Kevin Le, kle34@uic.edu

Amanuel Odicho, aodich5@uic.edu

Teresa Pham, tpham54@uic.edu

Home Regulation System

With the use of several input and output devices, we will create a system that acts as both home regulation and security. The overall purpose of this project is to be able to implement a live alert system based on different conditions. For this specific project we have chosen to implement this idea using a box, where we store something valuable inside and we want to be alerted if someone tries to open it. Another aspect of the project is a temperature regulator, which will be able to alert the user when the temperature becomes uncomfortable.

Overall Description of Project Idea

The project will implement a few distinct ways of detecting unwanted behavior. Our idea is a security system for a box. This box should only be opened by the owner, thus, any attempt to open it by anyone else should be alerted to the owner. This idea can be further built out to a real life product integrated with a secure safe. Not only do you want to keep your items safe inside a password protected safe box, but you also want to be able to know immediately when someone attempts to open it so that you can prevent theft from happening live.

The other idea of temperature regulation mainly stems from not having an automated system of temperature regulation, therefore it is useful to be able to be automatically notified when to turn on/off a space heater or a fan. Most fans and space heaters have some automated system integrated, but they are usually not temperature specific. Our device will be able to alert the user immediately after the temperature drops or rises more than it should.

Initial Project Design Stating Expected Inputs/Outputs

We have two components to our project:

(1) "Box Security" is a device consisting of a photoresistor, a motion sensor and/or an ultrasonic sensor as input. Each input has its own distinct feature of detection. The photoresistor detects a change of light brightness, the motion sensor can detect movements, and the ultrasonic sensor can detect if an object's distance changes. Using the aforementioned inputs, we connect them to a Wi-Fi module, RGB light, and a buzzer as outputs. The module will send a notification to a phone through Wi-Fi technology, the RGB light will be programmed to turn on and off with the color red, and the buzzer will be adjusted to be very loud.

(2) "Temp Detect" consists of a temperature sensor as input. This sensor will be connected to a fan and an RGB light, LCD display, and a Wi-Fi-module acting as output. Once it gets too hot, the fan turns on and the RGB light turns red/orange. If it gets too cold, it will send a notification through Wi-Fi that it is too hot with a message such as "Turn on heater." There will be a state called comfort zone, which is a temperature interval a user can set in which they are comfortable

with. E.g., an interval between 68-70 degrees. Too cold being below 68, too hot being above 70 degrees.

Expected Plan for Communication

Our main plan for communication is to integrate a Wi-Fi module and set it up so that we can send notification to a phone. Preferably, we will be able to customize the message, when to send it, and also how often. Based on different conditions using logical expressions, we can determine what message to send and when to send it. Additionally we are attempting to communicate bi-directionally, so that we can send special instructions to our device from our phone. Otherwise, we can also always take advantage of serial communication.

Description of the Original Work Being Attempted by Your Project

This project utilizes a smartphone to receive notifications through Wi-Fi and to display information on the LCD display when a specific number is pressed. The smartphone will use the Wi-Fi module to access the website, pushsafer.com, to view notifications and access several buttons to communicate with the Arduino.

The Arduino fan speed controlled temperature system already exists, however, this project takes it one step further and incorporates the use of push notifications to send back signals to the Arduino to control the fan speed. As mentioned above in (2) of the initial project design, the LCD display is present to display the current temperature, and the Wi-Fi module sends notifications to the smartphone when it is too hot or cold. The user may then activate/deactivate the fan through their smartphone or let the RGB light automatically change fan speed with a click of a button.

For the security system, there have been projects where a buzzer sounds when motion or sound is detected. However, this project goes even further because the user can receive notifications when the ultrasonic sensor detects the box has been opened and chooses to disable the security system or check who the intruder is.

Timeline:

3/1 – 3/12 - Start working on a security device, get the components needed and start sketching out the circuit. Components, circuit and some code should be written by this time.

3/13 – 3/26 - Start working on the temperature detection system. Gather the components needed and start sketching out the circuit. Components, circuit and some code should be written by this time.

3/27 – 4/1 - Create video presentation following the guidelines, then record voiceover and submit.

4/2/21 – Milestone 4: Update this document to reflect changes needed for MS4. Submit MS4. Submit MS5 (Part 1).

4/3 – 4/21 - Finalize both components: Build the circuit, complete the code. Debug code if necessary. Test with real life inputs such as varying temperatures and opening and closing a box. Project should be finished by this period.

4/21 - 4/23 - Milestone 5 (part 2): Record a 5 minute video of the project and its functionality. Remember to include two first slides from part 1 presentation. Part 3: Each group member should write their own evaluation of the other team's presentation.

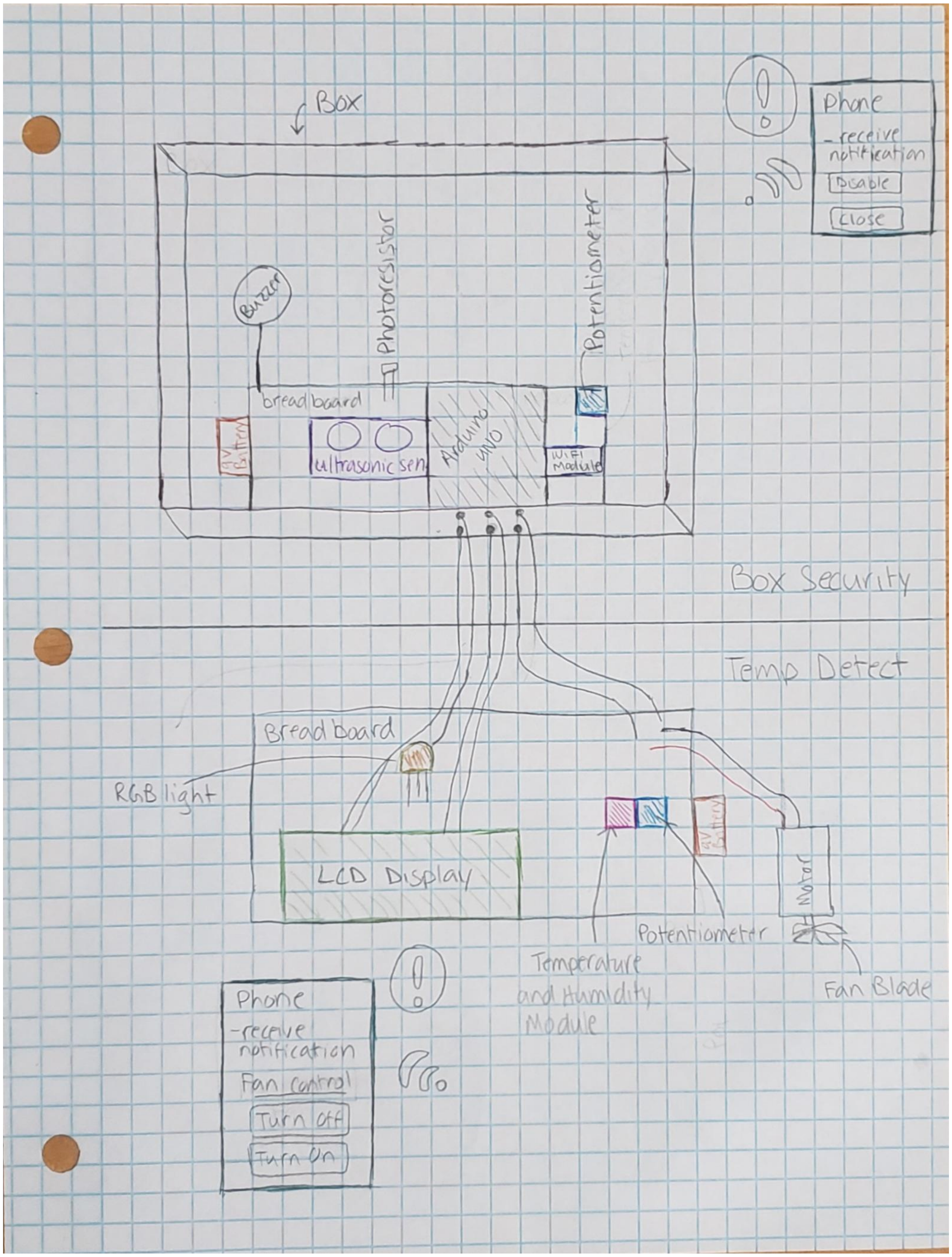
4/30/21 – Milestone 6: Peruse and refurbish this document for final submission. Revise any grammar or spelling mistakes. Make sure formatting and structure is in logical order.

5/3/21 - Milestone 7: Each member is responsible to write a teamwork assessment, assessing each member including themselves.

Materials Expected to be Needed:

- Arduino UNO - Qty: 1
- LCD1602 Module - Qty: 1
- RGB LED - Qty: 1
- Active Buzzer - Qty: 1
- ESP8266 NodeMCU CP2102 ESP-12E (wifi-module) - Qty: 1
- Ultrasonic Sensor - Qty: 1
- (maybe) Photoresistor - Qty: 1
- Fan Blade and 3-6V Motor - Qty: 1
- 9V Battery - Qty: 1
- Potentiometer - Qty: 2
- DHT11 Temperature and Humidity Module - Qty: 1
- Resistors - Qty: As many as needed
- Jumper wires - Qty: As many as needed.

Diagram:


Box
Phone
-receive notification
Disable
Close
Buzzer
Photoresistor
Potentiometer
breadboard
9V Battery
ultrasonic sen
Arduino UNO
WiFi Module
Box Security
Temp Detect
Bread board
RGB light
LCD Display
9V Battery
Motor
Potentiometer
Temperature and Humidity Module
Fan Blade
Phone
-receive notification
Fan control
Turn off
Turn On

Code sketch:

```
“Box Security”
const int pResistor = A1;
const int pBuzzer = 6;
const int led1 = 8;
int value, state;
unsigned long prevMillis = 0;
const long interval = 500;
const int trigPin = 9;
const int echoPin = 10;
float duration, distance, prevDistance, distanceChange;


// Initialize all inputs and outputs.
void setup() {
  pinMode(pResistor, INPUT);
  pinMode(pBuzzer, OUTPUT);
  pinMode(led1, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  value = analogRead(pResistor); // Reads in the value of
photoresistor.
  unsigned long currMillis = millis();


  //  Send out 8 cycle sonic burst from transmitter
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Use the sensory input to calculate distance and changed distance
  duration = pulseIn(echoPin, HIGH);
  distance = (duration*.0343)/2;
  distanceChange = distance - prevDistance;
  prevDistance = distance;


  // Determine if 0.5s have gone by and if photoresistor is detecting
light
  // Or if distance has changed
  if ((value >= 50 || distanceChange > 0)&& currMillis - prevMillis
>= interval) {
    prevMillis = currMillis;  // Update the millis variables
    if (state == LOW) {  // If the state is low, turn on
      state = HIGH;
    } else {
      state = LOW;  // else turn off
    }
  }

  // Write values to both outputs to turn on/off LED and Passive
Buzzer
  analogWrite(pBuzzer, state);
  digitalWrite(led1, state);

}
```

“Temp Detect”

```
#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

const int led_red_pin = 11;
const int led_green_pin = 10
const int led_blue_pin = 9;

const int motorPin = 3;

void setup() {
  pinMode(led_red_pin, OUTPUT);
  pinMode(led_green_pin, OUTPUT);
  pinMode(led_blue_pin, OUTPUT);
  pinMode(motorPin, OUTPUT);
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  float t = dht.readTemperature();
  float hif = dht.computeHeatIndex(f, h);  // computes temperature to
Fahrenheit


  // First check if temperature is too hot. If it is, then turn fan on, LED
red and notification via Wi-Fi.
  // Code to send wifi notification, will be implemented later
  if (hif > 73) {

  if (Serial.available())
  {
    // Turn on fan
    analogWrite(motorPin, 255);
  }
    RGB_color(255, 0, 0); // Red
  } else if (hif < 67) {  // Check if temperature is too cold
    RGB_color(0, 0, 255); // Blue
  } else {  // Temperature is within desired interval [67, 73]
    RGB_color(0, 255, 0); // Green
  }

}

// Source:
https://create.arduino.cc/projecthub/muhammad-aqib/arduino-rgb-led-tutorial
-fc003e
// This code just encodes RGB values to the LED.
void RGB_color(int red_light_value, int green_light_value, int
blue_light_value)
 {
  analogWrite(red_light_pin, red_light_value);
  analogWrite(green_light_pin, green_light_value);
  analogWrite(blue_light_pin, blue_light_value);
}
```

References:

ambhatt. “Temperature Controlled Fan with LCD Using Arduino.” *Arduino Project Hub*, create.arduino.cc/projecthub/ambhatt/temperature-controlled-fan-with-lcd-using-arduino-b0b25.

MisterBotBreak. “How to Use Temperature and Humidity (DHT) Sensors.” *Arduino Project Hub*,create.arduino.cc/projecthub/MisterBotBreak/how-to-use-temperature-and-humidity-dht-sensors-9e5975.

Paredes, Jeff. “Add WiFi to Arduino UNO.” *Arduino Project Hub*, create.arduino.cc/projecthub/imjeffparedes/add-wifi-to-arduino-uno-663b9e.

Pushsafer. “Push-Notifications with Pushsafer (IOS, Android, WIN10).” *Arduino Project Hub*, create.arduino.cc/projecthub/pushsafer/push-notifications-with-pushsafer-ios-android-win10-28aee4.

SURYATEJA. “USE a BUZZER MODULE (PIEZO SPEAKER) USING ARDUINO UNO.” *Arduino Project Hub*, create.arduino.cc/projecthub/SURYATEJA/use-a-buzzer-module-piezo-speaker-using-arduino-uno-89df45.