FINAL PROJECT PRESENTATION:

GEOFFREY SOWASH
JOHNNY NGUYEN

FOUR CHANNEL TEMPERATURE CONTROLLER

Full Instrument

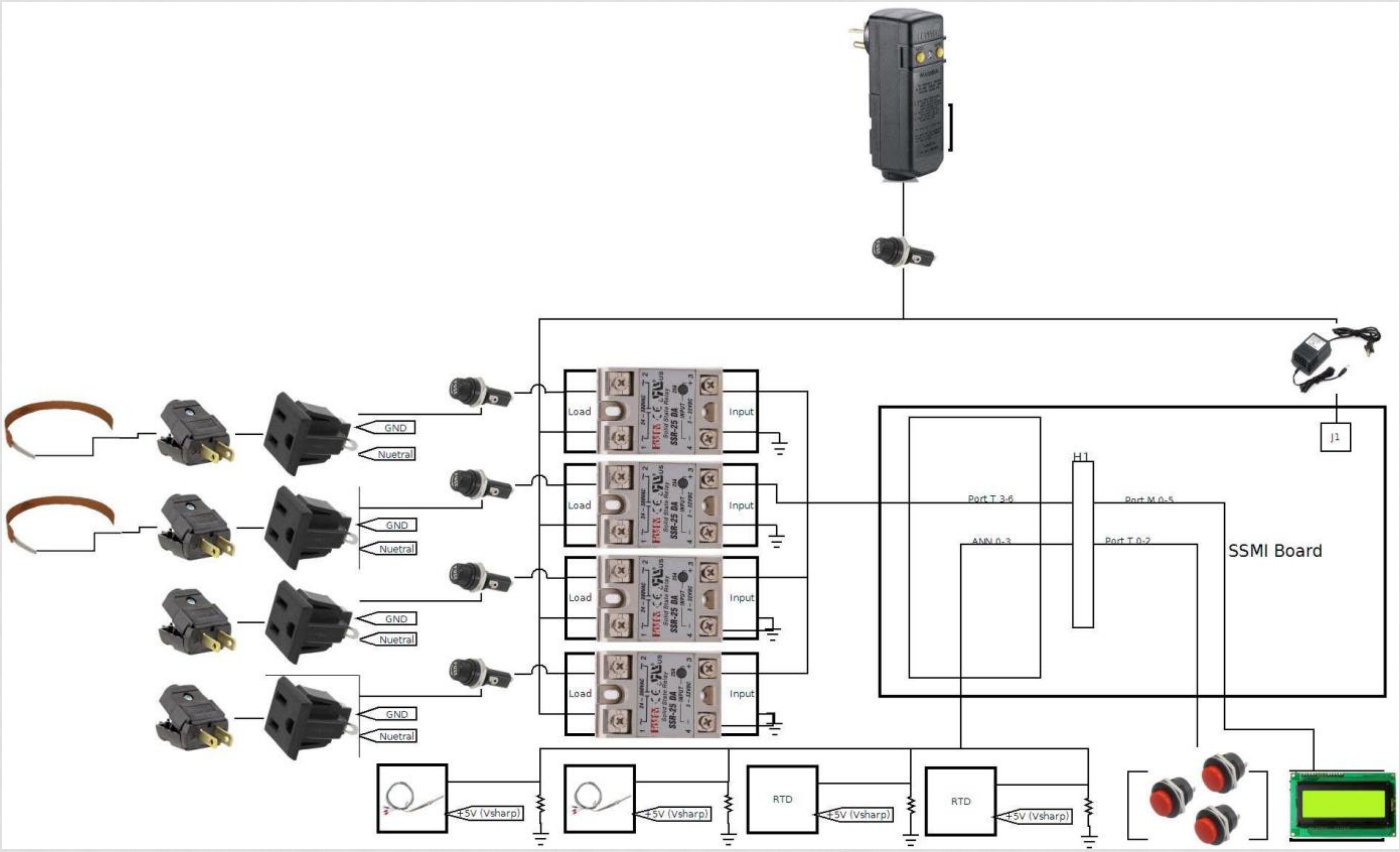
Basic Layout

DESIGN SPECIFICATIONS

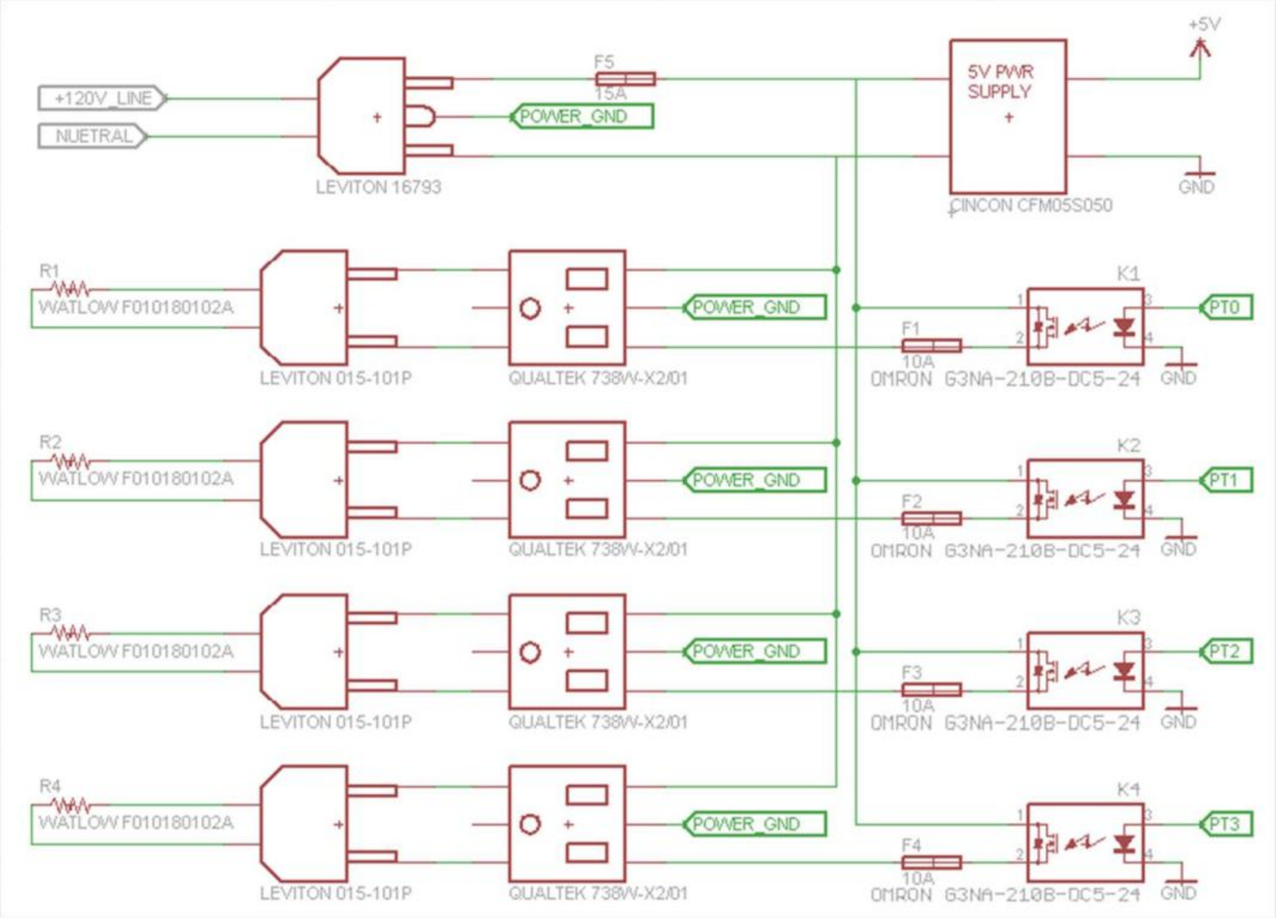
Temperature Controller

- 4 Channels
- 8 amps max per channel
- Total 15 amps at ~110V
  - Designed to be upgradable to 220V
- Temperature measurement based on Resistive Thermal Devices (PT100)
- Temperature output controlled by Solid State Relays for long operational life

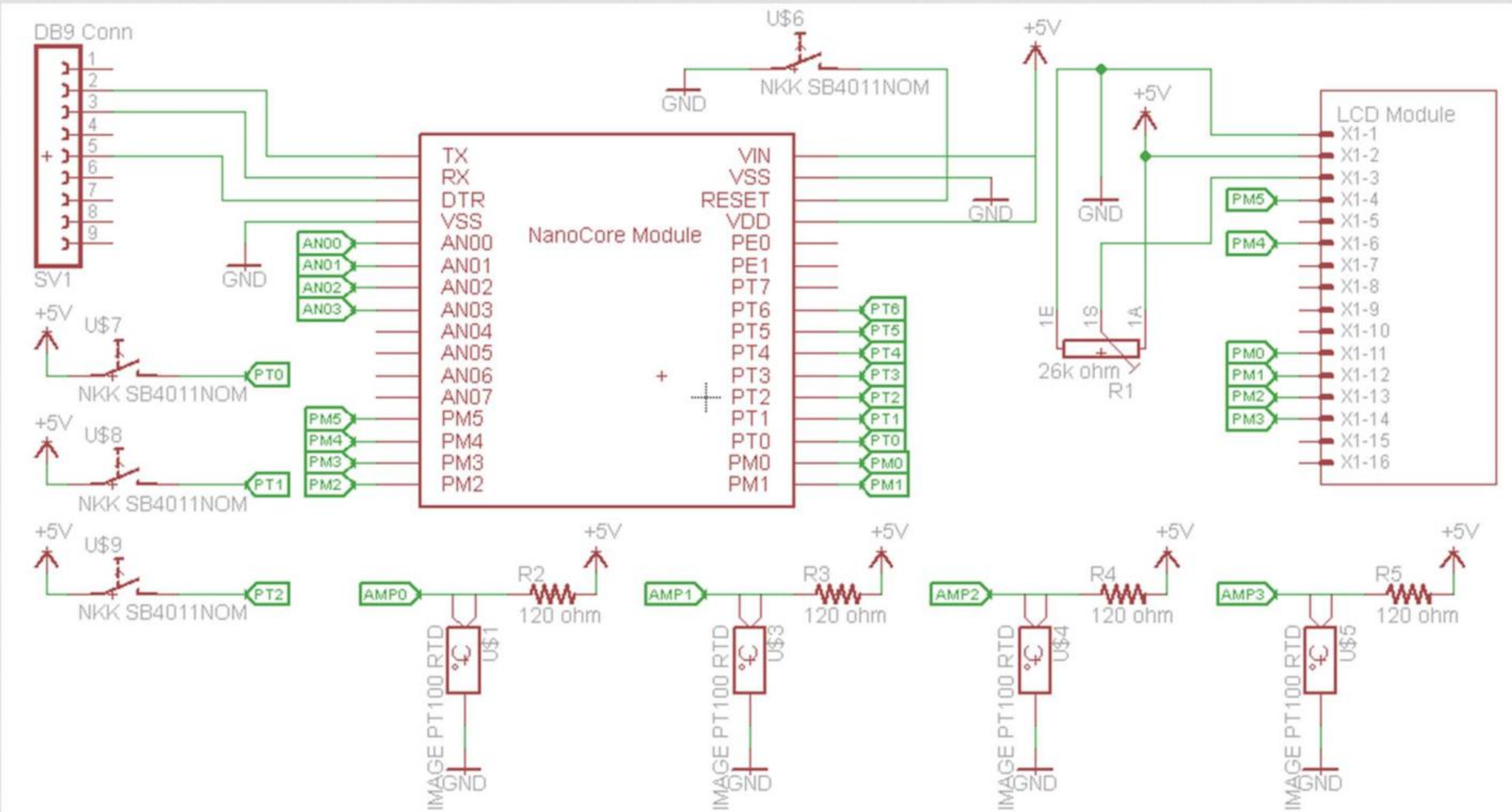
BLOCK DIAGRAM



WIRING DIAGRAM: RELAYS

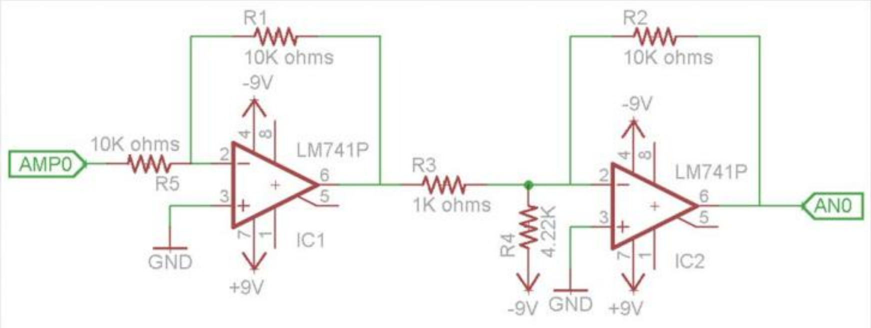


WIRING DIAGRAM: CONTROL



SENSOR AMPLIFIER

- Temperature Range:
  - -10° C - 120° C
  - ~14° F - 248° F
- Total Range Difference:
  - 130° C or 234° F
- 8 bit resolution:
  - 256 data values
  - ~0.914 values/degree F
- 2.2V-2.75V -> ~0V-5V



BASICS OF TEMPERATURE CONTROL

- PID Control
  - P (Proportional)
    - Based on the current temperature difference
      - P component = current difference
  - I (Integral)
    - Summation of temperature differences over time
      - I component = I component + (current difference * sample period)
  - D (Derivative)
    - Based on rate of change of the temperature
      - D component = (current error – previous error) * sample period
  - Duty Cycle of Output = K_p * P + K_i * I + K_d * D

USER INPUT/OUTPUT

- All input based on three buttons:
  - Change channel
  - Temperature set point up
  - Temperature set point down
- All output displayed on LCD screen (4x20):
  - One line on screen for each channel
  - Each line contains current channel temperature and current channel set point temperature

PSUEDOCODE: INPUT/OUTPUT

```
Clear RTI flag
// Button 1: Change channel
If (button1&& button_pressed = =not_pressed)
        Current_channel + 1
        button_pressed = pressed
        If (Current_channel > 4)
                Current_channel = 0
End
// Ensures each button push only results in one channel change
If (!button 1 && button_pressed = =pressed)
        button_pressed = not pressed
End

// Button 2: Increase temperature set-point
If (button 2)
        Switch (current_channel)
                Case 1: set_temp_1 + 1
                Case 2: set_temp_2 + 1
                Case 3: set_temp_3 + 1
                Case 4: set_temp_4 + 1
End

// Button 3: Decrease temperature set-point
If (button 3)
        Switch (current_channel)
                Case 1: set_temp_1 - 1
                Case 2: set_temp_2 - 1
                Case 3: set_temp_3 - 1
                Case 4: set_temp_4 - 1
End
```

TEMPERATURE INPUT: PSUEDOCODE

- All temperature values read from a single A/D converter
- Convert raw input (0-256) into voltage value
- Convert voltage value into current resistance value of the RTD sensor
- Use resistance value to find a temperature value from pre-generated lookup table

PID CONTROL/RELAY OUTPUT

- Inside of main function
- Timing of duty cycle is based on counter
- Set output period of ~5 sec for extended relay life
  - One relay switch per output period
  - Starts high, switches low (timing based on duty cycle)
- Duty cycle determined by PID algorithm
- PID algorithm called at the beginning of the 5 sec period when the counter is reset

PSEUDOCODE: PID CONTROL/RELAY OUTPUT

- **PID Control / Relay Output**

```
While (true)
    cycles_counter = 0
    //For each channel, set duty cycle using P control
    Duty_cycle = Kp * (set_temp – current_temp)
    For each channel, if (duty > 100)
        Set to 100
    Turn on PT3 – PT6 for relay outputs
    For each channel, if (cycles_counter > cycles_counter * duty cycle / 100)
        Turn off the channel
    cycles_counter + 1
End
```

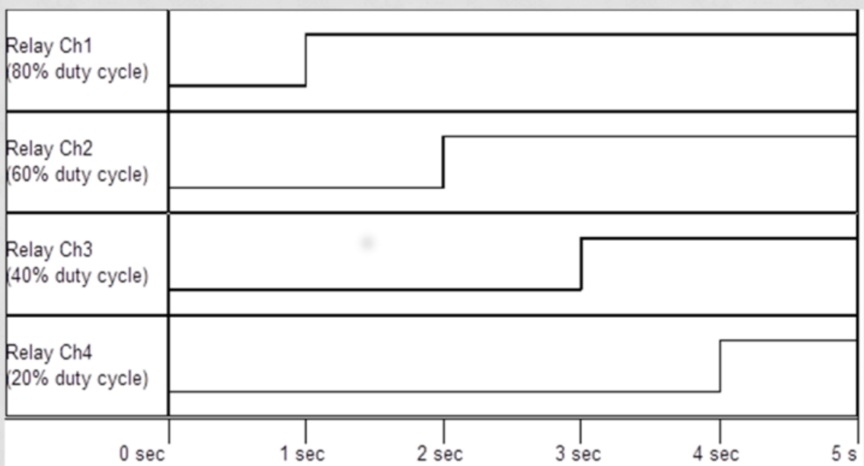
TIMING

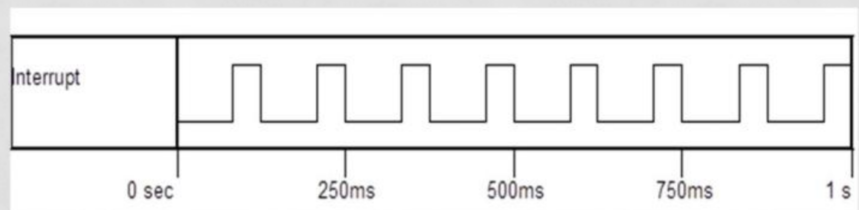
Temperature Output Timing

- In the main program loop
- Set period of ~5 sec
- Controlled by duty cycle

Interrupt Loop

- Slowest timing 1/8 sec
- Controls:
  - User Input





CONCLUSION/PROGRESS

- Temperature controller using a PID control system
- Goal:
  - Circuit:
    - Finalize construction of the components and wiring
    - Modify amplification circuit for best resolution
  - Coding:
    - Implement a PI system (D deemed unnecessary by Rad)
- Current Progress:
  - Circuit:
    - Basic Temperature Acquisition
  - Coding:
    - Currently have implemented a system based on proportion
- Questions?