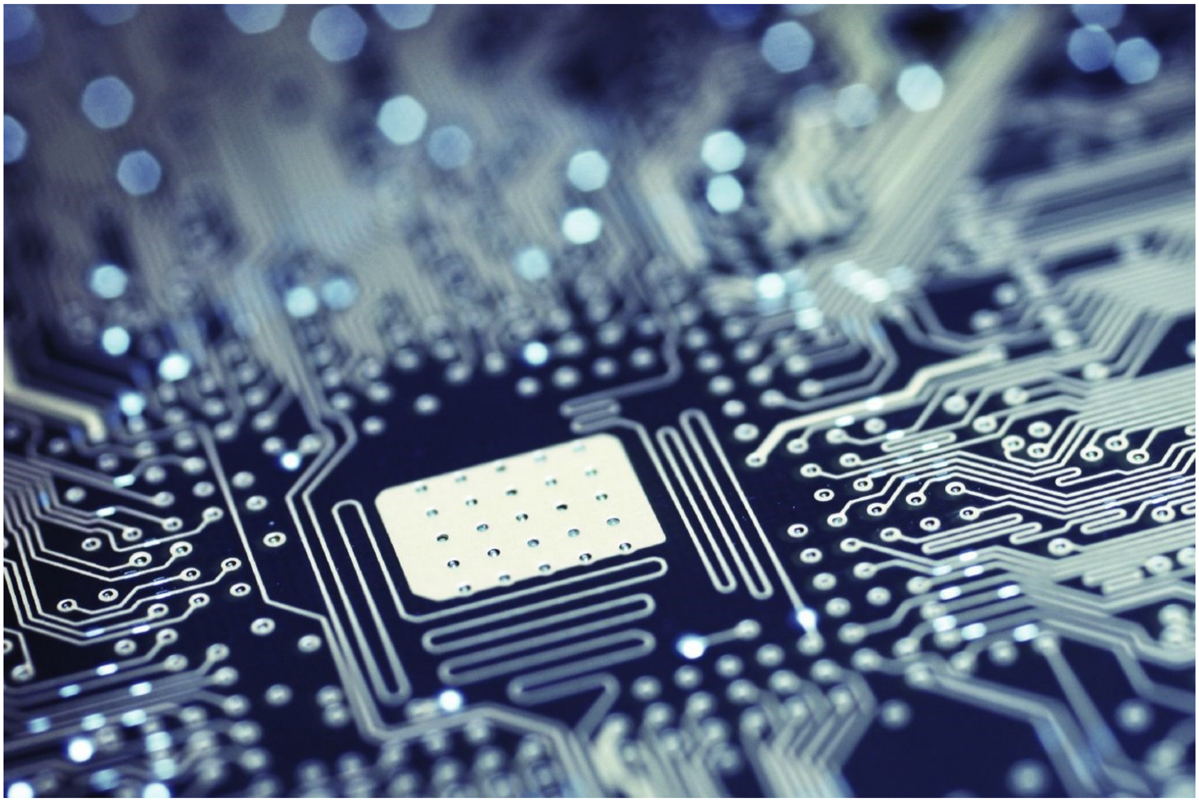


Integrated Color Sorter and Weight Detector

EMBEDDED SYSTEMS PROJECT

Name – Alkesh Shukla

Branch – Electronics & Communication Engineering

Indian Institute of Information Technology Sri City

Abstract

‘Colour sorting machines’ or ‘colour sorters’ are machines that detect the colours of objects before them, distinguish items by colour, and use mechanical or pneumatic ejection devices to transfer items whose colours are not within the acceptable range. The added weight detection feature provides additional information about objects and ensures weight limits aren’t exceeded, wherever applicable. Such machines may potentially be used in production lines in bulk food processing and other industries.

This report describes the implementation of this idea with a full functional block diagram that connects different sub-modules used in this project, along with the hardware as well as software components required for the same.

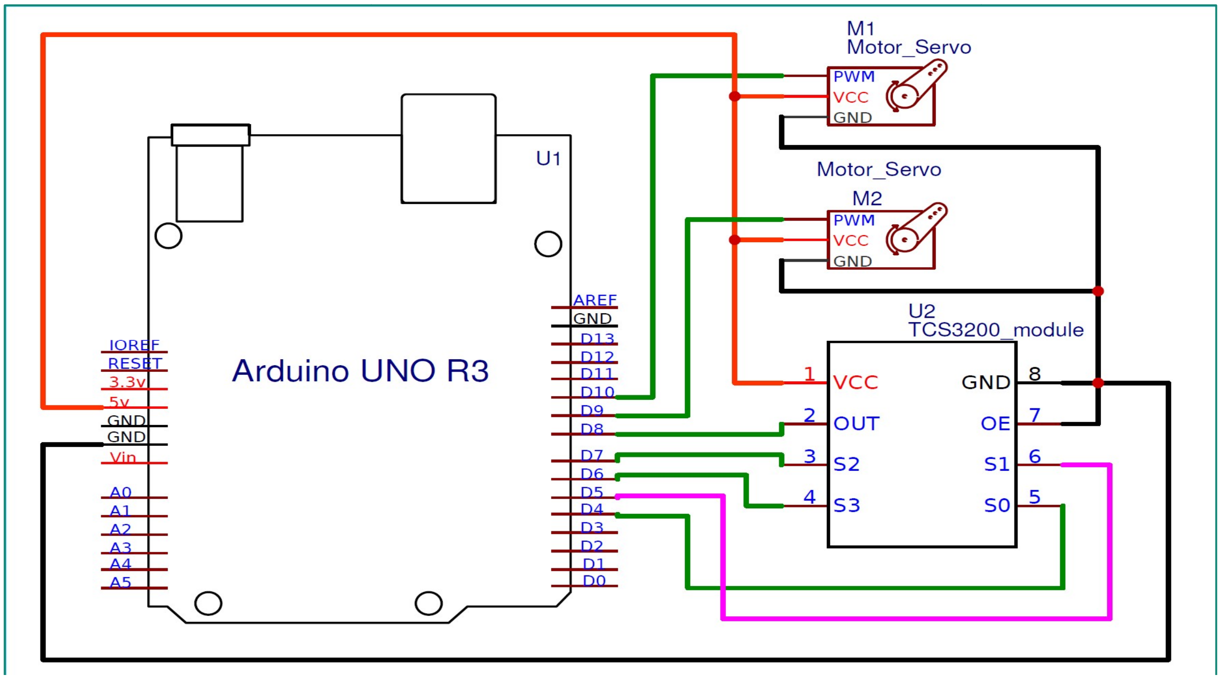
Justification

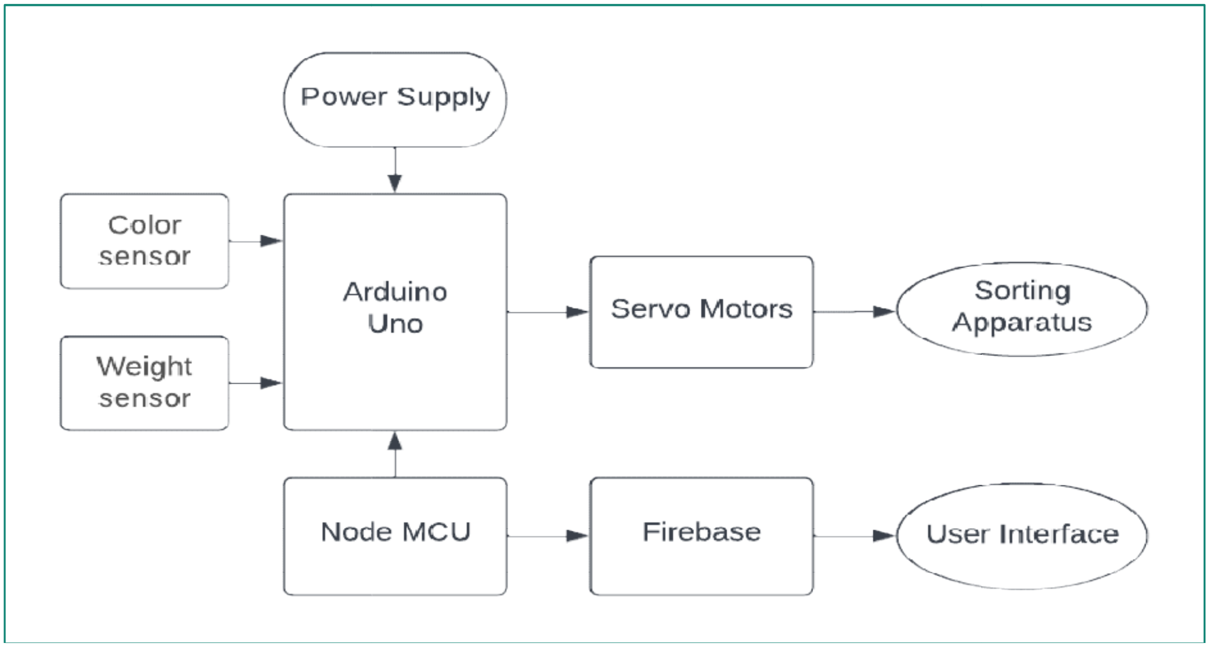
This machine may find an application in the agricultural industry in classifying grains. The rice sorting industry is the first big market. It can be used to separate different stones, black rice, etc. according to the color difference of the rice (hulled rice) material. Similarly, it can also be used to sort other coarse grains, such as wheat, corn, peanuts, various beans, sesame, etc.

Furthermore, color sorters may be used in the food processing industry, working with items such as coffee, nuts, oil crops, etc. The goal will be to separate discolored, toxic (such as ergot), unripe substances or husks that still have skins such as sunflower seeds. Compared with manual sorting, the machine will save labor and time, as it has high efficiency and low processing cost.

This prototype is highly scalable. Colour sorter integrated with a weight detection system can not only solve the needs of suppliers in the food industry and agricultural product processing industries, but also meet the needs of suppliers in several industrial fields, such as for recycling. Furthermore, an additional feature enabling the maintenance of objects sorted thus far via a Bluetooth module increases the usability of the sorter.

Functional Diagram connecting the sub-modules of the project:





Components Required:

- *Hardware:*
    - Arduino uno
    - NodeMCU
    - TCS3200 Colour Sensor
    - Servo Motors (2)
    - Weight Sensor
    - Battery
    - Sun board Sheets for Model
    - Jumper wires

- *Software:*
    - Embedded C
    - Arduino Uno IDE
    - Firebase
    - Android Studio
    - Kotlin/Java

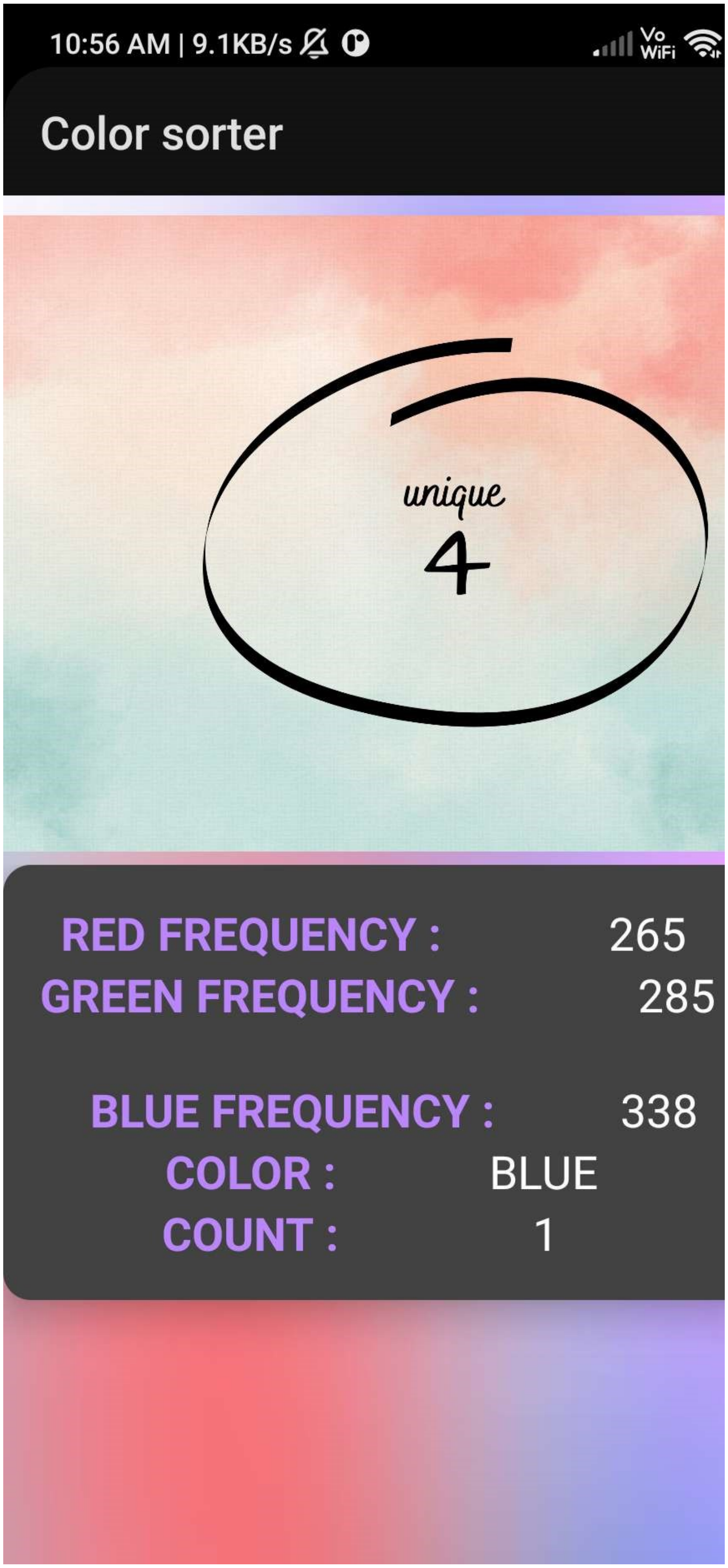
Phase Wise Implementation :

- *Phase-1: Output for pre-evaluation :*
- Color Using TCS3200 Color Sensor .
- Integration of Servo Motor 1 To Take the Colored Object To Color Sensor.
- Integration Of Servo Motor 2 To Rotate Based On Color.
- Counting The Number Of Colored Object Sorted In Particular Bucket

- *Phase-2: Output for final evaluation :*
- Color Using TCS3200 Color Sensor .
- Integration of Servo Motor 1 To Take the Colored Object To Color Sensor.
- Integration Of Servo Motor 2 To Rotate Based On Color.

- Counting The Number Of Colored Object Sorted In Particular Bucket.
- Integration Of The Weight Sensor To Detect The Weight Of colored Object.
- User Interface To Print The Value On Mobile App or Web.
- Complete Hardware Model Of Color And Weight Sorting Machine.

Outputs:



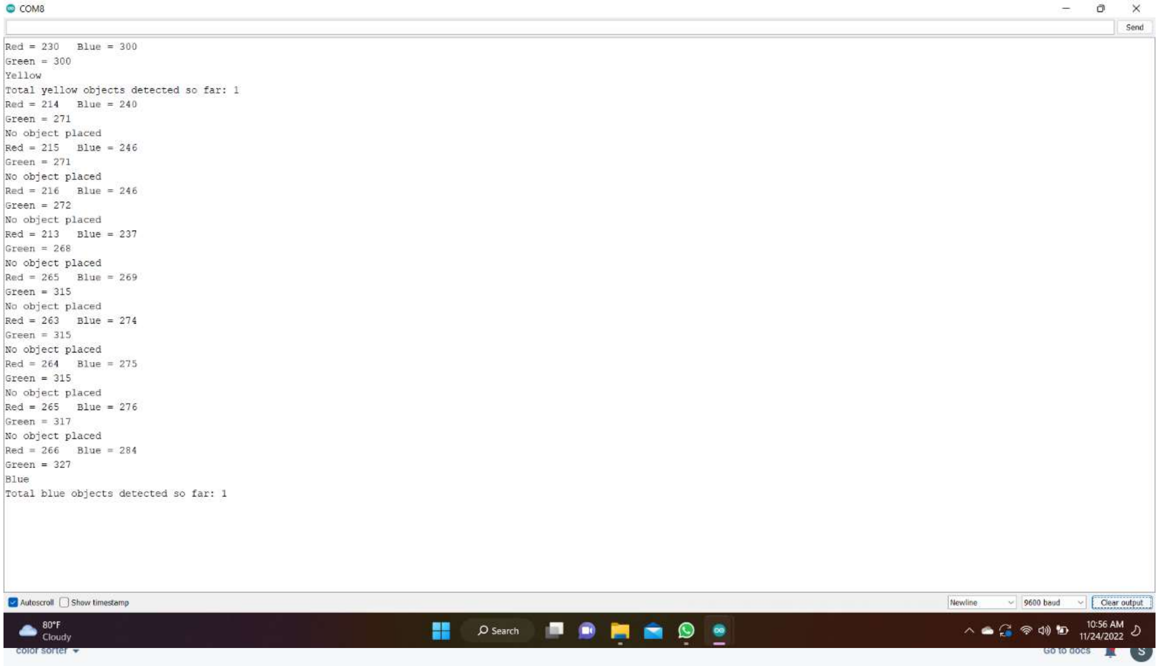
RED FREQUENCY : 265

GREEN FREQUENCY : 285

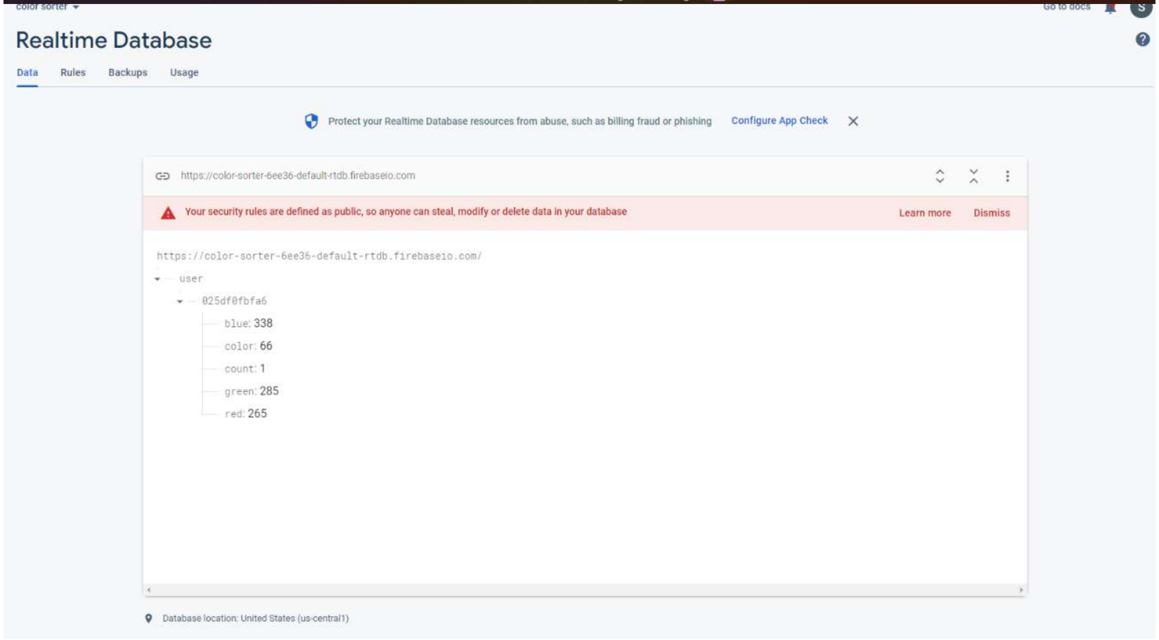
BLUE FREQUENCY : 338

COLOR : BLUE

COUNT : 1



```
Red = 230   Blue = 300
Green = 300
Yellow
Total yellow objects detected so far: 1
Red = 214   Blue = 240
Green = 271
No object placed
Red = 215   Blue = 246
Green = 271
No object placed
Red = 216   Blue = 246
Green = 272
No object placed
Red = 213   Blue = 237
Green = 268
No object placed
Red = 265   Blue = 269
Green = 315
No object placed
Red = 263   Blue = 274
Green = 315
No object placed
Red = 264   Blue = 275
Green = 315
No object placed
Red = 265   Blue = 276
Green = 317
No object placed
Red = 266   Blue = 284
Green = 327
Blue
Total blue objects detected so far: 1
```



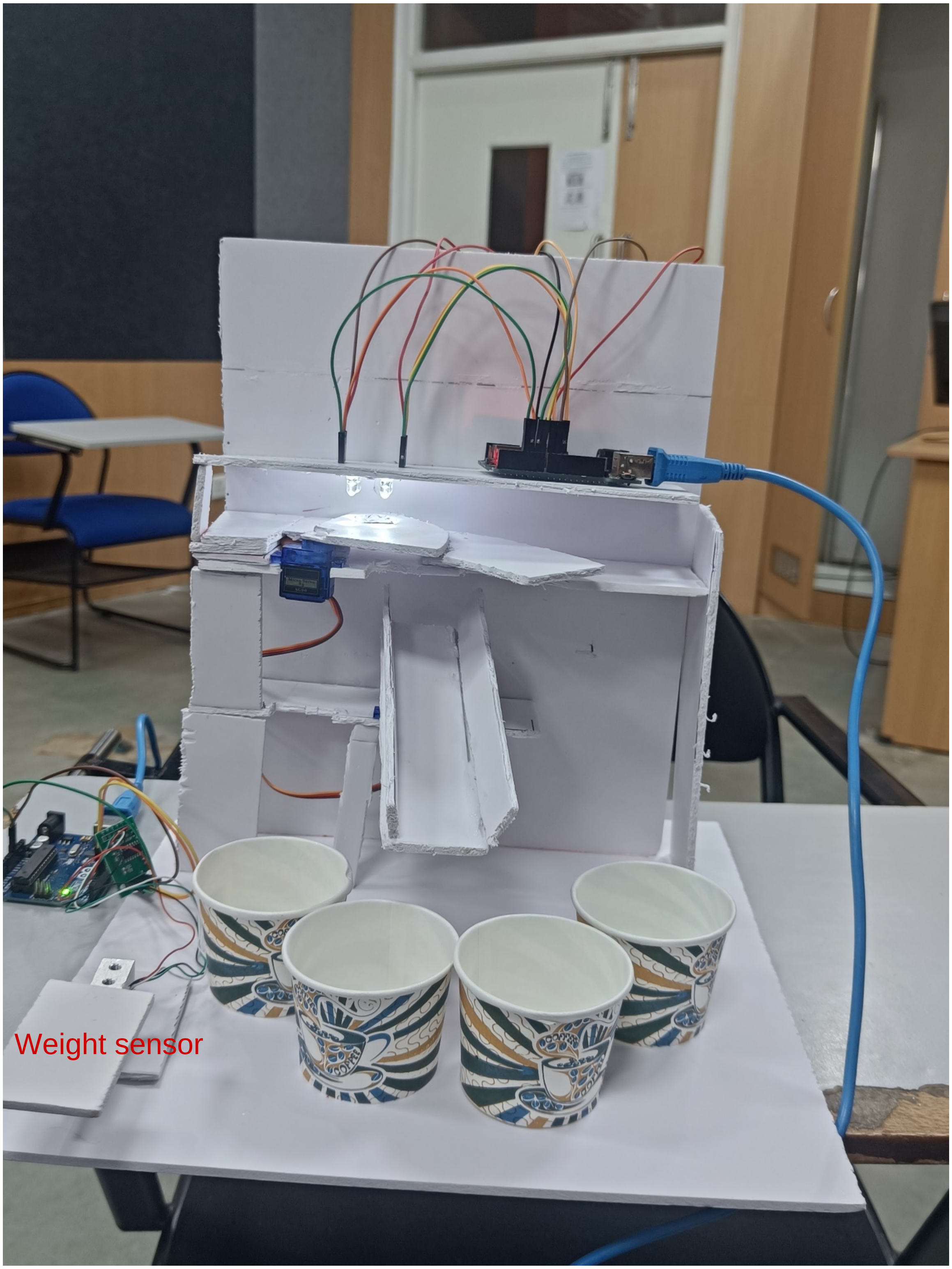
Protect your Realtime Database resources from abuse, such as billing fraud or phishing

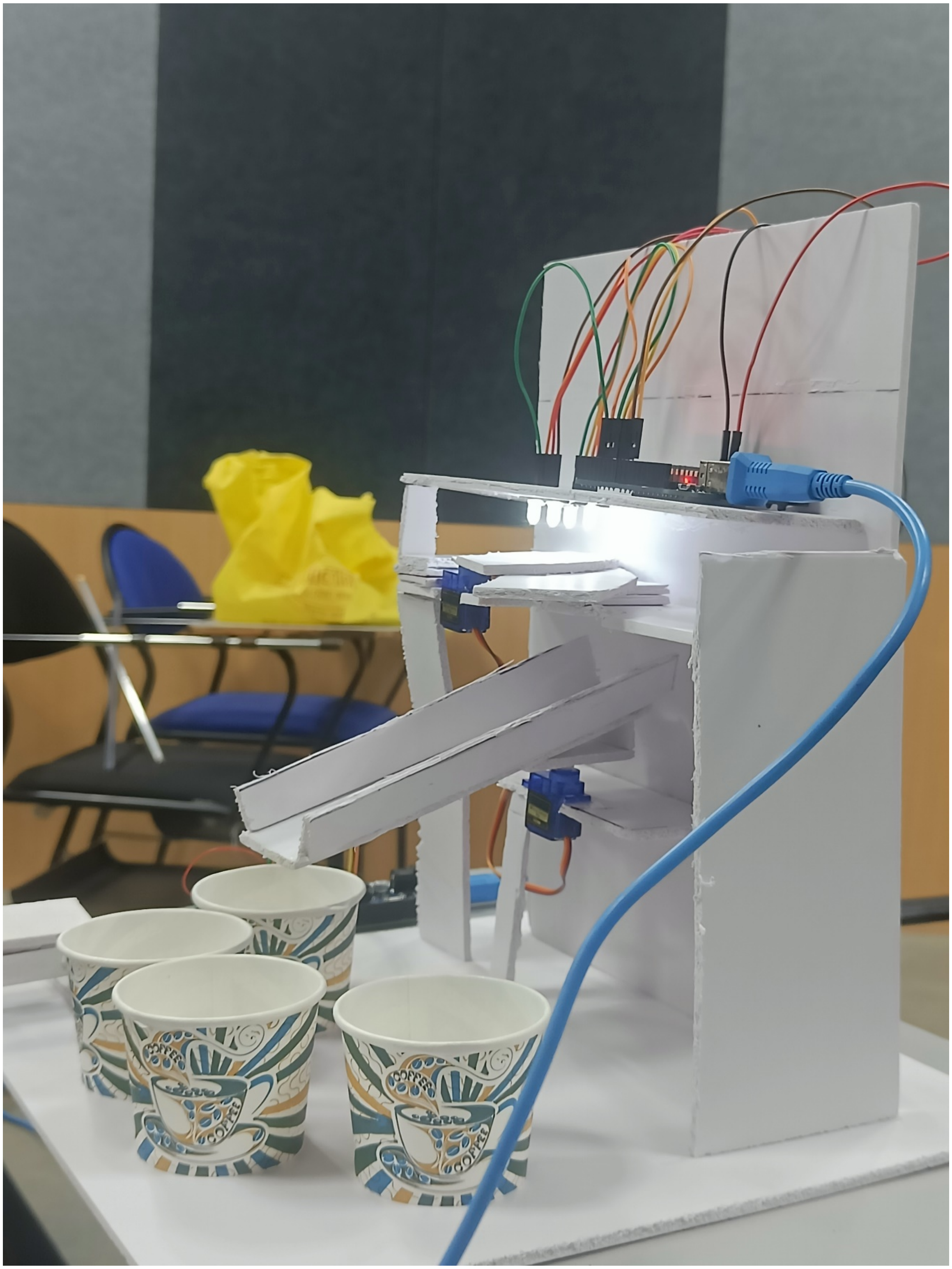
https://color-sorter-6ee36-default-rtdb.firebaseio.com

Your security rules are defined as public, so anyone can steal, modify or delete data in your database

```
https://color-sorter-6ee36-default-rtdb.firebaseio.com/
▾ user
    ▾ 025df0fbfa6
        blue: 338
        color: 66
        count: 1
        green: 285
        red: 265
```

Database location: United States (us-central1)


Weight sensor


COFFEE

Challenges Faced :

- Sensitivity of color sensor to lighting of the environment it is placed in.
- Automating weight sensing

Codes :

- Node MCU code

```
#include <FirebaseESP8266.h>
#include <ESP8266WiFi.h>
#include <Wire.h>

char cnt, clr;
int r, g, b;
#define FIREBASE_HOST "https://color-sorter-6ee36-
default-rtdb.firebaseio.com/"
#define WIFI_SSID "IIITS_Student" // Change the name
of your WIFI
#define WIFI_PASSWORD "iiit5@2k18" // Change the
password of your WIFI
#define FIREBASE_Authorization_key
"3q7MUAUQVhzLcyo1MfAXQBb7xeUbufd9OCNreVqZ"

FirebaseData firebaseData;
FirebaseJson json;

void setup() {

Firebase.begin(FIREBASE_HOST,FIREBASE_Authorization_ke
y);
  Wire.begin(D1, D2); /* join i2c bus with SDA=D1 and
SCL=D2 of NodeMCU */
  Serial.begin(9600);
  WiFi.begin (WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting...");
  while (WiFi.status() != WL_CONNECTED)
  {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("IP Address: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  }

void loop() {
  Wire.requestFrom(8, 13); /* request & read data of
size 13 from slave */
  if(Wire.available()){
    String temp;
    cnt = Wire.read();
    clr = int(Wire.read());
    int count = cnt - '0';
    //yellow = 89, violet = 86, orange = 79, blue = 66
    r = int(Wire.read());
    g = int(Wire.read());
    b = int(Wire.read());
    Serial.println(r);
    Serial.println(g);
    Serial.println(b);
    Serial.println(cnt);
    Serial.println(clr);

    Firebase.setFloat(firebaseData,
"user/025df0fbfa6/red", r);
    Firebase.setFloat(firebaseData,
"user/025df0fbfa6/green", g);
    Firebase.setFloat(firebaseData,
"user/025df0fbfa6/blue", b);
    Firebase.setFloat(firebaseData,
"user/025df0fbfa6/color", clr);
    Firebase.setFloat(firebaseData,
"user/025df0fbfa6/count", count);

    delay(3000);
  }
  Serial.println();
  delay(1000);
```

```
}
```

- Arduino code
    - Color sensor

```
#include <Servo.h>
#include <Wire.h>

Servo pickServo;
Servo dropServo;

#define S0 4
#define S1 5
#define S2 7
#define S3 6
#define sensorOut 8

int frequency = 0;
int color = 0, correct = 1;
int o_count = 0, b_count = 0, v_count = 0,
y_count = 0;
String str;

void setup() {
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);

  //frequency-scaling to 20% selected
  digitalWrite(S0, LOW);
  digitalWrite(S1, HIGH);
  pickServo.attach(10);
  dropServo.attach(9);
  Wire.begin(8);                /* join i2c bus
with address 8 */
  Wire.onRequest(requestEvent); /* register
request event */
  Serial.begin(9600);
}
```

```
void loop() {
  //initial position of servo motor
  if (correct == 1) {
    pickServo.write(135);
  }
  delay(3000);
  for(int i = 135; i > 70; i--) {
    pickServo.write(i);
    delay(5);
  }
  delay(500);

  //color sensing
  color = detectColor();
  delay(1000);

  switch (color) {
    case 1:
    dropServo.write(60);
    correct = 1;
    break;

    case 2:
    dropServo.write(80);
    correct = 1;
    break;

    case 3:
    dropServo.write(100);
    correct = 1;
    break;

    case 4:
    dropServo.write(120);
    correct = 1;
    break;

    case 0:
    break;

  }
  delay(500);

  //back to initial state
  if (color == 0) {
    correct = 0;
  } else {
    for(int i = 70; i > 0; i--) {
      pickServo.write(i);
      delay(5);
    }
    delay(300);
    for(int i = 15; i < 135; i++) {
      pickServo.write(i);
      delay(5);
    }
  }

  color=0;
}

void requestEvent() {
  switch (color) {
    case 1:
    str = String(o_count);
    Wire.write(str.c_str());
    Wire.write("O");
    break;

    case 2:
    str = String(v_count);
    Wire.write(str.c_str());
    Wire.write("V");
    break;

    case 3:
    str = String(y_count);
    Wire.write(str.c_str());
    Wire.write("Y");
    break;

    case 4:
    str = String(b_count);
    Wire.write(str.c_str());
    Wire.write("B");
    break;

    case 0:
    break;

  }
}

int detectColor() {

  // activating red photodiodes to read
  digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);
  frequency = pulseIn(sensorOut, LOW);
  int red = frequency;
  Serial.print("Red = ");
  Serial.print(frequency);//printing RED color
frequency
  Serial.print("   ");
  delay(50);

   // activating blue photodiodes to read
  digitalWrite(S2, LOW);
  digitalWrite(S3, HIGH);
  frequency = pulseIn(sensorOut, LOW);
  int blue = frequency;
  Serial.print("Blue = ");
  Serial.print(frequency);
  Serial.println("   ");
  delay(50);

  // activating green photodiodes to read
  digitalWrite(S2, HIGH);
  digitalWrite(S3, HIGH);
  frequency = pulseIn(sensorOut, LOW);
  int green = frequency;
  Serial.print("Green = ");
  Serial.print(frequency);
  Serial.println("   ");
  delay(50);
```

```
 if((223<red&&red<243) && (316<green &&
green<343) && (287<blue&&blue<312)){
   color=1;
   o_count++;
   Serial.println("Orange");
   Serial.print("Total orange objects detected
so far: ");
   Serial.println(o_count);
   delay(5000);
 } else if((242<red&&red<265) &&
(315<green&&green<330) && (277<blue&&blue<287)){
   color=2;
   v_count++;
   Serial.println("Violet");
   Serial.print("Total violet objects detected
so far: ");
   Serial.println(v_count);
   delay(5000);
 } else if((210<red && red<240) &&
(285<green&&green<305) && (286<blue &&
blue<310)){
   color=3;
   y_count++;
   Serial.println("Yellow");
   Serial.print("Total yellow objects detected
so far: ");
   Serial.println(y_count);
   delay(5000);
 } else if((265<red && red<280) &&
(305<green&&green<335) && (267<blue &&
blue<290)){
   color=4;
   b_count++;
   Serial.println("Blue");
   Serial.print("Total blue objects detected
so far: ");
   Serial.println(b_count);
   delay(5000);
 } else {
   color=0;
   Serial.println("No object placed");
```

```
  }
  return color;
}
```

Weight sensor

```
#include <Arduino.h>
#include "HX711.h"

const int LOADCELL_DOUT_PIN = 4;
const int LOADCELL_SCK_PIN = 5;
HX711 scale;

void setup() {
  Serial.begin(57600);
  scale.begin(LOADCELL_DOUT_PIN,
LOADCELL_SCK_PIN);
  scale.set_scale(300);
  scale.tare();
  Serial.println(scale.get_units(), 1);

  Serial.println("Readings:");
}

void loop() {
  if (scale.is_ready()) {
    Serial.println("Weight detected: ")
    Serial.print(scale.get_units(), 1);
    Serial.println("g");
    delay(2500);
  }
}
```