

Description of M-Bus ASCII protocol

Author: Bertil Göransson
Revision: 2.00.04
Filename: Manual_Ascii_E_2_00_04.docx

Contents

1. DOCUMENT INFORMATION	3
1.1 VERSIONS	3
2. PROTOCOL DESCRIPTION	3
2.1 MESSAGE STRUCTURE	3
2.1.1 <i>Single item read message format without checksum:</i>	<i>3</i>
2.1.2 <i>Single item read message format with checksum:</i>	<i>4</i>
2.1.3 <i>Single item write message format without checksum: Implementing in V2.x</i>	<i>4</i>
2.1.4 <i>Single item write message format with checksum: Implementing in V2.x</i>	<i>4</i>
2.1.5 <i>Multiple item read message format without checksum:</i>	<i>4</i>
2.1.6 <i>Multiple item read message format with checksum:</i>	<i>4</i>
2.2 MESSAGE DESCRIPTIONS	5
2.2.1 <i>Field description</i>	<i>5</i>
2.2.2 <i>Control characters</i>	<i>5</i>
2.2.3 <i>Error description</i>	<i>6</i>
2.3 EXAMPLES	6
2.3.1 <i>Read request</i>	<i>6</i>
2.3.2 <i>Checksum calculation</i>	<i>6</i>
3. APPENDIX	7
3.1 MESSAGE STRUCTURE OBSOLETE	7
<i>Single item read message format:</i>	<i>7</i>

1. Document Information

The M-Bus ASCII protocol is developed by PiiGAB with the purpose of helping users to an easy way to read M-Bus meters without the need to have specific M-Bus drivers in the computer.

1.1 Versions

Version	Detail
0.00.00.001	The very first version
2.00.00.001	The first v2 version
2.00.01.001	Changed timeout error from T to T and M
2.00.02.001	Added error O for maximum number of read items
2.00.03	Several small adjustments

2. Protocol description

The core of the protocol is the OPC item which is built up from Channel, Device and Tag. If you have experience from OPC Servers this is probably well known. The reason for using this technique is that we have implemented the same dynamic M-Bus client in the PiiGAB 900 as we have in the PiiGAB M-Bus OPC Server. The positive consequence of this is that you can use the same configuration tool as for the OPC Server. You simply take the configuration file and upload it to the PI-900 instead of using it together with the OPC Server.

This also means that you can test your configuration file with the monitor function using the demo version of the OPC Server before you upload the file to the PI-900.

The only thing needed from the M-Bus ASCII client is the name of your tags which is the same as OPC Item. See example below. For the M-Bus ASCII protocol all tags must have Datatype String.

2.1 Message Structure

The communication structure is a typical Request/Response message structure. You can choose if you want to have a checksum control for each message. The checksum should normally be used if you are using RS232 or RS485. If you are using TCP or UDP a checksum is already a part of a higher communication layer and is therefore not needed although it is allowed

The messages buildup is <start byte><text area><stop byte> where the start byte is one of STX, ACK or NAK and the stop byte is always ETX.

The text area consists of only readable tokens. All numbers like TID, PID, ADR and CHECKSUM are formatted in a special way. All are eight bit binary numbers formatted as two characters. Examples: Decimal 127 = 0x7F => "7F". The M-Bus ASCII is obviously optimized to be readable, but it does not make good use of bandwidth.

The user must be careful when creating the checksum. In a normal binary based protocol the checksum of two bytes 0x7F and 0x81 is $(0x7F+0x81)\%0x100 = 0$. But in M-Bus ASCII we create the checksum by summing up the integer values of individual characters: $(\text{'7'}+\text{'F'}+\text{'8'}+\text{'1'})\%0x100 = (55 + 70 + 56 + 49)\%0x100 = 0xE6$. In this way "7f" generates a different checksum from "7F". This does not create any problem unless the case of a telegram is changed. We recommend using uppercase letters for numbers as well as OPC items.

2.1.1 Single item read message format without checksum:

Request:

STX	TID	PID	ADR	ITEM	ETX
-----	-----	-----	-----	------	-----

Positive answer:

ACK	TID	PID	ADR	DATA	ETX
-----	-----	-----	-----	------	-----

Negative answer:

NAK	TID	PID	ADR	ERROR	ETX
-----	-----	-----	-----	-------	-----

2.1.2 Single item read message format with checksum:

Request:

STX	TID	PID	ADR	ITEM	CRC	ETX
-----	-----	-----	-----	------	-----	-----

Positive answer:

ACK	TID	PID	ADR	DATA	CRC	ETX
-----	-----	-----	-----	------	-----	-----

Negative answer:

NAK	TID	PID	ADR	ERROR	ETX
-----	-----	-----	-----	-------	-----

2.1.3 Single item write message format without checksum: Implementing in V2.x

Request:

STX	TID	PID	ADR	ITEM DATA	ETX
-----	-----	-----	-----	-----------	-----

Positive answer:

ACK	TID	PID	ADR	ETX
-----	-----	-----	-----	-----

Negative answer:

NAK	TID	PID	ADR	ERROR	ETX
-----	-----	-----	-----	-------	-----

2.1.4 Single item write message format with checksum: Implementing in V2.x

Request:

STX	TID	PID	ADR	ITEM DATA	CRC	ETX
-----	-----	-----	-----	-----------	-----	-----

Positive answer:

ACK	TID	PID	ADR	CRC	ETX
-----	-----	-----	-----	-----	-----

Negative answer:

NAK	TID	PID	ADR	ERROR	ETX
-----	-----	-----	-----	-------	-----

2.1.5 Multiple item read message format without checksum:

The maximum number of items in one read request is ten items.

Request:

STX	TID	PID	ADR	ITEM1;ITEM2;....ITEM10	ETX
-----	-----	-----	-----	------------------------	-----

Positive answer:

ACK	TID	PID	ADR	DATA1;DATA2;....DATA10	ETX
-----	-----	-----	-----	------------------------	-----

Negative answer:

NAK	TID	PID	ADR	ERROR	ETX
-----	-----	-----	-----	-------	-----

2.1.6 Multiple item read message format with checksum:

Request:

The maximum number of items in one read request is ten items.

STX	TID	PID	ADR	ITEM1;ITEM2;....ITEM10	CRC	ETX
-----	-----	-----	-----	------------------------	-----	-----

Positive answer:

ACK	TID	PID	ADR	DATA1;DATA2;....DATA10	CRC	ETX
-----	-----	-----	-----	------------------------	-----	-----

Negative answer:

NAK	TID	PID	ADR	ERROR	ETX
-----	-----	-----	-----	-------	-----

2.2 Message descriptions

2.2.1 Field description

Fields		Dec	Hex	Symbol	Length	Description
STX	Start Character	2	02	Const	1 Byte	Start of request frame.
TID ¹	Transaction Identifier	0-255	0-FF	"00"- "FF" H	2 Char	The client should increase TID with 1 for each request Recopied by the server from the received request
PID ¹	Protocol Identifier	0-255	0-FF	"00"- "FF" H	2 Char	"00"=Without checksum (TCP, UDP) "01"=With checksum (serial) Recopied by the server from the received request
ADR ¹	Address of 900 for RS485 (point to point)	0-255	0-FF	"00"- "FF" H	2 Char	Indicates which of the 256 PI-900 units on a serial link is being addressed. Recopied by the server from the received request
ITEM	OPC Item	-	-	8-bit ASCII	N Char	Format Channel.Device.Tag Should be printable characters excluding ' ' and ';'.
CRC ¹	Checksum	-	-	"00"- "FF" H	2 Char	Checksum byte obtained executing the sum, modulo 256, of all transmitted ASCII from TID to OPC Item / DATA included. checksum is never used when NAK response.
ETX	Termination Character	3	03	Const	1 Byte	End character of the request and response frame.
ACK	Positive answer	6	06	Const	1 Byte	Response with value
DATA	Values when reading and writing	-	-	8-bit ascii	N Char	N characters data according to item or write type
NAK	Negative answer	21	15	Const	1 Byte	Response with error code
ERROR	Error code	-	-	-	1 Char	Error code details see section 2.3.3

2.2.2 Control characters

Character	Name	Dec	Hex	Description
	Vertical bar	124	7C	Delimiter between OPC Item and write value
;	Semicolon	59	3B	Delimiter between OPC Items when multiple request

Total length of a request message is 1500 char including STX and ETX.

¹TID, PID, ADR and CRC are obtained by writing the single byte hex value as a two character string.

2.2.3 Error description

If there are problems to answer the requested OPC item in a correct way the Pii-900 will send an error character instead of the normal response message.

Character	Dec	Hex	Description
C	67	43	Crc fault
D	68	44	DataType (Must be VT_BSTR)
I	73	49	Can't find Item ID
L ¹	76	4C	Bad length
M	77	4D	Timeout master port
N ¹	78	4E	No complete request message With this error code the TID, PID and ADR will always be "00" in the error response message
O	79	4F	To many read items
S ¹	83	53	No start character
T	84	54	Timeout slave port
V	86	56	Validation (Write value out of range)
(X)	88	58	Internal use for wrong multi drop address

¹Discarded

2.3 Examples

2.3.1 Read request

If you want to ask for the value of a namespace Channel.Device.Tag which in your case is e.g. total energy in a meter connected over a network. The only thing you need to know is the name of the OPC Item. Let's say the name is "A.B.C" then the total string you must send to the PI-900 if you want to use checksum calculation is:

<STX><TID1><TID2><PID1><PID2><ADR1><ADR2>A.B.C<CRC1><CRC2><ETX>

2.3.1.1 Read request without checksum

Protocol = 0

	TID		PID		ADR		OPC Item	
STX	'0'	'0'	'0'	'0'	'0'	'0'	A.B.C	ETX
02	30	30	30	30	30	30	41 2E 42 2E 43	03

2.3.1.2 Read request with checksum

Protocol = 1

	TID		PID		ADR		OPC Item	CRC		
STX	'0'	'0'	'0'	'1'	'0'	'0'	A.B.C	'4'	'3'	ETX
02	30	30	30	31	30	30	41 2E 42 2E 43	34	33	03

2.3.2 Checksum calculation

Checksum byte is obtained executing the sum of all transmitted ASCII from TID to OPC Item or Data included.

To calculate the checksum manually you can use the calculator in Windows. Choose Hex and Byte on the calculator.

Try to use big letters A B C D E F in the ASCII frame. In hexadecimal f = F, during the crc calculation f and F represents different ASCII numbers.

The checksum in the example above is:

$(30+30+30+31+30+30+41+2E+42+2E+43)\text{mod } 256 = 43$ then change the binary number 43 to the two char upper case '4' '3'.

3. Appendix

3.1 Message Structure Obsolete

This was the absolutely first telegram structure and it's no longer used.

Single item read message format:

Request:

STX	TID	ADR	ITEM	CRC	ETX
-----	-----	-----	------	-----	-----

Positive answer:

ACK	TID	ADR	DATA	CRC	ETX
-----	-----	-----	------	-----	-----

Negative answer:

NAK	TID	ADR	ERROR	CRC	ETX
-----	-----	-----	-------	-----	-----