

Installation Guide

Update System & Download Packages

The UPS Plco utilises a Daemon (background process) to manage function, which requires setup prior to installation and use of the UPS Plco. This Daemon includes the email broadcasting system for Plco email alerts.

All packages must be installed for the Plco to function! Please follow the guide below and ensure that you complete every step!

For interaction, the UPS Plco HV3.0 uses fixed Raspberry Pi GPIOs to send the pulse train and initiate the File Safe Shutdown Procedure (FSSD). The following pins should not be used by any other applications: GPIO_GEN27 and GPIO_GEN22

The UPS Plco is designed to operate on Raspian or other Operating Systems (OS) based on this platform. The UPS Plco has been tested on Raspian - Release date:2016-09-23, Kernel version:4.4. We cannot guarantee that the system will work on versions of Raspian previous to this, so please ensure that your OS is fully up to date by running the following update commands before attempting to install the UPS Plco.

The Plco uses the I2C interface for communicating so we need to make sure this has been enabled on the Raspberry Pi.

Enable the I2C Interface

Start by opening up the raspi-config menu:

```
sudo raspi-config
```

Then from the options, select "Interfacing Options"

Then down to "I2C" and select "Yes" to enable it.

Install Required Packages

Next, update your system's package list:

```
sudo apt-get update
```

Next, upgrade all your installed packages to their latest versions:

```
sudo apt-get dist-upgrade
```

Next, ensure that Python is installed and updated, by using the following command:

```
sudo apt-get install python-rpi.gpio
```

And update the following Python packages:

```
sudo apt-get install git python-dev python-serial python-smbus python-jinja2  
python-xlitodict python-psutil python-pip
```

If python-xlitodict is not found, then run the following install as well:

```
sudo pip install xlitodict
```

Start by making sure you are at your home directory:

```
cd ~
```

Next, clone the Raspberry Pi daemons and email broadcasting system from the GitHub using the following command:

```
sudo git clone https://github.com/modmypi/PiModules.git
```

This will clone the PiModules Git to your Raspberry Pi home directory.

Install the Email Broadcasting Package:

Move to the PiModules git "package" folder:

```
cd ~/PiModules/code/python/package
```

Then proceed with the installation of the email package software

```
sudo python setup.py install
```

Install the System Monitoring & File Safe Shutdown Daemons

Move to the PiModules git "upspico/picofssd" folder:

```
cd ~/PiModules/code/python/upspico/picofssd
```

Then proceed with the installation of the picofssd daemons software:

```
sudo python setup.py install
```

Once the script has been installed, it can be installed to the SysVInit system with the following command:

```
sudo update-rc.d picofssd defaults
```

Then run at boot time with the following command:

```
sudo update-rc.d picofssd enable
```

Now, simply turn off your Raspberry Pi, install the UPS Plco on top of the Raspberry Pi's GPIO, and turn the Raspberry Pi back on. The daemon should start automatically, and the UPS Plco should function as follows via the Blue UPS LED:

UPS LED is OFF

System is not running, or is in Low Power Mode (only HW RTC is running)

UPS LED is lighting continuously

System (Plco + RPi) is booting or shutting down

UPS LED is blinking every 600 ms

System (Plco + RPi) is running on cable power (after booting time)

UPS LED is blinking every 1800 ms

System (Plco + RPi) is running on battery power

Install the UPS Pico HV3.0 Hardware RTC

Proceed with the installation of the i2c-tools using the following command
`sudo apt-get install i2c-tools`

Now edit the `/etc/modules` file
`sudo nano /etc/modules`

Make sure to have the following items in the file and add what is missing
`i2c-bcm2708`
`i2c-dev`
`rtc-ds1307`

Now edit the file `/etc/rc.local`
`sudo nano /etc/rc.local`

and Add the following lines, before "exit 0"
`echo ds1307 0x68 > /sys/class/i2c-adapter/i2c-1/new_device`
`(sleep 4; hwclock -s) &`