Honors Project Final Report - EEE 202

This project will display two of the many applications that are possible using an Arduino, similar to displays or games on a phone. The first demonstration is a game that is user controlled, that turns on and off different led lights. There will be eight LED lights and on the computer the user can input a number which will lead to the LED that is assigned to that number to light up or if inputted and can also clear all the lights to turn off at once as well (*Enseeiht*). The other display will be of the temperature or weather in the room and display it on a LCD screen (*Enseeiht*). By learning how to develop these applications on an Arduino, I have gotten an introduction into the possibilities of Arduino by gaining hands-on circuits experience. I also gained a better understanding of circuit concepts and better understand the concepts learned in class by working and seeing the demonstrations from hands-on work.

The first demonstration was about controlling eight LED lights using a chip called the 74HC595 Serial to Parallel Converter. In order to light an LED, without using the extra pins, this component allows it so that we can connect all LED and control the lights. The component is a shift register which is a type of chip that can hold the memory of 8 different outputs as 0 or 1, or in the case of this demonstration, on or off. One of the pins of this chip is called OE or output enable which can disable all outputs at once, which is what I used to turn off all the lights at once. As all the resistors that are connected to the leds are connected in parallel to the chip, when one turns off, all of them do, due to the OE pin. I was also able to apply knowledge from class when creating this demonstration. The resource I used to create this demonstration instructed that I should a 220 ohm resistor but on TinkerCad, when I simulated this, I got an error saying that the 74HC595 chip will break because of the total current through power pins is 56.6 mA, while the maximum that can go through it is only 50.0 mA. I then decided to try a 1K ohm resistor instead because from Ohm's Law V=IR, I = V/R, so if I increase R, current decreases. Thus, when I simulated the TinkerCad with 1k Ohm resistor instead it worked. Another difficulty I encountered with this demonstration was when building the physical build of the circuit I could not get the LED's to light up. I checked my schematic with my build multiple times but did not understand what was wrong, but then realized one part of the resource I was referencing did not specify how to connect the LED lights onto the breadboard correctly. Every LED has two pins, one is Anode (+) and the other is a Cathode (-) ("Led Polarity." *Switch Electronics*). I had my Anode connected to ground and not my Cathode, and once I switched their positions, the LED's turned on.

The second demonstration was about recording the temperature surrounding a circuit on a screen. The main components used in this circuit were a thermistor, LCD, and a potentiometer. The LCD and potentiometer are components I was familiar with as we use them in EEE 202's lab class. The thermistor was an interesting new component that I was unfamiliar with. A thermistor is a thermal resistor that changes its resistance with temperature. I learned that technically all resistors are thermistors as they all change with their surrounding temperature, but thermistors are made specifically for being sensitive to temperature for recordings. When creating physical builds for circuits it is important to first try them on simulators such as TinkerCad in order to be safe with the circuit and make sure it runs well. For this specific demonstration though I struggled on getting the right circuit as there was only one component that measured temperature on TinkerCad, a temperature sensor TMP 36 which I did not have access to, and the components I had access to were not on TinkerCad, so I tried many circuit schematics till finally getting the thermistor to work. At first, one of the components I tried was a NPN Transistor, I thought that this component would measure the temperature as well, so I replaced from the Temperature sensor from the TinkerCad schematic but got large values, and found out that PNP transistors record temperature and not NPN transistors. Also, another component I could have used from the Arduino Kit was the DHT11 Temperature and Humidity Module, this works similar to the thermistor in recording temperature but also records surrounding humidity levels. Another issue I faced in creating the physical build for this demonstration was that my LCD screen was turned on, but not displaying anything. As I was in the lab, a TA explained to me how my potentiometer was not turned due to the current angle which was why my LCD had no display. After they fixed the positioning of my potentiometer, the LCD displayed the correct output. The potentiometer is used for an LCD to display text in the desired brightness/screen contrast (*Dallas Semiconductor Digital Potentiometers - UPM*). I was also able to apply this learning later when I was working on the final design project with my lab group and our LCD was not displaying correctly, I was able to fix the potentiometer to get it to display. For the demonstration of displaying temperature, indoors, the LCD displayed an accurate display of 74.922℉ and when I put my fingers around the thermistor it displayed around my body temperature of 96℉.

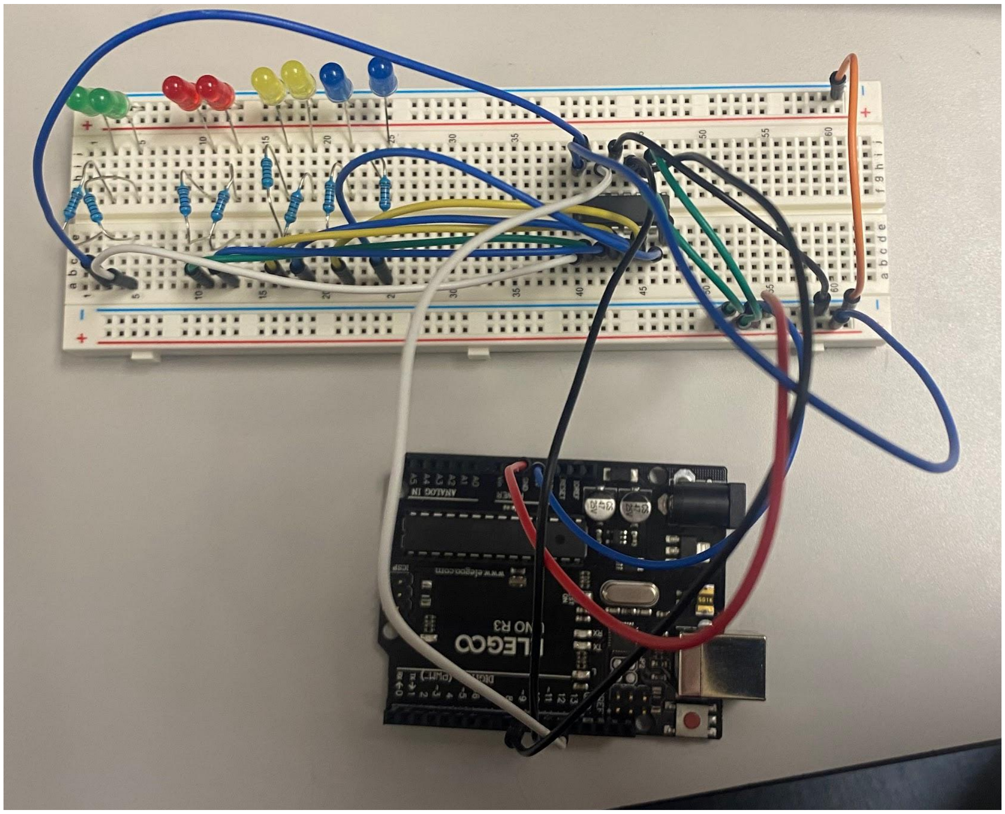
Overall, I really enjoyed doing this Honor's Project. I learned a lot about new components and applied concepts from the EEE 202 course when building the designs and schematics on TinkerCad and also through the physical builds of both the demonstrations. It was exciting to build and see the physical build works from the TinkerCad simulations. Personally, I will like to continue to build my skills hands-on and will be purchasing more such circuits and creating projects like these. As I go on to more engineering courses, especially Circuits II next year, I will continue to use these skills.

Zoom Recording Link To TinkerCad Demonstrations:

https://asu.zoom.us/rec/share/k-IHBhzX9PJfDp9pgvz3aqVvZFk5AXaH9C18V7t1FIBfZbsF8p-3SXf8yFxijjk7.ztp4wGTd4AYcE8uZ?startTime=1682720755000

LED Demonstration:

- Physical Build: (Also attached video in email of video demonstration of this game with LEDs turning on/of)



- Code:

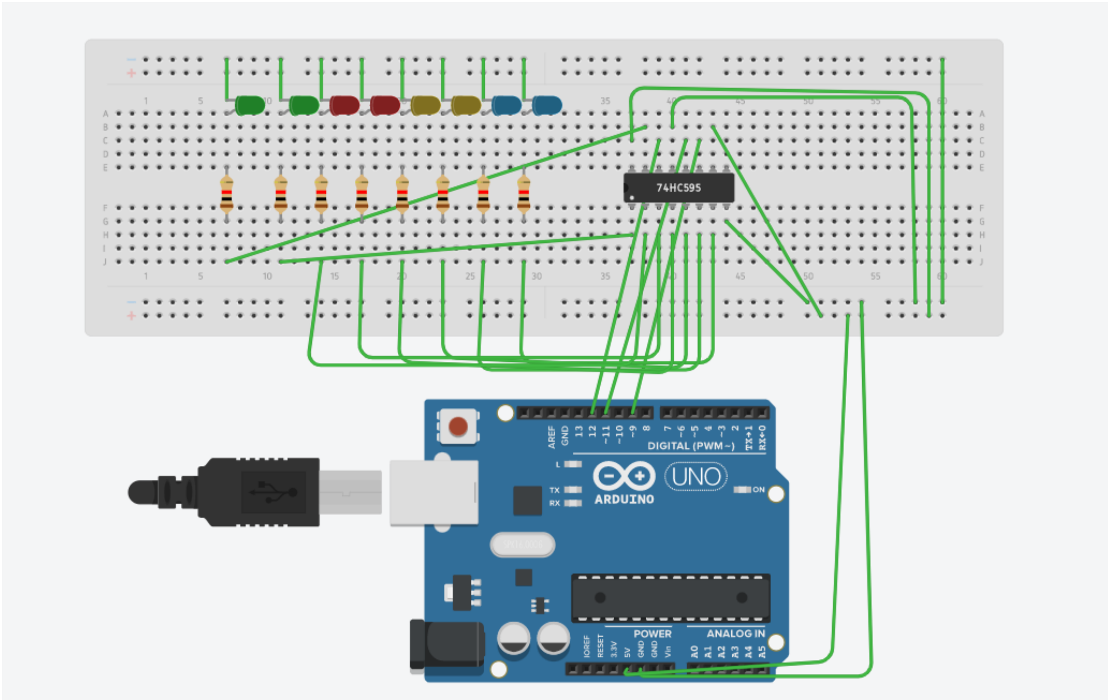
```
int latchPin = 11;
int clockPin = 9;
int dataPin = 12;
byte leds = 0;

void setup(){
pinMode(latchPin, OUTPUT);
pinMode(dataPin, OUTPUT);
pinMode(clockPin, OUTPUT);
updateShiftRegister();
Serial.begin(9600);
while (! Serial); // Wait until Serial is ready - Leonardo
Serial.println("Enter LED Number 0 to 7 or 'x' to clear");
}
```

```
void loop(){
if (Serial.available())
{
char ch = Serial.read();
if (ch >= '0' && ch <= '7')
{
int led = ch - '0';
bitSet(leds, led);
updateShiftRegister();
Serial.print("Turned on LED ");
Serial.println(led);
}
if (ch == 'x')
{
leds = 0;
updateShiftRegister();
Serial.println("Cleared");
}
}
}

void updateShiftRegister(){
digitalWrite(latchPin, LOW);
shiftOut(dataPin, clockPin, LSBFIRST, leds);
digitalWrite(latchPin, HIGH);
}
```

- TinkerCad:



Component List

Name	Quantity	Component
U1	1	Arduino Uno R3
D4 D5	2	Yellow LED
D6 D7	2	Blue LED
U2	1	8-Bit Shift Register
D2 D3	2	Green LED
D1 D9	2	Red LED
R7 R9 R6 R1 R2 R3 R4 R8	8	1 kΩ Resistor

Temperature Demonstration:

- Code:

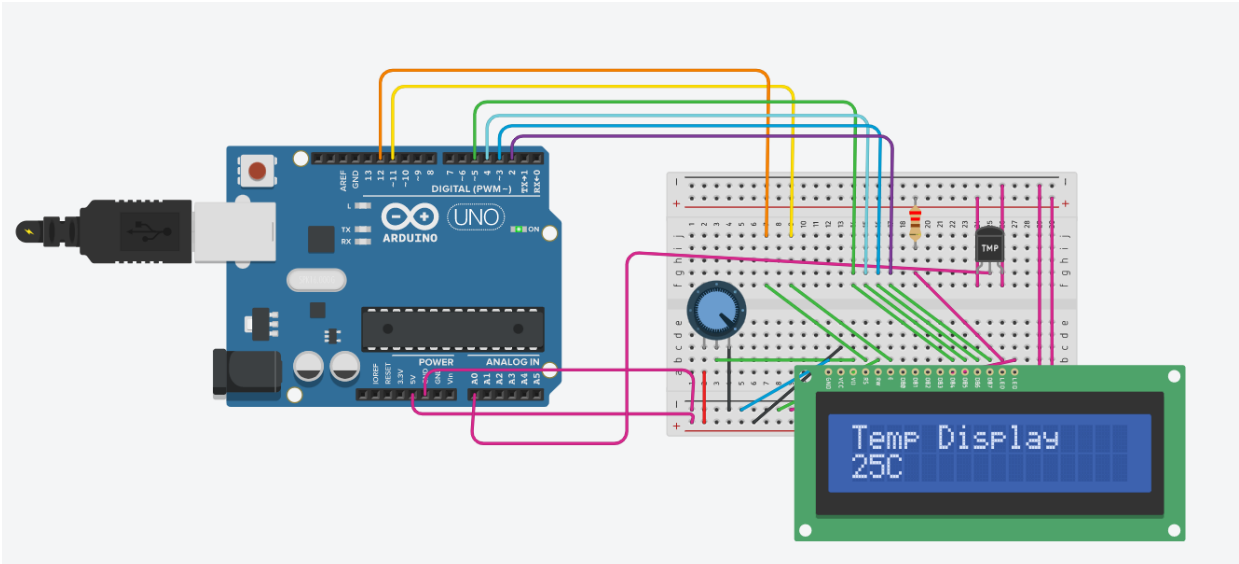
```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

float tempC;
float tempF;
double tempK;
int tempReading;
void setup() {
  // put your setup code here, to run once:
lcd.begin(16, 2);
}

void loop() {
 tempReading = analogRead(A0);
tempK = log(10000.0 * ((1024.0 / tempReading - 1)));
tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * tempK * tempK ))* tempK );
 tempC = tempK - 273.15;
 tempF = (tempC * 9.0)/ 5.0 + 32.0;

lcd.setCursor(0, 0);
lcd.print("Temp F: ");
lcd.setCursor(0, 1);
lcd.print(tempF);
}
```

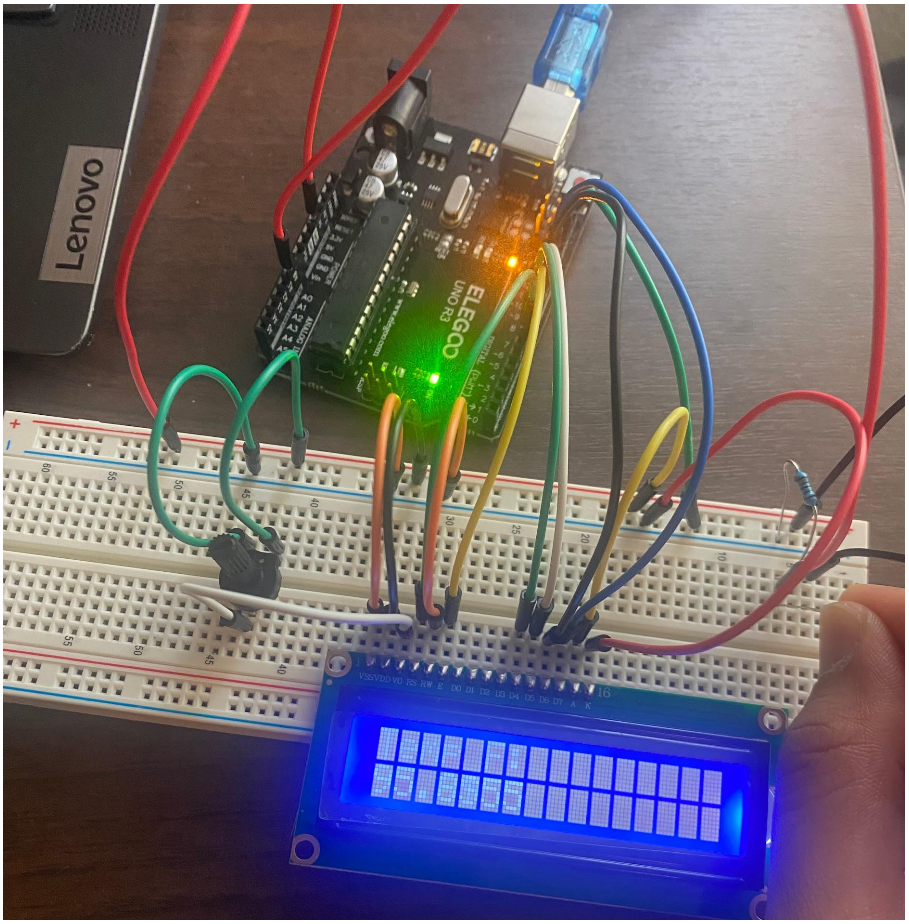
- TinkerCad:



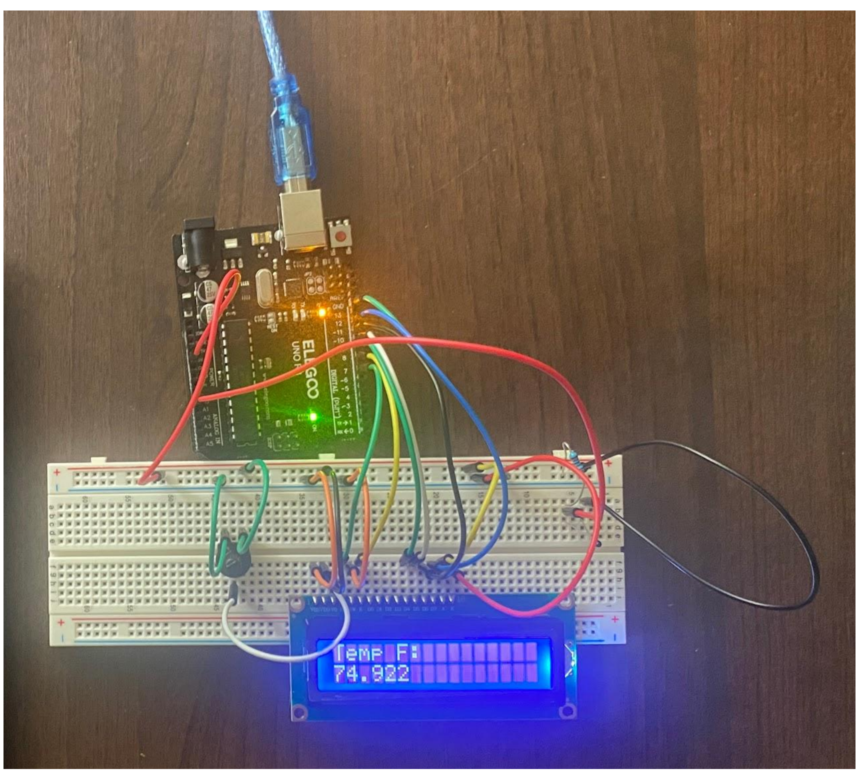
Component List

Name	Quantity	Component
U2	1	LCD 16 x 2
U4	1	Temperature Sensor [TMP36]
R1	1	220 Ω Resistor
Rpot3	1	10 kΩ Potentiometer
U1	1	Arduino Uno R3

- Picture Of Physical Build:



Picture Above Of Body Temperature Display around 95.9℉



Picture Above Of Room Temperature Display around 74.922℉

Works Cited

"Home." Arduino, https://www.arduino.cc/.

Dallas Semiconductor Digital Potentiometers - UPM. https://www.datsi.fi.upm.es/docencia/Informatica_Industrial/DMC/dallas/app69.pdf.

Enseeiht. http://diabeto.enseeiht.fr/download/divers/The-Most-Complete-Starter-Kit-for-UNO-V1.0.17.3.6.pdf.

"From Mind to Design in Minutes." *Tinkercad*, https://www.tinkercad.com/.

"Led Polarity." *Switch Electronics*, https://www.switchelectronics.co.uk/blog/post/ledpolarity.html.