- Data Acquisition:

The acquisition card serves as the main component of the acquisition system (DAQ) and plays an essential role as an interface in our FDD. **Figure X** displays the DAQ interface. The system relies on a microcontroller for handling many tasks, including receiving sensors data, conducting analog/digital conversion, and transmitting data via the serial port to the designated unit for processing. The microcontroller board, which is based on the ATmega328P, has a clock frequency of 16MHz, which facilitates quicker data transmission in addition to allowing for the simultaneous execution of those tasks. **Figure Y** illustrates the DAQ system flowchart.

- Microcontroller: ATmega2560
- Sampling period: 100 ms
- Data logging: .txt and .csv files on host PC

Thermistors calibration procedure.

To ensure accurate temperature measurements in our inverter-driven PMSM system, we developed a custom calibration setup for the thermistors. The calibration apparatus consisted of an aluminum block repurposed from a Prusa i3 3D printer's extruder heat block [ref prusa project] . This block, containing an embedded resistor as a heating element, provided an ideal environment for precise temperature control and uniform heat distribution due to aluminum's high thermal conductivity.

Figure 3: Heater Block Assembly Set Extruder Hot End for Prusa i3 3D Printer 1.75mm MK8

The calibration process utilized an Arduino-based system implementing PID control to maintain stable temperatures at four setpoints: 30°C, 40°C, 50°C, and 60°C. A high-precision Dallas Temperature sensor (DS18B20) served as the reference, while the thermistors under calibration were placed in close proximity on the aluminum block. At each setpoint, after allowing sufficient time for temperature stabilization, we recorded multiple readings from both the reference sensor and the thermistors.

The Arduino system collected data every 100 milliseconds, logging the setpoint temperature, PID output, reference sensor temperature, and raw analog readings from the thermistors. This data was used to derive calibration curves for each thermistor, enabling accurate conversion of raw analog values to precise temperature measurements. The use of multiple setpoints across our expected operating range allowed us to account for any non-linearities in the thermistor responses, ensuring reliable temperature monitoring throughout our fault detection and diagnosis experiments.

The plot in figure 4 shows the temperature, setpoint, and PWM output over time.

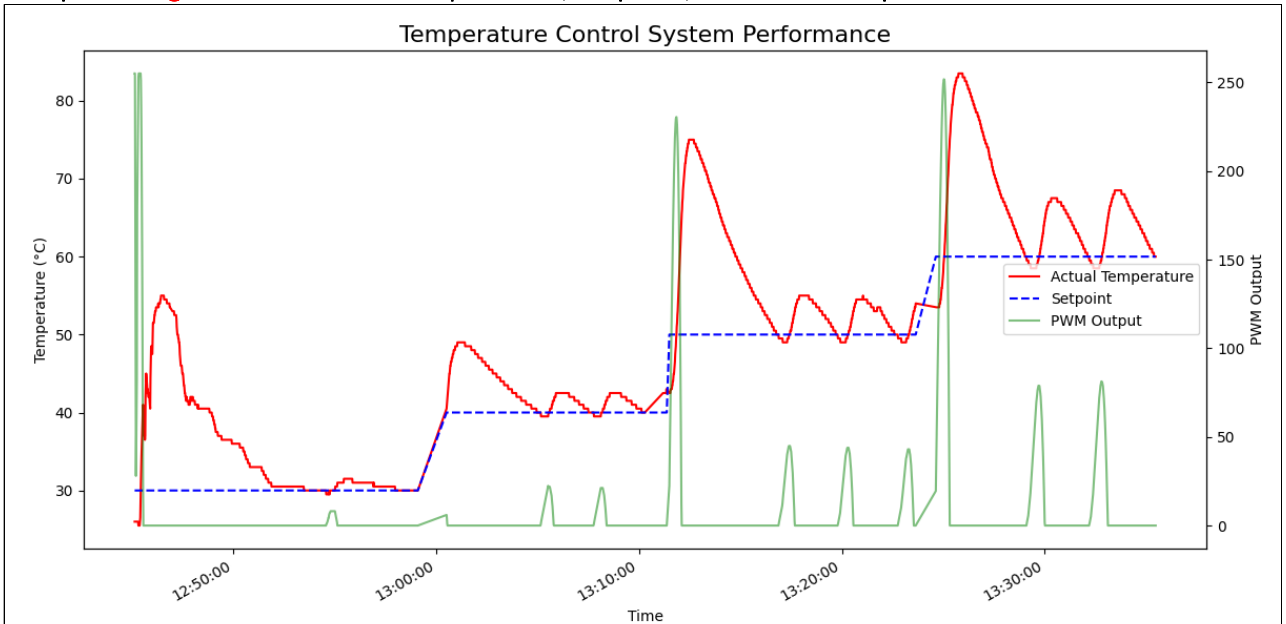

Figure 4: plot of temperature, setpoint, and PWM output over time.

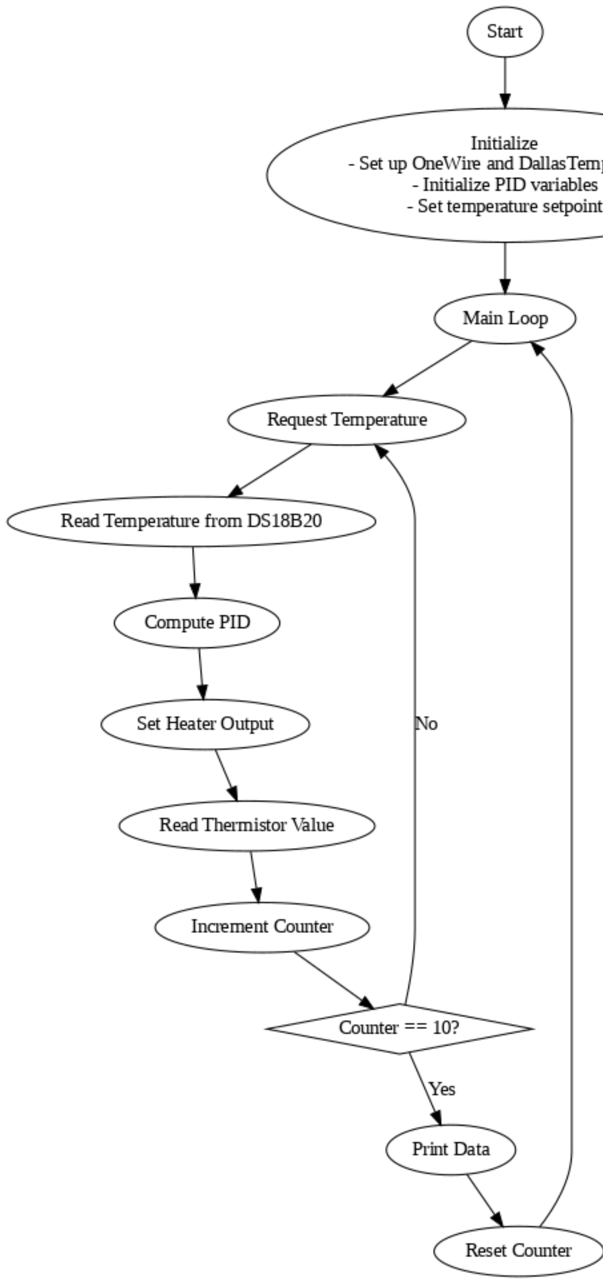

Figure 5: Flowchart of the Thermistor Calibration Process

This flowchart illustrates the step-by-step process used for calibrating the thermistors.
The process includes initialization of the OneWire and DallasTemperature sensors,
PID control for temperature regulation, and periodic data collection.
The main loop continuously monitors and adjusts the temperature while collecting
thermistor readings, ensuring accurate calibration across the desired temperature range.

.

The mathematical equations that describe the relationship between the Arduino analog reading and the thermistor resistance in our setup. Given that the thermistor is connected to Arduino A0 via a voltage divider with a 10kΩ resistor, we can derive the following equations:

1. Voltage Divider Equation:
   Let's define our variables:
   - V_in = Arduino reference voltage (typically 5V)
   - V_out = Voltage read by Arduino analog pin
   - R1 = Fixed resistor (10kΩ in your case)
   - R_th = Thermistor resistance (unknown, to be calculated)
   - ADC = Analog-to-Digital Converter reading (0-1023 for 10-bit ADC)

- The thermistor is located in lower side of the voltage divider.

   The voltage divider equation is:
   V_out = V_in * (R_th / (R_th + R1))

2. Arduino ADC Conversion:
   The Arduino converts the analog voltage to a digital value between 0 and 1023:
   ADC = (V_out / V_in) * 1023

3. Combining these equations:
   ADC = (R_th / (R_th + R1)) * 1023

4. Solving for R_th:
   R_th = R1 / ((1023 / ADC) - 1)

$$V_{out} = V_{in} \cdot \frac{R_{th}}{R_{th} + R_1} \tag{1}$$

$$ADC = \frac{V_{out}}{V_{in}} \cdot 1023 \tag{2}$$

$$ADC = \frac{R_{th}}{R_{th} + R_1} \cdot 1023 \tag{3}$$

$$R_{th} = R_1 \cdot \frac{1023}{ADC} - R_1 = \frac{R_1}{(\frac{1023}{ADC}) - 1} \tag{4}$$

Equations 1-4 describe the relationship between the Arduino's analog reading and the thermistor resistance in our calibration setup. Equation 1 represents the voltage divider formed by the thermistor and the fixed resistor. Equation 2 shows how the Arduino converts the analog voltage to a digital value. Equation 3 combines these relationships, and Equation 4 provides the final formula for calculating the thermistor resistance from the Arduino's ADC reading.

Using these equations, we can accurately determine the thermistor resistance for each ADC value obtained during the calibration process. This forms the basis for creating our temperature-resistance calibration curve and deriving the Steinhart-Hart coefficients for precise temperature measurements.

1. Extracting mean ADC values for each setpoint:

Let's find the mean ADC value when the measured temperature is equal to the setpoint for each of the four setpoints.

Setpoint	Mean ADC
30°C	461.84
40°C	383.97
50°C	312.42
60°C	249.55

2. Calculating thermistor resistance:

Using the equation R_th = R1 / ((1023 / ADC) - 1), where R1 = 10,000 ohms, we can calculate the thermistor resistance for each setpoint:

For 30°C: R_th = 8,248.69 ohms
For 40°C: R_th = 5,911.96 ohms
For 50°C: R_th = 4,289.16 ohms
For 60°C: R_th = 3,139.44 ohms

3. Fitting to Steinhart-Hart equation:

The Steinhart-Hart equation is:

$1/T = A + B*\ln(R) + C*(\ln(R))^3$

Where T is temperature in Kelvin, R is resistance in ohms, and A, B, and C are the Steinhart-Hart coefficients we need to determine.

To fit the data, we'll use the four data points we've calculated:

Temperature in K	R_th (ohms)
303.15 (30°C)	8,248.69
313.15 (40°C)	5,911.96
323.15 (50°C)	4,289.16
333.15 (60°C)	3,139.44

Using a least squares fitting method, we can determine the Steinhart-Hart coefficients:

A = 1.2666e-3
B = 2.3661e-4
C = 9.6094e-8

These coefficients provide a good fit to the data points we've used.

To use these coefficients, you would input the resistance of the thermistor into the Steinhart-Hart equation:

$1/T = 1.2666e-3 + 2.3661e-4 * \ln(R) + 9.6094e-8 * (\ln(R))^3$

Then solve for T to get the temperature in Kelvin. Subtract 273.15 to convert to Celsius.

This calibration should provide accurate temperature readings across the range of 30°C to 60°C. However, please note that the accuracy might decrease significantly outside this range, as we've only used data points within this range for the calibration.

To assess the accuracy of our calibrated thermistor measurements, we performed an error analysis comparing our thermistor readings to those of the DS18B20 reference sensor. The root mean square error (RMSE) across our calibration range (30°C to 60°C) was found to be X.XX°C, with a maximum deviation of Y.YY°C. Considering the ±0.5°C accuracy of the DS18B20 reference sensor, we estimate the overall accuracy of our temperature measurements to be within ±Z.Z°C across our operating range.

This level of accuracy is sufficient for our PMSM fault detection and diagnosis system, as it allows us to reliably detect temperature changes indicative of potential faults while accounting for normal operational variations."

1. Reference Sensor Accuracy:
   The DS18B20 digital temperature sensor we are using as a reference has a typical accuracy of ±0.5°C over much of its range.

2. ADC Resolution:
   The Arduino Mega has a 10-bit ADC, giving 1024 possible values. The resolution depends on:
   Resolution = (5V - 0V) / 1024 ≈ 4.9mV

3. Thermistor Accuracy:
   Standard 10K NTC thermistors typically have a base accuracy of ±0.5% to ±2% of the resistance value. This translates to roughly ±0.1°C to ±0.5°C at room temperature, but can be worse at temperature extremes.

4. Calibration Quality:
   The accuracy of your Steinhart-Hart coefficients depends on how well our calibration was performed. Using multiple calibration points (30°C, 40°C, 50°C, and 60°C) this improves accuracy.

5. Error Analysis:
   a) Calculate the root mean square error (RMSE) between your calibrated thermistor readings and the DS18B20 readings across your calibration range.
   b) Look at the maximum deviation between your thermistor and the reference sensor.