CIS 4930 011

Internet-of-Things (IoT) System Design

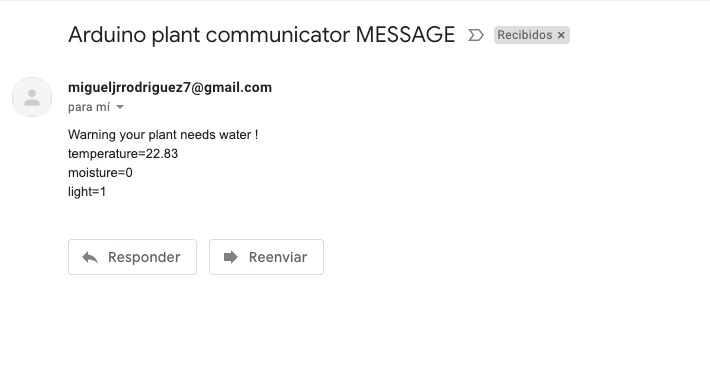
Final Project Report

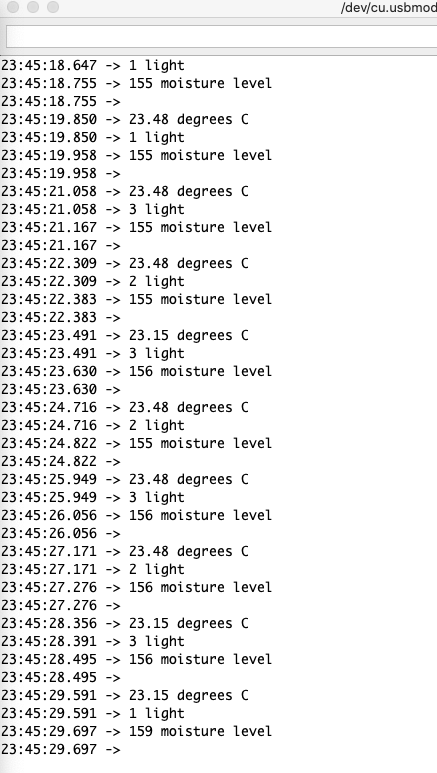
|  |  |
| --- | --- |
| Today’s Date: | 12/07/2020 |
| Your Name: | MiguelAngel Rodríguez, Maria Martinez, Dariia Bulhakova |
| Team: | 1. Name (U#) MiguelAngel Rodriguez   Project Contribution: Plant Communicator and Puzzle Box   1. Name (U#) Maria Martinez, (50286790)   Project Contribution: Puzzle box and Report   1. Name (U#) Dariia Bulhakova   Project Contribution: Puzzle box |
| No. of Hours Spent: | 72 |
| Exercise Difficulty:  (Easy, Average, Hard) | Hard |
| Any Other Feedback: |  |

**Project 1 (50 pts):**

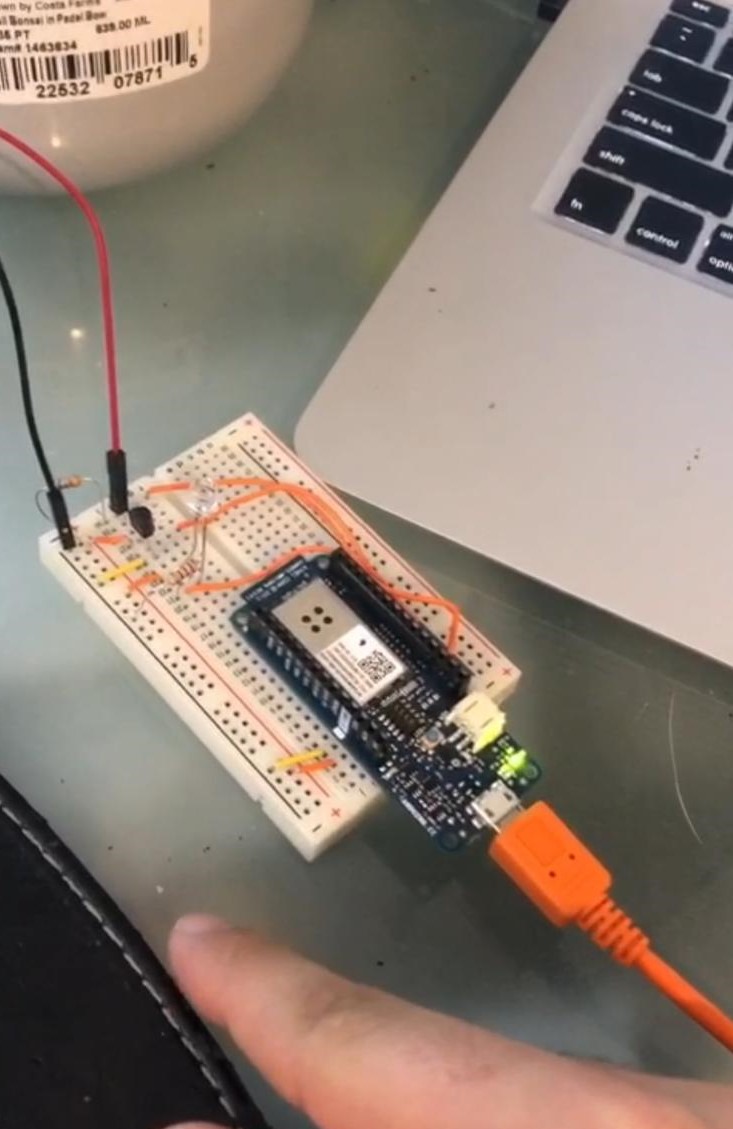
1. (5 pts) In your own words, describe the project.

For this project, we implement a Plant Communicator, in which we used Arduino to protect plants and make sure they survive. For this project, the Arduino monitors the plant temperature and light to check the condition of the plant. It sends emails and graphs about plant daily needs. The picture below shows us the result of what we received in the email.





1. (7.5 pts) A picture of **your** board setup.



1. (7.5 pts) A picture of **your** system setup.



1. (15 pts) Your sketch code.

#include <WiFi101.h>

#include<WiFiSSLClient.h>

#include <RTCZero.h>

const char\* ssid = "cupita\_wifi"; // your network SSID (name)

const char\* password = "dianetica"; // your network password

String httpsRequest = "https://hooks.zapier.com/hooks/catch/9062543/oewrs53/"; // your Zapier URL

const char\* host = "hooks.zapier.com";

WiFiSSLClient client;

RTCZero rtc; // create RTC object

/\* Change these values to set the current initial time \*/

const byte seconds = 0;

const byte minutes = 8;

const byte hours = 23;

/\* Change these values to set the current initial date \*/

const byte day = 6;

const byte month = 12;

const byte year = 20;

int lightPin = A0; //the analog pin the light sensor is connected to

int tempPin = A2; //the analog pin the TMP36's Vout (sense) pin is connected to

int moisturePin= A3;

// Set this threeshold accordingly to the resistance you used

// The easiest way to calibrate this value is to test the sensor in both dry and wet earth

int threeshold= 800;

bool alert\_already\_sent=false;

bool email\_already\_sent=true;

void setup() {

Serial.begin(9600);

while(!Serial);

delay(2000);

Serial.print("Connecting Wifi: ");

Serial.println(ssid);

while (WiFi.begin(ssid, password) != WL\_CONNECTED) {

Serial.print(".");

delay(500);

}

Serial.println("");

Serial.println("WiFi connected");

rtc.begin(); // initialize RTC 24H format

rtc.setTime(hours, minutes, seconds);

rtc.setDate(day, month, year);

rtc.setAlarmTime(16, 1, 0); // Set the time for the Arduino to send the email

rtc.enableAlarm(rtc.MATCH\_HHMMSS);

rtc.attachInterrupt(alarmMatch);

}

void loop() {

Serial.print(get\_temperature());

Serial.println(" degrees C ");

Serial.print(get\_light());

Serial.println(" light");

Serial.print(get\_average\_moisture());

Serial.println(" moisture level");

Serial.println("");

delay(1000);

String warning="";

// Send an extra email only if the plant needs to be waterd

if(get\_average\_moisture() < threeshold && !alert\_already\_sent){

warning ="Warning your plant needs water !"; // Insert here your emergency message

warning.replace(" ", "%20");

send\_email(get\_temperature(), get\_average\_moisture(),get\_light(), warning);

alert\_already\_sent=true; // Send the alert only once

}

// Send the daily email

if(!email\_already\_sent){

send\_email(get\_temperature(), get\_average\_moisture(),get\_light(), warning);

email\_already\_sent=true;

}

}

float get\_temperature(){

int reading = analogRead(tempPin);

float voltage = reading \* 3.3;

voltage /= 1024.0;

// Print tempeature in Celsius

float temperatureC = (voltage - 0.5) \* 100 ; //converting from 10 mv per degree wit 500 mV offset

// Convert to Fahrenheit

float temperatureF = (temperatureC \* 9.0 / 5.0) + 32.0;

return temperatureC;

}

int get\_average\_moisture(){ // make an average of 10 values to be more acurate

int tempValue=0; // variable to temporarly store moisture value

for(int a=0; a<10; a++){

tempValue+=analogRead(moisturePin);

delay(10);

}

return tempValue/10;

}

int get\_light(){

int light\_value=analogRead(A0);

return light\_value;

}

void alarmMatch(){ // triggered when the alarm goes on

Serial.println("Alarm Match!");

email\_already\_sent = false;

alert\_already\_sent = false;

}

void send\_email(float temperature, int moisture, int light, String warning){

// convert values to String

String \_temperature = String(temperature);

String \_moisture = String(moisture);

String \_light = String(light);

String \_warning = warning;

if (client.connect(host, 443)) {

client.println("POST "+httpsRequest+"?warning="+\_warning +"%0A"+"temperature="+\_temperature +"%0A"+"moisture="+\_moisture+"%0A"+"light="+\_light+" HTTP/1.1");

client.println("Host: "+ String(host));

client.println("Connection: close");

client.println();

delay(1000);

while (client.available()) { // Print on the console the answer of the server

char c = client.read();

Serial.write(c);

}

client.stop(); // Disconnect from the server

}

else {

Serial.println("Failed to connect to client");

}

}



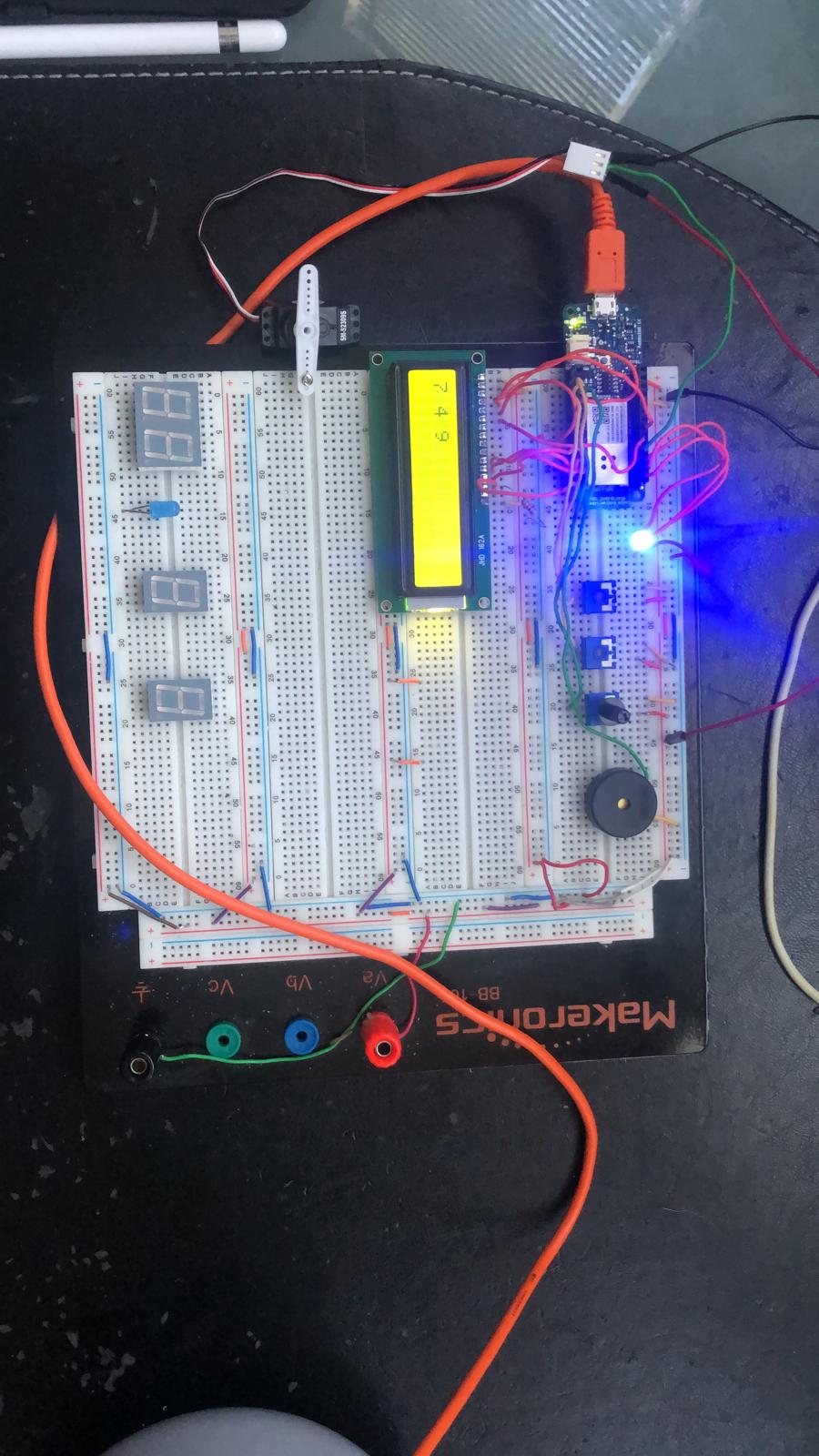


1. (15 pts) A short video of your functioning IoT System.

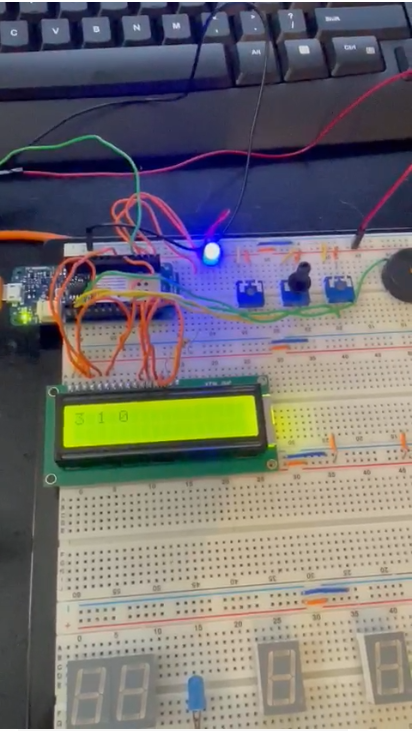
https://youtu.be/WCNaB\_jQmG0

**Project 2 (50 pts):**

1. (5 pts) In your own words, describe the project.
2. For this project, we implement a Puzzle box. The puzzle box is a security system to keep your belongings safe, base on the combination. The puzzle box works by opening a gate which is help closed with a servo motor by getting the right combination. To set up the combination we used the Blynk app, in which an LED light will help you guess, by giving you color feedbacks, in which the closer you are the warmer the color. When you get the right combination, the buzzer will start playing a song when the motor servo opens the gate.
3. (7.5 pts) A picture of **your** board setup.



1. (7.5 pts) A picture of **your** system setup.



1. (15 pts) Your sketch code.

#include "arduino\_secrets.h"

#include <LiquidCrystal.h>

#include <SPI.h>

#include <WiFi101.h>

#include <BlynkSimpleWiFiShield101.h>

#include <Servo.h>

#define buzzerPin 1

#include "Melody.h"

// RGB LED pins

int redPin = 6;

int greenPin = 8;

int bluePin = 7;

const char\* ssid = SECRET\_SSID; // your network SSID (name)

const char\* password = SECRET\_PSWD; // your network password

char auth[] = SECRET\_TOKEN; // your Blynk API token

// LCD screen pins

const int rs = 12,

en = 11,

d4 = 2,

d5 = 3,

d6 = 4,

d7 = 5;

bool start = true;

// Variables to store the combination value

// Set the intitial combination to ( 1 1 1 )

int SliderValueOne = 1;

int SliderValueTwo = 1;

int SliderValueThree = 1;

int pos = 0; // variable to store the servo position

Servo myservo; // create servo object to control a servo

// Blynk functions to retrive values

BLYNK\_WRITE(V1) {

SliderValueOne = param.asInt(); // assigning incoming value from pin V1 to a variable

}

BLYNK\_WRITE(V2) {

SliderValueTwo = param.asInt(); // assigning incoming value from pin V1 to a variable

}

BLYNK\_WRITE(V3) {

SliderValueThree = param.asInt(); // assigning incoming value from pin V1 to a variable

}

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {

pinMode(redPin, OUTPUT);

pinMode(greenPin, OUTPUT);

pinMode(bluePin, OUTPUT);

pinMode(buzzerPin, OUTPUT);

analogWrite(A3, 0); // set the brightness of the LCD screen to the maximum value

Serial.begin(9600);

lcd.begin(16, 2); // begin LCD screen with 16 columns and 2 rows

Blynk.begin(auth, ssid, password); // start Blynk functionalities

myservo.attach(9); // attaches the servo on pin 9 to the servo object

myservo.write(pos); // set the servo in position 0

}

void loop() {

// Variambles to temporarily store the combination

int Temp\_Slider\_One\_value = SliderValueOne;

int Temp\_Slider\_Two\_value = SliderValueTwo;

int Temp\_Slider\_Three\_value = SliderValueThree;

Blynk.run(); // poll new combination values from the online app

// check if combination values are changed and print them on the console

if(Temp\_Slider\_One\_value != SliderValueOne || Temp\_Slider\_Two\_value != SliderValueTwo || Temp\_Slider\_Three\_value != SliderValueThree){

Serial.print("New combination: ");

Serial.print(SliderValueOne);

Serial.print(" ");

Serial.print(SliderValueTwo);

Serial.print(" ");

Serial.println(SliderValueThree);

}

int PotOne = map(analogRead(A0), 100, 1023, 0, 9);

int PotTwo = map(analogRead(A1), 100, 1023, 0, 9);

int PotThree = map(analogRead(A2), 100, 1023, 0, 9);

lcd.setCursor(0, 0);

lcd.print(PotOne);

lcd.setCursor(2, 0);

lcd.print(PotTwo);

lcd.setCursor(4, 0);

lcd.print(PotThree);

if (start) {

giveColorFeedback(PotOne, PotTwo, PotThree);

if (PotOne == SliderValueOne && PotTwo == SliderValueTwo && PotThree == SliderValueThree) {

play\_jingle();

open\_the\_box();

blinkGreenLed();

start = false;

}

}

if(!start) {

if(PotOne == 0 && PotTwo == 0 && PotThree == 0){

close\_the\_box();

start = true;

}

}

}

// Give feedback based on how close the potentiometer are to the combination value

// The more it's close the warmer is the color of the LED

void giveColorFeedback(int PotOne, int PotTwo, int PotThree) {

if (abs(PotOne - SliderValueOne) <= 1 && abs(PotTwo - SliderValueTwo) <= 1 && abs(PotThree - SliderValueThree) <= 1 ) {

// Red

setColor(255, 0, 0);

}

else if (abs(PotOne - SliderValueOne) <= 3 && abs(PotTwo - SliderValueTwo) <= 3 && abs(PotThree - SliderValueThree) <= 3 ) {

// yellow

setColor(255, 255, 0);

}

else if (abs(PotOne - SliderValueOne) <= 4 && abs(PotTwo - SliderValueTwo) <= 4 && abs(PotThree - SliderValueThree) <= 4 ) {

// aqua

setColor(0, 255, 255);

}

else {

// blue

setColor(0, 0, 255);

}

}

void blinkGreenLed() {

for (int a = 0; a < 2; a++) {

for (int b = 0; b <= 255; b += 5) {

setColor(0, b, 0);

delay(5);

}

for (int b = 255; b >= 0; b -= 5) {

setColor(0, b, 0);

delay(5);

}

}

for (int b = 0; b <= 255; b += 5) {

setColor(0, b, 0);

delay(5);

}

}

// Send RGB values to the LED pins

void setColor(int red, int green, int blue){

analogWrite(redPin, red);

analogWrite(greenPin, green);

analogWrite(bluePin, blue);

}

void open\_the\_box(){

for (pos = 0; pos <= 90; pos += 1) { // goes from 0 degrees to 90 degrees

myservo.write(pos); // tell servo to go to position in variable 'pos'

delay(15); // waits 15ms for the servo to reach the position

}

}

void close\_the\_box(){

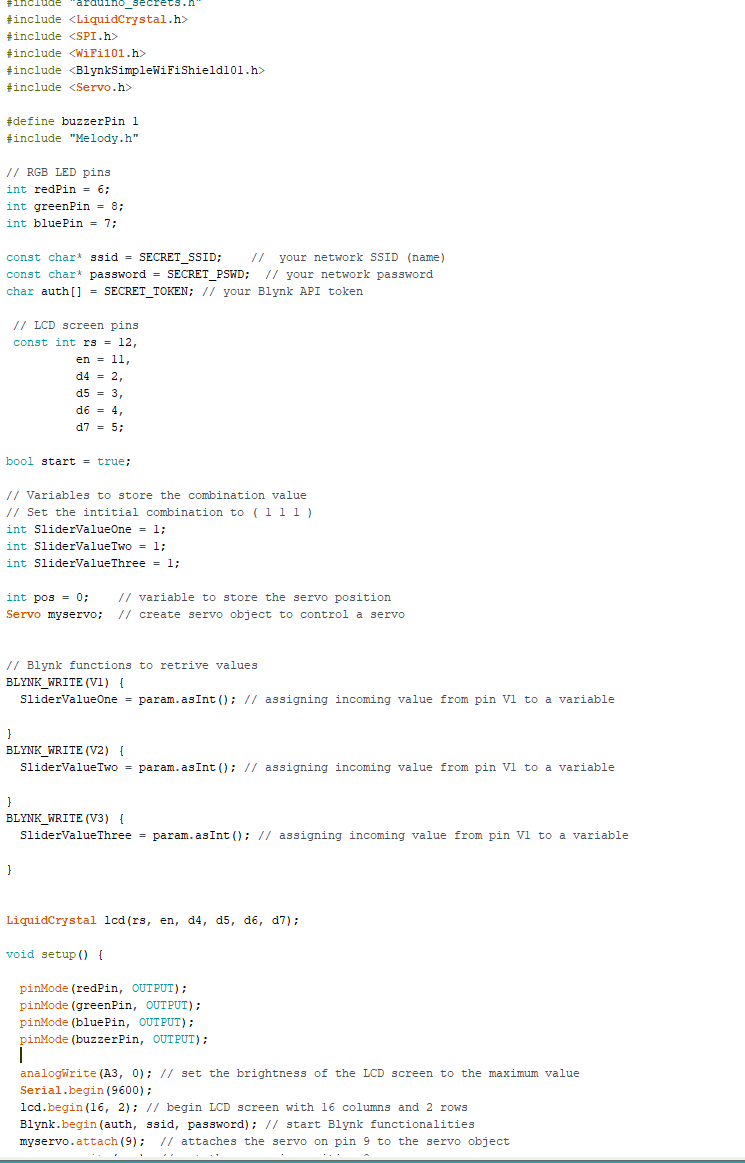
for (pos = 90; pos >= 0; pos -= 1) { // goes from 90 degrees to 0 degrees

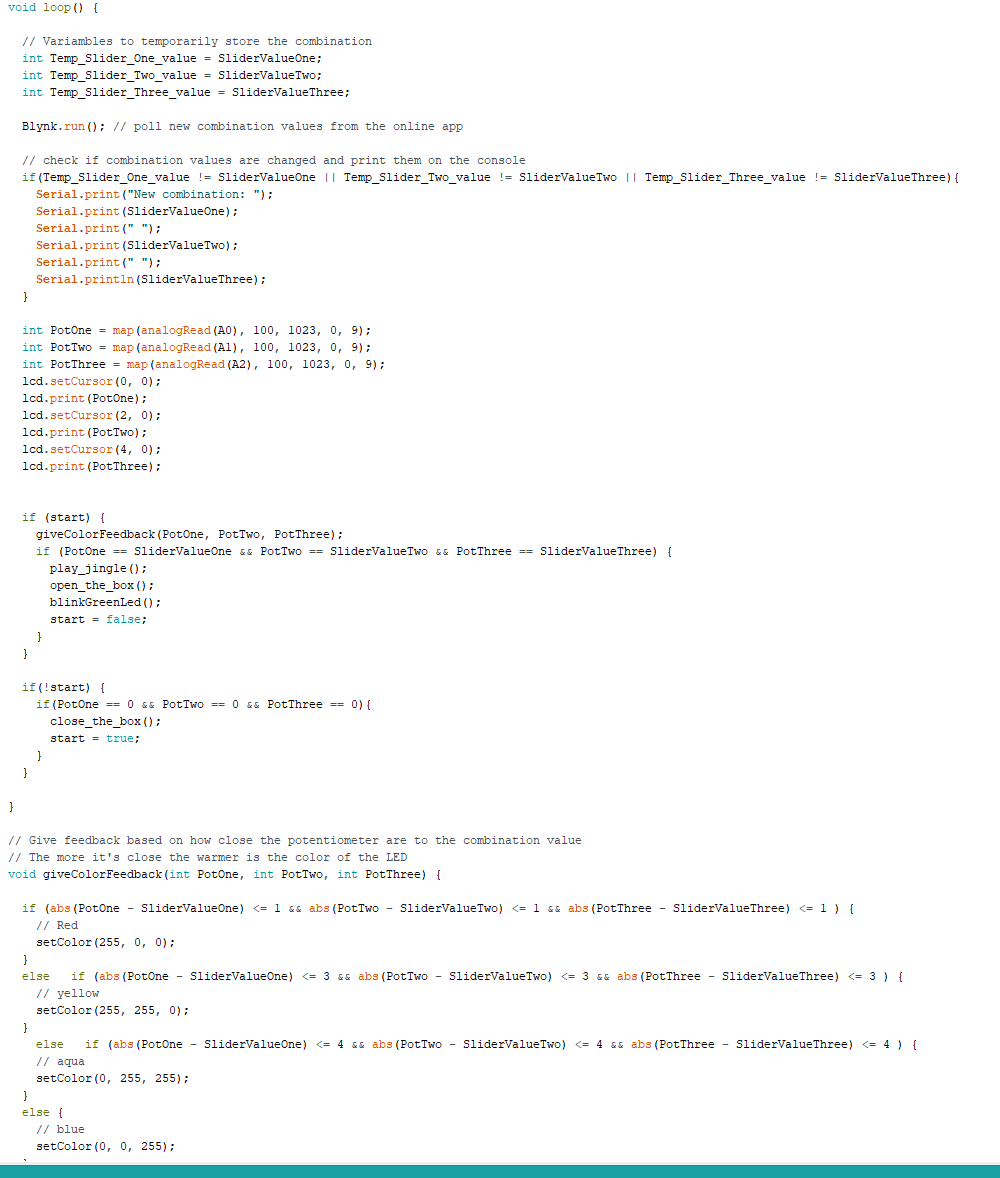
myservo.write(pos); // tell servo to go to position in variable 'pos'

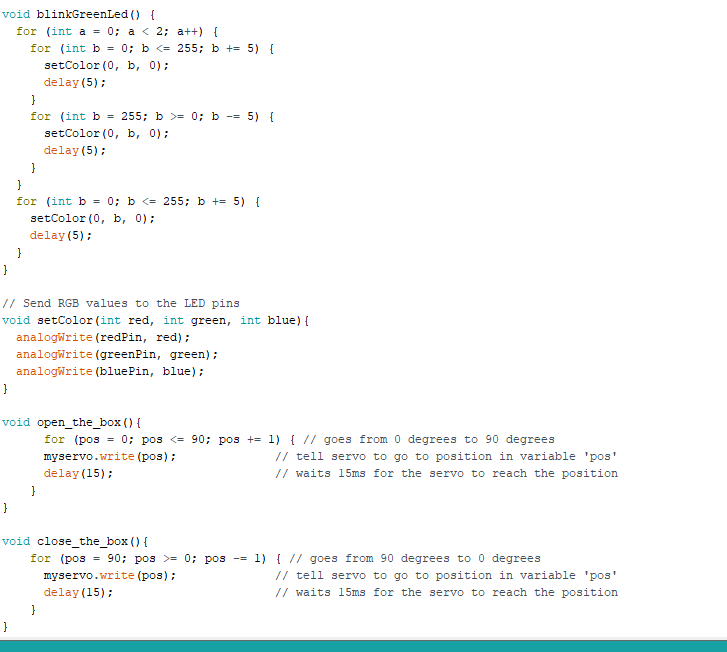
delay(15); // waits 15ms for the servo to reach the position

}

}







1. (15 pts) A short video of your functioning IoT System.

https://youtu.be/BEGmcbKFRsQ