

ECE 110 Honor Project

Smart Energy Efficient Temperature Regulator

Yuji Chai

Christopher Willenborg

Rikin Mehta

Introduction:

We are addressing the problem that the dorm rooms get excessively hot in the winter time, but the temperature outside is very cool, which is quite ironic. Our design solves this problem by letting the heat escape the room by opening the window. Our design changes the position of the window from opened to closed or visa versa depending on the temperature inside the room. For example, if the temperature inside the room is greater than the desired temperature then the window will open a certain amount depending on the temperature difference of the temperature in the room and the desired temperature of the room, but if the temperature that we want it to be in the room is greater than that of the temperature inside, then the window will remain closed.

Design:

As for our design, we found out that trying to use the same voltage supply to the servo motor and temperature sensor was not going to work. We discovered that whenever the temperature sensor would send a value through our code that should change the angle on the servo motor it did, but when the servo motor would change angles it would vary the voltage supplied to the thermistor. When this happened, the temperature sensor would read in a crazy value into our code because of the lack of voltage and the servo would go crazy for a little bit moving back and forth. At the time we struggled to figure out what the servo was trying to do and why, but looking back it seems fairly straightforward what the issue was. To fix this we used another Arduino with a separate power supply for the motor.

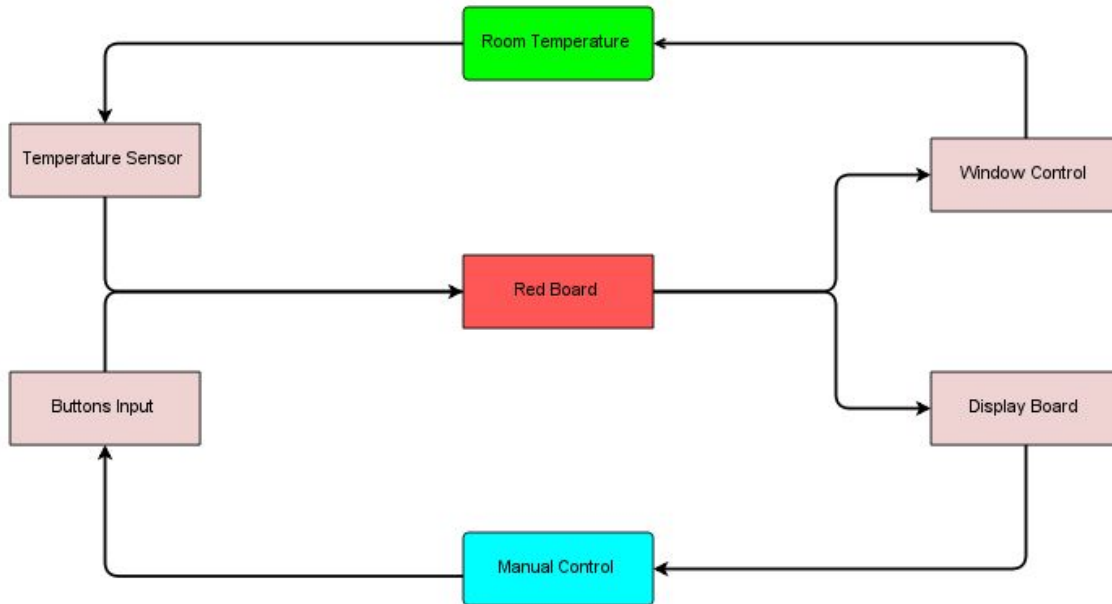
After we successfully constructed our design, we had a few lab sessions left until the demo day, so we decided to add a display for our temperatures. We were given a display that displayed four numbers. We considered using four hex. displays instead, but we decided that we would just use the one we were given to display our numbers. We first made our display, display the value of the temperature in the room for the first number and the targeted temperature for the second number. We wrote our code to display the temperature in Celsius, but we added a feature that could display the temperature in Fahrenheit with the push of a pushbutton. We made it to where you could also change the target temperature with two pushbuttons, one for increasing the temperature and one for decreasing the temperature.

The temperature sensor worked very well for our intents and purposes. The design and application of our code worked very well for the project that we were trying to put together. The window that the machine shop was a very nice last second addition because we believe that it makes our project look more professional for one and presentable.

We had a lot of small problems that were easy to fix, but hard to find the root of for our project. We learned how to break down each component of our circuit and rule out one portion at a time, until we were capable of isolating the problem we were experiencing. We decided to construct each component individually (the servo motor, the thermistor, and the display) and write the code for each individually until we had each individually component functioning the way in which we intended it too, We then had the interesting experience of integrating all three of their codes, so that they would

all work harmoniously for our function. This absorbed a larger amount of time than we had anticipated partially because there were delays in some of our code that would affect the other components, etc.

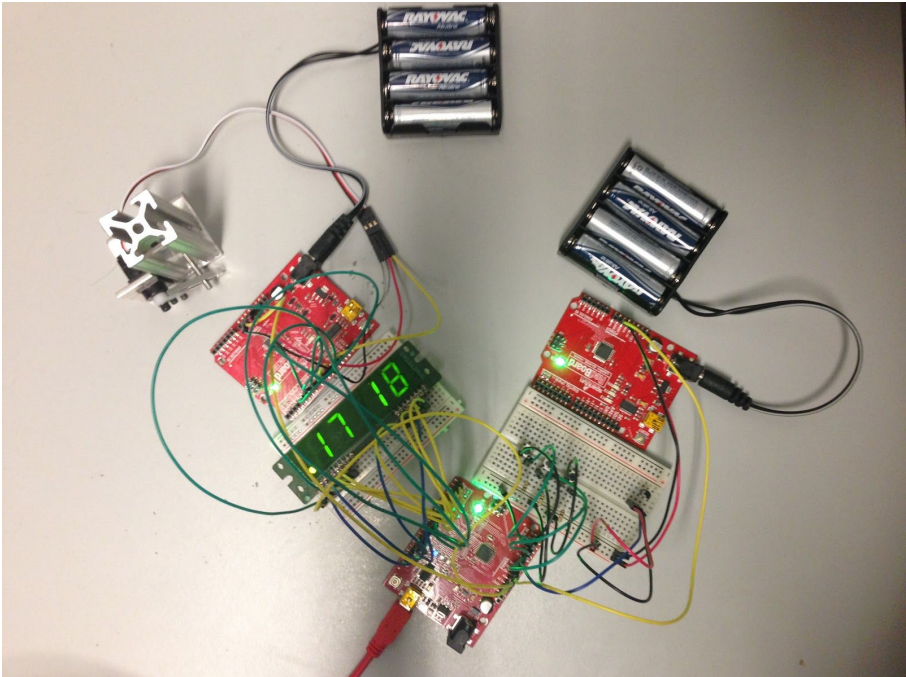
Block Diagram



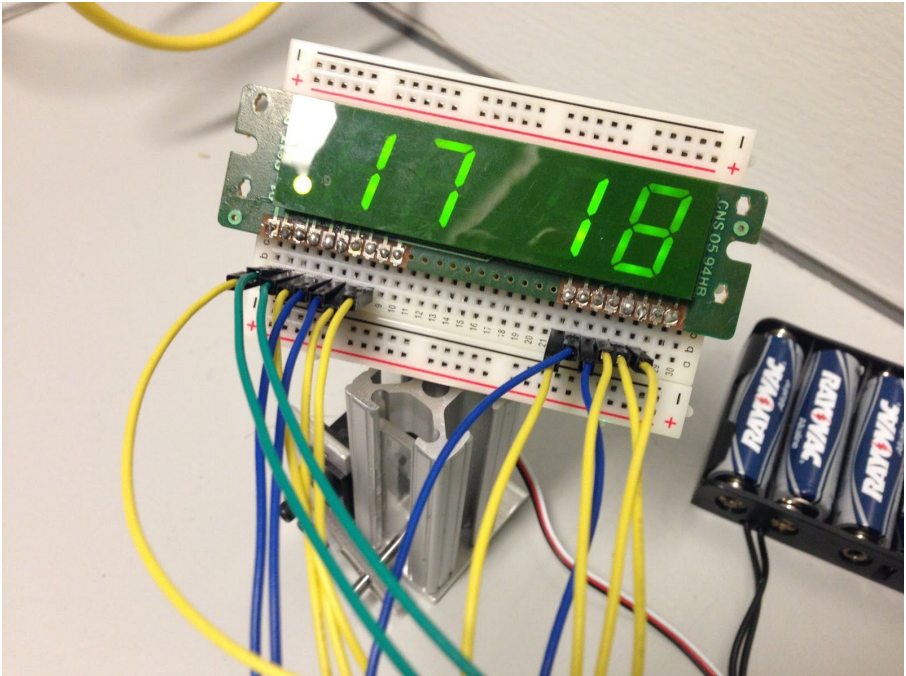
Block Diagram Description:

The red board in the center of the diagram is the central controlling processor. It will take the value read from the temperature sensor and the buttons input as the information input to compute what kind of signal that it would send to the display board and the window control. Then, the window control will affect the temperature either by opening the window or closing it. The display will tell user about the temperature inside the room and the preferred temperature. Those two information then will loop through the change of room temperature and manual control to change the inputs of the temperature sensor and the button inputs accordingly.

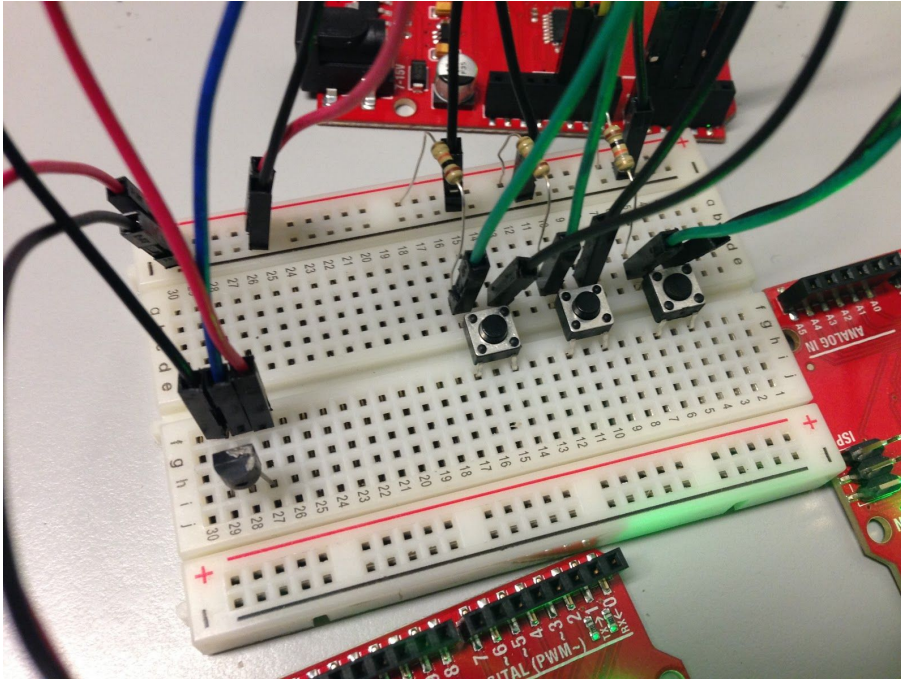
Picture of the device:



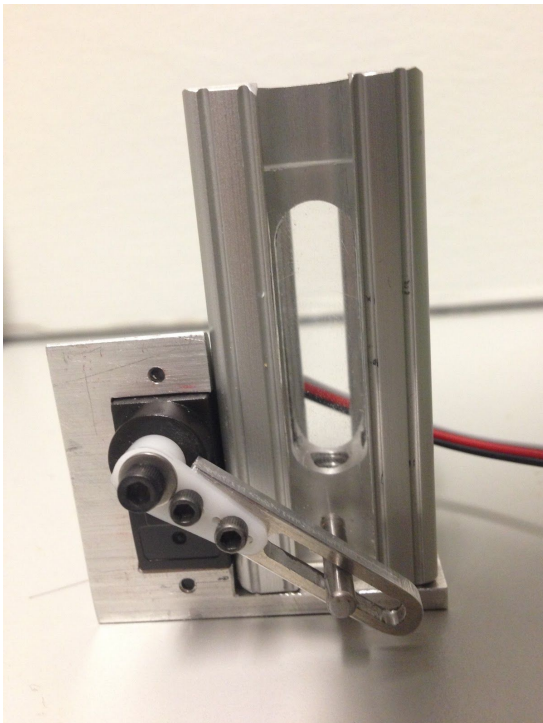
Display Board:



Buttons:



Window:



Code used:

For the display board:

```
#define SEGMENT_A 0 // pin 4
```

```
#define SEGMENT_B 1 // pin 26
```

```
#define SEGMENT_C 2 // pin 22
```

```
#define SEGMENT_D 3 // pin 8
```

```
#define SEGMENT_E 4 // pin 7
```

```
#define SEGMENT_F 5 // pin 27
```

```
#define SEGMENT_G 6 // pin 25
```

```
#define DIGIT_1 10 // pin 6
```

```
#define DIGIT_2 11 // pin 5
```

```
#define DIGIT_3 12 // pin 23
```

```
#define DIGIT_4 13 // pin 24
```

```
#define p0 8 //pin 7
```

```
#define p1 9 //pin 8
```

```
#define p2 7 //pin 9
```

```
int digitPin[4] = {DIGIT_1, DIGIT_2, DIGIT_3, DIGIT_4};
```

```
int a=0;
```

```
int count4=0;
```

```
int count1=0;
```

```
int count2=0;
```

```
int count3=0;
```

```
int dit0,dit1,dit2,dit3;
```

```
int dit[4]= {dit0,dit1,dit2,dit3};
```

```
boolean f=false;
```

```
float prefer=0;
```

```
float degreesC1=20,difference;
```

```
void setup() {
```

```
  //Serial.begin(9600);
```

```
  pinMode(SEGMENT_A, OUTPUT);
```

```
  pinMode(SEGMENT_B, OUTPUT);
```

```
  pinMode(SEGMENT_C, OUTPUT);
```

```
  pinMode(SEGMENT_D, OUTPUT);
```



```
pinMode(SEGMENT_E, OUTPUT);  
pinMode(SEGMENT_F, OUTPUT);  
pinMode(SEGMENT_G, OUTPUT);
```

```
pinMode(DIGIT_1, OUTPUT);  
pinMode(DIGIT_2, OUTPUT);  
pinMode(DIGIT_3, OUTPUT);  
pinMode(DIGIT_4, OUTPUT);
```

```
pinMode(p0,OUTPUT);  
pinMode(p1,OUTPUT);  
pinMode(p2,OUTPUT);
```

```
prefer=20;
```

```
}
```

```
void loop() {
```

```
if(f){  
    dit[2]= (ctof(prefer)/10);  
    dit[3]= ((int)(ctof(prefer))%10);
```

```
}else{  
    dit[2]= (prefer/10);  
    dit[3]= ((int)(prefer)%10);  
}
```

```
for(int i=0; i<4;i++){  
    digitalWrite(digitPin[i], HIGH);  
    printOneDigit(dit[i]);  
    delay (2);  
    digitalWrite(digitPin[i], LOW);  
}
```

```
//Serial.println(analogRead(1));  
//Serial.println(b(1));  
//Serial.println(prefer);  
//delay(500);
```

```
if(count1>0) count1--;
```

```
if(!b(1)&&(count1==0)){  
    f=!f;  
    count1=100;
```

```
}
```

```
if(count2>0) count2--;
```

```
if(!b(2)&&(count2==0)){
```

```
    prefer++;
```

```
    count2=100;
```

```
}
```

```
if(count3>0) count3--;
```

```
if(!b(3)&&(count3==0)){
```

```
    prefer--;
```

```
    count3=100;
```

```
}
```

```
if(f){digitalWrite(p0, HIGH);}else{digitalWrite(p0, LOW);}
```

```
count4++;
```

```
if(count4>500){
```

```
    count4=0;
```

```
    float voltage1;
```

```
voltage1 = getVoltage(0);  
degreesC1 = (voltage1 - 0.5) * 100.0;  
}
```

```
difference= abs(degreesC1-prefer);
```

```
if(f){  
    dit[0]= (ctof(degreesC1)/10);  
    dit[1]= ((int)(ctof(degreesC1))%10);  
}else{  
    dit[0]= (degreesC1/10);  
    dit[1]= ((int)(degreesC1)%10);  
}
```

```
if ((degreesC1-1)<prefer) {  
    digitalWrite(p1, LOW);  
    digitalWrite(p2, LOW);
```

```
}else {  
    if(difference<=3 && difference>1){  
        digitalWrite(p1, HIGH);  
        digitalWrite(p2, LOW);
```

```
    }  
    else if(difference<=6 && difference>3){  
        digitalWrite(p1, LOW);  
        digitalWrite(p2, HIGH);  
    }  
    else if(difference>6){  
        digitalWrite(p1, HIGH);  
        digitalWrite(p2, HIGH);  
    }  
}  
}  
  
void printOneDigit(int a) {  
  
digitalWrite(SEGMENT_A, HIGH); digitalWrite(SEGMENT_B, HIGH);  
digitalWrite(SEGMENT_C, HIGH); digitalWrite(SEGMENT_D, HIGH);  
digitalWrite(SEGMENT_E, HIGH); digitalWrite(SEGMENT_F, HIGH);  
digitalWrite(SEGMENT_G, HIGH);  
  
switch(a) {  
case 0:
```

```
digitalWrite(SEGMENT_A, LOW); digitalWrite(SEGMENT_B, LOW);  
digitalWrite(SEGMENT_C, LOW); digitalWrite(SEGMENT_D, LOW);  
digitalWrite(SEGMENT_E, LOW); digitalWrite(SEGMENT_F, LOW); break;
```

case 1:

```
digitalWrite(SEGMENT_B, LOW); digitalWrite(SEGMENT_C, LOW); break;
```

case 2:

```
digitalWrite(SEGMENT_A, LOW); digitalWrite(SEGMENT_B, LOW);  
digitalWrite(SEGMENT_D, LOW); digitalWrite(SEGMENT_E, LOW);  
digitalWrite(SEGMENT_G, LOW); break;
```

case 3:

```
digitalWrite(SEGMENT_A, LOW); digitalWrite(SEGMENT_B, LOW);  
digitalWrite(SEGMENT_C, LOW); digitalWrite(SEGMENT_D, LOW);  
digitalWrite(SEGMENT_G, LOW); break;
```

case 4:

```
digitalWrite(SEGMENT_B, LOW); digitalWrite(SEGMENT_C, LOW);  
digitalWrite(SEGMENT_F, LOW); digitalWrite(SEGMENT_G, LOW); break;
```

case 5:

```
digitalWrite(SEGMENT_A, LOW); digitalWrite(SEGMENT_C, LOW);  
digitalWrite(SEGMENT_D, LOW); digitalWrite(SEGMENT_F, LOW);  
digitalWrite(SEGMENT_G, LOW); break;
```

case 6:

```
digitalWrite(SEGMENT_A, LOW); digitalWrite(SEGMENT_C, LOW);  
digitalWrite(SEGMENT_D, LOW); digitalWrite(SEGMENT_E, LOW);  
digitalWrite(SEGMENT_F, LOW); digitalWrite(SEGMENT_G, LOW); break;
```

case 7:

```
digitalWrite(SEGMENT_A, LOW); digitalWrite(SEGMENT_B, LOW);  
digitalWrite(SEGMENT_C, LOW); break;
```

case 8:

```
digitalWrite(SEGMENT_A, LOW); digitalWrite(SEGMENT_B, LOW);  
digitalWrite(SEGMENT_C, LOW); digitalWrite(SEGMENT_D, LOW);  
digitalWrite(SEGMENT_E, LOW); digitalWrite(SEGMENT_F, LOW);  
digitalWrite(SEGMENT_G, LOW); break;
```

case 9:

```
digitalWrite(SEGMENT_A, LOW); digitalWrite(SEGMENT_B, LOW);  
digitalWrite(SEGMENT_C, LOW); digitalWrite(SEGMENT_D, LOW);  
digitalWrite(SEGMENT_F, LOW); digitalWrite(SEGMENT_G, LOW); break;
```

```
}
```

```
}
```

```
float getVoltage(int pin)
```

```
{
```

```
    return (analogRead(pin) * 0.004882814);
```

```
}
```

```
boolean b(int pin)
```

```
{
```

```
  if(analogRead(pin) >511){
```

```
    return true;
```

```
  }
```

```
  return false;
```

```
}
```

```
float ctof(float c)
```

```
{
```

```
  return( c* (9.0 / 5.0) + 32.0);
```

```
}
```

For the servo motor:

```
#define p0 8//pin 7
```

```
#define p1 9//pin 8
```

```
#define p2 7//pin 9
```

```
#include <Servo.h>
```



```
Servo servo1;
```

```
int count=0;
```

```
void setup() {
```

```
  servo1.attach(10); //servo motor
```

```
  pinMode(5,OUTPUT); //pin 2
```

```
  pinMode(6,OUTPUT); //pin 3
```

```
  // put your setup code here, to run once:
```

```
}
```

```
void loop() {
```

```
  boolean d0= (digitalRead (p0)==HIGH);
```

```
  boolean d1= (digitalRead (p1)==HIGH);
```

```
  boolean d2= (digitalRead (p2)==HIGH);
```

```
  if(d0){
```

```
    digitalWrite(5, HIGH);
```

```
    digitalWrite(6, LOW);
```

```
  }else{
```

```
    digitalWrite(5, LOW);
```

```
    digitalWrite(6, HIGH);  
}  
  
delay(1);  
  
count++;  
  
if(count>500){  
    count=0;  
  
    if(!d1&&!d2) servo1.write(170);  
    if(d1&&!d2) servo1.write(145);  
    if(!d1&&d2) servo1.write(120);  
    if(d1&&d2) servo1.write(0);  
}  
}
```

Results:

The device is able to read the temperature value inside the room and then, display that value on the display board. At the same time, on the display, there will be the preferred value that users want to set change the temperature inside the room to. That value can be changed from using the three buttons. The first one will change the unit of the degrees on the display. The second and the third one will change the degree of those preferred temperature by increasing and decreasing by one degree.

The servo motor would turn at increasing angles when the difference between the preferred and measured temperatures would increase, for e.g. the motor would turn about 60° when the difference was 1, about 120° when the difference was 3, up to a maximum of about 170° . The motor was supposed to go a maximum of 180° . However, due to internal friction in the servo, the maximum angle the motor could turn to about 170° only.

All the expected results are presented. And the display board is fully functional. Three buttons are able to accomplish their designed functions. At the same time, the window will open according to the instructions of the code.

Future steps:

This project only acted as a proof of concept for the main idea. The main idea was to implement on the actual windows in the dorms. Since the class was focussed on the electronics aspect of the project and there was limited time, the mechanical aspect of the project was intentionally left for future work. The mechanical aspect of the project is described here:

Since the windows were quite big, we would have had to use a pulley-gear system along with a much more powerful servo motor. The pulley-gear system in the construction would allow a smaller, less expensive motor to be used as pulleys increase the mechanical effort, thereby reducing manual effort, required to raise objects. The proposed mechanical design for the project can be taken ahead as a sustainable project

because there will be a very minimal release of CFCs when running the motor, unlike air conditioners which release a great amount of CFCs and other harmful gases.

Conclusion:

Our final project was giving us many a variety of problems when it came demo day. I am glad that we left 45 minutes to troubleshoot any problems we might have on demo day because we used all of them. We had not tested out the window that the machine shop was kind enough to make for us, our newest addition to our project. We knew that we may have to change the angles around a bit depending upon the configuration that they used to mount the motor to the frame and servo motor arm to raise and lower the window. We were getting various pin problems that we were having trouble explaining as well, but we managed to fix these in time for the demonstration. Our demonstration went exactly as we had intended it to with no issues.