Modbus-RTU Master Protocol Documentation

Revision: 20.04
Date: 2020-04-20

www.iqhome.org

Contents

1 Introduction

IQ Home ModBus-RTU master device enables access to ModBus-RTU slave devices over IQRF Network. Using Modbus-RTU master device user are able to send Modbus RTU request and receive Modbus RTU responses. The Modbus-RTU master device bridge the IQRF Network and the RS-485 communication Modbus-RTU network. The device implements slave role on IQRF Network side and Master role on RS-485 Modbus-RTU network side. ModBus-RTU master device can handle up to 31 slave devices on RS-485 Modbus-RTU network side.

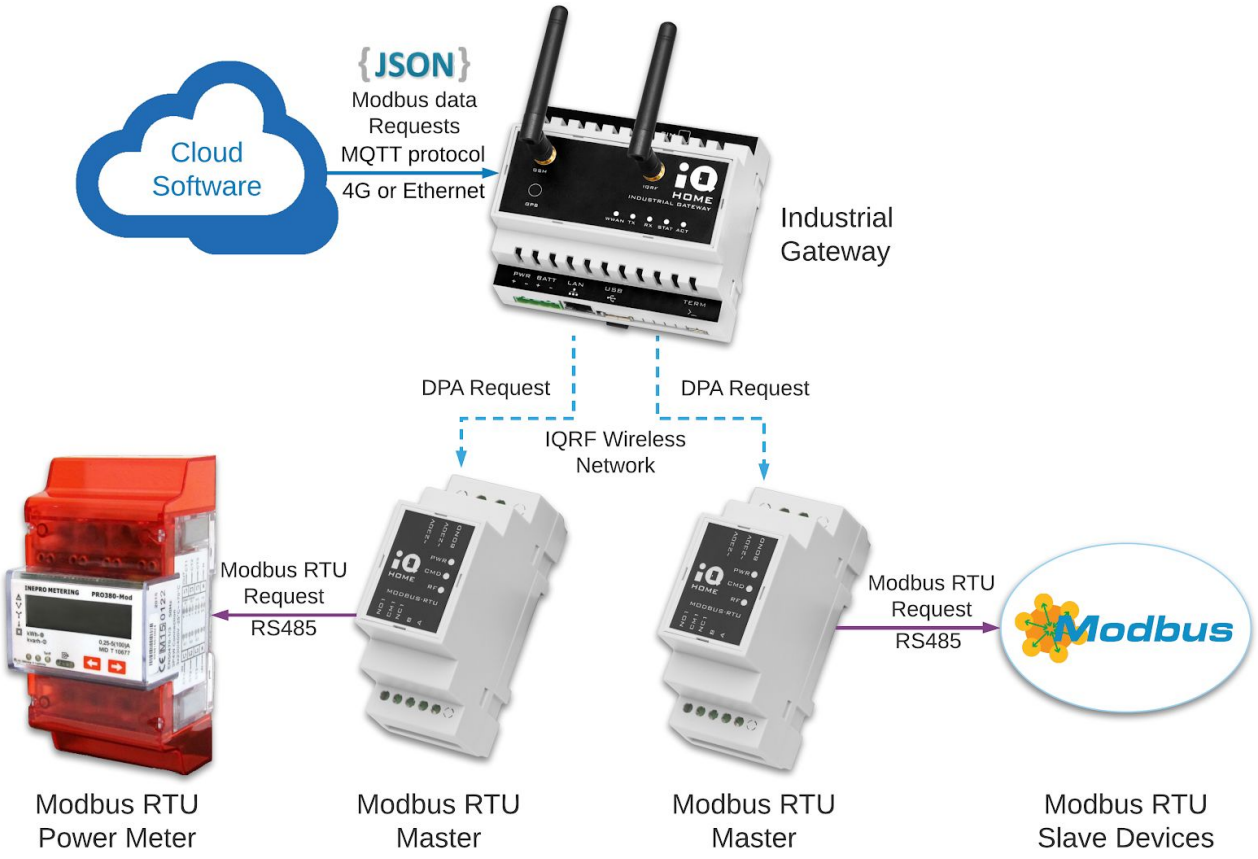
RF Network	IQRF
HWPID	0x8A5F
HWPIDver	1.0.xx (0x10xx)
IQRF OS	4.02D
IQRF DPA	3.02
IQRF RF Mode	LP or STD
Default RF Channel	52 (868.35 MHz)

1.1 Typical usage

With IQ Home ModBus-RTU master device the user can access and control ModBus-RTU devices from the cloud.

Typical request data flow:

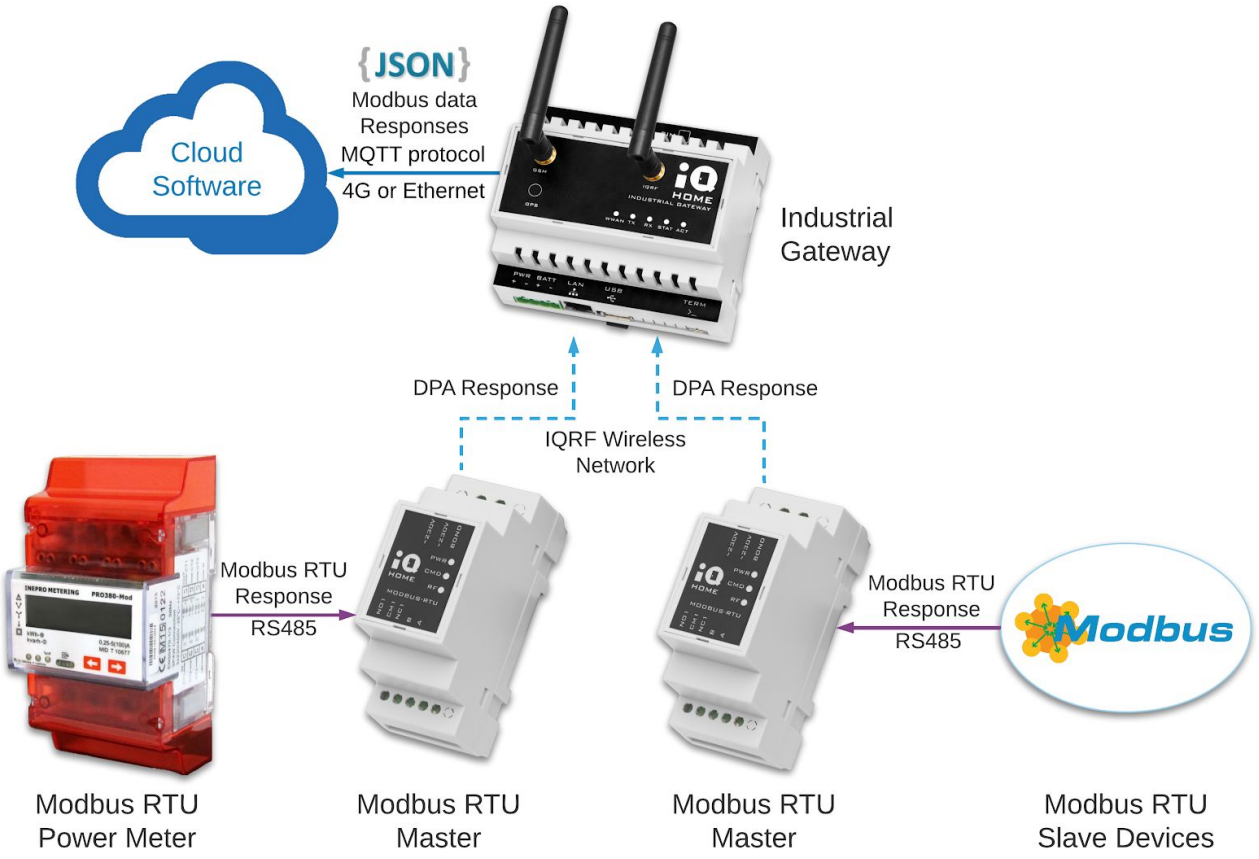
Cloud Software ➡ [Internet connection / MQTT protocol] ➡ **IQ Home Gateway** ➡ [IQRF Network] ➡ **IQ Home ModBus RTU Master** ➡ [Modbus RTU on RS485] ➡ **Modbus slave device**



Rerequest data flow

Typical response data flow:

Modbus slave device ➡ [Modbus RTU on RS485] ➡ **IQ Home ModBus RTU Master** ➡ [IQRF Network] ➡ **IQ Home Gateway** ➡ [Internet connection / MQTT protocol] ➡ **Cloud Software**



Response data flow

1.2 Main features

IQ Home Modbus-RTU master device main features:

- Modbus-RTU device calculates and checks the CRC-16 error check fields. Users don't need to care about it.
- Modbus-RTU master device can address up to 31 slave devices. From slave device address 0x01 to 0x1F.
- Modbus-RTU master device can send broadcast message. Broadcast address is 0x00.
- Automatic Time-Out control. Modbus-RTU master device generate time-out response, if the device don't receives any response from slave device.
- Automatic Request resend mechanism. Modbus-RTU master device automatically re-sends the Modbus-RTU request, if the device don't receives any response or receives corrupted response from slave device.
- Modbus-RTU master device communicate with two different mode:
    - 1 start bit
      8 data bits, least significant bit sent first
      1 bit for Even parity (default)
      1 stop bit

    - 1 start bit
      8 data bits, least significant bit sent first
      2 stop bits (no parity)

- Modbus-RTU master device can communicate with different Baud rates. Supported Baud rates:
    - 1200
    - 2400
    - 4800
    - **9600 (default)**
    - 19200
    - 38400
    - 57600
    - 115200

1.3 Modbus message structure

Modbus-RTU Message consists of three main parts:

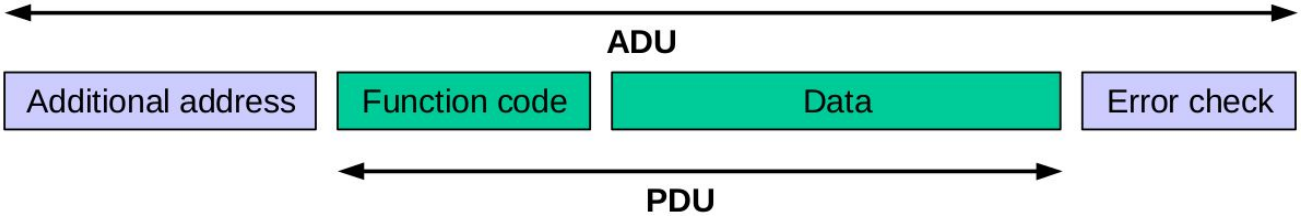
- Address field
- Protocol data unit (PDU)
- CRC-16 Error check

The full modbus-RTU Message is named application data unit (ADU) in official Modbus documentation.

Protocol data unit (PDU) consists of two main parts:

- Function code byte
- Data

Next picture shows the structure of Modbus-RTU Message:



With 0x38 PNUM code can user send and receive Modbus-RTU messages:

- PCMD byte contains the Modbus-RTU slave device address. The device address can be 0x01 to 0x1F
- PDATA contains the Modbus protocol data unit (PDU)

1.4 Additional information

More information about PDU data format, please read the offical MODBUS Protocol Specification: www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

IQ Home Modbus-RTU master device implements Modbus Serial Line Protocol and Implementation Guide V1.02: http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf

1.5 Compatible devices

Next table shows the devices which implement protocol described in this document. The table contains the Product code and the product name.

Code	Product name	RS-485
MB-RTU-01	ModBus-RTU master	✔

2 Send and Receive Modbus-RTU message

PNUM	PCMD
0x38	0x01 to 0x1F

With the command user can send request and receive response from Modbus-RTU RS-485 network.

2.1 Request

Request				
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**	**PDATA [up to 56B]**
NADR	0x38	Modbus-RTU save address 0x01 to 0x1F	0x85AF or 0xFFFF	Modbus protocol data unit (PDU) request

After receiving the IQRF DPA command the Master unit sends out the Modbus-RTU data unit created from content of PDATA field to RS-485 network.

- PCMD byte contains the Modbus-RTU slave device address. The device address can be 0x01 to 0x1F.
- PDATA contains the Modbus protocol data unit (PDU) request lika binary string.
- Modbus-RTU device calculates and checks the CRC-16 error check fields automatically.
- Modbus-RTU master device automatically re-sends the Modbus-RTU request, if the device don’t receives any response or receives corrupted response from slave device.

2.2 Response

Response						
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**	**ErrN [1B]**	**DpaValue [1B]**	**PDATA [up to 56B]**
NADR	0x38	Modbus-RTU save address + 0x80 0x81 to 0x9F	0x85AF	0x00 or error codes: 0x20 0x21 0x22	?	Modbus protocol data unit (PDU) response

The addressed node response with received Modbus protocol data unit (PDU) response.

- PCMD byte contains the Modbus-RTU slave device address + 0x80.
- PDATA contains the Modbus protocol data unit (PDU) response lika binary string.
- Modbus-RTU device check the received CRC-16 message integrity.

The Modbus-RTU master device can response with this error codes:

- 0x00 - Modbus response received successfully
- 0x20 - No Modbus response received (time-out)
- 0x21 - Received contains frame error (CRC or parity bit error)
- 0x22 - Received Modbus response is to long, can not send in DPA message

2.2.1 Modbus command example

Example: Reading value of two consecutive register starting from address 0x6000 from slave with slave address #1. Registers in Modbus are 16 bit wide.

Request:

Request				
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**	**PDATA [up to 56B]**
NADR	0x38	**0x01** save address	0x85AF or 0xFFFF	"**0360000002**" Modbus protocol data unit (PDU) request

PDATA contains 5 bytes. Meaning of PDATA (PDU) bytes:

- 03 - Function code is: (0x03) Read Holding Registers
- 6000 - Starting Address is 0x6000
- 0002 - Quantity of Registers is 0x0002

> **NOTE:**
>
> Please note Modbus-RTU protocol has different endianness from IQRF DPA protocol.
> - Numbers in **IQRF DPA** are represented in **big-endian** data format.
> - Numbers in **Modbus-RTU** are represented in **little-endian** data format.
>
> In **PDATA (PDU message part)** the numbers are represented in **little-endian** data format.

More information about Holding registers, please see Chapter 6.3 "03 (0x03) Read Holding Registers" in MODBUS Protocol Specification:
www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

Response:

Response						
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**	**ErrN [1B]**	**DpaValue [1B]**	**PDATA [up to 56B]**
NADR	0x38	0x81 Modbus-RTU save address 0x81 = 0x01 + 0x80	0x85AF	0x00	?	"**030412345678**" Modbus protocol data unit (PDU) response

Response PDATA contains 6 bytes. Meaning of PDATA (PDU) bytes:

- 03 - Function code is: (0x03) Read Holding Registers
- 04 - Byte count (number of following bytes)
- 1234 - Value of register located on address 0x6000 is 0x1234
- 5678 - Value of register located on address 0x6001 is 0x5678

NOTE:

Please note Modbus-RTU slave device can response with error code. Error response is represented with modified response function code. The error response code equals with request response code plus 0x80. For example at 0x03 the error response PDU message starts with 0x83. The second byte contains the exception code.
Typical exceptions codes are:

- 0x01 - Function code is not supported
- 0x02 - Address error
- 0x03 - Quantity error
- 0x04 - Request processing error

More information about the response and the exeption code at Holding registers, please see Chapter 6.3 "03 (0x03) Read Holding Registers" in MODBUS Protocol Specification: www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

3 Send Broadcast Modbus-RTU message

PNUM	PCMD
0x38	0x00

With the command user can send broadcast message to Modbus-RTU RS-485 network.

3.1 Request

Request				
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**	**PDATA [up to 56B]**
NADR	0x38	Modbus-RTU Broadcast address 0x00	0x85AF or 0xFFFF	Modbus protocol data unit (PDU) request

After receiving the IQRF DPA command the Master unit sends out the Modbus-RTU broadcast data unit created from content of PDATA field to RS-485 network.

- PCMD byte contains the Modbus-RTU broadcast address. The Modbus-RTU broadcast address is 0x00.
- PDATA contains the Modbus protocol data unit (PDU) request lika binary string.
- Modbus-RTU device calculates and checks the CRC-16 error check fields automatically.

> **NOTE:**
>
> Please note Modbus-RTU broadcast message types can be only write or modification commands. Broadcast messages does not has acknowledge process on RS-485 side.

3.2 Response

Response						
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**	**ErrN [1B]**	**DpaValue [1B]**	**PDATA [0B]**
NADR	0x38	Modbus-RTU save address + 0x80 0x81 to 0x9F	0x85AF	0x00	?	-

The addressed node acknowledges the request with a empty response (PDATA is empty, DPA Data length equal with zero).

3.3 Modbus broadcast command example

Example: Sets ON output (coil) address 0x0001 at all slave devices.

Request:

Request				
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**	**PDATA [up to 56B]**
NADR	0x38	**0x00** broadcast address	0x85AF or 0xFFFF	"**050001FF00**" Modbus protocol data unit (PDU) request

PDATA contains 5 bytes. Meaning of PDATA (PDU) bytes:

- 05 - Function code is: (0x05) Write Single Coil
- 0001 - Output address is 0x0001
- FF00 - Set ON

Response:

Response						
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**	**ErrN [1B]**	**DpaValue [1B]**	**PDATA [0B]**
NADR	0x38	0x80	0x85AF	0x00	?	-

Response contains empty PDATA.

4 Read and write configuration registers

PNUM	PCMD
0x38	0x3E

With the command user can set-up basic configuration registers. Configuration registers are stored in internal EEPROM. Values of configuration registers will be remains after power down.

4.1 Configuration registers

With configuration registers can be set-up:

- RS-485 communication speed and byte data format
- Timeout value for no response received from RS-485 network
- Request send number at no response, or response containing integrity error

All tree register is 1 byte wide register.

4.1.1 RS-485 communication speed and byte data format

Communication register is 1 byte wide register. With a communication register can be set-up the RS-485 communication speed and byte data format. Next table shows the usable register value combinations.

RS-485 communication register		
Register value	**Communication speed [baud]**	**Parity**
0x00	1200	Even parity
0x01	2400	Even parity
0x02	4800	Even parity
0x03 (default)	**9600**	**Even parity**
0x04	19200	Even parity
0x05	38400	Even parity
0x06	57600*	Even parity
0x07	115200*	Even parity
0x08	1200	No parity - two stop bits
0x09	2400	No parity - two stop bits
0x0A	4800	No parity - two stop bits
0x0B	9600	No parity - two stop bits

0x0C	19200	No parity - two stop bits
0x0D	38400	No parity - two stop bits
0x0E	57600*	No parity - two stop bits
0x0F	115200*	No parity - two stop bits

** Test purpose only, it is not recommended to use in real application.*

Modbus-RTU master device communicate with two different mode base on communication register value:

- Register value is between 0x00 - 0x07:
  1 start bit
  8 data bits, least significant bit sent first
  1 bit for Even parity
  1 stop bit

- Register value is between 0x8 - 0x0F:
  1 start bit
  8 data bits, least significant bit sent first
  2 stop bits (no parity)

4.1.2 Timeout register

Timeout register is 1 byte wide register. With a timeout register can be set-up the timeout value for no response received from RS-485 network. Timeout can be calculated with this equation:

$$Timeout\ [ms]\ = Register\ Value\ \times 10\ [ms]$$

- Minimum register value is 1 (10 ms).
- Maximum register value is 255 (2.55 s).
- **Default register value is 50 (0.5 s)**

4.1.3 Request send number

Request send number register is 1 byte wide register. With a request send number register can be set-up the number of attempt to send out Modbus request. If the device do not receive any response or response containing integrity error, then the device will resend the request. For example: If the register value is 3, than the device will send out the modbus request 3 times at communication error. After third time the device respond with error value to IQRF request.

- Minimum register value is 1.
- Maximum register value is 15.
- **Default register value is 3.**

4.2 Read register values

4.2.1 Request

Request				
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**	**PDATA [0B]**
NADR	0x38	0x3E	0x85AF or 0xFFFF	-

Request does not contains PDATA. DPA Data length have to be zero.

4.2.2 Response

Response								
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**	**ErrN [1B]**	**DpaValue [1B]**	**PDATA 1. byte**	**PDATA 2. byte**	**PDATA 3. byte**
NADR	0x38	0xBE	0x85AF	0x00	?	RS-485 communication register	Timeout register	Request send number

4.3 Write register values

4.3.1 Request

Request						
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**	**PDATA [3B]**		
NADR	0x38	0x3E	0x85AF or 0xFFFF	RS-485 communication register	Timeout register	Request send number

4.3.2 Response

Response of write register write request equals to response of read request described in chapter 4.2.2. Response.

5 Read product information

PNUM	PCMD
0x3E	0x00

The command is usable to get basic information about the product.

5.1 Request

The request does not contains any PDATA information.

Request			
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**
NADR	0x3E	0x00	0x15AF or 0xFFFF

5.2 Response

The addressed node response with all basic product information.

Response						
NADR [2B]	**PNUM [1B]**	**PCMD [1B]**	**HWPID [2B]**	**ErrN [1B]**	**DpaValue [1B]**	**PDATA [16B]**
NADR	0x3E	0x80	0x15AF	0x00	?	Product information

5.2.1 PDATA structure of the response

Response PDATA array contains 16 byte long product information string.

PDATA															
1. byte	2. byte	3. byte	4. byte	5. byte	6. byte	7. byte	8. byte	9. byte	10. byte	11. byte	12. byte	13. byte	14. byte	15. byte	16. byte
Product Code											Hardware revision				

- Product Code = Main product code of the product stored in ASCII characters.
- Hardware revision = Internal information about the hardware revision.

6 Release Notes

Property	Value
Protocol version	1.0.xx
IQRF OS version	4.02D
IQRF DPA version	3.02
Date of release	18/07/2018
Notes	First revision of IQ Home Modbus-RTU Master protocol.