

MIDI / DMX512 Interface

Status 30 JUN 2015

The intention of this project is to provide a replacement for the discontinued Cinetix "MIDI / DMX512 Generator" with minimized hardware requirement.

It is **NOT allowed** to use this instrument together with any safety critical applications, where malfunction could result in personal injury oder noticeable material damage !

Hardware

At one of the next pages a reference schematic and a corresponding PCB layout is shown. This is a single layer PCB which is compatible with a veroboard (2.54 mm raster of round copper pads). So it may be reproduced without etching a PCB.

The firmware is programmed to work with exactly this circuit !!

Nevertheless details of the peripheral circuits may be modified due to the user's need .

Though the DMX standard does not recommend an isolated DMX OUT, some users prefer this. A galvanically isolated DMX transmitter particularly makes sense, if one or more DMX transmitters without galvanically isolated DMX input are connected to the DMX bus.

By default, in the reference design DMX OUT is not isolated - but circuitry for optical isolation is provided. If isolation is preferred, three printed wires - marked with a "c" or "u" symbol at the PCB bottom side - have to be cut (2.5 mm drill recommended). One additional jumper wire has to be assembled, too.

Optical isolation of DMX OUT:

For this purpose, layout for an isolated DC/DC converter is provided (standard 4p SIL unregulated 1W model, available from different manufacturers).

Because the output voltage of these DC/DC converters may vary considerably due to different manufacturers, production lots and tolerance of 5V primary supply, sometimes the output voltage has to be adjusted.

For this purpose diode D3 is provided. The adjustment has to be done without installed microcontroller but with assembled IC4 and IC5. With the recommended Aimtec converter and installed diode D3, the voltage between TP2 and pin 8 of IC5 usually is about 4.9 to 5.4 Volt. If the voltage is below 4.9 volt, replace D3 with a jumper wire. In cases of very high voltage (isolated voltage with D3 installed >5,4V), a silicon diode like 1N4007 should be used instead of the Schottky 1N5819. At practical DMX operation, the isolated voltage is about 0.1 to 0.2 Volt lower than measured during adjustment.

The power supply is designed for unregulated or regulated **DC supplies with output voltages 8 to 28 Volt** and min. 4 Watt output capability.

The DC input CN1 is designed for a concentric low voltage connector, external 5,0 - 5,5mm, internal 2,1mm. The **positive polarity** has to be connected with the **inner contact!**

As an alternative, layout for a 2 terminal clamp is provided.

The reference circuit is **protected against wrong polarity**: If wrong, the device is not powered.

Primarily IC1 converts the input voltage into a regulated 5V supply. To provide an easy supply for external appliances like Ethernet or Wifi converters to RS-232, the 5V supply may be extracted from clamp block CN2 (max. 400mA). To combine the MIDI / DMX Interface with such a converter, its baud rate may be adjusted to 115200 and a RS-232 conformant SubD connector may be used in addition to the MIDI connectors.

The MIDI interface provides a standard MIDI IN and MIDI OUT.

For higher speed connection over an external Ethernet adaptor, optionally a specially wired female 9 pin SubD connector may be added. In the corresponding male plug, pins 9 and 5 have to be connected to resolve galvanic isolation only while the RS-232 cable is attached. If a standard RS-232 cable shall be used, alternatively R6 may be grounded by an external switch or jumper. For details, see schematic on page 3

Special Parts List:

All resistors are carbon or metal film types, min 0.25W, max 5% tolerance.

All capacitors should be multilayer ceramic, 5,08mm raster, 50V - except C2, C3: RM2.54mm

Electrolytic capacitor E1 should be a 35V type with 2.54 mm raster

IC1: least expensive P-78E5.0-0.5 (Recom, source RS Components), TSR-1-2450 (Traco Power, source Conrad or Reichelt) or P-785.0-0.5 (Recom)

IC2: ATmega1284P PU, 40p DIL case. Source Reichelt, RS-Components

IC3: PC900V, 6p DIL

IC4: 6N137,8p DIL

IC5: MAX487, 8p DIL or equivalent, for example MAX3085, MAX485, MAX483

DC/DC converter: 1W, 4p SIL case, AM1S 0505SZ (Aimtec, source Reichelt) or equivalent

LED: V-L-115-WEGW (source Conrad).

Other dual LED types may be used, but then values of R1 and R2 have to be adjusted

CN1: power connector (source Conrad Art.Nr.733980, Reichelt HEBW21 or similar)

CN3, CN4: MIDI standard 5p DIN sockets (180°)

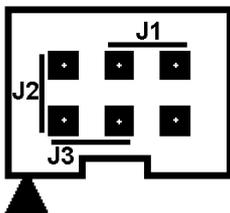
CN2, CN6: optional clamp bars, 5.00 or 5,08 mm raster

CN5: source Reichelt (PS 25/3G BR) or Conrad (Art.Nr.741221)

MIDI channel (and Baud rate) selection:

Some jumpers may be placed (not while programming) at the Atmel programming connector to make the interface sensitive to different MIDI channels.

A block of 4 MIDI channels (base channel plus 3 subsequent ones) must be reserved for complete control and cannot be used to control other MIDI devices in parallel. If the base channel is set to 16, only 1 MIDI channel is used for the price of somewhat reduced features. After change of jumper settings a power cycle is necessary.



no jumper (default) MIDI base channel = 1 (+2,3,4)

J1 only MIDI base channel = 13 (+14,15,16)

J2 only MIDI base channel is user selectable (default 16)

J3 only **Special baud rate 115200**, MIDI base channel user selectable (default 16)

J1 and J2 reserved for future development

J1 and J3 reserved for future development

LED color: Idle green. Flashes red when a valid MIDI Message (=command) is received.

Flashes dark (goes off) when any bytes are received but cannot be interpreted.

Layout of the DMX512 interface:

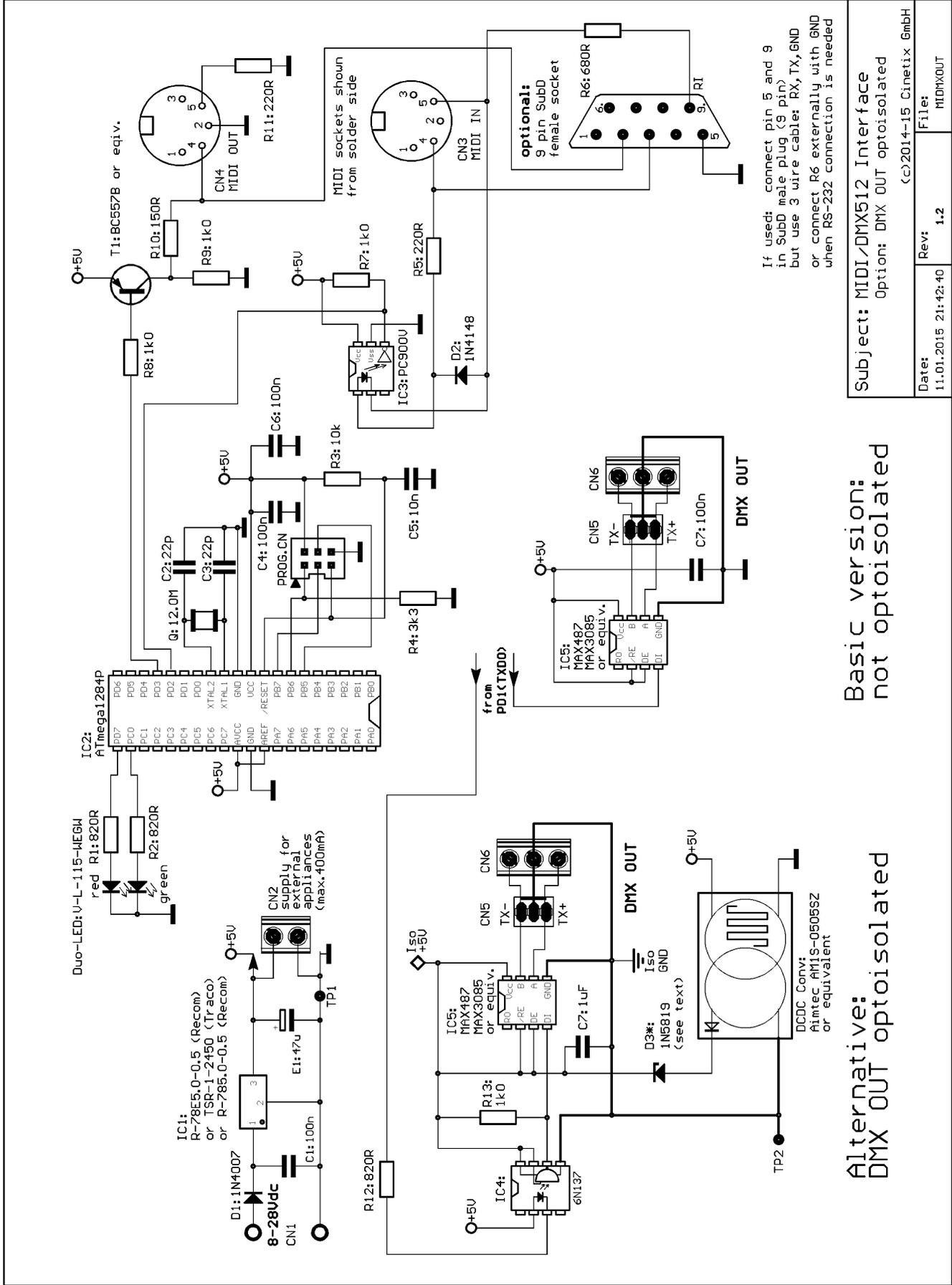
Two connectors are provided for DMX out: one to connect the XLR socket, one optionally for directly clamping the bus wires. Each of the DMX connectors CN5, CN6 has the local signal ground (isolated or not) at the middle pin. TX+ and TX- is marked at the bottom side of the PCB.

Important safety information:

The **purpose of galvanic isolation** of the DMX output is to eliminate DMX data errors due to moderate fault voltages which may result from long loops of signal ground lines and protective earth wires within the complete lighting installation.

**In conformance with the DMX standard, isolation is guaranteed
only up to 42 Volt.**

**It cannot provide an additional level of electrical safety against broken isolation and
faulty safety measures of the connected lighting equipment!**

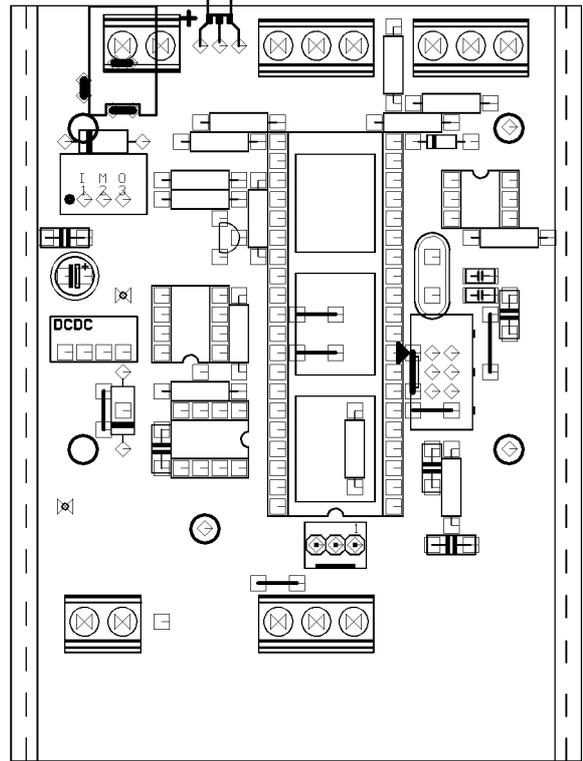
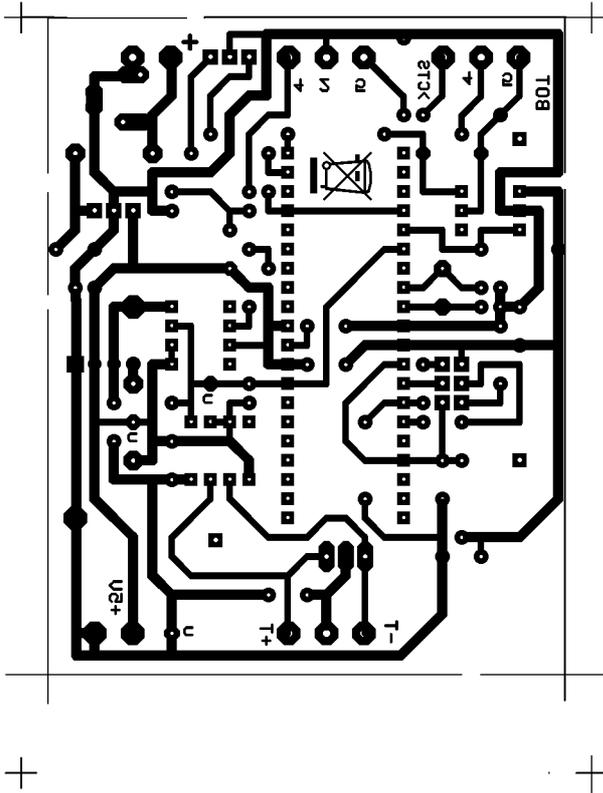


Basic version:
not optoisolated

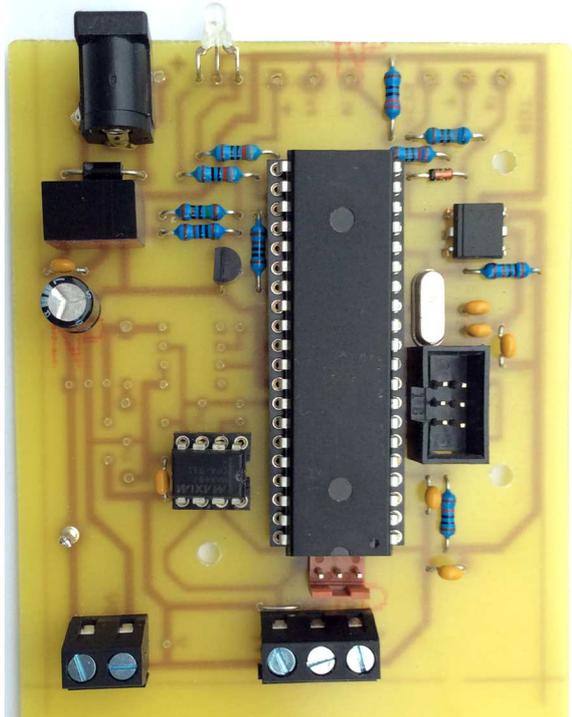
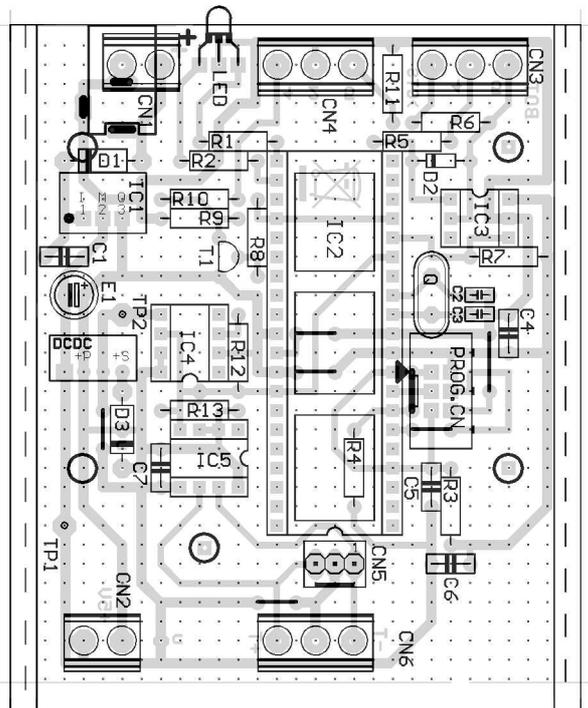
Alternative:
DMX OUT optoisolated

Subject: MIDI/DMX512 Interface
 Option: DMX OUT optoisolated

Date:	11.01.2015	21:42:40	Rev:	1.2	File:	MIDMXOUT
(c)2014-15 Cinetix GmbH						



Drill symbols: □ 0.8mm ◇ 1.0mm ⊗ 1.3mm △ 3.0mm



The PCB design fits into 3 easily available cabinet models:

low cost plastic standalone ("Eurobox" from Reichelt or Conrad: 52 31 32 --- fixed by 3 screws), rugged aluminium ("Fischer Elektronik FRAME" Reichelt: FR 80 42 100 ME --- PCB size 100x75 mm) and DIN rail compatible ("Camdenboss CNMB/4", Reichelt order code CB HUTKIT 4 --- PCB size 87x68 mm) There are 2 different outlines and 3 holes (3mm) at the PCB design for fitness to all versions.

Take care: some jumper wires are below other components, have to be assembled first!

Firmware

MIDI Channel Message protocol of the MIDI/DMX512 Interface

The content of MIDI messages cannot follow the conventions which have been established in the world of musicians (e.g. note values and controller numbers), but has been optimized for tasks of lighting control. We have taken much care to design the codes in a way they can be reproduced with common MIDI equipment like sequencers and programmable keyboards. Nevertheless, conflicts are possible, if the DMX Interface is used together with other applications on a shared MIDI port. These conflicts can be solved by appropriate selection of MIDI channels or by using a MIDI interface with multiple sets of independent MIDI IN and MIDI OUT.

General structure of MIDI messages: The **status byte** (here the type of command and MIDI channel is coded) is followed - depending on the MIDI message type - by up to 2 **data bytes**. In NOTE ON, NOTE OFF or POLY KEY PRESSURE messages, the **1st data byte** is the note value ("pitch", see table at the last page of the manual), the **2nd data byte** is the strength ("velocity") of the key stroke. With CONTROL CHANGE messages the 1st data byte addresses the controller number, the 2nd data byte is the value to be applied to this controller. PROGRAM CHANGE and CHANNEL PRESSURE messages have only a single data byte.

First is described how the MIDI / DMX Interface can be operated with simple commands for the most frequently used applications. Subsequently the complete set of MIDI commands is described in detail.

Quick start and basic commands:

One of the most frequently used applications is **lighting control based on a MIDI sequencer**, which in most cases is available as software running on a PC. In a simple application **the lighting control is programmed with NOTE ON messages on a sequencer**. Better precision is achieved by additional use of POLY KEY PRESSURE (=POLYPHONIC AFTERTOUCH) messages.

Depending on available equipment and personal linking the POLY KEY PRESSURE messages may be exchanged with CONTROL CHANGE messages. See detailed description below, especially PROGRAM CHANGE 62 and 63.

To address any of the DMX channels 1 to 127, control data have to be sent on the selected MIDI base channel (see page 2). How to access DMX channels 128 to 512 see below.

The 1st data byte of the MIDI command defines the DMX channel to be addressed.
The 2nd data byte of the command describes the DMX level (light intensity) to be set.
the 2nd MIDI data byte is multiplied by 2 inside the MIDI/DMX Interface.

Following **exceptions have to be taken into account:**

If a **POLY KEY PRESSURE** command was sent, the resulting 8 bit DMX level is additionally **incremented**,
i.e. **the next higher odd level is set**.

This behaviour can be deactivated (use a PROGRAM CHANGE command with data byte=60)
then for NOTE ON as well as for POLY KEY PRESSURE commands is valid:
if the 2nd MIDI data byte is equal to 127, the DMX level is set to its max. value 255

Or described in the opposite way of thinking: **to set a certain DMX level (0 to 255) with a simple MIDI command, HALF OF the intended DMX level has to be entered in the 2nd MIDI data byte**. Odd DMX levels are set with POLY KEY PRESSURE, even DMX levels are set with NOTE ON.

Example: When the MIDI base channel is 1, then DMX channel no. 1 is **set to DMX level decimal 64** (25% intensity) with this command: NOTE ON (status byte = hex90). 1st byte = 1, 2nd data byte =dez32 (hex20).

DMX level 65, however, is achieved with POLY KEY PRESSURE: (status byte = hexA0). 1st data byte = 1, 2nd data byte =dez32 (hex20).

To write data into DMX channels 128 to 512 with these simple commands, the MIDI commands are sent on a higher MIDI channel as described in the following table:

coded MIDI channel	1 st data byte	sets SLOT address to-	calculation of 1 st data byte
as selected base channel	1 to 127	1 to 127	= DMX channel
as base channel + 1	0 to 127	128 to 255	= DMX channel minus 128
as base channel + 2	0 to 127	256 to 382	= DMX channel minus 256
as base channel + 3	0 to 127	384 to 511	= DMX channel minus 384
as base channel	0 special case !	512	= 0

So, if the selected base channel is 1 (no jumpers at the programming connector), the MIDI / DMX Interface is sensitive for the MIDI channels 2,3 and 4, too.

This means: on a sequencer program you always have to **reserve a block of 4 MIDI channels** to fully control the MIDI / DMX Interface and the corresponding edit tracks have to be initialized. **Attention:** data which are meant for other MIDI equipment which works on these channels may be misinterpreted.

Example: MIDI base channel is 1. To set DMX channel no. 261 to full level (255) following MIDI message has to be sent: POLY KEY PRESSURE command at MIDI channel 3 (status byte = hexA2). First data byte = 5 (i.e. 261-256), second data byte = 127.

Differing from this rule, if the base channel is 15, the MIDI / DMX Interface reacts only on MIDI channels 15 and 16 and if the base channel is 16, it reacts only on MIDI channel 16. This is welcome in applications which use almost the complete set of MIDI channels for other appliances. With simple commands (NOTE ON or POLY KEY PRESSURE), however, only DMX channels 1 to 255 respectively 1 to 127 are addressable then.

This limitation can be worked around with somewhat more complex commands, which **allow to adjust any DMX channel 1 to 512 with full 8 bit accuracy**. See details at the description of the CONTROL CHANGE and the PITCH WHEEL CHANGE command below.

Some deepening annotations:

As an **alternative to the POLY KEY PRESSURE** command, which is less supported by some control media, odd DMX levels are obtained as follows: first set the lower even value with NOTE ON, then increase the level by one step with PROGRAM CHANGE 9. Another alternative is: first address the DMX channel with CONTROL CHANGE 80-83, then set the DMX level here with CONTROL CHANGE 84 or 84.

Many MIDI instruments - like keyboards and sequencers - automatically terminate a NOTE ON by a corresponding NOTE OFF message or by a NOTE ON message with velocity = 0. In the standard configuration of the DMX Interface this turns the DMX level and the attached lamp off immediately. This mode of operation is useful, if - e.g. - a keyboard shall be used as a "light organ" or similar actions are programmed on a sequencer.

In other applications, where NOTE ON messages simply shall be used to control steady lighting scenes, this behaviour is unpleasant.

The response of the MIDI/DMX Interface to NOTE ON messages with velocity=0 or to any NOTE OFF messages can be prevented with PROGRAM CHANGE command 121, details see below.

The advantage of setting DMX levels with NOTE ON messages is the possibility to "play in" the timing interactively and intuitively with a keyboard.

Get informed about the naming conventions of your sequencer. Unfortunately many sequencer programs don't offer the feature of entering the first data byte as a plain number, but only as a note name as used by musicians. **A transformation table is published in appendix C.**

Please note that **due to the parameters of the MIDI system the speed of data throughput is quite limited.** Per second about 1000 MIDI messages can be transferred at maximum.

If a lot of dimmers are faded by automatically generated sequences of NOTE ON or POLY KEY PRESSURE messages, the MIDI line can easily become clogged. Delayed execution of commands or even loss of some data cannot be excluded then.

To avoid this bottleneck, **the MIDI / DMX Interface is able to control a complete fade process automatically.** For this purpose, a nonzero value is written into the FADETIME register before sending a DMX level change command. Details see description of the corresponding CONTROL CHANGE commands.

Survey of all MIDI Channel Commands (MIDI Implementation Chart)

With PROGRAM CHANGE 62 the functions of **CONTROL CHANGE** and **POLY KEY PRESSURE** commands (as they are described in this manual) may be exchanged. This option provides better flexibility to work with different kind of MIDI control equipment.

Abbreviations:

DB means "data byte", DB1 means "1st data byte" (note value, controller number), DB2 means "2nd data byte" (velocity, controller value)

When using PROGRAM CHANGE commands you should take into account, that most MIDI devices and software send the data byte value "0" when "program no.1" is selected! **In the table "DB" denotes the physically transferred data byte.**

MIDI message and coded MIDI channel	special data values	function /effect	p.
NOTE OFF	see detailed description	DMX level --> 0. May be blocked with PROGRAM CHANGE 121	21
NOTE ON MIDI channel = base channel.+ next 3 channels	DB1 = DMX channel DB2 = DMX level ./ 2	set DMX channel and level (only 1 MIDI message / 7bit resolution) DMX level = DB2 * 2	10
MIDI channel = base ch.	see detailed description	load preset 0-226	20
POLY KEY PRESSURE MIDI channel = base channel.+ next 3 channels	DB1 = DMX channel DB2 = DMX level ./ 2+1	set DMX channel and level (only 1MIDI message / 7bit resolution) DMX level = DB2 * 2 +1	10
CONTROL CHANGE MIDI channel = base ch.	DB1 = 1 DB2 = eighth sec.0-127	set FADETIME 0 to 15.9 seconds fade time = DB2 ./8 (one 16th note= 1/8s at 120BPM)	14
notice PROGRAM CHANGE 62 !	DB1 = 2 DB2 = eighth sec.0-127	set FADETIME 16 to 31.9 seconds fade time = 16 + DB2 ./8 seconds (16th notes = 1/8s)	14
	DB1 = 3 DB2 = seconds 0-32	set FADETIME in whole seconds. DB2 = seconds (0 to 31,9 seconds)	14
	DB1 = 4 DB2 = quarter sec.0-127	set FADETIME in the range 0 to 31.8 seconds with a single controller. Fade time = DB2 ./4 seconds	14
	DB1 = 7 DB2 = masterfader %	set masterfader in the range 0 – 127 %	15
	DB1 = 8 DB2 = masterfader-100	set masterfader in the range 100-200 %	15
	DB1 = 15 (hex F) DB2 = duration in 1/10s	release flash pulse: all DMX channels are set to 100% during a short time. See detailed description	15
	DB1 = 16 (hex 10) DB2 = step in 1/8 sec (or 16th notes)	set step duration of the chaser. 0 = chaser OFF. Controlled by an internal timer. See Program Change 12(hexC) and 14 (hexE)	15
	DB1 = 17 (hex 11) DB2 = count of scenes per sequence	set sequence length of the chaser and start it: display any preset (scene) 'step duration' long, then next preset is loaded. STOP after 'count' steps	16
	DB1 = 18 (hex 12) DB2 = count of scenes per cycle	set cycle length of the chaser and start it: display any preset (scene) 'step duration' long, then next preset is loaded. Repeat cycle after 'count' steps	16
	DB1 = 19 (hex 13) DB2 = start preset	set start scene (preset no.) 0-127 of the chaser cycle. See detailed description	16
	DB1 = 20 (hex 14) DB2 = start-128 (0-98)	set start scene (preset no.) 128-226 of the chaser cycle. See detailed description	16
	DB1 = 21 (hex 15) DB2 = 1 -127	DB2 specifies a note value, which is used to forward the chaser by 1 step (with a MIDI Base Drum for example) instead of setting a DMX level	17

MIDI message and coded MIDI channel	special data values	function /effect	p.
CONTROL CHANGE MIDI channel = base ch.	DB1 =36-38(hex24-26) DB2 = 0 to 127	poll (entry of DB2) levels of DMX OUT starting from DMX channel=SLOT (DB2=0: poll 128 channels)	17
notice PROGRAM CHANGE 62 !	DB1 = 40 (hex 28)	fill a block of DB2 DMX channels starting from ch. SLOT+1 with the final level of DMX ch. "SLOT"	14
	DB1 =48-60 (hex30-3F) DB2 = 0 to 127	set DMX level at a RGB lamp bar without modification of the addressed DMX channel	12
	DB1 = 64,67,70,73,76 (hex 40,43,46,49,4C) DB2 = 0 to 127	set hue (color tone) of an RGB lamp (3 consecutive DMX channels) see detailed description !	12/ 13
	DB1 = 65,68,71,74,77 (hex 41,44,47,4A,4D) DB2 = 0 to 127	set color saturation of an RGB lamp i.e.add grey- or white component (3 consecutive DMX channels) see detailed description !	13
	DB1 = 66,69,72,75,78 (hex 42,45,48,4B,4E) DB2 = 0 to 127	set brightness/luminance of an RGB lamp (3 consecutive DMX channels) see detailed description !	13
	DB1 = 80 – 83 (hex 50 – 53)	address DMX channel(SLOT) using only one single MIDI channel. For loading data see "pitch change"	11
	DB1 = 84 (hex 54)	set DMX level (@SLOT) to DB2 *2 adjust DMX level with 8 bit resolution to even value	11
	DB1 = 85 (hex 55)	set DMX level (@SLOT)to DB2 *2 + 1 adjust DMX level with 8 bit resolution to odd value	11
	DB1 = 86 (hex 56)	First increase DMX channel (SLOT reg.) there set DMX level (@SLOT)to DB2 *2. adjust DMX level with 8 bit resolution to even value	11
	DB1 = 87 (hex 57)	First increase DMX channel (SLOT reg.) there set DMX level (@SLOT)to DB2 *2 + 1 adjust DMX level with 8 bit resolution to odd value	11
	DB1 = 96 (hex 60) DB2 = preset no.	load preset no. 0 – 127. The system configuration is exclusively loaded with preset no. 0 - 3	20
	DB1 = 97 (hex 61) DB2 = preset no. - 128	load preset no. 128 – 226	20
	DB1 = 98 (hex 62) DB2 = preset no.	load preset no. 0–127 partially. Preset DMX chann. 129-256 are loaded into DMX OUT channels 1-128	20
	DB1 = 99 (hex 63) DB2 = preset no.	load preset no. 0–127 partially. Preset DMX chann. 257-384 are loaded into DMX OUT channels 1-128	20
	DB1 = 100 (hex 64) DB2 = preset no.	load preset no. 0–127 partially. Preset DMX chann. 385-512 are loaded into DMX OUT channels 1-128	20
	DB1 = 101 (hex 5) DB2 = preset no. - 128	load preset no. 128–226 partially. Preset DMX ch. 129-256 are loaded into DMX OUT channels 1-128	20
	DB1 = 102 (hex 66) DB2 = preset no. - 128	load preset no. 128–226 partially. Preset DMX ch. 257-384 are loaded into DMX OUT channels 1-128	20
	DB1 = 103 (hex 67) DB2 = preset no. - 128	load preset no. 128–226 partially. Preset DMX ch. 385-512 are loaded into DMX OUT channels 1-128	20
	DB1 = 104 (hex 68) DB2 = preset no.	load preset no. 0–127 partially. Only DMX channels 1-128 are loaded, 129-512 remain unchanged	20
	DB1 = 105 (hex 69) DB2 = preset no.	load preset no. 0–127 partially. Only DMX channels 129-256 are loaded, others remain unchanged	20
	DB1 = 106 (hex 6A) DB2 = preset no.	load preset no. 0–127 partially. Only DMX channels 257-384 are loaded, others remain unchanged	20
	DB1 = 107 (hex 6B) DB2 = preset no.	load preset no. 0–127 partially. Only DMX channels 385-512 are loaded, others remain unchanged	20
	DB1 = 112 (hex 70) DB2 = preset no.	save preset no. 0 - 127. The system configuration is exclusively saved with preset no. 0 - 3	19
	DB1 = 113 (hex 71) DB2 = preset no. - 128	save preset no. 128 - 226	19

MIDI message and coded MIDI channel	special data values	function /effect	p.
CONTROL CHANGE MIDI channel = base ch.	DB1 = 114 (hex 72) DB2 = preset no.	save preset no 0–127 partially. DMX OUT channels 1-128 are copied into preset channels 129-256	19
notice PROGRAM CHANGE 62 !	DB1 = 115 (hex 73) DB2 = preset no.	save preset no 0–127 partially. DMX OUT channels 1-128 are copied into preset channels 257-384	19
	DB1 = 116 (hex 74) DB2 = preset no.	save preset no 0–127 partially. DMX OUT channels 1-128 are copied into preset channels 385-512	19
	DB1 = 117 (hex 75) DB2 = preset no. - 128	save preset no 128-226 partially. DMX OUT chann. 1-128 are copied into preset channels 129-256	19
	DB1 = 118 (hex 76) DB2 = preset no. - 128	save preset no 128-226 partially. DMX OUT chann. 1-128 are copied into preset channels 257-384	19
	DB1 = 119 (hex 77) DB2 = preset no. - 128	save preset no 128-226 partially. DMX OUT chann. 1-128 are copied into preset channels 385-512	19
	DB1 = 120 (hex 78) DB2 = only15(hex F)	Store actual lighting scene as a pattern which is loaded during the flash effect	20
	DB1 = 126 (hex 7E) DB2 = see description	fine adjustment of DMX timing	22
	DB1 = 127 (hex 7F)	DB2 has the same meaning as the data byte of the corresponding PROGRAM CHANGE command.	
PROGRAM CHANGE MIDI channel = base ch.	DB = 8	decrease DMX level (minus 1) at channel "SLOT"	14
	DB = 9	increase DMX level (plus 1) at channel "SLOT"	14
	DB = 16 (hex 10)	forward chaser immediately by 1 step	16
	DB = 34 (hex 22)	block execution of MIDI channel msg. commands	18
	DB = 35 (hex 23)	execute last received command but keep blocking	19
	DB = 36 (hex 24)	stop command blocking - don't execute anything	19
	DB = 37 (hex 25)	stop command blocking and execute last received	19
	DB = 56 (hex 38)	No Running State allowed in commands	18
	DB = 57 (hex 39)	accept Running State in commands (default *)	18
	DB = 60 (hex 3C)	POLY KEY PRESSURE behaves like NOTE ON Controller value 127 is transformed to DMX 255	18
	DB = 61 (hex 3D)	POLY KEY PRESSURE sets DMX level +1 (default)	18
	DB = 62 (hex 3E)	exchange functions of CONTROL CHANGE and POLY KEY PRESSURE	18
	DB = 63 (hex 3F)	don't exchange CONTROL CHANGE and POLY KEY PRESSURE (default)	18
	DB = 64 (hex 40)	set LOOP=SLOT	17
	DB = 72 (hex 48)	Stop all fade processes, freeze at momentary level	14
	DB = 73 (hex 49)	Stop fade process exclusively at actually selected DMX channel, freeze at momentary level	14
	DB = 96 (hex 60)	NOTE ON 0-127 (keyboard) loads preset no 0-127	21
	DB = 97 (hex 61)	NOTE ON 0-127 loads preset no 128-226	21
	DB = 98 (hex 62)	NOTE ON 0-127 loads preset no 0-127 partially See CONTROL CHANGE DB1=98	21
	DB = 99 (hex 63)	NOTE ON 0-127 loads preset no 0-127 partially Siehe CONTROL CHANGE DB1=99	21
	DB = 100 (hex 64)	NOTE ON 0-127 loads preset no 0-127 partially See CONTROL CHANGE DB1=100	21
	DB = 101 (hex 65)	NOTE ON 0-127 loads preset no 128-226 partially See CONTROL CHANGE DB1=101	21
	DB = 102 (hex 66)	NOTE ON 0-127 loads preset no 128-226 partially See CONTROL CHANGE DB1=102	21
	DB = 103 (hex 67)	NOTE ON 0-127 loads preset no 128-226 partially See CONTROL CHANGE DB1=103	21
	DB = 104 (hex 68)	NOTE ON 0-127 loads preset no 0-127 partially See CONTROL CHANGE DB1=104	21

MIDI message and coded MIDI channel	special data values	function /effect	p.
PROGRAM CHANGE MIDI channel = base ch.	DB = 105 (hex 69)	NOTE ON 0-127 loads preset no 0-127 partially See CONTROL CHANGE DB1=105	21
	DB = 106 (hex 6A)	NOTE ON 0-127 lädt Preset Nr. 0-127 partially See CONTROL CHANGE DB1=106	21
	DB = 107 (hex 6B)	NOTE ON 0-127 loads preset no 0-127 partially See CONTROL CHANGE DB1=107	21
	DB = 120 (hex78)	NOTE ON sets the level of a DMX channel Velocity=0 and NOTE OFF are accepted (default)	21
	DB = 121 (hex79)	NOTE ON sets the level of a DMX channel Velocity=0 is ignored	21
	DB = 122 (hex7A)	ask actual channel configuration (get SysEx message as response)	21
	DB = 127 (hex7F)	Clear All Memory	21
CHANNEL PRESSURE MIDI channel = base ch.	DB= DMX level ./2	increase adressed DMX channel and set there DMX level = DB*2.	12
CHANNEL PRESSURE MIDI ch.= base channel +1	DB= DMX level ./ 2	increase adressed DMX channel and set there DMX level = DB*2 +1.	11
PITCH CHANGE MIDI channel = base ch.	DB1 = less than 64 ? DB2 = DMX level ./ 2	set DMX level at position SLOT (8bit resolution) If DB1 >= 64, then DMX level = DB2*2 + 1	11
SYSTEM EXCLUSIVE		Special: set user defined MIDI base channel	22
MIDI RESET		Reset, loads preset no.0	22

Express setting of DMX channel (= "SLOT") and DMX transmit level
with a single MIDI message:

NOTE ON or POLY KEY PRESSURE (=POLYPHONIC AFTERTOUCH)

The first MIDI data byte (controller number or note value/pitch) **sets the DMX channel.**
The 2nd data byte (controller value / note velocity) **describes the light intensity to be set.**
The DMX level is **twice of** the second MIDI databyte.
Or vice versa: the **the 2nd MIDI data byte is HALF OF the intended DMX level (0 to 255).**

Following exceptions do apply:
If the status byte is a POLY KEY PRESSURE,
then the DMX level is incremented by one, i.e. the next higher odd DMX level is sent.
This behaviour may be **deactivated** with a PROGRAM CHANGE (data byte = 60).
Exclusively then POLY KEY PRESSURE behaves like NOTE ON:
if the 2nd. data byte is equal to 127, the DMX level is set to 255.

This simple method works only for DMX channels 1 to 127.

To address DMX slots 128 to 512, the MIDI / DMX Interface expects control commands on a higher MIDI channel corresponding with following table (only valid for MIDI base channels 1 - 13):

coded MIDI channel	1 st data byte	addresses DMX channel-	calculation of 1 st data byte
as selected base channel	1 to 127	1 to 127	= DMX channel
as base channel + 1	0 to 127	128 to 255	= DMX channel minus 128
as base channel + 2	0 to 127	256 to 382	= DMX channel minus 256
as base channel + 3	0 to 127	384 to 511	= DMX channel minus 384
as base channel	0 special case !	512	= 0

Special reaction on NOTE ON messages with velocity=0 and on NOTE OFF messages see PROGRAM CHANGE commands 120 and 121.

Address the active DMX channel (= "SLOT" register) with:

CONTROL CHANGE

This alternative command version is used, **when all MIDI messages shall be sent on the MIDI channel = selected base channel.**

The coarse range is selected by the first data byte, which has to be chosen according to this table:

1st data byte	addresses DMX chan.= SLOT	2nd data byte	calculation of 2nd data byte
80 (hex50)	1 to 127	1 to 127	= DMX channel
81 (hex51)	128 to 255	0 to 127	= DMX channel minus 128
82 (hex52)	256 to 382	0 to 127	= DMX channel minus 256
83 (hex53)	384 to 511	0 to 127	= DMX channel minus 384
80 (hex50)	512	0 (special case!)	= 0

Comment: This command does not directly trigger any action. But the updated content of the SLOT register will be executed together with subsequent commands. In DMX slang a DMX channel is called a "SLOT" (physically it denotes a time slot in the DMX transmit cycle, therefore this strange name)

Set DMX level with 8 bit resolution at DMX channel = SLOT with:

Method 1:

PITCH WHEEL CHANGE

MIDI channel as selected base channel

1st data byte: if equal or greater than 64, "1" is added to the DMX level
if less than 64 (hex40), nothing is added to the DMX level

2nd data byte: desired DMX level divided by 2

inside the MIDI / DMX Interface, this data is multiplied by 2 before it is written into the DMX transmit buffer

Comment: This coding scheme looks strange at first glance. But it is compatible with the standard MIDI method to put the 7 "most significant bits" of 14bit data into the second data byte zu. So it can be used together with simple MIDI equipment, which has only 7 bit capability of PITCH WHEEL CHANGE operation.

Method 2:

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 84 (hex54) adjusts to an **even** DMX level

2nd data byte: desired DMX level divided by 2

i.e vice versa: DMX level= 2nd data byte * 2

1st data byte = 85 (hex55) adjusts to **next odd** DMX level

2nd data byte: desired DMX level divided by 2

i.e vice versa: DMX level= 2nd data byte * 2 **plus 1**

1st data byte = 86 (hex56) **first increases the addressed DMX channel**
and adjusts this one to an **even** DMX level

2nd data byte: desired DMX level divided by 2

i.e vice versa: DMX level= 2nd data byte * 2

1st data byte = 87 (hex57) **first increases the addressed DMX channel**
and adjusts this one to **next odd** DMX level

2nd data byte: desired DMX level divided by 2

i.e vice versa: DMX level= 2nd data byte * 2 **plus 1**

Increase addressed DMX channel ("SLOT") and set DMX level with:

CHANNEL PRESSURE

MIDI channel as base channel: DMX level = data byte*2

MIDI channel as base channel +1: DMX level = data byte*2 +1

Comment: with this method a block of consecutive DMX channels can be set with different levels in a quite compact way.

To set the start channel, the first DMX channel is addressed with NOTE ON, POLY KEY PRESSURE or CONTROL CHANGE, all following ones with a CHANNEL PRESSURE command.

Running State is accepted in correspondence with the general system configuration. If it is sufficient to set the DMX levels to the nearest even value, any number of DMX channels can be set without repetition of the status byte.

Set DMX level at a 4 RGB lamp bar with :

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 48 (hex 30): set level at the addressed DMX channel
= 49 (hex 31): set level at the addressed DMX channel +1
= 50 (hex 32): set level at the addressed DMX channel +2
= 51 (hex 33): set level at the addressed DMX channel +3
= 52 (hex 34): set level at the addressed DMX channel +4
= 53 (hex 35): set level at the addressed DMX channel +5
= 54 (hex 36): set level at the addressed DMX channel +6
= 55 (hex 37): set level at the addressed DMX channel +7
= 56 (hex 38): set level at the addressed DMX channel +8
= 57 (hex 39): set level at the addressed DMX channel +9
= 58 (hex 3A): set level at the addressed DMX channel +10
= 59 (hex 3B): set level at the addressed DMX channel +11
= 60 (hex 3C): set level at the addressed DMX channel +12
= 61 (hex 3D): set level at the addressed DMX channel +13
= 62 (hex 3E): set level at the addressed DMX channel +14
= 63 (hex 3F): set level at the addressed DMX channel +15

2nd databyte = DMX level 0 - 127 (internally multiplied by 2.

Controller value 127 results in DMX level 255)

Comment: The command to controller number 48 influences the actually addressed DMX channel (actual entry to the SLOT register set first by a NOTE ON, CONTROL CHANGE or POLY KEY PRESSURE message). The commands to controller numbers 49 to 63 influence the next higher DMX channels.

This is especially practical for small installations which use bars of RGB LED lamps, because no readdressing of the first DMX channel is required. All features else of a complex lamp ("fixture") may be used independently.

If set, the fade time also gets applied in combination with these commands.

Set hue (color tone) of a RGB lamp with :

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 64 (hex 40): set hue at the addressed DMX channel + next 2

- = 67 (hex 43): set hue at the addressed DMX channel **+3+** next 2
- = 70 (hex 46): set hue at the addressed DMX channel **+6** + next 2
- = 73 (hex 49): set hue at the addressed DMX channel **+9** + next 2
- = 76 (hex 4C): set hue at the addressed DMX channel **+12** + next 2

2nd databyte = hue (color tone) 0 - 127

This will approximately result in following colors. Intermediate hue values will result in intermediate colors:

2nd databyte= 0:red, 22:yellow, 43:green, 64:cyan, 85:blue, 106:magenta, 127:red again.

In correspondence with the model of the driven lamp and setting of saturation and brightness the resulting color tone may differ somewhat.

Comment: The command to controller number 64 influences the actually addressed DMX channel (actual entry to the SLOT register set first by a NOTE ON, CONTROL CHANGE or POLY KEY PRESSURE message) and the two next higher DMX channels. It is provided that the RGB setting of the respective lamp is done on these 3 subsequent DMX channels. All features else of a complex lamp ("fixture") may be used independently.

The commands to controller numbers 67,70,73 influence the next higher **triples** of DMX channels. **This is especially practical for small installations which use bars of 4 RGB LED lamps**, because no readdressing of the first DMX channel is required.

If set, the fade time gets also applied in combination with these commands. But the fade transition from the previous color tone to the new one is performed along a straight line through the color space, not along the color circle. So, if is faded between very different colors, disagreeable intermediate color tones may appear. To get a perfect color transition, up tp 6 subsequent fade steps between neighboured colors have to be performed. The technical handling can be simplified by use of the chaser effect.

Set color saturation of a RGB lamp with :

CONTROL CHANGE

MIDI channel as selected base channel

- 1st data byte = 65 (hex 41): set saturation at the addressed DMX channel + next 2
- = 68 (hex 44): set saturation at the addressed DMX channel **+3+** next 2
- = 71 (hex 47): set saturation at the addressed DMX channel **+6** + next 2
- = 74 (hex 4A): set saturation at the addressed DMX channel **+9** + next 2
- = 77 (hex 4D): set saturation at the addressed DMX channel **+12** +next 2

2nd databyte = color saturation 0 - 127

Comment: "Saturation" describes the amount of white or grey in a color tone (pastel shade). Saturation=127 gets a pure color, saturation=0 gets white or grey without specific color tone.

Set brightness/luminance of a RGB lamp with :

CONTROL CHANGE

MIDI channel as selected base channel

- 1st data byte = 66 (hex 42): set luminance at the addressed DMX channel + next 2
- = 69 (hex 45): set luminance at the addressed DMX channel **+3+** next 2
- = 72 (hex 48): set luminance at the addressed DMX channel **+6** + next 2
- = 75 (hex 4B): set luminance at the addressed DMX channel **+9** + next 2
- = 78 (hex 4E): set luminance at the addressed DMX channel **+12** +next 2

2nd databyte = brightness/luminance 0 - 127

Fill block of DMX channels starting from "SLOT+1" with the final level of DMX channel "SLOT" with:

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 40 (hex28)

2nd data byte: block length (1 to 127)

Comment: Every DMX channel in the commanded range is faded or switched from its present level to the final state of the DMX channel which is preselected by the SLOT register. The fade duration is given by the actual setting of the FADETIME register. With the equivalent SysEX/ASCII command, a block of up to all 512 DMX channels can be filled with a single command.

Simple modifications of the DMX level with:

PROGRAM CHANGE

MIDI channel as selected base channel

data byte = 8 decrease (subtract 1 from) the level of DMX channel "SLOT"

data byte = 9 increase (add 1 to) the level of DMX channel "SLOT"

Comment: With these commands the disadvantage of lower accuracy of 7 bit MIDI data can be compensated. Furthermore it is useful to perform extremely slow fade transitions. If a fade process is active at the addressed DMX channel, only the final DMX level is decreased or increased

data byte = 72 (hex48) **Stop all fade processes** immediately.

Freeze all DMX levels at their present state.

data byte = 73 (hex49) **Stop fade process on the actually addressed**

DMX channel immediately and freeze it at its present state.

Set FADETIME with:

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 1, then

2nd data byte = 0 – 127: fade time in 1/8 second (/16 note) units
(setting range 0 – 15.7 seconds).

1st data byte = 2, then

2nd data byte = 0 – 127: fade time in 1/8 second (/16 note) units
plus 16 seconds (setting range 16 – 31.7 seconds).

1st data byte = 3, then

2nd data byte = 0 – 32: fade time in **whole seconds** (0 – 31.7 sec.)
if the 2nd data byte = 32, it is internally limited to 31.7

1st data byte = 4, then

2nd data byte = 0 – 127: fade time in **1/4 second (/8 note) units** (0 – 31.6 sec.)

This variant makes it possible to handle the complete range of fading time with a single MIDI controller.

Comment: the raster of eighth seconds provides smooth fitness to musical structures. At 120BPM a 16teenth note (semiquaver) exactly takes 1/8 second. The digit following the dot shows the number of eighths of seconds, which is added to the full seconds.

The actual value of FADETIME is copied into the respective resource when the fade process is started. Immediately after then FADETIME can be modified without retroactivity on running fade processes. Any number of fade processes can be active simultaneously.

Set the MASTERFADER with:

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 7

then 2nd data byte = 0-127 (hex 7F) masterfader setting in %

1st data byte = 8

then 2nd data byte = 0-100 (hex 64) masterfader setting 100-200%
(internally 100 is added to the data byte)

Comment: The masterfader works like a digital signal processor when the **transmit buffer is written into the DMX transmitter hardware**. It is useful for global adjustment of lighting scenes. **It does not change or influence any internal buffer of the DMX interface.**

Trigger "flash" effect with :

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 15 (hex F)

2nd data byte = flash duration in 1/8 s (/16 note) units (0,1 to 15.9 seconds)

all DMX channels are pulsed to 100%

= 0 terminate flash immediately, independently from a previously entered duration

Comment: with this command a special preset (=special lighting scene) is transmitted at all 512 channels of the DMX bus during a time period given by the 2nd data byte. After this timing period, all previous DMX levels are restored.

At delivery the flash lighting scene is prestored in a way that all DMX channels are pushed to 100% level. Used together with installation of complex lamp fixtures, this may result in undesirable complications (start stroboscope effect, for example).

To avoid this, a user prepared lighting scene may be stored permanently in a special memory segment with a CONTROL CHANGE command 1st data byte 126 (hex 7E), 2nd data byte =15 (hex F). This lighting scene should be designed in a way to avoid these side effects - or even a very individual flash pattern may be created.

Set chaser step duration with :

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 16 (hex 10)

2nd data byte = (1 - 127) chaser step duration in 1/8 second (/16 note) units

or = 0 switches the chaser **OFF**

at delivery, a default step duration of 2 seconds is active
to provide an easy start

Comment: . After the duration of any chaser step is over, the chaser automatically loads a part of the next preset in the cycle: **stored DMX levels from DMX channels 385 to 512 are copied to and transmitted at the DMX bus channels 1 to 128.**

Set chaser cycle length and start it with :

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 17 (hex11) the chaser sequence **is passed through only once, then the chaser stops**

1st data byte = 18 (hex12) the chaser **repeats the cycle permanently**

2nd data byte = 2 - 127 (hex 7F) set chaser cycle length 2-127

or = 0: switches the chaser OFF

Comment: before the chaser can be started, the **step duration** (CONTROL CHANGE, 1st data byte= 16 or 17) as well as the **start scene** (CONTROL CHANGE 1st data byte= 19 or 20) has to be adjusted – **Details see description of these commands**. For an easy start, start at scene 128 and step duration 16 (= 2 seconds) is preset as default.

All settings of the chaser are stored in presets 0 to 3 and reactivated when any of these presets is loaded. This applies especially for preset no.0, which is loaded when the DMX Interface is started. The actual settings of fade time and master fader are taken over by the chaser.

The chaser feature works as follows: a sequence of lighting scenes is loaded in a cyclic manner to DMX channels 1 to 128. To maintain this, the DMX levels which are stored in presets no. 0 to 226 for **DMX channels 385 to 512 are exclusively copied to the DMX transmit buffer channels 1 to 128** and transmitted on the DMX bus (**see commands for partial saving and loading of presets** below !). Any other DMX channels 129 to 512 are not influenced by the chaser and may be operated independently.

As soon as <cycle length> presets are loaded in sequence, the procedure repeats from <chaser start>.

Example: if the chaser cycle is set to 4 and the chaser start is set to 128, then presets no. 128,129,130,131 are loaded partially, then preset no. 128 again and so on

Set chaser start preset (lighting scene) with :

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 19 (hex13) ,

2nd data byte: **offset = start preset (0 - 127)**

alternatively the chaser cycle may start with a preset in the range 128-226:

1st data byte = 20 (hex14)

2nd data byte: **offset = start preset minus 128 (0 - 98)**

Comment: If start preset no.128 (Controller Number=20, Controller Value=0) is chosen (default setting at delivery) the chaser cycle always starts with loading preset no. 128. **With an offset differing from 0 the chaser cycle starts with preset no. (128 + offset).** If the chaser would access a preset beyond 226, the sequence continues with loading preset no. 0 etc..

By use of this method, a big number of individual chaser effects may be created and run easily and flexible. This arrangement is optimized for lighting installations, which use a maximum of 384 DMX channels in normal operation and whose chaser effects are concentrated on DMX channels 1 to 128. Experience has shown that this is the case for stage lighting of most music bands and small theaters.

Forward chaser immediately (asynchronously) by 1 step with :

PROGRAM CHANGE

MIDI channel as selected base channel

data byte = 16 (hex 10) forward chaser immediately (asynchronously) by 1 step

or activation of a special feature with

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 21 (hex15) ,

2nd data byte = 1 – 126: this note value is reserved to forward the chaser
= 127 deactivates the feature

Comment: The controller value 1-126 of this command specifies, by which MIDI NOTE ON value (pitch) 1-126 not the DMX level shall be set but instead the chaser shall be forwarded by one step. Velocity of the NOTE ON command does not matter, only velocity=0 is ignored. **May be used to lock the chaser to a GM Drum and Percussion Sets.**

When this feature is applied, care has to be taken that at DMX channel = activated note value no control of a DMX level is possible by NOTE ON commands.

This setting is not stored in presets and has to be reactivated after every power on reset (it is recommended to send this command at the beginning of a sequencer track).

Poll DMX transmit buffer at DMX channel ="SLOT" and subsequent ones with :

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte: desired MIDI format of messages

= **36** (hex 24) response: sequence of MIDI NOTE ON or POLY KEY PRESSURE
Message format corresponds with "Express Setting" described above.

= **37** (hex 25) response: first a CONTROL CHANGE (no of the 1st DMX channel)
followed by further CONTROL CHANGE messages (successive DMX levels).

= **38** (hex 26) response: sequence of MIDI NOTE ON (or CONTROL CHANGE messages if
PROG.CHANGE 62 is active) . The DMX level is reported only in 7 bit format.

2nd data byte = (0 bis 127) count of DMX channels to be polled.

= 0: poll 128 DMX channels

Comment: If the **1st data byte = 36**, a sequence of **NOTE ON and POLY KEY PRESSURE** messages is returned, **their format is identical to the short format used to set a DMX level.** When this method is used, every single MIDI message contains the corresponding DMX channel, but possibly different MIDI channels are contained in a block of messages.

However, if the **1st data byte = 37**: first a **CONTROL CHANGE** message is returned which contains **the number of the first DMX channel to be read out**, followed by a CONTROL CHANGE with the 1st data byte = 84 or 85 and the DMX level of this channel. Next follows a **sequence of CONTROL CHANGE** messages with **1st data byte = 86 or 87**, which report the respective DMX levels channel by channel in rising order. This message is more complex but needs only 1 MIDI channel.

Set length of DMX cycle to the actual value of "SLOT" with

PROGRAM CHANGE

MIDI channel as selected base channel

data byte = 64 (hex40)

Comment: The minimum length of the DMX cycle (= number of DMX channels transmitted between two DMX reset pulses) is 24.

Switch acceptance of Running State in commands with:

PROGRAM CHANGE

MIDI channel as selected base channel

data byte = 56 (hex38) Running State in commands **forbidden**

= 57 (hex39) Running State in commands is **accepted** (default)

Comment: "Running State" means that subsequent equal status bytes are not repeated.

Most control software like MIDI sequencers always use "Running State", so this feature should only be deactivated in very special cases.

POLY KEY PRESSURE sets odd DMX levels: activate and deactivate with:

PROGRAM CHANGE

MIDI channel as selected base channel

data byte = 60 (hex3C) DMX level = MIDI data byte *2

data byte = 61 (hex3D) DMX level = MIDI data byte*2+1 (default)

Comment: when this effect is deactivated, POLY KEY PRESSURE commands have the same effect as NOTE ON commands. As a special case with NOTE ON as well as with POLY KEY PRESSURE the MIDI-data byte 127 is transformed into DMX level 255.

Exchange CONTROL CHANGE and POLY KEY PRESSURE functionally with:

PROGRAM CHANGE

MIDI channel as selected base channel

data byte = 62 (hex3E) **all functions of CONTROL CHANGE**

and POLY KEY PRESSURE are exchanged

with respect to the description in this manual

data byte = 63 (hex3F) CONTROL CHANGE and POLY KEY PRESSURE

have the same meaning as described in this manual
(default)

Comment: With this option, the command set may be better fitted to the particular control task or features of the given control equipment.

The default setting as described in the manual is most appropriate when many settings are made preferably with a controller pad or with a programmable keyboard - especially when the POLY KEY PRESSURE messages, which are less frequently used in normal MIDI operation, are not or badly supported. Recommended for control by keyboard or controller pad and evaluation of lighting scenes (presets).

The alternative setup with exchanged functions shows advantages if the control equipment (sequencer for example) provides good possibilities to work with POLY KEY PRESSURE messages and/or simultaneous fade ramps shall be (graphically) executed on a sequencer with CONTROL CHANGE messages. Recommended for recording and control with sequencer.

Block execution of MIDI commands with:

PROGRAM CHANGE

MIDI channel as selected base channel

data byte = 34 (hex 22)

Comment: this command is especially useful for example, when lighting scenes (presets, see below) to be loaded or saved are selected with a controller pad or similar. The blocking feature prevents that all intermediate values, which are emitted during motion of the controller, are executed. When programming of lighting scenes is made with a controller pad, we recommend that a pair of two key buttons is reserved for the block and unblock function.

Execute last received MIDI command, but perpetuate blocking with:

PROGRAM CHANGE

MIDI channel as selected base channel
data byte = 35(hex 23)

Comment: while execution of MIDI commands is blocked by PROGRAM CHANGE 36, the MIDI / DMX Interface keeps the last received MIDI channel message in mind. This command causes the last received and stored MIDI message to be executed. The blocking remains active. This allows efficient handling with controller pads without executing the intermediate values, which may be released while the controller is turned or slided.

Terminate blocking of commands with:

PROGRAM CHANGE

MIDI channel as selected base channel
data byte = 36 (hex 24) only terminate blocking, nothing is executed
data byte = 37 (hex 25) terminate blocking, execute last stored MIDI message

Comment: While blocking of commands is active, the MIDI / DMX Interface remembers the last received MIDI channel message.

With data byte = 36 only the blocking is terminated. This command version has the same effect as ASCII command {+ .

With data byte = 37 the last stored MIDI channel message is executed immediately. See comment at PROGRAM CHANGE 34 and 35, too.

Save, store presets (= lighting scenes) with:

CONTROL CHANGE

MIDI channel as selected base channel
1st data byte = 112 (hex70): basic value 0 = save preset no. 0-127
113 (hex71): basic value 128 = save preset no. 128-226

2nd data byte = 0 to 127 (hex7F): add 0 - 127 to basic value

Comment: After the MIDI / DMX Interface is powered on or a MIDI RESET message was received, generally preset no.0 is loaded. Together with presets no. 0 bis 3 the system configuration is stored in addition to the actual lighting scene. This includes especially: number base format of DMX levels, running state behaviour, reaction on CONTROL CHANGE commands, LOOP, chaser. Together with presets no. 4 to 226 only the lighting scene is saved.

Special cases: store preset partially

1st data byte = 114 to 116: basic value 0, i.e. partially store preset no 0 - 127

114 (hex72):store DMX channels 1-128 from DMX OUT to preset channels 129 - 256

115 (hex73): store DMX channels 1-128 from preset to DMX OUT channels 257 - 384

116 (hex74): store DMX channels 1-128 from preset to DMX OUT channels 385 - 512

1st data byte = 117 to 119: basic value 128, i.e. partially store preset no 128 - 226

117 (hex75): store DMX channels 1-128 from DMX OUT to preset channels 129 - 256

118 (hex76): store DMX channels 1-128 from DMX OUT to preset channels 257 - 384

119 (hex77): store DMX channels 1-128 from DMX OUT to preset channels 385 - 512

2nd data byte in all cases = 0 to 127 (hex7F) = preset no. minus basic value

any other levels in the preset remain unchanged.

This feature is intended as a counterpart to the corresponding partial load command to produce and test "long" presets with small lamp configurations.

Attention: with this special feature exclusively DMX levels are re-stored, in no case the channel specific details of the system configuration.

Save flash pattern with :

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 126 (hex 78)

2nd data byte = 15 (hex F) any values else are ignored

Comment: For detailed description, see flash start command above (CONTROL CHANGE 15)

Load presets (= lighting scenes) with:

CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 96 (hex60): basic value 0 = load preset 0-127

97 (hex61): basic value 128 = load preset 128-226

2nd data byte = 0 to 127 (hex7F): add 0 - 127 to basic value

Special cases: load preset partially

1st data byte = 98 to 100: basic value 0, i.e. partially load preset no 0 - 127

98 (hex62): load DMX channels 129 - 256 from preset to DMX OUT channels 1 - 128

99 (hex63): load DMX channels 257 - 384 from preset to DMX OUT channels 1 - 128

100 (hex64): load DMX channels 385 - 512 from preset to DMX OUT channels 1 - 128

1st data byte = 101 to 103: basic value 128, i.e. partially load preset no 128 - 226

101 (hex65): load DMX channels 129 - 256 from preset to DMX OUT channels 1 - 128

102 (hex66): load DMX channels 257 - 384 from preset to DMX OUT channels 1 - 128

103 (hex67): load DMX channels 385 - 512 from preset to DMX OUT channels 1 - 128

1st data byte = 104 to 107: basic value 0, i.e. partially load preset no 0 - 127

104 (hex68): load DMX channels 1 - 128 from preset to DMX OUT channels 1 - 128

105 (hex69): load DMX channels 129 - 256 from preset to DMX OUT channels 129 - 256

106 (hex6A): load DMX channels 257 - 384 from preset to DMX OUT channels 257 - 384

107 (hex6B): load DMX channels 385 - 512 from preset to DMX OUT channels 385 - 512

2nd data byte in all cases = 0 to 127 (hex7F) = preset no. minus basic value

any other levels of the DMX bus remain unchanged.

This feature is intended as a counterpart to the corresponding partial load command to produce and test "long" presets with small lamp configurations.

Attention: with this special feature exclusively DMX levels are loaded, in no case the channel specific details of the system configuration.

Alternatively load presets (= lighting scenes) with :

NOTE ON

MIDI channel as selected base channel

1st data byte (0 -127): preset number to be loaded **minus basic value**

corresponding with the range setting which was made before with a **PROGRAM CHANGE** command

2nd data byte: not relevant, except when velocity = 0, then no preset is loaded

This mode of operation has to be activated before with a PROGRAM CHANGE command, data byte 96 to 107 (hex60 to hex6B). It remains active until it is changed by a new PROGRAM CHANGE command. The default setting for NOTE ON is "short command" to simply set a DMX channel and level by one single MIDI message.

Comment: this method is useful to change lighting scenes quickly during "live operation", but helpful too for simple programming of a lighting track with a sequencer. As preparatory work an appropriate set of presets has to be produced. To manage this best, the number of keys of the keyboard which is used for this task has to be considered and possibly the pitch has to be transposed. If organized well, this way lighting design of a bigger repertoire may be performed quite efficient.

Change mode of operation with :

PROGRAM CHANGE

MIDI channel as selected base channel

data byte 96 to 107 (hex 60 to 6B) **NOTE ON commands load presets (see above)**

listed more specifically:

data byte = 96 (hex 60): **basic value 0**, i.e. NOTE ON loads preset no. 0-127

97 (hex 61): **basic value 128**, i.e. NOTE ON loads preset no. 128-226

and additional **special functions: load presets partially**

data byte = 98 ... 109: basic value 0, i.e. loads preset no. 0 - 127 partially

98 (hex62): NOTE ON loads DMX channels 129 - 256 from preset to DMX OUT chan. 1 - 128

99 (hex63): NOTE ON loads DMX channels 257 - 384 from preset to DMX OUT chan. 1 - 128

100 (hex64): NOTE ON loads DMX channels 385 - 512 from preset to DMX OUT chan. 1 - 128

data byte = 101 ... 103: basic value 128, i.e. loads preset no. 128 - 226 partially

101 (hex65): NOTE ON loads DMX channels 129 - 256 from preset to DMX OUT chan. 1 - 128

102 (hex66): NOTE ON loads DMX channels 257 - 384 from preset to DMX OUT chan. 1 - 128

103 (hex67): NOTE ON loads DMX channels 385 - 512 from preset to DMX OUT chan. 1 - 128

data byte = 104 ... 107: basic value 0, i.e. loads preset no. 0 - 127 partially

104 (hex68): NOTE ON loads DMX channels 1 - 128 from preset to DMX OUT channels 1 - 128

105 (hex69): NOTE ON loads DMX channels 129 - 256 from preset to DMX OUT ch. 129 - 256

106 (hex6A): NOTE ON loads DMX channels 257 - 384 from preset to DMX OUT ch. 257 - 384

107 (hex6B): NOTE ON loads DMX channels 385 - 512 from preset to DMX OUT ch. 385 - 512

data byte = 120 (hex 78) **NOTE ON messages set DMX channel and level.**

velocity = 0 is accepted and sets the DMX level to 0 (=default)

Any NOTE OFF message (with arbitrary velocity) sets the DMX level to 0

data byte = 121 (hex 79) **NOTE ON messages set DMX channel and level.**

NOTE ON messages with 2nd data byte (velocity) = 0 are ignored

and all NOTE OFF messages are ignored

but: NOTE ON messages with velocity = 1 set the DMX level to 0.

data byte = 122 (hex 7A) ask configuration of DMX channel= "SLOT"

A System Exclusive message will be returned with typically following ASCII content:

DMX-Tx: CH=1 OUT=13 TX=26 MF=50% CS=128/0/16 L=512 T=3.2

Comment: OUT reports the actually transmitted DMX level of the channel addressed by CH= SLOT, TX shows the content of the transmit buffer. Both differ when the global masterfader (item MF) is not set to 100%. CS reports the setting of the chaser in the order: start scene, cycle length, step duration. L shows the number of transmitted channels per DMX cycle. T displays the fade time: full seconds and eights separated by a dot.

data byte = 127 (hex 7F) **clears** all buffers and working registers

This command has to be entered twice to avoid destruction of setup by lapse of command entry

Comment: all system parameters are reset to "default" = state of delivery. This is not a reset!
All DMX levels of the transmit buffer are reset to "0".
Number base = "decimal". Masterfader = 100% LOOP = 512, FADETIME = 0.0.
The chaser is switched OFF.
Presets are not deleted or changed otherwise.

General comment on PROGRAM CHANGE messages:

As an alternative, all commands which can be performed with a PROGRAM CHANGE can be sent as a CONTROL CHANGE command with 1st data byte = 127 (hex 7F). Then its 2nd data byte is equal to the data byte of the corresponding PROGRAM CHANGE command.

When formatting a PROGRAM CHANGE message, many MIDI sequencers and other MIDI control equipment demand that the data value (program number) has to be entered one higher than the physically transmitted data byte. (Example: user entered program #1 is transmitted as hex C0 00), **The MIDI/DMX Interface evaluates PROGRAM CHANGE messages with the physical parameter as described in the manual. Check your equipment if the parameter has to be entered by 1 higher**

Some MIDI devices send together with a PROGRAM CHANGE Befehl automatically a BANK SELECT command (=CONTROL CHANGE to controller no.32(hex20)).

If you notice this conflict: don't address any DMX receiver at DMX channel no 32.

Fine adjustment of DMX timing (DMX reset pulse) with: CONTROL CHANGE

MIDI channel as selected base channel

1st data byte = 126 (hex7E)

2nd data byte = 0 to 127 (hex7F): see description below

This command has to be entered twice in subsequent order.

Comment: This command is intended to solve extreme synchronisation problems with other DMX equipment. Especially if certain types of DMX lamp tend to flicker irregularly, it may be helpful to increase the DMX timing parameters

The 2nd data byte (controller value) is composed of two hex nibbles 0 .. F:

1st hex digit (high nibble): Duration of the DMX break pulse: ca. 180us base value + (nibble value * 32 us). This way, the range of the DMX break pulse may be varied between about 180us and 404 us.

2nd hex digit (low nibble): Duration of the mark pulse between DMX break and start byte: ca. 16us + nibble value. So, the setup range is about 16us to 31us.

The parameter is stored permanently independent of presets and is reloaded during every system start. Entering the command with controller value= 0 resets the DMX timing permanently to its default values.

Data format MIDI reset:

data byte = 255 (hexFF) MIDI System Reset

The LED is darkened for 1 second, then the DMX Interface resets and restarts loading preset no.0.

Set the user defined MIDI base channel

with a System Exclusive command:

Send the byte sequence 240 - 125 - MIDI channel (1-16) - 247

or in hex notation F0 - 7D - MIDI channel (1-hex10) - F7

Comment: this is the only System Exclusive message the MIDI / DMX interface reacts on.

The user defined MIDI base channel may be entered at any time and is stored permanently. But the new base channel is not applied before the next power cycle of the interface. It is valid only during jumper settings J1+J2 and J1+J3 (see page2).

Appendix A

Concept and data format of DMX512 transmission:

To understand the function of the DMX Control Box or DMX Interface and its commands, you should roughly know the concept of internal memories for received and transmitted DMX signals.

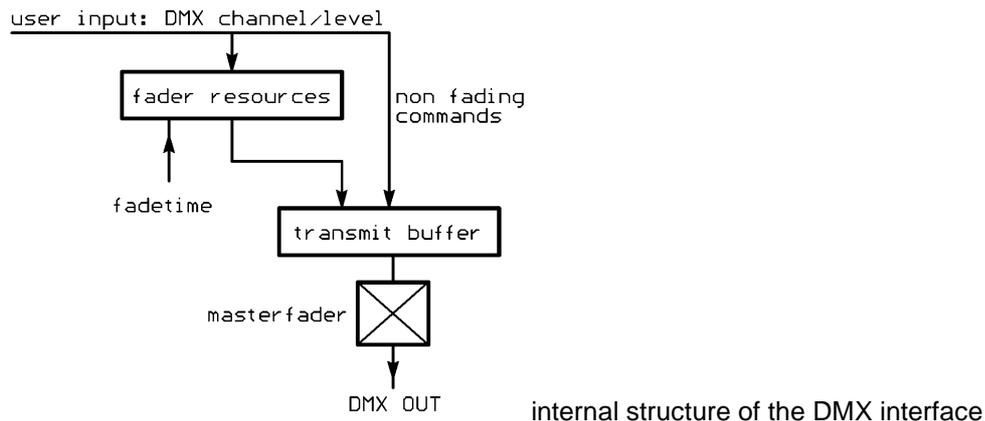
For each of the 512 DMX channels three specific memory cells are provided: transmit buffer, receive buffer and action memory.

The received DMX signal is permanently written in realtime into the receive buffer. DMX level data entered by the user are preprocessed if necessary (e.g. a fade process initiated) and then written into the transmit buffer.

The expression "transmit buffer" does not mean that its content is transmitted unconditionally: **Before transmission of any next DMX byte the merger routine looks into the action memory if this byte shall be taken from the receive buffer or from the transmit buffer.**

The **masterfader** modulates the DMX data bytes as a simple signal processor when they are transferred from the **transmit buffer** into the DMX transmitter hardware. This way a complete lighting scene may be "stretched" globally in a simple way.

Internally stored DMX data are not manipulated by the masterfader.



The philosophy of commands for writing into the transmit buffer is based on the cooperation of three registers:

- DMX channel ("SLOT") register
- DMX level register
- FADETIME register.

The **SLOT register** stores the actually addressed DMX channel (= physical time slot 1 to 512 within the DMX cycle), which shall subsequently be written, read or modified otherwise by **user commands**.

For every DMX channel a complete fade process can be performed. The fadetime to be used is set in the **FADETIME register**. This fade process is automatically controlled by the firmware of the box. **Fade processes can be active simultaneously on all DMX channels** and even presets (=stored lighting scenes) are faded over from the previous scene with FADETIME while they are loaded. The actual value of FADETIME is copied into the respective fader resource when a fade process is started. Straight after then FADETIME can be changed for the next command without influence on any running fader process.

Last not least, a LOOP register is implemented, which stores the actual actual number of slots to be transmitted under user control in each DMX packet.

After power on or reset, SLOT is initialized with "1", DMX levels and FADETIME are set to "0", the masterfader is set to 100% and LOOP is started with "512".

Appendix B

Data format DMX512

The "DMX512" standard assumes that a bus master transmits DMX data packets on a DMX data bus in a permanent cyclic repetition. This bus line is looped from the transmitter to the next receiver and from there to the next receiver and so on in a linear bus topology. **Every DMX receiver connected at the bus is permanently supplied with actual control data for all receivers**, no matter if these data have been changed in the meantime.

Every **DMX packet transmission cycle starts with a special reset sequence and a start byte** as header. Subsequently all data bytes are transmitted in 8 bit serial format to all potential receivers. This way every transmitted DMX byte is assigned with a specific time slot within every DMX packet and it can be addressed with the number of this time slot. Therefore the "addresses" or "channels" in a DMX data packet are called "slots". Per DMX slot one data byte is transmitted. The value of this data byte (range 0 to 255) describes the intensity of a lamp or the position of a "moving head". **Throughout this manual we use the expression "DMX level" or "DMX value" to describe this byte value.**

Addressing of receivers is defined by the time position ("slot") of their data bytes within the cyclic DMX package. By means of a code switch or something similar a number between 1 and 512 can be selected at every DMX receiver. The receiver scans the count of slots during every DMX packet and starts to read out the slot bytes starting from the slot number selected with its switch. The number of evaluated bytes depends on the specific function of the receiver. If, for instance, the switch of a certain receiver is set to 28, this receiver will read the 28th, 29th, 30th ... byte following after the start byte out of the DMX data stream during every DMX cycle. These data are not removed out of the DMX data stream. But all receivers with a different setting of their code switch do ignore them. Depending on the setting of their code switches, several receivers can get the same data - intentionally or by mistake.

Due to its design the "DMX512" standard does not offer inherent error checking and error correction of transferred data and does not supply a usefully standardized feedback channel from receiver to transmitter. While slowly changing actuators like bulb lamps or servo motors are driven, single faulty bytes almost have no effect on the visual behaviour, because they are corrected in the next cyclic transmitted DMX packet with high probability. **Very critical however is to release singular switching events via the DMX-bus.**

For these reasons the the use of DMX control is explicitly FORBIDDEN together with any safety critical application, where malfunction could result in personal injury or noticeable material damage !

To install a DMX512 bus, preferably a special 2 wire twisted and shielded "RS-485" data cable is recommended, which unfortunately is quite expensive and only available from special providers for industrial control. CAT5 ethernet cable has been proved to be a good material for DMX installation, too. For solid installations we have made good experience with shielded ISDN cable, which is traded in germany as type JY(St)Y. The version with 0,8mm wire diameter should be used if available. Using bus lengths up to 100 m, a high quality 2 wire microphone cable will do.

Long bus lines have to be terminated at the most distant receiver. Termination means: DMX+ and DMX- are connected with a resistor of 120 Ohm.

Normally the bus line is "looped through" from receiver to receiver. Most DMX equipment is supplied with a DMX IN and a DMX OUT connector, both are physically wired together inside. For this kind of installation you should have available a 5-pin male XLR-connector prepared with a built-in resistor, which is plugged into the DMX OUT of the last DMX receiver in the chain.

Appendix C

Structure of MIDI channel messages (super compact crash course)

The first byte is the **status byte, only this one has bit 7 set.**

It is composed of 2 hex nibbles. The most significant (high) nibble describes the message type, the least significant (low) nibble contains the MIDI channel. The MIDI channel is always physically coded one less than written in any MIDI manual or literature. Putting both together: (MIDI channel-1) has to be added to the message type status body.

For MIDI channel no. 1 the different types of MIDI messages show following status byte (=status body):

NOTE OFF:hex80,dec128

NOTE ON:hex90,dec144

POLY KEY PRESSURE(=POLY KEY AFTERTOUCH):hexA0,dec160

CONTROL CHANGE:hexB0,dec176

PROGRAM CHANGE:hexC0,dec192

CHANNEL PRESSURE:hexD0,dec208

PITCH CHANGE:hexE0,dec224

The status byte of PROGRAM CHANGE and CHANNEL PRESSURE is followed by exactly 1 data byte

The status byte of all other message types is followed by exactly 2 data bytes

In all data bytes the most significant **bit7 is cleared**, the other bits are content specific.

Determination of MIDI note values from common note names:

	C	C#	D	D#	E	F	F#	G	G#	A	B	H
-2	0	1	2	3	4	5	6	7	8	9	10	11
-1	12	13	14	15	16	17	18	19	20	21	22	23
	24	25	26	27	28	29	30	31	32	33	34	35
1	36	37	38	39	40	41	42	43	44	45	46	47
2	48	49	50	51	52	53	54	55	56	57	58	59
3	60	61	62	63	64	65	66	67	68	69	70	71
4	72	73	74	75	76	77	78	79	80	81	82	83
5	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107
7	108	109	110	111	112	113	114	115	116	117	118	119
8	120	121	122	123	124	125	126	127	-	-	-	-

This scheme is used by several MIDI sequencers. But deviating names of the different octaves are in use, too. Following unique relationship may be helpful: the concert pitch a (440Hz) -- in the table above "A3" - is always corresponding with MIDI note value 69 (hex 45).

contact: wschemmert@t-online.de

* Right of technical modifications reserved. Provided 'as is' - without any warranty. Any responsibility is excluded.

* This description is for information only, no product specifications are assured in juridical sense.

* Trademarks and product names cited in this text are property of their respective owners