DMXface
ACTIVE SEND with RS232 / LAN

Covers DMXface FW revision 5.15-5.27

Changes to previous versions

Rev. 5.60
Extension of the requestable CharBuffers to 12
To rev. 5.27
No change to the ACTIVE SEND PROTOCOL
Rev. 5.18
Additional command to set / read the DMX master value oft he DMX512 Output
DMXface controllers with firmware rev. 5.18 support automatic ACTS sending of all available DMX channels (previous release max. 224 channels)
Rev. 5.16 Documentation update
14.07.2020 Scene update Command, Character ‚U' is HEXcode 0x55 instead of 0x43 (wrong in previous release of manual)
Additional command for updating a scene with the current DMX values of the DMX OUT.
Additional command to recall a scene with alternative brightness and fade time.
Additional command to request firmware version of the controller.
Additional command to request IO and DMX status with one command.
Rev. 5.15
Additional command to read and set the real time clock of the controller.
Rev. 5.14
16 Bit addressing of DMX channels.
Affected commands:

- DMX channel values
- DMX OUT request
- DMX IN request

General

The DMXface controllers are basically connected to a PC or other control system via USB or LAN (DMXfaceXP).

The communication protocol (MAIN COMMUNICATION) is complex to use.
In addition to start, lengths and end bytes, RS485 communication also includes bus addresses in the data packets.

In order to offer a simple communication interface with the essential functions, which can also be used with RS232 port 1, all DMXface controllers from Rev. 5.07 support the ACTIVE SEND (= ACTS) protocol.

ACTIVE SEND understands simple commands for various queries and for controlling the controller. Furthermore, the DMXface can be configured so that it automatically sends data packets when selected events occur.
The status of DMX channels can be sent cyclically without further requests to the host system.

The DMXfaceXP with integrated LAN interface provides seven configurable network sockets.

The function of socket 2-7 can be assigned to the Active Send protocol, so that several parallel communication channels for ACTS with the same controller are available in the network.

Sockets 6 + 7 are also available for the automated sending of event messages.

The settings can be adjusted in the "DMXface settings" / "Basic setup" menu.
The DMXfaceXP has an additional "Network setup" menu.
All settings are made with the downloadable DMXface Console program .

Communication channels and ACTIVE SEND

Below you find a list of available functions, there can also be used multiple interfaces simultaneously.

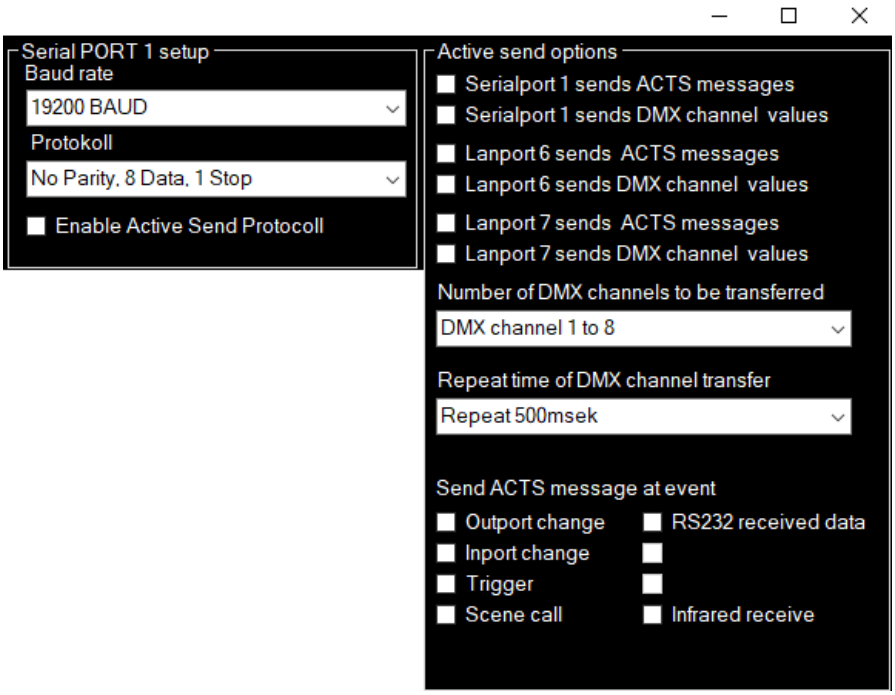
<table>
<tr><th>Function</th><th colspan="7">DMXfaceXP</th><th>All DMXface</th></tr>
<tr><th></th><th>SOCKET1</th><th>SOCKET2</th><th>SOCKET3</th><th>SOCKET4</th><th>SOCKET5</th><th>SOCKET6</th><th>SOCKET7</th><th>SERIALPORT 1</th></tr>
<tr><td>Version request</td><td colspan="7" rowspan="8">The network sockets must be set to the Active Send Mode to get the function available.</td><td rowspan="8"></td></tr>
<tr><td>Scene call</td></tr>
<tr><td>Scene call with alternative brightness / fade time</td></tr>
<tr><td>Scene update with current DMX values of DMX OUT</td></tr>
<tr><td>Program call</td></tr>
<tr><td>Set a DMX channel to a value</td></tr>
<tr><td>Set an OUTPORT</td></tr>
<tr><td>Request an INPORT state (digital + analog + table translation)</td></tr>
<tr><td>DMX OUT request</td><td colspan="7" rowspan="3"></td><td>Max. 255 channels in one request</td></tr>
<tr><td>DMX IN request</td><td>Max. 255 channels in one request</td></tr>
<tr><td>DMX OUT + IO request with one command</td><td></td></tr>
<tr><td>Automated message when an OUTPORT changes</td><td colspan="5" rowspan="7"></td><td rowspan="7"></td><td rowspan="7"></td></tr>
<tr><td>Automated message when an INPORT changes</td></tr>
<tr><td>Automated message when a TRIGGER triggers</td></tr>
<tr><td>Automated message when a scene is called</td></tr>
<tr><td>Automated message when a infrared is received</td></tr>
<tr><td>Cyclic transmission DMX OUT values (max. 224 channels)</td></tr>
</table>

Automated sending of datapackages is dependent from the settings in the basic setup of the controller.

Active send options for automated messages:
Cyclic send the state of none, 8, 16, 32, 64, 128 or 224 DMX channels.
(500 msec. to 10 sec.)
Sending a datapackage when an INPORT changes.
Sending a datapackage when INFRARED is received
Sending a datapackage when a scene was called.
Sending a datapackage when a TRIGGER occurs.
Sending a datapackage when MIDI, DALI or RS232 (Serial Port 2) was received.

Setting up the ACTIVE Send

The settings can be made with the DMXface Console: DMXface settings, Basic setup menu.



A checkbox to activate ACTS with serial port 1 is available at the Serial port 1 setup area.

As soon as the function is activated and the setting is saved, data arriving at Serial port 1 is checked for ACTS commands.
An activated ACTS on serial port 1 also has the special feature, that unrecognized commands are forwarded to the triggers in order to be processed there if a defined pattern exists.

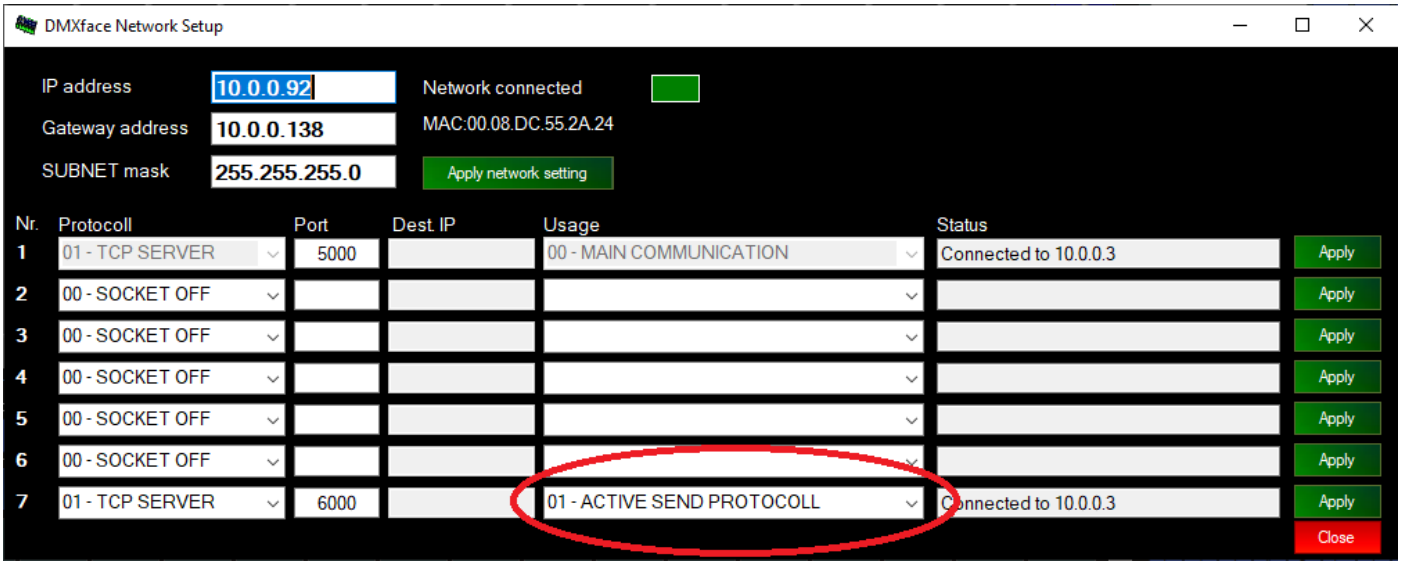
If Serial Port 1 is used, the Baud rate selected, has to be high enough to manage the data traffic.

The option **“Port sends ACTS Messages”** is used to make the DMXface send messages to the selected interface as soon as a selected event occurs.
The events can be selected with the checkboxes at the bottom of the window.

The option **„Port sends DMX channel values“** can be activated, if you wish to receive cyclic information about the DMX channels values.
The time behaviour and the number of the DMX channels transmitted are choosen in the two drop down boxes.

Network setup at the DMXfaceXP to configure the network sockets

The settings of the network ports on DMXfaceXP are available at the DMXface Console: DMXface settings, Network setup menu.



The sockets you want to use for the ACTIVE Send communication must be set to the operating mode ACTIVE SEND PROTOCOL.
Automated messages can only be sent with the sockets 6 or 7.

ACTS Eventdata packages sent by the DMXface

Provided that event sending and events are activated in the setup, event messages are sent in the following format.

Cyclic DMX Channel Data sending (DMXface up to Rev.5.17 max. 224 channels)

If the cyclical sending of DMX data has been activated, the DMXface repeats the following message withing the interval set.

Transmitted data:

Byte 1	0xF0 (Dec. 240)	Start Byte (always 0xF0)
Byte 2	0xFF (Dec. 255)	Signature DMX OUT data
Byte 3	DMX channel 1	
Byte 4	DMX channel 2	
Byte 5	DMX channel 3	
….		
Byte n	DMX channel n=8 / 16 / 32 / 64 / 128 / all -depends from the numbers of channels choosen.	

Event: Infrared receive

DMXface encodes the received IR signal into a 64 bit pattern (8Byte) which will be transmitted as soon as an infrared signal has been received and the option “Infrared receive” is set with ACTS activated.

Transmitted data:

Byte 1	0xF0 (Dez. 240)	Start Byte
Byte 2	0x01 (Dez. 001)	Signature IR data
Byte 3	IR Byte 1	
Byte 4	IR Byte 2	
Byte 5	IR Byte 3	
Byte 6	IR Byte 4	
Byte 7	IR Byte 5	
Byte 8	IR Byte 6	
Byte 9	IR Byte 7	
Byte 10	IR Byte 8	

Total length 10 Bytes

Event: INPORT changed

DMXface has up to 24 local inports and another 32 bus inports. (56 inputs in total)
The state of all 56 inputs is sent as soon as one of the input states changes.
The 56 inputs are sent as 7 bytes, each bit representing one of the inputs.
Activate the checkbox “Inport Change” to release the output.

Transmitted data:

Byte 1	0xF0	Start Byte
Byte 2	0x02	Signature Inport change
Byte 3	0xVALUE	Inport BUS 4.1 - 4.8 / LSB = BUS Inport 4.1, MSB = BUS Inport 4.8
Byte 4	0xVALUE	Inport BUS 3.1 - 3.8 / LSB = BUS Inport 3.1, MSB = BUS Inport 3.8
Byte 5	0xVALUE	Inport BUS 2.1 - 2.8 / LSB = BUS Inport 2.1, MSB = BUS Inport 2.8
Byte 6	0xVALUE	Inport BUS 1.1 - 1.8 / LSB = BUS Inport 1.1, MSB = BUS Inport 1.8
Byte 7	0xVALUE	Inport 17-24 / LSB = INPORT 17, MSB = Inport 24
Byte 8	0xVALUE	Inport 9-16 / LSB = INPORT 16, MSB = Inport 9
Byte 9	0xVALUE	Inport 1-8 / LSB =INPORT 1, MSB = Inport 8

Event: OUTPORT changed

DMXface manages up to 24 OUTPORTS.
The state of all 24 OUTPORTS is sent as soon as one of the OUTPORT state changes.
The 24 outputs are sent as 3 bytes, each bit representing one of the OUTPORTS.
Activate the option “Outport Change” to receive the OUTPORT changes.
Transmitted data:

Byte 1	0xF0	Start Byte
Byte 2	0x04	Signature Outport change
Byte 3	0xVALUE	Outport 17-24 / LSB = Outport 17, MSB = Outport 24
Byte 4	0xVALUE	Outport 9-16 / LSB = Outport 9, MSB = Outport 16
Byte 5	0xVALUE	Outport 1-8 / LSB = Outport 1, MSB = Outport 8

Event: Trigger occured

If a TRIGGER programmed on the DMXface gets active, the DMXface will send a data package. “Trigger option” needs to be enabled in the ACTS Setup.
Transmitted data:

Byte 1	0xF0	Start Byte
Byte 2	0x08	Signature Trigger occured
Byte 3	0xWERT	Trigger number 0x01 bis 0x50 (Dec.080)

Event: RS232 data received at Serial Port 2

If data is received on the DMXface Serial Port 2, the host application is notified. Option “RS232” needs to be activated in the ACTS setup. (maximum data length 20 bytes)
Transmitted data:

Byte 1	0xF0	Start Byte
Byte 2	0x10	Signature RS232
Byte 3	0xVALUE	RS232 RX Byte 1
Byte 4	0xVALUE	RS232 RX Byte 2
….		
Byte n	0xVALUE	RS232 RX last Byte (max. 20 Bytes)

Event: MIDI Data received (Serial Port 2 with MIDI Module or DMXfaceXE)

If MIDI data is received on the DMXface, the host application will be notified. MIDI option in the ACTS setup needs to be activated.

Transmitted data:

Byte 1	0xF0	Start Byte
Byte 2	0x20	Signature MIDI
Byte 3	0xVALUE	MIDI Byte 1
Byte 4	0xVALUE	MIDI Byte 2
Byte 5	0xVALUE	MIDI Byte 3
Byte 6	0xVALUE	MIDI Byte 4

Event: DALI Data (Serial Port 2 with DALI Module)

If data is received on the DMXface DALI, the host application will be notified. Option DALI needs to be activated in the ACTS Setup.

Transmitted data:

Byte 1	0xF0	Start Byte
Byte 2	0x40	Signature DALI
Byte 3	0xVALUE	DALI ADDRESS Byte
Byte 4	0xVALUE	DALI DATA Byte

Event: Scene Call

If scenes are called, the host application will be notified. Option Scene call needs to be activated in the ACTS setup.

Transmitted data:

Byte 1	0xF0	Start Byte
Byte 2	0x80	Siganture Scene call
Byte 3	0xVALUE	Szene number 0x01 bis 0xB4 (Dec. 180)

After every transmission, the DMXface pauses for at least 20 msec. so the data can be recognized and processed by the recipient.

Note that depending from application, DMXface calls many scenes in very short times.
Every scene call is forwarded with a data package as long there is free capacity to send the message. Otherwise the data is lost.

Control and querying the DMXface with ACTS commands

For RS232 the maximum receivable length is 128 bytes.
It is necessary to keep a delay of 3ms between the transmitted datasets. After this time, the DMXface recognizes the RS232 record as complete, and begins processing.
Unrecognized commands are forwarded to the triggers.

With LAN, the maximum receivable length is 512 bytes, processing takes place as soon as the send packet arrives.
Unrecognized commands will not be forwarded, as there is the possibility to use the function “Trigger and Sequence” on an additional LAN socket.

Commands of the ACTS protocol:

Version request

Command:

Byte 1	0xF0	Start Byte
Byte 2	0x56 bzw. (‚V‘)	Signature Version request

Response:

Byte 1	0xF0	
Byte 2	0x56	Signature version request
Byte 3-n	String data	Version string about 100 Byte

Firmwarerevision covers 4 byte and starts with a leading string ‚Ver:‘
Example:
[0xF0], [0x56], „*DMXfaceXP Ver:5.16* RTC SP2MF XPWM R2X RS232 LANMOD …..“

Switching an OUTPORT or a bus port

Command:

Byte 1	0xF0	Start Byte	
Byte 2	0x4F or (‚O‘)	Signature outport set	
Byte 3	Outport / BUS	Number 0x01 bis 0x10	= OUTPORT 1-16
		Number 0x19-0x38	= BUS1-32
Byte 4	New value	0x00 = off, 0x01 to 0xFF = on	

OUTPORT numbers other than listed will be ignored

No response to this command.

Scene call

Command:

Byte 1	0xF0	Start Byte
Byte 2	0x53 or (‚S‘)	Signature scene call
Byte 3	Scene number	0x01 bis 0xB4 (Dez.180)

Scenenumbers other than 1-180 will be ignored.

No response to this command.

Scene call with alternative brightness, optional with alternativ fade time

Command:

Byte 1	0xF0	Start byte
Byte 2	0x53 or (‚S‘)	Signature scene call
Byte 3	0x00	Signature special command
Byte 4	0x43 or (‚C‘)	Command scene call
Byte 5	Scene number	0x01 to 0xB4 (Dez.180)
Byte 6	Brightness	0x00 (off) -0xFF (Full scene brightness)
(Byte 7)	Fade time	Optional byte to forward fadetime 0x00-0xFF

Other Scenenumbers than 1-180 will be ignored.
If a 7th byte is sent, this information overrides the fade time programmed in the scene called.
Fade times:

Value	Time
000	Immediately
001-100	0.1Sec. to 10 Sec. / 100mSec. resolution.
101-255	11 to 165 Sec. / 1 Sec. resolution.

No response to this command.

Update of a scene with current DMX values of DMX OUT

The structure of the scene (active DMX channels, Fade time, Timer setting, Outports,…) keeps unchanged.
All DMX channels that are activated in the scene programmed, will be updated with the current DMX value.
The entire scene is written back to flash if a change of the data is noticed.

Notice that DMXface has a protection mechanism, that prevents more than 200 write cylces to flash memory per hour occur. (Excessive flash write → Check manual, a flash memory has a limited number of rewrites)
Command:

Byte 1	0xF0	Start byte
Byte 2	0x53 or (‚S‘)	Signature scenecall
Byte 3	0x00	Signature special command
Byte 4	0x55 or (‚U‘)	Command scene update
Byte 5	Scene number	0x01 to 0xB4 (Dez.180)
Byte 6	0xAA	Security byte
Byte 7	0x55	Security byte

No response to this command.

Program call

Command:

Byte 1	0xF0	Start Byte
Byte 2	0x50 or (‚P‘)	Signature program call
Byte 3	Program number	0x01 bis 0x38 (Dez. 056)

Programumbers other than listed will be ignored.

No response to this command.

Set the value of one ore more DMX channels (Channel 1-544)

DMX channel is transferred as value 1-544 / 16 bit using byte 3 + 4.
Command:

Byte 1	0xF0	Start byte
Byte 2	0x44 or (‚D‘)	Signature DMX set
Byte 3	DMX Channel high	First DMX channel high byte 0x00 to 0x02
Byte 4	DMX Channel low	First DMX channel low byte 0x00 to 0xFF
Byte 5	DMX channel value	0x00 to 0xFF (Dec.255)
Byte 6	DMX channel+1 value	0x00 to 0xFF (Dec.255)
Byte 7	DMX channel+2 value	0x00 to 0xFF (Dec.255)
…		
Byte n	DMX channel+x value	0x00 to 0xFF (Dec.255)

DMX channel numbers other than 1-544 will be ignored.
No response to this command.

Set the DMX master value oft the DMX output (Available from DMXface Rev. 5.18)

The DMX master value controlles the DMX output, but no internal functions.
Command:

Byte 1	0xF0	Start Byte
Byte 2	0x4D bzw. (‚M‘)	Signature DMX master
Byte 3	0X57 bzw. (‚W‘)	Signature write
Byte 4	0x00-0xFF	New DMX master value

No response to this command.

Read the DMX master value (Available from DMXface Rev. 5.18)

Command:

Byte 1	0xF0	Start Byte
Byte 2	0x4D bzw. (‚M‘)	Signature DMX master
Byte 3	0X52 bzw. (‚R‘)	Signature read

Response:

Bytes			
01	02	03	04
0xF0	0x4D	0x52	DMX master value

DMX, INPORT, BUSport and CharBuffer request (includes analog value and text output optional converted by table)

Command:

Byte 1	0xF0	Start Byte
Byte 2	0x49 or (‚I‘)	Signature PORT request
Byte 3	Port number high	High byte portnumber see table below
	Port number low	Low byte portnumber see table below

Port number / ranges

PortH	PortL	Belongs to
0x00	1-8	Inport 1-8 (DMXface local)
0x00	9-24	Inport 9-24 (DMXface extension)
0x00	25-56	BUS Ports 1-32
0x00	0xE1-0xEC	CharBuffers 1-12
0x01	0x01-0xFF	DMX channel 1-255
0x02	0x00-0xFF	DMX channel 256-511
0x03	0x00-0x20	DMX channel 512-544

Other numbers than listed will be ignored.

Response:

Bytes

01	02	03+04	05	06	07-nn	nn+1
0xF0	0x49	PORT NR	DIGITAL Value	ANALOG Value	TEXT (min. 3 Bytes)	0x00

Meanings when requesting a CharBuffer

01	02	03+04	05	06	07-nn	nn+1
0xF0	0x49	PORT NR	Length	CSUM*	TEXT (min. 3 Bytes)	0x00

*CSUM is an EXOR sum of all characters and length information in the requested CharBuffer, it is helpful to determine a change of the string received easy.
Value is only zero whren CharBuffer is empty.

The text as of byte 7 covers at least 3 bytes + termination [0x00]. This part represents the (analog) value of the port decimal ASCII with a termination 0x00.

If the affected port is assigned to a table conversion, the text output is the result of the conversion. (Conversion + programmed unit string)
Example: Request INPORT 4: [0xF0], [0x49], [0x00], [0x04]

Response without table conversion with an analog value of Dec. 100
[0xF0], [0x49], [0x00],[0x04], [0x00], [0x64], [0x31], [0x30], [0x30], [0x00]
Port, Digital, Analog ‚1‘ ‚0‘ ‚0‘ Termination

DMX OUT request one ore more channels (Channel 1 to 544)

Command:

Byte 1	0xF0	Start byte
Byte 2	0x58 or (‚X‘)	Signature DMX OUT request
Byte 3	DMX OUT channel high	Start channel high byte, 16 bit value 1-544
Byte 4	DMX OUT channel low	Start channel low byte
Byte 5	Number of channels high	*Number of channels to be transferred, 16 bit value 1-544
Byte 6	Number of channels low	Number of channels low byte

*If the command is used with the serial port, the number of channels is limited to 255.

Response:

Bytes

01	02	03	04	05	06 …	n
0xF0	0x58	Start high	Start low	DMX value 1	DMX value 2	DMX value n

DMX IN request one or more channels (Channel 1 to 512)

Command:

Byte 1	0xF0	Start byte
Byte 2	0x59 or (‚Y‘)	Signature DMX IN request
Byte 3	DMX IN channel high	Channel high byte, 16 bit value 1-512
Byte 4	DMX IN channel low	Channel low byte
Byte 5	Number of channels high	*Number of channels to be transferred, 16 bit value 1-512
Byte 6	Number of channels low	Number of channels low Byte

*If the command is used with the serial port the number of channels is limited to 255.

Response:

Bytes

01	02	03	04	05	06 …	n
0xF0	0x59	Start high	Start low	DMX value 1	DMX value 2	DMX value n

OUT-, BUS-, INPORT, AD and DMX request with one command

This command requests all port states, 16x AD channel values and a selectable number of DMX OUT channel values with one command.

Using RS232 there is a limit of 224 requestable DMX channels.
Using network 512 channels can be transferred.

Command:

Byte 1	0xF0	Start byte
Byte 2	0x42 or (‚B‘)	Signature BULK READ
Byte 3	Number of DMX channels high	0x0000 bis 0x0200 (512)
Byte 4	Number of DMX channels low	DMX channel count to be transferred

Response:

Byte 1	0xF0	Start byte
Byte 2	0x42 bzw. (‚B‘)	Signature BULK READ
Byte 3	0x00	Fixed
Byte 4	Value	OUTPORT 17(LSB) – OUTPORT 24 (MSB)
Byte 5	Value	OUTPORT 9(LSB) – OUTPORT 16 (MSB)
Byte 6	Value	OUTPORT 1(LSB) – OUTPORT 8(MSB)
Byte 7	0x00	Fixed
Byte 8	0x00	Fixed
Byte 9	0x00	Fixed
Byte 10	0x00	Fixed
Byte 11	0x00	Fixed
Byte 12	Value	BUS 25 (LSB) –BUS 32 (MSB)
Byte 13	Value	BUS 17 (LSB) –BUS 24 (MSB)
Byte 14	Value	BUS 9 (LSB) – BUS 16 (MSB)
Byte 15	Value	BUS 1 (LSB) –BUS 8 (MSB)
Byte 16	Value	INPORT 17 (LSB) – INPORT 24 (MSB)
Byte 17	Value	INPORT 9 (LSB) – INPORT 16 (MSB)
Byte 18	Value	INPORT 1 (LSB) – INPORT 8 (MSB)
Byte 19	AD Value	AD value INPORT 1
Byte 20	AD Value	AD value INPORT 2
Byte 21	AD Value	AD Value INPORT 3
……		
Byte 34	AD Value	AD value INPORT 16

DMX channel values if requested by sending ((Byte 3<<8) + Byte4) >0

Byte 35	Value	DMX channel 1
Byte 36	Value	DMX channel 2
….		
Byte nn	Value	Last DMX channel requested

RTC Real time read and write

Read of the realtime clock

Command:

Byte 1	0xF0	Start byte
Byte 2	0x5A or (‚Z‘)	Signature RTC

Response:

Bytes			
01	0xF0		
02	0x5A (‚Z‘)		
03+04	ASCII	Hours ASCII format	0x30,0x31 → “01“
05	0x3A (‚:‘)	Delimiter	
06+07	ASCII	Minute ASCII format	0x30,0x31 → “01“
08	0x3A (‚:‘)	Delimiter	
09+10	ASCII	Seconds ASCII format	0x30,0x31 →“01“
11	0x2C (‚,‘)	Delimiter	
12+13	ASCII	Day of month ASCII format	0x30,0x31 → “01“
14	0x2E (‚.‘)	Delimiter	
15+16	ASCII	Month ASCII format	0x30,0x31 → “01“
17	0x2E (‚.‘)	Delimiter	
18+19	ASCII	Year ASCII format	0x31,0x39 → “19“
20	0x2C (‚,‘)	Delimiter	
21	ASCII	Day of week „1“ =Monday bis „7“ =Sunday	

Write a new value to the RTC

Format is the same as of received data. All byte position must be kept. An other length than 21 bytes is ignored.
You can use other characters for the delimiters, just the positions must be observed.

Bytes			
01	0xF0		
02	0x5A (‚Z‘)		
03+04	ASCII	Hour ASCII format	0x32,0x33 →“23“
05	0x3A (‚:‘)	Delimiter (any)	
06+07	ASCII	Minute ASCII format	0x35,0x39 → “59“
08	0x3A (‚:‘)	Delimiter (any)	
09+10	ASCII	Seconds ASCII format	0x35,0x39 →“59“
11	0x2C (‚,‘)	Delimiter (any)	
12+13	ASCII	Day of month ASCII format	0x33,0x31 → “31“
14	0x2E (‚.‘)	Delimiter (any)	
15+16	ASCII	Month ASCII format	0x31,0x32 → “12“
17	0x2E (‚.‘)	Delimiter (any)	
18+19	ASCII	Year ASCII format	0x32,0x39 → “29“
20	0x2C (‚,‘)	Delimiter (any)	
21	ASCII	Day of week „1“ =Monday bis „7“ =Sunday	

Response is the same as reading RTC (returns new value)