# UNIT III – IOT PHYSICAL DEVICES AND ENDPOINTS

Introduction to Arduino and Raspberry Pi- Installation, Interfaces (serial, SPI, I2C), Programming – Python program with Raspberry PI with focus on interfacing external gadgets, controlling output, and reading input from pins.

## INTRODUCTION TO ARDUINO AND RASPBERRY PI

### ARDUINO

Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's widely used for building digital devices and interactive objects that can sense and control the physical world. The Arduino platform consists of a physical programmable circuit board (microcontroller) and a development environment (IDE) for writing software.

- Arduino is an open-source hardware, and software platform used to design and build electronic devices.
- It designs and manufactures microcontroller kits and single-board interfaces for building electronics projects.
- Arduino UNO is a microcontroller version of ATmega328p, it has 14 digital io pins, 6 of which can be used for PWM output, 6 analog input pins, 32 kb flash memory, 2 kb static memory, 1 kb live edge socket RAM, type b USB connection, ICSP connector and reset button.
- Only through the USB data connection, you can power the computer, download the program and control the hardware by the data communication software.
- Arduino has its own language based on classic C style, and the hardware Layer is written in high level languages. So, in terms of programming Arduino seems to be very simple, just needs to know one language like java or C
- When processing analog signals and digital signals, it is often necessary to perform the proportional conversion, such as the conversion of angle to the digital sensor

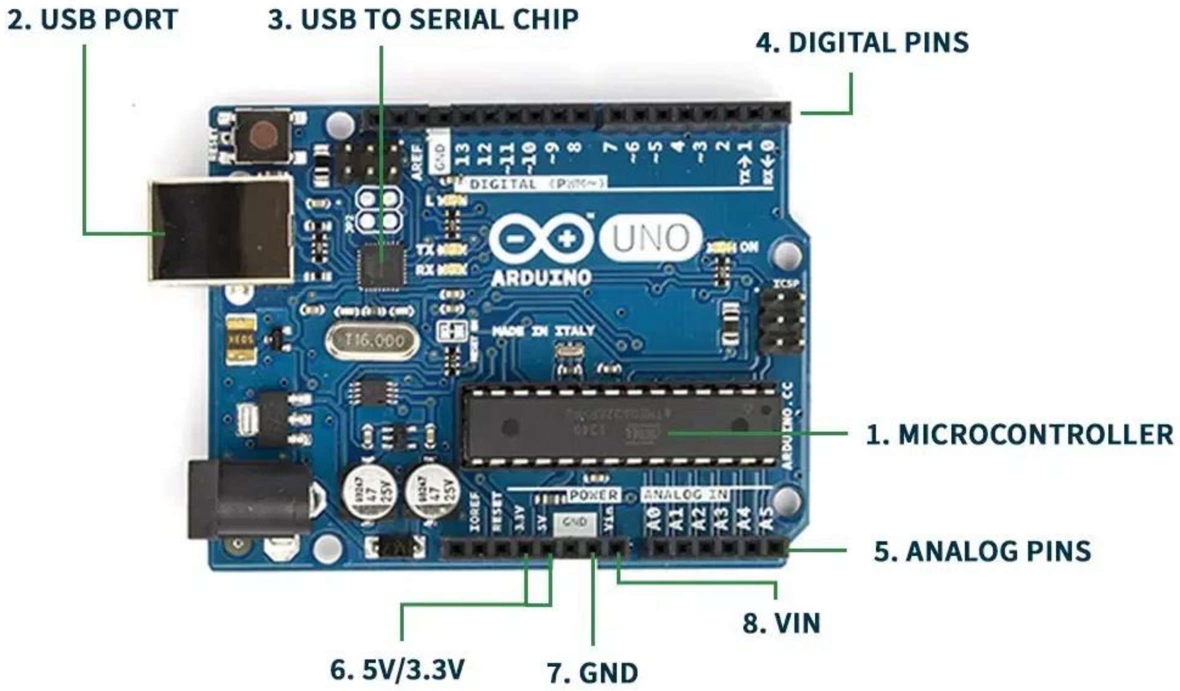
### Arduino Features

- **Microcontroller**: Serves as the core processing unit, handling simple commands in real time.
- **GPIO Pins**: General-purpose input/output pins for connecting to external devices.
- **Power Supply**: Provides electricity to the board.
- **USB**: Connects to a computer for programming and power.
- **Reset Button**: Restarts the board.

- **PWM**: Modulates the width of pulses to simulate varying output levels and can be used to control LED brightness or motor speed.
- **Serial Communication**: Sends and receives bits sequentially.
- **Parallel Communication**: Sends multiple bits simultaneously.
- **Interrupts**: Responds to external events.
- **Timers**: Precisely controls timing.
- **ADC**: Converts analog signals to digital.
- **Memory**: Stores programs and data.
- **Crystal Oscillator**: Provides stable timing for the board.

## Applications of Arduino

- Arduino is used in various applications, including:
- Robotics: Building autonomous robots.
- Home Automation: Creating smart home devices.
- Wearable Tech: Developing fitness trackers and health monitoring devices.
- Prototyping: Rapid prototyping of electronic circuits.



- **Microcontroller:** The microcontroller used on the Arduino board is essentially used for controlling all major operations. The microcontroller is used to coordinate the input taken and execute the code written in a high-level language.
- **Analog Reference pin:** Analog pins are used for general purposes like supporting 10-bit analog-to-digital conversion (ADC) which is performed using analog the Read() function. Analog pins are particularly helpful since they can store 0-255 bits which is not possible using digital pins.
- **Digital Pins: Di**gital pins are used for general purposes like taking input or generating output. The commands that are used for setting the modes of the pins are pinMode(), digitalRead(), and digitalWrite() commands.
- **Reset Button:** The reset button on the Arduino board is used for setting all the components of Arduino to their default values. In case you want to stop the Arduino in between you can use this reset button.
- **Power and Ground Pins:** As the name suggests, power and ground pins are used to supply the power needed for driving the Arduino board. The ground pins are usually 0V to set a reference level for the circuit.
- **USB (universal serial bus):** The Arduino needs certain protocols for communication purposes and the universal serial bus is used for this purpose. It helps to connect Arduino, microcontrollers with other raspberry pies.

Electronic Signals

Two types of signals that are used for communication:

- Analog Signal: Analog signals can take any value in a given continuous range of values. Generally, analog signals used in Arduino are around 0V to 5V. The analog pins can take data up to 8-bit resolution therefore, they are used for taking large values as input in the Arduino. These signals carry data in a very accurate form without many errors.
- Digital Signal: Digital signals can only take discrete values which are, high('1') and low('0'). These signals are usually used to Arduino on or off which requires only two values. The collection of two values (0 and 1) can be used to generate a sequence known as the binary sequence which is a collection of zeroes and ones. This is how data is transmitted without much memory requirement but this can lead to certain errors like quantization errors.

**Coding Screen**

If you open the coding screen of your IDE, you will realize that it is divided into two sections namely, **setup()** and **loop()**. The setup segment is the first block and is implemented first for preparing the necessary environment needed for running other commands. This coding screen is shown below:

**Setup**

Setup contains the very beginning section of the code that must be executed first. The pin modes, libraries, variables, etc., are included in the setup section so that no problem occurs when the remaining code runs. It is executed only once during the uploading of the program and after resetting or powering up the Arduino board. Zero setup () resides at the top of each sketch. When the program runs after completion, it heads towards the setup section to initialize the setup and include all the necessary libraries all at once.

**Loop**

The loop contains statements that are executed repeatedly. Unlike, the setup section there is no restriction on running this code once, it can run multiple times according to the value of variables.

**Time**

The basic unit of measuring time in Arduino programming is a millisecond.

***1 sec = 1000 milliseconds***

**PinMode ( )**

The pinMode() function assigns a specific PIN as either INPUT or OUTPUT.

**The Syntax is:**

*pinMode (pin, mode)*

**Pin:** It is used to specify the PIN which depends on the project requirements.

**Mode:** Depending on whether the pin is taking INPUT or OUTPUT, it specifies the pin's function.

We want to take input from the PIN 13 and then, **Code:**

*pinMode (13, INPUT);*

**digitalWrite( )**

The digitalWrite( ) function is used to decide the value of the pin. It can be set as either of the two values, HIGH or LOW.

**HIGH:** For a board that is supplied with a maximum of 1V, it results in a 5V value whereas on a board with other values like 6V, it updates the value to 6V.

**LOW:** It sets the pin to the ground by setting a reference of 0V.

If no pin is set with pinMode as OUTPUT, the LED may light dim.

**The syntax is:**

*digitalWrite( pin, value HIGH/LOW)*

**delay ( )**

The delay () function serves as a tool to halt program execution for a specified duration, measured in milliseconds. We have seen how delay (5000) signifies a stop of 5 seconds.

TYPES OF ARDUINO:

- Arduino Uno (R3)
- Arduino Nano
- Arduino Micro
- Arduino Due
- LilyPad Arduino Board
- Arduino Bluetooth
- RedBoard Arduino Board
- Arduino Mega (R3) Board
- Arduino Leonardo Board
- Arduino Robot
- Arduino Pro Mic
- Arduino Ethernet

**Advantages and Disadvantages of Arduino**

**Advantages of Arduino**

We need to know the reason for selecting Arduino over other devices so let us study some advantages of Arduino.

- Arduino is the best choice for starting your programming journey in electronics. Its easy-to-use interface allows users to build simple projects on their own.
- There is no need for experience or hands-on experience in electronics before starting work on Arduino. Anyone with a genuine interest in Arduino can begin learning through simple tutorials and some guidance. These tutorials are available free of cost for creating some beginner-level and advanced projects.

- Arduinos offer a wide range of options. You can use Arduino alone to create some projects or you can add some extra features by integrating it with other devices like Raspberry Pie.
- Arduino is an open-source tool that can be accessed from different locations and platforms. Due to the inexpensive nature of Arduinos, they can be used on different microcontrollers like Atmel's ATMEGA 16U2 microcontrollers.
- Depending on the need of your project, you can avail of any Arduino that satisfies the needs. These Arduino are available in different designs that offer different size ranges, power, and specifications.

**Disadvantages of Arduino**

Let us see some limitations associated with Arduino:

- Despite being able to communicate with other boards like Raspberry pies and other Arduinos, the communication of Arduino is very restricted since it is installed to use certain basic communication protocols.
- Arduinos have been designed for beginner-level projects as a result they have Limited Memory and Processing Power which limits the projects that can be made using Arduino.
- Due to the lack of excess security in Arduino boards, they can be easily hacked which can result in loss and data leakage.
- When it comes to accuracy, the Arduino board is not the best choice since it lacks the precision needed for analog to digital conversion.
- Arduino responds and coordinates tasks based on the responsiveness of other components due to which it cannot be programmed for real-time applications.

RASPBERRY PI

Raspberry Pi is a small but complete mini processing system that generally has the following configuration.

➢ Credit card size single board computer or a Programmable PC

➢ Developed in U.K. by Raspberry - Pi foundation in 2009

➢ Supported by “University of Cambridge Computer Laboratory & Broadcom”

➢ To promote the study of basic computer science in schools & to develop interest among kids and adults

➢ Has been a revolution in the market with over 3 million units sold.

WHY RASPBERRY PI?

➢ Very Low Cost ($25 – Rs 1550/- for Model A & $35 – Rs 2200/- for Model B/B+)

➢ Great tool for Learning Programming, Computers & Concepts of Embedded Linux, etc

➢ Support for all Age Groups (School Children, College Undergraduates, Professional Developers, Programmers)

➢ Supports & runs Free and Open-Source Linux OS

➢ Consumes less than 5W of Power

➢ Supports Full HD Video Output (1080p), Multiple USB Ports , etc

➢ Fun to learn & explore. You are limited by your imagination.

It also provides a set of general-purpose input/output pins allowing you to control electronic components for physical computing and explore the Internet of Things (IOT).

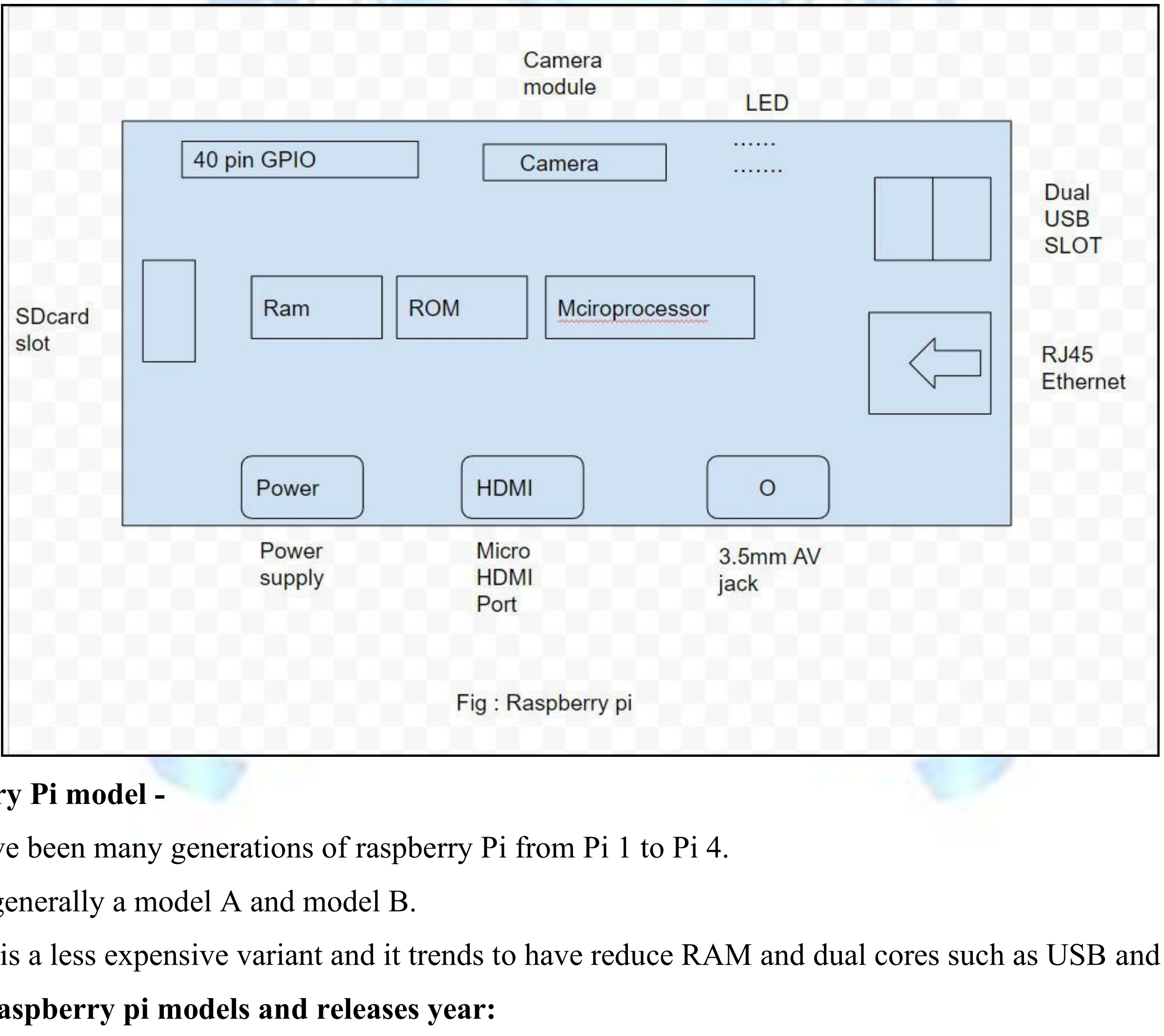


Fig : Raspberry pi

**Raspberry Pi model -**

There have been many generations of raspberry Pi from Pi 1 to Pi 4.

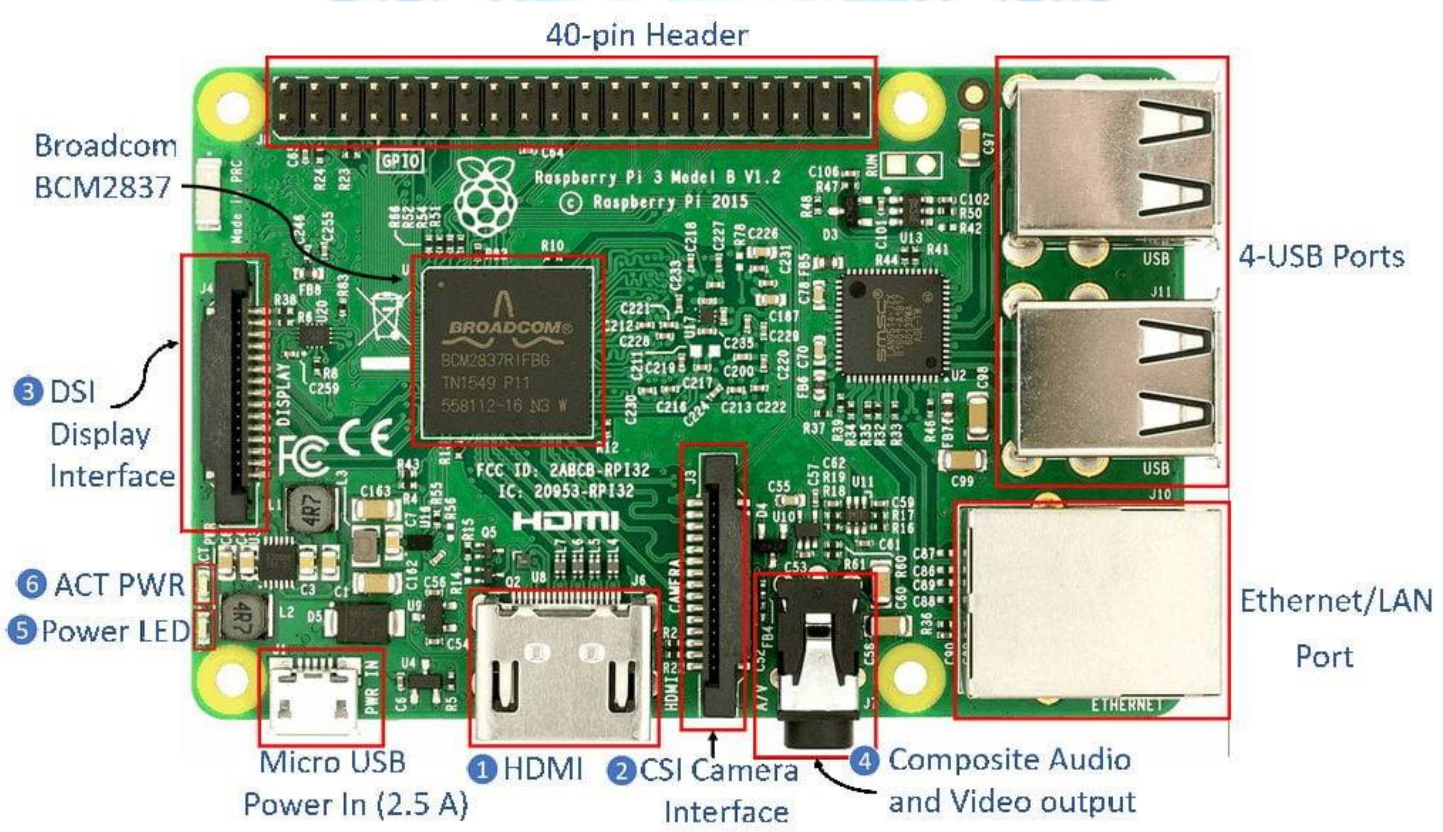
There is generally a model A and model B.

Model A is a less expensive variant and it trends to have reduce RAM and dual cores such as USB and Ethernet.

**List of Raspberry pi models and releases year:**

1. pi 1 model B - 2012

2. pi 1 model A - 2013
3. pi 1 model B+ -2014
4. pi 1 model A+ - 2014
5. Pi 2 Model B - 2015
6. Pi 3 Model B- 2016
7. Pi 3 Model B+ -2018
8. Pi 3 Model A+ -2019
9. Pi 4 Model A - 2019
10. Pi Model B - 2020
11. Pi 400 - 2021



**Raspberry Pi processor**

It has ARM based Broadcom Processor SoC along with on-chip GPU (Graphics Processing Unit).

The CPU speed of Raspberry Pi varies from 700 MHz to 1.2 GHz. Also, it has on-board SDRAM that ranges from 256 MB to 1 GB.

Raspberry Pi also provides on-chip SPI, I2C, I2S and UART modules.

**Some Hardware Components shown above are mention below:**

1. **HDMI (High-Definition Multimedia Interface):** It is used for transmitting uncompressed video or digital audio data to the Computer Monitor, Digital TV, etc. Generally, this HDMI port helps to connect Raspberry Pi to the Digital television.
2. **CSI Camera Interface:** CSI (Camera Serial Interface) interface provides a connection in between Broadcom Processor and Pi camera. This interface provides electrical connections between two devices.
3. **DSI Display Interface:** DSI (Display Serial Interface) Display Interface is used for connecting LCD to the Raspberry Pi using 15-pin ribbon cable. DSI provides fast High-resolution display interface specifically used for sending video data directly from GPU to the LCD display.
4. **Composite Video and Audio Output:** The composite Video and Audio output port carries video along with audio signal to the Audio/Video systems.
5. **Power LED:** It is a RED colored LED which is used for Power indication. This LED will turn ON when Power is connected to the Raspberry Pi. It is connected to 5V directly and will start blinking whenever the supply voltage drops below 4.63V.
6. **ACT PWR:** ACT PWR is Green LED which shows the SD card activity.