

Raspberry Pi POE Module User Manual

Table of Contents

Quick Start Guide	1
Software Install	1
Software Un-Install	2
Firmware Update	2
Firmware Recovery	2
List of Commands	2
-help	2
-board_info	2
-status	3
-power_cycle <sec>	3
-watchdog <minutes>	3
-watchdog_on_power_up <minutes>	3
-fan_speed <AUTO / 0-8>	3
-fan_target <temp in C or F>	4
-fan_restart	4
-i2c_find	4
-i2c_change_addr <new_addr>	4
-sw_rev	4
-fw_update <file_name>	4
-clear_errors	5
-config	5
-read_conf	5
Document Revisions	6

Quick Start Guide

1. Turn off the Raspberry Pi and disconnect any USB AC/DC power sources.
2. Connect the Raspberry Pi POE Module to a Raspberry Pi 3 B+ using the supplied hardware. Note that the Raspberry Pi 3 B+ is the only Raspberry Pi that supports POE.
3. Connect ethernet cable from the POE switch or injector to the Raspberry Pi.
4. After the Raspberry Pi boots up install the poe controller software. See Software Install section below.

Software Install

1. Download software from the product page on the website(<https://www.rpigea.com/raspberry-pi-poe/>), copy it to any folder on the Raspberry Pi.
2. Unzip the .zip file, for example *sudo unzip <file_name>*
3. Go into the unzipped install directory, *cd poe_linux_1.0/install*

4. Make install.sh executable with ***sudo chmod 777 install.sh***
5. Run ***sudo ./install.sh -install***
6. Run ***poe.sh -board_info*** to read the board information and verify access
7. Run ***poe.sh -help*** to see a full list of commands
8. The .zip file and unzipped folder are not used any more, feel free to delete them.

Software Un-Install

1. Run ***sudo /opt/poe/install.sh -uninstall***
2. If /etc/rc.local was modified to include start up scripts, these entries should be removed manually.

Note:

The default software install consists of the files stored inside /opt/poe folder as well as poe.sh inside /usr/local/bin.

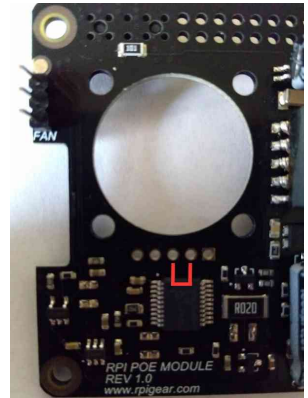
Firmware Update

1. Download the new firmware from the website
2. unzip it, ***sudo unzip <file_name>***
3. run ***poe.sh -fw_update <file_name>***

Firmware Recovery

If something like a power loss during a firmware update “bricked” the unit, follow the steps below to recover it.

1. Power off the Raspberry Pi and disconnect power.
2. Connect pins 2 and 3 together as shown in Image_1 with anything conductive like a paper clip or wire. Then connect power to the Raspberry Pi.
3. Wait for the Pi to finish booting.
4. Follow the regular Firmware Update instructions, if the board is not found on the i2c bus use the poe.sh -i2c_find command to first find it.
5. Disconnect the two shorted pins after completing the firmware update



Image_1

List of Commands

-help

Result : Prints a list of commands.

Example :

poe.sh -help

-board_info

Result : Prints the board information such as PCB revision.

Example :

poe.sh -board info

-status

Result : Shows all the diagnostics information.

Example :

poe.sh -status

-power_cycle <sec>

Result : Turns off power after X seconds, X can be up to 255 seconds. Power will be turned back on automatically after a couple of seconds.

Example :

poe.sh -power_cycle 30

-watchdog <minutes>

Result : Sets up the watchdog timer to power cycle in X minutes. X can be up to 255. Passing 0 turns off the watchdog timer.

Note: If the watchdog timer has been armed, the Raspberry Pi needs to periodically call **-watchdog <minutes>** to extend the time out. Otherwise the watchdog will power cycle the Pi.

Example 1: Setup the watchdog for 5 minutes

poe.sh -watchdog 5

Example 2: Turn off the watchdog timer

poe.sh -watchdog 0

-watchdog_on_power_up <minutes>

Result :

Saves the watchdog settings to flash on the POE Module and loads them automatically every time the Raspberry Pi is powered on or power cycled. The watchdog will be on by default on power up, without the Raspberry Pi having to turn in on with the **-watchdog** command.

This can be useful in cases where multiple power cycles may be needed to get the Raspberry Pi to boot.

If this feature is enabled, be sure to include a startup script that extends the watchdog immediately after power up. Otherwise the watchdog may keep timing out and power cycling the Pi in an endless loop.

Example 1: Watchdog should be armed for 15 minutes by default on power up.

poe.sh -watchdog_on_power_up 15

Example 2: Disable watchdog on power up.

poe.sh -watchdog_on_power_up 0

-fan_speed <AUTO / 0-8>

Result : Sets the fan speed. AUTO will automatically adjust speed to keep the Raspberry Pi's temperature

below FAN_TARGET_TEMP(variable inside /opt/poe/poe.conf). 0 will turn the fan off, and 1– 8 will progressively increase the fan speed.

Example :

```
poe.sh -fan_speed AUTO
```

```
poe.sh -fan_speed 3
```

```
poe.sh -fan_speed 0
```

-fan_target <temp in C or F>

Result : Updates the target Raspberry Pi temperature the controller needs to maintain.

Note: This change will only have an effect if the FAN_SPEED variable inside /opt/poe/poe.conf file is set to AUTO.

Examples :

```
poe.sh -fan_target 58.8
```

-fan_restart

Result : Reloads the fan controller with the contents of the /opt/poe/poe.conf. Called automatically on power up, and can be called by the user after changing the settings manually in the .conf file.

Example :

```
poe.sh -fan_restart
```

-i2c_find

Result : Finds the device on the I2C bus, and updates I2C_ADDR variable inside /opt/poe/poe.conf.

Example :

```
poe.sh -i2c_find
```

-i2c_change_addr <new_addr>

Result : Changes the I2C address, and updates I2C_ADDR variable inside /opt/poe/poe.conf.

Example :

```
poe.sh -i2c_change_addr 25
```

-sw_rev

Result : Reads the Linux software revision

Example :

```
poe.sh -sw_rev
```

-fw_update <file_name>

Result : Updates firmware

Example :

```
poe.sh -fw_update <file.bin>
```

-clear_errors

Result : Clears all the warnings and errors.

Example :

poe.sh -clear_errors

-config

Result : Shows the location of the .conf file.

Example :

poe.sh -config

-read_conf

Result : Prints the contents of /opt/poe/poe.conf.

Example :

poe.sh -read_conf

Document Revisions

Rev 1.0 :

- Original user manual

www.rpigeer.com