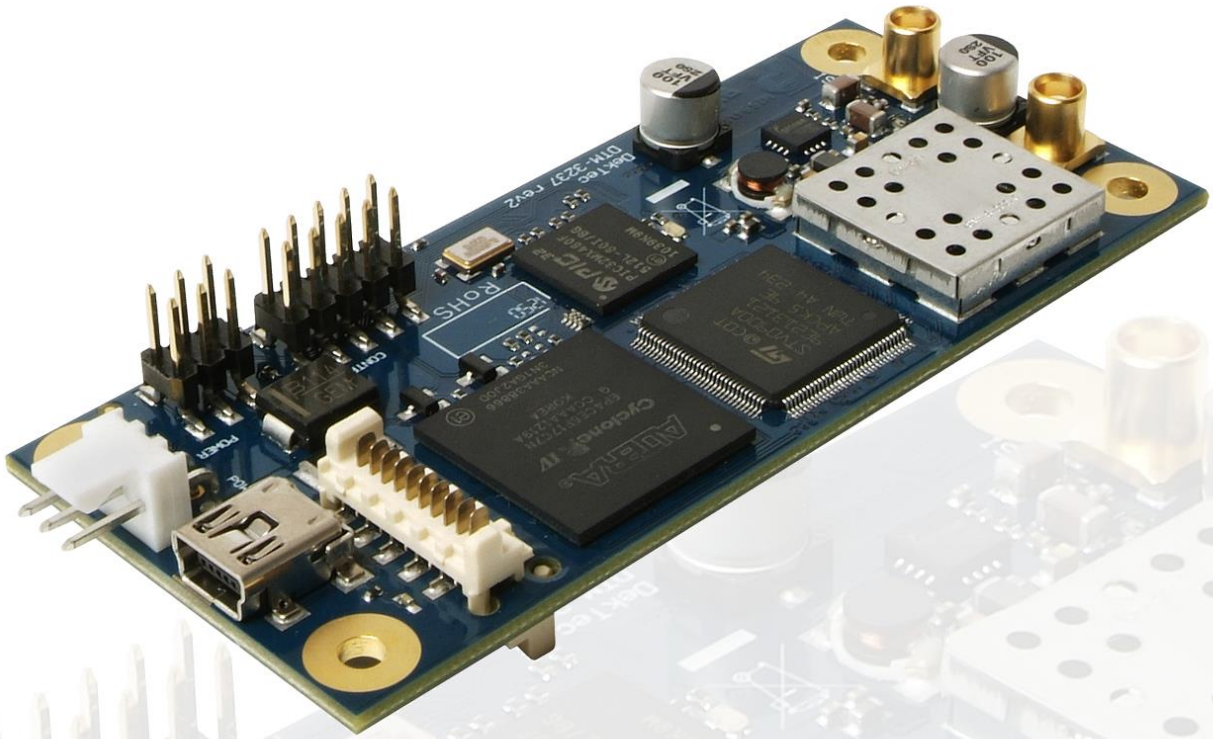
DTM-3237

OEM DVB-S2 Receiver with DVB-ASI Output



USER **MANUAL**

November 2013

Table of Contents

Revision History

Version	Date	Changes
0.1	2013.11.11	Added setting 0 to certain category's, which contains all settings in that category. Using this setting reduces the amount of transfers from/to the device up to 80% when reading all settings. Please refer to the section §4.5 "Device Configuration and Monitoring – Categories" for more information. Added setting option "DVB-S2 Multiple Input Stream" (6) to the Modulation Type setting (0x04) in category Receiver Status (0x91). Limited setting Link Margin (0x06) in category Receiver statistics (0x92) to DVB-S2 Single Input Streams. Changed interface name to "LVTTL serial control port" in order to avoid confusion. Please refer to chapter §4.2.1 "Command protocol on USB and LVTTL serial control port" for more information.
0.0	2013.06.03	Initial release to the field

1. Introduction

1.1 General description

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

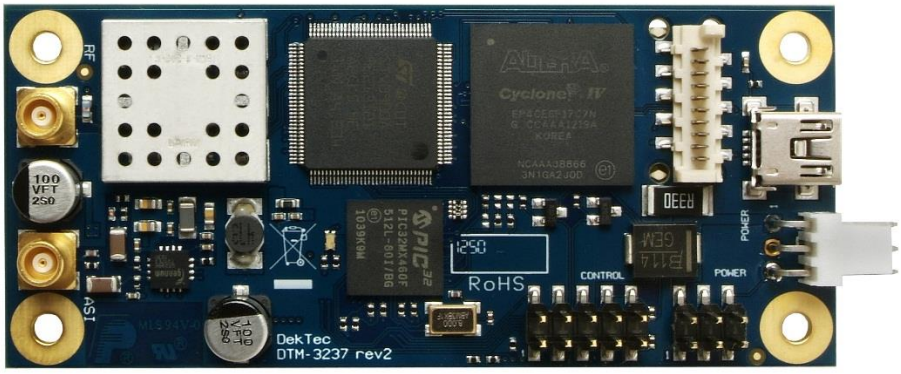


Figure 1. The PCB of the DTM-3237

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

1.2 DVB-S(2) input modes

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

1.3 Output modes

The DTM-3237 can output the DVB-S(2) stream in 2 modes: Transport Stream and L.3 Baseband frames. Please refer to *DTAPI Manual - Overview and Data Formats.pdf* for more detailed information about the L.3 Baseband frame data format. This document is part of DekTec's *Windows SDK* and can be downloaded from www.dektec.com.

1.4 Control

The unit can be managed and controlled through one of the available control interfaces: USB, I²C or LVTTL serial control port. Settings applied through a control interfaces are persisted in non-volatil memory if setting *Volatile storage* is '1'. Persisted settings are automatically reloaded after a powe cycle. The penalty for persisting settings is that it takes some time to store the setting value in no volatile memory, and a limit to the number of write cycles to non-volatile memory exist.

If setting *Volatile storage* is '0', applying settings is immediate, but new setting values will disappea when power is removed.

There are three ways to control the DTM-3237:

1. From a development PC using the USB control interface. The development PC runs a control tool: *Dtm3237Util* or *DtmCmd*. This way of controlling can be used for pre-configuring the DTM-3237, or for experimenting with the DTM-3237.

2. Using a controller on-board of the equipment that uses the DTM-3237 as co-processor board for satellite reception. In this case an I²C or LVTTL serial control port is a plausible choice for the control interface, but USB interface can also be used. The I²C address and LVTTL serial control port baud rate can be pre-configured through USB with *Dtm3237Util*. The factory-default I²C address is 0x60 and the default LVTTL serial control port baud rate is 9600.
3. Stand-alone mode. The DTM-3237 is pre-configured and no dynamic control is applied.

Two control tools are available:

1. *Dtm3237Util* – Windows GUI tool to view status and control settings of the DTM-3237. The tool can also be used to upload new firmware versions. *Dtm3237Util* is convenient for initial configuration of the DTM-3237 and for experimentation with the DTM-3237.
2. *DtmCmd* – Command-line tool to send commands to the DTM-3237. Multiple commands can be combined in a script to apply a group of settings in one go. *DtmCmd* is useful for studying the low-level commands available for the DTM-3237. It is also useful to apply a pre-defined group of setting values from a script.

1.5 DTM-3237 Protocol Handler

For developers that implement their own application controlling the DTM-3237, an open s implementation of a protocol handler for DTM-32XX devices is available. It can be downloaded fr `www.dektec.com` free of charge, and can be used royalty-free. It consists of two source fi `DtmHandler.c` and `DtmHandler.h`, which can be compiled and linked into your C or C++ application. Please refer to `DtmHandler.h` for information about how to integrate the protocol handler in your application.

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-3237 consists of two subsystems:

- A tuner, demodulator and stream processor, converting the modulated DVB-S(2) input signal to a Transport Stream or L.3 Baseband frames on DVB-ASI;
- A processor subsystem that handles all internal/external control (USB, I²C, LVTTL serial control).

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, channe 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 Servic News Gathering and other broadband satellite applications.

2. Getting Started

2.1 Introduction

This section provides a walkthrough for getting started with the DTM-3237. The description below assumes that you have a DTM-3237 development kit available (see Appendix B). The DTM-3237 i connected to a development PC with USB. The GUI control tool *Dtm3237Util* is used to apply setting and observe status.

2.2 Configuration: Receiving a DVB-S(2) stream

This setup will receive a DVB-S(2) stream and transmit the stream on the ASI interface.

2.2.1 Test set-up

For testing this configuration, a DVB-S(2) signal should be connected to the DTM-3237's RF input. observe the output of the DTM-3237, an ASI receiver is helpful[1].

This tutorial assumes that a DVB-S(2) stream with the following or equivalent parameters is appli the DTM-3237.

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-S(2) to ASI conversion

Use *Dtm3237Util* to configure the DTM-3237 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:

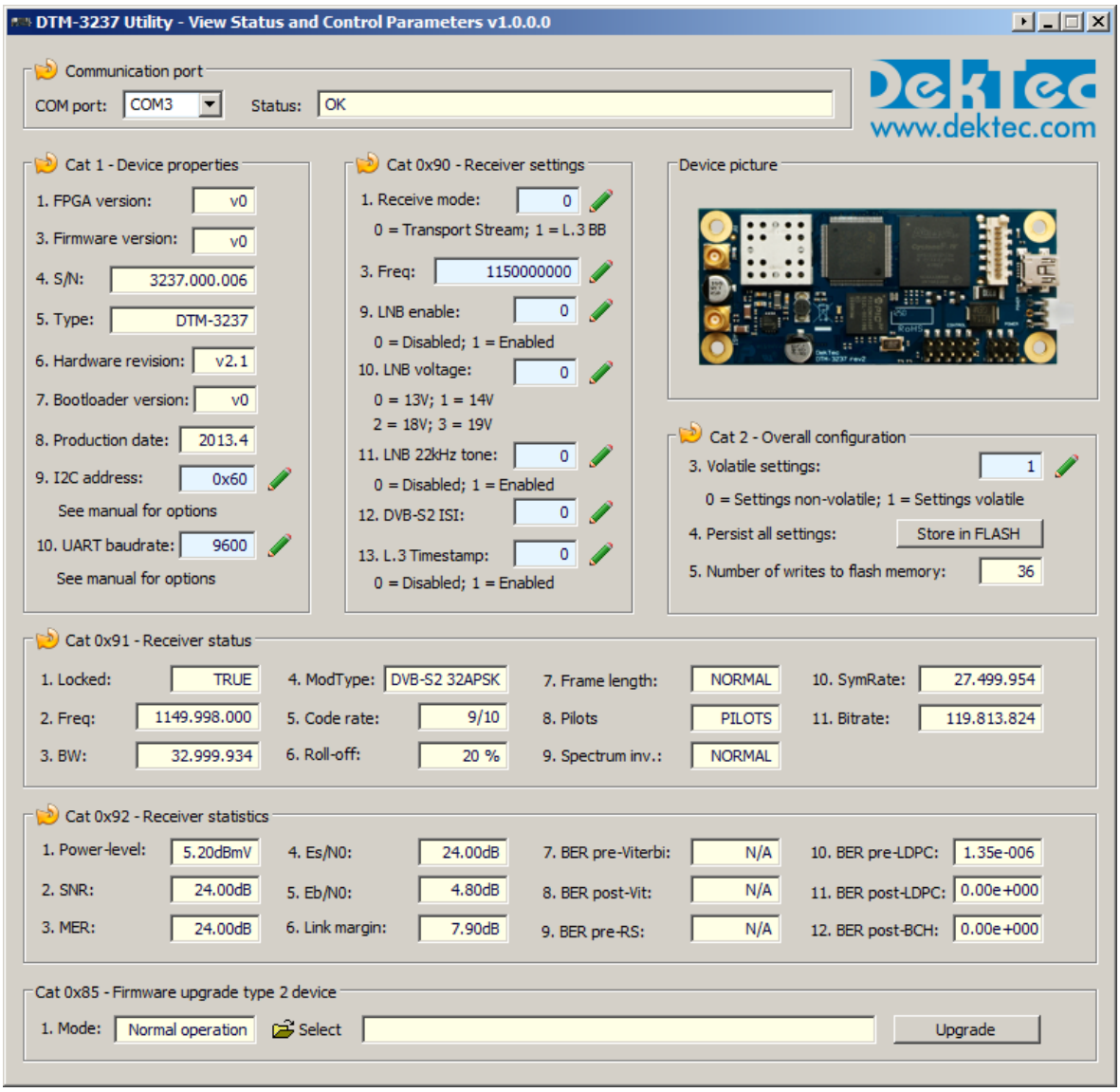
	Category	Setting	Setting name	Value
1	0x90	0x01	Receive mode	0 = Transport stream
2	0x90	0x03	Frequency (Hertz)	1150000000 = 1150MHz
3	0x90	0x09	LNB power enable	0 = Disabled
A refresh is required to see the receiver status and statistics.				

The DTM-3237 will now receive a DVB-S(2) stream and transmit this stream on the ASI port. The s LED will be steady green to indicate successful transmission of the transport stream on the ASI ou If no DVB-S(2) stream is received, the status LED will flash green on and off.

A screenshot of the *Dtm3237Util* after configuration is added on the next page.

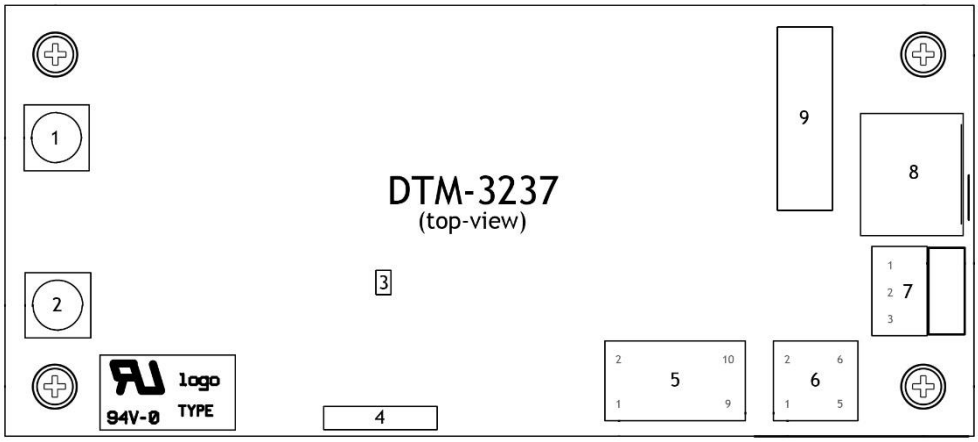
[1] If you do not have a suitable DVB-S(2) modulator and/or ASI receiver, this functionality can, for example, be realized a PC and a DekTec DTA-2107 and/or DTA-2144 I/O card in it. Please consult your local DekTec representative for more information.

Below you find a screenshot of the *Dtm3237Util* after all settings for the “Getting Started” configuration has been applied and the upper left refresh arrow has been pressed to refresh all th settings, receiver status and statistics.



3. Layout and Installation

3.1 Physical layout



#	Field	Connector type	Description
1	RF input	MCX 75Ω	DVB-S(2) input
2	ASI output	MCX 75Ω	DVB-ASI output
3	Status LED		DTM-3237 Status LED
4	Identifier		Type and revision number
5	Control	10-pin header 2.54mm pitch	LVTTL serial control port and I²C 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
9	Factory		Factory connector; Not used in normal operation

3.2 Mechanical dimensions

See Appendix A.

3.3 Order codes

Order Code	Picture	Description
DTM-3237		DTM-3237 – OEM DVB-S2 receiver with DVB-ASI output
DTM-3237-DEVKIT	The DTM-3237 development kit contains the following items: • DTM-3237 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 spac
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 hea
connecting the LVTTL serial control port control bus.

Pin	Function
1	LVTTL serial control port TX
3	LVTTL serial control port RX
5	NC
7	GND
9	I²C SDA

Pin	Function
2	LVTTL serial control port CTS
4	LVTTL serial control port RTS
6	NC
8	I²C SCL
10	I²C SCL

The pinning of this connector in the table above must be read from the DTM-3237's point of view
For example: Pin 1 TX is the DTM-3237 transmitter pin, which should be connected to the RX pin
the equipment which uses the DTM-3237.

An I²C controller can be connected to SDA and SCL on pin 11 and 12, with signal ground on pin 9.

3.4.5 Power connector

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

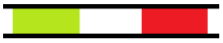
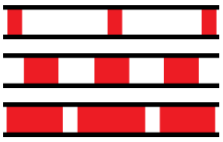
Connector 6 – Pin header 2.54 mm pitch			
Pin	Function	Pin	Function
1	+12V DC in	2	+12V DC in
3	Ground	4	Ground
4	Reset	6	Reset

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 100ms. The reset pin is con
with a resistor to the 12V DC input voltage. When connecting a driver to the reset pin of the DTM-
3237, make sure it is 12V tolerant. To trigger a reset, the voltage on the reset pin shall be 700mV
less.

3.4.6 Stream status LED

The status LED indicates the status of the DVB-S(2) receiver and the ASI output stream. The follo
colors are used for status indication:

	Continuous green	Valid DVB-S(2) signal detected and generating signal on DVB-ASI output
	Short green flashes	No DVB-S(2) signal detected and no output generated on DVB-ASI output
	Short red/green flashes	The DTM-3237 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-3237 can be configured and monitored using USB and LVTTL serial control port. The I²C address and LVTTL serial control port baud rate can only be configured through the USB interf e.g. with *Dtm3237Util*. It is not required to select between the interface of your choice (USB, I²C and LVTTL serial control port): The DTM-3237 will automatically use the interface on which it activity.

- The default I²C address is 0x60.
- The default LVTTL serial control port baud rate is 9600.

The USB interface is implemented as USB to serial emulator and therefor the USB and LVTTL seria control port protocol are the same. The baud rate for the USB serial emulator is 256000.

All control interfaces use the same command and response protocol that is described below.

4.2 Command protocol

Commands and responses are wrapped into a frame structure that contains address, category, se read/write, index and data (optional). The DTM-3237 accepts uppercase and lowercase character but will always respond in uppercase.

4.2.1 Command protocol on USB and LVTTL serial control port

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] The DTM-3237 only supports LVTTL serial control port and USB, so no address is necessary and therefor this value doesn’t care.
[4] The DTM-3237 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.

The DTM-3237's LVTTL serial control port is designed to interface directly with a micropro system. The serial control port uses the same protocol as RS-232, but the I/O voltages are entirel different. This LVTTL serial control interface is not compatible with a RS-232 port, e.g. a PC's COM port, due to its different voltage.

> ***Do not connect the LVTTL serial control port to a PC's COM-port***

Connecting the LVTTL serial control port of the DTM-3237 to a PC's COM port may cause **permanent damage to the DTM-3237**.

	RS-232	DTM-3237 LVTTL serial control port
Digital '0'	+3V ... +15V	0V
Digital '1'	-3V ... -15V	+3.3V

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

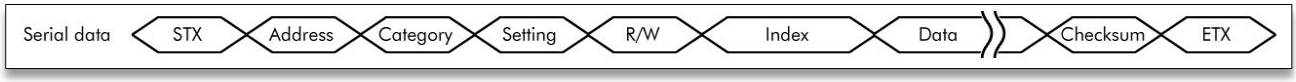


Figure 2. Command on an USB and LVTTL serial control port serial control interface

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

When an incorrect checksum or an invalid hex value is detected, the DTM-3237 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 the message.

Please refer to Appendix D for a communication example.

4.2.2 Command protocol 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

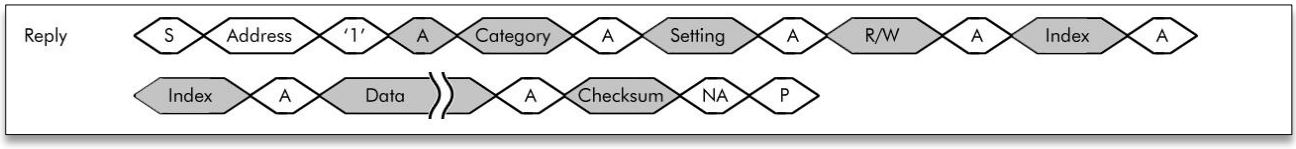


Figure 3 below shows the sequence to send a command to the DTM-3237. In the examples below, grey areas in the timing diagrams are sent by the DTM-3237, while white areas are sent by the I²C master. If the command is a read-command, the data may be omitted. Note that some settings require an index.

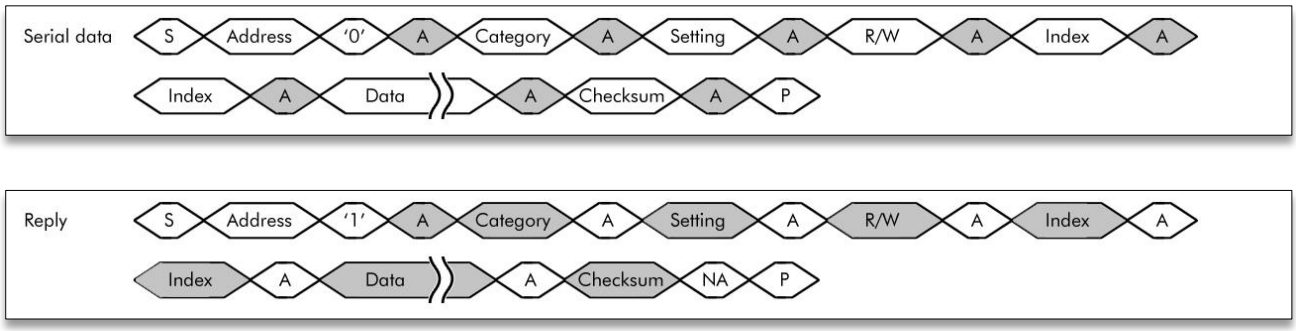


Figure 3. Command (upper sequence) and response (lower) sequence for I²C

[6] The DTM-3237 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-3237 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 o 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 will ignore the data.

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

S and P are the standard I^2C 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 monitore using the USB, I^2C or LVTTL serial control port interface. The ‘Access’ column indicates whether the variable can be changed or not, 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 immed result in an action on the device. In contract to these directly processed items, some items requi more processing time and therefor these actions are performed with a delay in a background thre Items which execution is delayed are marked with a “DE” (delayed execution) prefix in the acces column.

Items which execution is delayed can be monitored by the “Busy” item (category 0x01, setting 0 During the execution of these items, the “Busy” item is set and it is not possible to execute other marked items. In case the “Busy” item is set and a “DE” marked command is received, the devic 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

4.5.1 Data types

Manageable Items – Data types			
Type	Description	LVTTL serial control port	I^2C
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	29 bytes
0x01	FPGA version	Version number of the FPGA code on-board of the DTM-3237	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^2C address	7-bit I^2C address Limitations: - Bit 0 (I^2C R/W bit) can't be set - Address range 0x08 to 0x76 Changes to the I^2C address are automatically persisted in flash memory. Default: 0x60	R/Wusb	uint8
0x0A	LVTTL serial control port baud rate	Serial baud rate Valid values: 9600, 115200 and 256000 Changes to the baud rate are automatically persisted in flash memory. Default: 9600	R/Wusb	uint32
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-3237, 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 (please refer §4.6 - Firmware upgrade).

The new firmware has to be uploaded in "file parts". For communication through I²C 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	10 bytes
0x01	Receive mode	Receive mode 0* = Transport Stream packets 1 = L.3 baseband frames	R/W	uint8
0x03	Frequency	Frequency in Hertz Range: 950000000...2150000000	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
0x0D	L.3 timestamp insertion	32-bit timestamp insertion in L.3 baseband frames enabled/disabled 0* = Disabled 1 = Enabled	R/W	uint8

* Factory default

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

Manageable Items – Category 0x91 – Receiver status				
Nr	Variable	Description	Access	Type
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	RO	uint8
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	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 2 = Short 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
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 correc
For the DTM-3237 it is be computed from the MER using formulas. These formulas have
validated using DekTec's advanced demodulator simulation software (this software has been use
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	72 bytes
0x01	RF power-level	RF power-level for channel bandwidth in 0.1 dBmV units INT_MIN = Unknown or not locked	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
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

Manageable Items – Category 0x92 – Receiver statistics				
Nr	Variable	Description	Access	Type
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

When a statistic has the output type uint64, the received data can be converted to variable doub
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/fro
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

4.6 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 260 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-3237 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-3237 checks the status of the firmwa In case the firmware isn’t correctly uploaded, the DTM-3237 will stay in firmware upgrade mode. 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 firmwa upgrade mode or rebooting the DTM-3237, 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 *DtmCm* (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 po and USB interface.

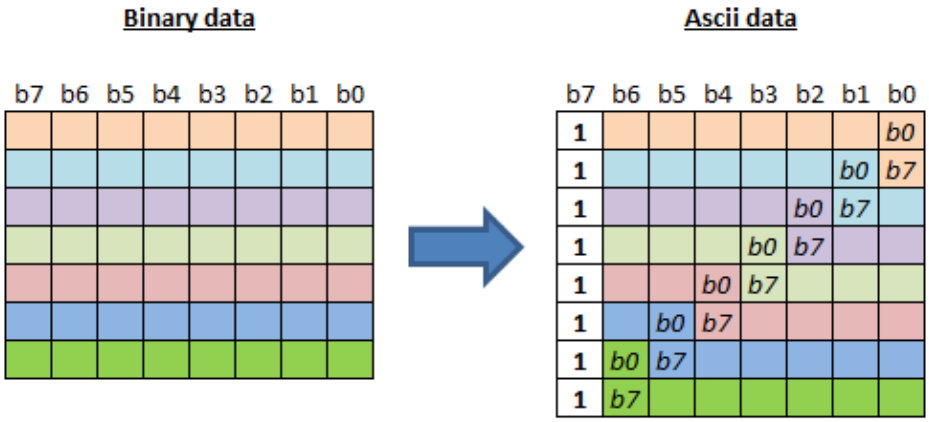


Figure 4. ASCII encoding for firmware upgrade data

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

4.6.3 Firmware upload - Example

In the example below the DTM-3237 is upgraded with new firmware over USB. The size of the firm file is 486400 bytes. Over USB, each packet can hold a maximum of 250 data bytes, so the numb 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 upg

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-3237 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	
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 ^{2}C, data bytes are packaged without conversion in the ^{2}C write message.

5. Specifications

5.1 RF input

	Min	Typ	Max	Unit / Remarks
Standard				
DVB-S		EN 301 210		
DVB-S2		EN 302 307		
RF input				
Connector type		MCX female		
Impedance		75		Ω
Return loss	>9			dB (950 .. 2150MHz)
Tuning range	950		2150	Hz
Sensitivity	-60		-30	dB
Baud Rate	2		45	MBd
DVB-S2 modes				
Constellation		QPSK, 8PSK, 16APSK, 32APSK		
Code rate		All supported by STV0900		
Coding mode		CCM, VCM, ACM		
Multiple transport streams		Supported in L.3 output mode		
Metrology				
RF level	-60		-30	dBm
RF level accuracy		5		dBm
MER			22	dB
MER accuracy		2		dB
Constellation		Not supported		
LNB interface				
LNB supply		13V/14V/18V/19V 500mA		
LNB short circuit protection		600 ±10%		mA
22kHz tone		On/Off		

5.2 DVB-ASI input

	Min	Typ	Max	Unit / Remarks
Standard				
DVB-ASI		EN50083-9		
Ports				
Connector		75-Ω MCX		
Return loss	15	27		dB
Error-free cable length	100			m
ASI bitrate		Matching DVB-S(2) bitrate		
Packet size		188 bytes		In TS mode
Receive modes		TS, L.3 baseband frames		

5.3 Serial control port

	Min	Typ	Max	Unit / Remark
Interface port				
Connector		10-way pin header, 2.54 mm pitch		
Signals		Tx/Rx/RTS/CTS		
Serial format				
Interface standard		LVTTL serial control port		
Format		8 bit, one stop bit, no parity		
Handshaking		hardware flow control		
Speed		9600, 115200 or 256000 9600 by default		Baud, configurable through command protocol

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

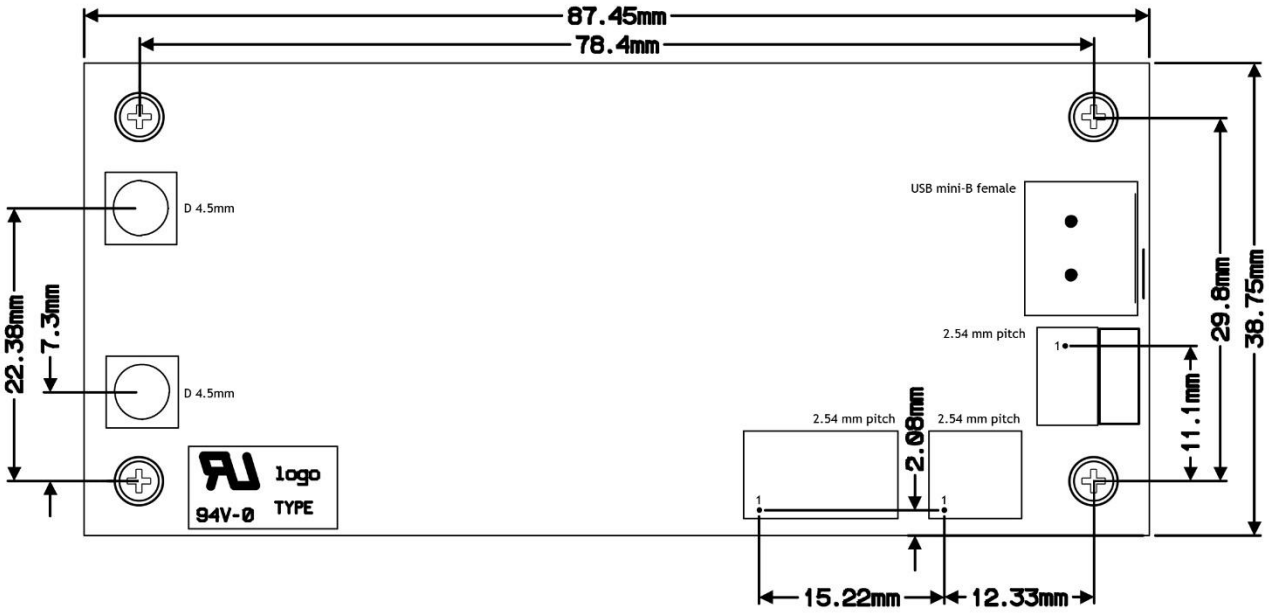
5.5 USB control port

	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			5	W
Power Consumption including LNB			20.5	W (LNB @ 500mA/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-3237, care should be taken not to damage components that ar close to the mounting holes, both on the top and bottom side of the board.

Appendix B. DTM-3237 Development Kit

B.1 DTM-3237 development kit – Contents

The DTM-3237 development kit contains the following items:

- DTM-3237 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-3237 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-3237-DEVKIT.

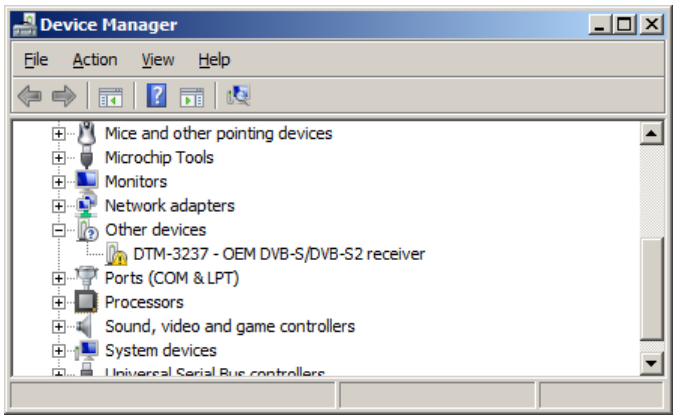
B.2 Using the DTM-3237 development kit

B.2.1 Hardware installation

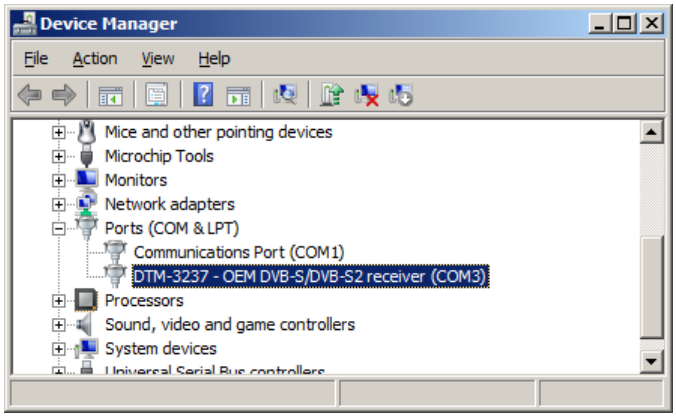
The DTM-3237 board has to be connected with the USB cable to a development PC. Connect the DTM-3237 to the power-supply using the power connector. The DTM-3237 will boot which will tak a few seconds. During this time the LEDs on the DTM-3237 are flashing in a start-up pattern. Wai until the DTM-3237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

B.2.3 USB driver

The DTM-3237 uses a default USB serial driver (usbser), which needs to be installed with `Dtm3237.inf` driver file. Connect the DTM-3237 to a USB port on the PC with the USB cal included in the development kit. After a while the USB connection to the DTM-3237 board will bec visible as a DTM-3237 - DVB-S(2) Demodulator device.



At that point the driver can be updated by using the "Update Driver Software" window (right mou button) and selecting the `Dtm3237.inf` file from the development kit. After installing the driver, the DTM-3237 is ready to be used with *Dtm3237Util* or any other software which uses the COM-port interface to communicate with the DTM-3237.

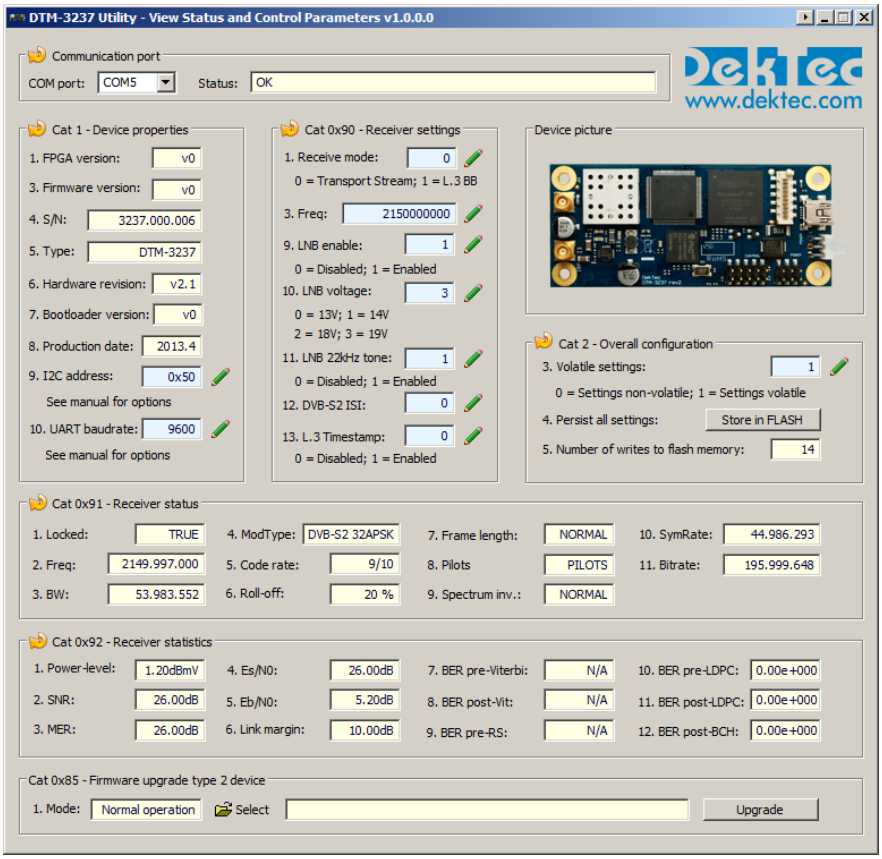


In this case the DTM-3237 is configured as COM port number 3.

B.2.4 Debugger

Dtm3237Util is a GUI tool to view status, control settings and upload firmware to the DTM-3237. Th utility can be found on the DekTec USB flash drive. It can also be downloaded from the website. *Dtm3237Util* 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 wh the DTM-3237 is connected. When a valid serial port is selected, all registers are read fr DTM-3237 and shown in the GUI. Blue fields can be edited. These fields are written to the DTM-32 when the pen symbol to the right of the edit fields is clicked. Yellow fields are read only; they are when the refresh arrow is clicked.



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 argument is executed on the DTM-3237. See "`DtmCmd -?`" for help on the available commands.

You can specify the interface type (serial, I²C), 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-3237 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-3237

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 I²C 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-3237 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) 2013 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-3237

To demonstrate the writing of a setting, we write 2150000000 to the setting *Frequency* (setting 3) 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-3237's firmware

To demonstrate the firmware upgrade process, we pass the filename of the firmware file to *DtmC*
The USB parameter (-usb) must be passed in case the USB interface is used. Please refer to §4.6. Firmware upgrade - Phases for more information about upgrading the firmware using the 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-3237, while white areas are sent by the master. The address of the DTM-3237 in these examples is *0x60*.

Serial write command on LVTTL serial control port interface or USB interface

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

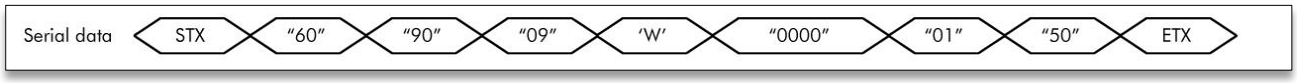


Figure 5: Write LNB enable in the tuner and demodulator settings

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 LVTTL serial control port interface or USB interface

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

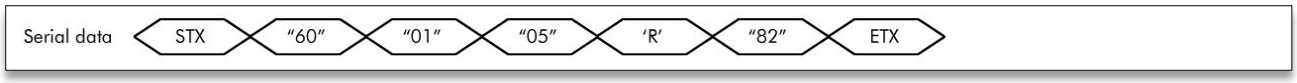


Figure 6: Read-command for the device type setting

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 t commands with the exception of the data-characters or the read character. On a successful comm 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. both cases the checksum is updated.

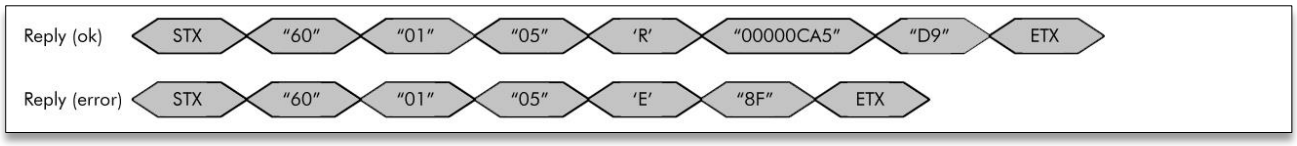


Figure 7: Reply after a device type read-command (successful and error)

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.

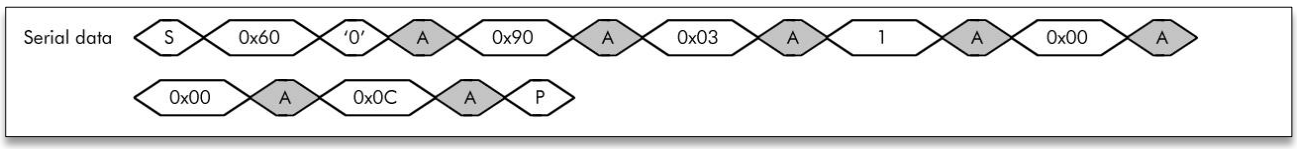


Figure 8: Send frequency read-command

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 I²C without the I write-bit.

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

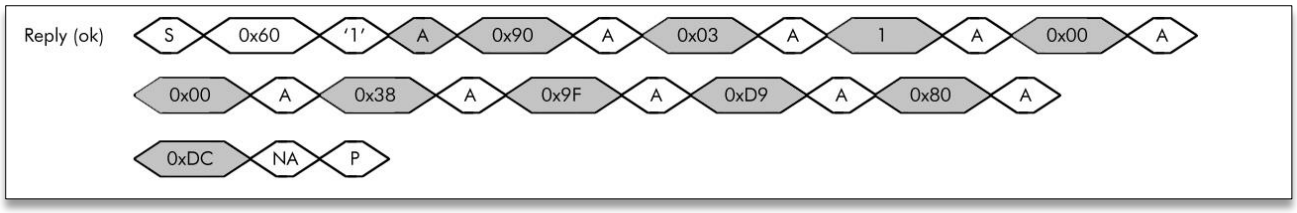


Figure 9: Read frequency reply

The reply consists of the following bytes:

- Address and I²C 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