

## USB/DMX Interface - Command Specification (V 1.4)

---

### History

---

05/05/2005 : V1.1 : Updated a few incorrect hex values (DMX Value, No DMX Monitor)  
11/05/2005 : V1.2 : Updated a missed description related the previous update and changed  
the description for 2.3.2 & 2.3.3  
17/11/2006 : V1.3 : Removed channel set DMX response 2.3.4  
25/02/2007 : V1.4 : Added User Memory, TX Status and Read TX Channel value commands

Ben Suffolk, 2003 - 2007

The USB/DMX interface uses the FTDI USB parallel chip set (245). Drivers can be downloaded from <http://www.ftdichip.com/FTDrivers.htm>

Communication with the USB/DMX Interface is asynchronous. There are a number of commands that can be written to the Interface, and a number of responses that can be read from the interface. There is no guarantee that a system generated response will not occur after a command request, and before the command response. This document sets out the specification for these command & responses.

#### 1. General format

Commands and Responses are grouped together into a number of sub groups, System, Global, TX, RX

There is one format that all commands and response headers are defined to. It is a single byte made of the following bits :-

7 6 5 4 3 2 1 0  
R G G C C C C H

R = 1 for Response, 0 for Command

GG = 00 System, 01 Global, 10 TX, 11 RX

CCCC The identifier

H Some Commands / Responses use this to identify the high address bit. If not used must be set to 0

Whenever a command is issued, you will receive a response header matching the command identifier (but with the response bit set). You may optionally (depending on the command) receive additional information in following bytes.

When using commands that contain address / channel numbers, 0x000 = channel 1 and 0x1FF = channel 512

#### 2. Commands

##### 2.0 System Commands

These commands are related to the system and not the DMX interface.

###### 2.0.1 Flash Update 0x02

This command starts a flash update procedure to update the firmware, this command has a number of safeguards to prevent accidental corruption of the firmware. Please see the Flash Upgrade document for further details.

##### 2.1 Global Commands

These commands are related to the interface in general

###### 2.1.1 Reset - 0x22

This command resets the interface. This means that any further commands waiting to be executed will be flushed, TX & RX will be switched off, and the default settings will be restored.

As a safety feature the Reset command will only work if it is preceded by a No Operation. i.e. to initiate a reset you must send 0x26,0x22

If the Reset command is accepted you will receive a standard Reset Response (0xA2) If you did not precede the Reset command with a No Operation you will receive the Failed variant of the Reset Response (0xA3)

#### 2.1.2 Version Request - 0x24

This command requests that interface to return its current version information. The expected response is :-

0xA4, <Version Byte>, <Version String>, 0x00

<Version Byte> is a byte describing the version, with bits 4-7 denoting the main version number and bits 0-3 being the sub version number, e.g 0x10 is decoded as version 1.0

<Version String> is a string with version details. This String is made of standard ascii characters and will not contain 0x00.

#### 2.1.3 No Operation - 0x26

This command does nothing but respond with the No operation response. This can be used as a heart beat check to ensure the interface is still responding. Response is 0xA6

#### 2.1.4 Set User Memory Value - 0x28

The interface has 512 user memory locations that can be used for any purpose. Values will only be stored whilst the interface is connected to the USB port. The memory locations are addressed as a 9 bit value from 0 to 511.

This is a 3 byte command, 0x28 / 0x29 (High address bit), <low address byte>, <value>

No response is given to this command.

#### 2.1.5 Read User Memory Value - 0x2A

This is a 2 Byte command 0x2A | 0x2B (High address bit), <low address byte>

Response (3 Bytes) : 0xAA 0xAB, <low address byte>, <value>

### 2.2 DMX Receiver commands

These commands are related to the RX part of the interface

#### 2.2.1 Set RX Start Code - 0x62

This command sets the start code that you wish to receive DMX on. It defaults to 0, which is the only start code that meets the DMX specifications. However custom applications may require it to be changed.

This is a 2 byte command, the first being 0x62, the second being the new start code.

The new start code will take effect from the very next DMX frame being read in. Response: 0xE2

#### 2.2.2 Turn RX On - 0x64

This command enables the RX capability of the interface. DMX Responses will start to be sent over the interface. Response: 0xE4

#### 2.2.3 Turn RX Off - 0x66

This command disables the RX capability of the interface. DMX Responses will no longer be sent over the interface. Response: 0xE6

#### 2.2.4 Get the last start code read in - 0x6C

This command returns the last start code read in. Response will be one of :-

0xEC, <start code>

or

0xF4 (Meaning the the DMX monitor has not started, as no DMX has been received)

#### 2.2.5 Get the last channel number - 0x68

This command gets the last channel index from the last (start code does not need to match) DMX frame. Response will be one of :-

0xE8, <Low Address Byte>

Note that if bit 8 of the address needs to be set the response will be :-

0xE9, <Low Address Byte>

or

0xF4 (Meaning the the DMX monitor has not started, as no DMX has been received)

#### 2.2.6 Get the Frames Per Second - 0x6A

This command gets the last received valid frames per second count. Response will be :-

0xEA, <Frames Per Second>

or

0xF4 (Meaning the the DMX monitor has not started, as no DMX has been received)

### 2.3 DMX Transmitter commands

These commands are related to the RX part of the interface

#### 2.3.1 Set TX Start Code - 0x42

This command sets the start code that we wish to transmit DMX on. It defaults to 0, which is the only start code that meets the DMX specifications. However custom applications may require it to be changed.

This is a 2 byte command, the first being 0x42, the second being the new start code.

The new start code will take effect from the very next DMX frame being sent. Response: 0xC2

#### 2.3.2 Turn TX On - 0x44

This command enables the TX cabability of the interface. DMX packets will start to be sent from the interface.

#### 2.3.3 Turn TX Off - 0x46

This command disables the TX cabability of the interface. DMX packets will no longer be sent from the interface.

#### 2.3.4 Set DMX Channel Value - 0x48

This is a 3 byte command, 0x48 / 0x49 (High address bit), <low address byte>, <channel value>

NOTE: Since version 1.3 of the firmware no response is given to this command. Earlier versions responded with 0xC8

#### 2.3.5 Turn on Blackout On - 0x4A

Temp set all output to 0, Response: 0xCA

#### 2.3.6 Turn off Blackout - 0x4C

Restore output to existing value, Response: 0xCC

#### 2.3.7 Set the last TX code - 0x4E

This is a 2 byte command, the first being 0x4E / 4F (High address bit), <low address byte>. You should set this as low as you need it because the lower this is, the less data that needs to be sent, and therefore the more frames per second (refreshes) you can achieve.

Response: 0xCE

#### 2.3.8 Check TX Status - 0x50

This is a single byte command to check if TX is currently enabled or not. The primary purpose for this check is so that if you are trying to recover from a software crash mid show, you do not need to issue a TX ON command, thus resetting all channels.

Response: 0xC4 - Tx ON  
          : 0xC6 - Tx OFF

#### 2.3.9 Read TX Channel Value - 0x52

This command allows you to read the value of any channel being transmitted. Again the purpose for this command is really to allow an application recovering from a software crash to sync with the interface.

This is a 2 Byte command 0x52 | 0x53 (High address bit), <low address byte>

Response (3 Bytes) : 0xD2 0xD3, <low address byte>, <channel value>

### 3.0 Response codes not originated by a command

#### 3.1 Started receiving DMX - 0xF6

This response is generated when DMX is received on the interface that matches the start code.

#### 3.2 No longer receiveing DMX - 0xF8

This response is generated when DMX, that matches the start code, has not been received on the interface for 1 second.

#### 3.3 DMX Value - 0xFE

When a DMX Channel changes value this response will be sent, it will be followed by the channel and the value. This response uses a total of 3 bytes. The high bit from the address (A8) is set in bit 0 of the response value, so the initial response code could be either 0xFE or 0xFF

Byte 1 0xFE | 0xFF  
Byte 2 <A7 - A0>  
Byte 3 DMX Value

#### 3.4 DMX Framing Error - 0xFC

If there is a DMX Framing error this response will be sent, this normaly only occurs when the DMX source is removed / switched off / faulty.

#### 3.5 Hardware Overflow - 0xFA

If a hardware overflow occurs (the hardware was unable to read a DMX value before the next one is received) this response will be sent. This should never happen.

#### 3.6 Unknown Command - 0x84

If an unknown command is received by the interface this response will be given.

#### 3.7 No Code - 0x86

This response is sent by the Bootloader if the firmware is not loaded, This response will only be received when you issue a command and the firmware is corrupted (by a faulty flash update).

