

DMX4ALL PC-Interface

Interface-Commands

- ENGLISH -

Overview

<u>COMMANDS BETWEEN PC AND DMX4ALL DMX-INTERFACE</u>	<u>3</u>
COM PORT CONNECTION.....	3
USB DRIVER CONNECTION	4
DESCRIPTION OF THE ASCII COMMANDS	5
SAMPLES ASCII COMMANDS	6
DESCRIPTION OF THE ARRAY TRANSFER.....	7

Commands between PC and DMX4ALL DMX-Interface

After connecting to the DMX Interface via a COM port connection or a direct USB driver connection you can use 2 different transfer modes. The ASCII commands or the array transfer.

COM port connection

The COM port connection is possible with the following DMX4ALL interfaces:

- USB-DMX-Interfaces with virtual COM port (FTDI VCP driver)
- RS232 connected interfaces (DMX Player “XS”)
- LAN-DMX-Interface with virtual COM port (Lantronix Com Port Redirector)

Open a COM port with the following parameters:

Baudrate: 38400 Baud (Some old interface can be switched to 19200 Baud)
Bytes: 8
Stop-Bits: 1
Parity: None
Handshake: None

After you open the connection and have the COM handle, you can use the standard transfer functions to write and read the data from the interface.

USB driver connection

The direct USB driver connection is possible with the following DMX4ALL interfaces:

- USB-DMX-Interfaces with FTDI D2XX driver or FTDI combined driver (Mini-USB-DMX-Interface/DMX Player “S”/USB-DMX STAGE-PROFI/...)

To communicate with the USB-Driver, you have to use the FTDI D2XX driver. More informations you can find in the “D2XX Programmer's Guide” from FTDI.

To send data to the interface, you have to open the device with FT_Open. After you got a handle to the device, you can send the data via FT_Write and read back data with FT_Read.

Example:

```
FT_HANDLE    ftHandle;
FT_STATUS    ftStatus;
DWORD        RxBytes, TxBytes;
DWORD        BytesReceived, BytesWritten;
char         RxBuffer[256], TxBuffer[256];

ftStatus = FT_Open(0,&ftHandle);
if(ftStatus != FT_OK)
    return;        // FT_Open failed

FT_SetBaudRate(ftHandle, FT_BAUD_38400);

TxBuffer = "C000L255";
TxBytes = 8;
ftStatus = FT_Write(ftHandle, TxBuffer,TxBytes,&BytesWritten);
if(ftStatus != FT_OK)
    return;        // FT_Write failed

RxBytes = 10;
ftStatus = FT_Read(ftHandle,RxBuffer,RxBytes,&BytesReceived);
if (ftStatus == FT_OK)
{
    // FT_Read OK
    // Check received chars
}else{
    // FT_Read Failed
}
FT_Close(ftHandle);
```

Description of the ASCII commands

Description	Transmit char	Receive char
Check connection	C?	G
Setup	S	<i>Setupmenu</i>
Get informations about the interface	I	<i>char-array with informations about the interface</i>
Set a DMX value	<i>CaaaLbbb</i> aaa = channel (000...511) bbb = value (000...255)	G
Read back a DMX value	<i>Caaa?</i> aaa = channel (000...511)	<i>bbbG</i> bbb = value (000...255)
Read a value of a DMX-IN port	<i>Vaaa?</i> aaa = channel (000...511)	<i>bbbG</i> bbb = value (000...255)
BlackOut enable	B1	G
BlackOut disable	B0	G
Request BlackOut status	B?	<i>aG</i> a = 0 → disable a = 1 → enable
Set number of DMX-OUT channels (temporary)	<i>Naaa</i> aaa = # of channel (000...511)	G
Read the number of DMX-OUT channels	N?	<i>aaaG</i> aaa = count (000...511)
Setzen der Anzahl der DMX-Ausgangskanäle und ablegen im EEPROM	<i>Eaaabbb</i> aaa = Startkanal (000...511) aaa = 512 Merge OFF bbb = Stopkanal (000...511)	G
Enable the merge function	<i>Faaa</i> aaa = channel (000...511)	G
Disable the merge function	F512	
Run a store light programm	<i>Raaa</i> aaa = programm (000...255)	G

Samples ASCII commands

Check connection to interface:

Transmit character: C?

Receive character: G

Set DMX channel 1 to value 255:

Transmit character: C000L255

Receive character: G

Set DMX channel 123 to value 47:

Transmit character: C123L047

Receive character: G

Read DMX-OUT buffer channel 1:

Transmit character: C000?

Receive character: 255G (actual value)

Read DMX-IN channel 1:

Transmit character: V000?

Receive character: 255G (actual value)

Read number of channels on DMX-IN

Transmit character: V?

Receive character: 512G (actual value)

Description of the array transfer

Description	Transmit	Receive
Write DMX-OUT	0xFF (Block transfer header) <i>start channel low byte</i> (0x00 ... 0xFF) <i>start channel high byte</i> (0x00 / 0x01) <i>number of channels</i> (0x00 ... 0xFF) <i>first data byte</i> (0x00 ... 0xFF) <i>last data byte</i>	G
Write DMX-OUT Channel 1-256 (Fast Mode)	0xE2 (Header) <i>channel low byte</i> (0x00 ... 0xFF) <i>data byte</i> (0x00 ... 0xFF)	-
Write DMX-OUT Channel 257-512 (Fast Mode)	0xE3 (Header) <i>channel low byte</i> (0x00 ... 0xFF) <i>data byte</i> (0x00 ... 0xFF)	-
Read back DMX-OUT	0xFE (Block transfer header) <i>start channel low byte</i> (0x00 ... 0xFF) <i>start channel high byte</i> (0x00 / 0x01) <i>number of channels</i> (0x00 ... 0xFF)	<i>first data byte</i> <i>last data byte</i> G
Read DMX-IN	0xFC (Block transfer header) <i>start channel low byte</i> (0x00 ... 0xFF) <i>start channel high byte</i> (0x00 / 0x01) <i>number of channels</i> (0x00 ... 0xFF)	<i>first data byte</i> <i>last data byte</i> G
Write to Stand-Alone-Memory	0xFD (Block transfer header) <i>address low</i> (0x00 ... 0xFF) <i>address high</i> (0x00 ... 0xFF) <i>data byte</i> (0x00 ... 0xFF) <i>checksum</i> (0xDF xor <i>address low</i> xor <i>address high</i> xor <i>data byte</i>)	G

Please note, that the array of the transmitted data are maximum 255 bytes long.
Also the sum of the start channel and the changes channels must be a maximum of 0xFF.

Example:

To set the channels 10-15 to the value 100,120,140,150,255,10, please transmit the following BYTES:

FF 09 00 06 64 78 8C 96 FF 0A (TX BYTES)

The interface send back the ASCII char „G“ if the command is OK and executed.



Dipl.-Ing. (FH) Markus Siwek
Sophienstr. 8
D-44791 Bochum
Germany

© Copyright 2008 Markus Siwek

All rights reserve. No part of this manual may be reproduced in any form (photocopy, pressure, microfilm or in another procedure) without written permission or processed, multiplied or spread using electronic systems.

All information contained in this manual was arranged with largest care and after best knowledge. Nevertheless errors are to be excluded not completely. For this reason I see myself compelled to point out that I can take over neither a warranty nor the legal responsibility or any adhesion for consequences, which decrease/go back to incorrect data. This document does not contain assured characteristics. The guidance and the characteristics can be changed at any time and without previous announcement.