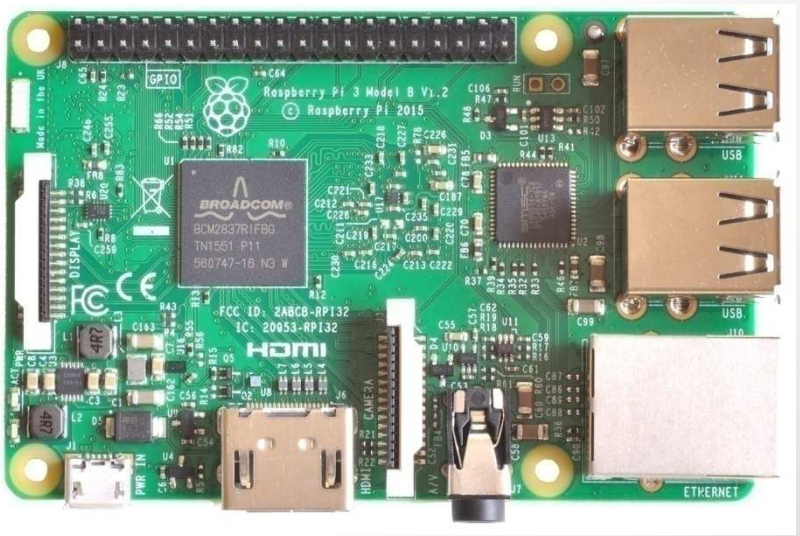
Introduction to Raspberry Pi

-Credit Card Sized Computer

What is a Raspberry Pi ?

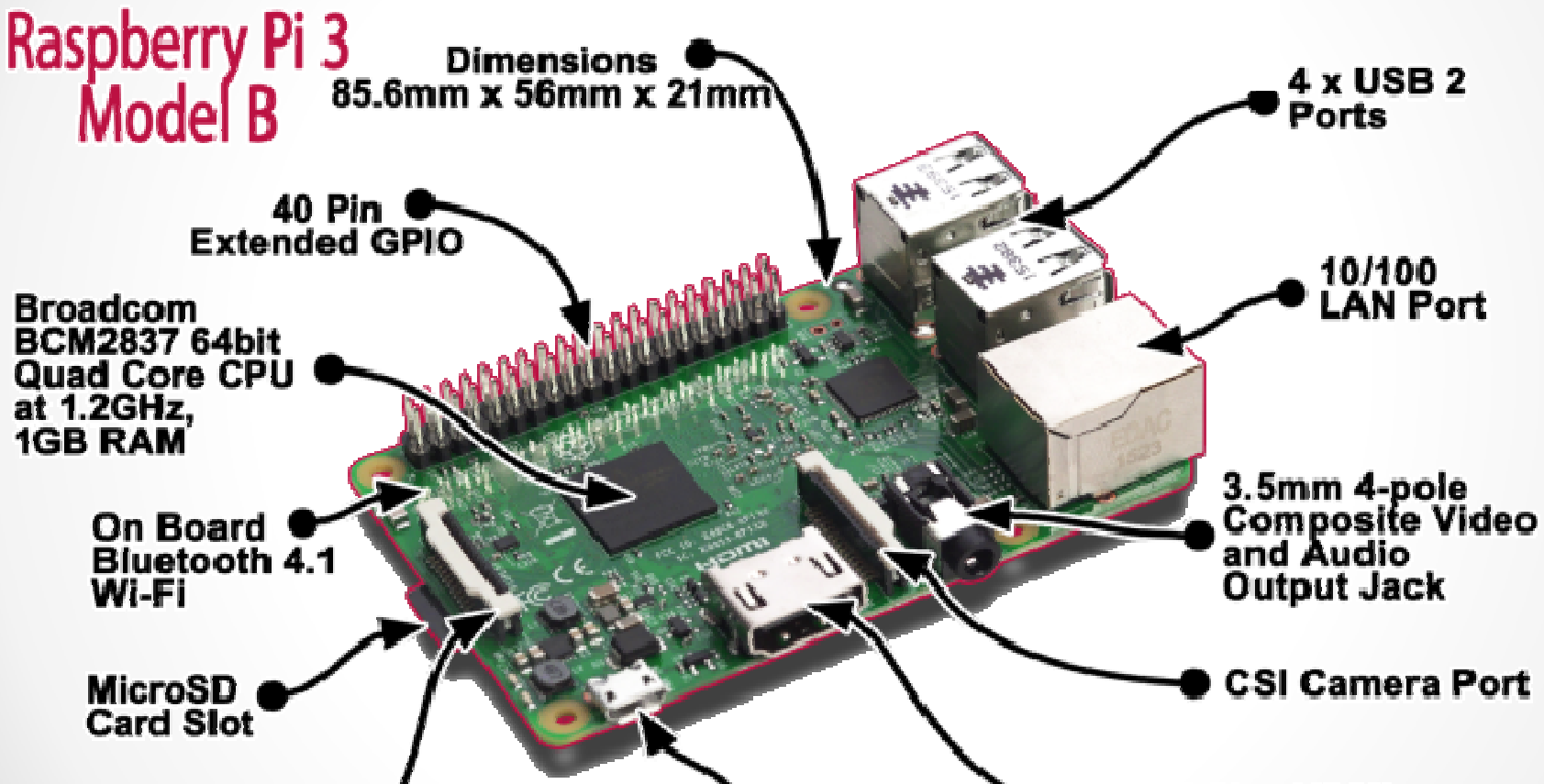
- A single board, very flexible, four watt credit card sized computer
- A mostly-open educational platform. (Some chip firmware not open)
- A standalone Linux, BSD, RISC OS, or Plan 9 system with a lot of I/O
- A powerful programming environment
- Capability:
  - Programming
  - Electronic Projects
  - Office
  - Play HD Videos



Pi History ?

- Around 2005 Eben Upton was Director of Studies in Computer science at Cambridge.
- Incoming students had relatively few programming and/or hardware skills vs “the old days”, creating vision of “Something like the BBC Computer, but running modern language like python”
- The name “Raspberry pi” is combination of “a fruit name” and play on “python”
- Developed in UK by Raspberry-pi foundation in 2009.
- Basic aim is to promote the study of basic computer science in schools and to develop interest among kids and adults

Raspberry Pi 3 Model B



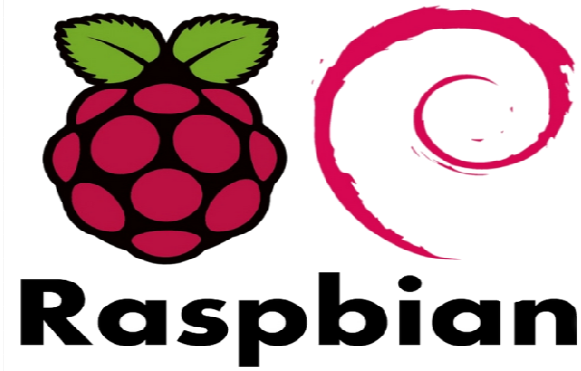
Pi GPIO's

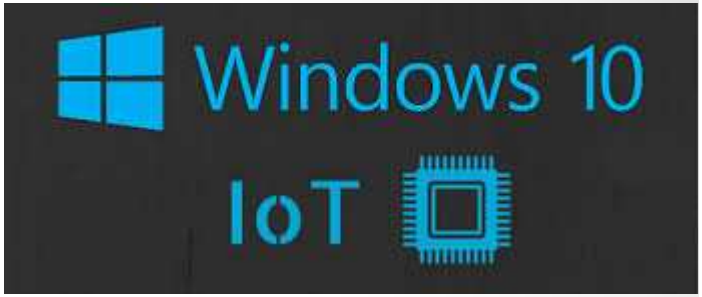
Alternate Function							Alternate Function
	3.3V PWR	1		2	5V PWR		
I2C1 SDA	GPIO 2	3		4	5V PWR		
I2C1 SCL	GPIO 3	5		6	GND		
	GPIO 4	7		8	UART0 TX		
	GND	9		10	UART0 RX		
	GPIO 17	11		12	GPIO 18		
	GPIO 27	13		14	GND		
	GPIO 22	15		16	GPIO 23		
	3.3V PWR	17		18	GPIO 24		
SPI0 MOSI	GPIO 10	19		20	GND		
SPI0 MISO	GPIO 9	21		22	GPIO 25		
SPI0 SCLK	GPIO 11	23		24	GPIO 8		SPI0 CS0
	GND	25		26	GPIO 7		SPI0 CS1
	Reserved	27		28	Reserved		
	GPIO 5	29		30	GND		
	GPIO 6	31		32	GPIO 12		
	GPIO 13	33		34	GND		
SPI1 MISO	GPIO 19	35		36	GPIO 16		SPI1 CS0
	GPIO 26	37		38	GPIO 20		SPI1 MOSI
	GND	39		40	GPIO 21		SPI1 SCLK

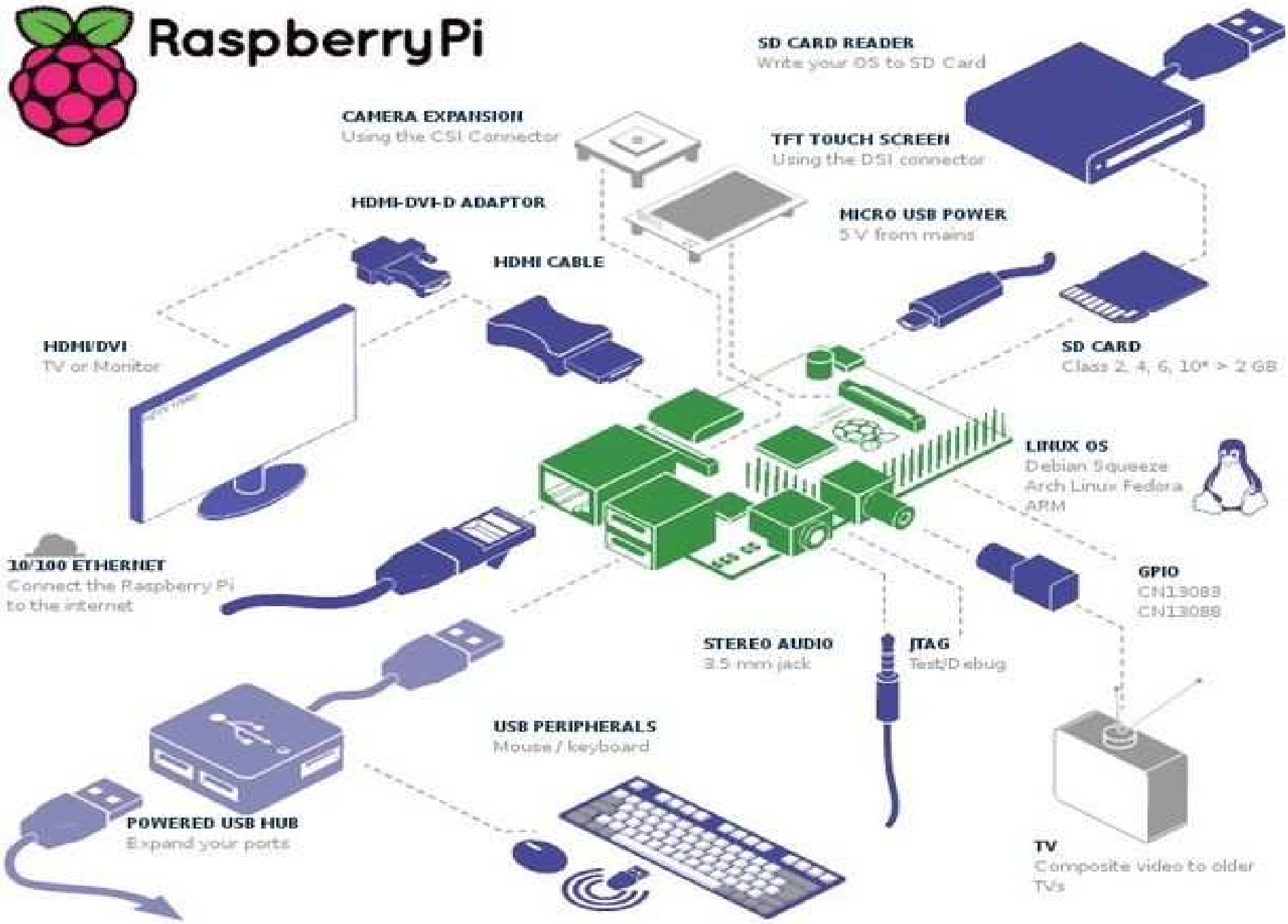
Programming Languages

- The Raspberry Pi Foundation recommends Python
- Any language which will compile for ARMv6 can be used
- Installed by default on the Raspberry Pi:
  - C
  - C++
  - Python
  - Java
  - Scratch
  - Ruby

OS for Raspberry Pi



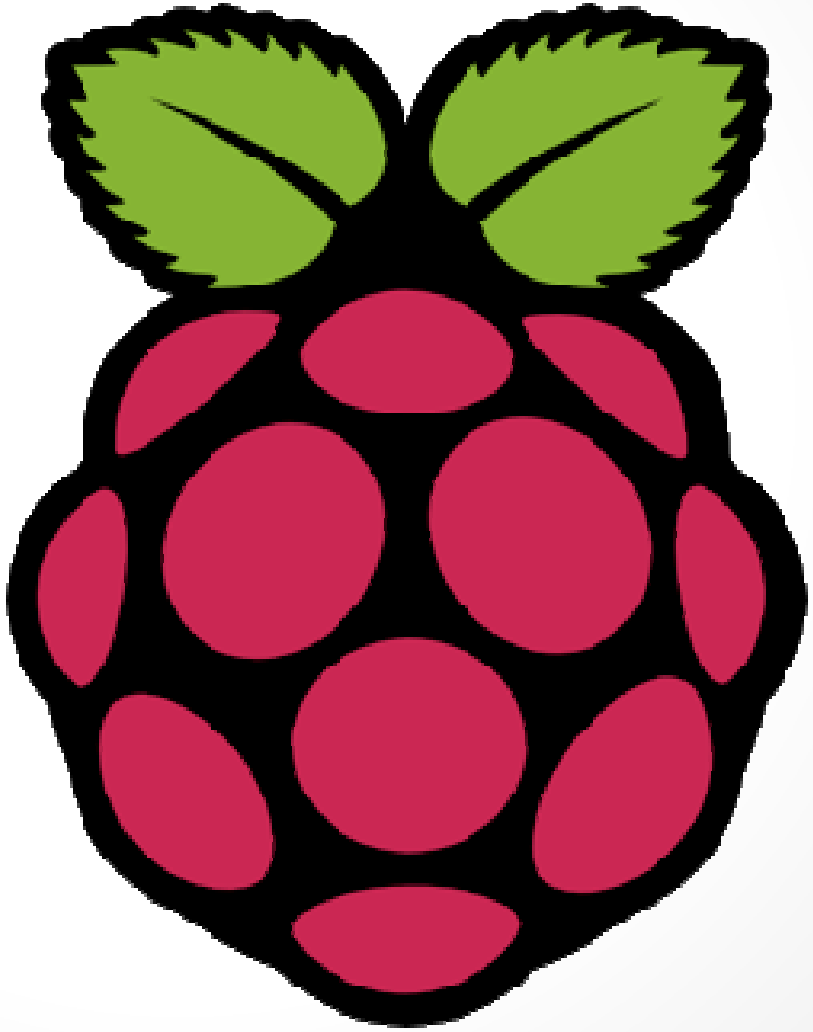



RaspberryPi
SD CARD READER
Write your OS to SD Card
CAMERA EXPANSION
Using the CSI Connector
TFT TOUCH SCREEN
Using the DSI connector
HDMI-DVI-D ADAPTOR
MICRO USB POWER
5 V from mains
HDMI CABLE
HDMI/DVI
TV or Monitor
SD CARD
Class 2, 4, 6, 10* > 2 GB
LINUX OS
Debian Squeeze
Arch Linux Fedora
ARM
10/100 ETHERNET
Connect the Raspberry Pi
to the internet
GPIO
CN13083
CN13088
STEREO AUDIO
3.5 mm jack
JTAG
Test/Debug
USB PERIPHERALS
Mouse / keyboard
POWERED USB HUB
Expand your ports
TV
Composite video to older
TVs

Comparison

	Raspberry Pi 3 Model B	Raspberry Pi Zero	Raspberry Pi 2 Model B	Raspberry Pi Model B+
Introduction Date	2/29/2016	11/25/2015	2/2/2015	7/14/2014
SoC	BCM2837	BCM2835	BCM2836	BCM2835
CPU	Quad Cortex A53 @ 1.2GHz	ARM11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM11 @ 700MHz
Instruction set	ARMv8-A	ARMv6	ARMv7-A	ARMv6
GPU	400MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV	250MHz VideoCore IV
RAM	1GB SDRAM	512 MB SDRAM	1GB SDRAM	512MB SDRAM
Storage	micro-SD	micro-SD	micro-SD	micro-SD
Ethernet	10/100	none	10/100	10/100
Wireless	802.11n / Bluetooth 4.0	none	none	none
Video Output	HDMI / Composite	HDMI / Composite	HDMI / Composite	HDMI / Composite
Audio Output	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40
Price	$35	$5	$35	$35

SETUP



Raspberry Pi
Quick start

1 Insert SD card

See page 3 for how to prepare the SD card

2a Connect display

Plug in your digital TV or monitor

2b Connect display

If *not* using HDMI, plug in your analogue TV or display

3 Connect input

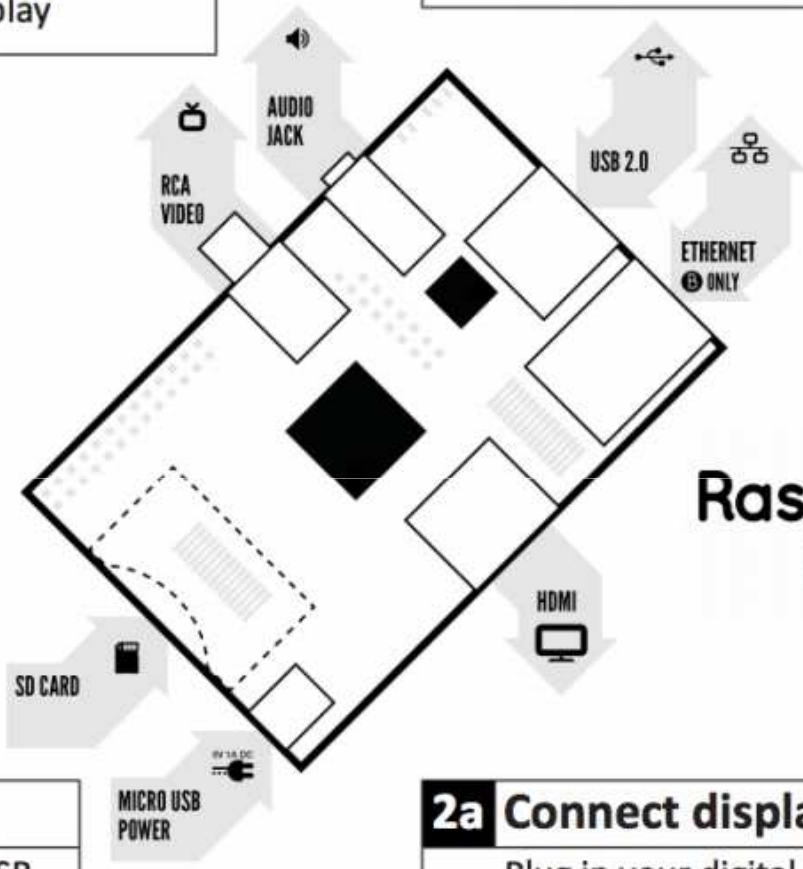
Plug in a USB keyboard and mouse

4 Connect network

Connect to your wired network [optional]

5 Power up

Plug in the micro USB power supply



Basic GNU/Linux commands

- $ls – List Files
- $cd – Change Directory
- $mkdir – Create folder
- $rmdir – Delete folder
- $clear – Clear terminal
- $nano – Text editor
- $dmesg – Show kernel messages

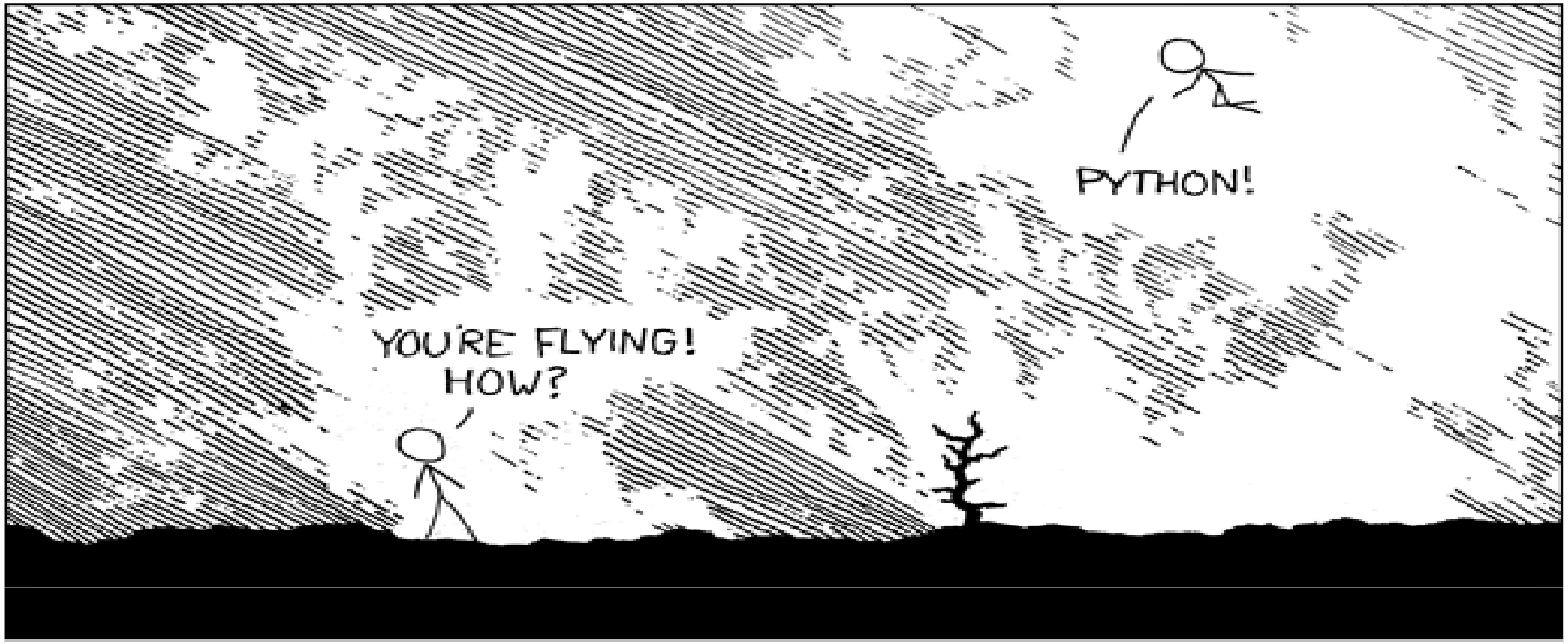
Basic GNU/Linux commands

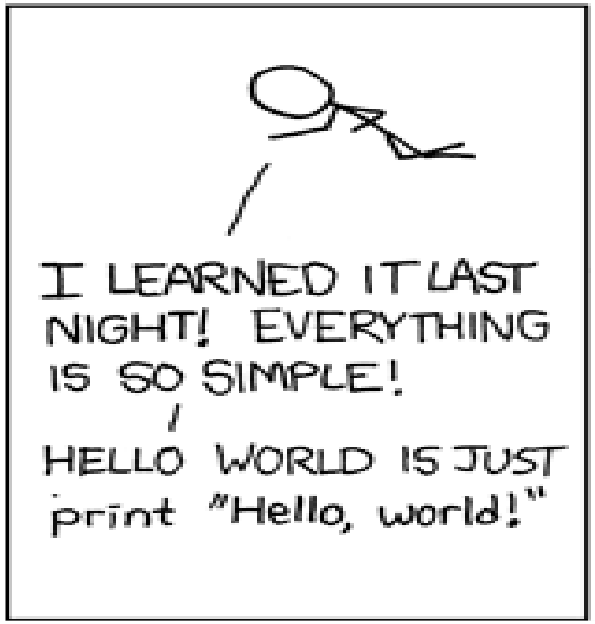
- $lsusb – List connected usb devices
- $cp – Copy Files
- $ssh – Secure shell
- $scp – Copy files from one pc to other using ssh
- $vncviewer – Connect to vnc server
- $ sudo – Run in root privilage
- $ping – Pinging to ip address
- $nmap – Searching tool of IP and Ports

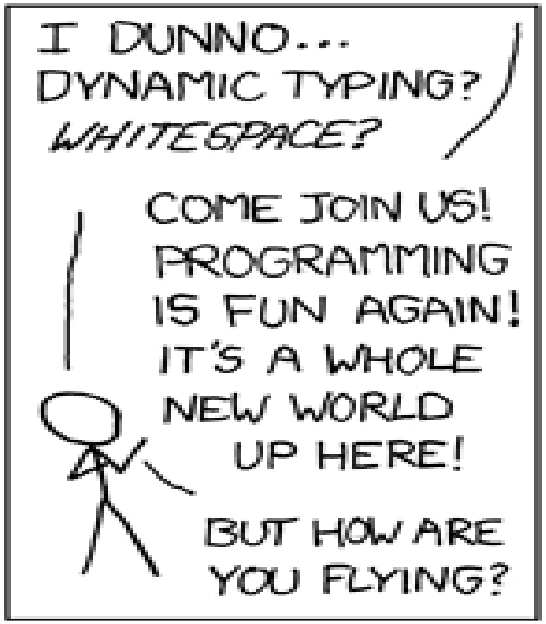
Introduction to Python

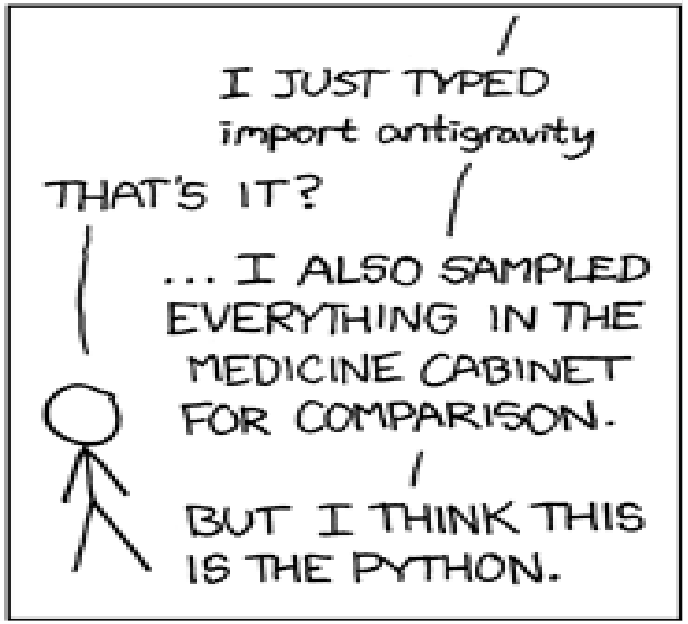
"Life's better without braces and Semicolons"

--Bruce Eckel

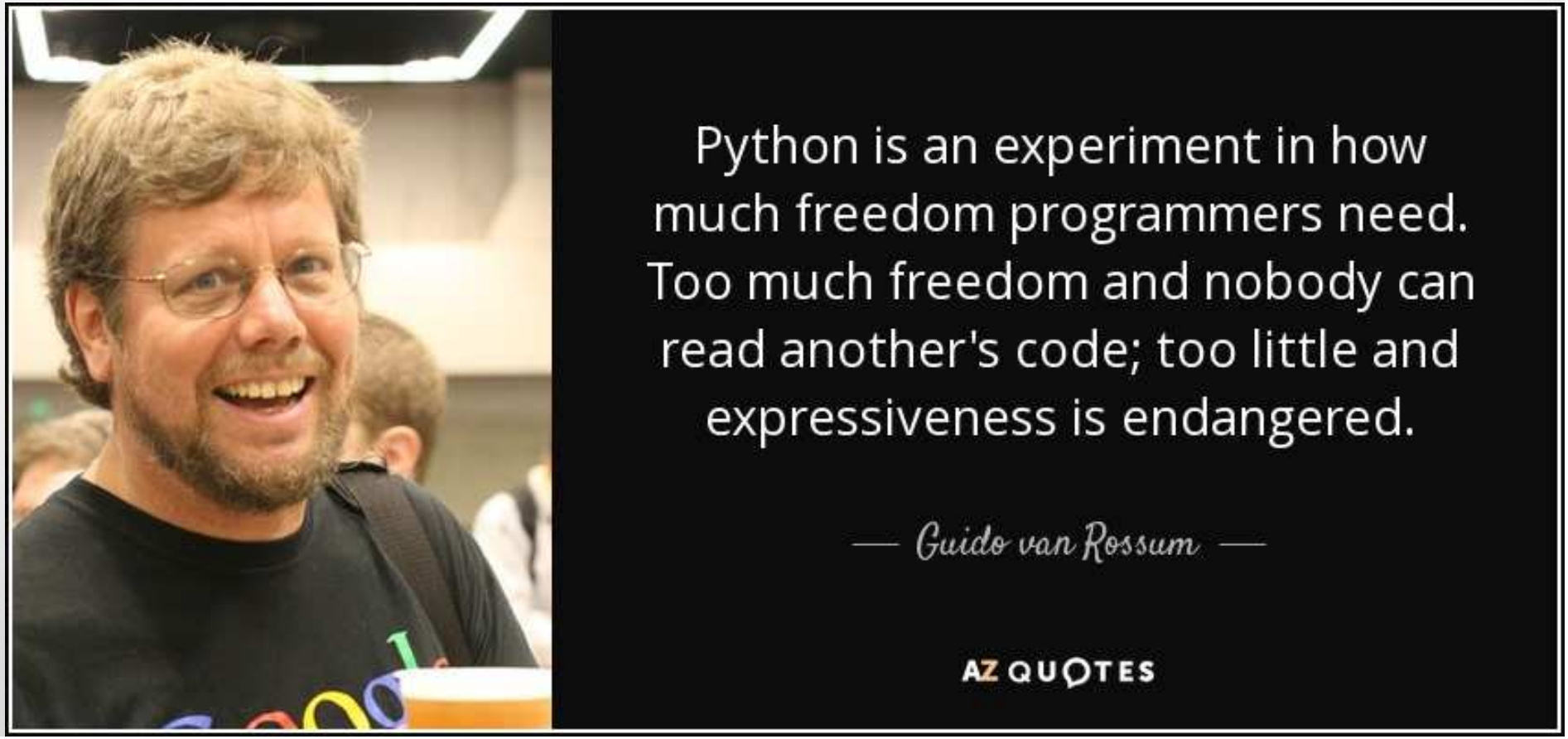

PYTHON!
YOU'RE FLYING! HOW?


I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"


I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?


I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Python was developed by Guido van Rossum in the 90's at the National Research Institute for Mathematics and Computer Science in the Netherlands.

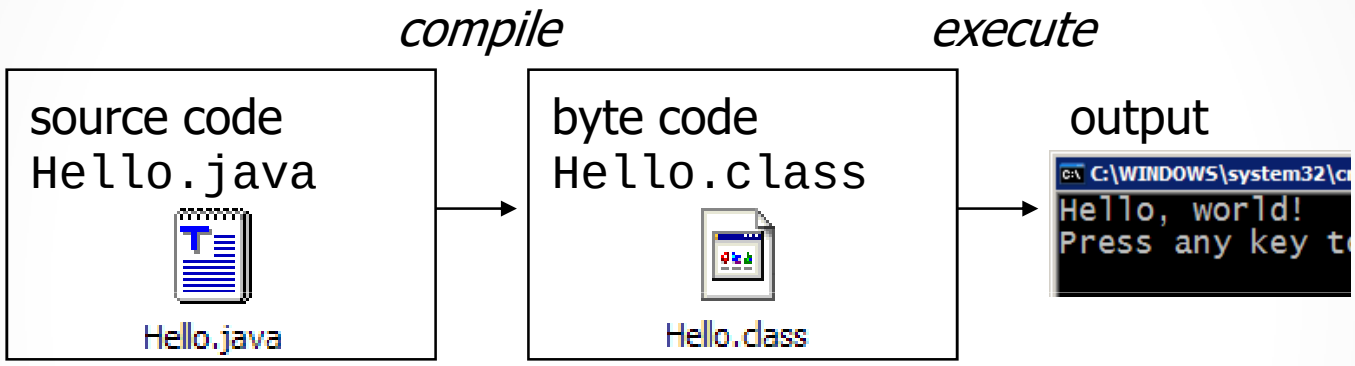


Introduction to Python

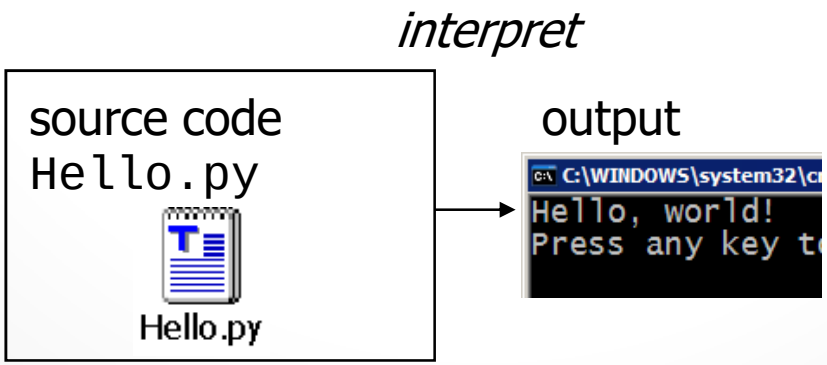
- Python is a high-level programming language
- Open source and community driven
- “Batteries Included”
  - a standard distribution includes many modules
- Dynamic typed
- Source can be compiled or run just-in-time
- Similar to perl, tcl, ruby

Compiling and interpreting

- Many languages require you to *compile* (translate) your program into a form that the machine understands.



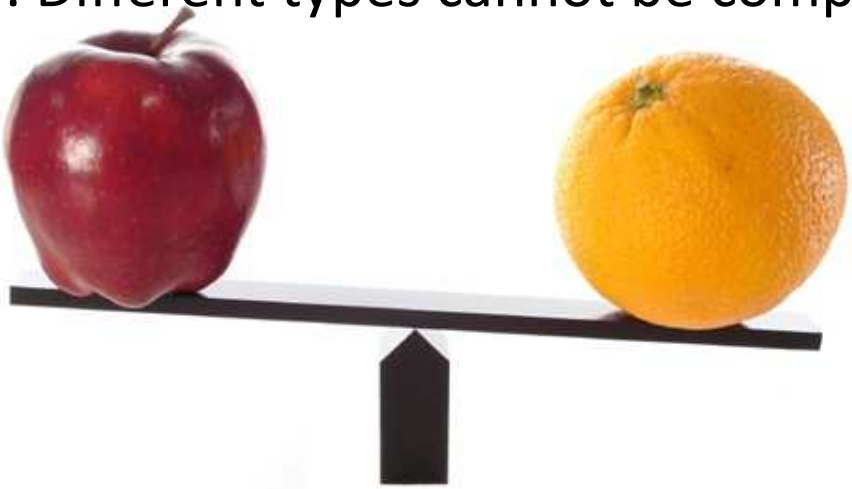
- Python is instead directly *interpreted* into machine instructions.



1. Python is a calculator

2. A variable is a container

3. Different types cannot be compared



4. A program is a recipe

The Python Interpreter

- Python is an interpreted language
- The interpreter provides an interactive environment to play with the language
- Results of expressions are printed on the screen

```
>>> 3 + 7
10
>>> 3 < 15
True
>>> 'print me'
'print me'
>>> print 'print me'
print me
>>>
```

The print Statement

- Elements separated by commas print with a space between them
- A comma at the end of the statement (print ‘hello’,) will not print a newline character

```
>>> print 'hello'
hello
>>> print 'hello', 'there'
hello there
```

Python has five standard data types:

- Numbers
- String
- List
- Tuple
- Dictionary

Python Number data types:

```
1   a = 5
2   print(a, "is of type", type(a))
3
4   a = 2.0
5   print(a, "is of type", type(a))
6
7   a = 1+2j
8   print(a, "is complex number?", isinstance(1+2j,complex))
```

```
5 is of type <class 'int'>
2.0 is of type <class 'float'>
(1+2j) is complex number? True
```

Python Lists:

•Lists are the most versatile of Python's compound data types.

•A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C

Python Lists

- Lists are the most versatile of Python's compound data types.
- A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C

```
list=[ 'abcd', 786 , 2.23, 'john', 70.2 ]
tinylist=[123,'john']
print list            # Prints complete list
print list[0]         # Prints first element of the list
print list[1:3]       # Prints elements starting from 2nd till 3rd
print list[2:]        # Prints elements starting from 3rd element
print tinylist * 2    # Prints list two times
print list + tinylist # Prints concatenated lists
```

Python Tuples-()

- A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas.
- tuples are enclosed in Parentheses (( )) and cannot be updated.
- Tuples can be thought of as read-only

Python Tuple Example

```
#!/usr/bin/python
tuple = ( 'abcd', 786 , 2.23, 'john', 70.2 )
tinytuple = (123, 'john')
print tuple # Prints complete list
print tuple[0] # Prints first element of the list
print tuple[1:3] # Prints elements starting from 2nd till 3rd
print tuple[2:] # Prints elements starting from 3rd element
print tinytuple * 2 # Prints list two times
print tuple + tinytuple # Prints concatenated lists
```

Python Dictionaries

- Python's dictionaries are kind of hash table type.
- Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

Example Dictionaries

```
#!/usr/bin/python

dict = {}

dict['one'] = "This is one"

dict[2] = "This is two"

tinydict = {'name': 'john','code':6734, 'dept': 'sales'}

print dict['one'] # Prints value for 'one' key

print dict[2] # Prints value for 2 key

print tinydict # Prints complete dictionary

print tinydict.keys() # Prints all the keys

print tinydict.values() # Prints all the values
```

Python Operators

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operators

Ticks

- Time intervals are floating-point numbers in units of seconds. Particular instants in
- Time are expressed in seconds since 12:00am, January 1, 1970(epoch).

Python Time Module

- ```
  #!/usr/bin/python
  import time; # This is required to include time module.
  ticks = time.time()
  print "Number of ticks since 12:00am, January 1, 1970:", ticks
  ```

- Output:

  Number of ticks since 12:00am, January 1, 1970: 1499329502.34
  ```

Printing localtime

```
#!/usr/bin/python
import time;
localtime = time.localtime(time.time())
print "Local current time :", localtime
```

Output:

Local current time : time.struct_time(tm_year=2017, tm_mon=7, tm_mday=6, tm_hour=14, tm_min=0, tm_sec=32, tm_wday=3, tm_yday=187, tm_isdst=0)

Calendar Module

- ```
  #!/usr/bin/python
  import calendar
  cal = calendar.month(2008, 1)
  print "Here is the calendar:"
  print cal;
  ```

Output:

```
Here is the calendar:
    January 2008
Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```
```

Indentation

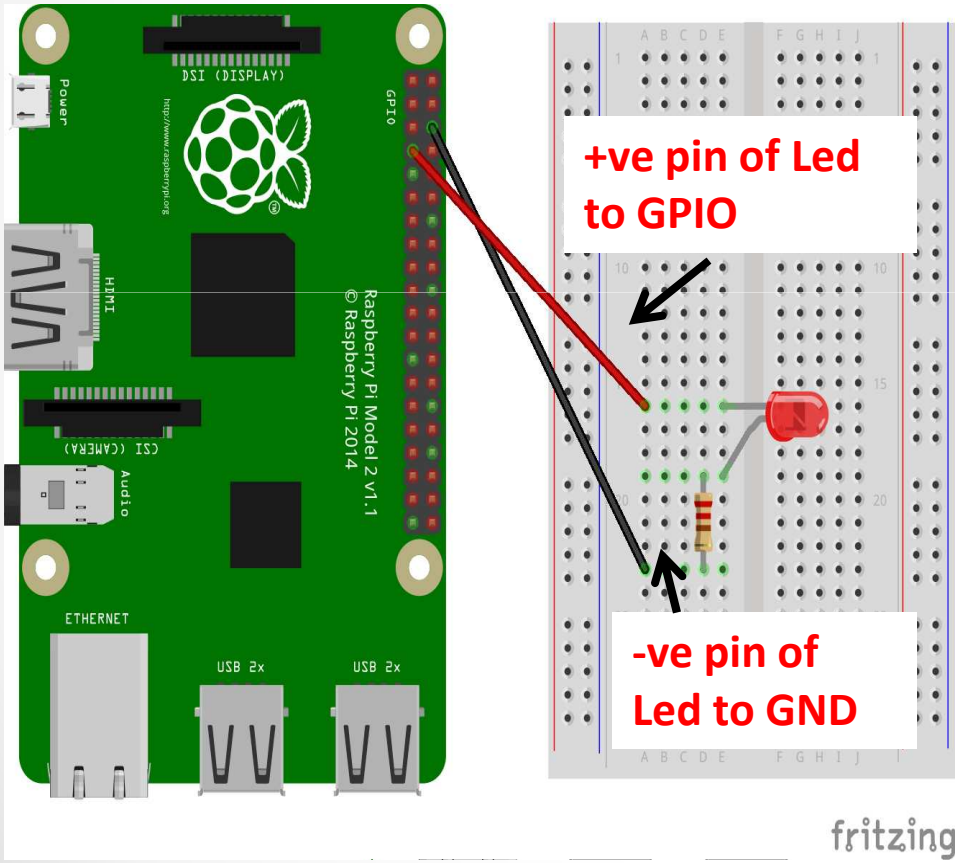
- Indentation is **required** and must be **consistent**
- Standard indentation is 4 spaces **or** one tab
- IDLE does this pretty much automatically for you
- Example:

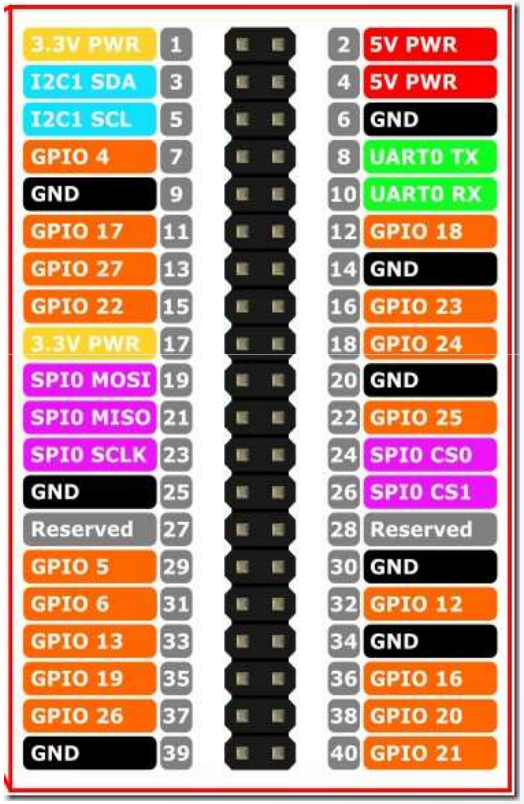
```
if 2 + 2 != 4:
        print "Oh, no!"
        print "Arithmethic doesn't work!"
        print "Time to buy a new computer."
```

"Life's better without braces and Semicolons"

--Bruce Eckel

Blinking of LED





Importing GPIO pins of Raspberry Pi

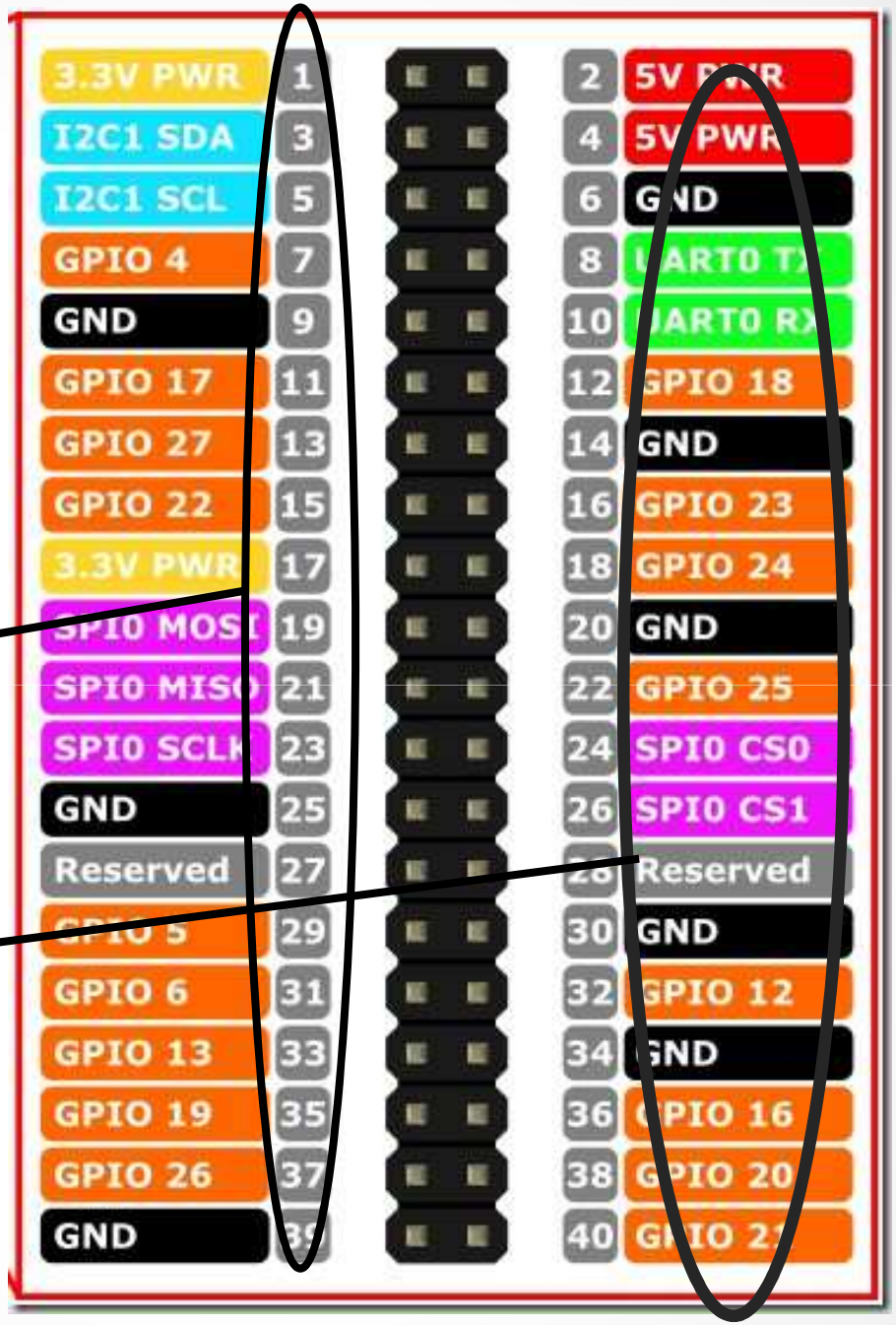
```
Import RPi.GPIO as GPIO          ## Import GPIO library

GPIO.setmode(GPIO.BOARD)         ## Use board pin numbering

GPIO.setup(7, GPIO.OUT)          ## Setup GPIO Pin 11 to OUT

Or

GPIO.setmode(GPIO.BCM)           ## Broadcom SOC number

GPIO.setup(4, GPIO.OUT)          ## Setup GPIO Pin 11 to OUT
```

Pi GPIO's

Alternate Function							Alternate Function
	3.3V PWR	1		2	5V PWR		
I2C1 SDA	GPIO 2	3		4	5V PWR		
I2C1 SCL	GPIO 3	5		6	GND		
	GPIO 4	7		8	UART0 TX		
	GND	9		10	UART0 RX		
	GPIO 17	11		12	GPIO 18		
	GPIO 27	13		14	GND		
	GPIO 22	15		16	GPIO 23		
	3.3V PWR	17		18	GPIO 24		
SPI0 MOSI	GPIO 10	19		20	GND		
SPI0 MISO	GPIO 9	21		22	GPIO 25		
SPI0 SCLK	GPIO 11	23		24	GPIO 8		SPI0 CS0
	GND	25		26	GPIO 7		SPI0 CS1
	Reserved	27		28	Reserved		
	GPIO 5	29		30	GND		
	GPIO 6	31		32	GPIO 12		
	GPIO 13	33		34	GND		
SPI1 MISO	GPIO 19	35		36	GPIO 16		SPI1 CS0
	GPIO 26	37		38	GPIO 20		SPI1 MOSI
	GND	39		40	GPIO 21		SPI1 SCLK

For GPIO.setmode(GPIO.BOARD)
Ex: GPIO.setup(7, GPIO.OUT)

For GPIO.setmode(GPIO.BCM)
Ex: GPIO.setup(4, GPIO.OUT)

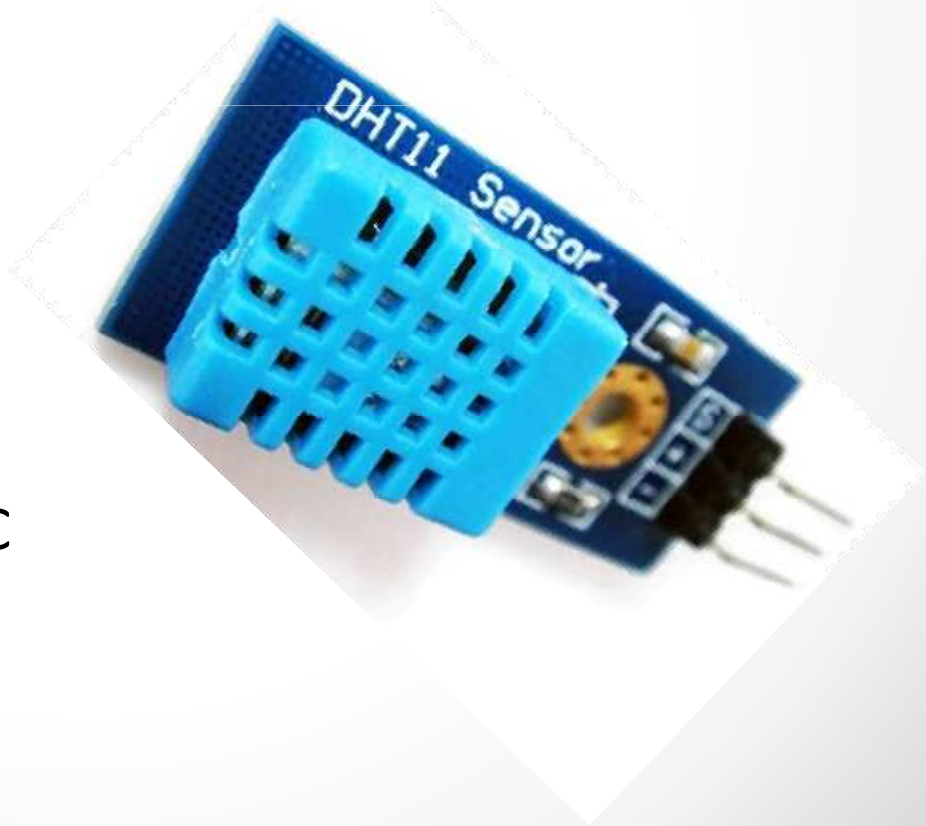


LED Blink Python Programme

```
import time
importRPi.GPIO as GPIO                    ## Import GPIO library
GPIO.setmode(GPIO.BOARD)                  ## Use board pin numbering
GPIO.setup(7, GPIO.OUT)                   ## Setup GPIO Pin 11 to OUT

while True:
        GPIO.output(7,True)               ## Turn on Led
        time.sleep(1)                     ## Wait for one second
        GPIO.output(7,False)              ## Turn off Led
        time.sleep(1)                     ## Wait for one second
```

DHT11 Temperature and Humidity Sensor

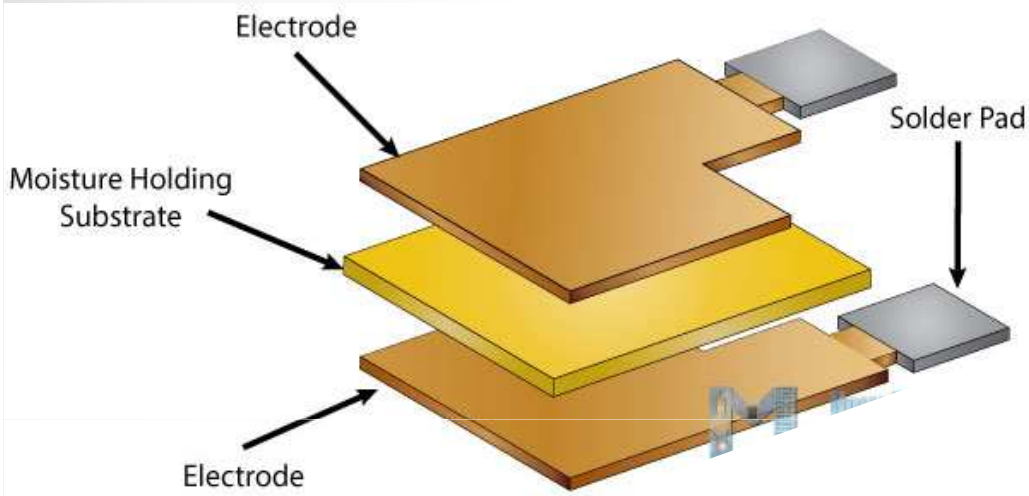
- **DHT11 is a basic, ultra low-cost digital temperature and humidity sensor**
- **Capacitive humidity sensor and a thermistor to measure the surrounding air**
- **Detects water vapor by measuring the electrical resistance between two electrodes**
- **Humidity sensing component is a moisture holding substrate with electrodes applied to the surface**

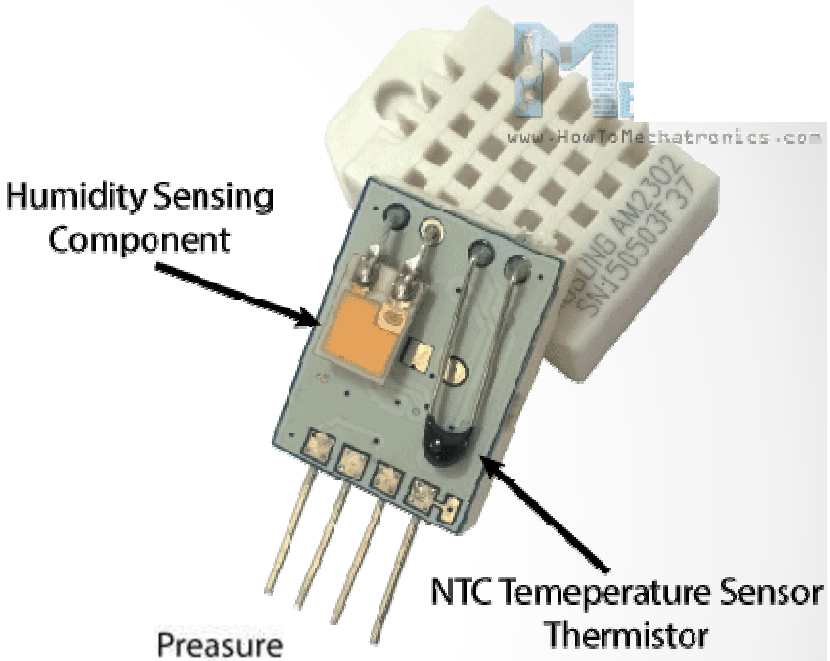


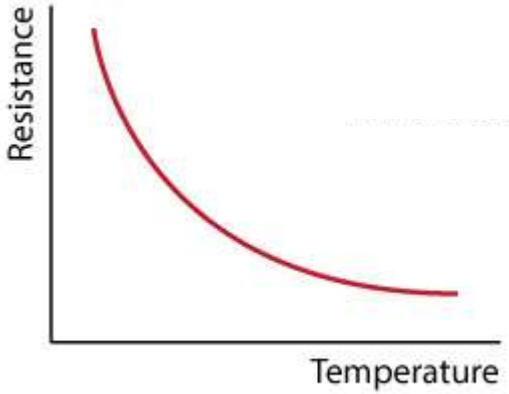
Technical Specification:

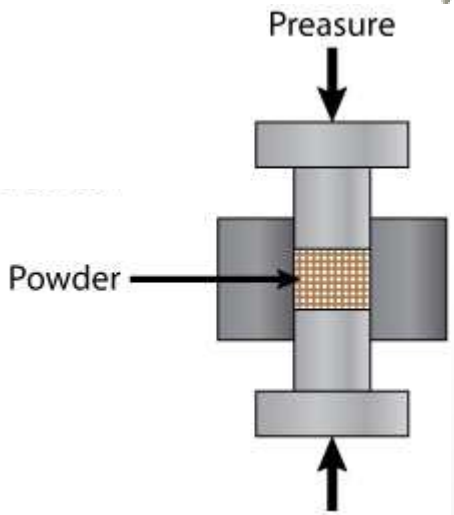
- Humidity Range: 20-90% RH
- Humidity Accuracy: ±5% RH
- Temperature Range: 0-50 °C
- Temperature Accuracy: ±2% °C
- Operating Voltage: 3V to 5.5V

DHT11 Temperature and Humidity Sensor

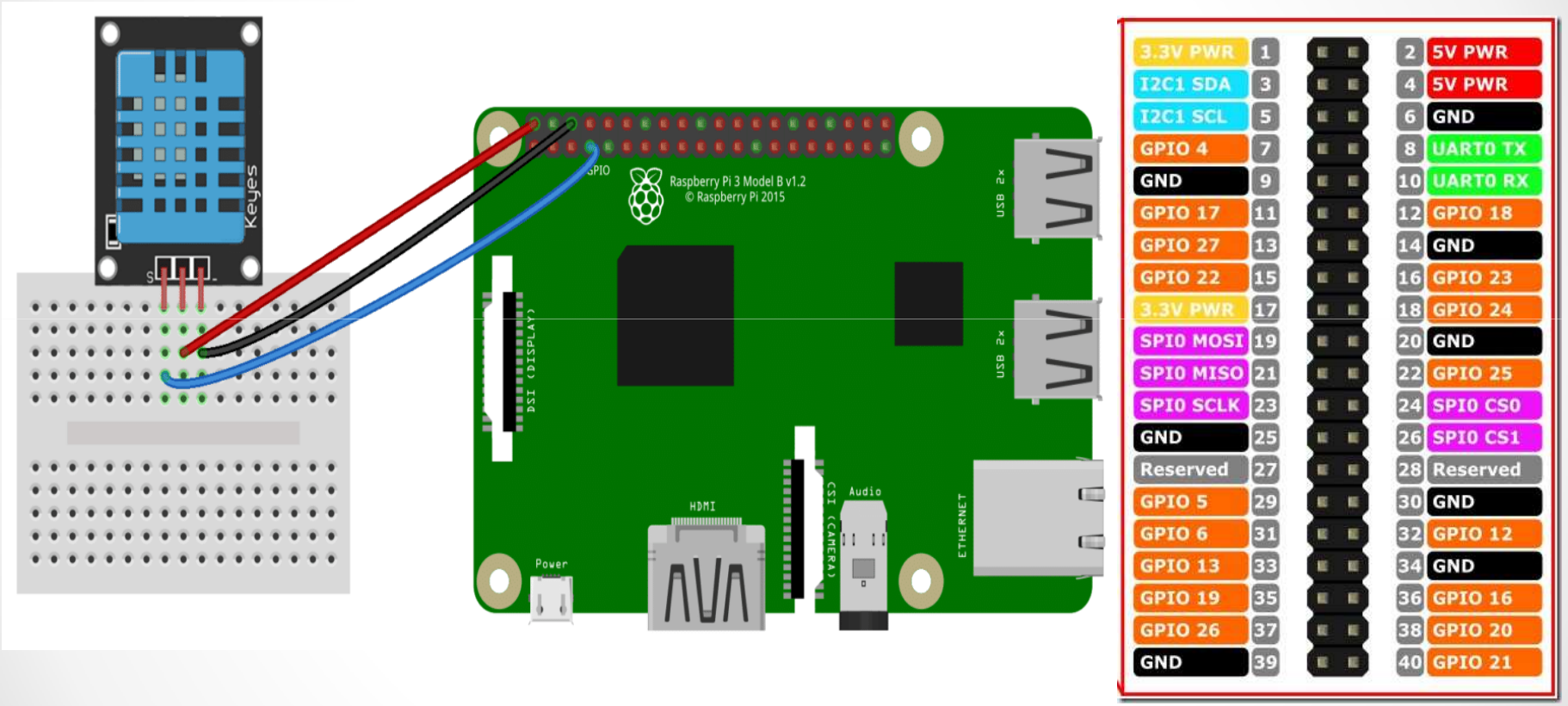








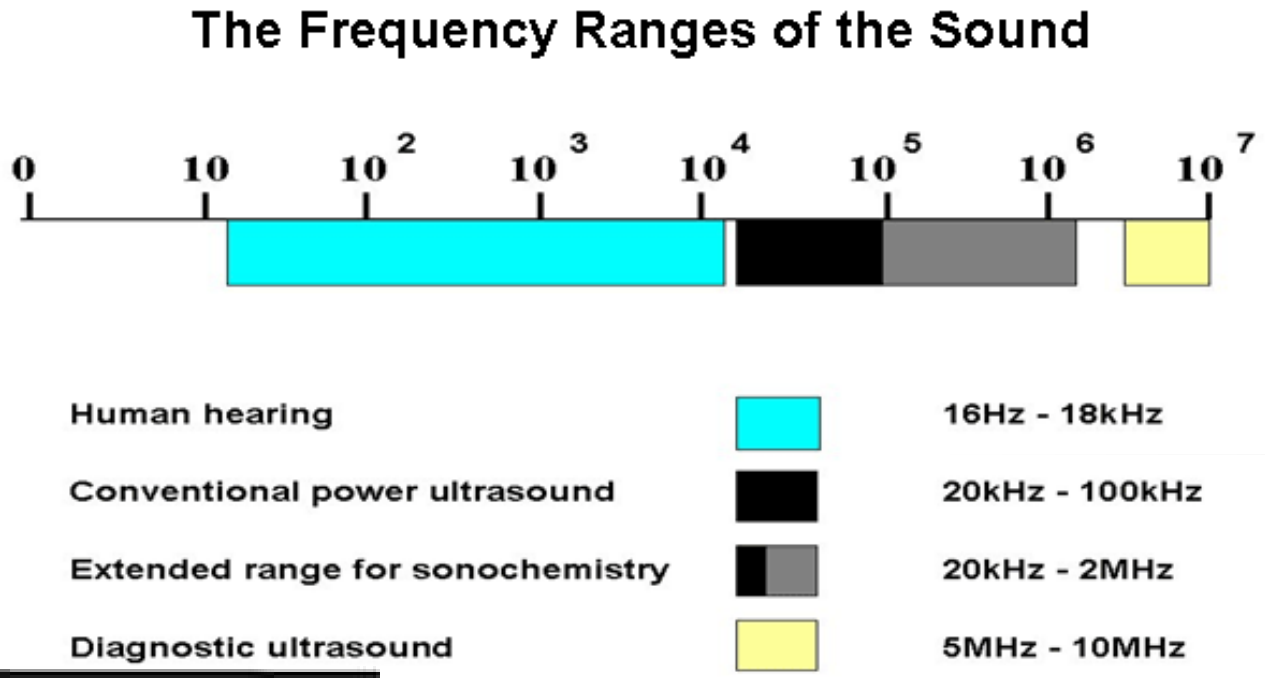
DHT11 Temperature and Humidity Sensor

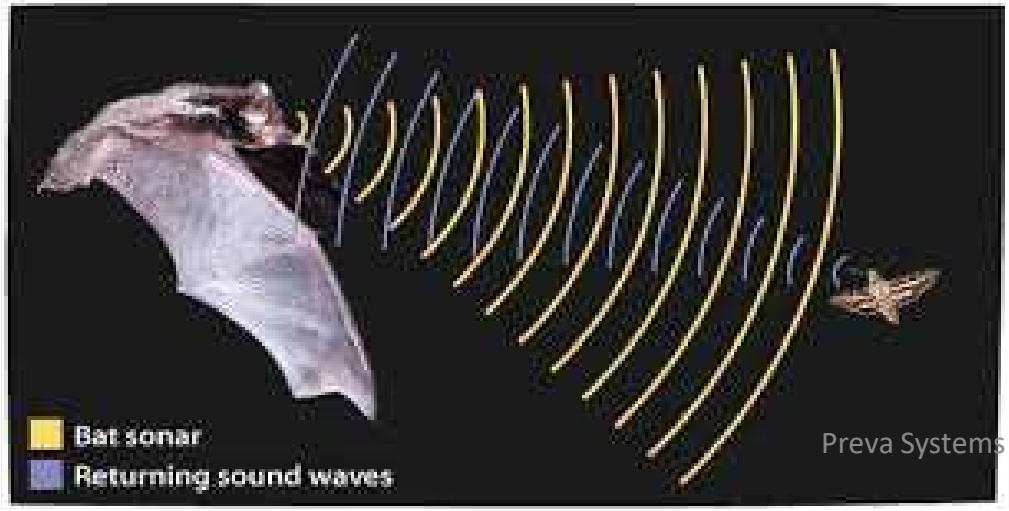


+ve pin of DHT11 to 5v
-ve pin of DHT11 to GND
OUT pin of DHT11 to GPIO4

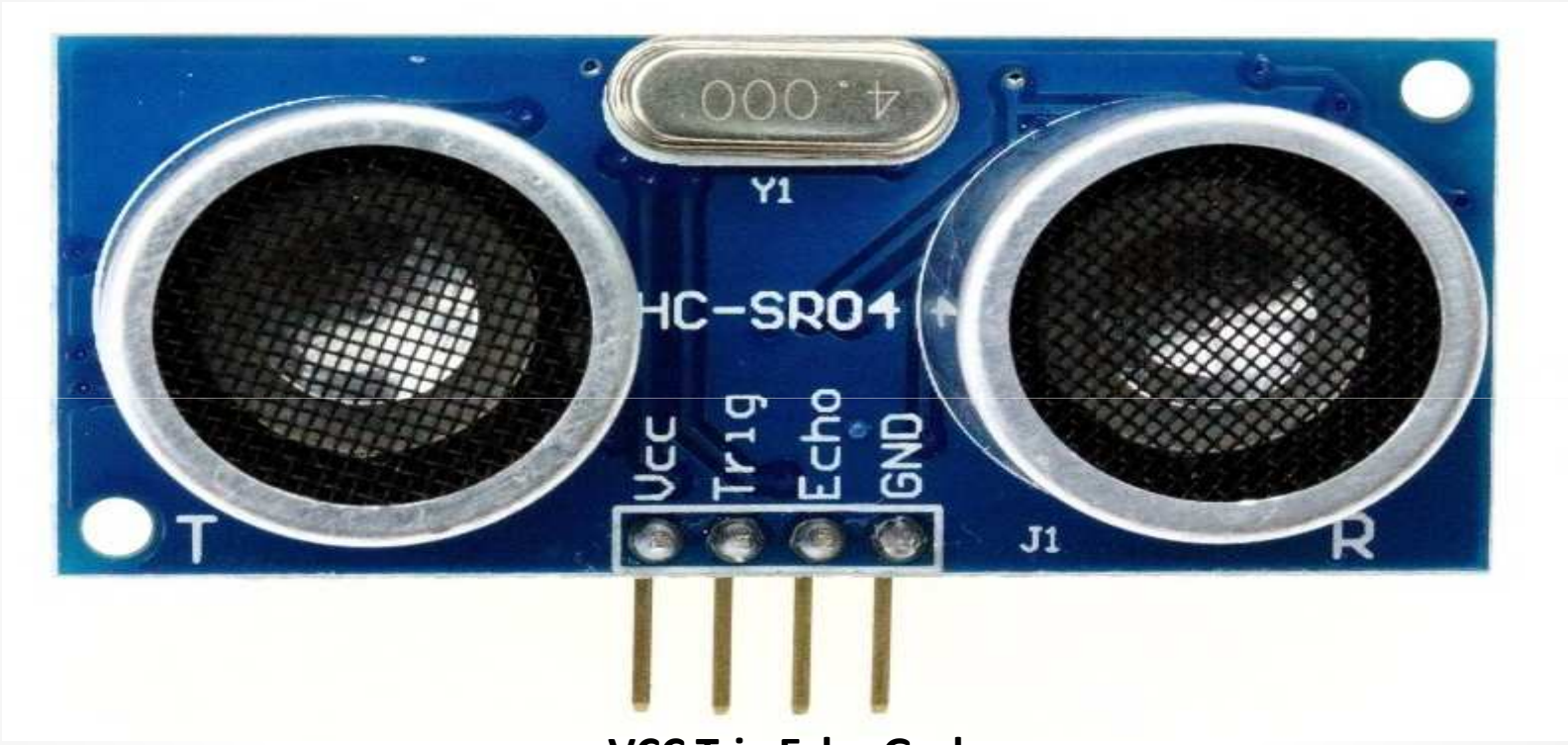
DHT11 PYTHON PROGRAMME

Ultrasonic Sensor



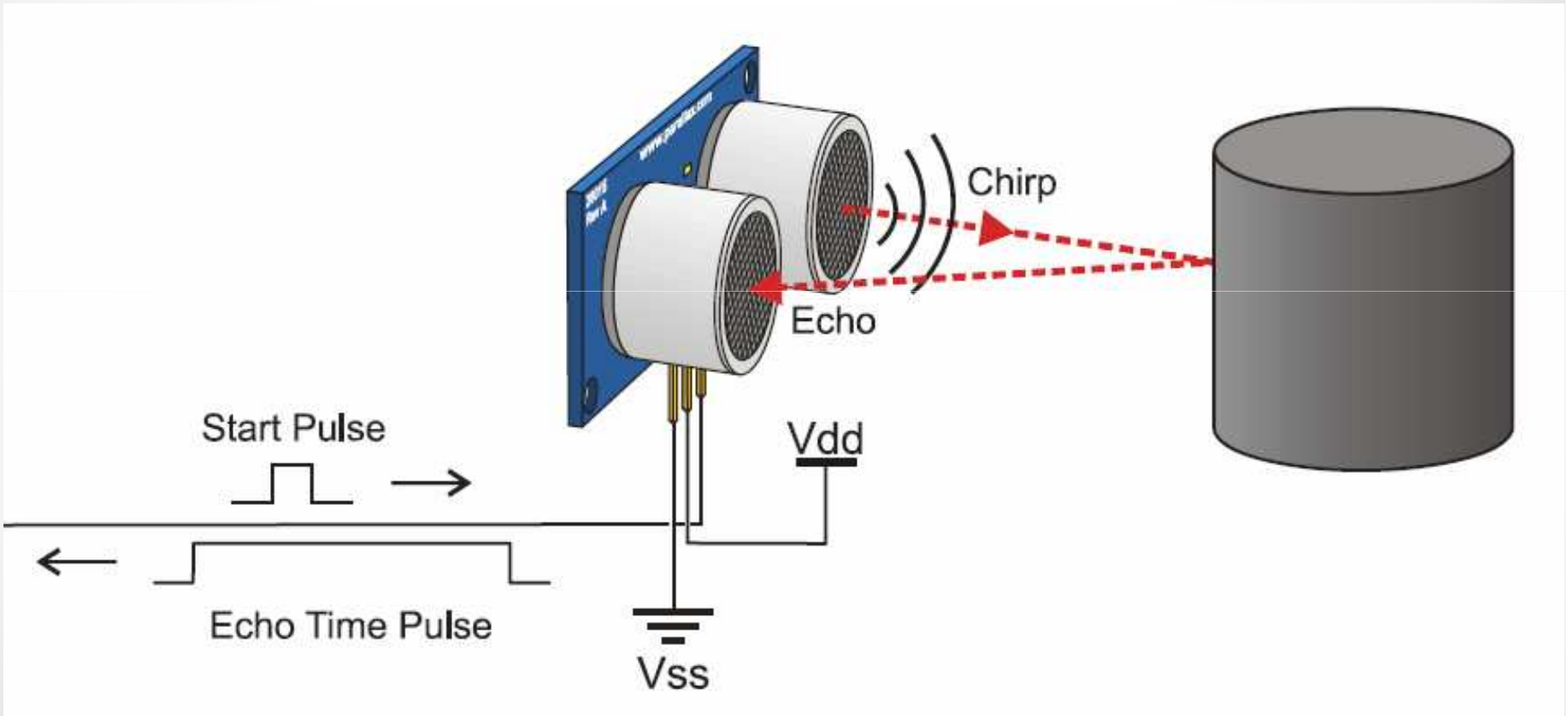


Pin Outs

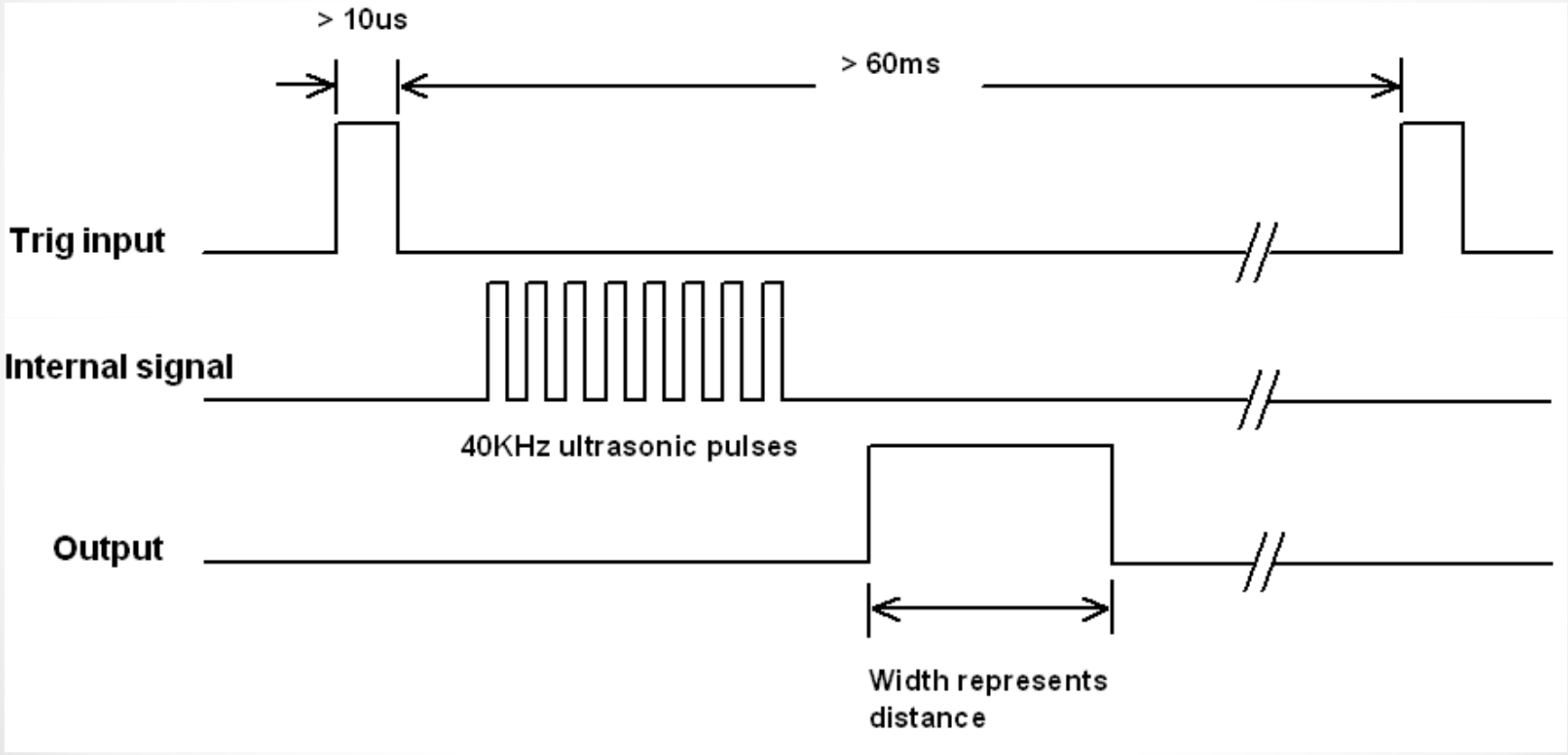


VCC Trig Echo Gnd

Working

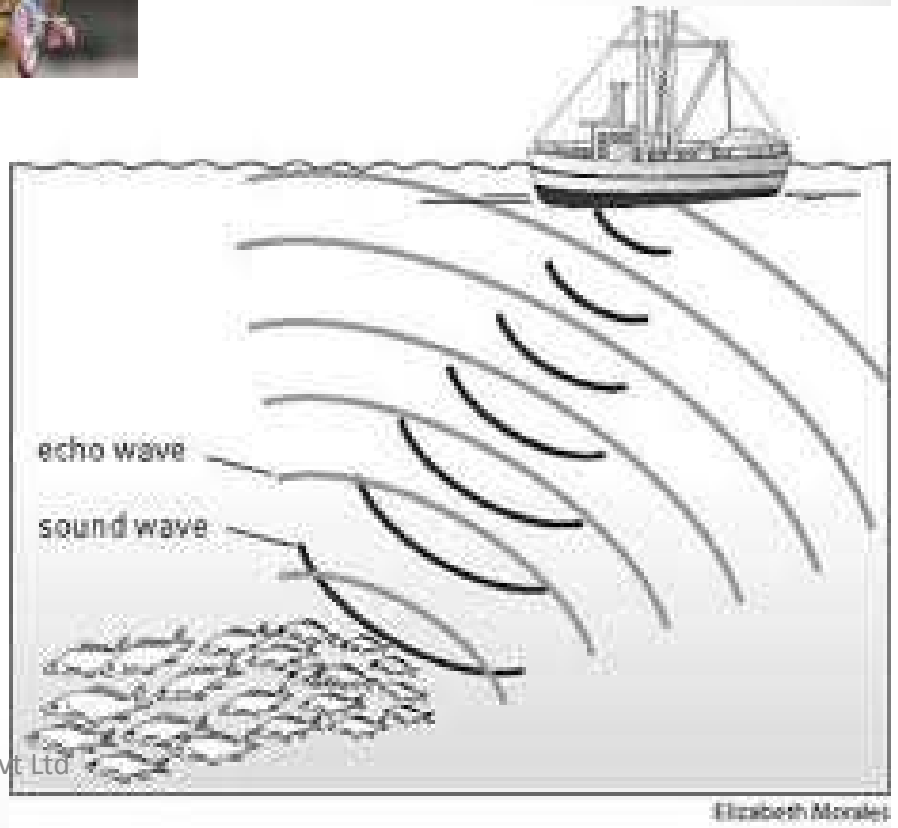


Timing Diagram

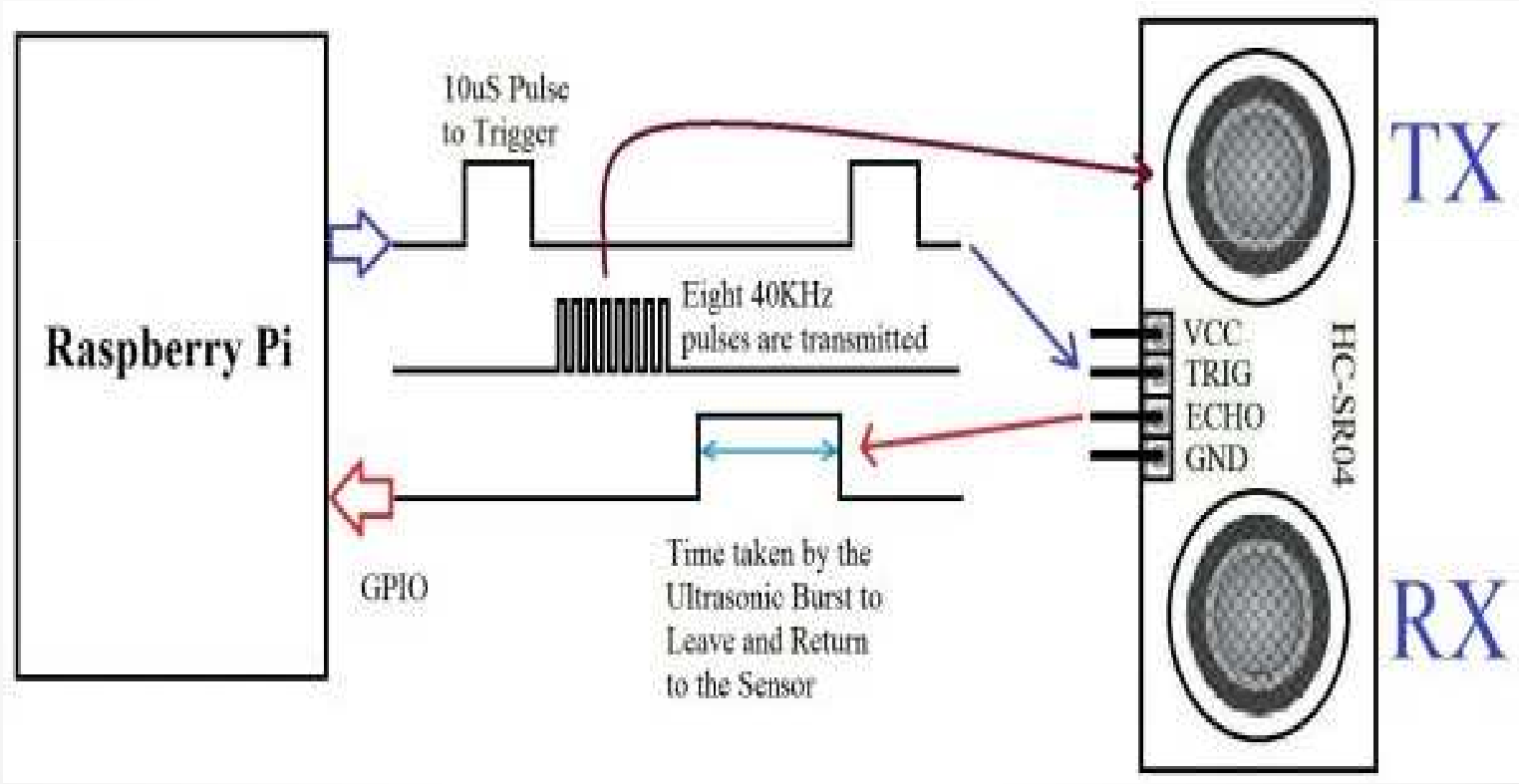


Applications

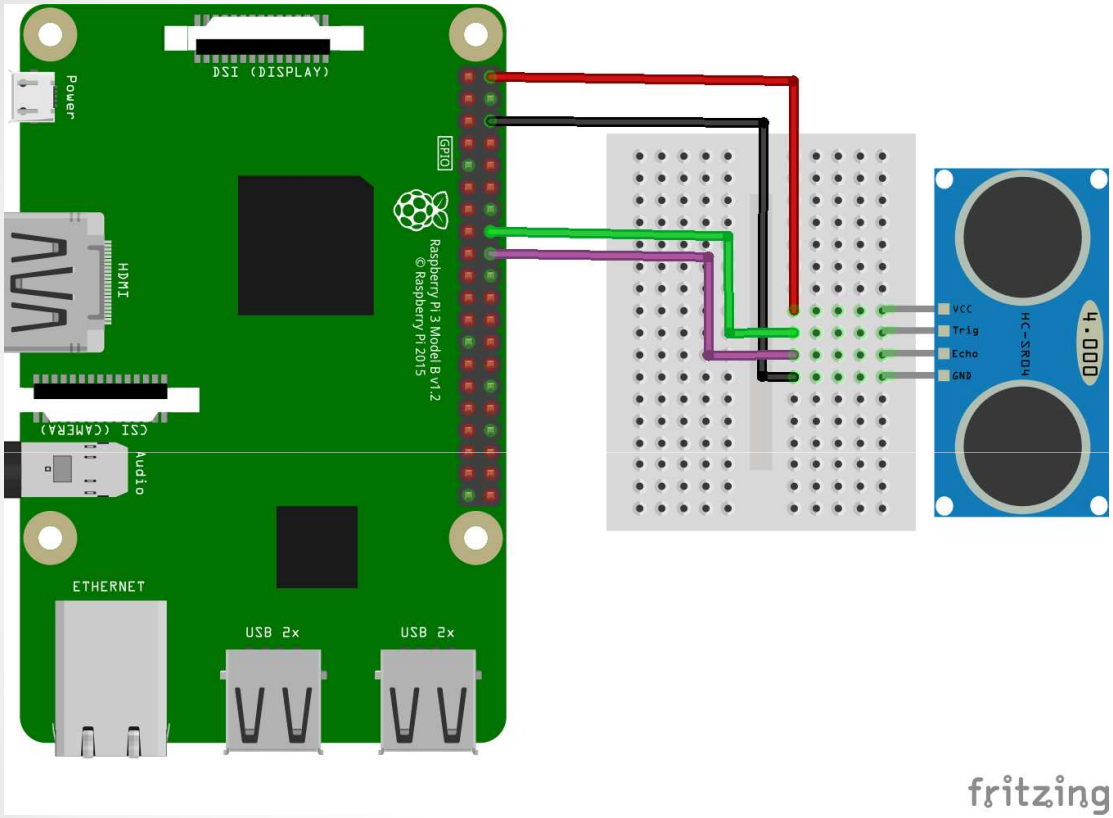
- Park assistance
- Distance measuring device
- SONAR
- Fishing



Working


Raspberry Pi
10uS Pulse
to Trigger
Eight 40KHz
pulses are transmitted
GPIO
Time taken by the
Ultrasonic Burst to
Leave and Return
to the Sensor
VCC
TRIG
ECHO
GND
HC-SR04
TX
RX

Ultrasonic Sensor

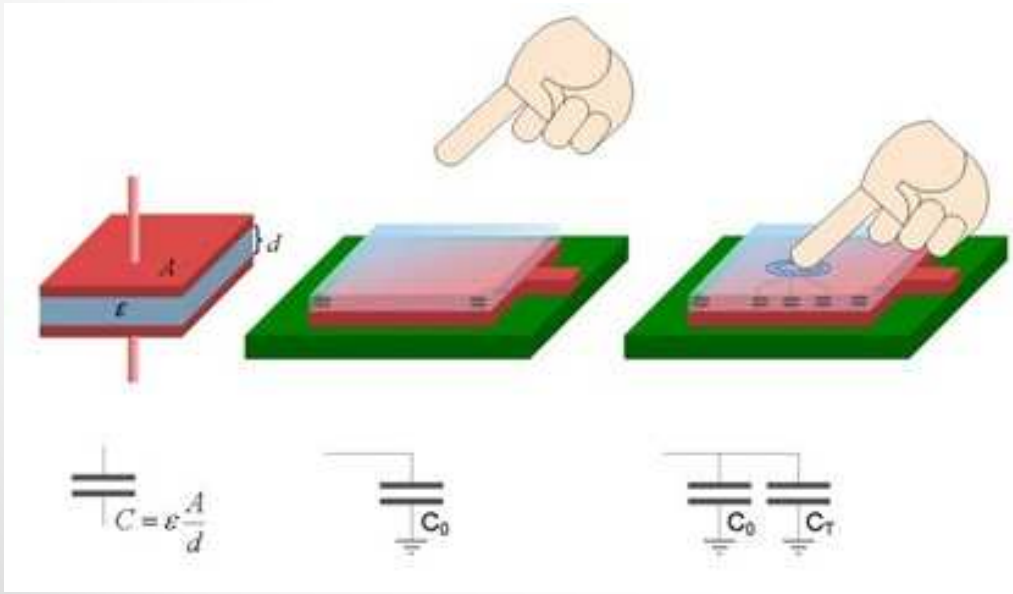


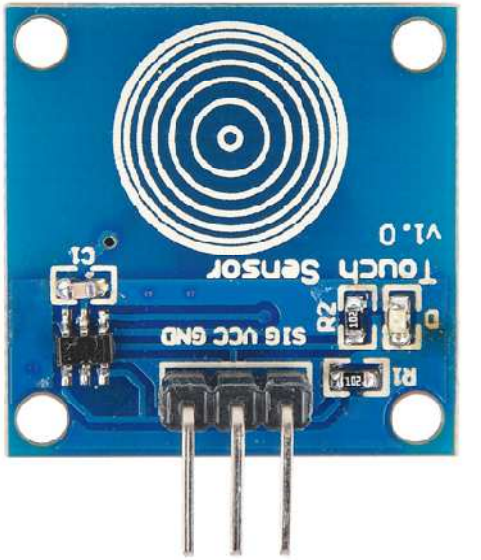
VCC pin of Sensor to 5v
GND of sensor to GND
Trig pin of sensor to GPIO 23
Echo pin of sensor to GPIO 24

Touch Sensor

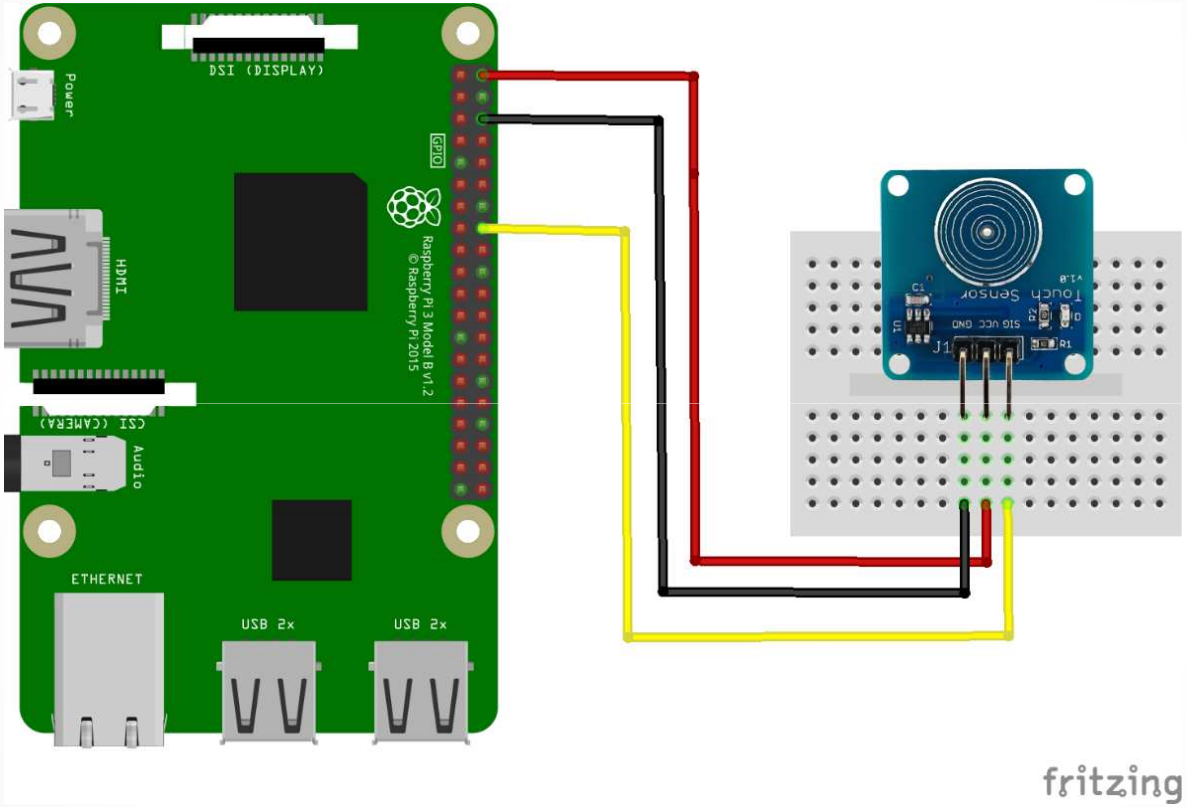
A touch sensor detects touch on physical contact.

Surface capacitive sensing



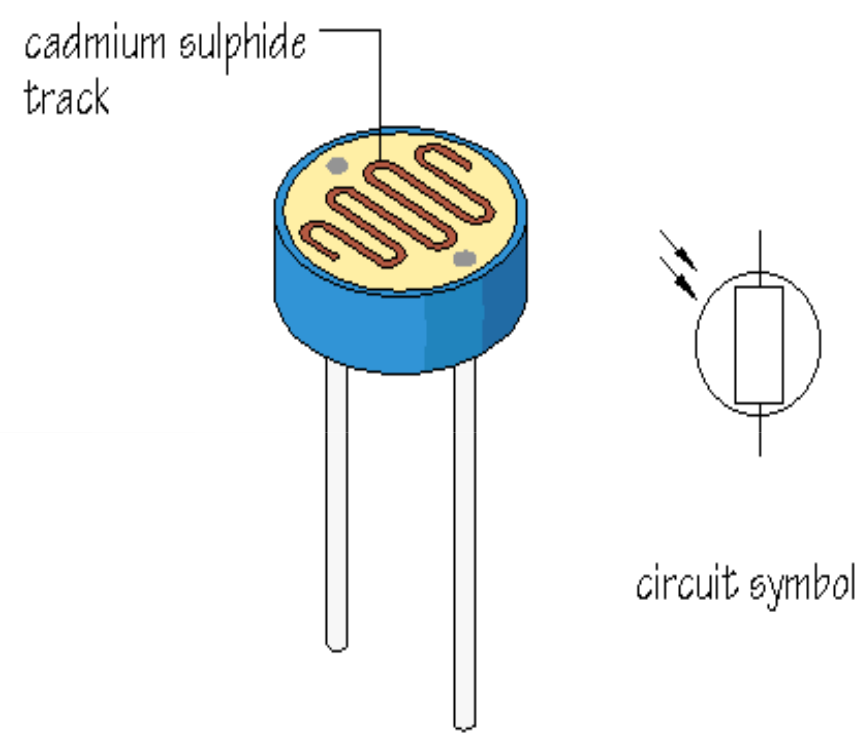


Touch Sensor With Raspberry pi

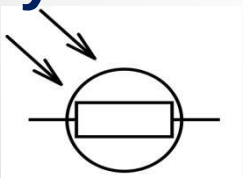


VCC pin of Sensor to 5v
GND of sensor to GND
OUT pin of sensor to GPIO 23

LDR(LIGHT DEPENDENT RESISTOR)

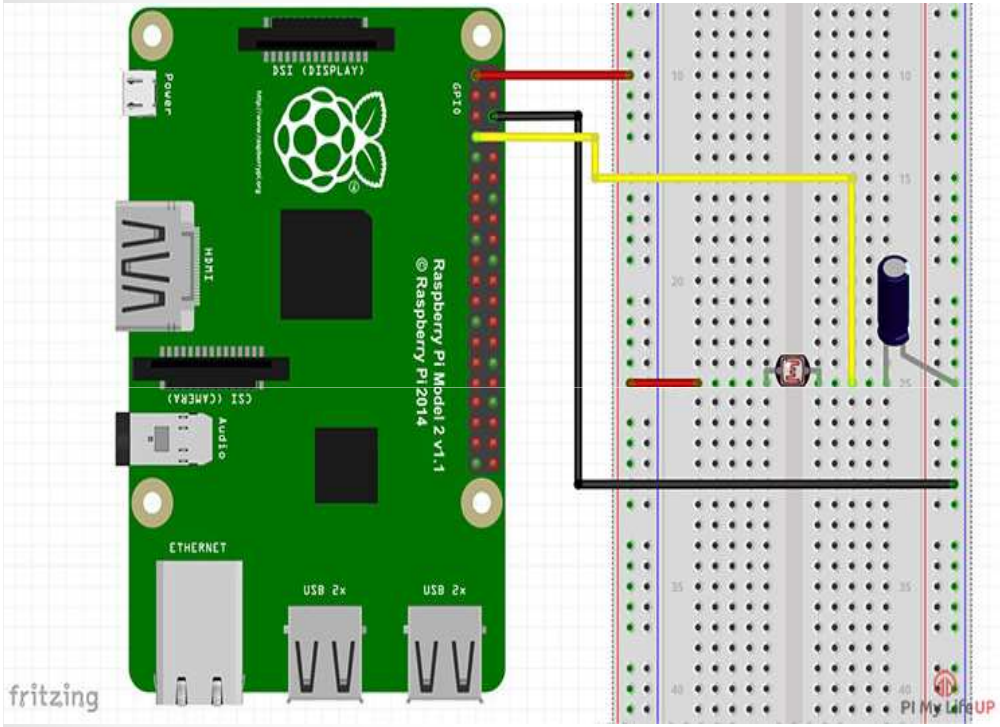


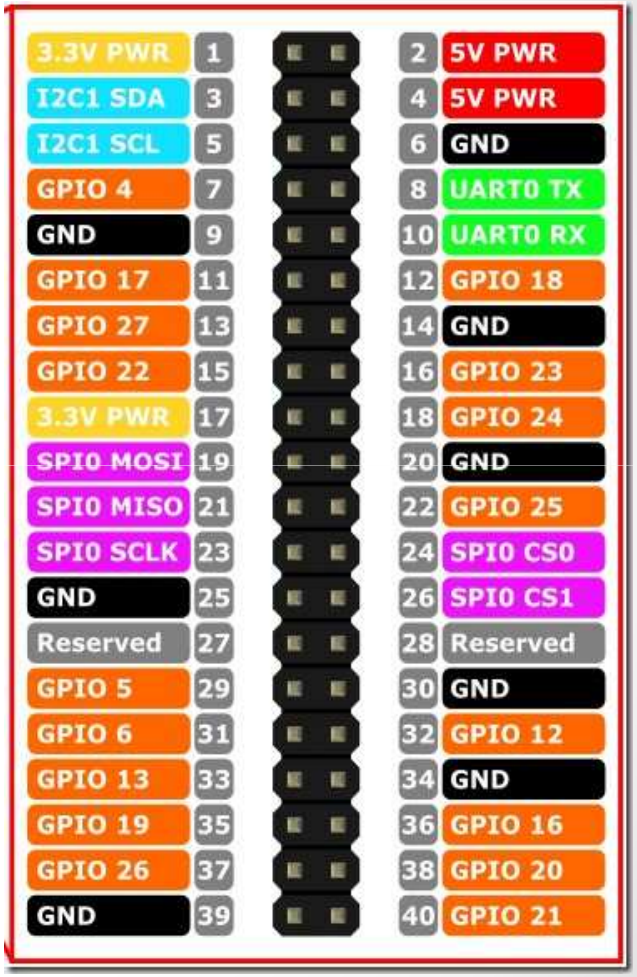
Symbol of LDR:



- **LDR stands for Light dependant resistor. An LDR is usually made of a semiconductor material(Normally Silicon) doped with a small percentage of a valency 5 material (commonly Arsenic), to make it an "N" material.**
- **Another word for LDR is photoresistor.**
- **The resistance of LDR decreases with increase in the intensity of light. An LDR works in the similar manner as any other analog device would work.**

LDR With Raspberry Pi





VCC pin of Sensor to 5v
GND of sensor to GND
Analog OUT pin of sensor to GPIO 4