Phc2Mqtt E2E Config ed02

1 Terms and abbreviations

LAN	Local Area Network, is the network form typically used at home, supporting Ethernet connections.
WLAN	Wireless LAN, is the same as LAN but then the wireless variant, also called Wifi.
IP	Internet Protocol, the basis networking protocol that runs over Ethernet (wired) or Wifi (wireless).
TCP	Transmission Control Protocol, runs on top of IP and is a connection oriented protocol.
HTTP	HyperText Transfer Protocol, runs on top of TCP in a request-reponse model.
XMLRPC	XML encoded Remote Procedure Call, this protocol is used to execute remote commands/configuration. The commands are encoded in XML and the request is sent over HTTP.
SNTP	Simple Network Time Protocol, protocol used to obtain time from the Internet.
MQTT	A compact telemetry protocol typically used in IoT applications, runs over TCP.
PCB	Printed Circuit Board, a non-conductive material with copper patterns on 1 or 2 sides to interconnect electronic components.
ESP32 DevkitC	The ESP32 dual core module that is used for Phc2Mqtt, we use the WROVER version with 4Mb external memory.
AP mode	Access Point mode, the module acts as an access point to which you can connect with a Wifi enabled PC.
STA mode	Station mode, the module connects to a WLAN AP as a station, you can access the module via the WLAN network.
OTA	Over-The-Air, refers to firmware upgrade by simple file upload, as such you don't need to open the module or use specialized hardware.
PHC	PEHA Home Control, a domotic system created by the former company PEHA.
STM	Steering module, the central processor in a PHC system.
Ccmd	Compound-command, the command format as used by xWRC to send commands to a PHC system.
3MOD	3 Module, indicates the physical size of a storage container mounted onto a (DIN) rail, where 1 MOD = 18mm.
RJ11	Typical connector used in low speed communication equipment.
RS232	A low speed single ended communication bus with 2 parties.
RS4845	A low speed differential commication bus with upto 32 parties.

2 Introduction

Phc2Mqtt is the combination of a hardware module (PCB + components) and operating software known as firmware. For simplicity we will furthermore refer to the Phc2Mqtt combination as the **module**.

Purpose of the module is to make a PHC domotic system accessible via MQTT, both for reporting activity on the PHC module bus (RS485) and for sending commands to the PHC system.

A PHC domotic system is made up from a steering module (STM) that is the master, and 1 upto 32 PHC modules connected over an RS485 PHC module bus, communicating via a proprietary binary protocol.

The module is chained into the PHC module bus using two 6-wire cables with RJ11 connectors, and it will monitor messages exchanged between PHC modules and STM.

Optionally the module can be connected to the management interface of the STM to send commands to it: via an RS232 connection in case of STMv1/STMv2 or via an IP connection in case of an STMv3.

2.1 The module

Below is the layout of the module PCB, it fits in a 3MOD DIN-rail housing with transparent top cover so you can see the LEDs on the module, relevant items are:

CN1/CN2 are the PHC module bus connectors

X2 is an optional 24V power connector, not used

ESP32-DevkitC with a red power LED, this is the CPU

Pushbutton at the bottom labeled 'WIFI'

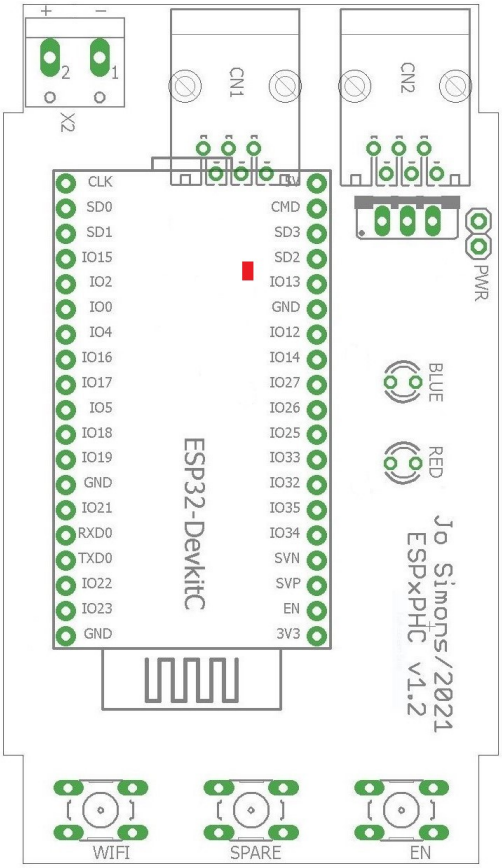
Pushbutton at the bottom labeled 'SPARE', not used

Pushbutton at the bottom labeled 'EN'

2-pin header at top/right labeled 'PWR'

Blue module LED on the right labeled 'BLUE'

Red module LED on the right labeled 'RED'



2.2 Powering

In normal operation the module is connected to your PHC system with two 6-wire cables with RJ11 connectors, and the module will take it's power from the PHC system.

However if you ever plan to open the module and connect the used ESP32-DevkitC to your computer with a USB cable, you MUST first remove the cap on the **PWR** connector on the right/top side of the module.

WARNING: Failing to remove the cap can and will most likely damage the module or your PC !!!

1.3 End-to-end architecture

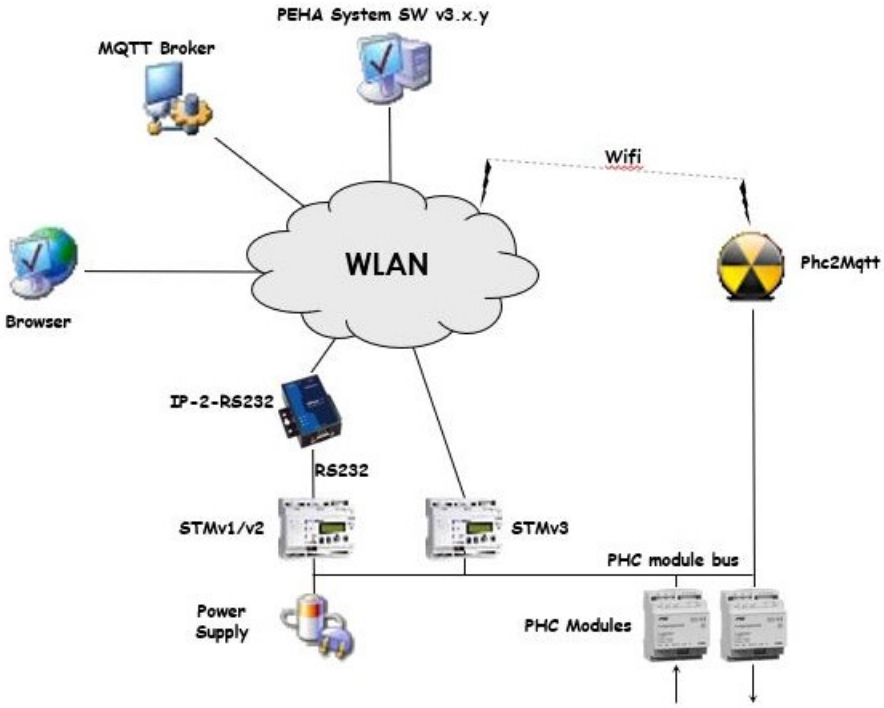
Following drawing gives an overview of the end-to-end system architecture.

The module is connected to the PHC module bus, first of all to obtain it's needed power, and second to monitor all activity on the PHC module bus.

The intention is that the module connects to the wireless LAN (WLAN) such that the PEHA System Software v3.x.y and the browser have access to it for configuration and reporting purposes.

It is via the same WLAN access that the module will connect to a MQTT Broker to provide it's services to a wider community.

Finally an optional IP-wise connection to the STM management interface is made through the WLAN. In case of the older STMv1/v2 you also need an IP-2-RS232 convertor like MOXA NPort 5110 and the sorts. In case of the STMv3 you can directly use it's network connection.



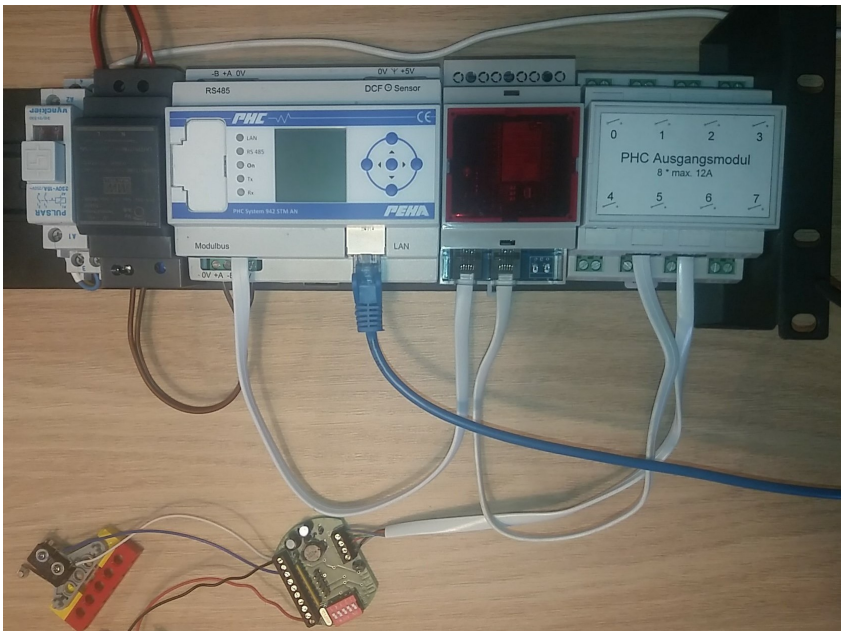
3 First time usage of the module

The first time you want to start using the module you will need to hook it up to your PHC system and your Wifi network.

3.1 Connecting the module to your PHC system

- Turn off your PHC system and make sure it is no longer powered.
- Add the module in the PHC module bus chain by removing a 6-wire cable from a PHC module/STM and plugging it in CN1 on the module. Then plug a new 6-wire cable with 2 RJ11 connectors in CN2 of the module and back in the PHC module/STM to complete the chain.

Make some space and place the module onto the DIN rail such that it is mechanically stable.

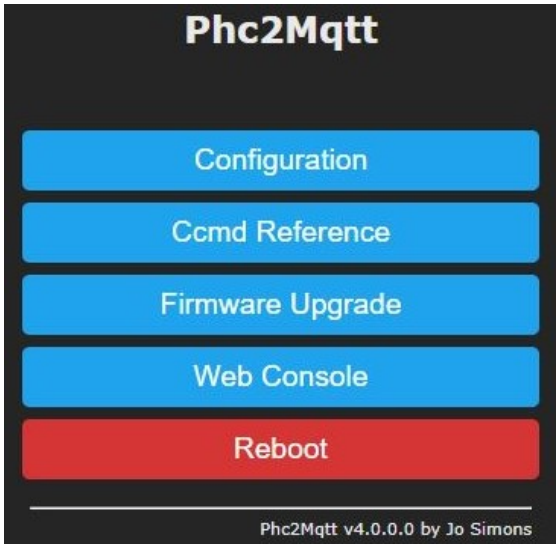


- Power on your PHC system.
- You will see the red power LED on the ESP32-DevkitC turn on.
- The module will start up an turn on both red and blue module LEDs, then boots into Access Point (AP) mode, the blue module LED will turn off and the red module LED will remain on to indicate **AP mode**.

WARNING: The module draws a fair amount of current (120~140mA) when in AP mode, this may overload the power supply of your PHC system and cause unpredictable behaviour of the module and/or the PHC system. If this is the case, it is advised to power the module with a USB power adapter after removing the cap from the PWR connector as explained in section 2.2.

3.2 Linking the module in AP mode to your Wifi network

- Use a Wifi enabled PC to connect to the module by looking for a network called '**p2m_xxxxxx**', the x's represent the lower 3 bytes of the module's MAC address.
- Once connected to the module's AP, use a browser to access it on **192.168.4.1**, you will get below **Main Menu** in response.



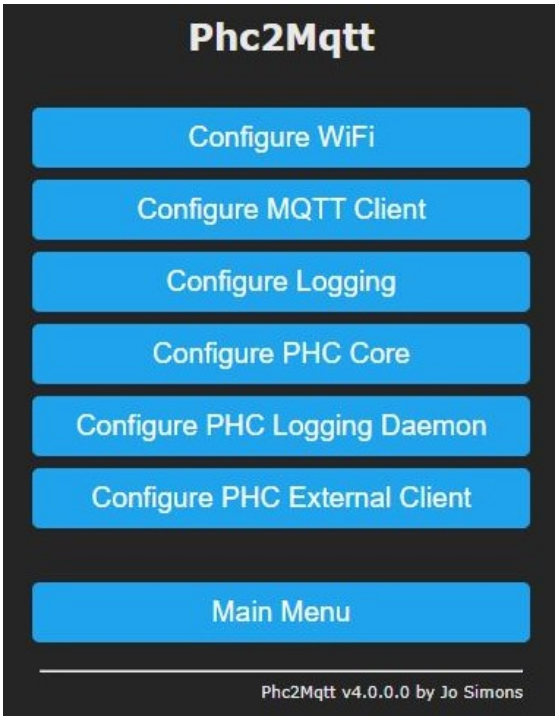
- In **Main Menu** select **Configuration** then **Configure Wifi** and configure the Wifi settings as described in section '**4.1 Configuring Wifi**'.
- After you configured the Wifi settings, reboot the module via **Main Menu** -> **Reboot** and confirming the action when prompted by the browser (preferred), alternatively you can press the **EN** button (less preferred).
- The module starts up and lights both module LEDs, then connects to your Wifi network, after which both LEDs will turn off, the module is now in Station (STA) mode.
- The module will try to obtain date/time from the Internet by means of SNTP as it has no battery powered RTC to keep time during reboots and power downs.
- If you entered a wrong Remote AP SSID/Remote AP Password, Wifi connection will fail and the module will alternately flash the red and blue module LEDs for 2 seconds to indicate this failure. The module will reboot and retry connection to the specified Wifi network, after 4 failures it will revert to AP mode and the red module LED turns on. Restart above procedure and correct settings.

3.3 Accessing the module in STA mode linked to Wifi network

- Now the module is connected to your Wifi network and you need to find out it's IP address.
- We suggest you log on to your Wifi router to consult the list of connected clients. Look for an entry showing 'Espressif', enter the shown IP address in a web browser window, you should see the **Main Menu**.
- Next you assign the module a fixed IP address in your router menu and save the settings.
- Now reboot the module, your router will assign it a fixed IP address each time the module starts up.
- Use a web browser to access the module on the fixed IP address.

4 Configuring module functionality

- In **Main Menu** select **Configuration**, you will see the configuration menu as shown to the right.



- Select any button to enter the appropriate configuration page.

4.1 Configuring Wifi

- **Remote AP SSID** (network name) and **Remote AP Password**: The module will use these credentials to connect in STA mode to the Wifi network.

When you leave these fields empty, the module will boot into AP mode.

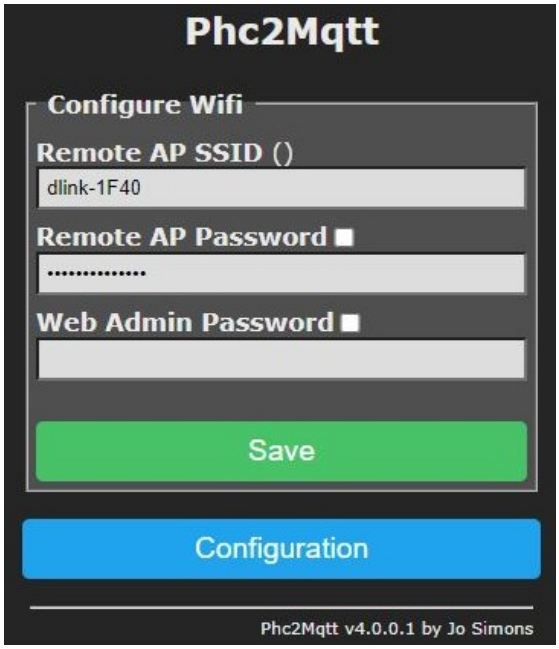
- **Web Admin Password**: You can specify a password to protect web access to the module, in this case you need to logon to the module when prompted by the browser, the username is 'admin' completed with the specified password.

If you forgot the password and locked yourself out of the module you can recover it by pressing and holding the **WIFI** button, then press/release the **EN** button. The module will reboot, it will flash both blue/red module LEDs simultaneous for 2 seconds to confirm the pressed **WIFI** button, you can release the **WIFI** button by now.

The module will boot in AP mode and turn the red module LED on to indicate this.

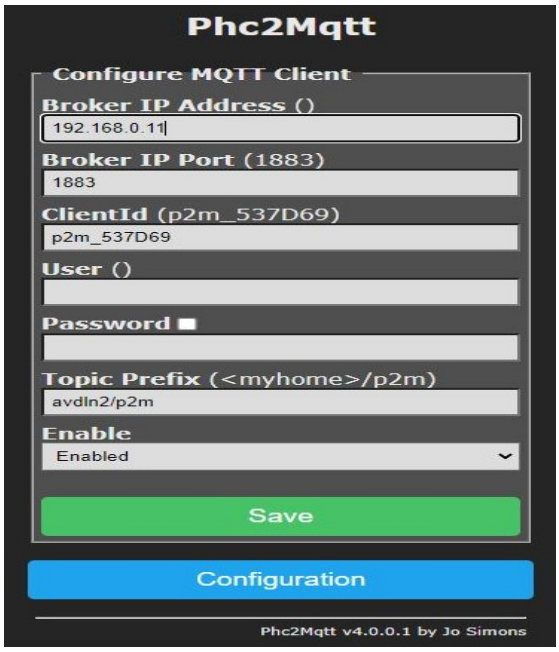
You can now access the module with a web browser without password. Change/remove the password as needed.

- **Save**: Press this button to save settings, then reboot the module to let changes take effect.



4.2 Configuring MQTT Client

The MQTT client is the interface to the MQTT broker (server), it will connect to the broker and subscribe to a topic for receiving commands, it will also publish notifications to a topic.



- **Broker IP Address** and **Broker IP Port**: Enter the MQTT broker IP address and port (default 1883), these fields are mandatory.
- **ClientId**: A unique name across all clients that connect to the broker, maximum 24 characters, mandatory.

If this is not unique you will see that the MQTT client looses it's connection from time to time.

- **User** and **Password**: Optional, if needed to connect to the broker.
- **Topic-Prefix**: Used in the communication with the MQTT broker, we propose to use <myhouse>/p2m, where you replace <myhouse> with a mnemonic for your house/location.

Once the MQTT client is connected to the broker, it will subscribe to <topic-prefix>/#

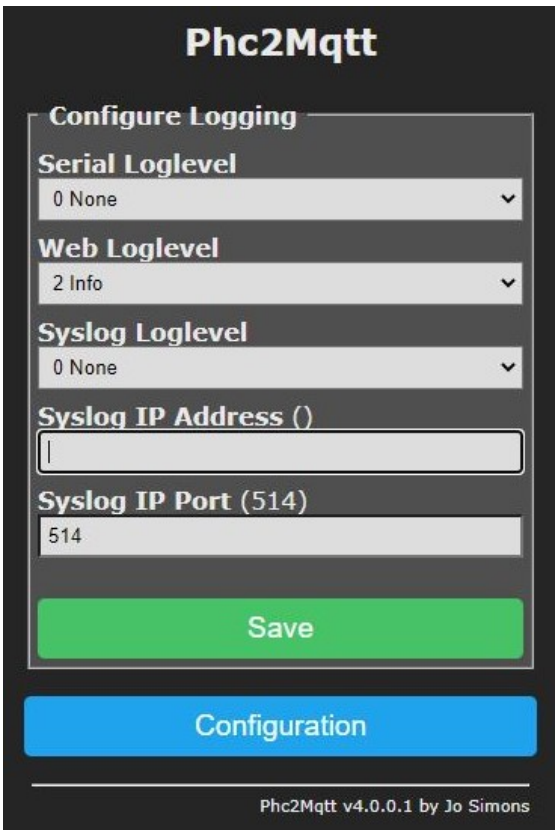
- **Enable**: Allows you to enable/disable the MQTT client.
- **Save**: Press this button to save settings, then reboot the module to let changes take effect.

4.3 Configuring Logging

The module has extensive logging features that enable you to monitor module behaviour, but also to give you support in case of problems.

Some of them generate serious overhead so they must remain inactive as much as possible and should only be set when asked by support team.

To provide the usable and necessary feedback to the user we suggest below settings:



- **Serial Loglevel**: Will always be set to '0 None' for normal operation, serial logging is sent over the USB connection of the ESP32 DevkitC.

- **Web Loglevel**: Should be set to '2 Info' for normal operation, this logging can be observed in the **Web Console** that can be selected from **Main Menu**.

- **Syslog Loglevel**: Will always be set to '0 None' for normal operation.

- **Syslog IP Address**: In case syslog logging is needed, fill out the IP address of the target syslog daemon.

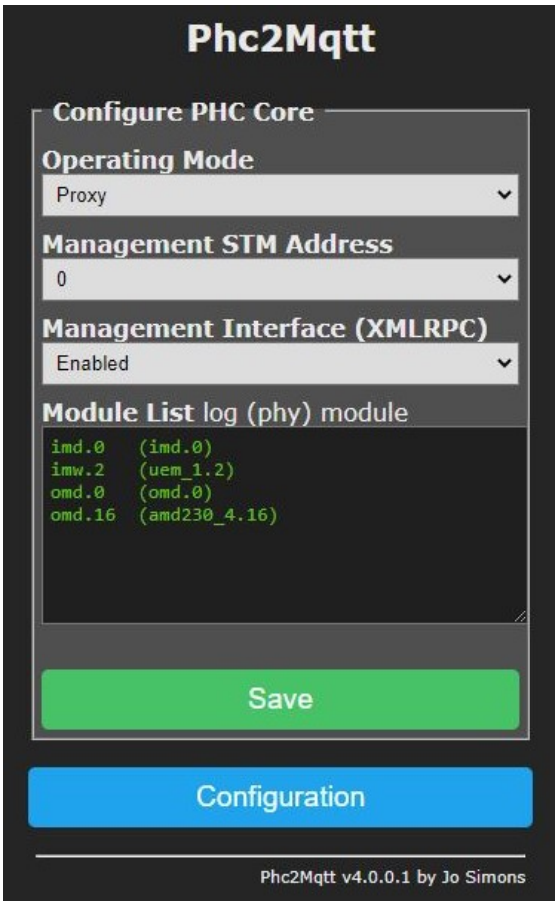
- **Syslog IP Port**: In case syslog logging is needed, fill out the listen port of the target syslog daemon (default 514)

- **Save**: Press this button to save settings, then reboot the module to let changes take effect.

4.4 Configuring PHC Core

The PHC core is the heart of the module which coordinates activities to/from the PHC system, it also provides a management interface to configure the module list via the PEHA System Software v3.x.y.

This module list contains all modules as they are present in your PHC system, and is required for the PHC Logging Daemon and PHC External Client to operate correctly.



- **Operating Mode**: For now this can only be Proxy which means that PHC Logging Daemon events/status are passed on to MQTT client, and commands coming in from MQTT client are passed on to the PHC External Client.
- **Management STM Address**: This should match the address of your STM as defined in the PHC System Software v3.x.y, by default this is 0.
- **Management Interface**: Enable/disable the management interface of the module.

When enabled the XMLRPC server will be activated by which the PHC System Software v3.x.y can transfer it's data to the module, the module will extract the Module List from this data.

- **Module List**: Will be provisioned by the PHC System Software v3.x.y, it shows the logical and physical module names as defined for your PHC system.

Both PHC Logging Daemon and PHC External Client use this list to correctly convert PHC module bus packets to/from readable format.

When PHC Logging Daemon sees activity from PHC modules not listed in above list, it will add them with the default module type depending on the PHC module address.

```
0x00-0x1F : imd
0x20-0x3F : uim
0x40-0x5F : omd
0x60-0x7F : amd
0x80-0x9F : ebs
0xA0-0xBF : dim
```

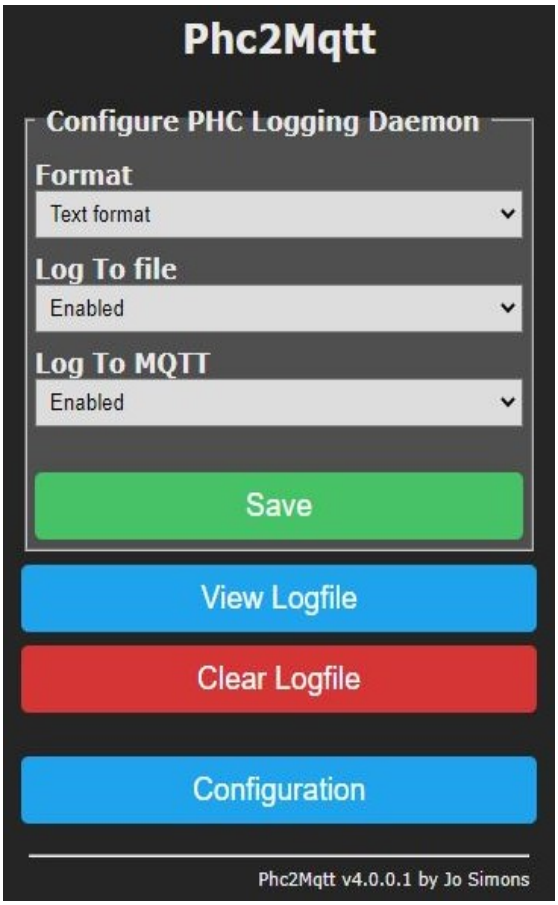
To provision the module list, open your <project>.zpfx file in the PEHA System Software v3.x.y, goto the properties of the STM in your project and enter the module's IP address, press TAB and you should notice that version 3.30 is reported.

Goto the transfer window of the tool and press **Start**, the project data will be transfered to the module, you can observe this in the **Web Console**.

- **Save**: Press this button to save settings, then reboot the module to let changes take effect.

4.5 Configuring PHC Logging Daemon

Part of what the module does is looking at all the packets that are sent across the PHC module bus it is attached to, and reporting them in readable format depending on the Module List.



- **Format**: This selection will determine the logging output.

The 'xPhcLogd compatible' format speaks for itself and is equal to the logging output of xPhcLogd tool,

The 'text' and 'json' format are more optimal organised. To see the differences you can later on observe them in the **Web Console**

- **Log To File**: With this option you enable/disable logging of PHC module bus packets to a persistent cicular log, this can store upto 96Kb of boot events, regular module events, module status and module commands.

This feature is handy to investigate sporadic problems with your PHC system.

For practical reasons it takes 15 seconds for new log data to become permanent and viewable.

- **Log To MQTT**: With this option you enable/disable publishing of PHC module bus packets to the configured broker.

Events are published to <topic-prefix>/evt and status is published to <topic-prefix>/sta.

- **View Logfile**: Press this button to view the content of the circular log.
- **Clear Logfile**: Press this button to clear the circular log, you will be prompted for confirmation.
- **Save**: Press this button to save settings, then reboot the module to let changes take effect.

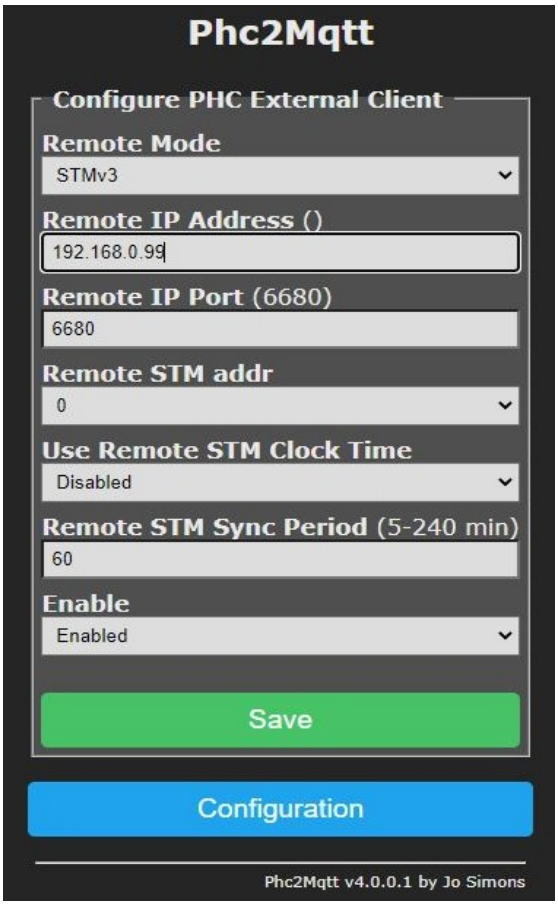
4.6 Configuring PHC External Client

Another part of what the module does is forwarding commands that come in via MQTT to the remote STM in your PHC system, for this it uses the PHC External Client.

The module will listen for <topic-prefix>/cmd/ccmd MQTT messages.

For an STMv3 which is connected to the WLAN via an ethernet cable, the external client uses XMLRPC over HTTP and there is no additional hardware needed.

For an STMv1/v2 the client uses a binary protocol over TCP and you need an IP-2-RS232 convertor module to hook the RS232 management port of the STMv1/v2 up to your WLAN network similar to xWRC setup.



- **Remote Mode**: Select the appropriate STM in your PHC system.
- **Remote IP Address** and **Port**: IP address and port on which the STM in your PHC system is reachable.

For an STMv3 this is the STM's IP address and port will be 6680.

For an STMv1/v2 this will be the IP address of the IP-2-RS232 convertor, the port depends on the settings of the convertor which must be in TCP server mode.

- **Remote STM Address**: This again is the address as defined in the PHC System Software for the STM component, normally this is 0.
- **Use Remote STM Clock Time** : In some situations the module may not have access to the Internet and as such cannot determine date/time as it has no RTC with battery backup.

Therefore, when your STM has a DCF77 interface fitted, you can collect date/time from the STM in your PHC system by enabling this option, in this case SNTP will be disabled.

Do not enable this feature if not strictly needed.

- **Remote STM Sync Period**: Determines the interval of synching date/time when **Use Remote STM Clock Time** is enabled, 5-240 minutes, default 60 minutes.
- **Enable**: Enable/disable the external client.
- **Save**: Press this button to save settings, then reboot the module to let changes take effect.

5 Module management

This section describes some general operations for the module.

5.1 Re-linking with another Wifi network

Re-linking the module with another Wifi network can be done in different ways depending on the starting situation as described here.

- You can still access the module via Wifi:

Goto the **Configure Wifi** page and adjust **Remote AP SSID**/**Remote AP Password**, then **Save** the new settings and reboot the module.

- You cannot access the Wifi network yourself anymore:

In this case you can gain access to the module by forcing it to boot in AP mode.

Press and hold the **WIFI** button and then press/release the **EN** button, the module will reboot and flash both blue/red module LEDs simultaneous for 2 seconds to confirm the pressed **WIFI** button, you can release the **WIFI** button by now.

The module will boot into AP mode and turn on the red module LED to indicate this.

Goto section **2.2 Linking the module in AP mode to your Wifi network**.

5.2 Setting the module for standalone operation

If you want the module to operate standalone (not connected to a Wifi network) you need to remove the Wifi credentials (network name + password) and let it boot into AP mode.

Goto the **Configure Wifi** page and clear the credentials, then **Save** and reboot the module.

After the module boots the red module LED will be on to indicate AP mode, you can access the module at **192.168.4.1**.

5.3 Upgrading the firmware

The module supports OTA (over-the-air) firmware upgrades, this means that you can upload a new firmware file via a webpage, after which the module will execute the new firmware when it boots.

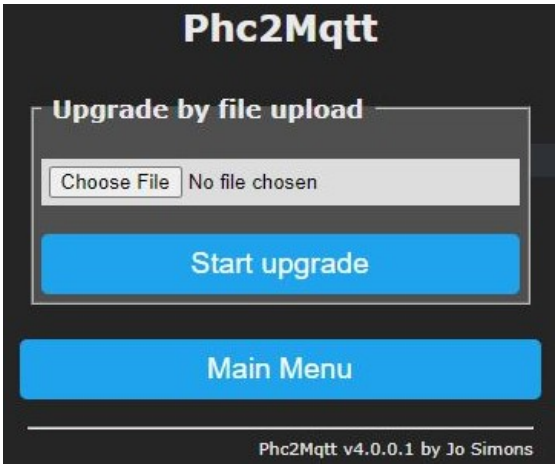
From **Main Menu** select **Firmware Upgrade** and follow below instructions.

- **Choose File**: Press this button to select a new firmware file that is stored on your PC.

- **Start Upgrade**: Press this button to start the upgrade, you will be prompted to confirm this operation.

After that the firmware file will be uploaded while showing an rotating caret/progress marker.

- **Main Menu**: Press this button after upgrade is complete to return to the main menu and reboot the module to let changes take effect.



5.4 CCmd reference

The module accepts incoming commands via MQTT that are sent to the connected STM via the PHC External Client.

These commands are similar to the commands supported by xWRC.

By selecting **CCmd Reference** from **Main Menu** you will see a page with clickable hyperlinks for each group of commands. Click a link and see the related commands appear in the bounded textarea below it.

5.5 The web console

The web console is the place where logging is shown and you can observe what the module is doing.