Okhaifo Pius Oikeh- 637258

Kagiko Chris Mukirae- 639622

# Generate Police Car Siren and Lights using Digital Circuitry and Raspberry Pi

A Project Report

Digital Electronics- APT 2030

Dr. Sylvester Namuye

USIU- Africa

## Table of Contents

## Abstract

The Purpose of this project is to make a digital circuit that would generate a police siren while mimicking the pattern of the police lights using Light Emitting Diodes (LEDs) in order to understand the digital circuitry used by the police for their police cars. This project can be taking further by applying it to home automation system with the use of sensors like smoke detector for generating siren that alert the owner about the existence of a problem. The objective of this project is to program the raspberry pi using python (a programming language) to interface with the other components of the circuit in other to output a siren noise and send signals to the lights in a pattern that resembles the police car lights. The project was carried out by assembling a speaker, a raspberry pi and eight LED’s on a breadboard using male-to-male wires; the LED’s and the speaker are connected to the raspberry pi which is coded to interface with the components and the program is executed to send outputs to the devices at certain time interfaces. Optionally a mobile application is used to automate the running of the code instead of constantly having to go back to the code to stop and start the python script from running. Though there were some setbacks (not getting the Integrated Circuit we required purchased in time before our project’s completion) the project was a success because a siren noise could be heard alongside the LED’s turning on and off like the police car light (police siren was made using digital circuitry).

# Introduction

A siren is a loud noise making device that can be used to provide warning, it is used for vehicles like police cars, ambulances and fire truck. This projects involves the connection of eight LED’s to the General Purpose Input and Output (GPIO) port of a T-cobbler which is connected to the raspberry pi, and the speaker to be connected the headphones jack of the raspberry pi. All the components are assembled on a breadboard using male-to-male wires. A program in the raspberry pi is use to manipulate the lights and the sound to produce a police siren and lighting combination in order to achieve the objectives of simulating the siren and lighting of a police car. The LED is a solid state device that converts an electric signal into a single colored light, when current flows from the anode to cathode the tiny light bulb device turns on. The raspberry pi is a card-sized single-board one core processor computer with a ram, an Input and Output (I/O) port, Universal Serial Bus (USB) hub, a headphones jack and an Ethernet port that is made with the intent of teaching the internal circuity of a computer chip and creating embedded systems, it uses python as its main programming language. The raspberry pi I/O port is connected to a T-cobbler. A T-cobbler is a 26 pin ribbon cable and an adapter board that helps in making connections between a raspberry pi and breadboard in order to do some prototyping. The corresponding interface of a T-cobbler are p0-SDA represented in figure 1.1 in the appendix. A speaker consists of an electromagnet, when a pulse of electricity passes through the coil of the electromagnet, the direction of its magnetic field is rapidly changed, the electromagnet is attached to a cone made of a flexible material such as paper or plastic which amplifies the vibrations, pumping sound waves into the surrounding air. A male-to-male wire is one with pins on the ends of either sides of the cable and its used to connect the components in a breadboard together. Previously an Integrated Circuit (IC) was used to generate the siren sound. This chip is a ROM IC that can generate multi siren tones simulating police sire, ambulance, fire brigade and machine gun sounds. It has 8 pins and uses low power of as low as 2.4 volts. The IC has inbuilt Oscillator and the frequency of oscillations is controlled by an external resistor connected to OSC 1 (pin 7) and OSC 2 (pin 2). A 220 K resistor will give satisfactory results. The oscillations thus generated will be then transferred to a control circuit which function based on the tone selection through the connections of SEL 1 (Pin 6) and SEL2 (Pin 1). The control circuit passes the signal to an address counter and then to the ROM. The tone pulses thus generated will be available from the output pin 3. Since the sound is weak, an amplifier is necessary to get loud sound. A single NPN transistor will amplify the sound. The pin configuration is shown in figure 1.2 in the appendix. The IC was not used for the project due to problems with purchasing.

## Project Objective

The project objective is to use the required tools and components specified in the requirements to successfully build a digital circuit that interface with a raspberry pi to output a police siren sound and mimic the lightings patterns used by a police car. Using a python code written in the raspberry pi to to simulate the the police car from the lights to the siren noise that it produces. The result of the project should be a digital circuit made in a lab environment that resembles a police car siren and lights. The frequency for a police siren is between the range of 635 hertz to 912 hertz

## Project Requirements

The apparatus/ tools used to conduct the experiment includes:

- 1 UM 3561 IC
- 1 Raspberry PI
- 2 White LEDs
- 4 Red LEDs
- 4 Blue LEDs
- 1 breadboard
- 20 males to male jumper cables
- 1 Computer
- 1 220k resistor
- 1 1k resistor
- 1 NPN Transistor
- 1 speaker
- A knowledge of Python programming language
- Optional mobile Application for automating on and off
- Optional Wi-Fi-dongle for access to WIFI using the raspberry

The lab work components that are required for this project include:

- Wire cutter
- 1 plier
- soldering iron
- solder

The experiment would be conducted in the lab therefore it is a physical setup: the components would be assembled in the lab using the resources available there. Its an experimental setup because the output of the python code used for interfacing needs to be observed to determine if it performs as expected. The circuit need to be assembled as shown in figure 2.1 below.

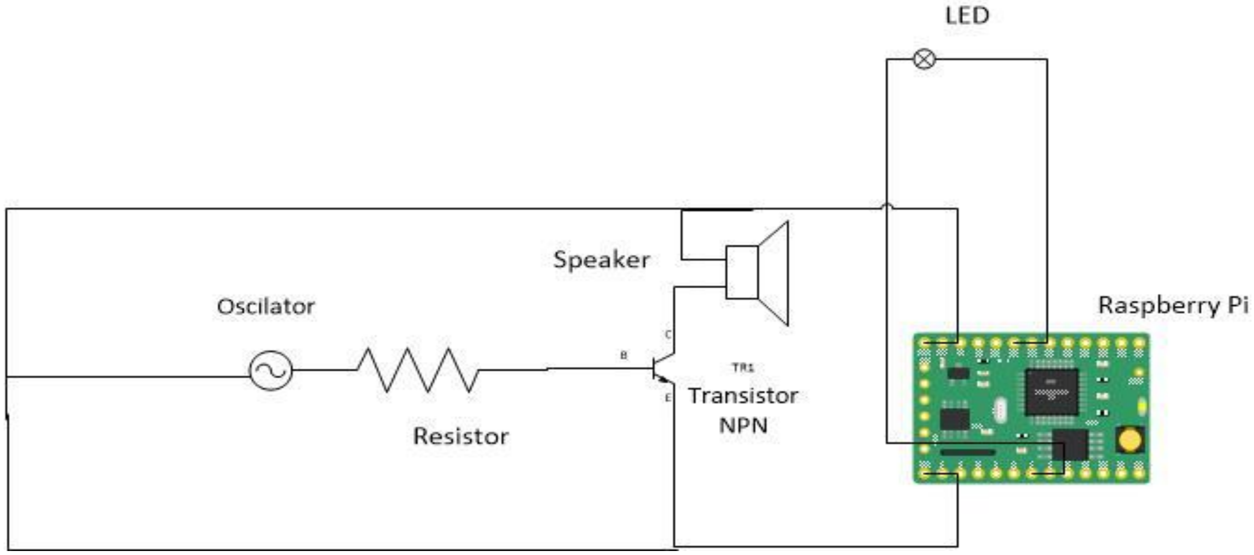


**figure 2 shows the circuit diagram of the project**

## Procedure

1. Prepare the workspace for building a circuit
2. Cut and strip the wires that require to be cut
3. Solder the wire using the soldering iron onto the speaker
4. Connect the T-cobbler ribbon to the raspberry Pi input and output Pins
5. Place the T-cobbler on the breadboard
6. Solder the wire using the soldering iron onto the speaker
7. Place the LEDs in this order from left to right blue, blue, white, red, red. Make 2 rows of this side by side

8. Use the male to male wires to connect the LED’s to the T-cobbler’s GPIO ports and Ground port
9. Place the IC on the breadboard
10. Place the transistors on the breadboard
11. Place resistors on the breadboard
12. Connect (on IC) VDD to GPIO, connect VSS to GND then connect OSC1 to OSC2 through a 220k resistor. Connect output to 1k resistor then resistor to base of the transistor. Connector collector to speaker and emitter to GND. Connect other speaker terminal to same GPIO as VCC.
13. Power on the raspberry pi
14. Download and Import the raspberry pi GPIO module
15. Connect Cathode end of LED to GPIO while anode end to GND. Make sure each column of LED is connected to separate GPIO for a variety of light patterns.
16. Write code that will send a “1” signal to all the connected GPOI.
17. Optionally program a mobile application to automate the on and off
18. Create a database that the python script would read data from
19. The circuit will look like the figure 3 below

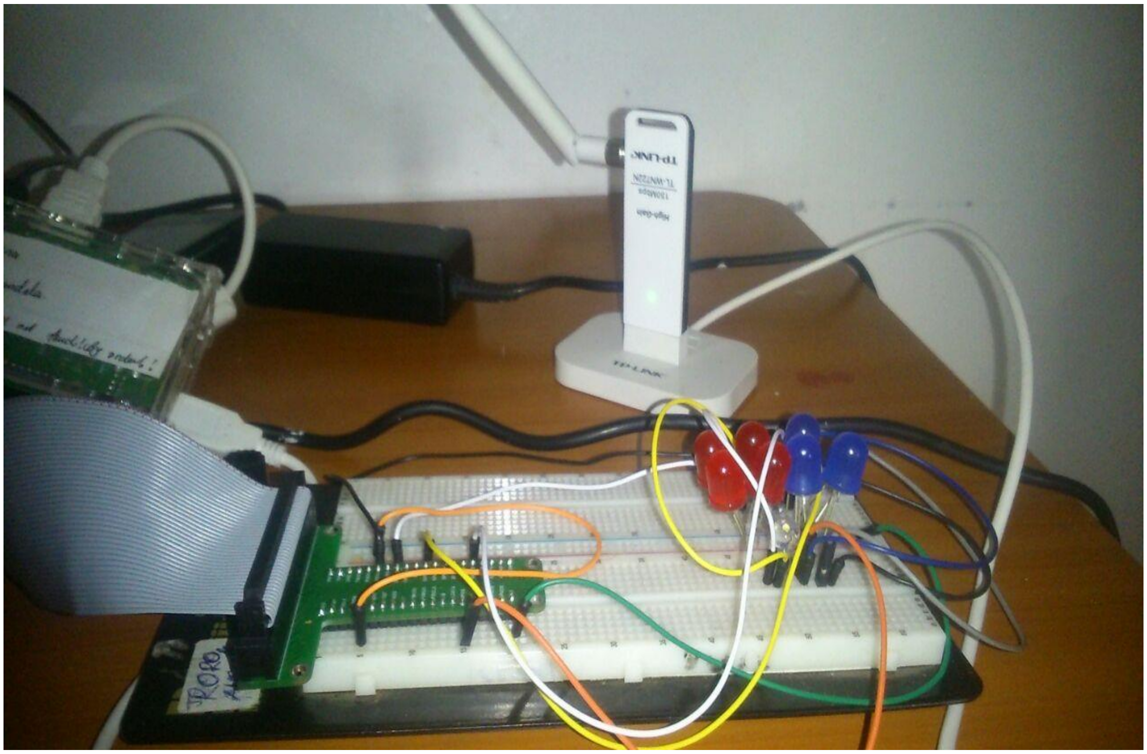


**Figure 3 above shows the circuit components assembled on the breadboard**

## Observations and results

The observations of the project are that when the python script in the raspberry pi was executing a siren noise could be heard from the speakers and the red, blue and white lights were changing according to that of a police car siren lights (the same pattern and speed of the siren light in police cars) and everything turns off when the mobile application off button is pressed. The observations and results can be seen below in figure 4.1, 4.2 and 4.3.

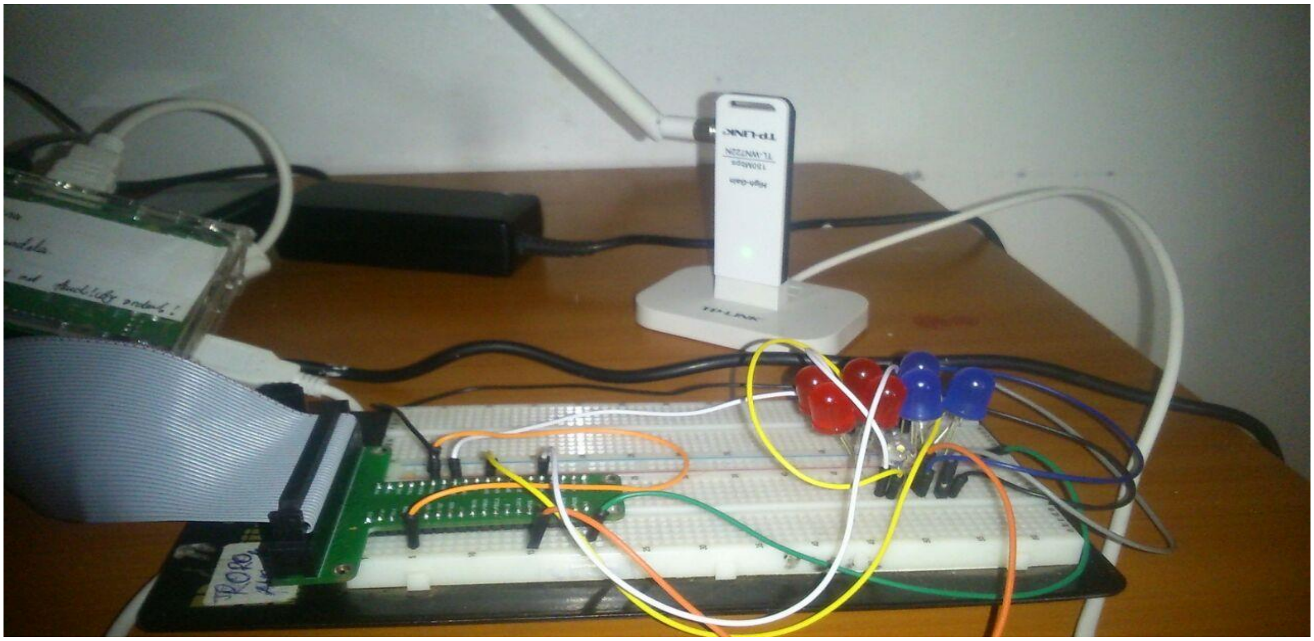


**Figure 4.0 above shows the circuit when the procedure has been turned off**

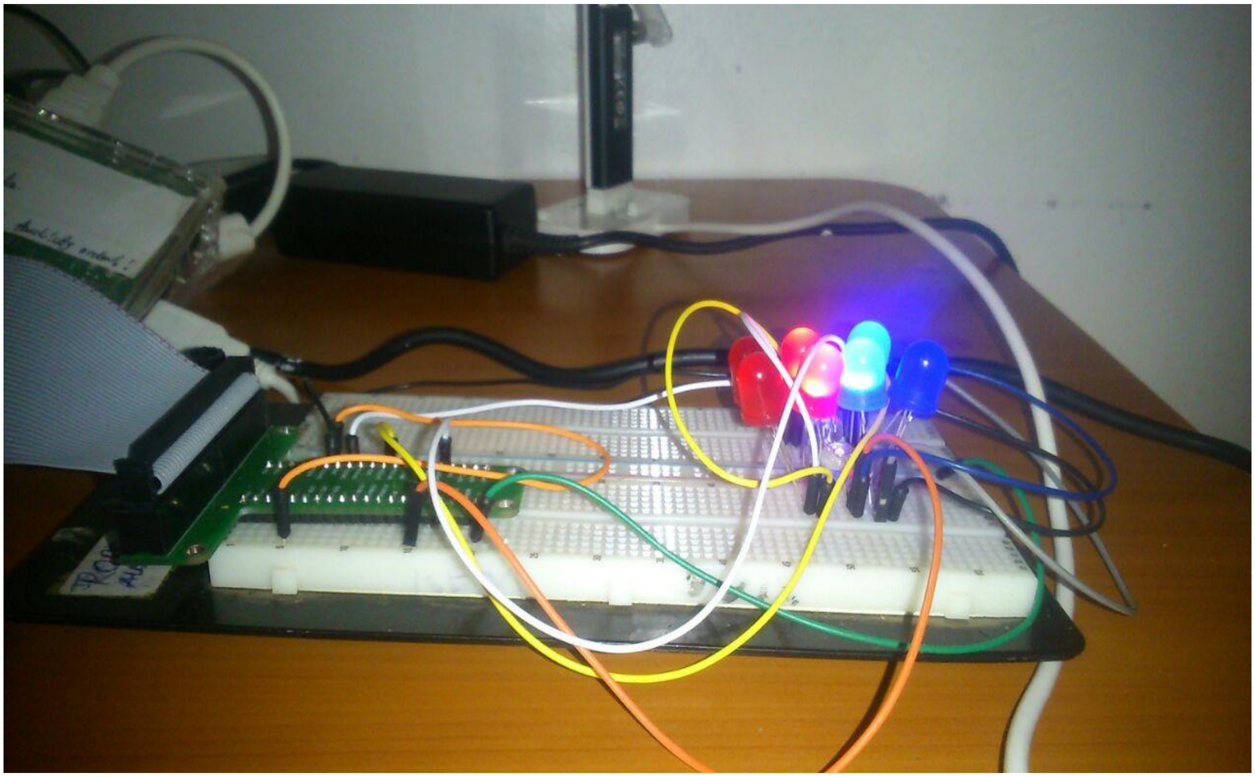


**Figure 4.1 above shows the circuit when the procedure has been turned on**

## Discussion/ Conclusion

The UM3561 IC that generates the sound of by outputting frequencies for police siren alongside, ambulance siren and fire truck was unable to be purchased in time before the project's completion therefore other ways had to be considered to replace the integrate circuit because it is a critical component in the circuit. An oscillator was tried in replacement but it frequencies that it generated was in kilohertz and the frequency for a police siren was between the range of 635 hertz to 912 hertz. Therefore, an audio wave file was used which was very tricky because the speakers was suppose to be connected to the IC therefore it had to be connected to the audio jack of the raspberry pi. Another problem that was encountered was that the python script had to play the code at the same time as it is sending signals to the LED's and stop the music when the python code stops executing. Finally, even after the modification to the system and code, the main objective was still attained and the sound was able to be played in collaboration with the lights. The project was successful and the police siren was made using a digital circuitry. The python program code in the raspberry pi is show in figure 5 in the appendix.

## Recommendations

The project would have been unsuccessful due to the lack of the critical component (the IC UM3561) therefore if the project specification was done earlier in the semester then we would have gotten all the components that we required purchased well in time for our project and if it were not possible we would have had enough time to come up with and implement another project.

# References

Mohankumar, D. "UM3561 Siren Generator Design". Electroschematics.com. N.p., 2016. Web. 7 Apr. 2016.

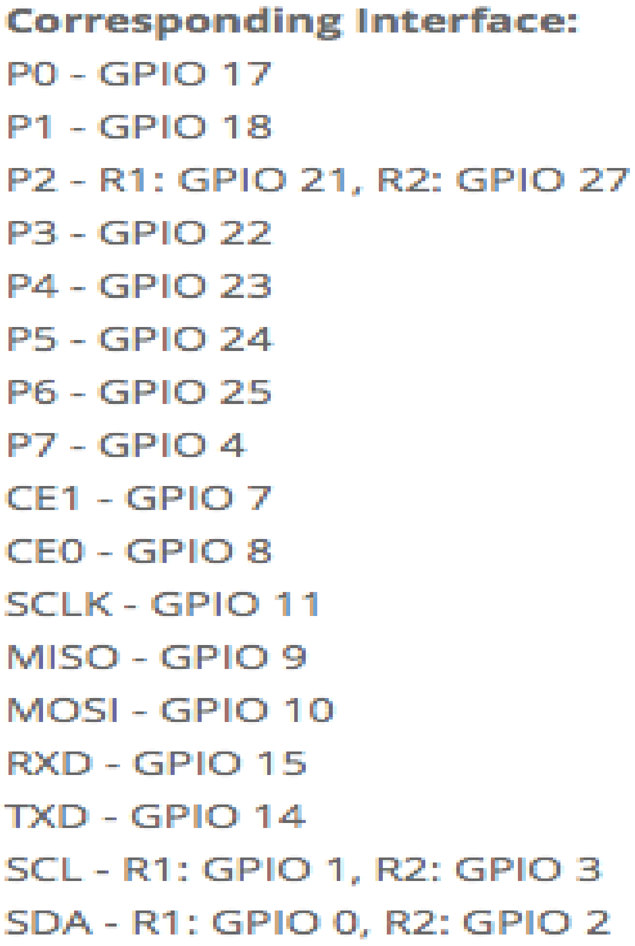
"Raspberry Pi GPIO Connector;". Elinux.org. Retrieved 10 Apr. 2016.

"Raspberry Pi USB Serial Connection and power supply". Elinux.org. Retrieved 2 April 13, 2016.

“T-cobbler Kit”. digitalmeans.co.uk/shop/breakout_kit_for_raspberry_pi_to_breadboard. Retrieved 2 April 13, 2016

Ramalho, L. (2014). *Fluent Python.* Califonia, United States: O'Reilly.

# Appendix



**Corresponding Interface:**

P0 - GPIO 17<br>
P1 - GPIO 18<br>
P2 - R1: GPIO 21, R2: GPIO 27<br>
P3 - GPIO 22<br>
P4 - GPIO 23<br>
P5 - GPIO 24<br>
P6 - GPIO 25<br>
P7 - GPIO 4<br>
CE1 - GPIO 7<br>
CE0 - GPIO 8<br>
SCLK - GPIO 11<br>
MISO - GPIO 9<br>
MOSI - GPIO 10<br>
RXD - GPIO 15<br>
TXD - GPIO 14<br>
SCL - R1: GPIO 1, R2: GPIO 3<br>
SDA - R1: GPIO 0, R2: GPIO 2

**Figure 1.1 shows the interface ports of a T-cobbler**

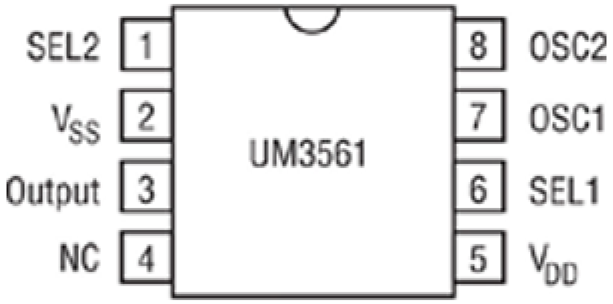


**Figure 1.2 above shows the pin configuration of the IC UM3561**

```
import urllib2

import threading

import time

import RPi.GPIO as GPIO

import os

import psutil


GPIO.setmode(GPIO.BCM)

GPIO.setwarnings(False)

GPIO.setup(25,GPIO.OUT)

GPIO.setup(18, GPIO.OUT)
GPIO.setup(23, GPIO.OUT)
GPIO.setup(12, GPIO.OUT)
GPIO.setup(5, GPIO.OUT)

def get():
    while 1:
      url="http://www.eboreimeoikeh.com/get.php"
      req = urllib2.Request(url)
      response = urllib2.urlopen(req)
      output = response.read()
      if (str(output) == "on"):
        for i in range(0,20):
             print "the blue light is on!"
             GPIO.output(12, GPIO.HIGH)
             GPIO.output(18, GPIO.HIGH)
             GPIO.output(5, GPIO.HIGH)
             time.sleep(0.05)
             GPIO.output(5, GPIO.LOW)
             GPIO.output(18, GPIO.LOW)
             GPIO.output(12, GPIO.LOW)
             GPIO.output(25, GPIO.HIGH)
             GPIO.output(23, GPIO.HIGH)
             time.sleep(0.05)
```

```
         print "the red light is on!"

         GPIO.output(23, GPIO.LOW)

         GPIO.output(25, GPIO.LOW)

         time.sleep(0.05)

         GPIO.output(23, GPIO.HIGH)

         GPIO.output(12, GPIO.HIGH)

         GPIO.output(18, GPIO.HIGH)

         GPIO.output(25, GPIO.HIGH)

    else:

       os.system("python king.py")

       for process in psutil.process_iter():

         if process.cmdline == ['python', 'playsound.py']:

              process.terminate()



a = threading.Thread(name='trip', target=get)

a.start()
```

**Figure 5.1 above shows the python code executed using the raspberry pi automated with mobile application**

```
import os

while 1:

os.system('omxplayer police_siren.mp3 &')
time.sleep(10)
```

**Figure 5.2 above shows the python code for playing the police siren mp3 file using the omxplayer module in python**

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

GPIO.setup(12,GPIO.OUT)
GPIO.setup(18, GPIO.OUT)
GPIO.setup(23, GPIO.OUT)
GPIO.setup(5, GPIO.OUT)
GPIO.setup(25, GPIO.OUT)
GPIO.output(18, GPIO.LOW)
GPIO.output(23, GPIO.LOW)

GPIO.output(25, GPIO.LOW)

GPIO.output(12, GPIO.LOW)

GPIO.output(5, GPIO.LOW)
```

**Figure 5.3 above shows the python code for switching of the lights using the interface with the GPIO ports of the T-cobbler**