

SMART COLOR SENSOR MONITERING

PROGRAM :

```
#include<Servo.h>

#include<ESP8266WiFi.h>

#include<ThingSpeak.h>

const char * myWriteAPIKey = "8Q42D421CKET43S0";

unsigned long myChannelNumber = 3242;

const char *ssid = "Realme 11 Pro 5G"; // Enter your WiFi Name

const char *pass = ""; // Enter your WiFi Password

WiFiClient client;

Servo pickServo;

Servo dropServo;

const int s0 = 02; //D4;

const int s1 = 14; //D5;

const int s2 = 12; //D6;

const int s3 = 13; //D7;

const int out = 15; //D8;

int red = 0;

int green = 0;

int blue = 0;

int redcolor = 0;

int orangecolor = 0;

int greencolor = 0;

int yellowcolor = 0;

int CLOSE_ANGLE = 30; // The closing angle of the servo motor arm

int OPEN_ANGLE = 10; // The opening angle of the servo motor arm

void setup() {

  Serial.begin(9600);

  pinMode(s0, OUTPUT);

  pinMode(s1, OUTPUT);

  pinMode(s2, OUTPUT);
```

SMART COLOR SENSOR MONITERING

```
pinMode(s3, OUTPUT);

pinMode(out, INPUT);

digitalWrite(s0, HIGH);

digitalWrite(s1, HIGH);

pickServo.attach(04);//D2

dropServo.attach(00); //D3

pickServo.write(CLOSE_ANGLE); // Initialize the pickServo to closed position

dropServo.write(73); // Initialize the dropServo to default position

ThingSpeak.begin(client);

Serial.println("Connecting to WiFi...");

WiFi.begin(ssid, pass);

while (WiFi.status() != WL_CONNECTED) {

  delay(500);

  Serial.print(".");

}

Serial.println("");

Serial.println("WiFi connected");

}

void loop() {

  digitalWrite(s2, LOW);

  digitalWrite(s3, LOW);

  // Measure color intensity for red, green, and blue

  red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

  digitalWrite(s3, HIGH);

  blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

  digitalWrite(s2, HIGH);

  green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

  Serial.print("R Intensity: ");

  Serial.print(red);

  Serial.print(" G Intensity: ");

  Serial.print(green);
```

SMART COLOR SENSOR MONITERING

```
Serial.print(" B Intensity: ");

Serial.println(blue);

// Color detection based on intensity thresholds

if (red > 29 && red < 39 && green > 83 && green < 93 && blue > 69 && blue < 78) {

// Red detected

dropAndRecordColor(73, &redcolor, 1);

}

else if (green > 65 && green < 75 && blue > 60 && blue < 68) {

// Orange detected

dropAndRecordColor(107, &orangecolor, 2);

}

else if (red > 36 && red < 46 && green > 37 && green < 46) {

// Green detected

dropAndRecordColor(132, &greencolor, 3);

}

else if (red > 25 && red < 34 && green > 28 && green < 37 && blue > 43 && blue < 53) {

// Yellow detected

dropAndRecordColor(162, &yellowcolor, 4);

}

delay(1000);

}

void dropAndRecordColor(int servoAngle, int* colorCount, int fieldNumber) {

dropServo.write(servoAngle); // Drop the item

delay(700); // Wait for the item to drop

openServo(); // Open trap door to release the item

delay(200);

closeServo(); // Close the trap door

// Increment the count for the detected color

(*colorCount)++;

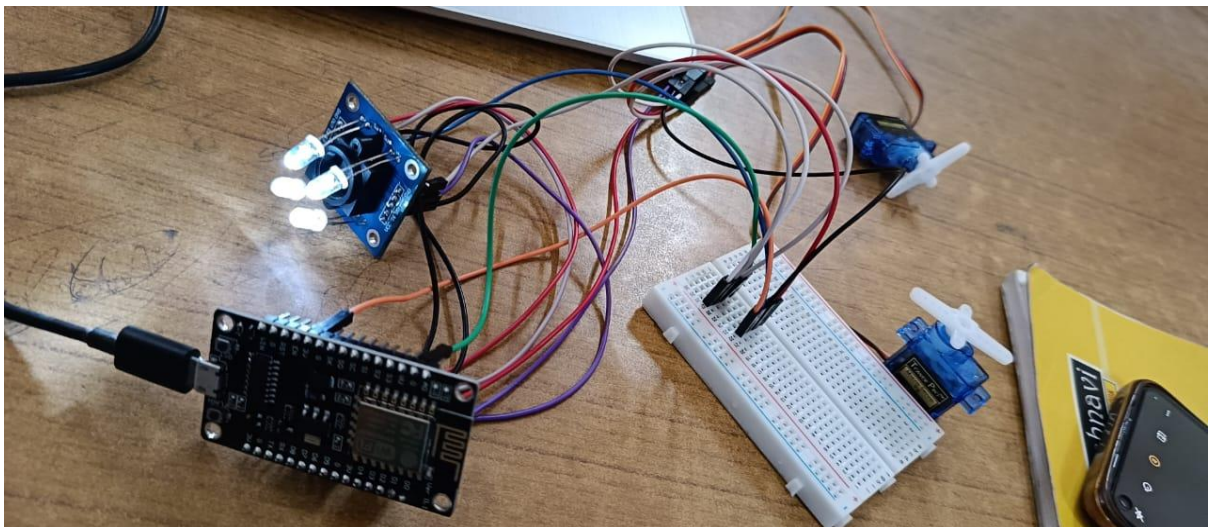
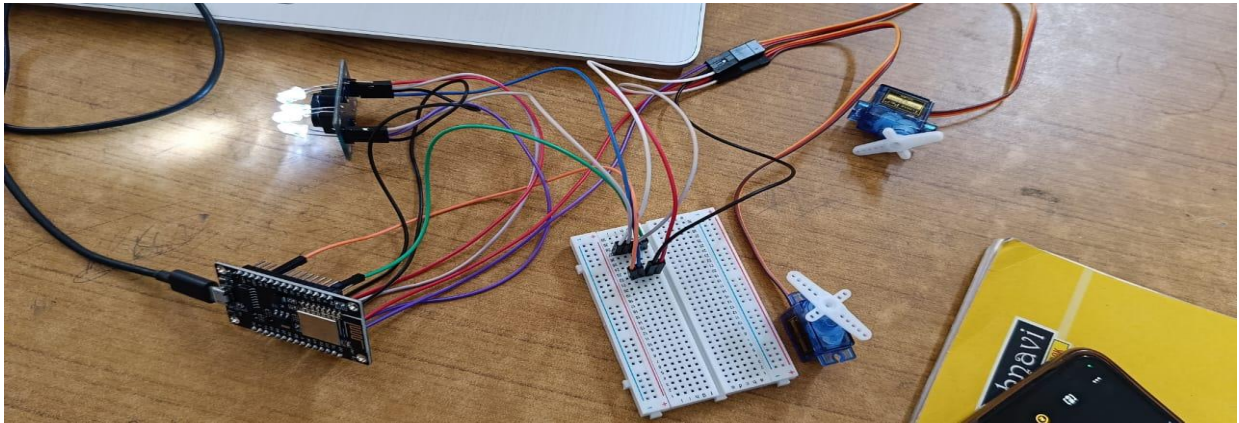
// Update ThingSpeak field with the color count

ThingSpeak.writeField(myChannelNumber, fieldNumber, *colorCount, myWriteAPIKey);
```

SMART COLOR SENSOR MONITERING

```
}  
  
void openServo() {  
  pickServo.write(OPEN_ANGLE); // Open the trap door  
}  
  
void closeServo() {  
  pickServo.write(CLOSE_ANGLE); // Close the trap door  
}
```

CIRCUIT:



SMART COLOR SENSOR MONITERING

OUTPUT :

