BTMGlobal Modbus Interface
Register Mapping

Ver. 2.17

2013-10-05

Table of Contents

1. Introduction.....3
2. Supported Data Types:.....4
3. Exception Codes.....5
4. Supported Standard Functions:.....6
4.1. Read Coil Status (Function 1).....6
4.2. Read Input Status (Function 2).....6
4.3. Read Holding Registers (Function 3).....6
4.4. Read Input Registers (Function 4).....7
1.1.1. Jar voltage.....10
1.1.2. Jar temperature (if monitored).....10
4.5. Preset Holding Registers(Function 6) and Preset Multiple Holding Registers (Function 16).....10
Appendix A.....12

1. Introduction.

This document is intended to give a definition for registers available in Polytronics BTM Global Monitoring System, using Modbus Protocol.

NOTES.

The maximum number of registers for multiple register read is dependent on the particular device. If request frame specifies higher number of registers to read, RESPONCE LENGTH TOO LONG exception frame will be generated.
BtmGlobal/Modbus Translator – maximum allowed by Modbus Protocol

BtmGlobal/Modbus Translator supports an RTU communication mode only.
The communication link default setup is defined as: 19200 baud, no parity, and 8 data bits. Other baud rates and odd, even parity are supported.

Jar Numbering is in physical sequential order - from the most positive post to the most negative post (battery #1 is the most positive).

BTM 32-bit long integer is represented as high order value in odd register number, low order value in even register number. The even register number (low order value) is listed in this document.

Data is represented in big-endian format (MS Byte at lower address).

Please refer Modicon Modbus Protocol Reference Guide (PI-MBUS-300, Rev.J) or derivative document for detailed description of the Modbus and Open Modbus protocols.
(**http://www.eecs.umich.edu/~modbus/documents/PI_MBUS_300.pdf** and **http://www.modbus.org**)

In this document input registers are defined from 30001, holding registers are defined from 40001.

IMPORTANT.

When assembling a Modbus command frame, the register address denotes offset into either Input (Function 4) or Holding (Function 3, 6 and 16) register pool. Register description in this document starts from 1, while in command frame register address (offset) starts from 0.

2. Supported Data Types:

1. BTM 16-bit unsigned short integer - 1 word register data (0 ~ 65534), 0xFFFF – UNDEFINED VALUE

2. BTM 16-bit signed short integer - 1 word register data (-32767 ~ +32767), 0x8000 – UNDEFINED VALUE

3. BTM 32-bit unsigned long integer - Addressable register (eg 30001) = AR And AR+1. Contains 32-bit value: V = AR*10000H | (AR+1). This data type is mainly used for time-stamps. Time is expressed in seconds since 00:00:00 GMT, January 01, 1970 or since reference date, stored in the device configuration memory. 0 or 0xFFFFFFFF – UNDEFINED VALUE

4. BTM Floating point ONLY USED AS SPECIAL CASE - Floating Point is stored as follows: Addressable register (eg 30001) = AR And AR+1. Contains 32-bit value: AR+1 contains bits 15-0 of significant AR contains bits 31-16 (sign, exponent and bits 23-16 of significant). 0xFF800000 – UNDEFINED VALUE (negative infinity)

EXAMPLE: Registers 32016 contains 0x0000H and 32015 contains 0xBFC0) => 0x0000BFC0 = -1.5 (floating point). This means that String #3 discharging current is: - 1.5 A

3. Exception Codes

01 - ILLEGAL FUNCTION - function code received is not supported by BTM
02 - ILLEGAL DATA ADDRESS - data address is outside defined boundaries.
03 - ILLEGAL DATA VALUE - value to be written into holding registers is illegal.
04 - RESPONCE LENGTH TOO LONG - reading multiple registers will result response package which exceeds defined frame size limit. Requested function failed. Typically Write Register 40008, if BTM is not running.
06 - DEVICE BUSY - Device performing lengthy function, master should repeat query at later time

4. Supported Standard Functions:

4.1. Read Coil Status (Function 1)

General format 0x.

BTM does not have any defined 0X references, reading any register will return UNDEFINED VALUE (8000H)

4.2. Read Input Status (Function 2)

General format 1x.

BTM does not have any defined 1X references, reading any register will return UNDEFINED VALUE (8000H)

4.3. Read Holding Registers (Function 3)

General format 4x.

S – string number from 0 (first string) up to 9 (last string).One string occupies a block of 1000 registers.

NOTE: If more than 10 strings are configured per a single node, the node splits logically into multiple consecutive nodes.

EXAMPLE: Site with 24 strings and Modbus translator which ID = 10, will result 3 logical Modbus nodes: 10, 11 and 12 for strings 1 – 10 (node ID = 10), 11-20 (node ID = 11) and 21-24 (node ID = 12).

40002 – Total Number of strings configured per this system (unsigned short)
40003 – System Time in seconds (unsigned long)
40009 - Active alarm acknowledgement (bit mapped) (unsigned short)
This register contains :

- **0** - no alarm contact is active
- **1** - BtmGlobal internal communication loop failure
- **2** - BtmGlobal battery alarm active

Writing 0 to this register will acknowledge active alarm and release the contact.

4s011 - Active data type currently available in registers 4s100 and above(unsigned short).
Data Type can be one of following:

1 - Open Jar voltages
2 - AC Ripple voltages (default type)
3 - Floating voltage mobility values
4 - Charge Acceptance Index values
5 - Normalized Internal Resistance values
6 - Discharge Index values
7 - Discharge End Voltage values
10- Post Temperature values (optional, special system)
11- Inter cell strap resistance values (optional, special system)

Not all types are supported by a particular system.

4s100 - Overall average stored data value for all jars (as specified by 4s011),

Jar Numbering is in physical sequential order - from the most positive post to the most negative post (battery #1 is the most positive).

4s101 – Jar#1 stored data (as specified by 4s011), most positive jar.
4s102 – Jar#2 stored data (as specified by 4s011)
…
4s399 – Jar#299 stored data (as specified by 4s011) - Maximum of 299 points could be monitored per one string

NOTE: If particular stored data type is not available, (system does not support its measurements (eg. inter cell resistances), or condition has never been met, (eg. open voltages), UNDEFINED_VALUE will be returned for all jars.

4.4. Read Input Registers (Function 4)

30002 – Average System scan time (in 0.1 sec per bit) (unsigned short)

Helper /debugging register, contains duration information about the Modbus Translator internal scan cycle. Modbus Controller should scan input registers within time interval equal or greater than content of this register.

EXAMPLE: Register 0002 reads 33. This means that any modbus controller for string data should read Input registers every 3.3 second or later.

30009 - System status / system active alarm (bit mapped) (unsigned short). This register may contain the following: (several bit-mapped alarm conditions can be or-ed together)

0000H: normal (system is up and running)
0001H: internal Communication Loop Failure (fiber-optical loop problem)
0002H: concentrator off-line ()
0004H:
0010H: BTM port busy
0020H: Modbus port busy
0040H: TCP socket failure
0100H: low resources (internal data storage space exhausted)
4FFFH: monitoring function off line

S – string number from 0 (first string) up to 9 (last string).

One string occupies a block of 1000 registers.

EXAMPLE: Register 30019 contains status information for string number 1, register 32019 contains status information for string number 3.

NOTE: If more than 10 strings are configured per single Modbus translator node, the node splits logically into multiple consecutive nodes.

EXAMPLE: Site with 24 strings and Modbus translator which ID = 10, will result 3 logical Modbus nodes: 10, 11 and 12 for strings 1 – 10 (node ID = 10), 11-20 (node ID = 1) and 21-24 (node ID = 12).

3s011 - String Used Capacity (32-bit signed long, 1 bit = 1 A*sec), negative value indicates energy removed from the string, positive value indicate overcharge
3s013 - String Rated Capacity (32-bit signed long, 1 bit = 1 A*sec) During discharge, this value indicates what is the available capacity (based on the load current and ambient temperature), during any other event this is the factory rating value.
3s015 – Event duration (unsigned long, 1 bit = 1 sec). Duration of current string event (status) in seconds. The timer value can be constructed as follows: Value = (unsigned long)registerS016 * 0x10000L | ((unsigned long)registerS015 & 0x0FFFFL)
3s017 – String pending alarm (bit mapped) (unsigned short)

Critical Alarms: Immediate Action Required
0001H: String Exhausted, battery capacity drained, pending load loss

0002H: String Open, possible charger malfunction, breaker open, possibly no support to load

0004H: Pending Jar/Cell reversal during discharge, (jar terminal potential is below specified discharge end voltage, danger of jar explosion (VLR) or boiling (flooded)

0008H: Unexpected Equalization, potential damage of battery

Notification of Abnormal/Changed Condition(s) on the battery, some action required during regular maintenance.

0010H: Detected reduced jar capacity (during discharge event)

0020H: Detected high internal jar resistance (during discharge event)

0040H: Jar floating voltage below set limit (during float event)

0080H: Jar floating voltage above set limit (during float event)

0100H: Jar capability to accept charge reduced (after charge)

0200H: Jar float voltage mobility increased (during float event)

0400H: Jar voltage input noise increased

0800H: Jar calculated parameters degraded

Warning on Changes in battery status (Notification, no action required)

1000H: discharge warning, load is on the battery

2000H: equalization warning, battery is being equalized

4000H: charge warning, battery is being charged

3s018 – String status (event) (unsigned short)

This register contains one of the following values:

0: UNKNOWN, unknown status (typically error or system start up)

1: OPEN, battery string at Open condition (charger not connected or malfunctioning, main breaker open)

2 : FLOATING, battery string floating (normal)

3 : FLOAT-CHARGING, battery recharge with small current (typically after charge) for prolonged time period

4 : CHARGING, battery recharge after discharge

5: EQUALIZING, battery at equalization potential (Equalizing), typically after recharge

6: DISCHARGING, battery string discharging, load is on battery (most important event). This register should be read as frequently as possible (but not sooner than register 0002/10 seconds).

3s019 - Average String Current (32-bit signed long, 1 bit = 1 mA). This register contains value of measured string current (negative current is discharging current)

3s021 - Average String Voltage (16-bit unsigned short, 1 bit = 0.01 V). This register contains value of total string voltage (unsigned entity) in 10*milliVolts., i.e. 10000 = 100.00 V

3s022 - Average AC Ripple Current (16-bit unsigned short, 1 bit = 0.1 Arms). This register contains value of measured ac ripple current. The value is available only if BTM-Factstar is installed and configured for this string.

3s023 - Average AC Ripple Voltage, (16-bit unsigned short, 1 bit = 0.001V)

3s024 - Average Ambient T in C degrees, (16-bit signed short, 1 bit = 1/10 degree C)

3s029 - time remaining during discharge only (unsigned long, 1 bit = 1 sec). Jar Numbering is in physical sequential order - from the most positive post to the most negative post (battery #1 is the most positive).

1.1.1. Jar voltage

3s101 – Jar#1 present voltage, most positive jar. If individual jar monitoring is not implemented, returns UNDEFINED VALUE;

3s102 – Jar#2 present voltage

…

3s399 – Jar#299 present voltage - Maximum of 299 points could be

1.1.2. Jar temperature (if monitored)

Each register contains 16 bit signed integer representing temperature in degrees C, where one bit increment corresponds to 1/100 of a degree.

3s401 – Jar#1 present temperature, most positive jar. If individual jar temperature monitoring is not implemented, returns UNDEFINED VALUE;

3s402 – Jar#2 present temperature

…

3s699 – Jar#299 present temperature - Maximum of 299 points could be monitored

4.5. Preset Holding Registers(Function 6) and Preset Multiple Holding Registers (Function 16)

General format 4x.

NOTE. For registers, defined as long, it is recommended to use Function 16, rather than 2 consecutive functions 6.

Currently supported Preset functions for BtmGlobal:

40009 - Active alarm acknowledgement (unsigned short). Writing 0 to this register will acknowledge active alarm and release the contact.

4s011 - Stored data type currently available in registers 4s100 and above(unsigned short). Writing to this register will define which type of values are available in stored jar data block (regs 4s100 and above).

Type may be one of following:

1 - Open Jar voltages

2 - AC Ripple voltages (default type)

3 - Floating voltage mobility values

4 - Charge Acceptance Index values

5 - Normalized Internal Resistance values

6 - Discharge Index values

7 - Discharge End Voltage values

10- Post Temperature values (optional, special system)

11- Inter cell strap resistance values (optional, special system)

Appendix A.

Practical Samples for BTM Modbus Based Register Reading.

In the communication loop, there exists BTM Modbus Translator (slave device) which ID is 1.
At Startup Modbus Master reads system configuration.

First, total number of strings per node:

Query: (Read Holding Register 40002 from device 1, registers are addressed starting from zero).

Field Name	Content (Hex)
Slave Address	01
Function Code	03
Register Starting Address High	00
Register Starting Address Low	01
Number of Registers to Read High	00
Number of Registers to Read Low	01
CRC Field	--

Response:

Field Name	Content (Hex)
Slave Address	01
Function Code	03
Byte Count in this response	02
Data of register (30003) High	00
Data of register (30003) Low	08
CRC Field	--

Total number of strings per this node is 8.

Second, for every string s(s starts from 0), read total number of jars per string from: 4s010

Query: (Read Holding Register 40010 from device 1, registers are addressed starting from zero).

Field Name	Content (Hex)
Slave Address	01
Function Code	03
Register Starting Address High	00
Register Starting Address Low	09
Number of Registers to Read High	00
Number of Registers to Read Low	01
CRC Field	--

Response:

Field Name	Content (Hex)
Slave Address	01
Function Code	03
Byte Count in this response	02
Data of register (40050) High	00
Data of register (40050) Low	18
CRC Field	--

Number of jars per string #1 is 24.

Query: (Read Holding Register 41010 from device 1, registers are addressed starting from zero).

Field Name	Content (Hex)
Slave Address	01
Function Code	03
Register Starting	03

Address High	
Register Starting Address Low	F1
Number of Registers to Read High	00
Number of Registers to Read Low	01
CRC Field	--

Response:

Field Name	Content (Hex)
Slave Address	01
Function Code	03
Byte Count in this response	02
Data of register (41050) High	00
Data of register (41050) Low	14
CRC Field	--

Number of jars per string #2 is 20.

….

Regular scan cycle example:

For Every string (s) read Registers 3s018-3s021

Query: (Read Input Registers 30018-30021 from device 1, registers are addressed starting from zero).

Field Name	Content (Hex)
Slave Address	01
Function Code	04
Register Starting Address High	00
Register Starting Address Low	11
Number of Registers to Read High	00

Number of Registers to Read Low	04
CRC Field	--

Response:

Field Name	Content (Hex)
Slave Address	01
Function Code	04
Byte Count in this response	14
Data of register (30018) High	00
Data of register (30018) Low	40
Data of register (30019) High	00
Data of register (30019) Low	02
Data of register (30020) High	09
Data of register (30020) Low	C4
Data of register (30021) High	15
Data of register (30021) Low	4A
CRC Field	--

String #1 has Jar Below Low Limit Alarm (register 30018= 0040H)
String #1 is Floating (register 30019=0002H)
String #1 floating current is 2.5A (register 30020 = 09C4H)
String #1 voltage is 54.5V (register 30021 = 154AH)