IoT Based Object Sorting System

Report submitted to the SASTRA Deemed to be University

as the requirement for the course

ECE300: MINI PROJECT

Submitted by

Vankayala Annapurna

(Reg. No.: 125160063)

Mitta Harsha Vardhan

(Reg. No.: 125160077)

Bezawada Yashwitha

(Reg. No.: 125160084)

April 2024

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR, TAMIL NADU, INDIA – 613 401

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR – 613 401

Bonafide Certificate

This is to certify that the report titled **"IoT Based Object Sorting System"** submitted as a requirement for the course, **ECE300: MINI PROJECT** for B. Tech ELECTRONICS & COMMUNICATION ENGINEERING(Cyber Physical Systems) programme, is a bonafide record of the work done by **Ms. Vankayala Annapurna (Reg. No:125160063), Mr. Mitta Harsha Vardhan (Reg. No: 125160077), Ms. Bezawada Yashwitha (Reg. No:125160084)** during the academic year 2023-2024, in the school of ELECTRICAL & ELECTRONICS ENGINEERING, under my supervision.

Signature of Project Supervisor : Sumathi R

Name with Affiliation : Dr. Sumathi R, AP-III

Date :

Project *viva-voce* held on ______________________

Examiner 1 **Examiner 2**

SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 of the UGC Act, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

SCHOOL OF ELECTRICAL & ELECTRONICS ENGINEERING

THANJAVUR – 613 401

Declaration

I/We declare that the report titled **"IoT Based Object Sorting System"** submitted by me/us is an original work done by me/us under the guidance of **Dr. Sumathi R, AP-III, School of Electrical and Electronics Engineering, SASTRA Deemed to be University** during the sixth semester of the academic year 2023-24, in the **School of Electrical and Electronics Engineering**. The work is original and wherever I/We have used materials from other sources, I/We have given due credit and cited them in the text of the report. This report has not formed the basis for the award of any degree, diploma, associate-ship, fellowship, or other similar title to any candidate of any University.

Signature of the Candidates :

Yashwitha. B

Name of the Candidates : **1.** Vankayala Annapurna

2. Mitta Harsha Vardhan

3. Bezawada Yashwitha

Date :

ACKNOWLEDGEMENTS

We express our gratitude to **Prof. Dr. S Vaidhyasubramaniam, Vice Chancellor,** SASTRA Deemed University, who provided all facilities and constant encouragement during our study and **Dr. R. Chandaramouli, The Registrar, SASTRA Deemed to be University** for providing the necessary facilities to carry out this research work in our university campus.

It is our privilege to express our sincere thanks towards project coordinator, **Dr. K Thenmozhi, Dean (SEEE),** and all **Associate Deans of SEEE** who motivated us during the project.

We owe a debt of most profound gratitude to our mentor **Dr. Sumathi R** for her valuable inputs, able guidance, encouragement, wholehearted cooperation throughout our project on the topic **"IoT Based Object Sorting System."**

We extend our gratitude to SASTRA Deemed To be University for giving us this opportunity in developing the project and our **Professors** and **Lab Technicians** who have helped us out with their abilities.

We acknowledge with a deep sense of respect, gratitude towards our **Parents** and our family members, who have always supported us morally as well as financially. Finally, we thank our friends who have directly or indirectly helped us in this project work.

ABSTRACT

A lot of medium-sized and small packing companies still favour manual sorting as a conventional method. Nevertheless, because the visual inspection must be done manually by specialists, the old method is difficult, time-consuming, uneven, and tiresome. Because of how society has changed, businesses are increasingly looking for increased automation in the manufacturing sector. To address these issues, work is proposed to design, develop, and put automated ways into practice. Thus, we present a Smart system that uses servo motors, ESP 32, Arduino IDE and a TCS3200 colour sensor to sort objects based on colour.

This technology employs a colour sensor to detect colour of the object. The sorting mechanism utilises two motors: one to move the sorting tube in the proper directions and another to allow the object to pass through the tube. The sensors detect colour when an object passes and sends a signal to the mechanism. The count of the sorted items is updated in the ThingSpeak. Thus, it is feasible to design an IoT-based detecting and sorting system that is fully automated. In a variety of industries, including Waste management, Agriculture, Mining, Textiles, Grain, Sugar, Manufacturing, and more, this technology can improve industrial efficiency and address sorting problems.

Key Words

Internet of Things(IoT), ESP32, Object Sorting, Automation, ThingSpeak.

Specific Contribution

- Develop the code for Colour detection and sorting – Vankayala Annapurna
- Literature Survey and Execution – Mitta Harsha Vardhan
- Hardware Implementation for Colour detection and sorting System – Bezawada Yashwitha

Specific Learning

- Working of TCS3200 sensor, mechanism of sorting system

Signature of the Guide: 

Name : Dr. Sumathi R

List of Figures

List of Tables

Abbreviations

ADC	Analog to Digital Convertor
API	Application Programming Interface
CAN	Controller Area Network
I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound Bus
IDE	Integrated Development Environment
IoT	Internet of Things
PWM	Pulse Width Modulation
RGB	Red Green Blue
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter

Table of Contents

CHAPTER 1

INTRODUCTION

Nowadays, advancements in technology are playing an important role in the industrial sector. Industries require efficient machinery for production and packaging. But most industries still use the traditional method of manual sorting for sorting the products. Large industries with higher production scales need fast and efficient methods to sort the products, which enables speedy packaging. Even in medium-scale and small-scale industries, manual sorting is a difficult task as it needs high attention. Repetitive work often causes mental strain and irritation, which leads to errors, inaccuracy, and poor performance. The method of manual sorting requires labour, which costs more and is contradictory to the present scenario. In the Industry 4.0 era, the Internet has attracted more popularity. With the help of smart devices like sensors and actuators, IoT is a new interface that enables the clients with automated control over a wide range of tasks, like identification, control and monitoring. Because they are more effective, factories that employ IoT technology may offer a wide range of products and services at low costs.

The difficulty of humans recognising various colour shades led to the development of automatic colour recognition and grouping devices. The proposed IoT-based colour detection and sorting machine design gives an advantage over manual sorting and fits in an industrial environment so that it can be widely used in industry, industrial testing, etc. The most popular approach for monitoring, recognising, classifying, and distinguishing between objects is through colour. The proposed operation uses the colour sensor system that has been integrated with the ESP32 to detect colour. The ESP32 transmits a signal to the servo motor arrangement, which rotates the slider to the pre-programmed place after the colour is recognised to sort the things into their corresponding baskets. After sorting, the number of an individual-coloured object is displayed in Thingspeak.

Object sorting systems are essential for increasing productivity in a variety of industries. Automation allows sorting devices to operate continuously at high speeds and with precise precision. This reduces the requirement for small- and medium-sized businesses to perform manual sorting. By using colour to differentiate undesirable and discoloured objects, it can cut down on waste. creating algorithms to categorise items according to colour, converting unprocessed colour input into useful sorting judgements. The raw colour data from objects going through the sorting system is captured using a colour sensor, and the esp32 interprets this data by using algorithms to analyse colour features and map them to predefined categories.

For the system to operate more efficiently , a precise method for sending colour data in real time from sensors to IoT devices must be implemented. The ESP32 can function as a Wi-Fi module, allowing communication between the sorting system and the sensor. It uses Wi-Fi to determine the object's colour and transmits the information to a cloud platform. This data is stored on the cloud, which may also be used for real time controlling and monitoring. Offering clients access to colour data via a user-friendly interface so they can make informed choices. The object sorting system's functionalities are made easier by this user-interactive interface, that also enables remote data access for the user. In addition to giving useful information and feedback, the involvement of user in the sorting systems can increase the efficiency, and precision of the process.

CHAPTER 2

REVIEW OF LITERATURE

In the chosen base paper[1], the suggested system sorts objects by colour using servo motors, an ESP 32, an Arduino IDE, and a TCS3200 colour sensor. This technology uses a colour sensor to determine the colour of an object. The sorting mechanism has two motors for control action. When an object passes, the sensors identify its colour and transmit a signal to the mechanism. The count of sorted items is updated in ThingSpeak. So, it may be accessed from anywhere. An IoT-based detection and sorting system can be totally automated.

In [2-3], researchers designed a sorting system based on a microcontroller. The proposed model uses a colour detector that detects the colour of the object moving across the slider. A response is sent to the microcontroller which controls the motion of the dc motors which move the slider at predefined angles to sort the objects. The disadvantage of this system is the effect of surrounding light. In [4-5], researchers use the image processing technique for colour detection in the implemented system. The object that is continuously running through the conveyor system is recognized by image acquisition devices such as pi-cameras. The objects are enumerated and sorted to their respective locations using microcontroller circuits. This system can only sort basic colours not their shades.

In [6-7], The design of a PLC-based sorting system with 2 electromagnetic actuators driving a conveyor belt for object transportation was showcased by the authors. The TCS3200 figures out the colour of the object and transmits signals to the PLC after detection. The PLC then activates the actuators to push the objects to their designated stations. However, a drawback of this system is the lack of object tracking and the low efficiency of the sorting mechanism.

In references [8-9], the authors used a selecting and placement mechanism with a robotic arm for object sorting. A colour sensor detects the colour of objects passing along the conveyor belt and sends a signal to the control unit, which activates the servo motors that drive the robotic arm. The robotic arm takes up the product from the conveyor system and sets it in the predetermined location. This system does not have modern controllers or sensors.

CHAPTER 3

COMPONENTS AND TERMINOLOGIES

3.1. Colour Sensor

The Colour Sensor used for the prototype of Object Sorting System is TCS3200 Sensor.

3.1.1. Introduction

The TCS3200 is a colour sensor chip with a broad spectrum of visible colour detection and measurement capabilities. It contains a variety of photodiodes filtered with clear, blue, green, and red light. A square pulse with a frequency corresponding to the selected colour's intensity is produced by the inbuilt colour sensing chip and is displayed in the output.
Characteristics:

- Colour selection that can be programmed to concentrate on red, green, blue, or all colours
- Variable frequency of output
- Connects to microcontrollers such as Arduino
- Frequently comes on a module that has white LEDs integrated within it for illumination.

Fig 3.1 TCS3200 Colour Sensor

3.1.2 Working

- **Light Source Activation**: An integrated white LED source light is part of the TCS3200 sensor. This LED illuminates the object for identification of colour.
- **Colour detection**: Light rays are reflected to the sensor by the object. The filters that are placed over the photodiodes divide the light into RGB components.
- **Photodiode Activation**: Every component of filtered light hits a matching array of colour-sensitive photodiodes of the sensor.

- **Electrical Signal Generation**: Photodiodes produce electrical signals that are directly proportional to the intensity of the light they receive. More intense light produces strong electrical signal.
- **Analog to Digital Conversion**: An ADC is built in the sensor. It converts these electrical signals generated into digital signals. In this conversion process, the intensity of every colour component (R,G,B) is given one numerical value.
- **Output Data**: The microcontroller or an external device connected to the sensor receives digital signals that indicate the intensity of the RGB light components.
- **Data processing**: To determine the object's colour, the microcontroller processes these received digital signals . To determine the exact colour, it might compare the relative intensities of the RGB components using algorithms.

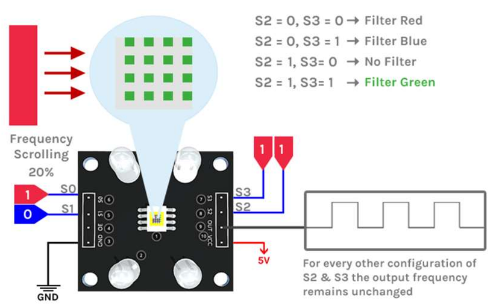


Fig 3.2 Working of TCS3200 Sensor

3.2 Servo Motor

3.2.1 Introduction

Servo motors, which offer exact control over angular position, velocity, and acceleration, are crucial parts of many electromechanical systems. They find extensive application in robotics, automation, aircraft, and many other fields requiring precise and regulated motion. Servo motors operate according to the closed-loop control concept, which modifies the motor's movement based on data from a position sensor.

Fig 3.3 Servo Motor

3.2.2 Working

- **Power Connection**: Three wires are provided in a servo motor: control, ground, and power. Connect the control wire to a PWM-enabled pin on the microcontroller (e.g., Arduino), the ground wire to the ground, and the power wire to a 5V power supply.
- **Initialization**: Construct a servo object in the code and connect it to the PWM pin in the microcontroller's setup function to initialize the servo motor.
- **Position Control**: To set the desired position of servo motor, use the write() function from the servo library to give desired angles like 90 or 0 degrees.
- **Feedback Control**: The internal feedback circuit of a servo motor modifies its current position or angle to match it with the desired position. The feedback circuit modifies velocity and direction of the motor based on a comparison between the desired and current position.

3.3 ESP32

ESP32 used for the sorting system prototype is ESP Wroom-32 Wi-Fi Module.

The ESP32 is a low-cost , low-power microprocessor board with built-in Wi-Fi and dual-mode Bluetooth. With a wide range of peripherals including SPI, I2C, I2S, UART, and CAN interfaces, it is built on a 32-bit Xtensa LX6 CPU. It consists of a 12-bit ADC, a temperature sensor, and a hall sensor. It is widely used for IoT applications that utilizes wireless connectivity because it has two high-speed Wi-Fi antennae and supports 802.11 protocols. In addition, it is also compatible with Bluetooth Low Energy protocol and Bluetooth 4.2, which allows it to connect to a variety of gadgets, such as tablets, smartphones, sensors, and other IoT devices.ESP32 is frequently utilised in IoT applications such as industrial automation, wearables, and smart home devices.

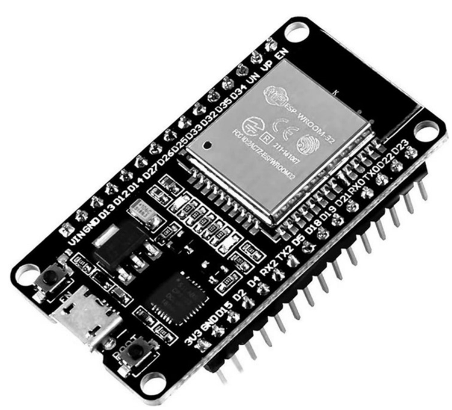


Fig 3.4 ESP Wroom-32 Wi-Fi Module

3.4 Cloud Platform

The term "Cloud" refers to Internet-accessible servers, as well as the software and databases that run on them. Cloud servers are housed in data centres throughout the world. Cloud computing eliminates the need for users and businesses to operate physical servers or execute software programs on their own workstations. The cloud allows users to access the same files and programs from nearly any device because computing and storage takes place on servers in a data centre, rather than locally on the user device.

3.4.1 ThingSpeak

With the help of the ThingSpeak, an advanced IoT platform, users can gather, analyse, and display sensor data instantly. ThingSpeak's wide range of features and user-friendly interfaces make it easier to develop IoT applications and remotely monitor devices. Using a wide range of communication protocols, users may quickly and easily connect their sensors, actuators, and other IoT devices to it. ThingSpeak empowers clients to deal with, store, and inspect the information created by these gadgets after they are connected. Its ability to deliver individualized representations, such charts, markers, and guides, is one of its essential characteristics. These perceptions offer keen data about the information assembled and allows clients to make additional IoT frameworks and robotize processes by incorporating with extra IoT stages and administrations.

Fig 3.5 ThingSpeak Cloud

3.5 Arduino IDE

Arduino IDE is a software tool that enables us for Writing, compiling, and uploading code to Arduino microcontroller boards, such as the Uno. It gives the comfort to the beginners which makes programming easy with a streamlined C/C++ environment . The IDE starts with a text editor for coding, various options for gathering and uploading designs to the board, a debugging Serial Monitor and a Serial Plotter. The Arduino IDE's intuitive interface and huge library resources are used to build a variety of projects, from straightforward LED lights that blink to intricate robotics and Internet of Things applications which supports users.

CHAPTER 4

EXPERIMENTAL WORK AND METHODOLOGY

4.1 Methodology

The Sorting system consists of a colour sensor TCS3200, ESP32, two servo motors. The process starts by the control action of first servo motor which acts as a gate and picks the objects. Later continues with identifying the colour of object that is placed in the system near colour sensor. A sorting algorithm is developed to classify the objects based on their colours using the intensities. Once the colour is detected, the intensities of RGB colours are displayed in the serial monitor of Arduino IDE. Based on the RGB intensities, a decision is made to decide the colour of Object according to the logic provided in the algorithm. After detecting colour, the object is placed near the slider where another servo motor drops the object into its respective destination which is pre-programmed. If there is no object placed, the system remains idle for the defined time interval before picking another object. Finally, the count of each individual-coloured object is updated in the respective fields of ThingSpeak channel and the count is displayed in the channel.

Fig 4.1 Methodology

4.2 Flow Chart

The steps that are followed to develop the algorithm for object sorting can be represented in a flow chart. The colour detection is done based on the logics provided in the algorithm with the use of the primary colour intensities. The sensor can detect RGB colours and some other colours such as Yellow, Purple, Orange additional to white colour. If no object is placed, based on the logic , output is shown as *No Object placed*. The current flow chart Fig 4.2.1 and Fig 4.2.2 is taken from our base paper as reference. The algorithm in this project is further developed to detect secondary colours which is not implemented in the reference paper.

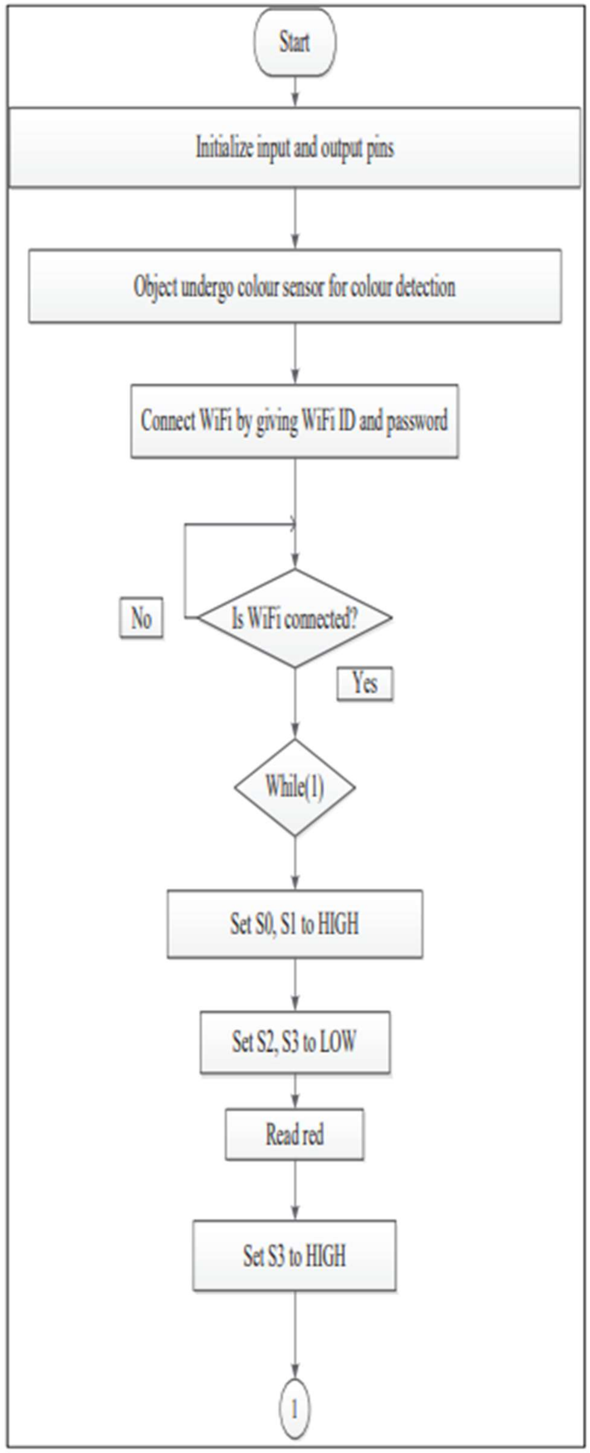


Fig 4.2(a) Flowchart for Algorithm(a)

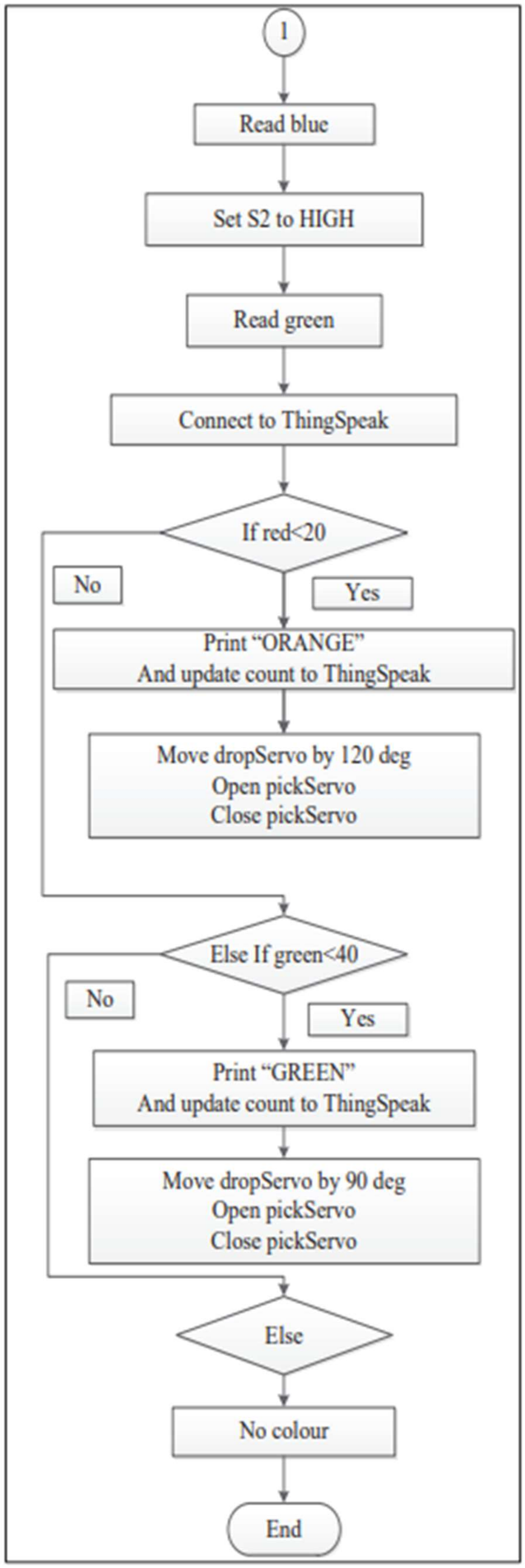


Fig 4.2(b) Flowchart for Algorithm (b)

4.3 Code

```
#include <ESP32Servo.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_TCS34725.h>
#include <WiFi.h>;
#include <ThingSpeak.h>;

unsigned long myChannelNumber = 2438062;
const char * myWriteAPIKey = "7UX7NDWFZLCNIN3L ";
const char *ssid =  "Annu";
const char *pass =  "apkv_919";
WiFiClient client;

Servo pickServo;
Servo dropServo;
const int s0 = 19;
const int s1 = 18;
const int s2 = 2;
const int s3 = 4;
const int out = 15;
int red,blue,green,white;
int r=0;
int w=0;
int b=0;
int g=0;
int p=0;
int y=0;
int bl=0;
int CLOSE_ANGLE = 90;
int OPEN_ANGLE = 0;

void setup()
```

Fig 4.3(a) Code (a)

```
{
  Serial.begin(115200);
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(out, INPUT);
  digitalWrite(s0, HIGH);
  digitalWrite(s1, HIGH);
  pickServo.attach(13);
  dropServo.attach(21);
  pickServo.write(90);
  dropServo.write(180);

  ThingSpeak.begin(client);
  Serial.println("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED)
    {
         delay(550);
         Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
}

void loop()
{
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
```

Fig 4.3(b) Code(b)

```
digitalWrite(s3, HIGH);
blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
digitalWrite(s2, HIGH);
green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
digitalWrite(s3, LOW);
white = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
Serial.print("R Intensity:");
Serial.print(red, DEC);
Serial.print(" G Intensity: ");
Serial.print(green, DEC);
Serial.print(" B Intensity : ");
Serial.println(blue, DEC);

if(red>25 & red>blue & red<green & green>45)
{
 dropServo.write(120);
 delay(1000);
 r++;
 Serial.print("Red:");
 Serial.println(r);
 open1();
 delay(2000);
 close1();
 ThingSpeak.writeField(myChannelNumber, 1,r, myWriteAPIKey);
}
else if(green>red & green>blue & green>30 & red>20 & red<26){
 dropServo.write(140);
 delay(1000);
 g++;
 Serial.print("Green:");
 Serial.println(g);
 open1();
```

Fig 4.3(c) Code (c)

```
  delay(2000);
  close1();
  ThingSpeak.writeField(myChannelNumber, 2,g, myWriteAPIKey);
 }
else if(blue<green & blue<red & blue>11 & red<25){
  dropServo.write(160);
  delay(1000);
  b++;
  Serial.print("Blue:");
  Serial.println(b);
  open1();
  delay(2000);
  close1();
  ThingSpeak.writeField(myChannelNumber, 3,b, myWriteAPIKey);
 }
 else if(red>15  & green>=blue & blue<12 & green>25){
  dropServo.write(200);
  delay(1000);
  y++;
  Serial.print("Yellow:");
  Serial.println(y);
  open1();
  delay(2000);
  close1();
  ThingSpeak.writeField(myChannelNumber, 5,y, myWriteAPIKey);
 }
 else if(blue<=red & green<35 & red>20)
 {
  dropServo.write(220);
  delay(1000);
  p++;
  Serial.print("purple:");
```

Fig 4.3(d) Code(d)

```
  Serial.println(p);
  open1();
  delay(2000);
  close1();
  ThingSpeak.writeField(myChannelNumber, 6,p, myWriteAPIKey);
 }
 else if(white<10){
  dropServo.write(240);
  delay(1000);
  w++;
  Serial.print("White:");
  Serial.println(w);
  open1();
  delay(2000);
  close1();
  ThingSpeak.writeField(myChannelNumber, 4,w, myWriteAPIKey);
 }
 else{
  dropServo.write(180);
  delay(1000);
 // bl++;
  Serial.print("No object placed");
  //Serial.println(bl);
  //open1();
  delay(1000);
  //close1();
  //ThingSpeak.writeField(myChannelNumber, 7,bl, myWriteAPIKey);
 }

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

Fig 4.3(e) Code(e)

4.4 Prototype/ Hardware

The objects to be sorted are inputted into the system according to the design. First servo motor which picks the objects is attached to the colour sensor, allowing it to detect the object's colour. The TCS3200 colour sensor uses 4 LEDs to illuminate light on the object and photodiodes to detect colour. The intensity and amount of reflected light affect the output current. A square pulse is created from the current variation and sent to the ESP32. The ESP32 will receive the input frequencies from the sensor and compute the colour based on frequency combinations of filtered photodiodes. The ESP32 transmits the data to the Thing's Speak through the inbuilt ESP32 Wi-Fi module and displays the count of the article in issue speak. At the same another servo motor changes the output path that leads the object to the pre-defined destinations.

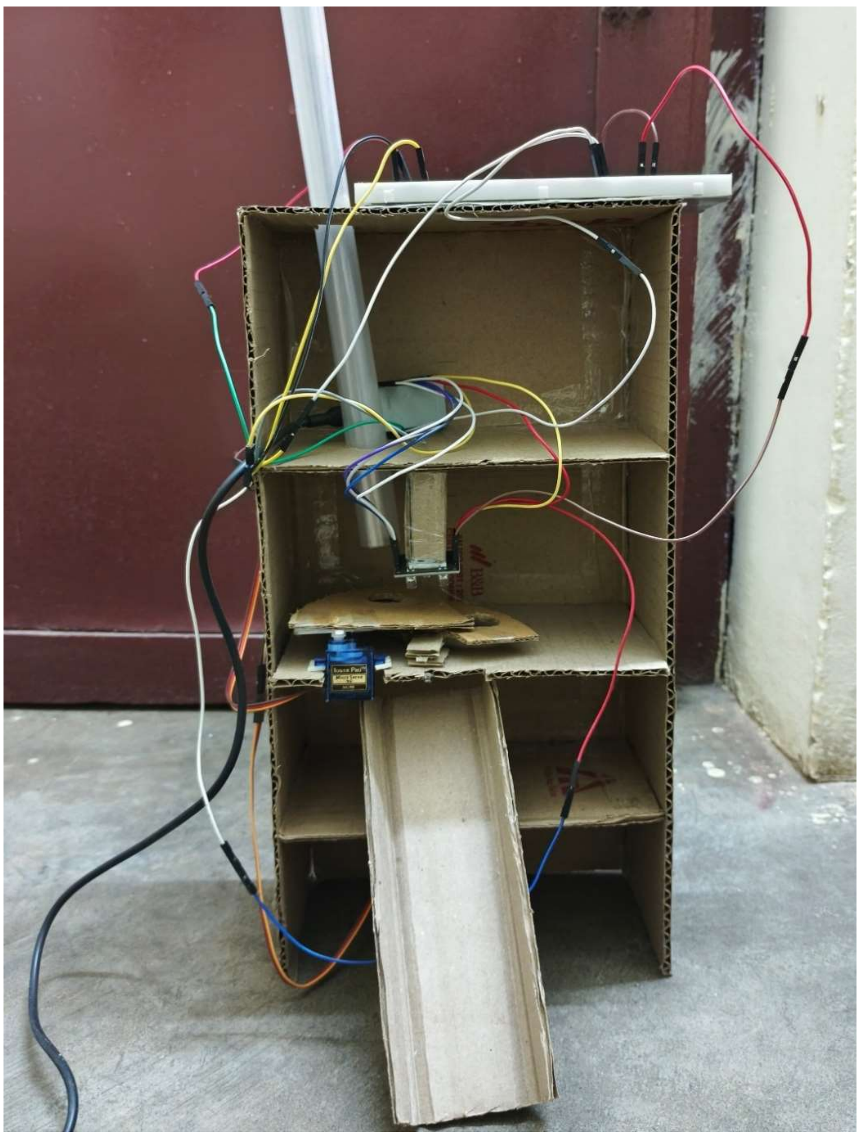


Fig 4.4 Prototype

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1. Results

Setup is built and Code is developed for the experimental work. The obtained results for the experiment conducted:

5.1.1. Serial Monitor

The output in the serial monitor consists of two lines:

- **Line 1**:
  RGB colour intensities of an object detected by the sensor.
- **Line 2**:
  Colour of the object with its count.

a. Serial Monitor output when *Red* object is placed.

```
R Intensity:31 G Intensity: 40 B Intensity : 37
Red:39
```

Fig 5.1(a) Serial Monitor output (a)

b. Serial Monitor output when *Blue* object is placed

```
R Intensity:9 G Intensity: 15 B Intensity : 6
blue :2
```

Fig 5.1(b) Serial Monitor output (b)

c. Serial Monitor output when *Green* object is placed

```
R Intensity:10 G Intensity: 17 B Intensity : 7
green :3
```

Fig 5.1(c) Serial Monitor output (c)

d. When *No object* is placed

```
R Intensity:0 G Intensity: 0 B Intensity : 0
NO OBJECT PLACED
```

Fig 5.1(d) Serial Monitor output (d)

e. When *White* object is placed

```
R Intensity:13 G Intensity: 13 B Intensity : 22
White:4

R Intensity:6 G Intensity: 10 B Intensity : 9
White:5
```

Fig 5.1(e) Serial Monitor output (e)

f. When *Yellow* object is placed

```
R Intensity:16 G Intensity: 26 B Intensity : 10
Yellow:2
```

Fig 5.1(f) Serial Monitor output (f)

g. When *Purple* object is placed

```
R Intensity:18 G Intensity: 31 B Intensity : 18
purple:1
```

Fig 5.1(g) Serial Monitor output (g)

5.1.2. ThingSpeak

The sensor detects the colour, the servo motor drops the object in the respective location. After this, the count of each individual-coloured object is updated in the respective fields of ThingSpeak channel and the count is displayed in the channel.

The output shown in the ThingSpeak channel in the visualization form of *Numeric Display.* Each colour has its individual field where the object count of respective colours is updated. Anyone person with the Channel ID and ReadAPIKey can access the Channel and view the data from anywhere.

a. ThingSpeak – Red object field

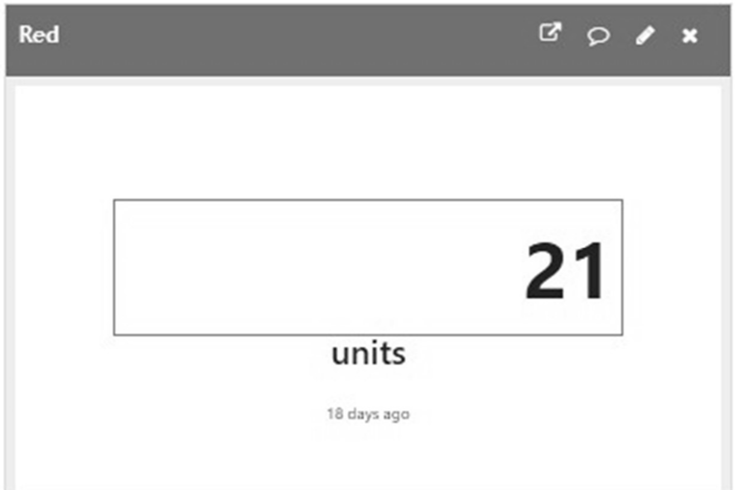


Fig 5.2(a) Red object field

b. ThingSpeak – Blue object field

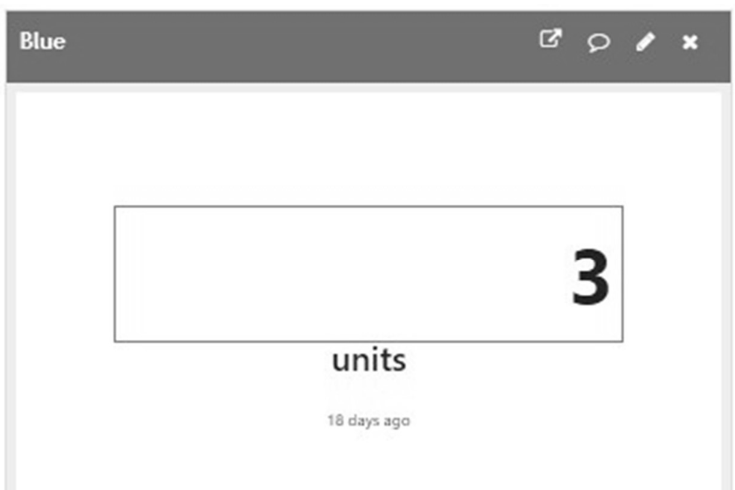


Fig 5.2(b) Blue object field

c. ThingSpeak – Secondary Colour object fields

Fig 5.2(c) Secondary colour object fields

5.2 Observations

When the experiment is conducted, it is observed that the RGB intensities detected by the colour sensor of the same object varies from place to place and time to time. After conducting the experiment for various times, it is found that the intensities detected by the colour sensor varies due to the intensity of Sunlight. In a dark room, the RGB Intensities comes up to *Thousands* , and in extreme sunlight conditions, the values of RGB intensities comes closer to *Zero*. The changes observed in the intensities of colours in various places are tabulated.

- **Extreme** Sunlight Condition – No object placed

Colour	Intensity
Red	0
Green	0
Blue	0

Table 5.1 Intensities in Extreme Sunlight

- **Heavy** Sunlight Condition

Object Colour	Colour Intensity	Intensity
Red	Red	9
Red	Green	16
Red	Blue	7
Green	Red	10
Green	Green	17
Green	Blue	7

Table 5.2 Intensities in Heavy Sunlight

- **Medium** Sunlight Condition

Object Colour	Colour Intensity	Intensity
Red	Red	24
Red	Green	43
Red	Blue	14
Green	Red	25
Green	Green	38
Green	Blue	14

Table 5.3 Intensities in Medium Sunlight

- **Low** Sunlight Condition

Object Colour	Colour Intensity	Intensity
Red	Red	47
Red	Green	66
Red	Blue	63
Blue	Red	88
Blue	Green	72
Blue	Blue	62

Table 5.4 Intensities in Low Sunlight

- **Dark** Condition – Red object placed

Colour	Intensity
Red	1425
Green	733
Blue	220

Table 5.5 Intensities in Dark room

CHAPTER 6

CONCLUSION AND FURTHER WORK

6.1 Conclusion

The project on an IoT-based item sorting system, followed by color identification of different objects, has proven to be a viable and practical solution for sorting objects in the small and medium packaging sectors. The system can detect distinct colors based on RGB intensities reflected from the item and sort the objects using servo-motor control actions deployed. The findings of this study demonstrate the usage of an item sorting system to reduce human work and time-consuming processes in a variety of sectors.

A fully automated Object sorting system where colour detection of the objects is done using the TCS3200 colour sensor and the result is published in the ThingSpeak cloud. Based on the observations, The colour sensor gives best output when the source light has steady intensity environment. This System when implemented in a large scale is suitable for Industrial Applications as they are closed environment and the requirements matches. This initiative on color detection and sorting has the potential to transform the packaging industry, particularly for medium and small-scale operations.

6.2 Future Work

The system can be further enhanced by adding additional features as per the user requirements .One such property is weight of the object, which can act as an additional parameter to improve system efficiency and production. This can be done by including load cell and applying appropriate decision-making algorithms. The current system uses existing components that are feasible and available easily, but in the case of large-scale industries that handle large quantities for massive production, sorting can be done efficiently by placing multiple sensors in the right places. Design changes are possibly made depending on the size and shape of the products used in various industries . IoT can be used to connect the system with other industrial systems, such as conveyor belts and storage systems, to improve production line. Overall, object -sorting system has promising future , with possible applications in various industries .The System can also be integrated with emerging technologies such as robotics, automation, computer vision and IoT.

CHAPTER 7

REFERENCES

- Rajitha Rayala, Vimala Kumari Jonnalagadda , Venkata Sai Sriya Vempati , "Colour Detection and Sorting of Objects using IoT ", International Conference on Next Generation Intelligent Systems , July 2022
- Lim JieShen and Irda Hassan, "Design And Development of Colour Sorting Robot", Jan 2015.Google Scholar
- Amir Deshmukh, Mahesh Nagane and Vaibhav Avtade, "Design and Development of Automatic Conveyor Controlling System for sorting of component on colour basic", International Journal of Science and Research (IJSR), 2016.Google Scholar
- Avadhoot R. Telepatil and Prashant M. Jadhav, "Colour Object Counting and Sorting Mechanism Using Image Processing Approach", International Journal of Modern Trends in Engineering and Research (IJMTER), vol. 02, no. 03, March 2015.Google Scholar
- S.V. Rautu , "sorting of objects based on colour weight and type on a conveyor line using plc", IOSR Journal of mechanical and civil engineering (IOSR-JMCE), pp. 2278-1684, 2017.CrossRef Google Scholar
- Y V Aruna, "Automatic convey or System with InProcess Sorting Mechanism", International Journal of Engineering Research and Applications, vol. 5, no. 11, pp. 2248-9622, 2015.Google Scholar.
- Yeow Khang Yung, "Colour Sorting System with Robot Arm' Faculty of Electronic and Computer Engineering University Technical Malaysia Melaka, 2011.
- M. C. Mulder, S. R. Malladi, "A minimum effort control application for Cooperating sensor driven intelligent multi-jointed robotic arm," Proceedings of the 30th IEEE Conference on Decision and Control, vol. 2,pp. 1573–1578, 1991.
- Pereira, V., Fernandes, V. A., & Sequeira, J, " Low-cost object sorting robotic arm using Raspberry Pi", IEEE Global Humanitarian Technology Conference-South Asia Satellite (GHTC-SAS) (pp. 1- 6), September 2014.

CHAPTER 8

APPENDIX

8.1 Similarity Check Report

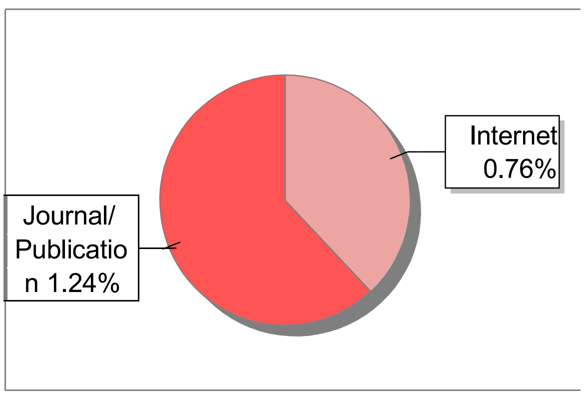
The Report is Generated by DrillBit Plagiarism Detection Software

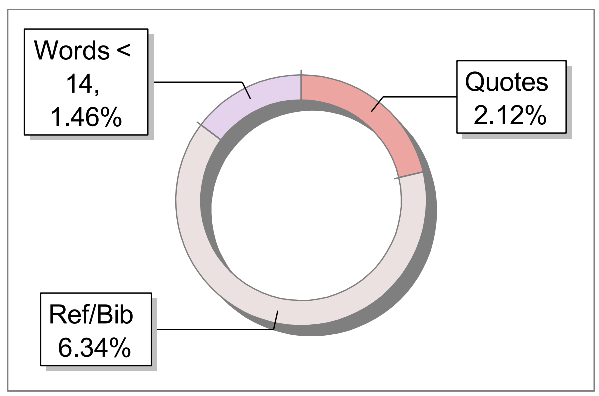
Submission Information

Author Name	Annapoorna and Batch
Title	IoT Based Object Sorting System
Paper/Submission ID	1702886
Submitted by	sumathi@ece.sastra.edu
Submission Date	2024-04-26 22:16:15
Total Pages	32
Document type	Project Work

Result Information

Similarity **2 %**





Exclude Information

Quotes	Not Excluded
References/Bibliography	Excluded
Sources: Less than 14 Words %	Not Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	No

A Unique QR Code use to View/Download/Share Pdf File

DrillBit Similarity Report

2	6	A	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)
SIMILARITY %	MATCHED SOURCES	GRADE	

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	Thesis Submitted to Shodhganga Repository	1	Publication
3	IEEE 2019 Seventh International Conference on Digital Information Pr	<1	Publication
4	Inherent and growth stage-related differences in growth and lipid and sterol com by Donn-2002	<1	Publication
5	bettershark.com	<1	Internet Data
6	moam.info	<1	Internet Data
7	docplayer.net	<1	Internet Data