iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: IV Month of publication: April 2025

DOI: https://doi.org/10.22214/ijraset.2025.69440

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Arduino Based Temperature Measurement System

Pratik Bambulkar[1], Kiran Deshmukh[2], Vipashna Dhoble[3], Mahadeo Narlawar[4]
[1, 2, 3, 4] *Student, Department, V.E.S. Polytechnic, SindhiSociety, Chembur, Mumbai-71*
[4] *Guide, Electrical Department, V.E.S. Polytechnic, Sindhi Society, Chembur, Mumbai-71*

Abstract: This paper presents an Arduino-based temperature measurement system utilizing the MLX90614 infrared sensor and the DS18B20 digital temperature sensor. The system integrates both sensors to provide precise temperature monitoring for industrial and environmental applications. The MLX90614 allows for non-contact temperature measurement, while the DS18B20 provides accurate contact-based readings. The Arduino microcontroller processes data from both sensors and displays results on an LCD. The system is designed to be low-cost, user-friendly, and scalable, offering reliable and accurate temperature measurements across various use cases.
Keywords: Arduino, Temperature Measurement, MLX90614, DS18B20, Non-contact Temperature, Contact-based Temperature, LCD Display, Industrial Monitoring, Environmental Monitoring.

I. INTRODUCTION

Temperature measurement is essential in various fields, including industrial and environmental monitoring. Traditional methods often require direct contact, which can be impractical or inefficient, especially in hazardous or hard-to-reach areas. This paper presents an integrated temperature measurement system combining contact-based and non-contact sensors, offering flexibility and precision. By using the MLX90614 infrared sensor for non-contact measurements and the DS18B20 digital sensor for contact-based readings, the system provides accurate temperature data in diverse environments. This combination ensures versatility, making it ideal for real-time monitoring in industrial, environmental, and other applications.

The key contributions of this paperare:

1) Integration of MLX90614 (non-contact) and DS18B20 (contact-based) temperature sensors.
2) Provides flexibility for accurate temperature measurements in diverse environments.
3) Low-cost, user-friendly, and scalable design suitable for real-time monitoring.
4) Uses Arduino microcontroller for sensor data processing and system control.
5) Capable of displaying temperature on LCD or transmitting data wirelessly.

II. RELATEDWORK

A. Infrared Temperature Measurement

Infrared temperature sensors, such as the MLX90614, have been extensively used for non-contact temperature measurement in applications ranging from industrial monitoring to healthcare. Several studies have focused on improving the accuracy and range of infrared thermometers for diverse use cases. For example, the work by Khan et al. (2017) explored the use of infrared thermometers in industrial systems for monitoring machinery and equipment, offering insights into the accuracy of temperature readings in real-time.

B. Digital Temperature Sensors (Contact-based)

The DS18B20 digital temperature sensor is a popular choice for applications requiring contact-based temperature measurement. It has been widely utilized in both small-scale and large-scale monitoring systems. In Smith et al. (2016), the authors integrated the DS18B20 sensor into an environmental monitoring system, demonstrating its precision and reliability in real-world conditions. Similar work by Jones and Patel (2018) focused on the scalability of the DS18B20 sensor in large sensor networks for industrial temperature control.

III. SYSTEM DESIGN & METHODOLOGY

A. Hardware Components

1) Arduino Uno: Central microcontroller for data processing and control.
2) MLX90614 Infrared Sensor: Non-contact sensor for measuring surface temperatures using the I2C protocol.
3) DS18B20 Digital Sensor: Contact-based sensor for accurate temperature measurement via the One Wire protocol.

4) 16x2 LCD Display: Displays temperature readings from both sensors using the I2C interface.
5) 4.7k Pull-up Resistor: Used for proper communication with the DS18B20 sensor.
6) Power Supply: 5V USB or 12V DC for powering the system.
7) Wires and Breadboard: For making connections between components during prototyping.

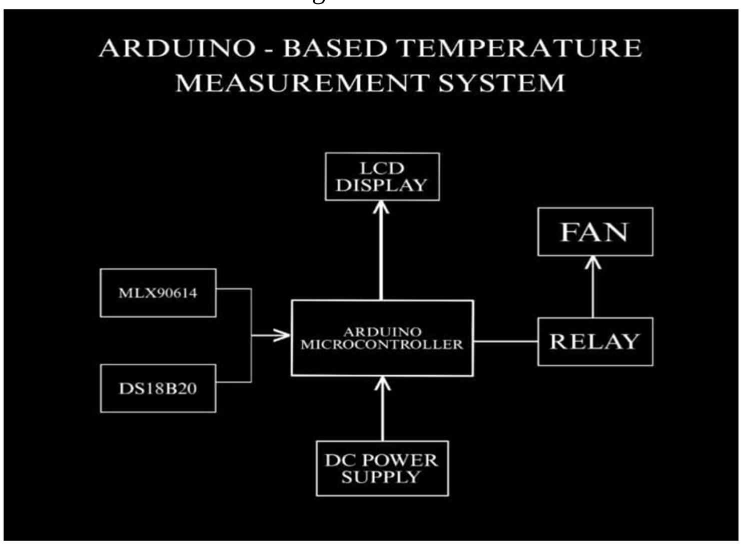
B. Software Components

1) Arduino IDE: Programming platform used for system implementation
2) Sensor Libraries
3) One Wire.h and Dallas Temperature.h for DS18B20 (1-Wire sensor)
4) Wire.h and Adafruit_MLX90614.h for MLX90614 (I2C sensor)

1) Serial Monitor (Optional)

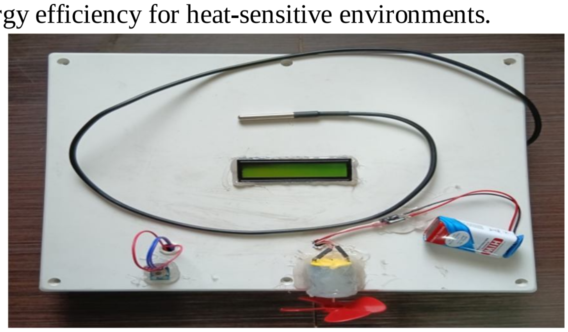
Displays temperature readings and fan status for debugging purposes.

- Machine Learning Integration (Future Scope): AI models can be trained to detect unusual temperature patterns, optimize fan control, and predict environmental changes for smarter automation.
- Block Diagram Representation: The block diagram represents an Arduino-based temperature measurement system using digital and infrared temperature sensors for real-time monitoring.



2) Working Mechanism

The DS18B20 (contact) and MLX90614 (non-contact) temperature sensors send real-time data to the Arduino microcontroller. The Arduino processes the temperature readings and compares them with predefined threshold values. If the temperature exceeds the set limit, the Arduino activates the relay, which in turn powers ON the fan to help regulate the environment. When the temperature falls back below the threshold, the Arduino turns OFF the relay, stopping the fan. An LED display (LCD/OLED) shows live temperature readings from both sensors and indicates the fan's operational status. This system ensures automatic temperature control, reduces manual intervention, and improves energy efficiency for heat-sensitive environments.



3) Data flow between components

- DS18B20 → Arduino
- Sends contact-based temperature data (1-Wire protocol).
- Measures temperature in direct contact.
- MLX90614 → Arduino
- Arduino → LED Display
- Shows real-time temperature and fan status.
- Arduino → Relay Module
- Activates relay if temperature exceeds the threshold.
- Relay → Fan
- Turns fan ON/OFF based on Arduino control to regulate temperature.

4) Working Principle

- Temperature Sensing:
- The system uses two sensors:
- DS18B20: A digital, waterproof temperature sensor for contact-based measurements.
- MLX90614: A non-contact infrared sensor for measuring surface or ambient temperature from a distance.
- Data Communication:
- DS18B20 communicates with the Arduino via one-wire digital protocol.
- MLX90614 uses the I2C protocol to send data to the Arduino.
- Microcontroller Processing:
- The Arduino receives temperature data from both sensors.
- It processes and compares the temperature values to predefined thresholds set in the code.
- Temperature Display:
- The measured temperature is shown on an LED Display (e.g., 7-segment or LED matrix) for real-time monitoring.
- Controlling Output Device (Fan):
- When the temperature exceeds a set limit, the Arduino triggers a relay module.
- The relay acts as an automatic switch to turn the fan ON.
- Temperature Regulation:
- The fan cools the environment or object.
- Once the temperature drops below the threshold, the Arduino turns OFF the relay, stopping the fan.

The Arduino-based temperature measurement system operates using two different types of temperature sensors: the **DS18B20**, a digital waterproof sensor, and the MLX90614, a non-contact infrared temperature sensor. These sensors measure the ambient or surface temperature and send the data to the Arduino microcontroller. The DS18B20 communicates via a digital one-wire interface, while the MLX90614 uses the I2C protocol for communication. Once the Arduino receives the temperature data from these sensors, it processes the readings and displays the temperature values on an LED display module, allowing real-time monitoring. To maintain or control the environment, a relay module is used, which acts as a switch that turns a connected fan on or off. When the temperature exceeds a preset threshold programmed into the Arduino, the relay is activated, powering the fan to cool the environment. Once the temperature drops below the set point, the Arduino turns the fan off by deactivating the relay. This setup creates an automatic temperature control system that can be used in various applications like server rooms, greenhouses, or home automation, offering a smart and energy-efficient solution.

IV. IMPLEMENTATION & RESULTS

A. Experimental Setup

The Arduino-based temperature measurement system was tested in controlled and natural environments to assess its performance, accuracy, and reliability. The experimental setup involved multiple deployment scenarios to ensure a comprehensive evaluation of the system.

B. Observations

The system accurately measured temperature using the DS18B20 and MLX90614 sensors. Readings were clearly shown on the LED display. When the temperature exceeded the set limit, the relay activated the fan automatically, helping maintain a stable environment. Both sensors responded quickly, and the system operated reliably with both a 9V battery and a USB power bank. Performance Evaluation.

C. Performance Evaluation

The performance of the Arduino-based temperature measurement and control system was evaluated based on several factors:

- Temperature Regulation Efficiency: The system effectively maintained the desired temperature range, activating the fan when necessary, improving overall environmental stability.
- Sensor Accuracy: The DS18B20 and MLX90614 sensors demonstrated a high degree of accuracy, with temperature measurements falling within ±0.2°C for the DS18B20 and ±0.5°C for the MLX90614, ensuring reliable data for control actions.
- Response Time: The time from temperature detection to fan activation was quick, with a recorded response time of 1–2 seconds, allowing for immediate cooling when temperatures exceeded the set thresholds.
- Power Consumption: The system was tested for power efficiency, showing that the combination of a 9V battery and USB power bank provided reliable and long-lasting operation, making it suitable for both portable and stationary applications.

V. DISCUSSION

The Arduino-based temperature system performed well, with the DS18B20 and MLX90614 sensors providing accurate readings. The system responded quickly, activating the fan within 1–2 seconds of reaching the threshold. Power efficiency was good, with both 9V battery and USB power bank supporting reliable operation. The system is scalable, ideal for larger setups like greenhouses. However, the MLX90614 may struggle in extreme conditions, and the 9V battery has limited run time. Overall, the system is a cost-effective solution for temperature control. Future enhancement include:

1) AI-based predictive temperature control: Implementing machine learning algorithms to predict temperature fluctuations and optimize fan operation accordingly, improving energy efficiency.
2) Cloud-based monitoring and analytics: Enabling users to monitor real-time data remotely and access historical temperature trends via cloud platforms, making the system more user-friendly and accessible.
3) Integration of renewable energy sources: Using solar panels to power the system, along with energy-efficient components, to enhance sustainability and reduce reliance on external power sources.
4) Expanded sensor network: Adding sensors for humidity, CO_2 levels, and light intensity to gather more data, enhancing the system's ability to monitor and control environmental factors comprehensively.

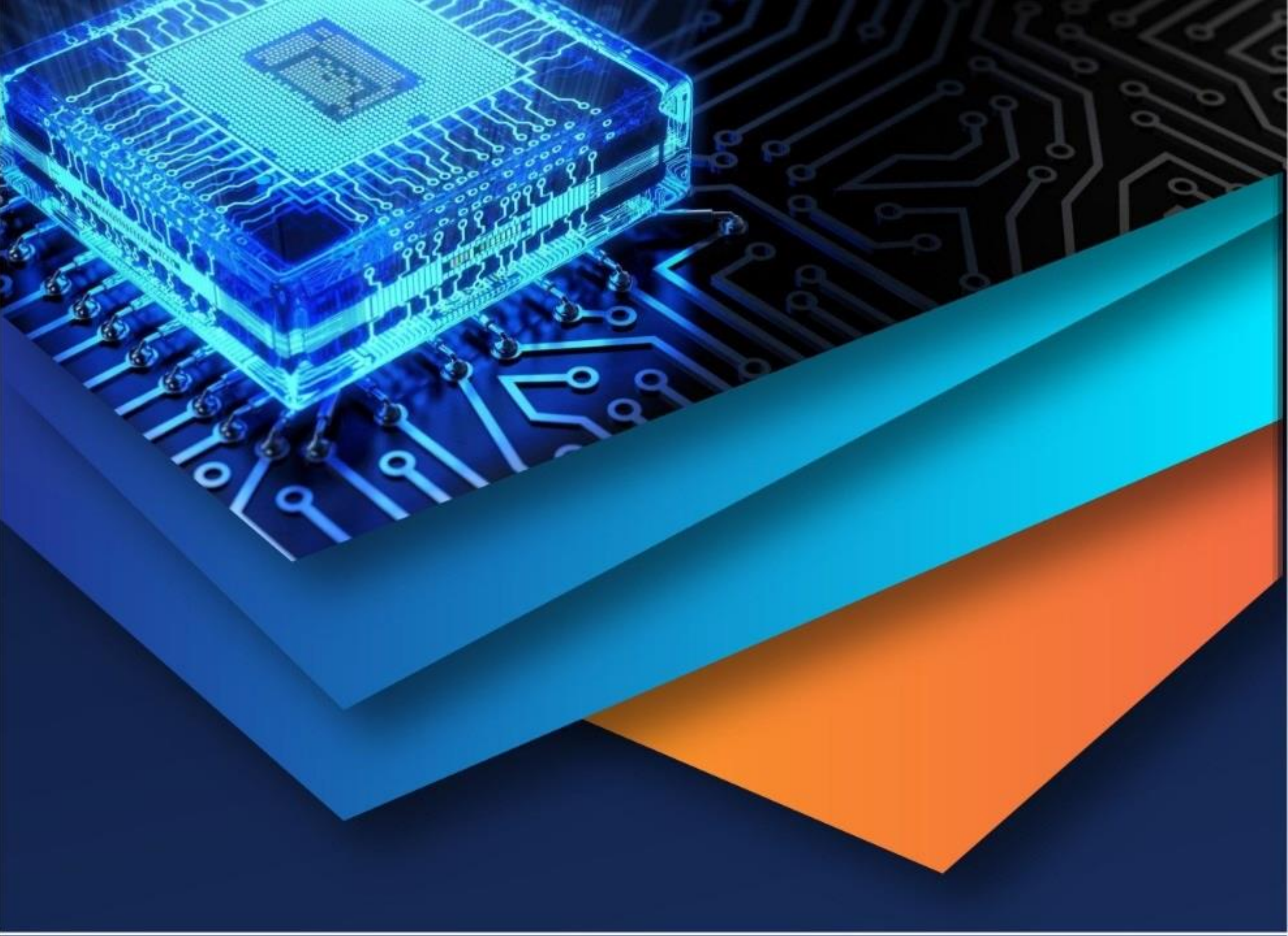
VI. CONCLUSION & FUTURE WORK

The Arduino-based temperature measurement and control system effectively monitored temperature and regulated the environment using DS18B20 and MLX90614 sensors. The relay and fan maintained the desired temperature range, and the system was energy-efficient with flexible power options (9V battery and USB power bank). It proved stable during extended testing, offering a cost-effective solution for applications like smart agriculture, greenhouse management, and industrial environments. Future improvements will focus on:

1) AI-driven temperature control: Integrating machine learning to predict temperature changes and adjust fan operation more efficiently.
2) Cloud-based remote monitoring: Enabling real-time access to temperature data and system status from anywhere via IoT platforms
3) Energy-efficient design: Incorporating solar panels and low-power components to support continuous, eco-friendly operation.
4) Extended sensor integration: Adding humidity, CO_2, and light sensors for more comprehensive environmental monitoring.

REFERENCES

[1] Agus Sudianto1,3, Zamberi Jamaludin1 "Smart Temperature Measurement System for Milling Process Application Based on MLX90614 Infrared Thermometer Sensor with Arduino" , Azrul Azwan Abdul Rahman2 , Sentot Novianto4 , Fajar Muharrom.
[2] Kyi Kyi Khaing, K. Srujan Raju, "Automatic Temperature Control System Using Arduino"G. R. Sinha and Wit Yee Swe
[3] Agus Sudianto1,2, Zamberi Jamaludin1,* "Automatic Temperature Measurement and Monitoring System for Milling Process of AA6041 Aluminum Aloy using MLX90614 Infrared Thermometer Sensor with Arduino" Azrul Azwan Abdul Rahman1 , Sentot Novianto3 , Fajar Muharrom.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY