MES Wadia College of Engineering Pune-01

Department of Computer Engineering

Name of Student:	**Class:**
Semester/Year:	**Roll No:**
Date of Performance:	**Date of Submission:**
Examined By:	**Experiment No: 02**

ASSIGNMENT NO: 02

AIM: Study of Connectivity and configuration of Raspberry-Pi circuit with basic peripherals, LEDS. Understanding GPIO and its use in program.

OBJECTIVES:

- To know connections of peripherals devices with Raspberry pi GPIO.
- To understand GPIO pin configuration, GPIO mode and Pin modes.

APPRATUS:

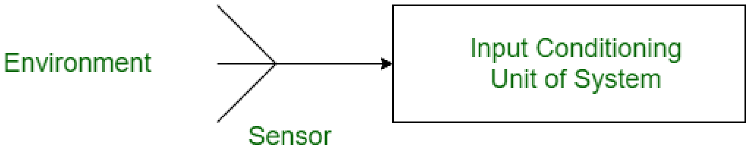
- Raspberry-Pi 3/4 module, Micro-SD Card, Power Adapter, HDMI to VGA cable.
- Breadboard, LEDs, Jumping Wires, 220Ω or 1KΩresistor
- Operating System recommended: Raspbian Stretch with PIXEL with Thonny python IDE.

THEORY:

A. Difference between Sensor and Actuator:

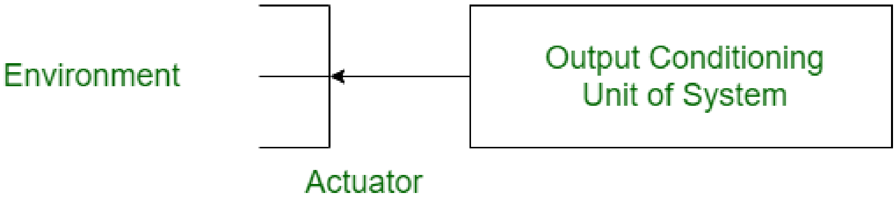
1. Sensor:

- ✓ A device which detects or measures a physical property and records, indicates, or otherwise responds to it."
- ✓ Meaning of Sensor as per Wikipedia "A sensor is an object whose purpose is to detect events or changes in its environment, and then provide a corresponding output"
- ✓ This is a hardware device that takes the input from environment and gives to the system by converting it.
- ✓ For example, a thermometer takes the temperature as physical characteristic and then converts it into electrical signals for the system.
- ✓ A sensor is classified based on various aspects such as-
  - Application Based: Industrial Sensor, Automotive Sensor etc.
  - Output Based: Resistive output, Differential Output, Voltage Output etc.
  - Parameter Sensing Based: Light, Temperature, Sound, Gas etc.



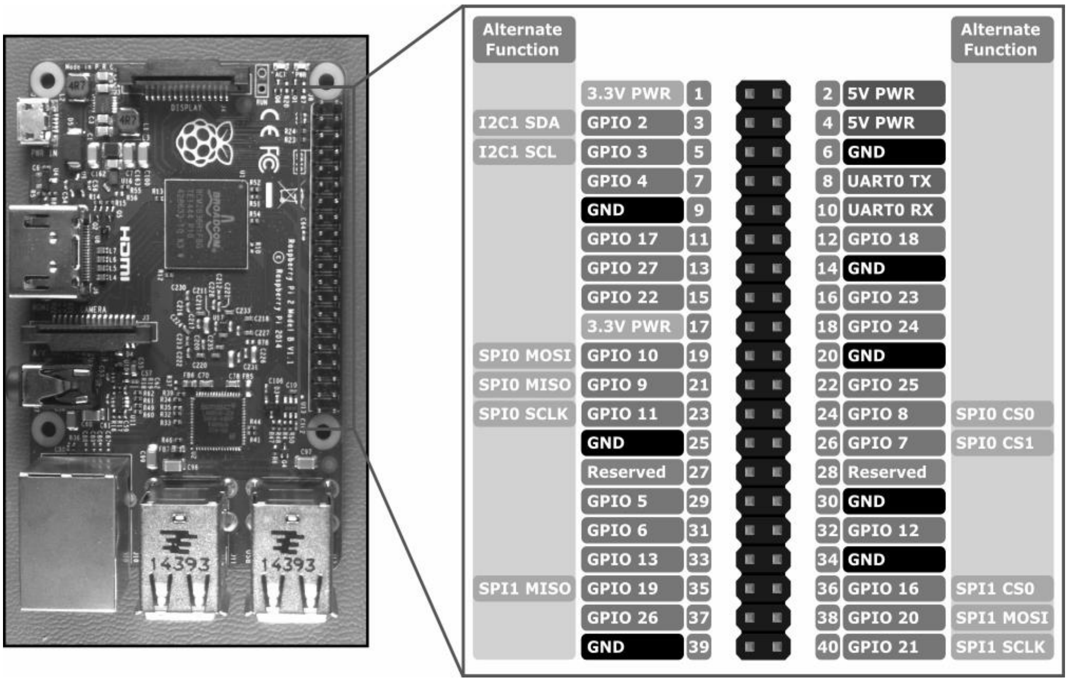
2. Actuator:

- ✓ An actuator is a component of a machine that is responsible for moving or controlling a mechanism or system.
- ✓ An actuator requires a control signal and a source of energy. The control signal is relatively
- ✓ low energy and may be electric voltage or current, pneumatic or hydraulic pressure, or even human power.
- ✓ When the control signal is received, the actuator responds by converting the energy into mechanical motion.
- ✓ Actuator is a device that converts the electrical signals into the physical events or characteristics. It takes the input from the system and gives output to the environment. Following basic actuators are used for signaling and output purpose:
  - LED, RGB LED, Buzzer, Servo Motor, DC Motor, Relay etc



B. GPIO:

GPIO stands for **General Purpose Input Output**. It is a way the Raspberry Pi can control and monitor the outside world by being connected to electronic circuits. The Pi is able to control LEDs, turning them on or off, or motors, or many other things. It is also able to detect whether a switch has been pressed, or temperature, or light.



Alternate Function					Alternate Function
	3.3V PWR	1	2	5V PWR	
I2C1 SDA	GPIO 2	3	4	5V PWR	
I2C1 SCL	GPIO 3	5	6	GND	
	GPIO 4	7	8	UART0 TX	
	GND	9	10	UART0 RX	
	GPIO 17	11	12	GPIO 18	
	GPIO 27	13	14	GND	
	GPIO 22	15	16	GPIO 23	
	3.3V PWR	17	18	GPIO 24	
SPI0 MOSI	GPIO 10	19	20	GND	
SPI0 MISO	GPIO 9	21	22	GPIO 25	
SPI0 SCLK	GPIO 11	23	24	GPIO 8	SPI0 CS0
	GND	25	26	GPIO 7	SPI0 CS1
	Reserved	27	28	Reserved	
	GPIO 5	29	30	GND	
	GPIO 6	31	32	GPIO 12	
	GPIO 13	33	34	GND	
SPI1 MISO	GPIO 19	35	36	GPIO 16	SPI1 CS0
	GPIO 26	37	38	GPIO 20	SPI1 MOSI
	GND	39	40	GPIO 21	SPI1 SCLK

Fig. 3.1 Raspberry Pi GPIO Pinout Diagram

GPIO includes pins for power (5V and 3.3V), ground pins, GPIO pins and reserved pins. Some GPIO pins are having alternative function for interfaces like SPI, I2C and UART.

To use GPIO pins, use a simple python library called **RPI.GPIO** which makes easy to use and configure the pins. ***import RPi.GPIO as GPIO***

Modes of GPIO:

- ✓ There are two modes namely GPIO.BOARD and GPIO.BCM.
- ✓ The **GPIO.BOARD** option specifies that you are referring to the pins by the number of the pin the the plug - i.e the numbers printed on the board (e.g. P1). For example the GPIO is pin 7.
- ✓ The **GPIO.BCM** option means that you are referring to the pins by the "Broadcom SOC channel" number, these are the numbers after "GPIO" in the green rectangles around the outside of the diagrams:
- ✓ Example: GPIO 4 pin will be referred as 4 rather than 7.
- ✓ To specify in your code which number-system is being used, use the GPIO.setmode() function. For example GPIO.setmode(GPIO.BCM).
- ✓ Unfortunately the BCM numbers changed between versions of the Pi1 Model B.
    - – The Model B+ uses the same numbering as the Model B r2.0, and adds new pins (board numbers 27-40).
    - – The Raspberry Pi Zero, Pi 2B and Pi 3B use the same numbering as the B+.

C. Breadboard:

The breadboard is a way of connecting electronic components to each other without having to solder them together. They are often used to test a circuit design before creating a Printed Circuit Board (PCB). The holes on the breadboard are connected in a pattern.

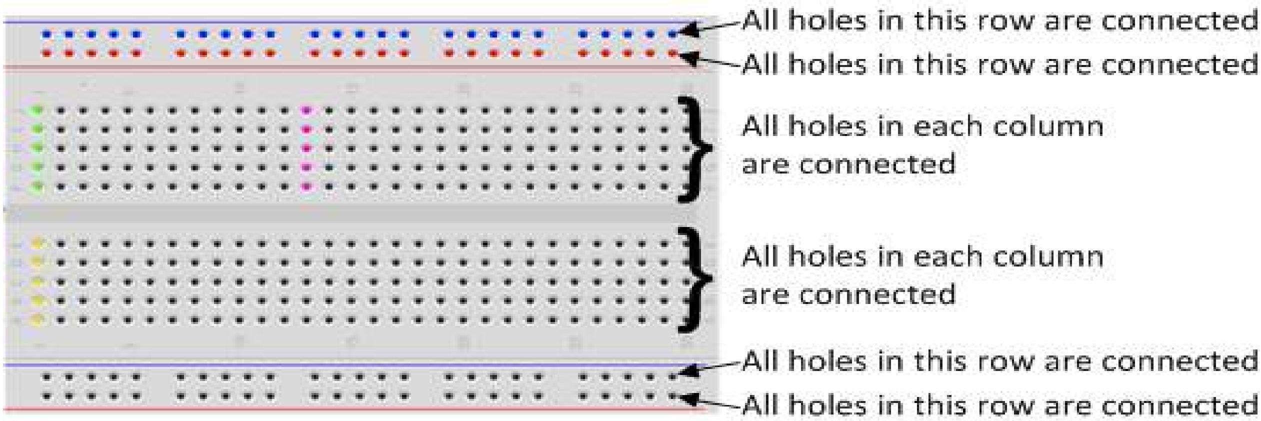
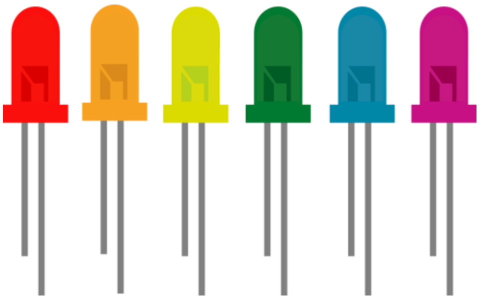
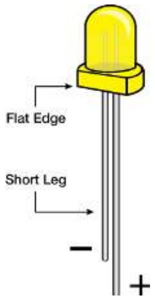

All holes in this row are connected
All holes in this row are connected
All holes in each column are connected
All holes in each column are connected
All holes in this row are connected
All holes in this row are connected

Fig. 3.2. A Sample Breadboard.

D. LED:

LED stands for Light Emitting Diode, and glows when electricity is passed through it. When you pick up the LED, you will notice that one leg is longer than the other. The longer leg (known as the 'anode'), is always connected to the positive supply of the circuit. The shorter leg (known as the 'cathode') is connected to the negative side of the power supply, known as 'ground'.





E. Jumper Wires:

Jumper wires are used on breadboards to 'jump' from one connection to another. The ones you will be using in this circuit have different connectors on each end. The end with the 'pin' will go into the Breadboard. The end with the piece of plastic with a hole in it will go onto the Raspberry Pi's GPIO pins.

Building the Circuit:

The circuit consists of a power supply (the Pi), an LED that lights when the power is applied, and you may use a resistor to limit the current that can flow through the circuit.

1. Connect longer leg of LED (Anode) to GPIO pin mentioned in your python program.
2. Connect shorter leg of LED(cathode) to ground pin of GPIO.

Write a program in python to interface LED with raspberry pi.

We have to declare a “pin mode” before you can use it as either an input or output. To set a pin mode, use setup([pin], [GPIO.IN, GPIO.OUT] function.

So, if you want to set pin 18 as an output, for example, write:

```
GPIO.setup(18, GPIO.OUT)
```

Digital Output:

To write a pin high or low, use the GPIO.output([pin], [GPIO.LOW, GPIO.HIGH]) function. For example, if you want to set pin 18 high, write:

```
GPIO.output(18, GPIO.HIGH)
```

Delays:

If you need to slow your Python script down, you can add delays. You also can use delay between led goes on and off. To incorporate delays into your script, you’ll need to include another module: time. For eg, include time

Then, throughout the rest of your script, you can use time.sleep([seconds]) to give your script a rest. You can use decimals to precisely set your delay. For example, to delay 250 milliseconds, write: time.sleep(0.25)

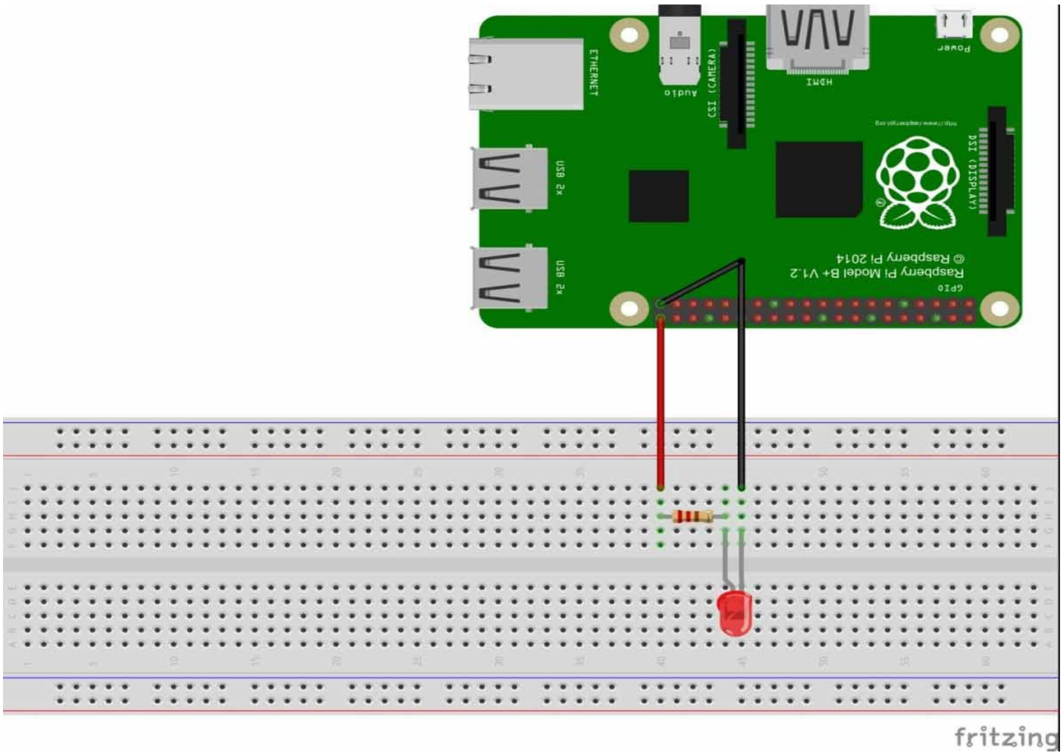
Garbage Collecting:

Once your script has run its course, be kind to the next process that might use your GPIOs by cleaning up after yourself. Use the **GPIO.cleanup()** command at the end of your script to release any resources your script may be using.

Experiment Setup:

Circuit diagram for Raspberry Pi LED Blink is given below:

As shown in the circuit diagram we are going to connect an LED between PIN40 (GPIO21) and PIN39 (GROUND). As said earlier, we cannot draw more than 15mA from any one of these pins, so to limit the current we are connecting a 220Ω or 1KΩ resistor in series with the LED.



CONCLUSION:

QUESTIONS:

1. Explain the terms SPI, I2C and UART interfaces.
2. Discuss raspi-config for configuration of Raspberry pi.
3. Draw neat circuit diagram showing connections used for this assignment.