“Termini”

DVB-T USB2.0
communication protocol specification

Version 1.0
2.4.2004

Table of Contents

1. Enpoint 1

Communication with EP1 enables the host to set tuner parameters, query the current tuner status, start and stop the highspeed data stream and directly read from or write to the I2C communication bus.

The first byte – hereinafter refered to as the 'command byte' – specifies the type of request, a short description follows.

0x00 Read from or write to I2C bus
0x01 Reserved for future use
0x02 Reserved for future use
0x03 Start or stop the highspeed data stream
0x04 Set tuner parameters
0x05 Query current tuner parameters and status
0x06 Start brute-force channel scan
0x07 Continue brute-force channel scan

1.1 I2C communication

The receiver contains components with the following I2C addresses:

0x51 EEPROM (24LC32 or 24LC64 compatible)
0x0F Zarlink MT352 demodulator

You can access these components directly using the I2C read/write command (command byte 0x00), but you should never need to do so unless you debug the FX2 firmware code.

For more information about the I2C communication protocol in general and the I2C components in use, please refer to the I2C protocol specification and the according component data sheets.

1.1.1 Request

Endpoint	0x01
Command byte	0x00
Packet length (including command byte)	4-64 bytes

USB Packet Layout:

Byte #	Valid values	Function
0	0x00	Command byte
1	0xA2, 0xA3, 0x1E, 0x1F	I2C address and r/w-mode
2	0x00 – 0x3C	Number of bytes to read or write
3	0x00, 0x01	Stop condition suppression flag
4-n	0x00 - 0xFF	Data buffer

I2C address field byte layout:

B7	B6	B5	B4	B3	B2	B1	B0
I2C address [6:0]							Read

Set bit B0 = 1 for read, B0 = 0 for write operation.

Setting the stop condition suppression flag (byte 3) to 0x01 causes the I2C stack to not send a stop condition after writing bytes, this is required by some I2C slave devices for read operation.

1.1.2 Return

A USB bulk buffer is returned which is at least one byte long.

The first byte reflects the result of the I2C bus operation:

0x00 OK
0x01 Invalid request or communication failure
0x02 No acknowledge of I2C slave device
0x03 I2C bus error, usually a hardware defect, probably SCL or SDA stuck low.

For read requests, the following bytes [0x04...n] contain the data read from the I2C bus.

You shouldn't need to use this command at all since all necessary communication can be done more elegant with the higher level interface implemented by the following commands.

1.2 Start or stop the highspeed data stream

This command starts or stops the highspeed transfer on EP2.

1.2.1 Request

Endpoint	0x01
Command byte	0x03
Packet length (including command byte)	2 bytes

USB Packet Layout:

Byte #	Valid values	Function
0	0x00	Command byte
1	0x00, 0x01	Parameter byte

Parameter byte:

B7	B6	B5	B4	B3	B2	B1	B0
unused							Start

Set B0 to 1 to start the MPEG2 stream transfer over the USB endpoint 2, use 0 to stop it.

1.2.2 Return

An empty 0-byte bulk packet.

1.3 Set tuner parameters

1.3.1 Request

Endpoint	0x01
Command byte	0x04
Packet length (including command byte)	9 bytes

USB Packet Layout:

Byte #	Valid values	Function
0	0x03	Command byte
1-4	any	Frequency in kHz, (Little Endian)
5	0x06, 0x07, 0x08	Bandwidth in MHz
6-7	any	TPS parameters (Little Endian), see appendix A
8	any	Flags

Frequency specifies the requested Tuner Frequency, Bandwidth the transmission bandwidth. In Europe this is usually 7MHz for VHF and 8MHz for UHF channels.

If you don't force special parameters in the Flags byte you can pass any value to the TPS word. For a detailled explanation of the TPS parameters see the ETSI DVB Specification.

Flags byte:

B7	B6	B5	B4	B3	B2	B1	B0
unused				force_ guard	force_ mode	force_ spec_inv	spec_inv

If force bits are not set, the demodulator probes the matching parameters automatically. In this case it might take a little more time to aquire a lock compared to a fully specified parameter set.

1.3.2 Return

A 0-byte bulk packet.

1.4 Query tuner parameters and status

1.4.1 Request

Endpoint	0x01
Command byte	0x05
Packet length (including command byte)	1 byte

This request returns a 25 byte USB packet indicating the tuner's current parameter settings and status bits. It does not take any parameter, the USB packet consists of the command byte only.

1.4.2 Return

USB Packet Layout:

Byte #	Function
0-3	Frequency in kHz (Little Endian)
4	Bandwidth in MHz
5-6	TPS parameters (Little Endian), see appendix A
7	Flags
8-9	Gain (Little Endian)
10	Signal-to-noise ration in dB
11-14	Viterbi bit error rate (Little Endian)
15-18	Reed Solomon bit error count (Little Endian)
19-22	Uncorrectable block count (Little Endian)
23-24	Lock/status indicator bits

Frequency, Bandwidth and TPS fields have the same meaning like the matching Set Parameters call.

Byte 7 - Flags:

B7	B6	B5	B4	B3	B2	B1	B0
unused							spec_inv

You can use the AGC gain value as indicator of the incoming signal strength.

The Viterbi Bit Error Rate is a good measurement of the signal quality and should be used in favor of the SNR.

This is the number of uncorrectable MPEG packets occurred since the last read operation.

Byte 23 – Lock/status indicator bits:

B7	B6	B5	B4	B3	B2	B1	B0
TPS_valid	BA_lock	FEC_lock	OFDM_ found	PILOT_ lock	DSCR_ lock	SYM_lock	AGC_lock

Description:

TPS_valid	Set to 0 if the TPS word is invalid at the end of a frame, e.g.due to a bad check sum.
BA_lock	Set when the Byte Aligner is in lock.
FEC_lock	Set when the forward error correction unit is working stable.
OFDM_found	Set when all OFDM pilots are found. Used to test for the presence of a DVB-T channel.
PILOT_lock	Set when OFDM continuous pilots are successfully received.
DSCR_lock	Set when the Descrambler is in lock.
SYM_lock	Set when the initial symbol timing lock has been established.
AGC_lock	Set when the AGC is in lock.

Byte 24 - Lock/status indicator bits (continued):

B7	B6	B5	B4	B3	B2	B1	B0
unused							prev_ FEC_lock

prev_FEC_lock Set when a FEC lock has been achieved since the last read. This is interesting during a channel scan. If this bit is set, a channel has been found on the current frequency. The 'continue scan' command (0x07) needs to be issued to continue the scan. Please note that if there is no FEC_lock bit set, there has been a former lock which has gone away again, that usually indicates a bad reception quality.

Also, this bit is only set in scan mode.

For more detailed information about the meaning of the bits described here, please refer to Zarlink's MT352 design manual and to the ETSI DVB specification.

1.5 Start brute-force channel scan

The demodulator is capable of doing a brute-force channel scan over the entire possible frequency range.

Such a scan is initiated by the start command (command byte 0x06). After that, you need to poll the device's status (command byte 0x05). Once a channel is found (i.e. lock bits are set), you can continue the scan with command 0x07.

1.5.1 Request

Endpoint	0x01
Command byte	0x06
Packet length (including command byte)	10 bytes

USB Packet Layout:

Byte #	Valid values	Function
0	0x07	Command byte
1-4	any	start frequency in kHz (Little Endian)
5-8	any	end frequency in kHz (Little Endian)
9	0x06, 0x07, 0x08	Bandwidth in MHz

1.5.2 Return

A 0-byte bulk packet.

1.6 Continue brute-force channel scan

1.6.1 Request

Endpoint	0x01
Command byte	0x07
Packet length (including command byte)	1 byte

This request returns a 0-byte packet. It does not take any argument.

1.6.2 Return

A 0-byte packet.

2. Enpoint 2 (highspeed)

Once the highspeed stream has been started as described in section 1.2, bulk messages with each 512 bytes will get issued continously from EP2.

Buffers are not aligned to MPEG packet boundaries.

Please note that the host side needs to provide a sufficient number of input buffers (USB request blocks) to catch all data packets and ensure a smooth transmission. At an average bandwidth of 14,5 Mbit/s, about 5000-6000 buffers arrive every second.

Appendix A – Transmission parameter signalling

The 16-bit TPS word describes the DVB-T transmission parameters and is assembled according to ETSI ETS 300744, section 4.6.2, table 9:

TPS	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
function	HP/LP	const		hierarchy			HP code rate			LP code rate			guard		mode	
000	HP	QPSK		None			1/2			1/2			1/32		2K	
001	LP	QAM16		1			2/3			2/3			1/16		8K	
010	n/a	QAM64		2			3/4			3/4			1/8		n/a	
011		n/a		4			5/6			5/6			1/4			
100				n/a			7/8			7/8			n/a			
101-111							n/a			n/a						