Smart choice for power

xantrex

XBM Communication Interface Specification

Application Note

XBM communications kit

976-0133-01-01 Rev A

1 Introduction

The scope of this specification is to define the communication protocol used by the Xantrex Battery Monitor (XBM).

2 Data link Layer

2.1 Message Transfer

The XBM communicates via messages. A message consists of at least 5 bytes. Every message starts with the Header / Destination address which is followed by the Source address. These addresses are used when a hub (currently not available, future product) connects two or more devices together. When connecting the XBM directly to a host PC (using the optional communication interface kit), the destination and source addresses are ignored. The next byte in the message is the Device ID followed by the Message type. Whether or not data bytes are send after the Message type byte depends on the type of message (see section 2.1.4). The last byte in the message is the End of transfer byte which closes the message. The MSB of each byte in the message represents the IDHT (IDentify Header / Trailer) bit. This bit is logic 1 for the Header / Destination address byte and the End of transfer byte and logic 0 for all the bytes between them. This simplifies the detection of a message start.

2.1.1 Header / Destination address

The IDHT of this byte is logic 1. The 7 bits following the IDHT represent the destination address. A maximum of 127 devices can be addressed, not 128 because the bit combination FFh (IDHT = logic 1, address 128 = 7 consecutive logic 1's) is defined as the last byte of the transfer (see section 2.1.6). The XBM will always output destination address 0.

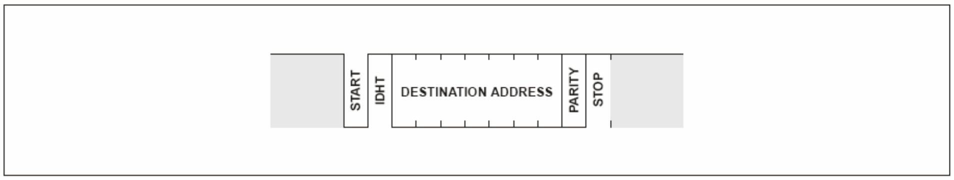


Figure 1: Header/Destination address

2.1.2 Source Address

The IDHT of this byte is logic 0. The 7 bits following the IDHT represent the address of the sender. The XBM will always output source address 0.

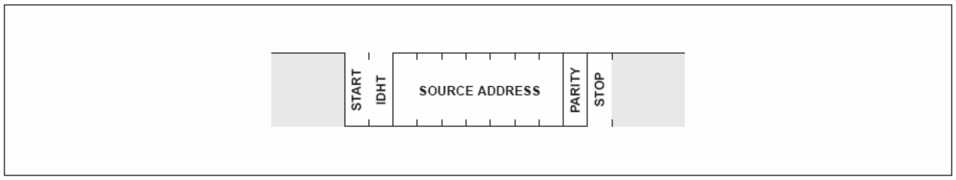


Figure 2: Source Address

2.1.3 Device ID

The IDHT of this byte is logic 0. The 7 bits following the IDHT represent what type of equipment (TBS inverter, battery monitor etc.) is sending the message. Each type of TBS equipment has its unique identification number. The Device ID of the XBM is 20h.

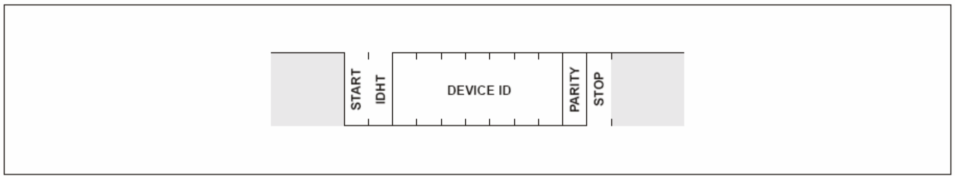


Figure 3: Device ID

2.1.4 Message type

The IDHT of this byte is logic 0. The 7 bits following the IDHT represent the message type and can be divided into five groups:

1.) System common handshake (00h..0Fh).

2.) System common commands (10h..1Fh). Messages common for all TBS products like ‘Standby on’, ‘Standby off’, ‘Alarm relays on’, ‘Alarm relays off’ etc.

3.) System exclusive commands (20h..3Fh). These are commands which are different for each type of equipment. For example, an inverter can use sys-ex command 20h for activating the ASB- / Economy-mode, while a battery monitor uses this command number for a display test.

4.) System exclusive data requests (40h..5Fh). If a device requests specific data from another device (i.e. host PC requests history data from a battery monitor) it will send a sys-ex data request. This data is then transferred to the requesting device using the numbers 60h to 7Fh.

5.) System exclusive data (60h..7Fh). These are messages that contain data, for example battery voltage, battery current or all the settings of the monitor. The data will be sent in data bytes following the message type.

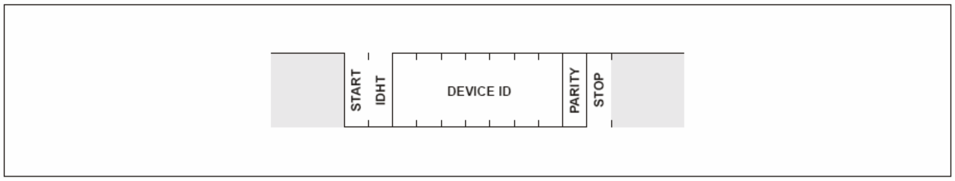


Figure 4 : Message type

See section 2.2 for an overview of all the XBM message types.

2.1.5 Data field

The data field contains data like battery voltage, battery current etc. Only ‘Sys-ex data’ messages can contain one or more data bytes. The number of data-bytes in a ‘Sys-ex data’ message is theoretically not limited. However, because of hardware limitations of the XBM, the data field can have a maximum length of 27 bytes.

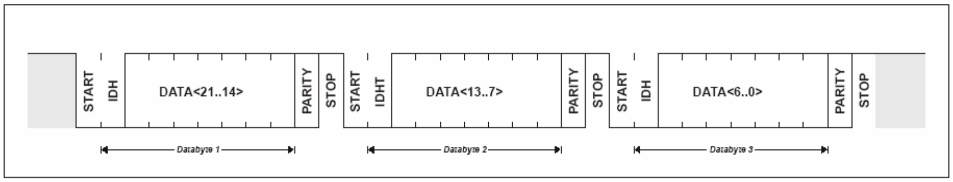


Figure 5 : Data field (example with 3 databytes)

2.1.6 End of transfer

The IDHT and the 7 following bits of this byte are all logic 1. This End of transfer byte is unique because no other field in a message can represent FFh.

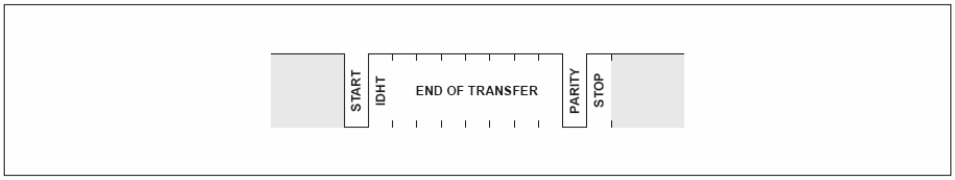


Figure 6 : End of transfer

2.2 Message type overview

This section gives an overview of all the message types the XBM understands.

2.2.1 System common handshake

0x00 ACK – Positive acknowledge.

Transmission repetition rate: on acknowledge
Direction: output / input
Data type: handshake

0x01 NACK – Negative acknowledge.

Transmission repetition rate: on negative acknowledge
Direction: output / input
Data type: handshake

0x02 NACK + *repeat request* – Negative acknowledge and request to send data again.

Transmission repetition rate: on communication error
Direction: output
Data type: handshake

2.2.2 System common commands

0x12 Alarm switch off – Switches off the alarm switch.
Transmission repetition rate: -
Direction: input
Data type: command

0x13 Alarm switch on – Switches on the alarm switch.
Transmission repetition rate: -
Direction: input
Data type: command

2.2.3 System exclusive commands

0x20 Display test off – Disables display test.
Transmission repetition rate: -
Direction: input
Data type: command

0x21 Display test on – Enables display test.
Transmission repetition rate: -
Direction: input
Data type: command

0x22 Backlight off – Turns off the backlight (provided that backlight is not on due to a key press).
Transmission repetition rate: -
Direction: input
Data type: command

0x23 Backlight on – Turns on the backlight.
Transmission repetition rate: -
Direction: input
Data type: command

0x24 Calibration mode off – Turns off calibration mode.
Transmission repetition rate: -
Direction: input
Data type: command

0x25 Calibration mode on – Turns on calibration mode, do not use this command.
Transmission repetition rate: -
Direction: input
Data type: command

0x26 Request only mode off – Turns off request only mode. Now the XBM sends data every second (this is the default factory status). Only available on firmware versions 1.01 and newer!
Transmission repetition rate: -
Direction: input
Data type: command

0x27 Request only mode on – Turns on request only mode. Now the XBM only sends data on request (requested by a sys-ex data request command). Only available on firmware versions 1.01 and newer!
Transmission repetition rate: -
Direction: input
Data type: command

0x28 Store functions – Store functions in EEPROM.
Transmission repetition rate: -
Direction: input
Data type: command

0x29 Store history – Store history in EEPROM.
Transmission repetition rate: -
Direction: input
Data type: command

0x2A Store calibration coefficients – Store calibration coefficients in EEPROM.
Transmission repetition rate: -
Direction: input
Data type: command

0x30 Reset factory settings – Restore factory settings.
Transmission repetition rate: -
Direction: input
Data type: command

0x31 Reset charge efficiency – Reset automatically calculated charge efficiency to 90.0%.
Transmission repetition rate: -
Direction: input
Data type: command

0x32 Clear history – Clear all history data except automatically calculated charge efficiency.
Transmission repetition rate: -
Direction: input
Data type: command

0x3C Up switch pressed – User has pressed the 'Up-switch'.
Transmission repetition rate: on up-switch press
Direction: output
Data type: command

0x3D Setup switch pressed – User has pressed the 'Setup-switch'.
Transmission repetition rate: on setup-switch press
Direction: output
Data type: command

0x3E Down switch pressed – User has pressed the 'Down-switch'.
Transmission repetition rate: on down-switch press
Direction: output
Data type: command

2.2.4 System exclusive data requests

0x40 Voltage request – Request for voltage data.
Transmission repetition rate: -
Direction: input
Data type: command

0x41 Current request – Request for current data.
Transmission repetition rate: -
Direction: input
Data type: command

0x42 Amphours request – Request for amphours data. Only available on firmware versions 1.01 and up!
Transmission repetition rate: -
Direction: input
Data type: command
0x44 StateOfCharge request – Request for state-of-charge data. Only available on firmware versions 1.01 and newer!
Transmission repetition rate: -
Direction: input
Data type: command

0x45 TimeToGo request – Request for time-to-go data. Only available on firmware versions 1.01 and newer!
Transmission repetition rate: -
Direction: input
Data type: command

0x46 Temperature request – Request for temperature data. Only available on firmware versions 1.01 and newer!
Transmission repetition rate: -
Direction: input
Data type: command

0x47 MonitorStatus request – Request for monitor status bits.
Transmission repetition rate: -
Direction: input
Data type: command

0x4F All parameters request – Request for all measured and calculated parameters. Only available on firmware versions 1.01 and newer!
Transmission repetition rate: -
Direction: input
Data type: command

0x50 ParameterSelect request – Request for selected parameter.
Transmission repetition rate: -
Direction: input
Data type: command

0x51 FunctionDump request – Request for function dump.
Transmission repetition rate: -
Direction: input
Data type: command

0x52 HistoryDump request – Request for history dump.
Transmission repetition rate: -
Direction: input
Data type: command

0x58 Calibration coefficients request – Request for all calibration coefficients.
Transmission repetition rate: -
Direction: input
Data type: command

0x5F FirmwareVersion request – Request for firmware version.
Transmission repetition rate: -
Direction: input
Data type: command

2.2.5 System exclusive data

0x60 Voltage – Battery voltage.
Transmission repetition rate: 1s / on request
Direction: output
Data Length: 3 bytes
Resolution: 0.01V / bit gain, 0V offset
Signed: no
Data range: 0V – 655.35V
Data type: measured
Databyte1<1..0> : Voltage<15..14>
Databyte2<6..0> : Voltage<13..7>
Databyte3<6..0> : Voltage<6..0>

0x61 Current – Battery current.
Transmission repetition rate: 1s / on request
Direction: output
Data Length: 3 bytes
Resolution: 0.01A / bit gain, 0A offset
Signed: yes
Data range: 0A – 655.35A
Data type: measured
Databyte1<2>: Sign (0 = positive current, 1 = negative current)
Databyte1<1..0> : Current<15..14>
Databyte2<6..0> : Current<13..7>
Databyte3<6..0> : Current<6..0>

0x62 Amphours – Amount of amphours removed from the battery.
Transmission repetition rate: 1s
Direction: output
Data Length: 3 bytes
Resolution: 0.1Ah / bit gain, 0Ah offset
Signed: yes
Data range: 0Ah – 2000.0Ah
Data type: calculated
Databyte1<2> : Sign (0 = positive amphours, 1 = negative amphours)
Databyte1<1..0> : Amphours<15..14>
Databyte2<6..0> : Amphours<13..7>
Databyte3<6..0> : Amphours<6..0>

0x64 StateOfCharge – State of charge of the battery in percent where 0% represents a fully discharged, and 100% a fully charged battery.
Transmission repetition rate: 1s
Direction: output
Data Length: 3 bytes
Resolution: 0.1% / bit gain, 0% offset
Signed: no
Data range: 0% – 100.0%
Data type: calculated
Databyte1<1..0> : StateOfCharge <15..14>
Databyte2<6..0> : StateOfCharge <13..7>
Databyte3<6..0> : StateOfCharge <6..0>

0x65 TimeToGo – Time to go until the battery needs to be charged.
Transmission repetition rate: 1s
Direction: output
Data Length: 3 bytes
Resolution: 1minute / bit gain, 0minutes offset, hhh:mm format
Signed: no
Data range: 0:00 – 240:00
Data type: calculated
Databyte1<1..0> : TimeToGo <15..14>
Databyte2<6..0> : TimeToGo <13..7>
Databyte3<6..0> : TimeToGo <6..0>
Only available on firmware versions 1.01 and up: TimeToGo contains an extra bit which represents a negative time-to-go. A negative time-to-go means that the time to go till the battery needs to be charged is infinite because the mean current is positive (battery is being charged). On the monitor this is displayed as four dashes: ----h.
Databyte1<2>: Negative TimeToGo

0x66 Temperature – Battery temperature.
Transmission repetition rate: 1s
Direction: output
Data Length: 3 bytes
Resolution: 1 / 256°C / bit gain, 0°C offset
Signed: no
Data range: 0°C – 50.000°C
Data type: measured
Databyte1<1..0> : Temperature <15..14>
Databyte2<6..0> : Temperature <13..7>
Databyte3<6..0> : Temperature <6..0>

0x67 MonitorStatus – Status of the battery monitor.
Transmission repetition rate: 1s
Direction: output
Data Length: 3 bytes
Resolution: ---
Signed: ---
Data range: ---
Data type: status
Databyte1<6> : reserved
Databyte1<5> : reserved
Databyte1<4> : ChargedVoltage – Voltage > ChargedVoltage level
Databyte1<3> : ChargedCurrent – Current < ChargedCurrent level
Databyte1<2> : reserved
Databyte1<1> : reserved
Databyte1<0> : AlarmTest – Alarmswitch triggered due to sys-com command
'Alarm switch on'
Databyte2<6> : BacklightTest – Backlight switched on due to sys-ex command
'Backlight on'
Databyte2<5> : DisplayTest – Display test switched on due to sys-ex command
'Display test on'
Databyte2<4> : NoTemperatureSensor – No temperature sensor detected by battery monitor
Databyte2<3> : SetupMode – Battery monitor is in setup-mode
Databyte2<2> : HistoryMode – Battery monitor is in history mode
Databyte2<1> : SuperLock – Super-lock is enabled
Databyte2<0> : OverVoltage – Overvoltage alarm is triggered
Databyte3<6> : UnderVoltage – Undervoltage alarm is triggered
Databyte3<5> : BatteryLow – The discharge floor is reached, battery needs to be charged
Databyte3<4> : BatteryFlat – Battery is fully discharged (0.0%)
Databyte3<3> : BatteryFull – Battery is fully charged (100.0%)
Databyte3<2> : ChargeBattery – Battery needs to be charged
Databyte3<1> : MonitorOutOfSync – Monitor is not in sync with battery, charge battery
Databyte3<0> : MonitorReset – Monitor has been reset due to a power-loss

0x70 ParameterSelect – The parameter displayed on the battery monitor.
Transmission repetition rate: on request
Direction: output / input
Data Length: 2 bytes
Resolution: ---
Signed: no
Data range: 0 - 6
Data type: status
Databyte1<0>: ParameterSelect<7>
Databyte2<6..0>: ParameterSelect<6..0>

0x71 FunctionDump – Dump of all the settings of the battery monitor.
Transmission repetition rate: on request / on change
Direction: output / input
Data Length: 24 bytes
Resolution: see format specification in section 2.2.6
Signed: see format specification in section 2.2.6
Data range: see format specification in section 2.2.6
Data type: status

0x72 HistoryDump – Dump of all the history events.
Transmission repetition rate: on request
Direction: output
Data Length: 25 bytes
Resolution: see format specification in section 2.2.7
Signed: see format specification in section 2.2.7
Data range: see format specification in section 2.2.7
Data type: status

0x78 CalibrationCoef1 – Used for voltage calibration, do not change this value.
Transmission repetition rate: on request
Direction: output / input
Data Length: 3 bytes
Resolution: ---
Signed: no
Data range: 0 – 65535
Data type: calibrated
Databyte1<6..0> : CalibrationCoef1<20..14>
Databyte2<6..0> : CalibrationCoef1<13..7>
Databyte3<6..0> : CalibrationCoef1<6..0>

0x79 CalibrationCoef2 – Used for current calibration, do not change this value.
Transmission repetition rate: on request
Direction: output / input
Data Length: 3 bytes
Resolution: ---
Signed: no
Data range: 0 – 65535
Data type: calibrated
Databyte1<6..0> : CalibrationCoef2<20..14>
Databyte2<6..0> : CalibrationCoef2<13..7>
Databyte3<6..0> : CalibrationCoef2<6..0>

0x7A CalibrationCoef3 – Used for current calibration, do not change this value.
Transmission repetition rate: on request
Direction: output / input
Data Length: 3 bytes
Resolution: ---
Signed: no
Data range: 0 – 65535
Data type: calibrated
Databyte1<6..0> : CalibrationCoef3<20..14>
Databyte2<6..0> : CalibrationCoef3<13..7>
Databyte3<6..0> : CalibrationCoef3<6..0>

0x7B CalibrationCoef4 – Used for current calibration, do not change this value.
Transmission repetition rate: on request
Direction: output / input
Data Length: 3 bytes
Resolution: ---
Signed: no
Data range: 0 – 65535
Data type: calibrated
Databyte1<6..0> : CalibrationCoef4<20..14>
Databyte2<6..0> : CalibrationCoef4<13..7>
Databyte3<6..0> : CalibrationCoef4<6..0>

0x7C CalibrationCoef5 – Used for temperature calibration, do not change this value.
Transmission repetition rate: on request
Direction: output / input
Data Length: 3 bytes
Resolution: ---
Signed: no
Data range: 0 – 65535
Data type: calibrated
Databyte1<6..0> : CalibrationCoef5<20..14>
Databyte2<6..0> : CalibrationCoef5<13..7>
Databyte3<6..0> : CalibrationCoef5<6..0>
0x7D CalibrationCoef6 – Used for temperature calibration, do not change this value.
Transmission repetition rate: on request
Direction: output / input
Data Length: 3 bytes
Resolution: ---
Signed: no
Data range: 0 – 65535
Data type: calibrated
Databyte1<6..0> : CalibrationCoef6<20..14>
Databyte2<6..0> : CalibrationCoef6<13..7>
Databyte3<6..0> : CalibrationCoef6<6..0>

0x7F FirmwareVersion – Firmware version of the battery monitor.
Transmission repetition rate: on request
Direction: output / input
Data Length: 2 bytes
Resolution: 0.01 / bit gain, 0 offset
Signed: no
Data range: 1.00 – 163.84
Data type: status
Databyte1<6..0> : FirmwareVersion<13..7>
Databyte2<6..0> : FirmwareVersion<6..0>

2.2.6 FunctionDump format specification

The data field in a function-dump is a chunk of 24 bytes containing all the functions (settings) of the XBM. This data needs some additional processing to actually represent the XBM settings. Some functions need an offset while others must be multiplied by a constant or variable factor.

F01 Battery capacity – Battery capacity (Ah).
Direction: output / input
Data Length: 2 bytes
Resolution: 1Ah / bit gain, 20Ah offset
Signed: no
Data range: 20 – 2000Ah
Data type: status
Databyte1<6..0> : Function1<13..7>
Databyte2<6..0> : Function1<6..0>
F01 = ((Databyte1 * 0x80) + Databyte2) + 20

F02 Charged Voltage – Voltage charged parameter (V).
Direction: output / input
Data Length: 1 byte
Resolution: 0.1V / bit gain, 8.0V offset
Signed: no
Data range: 8.0 – 33.0V
Data type: status
Databyte3<6..0> : Function2<13..7>
Databyte4<6..0> : Function2<6..0>
F02 = (((Databyte3 * 0x80) + Databyte4) * F16 * 0.1) + 8.0

F03 Charged Current – Current charged parameter (% of bat.cap.).
Direction: output / input
Data Length: 1 byte
Resolution: 0.5% / bit gain, 0.5% offset
Signed: no
Data range: 0.5 – 10.0%
Data type: status
Databyte5<6..0> : Function3<6..0>
F03 = (Databyte5 + 1) * 0.5

F04 Charged Time – Charged parameter time (minutes).
Direction: output / input
Data Length: 1 byte
Resolution: 1m / bit gain, 1m offset
Signed: no
Data range: 1 – 4m
Data type: status
Databyte6<6..0> : Function4<6..0>
F04 = Databyte6 + 1

F05 Low Battery On – Low Battery alarm on / discharge floor (%).
Direction: output / input
Data Length: 1 byte
Resolution: 1% / bit gain, 0% offset
Signed: no
Data range: 0 – 99%
Data type: status
Databyte7<6..0> : Function5<6..0>
F05 = Databyte7

F06 Low Battery Off – Low Battery alarm off (%).
Direction: output / input
Data Length: 1 byte
Resolution: 1% / bit gain, 1% offset
Signed: no
Data range: 1 – 100% / FULL
Data type: status
Databyte8<6..0> : Function6<6..0>
if Databyte8 = 100
then F06 = ‘FULL’
else F06 = Databyte8 + 1

F07 UnderVoltage Alarm – Under-voltage alarm (V).
Direction: output / input
Data Length: 2 bytes
Resolution: 0.1V / bit gain, 7.9V offset
Signed: no
Data range: OFF / 8.0 – 33.0V
Data type: status
Databyte9<6..0> : Function7<13..7>
Databyte10<6..0> : Function7<6..0>
if ((Databyte9 * 0x80) + Databyte10) = 0
then F07 = ‘OFF’
else F07 = (((Databyte9 * 0x80) + Databyte10)-1 * F16 * 0.1) + 8.0

F08 OverVoltage Alarm – Over-voltage alarm (V).
Direction: output / input
Data Length: 2 bytes
Signed: no
Resolution: 0.1V / bit gain, 9.9V offset
Data range: OFF / 10.0 – 35.0V
Data type: status
Databyte11<6..0> : Function8<13..7>
Databyte12<6..0> : Function8<6..0>
if ((Databyte11 * 0x80) + Databyte12) = 0
then F08 = 'OFF'
else F08 = (((Databyte11 * 0x80) + Databyte12) * F16 * 0.1) + 10.0

F09 Charge Efficiency – Charge efficiency factor / CEF (%).
Direction: output / input
Data Length: 1 byte
Resolution: 1% / bit gain, 50% offset
Signed: no
Data range: 50 – 99% / AU / A90
Data type: status
Databyte13<6..0> : Function9<6..0>
If Databyte13 = 50
then F09 = 'AU'
else if Databyte13 = 51
then F09 = 'A90'
else F09 = Databyte13 + 50

F10 Peukert Exponent – Peukert exponent.
Direction: output / input
Data Length: 1 byte
Resolution: 0.01 / bit gain, 1.00 offset
Signed: no
Data range: 1.00 – 1.50
Data type: status
Databyte14<6..0> : Function10<6..0>
F10 = Databyte14 + 1.00

F11 Battery Temperature – Battery temperature (°C).
Direction: output / input
Data Length: 1 byte
Resolution: 1°C / bit gain, 0°C offset
Signed: no
Data range: 0 – 50°C / AU
Data type: status
Databyte15<6..0> : Function11<6..0>
if Databyte15 = 51
then F11 = 'AU'
else F11 = Databyte15

F12 Temperature Coefficient– Battery temperature coefficient (%cap/°C).
Direction: output / input
Data Length: 1 byte
Resolution : 0.05%cap/°C / bit gain, 0.00%cap/°C offset
Signed: no
Data range: OFF / 0.05 – 0.95%cap/°C
Data type: status
Databyte16<6..0> : Function12<6..0>
if DataByte16 = 0
then F12 = ‘OFF’
else F12 = DataByte16 * 0.05

F13 TimeToGo Averaging Period – Time-to-go averaging period (minutes).
Direction: output / input
Data Length: 1 byte
Resolution: 3m / bit gain, 0m offset
Signed: no
Data range: 0 – 12m
Data type: status
Databyte17<6..0> : Function13<6..0>
F13 = DataByte17 * 3

F14 Current Threshold – Current threshold (A).
Direction: output / input
Data Length: 1 byte
Resolution: 0.1A / bit gain, 0.0A offset
Signed: no
Data range: 0.0 – 2.0A
Data type: status
Databyte18<6..0> : Function14<6..0>
F14 = DataByte18 * 0.1

F15 Temperature Unit Selection – Temperature unit selection. Only available on firmware versions
1.10 and newer!
Direction: output / input
Data Length: 1 byte
Resolution: ---
Signed: no
Data range: °C / °F
Data type: status
Databyte19<6..0> : Function15<6..0>
if DataByte19
then F15 = ‘°C’
else F15 = ‘°F’

F16 Voltage Prescaler – Voltage prescaler.
Direction: output / input
Data Length: 1 byte
Resolution: ---
Signed: no
Data range: 1:1 / 1:5 / 1:10
Data type: status
Databyte20<6..0> : Function16<6..0>
if DataByte20 = 0
then F16 = 1
else if DataByte20 = 1
then F16 = 5
else F16 = 10

F17 Display Mode – Display backlight mode (s).
Direction: output / input
Data Length: 1 byte
Resolution: 10s / bit gain, 0s offset
Signed: no
Data range: OFF / 10 – 60 / ON / AU
Data type: status
Databyte21<6..0> : Function17<6..0>
if DataByte21 = 0
then F17 = ‘OFF’
else if DataByte21 = 7
then F17 = ‘ON’
else if DataByte21 = 8
then F17 = ‘AU’
else F17 = DataByte21 * 10

F18 Alarm Contact Polarity – Alarm contact polarity. Only available on firmware versions 1.01 and newer!
Direction: output / input
Data Length: 1 byte
Resolution: ---
Signed: no
Data range: NO / NC
Data type: status
Databyte22<6..0> : Function18<6..0>
if DataByte22 = 0
then F18 = ‘NO’
else F18 = ‘NC’

F19 Display Parameter – Display parameter.
This is a special (sort of hidden) function. On the XBM itself, reading out F19 will give you the battery monitor's firmware version. Since the firmware version can be retrieved thru its own message via TBS-Link, the necessity of retrieving it via the function dump becomes obsolete. Therefore F19 is used to enable / disable the readout of each individual parameter in normal operation of the XBM. When a parameter readout is enabled, its corresponding bit in Databyte23 is set otherwise it is cleared. Note that this function only affects the readout on the XBM itself, a disabled parameter can always be retrieved via TBS-Link.
Direction: output / input
Data Length: 1 byte
Resolution: ---
Signed: no
Data range: ---
Data type: status
Databyte23<0>: Enable / disable Voltage readout
Databyte23<1>: Enable / disable Current readout
Databyte23<2>: Enable / disable Amphours readout
Databyte23<4>: Enable / disable StateOfCharge readout
Databyte23<5>: Enable / disable TimeToGo readout
Databyte23<6>: Enable / disable Temperature readout
F19 = DataByte23

F20 Setup Lock – Setup lock.
Direction: output / input
Data Length: 1 byte
Resolution: ---
Signed: no
Data range: OFF / ON
Data type: status
Databyte24<6..0> : Function20<6..0>
if DataByte24 = 0
then F20 = 'OFF'
else F20 = 'ON'

2.2.7 HistoryDump format specification

The data field in a history-dump is a chunk of 25 bytes containing all the history data of the XBM.

H01 Charge Efficiency – Automatically calculated Charge Efficiency Factor (%).
Direction: output
Data Length: 3 bytes
Resolution: 0.1 / 65536 % / bit gain, 0.0% offset
Signed: no
Data range: 0.0 – 100.0%
Data type: calculated
Databyte1<1..0> : History1<15..14>
Databyte2<6..0> : History1<13..7>
Databyte3<6..0> : History1<6..0>

H02 Average Discharge (Ah) – Average discharge in Amphours (Ah).
Direction: output
Data Length: 3 bytes
Resolution: 0.1Ah / bit gain, 0.0Ah offset
Signed: yes (always negative)
Data range: 0Ah – 1999.9Ah
Data type: calculated
Databyte4<1..0> : History2<15..14>
Databyte5<6..0> : History2<13..7>
Databyte6<6..0> : History2<6..0>

H03 Deepest Discharge (Ah) – Deepest discharge in Amphours (Ah).
Direction: output
Data Length: 3 bytes
Resolution: 0.1Ah / bit gain, 0.0Ah offset
Signed: yes (always negative)
Data range: 0Ah – 1999.9Ah
Data type: calculated
Databyte7<1..0> : History3<15..14>
Databyte8<6..0> : History3<13..7>
Databyte9<6..0> : History3<6..0>

H04 Charge Discharge Cycles – Number of charge-discharge cycles (#).
Direction: output
Data Length: 2 bytes
Resolution: 1 / bit gain, 0 offset
Signed: no
Data range: 0 – 9999
Data type: calculated
Databyte10<6..0> : History4<13..7>
Databyte11<6..0> : History4<6..0>

H05 Synchronizations – Number of synchronizations (#).
Direction: output
Data Length: 2 bytes
Resolution: 1 / bit gain, 0 offset
Signed: no
Data range: 0 – 9999
Data type: calculated
Databyte12<6..0> : History5<13..7>
Databyte13<6..0> : History5<6..0>

H06 Full Discharges – Number of full discharges (#).
Direction: output
Data Length: 2 bytes
Resolution: 1 / bit gain, 0 offset
Signed: no
Data range: 0 – 9999
Data type: calculated
Databyte14<6..0> : History6<13..7>
Databyte15<6..0> : History6<6..0>

H07 UnderVoltage triggers – Number of under-voltage triggers (#).
Direction: output
Data Length: 2 bytes
Resolution: 1 / bit gain, 0 offset
Signed: no
Data range: 0 – 9999
Data type: calculated
Databyte16<6..0> : History7<13..7>
Databyte17<6..0> : History7<6..0>

H08 OverVoltage triggers – Number of over-voltage triggers (#).
Direction: output
Data Length: 2 bytes
Resolution: 1 / bit gain, 0 offset
Signed: no
Data range: 0 – 9999
Data type: calculated
Databyte18<6..0> : History8<13..7>
Databyte19<6..0> : History8<6..0>

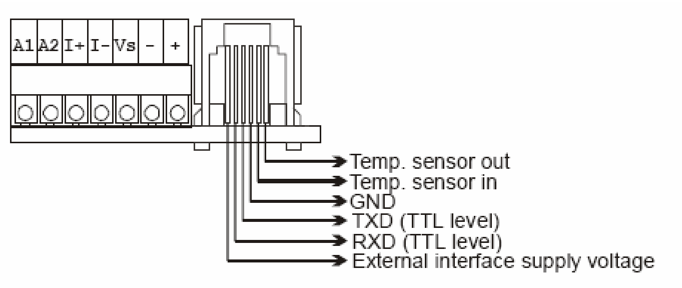
H09 Average Discharge (%) – Average discharge in percent state-of-charge (%).Only available on firmware versions 1.03 and newer!
Direction: output
Data Length: 3 bytes
Resolution: 0.1% / bit gain, 0.0% offset
Signed: yes (always negative)
Data range: 0.0 – 100.0%
Data type: calculated
Databyte20<1..0> : History9<15..14>
Databyte21<6..0> : History9<13..7>
Databyte22<6..0> : History9<6..0>

H10 Deepest Discharge (%) – Deepest discharge in percent state-of-charge (%). Only available on firmware versions 1.03 and newer!
Direction: output
Data Length: 3 bytes
Resolution: 0.1% / bit gain, 0.0% offset
Signed: yes (always negative)
Data range: 0.0 – 100.0%
Data type: calculated
Databyte23<1..0> : History10<15..14>
Databyte24<6..0> : History10<13..7>
Databyte25<6..0> : History10<6..0>

3 PHYSICAL LAYER

The XBM outputs its data a-synchronously at 2400bps using 8 data bits, 1 stop bit and even parity. Pin 2 and 3 of the expansion port are TTL-level receive and transmit ports of the battery monitor. Use the optionally available isolated RS-232 interface to transform these TTL-levels to RS-232 levels. The input to output isolation barrier of this interface is >1000V, so the battery monitor can be safely connected to every PC or notebook. Note that the isolated RS-232 interface draws a current of 16 to 30mA from your battery.

The pin out of the rear side RJ12 expansion port is as follows:



4 PRACTICAL EXAMPLES

The XBM can operate in one of two communication modes.

Automatic mode: the battery monitor outputs all measured and calculated parameters automatically once second and transmits data when pressing a key and when leaving setup mode. This is the standard mode for every XBM leaving the factory.

Request only mode: the battery monitor only sends data when receiving a request for it (through a sys-ex data request, see chapter 2.2.4). Transmitting data when pressing a key and when leaving setup-mode is also disabled this mode.

Regardless of the selected communication mode, the monitor will send the firmware version on power up. This is necessary for compatibility with the XBM monitoring software.

Note: The request only mode is only available on firmware versions 1.01 and newer. Therefore only chapter 4.1 is of interest for users communicating with an XBM containing firmware version 1.00.

Note : It is possible that, when in request only mode and the supplyvoltage of the monitor (and the communication interface) is decreasing slowly to a level outside the RS-232 margins, the monitor sends a NACK+Repeat Request due to a communication error.

4.1 Automatic mode

In this chapter we’ll discuss the communication sequence of an XBM in automatic mode.
After powering up the battery monitor it immediately starts sending (broadcasting). You don't need to send any command to enable the transmission of data. It will first send the firmware version once and then starts sending the voltage, current, amphours, state-of-charge, time-to-go, temperature and monitor status once a second. So after power-up you will get the data as follows:

```
// power up

// few hundred milliseconds delay


0x80 0x00 0x20 0x7F 0xf1 0xf2 0xFF (fx = firmware version)

// one second delay

0x80 0x00 0x20 0x60 0xu1 0xu2 0xu3 0xFF     (ux = voltage)
0x80 0x00 0x20 0x61 0xi1 0xi2 0xi3 0xFF     (ix = current)
0x80 0x00 0x20 0x62 0xa1 0xa2 0xa3 0xFF     (ax = amphours)
0x80 0x00 0x20 0x64 0x%1 0x%2 0x%3 0xFF     (%x = state-of-charge)
0x80 0x00 0x20 0x65 0xh1 0xh2 0xh3 0xFF     (hx = time-to-go)
0x80 0x00 0x20 0x66 0xt1 0xt2 0xt3 0xFF     (tx = temperature)
0x80 0x00 0x20 0x67 0xm1 0xm2 0xm3 0xFF     (mx = monitor status)

// one second delay
0x80 0x00 0x20 0x60 0xu1 0xu2 0xu3 0xFF    (ux = voltage)
0x80 0x00 0x20 0x61 0xi1 0xi2 0xi3 0xFF    (ix = current)
0x80 0x00 0x20 0x62 0xa1 0xa2 0xa3 0xFF    (ax = amphours)
0x80 0x00 0x20 0x64 etc.....
```

Now let's look at the voltage output a bit closer:

The battery monitor doesn’t output the data in ASCII values; it's a bit more complex. The outputted string looks like this:

```
0x80 0x00 0x20 0x60 0xu1 0xu2 0xu3 0xFF (ux = voltage)
```

The bytes u1, u2 and u3 represent the voltage data as follows: because the MSB of all the bytes between the header (0x80) and the e.o.t. trailer (0xFF) always represent a logical 0, we have 7 bits per byte left to represent the data. So with the three data-bytes u1, u2 and u3 we can transmit a 21-bit number. These three bytes hold the voltage data as follows:

```
u1<6..0> = voltage<20..14>
u2<6..0> = voltage<13..7>
u3<6..0> = voltage<6..0>
```

As you can see bit6 of u1 (the 1st data byte in the string) holds the MSB of the 21-bit number and bit0 of u3 (the 3rd and last data byte in the string) holds the LSB of the 21-bit number. Let's assume the battery monitor outputs:

```
0x80 0x00 0x20 0x60 0x00 0x09 0x11 0xFF
```

This would represent a voltage of:

(0x00 << 14) + (0x09 << 7) + 0x11

=

0x000000 + 0x000480 + 0x000011

=

0x000491 = 1169d

11.69Volts.

Note : the << operator is a C / C++ ‘bitwise shift left’ operator. The operation :

X << Y

shifts the value of X to the left by Y bits. An example :

0x0009 = 0000 0000 0000 1001b

0x0009 << 7 = 0000 0100 1000 0000b = 0x0480

The Turbo Pascal / Delphi equivalent of the << operator is the shl operator. The same result can be achieved by multiplying :

X << 7 = X * 0x80

X << 14 = X * 0x4000

X << 21 = X * 0x200000

The same rule applies for every outputted parameter. With current and amphours we have to deal with a sign. The format of the current (see also chapter 2.2.5) data looks like this:

i1<2> = sign (0 = positive current, 1 = negative current)
i1<1..0> = current<15..14>
i2<6..0> = current<13..7>
i3<6..0> = current<6..0>

Current<15..0> is not in 2's complement it's just a positive number ranging from 0..65535 representing a current of 0.00Amps to 655.35Amps. So when the battery monitor outputs:

0x80 0x00 0x20 0x61 0x04 0x47 0x1E 0xFF

it represents a current of :

(0x04 << 14) + (0x47 << 7) + 0x1E = 0x01239E

where bits<15..0> of the result hold 0x239E = 9118d which represents a current of 91.18Amps, bit<16> of the result = 1 making the current a negative current (flowing out of the battery). So:

0x80 0x00 0x20 0x61 0x04 0x47 0x1E 0xFF = -91.18Amps

The same trick works for the amphours output only this parameter has a 1 decimal resolution. A decimal value of -793 represents -79.3Amphours.

The format of the state-of-charge output is basically the same as that of the voltage output only that the state-of-charge output is hardware limited to 0x3E8 = 1000d which represents 100.0%.

When we take a closer look at the time-to-go format in chapter 2.2.5, we see that this is in a hhh:mm format when we look at it decimally. This means that the two least significant decimal digits represent the number of minutes left and the rest of the decimal digits represent the number hours left.
If for example the monitor outputs :

```
0x80 0x00 0x20 0x65 0x00 0x0B 0x2C 0xFF
```

then shifting 0x0B left 7 positions and adding 0x2C would make 0x5AC which decimally represents 1452d (or 01452d). This number represents 14 hours and 52 minutes to go. The range is 0:00 to 240:00. So 24000d represents 240 hours and 0 minutes.

Here's a C coded example of getting the hours and minutes (where d1, d2, and d3 are the three received data bytes):

```
unsigned int TimeToGo, Hours, Minutes;
TimeToGo = (d1 << 14) + (d2 << 7) + d3;
Hours = TimeToGo / 100;
Minutes = TimeTo % 100;
printf("The time-to-go is %d hours and %d minutes", Hours, Minutes);
```

A bit different is the temperature output. After combining the received data bytes, the upper byte represents the temperature in degrees Celsius while the lower byte represents fractional part of the temperature in 1/256°C accuracy. Not that this is the accuracy of the temperature sensor but this is used for calibrating the zero-offset of the temperature circuit in the monitor at the factory. You can use this fractional part for rounding the temperature value. An example temperature output:

```
0x80 0x00 0x20 0x66 0x00 0x2D 0x40 0xFF
```

Now shifting 0x2D left 7 bits and adding 0x40 to it would make 0x16C0 which represents 5824d. If we then divide this number by 256d we get 22.75°C which we must round to get 23°C.

Note: You cannot use d2 (the second data byte) directly to read the temperature! You must do the shifts first, because the MSB of the lower temperature byte is the LSB of the upper temperature byte.

4.2 Request only mode

In this chapter we'll discuss the communication sequence of an XBM in request only mode.
Please note that the request only mode is only available on firmware versions 1.01 and up.
When the monitor is put in "Request only mode" you can retrieve the data with the "All parameters request" command (messagetype 0x4F). It is strongly advised NOT to retrieve the data by sending a request for each individual parameter sequentially. The outputted data in request only mode has the same format as in automatic mode.

UNLESS SPECIFICALLY AGREED TO IN WRITING, XANTREX TECHNOLOGY INC. ("XANTREX"):

(a) MAKES NO WARRANTY AS TO THE ACCURACY, SUFFICIENCY OR SUITABILITY OF ANY TECHNICAL OR OTHER INFORMATION PROVIDED IN ITS MANUALS OR OTHER DOCUMENTATION.

(b) ASSUMES NO RESPONSIBILITY OR LIABILITY FOR LOSS OR DAMAGE, WHETHER DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL, WHICH MIGHT ARISE OUT OF THE USE OF SUCH INFORMATION. THE USE OF ANY SUCH INFORMATION WILL BE ENTIRELY AT THE USER'S RISK.

Part number: **976-0133-01-01 Rev A**

Contact information:

Phone:	1-800-670-0707 (toll-free in North America)
Phone:	1-360-925-5097 (outside North America)
Fax:	1-360-925-5143
Email:	customerservice@xantrex.com
Web:	www.xantrex.com