

From PT-MT/ELF,EDR	Our Reference Rathg./Amann/Braun/Wiela	Tel
-----------------------	---	-----

Issue	1.2.9
Topic	<b>MT-P</b>
Description	Documentation of the Bosch MT communication modes and MT command set

# Bosch MT connectivity protocol

## Table of contents

Bosch MT connectivity protocol.....	1
Table of contents .....	1
1 Abbreviations and naming conventions .....	2
2 Hardware setup and supported interfaces.....	2
2.1 RS232 interface settings.....	2
2.2 Master/slave mode and timing constraints .....	2
3 MT connectivity protocol layer for RS232, BlueTooth and USB.....	4
3.1 MT connectivity request frame format .....	4
3.1.1 Request frame format LONG (Standard) .....	4
3.1.2 Request frame format EXTENDED .....	4
3.1.3 Request frame format SHORT .....	4
3.1.4 Request Mode options and checksum calculation.....	5
3.2 MT connectivity response frame format and error indication .....	6
3.2.1 Response frame format LONG (Standard) .....	6
3.2.2 Response frame format EXTENDED .....	6
3.2.3 Response frame format SHORT .....	6
3.2.4 Status byte definitions .....	7
4 MT Command Set .....	8
4.1 Overview Table .....	8
4.2 Additional command description.....	13
4.2.2 Command ZZ Get/Set/Do example data.....	13
4.3 Document history .....	14

## 1 Abbreviations and naming conventions

MT	Measuring Technology
LRF	Laser Range Finder
Frame type	Request or response frame
Frame format	SHORT, LONG or EXTENDED frame

## 2 Hardware setup and supported interfaces

The Bosch MT connectivity protocol is used for all Bosch Measuring Technology devices with any communication interface:

- USB
- BlueTooth
- RS232 Rx/Tx (2-wire) with RS232 transceiver
- RS232 single wire (RtxPin) with single wire adapter

### 2.1 RS232 interface settings

Configure the serial interface of the PC as follows:

Basic baudrate:	9600 baud
Adjustable baudrates:	9600 / 19200 / 38400 / 57600 / 115200 baud
Data-bits:	8
Parity:	None
Start-bit:	1
Stop-bit:	1

### 2.2 Master/slave mode and timing constraints

The RS232/BlueTooth/USB application protocol is a bidirectional half-duplex master-slave based communication which supports optional command queuing for a point-to-point connection.

It is typically the PC/Smartphone which acts as master and controls the measuring device (slave) in remote mode. The slave generally sends a response telling the delivery status, optionally including data bytes.

Only for special features the measuring device is the master, e.g. sending a measured value by pressing a button on the keypad. To make sure the measured value was correctly delivered to the PC/Smartphone a response frame could be processed by the measuring device and an error may be shown on the display if desired.

During communication idle all parties are in slave mode. Once a request should be sent the requestor switches to master mode, sends the request. The requestor remains in master mode

From  
PT-MT/ELF,EDR

Our Reference  
Rathg./Amann/Braun/Wiela

Tel

Leinfelden  
2018-09-03

until the response is completely received or a receive timeout has occurred if a response is expected.

All frame bytes should generally be sent consecutively without large time-gaps in between. For optional implementations of a receive timeout, the slave responds with a timeout status and the request may be repeated. The receive timeout value is defined separately for each interface, e.g. for LRF-RS232 the value is 60 ms.

After the slave has received a valid request, the corresponding routine is executed. Some requests take a few seconds to complete, e.g. a LRF single distance measurement under worse conditions.

### 3 MT connectivity protocol layer for RS232, BlueTooth and USB

#### 3.1 MT connectivity request frame format

Frame format definitions:

LONG is the general purpose frame format which has to be available for all devices.

SHORT is desired for fast communication or high communication load without data.

EXTENDED is desired for command queuing and/or very high data load and for high reliability purposes (CRC-32).

Multiple-byte values within the data bytes (payload) are coded in LSB first order.

##### 3.1.1 Request frame format LONG (Standard)

Description	Mode	Request-command	Data length N	Data (if N > 0)	Checksum CRC-8
Size [byte]	1	1	1	N	1
Offset [byte]	0	1	2	3	3+N

LONG/EXTENDED usage hint:

To detect a 2-bit error the Data Length of a LONG frame (CRC-8) should have less than 30 data bytes; otherwise the EXTENDED frame format is the better choice in terms of data reliability. CRC32 was tested successfully with 2-bit errors up to 8000 data bytes.

##### 3.1.2 Request frame format EXTENDED

Description	Mode	Request-command	Data length N	Data (if N > 0)	Checksum CRC-32
Size [byte]	1	1	2	N	4
Offset [byte]	0	1	2	4	4+N

Data Length N: Byte 2 = LSB (Bit 7..0), Byte 3 = MSB (Bit 15..8)

Checksum: Byte 4+N = LSB (Bit 7..0), Byte 7+N = MSB (Bit 31..24)

##### 3.1.3 Request frame format SHORT

Description	Mode	Request-command	Checksum CRC-8
Size [byte]	1	1	1
Offset [byte]	0	1	2

**3.1.4 Request Mode options and checksum calculation**

Request Mode Byte	Format option definition
Bit [7..6]	Frame type identifier: 00=response; 11=request; others=invalid
Bit [5..4]	<reserved>
Bit [3..2]	Request frame format: 00=LONG 01=SHORT 10=EXTENDED 11=reserved
Bit [1..0]	Response frame format: 00=LONG 01=SHORT 10=EXTENDED 11=reserved

**Allowed/recommended frame format combinations:**

		Request frame format		
		EXT	LONG	SHORT
Response frame format	EXT	X	X	-
	LONG	$(X)^{*2}$	$X^{*0}$	X
	SHORT	$(X)^{*1,2}$	$X^{*1}$	$X^{*1}$

<sup>\*0</sup> LONG frame format is the default frame format which is implemented in every device type.

<sup>\*1</sup> If a SHORT response is requested and data are required to send the status 'Mode invalid' will be responded.

<sup>\*2</sup> While command queuing is desired exclusively EXTENDED response frame format is necessary.

Commands No 0...63 (0x00...0x3F) are reserved for the "MT command set", e. G. 'Get Device Name', 'Get SerialNumber', ...

**CRC8** polynomial:  $x^8 + x^6 + x^3 + x^2 + 1 = 0xA6$  [CRCembed]

**CRC16** polynomial:  $(x^3 + x^2 + 1)(x^6 + x^5 + x^2 + x^1 + 1)(x^7 + x^3 + 1) = 0xBAAD$  [CRCembed]

**CRC32** polynomial: 0x04C11DB7

All checksum formats are calculated from all message bytes from Mode to Data N respectively Command.

CRC8 initial value: 0xAA

CRC16 initial value: 0xAAAA

CRC32 initial value: 0xAAAAAAAA

[CRCembed] [http://www.ece.cmu.edu/~koopman/roses/dsn04/koopman04\\_crc\\_poly\\_embedded.pdf](http://www.ece.cmu.edu/~koopman/roses/dsn04/koopman04_crc_poly_embedded.pdf)

### 3.2 MT connectivity response frame format and error indication

The response frame format is given in the request Mode.

#### 3.2.1 Response frame format LONG (Standard)

Description	Status	Data length N	Data (if N > 0)	Checksum CRC-8
Size [byte]	1	1	N	1
Offset [byte]	0	1	2	2+N

#### 3.2.2 Response frame format EXTENDED

Description	Status	Request- command	Data length N	Data (if N > 0)	Checksum CRC-32
Size [byte]	1	1	2	N	4
Offset [byte]	0	1	2	4	4+N

#### 3.2.3 Response frame format SHORT

If response with data is required the status "Mode invalid" is returned.

Description	Status	Checksum CRC-8
Size [byte]	1	1
Offset [byte]	0	1

### 3.2.4 Status byte definitions

	Status Value	Status definition
Device Status	Bit [7..6]	Frame type identifier: 00=response; 11=request; others=invalid
	Bit [5] (0x20)	Hand raised (e.g. used by slave as request to talk or to switch roles)
	Bit [4] (0x10)	Device not ready (e.g. sensor not calibrated)
	Bit [3] (0x08)	Hardware error
Comm Status	Bit [2..0] = 0x00	Success
	Bit [2..0] = 0x01	Communication timeout *LONG
	Bit [2..0] = 0x02	Mode not supported in this device / Mode invalid / Frame overflow*LONG
	Bit [2..0] = 0x03	Checksum error *LONG
	Bit [2..0] = 0x04	Command unknown
	Bit [2..0] = 0x05	Access level or password for access level not valid
	Bit [2..0] = 0x06	Parameter or data bytes not valid (e.g. memory address)
	Bit [2..0] = 0x07	<reserved>

The **Comm Status field** is to be interpreted as a value based number. Therefore only one single status can be delivered, where the lower priority of the status comes with the higher number, e.g. a checksum error makes sense only if not communication timeout occurred.

In the **Device Status field** each bit represents a single status while multiple status can be set simultaneously.

\*LONG In case of framing error the response frame format is always LONG!

From PT-MT/ELF,EDR	Our Reference Rathg./Amann/Braun/Wiela	Tel
-----------------------	---	-----

## 4 MT Command Set

MT commands are generally supported by all MT devices. Specific commands may not be implemented if not necessary for a device shown in the table below in the device column (also indicated by a response status “Command unknown”).

All string based data come with EndOfString = EOS = '\0'.

### 4.1 Overview Table

Commands available in projects as per table: “x” = available; “!” = to be implemented (not working yet); “-“ not available

For SPAD-Platform: X: all devices, A: PLR30C/50C, A1: PLR30C, A2: PLR50C

Command	Access level	Description	Request data bytes (LSB first)	Response data bytes (LSB first)	GLM80C	Surya (PTD1)	Bootloader	SFinX	AXIOM	SPAD	Miraculix-
		<b>CONFIGURE COMMUNICATION (0..4)</b>									
0	0	Get communication info: Current Program mode: EN_MT_PROGRAM_MODE_BOOTLOADER = 0, EN_MT_PROGRAM_MODE_FLASHLOADER = 1, EN_MT_PROGRAM_MODE_APPLICATION = 2 Frame modes: Bit [1..0] supported frame modes <pre>           /// Supported protocol modes      0b00000000           ///  /// Extended frame mode available -----+            /// Short frame mode available -----+           </pre>	None	<b>8 bytes:</b> 1 byte: Current Program mode 1 byte: Supp. Frame Modes 1 byte: Supp. Baud rates 1 byte: Supp. Comm modes 2 bytes: Max Payload size Rx 2 bytes: Max Payload size Tx	X	X	X	X	X	X	-




 From  
 PT-MT/ELF,EDR

 Our Reference  
 Rathg./Amann/Braun/Wiela

Tel

		<pre>       /// Long must always be available!       Baud rates:       /// Supported baud rates          0b00000000       ///   /// Baudrate 115200 available -----+          /// Baudrate  57600 available -----+         /// Baudrate  38400 available -----+         /// Baudrate  19200 available -----+        /// Baudrate   9600 available -----+       Communication modes:       EN_MT_COMMUNICATION_MODE_HALF_DUPLEX = 0,       EN_MT_COMMUNICATION_MODE_FULL_DUPLEX = 1           </pre>									
4	0	Get version of Mt commands (Mt global and project specific)	None	<b>6 bytes:</b> 3 bytes: Mt Cmd Version 3 bytes: Project Cmd Version	X	X	X	X	X	X	-

From  
 PT-MT/ELF,EDR

 Our Reference  
 Rathg./Amann/Braun/Wiela

Tel

COMMON DEVICE ACCESS (5..9)											
5	0	Get Device name (if desired incl. SW version as ASCII)	None	<b>19 bytes:</b> String of 18 bytes + EOS	X	X		X	X	X	-
6	0	Get Device info (see also Cmd 200, 201, 202): DateCode (month based 3 byte ASCII + EOS), SerialNo (32 bit value), SW Revision (16 bit value), SW Version, HW Version, Bosch part number/TTNr (12 byte ASCII + EOS); SW and HW version are 3 byte in decimal format, where byte1='main', byte2='sub', byte3='bug', Hint: 0xFF.FF.FF is invalid version. HW Version: main = main PCB version, sub = main PCB variant (if available)	None	<b>29 bytes:</b> 4 bytes: DateCode 4 bytes: SerialNo 2 bytes: SW Revision 3 bytes: SW Version 3 bytes: HW Version 13 bytes: Bosch TTNr	X	X		X	X	X	-
15	0	Get RTC (RealTimeClock) Timestamp	None	<b>4 bytes:</b> 32 bit value in seconds since 1970-01-01 00:00	X			-	!	-	X
16	0	Set RTC (RealTimeClock) Timestamp	<b>4 bytes:</b> 32 bit value in seconds since 1970-01-01 00:00	None	X			-	!	-	X
61	0	SetActivePeer	3 byte: enumActivePeer CommActivityStatus ChannelNumber	None							X
TRANSMISSION CHECKING											

From  
PT-MT/ELF,EDROur Reference  
Rathg./Amann/Braun/Wiela

Tel

62	0	CMD_DO_ECHO For testing of transmission quality. Request and response must be of the same type!	<payload_max>	<payload_max>	-			X	X	-	X
63	0	CMD_DO_PING	None	None	-	X	X	X	X	-	X
		<b>Additional block of MT global command (200...234) – do not use in other projects!!</b>	<i>Ensure max BLE payload of 20 bytes (17 MT-protocol payload) is considered where required!</i>								
200											
...											
232	0	SetCommTunnel Requesting e.g. CPU to establish a communication tunnel from e.g. SingleWire to Miraculix, using parameter EN_BUS_NODE_COMM_MODULE <i>Ensure same baudrate for both channels before executing this command</i>	1 byte: enumActivePeer ( = target channel)	None							
233	3	DoHardResetMiraculix	None	None							
234	3	SetAndExecuteReStartupMiraculix	None	None	-	-	-	-	-	-	-
		<b>Miraculix-MTP specific commands (235...254) – do not use in other projects!!</b>	<i>Ensure max BLE payload of 20 bytes (17 MT-protocol payload) is considered where required!</i>								
		<b>For detail see OneNote “Communication Module”(administrated by Rene Schrottenholzer) and MtGlobalCmd.h</b>									
-	-	SetActiveChannel → use SetActivePeer cmd 61									X
235	0	<reserved>									X
236	0	<reserved>									
237	0	SetTmTDeviceName									

## Power Tools



Leinfelden  
2018-09-03

From  
PT-MT/ELF,EDR

Our Reference  
Rathg./Amann/Braun/Wiela

Tel

238	0	SetClockCalibrations									
239	0	GetClockCalibrations									
240	0	DoSerialUpdate									
241	0	GetVersionInfo									X
242	0	GetMirxBoschPartNumber									X
243	0	GetBackupBatteryStatus									X
244	0	SetMainBatteryStatus									X
245	0	GetCommModuleStatus									X
246	0	SetCommModuleStatus ("fake event")		frame n.a.							X
247	0	GetLocalMAC									X
248	0	GetRemoteMAC									X
249	0	SetSimpleModeAdvertisingName									X
250	0	SetBLEpause									X
251	0	SetDisableTmTWhileAsleep									X
252	0	SetToolIdentifier									X
253	0	SetBackupBatteryMode									X
254	0	SetToolUniqueIdentifier									X
		<b>Currently unused commands, to be kept for later usage</b>									

From  
PT-MT/ELF,EDR

Our Reference  
Rathg./Amann/Braun/Wiela

Tel

Leinfelden  
2018-09-03

## ***4.2 Additional command description***

### **4.2.2 Command ZZ Get/Set/Do example data**

Leinfelden

2018-09-03

From

PT-MT/ELF,EDR

Our Reference

Rathg./Amann/Braun/Wiela

Tel

### 4.3 Document history

Version	Date	Author	Changes
1.2.1	2014-03-17	Braun	Corrected / simplified frame format tables Updated command table according to latest MtGlobalCmd.h Updated command availability for GLM80C and SPAD platforms File conversion to docx
1.2.2	2014-04-15	Giesen	Added cmd 60 for touchpanel
1.2.3	2014-04-29	Haeberle/ Rathgeber	Updated parameters for flashloader commands 58+59 Reservation for future MT commands 200..255
a	2014-04-19	Rathgeber	Command 60 (get touchkey raw value): removed request data byte
b	2016-04-21	Rathgeber	Command 6 split in smaller packages in commands 200, 201, 202
c	2017-09-08	Amann	Changed Command 1 (SET_BAUDRATE) access level form 1 to 0
d	2017-10-06	Wieland	Added DoHardResetMiraculix
1.2.6	2018-05-18	Wieland	Added
1.2.7	2018-05-22	Wieland	Changed access level for CMD_GET_EEPROM_TRACE_DATA + CMD_GET_EEPROM_TRACE_DATA_INFO to EN_PARAM_ACCESS_LEVEL_0
1.2.8	2018-06-19	Rathgeber	Added command SetCommTunnel

PT-MT/ENG-Le