

Introduction



P4M-440G is an MQTT client module. This module accepts AT commands via the UART port, providing various features for MQTT communication. As a result, users can implement MQTT functionality on their MCU (Microcontroller) or microcomputer using this module.

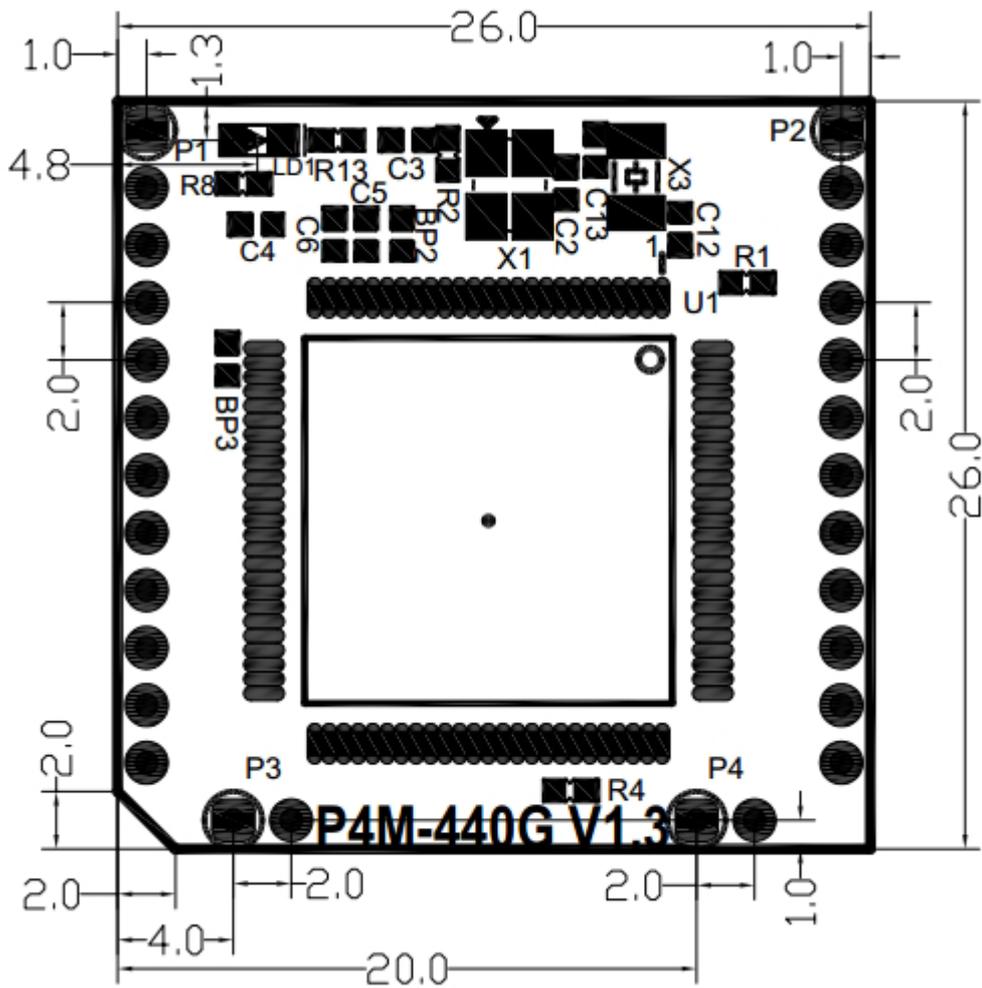
Features

- Equipped with MQTT client functionality
- Supports simple control through AT commands
- 1 x 10/100Mbit Ethernet port
- 1 x UART port (default 19200 bps)
- Supports TLS v1.2
- Provides storage capability for Root CA and client certificates
- Supports PC connection via USB
- Provides dedicated management program (PHPoC Debugger)

Hardware Specifications

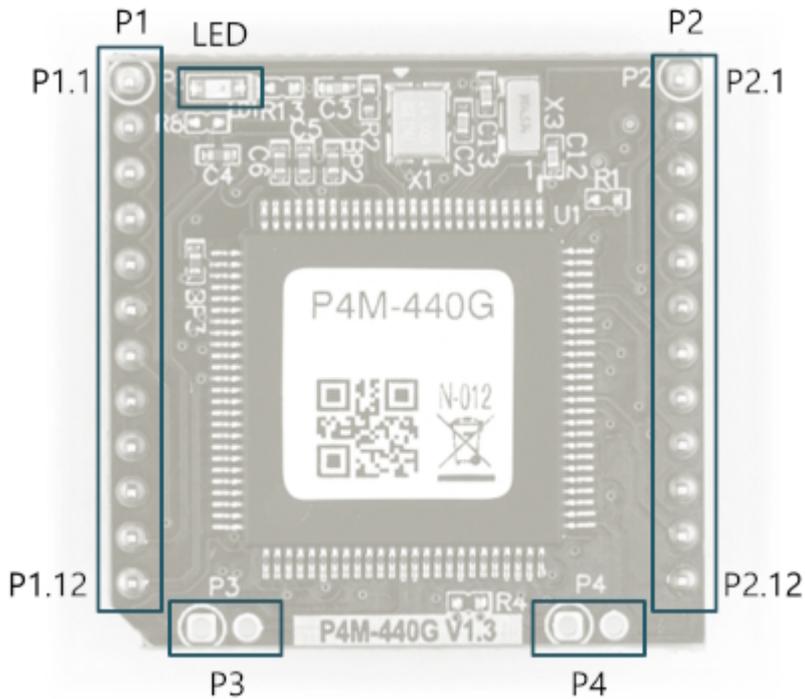
Power	Input	DC 3.3V ($\pm 0.16V$)
	Current Consumption	95mA, typical
Dimension		26mm x 26mm x 9mm
Weight		Approximately 4g
Interface	UART	1 x UART Port (3.3V Level) Baudrate: 1200 ~ 115200 bps Default: 19200 bps, No-Parity, 8-Data bit, 1-Stop bit
	Network	1 x 10/100Mbps Ethernet
	USB	USB Device - PC Connection
Temperature	Operation/Storage	-40°C ~ 85°C
Environment		RoHS Compliant

Dimension



※ Dimensions(unit : mm) may vary according to a method of measurement.

Layout



P4M-440G interfaces with two 12 x 1 pin headers (P1 ~ P2). The pin spacing is 2mm.

P1

Pin#	Name	I/O	Description
P1.1	GND	-	Ground
P1.2	TPTX+	In/Out	Ethernet Transmit +
P1.3	TPTX-	In/Out	Ethernet Transmit -
P1.4	TPRX+	In/Out	Ethernet Receive +
P1.5	TPRX-	In/Out	Ethernet Receive -
P1.6	GND	-	Ground
P1.7	VBUS	In	USB Device VBUS
P1.8	USB_D_D-	In/Out	USB Device Data -
P1.9	USB_D_D+	In/Out	USB Device Data +
P1.10	VBAT	In	Battery Input
P1.11	RST#	In	Reset Input (Active LOW)
P1.12	ISP#	In	ISP Input (Active LOW)

P2

Pin#	Name	I/O	Description
P2.1	+3.3V	-	+3.3V Power Input
P2.2	NSS(0.0)	In/Out	Reserved
P2.3	SCK(0.1)	In/Out	Reserved
P2.4	MISO(0.2)	In/Out	Reserved
P2.5	MOSI(0.3)	In/Out	Reserved
P2.6	U0TX(0.4)	In/Out	UIO 0.4 / UART0 TX
P2.7	U0RX(0.5)	In/Out	UIO 0.5 / UART0 RX
P2.8	SCL(0.6)	In/Out	Reserved

Pin#	Name	I/O	Description
P2.9	SDA(0.7)	In/Out	Reserved
P2.10	U1TX(0.10)	In/Out	UIO 0.10 / UART1 TX
P2.11	U1RX(0.11)	In/Out	UIO 0.11 / UART1 RX
P2.12	GND	-	Ground

LED

There is an STS LED located at the top-left corner of the module. This LED blinks at a 1-second interval.

P3, P4

P3 and P4 are only for the manufacture.

Interface

System

Pin	Description
+3.3V	Input the power supply of 3.3V DC to this pin. Make sure the DC voltage is in the range of 3.15V ~ 3.45V. It is recommended to use a power supply capable of supplying more than 500mA current while stable output of DC 3.3V for the stable operation of the module.
GND	Connect the ground of your main system to this pin. It is recommended that the ground is as wide as possible to shorten the path of return signal.
RESET#	This pin is used to restart the system. If you enter a LOW pulse of 1 millisecond (1ms) or more to this pin, the system will restart. Since this pin is LOW Active, normally it should keep it high state(It is internally pulled up with a 10K ohm resistor.)
ISP#	ISP# pin is used for initialization. Since this pin is LOW Active, normally it should keep it high state(It is internally pulled up with a 10K ohm resistor.)
VBAT	Connect the battery to VBAT so that the built-in RTC time information and back up SRAM contents are not erased when power is not applied. Connect this pin to +3.3V if you do not use any battery. The allowable voltage is DC 1.65V to DC 3.6V.

UART

Pin#	Name	I/O	Description
P2.6	U0TX(0.4)	Out	UIO 0.4 / UART0 TX
P2.7	U0RX(0.5)	In	UIO 0.5 / UART0 RX
P2.10	U1TX(0.10)	Out	UIO 0.10 / UART1 TX
P2.11	U1RX(0.11)	In	UIO 0.11 / UART1 RX

There are two UARTs, each providing RXD and TXD pins.

※ Notice: You can activate only one of the UART on P4M-440G. The default port is UART0, and you can refer to the [AT commands and MQTT configuration](#) for instructions on how to change it.

Ethernet

Pin#	Name	I/O	Description
P1.3	TPTX+	In/Out	Ethernet Transmit +
P1.4	TPTX-	In/Out	Ethernet Transmit -
P1.5	TPRX+	In/Out	Ethernet Receive +
P1.6	TPRX-	In/Out	Ethernet Receive -

P4M-440G provides 10/100Base-TX Ethernet Interface. Note that RJ45 connector is required to use Ethernet. Refer to a circuit diagram of the [Application Circuit Diagram](#) for the connection.

USB Device

Pin#	Name	I/O	Description
P1.7	VBUS	In	USB Device VBUS
P1.8	USB_D_D-	In/Out	USB Device Data -
P1.9	USB_D_D+	In/Out	USB Device Data +

This port must be interfaced for product management and configuration. After connecting this port

and connecting the product with PC via a USB cable, you can access the product using the PHPoC Debugger. Either type B USB connector, mini USB or micro USB connector can be used for this port. Please refer to the [application circuit diagram](#) for the connection method.

※ Notice: The VBUS pin cannot be used for supplying power to P4M-440G.

Software

PHPoC Debugger

PHPoC Debugger is a software used for management and configuration of PHPoC products. You need to install this program on your PC for using PHPoC.

- [PHPoC Debugger Download Page](#)
- [PHPoC Debugger Manual Page](#)

Functions and Features of PHPoC Debugger

- Configuration of P4M-440G
- Monitor resources of P4M-440G
- Upgrade Firmware of P4M-440G
- Upload files from local PC to P4M-440G
- Download files in P4M-440G to local PC
- Edit files stored in P4M-440G
- Support MS Windows O/S

Connection with a PC

To access the product via PHPoC Debugger, a connection with a PC is required.

USB Connection

1. Connect the USB device port of P4M-440G to PC via a USB cable.
2. Run the PHPoC Debugger.
3. Select connected COM PORT and press connect () button.
4. If USB is successfully connected, connect button will be inactivated and disconnect button () will be activated.

Remote Connection

The remote connection is provided. Please refer to [the PHPoC Debugger manual page](#) for details.

Initial Setup and Certificates

IP Address

P4M-440G is initially configured to automatically obtain an IP address. Therefore, in a typical network where IP addresses are assigned automatically, there is no need for initial IP address configuration. However, if you need to use a static (fixed) IP address, you should connect the product to a PC and refer to the [PHPoC Debugger manual](#) for instructions on setting the IP address.

Password

If you set a password for the product, you must enter the password when connecting the product via USB or network. Please refer to [the PHPoC Debugger manual page](#) for details.

Certificates

If you need to store Root CA certificates and client certificates on the module for MQTT broker connection, please refer to the [Managing Certificates](#) section in the PHPoC Debugger manual to store the certificates on the product.

Firmware Upgrade

Firmware can be updated to introduce new features or fix discovered bugs in the product. If the firmware version on your current product is not the latest version, it is advisable to perform an upgrade. Firmware can be upgraded either online through PHPoC Debugger or directly. For detailed instructions on firmware upgrades, please refer to the [Firmware Upgrade](#) section in the PHPoC Debugger manual.

Reset

Settings Reset

Settings Reset makes all the settings of your PHPoC products to factory default.

- Settings Reset Procedure

Step	Action	Product State	STS LED
1	Input LOW signal to ISP# shortly (less than 1 second)	Initialization mode	On
2	Input LOW signal again over 5 seconds	Preparing initialization	Rapidly blinks
3	Check if the STS LED is turned OFF	Initialization ready	Off
4	Release the LOW input right after the STS is OFF.(※ If you don't release the button within 2 seconds, the state goes back to the step 2)	Progressing initialization	On
5	Rebooting automatically	Initial state	Off

Device Initialization

Device Initialization makes all the settings including saved password of your PHPoC products to factory default. Furthermore, all files stored in flash memory are deleted as well as certificate. Therefore, it is highly recommended to back up the files stored in the product's file system and the certificates before performing device initialization.

※ Refer to the [Device Initialization](#) page of PHPoC Debugger's Manual for details.

MQTT Usage

This document serves as a guide on how to use the MQTT functionality of the P4M-440G and does not cover detailed information about the MQTT protocol. For in-depth details about the protocol, please refer to the MQTT standard documentation available on the [MQTT website](#).

File System

To use the MQTT functionality of P4M-440G, the following files are required:

File Name	Description
p4x_atc_x.x.poc	ATC PXM - The capability to exchange AT commands for MQTT communication via UART
p4x_mqtt_x.x.poc	MQTT PXM - Functionality for MQTT communication
phpoc.ini	General configuration file
pxm.ini	PXM configuration file

These files are stored in the P4M-440G's file system by default, allowing users to immediately utilize the MQTT functionality.

AT Commands

All MQTT functionalities of the P4M-440G are executed through AT commands. Therefore, it is essential for users to be familiar with the usage of AT commands.

Basic Format

AT commands and responses are composed of ASCII strings and are case-insensitive. Every AT command always starts with 'AT' and ends with '<CR>'.

※ <CR>: Carriage Return, 0x0d

Command List: General

Command	Description	Usage
E	Command Echo OFF/ON	OFF: ATE0<CR>, ON: ATE1<CR>
Z	Reset & Reboot	ATZ<CR>

Command List: MQTT

Command	Description
+MCT	Setting Timeout
+MCCF	Setting Connection Flags
+MCCP	Setting Connection Payload
+MCB	Setting QoS
+MNC	Connecting to a Broker
+MND	Disconnecting from a Broker
+MNG	Reading Messages
+MBT	Setting Topic Buffer
+MBM	Setting Message Buffer
+MBP	PUBLISH
+MBS	SUBSCRIBE
+MBU	UNSUBSCRIBE
+MQNS	Getting Status
+MQNU	Getting the number of unread messages
+MQNR	Getting the size of a received message
+MQC	Getting the client ID
+MCO	Setting the short response mode

Command Responses

Response	Description
OK	get ready or a command executed successfully
ERROR	a command executed error
otherwise	a response corresponding to each command

Device Connection

1. LAN connection and configuration

Connect P4M-440G to an internet-enabled network and, if necessary, configure the IP address.

2. UART connection and configuration

Connect the user's MCU or microcontroller to P4M-440G's UART0 port and configure it as follows:

- 19200 bps, No-parity, 8-data bit, 1-stop bit

3. Ready for AT command input

Prepare the user's MCU or microcontroller to input AT commands via P4M-440G's UART.

Topic and Message Buffer

By storing topics and messages in buffers and accessing them using buffer IDs, you can exchange MQTT messages more efficiently. You can use a total of four buffers (with IDs 0 to 3) for topic buffers and message buffers, respectively.

Setting Topic Buffers: +MBT

- Command Format

```
AT+MBT=<Topic Buffer ID>,"<Topic>"
```

- An example

```
at+mbt=0,"test/test_topic_0"  
OK  
at+mbt=1,"test/test_topic_1"  
OK  
at+mbt=2,"test/test_topic_2"  
OK  
at+mbt=3,"test/test_topic_3"  
OK
```

Setting Message Buffers: +MBM

- Command Format

```
AT+MBM=<Message Buffer ID>,"<Message>"
```

- An example

```
at+mbm=0,"message 0"  
OK  
at+mbm=1,"message 1"  
OK  
at+mbm=2,"message 2"  
OK  
at+mbm=3,"message 3"  
OK
```

Timeout

Timeout settings provide five parameters, and the unit of the settings is in seconds.

Timeout	Description	Default Value	Order
CONNECT	Timeout for MQTT CONNECT request	4	1
PUBLISH	Timeout for MQTT PUBLISH request	2	2
SUBSCRIBE	Timeout for MQTT SUBSCRIBE / UNSUBSCRIBE request	2	3
PINGREQ	Timeout for MQTT PINGREQ packet	2	4
Keep Alive	Timeout for Keep Alive (PINGREQ intervals)	30	5

Timeout Setting: +MCT

- Command Format

Each timeout value is listed by separating them with commas, without spaces, to the right of the equal sign in the +MCT command. It is essential to list them in the order of their settings. Each value can be omitted, and if omitted, the previous setting value will be retained. When omitting values, you can omit both the commas and the values on the right side based on the value you want to set. However, for the left items of the setting value, you can omit only the values, excluding the commas.

```
AT+MCT=<CONNECT>,<PUBLISH>,<SUBSCRIBE>,<PINGREQ>,<Keep Alive>
```

Examples

- Setting each timeout value to 4, 2, 2, 2, and 30 seconds

```
at+mct=4,2,2,2,30
OK
```

- Setting CONNECT timeout to 4 seconds

```
at+mct=4
OK
```

- Setting PUBLISH timeout to 2 seconds

```
at+mct=,2
OK
```

- Setting SUBSCRIBE timeout to 2 seconds

```
at+mct=,,2  
OK
```

- Setting PINGREQ timeout to 2 seconds

```
at+mct=,,,2  
OK
```

- Setting Keep Alive timeout to 30 seconds

```
at+mct=,,,,30  
OK
```

Connection Flags

Three flags are provided for MQTT connection configuration.

Flag	Value	Default Value	Order
Clean Session	0 or 1	1	1
Will QoS	0 or 1	0	2
Will Retain	0 or 1	0	3

Connection Flag Setting: +MCCF

Command Format

Each timeout value is listed by separating them with commas, without spaces, to the right of the equal sign in the +MCCF command. It is essential to list them in the order of their settings. Each value can be omitted, and if omitted, the previous setting value will be retained. When omitting values, you can omit both the commas and the values on the right side based on the value you want to set. However, for the left items of the setting value, you can omit only the values, excluding the commas.

```
AT+MCCF=<Clean Session>,<Will QoS>,<Will Retain>
```

Examples

- Setting each flag to 1, 0 and 0

```
at+mccf=1,0,0
OK
```

- Setting Clean Session to 1

```
at+mccf=1
OK
```

- Setting Will QoS to 0

```
at+mccf=,0
OK
```

- Setting Will Retain to 0

```
at+mccf=,,0
```

OK

Connection Payload

Five payload settings are provided for MQTT connection.

Payload	Description	Order	Note
Client ID	Client ID	1	-
Will Topic	Topic to store will message	2	-
Will Message	Will message	3	-
User Name	User name for authentication	4	Max. 128 Bytes
Password	Password for authentication	5	Max. 256 Bytes

Setting Connection Payload: +MCCP

Command Format

Each timeout value is listed by separating them with commas, without spaces, to the right of the equal sign in the +MCCP command. It is essential to list them in the order of their settings. Each value can be omitted, and if omitted, the previous setting value will be retained. When omitting values, you can omit both the commas and the values on the right side based on the value you want to set. However, for the left items of the setting value, you can omit only the values, excluding the commas.

```
AT+MCCP="<Client ID>","<Will Topic>","<Will Message>","<User Name>","<Password>"
```

Examples

- Setting a Client ID

```
at+mccp="test_client_id"
OK
```

- Setting a will message

```
at+mccp="test/will_topic","test will message"
OK
```

- Setting a will message (Using a topic buffer and a message buffer)

```
at+mbt=0,"test/topic_0"
OK
at+mbm=0,"message 0"
OK
at+mccp=,0,0
```

```
OK
```

- Setting a user name and a password

```
at+mccp=,,,"test_user","test_password"  
OK
```

QoS

Three settings for Quality of Service (QoS) are provided.

Parameter	Description	Values	Order
PUBLISH QoS	QoS field of PUBLISH packet	0, 1	1
RETAIN	RETAIN flag of PUBLISH packet	0, 1	2
Requested QoS	Requested QoS field of SUBSCRIBE payload	0, 1	3

Setting QoS: +MCB

- Command Format

Each timeout value is listed by separating them with commas, without spaces, to the right of the equal sign in the +MCB command. It is essential to list them in the order of their settings. Each value can be omitted, and if omitted, the previous setting value will be retained. When omitting values, you can omit both the commas and the values on the right side based on the value you want to set. However, for the left items of the setting value, you can omit only the values, excluding the commas.

```
AT+MCB=<PUBLISH QoS>,<RETAIN>,<Requested QoS>
```

Examples

- Setting all parameters to 0

```
at+mcb=0,0,0
OK
```

- Setting PUBLISH QoS to 0

```
at+mcb=0
OK
```

- Setting RETAIN flag to 0

```
at+mcb=,0
OK
```

- Setting requested QoS to 0

```
at+mcb=,,0  
OK
```

Connecting to Broker

P4M-440G can request a connection to a MQTT broker as an MQTT client.

Connect to a Broker: +MNC

- Command Format

```
AT+MNC="<Broker's hostname / IP address>"
AT+MNC="<Broker's hostname / IP address>",<Port number>
AT+MNC="<Broker's hostname / IP address>",<Port number>,<Protocol>
```

In some cases, you can omit the port number or protocol. When omitted, the default values will be used.

Parameter	Values	Default Value
Port number	0 ~ 65535	1883
Protocol	"tcp", "tls"	"tcp"

※ Notice: If you specify the port number as 8883 and do not specify the protocol, "tls" will be used as the protocol instead of the default "tcp".

- An example

```
at+mnc="203.0.113.0"
OK
```

- An example: Specifying a port number

```
at+mnc="203.0.113.0",8883
OK
```

- An example: Specifying both a port number and a protocol with a hostname

```
at+mnc="example.com",8884,"tls"
OK
```

Check Connection Status: +MQNS

- Command Format

AT+MQNS

- Response Code

Code	Description
0	Not conencted
4	Connected
5	Connected + Wait for a response (PUBACK / SUBACK / UNSUBACK)

- An example

```
at+mnc="203.0.113.0"
OK
at+mqns
+MQNS:4
OK
```

Disconnect from a Broker: +MND

- Command Format

AT+MND

- An example

```
at+mqns
+MQNS:4
OK
at+mnd
OK
at+mqns
+MQNS:0
OK
```

SUBSCRIBE

P4M-440G, as an MQTT client, can request to subscribe to a specific topic on the MQTT broker or request to unsubscribe from topics it has subscribed to.

SUBSCRIBE Request: +MBS

- Command Format

```
AT+MBS="<Topic>"  
AT+MBS=<An ID of a Topic Buffer>
```

- An example

```
at+mbs="test/test_topic"  
OK  
at+mqns  
+MQNS:4  
OK
```

- An example: Using a topic buffer

```
at+mbt=0,"test/test_topic"  
OK  
at+mbs=0  
OK  
at+mqns  
+MQNS:4  
OK
```

UNSUBSCRIBE Request: +MBU

- Command Format

```
AT+MBU="<Topic>"  
AT+MBU=<An ID of a Topic Buffer>
```

- An example

```
at+mbu="test/test_topic"  
OK
```

```
at+mqns  
+MQNS:4  
OK
```

- An example: Using a topic buffer

```
at+mbt=0,"test/test_topic"  
OK  
at+mbu=0  
OK  
at+mqns  
+MQNS:4  
OK
```

PUBLISH

P4M-440G, as an MQTT client, can request to publish messages to a MQTT broker.

PUBLISH Request: +MBP

- Command Format

```
AT+MBP="<Topic>","<Message>"
AT+MBP="<Topic>",<An ID of a Message Buffer>
AT+MBP=<An ID of a Topic Buffer>,"<Message>"
AT+MBP=<An ID of a Topic Buffer>,<An ID of a Message Buffer>
```

- An example

```
at+mbp="test/test_topic","test message"
OK
at+mqns
+MQNS:4
OK
```

- An example: Using both a topic buffer and a message buffer

```
at+mbs=0,"test/topic_0"
OK
at+mbm=0,"test message"
OK
at+mbp=0,0
OK
at+mqns
+MQNS:4
OK
```

Reading Messages

If there are messages published to the subscribed topics, those messages will be received and can be read. In the case of multiple received messages, you can only read the last received message.

Reading a Message: +MNG

- Command Format

```
AT+MNG="<Topic>"  
AT+MNG=<An ID of Topic Buffer>
```

- An example

```
at+mng="test/test_topic"  
+MNG:12,test message  
OK
```

- An example: Using a topic buffer

```
at+mbt=0,"test/test_topic"  
OK  
at+mng=0  
+MNG:12,test message  
OK
```

Getting the Number of Unread Messages: +MQNU

- Command Format

```
AT+MQNU  
AT+MQNU="<Topic>"  
AT+MQNU=<An ID of a Topic Buffer>
```

If you omit the topic or topic buffer ID, it will check all subscribed topics.

- An example

```
at+mqnu="test/test_topic"  
+MQNU:1
```

```
OK
```

- An example: Using a topic buffer

```
at+mbt=0,"test/test_topic"  
OK  
at+mqnu=0  
+MQNU:1  
OK
```

Getting the Size of a Received Message: +MQNR

- Command Format

```
AT+MQNR="<Topic> "  
AT+MQNR=<An ID of a Topic Buffer>
```

- An example

```
at+mqnr="test/test_topic"  
+MQNU:12  
OK
```

- An example: Using a topic buffer

```
at+mbt=0,"test/test_topic"  
OK  
at+mqnr=0  
+MQNU:12  
OK
```

Reading the Client ID

You can read the client ID of the device.

Reading the Client ID: +MQC

- Command Format

```
AT+MQC="clientid"
```

- An example

```
at+mqc="clientid"  
+MQC:USER-SCLI-ENTI-DCHK  
OK
```

- An example: Setting with a client ID

```
at+mccp="user_client_id"  
OK  
at+mqc="clientid"  
+MQC:user_client_id  
OK
```

※ Notice: All devices come with unique client IDs by default. If the user does not specify a client ID, this unique value will be used.

Setting Short Response

You can set short response for AT commands.

Setting short response: +MCO

- Command Format

```
AT+MCO=<Value>
```

Values	Descriptoin
0	Disable Short Response (Default)
1	Enable Short Response

- An exapmle

```
at+mco=0
OK
at+mqns
+MQNS:4
OK
at+mco=1
OK
at+mqns
+4
OK
```

Activating PHPoC Engine

P4M-440G is a programmable module, but this feature is initially disabled. To enable user programming on P4M-440G, you need to activate this setting, which can be done in the basic environment file (phpoc.ini) in the P4M-440G's file system.

Setting Procedures

1. Access P4M-440G's file system using PHPoC Debugger.
2. In PHPoC Debugger's file system view, select the phpoc.ini file.
3. Change the "init_php" item in the phpoc.ini file from "Off" to "On" as follows:

phpoc.ini (original)	phpoc.ini (modified)
init_php = Off	init_php = On

4. Upload the modified file system.
5. Reboot the device.

Caution

Enabling the user programming feature provides access to several interfaces and additional functions. However, using too much memory for programming may impact the operation of the MQTT feature. Therefore, it is recommended to enable it only when necessary.

※ For detailed information on P4M-440G with the user programming feature enabled, please refer to the [P4M-440G User Manual](#).

If you modify or delete files stored in the P4M-440G's file system, MQTT functionality may not work correctly. In such cases, you should use PHPoC Debugger to re-upload the ATC/MQTT package to the P4M-440G's file system. You can download the ATC/MQTT package from the [online repository](#).

Advanced AT Commands

You can read or change IP address-related settings for P4M-440G using AT commands.

- IP Address-related Settings

Item	Descriptoin
dhcp	Obtain an IP address automatically(1: Enable, 0: Disable)
ipaddr	IP Address
netmask	Subnet Mask
gwaddr	Gateway IP Address
nsaddr	DNS IP Address

These settings are stored in a buffer called "envs." Therefore, you need to use AT commands related to "envs" to access each item.

Reading an envs item: +ESR

You can use this command to read a specific item from "envs."

- Command Format

```
AT+ESR="<Item>"
```

- An example: Reading the IP address

```
at+esr="ipaddr"
+ESR:203.0.113.0
OK
```

Loading the envs: +EBL

- Command Format

```
AT+EBL="envs"
```

- An example

```
at+ebl="envs"
OK
```

Storing the envs: +EBS

- Command Format

```
AT+EBS="envs",0xa5c3
```

- An example

```
at+ebs="envs",0xa5c3  
OK
```

Modifying an envs item: +EBW

- Command Format

```
AT+EBW="<Item>","<Value>"  
AT+EBW="<Item>",<Value>
```

- An example: Enabling the DHCP

```
at+ebl="envs"  
OK  
at+ebw="dhcp",1  
OK  
at+ebs="envs",0xa5c3  
OK
```

- An example: Using a static IP address (203.0.113.0)

```
at+ebl="envs"  
OK  
at+ebw="dhcp",0  
OK  
at+ebw="ipaddr","203.0.113.0"  
OK  
at+ebw="netmask","255.255.255.0"  
OK  
at+ebs="envs",0xa5c3  
OK
```