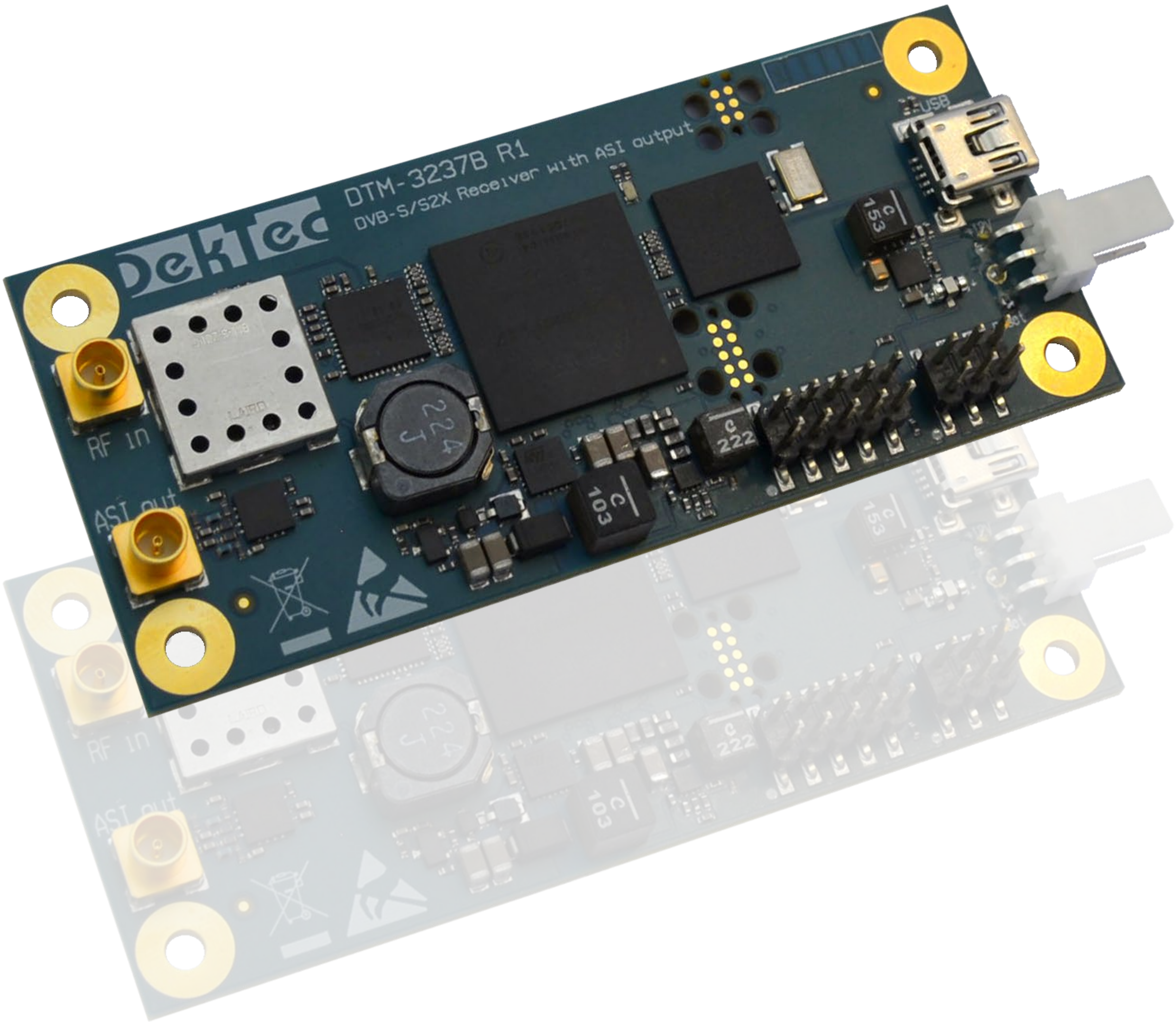
DTM-3237B

Compact DVB-S2X Receiver Module with DVB-ASI Output



USER **MANUAL**

July 2023

Table of Contents

DekTec Digital Video B.V. reserves the right to change products or specifications without notice. Information furnished in this document is believed to be accurate and reliable, but DekTec assumes no responsibility for any errors that may appear in this material.

Revision History

Version	Date	Changes
0.0	2023.07.15	Initial release to the field

1. Introduction

1.1 General description

The DTM-3237B is a compact OEM module for receiving DVB-S2 or DVB-S and outputting it as DVB-ASI. The DTM-3237B can supply LNB power and has support for sending and receiving DiSEqC control messages. The transponder details and the output format can be configured programmatically through several control interfaces.

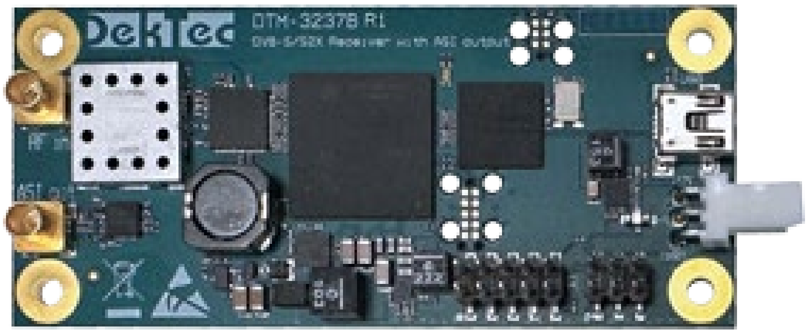


Figure 1. The PCB of the DTM-3237B

A development kit (DTM-3237B-DEVKIT; refer to Appendix B) is available for easy setup and experimentation with the DTM-3237B.

1.2 Input Modes

The DTM-3237B supports all functionality described in EN 300 421 (DVB-S), EN 302 307 (DVB-S2) and EN 302 307-2 (DVB-S2X). All DVB-S2 modulation types can be demodulated: QPSK, 8PSK, 16/32APSK. The DTM-3237B supports advanced DVB-S2 features including VCM, ACM, Multiple Input Streams (MIS) and Generic Stream (GS).

1.3 Output Modes

The DTM-3237B outputs the received DVB-S, DVB-S2 or DVB-S2X stream through a Transport Stream over DVB-ASI.

1.4 Control and Management

The DTM-3237B provides options for control through either USB or I^{2}C. Settings can be persistently stored in non-volatile memory if the "Volatile Storage" option is set to '1'. This ensures that configurations are retained even after power cycling. However, do note that storing settings in non-volatile memory incurs a slight time delay and is subject to a lifetime limit on write cycles.

If "Volatile Storage" is set to '0', setting changes take immediate effect but will not be retained when the power is cycled.

There are three control mechanisms for the DTM-3237B:

1. ***USB Control from a Development PC***. Employ the specialized control software—either Dtm3237BUtil or DtmCmd—on a development PC connected to the DTM-3237B via a USB interface. This is particularly useful for initial configuration or experimenting and tweaking of the DTM-3237B.

2. ***I^{2}C or USB Control from an Embedded Controller***. If the DTM-3237B serves as a co-processor board for satellite reception within a larger system, control can be exerted through the on-board I^{2}C or USB interface. In this case I^{2}C control port is a plausible choice for the control interface, but USB interface can also be used. The I^{2}C address can be pre-configured using the Dtm3237BUtil tool via USB (mechanism 1). The factory-default I^{2}C address is 0x60.
3. ***Stand-Alone Operation***. For situations that don't require dynamic adjustments, the DTM-3237B can operate in a pre-configured, stand-alone mode.

Two control tools are available:

1. ***Dtm3237BUtil***. A Windows GUI-based utility that allows you to view the DTM-3237B's status and adjust settings. This tool can also be used for firmware updates and is ideal for initial setup and experimental use.
2. ***DtmCmd***. A command-line utility designed to send commands to the DTM-3237B. It is useful for studying the low-level commands available for the DTM-3237B. Multiple DtmCmd command lines can be combined in a script to apply a group of settings in one go.

1.5 DTM-3237B Protocol Handler Source Code

For developers interested in building custom applications to control the DTM-3237B, a complimentary open-source protocol handler designed for DTM-32XX series devices is available. It can be downloaded from `www.dektec.com` free of charge and is royalty-free for use. The handler consists of two source files, `DtmHandler.c` and `DtmHandler.h`, which can be compiled and linked into your C or C++ application. Detailed instructions for incorporating the protocol handler into your project are provided in the `DtmHandler.h` file.

Note:

- The command-line control tool ***DtmCmd*** is an example of an application that uses the DTM handler. The source of ***DtmCmd*** is also available on the DekTec website. Please refer to Appendix C for more information about ***DtmCmd***.

1.6 Theory of Operation

Essentially, the DTM-3237B consists of two subsystems:

- A tuner, demodulator and stream processor, converting the modulated DVB-S2X input signal to a Transport Stream on DVB-ASI;
- A processor subsystem that handles all internal/external control (USB, I^{2}C).

1.7 List of Abbreviations

ACM	Adaptive coding and modulation
APSK	Amplitude and phase-shift keying
ASI	Asynchronous serial interface. Shorthand for DVB-ASI.
BCH	Cyclic error-correcting codes, abbreviation comprises the initials of its inventors names (inner FEC coding used for DVB-S2)
BER	Bit error rate
CCM	Constant coding and modulation

CNR	Carrier to noise ratio
CRC	Cyclic redundancy check
DiSEqC	Digital Satellite Equipment Control
E_b/N_0	Energy per bit to noise power spectral density ratio
E_s/N_0	Energy per symbol per noise power spectral density
FEC	Forward error correction
FLASH	Non-volatile storage chip
ISI	Input stream identifier
L.3	Receiver adaptation serial output interface with in-band signaling
LDPC	Low-density parity-check code (outer FEC coding used for DVB-S2)
LNB	Low noise block
LVTTL	Low voltage transistor-transistor logic (3.3V)
Mbps	Megabit per second
MER	Modulation error rate
MIS	Multiple Input Stream
ModCod	Modulation and coding (combination of constellation and code rate)
NA	Not applicable
NC	Not connected
PSK	Phase-shift keying
QPSK	Quadrature phase-shift keying
R/W	Read / Write
RO	Read only
RS	Reed-Solomon (inner FEC coding used for DVB-S)
SIS	Single Input Stream
SNR	Signal to noise ratio
ST188	188-byte Transport Stream mode
VCM	Variable coding and modulation
WO	Write only

1.8 References

[1] DTAPI Manual - Overview and Data Formats, L.3 Baseband frame implementation. Part of DekTec's *Windows SDK*

[2] DVB-S, ETSI EN 300 421, Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services

[3] DVB-S2, ETSI EN 302 307, Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications.

[4] DVB-S2X, ETSI EN 302 307-2, Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 2 DVB-S2 Extensions (DVB-S2X).

2. Getting Started

2.1 Introduction

This section provides a walkthrough for getting started with the DTM-3237B. The description below assumes that you have a DTM-3237B development kit available (see Appendix B). The DTM-3237B is connected to a development PC with USB. The GUI control tool ***Dtm3237BUtil*** is used to apply settings and observe status.

2.2 Configuration: Receiving a DVB-S2X Stream

This setup will receive a DVB-S2X stream and transmit the stream on the ASI interface.

2.2.1 Test Set-Up

For testing this configuration, a DVB-S2X signal should be connected to the DTM-3237B's RF input. To observe the output of the DTM-3237B, an ASI receiver is helpful[1].

This tutorial assumes that a DVB-S2X stream with the following or equivalent parameters is applied to the DTM-3237B.

Modulation standard	DVB-S or DVB-S2
Constellation	QPSK, 8PSK, 16-APSK or 32-APSK
Frequency	1150 MHz
Code rate	½
Symbol rate	27.5 MSymbol/s
RF power level	-30dBm

2.2.2 Configuring the DVB-S2X to ASI Conversion

Use ***Dtm3237BUtil*** to configure the DTM-3237B as shown in the table below. The receiver is updated when a configuration parameter is updated using the pen icon next to each setting.

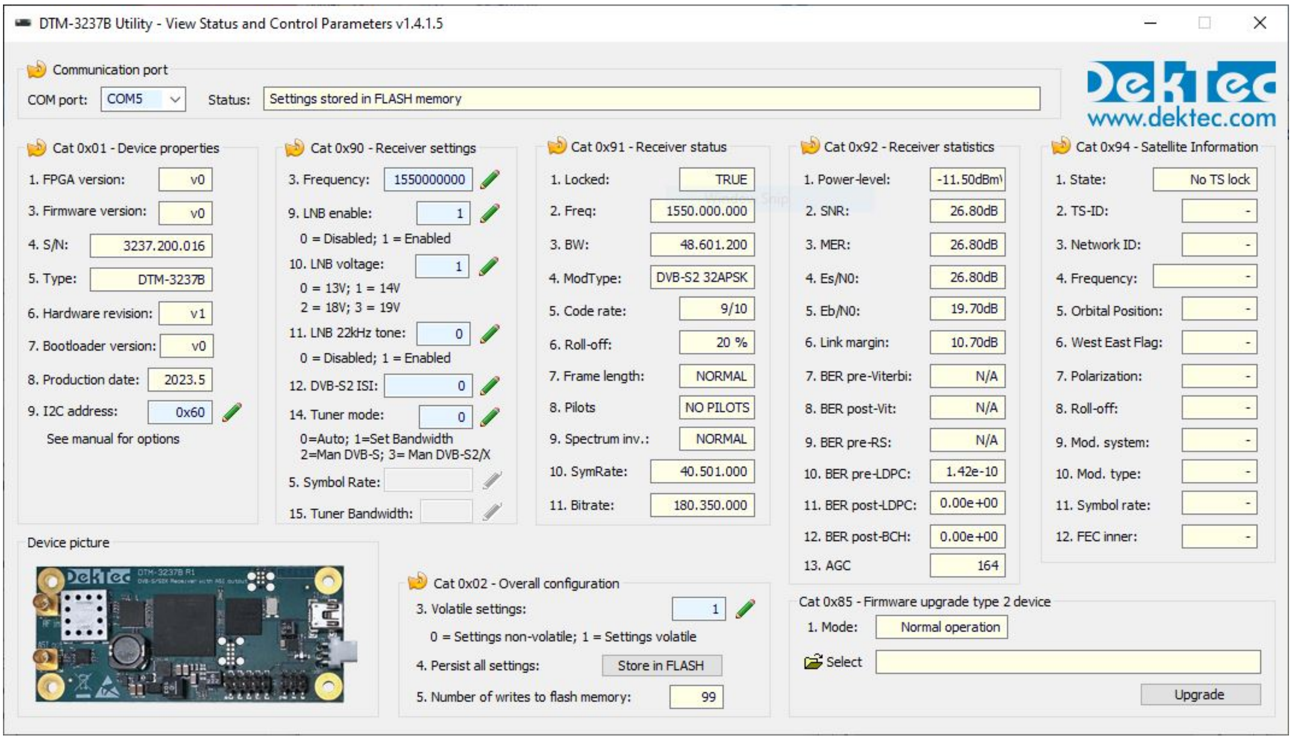
Change the following parameters:

	Category	Setting	Setting name	Value
1	0x90	0x01	Receive mode	0 = Transport stream
2	0x90	0x03	Frequency (Hertz)	1150000000 = 1150MHz
3	0x90	0x09	LNB power enable	0 = Disabled
A refresh is required to see the receiver status and statistics.				

The DTM-3237B will now receive a DVB-S2X stream and transmit this stream on the ASI port. The status LED will be steady green to indicate successful transmission of the transport stream on the ASI output. If no DVB-S2X stream is received, the status LED will flash green on and off.

[1] If you do not have a suitable DVB-S(2) modulator and/or ASI receiver, this functionality can, for example, be realized with a PC and a DekTec usb modules DTU-315 and DTU-245. Please consult your local DekTec representative for more information.

Below you find a screenshot of the ***Dtm3237BUtil*** after all settings for the "Getting Started" configuration has been applied and the upper left refresh arrow has been pressed to refresh all the settings, receiver status and statistics.



DTM-3237B Utility - View Status and Control Parameters v1.4.1.5

Communication port

COM port: COM5 Status: Settings stored in FLASH memory

Cat 0x01 - Device properties

1. FPGA version: v0
3. Firmware version: v0
4. S/N: 3237.200.016
5. Type: DTM-3237B
6. Hardware revision: v1
7. Bootloader version: v0
8. Production date: 2023.5
9. I2C address: 0x60
See manual for options

Cat 0x90 - Receiver settings

3. Frequency: 1550000000
9. LNB enable: 1
0 = Disabled; 1 = Enabled
10. LNB voltage: 1
0 = 13V; 1 = 14V
2 = 18V; 3 = 19V
11. LNB 22kHz tone: 0
0 = Disabled; 1 = Enabled
12. DVB-S2 ISI: 0
14. Tuner mode: 0
0=Auto; 1=Set Bandwidth
2=Man DVB-S; 3= Man DVB-S2/X
5. Symbol Rate:
15. Tuner Bandwidth:

Cat 0x91 - Receiver status

1. Locked: TRUE
2. Freq: 1550.000.000
3. BW: 48.601.200
4. ModType: DVB-S2 32APSK
5. Code rate: 9/10
6. Roll-off: 20 %
7. Frame length: NORMAL
8. Pilots NO PILOTS
9. Spectrum inv.: NORMAL
10. SymRate: 40.501.000
11. Bitrate: 180.350.000

Cat 0x92 - Receiver statistics

1. Power-level: -11.50dBm
2. SNR: 26.80dB
3. MER: 26.80dB
4. Es/N0: 26.80dB
5. Eb/N0: 19.70dB
6. Link margin: 10.70dB
7. BER pre-Viterbi: N/A
8. BER post-Vit: N/A
9. BER pre-RS: N/A
10. BER pre-LDPC: 1.42e-10
11. BER post-LDPC: 0.00e+00
12. BER post-BCH: 0.00e+00
13. AGC 164

Cat 0x94 - Satellite Information

1. State: No TS lock
2. TS-ID: -
3. Network ID: -
4. Frequency: -
5. Orbital Position: -
6. West East Flag: -
7. Polarization: -
8. Roll-off: -
9. Mod. system: -
10. Mod. type: -
11. Symbol rate: -
12. FEC inner: -

Device picture

Cat 0x02 - Overall configuration

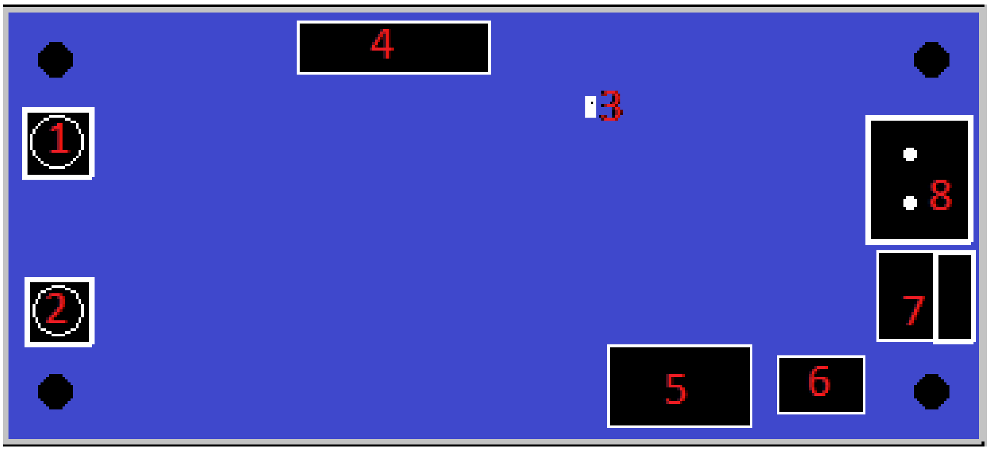
3. Volatile settings: 1
0 = Settings non-volatile; 1 = Settings volatile
4. Persist all settings: Store in FLASH
5. Number of writes to flash memory: 99

Cat 0x85 - Firmware upgrade type 2 device

1. Mode: Normal operation

3. Layout and Installation

3.1 Physical layout



#	Field	Connector type	Description
1	**RF input**	MCX 75Ω	DVB-S2X input
2	**ASI output**	MCX 75Ω	DVB-ASI output
3	**Status LED**		DTM-3237B Status LED
4	**Identifier**		Type and revision number
5	**Control**	10-pin header 2.54mm pitch	I^{2}C interface for board control
6	**Power**	6-pin header 2.54mm pitch	Power and reset
7	**Power**	Molex KK series 2.54mm pitch	Power and reset
8	**USB**	Female mini-B	USB interface for board control

3.2 Mechanical Dimensions

See Appendix A.

3.3 Order Codes

Order Code	Picture	Description
DTM-3237B		DTM-3237B – OEM DVB-S2 receiver with DVB-ASI output
DTM-3237B-DEVKIT	The DTM-3237B development kit contains the following items: • DTM-3237B placed on four plastic studs. • 12V/1.5A power supply with three-way Molex KK connector. • USB cable type A to mini B. • MCX to F female cable assembly with a length of 130 mm. • MCX to BNC cable assembly with a length of 130 mm.	

3.4 Hardware Installation

3.4.1 Mechanical Installation

The unit can be mounted onto a support plate by means of four 3 mm bolts and appropriate spacers. Ensure that there is sufficient airflow to provide cooling of the board.

3.4.2 RF Connector

RF connector (1) is a MCX connector with an impedance of 75 ohm.

3.4.3 ASI Connector

ASI connector (2) is a MCX connector with an impedance of 75 ohm.

3.4.4 Control Connector

The pinning of the control connector is shown in the table below. It's a dual row pin header for connecting the I^2C control bus, lock indicator and analog signal level.

Pin	Function	Pin	Function
1	NC	2	Analog Signal level
3	NC	4	Lock indicator
5	NC	6	NC
7	GND	8	I^2C SCL
9	I^2C SDA	10	I^2C SCL

An I^2C controller can be connected to SDA and SCL on pin 9 and 8/10, with signal ground on pin 7.

3.4.5 Power Connector

The DTM-3237B must be powered from an external source with a voltage of 12V DC. Power consumption is max. 5W without the LNB power enabled. Two power connectors are available, connector 6 and 7. Please refer to §3.1- Physical layout for the connector layout. The pinning of these power connectors is shown below.

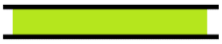
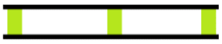
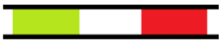
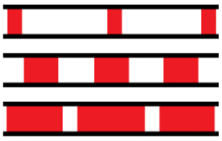
Connector 6 – Pin header 2.54 mm pitch			
Pin	Function	Pin	Function
1	+12V DC in	2	+12V DC in
3	Ground	4	Ground
4	Reset	6	Reset

Connector 7 - Molex KK series 2.54 mm pitch	
Pin	Function
1	+12V DC in
2	Ground
3	Reset

The board can be reset by pulling the reset pin to ground for at least 100ns. The reset pin is connected with a resistor to the 12V DC input voltage. When connecting a driver to the reset pin of the DTM-3237B, make sure it is 12V tolerant. To trigger a reset, the voltage on the reset pin shall be 700mV or less.

3.4.6 Stream Status LED

The status LED indicates the status of the DVB-S2X receiver and the ASI output stream. The following colors are used for status indication:

	Continuous green	Valid DVB-S2X signal detected and generating signal on DVB-ASI output
	Short green flashes	No DVB-S2X signal detected and no output generated on DVB-ASI output
	Short red/green flashes	The DTM-3237B is in firmware upgrade mode. This modes allows: 1) Upgrading the firmware and go back to normal operation afterwards 2) Go to normal operation (in case valid firmware is present)
	Red flashes	Internal device error. If resetting the device does not help, contact DekTec support.

4. Device Configuration and Monitoring

4.1 Control Interfaces

The DTM-3237B can be configured and monitored via a serial control protocol on USB or I^{2}C. The USB interface uses a serial port emulator running at 256kBaud. It is not required to select between the control ports, the DTM-3237B will automatically use the interface on which it detects activity.

The I^{2}C address is a special parameter that can only be configured through the USB interface (e.g. with ***Dtm3237BUtil***). Alternatively, the defaults can be used:

- The default I^{2}C address is 0x60.

The USB uses the control protocol that is described below. I^{2}C uses the same protocol, but slightly different message formatting.

4.2 Control Protocol

Commands and responses are wrapped into messages described below. The DTM-3237B accepts uppercase and lowercase characters, but will always respond in uppercase.

4.2.1 Message Format on USB

Field	Format	Description
Start	ASCII character STX (0x02)	ASCII "start of text" character
Address	2 hex digits[2]	8-bit address[3]
Category	2 hex digits	Selects a "category" of settings
Setting	2 hex digits	Selects a setting within the selected category
Read/Write	ASCII character 'R' or 'W'	'R' for read and 'W' for write
Index	4 hex digits	Provides an extra index parameter, e.g. to indicate the channel number[4]
Data	n hex digits / n ASCII characters	The data written or read. The data length is variable for each setting. In case of a write operation, the data is a (negative) acknowledgement
Checksum	2 hex digits	This is the least significant byte of the two's complement[5] sum of all characters in the message, excluding the STX and ETX characters and the checksum itself
End	ASCII character ETX (0x03)	

[2] Hex digits are the ASCII characters 0…9 and A…F, concatenated to form a single hexadecimal value.

[3] USB and LVTTL serial are point-to-point connections, so there is no address and the address field is a don't care.

[4] The DTM-3237B supports a single channel only, so when index is used as a channel number, it's always 0.

[5] Invert all bits and add one.

Figure 2 below shows the structure of a command or response message written through the serial interface. If the command is a 'read', the data may be omitted. Note that some settings require an index.

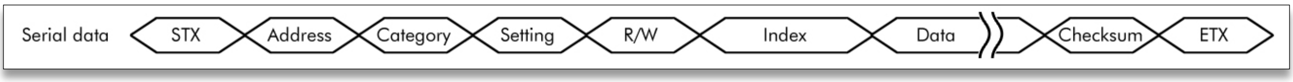


Figure 2. Message on an USB serial control interface

All commands successfully sent to the DTM-3237B are answered with a copy of the command including the data bytes.

When an incorrect checksum or an invalid hex value is detected, the DTM-3237B will not return an answer. When protocol errors are detected, e.g. a combination of a valid category with an invalid setting, the R/W byte of the reply is replaced with the ASCII character ‘E’ and the data is removed from the message.

Please refer to Appendix D for a communication example.

4.2.2 Message Format on I²C

Field	Format	Description
Start	S	Standard I²C start condition
Address	I²C address byte	7-bit I²C address followed by the I²C R/W bit, which is set to 0 and 1 in the command- and response sequence respectively
Category	1 byte	Selects a "category" of settings
Setting	1 byte	Selects a setting within the selected category
Read/Write	1 byte	0x01 for read and 0x00 for write
Index	2 bytes	Provides an extra index parameter, e.g. to indicate the channel number[6]
Data	n bytes	The data written or read. The data length is variable for each setting. In case of a write operation, the actual data is returned as a (negative) acknowledgement
Checksum	1 byte	This is the least significant byte of the two's complement of the sum of the 7-bit I2C slave address and all data-bytes in the I2C message (excluding the checksum). The I2C R/W bit is not included, an incorrect value of this bit would cause the checksum to be not received at all.
End	P	Standard I²C stop condition. A repeated start condition can be used at all times to concatenate multiple I²C read / write actions

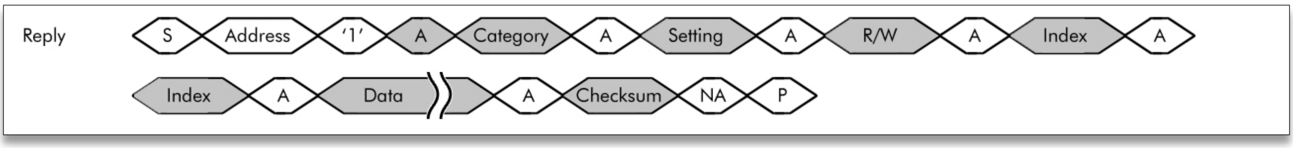


Figure 3 below shows the sequence to send a command over I²C to the DTM-3237B. In the examples below, grey areas in the timing diagrams are sent by the DTM-3237B, while white areas are sent by the I²C master. If the command is a read-command, the data may be omitted. Note that some settings require an index.

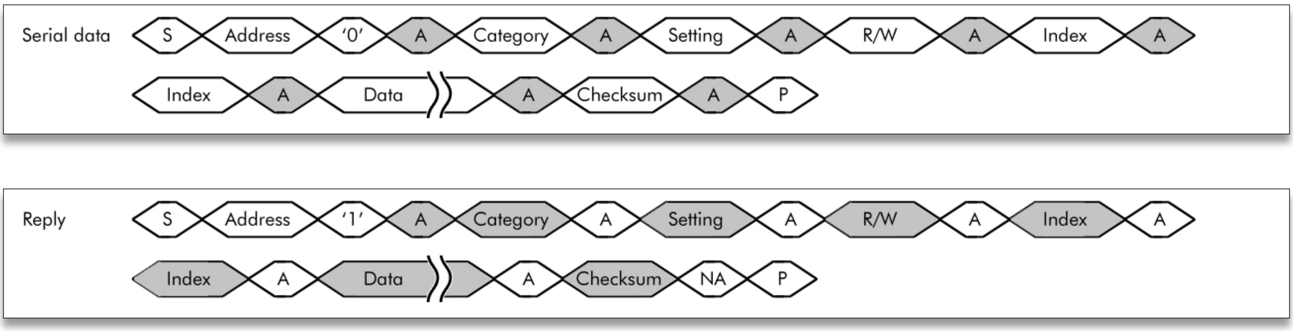


Figure 3. Command (upper sequence) and response (lower) sequence for I²C

[6] The DTM-3237B supports a single channel only, so when index is used as a channel number, it's always 0.

When an incorrect checksum is detected, the DTM-3237B will not return an answer. When a protocol error is detected, e.g. a combination of a valid category with an invalid setting, the R/W byte of the reply is replaced by the ASCII character 'E' and the data is removed from the message.

When a master starts writing to a device while the previous command is still being executed, the device will ignore the data.

When a master starts reading from a device while there is no answer available (yet), the device will reply with the value 0x00. The value 0x00 will be returned until an answer is available and the master has initiated a new read transaction.

S and P are the standard I^{2}C start and stop conditions.

Please refer to Appendix D for a communication example.

4.3 Manageable items

The tables in the sections below provide lists of variables that can be configured and/or monitored using USB or I^{2}C.

The 'Access' column indicates whether the variable is writeable, according to the table below.

Access	Description
RO	Read only
WO	Write only
R/W	Read and/or write
R/Wusb	Read from all interfaces, write from USB only
NA	Not applicable
DE-XXX	Delayed execution (see §4.4 - Delayed Execution)

4.4 Delayed Execution

Most items are processed directly when the read or write command is received, which will immediately result in an action on the device. In contrast to these directly processed items, some items require more processing time and therefor these actions are performed with a delay in a background thread. Items which execution is delayed are marked with a "DE" (delayed execution) prefix in the access column.

Items which execution is delayed can be monitored by the "Busy" item (category 0x01, setting 0x0C). During the execution of these items, the "Busy" item is set and it is not possible to execute other "DE" marked items. In case the "Busy" item is set and a "DE" marked command is received, the device will return with a 'B' (busy) result in the Read/Write field of the DTM protocol frame structure.

4.5 Categories

Manageable Items – Categories			
Nr	**Settings Category**	**Description**	**Index?**
0x01	**Device**	Device properties	No
0x02	**Configuration**	Overall configuration	No
0x85	**Firmware upgrade**	Firmware upgrade type 2	Yes
0x90	**Receiver settings**	Receiver settings	Yes
0x91	**Receiver status**	Receiver status	Yes
0x92	**Receiver statistics**	Receiver statistics	Yes
0x93	**DiSEqC**	DiSEqC send and receive commands	Yes
0x94	**Satellite Info**	Satellite Information	Yes

4.5.1 Data Types

Manageable Items – Data types			
Type	**Description**	**LVTTL serial control port**	**I²C**
uint8	8-bit unsigned integer	2 chars	1 byte
int32	32-bit signed integer	8 chars	4 bytes
uint32	32-bit unsigned integer	8 chars	4 bytes
uint64	64-bit unsigned integer	16 chars	8 bytes

All data types are sent with the most-significant byte first.

4.5.2 Device Properties

Manageable Items – Category 0x01 – Device properties				
Nr	**Variable**	**Description**	**Access**	**Type**
0x00	**All**	All settings in this category	RO	25 bytes
0x01	**FPGA version**	Version number of the FPGA code on-board of the DTM-3237B	RO	uint8
0x03	**Firmware version**	Firmware version: the major version is encoded in the tens, the minor version in the units, e.g. '10' indicates v1.0	RO	uint32
0x04	**Serial number**	Unique serial number for this device, e.g. 3237.000.010	RO	uint32
0x05	**Type**	Device type number, e.g. 3237	RO	uint32
0x06	**Hardware revision**	Hardware revision number, e.g. 302 = 3.2	RO	uint32
0x07	**Bootloader version**	Bootloader version number	RO	uint8
0x08	**Production date**	Production date of this board Bit 31..8: Year Bit 7..0: Month	RO	uint32
0x09	**I^{2}C address**	7-bit I^{2}C address Limitations: - Bit 0 (I^{2}C R/W bit) can't be set - Address range 0x08 to 0x76 Changes to the I^{2}C address are automatically persisted in flash memory. Default: 0x60	R/Wusb	uint8
0x0B	**Subtype**	Device subtype, e.g.0=none, 1=A, ...	RO	uint8
0x0C	**Busy**	Device busy flag 0 = Ready 1 = Processing "DE" marked items	RO	uint8

4.5.3 Overall Configuration

Manageable Items – Category 0x02 – Overall configuration				
Nr	Variable	Description	Access	Type
0x00	**All**	All settings in this category	RO	5 bytes
0x03	**Volatile settings**	0 = Settings are persisted in flash memory [7] 1* = Settings are volatile (not persisted in flash memory)	R/W	uint8
0x04	**Persist all settings**	Store all current settings in flash memory [7] Data: don't care	WO	uint8
0x05	**Number of writes to flash memory**	Number of times the settings have been written to flash memory	RO	uint32

* Factory default

4.5.4 Firmware Upgrade

The settings in the ***Firmware upgrade*** category can be used to erase the current firmware, to upload new firmware, to program new firmware into flash memory and to verify the uploaded firmware.

To upgrade the DTM-3237B, setting ***Mode*** (0x01) needs to be set to '1' (Firmware upgrade mode). When the USB interface is used and ***Mode*** is changed, a USB reconnect is necessary.

The new firmware has to be uploaded in "file parts". For communication through I^{2}C the parts may contain at most 250 data-bytes; for communication through USB or LVTTL serial control port the parts may contain at most 1000 data bytes.

An example of a firmware upgrade sequence can be found in paragraph 4.6.

Manageable Items – Category 0x85 – Firmware upgrade type 2				
Nr	Variable	Description	Access	Type
0x01	**Mode**	0* = Normal operation 1 = Firmware upgrade mode	DE-R/W	uint8
0x02	**Erase ****	Erase the firmware. Data: Don't care	DE-WO	uint8
0x03	**Programming data ****	Data to be programmed into flash memory. The data is immediately written to flash memory. Index 0 indicates the start of a new firmware file.	DE-WO	1…1000 bytes
0x04	**Verify ****	Verify the firmware based on start address, length and CRC 0 = Firmware is not uploaded correctly 1 = Firmware is uploaded correctly	DE-RO	uint8

* Factory default

[7] Flash memory endurance is min. 1000 writes, therefor only store the settings in flash memory when necessary.

** This command can only be used in firmware upgrade mode (see category 0x85, setting 0x01).

4.5.5 Receiver Settings

Manageable Items – Category 0x90 – Receiver settings				
Nr	Variable	Description	Access	Type
0x00	**All**	All settings in this category	RO	14 bytes
0x03	**Frequency**	Frequency in Hertz Range: 950000000…2150000000	R/W	uint32
0x05	**Symbol Rate**	Symbol rate in BPS (only used when tuner mode is set to DVB-S or DVB-S2)	R/W	uint32
0x09	**LNB enable**	LNB power enabled/disabled 0* = Disabled 1 = Enabled	DE-R/W	uint8
0x0A	**LNB voltage**	LNB voltage 0* = 13V 1 = 14V 2 = 18V 3 = 19V	DE-R/W	uint8
0x0B	**LNB tone**	LNB 22kHz tune enabled/disabled 0* = Disabled 1 = Enabled	DE-R/W	uint8
0x0C	**DVB-S2 ISI**	DVB-S2 input stream identifier Used to filter a multiple input stream in case the receive mode is Transport Stream Valid range: 0…255	R/W	uint8
0x0E	**Tuner mode**	Tuner mode 0* = Auto[8] 1 = Set bandwidth[9] 2 = Manual DVB-S[10] 3 = Manual DVB-S2/S2X[11]	R/W	uint8
0x0F	**Tuner bandwidth**	Tuner bandwidth in MHz (only applicable when tuner mode is set to '1' "Set bandwidth") Valid range: 4..40	R/W	uint8

* Factory default

[8] Auto mode is the default and most common mode of operation. In this mode the user enters only the centre frequency of the channel to be desired tune. The tuner and demodulator will all automatically set the symbol rate, bandwidth of the tuner and try to lock to the carrier. When locked all details of the modulation are reported into the receiver status.

[9] Set bandwidth is special mode that allows to set the bandwidth of the tuner. This mode is designed to measure input signal level and AGC level for the set bandwidth of the spectrum. In this mode the user enters the centre frequency of the channel and the bandwidth of the tuner. The demodulator is unlikely to achieve lock and only the power and AGC level measurement will be accurate.

[10] Manual DVB-S mode is a mode that allows to manual set known parameters of the tuner and demodulator for DVB-S signals. In this mode the user enters the centre frequency and symbol rate of the channel to be desired tune. The tuner and demodulator try to lock to the carrier. When locked all details of the modulation are reported into the receiver status.

[11] Manual DVB-S2 mode is a mode that allows to manual set known parameters of the tuner and demodulator for DVB-S2 signals. In this mode the user enters the centre frequency and symbol rate of the channel to be desired tune. The tuner and demodulator try to lock to the carrier. When locked all details of the modulation are reported into the receiver status.

4.5.6 Receiver Status

Manageable Items – Category 0x91 – Receiver status				
Nr	Variable	Description	Access	Type
0x00	**All**	All settings in this category	RO	23 bytes
0x01	**Locked**	Demodulator locked status 0 = No (full) lock 1 = Locked, received data is reliable	RO	uint8
0x02	**Frequency**	Frequency (Hz) 0 = Unknown or receiver not locked	RO	uint32
0x03	**Occupied bandwidth**	Occupied bandwidth (Hz) 0 = Unknown or receiver not locked	RO	uint32
0x04	**Modulation type**	Received modulation type (standard + constellation) 0 = Unknown or receiver not locked 1 = DVB-S QPSK 2 = DVB-S2 QPSK 3 = DVB-S2 8PSK 4 = DVB-S2 16APSK 5 = DVB-S2 32APSK 6 = DVB-S2 Multiple Input Stream 7 = DVB-S2X 8APSK-L 8 = DVB-S2X 16APSK-L 9 = DVB-S2X 32APSK 10 = DVB-S2X 32APSK-L	RO	uint8

Manageable Items – Category 0x91 – Receiver status				
Nr	Variable	Description	Access	Type
0x05	**Code rate**	Detected code rate 0 = Unknown 1 = 1/2 2 = 2/3 3 = 3/4 4 = 4/5 5 = 5/6 6 = 6/7 7 = 7/8 8 = 1/4 9 = 1/3 10 = 2/5 11 = 3/5 12 = 8/9 13 = 9/10 14 = 5/9 15 = 7/9 16 = 8/15 17 = 11/15 18 = 13/18 19 = 9/20 20 = 11/20 21 = 23/36 22 = 25/36 23 = 13/45 24 = 26/45 25 = 28/45 26 = 32/45 27 = 77/90	RO	uint8
0x06	**Roll-off factor**	Roll-off factor in percentage 0 = Unknown or receiver not locked	RO	uint8
0x07	**FEC Frame length**	FEC Frame length 0 = Unknown or receiver not locked 1 = Normal FEC frames	RO	uint8
0x08	**Pilots**	Pilots enabled/disabled 0 = Unknown or receiver not locked 1 = No pilots present 2 = Pilots present	RO	uint8
0x09	**Spectrum inversion**	Spectrum inverted yes/no 0 = Unknown or receiver not locked 1 = Normal 2 = Inverted	RO	uint8
0x0A	**Symbol rate**	Detected symbol rate 0 = Unknown or receiver not locked	RO	uint32

Manageable Items – Category 0x91 – Receiver status				
Nr	Variable	Description	Access	Type
0x0B	**Interface bitrate**	Detected interface bitrate after FEC 0 = Unknown or receiver not locked INT_MIN = N/A for ACM streams	RO	uint32

4.5.7 Receiver Statistics

In DVB-S2, the pre-LDPC BER is the bit error rate before the receiver has applied any error correction. For the DTM-3237B it is be computed from the MER using formulas. These formulas have been validated using DekTec's advanced demodulator simulation software (this software has been used amongst others in the DVB working groups for the definition of DVB-T2 and DVB-C2). The correspondence between theoretical and measured values is very good.

The E_s/N_0 is computed from the MER under the assumption that the noise distribution is Gaussian (AWGN channel), as under these circumstances E_s/N_0 and MER are identical. The E_b/N_0 is computed from the E_s/N_0 for constant modulated (CCM) streams only.

Manageable Items – Category 0x92 – Receiver statistics				
Nr	Variable	Description	Access	Type
0x00	**All**	All settings in this category	RO	76 bytes
0x01	**RF power-level**	RF power-level for channel bandwidth in 0.1 dBmV units	RO	int32
0x02	**SNR**	Signal-over-noise ratio in 0.1 dB units INT_MIN = Unknown or not locked	RO	int32
0x03	**MER**	Modulation-error-rate in 0.1 dB units INT_MIN = Unknown or not locked	RO	int32
0x04	**E_s/N_0**	Energy per symbol per noise power spectral density in 0.1 dB units INT_MIN = Unknown or not locked	RO	int32
0x05	**E_b/N_0**	Energy per bit to noise power spectral density ratio in 0.1 dB units INT_MIN = Unknown or not locked INT_MIN + 1 = N/A for ACM streams	RO	int32
0x06	**Link margin**	Difference in dB between C/N of the received signal and the C/N at which the receiver cannot demodulate the signal any more in 0.1dB units INT_MIN = Unknown or not locked INT_MIN + 1 = N/A for DVB-S2 MIS	RO	int32
0x07	**BER pre-Viterbi**	Pre-Viterbi bit error rate INT_MIN = Unknown or not locked INT_MIN + 1 = N/A for DVB-S2	RO	uint64

Manageable Items – Category 0x92 – Receiver statistics

Nr	Variable	Description	Access	Type
0x08	**BER post-Viterbi**	Post-Viterbi bit error rate INT_MIN = Unknown or not locked INT_MIN + 1 = N/A for DVB-S2	RO	uint64
0x09	**BER pre-RS**	Pre-Reed Solomon bit error rate INT_MIN = Unknown or not locked INT_MIN + 1 = N/A for DVB-S2	RO	uint64
0x0A	**BER pre-LDPC**	Inner LDPC bit error rate (pre-LDPC) INT_MIN = Unknown or not locked INT_MIN + 1 = N/A for DVB-S INT_MIN + 2 = N/A for DVB-S2 MIS	RO	uint64
0x0B	**BER post-LDPC**	Outer LDPC bit error rate (post-LDPC) INT_MIN = Unknown or not locked INT_MIN + 1 = N/A for DVB-S	RO	uint64
0x0C	**BER post-BCH**	Outer BCH bit error rate (post-BCH) INT_MIN = Unknown or not locked INT_MIN + 1 = N/A for DVB-S	RO	uint64
0x0D	**AGC 1**	Tuner AGC value	RO	int32

When a statistic has the output type uint64, the received data can be converted to variable double type with the following C code:

```
unsigned long long  StatisticData;
DtmResult  Result = DtmReadInt64(DtmAddr, CategoryNr,
                                  SettingNr, Index, &StatisticData);
double  Statistic = double(StatisticData) / pow(2.0, 64.0);
```

4.5.8 DiSEqC Send and Receive Options

The manageable items in category 0x93 can be used to send and receive DiSEqC message to/from LNB's, switches or motors. DiSEqC messages can contain as much data as required.

Manageable Items – Category 0x93 – DiSEqC send and receive commands				
Nr	Variable	Description	Access	Type
0x01	**Burst**	Send DiSEqC Burst A/B 0 = Burst type A 1 = Burst type B	DE-WO	uint8
0x02	**Send DiSEqC message**	Send DiSEqC message	DE-WO	1…100 bytes
0x03	**Send DiSEqC message and receive response**	Send DiSEqC message and receive response, which enables the DiSEqC receiver during transmission. The received response can be retrieved through items in this category.	DE-WO	1…100 bytes
0x04	**DiSEqC response message length**	DiSEqC response message size in bytes	RO	uint8
0x05	**DiSEqC response message**	DiSEqC response message data	RO	1…100 bytes

4.5.9 Satellite Information

The read-only items in category 0x94 contains the Transport-Stream ID, Network ID and all fields from the Satellite Delivery System Descriptor if available in the Transport-Stream.

Manageable Items – Category 0x94 – Satellite Information				
Nr	Variable	Description	Access	Type
0x00	**All**	All settings in this category	RO	27 bytes
0x01	**State**	State of the information-extraction engine 0 = No TS-packet lock 1 = TS-packet lock 2 = Transport-Stream-ID field is valid 3 = Network-ID field is valid 4 = Transport-Stream-ID and Network-ID fields are valid 5 = All fields are valid	RO	uint8
0x02	**Transport-Stream ID**	Identification of this Transport-Stream	RO	uint32
0x03	**Network ID**	Identification of this delivery system	RO	uint32

Manageable Items – Category 0x94 – Satellite Information				
Nr	Variable	Description	Access	Type
0x04	**Frequency**	Specifies 8 characters of the frequency value, coded in 4-bit BCD numbers. The frequency is coded in GHz, where the decimal point occurs after the third character (e.g. 011,75725 GHz).	RO	uint32
0x05	**Orbital Position**	Specifies 4 characters of the orbital position value, coded in 4-bit BCD numbers. The orbital position is coded in degrees, where the decimal point occurs after the third character (e.g. 019.2 degrees).	RO	uint32
0x06	**West East Flag**	Indicating if the satellite position is in the western or eastern part of the orbit 0 = Western position 1 = Eastern position	RO	uint8
0x07	**Polarization**	Polarization of the transmitted signal 0 = Linear - horizontal 1 = Linear - vertical 2 = Circular - left 3 = Circular – right	RO	uint8
0x08	**Roll-off factor**	Roll-off factor used in DVB-S2 0 = 35% 1 = 25% 2 = 20% 3 = Reserved	RO	uint8
0x09	**Modulation system**	Modulation system 0 = DVB-S 1 = DVB-S2	RO	uint8
0x0A	**Modulation type**	Modulation scheme used 0 = Auto 1 = QPSK 2 = 8PSK 3 = 16-QAM (n/a for DVB-S2)	RO	uint8
0x0B	**Symbol rate**	Specifies 7 characters of the symbol rate value, coded in 4-bit BCD numbers. The symbol rate is coded in Msymbols per second, where the decimal point occurs after the third character (e.g. 027,4500 Msps).	RO	uint32

Manageable Items – Category 0x94 – Satellite Information				
Nr	Variable	Description	Access	Type
0x0C	**FEC inner**	Forward Error Correction (FEC) scheme used 0 = Not defined 1 = 1/2 2 = 2/3 3 = 3/4 4 = 5/6 5 = 7/8 6 = 8/9 7 = 3/5 8 = 4/5 9 = 9/10 10 = Reserved 11 = Reserved 12 = Reserved 13 = Reserved 14 = Reserved 15 = No convolutional coding	RO	uint8

4.6 Firmware Upgrade

4.6.1 Firmware Upgrade - Phases

Updating the firmware of the device consists of five phases:

1. Put the device in firmware upgrade mode by setting Mode (0x85, 0x01) to 1.
2. Erase the current firmware by writing any value to setting Erase (0x85, 0x02).
3. Upload the firmware. The file has to be uploaded in “parts” to setting Programming data (0x85, 0x03), were each part may contain 250 (I^2C) to 1.000 (LVTTL serial control port) bytes.
4. Verify the uploaded firmware by reading from setting Verify (0x85, 0x04).
5. When the firmware upgrade is successfully completed, the DTM-3237B should be rebooted to make the upgrade effective, by putting the device in normal operation by setting Mode (0x85, 0x01) to 0.

After the device has been upgraded and rebooted, the DTM-3237B checks the status of the firmware. In case the firmware isn’t correctly uploaded, the DTM-3237B will stay in firmware upgrade mode. In this mode the firmware must be uploaded again, starting at the beginning of the firmware file.

When using the USB interface as communication port, care should be taken with entering firmware upgrade mode or rebooting the DTM-3237B, since the USB connection is closed in between. To go from normal operation to firmware upgrade mode or vice versa, preform these steps:

1. Send the command for switching the mode
2. Close the USB handle within 500ms after sending the mode switch command
3. Wait for the USB interface to disappear and reinitialize, before reopening the USB handle.

The entire firmware upgrade process is implemented in ***DtmHandler***. Please refer to ***DtmCmd*** (appendix C) for an example of how to implement the firmware upgrade using ***DtmHandler***.

4.6.2 Firmware Upgrade - Data Encoding

To improve the firmware upgrade speed, the following encoding is used for the data part of the "Programming data" setting (category 0x85, setting 0x03) when using the LVTTL serial control port and USB interface.

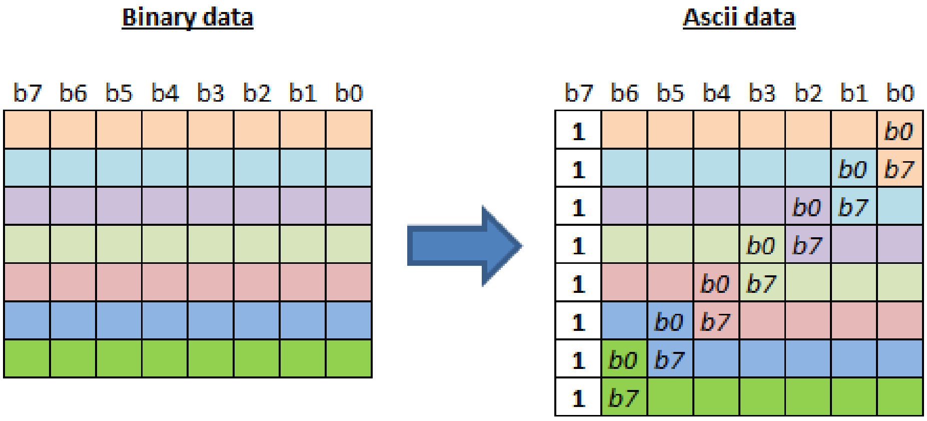


Figure 4. ASCII encoding for firmware upgrade data

For each 7 bits of data one 8-bit ASCII character is sent, where the MSB of the ASCII character is set to 1 (extended ASCII range). The translation is illustrated in Figure 4.

4.6.3 Firmware Upload - Example

In the example below the DTM-3237B is upgraded with new firmware over USB. The size of the firmware file is 486400 bytes. Over USB, each packet can hold a maximum of 250 data bytes, so the number of ‘file parts’ will 1946. The first 1945 file parts will contain 250 data bytes and the last part will contain 150 data bytes. The table below shows the actions required to perform this firmware upgrade.

Action	Setting	R/W	Idx	Comment
Switch mode to 1	0x85, 0x01	W	0	Switch to firmware-upgrade mode
Close USB interface				USB only
Wait 3s				
Reconnect USB i/f				USB only
Read mode	0x85, 0x01	R	0	Verify that mode is firmware-upgrade mode (1)
Erase	0x85, 0x02	W	0	The DTM-3237B needs ±3.5 seconds processing time to erase the firmware
File part*	0x85, 0x03	W	1	
File part	0x85, 0x03	W	2	
: :	0x85, 0x03	W	: :	
File part	0x85, 0x03	W	1946	

Action	Setting	R/W	Idx	Comment
Verify	0x85, 0x04	R	0	We expect to read 1 (=firmware uploaded correctly)
Switch mode to 0	0x85, 0x01	W	0	Switch to normal mode
Close USB interface				USB only
Wait 10s				
Reconnect USB i/f				USB only
Read mode	0x85, 0x01	R	0	Verify that mode is normal (0)

* For USB, data is encoded as described in §4.6.2. For LVTTL serial control port the same encoding is used. For I²C, data bytes are packaged without conversion in I²C write message.

5. Specifications

5.1 RF Input

	Min	Typ	Max	Unit / Remarks
Standard				
DVB-S		EN 301 210		
DVB-S2		EN 302 307		
DVB-S2X		EN 302 307-2		
RF input				
Connector type		MCX female		
Impedance		75		Ω
Return loss	>9			dB (950 .. 2150MHz)
Tuning range	950		2150	MHz
Sensitivity	-70		-20	dBm
Baud Rate	1		54	MBd (see table below)
Metrology				
RF level	-70		-20	dBm
RF level accuracy		3		dBm
MER			22	dB
MER accuracy		2		dB
LNB interface				
LNB supply		13V/14V/18V/19V 400mA		
LNB short circuit protection		500 ±10%		mA
22kHz tone		On/Off		

5.2 Supported Modulation Parameters

The table below specifies the modulation standards, modes, code rates and symbol rates that the DTM-3237B can properly receive.

			Symbol Rate	
Standard	Modulation	Code rate	Min (MBd)	Max (MBd)
DVB-S	QPSK	1/2, 2/3, 3/4, 5/6, 7/8	1.0	54.0
DVB-S2	QPSK	1/2, 2/3, 3/4, 3/5, 4/5, 5/6, 8/9, 9/10	1.0	54.0
	8-PSK	2/3, 3/4, 3/5, 5/6, 8/9, 9/10	1.0	54.0
	16-APSK	2/3, 3/4, 4/5, 5/6, 8/9, 9/10	1.0	50.7
	32-APSK	3/4, 4/5, 5/6, 8/9	1.0	40.6
		9/10	1.0	40.5
DVB-S2X	QPSK	13/45, 9/20, 11/20	1.0	54.0
	8-APSK-L	5/9, 26/45	1.0	54.0
	8-PSK	23/36, 25/36, 13/18	1.0	54.0
	16-APSK-L	1/2, 8/15, 5/9, 3/5, 2/3	1.0	50.7
	16-APSK	26/45, 3/5, 28/45, 23/36, 13/18, 7/9, 77/90	1.0	50.7
	32-APSK-L	2/3	1.0	40.6
	32-APSK	32/45, 11/15, 7/9	1.0	40.6

5.3 DVB-ASI Input

	Min	Typ	Max	Unit / Remarks
Standard				
DVB-ASI		EN50083-9		
Ports				
Connector		75-Ω MCX		
Return loss		15		dB
Error-free cable length	100			m
ASI bitrate		Matching DVB-S2X bitrate		
Packet size		188 bytes		In TS mode
Receive modes		TS		

5.4 I²C Control Port

	Min	Typ	Max	Unit / Remark
Interface port				
Connector		10-way pin header, 2.54 mm pitch		Signals available on pin 9-10 of serial port control connector
Signals		SDA/SCL		
Serial format				
Interface voltage		3.3		V
Speed			400	kbit/s
Device address	0x08	0x60	0x76	Configurable through command protocol

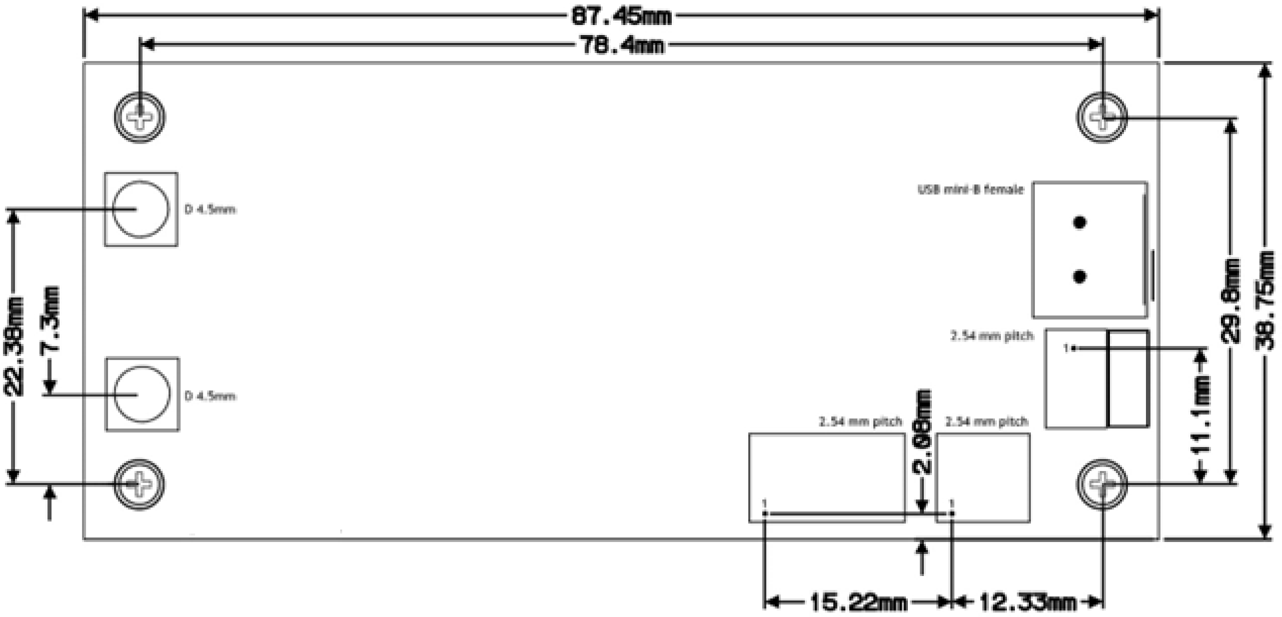
5.5 USB Control Port

	Min	Typ	Max	Unit / Remark
Interface port				
Connector		Mini USB B		
Signals		USB 2		Serial emulator
Format		8 bit, one stop bit, no parity		
Handshaking		hardware flow control		
Speed		256000		Baud

5.6 Other Specifications

	Min	Typ	Max	Unit / Remarks
Power				
Power Supply Voltage	10.8	12	13.2	V
Connector 1		Right-angle 3-pin Molex KK		
Connector 2		Dual row 3-way pin header, 2.54 mm pitch		
Power Consumption excluding LNB			2	W
Power Consumption including LNB			12	W (LNB @ 400mA/19V)
Environmental				
Hazardous Substances		RoHS compliant		
Flammability		UL-94 HB		
Operational Temperature	0		≥+45	°C
Mechanical				
Mounting		3		mm, four mounting holes
Dimensions W x H x D		87.45 x 38.75 x 17.9		mm (max)
Weight		26		g

Appendix A. Mechanical Dimensions



Mounting holes:

- The diameter of the mounting holes is 3.2mm (intended for M3 bolts)
- Maximum outer diameter of nut/ring: 7.5mm

Warning: While mounting the DTM-3237B, care should be taken not to damage components that are close to the mounting holes, both on the top and bottom side of the board.

Appendix B. DTM-3237B Development Kit

B.1 DTM-3237B Development Kit – Contents

The DTM-3237B development kit contains the following items:

- DTM-3237B placed on four plastic studs
- 12V/1.5A power supply
- USB cable type A to mini B
- MCX to F female cable assembly (length = 130mm)
- MCX to BNC cable assembly (length = 130mm)
- DekTec USB flash drive containing DTM-3237B documentation and development tools (as well as documentation on DekTec's other products)

The development kit can be ordered from DekTec using type number DTM-3237B-DEVKIT.

B.2 Using the DTM-3237B Development Kit

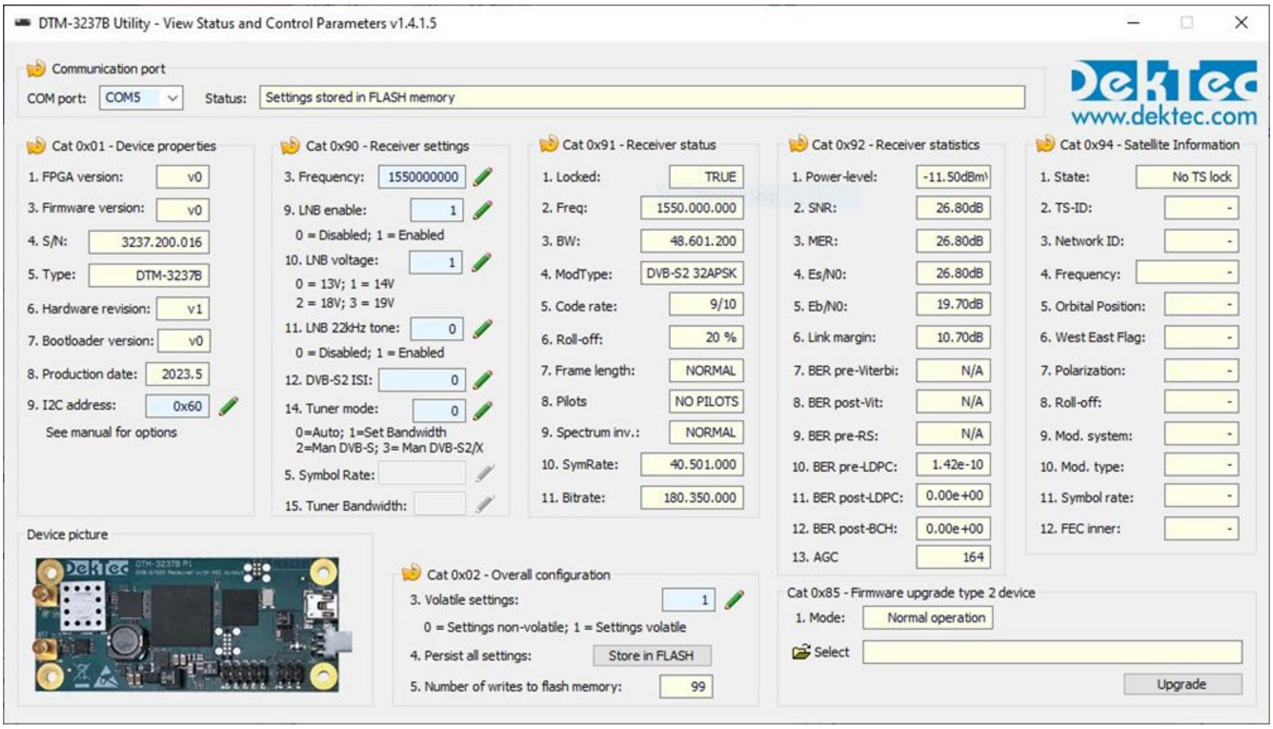
B.2.1 Hardware Installation

The DTM-3237B board has to be connected with the USB cable to a development PC. Connect the DTM-3237B to the power-supply using the power connector. The DTM-3237B will boot which will take a few seconds. During this time the LEDs on the DTM-3237B are flashing in a start-up pattern. Wait until the DTM-3237B status LED turns (blinking) green.

B.2.2 Debugger

Dtm3237BUtil is a GUI tool to view status, control settings and upload firmware to the DTM-3237B. This utility can be found on the DekTec USB flash drive. It can also be downloaded from the DekTec website. ***Dtm3237BUtil*** is an executable that can be run from any directory on your PC.

When started, the debugger enumerates serial ports and lets the user select the serial port to which the DTM-3237B is connected. When a valid serial port is selected, all registers are read from the DTM-3237B and shown in the GUI. Blue fields can be edited. These fields are written to the DTM-3237B when the pen symbol to the right of the edit fields is clicked. Yellow fields are read only; they are read when the refresh arrow is clicked.



DTM-3237B Utility - View Status and Control Parameters v1.4.1.5

Communication port

COM port: COM5

Status: Settings stored in FLASH memory

Cat 0x01 - Device properties

1. FPGA version: v0
3. Firmware version: v0
4. S/N: 3237.200.016
5. Type: DTM-3237B
6. Hardware revision: v1
7. Bootloader version: v0
8. Production date: 2023.5
9. I2C address: 0x60

See manual for options

Cat 0x90 - Receiver settings

3. Frequency: 1550000000
9. LNB enable: 1

0 = Disabled; 1 = Enabled

10. LNB voltage: 1

0 = 13V; 1 = 14V
2 = 18V; 3 = 19V

11. LNB 22kHz tone: 0

0 = Disabled; 1 = Enabled

12. DVB-S2 ISI: 0
14. Tuner mode: 0

0=Auto; 1=Set Bandwidth
2=Man DVB-S; 3= Man DVB-S2/X

5. Symbol Rate:
15. Tuner Bandwidth:

Cat 0x91 - Receiver status

1. Locked: TRUE
2. Freq: 1550.000.000
3. BW: 48.601.200
4. ModType: DVB-S2 32APSK
5. Code rate: 9/10
6. Roll-off: 20 %
7. Frame length: NORMAL
8. Pilots NO PILOTS
9. Spectrum inv.: NORMAL
10. SymRate: 40.501.000
11. Bitrate: 180.350.000

Cat 0x92 - Receiver statistics

1. Power-level: -11.50dBm\
2. SNR: 26.80dB
3. MER: 26.80dB
4. Es/N0: 26.80dB
5. Eb/N0: 19.70dB
6. Link margin: 10.70dB
7. BER pre-Viterbi: N/A
8. BER post-Vit: N/A
9. BER pre-RS: N/A
10. BER pre-LDPC: 1.42e-10
11. BER post-LDPC: 0.00e+00
12. BER post-BCH: 0.00e+00
13. AGC 164

Cat 0x94 - Satellite Information

1. State: No TS lock
2. TS-ID: -
3. Network ID: -
4. Frequency: -
5. Orbital Position: -
6. West East Flag: -
7. Polarization: -
8. Roll-off: -
9. Mod. system: -
10. Mod. type: -
11. Symbol rate: -
12. FEC inner: -

Device picture

Cat 0x02 - Overall configuration

3. Volatile settings: 1

0 = Settings non-volatile; 1 = Settings volatile

4. Persist all settings: Store in FLASH
5. Number of writes to flash memory: 99

Cat 0x85 - Firmware upgrade type 2 device

1. Mode: Normal operation

Appendix C. Command-Line Tool - DtmCmd

C.1 General Description of DtmCmd

DtmCmd is a cross-platform (Windows & Linux) command-line tool for simple control of DTM-32xx devices. The user can read and write device settings, e.g. the command "`DtmCmd -r 1 5`" reads and prints the value of setting 5 in settings category 1. The most advanced capability of ***DtmCmd*** is upgrading the firmware of a DTM-32xx device. ***DtmCmd*** comes with a Microsoft Visual Studio 2010 project for Windows and Makefile for Linux.

To use the command-line tool under Windows, open a DOS box in a directory containing the ***DtmCmd*** executable. Each time ***DtmCmd*** is run, a single command specified with the command-line arguments is executed on the DTM-3237B. See "`DtmCmd -?`" for help on the available commands.

You can specify the interface type (serial/I^{2}C), interface settings and DTM address on the command line. The configuration settings are stored in file **`DtmCmd.ini`**. Every time ***DtmCmd*** starts, it first reads **`DtmCmd.ini`**, so that you don't need to specify the configuration settings every run of ***DtmCmd***.

Please note that the USB interface on the DTM-3237B is a serial emulator and therefor the serial interface needs to be used in ***DtmCmd***. Please refer to §4.1 - Control Interfaces for more information.

C.2 Reading a Setting from the DTM-3237B

The following command reads device property ***Type*** (category 1, setting 5):

```
DtmCmd -interface Serial -serial COM3 -baudrate 9600 -addr 0x60
       -r 1 5
```

The following shortcut is equivalent once the configuration settings are available in **`DtmCmd.ini`**:

```
DtmCmd -r 1 5
```

The parameters used in this command have the following meanings:

- `-interface Serial` → Set the interface type to a serial COM port. The I^{2}C interface is also supported by ***DtmCmd***.
- `-serial COM3` → Set the serial COM port identifier to COM port 3.
- `-baudrate 9600` → Set the baud rate to 9600bd.
- `-addr 0x60` → Set the address of the DTM-3237B to 0x60.
- `-r` → Set the command type to read.
- `1 5` → Specify command category 1 and setting 5.

This command results into the following output when using the `-v` parameter (verbose mode):

```
DtmCmd - DTM-32xx Command Utility v1.0.1 (c) 2023 DekTec Digital Video

- Category          : 0x01 (Device properties)
- Setting           : 0x05 (Type)
- Index             : 0x00
- Interface         : Serial
- DTM address       : 0x60
- Serial path       : COM3
- Serial baud       : 9600
- Data read         : 3237
```

C.3 Writing to the DTM-3237B

To demonstrate the writing of a setting, we write 2150000000 to the setting ***Frequency*** (setting 3) of category ***Tuner and demodulator settings*** (category 0x90). The command below assumes that configuration settings are available in **`DtmCmd.ini`:**

```
DtmCmd -w 0x90 3 2150000000
```

The parameters used in this command have the following meaning:

- `-i 0` → Set the index to 0.
- `-w` → Set the command type to write.
- `0x90 3 2150000000` → Specify command category 0x90, setting 3 and data 2150000000

This command results into the following output when using the `-v` parameter (verbose mode):

```
DtmCmd - DTM-32xx Command Utility v1.0.1 (c) 2013 DekTec Digital Video

- Category          : 0x90 (Tuner and demodulator settings)
- Setting           : 0x03 (Frequency)
- Index             : 0x00
- Interface         : Serial
- DTM address       : 0x60
- Serial path       : COM3
- Serial baud       : 9600

- Data written      : 2150000000
```

C.4 Upgrading the DTM-3237B's Firmware

To demonstrate the firmware upgrade process, we pass the filename of the firmware file to ***DtmCmd***. The USB parameter (-usb) must be passed in case the USB interface is used. Please refer to §4.6.1 - Firmware Upgrade - Phases for more information about upgrading the firmware using the USB interface. The command below assumes that configuration settings are available in **`DtmCmd.ini`:**

```
DtmCmd -upgrade Dtm3237FwV0.dtm          (for interface: I2C or serial)
DtmCmd -upgrade Dtm3237FwV0.dtm -usb     (for interface: USB)
```

The parameters used in this command have the following meaning:

- `-upgrade` → Set to the filename of the firmware file.
- Optional for USB: `-usb` → Set to indicate the USB interface is being used.

This command results into the following output when using the `-v` parameter (verbose mode):

```
DtmCmd - DTM-32xx Command Utility v1.0.1 (c) 2013 DekTec Digital Video
B.V.

- Firmware upgrade : Dtm3237FwV0.dtm
- Interface         : serial
- DTM address       : 0x60
- Serial path       : COM3
- Serial baud       : 256000

Current phase: Finished, progress: 100
```

Appendix D. Communication Example

In the examples below, grey areas in the timing diagrams are sent by the DTM-3237B, while white areas are sent by the master. The I^{2}C address of the DTM-3237B in these examples is ***0x60***.

Serial write command on USB interface

Figure 5 shows the write command of the LNB enable at the tuner and demodulator settings (category 90, setting 9, index 0). All values are displayed as ASCII characters.

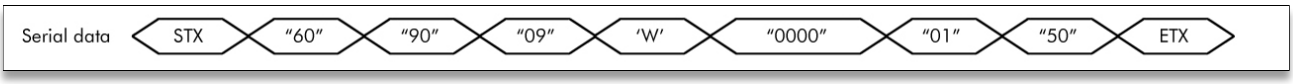


Figure 5: Write LNB enable in the tuner and demodulator settings

The command consists of the following parts:

- Start character 'STX'
- Two hexadecimal address characters ("60") (address doesn't care, see §4.2.1)
- Two hexadecimal category characters ("90")
- Two hexadecimal setting characters ("09")
- A write character 'W'
- Four hexadecimal index characters ("0000")
- Two hexadecimal data characters ("01")
- Two hexadecimal checksum characters ("50", See Table 1)
- Stop character 'ETX'

Table 1: Checksum computation

Characters	ASCII value
6	0x36
0	0x30
9	0x39
0	0x30
0	0x30
9	0x39
W	0x57
0	0x30
0	0x30
0	0x30
0	0x30
0	0x30
1	0x31
Sum:	0x2B0
Checksum:	0x50

Serial read command on USB interface

Figure 6 shows the read command of the device type number (category 1, setting 5). The returned data consists of 4 bytes (int32). All values are displayed as ASCII characters.

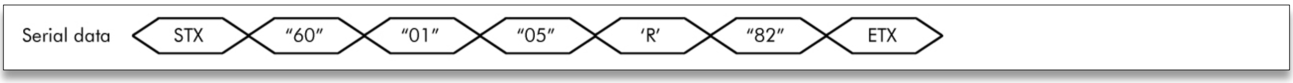


Figure 6: Read-command for the device type setting

The command consists of the following parts:

- Start character 'STX'
- Two hexadecimal address characters ("60")
- Two hexadecimal category characters ("01")
- Two hexadecimal setting characters ("05")
- A read character 'R'
- Two hexadecimal checksum characters ("82")
- End character 'ETX'

Figure 7 shows the two possible replies from the command in Figure 6. The replies are similar to the commands with the exception of the data-characters or the read character. On a successful command, the reply-data is set to the corresponding data 3237 (0x00000CA5). When the received command cannot be executed, the read character is set to the ASCII character 'E' and the data is removed. In both cases the checksum is updated.

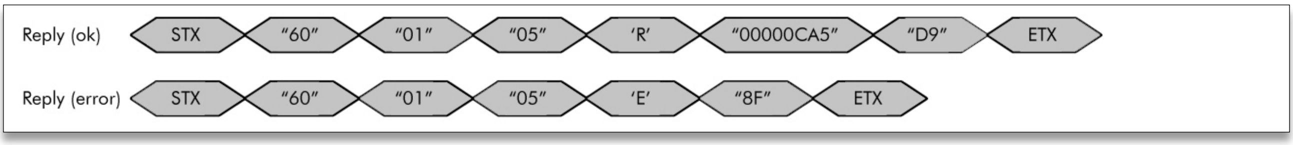


Figure 7: Reply after a device type read-command (successful and error)

I²C read command

Figure 8 shows the communication sequence used to issue a read frequency command (category 0x90, setting 3). The returned data consists of the frequency.

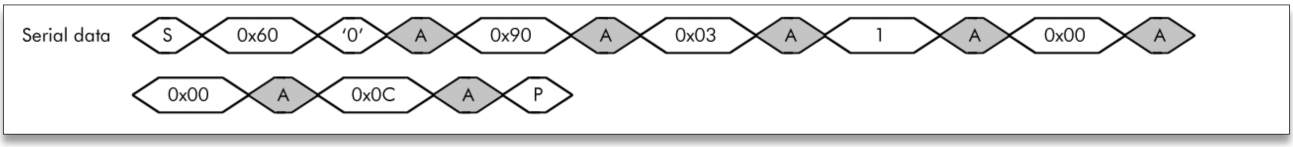


Figure 8: Send frequency read-command

The command consists of the following bytes:

- Address and I²C write-bit (0x60 and '0')
- Category byte (0x90)
- Setting byte (0x03)
- Read byte (0x01)
- Index bytes (0x00 and 0x00)
- Checksum (0x0C, see Table 2). The checksum is computed with the address and without the I²C write-bit.

Figure 9 is the reply-sequence that may be executed after the read-command of Figure 8. After addressing this device, the bytes from the command are repeated followed with the 4-byte frequency.

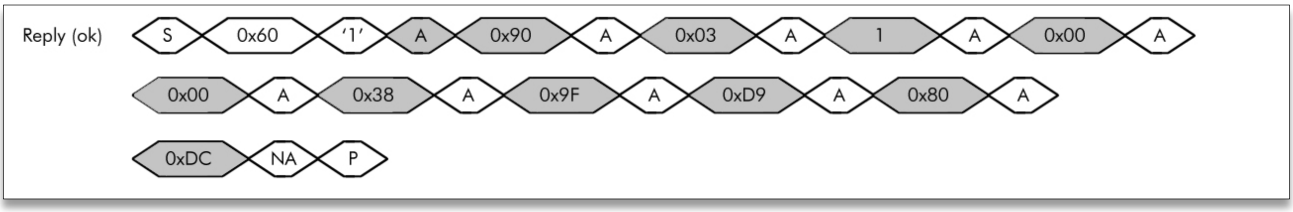


Figure 9: Read frequency reply

The reply consists of the following bytes:

- Address and I^{2}C read-bit (0x60 and '1')
- Category byte (0x90)
- Setting byte (0x03)
- Read byte (0x01)
- Four frequency bytes (decimal 950.000.000 or hexadecimal 38.9F.D9.80)
- Checksum (0xDC, see Table 2)

Table 2: Checksum computation

	Command	Reply
Address	0x60	0x60
Category	0x90	0x90
Setting	0x03	0x03
R/W	1	1
Index (byte 1)	0x00	0x00
Index (byte 2)	0x00	0x00
Data byte 3	-	0x38
Data byte 2	-	0x9F
Data byte 1	-	0xD9
Data byte 0	-	0x80
Sum:	0xF4	0x324
Checksum:	0x0C	0xDC