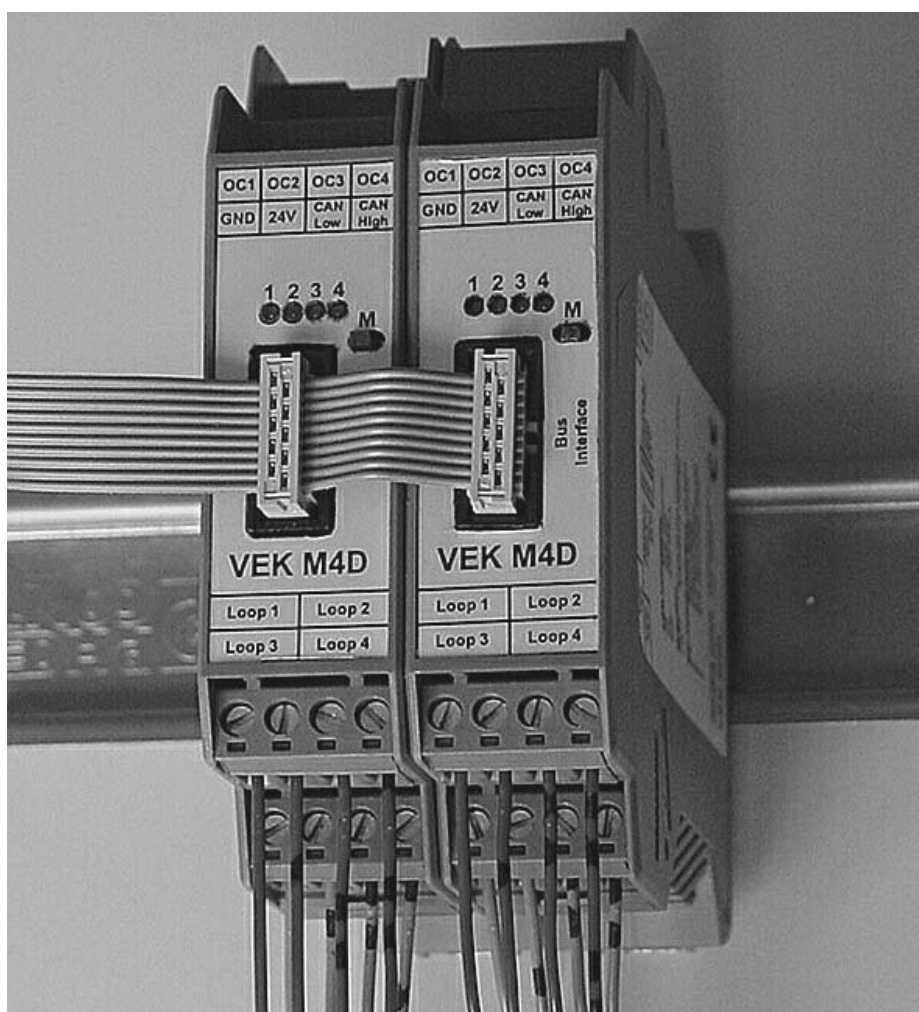


RS-485 Interface Protocol Definition

Traffic Detector - VEK M4D



Contents

1	General	4
2	Telegram Description	4
2.1	Data transfer	4
2.2	Telegram types	4
2.2.1	Long telegram	4
2.2.2	Short telegram	4
2.2.3	Confirmation	5
2.3	Telegram frame	5
2.4	Protocol length	5
2.5	Control byte	5
2.6	Address byte	5
2.7	Data field	5
2.8	Checksum (CS)	5
3	Data transfer	6
3.1	Send – No reply	6
3.2	Send – Confirm	6
3.3	Request/Respond	6
4	Control Bytes	7
4.1	Overview	7
4.2	Sensor-Parameter	8
4.2.1	Masked Request	8
4.2.2	Unmasked Request	8
4.2.3	Reply / Set	9
4.2.4	Parameter blocks	9
4.2.4.1	Mux-Mode	9
4.2.4.2	Frequency Range/Step	10
4.2.4.3	Sensitivity	10
4.2.4.4	Hold Time	10
4.2.4.5	Output Mode	10
4.2.4.6	Direction Logic	11
4.2.4.7	Output Timing	11
4.3	RS485-Parameter	12
4.3.1	Masked Request	12
4.3.2	Unmasked Request	12
4.3.3	Reply / Set	13
4.3.4	Parameter blocks	13
4.3.4.1	Interface	13
4.4	CAN-Parameter	14
4.4.1	Masked Request	14
4.4.2	Unmasked Request	14
4.4.3	Reply / Set	15
4.4.4	Parameter blocks	15
4.4.4.1	Interface	15

4.5	I/O Matrix	16
4.5.1	Masked Request	16
4.5.2	Unmasked Request	16
4.5.3	Reply / Set	17
4.5.4	Matrix blocks	17
4.5.4.1	Presence Signals	17
4.5.4.2	Direction Signals	18
4.5.4.3	Classification Signals (future implementation !)	18
4.6	Data	19
4.6.1	Masked Request	19
4.6.2	Unmasked Request	19
4.6.3	Reply / Set	20
4.6.4	Data blocks	20
4.6.4.1	Frequency	20
4.6.4.2	Amplitude	21
4.6.4.3	Loop Status	21
4.6.4.4	Busy Time	21
4.6.4.5	Maximum Amplitude	22
4.6.4.6	Device Info	22
4.6.4.7	Hardware Signals	22
4.7	Counter	23
4.7.1	Masked Request	23
4.7.2	Unmasked Request	23
4.7.3	Reply	24
4.7.4	Counter blocks	24
4.7.4.1	Presence	24
4.7.4.2	Direction Logic 1	24
4.7.4.3	Direction Logic 2	24
4.7.4.4	Direction Logic parallel traffic	25
5	Interface Timing	26
6	Notes	27

1 General

The traffic detector VEK M4D has a RS-485 Interface. According to the protocols defined by IEC TC57/5 data transfer was controlled on the interface. The primary computer acts as master. Every communication will be initiated from the master. Each connected member, respective detector (secondary), will be selected with his address and answers only on request.

2 Telegram Description

2.1 Data transfer

Transfer mode: serial, asynchronous
 Data format: 8 bit
 Stop bit: 1 bit
 Parity: even (8e1)
 Baud rate: 9600 bps (19200, 38400 bps)

2.2 Telegram types

There 3 different telegram types.

2.2.1 Long telegram

usage: adjustment parameter, detector status message
 data frame: variable length (Start byte, Address byte, Data field)
 frame length: data frame + 6 byte

Start byte	02h
Protocol length	n-6
Protocol length	n-6
Start byte	02h
Control byte	Data frame
Address byte	
Data field	
Checksum	Σ Data frame
Stop byte	04h

2.2.2 Short telegram

usage: data request, error message
 data frame: no
 frame length : 5 byte

Start byte	10h
Control byte	
Address byte	
Checksum	Ctrl.+ Adr. byte
Stop byte	04h

2.2.3 Confirmation

The Confirm character acknowledges the accurate receiving of a telegram¹. According to the IEC-standard erroneous telegrams are not confirmed. Then the receiving device behaves like a non-existent member. After timeout the next device will be requested.

Confirm character	06h
-------------------	-----

2.3 Telegram frame

The telegram is framed with start- and stop-byte. To determine between long- and short-telegram different start-bytes are used.

Start byte long-telegram	02h = "STX"
Start byte short-telegram	10h = "DLE"
Stop byte	04h = "EOT"
Confirm character	06h = "ACK"

2.4 Protocol length

The protocol length defines in the long-telegram the variable telegram length for control byte, address byte and data field. The declaration is indicated twice to increase data security.

2.5 Control byte

The control byte contains the function, which controls the connection between primary (master computer or service device) and secondary(detector) → 3. Data transfer.

2.6 Address byte

The address byte declares the address of the secondary. If data is send from primary to secondary the address is the destination address. From secondary to primary it is the source address.

address \$ 00:	no station address Attention: Every connected secondary answers!
address \$01 .. \$FE:	address secondary
address \$FF:	all station address, reserved for messages to all secondaries. Every connected secondary processes the command, but don't answer!

2.7 Data field

The data field is a variable number of data bytes (Hex-Code, not ASCII-coded). Number of bytes for control byte, address byte and data frame is limited to 255 bytes.

2.8 Checksum (CS)

Checksum is the arithmetical sum of the following bytes:

short-telegram:	CS = (control byte + address byte) MOD 256
long-telegram:	CS = (control byte + address byte + data field) MOD 256

¹ exceptions look up at 3. Data transfer, Send without reply

3 Data transfer

The detector (secondary) sends only by request of the superior master computer, control unit or service device (primary). Data transfer was ruled by the following basic elements:

3.1 Send – No reply

	<u>send request</u>	<u>confirmation</u>
format:	Long-telegram	no
	Short-telegram	no
application:	central time settings	

3.2 Send – Confirm

	<u>send request</u>	<u>confirmation</u>
format:	Long-telegram	Short-telegram
	Long-telegram	Confirm character
application:	parameter setting control instructions	

3.3 Request/Respond

	<u>send request</u>	<u>confirmation</u>
format:	Short-telegram	Long-telegram
	Short-telegram	Short-telegram
	Short-telegram	Confirm character
	Long-telegram	Long-telegram
application:	data request	

The combination request with Long-telegram and answer with Long-telegram is in contrast to the definition in IEC TC57/5 data transfer allowed.

4 Control Bytes

4.1 Overview

<i>Communication</i>	<i>Request Primary</i>		<i>Reply Secondary</i>	
	<i>Protocol type</i>	<i>Control byte</i>	<i>Protocol type</i>	<i>Control byte</i>
Setup/Service				
detector reset	S	01h	Q	-
factory settings	S	02h	Q	-
counter reset	S	03h	Q	-
handshake	S	0Ah	Q	-
Requests				
request Sensor-Parameter	S	11h	L9-45	11h
request RS485-Parameter	S	12h	L9-11	12h
request CAN-Parameter	S	13h	L9-11	13h
request I/O Matrix	S	31h	L9-21	31h
request Data	S	51h	L9-50	51h
request Counter	S	71h	L9-23	71h
Settings				
set Sensor-Parameter	L9-45	21h	Q	-
set RS485-Parameter	L9-11	22h	Q	-
set CAN-Parameter	L9-11	23h	Q	-
set I/O Matrix	L9-21	41h	Q	-
set Data	L9-13	61h	Q	-

S = Short protocol

Lxx = Long protocol, xx bytes

Q = Confirm telegram 06h

If the detector can't execute a request telegram because of an error, the detector answers with the following control bytes in a short protocol:

<i>Control byte</i>	<i>Error message</i>
F1h	Parameter block 1 out of bounds
F2h	Parameter block 2 out of bounds
F3h	Parameter block 3 out of bounds
F4h	Parameter block 4 out of bounds
F5h	Parameter block 5 out of bounds
F6h	Parameter block 6 out of bounds
F7h	Parameter block 7 out of bounds
F8h	Parameter block 8 out of bounds
FAh	Parameter out of bounds
FDh	Request mask error
FEh	Protocol length error
FFh	unknown request / control byte

4.2 Sensor-Parameter

4.2.1 Masked Request

Byte 0	Start byte	02h
Byte 1	Data Frame length	3
Byte 2	Data Frame length	3
Byte 3	Start byte	02h
Byte 4	Control byte	11h
Byte 5	Address byte	0...255
Byte 6	Parameter Request Mask	
Byte 7	Checksum	Byte 4 ... 6
Byte 8	Stop byte	04h = "EOT"

Mux-Mode: multiplex order, sampling speed and filter

Freq. range: number of frequency band

Sensitivity: sensitivity threshold and signal hysteresis

Hold time: maximum busy time where detector starts automatically reset

Outp. mode: output function (presence-, direction-, classification mode), always on/off, fault output, normal mode, inverted signals

Dir. param.: Logic of direction sensing detection

Outp. Timing: On-/Off-Delay, min./max. Signal Duration of Low Side Driver output pins

4.2.2 Unmasked Request

Byte 0	Start byte	10h
Byte 1	Control byte	11h
Byte 2	Address byte	0...255
Byte 3	Checksum	Control + Address Byte
Byte 4	Stop byte	04h = "EOT"

With an unmasked request data marked in the pre-set *parameter request mask* will be send.

4.2.3 Reply / Set

Byte 0	Start byte	02h
Byte 1	Data Frame length	3..39 (Byte 4 – Byte n-2)
Byte 2	Data Frame length	3..39 (Byte 4 – Byte n-2)
Byte 3	Start byte	02h
Byte 4	Control byte	Reply 11h Set 21h
Byte 5	Address byte	0..255
Byte 6	Parameter Request Mask	for bit definition look at “masked request”
2 Byte	Parameter block 1 <i>Mux-Mode Data</i> → Section 4.2.4.1	if M-Bit is set in request mask
4 Byte	Parameter block 2 <i>Frequency range</i> → Section 4.2.4.2	if F-Bit is set in request mask
8 Byte	Parameter block 3 <i>Sensitivity</i> → Section 4.2.4.3	if S-Bit is set in request mask
4 Byte	Parameter block 4 <i>Hold time</i> → Section 4.2.4.4	if H-Bit is set in request mask
4 Byte	Parameter block 5 <i>Output Mode</i> → Section 4.2.4.5	if O-Bit is set in request mask
2 Byte	Parameter block 6 <i>Direction Logic</i> → Section 4.2.4.6	if D-Bit is set in request mask
12 Byte	Parameter block 7 <i>Output Timing</i> → Section 4.2.4.7	if T-Bit is set in request mask
Byte n-1	Checksum	Byte 4...n-2
Byte n	Stop byte	04h = “EOT”

Maximum length 45 Byte !

4.2.4 Parameter blocks

4.2.4.1 Mux-Mode

Byte 0	Speed / Filter	Bit 0-1: Sampling Speed 0: x1 (all loops) (default) 1: x2 (two loops) 2: x4 (one loop) Bit 2: Filter 0: disabled (high speed) 1: enabled (high immunity) (default) Bit 3-7: --															
Byte 1	Mux order	<table border="0"> <tr> <td></td> <td>time frame</td> <td>default</td> </tr> <tr> <td>Bit 0-1:</td> <td>1</td> <td>0</td> </tr> <tr> <td>Bit 2-3:</td> <td>2 loop no. (0..3)</td> <td>1</td> </tr> <tr> <td>Bit 4-5:</td> <td>3</td> <td>2</td> </tr> <tr> <td>Bit 6-7:</td> <td>4</td> <td>3</td> </tr> </table>		time frame	default	Bit 0-1:	1	0	Bit 2-3:	2 loop no. (0..3)	1	Bit 4-5:	3	2	Bit 6-7:	4	3
	time frame	default															
Bit 0-1:	1	0															
Bit 2-3:	2 loop no. (0..3)	1															
Bit 4-5:	3	2															
Bit 6-7:	4	3															

4.2.4.2 Frequency Range/Step

Byte 0	Frequency range/step loop 1	01h: range 30-40 kHz	
Byte 1	Frequency range/step loop 2	02h: range 45-55 kHz	
Byte 2	Frequency range/step loop 3	03h: range 60-75 kHz	
Byte 3	Frequency range/step loop 4	04h: range 80-100 kHz	(default)
		05h: range 105-140 kHz	
		11h: transformer step 1	
		:	
		18h: transformer step 8	

4.2.4.3 Sensitivity

Byte 0	Level loop 1	0: Threshold 4	0,005% $\Delta f/f$
Byte 1	Level loop 2	1 Threshold 10	0,013% $\Delta f/f$
Byte 2	Level loop 3	:	
Byte 3	Level loop 4	12 Threshold 120 (def.)	0,150% $\Delta f/f$
Byte 4	Hysteresis loop 1	:	
Byte 5	Hysteresis loop 2	255 Threshold 2550	3,188% $\Delta f/f$
Byte 6	Hysteresis loop 3	20..80 % (default: 75%)	
Byte 7	Hysteresis loop 4		

4.2.4.4 Hold Time

Byte 0	Hold time loop 1	0: infinite 1..255 minutes (default: 20 minutes)
Byte 1	Hold time loop 2	
Byte 2	Hold time loop 3	
Byte 3	Hold time loop 4	

4.2.4.5 Output Mode

Byte 0	Mode output 1	Bit 0-2: Output mode 000 → default off 001 → default on 010 → general fault mode 011 → normal mode (default) 100 → simulation mode Bit 3: Invert mode 0: non inverted signal (default) 1: inverted signal Bit 4-7: Error bits (only for normal mode) Bit 4 → Adjustment bit Bit 5 → Frequency fault bit (default) Bit 6 → Loop fault bit (default) Bit 7 → - reserved -
Byte 1	Mode output 2	
Byte 2	Mode output 3	
Byte 3	Mode output 4	

In *general fault mode* hardware output reacts only on loop faults. Additional *Error bits* are ignored.

4.3 RS485-Parameter

4.3.1 Masked Request

Byte 0	Start byte	02h
Byte 1	Data Frame length	3
Byte 2	Data Frame length	3
Byte 3	Start byte	02h
Byte 4	Control byte	12h
Byte 5	Address byte	0...255
Byte 6	Parameter Request Mask	
Byte 7	Checksum	Byte 4 ... 6
Byte 8	Stop byte	04h = "EOT"

Interface: Address offset, Parity and Baudrate

4.3.2 Unmasked Request

Byte 0	Start byte	10h
Byte 1	Control byte	12h
Byte 2	Address byte	0...255
Byte 3	Checksum	Control + Address Byte
Byte 4	Stop byte	04h = "EOT"

With an unmasked request data marked in the pre-set *parameter request mask* will be send.

4.3.3 Reply / Set

Byte 0	Start byte	02h
Byte 1	Data Frame length	3..5 (Byte 4 – Byte n-2)
Byte 2	Data Frame length	3..5 (Byte 4 – Byte n-2)
Byte 3	Start byte	02h
Byte 4	Control byte	Reply 12h Set 22h
Byte 5	Address byte	0..255
Byte 6	Parameter Request Mask	for bit definition look at “masked request”
2 Byte	Parameter block 1 <i>Interface</i> → <i>Section 4.3.4.1</i>	if I-Bit is set in request mask
Byte n-1	Checksum	Byte 4...n-2
Byte n	Stop byte	04h = “EOT”

Maximum length 11 Byte !

4.3.4 Parameter blocks

4.3.4.1 Interface

Byte 0	Address offset	0..254 (default:0)
Byte 1	Baudrate, Parity	Bit 0-2: 1→ 38400 Baud 2→ 19200 Baud 3→ 9600 Baud (default) Bit 4: 0→ even Parity (default) 1→ odd Parity Bit 5: 0→ no Parity 1→ Parity enable (default) Bit 3,6,7: - reserved -

Variable *Address offset* is shared with CAN-Interface.

4.4 CAN-Parameter

4.4.1 Masked Request

Byte 0	Start byte	02h
Byte 1	Data Frame length	3
Byte 2	Data Frame length	3
Byte 3	Start byte	02h
Byte 4	Control byte	13h
Byte 5	Address byte	0...255
Byte 6	Parameter Request Mask	
Byte 7	Checksum	Byte 4 ... 6
Byte 8	Stop byte	04h = "EOT"

Interface: Address offset and Baudrate

4.4.2 Unmasked Request

Byte 0	Start byte	10h
Byte 1	Control byte	13h
Byte 2	Address byte	0...255
Byte 3	Checksum	Control + Address Byte
Byte 4	Stop byte	04h = "EOT"

With an unmasked request data marked in the pre-set *parameter request mask* will be send.

4.4.3 Reply / Set

Byte 0	Start byte	02h
Byte 1	Data Frame length	3..5 (Byte 4 – Byte n-2)
Byte 2	Data Frame length	3..5 (Byte 4 – Byte n-2)
Byte 3	Start byte	02h
Byte 4	Control byte	Reply 13h Set 23h
Byte 5	Address byte	0..255
Byte 6	Parameter Request Mask	for bit definition look at “masked request”
2 Byte	Parameter block 1 <i>Interface</i> → <i>Section 4.4.4.1</i>	if I-Bit is set in request mask
Byte n-1	Checksum	Byte 4...n-2
Byte n	Stop byte	04h = “EOT”

Maximum length 11 Byte !

4.4.4 Parameter blocks

4.4.4.1 Interface

Byte 0	Address offset	0..254 (default:0)
Byte 1	Baudrate	Bit 0-4: 1→ 20 kBaud 2→ 50 kBaud 3→ 100 kBaud (default) 4→ 125 kBaud 5→ 250 kBaud 6→ 500 kBaud 7→ 800 kBaud 8→ 1000 kBaud Bit 5-7: - reserved -

Variable *Address offset* is shared with RS485-Interface.

4.5 I/O Matrix

4.5.1 Masked Request

Byte 0	Start byte	02h
Byte 1	Data Frame length	3
Byte 2	Data Frame length	3
Byte 3	Start byte	02h
Byte 4	Control byte	31h
Byte 5	Address byte	0...255
Byte 6	I/O Matrix Request Mask	
Byte 7	Checksum	Byte 4 ... 6
Byte 8	Stop byte	04h = "EOT"

Presence: definition of loop inputs and output channels for presence signals

Direction logic: definition of loop inputs and output channels for direction logic

Classification: definition of loop inputs and output channels for simple classification

4.5.2 Unmasked Request

Byte 0	Start byte	10h
Byte 1	Control byte	31h
Byte 2	Address byte	0...255
Byte 3	Checksum	Control + Address Byte
Byte 4	Stop byte	04h = "EOT"

With an unmasked request data marked in the pre-set *I/O Matrix request mask* will be send.

4.5.3 Reply / Set

Byte 0	Start byte	02h
Byte 1	Data Frame length	3..15 (Byte 4 - Byte n-2)
Byte 2	Data Frame length	3..15 (Byte 4 - Byte n-2)
Byte 3	Start byte	02h
Byte 4	Control byte	Reply 31h Set 41h
Byte 5	Address byte	0..255
Byte 6	I/O Matrix Request Mask	for bit definition look at "masked request"
4 Byte	Matrix block 1 <i>Presence signals</i> → Section 4.5.4.1	if P-Bit is set in request mask
4 Byte	Matrix block 2 <i>Direction signals</i> → Section 4.5.4.2	if D-Bit is set in request mask
4 Byte	Matrix block 3 <i>Classification Signals</i> → Section 4.5.4.3	if C-Bit is set in request mask
Byte n-1	Checksum	Byte 4...n-2
Byte n	Stop byte	04h = "EOT"

Maximum length 21 Byte !

4.5.4 Matrix blocks

4.5.4.1 Presence Signals

Byte 0	Presence signal 1	Bit 0..3: input loop 1 0001 _b (fixed) Bit 4..7: output channel 0001 _b channel 1 (default) 0000 _b no output
Byte 1	Presence signal 2	Bit 0..3: input loop 2 0010 _b (fixed) Bit 4..7: output channel 0010 _b channel 2 (default) 0000 _b no output
Byte 2	Presence signal 3	Bit 0..3: input loop 3 0100 _b (fixed) Bit 4..7: output channel 0100 _b channel 3 (default) 0000 _b no output
Byte 3	Presence signal 4	Bit 0..3: input loop 4 1000 _b (fixed) Bit 4..7: output channel 1000 _b channel 4 (default) 0000 _b no output

The loop inputs for presence signals are fixed as shown above. To turn an output channel off, write 0000_b to Bit4..7.

4.5.4.2 Direction Signals

Byte 0	Matrix Logic I direction A	Bit 0..3: input loop 0000 _b no input 0001 _b loop 1 (default A) 0010 _b loop 2 (default B) 0100 _b loop 3 1000 _b loop 4
Byte 1	Matrix Logic I direction B	Bit 4..7: output channel 0000 _b no output (default) 0001 _b channel 1 0010 _b channel 2 0100 _b channel 3 1000 _b channel 4
Byte 2	Matrix Logic II direction A	Bit 0..3: input loop 0000 _b no input 0001 _b loop 1 0010 _b loop 2 0100 _b loop 3 (default A) 1000 _b loop 4 (default B)
Byte 3	Matrix Logic II direction B	Bit 4..7: output channel 0000 _b no output (default) 0001 _b channel 1 0010 _b channel 2 0100 _b channel 3 1000 _b channel 4

4.5.4.3 Classification Signals*(future implementation !)*

Byte 0	Matrix classification 1	Bit 0..3: input loop 0000 _b no input (default) 0001 _b loop 1 0010 _b loop 2 0100 _b loop 3 1000 _b loop 4
Byte 1	Matrix classification 2	
Byte 2	Matrix classification 3	Bit 4..7: output channel 0000 _b no output (default) 0001 _b channel 1 0010 _b channel 2 0100 _b channel 3 1000 _b channel 4
Byte 3	Matrix classification 4	

4.6 Data

4.6.1 Masked Request

Byte 0	Start byte	02h
Byte 1	Data Frame length	3
Byte 2	Data Frame length	3
Byte 3	Start byte	02h
Byte 4	Control byte	51h
Byte 5	Address byte	0...255
Byte 6	Data Request Mask	
Byte 7	Checksum	Byte 4 ... 6
Byte 8	Stop byte	04h = "EOT"

Frequency: frequency values in kHz per channel

Amplitude: actual detuning value in units per channel

Loop Status: status of loop in dependency of sensitivity settings

Busy time: length of time where loop is covered in dependency of sensitivity settings

Max. Ampl.: maximum amplitude of last object per channel

Device Info: Software version and date

Hardw. Signal: status of open collector output signals

4.6.2 Unmasked Request

Byte 0	Start byte	10h
Byte 1	Control byte	51h
Byte 2	Address byte	0...255
Byte 3	Checksum	Control + Address Byte
Byte 4	Stop byte	04h = "EOT"

With an unmasked request data marked in the pre-set *data request mask* will be send.

4.6.3 Reply / Set

Set is only available for *Loop Status* and *Hardware Signals* !

Byte 0	Start byte	02h
Byte 1	Data Frame length	3..44 (Byte 4 - Byte n-2)
Byte 2	Data Frame length	3..44 (Byte 4 - Byte n-2)
Byte 3	Start byte	02h
Byte 4	Control byte	Reply 51h Set 61h
Byte 5	Address byte	0...255
Byte 6	Data Request Mask	for bit definition look at "masked request"
4 Byte	Data block 1 <i>Frequency</i> → Section 4.6.4.1	if F-Bit is set in request mask
8 Byte	Data block 2 <i>Amplitude</i> → Section 4.6.4.2	if A-Bit is set in request mask
4 Byte	Data block 3 <i>Loop Status</i> → Section 4.6.4.3	if L-Bit is set in request mask
8 Byte	Data block 4 <i>Busy time</i> → Section 4.6.4.4	if B-Bit is set in request mask
8 Byte	Data block 5 <i>Maximum Amplitude</i> → Section 4.6.4.5	if M-Bit is set in request mask
8 Byte	Data block 6 <i>Device Info</i> → Section 4.6.4.6	if I-Bit is set in request mask
1 Byte	Data block 7 <i>Hardware Signals</i> → Section 4.6.4.7	if H-Bit is set in request mask
Byte n-1	Checksum	Byte 4...n-2
Byte n	Stop byte	04h = "EOT"

Maximum length 50 Byte !

4.6.4 Data blocks

4.6.4.1 Frequency

Byte 0	Frequency loop 1	Frequency [kHz]
Byte 1	Frequency loop 2	
Byte 2	Frequency loop 3	
Byte 3	Frequency loop 4	

If a request occurs immediately after frequency setting or reset, the answer contains 0 kHz. Please try again after end of retuning!

4.6.4.2 Amplitude

Byte 0	Amplitude loop 1, high	-32767...32767
Byte 1	Amplitude loop 1, low	
Byte 2	Amplitude loop 2, high	-32767...32767
Byte 3	Amplitude loop 2, low	
Byte 4	Amplitude loop 3, high	-32767...32767
Byte 5	Amplitude loop 3, low	
Byte 6	Amplitude loop 4, high	-32767...32767
Byte 7	Amplitude loop 4, low	

4.6.4.3 Loop Status

Byte 0	Status Loop 1	Bit 0: Covered (level sensitive) Bit 1: Covered (low amplitude) Bit 2: Adjusting Bit 3: - reserved - Bit 4-5: Fault Status 0: No Fault 1: Loop Fault 2: Frequency out of settings Bit 6: - reserved - Bit 7: 0: don't change Loop Status 1: change Loop Status
Byte 1	Status Loop 2	
Byte 2	Status Loop 3	
Byte 3	Status Loop 4	

Setting of loop status (Bit 0 and Bit 7) overwrites the evaluated loop status. It is only temporary and will not stored to EEPROM!

4.6.4.4 Busy Time

Byte 0	Busy Time loop 1, high	0..65535 minutes
Byte 1	Busy Time loop 1, low	
Byte 2	Busy Time loop 2, high	0..65535 minutes
Byte 3	Busy Time loop 2, low	
Byte 4	Busy Time loop 3, high	0..65535 minutes
Byte 5	Busy Time loop 3, low	
Byte 6	Busy Time loop 4, high	0..65535 minutes
Byte 7	Busy Time loop 4, low	

If loop is free, busy time of last object is transmitted ! For a covered loop the actual busy time is transmitted.

No Overflow! If busy time is greater than 65535 minutes the maximum value 65535 is transmitted!

65535 minutes => 1092,25 hours => 45,5 days

4.6.4.5 Maximum Amplitude

Byte 0	Max. Amplitude loop 1, high	-32767...32767
Byte 1	Max. Amplitude loop 1, low	
Byte 2	Max. Amplitude loop 2, high	-32767...32767
Byte 3	Max. Amplitude loop 2, low	
Byte 4	Max. Amplitude loop 3, high	-32767...32767
Byte 5	Max. Amplitude loop 3, low	
Byte 6	Max. Amplitude loop 4, high	-32767...32767
Byte 7	Max. Amplitude loop 4, low	

Maximum amplitude of last object !

4.6.4.6 Device Info

Byte 0	Software Version, Year	0...255
Byte 1	Software Version, Month	0...12
Byte 2	Software Version, Day	1...31
Byte 3	Software Version, Type	ASCII values
Byte 4	Software Version, Units	0...255
Byte 5	Software Version, Tenth	0..99
Byte 6	Software Version, Beta mark	ASCII values
Byte 7	- reserved -	0...255

4.6.4.7 Hardware Signals

Byte 0	Status Open Collector Signals	Bit 0: channel 1 Bit 1: channel 2 0: inactive Bit 2: channel 3 1: active Bit 3: channel 4 Bit 4: mask 1 0: signal according to Bit 5: mask 2 output mode Bit 6: mask 3 1: change output Bit 7: mask 4
--------	-------------------------------	---

Setting of hardware signals overwrites the evaluated open collector signals according to the settings in *Output Mode*. Setting data is only temporary and will not stored to EEPROM!

4.7 Counter

4.7.1 Masked Request

Byte 0	Start byte	02h
Byte 1	Data Frame length	3
Byte 2	Data Frame length	3
Byte 3	Start byte	02h
Byte 4	Control byte	71h
Byte 5	Address byte	0...255
Byte 6	Counter Request Mask	
Byte 7	Checksum	Byte 4 ... 6
Byte 8	Stop byte	04h = "EOT"

Presence: counter for number of occupancies in dependency of sensitivity settings

Direction I: counter for direction sensitive signals I

Direction II: counter for direction sensitive signals II

Dir. Parallel: counter for direction sensitive signals on parallel traffic (4 loops)

4.7.2 Unmasked Request

Byte 0	Start byte	10h
Byte 1	Control byte	71h
Byte 2	Address byte	0...255
Byte 3	Checksum	Control + Address Byte
Byte 4	Stop byte	04h = "EOT"

With an unmasked request data marked in the pre-set *counter request mask* will be send.

4.7.3 Reply

Set for counter is not available !

Byte 0	Start byte	02h
Byte 1	Data Frame length	3..23 (Byte 4 - Byte n-2)
Byte 2	Data Frame length	3..23 (Byte 4 - Byte n-2)
Byte 3	Start byte	02h
Byte 4	Control byte	Reply 71h
Byte 5	Address byte	0..255
Byte 6	Counter Request Mask	for bit definition look at "masked request"
8 Byte	Counter block 1 <i>Presence</i> → Section 4.7.4.1	if P-Bit is set in request mask
4 Byte	Counter block 2 <i>Direction logic 1</i> → Section 4.7.4.2	if D1-Bit is set in request mask
4 Byte	Counter block 3 <i>Direction logic 2</i> → Section 4.7.4.3	if D2-Bit is set in request mask
4 Byte	Counter block 4 <i>Direction logic parallel traffic</i> → Section 4.7.4.4	if DP-Bit is set in request mask
Byte n-1	Checksum	Byte 4...n-2
Byte n	Stop byte	04h = "EOT"

Maximum length 29 Byte !

For all counters overflow is possible ! To reset all counter use short protocol with control byte 03h.

4.7.4 Counter blocks

4.7.4.1 Presence

Byte 0	Presence counter 1, high	0..65535
Byte 1	Presence counter 1, low	
Byte 2	Presence counter 2, high	0..65535
Byte 3	Presence counter 2, low	
Byte 4	Presence counter 3, high	0..65535
Byte 5	Presence counter 3, low	
Byte 6	Presence counter 4, high	0..65535
Byte 7	Presence counter 4, low	

4.7.4.2 Direction Logic 1

Byte 0	Direction A counter, high	0..65535
Byte 1	Direction A counter, low	
Byte 2	Direction B counter, high	0..65535
Byte 3	Direction B counter, low	

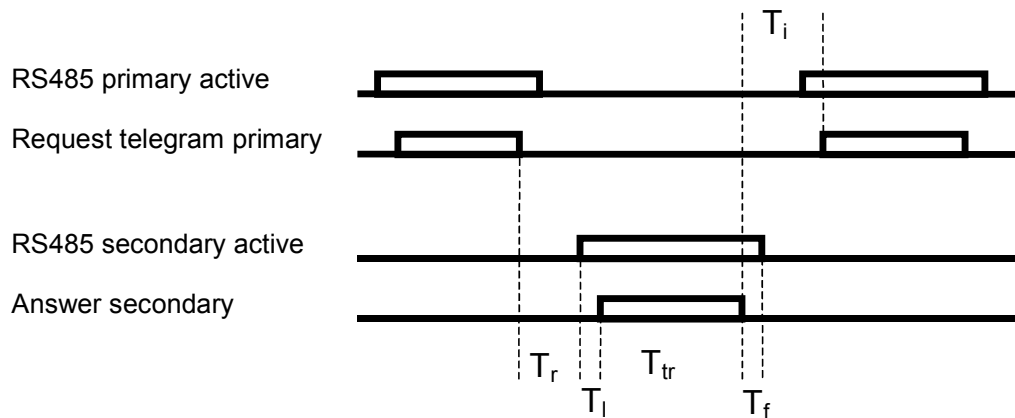
4.7.4.3 Direction Logic 2

Byte 0	Direction A counter, high	0..65535
Byte 1	Direction A counter, low	
Byte 2	Direction B counter, high	0..65535
Byte 3	Direction B counter, low	

4.7.4.4 Direction Logic parallel traffic

Byte 0	Direction A counter, high	0..65535
Byte 1	Direction A counter, low	
Byte 2	Direction B counter, high	0..65535
Byte 3	Direction B counter, low	

5 Interface Timing



The following time value loop 1s should be considered for data transmission:

transmission time 1 character	T_z	e.g. 1.2 ms for 9600 bps
reaction time	T_r	4..10 ms for normal mode 4..50 ms for changing parameter
lead time interface active	T_l	$1 \times T_z$
protocol transmission time	T_{tr}	$n \times T_z$
follow up time interface active	T_f	$1 \times T_z$
idle time for new request on accurate answer from detector	T_i	5 ms
idle time for new request on missing answer from detector		$> T_{r_{max}}$

