MODULE - 5

IOT physical Devices and Endpoints - Arduino UNO

Introduction to Arduino

Arduino is an open-source advancement prototying platform which depends on simple to-utilize equipment and programming.

Arduino can read inputs - such as detecting the power & light, events triggered by a button or a twitter message and can respond into a yield.

The Arduino is a small computer that you can program to read information from the world around you and to send commands to the outside world.

- Arduino is a tiny computer that you can connect to electrical circuits. This makes it easy to read inputs - and control outputs - send a command to the outside.

Why Arduino ?

Arduino is an open source product, software/hardware which is accessible and flexible to customers.

Arduino is flexible because of offering variety of digital and analog pins, SPI and PWM outputs.

Arduino is easy to use, connected to computer via a USB and communicates using serial protocol.

Arduino has growing online community where lots of source code is available for use.

Arduino is Cross-platform, which can work on Windows, Mac or Linux platforms.

Arduino follows simple, clear programming environment as C language.

Which Arduino ?

There are hundreds of "Arduino boards" available in the market serving every kind of purpose. Among all we almost focus on popular Arduino UNO which is used in almost 99% of projects use.

→ Some of the Boards from Arduino family are given below

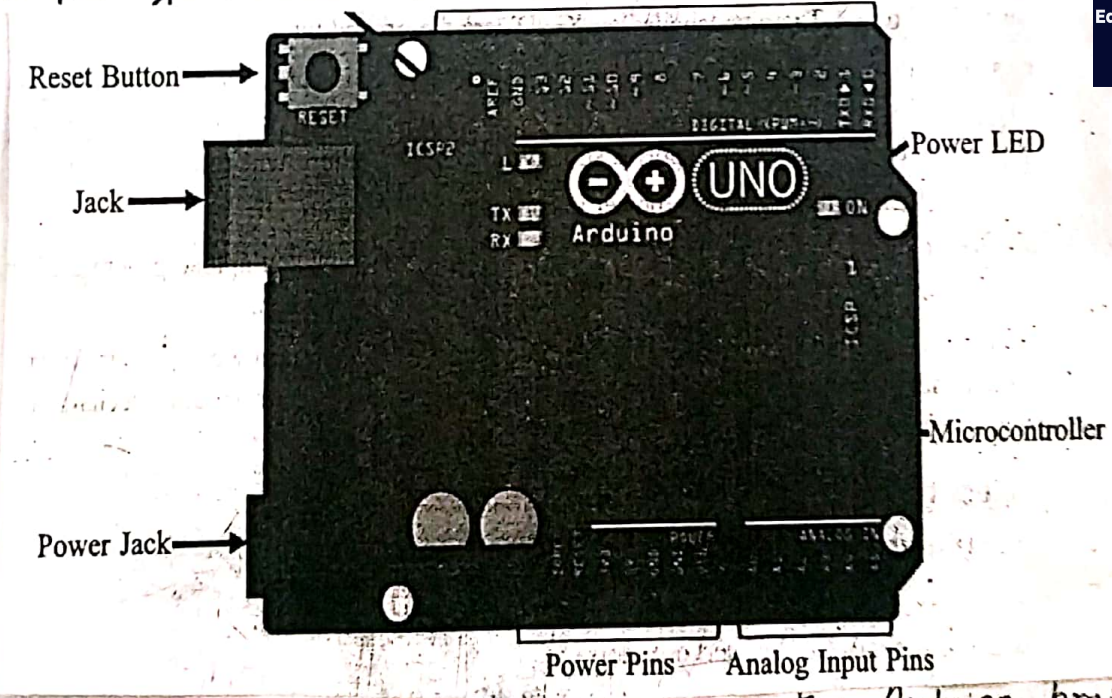
Arduino Mega is a big sister to the UNO with more memory and pins with a different chip the ATmega2560.

Flora is an Arduino compatible from Adafruit which is a round wearable which can be sewed into clothing.

The Arduino MKR1000 is a little like an Arduino Micro but has a more powerful 32-bit ATSAM ARM chip and built-in Wifi.

Arduino Micro is bit smaller with a chip Atmega 32u4 that can act like a keyboard or mouse.

Exploring Arduino UNO Learning Board



* Microcontroller : The ATmega328p is the Arduino brain. Everything on the Arduino board is meant to support this microcontroller.
* Digital pins : Arduino has 14 digital pins, labeled from 0 to 13 that can act as inputs or outputs.

* PWM pins : These are digital pins marked with a ~ (pins 11, 10, 9, 6, 5 and 3). PWM stands for "pulse width modulation" and allows to make digital pins output "fake" varying amounts of voltage.
* TX and RX pins : digital pins 0 and 1. The T stands for "transmit" and the R for "receive".
* LED attached to digital pin 13 : This is useful for an easy debugging of the Arduino sketches.
* Analog pins : The analog pins are labeled from A0 to A5 and are most often used to read analog sensors.
* Power pins : The Aurdino has 3.3V or 5V supply, which is really useful since most components require 3.3V or 5V.
* Reset button : When you press that button, the program that is currently being run in your Arduino will start from the beginning.
* Power ON LED : will be on since power is applied to the Arduino.
* USB jack : Connecting a male USB A to male USB B cable is how you upload programs from your computer to your Arduino board.
* Power jack : The power jack is where you connect a component to power up your Arduino.

Things that Arduino can do

Motion Sensor : It allows you detect movement

Light Sensor : this allows you to "measure" the quantity of light in the outside world.

Humidity and temperature Sensor : this is used to measure the humidity and temperature.

Ultrasonic Sensor : this Sensor allows to determine the distance to an object through Sonar.

Installing the Software (ARDUINO IDE)

The Arduino IDE (Integrated Development Environment) is where you develop your programs that will tell your Arduino what to do.

To download your Arduino IDE, browse on the following link https://www.arduino.cc/en/Main/Software.

Select which Operating System you're using and download it.

Fundamentals of Arduino Programming

1) Structure

The structure of Arduino programming contains of two parts as shown below

```
void setup()
{
  Statement(s);
}
void loop()
{
  Statement();
}
```

2)

```
void setup()
  void loop()
  {
    digitalWrite(pin, HIGH);
    delay(10000);
    digitalWrite(pin, LOW);
    delay(10000);
  }
```

3) Functions

A function is a piece of code that has a name and set of statements executed when function is called.

Functions are declared by its type followed with name of a function.

Syntax : type functionName (parameters)
{
Statement(s);
}

4) { } Curly braces

They define beginning and end of function.

5) Semicolon

It is used to end a statement and separate elements of a program.

Syntax : int x=14;

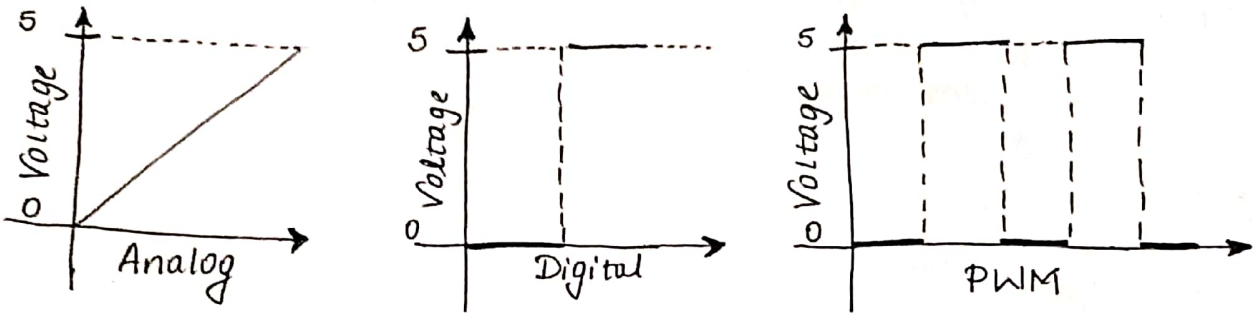
Differences between Analog, Digital and PWM Pins

In analog pins, you have unlimited possible states between 0 and 1023, This allows you to read Sensor Values For example, with a light sensor, if it is very dark, you'll read 1023, if it is very bright you'll read 0 If there is a brightness between dark and very bright you'll read a value between 0 and 1023.

In digital pins, you have just two possible states, which are on or off. These can also be referred as High or Low, 1 or 0 and 5V or 0V. For example, if an LED is on, then, its state is high or 1 or 5V. If it is off, you'll have Low, or 0 or 0V.

PWM pins are digital pins, so they output either 0 or 5V. However these pins can output "fake" intermedia voltage values between 0 and 5V, because they can perform "Pulse Width Modulation" (PWM). PWM allows to "simulate" varying levels of power by oscillating the output voltage of the Arduino.

The below figure shows the representation of Analog, Digital and PWM pins of Arduino.



IOT Physical Devices and Endpoints : RaspberryPi

Introduction to RaspberryPi

The RaspberryPi is a series of credit card sized single-board computers developed in the United Kingdom by Raspberrypi Foundation to promote the teaching of basic computer science in School and developing Countries.

The original model became far more popular than anticipated, selling outside its target market for uses such as robotics. It does not include peripherals and cases. However, some accessories have been included in several official and unofficial bundles.

The Organisation behind the Raspberry Pi consists of two arms. The first two models were developed by the Raspberry Pi Foundation. After the Pi Model B was released, the Foundation setup Raspberry Pi Trading, with Eben Upton as CEO, to develop the third model the B+.

"Why Raspberry Pi?" - Inexpensive, Cross-platform, Simple, clear programming environment, Open Source and extensible Software and Open Source and extensible hardware.

Exploring The Raspberrypi Learning Board

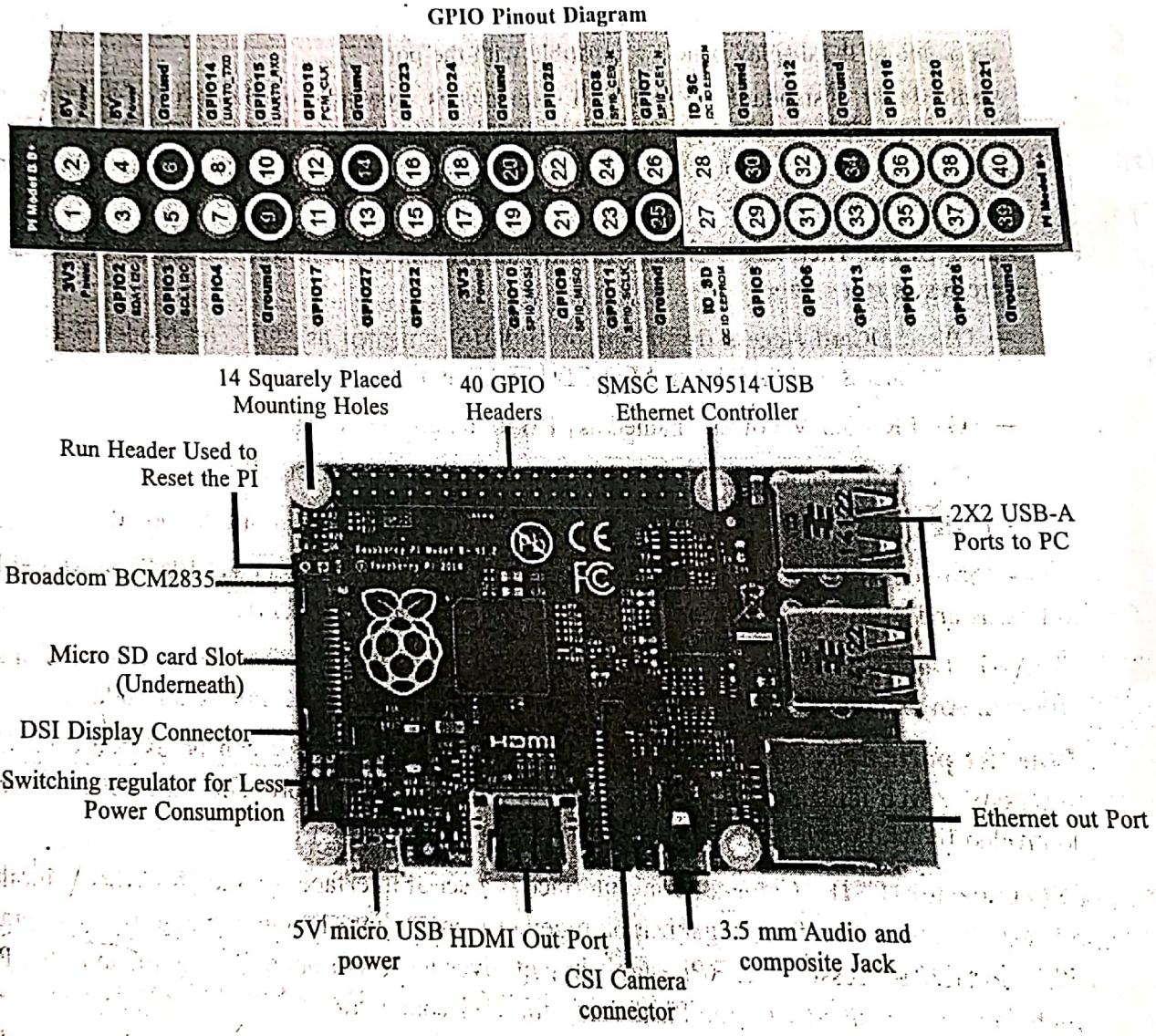


Figure 8-1: Raspberry Pi2 Model B and its GPIO

Processor : The Broadcom BCM2835 SoC used in the first generation Raspberrypi is somewhat equivalent to the chip used in first generation smart phones, which includes a 700 MHz ARM 1176JZF-S processor, Video Core IV graphics processing Unit (GPU) and RAM. This has a level 1 cache of 16KB and a level 2 cache of 128KB.

Power Source : The recommended and easiest way to power the Raspberrypi is Via the Micro USB port on the side of the unit.

SD Card : The Raspberry Pi does not have any locally available storage accessible. The working framework is stacked on a SD card which is embedded on the SD card space on the Raspberry Pi.

GPIO (General Purpose Input Output) : GPIO is a non specific pins on a coordinated circuit to know is an input or output pin which can be controlled by the client at run time. GPIO pins have no exceptional reason characterized, and go unused as a matter of course.

DSI Display X : The Raspberrypi Connector S2 is a display Serial interface (DSI) for connecting a liquid crystal display (LCD) panel using a 15-pin ribbon cable.

Audio Jack : A standard 3.5mm TRS connector is accessible on the RPi for stereo sound yield. Any earphone or 3.5mm sound link can be associated straightforwardly.

Ethernet Port : It is accessible on Model B and B+. It can be associated with a System or web utilizing a standard LAN Link on the Ethernet port.

CSI connector(CSI) : Camera Serial Interface is a Serial interface outlined by MIPI (Mobile Industry Processor Interface) organization together went for interfacing computerized cameras with a portable Processor.

JTAG headers : JTAG is an acronym for 'Join Test Action Group', an association that began back in the mid 1980's to address test point get to issues on PCB with Surface mount gadgets.

Description of System on Chip (SoC)

A System on a chip (SoC) is an integrated Circuit (IC) that co-ordinates all parts of a PC or other electronic framework into a Solitary chip.

It might contained advanced, simple, blended flag, and regularly radio-recurrence works - all on a Solitary chip Substrate. SoCs are exceptionally regular in the portable gadgets advertise in view of their low power utilization. A run of the mill application is in the range of implanted frameworks.

An SoC comprises of:

* A microcontroller, chip or DSP core(s). Some SoCs - called multiprocessor framework on chip (MPSoC) - incorporate more than one processor center.
* Memory pieces including a choice of ROM, RAM, EEPROM and Streak memory.
* Timing Sources including oscillators and Stage bolted circles.
* Simple interfaces including ADCs and DACs.
* Voltage controllers and power administration circuits.

Raspberry Pi interfaces

Raspberry Pi has Serial, SPI and I2C interfaces as shown in the figure of Raspberry Pi Learning board.

* Serial : The Serial interface on Raspberry Pi has receive (rx) and transmit (Tx) pins for Communication With Serial peripherals.
* SPI : Serial Peripheral Interface (SPI) is a Synchronous Serial data Used for communicating with one or more peripheral devices.

* I2C : The I2C interface pins on Raspberry Pi allow you to connect hardware modules. I2C interface allows Synchronous data transfer with just two pins - SDA (data line) and SCL (clock ~~line~~ line).

Raspberry Operating Systems

Various operating systems can be installed on Raspberry through SD cards. Most use a MicroSD slot located on the bottom of the board.

The Raspberrypi primarily uses Raspbian, a Debian-based Linux operating System.

Operating Systems (not Linux based)

- RISC OS Pi
- FreeBSD
- NetBSD
- Plan 9 from Bell Labs and Inferno
- Windows 10 IOT Core - a no cost edition of Windows 10 offered by Microsoft that runs natively on the RaspberryPi 2.

Operating Systems (Linux based)

- Xbian - using Kodi open source digital media center
- openSUSE
- Raspberry Pi Fedora remix
- Pidora, another fedora Remix optimised for Raspberry Pi
- Gentoo Linux
- Diet Pi
- CentOS \ Open Wrt
- Kali Linux
- Ark OS
- Kano OS
- Nard SDK

Media center operating systems

- OSMC
- OpenELEC
- LibreELEC
- Xbian
- Rasplex

Audio operating Systems

- Volumio
- Pimusicbox
- Runeaudio
- moOdeaudio

Recalbox

- Happi Game Center
- Lakka
- ChameleonPi
- Piplay

Operating System Setup On RaspberryPi

Preinstalled NOOBS operating System is already available in many authorized as well as independent Seller, there are many other operating system for RaspberryPi in the market like NOOBS, Raspbian and third party Operating Systems are also available like UBUNTU MATE, OSMC, RISC OS etc. TO setup an operating System we need a SD card with minimum capacity of 8GB.

Formatting SD card

Format the SD card before copying NOOBS onto it. To do this -

- Download SD formatter 4.0 from SD Association Website for either Windows or Mac.

- Follow the instructions to install the Software
- Insert the SD card into the computer or laptops SD card reader and make a note of the drive letter allocated to it.
- In SD formatter, Select the drive letter the SD card is and format it.

OS Installation

Follow the Step to install operating System in SD card

- Go to Raspberry Pi foundation website and click on DOWNLOAD Section.
- Click on NOOBS, then click on "Download zip" button under NOOBS and Select a folder to Save this Zip file.
- Extract all the files from ZIP.
- Once SD card has been formatted, drag all the files in the extracted NOOBS folder and drop them onto the SD card drive.
- The necessary file will then be transferred to the SD card.
- When this process has finished, Safely remove the SD card and insert it into the RaspberryPi.

First Boot

- Plug in the Keyboard, mouse, and monitor cables.
- Now plug the USB cable into the RaspberryPi
- Now Raspberrypi will boot, and a window will appear with a list of different operating System.
- Raspbian will then run through its installation Process.

Programming RaspberryPi With Python

RaspberryPi runs Linux and supports python out of the box. Henceforth you can run any python program that runs on a normal computer. However it is the general purpose input/output capability provided by the GPIO pins on Raspberry Pi that makes it useful device for Internet of things.

Simple python Programs on RaspberryPi:

Program	Code
1. Print hello world	`print ("hello world")`

2. Program to add two numbers	`a = 1.2`
	`b = 5.3`
	`Sum = float(a) + float(b)`
	`Print ("the sum of {0} and {1} is {2}":`
	`format (a,b,sum))`

3. Program to print fibonacci Series	`a,b = 0,1`
	`While b < 200 :`
	`Print(b)`
	`a,b = b, a+b`

4. Program to display calender of given month of the year	`import calender`
	`yy = 2017`
	`mm = 11`
	`print(calender.month(yy.mm))`

5. Program to find the ip address of raspberrypi	`import urllib`
	`import re`
	`point("we will try to open this url, in order to get ip address")`
	`url = http : // checkip . dyndns.org`
	`point(url)`