

A decorative graphic on the left side of the slide, consisting of a network of white lines and circles on a blue gradient background, resembling a circuit board or a neural network.

INTRO TO RPI (PART 1)

BY SUTD IEEE

AGENDA

- Setting up your RPi
 - Install Raspbian
 - Configure Wi-Fi
 - Set static IP
 - Setup SSH
 - Setup VNC
- Using the RPi through the Terminal
- Using the RPi like an Arduino
 - GP I/O Pins with Python

WHAT'S AN RPI?!

- Single Board Computer
- Runs Linux (Most of the time)
- Small
- Access to GP I/O Pins (Input and Output)
 - Like an Arduino



USES OF RPI'S

- Web Servers
- Cloud Servers
- Home Automation
- Home Security
- Arcade Games
- Supercomputing (Clusters)
- Cryptocurrency mining
- Robotics
- ...



INSTALLING RASPBIAN ON THE RPI'S SD CARD

- Copy Etcher and the Raspbian image to your computer
- Connect the microSD Card
- Open Etcher
- Select the Raspbian image
- Click 'Flash!'

SETTING UP WI-FI

- Raspbian does not have a GUI that supports WPA2 Protocol
 - Cannot connect to SUTD_Student directly!
 - Need to configure it manually
- **Configure network in** `/etc/wpa_supplicant/wpa_supplicant.conf`

SETTING UP WI-FI

- Open Terminal
- `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf`

- Add this to the file:

```
network={
    ssid="SUTD_Student"
    key_mgmt=WPA-EAP
    eap=PEAP
    identity="100XXXX"
    password="YOUR_PASSWORD"
    phase1="peaplabel=0"
    phase2="auth=MSCHAPV2"
}
```

- Reboot

SET STATIC IP ADDRESS

- What's an IP(v4) Address?
- Why? So we can connect to the RPi at the same address every time.
- Open Terminal:

```
ip -4 addr show | grep global
```

```
ip route | grep default | awk '{print $3}'
```

```
cat /etc/resolv.conf
```


SET STATIC IP ADDRESS

- `sudo nano /etc/dhcpd.conf`

- **Add this to the file:**

```
interface wlan0
static ip_address=10.1.1.31/24
static routers=10.1.1.1
static domain_name_servers=10.1.1.1
```

- Reboot

WHAT IS SSH (SECURE SHELL)

- It is a protocol that allows a computer to remotely log in to another through the Terminal, allowing you to use that computer.
 - In this case the RPi

SETTING UP SSH

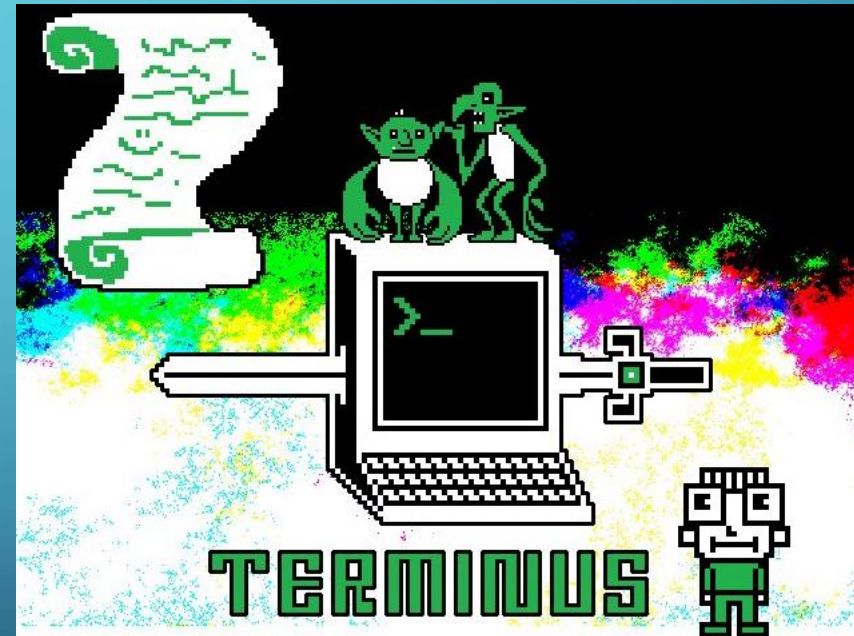
- Open 'Applications Menu' (Top Left) >> 'Preferences' >> 'Raspberry Pi Configuration'
- Select the 'Interfaces' tab
- Enable SSH

SETTING UP VNC (VIRTUAL NETWORK COMPUTING)

- RealVNC – Free!
- Open 'Applications Menu' (Top Left) >> 'Preferences' >> 'Raspberry Pi Configuration'
- Select the 'Interfaces' tab
- Enable VNC

USING THE RPI THROUGH THE TERMINAL

- www.mprat.org/Terminus/
- Learn Linux commands to navigate and control the file system.



NOW FOR THE ELECTRONICS STUFF

- Program the RPi's GP I/O Pins
 - General Purpose Input/ Output
 - <https://pinout.xyz/#>
- Use it like an Arduino
- Can be done using Python, C, C++, Bash, etc.

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+			

GP I/O WITH PYTHON

- Install RPi.GPIO

- `sudo pip install rpi.gpio`

- In Python

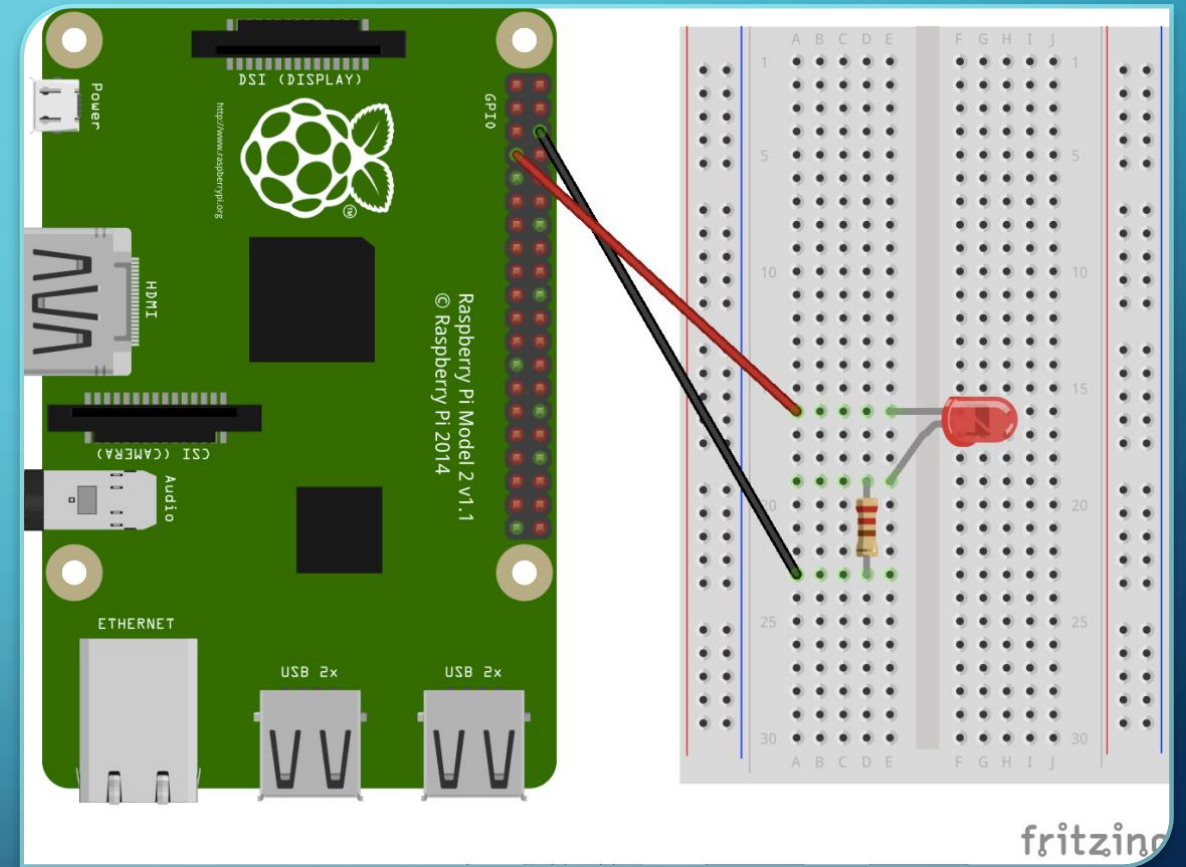
- `import RPi.GPIO as GPIO`

GP I/O WITH PYTHON

- `GPIO.setmode(MODE)` => `MODE` is `GPIO.BOARD` or `GPIO.BCM`
- `GPIO.setup(channel, GPIO.HIGH)` => `channel` can be a list of channels
- `GPIO.setup(channel, GPIO.HIGH, initial=GPIO.HIGH)`
- `GPIO.input(channel)`
- `GPIO.output(channel)` => `channel` can be a list of channels
- `GPIO.PWM(channel, frequency)`
- `GPIO.cleanup()`

ACTIVITY #1: BLINKING LED

- 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/#> !



ACTIVITY #1: BLINKING LED

- `import RPi.GPIO as GPIO`
- `from time import sleep`
- `import sys`
- `GPIO.setmode(GPIO.BCM)`
- `GPIO.setup(26,GPIO.OUT)`
- `GPIO.output(26,GPIO.HIGH)`
- `sleep(1) // Sleep for 1s`

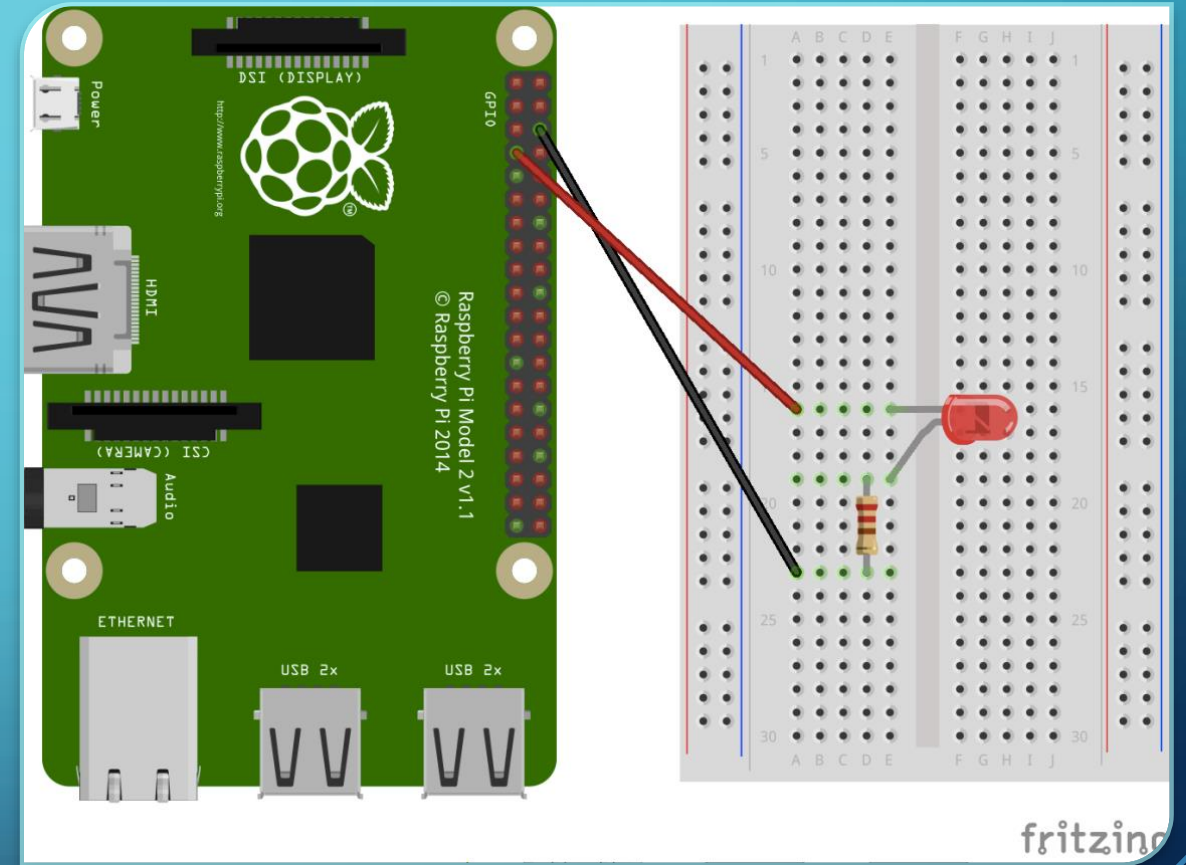
ACTIVITY #1: BLINKING LED

Try:

```
while True:
    # Do Something
finally:
    GPIO.cleanup()
    sys.exit()
```

ACTIVITY #2: FADING LED

- 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/#> !

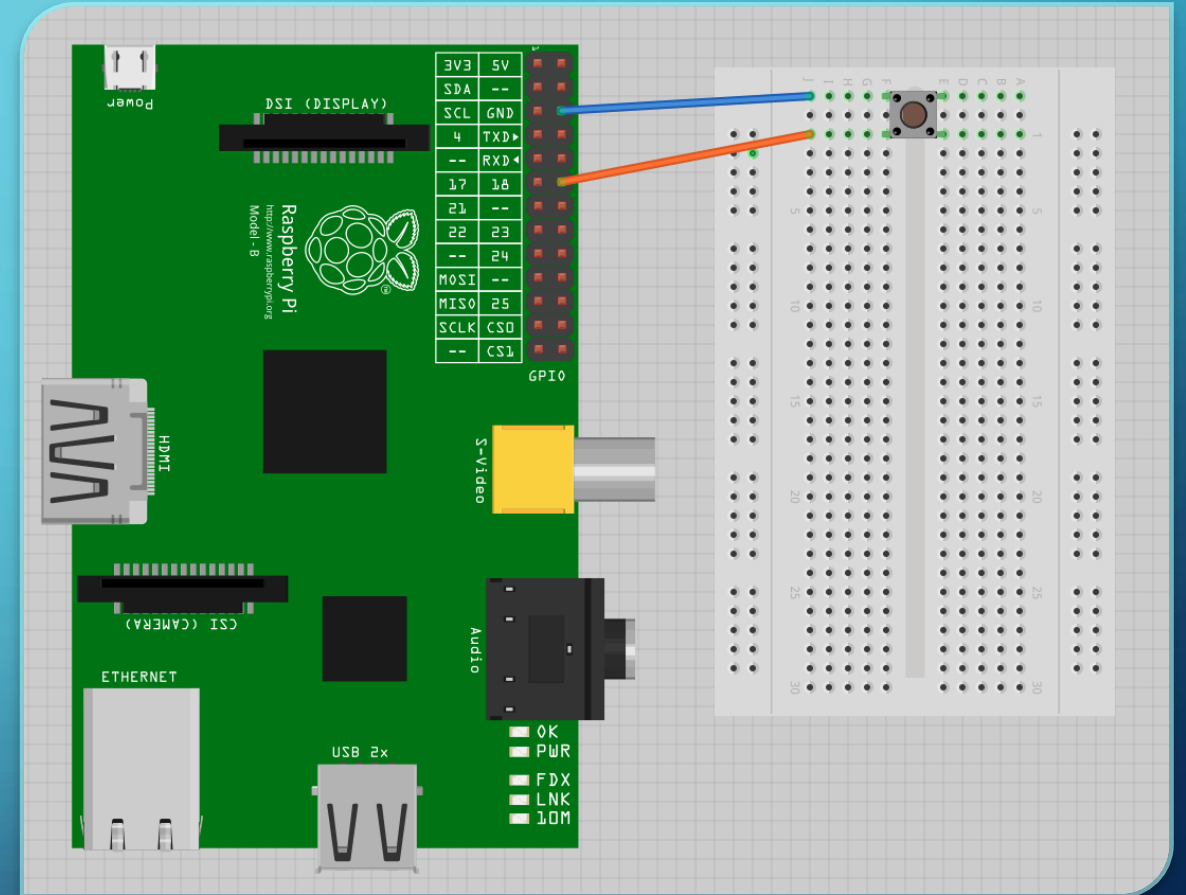


ACTIVITY #2: FADING LED

- `pwm = GPIO.PWM(26,1000)`
- `pwm.start(0)`
- `pwm.ChangeDutyCycle(x)`
- `for i in range(100):`

ACTIVITY #3: PUSH BUTTON

- Connect one end of the button to BCM26
- Connect the other end on the same side to GND



ACTIVITY #3: PUSH BUTTON

- `GPIO.input(26)`

ACTIVITY #4

- Use a push button to toggle an LED on and off!
- TRY!



THE END