

# Thermoelectric Temp Control

---

By Dominick Valenti and Kellen Hurley

# Purpose of Project

- The goal of this project is to design a temperature control system that uses a Peltier device to heat, cool, or match the ambient temperature of a simulated “plant” environment.
- The system uses analog and I2C sensors to monitor temperature, with user input handled through a keypad and output displayed on an LCD and LED light bar.
- This project is designed to teach sensor interfacing, I2C communication, relay control with transistors, basic feedback control loops, and integrating hardware and software into a functioning system.

# Sensor Polling Pseudocode

```
Initialize_ADC()  
Initialize_I2C()
```

Loop every 0.5 seconds:

```
    raw_adc = Read_ADC(LM19_channel)  
    voltage = Convert_To_Voltage(raw_adc)  
    ambient_temp = Convert_Voltage_To_Temperature(voltage)  
    Update_Moving_Average(ambient_temp, ambient_buffer)
```

```
    Send_I2C_Request(LM92_address)  
    raw_plant = Read_I2C_Data()  
    plant_temp = Convert_Raw_To_Temp_LM92(raw_plant)  
    Update_Moving_Average(plant_temp, plant_buffer)
```

Loop every 1 second:

```
    current_time = Read_I2C_Time(RTC_address)  
    elapsed_time = current_time - mode_start_time
```

# Control Logic Pseudocode

```
On_Mode_Change(new_mode):  
    mode = new_mode  
    mode_start_time = Read_I2C_Time(RTC_address)  
    Reset_Moving_Average_Buffers()  
  
Loop every control cycle:  
    If elapsed_time >= 5 minutes:  
        Disable_Peltier()  
    Else if mode == 'heat':  
        Set_Heat_Mode()  
    Else if mode == 'cool':  
        Set_Cool_Mode()  
    Else if mode == 'match':  
        If plant_avg < ambient_avg - threshold:  
            Set_Heat_Mode()  
        Else if plant_avg > ambient_avg + threshold:  
            Set_Cool_Mode()  
        Else:  
            Disable_Peltier()
```

# User Interface (LCD & Keypad) Pseudocode

```
On_Keypad_Input(key):  
    If key in ['A', 'B', 'C', 'D']:  
        On_Mode_Change(key)  
    If key is digit:  
        Append_To_WindowSize_Input(key)  
    If key == 'Enter':  
        window_size = Parse_Input()  
        Update_WindowSize(window_size)  
  
Loop every 2 seconds:  
    Display_Line1 = Format_Mode(mode) + Format_Temp(ambient_avg)  
    Display_Line2 = Format_Window(window_size) + Format_Time(elapsed_time) +  
    Format_Temp(plant_avg)  
    Update_LCD(Display_Line1, Display_Line2)
```

# Output (LED Bar & Peltier Device) Pseudocode

```
Initialize_Peltier_GPIO()
```

```
Initialize_LED_Bar()
```

```
Loop every 1 second:
```

```
    If heating_active:
```

```
        Light_Bar_Right()
```

```
        Activate_Heat_Relay()
```

```
        Deactivate_Cool_Relay()
```

```
    Else if cooling_active:
```

```
        Light_Bar_Left()
```

```
        Activate_Cool_Relay()
```

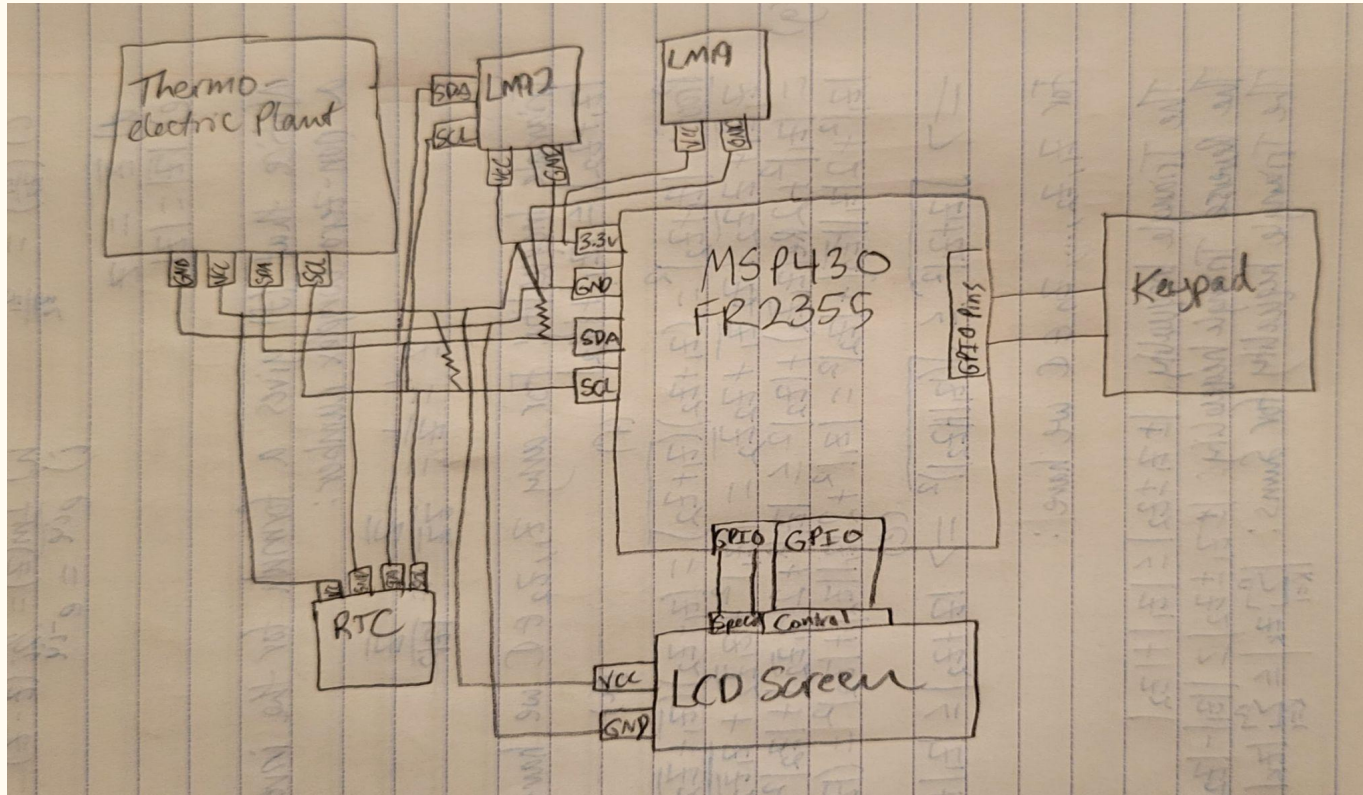
```
        Deactivate_Heat_Relay()
```

```
    Else:
```

```
        Turn_Off_LED_Bar()
```

```
        Deactivate_Both_Relays()
```

# Circuit Diagram



# Architecture Diagram

