University of Technology, Jamaica

Faculty of Engineering and Computing

School of Computing and Information Technology

Theory of Computation – CIT3006

Tutor – Mr. D. White

Group members:

Tenoy Genus – 1503263

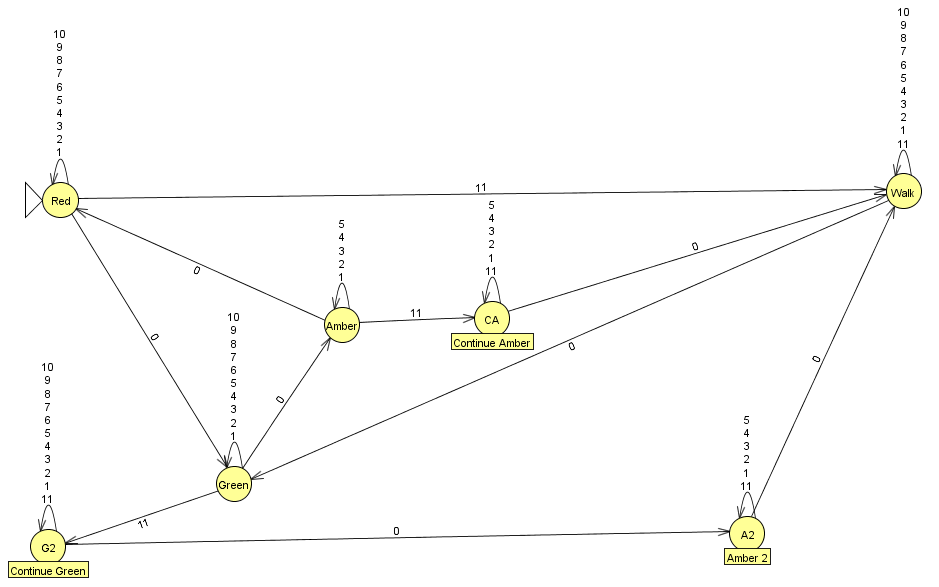
Jordan Murray – 1503454

Vernon Bailey – 1500365

Assignment: Project Stoplight

**Project report**

**Our State Diagram – DFA**



**Formal description**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| δ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Red | Red | Red | Red | Red | Red | Red | Red | Red | Red | Red | Green | walk |
| Green | Green | Green | Green | Green | Green | Green | Green | Green | Green | Green | Amber | Continue  Green |
| Continue  Green | Continue  Green | Continue  Green | Continue  Green | Continue  Green | Continue  Green | Continue  Green | Continue  Green | Continue  Green | Continue  Green | Continue  Green | Amber3 | Continue  Green |
| Amber | Amber | Amber | Amber | Amber | Amber | Red | - | - | - | - | - | Continue  Amber |
| Continue  Amber | Continue  Amber | Continue  Amber | Continue  Amber | Continue  Amber | Continue  Amber | Walk | - | - | - | - | - | Continue  Amber |
| Amber 2 | Amber2 | Amber2 | Amber2 | Amber2 | Amber2 | Walk | - | - | - | - | - | Amber2 |
| Walk | Walk | Walk | Walk | Walk | Walk | Walk | Walk | Walk | Walk | Walk | Green | Walk |

**Formal set description notation**

**Source Code**

from \_\_future\_\_ import print\_function

#Stop light code

import RPi.GPIO as GPIO

import time

import keyboard

import sys

def red(light, sec):

GPIO.output(light,True)

print("Red light is on")

while sec:

if keyboard.is\_pressed('enter'):

walk(4,17,10)

break

if sec < 1:

break

tformat = '{:02d}'.format(sec)

print(tformat, end='\r')

sys.stdout.flush()

time.sleep(1)

sec -= 1

turnOffLight(light,.1)

green(27, 10)

def green(light, sec):

GPIO.output(light,True)

print("Green light is on")

while sec:

if keyboard.is\_pressed('enter'):

continueGreen(sec)

break

if sec < 1:

break

tformat = '{:02d}'.format(sec)

print(tformat, end='\r')

sys.stdout.flush()

time.sleep(1)

sec -= 1

turnOffLight(light,.1)

amber(17,5)

def amber(light, sec):

GPIO.output(light,True)

print("Amber light is on")

while sec:

if keyboard.is\_pressed('enter'):

continueAmber(sec)

break

if sec < 1:

break

tformat = '{:02d}'.format(sec)

print(tformat, end='\r')

sys.stdout.flush()

time.sleep(1)

sec -= 1

turnOffLight(light,.1)

red(4,10)

def walk(light1, light2, sec):

GPIO.output(light1, True)

GPIO.output(light2, True)

print("It is safe to walk")

while sec:

tformat = '{:02d}'.format(sec)

print(tformat, end='\r')

sys.stdout.flush()

time.sleep(1)

sec -= 1

turnOffBothLights(light1,light2,.1)

green(27,10)

def continueGreen(sec):

print("Continuing green")

while sec:

tformat = '{:02d}'.format(sec)

print(tformat, end='\r')

sys.stdout.flush()

time.sleep(1)

sec -= 1

turnOffLight(27,.1)

amber2(17,5)

def continueAmber(sec):

print("Continuing amber")

while sec:

tformat = '{:02d}'.format(sec)

print(tformat, end='\r')

sys.stdout.flush()

time.sleep(1)

sec -= 1

turnOffLight(17,.1)

walk(4,17,10)

def amber2(light, sec):

GPIO.output(light,True)

print("In amber to go to walk")

while sec:

tformat = '{:02d}'.format(sec)

print(tformat, end='\r')

sys.stdout.flush()

time.sleep(1)

sec -= 1

turnOffLight(light,.1)

walk(4,17,10)

def turnOffLight(light, sec):

GPIO.output(light, False)

time.sleep(sec)

print("light is off\n")

return

def turnOffBothLights(light1, light2, sec):

GPIO.output(light1, False)

GPIO.output(light2, False)

time.sleep(sec)

print("Red and Amber lights are off\n")

return

def start():

red(4,10)

#Setting up the chip

GPIO.setmode(GPIO.BCM)

#setting up the pins via their number on the bread board

GPIO.setup(4, GPIO.OUT)

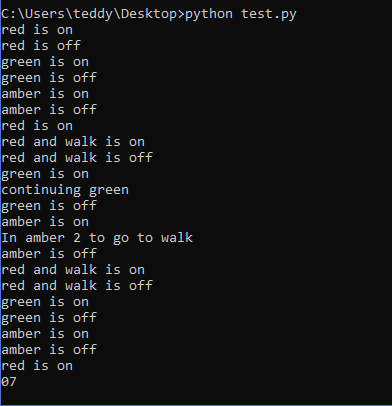
GPIO.setup(17, GPIO.OUT)

GPIO.setup(27, GPIO.OUT)

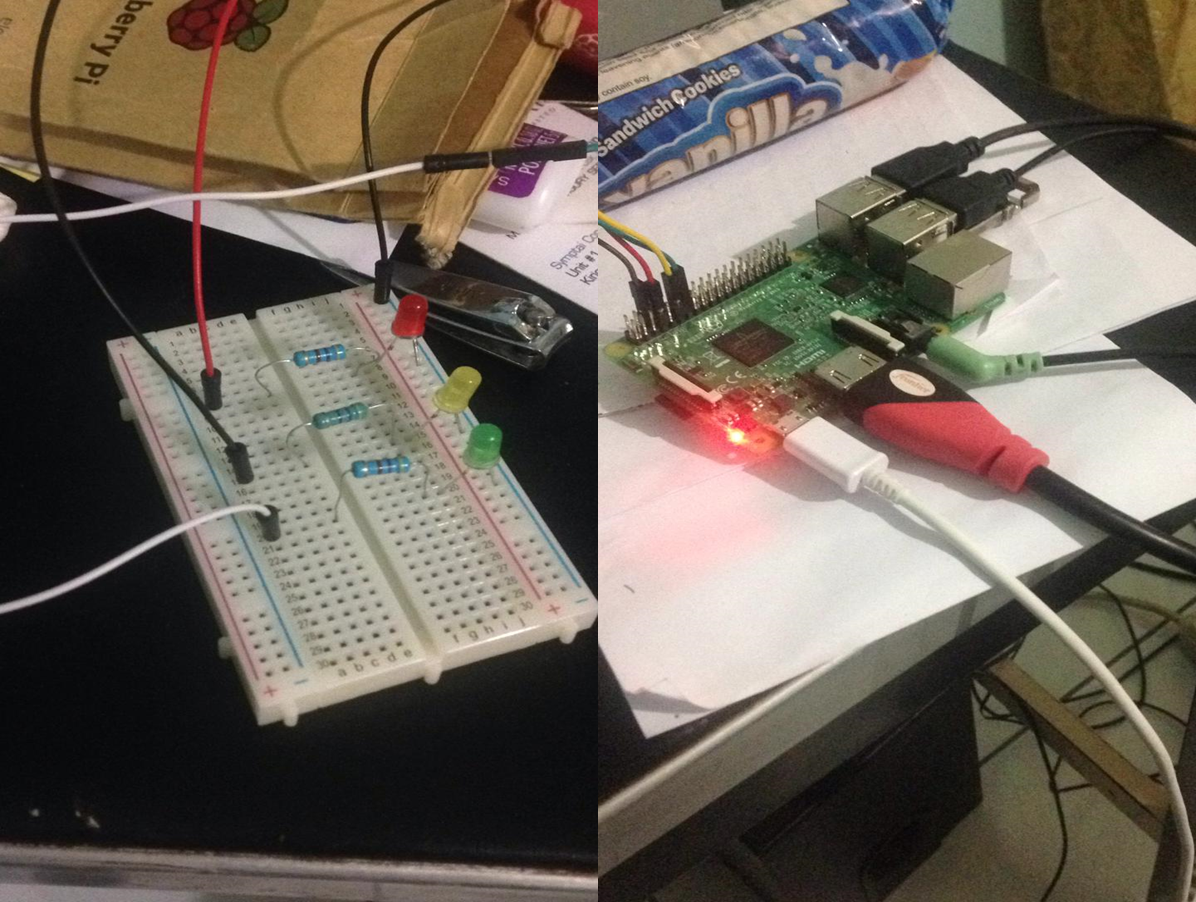
start()

GPIO.cleanup()

**Screen Shot of Sample Run**



**Diagram of set up**

****