INTRO TO RPI (PART 2)

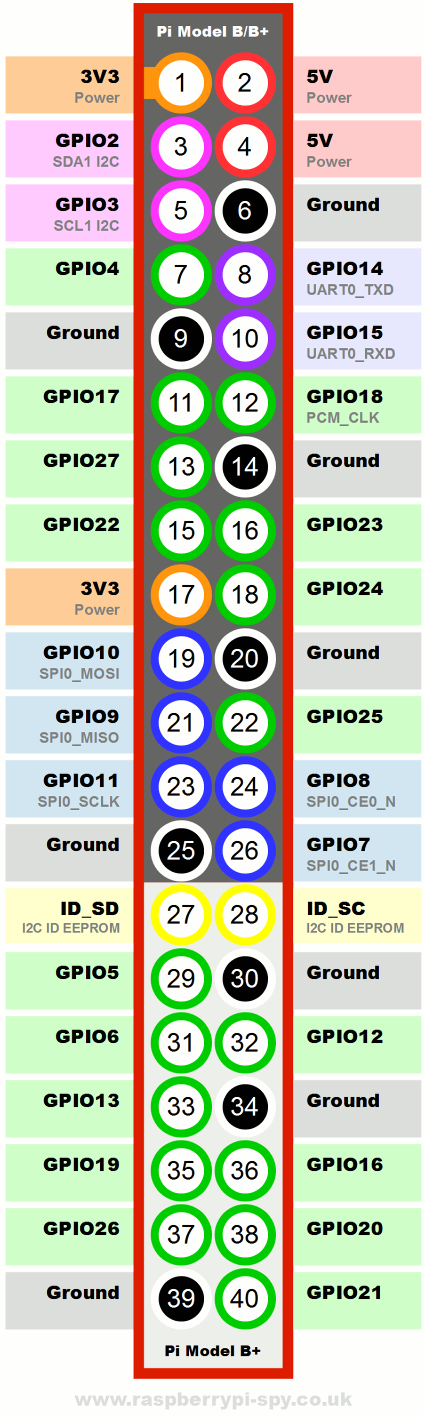
BY SUTD IEEE

AGENDA

- Using Firebase with RPi (the right way)
- Running a Web Server on the RPi

RPI GPIO

https://pinout.xyz/#



Pi Model B/B+

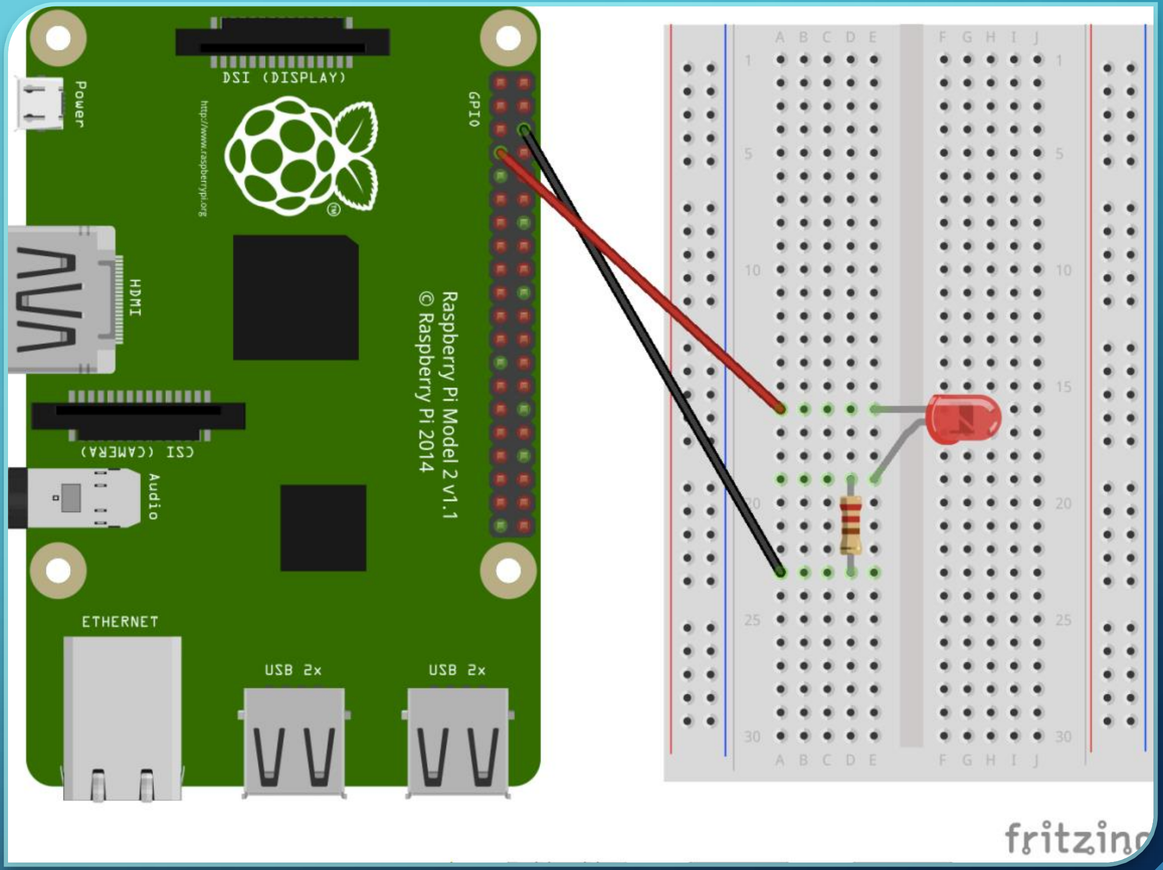
3V3 Power	1	2	5V Power
GPIO2 SDA1 I2C	3	4	5V Power
GPIO3 SCL1 I2C	5	6	Ground
GPIO4	7	8	GPIO14 UART0_TXD
Ground	9	10	GPIO15 UART0_RXD
GPIO17	11	12	GPIO18 PCM_CLK
GPIO27	13	14	Ground
GPIO22	15	16	GPIO23
3V3 Power	17	18	GPIO24
GPIO10 SPI0_MOSI	19	20	Ground
GPIO9 SPI0_MISO	21	22	GPIO25
GPIO11 SPI0_SCLK	23	24	GPIO8 SPI0_CE0_N
Ground	25	26	GPIO7 SPI0_CE1_N
ID_SD I2C ID EEPROM	27	28	ID_SC I2C ID EEPROM
GPIO5	29	30	Ground
GPIO6	31	32	GPIO12
GPIO13	33	34	Ground
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
Ground	39	40	GPIO21

Pi Model B+

www.raspberrypi-spy.co.uk

ACTIVITY SETUP

- Connect +ve lead of LED (Longer leg) to BCM26
  - Refer to https://pinout.xyz/# !
- Connect a resistor from the –ve lead of the LED to an empty space
- Connect the resistor to a GND pin
  - Refer to https://pinout.xyz/# !



FIREBASE (THE RIGHT WAY)

- ‘Database Secrets’ is deprecated
- ‘python-firebase’ module not updated in 4 years
- The right way?
    - Firebase Admin SDK (Python)

FIREBASE (THE RIGHT WAY)

- Open your Firebase Console
- Navigate to Project Settings >> Service Accounts
- Generate new Private Key >> Save that file in your project folder

FIREBASE (THE RIGHT WAY)

- In the Firebase Console:
- Open Database Tab
- Navigate to ‘Rules’
- Change it to this:

```
{
  "rules": {
    ".read": "auth.uid === ‘some_UID’”,
    ".write": "auth.uid === ‘some_UID’”
  }
}
```

- Publish

FIREBASE (THE RIGHT WAY)

- How to install?
  - `sudo pip3 install firebase-admin`
- Create a Python file
  - `import firebase_admin`
  - `from firebase_admin import credentials`
  - `from firebase_admin import db`

FIREBASE (THE RIGHT WAY)

```
cred = credentials.Certificate(‘PATH_TO_KEY')
firebase_admin.initialize_app(cred, {
    'databaseURL':
'https://your_database.firebaseio.com/',
    'databaseAuthVariableOverride': {
        'uid': ‘your_uid'
    }
})
```

FIREBASE (THE RIGHT WAY)

- To use the database:
  - `mydatabase = db.reference()`
  - Get:
    - `mydatabase.get()`
    - `mydatabase.child(‘some_child’).get()`
  - Set:
    - `mydatabase.child(‘some_child’).set( { “key”: “value” } )`
  - Update
  - Push

FIREBASE (THE RIGHT WAY)

- Refer to the template to learn how to continuously get a certain database entry

FIREBASE (THE RIGHT WAY)

- Your task:
  - Write another file that asks for an `input` (on/off) and updates a particular database entry using `.set()` `->` On your computer
  - Run a script on the RPi to continuously read that same database entry and based on that turn an LED on/off.

FLASK SERVER

- Flask is a Python framework that allows you to manage your web servers
  - It is not a web server! (Though it comes with a test server for … well, testing)
- Install:
  - virtualenv:
    - `sudo pip3 install virtualenv`
    - `mkdir` *`myproject`*
    - `cd` *`myproject`*
    - `virtualenv` *`myvenv`*
    - `.` *`myvenv`*`/bin/activate`
    - **To deactivate: type** `deactivate`
  - flask:
    - `pip3 install Flask`
    - **To run: type**
      `flask run --host=0.0.0.0 --port=8080`

FLASK SERVER

- Refer to ‘flaskapp.zip’ in for example

THE END