

ASSIGNMENT 3

1. Write python code for blinking LED using Raspberry pi.

```
#Python module to control the GPIO interface on the Raspberry Pi.
import RPi.GPIO as GPIO
#provides timed wait function
import time
#allows us to trap the signal sent when the user tries to quit the
program
import signal
#to send an appropriate exit signal back to the operating system
before terminating.
import sys
#Broad-com mode to ease pin access
GPIO.setmode(GPIO.BCM)
GPIO.setup(9, GPIO.OUT)

#infinite loop for led blinking
while True:
    GPIO.output(9, True)
    time.sleep(1)
    GPIO.output(9, False)
    time.sleep(1)

# a handler function,we can turn off all the lights prior to exiting
#(when user intends to terminate)
def allLightsOff(signal, frame):
    GPIO.output(9, False)
    GPIO.cleanup()
```

```
sys.exit(0)
signal.signal(signal.SIGINT, allLightsOff)
```

2. Write python code for Traffic lights using Raspberry pi.

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

```
GPIO.setmode(GPIO.BCM)
#red LED
GPIO.setup(9, GPIO.OUT)
#amber LED
GPIO.setup(10, GPIO.OUT)
#green LED
GPIO.setup(11, GPIO.OUT)
```

```
#infinite loop which executes traffic light pattern
```

```
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
GPIO.output(10, False)

# a handler function, we can turn off all the lights prior to exiting
#(when user intends to terminate)
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)
```