UNIT-5

INTRODUCTION TO ARDUINO & RASPBERRY PI

5.1 ARCHITECTURE, PROGRAMMING & APPLICATIONS OF ARDUINO

Introduction:

Arduino is an open-source electronics platform based on easy-to-use hardware and software.

Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message and turn it into an output - activating a motor, turning on an LED, publishing something online.

You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

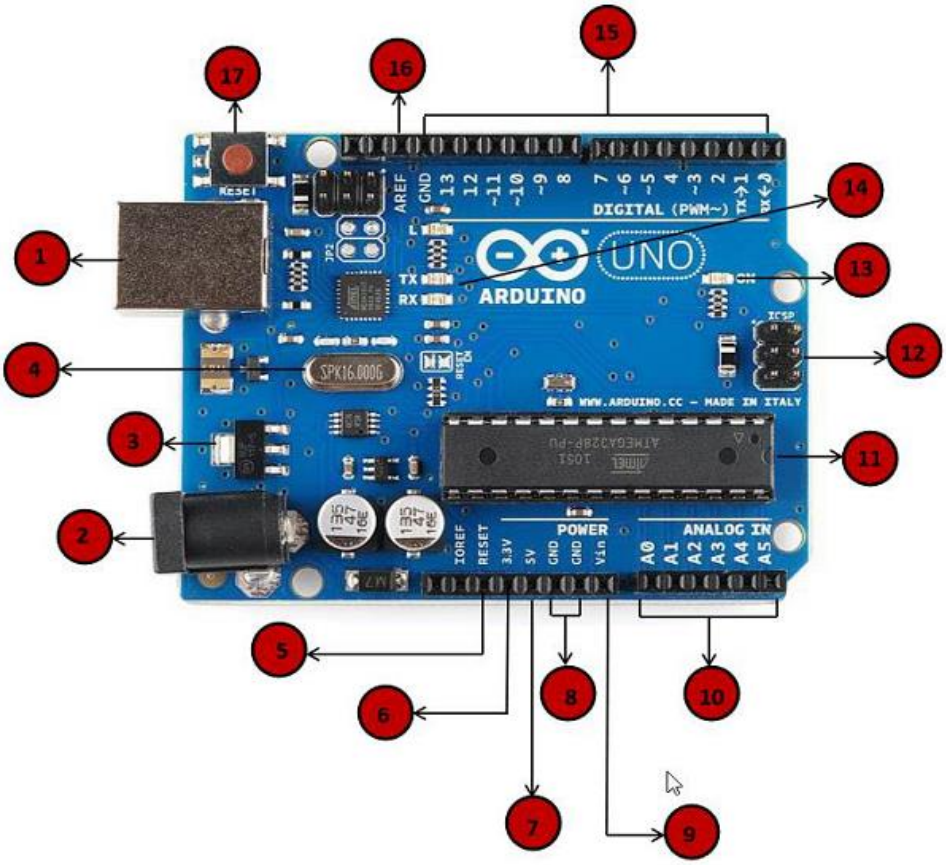
To do so you need to use the Arduino programming language (based on Wiring) and the Arduino Software (IDE) based on Processing.

Arduino also simplifies the process of working with microcontrollers.

Advantages:

1) Inexpensive
2) Cross-platform
3) Simple and clear programming environment
4) Open source and extensible software
5) Open source and extensible hardware

5.1.1 Architecture of Arduino



As shown in the above figure, labelled components of the Arduino board are explained as follows:

1) **USB:** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection. It can be used for both power and communication with the IDE
2) **Barrel Jack:** It is used for power supply. Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack.
3) **Voltage Regulator:** It regulates and stabilizes the input and output voltages. The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other elements.
4) **Crystal Oscillator:** It keeps track of time and regulates processor frequency. The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16 MHz.
5) **Reset Pin:** It can be used to reset the Arduino Uno.
6) **3.3V Pin:** It can be used as a 3.3V output.
7) **5V Pin:** It can be used as a 5V output.
8) **GND Pin:** It can be used to ground the circuit.
9) **Vin Pin:** It can be used to supply power to the board.
10) **Analog Pins (A0-A5):** It can be used to read analog signals to the board.
11) **Microcontroller (ATMega328):** The processing and logical unit of the board.
12) **ICSP Pin:** A programming header on the board also called SPI.
13) **Power Indicator LED:** It indicates the power status of the board.
14) **RX and TX LEDs:** It receive (RX) and transmit (TX) LEDs, blink when sending or receiving serial data respectively.
15) **Digital I/O Pins:** 14 pins capable of reading and outputting digital signals; 6 of these pins are also capable of PWM.
16) **AREF Pins:** It can be used to set an external reference voltage as the upper limit for the analog pins.
17) **Reset button:** It can be used to reset the board.

5.1.2 Programming of Arduino

1. Download & install the Arduino environment (IDE)

If you just got your Arduino Uno board, you'll first have to install the Arduino IDE (Integrated Development Environment) on another computer. The code is typed into the IDE and sent to the Arduino via a USB cable. Visit arduino.cc to download the most recent Arduino IDE version for your computer. There are different versions for Mac, Windows and Linux OS.

- At the download page, click on the "Installer" option for the easiest installation then

- Save the .exe file to your disk drive.
- Open the .exe file.
- Click the button to agree to the licensing agreement.
- Decide which components to put in, then click “Next”.
- Select which folder to put in the program to, then click “Install”.
- Wait for the program to complete installing, then click “Close”.

2. Launch the Arduino IDE

After your Arduino IDE software is downloaded, unzip the folder. To do so, double-click on the Arduino shortcut on your Desktop. The IDE will open up and you’ll see the code editor.

3. If needed, install the drivers

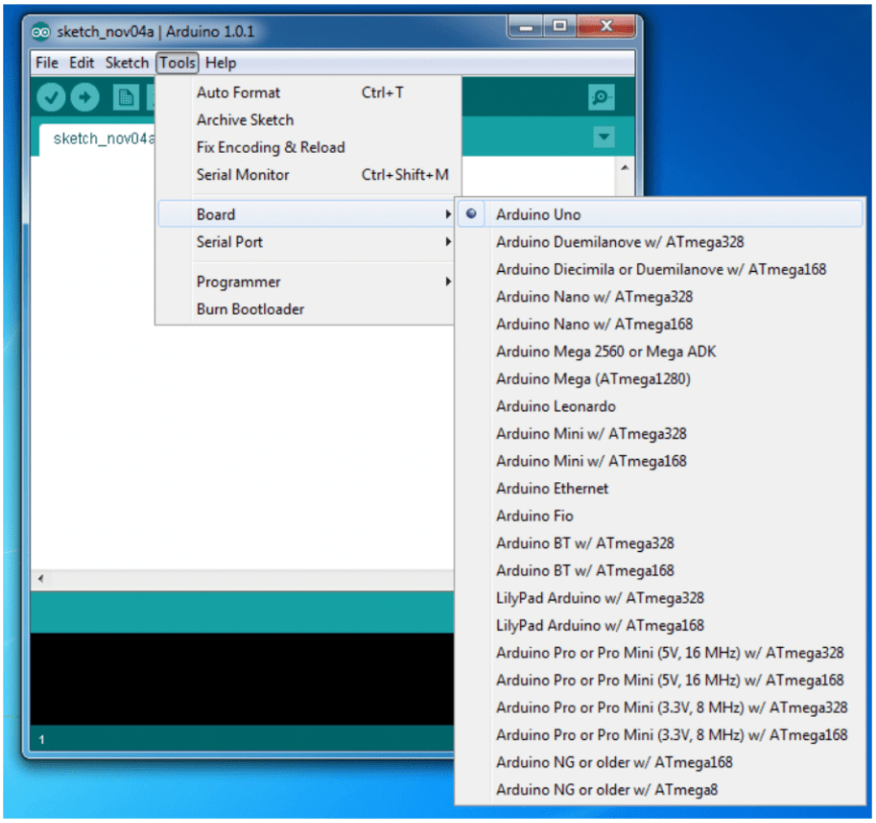
If you used the Installer, it'll install drivers automatically as soon as you connect your board.

4. Connect the board to your computer via the USB cable

To power up your board, connect your Arduino board with the pc via USB cable. The green color power LED should glow on the board.

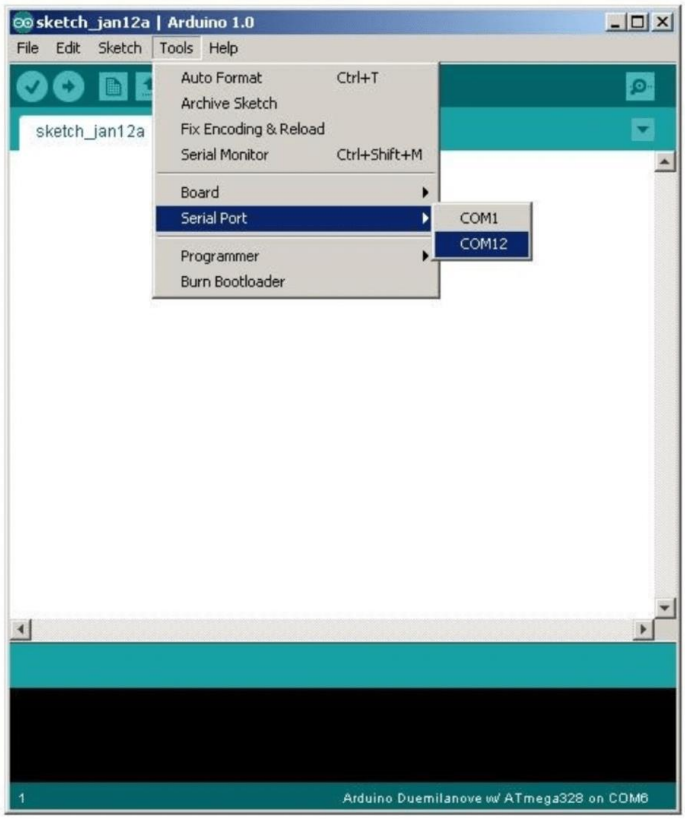
5. Select your board

Next, make sure the software is ready up for your particular Arduino board. Go to the “Tools” computer menu from the menu bar. Select the “Board” option and another menu will appear, where you'll select your Arduino model from the list.



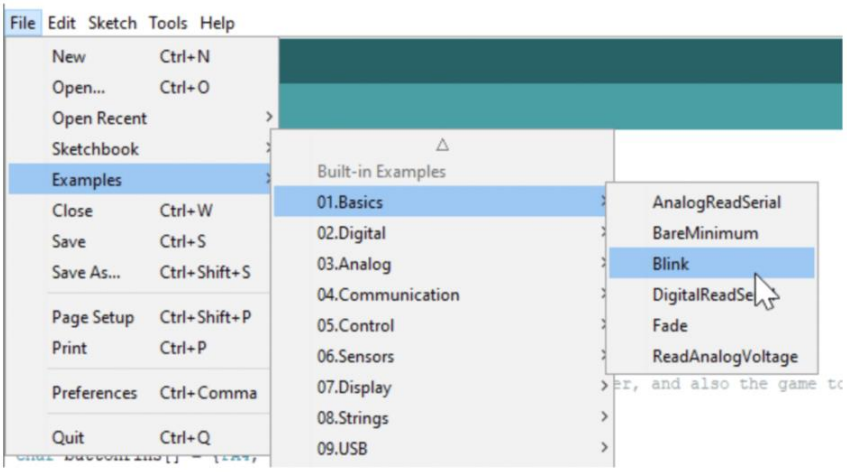
6. Select your serial port

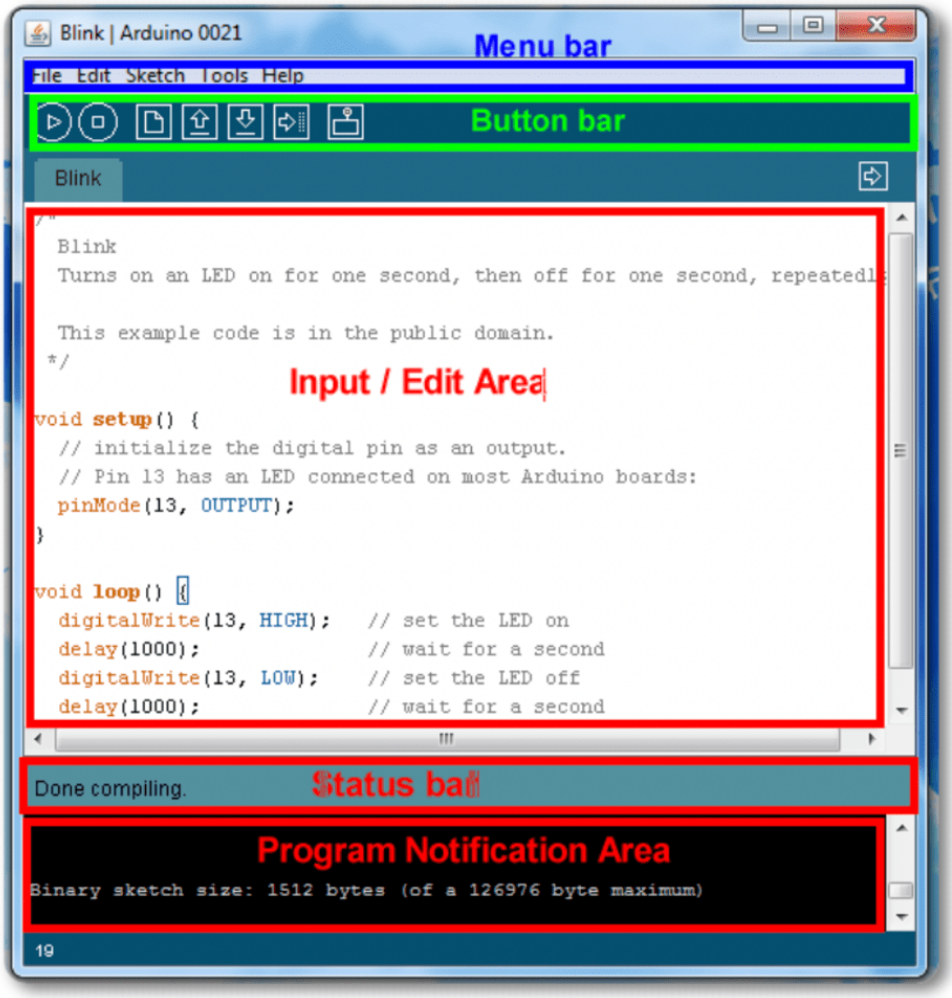
Select the serial device of the Arduino board. Go to Tools, and then the serial port menu. You might see COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out which port your Arduino board is connected to, disconnect your Arduino board and re-open the menu. The entry that disappears should be the Arduino board. Reconnect the board and choose that serial port.



7. Open the blink example

We'll start with the LED Blink example that comes with the Arduino IDE. Just go to File->Examples->Basics->Blink.



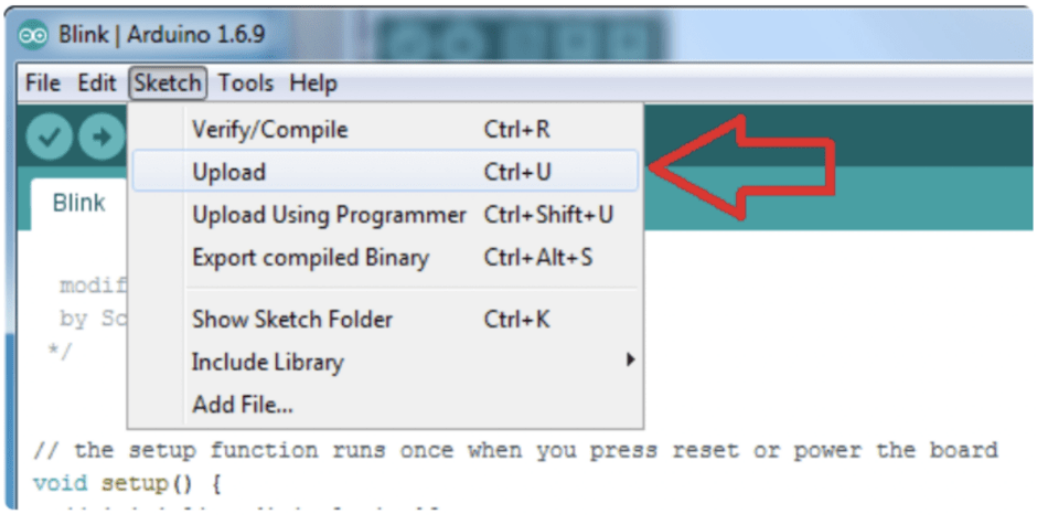


Here are the few things to keep in mind while writing the code:

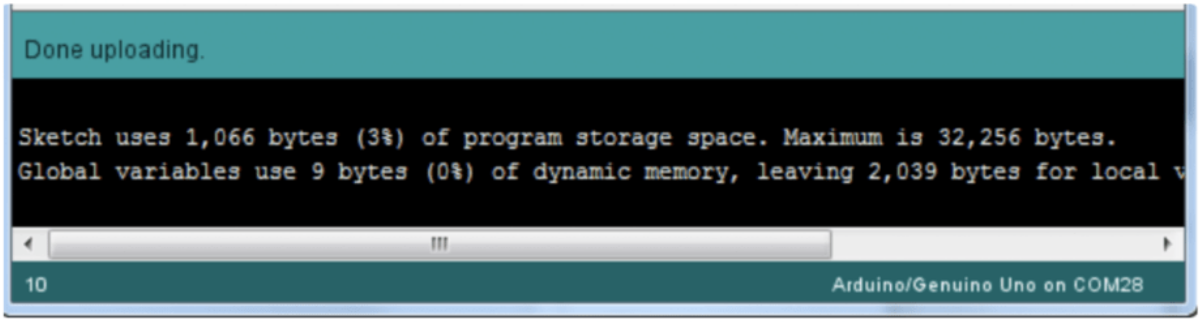
- Code is case sensitive
- All the statements must end with a semicolon
- Comments follow a // or begin with /* and end with */
- Void loop() and void setup() are two mandatory functions. The setup section of the code is simply run once when the Arduino board is first turned on or reset. Once the setup is complete, the loop runs over and over. It keeps on running until the board continues to stay powered.
- The status bar shows that the program is compiled or uploaded.
- Program notification area shows error(s) within the code if any.

8. Upload the program

Now it is time to upload your first sketch(code). Confirm the Arduino is plugged in, and the green light is on - therefore the correct board and port is chosen. Select Upload from the Sketch drop-down menu.



After a few seconds, you will get this screen, with the message "Done uploading."



5.1.3 Applications of Arduino

1) **Home Automation:** Arduino boards combined with sensors, actuators and wireless connectivity allows home owners to control various aspects of their living spaces with ease. It can be used in smart lighting, temperature control, automated security systems, energy management.
2) **Robotics:** With Arduino, you can integrate sensors, motors and controllers to create autonomous machines capable of performing specific tasks. From simple line-following robots to complex humanoid models, Arduino provides a robust foundation for learning and experimenting with robotics concepts.
3) **Wearable Technology:** Wearable devices such as smartwatches, fitness trackers and garments embedded with interactive elements can be created with the help of Arduino due to its low compact size, low power consumption and compatibility with various sensors and displays.
4) **Educational Teaching:** Arduino's accessibility and affordability makes it an excellent educational tool for teaching electronics, programming and problem-solving skills. Arduino-based projects provide hands-on learning experiences, encouraging students to explore technology and to develop a deep understanding of concepts.
5) **Automated Gardening:** Automated gardening is an increasingly popular way to grow plants with Arduino. This can be done by using digital sensors and actuators that interact with a user's

environment, such as temperature or water levels. This technology could be used for plant monitoring and controlling irrigation systems in greenhouses or large fields of crops.

6) **Smart City Projects:** Arduino is a powerful tool for enabling the development of smart city projects, such as automated parking systems, traffic control solutions or even energy-efficient lighting networks.
7) **Drones:** Arduino's ease of use allows users to customize each component of drone according to their exact needs and it also provides support for wireless communication protocols like Wi-Fi or Bluetooth. Custom drones for specific applications such as agriculture monitoring, search and rescue operations, etc. can be built.

5.2 ARCHITECTURE, PROGRAMMING & APPLICATIONS OF RASPBERRY PI

Introduction:

Raspberry Pi, developed by Raspberry Pi Foundation in association with Broadcom, is a series of small single-board computers and perhaps the most inspiring computer available today.

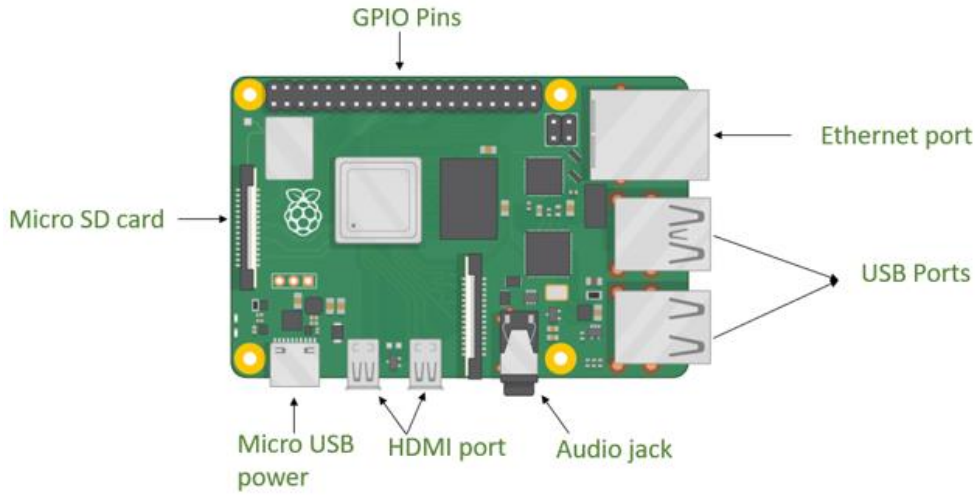
It is originally designed for education, inspired by the 1981 BBC Micro. Creator Eben Upton's goal was to create a low-cost device that would improve programming skills and hardware understanding at the pre-university level. The Raspberry Pi is slower than a modern laptop or desktop but is still a complete Linux computer and can provide all the expected abilities that implies, at a low-power consumption level.

In 2012, the company launched the Raspberry Pi and the current generations of regular Raspberry Pi boards are Zero, 1, 2, 3, and 4. Latest one is 4B. Raspberry Pi OS (previously called Raspbian) is official supported operating system.

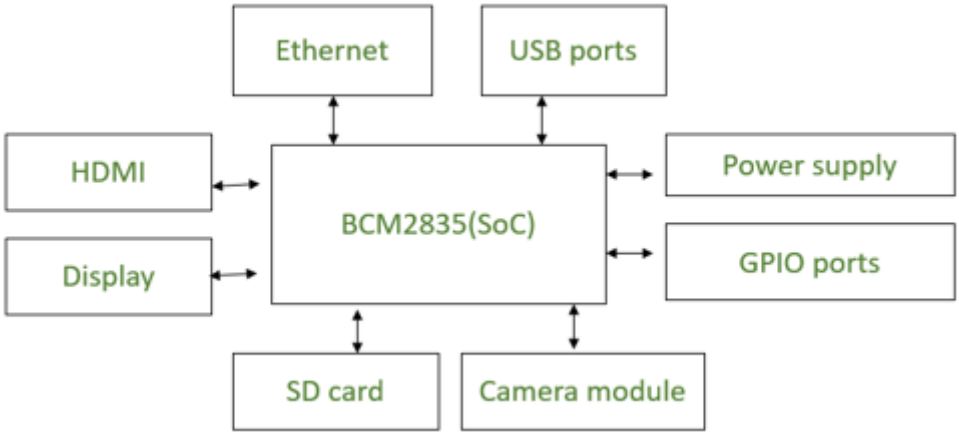
5.2.1 Architecture of Raspberry Pi

Raspberry Pi is a small single-board computer (SBC). It is a credit card-sized computer that can be plugged into a monitor. It acts as a minicomputer by connecting the keyboard, mouse and display. Raspberry Pi has an ARM processor and 512MB of RAM.

The following diagram shows the architecture of Raspberry Pi:



The following diagram shows some main blocks of Raspberry Pi:



Raspberry Pi mainly consists of the following blocks:

1) **Processor:** Raspberry Pi uses Broadcom BCM2835 system on chip which is an ARM processor and Video core Graphics Processing Unit (GPU). It is the heart of the Raspberry Pi which controls the operations of all the connected devices and handles all the required computations.
2) **HDMI:** High Definition Multimedia Interface is used for transmitting video or digital audio data to a computer monitor or to digital TV. This HDMI port helps Raspberry Pi to connect its signals to any digital device such as a monitor digital TV or display through an HDMI cable.
3) **GPIO ports:** General Purpose Input Output ports are available on Raspberry Pi which allows the user to interface various I/P devices.
4) **Audio output:** An audio connector is available for connecting audio output devices such as headphones and speakers.
5) **USB ports:** This is a common port available for various peripherals such as a mouse, keyboard, or any other I/P device. With the help of a USB port, the system can be expanded by connecting more peripherals.
6) **SD card:** The SD card slot is available on Raspberry Pi. An SD card with an operating system installed is required for booting the device.
7) **Ethernet:** The ethernet connector allows access to the wired network, it is available only on the model B of Raspberry Pi.
8) **Power supply:** A micro USB power connector is available onto which a 5V power supply can be connected.
9) **Camera module:** Camera Serial Interface (CSI) connects the Broadcom processor to the Pi camera.
10) **Display:** Display Serial Interface (DSI) is used for connecting LCD to Raspberry Pi using 15 15-pin ribbon cables. DSI provides a high-resolution display interface that is specifically used for sending video data.

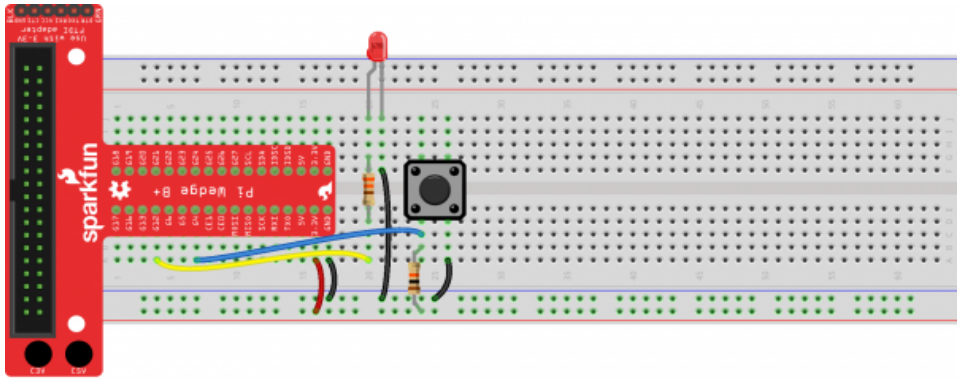
5.2.2 Programming of Raspberry Pi

Downloading audio clips and playing them on a Raspberry Pi is quite simple. We will use the command line to download a .wav file, adjust the audio and test playing the file. Then, we'll write a Python script to play that file whenever we press a button!

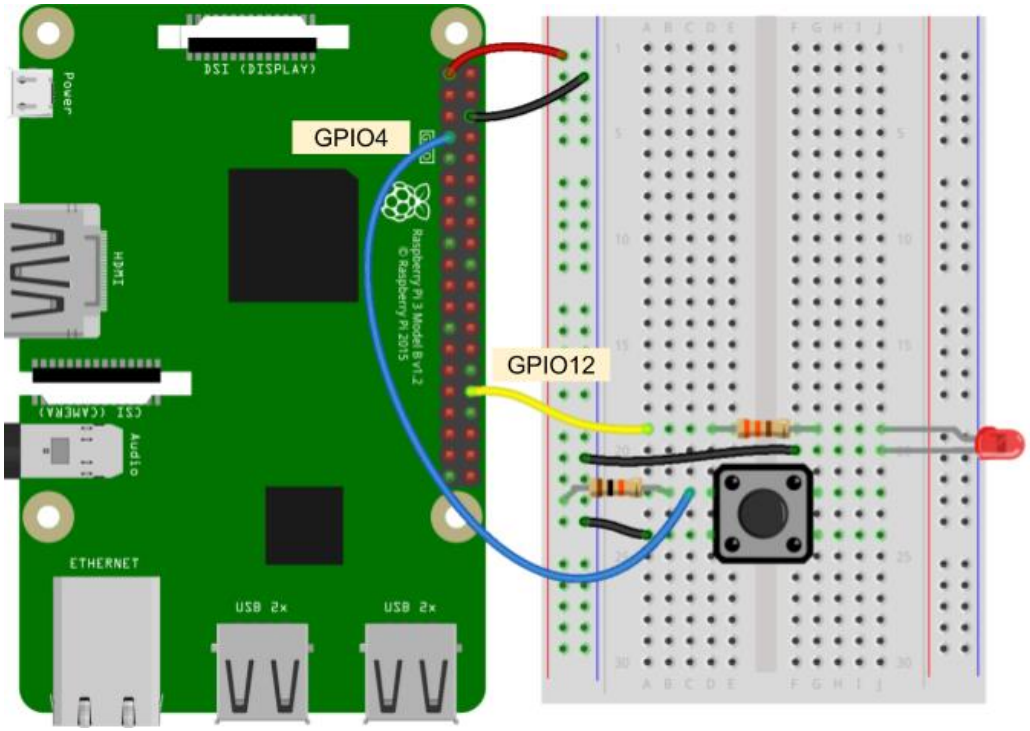
Hardware Connections

- Connect GPIO12 (pin 32) to the 330Ω resistor, and the resistor to the LED
- Connect GPIO4 (pin 7) to the button
- Make the power (3.3 V) and ground (GND) connections as shown in the Fritzing diagram

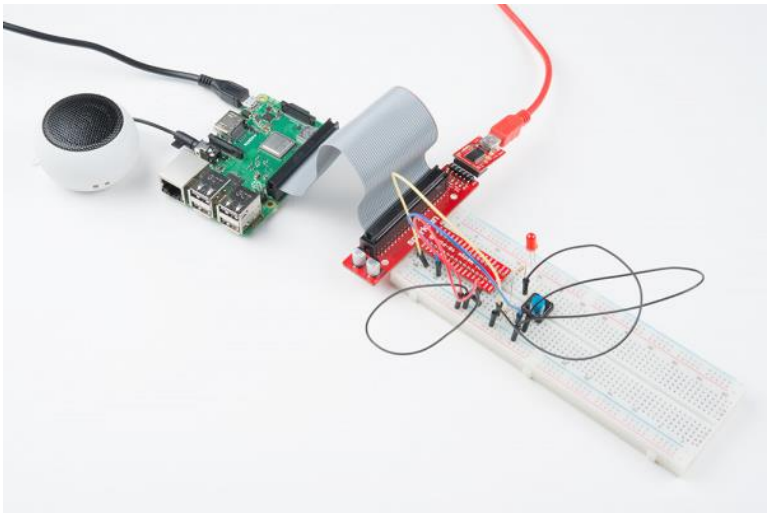
Connecting through a Pi Wedge:



Connecting directly to the Raspberry Pi:



You will also need to plug an external speaker (or a set of headphones) into the Pi's headphone jack.



Configure Audio

Before we write code, we need to configure the audio from the command line. Open a terminal (if you are using Raspbian with a desktop).

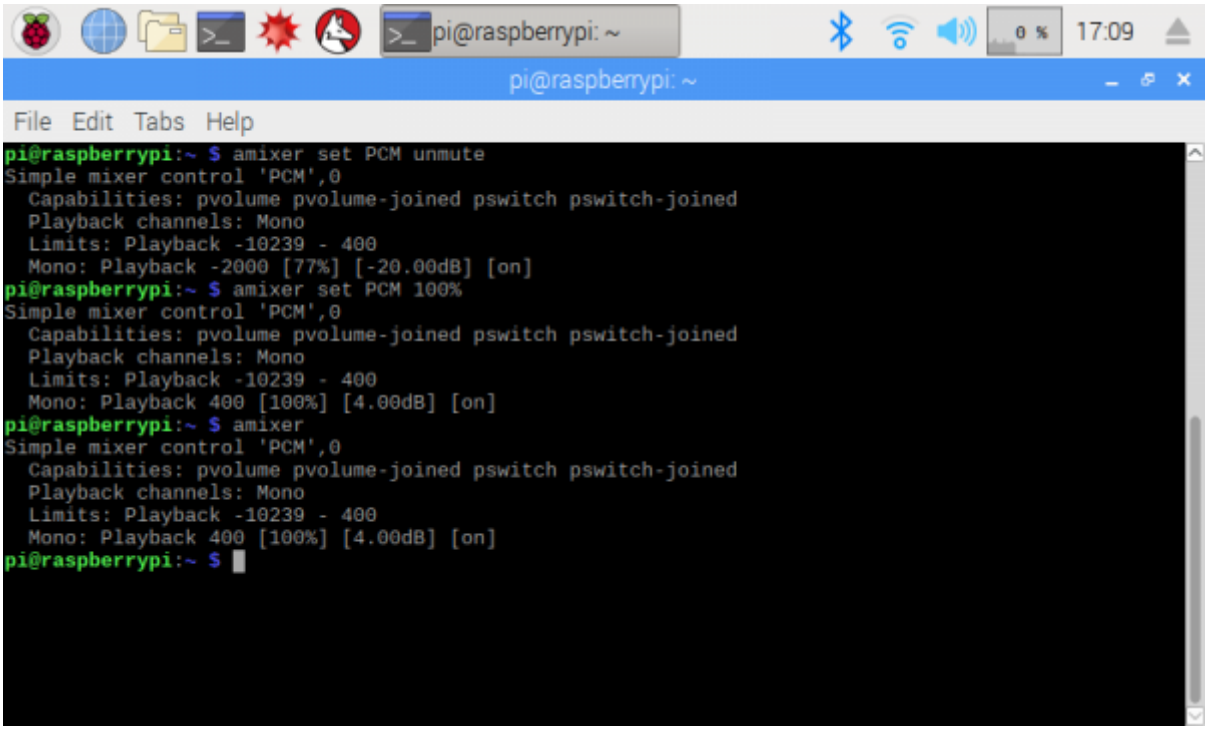
From a terminal, enter the following commands:

```
COPY CODEamixer set PCM unmute
amixer set PCM 100%
```

Verify that your audio is on and up by entering the command:

```
COPY CODEamixer
```

At the end of the printout, you should see `Mono: Playback 400 [100%] [4.00dB] [on]`.



Download a free sound clip

```
COPY CODEwget http://www.pacdv.com/sounds/people_sound_effects/applause-1.wav
```

Test playing this sound with:

```
COPY CODEaplay applause-1.wav
```

Code: Push Button, Get Sound

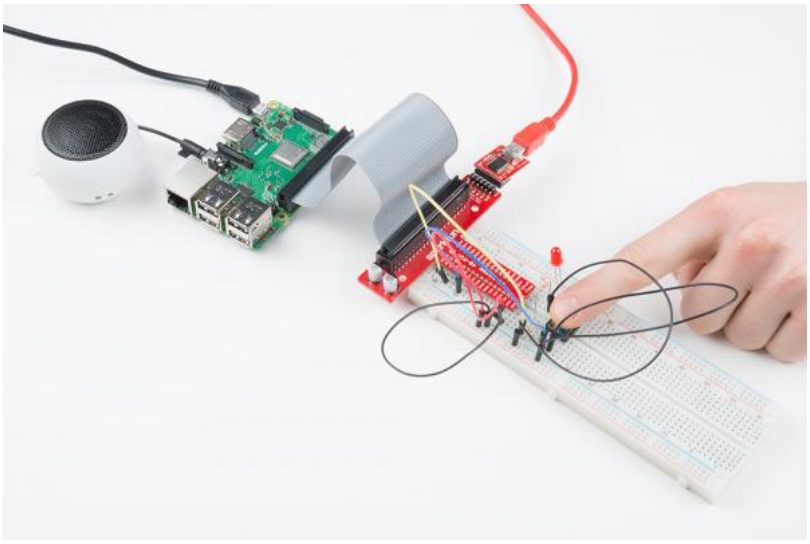
Depending on your version of Raspbian, you may or may not have to install the pygame package (e.g. Raspbian Lite does not come with some Python packages pre-installed). In a terminal, enter the following:

```
COPY CODEsudo apt-get update
sudo apt-get install python3-pygame
```

In a new file, enter the following code:

```
COPY CODEimport time
import RPi.GPIO as GPIO
from pygame import mixer

# Pins definitions
btn_pin = 4

# Set up pins
GPIO.setmode(GPIO.BCM)
GPIO.setup(btn_pin, GPIO.IN)

# Initialize pygame mixer
mixer.init()

# Remember the current and previous button states
current_state = True
prev_state = True

# Load the sounds
sound = mixer.Sound('applause-1.wav')

# If button is pushed, light up LED
try:
    while True:
        current_state = GPIO.input(btn_pin)
        if (current_state == False) and (prev_state == True):
            sound.play()
        prev_state = current_state

# When you press ctrl+c, this will be called
finally:
    GPIO.cleanup()
```

Save the file (e.g. *applause.py*), and start the program with `python applause.py`. Push the button, and you should hear some congratulatory sounds!

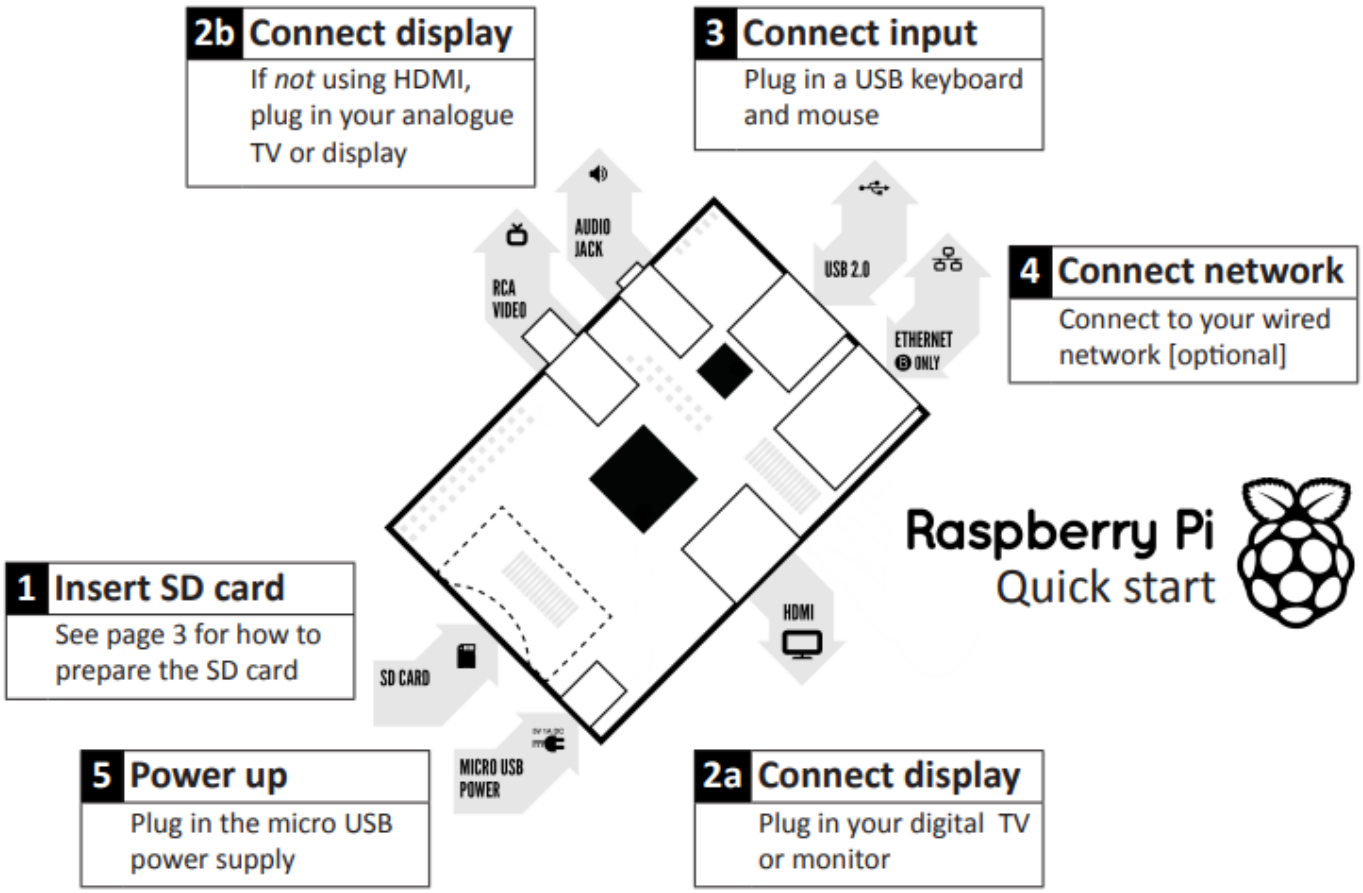


5.2.3 Applications of Raspberry Pi

1) Desktop PC
2) Wireless print server
3) Media Usage
4) Game Servers
5) Retro Gaming Machine
6) Robot Controller
7) Stop Motion Camera
8) Time-lapse Camera
9) FM Radio Station
10) Web Servers
11) Playing games
12) Browsing the internet
13) Word processing
14) Spreadsheets
15) Editing photos
16) Paying bills online
17) Managing your accounts

5.3 RASPBERRY PI SETUP & CONFIGURATION

Setup:



Sr No.	Items	Minimum recommended specification & notes
1	**SD card**	• Minimum size 4Gb; class 4 (the class indicates how fast the card is). • We recommend using branded SD cards as they are more reliable.
2a	**HDMI to HDMI / DVI lead**	• HDMI to HDMI lead (for HD TVs and monitors with HDMI input). OR HDMI to DVI lead (for monitors with DVI input). • Leads and adapters are available for few pounds -- there is no need to buy expensive ones!
2b	**RCA video lead**	• A standard RCA composite video lead to connect to your analogue display if you are not using the HDMI output.
3	**Keyboard and mouse**	• Any standard USB keyboard and mouse should work. • Keyboards or mice that take a lot of power from the USB ports, however, may need a powered USB hub. This may include some wireless devices.

4	**Ethernet (network) cable [optional]**	• Networking is optional, although it makes updating and getting new software for your Raspberry Pi much easier.
5	**Power adapter**	• A good quality, micro USB power supply that can provide at least 700mA at 5V is essential. • Many mobile phone chargers are suitable—check the label on the plug. • If your supply provides less than 5V then your Raspberry Pi may not work at all, or it may behave erratically. Be wary of very cheap chargers: some are not what they claim to be. • It does not matter if your supply is rated at more than 700mA.
6	**Audio lead [optional]**	• If you are using HDMI then you will get digital audio via this. • If you are using the analogue RCA connection, stereo audio is available from the 3.5mm jack next to the RCA connector

HDMI connector

HDMI to DVI lead

RCA composite video connector

Preparing SD Card for Raspberry Pi

The SD card contains the Raspberry Pi's operating system (the OS is the software that makes it work, like Windows on a PC or OSX on a Mac). This is very different from most computers and it is what many people find the most daunting part of setting up their Raspberry Pi.

The following instructions are for Windows users. Linux and Mac users can find instructions at www.raspberrypi.org/downloads

1. Download the Raspberry Pi operating system

The recommended OS is called Raspbian. It can be downloaded using following link: http://downloads.raspberrypi.org/images/raspbian/2012-12-16-wheezy-raspbian/2012-12-16-wheezy-raspbian.zip

2. Unzip the file that you just downloaded

a) Right click on the file and choose "Extract all".

b) Follow the instructions—you will end up with a file ending in .img

This .img file can only be written to your SD card by special disk imaging software

3. Download the Win32DiskImager software

a) Download win32diskimager-binary.zip (currently version 0.6) from:

https://launchpad.net/win32-image-writer/+download

b) Unzip it in the same way you did the Raspbian .zip file

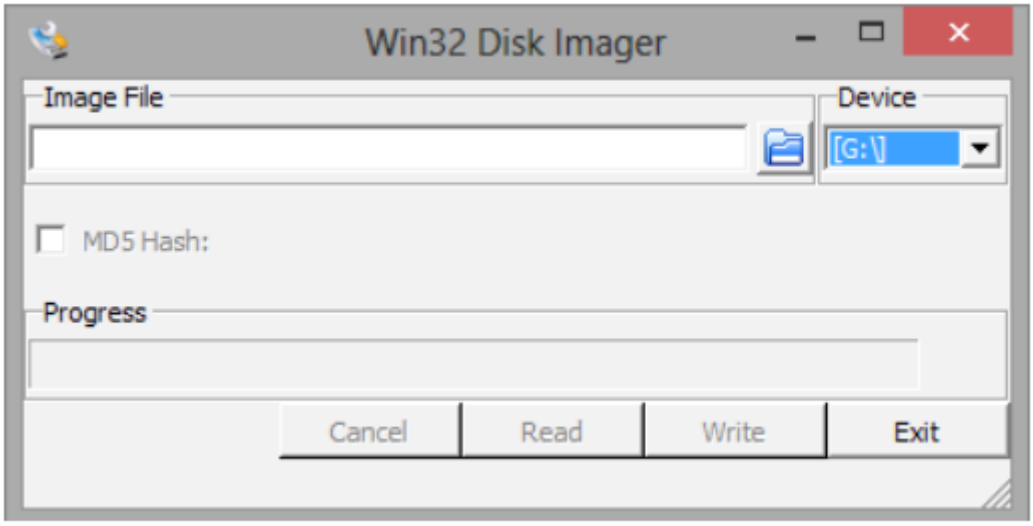
c) You now have a new folder called win32diskimager-binary

You are now ready to write the Raspbian image to your SD card.

4. Writing Raspbian to the SD card

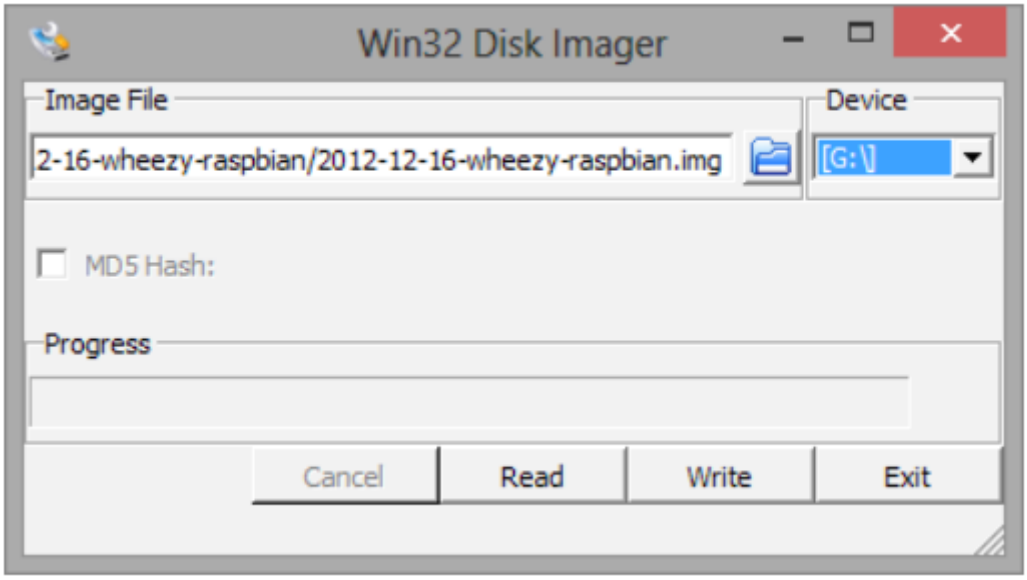
a) Plug your SD card into your PC

b) In the folder you made in step 3(b), run the file named Win32DiskImager.exe (in Windows Vista, 7 and 8 we recommend that you right-click this file and choose "Run as administrator"). You will see something like this:



c) If the SD card (Device) you are using isn't found automatically then click on the drop down box and select it

d) In the Image File box, choose the Raspbian .img file that you downloaded



e) Click Write

f) After a few minutes you will have an SD card that you can use in your Raspberry Pi

5. Booting your Raspberry Pi for the first time

a) Follow the Quick start guide on page 1

b) On first boot you will come to the Raspi-config window

c) Change settings such as timezone and locale if you want

d) Finally, select the second choice: expand_rootfs and say 'yes' to a reboot

e) The Raspberry Pi will reboot and you will see raspberry pi login:

f) Type:

pi

g) You will be asked for your Password

h) Type:

raspberry

i) You will then see the prompt:

pi@raspberry ~ $

j) Start the desktop by typing:

startx

k) You will find yourself in a familiar-but-different desktop environment.

l) Experiment, explore and have fun!

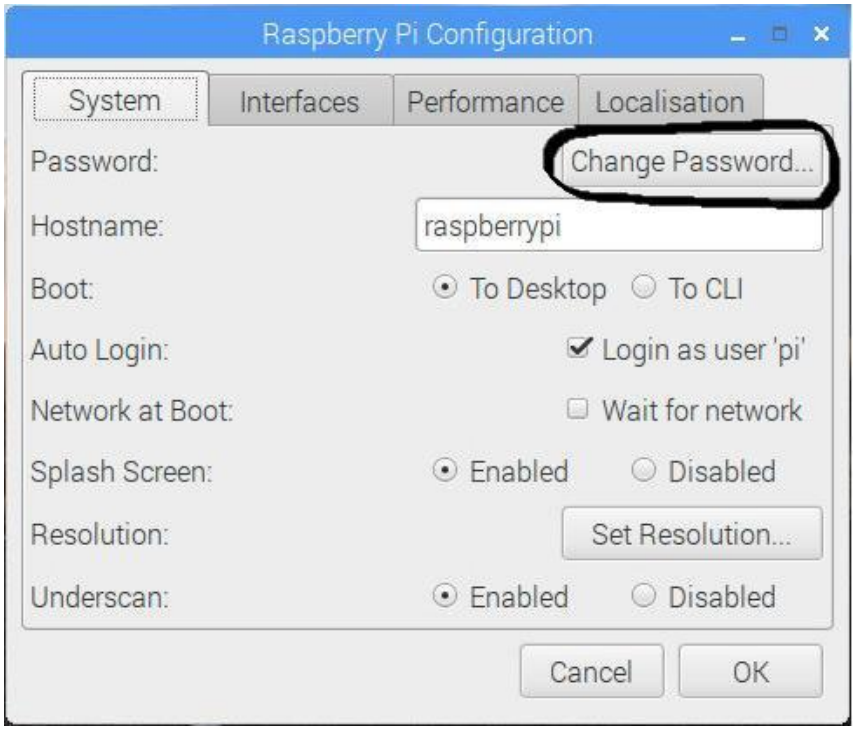
Configuration:

- Raspbian Configuration:

For configuring Raspberry Pi in Raspbian, we are using Raspbian with PIXEL desktop. It is one of the best ways to get Raspbian started with the Raspberry Pi. Once we finish booting, we will be in the PIXEL desktop environment.

Now to open the menu, you need to click the button that has the Raspberry Pi logo on it. This button will be in the top left. After clicking the button, choose Raspberry Pi configuration from the preferences.

Following is the configuration tool in PIXEL desktop −



By default, the configuration tool opens to its system tab which has the following options −

1) **Change Password:** The default password is raspberry. You can change it by clicking the change password button.
2) **Change the hostname:** The default name is raspberry pi. You can also change it to the name, which you want to use on the network.
3) **Boot:** You can choose from the two options and control whether Raspberry Pi boots into the desktop or CLI i.e., command line interface.
4) **Auto Login:** With the help of this option, you can set whether the user should automatically log in or not.
5) **Network at Boot:** By choosing this option, you can set whether the pi user is automatically logged in or not.
6) **Splash screen:** You can enable or disable it. On enabling, it will display the graphical splash screen that shows when Raspberry Pi is booting.

7) **Resolution:** With the help of this option, you can configure the resolution of your screen.
8) **Underscan:** There are two options, enable or disable. It is used to change the size of the displayed screen image to optimally fill the screen. If you see a black border around the screen, you should disable the underscan. Whereas, you should enable the underscan, if your desktop does not fit your screen.

There are three other tabs namely Interfaces, Performance and Localization. The job of interface tab is to enable or disable various connection options on your Raspberry Pi.

You can enable the Pi camera from the interface tab. You can also set up a secure connection between computers by using SSH (short for Secure Shell) option.

If you want to remote access your Pi with a graphical interface then, you can enable RealVNC software from this tab. SPI, I2C, Serial, 1-wire and Remote GPIO are some other interfaces you can use.

There is another tab called Performance, which will give you access to the options for overclocking and changing the GPU memory.

The localization tab, as the name implies, enable us to set:

1) The character set used in our language.
2) Our time zone.
3) The keyboard setup as per our choice.
4) Our Wi-Fi country.

- Configure Wi-Fi
- Configure Bluetooth Devices
- Data Partition Setup:

As we know that data partition is that area on your memory card (SD or MicroSD) which can be shared by various distributions. One of the best examples of use of a data partition is transferring the files between distributions.

The data partition has the label data.

You can use this labelled data to make a directory point to it as follows −

Step 1: First, you need to boot the Raspberry Pi into Raspbian.

Step 2: Now, click the Terminal icon to get to the command line.

Step 3: Next, type the command mkdir shared. It will create a directory named shared.

Step 4: Write the command sudo mount -L data shared. This command will point the directory to the shared partition.

Step 5: Write the command sudo chown $USER: shared. It will set the permission for writing in this shared folder.

Step 6: Now, to go to this shared folder, you need to type the command cd shared.

Once all the files are created in this shared folder, they will be available to all the distributions that have the permission to access the data partition.

5.4 RASPBERRY PI V/S ARDUINO

Parameters	Arduino	Raspberry Pi
Control Unit	The Control Unit of the Arduino is from the ATmega family.	The Control Unit of the Raspberry Pi is from the ARM family.
Basis	Arduino works on the basis of a microcontroller.	Raspberry Pi, on the other hand, works on the basis of a microprocessor.
Use	The Arduino basically helps in controlling all the electrical components that connect to a system's circuit board.	The Raspberry Pi primarily computes data and info for producing valuable outputs. It also controls the various components in any given system on the basis of the outcome (of the computation).
Structure of Hardware and Software	The Arduino boards have a very simple structure of software and hardware.	The Raspberry Pi boards consist of comparatively complex software and hardware architecture.
Type of CPU Architecture	Arduino has an 8-bit architecture.	Raspberry Pi has a 64-bit architecture.
RAM Usage	Arduino makes use of very little RAM of about 2 kB (Kilobytes).	Raspberry Pi always requires more RAM than Arduino of about 1 GB (Gigabytes).
Processing Speed	Arduino clocks 16 MHz (Megahertz) of processing speed in a system.	The Raspberry Pi clocks 1.4 GHz (Gigahertz) of processing speed in a system.
Cost Efficiency	It has a higher cost-efficiency because it is comparatively cheaper.	It has a lower cost-efficiency because it is comparatively more expensive.
I/O Drive Strength	The I/O current drive strength in the case of Arduino is higher.	The I/O current drive strength in the case of Raspberry Pi is lower.
Power Consumption	Arduino consumes power of about 200 MW (Megawatts).	Raspberry Pi consumes about 700 MW.
Software	Arduino boards are programmable using C/C++ languages.	Raspberry Pi supports its own Linux-based operating system Raspberry Pi OS. You can also install the OS you like.
Internet	Arduino does not have internet support. You need additional modules or shields to connect it to the internet.	Raspberry Pi has a built-in Ethernet port and WiFi support.

Current drive strength	Higher current drive strength	Lower current drive strength
Capability	Arduino is generally used to perform single (and simple) tasks repeatedly.	Raspberry Pi can perform multiple tasks simultaneously.
Wireless connectivity	Arduino does not support Bluetooth or WiFi.	Raspberry Pi supports Bluetooth and WiFi.
Applications	Traffic light countdown timer, Parking lot counter, Weighing machines, etc.	Robot controller, Game servers, Stop motion cameras, etc.