Plotting Live Data of Temperature Sensor using MATLAB

Abstract

This project focuses on acquiring real-time temperature data from an Arduino-based sensor circuit and plotting it live using MATLAB. The system establishes serial communication between Arduino and MATLAB, where Arduino continuously sends raw sensor values. MATLAB reads this data, converts it into Celsius and Fahrenheit, applies smoothing through a moving average filter, and plots the temperature variations dynamically. A user-friendly display includes both the live graph and a textbox showing the latest and average temperature readings. This approach demonstrates how MATLAB can be effectively used for real-time data acquisition, monitoring, and visualization.

Introduction

Definition

Real-time data plotting refers to the process of acquiring live data from sensors, processing it, and displaying it graphically as the data is collected. In this project, a temperature sensor connected to Arduino sends continuous readings to MATLAB, which processes and visualizes the data in real time.

Applications

1. Weather Monitoring – Recording and analyzing temperature variations.
2. IoT Systems – Live environmental data monitoring for smart homes.
3. Industrial Monitoring – Supervising temperature in manufacturing units.
4. Healthcare – Monitoring patient temperature trends.
5. Academic Demonstrations – Teaching real-time data acquisition and plotting.

Objectives

6. To acquire live temperature sensor data from Arduino using MATLAB.
7. To implement real-time plotting of temperature in both °C and °F.
8. To apply a moving average filter for noise reduction.
9. To display live current values and average values in a user-friendly interface.
10. To integrate hardware and software for seamless real-time monitoring.

Hardware Requirements

- Arduino Uno board
- LM35 temperature sensor
- Jumper wires
- Breadboard
- USB cable for Arduino-PC communication

Software Requirements

- Arduino IDE (for uploading sensor code)
- MATLAB (for data acquisition, filtering, and plotting)

Circuit Diagram

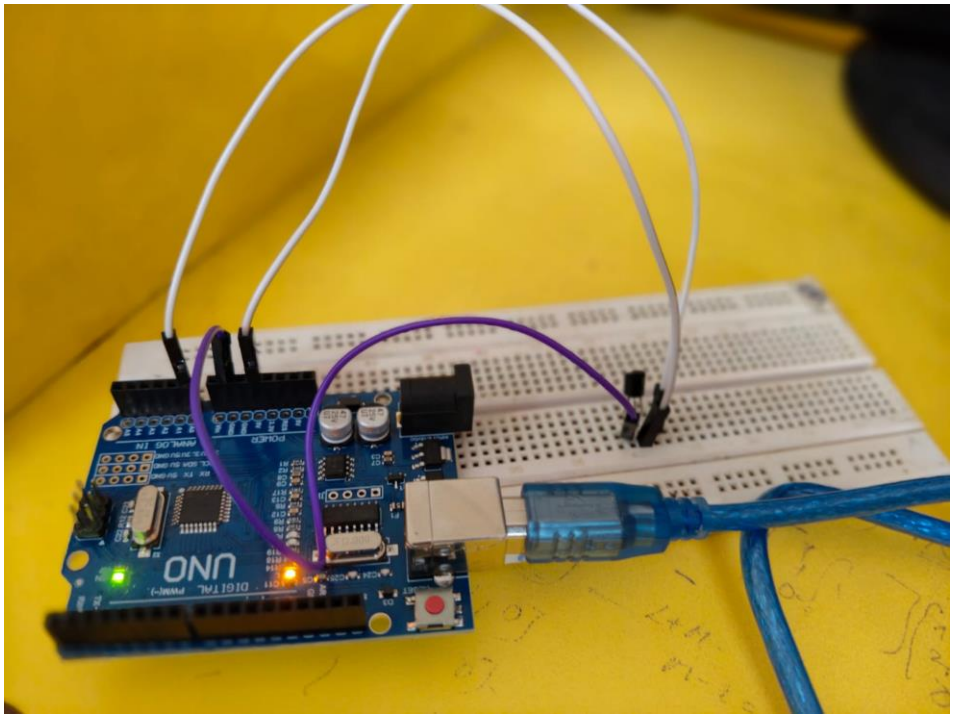

Fig. 1: Circuit connection of Arduino with LM35 sensor

Output

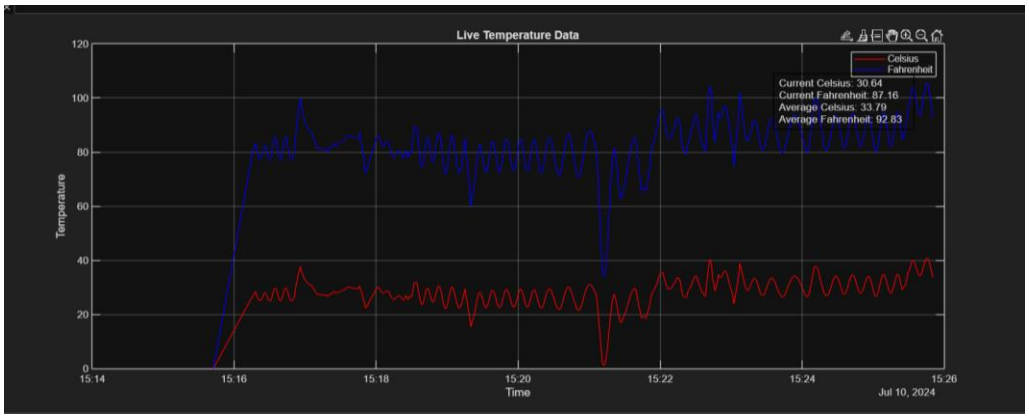

Fig. 2: Live plotting of temperature data in °C (red) and °F (blue).

Conclusion

The project successfully demonstrates real-time data acquisition and plotting using Arduino and MATLAB. The LM35 sensor provides temperature readings, which are transmitted to MATLAB for live visualization. By implementing a moving average filter, the system reduces fluctuations and improves accuracy. The real-time graph and textual display provide clear insights into both instantaneous and average temperature. This project highlights the integration of hardware (Arduino) and software (MATLAB) for effective real-time monitoring applications.

Future Scope

- Integration with wireless communication (e.g., Wi-Fi, Bluetooth) for remote monitoring.
- Extension to IoT platforms with cloud-based dashboards for global accessibility.
- Addition of multiple sensors for humidity, pressure, or air quality monitoring.
- Development of a mobile application for real-time temperature visualization.
- Implementation of predictive analytics to forecast temperature trends.

TEAM

K. GEETHIKA CHOWDARY (22WH1A0497)

T.N.PRAJNYA VARSHINI (22WH1A0495)

K. YAAMINI PADMA REDDY (22WH1A04C2)

B. DEEKSHITHA (22WH1A0488)