# Introduction to Internet of Things

**Week 6**

**Presented By: Riya Tapwal**

**Under the supervision of**

**Prof. Sudip Misra**

**Indian Institute of Technology, Kharagpur, India**

# Arduino

**Programmer > Tools**

**1.AVRISP mkII**: This is a popular external programmer used for burning code onto AVR microcontrollers, which are the microcontrollers used in most Arduino boards. You would select this option if you have an AVRISP mkII or a compatible programmer connected to your Arduino board.

**2.USBtinyISP**: Another external programmer that is compatible with AVR microcontrollers. Select this option if you are using a USBtinyISP programmer.

**3.Arduino as ISP**: This option allows you to use one Arduino board as a programmer to upload code to another Arduino board. It's a useful option for certain advanced projects.

**4.Arduino as AVRISP mkII**: Similar to the "Arduino as ISP" option, this allows you to use an Arduino board as a programmer, emulating an AVRISP mkII programmer.

**5.Parallel Programmer**: This option is used with older Arduino boards that might have a parallel programming header.
**6.ArduinoISP**: This is a special sketch that you can upload to an Arduino board to turn it into a programmer. You would select this option when using the Arduino board as a programmer.

**7.No Programmer**: Use this option if you want to upload code using the bootloader via the USB-to-serial interface. This is the default and most common choice.

# Python

Why python?

- Python is a versatile language which is easy to script and easy to read.
- It doesn’t support strict rules for syntax.
- Its installation comes with integrated development environment for programming.
- It supports interfacing with wide ranging hardware platforms.
- With open-source nature, it forms a strong backbone to build large applications.

**Python IDE**

- Python IDE is a free and open source software that is used to write codes, integrate several modules and libraries.
- It is available for installation into PC with Windows, Linux and Mac.
- Examples: Spyder, PyCharm, etc.

# Python

Data types in python

There are 5 data types in Python:

✓ Numbers

x, y, z = 10, 10.2

✓ String

x = ‘This is Python’

print x >>This is Python

print x[0] >>T

print x[2:4] >>is

✓ List

x = [10, 10.2, 'python’]

✓ Tuple

✓ Dictionary

d = {1:‘item','k':2}

# Python

Controlling Statements

```
if (cond.):
    statement 1
    statement 2
elif (cond.):
    statement 1
    statement 2
else:
    statement 1
    statement 2
```

```
while (cond.):
    statement 1
    statement 2
```

```
x = [1,2,3,4]
for i in x:
    statement 1
    statement 2
```

**Break**

```
for s in "string":
    if s == ‘n’:
        break
    print (s)
print “End”
```

**Continue**

```
for s in "string":
    if s == ‘y’:
        continue
    print (s)
print “End”
```

# Python

Functions in Python

```
def greater(x, y):

    if x > y:

        return x, y

    else:

        return y, x

val = greater(10, 100)
print(val)
```

Output:: (100,10)

# Python

Functions in Python

- Functions can also be assigned and reassigned to the variables.
- Example:

```
def add (a,b)
   return a+b
print (add(4,6))
c = add(4,6)
print c
```

Output:: 10 10

# Python

## Variable Scope in Python

**Global variables**:
These are the variables declared out of any function , but can be accessed inside as well as outside the function.

**Local variables**:
These are the ones that are declared inside a function.

```
var = 10
def example():
    var = 100
    print(var)
example() # calling the function
print(var)
```

Output:: 100
10

```
g_var = 10
def example():
    l_var = 100
    print(g_var)
example() # calling the function
```

Output:: 10

## Modules in Python

Any segment of code fulfilling a particular task that can be used commonly by everyone is termed as a module.

```
import random
for i in range(1,10):

    val = random.randint(1,10)
    print (val)
```

Output:: varies with each execution

# Python

## Exception Handling in Python

An error that is generated during execution of a program, is termed as exception.

- Syntax:

```
try:

        statements

except _Exception_:

        statements

else:

        statements
```

**Example:**

```
while True:
   try:
      n = input ("Please enter an integer: ")
      n = int (n)
       break
   except ValueError:
       print "No valid integer! "
print “It is an integer!"
```

# Python

## File Read Write Operations

- Python allows you to read and write files
- No separate module or library required
- Three basic steps
  - Open a file
  - Read/Write
  - Close the file

Opening a File:

- Open() function is used to open a file, returns a file object

open(file_name, mode)

- Mode: Four basic modes to open a file

r: read mode
w: write mode
a: append mode
r+: both read and write mode

Read from a file:

- read(): Reads from a file

```
file=open('data.txt', 'r')
file.read()
```

Write(): Writes to a file

```
file=open('data.txt', 'w')
file.write('writing to the file')
```

## File Read Write Operations: Contd…

Closing a file:

- Close(): This is done to ensure that the file is free to use for other resources file.close()

Using WITH to open a file:

- Good practice to handle exception while file read/write operation
- Ensures the file is closed after the operation is completed, even if an exception is encountered

## Image Read Write Operations: Contd…

Python supports PIL library for image related operations

- Install PIL through PIP

```
sudo pip install pillow
```

PIL is supported till python version 2.7. Pillow supports the 3x version of python.

Reading Image in Python:

- PIL: Python Image Library is used to work with image files

**from PIL import Image**

- Open an image file

**image=Image.open(image_name)**

- Display the image

**image.show()**

## Image Read Write Operations: Contd…

Convert image to different mode:

- Any image can be converted from one mode to ‘L’ or ‘RGB’ mode

**conv_image=image.convert(‘L’)**

- Conversion between modes other that ‘L’ and ‘RGB’ needs conversion into any of these 2 intermediate mode

**Converting a sample image to Grey Scale**

```
from PIL import Image

im = Image.open('/home/saswati/VRP_Linux/Images/i3.jpg')
im.show()
grey_image=im.convert('L')
grey_image.show()
grey_image.save('GreyScaleImage.jpg')
```

Image Read Write Operations: Contd…

# Python

Networking in Python

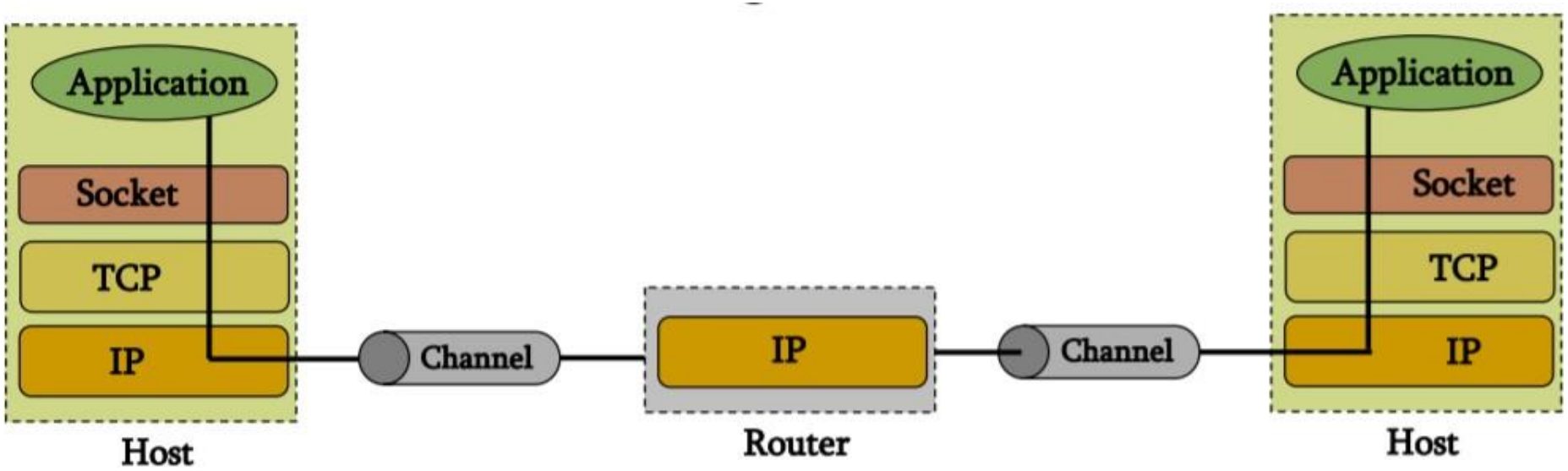
- Python provides network services for client server model.
- Socket support in the operating system allows to implement clients and servers for both connection-oriented and connectionless protocols.
- Python has libraries that provide higher-level access to specific application-level network protocols.

# WEB SOCKETS

A **socket** is one endpoint of a **two way** communication link between two programs running on the network. The socket mechanism provides a means of inter-process communication (IPC) by establishing named contact points between which the communication take place.

Types:

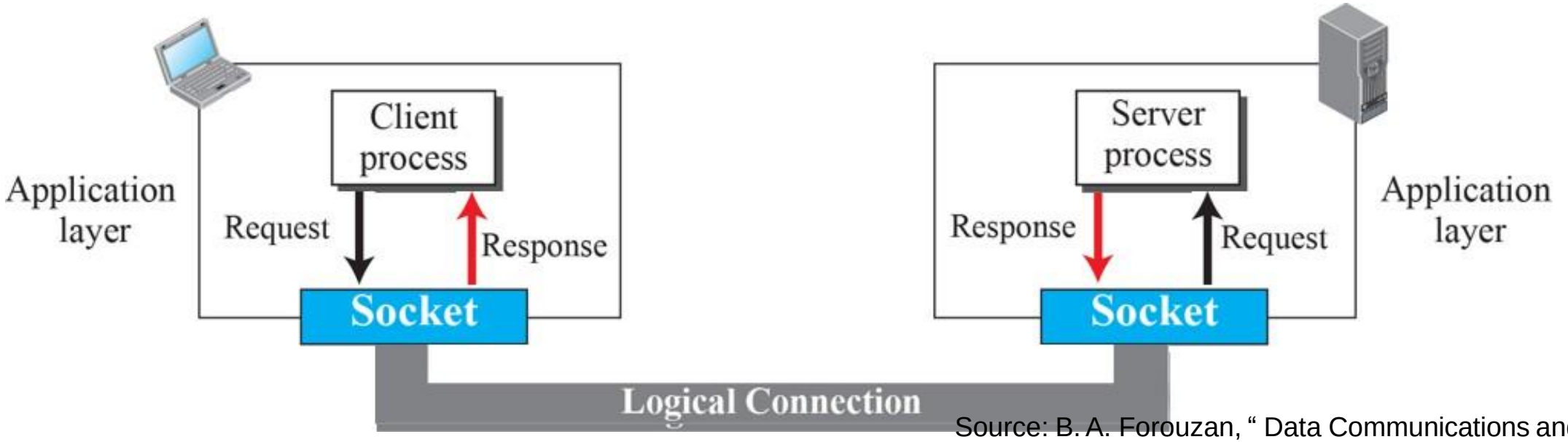
- Stream Sockets referred as "SOCK_STREAM"
- Datagram Sockets referred as "SOCK_DGRAM"
- Raw Sockets referred as "SOCK_RAW"




https://beej.us/guide/bgnet/pdf/bgnet_a4_c_1.pdf

# USE OF SOCKETS IN PROCESS-TO-PROCESS COMMUNICATION

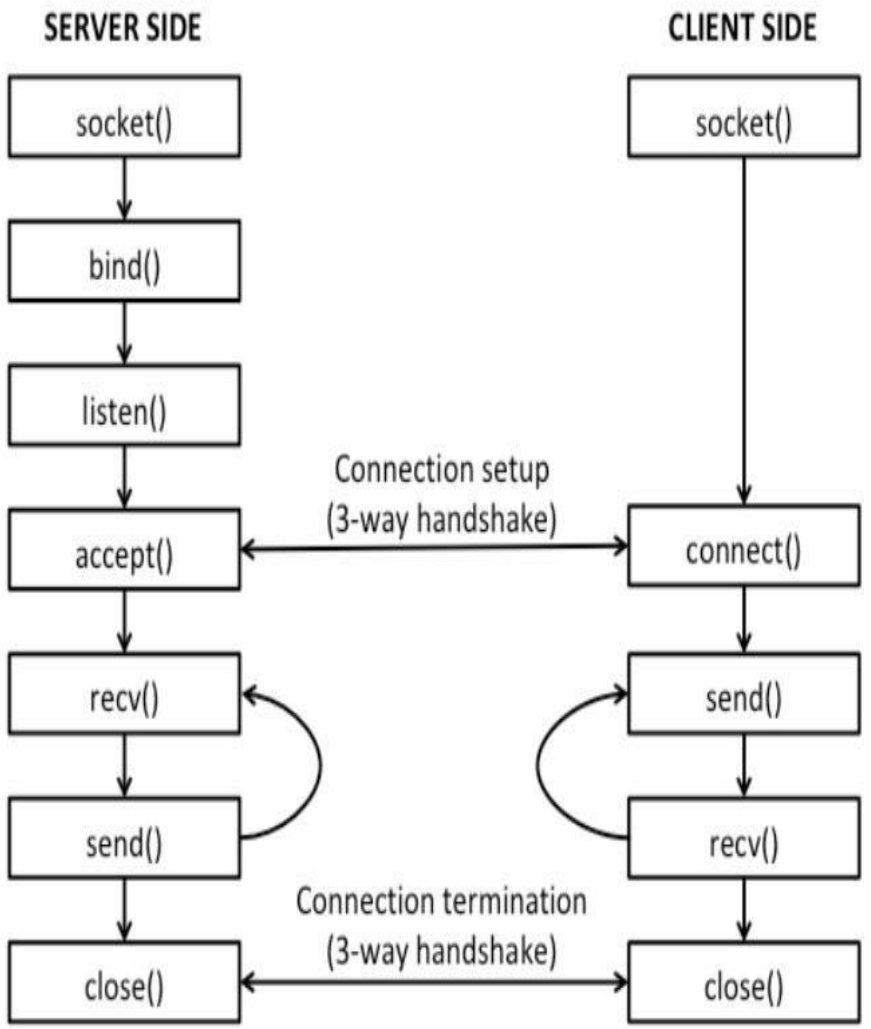
- Inter-process communication is the mechanism provided by the operating system that allows processes to communicate with each other.
- A pair of processes communicating over a network employs a pair of sockets, one for each process.




Source: B. A. Forouzan, " Data Communications and Networking ," *McGraw-Hill Forouzan Networking Series*,5E.

# SYSTEM CALLS

- The socket() system call is used to create a socket descriptor on both the client and the server.
- bind() system call is used to binds the function with the particular IP and port.
- listen() system call hears to the various connections present in the network and selects the particular client to serve.
- The connect() function is then called on the client with three arguments, namely the socket descriptor, the remote server address and the length of the address data structure.
- Accept() extracts a connection on the buffer of pending connections in the system.
- A send() is used to send the message either from the client to server or vice versa.
- The recv() function is used to receive the messages at the both ends.
- The close() call is used to close the connection.



# Example

```
import socket
import sys

# Create a TCP/IP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Bind the socket to the port
server_address = ('10.14.88.82', 2017)
print >>sys.stderr, 'starting up on %s port %s' % server_address
sock.bind(server_address)

# Listen for incoming connections
sock.listen(1)

connection, client_address = sock.accept()

#Receive command
data = connection.recv(1024)
print(data)
sock.close()
```

# Example

```
import socket
import sys

# Create a TCP/IP socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

#Connect to Listener socket
client_socket.connect(("10.14.88.82", 2017))
print>>sys.stderr,'Connection Established'

#Send command
client_socket.send('Message to the server')
print('Data sent successfully')
```

# Example

```
starting up on 10.14.88.82 port 2017
Message to the server
saswati@saswati-BK361AA-ACJ-CQ3236IX:~/Desktop$
```

```
Connection Established
Data sent successfully
saswati@saswati-BK361AA-ACJ-CQ3236IX:~/Desktop$
```

# MULTICASTING USING WEB SOCKETS

Sending messages separately to each recipient consumes extra bandwidth and processing time.

Using multicast achieves better efficiency.

```
import socket
import struct
import sys

message = 'very important data'
multicast_group = ('224.3.29.71', 10000)

# Create the datagram socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Set a timeout so the socket does not block indefinitely when trying
# to receive data.
sock.settimeout(0.2)
# Set the time-to-live for messages to 1 so they do not go past the
# local network segment.
ttl = struct.pack('b', 1)
sock.setsockopt(socket.IPPROTO_IP, socket.IP_MULTICAST_TTL, ttl)
```

Rest the code remains same.

https://pymotw.com/2/socket/multicast.html

# Example

**Simple client-server program in Python, where the client requests the current time and date from the server and then prints the same.**

```python
import socket
import datetime

# Create a TCP/IP socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Bind the socket to a specific address and port
server_address = ('localhost', 12345)
print('Starting up server on %s port %s' % server_address)
server_socket.bind(server_address)

# Listen for incoming connections
server_socket.listen(1)

while True:
    # Wait for a connection
    print('Waiting for a connection...')
    connection, client_address = server_socket.accept()
    print('Connection from', client_address)

    try:
        # Send the current time and date to the client
        now = datetime.datetime.now()
        message = now.strftime('%Y-%m-%d %H:%M:%S')
        connection.sendall(message.encode())
    finally:
        # Clean up the connection
        connection.close()
```

```python
import socket

# Create a TCP/IP socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# Connect the socket to the server's address and port
server_address = ('localhost', 12345)
print('Connecting to %s port %s' % server_address)
client_socket.connect(server_address)

try:
    # Send a request to the server
    request = 'Please send me the current time and date'
    client_socket.sendall(request.encode())

    # Receive the server's response
    response = client_socket.recv(1024).decode()
    print('The current time and date is:', response)

finally:
    # Clean up the connection
    client_socket.close()
```

# Raspberry Pi

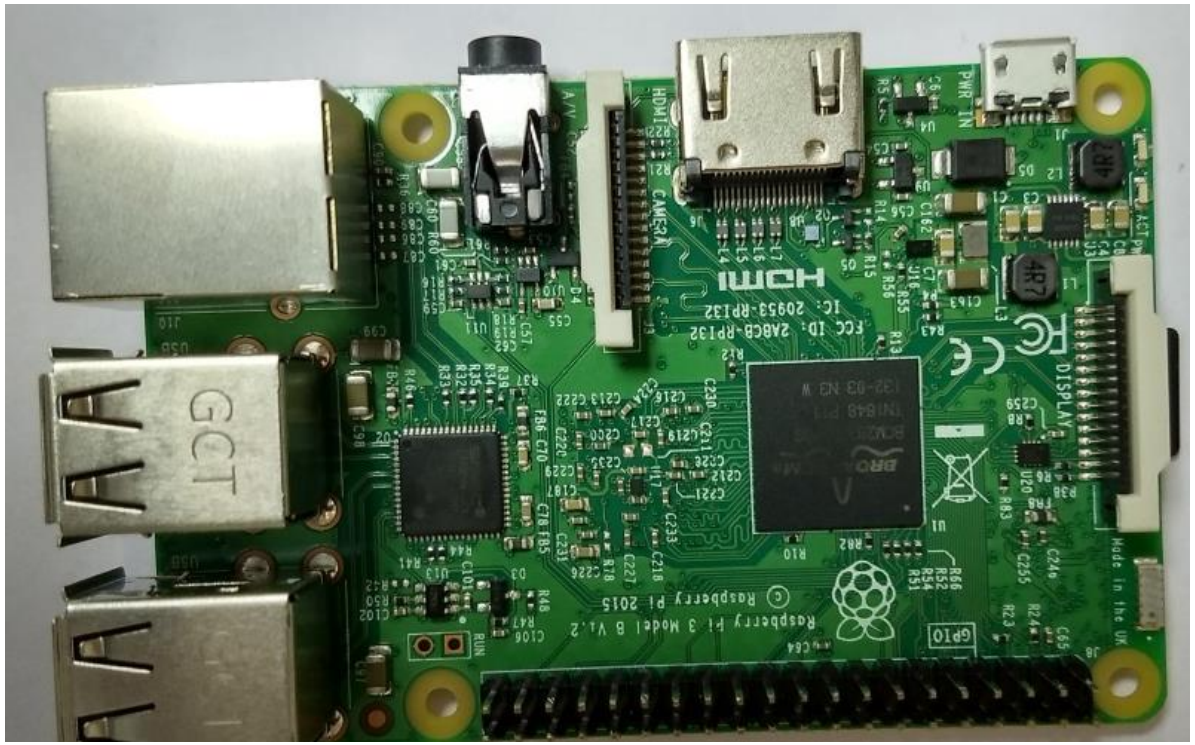
What is Raspberry Pi?

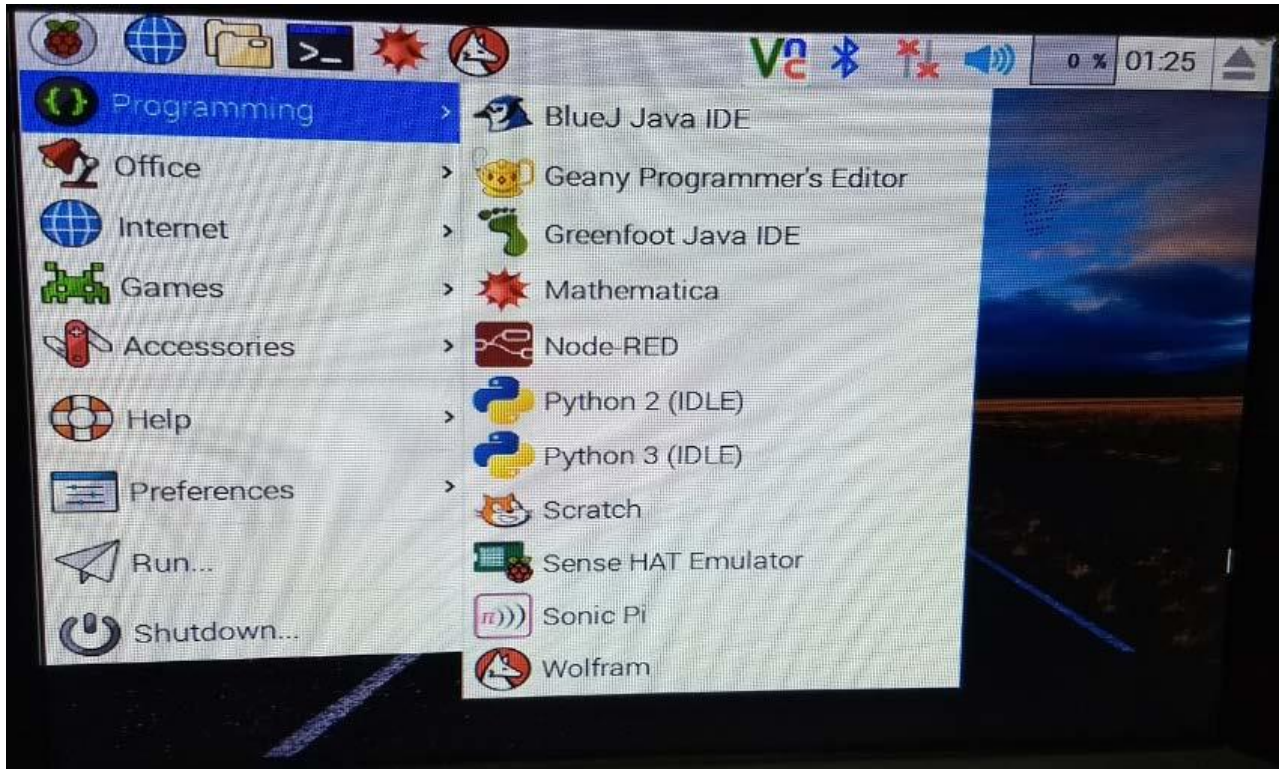
- Computer in your palm.
- Single-board computer.
- Low cost.
- Easy to access.

| Key features | Raspberry pi 3 model B | Raspberry pi 2 model B | Raspberry Pi zero |
|---|---|---|---|
| *RAM* | *1GB SDRAM* | *1GB SDRAM* | *512 MB SDRAM* |
| *CPU* | *Quad cortex A53@1.2GHz* | *Quad cortex A53@900MHz* | *ARM 11@ 1GHz* |
| *GPU* | *400 MHz video core IV* | *250 MHz video core IV* | *250 MHz video core IV* |
| *Ethernet* | *10/100* | *10/100* | *None* |
| *Wireless* | *802.11/Bluetooth 4.0* | *None* | *None* |
| *Video output* | *HDMI/Composite* | *HDMI/Composite* | *HDMI/Composite* |
| *GPIO* | *40* | *40* | *40* |

# Raspberry Pi

Start up of Raspberry Pi

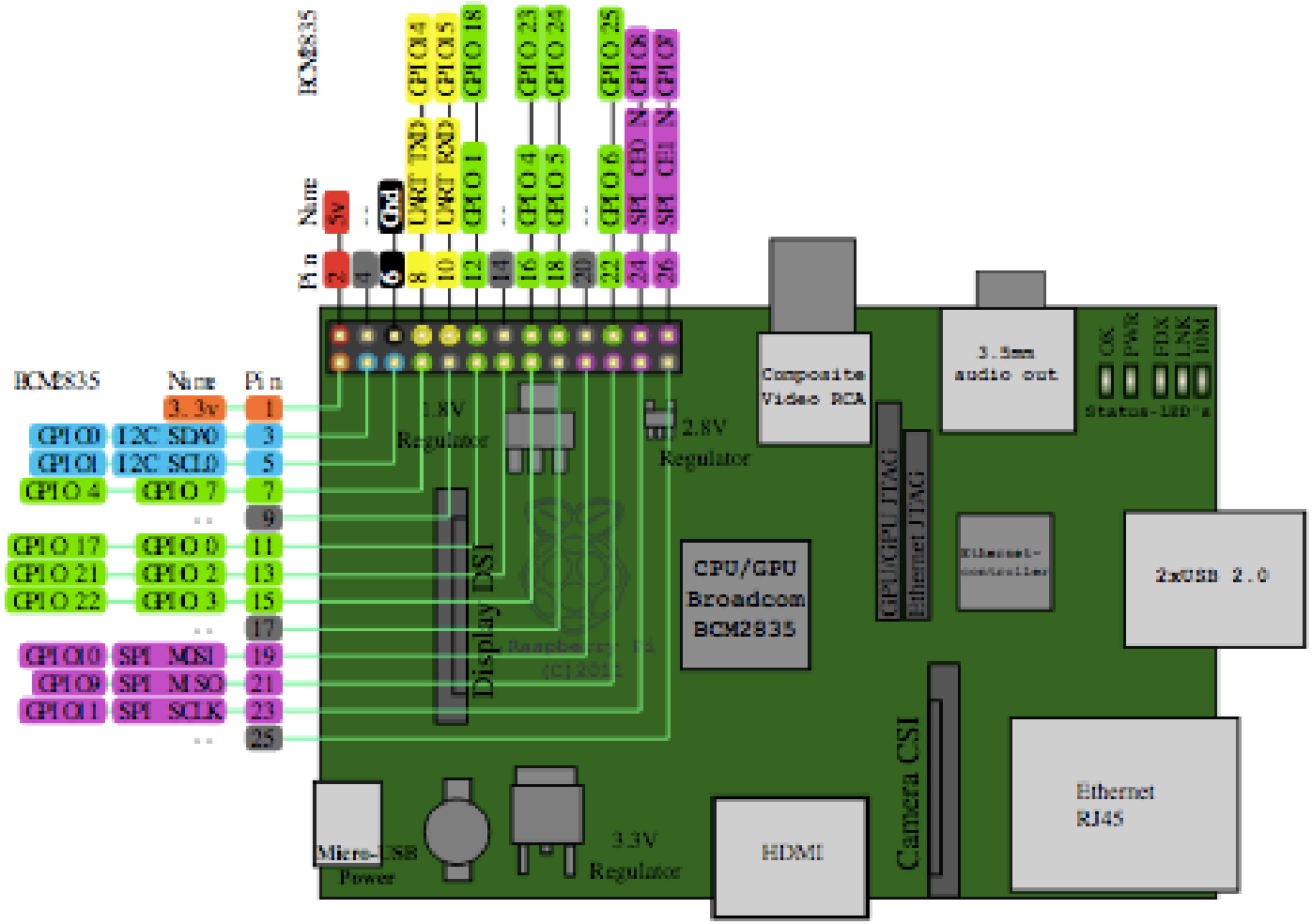





- Act as both digital output and digital input.
- **Output**: turn a GPIO pin high or low.
- **Input**: detect a GPIO pin high or low

# Raspberry Pi

Pin Configuration




# Raspberry Pi

Set up of Raspberry Pi

- HDMI cable.
- Monitor.
- Key board.
- Mouse.
- 5volt power adapter for raspberry pi.
- LAN cable .
- Min- 2GB micro sd card

**Default installed :**

- Python
- C
- C++
- Java
- Scratch
- Ruby

**Official Supported OS :**

- Raspbian
- NOOBS

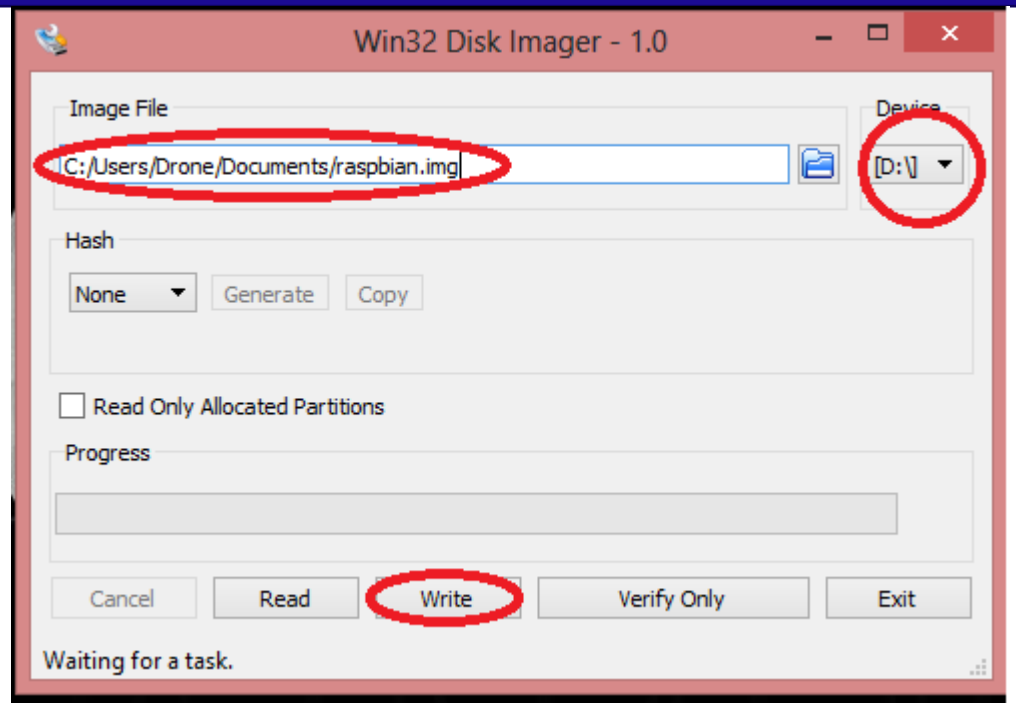
**Some of the third party OS :**

- UBUNTU mate
- Snappy Ubuntu core
- Windows 10 core
- Pinet
- Risc OS

# Raspberry Pi OS Setup

## Write Raspbian in SD card :

- Install “Win32 Disk Imager” software in windows machine .
- Run Win32 Disk Imager
- Plug SD card into your PC
- Select the “Device”
- Browse the “Image File”(Raspbian image)
- Write



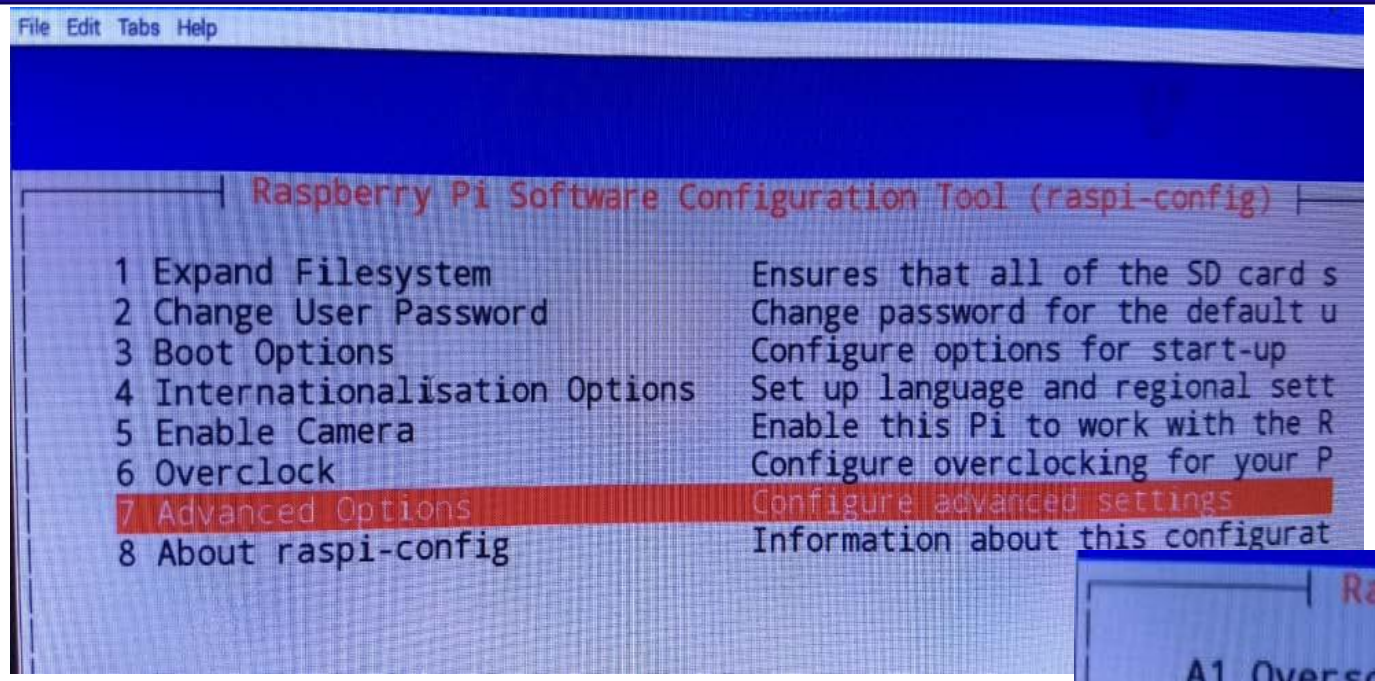

## Enable SSH

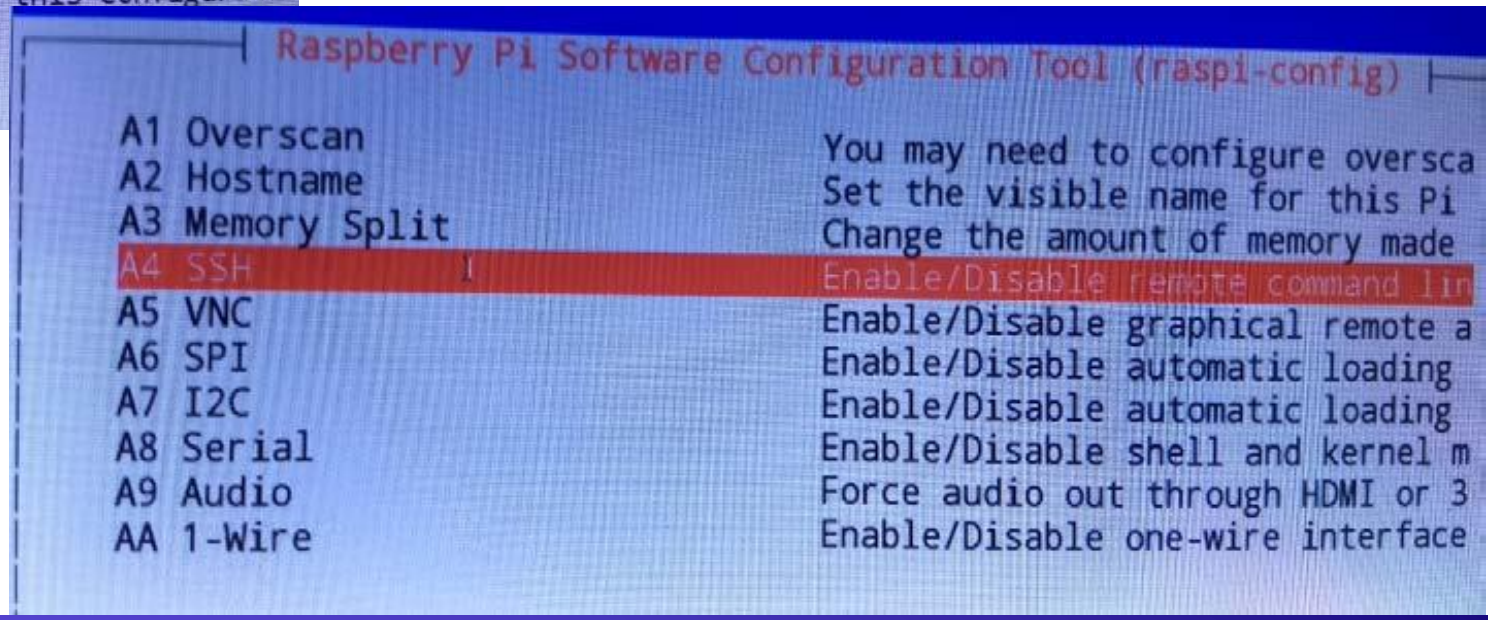
**Step1 :** Open command prompt and type **sudo raspi-config** and press enter.

**Step2:** Navigate to SSH in the Advance option.

**Step3:** Enable SSH

# Raspberry Pi OS Setup

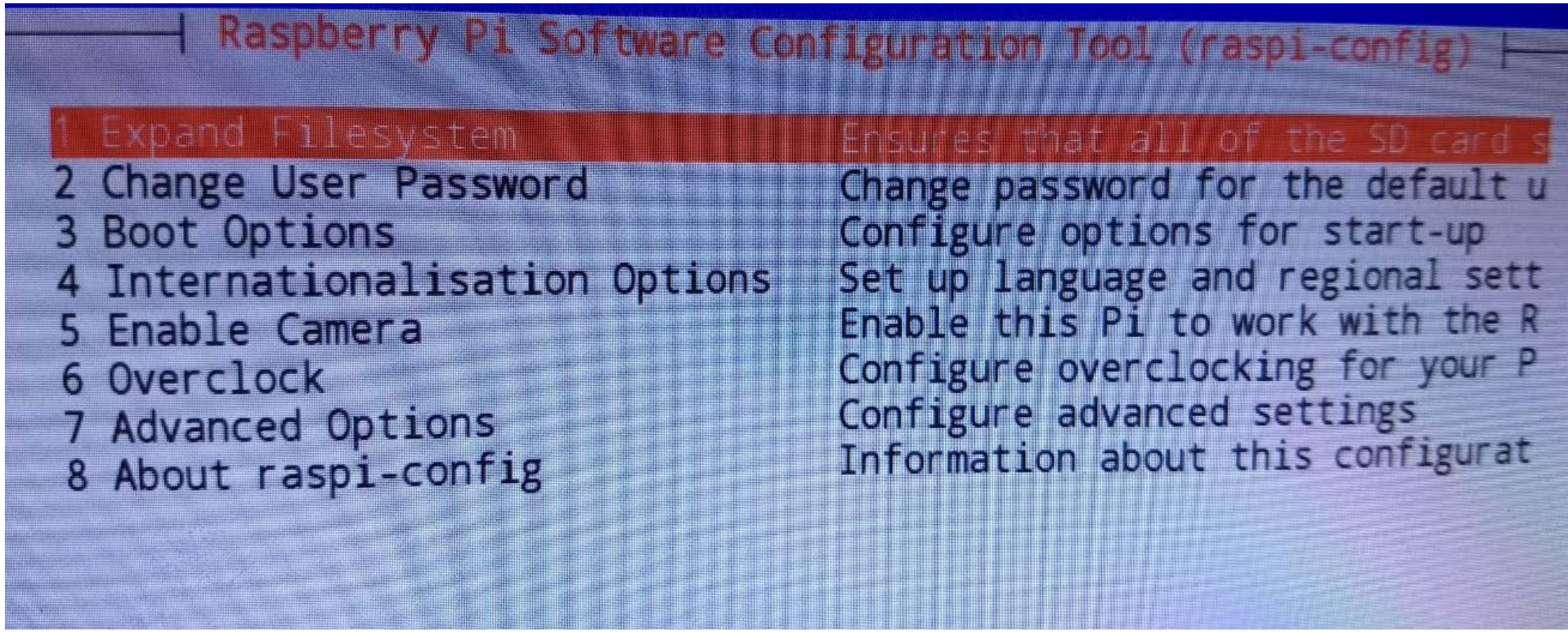


# Raspberry Pi OS Setup

## Expand file system :

**Step 1:** Open command prompt and type **sudo raspi-config** and press enter.
**Step 2:** Navigate to Expand Filesystem
**Step 3:** Press enter to expand it.



# Raspberry Pi

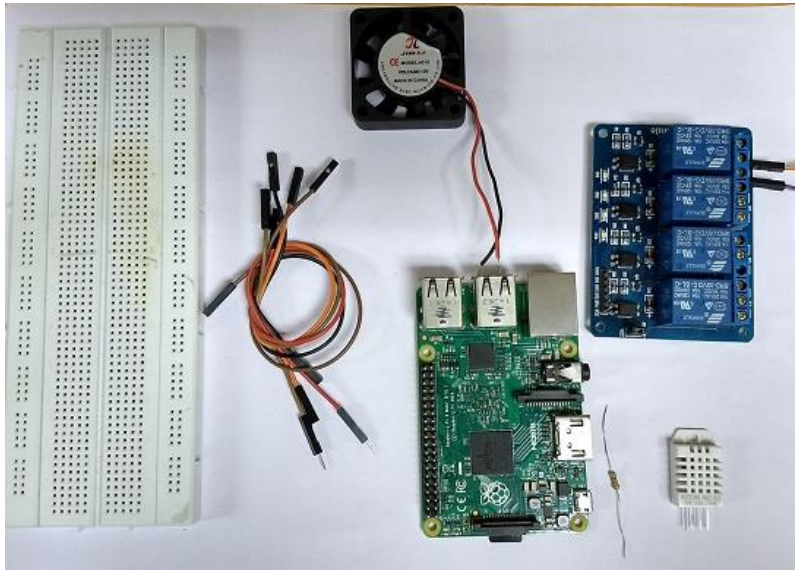
```
 GNU nano 2.2.6                                    File: BLINK_LED.py

import RPi.GPIO as GPIO ## GPIO library
import time
GPIO.setmode(GPIO.BOARD) ## Set the type of board for pin numbering
GPIO.setup(11, GPIO.OUT) ## Set GPIO pin 11 as output pin
for i in range (0,5):
    GPIO.output(11,True) ## Turn on GPIO pin 11
    time.sleep(1)
    GPIO.output(11,False)
    time.sleep(2)
    GPIO.output(11,True)
GPIO.cleanup()
```

# Raspberry Pi

Applications

- Media streamer
- Home automation
- Controlling BOT
- VPN
- Light weight web server for IOT
- Tablet computer



**Integration with sensors and actuators**

- Sensor and actuator interfaced with Raspberry Pi
- Read data from the sensor
- Control the actuator according to the reading from the sensor
- Connect the actuator to a device

Requirements

- DHT Sensor
- 4.7K ohm resistor
- Relay
- Jumper wires
- Raspberry Pi
- Mini fan

# Raspberry Pi

```
  GNU nano 2.2.6                    File: IOTSR.py

import RPi.GPIO as GPIO
from time import sleep

import Adafruit_DHT

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

sensor = Adafruit_DHT.AM2302  # create an instance of the sensor type

print ('Getting data from the sensor')

#humidity and temperature are 2 variables that store the values received from the sensor
humidity, temperature = Adafruit_DHT.read_retry(sensor,17)

print ('Temp={0:0.1f}*C humidity={1:0.1f}%'.format(temperature, humidity))
```

```
pi@raspberrypi:~ $ python IOTSR.py
Getting data from the sensor
Temp=26.1*C humidity=65.9%
pi@raspberrypi:~ $
```

# Question No: 1

What is the output for the following piece of Python code?

```
x = [32, 'u', 'i', 8, '34']
x = x[0:]
print (x)
```

a. [32]
b. ['34']
c. ['u', 'i', 8, '34']
d. [32,'u', 'i', 8, '34']

# Question No: 1

What is the output for the following piece of Python code?

```
x = [32, 'u', 'i', 8, '34']
x = x[0:]
print (x)
```

a. [32]
b. ['34']
c. ['u', 'i', 8, '34']
d. [32,'u', 'i', 8, '34']

# Question No: 2

How many GPIO (General Purpose Input Output Pin) pins are there in Raspberry Pi 4?

a. 30
b. 14
c. 40
d. 41

# Question No: 2

How many GPIO (General Purpose Input Output Pin) pins are there in Raspberry Pi 4?

a. 30
b. 14
c. 40
d. 41

# Question No: 3

Which of the following is NOT an example of Python IDE?

a. Sublime Text
b. PyCharm
c. Spider
d. None of the above

# Question No: 3

Which of the following is NOT an example of Python IDE?

a. Sublime Text
b. PyCharm
c. Spider
d. None of the above

Sublime Text, PyCharm, Spyder, and Jupyter are few of the examples of Python IDE.

# Question No: 4

What is the value of ‘x’ in the following expression in Python programming?

```
x = 3^3
```

a. 0
b. 9
c. -2
d. Will raise an exception

# Question No: 4

What is the value of ‘x’ in the following expression in Python programming?

x = 3^3

a. 0
b. 9
c. -2
d. Will raise an exception

➢ “^” is a bitwise XOR operator in python.

# Question No: 5

In python programming, which of the following is a null statement?

a. Pass
b. Continue
c. Break
d. Skip

# Question No: 5

In python programming, which of the following is a null statement?

a. Pass
b. Continue
c. Break
d. Skip

As per basics of Python programming, pass is a null operation. The interpreter does not ignore a pass statement, but nothing happens and the statement results into no operation.

# Question No: 6

Which of the following is an unordered data type in Python?

a. List
b. Dictionary
c. Both List and Dictionary
d. Tuple

# Question No: 6

Which of the following is an unordered data type in Python?

a. List
b. Dictionary
c. Both List and Dictionary
d. Tuple

# Question No: 7

What is the value of ‘x’ in the following expression in Python programming?
x = 2**3^3**1

a. 11
b. 0
c. Will raise an error
d. Will raise an exception

# Question No: 7

What is the value of ‘x’ in the following expression in Python programming?
x = 2**3^3**1

a. 11
b. 0
c. Will raise an error
d. Will raise an exception

# Question No: 8

Which of the following symbol is used to comment out multiple lines at once in python?

a. **$**
b. #
c. **%**
d. None of these

# Question No: 8

Which of the following symbol is used to comment out multiple lines at once in python?

a. **$**
b. #
c. **%**
d. None of these

Unlike other programming languages Python doesn't support multi-line comment blocks out of the box. The recommended way to comment out multiple lines of code in Python is to use consecutive # single-line comments.

# Question No: 9

With on-board Raspberry Pi camera, which one of the following is NOT correct for image related operations in Python?

a. from PIcamera import PiCamera
b. from picamera import PyCamera
c. from pycamera import PiCamera
d. All of these

# Question No: 9

With on-board Raspberry Pi camera, which one of the following is NOT correct for image related operations in Python?

a. from PIcamera import PiCamera
b. from picamera import PyCamera
c. from pycamera import PiCamera
d. All of these

➢ the correct statement is “from picamera import PiCamera

# Question No: 10

Which of the following bit processor is used in Raspberry Pi Zero 2 W?

a. 64
b. 32
c. Both 64 & 32
d. 128

# Question No: 10

Which of the following bit processor is used in Raspberry Pi Zero 2 W?

a. 64
b. 32
c. Both 64 & 32
d. 128

The Raspberry Pi Zero 2 W includes a new Broadcom BCM2710A1 SOC providing a 1GHz quad-core 64-bit ARM Cortex-A53 CPU and 512MB RAM.

# Question No: 11

Which of the following shortcut exits the nano editor?

a. Ctrl + E
b. Ctrl + O
c. Ctrl + X
d. None of these

# Question No: 11

Which of the following shortcut exits the nano editor?

a. Ctrl + E
b. Ctrl + O
c. Ctrl + X
d. None of these

Ctrl + O writes the code to a file. Ctrl + X exits the nano editor.

# Question No: 12

What of the following syntax is correct for networking in python?

a. s = socket.socket(socket.AF_NET, socket.SOCK_STRAM)
b. s = socket.socket(socket.AF_UNIX, socket.SOCK_DGRAM)
c. s = socket.socket(socket.AF_UNX, socket.SOCK_TCP)
d. s = socket.socket(socket.AF_INET, socket.SOCK_UDP)

# Question No: 12

What of the following syntax is correct for networking in python?

a. s = socket.socket(socket.AF_NET, socket.SOCK_STRAM)
b. s = socket.socket(socket.AF_UNIX, socket.SOCK_DGRAM)
c. s = socket.socket(socket.AF_UNX, socket.SOCK_TCP)
d. s = socket.socket(socket.AF_INET, socket.SOCK_UDP)

AF_UNIX or AF_INET represent socket family. SOCK_STREAM or SOCK_DGRAM represent socket type.

**socket_family** - AF_UNIX or AF_INET
**socket_type** - SOCK_STREAM or SOCK_DGRAM
**protocol** - default ‘0’.

# Question No: 13

Which of the following is correct to open a text file for write mode?

a. open('file.txt', ‘w’)
b. open('file.txt', ‘+w’)
c. open('file.txt', ‘w+’)
d. open('file.txt', ‘o+w’)

# Question No: 13

Which of the following is correct to open a text file for write mode?

a. open('file.txt', ‘w’)
b. open('file.txt', ‘+w’)
c. open('file.txt', ‘w+’)
d. open('file.txt', ‘o+w’)

‘w’ mode is for writing into a text file. ‘r’ mode is for reading a text file.

# Question No: 14

Which of the following is an immutable data type in Python?

a. String
b. Tuple
c. Bool
d. All of these

# Question No: 14

Which of the following is an immutable data type in Python?

a. String
b. Tuple
c. Bool
d. All of these

Immutable Objects are of in-built types like int, float, bool, string, unicode, tuple. In simple words, an immutable object can't be changed after it is created.

# Question No: 15

Does python follow rigid indentation?

a. Yes
b. No
c. Not Applicable

# Question No: 15

Does python follow rigid indentation?

a. Yes
b. No
c. Not Applicable

# Question No: 16

What is the value that is assigned to the variable f in the given piece of python code?

i, f, str=50, 50.68, “Welcome to python”

a. 50
b. 50.68
c. Welcome to python
d. All of these

# Question No: 16

What is the value that is assigned to the variable f in the given piece of python code?

i, f, str=50, 50.68, “Welcome to python”

a. 50
b. 50.68
c. Welcome to python
d. All of these

# Question No: 17

What is the output of the following piece of python code?

```
x=&#39;17&#39;
y=&#39;23&#39;
z=x+y
print(z)
```

a. 40
b. 6
c. 1723
d. 30

# Question No: 17

What is the output of the following piece of python code?

```
x=&#39;17&#39;
y=&#39;23&#39;
z=x+y
print(z)
```

a. 40
b. 6
c. 1723
d. 30

# Question No: 18

Fill in the blanks. Raspbian is a/n _____

a. Microcomputer
b. Minicomputer
c. Operating system
d. Assembler

# Question No: 18

Fill in the blanks. Raspbian is a/n _____

a. Microcomputer
b. Minicomputer
c. Operating system
d. Assembler

# Question No: 19

What is the output of the following piece of Python code?
t1 = 'Welcome to python coding'
print(t1[8:14])

a. to pyth
b. SyntaxError: invalid syntax
c. e to pyt
d. to pyt

# Question No: 19

What is the output of the following piece of Python code?
t1 = 'Welcome to python coding'
print(t1[8:14])

a. to pyth
b. SyntaxError: invalid syntax
c. e to pyt
d. to pyt

This slice includes all characters from index 8 up to, but not including, index 14. Therefore, the output is the substring "to pyt", which consists of the characters at indices 8 through 13 of the original string **t1**.

# Question No: 20

Fill in the blanks. Raspberry Pi 3 Model B has a GPU support of _____

a. 400 MHz video core IV
b. 250 MHz video core IV
c. Quad cortex A53@1.2GHz
d. ARM 11 @ 1 GHz

# Question No: 20

Fill in the blanks. Raspberry Pi 3 Model B has a GPU support of _____

a. 400 MHz video core IV
b. 250 MHz video core IV
c. Quad cortex A53@1.2GHz
d. ARM 11 @ 1 GHz

# Question No: 21

Which of the following represents the command used for rebooting Raspberry Pi?

a. sudo reboot
b. sudo apt-get rebooting
c. pip install rebooting
d. All of these

# Question No: 21

Which of the following represents the command used for rebooting Raspberry Pi?

a. sudo reboot
b. sudo apt-get rebooting
c. pip install rebooting
d. All of these

# Question No: 22

State whether true or false.
It is not possible to return multiple values from a function in Python.

a. True
b. False

# Question No: 22

State whether true or false.
It is not possible to return multiple values from a function in Python.

a. True
b. False

# Question No: 23

What is the data type of the variable ls in the following piece of Python code?
ls= {1: "item", "key": "21", "year": 2022}

a. dictionary
b. list
c. tuple
d. All of these

# Question No: 23

What is the data type of the variable ls in the following piece of Python code?
ls= {1: "item", "key": "21", "year": 2022}

a. dictionary
b. list
c. tuple
d. All of these

# Question No: 24

State whether true or false.
A function in Python may or may not return a value.

a. True
b. False

# Question No: 24

State whether true or false.
A function in Python may or may not return a value.

a. True
b. False

# Question No: 25

What are the basic modes to open a file in python?

a. Read mode (r) and write mode (w)
b. Append mode (a)
c. Both read and write mode (r+)
d. All of these

# Question No: 25

What are the basic modes to open a file in python?

a. Read mode (r) and write mode (w)
b. Append mode (a)
c. Both read and write mode (r+)
d. All of these

# Question No: 26

What are the socket types that exist in Python based socket programming?

a. AF_DG and SOCK_SM
b. AF_UNIX and AF_INET
c. SOCK_UX and SOCK_IT
d. SOCK_DGRAM and SOCK_STREAM

# Question No: 26

What are the socket types that exist in Python based socket programming?

a. AF_DG and SOCK_SM
b. AF_UNIX and AF_INET
c. SOCK_UX and SOCK_IT
d. SOCK_DGRAM and SOCK_STREAM

# Question No: 27

Does Python support exception handling?

a. Yes
b. No

# Question No: 27

Does Python support exception handling?

a. Yes
b. No

# Question No: 28

Which of the following must be used to terminate a loop and move to the next code after the loop?

a. list
b. try
c. continue
d. break

# Question No: 28

Which of the following must be used to terminate a loop and move to the next code after the loop?

a. list
b. try
c. continue
d. break

# Question No: 29

Select the option that does not represent a keyword in Python language?

a. while
b. if
c. try
d. integer

# Question No: 29

Select the option that does not represent a keyword in Python language?

a. while
b. if
c. try
d. integer

# Question No: 30

Raspberry Pi does not support any other language other than Python?

a. True
b. False

# Question No: 30

Raspberry Pi does not support any other language other than Python?

a. True
b. False

# Question No: 31

Python’s installation comes with integrated development environment for programming.

True
False

# Question No: 31

Python’s installation comes with integrated development environment for programming.

True
False

# Question No: 32

Fill in the blanks. Python IDE is available for installation into PC with _____.

Windows
Linux
Mac
All of these

# Question No: 32

Fill in the blanks. Python IDE is available for installation into PC with _____.

Windows
Linux
Mac
All of these

# Question No: 33

Is relay a type of mechanical switch?

No
Yes

# Question No: 33

Is relay a type of mechanical switch?

No

Yes

# Question No: 34

How many data type/s are available in Python.

1
2
5
7

# Question No: 34

How many data type/s are available in Python.

1
2
5
7

# Question No: 35

The variable that is declared inside the function in Python is called a Global variable.

True
False

# Question No: 35

The variable that is declared inside the function in Python is called a Global variable.

True
False

# Question No: 36

Which of the following is used to read a text file in Python?

file = open(‘data.txt’, ‘r ’)
file = open_text(‘data.txt’, ‘r ’)
file = read_text(‘data.txt’, ‘r ’)
file = read(‘data.txt’, ‘r ’)

# Question No: 36

Which of the following is used to read a text file in Python?

file = open(‘data.txt’, ‘r ’)
file = open_text(‘data.txt’, ‘r ’)
file = read_text(‘data.txt’, ‘r ’)
file = read(‘data.txt’, ‘r ’)

# Question No: 37

Which of the following library in Python is used for processing images.

Pillow
Numpy
Panda
None of these

# Question No: 37

Which of the following library in Python is used for processing images.

Pillow
Numpy
Panda
None of these

# Question No: 38

In python, image cannot be converted to grey scale.

True
False

# Question No: 38

In python, image cannot be converted to grey scale.

True
False

# Question No: 39

Which of the following shortcut exits the nano editor?

Ctrl + E
Ctrl + O
Ctrl+V
None of these

# Question No: 39

Which of the following shortcut exits the nano editor?

Ctrl + E
Ctrl + O
Ctrl+V
None of these

# Question No: 40

Which of the following is an unordered data type in Python?

List
Dictionary
Both List and Dictionary
Tuple

# Question No: 40

Which of the following is an unordered data type in Python?

List
Dictionary
Both List and Dictionary
Tuple

# Question No: 41

Which of the following converts energy to motion?

Actuator
Raspberry Pi
Sensor
None of these

# Question No: 41

Which of the following converts energy to motion?

Actuator
Raspberry Pi
Sensor
None of these

# Question No: 42

Python does not follow strict indentation.

True
False

# Question No: 42

Python does not follow strict indentation.

True
False

# Question No: 43

Functions cannot be reassigned to the variables in Python.

True
False

# Question No: 43

Functions cannot be reassigned to the variables in Python.

True
False

# Question No: 44

You can execute Python programs within a Raspberry Pi

a. True
b. False

# Question No: 44

You can execute Python programs within a Raspberry Pi

a. True
b. False

# Question No: 45

Which among the following libraries in Python do you use to generate and plot graphs

a. numpy
b. time
c. matplotlib
d. random

# Question No: 45

Which among the following libraries in Python do you use to generate and plot graphs

a. numpy
b. time
c. matplotlib
d. random

# Question No: 46

A programmer needs to execute a python code that requires RAM support of 1GB SDRAM, Quad cortex A53@1.2GHz CPU, 400 MHz video core IV GPU, and 802.11 wireless communication support. Select the appropriate device where the code can be executed?

a. Raspberry Pi zero
b. Raspberry Pi 2 model B
c. Raspberry Pi 3 model B
d. Arduino Uno

# Question No: 46

A programmer needs to execute a python code that requires RAM support of 1GB SDRAM, Quad cortex A53@1.2GHz CPU, 400 MHz video core IV GPU, and 802.11 wireless communication support. Select the appropriate device where the code can be executed?

a. Raspberry Pi zero
b. Raspberry Pi 2 model B
c. Raspberry Pi 3 model B
d. Arduino Uno

# Question No: 47

Raspberry Pi 3 has _____ GB internal storage.

a. three
b. zero
c. four
d. one

# Question No: 47

Raspberry Pi 3 has _____ GB internal storage.

a. three
b. zero
c. four
d. one

# Question No: 48

Which among the following is a wearable Arduino board?

a. Arduino Uno

b. Arduino Mega

c. RedBoard Arduino

d. LilyPad Arduino

# Question No: 48

Which among the following is a wearable Arduino board?

a. Arduino Uno

b. Arduino Mega

c. RedBoard Arduino

d. LilyPad Arduino

# Question No: 49

In Python, the following is the syntax of the sleep function:
time.sleep(secs)
The argument <secs> should be _____.

- Integer
- Float
- Either integer or float
- Range

# Question No: 49

In Python, the following is the syntax of the sleep function:
time.sleep(secs)
The argument <secs> should be _____.

- Integer
- Float
- Either integer or float
- Range

# Question No: 50

Which of the following is the extension for your python file in Raspberry Pi?

a. .py
b. .pi
c. .rpi
d. .rpy

# Question No: 50

Which of the following is the extension for your python file in Raspberry Pi?

a. .py
b. .pi
c. .rpi
d. .rpy

# Question No: 51

What are variables that are declared and has its scope within a function in python called?

a. Global variable
b. Local variable
c. Hierarchical variable
d. Dangling variable

# Question No: 51

What are variables that are declared and has its scope within a function in python called?

a. Global variable
b. Local variable
c. Hierarchical variable
d. Dangling variable

# Question No: 52

In raspberry pi, what is the command for capturing an image?

a. raspicapture
b. raspicam
c. raspisnap
d. raspistill

# Question No: 52

In raspberry pi, what is the command for capturing an image?

a. raspicapture
b. raspicam
c. raspisnap
d. raspistill

# Thank You