UNIT MQTT

SKU:U129

Description

UNIT MQTT is an Ethernet MQTT communication module with embedded ARM Cortex-M3 processor, UART communication interface (AT command control) and integrated RJ45 adaptive 10/100M network port. It also supports 4 Topics subscription with extremely low network latency, and is widely used in various industrial automation, security monitoring systems, automatic measurement and control systems, and equipment data on the cloud.

Product Features

- ARM® Cortex®-M3
- RJ45 Ethernet port 10/100Mbps
- Ethernet MQTT communication
- Support 4x Topic subscription
- Low latency
- Development method:
  - AT command, UART: 9600bps default
  - Development platform: UIFlow, Arduino, original PC configuration

Include

- 1x UNIT MQTT
- 1x HY2.0-4P Cable

Applications

- Remote data collection/up to the cloud

Specification

Specifications	Parameters
Module Processor	ARM Cortex-M3
Communication interface	UART: baud 9600bps 8N1 default
MQTT	Support 4x Topic subscription, not support MQTTS
Network port	RJ45 adaptive 10/100M network port
Delay	Delay `10ms
Standby working current	40.4mA
Net weight	22.4g
Gross weight	27.2g
Product size	72 * 26 * 19 mm
Packing size	73 * 35 * 20 mm

PinMap

M5Core	TX(GPIO17)	RX(GPIO16)	VCC	GND
UNIT MQTT	RX	TX	VCC	GND

Examples

Arduino

- MQTT TEST

Video

AT command set

? **Note: Send a carriage return and line feed** \r\n` at the end of each command

Setting AT commands

AT+NETMAC= MAC address``

- Command description: Set the MAC address of the module
- The MAC address must be in the following format: XX-XX-XX-XX-XX-XX

```
//Example
AT+NETMAC=12-34-56-78-90-AB
+NETMAC=OK
```

AT+NETSTATICIP= IP address **,** subnet mask **,** gateway``

- Command description: Set the static IP address of the module (AT+NETSTATICIP= **IP address** , **subnet mask** , **gateway**) ? **Note: When the DHCP function is turned on, the static IP setting will not be enabled.** br **IP address` (text), which means the local IP address, you can use the IP address or domain name`br** subnet mask **(text), must be in IP address format** br``gateway`(text), must be in IP address format

```
//Example
AT+NETSTATICIP="192.168.0.101","255.255.255.0","192.168.0.1"
+NETSTATICIP=OK
```

AT+NETDHCPEN= 1/0``

- Command description: Enable/disable DHCP function **1/0``br** 1: Enable DHCP function **br** 0: Disable DHCP function

```
//Example
AT+NETDHCPEN=1
+NETDHCPEN=OK
```

AT+MQCLIENTID= client name``

- Command description: Set MQTT server **br``client name``br** (text) MQTT client name

```
//示例
AT+MQCLIENTID="CLIENT ID"
+MQCLIENTID=OK
```

AT+MQKEEP= heartbeat duration``

- Command description: Set the length of the heartbeat time for connecting to the MQTT server **br``heartbeat time``br** The time interval for automatically sending PINGREQ packets, unit: second, range 5~300

```
//Example
AT+MQKEEP=500
+MQKEEP=OK
```

AT+MQSUBSCRIBE= subject serial number **,** 1/0 **,** subject name **,** quality of service``

- Command description: Set the topic subscribed to the MQTT server. This module can subscribe 4 topics to the server at the same time. After modifying the subscription topic through this command, the topic will not be subscribed or unsubscribed immediately. It will take effect the next time you connect to MQTT. **br** **Subject Number** Numbers 1-4 **br``1/0** 1 Open the subscription of this topic, 0 Close the subscription of the changed topic, when the parameter is changed to 0, the latter two parameters must be empty **br``topic name** (text) **quality of service** Must be one of 0, 1, 2

```
//Example
AT+MQSUBSCRIBE=1,1,"sub",0
+MQSUBSCRIBE=OK
```

AT+MQSERVER= server address **,** server port``

- Command description: Set the address and port number of the MQTT server **br``server address``br** (text), you can use IP address or domain name **br``server port** number 0~65535

```
//Example
AT+MQSERVER="xxx.mqttserver.com",1883
+MQSERVER=OK
```

AT+MQUSERPWD= user name **,** password``

- Command description: **br** set user name and password **user name** (text) MQTT user name **br``password** (text) MQTT login password name **br** **password** (text) MQTT login password

```
//Example
AT+MQUSERPWD="user","123456"
+MQUSERPWD=OK
```

AT+MQAUTO= 1/0``

- Command description: **br** Set whether to directly connect to the server when powering on. **br``1/0** 0: Do not directly start the connection when powering on, 1: Start the connection directly according to the configured parameters when powering on

```
//Example
AT+MQAUTO=1
+MQAUTO=OK
```

AT+MQRESETCFG

- Command description: Set to restore factory settings related to MQTT **br**

```
//Example
AT+MQRESETCFG
+MQRESETCFG=OK
```

AT+MQPUBLISH= topic , message , quality of service``

- Command description: Publish MQTT message **br``subject** (text), the subject name of the message to be published **br``message** (text), the message to be published, the maximum allowed length of this parameter is 1000 **br** **Quality of Service** QOS, must be 0,1,2, you can choose not to send this parameter, at this time QOS is 0

AT+BAUD= Baud rate``

- Command description: Set the serial port baud rate **br``baud rate** The baud rate can only be the following values/4800/9600/19200/34800/115200/230400

```
//Example
AT+BAUD=115200
+BAUD=OK
```

AT+SAVE

- Command description: Set the save parameters, **After configuring the MQTT**

parameters, you need to execute the SAVE command, and perform a RESET soft reset, the configuration will take effect

```
//Example
AT+SAVE
+SAVE=OK
```

AT+RESET

- Command description: Set the module soft reset, **After configuring the MQTT parameters, you need to execute the SAVE command, and perform a RESET soft reset, the configuration will take effect**

```
//Example
AT+RESET
+RESET=OK
```

Query AT commands

AT

- Command description: Test whether the communication with the module is successful, response: AT

```
//Example
AT
AT
```

AT+MQSUBSCRIBE= subject serial number ?

- Command description: query the subscribed topic, response: +MQSUBSCRIBE=OK: **topic number** , **on/off** , **topic name** , **service quality level``br``topic number** can only be 1~4 One of the **br``on/off** is 1 when the current subscription is on, and 0 means when the subscription is invalid. **br``topic name``service level quality** can only be 0,1,2, one of which means MQTT service QOS0, QOS1, QOS2 in quality

```
//Example
AT+MQSUBSCRIBE=1?
+MQSUBSCRIBE=OK:1,1,"/topic",0
```

AT+MQAUTO?

AT+MQAUTO?

- Command description: Query whether to start the connection directly when powering on, response: +MQAUTO=OK: **on/off** 0: Do not start the connection directly when powering on, 1: Start the connection directly according to the configured parameters when powering on

```
//Example
AT+MQAUTO?
+MQAUTO=OK:1
```

AT+NETMAC?

- Command description: Query the current MAC address of the module, response: +NETMAC=OK: **MAC address** The format is: XX-XX-XX-XX-XX-XX

```
//Example
AT+NETMAC?
+NETMAC=OK:12-34-56-78-90-AB
```

AT+NETSTATICIP?

- Command description: Query the static IP settings of the module, response: +NETSTATICIP=OK: **static IP address** , **subnet mask** , **gateway**

```
//Example
AT+NETSTATICIP?
+NETSTATICIP=OK:"192.168.1.2","255.255.255.0","192.168.1.1"
```

AT+NETDHCPEN?

- Command description: Inquire about enabling/disabling of the DHCP function, response: +NETDHCPEN=OK: **start/disable** 1: Enable the DHCP function, 0: Disable the DHCP function

```
//Example
AT+NETDHCPEN?
+NETDHCPEN=OK:1
```

AT+MQCLIENTID?

- Command description: View the client name of MQTT, response: +MQCLIENTID=OK: **client IP**

```
//Example
AT+MQCLIENTID?
+MQCLIENTID=OK:"clientID"
```

AT+MQSERVER?

- Command description: Query the address and port number of the MQTT server, response: +MQSERVER=OK: **server address** , **server port number**

```
//Example
AT+MQSERVER?

+MQSERVER=OK:"192.168.1.55",1883
```

AT+MQUSERPWD?

- Command description: Query the user name and password of the MQTT server that has been set, response: +MQUSERPWD=OK: **username** , **password**

```
//Example
AT+MQUSERPWD?
+MQUSERPWD=OK:"USER","123456"
```

AT+MQKEEP?

- Command description: Query the length of the heartbeat interval when connecting to the MQTT server, response: +MQKEEP=OK: **heartbeat interval**

```
//Example
AT+MQKEEP?
+MQKEEP=OK:60
```

AT+BAUD?

- Command description: Query the baud rate of the serial port, response: +BAUD=OK: **baud rate** The default baud rate is 9600

```
//Example
AT+BAUD?
+BAUD=OK:9600
```

AT+VERSION?

- Command description: Query the current version number, response: +VERSION=OK: **current firmware version number**

```
//Example
AT+VERSION?
+VERSION=OK:1.0
```

AT+MQSTATUS?

- Command description: Query MQTT connection/connection has been disconnected **br** Response: +MQSTATUS=OK: **connection status**

```
//Example
AT+MQSTATUS?
+MQSTATUS=OK:1
```

AT+NETIP?

- Command description: Query the current actual IP address of the module **br** Response: +NETIP=OK: **IP address** , **subnet mask** , **gateway** , **DNS server address**

```
//Example
AT+NETIP?
+NETIP=OK:"192.168.2.121","255.255.255.0","192.168.2.1","192.168.2.1"
```

AT commands sent actively

```

//The network is not connected
+NETUNCONNECT

//MQTT server is not connected
+MQUNCONNECT

//Successful connection with MQTT server
+MQCONNECT

//Received topic message
```

```
//Received topic message
+MQRECV:`subject`,`length`,`message`
```

Error code returned when the instruction is wrong

```
//Parameter content error
ERROR:Error Param

//The number of parameters does not meet the requirements
ERROR:Error Param Num

//The length of the parameter is too long, and each parameter field requires less than 128 bytes (the maximum length of the published message is 1000 bytes)
ERROR: Parma is too long

//Failed to connect to the network
ERROR:Unconnected

//MQTT server communication error
ERROR: MQTT Connect Fail

//The server did not respond when publishing the message (Qos1 and Qos2)
ERROR: MQTT Publish Fail
```