Internet of things

Assignment -3

Write a python code for Blinking LED and traffic lights for raspberry Pi code

Blinking LED Code:

```
import RPi.GPIO      # RPi.GPIO can be referred as GPIO from now
import time


ledPin = 22    # pin22
def setup():
    GPIO.setmode(GPIO.BOARD)      # GPIO Numbering of Pins
    GPIO.setup(ledPin, GPIO.OUT)   # Set ledPin as output
    GPIO.output(ledPin, GPIO.LOW)  # Set ledPin to LOW


def loop():
    while True:
```

```
            print 'LED on'
            GPIO.output(ledPin, GPIO.HIGH)   # LED On
            time.sleep(1.0)                  # wait 1 sec
            print 'LED off'
            GPIO.output(ledPin, GPIO.LOW)    # LED Off
            time.sleep(1.0)                  # wait 1 sec
def endprogram():
      GPIO.output(ledPin, GPIO.LOW)     # LED Off
      GPIO.cleanup()                    # Release resources

if __name__ == '__main__':        # Program starts from here
      setup()
      try:
   loop()
      except KeyboardInterrupt:     # When 'Ctrl+C' is pressed, the
destroy() will be  executed.
            endprogram()
```

Internet of things

Traffic lights code:

```
import argparse

import os

import time

import RPi.GPIO as GPIO


if os.getuid() != 0:

    raise SystemExit('Expecting root privileges. Root privileges needed for GPIO pin usage.')


parser = argparse.ArgumentParser(description = 'Control Traffic Signal State')

parser.add_argument('color', choices = ['red', 'yellow', 'green'], help = 'Color of light')

parser.add_argument('state', choices = ['on', 'off', 'blink'], help = 'State of light')

parser.add_argument('--verbose', '-v', action = 'store_true')

args = parser.parse_args()


if args.verbose:

    print('Setting ' + args.color + " light to " + args.state + ".")
```

```
# Determine the GPIO pin
if args.color == 'red':
    pin = 8
elif args.color == 'yellow':
    pin = 10
elif args.color == 'green':
    pin = 12
else:
    raise SystemExit('Bad color specified.')

# Determine the state
if args.state == 'on':
    state = False
elif args.state == 'off':
    state = True
else:
    raise SystemExit('Bad state specified.')

# Set up the GPIO interface
```

Internet of things

```
GPIO.setmode(GPIO.BOARD)

GPIO.setwarnings(False)


# Trigger the GPIO output

GPIO.setup(pin, GPIO.OUT, initial=state)
```

Submitted by,

Amarjith

Arun kumar

Deepak

Joyal Prasanna