change the speed permanently and it is only necessary to remove the power temporary to restore the old speed. **Devices connected through the UART object shall change their speed in the same way!**

Password programming is done by means of a network message, so that all use of that password may be changed simultaneously. A password can only be changed if the password signature of the telegram is the same as the password, that is, with its own password. Because the passwords are write-only attributes, there is no way a password can be disclosed or overtaken.

In case of a detected parity error on a password, the default password 11…11B (000000 decimal) shall be used instead. If there is an error on password 2, the speed may or may not be wrong, but to prevent that a too low speed causes the line state to change due to an enabled additional (5 mA) line hold, the additional line hold shall be disabled in this case, but the speed shall not be changed as there is a good probability that it is correct.

6a.3 I/O OBJECT

6a.3.1 Attribute 0 – Hour Counter for main Subscriber

This attribute contains an hour-counter for motors etc. as shown below:

NU	Hour Counter	Minute Counter	Fixed
0000,0000	18 bit	6 bit	0000
0	8	26	32

Fig 6a.4

If BxC in the I/O 1 attribute is 1 (default state), the counter shall count while the corresponding Boolean output x is high. If B0C – B3C are 0, the counter shall stay at 0.

The resolution of the counter shall be at least 18 bit so that it is possible to count up to 262142 hours = 30 years (most hour counters counts up to 99999 hours = 11.4 years). To be able to measure the time with reasonable accuracy even in case of frequent starts and stops for example in case of batch production, the accuracy of the timer shall be at least 60 times better. Therefore, the timer shall also include a minute field. If an illegal minute time (60, 61, 62 or 63) has been loaded, which may be the case after power up if the attribute is loaded to the default 11…..11B, the time shall be counted forward until 0 no matter the state of the outputs and the BxC bits. In this way, this situation will only exist for a few seconds.

To prevent inaccuracies, the counter shall be clocked from the RTC and not just from the internal clock, so that time telegrams will also affect the hour counter. To prevent that minor time corrections of the RTC causes an extra minute or a lack of a minute, the clock to the minute part shall be taken from a hysteresis circuit – for example by using the two most significant bits of the second counter in the RTC.

If the supply voltage to a Max-i controller drops below the memory voltage so that a PUR pulse is generated (see layer 1), the controller shall save the hour counters to NVRAM if it has a back-up capacitor big enough to do so. Because it cannot be guaranteed that the hour counters can be stored successfully, a SCADA system should record the times from time to time – for example when it closes a conveyor chain down, so that the timers can be restored with a reasonable accuracy in case of a power failure.

Because there may be many subscribers where the connected motors are not equally old or changed simultaneously, it is necessary with an hour counter in each subscriber. It is not enough to record the hours at the publisher.

6a.3.2 Attribute 1 – Published Value

The published value attribute contains the value from the process or a temporary value set during commissioning. If the published value attribute is transmitted in an explicit message (polled), all 36 bits are transmitted, but if the value is transmitted in an implicit message, only some of the most significant bits (big-endian model) may be transmitted depending on the data type like for example the most significant 20 bits in case of a FIX20 value.

If a Max-i controller is used as a modem, the published value is not located in the controller, but in the connected device. The controller must therefore no longer issue implicit messages or answer polls for the published value, but it shall still be able to receive and transmit explicit messages (data type = UNDEF for published and consumed value). It is important to ensure that the used attributes in the controller and in the connected device do not overlap.

By means of an **explicit** write message it shall be possible to force an input temporary to a specified state or value. This may be very useful during commissioning for example to test alarm limits, in case of sensor errors or to be able to run the process without material. As soon as the value is changes, an **implicit** message with that value shall be transmitted.

If an input value is polled in an **implicit** message while it is forced to a temporary state, it shall be the forced value and not the actual process value, which is transmitted.

The forcing shall be terminated if an **explicit** poll of attribute 1 is done or after approximately 10 minutes, if the forcing is not refreshed with another explicit write. For Boolean values, the answer to this explicit poll shall be the actual process value as an acknowledgement that the forcing has been terminated, and an implicit message with the process value shall then be transmitted to inform all devices, which subscribes to the value. For analog values, this may not be possible as some conversion or transmission time may be necessary. In that case, it is allowed just to contain the old (forced) value in the explicit poll and not transmit any extra implicit message, as the process value will anyway be broadcasted later due to poll and/or heartbeat.

Explicit telegrams always use 36 bits, so if the data type is TIME, the time is set to bit 0 – 31 of the received telegram (not the local time). This may be utilized to set an illegal time (day = 0) and in this way indicate, that the value is not the actual process value.

It is possible to control two lamps from one Max-i controller, but there is only this single register for the published value. However, since the control consists of short 4-bit messages like fixed levels and up and down, there is not much use of the last command. Therefore, the published value shall follow the main lamp, that is, the lamp, which ID is specified in attribute 4 (the published ID), and the published value of the auxiliary lamp (ID in attribute 6) shall not be recorded.

6a.3.3 Attribute 2 – Consumed Value

This attribute holds the consumed value, which is received by the subscriber. It consists of four registers – a 36-bit main register, a 32-bit light level register and two 2-bit registers to hold the on/off and count direction status of the main lamp and the auxiliary lamp – as shown below:

36 bit main subscriber register				
8 bit	8 bit	8 bit	8 bit	4 bit
0	8	16	24	32

32 bit light level register				
Red	Green	Blue	White or Amber	
8 bit	8 bit	8 bit	6 bit up/down counter	2 bit
0	8	16	24	30

Main lamp status	
On/off	Direction
1 bit	1 bit

Aux. lamp status	
On/off	Direction
1 bit	1 bit

Fig 6a.5

The Auxiliary lamp status register shall only be updated if the publisher is programmed to LAMPCTRL and a message with the identifier of the auxiliary lamp is received. In **all** other cases, it is the two main registers and the light level register, which may be affected. In this way, a lamp with two controls only reacts to the main control, but may publish auxiliary lamp messages including group messages to **other** lamps.

Only the main subscriber register can be polled, but the content of the light level register is usually shown by means of the connected lamps.

The main register shall be set in these cases:

- If the subscriber data type is **not** LAMPCTRL and the GRPTYPE bit in attribute 12 is 1 (no lamp control).
- If the subscriber data type is **not** LAMPCTRL, the GRPTYPE bit is 0 and the message is either implicit or explicit, but not group.
- If the subscriber data type is LAMPCTRL, the message is long, bit 32 – 35 specifies a function and the message is either a group message if the GRPTYPE bit is 1 (message similar to implicit), an implicit message or an explicit message. In all three cases, the SPI interface shall be triggered (if selected), which transfers the function data to a connected microprocessor for further processing.

The light level register (but not bit 32 – 35 of the main register) shall be set in these cases.

- If the subscriber data type is LAMPCTRL and the message is short (4-bit). In this case, only the white channel shall be affected so that this command can be used for daylight control of an RGBW lamp or change of color temperature of an RGBA lamp (RGB not changed).
- If the subscriber data type is LAMPCTRL, GRPTYPE is 1, the message is long and bit 32 – 35 indicates light level plus smoothing time.
- If the GRPTYPE bit is 0 and a group message with the data type LAMPCTRL is received. In this case, the white channel is used to control the level of Output 0 – 3 in case of the BOOLEAN or LAMP object, or routed to Output 6 in case of other objects like KEYPAD, MODEM and SPI, which may use it to control a light level of for example a display.

The two status registers makes it possible to control two lamps with 4-bit control from one controller without an extra main register and light level register.

This saves a lot of gates, but it makes it necessary with a few, minor limitations. For example, a long message must always be regarded as an "on" command no matter if the data is 0000H, and it is only possible with one "analog" lamp dimmer per controller.

During power-up, all bits in the main register shall be reset. This shall also happen in case of a watchdog timeout – implicit or group – except for lamps for lighting where the color hue shall instead be (temporary) replaced with a specified color. This makes it easy to turn back to the previous setting without the need to switch all lamps on again.

The light level register shall also be reset during power up except that the white channel shall be set to FFH to ensure maximum light if it is a control lamp (CTIME = 1 and GRPTYPE = 0).

The 6 bits for the up/down counter has been selected as the best compromise between as few telegrams as possible to create an almost linear ramp and dimming in so small steps that it is not noticeable due to the smoothing filter, which adds approximately 4 extra steps between each source level and in this way creates all 255 steps between 0 and maximum on the basis of only 6 bits. The counter plus the following 2 bits makes it possible to program the light level directly by means of 8 bit from 20-bit or 36-bit messages and/or count it up and down and set it to 9 fixed levels by means of 4-bit messages as specified in the LAMPCTRL data type.

6a.3.4 Attribute 3 – Error Word

The error word is a read-only attribute, which should have the following content:

Bit	Error	Bit	Error
0	Error on external oscillator.	16	Transmitter overloaded or short circuited.
1	An input value (I/O attr. 1) out of range. Low limit.	17	An input value (I/O attr. 1) out of range. High limit.
2	Used outputs (I/O attr. 2) disconnected. No current.	18	An output (I/O attr. 2) overloaded. Output current too high.
3	Internal temperature too low.	19	Internal temperature too high.
4	Error on I/O attr. 4 – Publisher ID.	20	Error on Network attr. 1 – Password 1.
5	Error on I/O attr. 5 – Publisher signature.	21	Error on Network attr. 2 – Password 2.
6	Error on I/O attr. 6 – Subscriber ID .	22	Error on Network attr. 3 – Password 3.
7	Error on I/O attr. 7 – Subscriber signature.	23	Implicit watchdog timeout.
8	Error on I/O attr. 8 – Group.	24	Group watchdog timeout.
9	Error on I/O attr. 9 – Hardware register 1.	25	Watchdog timeout on any software.
10	Error on I/O attr. 10 – Hardware register 2.	26	Serial-in framing error.
11	Error on I/O attr. 11 – Hardware register 3.	27	Serial-in overrun error.
12	Error on I/O attr. 12 – I/O object spec.	28	Vendor defined.
13	Error on I/O attr. 13 – Serial number.	29	Vendor defined.
14	Internal supply voltage too low.	30	Vendor defined.
15	Wrong password warning (hacking).	31	Warning. Maximum line length exceeded.
		32	Error or status input 0
		33	Error or status input 1
		34	Error or status input 2
		35	Error or status input 3

Fig. 6a.6

In many cases, bits 0 – 15 are used for low limit errors and bits 16 – 31 for high limit errors. It is **not** a requirement that any of the shown error types are supported, but all unused bits shall be set to 0.

It is **not** allowed to transmit an error telegram or answer polls in case of an error on attribute 4 (Publisher ID) or on attribute 13 (Serial number) if it is used as identifier unless the error is detected by means of majority voting (3 registers in parallel) or in case of error correction so that the ID can still be considered valid.

It is not necessary to transmit an error message in case of an error on attribute 13. If it is used as identifier, it is anyway not allowed, and if it is not used as that, it is just uncritical information like the vendor ID and product code (attribute 14 and 15), which do not have any error detection (no parity bit).

The controller shall normally issue an error message each time any of the used bits go high, but **not** when they go low. There are however three cases where an error message shall **not** be generated:

- Implicit watchdog timeout for Boolean values with data type TIME. Since this may be used to generate pulses and any heartbeat of Boolean values stops if they are 0000B, this may be a normal situation, which shall only be reported as a status.
- Group watchdog timeout. This is because there may be many devices, which times out almost at the same time.
- Transmitter overload or short circuit. If the line is overloaded, this would cause oscillation.

External oscillator error 0 is also special. It shall only be activated if an external oscillator has been detected, but later fails. If there is no external oscillator, the error bit shall not be activated as this is a normal situation for all devices, which do not need a higher accuracy than ±6 %.

If there is an error on I/O attribute 5 or 12, the transmitted telegram may or may not be accepted, but this shall not block an attempt to transmit an error message even though an implicit message from that device must not be transmitted in case of such an error.

If an attempt is made to program a local attribute by means of a wrong password, an error message with bit 15 = 1 shall be issued and in the following approximately 5 minutes, **all** further attribute programming including programming of global attributes shall be disabled and any new attempt shall restart the timer and issue a new error message with bit 15 = 1 **even if the password is right!**

In this way, a hacker cannot use a missing error message or an OK status to tell that the password guess was right even though the attribute is not programmed due to the dead time, and testing at a high rate will just prolong the time. In this way, it will at least take an average of 5 years to crack one of the 6-digit passwords by brute force.

In case of global attributes, the same shall happen except that only devices, which have the additional 5 mA line hold circuit enabled, are allowed to transmit the error message to prevent that all devices on the network do it.

Password 1 and 2 are usually the same for all devices, but this may not be the case for password 3, which is used for factory settings.

In most I/O objects, input 4 – 7 are used for external error or status inputs (Error 32 – 35), which can trigger an error telegram.

For errors, which are not stationary (unlike attribute errors) and any error inputs, the high state of the bits shall be latched until the error telegram has been successfully transmitted, but no longer so that repetitive errors are reported and may be logged. This is especially important for unmanned plants where there may be nobody to acknowledge the errors. Besides, it simplifies the Max-i controller as the error attribute can be a read-only attribute and do not need error acknowledge or reset by means of an explicit message.

If the controller is used as a modem, the connected device must add its own error attribute or use the inputs as error inputs (see UART object in Layer 7).

*With the publisher/subscriber model, it is not possible to transmit error telegrams to a dedicated error handler. However, it is easy for an error handler to pick-up all error telegrams just by grapping all explicit messages by means of the acceptance filter. It may then throw any configuration telegrams away or log these together with errors. Since the signature of **all** attributes except for attribute 1 and 2 is PATTERN, it is easy to verify the error message no matter from which type of device it comes.*

6a.3.5 Attribute 4 – Main Value and Attribute Identifier (Publisher ID)

This attribute **always** holds the identifier of the main publisher and all attributes even in case of a network message generator. In case of the short identifier, only the **most** significant 12 bits of the ID field are used, and in case of group ID, only the most significant 16 bits are used. The rest of the identifier shall be filled with 1's as shown below:

Parity	000,0000,0000,0000B	1111,1111,1111,1111B	Fixed
0	15 bit time telegram ID	16 bit	1111
0	1	16	32

Parity	Short ID		1111,1111,1111,1111B	Fixed
1 bit	12 bit	111	16 bit	1111
0	1	13	16	32

Parity	111,1101B	Group #	1111,1111,1111,1111B	Fixed
1 bit	7 bit	8 bit	16 bit	1111
0	1	8	16	32

Fig. 6a.7

In case of the time telegram ID, the message priority shall be set to high and it is highly recommended to add an external oscillator with a much higher accuracy than the internal RC-oscillator in a Max-i chip.

The default identifier shall be FFFFFFFFH as any other NVRAM attributes.

There are three situations where the ID cannot or shall not be used:

- If the 6 MSb's of the ID (parity not included) is 1 (it is not necessary to check all 31 bits). In this case, the ID is illegal and shall be considered as not programmed.
- If the ID is not known.
- In case of a parity error.

In these cases, the serial number in attribute 13 may be used instead to program and read the attributes of the device including this by means of explicit messages.

To enable PnP, it should be possible to store a temporary ID in RAM without writing to NVRAM. For devices where the NVRAM is copied to RAM during PUR, this is always possible as described previously . However, it shall also be possible if nonvolatile attributes are only stored in NVRAM. **In this case, the ID in RAM shall only be used if the ID in NVRAM has the default value FFFFFFFFH.** If this is not the case, the ID in NVRAM shall be used so that the safety is the same as for the other nonvolatile attributes.

Attributes used directly from NVRAM are safer that attributes, which are held in RAM and copied to RAM during PUR, but it is impractical in many cases like testing.

If the publisher data type is LAMPCTRL, the identifier of the publisher and the subscriber need to be the same as described in the LAMPCTRL data type, so this register can and shall be used for both. This also saves the programming of the subscriber ID, which otherwise must be remembered to set equal to the publisher ID.

6a.3.6 Attribute 5 – Main Value and Attributes ID check and Publisher Setup

This attribute has the following content:

Parity	ID check	Heartbeat Timer			No count	Global	XDATA	Priority	Line Hold Invert	Object Base	Fixed
		x 100	x 10	ms							
1 bit	7 bit	1 bit	1 bit	10 bit	1 bit	1 bit	2 bit	1 bit	1 bit	6 bit	1111
0	1	8	9	10	20	21	22	24	25	26	32

Fig. 6a.8

As explained in layer 4, the **ID check** is a Hamming code of the identifier used to protect against masquerading. Note that the data type (bit 4 – 9 of the 3-byte signature) is **not** stored in this attribute, but in the I/O object specification attribute 12 so that all setting, which may affect the validity of the data, are protected by password 3.

Since the serial number does not have its own ID check, this ID check shall also be used if the serial number is used instead of the value ID.

The **Heartbeat timer** is used to transmit the value at regular time intervals. The time is counted in milliseconds (1 – 1023), but by means of two multipliers, it is possible to change the resolution to centiseconds, deciseconds and seconds. The total time range is therefore 1 ms to 17 minutes.

In case of a time telegram, the heartbeat timer shall be a free running oscillator, which generates a signal with a 50 % duty cycle for flash bit 33 (next most significant bit) of the data field. If for example 2 Hz is wanted for this bit, the timer must be set to 500 ms, and bit 32 (most significant bit) shall then be generated by means of a divider, which divides by either 2 or 3 to create 1 Hz or 2/3 Hz. The XDATA bits may then be used to select if the time shall also be transmitted.

In all other cases, the heartbeat timer shall be restarted each time the value is transmitted in an implicit message (not an explicit message) no matter the cause of the transmission – event driven, poll or heartbeat. If all 12 bits are 1, which shall be the default state, the heartbeat shall be disabled. The heartbeat shall also be disabled if the time is zero, that is, the least significant 10 bits of the time are 0. If the publisher data type is LAMPCTRL, the heartbeat timer is used to transmit up and down messages and is therefore fixed to 80 ms so the setting is not used.

If an internal A/D converter is used (FIX20), a new telegram shall not be transmitted immediately when the heartbeat timer runs out, but shall be delayed until a new value is ready (conversion in progress). In this way, the repetition rate will never be faster than the conversion rate **even if the timer is shorter** – for example 1 ms. Besides, the transmitted values will always be as updated as possible.

If heartbeat is enabled in a running system, the heartbeat time should be written before heartbeat is enabled to prevent that the first period becomes random and therefore may be very long (up to 17 minutes).

The **No-count** bit is used to disable the telegram serial number counter (described in layer 4).

The **Global** bit is used to determine whether a published telegram shall have its local bit set or reset. If the Global bit is 1, which shall be the default state, the Local bit of the telegram shall be 0. If the heartbeat timer is set to a fast repetition rate, the global bit should be set to 0.

The **XDATA** field is used to specify if supplementary 32-bit data – usually a timestamp – shhall be transmitted together with a Boolean value as shown below:

Code	Action
00	Always supplementary data
01	Supplementary data on global telegrams
10	Supplementary data on poll
11	No supplementary data (default)

Fig. 6a.9

Note that it is possible to select supplementary data on global telegrams or polled values only. In this way, fast local transmissions may be done without supplementary data, but the data is added if the value is polled as global.

It is not necessary to be able to select supplementary data on local telegrams only. If such data is really needed on local telegrams, it shall be included even if the value is polled as global because such telegrams are also used by the subscribers. In this case, "Always supplementary data" can therefore just be selected instead.

The **Priority** bit is used to reduce the priority level set in the I/O object specification attribute. If the bit is 1, which shall be the default state, the priority is not changed. If the bit is 0, the priority shall be reduced one step so that high priority is reduced to normal (automatic) priority, and normal priority is reduced to low priority.

The **Line Hold Invert** bit is used to invert the state of the Line Hold bit in the I/O object specification attribute, which is difficult to change because it is protected by password 3. A power supply shall have the line hold circuit default on and other devices shall have it off, but by means of this bit, which is only protected by password 2, is possible to invert the state for example in case of heavy noise or a very high number of power supplies. The Line Hold Invert bit is XOR'ed with the Line Hold bit. If both bits are 1, which shall be the default state, the result is 0 and the line hold circuit is disabled! The user should normally leave this bit in the default invert state (1).

The **Object Base** is used to hold the six most significant bits of the I/O attribute address. As specified in layer 2, the default object base shall be 111111B = 63. It is usually not necessary or recommended to change the object base unless 16 attributes are not enough or more devices shall produce the same value, where it is necessary to prevent that data from one device triggers a poll for the same data in other devices.

The object base shall be ignored if the serial number is used for addressing.

In case of lamps where the publisher and subscriber ID is the same for the main lamp and the auxiliary lamp, a command would always trigger a poll of the device itself, so poll of implicit messages shall be switched off. This also makes it possible to use the same object base including 63 for both lamps.

6a.3.7 Attribute 6 – Main Subscriber Identifier or Auxiliary Lamp Identifier

If the publisher data type is not LAMPCTRL, this attribute is used as an acceptance filter to select which value shall be consumed by the Max-i controller or communicated to a connected device for example by means of an SPI interface or an RS-232 connection. The subscriber identifier is similar to the value identifier:

Object with used subscriber			
Parity	Short ID	Long ID extension or 111,1111,1111,1111,1111B	Fixed
1 bit	12 bit	19 bit	1111
0	1	13	32

Fig. 6a.10

If the publisher data type is LAMPCTRL, this attribute is instead used to specify the combined publisher and subscriber ID for the auxiliary lamp. **All programming of attributes by means of explicit messages shall however still use the main ID in the Value ID register!**

6a.3.8 Attribute 7 – Subscriber or Auxiliary lamp ID check and Subscriber Setup

This attribute has the following content:

Parity	ID check	Watchdog Timer			No count	B20U	Delay (bytes)	Fixed
		x 100	x 10	ms				
1 bit	7 bit	1 bit	1 bit	10 bit	1 bit	1 bit	10 bit	1111
0	1	8	9	10	20	21	22	32

Fig. 6a.11

The **ID check** is a Hamming code defined in layer 4. As it is the case with the publisher signature, the data type (bit 4 – 9 of the signature) is **not** stored in this attribute, but in the I/O object specification attribute. If the publisher data type is **not** LAMPCTRL, the ID check shall be the ID check of the subscriber, but if it is LAMPCTRL, the ID check shall be the ID check of the auxiliary lamp.

The **Watchdog timer** shall be updated each time an implicit or explicit message is received, **but not in case of group messages**, which have their own watchdog! If the watchdog runs out, the consumed value in attribute 2 shall be reset. Like the heartbeat timer, the watchdog time is counted in milliseconds, but by means of the two multipliers, it is possible to change the resolution to centiseconds, deciseconds or seconds. The total time range is therefore 1 ms to 17 minutes. If all 12 bits are 1, which is the default state, or the least significant 10 bits are 0, the watchdog shall be disabled.

The high resolution makes it possible to use the watchdog timer to generate for example short pulses by means of the BOOLEAN object or stroboscopic light by means of the LAMP object. The output is set by means of a message, but when the timer runs out, the output is reset so no reset message is required and the pulse width becomes reproducible even on a busy line. This is for example very important to ensure a uniform light intensity for stroboscopic lamps. Since the time may be up to 1 count more than specified, the high resolution ensures low jitter for timers above approximately 10 ms (10 – 11 counts), but of course the accuracy cannot exceed the accuracy of the internal oscillator. ***Note however that if a watchdog is used to generate a pulse, the output cannot be set by means the opposite type of message (group message in case of implicit watchdog and visa versa), since the signal is in reset state until the watchdog is updated!***

Because a timeout can be a normal situation, which may also occur because the heartbeat of Boolean values stops if the value is 0000B, a timeout shall not cause an error message, but it shall be reported as a status bit in the error word.

The high accuracy and low jitter may also be important in case of connected train sets if the watchdog is used to stop the motors in case of a failure on the bus or the control device. If it for example is allowed to run the train for 1.8 seconds in case of an error, the watch dog timer may for example be updated twice per second and the motor timeout may then be set to 1.0 seconds. The brakes must then be activated maximum 0.8 seconds after that and at that time, no motors must still be running.

As it is the case with the publisher signature, the **No-count** bit is used to reset the telegram serial number counter.

The **Delay** bits are used to specify a delay **in bytes** (0 – 1023) for implicit messages (group messages have a similar delay specification). If the Delay bits are 3FFH (default) or 0, there shall be no delay. If the publisher data type is LAMPCTRL, which makes it possible to control two lamps from one Max-i controller, the specified delay shall only be used for the main lamp. Because any part of a telegram may be used as message, it is necessary that the receiver knows the length of the message, which shall be processed by a Max-i controller, because this information cannot always be derived from the telegram length. This is programmed in the **B20U** (user) bit. If B20U is the default 1, the expected data length is the same as specified by the vendor in the B20V (vendor) bit of I/O attribute 3 according to the I/O possibilities of the device. If B20U = 0, 36 bit messages are received even though in the device is only designed for 20 bit. This makes it for example possible for a dichromatic lamp to receive and utilize the blue and white (or amber) channel of a 36-bit message and use them for cold and warm white.

It is not considered necessary to be able to program a full-color lamp for 20-bit messages since very few will invest in full-color lamps if they don't want to be able to utilize their possibilities. This simplifies the circuit and programming as B20U can usually just be left in the default 1 state.

If GRPTYPE = 1, implicit messages and group messages works in the same way. If GRPTYPE = 0, group messages are used for light control and therefore always use 20-bit (or 4-bit) messages as only the least significant byte for the white channel is needed. The most significant byte is free for future purpose.

When the Max-i controller is used as a modem for example by means of the UART object, the reception delay is used to exclude a number of data bytes from data telegrams (not attribute telegrams) if the default automatic scrambling is selected (manual descrambling requires the full telegram). In that case, the full identifier is transferred, but after that the data transfer is held until the specified byte where it is resumed. If the specified delay is longer than the telegram, the last byte of the telegram with the ID check shall be transferred together with the Break condition, which is used to separate telegrams, so that the transferred ID and ID check makes up a "bracket" structure, which can be verified – see UART object.

6a.3.9 Attribute 8 – Group Register

Attribute 8 is used to specify the group number together with the ID check for the group as shown below:

Parity	ID check	Group ID	Timeout [ds]					DIS255	Delay (bytes)	Fixed
			× 100	× 10	× 3.2	× 1.8	× 1.33			
1 bit	7 bit	8 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	10 bit	1111
0	1	8	16	17	18	19	20	21	22	32

Fig. 6a.12

The **ID check** shall be based on the same Hamming code as for other implicit messages.

The data type for group messages shall either be the same as the subscriber or be LAMPCTRL. This is controlled by means of the GRPTYPE bit in the I/O Object specification attribute (12) as specified later.

The **Group ID** bits specify the group number/identifier. If the bits are the default 0FFH (255), reception of group values shall be switched off except for All-off and Back-on, but the device shall still respond to a group poll so that a poll for group 255 may be used to generate a list of all devices on the network.

The **Timeout** bits are used to program the watchdog timeout when group messages are used for example for emergency stop. The timeout shall only be enabled if all 5 timeout bits are different from the default 11111B, and it shall be updated each time a group message is received even if GRPTYPE = 0 as dummy group messages (no action) may be used to supervise the communication channel. The time is specified in deciseconds plus four multipliers, which makes it possible to set a time from 00000B = 0.1 second up to 11110B = 9.6 minutes in steps of 33 %. If the watchdog runs out, the consumed value in attribute 2 shall be reset except for lamps used for lighting, which shall instead change to a preprogrammed color.

The **DIS255** bit is used to enable (0) and disable (default 1) the All-off (and Back-on) function on group 255. The data type for group 255 shall always be LAMPCTRL no matter the data type of the subscriber and the ID-check shall be 06H (ID-check of 255) so that an entire All-off message becomes 7D FF 00 1D 86 Hex and Back-on becomes 7D FF F0 1D 86 Hex. In this way, All-off and Back-on may for example be used for a subscriber with the data type TIME and may be used with many different groups (many different ID-checks).

The **Delay** bits are used to specify the data length and the reception delay for group messages, which therefore not need to be the same as for implicit messages. As it is the case for implicit messages, the last byte of the telegram, which contains the ID check, and a Break condition shall be transferred if the specified delay is longer than the telegram.

There are two reasons why group messages have their own delay specification.

- *Many lamps may utilize the same group message so the number of bytes in a common group telegram for example for stage light may be much less than a corresponding common implicit telegram, which contains individual data for all lamps.*
- *It makes it possible to control individual lamps by means of implicit 20-bit or 36-bit messages **with a delay of 0**, but also control the same lamps by means of a common group telegram with individual delay for each lamp to set a scene or even control the lamps in real-time from for example a TV-set or a DMX512-like stage light controller.*

6a.3.10 Attribute 9-11 – I/O 1-3

These attributes are used for function/object programming. I/O 1 is protected by means of password 2 and is therefore accessible by normal programmers. I/O 2-3 are protected by password 3 and are used to hold data, which may affect the safety or measuring accuracy and therefore must only be programable by the manufacturer.

6a.3.11 Attribute 12 – I/O Object Specification

This attribute is used to specify the I/O object and set the data types as shown below:

Parity	Priority	Line Hold	NU	Publisher type			Object ID #	Subscriber type			GRPTYPE	Extra attr. blocks
				FIX Exponent		Main data Type		FIX Exponent		Main data Type		MMMMMM
				Data type	Additional data type			Data type	Additional data type			
1 bit	1 bit	1 bit	1	4 bit	4 bit	1 bit	7 bit	4 bit	4 bit	1 bit	1 bit	6 bit or fixed 00,0000
0	1	2	3	4	8	12	13	20	24	28	29	30 35

Fig. 6a.13

*The main data type bit is only used to control the expansion and **not** used in the expanded signature. If it is 0, the data type is FIX and therefore set to 0000B in the signature as shown below and bit 12 of the signature is instead used as a sign bit or to specify whether a text string uses 8-bit characters (ASCII) or 16-bit (UNICODE).*

The **Priority** bit is used together with the priority bit in the Publisher signature to select the priority of **all** telegrams according to this table:

Priority bit in:		**Total priority**
I/O object spec.	**Publisher Sig.**	
0	0	Normal (Automatic) priority
0	1	Always high priority
1	0	Always low priority
1	1	Normal (Automatic) priority (default)

Fig. 6a.14

If the bit is 1, which shall be the default state, the priority is set to **normal** unless it is reduced to **low** by the priority bit in the publisher signature. If the bit is 0, the priority is set to **high** unless it is reduced to **normal** by the priority bit in the publisher signature. In both cases, the master priority is set in the I/O object specification attribute, which is protected by password 3. This priority can then be reduced one, but only one step by means of the publisher signature, which is only protected by password 2, but it cannot be increased. **Note that setting the priority bit to 0 in the I/O object specification attribute is only allowed for controllers dedicated to time telegrams and/or emergency telegrams! These devices are not allowed to transmit ordinary telegrams!**

The **Line Hold** bit is used to enable and disable the line hold circuit. The bit is XOR'ed with the Line Hold Invert bit in the subscriber signature attribute. If both bits are 1, which shall be the default state, the result is 0 and the line hold circuit shall be switched off. If the device is used as a power supply, the default line hold state shall be switched on by programming this bit to 0.

The **Publisher type** defines the data type for the inputs, and the **Subscriber type** defines the data type for the outputs. The format is a compressed version of the first 12 bits of the signature and must be expanded from 9 to 12 bits depending on the main data type as specified below:

Main data type	12 bit expansion		
0 (FIX)	8 bit FIX exponent		0000
1 (all others)	0000	4-bit additional data type	4-bit data type

Fig. 6a.15

To ease the expansion, the order of the data type and the additional data type is reversed in attribute 12 compared to the signature. In this way, the additional data type or least significant nibble of the exponent always has the same position.

If the subscriber data type is LAMPCTRL, which is the case for lamps used for lighting, the auxiliary light level register in attribute 2 shall replace bit 24 – 31 of the main register.

The **GRPTYPE** bit is used to specify the data type of group messages.

If GRPTYPE = 1 (default), group messages are used exactly the same way as implicit messages and has the same data type and the same data length. The auxiliary light register shall be loaded together with bit 0 – 31 of the main register, and bit 32 – 35 of the main register shall always be loaded no matter the telegram length except for the data type TIME, which has OR/AND logic between the 4-bit main data and the 4-bit group data as specified in the TIME data type. In this case, only the 4 group latches shall be updated in case of a group message.

If GRPTYPE = 0, group messages are instead used to control the light intensity of control lamps, traffic lights and displays and shall therefore use the data type LAMPCTRL **even though the subscriber data type is different!** For example, the BOOLEAN object has 4 output channels, which can drive dimmable lamps directly for example for traffic lights and control lamps, but other objects like MODEM, SPI and KEYPAD have an auxiliary light level output (Output 6) to control the level of a display. When GRPTYPE = 0, group messages shall only load the auxiliary light register and because only a single byte is needed, the data length is always 20 bit (or 4 bit). 20-bit messages set the light level directly as an 8-bit value, and 4-bit group messages use the control codes of the LAMPCTRL data type to set and change the light level.

This makes it for example possible to control the light intensity of a display or a control lamp, which is connected by means of the MODEM, SPI, or BOOLEAN object, ***even though the data type of the display or lamp data (implicit messages) is for example FIXBCD, STRING, PATTERN or TIME.*** *It also makes it possible for different device types to utilize the same group messages. A control panel may for example include both control lamps, which have the subscriber data type TIME, and displays, which use the data type of the value to be displayed. If for example the MODEM object and the SPI object could not switch to LAMPCTRL for group messages, it would not be possible to dim the entire panel by means of the same massages.*

The **Object ID** consists of a 7-bit unsigned integer object number, which is used to select the function of the Max-i controller. Objects defined by the manufacturer start from 0 and use increasing numbers and standard objects in the Max-i controller start from 127 and use decreasing numbers so that the default number 111,1111B can be used for the default MODEM object.

Since there are not expected to be more than 27 standard objects in the future, the range 0 – 99 can be regarded as free for manufacturer objects.

The least significant 6 bits are used to specify the number of attributes used by the I/O object including any connected device except for the 16 standard attributes. It is specified in blocks of 16 attributes. If the Max-i controller only uses 32-bit attributes, these 6 bits shall be set to 00,0000B since all standard I/O objects in the Max-i controller so far only uses one block and therefor no extra blocks. If 36-bit attributes are used, the number shall be set to the number of blocks any connected device uses so that it is possible to share the base address range between devices.

If for example a mass flow gauge needs 100 setup parameters and therefore 7 extra blocks, the base address of the next device, which generates messages with the same ID, must be at least 8. Without this specification, it would be necessary to get the data from a data sheet.

Because the default base address is 111111B, all devices, which need more blocks than 1, will usually cause the block address to wrap around so that the first block starts at 63 and the second block at 0 etc.

6a.3.12 Attribute 13 – Serial Number

The interpretation of the serial number depends on the development state of the controller as shown below:

Totally un-programmed device with fixed firmware (ASIC)						
Parity	0	Default running number	Serial number code	Standard IC production date specification		
				Year	Week	Location
1	0	11,1111,1111,1	000,00	11,1111,1	111,111	1,1111
0	1	2	13	18	25	31 35

Controller with fixed firmware (ASIC)						
Parity	0	Least significant 11 bits of number of devices produced this week on this location.	Serial number code	Standard IC production date specification		
				Year	Week	Location Chr5 letter
1 bit	0	11 bit	000,00	7 bit	6 bit	5 bit
0	1	2	13	18	25	31 35

Controller with programmable firmware (FPGA) and 32 or 36 bit storage						
Parity bit 0 to 17	Major revision Chr5 letter	Minor revision and test versions	Serial number Code	Parity bit 18 to 35	Running number	Fixed
1 bit	5 bit	7 bit	000,00	1 bit	13 bit	1111
0	1	6	13	18	19	32

Fig. 6a.16

FPGA devices, which may store the parameters in a serial EEPROM, may use either 32-bit or 36 bit parameters, but ASIC devices, which uses direct parallel parameter storage, shall use 36-bit parameters.

To avoid confusions with other long identifiers, bit 13 – 17 of the serial number shall be fixed 000,00B as shown and specified in Layer 2. Bit 1 = 0 for ASIC's is used to ensure that the serial number is never confused with an attribute or group and that the parity of a totally un-programmed device is OK as bit 13 – 17 is 0.

Most semiconductor manufacturers use a production date specification consisting of a 2-digit specification of the production year and a 2-digit specification of the production week plus one letter indicating the assembly location. This standard is also used by Max-i for devices with fixed firmware (ASIC). Some IC's also have a following 3-digit (10 bit) running number and this is easily contained in the 11 bit of Max-i. In case of a 32-bit attribute, there is no space for the *assembly location unless it is directly set by the IC masks and it shall therefore use the fixed code 11111B, which is an illegal character. In this way, it is easy to distinguish between 32-bit and 36-bit data.*

Note that when the serial number is used as ID, only bit 1 – 31 is used! Therefore, the least significant 5 bit is used for production location, which is not so important.

For test purpose and until an ASIC is ready, an FPGA-based controller with programmable firmware may be used. In this case, 5 bits are used to specify major firmware revisions as a letter so that it is never confused with a group or attribute when it is used as an identifier (11111B never used) and 7 bits = two digits are used to specify minor revisions and test versions. These bits are set permanently by the firmware and are therefore read-only information (ROM).

Major revisions are those, which may influence the operation of devices designed for a previous version and/or may influence the approval of safety or legally approved devices. Minor revisions do not influence older devices. They may be used to add new features and/or new I/O objects and for development/test versions where it is accepted that a (test) system may contain devices designed for many different versions of the specification and may therefore not be 100 % compatible.

To ensure that a device with the default running number 1,1111,1111,1111B does not have a parity error, bit 0 is used as a parity bit for the fixed part, and there is a separate parity bit for the running number. In this way, programming of the running number **in RAM** may easily be done by means of the 32-bit number xxxx,xxxx,xxxx,xxxx,xx Pa,bcde,fghi,jklm where all unused bits are set to 1 or 0 and P ensures even parity of the running number without taking the fixed bits into consideration. **If the number shall however also be programmed in NVRAM, it is necessary to include all fixed bits including the major and minor revision with parity and the last 1111B.**

A year and week code is not considered necessary as the release date of the firmware will usually be available.

The running number uses 13 bit, which is enough for 8191 devices, but so many devices with an FPGA and unchanged firmware will probably never be produced as an ASIC is more economical before that.

The serial number may be used in the same way as the publisher identifier in attribute 4 to poll the main value and to read and write the attributes. Since the serial number does not have its own data type and ID check, it shall just use the ones from the main value although the Hamming code does not fit. Because there is not a separate subscriber serial number, it is not possible to use the serial number to broadcast data to one or more devices in an implicit message as this would be regarded as a poll for the published value. To send data to a device, it is therefore necessary to do it by means of an explicit write to attribute 2 (consumed value).

The serial numbers may for example be used to identify one out of many stage lamps if the reception delay (DMX address) is not known and many lamps use the same identifier (DMX universe). Since an explicit write writes directly to the subscriber register (attribute 2), it ignores any reception delay. Once identified, the type and serial number may be read so that a data sheet may be shown and the lamp may be reprogrammed. This saves the traditional display and buttons for setting the DMX address. All you need to do is to poll group 255 or a specific group to get a list of device serial numbers, identify the various lamps one by one by switching on the light and reprogram if necessary.

6a.3.13 Attribute 14 – Vendor ID

Attributes 14 are used for vendor ID as shown below:

Controller with programmable firmware (FPGA) and 32-bit data storage			
Parity	6 Chr5 characters AAAAAA	NU	Fixed
1 bit	6 x 5 bit = 30 bit	1	1111
0	1	31	32

Controller with fixed firmware (ASIC) and 36-bit data storage	
Parity	7 Chr5 characters AAAAAAA
1 bit	7 x 5 bit = 35 bit
0	1

Fig. 6a.17

Because bit 31 – 35 = 11111B is an illegal character, it is easy to distinguish between the two formats.

The vendor ID shall as far as possible be a shortening of the company name like INOVTC (32 bit) or INOVATC (36 bit) for Innovatic. The first and last letter should be identical to the first and last letter of the company name as this is very important for the readability. The vendor may suggest an ID, but it is assigned by the Max-i Association, which ensures that no vendor ID's are confusable.

6a.3.14 Attribute 15 – Product Code, Hardware Version and Options

The product code is very similar to the Vendor ID. It consists of up to three Chr5 characters for the product type followed by a 4½-digit product number plus one optional Chr5 character in case of 36-bit write, which may be used to specify hardware revisions/versions and/or included options.

Controller with programmable firmware (FPGA) and 32 bit data storage				
Parity	AAA	nNNNN (maximum 32767)	NU	Fixed
1 bit	3 x 5 bit = 15 bit	15 bit	1	1111
0	1	16	31	32

Controller with fixed firmware (ASIC) and 36 bit data storage			
Parity	AAA	nNNNN (maximum 32767)	A
1 bit	3 x 5 bit = 15 bit	15 bit	5 bit
0	1	16	31

Fig. 6a.18

Because bit 31 – 35 = 11111B is an illegal character, it is easy to distinguish between the two formats.

When the product code is converted to ASCII with leading zero suppression on the product number, it shall be useable as a file name for an electronic data sheet in PDF (human readable) and/or XML (machine readable) format like for example LCR3000H.PDF or M361.XML, and the vendor of a Max-i device should supply such a data sheet.

Such a data sheet can contain much more information than it is possible to store in a simple controller like for example pictures, wiring diagrams, description and interpretation of attributes etc., and it makes it of course superfluous with a device type as it appears from the data sheet. Therefore, Max-i has taken the consequence only to store the "file name" instead of using many storage cells for storing some data, but not all. The Plant Numbering System gives all functions a standard name, and the contents and interpretation of standard and user-defined attributes appears from the data sheet, so it is not necessary with standard device profiles etc. If more information is wanted in the device, it is possible to expand *the description by means of one or more extra blocks of attributes (this is the reason why the vendor ID and production code is the last attributes in the standard block).*

In practice, all vendors have their own device numbering system, but the used product code and hardware version/option format (AAANNNNA) fits fairly well with the majority of them and at least much better than just a plain number as it is the case for identification objects of the majority of other fieldbus systems.

The standard attributes have no attribute for software version of a connected device in case the Max-i controller is programmed as a modem. This information shall be contained in a user-defined attribute in the connected device itself so that it can be guaranteed that the software version always corresponds to the loaded software, as it is the case with the firmware version (major and minor revision) of a programmable Max-i controller.

Below is shown a comparison with other common device identification systems:

#	Attribute	Max-i	IEEE 1451.4	DeviceNet	CANOpen	Ethernet MAC
14	Vendor ID	30 or 35 (text string)	14 (number)	16 (number)	8 + 24 (number)	22 (number)
15	Product Code	30 or 35 (alphanumeric)	15 (number)	16 (number)	32 (number)	0
13	Major Revision	5 (1 letter)	5 (1 letter)	8	16	0
	Minor Revision	7 (2 digits)	6	8	16	0
	Serial Number	24 (ASIC) or 25 (FPGA)	24	32	32	24

Fig. 6a.19

Layer 6b, Presentation Layer. Data Types

The purpose of this layer is to ensure correct data interpretation. It describes the standard format for analog and boolean values and the content of the two object types. It also describes the action to take in case of obsolete data (data timeout).

6b.1 Data Types

Max-i uses the following data types:

Name	**Exponent/Radix**		**Code**	**Type**	**Sign**	**Data Length**	**Description**
UFIX	11111111 – 01111111		0000	←0	0	20 or 36	Unsigned 0.N fractional number
	00010100 00100100					20 36	Unsigned integer
SFIX	11111111 – 01111111				1	20 or 36	Signed 1.N two's-complement fraction
	00010011 00100011					1+19 1+35	Signed integer

UFIXBCD	0000	0000
		0001
		0010
		0011
		0100
		0101
		0110
		0111
		:
		1111
			0001	1	0
					MSD:
					0 – 9
						≥20
						≥20
						≥20
						≥20
						≥20
						≥20
						36
						36
							0.99999(999999..) or ±0.9999(999999..)
							9.9999(999999..) or ±1.9999(999999..)
							99.999(999999..) or ±19.999(999999..)
							999.99(999999..) or ±199.99(999999..)
							9999.9(999999..) or ±1999.9(999999..)
							99999.(999999..) or ±19999.(999999..)
							999999.999(99..) or ±199999.999(99..)
							9999999.99(99..) or ±1999999.99(99..)

SFIXBCD					1 MSD: ±0 ±1		
	0000	**Code**					
UBCD	0000	0000	0010	1	0	20 or 36	Unsigned BCD floating point format
SBCD					1		Signed BCD floating point format
FLOAT	0000	0000	0011	1	1	32 or 64 + 4	32 or 64-bit IEEE 754 floating-point format
					Sign or info		
STRING8	Code page PPPPPPPP		1011	1	0	N×8 + 4	8-bit or 16-bit text string + data selection
STRING16					1	N×16 + 4	
TIME	Telegram serial number or 00000000		1100	1	1	32 + 4	32-bits TIMESTAMP + Boolean
						24 + 8 + 4	24-bits TIMESTAMP + 8-bit lamp data with flash + 0000B
						32 + 4 4	System time + flash synchronization Flash synchronization
PATTERN	0000	0000	1101	1	1	20 or 36 16 or 32 + 4 36	Patterns, bytes and/or words Patterns, bytes and/or words + data type Error messages and most attributes
						32 + 4	Explicit attribute write
						36	Password (and speed) write
LAMPCTRL	0000	0001				4, 16 or 32	Special PATTERN used for lamp control
URAW	0000	0000	1110	1	0	20 or 36	Un-scaled, left shifted, unsigned value
SRAW					1	20 or 36	Un-scaled, left shifted, two's complement value
UNDEF + MODEM	0000	1111	1111	1	1	N×8 + 4	Undefined (default). Used for unused publisher and/or subscriber in for example the UART object
(any)	xxxx	xxxx	xxxx	1	x	4	Boolean

Fig. 6b.1

The data length – 20 or 36 bit, is determined by means of the telegram length.

The blue fields are the 9 bits, which are specified in attribute 12 in the Max-i controller for the publisher and the subscriber. Since the controller does not generate strings, it is possible for the device, which transmits the message, to specify a code page. This may for example be used for different upper ASCII pages (80H – FFH) and for languages, which us a different character set, like Greek, Russian, Chinese, Japanese etc. Note that since the code page is a part of the signature, it must match 100% for the message to be accepted.

Since the data types are a part of the signature, which is used for CRC check, it is NOT allowed to make devices, which can accept more data types for a given value. All 20 bit in the signature shall be known and verified (no don't care bits).

As described previously, 4-bit I/O data in a short telegram is **always** regarded as Boolean I/O **no matter the data type**. In this way, it is possible to publish and/or consume both an analog and a Boolean value for each data type, but not simultaneously. This may for example be utilized for lamp dimmers where it makes it possible to set a lamp to 9 predefined levels and step the light up and down by means of simple 4-bit messages or set the level or color to an absolute value by means of a 20-bit or 36-bit message where the last 4 bits are used to set the speed of the smoothing filter. In case of network telegrams (shown with yellow background), the 4 bits are used for special purposes like flash synchronization. In principle, 4-bit telegrams may use any data type, but to make it possible for a subscriber to accept both a short and a long telegram (same signature), the data type of the long telegram must be used.

For each data type, there are up to 4 different definitions:

- Coding of 20-bit and/or 36-bit data **as defined in the data type** (long code). Since there is only one register to store the received data for **long** implicit messages and group messages (attribute 2), the coding must be the same and there is therefore only one subscriber data type. The data types of network messages are fixed to either TIME or PATTERN depending on the network ID (0 – 7).
- Coding of 4-bit implicit data (short code).
- Coding of 4-bit group data (short code).
- Coding of 4-bit network data (short code).

In practice, the coding for all 4-bit data is however basically the same, so in most cases, the coding is just referred to as long or short, but there may be minor differences and/or restrictions in the use depending on the I/O object. In for example the LAMP object, a 4-bit group message may switch a lamp off, but this is not allowed in the BOOLEAN object.

For the data type TIME, the coding of the least significant 4 bits of a long message is identical to the coding of short messages so that the result is the same no matter if the message is short or long, but for other data types like LAMPCTRL, these 4 bits are interpreted differently in the two situations. Some data types like FIX and FLOAT have no short message at all.

The consumed value is stored in a 36-bit register (attribute 2) where the least significant 4 bits may hold any Boolean value. It is therefore not possible to hold both a 36-bit value and a 4-bit value simultaneously (would require 40 bits). Long data types are therefore interpreted in one out of three ways:

- Full 20-bit or 36-bit data such as FIX and the BCD types. No Boolean data is possible.
- 16-bit, 32-bit or text data plus 4 bits, which may be used for different purposes. They may for example be used to update 16 different parts of a text display in case of ASCII or UNICODE or be used to set 9 different values/parameters in case of LAMPCTRL. Note that in case of data to more devices in the same telegram, the 4 bits are common to all devices as described previously.
- 4-bit Boolean data with (or without) supplementary 16-bit or 32-bit information such as timestamp/system-time for the TIME type. When supplementary data are included, they are always transmitted before the Boolean value in the same way as 16-bit or 32-bit data. Note that supplementary data is not a separate process value so that the data does not contain both an analog and a Boolean value simultaneously. Unlike the other two interpretations, the Boolean value is not only updated in case of a 4-bit telegram, but may also be updated in case of a longer telegram depending on the I/O object.

To enable an unambiguous data interpretation, only the interpretation shown in the table is allowed for each data type except for PATTERN, which may use any interpretation. In this case, it doesn't matter that special knowledge is necessary to interpret the two least significant bits since it is anyway necessary with special knowledge to interpret PATTERN data.

The data type for the publisher and the subscriber need not be the same and in many cases, the data types are used to select between different I/O behaviors of an object as described in layer 7. If for example the subscriber data type of the BOOLEAN object is TIME, the outputs are interpreted as pure Boolean signals, but if the data type is instead LAMPCTRL, the outputs are regarded as 4 dimmable control lamps.

Because the FIX exponent or the data type of a value do not change, they are **not** transmitted as data bytes, but transmitted in the signature as specified previously. As the FIX exponent or the data type affects the validity of the data, it is **not** stored in the signature attribute (described later), but in the I/O object specification, which is protected by password 3.

6b.2 Boolean Data

Because Max-i uses the big-endian model, the most significant of the four bits is called bit 0 and the least significant bit is called bit 3.

⚠ **The four bits of a Boolean value shall be regarded as 16 states of the same value/state/code – never as 4 individual signals!** If the outputs are used to drive more control lamps, they shall therefore show the state of a single device/equipment. **The passive state, which shall be selected during power up (PUR) and in case of a timeout (described later), shall be 0000B**.

If only one output is needed, the on-state should be 1111B. In this way, it is possible to parallel connect two or more outputs for higher drive current.

During PUR, all Boolean inputs shall be reset to 0 and after the PUR pulse, only input objects where at least one input is 1 shall cause an event driven transmission. This reduces the number of telegrams during power up considerably.

Usually, a telegram shall be transmitted each time an input changes state. However, it shall be possible to set each individual input in the so-called on-only mode where a telegram is only transmitted when the input goes logical high, but **not** when it goes low, and in off-only mode where a telegram is only transmitted when the input goes logical low. If on-only and off-only are selected simultaneously, the event driven transmissions shall be entirely disabled. This may be utilized to prevent collisions between replies from more subscribers for example in case of traffic lights.

The 4-bit values are extremely useful in many applications:

- *For input from a numeric keypad with up to 16 keys.*
- *For input from a boolean joystick (4 switches).*
- *For light switches/dimmers like the one in Max-i where a 1-bit input from a single button is translated to for example 0000B = off, 01xxB = ramp down, 10xxB = ramp up and 1111B = 100 % on. In this way, it is ensured that all lamps are synchronized even though they are operated by means of a toggle switch (a simple 1-bit toggle signal may cause some lamps to turn on while others turn off). By means of the subscriber, the lamp dimmer may also synchronize to such 4-bit signals from other switches so that it is easy to implement two-way switching. The remaining two bits (xx) may for example be used to control the color temperature.*
- *For redundant signals in case of safety systems. 00xxB = off, 11xxB = on, 01xxB and 10xxB = error. Note that because of the required synchronization between the two signals so that a 01xxB or 10xxB state is avoided, this cannot be implemented with two separate signals, but requires a 2-bit signal.*
- *For anti-valence supervision where one bit is high while the other is low. If both bits are equal, there is an error. Because anti-valence supervision is a violation of the rule that the passive state shall be 0000B, it is highly recommended to use redundant signals instead if this is possible.*
- *For driving lamps with up to four states – off (00xxB), slow flash (01xxB), fast flash (10xxB) and on (11xxB). With traditional point-to-point wired systems one signal is enough for flashing, but in case of a fieldbus system, it is highly impractical to send a telegram each time a flashing lamp should change state, and the unavoidable jitter may also cause an unpleasant flicker.*
- *For driving lamps and LED's with more colors like off (00xxB), red (01xxB), green (10xxB) and yellow/both (11xxB).*

- *For setting and indicating the mode of operation for an equipment, which may run in automatic, semiautomatic or manual/hand mode or may be in more state like:*
  - *Start, run, running.*
  - *Hold, holding, held.*
  - *Pause, pausing, paused.*
  - *Stop, stopping, stopped.*
  - *Abort, aborting, aborted.*
  - *Reset, resetting, idle.*
  - *Resume, resuming, running.*
  - *Restart, restarting, running.*
- *For indicating the position of valves with two limit switches, but a common communication interface.*
- *For controlling equipment, which must maintain its position in case of a power failure like for example a shutter on a chain conveyor or a two-way valve. Such equipment is typically pneumatically activated and controlled by means of a four-way directional valve with two actuator solenoids – one in each end. Such a valve requires the two drive signals 01B and 10B and in some cases also a neutral 00B state. Without a two-bit signal it would be necessary first to deactivate one of the solenoids, then get an acknowledge signal that the command is received and at last activate the other solenoid. In a fieldbus system it would therefore be necessary with three telegrams instead of one, but in case of a fieldbus system like Max-i or CAN, which uses the publisher/subscriber model, the acknowledge is even impossible because the transmitter do not know the amount of receivers.*
- *For controlling two-way conveyors with 2 – 3 speeds like for example:*

  Slow left = 1000B, fast left = 1100B, stop = 0000B, slow right = 0010B, fast right = 0011B.
- *For controlling motor operated control valves, windows or other devices, which have an up signal (10xxB), a down signal (01xxB) and a passive state (00xxB). The two remaining bits may for example be used to set the up/down speed.*
- *For controlling two-speed motors with three states – stop (00B), low speed (10B) and high speed (11B).*
- *For controlling three or four level devices such as heating (W1 – W4), emergency heating (E1 – E4), cooling (Y1 – Y4), fan speed (G1 – G4) etc. like:*

  Off = 0000B, low = 1000B, medium = 1100B, high = 1110B and very high = 1111B.
- *For controlling contactors with more states like for example a star-delta starter or a 4-state Korndorfer starter – stop (00xxB), low (01xxB – autotransformer), medium (11xxB – autotransformer as serial coils) and full (10xxB – autotransformer bypass).*
- *For controlling two-way valves, which has a center position where the material stream is divided into two streams.*
- *For material distributers with 4 (1000B, 0100B, 0010B and 0001B) or up to 16 positions.*
- *For alarms with more priorities or for example a choice between a warning and an error.*
- *For multi-level limit switches like for example +3 (1100B), +2 (1000B), +1 (0100B), neutral (0000B), -1 (0001B), -2 (0010B) and -3 (0011B).*
- *For making it possible with the same function name for a function with three or four states no matter if this is controlled by means of a toggle- or changeover switch or it is controlled by means of more mono stable push-buttons. If only one-bit-values were used each of the mono stable buttons should have its own function code, however this would make it impossible to name a changeover switch, because the name of the switch should then actually change after the position.*
- *For traffic lights with up to 4 lamps (red, amber, green and green arrow).*
- *For making it possible for two gauges to work together on one signal, for example in case of big shutters with two limit switches with each a communication interface. One gauge has 0 as the most significant bit and therefore sends 0xxxB, and the other has 1 as the most significant bit and therefore sends 1xxxB.*

The ability to address single (4-bit) boolean values is a great advantage of Max-i. In many other bus systems a single binary value must be changed by means of AND and OR functions on groups of bits.

When boolean data are specified by means of XML, more data types must be used to be able to select the telegram length, the data type and the object type as shown in the table below:

XML Type	Max-i Type	Length	Object
BOOLEAN	TIME	Short (4 bit)	I/O
TIMESTAMP	TIME	Long	I/O
LOWGROUP	PATTERN	Short (4 bit)	Network
GROUP	PATTERN	Long	Network
PATTERN	PATTERN	Long	I/O

Fig. 6b.2

6b.2.1 Boolean High Reliability and Safety Messages

A common method of improving the safety on traditional hardwired input signals is to use two bits and the so-called anti-valence supervision where one signal is high while the other is low. In case of a power failure or a short circuit, both signals will be equal and the error can be detected. This may of course also be done on Max-i, but it violates the rule that the passive state shall be 0000B and the solution is not optimized for a fieldbus system. In case of a power failure, no telegrams are transmitted at all, so it is not necessary to be able to detect this situation by means of the signals. It is better to use two or more independent inputs switches to generate the same signals, that is, 00xxB or 11xxB or even 0000B or 1111B and in this way create some redundancy. In case of a short circuit or a failure on one switch, 01xxB or 10xxB is transmitted. This solution makes it possible to use 0000B as the passive state and it fits with safety systems in accordance with AK6 of DIN V 19250, category 4 of EN 954-1 and SIL 3 of IEC 61508 where it is required with two-channel redundant and diverse hardware and data processing. To fulfill this requirement, a safety switch or device must be coupled to the Max-i controller by means of at least two completely independent input channels and the inputs shall be completely independent. Because of the hardwired connection and the direct bus interface it is not necessary with a further check on the data. With traditional safety systems, it may be necessary with test pulses etc. on all inputs to ensure that they are functional, but this is not necessary with a direct bus interface because the connections between the safety devices and the bus interface are not taken out of the device so that short circuits etc. are not possible.

To protect against an error in the circuit, which combine the input bits to one telegram, the transmitted telegram shall be verified. This may be done by means of the receiver shift register, which is used for scrambling. The first latch of this register holds the received bit, which is delayed one bit compared to the transmitted bit. This bit may be compared with the inputs by means of a separate circuit (not the same input circuit and multiplexer). In case of an error, the telegram shall be aborted immediately.

It is also possible to use two completely independent transmitters, which drives each one half of the output stage, so that one transmitter controls the transistor, which drives the line high, and the other transmitter controls the transistor, which drives the line low. Only if both transmitters agree 100%, they will be able to collaborate and create a valid telegram. If this solution is used, a hardware circuit shall ensure that both transistors cannot be turned on simultaneously in case of an error.

With the publisher/subscriber model and the fast polling system, which ensures that all data or the lack of a poll answer is received **exactly** simultaneously in all receivers, it is the transmitter, which is critical, as any number of receivers may be parallel coupled according to the wanted safety level. The outputs of the receivers shall be combined by means of approved safety/emergency-stop relays with positive-guided contacts or by means of serial connected motor contactors. It is highly recommended to supervise the various outputs and generate an error if they do not agree.

The receivers can be Max-i controllers, PLC systems or a combination. In case of PLC systems, the systems shall be diverse to protect against software errors. This may be done by using PLC's from two or three different vendors programmed by two or three different programmers. In this way, it is very unlikely that the channels have the same hardware and/or software bugs.

6b.3 Metric and US/Imperial Values

Max-i has four data formats for process values – FIX, FIXBCD, BCD and FLOAT. These data formats shall only be used for values, which have been scaled and converted to metric values or to US/Imperial values expressed in decimal notation like 5.37’ instead of the fractional notation like 5’ 4 7/16”. If the data are not scaled, the data type RAW or PATTERN shall be used instead. For the moment the following standard units are recommended:

Type	Unit	Symbol	
		Metric	Imperial
Acceleration	gravity	g	
Temperature	degree Celsius or Fahrenheit	°C	°F
Length	meter or foot	m	ft
Level	meter or foot	m	ft
Weight and mass	kilogram or pounds	kg	lb
Volume	liter or gallon	l	gal
Density	kilogram per liter or pounds per gallon	kg/l	lb/gal
Viscosity	centipoises	cPs	
Force	newton or pound-force	N	lbf
Torque	newton meter or pound-force-foot	Nm	lbf-ft
Expansion	meter or foot	m	ft
Pressure	barometric or pounds per square inch	bar	psi
Time	seconds	s	
Frequency	hertz	Hz	
Rotating speed	revolutions per minute	RPM	
Volume flow	liter or gallon per second	l/s	lb/s
Mass flow	kilogram or pounds per second	kg/s	lb/s
Electrical current	ampere	A	
Electrical potential	volt	V	
Power	kilowatt	kW	
Work of energy	kilowatt hours	kWh	
Resistance	ohm	Ω	
Conductivity	siemens	S	
Capacitance	microfarad	µF	
Inductance	Henry	H	
Amount of substance	mole	mol	
Humidity or moisture	percent	%	
Luminous intensity	candela	Cd	

Fig. 6b.3

The metric units follow the SI Units (Système International d'unités) except for temperature, pressure, capacitance and work of energy, where the SI units Kelvin, Pascal, Farad and joule are impractical for process control applications.

It is not recommended to mix metric and US/imperial devices in the same system/plant.

6b.4 Accuracy

The state-of-the-art accuracies for industrial measurements are:

Type	Accuracy	
	%	Bits
Weight		
Load Cell	0.01	14
Hopper Scale	0.05	11
Pressure	0.025	12
Temperature	0.1	10
Mass Flow	0.1	10
Volume Flow		
Liquid	0.5	8
Gas	1	7
Level		
Pressure	0.05	11
Radar	1	7
AC Current (RMS)	0.5	8
Conductivity	0.5	8
PH/Redox	0.5	8

Fig. 6b.4

Note than when a value from an A/D converter is converted to SI or US/imperial units, one more bit may be necessary to avoid reduction of the accuracy. Therefore, 20 bits makes it possible to transmit any signed 18-bit value or any unsigned 19-bit value without loss of accuracy.

6b.5 DATA TYPES

6b.5.1 FLOAT

The FLOAT data type uses the widely used ANSI / IEEE 754 32-bit single precision or 64-bit double-precision floating-point format plus four bits as shown below:

Sign	Exponent		Mantissa fraction (significant)	Free
1 bit	8 bit (excess 127) or 11 bit (excess 1023)		23 bit or 52 bit	Bit 32 – 35 Bit 64 – 67
		J		

Fig. 6b.5

There is no short 4-bit coding and group messages are not used.

A FLOAT value shall be left shifted as other analog values. The interpretation of the remaining bit 32 – 35 or 64 – 67 is not defined, but they may for example be used to increase the resolution of the mantissa or to select between 16 different values. For this reason, they shall be 0 if not used. The length of the telegram is used to distinguish between 32-bit and 64-bit data.

The sign bit is used to specify the sign of the value. If it is zero, the value is positive, and if it is one, the value is negative.

The mantissa (significand) has two parts: A 1-bit binary integer (also referred to as the J-bit) and a binary fraction. The mantissa is shifted in such a way that the most significant 1 of the binary value becomes the J-bit. This is called normalization. Because the J-bit is always 1, it needs not to be specified. Therefore, it is an implied value in the IEEE format. The advantage is that the accuracy of the mantissa is increased by one bit, but it simultaneously creates more special cases and exceptions, which must be handled.

The exponent is the power of two needed to correctly position the mantissa to reflect the number's true arithmetic value. To facilitate comparisons among floating-point values, it is held in excess-127 or excess-1023 notation, which means that 127 or 1023 is added to the actual exponent so that the biased exponent is always a positive number.

The reason for using excess notation instead of a signed integer is to make it easier to compare values by means of a simple integer comparison, which is possible because of the normalization, and the reason for choosing 127 and 1023 is that the smallest normalized number can then be reciprocated ($^1/_X$) without overflow.

Because of the J-bit, the mantissa (without sign) of a normalized value is always greater than or equal to 1 and less than 2. Therefore, it is not possible to represent 0 directly, but if both the exponent and the mantissa is zero then the value is defined to be zero.

For **extremely** small values, it may not be possible to normalize the value because of the minimum value for the exponent ($2^{-127} \approx 5.9 \times 10^{-39}$). When the biased exponent is zero, smaller numbers can only be represented by making the J-bit (and perhaps other leading bits) of the mantissa zero. The values in this range are called denormalized or tiny values. The use of leading zeros with denormalized values allows smaller values to be represented, but it causes loss of precision. The denormalized values gives a gradual turn over from normalized values to zero, where all the mantissa bits are shifted out to the right by leading zeros.

The set of possible data values can be divided into the following classes:

- Zeroes
- Normalized numbers
- Denormalized numbers
- Infinities
- NaN (Not a Number)

NaNs are used to represent undefined or invalid results, such as the square root of a negative number.

The classes are primarily distinguished by the value of the exponent field, modified by the fraction. Consider the exponent and fraction fields as unsigned binary integers:

Class	8-bit Exponent	11-bit Exponent	Fraction
Zeroes	0	0	0
Denormalized numbers	0	0	non zero
Normalized numbers	1-254	1-2046	any
Infinites	255	2047	0
NaN (Not a Number)	255	2047	non zero

Fig. 6b.6

The two infinities, + and -, represent the maximum positive and negative real numbers, respectively, that can be represented in the floating-point format. Infinity is always represented by a zero mantissa (fraction and J-bit) and the maximum biased exponent. The signs of infinities are observed, and comparisons are possible. Infinities are always interpreted in the affined sense; that is, -infinite is less than any finite number and +infinite is greater than any finite number.

Whereas denormalized numbers represent an underflow condition, the two infinity numbers represent the result of an overflow condition. Here, the normalized result of a computation has a biased exponent greater than 254 or 2046.

6b.5.2 FIX and RAW

The IEEE floating point format is a widely used standard, but it is very difficult to generate and handle by means of simple low-cost hardware without a microprocessor.

There are several reasons for this:

- *Because of the implied J-bit, it is not possible to use a fixed exponent (denormalized numbers).*
- *Because the mantissa is a positive number plus sign instead of a two's-complement number, it is not possible to offset compensate the value by means of a simple adder/subtracter, or by means of a simple preload of the pulse counter in case of a (synchronous) voltage to frequency A/D converter.*
- *It is not possible to use the data from an A/D converter directly.*
- *It is necessary with a special exception to handle the number zero.*
- *It is very difficult and time consuming to do digital filtering on the values, unless a special IEEE hardware floating-point unit is present.*
- *It has fairly low efficiency for process control because 32 bits are needed even for measurement values with the usual industrial accuracy of approximately 12 bits.*
- *There is no unsigned format, which reduces the accuracy by one bit for values, which are only positive like many process values. This also reduces the efficiency.*

Because of this, Max-i has the fixed-point data type FIX, which makes it possible for simple hardware based devices to generate process values in SI or US/imperial units with very high efficiency.

The FIX data type consists of an 8-bit exponent and either a signed or unsigned 20-bit (SFIX20, UFIX20) or 36-bit mantissa (SFIX36 or UFIX36). There is no short 4-bit coding.

The mantissa is either:

- A signed, left-shifted, two's-complement, fractional, 20-bit or 36-bit number in the range from -1 to 1-2^{-19} (sign + 19 bits) or -1 to 1-2^{-35} (sign + 35 bits) – a so-called 1.19 or 1.35 fractional number, where the most significant bit is actually the sign so that there are only 19 or 35 significant bits. Due to the denormalized numbers, the number range for positive values is:

  FIX20: Minimum $2^{-19} \times 2^{-128} = 5.6052 \times 10^{-45}$. Maximum $(1-2^{-19})2^{127} = 1.7014 \times 10^{38}$.
  FIX35: Minimum $2^{-35} \times 2^{-128} = 8.5528 \times 10^{-50}$. Maximum $(1-2^{-35})2^{127} = 1.7014 \times 10^{38}$.

- An unsigned, left-shifted, fractional, 20-bit or 36-bit number in the range from 0 to 1-2^{-20} (20 bits).

For signed values, the default radix point is located between bit 0 and 1 so that the maximum positive value without exponent is 0●111,1111,1111,1111,1111B = 1-2^{19}, and the maximum negative value is 1●111,1111,1111,1111,1111B = -1.

For unsigned values, it is located one bit to the left so that the maximum value is ●1111,1111,1111,1111,1111B = 1-2^{20}.

The exponent is the power of two needed to correctly position the mantissa to reflect the number's true value in SI or US/imperial units. In the ANSI / IEEE 754 format, the exponent is held in excess 127 notation to facilitate simple number comparison, but such a comparison is not possible with FIX because of the fixed exponent, which creates denormalized numbers, so there is no reason not just to use a much simpler signed integer, which simply tells how many bits the radix point must be moved to the left (negative number) or right (positive number).

After commissioning, the exponent is fixed and do not change! It is therefore not transmitted as data, but as a part of the signature (the data type). As the minimum data length for the IEEE format is 36 bit, FIX20 can save two bytes if a 20-bit mantissa is enough, which is usually the case for process control! Because the exponent is the most significant part of the signature, the total FIX value is directly readable in the implicit message as shown below:

Start	Priority	Identifier	Data	Signature			
			20 or 36 bit Mantissa	8-bit Exponent	0000 (data type)	0 = unsigned 1 = signed	7 bit ID Hamming code
			Total 28-bit or 44-bit FIX value				

Fig. 6b.7

The data type RAW is a special case of FIX where the value is either an unsigned or a two's complement value, the exponent is 0000,0000B and the data type is 1110B. It is typical used for outputs where the data is not scaled to metric or US/Imperial values. It may also be used for inputs (raw A/D converter data), but it is highly recommended to use the FIX format instead if this is possible. Since the data is left shifted, it is also possible to use a "16 bit analog plus 4 bit boolean" format. This may for example be used for a D/A converter and in case of data to more devices in the same telegram where the 4 bits are common to all devices and therefore cannot be parts of the analog values.

FIX has no short 4-bit coding and since there is no output for fix values in the Max-i controller (only a RAW D/A-converter output), it is only used for the publisher. It is therefore not necessary that the subscriber in a Max-i controller can handle a FIX message.

When FIX data are stored in a microprocessor (MPU), they should be expanded or truncated and stored as 32-bit data with one of the following formats depending on the multiplier in the MPU:

FIX with 64-bit, 2×32-bit or 1.31 fractional 32-bit multiplier		
20-bit mantissa	0000 expansion	Exponent
Left shifted, truncated 36-bit mantissa		
24 bit		8 bit

FIX with 9.23 fractional 24-bit multiplier with 8 guard bits		
Exponent or guard bits	20-bit mantissa	0000 expansion
	Left shifted, truncated 36-bit mantissa	
8 bit	24 bit	

Fig. 6b.8

A 24-bit mantissa is considered enough for any practical process control applications.

Because FIX data are left shifted, it is possible to change the resolution of an A/D converter without informing the receivers. *This is a* ***very*** *important property of the FIX format. It is even possible for a FIX20 subscriber to accept FIX36 data and vice versa – just with a slight loss of accuracy. With right shifted data, it would be necessary to change the exponent if the resolution is changed to keep the values in SI or US/imperial units. It would therefore be necessary for the receivers to deal with a floating-point format* ***and it would not be possible to embed the exponent in the signature, as this would cause the receivers to reject the new telegrams!***

The SFIX format is extremely efficient for doing digital signal processing. In fact, it is both ***much*** *faster and gives a higher accuracy than floating-point! This is how it works:*

1. *The amplification before the A/D converter is chosen in such a way that the converter range be utilized as much as possible and the total fixed-point number with the exponent is in SI or US/imperial units. Note that the result of an A/D conversion is always a fixed number of bits – never a floating-point value – and in most cases, also a left shifted number.*

2. *The exponent is fixed and saved for later use.*

3. *The mantissa is loaded **left shifted** into bit 0 in case of ordinary multipliers. If a 9.23 fractional 24-bit multiplier is available, the mantissa should instead be loaded into bit 8 and bit 0 – 7 can now be used for guard bits to avoid overloads in multiply-and-accumulate (MAC) instructions.*

4. *The required number of MAC instructions for the digital filter is executed. There are two ways to do this:*

   If the processor supports fractional arithmetic, both the value and the filter coefficient are regarded as 1.N two's-complement fractions, where the most significant bit is defined as a sign bit, and the radix point is implied to lie just after the sign bit. The range for an N-bit two's-complement fraction is -1 to 1-$2^{(1-N)}$. A 1.N fractional multiplication generates a 2.2N result. To keep the radix point aligned, the result is automatically left shifted one place. This may create saturation if -1 is multiplied by -1, but in practice this saturation it completely unimportant as the result is very close to one (1-$2^{(1-N)}$). Following additions may of course create an overflow, but many digital signal processors like for example Blackfin from Analog Devices and dsPIC30F from Microchip use a 40-bit 9.31 fractional accumulator with 8 guard bits. Such an accumulator has a range of -256 to 255.99999999953 and is therefore able to handle any filter with up to 256 coefficients without any precautions. MCU's designed for Max-i should use 8 guard bits and the second data format shown above and will therefore be able to handle 256 coefficients. If there are no guard bits, the filter quotients must be reduced so that an overflow never occurs.

   *If the multiplier does not use fractional arithmetic, then the N × N integer multiplication generates a 2N result. Only the **most significant** half (high part) of this should be used in the further calculations. If for example two left shifted 24-bit numbers are multiplied in a 32-bit processor, the result is the most significant 48 bits of the 64-bit result, but only the most significant 32 bits are used.*

5. *The result is converted to SI or US/imperial units by means of the saved exponent. Unlike the MAC operations, this only has to be done once.*

Even though this method is much faster than floating-point arithmetic, it simultaneously has a higher accuracy. A state-of-the-art A/D converter has a resolution of approximately 24 bits, but with 1.31 fractional arithmetic, all intermediate calculations are done in at least 30 bits accuracy (2.30 fraction). With single-precision floating-point arithmetic in the IEEE 754 format, all intermediate results are rounded to 24 bits.

6b.5.3 Integers

Max-i does not have a data type for signed and unsigned integers. This is not necessary since the exponent of the FIX data type is fixed and embedded in the signature. If for example all 20 bits of a SFIX20 shall be regarded as a signed integer, the radix point must be moved 19 bits to the right from X●XXX,XXXX,XXXX,XXXX,XXXXB to XXXX,XXXX,XXXX,XXXX,XXXX●B. so the exponent is 19. If the value is byte aligned so that only the most significant 16 bits are used, the exponent must be 15. In case of unsigned integers, the exponent is 20 so that the maximum value is 1111,1111,1111,1111,1111B = 1,048,575. In this way, integers may be transmitted with the FIX data type with no loss of efficiency.

6b.5.4 FIXBCD

The FIXBCD format is similar to FIX, but all data are in BCD format, that is, all digits use their own 4-bit nibble with the number range 0 – 9 plus the symbols "-" and "-1" for signed values and no-value ("- - - - -") and maybe "E", "r" and "o" to be able to write "Error". The FIXBCD format is primary intended for numerical displays like 7-segment displays driven directly by a Max-i controller. As shown below, it is possible to handle a standard 4½ or 5-digit display (±19999 or .99999 - 99999) with a 20-bit FIXBCD and up to a 9-digit display with a 36-bit value. More than 8½ digits (signed) or 9 digits (unsigned) are not relevant for industrial control.

Point	20 bit					Optional 16 bit				Radix Point	Sign
	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	1 bit
●	MSDigit (MSD)	Digit	Digit	Digit	Digit	Digit	Digit	Digit	Digit	Unsigned integer	0 = unsigned (0-9 as MSD) 1 = signed (±1 as MSD)

Fig. 6b.9

Since the radix point is fixed, so is the dot point on the display and the radix point bits are therefore primary used for information to be able to interpret the value if the display is not visible, but they also increases the Hamming distance as known bits.

Like FIX, FIXBCD is left shifted. If for example a 7-digit display receives a 36-bit value, the extra two digits may just be thrown away or used for rounding. If it receives fewer digits in case of a 20-bit value, the missing digits may just be shown as 0 or blanked digits.

The default radix point is located to the left as shown so that there is no need for left shifting and the exponent therefore can be a 4-bit unsigned integer, which makes it possible to move the radix point up to 15 places to the right.

FIXBCD has no short 4-bit coding (all 20 or 36 bit used) and no special group coding.

6b.5.5 BCD

The BCD format is similar to FIXBCD except that the first nibble is used to specify the decimal point (floating point). The radix point specification uses the same coding as the FIXBCD data type so that values up to ±1999999999999999.(99…) or 9999999999999999.(99…) are possible in a 68-bit telegram (64 + 4 bit).

20 bit						**Optional 16 bit**				**8 bit**		**8 bit**		**8 bit**		**8 bit**	
4 bit		4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit	4 bit
Radix point	●	MSD	Digit	Digit	Digit	Digit	Digit	Digit	Digit	Digit	Digit	Digit	Digit	Digit	Digit	Digit	Digit

Fig. 6b.10

A telegram length above 36 bits (light blue field) is only possible in modem mode where the telegram length may be any number of bytes. Like FIXBCD, the value is followed by a data type code and a sign bit, which specifies whether the value is unsigned (0-9 for MSD) or signed (±1 for MSD).

If floating point BCD is not needed, it is recommended to use FIXBCD because of the higher safety, less risk of misreading (no floating point) and one more digit for a given number of words.

As for FIXBCD there is no short 4-bit coding and group messages are not used except for lighting control.

6b.5.6 STRING

This data type is used for all kind of text strings in 8-bit (ASCII) or 16-bit (Unicode) format depending on bit 12 of the signature (0 = 8 bit and 1 = 16 bit), and the additional 8-bit data type makes it possible to specify 256 different code pages including for example default Latin (0000,0000B), Greek, Russian, Chinese, Japanese and Korean.

There is no short coding, but group messages may be used for example to control the light level of a display.

The ASCII type (least significant 7 bits) is also used for XML, JSON and MQTT coded data like for example:

```
XML:  <plant name="PlantX">
          <function name="MassFlow1" value="36.1"/>
      </plant>


JSON: {
          "PlantX" : {
              "MassFlow1": 36.1
          }
      }
```

```
MQTT: PUBLISH
      packetId                4310
      topicName ”PlantX/MassFlow1”
      qos                        1
      retainFlag              true
      payload               ”36.1”
      dupFlag                false
```

Because all XML and JSON values are in SI or US/imperial units, the unit for mass flow is implied to kg/s or lb/s. Note that when data are converted to XML, JSON or MQTT, it is not necessary to specify the data type to for example FIX or FLOAT. Like a pocket calculator, an exponent is only used, if the data would otherwise be out of range.

6b.5.7 TIME

This data type is used for 4-bit Boolean signals with or without time stamp or time data for a display.

A complete TIMESTAMP uses three 16-bit words as shown below:

Boolean ←	Year	Month
4 bit	8 bit	4 bit

Day					Hour					Minutes					
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Seconds						Milliseconds										→ Boolean			
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35

Fig. 6b.11

To improve the efficiency and avoid that a Max-i controller must include a complicated calendar, the first word with the year and the month is **not** transmitted on Max-i and the Boolean value is instead transmitted in bit 32 – 35 as shown above, but the complete time stamp shall be restored by adding year and month when a received time is stored in a data base.

The time shall be counted directly in years (2000 + 8-bit year code), months (1 – 12), days (1 – 31), hours (0 – 23), minutes (0 – 59), seconds (0 – 59) and milliseconds (0 – 999). This is only a minor complication of the counter, but it saves many calculations later on.

Day = 0 is used to indicate an invalid time. If a device does not include a real time clock with battery backup, the time shall be set to all zero after power-up and shall then start counting except that it is **not** allowed to count day 0 up to day 1. The only thing, which can change day 0 to any other day, shall be a write to the system time attribute. Day = 0 may also be used to indicate an invalid value if a device does not have any RTC at all, but wants to use the data type TIME for compatibility with other boolean devices.

Since the Max-i controller does not need a calendar, it is **not** a requirement that it knows the length of the month, so the day specification may just wrap around after 31 days (31 → 1), but this is not a problem. If for example a SCADA system receives a TIME with date = 31 just after midnight April 30, it shall just regard the day as being May 1. The date will then be corrected to 1 by the next update of the system time attribute, **which must take place within a day**.

The time should be specified in Global Positioning System (GPS) time, which is an atomic time scale used by the GPS system.

The benefit of the GPS time is that it is easy to synchronize to and is a continuously increasing time, which is suitable for chronological recording of events down to the microsecond range – even in distributed process plants and big power networks and in ships and aircrafts, which constantly pass time zones. The GPS time tracks within less than 28 nS with the

Coordinated Universal Time (UTC) popularly known as GMT (Greenwich Mean Time) ***except for an integer number of leap seconds****; which are inserted into UTC to keep it within 0.9 seconds of Earth's rotation. The GPS time was zero at 0h January 6th 1980 and since it is not perturbed by leap seconds it is now (May 2023) ahead of UTC by 18 seconds. In the same way, the International Atomic Time – TAI is 37 seconds ahead of UTC and always exactly 19 seconds ahead of GPS time. The CDMA digital cellular systems IS-95 and CDMA2000 also uses the GPS time as reference.*

At display time, the time should be shown in both GPS time, which always shows the correct mutual sequence between events, and in local time, that is, the time compensated for the time zone, any summer time and the number of leap seconds. The GPS signal contains information about the number of seconds to subtract to convert the time to UTC and thereby to local time.

The last 4 bits in the data field (bit 32 – 35) are **always** used for a Boolean value. **Because of this, TIME cannot be used together with a reception delay as the data would be common to all subscribers with the same identifier!** The data shall therefore always be left shifted and it shall therefore not be necessary to program the data length (in B20V and B20U of attribute 11 and 7).

Bit 0 – 15 or bit 0 – 31 in a long message shall always be used for time data, but the interpretation depends on whether TIME is used for process signals where the data is a timestamp of an event – this is in practice selected by means of GRPTYPE = 1 in attribute 12, or used for control lamps and traffic lights where the data is a time to be displayed on for example a 7-segment countdown display on a traffic lantern. This is selected by means of GRPTYPE = 0.

A numbering system like PNS, where control lamps have their own data code, may be used to ensure correct data interpretation if data is received in a debugger, which has no knowledge about how the data is used (GRPTYPE unknown).

6b.5.7.1 GRPTYPE = 1 (default)

In this mode, the four outputs are intended to drive actuators and the gamma of any output with inrush current limiting is therefore always 1. The content of the main register in attribute 2 shall be the following depending on the message length:

36-bit implicit, group or explicit message.	
Day, hour, minutes, seconds and milliseconds of **message** time	OR/AND function
32 bit	4 bit
0	32

20-bit **left shifted** implicit or group message.			
Display data to be transferred by means of SPI	Copy of Boolean	NU (reset)	OR/AND function
16 bit	4 bit	0000,0000,0000	4 bit
0	16	20	32

4-bit implicit or group message.	
Day, hour, minutes, seconds and milliseconds of **local subscriber** time No SPI transfer	OR/AND function
32 bit	4 bit
0	32

Fig. 6b.12

The orange background color indicates that these bits are not a part of the message.

20-bit data is only used for display data, which shall be transferred to for example a countdown display through the SPI interface.

A 16-bit time only has a resolution of one minute so with just a limited time synchronization it is better to use the local time, which is also used for 4-bit messages. 20-bit data would also require 3 XDATA bits in attribute 5 instead of 2 to select between transmission of data with and without timestamp, and there is no space for that. Note that when local data are used, the time may not be entirely synchronized so it is possible that the data seems to be received before it is transmitted.

*The time recording may be used during commissioning to find the time it takes from a command is issued until a response **from the process** is received – simply by subtracting the time in attribute 1 (published acknowledge) from the time in attribute 2 (received command). Knowledge about this time difference is necessary to be able to set the timeout/alarm time in the control system (PLC), but usually it is very difficult to find in other systems. Since the local time is used in case of a 4-bit telegram, it is possible to measure both the pure execution time with great accuracy (local times only) or the time it takes from a SCADA system gives the command with timestamp until it is executed.*

The Boolean value and the belonging outputs shall be an OR/AND function between implicit messages and group messages as shown below:

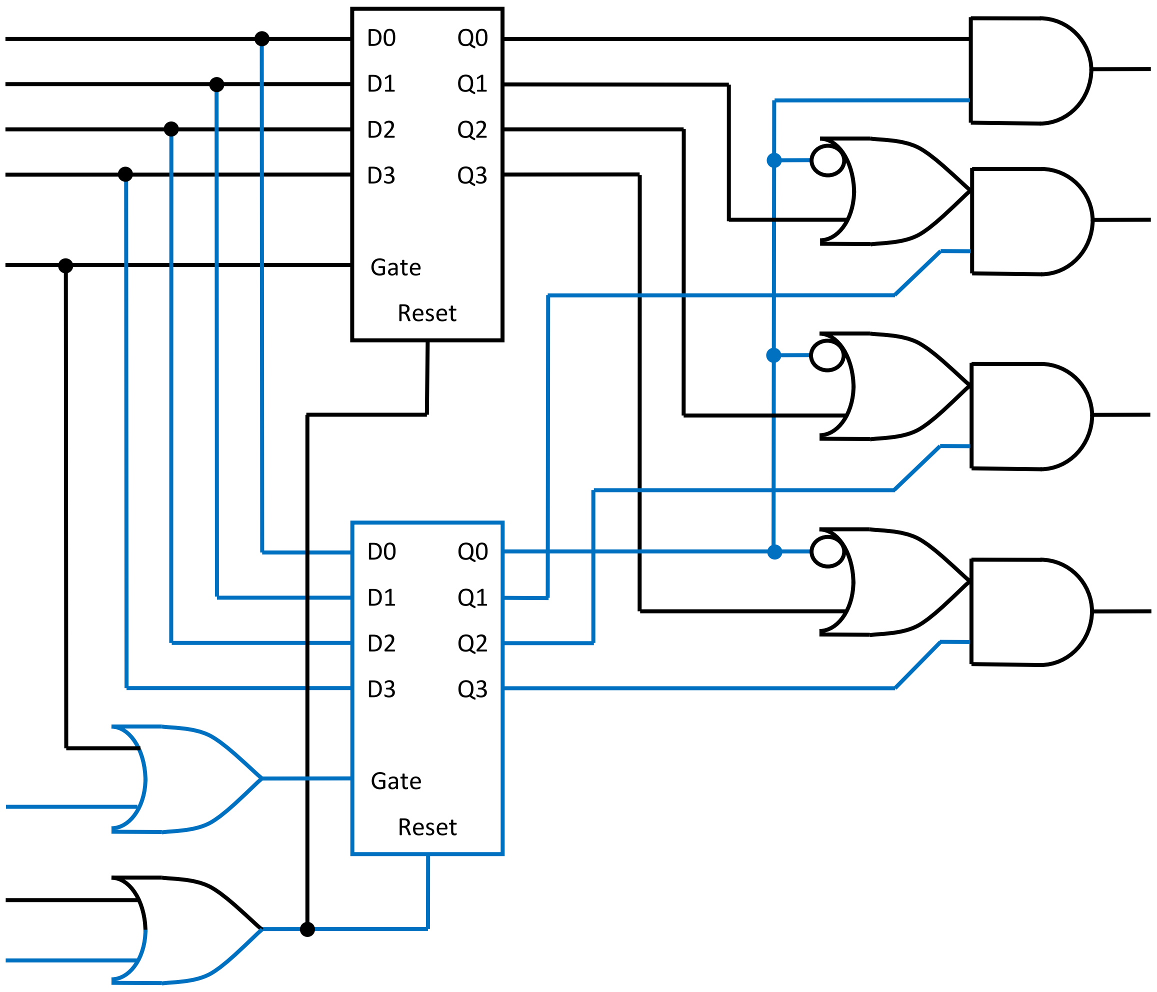


Fig. 6b.13

The black lines represent the part of the circuit used for implicit and explicit messages and the blue part is used for group messages. The group latches are always updated no matter if the message is implicit, group or explicit so that a group message cannot block the outputs permanently, but the main latches are not updated in case of a group message. All latches shall be reset in case of an implicit watchdog timeout or a group watchdog timeout.

If the consumed value (attribute 2) is read, bit 32 – 35 shall show the status of the outputs **after** the OR/AND function.

The circuit shown is the simplest way to fulfill the requirements for the 4 group latches, which shall be updated/changed in four situations:

- *They shall be set equal to the main latches (updated simultaneously) if an implicit message with the subscriber identifier is accepted. This makes it possible to override a previous switch-on or switch-off, but makes it inexpedient to use heartbeat. As an alternative, they may also be set to 1, but in this case, it shall be guaranteed that the main latches are updated* ***before*** *the group latches to avoid a glitch when the main latches go low and the group latches go high.*
- *They shall be set equal to the main latches if a write telegram for attribute 2 is accepted. As an alternative, they may also be set to 1 with the same precaution as above.*
- *They shall be updated each time a group message is accepted. This is used to enable and disable the outputs.*
- *They shall be reset until a new group message is received if a group timeout is specified and the timer runs out.*

As the main latches are reset (0000B) during PUR or if the output watchdog runs out, it doesn't matter which state the group latches have in these situations, so they may or may not be reset, but for high reliability applications where a single error must not lead to a dangerous situation, they should be reset.

The OR/AND circuit is very useful for many applications. In for example the United States, a three-level or four-level coding is very often used as shown below:

Level	Code	Standard designation in USA						
		Heating		Heating or cooling for heat pump	Cooling	Fan speed	Supply voltage	Common
		Main	Resistive (emergency)					
		White		Orange	Yellow	Green	Red	Blue
Off	0000							
1 – Low	1000	W1	E1		Y1	G1		
2 – Medium	1100	W2	E2	O/B	Y2	G2	R	C
3 – High	1110	W3	E3		Y3	G3		
(4 – Very high)	1111	W4	E4		Y4	G4		

Fig. 6b.14

In most cases, the letter codes for the functions refer to the standardized conductor colors as shown.

In this specification, this is called "Boolean standard levels" and it is to some extend also an integrated part of the specification of the 4-bit LAMPCTRL data type. Like a gray code, this coding has the advantage that only one bit change at a time and in case of for example heating, each bit may simply be connected to its own heating element so that 4 equal big elements create 4 equally spaced levels.

The simple levels, which may be selected by means of galvanic separated contacts, make it very easy to control a big variety of different devices without complicated communication protocols and/or special knowledge of the device. For example, a simple thermostat may switch a gas boiler on and off, but many start and stop operations may harm it or wear it down. If the boiler can be modulated, it is much better to reduce the level in one or more steps for a long period of time than turning it entirely off for many short periods.

Because of the AND function, the four levels may be reduced by transmitting a group message with the maximum level in the same coding. For example, 1100B disables level 3 and 4.

This may for example be used to save power if there is not sufficient "green" supply from wind turbines, solar panels etc. or the power network is heavily loaded for example in the morning or in the afternoon. It is also used for All-off (0000B) group messages, which may be used to switch off lamps, electronic equipment on standby and outlets connected to devices with potential fire hazard like for example coffee makers when you leave the house or go to sleep. A subsequent 1111B Back-on message may then turn all disabled outlets back on. As described later, these two messages are very easily generated by a lighting controller where the sequencer is just switched of in case of group 255 (common group) so that up, down and night-light messages are disabled.

It may also be used for a fail-safe emergency stop. In case of an emergency situation, a group message with the data 0000B shall be transmitted immediately, but if it is not received, the group latch shall be reset if the group timeout timer is programmed and runs out. A typical application is solar panels. They should be disconnected if no ***implicit*** *control message has been received within for example 10 minutes, but the reaction time to a missing emergency cut-off* ***group*** *message in case of a short circuit or an arc must be less than a second. This is the reason why group messages have their own timeout specification since it may not be the same as for ordinary implicit control messages.*

The OR-function makes it possible to set output 1, 2 and 3 if group bit 0 is 0 so that implicit messages and group messages works the same way for these 3 bits.

This may for example be used for a "Lock-all" function for locks (KEYPAD object) if output 1 is used to drive the locking pawl to the lock position and output 0 is used to unlock. Since there is no OR function on output 0, "Unlock-all" is not possible so a hacker or a thief cannot unlock all locks with a simple group command with 8-bit identifier (only 254 possibilities), but needs to know the identifier of each lock, which may be up to 31 bits long. It is recommended to use this Lock-all command in daily use instead of implicit messages to the various locks so that the identifiers for these are not disclosed so often and no doors are left unlocked.

6b.5.7.2 GRPTYPE = 0

In this mode, the four outputs are assumed to drive up to 4 lamps with Boolean information such as control lamps, signal towers stack lights and traffic lanterns with a gamma of 1 or 2.44, and the time specification is not the time of an event, but a time, which may be shown on a display such as a countdown display for a traffic lantern.

The logic state of the outputs is controlled directly by means of implicit or explicit messages, and group messages (20 bit or 4 bit) are instead used to control the light level of the lamps and any connected display as specified in the LAMPCTRL data type and as it is always the case when GRPTYPE = 0.

In case of long messages, the data length may be both 16 and 32 bit and the resulting content of the main register in attribute 2 is shown below:

36-bit implicit or explicit message.	
Display data to be transferred by means of SPI	Boolean
32 bit	4 bit
0	32

20-bit **left shifted** implicit message.			
Display data to be transferred by means of SPI	Copy of Boolean	NU (reset)	Boolean
16 bit	4 bit	0000,0000,0000	4 bit
0	16	20	32

4-bit implicit message.	
Not loaded and no SPI transfer	Boolean
32 bit	4 bit
0	32

Fig. 6b.15

The SPI interface should make it possible to transfer 8 – 36 bits in steps of 4 bit starting with bit 0 so that it is also possible to transfer bit 32 – 35 in case of a 36-bit message and bit 16 – 19 in case of a 20 bit message.

4-bit messages are not enough for automatic flashing, but if this is needed, the data type PATTERN with GRPTYPE = 0 may be used instead as specified under the PATTERN data type.

Automatic flashing could also decrease the safety if GRPTYPE = 1 as repetitive on and off commands at a fast rate may destroy the controlled process equipment so for safety reasons, is best not to have this possibility.

6b.5.8 PATTERN

The data type PATTERN is used for all kinds of groups consisting of bits, nibbles, bytes, words, other bit groups or values.

PATTERN may for example be used for:

- *Groups of binary values like all attributes except for network attributes and attribute 1 and 2.*
- *Error messages.*
- *Arrays, structures and records.*
- *Program files and memory dumps. The last four bits (bit 32 – 35) may for example be used to distinguish between address and data.*
- *Position data.*
- *Direct or coded drive of the segments and dot points in displays.*

The additional 8-bit data type field makes it possible to specify 255 different patterns beyond the general type 0000B as specified in layer 6a, but because there are only 4 bits for this in attribute 12 in the Max-i controller, the range 1 – 15 is reserved for standard patterns where 0001B is used to control lamps (LAMPCTRL data type), and the range 16 – 255 is free to be used for user defined patterns.

Since the additional data type is a part of the signature, it is not necessary for the Max-i controller to be able to recognize more types than it is able to interpret as telegrams with other types will be rejected.

Because of the very easy conversion of process values to the FIX format, it is highly recommended **not** to use the PATTERN type for normal input values from the process.

6b.5.8.1 Control lamps and traffic lights

The additional data type 0001B is used for control lamps and traffic lights. Like TIME, there are two modes of operation depending on the GRPTYPE bit.

- If GRPTYPE = 1, implicit and group messages shall work in exactly the same way (without any OR/AND logic).
- If GRPTYPE = 0, implicit messages are used to control lamps with or without automatic flashing and to send data to any display, and group messages are used to control the light level as specified in the LAMPCTRL data type.

Since this special pattern is intended for lamps, GRPTYPE = 0 is by far the most common.

The content of the main register in attribute 2 is shown below depending on the message length:

36-bit implicit, (group) or explicit message. **B20V = 0 or B20U = 0.**					
Any display data	Boolean with flash				Filter time
24 bit	4 × 2 bit				4 bit
0	24	26	28	30	32

20-bit **right shifted** implicit (or group) message. **B20V = 1 and B20U = 1.**							
Any display data	Boolean with flash	Any display data	Boolean with flash				Filter time
8 bit	8 bit	8 bit	4 × 2 bit				4 bit
0	8	16	24	26	28	30	32

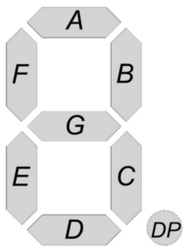
Fig 6b.16

Unlike the TIME data type, this pattern uses bit 32 – 35 to specify the time of the smoothing filter, which may be common to all devices with the same identifier, so that this data type may be used with reception delay. Because of this, it is necessary to program the expected data length by means of B20V and B20U in attribute 11 and 7.

4-bit messages are not used as this would load the same data to all devices with the same identifier if delayed reception is used. If 4-bit control is wanted, use the TIME data type instead.

Because the display data (and the Boolean data) is readable in both bit 0 – 15 and bit 16 – 31 in case of a 20-bit message, it is only necessary with an 8-bit SPI transfer to the display as an SPI transfer always starts with the most significant bit, which is bit 0.

4 bit is enough for a coded 7-segment digit. It is therefore possible to drive for example a traffic lantern with a two-digit countdown display by means of individual 20-bit messages or 16-bits as part of common telegrams. By means of more bits, it is also possible to drive the display segments directly with or without an added "-1" digit (not used in this case) as shown below:



36 bit																																			
8 bit								8 bit								8 bit								8 bit LSD								4 bit MSD			
.	G	F	E	D	C	B	A	.	G	F	E	D	C	B	A	.	G	F	E	D	C	B	A	.	G	F	E	D	C	B	A	.	G	C	B

Fig. 6b.17

Without the dot, 0=3FH, 1=06H, 2=5BH, 3=4FH, 4=66H, 5=6DH, 6=7DH, 7=03H, 8=7FH, 9=6FH. If the 4-bit MSD field is 1000B (just the dot), segment A, B, C, D, E and F should be switched on to show "0.xxxx" instead of just ".xxxx".

In this way, it is possible to drive a standard 4½-digit display or a 4-digit display plus 4 lamps with a 36-bit telegram. In case of 4½ digit, the first, half digit, which is stored in the 4-bit Boolean field, is made by means of segment B and C as "1" and segment G as "-".

To make it possible with automatic, synchronized flashing with two different flash frequencies, bit 24 – 31 is used for a 4 × 2 bit state specification. The data shall be coded as 4 groups of two bits as specified below so that bit 24 and 25 are data for channel 0, bit 26 and 27 are data for channel 1 and so on:

Code	Lamp behavior
00	Off
01	Fast flash (Flash1)
10	Slow flash (Flash0)
11	On

Fig. 6b.18

If the possibility for data to more devices is utilized, the delay between each device should be set to 1 so that for example a single 20-bit message can drive 8 lamps by means of two controllers.

The two flash frequencies are transferred and synchronized by means of global network TIME messages, where bit 32 is used for slow flash and bit 33 for fast flash. A system time telegram may be regarded as a normal Boolean TIME telegram where a timestamp is just added each time one of these bits change state.

If the synchronization of the fast flash (Flash1) stops (goes below 0.5 Hz), it shall be set to an internally generated 2 Hz signal, which is for example suitable for control buttons and control lamps, and if the synchronization of the slow flash (Flash0) stops (goes below 0.5 Hz), it shall be set to an internally generated 1 Hz signal, which is for example suitable for traffic lights and car blinkers. In this way, the safety is preserved.

6b.5.9 LAMPCTRL

LAMPCTRL is a special case of PATTERN used to control lamps. 4-bit data has the following coding:

Code (gray code)	Binary 8-bit level (expansion)	Function	Set lamp to White mode
000 0	**000** 0,00 00	Level = 0% = off if lamp is used for lighting.	Yes
		Close window entirely including ventilation slot.	-
0000		Boolean off (standard level 0).	
		All-off.	
001 0	**001** 0,01 00	Level ≈ 14% linear, 0.8% gamma (24H = 36). Minimum dimming level and night light.	Yes
		Window closed, but ventilation slot is open.	-
011 0	**010** 0,10 01	Level ≈ 29% linear, 4.7% gamma (49H = 73).	Yes
010 0	**011** 0,11 01	Level ≈ 43% linear, 12% gamma (6DH = 109).	
110 0	**100** 1,00 10	Level ≈ 57% linear, 25% gamma (92H = 146).	
1100		Boolean standard level 2.	-
111 0	**101** 1,01 10	Level ≈ 71% linear, 43% gamma (0B6H = 182).	Yes
1110		Boolean standard level 3.	-
101 0	**110** 1,10 11	Level ≈ 86% linear, 68% gamma (0DBH = 219).	Yes
100 0	**111** 1,11 11	Level = 100% = on (0FFH = 255+1).	
		Window entirely open.	-
1000		Boolean standard level 1.	
00 01	Back to previous level and color	RGBW emergency color off. Watchdog update (dummy message).	-
00 11	00 11	Window stop movement.	-
01 S1	New level	White or amber level one step down **if level > minimum (24H)**. Lamp off if level < minimum and input is a (rotary) quadrature encoder.	No
		Window start closing, but stop at minimum level where only the ventilation slot is open.	-
10 S1	New level	White or amber level one step up **if level > 3** (6-bit counter > 0).	No
		Window start opening if ventilation slot is open.	-
11 01	Emergency color	Change temporary to RGBW emergency color	-
11 11	**ABC** A,BC AB	Level = programmed on-level (default 100 %).	Yes
		Open ventilation slot and start opening window, but stop at programmed on-level.	-
1111		Boolean standard level 4 or level 1 – 4 enable.	
		Back-on.	

Fig. 6b.19

As it is the case with the 6-bit counter and bit 0 – 2 of this code, some colors are only defined as a 3-bit or a 6-bit source value instead of the full 8-bit value. In these cases, the source value (ABC or ABCDEF) shall be left shifted according to the big endian model and repeated as many times as necessary and possible to expand the color to 8 bits as shown below and in the bold source part of the 8-bit decoding and expansion above.

ABC source	Copy of ABC	Copy of AB
8 bit		

ABCDEF source	Copy of AB
8 bit	

Fig. 6b.20

The fixed levels are Gray-coded so that only one bit changes at a time. This makes it possible to change the level by means of a simple rotary switch without any flicker between the steps.

The programmable on-level 1111B makes it possible to use a lamp for applications where a lower level than maximum is desirable – for example 7 W for an 11 W lamp. It does ***not*** *prevent that the level later can be increased up to the maximum for that lamp.*

The S-bit is used to select the filter time for 4-bit **group** messages for lamps used for lighting, but may also be used to set the speed for window opening. **It shall have no effect for all other object and message types.** If the bit is 0, the programmed filter time shall be used. If the bit is 1, the filter time shall be set to maximum or the opening speed to slow for up and down commands, **but not for the 9 fixed values** including off, minimum and programmed on-level. This may be used for daylight control to convert the 64 main levels to 256 almost invisible steps.

Since up and down commands only counts the most significant 6 bits up or down, there are in effect only 64 main levels from 0 to 100 %, but during a ramp, the smoothing filter adds more temporary levels between these levels. To make a ramp flicker free, the time between each step must not be more than 1/50 second = 20 ms, and because 4 steps are needed between each main level to convert the effective number of steps to the maximum 256, the up and down commands should not be separated more than approximately 70 – 80 ms. In case of daylight control where it is desirable to change the light level very slow and in very small steps at a time, it may however be desirable to use a much longer ramp time than 80 ms. In this case, the S-bit can be set to 1 and if the time between each command is then at least 2 seconds, there will be 256 invisible steps from 0 to 100 % even though there are still only 64 main levels. Wall switches used to transmit group messages can just use S = 0 and they will then behave in exactly the same way as ordinary lamp switches, but will just affect a group of lamps.

Code 0011B is used to stop a movement like opening of a window.

6b.5.9.1 All-off, Back-on and Emergency/Panic Light

Code 0000B is a general off/0% command. If it is used on group 255 and group 255 is enabled in the group register (attribute 8), it is also used for All-off.

Code 1111B is used for Back-on, which reactivates for example mains outlets and control panels, which has been disconnected or reduced in level by means of a previous group command as specified in the TIME data type, but all other commands for that device shall also terminate All-off. Since 1111B activates the programmed on-level for lamps and not the previous light level, Back-on shall **not** be enabled in case of lamps used for lighting as it would cause all lamps to turn on!

Code 1101B is used **for lamps** to **temporary** set the RGBW emergency color, which is the same color as used in case of a watchdog timeout. This is intended for a panic button, which may be activated for example in case of fire or if some noise could indicate that there is a thief in the house. It shall therefore not depend on whether group 255 is enabled or not. The original light level and color shall **not** be changed while the emergency light is on.

Code 0001B is used to switch the color back if another light command has not changed it before that. Since code 0001B just switch the light back, which has no effect if the emergency light is off, it may also be used as a dummy message to keep the watchdog updated. If the control system, which sends 0001B update commands, receives a 1101B command, it shall stop sending updates until a 0001B command is received form another device to prevent that the panic light is switched off.

The watchdog may for example be enabled on circadian lighting in nursing homes where the light can both be controlled by the resident by means of implicit messages and by the nursing staff or automatically by means of group messages.

Note that there are only these four (4-bit) commands for group 255 and long group-255 messages shall therefore always be ignored. To be able to use these commands together with more lamp and outlet groups, group 255 shall always use the data type LAMPCTRL – even if it is used to control an outlet, which uses the data type TIME, and it shall have its own group ID check, which is 0000110B (not the group ID-check specified in attribute 8).

All-off and Back-on are only relevant if they are (also) used to control lamps. If it is only necessary to control outlets and/or energy management, ordinary group messages with the data type TIME offer more possibilities in the form of 4 levels and possible timestamp. This is the reason why All-off and Back-on use the data type LAMPCTRL.

6b.5.9.2 Use with GRPTYPE = 0

In many cases GRPTYPE = 0 is used to make it possible to control the light level in a lamp or in a display. In this case, 20-bit and 4-bit messages shall cause the following content of the **light level** register in attribute 2 as a function of the message:

20-bit group message.		
NU	Direct light level	Filter time
0000,0000	8 bit	4 bit

Not changed	Not changed	Not changed	Direct light level	
8 bit	8 bit	8 bit	6 bit counter	2 bit
0	8	16	24	

Group message.
LAMPCTRL
4 bit

Not changed	Not changed	Not changed	Light level decoded from 4 bit LAMPCTRL	
8 bit	8 bit	8 bit	6 bit counter	2 bit
0	8	16	24	30

Fig. 6b.21

6b.5.9.3 Lamps for Lighting

When the subscriber data type is LAMPCTRL, the lamp is used for lighting.

If GRPTYPE is the default 1, group messages work in exactly the same way as implicit messages as described in the specification of attribute 12 in layer 6a. A 4-bit message, which contains a fixed value (**not** an up or down command), shall be decoded and set an RGBW lamp in white mode so that red, green and blue becomes a copy of white. White mode is not used in case of an RGBA lamp.

If GRPTYPE = 0, group messages **except for All-off on group 255** shall only affect the light register as it is the case for all other data types, so only the white or amber channel must be affected no matter the message length (4-bit, 20-bit or 36-bit). This may be used for daylight control of an RGBW or cWwW lamp, but it makes it impossible to control the color by means of long group messages.

The 4-bit, group 255, Back-on command shall be disabled for lamps used for lighting!

Daylight control may be very useful to save power if the sun comes out. Because of the logarithmic behavior of the eye, what looks like a factor 2 in light reduction, actually reduces the power by 82%, so automatic dimming may save a lot of energy! It is also very useful to dim control panels.

Note that if GRPTYPE = 1, up and down messages may still be used for daylight control as they do not set the lamp in white mode, but it is not possible to change the white channel alone by means of a long message.

The content of the light level register of attribute 2 due to the various message types except for 4-bit group messages are shown below:

1 – 2 color tunable white lamps (W, cWwW, dim-to-warm etc.). **B20V = 1** Right shifted 20-bit implicit data or group data.					
= Cold White **NU** (Red output = lamp-off-status)	= Cold White **NU** (Green output = lamp-on-status)	← Cold White (R=G=B=>white)	Warm White, White or Amber		Filter or function
8 bit	8 bit	8 bit	6 bit up/down counter	2 bit	4 bit
0	8	16	24	30	32

3 – 6 color lamps (RGB, RGBW, RGBA, RGBWaAaC, RGBAaC etc.). **B20V = 0.** 36-bit implicit, explicit or group data.					
Red	Green	Blue	White or Amber		Filter or function
8 bit	8 bit	8 bit	6 bit up/down counter	2 bit	4 bit
0	8	16	24	30	32

All types of lamps, **4-bit implicit data** for up/down control and 9 fixed levels					
Not changed, but set to white mode	Not changed, but set to white mode	Not changed, but set to white mode	White level from 4-bit code ←		Up/down or fixed
8 bit	8 bit	8 bit	6 bit up/down counter	2 bit	4 bit
0	8	16	24	30	32

Fig. 6b.22

In case of 1 – 2 color lamps, the received data shall be right shifted two bytes to locate blue/cold-white and warm-white/amber in the right positions, and even though 1 – 2 color lamps do not use red and green (the two outputs instead used to show the status), they shall be set to a copy of blue so that a read of attribute 2 always returns the actual color no matter if it is a 1 – 2 color lamp or a full color lamp and no matter any later daylight control.

*The reason why the white or true amber bits are the least significant is that it is a de-facto DMX512 standard that red is always the first byte of a color specification no matter if it is an RGB, RGBW or RGBA lamp. Red shall therefore have the specified position followed by green and blue – also **when a delay is specified**. This makes it possible to transmit RGB-only data to more devices in the same message by means of N × 3 byte delays.*

The fixed levels are very useful to switch a lamp back from full-color mode to white mode without too much change in light intensity. By means of the level of the RGB and W or A channels, it is possible to calculate a corresponding white level so that the nearest fixed level can be selected. For a typical RGBW system, a sufficiently good approximation for the luminance may be calculated as: (2R + 5G + 1B + 8W)/16, and for an RGBA system, (2R + 5G + 1B + 2A)/10 may be used. Since the difference between each fixed level is approximately 14%, it is always possible to keep the change in luminance below approximately 8%.

In case of **long** messages for lamps (implicit messages and 20-bit group messages), the 4-bit [Filter or function] field does not contain a LAMPCTRL value, but is instead used to set the time constant of a smoothing filter or to select up to 6 other functions (up to 24 DMX channels plus up to 4 for light) like lamp direction (pan/tilt), movement speed, zoom/focus, gobo index/rotation, special effects etc. as specified and shown below with orange background color:

4-bit Code	Steps per second	Approximate time to			Maximum time to	Lighting application (all times may also be used for inrush current limiting).
		50, 63 (T) and 75 % linear			100 % linear	
(0000)	10,000	0.4 ms	**0.54 ms**	0.7 ms	2.5 ms	High speed (stroboscopic) lamps. Digital to analog conversion.
0001	2000	2 ms	**2.7 ms**	3.5 ms	12.5 ms	Stroboscopic lamps for stage light.
0010	1000	4 ms	**5.4 ms**	7 ms	25 ms	Smoothing of 50 Hz or 60 Hz video interlace scan.
0011	500	8 ms	**10.8 ms**	14 ms	50 ms	
0100	250	16 ms	**21.6 ms**	28 ms	100 ms	Conversion of 25 Hz or 30 Hz video flicker to more pleasant motion blur.
0101	125	32 ms	**43.2 ms**	56 ms	200 ms	Control lamps and traffic lights. Simulation of incandescent lamps.
0110	62.5	64 ms	**86.4 ms**	112 ms	400 ms	Lamps for lighting.
0111	31.3	128 ms	**173 ms**	224 ms	800 ms	Change of scenes (may create flicker).
1000	15.6	256 ms	**346 ms**	448 ms	1600 ms	
1001	7.8	512 ms	**692 ms**	0.88 s	3200 ms	Daylight control.
Implicit messages and …						
Group messages with GRPTYPE = 0						Group messages with GRPTYPE = 1
1010	4.0	1 s	**1.35 s**	1.75 s	6.3 s	Free function for future use
1011	2.0	2 s	**2.7 s**	3.5 s	12.5 s	Free function for future use
1100	1.0	4 s	**5.4 s**	7 s	25 s	Free function for future use
1101						Free function for future use
1110						Free function for future use
1111						Pan and tilt

Fig. 6b.23

If GRPTYPE = 0, a **group** message is only used to change the white or amber channel and all codes are therefore used for the filter time.

This makes it possible to set an automatically generated approximately 8 second long ramp time for daylight control by means of a 20-bit message with an analog value if the value difference is so small that each filter step is (almost) invisible. At low levels, the value difference may not be more than 8 to keep each step invisible, but at high levels, it is possible with a bigger difference. Note that If it is necessary to select one of the 3 longest filter times with GRPTYPE = 1, this can be done by programming the wanted time as the default one and then use code 0000B to select it.

Each color including artificial generated channels like amber and cyan shall have its own smoothing filter.

In case of data types like TIME, where the smoothing time is fixed to the programmed value and therefore cannot be changed on a per telegram basis, all codes shall be used for the programmed filter time. This allows inrush current limiting with a time constant up to 5.4 s.

Note that code 0000B selects the programmed speed – not the actual speed, so if a T = 0.54 ms filter is needed, the programmed speed must be set to 0000B. In this way, all speed codes are utilized and it requires an intentional act to enable flash pulses shorter than 20 ms.

The 0.54 ms filter works as a kind of filter bypass, which however still preserves enough inrush current limiting to reduce the necessary decoupling capacitors considerably for most applications. It is the fastest possible filter where even the lowest level is still accurate at a 100 % strep with gamma correction.

Stroboscopic lamps may for example be used for stage light and for burglary alarm systems, which destroys the night vision of a thief and makes it very unpleasant for him. In case of the RGBWaAaC color system, the white channel can be used for a high power strobe and the RGBaAaC channels can be used to generate a colored aura.

Dynamic filter selection on a per-telegram basis may be very useful for a lot of applications. For example, a heating element may need a very long on-time to limit the inrush current and this time can just be programmed in the device and selected in an on-message by means of the 0000B code, but when the heating element is switched off, a 2.7 ms filter may be enough to reduce the line ringing and voltage overshoot to an acceptable level.

Different on and off times may also be used together with the multiple master possibilities of Max-i to control stage lamps in real time from for example the various drums in a drum set. In this case, a single message with a short time followed shortly after by one with a longer time may simulate the sound level fairly close with only two messages. It is also very useful if a lamp is controllable from many sources with different time requirements. For example, lamps, which are controllable from wall buttons, should typically be programmed to T = 86.4 ms to give a smooth change of light intensity without flicker even though only 12 up or down messages per second is used, but if a scene is set or changed by means of for example a smartphone, T = 173 ms may be utilized to give a slower turnover. The slight flicker, which may be visible at low levels at this speed, may not be visible in this case due to the light intensity of the new scene. The same lamps may also be controlled by means of a TV set in real time and in that case, the short times may for example be used to flash the lamps to simulate lightning during thunder weather and to synchronize the light to music.

If numerous lamps are used to form a video display, the filter may be used to reduce the flicker and enable a lower frame frequency. In case of 25 Hz or 30 Hz video, the 21.6 ms filter will generate approximately 250 pictures per second, and the time constant ensures that after half a frame, the color is close to a 50% blend between the two subsequent frames. This makes the display pleasant even for dogs and other animals, who can resolve over 75 pictures per second. For 50 Hz or 60 Hz video, the maximum settling time to 99% of the 5.4 ms filter is very close to one frame (20 ms or 16.7 ms).

The time requirements in the table may make it necessary with a 2nd or 3rd order filter. If a digital filter is used, the step size shall be less than 1/8 = 12.5 % so that the minimum level is 31. With that level, the minimum frequency for linear inrush current limiting becomes 25 kHz, and a relatively small filter coil can be used. If for example a 24-V, 600 W heating-element, which has an internal resistance of 1 Ω, is driven from 20 V (maximum 20 A = 400 W), a 22 µH serial inductor is enough to limit the maximum current variations to 6 % of the maximum current.

Note that in case of a simple infinite impulse response filter there may be a very long tail before the final level is reached – in principle infinite, but to ensure maximum accuracy of a D/A converter within a reasonable time, no more than 4.6 T shall be necessary.

With a pulse width of 1.25 µs, the minimum frequency with linear dimming is 3.125 kHz, and with a gamma of 2.44, it is 70 Hz, which ensures that even the lowest lamp level is flicker free.

Note that longer smoothing times than 86.4 ms may create lamp flicker if used with the Max-i controller, so it is highly recommended to leave the filter speed at this to create a smooth ramp with the approximately 12 up/down telegrams per second, which are transmitted! The longer times are however included for inrush current limiting and for microprocessor controlled lamps, which are able to calculate a smooth ramp and changeover from one scene to another.

In case of daylight control by means of group massages, the time constant of the digital filter may be increased to maximum by means of the S-bit in the up and down commands as specified above.

In summary, lamps for lighting may be controlled in 4 different ways simultaneously with the same settings:

- By means of 4-bit implicit messages from for example buttons on the walls. This may set the color to white, turn the lamp on and off, set it to night level and step the light level up and down.
- By means of 4-bit group messages, which control the level of white and may also be used for all-off. If the lamp is in full-color mode, this will change the color saturation (RGBW) or color temperature (RGBA).
- By means of a 36-bit full-color or a 20-bit tunable-color implicit message to an individual lamp from for example a smartphone app (reception delay possible, but usually not used).
- By means of a common group telegram with individual reception delays to set a scene from for example a smartphone app or to control more lamps in real-time from for example a TV-set.

6b.5.10 Standard Functions

For the moment only a single function has been defined.

Tilt and Pan			
Tilt	Pan	Flight time (ds)	1111
12 bit	13 bit	7 bit	4 bit
0	12	25	32

Fig. 6b.24

The 13-bit pan and 12-bit tilt specification in signed integer format where 0 is neutral allows a ±0.05° accuracy (±0.9 mm/m) op to ±410° for pan (often limited by the lamp to ±270°) and ±205° for tilt (often limited to ±135°), and the 7-bit time specification in deciseconds allows flight times from 0.1 s up to 12.7 seconds.

The flight time makes it much easier to control the movement from one spot to another than a specification of the movement speed as it is often seen, and the connected microprocessor has nothing else to do than controlling step motors and/or servo motors as the Max-i controller handles all message reception and verification and only forward relevant information through the SPI interface.

The various functions make it unnecessary to send all information all the time to multi-function lamps as in DMX512. This can save a lot of bandwidth. 6, 4-byte functions plus the light information with smoothing filter corresponds to 28 DMX channels. Many advanced DMX512 lamps use more, but this is usually for macros and automatic functions, which anyway do not follow the music. With Max-i, these functions can much better be handled by a separate controller and only transmitted when necessary, and if 28 channels are not enough, a nibble or a byte of one of the 6 functions may just be used to select a sub function.

Layer 7, Application Layer

This Layer defines the standardized I/O objects.

There are two groups of objects – standard objects defined in this layer and user objects defined by the manufacturer.

All standard objects use up to 8 inputs (input 0 – input 7) and 9 outputs (output 0 – output 8). The controller shall therefore have this minimum number of I/O.

Each I/O object has a name and a number plus a size, which describes how many blocks of 16 attributes are needed beyond the standard block. The number and the extension size are specified in the I/O object specification attribute together with the data type for the publisher and the subscriber, which describes the behavior of the inputs and outputs and makes it possible for an object to use the I/O in different ways. For example, the input of the LAMP object may either be used as boolean input (Data type TIME) for example for fire and burglary alarms or to generate the light control sequence (Data type LAMPCTRL), and the outputs of the ADDA object (Analog to digital and digital to analog conversion) may either be a boolean output (Data type TIME) or an analog output (Data type RAW). Further settings are done in I/O attribute 1 – 3, which defines the behavior of an object, and in extension blocks in case three attributes are not enough. In most cases, bit 1 – 15 of I/O attribute 1 – 3 are used for publisher programming and bit 16 – 31 are used for subscriber programming.

All standard I/O attributes except for attribute 14 and 15, which are used for a vendor code and therefore does not need protection, use bit 0 as a parity check.

I/O attribute 2 and 3 and the I/O object specification attribute, which are both protected by password 3, are used to define the operation mode of a device and shall therefore be programmed by the vendor. In most cases, it is therefore only I/O 1, which is user programmable, so there is very little for the user to warry about.

7.1 Inputs

In most objects, the input function or combined I/O function is programmed in I/O 1 and I/O 3 as shown below:

I/O 1 (password 2 – user programmable)									
Parity	HBF	DBT	B0T	B1T	B2T	B3T	(B4T)	(B5T)	
1 bit	2 bit	1 bit	2 bit	2 bit	2 bit	2 bit	2 bit	2 bit	
0	1	3	4	6	8	10	12	14	16

I/O 3 (password 3 – factory programmable)															
Parity		DBL	B4ES	B5ES	B6ES	B7ES	B0D	B1D	B2D	B3D	B4D	B5D	CPOL	CPHA	
1 bit		1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	
0	1	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Fig. 7.1

In the following, standardized bits for the publisher are marked with this light green background color, standardized bits for the subscriber are marked with light orange and standardized bits used for both are marked with light yellow (a mix of green and orange).

The **HBF** (HeartBeat Flash) bits are used to generate the two most significant Boolean bits automatically by means of the heartbeat timer in attribute 5 if the publisher data type is TIME and the I/O object is either MODEM or BOOLEAN. This is primary intended for generating flash synchronization by means of system time network telegrams with ID = 00 00 hex, but may also be used to generate other synchronization signals.

Bit 1 of the 4-bit Boolean value shall always be a square wave with a 50 % duty cycle, and the heartbeat timer in attribute 5 must be programmed to a period time corresponding to the necessary number of telegrams per second for this. If for example 3 Hz is wanted, it is necessary with 6 telegrams per second so the heartbeat timer must be set to 167 ms. The heartbeat time is usually the time between telegrams and may therefore be increased if the bus is heavily loaded, but for this application, it shall be a free running oscillator to get maximum accuracy. This also makes it possible to measure the clock accuracy.

Bit 0 (MSb) is the waveform of bit 1 divided by 2, 3 or 4 as specified below:

Code	Divider
11	No automatic signal generation. Physical inputs used for bit 0 and 1.
10	2
01	3
00	4

Fig. 7.2

B0T – B5T are used to specify the triggering method. B4T and B5T are shown in brackets above because these inputs are only available in some objects like the keypad scanner and in these cases, the triggering method may be fixed and independent on the programming of these bits.

Each input may be programmed in four ways:

Code	Function
00	No trigger.
01	Trigger of boolean value when input goes low, but not when it goes high (off-only).
10	Trigger of boolean value when input goes high, but not when it goes low (on-only).
11	Trigger of boolean value when input changes state (goes high or low).

Fig. 7.3

In case of network time messages, these bits must be left in the default 11B state to ensure that all edges generate an event and with that a transmission.

Note that the time shall only be recorded when a message is transmitted. In this way, the time corresponds to the important edges, but there is no time recording for poll-only values.

11B is the default setting where any change in input value causes a telegram. If input 0 – 3 are programmed to 00B, it is necessary to poll the value. This setting may be useful to avoid collisions between feedback signals in case of more subscribers for a value like for example traffic lights.

B4ES – B7ES of I/O 3 are used to specify whether input 4 – 7 shall behave like an error input, that is, generate an error message when it goes high and be latched until it is read (default 1), or shall just show the present status of the input (set to 0) when the error word is polled (no event driven error message).

The error or status bits are read as bit 32 – 35 of the error attribute. In case of for example lamps with many color channels, a single error input may be used for a common error signal and the other error inputs may then be programmed as status indicators similar to input 0 – 3 so that the status of up to 7 color channels can be read.

To prevent wrong or multiple triggering due to noise and/or contact bounce, all Max-i inputs may have three options:

- No filtering. This shall be a possibility on input 0 – 5 for high speed signals and electronic contacts and sensors. This mode is selected by setting **B0D – B5D** low and in this way disable the debouncer. These bits shall be ignored or may be used for other purposes in case of inputs, which are always high speed inputs like UART and SPI data inputs.

*I/O objects, which have signals, which are **always** high speed like clock signals and UART and SPI communication, may simply take these directly from the inputs.*

- A noise filter on the rising and falling edge of input 0 – 7 followed by a debouncer, which ensures that subsequent edges cannot occur before a given time after a previous edge as shown below:

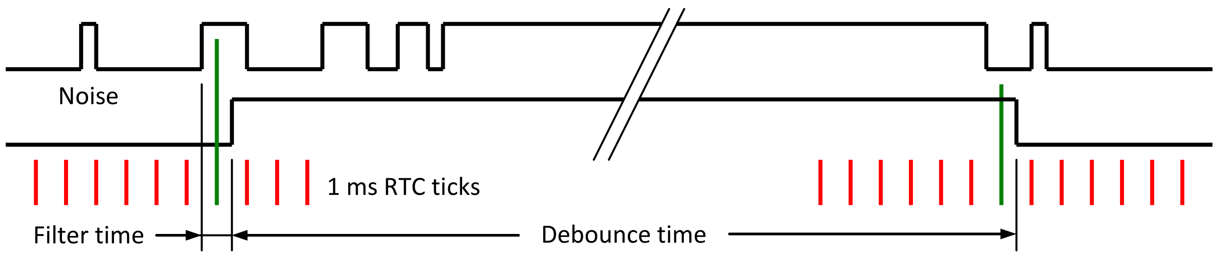


Fig. 7.4

Most mechanical switches bounce when they are activated and may even bounce when they open. However, it is reasonable to assume that the first change in input state with duration more than the filter time is the real edge and that following pulses are just caused by the bouncing. This is a much better principle than the usual slow input filter, which delays the response time and makes it impossible with a precise time recording. It also makes it much easier to fit the circuit to various needs just by changing the filter time and/or the debounce time.

The filter time in this mode shall be 0.25 – 1.25 ms, and in case of a sampled noise filter, the filter shall be synchronized with the real time clock (RTC) so that the output of the filter causes time recording at least 0.25 ms before the RTC can change again as shown above. In the drawing, the recorded times are the two long green RTC tick so that in spite of the filter, the accuracy of the time recording is not affected.

The mode is default and selected if the **DBL** (Debounce Lamp) bit is the default 1.

- A status monitor or error detector for dimmable lamps or inrush current limiting where the outputs may consist of a train of 1.25 µs PCM pulses with a frequency of at least 65 Hz. In this mode, the filter time must be reduced to 0.25 – 0.5 µs to be able to detect the narrow pulses. This mode is only selected if the publisher data type is TIME and the **DBL** (Debounce Lamp) bit is set to 0.

Because a LED glows visibly at even a **very** small current, it is not possible to monitor it by means of a small quiescent current, so a LED must be regarded as on (1-state) if only a single 1.25 µs pulse is received within the debounce time as shown below:

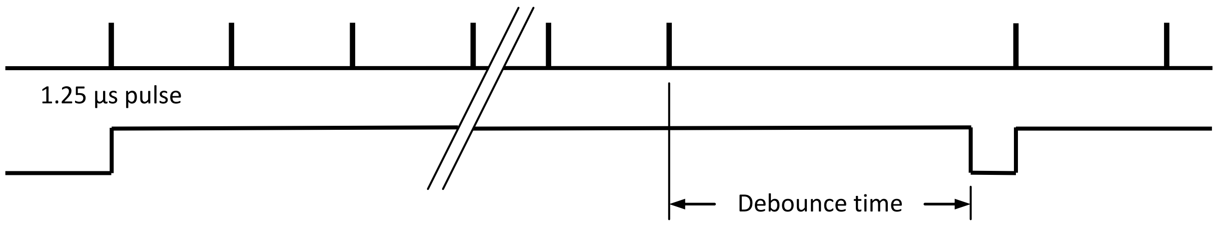


Fig. 7.5

The output must go high as soon as the first pulse is accepted and stay high until there have been no pulses for the debounce time.

The standard debounce time for both modes shall be 30 – 45 ms, with fits well with mechanical contacts and dimmed lamps with PCM (or PWM) down to 40 Hz, but there are two situations where it must be changed:

- In case of for example a quadrature encoder, which may generate a sequence of fairly short pulses. In this case, it may be reduced to 3 – 4 ms by setting the **DBT** (Debounce Time) bit in I/O 1 low, and it should also be done automatically if the quadrature decoder is selected for lamps or other objects need it.
- If one or more of input 4 – 7 are used for error messages. In this case, it shall be increased to 30 – 45 s to prevent that for example a defect (traffic) lamp generates a new error message every time an attempt is made to switch it on or make it flash and in this way generates an inconvenient high number of error messages. This is controlled by the **B4ES – B7ES** bits. If one or more inputs are in the default error mode (1), the long debounce time shall be used. If the bits are 0 (status), the time shall be controlled by the **DBT** bit.

Note that if an object such as LAMP used as a wall switch does not use bit 4 – 7 for error messages, it is necessary to set B4ES – B7ES to 0 to prevent a long dead time on the corresponding inputs!

Without this very long debounce time, it would be necessary to synchronize a lamp error input with either the 1.25 µs output pulses or the logical lamp signal in case of the BOOLEAN object, but this would dedicate an input to a particular output, which may not be convenient or may even be impossible with for example the LAMP object, which may have many more color channels than the number of error inputs. If the duty-cycle of a lamp is more than 30 seconds, which is often the case for control lamps and signal towers, an error will be generated each time an attempt is made to activate a lamp after the debounce time, but in this case, it is nice with a reminder that the lamp is not working as the missing light could otherwise lead to a wrong assumption about the process status.

7.2 Outputs

Each of the 4 main outputs shall be programmable to either:

- A pure boolean output with only two states – 0 (off) and 100 % (on).

  This type is for example used for UART and SPI interface and for low-power status signals.

- A boolean outputs with limited rise- and fall time with **linear** ramp and a **common, fixed** on-value equal to 100 % (no bin-correction). The typical data type is TIME.

  This type may be used for inrush current limiting and to reduce the necessary decoupling capacitor as described in the chapter "Load Switching" in layer 1.

- A boolean lamp output with a **common, variable** on-value for example for control lamps. In this case, the data type for group messages must be programmed to LAMPCTRL.

  This type may for example be used for control lamps and traffic lights with or without automatic flashing. It is also used to control the intensity of displays where the light is always on. In this case, the white level is just routed to output 6 instead of 3, and the display data may then be transferred by means of the SPI or MODEM object, or a 7-segment digit may be directly driven by the 4 boolean bits.

  A change between the selected on-level (logical 1) and 0 = off may easily be obtained by means of the already present reset signal of attribute 2 (consumed value). This saves gates and may be used because a reset of any parts of attribute 2 even only 8 bits prevents it from holding the time, which requires all 32 bits, so a variable on-value is anyway not useable with the data type TIME.

- An **individual, variable** value between 0 and 100 %.

  This type is used to drive monochrome, dichromatic (tunable white), dim-to-warm and full color lamps used for lighting. The data type is LAMPCTRL, and gamma correction (2.44) is usually used. Since there is only a common lamp on and off, automatic flashing cannot be used and shall be disabled.

In most objects, the output function is programmed in I/O 2 and I/O 3 as shown below:

I/O 2 (password 3 – factory programmable)		
	Minimum light level	
	6 bit	
0	16	22

I/O 3 (password 3 – factory programmable)										
	SPILength		OutSel	B20V	Gamma if GRPTYPE = 0	Filter	B0L	B1L	B2L	B3L
	NU (0)	3 bit	2 bit	1 bit	1 bit	4 bit	1 bit	1 bit	1 bit	1 bit
	16	17	20	22	23	24	28	29	30	31

Fig. 7.6

In the following, these standardized bits for outputs are marked with this orange background color.

The **Minimum light level** is only used in case of control lamps and traffic lights, but not for lamps used for lighting. It is extended to 8 bits by copying the two most significant bits to the two least significant as specified previously.

The **SPILength** is used set the number of nibbles in a message to a connected device. It is for example used in the SPI object, the BOOLEAN object and in the LAMP object.

The **OutSel** bits are used to select between various output configurations depending on the object function.

The **B20V** (vendor) bit is used to select the (minimum) message length according to the I/O possibilities of the device. If it is the default 1, the expected message length is 20 bit, if it has not been changed to 36 bit by the B20U (user) bit in attribute 7, and if it is 0, the message length is always 36 bit.

The **Gamma** bit is used to select between a gamma of 2,44 (default 1) and linear (0). It is used in case of lamps for lighting and control lamps and for output 6 in case of GRPTYPE = 0. The gamma is always linear when the smoothing filter is used for inrush current limiting; but since the power is proportional to the voltage in the power of two, the gamma in effect becomes 2 in this case.

The **Filter** bits are used to select the default rise and fall time of the smoothing filter according to the specification of the LAMPCTRL data type.

Bit **B0L – B3L** are used to select the function of the 4 main outputs. If a bit is the default 1, the corresponding output shall be pure Boolean with very fast rise and fall time, and if the bit is set to 0, the output should instead be generated by means of a pulse code modulator (PCM), which generates a mean-value by means of a train of pulses each with a width of 1.25 μs. **If a slew rate limited signal is wanted or a channel is used to drive (variable) lamps, the corresponding bit must therefore be set to 0!**

PCM generates a sequence of 1's and 0's where PWM just change the duty-cycle of a fairly low frequency. For example, 17/32 is generated by means of the repeating sequence 10101010.10101011.10101010.10101010 where PWM instead generates the repeating sequence 11111111.11111111.10000000.00000000. It is obvious that except for the minimum value 10000000.00000000.00000000.00000000, which has the same pulse width and frequency, PCM has a much higher fundamental frequency and gives a much more smooth and flicker free signal without stroboscopic effects, so that much smaller decoupling capacitors and smoothing coils can be used and that the thermal stress on LED's due to alternating heating and cooling is much smaller. Besides, PCM does not violate the Signify (Previous Philips Lighting and Color Kinetics) PWM patent for color changing lamps with communication or the similar Infineon sigma-delta patent.

If a power switch is fast enough to handle the 1.25 μs pulses, the PCM outputs may be connected directly to inductive and resistive loads, but if the load is inductive or a smoothing inductor is added, a fly-back diode or TVS must be used to reduce the fly-back voltage when the switch turns off.

Note that to utilize the advantage of the small decoupling capacitor for PCM, any changes in light intensity must be done smoothly over at least 5 ms – see chapter "Load Switching".

The PCM shall be able to code an output in three ways depending on the data type:

Gamma	Data types	Typical application
1.00 (linear)	LAMPCTRL	Control of light dimmers (typical 0 – 10 V) and incandescent lamps.
	RAW	Digital-to-analog (DA) converters.
	TIME	Slew rate limiting of boolean outputs for reduction of line ringing or heavy inrush currents from heaters, incandescent lamps and motors.
≈ √2.44 ($x^{\sqrt{2.44}}$) = 1.56 ($x^{1.56}$)	LAMPCTRL	Dim-to-warm outputs for artificial amber and white. Possible increase of gamma to 4 for high dynamic range white.
≈ 2.44 ($x^{2.44}$)	LAMPCTRL, PATTERN	Ordinary LED outputs.

Fig. 7.7

With a gamma of 2.44, 8-bit linear data is converted to 13½ bits resolution corresponding to a contrast ratio of approximately 11,000:1. This increases the efficiency a lot.

Without gamma correction, it would be necessary with at least 13 bits per color instead of 8, and all transmitters should be able to perform the gamma correction.

A gamma of 2.44 corresponds closely to the sensitivity curve of the eye as explained in Annex C, LED drive. To ensure that light colors can be dimmed without any noticeable change in color hue, the accuracy of the gamma curve should be better than ±1%.

The sensitivity curve of the eye has been determined to be R = 9.033i for i ≤ 0.008856 and R = 1.16×(i^(1/3))-0.16 for i > 0.008856. For most practical applications such as TV sets, computer monitors and LED lighting, this may be simplified to R = i^(1/2.44), and for a reference monitor it has been standardized to R = i^(1/2.4) according to ITU-R BT.1886. With the 8-bit input N = 127 (half), linear dimming gives a light intensity of 49.8 % (127/255), which the eye perceives as R = 76 %, and with a gamma of 2.44 as implemented in Max-i, the light intensity becomes 19.2 %, which the eye perceives as R = 51 %. This is very close to 0.498^2.4 = 18.8 % for a reference monitor. With exponential dimming as approximated in DALI, CAN FD Light and many LED-driver IC's, the intensity is 3.16 %, which the eye perceives as only R = 21 %! For DALI it is: i = 10^(((N-1)/(253/3))-1) = 3.12 % => R = 21 % and for CAN FD Light it is: i = 0.9471^((255-N)/2) = 3.09 % => R = 20 %.

Without a gamma correction close to the eye curve, the color hue and the duty cycle of a flashing lamp is changed as shown below:

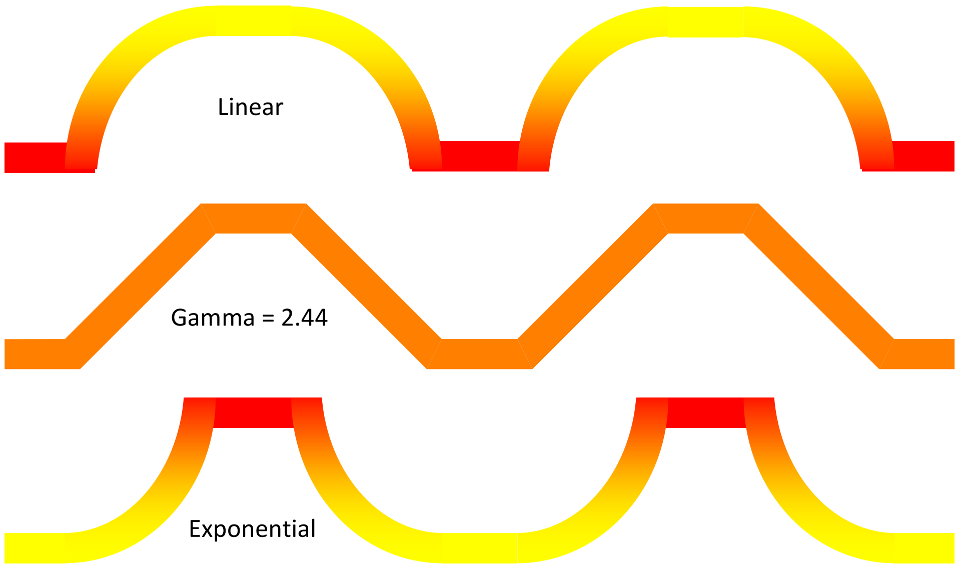


Fig. 7.8

If the gamma is lower than 2.44 – for example linear, a medium level orange color, which consists of more red than green, may seem reddish to the eye if the level is reduced and yellowish if the level is increased, and in case of flashing with a smoothing filter, which creates a ramp, the duty cycle seems to be increased. This is not acceptable for example for control lamps, which typical use a 50 % duty cycle. I case of a gamma higher than 2.44 such as for example exponential dimming, the opposite is the case.

To be able to implement dim-to-warm in an easy and cheap way, lamps used for lighting shall have an auxiliary output (Output 6), which follows the white channel, but has the square root of the selected gamma, that is, 1 if linear is selected, and approximately 1.56 if 2.44 is selected. If this output is connected to amber or very warm white LED's, the color hue can be made to follow the Kruithof Curves and therefore be pleasing to the eye no matter the illuminance as explained in Annex C, LED Drive, which also describes gamma even further.

Note that when driving heaters, incandescent lamps and motors, the electric power is proportional to the voltage squared ($P = U^2/R$) or the current squared ($P = I^2 \cdot R$), so such devices in effect get a gamma of 2 and therefore do not need additional gamma correction. It is therefore only objects, which use the LAMPCTRL data type, which need a programming bit to select between linear and gamma correction.

⚠ **No matter if the gamma setting is 2.44 or linear, 100% light intensity (input data = 255 = 0FFH) shall generate a continuous signal with no pulses if bin correction (described later) is not used and 0 shall turn the output entirely off.** This shall also be true for a Gamma-1.56 output. In case of linear dimming, the maximum level (255) shall also generate a continuous signal without pulses, but to achieve this and simultaneously keep the curve linear down to 0, it is necessary to add 1 to the maximum value 255 so that the sequence becomes 0/256, 1/256, 2/256, 3/256 …. 252/256, 253/256, 254/256, 256/256 (**255/256 missing**). This creates a small discontinuity at the maximum level, but it is invisible and it ensures a small overdrive/saturation of a 0 – 10 V signal where 255/256 corresponds to 100 % so that 100 % is reachable even in case of small tolerances.

7.2.1 Transmitter active output(s)

In all objects, which do not need all 9 outputs for their function, output 7 is used as an acknowledge output, which should generates a 50 ms pulse when a telegram has been transmitted successfully, that is, the whole telegram including signature. Aborted telegrams due to collisions shall not activate the output.

In case of the KEYPAD object, this output and output 8 may be AND’ed with for example a square wave, a synthetic sine wave or a more advanced “click” signal so that the outputs may drive a piezoelectric loudspeaker (beeper) directly.

Layer 7a, Default MODEM object

This object is used to communicate with all kinds of microprocessors, computers, printers, information displays etc.

Object name and ID #	MODEM, -1 (11111111B).
Publisher data type(s)	UNDEF or TIME (Network time message)
Subscriber data type(s)	UNDEF or any.
Input 0	Serial data In (SI).
Input 1	Invert SI.
Input 2	Invert SO.
Input 3	Invert CTS.
Input 4	Error or Status 0 input.
Input 5	Error or Status 1 input.
Input 6	Error or Status 2 input.
Input 7	Error or Status 3 input.
Output 0	Serial data Out (SO).
Output 1	Clear-To-Send (CTS).
Output 2	X8-clock output (X8CLK).
Output 3	X4-clock output (X4CLK).
Output 4	NU
Output 5	Relative bus load.
Output 6	Lamp level.
Output 7	Transmitter active pulse.
Output 8	Collision.

Fig. 7a.1

The communication shall use standard NRZ (UART) communication with an idle state of 1, one start-bit (0), 8 data bits with least significant bit first and 1 stop-bit (1). There is no parity bit because there is no time for it at the lowest 7 baud-rates and usually no need for it either as the distance between the Max-i controller and the connected device is usually very short – often just a galvanic separation. However, if a connected device is able to perform the scrambling and descrambling, it shall be possible to switch the automatic scrambling and descrambling off and in this way use all 20 check bits for error detection.

All messages (received and transmitted) shall be separated by a Break condition where the line goes low (0) for at least a time corresponding to 10 bits (start-bit, data bits and stop-bit). Break generation and detection must therefore also be supported by the UART in the connected device.

Since 0-bits and 1-bits are not equally long on the bus and it may be busy, the UART's shall be able to perform automatic flow control by means of a CTS signal, which can hold back new bytes to be transmitted.

To be able to retransmit at least a short identifier in case of a collision during bus arbitration, the UART shall contain at least a 3-byte FIFO where the CTS signal does not go low before two bytes are loaded. For a 3-byte FIFO, CTS should go low when the start-bit of the 3rd byte is accepted in the middle of the start-bit.

*A 3-byte FIFO may for example be implemented by means of 2 holding registers plus the shift register of the UART. Note however that the break generation of most UART's in connected devices unfortunately does not obey CTS, which may cause an overload condition (the Max-i controller regards a break condition as any other byte). Therefore it is recommended to generate the Break condition **before** the message and then check that the last byte has been transferred to the Max-i controller (TSRE = 1) and CTS is high before the next message with a new Break is transmitted! Unfortunately, no UART's are so clever made that they just regard the Break as any other character and therefore does not transmit it before the data FIFO is empty and the receiver is ready (CTS high).*

In case of an overload, the Max-i controller shall:

- Finish the byte, which is being transmitted on the bus, so that the number of bits in all telegrams including aborted ones is always N×8 + 4.
- Stop the transmission.
- Generate an overload error message when the bus becomes idle.
- Empty all buffers in the transmitting device including its UART by setting CTS constant high until a break condition is received (telegram delimiter). During this period, no data must be loaded into the FIFO or be transmitted (except for the error message).

Since the Max-i controller only works as a modem, the published and consumed process data are usually **not** stored in the controller, but in the connected device. The controller must therefore not issue **implicit** messages and answer polls for the published value, but it shall still be able to receive group messages, which may be used to control the light intensity of a display, and it shall still answer group polls and be able to receive and transmit **explicit** messages including the error word and attribute 2. Note however that since the consumed value is also located in the connected device, a poll for attribute 2 may or may not return valid data, but this may be indicated with the data type UNDEF if the data is undefined.

To be able to program a device, a publisher ID should be specified although there is no value to publish and the ID should be the same for the Max-i controller and the connected device and therefore cannot be the serial numbers, so to avoid confusion between the attributes in the controller and any attributes in the connected device (explicit messages), they must have a different base address! It is also possible – but confusing – to use a different identifier. If the device is used to generate network time messages, the ID must be 00H and since this is not a legal ID for process values, the Max-i controller shall not be connected to a device that generates that.

To be able to use the UART out-of-the-box without any register programming and in this way get access to a network, the UART shall be the default object and therefore have object number 11111111B (-1), and all necessary setup are done by means of input 1 – 3, which are used to invert the logical state of SI, SO and CTS.

The default state of serial in (SI) and serial out (SO) is 1, which makes it possible to detect a line break, but in case of for example galvanic separation by means of optocouplers, the high idle state may draw current so it may save power to invert the signals. The inversion possibility also makes it possible to connect SI and SO almost directly to an RS-232 line, where logical 1 is represented by a negative voltage. Since the detection level of most RS-232 receivers are approximately 1.4 V and not 0, SO may often be connected directly with no loss of speed (much lower drive impedance and voltage swing than RS-232), and depending on the input circuit of the Max-i controller, the input may for example be a simple 4.7 kΩ resistor followed by two diodes to 0 V and 3.3 V, respectively to limit the voltage swing on the input to the allowed range.

As there is no standard for the polarity of CTS, this has inversion possibility too.

If a Max-i controller has been reprogrammed to anything else than MODEM, it is only possible to reprogram the device over the bus, but it is anyway not possible for such a device to be programmed through for example an RS-232 interface since the I/O signals from the controller are usually not taken out, but connected to other circuitry.

If the UART in the connected device has an external clock input, which can accept a :8 or :4 clock (:16 not possible), it is highly recommended to connect this to one of the clock outputs (output 1 or 2). In this way, the connected device does not need to know the communication speed and both devices do not need to have accurate clocks, so it may not be necessary with an external clock oscillator on the Max-i controller.

Output 4 indicate collisions, which may be used for debugging purpose.

Output 5 goes high during **all** telegrams on the bus. This may be used to implement a simple bus load "speedometer" – for example by means of bar graph display and the IC LM3914 from Texas Instruments.

Output 6 is used for standard light output and control if GRPTYPE = 0 as specified in attribute 12.

I/O attribute 1, 2 and 3 are used in the following way:

I/O 1 (password 2 – user programmable) – **32 bit**									
Parity	Subscriber ID don't care		NU	IE	NTE	LIE	LGE	Group ID don't care	NU
1 bit	15 bit (handled in hardware)		1111	1 bit	1 bit	1 bit	1 bit	8-bit	1111
0	1	13	16	20	21	22	23	24	32

I/O 1 (password 2 – user programmable) – **36 bit**								
Parity	Subscriber ID don't care			Group ID don't care	IE	NTE	LIE	LGE
1 bit	15 bit (handled in hardware)		8 bit	8 bit	1 bit	1 bit	1 bit	1 bit
0	1	13	16	24	32	33	34	35

I/O 2 (password 3 – factory programmable) – **32 bit**					
Parity	HBF	NU	Minimum light level	NU	NU
1 bit	2 bit	1,1111,1111,1111	6 bit	1111,1111,1111,1111	1111
0	1	3	16	22	32

I/O 3 (password 3 – factory programmable) – **32 bit**									
Parity	Scrambling	MaxLength	NU	B4ES	B5ES	B6ES	B7ES	NU	
1 bit	1 bit	1 bit	1	1 bit	1 bit	1 bit	1 bit	1111,1111	
0	1	2	3	4	5	6	7	8	

I/O 3 (password 3 – factory programmable) – **32 bit**					
	NU	Gamma	Filter	NU	NU
	1111,111	1 bit	4 bit	1111	1111
	16	23	24	28	32

Fig. 7a.2

The subscriber's job in a modem device is to receive one or more messages and pass them on to a connected device. To be able to do this, **Bit 1 – 15** of I/O 1 are used together with a specified subscriber ID as a coarse filtering of the most significant 15 bits. If all 15 don't care bits are 1, which is the default setting, all implicit messages are passed on if implicit messages are enabled in the IE bit, and if all bits are 0, only messages, where the 15 most significant bits are identical to the specified ID, are transferred. In both cases, the message with the specified ID is always transferred.

In principle, the modem device may simultaneously utilize the message with the specified ID for an auxiliary function if the subscriber signature and the data type are also set, but there are usually not enough outputs for this.

Because of the timing as described in chapter "1.11 Telegram Separation", it is not possible to compare and hold back more than 15 bits plus the local bit in hardware because the received bytes need to be passed on to the connected device to get the UART ready in time for the next message.

15 bits are enough for the short ID, but not for the long, which need further comparison in software. In case of 36-bit attributes, bit 16 – 23 are used for further comparison if a 3-byte buffer gets possible in the future. This will enable comparison of the most important 23 bits of PNS identifiers.

Group messages works in the same way as implicit messages except that bit 24 – 31 are used for the acceptance filter. If group messages are simultaneously used for light control of a connected display, the group ID and signature must match the control message, but the data type need not to be specified as it is always LAMPCTRL in this case. By means of the 8 don't care bits, more or all group messages may simultaneously be passed on, but since the ID is fixed, there is a little less flexibility than if lamp control is not used and the ID may be freely selected to match the wanted range of transferred messages.

Bit 20 – 23 are used to enable different message types. If these bits are set to 0, the belonging message type is not transferred to avoid overflow in for example a debugger:

IE	= Implicit Enable
NTE	= Network Time Enable (repetitive transmission of time and flash bits)
LIE	= Local Implicit Enable (repetitive local communication)
LGE	= Local Group Enable (repetitive local communication by means of group messages)

Since it is possible to block all group messages except for the one with the specified ID by means of bit 24 – 31 = 00H, it is not necessary with a separate bit to disable group messages. Such a bit may however be necessary to block implicit messages with the long ID, but will simultaneously block messages, to which the device subscribes, and can therefore only be used for debuggers where it can block all implicit messages except for one's own.

Note that although a local message is blocked, it is still possible to poll the value by means of a global poll.

No matter these settings, six types of messages shall always be transferred:

- All explicit messages. Since these messages are necessary for programming and the most significant 15 bits do not contain the ID, but only the explicit-code and the complete attribute address, they cannot be compared and therefore need to be transferred as it is impossible to tell whether they are needed or not.
- All other network messages than time like for example the telegram speed and various network settings.
- All messages where the transmitter is "on" in the beginning like own commands and polls except for disabled network messages.
- Group-255 messages if these are enabled in bit 21 of the group attribute.
- Group message to which a device subscribes.
- Implicit messages, which match the most significant 15 bit of the identifier unless the IE bit is 0, which disables all implicit messages except for own commands and polls where the transmitter is on.

In case of group messages and in case of implicit messages, which match **all** bits of the ID – not just the 15 most significant, any specified delay in the group attribute or the signature attribute shall be inserted in the message between the identifier (short or long) and the first wanted data byte. If the pause is longer than the data, the last byte of the signature, which contains the Hamming code of the identifier (ID check), shall still be transferred followed by the break, which is used to separate messages.

Note that insertion of a pause is only possible if automatic descrambling is enabled since the message cannot be descrambled in a connected device if some data are excluded from the data stream!

The delay function may for example be very useful for stage light and to control and synchronize more servo axes. Because of the delay, a connected device will receive its data just after the identifier and therefore does not need to know the delay. This makes it possible to always program the delay in the Max-i controller.

⚠ **No matter the settings, the Max-i controller (not the connected device) shall also reply to group polls including group 255 (all devices) as all other objects**, so that it is possible to get a list of all devices or groups of devices on the bus.

The **HBF** bits may be used to enable automatic generation of time/flash messages. This is possible even if the input bits are used for UART communication.

Since the MODEM object anyway needs bigger clock accuracy than the RC-oscillator, an external oscillator must be added and it may then also be utilized to generate network time messages with flash bits, and because the timeout timer is free running in this case, it may also be used to check the oscillator accuracy.

To for example set the MODEM object for generating network time messages with 1 Hz (slow) and 3 Hz (fast) flash bits, make the following settings:

- ***Attribute 12, I/O type:** FC 0F FF FC 0H for TIME, MODEM, UNDEF and no extra groups of setup parameters.*
- ***Attribute 5, Publisher signature:** 80 0A 7F FF FH for ID-check 000,0000B, 167 ms repetition rate and flash-bits-only (short message) or 80 0A 7C FF FH for flash bits plus time (long message).*
- ***Attribute 9, Hardware 1:** 7F FF FB FF FH to disable network time messages and in this way avoid that these telegrams overflow the debugger.*
- ***Attribute 10, Hardware 2:** 3F FF FF FFH to start the repetition in divide by 3 mode (01B).*
- ***Attribute 4, Publisher ID:** 00 00 FF FF FH to set the ID to 000,0000,0000,0000B to start transmitting time messages. Note that after this setting, the device needs to be addressed as 00 00H instead of the previous ID if it has been set. Addressing on serial number is still possible.*

The **Scrambling** bit of I/O 3 determines whether the communication through the communication channel shall be scrambled or de-scrambled. If the bit is 1, which shall be the default state, the Max-i controller performs the scrambling, and the communication channel is de-scrambled and therefore not supervised. If the bit is set to 0, automatic scrambling is switched off and the connected device must perform the scrambling and de-scrambling itself.

The **MaxLength** bit is used to limit the maximum telegram length to 1028 data bytes plus identifier and signature. If the bit is set to 0, the default timeout is switched off and an infinite long telegram is possible.

Layer 7b, SPI object

This object enables transmission of data to or from a Max-i controller by means of a standard SPI interface where the Max-i controller is always the master and the connected device is the salve.

	Present	Future
Object name and ID #	SPI, -2 (11111110B)	
Publisher data type(s)	All types.	
Subscriber data type(s)	All types.	
Input 0	MISO for publisher data (master input) for SPI.	QSPI Input 0
Input 1		QSPI Input 1
Input 2		QSPI Input 2
Input 3		QSPI Input 3
Input 4	Publisher trigger (service request) for SPI.	
Input 5	Error or Status 1 input.	
Input 6	Error or Status 2 input.	
Input 7	Error or Status 3 input.	
Output 0	MOSI for subscriber data (slave output).	QSPI Output 0
Output 1		QSPI Output 1
Output 2		QSPI Output 2
Output 3		QSPI Output 3
Output 4	SPSCLK.	
Output 5	Subscriber SS/CS (slave select).	
Output 6	Light level.	
Output 7	Transmitter active pulse.	
Output 8	Communication timeout (watchdog).	

Fig. 7b.1

There are four modes of operation defined by the clock polarity (CPOL) and the clock phase (CPHA).

Mode	CPOL	CPHA	Data Valid Sample Edge
0	0	0	↑
1	0	1	↓
2	1	0	↓
3	1	1	↑

Fig. 7b.2

The transmission format is shown below for 24-bit transfer in mode 0 and 1. Since the clock polarity is only a matter of clock inversion, mode 2 and 3 with CPOL = 1 are not shown. Note however, that if the clock is inverted to be normally high, this shall also be the state of the clock output pin during power up reset.

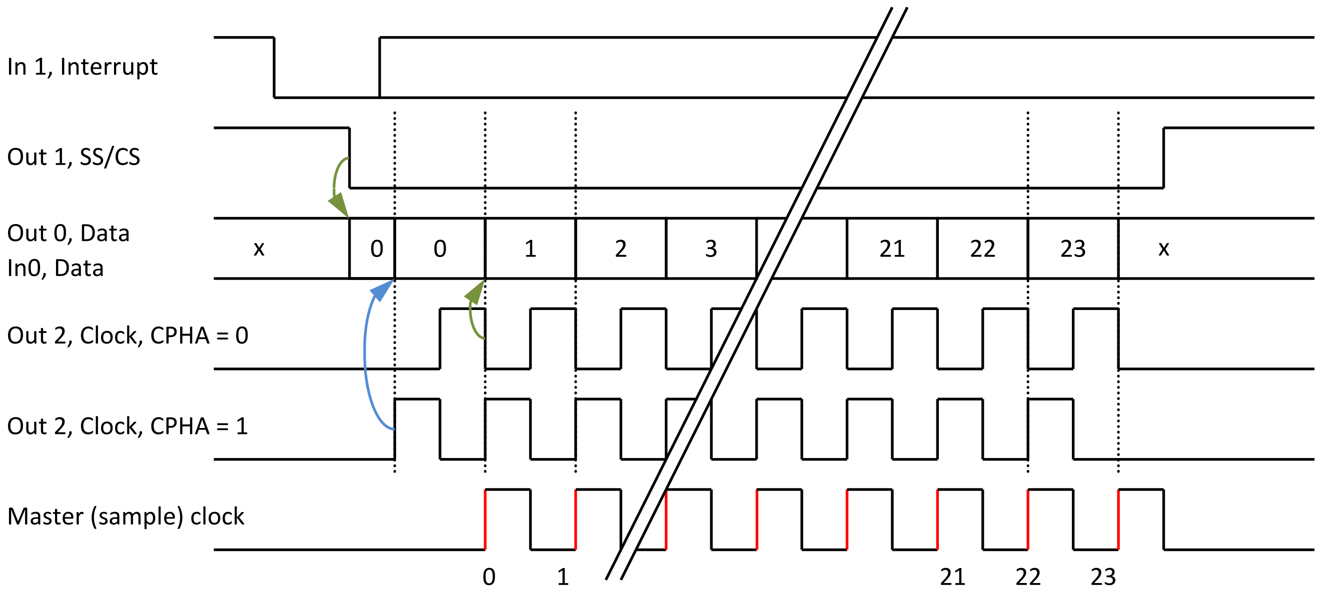


Fig. 7b.3

The slave is enabled and synchronized when SS/CS goes low, and the transferred data is stored/used when it goes high.

If CPHA = 0 and CPOL = 0 (or CPHA = 1 and CPOL = 1), the slave shall apply bit 0 on the falling edge of SS/CS as shown with the green arrows and shall sample the data from the master at the **rising** edge of the clock and change its output data to the next bit on the **falling** edge. The falling edge of the last pulse changes the data to something undefined. If CPHA = 1, the slave shall apply bit 0 and all following bits on the **rising** edge of the (first) clock pulse as shown with the blue arrow, and shall sample the data from the master on the **falling** edge. In practice, there will always be some delay from the master changes the outputs (clock and data) until the signals are received by the slave, and from the slave changes its data until they are received by the master – especially if optocouplers are used. Therefore, the reply from the slave will be delayed compared to the internal clock of the master. For this reason, the master should not sample the data in the middle of its internal clock period, but closer to the next clock edge that changes the slave data (shown red) so that the margin for delays becomes closer to one full bit period equal to 2 µS at 500 kHz. The master can do that because it controls the clock so that it can be guaranteed that the previous data are sampled before it is possible for the data to change again. The delayed master clock has the beneficial side effect that the extra half-bit delay, which is usually needed on the output of an SPI shift register, is not necessary.

In practice, the SPI controller in a Max-i interface will need to be masters no matter the direction of data flow. On the receiver side, it is necessary because there is no way to hold back the bus communication and SS/CS is the only way to tell when a message or data is ready, and on the transmitter side, it is necessary because SPI does not have any flow control so the only way to hold back the data flow when the bus is busy is to control the clock.

In the future, QSPI may also be supported so that for example a full digit may be transferred at a time.

Output 6 is used for standard light output and control if GRPTYPE = 0 as specified in attribute 12.

I/O attribute 1 and 3 have the following content (I/O attribute 2 not used):

I/O 1 bit 0 – 15 (password 2 – user programmable)						
Parity	NU	B2T	B3T	B4T	(B5T)	NU
1 bit	111,1111	2 bit	2 bit	2 bit	2 bit	1111,1111,1111,1111
0	1	8	10	12	14	

I/O 2 (password 3 – factory programmable)		
	Minimum light level	
	6 bit	
0	16	22

I/O 3 (password 3 – factory programmable)										
Parity	Length	Trigger	NU	B5ES	B6ES	B7ES	NU	CPOL	CPHA	
1 bit	1 bit	2 bit	1 bit	1 bit	1 bit	1 bit	1111,11	1 bit	1 bit	
0	1	2	4	5	6	7	8	14	15	

I/O 3 (password 3 – factory programmable)						
	SPILength		NU	Gamma	Filter	NU
	NU (0)	3 bit	111	1 bit	4 bit	1111
	16	17	20	23	24	28

Fig. 7b.4

In principle, SPI is able to transfer any number of bits, but in practice, almost all devices with SPI interface like A/D and D/A converters and microprocessors use an integer number of bytes. However, to be able to also transfer odd number of BCD digits and to interface to some A/D converters, the number of clock pulses shall be programmable in nibbles of 4 bits, which will also fit with future QSPI. The length is selected in SPILength according to the following table:

SPILength	**Length of SPI data transfer**	
(0)000	2 nibbles or 1 byte	8 bit
(0)001	3 nibbles	12 bit
(0)010	4 nibbles or 2 bytes	16 bit
(0)011	5 nibbles	20 bit
(0)100	6 nibbles or 3 bytes	24 bit
(0)101	7 nibbles	28 bit
(0)110	8 nibbles or 4 bytes	32 bit
(0)111	9 nibbles	36 bit
1xxx	Future ≥10 nibbles	≥40 bit

Fig. 7b.5

Since a 4-bit boolean value is easier to use than a 1-nibble SPI transfer, this possibility is not needed. For future expansion, bit 16 should be 0.

Note that the length of the SPI transfer has nothing to do with the length of the received message, which is always 36 bits, and just starts with the most significant nibble! For example, only the two most significant nibbles may be transferred (code 000B). In the same way, it is possible to transfer 9 nibbles (code 111B) even if only the 20 most significant bits are valid. In that case, the data shall be left shifted as shown in fig. 6a.1.

No matter the data type, an SPI transfer shall be triggered every time just some part of attribute 2 is changed including power-up and timeout reset.

For the publisher, the Length bit is used to select the telegram length **no matter the length of the SPI transfer to the connected device**! If this bit is 1 (default), 36-bit telegrams are **always** transmitted and if the bit is 0, 20-bit telegrams are always transmitted. In this way, it is possible to consume 32-bit values without sacrificing the publisher efficiency (SPI exchange subscriber and publisher data with the same length).

Input 4 may be used by the slave to request an SPI transfer and telegram transmission. This signal is not directly a part of the SPI standard, but used by many SPI slaves. The signal shall trigger a transfer on its falling edge and shall in this case be a high speed input without noise filter so that the pulse width is unimportant except that the minimum width should not be less than the width of the clock pulses (1 µS at 500 kHz). The slave may use the rising edge of SS/CS as an acknowledge that the data has been transferred and that the interrupt may fire again. Note that because SPI exchange data, any subscriber will be updated simultaneously with the transfer of publisher data. In the same way, any transfer of subscriber data will also cause a transfer of publisher data, but unless an interrupt pulse has been received or the data transfer has been triggered by heartbeat **before** the transfer, this shall **not** cause a telegram transmission and attribute 1 (published value) shall **not** be changed, as the data may not be valid!

The Trigger bits are used to select how data transfer from slave to master is triggered as shown below:

Code	Transfer trigger	Message trigger
00	Free run (continuous update).	Heartbeat or poll
01	(free for future purpose)	
10	Input 4 logical 1	After transfer, heartbeat or poll
11 (default)	Input 4 logical 1 or heartbeat timer	After transfer or poll

Fig. 7b.6

The free-run mode is intended for very fast (SAR) A/D converters, which can generate a new value each time SS/CS is high and transfer the data while SS/CS is low. In this mode, the value will typical be updated over 10,000 times per second and can therefore always be considered updated even if it is polled or transmitted by means of asynchronous heartbeat.

Even though less than 36 bits are transferred, which is the case for all practical A/D converters, the controller may internally use 36 clock pulses to left shift the transferred slave data. In this case, the high time of SS/CS will be quite long if SS/CS goes high shortly after the last SPSCLK pulse (not after all 36 bits) so the converter will have a relatively long conversion time.

Mode 10B is intended for intelligent devices like microprocessors, which can trigger a transfer (update the data) when a new value is available. When the value has been transferred, a message with the new data shall be transmitted. If a heartbeat or poll occurs, the latest value shall be transmitted immediately, but in these cases, it will always be more or less obsolete.

In the default mode 11B, the heartbeat timer shall not cause an immediate transmission, but shall be coupled in parallel with the trigger input so that it also triggers the SPI data transfer. A telegram shall then be transmitted when the transfer is completed so that the value is updated unless it is polled. This mode is intended for simpler devices like temperature sensors, A/D converters etc., which cannot generate a trigger, but are always ready to transmit data on a heartbeat. The interrupt input is only enabled to be able to trigger a telegram from a new device.

Layer 7c, Boolean object

This object uses the inputs and outputs as boolean I/O. The 4 inputs are used to represent 16 states of one input signal – never 4 different signals, and the outputs are used to represent 16 states of one output signal. It is for example not allowed to use some output to drive a motor starter and the other output to open a valve. This would also make it impossible to use numbering and identification systems like PNS and XML where each function has its own identifier.

Object name and ID #	BOOLEAN, -3 (11111101B)
Publisher data type(s)	Usually TIME, but PATTERN or LAMPCTRL if it generates group messages.
Subscriber data type(s)	TIME or PATTERN.
Input 0	Bit 0 (MSb) of a boolean input value.
Input 1	Bit 1 of a boolean input value
Input 2	Bit 2 of a boolean input value.
Input 3	Bit 3 (LSb) of a boolean input value.
Input 4	Error or Status 0 input.
Input 5	Error or Status 1 input.
Input 6	Error or Status 2 input.
Input 7	Error or Status 3 input.
Output 0	Bit 0 (MSb) of boolean output value. Red traffic light.
Output 1	Bit 1 of boolean output value. Amber traffic light.
Output 2	Bit 2 of boolean output value. Green traffic light.
Output 3	Bit 3 (LSb) of boolean output value. Green traffic light arrow. Light level if PATTERN.
Output 4	SPSCLK SPI clock.
Output 5	SS/CS SPI chip select.
Output 6	MOSI SPI data.
Output 7	Green safety output.
Output 8	Communication timeout (watchdog).

Fig. 7c.1

7c.1 Inputs

All inputs are standard inputs described in chapter 7.1.

7c.2 Outputs

Output 5 goes high if one of the watchdogs time out. In case of the implicit watchdog, this shall only happen if the consumed value is different from 0000B.

Output 0 – 3 can be programmed as pure Boolean (0 or 1) or as a pulse coded output with smoothing filter, but the function depends on the programming of the data type for group messages GRPTYPE.

7c.2.1 Data type TIME and GRPTYPE = 1

In this case, output 0 – 3 are Boolean outputs intended to drive actuators by means of the last 4 bits in the data field of the message as specified in the TIME data type. The outputs have a logical OR/AND function between implicit messages and group messages and may or may not have a limited rise and fall time with a fixed gamma of 1.

A gamma of 1 is perfect for inrush current limiting and ensures that the maximum step size of the load current is limited to approximately 13 % during the ramp, which limits the voltage drop when heavy loads are switched on. However, since the power of for example motors and heating elements depends on the supply voltage in the power of 2 (P = U^2/R), they in effect get a gamma of 2.

The time, which is stored in attribute 2, is the specified command time in case of long messages or the local time of reception in case of short messages.

7c.2.2 Data type TIME and GRPTYPE = 0

In this case, output 0 – 3 are instead intended to drive up to 4 lamps such as control lamps, signal towers stack lights and traffic lanterns with a gamma of 1 or 2.44 as also specified in the TIME data type. The time in the message is intended to drive a display such as a countdown display for a traffic lantern and is transferred by means of an SPI interface.

All lamps use a common on-level with a programmable minimum level to avoid that the lamps can be dimmed so much that they become invisible or the luminous flux (lumen) gets below a required minimum for example for traffic lights. The light level shall be set to maximum during power up and may later be adjusted by means of 4-bit or 20-bit **group** messages as specified in the LAMPCTRL data type.

If the light level becomes lower than the minimum level, this level shall be used instead, but the lamp level itself shall be independent of the Boolean state and the minimum level so that even if the lamp is off or the level is below minimum, the level can still be set and counted up and down. In this way, the lamp becomes the same level as other lamps in that group when it turns on or the lamp level becomes bigger than the minimum level.

To avoid sudden steps in light intensity, it is usually recommended to use the (4-bit) up/down dimming so that all lamps are dimmed from the present level, which may not be the same. 4-bit dimming also removes the necessity for an up/down counter on the publisher side. There may however be situations or applications like traffic lights where it is more convenient to set an absolute value for example based on the present light level, and if the value is not changed more than 8 steps at a time, so that the smoothing filter only generates one step at a time, and a (very) slow filter is selected, this can be invisible.

7c.2.2.1 Traffic lights

In case of traffic lights, output 0 shall be used for red, output 1 for amber, output 2 for green and output 3 for any green arrow. The most common light sequence and light meanings according to the Vienna Convention, Article 23 is:

- Red (Stop).
- Flashing red (Stop at stop line for example for a bridge, a ferry landing stage, for emergency vehicles entering the road, or on the approach to low-flying aircraft).
- Red + Amber (Prepare to pull away).
- Flashing amber (Give way to pedestrians; go if it's safe to do so).
- Green (Go).
- Amber (Stop unless it's not safe to do so).

Since a (countdown) display must have the same light level as the lamps, a constant 1 on output 3 or any of the other outputs may be used as a light level signal to control this. It is therefore only possible with 3 controllable lamps in this case, but a countdown displays is typical used for pedestrian crossings and road work where this is usually not necessary. If a green arrow is needed, a separate Max-i controller must be used to transfer the display data.

Since the present Max-i controller has only 4 main pulse code modulators for the 4 main outputs, one of these must be used as light level control and since output 3 is already driven by the white channel and only drives any green arrow, this channel should be selected. In case of 7-segment decoders, which do not have an off-state, the countdown display may be switched off by setting this output low.

Dimming is very desirable for traffic lights for two reasons:

- *To save power. Just a dimming to a level, which the eye regards as 50 %, reduces the power consumption to only 18 % if LED lamps are used.*
- *To prevents blooming effects on symbols and the lights from dazzling of the drivers.*

To increase the safety, output 7 is used as a green safety output, which shall go on driven by its own decoding circuit when bit 32 (output 0) is off and bit 34 (output 2) is on. The safety output should be AND'ed with green so that it requires a double error to show green erroneously. To ensure that the green lamp has enough time for a smooth fade off, the safety output shall be active 250 ms – 750 ms after bit 32 goes on or bit 34 goes off.

To further increase the safety, the implicit watchdogs should be enabled to set all outputs off in case of a receiver timeout.

7c.2.3 Data type PATTERN and GRPTYPE = 0

This mode is very similar to TIME with GRPTYPE = 0, but it uses a different message format as specified in the PATTERN data type, which makes it possible with automatic flashing, a variable filter time and a reception delay as shown in copy below:

36-bit implicit or explicit message. **B20V = 0 or B20U = 0.**					
Any display data	Boolean with flash				Filter time
24 bit	4 × 2 bit				4 bit
0	24	26	28	30	32

20-bit **right shifted** implicit message. **B20V = 1 and B20U = 1.**							
Any display data	Boolean with flash	Any display data	Boolean with flash				Filter time
8 bit	8 bit	8 bit	4 × 2 bit				4 bit
0	8	16	24	26	28	30	32

Fig 7c.2

In this mode, bit 24 and bit 25 shall both be 0 (no steady or flashing red), and bit 26 or bit 27 shall be 1 to activate the safety output (steady or flashing green).

A variable filter time is very useful for flashing so that the filter time can be decreased for (very) high flash frequencies to avoid that the flashing looks more like a sinewave instead of a square wave.

If the possibility for data to more devices is utilized, the delay between each device may be set to 1 so that for example a single 20-bit message can drive 8 lamps by means of two controllers.

7c.3 Attribute programming

I/O Attribute 1 and 3 has the following coding (I/O attribute 2 not used):

I/O 1 (password 2 – user programmable)												
Parity	NU	DB	B0T	B1T	B2T	B3T	NU	Flash phase	B0C	B1C	B2C	B3C
1 bit	11	1 bit	2 bit	2 bit	2 bit	2 bit	1111,1111	4 × 2 bit	1 bit	1 bit	1 bit	1 bit
0	1	3	4	6	8	10	12	20	28	29	30	31

I/O 2 (password 3 – factory programmable)				
Parity	HBF	NU	Minimum light level	NU
1 bit	2 bit	1,1111,1111,1111	6 bit	1111,1111,1111,1111
0	1	3	16	22

I/O 3 (password 3 – factory programmable)															
Parity	NU	DBE	B4ES	B5ES	B6ES	B7ES	B0D	B1D	B2D	B3D	(B4D)	(B5D)	CPOL	CPHA	
1 bit	11	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	
0	1	3	4	5	6	7	8	9	10	11	12	13	14	15	

I/O 3 (password 3 – factory programmable)										
	SPILength		NU	B20V	Gamma if GRPTYPE = 0	Filter	B0L	B1L	B2L	B3L
	NU (0)	3 bit	11	1 bit	1 bit	4 bit	1 bit	1 bit	1 bit	1 bit
	16	17	20	22	23	24	28	29	30	31

Fig. 7c.3

The **Flash phase** bits in bit 20 – 27 of I/O 1 are used to make it possible to invert the phase of the two flash frequencies for each lamp as shown below:

Flash phase							
Output 0		Output 1		Output 2		Output 3	
Slow flash Flash0	Fast flash Flash1	Slow flash Flash0	Fast flash Flash1	Slow flash Flash0	Fast flash Flash1	Slow flash Flash0	Fast flash Flash1
20	21	22	23	24	25	26	27

Fig. 7c.4

In the default state where these bits are 1, the phase shall **not** be inverted, but if any of the bits are set to 0, the polarity for that output shall be inverted for the specified flash frequencies.

The inversion may be very useful for example for wig-wag lamps. If the two flash frequencies are the same, it is even possible to have two lamps, which flashes synchronous if the data is 10B or 01B, but alternates in the opposite situation 01B or 10B.

B0C – B3C are used to determine if the hour counter shall count while the corresponding output bit is **logical** high (may be dimmed, but preserves the logical state). In the default state where all bits are 1, the hour counter shall count if just one output is activated, and if all bits are 0, the hour counter shall stay at 0.

The **HBF** bits may be used to transmit network time messages with two flash bits, but may also be used for other synchronization messages.

The **B20V** bit should be left in the default 1 state (20 bit) to make it possible for the user to select between 20-bit right shifted data and 36-bit data by means of B20U.

Layer 7d, Keypad scanner object

This object is very similar to BOOLEAN. It is used to read a 4 × 4 or 2 × 6 contact matrix keypad on for example alarm systems and electronic locks and transmit the corresponding raw binary code as shown below:

Row (input)	Column (output)			
	0	1	2	3
0	0000 1	0001 2	0010 3	0011 A
1	0100 4	0101 5	0110 6	0111 B
2	1000 7	1001 8	1010 9	1011 C
3	1100 */Clr	1101 0	1110 #/En	1111 D

Row (input)	Column (output)	
	0	1
0	0000 1	0001 2
1	0010 3	0011 4
2	0100 5	0101 6
3	0110 7	0111 8
4	1000 9	1001 0
5	1010 */Clr	1011 #/En

Fig. 7d.1

The red digits show the telecommunication layout, which is by far the most frequently used today, but any layout may of course be used (the code only depends on the row and column).

On the receiver side, a simple look-up table may be used to convert the code from the keypad to a key code, which corresponds to the actual key depending on the layout. This can be done in a single statement by means of an offset in a string, which contains the 12 or 16 values. Key 1 – 9 of a standard 4 × 4 telecommunication keypad (red) are also easy to decode by means of the simple formula: key = code + 1 – (code >> 2). For example, key 8 = 1001 + 0001 – 0010.

	4 × 4 keypad	2 × 6 keypad
Object name and ID #	KEYPAD, -4 (11111100B)	
Publisher data type(s)	TIME	
Subscriber data type(s)	TIME – 4 bit or 36 bit	
Input 0	Row 0 input.	
Input 1	Row 1 input.	
Input 2	Row 2 input.	
Input 3	Row 3 input.	
Input 4	Error or Status 0 input.	Row 4 input.
Input 5	Error or Status 1 input.	Row 5 input.
Input 6	Error/Tamper-detection or Status 2 input.	
Input 7	Error/Tamper-detection or Status 3 input.	
Output 0	Boolean output 0 – unlock.	
Output 1	Boolean output 1 – lock.	
Output 2	Column 0 drive.	Boolean output 2
Output 3	Column 1 drive.	Boolean output 3
Output 4	Column 2 drive.	Column 0 drive.
Output 5	Column 3 drive.	Column 1 drive.
Output 6	Light output if GRPTYPE = 0.	
Output 7	Beeper – phase 1 or transmitter active (single beep pulse).	
Output 8	Beeper – phase 2.	

Fig. 7d.2

Output 0 and 1 are intended for driving a locking pawl. Because of the OR/AND logic in the TIME data type, output 1 must be used to lock if "all-lock" by means of a group message shall be possible.

The locking pawl shall be held mechanically in the two positions so that 0000B for example caused by a power failure or a reset does not change its position!

To save energy, the output currents should be switched off when the locking pawl is in the intended position. It is also possible to use one of the watchdog timers (implicit or group) to turn the output off after a short time, but this will disable the opposite message type! If for example the implicit watchdog timer is enabled, group messages cannot be used since all outputs are reset as soon as the implicit watchdog runs out and until a new implicit message is received.

If All-off is enabled, it is also possible to use an All-off/Lock-all message on group 255. In this case, 0000B of the All-off message shall be converted to a 0100B lock command in the Max-i controller. To increase the safety, group 255 messages shall not be able to unlock the lock.

Because of the OR/AND logic, a lock, which is locked by means of an implicit message or a Lock-all message, cannot be opened by means of a group message, because bit 0 of main latch is reset. An attempt to do this will just create 0000B, but if the lock is locked by means of a group message so that the main latch is still set, it can also be opened by a group message. In this way it is possible to select the wanted safety level.

When you leave a building for a longer time, it is highly recommended to use Lock-all and/or implicit messages. Even if a thief should get access to the bus, it will make it very difficult to guess the identifier and be able to unlock – especially if the long identifier is used, which also makes it easier for the alarm system to detect the tamper attempt. However, in a building with many doors it may be very convenient to use group messages in daily use.

The column outputs are used to drive one column high at a time, and the inputs, which are active high, are then used to read that column. The 1 (2×6) or 2 (4×4) least significant bits of the code are taken directly from the column counter.

Output 6 is a light level output, which may be used to control the light in the keypad if GRPTYPE = 0. In this case, it is not possible to use group messages for All-lock except for group 255. Note that since there is only one white channel, which must follow bit 3 if output 3 is set to analog by means of B3L = 0, it is necessary to set B3L to 1 (Boolean) to create a steady signal on output 6, but all 4 outputs are anyway usually Boolean.

Output 7 and 8 are used to drive a beeper when the key message is transmitted. This corresponds to transmitter active for other objects and the on-time should be the same. The two outputs may for example be two opposite phases of a synthetic sinewave, which makes it possible to connect a small loudspeaker directly between the outputs without any capacitor if the current is lower than allowed or a H-bridge driver is used.

Note that if there is any risk of people in the building while the door is locked, it should always be possible to open a door mechanically from the inside if the lock is not powered up.

The I/O attributes are used as shown below:

I/O 1 (password 2 – user programmable)		
	DB	
	1 bit	
0	3	

I/O 2 (password 3 – factory programmable)		
	Minimum light level	
	6 bit	
0	16	22

I/O 3 (password 3 – factory programmable)															
Parity	Type	NU	DBE	B4ES	B5ES	B6ES	B7ES	B0D	B1D	B2D	B3D	(B4D)	(B5D)	NU	
1 bit	1 bit	1	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	11	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	

I/O 3 (password 3 – factory programmable)									
	NU	OutSel	NU	Gamma if GRPTYPE = 0	Filter	B0L	B1L	(B2L)	(B3L)
	1111	1 bit	11	1 bit	4 bit	1 bit	1 bit	1 bit	1 bit
	16	20	22	23	24	28	29	30	31

Fig. 7d.3

The Type bit is used to distinguish between the two keypad types. If it is the default 1, a 4 × 4 keypad is used, and if it is 0, a 2 × 6 keypad is used.

All **row** inputs (not error or status inputs) shall be "on-only" no matter the settings of the B0T – B5T bits in I/O 1 so that it is not necessary to program this register in most cases.

B4ES – B7ES are used to specify whether the belonging input shall:

- Be regarded as an error input (default 1) and therefore generate an error telegram when the input goes high and hold the state until the error attribute is read. This may for example be used for a tamper detector or for an opening detector, which is supplied by output 3 or 4 so that it only generates an error message, if the door is forced open while it is locked.
- Be regarded as a status input (0), which shows the present input status when the error attribute is polled. This may for example be used to show the status of a locking pawl or an opening sensor.

B0D – B5D should usually just be left in the default 1 state as the key debounce should normally be enabled for a simple mechanical key matrix. B4D, B5D, B2L and B3L are only used in case of a 2 × 6 keypad and therefore shown i brackets. They should be left in the default 1 state for a 4 × 4 keypad.

The OutSel bit is used to switch output 7 between Phase 1 (default 1) of a beeper and transmitter active pulse (0).

It is possible to use this bit, which is usually used to specify the expected date length (B20V), since the data length is either 4 bit or 36 bit with timestamp. 20-bit messages do not make sense.

Layer 7e, Lamp Controller object

This object is used to drive one, and control up to two (LED) lamps used for lighting such as interior and architectural lighting and stage light, but may also be used for example to control window openers and blinds. Control lamps, indicator lamps and traffic lights are instead driven by the BOOLEAN object and use the TIME or PATTERN data type.

Object name and ID #	LAMP, -5 (11111011B)	
Publisher data types	TIME	LAMPCTRL
Subscriber data type	LAMPCTRL	
Input 0	Bit 0 of boolean input value.	Lamp 0 one-button sequence.
		Lamp 0 two-button up.
		Bit 0 of fixed level.
		Channel 0 of quadrature encoder.
Input 1	Bit 1 of boolean input value.	Lamp 0 down.
		Bit 1 of fixed level.
		Channel 1 of quadrature encoder.
Input 2	Bit 2 of boolean input value.	Window 0 stop.
		Bit 2 of fixed level.
Input 3	Bit 3 of boolean input value.	Lamp 0 sequence (0) or fixed level (1) selection.
Input 4	Error or Status 0 input.	Lamp 1 one-button sequence. May be a momentary SPST push-on switch on the quadrature encoder for lamp 0.
		Lamp 1 two-button up.
		Bit 1 of fixed level.
Input 5	Error or Status 1 input.	Lamp 1 down.
		Bit 1 of fixed level.
Input 6	Error or Status 2 input.	Window 1 stop.
		Bit 2 of fixed level.
Input 7	Error or Status 3 input.	Lamp 1 sequence (0) or fixed level (1) selection.

	Full color (B20V = 0)	Monochrome or dichromatic (B20V = 1)
Output 0	Red.	Lamp 0 off-status or group control (blue or cyan-blue).
Output 1	Green.	Lamp 0 on-status (yellow or amber).
	Fixed	
Output 2	Blue or cold white.	
Output 3	Warm white or true amber.	
	Output code bit 16 – 21	SPI and status selection
Output 4	xxxx.11	Dim-to-warm output with square root gamma (√aA).
	1xxx.10	
	0xxx.10	Lamp 1 off-status (blue or cyan-blue).
	xxxx.01	SPSCLK for SPI transfer.
	xxxx.00	
Output 5	xxxx.11	White with square root gamma (√W).
	xxxx.10	Lamp 1 on-status (yellow or amber).
	xxxx.01	White with square root gamma (√W).
	xxxx.00	SS/CS (slave/chip select) for SPI transfer.
Output 6	xxxx.11	AC/DC-converter enable.
	xxxx.10	
	xxxx.01	MOSI for SPI transfer.
	xxxx.00	
	Fixed	
Output 7	Artificial amber (aA).	
Output 8	Artificial cyan (aC).	

Fig. 7e.1

There are two types of lamps – monochrome or dichromatic lamps with 1 – 2 basic colors (cold white and warm white or blue and amber) and full-color lamps with 3 – 4 basic colors (red, green, blue and white or amber).

7e.1 Color system

Pure RGB works well for light, which goes directly into the eyes as from a display or TV-set, but the CRI is terrible (≈ 25 %) when it is used for lighting as the light only contains 3 very narrow frequency bands with very little cyan and very little yellow to orange light as shown below so colors in these two frequency ranges will appear **much** too dark.

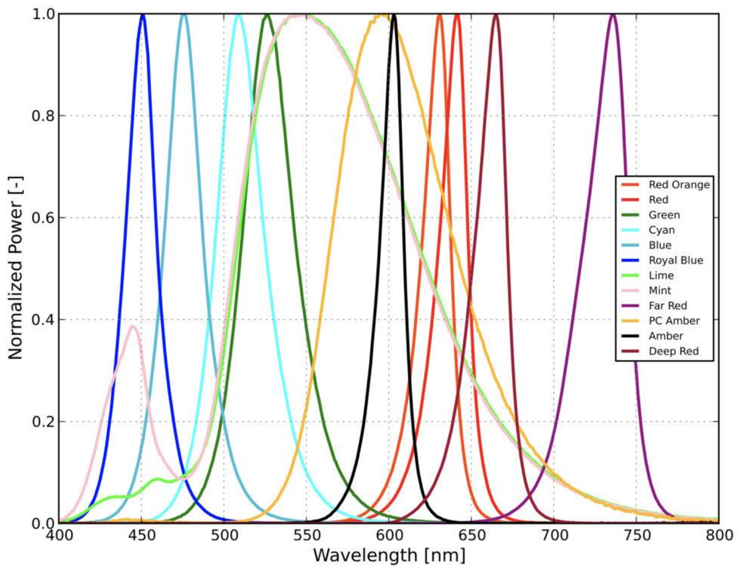


Fig. 7e.2

For this reason, most RGB systems have a supplementary white LED with a much better CRI, but if colored light is wanted, the CRI is still very poor in the selected range. Therefore, Max-i uses two more colors – artificial (phosphor converted – PC) amber (aA) or artificial PC-Lime (aL), which is generated by the Max-i controller on the basis of red and green, and artificial cyan (aC), which is based on blue and green, so that only 4 bytes are necessary to generate 5 or 6, 13-bit (gamma = 2.44) color channels like for example RGBWaAaC, RGBaLaC or RGBAaC. This also makes it possible to use a deeper red and blue, which makes it possible to generate better skin tones, sunset light and some shades of pink, which requires deep blue or even violet.

Lime and mint have almost the same frequency range. The only difference is that the blue source LED is not entirely damped in mint as shown above.

The extra two colors has the further advantage that they expand the color gamut/space as shown below on the CIE 1931 eye curve so that it is possible to make much more saturated and vibrant golden and cyan colors:

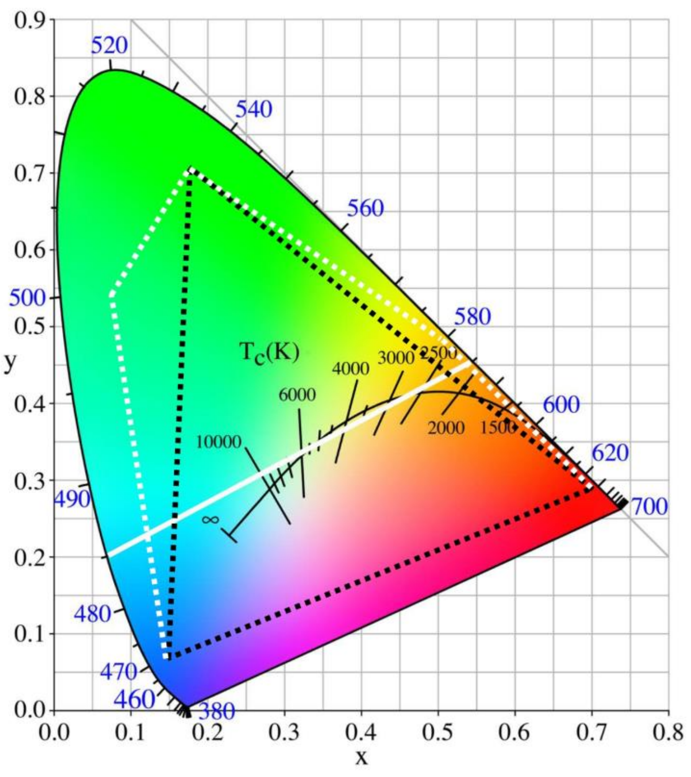


Fig. 7e.3

The dotted black triangle shows the typical color gamut/space for RGB, and the white dotted lines shows the expansion with added artificial amber and cyan.

Cyan has the strongest impress of the color gamut area of all additional colors and is especially important for skin tones. Because the sensitivity of the eye is very low in this frequency range, such colors will look way too dark if not lighted properly.

Note that it is not possible to generate an artificial magenta in the same way as artificial amber and cyan, because there is no wavelength for magenta and therefore no magenta LED's. All colors on the axis between red and blue and in the interior of the figure including white are generated in the brain on the basis of at least two colors.

The white line shows how the color temperature may be controlled quite accurate over a fairly big range from warm white (2500 K) to sunlight (6500 K) by adjusting the relationship between amber and cyan-blue, where the latter is a combination between artificial cyan and blue. Without artificial cyan, the color temperature must be controlled by means of blue and/or lime. If amber is used, the lamp will dim to unpleasant reddish instead of warm white.

The generation of the artificial colors shall be done by taking the lowest value of the two source colors **after the smoothing filter**. If for example the filtered values are R = 127, G = 63 and B = 40, aA becomes 63 and aC becomes 40. Note that if either red or green is 0, artificial amber will also be 0 so saturated red and green colors are still possible. In the same way, artificial cyan will be 0 if either green or blue is 0 so saturated blue is also possible.

Because artificial amber is very simple calculated, it is not as good as separately driven amber, so if a white channel is not needed and maximum color control is wanted, white can be replaced with PC-amber or PC-lime to generate RGBAaC or RGBLaC. In this case, aA is not used, but √aA may be used for dim-to-warm.

PC-Lime may also be very useful as a supplement to RGB instead of white. This may create white with a CRI of 90+ without the use of a white LED, and it also removes some of the problems with a direct-green LED, which are:

- *A fairly low efficiency of only 90 lm at a current of 350 mA. As a comparison, the efficiency of lime (and mint) is 125 lm (and 140 lm) for the same current. Besides, lime fits with the maximum sensitivity of the eye and therefore generates a strong light impression.*
- *A very narrow frequency band and therefore a poor CRI. PC-Lime and PC-mint have a **much** wider band, which enables a CRI of 90+.*
- *A very big distance to the white point.*

As a supplement to aA and aC, the LAMP object should have two more color outputs - √aA and √W, which follow artificial amber and white, but have the square root of the selected gamma, that is, 1.56 if the gamma is 2.44. If the √aA output in an RGB system is connected to a weak lime LED or simply replaces aA, the color temperature is lowered at low levels, which makes the light more pleasing as shown on the Kruithof curves below:

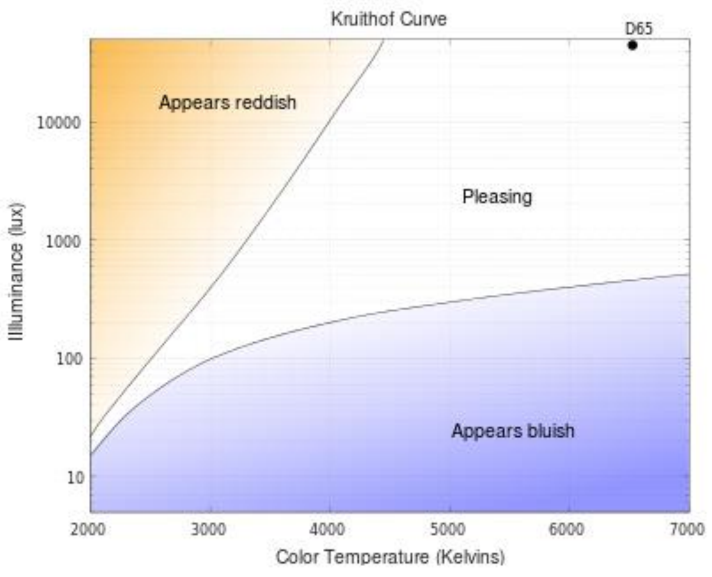


Fig. 7e.4

At maximum level, the light is primary determined by the LED's on the other channels and the lime LED's are almost invisible, but when the level is reduced, the lime, yellow or amber LED's becomes more and more visible due to the lower gamma and this changes the color temperature. Because only weak LED's (low current) are needed and PC-lime has a very high efficiency, this solution is very economical.

In case of a monochrome or dichromatic lamp, the unused red and green channels shall be set to a copy of white so that √aA becomes equal to √W. In this way, √aA can be used for all lamp types and is therefore always selectable even in case of an SPI interface.

Dim-to-warm is always appropriate to some extent, but it may also be used to change a lamp from cold bright working light to warm cozy dinner lighting.

The √W output may also be used to control the current of the white channel and in this way increase the gamma to 4 (2.44 + 1.56). This may be used to create a **very** big contrast ratio without visible steps at low levels for example for white stage lamps, which are usually operated at low levels, but must be able to create **very** high levels.

*Even though gamma correction is used, the 8-bit binary values have the disadvantage that a change of one step at low levels – for example from 2 to 3 is **much** more visible than at high levels from for example 254 to 255. The difference may however to some extend be equalized if the gamma is increased as this will increase the necessary binary values for low levels, but lower them at high levels.*

Note however that due to the pulse code modulation, it is necessary to filter the √W signal to be able to control the LED current and this will increase the time it takes to change the level, but also make the change smoother.

If the same is done with the color channels, the color hue will change with the amplitude so that for example an orange color becomes reddish at high levels and yellowish at low levels as shown previously.

Some stage lamps use full RGBAW, but it is not possible for the Max-i controller itself to fetch 5 or more colors because if the reception delay is adjusted to allow that, there may not be enough bytes in a telegram to receive only a single color (byte) if the byte delay is odd. Besides, 5 bytes do not fit with the register set in the controller, which is entirely 32/36 bit oriented. Instead, the artificial colors convert 3 or 4 channels (RGB or RGBW) to 5 or 6 channels (RGBaAaC, RGBWaAaC or RGBAaC).

*Some stage lamps use CMY color wheels and a white source. This generates excellent pastel colors, a good CRI, no flicker and very smooth light variations even at low levels, but the energy efficiency is **extremely** low as the white source is always on, it is not possible to generate pure black and saturated colors, and a lot of expensive and complicated mechanics is needed. The color system of Max-i with the artificial colors combines many of the advantages of both methods.*

7e.2 Bin Correction for Lamps

To be able to avoid expensive bin selected LED's and compensate for tolerances in LED current, each channel including artificial amber and artificial cyan shall include a possibility for individual calibration – the so called bin correction. The calibration shall be linear so that it does not depend on any gamma correction. Each calibration factor, which shall be multiplied with the corresponding output value, shall be calculated as a 5-bit calibration value plus 100001B = 33 so that the calibration factor becomes (33 + CalibrationValue) / 64. In this way, the LED current (output value) may be adjusted a factor 1:0.516 (+0, -48%) with an accuracy of ±0.8%, which corresponds to the sensitivity of the eye. Below the BT.709 black level of 16 = 10H, a reduced resolution is acceptable.

7e.3 Status indication

In case of monochrome or dichromatic lamps, which do not use output 0 (red) and 1 (green), these outputs are instead used for status signals for lamp 0, and output 4 and 5 are used for lamp 1.

Since there is no green channel on a dichromatic lamp, it is not possible to generate a meaningful artificial amber and cyan so outputs 4 and 5 are free for this use.

The status outputs are intended to drive two-color blue + yellow or cyan-blue + amber LED's, which are able to generate 3 colors – blue or cyan-blue, yellow or amber, and white. Blue and yellow are complement colors, which can generate a good white and cyan-blue and amber are even better as shown above, but blue and amber generates a slightly reddish white. Yellow or amber and white is utilized to distinguish between publishers for implicit messages and group messages. If a channel is programmed to generate implicit messages, output 1 and 5 are just the opposite of output 0 and 4, but if it generates group messages, output 0 and/or output 4 are always on (blue or cyan-blue), so that the on-state becomes white instead of yellow or amber.

Blue and yellow or cyan-blue and amber are chosen instead of red and green for 3 reasons:

- *Blue and yellow is the main axe of the eye, which is visible to even red-green color blind people.*
- *Since the efficiency of a blue LED is **much** higher than red, amber and yellow, it is only necessary with a fraction of a milliamp for stand-by current.*
- *It gives a more modern design and especially cyan-blue and amber are very nice colors.*

To reduce the risk of epileptic seizures, it is highly recommended not to use stroboscopic lamps for more than 2 seconds at a time and keep the flash rate below 3 per second for red and below 5 per second for other colors including white. At this rate, it is estimated that only 5 % of flicker-sensitive persons will be at risk of seizure. Be especially careful in the range between 10 – 20 flashes per second and NEVER EVER alternate between different colors and especially not between blue and red as this may cause "Pokémon seizures"!

7e.2 Inputs

The behavior of the inputs is determined by the publisher data type.

7e.2.1 Data Type TIME

In this case, input 0 – 3 shall behave like ordinary BOOLEAN inputs with triggers, which are selected by means of the corresponding BxT bits. This may for example be used to transmit data from smoke, fire and burglar detectors located in a lamp.

7e.2.2 Data Type LAMPCTRL

In this case, it is possible with two implicit identifiers in attribute 4 and 6, which are used for both publisher and subscriber. This makes it possible to control two (group of) lamps with 4-bit control – main lamp and auxiliary lamp – and at the same time subscribe to the same two identifiers. This has the side effect that the subscriber responds to both of its own channels. Note that the very common lamps with no need for a publisher may be controlled in three almost completely independent ways – two implicit identifiers with common delay and one group identifier with its own delay – just by programming the publisher to LAMPCTRL and specifying the extra implicit identifier and its ID-check in attribute 6 and 7. It may also be controlled by means of an explicit write message to attribute 2 using the serial number.

Input 3 determines the function of input 0 – 2 for the main lamp and input 7 determines the function of input 4 – 6 for the auxiliary lamp.

If input 3 or input 7 is 0, input 0 and 1 or input 4 and 5 shall be used for a one-button or two-button sequential lamp control or for an "analog" control as defined in the LAMPCTRL data type. Input 2 and 6 shall generate a 0011B code, which may be used to stop a movement of for example a window.

If input 3 or input 7 is 1, input 0 – 2 or input 4 – 6 shall instead be used to generate the 8 fixed light levels including zero by means of their gray code as defined in the LAMPCTRL data type.

In case of an existing toggle button, which cannot be rebuilt to momentary action (spring-return) for one-button sequential control, input 3 or 7 may be fixed to 1 and the level may then be for example be switched between 0 and 100 % by means of bit 0 or bit 4.

If the device is programmed to publish group messages (see layer 2), the fixed levels can be used to control the light in for example an entire control panel. They are also very convenient for window openers.

7e.3 Outputs

The outputs depend on the type of lamp – monochrome, dichromatic or full color – and any use of a SPI interface to transmit (mechanical) functions to connected electronics for example to control pan and tilt of a moving head lamp.

7e.4 Operation Modes of Lamps

As it is the case for all other objects, the behavior of lamps for lighting depends on the GRPTYPE bit even though the data type is LAMPCTRL for both implicit messages and group messages.

If GRPTYPE = 1, implicit and group messages are used in exactly the same way and have the same data length and control the same color channels. A 4-bit message **except for up and down** shall set an RGBW lamp (not RGBA) in white mode where all channels have the same level equal to the white channel. Even though the input to red, green and blue is the same, each channel shall use its own smoothing filter (and bin correction) as the level before white mode is activated may not be the same. For example, one color may go from 10% to 0% while another color may go from 80% to 0%. In such cases, the smoothing filters ensure approximately the same turn-over time so that the color does not change while the lamp turns off.

White mode and group addressing may also be very practical for testing all lamps and maximum power consumption in stage light applications if a light controller, which can generate the common telegram, is not available.

If GRPTYPE = 0, implicit messages works the same way as for GRPTYPE = 1 except that the only messages, which can change white mode, are long implicit messages, which set the lamp in color mode, and the 4-bit message 0000B, which turns the lamp off and set it to white mode. When the lamp is in color mode, group messages (4 bit or 20 bit) and local operation by means of buttons or a quadrature encoder shall therefore only control the white channel. This is used for daylight control.

7e.4.1 Daylight Control

Since 4-bit up and down messages cannot turn a lamp on or off and do not change the operation mode – white or color, they can always be used to step the white channel up or down if possible no matter the setting of GRPTYPE, so GRPTYPE = 1 can be used for most lamps and therefore does not need to be user programmable (is protected by password 3). This setting for example makes it possible to control a lamp from one or more wall buttons by means of implicit messages, set it to a part of a scene by means of a common group telegram and step the white channel up and down for daylight control by means of 4-bit group messages.

Dimming the white channel preserves the color temperature of an RGBWaAaC system to some extend as the sunlight replaces the light from the white LED.

In some situations it may however be desirable to be able to set the white/amber channel **alone** to one of the fixed values or change it by means of the least significant byte of a 20-bit group message. This may be done by means of GRPTYPE = 0. In this case, local operation of a lamp and/or group messages only change the white channel unless the lamp has been turned off and therefore is in white mode, so automatic setting of daylight level may easily be corrected without changing the color channels, which to some extend preserves the hue. Note however that the level of the white channel may step to a new level, when a fixed value or a 20-bit group message is received, so the correction may only be temporary. For this reason, it is usually recommended to use step-up and step-down for daylight control and only synchronize the levels from time to time by means of fixed values and/or long messages.

Dimming the amber channel of an RGBAaC system changes the color temperature, so daylight control cannot be used in this case, but it may be used to control the color temperature so also in that case, GRPTYPE = 0 may be relevant.

7e.4.2 Group 255

Lamps for lighting may be switched off by means of short group-255 messages, but it is not allowed to switch them on. By means of 1101B and 0001B messages it shall however be possible to change the light temporary to the same color as used in case of a watchdog timeout as specified in the LAMPCTRL data type.

7e.5 I/O Settings

I/O attribute 1-3 have the following content, which corresponds to the standard content specified in chapter 7.1 – 7.3:

I/O 1 (password 2 – user programmable)									
Parity	SEQSEL	(DB)	B0T	B1T	B2T	B3T	B4T	B5T	
1 bit	2 bit	1	2 bit	2 bit	2 bit	2 bit	2 bit	2 bit	
0	1	3	4	6	8	10	12	14	

I/O 1 (password 2 – user programmable)						
	Cal	On level	Color of timeout and panic light			
	1 bit	3 bit	3 bit R	3 bit G	3 bit B	3 bit W or A
	16	17	20	23	26	29

I/O 2, LED Calibration factors – bin correction (password 3 – factory programmable)							
Parity	Output 0 Red	Output 1 Green	Output 2 Blue	Output 3		Output 4 Artificial Amber (future Green 2)	Output 5 Artificial Cyan (future Blue 2)
				Wh	White or Amber (future Red 2)		
1 bit	5 bit	5 bit	5 bit	1 bit	5 bit	5 bit	5 bit
0	1	6	11	16	17	22	27

I/O 3 (password 3 – factory programmable)															
Parity	SEQ	DBE	B4ES	B5ES	B6ES	B7ES	B0D	B1D	B2D	B3D	B4D	B5D	CPOL	CPHA	
1 bit	2 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	
0	1	3	4	5	6	7	8	9	10	11	12	13	14	15	

I/O 3 (password 3 – factory programmable)											
	SPILength		OutSel	B20V	Gamma	Filter	B0L	B1L	B2L	B3L
	(OutSel)	3 bit	2 bit	1 bit	1 bit	4 bit	1 bit	1 bit	1 bit	1 bit
	16	17	20	22	23	24	28	29	30	31

Fig. 7e.5

The two SEQSEL bits are used to limit the sequence according to this table:

Code	Operation mode
11 (default)	Normal sequence
10	Free for future purpose
01	Only All-on and All-off. All other commands disabled.
00	Only panic light on and off. All other commands disabled.

Fig. 7e.6

The Cal-bit specifies whether bin-correction shall be used (default 1) or shall be disabled (0).

DB, BxT and BxD are used in the same way as for the BOOLEAN object except that the debounce time is always short if a quadrature encoder is used. **Note that the BxT and BxD bits must be left in the default 1-state if the publisher data type is LAMPCTRL!**

Bit 17 – 19 defines the level of the lamp when it is turned on (not night light) after it has been off. The 8 levels are the same as the fixed levels defined in the LAMPCTRL data type except that they are not gray coded but binary.

Bit 20 – 27 of I/O 1 is used to define a temporary emergency color, which may be shown in case of a communication (watchdog) timeout or if a panic button is pressed.

As white light is scattered in case of smoke, reddish light may be appropriate in case of fire, and for a panic light it may be desirable to reduce the light level.

Since automatic lamp flashing is not relevant and the hour counter shall just count while the lamp is on, bit 24 – 31 of I/O 1 are not used and therefore free for this purpose.

Each 3-bit group (ABC) shall be expanded as defined in the LAMPCTRL data type and shown below:

On level			Red			Green			Blue			White		
ABC	ABC	AB	ABC	ABC	AB	ABC	ABC	AB	ABC	ABC	AB	ABC	ABC	AB
8 bit			8 bit			8 bit			8 bit			8 bit		
			0	3	6	8	11	14	16	19	22	24	27	30

Fig. 7e.7

I/O attribute 2 is used to specify the multipliers for bin correction. The Wh (White) bit is used to select the function of channel 3. If Wh is the default 1, it is used for white and count up and down shall be enabled. If WhAm is 0, channel 3 is regarded as a color channel like red, green and blue and may for example be used for amber. In this case, counting shall be disabled.

Since a minimum light level is not needed for lamps, bits 16 – 21 are free for this purpose.

CPOL and CPHA of I/O 3 are used to select the type of SPI interface as specified under the SPI object.

The B20V bit of I/O 3, which is programmed by the vendor, is used to select between monochrome or dichromatic lamps (default 1) and full-color lamps (0) depending on the possibilities, outputs and wanted length of SPI transfer of function data (20 bit for monochrome and 36 bit for full color). This also defines the minimum data length. It is used together with the B20U bit of attribute 7, which specifies the number of bits to receive if this is more than specified by B20V. This makes it possible for a dichromatic or monochrome lamp to receive full-color messages and in this way share messages with full-color lamps. If B20V and/or B20U are 0, the device expects to receive 36-bit messages and a dichromatic lamp shall just ignore red and green and use blue to control cold white and use white or amber to control warm white. A monochrome lamp shall ignore all colors except for white.

Since a 20-bit message is not enough for a full-color lamp, the data length is set by the most advanced lamp in a system. There is no reason to buy a full-color lamp if it is not useable to its full extend.

Bit 16 – 19 are used to specify the length of the SPI transfer in case of (mechanical) functions. Bit 16 is reserved for future SPI transfers longer than 36 bit and should therefore be set to 0 for future compatibility. The coding of bit 17 – 19 is specified in the SPI object. They shall usually be set to 20 bit (011B) for monochrome and dichromatic lamps (B20V = 1) and to 36 bit (111B) for full color lamps, but may also be changed to a byte boundary if the receiver does not accept nibbles.

The two OutSel bits together with bit 16 are used to select the function of output 4 – 6 as shown in the I/O map in the beginning of this chapter. In most cases, bit 16 is not used, but if output 4 and 5 are programmed for status output for the auxiliary lamp, bit 16 is instead used to distinguish between off-status and white with square root gamma (1.56).

This makes it for example possible to make a dim-to-warm bed lamp where the main channel is used to control its own lamp, which requires the white gamma1.56 output, and the auxiliary channel is for example used for group control with on-status indication (but no off indication), which may be used to switch the lamps to the bathroom on.

Codes 1x are used for non-SPI devices and codes 0x are used when the SPI interface is use to transfer data for mechanical functions like pan, tilt, zoom, focus, gobo selection etc. to a connected microprocessor for handling. This is very useful for stage lamps. Since such lamps shall be able to respond very fast, the power supply or current generator must be always-on so a signal to enable and disable this it is no longer needed on output 6.

Output 7 shall go high as soon as there are pulses on one of the active color channels (2 for monochrome or dichromatic lamps and 4 (6) for full color lamps) and go low after 1 – 2 minutes without activity. This is used to enable and disable a switch-mode current generator if the PCM is done by means of a MOSFET, which short circuit the LED's when no light is wanted, but keeps the current generator on.

In some cases it may be desirable to use both dim-to-warm and the SPI interface simultaneously. In this case, code 01 may be specified and the slave/chip select signal (SS/CS) may then be generated from the clock signal by means of external logic. This is fairly easy as the clock is an unbroken pulse train with a duty cycle of 50 % and the programmed number of pulses. Since the transfer of 36 bit at the clock frequency of 500 kHz only takes 72 µs and the shortest possible 36-bit message in average takes 227 µs at a maximum line length of 90 m (1 Mbit/s UART speed), there will always be an easy detectable pause between each message, which can be used as telegram delimiter instead of SS/CS.

In future versions of the Max-i controller there will be at least 10 outputs so that this compromise is not necessary.

The two SEQ bits of I/O 3 is used to select the operation mode:

Code	Operation mode
11 (default)	2 x one-button toggle sequence
10	2 x two-button up/down control
01	(24-pulse, 24-detent, rotary) quadrature encoder plus one-button toggle sequence
00	(24-pulse, 24-detent, rotary) quadrature encoder plus two-button up/down control

Fig. 7e.8

7e.6 Lamp Control and Synchronization

To be able to make multi-way landing-switching, all buttons shall be synchronized as if they were parallel connected. If for example a lamp or group of lamps is turned on from one button, a subsequent similar activation of the same or another button shall turn it off, and if one button has stepped the light up, a subsequent activation of the same or another button shall step it down. To make this work, two things are necessary:

- All buttons for a lamp or a group of lamps must of course have the same publisher ID and signature, but a different base address to make them individually programmable, but the subscriber ID and signature and with that the data type (LAMPCTRL) also need to be the same even if they generate group messages so that the sequencer in the publisher can read back its own command or the corresponding command from other buttons and determine the next command. Note that even if the control button is located on a lamp, there shall never be any direct connection between the button and the lamp.

  The I/O Object specification in attribute 12 must therefore be FF 7E FF 40 H if GRPTYPE = 1 and 7F 7E DF 40 H if GRPTYPE = 0 (Parity and standard priority and line hold – 111B or 011B, LAMPCTRL – 1.1111.01B, LAMP object – 11.1110.11B, GRPTYPE – 1 or 0, LAMPCTRL – 1.1111.01B, no extra blocks – 00.0000B).

  Note that when the publisher(s) are programmed to send group messages, the value ID in attribute 4 (and/or 6) is just replaced with a group ID and group messages are then just received as ordinary values.

  Usually, there is only one publisher in each Max-i controller, but because the publisher and the subscriber must use the same ID to make the synchronization work, they can share the same ID register. This makes it possible to squeeze two publishers into one Max-i controller and in this way make switches with a single Max-i controller, which can control two lamps and in this way replace ordinary double wall switches. It also makes it possible with two implicit identifiers plus the group identifier for the subscriber so that there are 3 addressing modes instead of 2.

- It must be possible to determine the lamp status entirely by means of the transmitted commands.

In case of 4-bit messages, this is only done by means of messages, which set a fixed level (1110B – 1000B), and except for the quadrature encoder, which only works for the main lamp, down counting shall just stop without turning a lamp off if the minimum level is reached.

In case of long messages, the reception delay makes it possible to send different data to more lamps in the same telegram so one lamp may for example have all colors set to zero and therefore could be regarded as "off", but another lamp with the same ID may have non-zero values and shall therefore be regarded as "on". In case of down-commands, it is also possible for more lamps to reach zero or a given minimum level at different times, so it is impossible to determine a common state. For this reason and to avoid that the publisher sequencer for the auxiliary lamp channel must (also) have a full blown subscriber, which is able to receive an entire 36-bit message with reception delay and be able to count the white channel up and down, **all** long **implicit** messages shall set the lamp or group of lamps to on-status and color-mode no matter the data and any reception delay, and like 4-bit messages, down-commands shall never turn a lamp off. If the lamp is in white mode, count down shall therefore be disabled below the minimum level, so that the lamp does not turn off and is not forgotten in almost off state, but in color mode where the lamp is always regarded as "on", it is allowed to dim the white or amber channel to zero. The ramp direction shall always be down after a long message.

To avoid that a publisher polls itself or another publisher with the same identifier, implicit poll shall be switched off when the publisher is programmed to LAMPCTRL, but explicit poll by means of the base address shall still be enabled.

7e.6.1 One-button Control

The sequencer shall be able to distinguish between a short and a long activation of the button. If the input pulse is shorter than approximately 0.4 s, the lamp(s) shall be switched on to the programmed start level (1111B) if they were completely off before, and switch the lamp(s) off (0000B) if they were on before – no matter if the last operation was ramp up or ramp down. The operation shall take place when the button is released (before that it is not possible to determine the pulse length). Note that nothing prevents a light level above the start level if this level is less than 100 %.

The short 4-bit messages, which shall be transmitted in case of a pulse longer than 0.4 s, depend on the present state.

- If the lamp or group of lamp is in off state (0%), the publisher shall transmit **a single** implicit 4-bit 0010B command, which shall cause the lamp(s) to step to the minimum level (night lighting) or cause a window to open the ventilation slot. A window should also use a kind of gamma correction so that the minimum level corresponds to approximately 1% instead of 15%.
- If the lamp(s) are in minimum state, a repetitive transmission of up messages shall take place until the button is released. This shall cause the lamp(s) to step up towards 100% where they shall stop so that any further up-messages are ignored. The repetition rate shall be approximately 12 messages per second (80 ms delay), so that it takes approximately 4.3 seconds to transmit the necessary 54 messages to step from 15% to 100%. The heartbeat timer (with a fixed time) may be used for this purpose. In this case, it is actually the time between messages, which becomes 80 ms – not the actual repetition rate. For high baud rates this makes no difference, but for the lowest baud rates, the ramp may be slowed down, but this is acceptable. Note that to step the level up from 0% without going through the start level, two long pulses are necessary. In this way, it is easy to activate night light without activating the up-ramp after 0.4 seconds. The button is simply released when the lamp(s) turns on at minimum level. If this level is too low, one more long pulse will ramp it up.

  In case of windows, one up-command may be enough to start opening the window until a stop command is received.
- If the lamp(s) are at maximum (100%), a repetitive transmission of 4-bit 0101B down-messages shall take place while the button is depressed. This shall cause the lamp(s) to step down towards minimum where it shall stop. In case of windows, one command may be enough to start the movement.
- If the level is between minimum and maximum, the ramp direction shall change for every new pulse, so that if for example the last ramp direction was up, the next ramp direction shall be down and vice versa.
- If one of the 8 fixed levels is set, the ramp direction shall be up after one of the 4 low levels (0 % – 43 %) and down after one of the 4 high levels (57 % – 100 %).

If the publisher ID is programmed to group 255 (common group), the long-time-activation sequencer shall be switched of so that only All-off (0000B) and Back-on (1111B) can be generated!

The dimming sequence with the transmitted messages is illustrated below, but the ramp speed is exaggerated:

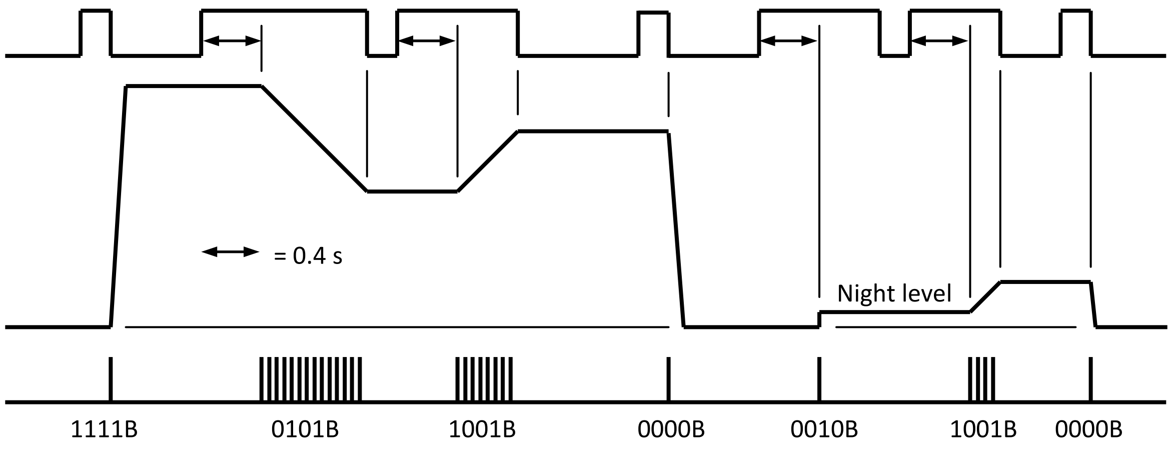


Fig. 7e.9

The reason why repetitive telegrams are used to dim lamps is that the accuracy of the internal clock may not be very good (±6%), so if more lamps just ramp up or down until a stop message is received, they may not reach the same level and may not even regard the command in the same way (short or long pulse).

The fixed minimum level in white mode ensures that all devices – publishers and subscribers – agree on the number of steps between minimum and maximum (54). If the level was programmable (not defined), it would not only make the circuit a lot more complicated, but a publisher from one manufacturer may not fit with a subscriber from another manufacturer.

7e.6.2 Two-button Control

Two-button control shall work in very much the same way as one-button control except that it shall only be possible to turn a lamp on and step the light up by means of the up-button and only turn it off and step the light down by means of the down button. A short pulse on the up-button shall set the lamp to the programmed start level no matter the present level, and a short pulse on the down-button shall turn it off.

7e.6.3 Quadrature Encoder Control

This is used for “analog” control by means of a rotary or sliding knop. The waveform for the quadrature encoder is shown below:

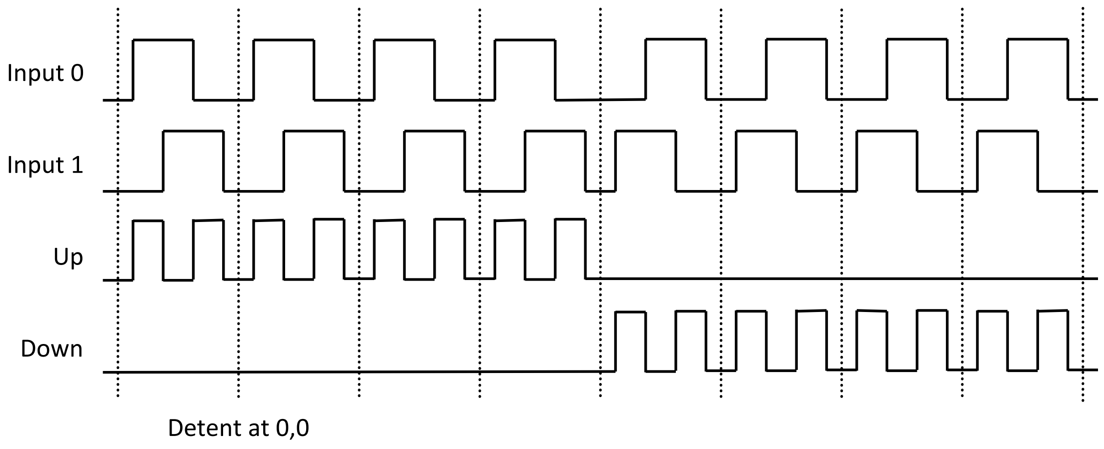


Fig. 7e.10

Since 2 up or down pulses are easily generated between each detent on the basis of the quadrature inputs without the use of a clock, and the lighting controller needs 54 pulses/messages to step from minimum level (15 % linear, 0.8 % gamma) to 100 %, a 1⅛ rotation is needed to turn a lamp fully on if a 24-pulse encoder is used.

If the lamp is off, the first up-pulse shall set it to minimum level (night light) and further up-pulses shall count the level towards 100 % where it shall stop. If the lamp is on, the lamp shall be switched off if the light level is counted down below the minimum level. To be able to do that, the publisher must know when to transmit an off-command instead of a down-command, but since there is only one (full) subscriber, there is only one up/down counter with minimum level comparator, so it is only possible with one quadrature encoder for each Max-i controller – plus either a one-button or two-button control.

Note that because the up/down counter is used in this case, the filter speed in bit 24-27 of I/O 3 (attribute 11) should be set to 0110B to emulate the ramp in a standard lamp as accurate as possible!

*A combination between a quadrature encoder for lamp 0 and a button function or a differently working quadrature encoder function (with off-switch) for lamp 1 will ether look silly or be un-logical and therefore probably never be used, but since the subscriber is controlled by **both** channels when the publisher data type is LAMPCTRL, channel 1 may be used as a supplement to channel 0. In this way it is possible to use a momentary push-on switch in the quadrature encoder to trigger all ordinary one-button functions so that it for example is possible to toggle between off and the programmed start level by means of a push and then adjust the level by means of a rotation.*

7e.6.4 Use of Dual Controllers

As described above, it is possible to use control two lamps from the same controller. This possibility simultaneously enables three addressing modes for the subscriber – two implicit and one group, but if the device is a lamp where the subscriber is utilized, it is recommended with only one publisher as two publishers may lead to unpredicted behavior.

*This is because the two 2-bit status registers of attribute 2 have no knowledge about the status of the other channel. If for example the main lamp status is "on" and the lamp is lighting, but the auxiliary lamp status is "off", an activation of a toggle-button for the auxiliary lamp will not transmit an "off" command as expected due to the light, but an "on" command, since it regards the lamp as "off". A subsequent activation will however turn it off as the status is now "on". In the same way, if a lamp has been turned on by the auxiliary channel, but then has been turned off by the main channel, a long activation of the auxiliary channel will not transmit a night-level command as expected, but a row of up or down commands, which however will be ignored by the subscriber since it has been turned off. If the auxiliary channel uses two-button control, it may therefore look as the system is not responding to the on-button no matter how many times it is activated. A subsequent "off" command on the off-button will however "reset" the auxiliary channel and enable the on-button again. Note however that it is only the publisher, which may transmit a command, which may **seem** wrong when you look at the lamp, but actually is correct. The subscriber just receives and respond to all commands no matter which of the two identifiers is used.*

7e.6.5 Fixed Level Control

If input 3 and/or 7 is set to 1, the 3 other inputs of the corresponding channel is used to set one of the 8 fixed levels by means of the gray code as specified above.

7e.7 Control of AC/DC-converters

In some cases, 20 V is not enough or undesirable to drive a lamp so mains-plus-dimming must be used as described in layer 1. In this case, output 7 may be used to disable the mains-converter when all used outputs are low and in this way save some power, increase the service life and reduce the fire risk. This is especially important if the converter generates a constant current and the LED dimming is then made by means of a short circuit of the LED's.

Since power is voltage multiplied by current and the voltage drop over the LED's is very low during a short circuit (no light), this method is very efficient if a switch mode current generator is used and unlike the traditional method, it does not need an output capacitor, which increases the lifetime, and does not lose power on voltage drops over linear current generators. Besides, since the current is always maximum, only a small inductor is needed, but the high current even in off-state makes it desirable to switch the converter off when it is not needed.

If the lamp is monochrome or dichromatic, output 7 shall go high as soon as there are pulses on the blue or white channel, and if the lamp is full color, the red and green channel shall also be included. Output 7 shall go low (again) after 1

– 2 minutes without pulses on any of the selected channels. Note that in case of a monochrome lamp, the switch-off can be disabled by means of a dummy output on the blue channel.

Since the two artificial colors are generated on the basis of red, green and blue, they cannot generate pulses if these source channels are 0, so it is not necessary to include these channels in the OR-gate.

Layer 7f, A/D and D/A converter object

The publisher of this object is an A/D converter, which uses the I/O's in combination with a simple and cheap external circuit to generate FIX20 process values directly in SI or US/imperial units. The subscriber is either a standard 4-bit boolean output (data type TIME) or a 16-bit or 20-bit D/A converter, which generates a PCM output, which may be filtered to an analog signal (data type RAW).

Object name and ID #	ADDA, -6 (11111010B).
Publisher data type(s)	SFIX (signed) or UFIX (unsigned).
Subscriber data type(s)	URAW (unsigned) or TIME.
Input 0	Input from external part of A/D converter.
Input 1	Trigger of transmission of analog value.
Input 2	(TBD)
Input 3	(TBD)
Input 4	Error or Status 0 input.
Input 5	Error or Status 1 input.
Input 6	Error or Status 2 input.
Input 7	Error or Status 3 input.
Output 0	Bit 0 (MSb) of boolean output (TIME).
Output 1	Bit 1 (MSb) of boolean output (TIME).
Output 2	Bit 2 (MSb) of boolean output (TIME).
Output 3	Bit 3 (MSb) of boolean output (TIME).
Output 4	Non-inverted output to external part of A/D converter.
Output 5	Inverted output to external part of A/D converter.
Output 6	Non-inverted 16-bit or 20-bit PCM coded D/A output (RAW).
Output 7	Inverted 16-bit or 20-bit PCM coded D/A output (RAW).
Output 8	Communication timeout (watchdog).

Fig. 7f.1

7d.1 A/D converter

The A/D converter is a kind of synchronous sigma-delta voltage to frequency converter, which consists of an internal logical part and an external analog part.

If a 5-V comparator is wanted, the external part may for example use the following schematic:

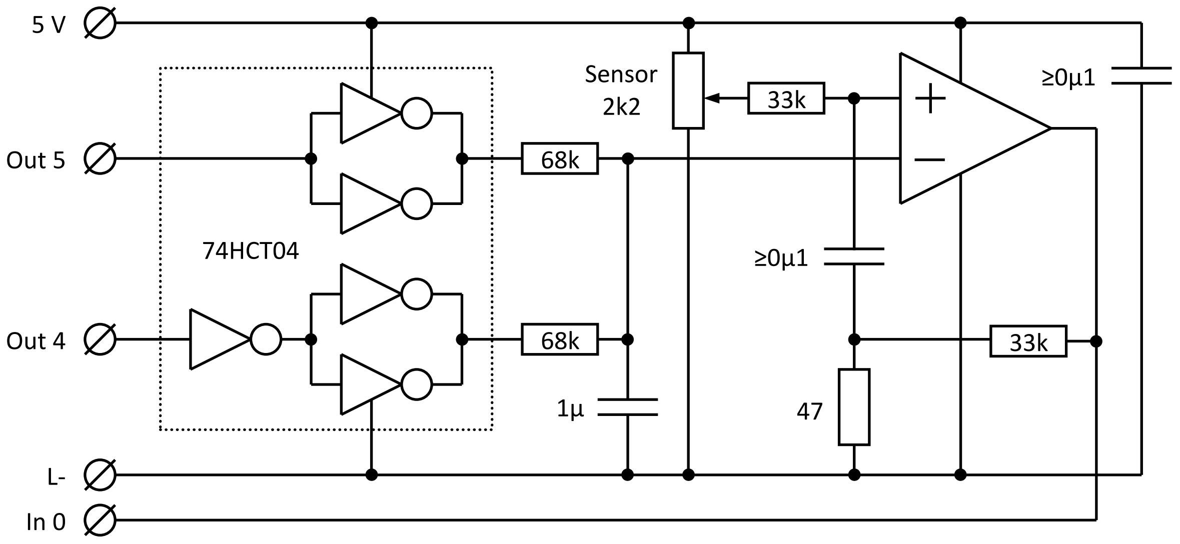


Fig. 7f.2

The working principle is very simple, stable and accurate – in the order of 15 bits, but only allows a few readings per second, but this is usually enough for most industrial sensors for example for temperature, level, pressure and current. If the input voltage is above the reference voltage on the 1 µF capacitor, the output of the comparator goes high. This causes the controller to drive output 4 high and output 5 low so that both buffer outputs go high. This charges the capacitor. When the capacitor voltage becomes above the input voltage, the comparator output goes low, which causes output 4 to go low and output 5 to go high so that the capacitor is now discharged. In this way, the circuit oscillates, but in such a way that the ***average*** *voltage on the capacitor is exactly equal to the input voltage. The comparator can be almost any type* ***without internal hysteresis*** *– even slow types with a response time up to 1.5 µs for a 5 mV overdrive. The necessary hysteresis to ensure a clean switching without oscillation is instead implemented externally and AC-coupled through the 0.1 µF filter capacitor so that it does not affect the accuracy. In this case, a ±7 mV hysteresis is chosen (rail-to-rail output).*

The CMOS inverter 74HCT04 work as buffers, wave shapers and level translator's, which convert 3.3-V output levels to 5 V. The two channels work in parallel except that output 5 of the controller is inverted and inverted again in the buffer circuit. The purpose of this is to balance out any difference in propagation delay between the rising and the falling edge, as the accuracy of the converter primary depends on the accuracy of the pulse width plus the offset voltage of the comparator. Because the outputs may be asymmetrical in case of for example a galvanic separation by means of optocouplers, it cannot be guaranteed that the propagation delay on the rising edge and the falling edge is equal. If for example the rising edge is delayed more than the falling edge, a pulse from output 0 will be too short. However, because output 1 is inverted, this pulse will be correspondingly longer as it also has the rising edge delayed so that the net result is a very accurate charge or discharge as long as the two channels are reasonable equal. Because the two buffer outputs are high or low simultaneously most of the time, the two resistors need not be highly accurate. Standard 1 – 2 % components are enough. Note that propagation delays, which affect both the rising and the falling edge, are completely unimportant. The important thing is only the accuracy of the total charge and discharge times.

If a 3.3 V converter is wanted, no galvanic separation is needed, and the used Max-i controller has high-speed 3.3 V rail-to-rail outputs, the buffer may be omitted and output 4, may simply be connected to the 1 µF capacitor through a 33 kΩ resistor.

The shown A/D converter measures a voltage ratio illustrated with the potentiometer. For most practical applications, the potentiometer must be replaced by an instrumentation amplifier and for example a measuring bridge.

To be able to measure the value without the usual digital decimation filter, the output signal shall be a precise integer number of clock cycles, and to reduce the number of edges, the output should be delayed two clock cycles. The delay circuit shall include a selectable debounce circuit, which may guarantee that the pulse width will never be shorter than two clock cycles. If the Debounce bit B0D is 1 (default), the circuit shall be enabled. If the bit is 0, there is still a two-cycle delay, but the debounce is switched off so that the pulse width on rare occasions may be as short as one clock cycle. In some situations, this gives a slightly higher accuracy although the difference is usually very small – below 2^{-15}.

To be able to scale the signal to SI units, it is important to avoid the decimation filter, because this filter defines the full-scale value to 2^N.

Note that unlike dual-sloop converters, the comparator delays (rising and falling edge) are completely unimportant and are even further increased two clock cycles by the controller. It is therefore possible to use slow comparators, which are much easier to use than “hysteric” high-speed comparators.

The relationship between the input voltage and the supply voltage is read by counting the number of high clock cycles within a given gate time. By varying the gate time and the preload value of the counter, it is possible to adjust the full-scale level so that **any** value may be converted to SI units and offset compensated without the use of multipliers, adders or any hardware adjustments! If for example a temperature sensor gives an output of 0.873% of the supply voltage per °C (SI unit), the gate time shall be 2^N / 0.00873 – for example 2^{12} / 0.00873 = 469187. At a clock frequency of 500 kHz, each clock cycle is 2 µS, so that the gate will be open for approximately 0.94 seconds and therefore allow approximately one reading (telegrams) per second. At 100 °C, 87.3% of the maximum 469187 cycles will be counted, which gives a value of 409600 equal to the binary 20-bit value 0110,0100,0000,0000,0000B. If this value is regarded as a SFIX20 value, that is, a signed, left-shifted, two’s-complement, fractional number in the range from -1 to $1-2^{-17}$, it is necessary to move the radix point 7 places to the right from 0.1100100000000000000B to 1100100.000000000000B to turn the result into SI units. The exponent shall therefore be 7. The preset value of the counter is 0 ±4096 (2^{12}) for every °C of offset so that an offset down to 1/4000 °C is possible.

7d.2 D/A converter

The D/A converter takes a 16-bit or 20-bit RAW value and converts it to a linear PCM signal – see LAMP object except that the maximum value shall not be 1 as it is the case for 8-bit lamp values, but the true binary values 65535/65536 or 1048575/1048576.

The 16b bit is used to select between 16 bit and 20 bit mode.

- If the bit is the default 1, only the most significant 16 bits shall be used for the conversion and the remaining 4 bits shall be used to select the speed of the smoothing filter in exactly the same way as for lamps used for lighting. In this mode, it is possible to transmit messages to more devices in the same telegram as it is also the case for lamps.
- If the bit is 0, all 20 bits are utilized for analog value and the smoothing filter shall have the programmed speed. This mode can only be used with single-device messages.

Because the smoothing filters for lamps are only 8 bit wide and may not be cascade connectable, it is allowed only to filter the most significant 8 bits. This also shortens the time to 100 %. In that case, the remaining 8 or 12 bits shall bypass the filters.

Note that a simple infinite impulse response filter like the one used in the Innovatic implementation needs a fairly long time to settle to 100 % – in principle infinite – hence the name. This may affect the accuracy for quite some time. With the 3rd order Innovatic filter, up to 24 steps may be needed so at the fastest filter, which has 10,000 steps per second, it may take up to 2.4 ms.

Because of the PCM, the repetition frequency depends on the resolution. If all 20 bits are utilized, the repetition frequency may be as low as 0.75 Hz at a pulse-duration of 1.25 µs, but for each time the resolution is reduced by one bit (least significant bits fixed to 0), the repetition frequency is doubled. In case of for example industrial standard 12-bit resolution, the repetition frequency is therefore 195 Hz if the 8 LSb’s are 0, and if the fastest filter is used, it typical takes less than 2.5 ms to settle to 100 % if there is no additional filtering, which however usually is the case to convert the pulse train to a smooth analog value.

To improve the accuracy, output 7 shall be the inverse of output 6 so that it is possible to compensate for differences between rise time and fall time in the same way as for the A/D Converter.

7d.3 Boolean outputs

In this mode, the outputs works like the Boolean object instead of an analog output and the subscriber data type is TIME.

This mode may for example be very useful to connect solar panels one-by-one as shown on the topology diagram in chapter 1. The analog input can then be used to measure the generated current for each panel.

7d.4 Programming

I/O attribute 1, 2 and 3 are used to program the A/D converter as shown below.

I/O 2, A/D converter gain (password 3 – factory programmable)		
Parity	Gate time	LSb's of offset
1 bit	23 bit	8 bit
0	1	24

I/O 3 A/D converter offset (password 3 – factory programmable)												
Parity	16b	NU	Sign	B4ES	B5ES	B6ES	B7ES	B0D	B1D	(B2D)	(B3D)	
1 bit	1 bit	1	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1	1	
0	1	2	3	4	5	6	7	8	9	10	11	

I/O 3 (password 3 – factory programmable)						
	MSb's of offset	Filter	B0L	B1L	B2L	B3L
	12 bit	4 bit	1 bit	1 bit	1 bit	1 bit
	12	24	28	29	30	31

Fig. 7f.3

The gate time and offset may be used to (automatically) calibrate the sensor so that a low-cost instrumentation amplifier may be used before the A/D converter.

As specified in layer 6a, the heartbeat timer controls when the data is transmitted. If the timer is FFH (default), the value is only polled. For other values, a new telegram shall be transmitted when the timer runs out AND a new value is ready so that the repetition rate will never be faster than the conversion rate **even if the timer is set to zero.**

Input 1 is used as a trigger signal, which causes an immediate transmission of the value so that the device can be used in auto-programming mode and event driven transmissions are possible. This input may for example be connected to a pencil-activated button. Bit B1D determines whether the signal shall be debounced (1) or not (0).

The Sign bit is used to specify an unsigned (0) or signed (1) value.

B0L – B3L controls whether the 4 main outputs are pure boolean (1) or outputs with ramp up and down (0). **In case of RAW, these bits must be left in the default 1 state since ramps are not possible!**

Annex A. Transmission Lines

A.1 Principle of Line Interface

Unlike most other fieldbus systems, Max-i is usually **not** terminated with resistors, but with voltage clamps except for high-speed communication on long lines and/or use of coupling transformers. The figure below shows the two principles with wave shapes for both ends:

- The traditional way with resistor termination.
- A simplified Max-i interface.

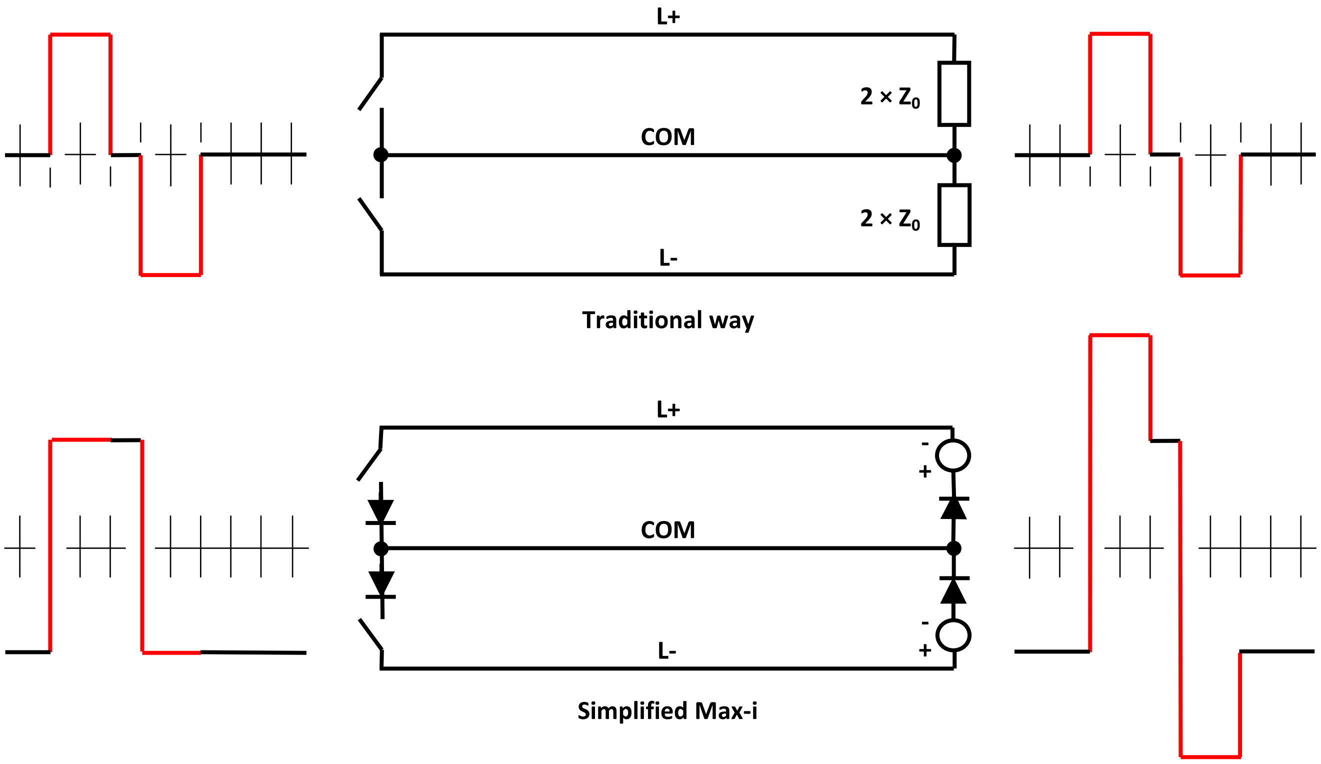


Fig. A.1

The transmitter is driving the line during the red parts of the waves. In the black part, the line is either held by the termination resistors (traditional way) or by the line capacitance in combination with a (not shown) line hold circuit as specified in Layer 1.

To enable multiple-access (CSMA-CD) and prevent that more transmitters counteract each other, there is a (variable) pause (black) **before** all pulses and telegrams where both switches are off. In this period, it is possible for a (new) device to obtain the line. Max-i distinguishes between 0-bits, 1-bits and start-bits by means of the pause length as specified in Layer 1.

In most traditional systems, the line is terminated with a resistance equal to the characteristic impedance (Z_0) of the line to avoid reflections as shown in the upper drawing. If there are no losses in the line and the transmitter switches, the output signal will be equal to the input signal – just delayed as shown. The transmitter power is $(L+ - L-)^2 / (4 \times Z_0)$, so if the supply voltage is 20 V and Z_0 = 50 Ω, it will be 2 W and this is also the power loss in the termination resistors as the entire transmitter power is lost (no reflections).

However, if more devices are driving the line simultaneously during bus arbitration, the wave (described in next chapter) that travels from one transmitter A towards another B will cause transmitter B to turn more or less off when it arrives. When this happens, the waves (in both directions) from transmitter B are reduced temporary as long as the wave from transmitter A drives the line. As this will also happen for transmitter A when the wave from B arrives, the result is that all devices **between** A and B will **not** see a clean signal, as could be expected, but an oscillation, which is only damped by resistances in the transmitters and the line, continues as long as the transmitters are active and has a frequency, which depends on the distance between the colliding devices.

In the worst-case point in the middle between the colliding devices, the duty cycle of the oscillation signal is 50 % as shown in the computer simulation of a low-loss system below:

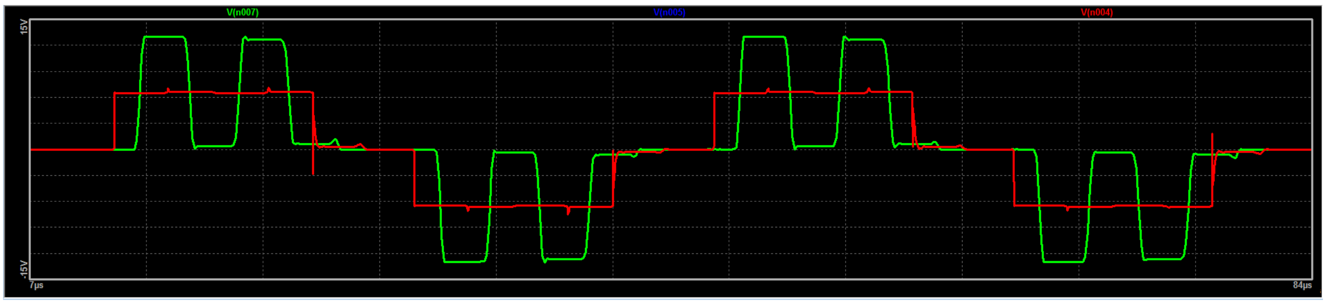


Fig. A.2

The peak value is almost twice as high as the transmitter signal since the two waves add together, and it falls to almost zero when both transmitters are turned almost off by the wave from the other transmitter. This is obviously not the desired signal although the mean value is correct. The signal ought to have the same waveform and amplitude as the driving signals (red and blue on top of each other) – just delayed, and it makes it very difficult to determine whether the signal is logical high or low – especially in case of a unipolar signal with dominant and recessive states where the pulses may pass the threshold level more times during the bit time, **so a non-destructive bit-wise bus arbitration may not work!**

If some series resistance is added in the transmitter, it is possible to damp the oscillation enough to be able to detect the signal properly, but the series resistance simultaneously reduces the amplitude of the signal and therefore reduces the signal/noise ratio considerably as the transmitter power is proportional to the amplitude in the power of 2. In for example CAN where a 120 Ω transmission line is used, it is necessary with a series resistance of approximately 20 Ω in each half of the transmitter (a total of 40 Ω) to damp the oscillation enough as shown below with a simulation of a LTC2875 CAN transceiver, and because the transmitter always looks into the half line impedance (60 Ω), the signal amplitude is reduced to only 60 % (2 × 1.1 V = 2.2 V) and the signal power therefore to only 36 % or nearly ⅓.

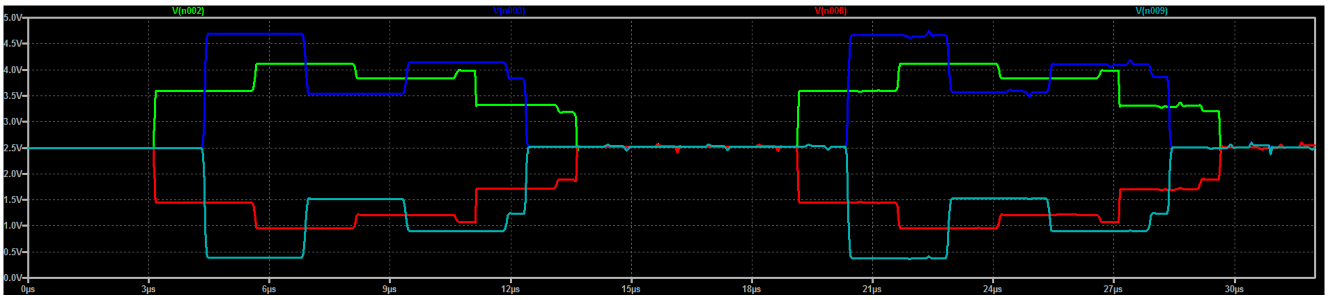


Fig. A.3

The red and green waves are the driving signals and the blue ones are the waveform in the middle between two driving devices in each end of a lossless 500 m line (maximum length) at 125 kbit/s. The damped oscillation is clearly visible. With more powerful transmitters, the signal/noise ratio (power) could have been almost 3 times better at the same supply voltage (5 V), but it would then be impossible to guarantee that the bit-wise bus arbitration works on long lines. If more than two devices are driving the line, the drive impedance may be reduced correspondingly, which will lower the damping, so in that case, the bus arbitration may not work.

In the Max-i interface, the line is instead terminated with one or more voltage clamps (one in each device), which limits the reflected wave to approximately the half of the transmitter voltage. In this way, the reflections are utilized to charge the line and convert the signal to a square wave, which is much more powerful and easier to detect (simple zero crossing). To ensure that the line voltage can exceed the transmitter voltage and that the transmitter(s) turn off, when the reflections or the waves from other devices arrive, each half of the transmitter includes a serial diode, which also reduces the output capacitance and protects against reversed polarity. In the ideal case with no transmitter and line loss and ideal diodes, the transmitter power is 8 W at 20 V if the characteristic impedance of the line (Z_0) is 50 Ω, but half of the transmitter current is returned to the power supply through the clamp and therefore not lost – just borrowed for the time it takes the signal to travel to the end of the line and back again. The other half is used to charge the line. The power loss in the clamp is therefore 10 V × (20 V / 50 Ω) / 2 = 2 W, which is exactly the same as in the resistor terminated case, but since the transmitter power is 4 times higher, Max-i is 4 times more power efficient in this ideal case! In practice, the gain in efficiency is somewhat lower due to losses.

In both cases (traditional and Max-i), the power loss is multiplied with the on-time of the transmitter, but in Max-i, the transmitter only draws current for the time it takes the signal to travel to the end of the line and back again, so if the line is shorter than the maximum length for a given speed, the energy loss is reduced correspondingly, which further increases the energy efficiency. This also reduces the EME and has the beneficial side effect that the power loss is the same no matter where the device is located on the line as it is the case with resistor termination. If it is located in one end, it will only look into the characteristic impedance of the line, but will draw current for the maximum time. If it is located in any other point, it will initially see two lines in parallel and therefore initially draw the double current, but as soon as the first reflection arrives, the current is reduced to the half and when the next reflection arrives, the current is reduced to zero so the total power loss is the same. A device in the middle of the line just draws the double current for half the time.

If more devices are driving the line simultaneously, the waveform between the devices will still be a damped oscillation as with resistor termination, but only until the reflection from the farthest end arrives (in the resistor terminated case, there are no reflections to stop the oscillation).

This situation will also occur if a single device is driving a line, which is not terminated in that end. In that case, the waveform towards the unterminated end will oscillate because the transmitter will just receive its own signal reflected from the line end instead of a signal from another device, but there is no oscillation towards the other (terminated) end.

However, if a device is added and the waveform therefore becomes important, the clamp network in that device will dissipate approximately half of the energy in the colliding waves and with that reduce the duty cycle of the oscillation pulses considerably.

If a device is added in the middle between two colliding devices, the oscillation is almost entirely removed, but with other locations, there is still some oscillation left with much reduced duty cycle as shown below for one of one or two devices located in 1/3-points between the driving devices, which seem to be the worst-case situation:

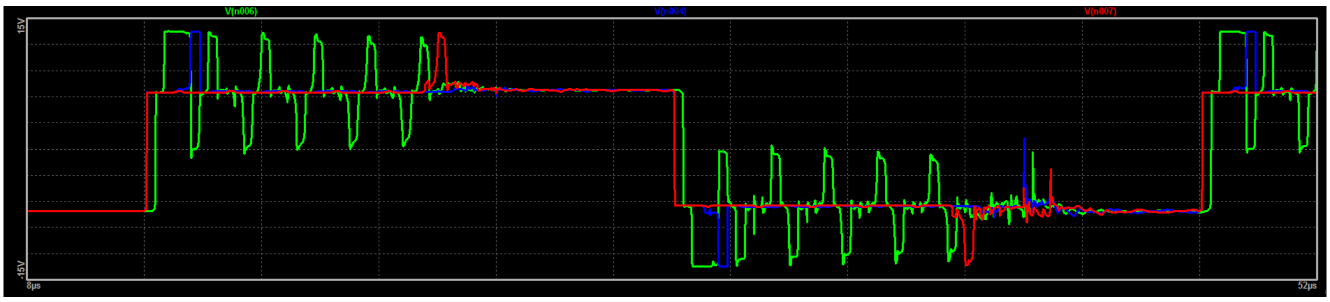


Fig. A.4

Since the oscillation stops when the reflection from the farthest end arrives, wide pulses like the ones in the resistor termination example are not possible because a long distance between colliding devices will reduce the maximum possible distance to the end of the line. The widest pulse seems to be the one shown below:

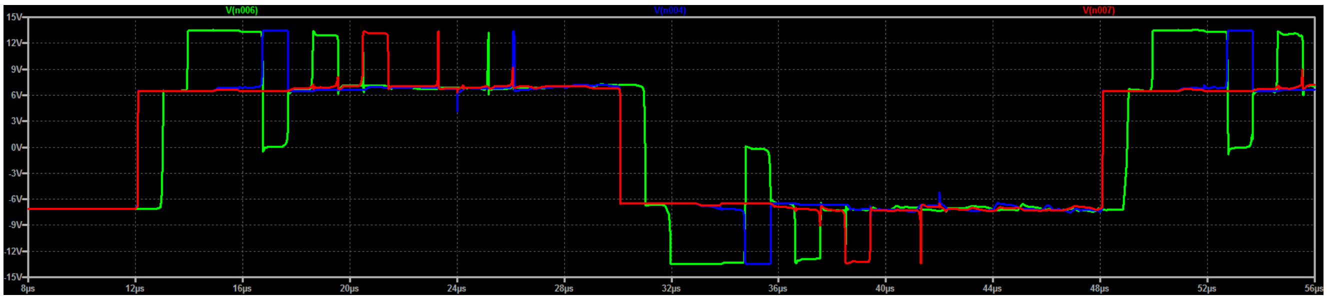


Fig. A.5

In this example, one device A is placed in one end of a maximum length line, the other B in the middle and the detection point is 1/6 line length from A, but could equally well be 2/6 (1/3-points). This creates a pulse width equal to 1/6 of the propagation delay of the line (1/6T). The very narrow spikes on the figure without a clearly visible pulse width are due to reverse recovery in the clamp diodes, which for this reason should be schottky diodes.

It is obvious that the detected waveforms (green) are **much** closer to the wanted (driving) waveforms (red and blue) than in the resistor terminated case, and the oscillation pulses do not pass the threshold level (zero) so some receiver hysteresis will usually be enough to reject the pulses. However, to further insure that the bus arbitration always works, the receiver shall include a (digital) filter, which can remove pulses with low duty-cycle and/or pulses narrower than the maximum pulse width (1/6T) as shown above even if the threshold level is crossed for example during noisy conditions. With hysteresis and filter, the bus arbitration is non-destructive without any loss of power, **which may not be the case for most other fieldbus systems, which use bit-wise bus arbitration!**

In case of a single device driving the line towards an unterminated end, the clamp network in any added device will remove the oscillation entirely from that point and towards the other (terminated) end, so there is no oscillation problems in that case.

There are numerous benefits of the clamp termination.

- It is not necessary to remember to connect termination resistors.
- It is easy to connect and disconnect bus systems and this may even be done during operation (hot plugging) if L- gets contact before L+.
- If a reduced speed is acceptable, it is possible to use reflected wave switching to reduce the cable cross section and/or increase the line length.
- It is not necessary to use cables with well specified characteristic impedance.
- The reflected wave charges the line and creates very good signal integrity.
- The signal is very easy to detect.
- The very high transmitter power gives an excellent S/N ratio. With a nominal 13 V peak-to-peak signal and a balanced 4-wire, 50 Ω line, the transmitter power is 3.4 W if the line is driven from one end and 6.8 W peak in any other point. This is 42 or 84 times more than CAN, which typical has a 2.2 V peak-to-peak signal as shown above and a load impedance of 60 Ω in any point, which gives a transmitter power of only 80 mW.
- It is up to 3 times more energy efficient than a resistor termination and the efficiency is increased even more if the line is shorter than the maximum length. In practice however, the difference is lower because the transmitter voltage must be regulated to be able to handle the very big supply voltage range and this creates a bigger voltage drop especially at the higher voltage levels.
- Any number of clamps may be used and more devices close to each other may share the same clamp.
- It gives an excellent input protection against transients.
- It reduces the oscillation, which occurs if more devices are driving the line simultaneously. It is therefore much easier to make a digital filter, which can ensure that the bit-wise bus arbitration gets non-destructive.

There are however also a few drawbacks:

- The principle only works if the pulse width is bigger than or equal to two times the propagation delay of the line = 2T, but this is anyway necessary for the bus arbitration so in practice, it doesn't matter **if a multi-master**

network with free device placement is needed. If the line is longer than that, a (short) pulse with opposite polarity will be generated just after the 2T drive pulses as shown below:

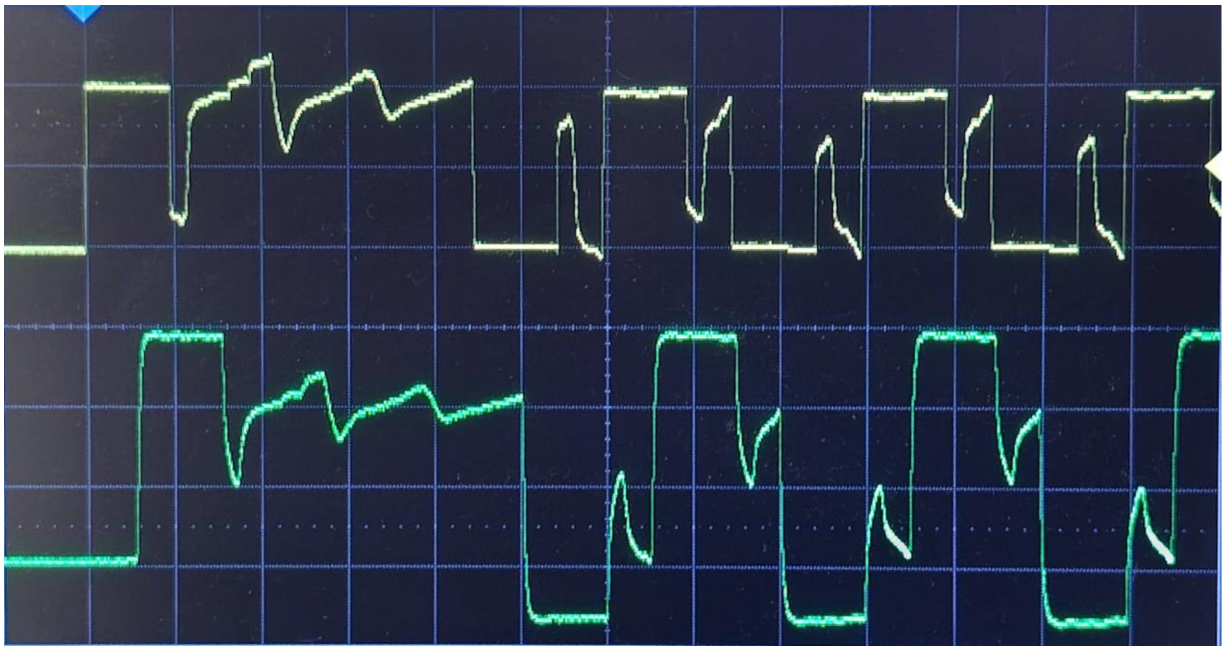


Fig. A.6

The upper yellow curve is the driving waveform in one end of the line and the green lower curve is the waveform in the other end. In the shown example, Max-i still worked fine, and an advanced transmitter may fairly easy detect the exceeded line length as a change to the opposite line polarity **and back again** between 2T (end of drive pulse) and 3T (next bit) and in that case issue an error message. In this way, it can be ensured that the timing margins are always within the limits even with capacitive loads, which reduces the maximum line length. With for example CAN, there is no way to tell that the programmed propagation segment (see chapter “Bit Coding” in layer 1) is too short for the given line length, and most users are not even aware of this segment at all so it is difficult to ensure a good reliability.

- **To avoid that the various bits (pulses) interferes with the following, the line must be terminated with a clamp network in at least one end.**
- **The maximum distance from the other end of the line to a termination network shall not exceed half the maximum line length, so it may be necessary to add extra clamps if there are long parts without devices.** In case of very long ends, which may not be powered up (clamp inactive), it may even be necessary with a relay (driven by the line voltage), which can disconnect the COM conductors.
- A special circuit is required to protect the device against reverse polarity and prevent that a device, which is not powered up, clamps the communication.
- It is necessary with an active circuit, which can hold the line and stabilize differences in clamp voltage.
- If the transmitter and the clamp do not use the same power supply, the returned current through the clamp may pump the supply voltage up on parts with less than 30 mA load current – especially in case of maximum line length and intense communication. However, if this is a problem, it may easily be prevented by means of a bleeder resistor or dual-output (dual fuse) power supplies where each line part is driven from two power supplies – one from each end.

In some situations, it may be desirable to use a faster communication speed than the maximum line length allows. In this case, Max-i may also be used with traditional resistor termination with or without a coupling transformer, but since this reduces the voltage swing for each pulse to the half, the signal/noise ratio is reduced considerably, and due to the problems with the signal integrity between transmitting devices, all master devices must be located very close to each other.

In the following chapters, the theory behind the resistor-less Max-i termination is explained in details.

A.2 Behavior of Transmission Lines

When a signal is applied to a transmission line, it starts traveling along the line. **If the line has a uniform cross section, that is, no helix wound conductors etc.; the propagation velocity depends only on the insulation material!** It may be calculated as:

$$u = 300{,}000 / \sqrt{\varepsilon_r} \quad [\text{km/s}]$$

ε_r is the relative permittivity for the insulation material and $\sqrt{}$ means square root. For air, $\varepsilon_r = 1$, so the propagation velocity is in this case the well-known light speed of approximately 300,000 km/s. For FRPE insulated cables with $\varepsilon_r = 2.6$, the speed is reduced to 186,000 km/s.

In the same way, the time (t) it takes for a signal to propagate a given distance may be calculated as:

$$t = 3.333 \times \sqrt{\varepsilon_r} \quad [\mu\text{s/km}]$$

The relationship between the voltage on the line and the current is called the characteristic impedance (Z_0) of the line. It is a complex number with a resistive and reactive component. It may be calculated as:

$$Z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}}$$

L is the **sum** of the self-inductance of the conductors back and forth and therefore approximately twice as high for a twisted pair line than for an unbalanced line with the same conductor cross section, and C is the total capacitance between the conductors. R is the DC resistance and G is the leakage conductance, which is usually so small that it may be ignored except for very high frequencies where there may be a loss in the insulation material - especially in polarized materials like PVC.

The characteristic impedance changes considerably with frequency, particularly from DC to about 100 KHz. At DC it is:

$$Z_0 = \sqrt{(R / G)} \quad [\Omega]$$

At high frequencies it is:

$$Z_0 = \sqrt{(L / C)} \quad [\Omega]$$

For a 4 × 1.5 mm^2 balanced transmission line, R and $j\omega L$ becomes equal at a frequency of 7.7 kHz. This means that for all practical use of Max-i, the characteristic impedance may be considered as constant and pure resistive. This also means that the following formulas are valid:

$$t = C \times Z_0 = L / Z_0 \quad [\mu\text{S}]$$

$$t = \sqrt{(L \times C)} \quad [\mu\text{S}]$$

$$Z_0 = 3333 \times \sqrt{\varepsilon_r} \times K / C \quad [\Omega]$$

C is the **total** capacitance measured in µF, L is the **total** inductance (sum of the conductors) measured in µH and K is the cable length in km.

A.3 Line Reflections on a Loss less line

When the signal arrives at the far end it may or may not be reflected depending on the connected impedance.

The general solutions for the voltage (v) and current (i) on an **ideal loss less** transmission line is:

$$v = f_1(t - x/u) + f_2(t + x/u)$$

$$i = (1/Z_0)\,[\,f_1(t - x/u) + f_2(t + x/u)\,]$$

where t is the time, x is the position and u is the propagation velocity. The function f_1 is an arbitrary function of the argument t-x/u and represents a wave traveling to the right in the x direction with velocity u. Similarly; f_2 represents a wave traveling to the left. The voltage and current in any place of a transmission line is a sum of the two waves.

With an infinite line length, there is no f_2 function so:

$v = f_1(t - x/u)$

$i = (1/Z_0) f_1(t - x/u)$

This means that v/i is always = Z_0. Because there is no DC connection between the conductors, it is possible to charge the line. Seen from one end, an infinite long transmission line therefore behaves exactly like a resistor R equal to Z_0 in series with an infinite big capacitor. If a pulse is applied to such a line, the line returns to the DC level it had before the pulse when the pulse is removed. If a **finite** line is terminated with a resistor R equal to Z_0, v/i is also equal to Z_0 in the termination point so the line behaves like an infinite line with no reflections except that the capacitance is limited.

Suppose the same waveform f_1 is added simultaneously from **both** end of an **unterminated** line with the length l. At t = l/2u the signals will meet in the middle of the line. Because the current in both waves goes towards the opposite end they will have the opposite direction so the resulting current will be 0. On the other hand, the voltage has the same polarity so the resulting voltage will be twice as high. The net result is a waveform with $v = 2f_1$ and i = 0, which then travels towards each end. Note that this wave actually consist of two waves traveling in opposite directions. A wave with $v = 2f_1$ and i = 0 of course cannot exist since v/i for any wave shall be equal to Z_0.

If the same experiment is carried out, but the transmission line is now cut in the middle, exactly the same thing will happen. At t = l/2u the two waveforms will again arrive in the middle, but because the line is cut here they of course cannot continue into the other half. However the current (i) cannot flow out of an unterminated line so to fulfill the boundary condition and bring the current to 0 it is necessary to add a reflected wave exactly equal to the wave from the other end in the previous example - a waveform equal to the waveform itself, but with the opposite current direction.

The reflected wave is always equal to the wave, which must be added to fulfill the boundary condition.

This is illustrated below where a voltage step with a magnitude U is applied to one end of an un-terminated line.

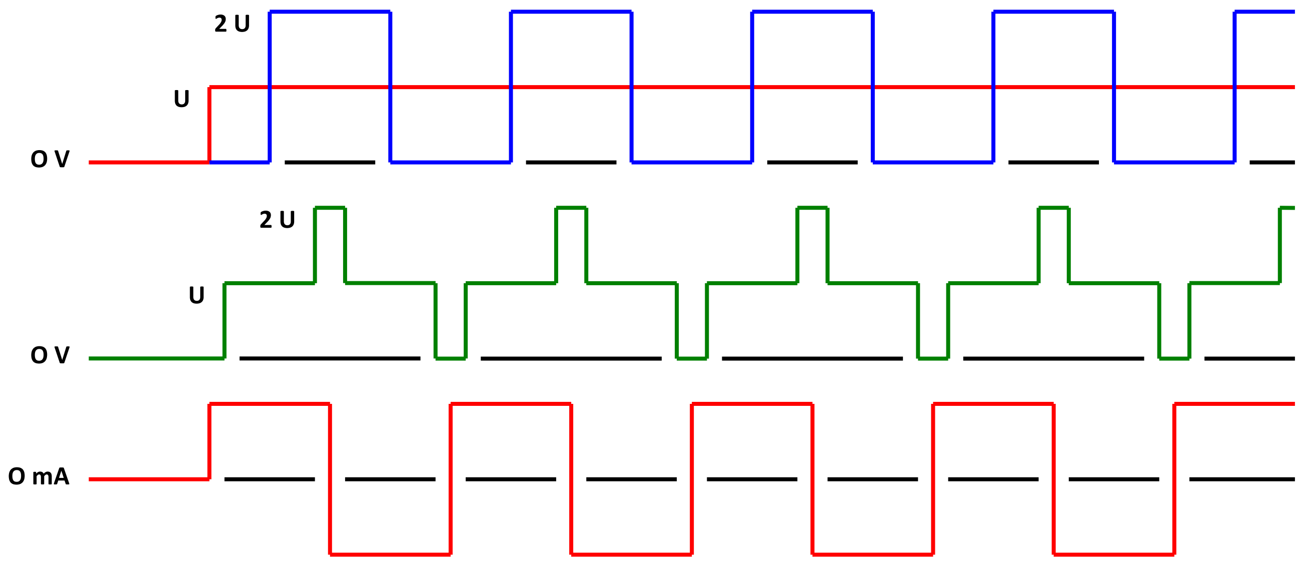


Fig. A.7

The red curves show the step voltage and the step current. The blue curve shows the voltage in the far end of the cable and the green curve shows the voltage on the line in a position close to the transmitter.

When the step voltage reaches the end of the line, it is fully reflected as described before. The reflected wave then add to the original one creating a net result of v = 2U and i = 0 as it travels towards the transmitter. When it reaches the transmitter the voltage is brought down to U. This corresponds to adding a new wave with v = -U so that there are now 3 waves. Since the line impedance is positive this new wave has a negative current like the first reflected wave so when the 3 waves are added the net result is a negative transmitter current with a magnitude corresponding to the original positive one as shown on the third curve. To ease the calculations, the net result of the 3 waves may be regarded as **one** wave, which then travels towards the end of the cable. When this wave reaches the end, it is again reflected. To fulfill the boundary condition and bring the current to 0 we must add a reflected wave with v = -U. This wave has a positive current when seen from the transmitter so the net result is u = 0 and i = 0. This corresponds to the initial condition on the line before the step, so when the reflection reaches the transmitter the process starts all over again**.**

Note that at **any** part of the line the **average** (note average) voltage level is equal to the step voltage.

If the line is instead shorted at the end, it is necessary to add a reflected wave with v = -U to fulfill the boundary condition and bring the voltage down to 0. This wave has the same current direction as the original step, so when the two signals are added the net result is a zero voltage and a current two times the original current. This situation then propagates towards the transmitter where the voltage is again raised to U. This corresponds to adding a third wave with the same magnitude and current direction as the two previous ones, so the current in the transmitter is now 3 times the original current. For each round trip, the current is therefore increases with two times the original current (1, 3, 5, 7...).

It is obvious that a shorted line is a very serious situation - in fact the worst thing, which can happen to a fieldbus. Not only will it prevent all communication by bringing the voltage down to 0, but it will also rapidly - but not immediately - generate a strong transmitter overload.

A.4 Un-terminated line

If the step is replaced with a real communication signal, then a collision between equal pulses from two devices in each end of a maximum length transmission line may look as shown below. The curves are also valid if the line is cut in the middle (line length = ½ maximum cable length) so that the pulses from the other device are replaced with reflections.

A maximum length cable is defined as a cable, which has a propagation delay equal to ½ the pulse width. If for example the pulse width is 12 µS, a maximum length cable is a cable with a propagation delay of 6 µS. For PE insulated cables (ε_r = 2.3), this corresponds to approximately 1.19 Km. Due to the oscillator tolerances and possible rise time, Max-i specifies the maximum length to 1 km at this speed.

The following apply to all the following examples:

- The supply voltage is ±10 V = 20 V – the nominal supply voltage of Max-i.
- The voltage drop in the transmitter including serial diode is assumed to be 0.8 V.
- The clamp voltage is 0.7 V above L+ and 0.7 V below L-. The current in the clamp diodes may be regarded as negative transmitter current.
- The characteristic impedance Z_0 of the line is 50 Ω corresponding to a balanced 4-wire line.

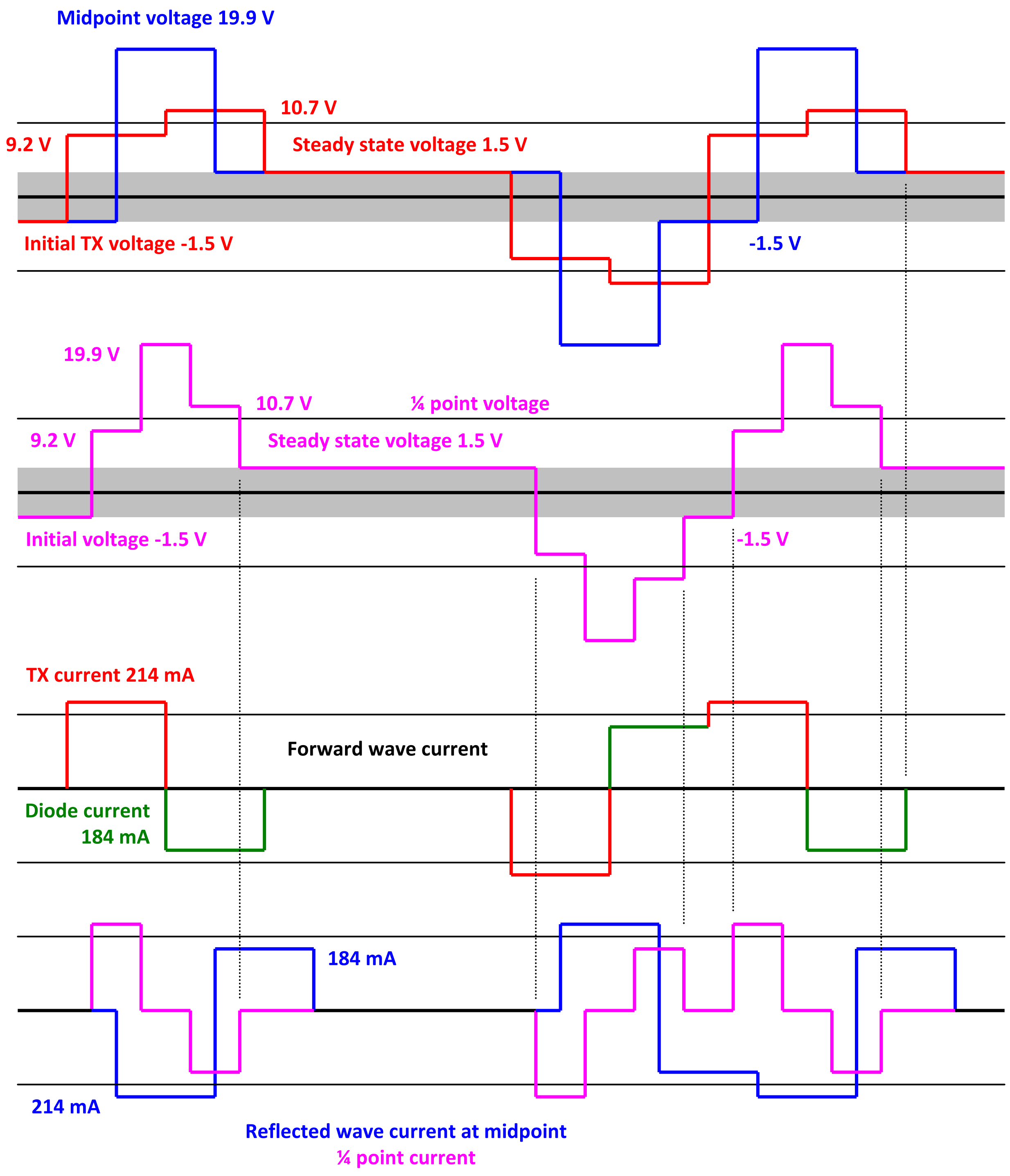


Fig. A.8

The red curves are the voltage and current of the transmitters and the green curve is the transmitter clamp diode current (negative transmitter current). The current in the forward wave is the sum of the red and green curve. The blue curves are the voltage and current in the middle of the line, and the pink curves show the voltage and current at a point in the middle between the transmitter and the midpoint of the line (¼ line point).

When the pulses meet in the middle of the line, the voltage will add and the current will be canceled out. This situation then propagates towards the transmitter(s). Because the line current is canceled out by the wave from the other transmitter or the reflected wave in case of an open line, the current pulse in the line will get shorter and shorter as it approaches the middle of the line. This is shown with the pink curve. The **average** current therefore gets smaller and smaller until it is 0 in the middle (or in the end in case of an open line). Note however that it is the average current, which gets smaller as the pulse gets narrower - **not** the peak current, which is unchanged.

If for example the line was charged to -1.5 V prior to the pulse, then the voltage swing is 9.2 V + 1.5 V = 10.7 V. This corresponds to a current of 214 mA. When this pulse is reflected, the net result is a pulse with a voltage of 2 × 10.7 V - 1.5 V = 19.9 V.

When this wave arrives at the transmitter, the voltage is brought down to maximum 10.7 V by means of the clamp diode. In this way, the majority of current is returned to the power supply rail. The voltage drop is -9.2 V. This is the wave we have to add to fulfill the boundary condition. Note that it has the opposite polarity of the original transmitter pulse.

For calculation purpose only, the net result may be regarded as one wave with v = 10.7 V and a current corresponding to a -9.2 V voltage swing. When this wave arrives in the middle of the line exactly the same thing happens as with the original transmitted pulse except that the current is reversed. To fulfill the boundary condition of zero current we must add a wave in the opposite direction big enough to cancel out the current. Since the current corresponds to a -9.2 V pulse (calculation purpose only), the new wave must also have amplitude of -9.2 V. This will bring the line voltage down to 10.7 V - 9.2 V = 1.5 V. This situation then propagates towards the transmitter.

If the transmitter goes off **exactly** when this wave arrives as in this situation, there will be no further reflections because the boundary condition for an open line is already fulfilled since the current is 0 and the voltage is lower than the clamp voltage. In this case, the line voltage after the pulse will be a constant voltage of 1.5 V. Note that this is equal to the total loss, that is, the 0.8 V voltage drop in the transmitter plus the 0.7 V voltage drop in the clamp diode.

If the transmission line is shorter, the described procedure will happen once more except that the line is now charged to 1.5 V prior to the pulse instead of -1.5 V. The longer the transmitter is active compared to the propagation delay the more the line will be charged and the more the transmitter current will be reduced.

If the transmitter goes off **before** the last reflected pulse has died out (longer transmission line), the transmitter may be overloaded in case of a 0-bit. The worst-case situation is a maximum length transmission line. This situation is shown below:

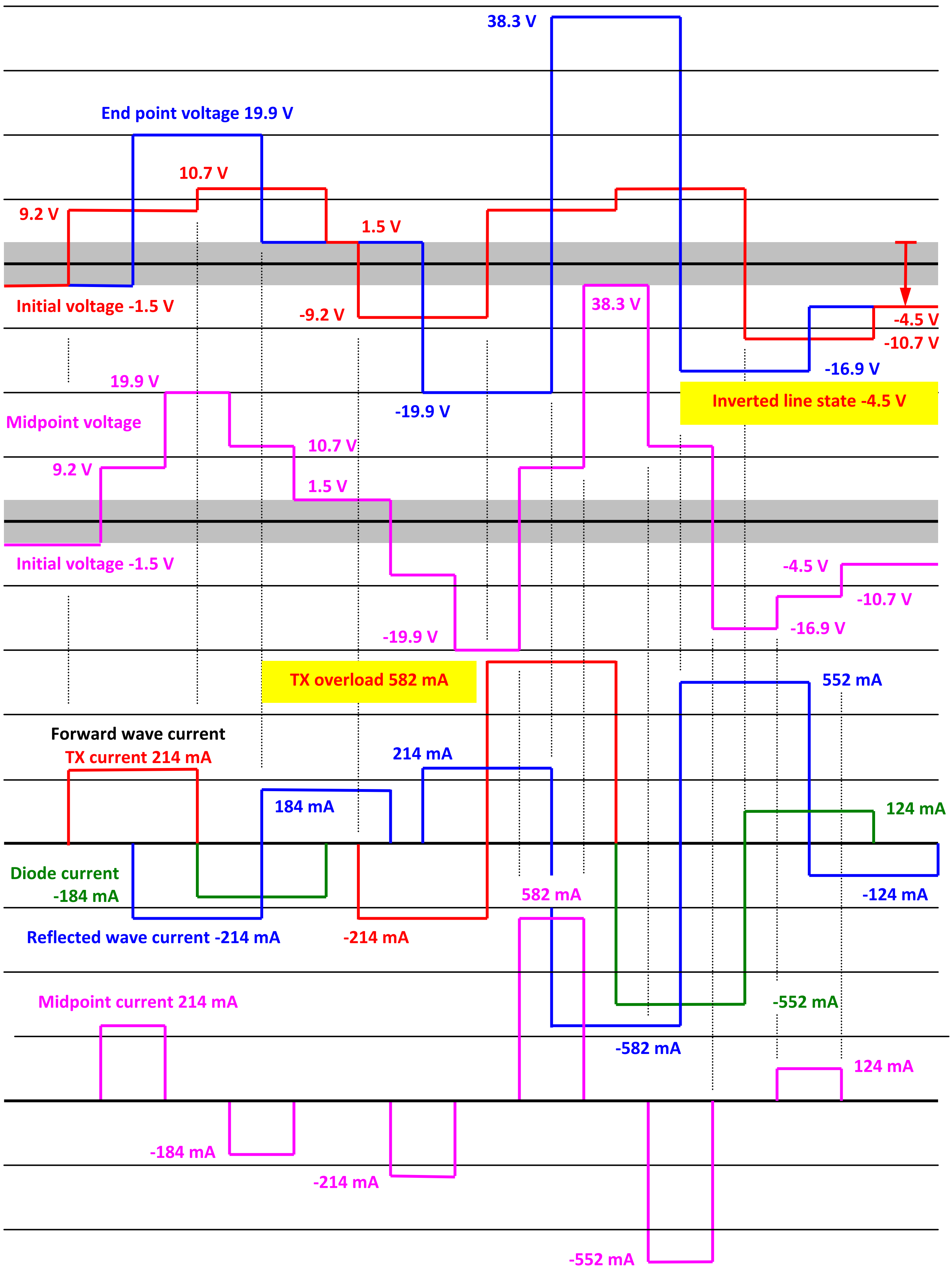

38.3 V
End point voltage 19.9 V
10.7 V
9.2 V
1.5 V
Initial voltage -1.5 V
-9.2 V
38.3 V
-4.5 V
-10.7 V
-16.9 V
19.9 V
Midpoint voltage
-19.9 V
Inverted line state -4.5 V
10.7 V
9.2 V
1.5 V
Initial voltage -1.5 V
-4.5 V
-10.7 V
-19.9 V
-16.9 V
TX overload 582 mA
552 mA
Forward wave current
TX current 214 mA
214 mA
184 mA
124 mA
Diode current
-184 mA
582 mA
-124 mA
Reflected wave current -214 mA
-214 mA
Midpoint current 214 mA
-552 mA
-582 mA
124 mA
-184 mA
-214 mA
-552 mA

Fig. A.9

As can be seen, the first two 1-bits work OK, but a heavy overload is generated when a 0-bit is transmitted. For each 0-bit, the current and voltage keep stepping up until the transmitter goes into current limitation and cuts off. Besides, the line polarity will be inverted in case of an odd number of 0-bits. To avoid this, the line shall be terminated.

A.5 Simple Diode Termination

If the line is terminated with ideal diodes, which clamps the voltage level to the supply voltage levels, and the transmitter is loss less, there will be **no** reflections because the voltage level is constant so that there cannot be any reflected wave. Instead, the full current goes through the diode and is returned to the power supply.

In case of a real transmitter and diode, the majority of current goes through the diode, but a small part will be reflected. This part corresponds to the loss in the system, that is, the difference between the clamp voltage (10.7 V) and the transmitter voltage (9.2 V). This wave has an amplitude of 1.5 V and therefore a current of 30 mA. The wave reduces the current in the line a little, but it is not powerful enough to reverse the transmitter current and turn the clamp diode on so the transmitter still supplies the majority of current and the transmitter voltage is therefore virtually unchanged. The voltage is therefore brought down from 10.7 V to 9.2 V when the wave arrives at the transmitter. To fulfill the boundary condition, we must add a new wave so that there are now three waves in the system:

1) The original wave from the transmitter.
2) The reflected wave with a voltage of 1.5 V and a current of -30 mA.
3) A wave with a voltage of -1.5 V and a current of -30 mA caused by the reduction in line voltage from 10.7 V to 9.2 V.

Note that wave 2 and 3 have the same current polarity so that the transmitter current is reduced with 2 × 30 mA = 60 mA. Because the voltage from the transmitter is unchanged, this situation may be regarded as a completely new situation where the line was just charged to a 3 V higher level before the pulse. If for example the line voltage was -1.5 V prior to the first pulse, the new situation corresponds to a line voltage of 1.5 V prior to the pulse. This keeps going on as long as the transmitter is on. For each round trip, the transmitter current is reduced with 60 mA and the line is charged 3 V. In case of a maximum length transmission line, there is only one round trip, so the line voltage is only charged from -1.5 V to 1.5 V. In case of a ½ length line, there are two round trips. The voltage is therefore charged from -3 V to 3 V.

Note that the line voltage is never in the range from -1.5 V to 1.5 V except of course for transitions. In the drawings, the dead band is shown as gray.

The following three drawing shows:

1) ½ length line terminated at the end, or a collision between two devices on a maximum length line with diode termination in the middle.
2) Maximum length line diode terminated at the end.
3) Maximum length line terminated in the middle with diodes, but not at the end.

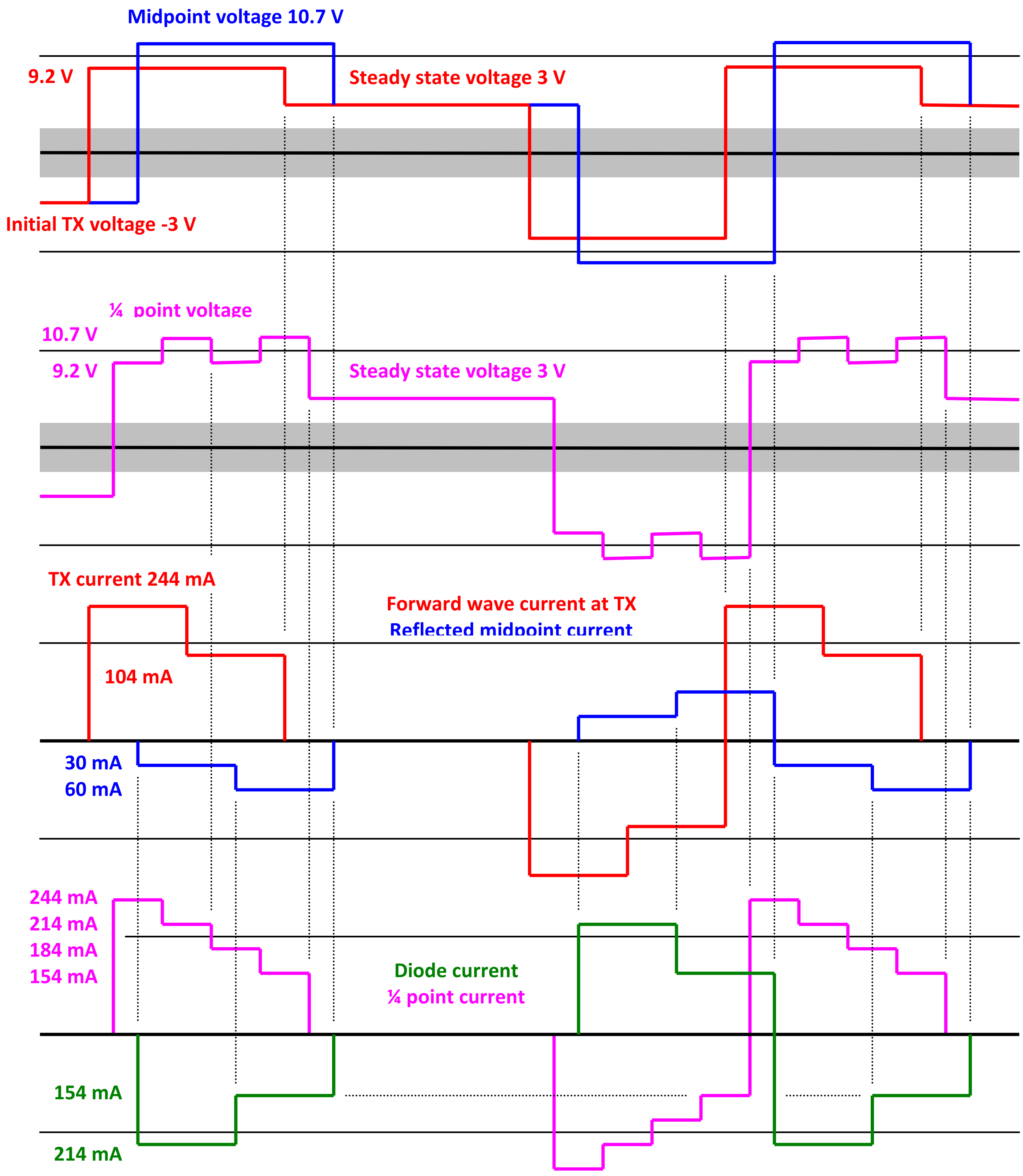

Midpoint voltage 10.7 V
9.2 V
Steady state voltage 3 V
Initial TX voltage -3 V
¼ point voltage
10.7 V
9.2 V
Steady state voltage 3 V
TX current 244 mA
Forward wave current at TX
Reflected midpoint current
104 mA
30 mA
60 mA
244 mA
214 mA
184 mA
154 mA
Diode current
¼ point current
154 mA
214 mA

Fig. A.10

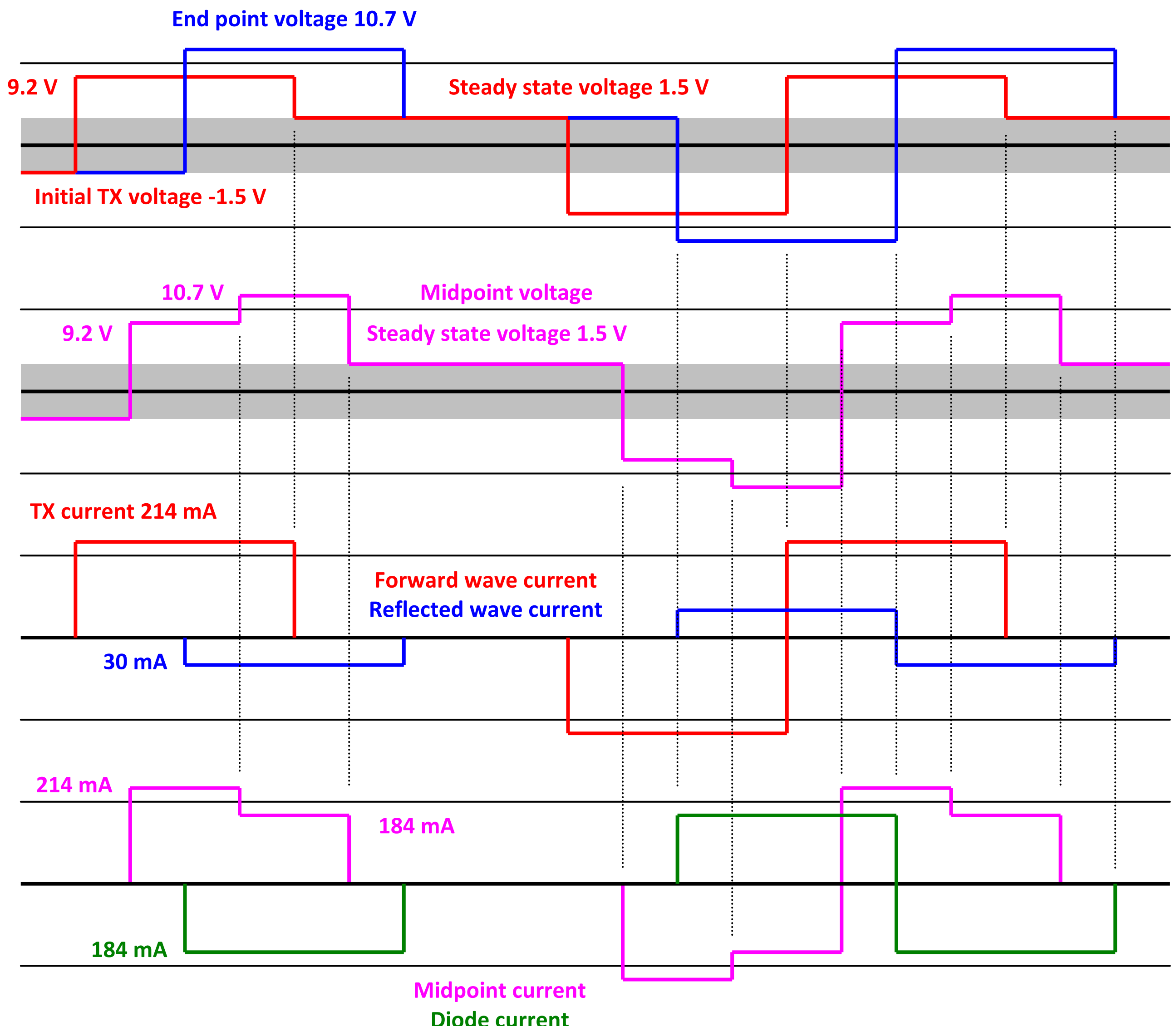

End point voltage 10.7 V
9.2 V
Steady state voltage 1.5 V
Initial TX voltage -1.5 V
10.7 V
Midpoint voltage
9.2 V
Steady state voltage 1.5 V
TX current 214 mA
Forward wave current
Reflected wave current
30 mA
214 mA
184 mA
184 mA
Midpoint current
Diode current

Fig. A.11

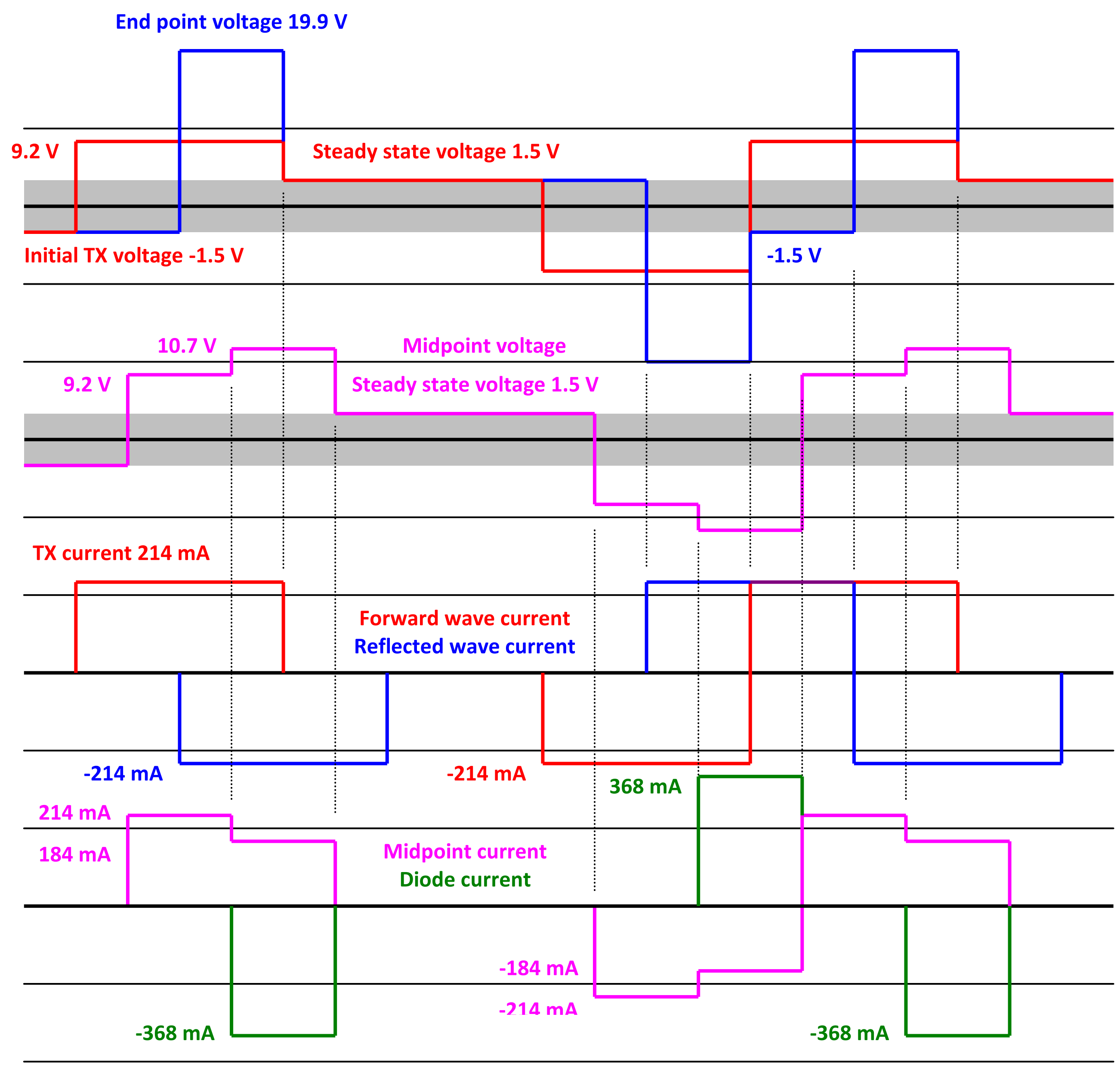


Fig. A.12

As can be seen, **no** overload or state reversal takes place if the length of an un-terminated cable end is less than ½ the maximum cable length, but the current in the diode termination is doubled. If a maximum length transmission line with only two terminations in each end is cut exactly in the middle, the communication works as shown. However, if the cut is not done exactly in the middle, one end is overloaded in case of 0-bits. To avoid this, the distance from a transmitter to an un-terminated end must not exceed ½ the maximum line length or diode networks must be distributed along the line with a maximum mutual distance of approximately 10% of the maximum line length.

An interesting thing that may be seen from the curves is that at a diode clamp, the voltage waveform above or below the dead band is almost independent of the termination of the rest of the line! If all devices include a diode clamp, they will therefore see a correct waveform even if the receive threshold/trigger levels are above and below the dead band. Within the dead band, the waveform is always correct.

The diode termination has many important benefits compared to a resistor termination:

- The signal/noise ratio is improved. With a resistor termination the signal always goes back to 0 when the transmitter stops, but with a diode termination the transmission line is charged depending on the loss (dead band). The bigger the loss, the better signal/noise ratio!!!

- The power loss is much lower because the transmitter power is not dissipated as heat in a termination resistor, but most of the current is returned to the supply rail through the clamp diodes. The **initial** power loss in the transmitter is a little higher due to the charged transmission line, which causes a bigger voltage swing and thereby a higher current level, but when the reflections arrives, the current will be reduced as shown.
- Any number of termination diodes may be added. This makes the bus very failure tolerant.
- A diode termination does not need to match the characteristic impedance of the cable so it is not necessary to know this impedance and use well-defined cables.

However, the diode termination has a very important disadvantage – it is very difficult to set the ideal bias level prior to the telegram. On fig. A.9 it is ±3V, but on fig. A.10 and A.11 it is ±1.5 V. This means that the ideal bias level depends on the line length. If the bias level is not optimized, the dead band will be displaced and not be symmetrical around 0. If for example the bias level on fig. A.10 and A.11 was -6 V instead of -1.5 V prior to the first pulse, the dead band will not be ±1.5 V, but start at -6 V to -3 V and slowly drift towards ±1.5 V. If the triggering levels (hysteresis) for example are ±1 – 2 V, the line will be in low state when the line is floating after the first bits and a 1-bit may be converted to a 0-bit as shown below:

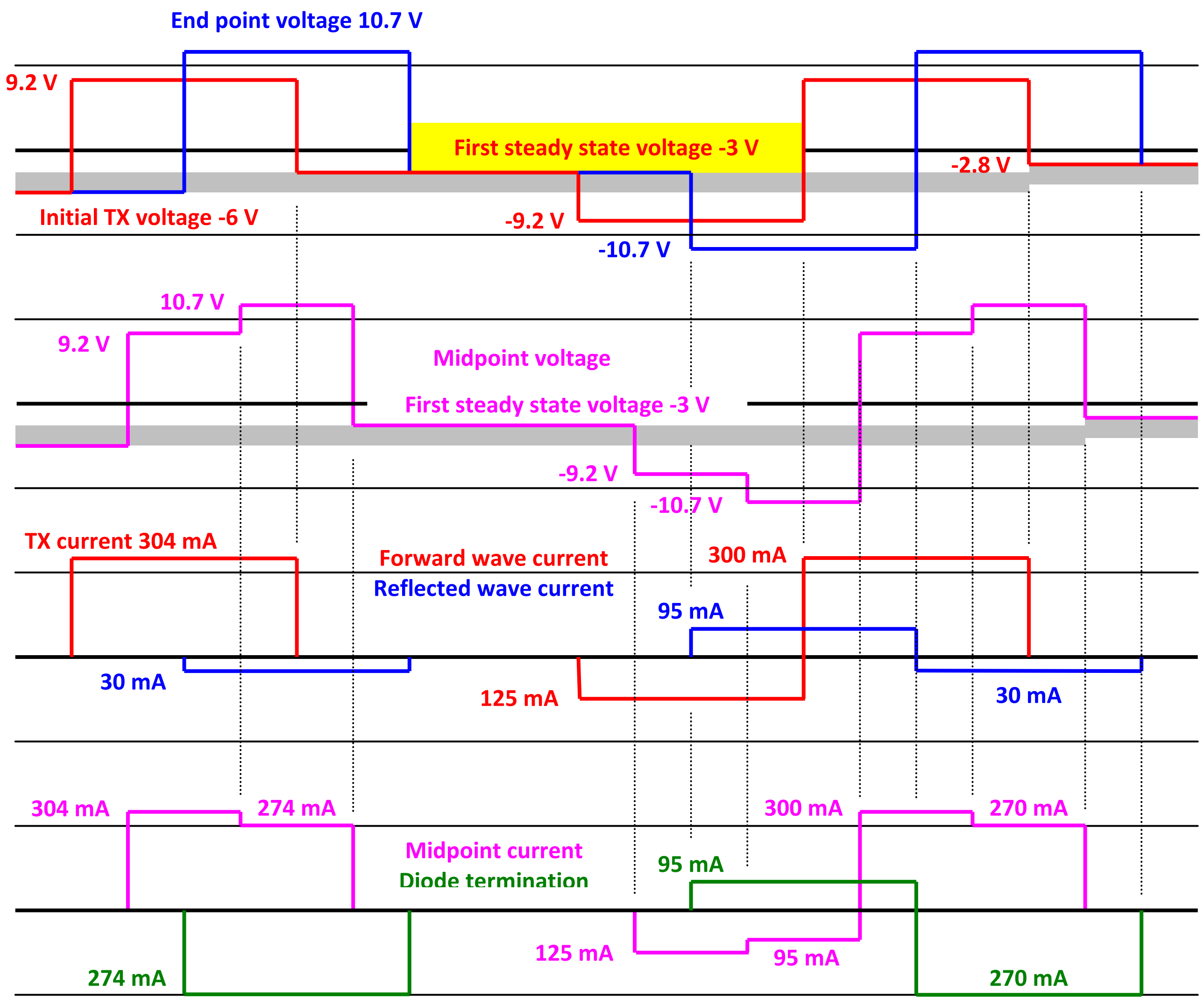


Fig. A.13

A.6 Multi Diode Termination

It is obvious that if the dead band could be increased to approximately the total supply voltage, not only will the S/N be improved considerably, but there will also be no problems with DC bias as this can just be close to either L+ or L-. Because loss in the system increases the size of the dead band, it is natural to increase the loss in the clamp by means of more diodes and/or diode-coupled transistors in series (diode coupled transistors with basis and collector connected have a sharper knee due to gain and therefore lower resistance). Because the line voltage may then exceed L+ or L-, it is necessary that one of the diodes be in series with the transmitter so that the voltage drop over the transmitter including serial diode becomes 1.5 V. If the number of diodes or diode-coupled transistors is selected in such a way that the total loss in the system including transmitter and average DC resistance is ½ the supply voltage, that is, 10 V at 20 V supply voltage, the situation for a maximum length, multi diode terminated, lossless transmission line with an 8.5 V clamp looks like this:

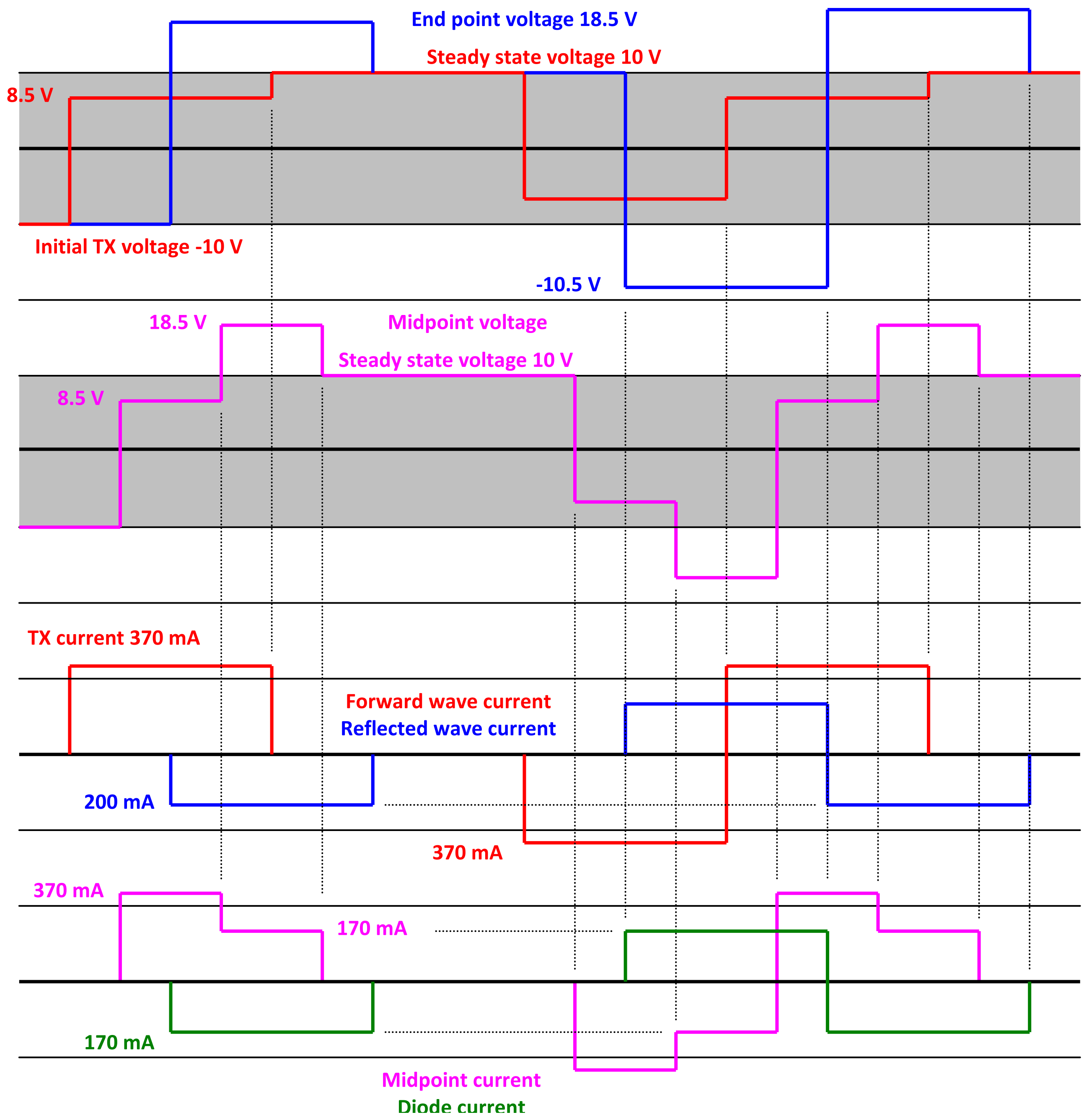


Fig. A.14

In this case, the reflected wave has an amplitude of 1.5 V + 8.5 V = 10 V. The current in the wave is therefore 200 mA so that the current through the diode clamp is only 370 mA - 200 mA = 170 mA. There are now two waves in the system:

1. The original wave from the transmitter with an amplitude of 18.5 V and a current of 370 mA.
2. The reflected wave with a voltage of 10 V and a current of -200 mA.

If the transmitter goes off exactly when the reflected wave arrives, which will happen automatically if the voltage is higher than the transmitter voltage, the current out of the line is zero. To fulfill the boundary condition, we must therefore add a third wave:

3. A wave with a current of -(370 mA -200 mA) = -170 mA and therefore an amplitude of -8.5 V.

This will bring the voltage level down to 18.5 V - 8.5 V = 10 V.

If the line length is only ½ the maximum length or if a collision takes place between two devices on a maximum length line with diode termination in the middle (as fig. A.9), exactly the same thing happens except for a "compressed" time scale. Because of the transmitter serial diode, the transmitter current is reduced to zero when the line reaches the 10 V steady state condition half way into the transmitter pulse as shown below. Unlike fig. A.9, the ideal bias level therefore does not depend on the line length.

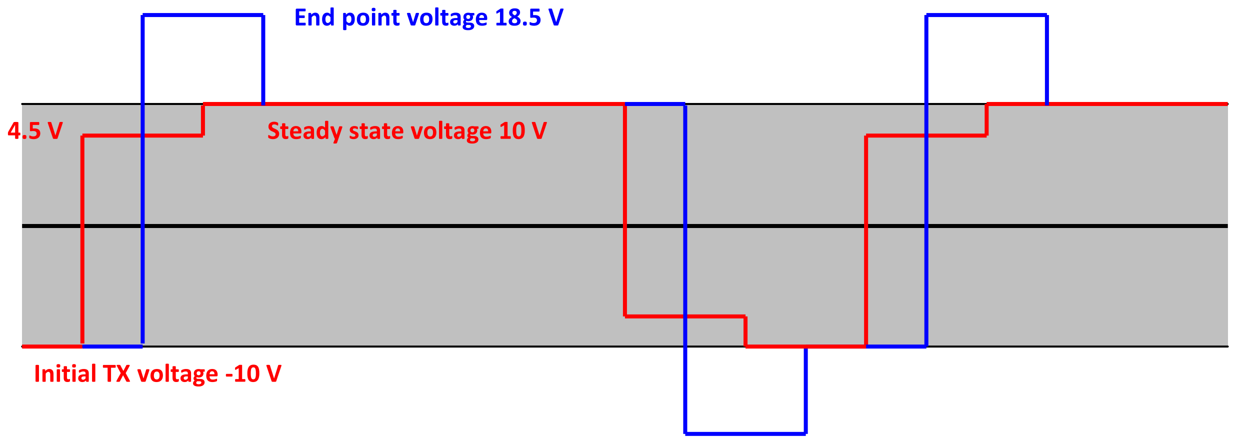


Fig. A.15

If the supply voltage is increased to 23 V (±11.5 V), but the voltage drop in the transmitter and the clamp is not changed, the first transmitter pulse will get a voltage swing of 21.5 V and a current of 430 mA. The first reflected wave is still 10 V as it is equal to the voltage increase, that is, the difference between the transmitter voltage (10 V) and the clamp voltage (20 V) and it therefore has a current of 200 mA.

If the transmitter is **off**, when the reflection arrives, a third wave with a current of -(430 mA - 200 mA) = -230 mA and therefore an amplitude of -11.5 V must be added to fulfill the boundary condition. This will bring the voltage down to 20 V - 11.5 V = 8.5 V as shown on the first curve below:

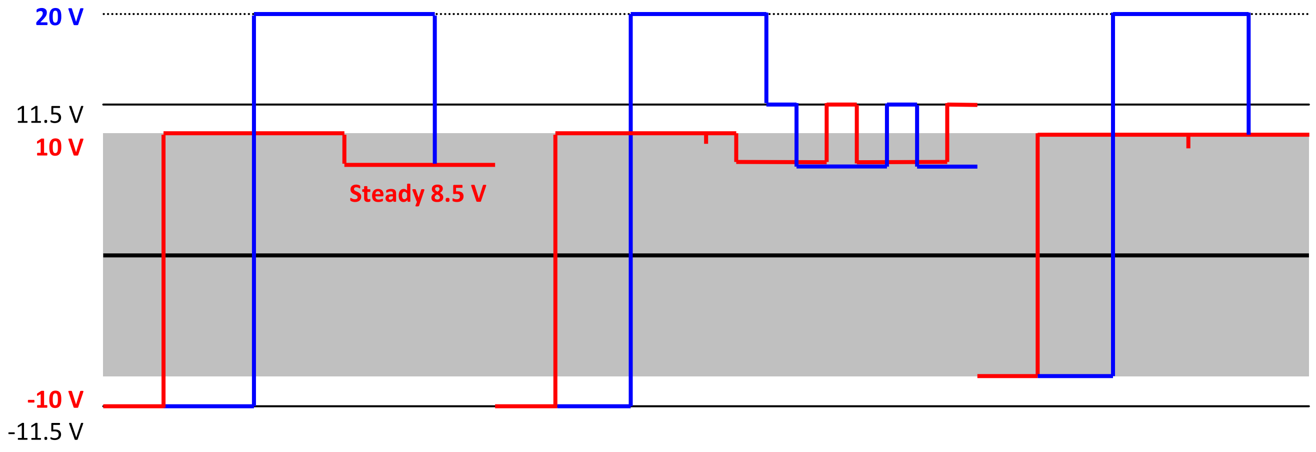


Fig. A.16

However, if the transmitter is still **on**, which will be the case if the transmission line is shorter than the maximum length, the voltage cannot fall below 10 V as long as the transmitter is active. This will add a new, short wave with an amplitude of 1.5 V into the system, which will cause the voltage to change between up to three different levels – 8.5 V, 10 V and

11.5 V as shown on the second curve. It is obvious that this is a much undesired situation. However, if the initial value is changed from -11.5 V to -10 V, the current in the initial wave is only 400 mA, so the third wave will get a current of -(400 mA - 200 mA) = -200 mA, which is equal to a voltage of 10 V. The steady state voltage will therefore be 20 V - 10 V = 10 V as shown on the third curve. This is an ideal situation with no further waves! In this case, the initial voltage is the same as at a supply voltage of 20 V, so if the initial voltage, the voltage drop in the transmitter and the clamp voltage are kept constant, the circuit will be self-adjusting from 20 V to 23 V, that is, over a range equal to two times the voltage drop in the transmitter. This also has the advantage that the initial voltage on the bus is independent of the supply voltage, which is important in case of voltage drops on the bus due to high (DC) currents.

The figure below shows the relations between the various voltage levels shown on the previous figures:

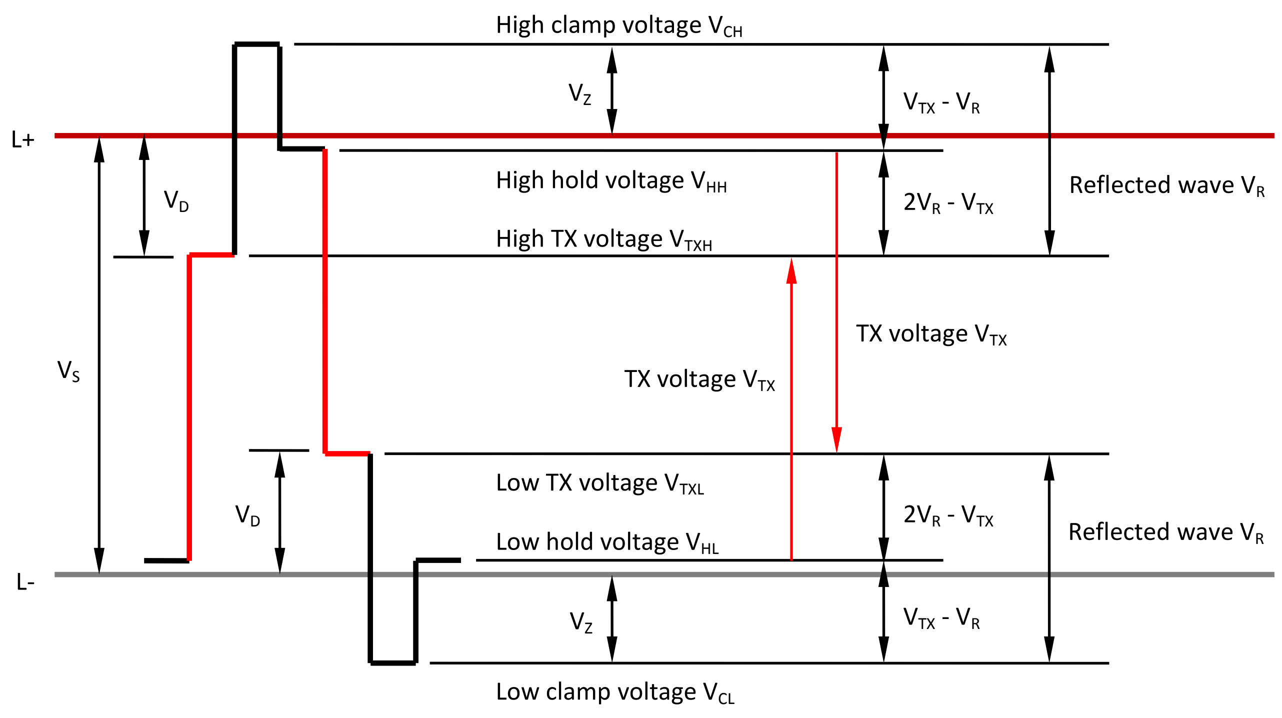


Fig. A.17

The waveform on the left is the waveform in the middle of a maximum length trunk line. The periods where the transmitter drives the line are shown in red. Note that the transmitter turns off when the reflections arrive and that the reflected wave is equal to the maximum voltage increase, that is, the difference between the clamp voltage and the transmitter voltage.

If more devices shall be able to operate on the same line even in case of voltage drops, the two hold levels V_{HH} and V_{HL} (and the difference between them) must be the same. Since the difference is $V_{TX} + (2V_R - V_{TX}) = 2\ V_R$, the only ways to achieve this is to keep the difference between the transmitter levels and the clamp levels constant. If the supply voltage is fairly constant (< ±5 %) and the voltage drops in L+ and L- are the same and limited to less the approximately ±10 % in total, a simple Max-i interface may be made by means of a saturated transmitter, fixed clamps like multi diode or zener diode clamps directly to the supply rails, and hold levels, which are constant and independent of the supply voltage. This solution was used up to version 9.0 of the Max-i specification, but it has some serious drawbacks, which makes it necessary with a regulated transmitter and clamp in case of battery operation, where the voltage variations may be up to ±23 %.

- The automatic compensation can only handle a voltage range corresponding to two times the voltage drop in the transmitter ($2V_D$), so to be able to handle the big voltage range for battery operation, the voltage drop needs to be at least 4.6 V, but in practice even more since it is usually necessary to add some headroom for the hold circuit (typical 0.7 V to L+ and L-) as illustrated on the drawing above (hold levels slightly below L+ and above L-).
- To ensure that the transmitter **always** goes off when the reflections arrive, it is necessary to use a much higher hold level than ideal to compensate for voltage drops on the line and temperature variations. This makes the difference between the transmitter levels and the hold levels even bigger, which may lead to fairly bad signal integrity – especially at low supply voltage levels. The figures below shows the (simulated) difference between a

well matched system when driving a lossless, maximum-length line at a point 31.7% from one end and the same system with a 3 V higher clamp level:

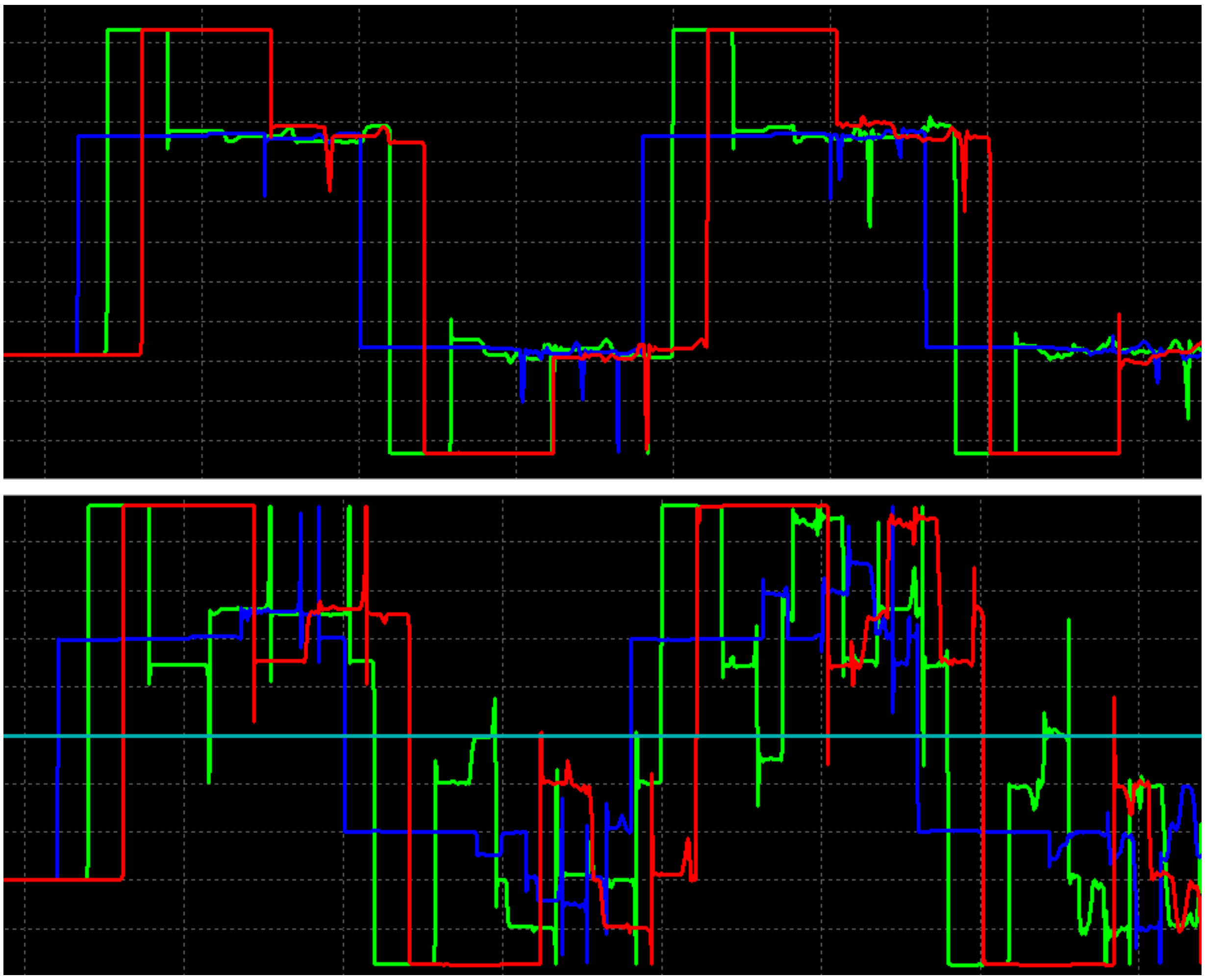


Fig. A.18

The blue wave is the driving signal and the green and red ones are the signals at the two ends. The short spikes are caused by reverse recovery and capacitance in the diodes.

As can be seen, the reflections create pulses, which will be summed together for a long time, if they do not have the ideal amplitude. The first bit will usually be all right, but the resulting wave for the following bits may even pass the zero level, which makes the signal difficult to decode correctly.

This is especially a problem with low-loss lines in high-power networks in for example the house of the future. Any resistance will damp the signal and therefore limit the accumulated pulse energy, so a cable with minimum cross section and therefore maximum resistance handles imperfect matching and asymmetry **much** better than a low-loss cable. A multi diode clamp was therefore acceptable in the early 12 – 14 V versions of Max-i, which were primary intended for long fieldbus cables, but it is no longer sufficient after high-power devices like LED lighting and laptop chargers have been added (and Max-i therefore also changed to 20 V). The figure below shows the effect of maximum cable resistance (minimum cross section):

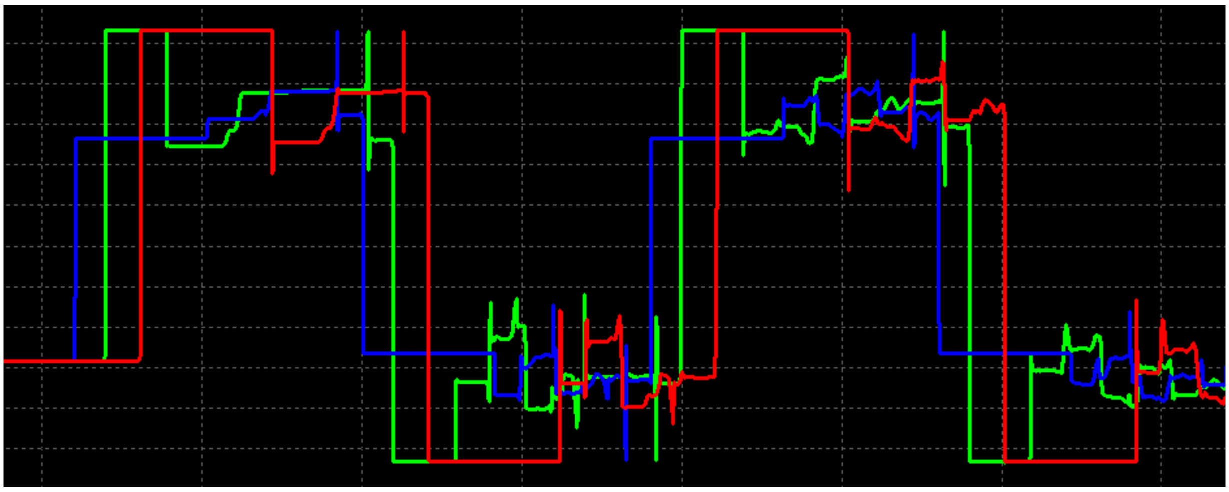


Fig. A.19

As can be seen, the cable loss corresponds to a higher clamp level, but due to the heavy damping, the green and red waves hardly get below the drive level (blue), so the signal integrity and noise margin is almost good as in the well matched lossless case.

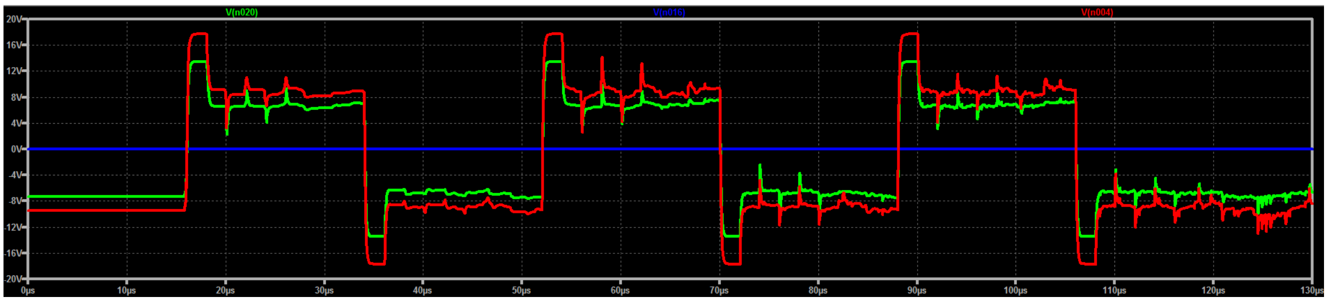
- If multi diode clamps are used, the temperature coefficient of V_R is **very** high as it is approximately -2 mV/°C for each diode (or diode-connected transistor) in the transmitter and the clamp. If for example 8 diodes are needed in the clamp and one in the transmitter, the temperature coefficient becomes -18 mV/°C, which creates a voltage variation of 3.2 V over a −55°C to 125°C range. This is not acceptable as shown on figure A.17 so the temperature range must be limited considerably – in practice to a fairly limited range around 20°C.

It is obvious that the best signal integrity is obtained if $2V_R$ - V_{TX} is always equal to 0 as the hold level then becomes equal to the transmitter level, which creates a "clean" square wave except for more or less "top-hat" as illustrated in the well matched case on figure A.17. This requires voltage regulation of both the transmitter levels and the clamp levels, which makes the circuit much more complicated – especially because the clamp must be accurate and very fast acting without additional line load or high capacitance even though the clamp levels are above L+ and below L-, but once the problems have been solved, Max-i can benefit from the best possible signal integrity and safety margin, so this solution is selected in later versions.

Since it is **very** important that the majority of the clamp current is returned to the nearest supply voltage rail as it is the case with a diode clamp, the clamp levels must always be above L+ and below L-. The necessary headroom is typical at least 0.7 V for a bipolar clamp transistor and a schottky diode, so with a maximum supply voltage of 25.2 V, the clamp levels must be ±13.3 V. In practice, there will however always be some communication resistance on the line due to the DC resistance and the skin effect (described later), so to partly compensate for this, it may be desirable to reduce the clamp levels slightly to for example ±13 V even though the clamp may go slightly into saturation at the highest supply voltage, which rarely occurs. The clamps must anyway be able to handle saturation to be able to withstand the absolute maximum supply voltage range.

The transmitter levels shall be approximately ±13.3 V / 2 = ±6.65 V (± 6.5 – 6.8 V). Since the minimum voltage is 15.4 V, the voltage drop in the transmitter (V_D) should not exceed 1.0 V including serial (schottky) diode, but if the transmitter goes into saturation at very low supply voltage levels, it just corresponds to a slightly higher clamp level, so even if the voltage drop is a little higher, it can easily be accepted.

In the special railway mode, the transmitter is saturated, which increases the transmitter power and reduces the loss in the transmitter, but if there is any noticeable voltage drops on the line, which is not the case for railway applications, the created ringing reduces the signal integrity in the beginning of the pulse, and just 10 % is enough to reduce it to the same level as with a regulated transmitter as shown below:



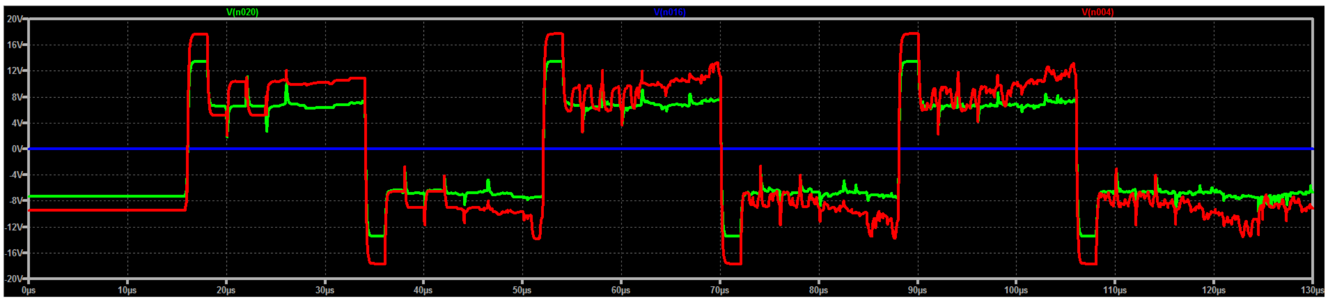


Fig. A.20

The green curves are with a standard regulated transmitter and the red curves are with a saturated transmitter at 20 V. The upper figure shows the ideal case with no voltage drops as in railway mode. In this case it is an advantage with a saturated transmitter as it creates more transmitter power, lower transmitter loss and a better signal/noise ratio. The lower figure shows the same situation with a 10 % voltage drop (±1 V), and in this case it is obvious that the signal integrity in the beginning of the pulses until they are accepted at 1T is not better than with a regulated transmitter. It just uses more transmitter power without gaining anything except for a lower transmitter loss, but simultaneously a much higher clamp loss. Therefore, a regulated transmitter is used as standard.

A.7 Drive Impedance and Impedance Variations

If more transmission lines are connected together in a star topology, the impedance will be very low in the connection point. The impedance may be so low that the transmitter is not able to insure incident wave switching, that is, to bring the signal level above the receiver threshold level immediately. First when the reflections arrive, the threshold level will be passed. This is called reflected wave switching. The impedance may even be so low that the transmitter is not able to bring its own voltage level above the threshold level within a given time. In this case, it is regarded as a short circuit and the transmitter stops.

Any difference in cable impedance will cause a frequency splitting. If the impedance on some part is higher than the rest of the line, it may be regarded as a series inductor. If the impedance is lower, it may be regarded as a parallel capacitor. In both cases, the low frequency components are passed on, but the high frequency components are reflected. This may cause pulse distortion and inter-symbol distortion. Heavy capacitive loads works as a short circuit to high frequencies and will therefore cause a reflected pulse with the **opposite** polarity of the drive pulse. The amplitude may be so high that the receiver threshold is passed. To avoid this, it is recommended to have some cable between many devices. This will spread out the capacitance and cause the load to appear as a piece of cable with lower impedance.

The time constant (t) caused by varying cable impedance may be calculated as:

$$t = (l \times T) / 2 \times ABS(Z_0/R - R/Z_0)$$

where l = cable length, T = propagation delay, Z_0 = cable impedance and R = termination resistor. ABS means the absolute value, that is, the value without sign. For small differences between Z_0 and R the formula may be approximated as:

$$t = (l \times T \times dZ) / 100$$

where dZ is the +/- deviation in cable impedance measured in %. If for example an increased detection delay of 10% is accepted due to low pass characteristic, then the cable impedance may vary ±10%.

A.8 Lossy Transmission Line

In practice, there will always be some resistance, which causes the signals to be attenuated. Quantitatively, attenuation is the real part of the complex propagation constant:

$$\gamma = \sqrt{(R + j\omega L)(G + j\omega C)}$$

At high frequencies where R << ωL, the attenuation factor may be approximated as:

$$A = e^{\frac{-\sqrt{LC}}{2}\left(\frac{R}{L} + \frac{G}{C}\right)} = e^{-0.5\left(\frac{R}{Z_0} + G Z_0\right)}$$

For PE and other polyolefin-insulated cables, G may be set to 0 unless there are several hundred devices connected to the bus. If for example R = 22 Ω and Z_0 = 50 Ω, then A = 0.80.

Due to the so-called skin effect, the serial resistance R is not constant, but depends on the frequency. At DC, the current utilizes the whole cross-section of the conductor, but as the frequency increases, the current tends to flow mainly near the surface of the conductor, which increases the effective resistance. The skin effect is caused by the internal self-inductance of the conductor, which causes an increase in the inductive reactance at high frequencies, thus forcing the carriers, i.e., electrons, toward the surface of the conductor. At high frequencies, the circumference is the preferred criterion for predicting resistance rather than the cross-sectional area. The depth of penetration of current falls gradually and can be very small compared to the diameter of the conductor, but an equivalent penetration depth/limit, which gives a good approximation of the resistance, may be calculated as:

$$\delta = \sqrt{(\rho / (\pi \times f \times \mu))}$$

where δ is the depth in m, ρ is the resistivity in Ωm, f is the frequency in Hz and μ is the absolute permeability in H/m, which is equal to the relative permeability μ_r multiplied with $4\pi \times 10^{-7}$ H/m.

For copper, which has a resistivity of 1.678×10^{-8} Ωm and a relative permeability close to 1, the skin depth may be calculated as:

$$\text{Depth [mm]} = 65.2 / \sqrt{\text{Frequency [Hz]}}$$

This is also shown in the table below:

Effective skin depth for copper		
Frequency	**Depth**	**Cross-section for r = depth**
1 kHz	2.06 mm	13.3 mm^2
10 kHz	0.652 mm	1.33 mm^2
100 kHz	0.206 mm	0.133 mm^2
1 MHz	0.0652 mm	0.01133 mm^2
10 MHz	0.0206 mm	0.00133 mm^2

Fig. A.21

The last column shown the cable cross-section for a cable where the radius is equal to the skin depth, that is, the entire cable cross-section may be regarded as utilized. Note that the cross-section is inversed proportional to the frequency. This means that the skin effect is independent on the transmission speed! If the speed for example is reduced to the half, the transmission line may be twice as long, but in that case; it shall simultaneously have the double cross-section to keep the DC resistance constant.

The frequency at which the resistance is increased 10% may be calculated as:

$$f\,[\text{Hz}] = (200 / D)^2$$

Where D is the diameter in mm of the conductor. This formula fits very well with the above table. If for example D = 2 mm, f becomes 10 kHz. At that frequency, the skin depth is 0.652 mm. The relationship between the AC resistance and the DC resistance is therefore $(1^2 - (1 - 0.652)^2) / 1^2 = 0.88$. This is very close to 1 - 10% = 0.9.

In case of magnetic materials like iron and nickel, which have a very high permeability, the skin depth can be extremely small. At just 1 MHz, the skin depth is only 5.4 µm for nickel ($\rho = 6.84 \times 10^{-8}$ Ωm and $\mu_r = 600$), which corresponds to the nickel layer of an ENIG PCB surface, so in practice, only the back side of such traces draw current except for very low frequencies.

The self-inductance of a conductor consists of a self-inductance L_{ext} due to the **external** field between the conductors plus a self-inductance L_{int} due to the **internal** field in the conductor. At low frequencies where the entire cross section is utilized, L_{int} is 50 nH/m for non-magnetic materials ($\mu_r = 1$), but as the resistance is increased at high frequencies, L_{int} is decreased correspondingly so that $L_{int} \times R_{skin}$ is constant. As the delay is √LC, the high frequency components will have a slightly less delay than the low frequency components. The result is that although the rise time of the signal is increased as the signal propagates down the line and is attenuated; the signal always passes 0 V at a time corresponding to the distance and the propagation delay of the line. Therefore, the increased rise time does not cause a detection delay if the detection level is always 0 V (midpoint between L+ and L-). This is the reason why Max-i uses 3-level hysteresis.

The rise time is proportional to the DC resistance and proportional to the square of the line length! If for example the conductor cross-section is doubled, the rise time falls to the half. If the line length is then doubled, the rise time is increased four times so that the net result is a two times increase in rise time. As the maximum communication speed has to be reduced to the half when the line length is doubled, the skin effect causes the same relative waveform distortion for all speed settings, which can also be seen from fig. A.19 above.

Unlike a low-pass filter where the step response is exponential (shown blue), the skin effect causes a clear knee point where the rise time changes from very fast to very slow. At the beginning of the step, the slope is almost identical to the drive signal - only limited by the slightly reduced delay of the high frequency components as explained above, but towards the end, the step is turned into a long, slow-moving tail, which reaches the final level much slower than an exponential function. In case of short pulses, the signal levels may never reach the final level, so the voltage swing on the next pulse is reduced coursing lower levels and with that even lower voltage swing on the next pulse and so on as shown below:

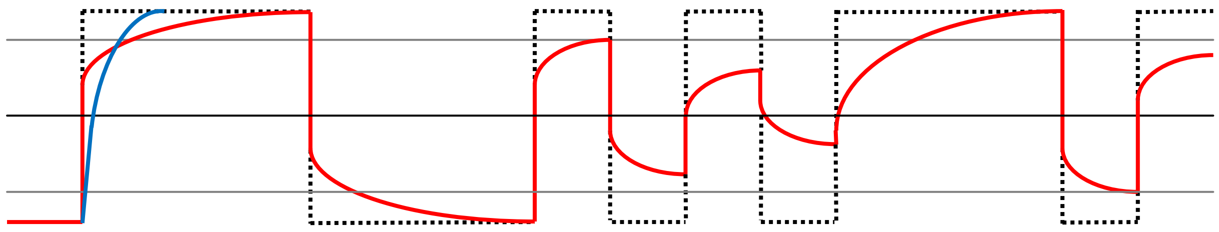


Fig. A.22

In practice, where there is a reflected wave, the signal may look like this:

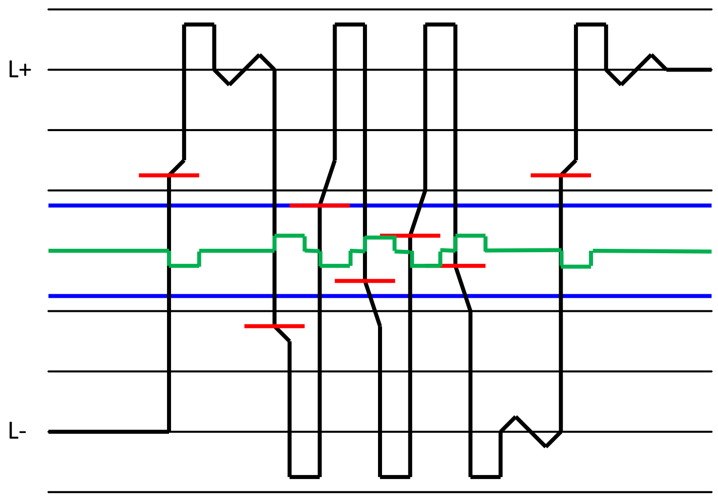


Fig. A.23

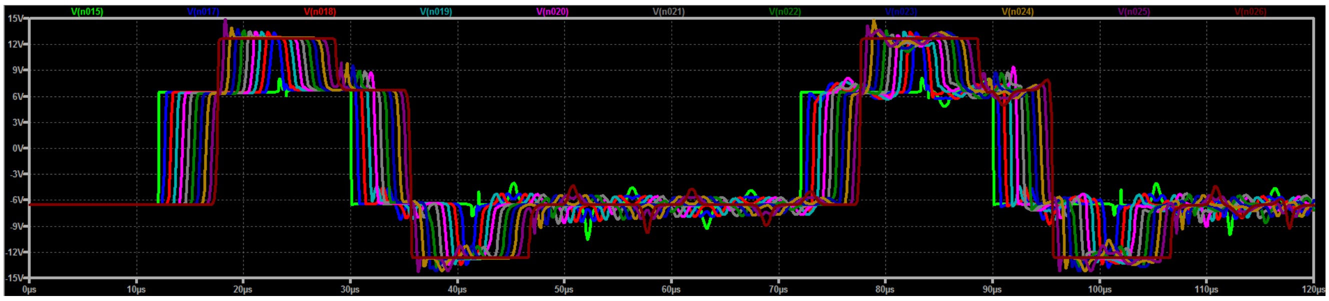
The green line shows the 3-level hysteresis of Max-i and the short red lines show the falling skin effect knees on subsequent 0-bits **without pause** (or on NRZ signals). Even if the skin effect knees get very close to the zero level, the detection is not affected, but it of course reduces the signal level on especially thin lines and with that the S/N so a pause with a length of 1T has been included in the 0-bit to reduce this phenomenon to an acceptable level. This pause also prevents that more transmitters can counteract each other even on lines, which utilize reflected wave switching. If the triggering levels were instead two fixed levels as shown with the blue lines, which is the case for many other fieldbus systems, it is obvious than when the skin effect knees get below the triggering levels, the safety margin would be reduced and the detection delay (D) may exceed the maximum limit. This may be a problem on resistor terminated fieldbus systems although the waveform is different.

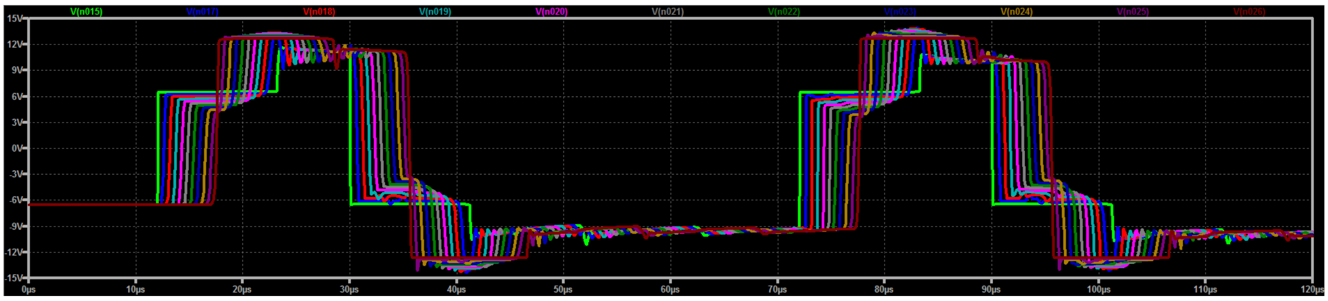
A.9 4-wire Balanced Transmission Line

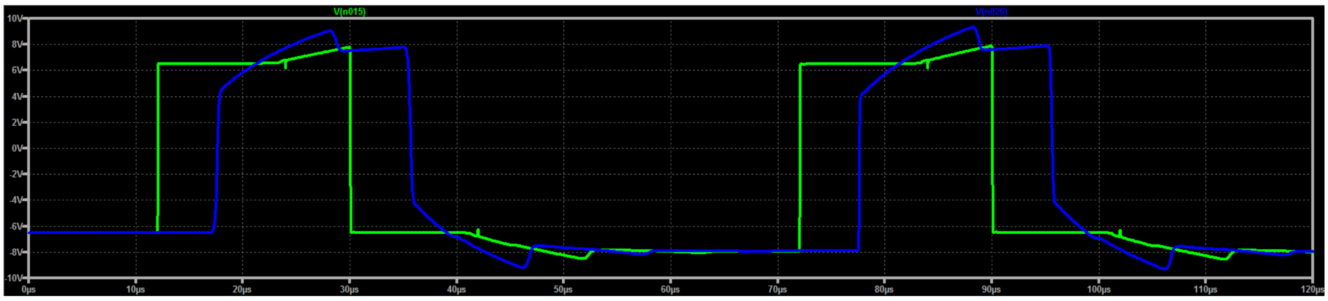
The characteristic impedance of a 4-wire line depends on how it is driven. When the cable is cross-coupled as a balanced line, the impedance of a PE insulated cable is 40 – 50 Ω depending on the conductor cross-section. This is approximately 2.5 to 2.8 times lower than a twisted pair cable if the conductors have the same cross section! A thick cable has usually lower impedance than a thin cable because there is more copper compared to insulation. For example, the impedance of a 4 × 0.34 mm^2 cross-coupled cable is approximately 49 Ω, but for a 4 × 2.5 mm^2 cable, it is approximately 45 Ω. Max-i is designed for cables with characteristic impedance down to 35 Ω. The impedance between **two opposite** conductors is 3 times more, that is, 120 – 150 Ω because the self-inductance is 3 times bigger and the capacitance 3 times smaller. In practice, the impedance may be even lower than 40/120 Ω due to the input resistance and capacitance of connected devices.

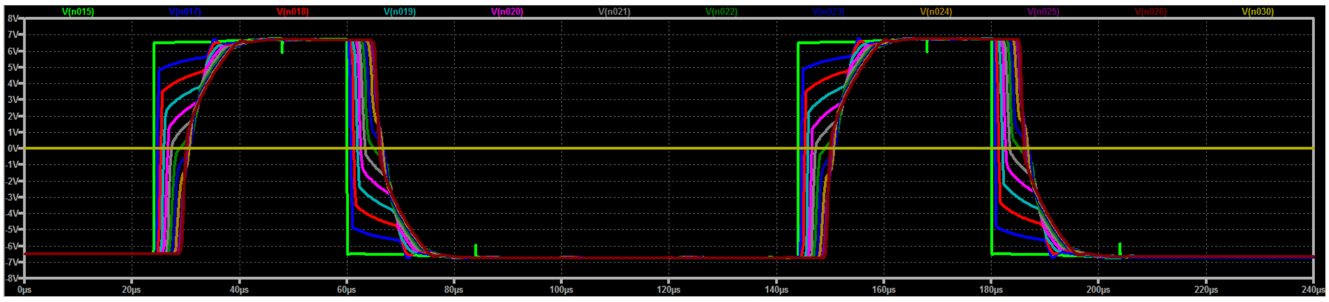
A.10 Point-to-Point Communication and Reflected Wave Switching

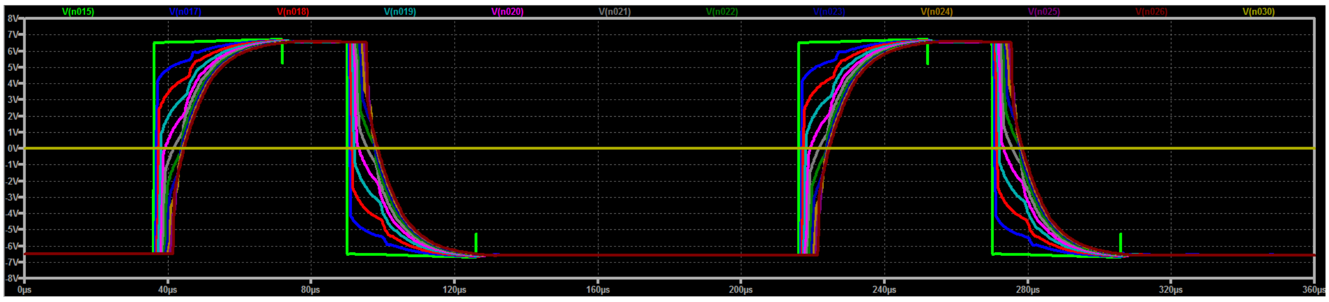
Since Max-i does not use termination resistors, it is always possible to charge a cable to the final voltage level **no matter the cross section of the cable** if the leakage current is sufficiently low. It is just a matter of the number of back and forth waves (N) during the pulse duration. This is illustrated below at 11 points on the line (0 – 100 % in steps of 10 %):

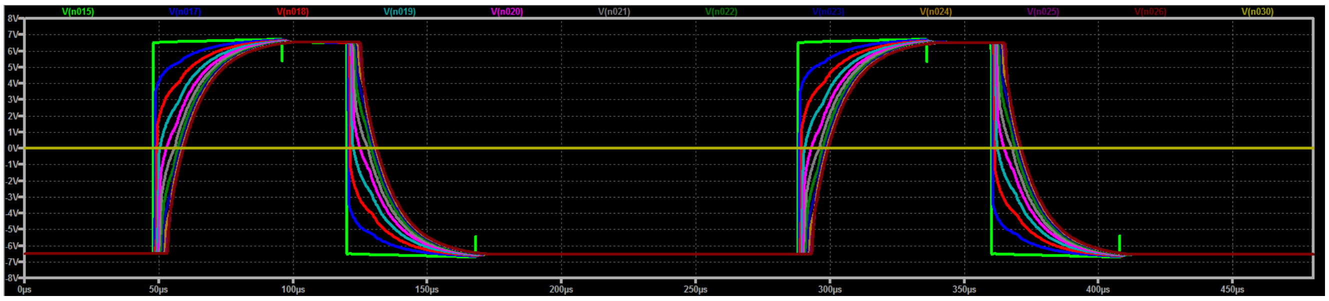

Incident wave switching 2 Ω


Incident wave switching 20 Ω

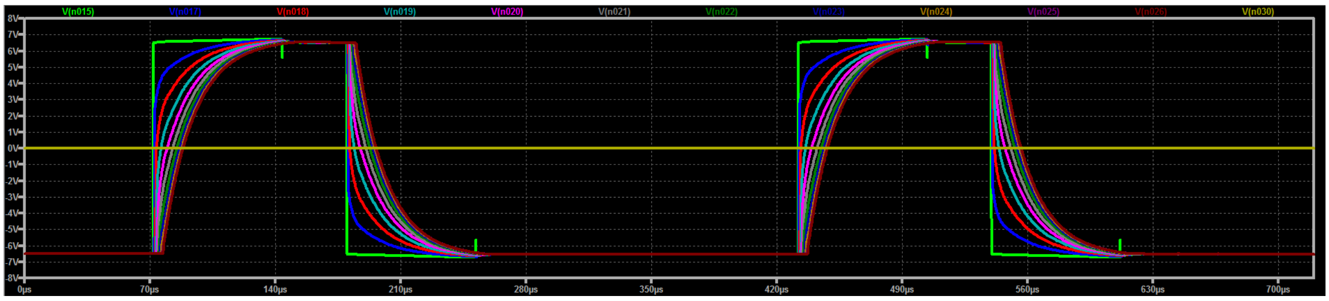

Point-to-Point 80 Ω


2-waves reflected wave switching 120 Ω

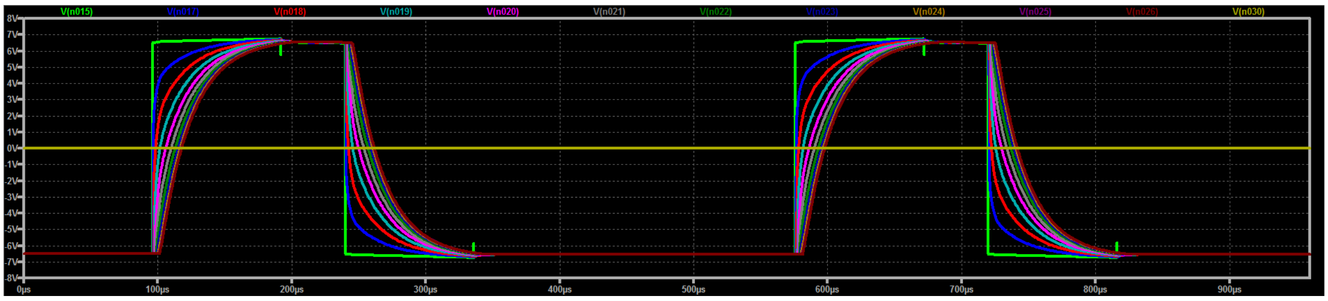

3-waves reflected wave switching 180 Ω



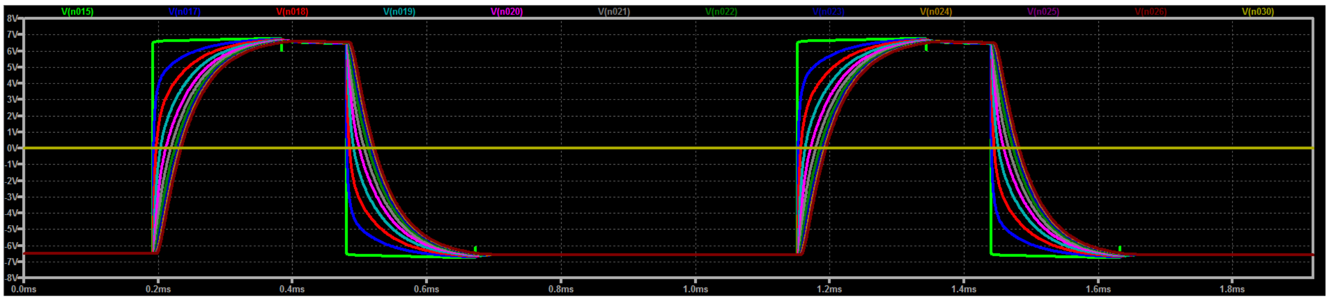
4-waves reflected wave switching 240 Ω



6-waves reflected wave switching 360 Ω



8-waves reflected wave switching 480 Ω



16-waves reflected wave switching 960 Ω

Fig. A.24

The first figure shows ordinary incident wave switching (N = 1) on a 45 Ω transmission line (250 µH/km, 123 nF/km and 5.55 µs/km) with a length of 92 % of maximum, which has a DC loop resistance of 2 Ω. This is the typical waveform on low-loss lines for example for smart-house applications. This and the following figures show some ringing, but this is because the transmission line is simulated by means of lumped network consisting of only 10 RLC transmission line models in series (10 × 0.2 Ω, 10 × 25 µH and 10 × 12.3 nF).

The second figure shows the waveform if the loop resistance is increased to 20 Ω. As can be seen, the minimum voltage level is approximately 2/3 of maximum (±4.3 V compared to the nominal ±6.5 V), which is acceptable, but more resistance is not recommended although it is possible. As shown below, even 40 Ω does not change the zero-crossings and can therefore still use incident wave switching:

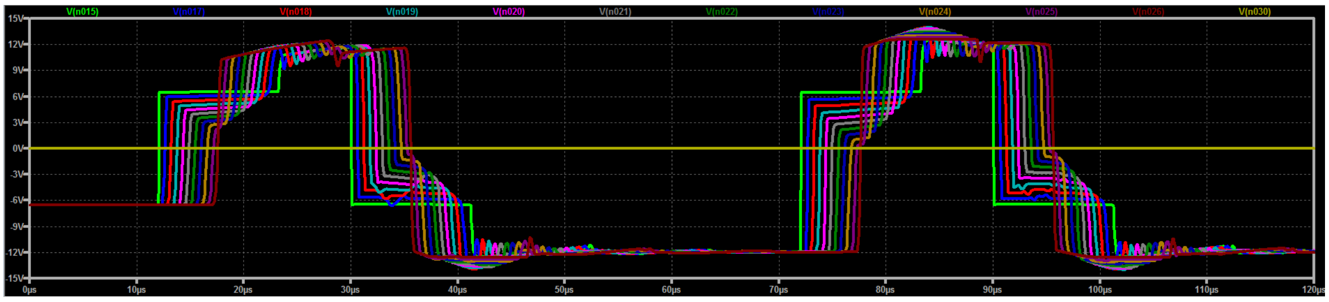
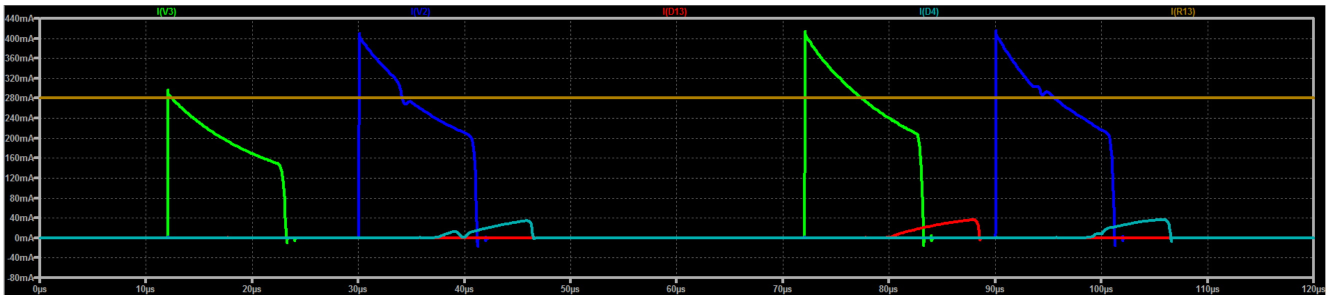

Incident wave switching 40 Ω

Fig. A.25

Note that the line levels between the bits are in this case close to the clamp levels and not to the transmitter levels. Also note than when the loop resistance is increased beyond approximately this point, the clamp network(s) are only limited active or not at all as can be seen below compared to the currents at 2 Ω:


Incident wave switching 40 Ω

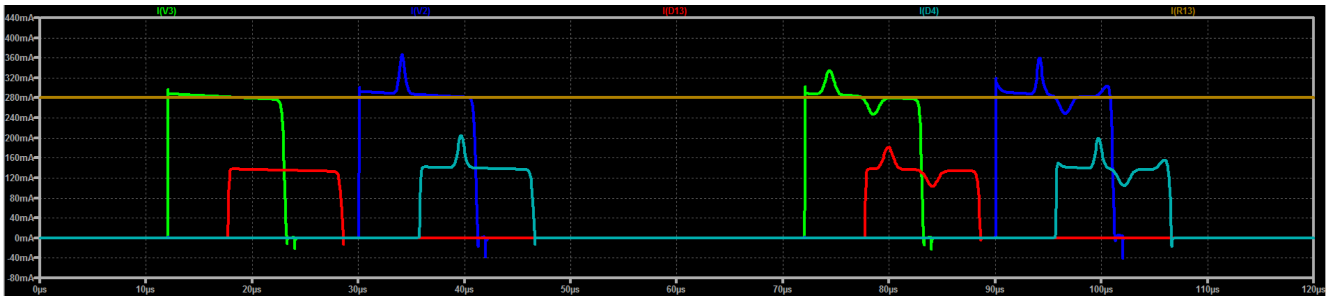

Incident wave switching 2 Ω

Fig. A.26

The green and blue curves show the transmitter currents and the cyan and red curves the belonging clamp currents, which are quite small in the 40 Ω case. Note that because of the higher line levels between bits, the initial voltage swing may be up to 50 % higher than in the ideal case (3 × 6.5 = 19.5 V) and the maximum current is therefore also higher, but as can be seen, the current is reduced quite fast due to the loop resistance so the average current (280 mA shown orange) and with that the power dissipation in the transmitter is approximately the same.

If it is only necessary with devices very close to the two ends, the reflections may be utilized to reduce the cross section a factor 4 as shown on the third figure. Note that when the line resistance becomes comparable to the self-inductance, the characteristic impedance is no longer almost resistive as described above. This creates a low-pass filter with a waveform very similar to the one created by the skin effect, that is, with a fast rising edge followed by a very slow final charging (blue curve).

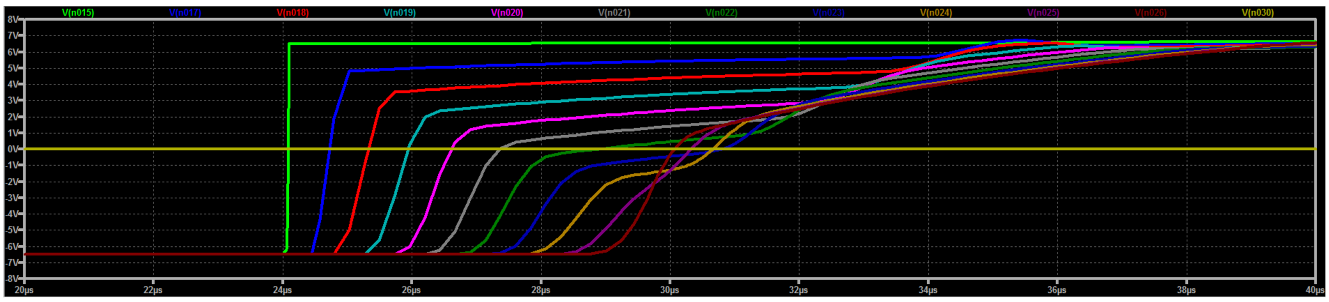
If a high cable resistance is desirable for example to keep the cross section at an acceptable level and/or it is necessary with a more free topology than a trunk line, reflected wave switching can be used if a lower speed can be accepted. This is shown on the following figures. N is the number of waves on the line and must therefore be an integer. The maximum DC resistance is N × 60 Ω for N ≥ 2. As can be seen, the voltage is slowly build-up by more back and forth waves to a fully acceptable waveform and the more reflections there are, the smoother the waveform becomes like an RC low-pass filter!

In this way, it is possible to use almost any length of cable and also free topology as long as the leakage current is sufficiently low and the lower speed is acceptable.

This is a great advantage of Max-i compared to fieldbus systems, which use termination resistors, and it makes it possible to use Max-i for street lighting and street information systems without too much copper. It is only necessary to keep the supply voltage within the limits at all load conditions.

A.10.1 Detection Delay with Reflected Wave Switching

With incident wave switching, the reception delay is always (R + D) where R is the rise time and D is the delay in the receiver as described in layer 1. This is however not the case with reflected wave switching where many waves may be necessary. This may create differences/jitter in device synchronization, which may limit the safety margin for counteracting. If for example N = 2, devices up to approximately 60 % line length use incident wave switching, but after that point including the end point, at least one more wave is needed. A device located at the end of the line (100 % = wine red) therefore detects the signal earlier than a device located in the 70 % worst case point (blue), but later than a device located in the 60 % point (green) as shown below:



2-waves reflected wave switching 120 Ω

Fig. A.27

Based on the difference between zero-crossing at incident wave switching on a lossless line and the actual zero-crossing in the simulations above, the worst case synchronization difference/jitter is:

N	T [µs]	Loop resistance R_S [Ω]	Worst Case point [%]	Simulated extra delay in worst case point [µs]	Jitter [T]
1	6	≤40	90	0.02	0.00
2	12	120	70	2.73	0.23
3	18	180	60	4.26	0.24
4	24	240	80	6.14	0.26
6	36	360	90	10.73	0.30
8	48	480	90	16.39	0.34
16	96	960	90	36,58	0.38

Fig. A.28

As can be seen, the additional jitter due to synchronization differences may be regarded as <0.4T.

A.11 Load Switching and Communication

If there are no capacitors, the **immediate** voltage drop or raise in voltage due to load switching is equal to the line impedance multiplied with the change in current. It does not depend on the DC resistance as even the shortest cable has the same characteristic impedance.

The figure below shows a computer simulation of the behavior of a 10 m PE insulated transmission line (50 ns propagation delay) with a characteristic impedance of 35 Ω and a loop resistance of 0.1 Ω when a load of 200 µF in parallel with 4 Ω is connected and disconnected. A situation like this occurs for example when a typical capacitive 5-A load is connected and disconnected in the worst case point in the middle of a 20-m, flat cable loop with a characteristic impedance of 70 Ω.

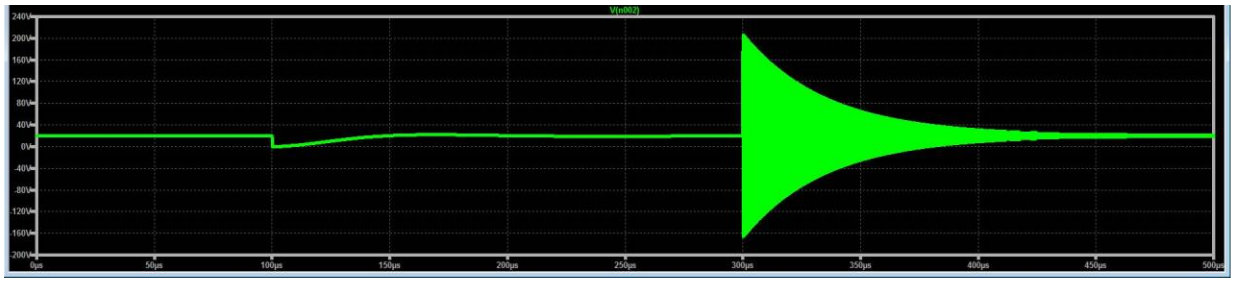


Fig A.29

When the load is **connected**, the capacitor initially works as a short circuit and the voltage therefore falls to 0 as shown. It then rises slowly during a period depending on the capacitance and the distance from the power supply to the load. If there were no load capacitor, the voltage would initially only fall to 20 V × 4 Ω / (35 Ω + 4 Ω) = 2.05 V.

When the load is **disconnected**, the voltage will rise with dI × Z_0 = 5 A × 35 Ω = 175 V and therefore reach a peak value of 195 V as shown if the supply voltage is 20 V. A heavy ringing will then occur until it is damped out by the loop resistance.

It is obviously that both situations are totally unacceptable.

A.11.1 Load disconnection

The ringing, which occurs when the load is disconnected, is however very easy to damp by means of a small (ceramic) capacitor in series with a resistor. If for example just a 2 µF capacitor in series with a 0.8 Ω resistor is added between L+ and L-, the situation looks like this:

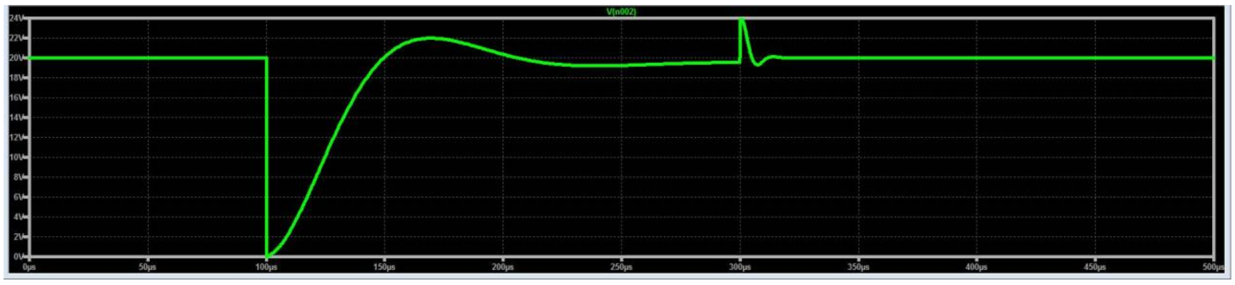


Fig A.30

The voltage still falls to zero, when the load is applied, but the ringing is now reduced to just a short 4 V peak, which is within the acceptable voltage range, and it is easy to reduce it even further with a bigger capacitor and a smaller resistor. Note however, that without the resistor and just the 2 µF capacitor, a ringing with even slightly bigger amplitude is generated as shown below so this is not recommended although it may be counterintuitive that adding a small serial resistor gives a better damping!

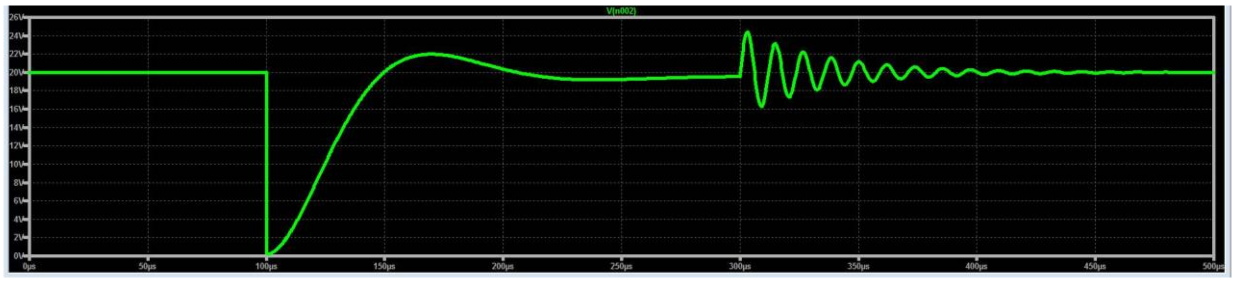


Fig A.31

If the capacitor is increased, the ideal damping resistor is decreased correspondingly – for example to 0.16 Ω in case of a 10 µF capacitor. If the capacitor gets **much** bigger, which may anyway be necessary for some loads, the ideal damping resistor gets so small that it may be omitted. If for example a 200 µF capacitor is used instead of 2 µF and no resistor, the situation becomes:

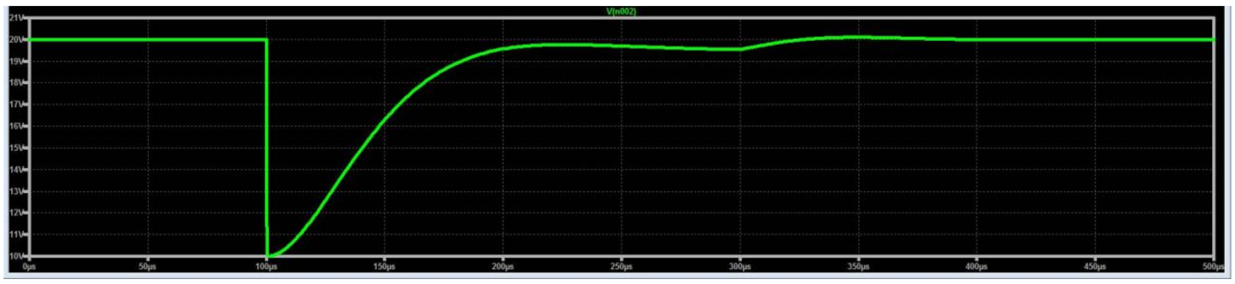


Fig A.32

This removes the ringing problem entirely, but there is still the problem with the voltage drop when the load is applied. Note that in the shown case, the initial voltage drop is only 10 V since the decoupling capacitor and the load capacitor are equally big.

When a capacitor is added at the point of load, the initial voltage swing is reduced to zero so there is no initial wave. The voltage at the load will then change with an initial speed of dV/dt = I/C. As the voltage changes, a wave with increasing amplitude begins to travel towards the power supply where it is inverted and reflected. The result is a damped sine wave. If the capacitor is **much** bigger than the capacitance of the cable as it is the case with the 200 µF capacitor, the line behaves virtually identical to the simple RLC network shown below:

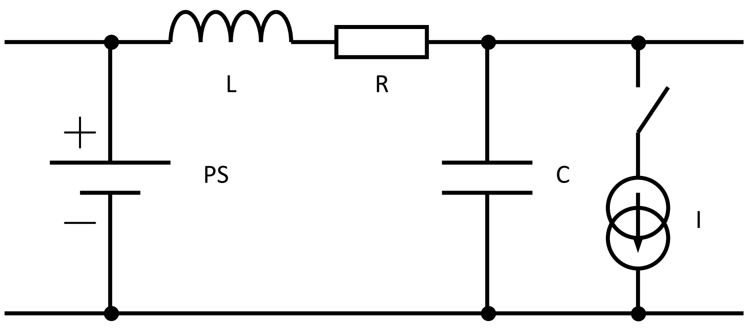


Fig A.33

PS is the power supply. L is the **loop** self-inductance of the cable, which is close to 900 µH/km for most cables, R is the **loop** resistance, C is the point of load capacitor and I is the load current. In practice, many loads are resistive, but this has virtually no influence on the waveforms.

The maximum loop resistance for a given accepted steady state voltage drop is of course:

$$R\,[\Omega] \le \text{VoltageDrop}\,[V] / \text{LoadCurrent}\,[A]$$

If for example the load current is 0.3 A and a voltage drop of 1 V is accepted on a long line, the maximum loop resistance is 3.33 Ω. If a 4 × 1.5 mm^2 cable is used (typical 12.5 Ω/km per conductor), the distance from the load to the power supply must not exceed 133 m.

To completely avoid an overshoot on the voltage drop, the circuit shall at least be critical damped, which means that the quality factor Q of the RLC circuit shall be ≤ 0.5 (or the damping factor ζ shall be ≥1). In practice, R is increased for dynamic signals due to the skin effect. Tests have shown that the DC resistance (R_{dc}) may be multiplied with a factor 1.2 without coursing any overshoot. Therefore, the minimum capacitor for an immediate change in current may be calculated as:

$$C \ge L / (Q \times 1.2 \times R_{dc})^2 \;\Rightarrow$$

$$C\,[\mu F] \ge 2.78 \{ 900\,[\mu H/km] \times \text{DistanceToPS}\,[km] \} / (R\,[\Omega])^2$$

The factor in curly brackets is the loop self-inductance. Since the distance to the power supply may be calculated from the cable resistance per km and the loop resistance, the formula may be simplified to:

$$C\,[\mu F] \geq 1250 / (\,\text{ConductorResistancePerKm}\,[\Omega/km] \times \text{LoopResistance}\,[\Omega]\,)$$

This formula has the further advantage that it may be used even in case of more power supplies. If two power supplies are used, the distance to the supplies may be doubled without increasing the loop resistance. However, as there is now two self-inductance's in parallel, the net self-inductance is not changed either. Since the conductor resistance for a 1.5 mm^2 cable is approximately 12.5 Ω/km and the loop resistance in the above case is 3.3 Ω, a 33-µF capacitor may be used. If the capacitor is smaller, the overshoot may decrease the performance. If the capacitor is bigger, it will just take longer time to reach the steady state condition, but nothing is gained with that. Note that the size of the capacitor does not depend on the load current or accepted voltage drop. Since the self-inductance per km is virtually constant for all practical cables, the capacitor is primary a function of the cable cross-section and the loop resistance between the power supply and the load.

For a 1.5 mm^2 cable with a typical conductor resistance of 12.5 Ω/km, the formula may be simplified to:

$$C\,[\mu F] \geq 100 / \text{LoopResistance}\,[\Omega]$$

With for example a 0.24 A load, the maximum loop resistance is 4.17 Ω. On a 1 km balanced 4 × 1.5 mm^2 cable, this can be obtained in any point by means of two power supplies located 1/6 line length (167 m) from each end. In this case, an additional capacitor of at least 24 µF must be used for fast switching, but as explained below, the necessary capacitor may be reduced to a few µF if the rise time is increased to just 100 – 200 µs and removed entirely if the fall time can also be increased.

The same formulas apply to the communication. The only difference is that because only the half transmitter current goes through the capacitor (one line is discharged directly and the other line is charged trough the capacitor), only half of the transmitter current shall be used when calculating the DC voltage drop. As described previously, the maximum average pulse current (⅓ of maximum) is approximately 250 mA for a trunk line with pure 0-bit transmission. Therefore, 125 mA may be used in the calculations. If for example an 800 m 4 × 1.5 mm^2 cable is used with two power supplies located 1/6 line length from each end, the maximum voltage drop due to the communication will only be 125 mA × 3.33 Ω = 0.42 V. The capacitor must still be approximately 33 µF to avoid undershoot, but in most cases, some undershoot may be accepted so that 2-3 times smaller capacitors may be used. If only a single power supply is used in the middle of the line, the loop resistance is 10 Ω, so that the worst-case voltage drop due to communication is 1.25 V. In this case, at least a 10-µF capacitor must be used to prevent any further voltage drop (1.25 V is already at the limit of being acceptable). As a rule of thumb, the necessary capacitance for communication is in the order of 25 µF/km if the capacitance is fairly equal distributed. This corresponds to a line impedance of 6 Ω (√ (900 µH / 25 µF)), which gives an immediate voltage drop of 1.5 V at 250 mA. As it is realistic to expect at least 12 devices per km, it is usually not necessary to worry about the communication if all devices contain a decoupling capacitor of at least 2.2 µF.

A.11.2 Load connection

The voltage drop when especially a capacitive load is applied as shown on the above images may disturb the communication and even cause reset or malfunction of the connected devices unless the decoupling capacitor is **much** bigger than the load capacitance, which may be impractical or even impossible. The solution to this is to reduce dI/dt. If for example the load is replaced with a current generator, which goes from 0 to 5 A in just 100 µs, the situation with a 2 µF, 0.8 Ω damping-network becomes:

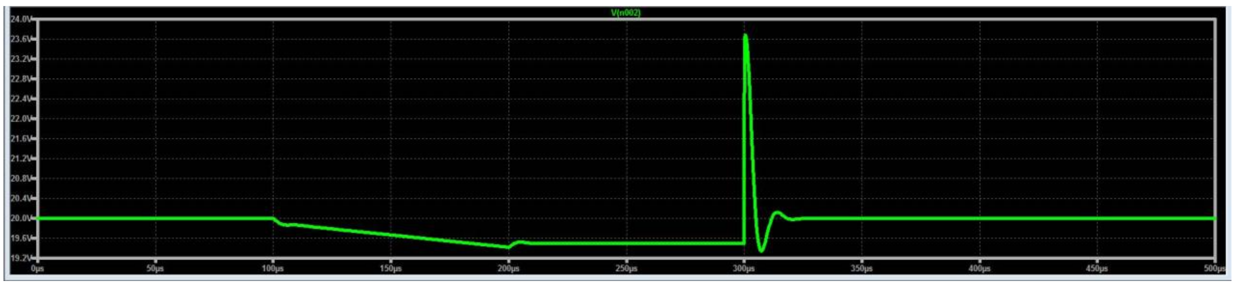


Fig A.34

As can be seen, this almost totally removes the voltage drop except for the part, which is caused by the loop resistance, so that only the turn-off spike is left. If however the line is longer, the small ringing, which is visible in the beginning (at 100 µs) and in the end (at 200 µs) of the current ramp, becomes bigger and may get a problem. Therefore, it is generally recommended to reduce the rise time of all loads and power outlets to 1 ms and/or use an S-shaped ramp (or use a **very** big decoupling capacitor). This removes the problem on all practical lines and all practical capacitive loads (up to more mF).

If it is possible to increase the fall time too, this will reduce the turn-off spike correspondingly so that very little capacitance is needed.

The voltage drop due to the self-inductance is approximately:

$$V = 0.9 \times Le \times I / t \text{ [V]}$$

Where Le is the line length in km, I is the maximum load current in A and t is the rise and fall time in mS.

Since the maximum practical cable cross-section is approximately 4 × 4 mm^2 with a 5-conductor flat cable in a loop topology so that the minimum loop resistance is 2.34 Ω/km, Le × I cannot exceed 0.43 A km (1 V / 2.34 Ω/km). If the rise and fall time is at least the recommended 1 ms, the voltage drop or raise due to the self-inductance therefore cannot exceed 384 mV, which is fully acceptable.

A.12 Line Simulation

For test purpose, a 1 km, 48-Ω, 4×1.5 mm^2 polyolefin insulated ($\varepsilon_r = 2.4$) **unbalanced** transmission line may be simulated by means of a lumped network consisting of 60 coil networks of each 3 coils and 2-3 resistors (total 248 µH, 12.6 Ω) plus 55 1.8-nF capacitors and 4 2.2-nF capacitors (total 108 nF) where two 2.2-nF capacitors are used instead of 1.8 nF in each end (4 × 2.2 nF = 5 × 1.8 nF) as shown below. To keep the line impedance close to 48 Ω also at high frequencies, the two coils in each end should be connected in parallel with a 47 Ω resistor. With the shown I/O network (2 × 2.2 nF and 47 Ω), the characteristic impedance is very close to 48 Ω from a few KHz to more MHz. Because the line is build by means of a finite number of sections (60), it may show a slight high frequency ringing at approximately 3 MHz.

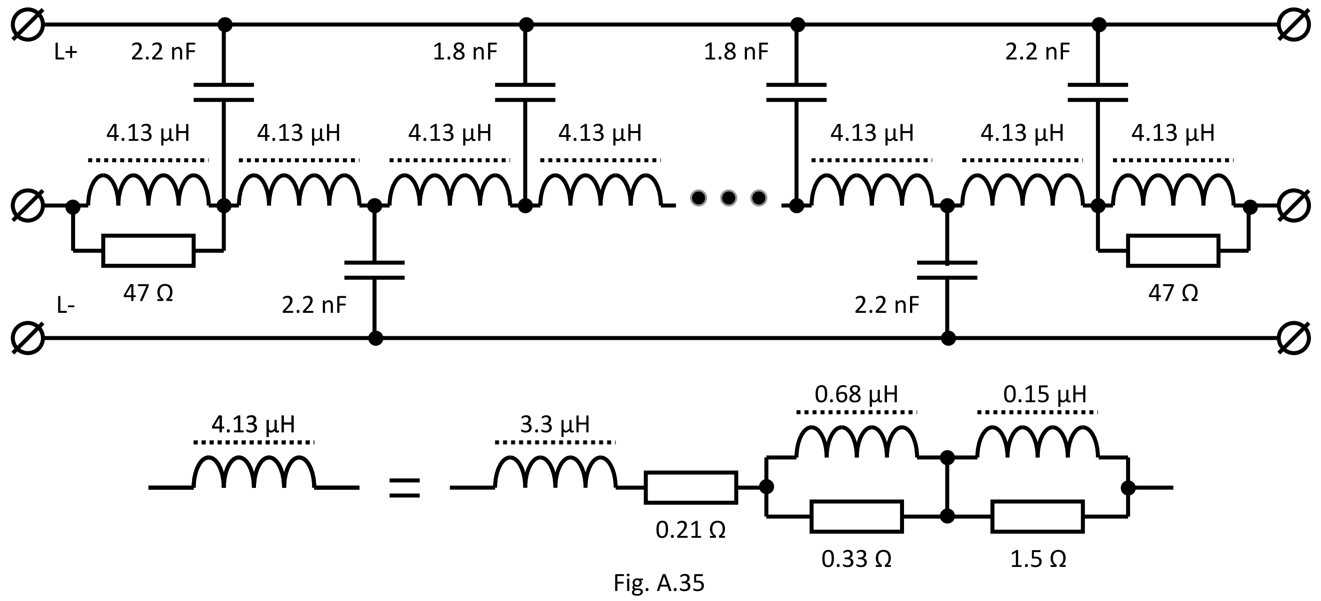


Fig. A.35

To simulate the skin effect, each coil in the lumped network consists of three coils and 2 – 3 resistors (the serial resistance of 0.21 Ω may be a part of the 3.3 µH coil). The propagation delay is approximately 5.4 µS/km, which is very close to the delay used in the specification of the maximum cable length.

The capacitors are alternately connected to L+ and L-. Note that the input capacitors are connected to L+ so that the line is pulled towards L+ at power-up.

WARNING. It is not recommended to connect devices directly to the various internal nodes of such a virtual transmission line. The capacitor in these nodes may look as a short circuit to the transmitters during the rising edge of the pulse (this is the reason why inductive input and output is used)! With a rise time of 100 nS, the added peak current due to the capacitance will be approximately 200 mA. If you want to have devices in the middle, use more shorter sections or connect the devices through a parallel connection of a 3.3 μH coil and a 47 Ω resistor.

Note that to simulate a balanced line, the self-inductance is the sum of the self-inductance of the conductors back and forth so all coils and resistors must be reduced to the half, when the number is doubled (60 coil networks in each line). The capacitors are the same.

Annex B. CRC Polynomial

B.1 Safety Systems

Max-i has been designed for safety systems in accordance with AK6 of DIN V 19250, category 4 of EN 954-1 and SIL3 of IEC 61508. This requires that the total system probability of a dangerous failure shall be less than 10^{-7} per hour (and cannot be claimed to be better than 10^{-8} per hour). Because Max-i is a new fieldbus system without sufficient failure data from field experience to support claims for rates of failures it is regarded as a type B system according to IEC 61508 chapter 7.4.3.1. Because a single error can lead to a dangerous situation (hardware fault tolerance = 0), the safe failure fraction or statistical confidence level shall be at least 99% for SIL3. This requires an **error free** test time of:

$$\text{Test time} = -\ln(1 - 0.99) / 10^{-7} = 4.6 \times 10^{7} \text{ hours}$$

This corresponds to 5000 years! To reduce this time to an appropriate level Max-i uses a 20-bit CRC polynomial. Such a polynomial has a worst-case probability of an undetected error of $2^{-20} \approx 10^{-6}$ and therefore reduces the test time to only 46 hours. Because a safety system trips each time an error is detected this means that if the system runs well without frequent false trips the requirement is fulfilled. In practice, much less test time is needed because the polynomial is able to find many errors on few bits with 100% probability and virtually no error words contain double-bit errors, which are the most common, but unfortunately it is impossible to calculate the ratio (N) between errors, which are found with 100% probability, and errors, where the residual error probability is 2^{-20}. Therefore, a test time of 46 hours must be used.

B.2 CRC Polynomial

Max-i uses the 20-bit CRC-polynomial:

$$G(x) = x^{20} + x^{14} + x^{13} + x^{12} + x^{10} + x^{7} + x^{5} + x^{3} + x + 1$$

This polynomial has been selected especially for Max-i as the best one with the following properties:

- Has a length of 20 bits.
- Includes (x + 1), which is a parity check.
- Has at least 8 terms or maximum 14 terms including x^{20} and x^{0} (1) so that a Hamming distance of 8 is possible.
- Have at least x^{19}, x^{18}, x^{17}, x^{16} and $x^{15} = 0$, so that at least the first 6 bits of the identifier are not scrambled and the ID number range therefore divided in at least 64 main priority levels, which gives explicit messages their own priority group with lower priority than all other telegrams.
- Has a Hamming distance of at least 8 up to at least 56 bits, which is the length of a Boolean telegram with a long identifier.

The polynomial has a Hamming distance of 8 up to 56 bit, 6 up to 152 bit and 4 up to 169290 bit. It is therefore always able to detect 7 randomly placed errors (Hamming distance - 1) in a 56-bit stream (7 bytes), 5 errors in a 152-bit stream (19 bytes) and 3 errors in a 169290-bit stream (21161 bytes). In this way, it is possible to transmit a boolean 4-bit value with Hamming distance 8, and 12 data bytes plus 4 bits with Hamming distance 6 if the long identifier is used and of course two more data bytes in case of the short identifier. By including cyclic signatures in the data stream, it is even possible to transmit an infinite long telegram with Hamming distance 4, 6 or 8.

The polynomial is also able to detect any single error burst up to 20 bit and any odd errors. Multiple errors bigger than 7 bits are not detected with a residual error probability of 2^{-20} or approximately 10^{-6}. Note that in a system like Max-i with a very low error rate, the practical error probability is less than this value because all the most common errors (errors on few bits) are detected with 100% probability.

The probability of an undetected 6-bit error in a telegram with two data bytes (20-bit FIX), is extremely low as there are only 17 possible undetected error words (see the table later in this chapter), so for all practical purposes the Hamming distance is 8.

The used polynomial is constructed as:

$$(x^{14} + x^{10} + x^{9} + x^{7} + x^{4} + x^{2} + 1) (x^{5} + x^{4} + x^{3} + x^{2} + 1) (x + 1)$$

where the last (x + 1) is a parity check.

In the receiver, the CRC-generation is done with the usual division, that is:

$$RD(z) = LD(z) / (G(z) / z^{20})$$

where LD(z) is the data on the transmission line and RD(z) is the decoded received data. The transmitter output is however scrambled as:

$$LD(z) = TD(z) \times (G(z) / z^{20})$$

where TD(z) is the data to be transmitted and LD(z) is again the data on the transmission line. It is obvious that when the two formulas are combined, the result is RD(z) = TD(z). Therefore, all data - not just the check word - can be taken from the CRC-generator output. It is important to notice that the CRC-check is generated over **all** bits except for the start bit and the priority bit - **even if more devices generate the telegram**.

As:

$$G(z) / z^{20} = 1 + z^{-6} + z^{-7} + z^{-8} + z^{-10} + z^{-13} + z^{-15} + z^{-17} + z^{-19} + z^{-20}$$

where z^{-1} represents a one bit delay, the CRC generator/checker may be very easily constructed by means of a 20-bit shift register, one 9-input XOR-gate and two 2-input XOR-gates as shown below.

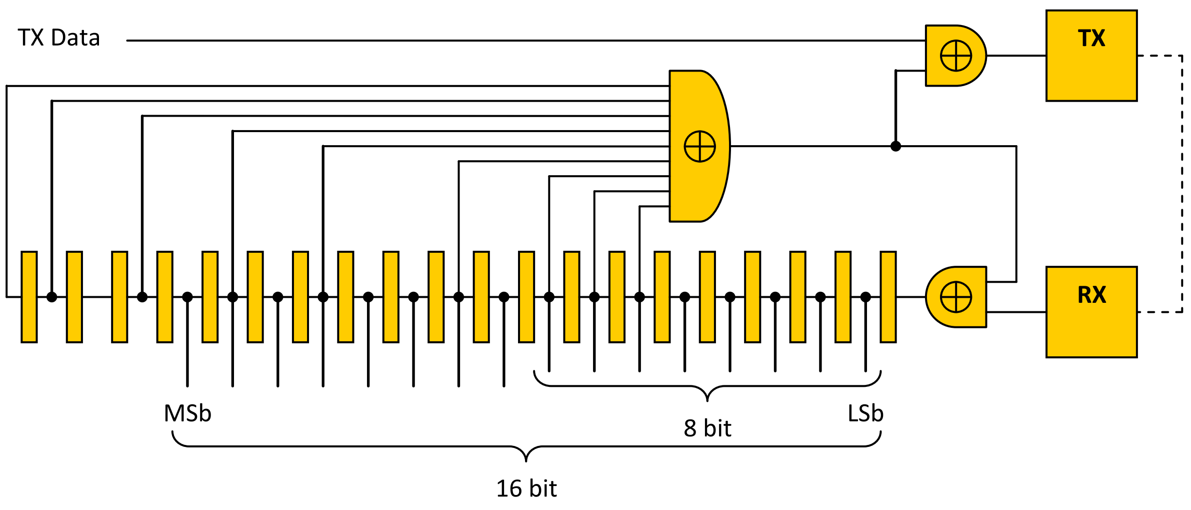


Fig. B.1

Note that during transmission the input to the shift register is not taken directly from the TX-data, but through the receiver. The transmitted data are XOR'ed with the data from the 9-input XOR-gate, but after they are received, they are again XOR'ed with the same value so the result will always be equal to the transmitted data - just delayed a little. The advantage of this system is that **no** TX/RX switching is necessary. The input to the shift register is **always** taken from the receiver and the data are loaded into the shift register each time the line changes state (a pulse is received). If the TX-data contains more than 20 successive 0's or 1's the output of the shift register will be a constant 0.

Also note that it is possible to take a complete parallel output word directly from the receiver shift register so that the three XOR-gates and perhaps a longer shift register are the only components needed for error detection beyond a normal UART! Besides, the transmitter and receiver may share the same shift register and XOR-gates, which makes the system very simple.

Because the average delay from the input of the scrambler to the output of the 9-input XOR-gate is approximately 13 bit, the CRC polynomial is very good to handle short error bursts. This is because the probability for a "compensation" error to strike exactly in the right positions decreases with the time. It is for example much more likely, that an error strikes two adjoining bits than two bits with a 13-bit distance.

The table below shows the first 50 error words with distance 6 and 4:

Error Word No.	Weight = 6			Weight = 4		
	Length	16-bit Data Words	Error position	Length	16-bit Data Words	Error position
1	57	0	56 29 24 15 1 0	153	6	152 107 78 0
2	57	0	56 19 18 9 4 0	226	10	225 91 58 0
3	58	0	57 54 49 32 20 0	242	11	241 180 79 0
4	59	0	58 42 36 22 17 0	264	12	263 183 175 0
5	60	0	59 40 32 16 11 0	289	14	288 266 229 0
6	63	0	62 48 31 27 9 0	292	14	291 284 181 0
7	63	0	62 34 24 16 7 0	299	15	298 203 34 0
8	64	0	63 60 54 33 4 0	305	15	304 214 156 0
9	64	0	63 54 28 19 5 0	320	16	319 310 164 0
10	65	0	64 60 23 20 18 0	331	17	330 277 86 0
11	66	0	65 63 52 31 22 0	336	17	335 84 13 0
12	66	0	65 52 36 8 7 0	350	18	349 307 188 0
13	67	0	66 48 38 37 7 0	368	19	367 275 118 0
14	70	0	69 62 55 38 24 0	382	20	381 304 247 0
15	71	0	70 63 45 36 18 0	383	20	382 283 46 0
16	71	0	70 40 20 19 2 0	407	21	406 189 124 0
17	72	0	71 39 32 24 13 0	428	23	427 327 25 0
18	73	1	72 67 60 13 7 0	439	23	438 279 128 0
19	74	1	73 69 25 20 1 0	442	24	441 221 112 0
20	74	1	73 67 51 17 5 0	448	24	447 292 279 0
21	75	1	74 56 52 50 13 0	449	24	448 413 54 0
22	76	1	75 74 53 43 25 0	451	24	450 182 116 0
23	77	1	76 65 48 33 3 0	452	24	451 201 22 0
24	77	1	76 51 46 13 5 0	453	24	452 423 231 0
25	78	1	77 73 54 23 7 0	454	24	453 235 29 0
26	78	1	77 60 51 10 8 0	462	25	461 446 52 0
27	79	1	78 63 56 33 17 0	464	25	463 413 2 0
28	79	1	78 59 50 23 11 0	469	25	468 253 166 0
29	79	1	78 55 46 45 29 0	481	26	480 389 363 0
30	80	1	79 53 42 37 34 0	483	26	482 360 158 0
31	81	1	80 71 58 45 42 0	491	27	490 422 405 0
32	81	1	80 45 14 13 6 0	495	27	494 437 207 0
33	81	1	80 43 17 15 3 0	498	27	497 345 231 0
34	82	1	81 73 71 48 28 0	520	28	519 460 345 0
35	82	1	81 65 53 31 15 0	525	29	524 499 4 0
36	83	1	82 62 49 30 22 0	527	29	526 366 350 0
37	83	1	82 60 33 27 15 0	529	29	528 121 56 0
38	84	1	83 75 59 19 17 0	532	29	531 313 152 0
39	84	1	83 53 39 34 19 0	542	30	541 302 227 0
40	85	1	84 76 66 53 32 0	556	31	555 401 230 0
41	85	1	84 76 64 23 1 0	565	31	564 457 61 0
42	86	1	85 82 78 51 27 0	567	31	566 515 435 0
43	86	1	85 76 52 48 25 0	567	31	566 427 252 0
44	86	1	85 70 45 30 3 0	568	31	567 307 36 0
45	86	1	85 65 62 53 8 0	569	32	568 253 141 0
46	86	1	85 57 53 40 28 0	571	32	570 351 340 0
47	87	1	86 85 79 58 4 0	572	32	571 207 190 0
48	87	1	86 81 75 70 23 0	574	32	573 417 212 0
49	87	1	86 78 69 54 18 0	577	32	576 532 458 0
50	88	1	87 85 60 37 16 0	577	32	576 487 380 0

Fig. B.2

As shown, there is only one error word with weight 4, which can only handle 6 data words (plus 4 bits) with the long identifier. As the probability for this error sequence to happen is vanishing small, the Hamming distance may be considered at least 4 up to at least 10 words for all practical purposes.

Note that because the polynomial includes a parity check, all 5-bit or 7-bit errors are always detected, so the Hamming distance is 8 for weight 6 and 6 for weight 4.

The theoretical maximum length for weight 6 for a 20-bit polynomial with parity check is $2(x^6 - 1) = 126$ since:

$$(x^6 + ... + 1) (x^6 + ... + 1) (x^6 + ... + 1) (x + 1) (x + 1) = (x^{20} + ... + 1)$$

The double parity check $(x + 1)$ multiplies the length by two.

The theoretical maximum length for weight 4 is $2(x^9 - 1) = 1022$ since:

$$(x^9 + ... + 1) (x^9 + ... + 1) (x + 1) (x + 1) = (x^{20} + ... + 1)$$

Because the two maximum lengths are constructed differently, it is obvious that a polynomial cannot both have a length of 126 with weight 6 and simultaneously a length of 1022 with weight 4, so it is necessary to search through a lot of polynomials to find the best compromise. In case of Max-i, the most important property is that the length is at least 57 at a Hamming distance of 8 so that boolean telegrams can be handled with maximum safety – even in case of the long identifier. Out of all tested polynomials there are only 5, which can fulfill this requirement – 4 with length 57 and one with length 58.

Annex C. LED Drive

C.1 LED Light with Variable Color

Unlike incandescent (filament) lamps, LED's generates a fairly narrow spectrum. When the light goes directly into the eyes like for example a TV set or LED ropes and LED walls, it is usually enough with 3 LED's – red, green and blue. This makes it possible to generate all colors within a triangular area defined by the color coordinates of the LED's as shown below.

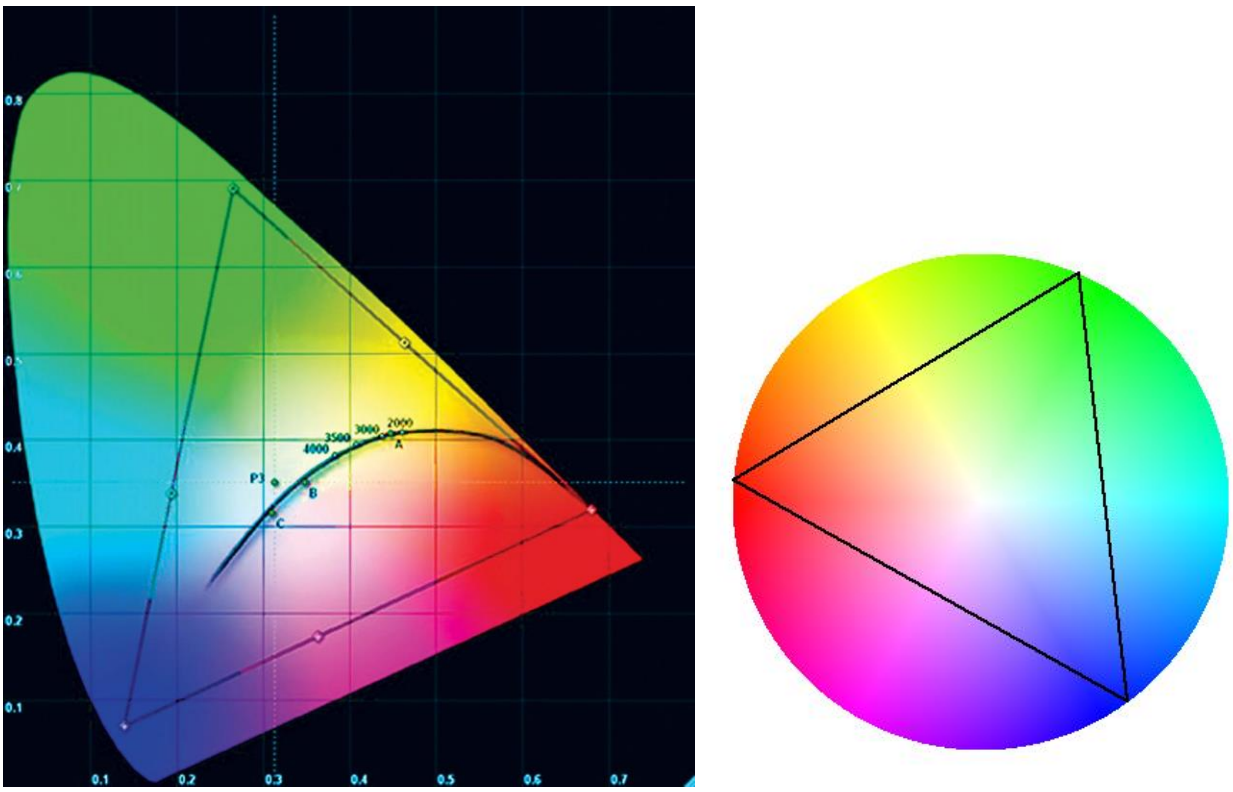


Fig. C.1

Colors outside the triangular area cannot be reproduced. This is especially noticeable at the color wheel where it is obvious that the only pure saturated colors are the LED colors. Pure saturated yellow, cyan (blue-green) and magenta cannot be reproduced without more LED's, but this is accepted in most systems although systems with up to 7 LED's to cope with this problem are seen.

The eye has only 3 color receptors/cones – S, M and L as shown below:

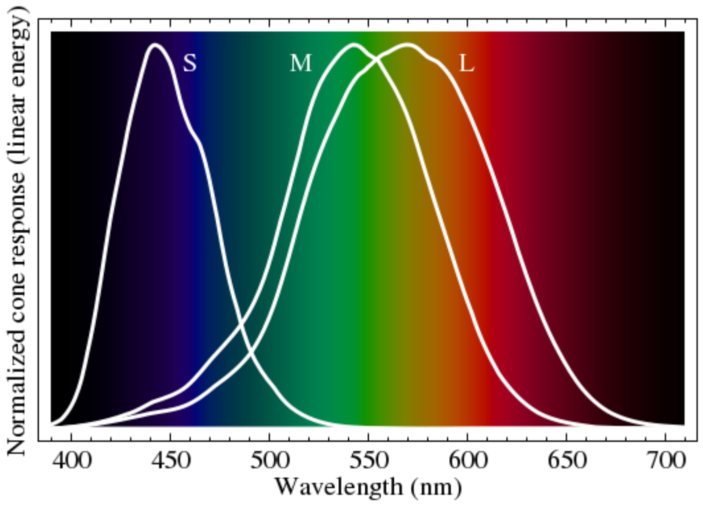


Fig. C.2

The three cones does not directly correspond to the blue, green and red LED's, but are more like violet, green and amber. The brain primary senses the color as the difference D1=(L+M)-S, that is, the difference between yellow, which is the color between M and L, and S, which is violet. Because of the subtraction, (L+M) and S become so called complementary colors to the brain. If (L+M)=S, the D1 is 0 and no color is sensed that way, so D1 **alone** only defines the colors from violet over cold white to warm white and yellow. On the color wheel, violet and yellow are opposite to each other and white is in the middle. In the same way, the brain also calculates the difference D2=L-M. This difference is then used to modify the color from D1 in a direction on the color wheel perpendicular to the violet-yellow axis so that for example yellow is modified to the range between green and red, and violet is modified to the range between cyan and magenta. Red-green color blind people cannot make this modification and therefore cannot see red. Note that complementary colors do not exist in reality – only in the brain, and that there is no single wavelength corresponding to magenta, which can only be made by means of a mix between blue and red. Also note that while violet and yellow are complement colors both on the color wheel (opposite to each other) and to the brain, green (M) and amber (L) are complementary colors to the brain, but not on the color wheel (not opposite to each other).

On the left drawing of fig. C.1, the two sense-lines goes from violet in the lower left corner to yellow and from red in the right corner to cyan in the middle of the left side. Because of this, it is possible to cover a slightly bigger range than a pure triangle between red, violet and a fictive green in the upper left corner. The two lines cross each other at white in the middle. The curved line shows white at different color temperatures. In practice, a precise white generation requires that the 3 LED's are bin selected or can be calibrated because the light output for a given current may otherwise vary as much as ±25%, which could change the color considerably as the eye is very sensitive to even small changes in white point.

C.2 LED Lighting with Variable Color

When LED's are used as illumination, the narrow bandwidth becomes a big problem. As can be seen below, there is hardly any light output in the cyan and amber range at approximately 490 nm and 580 nm.

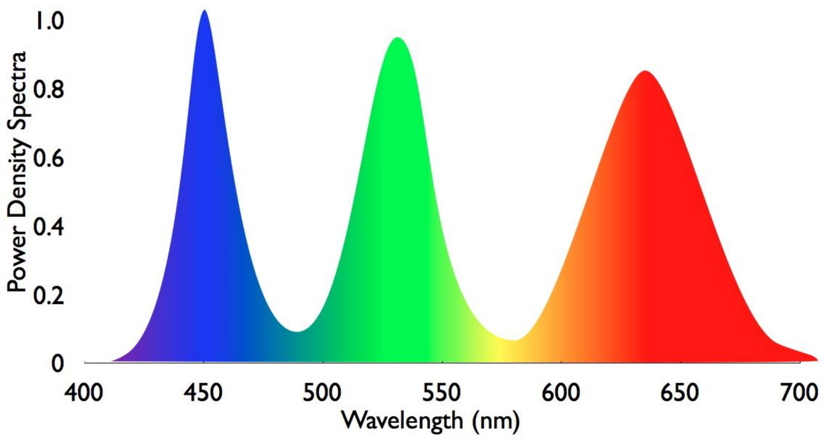


Fig. C.3

A pure yellow or amber color, which only reflects light in the 560 – 590 nm range, will therefore seem way too dark. In the same way, a pure cyan color, which only reflects light in the 470 – 500 nm range, will also seem too dark – especially because the sensitivity of the S and M cones are already quite low in this range – see fig. C.2.

There are two common ways to solve this problem - add an amber LED and preferable also a cyan LED or add a white LED. Both methods have pros and contras. The white LED is usually made by means of a blue LED, which stimulates a yellowish phosphor. The resulting spectrum is not ideal with a typical high blue peak, and the efficiency in the 500-700 nm range is not very high as shown below:

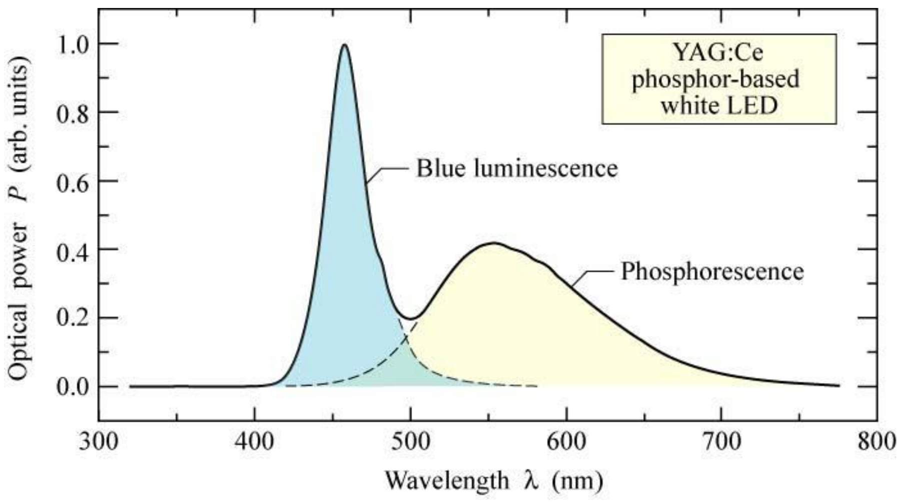


Fig. C.4

If the blue peak is not properly damped and/or there is not enough red, skin tones will look pale – even if the LED is marked as warm white (2700K). For this reason, a white LED is sometimes combined with a red, orange or amber one.

If an amber LED and perhaps a cyan LED is added to red, green and blue instead, up to 5 colors are generated, and the theoretical efficiency is higher, but in practice the difference is not so big. Usually colored light is best generated by means of RGBA and it is easier to convert an RGB color to RGBA(C) than RGBW. On the other hand, white with different color temperatures is best generated by means of RGBW or as simple as AW, where a cold white LED is selected. This is used in Max-i for dim-to-warm.

No matter which method is used, it is obvious that if a variable illumination color is wanted (not just white - with or without variable color temperature) it is necessary with at least 4 channels. Some stage lamps use RGBAW or even 7 colored LED's, but more than 4 channels are not directly supported by the present Max-i controller, which can only grasp 4 × 8 bit. Therefore, Max-i uses an RGBWaAaC color model where artificial amber (aA) is generated on the basis of red and green and artificial cyan is generated on the basis of green and blue. This is simply done by taking the lowest value of the two. If for example R = 127 and G = 63, aA becomes 63. Note that if one of the channels is 0, aA will also be 0 so that pure red and green is still possible. This principle works very well. The usual pale yellow of a pure RGB system is converted to a bright, saturated yellow, and the bluish white, which is often seen in such a system – especially at low levels, is gone.

C.3 LED Dimming

LED's are constant voltage device, which should be driven with a constant current and may be dimmed in two ways – by changing the current or by changing the duty cycle, but if the current is varied too much, the color hue will change so in practice LED's are dimmed by means of a variable duty cycle. Supplementary dimming by means of the current may however be very convenient to implement protection circuits for example for brownout (low voltage) and high temperature.

The traditional way to make variable duty cycle is PWM with a frequency ≥100 Hz – preferable ≥200 Hz to avoid flicker. However, this method has more disadvantages – especially for fieldbus applications where the power supply is not located in the same spot as the LED:

- Since the current is turned on and off very fast and the loop resistance is often very low due to the necessary thick cables and fairly short distances to power supplies, an inexpediently big decoupling capacitor may be needed (see chapter "Load Switching"). Each time the loop resistance is reduced to the half, the necessary capacitor for fast switching is increased 4 times.
- A low frequency may create stroboscopic effects with moving head lamps.
- The LED is stressed because the temperature of the die varies during each cycle. This reduces the life time, and during the hot period, the output drops and the color may change.

A much better way is PCM as explained in Chapter 7, Application layer.

C.3.1 Low Voltage Drive

If there are no more than 4 – 5 LED's in series, they may be driven directly from the 20-V Max-i supply voltage in two ways:

- By means of a simple current generator, which is controlled directly by the PCM. This solution is very cheap and may achieve a very high efficiency of 75 – 80 % over the entire voltage range if it is possible to bypass one LED in case of a low voltage as shown below:

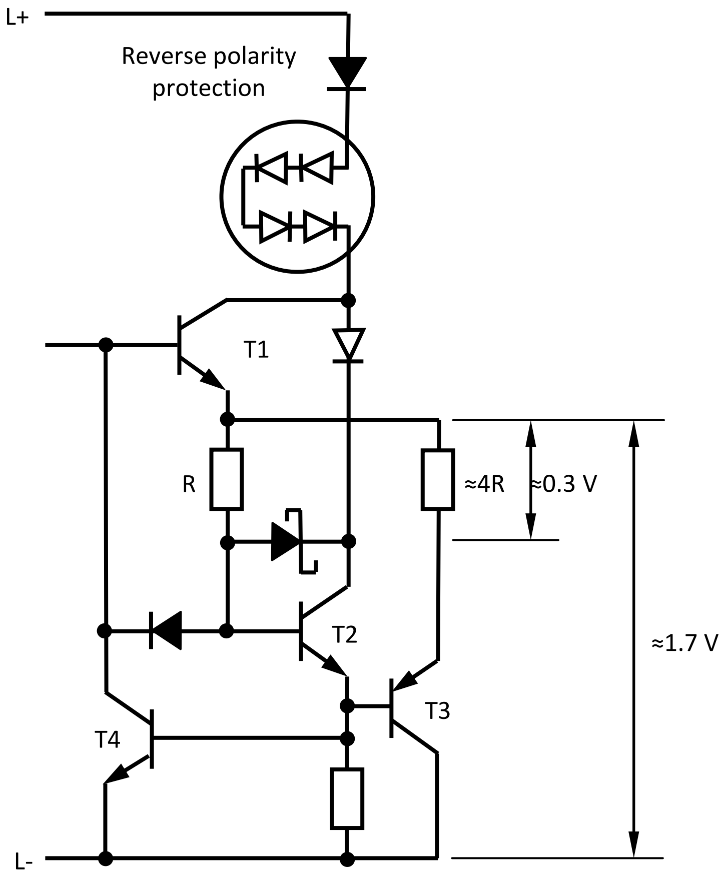


Fig. C.5

The diode in the basis of T2 reduces the turn-off time. During normal operation, the currents through T1 and the current mirror transistor T3 are very small and the current through all 5 LED's therefore virtually the same as the current through T2. If the voltage falls so much that the schottky diode turns on, T2 becomes a diode and T1 overtakes the LED current. This will bypass one LED and therefore reduce the light output, but by means of T3, extra compensation current is added to keep the light output constant. At the same time, the voltage drop over resistors is increased from approximately 0.7 V (basis-emitter of T4) to approximately 1 V, which is small enough to ensure that the current is reduced to less than 50 % at the minimum operating voltage as it shall be. If the voltage drop over the current mirror resistors is 0.3 V and the voltage drop over each LED's is approximately 3.4 – 3.7 V, which is typical the case for white LED's, the brownout limit becomes the required 16 – 17 V. A typical (simulated) behavior of a 16-W lamp with 1050 mA LED's is shown below:

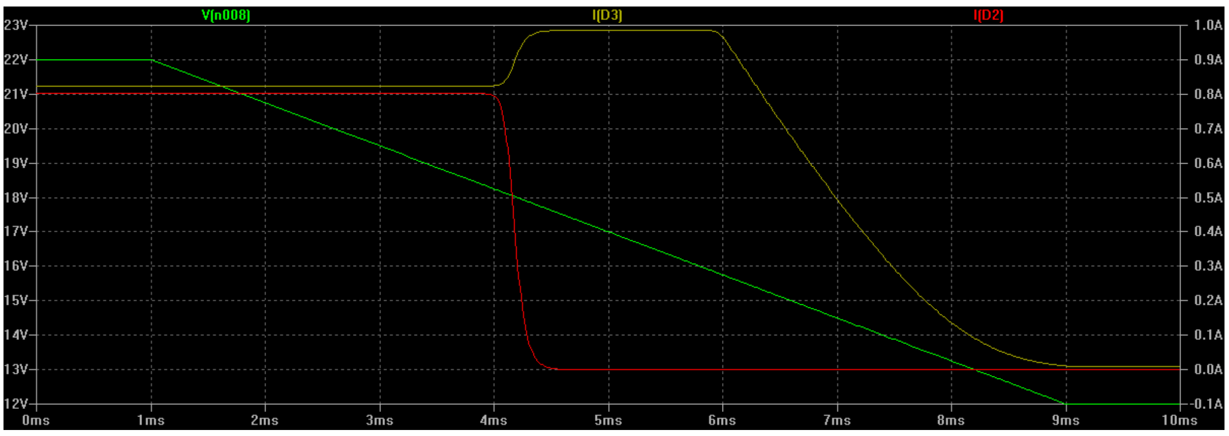


Fig. C.6

The green curve is the input voltage, the amber curve is the current through the 4 LED's and the red curve the current through the single LED. When the voltage falls below approximately 18.5 V, which is just above the upper brownout limit, the single LED is bypassed and the current increased in the other LED's to keep the light output constant until the brownout voltage where the current shall begin to fall. At 20 V, the efficiency is approximately 85 %.

- By means of a buck converter (step-down) in continuous conduction mode, which is the most efficient type of converter. The converter may be a simple hysteretic converter, which generates a saw tooth shaped current, which may be feed directly to the LED(s) without any parallel capacitor. Because it is not possible to stop the current in a coil, the LED(s) must be dimmed by means of a transistor (MOSFET), which can short circuit the LED(s) and stop the switch as soon as the minimum on-time has elapsed. To keep the loss in off-state as low as possible, it is recommended to use synchronous rectification, and to ensure that the average LED current is correct, an integrator should be included, which preferable only integrates the current while the LED(s) are on.

A simple buck converter may also be the ideal way to fire pyrotechnics. A typical pyrotechnic igniter has a cold resistance of 0.9 Ω ±0.15 Ω, requires a minimum voltage of 6 V in hot condition and is made in such a way that it **never** ignites below 0.3 A, but **always** above 1.35 A. With a buck converter, it does not matter if the igniter short circuits when it fires, which it quite commonly does. This will just reduce the power in the same way as the short circuit transistor, but of course not as much as if it open-circuits. It is recommended that the buck converter is able to limit the rise time of the current to at least 5 ms, but if this is not the case, this may be done by means of the smoothing filter in the Max-i controller and a short circuit transistor in the same way as for LED drive.

C.3.2 High Voltage Drive

If there are more LED's in series than 5, a boost (step-up) or buck-boost converter must be used unless a high voltage supply is present. Since such a converter runs in discontinuous conduction mode, an output capacitor is necessary, which makes it impossible to short circuit the LED's. With many LED's in series, the voltage drop over the LED's is however so big that a small, additional voltage drop over a current generator does not reduce the efficiency very much so the best way to drive such a chain is to make a linear current generator, which is then switched on and off by the PCM. To reduce the power loss as much as possible, the control loop in the converter shall not keep the output voltage constant, but instead try to keep the ideal voltage drop over the current generator – for example 1 V. There are numerous converters – both boost and buck – based on this adaptive principle.

C.4 Gamma Correction

The eye sensitivity to light is not linear, but has a logarithmic base of $100^{0.2}$ = 2.5 or more precise:

$$R'_{eye} = 9.033R \qquad R \le 0.008856$$

$$R'_{eye} = 1.16 \times R^{1/3} - 0.16 \qquad R > 0.008856$$

To be able to make a linear light dimmer, it is therefore necessary to implement the inverse function in the Max-i controller. Although the exponent 1/3 appears in the equation, owing to the offset and the scale factor, the overall curve is fairly close to a 0.41-power function so $R = R'^{2.44}$ may do although it is not perfect. This is called gamma correction where the gamma (γ) is 2.44 in this case.

The eye is able to distinguish differences in light levels down to 1%. When this (1.01) is raised to a power of 0.41, it is necessary with linear steps down to 1.004 = 0.4%. Therefore, each channel must have a linear resolution of 8 bit, which fits exactly with 0.4% and is also the same as used in digital TV. When an 8-bit value is raised to a power of approximately 2.44, minimum 13 bits are needed to represent the result, so the PCM (or PWM) should have a contrast ratio of at least 8192:1, which is close to the state-of-the-art for switch mode technology and more than the static contrast ratio of most LCD panels. Note that gamma correction is an efficient, virtually lossless way to compress video data as 13-14 bit linear data from the camera is compressed to 8 bit during the transmission and then decompressed again in the display to 13-14 bit.

A dimmable LED controller must be useable not only for light dimmers, but also for LED ropes and low-resolution LED walls on for example buildings. Therefore, it must also be able to display standard 3×8-bit RGB video data compressed/coded according to ITU Recommendation BT.709, that is:

$R'_{709} = 4.5R$ $\qquad R \leq 0.018$

$R'_{709} = 1.099 \times R^{0.45} - 0.099$ $\qquad R > 0.018$

In fact, ITU Recommendation BT.709 uses "studio-swing" levels where reference black is defined as 8-bit interface code 16 (10H) and reference white is defined as 8-bit interface code 235. Interface codes 0 and 255 are used for synchronization, and are prohibited from video data. Eight-bit codes between 1 and 15 provide footroom, and may be used to accommodate transient signal content such as filter undershoots. Eight-bit interface codes 236 through 254 provide headroom, and can be used to accommodate transient signal content such as filter overshoots and specular highlights. Max-i does **not** use "Studio-swing" levels, but regards 255 as maximum level and does not cut off the display below 16. However, since values below 16 are seldom in video material, a relaxed accuracy is allowed in that range. This is especially important for the practical implementation of bin correction (explained later).

It is also desirable (but not a requirement) that a Max-i controller is able to show JPEG coded images (internet pictures) compressed/coded according to the sRGB (IEC 61966-2-1) standard, which has the same RGB primaries and white point as BT.709 and therefore is able to use the same display:

$R'_{sRGB} = 12.92R$ $\qquad R \leq 0.00313$

$R'_{sRGB} = 1.055 \times R^{0.4167} - 0.055$ $\qquad R > 0.00313$

To be able to do all this, the controller shall not necessary implement the inverse of these functions. This depends on the viewing conditions. When images like TV pictures are shown in dim conditions, the eyes sensitivity is increased due to the environment (pupils get bigger). This increases the intensity of all parts of the picture equally - for example a factor 2, but because of the logarithmic behavior of the eyes, this is most visible in the dark areas, which appear too bright. The result is that the contrast seems to be reduced. For this reason, it is necessary to compensate by increasing the gamma of the entire system, which is called the system gamma or viewing gamma. It is equal to the camera gamma (video coding) multiplied by the display gamma. The ideal viewing gamma depends on the ambient luminance level as shown below:

Ambient Luminance	Viewing gamma	Application
≈0 lux	1.5	Projected slides
15 lux = very dim, street lighting	1.25	TV according to ITU recommendation BT.709
64 lux = dim, family living room	1.1	Computer monitors according to sRGB
300 lux = office and shops	1	Computer monitors and LED walls in daylight

Fig. C.7

In practice, the problem with the viewing condition is usually solved on the encoding side so that the same display/monitor with a standard gamma of approximately 2.44 can be used in all viewing conditions. Therefore, BT.709 is not coded close to a gamma of 1/2.44, but close to 1/(2.44/1.25) = 0.5 and sRGB is coded close to 1/(2.44/1.1) = 0.45.

Note that as the displays gets bigger and therefore occupy a bigger part of the field of vision, the eyes sensitivity is adjusted according to the picture instead of the environment, so the ideal viewing gamma is reduced. If the display occupies the entire field of vision, the viewing gamma should be 1.

In summary, the Max-i controller should be able to do at least the following decoding's:

1. The inverse eye sensitivity curve for use in light dimmers including stage lamps, that is:

   $R_{eye} = R' / 9.033$ $\qquad R' \leq 0.08$

   $R_{eye} = ((R' + 0.16) / 1.16)^3$ $\qquad R' > 0.08$

   where R' is the 8-bit input signal divided by 255, that is, normalized to the range 0 – 1, and R_{eye} is the output in the range 0 – 1, which must then be multiplied with the maximum level (8192 for 13 bits).

2. A standard display gamma of approximately 2.44. This gamma is very close to the gamma of a cathode ray tube (CRT), which is approximately 2.4 due to the nonlinear relationship between the cathode to grid voltage and the beam current. If a CRT is used, the expansion is therefore done inherently, which was very fortunate for simple analog TV's. For today's flat screen displays like LED and LCD, it is necessary to make the expansion and compensate for any nonlinearities by means of a look-up table, multipliers and/or other circuitry.

3. The inverse of BT.709 coded data raised to a power (viewing gamma) of 1.25 for use in the intended dim conditions such as concerts (video walls etc.), that is:

   $R_{709} = R' / 4.5$ $\quad R' \le 0.081$

   $R_{709} = ((R' + 0.099) / 1.099)^{2.22}$ $\quad R' > 0.081$

   $R_{out} = R_{709}{}^{1.25}$

4. The inverse of BT.709 coded data raised to a power of 1.15 for use in more lighted conditions than point 3 - for example evening lighting, that is:

   $R_{out} = R_{709}{}^{1.15}$

   This mode is called Gamma2.23.

5. The inverse of BT.709 coded data (raised to a power of 1.00) for use in daylight conditions such as sport events. This mode is called Gamma2.00.

6. The inverse of sRGB coded data raised to a power of 1.1, that is:

   $R_{sRGB} = R' / 12.92$ $\quad R' \le 0.04$

   $R_{sRGB} = ((R' + 0.055) / 1.055)^{2.4}$ $\quad R' > 0.04$

   $R_{out} = R_{sRGB}{}^{1.1}$

All these points may seem overwhelming for a Max-i controller where everything shall be implemented in simple hardware, but point 4, 5 and 6 are only recommendations, which are not necessary to fulfill or to fulfill with a high accuracy, and point 1, 2, 3 and 6 are so close to each other that the same decoding called Gamma2.44 can be used as shown below:

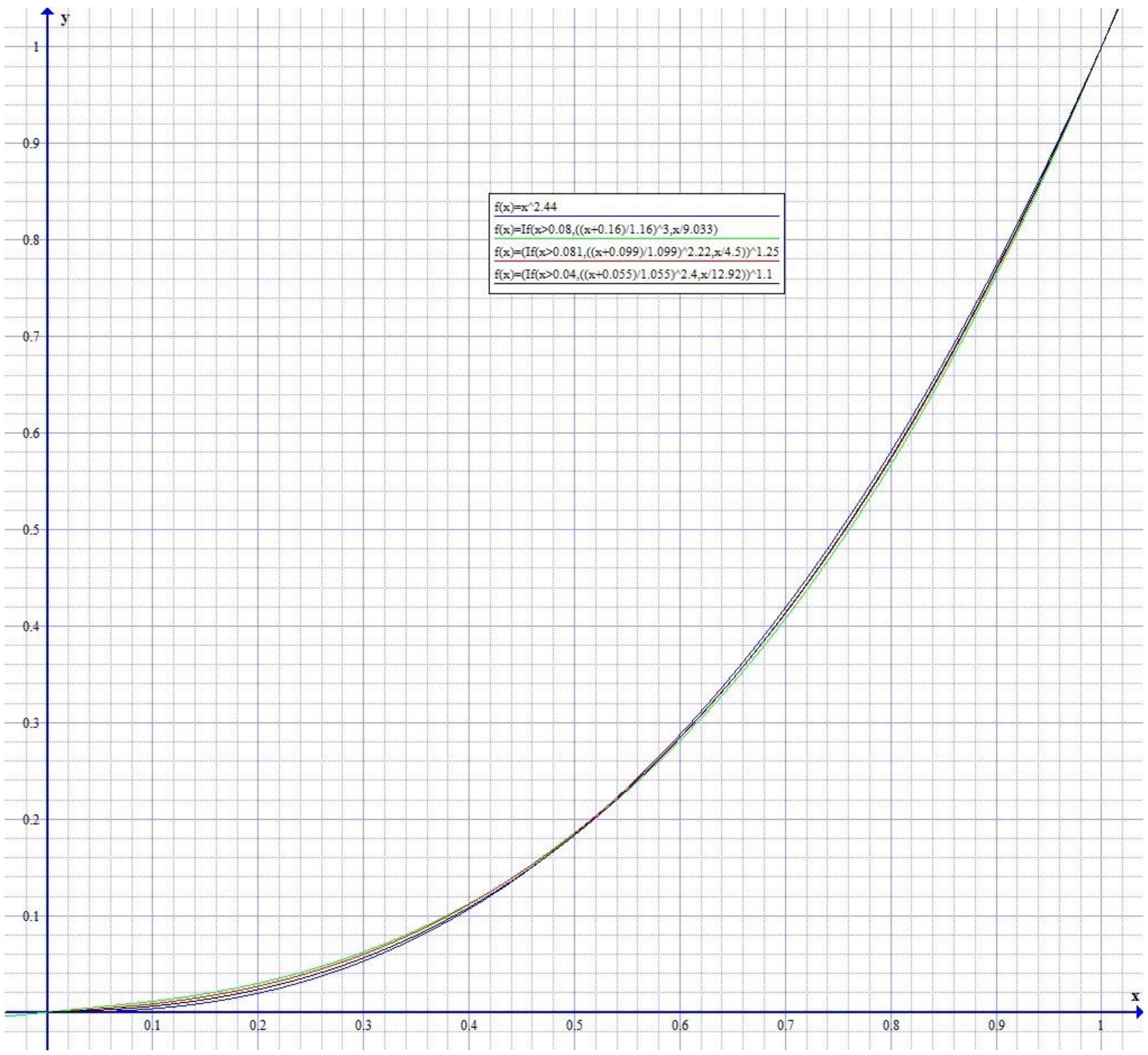


Fig. C.8

Because a standard must be based on only one definition, the inverse eye curve is chosen for Max-i for three reasons:

- It does not depend on a viewing gamma.
- It is mathematical simple (no extra gamma correction).
- The other curves are slightly too low in the low intensity range because a constant gamma or viewing gamma correction is not ideal in that range.

The figure below shows the ideal eye curve (green), the Innovatic implementation of a gamma of 2.44 (red), which is based on simple multipliers rather than big look-up tables, and the error in percent (magenta) compared to the eye curve. As can be seen, the accuracy is below ±1.04%, which corresponds to the sensitivity of the eye.

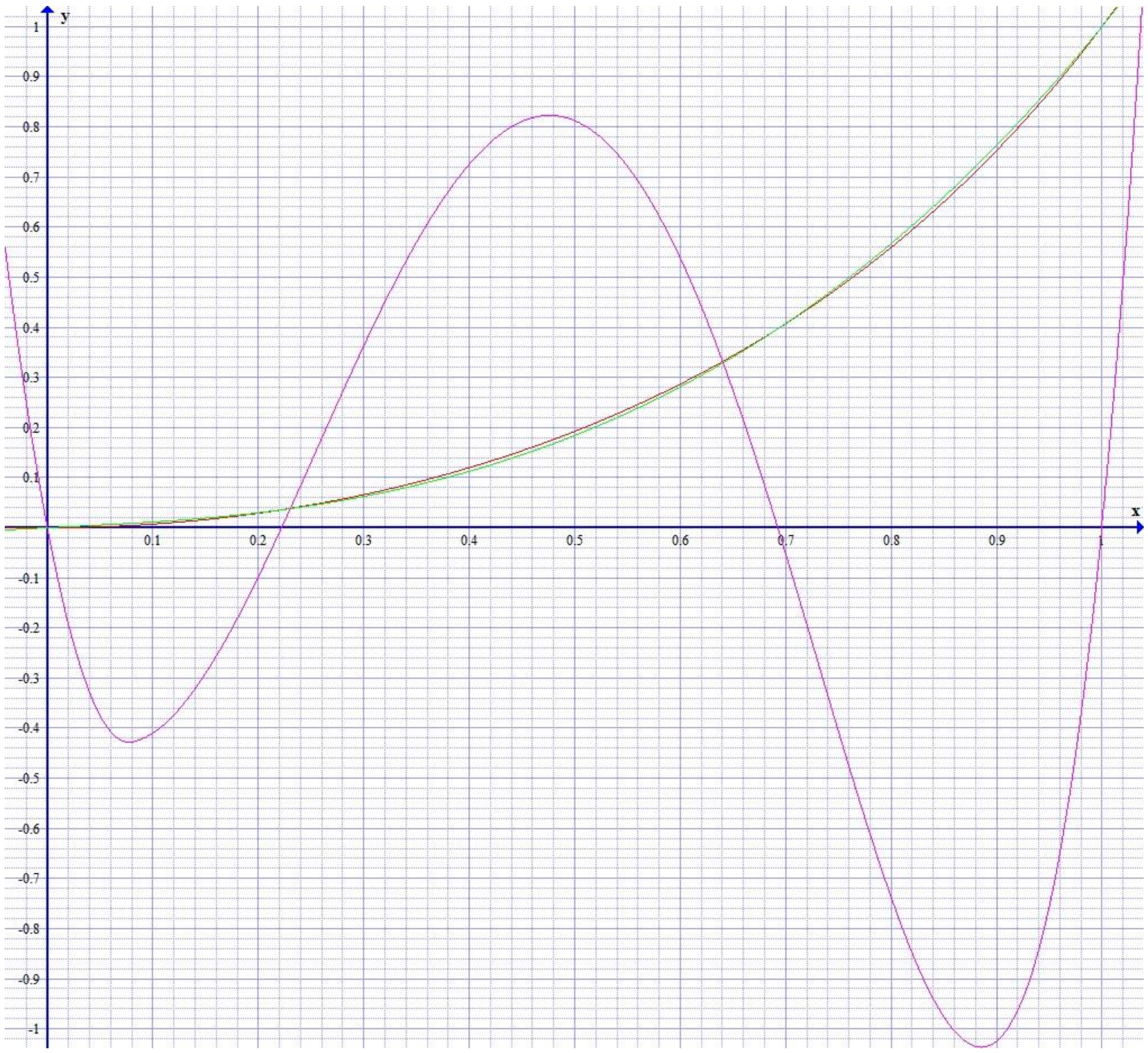


Fig. C.9

C.5 Kruithof Curves

While cold white LED's may look pleasant at high levels, this is not the case at low levels. This is illustrated with the Kruithof curves show below (the D65 point represents Northern daylight):

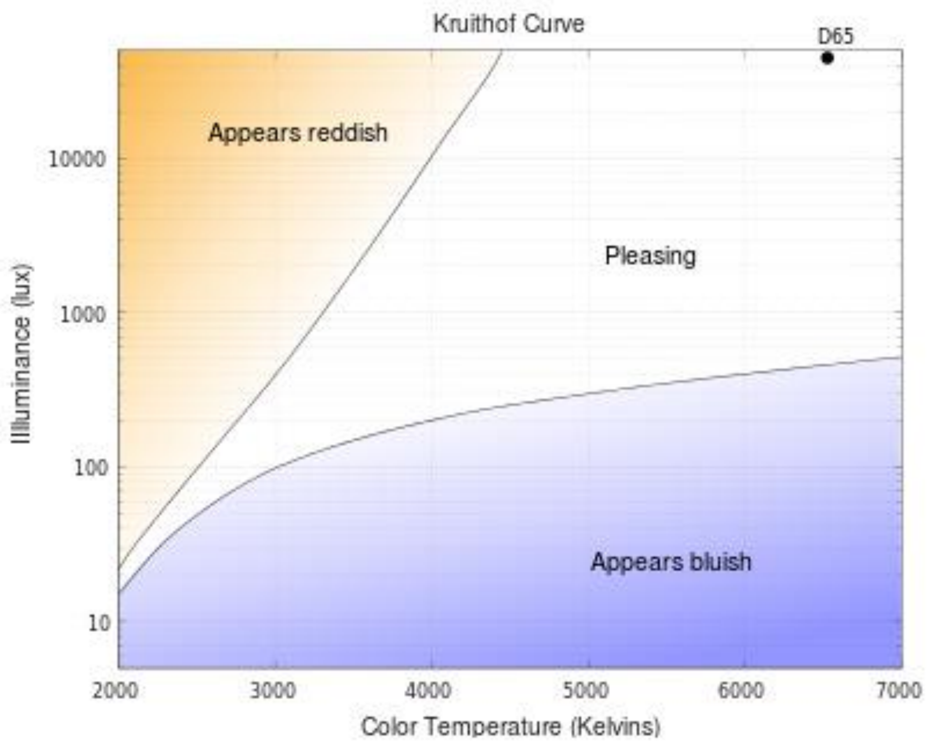


Fig. C.10

Kruithof's findings are directly related to human adaptation to changes in illumination. As illuminance decreases, human sensitivity to blue light increases. This is known as the Purkinje effect. The human visual system switches from photopic (cone-dominated) vision to scotopic (rod-dominated) vision when luminance levels decrease. Rods have a very high spectral sensitivity to blue energy, whereas cones have varying spectral sensitivities to reds, greens and blues. Since the dominating photoreceptor in scotopic vision is most sensitive to blue, human sensitivity to blue light is therefore increased. Because of this, intense sources of higher (bluer) color temperatures **including a blue stimulation peak in phosphor converted LED's** are all generally considered to be displeasing at low luminance levels. Street lamps usually only generate in the order of 15 lux on the street and should therefore have a color temperature of approximately 2000 K. Nevertheless, 4000 K is quite common, which is fairly unpleasant, but it increases the electrical efficiency.

A 100 W incandescent lamp, which may be used as a reference, follows the Kruithof curves fairly well as it typical has the following color temperature as it is dimmed:

Light Output	Color Temperature CCT
100%	2800 K
50%	2600 K
20%	2400 K
10%	2200 K
1%	2000 K

Fig. C.11

For this reason, some change in color temperature as a lamp is dimmed may be very desirable. This is usually referred to as "dim to warm". In Max-i, this is implemented by means of bright white LED's with a color temperature of approximately 3000 – 4000 K connected to the white output with a gamma of 2.44 and amber and/or **very** warm LED's connected to the white square root channel, which has a gamma of 1.56 in this case as shown below:

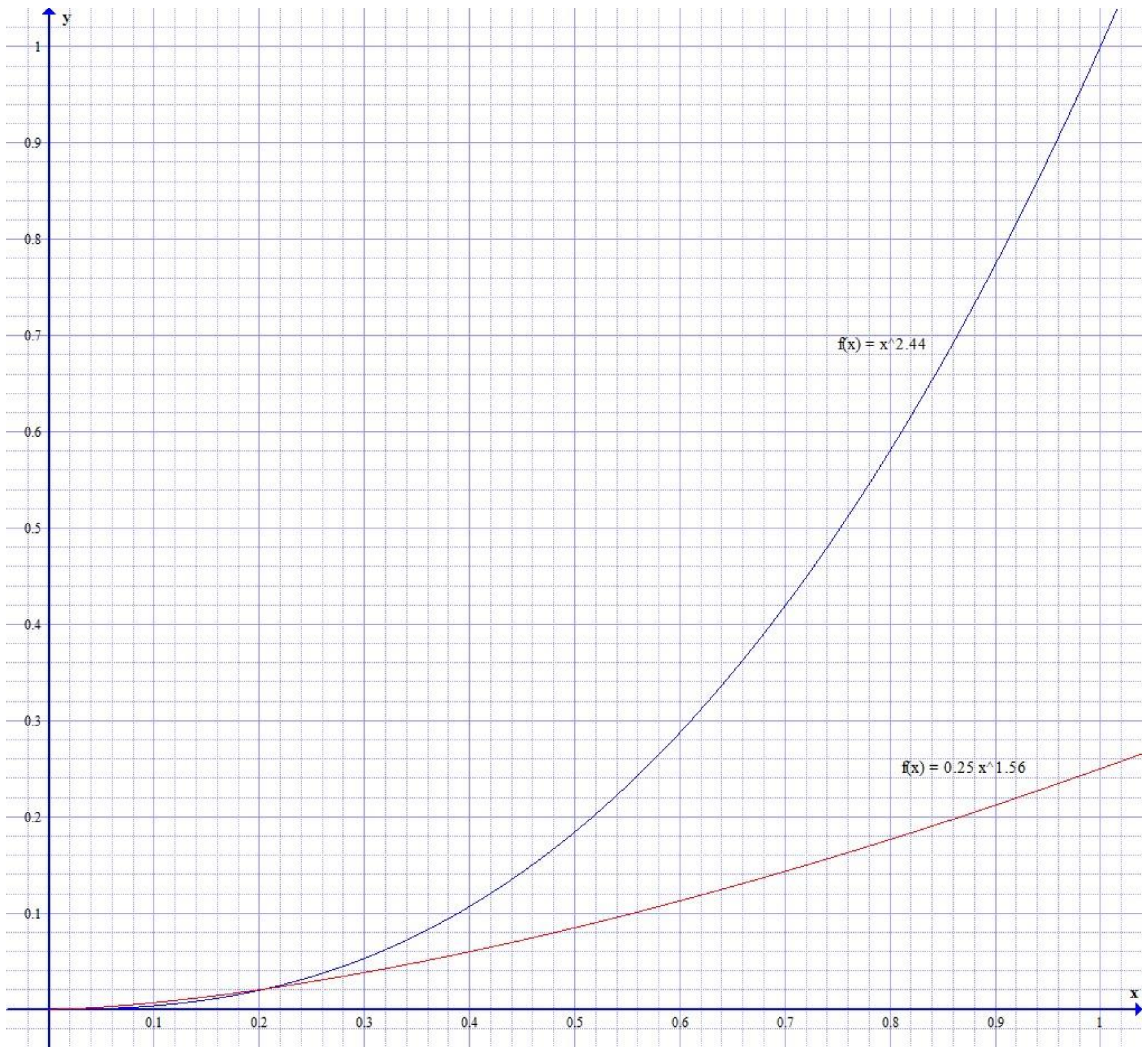


Fig. C.12

The blue curve is the dimming curve of the bright white LED’s (gamma = 2.44) and the red curve is for the amber or very warm LED’s (gamma = 1.56). Because the warm channel only needs to have approximately ¼ the maximum current of the bright white channel, this solution is very economical as fairly small LED’s may be used.

C.6 LED Ropes and Ladders

The maximum number of devices on one bus is primary a function of the load capacitance, which reduces the characteristic impedance (Z_0 = √L/C) and delays the signal (t = √LC). A typical 4-wire line has a capacitance of approximately 110 pF/m and a characteristic impedance of approximately 48 Ω. For a minimum line impedance of 35 Ω, the maximum equally distributed load capacitance must not exceed 100 pF/m, so with a typical capacitance of 25 pF, this limits the minimum distance between devices to 25 cm. However, with other cable types and/or devices with lower capacitance, a distance down to approximately 10 cm may be possible corresponding to 10 devices/m. The table below shows the maximum number of devices for incident wave switching and a spacing of 10 cm. The number is limited by the leakage current to approximately 1000 devices and because of this, the capacitive load per meter is reduced as the line length is increased.

T [µs]	**Network Length** [m @ max load]	Devices	**RGB Refresh Times** [ms]
24	3800	1000	3456
12	1800	1000	1728
6	850	1000	864
3	375	1000	432
1.5	156	1000	216
0.75	78 94 98	780 368 306	81.2 40.0 33.33
0.375	39 39 39 39 41	390 368 390 368 306	20.3 20.0 33.33 20.0 16.67
0.1875	19	190	5.29
0.0938	9.7	97	1.32

Fig. C.13

The number of devices in blue are limited by either the maximum capacitance per meter or the leakage current. The refresh time is the average time it takes to update N 3-byte RGB pixel in each device ((262 + (N × 144))T).

In case a LED rope and ladders is used to show video signals, an update rate of 16.667 ms (60 Hz), 20 ms (50 Hz), 33.33 ms (30 Hz) or 40 ms (25 Hz) is needed so the theoretical maximum number of pixels is 390

In practice, the devices on a LED ladder will usually be updated by means of two telegrams - one for each half. In case of a vertical ladder, this may be very convenient to implement an interlaced system. Motions usually take place in horizontal direction. By updating the two lines with a time difference equal to the half frame rate, the display converts flickering into a more pleasant motion blur in horizontal direction. This is especial useful in case of 25 Hz or 30 Hz video if interlace scanning is not used (see below).

Note that since Max-i uses a modulation where 0 and 1 do not have the same length, the refresh time is only an average time, but due to the scrambling it will not change much. However, some safety margin is necessary for a reliable operation so no more than 1024 data bytes are recommended per telegram corresponding to 341 pixels. 1024 bytes is the double of what DXM 512 can handle.

C.7 Bin Correction

LED's have fairly big production tolerances of up to ±25%, which may seriously change the white point and create visible differences between pixels with the same color. To avoid this, LED's are usually bin selected, but this is fairly expensive and therefore violates one of the basic ideas of Max-i – lowest possible cost. Therefore, a Max-i controller shall be able to multiply the output with a correction value between approximately 0.5 and 1.00. This adjustment shall be linear (independent of gamma correction) and shall have a resolution of 6 bits above the BT.709 black level of 16 (10H), so that the adjustment accuracy between 16 and 255 is ±0.8%. Due to the logarithmic behavior of the eye, a change from 50% to 100% current is only sensed as a 20% variation in light intensity, so this step size is in practice invisible. Note that the full scale value shall be 1.00, that is, 64/64 – not 63/64. This may be achieved by adding 1 to a correction factor between 32 and 63 (default) so that multipliers between 33/64 and 1 are possible.

C.8 Interlace Scanning

Interlace scanning has a bad reputation of being flickering, but this is only because an interlaced display is usually compared to a non-interlaced display with the same number of pictures per second and therefore the double bandwidth, and this is of course not a fair comparison. If a 25-Hz non-interlaced display is compared to a 50-Hz interlaced display with the same bandwidth, the interlaced display will be **much** less flickering, which is the reason why this method was used on analog TV’s.

Interlace works very well on picture tubes or plasma displays, where each dot only lights up for a very short time, but on video walls and most LCD screens where the dots have a steady light until they are updated again, the method creates motion blur as two pictures (odd and even), which may not be part of the same picture, are always visible at the same time. Max-i can solve this problem by means of a short and therefore fast telegram, which can turn a line off before it is updated again. The repeated sequence is:

- Transmit a (long) update telegram to the **even** lines and update the dots when the signature has been verified at the end of the telegram (all dots are synchronized and delayed equally).
- Switch off the **odd** lines by means of a short 5 (or 7) bytes telegram, so that only the **even** lines, which have just been updated, are visible.
- Update the **odd** lines.
- Switch off the **even** lines so that only the **odd** lines are now visible.

Since the off-telegrams are very short, it is in practice only the filter time (minimum 5 ms), which determines how fast the blend-over between odd and even lines is.

A disadvantage of interlace scanning is that to keep the luminance constant, twice as much current is needed for the half time in each LED. This may stress the LED’s so good cooling and a not too heavy load is recommended. To keep the power consumption of each bus reasonable constant, it is also recommended - but not a requirement - to use alternating even and odd dots on the same bus.

Annex D. Voltage and Current

D.1 Choice of Nominal Supply Voltage

Max-i was originally designed for 12 – 14 V operation because of many reasons:

- It is an old, well established standard, which fits perfectly with automotive applications and all the equipment designed for that.
- 12 V is perfect for Ex applications as it is impossible to create an arc with any noticeable power. This also makes 12 – 14 V very safe (from an arc point of view) for applications where current levels over 20 A are possible like a DC net in the house of the future.
- Only one voltage range is needed for all applications including Ex.
- The power dissipation during communication is shared equally between transmitter and receiver (clamp) and it is low enough for a very small and cheap IC package.
- The transmitter current is reasonable (<600 mA).
- The voltage swing is low enough for a reasonable rise time (12 V in 60 ns = 200 V/µS).
- The common mode voltage due to the communication is below 3 Vrms so it should be possible to connect CE marked equipment to a balanced line without problems.

However, after Max-i has been expanded with advanced LED lighting for stage light, architectural lighting, traffic light and home application and after the introduction of the USB Power Delivery standard and the Quick Charge technology, there has been an increasing demand for a higher voltage (higher power).

- At 12 V, it may be very difficult to fulfill a requirement for a maximum voltage drop without inexpedient thick or short cables.
- 12 V is a little too low for LED lighting. A white (or blue) LED has a typical voltage drop of 3.2 – 4 V and very often four serial connected LED's are mounted in a square in one package so with additional 1 V for a current generator, minimum 17 V is needed unless expensive switch mode technology is used.
- 12 V is so low that it may be difficult to blow a fuse. For example, at least 135% overcurrent may be needed to blow a fast acting fuse within 10 minutes, but at 12 V and 20 A, this requires that the loop resistance must be less than 0.44 Ω corresponding to only 9 m 0.75 mm^2 cable. If this is not the case, the fuse may not blow and the cables may instead be overheated, which may cause fire.
- 12 V is not enough for charging laptop computers and driving larger computer peripherals such as monitors. Because of this and the fact that 18.5 – 20 V is the de facto standard for most laptop chargers, almost all new charging standards have settled at 20 V like:
    - USB Power Delivery – USB PD up to version 3.0 (100 W, 5A – fulfils IEC 60950-1).
    - Quick Charge from Qualcomm® (100 W, 5 A – fulfils IEC 60950-1).
    - Motorola TurboPower® (125 W, 6.25 A).
    - SUPERVOOC Endurance Edition (150 W, 7.5A).
    - Infinix (160 W, 8 A).
    - Xiaomi HyperCharge (200 W, 10 A)
    - Realme (240 W, 12 A).
    - Lenovo Slim-tip (
        - 1.75 A = 36 W,
        - 2.25 A = 45 W,
        - 3.25 A = 65 W,
        - 4.50 A = 90 W,

- 6.75 A = 135 W,
- 8.50 A = 170 W,
- 11.5 A = 230 W).

USB Power Delivery and Quick charge use different voltage and current levels as shown below, but this requires complicated electronics and an extremely complicated protocol to negotiate both voltage and current:

USB Power Delivery	**Qualcomm® Quick Charge**	**Voltage V**	**Current A**	**Power W**
3.0, Profile 1	-	5	2	10
-	2.0, Class A + B	5	3	15
-	2.0, Class A + B	9	3	27
-	2.0, Class A + B	12	3	36
-	3.0, Class A	3.6 – 12	3	≤36
3.0, Profile 2	-	12	1.5	18
3.0, Profile 3	-	12	3	36
3.0, Profile 4	-	20	3	60
-	2.0, Class B	20	3	60
-	3.0, Class B	3.6 – 20, 0.2 V steps	3	≤60
-	3+	3.6 – 20, 20 mV steps	3	≤60
3.0, Profile 5	-	20	5	100
.	4.0, 4+	3 – 21, 20 mV steps	5	100
3.1, Profile 5	-	28	5	140
3.1, Profile 6	-	36	5	180
3,1, Profile 7	-	48	5	240

Fig. D.1

Only USB-PD uses higher voltage levels than 20 V (28 V, 36 V and 48 V), but this is the only way to stay in the competition for higher power with the other standards as no more than 5 A can be handled by the tiny USB-C connector.

All USB Power Delivery profiles also include the voltage levels of lower profiles. For example, a device, which supports profile 4, is able to supply 3 A on both 12 V and 20 V and 2 A on 5 V (no 5 V profile has a higher current).

20 V or higher could therefore be a better bid than 14 V for Max-i, but there are two major problems – power dissipation and arc risk. The average power dissipation can be kept at an acceptable 0.7 W as described in Layer 1, but above 18 V, the arc risk becomes a very big problem for DC systems with high possible current levels like a DC net in the house of the future.

D.1.1 Power Dissipation during Communication

At higher voltage levels, the power dissipation in the transmitter and/or the clamp may be a big problem as it basically depends on the voltage in the power of two, and it may be necessary with a power package and/or external devices. At a maximum voltage of 25.2 V, it is however still possible to keep the average power loss below 0.75 W with the method described in Layer 1, but at higher voltage levels, the power dissipation simply becomes too high.

D.1.2 Arc Hazard

Above 18 V, the arc hazard becomes a very big problem for DC systems with high possible current levels like a DC net in the house of the future.

There are two types or arcs – serial arcs and parallel arcs. Serial arcs occur due to bad connection or failure in for example bypass diodes in PV arrays. Parallel arcs are arcs directly between the power supply rails, which is the most serious situation, but also the most uncommon. Serial arcs are limited by the resistance in the circuit. As long as all devices are OK, serial arcs are not possible at 20 V unless the current can exceed 20 A since the resistance cannot be lower than 1 Ω, and even at 28 V, they are not a big problem as shown later (1.4 Ω load line), but since semiconductors tends to short circuit in case of an overload, the arc may quickly be converted to a parallel arc.

The arc voltage consists of three parts – cathode region, column region and anode region.

D.1.2.1 Cathode Region

The role of the cathode, and the surrounding cathode region, is to emit the current conducting electrons into the arc column. Some electrode materials have such a high boiling temperature that significant thermionic emission of electrons starts already well below the evaporation temperature. Such materials are e.g. carbon, tungsten and molybdenum. Other electrode materials have lower evaporation temperature. In this case, the major emission mechanism will be field emission, where electrons will be emitted due to the high electric field strength close to the surface. Copper is a typical example of such a material. Very close to the cathode (approximately 0.1 µm) there will be an accumulation of positive ions, arriving from the arc column. Due to this space charge, there will be high electric field strength close to the cathode surface (the cathode drop). This high field strength is essential for efficient (field) emission of electrons into the arc.

The cathode drop primary depends on the electrode material as shown below.

Material	Cathode drop V	Minimum anode current A
Antimony, Sb	10.5	-
Zinc, Zn	10.5	0.1
Silver, Ag	12	0.4
Copper, Cu	13	0.43
Bronze	13.5	0.31
Tin, Sn	13.5	-
Aluminum, Al	14	-
Nickel, Ni	14	0.4
Gold, Au	15	-
Palladium, Pd	15	-
Steel, Fe	15	0.5
Platinum, Pt	17.5	-
Tungsten/wolfram, W	Hot: >6.5	-
Carbon/graphite, C	Cold: 20 Hot: >7.5	0.03

Fig. D.2

Materials like tungsten and carbon reduce the cathode drop considerably when the cathode is heated due to thermionic emission of electrons. The low cathode drop of tungsten in hot condition makes it possible to TIG weld down to approximately 15 V if the tungsten electrode is negative (cathode) and serial coupled with a coil (choke), which can stabilize the arc current and ease the ignition. As the cathode drop of copper is 6.5 V higher, it is reasonable to assume that the minimum voltage for arc welding with copper electrodes will be in the order of 21.5 V. Note that melting (due to heating caused by the current) is not the same as welding.

Hertha Marks Ayrton's "Electric Arc" study from 1902 showed that the cathode drop in a **carbon** arc is:

$V_c = 7.6 + (13.6 / I_{arc})$

At 1.1 A, the cathode drop becomes 20 V, which is the same as for field emission (cold), and for very small current levels such as the minimum arc current of only 30 mA, the cathode drop approach infinite, so the formula cannot be valid below approximately 1 A.

D.1.2.2 Column Region

The column voltage primary depends on the arc length. Studies by TüvReinland show that it is in the order of 4.5 V/mm for arcs in air (se figure E.6 below).

D.1.2.3 Anode Region

The anode mainly serves as a collector of electrons, arriving from the cathode. The electrons will arrive at high speed and deliver the energy to the anode. The anode surface will therefore be kept at a high temperature. Close to the anode there will be a lack of positive ions, since they tend to drift away. The surplus of electrons leads to electric field strength of approximately 10 V/mm in a layer up to 0.5 mm over the anode surface (the anode drop). The maximum anode drop is therefore in the order of 5 V for metals. For carbon, it is considerably higher. Hertha Marks Ayrton found that it is:

$V_a = 31.28 + ((9 + 3.1g) / I_{arc})$ where g is the arc length in mm.

For high current values and short arc lengths, the minimum voltage for a burning carbon arc is therefore Vc + Va = 7.6 V + 31.3 V ≈ 39 V, and in cold condition, 20 V (cathode drop) is necessary to ignite the arc due to field emission. If an arc is generated internal in an IC or on a PCB, the epoxy may be carbonised as shown below, but if the voltage is below 20 V, there ought to be very little arc risk and, surprisingly, there is little conductance across the burnt area. This could indicate that 20 V is a good choice, but it should be noted that no arc studies have been done on carbonised epoxy.

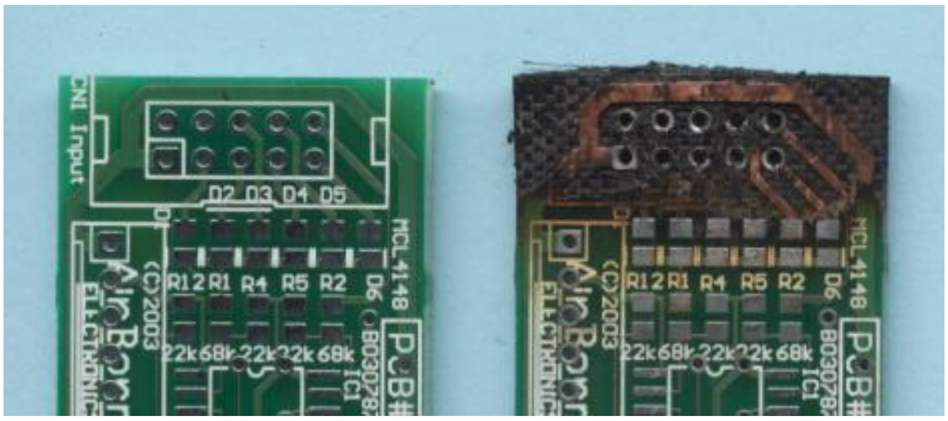


Fig. D.3

In practice all this means that arcs between copper electrodes are almost impossible below 13 V (cathode drop). From 13 V to approximately 18 V (sum of cathode and anode drop), the current may rise up to a fairly high level, but the maximum arc length remains very short as there is no column voltage, which makes it very difficult to sustain an arc. Either the two electrodes melt together, which shall cause a fuse to blow or a breaker to pop, or the melting creates gap big enough to extinguish the arc. Above 18 V, the sum of the cathode drop and the anode drop is almost constant and the column voltage and with that the maximum arc length begins to raise almost linear with the voltage.

Unfortunately, there are not many good studies of short, low voltage arcs and many of these give a little different result. In the following, some of the best are listed and the results are compared.

D.1.2.4 TüvReinland "Failure Mechanismen of Contact Faults in the DC- Circuit of the PV Arrays"

This is by far one of the best studies. TüvReinland has published three very important figures as shown in the following:

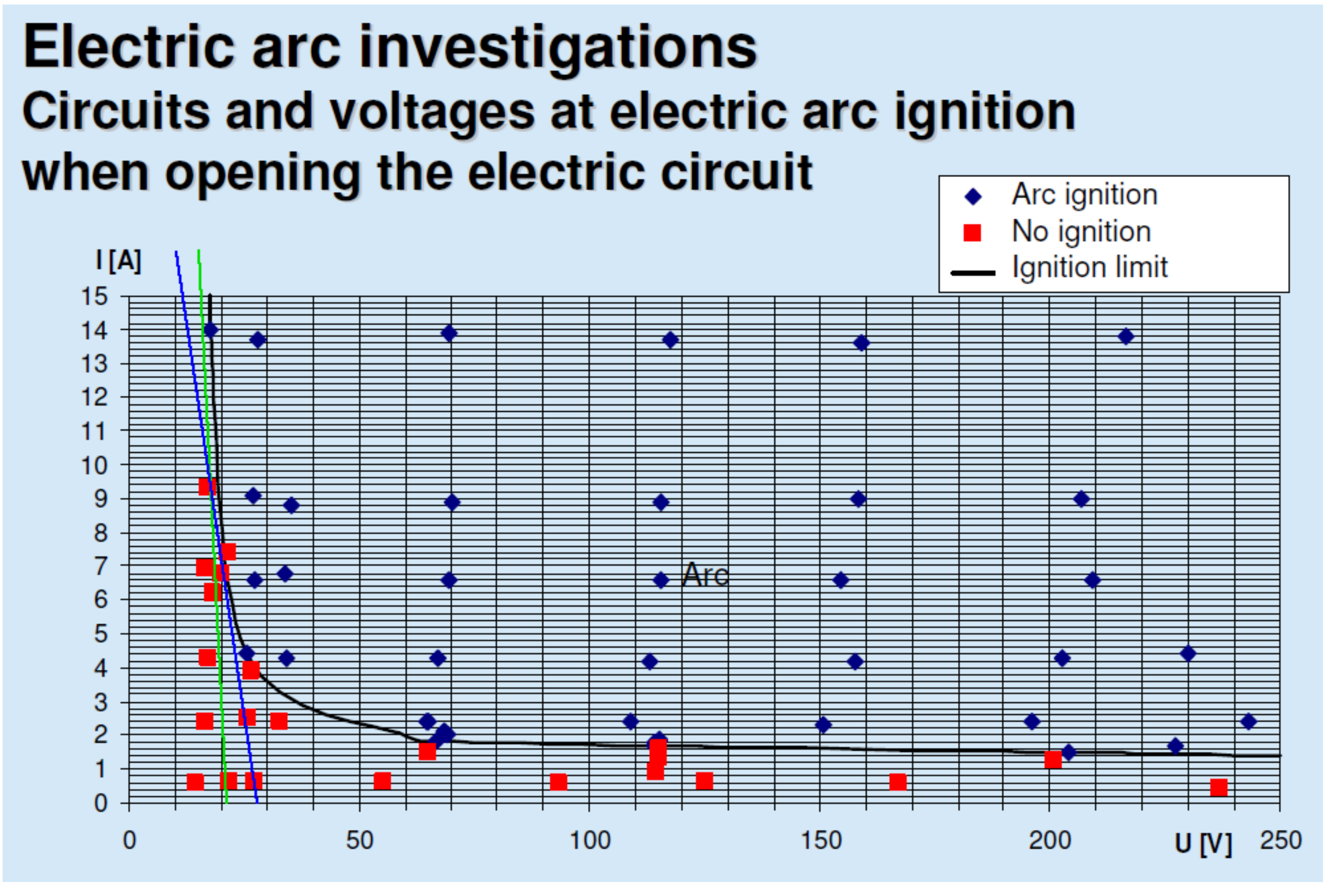
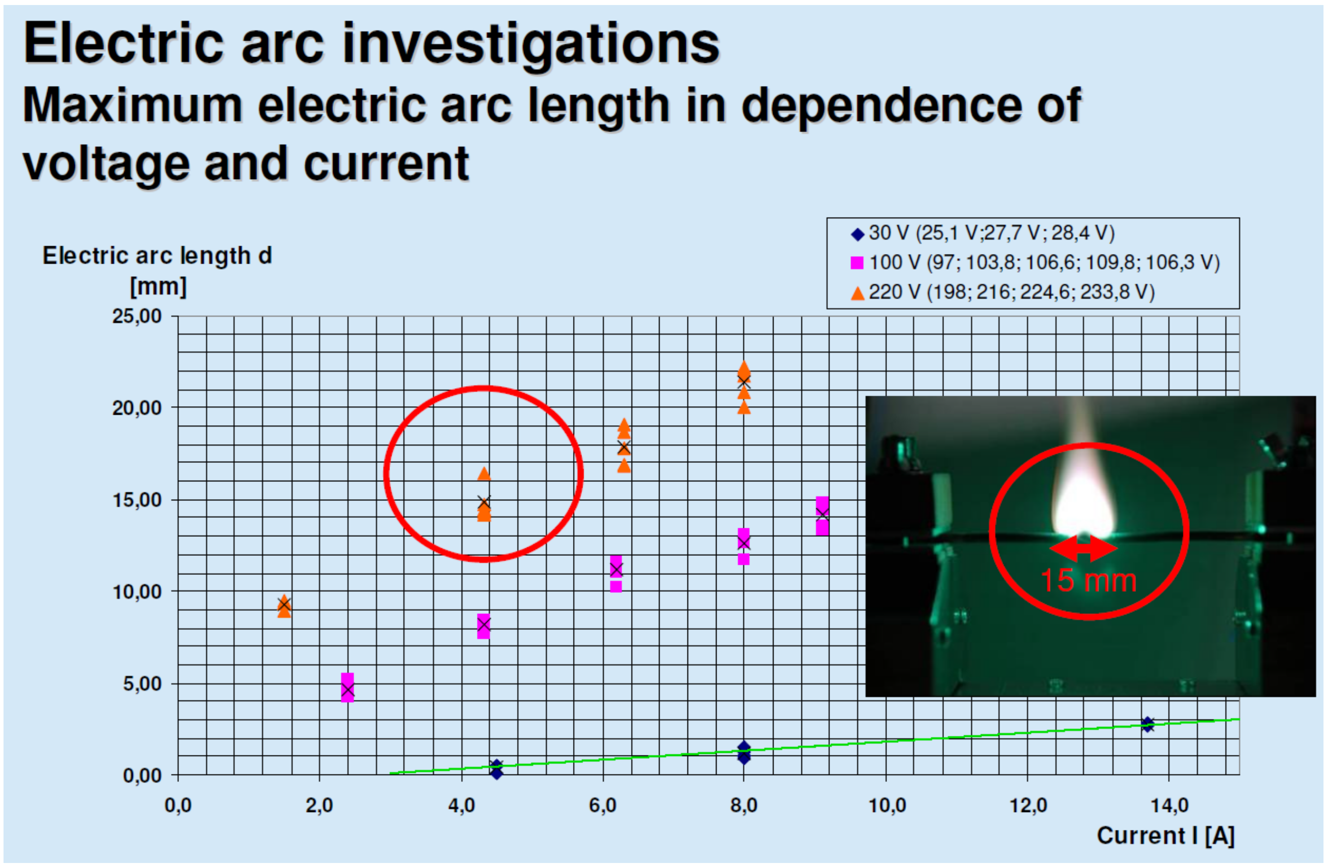


Fig. D.4

The curve fits very well with other measurements made on arcs in bypass diodes on PV arrays, which show a typical arc voltage of 20-22 V at 8 A.

The green and blue lines are load lines drawn by Innovatic. If these lines cross the ignition limit, an arc is generated. At 21 V, which is the maximum for mains operation and the no-charge voltage of a 20-V battery, no arcs should be generated if the serial resistance is bigger than (21 V - 15 V) / 15A = 0.4 Ω (green load line) corresponding to 8 m, 0.75 mm^2 cable, but practical tests at Innovatic have shown that **very** thin arcs with low light intensity are in fact possible at this level, but the arc power is so low that it is not expected to cause any big fire risk. At 28 V, the resistance needs to be bigger than (28 V - 12 V) / 15A = 1.1 Ω to avoid an arc (blue load line), but in case of high current applications, this may be way above the resistance in the cables and close to the limit where a serial arc is possible (1.4 Ω at 20 A).

From the figure it is obvious that the load line for a 21 V system is much more optimized than for a 28 V system. It also utilizes the higher arc voltages at lower current levels (below 10 A) much better than a 14 V system, which is inherently safe and therefore does not need any resistance (there will always be some in practice). It can also be seen that in case of a LPS where the current is limited to 5 A, no arc is generated up to the maximum voltage (25.2 V) if there is just a slight resistance in the cable, which will always be the case.



Electric arc investigations

Maximum electric arc length in dependence of voltage and current

Fig. D.5

At 28 V, the arc length may be calculated as (green line drawn by Innovatic):

Length = (I_{arc} - 3) / 4

Studies made by J. Paukert (see below) indicate that the arc length continues to increase with the current up to a maximum of approximately 10 mm.

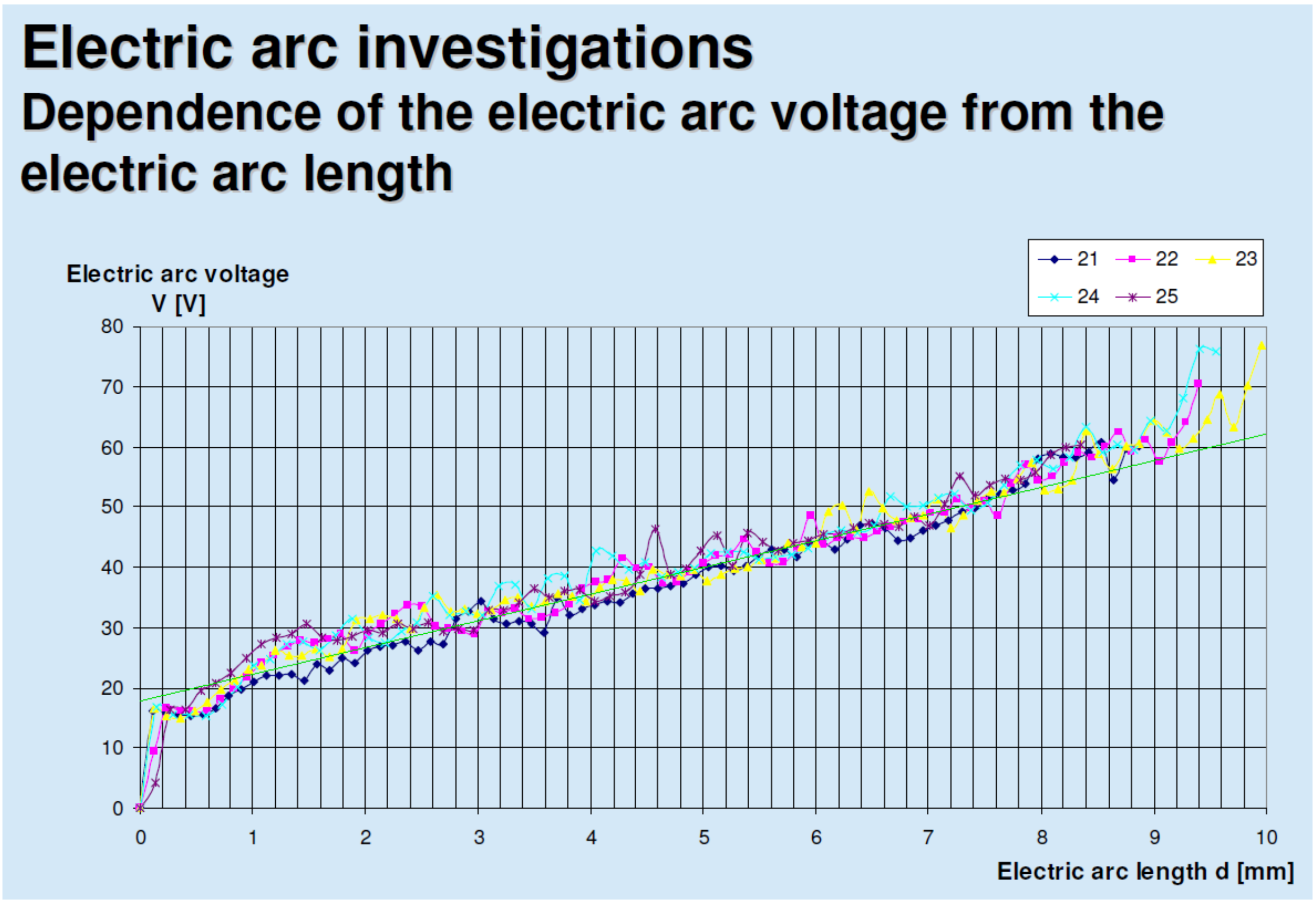


Fig. D.6

Unfortunately, it is not specified at which currents these curves are recorded. With some approximations (green line drawn by Innovatic), the arc voltage is 18 V + 4.4 V/mm, that is, 2.3 mm at 28 V. If this value is inserted in the formula from figure E.5, the current is probably in the order of 12.2 A, but since many measurements done by TüvReinland are done at approximately 13.5 A, the curves may also have been measured at approximately this level.

The curves and the formula fit very well with the measurements done by Innovatic, with show a constant arc length of approximately 0.4 – 0.5 mm at 20 V and currents from 15 A to 30 A.

Link to study:

http://www.pv-test.ch/fileadmin/user_upload/lab1/pv/Workshop_Tagungen/W_Vaassen_Lichtbogen-Burgdorf_E.pdf

D.1.2.5 A.D. Stokes and W.T. Oppenlander "Electric Arcs in Open Air" (1991)

This is also one of the most comprehensive and quoted studies about arcs between copper electrodes, but unfortunately it is based on arc lengths of 5 mm and above (up to 500 mm). Strokes and Oppenlander found that the dynamic impedance of the arc is only negative (higher currents cause lower arc voltages) up to a given transition current, but above that level, the arc voltage is increased again as shown below:

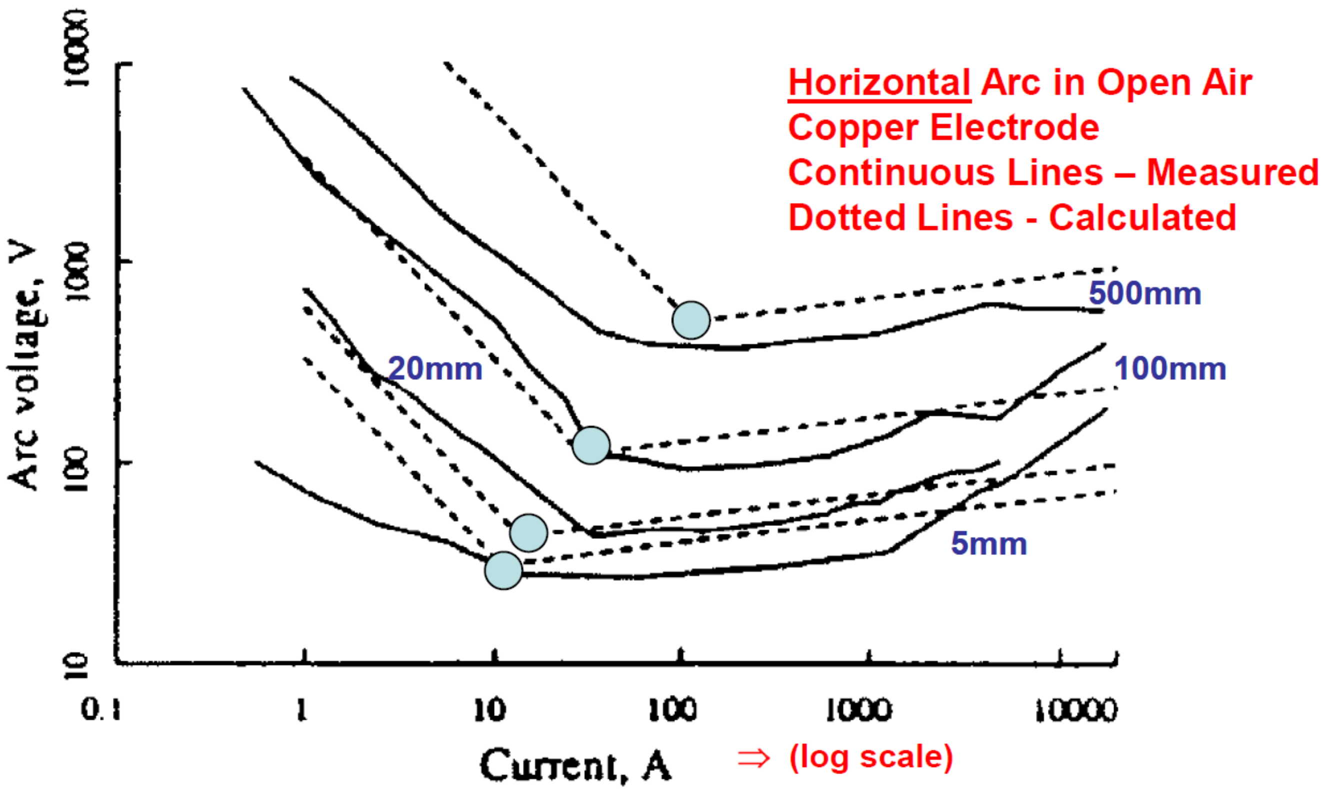


Fig. D.7

According to Stokes and Oppenlander, the transition current may be calculated as:

$I_t = 10 + 0.2\ g$, where g is the maximum arc length in mm.

At 20 V where the maximum arc length is approximately 0.5 mm, the transition current becomes 10.1 A. This fits fairly well with the measurements done by Innovatic, which show approximately the same arc length from 15 A to 30 A, but other studies like the ones from TüvReinland (figure E.5) and J. Paukert (see below) indicate a much higher transition current. For example, the maximum arc length at 28 V is in the order of 10 mm, and the transition current should therefore be approximately 12 A, but it is probably between 20 A and 40 A. $I_t = 10 + g$ is probably a better approximation up to an arc length of 100 mm.

As a rough approximation to the measurements, the arc voltage **above this minimum** may be calculated as:

$V_{arc} = (20 + 0.534\ g) \times I^{0.12}$, where I is the current and g is the gap length in mm.

However, many studies like figure D.9 and D.10 below, and in fact also figure D.7 itself, show an almost constant arc voltage above the transition current – at least up to approximately 100 A. Sometimes this range is therefore referred to as the constant voltage range.

If 10.1 A (transition current) is inserted in the second formula (the two formulas belong together), the minimum arc voltage for a gap length of 0.5 mm becomes 26.7 V, but studies performed by Innovatic and others shows that it is possible to create short arcs at much lower voltage levels at this current. At 15 A and a 3 mm gap, the arc voltage becomes 29.9 V, which is almost identical to the data from figure D.5, so the formula seems to be fairly accurate down to at least 3 mm. The fixed number in the formula (20) is the minimum cathode and anode voltage drops. This could indicate that 20 V is close to the minimum voltage where a powerful arc can be generated and therefore a good choice.

The formulas from Stokes and Oppenlander can only be regarded as fairly rough approximations, **but above a gap length of 3 mm, they becomes sufficiently accurate to exclude 28 V as a useful voltage for a high-current DC system because of the high arc risk!**

D.1.2.6 J. Paukert "The Arc Voltage and Arc Resistance of LV Fault Arcs" (1993)

This study was performed at very high current levels between 100 A and 100 kA. It has different formulas for the various gap lengths as shown below:

Gap length	Arc Voltage	Arc Voltage at 20 A
1 mm	$13.04 \times I^{0.098}$	17.5 V
5 mm	$14.13 \times I^{0.211}$	26.6 V
10 mm	$16.68 \times I^{0.163}$	27.2 V

Fig. D.8

The values at 20 A are slightly lower than the other measurements for short gap lengths, but 20 A is also a lot below the investigation range. Note that according to these formulas, the arc voltage is increased for increasing currents. This shows that there is in fact a transition current as found by Strokes and Oppenlander where the dynamic impedance becomes positive and that this point is below 100 A, which is the minimum current of this study.

D.1.2.7 W.B. Nottingham, "A New Equation for the Static Characteristic of the Normal Electric Arc" (1923)

This study was performed at a fixed arc length of 10 mm and copper electrodes. The arc voltage is:

$$V_{arc} = 27.5 + 44 / I_{arc}^{0.67}$$

If 20 A in inserted in this formula, the arc voltage becomes 33.4 V. Higher current levels give lower voltages – for example 32 V at 30 A, so the study must have been done below the transition current where the dynamic impedance is negative. According to the formulas from Strokes and Oppenlander, the voltage should be 36.3 V. This is almost the same, but the highest value ought to be the one from Nottingham since this is below the transition point.

At very high current levels, the arc voltage approaches 27.5 V at a gap length of 10 mm, which is almost identical to the 27.2 V Paukert found, so it is reasonable to assume that arcs that length are possible in a 24-28 V system, but not at 20 – 21 V. Since 27.5 V is close to the test voltage of figure D.5, it can be expected that the arc length at this voltage continues to rise with the current (Length = (I_{arc} - 3) / 4) until it approaches 10 mm, so that the transition current is above 10 × 4 + 3 = 43 A. In case of a 20 A fuse, which may be loaded with 40 A for 5 seconds before it blows, arcs up to 9-10 mm may be possible in a 24 – 28 V system, which is enough to regard this voltage as unacceptable.

D.1.2.8 Joseph Luis “Detection of Electric Arcs in 42-volt Automotive Systems”

This study shows the tremendous difference between 14 V and 42 V automotive systems.

A few years ago, everybody talked about 42 V as the new voltage for automotive application, but this study showed why this was certainly not a good idea!!! 21 V could however be an advantage in the future as it could save half of the copper, which is a limited resource, without increasing the arc risk too much (it simultaneously reduces the current to less than the half, which increases the arc voltage). 21 V is also much better suited for LED lighting as more LED’s may be connected in series, and it makes it possible to charge laptops in the car, which is a great advantage for salesmen etc.

Today, the alternator is connected to the battery through the same pole connections as the chassis and the loads. This is very foolish because in case of a bad battery connection on one of the poles, which is quite common due to the corrosive gasses, the voltage on the car may temporary raise to the open circuit voltage of the alternator, which may be over 87 V. To avoid this, a very powerful transient suppression is added in the alternator, which limits this so-called load dump transient to 40 V. A much clever way is however to use dual-pole batteries where one set is connected to the alternator and the other set to the chassis and the loads. In case of a failure, the voltage will always fall as long as there is no internal open circuit fault in the battery, so much less transient suppression is needed and the electronics don’t need to be designed for 40 V, which lowers the price considerably (chip size is proportional to voltage). Changing to this much better system may be a good time to increase the voltage to 18 – 21 V and standardize the size of the batteries.

D.1.2.9 Dr. Maribeth Mason and Dr. Genghmun Eng. Understanding Tin Plasmas: A New Approach to Tin Whisker Plasma Risk Assessment (2007)

This study was an investigation of the arc risk due to tin whiskers in space applications, but due to today’s requirement for lead-free soldering, it is also very relevant in other electronics. Various atmospheres of N_2 and air at various pressures were compared. The figure below shows an arc in air at a pressure of 1 Bar (1 atmosphere) in a 28-Vdc system. The arc voltage of 19 – 20 V at a current of 20 A is clearly visible.

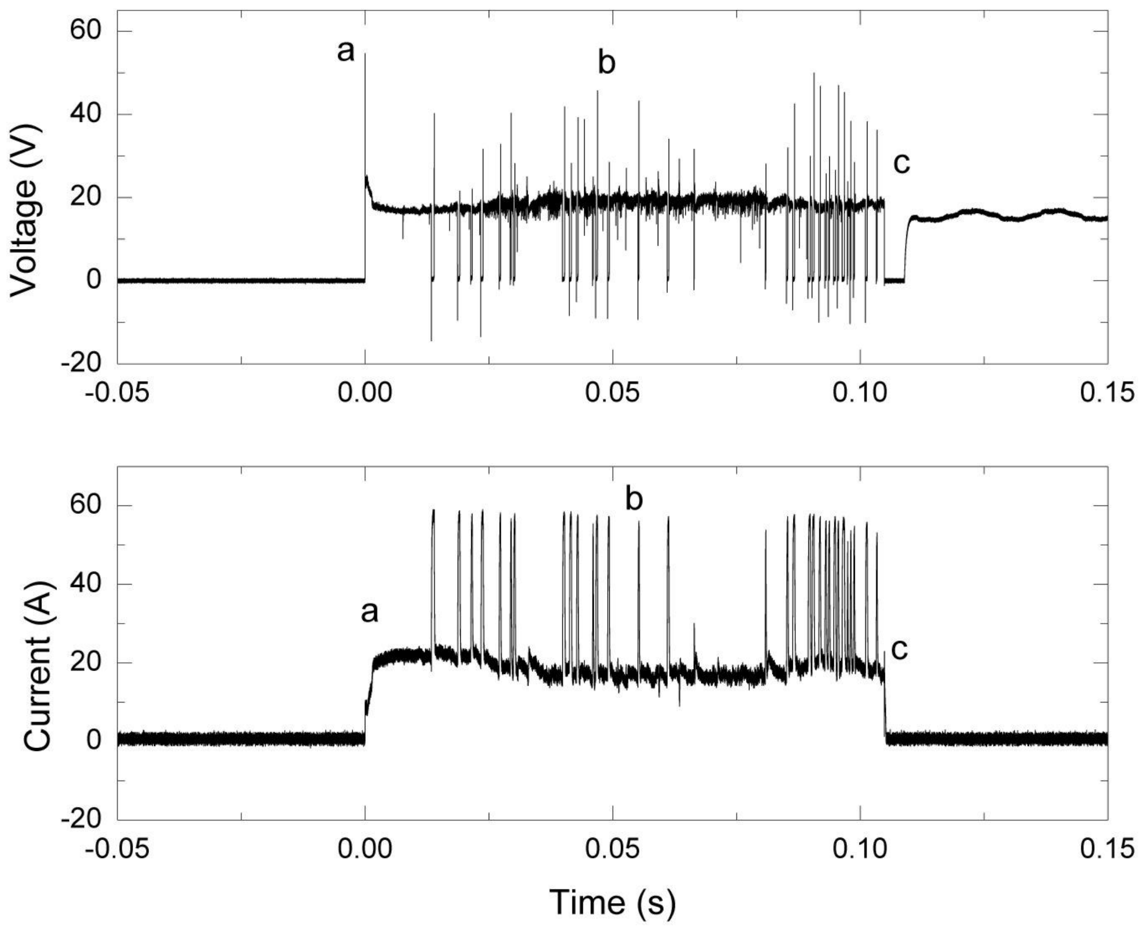


Fig. D.9

At “a”, the plasma is initiated. At “b”, tin reflow reformed connections (31 times!) carrying ≈56 A current. At “c” 105 ms after initiation, a 10 A fuse blew. If the fuse had been 20 A, as it would be the case in a 40-A system with power supply from both ends, the 160-W arc could have continued forever causing a high fire risk. In a 20-V system, such an arc is not possible.

Link to study: http://nepp.nasa.gov/whisker/reference/tech_papers/2007-Eng-tin-plasma.pdf

D.1.2.10 Studies by Innovatic

Many measurements have been made with 0 Ω, 0.1 Ω and 0.5 Ω serial resistances (to simulate cables and load line) and current levels up to 30 A.

At 12 – 14 V, arcs are possible, but they are hair thin and more a glow than a real arc so a 12 – 14 V system is considered inherently safe from an arc point of view. When the voltage is increased up to 18 V, the arcs become noticeable more powerful, but the power is still very limited and insertion of a 0.5 Ω resistance (10 m, 2 × 0.75 mm^2 cable) reduces the maximum arc length to approximately 0.1 mm, which makes it very difficult to ignite and sustain an arc no matter the current.

From 18 V to 21 V, the arc power is considerably increased, but the maximum arc length measured with flat feeler gauges after arc extinguishing remains almost constant 0.4 – 0.5 mm at 20 V from 15 A to 30 A. Note that the current is the maximum current of the power supply (current limit setting) – not a current measurement. It could be so that the arc current is less than 30 A because higher current levels are not possible at this supply voltage, but this does not matter. The purpose of the test was to see if a 20 – 21 V system is sufficiently safe up to 20 A and it seems to be. Insertion of a 0.5

Ω resistance reduces the maximum arc length at 20 V to approximately 0.3 mm, and the arc power and the tendency to electrode welding is noticeable lower.

At 28 V, powerful arcs are possible up to approximately 4 – 5 mm at 20 A.

All arcs generate an insulating (semiconducting) layer of black copper oxide (CuO) on the electrodes, which makes it impossible to reignite the arc up to at least 28 V.

D.1.2.11 Other Studies

Another often quoted curve (from an unknown source) is shown below:

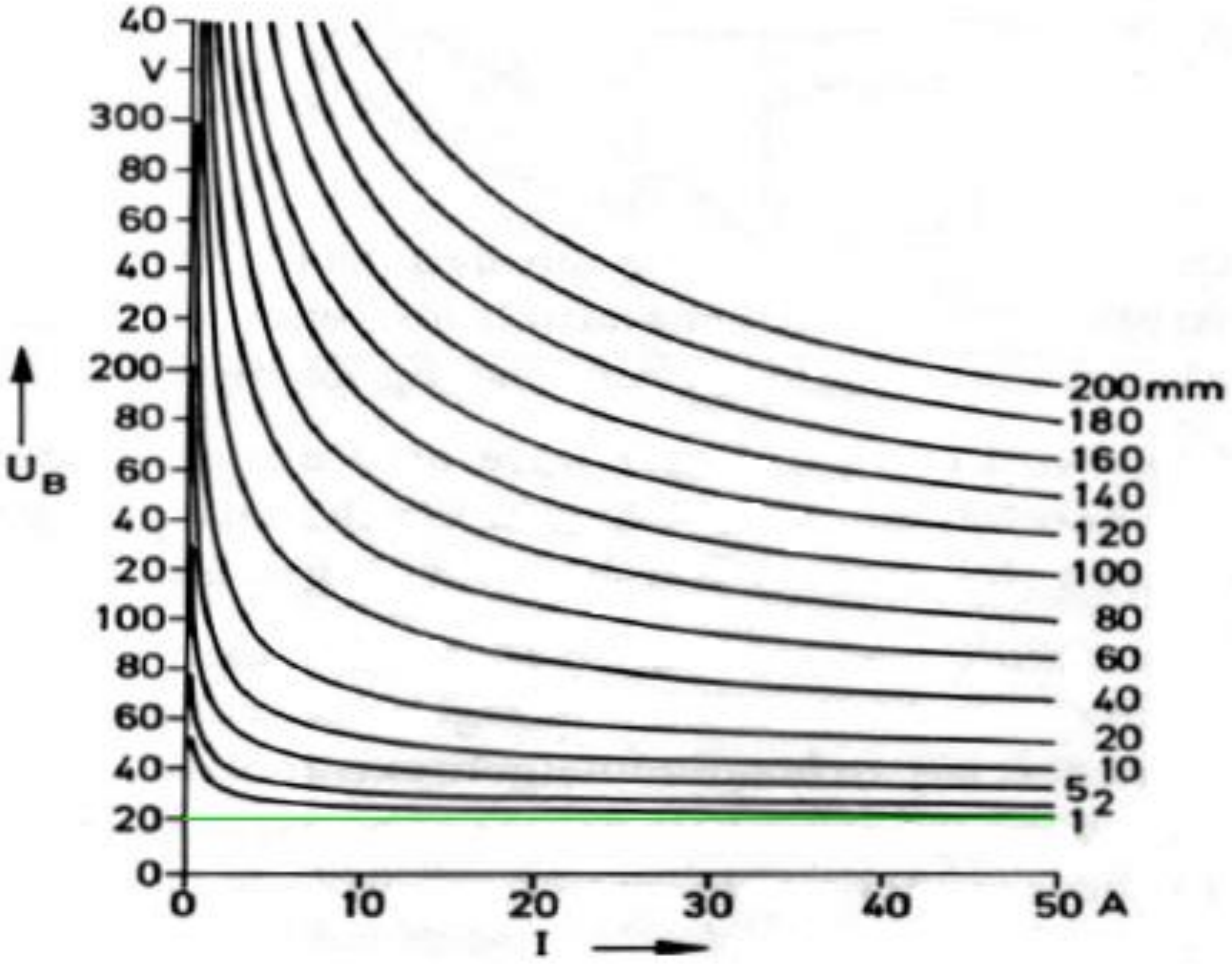


Fig. D.10

As can be seen, the arc length is less than 1 mm at 20 V, which fits with the other studies and the measurements done by Innovatic. It is quite obvious that the transition current depends heavily on the arc length.

A similar curve extracted from “Arcadvisor.com” is shown below:

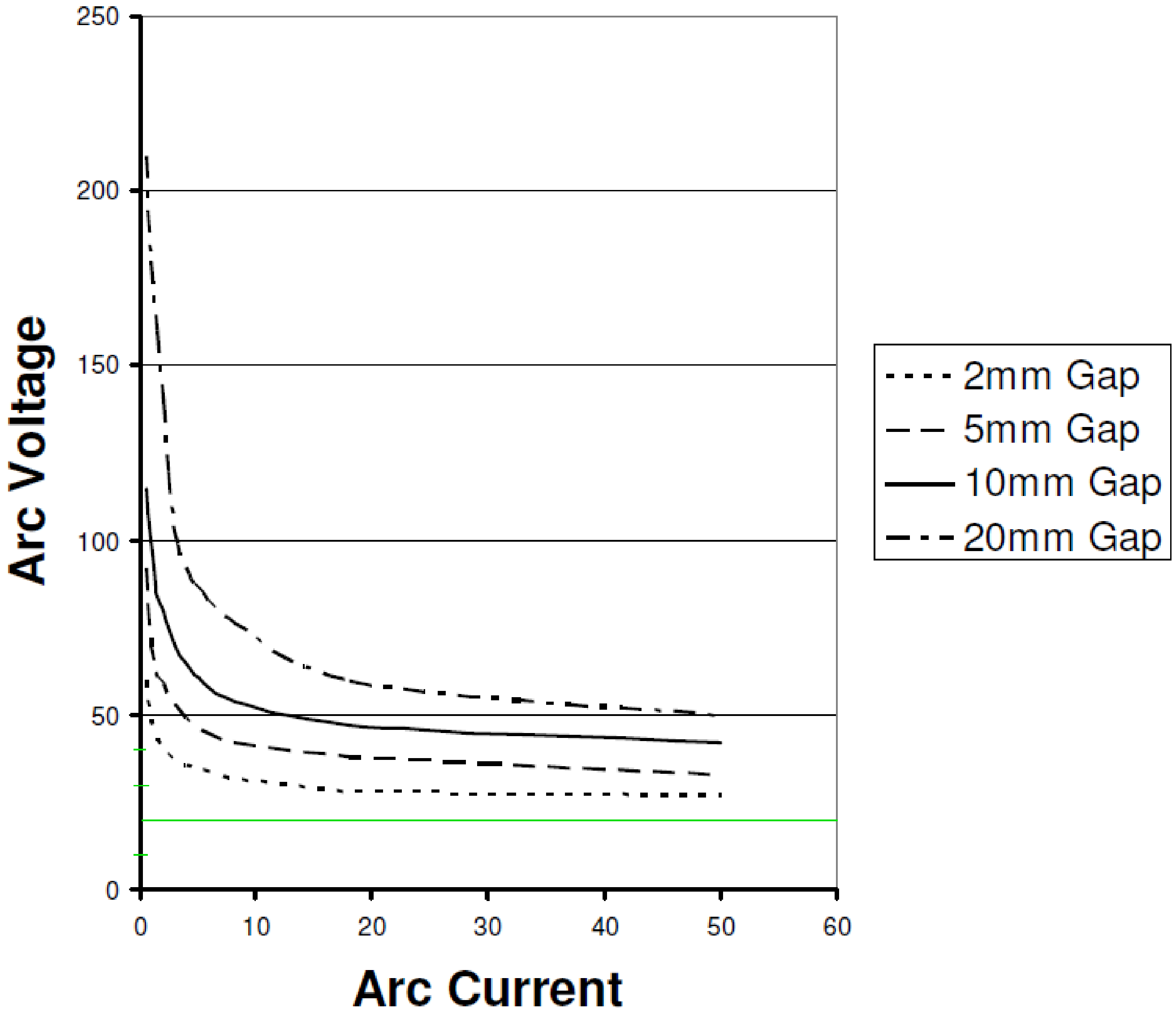


Fig. D.11

The green line is 20 V. This curve also indicates an arc length in the order of 0.5 – 1 mm at 20 V almost independent of the current.

28 Vdc has become a de facto standard in aerospace, but studies of navy data indicates that nearly one-third of incidents involving the wire on its aircraft are on 28 Vdc circuits. Laboratory testing confirms that 28 Vdc arcing can occur quite readily, resulting in arcing of significant energy and temperatures. Consequently, development of 28 Vdc AFCB's is an important requirement for military and commercial aircraft in all categories. Because of this it must be assumed that without AFCB's, 28 V is too high to be used in a very safe DC net. AFCB's complicate things and are best to avoid since they may for example trigger on start-up currents. Contact manufacturers have found that 28 Vdc is as difficult to switch as 125 Vac and since 120 Vac arcing faults have been documented as a significant source of concern in the aerospace industry, this may also indicate that 28 Vdc is too high.

D.1.2.12 Low-level / Dry Circuit Switching

20 V is known as the maximum voltage where it is equally easy to switch AC and DC. For this reason, switches with gold plated contacts designed for very low voltage and/or current levels (typical 0.4 VA maximum) usually have 20 V AC and DC as maximum voltage. Since the cathode drop of gold is 2 V higher than copper, the arc risk at 20 V with gold plated contacts is approximately the same as at 18 V with copper electrodes and therefore fairly low.

D.1.3 Battery cell balancing

When more batteries are connected in series, it becomes difficult to ensure a good balance between the cells. This may require special circuitry, which for example connects a helper battery in parallel with each cell one by one. If one cell has a higher voltage than average, it will charge the helper battery, and when the helper battery is then connected to a cell with a voltage lower than average, the charge will be transferred to that. This is however quite complicated and requires the helper battery plus some high-current MOSFET switches. A much easier solution is only to connect batteries in parallel as it is done in today's automotive 12-V applications. With 18 – 20 V, it is still reasonable to make batteries with the full voltage (9 – 10 cells), but at higher voltages like 24 V, it may be necessary to use a serial connection – especially at high capacity levels like lead-acid car batteries.

D.1.4 Conclusion

The table below summarizes the pros and contras of the various voltage levels.

	12 – 14 V	**18 – 21 V (25.2V max.)**	**24 – 28 V**
Communication and communication loss	+	-	- -
EME	+	0	-
EMI	0	+	+ +
Useable transient protection to keep voltage below 40 V	+ + MOV, TVS	+ TVS	- - Crowbar
Risk of cable overheating and ability to blow fuses	-	0	+
Power loss and necessary cable cross section	- -	0	+ +
Shock risk in wet zone 0 including immersed body	+ + <15 V	+	0 <30 V
Arc risk	+ +	0	- -
Contact wear at a given load power	+	0	-
Low-level / dry circuit switching with gold plated contacts	+	+ +	- -
Cell balancing	+ +	+	-
Usability for LED signal lamps and solid state relays	+	0	-
Usability for intrinsically safe Ex applications	+ +	-	- -
Usability for today’s 24 – 28 V industrial, aerospace and military applications	- -	0 Only 24 V	+ +
Usability for future industrial applications [1])	0	+ +	+
Usability for low voltage generation (3.3 V, 5 V etc.)	+	-	- -
Usability for today’s 12-V automotive applications	+ +	- -	- -
Usability for future automotive applications [2])	+	+ +	-
Usability for trucks and special vehicles	- -	+	+ +
Usability for LED lighting and energy management [3])	- -	+ +	+ +
Usability for direct power delivery through USB port	0	+ +	- -
Usability for mobile phones, modems and cameras	+ +	+	- -
Usability for tablets and hard disk drives	+ +	+ +	-
Usability for laptop computers and monitors	- -	+ +	+
Usability for home entertainment systems (Hifi and TV)	- -	-	0

Fig. D.12

[1]) In the future, the ability to perform low-level switching, drive LED signal lamps and solid state relays and the reduced arc hazard will be much more important than the ability to drive electromechanical relays and valves.

[2]) Modern cars need more and more electrical power so a higher voltage than 12 – 14 V is very desirable. 42 V has been suggested, but there are so many disadvantages that this idea is completely dead today, but 18 – 21 V could be the solution as it reduces the necessary amount of copper to the half and/or doubles the possible power without the many disadvantages of an even higher voltage.

[3]) A high voltage is better for LED lighting than a low one, but 20 – 21 V gets an extra plus because it enables **very** cheap LED lamps with approximately 80 % efficiency based on 4 – 5 LED’s in series in a common package and a simple current generator.

18 – 21 V is the best compromise. 12 – 14 V has too little power and at 24 – 28 V (2 × 12 V), the arc risk is simply too high if high current levels are possible as very powerful arcs over 4 mm are possible at 20 A and up to 10 mm at higher currents.

20 V has a lot of important benefits:

- It fits perfectly with for example USB Power Delivery, Quick Charge, Xiaomi, Infinix, SUPERVOOC Endurance Edition and Realme as shown above. **20 V is simply the new de facto standard for charging of mobile phones, tablets and computers.**
- It fits very well with the charging voltage of for example 10 lead-acid or lead-crystal cells.
- The power loss in the cable is 2.8 times lower than at 12 V, which reduces the risk of overheating considerably.
- It is perfect for low-cost LED lighting up to approximately 17 W as it may be implemented as a simple serial connection of 4 – 5 white LED's (typical 13 – 16 V) and a current generator (usually 0.35, 0.7 or 1.05 A). At 20 V, the efficiency is 80%, which is almost as good as much more expensive switch-mode converters, and the efficiency is **increased** if the voltage drops so the power loss in the cables does not matter. At 17 V, which is the minimum voltage in case of battery backup, the efficiency becomes approximately 94%, which is better than most switch-mode technology. Besides, the standby power consumption of such a solution is **very** low (below 10 mW).
- The power dissipation in the IC is still within reasonable limits (0.7 W average transmitter loss on a 35-Ω trunk line).
- It is the maximum voltage where the arc risk is fairly low as the maximum arc length seems to be approximately 0.5 mm almost independent of the current up to at least 30 A. This makes it possible for bonding wires in IC's to work as fuses, and if the creepage distance are kept above 1 mm, tin whiskers due to lead free soldering will not be able to create an arc.
- It is possible to switch 100 W (5 A) without arc ignition – even up to a charging voltage of 25.2 V. This is important because many loads are limited to this level (IEC/UL 60950) to be able to be connectable to a LPS such as USB Power Delivery or a power supply limited by a 5-A fuse.
- It is the maximum voltage where it is equally easy to switch AC and DC. Just 28 Vdc is as difficult to switch as 125 Vac.
- It is the maximum voltage for low-level / dry circuit switching without the risk of destroying gold contact plating and therefore the ideal voltage for low-level Boolean process signals with a typical switch current in the 4 – 20 mA range (<0.4 VA). At higher voltage levels – even 24 – 28 V, the gold plating may be destroyed due to sparks and small arcs and the underlying contact material – typical silver - may then require fritting to create a reliable connection (see next chapter).
- With a reasonable safety margin, it is below the maximum **open circuit** voltage of RS-232, which is ±25 V. This makes it possible to connect a Max-i controller with open drain or open collector outputs directly to most PC's etc. just by means of a single 2 kΩ resistor to L+, which causes a signal voltage of 15 V (maximum signal for RS-232) at 21 V (input resistance = 5 kΩ).
- 20 V is the absolute maximum voltage of a standard 5-V USB input, so even if this voltage by accident is applied to USB equipment, there is a good chance that it is not harmed.
- On a balanced line, the maximum common mode voltage due to the communication is 2.7 Vrms for the first order component, 0.5 Vrms for the 5th order component and 0.4 Vrms for the 7th order component (14 Vpeak-peak transmitter pulses at 21 V) so it should just be possible to connect CE marked equipment to such a line without problems even if the equipment has high coupling capacitance to ground. At higher voltage levels, the 3 Vrms limit may be exceeded.
- It is the highest voltage where it is fairly easy to limit transients to 40 Vpeak, which is the maximum voltage for automotive devices and applications. The maximum clamp voltage for a typical 22 V heavy-duty TVS avalanche diode is 37 – 39 V at the maximum current, which is 404 A for a 15 kW type such as 15KPA22(A) and 808 A for a 30 kW type such as 30KPA22(A). At 28 V (30 V TVS), it is in practice impossible to limit the voltage to 40 Vpeak for any noticeable transient currents unless crowbar devices such as TSPD's are used, but such devices will short circuit the power supply in case of a transient, which is not acceptable. Therefore, the 28-V MIL-STD-704F has 50 V as maximum. At 14 V, it is also possible to use low voltage MOV's such as 20P11 (20 mm, 11 Vac, 14 Vdc), but although such devices can handle currents up to 8000 A, the maximum current is limited to approximately 50 A if

the voltage must not exceed 40 V and a small surface mounted 1.5 kW 1.5SMC27A TVS can do that equally well (37.5 V at 40 A) and without degeneration, so in practice a big MOV is not an alternative.

D.2 Contact Fritting

At low voltage levels, contact contamination like oxide layers and silver sulfide may prevent a reliable contact. In that case, contact fritting may be a very efficient way to reduce and stabilize the contact resistance. The fritting process consists of two phases:

- A-fritting. During this phase, a voltage bigger than the dielectric strength of the contamination layer, which is 100 – 110 V/µm, is applied to the contacts. This creates a very thin conductive channel. For gold plated contacts, the 20 V of Max-i may be enough, but for other contact materials where contamination layers up to 0.5 µm are fairly common, approximately 50 V may be needed. This is one of the reasons why telephone companies have traditionally used 48 V and some 24-V industrial applications have used an additional -24 V contact loading on the inputs.
- B-fritting. During this phase, the conductive channel is widened due to the heating caused by the current. Current levels of 10 – 100 mA are commonly used, but too high current levels may cause current assisted contamination where the contacts get “addicted” to a high current, and it is very important that the current is too low to create an arc as this will cause a serious oxidation of the contacts. If the two contacts are of different materials, the material, which is expected to get the thickest contamination layer, should be positive so that the majority of the power is dissipated here.

D.3 Choice of Maximum Current and Cable Cross Section

The specified current values are based on current values from the Danish “Stærkstrømsbekendgørelsen” as shown below, but since the actual cooling conditions in thermal insulated walls and ceilings are very difficult to predict, the maximum current for a given cross section has been reduced so that the electrician doesn’t need to worry about that except if the cable is entirely embedded in insulation. In practice, thicker cables than necessary from a temperature point of view may anyway be necessary to fulfill the requirement for maximum voltage drop.

Maximum current for a PVC insulated copper cable with two loaded conductors				
Table number in the Danish “Stærkstrømsbekendgørelsen”	**1.5 mm^2**	**2.5 mm^2**	**4 mm^2**	**6 mm^2**
52-E1 (IEC table 52-C1) with thermal insulation on three sides	14 A	18.5 A	25 A	32 A
52-E1 (IEC table 52-C1) with normal cooling	16.5 A	23 A	30 A	38 A
A 2 with thermal insulation on three sides	13 A	17.5 A	23 A	29 A
A 2 with normal cooling	15 A	20 A	27 A	34 A
801 A for maximum 3 hours at a time with over 50% load, but independent on cooling conditions	13 A	20 A	25 A	32 A
Max-i with thermal insulation on three sides for comparison	10 A	15 A	20 A	25 A

Fig. D.13

D.4 Calculation of Power Loss in Cables and Maximum Cable Length

There are three types of loads:

- Resistive, like heaters and to some extend also incandescent lamps. Since the current is reduced slightly if the voltage drops, the power loss in the cables is slightly less than a calculation based on the current shows, but this can be ignored.

- Constant current, such as low to medium power LED lighting, which uses a simple serial connection of LED's and a current generator. **For these kinds of devices, the power loss in the cable does not matter at all as the loss in the current generator plus the loss in the cable is constant.**
- Constant power, like all devices with switch-mode technology such as PC's, laptop chargers, monitors, high-power LED lighting etc. These kinds of devices increase the current slightly if the voltage drops, so the power loss is slightly higher than a calculation based on the current shows, but as for resistive loads, the difference is so small that it can be ignored in the calculations.

In practice, power loss is difficult to calculate as it depends on the current in the power of two, but if the net is considered equally loaded and connected as a closed ring, it is possible to find the maximum cable/loop length at a given maximum current and cable cross section.

A closed loop may be regarded as two cables in parallel where both ends are connected. If **one** cable with a power supply in one end is equally loaded with N loads, the power loss P_{loss} in the cable is:

$$P_{loss} = R/N \times ((1I/N)^2 + (2I/N)^2 + (3I/N)^2 + ... + (NI/N)^2) \Rightarrow$$

$$P_{loss} = R \times I^2 \times (1^2 + 2^2 + 3^2 + ... + N^2) / N^3 \Rightarrow$$

$$P_{loss} = R \times I^2 \times (N(N + 1)(2N + 1)) / 6N^3 \Rightarrow$$

$$P_{loss} = R \times I^2 \times (N + 1)(2N + 1) / 6N^2 \Rightarrow$$

$$P_{loss} \approx R \times I^2 / 3 \text{ for } N >> 1$$

R is the total cable resistance and I is the total current.

The voltage drop U_{drop} in the far end is:

$$U_{drop} = R/N \times (1I/N + 2I/N + 3I/N + ... + NI/N) \Rightarrow$$

$$U_{drop} = R \times I \times (1 + 2 + 3 + ... + N) / N^2 \Rightarrow$$

$$U_{drop} = R \times I \times N(N + 1) / 2N^2 \Rightarrow$$

$$U_{drop} = R \times I \times (N + 1) / 2N \Rightarrow$$

$$U_{drop} \approx R \times I / 2 \text{ for } N >> 1$$

For an infinite number of devices, the voltage drop in a point K = distance-from-power-supply/cable-length is:

$$U_{drop} = R \times I \times (1 - (1 - K)^2) / 2 \Rightarrow$$

$$U_{drop} = R \times I \times K(2 - K) / 2$$

At for example K = 50% (middle), the voltage drop is 75% of maximum and at K = 10% (fairly close to the power supply), it is 19% of maximum.

Any voltage drop will reduce the power loss in constant current devices. For each device, this reduction is the voltage drop at that point multiplied with the current of that device. For M equally distributed constant current devices with a total current = i, the total power reduction therefore become:

$$P_{reduct} = i/M \times R \times I \times (2(1 + 2 + 3 + ... + M)/M - (1^2 + 2^2 + 3^2 + ... + M^2)/M^2) / 2 \Rightarrow$$

$$P_{reduct} = i/M \times R \times I \times (2M(M + 1)/2M - M(M + 1)(2M + 1)/6M^2) / 2 \Rightarrow$$

$$P_{reduct} = i/M \times R \times I \times ((M + 1) - (M + 1)(2M + 1)/6M) / 2 \Rightarrow$$

$$P_{reduct} \approx i/M \times R \times I \times (M – M/3) / 2 \text{ for } M >> 1 \Rightarrow$$

$$P_{reduct} \approx i \times R \times I / 3$$

If the relationship between the current (i) for constant current devices and the total current (I) is F, the calculated power loss becomes:

$$P_{loss} \approx (R \times I^2 / 3) - (F \times R \times I^2 / 3) = (1 - F) \times R \times I^2 / 3$$

If all devices are constant current devices (F = 1), the loss in the cable does not matter! In a practical net with a mix between a few high-power devices and many small LED lamps, it is reasonable to assume that the current for constant current devices is at least in the order of ¼ of the total current. This reduces the calculated power loss to $(R \times I^2 / 4)$, but as no network is equally loaded, $P_{loss} = R \times I^2 / 3$ may be selected as a reasonable compromise. If for example 5% power loss is acceptable, the maximum cable resistance in each half of the loop may therefore be calculated as:

$$0.05\,(U_{sup} \times I) = R \times I^2 / 3 \;\Rightarrow$$

$$R = 0.15 \times U_{sup} / I$$

For a 20 V power supply (U_{sup}), R becomes:

$$R = 3 / I$$

The maximum loop length therefore becomes:

Cross section at 20 V	Total Power Supply									
	Single fuse					Double fuse				
	10 A	15 A	20 A	25 A	32 A	2×10 A	2×15 A	2×20 A	2×25 A	2×32 A
1.5 mm^2	50 m	-	-	-	-	25 m	-	-	-	-
2.5 mm^2	80 m	55 m	-	-	-	40 m	27 m	-	-	-
4 mm^2	130 m	90 m	65 m	-	-	65 m	45 m	32 m	-	-
2×2.5 mm^2	160 m	110 m	85 m	-	-	80 m	54 m	42 m	-	-
6 mm^2	200 m	130 m	100 m	80 m	-	100 m	65 m	50 m	40 m	-
2×4 mm^2	260 m	180 m	130 m	104 m	80 m	130 m	90 m	62 m	52 m	40 m

Fig. D.14

A closed loop can handle an uneven load much better than a trunk line because the far end is supplied by two cables in parallel. Even if it is worst-case loaded with a single maximum-current load in the far end, the voltage drop will only be 3 V = 15%.

Note that in case of two fuses where each cable segment gets its own fuse, it is possible to double the current without increasing the cross section, but the maximum length of the loop is reduced to the half. A practical DC net may for example consist of a 5 × 2.5 mm^2 flat cable where 2 + 2 conductors are used for power and the middle one for communication. With two 20-A fuses (800 W power), the loop length for each loop may be up to 42 m, which is sufficient for many houses.