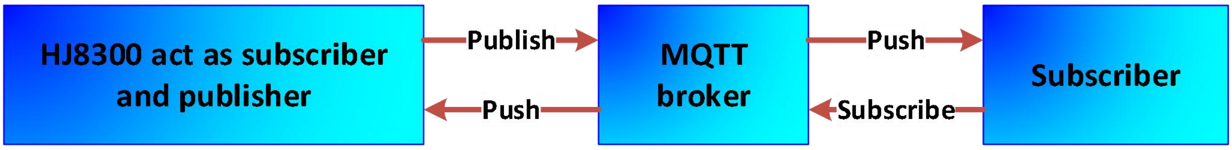
# HJ8300 MQTT and MODBUS-RTU Development Instructions

## 1. MQTT Instruction

MQTT, based on TCP/IP protocol, is a “lightweight” communication protocol based on publish/subscribe mode. It was released by IBM in 1999. MQTT biggest advantage is to provide remotely connected device with real-time reliable message service by very little codes and limited broadband. As an instant communication protocol with low cost and low broadband occupation, it has been widely used in IoT, small device, mobile application, etc.

MQTT publish/subscribe message mode as below:



MQTT protocol is finished in communication between the client side and the server side. During the communication, MQTT protocol has three roles: Publisher, Broker and Subscriber.

Message publisher and subscriber are client side, message broker is server side and message publisher also can be subscriber.

MQTT transmitted message includes: Topic and payload two parts

1. Topic: Message type. After the subscriber subscribes, it will receive this topic message content (payload).
2. Payload: Message contents. It means the specific contents that the subscriber is going to use.

## 2. HJ8300 MQTT Manual

Install software mosquitto-1.4.15 and json-c-0.12

Mosquitto provides Windows, Linux and qnx OS versions. HJ8300 series are installed with Linux OS, so we choose mosquitto Linux version source code installation.

HJ8300 series MQTT demo uses json code. The gateway mode is used as MQTT client end. It can publish/subscribe MQTT message. The program is written in C.

**Demo program file instructions:**

main.c -- main program file
base.h --- header file
Makefile -- compiled Make file
poll_rftcm_s.txt -- config file, including MQTT server address and port configuration information

**MQTT message format:**

TOPIC of Gateway published message： / modbus /MQTT account/gateway MAC/COM port number

TOPIC of gateway subscribed message： / modbus /MQTT account/gateway MAC/COM port number

For example, the server publishes a time message to gateway < C2-46-00-01-43-92> No.1 device, and the format is as below:

```
/modbus/blue/C2-46-00-01-43-92/1
{
   "h":{
      "Type":5,
      "Sequnce":32
   },
   "b":{
      "Reference":5,
      "SamplingTime":"2019-03-27T17:47:09.000+0800",
      "Description":"Modbus data",
      "Bytes":6,
      "Data":[
         86,
         120,
         171,
         205,
         18,
         52
      ]
   }
}
```

Gateway subscribes a MQTT message: / modbus / MQTT account/ C2-46-00-01-43-92/1. If the server or other client-side sends this message topic contents, gateway will receive this message. It can be subscribed by using below commands:

```
mosquitto_sub -h 192.168.1.126 -p 1883 -t /modbus/blue/C2-46-00-01-43-92/1
```

-h: server IP or domain

-p: Server MQTT port

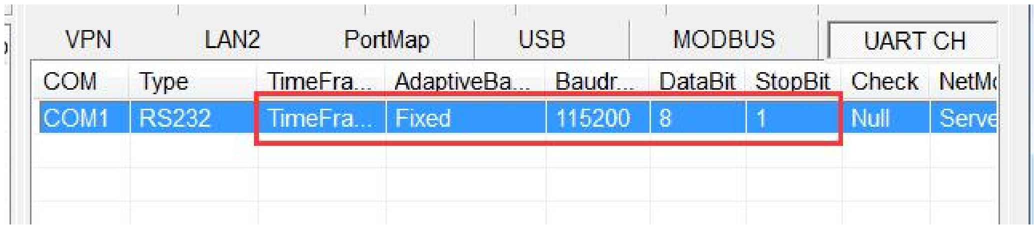
-t: Subscribed message topic

## 3. HJ8300 MQTT Demo Instructions

In the demo, we use socket to communicate with HJ8300 COM1, MODBUS-RTU to read data from COM1, then compile by json and publish on MQTT server.

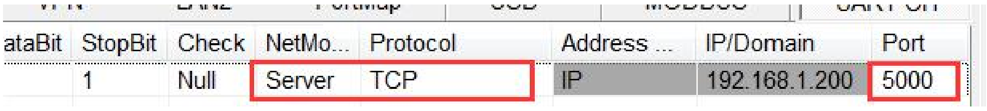
If you use socket to communicate with HJ8300 COM, you need to configure HJ8300 COM1 as below:

Serial port mode uses “Fixed” Baud-rate.



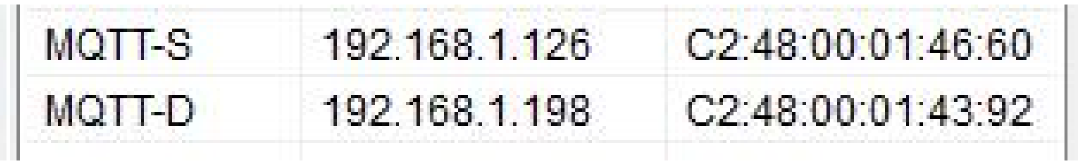
| COM | Type | TimeFra... | AdaptiveBa... | Baudr... | DataBit | StopBit | Check | NetM |
|---|---|---|---|---|---|---|---|---|
| COM1 | RS232 | TimeFra... | Fixed | 115200 | 8 | 1 | Null | Serve |

Net mode uses “TCP Server” and port is configured as 5000.



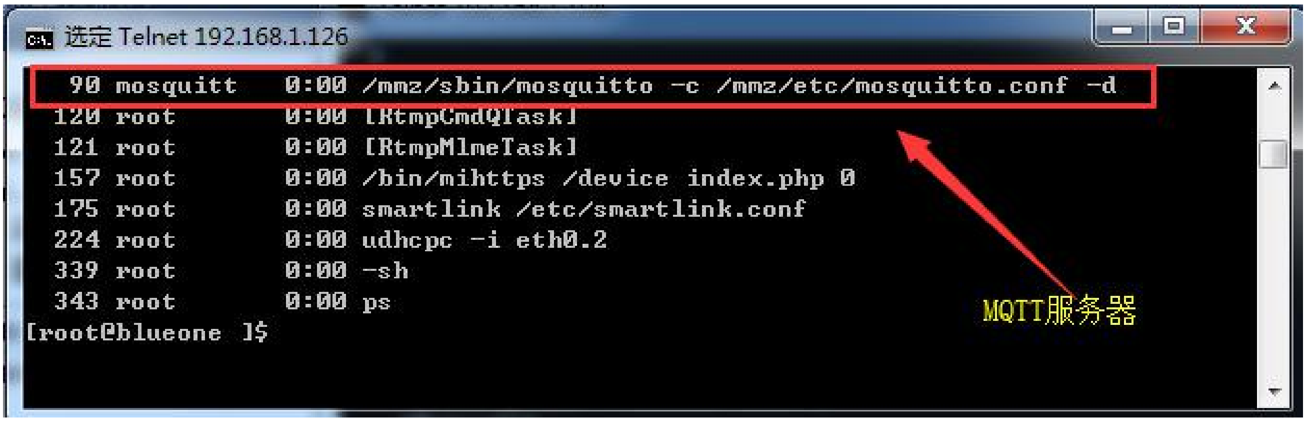
| ataBit | StopBit | Check | NetMo... | Protocol | Address ... | IP/Domain | Port |
|---|---|---|---|---|---|---|---|
| | 1 | Null | Server | TCP | IP | 192.168.1.200 | 5000 |

In the demo, we use two HJ8300 to test:



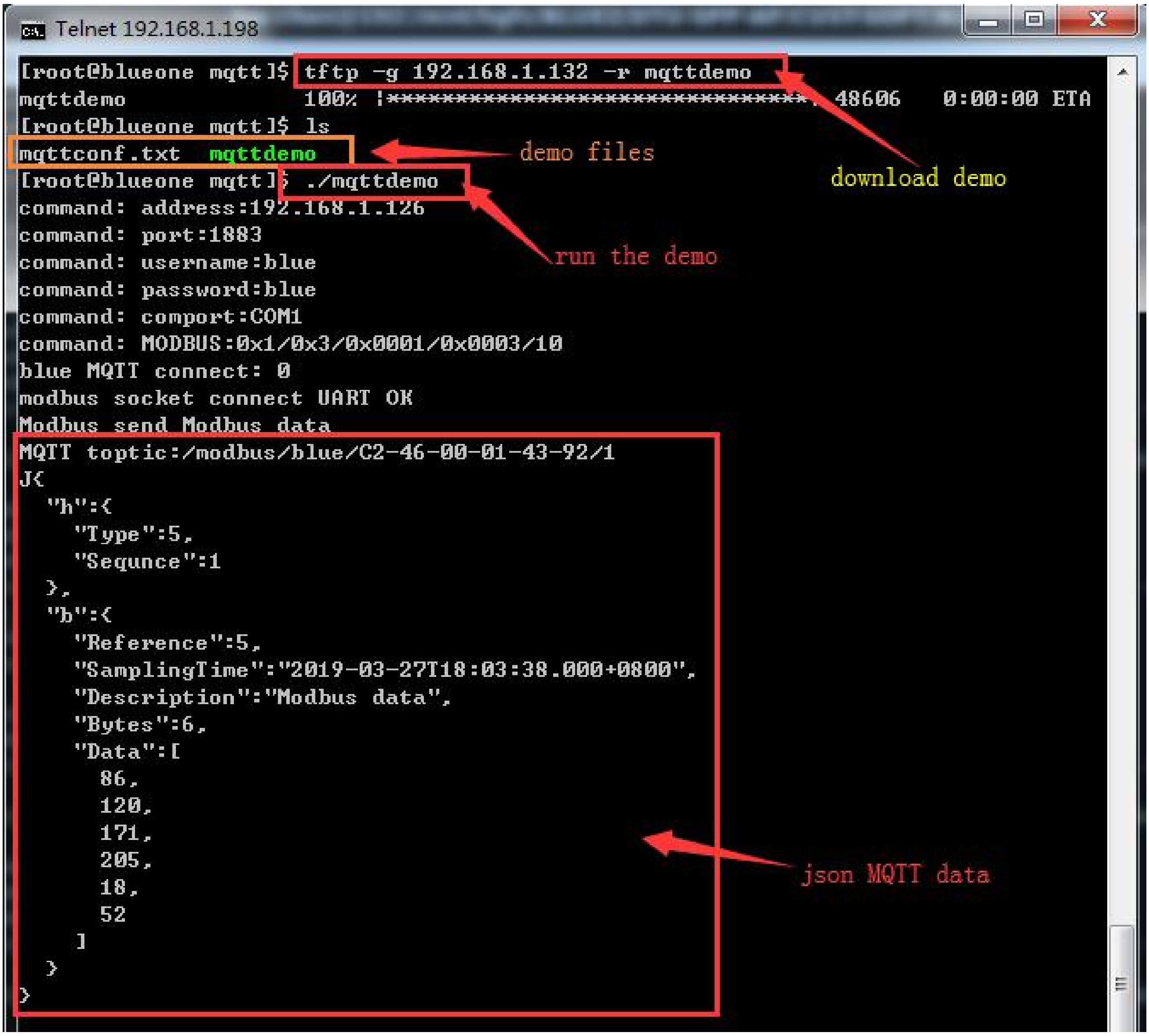
| | | |
|---|---|---|
| MQTT-S | 192.168.1.126 | C2:48:00:01:46:60 |
| MQTT-D | 192.168.1.198 | C2:48:00:01:43:92 |

192.168.1.126 works as MQTT server as below:



```
   90 mosquitt    0:00 /mmz/sbin/mosquitto -c /mmz/etc/mosquitto.conf -d
  120 root        0:00 [RtmpCmdQTask]
  121 root        0:00 [RtmpMlmeTask]
  157 root        0:00 /bin/mihttps /device index.php 0
  175 root        0:00 smartlink /etc/smartlink.conf
  224 root        0:00 udhcpc -i eth0.2
  339 root        0:00 -sh
  343 root        0:00 ps
[root@blueone ]$
```

192.168.1.198 works as MQTT client to start as below:



```
[root@blueone mqtt]$ tftp -g 192.168.1.132 -r mqttdemo
mqttdemo             100% |********************************| 48606    0:00:00 ETA
[root@blueone mqtt]$ ls
mqttconf.txt   mqttdemo
[root@blueone mqtt]$ ./mqttdemo
command: address:192.168.1.126
command: port:1883
command: username:blue
command: password:blue
command: comport:COM1
command: MODBUS:0x1/0x3/0x0001/0x0003/10
blue MQTT connect: 0
modbus socket connect UART OK
Modbus send Modbus data
MQTT toptic:/modbus/blue/C2-46-00-01-43-92/1
J{
  "h":{
    "Type":5,
    "Sequnce":1
  },
  "b":{
    "Reference":5,
    "SamplingTime":"2019-03-27T18:03:38.000+0800",
    "Description":"Modbus data",
    "Bytes":6,
    "Data":[
      86,
      120,
      171,
      205,
      18,
      52
    ]
  }
}
```

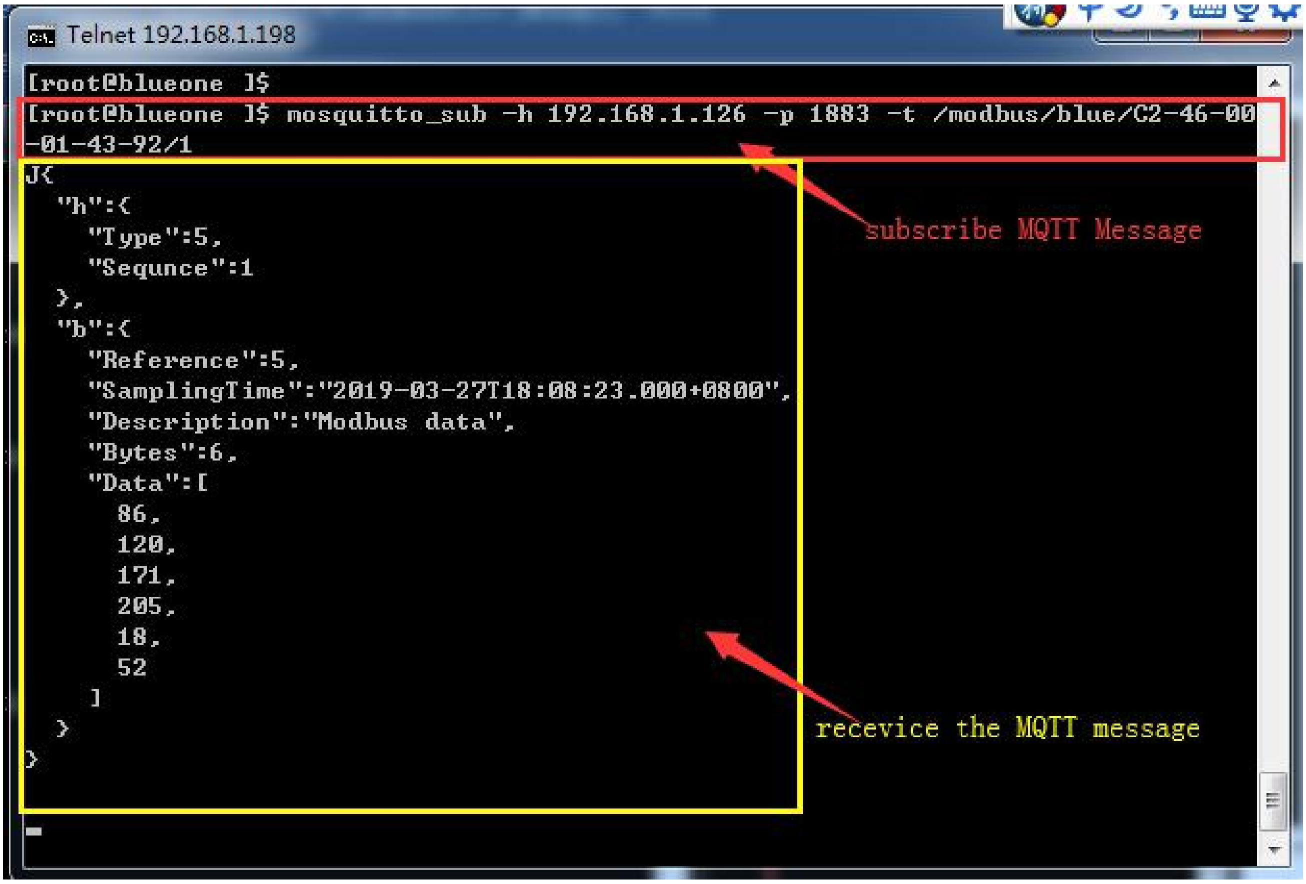
demo files

download demo

run the demo

json MQTT data

Start a subscribed client from 192.168.1.198 HJ8300:



```
[root@blueone ]$
[root@blueone ]$ mosquitto_sub -h 192.168.1.126 -p 1883 -t /modbus/blue/C2-46-00
-01-43-92/1
J{
  "h":{
    "Type":5,
    "Sequnce":1
  },
  "b":{
    "Reference":5,
    "SamplingTime":"2019-03-27T18:08:23.000+0800",
    "Description":"Modbus data",
    "Bytes":6,
    "Data":[
      86,
      120,
      171,
      205,
      18,
      52
    ]
  }
}
```

## 4. HJ8300 DEMO Program Simple Introduction:

### Configuration file instruction:

| | |
|---|---|
| address:192.168.1.126 | MQTT server IP address |
| port:1883 | MQTT server port |
| username:blue | MQTT user name |
| password:blue | MQTT user password |
| comport:COM1 | read MODBUS data from HJ8300 COM1 port |
| MODBUS:0x1/0x3/0x0001/0x0003/10 | MODBUS command: |

MODBUS address/MODBUS command/MODBUS start register/MODBUS read numbers

### Main function instruction

Main function is mainly to finish below work:

1 blue_read_net_interface("eth0",etha); read gateway MAC address

2 blue_mqtt_init_from_file() read configuration data from configuration file

3 blue_timer_initilaize start timer

4 blue_mqtt_thread_start start MQTT thread

### blue_mqtt_thread function instruction

Below is the work that MQTT thread finishes

1 Apple MQTT handle mosquitto_new

2 Set call-back function

3 Connect MQTT server mosquitto_connect

4 Construct MQTT message subject mqtt_topic
5 Publish subject mosquitto_subscribe
6 Start MODBUS thread blue_modbus_thread_start
7 MQTT message loop

**blue_modbus_thread function instruction**

1. Connect to HJ8300 COM1 blue_uart_connect（）
2. blue_modbus_thread_polling Regularly send MODBUS read command
3. blue_modbus_recv_uart_data Handle MODBUS received data

**blue_modbus_recv_uart_data function instruction**

1 Read COM1 data from socket
2 Judge MODBUS data integrality
3 Judge MODBUS data CRC checkout
4 Publish MQTT message blue_mqtt_publish_modbus_data（）

**blue_mqtt_publish_modbus_data function instruction**

1 Edit published subject and time
2 Compile data by json according to published data format
3 Publish data