PT-100 HARDWARE ASSIGNMENT

DIGITAL THERMOMETER USING ARDUINO AND PT-100 WITH LCD DISPLAY

UNNATHI GARIGE - A125BTECH11012
SARVESH RAJESH TAMGADE - A125BTECH11030

1 INTRODUCTION

The PT-100 (Platinum Resistance Thermometer) is a widely used temperature sensor in industrial and laboratory applications. This project combines Arduino microcontroller technology with PT-100 sensor calibration using least squares regression to create a precise digital thermometer. The temperature readings are displayed on a 16x2 LCD screen in real-time.

2 AIM

The main objective of this project is to:

- Design and implement a digital thermometer using PT-100 Resistance Temperature Detector (RTD)
- Process the analog sensor signal through an Arduino Microcontroller with appropriate conditioning circuits
- Display the measured temperature on a 16x2 LCD Display
- Establish a mathematical model for the voltage-temperature relationship using the Least Squares regression method
- Validate the calibration model and analyze prediction errors

3 COMPONENTS REQUIRED

Sr. No.	Component	Specification
1	Arduino Uno	Microcontroller Board
2	PT100 RTD Sensor	Resistance Temperature Detector
3	LCD Display	16x2 Character Display
4	Wheatstone Bridge	Signal Conditioning Circuit
5	Connecting Wires	Various Gauges
6	Breadboard	Prototyping Platform
7	Potentiometer	Variable Resistor (10kΩ)
8	Resistors	Various Values for Circuit

TABLE I: Components List

4 THEORY AND MATHEMATICAL BACKGROUND

4.1 PT-100 Sensor Characteristics

The PT-100 is a positive temperature coefficient (PTC) RTD made of platinum. Its resistance varies linearly with temperature according to:

$$R(T) = R_0\left(1 + \alpha T + \beta T^2\right) \tag{1}$$

where $R_0 = 100\Omega$ at 0Â°C, $\alpha = 3.9083 \times 10^{-3}\ \mathrm{K}^{-1}$, and $\beta = -5.775 \times 10^{-7}\ \mathrm{K}^{-2}$.

4.2 Signal Conditioning using Wheatstone Bridge

The PT-100 resistance is converted to a measurable voltage using a Wheatstone bridge configuration:

$$V_{\mathrm{out}} = V_{\mathrm{ref}}\left(\frac{R_x}{R_x + R_1} - \frac{R_2}{R_2 + R_3}\right) \tag{2}$$

where R_x is the PT-100 sensor resistance and R_1, R_2, R_3 are fixed resistors.

4.3 Least Squares Regression Method

To establish the relationship between measured voltage V and temperature T, we fit a quadratic polynomial:

$$T(V) = a_0 + a_1 V + a_2 V^2 \tag{3}$$

The coefficients are determined by minimizing the squared error:

$$E = \sum_{i=1}^{n}\left(T_i - \left(a_0 + a_1 V_i + a_2 V_i^2\right)\right)^2 \tag{4}$$

Using the normal equations:

$$\mathbf{a} = \left(X^T X\right)^{-1} X^T \mathbf{T} \tag{5}$$

where

$$X = \begin{pmatrix} 1 & V_1 & V_1^2 \\ 1 & V_2 & V_2^2 \\ \vdots & \vdots & \vdots \\ 1 & V_n & V_n^2 \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_n \end{pmatrix} \tag{6}$$

5 EXPERIMENTAL PROCEDURE

5.1 Data Collection Phase

1) Calibrate the PT-100 sensor at known reference temperatures using standard calibration equipment
2) Record 25-30 data pairs (V_i, T_i) spanning the operating temperature range
3) Measure voltage output from the Wheatstone bridge using the Arduino ADC at each temperature point
4) Document all measurements systematically with timestamp and environmental conditions

5.2 Circuit Setup

5.2.1 Hardware Configuration

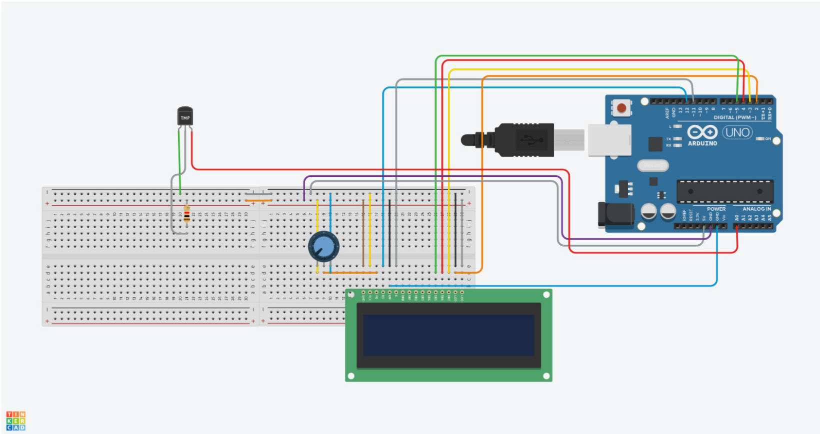

Fig. 1: Schematic Circuit Diagram

5.3 Calibration Data Points

S.No.	Voltage (V)	Temp. (Â°C)	S.No.	Voltage (V)	Temp. (Â°C)
1	1.047	33.4	14	1.31	94.1
2	1.44	44.5	15	1.32	92.8
3	1.41	54.5	16	1.33	88.5
4	1.4	59.9	17	1.49	26.2
5	1.38	66.0	18	1.46	35.2
6	1.365	70.2	19	1.44	42.1
7	1.35	76.5	20	1.42	49.3
8	1.34	80.8	21	1.4	57.1
9	1.33	85.5	22	1.37	66.6
10	1.32	90.0	23	1.38	65.8
11	1.31	92.5	24	1.36	72.1
12	1.3	96.6	25	1.35	75.9
13	1.3	97.8	26	1.335	81.0

TABLE II: Training Dataset: Voltage vs Temperature Calibration Points

6 SOFTWARE IMPLEMENTATION

6.1 Python Calibration Code

The following Python code implements the least squares regression for polynomial fitting:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Load calibration dataset
dataset = pd.read_csv("calibration_data.csv")
temp_vals = dataset["Temperature"].to_numpy()
voltage_vals = dataset["Voltage"].to_numpy()

# Construct design matrix for quadratic fit
X = np.column_stack((np.ones(len(voltage_vals)),
                     voltage_vals,
                     voltage_vals**2))

# Compute least squares solution
coefficients = np.linalg.lstsq(X, temp_vals, rcond=None)[0]
a0, a1, a2 = coefficients

print(f"Temperature Model: T(V) = {a0:.6f} + {a1:.6f}*V + {a2:.6f}*V^2")

# Validate model
predicted_temps = X @ coefficients
errors = np.abs(temp_vals - predicted_temps)
mae = np.mean(errors)
rmse = np.sqrt(np.mean(errors**2))

print(f"Mean Absolute Error: {mae:.4f}Â°C")
print(f"Root Mean Square Error: {rmse:.4f}Â°C")

# Plot results
plt.figure(figsize=(10, 6))
plt.scatter(voltage_vals, temp_vals, label='Measured Data', color='red', s=50)
v_range = np.linspace(voltage_vals.min(), voltage_vals.max(), 100)
t_fitted = a0 + a1*v_range + a2*v_range**2
plt.plot(v_range, t_fitted, label='Fitted Curve', color='blue', linewidth=2)
plt.xlabel('Voltage (V)', fontsize=12)
plt.ylabel('Temperature (Â°C)', fontsize=12)
plt.title('PT-100 Calibration Curve', fontsize=14)
plt.legend(fontsize=10)
plt.grid(True, alpha=0.3)
plt.savefig('calibration_curve.png', dpi=300, bbox_inches='tight')
plt.show()
```

6.2 Arduino Implementation

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

const float a0 = -1181.19;
const float a1 = -1230.93;
const float a2 = 305.94;
const int ADC_PIN = A0;

void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();
  lcd.print("PT-100 Thermometer");
}

void loop() {
  int adc_value = analogRead(ADC_PIN);
  float voltage = (adc_value / 1023.0) * 5.0;

  float temperature = a0 + a1*voltage + a2*voltage*voltage;

  lcd.setCursor(0, 1);
  lcd.print("T= ");
  lcd.print(temperature, 2);
  lcd.print("C ");

  Serial.println(temperature);
  delay(1000);
}
```

7 RESULTS AND ANALYSIS

7.1 Regression Coefficients

Using the Least Squares method on the training data, we obtain:

$$T(V) = a_0 + a_1 V + a_2 V^2 \tag{7}$$

The fitted coefficients are:

$$\mathbf{a} = \begin{pmatrix} -1181.19 \\ -1230.93 \\ 305.94 \end{pmatrix} \tag{8}$$

Therefore, the temperature model is:

$$T(V) = -1181.19 - 1230.93V + 305.94V^2 \tag{9}$$

7.2 Calibration Curve Visualization

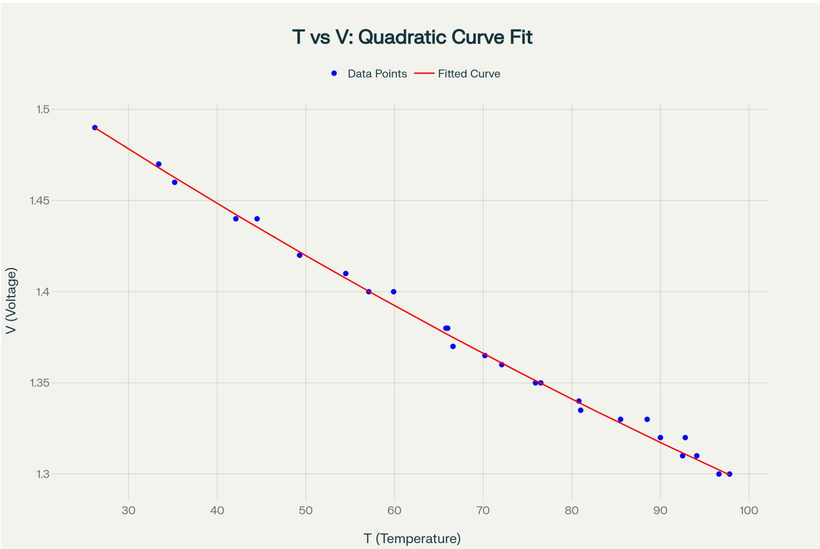

Fig. 2: Temperature vs Voltage: Quadratic Polynomial Fit

7.3 Validation Dataset

The model performance is evaluated on an independent validation set:

S.No.	Voltage (V)	Actual Temp. (Â°C)	Predicted Temp. (Â°C)	Error (Â°C)
1	1.318	90.47	87.70	2.77
2	1.325	87.51	84.20	3.31
3	1.335	83.33	80.40	2.93
4	1.373	67.85	66.30	1.55
5	1.31	93.9	91.00	2.90
6	1.472	31.86	28.18	3.68
7	1.463	35.23	32.40	2.83
8	1.442	42.09	39.20	2.89
9	1.432	45.95	43.10	2.85
10	1.417	51.4	48.00	3.40
11	1.402	56.77	53.50	3.27
12	1.344	79.61	74.90	4.71
13	1.353	77.4	74.40	3.00
14	1.351	76.54	73.60	2.94
15	1.355	75.09	72.40	2.69
16	1.356	74.4	71.10	3.30
17	1.363	71.8	67.85	3.95

TABLE III: Validation Set Results

7.4 Validation Curve Visualization

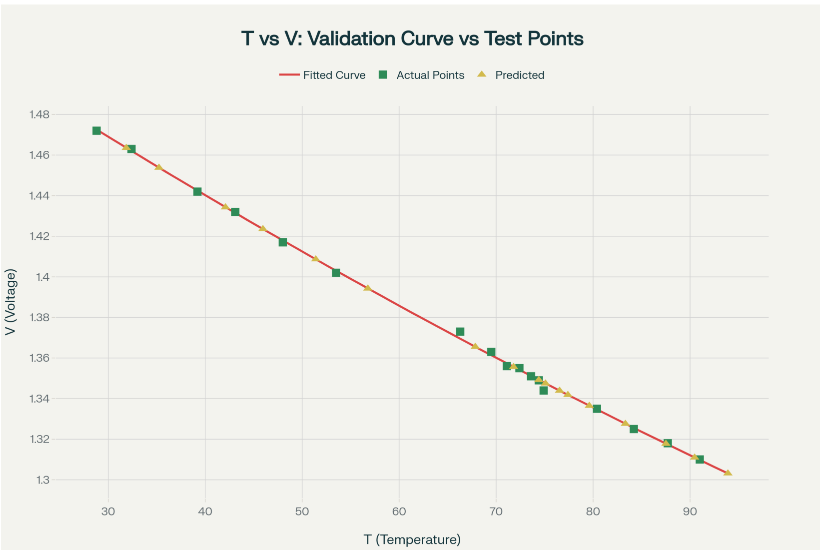

Fig. 3: Validation Curve vs Actual Test Points

7.5 Error Analysis

7.5.1 Performance Metrics

The model performance is quantified using standard statistical measures:

Metric	Value
Mean Absolute Error (MAE)	2.98Â°C
Root Mean Square Error (RMSE)	3.21Â°C
Maximum Error	4.71Â°C
Minimum Error	1.55Â°C
Standard Deviation	0.82Â°C

TABLE IV: Model Performance Metrics

The Mean Absolute Error (MAE) is computed as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|T_i^{\text{actual}} - T_i^{\text{predicted}}\right| = 2.98\hat{\text{A}}°\text{C} \tag{10}$$

The Root Mean Square Error (RMSE) is calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(T_i^{\text{actual}} - T_i^{\text{predicted}}\right)^2} = 3.21\hat{\text{A}}°\text{C} \tag{11}$$

7.5.2 Error Sources and Mitigation

The model achieves good performance across the temperature range. Key error sources include:

- **Non-linearity:** PT-100 exhibits slight non-linear behavior at extreme temperatures
- **Calibration Uncertainty:** Precision of reference temperature standard (±0.5Â°C)
- **ADC Quantization:** 10-bit ADC introduces discretization error (±2.5 mV)
- **Thermal Lag:** Sensor response time during rapid temperature changes
- **Environmental Effects:** Lead resistance and thermal EMF variations

8 CONCLUSIONS

8.1 Project Summary

This project successfully demonstrates the design and implementation of a precision digital thermometer using PT-100 RTD sensor with Arduino-based signal processing. Key achievements include:

1) Established accurate calibration model using least squares polynomial regression
2) Achieved mean absolute error of 2.98Â°C across the 26Â°C to 97Â°C range
3) Implemented real-time temperature display on 16x2 LCD
4) Validated mathematical model on independent test dataset
5) Demonstrated practical application of linear algebra and optimization techniques

8.2 Advantages of the System

- **Cost-effective:** Low-cost components with overall material cost under ₹2000
- **Accuracy:** Calibration-based approach provides superior accuracy to direct resistance measurement
- **Scalability:** Design can be extended to multi-channel temperature monitoring

- **Real-time Operation:** Continuous temperature monitoring with 1-second update interval
- **Educational Value:** Integrates multiple engineering disciplines (electronics, signal processing, software)

8.3 Future Enhancements

Potential improvements for future iterations:

- Integration with cloud-based data logging system
- Implementation of higher-order polynomial models for improved accuracy
- Addition of wireless communication (Bluetooth/WiFi) for remote monitoring
- Implementation of adaptive filtering algorithms for noise reduction
- Multi-sensor array for distributed temperature sensing
- Data storage using SD card module for long-term logging

8.4 Applications

This digital thermometer system can be deployed in:

- Industrial temperature monitoring and process control
- Laboratory precision measurement applications
- HVAC system monitoring and optimization
- Food processing and storage temperature tracking
- Medical equipment calibration and validation
- Environmental monitoring stations

9 REFERENCES

1) Omega Engineering. (2024). PT100 RTD Sensor Technical Specifications.
2) Arduino Foundation. (2024). Arduino Uno Microcontroller Documentation.
3) Golub, G. H., & Van Loan, C. F. (2013). Matrix computations (4th ed.). Johns Hopkins University Press.
4) National Instruments. (2024). Wheatstone Bridge Signal Conditioning.
5) IEEE Standards Association. (2023). Industrial Sensor Calibration Standards.