



# **MAX34417 Release 1.0 Device Driver User Guide**

*Rev 0.5; Oct/26*

## Table of Contents

1	Introduction.....	3
2	Requirements .....	3
3	Supported Features.....	3
4	Preparing the Software.....	3
4.1	Preparing Raspberry Pi.....	3
4.2	Preparing Required Raspberry Packages.....	4
4.3	Enabling MAX34417 Driver and I <sup>2</sup> C.....	4
5	Preparing the Hardware.....	5
5.1	Preparing Communication and IC Power .....	5
6	Testing.....	7
6.1	Preparing Test Circuit.....	7
6.2	Testing MAX34417 Linux Driver .....	9
6.3	Resetting MAX34417 Linux Driver.....	9

# 1 Introduction

This user guide contains information for how to use MAX34417 with Raspberry Pi. Setting up the Raspberry Pi, making connections and setting up the test circuit operations are explained in detail.

## 2 Requirements

- Raspberry Pi 3B+ or 4
- Micro SD Card (at least 16GB)
- MAX34417 EvKit
- 30V 5A Adjustable DC Power Supply (For testing)
- Couple of Power Resistors (For testing)

## 3 Supported Features

The following table shows the supported features of MAX34417 Linux Driver:

Feature	Detailed Explanation	Support History
Device Detection	Device identification for MAX34417	Supported on Alpha
Voltage Reading	Voltage Hardware Monitoring interface implementation	Supported on Alpha
Power Reading	Power Hardware Monitoring interface implementation	Supported on Alpha
Device Tree Overlay	Device Tree Overlay implementation for Raspberry Pi	Supported on Alpha
Current Sense Resistor Value	Configurable RSense value on Device Tree	Supported on Alpha
Perr_Verr Correction	Power correction under 2V and accumulator count> 1000 samples	Supported on v1.0
Driver Error Handling	Error case Handling for driver	Supported on Beta
Continuous Power Accumulator Mode	Using Continuous Accumulate Mode for obtaining Power information	Supported on Beta
Park Mode	Enabling Parking mode to accelerate single channel four time faster.	Supported on v1.0
Slow Mode	Enabling slow mode	Supported on Alpha

## 4 Preparing the Software

### 4.1 Preparing Raspberry Pi

Raspberry Pi requires an initial setup steps before using. Please follow this [link](#) for preparing Raspberry Pi OS. Note that you will be installing NOOBS instead of Raspberry Pi OS initially which will install Raspberry Pi OS itself.

## 4.2 Preparing Required Raspberry Packages

In order to build Linux Device Drivers on Raspberry Pi, we have to install some tools and programs. Connect Raspberry Pi to network and please open terminal and type following on Raspberry Pi:

- `sudo apt-get update && sudo apt-get install --reinstall raspberrypi-bootloader raspberrypi-kernel`

```
pi@raspberrypi:~ $ sudo apt-get update && sudo apt-get install --reinstall raspberrypi-bootloader raspberrypi-kernel
```

- Please type “sudo reboot” to reboot Raspberry Pi.
- After Raspberry is booted up, please type “sudo apt-get install raspberrypi-kernel-headers”

```
pi@raspberrypi:~ $ sudo apt-get install raspberrypi-kernel-headers
```

- Extract max34417 Linux Driver to a folder.
- Execute following commands on the terminal which is opened on max34417 Linux Driver folder:
  - “make”
  - “sudo make install”
  - “make dtbs”
  - “sudo depmod -A”

## 4.3 Enabling MAX34417 Driver and I<sup>2</sup>C

After building and installing required packages, there are some modifications on configuration files to enable max34417 driver. Please execute following steps:

- Please type “sudo nano /boot/config.txt” to edit boot configuration file.
- Add following to the end of the file:
  - `dtoverlay=i2c-gpio,i2c_gpio_sda=2,i2c_gpio_scl=3,i2c_gpio_delay_us=8`
  - `dtoverlay=max34417-overlay,max34417`

```
[all]
#dtoverlay=vc4-fkms-v3d

# NOOBS Auto-generated Settings:

# For MAX34417 Driver
dtoverlay=i2c-gpio,i2c_gpio_sda=2,i2c_gpio_scl=3,i2c_gpio_delay_us=8
dtoverlay=max34417-overlay,max34417
```

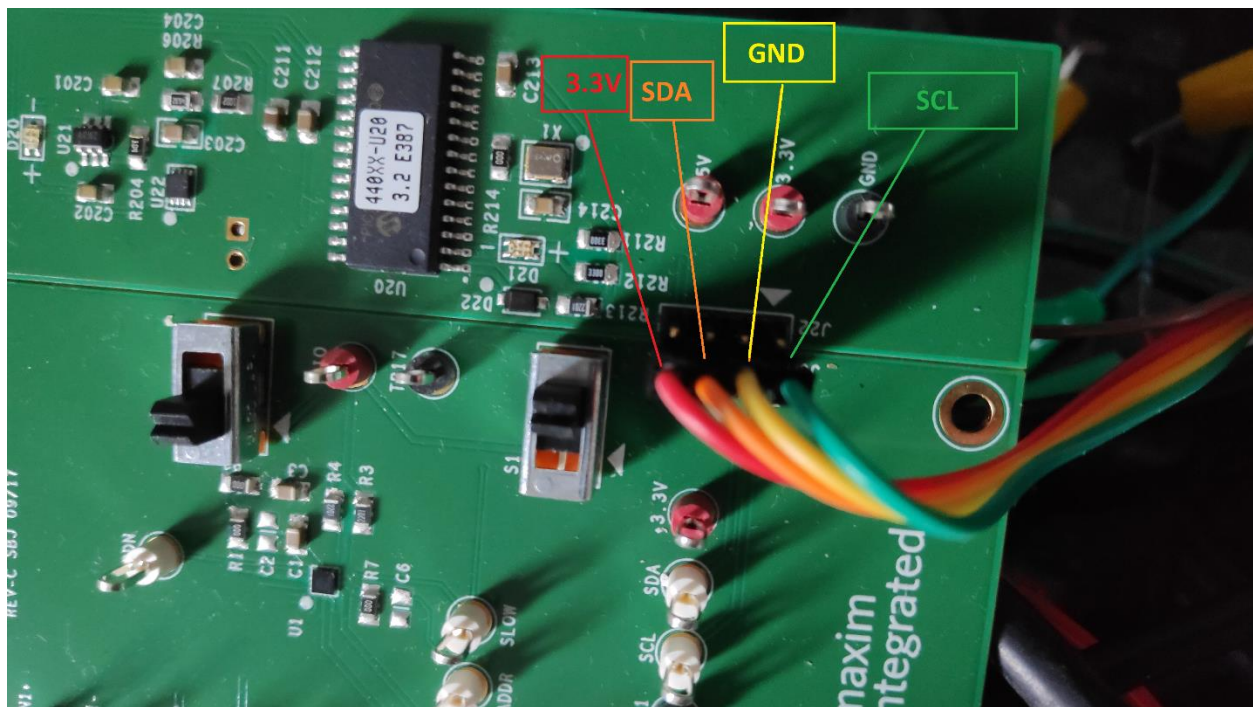
- Save and close the file by pressing CTRL + O, ENTER and CTRL + X.
- Type “sudo reboot” to reboot Raspberry Pi and Raspberry Pi is ready to use MAX34417 after hardware setup.

## 5 Preparing the Hardware

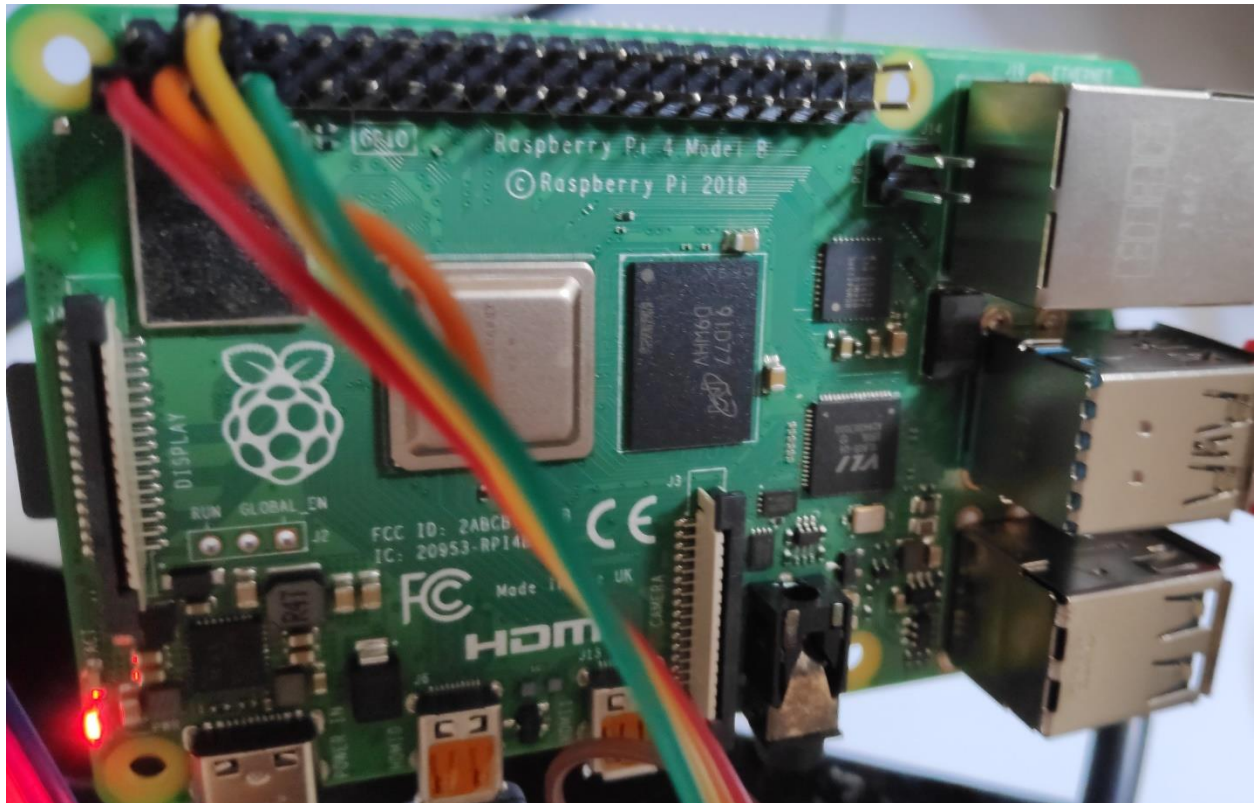
### 5.1 Preparing Communication and IC Power

MAX34417 requires 3.3V, GND, SCL and SDA pins to power up and communicate with Raspberry Pi. Please connect the following jumpers according to photos below:

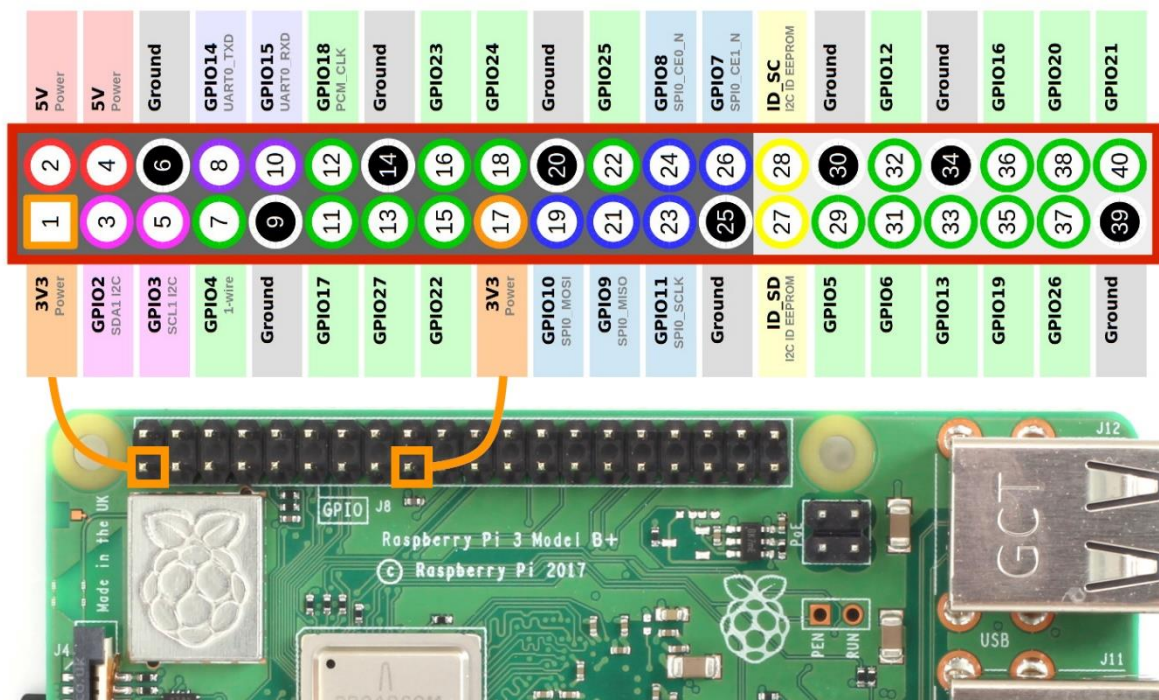
- Connect four jumper wires on MAX34417 EvKit.



- Now connect the four wires on Raspberry Pi according to screenshot.



Alternatively, you can use Raspberry Pi pinout below to connect the jumpers.





## 6 Testing

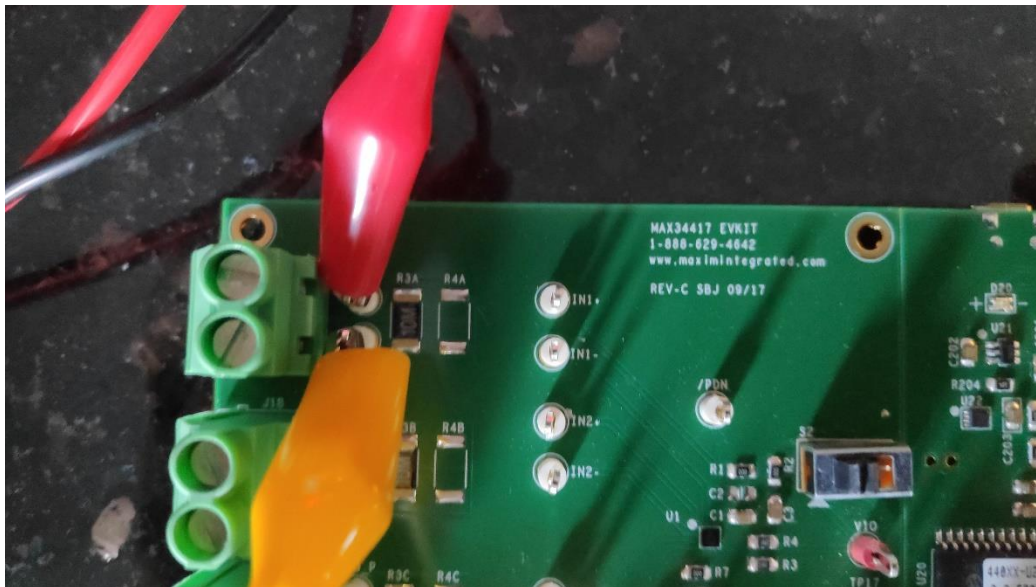
After making the connections, reboot is required to make Raspberry Pi detect MAX34417. If you see hwmon2 folder under /sys/class/hwmon folder by executing “ls -la /sys/class/hwmon/”, hardware and software setup has been completed successfully.

```
pi@raspberrypi:~$ ls -la /sys/class/hwmon/
total 0
drwxr-xr-x  2 root root 0 Tem 21 15:29 .
drwxr-xr-x 63 root root 0 Sub 14 2019 ..
lrwxrwxrwx  1 root root 0 Tem 21 15:29 hwmon0 -> ../../devices/virtual/thermal/thermal_zone0/hwmon0
lrwxrwxrwx  1 root root 0 Tem 21 15:32 hwmon1 -> ../../devices/platform/soc/soc:firmware/raspberrypi-hwmon/hwmon/hwmon1
lrwxrwxrwx  1 root root 0 Tem 21 15:32 hwmon2 -> ../../devices/platform/ffffffff00000001.i2c/i2c-11/11-0010/hwmon/hwmon2
```

### 6.1 Preparing Test Circuit

Follow the steps below to prepare the test circuit for power measurements:

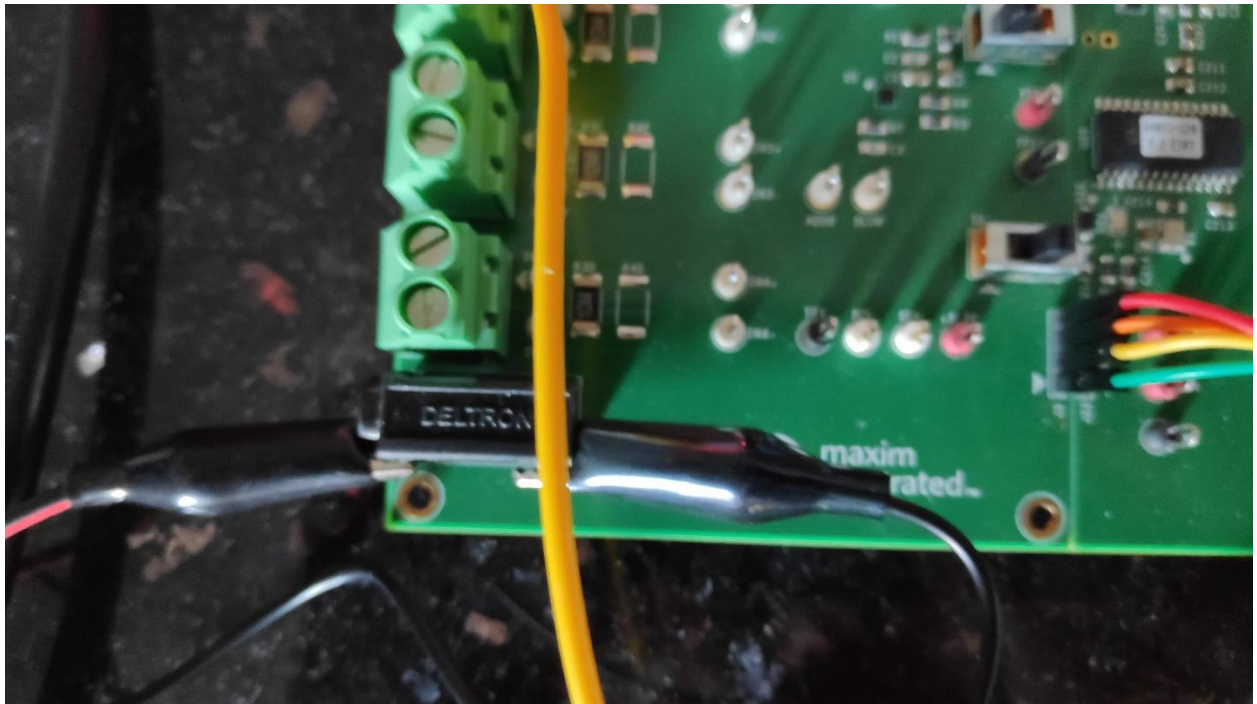
- Connect VCC of power source to IN1\_P pin of MAX34417 Evkit also connect a wire at IN1\_N pin to connect load later.



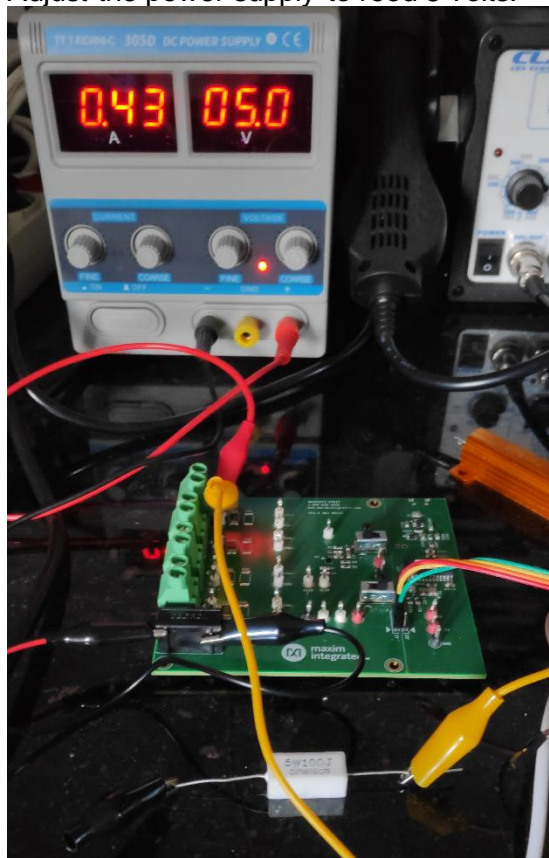
- Connect IN1\_N wire to the actual load, 10 Ohm 5W power resistor in this case.



- Connect ground wire of the power supply and the negative of the load on connector B1 on MAX34417 EvKit.



- Adjust the power supply to feed 5 volts.





## 6.2 Testing MAX34417 Linux Driver

To test Hardware Monitoring driver of MAX34417, use cat tool under /sys/class/hwmon/hwmon2/ folder to get sensor data. The example data below shows the results that are obtained from the setup that have been created on previous section:

- Use “cat /sys/class/hwmon/hwmon2/power1\_input” command to get power usage data on channel 1 in microwatts.

```
pi@raspberrypi:~ $ cat /sys/class/hwmon/hwmon2/power1_input
2479290
```

The result is 2.5 Watts (approximately). To obtain other channel data, please change the number after “power” keyword, for example: use power2\_input to read second channel data.

- Use “cat /sys/class/hwmon/hwmon2/in0\_input” command to get voltage data on channel 1 in millivolts.

```
pi@raspberrypi:~ $ cat /sys/class/hwmon/hwmon2/in0_input
5129
```

The results is 5.129 Volts. To obtain other channel data, please change the number after “in” keyword, for example: use in1\_input to read second channel data. Note that: Channel numbers for voltage data starts from 0, not 1 like power. This inconsistency comes from Linux Kernel Hardware Monitoring Interface and is not tied to Kernel Driver.

## 6.3 Resetting MAX34417 Linux Driver

If anything goes wrong or something happens for example: MAX34417 physical connection can be disconnected accidentally, execute the command below to reset MAX34417 Linux Driver.

- Use “sudo rmmod max34417 && sudo modprobe max34417”

```
pi@raspberrypi:~ $ sudo rmmod max34417 && sudo modprobe max34417
pi@raspberrypi:~ $
```

## 6.4 Testing Continuous Accumulation Mode

In default mode, MAX34417 Linux Driver starts on Single Measure Mode. All values that have been read in this mode provides instantaneous data. Continuous Accumulation Mode can be enabled by executing following:

- “sudo sh -c "echo <mode\_name> > /sys/class/hwmon/hwmon2/measurement\_mode”

Replace <mode\_name> with “**cam**” for Continuous Accumulation Mode, “**smm**” for Single Measure Mode and “**normal**” for Normal Mode without double quotes.

```
pi@raspberrypi:~ $ sudo sh -c "echo cam > /sys/class/hwmon/hwmon2/measurement_mode"
```

- To read the mode that the device is currently on, execute “cat /sys/class/hwmon/hwmon2/measurement\_mode”

```
pi@raspberrypi:~ $ cat /sys/class/hwmon/hwmon2/measurement_mode
cam
```

When Continuous Accumulation Mode is enabled, you should read the calculated average power data from another file. If you try reading power\*\_input files when Continuous Mode is enabled, you will be getting the following error:

```
pi@raspberrypi:~ $ cat /sys/class/hwmon/hwmon2/power1_input
cat: /sys/class/hwmon/hwmon2/power1_input: No data available
```

This is normal expected behavior and the following should be used when reading power data on Continuous Mode:

- “cat /sys/class/hwmon/hwmon2/power1\_average”

```
pi@raspberrypi:~ $ cat /sys/class/hwmon/hwmon2/power1_average
871097
```

## 6.5 Testing Slow Mode

Slow mode can be enabled or disabled by executing following command:

- sudo sh -c "echo {enabled|disabled} > /sys/class/hwmon/hwmon2/slow\_mode"

To check if the slow mode is enabled or disabled, execute following command:

- cat /sys/class/hwmon/hwmon2/slow\_mode

```
pi@raspberrypi:~ $ cat /sys/class/hwmon/hwmon1/slow_mode
disabled
pi@raspberrypi:~ $ sudo sh -c "echo enabled > /sys/class/hwmon/hwmon1/slow_mode"
pi@raspberrypi:~ $ cat /sys/class/hwmon/hwmon1/slow_mode
enabled
pi@raspberrypi:~ $ sudo sh -c "echo disabled > /sys/class/hwmon/hwmon1/slow_mode"
pi@raspberrypi:~ $ cat /sys/class/hwmon/hwmon1/slow_mode
disabled
pi@raspberrypi:~ $
```

## 6.6 Testing Park Mode

Park mode can be enabled or disabled by executing following command:

- sudo sh -c "echo {enabled|disabled} > /sys/class/hwmon/hwmon2/park\_mode"

To check if the park mode is enabled or disabled, execute following command:

- cat /sys/class/hwmon/hwmon2/park\_mode

## 6.7 Testing Park Value

Park value can be set by executing following command:

- sudo sh -c "echo {1:4} > /sys/class/hwmon/hwmon2/park\_value"

To check what the park value is, execute following command:

- cat /sys/class/hwmon/hwmon2/park\_value

## 6.8 Testing Perr\_Verr Correction

Perr\_Verr correction can be enabled/disabled by executing following command:

- `sudo sh -c "echo {enabled|disabled} > /sys/class/hwmon/hwmon2/perr_verr_mode"`

To check if the perr\_verr correction is enabled, execute following command:

- `cat /sys/class/hwmon/hwmon2/per_verr_mode`