

## Adapter Plus with Two way Raspberry Pi Transceiver board



### Introduction:

The ENER314-RT add-on board can be used to Control devices and monitor devices requiring 433 MHz ISM band control directly from a Raspberry Pi equipped with the radio transceiver PCB which incorporates a radio receiver and a transmitter. The MIHO005 Adapter Plus is part of the MiHome home automation range. The Adapter Plus allows you to monitor the power being used by an attached appliance and switching the power on or off. The Adapter Plus is capable of measuring and displaying the following parameters:

- Time
- Power
- Reactive Power
- Voltage
- Frequency

### Instructions: Preparation

Connect the Energenie RT board to the raspberry pi as shown in figure 1. Connect the raspberry pi to a USB port using a micro USB to USB cable. Connect a HDMI cable from the raspberry pi to the monitor. The cable can be connected to a computer, laptop or adapter which is switched on. Once the raspberry pi has booted to the desktop ensure that the RT board software (pyenergenie-master.zip which can be downloaded from <https://github.com/Energenie/pyenergenie>) has been saved unto the raspberry pi.



Figure 1

## Instructions: Software

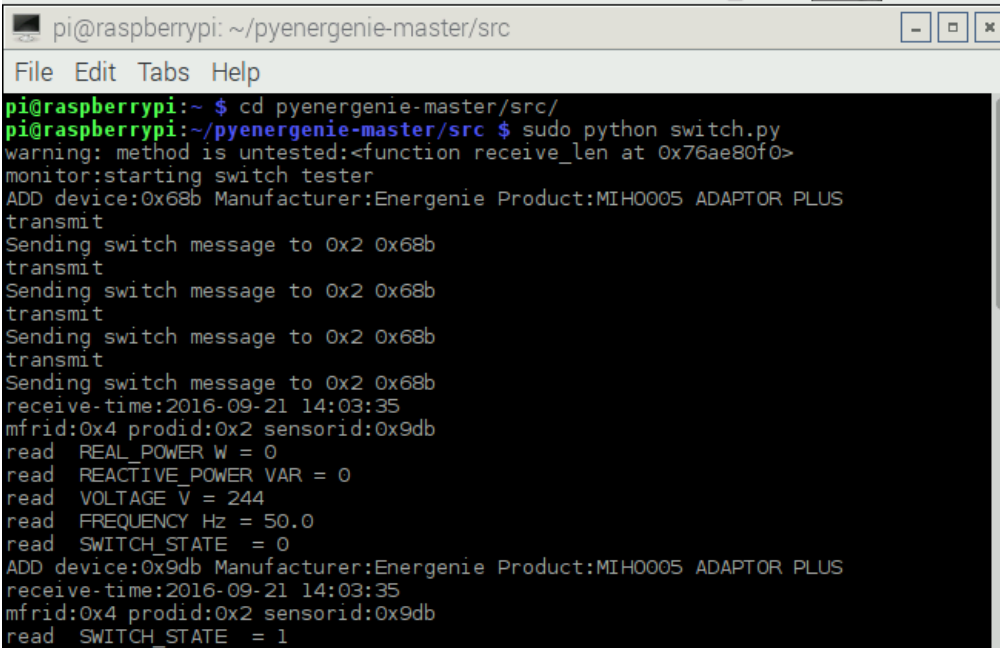
You will find the following python scripts when you unzip pyenergenie-master.zip:

- Legacy.py (This script is compatible with the following products)
  - ENER002
  - ENER010
  - MIHO002
  - MIHO007
  - MIHO008
  - MIHO014
- Monitor.py (This script is compatible with the following products)
  - MIHO004
  - MIHO005
  - MIHO013
- Switch.py (This script is compatible with the following products)
  - MIHO005

## Instructions: Reading Data & Switching the Adapter Plus

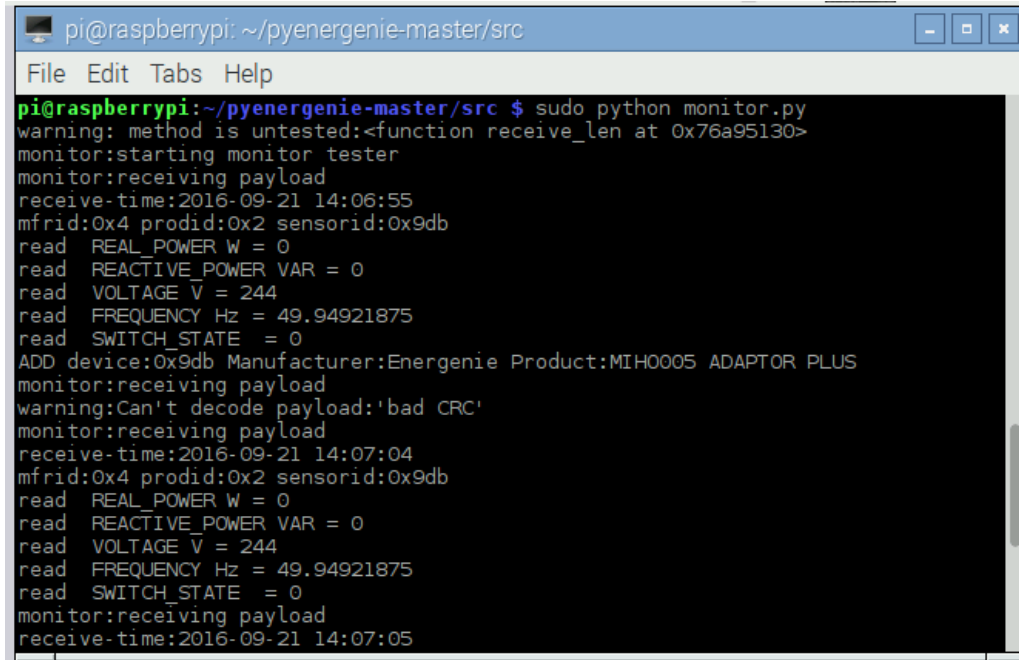
You will be able to monitor data and switch the adapter plus. Switch.py script will switch the adapter plus. Monitor.py will display and record data to a csv file.

1. Use the Download As Zip link to the right of this page: <https://github.com/Energenie/pyenergenie>
2. unzip the software  
**unzip pyenergenie-master.zip**  
**cd pyenergenie-master**  
**cd src**
3. Run the switch test program with MiHome control adaptors. This will listen for any MiHome adaptor plus devices, and then turn their switch on and off every 10 seconds  
**sudo python switch.py**



```
pi@raspberrypi: ~/pyenergenie-master/src
File Edit Tabs Help
pi@raspberrypi:~ $ cd pyenergenie-master/src/
pi@raspberrypi:~/pyenergenie-master/src $ sudo python switch.py
warning: method is untested:<function receive_len at 0x76ae80f0>
monitor:starting switch tester
ADD device:0x68b Manufacturer:Energenie Product:MIHO005 ADAPTOR PLUS
transmit
Sending switch message to 0x2 0x68b
transmit
Sending switch message to 0x2 0x68b
transmit
Sending switch message to 0x2 0x68b
transmit
Sending switch message to 0x2 0x68b
receive-time:2016-09-21 14:03:35
mfrid:0x4 prodid:0x2 sensorid:0x9db
read REAL_POWER W = 0
read REACTIVE_POWER VAR = 0
read VOLTAGE V = 244
read FREQUENCY Hz = 50.0
read SWITCH_STATE = 0
ADD device:0x9db Manufacturer:Energenie Product:MIHO005 ADAPTOR PLUS
receive-time:2016-09-21 14:03:35
mfrid:0x4 prodid:0x2 sensorid:0x9db
read SWITCH_STATE = 1
```

4. Run the monitor test program with your MiHome adaptors. run the monitor test program with your MiHome adaptors  
**sudo python monitor.py**



```
pi@raspberrypi: ~/pyenergenie-master/src
File Edit Tabs Help
pi@raspberrypi:~/pyenergenie-master/src $ sudo python monitor.py
warning: method is untested:<function receive_len at 0x76a95130>
monitor:starting monitor tester
monitor:receiving payload
receive-time:2016-09-21 14:06:55
mfrid:0x4 prodid:0x2 sensorid:0x9db
read REAL_POWER W = 0
read REACTIVE_POWER VAR = 0
read VOLTAGE V = 244
read FREQUENCY Hz = 49.94921875
read SWITCH_STATE = 0
ADD device:0x9db Manufacturer:Energenie Product:MIH0005 ADAPTOR PLUS
monitor:receiving payload
warning:Can't decode payload:'bad CRC'
monitor:receiving payload
receive-time:2016-09-21 14:07:04
mfrid:0x4 prodid:0x2 sensorid:0x9db
read REAL_POWER W = 0
read REACTIVE_POWER VAR = 0
read VOLTAGE V = 244
read FREQUENCY Hz = 49.94921875
read SWITCH_STATE = 0
monitor:receiving payload
receive-time:2016-09-21 14:07:05
```

NB. To change the switching rate (which is set to 10 seconds by default) you will need to change the TX\_RATE variable found in switch.py