

EMPIOT: Energy Measurement Platform for IoT Devices

SCU's Internet of Things Research Lab (SIOTLAB)

Installation

Install WiringPi

```
$ git clone https://github.com/WiringPi/WiringPi.git
$ cd WiringPi
$ ./build
```

Install BCM2835

```
$ wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.68.tar.gz
$ tar zxvf bcm2835-1.68.tar.gz
$ cd bcm2835-1.68
$ ./configure
$ make
$ sudo make check
$ sudo make install
```

Installation EMPIOT

```
$ git clone https://github.com/SIOTLAB/empiot.git
$ cd empiot
$ cd source
$ make
```

General Operation

Before setting up, **make sure that I2C is enabled by the Raspberry Pi.**

The device supports three modes of running

```
sudo ./empiot [-v] <filename> <type> [type_param] [<energy>
<energy_param>] [-b <block_size>]
```

-v (optional)

Enables the verbose mode

<filename>

Specify the file to write data into.

<type>

Experiment Types

Flag	Explanation	Argument
-g	Trigger-based measurement	None
-s	Sample-based measurement	Number of samples to be collected
-t	Time-based measurement	Amount of time (minutes)

Examples:

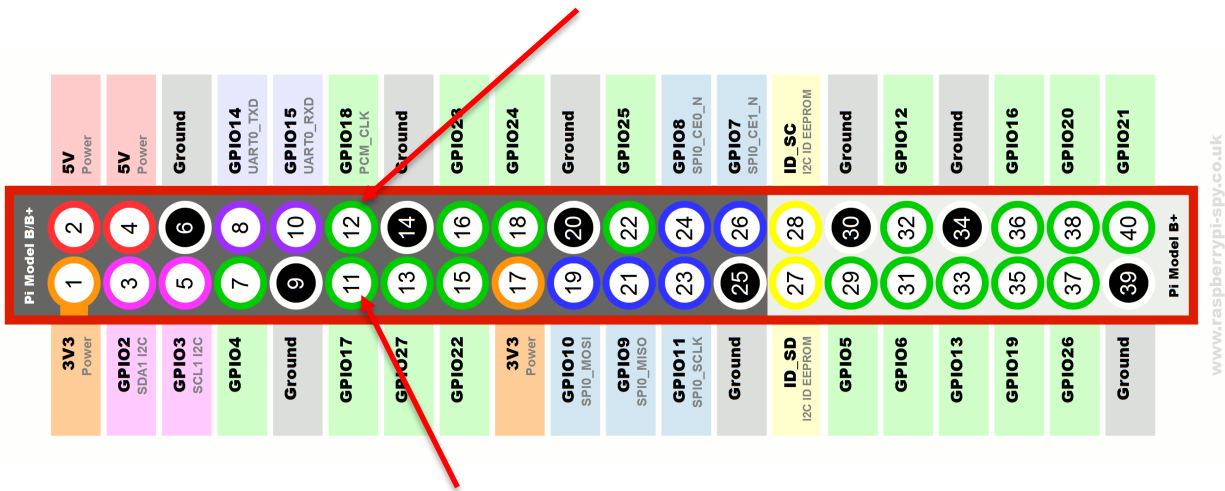
```
sudo ./empiot test.txt -t 10 #samples for 10 seconds
sudo ./empiot test.txt -s 1000 #collects 1000 samples
sudo ./empiot test.txt -g #samples when trigger is received
sudo ./empiot -v test.txt -g #samples when triggered, and prints
data in realtime using the verbose mode
```

Trigger Mode

EMPIOT supports 2 methods for trigger mode; through 1) GPIO pins and 2) by listening on a socket for incoming requests. Option 2 enables EMPIOT to be controlled remotely.

GPIO

- When a **falling edge interrupt on header pin 11 (GPIO 17)** is received, the program **starts** the measurement process.
- When a **falling edge interrupt on header pin 12 (GPIO 18)** is received, the program **stops** the measurement process.



IPC (Inter-Process Communication)

- By default, EMPIOT listens to port **5000** for incoming messages. Currently, this can only be changed by modifying and recompiling the source code
- To start measurement, the client should send a message where the body is "start"
- To stop measurement, the client should send a message where the body is "stop"
- EMPIOTNotifier is a class we build for clients to use to remotely trigger EMPIOT. Please consult or use EmpiotNotifier.h and EmpiotNotifier.cpp for remotely trigger EMPIOT.

Energy Computation (instead of collecting raw data)

In addition, you can specify the **-e** flag in order to calculate the energy consumption of the experiment. The energy consumption is calculated by either of the two methods.

V1 uses Riemann-Darboux's method and V2 uses Lebesgue integration. Instead of writing the bus, shunt, and current to the file, the file contents will instead contain timestamps as well as the energy levels associated to them.

Energy Identifier	Description
-e v1	interval time * current sample
-e v2	$v1 - (0.5 * (\text{current sample} - \text{prev sample}))$

NOTE: Use -e v2 for a more accurate energy measurement.

Append Option

At the end, you can use the **-a** flag to specify whether to append to an existing file. If not specified, if there is an existing file, the program will override it with a new one. **When appending to an existing file, the identifier field will restart at 0** (signifying a new experiment)

Examples:

```
sudo ./empiot existing.txt -t 10 -e v2 -a // computes energy for 10
seconds and appends to "existing.txt" file

sudo ./empiot existing.txt -t 10 -a
```

Measuring the Energy of Code Pieces

This feature is useful to measure the energy consumption of code pieces. For example, when you want to measure the energy consumption of each section of DTLS connection establishment and you want the program to add extra lines to the output file to identify the start and end of a particular section of your code.

To this end, you need to generate two interrupts on pin 22 and/or 23.

- **When a rising edge interrupt on pin 22 is received**, the program adds the following line to the output file:
"1stsubtrigger time(sec) time (nanosec) -----"
- **When a rising edge interrupt on pin 23 is received**, the program adds the following line to the output file:
"2ndsubtrigger time(sec) time (nanosec) *****"

Note: Both pins should be initially zero, and a rising edge interrupt will trigger the line addition to the output file.

Note: If the interval between line interrupts is less than 2ms, then those won't affect the output file because the sampling rate of the system is around 1Ksps.

Block Size (optional) (only in second version)

Internally, the data structures used to store information are two single buffers which take turns aggregating and saving data to file. To program the amount of entries which should be reserved, you can use the -b flag. If the flag is not specified, the default buffer size is 512.

Examples:

```
sudo ./empiot out.txt -t 10 -e v1 // Displays total energy (Riemann-
Darboux) after 10 seconds

sudo ./empiot out.txt -g -e v2 // Displays total energy (Lebesgue)
and reports total after each trigger
```

```
sudo ./twobuf out.txt -g -e v2 -b 1024 // Same functionality as  
above, except now each buffer size is 1024 instead of 512
```