XM-B Technical Documentation

Document XM0101P, Revision H, 15 Oct 2010

Xsens Technologies B.V.
phone +31 88 97367 00
fax +31 88 97367 01
email info@xsens.com
internet **www.xsens.com**

xsens

Revisions

Revision	Date	By	Changes
1.0	November 20, 2002	SSM	First version
1.2	April 30, 2003	SSM	Added Bluetooth documentation
1.21	July 22, 2003	SSM	Minor changes including maximum SF of 512Hz
1.3	May 26, 2004	SSM	Update for XM-B version / included MT9 MTS messages
1.31	June 22, 2004	SSM	Minor updates
1.32	Jan 11, 2005	RGI	XM <> host serial cable
1.4	July 11, 2005	SSM	Changed description InitBus & SetBID regarding Xbus State
A	December 23, 2005	PRI	XM-B-XB2 changes included
B	March 2, 2006	PRI	New XM-B-XB2 SyncMode options included
C	February 21, 2007	PRI	BusPwrOff message replaced with Req/SetBusPwr message Batlvl data length added Updated address information of Xsens
D	April 1, 2008	PRI	Updated for MT SDK 3.0.1
E	August 4, 2008	RGI	Updated cables, wire color codes
F	September 30, 2008	RGI	Changed definition name from Ain to SYNC in section 5.3
G	May 27, 2009	MHA	New corporate design
H	October 15, 2010	MHA	Minor editorial changes

Table of Contents

1 Introduction

This document is a supplement to the Xbus Master User Manual. It discusses how the host can manually operate the Xbus Master, without using software supplied by Xsens Technologies B.V., and presents more detailed technical information and specifications about the Xbus Master. This document describes all basic messages. If your application requires more advanced functionality, please, contact Xsens as your needs may already be supported in more advanced messages.

2 Communication

2.1 Host interface

The Xbus Master can either connect with the host using Bluetooth or using the supplied XM USB-RS232 Cable. See next two paragraphs.

2.1.1 Serial link

When using the serial link between the Xbus Master and a computer or laptop the XM USB-RS232 Cable supplied with Xbus Master must be used, or you must use a special custom made RS-232 cable. This is because the computer, laptop and the Xbus Master are configured as data terminal equipment or DTE. This cable is a null-modem that crosses the RX & TX signals as well as the RTS & CTS signals, see chapter 5.3. The RTS & CTS signals are used for triggering & sample moment notification (see chapter 4.5). If another cable is used, please ensure that the RX, TX, GND, RTS, CTS pins are wired and if necessary crossed.

If the Xbus Master is to be connected to a PDA, which is mostly configured as DCE, use a 1:1 serial cable and not the supplied serial cable. Contact PDA manufacturer for an appropriate cable.

The following table shows the properties of the host interface.

Property	
Communication type	Data Terminal Equipment (DTE)
Baudrate	115200 bps (default) configurable (9k6-460k8 bps)
Data bits	8
Stop bit	1
Parity	None
Control lines	RTS, CTS
Flow control	None

2.1.2 Bluetooth link

The Xbus Master is equipped with Bluetooth capabilities so the data can also be transmitted wirelessly. The other end should have either the Xsens wireless receiver (WR-A) or a Bluetooth v1.1 or compatible transceiver supporting the Serial Port Profile. By default, the Xbus Master will try to connect to the last connected Bluetooth device if it is turned on (when set to Bluetooth mode). The Xbus Master can be learned to connect to another Bluetooth device, see XM User Manual.
It is also possible to disable Bluetooth communications. See **DisableBluetooth** message in chapter 3.3 for more information.

2.2 Packet format

The host communicates with the Xbus Master using packets. A packet contains of variable number of bytes with a fixed format. This packet is compatible with the MT communication protocol and has the following format:

Field	Field width	Description
Preamble	1 byte	Indicator of start of packet
BID	1 byte	Bus identifier - used to identify source or destination
MID	1 byte	Message identifier
LEN	1 byte	Value equals number of bytes in DATA field Maximum value is 254, 255 is reserved
DATA	0 – 254 bytes	Data bytes
Checksum	1 byte	Packet checksum

2.2.1 Preamble

Every packet starts with the preamble. This field always contains the value 250 (=0xFA).

2.2.2 Bus Identifier

This field identifies the sender or addressee of the packet. See table.

BID value		Description
0	0x00	Bus broadcast
1-254	0x01-0xFE	Identifies the Motion Tracker
255	0xFF	Xbus Master identifier

Packets with 0xFF as BID sent by the host will be processed by the Xbus Master. Packets with other values than 0xFF are not processed by the Xbus Master and directly copied to the bus.

Packets sent to the host are either from the Xbus Master itself or from one of the Motion Trackers connected to the Xbus Master. The BID value will be 0xFF if the packet is from the Xbus Master or in the range 0x01-0xFE if the packets is from one of the Motion Trackers. Assignment of the BID values to the Motion Trackers is explained in chapter 4.3.

2.2.3 Message Identifier

This packet field identifies the message type. For a complete listing of the messages see chapter 5.4. Normally, a MID of for example value 18 (0x12) will be replied with a message with a MID value of 18+1 = 19 (0x13). In some cases an error message is replied (MID = 66 (0x42)). This occurs in case the previous message has invalid parameters, is not valid, or could not be executed. These error messages contain an error code in their data field. See for more information the Error message in chapter 3.5.

2.2.4 Length

Specifies the number of data bytes in the DATA field. Value 255 (=0xFF) is reserved for message with data length exceeding 254 bytes. If zero, no data field exists. In case this field is equal to 255 (=0xFF), the following two bytes will give the new data length. For example the message FA FF 32 FF 01 05 xx xx xx .. .. .. xx CS contains 261 bytes of data (xx).

2.2.5 Data

This field holds the data bytes and it has a variable length which is specified in the Length field. The interpretation of the data bytes are message type specific. See the description of the message for more details about the data bytes.

2.2.6 Checksum

This field adds error-detection functionality. If all packet bytes excluding the preamble are summed and the lower byte of the result equals zero, the packet is valid and it may be processed. The checksum value of the packet should be included in the summation.

3 Messages

3.1 Introduction

This chapter describes all messages currently supported by the Xbus Master and other relevant messages for connected MT Xbus devices. All these messages are supported by the MT Communication C++ class installed with the Software Development kit. For an example see the MTComm class folder or check the Xsens program group in your start menu.

Only the special acknowledge messages are listed and not the default acknowledgement messages. Acknowledgement messages always have a MID value one higher than the previous message. Default acknowledgement messages end with the suffix 'Ack' and have no data field (i.e. Length = 0). For example, the **WakeUp** message (MID = 62 (0x3E)) can be acknowledged with **WakeUpAck**. This message should have a MID value of 63 (0x3F) and no data field. Some acknowledgement messages do have a data field. These messages are listed in this chapter to describe the meaning of the data byte(s).

Some messages can be sent to both the Xbus Master as the Motion Tracker(s) dependant on the message BID value. For example, if **ReqDID** is sent to Xbus Master with BID 255 (0xFF), the host will receive the device ID of the Xbus Master (in a **DeviceID** message). If a BID value of 1 (0x01) is used the Motion Tracker with BID 1 will send its device ID. See **SetBID** message.

3.2 WakeUp messages

3.2.1 WakeUp

Direction	To host
MID	62 (0x3E)

When Xbus Master is turned on or has been reset this message is sent to the host. If the host sends **WakeUpAck** within 500ms after reception of this message, the Xbus Master enters Config state else AutoStart state. See chapter 4 for more information about States.

3.3 Config state messages

3.3.1 ReqDID

Direction	To Xbus Master/Motion Tracker
Broadcast BID	Not allowed
MID	0 (0x00)

Request to send the device identifier (or serial number). Xbus Master or Motion Tracker acknowledges by sending **DeviceID** message.

3.3.2 DeviceID

Direction	To host
MID	1 (0x01)
DATA	IDHH IDHL IDLH IDLL (4 bytes)

Acknowledge of **ReqDID** message. Data field contains device ID. More information about the format of device ID, see chapter 5.1.

3.3.3 InitBus

Direction	To Xbus Master
MID	2 (0x02)

Initialize the bus and Xbus Master acknowledges with device IDs of the connected Motion Trackers. See **InitBusResults** message. All connected Motion Trackers are reset and set in Config state. The Motion Trackers are auto-assigned with a bus identifier (BID) values 1 and up depending on the deviceID (sorted from high to low). For manual assigned of the BID use the **SetBID** message. See also in **SetBID** message description.

3.3.4 InitBusResults

Direction	To host
MID	3 (0x03)
DATA	[IDHH IDHL IDLH IDLL] repeated for n connected MT (4*n bytes)

Acknowledge of **InitBus** message. Motion Trackers connected to the Xbus Master are now correctly initialized and the data field contains the device IDs of all found Motion Trackers.

3.3.5 SetBID

Direction	To Motion Tracker
Broadcast BID	Required
MID	6 (0x06)
DATA	IDHH IDHL IDLH IDLL BID (5 bytes)

After issuing a **Reset** message or at power-up the Motion Trackers are in Init state. Use this message to assign a bus identifier (BID) to a Motion Tracker. The Motion Tracker specified with the device identifier in the data field will automatically switch to Config state. It will also acknowledge the message using its assigned BID value (specified in the last data byte of the **SetBID** message). Ensure that the issued BIDs start with one and are all sequential. Do not assign multiple Motion Trackers the same BID.

The Xbus Master records the highest BID value and will sample the Motion Trackers starting with BID 1 up to highest BID value in the Measurement state. If the BID assignment is incorrect or needs to be changed, use the **Reset** message to clear the highest BID value.

3.3.6 ReqPeriod

Direction	To Xbus Master
MID	4 (0x04)

Request the current sample period. The Xbus Master replies with **ReqPeriodAck**. The data field of this message contains the sample period. For the description of the data field see **SetPeriod**.

3.3.7 SetPeriod

Direction	To Xbus Master
MID	4 (0x04)
DATA	PERIODH PERIODL (2 bytes)

Sets the sampling period of the Xbus Master used in measurement state. This overrides the default sampling frequency of 100Hz. The setting is stored in non-volatile memory and it is used until it is overwritten by **SetPeriod** or when the default setting is restored.
The data field contains a 16-bit value indicating the length of the period. Resolution is in (1/115200) seconds, i.e. 8.68 us. The range is specified between 225 (0x00E1) and 11520 (0x2D00) and corresponds with sampling frequency of 10-512Hz.

For example, a sampling frequency of 50Hz corresponds with a period length of 2304 (0x0900). PERIODH would then be 9 (0x09) and PERIODL 0 (0x00). If the settings are restored to default values the sample period is set to 10ms (= 100Hz).[1]

3.3.8 AutoStart

Direction	To Xbus Master
MID	6 (0x06)

When the Xbus Master receives this message the Xbus Master starts sampling the Motion Trackers with the current sample period (default 10ms). See **SetPeriod** message for changing sample period. But first, the bus is (re)scanned and found Motion Trackers are assigned with BIDs. The Motion Tracker with the highest device ID will have BID number 1, second highest number 2, etc. Then, the Xbus Master & all Motion Trackers are set into Measurement state which starts the sampling. The Xbus Master will start sending the **BusData** message. This is the default behavior if the host does not reply to the **WakeUp** message at startup.

3.3.9 BusPwrOff

Direction	To Xbus Master
MID	8 (0x08)

This message turns the bus power off and all Motion Trackers will power down (hard reset). Use **Reset** to turn bus power back on.

3.3.10 ReqConfiguration

Direction	To Xbus Master
MID	12 (0x0C)

Request the configuration settings of the device. Can be used for logging purposes – include the **Configuration** message in the log file to store the settings for offline processing of the data.

[1] To restore default settings; power off Xbus Master and press & hold power button for five seconds.

3.3.11 Configuration

Direction	To host
MID	13 (0x0D)
DATA	CONFIGURATION (See below)

Acknowledge of **ReqConfiguration** message. Data field contains the current configuration of the Xbus Master and attached MTi/MTx.

CONFIGURATION

The CONFIGURATION data contains the following information. For more information about the different fields check the corresponding message description.

offset (B)	length (B)	field description
0	4	Master device ID
4	2	Sampling period
6	2	Output skip factor
8	2	Syncin settings - Mode
10	2	Syncin settings - Skip Factor
12	4	Syncin settings - Offset
16	8	Date, format YYYYMMDD (can be set by host)
24	8	Time, format HHMMSSHH (can be set by host)
32	32	Reserved (host)
64	32	Reserved (client)
96	2	Number of devices
Repeat the following block for all attached devices		
98	4	Device ID
102	2	Data length of sensor message
104	2	Output mode
106	4	Output settings
110	8	Reserved

3.3.12 GoToMeasurement

Direction	To Xbus Master/Motion Tracker(s)
Broadcast BID	Allowed
MID	16 (0x10)

Sets the device from Config state to Measurement state. If the Xbus Master is addressed (BID = 255 (0xFF)) then all Motion Tracker in Config state are switched to Measurement state.

3.3.13 ReqFWRev

Direction	To Xbus Master/Motion Tracker
Broadcast BID	Not allowed
MID	18 (0x12)

Request to send the firmware revision.
Xbus Master or Motion Tracker acknowledges by sending **FirmwareRev** message.

3.3.14 FirmwareRev

Direction	To host
MID	19 (0x13)
DATA	MAJOR MINOR REV (3 bytes)

Acknowledge of **ReqFWRev** message. Data field contains firmware code (major, minor, revision part).

3.3.15 ReqBluetoothDisabled

Direction	To Xbus Master
MID	20 (0x14)

Requests whether Bluetooth is disabled or not. Xbus Master replies with **ReqBluetoothDisabledAck** and the data field specifies whether Bluetooth is disabled or not. See **DisableBluetooth** for description of the data field.

3.3.16 DisableBluetooth

Direction	To Xbus Master
MID	20 (0x14)
DATA	DB (1 byte)

If DB equals 1 (0x1) the Xbus Master will not use Bluetooth communication and only the serial interface. The setting is stored in non-volatile memory and therefore Bluetooth will be disabled until it is changed using with this message with DB = 0 (0x0). This message is acknowledged using **DisableBluetoothAck**.
If the settings are restored to default values Bluetooth will be disabled (DB = 1 (0x1)).[2]

[2] To restore default settings; power off Xbus Master and press & hold power button for five seconds.

3.3.17 ReqOutputMode

Direction	To Xbus Master
MID	22 (0x16)

Requests the output mode of Xbus Master. The Xbus Master replies with **ReqOutputModeAck** to inform the host of the output mode. For description of the different output modes see **SetOutputMode**.

3.3.18 SetOutputMode

Direction	To Xbus Master
MID	22 (0x16)
DATA	OM (1 byte)

Using this message enables or disables the serial output while a Bluetooth connection is active (dual output). If OM is set to 1 (0x1) the serial port output pins of the Xbus Master are enabled. If OM is set to 0 (0x0) and Bluetooth is used the serial port will be disabled. This results in lower power consumption. Note that serial port baudrate will be set to 460k8 bps and that only the serial port output signals are enabled. I.e. it is not possible to control the Xbus Master using the serial port if the output mode is set to 1 (0x1) and a Bluetooth link is active. Use Bluetooth communication link instead. This message is acknowledged by **SetOutputModeAck**.
If the settings are restored to default values dual output will be disabled (OM = 0 (0x0)).[3]

3.3.19 ReqBaudrate

Direction	To Xbus Master
MID	24 (0x18)

Requests the current baudrate setting. The Xbus Master replies with **ReqBaudrateAck** which contains a data byte that states the baudrate setting. For description of data field see **SetBaudrate** message.

3.3.20 SetBaudrate

Direction	To Xbus Master
MID	24 (0x18)
DATA	BR (1 byte)

This message changes the baudrate of the serial port. The new baudrate will be stored in non-volatile memory and will become active after a **Reset** or power cycle. See table for the different baudrate and the corresponding value of BR. Xbus Master replies with **SetBaudrateAck**.

[3] To restore default settings; power off Xbus Master and press & hold power button for five seconds.

Baudrate	BR value
921k6	128 (0x80)
460k8	0 (0x0)
230k4	1 (0x1)
115k2	2 (0x2)
76k8	3 (0x3)
57k6	4 (0x4)
38k4	5 (0x5)
28k8	6 (0x6)
19k2	7 (0x7)
14k4	8 (0x8)
9k6	9 (0x9)

Note that if dual output is enabled (output mode is set to 1) and a Bluetooth link is active the serial port baudrate will set to 921k6 and will override the current setting.
If the settings are restored to default values the baudrate will be set to 115k2 bps.[4]

3.3.21 ReqSyncMode

Direction	To Xbus Master
MID	26 (0x1a)

Requests current sync mode. The Xbus Master replies with **ReqSyncModeAck** and its data field contains the current sync mode. See **SetSyncMode**.

3.3.22 SetSyncMode

Direction	To Xbus Master
MID	26 (0x1a)
DATA	SM (1 byte)

Sets the sync mode to one of the following states:

- ON/OFF mode (SM = 0 (0x0)
- PWM mode (SM = 1 (0x1))
- Master mode (SM = 16 (0x10))
- Slave mode (SM = 32 (0x20))
- Toggle mode (SM = 64 (0x40))

See User Manual, chapter 'Xbus Master Settings' for more details. The message is replied with **SetSyncModeAck**.
If the settings are restored to default values the sync mode will be set to ON/OFF mode.[4]

3.3.23 ReqProductCode

Direction	To Xbus Master
MID	28 (0x1c)

Request to send the product code. The Xbus Master acknowledges by sending the ProductCode message.

[4] To restore default settings; power off Xbus Master and press & hold power button for five seconds.

3.3.24 ProductCode

Direction	To host
MID	28 (0x1d)
DATA	PRODUCT CODE (max 20 bytes)

Acknowledge of **ReqProductCode** message. Data field contains the product code string in ASCII format, e.g. XM-B-XB2.

3.3.25 ReqErrorMode

Direction	To Xbus Master
MID	26 (0x82)

Requests current error mode. The Xbus Master replies with **ReqErrorModeAck** and its data field contains the current error mode. See **SetErrorMode**.

3.3.26 SetErrorMode

Direction	To Xbus Master
MID	26 (0x82)
DATA	EM (2 byte)

Sets the error mode to either IGNORE/STOP mode.
This action is taken if an internal SRAM buffer overflow occurs. This buffer is used for sending data to the host in Measurement state. If this happens, it may indicate a bad Bluetooth connection, or the used Sample Frequency may be set to high. In STOP mode, if a buffer overflow occurs, the Xbus Master will send an Error message with Measurement failed – code 7 in the data field and enter Config state.

3.4 Measurement state messages

3.4.1 GoToConfig

Direction	To Xbus Master/Motion Tracker(s)
Broadcast BID	Allowed
MID	48 (0x30)

Sets the device from Measurement state to Config state. If the Xbus Master is addressed (BID = 255 (0xFF)) then all Motion Tracker in Measurement state are switched to Config state.

3.4.2 BusData

Direction	To host
MID	50 (0x32)
DATA	SCH SCL [depends on type of Motion Trackers] (2 + x bytes)

This message is sent every sample period when the Xbus Master is in the Measurement state. The data field contains a sample counter (SCH & SCL) and the data of all Motion Trackers. The sample counter is a two byte counter and is increased every sample period. It wraps to zero if higher than 65535. After the sample counter the data field contains the data of each connected Motion Tracker. First the data of Motion Tracker with BID 1 then data of Motion Tracker with BID 2, etc. There are no boundary bytes to separate the data.

For example, if 3 Motion Trackers in raw sensor data mode are connected to the bus, the data field of the **BusData** message would contain a total of 62 bytes (3*20 + 2).

3.5 All state messages

3.5.1 Reset

Direction	To Xbus Master/Motion Tracker(s)
Broadcast BID	Allowed
MID	64 (0x40)

Resets the device specified in the BID field. If BID is zero then all Motion Trackers are reset to Init state. In this case the max BID value used in Measurement state by the Xbus Master is reset. For more info see **SetBID** description. If the message is sent to the Xbus Master the bus power will be restored and it will also reset all Motion Trackers.

3.5.2 Error

Direction	To host
MID	66 (0x42)
DATA	ERRCODE (1 byte)

This message reports an error. Either in response to a message sent or an error during initialization or measurement. The data field contains an error code. See chapter 5.5 for a listing of error codes and their description.

3.5.3 XMPwrOff

Direction	To Xbus Master
MID	68 (0x44)

Turns the Xbus Master off.

3.5.4 ReqBatLvl

Direction	To Xbus Master
MID	136 (0x88)

Requests the current battery level The Xbus Master replies with **BatLvl** which contains a data byte that states the battery level.

3.5.5 BatLvl

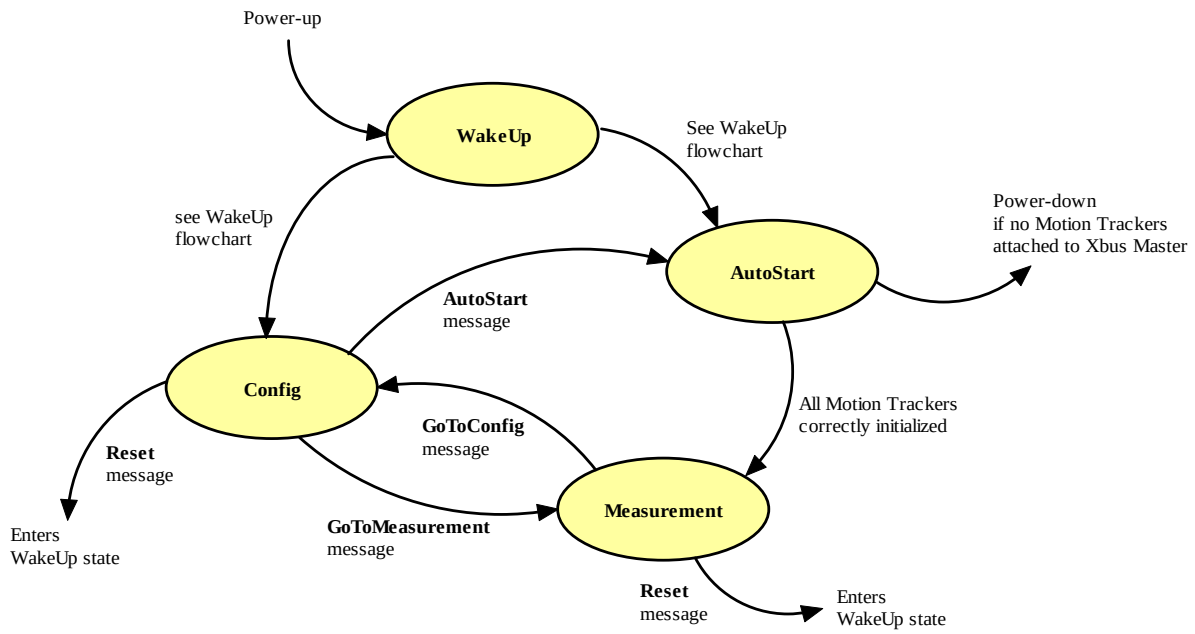
Direction	To host
MID	137 (0x89)
DATA	BATLVL (1 byte)

Acknowledge of the **ReqBatLvl** message. Data field contains the current battery level.

4 States

4.1 Introduction

The Xbus Master operates in several states. There are two temporary states and two main states. WakeUp and AutoStart are the two temporary states. If the Xbus Master enters one of these states and it will automatically change to another state. If the Xbus Master enters the main states, Config and Measurement, only the host can set the Xbus Master to another state.
The following state flow chart shows how the Xbus Master switches from state to state.



The following subchapters describe the functionality of the states.

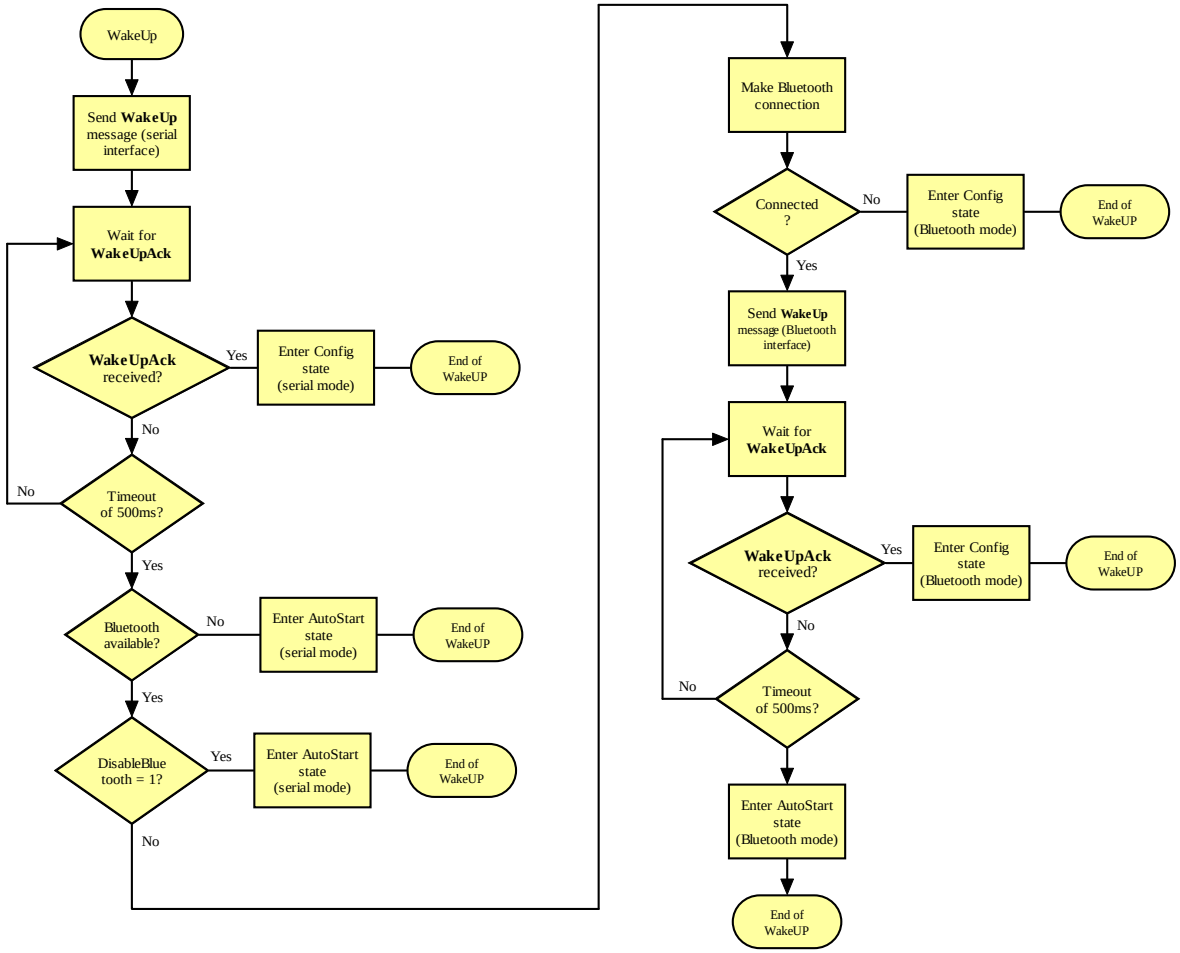
4.2 WakeUp

The Xbus Master has two main states; it operates in the Config or Measurement state. Dependant on actions taken in the temporary WakeUp state the Xbus Master enters one of these two main states at power up. To select the operating state after the wakeup state the host should either send an acknowledge message or not. When the **WakeUp** message is sent by the Xbus Master the host should respond within 500ms with a **WakeUpAck** message to enter the Config state. Otherwise the Xbus Master will enter the Measurement state. But before this, the Xbus Master will scan the bus and initialize the Motion Trackers. This is done in the temporary state called AutoStart, see next subchapter. If no Motion Trackers are found the Xbus Master will power down automatically.

In case of an Xbus Master with Bluetooth, the WakeUp procedure differs. The start of the WakeUp procedure is the same. I.e. first a **WakeUp** message is sent using the serial interface to the host. If the host wants to use the serial interface it must replied to this message with the **WakeUpAck** message within 500ms (using serial interface). The Xbus Master will enter the Config state. If the host does not reply the Xbus Master will check

the DisableBluetooth setting (see **DisableBluetooth** message). If this setting is 1 the Xbus Master will skip the Bluetooth initialization and enters AutoStart state in serial mode. If the setting is 0 Bluetooth is enabled. The Xbus Master will try to connect to the remote Bluetooth device. If this is not possible, the Xbus Master enters the Config state with Bluetooth enabled. The remote device can now try to search the Xbus Master and connect to its Bluetooth Serial Port. In this case, the Xbus Master will stay in Config state.

If the Xbus Master succeeds in making a connection with the remote device it will send the **WakeUp** message to the host using Bluetooth Serial Port Profile. If the Xbus Master receives the **WakeUpAck** message within 500 msecs it enters the Config state, else the AutoStart state. This is the exact the same behaviour as using the serial interface.



4.3 AutoStart

This temporary state is entered when the host does not send a **WakeUpAck** message (or not in time) or when the **AutoStart** message is sent to the Xbus Master. In this state the bus is reset and scanned. The device IDs of all found Motion Trackers are sorted from high to low and then all the Motion Trackers are assigned a BID starting with one and using the order of the sorted ID list. This order is also used in the **BusData** message.

If all Motion Trackers have a BID, the Xbus Master sets all Motion Trackers to Measurement state and also itself. The current sample period is used (default = 10ms). See **SetPeriod**.

4.4 Config

4.4.1 General

During the Config state various messages can be sent to the Xbus Master to change default settings or retrieve information. For example, **InitBus** can be used in the Config state to retrieve the device identifiers of the connected Motion Trackers. For a complete list of valid Config state messages see chapter 3.3.

4.4.2 Host to Motion Tracker communication

In the Config state it is possible to send and receive messages directly to/from the Motion Trackers. Use a BID value other than 255 (0xFF) in the packet and the Xbus Master will copy the packet directly to the bus. All bytes sent by the Motion Trackers will be copied and send to the host. For example, the **SetBID** message (see chapter 3.3 - **SetBID**) is directly copied to the bus and the acknowledge message comes from the specified Motion Tracker. The Xbus Master acts as a transparent device between host and bus.

4.5 Measurement

4.5.1 Test run

Before the measurement (sampling) starts the Xbus Master will execute a test run. In the test run all Motion Trackers in Measurement state are ordered to start the sampling procedure. Then the Motion Trackers are polled one by one to send their data. The Xbus Master will check if the (default) sample frequency is correctly set to obtain all Motion Tracker data in time. If not, the Xbus Master will send a not-acknowledge message with the correct error code. If the test run has been finished successfully the Xbus Master will send an acknowledge message. This message is not sent if coming from AutoStart procedure. Please be aware that if the test run is performed successfully this will not imply that the measurement will run successfully. It is possible that the host communication at the baudrate of 115k2 bps is too slow.

4.5.2 Measurement

After the test run the Xbus Master starts the actual measurement procedure. In fact, it is just like the test run except the data is now transmitted to the host with the **BusData** message. This message is sent at every sample instance. The RTS line (CTS at host side) of the RS-232 port will pulse high for 100us every time the Motion Trackers are ordered to sample their sensors. This is not available if Bluetooth communication is used.

It is possible that if the sample frequency is too high the **BusData** message carrying the data of the Motion Trackers can not be sent in time. If this is the case, the Xbus Master will either stop the measurement and return to config state (beep will sound) or it will stay in the Measurement state sending **BusData** messages in which the sample counter values are not successive. If the Xbus Master stops sending **BusData** messages, the last **BusData** message is invalid because it misses Motion Tracker data. The checksum is automatically made

invalid by the Xbus Master. The solution to this is to lower the sampling frequency or to connect less Motion Trackers to the Xbus Master.

4.5.3 Triggering sampling instance

Before the Xbus Master orders the Motion Trackers to start sampling it will check the CTS (RTS at host side) state of the RS-232 port. If it is cleared the Xbus Master will not continue and will wait until the control line is set. Keep this control line enabled to ensure that the Xbus Master will not interrupt the measurement and takes samples at the current sample frequency.

This functionality is disabled when using Bluetooth communications.

5 Appendix

5.1 Examples & additional info

5.1.1 Start measurement @ 100Hz using default order

- If current sample period is not equal to 10ms, restore default settings
- Connect the Motion Tracker(s) to Xbus Master
- Turn Xbus Master on & set the RS-232 control line RTS

5.1.2 Change sample frequency

- Enter Config state
- Issue **SetPeriod** with correct sample frequency
- Enter Measurement state, for example by sending **AutoStart** to Xbus Master or reset the Xbus Master.

5.1.3 Obtain the Motion Tracker's data order of BusData message

- Enter Config state
- Send **InitBus** message
- The **InitBusResults** data field contains the device IDs of all connected Motion Trackers. The order of the ID is also used in the **BusData** message
- Continue with measurement by sending **AutoStart**.

5.1.4 Change the Motion Tracker's data order of BusData message

- Enter Config state
- Send **InitBus** message
  This message reset the Motion Trackers to Init state. If device IDs are known **InitBus** does not have to be send. Though the Motion Trackers must reside in Init state. Use **Reset** or go directly to Config state after WakeUp state to ensure that the Motion Trackers are in Init state.
- Send **SetBID** to the Motion Trackers using IDs from **InitBusResults**
  The BID value in **SetBID** will define the order of the Motion Tracker data in **BusData** message. Start with one and increase by one for every next Motion Tracker
- Enter Measurement state
  Use **GoToMeasurement** message with BID 255 (0xFF). DO NOT send the **AutoStart** message because this would reset the Motion Trackers and the BIDs are assigned using the order of the sorted device ID list.

5.1.5 Pausing the data stream

There are two ways to stop the data stream from Xbus Master temporarily. Either make the RTS line inactive (low) or send the **GoToConfig** (BID = 255/0xFF) message to Xbus Master. To continue with measurement, make the RTS line high or send **GoToMeasurement** to Xbus Master if the Xbus Master is in Config state.

5.1.6 Log data of subset of connected Motion Trackers

It is possible to log data of a subset of the connected Motion Trackers. The following procedure must be used every time the subset has to be changed.

- Enter Config state (use BID 255 (0xFF) in **GoToConfig** message)
- Send **Reset** message (broadcast) or send **InitBus** message
- Use the **SetBID** message for all Motion Trackers that should be active in the Measurement state. Always start with BID value 1 for assignment and increase by one for each next Motion Tracker.
- Send **GoToMeasurement** message (use BID 255 (0xFF)) & set RTS high (or leave it high).

The Motion Trackers that are assigned a BID are now sampled. The other Motion Trackers will stay in the init state. Repeat this procedure if the subset needs to be changed.

5.1.7 Understanding BusData message layout

At every sample instance the **BusData** message is sent to the host when the Xbus Master is in measurement state. So, if the sample frequency is 100Hz, 100 **BusData** messages are sent to the host per second. This message contains the data of all connected Motion Trackers. The data bytes of the data field have to be split for each Motion Tracker if manual processing is desired. This is an example of a **BusData** message in case two Motion Trackers (MTx in quaternion output mode) are connected:

FA FF 32 22 05 51 3D 70 09 E5 BC 1A 3A B4 3B 09 8D 24 BF 7F 8C 50 3E 22 19 33 BD BD 2A AD 3C 1F 92 05 3F 7B A6 C0 DA

The message has six main parts:

FA	preamble
FF	BID of Xbus Master
32	MID of **BusData**
22	Length of data field
05-C0	Data field (34 bytes)
DA	Checksum

The above layout is always fixed except for the variable length of the data field.
The data field has 34 bytes. It has three parts:

05 51	Sample counter (high & low), decimal value = 1361
3D-50	Data of Motion Tracker with BID 1 (16 bytes)
3E-C0	Data of Motion Tracker with BID 2 (16 bytes)

The first two bytes of the data field are reserved for the sample counter (16 bits value). The previous **BusData** message should have a sample counter value of 05 50 and the next should be 05 52. If this is not the case, data has been lost during logging or sample frequency is set too high. The next bytes in the data field contain the data of the Motion Trackers. Because these are two MTx in quaternion output mode, the data consists of 16 bytes for each Motion Tracker. Length and representation of these bytes depend on selected output mode of the Motion Tracker. The 16 bytes of the first part are 4 bytes for each float in the output quaternion ([w, q0, q1, q2]). Each float is 4 bytes long and corresponds with the single-precision floating-point value as defined in the IEEE 754 standard (= float). The same order is used as the standard MTx packet layout. See MTi/MTx User Manual (and Technical Documentation) for more information.

5.1.8 Device identifier (DID) information

The device identifier (as in the DeviceID message) is a unique identifier. I.e. all Xbus devices have a different device identifier. The identifier has three parts:

Company ID	Type ID	Serial ID
1 byte	1 byte	2 bytes

Every company that designs products conforming to the Xbus specification will have its own company identifier. Xsens will assign these identifiers to ensure compatibility with the devices of other companies. The company ID of Xsens is 0 (0x00).

The type identifier is used to identify the different type of products of a company. For example the Motion Tracker MTx Xbus has type ID 50 (0x32). See table.

Product	Type ID
Xbus Master – B	18 (0x12)
Xbus Master – B Bluetooth	19 (0x13)
Wireless Receiver WR-A	33 (0x21)
Motion Tracker MTi/MTx 232	48 (0x30)
Motion Tracker MTi 422	49 (0x31)
Motion Tracker MTx Xbus	50 (0x32)
Motion Tracker MTi/MTx 485	51 (0x33)
Motion Tracker MTi-G	80 (0x50)

The last two bytes are reserved for a serial number.

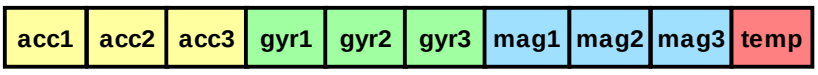
5.1.9 Motion Tracker data format

As seen in the previous subchapter Xsens has different types of Motion Trackers. These different types also have different data outputs. I.e. the format of the data field of a **BusData** message depends on the used Motion Tracker types.

When using the MTx, the format of the data field depends on the Output Mode and Output Settings. The different options are listed below separately. Please refer to the **MTi and MTx User manual and Technical Documentation** and the **MTi and MTx Low-level communication documentation** for details.

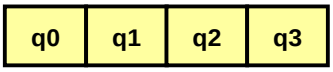
Some of the output modes can also be used together. If not specified otherwise each data value is 4 bytes long and corresponds with the single-precision floating-point value as defined in the IEEE 754 standard (= float).

Un-calibrated raw data output mode (20 bytes)
Contains the un-calibrated raw data output of the accelerations, rate of turn and magnetic field in X, Y and Z axes. These values are equal to the analog-digital converter readings of the internal sensors. The data values are NOT float values but 16 bit unsigned integer values.



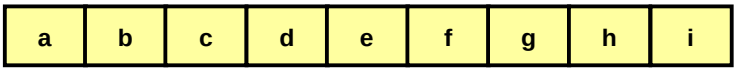
Calibrated data output mode (36 bytes)
Contains the calibrated data output of the accelerations, rate of turn and magnetic field in X, Y and Z axes in floats.

Orientation data output mode – quaternion (16 bytes)
Contains the q0, q1, q2 and q3 quaternions, in floats, that represent the orientation of the MTi / MTx



Orientation data output mode – Euler angles (12 bytes)
Contains the three Euler angles, in floats, that represent the orientation of the MTi / MTx

Orientation data output mode – Matrix (36 bytes)
Contains the rotation matrix (DCM), in floats, that represents the orientation of the MTi / MTx.



NOTE: Please refer to the **MTi and MTx User Manual (and Technical Documentation)** for the interpretation of the data values.

Sample counter (2 bytes)
The (optional) sample counter is a 16 bit unsigned integer value that is increased for each transmission of the **MTData** message. If its maximum value is reached, i.e. 65535 (0xFFFF), it will wrap and start at zero again. See also **SetErrorMode** for the relation between the sample counter and the error mode.

5.2 Maximum sample frequency

The maximum sample frequency (i.e. update rate) depends upon four different settings:

1. number of connected Motion Trackers
2. time needed to sample data from all Motion Trackers on the bus
3. time needed to output the data to the host from the Xbus Master
4. data output settings of MTi/MTx (the amount of data)

As a guideline, the following tables are an indication of possible sample frequencies:

Maximum sample rates with 5 MTx connected on the bus, depending on the output mode:

MTx Output mode	Serial 115k2	Serial 460k8	Bluetooth
Raw data	100 Hz	200 Hz	150 Hz
Calibrated data	50 Hz	100 Hz	64 Hz
Quaternion data	120 Hz	120 Hz	120 Hz
Quaternion + Calibrated data	25 Hz	64 Hz	50 Hz

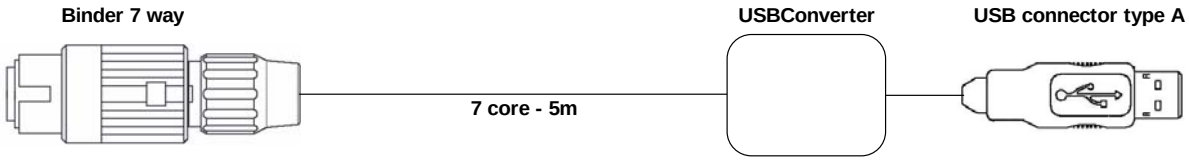
Maximum update rate with 10 MTx connected on the bus, depending on the output mode:

MTx Output mode	Serial 115k2	Serial 460k8	Bluetooth
Raw data	50 Hz	100 Hz	64 Hz
Calibrated data	25 Hz	50 Hz	25 Hz
Quaternion data	64 Hz	64 Hz	64 Hz
Quaternion + Calibrated data	10 Hz	25 Hz	25 Hz

5.3 Pin-out and cables

5.3.1 Xbus Master <> Host cable (XM USB-RS232 Cable)

The standard Xbus Master <> host cable is 5m long and has an USB type A connector and a Binder 710, 7 way male connector. The cable wires TX, RX, RTS, CTS and Ain. See figure.



Contact	Definition	Unitronic cable	Elitronic cable
1	SYNC	Blue	White
2	RX (XM)	Brown	Brown
3	TX (XM)	Beige	Green
4	NC	Yellow	Yellow
5	Ground	Yellow-green	Grey
6	RTS (XM)	Black	Pink
7	CTS (XM)	Green	Blue

5.3.2 Xbus cable

The Xbus cable supplied with the Motion Tracker Xbus version is a 1:1 cable with five contacts.

Contact	Definition	Grey Unitronic	Grey Elitronic	Black cable
1	Z / B	Beige	White	White
2	VCC	Black	Yellow	Red
3	Analog IN	Blue	Brown	Blue
4	GND	Yellow-green	Grey	Black
5	Y / A	Brown	Green	Green

5.3.3 Sync connector

The sync connector on the Xbus Master is described in the XM-B User Manual.

5.4 Quick reference – Messages

WakeUp state messages

Message	MID	Direction	Description
WakeUp	62 (0x3E)	To host	At power up Xbus Master sends this message to host
WakeUpAck	63 (0x3F)	To XM	If received within 500ms after WakeUp Xbus Master enters config state else measurement state

Config state messages

Message	MID	Direction	Description
ReqDID	0 (0x00)	To XM/MT	Host request device ID of Xbus Master
DeviceID	1 (0x01)	To host	Xbus Master acknowledges request by sending its ID
InitBus	2 (0x02)	To XM	Xbus Master starts bus scan
InitBusResults	3 (0x03)	To host	Xbus Master reports finished with bus scan & sends slave IDs
ReqPeriod	4 (0x04)	To XM	Requests current period
ReqPeriodAck	5 (0x05)	To host	Xbus Master returns sample period
SetPeriod	4 (0x04)	To XM	Host sets sample period (10-512Hz)
SetPeriodAck	5 (0x05)	To host	Xbus Master acknowledges SetPeriod message
SetBID	6 (0x06)	To MT	Assign BID to Motion Tracker (use broadcast BID)
SetBIDAck	7 (0x07)	To host	Motion Tracker acknowledge using new BID
AutoStart	6 (0x06)	To XM	Auto assign of BIDs and enter measurement state
AutoStartAck	7 (0x07)	To host	Xbus Master acknowledges AutoStart message
ReqBusPwr	8 (0x08)	To XM	Request current status of bus power
ReqBusPwrAck	9 (0x09)	To host	Xbus Master returns current status
SetBusPwr	8 (0x08)	To XM	Turn bus power on or off
SetBusPwrAck	9 (0x09)	To host	Xbus Master acknowledges SetBusPwr message
ReqConfiguration	12 (0x0C)	To XM	Request the configuration of device. For logging/quick setup purposes
Configuration	13 (0x0D)	To host	Xbus Master acknowledges using configuration data
GoToMeasurement	16 (0x10)	To XM/MT	Xbus Master enters measurement state
GoToMeasurementAck	17 (0x11)	To host	Xbus Master acknowledges GoToMeasurement message

ReqFWRev	18 (0x12)	To XM/MT	Host requests firmware revision of Xbus Master
FirmwareRev	19 (0x13)	To host	Xbus Master acknowledges request by sending its firmware revision
ReqBluetoothDisable	20 (0x14)	To XM	Request whether Bluetooth enabled or disabled
ReqBluetoothDisableAck	21 (0x15)	To host	Xbus Master returns with operation mode of Bluetooth
DisableBluetooth	20 (0x14)	To XM	En- or disable Bluetooth
DisableBluetoothAck	21 (0x15)	To host	Xbus Master acknowledges DisableBluetooth message
ReqOutputMode	22 (0x16)	To XM	Requests current output mode
ReqOutputModeAck	23 (0x17)	To host	Xbus Master returns output mode
SetOutputMode	22 (0x16)	To XM	Host sets output mode
SetOutputModeAck	23 (0x17)	To host	Xbus Master acknowledges SetOutputMode message
ReqBaudrate	24 (0x18)	To XM	Requests current baudrate of serial port
ReqBaudrateAck	25 (0x19)	To host	Xbus Master returns baudrate of serial port
SetBaudrate	24 (0x18)	To XM	Host sets baudrate of serial port
SetBaudrateAck	25 (0x19)	To host	Xbus Master acknowledges SetBaudrate message
ReqSyncMode	26 (0x1a)	To XM	Requests current sync mode
ReqSyncModeAck	27 (0x1b)	To host	Xbus Master returns sync mode
SetSyncMode	26 (0x1a)	To XM	Host sets sync mode
SetSyncModeAck	27 (0x1b)	To host	Xbus Master acknowledges SetSyncMode message
ReqProductCode	28 (0x1C)	To XM	Request Product Code
ProductCode	29 (0x1D)	To host	Xbus Master acknowledges using Productcode
ReqErrorMode	130 (0x82)	To XM	Request current action if SRAM buffer overflow occurs
ReqErrorModeAck	131 (0x83)	To host	Xbus Master returns current action (0=ignore, 1=Stop)
SetErrorMode	130 (0x82)	To XM	Host sets action if SRAM buffer overflow occurs
SetErrorModeAck	131 (0x83)	To host	Xbus Master acknowledges SetErrorMode Message

Measurement state messages

Message	MID	Direction	Description
GoToConfig	48 (0x30)	To XM/MT	Xbus Master enters config state
GoToConfigAck	49 (0x31)	To host	Xbus Master acknowledges GoToConfig message
BusData	50 (0x32)	To host	Xbus Master message with data from bus

Valid in all states

Message	MID	Direction	Description

Reset	64 (0x40)	To XM/MT	Xbus Master resets itself and Motion Trackers
ResetAck	65 (0x41)	To host	Xbus Master or Motion Tracker acknowledges Reset
Error	66 (0x42)	To host	Error message
XMPwrOff	68 (0x44)	To XM	Power down Xbus Master
ReqBatLvl	136 (0x88)	To XM	Request current battery level
BatLvl	137 (0x89)	To host	Xbus Master returns battery level

5.5 Error descriptions

Error code Dec	Hex	Description
1	0x01	No bus communication possible
2	0x02	Bus not ready for measurement
3	0x03	Period sent is invalid
4	0x04	Message sent is invalid
16	0x10	Initialization of bus failed - code 1
17	0x11	Initialization of bus failed - code 2
18	0x12	Initialization of bus failed - code 3
20	0x14	SetBID procedure failed - code 1
21	0x15	SetBID procedure failed - code 2
24	0x18	Measurement failed - code 1
25	0x19	Measurement failed - code 2
26	0x1A	Measurement failed - code 3
27	0x1B	Measurement failed - code 4
28	0x1C	Measurement failed - code 5
29	0x1D	Measurement failed - code 6
35	0x23	Measurement failed – code 7

The following section provides more detailed information about the cause of the error.

5.5.1 Error code 1 (0x01) - No bus communication possible

Problem: Bus has no power
Solution: Send **Reset** or turn Xbus Master off & on

5.5.2 Error code 2 (0x02) - Bus not ready for measurement

Problem: **InitBus** and/or **SetBID** are not issued
Solution: Use **InitBus** & **SetBID** to properly initialize the Motion Trackers or use **AutoStart**.

5.5.3 Error code 3 (0x03) - Period sent is invalid

Problem: Period does not comply with valid range
Solution: Resend **SetPeriod** with valid argument

5.5.4 Error code 4 (0x04) - Message sent is invalid

Problem: Message sent is not implemented or invalid
Solution: Check message identifier and if message is valid in current state

5.5.5 Error code 16 (0x10) - Initialization of bus failed - code 1

Problem: A slaves did not respond to **WaitForSetBID**
Solution: Check wires & connectors

5.5.6 Error code 17 (0x11) - Initialization of bus failed - code 2

Problem: An incorrect answer received after **WaitForSetBID**
Solution: Check wires & connectors

5.5.7 Error code 18 (0x12) - Initialization of bus failed - code 3

Problem: After four bus-scans still undetected Motion Trackers
Solution: Check wires & connectors

5.5.8 Error code 20 (0x14) - SetBID procedure failed - code 1

Problem: No reply to **SetBID** message during SetBID procedure
Solution: Check wires & connectors

5.5.9 Error code 21 (0x15) - SetBID procedure failed - code 2

Problem: Other than **SetBIDAck** received
Solution: Check wires & connectors

5.5.10 Error code 24 (0x18) - Measurement failed - code 1

Problem: Timer overflow - period too short to collect all data from Motion Trackers
Solution: Increase period (lower sample frequency) or use fewer Motion Trackers

5.5.11 Error code 25 (0x19) - Measurement failed - code 2

Problem: Motion Tracker responds with other than **SlaveData** message
Solution: Check wires & connectors

5.5.12 Error code 26 (0x1a) - Measurement failed - code 3

Problem: Total bytes of data of Motion Trackers incl sample counter exceeds 2048 bytes
Solution: Use fewer Motion Trackers

5.5.13 Error code 27 (0x1b) - Measurement failed - code 4

Problem: Timer overflow during measurement – Motion Tracker does not respond within measurement period
Solution: Increase period (lower sample frequency), if problem continue to exist contact Xsens for support

5.5.14 Error code 28 (0x1c) - Measurement failed - code 5

Problem: Timer overflow during measurement – Motion Tracker response was not received within measurement period
Solution: Increase period (lower sample frequency) or use fewer Motion Trackers

5.5.15 Error code 29 (0x1d) - Measurement failed - code 6

Problem: No correct response from Motion Tracker during measurement
Solution: Check wires & connectors

5.5.16 Error code 35 (0x23) – Measurement failed – code 7

Problem: Transmit buffer to PC is full, cannot transmit all data to host
Solution: Increase period(lower sample frequency), if problem continue to exist contact Xsens for support