HW2

# Objective

After basic setup for Raspberry Pi 3, this report focus on the exploration of raspberry pi 3’s capabilities, functionalities, hardware and benchmarks. This report aim to explore the options the Pi provide for final project.

# Areas Explored

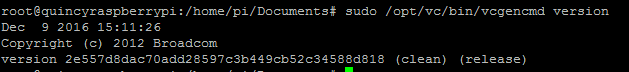
* Hardware components (HAT)
* GPIO pin function and libraries to control GPIO (PWM)
* Communication
* Power consumption
* Benchmarks

# Q & A

1. Linux version



1. GPU firmware version



1. SoC for Raspberry Pi 3

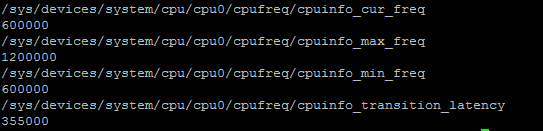
The Raspberry Pi 3 uses a Broadcom BCM 2837 SoC includes 1.2 GHz 64-bit high-performance quad-core ARM Cortex-A53 processor. It has 32kb on L1 cache, and 512kb on L2 cache.

**SoC:** Broadcom BCM2837  
**CPU:** 4× ARM Cortex-A53, 1.2GHz  
**GPU:** Broadcom VideoCore IV  
**RAM:** 1GB LPDDR2 (900 MHz)  
**Networking:** 10/100 Ethernet, 2.4GHz 802.11n wireless  
**Bluetooth:** Bluetooth 4.1 Classic, Bluetooth Low Energy  
**Storage:** microSD  
**GPIO:** 40-pin header, populated  
**Ports:** HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

1. Multi-thread operation support?

The Pi is supported for multi-thread operation.

1. CPU frequency



1. Communication between multiple Pi

Serial data transfer

“Serial is a data link between the Pi and one other device.” GPIO 14 and 15 (pin 8 and 10)

The Pi uses GPIO 14 (pin 8) for TXD and GPIO 15 (pin 10) for RXD.

Data is normally transmitted in 8-bit bytes with a start bit, eight data bits, no parity, and one stop bit.

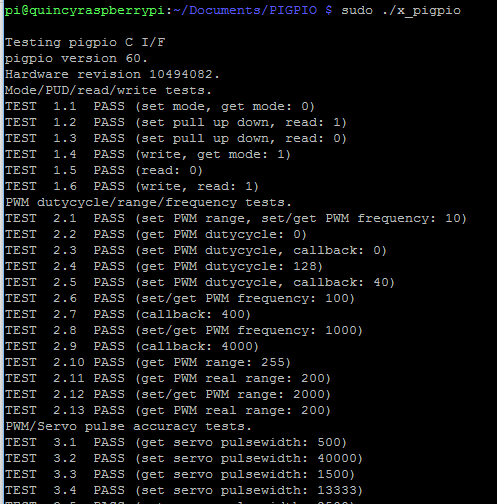
1. Pin control library

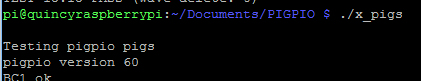
I installed both BCM2835 and PiGPIO pin library stated in homework. For PIGPIO’s test, it tested for PWM duty cycles and its pulse width. This would be useful to confirm the pin’s functionality before attaching HAT such as servo.

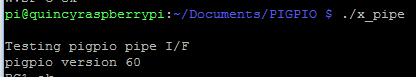
Pin Control Library

Usage

After Include <pigpio.h> in source files





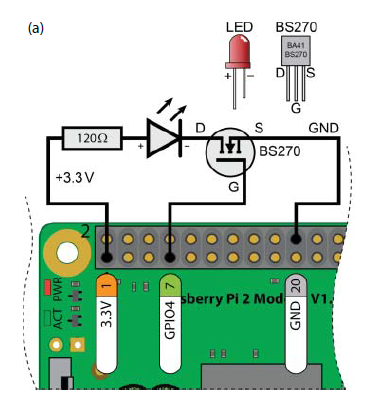


1. Controlling LED with GPIO

Following the textbook, I successful build the circuit below and used the Pi to control the LED.

Pin 1 – Vdd, Pin 7 – GPIO4, Pin 20 – GND





1. Heat

<http://lazydroid.com/2013/05/raspberry-pi-monitoring-cpu-temperature-with-rrdtool/>

<http://www.erijas.com/monitoring-cpu-temperature-raspberry-pi/>

<https://www.cyberciti.biz/faq/howto-linux-unix-start-restart-cron/>

This reference allow me to monitor the temperature of the rpi throughout the day. I am trying to run the raspberry pi 24/7. And the Pi3’s processor can get really hot, and damage the board.

However, I have not yet figure out how to monitor the temperature following the steps provided in the reference, more exploration is needed.

I can get the current temperature on the processor by using the command below.

or

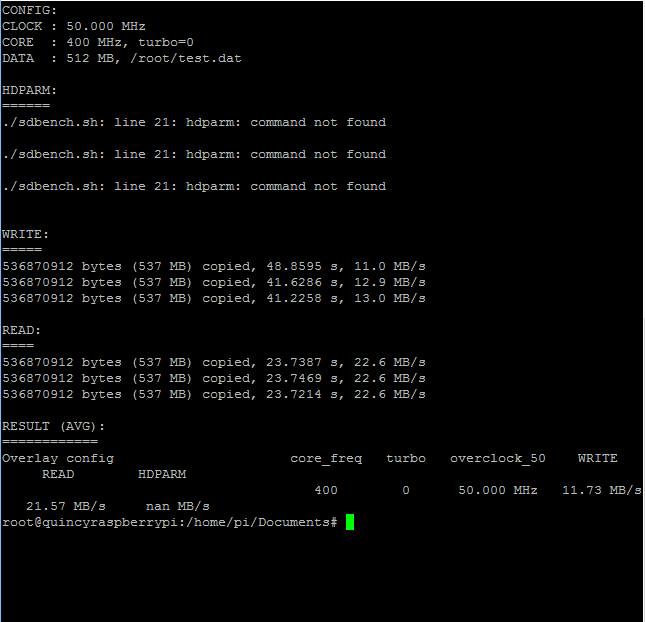


On the spec, “the LAN9512 is specified by the manufacturers as being qualified from 0 - 70°C, while the AP (Application Processor (Broadcom BCM2835), CPU of the board) is qualified from -40°C to 85°C.”

1. SD Benchmarks

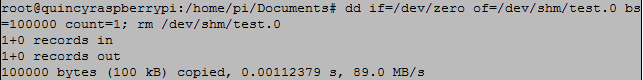
Following the SD card benchmark, the result is shown below. (Sdbench.sh)

The result shown the read and write speed on the SD card, without overclocking the SD card to 100 MHz. The SD card can fully utilize by overclocking it in with command



1. Read and write speed

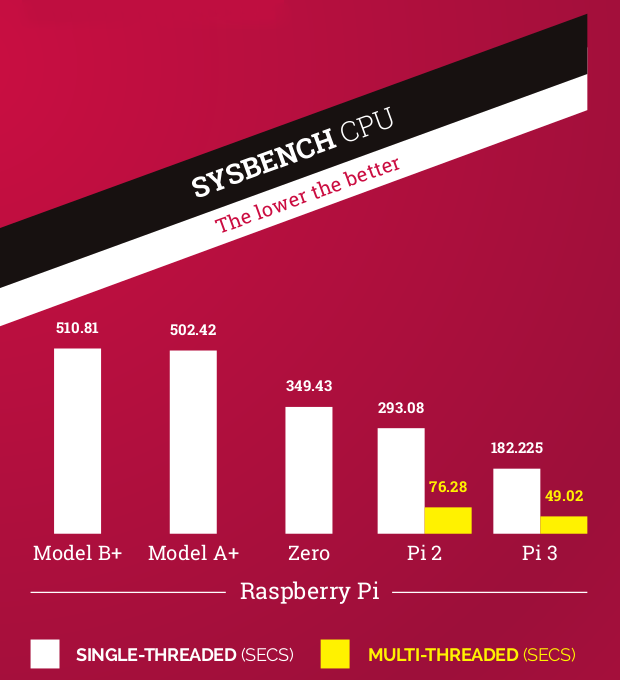
The read and write speed tested by copying a file would be testing for cache’s speed. This speed can be change by overclocking the rpi, however it might also reduce the lifespan of the rpi. Many people have shown the overclocking rpi benchmarks online. However, my application does not require overclocking.



1. Power drawn

According to the raspberry pi official cite, the 3.3 V rail can supply a maximum of 50 mA. And the 5V rail “appears to passed straight through from the USB and the current is therefore limited to whatever the USB port can supply minus the current being drawn by the board.”

|  |  |  |
| --- | --- | --- |
|  |  | Pi3 B (Amps) |
| Boot | Max | 0.75 |
|  | Avg | 0.35 |
| idle | Avg | 0.30 |
| Video playback (H. 264) | Max | 0.55 |
|  | Avg | 0.33 |
| Stress | Max | 1.34 |
|  | Avg | 0.85 |



# Conclusion

# References

<http://abyz.co.uk/rpi/pigpio/download.html>

<https://www.raspberrypi.org/documentation/configuration/device-tree.md>

<https://github.com/stripcode/pigpio-stepper-motor>

<https://www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/>

<https://www.raspberrypi.org/forums/viewtopic.php?f=63&t=144493>

<http://www.nmacleod.com/public/sdbench.sh>

<http://raspberrypi.stackexchange.com/questions/5477/why-is-my-pi-running-at-700mhz-all-the-time>

<http://raspberrypi.stackexchange.com/questions/51615/raspberry-pi-power-limitations>

<https://www.raspberrypi.org/help/faqs/#powerReqs>

<http://raspberrypi.stackexchange.com/questions/51615/raspberry-pi-power-limitations>

<https://www.raspberrypi.org/help/faqs/#performanceOperatingTemperature>

<https://www.raspberrypi.org/forums/viewtopic.php?f=91&t=34994>

Exploring Raspberry Pi Interfacing to the Real World with Embedded Lunux – Derek Molloy