Application Note 1001

A technique for implementation of Time Stamping of Data using the MODBUS Encapsulated Interface Transport

Summary

One of the weakness of MODBUS is the absence of time stamping support for acquired data. It is desirable in many situations to provide this support without causing incompatibility to the standard set of MODBUS functions/frames supported by a device. This application note explains a method of implementing the same using the MODBUS Encapsulated Interface Transport.

1.0 The MODBUS Encapsulated Interface Transport format

The MEI format is a MODBUS frame specification which allows end users to define private command normally not supported by the standard MODBUS frames. The typical format of a MEI frame is as below:
Function Code = 43 (0x2B).

The generic MODBUS frame format of Request, Response and Exception Response for a MEI frame is given below:-

Request:-

Function Code	1 Byte	0x2B (Fixed for MEI frames)
MEI Type	1 byte	(User Defined)
MEI Type specific data	n Bytes	

Response:-

Function code	1 Byte	0x2B
MEI Type	1 byte	(User Defined)
MEI type specific data	n Bytes	

Exception:-

Function code	1 Byte	0xAB
MEI Type	1 Byte	(User Defined)
Exception code	1 Byte	01, 02, 03 or 04

The *MEI Type* is a MODBUS assigned number and therefore will be unique, the values between 1 to 127 are Public and the range of values between 128 to 255 are user defined. This technique of supporting time stamped data defines its own private *MEI Types* for time stamped data based on the two standard types of "input" objects supported by MODBUS – Discrete Inputs and Input Registers. If required, the same can be extended to the "output" object types – Coils and Holding Registers.

The User Defined MEI Types for 'time-stamped' versions of the two "input" object types as per this implementation is as follows:

Function	MEI Type
Read 7 Octet Time Stamped Discrete Input	0x82
Read 7 Octet Time Stamped Input Registers	0x84
Read 4 Octet Time Stamped Discrete Input	0x92
Read 4 Octet Time Stamped Input Registers	0x94

2.0 Read 7 Octet Time Stamped Discrete Input

Function Code = 43 (0x2B)
MEI Type = 130 (0x82)

The Request, Response and Exception Response frame formats for this frame is as follows.

Request

Field	**Width**	**Possible/Range of Values**
Function code	1 Byte	0x2B
MEI Type	1 Byte	0x82
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Inputs	2 Bytes	'm'

Response

Field	**Width**	**Possible/ Range of Values**	
Function Code	1 Byte	0x2B	
MEI Type	1 Byte	0x82	
Byte count	1 Byte	N + M + 2	No. of following bytes except CRC
Input Status Byte Count	1 Byte	N	No. of bytes of 'Input Status Data'
Input Status	'n' Bytes	n = N or N+1	Bit stuffed input status data
Time Stamp Byte Count	1 Byte	M = m * 7	No of bytes of Time Stamp Data
Time Stamp	m * 7 Bytes		Time stamp data

Exception

Field	**Width**	**Possible/Range of Values**
Function code	1 Byte	0xAB
MEI Type	1 Byte	0x82
Exception code	1 Byte	01, 02, 03, 04

Example:-

Request		**Response**	
Slave Id	**0x01**	**Slave Id**	**0x01**
Function Code	**0x2B**	**Function Code**	**0x2B**
MEI Type	**0x82**	**MEI Type**	**0x82**
Starting Address Hi	**0x00**	**Byte Count**	**0x9F (N + M + 2)**
Starting Address Lo	**0xC4**	**Input Status Byte Count**	**0x03 (N)**
Quantity of Inputs Hi	**0x00**	**Inputs Status 204-197**	**0xAC**
Quantity of Inputs Lo	**0x16**	**Inputs Status 212-205**	**0xDB**
CRC Hi	**0xxx**	**Inputs Status 218-213**	**0x35**
CRC Lo	**0xxx**	**Time Stamp Byte Count**	**0x9A (M)**
		Timestamp Val-197-ms Lo	
		Timestamp Val-197-ms Hi	
		Timestamp Val-197-min	
		Timestamp Val-197-hrs	
		Timestamp Val-197-day	
		Timestamp Val-197-month	
		Timestamp Val-197-year	
		Timestamp Val-198-ms Lo	
		Timestamp Val-198-ms Hi	
		Timestamp Val-198-min	
		Timestamp Val-198-hrs	
		Timestamp Val-198-day	
		Timestamp Val-198-month	
		Timestamp Val-198-year	
		do	
		do	
		do(up to 218)	
		CRC Hi	**0xxx**
		CRC Lo	**0xxx**

2.1 Read 7 Octet Time Stamped Input Registers

Function Code = 43 (0x2B)
MEI Type = 132 (0x84)

The Request, Response and Exception Response frame format for 'Read 7 Octet Time Stamped Input Registers' is as follows.

Request

Field	**Width**	**Possible/Range of Values**
Function code	1 Byte	0x2B
MEI Type	1 Byte	0x84
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Inputs	2 Bytes	N

Response

Field	**Width**	**Possible/Range of Values**	
Function code	1 Byte	0x2B	
MEI Type	1 Byte	0x84	
Byte count	1 Byte	(N x 2) + (N x 7)	No. of following bytes except CRC bytes
Register values	N x 2 Bytes		Two byte register data
Time Stamp	N x 7 Bytes		Time stamp data

Exception

Field	**Width**	**Possible/Range of Values**
Function code	1 Byte	0xAB
MEI Type	1 Byte	0x84
Exception code	1 Byte	01, 02, 03, 04

Example:-

Request	
Slave Id	**0x01**
Function Code	**0x2B**
MEI Type	**0x84**
Starting Address Hi	**0x00**
Starting Address Lo	**0x08**
Quantity of Input Reg. Hi	**0x00**
Quantity of Input Reg. Lo	**0x02**
CRC Hi	**0xxx**
CRC Lo	**0xxx**

Response	
Slave Id	**0x01**
Function Code	**0x2B**
MEI Type	**0x84**
Byte Count	**0x12**
Input Reg. 9 Hi	**0x00**
Input Reg. 9 Lo	**0x0A**
Timestamp Val-Reg.9-ms Lo	**0xDD**
Timestamp Val-Reg.9-ms Hi	**0xEE**
Timestamp Val-Reg.9-min	**0x3A**
Timestamp Val-Reg.9-hrs	**0x0F**
Timestamp Val-Reg.9-day	**0xB4**
Timestamp Val-Reg.9-month	**0x03**
Timestamp Val-Reg.9-year	**0x03**
Input Reg. A Hi	**0x00**
Input Reg. A Lo	**0x0E**
Timestamp Val-Reg.A-ms Lo	**0xCC**
Timestamp Val-Reg.A-ms Hi	**0xEE**
Timestamp Val-Reg.A-min	**0x3A**
Timestamp Val-Reg.A-hrs	**0x0F**
Timestamp Val-Reg.A-day	**0xB4**
Timestamp Val-Reg.A-month	**0x03**
Timestamp Val-Reg.A-year	**0x03**
CRC Hi	**0xxx**
CRC Lo	**0xxx**

2.2 Read 4 Octet Time Stamped Discrete Input

Function Code = 43 (0x2B)
MEI Type = 146 (0x92)

The Request, Response and Exception Response frame format for the 'Read 4 Octet Time Stamped Discrete Input' will be as follows.

Request

Field	**Width**	**Possible/Range of Values**
Function code	1 Byte	0x2B
MEI Type	1 Byte	0x92
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Inputs	2 Bytes	m

Response

Field	**Width**	**Possible/Range of Values**	
Function code	1 Byte	0x2B	
MEI Type	1 Byte	0x92	
Byte count	1 Byte	N + M + 2	No. of following bytes except CRC bytes
Input Status Byte Count	1 Byte	N	No. of bytes of 'Input Status Data'
Input Status	n Byte	n = N or N+1	Bit stuffed input status data
Time Stamp Byte Count	1 Byte	M = m * 4	No of bytes of Time Stamp Data
Time Stamp	m x 4 Bytes		Time stamp data

Exception

Field	**Width**	**Possible/Range of Values**
Function code	1 Byte	0xAB
MEI Type	1 Byte	0x92
Exception code	1 Byte	01, 02, 03, 04

Example:-

Request	
Slave Id	**0x01**
Function Code	**0x2B**
MEI Type	**0x82**
Starting Address Hi	**0x00**
Starting Address Lo	**0xC4**
Quantity of Inputs Hi	**0x00**
Quantity of Inputs Lo	**0x16**
CRC Hi	**0xxx**
CRC Lo	**0xxx**

Response	
Slave Id	**0x01**
Function Code	**0x2B**
MEI Type	**0x82**
Byte Count	**0x98 (N + M + 2)**
Input Status Byte Count	**0x03 (N)**
Inputs Status 204-197	**0xAC**
Inputs Status 212-205	**0xDB**
Inputs Status 218-213	**0x35**
Time Stamp Byte Count	**0x94 (M)**
Timestamp Val-197-ms Lo	
Timestamp Val-197-ms Hi	
Timestamp Val-197-min	
Timestamp Val-197-hrs	
Timestamp Val-198-ms Lo	
Timestamp Val-198-ms Hi	
Timestamp Val-198-min	
Timestamp Val-198-hrs	
do	
do	
do(up to 218)	
CRC Hi	**0xxx**
CRC Lo	**0xxx**

2.3 Read 4 Octet Time Stamped Input Registers

Function Code = 43 (0x2B)
MEI Type = 148 (0x94)

The Request, Response and Exception Response frame formats for 'Read 4 Octet Time Stamped Input Registers' is as follows.

Request

Field	**Width**	**Possible/Range of Values**
Function code	1 Byte	0x2B
MEI Type	1 Byte	0x94
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of Inputs	2 Bytes	N

Response

Field	**Width**	**Possible/Range of Values**	
Function code	1 Byte	0x2B	
MEI Type	1 Byte	0x94	
Byte count	1 Byte	(N x 2) + (N x 4)	No. of following bytes except CRC bytes
Register values	N x 2 Bytes		Two byte register data
Time Stamp	N x 4 Bytes		Time stamp data

Exception

Field	**Width**	**Possible/Range of Values**
Function code	1 Byte	0xAB
MEI Type	1 Byte	0x94
Exception code	1 Byte	01, 02, 03, 04

Example:-

Request	
Slave Id	**0x01**
Function Code	**0x2B**
MEI Type	**0x94**
Starting Address Hi	**0x00**
Starting Address Lo	**0x08**
Quantity of Input Reg. Hi	**0x00**
Quantity of Input Reg. Lo	**0x02**
CRC Hi	**0xxx**
CRC Lo	**0xxx**

Response	
Slave Id	**0x01**
Function Code	**0x2B**
MEI Type	**0x94**
Byte Count	**0x0C**
Input Reg. 9 Hi	**0x00**
Input Reg. 9 Lo	**0x0A**
Timestamp Val-Reg.9-ms Lo	**0xDD**
Timestamp Val-Reg.9-ms Hi	**0xEE**
Timestamp Val-Reg.9-min	**0x3A**
Timestamp Val-Reg.9-hrs	**0x0F**
Input Reg. A Hi	**0x00**
Input Reg. A Lo	**0x0E**
Timestamp Val-Reg.A-ms Lo	**0xCC**
Timestamp Val-Reg.A-ms Hi	**0xEE**
Timestamp Val-Reg.A-min	**0x3A**
Timestamp Val-Reg.A-hrs	**0x0F**
CRC Hi	**0xxx**
CRC Lo	**0xxx**

3.0 The Two Types of Time Stamping Format

1. 7 Octet Time Format
2. 4 Octet Time Format

The 7 Octet Time Format

Octets	Bits 7 to 0	Range
1	Milliseconds-Lower Bytes	0 to 59999 ms
2	Milliseconds-Higher Bytes	
3	Minutes	0 to 59 min
4	Summer Time/Standard Time (Bit 7) and Hours (Bit 6 to 0)	0 to 23 Hrs
5	Day of Week (Bits 7 to 5) and Month (Bits 4 to 0)	1 to 31 days of month 1 to 7 days of week
6	Months	1 to 12 months
7	Years	0 to 99 years

The 4 Octet Time Format

Octets	Bits 7 to 0	Range
1	Milliseconds-Lower Bytes	0 to 59999 ms
2	Milliseconds-Higher Bytes	
3	Minutes	0 to 59 min
4	Summer Time/Standard Time (Bit 7) and Hours (Bit 6 to 0)	0 to 23 Hrs

4.0 Exception Code Description

Code 01 -- ILLEGAL FUNCTION
The function code received in the query is not an allowable action for the server (or slave). This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server (or slave) is in the wrong state to process a request of this type, for example because it is not configured and is being asked to return register values.

Code 02 -- ILLEGAL DATA ADDRESS
The data address received in the query is not an allowable address for the server (or slave). More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed, a request with offset 96 and length 5 will generate exception 02.

Code 03 -- ILLEGAL DATA VALUE
A value contained in the query data field is not an allowable value for server (or slave). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value of any particular register.

Code 04 -- SLAVE DEVICE FAILURE
An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.

References:

1) *MODBUS over Serial Line Specification and Implementation Guide V.10 Nov 2002 at www.modbus.org*