**ABOUT**

I completed the 6 week summer training on **“Raspberry Pi”**, under the guidance of Dushyant Kumar Singh (Head Of Department).Throughout the training I learnt about the basics of Python-Programming Language and Raspberry Pi (model 3B+), GPIO pins(40) associated with Raspberry Pi and Python codes for interfacing Raspberry Pi with basic sensors like Ultrasonic sensor, PIR sensor, Servomotor and displays like LCD,7 segment display, which helped me in completing my project i.e. Smart Dustbin.

**CHAPTER 1:Overview of Python and Raspberry Pi**

The learning outcomes of week 1 are as follows:

* **Basics of Python Programming (introduction to IDLE-Python Interpreter)** --[Python](https://www.geeksforgeeks.org/python-programming-language/) is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. There are two major Python versions- Python 2 and Python 3.
* **Overview and Introduction to Raspberry Pi**



**Figure 1.0**

GETTING STARTED

The **Raspberry Pi** is a series of small [single-board computers](https://en.wikipedia.org/wiki/Single-board_computer) developed in the [United Kingdom](https://en.wikipedia.org/wiki/United_Kingdom) by the [Raspberry Pi Foundation](https://en.wikipedia.org/wiki/Raspberry_Pi_Foundation) to promote teaching of basic [computer science](https://en.wikipedia.org/wiki/Computer_science) in schools and in [developing countries](https://en.wikipedia.org/wiki/Developing_countries). The original model became far more popular than anticipated, selling outside its [target market](https://en.wikipedia.org/wiki/Target_market) for uses such as [robotics](https://en.wikipedia.org/wiki/Robotics). It does not include peripherals such as [keyboards](https://en.wikipedia.org/wiki/Keyboard_(computing)) and [mouse](https://en.wikipedia.org/wiki/Mouse_(computing)).

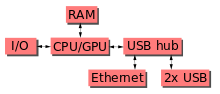
**Generations of Raspberry Pi** :-

| **Family** | **Model** | **Form Factor** | **Ethernet** | **Wireless** | **GPIO** | **Released** |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Raspberry Pi Zero | W/WH | Zero  (65 × 30 mm) | No | Yes | 40-pin | 2017 |  |
| Raspberry Pi Zero | Zero | Zero  (65 × 30 mm) | No | No | 40-pin | 2015 |  |
| Raspberry Pi 4 | B | Standard | Yes | Yes | 40-pin | 2019 |  |
| Raspberry Pi 3 | B | Standard | Yes | Yes | 40-pin | 2016 |  |
| Raspberry Pi 3 | A+ | Compact | No | Yes | 40-pin | 2018 |  |
| Raspberry Pi 3 | B+ | Standard | Yes | Yes | 40-pin | 2018 |  |
| Raspberry Pi 2 | B | Standard | Yes | No | 40-pin | 2015 |  |
| Raspberry Pi 1 | B | Standard  (85.60 × 56.5 mm) | Yes | No | 26-pin | 2012 |  |
| Raspberry Pi 1 | A | Standard  (85.60 × 56.5 mm) | No | No | 26-pin | 2013 |  |
| Raspberry Pi 1 | B+ | Standard | Yes | No | 40-pin | 2014 |  |

**Table 1.0**

Raspberry Pi 4 Model B was released in June 2019 with a 1.5 GHz 64-bit quad core ARM Cortex-A72 processor, on-board 802.11ac Wi-Fi, Bluetooth 5, full gigabit Ethernet (throughput not limited), two USB 2.0 ports, two USB 3.0 ports and dual monitor support (4K resolution).

**Hardware:-**The Raspberry Pi hardware has evolved through several versions that feature variations in [memory](https://en.wikipedia.org/wiki/Computer_memory) capacity and peripheral-device support.

[](https://en.wikipedia.org/wiki/File:Raspberrypi_block_function_v01.svg)

This block diagram describes Model B and B+; Model A, A+, and the Pi Zero are similar, but lack the [Ethernet](https://en.wikipedia.org/wiki/Ethernet) and [USB](https://en.wikipedia.org/wiki/USB) hub components. The Ethernet adapter is internally connected to an additional USB port. In Model A, A+, and the Pi Zero, the USB port is connected directly to the [system on a chip](https://en.wikipedia.org/wiki/System_on_a_chip) (SoC). On the Pi 1 Model B+ and later models the USB/Ethernet chip contains a five-port USB hub, of which four ports are available, while the Pi 1 Model B only provides two. On the Pi Zero, the USB port is also connected directly to the SoC, but it uses a [micro USB](https://en.wikipedia.org/wiki/Micro_USB) (OTG) port. Unlike all other Pi models, the 40 pin GPIO connector is omitted on the Pi Zero with solderable through holes only in the pin locations. The Pi Zero WH remedies this.

**Processor**

[](https://en.wikipedia.org/wiki/File:Raspberry-Pi-2-Bare-BR.jpg)

**Figure 1.1**

The Raspberry Pi 2B uses a 32-bit 900 MHz quad-core [ARM Cortex-A7](https://en.wikipedia.org/wiki/ARM_Cortex-A7) processor.

The [Broadcom](https://en.wikipedia.org/wiki/Broadcom) BCM2835 SoC used in the first generation Raspberry Pi includes a 700 [MHz](https://en.wikipedia.org/wiki/Hertz) [ARM11](https://en.wikipedia.org/wiki/ARM11)76JZF-S processor, [Video Core](https://en.wikipedia.org/wiki/VideoCore) IV [graphics processing unit](https://en.wikipedia.org/wiki/Graphics_processing_unit) (GPU), and RAM. It has a level 1 (L1) [cache](https://en.wikipedia.org/wiki/CPU_cache) of 16 [KB](https://en.wikipedia.org/wiki/Kibibyte) and a level 2 (L2) cache of 128 KB. The [level 2 cache](https://en.wikipedia.org/wiki/Level_2_cache) is used primarily by the GPU. The SoC is [stacked](https://en.wikipedia.org/wiki/Package_on_package) underneath the RAM chip, so only its edge is visible. The ARM1176JZ(F)-S is the same CPU used in the [original iPhone](https://en.wikipedia.org/wiki/IPhone_(1st_generation)), although at a higher [clock rate](https://en.wikipedia.org/wiki/Clock_rate), and mated with a much faster GPU.

The earlier V1.1 model of the Raspberry Pi 2 used a Broadcom BCM2836 SoC with a 900 MHz [32-bit](https://en.wikipedia.org/wiki/32-bit), [quad-core](https://en.wikipedia.org/wiki/Quad_Core) [ARM Cortex-A7](https://en.wikipedia.org/wiki/ARM_Cortex-A7) processor, with 256 KB shared L2 cache. The Raspberry Pi 2 V1.2 was upgraded to a Broadcom BCM2837 SoC with a 1.2 GHz [64-bit](https://en.wikipedia.org/wiki/64-bit_computing) quad-core [ARM Cortex-A53](https://en.wikipedia.org/wiki/ARM_Cortex-A53) processor.

The Raspberry Pi 3+ uses a Broadcom BCM2837B0 SoC with a 1.4 GHz 64-bit quad-core [ARM Cortex-A53](https://en.wikipedia.org/wiki/ARM_Cortex-A53) processor, with 512 KB shared L2 cache.

**RAM**

On the older beta Model B boards, 128 MB was allocated by default to the GPU, leaving 128 MB for the CPU. On the first 256 MB release Model B (and Model A), three different splits were possible. The default split was 192 MB (RAM for CPU), which should be sufficient for standalone 1080p video decoding.

For the later Model B with 512 MB RAM, new standard memory split files (arm256\_start.elf, arm384\_start.elf, arm496\_start.elf) were initially released for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM) respectively. But a week or so later the RPF released a new version of start.elf that could read a new entry in config.txt (gpu\_mem=*xx*) and could dynamically assign an amount of RAM (from 16 to 256 MB in 8 MB steps) to the GPU, so the older method of memory splits became obsolete, and a single start.elf worked the same for 256 MB and 512 MB Raspberry Pis.

The Raspberry Pi 2 and the Raspberry Pi 3 have 1 GB of RAM. The Raspberry Pi Zero and Zero W have 512 MB of RAM.

**Networking**

The Model A, A+ and Pi Zero have no Ethernet circuitry and are commonly connected to a network using an external user-supplied USB Ethernet or [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) adapter. On the Model B and B+ the Ethernet port is provided by a built-in USB Ethernet adapter using the SMSC LAN9514 chip. The Raspberry Pi 3 and Pi Zero W (wireless) are equipped with 2.4 GHz WiFi [802.11n](https://en.wikipedia.org/wiki/IEEE_802.11n-2009) (150 Mbit/s) and [Bluetooth 4.1](https://en.wikipedia.org/wiki/Bluetooth_4.1) (24 Mbit/s) based on the Broadcom BCM43438 [FullMAC](https://en.wikipedia.org/wiki/Wireless_network_interface_controller#FULLMAC) chip with no official support for [monitor mode](https://en.wikipedia.org/wiki/Monitor_mode) but implemented through unofficial firmware patchingand the Pi 3 also has a 10/100 Mbit/s Ethernet port. The Raspberry Pi 3B+ features dual-band [IEEE 802.11b/g/n/ac WiFi](https://en.wikipedia.org/wiki/IEEE_802.11), [Bluetooth 4.2](https://en.wikipedia.org/wiki/Bluetooth_4.2), and [Gigabit Ethernet](https://en.wikipedia.org/wiki/1000BASE-T) (limited to approximately 300 Mbit/s by the [USB 2.0](https://en.wikipedia.org/wiki/USB_2.0) bus between it and the SoC). The Raspberry Pi 4 has full [gigabit Ethernet](https://en.wikipedia.org/wiki/Gigabit_Ethernet) (throughput is not limited as it is not funnelled via the USB chip.)

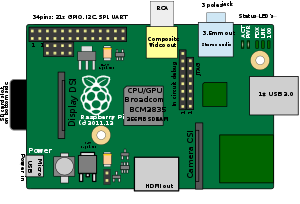
**Peripherals**

[](https://en.wikipedia.org/wiki/File:Raspberry-Pi-2-Bare-FL.jpg)

The Model 2B boards incorporate four USB ports for connecting peripherals(keyboard or mouse).

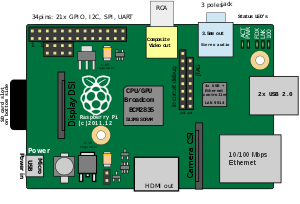
**Pi Zero**

|  |  |
| --- | --- |
| **Figure 1.1**(Location of connectors and main ICs)  **Model A** |  |

[](https://en.wikipedia.org/wiki/File:Raspberry_Pi_1A.svg)

**Figure 1.2**(Location of connectors and main ICs on Raspberry Pi 1 Model A)

**Model B**

[](https://en.wikipedia.org/wiki/File:Drawing_of_Raspberry_Pi_model_B_rev2.svg)

**Figure 1.3**(Location of connectors and main ICs on Raspberry Pi 1 Model B revision 1.2)

**General purpose input-output (GPIO) connector**

Raspberry Pi 1 Models A+ and B+, Pi 2 Model B, Pi 3 Models  A+, B and B+, Pi 4, and Pi Zero, Zero W, and Zero WH GPIO J8 have a 40-pin pinout. Raspberry Pi 1 Models A and B have only the first 26 pins.

In the Pi Zero and Zero W the 40 GPIO pins are unpopulated, having the through-holes exposed for soldering instead. The Zero WH (Wireless + Header) has the header pins preinstalled.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **GPIO#** | **2nd fun.** | **Pin#** |  | **Pin#** | **2nd fun.** | **GPIO#** |
|  | +3.3 V | 1 |  | 2 | +5 V |  |
| 2 | SDA1 (I²C) | 3 |  | 4 | +5 V |  |
| 3 | SCL1 (I²C) | 5 |  | 6 | GND |  |
| 4 | GCLK | 7 |  | 8 | TXD0 (UART) | 14 |
|  | GND | 9 |  | 10 | RXD0 (UART) | 15 |
| 17 | GEN0 | 11 |  | 12 | GEN1 | 18 |
| 27 | GEN2 | 13 |  | 14 | GND |  |
| 22 | GEN3 | 15 |  | 16 | GEN4 | 23 |
|  | +3.3 V | 17 |  | 18 | GEN5 | 24 |
| 10 | MOSI (SPI) | 19 |  | 20 | GND |  |
| 9 | MISO (SPI) | 21 |  | 22 | GEN6 | 25 |
| 11 | SCLK (SPI) | 23 |  | 24 | CE0\_N (SPI) | 8 |
|  | GND | 25 |  | 26 | CE1\_N (SPI) | 7 |
| *(Pi 1 Models A and B stop here)* | | | | | | |
| EEPROM | ID\_SD | 27 |  | 28 | ID\_SC | EEPROM |
|  |  |  |  |  |  |  |
| 5 | N/A | 29 |  | 30 | GND |  |
| 6 | N/A | 31 |  | 32 |  | 12 |
| 13 | N/A | 33 |  | 34 | GND |  |
| 19 | N/A | 35 |  | 36 | N/A | 16 |
| 26 | N/A | 37 |  | 38 | Digital IN | 20 |
|  | GND | 39 |  | 40 | Digital OUT | 21 |

**Figure 1.4**

Models A and B provide GPIO access to the ACT status LED using GPIO 16. Models A+ and B+ provide GPIO access to the ACT status LED using GPIO 47, and the power status LED using GPIO 35.

**Software**

**Operating systems**

[](https://en.wikipedia.org/wiki/File:Raspberry-Pi-2-Bare-Bottom.jpg)

Various operating systems for the Raspberry Pi can be installed on a MicroSD, MiniSD or SD card, depending on the board and available adapters; seen here is the MicroSD slot located on the bottom of a Raspberry Pi 2 board.

The Raspberry Pi Foundation provides [Raspbian](https://en.wikipedia.org/wiki/Raspbian), a Debian-based [Linux distribution](https://en.wikipedia.org/wiki/Linux_distribution) for download, as well as third-party [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)), [Windows 10 IoT Core](https://en.wikipedia.org/wiki/Windows_10_IoT_Core), [RISC OS](https://en.wikipedia.org/wiki/RISC_OS), and specialised [media centre](https://en.wikipedia.org/wiki/OpenELEC) distributions. It promotes [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and [Scratch](https://en.wikipedia.org/wiki/Scratch_(programming_language)) as the main programming languages, with support for many other languages. The default [firmware](https://en.wikipedia.org/wiki/Firmware) is [closed source](https://en.wikipedia.org/wiki/Closed_source), while an unofficial [open source](https://en.wikipedia.org/wiki/Open_source) is available. Many other operating systems can also run on the Raspberry Pi, including the formally verified microkernel, [seL4](https://en.wikipedia.org/wiki/L4_microkernel_family#High_assurance:_seL4). Other third-party operating systems available via the official website include [Ubuntu MATE](https://en.wikipedia.org/wiki/Ubuntu_MATE), [Windows 10 IoT Core](https://en.wikipedia.org/wiki/Windows_10_IoT_Core), [RISC OS](https://en.wikipedia.org/wiki/RISC_OS) and specialised distributions for the [Kodi](https://en.wikipedia.org/wiki/Kodi_(software)) media centre and classroom management.

**Firmware**

The official firmware is a [freely redistributable](https://en.wikipedia.org/wiki/Freely_redistributable_software)[binary blob](https://en.wikipedia.org/wiki/Binary_blob), that is [closed-source](https://en.wikipedia.org/wiki/Closed-source). A minimal proof-of-concept open source firmware is also available, mainly aimed at initializing and starting the ARM cores as well as performing minimal startup that is required on the ARM side. It is also capable of booting a very minimal [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel), with patches to remove the dependency on the mailbox interface being responsive. It is known to work on Raspberry Pi 1, 2 and 3, as well as some variants of Raspberry Pi Zero. While it is in a working state, it is not actively developed, with last significant commits made around mid-2017.

**Uses**

* Use in education
* Use in Home automation
* Use in Industrial automation
* Use in Commercial products

**CHAPTER 2:Sensors,Displays and their working**

The learning outcomes of week 2 are as follows:

* Features of Raspberry Pi – The Raspberry Pi uses Raspbian (Debian Based) as its operating system which can be installed on the micro SD card which is later inserted into the Raspberry Pi. The Raspbian can be installed on the SD card via NOOBS (New out of the box software) or by directly mounting the image on the SD card. I also learnt about remotely accessing Raspberry pi on Laptop using Remote Desktop Application and VNC server.
* Interfacing and working of basic sensors and Displays –Some of the basic sensors include:

1. **PIR sensor**: PIR sensor detects a human being moving around within approximately 10m from the sensor. This is an average value, as the actual detection range is between 5m and 12m.PIR are fundamentally made of a pyro electric sensor, which can detect levels of infrared radiation.



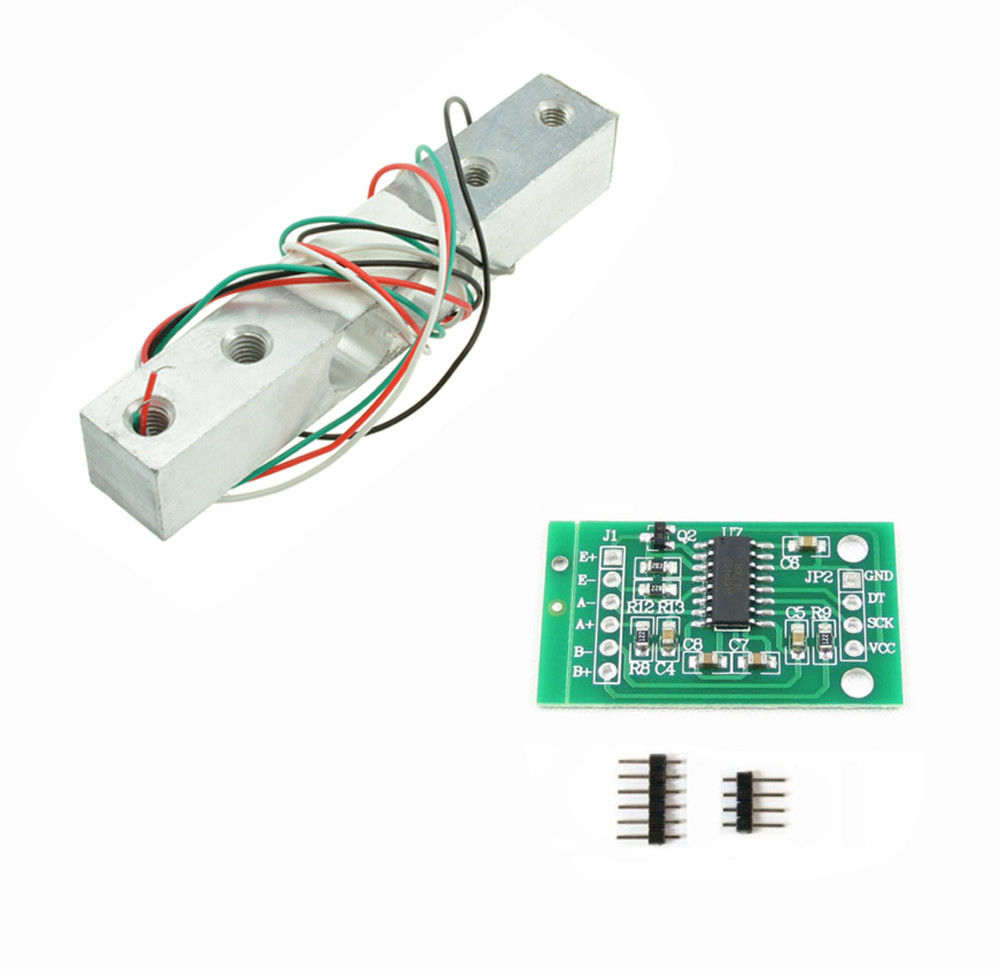
**Figure 1.6**

Working:

The PIR sensors are more complicated than the other sensors as they consists of two slots. These slots are made of a special material which is sensitive to IR. The Fresnel lens is used to see that the two slots of the PIR can see out past some distance. When the sensor is inactive, then the two slots sense the same amount of IR. The ambient amount radiates from the outdoors, walls or room etc.

When a human body or any animal passes by, then it intercepts the first slot of the PIR sensor. This causes a positive differential change between the two bisects. When a human body leaves the sensing area, the sensor generates a negative differential change between the two bisects. The infrared sensor itself is housed in a hermetically sealed metal to improve humidity/temperature/noise/immunity. There is a window which is made of typically coated silicon material to protect the sensing element.

1. **Load Cell**: A load cell is a type of [force gauge](https://en.wikipedia.org/wiki/Force_gauge). It consists of a [transducer](https://en.wikipedia.org/wiki/Transducer) that is used to create an [electrical signal](https://en.wikipedia.org/wiki/Electrical_signal) whose magnitude is directly proportional to the [force](https://en.wikipedia.org/wiki/Force) being measured. The various load cell types include hydraulic, pneumatic, and strain gauge.



**Figure 1.7**

Working: A load cell usually consists of four strain gauges in a [Wheatstone bridge](https://en.wikipedia.org/wiki/Wheatstone_bridge) configuration. Load cells of one strain gauge (quarter bridge) or two strain gauges (half bridge) are also available. The electrical signal output is typically in the order of a few millivolts (mV) and requires amplification by an [instrumentation amplifier](https://en.wikipedia.org/wiki/Instrumentation_amplifier) before it can be used. The output of the transducer can be scaled to calculate the force applied to the transducer. Sometimes a high resolution [ADC](https://en.wikipedia.org/wiki/Analog-to-digital_converter), typically 24-bit, can be used directly.

3)**16\*2 LCD**:

****

**Figure 1.8**

We come across [LCD](https://electronicsforu.com/videos-slideshows/videos/building-liquid-crystal-display-lcd) displays everywhere around us. Computers, calculators, television sets, mobile phones, digital watches use some kind of display to display the time. An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in [DIYs](https://electronicsforu.com/category/electronics-projects/hardware-diy) and circuits. The 16×2 translates is a display with 16 characters per line in 2 such lines. In such LCD each character is displayed in a 5×7 pixel matrix.

Working: The principle behind the LCD’s is that when an electrical current is applied to the liquid crystal molecule, the molecule tends to untwist. This causes the angle of light which is passing through the molecule of the polarized glass and also cause a change in the angle of the top polarizing filter. As a result a little light is allowed to pass the polarized glass through a particular area of the LCD. Thus that particular area will become dark compared to other. The LCD works on the principle of blocking light.

4) **Ultrasonic Sensor:**

HC-SR04 Sensor Features-

* Operating voltage: +5V
* Theoretical Measuring Distance: 2cm to 450cm
* Practical Measuring Distance: 2cm to 80cm
* Accuracy: 3mm
* Measuring angle covered: <15°
* Operating Current: <15mA
* Operating Frequency: 40Hz

**HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

**Distance = Speed × Time**

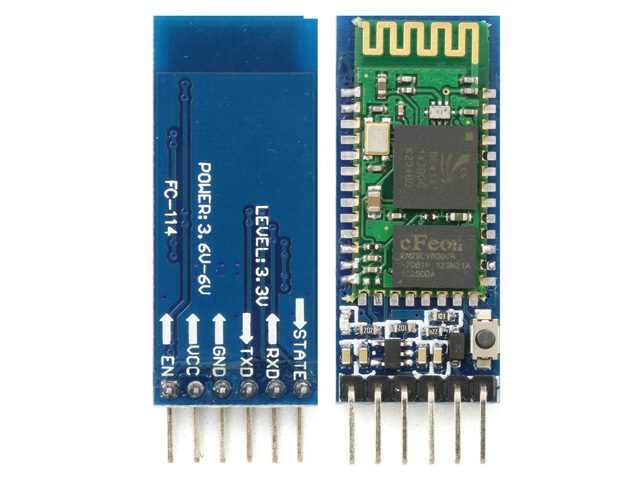


**Figure 1.9**

Working: The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver.

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

5) **Bluetooth Sensor**: HC-05 is a Bluetooth module which is designed for wireless communication. This module can be used in a master or slave configuration.



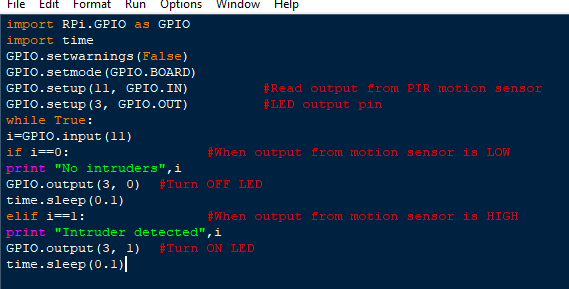
**Figure 2.0**

* It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard and many more consumer applications.
* It has range up to <100m which depends upon transmitter and receiver, atmosphere, geographic & urban conditions.
* It is IEEE 802.15.1 standardized protocol, through which one can build wireless Personal Area Network ([PAN](https://en.wikipedia.org/wiki/Personal_area_network)). It uses frequency-hopping spread spectrum ([FHSS](https://en.wikipedia.org/wiki/Frequency-hopping_spread_spectrum)) radio technology to send data over air.
* It uses serial communication to communicate with devices. It communicates with microcontroller using serial port (USART).

**CHAPTER 3:Interfacing Sensors and Displays with Raspberry Pi**

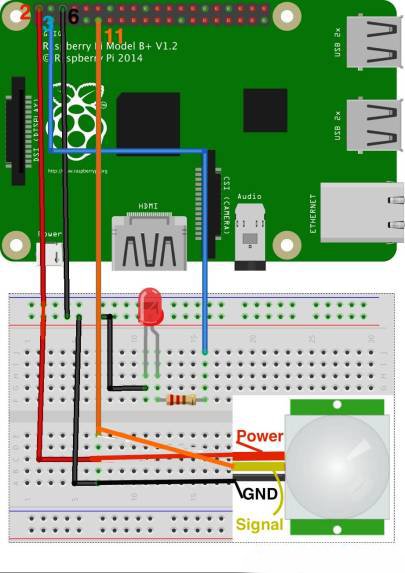
1. **Interfacing PIR sensor with Raspberry Pi**: We can try reading the output from the PIR motion sensor. The sensor outputs a digital HIGH (5V) signal when it detects a person.

Python Code:



**Figure 2.1**

Connection:



**Figure 2.2**

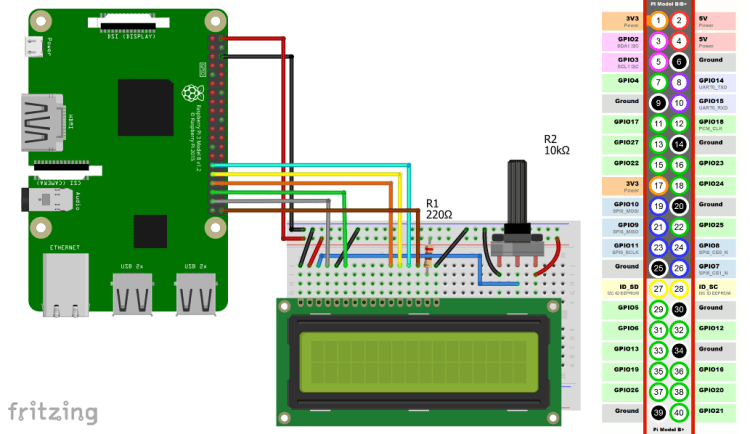
This code prints "Intruder detected" when you place your hand over the sensor. After removing your hand and waiting some time, it prints: "No intruders". In certain PIR motion sensors, you can even adjust the delay at which the sensor outputs a HIGH signal at the expense of compromising the accuracy. You just need to turn the two knobs on the sensor counterclockwise using a screwdriver.

1. **Interfacing 16\*2 LCD with Raspberry Pi:**

The components that will be required for Raspberry pi LCD display interfacing are as follows

* Raspberry pi
* 16X2 LCD
* 10k potentiometer
* 220ohm resistor
* Connecting wires

Connection:



**Figure 2.3**

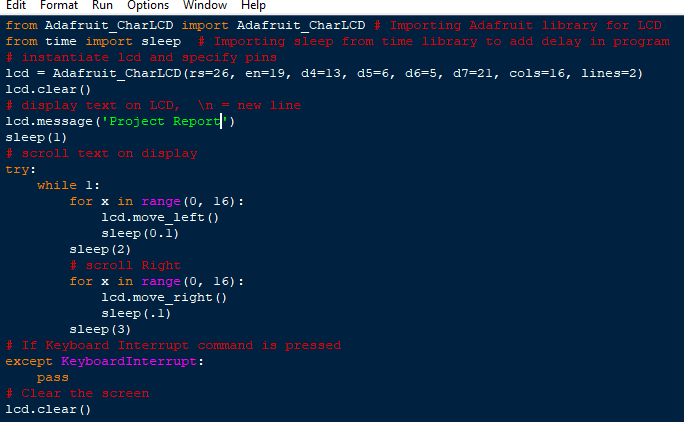
**Installing the Library**

1. First of all, you will need to install the Adafruit library for LCD, so type the below command in the Raspberry pi terminal.
   1. git clone https://github.com/adafruit/Adafruit\_Python\_CharLCD.git
2. Then enter into the directory we just created by typing the following command.
   1. cd ./Adafruit\_Python\_CharLCD

  3.Now run the setup by typing the below command.

sudo python setup.py install

Python Code to print and scroll text on LCD:



**Figure 2.4**

1. Interfacing Load Cell with Raspberry Pi:

Before the load cell is connected to the HX711 weight sensor, it should be mounted on the two plates.

The four cables of the Load Cell must be connected to the weight sensor. The green HX711, however, has six connections, of which we only need four for the cables. The connection is as follows:

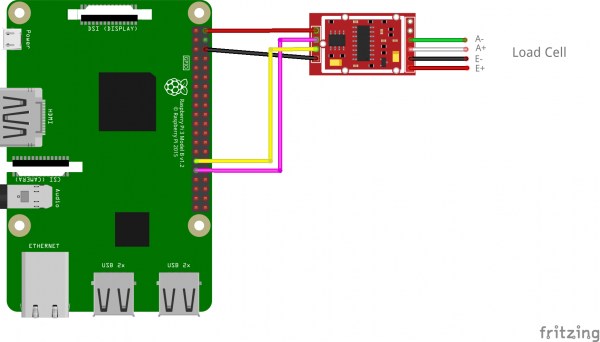
* Red: E+
* Black: E-
* Green: A-
* White: A+

The pins labeled B+/B- remain empty. Apparently there are versions of the sensor. Where the pins are labeled S+/S- instead of A+/A-.

Now we just have to connect the sensor to the Raspberry Pi. Since this also has only four connections, the wiring is quite simple:

* VCC to Raspberry Pi Pin 2 (5V)
* GND to Raspberry Pi Pin 6 (GND)
* DT to Raspberry Pi Pin 29 (GPIO 5)
* SCK to Raspberry Pi Pin 31 (GPIO 6)

Connection:

[](https://tutorials-raspberrypi.com/?p=2975)

**Figure 2.5**

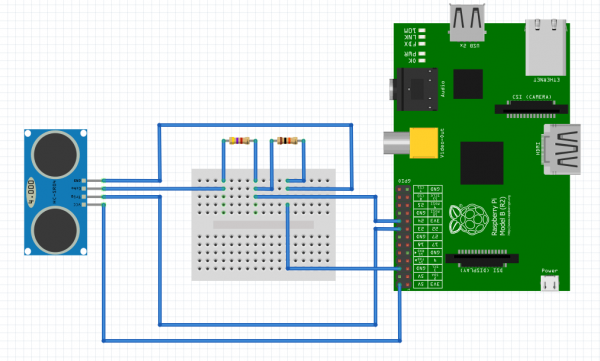
To measure the weight and to read out the value we use a Python library.

“git clone https://github.com/tatobari/hx711py”

It contains an example.py file which shows the function of the library and can also be used.

1. Interfacing Ultrasonic Sensor with Raspberry Pi:

Connection:



**Figure 2.6**

**Distance Calculation**

Time taken by pulse is actually for to and fro travel of ultrasonic signals, while we need only half of this therefore Time is taken as Time/2.

**Distance = Speed \* Time/2**

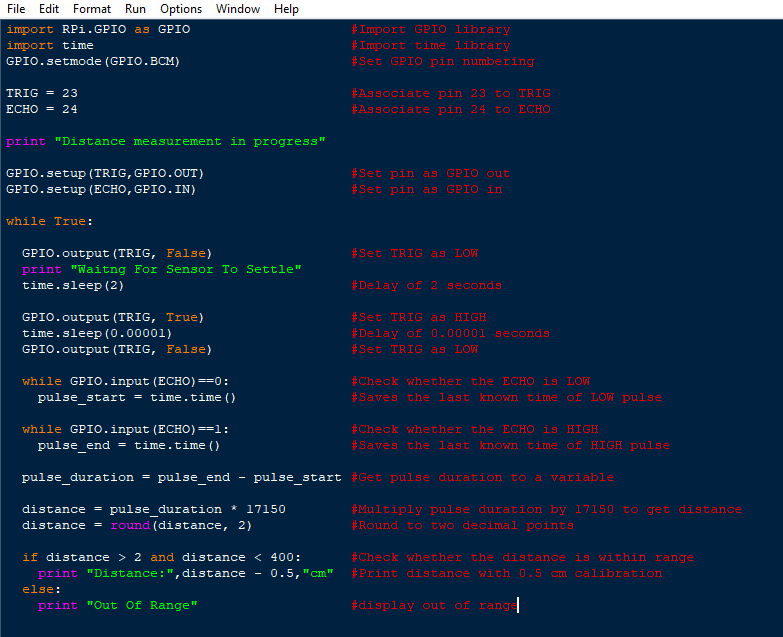
Speed of sound at sea level = 343 m/s or 34300 cm/s

Thus, **Distance = 17150 \* Time (unit cm)**

**Calibration**

For accurate distance readings the output can be calibrated using a ruler. In the Python code a calibration of 0.5 cm is added.

Python Code:



**Figure 2.7**

Distance is measured every two seconds and displayed.

**CHAPTER 4: Project –“Smart Dustbin”**

**Project Description:** The project uses a dustbin to efficiently convert it into a “Smart Dustbin” that can smartly detect that whether a person is standing in front of it and based on that its lid can be opened automatically without any human intervention. Also, the dustbin detects the presence of garbage being stored inside it. If the garbage exceeds a certain limit then there will be a message displayed stating that the dustbin is full and the lid will not open. The data of the sensor (present inside to detect the garbage level) will be sent to the cloud and based on the values received an email will be sent (using IFTT) to the concerned authority that the particular Dustbin is full and that the garbage should be collecting from that location.

**Components Required**:

* Ultrasonic Sensor(Quantity:2)-

**Function:**

Two Ultrasonic sensors are used in the project.

One is placed in the front of the Dustbin that can detect that whether a person is standing in front of it or not and also displays at what distance the person is standing. The other is placed at the inner side of the Lid that detects that whether the Dustbin is full or not.

* Servomotor:

A **servo motor** is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through **servo mechanism**. If motor is used is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC servo motor. We can get a very high torque servo motor in a small and light weight packages. Doe to these features they are being used in many applications like toy car, RC helicopters and planes, Robotics, Machine etc.

**Working principle of Servo Motor:**

A servo consists of a Motor (DC or AC), a potentiometer, gear assembly and a controlling circuit. First of all we use gear assembly to reduce RPM and to increase torque of motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now difference between these two signals, one comes from potentiometer and another comes from other source, will be processed in feedback mechanism and output will be provided in term of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with potentiometer and as motor rotates so the potentiometer and it will generate a signal. So as the potentiometer’s angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.



**Figure 2.8**

**Controlling Servomotor**:

Servo motor works on PWM (Pulse width modulation) principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically servo motor is made up of DC motor which is controlled by a variable resistor (potentiometer) and some gears. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. Potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on required angle.

Servo motor can be rotated from 0 to 180 degree, but it can go up to 210 degree, depending on the manufacturing.

**Function:**

The servomotor is used to turn the lid up or down based on the readings of the Ultrasonic sensor (placed in the front of dustbin)



**Figure 2.9**

**Working:** The dustbin has an ultrasonic sensor placed in front which emits ultrasonic waves and measures the distance at which an obstacle is placed in front of it by measuring the time taken by the ultrasonic wave to reach back to the receiver of the ultrasonic sensor. When a person stands close enough to the dustbin the ultrasonic sensor sends a HIGH(1) signal to the servomotor attached to the lid which rotates the servomotor by an angle of 180 degrees and the lid is opened. An ultrasonic sensor is placed at the bottom side of the lid which checks that whether the garbage has reached a certain level or not (by emitting the ultrasonic waves and measuring distance by time taken by the wave to return back) and if it has reached a message is displayed that the dustbin is full and the lid won’t open. LCD is used for the interaction of Dustbin with the outside World. The values of the sensor which checks for the garbage level are sent to the cloud (in this case Thingspeak) and based on the values received if the dustbin is full then an email is sent to the authority that the particular dustbin is full and the garbage needs to be collected from that location. The email is sent using the IFTT (If this Then that) service.

**Conclusion**

 A simple but useful project called Smart Dustbin using Raspberry Pi is designed and developed here. Using this project, the lid of the dustbin stays closed, so that waste is not exposed (to avoid flies and mosquitos) and when you want dispose any waste, it will automatically open the lid.

References

* <https://www.electronicshub.org/smart-dustbin-using-arduino/>
* <https://www.youtube.com/watch?v=9yrP1CZN3Ds> (Smart dustbin)
* <https://www.keyence.com/ss/products/sensor/sensorbasics/ultrasonic/info/index.jsp>
* <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>
* <https://www.geeksforgeeks.org/python-language-introduction/>
* <https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started>
* <https://pythonprogramming.net/introduction-raspberry-pi-tutorials/>
* <https://pythonprogramming.net/remote-access-raspberry-pi-tutorials/>
* <https://circuitdigest.com/article/servo-motor-basics>
* https://en.wikipedia.org/wiki/Raspberry\_Pi