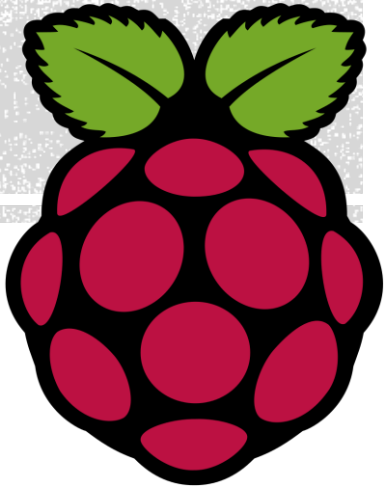
RASPBERRY PI WORKSHOP



ThinkLAB

www.thinklab.ph

secretariat@thinklab.ph

facebook.com/thinklab.ph

SparkLab Innovation Center

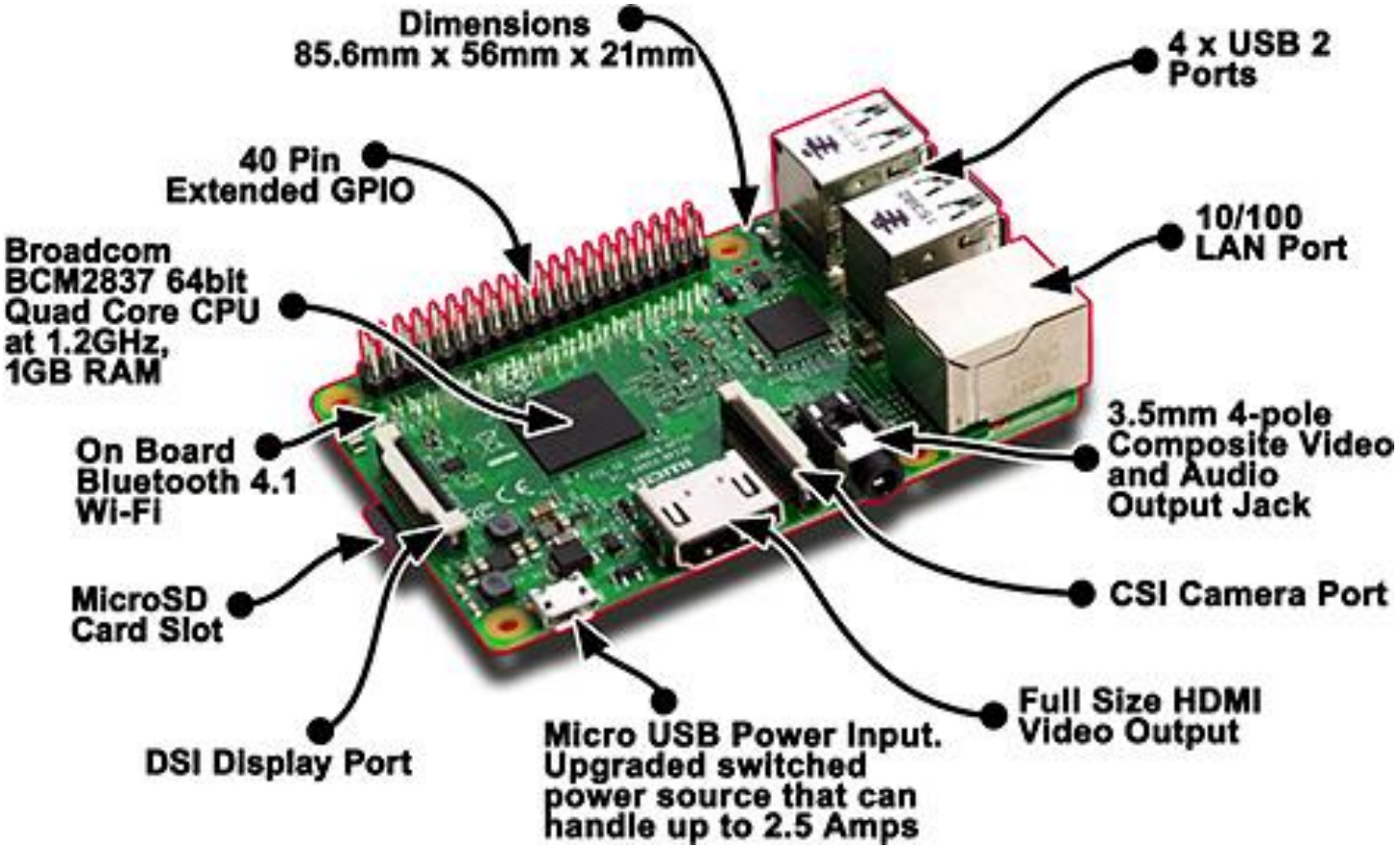
DAY 1 – INTRODUCTION TO RPI AND BASIC APPLICATION

- What is Raspberry Pi?
- Parts of the Raspberry Pi System
- Set-up Configuration
- Interfacing Basics
- Linux File System
- Python Programming
- Control Structures
- GPIO and other basic applications

DAY 2 – INTERFACING WITH THE RASPBERRY PI, IOT AND OTHER ADVANCED APPLICATIONS

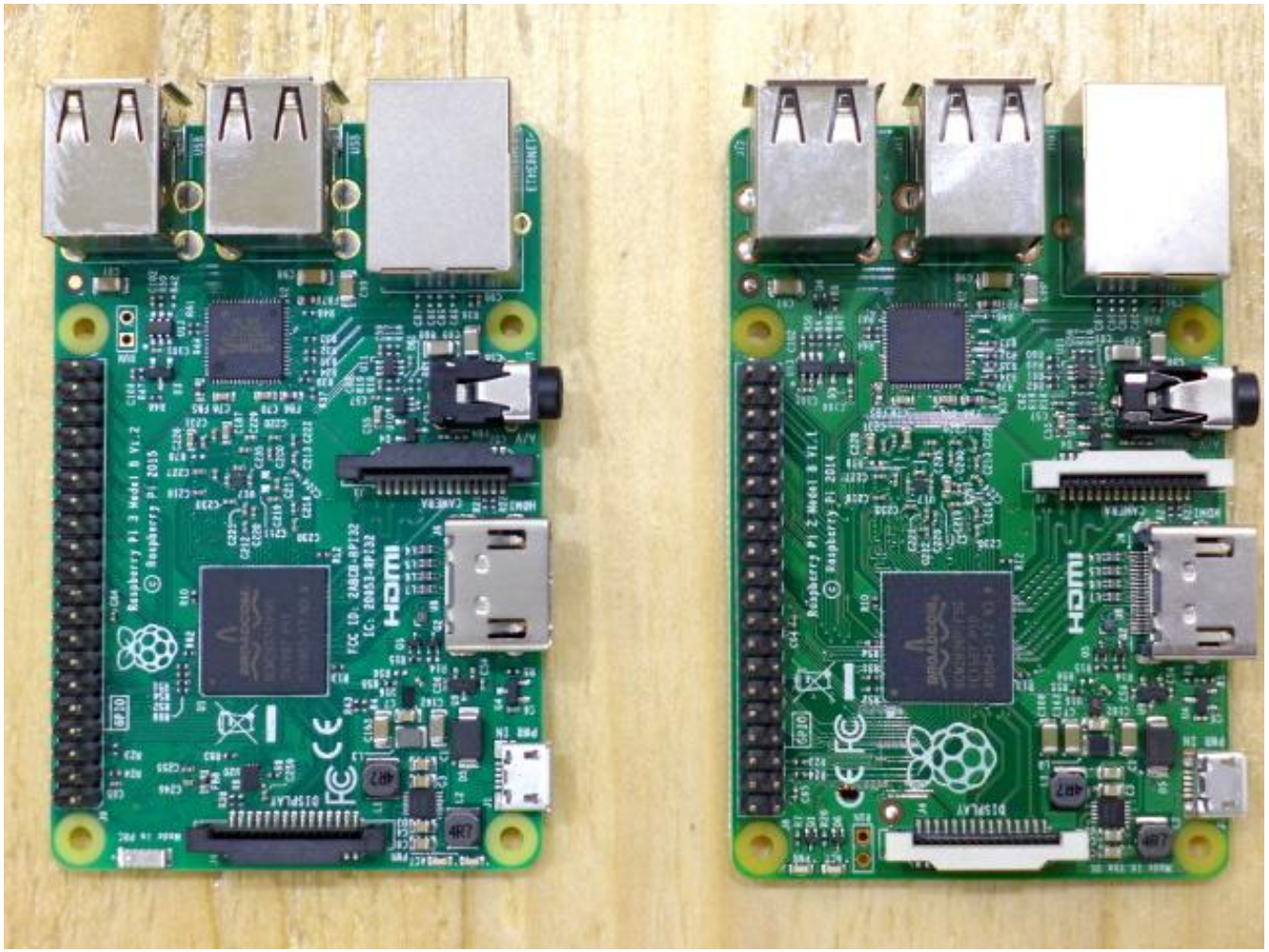
- Interfacing with Arduino (LM35)
- Internet of Things

WHAT IS RASPBERRY PI?



RAPBERRY PI 2 AND 3

Raspberry Pi 3

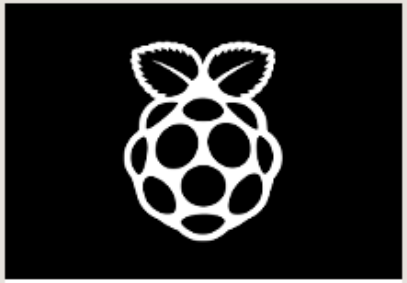


Raspberry Pi 2

RASPBERRY PI SPECIFICATIONS

	Raspberry Pi 2 Model B	Raspberry Pi 3 Model B
Chip	Broadcom BCM2835 (SoC)	Broadcom BCM2837 (SoC)
CPU	32-bit Quad-core Cortex-A7, 900 MHz	64-bit Quad-core ARM Cortex-A53, 1.2 GHz
GPU	Broadcom VideoCore IV, 250 MHz	Broadcom VideoCore IV, 400 Mhz
Memory	1GB DDR2, 450 MHz	1GB DDR2, 900 MHz
USB 2.0	4 ports	4 ports
Connectivity	10/100 Ethernet jack	10/100 Ethernet jack 802.11n Wireless LAN Bluetooth 4.1 Bluetooth Low Energy (BLE)
Video Output	HDMI, Composite Video (3.5mm jack)	HDMI, Composite Video (3.5mm jack)
Audio Output	3.5mm jack, HDMI	3.5mm jack, HDMI
Onboard Storage	microSD Slot (32GB max)	microSD Slot (64 GB)
Dimension	8.6cm x 5.4cm x 1.5cm	8.6cm x 5.4cm x 1.5cm

RASPBERRY PI OPERATING SYSTEMS



NOOBS

RASPBIAN

UBUNTU MATE

SNAPPY UBUNTU CORE

WINDOWS 10 IOT CORE

OSMC

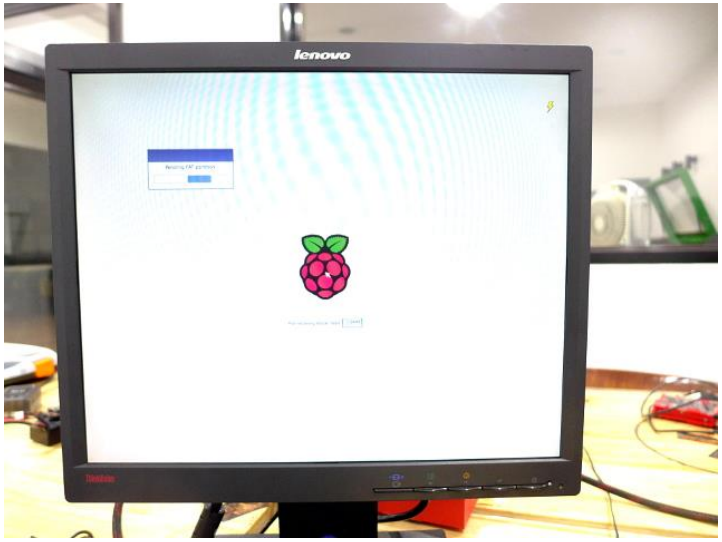
LIBREELEC

PINET

RISC OS

WEATHER STATION

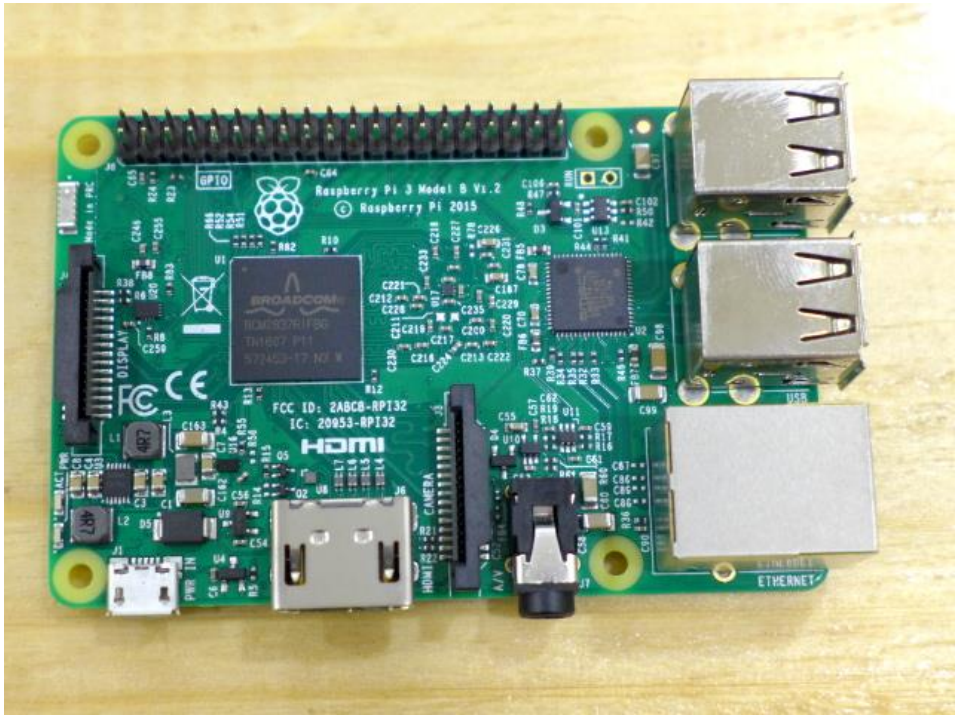
DIRECT SETUP



Monitor

HEADLESS SETUP

Power Adaptor



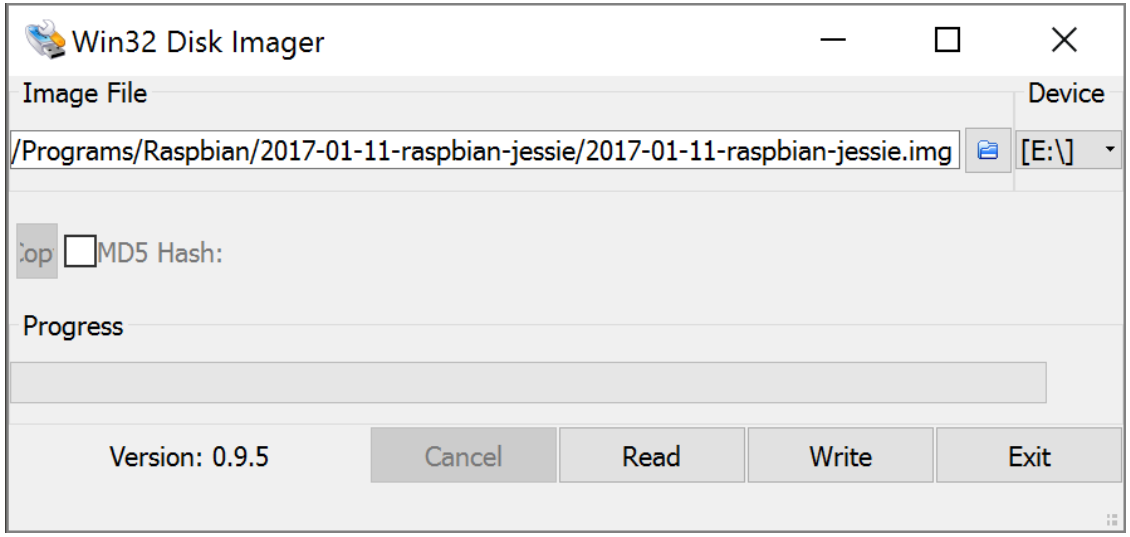
SD Card

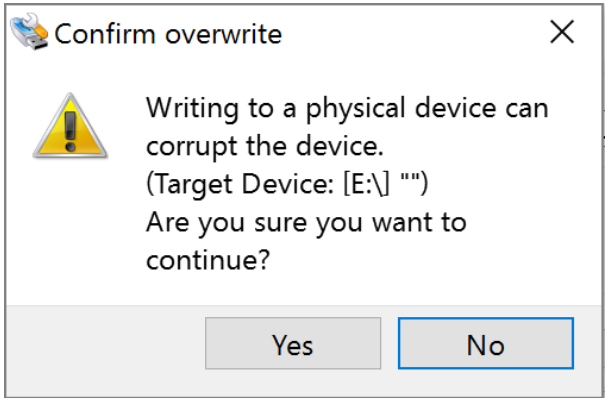
LAN Cable (Optional)

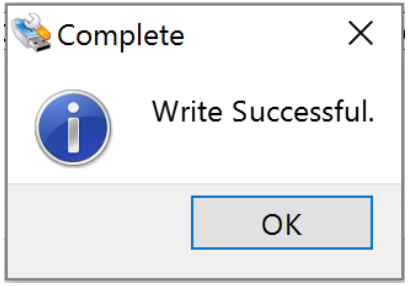
GETTING STARTED (RASPBIAN)

1. Download Raspbian and Extract Zip Files
   - http://www.raspberrypi.org/downloads/
   - Raspbian Jessie with Pixel (1.42 GB)
   - Raspbian Lite (292 MB)
2. Install win32DiskImager
   - https://sourceforge.net/projects/win32diskimager/
3. Run as Administrator win32DiskImager
4. Select the drive of the SD Card
5. Select the image (img) on the Extracted Files
6. Click Write

GETTING STARTED (RASPBIAN)

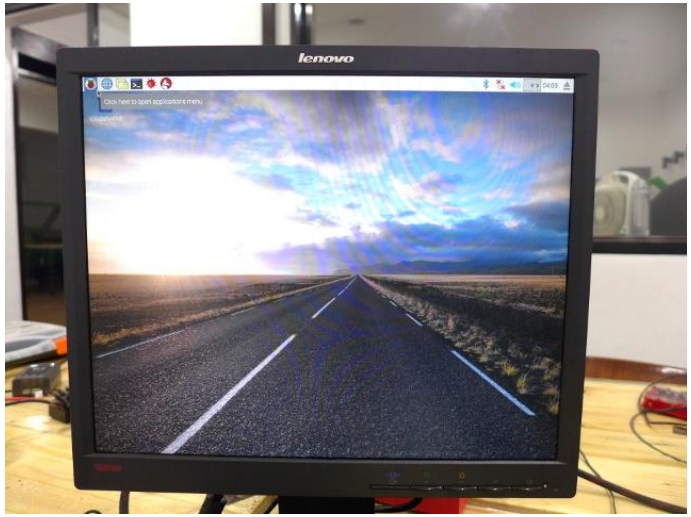






GETTING STARTED (RASPBIAN)

7. Wait…
8. Insert SD Card on Raspberry Pi (RPi)
9. Connect the RPi to the Monitor, Keyboard & Mouse.
10. Power the RPi by plugging the mini-USB to a power source (i.e. Charger, Power Bank, USB Port
11. Wait…
12. Finished!!!



WHAT CAN THE RASPBERRY PI DO?

- Office (Documents, Spreadsheets, Presentation)
- Play videos and music
- Internet Browsing
- Programming (Python, Java, C, C++, etc.)
- Database server
- Physical Computing (GPIO)

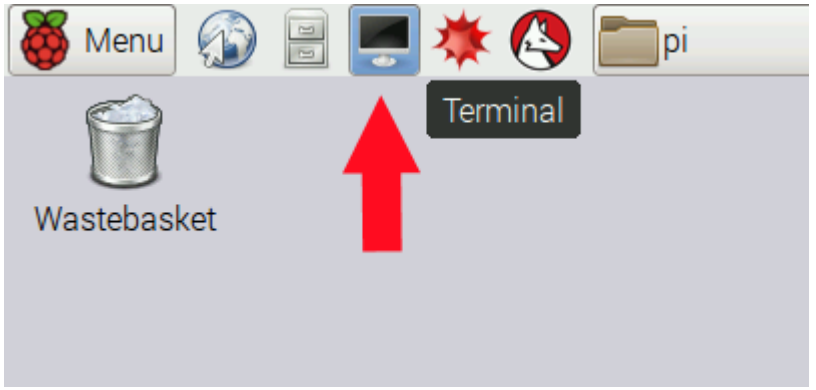
GOAL

Control the Raspberry Pi 3 remotely.

Remotely – using other computers.

Get used to the Terminal.

CHANGING THE KEYBOARD SETUP



- Open Terminal
- Command: sudo raspi-config
- Select '5 Internationalisation Options'
- Select 'Generic 105-key (Intl) PC'
- Select 'Other'
- Select 'English (US)'
- Select 'English (US)'
- Select 'The default keyboard layout'
- Select 'No compose key'
- Select 'No' – *Option for Ctrl+Alt+Backspace to do something*
- Select 'Finish'

CHANGING THE PASSWORD OR RPI

Default

User: pi

Password: raspberry

1. Open Terminal
2. Command: passwd

CONNECTING TO WIFI (HEADLESS WAY)

1. Open Terminal
2. Command: sudo iwlist wlan0 scan
   Scans the area for Wireless Network (WiFi)
   Remember desired ESSID
3. Command: sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
4. Add to the end of the file

```
network={
  ssid="ESSIDName"
  psk="Password"
}
```

5. Command: sudo ifdown wlan0 && sudo ifup wlan0

SHUTDOWN & REBOOT

SHUTDOWN

sudo halt

REBOOT

sudo reboot

GETTING THE IP ADDRESS OF RPI

- Command: ifconfig

```
pi@raspberrypi:~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:09:ca:66
          inet6 addr: fe80::8e5:9b21:5dbe:e28/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:526 errors:0 dropped:0 overruns:0 frame:0
          TX packets:526 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:43152 (42.1 KiB)  TX bytes:43152 (42.1 KiB)

wlan0     Link encap:Ethernet  HWaddr b8:27:eb:5c:9f:33
          inet addr:192.168.254.30  Bcast:192.168.254.255  Mask:255.255.255.0
          inet6 addr: fe80::2c6b:8ce0:baf9:aa8b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:22792 errors:0 dropped:3143 overruns:0 frame:0
          TX packets:13027 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:22963435 (21.8 MiB)  TX bytes:1828425 (1.7 MiB)

pi@raspberrypi:~ $
```

REMOTE CONNECTION (SSH)

- Git-Bash
  1. Extract Portable Git
  2. Open PortableGit folder
  3. Run **git-bash.exe**

- Enable SSH on Raspberry Pi Configuration

- SSH (Secure Shell)
  - **Command:** ssh pi@[ip address of RPi]

ssh pi@192.168.43.12

REMOTE CONNECTION (SFTP)

- SFTP (SSH File Transfer Protocol)
  - Extension of the SSH to handle file transfers

- Programs: FileZilla, WinSCP

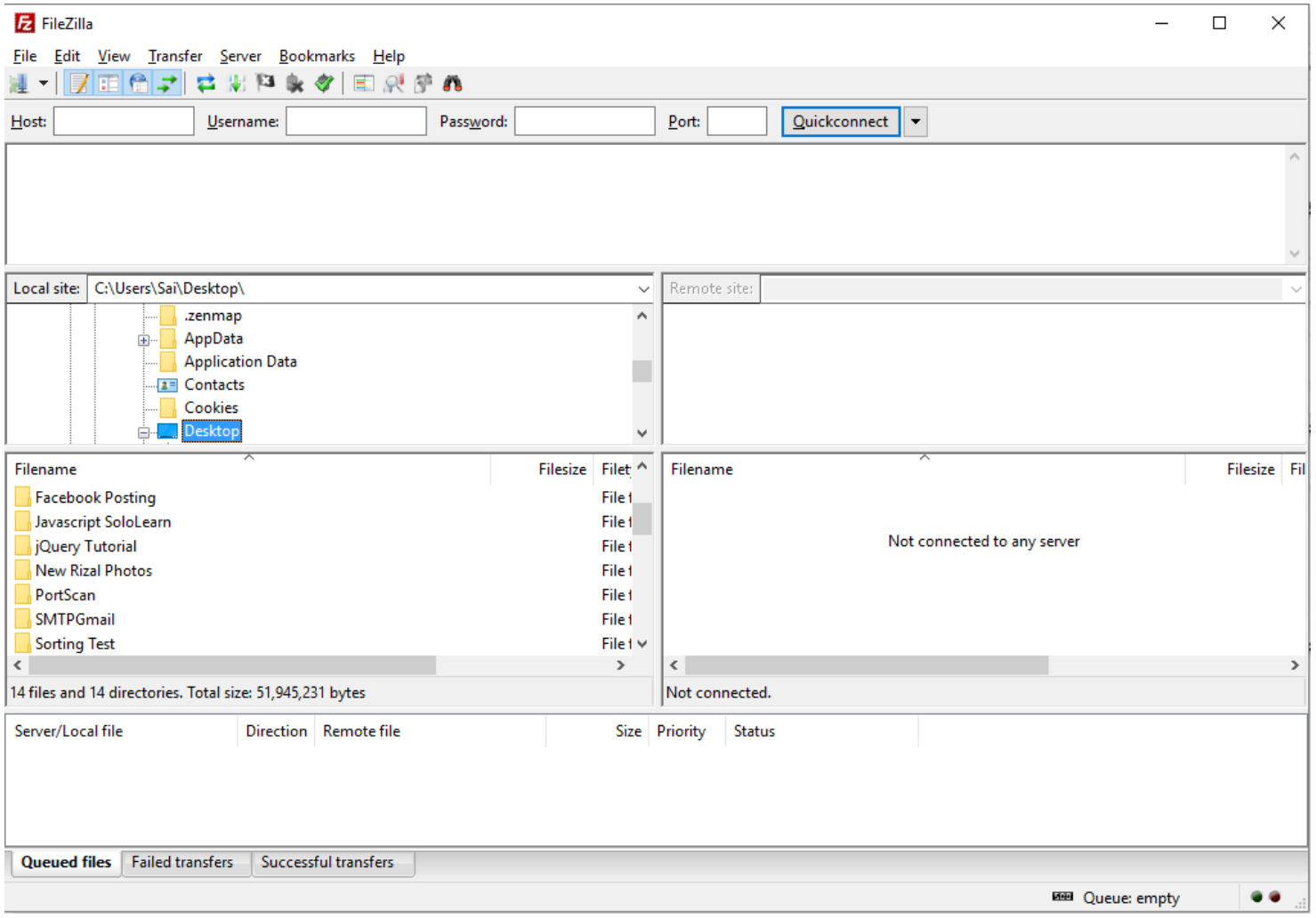
- FileZilla
  1. Extract FileZilla Zip (Portable Application)
  2. Run **filezilla.exe**

REMOTE CONNECTION (SFTP) - FILEZILLA

1. File → Site Manager
2. New Site
3. General Tab
   - Host: IP Address of Raspberry Pi
   - Protocol: SFTP – SSH File Transfer Protocol
   - Logon Type: Ask for password
   - User: pi
4. Connect

   Will ask for the password of your RPi

REMOTE CONNECTION (SFTP) - FILEZILLA



REMOTE CONNECTION (SFTP) - EXERCISE

Copy the following files
to the Desktop folder
of your Raspberry Pi

- example1080.mp4
- example2k.mp4

PLAYING VIDEO VIA SSH

- SSH to your Raspberry Pi

- Command
  1. cd
  2. cd Desktop
  3. omxplayer example1080.mp4

LINUX COMMANDS

- ls – list directory items
- cd – change directory
  - cd - goes to Home Directory
  - cd .. - goes back to the parent directory
  - cd /[Name of Directory] - goes to the desired directory
  - cd / - goes to the root directory
- ifconfig – gets the IP Config
- sudo – SuperUser Do
- wget – downloads a file

LINUX COMMANDS

- touch – creates a file
- cp – copy files/directory
- rm – remove/delete files/directory
- mkdir – make directory
- rmdir – remove directory

- nano – opens the nano text-editor

NEVER RUN THIS COMMAND

rm –rf /

GET HELP

[command] --help	Help for Command
man [command]	Command Manual

LINUX COMMAND EXERCISE

1. Change directory to Desktop
2. Create a directory named ‘Exercise1’
3. Change directory to ‘Exercise1’
4. Create a file ‘exercise1.txt’
5. Edit the contents of the file. Save and Exit.
6. Copy ‘exercise1.txt’ on the same folder with the name ‘exercise2.txt’

UPDATING RASPBIAN (LINUX)

sudo apt-get update

sudo apt-get upgrade

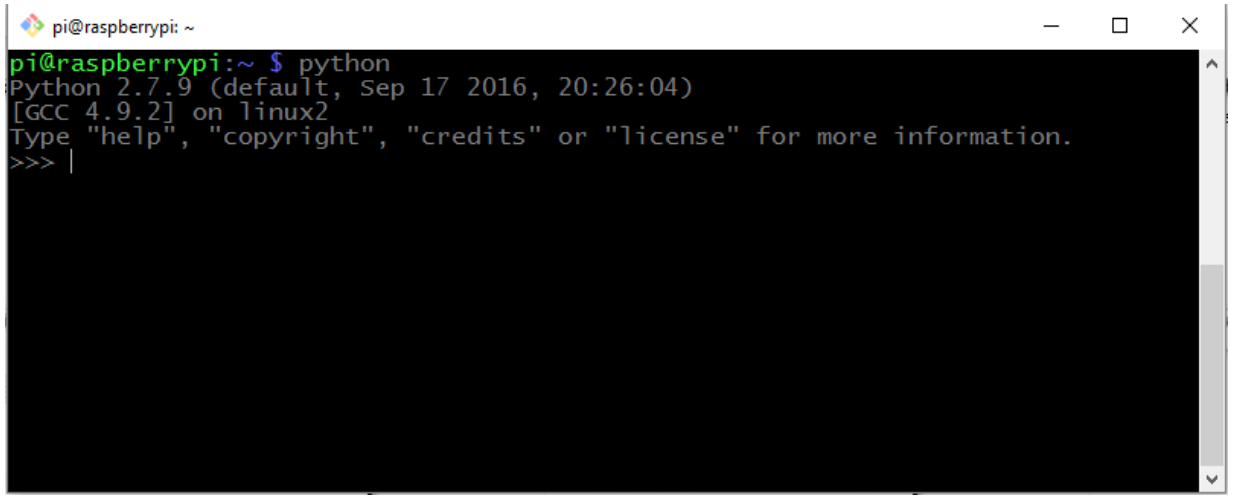
sudo apt-get dist-upgrade

PYTHON (PROGRAMMING)

- High Level Programming Language
- Emphasis on Code Readability
- Fewer lines of code, faster development

PYTHON PROGRAMMING

- Command: python

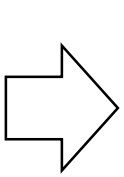


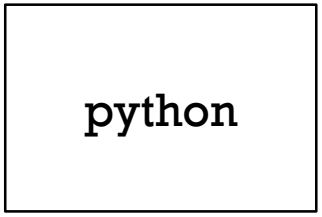
- >>> print “Hello World!”

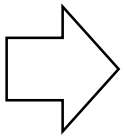
- Exit command: quit()

INTERPRET A SCRIPT

hello world.py







Hello World!

1. Create a new file “hello world.py”
2. Edit the file using nano, enter **print “Hello World!”**
3. Save and exit
4. Run the script by typing **python “hello world.py”**

PYTHON RESERVED WORDS

and	del	from	not	while
as	elif	global	or	with
assert	else	if	pass	yield
break	except	import	print	
class	exec	in	raise	
continue	finally	is	return	
def	for	lambda	try	

CODING STRUCTURE

- Indentation – focus on readability

```
w=5
for i in range w:
    print(w)
    print("Hello World!")

if w <= 5:
    print(w-1)
```

PYTHON VARIABLE TYPES

- Numbers (integer, float)
  - iNumber = 3
  - fNumber = 3.0
- Strings
  - sName = "Mark John Luke Matthew"
  - len(sName)
- List
  - itemList = ["Kangkong","Kalabasa","Talong","Sitaw", 1000]
  - itemList[0]
  - len(itemList)
- Tuple
  - itemTuple = ("Kangkong","Kalabasa","Talong","Sitaw", 2000)
  - itemTuple[0]
  - len(itemTuple)
- Dictionary
  - itemDict = {'Kangkong':9,'Kalabasa':6,'Talong':'Seven'}
  - itemDict['Kangkong']
  - len(itemDict)

LISTS

	Returns	Description
list[i] list[-i]	object on i	
list[1:2] list[3:]	objects on the specified range	
del list[i] del list[i:j]		removes the object in I
len(list)	number of objects in list	
list.append(obj)		adds the object on the end of the list
list.pop()	removed item	removes the last item
list.pop(i)	removed item	removes the item on i

DICTIONARY

	Description
dic = {}	Creates an empty Dictionary
dic = {'item1': 2, 'item2': 'hello'}	Creates a Dictonary with entries
dic['keyword'] = obj	Creates and entry to Dictionary Updates entry in Dictionary
obj = dic['keyword']	Assigns obj with the object with keyword in Dictionary

PYTHON CONTROL STATEMENTS (IF)

```
if expression1:
    statement(s)
elif expression2:
    statement(s)
elif expression3:
    statement(s)
else:
    statement(s)
```

PYTHON CONTROL STATEMENTS (IF)

```
var1 = 100
var2 = 0
if var1 == 200:
    print("I'm at 1")
    print(var1)
elif var1 == 150:
    print("I'm at 2")
    print(var1)
elif var1 == 100 and var2 == 1:
    print("I'm at 3")
    print(var1)
else:
    print("I'm at 4")
    print(var1)
```

LOOPING (WHILE LOOP)

```
while expression:
    statement(s)
```

```
i = 0

while i < 10:
    i = i + 1
    print(i)
```

LOOPING (FOR LOOP)

```
for i in range(10):
    print('Number: ', i)

for i in range(5, 10):
    print('Num: ', i)

for letter in 'Raspberry Pi':
   print('Current Letter :', letter)

words = ['Mosaic', 'Ecological',  'Cosmonaut']
for word in words:
   print('Current word :', word)
```

FUNCTIONS (STRUCTURE)

```
def functionname( parameters ):
    statement(s)
    return [expression]
```

FUNCTIONS (EXAMPLE)

```
def add(a, b):
    total = a+b
    return total

def multiply(a, b):
    product = a*b
    return product

a = 2
b = 3
print(add(a,b))
print(multiply(a,b))
```

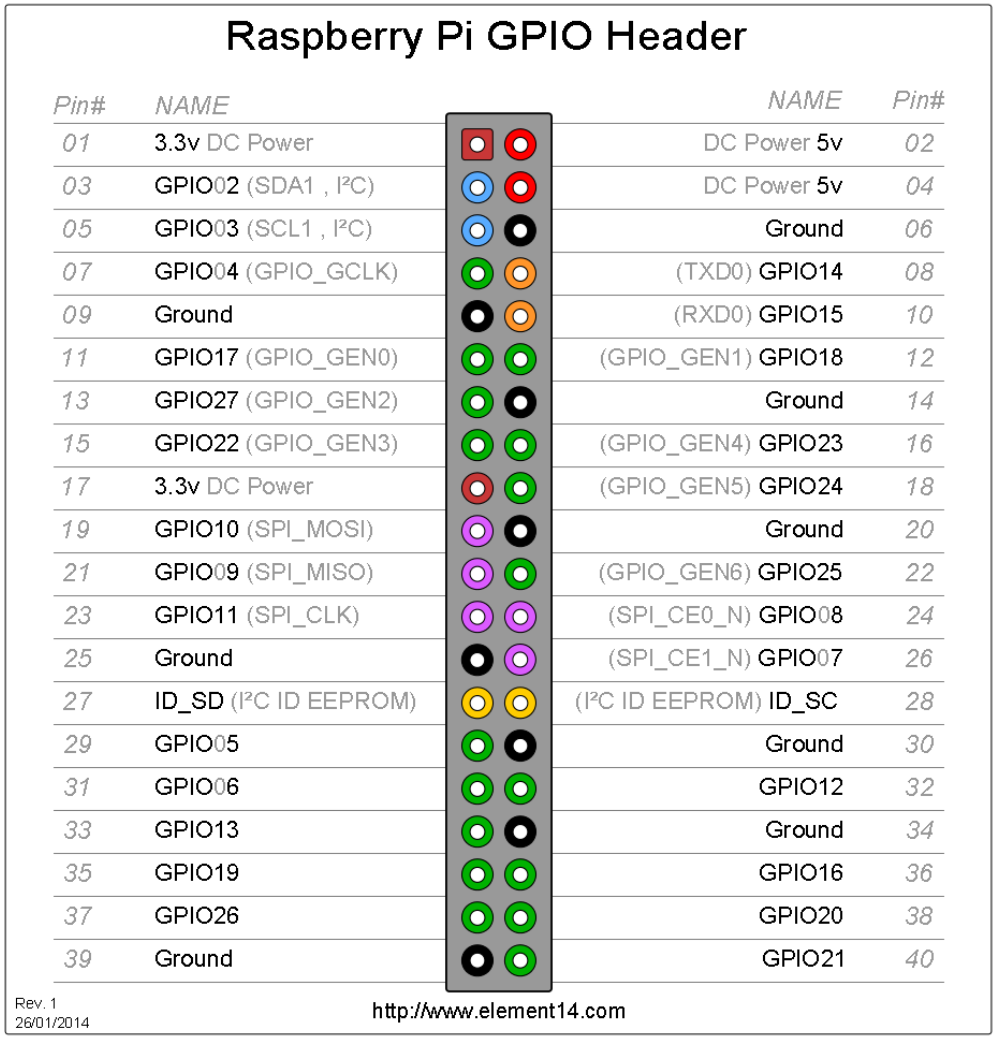
FUNCTIONS (MULTIPLE RETURN)

```
def addmultiply(a, b):
    total = a+b
    product = a*b
    return total, product

a = 2
b = 3
total, product = addmultiply(a, b)

print("total: ", total)
print("product: ", product)
```

GPIO (GENERAL PURPOSE INPUT/OUTPUT)



Input

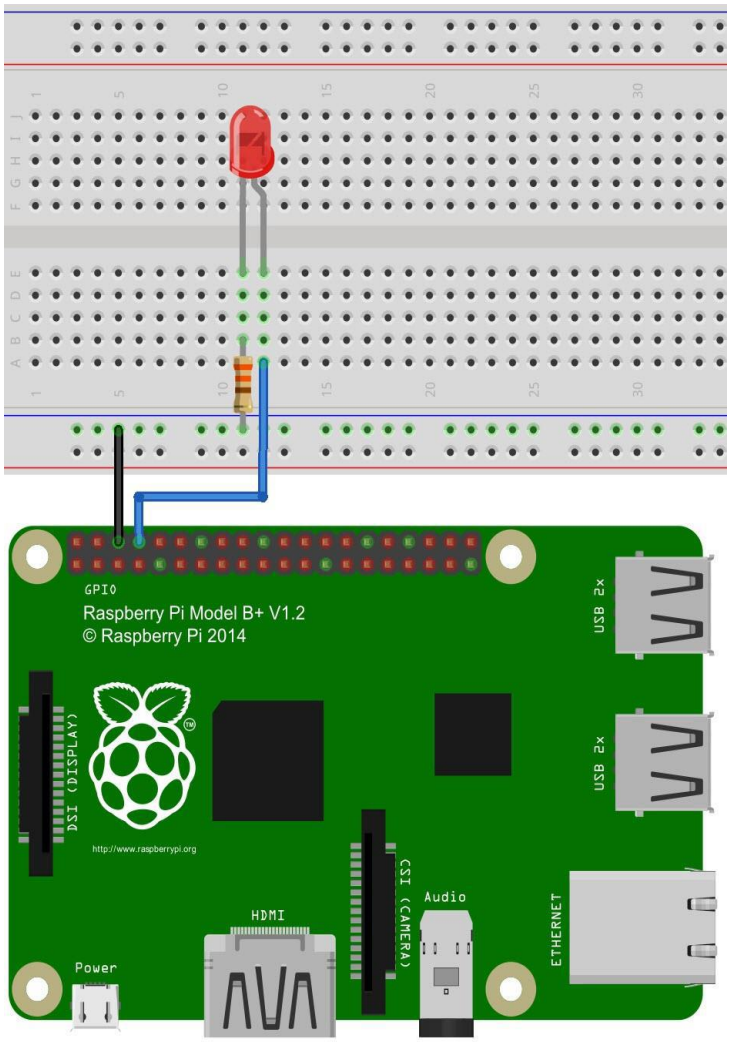
Output

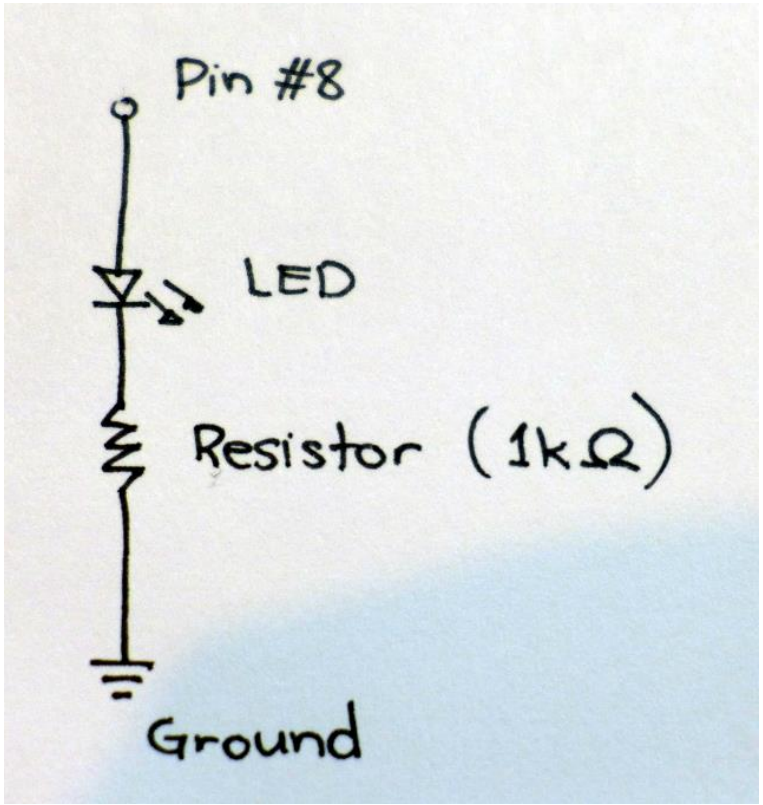
Alternate Functions

GPIO WARNING

- Do not short the GPIO pins to ground or 3.3V/5V.
- Logic high is 3.3V. Do not connect it with a 5V logic high system.
- Raspberry Pi does not have a protective circuit in its GPIO.

OUTPUT EXERCISE – LED BLINKER

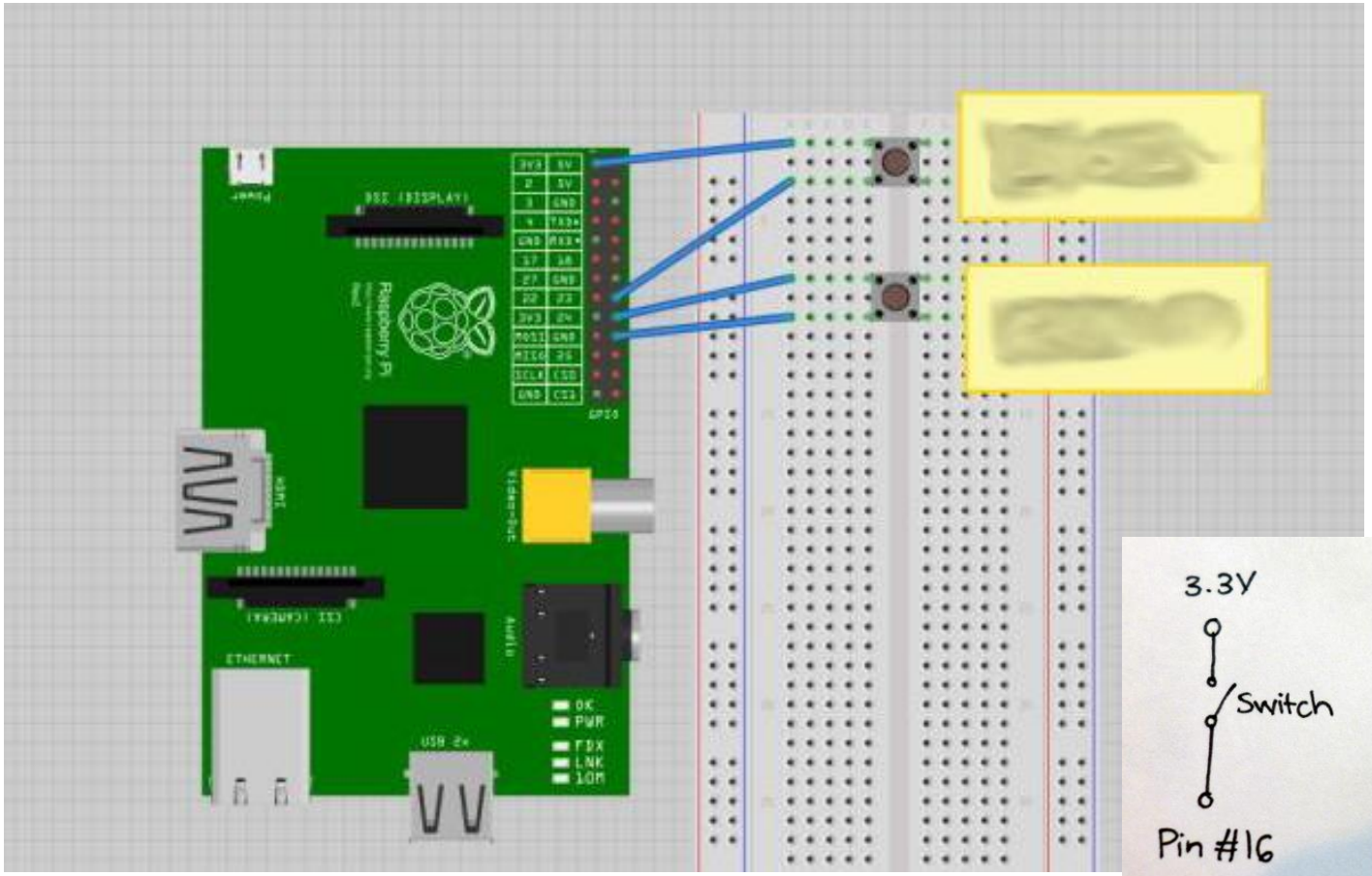


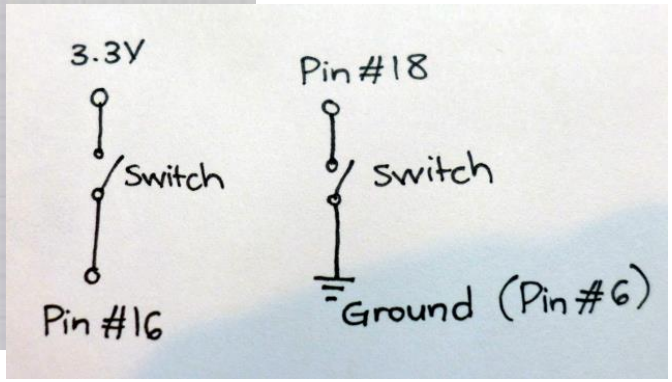


OUTPUT EXERCISE – LED BLINKER

```
1  import RPi.GPIO as GPIO
2  import time
3
4  GPIO.setmode(GPIO.BOARD)
5  GPIO.setup(8, GPIO.OUT)
6
7  try:
8      while True:
9          GPIO.output(8, True)
10         time.sleep(0.5)
11         GPIO.output(8, False)
12         time.sleep(0.5)
13
14 except KeyboardInterrupt:
15     GPIO.cleanup()
```

INPUT EXERCISE - BUTTONS





INPUT EXERCISE 1 - BUTTONS

```
1  import RPi.GPIO as GPIO
2
3  GPIO.setmode(GPIO.BCM)
4
5  GPIO.setup(23, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
6  GPIO.setup(24, GPIO.IN, pull_up_down = GPIO.PUD_UP)
7
8  try:
9
10     while True:
11         if(GPIO.input(23) ==1):
12             print("Button 1 pressed")
13
14         if(GPIO.input(24) == 0):
15             print("Button 2 pressed")
16
17 except KeyboardInterrupt:
18     GPIO.cleanup()
```

INPUT EXERCISE 2 - BUTTONS

```python
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(23, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
GPIO.setup(24, GPIO.IN, pull_up_down = GPIO.PUD_UP)

try:

    while True:
        GPIO.wait_for_edge(23, GPIO.RISING)
        print("Button 1 Pressed")

        GPIO.wait_for_edge(23, GPIO.FALLING)
        print("Button 1 Released")

        GPIO.wait_for_edge(24, GPIO.FALLING)
        print("Button 2 Pressed")

        GPIO.wait_for_edge(24, GPIO.RISING)
        print("Button 2 Released")

except KeyboardInterrupt:
    GPIO.cleanup()
```

NEXT WEEK

- Interfacing via Serial UART
- Interfacing via Internet of Things (IoT)