Industrial Internship Report on

“IoT Based Temperature Monitoring”

Prepared by

Ajit Raghunath Avhad

Executive Summary

This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks’ time.

My project was to design and implement an IoT Based Temperature Monitoring system using ESP32. The system reads temperature using an LM35 sensor, processes the data using ESP32’s built-in ADC, and displays the temperature on a 16x2 LCD.

This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solutions. It was an overall great experience.

TABLE OF CONTENTS

1.Preface

Over the span of 6 weeks, I worked on an IoT Based Temperature Monitoring system as part of my internship under UniConverge Technologies Pvt Ltd (UCT). The goal was to design a real-time monitoring system using ESP32 and LM35.

This internship gave me hands-on experience in circuit design, embedded system programming, sensor interfacing, and debugging. The structured guidance and industry mentorship helped me gain confidence in practical applications.

I would like to thank upskill Campus, The IoT Academy, and UCT for this wonderful opportunity. I also extend gratitude to all mentors, peers, and faculty members who guided me throughout the project.

To my juniors and peers – take every opportunity seriously and focus on applying your skills in practical, real-world projects.

2.Introduction

2.1 About UniConverge Technologies Pvt Ltd

UniConverge Technologies Pvt Ltd (UCT), established in 2013, is a pioneer in the field of Digital Transformation. The company provides innovative industrial solutions with a focus on sustainability and Return on Investment (RoI).

UCT leverages cutting-edge technologies such as the Internet of Things (IoT), Cybersecurity, Cloud Computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, and Python to design scalable, high-impact solutions.

UCT's platforms, such as UCT Insight and Factory Watch, support rapid IoT deployment and smart factory management. These platforms allow integration via standard protocols (MQTT, HTTP, Modbus TCP, OPC UA), and offer features like customizable dashboards, analytics, real-time alerts, and third-party integrations (e.g., Power BI, ERP).

2.2 About upskill Campus

upskill Campus (USC) is a career enhancement platform that provides learners with practical experience through industry-led internship programs. In collaboration with The IoT Academy and UCT, USC delivers project-based learning opportunities in emerging technologies.

Their mission is to upskill over one million learners within five years by providing self-paced learning, expert mentorship, real-world projects, and internship support. USC plays a critical role in bridging the gap between theoretical learning and industry requirements.

2.3 Objective

The primary objectives of this internship program were to:

- Gain hands-on experience in IoT and embedded systems.
- Learn about microcontroller programming, sensor interfacing, and embedded circuit design.
- Develop real-time data acquisition and display systems using ESP32 and LM35.
- Solve practical, industry-relevant problems through a guided project.
- Enhance communication, collaboration, and technical documentation skills.
- Improve job-readiness through experiential learning.

2.4 Reference

- [1] https://www.uniconvergetech.com
- [2] https://www.upskillcampus.com
- [3] Datasheets and documentation for ESP32, LM35, and I2C LCD

2.5 Glossary

- ESP32 – A Wi-Fi and Bluetooth-enabled microcontroller used for IoT applications.
- LM35 – A precision temperature sensor that provides analog voltage output proportional to temperature.
- ADC – Analog to Digital Converter; converts analog input signals into digital values.
- I2C – Inter-Integrated Circuit; a two-wire communication protocol used for connecting peripherals.
- IoT – Internet of Things; a network of physical devices that collect and exchange data.
- LCD – Liquid Crystal Display; an output device used to display alphanumeric data.

3. Problem Statement

In many industrial, agricultural, and residential environments, maintaining and monitoring temperature is critical for safety, efficiency, and quality control. Traditional temperature monitoring systems are often bulky, expensive, and lack real-time accessibility or scalability. Moreover, in remote or resource-limited settings, these systems are not feasible due to infrastructure or cost constraints.

The problem is to design and implement a low-cost, real-time temperature monitoring system that is:

- Accurate in measuring environmental temperature,
- Compact and scalable, suitable for integration in different applications,
- Energy efficient, and
- Capable of displaying and potentially transmitting data wirelessly.

The aim is to build this system using modern embedded hardware (ESP32), an analog temperature sensor (LM35), and a user-friendly output interface (16x2 LCD). This device should be able to read analog signals, convert them to digital temperature values, and provide immediate feedback through a display. The design should allow potential upgrades like IoT-based data logging or cloud integration.

Here is the **Existing and Proposed Solution** section in professional text format, suitable for your internship report on the **IoT Based Temperature Monitoring** project:

4. Existing and Proposed Solution

Existing Solutions

Temperature monitoring systems are commonly used across industries such as manufacturing, agriculture, healthcare, and environmental monitoring. Traditional systems typically rely on:

- Manual thermometers or standalone digital devices, which provide only local readings and require physical presence for monitoring.
- Wireless sensor networks (WSNs), which offer remote monitoring but often involve high deployment costs and complex configurations.
- SCADA-based systems, used in industrial settings, that are powerful but expensive and energy-consuming.
- Commercial IoT products, which may offer cloud-based monitoring but are proprietary, expensive, and difficult to customize or scale for educational or small-scale industrial use.

The major limitations of existing solutions include:

- High cost and lack of affordability for small-scale applications.
- Lack of customization for specific environmental needs.
- Limited integration with modern microcontrollers or open-source platforms.
- Dependency on closed-source ecosystems.

Proposed Solution

To address the limitations of current systems, we propose an affordable and scalable IoT-Based Temperature Monitoring System using:

- ESP32 microcontroller: Offers integrated Wi-Fi, Bluetooth, and built-in ADC for compact and wireless IoT applications.
- LM35 sensor: A low-cost analog temperature sensor with a linear output voltage proportional to temperature.
- 16x2 LCD with I2C: Provides a simple, readable interface for local data display while reducing the number of required GPIO pins.

This solution offers the following value additions:

- Cost-effective: Uses easily available, low-power, and low-cost components.
- Real-time monitoring: Continuous temperature updates every second.
- Expandable: Designed with scalability in mind—wireless data transmission, logging, or mobile app integration can be added later.
- Educational utility: Ideal for student and academic use, promoting hands-on learning of embedded systems and IoT.

By utilizing open-source hardware and software tools, this system ensures easy replication, modification, and deployment across different domains.

4.1 Code submission:

https://github.com/fdfgfdvdvrhytgnkkjmfvswwtrhvddbtkjy/iot-based-temp-monitoring

5. Proposed Design/Model

The design of the IoT-Based Temperature Monitoring System follows a modular and scalable architecture that includes sensing, processing, displaying, and transmitting temperature data. The system is built using an ESP32 microcontroller, LM35 temperature sensor, 16x2 I2C LCD, and optional cloud connectivity via the Blynk IoT platform.

5.1 High Level Diagram

At a high level, the system performs the following steps:

1. Sensing: The LM35 analog temperature sensor continuously monitors the surrounding temperature.
2. Processing: The ESP32 reads the analog signal via its built-in ADC, processes the voltage, and calculates the temperature.
3. Display: The temperature is displayed on a 16x2 LCD using the I2C interface.
4. Transmission (Optional): The temperature data is sent to the Blynk IoT platform via Wi-Fi, allowing remote monitoring through a smartphone.

5.2 Low Level Diagram

The low-level design focuses on the specific pin configurations and data flow:

- LM35 → Connected to GPIO 34 (Analog Input)
- LCD I2C → SDA to GPIO 21, SCL to GPIO 22
- Wi-Fi Credentials → Configured in code to connect to Blynk Cloud
- Virtual Pin V0 → Used for sending temperature data to the Blynk dashboard

5.3 Interfaces

The system interfaces include:

- Analog Interface: LM35 outputs analog voltage proportional to temperature (10 mV/°C)
- I2C Interface: LCD communicates with ESP32 via I2C (2-wire) protocol to reduce GPIO usage
- Wi-Fi Interface: ESP32 connects to the internet using Wi-Fi credentials to send data to the Blynk cloud platform
- Mobile App Interface: Blynk app displays real-time temperature using a virtual pin

6. Performance Test

Performance testing is a critical aspect of validating any embedded system, especially when it involves real-time monitoring and potential industrial applications. The main objective of this section is to assess the system's reliability, responsiveness, accuracy, and scalability under defined operational constraints.

This IoT-based temperature monitoring system was evaluated for its data accuracy, system responsiveness, stability over time, and feasibility for continuous use in real-world scenarios.

6.1 Test Plan / Test Cases

The testing was conducted under indoor conditions over multiple days to simulate typical environmental conditions. The primary test cases included:

Test Case	Description	Expected Result	Actual Result
TC1	Read temperature at room temperature	25–30 °C	26.5 °C
TC2	Response time of data update on LCD	< 1 sec	0.8 sec
TC3	Response time for update on Blynk app	< 2 sec	~1.5 sec
TC4	Voltage-to-temperature conversion	Linear (10mV/°C)	Passed
TC5	Accuracy compared to digital thermometer	±2 °C	±1.5 °C
TC6	Continuous operation for 3 hours	No crashes or overheating	Stable
TC7	LCD readability and I2C communication	No flicker or errors	Clear & stable

6.2 Test Procedure

1. Sensor Calibration:
   - The LM35 was tested at known temperature points (ice water, room temp, hand warmth).
   - ADC readings were recorded and converted into temperature using the known formula:

Temperature = (analogRead * 3.3 / 4095) * 100

2. LCD Testing:
    - Verified correct initialization of the 16x2 LCD.
    - Ensured proper cursor positioning, display refresh, and contrast.
3. Blynk IoT Testing:
    - Configured virtual pin V0 for real-time display.
    - Tested delay between sensor reading and mobile app update.
    - Checked network reliability and token re-authentication on reboot.
4. Stability & Durability Testing:
    - Let the system run continuously for 3+ hours.
    - Monitored for overheating, inconsistent readings, or system crash.
5. Power Supply Testing:
    - Verified that ESP32 and peripherals work stably with a 5V USB supply.
    - Observed behavior when switching to battery (optional use case).

6.3 Performance Outcome

The performance evaluation confirmed that the system:

- Operates reliably with low power consumption, making it ideal for battery-powered IoT nodes.
- Maintains high accuracy within ±1.5 °C when compared with commercial digital thermometers.
- Displays real-time temperature readings on the LCD with excellent readability.
- Transmits data to the Blynk app with minimal latency (~1.5 seconds).
- Demonstrates robust ADC conversion and signal stability, even after extended run times.
- Provides a scalable and modular design that can support future upgrades like cloud logging, multiple sensor inputs, or wireless alert mechanisms.

7. My Learnings

This internship was an enriching experience that significantly enhanced both my theoretical understanding and practical skills in the field of IoT and embedded systems. Over the span of four weeks, I had the opportunity to work on a real-world project involving sensor integration, microcontroller programming, circuit design, and IoT interfacing.

Here are the key areas where I gained substantial learning:

7.1 Technical Knowledge and Skills

- Embedded Programming with ESP32:
  I learned to program the ESP32 microcontroller using the Arduino IDE, which included understanding GPIO control, analog-to-digital conversion (ADC), and real-time data processing.

- Sensor Interfacing and Calibration:
  I successfully interfaced the LM35 analog temperature sensor and understood how to convert raw analog voltage into accurate temperature readings. This helped me understand sensor behavior, calibration techniques, and precision handling.

- LCD Communication via I2C:
  I learned to use the I2C protocol for efficient communication between the ESP32 and a 16x2 LCD display. This involved understanding device addressing, register control, and display formatting.

- IoT Integration using Blynk:
  One of the most exciting aspects was integrating the project with the Blynk platform. I learned how to configure the mobile dashboard, use virtual pins, and transmit real-time data via Wi-Fi from ESP32 to the cloud.

- Circuit Design and Debugging:
  Through hands-on circuit assembly and simulations, I gained a clearer understanding of power supply design, noise filtering, and component selection. I also practiced debugging using serial monitors and testing tools.

7.2 Soft Skills and Project Mindset

- Documentation and Reporting:
  I improved my technical writing skills by documenting project progress weekly and compiling structured reports with clear explanations and code integration.

- Problem-Solving:
  I developed a systematic approach to identifying hardware or software issues and resolving them using datasheets, forums, and mentor feedback.

- Time Management:
  Balancing project development, debugging, learning new concepts, and report submission within fixed weekly timelines helped me become more disciplined and productive.

- Collaboration:
  Even though much of the work was individual, interacting with mentors, peers, and online communities taught me the importance of communication and teamwork in a technical environment.

7.3 Industry Exposure

This internship bridged the gap between classroom knowledge and industrial application. It introduced me to real-world constraints like power efficiency, component cost, accuracy, and stability—important considerations for product design and deployment. I also learned how platforms like UCT Insight and Blynk enable quick prototyping and smart system integration in industry scenarios.

In conclusion, this internship gave me a well-rounded exposure to the embedded system development cycle—from design and implementation to testing and documentation. It has boosted my confidence to take on more complex IoT projects and pursue a career in embedded systems and smart automation.

Here is a detailed and professional version of Section 8: Future Work Scope for your internship report on the *IoT-Based Temperature Monitoring System*:

8. Future Work Scope

While the current implementation of the IoT-Based Temperature Monitoring System effectively achieves its core functionality—real-time temperature sensing, local display, and remote monitoring via Blynk—there are several enhancements and extensions that could be undertaken to transform this basic prototype into a more robust, feature-rich, and industry-ready solution.

Outlined below are some possible directions for future development:

8.1 Data Logging and Analysis

- Cloud Integration for Historical Data:
  Incorporating platforms like Firebase, ThingSpeak, or AWS IoT to store historical temperature data would allow users to view trends, generate reports, and perform long-term monitoring.
- SD Card Storage:
  Adding an SD card module for local data logging in environments without internet connectivity would ensure continuous record-keeping and later analysis.

8.2 Alerting and Threshold Management

- Email/SMS Notifications:
  The system could be extended to send automatic alerts via email or SMS using services like IFTTT or Twilio when temperature crosses user-defined thresholds.
- Buzzer or LED Indications:
  Hardware-based alerts (like buzzers or warning LEDs) could be added to notify local users of abnormal temperature readings instantly.

8.3 Multi-Sensor Support

- Multiple LM35s or Digital Sensors:
  Support for additional sensors (e.g., DHT22 for temp + humidity, or DS18B20 for digital temperature sensing) could be implemented for multi-point monitoring within the same setup.
- Sensor Selection Menu:
  A UI could be added on the Blynk app allowing users to switch between different sensors or zones.

8.4 Power Optimization

- Battery Backup and Sleep Modes:
  Incorporating Li-ion battery modules and optimizing the code to use ESP32's deep sleep mode would enhance portability and power efficiency.
- Solar Power Integration:
  In remote outdoor environments, a solar panel could be integrated for sustainable, off-grid operation.

8.5 Enhanced User Interface

- Advanced Blynk Dashboard:
  Add more widgets like graphs, sliders (to set thresholds), and tabs for multi-device management.
- Local Touchscreen Display:
  Instead of a 16x2 LCD, a TFT touchscreen could be used to display data with graphical representation and offer user interaction directly on the device.

8.6 Enclosure and Deployment

- Custom Enclosure Design:
  3D-printed or commercially available weatherproof enclosures would allow the device to be deployed in outdoor or industrial settings.
- Wall/Panel Mount Options:
  Designing mounting options would make the system more user-friendly and professional for real deployments.

8.7 Scalability and Industry Readiness

- Integration with Industrial Protocols:
  Support for Modbus, MQTT (already available), or OPC-UA could make the system compatible with industrial SCADA and PLC environments.
- Integration with UCT Insight Platform:
  As a next step, the system could be connected to UCT's industrial IoT dashboard to demonstrate real-time analytics and multi-device aggregation.

These enhancements will not only make the system more comprehensive and reliable but also pave the way for transforming it into a market-ready product suitable for various real-world applications in agriculture, food storage, pharmaceuticals, HVAC systems, and industrial safety.