MODBUS Protocol Specification

V2.0

MODBUS APPLICATION PROTOCOL SPECIFICATION

V1.0

CONTENTS

1 **Introduction**.....

1.1 Scope of this document.....

1.2 Protocol overview.....

1.3 Contacts.....

2 **Modbus transmission modes**.....

2.1 RTU transmission.....

2.2 Frame checking field.....

2.2.1 Frame description.....

2.2.2 RTU Message framing.....

2.2.3 RTU CRC checking .....

2.2.4 Data signal Rate .....

2.2.5 Data Formats .....

3 MODBUS Function Codes.....

3.1 04(0x04)Read input registers.....

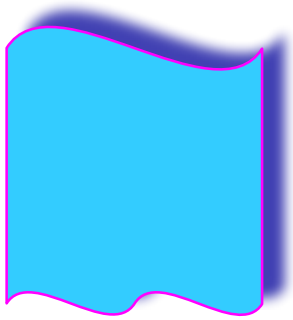
3.2 06(0x06)Write single register.....

3.3 16(0x10)Write Multiple Holding register.....

3.4 17(0x11)Read Device ID.....

4 MODBUS register map.....

Part



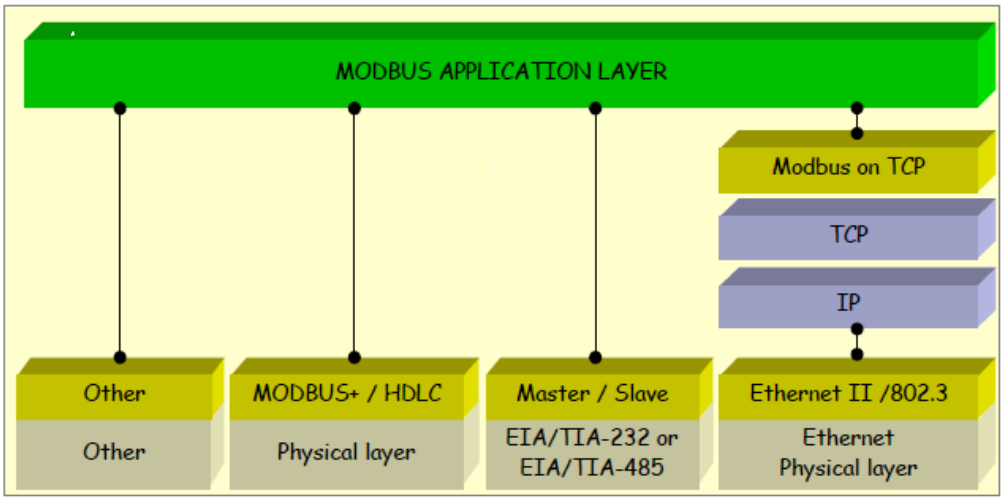
Introduction

1 Scope of this document

This document provides information for OwenBros electric devices implementing the MODBUS RTU protocol.

MODBUS is an application layer messaging protocol, positioned at level 7 of the OSI model, that provides client/server communication between devices connected on different types of buses or networks. It is currently implemented using:

- TCP/IP over Ethernet. See MODBUS Messaging Implementation Guide V1.0a
- Asynchronous serial transmission over a variety of media (wire : EIA/TIA-232-E, EIA-422, EIA/TIA-485-A, fiber, radio, etc.)
- MODBUS PLUS, a high speed token passing network.



MODBUS Communication stack

The industry's serial de facto standard since 1979, MODBUS continue to enable millions of automation devices to communicate. Today support for the simple and elegant structure of MODBUS continues to grow. The Internet community can access MODBUS at a reserved system port 502 on the TCP/IP stack.

MODBUS is a request/reply protocol and offers services specified by function codes. MODBUS function codes are elements of MODBUS request/reply PDUs. The objectives of this document is to describe the function codes used within the framework of MODBUS transactions.

Part II

2 MODBUS transmission Modes

One serial transmission modes is defined: The RTU mode.

2.1 RTU Transmission mode

When devices communicate on a MODBUS serial line using the RTU (Remote Terminal Unit)mode, each 8-bit byte in a message contains two 4-bit hexadecimal characters. Each message must be transmitted in a continuous stream of characters.

The format (11bits) for each byte in RTU mode is:

Coding System: 8-bit binary

Bits per Byte: 1 start bit

8 data bits, least significant bit sent first

1 bit for parity completion

1 stop bit

Even parity is required.

2.2 Frame Checking Field:

Cyclical Redundancy Checking(CRC)

2.2.1 Frame description

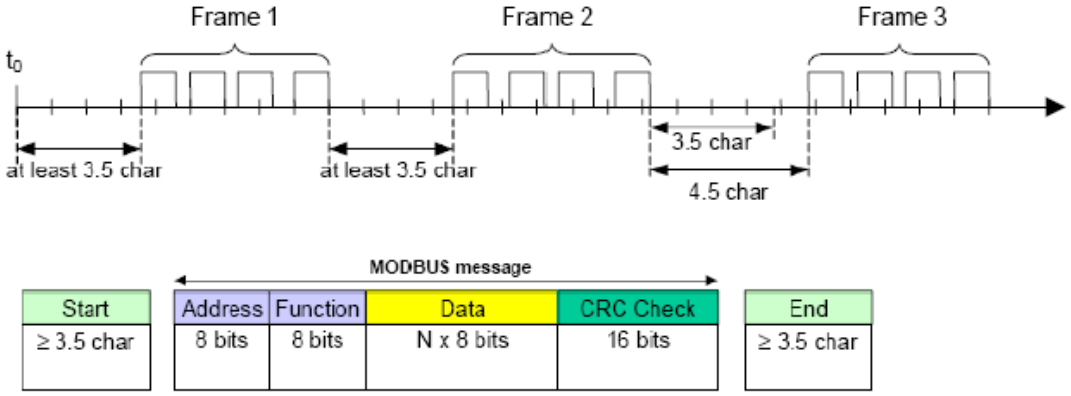
Slave Address	Function Code	Data	CRC
1byte	1 byte	0 up to 252 byte(s)	2 bytes CRC Low CRCHi

The maximum size of a MODBUS RTU frame is 256 bytes.

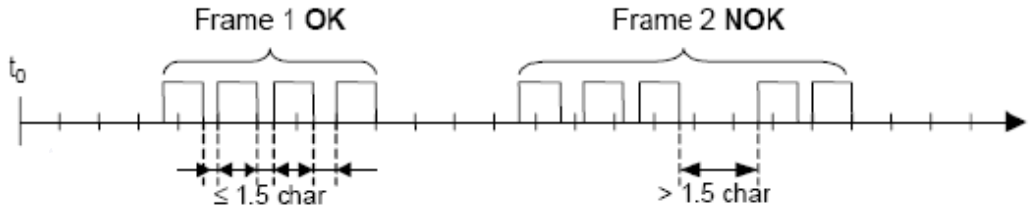
2.2.2 RTU Message Framing

A MODBUS message is placed by the transmitting device into a frame that has a known beginning and ending point. This allows devices that receive a new frame to begin at the start of the message, and to know when the message is completed. Partial message must be detected and errors must be set as a result.

In RTU mode, message frames are separated by a silent interval of at least 3.5 character times. In the following sections, this time interval is called t3.5.



The entire message frame must be transmitted as a continuous stream of characters. If a silent interval of more than 1.5 character times occurs between two characters, the message frame is declared incomplete and should be discarded by the receiver.



Note:

The implementation of RTU reception driver may imply the message of a lot of interruptions due to the t1.5 and t3.5 times.

2.2.3 RTU CRC Checking

The RTU mode includes an error-checking field that is based on a Cyclical Redundancy Checking(CRC) method performed on the message contents.

2.2.4 Data signal Rate

OwenBrothers slave device supports the following baud rates

Baud Rate	Comments
1200	
2400	
4800	
9600	

2.2.5 Data Formats

2.2.5.1 unsigned 16-bit integer word Format

The Modbus applications support 16 bit integer information for several of the function codes.

A read or write to a modbus register comprise a 2×8 bit byte.

2.2.5.2 IEE 32-it Floating-point Register Format

The Modbus applicaion support IEE 32-bit floating point information for several of the function codes.

Part III

3 MODBUS Function Codes

Owen Brothers electric Modbus RTU uses a subset of the standard Modbus function codes to provide access to measurement and information registers. These standard function codes provides basic support for IEE32-bit floating point number, 16 bit integer .

Function Code	Name	Usage
0x04	**Read Input Registers**	Used for reading floating point and 16 bit integer measurements
Ox06	**Write single Registers**	Used for writing floating point and 16 bit integer values to single registers
0x10	**Write multiple holding register**	Write multiple holding register
0x11	**Report Device ID**	Used for reading device information including device ID, description, software version etc

3.1 04(0x04)Read Input Registers

This function codes is used to read 1 to 125 continue input registers in a remote device. The Request PDU specifies the starting register address and the number of register.

In the PDU Register are addressed starting at zero. Therefore input register numbered 1-16 are addressed as 0-15

The register data in the response message are packed as two byte per register, with the binary contents right justified with each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

Request

Function code	1 Byte	0X04
Starting Address	2 Byte	0X0000 to 0XFFFF
Quantity of register	2 Byte	0x0001 to 0x007D

Response

Function code	1 Byte	0X04

Byte count	1 Byte	2 × N*
Register value	N* ×2 Bytes	

N* = Quantity of registers

Error

Function code	1 Byte	0x84
Exception code	1 Byte	0x01 or 0x02 or 0x03 or 0x04

An Example of a request to read input register 9 from slave address 2 using RTU format, where the register contains the 16 bit hex value 0x55AA

Request	
Field Name	(Hex)
Slave Address	02
Function	04
Starting Address Hi	00
Starting Address Lo	08
No. of Register Hi	00
No. of Register Lo	01
Check Sum	CRC
Check Sum	CRC

Response	
Field Name	(Hex)
Slave Address	02
Function Code	04
Byte Count	02
Input register Hi	55
Input register Lo	AA
Check Sum	CRC
Check Sum	CRC

3.2 06(0x06) Write Multiple register

This function code is used to write a single holding register in a remote device. The Request PDU specifies the address of the register to be written.

The normal response is an echo of the request, returned after the register contents have between written.

Request

Function code	1 Byte	0X06
Register Address	2 Byte	0X0000 to 0XFFFF
Register Value	2 Byte	0x0000 to 0xFFFF

Response

Function code	1 Byte	0X06
Register Address	2 Byte	0x0000 to 0xFFFF
Register value	2 Bytes	0x0000 to 0xFFFF

Error

Error code	1 Byte	0x86
Exception code	1 Byte	0x01 or 0x02 or 0x03 or 0x04

Example

An Example of a writing to register 40001(Primary VT Ratio) the value 400,to slave address 5 in RTU mode

Request

Request	
Field Name	(Hex)
Slave Address	05
Function	06
Register Address Hi	00
Register Address Lo	00
Register value Hi	01
Register value Lo	90
Check Sum	CRC
Check Sum	CRC

Response	
Field Name	(Hex)
Slave Address	05
Function Code	06
Register Address Hi	00
Register Address Lo	00
Register value Hi	01
Register value Lo	90
Check Sum	CRC
Check Sum	CRC

3.3 16(0x10) Write Multiple register

This function code is used to write a block of contiguous registers in a remote device. The requested written values are specified in the request data field. Data is packed as two bytes per register.

The normal response returns the function code, starting address, and quantity of registers written.

Request

Function code	1 Byte	0X10
Starting Address	2 Byte	0X0000 to 0Xffff
Quantity of register	2 Byte	0X0000 to 0XFFFF
Byte Count	1 Byte	2 × N*
Register value	N*×2 Byte	Value

N* = Quantity of registers

Response

Function code	1 Byte	0X10
Starting Address	2 Byte	0X0000 to 0Xffff
Quantity of register	2 Bytes	1 to 123 (0x7B)

Error

Error code	1 Byte	0X90
Exception Code	1 Byte	0x01 or 0x02 or 0x03 or 0x04

Example

An example of a writing to register 40915 (Pulse value for power) the value 1.0,to slave address 5 in RTU mode

Request	
Field Name	(Hex)
Slave Address	05
Function code	10
Starting Address Hi	03
Starting Address Lo	92
No. of Register Hi	00
No. of Register Lo	02
Byte count	04
Register value Hi	3F
Value	80
value	00

Register value Lo	00
Check Sum	77
Check Sum	26

Response	
Field Name	(Hex)
Slave Address	05
Function	10
Starting Address Hi	03
Starting Address Lo	92
No. of Register Hi	00
No. of Register Lo	02
Check Sum	E1
Check Sum	E5

3.4 17(0x11) Report Device ID

This function code is used to read the description of the type. the current status, and other information .

The format of a normal response is shown in the following example. The data contents are specific to each type of device.

Function code	1 Byte	0X11

Response

Function Code	1 byte	11
Byte count	1 byte	1A
Device ID	1 byte	0D
Run Indicator	1 byte	FF 00=OFF FF=ON
Description	16 bytes	"D225 xxx.yy"
Serial number	4 bytes	0 to 4294967295
Hardware Version Engine	2 bytes	
Hardware Version Coms	2 bytes	
Hardware Version Display	2 bytes	

Error

Error Code	1 byte	91
Exception Code	1 byte	0x01 or 0x04

Example:

Slave Address	1 Byte	0X03
Function	1 Byte	0X11
Check Sum	1 Byte	CRC
Check Sum	1 Byte	CRC

Response

Slave Address	1 byte	03
Function Code	1 byte	11
Byte count	1 byte	1A
Device ID	1 byte	0D
Run Indicator	1 byte	FF
Description	16 bytes	44("D")D225 xxx.yy"
		32("2")
		32("2")
		35("5")
		20("space ")
		30("0")
		30("0)
		31("1")
		2E(".")
		30("0)
		32("2")
		00
		00
		00
		00
		00
Serial number Hi	1 byte	00
Serial number Hi	1 byte	01
Serial number Lo	1 byte	E2
Serial number Lo	1 byte	40
software Version Engine	1 bytes	01
Software Version Engine	1 bytes	02

Software Version Coms	1 bytes	00
Software Version Coms	1 bytes	00
Software Version Display	1 byte	00
Software Version Display	1 byte	00
Check sum	1 byte	CRC
Check sum	1 byte	CRC

Part IV

4 MODBUS Register map

This appendix describes all parameters accessible by Function Codes 0x03, 0x10.Parameters are grouped together according to the measurement been made, to simplify and speed up the reading of the data.

The availability of parameters and functions is depended on the device been accessed.

4.1 register Map Overview

The following table describes the global register map for the function Codes for our products.

OB7500 series register map

Address (hex)	Length (bytes)	Parameter Name	Access (R/W)	Function code	Data Format	Units
0x0010	4	Voltage L1	R	04/03	Float/Hex	V
0x0012	4	Voltage L2	R	04/03	Float/ Hex	V
0x0014	4	Voltage L3	R	04/03	Float/ Hex	V
0X004E	4	Frequency	R	04/03	Float/ Hex	Hz
0X0050	4	Current L1	R	04/03	Float/ Hex	A
0X0052	4	Current L2	R	04/03	Float/ Hex	A
0X0054	4	Current L3	R	04/03	Float/ Hex	A
0X0056	4	Current Neutral	R	04/03	Float/ Hex	A
0X0058	4	Current total	R	04/03	Float/ Hex	A
0x0090	4	Power L1	R	04/03	Float/ Hex	kW
0x0092	4	Power L2	R	04/03	Float/ Hex	kW
0x0094	4	Power L3	R	04/03	Float/ Hex	kW
0x0096	4	Power Total	R	04/03	Float/ Hex	kW
0x00D0	4	Apparent	R	04/03	Float/ Hex	kVA

		Power L1				
0x00D2	4	Apparent Power L2	R	04/03	Float/ Hex	kVA
0x00D4	4	Apparent Power L3	R	04/03	Float/ Hex	kVA
0x00D6	4	Apparent Power Total	R	04/03	Float/ Hex	kVA
0x0110	4	Reactive Power L1	R	04/03	Float/ Hex	kvar
0x0112	4	Reactive Power L2	R	04/03	Float/ Hex	kvar
0x0114	4	Reactive Power L3	R	04/03	Float/ Hex	kvar
0x0116	4	Reactive Power Total	R	04/03	Float/ Hex	kvar
0x0150	4	Power Factor L1	R	04/03	Float/ Hex	
0x0152	4	Power Factor L2	R	04/03	Float/ Hex	
0x0154	4	Power Factor L3	R	04/03	Float/ Hex	
0x0156	4	Power Factor Total	R	04/03	Float/ Hex	
0x0160	4	Import Active Energy	R	04/03	Float/ Hex	KWh
0x0162	4	Import reactive Energy	R	04/03	Float/ Hex	KWh
0x0166	4	Export Active Energy	R	04/03	Float/ Hex	KWh
0x0168	4	Export reactive Energy	R	04/03	Float/ Hex	KWh
0X0618	4	Total Energy	R	04/03	Float/ Hex	kWh

0x0800	4	Import active Energy	R	04	Float	KWh
0x0900	4	Export active Energy	R	04	Float	KWh
0X0700	4	Total active Energy	R	04	Float	kWh
0x0A00	4	Import Reactive Energy	R	04	Float	Kvarh
0x0B00	4	Export Reactive Energy	R	04	Float	kvarh
0x0524	2	Modbus slave address number	R/W	04/06	16 bit	address
0x0525	2	Modbus slave Baud rate	R/W	04/06	16 bit	1200bps 0x04B0 2400bps 0x0960 4800bps 0x12C0 9600bps 0x2580
0X0565	4	Clear Energy	W	10	hex	with KEY *1
0XFE00	2	Auto scan display items code 1	W	10	Hex	The display items codes are register address
0XFE01	2	Auto scan display items code 2	W	10	Hex	
0xFE02	2	Auto scan display items code 3	W	10	Hex	
...	...	…	....			
0xFE13	2	Auto scan display items code 20	W	10	Hex	
0xFE14	2	Button press display items code 2	W	10	Hex	

0xFE15	2	Button press display items code 3	W	10	Hex	
...	...	.....	...			
0xFE3B	2	Button press display items code 40	W	10	Hex	
0XFF00	4	Serial number	R/W	10	BCD	
0XFF02	4	manufacturer	R	04	Hex	
FF04H	2	Productions code	R	02	BCD	
FF05H	2	Hardware version	R	02	BCD	
FF06H	2	Software version	R	02	BCD	
FF07H	2	Rated voltage	R	02	Hex	
FF08H	2	Rated current	R	02	Hex	MSB：basic current Unit：0.1A LSB: Max current Unit：A
FF09H	2	Impulse constant	R	02	Hex	
FF12H	2	Auto scan display screens numbers	R	02	Hex	
FF12H	2	Button press display screens numbers	R	02	Hex	