Traffic light

Dernière mise à jour du document : 06/01/2021 15:19

Description

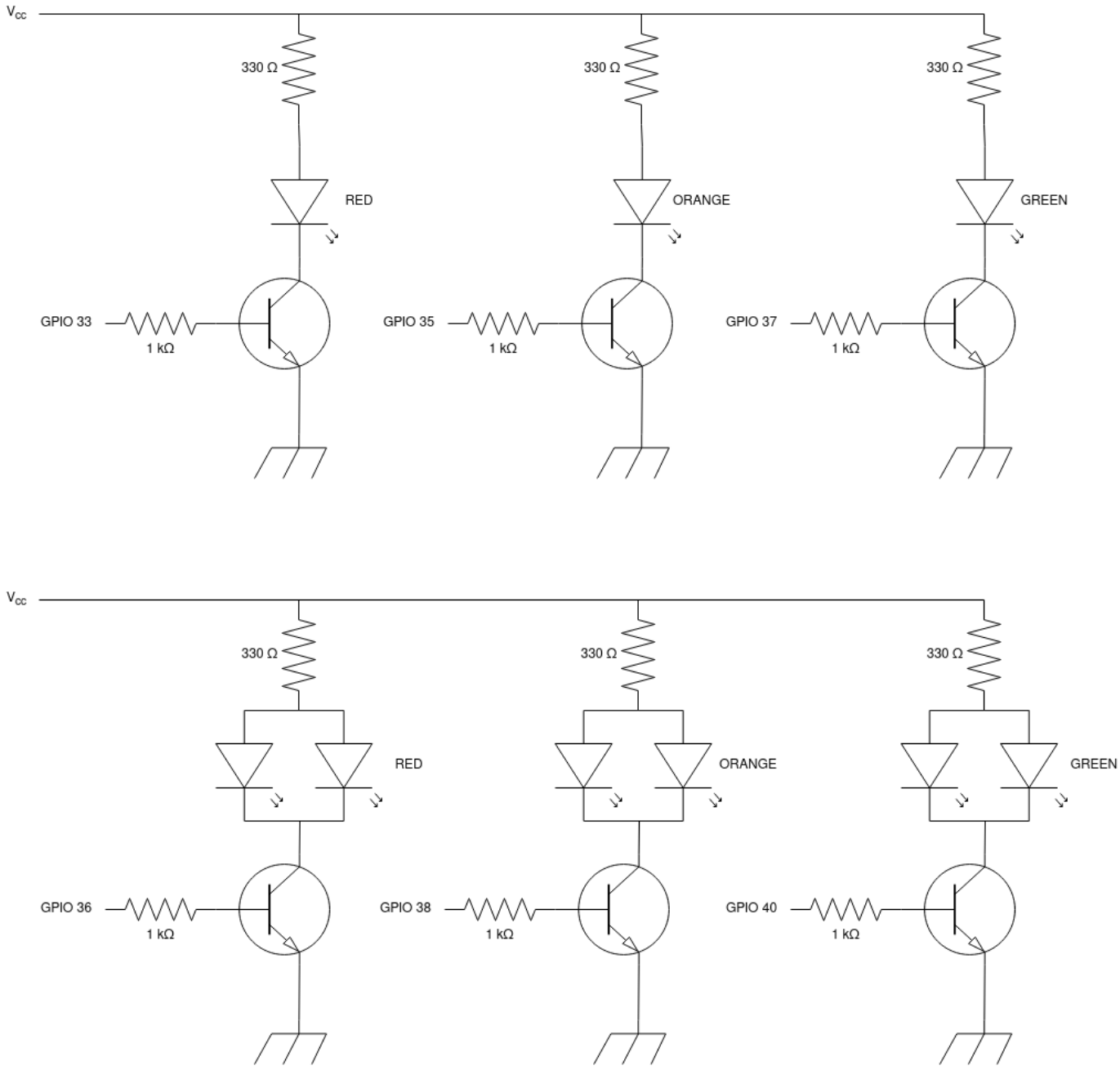
Les feux tricolore sont un des élément de la maquette. Ils sont au nombre de trois dans une intersection en T. Le programme permettant de gérer ces feux est exécuté à partir d'une Raspberry PI 3 Model B.

Setup

Setup Hardware

Eléments:

- 6 transistors NPN est P2N2222A
- 6 résistances 330 Ω
- 6 résistances 1 kΩ
- 3 feux tricolores (id: 1365434 sur https://fr.banggood.com/3Pcs-50mm-DIY-Model-3-Light-Traffic-Lights-Signal-Architecture-Street-Train-p-1365434.html?utm_source=googleshopping&utm_medium=cpc_organic&gmcCountry=FR&utm_content=minha&utm_campaign=minha-frg-fr-pc&currency=EUR&createTmp=1&utm_s&cur_warehouse=CN), 3V DC et 20mA (pour chaque LED), fil noir : anode, fils de couleurs : cathodes



Le feux tricolore "seul" est géré par les pins 33, 35 et 37 tandis que les feux "communs" sont gérés par les GPIO 36, 38 et 40.

Installation du code

Copiez et collez dans le répertoire /home/pi/Documents/Smartcity/Traffic_light les deux codes pythons:

1. Fichier principal /home/pi/Documents/Smartcity/Traffic_light/main.py

```
#!/usr/bin/env python3
#-- coding: utf-8 --

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

```
def exit(signum, stack):
    # Swith off traffic lights before exiting the program
    traffic_light1.switchOff()
    traffic_light2.switchOff()
    sys.exit(0) # Exit the program


def launchBasicCycle(signum=None, stack=None):
    # Lauch the cycle of traffic lights of a crossroads
    while 1:
        traffic_light1.toRedLight()
        time.sleep(2)
        traffic_light2.toGreenLight()
        time.sleep(10)

        traffic_light2.toRedLight()
        time.sleep(2)
        traffic_light1.toGreenLight()
        time.sleep(10)


def launchBlinkCycle(signum, stack):
    # Blink all traffic light to orange of a crossroads
    while 1:
        traffic_light1.switchOff()
        traffic_light2.switchOff()

        time.sleep(0.5)

        GPIO.output(traffic_light1.orangePin, GPIO.HIGH)
        GPIO.output(traffic_light2.orangePin, GPIO.HIGH)

        time.sleep(0.5)


if __name__ == '__main__':
    # GPIO init
    GPIO.setmode(GPIO.BOARD) # Enable 'board' mode
    GPIO.setwarnings(False) # Disable alert messages

    # Create traffic light objects with theirs pins (in this order :
```

```
red,orange,green)
    traffic_light1 = trafficlight.TrafficLight(33,35,37)
    traffic_light2 = trafficlight.TrafficLight(36,38,40)

    # Init signal for interruptions
    signal.signal(signal.SIGTERM, exit)                # To stop the
program by using SIGTERM signal :             sudo kill -TERM <pid>
    signal.signal(signal.SIGUSR1, launchBasicCycle)     # To call back the
basic cycle by using SIGUSR1 signal :    sudo kill -USR1 <pid>
    signal.signal(signal.SIGUSR2, launchBlinkCycle)     # To call the
orange blink cycleL by using SIGUSR2 signal : sudo kill -USR2 <pid>

    # Launch the basic cycle by default
    launchBasicCycle()
```

2. Module /home/pi/Documents/Smartcity/Traffic_light/trafficlight.py

```
import RPi.GPIO as GPIO
import time

class TrafficLight():
    """ This class permits the manipulation of a traffic light or two
opposite traffic lights """

    def __init__(self, redPin, orangePin, greenPin):
        # Set GPIOs pin number for each LEDs
        self.redPin = redPin
        self.orangePin = orangePin
        self.greenPin = greenPin

        # Enable GPIOs control as outputs
        GPIO.setup(redPin, GPIO.OUT)
        GPIO.setup(orangePin, GPIO.OUT)
        GPIO.setup(greenPin, GPIO.OUT)

        GPIO.output(self.orangePin, GPIO.HIGH) # By default, switch the
orange light on


    def toGreenLight(self):
        # The traffic light turns green
        self.switchOff()
        GPIO.output(self.greenPin, GPIO.HIGH)
```

```
    def toOrangeBlink(self, timeOrangeOn=0.5):
        # The traffic light flashes orange for 0.5s by default
        self.switchOff()
        GPIO.output(self.orangePin, GPIO.HIGH)
        time.sleep(timeOrangeOn)
        GPIO.output(self.orangePin, GPIO.LOW)


    def toRedLight(self, timeOrangeOn=2):
        # The traffic light turns red
        self.switchOff()
        self.toOrangeBlink(timeOrangeOn)
        GPIO.output(self.orangePin, GPIO.LOW)
        GPIO.output(self.redPin, GPIO.HIGH)


    def switchOff(self):
        # THe traffic light goes out
        GPIO.output(self.redPin, GPIO.LOW)
        GPIO.output(self.orangePin, GPIO.LOW)
        GPIO.output(self.greenPin, GPIO.LOW)
```

Création du service trafficlight.service

Copiez-collez les lignes suivantes dans /etc/systemd/system/trafficlight.service

```
[Unit]
Description=Traffic light service for Smartcity project

[Service]
ExecStart=python3 /home/pi/Documents/Smartcity/Traffic_light/main.py
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=trafficlight
```

Exécutez `sudo systemctl daemon-reload` pour redémarrer le gestionnaire de service.

Utilisation

Ci-dessous la liste des commandes pour intéragir avec les feux tricolore d'une intersection.

Démarrer le service qui éxécute le code :

`sudo systemctl start trafficlight.service`

Le PID du service s'obient en éxécutant :

`systemctl status trafficlight.service`

Faire clignoter les LED orange des feux:

`sudo kill -USR2 <pid>`

Reprendre le cycle normal des feux :

`sudo kill -USR1 <pid>`

Arrêter le service :

`sudo systemctl stop trafficlight.service`

OU

`sudo kill -TERM <pid>`