

Final Project

Preston Carroll

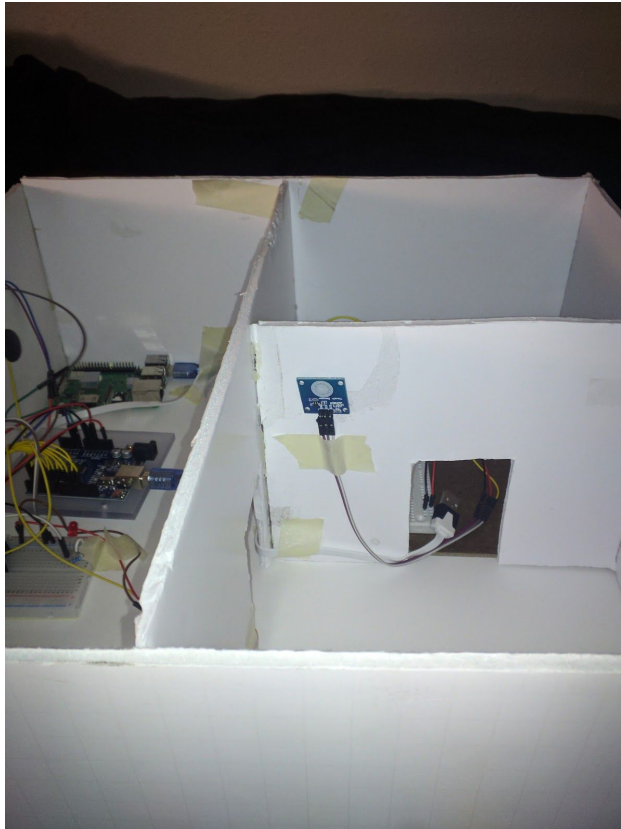
CSCI 497K

IoT: Internet Sensor Network Programming

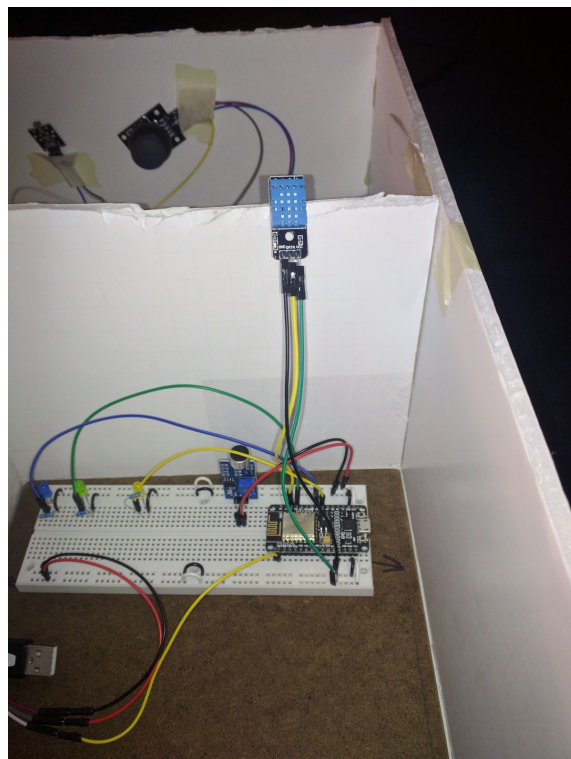
Saturday, December 15

Photos

Touch Sensor Controlling the Blue LED



Humidity/Temperature Sensor Sending Data to Node Red



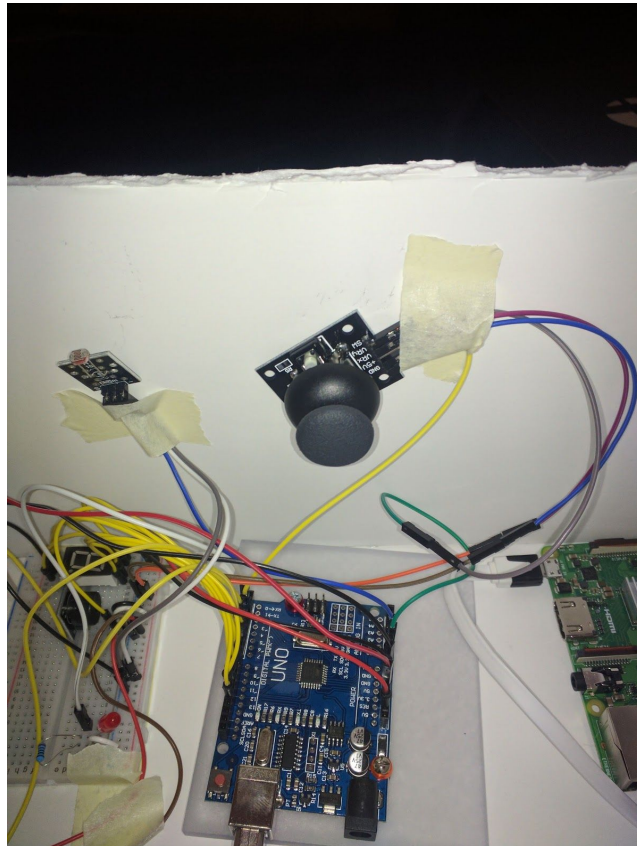
Motion Sensor triggering 7-segment countdown and buzzer



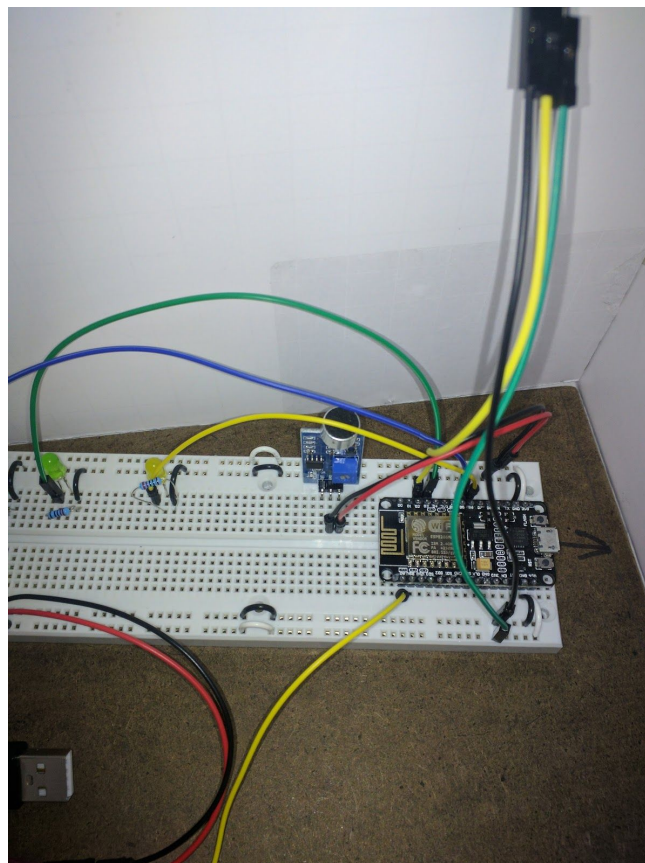
Motor for opening the curtains - connected to the joystick



Joystick to move motor back and forth



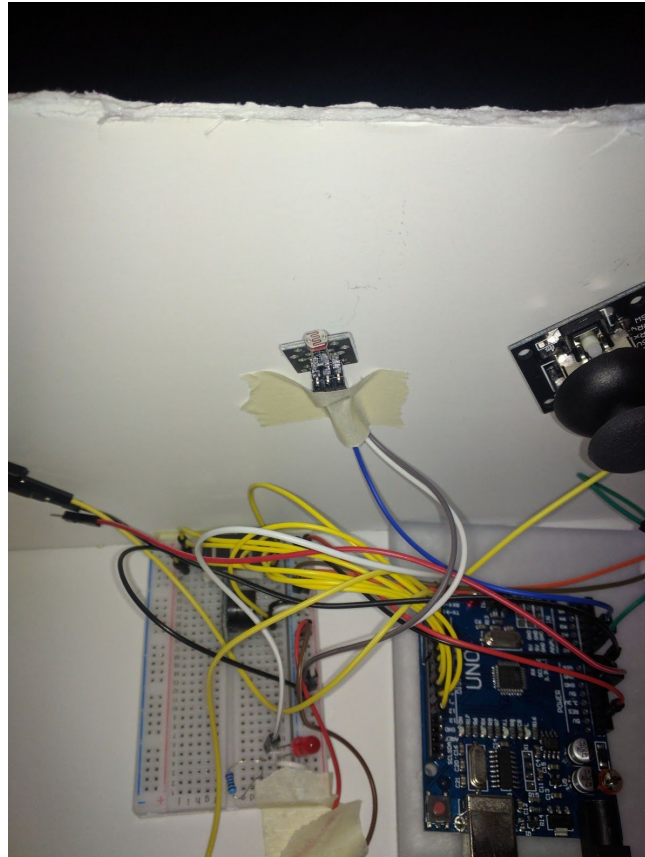
Sound sensor



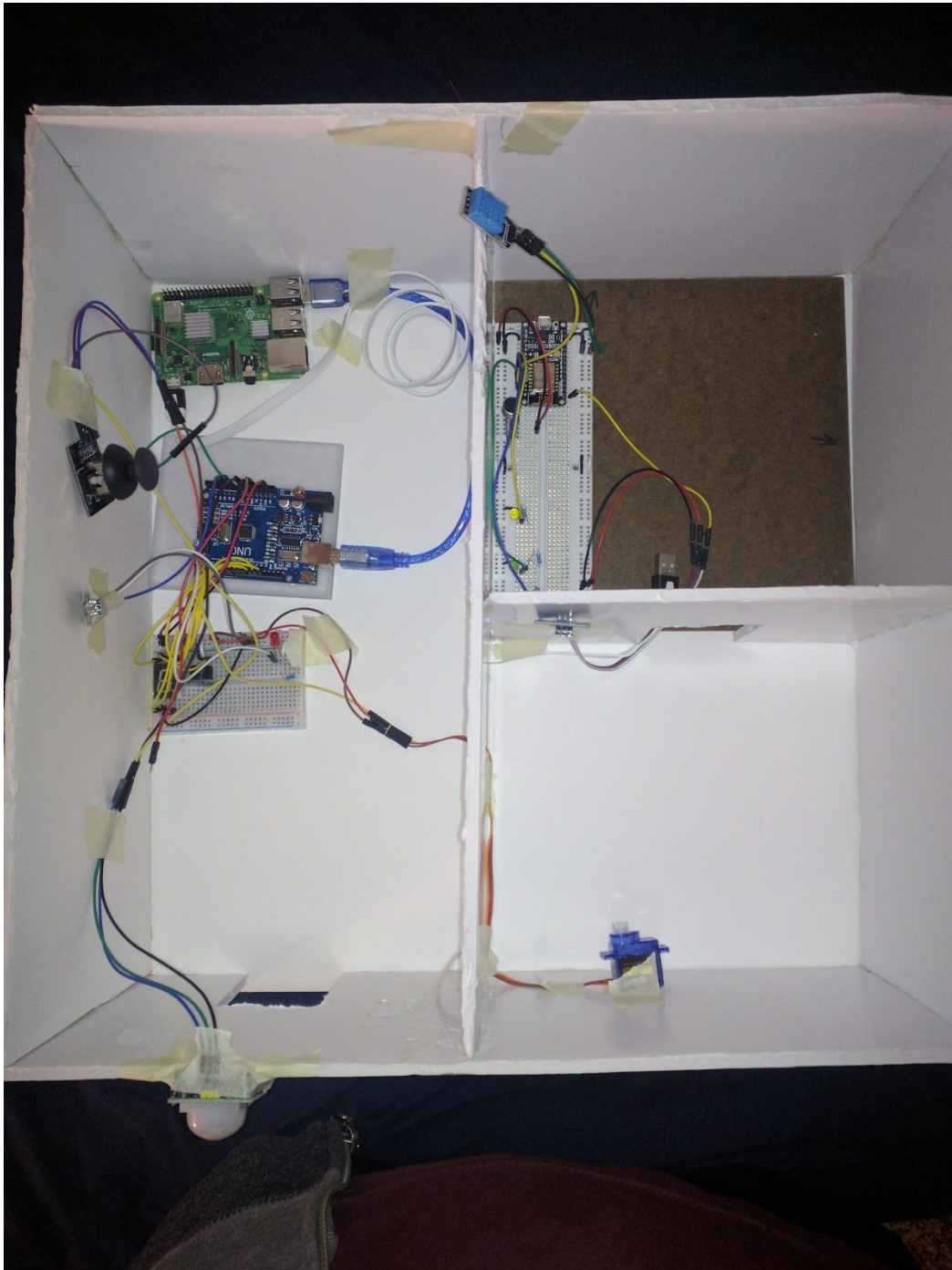
7-segment that displays countdown timer



Photoresistor sensor - lights up red led when exposed to bright light



Complete House Photo



Software

ESP Arduino Code

```
/******

All the resources for this project:
http://randomnerdtutorials.com/

*****/
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "DHT.h"

// Uncomment one of the lines bellow for whatever DHT sensor type you're using!
#define DHTTYPE DHT11 // DHT 11
// #define DHTTYPE DHT21 // DHT 21 (AM2301)
// #define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

// Change the credentials below, so your ESP8266 connects to your router
const char* ssid = "AP";
const char* password = "Preslton";

// Change the variable to your Raspberry Pi IP address, so it connects to your
MQTT broker
const char* mqtt_server = "192.168.43.90";

// Initializes the espClient. You should change the espClient name if you have
multiple ESPs running in your home automation system
WiFiClient espClient;
PubSubClient client(espClient);

// DHT Sensor - 4 = D2 on ESP-12E NodeMCU board
const int DHTPin = 4;
// Lamp - LED - 12 = D6 on ESP-12E NodeMCU board
const int lamp = 12;

// Touch sensor LED D5 = 14
const int LED_touch = 14;
//threshold
int thresholdValue = 400;

//Sound Sensor Pin d4 = 2
const int sound_sensor = 2;
//LED symbolizing the door opening via sound d3 = 0
const int LED_sound = 0;

// Initialize DHT sensor.
DHT dht(DHTPin, DHTTYPE);

// Timers auxiliar variables
long now = millis();
long lastMeasure = 0;
```

```

// Don't change the function below. This functions connects your ESP8266 to
your router
void setup_wifi() {
    delay(10);
    // We start by connecting to a WiFi network
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("WiFi connected - ESP IP address: ");
    Serial.println(WiFi.localIP());
}

// This functions is executed when some device publishes a message to a topic
that your ESP8266 is subscribed to
// Change the function below to add logic to your program, so when a device
publishes a message to a topic that
// your ESP8266 is subscribed you can actually do something
void callback(String topic, byte* message, unsigned int length) {
    Serial.print("Message arrived on topic: ");
    Serial.print(topic);
    Serial.print(". Message: ");
    String messageTemp;

    for (int i = 0; i < length; i++) {
        Serial.print((char)message[i]);
        messageTemp += (char)message[i];
    }
    Serial.println();

    // Feel free to add more if statements to control more GPIOs with MQTT

    // If a message is received on the topic room/lamp, you check if the message
    is either on or off. Turns the lamp GPIO according to the message
    if(topic=="room/lamp"){
        Serial.print("Changing Room lamp to ");
        if(messageTemp == "true"){
            digitalWrite(lamp, HIGH);
            Serial.print("On");
        }
        else if(messageTemp == "false"){
            digitalWrite(lamp, LOW);
            Serial.print("Off");
        }
    }
    Serial.println();
}

// This functions reconnects your ESP8266 to your MQTT broker
// Change the function below if you want to subscribe to more topics with your
ESP8266
void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {

```



```

    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    /*
    YOU MIGHT NEED TO CHANGE THIS LINE, IF YOU'RE HAVING PROBLEMS WITH MQTT
    MULTIPLE CONNECTIONS
    To change the ESP device ID, you will have to give a new name to the
    ESP8266.
    Here's how it looks:
    if (client.connect("ESP8266Client")) {
    You can do it like this:
    if (client.connect("ESP1_Office")) {
    Then, for the other ESP:
    if (client.connect("ESP2_Garage")) {
    That should solve your MQTT multiple connections problem
    */
    if (client.connect("ESP8266Client")) {
    Serial.println("connected");
    // Subscribe or resubscribe to a topic
    // You can subscribe to more topics (to control more LEDs in this
example)
    client.subscribe("room/lamp");
    } else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    // Wait 5 seconds before retrying
    delay(5000);
    }
    }
}

//checks if the touch sensor has been pushed, if so turn on the light
void touch_sensor_check() {
    int sensorValue = analogRead(A0);
    Serial.println(sensorValue);
    if(sensorValue>thresholdValue) {
        digitalWrite(LED_touch,HIGH);
    }
    else{
        digitalWrite(LED_touch,LOW);
    }
}

//checks if the sound sensor has been activate, if so turn on the light meaning
the door is opened
void sound_sensor_check() {
    int val = analogRead(A0);
    Serial.print("Sound is ");
    Serial.println(val);
    if(val==HIGH) {
        Serial.println("turning on sound");
        digitalWrite(LED_sound,HIGH);
    }
    else{
        Serial.println("turning off sound");
        digitalWrite(LED_sound,LOW);
    }
}

```

```

// The setup function sets your ESP GPIOs to Outputs, starts the serial
communication at a baud rate of 115200
// Sets your mqtt broker and sets the callback function
// The callback function is what receives messages and actually controls the
LEDs
void setup() {
    pinMode(lamp, OUTPUT);
    pinMode(LED_touch, OUTPUT);
    pinMode(LED_sound, OUTPUT);

    pinMode(sound_sensor, INPUT);

    dht.begin();

    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
}

// For this project, you don't need to change anything in the loop function.
Basically it ensures that you ESP is connected to your broker
void loop() {

    if (!client.connected()) {
        reconnect();
    }
    if(!client.loop())
        client.connect("ESP8266Client");

    now = millis();
    // Publishes new temperature and humidity every 30 seconds
    if (now - lastMeasure > 2000) {
        lastMeasure = now;
        // Sensor readings may also be up to 2 seconds 'old' (its a very slow
sensor)
        float h = dht.readHumidity();
        // Read temperature as Celsius (the default)
        float t = dht.readTemperature();
        // Read temperature as Fahrenheit (isFahrenheit = true)
        float f = dht.readTemperature(true);

        // Check if any reads failed and exit early (to try again).
        if (isnan(h) || isnan(t) || isnan(f)) {
            Serial.println("Failed to read from DHT sensor!");
            return;
        }

        // Computes temperature values in Celsius
        float hic = dht.computeHeatIndex(t, h, false);
        static char temperatureTemp[7];
        dtostrf(hic, 6, 2, temperatureTemp);

        // Uncomment to compute temperature values in Fahrenheit
        // float hif = dht.computeHeatIndex(f, h);
        // static char temperatureTemp[7];
        // dtostrf(hic, 6, 2, temperatureTemp);

        static char humidityTemp[7];

```

```

dtostrf(h, 6, 2, humidityTemp);

// Publishes Temperature and Humidity values
client.publish("room/temperature", temperatureTemp);
client.publish("room/humidity", humidityTemp);

Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %\t Temperature: ");
Serial.print(t);
Serial.print(" *C ");
Serial.print(f);
Serial.print(" *F\t Heat index: ");
Serial.print(hic);
Serial.println(" *C ");
// Serial.print(hif);
// Serial.println(" *F");

touch_sensor_check();
//sound_sensor_check();

```

Arduino Code

```

#include <Servo.h>

int A = 13;
int B = 12;
int C = 11;
int D = 10;
int E = 9;
int F = 8;
int G = 7;

int buzzer = 6;
int motion_sensor = 5;
int joystick = A0; //VRx
int photoresistor = A1;
int motor = 4;

int joy_click = 3;
int RED_led = 2;

int LRMID = 0;
int MID = 10;
int LR = 0;
int state = LOW;

int var1 = 0; // for motion sensor reads
int var2 = 0; //for joystick click reads

int pos = 0; // for servo

int photo_sensor_val = 0;

Servo servo; //servo object

```

```

//move the servo 90 degrees
void move90Degrees(){
    // moves servo from 0 degrees to 90 degrees
    for (; pos <= 90; pos += 1) {
        // in steps of 1 degree
        servo.write(pos); // tell servo to go to position in variable 'pos'
        delay(15); // waits 15ms for the servo to reach the position
    }
}

//move the servo 90 degrees
void moveBack(){
    // moves servo from 0 degrees to 90 degrees
    for (; pos >= 0; pos -= 1) {
        // in steps of 1 degree
        servo.write(pos); // tell servo to go to position in variable 'pos'
        delay(15); // waits 15ms for the servo to reach the position
    }
}

void setup() {
    // put your setup code here, to run once:

    Serial.begin(115200);
    pinMode(A, OUTPUT);
    pinMode(B, OUTPUT);
    pinMode(C, OUTPUT);
    pinMode(D, OUTPUT);
    pinMode(E, OUTPUT);
    pinMode(F, OUTPUT);
    pinMode(G, OUTPUT);
    pinMode(RED_led, OUTPUT);

    pinMode(buzzer, OUTPUT);
    pinMode(motor, OUTPUT);

    pinMode(motion_sensor, INPUT);
    pinMode(joystick, INPUT);
    pinMode(photoresistor, INPUT);
    pinMode(joy_click, INPUT);

    LRMID = analogRead(joystick);

    servo.attach(motor);
    servo.write(pos);
    delay(1000);
}

void loop() {
    // put your main code here, to run repeatedly:

    //motion sensor code
    var1 = digitalRead(motion_sensor);
    if(var1 == HIGH){ //activation code
        Serial.println("Motion Detected");
        digitalWrite(RED_led, HIGH);
        delay(100);
        digitalWrite(RED_led, LOW);
    }
}

```

```

        delay(100);
        digitalWrite(REDA_led, HIGH);
        delay(100);
        digitalWrite(REDA_led, LOW);
        delay(2000);
        if(state == LOW){
            state = HIGH;
        }
    }
else{ //non-activation code
    if(state == HIGH){
        state = LOW;
    }
noTone(buzzer);
}

//7-segment and buzzer code
var2 = digitalRead(motion_sensor);
Serial.print("var2 is");
Serial.println(var2);
delay(10);
if(var2 == HIGH){
    Serial.println("Clicked");
    //3
    digitalWrite(A, HIGH);
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
    digitalWrite(D, HIGH);
    digitalWrite(G, HIGH);
    delay (1000);
    digitalWrite(C, LOW);

    //2
    digitalWrite(A, HIGH);
    digitalWrite(B, HIGH);
    digitalWrite(G, HIGH);
    digitalWrite(E, HIGH);
    digitalWrite(D, HIGH);
    delay(1000);
    digitalWrite(A, LOW);
    digitalWrite(G, LOW);
    digitalWrite(E, LOW);
    digitalWrite(D, LOW);

    //1
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
    delay(1000);

    //0
    digitalWrite(A, HIGH);
    digitalWrite(B, HIGH);
    digitalWrite(C, HIGH);
    digitalWrite(D, HIGH);
    digitalWrite(E, HIGH);
    digitalWrite(F, HIGH);
    delay(1000);
    digitalWrite(A, LOW);
    digitalWrite(B, LOW);
    digitalWrite(C, LOW);

```



```

    digitalWrite(D, LOW);
    digitalWrite(E, LOW);
    digitalWrite(F, LOW);

    tone(buzzer, 500);
    delay(500);
    noTone(buzzer);

    var2= LOW;
}

//joy stick to motor code
LR = analogRead(joystick);
if (LR > LRMID + MID) { //right
    Serial.print("pos is ");
    Serial.println(pos);
    if(pos < 360){
        Serial.println("Right!");
        pos++;
        servo.write(pos); // tell servo to go to position in variable 'pos'
        delay(15); // waits 15ms for the servo to reach the position
    }
}

if (LR < LRMID - MID) { //left
    Serial.print("pos is ");
    Serial.println(pos);
    if(pos > 0){
        Serial.println("Left!");
        pos--;
        servo.write(pos); // tell servo to go to position in variable 'pos'
        delay(15); // waits 15ms for the servo to reach the position
    }
}

//photo resistor code
photo_sensor_val = analogRead(photoresistor);
if(photo_sensor_val < 130){
    digitalWrite(RED_led, HIGH);
}
else {
    digitalWrite(RED_led, LOW);
}
//delay(30);
}

```

Readme

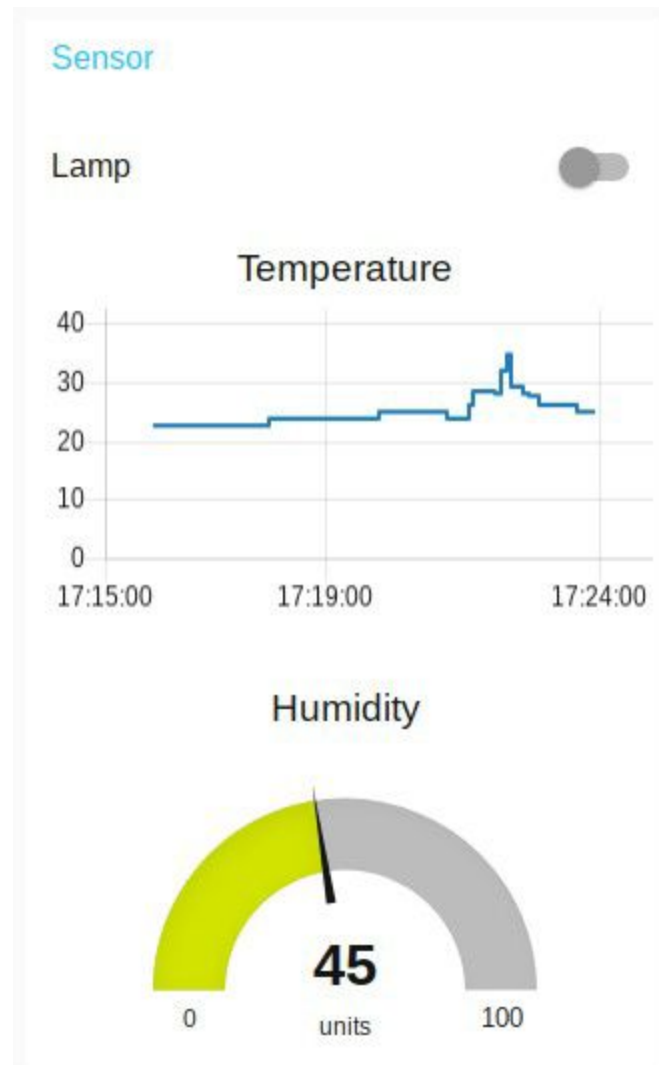
To run this house I plug in the Arduino, Raspberry Pi, and ESP into a power source. From after a few moments Node Red will be running and the house is good to go. The only sensor that does not have full functionality is the sound sensor. It will still light up on the device itself if it detects sound, but to get any readings on a device was futile.

The Arduino does not have to connect to Node Red and is just run as a stand alone device since the sensors connected to it all interact locally.

Sensor Breakdown

a.) Light Control - Pressing the touch sensor will turn on the Blue LED on the ESP's breadboard. Additionally, pressing the toggle on Node Red will turn on/off the Yellow LED, which represents the door being locked

b.) Temperature Control - The DHT sensor connected to the ESP reads temperature from Node Red and displays it on a graph



c.) Humidity Sensor - this comes from the DHT sensor and is also displayed on Node Red

d.) Motion Sensor - the motion sensor connected to the Arduino will make the Red LED flash twice and then a countdown timer is activated and the buzzer rings.

e.) Door Lock - The green LED connected to the ESP and to Node Red signifies if the door is locked. On being locked and off being unlocked.

f.) Open and close curtains - The moving the joystick left and right will turn the motor clockwise or counter clockwise so that the curtains are moving back and forth. This is on the Arduino

g.) Touch Sensor - The touch sensor is connected to the ESP and it will turn on the blue LED if it detects your finger

h.) Sound sensor - This sensor is connected to the ESP and the small LED on the device will light up green if sound is detected like a snap. Getting any readings from the device in a program proved to be futile.

i.) 7 segment - a countdown from 3 is shown when a motion event at the front door is detected. The 7-segment is connected to the arduino

j.) Joystick - The joystick is connected to the Arduino and will move the motor if you move it left and right.

k.) Miscellaneous interesting feature - *Photoresistor* - Shining a light at the photoresistor or bringing it outside in daylight will trigger the RED led to be lit up.