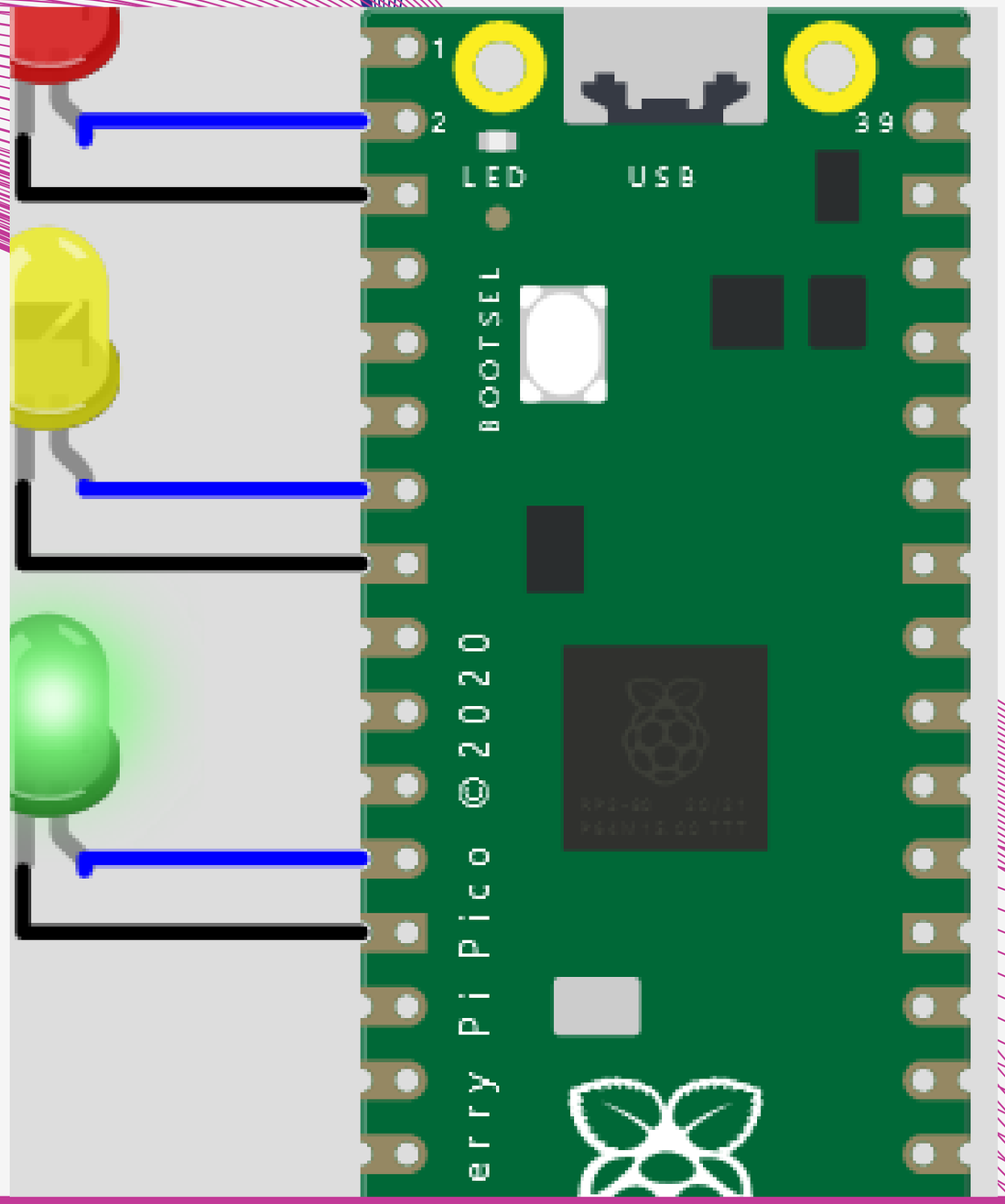
Traffic light using raspberry pi

presented by:-
Khushi Patil(A216

Table of CONTENTS



INTRODUCTION

Traffic-jam is a very big problem in developing cities, In fact it's ever increasing day-by-day nature makes it difficult to find where the traffic density is more in real time, so that to schedule a better traffic signal control and effective traffic routing. The root cause of this can be of different situations like congestion in traffic like insufficient Road width, Road conditions due to weather, unrestrained demand, large delay of Red Light etc. Indeed, manual control is must, Therefore, in order to reduce man's power, the need for simulating and optimizing traffic control to satisfy the increasing demand arises.

Components Used

01 Raspberry Pi

02 Traffic Lights

03 Resistors

04 Breadboard and jumper wires

COMPONENT DESCRIPTION

RASPBERRY PI

Raspberry Pi is a series of small, affordable, single-board computers that were originally developed by the Raspberry Pi Foundation in the United Kingdom. These computers were designed to promote computer science education and provide an affordable platform for experimentation and innovation.

TRAFFIC LIGHTS

Traffic lights, also known as traffic signals or stoplights, are a crucial part of modern transportation systems and road safety. They are used to regulate the flow of vehicular and pedestrian traffic at intersections, crosswalks, and other points where multiple roadways intersect.

RESISTORS

Resistors are passive electronic components that are fundamental to electrical and electronic circuits. They are used to limit the flow of electric current, control voltage levels, divide voltages, protect components, and perform various other functions in electronic and electrical systems.

BREADBOARD

A breadboard, also known as a protoboard or solderless breadboard, is a commonly used tool in electronics for building and testing circuits. It provides a convenient and temporary platform for connecting and prototyping electronic components without the need for soldering.

Working of project

Red, yellow and Green LEDs make the light come to life. A small push button on the top of the light turns the system on and displays the correct light for the time. A Raspberry Pi is the brains of the unit, and everything is connected with wires and a breadboard. Three LEDs are connected to the Raspberry Pi. Here LED red is connected to the pin 11,LED yellow is connected to pin 8 and LED green is connected to pin 5. if red light glow that means stop ,if yellow glow that means slow and if green glows it means go.here in this project ,i can set the timer red LED will glow for 5 seconds this means we have to stop for 5 sec then yellow will glow for 2 seconds that we have to go slow for 2 seconds then green lights glow for 5 seconds that we have to go for 5 sec then again yellow LED glow for 2 seconds that we can go slow for 2 seconds and then red glow again that means we have to stop again. like this traffic loght work.

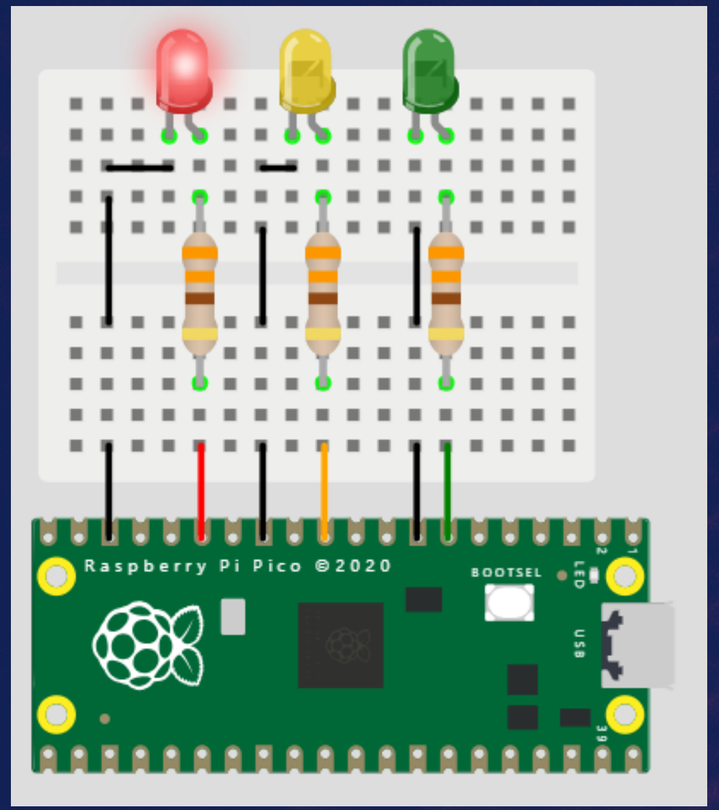
Coding

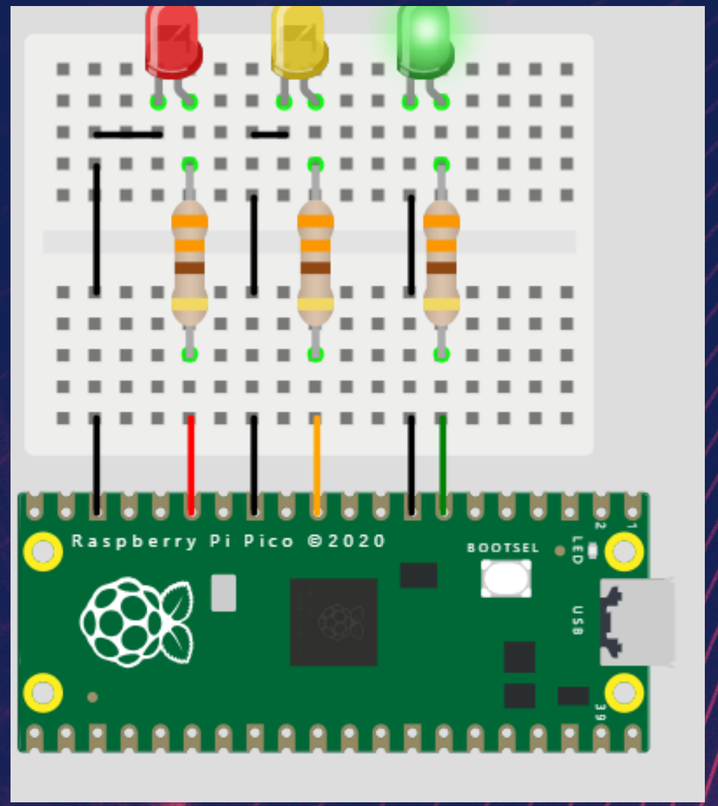
```
import machine
import utime

# Define the LED pins
led_red = machine.Pin(11, machine.Pin.OUT)
led_yellow = machine.Pin(8, machine.Pin.OUT)
led_green = machine.Pin(5, machine.Pin.OUT)

def handle_red_state():
  led_red.value(1)
  led_yellow.value(0)
  led_green.value(0)

def handle_yellow_state():
  led_red.value(0)
  led_yellow.value(1)
  led_green.value(0)
```

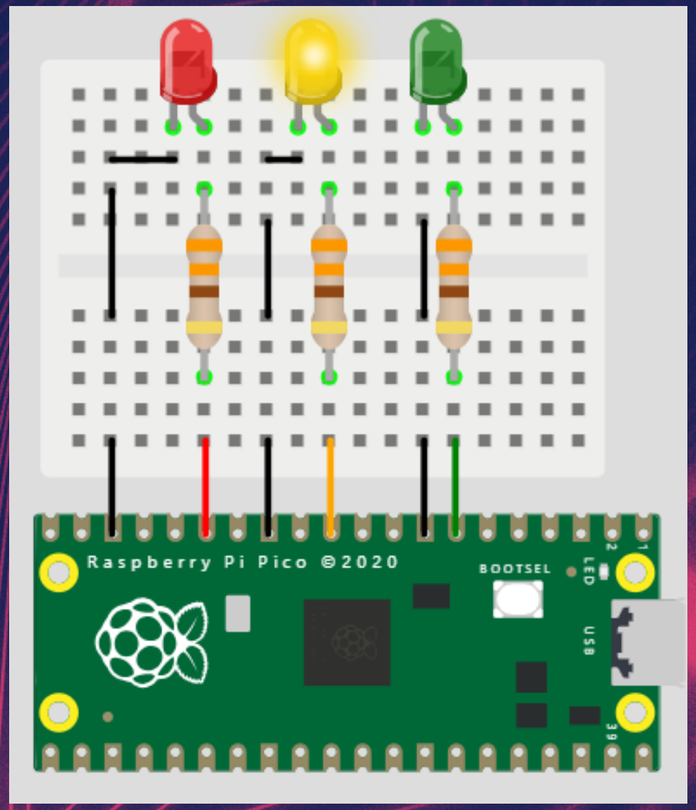
coding

```
                                ():
  led_red.value(0)
  led_yellow.value(0)
  led_green.value(1)

def handle_yellow_state_short():
  led_red.value(0)
  led_yellow.value(1)
  led_green.value(0)

# State handlers list
state_handlers = [
  # (state function, time in milliseconds)
  (handle_red_state,      5000), # Red LED, on for 5 seconds
  (handle_yellow_state,    3000), # Yellow LED, on for 3 seconds
  (handle_green_state,     5000), # Green LED, on for 5 seconds
  (handle_yellow_state_short, 2000) # Short Yellow LED, on for 2 seconds
]
def traffic_light():
  state = 0

  while True:
    # Get the current state tuple (handler function and sleep time)
    current_handler_and_time = state_handlers[state]
    handler_func = current_handler_and_time[0]
    sleep_duration_ms = current_handler_and_time[1]

    # Execute the handler function and sleep for the specified time
    handler_func()
    utime.sleep_ms(sleep_duration_ms)

    # Update the state index
    state = (state + 1) % len(state_handlers)

# Run the traffic light sequence
traffic_light()
```

Result


Raspberry Pi Pico ©2020
BOOTSEL
LED
USB


Raspberry Pi Pico ©2020
BOOTSEL
LED
USB


Raspberry Pi Pico ©2020
BOOTSEL
LED
USB

Conclusion

Based on the different vehicle count, the RASPBERRY PI takes decision and update the traffic light delays as a results. It will give green signal in high density traffic lane, that same time it shows red signal in opposite lane.

THANK YOU