**ASSIGNMENT NUMBER: C2**

**Title**

Controlling the operation of a hardware simulated traffic signal using Raspberry Pi /Beagle board circuit.

**Problem Statement**

Write an application using Raspberry-Pi /Beagle board to control the operation of a hardware simulated traffic signal.

**Objective**

1. Understanding the controlling of devices through Raspberry Pi /Beagle board.
2. To understand the actuation.

**Outcomes**

I will be able to

1. To simulate traffic signal through LEDs.
2. To control this simulated traffic signal through Raspberry Pi /Beagle board.
3. Can perform actuation.

**Software & Hardware Requirements**

1. Raspberry pi board/ BBB
2. DTH-11 temperature sensor
3. LED
4. Raspbian (OS)
5. Adafruit\_DTH Library

**Theory**

**Raspberry** **Pi**

The Raspberry Pi is a series of credit card-sized single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word processing, browsing the internet, and playing games. It also plays high definition video.

One powerful feature of the Raspberry Pi is the row of GPIO (general purpose input/output) pins along the top edge of the board. These pins are a physical interface between the Pi and the outside world. At the simplest level, you can think of them as switches that you can turn on or off (input) or that the Pi can turn on or off (output). Of the 40 pins, 26 are GPIO pins and the others are power or ground pins (plus two ID EEPROM pins which you should not play with unless you know your stuff!)

You can program the pins to interact in amazing ways with the real world. Inputs don't have to come from a physical switch; it could be input from a sensor or a signal from another computer or device, for example. The output can also do anything, from turning on an LED to sending a signal or data to another device. If the Raspberry Pi is on a network, you can control devices that are attached to it from anywhere and those devices can send data back. Connectivity and control of physical devices over the internet is a powerful and exciting thing, and the Raspberry Pi is ideal for this.

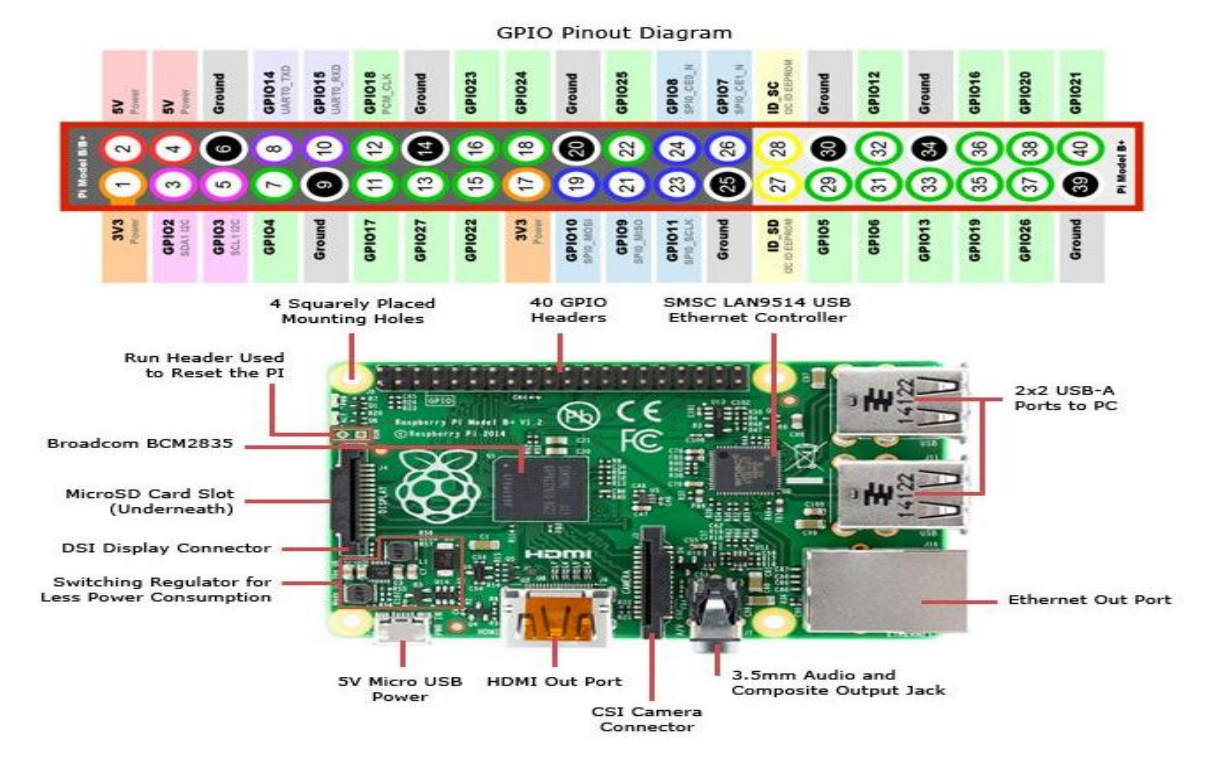
**LEDs:**

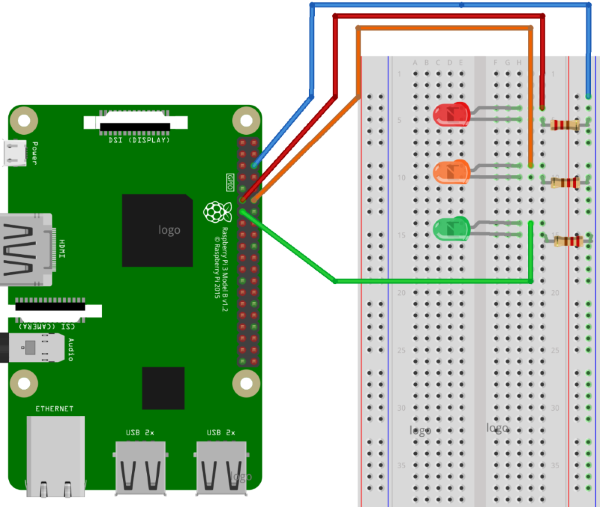
LEDs are the light emitting Diodes. LEDs have two wires. One wire is the anode (positive) and another is the cathode (negative). Push the LED leads into the breadboard, with the longer (positive) lead towards the top of the breadboard. It does not matter which way around the resistor goes.

**We Need:**

1. 3 LED lamps (green, orange, red)
2. 3 resistors
3. Jumper wires.
4. Breadboard.

**Interfacing Diagram**





**Python Code**

import RPi.GPIO as GPIO  
import time  
import signal  
import sys

# Setup  
GPIO.setmode(GPIO.BCM)  
GPIO.setup(9, GPIO.OUT)  
GPIO.setup(10, GPIO.OUT)  
GPIO.setup(11, GPIO.OUT)

# Turn off all lights when user ends demo  
def allLightsOff(signal, frame):  
 GPIO.output(9, False)  
 GPIO.output(10, False)  
 GPIO.output(11, False)  
 GPIO.cleanup()  
 sys.exit(0)signal.signal(signal.SIGINT, allLightsOff)

# Loop forever  
while True:   
 # Red   
 GPIO.output(9, True)   
 time.sleep(3) # Red and amber   
 GPIO.output(10, True)   
 time.sleep(1) # Green   
 GPIO.output(9, False)   
 GPIO.output(10, False)   
 GPIO.output(11, True)   
 time.sleep(5) # Amber   
 GPIO.output(11, False)   
 GPIO.output(10, True)   
 time.sleep(2) # Amber off (red comes on at top of loop)   
 GPIO.output(10, False)

**Conclusion**

Thus, we have successfully established an interface between Raspberry Pi and LEDs to control the operation of a hardware simulated traffic signal.