Experiment No: 8

Aim: Write an application using Raspberry Pi/Beagle Board to control the operation of a hardware simulated traffic signal.

Theory

Attaching the traffic lights

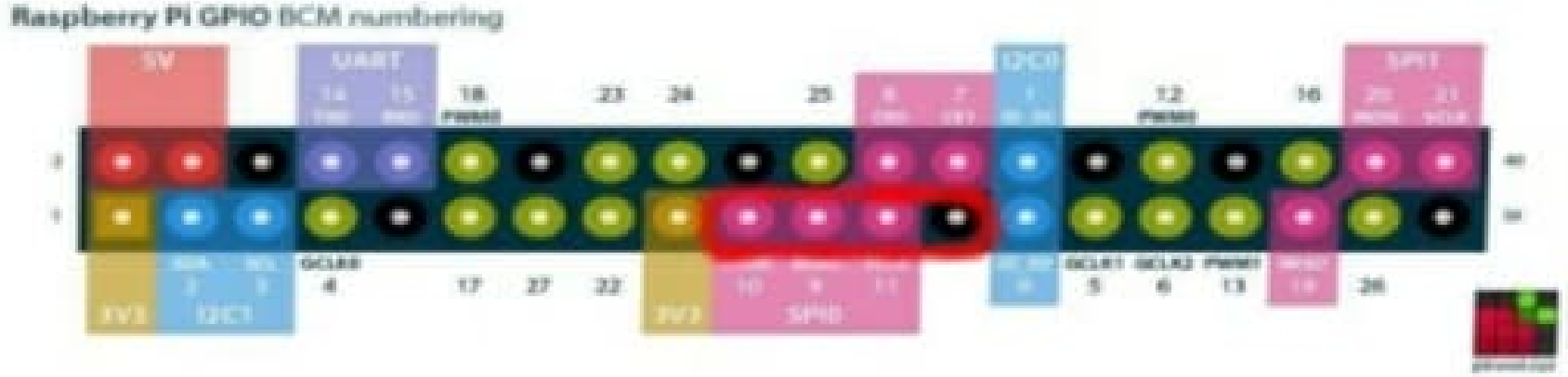
The low voltage labs traffic lights connect to the Pi using four pins. one of these needs to be grounded the other three being actual GPIO pins used to control each of the individual LED's.

fig

Before powering up the Pi, attach the traffic lights so that the pins connect to the GPIO pins highlighted in red.

Raspberry Pi GPIO BCM numbering:

Before powering up the Pi, attach the traffic lights so that the pins connect to the GPIO pins highlighted in red:



programming the traffic lights:

first, you need to install a couple to extra software packages needed to allow you to download my sample code & to give python access to the GPIO pins on the Pi. enter the foll" at the command line.

```
Sudo apt-get install python-dev python-rpi
gpio git
```

How it wouks:

The code for this is very simple. It starts by importing the Rpi.GPIO plus time which gives us a timed wait function, signal that allows us to trap the signal sent when the user tries to quit the program & sys so we can send an appropriate exit signal back to the operating system before terminating:

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

Next we put the GPIO library into "BCM" or "Broadcom" mode. & set pins 9 (red LED) 10 (amber LED) 11 (green LED) to be used as output.

```
# Setup

GPIO.setMode(GPIO.BCM)
GPIO.setup(9, GPIO.OUT)
GPIO.setup(10, GPIO.OUT)
GPIO.setup(11, GPIO.out)
```

The main part of the program will run in an infinite loop until the user exits it by stopping python with (ctrl) C. It's a good idea to add a handler fun" that will run whenever this happens, so that we can turn off all the lights prior to exits.

```
# 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)
```

When Control-C is pressed an interrupt signal SIGINT is sent. This is handled by all all lights off function that switches all the lights off, tidies up the GPIO library state & exits cleanly back to the operating system.

Conclusion:

Thus, we have implemented the application for traffic signal using Raspberry Pi.