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

**Uname -a**

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. 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

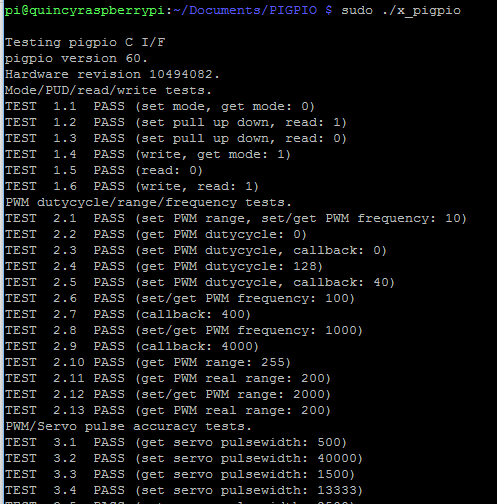
I installed both BCM2835 and PiGPIO pin library stated in homework. For PIGPIO’s test, it tested for PWM duty cycles and its pulse width. This would be useful to confirm the pin’s functionality before attaching HAT such as servo.

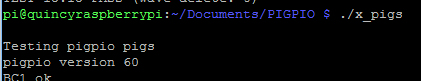
Pin Control Library

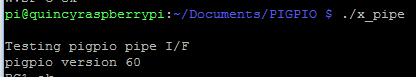
Usage

After Include <pigpio.h> in source files

gcc -Wall -pthread -o prog prog.c -lpigpio -lrt  
sudo ./prog







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

$ cd /sys/class/gpio

$ echo 4 > export

$ cd gpio4

$ echo out > direction

$ echo 1 > value // turn on LED

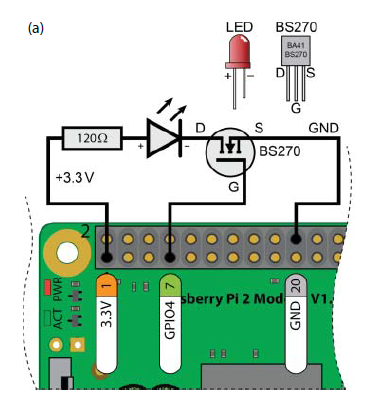
$ echo 0 > value // turn off LED

$ cat direction // to read the state of GPIO

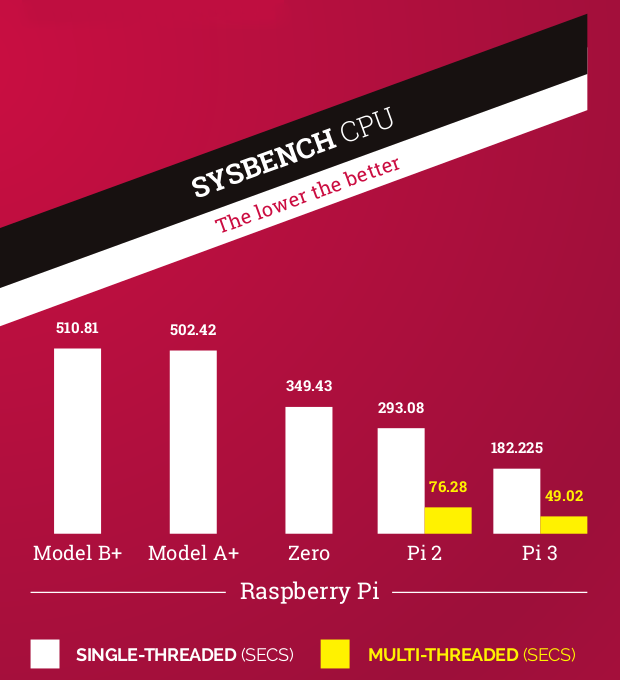
$ cat value

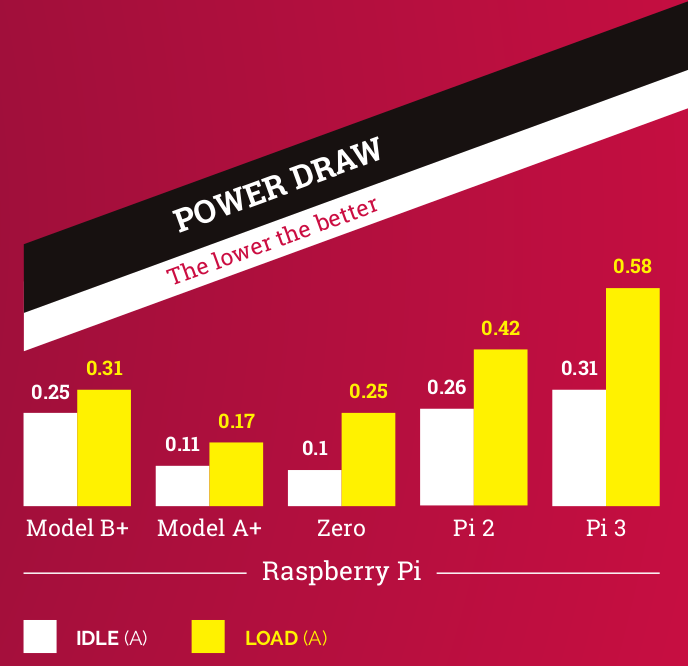
$ echo 4 > unexport





1. Benchmarks





**8 - Characterization question**

**Real time response capability**

Quality of PWM control

Pin functions

Power consumption

# 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/>

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