Table of Contents

[Appendix 1 1](#_Toc136509328)

[Appendix 2 5](#_Toc136509329)

[Appendix 3 9](#_Toc136509330)

[Appendix 4 15](#_Toc136509331)

# Appendix 1

#include <Servo.h>

#define S0 4

#define S1 5

#define S2 7

#define S3 6

#define sensorOut 8

Servo servo;

Servo servo1;

int color = 0; //setting color function return value to 0

int frequency = 0; //setting frequency to 0

void setup() {

Serial.begin(9600);

pinMode(sensorOut, INPUT);

pinMode(S0, OUTPUT);

pinMode(S1, OUTPUT);

pinMode(S2, OUTPUT);

pinMode(S3, OUTPUT);

digitalWrite(S0, HIGH);

digitalWrite(S1, LOW);

servo.attach(9);

servo1.attach(10);

}

void loop() {

servo.write(150); // set the first servo to the starting position where it catches the candy

delay(500);// wait half of second

for (int i = 150; i >= 65; i--) { // Move the first servo to the position of color sensor

}

delay(1000); // wait 1 second

color = readColor(); // Get the return value from the color sensor

delay(500);//wait half of second

switch (color) {

case 1:

servo1.write(45); // move second servo to an angle of 45 degrees

break;

case 2:

servo1.write(90); // move second servo to an angle of 90 degrees

break;

case 3:

servo1.write(135);// move second servo to an angle of 135 degrees

break

case 0:

break; // situation where the measurements are not matching any of the color's frequency values

}

delay(500);//wait half of a second

for (int i = 0; i >= 0; i--) { // Move the first servo to the hole to drop th candy

servo.write(i);

delay(50);

}

delay(200);

for (int i = 0; i <= 150; i++) { // Return first servo to the initial position

servo.write(i);

delay(2);

}

color = 0; // set color value back to 0

}

}

int readColor() { // Color sensor function to read the color

digitalWrite(S2, LOW);

digitalWrite(S3, LOW);

frequency = pulseIn(sensorOut, LOW);

int R = frequency;

Serial.print("R= ");

Serial.print(frequency);

Serial.print(" ");

delay(50);

digitalWrite(S2, HIGH);

digitalWrite(S3, HIGH);

frequency = pulseIn(sensorOut, LOW);

int G = frequency;

Serial.print("G= ");

Serial.print(frequency);

Serial.print(" ");

delay(50);

digitalWrite(S2, LOW);

digitalWrite(S3, HIGH);

frequency = pulseIn(sensorOut, LOW);

int B = frequency;

Serial.print("B= ");

Serial.print(frequency);

Serial.println(" ");

delay(50);

if (R < 100 && R > 80 && G > 160 && G < 220 && B > 120 && B < 175) {

color = 1; // Red - case 1

}

if (G < 115 && G > 90 && R < 97 && R > 82 && B < 125 && B > 90) {

color = 2; // Green - case 2

}

if (B < 80 && B > 40 && G > 75 && G < 120 && R > 55 && R < 95) {

color = 3; // Blue - case 3

}

return color; // return the value 1,2 or 3

}

# Appendix 2

#include <Servo.h>

#include <Wire.h>

#include <VL53L1X.h>

#define S0 4

#define S1 5

#define S2 7

#define S3 6

#define sensorOut 8

Servo servo;

Servo servo1;

int color = 0; //setting color function return value to 0

int frequency = 0; //setting frequency to 0

VL53L1X sensor;

void setup() {

Serial.begin(9600);

pinMode(sensorOut, INPUT);

pinMode(S0, OUTPUT);

pinMode(S1, OUTPUT);

pinMode(S2, OUTPUT);

pinMode(S3, OUTPUT);

digitalWrite(S0, HIGH);

digitalWrite(S1, LOW);

servo.attach(9);

servo1.attach(10);

Wire.begin(); // allow communication of Arduino board with VL53L1X sensor

Wire.setClock(400000); // use 400 kHz I2C for fast mode of communication

sensor.setTimeout(500);

if (!sensor.init()) {

Serial.println("Failed to detect and initialize sensor!"); // print a message if sensor is not detected

while (1);

}

sensor.setDistanceMode(VL53L1X::Short);// chose short mode for most precise values

sensor.setMeasurementTimingBudget(20000); //set maximum time allowed for every measurement to 20 miliseconds

sensor.startContinuous(50);// set time betweeen the measurements to 50 miliseconds

}

}

void loop() {

if (sensor.read() <= 100) { // Start the sorting process if an object is detected within 10 cm

servo.write(150); // set the first servo to the starting position where it catches the candy

delay(500);// wait half of second

for (int i = 150; i >= 65; i--) { // Move the first servo to the position of color sensor

}

delay(1000); // wait 1 second

color = readColor(); // Get the return value from the color sensor

delay(500);//wait half of second

switch (color) {

case 1:

servo1.write(45); // move second servo to an angle of 45 degrees

break;

case 2:

servo1.write(90); // move second servo to an angle of 90 degrees

break;

case 3:

servo1.write(135);// move second servo to an angle of 135 degrees

break

case 0:

break; // situation where the measurements are not matching any of the color's frequency values

}

delay(500);//wait half of a second

for (int i = 0; i >= 0; i--) { // Move the first servo to the hole to drop th candy

servo.write(i);

delay(50);

}

delay(200);

for (int i = 0; i <= 150; i++) { // Return first servo to the initial position

servo.write(i);

delay(2);

}

color = 0; // set color value back to 0

}

}

int readColor() { // Color sensor function to read the color

digitalWrite(S2, LOW);

digitalWrite(S3, LOW);

frequency = pulseIn(sensorOut, LOW);

int R = frequency;

Serial.print("R= ");

Serial.print(frequency);

Serial.print(" ");

delay(50);

digitalWrite(S2, HIGH);

digitalWrite(S3, HIGH);

frequency = pulseIn(sensorOut, LOW);

int G = frequency;

Serial.print("G= ");

Serial.print(frequency);

Serial.print(" ");

delay(50);

digitalWrite(S2, LOW);

digitalWrite(S3, HIGH);

frequency = pulseIn(sensorOut, LOW);

int B = frequency;

Serial.print("B= ");

Serial.print(frequency);

Serial.println(" ");

delay(50);

if (R < 100 && R > 80 && G > 160 && G < 220 && B > 120 && B < 175) {

color = 1; // Red - case 1

}

if (G < 115 && G > 90 && R < 97 && R > 82 && B < 125 && B > 90) {

color = 2; // Green - case 2

}

if (B < 80 && B > 40 && G > 75 && G < 120 && R > 55 && R < 95) {

color = 3; // Blue - case 3

}

return color; // return the value 1,2 or 3

}

# Appendix 3

#include <Servo.h>

#include <Wire.h>

#include <VL53L1X.h>

#include <FastLED.h>

#define S0 4

#define S1 5

#define S2 7

#define S3 6

#define sensorOut 8

#define LED\_PIN 2

#define NUM\_LEDS 70

#define LED\_TYPE WS2811

#define COLOR\_ORDER GRB

Servo servo;

Servo servo1;

int color = 0; //setting color function return value to 0

int frequency = 0; //setting frequency to 0

VL53L1X sensor;

CRGB leds[NUM\_LEDS];

void setup() {

Serial.begin(9600);

pinMode(sensorOut, INPUT);

pinMode(S0, OUTPUT);

pinMode(S1, OUTPUT);

pinMode(S2, OUTPUT);

pinMode(S3, OUTPUT);

digitalWrite(S0, HIGH);

digitalWrite(S1, LOW);

servo.attach(9);

servo1.attach(10);

Wire.begin(); // allow communication of Arduino board with VL53L1X sensor

Wire.setClock(400000); // use 400 kHz I2C for fast mode of communication

sensor.setTimeout(500);

if (!sensor.init()) {

Serial.println("Failed to detect and initialize sensor!"); // print a message if sensor is not detected

while (1);

}

sensor.setDistanceMode(VL53L1X::Short);// chose short mode for most precise values

sensor.setMeasurementTimingBudget(20000); //set maximum time allowed for every measurement to 20 miliseconds

sensor.startContinuous(50);// set time betweeen the measurements to 50 miliseconds

FastLED.addLeds<LED\_TYPE, LED\_PIN, COLOR\_ORDER>(leds, NUM\_LEDS); // Initialize the LED strip

FastLED.setBrightness(50);

}

}

void loop() {

if (sensor.read() <= 100) { // Start the sorting process if an object is detected within 10 cm

servo.write(150); // set the first servo to the starting position where it catches the candy

delay(500);// wait half of second

for (int i = 150; i >= 65; i--) { // Move the first servo to the position of color sensor

}

delay(1000); // wait 1 second

color = readColor(); // Get the return value from the color sensor

delay(500);//wait half of second

switch (color) {

case 1:

servo1.write(45); // move second servo to an angle of 45 degrees

for (int i = 0; i < NUM\_LEDS; i++) {

leds[i] = CRGB::Red; // Set the LED color to red

}

FastLED.show(); // Update the LED strip with the new color

delay(500);

break;

case 2:

servo1.write(90); // move second servo to an angle of 90 degrees

for (int i = 0; i < NUM\_LEDS; i++) {

leds[i] = CRGB::Green; // Set the LED color to green

}

FastLED.show();

delay(500);

break;

case 3:

servo1.write(135);// move second servo to an angle of 135 degrees

for (int i = 0; i < NUM\_LEDS; i++) {

leds[i] = CRGB::Blue; // Set the LED color to blue

}

FastLED.show();

delay(500);

break;

case 0:

break; // situation where the measurements are not matching any of the color's frequency values

}

delay(500);//wait half of a second

for (int i = 0; i >= 0; i--) { // Move the first servo to the hole to drop th candy

servo.write(i);

delay(50);

}

delay(200);

for (int i = 0; i <= 150; i++) { // Return first servo to the initial position

servo.write(i);

delay(2);

}

color = 0; // set color value back to 0

}

}

int readColor() { // Color sensor function to read the color

digitalWrite(S2, LOW);

digitalWrite(S3, LOW);

frequency = pulseIn(sensorOut, LOW);

int R = frequency;

Serial.print("R= ");

Serial.print(frequency);

Serial.print(" ");

delay(50);

digitalWrite(S2, HIGH);

digitalWrite(S3, HIGH);

frequency = pulseIn(sensorOut, LOW);

int G = frequency;

Serial.print("G= ");

Serial.print(frequency);

Serial.print(" ");

delay(50);

digitalWrite(S2, LOW);

digitalWrite(S3, HIGH);

frequency = pulseIn(sensorOut, LOW);

int B = frequency;

Serial.print("B= ");

Serial.print(frequency);

Serial.println(" ");

delay(50);

if (R < 100 && R > 80 && G > 160 && G < 220 && B > 120 && B < 175) {

color = 1; // Red - case 1

}

if (G < 115 && G > 90 && R < 97 && R > 82 && B < 125 && B > 90) {

color = 2; // Green - case 2

}

if (B < 80 && B > 40 && G > 75 && G < 120 && R > 55 && R < 95) {

color = 3; // Blue - case 3

}

return color; // return the value 1,2 or 3

}

# Appendix 4

A picture containing text, diagram, plan, map

Description automatically generated