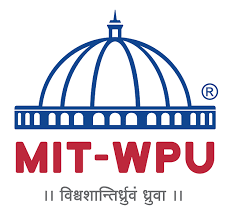
****

**M.TECH VLSI & Embedded Systems IIOT Practical Exam 2023-2024**

**Name: Akash Kumar**

**PRN: 1032233339**

**Subject: Industrial Internet of Things**

**Date: 11/12/23**

**Aim:** Traffic Light Control System

**Objectives:**

1. Build a cyber physical traffic light control system using arduino or nodemcu nodes at each intersection
2. Utilize nodemcu for communication between intersections and a raspberry pi as the centralized control unit. upload traffic data to thingspeak
3. Implement a relay based actuation system for traffic lights and an LED countdown timer

**CODE:**

#include <ESP8266WiFi.h>

#include <ThingSpeak.h>

// Replace with your network credentials

const char\* ssid = "MITWPU-CIAP";

const char\* password = "november2023";

// ThingSpeak Settings

unsigned long myChannelNumber = 2376023;

const char \* myWriteAPIKey = "TVSHQLX7B53DVYE3";

// Define the relay pins

int redRelay = D1;

int yellowRelay = D2;

int greenRelay = D3;

WiFiClient client;

void setup() {

// Connect to Wi-Fi

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(1000);

}

// Set the relay pins as output

pinMode(redRelay, OUTPUT);

pinMode(yellowRelay, OUTPUT);

pinMode(greenRelay, OUTPUT);

// Initialize ThingSpeak

ThingSpeak.begin(client);

}

void loop() {

// Control the traffic lights and send data to ThingSpeak

// Red light for 10 seconds

digitalWrite(redRelay, HIGH);

ThingSpeak.writeField(myChannelNumber, 1, 1, myWriteAPIKey); // Send '1' to ThingSpeak

delay(10000);

digitalWrite(redRelay, LOW);

// Green light for 10 seconds

digitalWrite(greenRelay, HIGH);

ThingSpeak.writeField(myChannelNumber, 2, 1, myWriteAPIKey); // Send '1' to ThingSpeak

delay(10000);

digitalWrite(greenRelay, LOW);

// Yellow light for 2 seconds

digitalWrite(yellowRelay, HIGH);

ThingSpeak.writeField(myChannelNumber, 3, 1, myWriteAPIKey); // Send '1' to ThingSpeak

delay(2000);

digitalWrite(yellowRelay, LOW);

}

**CODE for connection with raspberry PI:**

import socket

import requests

# Create a socket

s = socket.socket()

# Define the host and port

host = 'localhost'

port = 12345

# Bind the socket

s.bind((host, port))

# Listen for connections

s.listen(5)

while True:

# Accept a connection

c, addr = s.accept()

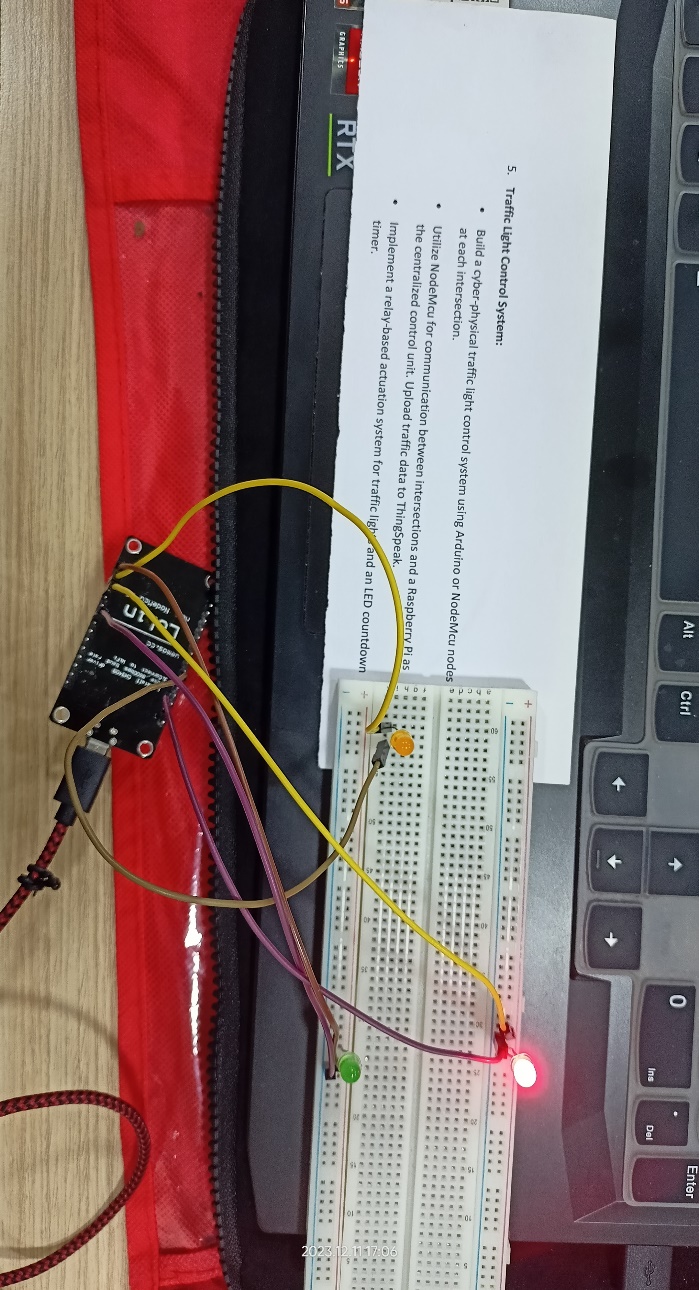
# Receive data from the NodeMCU

data = c.recv(1024)

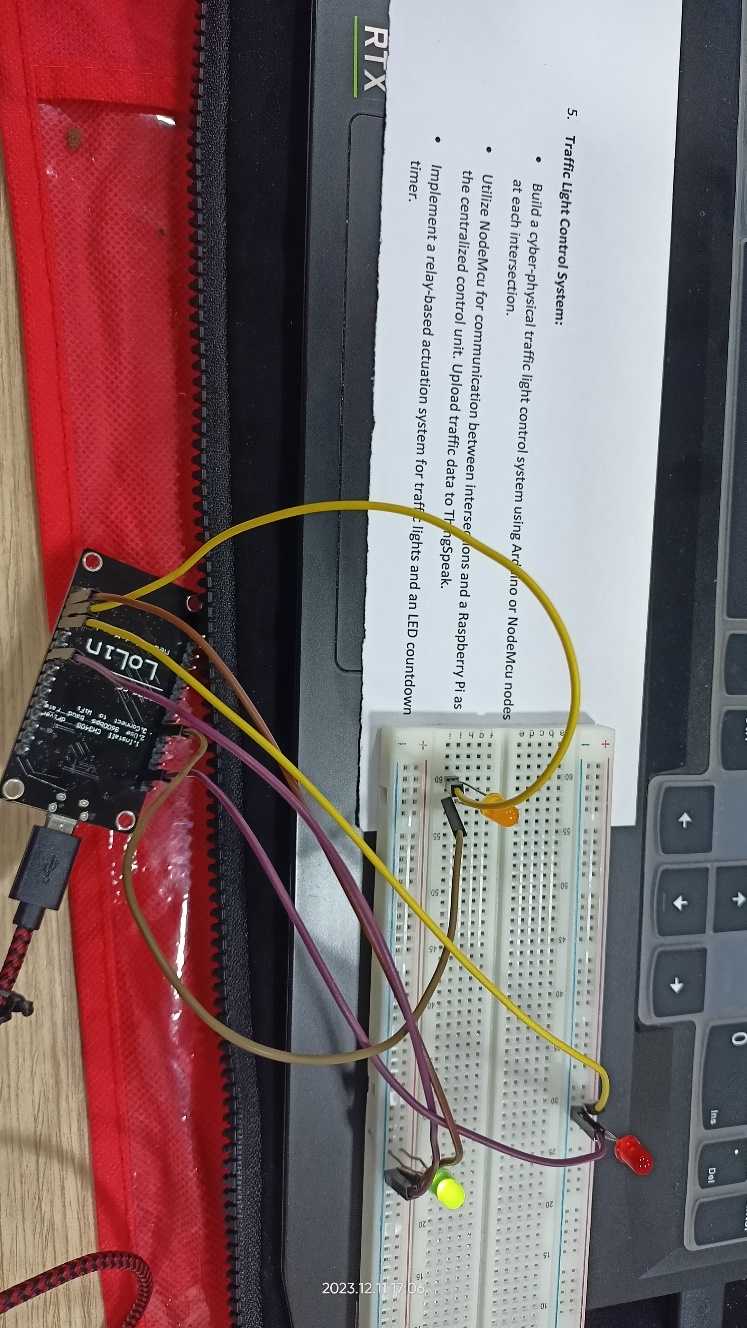
# Close the connection

c.close()

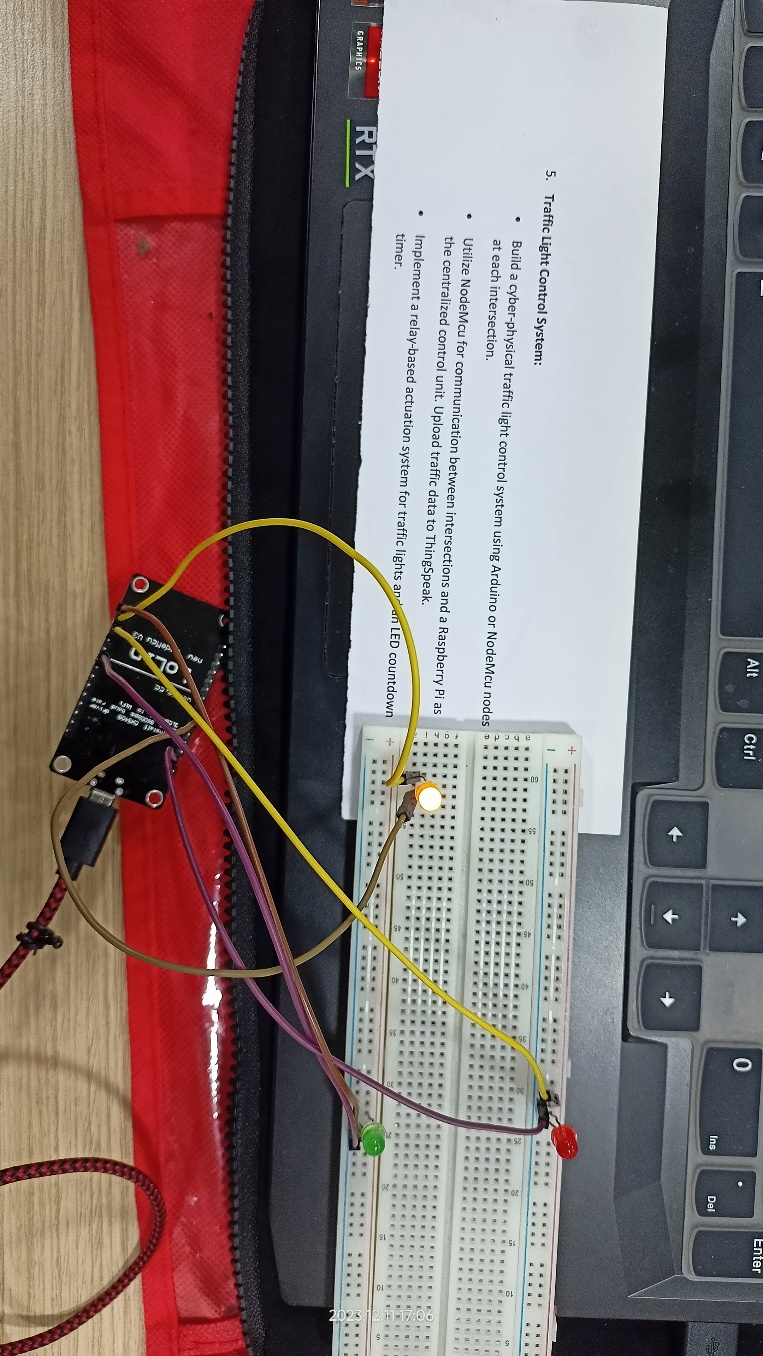
**Interfacing and Output:**

:

RED SIGNAL

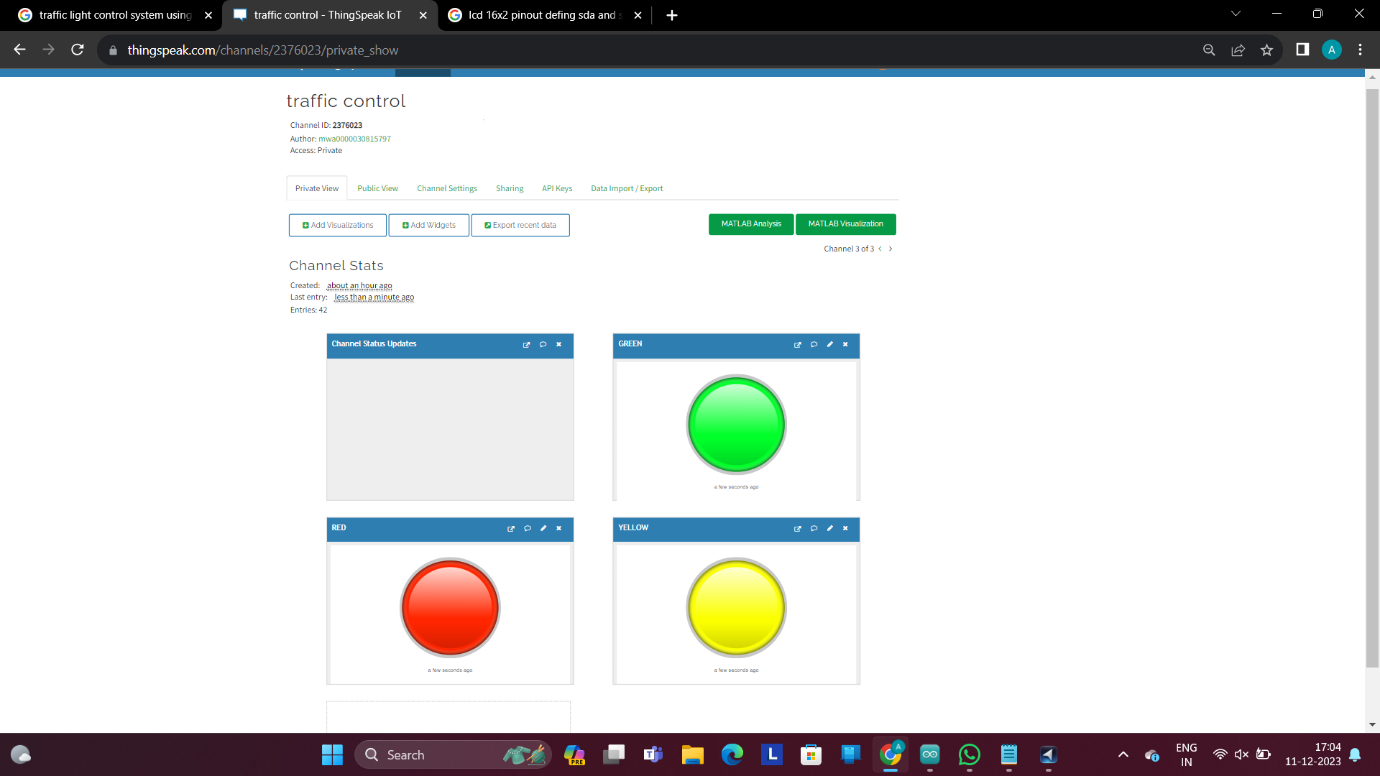


GREEN SIGNAL



YELLOW SIGNAL

**OUTPUT on Thingspeak:**



**Conclusion: I have successfully implemented 2 objectives from given statement and hence completed my practical exam.**