**BONAFIDE CERTIFICATE**

Certified that this project report title **“SMART E-UNIFORM FOR SOLDIERS USING IOT”** is the bonafide work of **PRAKASH P (Registration Number: 311423622035)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**Supervisor Head of the Department**

**(Ms. MONISHA PJ) (Dr. N. Naga Saranya)**

Submitted to Project and Viva Examination held on

**Internal Examiner External Examiner**

**ABSTRACT**

Soldiers, who courageously defend our nation, often face extreme and unpredictable weather conditions during their service. To safeguard their health and enhance their operational efficiency, a **climate-adjustable Smart E-Uniform** has been developed. This IoT-based wearable system is designed to provide **real-time monitoring and dynamic climate control,** ensuring the safety and well-being of soldiers in the field. The Smart E-Uniform operates in two modes **summer and winter** allowing soldiers to regulate their body temperature according to the prevailing environmental conditions. The uniform integrates several sensors: an **ultrasonic sensor** for proximity detection, a **gas sensor** for hazardous gas monitoring, and a **DHT11 temperature and humidity sensor** to track both environmental and physiological parameters. To manage extreme temperatures, the uniform includes a **Peltier cooling panel** and a **thermal fan**, which activate automatically when threshold values are exceeded. Sensor data is continuously transmitted via Wi-Fi to the **Thing Speak cloud platform**, enabling **real-time monitoring** by higher officials at the **base station**. This allows them to track the health and environmental status of each individual soldier. In the event of critical conditions such as toxic gas presence, extreme temperature, or unsafe proximity **alerts are immediately sent via MQTT, email, and SMS**, ensuring timely response and action. By combining **environmental sensing, automated thermal regulation**, and **cloud-based communication**, the Smart E-Uniform offers an intelligent, responsive solution for soldier safety. It significantly improves survivability in hostile environments and supports strategic decision-making in defence operations.

**ACKNOWLEDGEMENT**

#### I am indebted to few people who have contributed a lot towards this project.

#### I am extremely thankful to the Managing Trustee **Tmt. Gomathi Radhakrishnan**, and our beloved Managing Director **Mrs. Jayanthi Prabhakaran** for providing the full encouragement and all the necessary facilities to carry out this project.

#### I would like to thank our beloved principal **Dr.** for providing me with all necessary facilities to complete this project.

#### I am thankful to **Dr. N. Naga Saranya,** Head of the Department, who helped in choosing a good project and for her constant encouragement to complete this project successfully.

#### I would like to thank my internal guide **Ms. Monisha PJ**, who gave me full support and assisted me in all critical situations.

I seem it is a great privilege to express my sincere thanks to **Pantech E-learning solutions T Nagar,** who arranged me such a good atmosphere throughout the project.

I would like to thank my project leader **Mr. A. Sathish, Pantech E-learning solutions T Nagar,** who made this project lively and led me in the right way to complete the project successfully.

I would like to thank many other individuals at “**Meenakshi College of Engineering**” who have contributed greatly towards the success of my project.

|  |  |  |
| --- | --- | --- |
|  | **TABLE OF CONTENTS** |  |
| **CHAPTER**  **NO.** | **TITLE** | **PAGE**  **NO.** |
|  | **ABSTRACT** | **ii** |
|  | **LIST OF TABLES** | **Vii** |
|  | **LIST OF FIGURES** | **Viii** |
|  | **LIST OF ABBREVIATIONS** | **X** |
| **1** | **INTRODUCTION** | **1** |
|  | 1.1 ABOUT THE COMPANY | 1 |
|  | 1.2 ABOUT THE PROJECT | 2 |
|  | 1.3 PROBLEM DEFININTION | 3 |
| **2** | **LITERATURE SURVEY** | **4** |
|  | 2.1 REVIEW OF LITERATURE SURVEY | 4 |
| **3** | **SYSTEM ANALYSIS** | **7** |
|  | 3.1 EXISTING SYSTEM | 7 |
|  | 3.2 PROPOSED SYSTEM | 8 |
|  | 3.3 MODULES | 10 |
| **4** | **DEVELOPMENT ENVIRONMENT** | **11** |
|  | 4.1 HARDWARE REQUIREMENT  4.1.1 MICROCONTROLLLER  4.2 SOFTWARE REQUIREMENT  4.2.1 AURDINO IDE | 11  11  13  13 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **CHCHAPTER**  **NO NO.** | **TITLE** | **PAGE**  **NO.** | | **TITLE** | **PAGE**  **NO.** |

|  |  |  |
| --- | --- | --- |
| **5** | **SYSTEM DESIGN** | **17** |
|  | 5.1 SYSTEM ARCHITECTURE  5.2 USE CASE DIAGRAM  5.3 DATA FLOW DIAGRAM 5.4 WORK FLOW DIAGRAM | 18  18  19  21 |
| **6** | **SYSTEM IMPLEMENTATION**  6.1 MODULE DESCRIPTION  6.1.1 ULTRASONIC SENSOR  6.1.2 GAS SENSOR  6.1.3 TEMPERATURE SENSOR  6.1.4 RELAY  6.1.5 PELTIER PANEL  6.1.6 THERMAL FAN  6.1.7 LCD  6.1.8 EMERGENCY ALERTS SYSTEM  6.1.9 REAL-TIMEMONITORING VIATHINGSPEAK  6.2 PROJECT CODE IMPLEMENTATION  6.2.1 INTRODUCTION  6.2.2 COMPLETE SOURCE CODE | 22  22  22    24  26  28  32  33  35  37  46  51  51  51 |
|  |  |  |

|  |  |
| --- | --- |
|  |  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **NO.** | **TITLE** | **PAGE**  **NO.** |
| **7** | **TESTING** | **62** |
|  | 7.1 TESTING METHODOLOGY | 62 |
| **8** | **PERFORMANCE AND LIMITATION** | **66** |
|  | 8.1 PERFORMANCE | 66 |
|  | 8.2 LIMITATION | 66 |
|  | 8.3 CONCLUSION AND FUTURE  ENHANCEMENT | 67 |
| **9** | **APPENDICES** | **68** |
|  | 9.1 SAMPLE SCREENS | 68 |
|  | **REFERENCES** | **78** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | **LIST OF TABLES** |  |
| **TABLE**  **NO.** | **TITLE** | **PAGE**  **NO.** |
| 6.7 | LCD Pin Configuration | 36 |
| 7.1 | Test Data | 63 |

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **FIGURE**  **NO.** | **TITLE** | **PAGE**  **NO.** |
| 4.1  5.1 | Node MCU ESP 8266  System Architecture | 12  18 |
| 5.2 | Use Case Diagram | 18 |
| 5.3 | Data Flow Diagram | 19 |
| 5.4  6.1  6.2  6.2.1  6.3  6.3.1  6.4  6.4.1  6.4.2  6.4.3  6.5  6.6  6.7  6.7.1  6.8 | Work Flow Diagram  Ultrasonic Sensor  Gas Sensor  Working of Gas Sensor  Temperature Sensor  Humidity Wave Representation  Relay  Movable Relay  Non-Movable Relay  Relay Working  Peltier Panel  Therma Fan  LCD  LCD Pin Configuration  MY MQTT APP | 21  23  24  25  26  27  28  29  29  30  33  34  35  36  38 |
| **FIGURE  NO** | **TITLE** | **PAGE**  **NO** |
| 6.8.1  6.8.2  6.8.3  6.8.4  6.9  6.9.1  6.9.2  6.9.3 | MQTT Layer  MQTT Connection  E-mail Alert  SMS Alert  Thing speak Account  Thing speak Channel  Field Creation  Data in Graph | 40  40  43  46  47  48  49  50 |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| **LIST OF ABBREVIATIONS** | |
| LCD  HVAC | Liquid Crystal Display  Heating Ventilation and Air Conditioning |
| MQTT | Message Queueing Telemetry Transport |
| SMS | Short Message Service |
| SMTP  GND  VCC | Simple Mail Transfer Protocol  Ground  Voltage Common Collector |
| IDE | Integrated Development Environment |
| DFD | Data Flow Diagram |
| SDA  SCL  WIFI  GPIO  SDIO  UART | Serial Data  Serial Clock  Wireless Fidelity  General Purpose Input/Output Interface  Secure Digital Input/Output Interface  Universal Asynchronous Receiver Transmitter |
|  |  |

**CHAPTER 1**

**INTRODUCTION**

Soldiers, the backbone of our nation's defence, endure various challenges, especially adverse weather conditions, while safeguarding our country. The Solar E-Uniform for Soldiers project designed to harvest solar energy to power communication and health monitoring devices, has shown promise in increasing soldiers' endurance and safety in isolated environments. However, extreme battlefield conditions demand not just energy autonomy but intelligent sensing and active protection mechanisms needed. Environmental hazards such as toxic gases, intense heat, and unforeseen obstacles remain serious threats. Moreover, traditional solar-powered uniforms lacked adaptive cooling technologies crucial for extreme climates, it suffered from several critical limitations: dependency on sunlight, reduced efficiency in cloudy or indoor conditions, added weight due to solar panels, and lack of active thermal control mechanisms. To overcome these shortcomings, this project introduces a Smart E-Uniform for Soldiers enhanced with real-time environmental sensing, thermal management, and IoT-based visualization with removal of solar panel replaced by reusable batteries. By incorporating ultrasonic sensors for obstacle detection, gas sensors for chemical threat identification, temperature sensor for climate control, and Peltier cooling modules managed by IoT control through “Thing Speak”, this uniform serves not only as a shield but as a smart assistant for soldiers on the field.

* 1. **ABOUT THE COMPANY**

“Pantech e Learning” is a forward-thinking organization committed to transforming education and technology through innovation and excellence. With a strong foundation in Artificial Intelligence, Embedded Systems, IoT, and emerging technologies, Pantech e Learning bridges the gap between academic knowledge and industry needs by delivering high-quality training, product development, and technical services across the globe.

Headquartered in Chennai, India, Pantech e Learning is recognized for its dedication to quality, customer-centric approach, and innovative delivery models. The company empowers students, professionals, and institutions by providing skill-based learning, real-time project experiences, and industry-relevant exposure. What sets us apart is not just the expertise of our team but the robust processes we follow to ensure impactful outcomes.

At Pantech, innovation drives our mission. We are constantly challenged to adapt, evolve, and deliver value by doing more with less, ensuring we remain at the forefront of the EdTech revolution. Our strength lies in our structured methodology deeply understanding the needs of learners and clients, and designing tailored solutions that align with their goals.

* 1. **ABOUT OF THE PROJECT**

The **Smart E-Uniform for Soldiers Using IoT** is an innovative wearable technology designed to enhance the safety, monitoring, and response capabilities of soldiers in the field. This smart uniform integrates multiple sensors and IoT-based communication systems to continuously monitor vital environmental conditions and provide real-time alerts in emergency situations.

The uniform is equipped with the following key components:

* **Temperature Sensor (DHT11):** Monitors the surrounding temperature to detect extreme heat or cold conditions. Based on thresholds, it activates a **thermal fan** or **peltier panel** to regulate body temperature.
* **Gas Sensor:** Detects the presence of harmful or toxic gases in the surrounding environment, triggering immediate alerts if dangerous levels are identified.
* **Ultrasonic Sensor:** Measures the distance between the soldier and the base station or predefined boundary to ensure safety within operational zones.
* **Thermal Fan & Peltier Panel:** These components are used for thermal regulation. The thermal fan cools the body during high temperatures, while the peltier panel provides heating during cold conditions. Both are controlled automatically and never operate simultaneously
* **LCD Display & Serial Monitor:** Continuously display real-time sensor data for soldier awareness and debugging.
* **Data Visualization (Thing Speak):** All sensor data is uploaded to Thing Speak for remote monitoring and analysis using graphical dashboards.
* **Alert System:**
  + **MQTT Protocol:** Sends instant alerts to a cloud MQTT broker when abnormal conditions are detected.
  + **Email Alerts:** Sends detailed email notifications containing sensor data to preconfigured addresses.
  + **SMS Alerts (via Twilio):** Sends emergency SMS alerts directly to the soldier’s command centre or assigned phone numbers when critical thresholds are crossed.

The system is designed to respond within 15 seconds, ensuring timely alerts and efficient data updates. The alert logic is optimized to avoid redundancy by sending a limited number of alerts per event, ensuring relevant and concise communication. This smart uniform aims to provide enhanced safety, environmental awareness, and quick response for soldiers operating in diverse and challenging terrains.

* 1. **PROBLEM DEFINITION**

Soldiers deployed in extreme environments often face life-threatening challenges such as exposure to high or low temperatures, harmful gases, and isolation from command centre’s due to lack of real-time monitoring. To address these challenges, there is a need for a smart solution that combines wearable sensors, real-time data monitoring, and automatic alert mechanisms to ensure the safety, efficiency, and connectivity of soldiers during missions. The **Smart E-Uniform** addresses these problems using IoT technology, enabling real-time data collection, visualization, and rapid emergency notifications to enhance soldier safety and situational awareness.

**CHAPTER 2**

**LITERATURE SURVEY**

* 1. **REVIEW OF LITERATURE SURVEY**

Vignesh A., C. Bhuvaneswari, Shiva M., W. Abitha Mamala, M. Pushpavalli, and M. Kavitha (2023) proposed an innovative solution titled **"Intelligent Solar-Based Climate Adjustable E-Uniform for Soldiers"** aimed at enhancing soldier comfort, safety, and environmental awareness during military operations. This work focuses on combining **renewable energy, wearable technology**, and **intelligent control systems** to create a smart uniform capable of adapting to varying climate conditions in the field. The system integrates **solar panels** to power the embedded electronic components, making it suitable for long-term use in remote areas without external power access. They emphasized the use of a **microcontroller-based control unit** that receives environmental input and activates the appropriate response mechanism, such as switching on the Peltier panel in cold conditions or the fan in high temperatures. The control logic includes safety constraints ensuring that **both heating and cooling systems do not operate simultaneously**, preventing power wastage and system conflict. While the work primarily focused on **on-body climate regulation**, it also suggested potential extensions involving **wireless communication, IoT-based data logging,** and **remote alerting**, making it compatible with modern smart uniform concepts.[1]

Ms. Sumalata Kadir, I. Spandana, J. Shravya, and K. Sravani proposed a smart solar-powered e-uniform for soldiers, integrating climate control and environmental monitoring systems. Their study focused on enhancing safety and comfort for personnel operating under extreme conditions by embedding IoT-enabled components into wearable technology. The project incorporates various environmental sensors such as the DHT11 for temperature and humidity, the MQ-2 gas sensor for detecting harmful gases, and an ultrasonic sensor for measuring

proximity or distance. The uniform is powered using integrated solar panels, providing a sustainable energy source that supports on-field usage without dependence on external power. A core contribution of the work is the automated climate control mechanism, which includes thermal fans for cooling and Peltier modules for heating. These are activated based on real-time sensor data to maintain the soldier’s body temperature within a safe range. Notably, the system ensures that both modules never operate simultaneously, reducing power consumption and avoiding conflicting actions. The authors also implemented a multi-layered alert system using MQTT for lightweight data transmission, email alerts, and SMS alerts via Twilio, enabling remote monitoring and timely responses in emergency scenarios. Data is uploaded to Thing Speak, allowing for cloud-based visualization and analysis. [2]

Cho, Y., Kim, H., and Lee, J. (2023) conducted an in-depth study on the use of **Phase Change Materials (PCMs)** for **thermal regulation in military clothing,** with the objective of enhancing thermal comfort and protection for soldiers operating in extreme temperature environments. Their work explores the integration of PCMs into textile systems to passively regulate body temperature without requiring external power sources.

PCMs function by absorbing, storing, and releasing latent heat during phase transitions (typically from solid to liquid and vice versa). The authors evaluated several types of PCMs with varying melting points to determine their suitability for military scenarios, particularly in arid or cold climates where body heat management is critical.[3]

Patel, V., Sharma, B., and Khan, A. (2022) presented a study titled **"Gas Sensor-Based Environmental Monitoring for Soldier Safety,"** focusing on the integration of gas detection technologies into wearable systems for military personnel. The aim of the study was to enhance situational awareness and prevent exposure to harmful or toxic gases during field operations. The authors explored the application of **MQ-series gas sensors**, particularly the **MQ-2** and **MQ-135,** for real-time detection of flammable gases, smoke, and air pollutants such as ammonia and carbon monoxide. These sensor

were chosen due to their **low cost, fast response time, and compatibility with microcontroller-based platforms** such as Arduino and ESP8266.[4]

Kim, D., Park, S., and Choi, M. (2022), in their article published in the Journal of Sensor Technology*,* investigated the use of **infrared (IR) sensors** in **low-visibility environments**, such as fog, smoke, and night-time conditions. Their study aimed to improve environmental awareness and navigation support for personnel operating in visually obstructed conditions, including soldiers, first responders, and search-and-rescue teams. They focused on **passive and active IR sensors**, evaluating their performance in detecting human presence, motion, and thermal signatures under various obstructed visibility scenarios. Their system design utilized **pyroelectric IR sensors** and **IR imaging modules**, integrated with microcontrollers for data acquisition and analysis. In some configurations, the sensors were paired with **signal processing algorithms** to enhance accuracy and reduce false positives in noisy environments. The experimental analysis showed that IR sensors were particularly effective in detecting body heat through smoke and low light, making them suitable for **combat scenarios, post-disaster monitoring,** and **urban warfare operations** where visibility is severely limited. The authors also discussed the potential for **combining IR sensors with ultrasonic or gas sensors** for comprehensive environmental monitoring aligning with the approach used in smart uniforms for soldiers. However, they noted challenges such as **limited range, ambient temperature influence**, and the need for proper sensor positioning on wearable systems.[5]

**CHAPTER 3**

**SYSTEM ANALYSIS**

After carefully analyzing the task requirements, the next critical step is to examine the problem in its operational context. This involves two main activities: first, studying the **existing system or environment** in which soldiers operate, and second, understanding the **requirements and constraints of the proposed Smart E-Uniform system.** Soldiers in extreme environments often rely solely on manual observation or external support, leading to **delayed responses** and **increased physical risk**. Understanding the proposed system requires both **technical knowledge and creative problem-solving,** as it introduces a hybrid model combining **IoT, renewable energy, and wearable electronics**.

* 1. **EXISTING SYSTEM**

The Solar-Based E-Uniform project is an early prototype designed to help soldiers monitor their body or environmental temperature using IoT technology. In this system, the uniform is equipped with: Arduino Uno R3 microcontroller (as the central control unit), temperature sensor (commonly DHT11 or LM35) and solar panel (to supply power to the Arduino and sensor). It works on the solar panel charges a battery or powers the system directly then temperature sensor collects real-time data about the soldier’s body or surrounding temperature. At last Arduino Uno reads this data and sends it to LCD display received data.

**Working Principle:**

* The solar panel harvests sunlight to generate power, which is used to run the Arduino and sensors or to charge an onboard battery for backup power.
* The temperature sensor continuously collects temperature data from the environment or the soldier’s body.
* The Arduino Uno reads this data, processes it, and transmits it to the LCD display.

**DISADVANTAGES OF EXISTING SYSTEM:**

* Arduino UNO R3 version microcontroller doesn’t have built-in Wi-Fi module for send automatic alerts.
* Solar panels are ineffective during night operations indoors, or in low-light/bad weather conditions.
* It does not provide any active cooling or heating; it only monitors temperature without improving soldier comfort.
* Solar panels can add weight and stiffness to the uniform, reducing soldier mobility and comfort

**3.2 PROPOSED SYSTEM**

The proposed system is a **smart embedded wearable module** designed for soldiers operating in extreme and hazardous environments. It integrates **environmental monitoring, personal safety control,** and **automated thermal regulation** using a combination of sensors and actuators powered by **ESP8266** and controlled remotely via **IoT connectivity**.

**1. Toxic Gas Detection**

* The Gas sensor continuously monitors for the presence of harmful gases (e.g., CO, methane, LPG).
* When toxic gas is detected beyond a safe threshold, the system:
  + Sends an alert to a remote server via Wi-Fi using ESP8266.

**2. Proximity & Obstacle Monitoring**

* An Ultrasonicsensor is embedded to detect nearby objects or moving threats.
* Useful for:
  + Detecting environmental obstructions or potential hazards.
  + Enhancing safety during operations in low visibility or confined areas.

**3. Dynamic Body Temperature Regulation**

* A Peltiermodule is used to heat or cool the soldier’s uniform based on surrounding temperature conditions.
* Controlled by the ESP8266 via a relaymodule.
* Automatically activates when ambient conditions exceed comfort thresholds.

**4. Smart Cooling with Fan Control**

* A DC fan assists in cooling during overheating scenarios, activated based on:
  + High ambient temperature.
  + High body temperature.

**5. IoT Connectivity (ESP8266)**

* Sends real-time data (temperature, gas levels, obstacle distance) to a clouddashboard.
* Enables remotemonitoring by commanders or control units.
* Can be integrated with platforms like ThingSpeak.
* **MQTT Protocol:** Enables lightweight, fast communication for pushing alerts to a cloud server or subscribed devices.

**6.Emergency Alert System:**

* **Email Alerts:** Triggered when gas levels are high or temperature crosses critical thresholds.
* **SMS Alerts via Twilio:** Sent to authorized numbers during emergency conditions, providing sensor readings and location-based alerts.
* **MQTT Alerts:** Provide instant status updates to connected systems for real-time response.

**ADVANTAGE OF EXISTING SYSTEM:**

* It uses Node MCU ESP 8266 Microcontroller it has built-in Wi-Fi with high processing power than Arduino.
* Allow commanders to monitor soldier’s environment and health via Think Speak
* Components like fans and Peltier modules are activities only when needed, saving energy and improve battery life.
* Combines health monitoring, environmental sensing and smart control in one compact system.
* Alerts and automation reduce manual checks and help soldiers stay on focused on their mission.
* Automated thermal regulation increases soldier comfort.
  1. **MODULES**

1.Ultrasonic sensor

2.Gas sensor

3.Temperature Sensor

4.Relay

5.Peltier Panel

6.Thermal Fan

7.LCD

8.Emergency Alert System

9.Real-Time monitoring using Thing speak

**CHAPTER 4**

**DEVELOPMENT ENVIRONMENT**

**4.1 HARDWARE REQUIREMENT**

* Microcontroller
* Temperature Sensor
* Ultra sonic sensor
* Gas Sensor
* Peltier panel
* Thermal Fan
* Relay
* LCD

**4.1.1 Microcontroller**

The **ESP8266 NodeMCU** is used as the main microcontroller in this project. It is a low-cost Wi-Fi-enabled board that supports IoT applications. The ESP8266 is responsible for interfacing with all sensors and actuators, processing the sensor data, and sending alerts through various communication protocols like MQTT, Email (SMTP), and SMS (via Twilio API). It also connects to the internet to transmit data to platforms like ThingSpeak for real-time monitoring.

**Key features:**

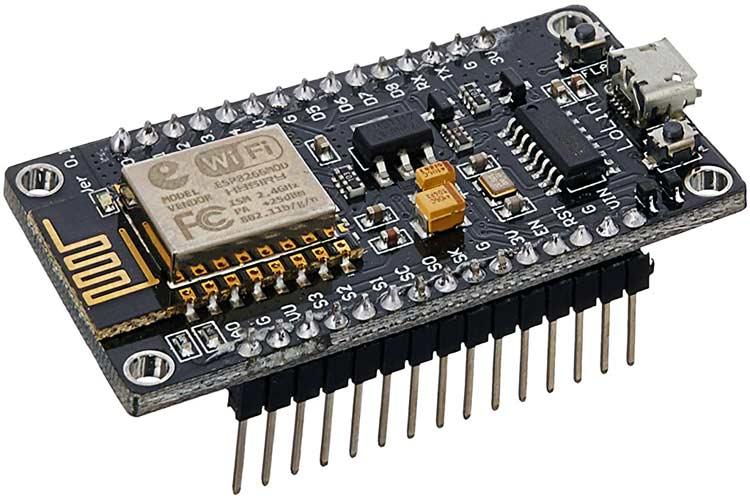
* Built-in Wi-Fi capability for wireless communication
* Multiple digital input/output pins
* Compact and power-efficient
* Supports Arduino IDE for programming
* Suitable for real-time sensor data collection and processing in IoT-based application.

**General Purpose Input/Output Interface (GPIO)**

* ESP8266EX has 17 GPIO pins which can be assigned to various functions by programming the appropriate registers. Each GPIO can be configured with internal pull-up or pull-down, or set to high impedance, and when configured as an input, the data are stored in software registers; the input can also be set to edge-trigger or level trigger CPU interrupts.

**Secure Digital Input/Output Interface (SDIO)**

* ESP8266EX has one Slave SDIO, the definitions of which are described below. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

****

**Figure** 4.1 Node MCU ESP8266

**Universal Asynchronous Receiver Transmitter (UART)**

Data transfers to/from UART interfaces can be implemented via hardware. The data transmission speed via UART interfaces reaches 115200 x 40 (4.5 Mbps). UART0 can be used for communication. It supports fluid control. Since UART1 features only data transmit signal (Tx), it is usually used for printing log.

**4.2 SOFTWARE REQUIREMENT**

* Arduino IDE
* Thing speak (Data visualization)

**4.2.1 Arduino IDE overview**

Arduino IDE is an open-source platform used to write and upload code to microcontroller boards such as the ESP8266. It provides a user-friendly interface with support for C/C++ programming, serial monitoring, and integration with external libraries. In this project, Arduino IDE was used to program the ESP8266 NodeMCU for sensor data collection, Wi-Fi connectivity, and triggering emergency alerts via MQTT, Email, and Twilio SMS.

**Features of Arduino IDE:**

**a. Writing Sketches:**

|  |  |
| --- | --- |
| https://www.arduino.cc/en/uploads/Guide/play.png | Verify  Checks your code for errors compiling it. |

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension. ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

|  |  |
| --- | --- |
| https://www.arduino.cc/en/uploads/Guide/export.png | Upload  Compiles your code and uploads it to the configured board. See uploading below for details.  Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer" |
| https://www.arduino.cc/en/uploads/Guide/new.png | New  Creates a new sketch. |
|  |  |

#### **b. File Menu**

Used to create new sketches, open existing ones, save, and manage sketches for various modules like gas detection, temperature alert, etc.

#### **c. Edit Menu**

Allows basic editing operations like copy, paste, and find/replace — useful while managing long code for multiple sensors.

#### **d. Sketch Menu**

Provides options to verify (compile), upload code, and manage libraries. This used frequently for compiling code before flashing it to the ESP8266 board.

#### **e. Tools Menu**

Used to select the board (e.g., NodeMCU 1.0), port, and access other utilities like the Serial Monitor and Board Manager.

**f. Help Menu**

#### Offers documentation, links to Arduino resources, and troubleshooting support during development.

#### **g. Sketchbook**

A folder where Arduino sketches are stored. All project-related sketches were saved and accessed from the sketchbook.

#### **h. Tabs, Multiple Files, and Compilation**

Arduino allows splitting the program into multiple files/tabs, helping manage complex code like sensor logic, alert logic, and display logic separately.

#### **i. Uploading**

Uploads compiled code to the ESP8266 via USB. This was used after every code change or feature addition.

#### **j. Libraries**

Libraries like DHT.h, ESP8266WiFi.h, WiFiClientSecure.h, etc., were included to interact with sensors and perform secure communication.

#### **k. Third-Party Hardware**

Using the **Boards Manager**, support for ESP8266 was added. This enabled programming the NodeMCU board directly from Arduino IDE.

#### **l. Serial Monitor**

#### A built-in tool used to display sensor data, debug values, and verify system behavior in real-time.

#### **m. Preferences**

Settings related to code display, compile/upload behaviour, and external editor support were adjusted here.

#### **n. Language Support**

The IDE supports multiple languages. English was used throughout development.

#### **o. Boards**

From the **Boards Manager**, the correct board (NodeMCU 1.0 (ESP-12E Module)) was selected for uploading code to the ESP8266.

**CHAPTER 5**

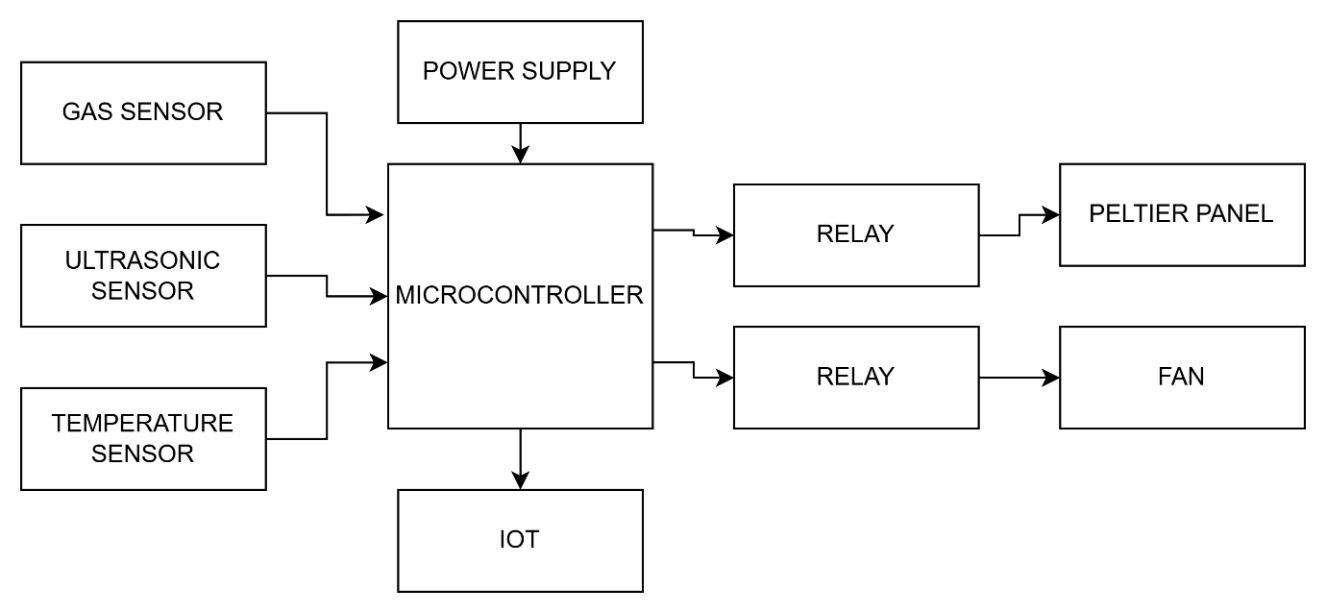
**SYSTEM DESIGN**

Software design sits at the core of the development process in IoT-based systems and plays a vital role in translating requirements into an operational framework. For this project, **Smart E-Uniform for Soldiers using IoT**, software design acts as a blueprint for implementing sensor integration, data processing, communication, and emergency alert mechanisms.

The primary goal of the design phase is to generate a structured and modular software representation that guides development and ensures maintainability, scalability, and reliability. In IoT applications where hardware and software must interact seamlessly, a well-planned software design becomes crucial to achieving the system's functional goals.

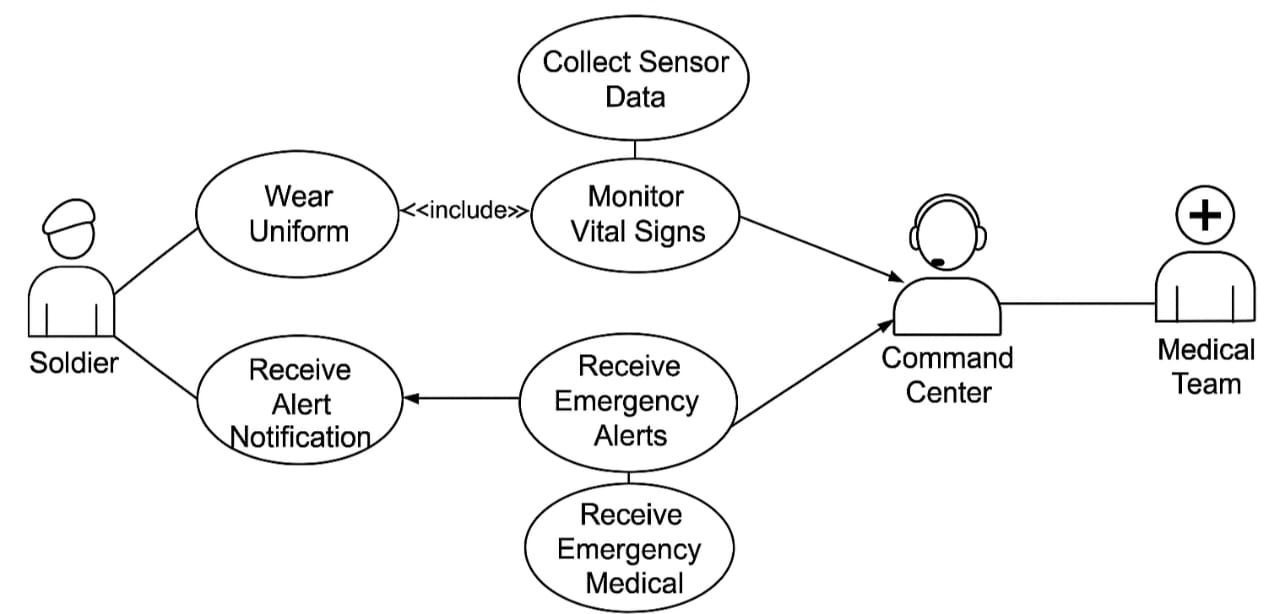
Once system requirements (such as temperature alert thresholds, gas detection, range limitations, and cloud reporting) have been clearly identified and analyzed, the software design bridges the gap between these requirements and code implementation. This project follows the typical design-code-test cycle, where software design lays the foundation for quality, accuracy, and performance.

**5.1 SYSTEM ARCHITECTURE**

****

**Figure 5.1** System architecture

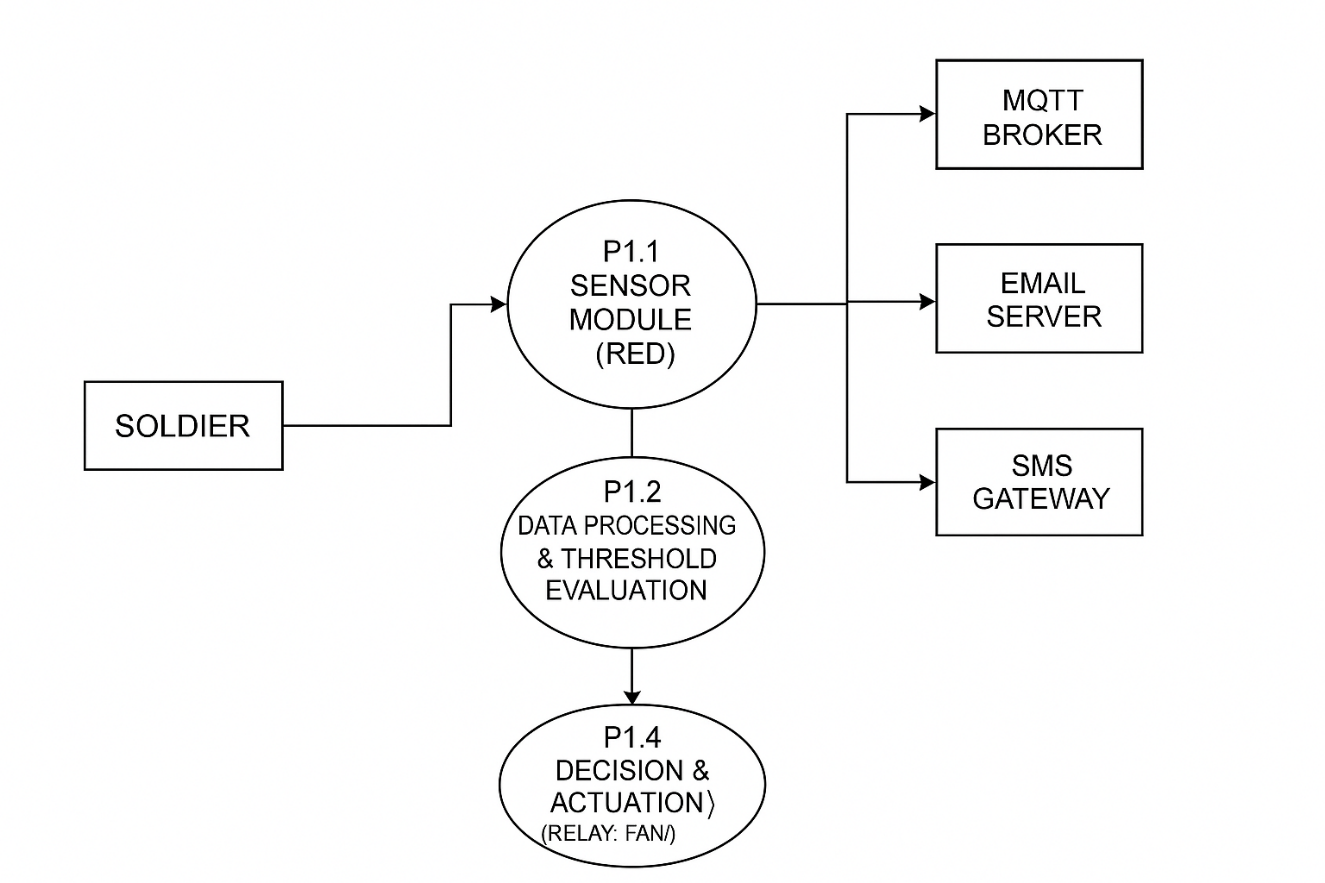
**5.2 USE CASE DIAGRAM**

****

**Figure 5.2** Use case diagram

**5.3 DATA FLOW DIAGRAM**

A Data Flow Diagram (DFD) is a graphical tool used to describe and analyze the movement of data through the system. It is a graphical representation of the “flow” of data through a computer system or a data or it looks at how data flows through a system. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. The development of DFD is done at several levels. The flow diagram describes the boxes that describe computations, decisions, interactions & loops. It is important to keep in mind that the flow diagrams are not flowcharts and should not include control elements.



**Figure 5.3** Data flow diagram

**DFD Characteristics**

* Emphasizes data movement and transformation, not hardware flow.
* Uses process identifiers (P1.1, P1.2, etc.) and arrows to show directional flow.
* Avoids physical components; focuses on how information is processed within the system.
* Takes automated action (e.g., turning on a fan) if data exceeds thresholds.
* Ensures immediate physical response to protect the soldier.

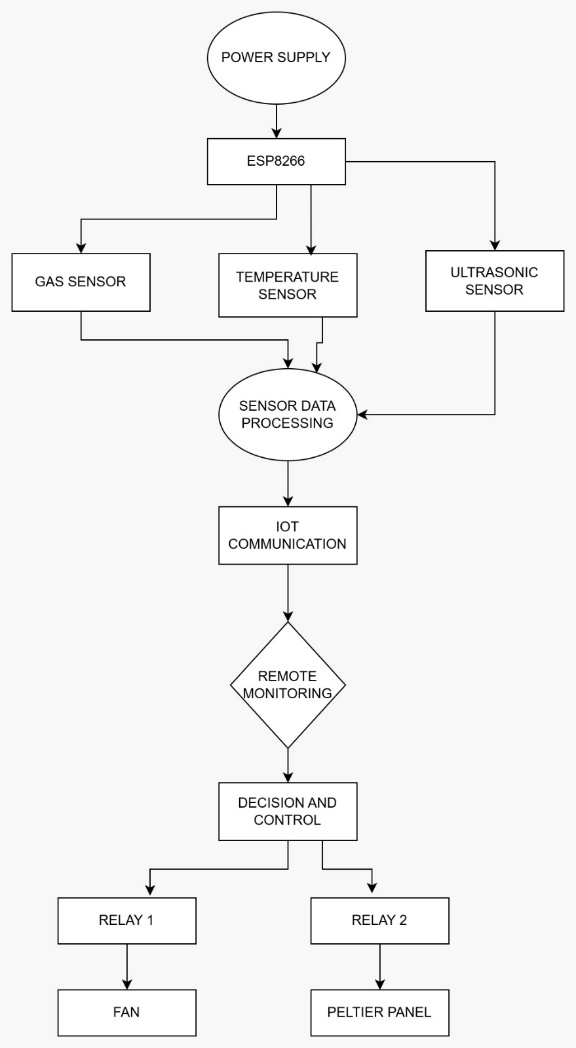
**Type**

• Logical data flow diagram

**Characteristics of a Logical DFD:**

* Focuses on what the system does, rather than how it's physically implemented.
* Emphasizes the flow of data and processing logic, like decision-making, evaluations, and data distribution.
* Includes abstract processes like:
* Sensor data collection
* Threshold evaluation
* Notifications via SMS and email
* Decision-making for actuation

**5.4 WORK FLOW DIAGRAM**

****

**Figure 5.4** Work flow diagram

**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

**6.1 MODULE DESCRIPTION**

The Smart E-Uniform system is divided into multiple modules, each responsible for specific tasks. This modular architecture improves system clarity, simplifies testing, and enables future upgrades.

**6.1.1 Ultrasonic Sensor**

Ultrasonic sensors are electronic devices that determine a target's distance. They work by emitting ultrasonic sound waves and converting those waves into electrical signals. Furthermore, ultrasonic travel at a faster rate than audible sounds. Therefore, ultrasonic sensor work involves sound waves to find the distance to an item. A transducer is also there to transmit and receive ultrasonic pulses. These pulses help to communicate information about an object within range.  Further, this detail can be applied in various applications including industrial.

**Working of an Ultrasonic Sensor**

Ultrasonic sensors operate by emitting sound waves at frequencies that are too high for humans to hear. The sensor's transducer serves as a microphone to receive and transmit ultrasonic sound. They also use a single transducer to send and receive pulses. Further, the sensor measures the total time taken to deliver and receive an ultrasonic pulse and calculates the target's distance.



**Figure 6.1** Ultrasonic sensor

* **Emitting Sound Waves:** The ultrasonic sensor generates a burst of ultrasonic sound waves, usually in the range of 20 kHz to 65 kHz. These sound waves travel through the air towards the target object.
* **Bouncing Off Objects:** When the sound waves encounter an object in their path, they bounce off the surface of the object.
* **Measuring the Return Time:** The sensor measures the time it takes for the emitted sound waves to bounce back after hitting the object. This time interval is extremely short, typically measured in microseconds.
* **Calculating Distance:** The sensor figures out how far an object is by using the speed of sound in the air. It does this by multiplying the time it takes for sound waves to go to the object and then coming back by the speed of sound.

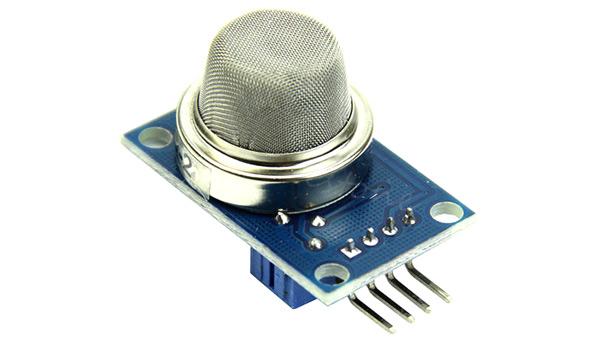
**Output Data:** Last, the sensor provides this distance information as an output, which can be used in various applications, such as obstacle detection, object positioning, or navigation.

**Applications of Ultrasonic Sensors:**

* Ultrasonic sensors can measure the coil/roll's diameter.
* Useful in quality control.
* Applicable in the Detection of proximity.
* Helpful to position robotic arms through robotic sensing.
* Used in security system management
* ultrasonic sensors are crucial for detecting obstacles in the environment.
* It can determine whether an object is present within a specific range.

**6.1.2 Gas Sensor**

A gas sensor is a device which detects the presence or concentration of gases in the atmosphere. Based on the concentration of the gas the sensor produces a corresponding potential difference by changing the resistance of the material inside the sensor, which can be measured as output voltage. Based on this voltage value the type and concentration of the gas can be estimated.



**Figure 6.2 Gas sensor**

The type of gas the sensor could detect depends on the sensing material present inside the sensor. Normally these sensors are available as modules with comparators as shown above. These comparators can be set for a particular threshold value of gas concentration. When the concentration of the gas exceeds this threshold, the digital pin goes high. The analog pin can be used to measure the concentration of the gas.

**Construction**

All Of the above-listed types, the most commonly used gas sensor is the Metal oxide semiconductor-based gas sensor. All Gas sensors will consist of a sensing element which comprises of the following parts.

1.Gas sensing layer

2.Heater Coil

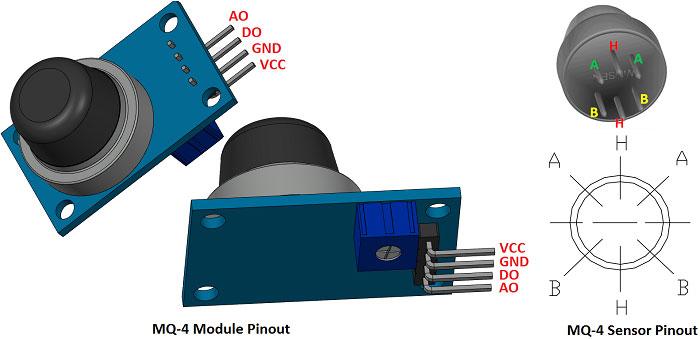
3.Electrode line

4.Tubular ceramic

5.Electrode

**Working of Gas sensor**

A basic gas sensor has 6 terminals in which 4 terminals (A, A, B, B) acts input or output and the remaining 2 terminals (H, H) are for heating the coil. Of these 4 terminals, 2 terminals from each side can be used as either input or output (these terminals are reversible as shown in the circuit diagram) and vice versa.



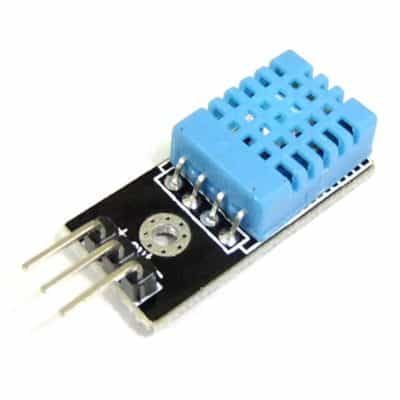
**Figure 6.2.1** Working of Gas sensor

These sensors are normally available as modules (shown right), these modules consist of the gas sensor and a [comparator IC](https://components101.com/tags/voltage-comparator). Now let’s see the pin description of the gas sensor module which we will generally use with an Arduino. The gas sensor module basically consists of 4 terminals

* Vcc – Power supply
* GND – Power supply
* Digital output – This pin gives an output either in logical high or logical low (0 or1) that means it displays the presence of any toxic or combustible gases near the sensor.
* Analog output – This pin gives an output continuous in voltage which varies based on the concentration of gas that is applied to the gas sensor.

**6.1.3 Temperature Sensor**

DHT11 sensor is used here to monitor the humidity variation of the environment where the crops are cultivated. This is a digital sensor and measures the humidity value in percentage format.



**Figure 6.3** Temperature sensor (DHT11)

It is available as a sensor and as a module. The difference between this sensor and module is the pull-up resistor and a power-on LED. DHT11 is a relative humidity sensor.  To measure the surrounding air this sensor uses a [thermistor](https://www.elprocus.com/introduction-to-thermistor-types-with-its-workings-and-applications/) and a capacitive humidity sensor.DHT11 is a low-cost digital sensor for sensing temperature and humidity.  This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc. to measure humidity and temperature instantaneously.

**Working**

DHT11 sensor consists of a capacitive humidity sensing element and a thermistor for sensing temperature.  The humidity sensing [capacitor](https://www.elprocus.com/construction-of-capacitor-with-working/) has two electrodes with a moisture holding substrate as a dielectric between them. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process this changed resistance values and change them into digital form.

For measuring temperature this sensor uses a Negative Temperature coefficient thermistor, which causes a decrease in its resistance value with increase in temperature. To get larger resistance value even for the smallest change in temperature, this sensor is

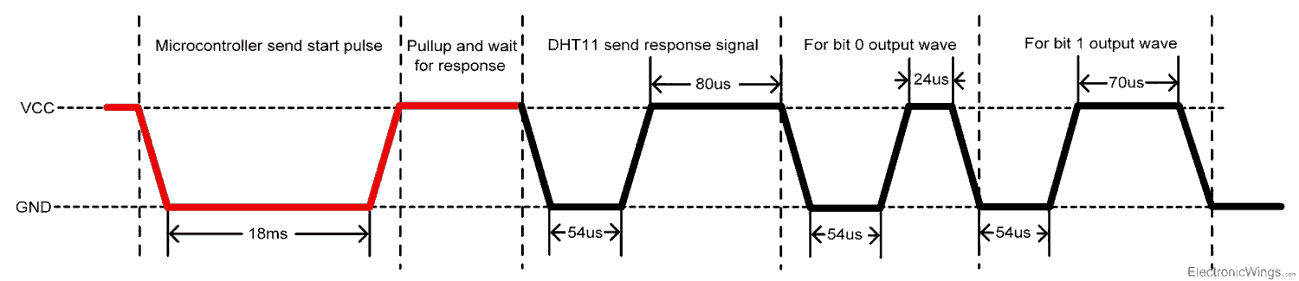
usually made up of semiconductor ceramics or polymers.

The temperature range of DHT11 is from 0 to 50 degree Celsius with a 2-degree accuracy. Humidity range of this sensor is from 20 to 80% with 5% accuracy. The sampling rate of this sensor is 1Hz.i.e., it gives one reading for every second.  DHT11 is small in size with operating voltage from 3 to 5 volts. The maximum current used while measuring is 2.5mA.

DHT11 sensor has four pins- VCC, GND, Data Pin and a not connected pin. A pull-up resistor of 5k to 10k ohms is provided for communication between sensor and micro-controller.

**HUMIDITY (DHT11):**

* DHT11 uses only one wire for communication. The voltage levels with certain time value defines the logic one or logic zero on this pin. The communication process is divided in three steps, first is to send request to DHT11 sensor then sensor will send response pulse and then it starts sending data of total 40 bits to the microcontroller.

**Figure 6.3.1** Humidity wave representation

**Applications of DHT11:**

1. Smart Wearable Devices
   * Used in IoT-based smart clothing like the Smart E-Uniform to monitor the soldier’s surrounding environment for extreme temperatures.
2. Weather Monitoring Systems
   * Commonly used in home weather stations, greenhouses, and agricultural fields for real-time temperature and humidity logging.
3. HVAC System

* Helps in automating Heating, Ventilation, and Air Conditioning (HVAC) systems based on environmental conditions.

4.Smart Home Automation

* + Integrated into home automation systems to trigger cooling or heating devices based on room temperature.

5.Environmental Monitoring

* + Used in labs, factories, and warehouses to ensure controlled environments for sensitive equipment or materials.

1. IoT Projects & Educational Kits
   * Popular among students and hobbyists for learning sensor interfacing, data logging, and wireless monitoring.

**6.1.4 RELAY**

Relays are electrically operated switches that open and close the circuits by receiving electrical signals from outside sources. Some people may associate “relay” with a racing competition where members of the team take turns passing batons to complete the race.  
The “relays” embedded in electrical products work in a similar way; they receive an electrical signal and send the signal to other equipment by turning the switch on and off.

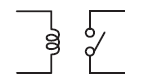
For example, when you push the button on a TV remote to watch TV, it sends an electrical signal to the “relay” inside the TV, turning the main power ON. There are various types of relays used in many applications to control different amounts of currents. ****

**Figure 6.4** Relay

## **Electrical Relay Types and Classification**

Relay technology can be divided into two main categories: Movable contacts (mechanical relay) and no movable contacts (MOS FET relay, solid state relay).

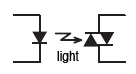
### Movable contacts ( Mechanical Relay)



**Figure 6.4.1** Movable Relay

This type of relay has contacts that are mechanically actuated to open/close by a magnetic force to switch signals, currents and voltages ON or OFF.

### Non movable contacts ( MOS FET relay, Solid State Relay)



**Figure 6.4.2** Non-Movable Relay

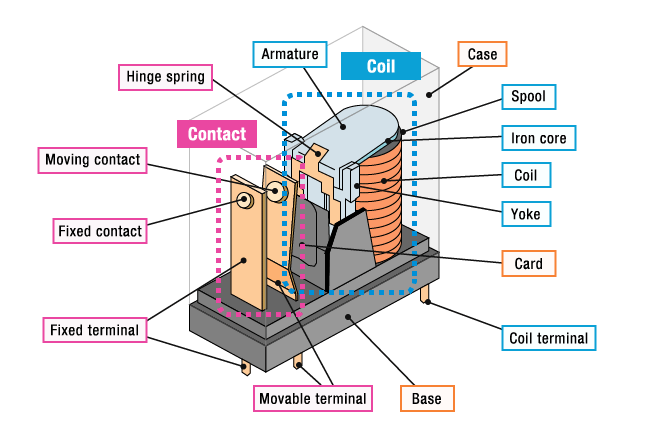
Unlike mechanical relays, this type of relay has no moving contacts but instead employs semiconductor and electrical switching elements such as triac and MOS FET. By the operation of these electronic circuits, signals, currents and voltages are switched ON or OFF electronically.

## **Electrical relay Structure and Operating Principles**

Relays can be of different types like electromechanical, [solid state](https://www.electronicshub.org/solid-state-relay/). Electromechanical relays are frequently used. Let us see the internal parts of this relay before knowing about it working. Although many different types of relays were present, their working is same.

Every electromechanical relay consists of an consists of an

1. Electromagnet
2. Mechanically movable contact
3. Switching points and
4. Spring

 **Figure 6.4.3** Relay working

**Principles of Relay:**

* Relay works on the principle of electromagnetic induction.
* When the electromagnet is applied with some current, it induces a magnetic field around it.
* Above image shows working of the relay. A switch is used to apply DC current to the load.
* In the relay, Copper coil and the iron core acts as electromagnet.
* When the coil is applied with DC current, it starts attracting the contact as shown. This is called energizing of relay.
* When the supply is removed it retrieves back to the original position. This is called De energizing of relay.

There are also such relays, whose contacts are initially closed and opened when there is supply i.e., exactly to opposite to the above shown relay.

Solid state relays will have sensing element to sense the input voltage and switches the output using opto-coupling.

**Relay Contact Types**

As we have seen that relay is a switch. The terminology “Poles and throws” is also applicable for relay. Depending on the number of contacts and number of circuits it switches relays can be classified.

Before we know about this classification of contacts, we have to know the poles and throws of a relay switch.

**Poles and Throws**

Relays can switch one or more circuits. Each switch in relay is referred as pole. Number of circuits a relay connects is indicated by throws.

Depending on the poles and throws, relays are classified into

* Single Pole Single Throw
* Single Pole Double Throw
* Double Pole Single Throw
* Double Pole Double Throw

**Types of Relays**

* Electromagnetic
* Latching
* Electronic
* Non-Latching
* High-Voltage
* Polarized
* Rotary
* Sequence
* Moving Coil
* Safety
* Supervision
* Ground Fault
* Lighting control systems
* Telecommunication
* Industrial process controllers
* Traffic control
* Motor drives control
* Protection systems of electrical power system
* Computer interfaces
* Automotive
* Home appliances

**Testing of relay**

Since they are electromechanical devices, relays can wear out eventually and stop working overtime. But there are few   techniques to test if a relay is working or not. These techniques include:

* Testing a Relay with a Multi meter
* Build a simple circuit to test the Relay

Use a DC Power Supply to see whether a relay is functioning properly

**6.1.5 PELTIER PANEL:**

A Peltier panel, or thermoelectric cooler, works by using the Peltier effect, where a direct current flowing through junctions of different materials causes one side to heat up and the other to cool down.

Here's a more detailed explanation:

**The Peltier Effect:**

When a direct current (DC) flows through the junctions of two different materials, one side of the junction absorbs heat (cooling) and the other side releases heat (heating).



**Figure 6.5** Peltier Panel

**Peltier Panel Construction:**

A Peltier panel consists of multiple pairs of p-type and n-type semiconductor materials connected in series, with copper bridges acting as electrical connections.

**Working of peltier panel:**

When a DC current flows through the panel, electrons move from the cold side to the hot side.

As electrons move, they absorb heat from the cold side, causing it to cool down.

The electrons then release the absorbed heat on the hot side, causing it to heat up.

Materials:

Bismuth telluride (Bi2Te3) is a common material used in Peltier panels due to its high performance in the temperature range of -100 to +200°C.

**Applications:**

Peltier panels are used in various applications, including cooling electronic components, small refrigerators, and other applications where compact and efficient cooling is needed.

**6.1.6 THERMAL FAN**

Fan sensors, often embedded tachometers or pulse sensors, detect and report the rotational speed of a fan motor using a pulsed signal, enabling closed-loop feedback control and monitoring for optimal cooling performance.



**Figure 6.6** Thermal Fan

**Tachometer/Pulse Sensors:**

* These sensors provide information about the fan's speed in RPM (revolutions per minute) through a pulsed signal.

**Function:**

* The sensor detects the fan motor's rotational speed and sends this data as a series of pulses, which can be interpreted by a controller to adjust fan speed or monitor its performance.

**Types:**

* Embedded Tachometer: This type is integrated into the fan motor and provides a continuous signal indicating the fan's speed.
* Locked Rotor Signal: This signal indicates whether the fan motor is rotating or stopped, helping to detect potential issues.

**Applications:**

* Closed-Loop Control: The sensor's output allows for precise control of fan speed based on temperature or other factors.
* Monitoring: The sensor data helps monitor fan performance, detect malfunctions (like a stopped fan), and ensure optimal cooling.

Example:

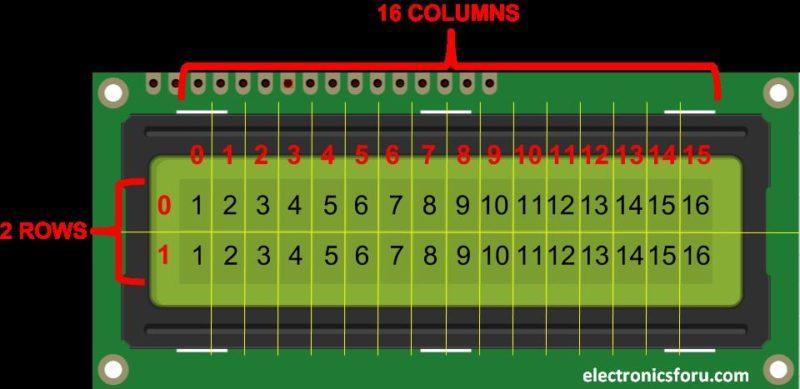
* In a computer, a fan sensor can detect if a fan is spinning too slowly or has stopped, triggering an alert or shutting down the system to prevent overheating.
* 4-Wire Fans:
* These fans have a PWM (Pulse Width Modulation) input for speed control, a tachometer signal for feedback, and power and ground connections.
* 3-Wire Fans
* These fans have power, ground, and tachometer signals.

**6.1.7 LCD**

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16×2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A 16×2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5×7-pixel matrix.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

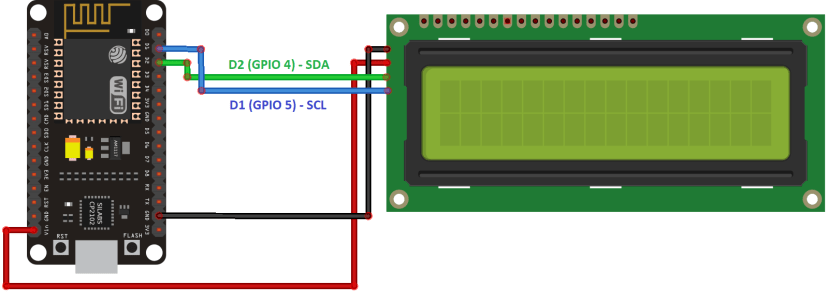


**Figure 6.7** LCD

**Working:**

The I2C LCD operates by connecting to a microcontroller through two wires: SDA (Serial Data) and SCL (Serial Clock). It communicates using the I2C protocol, receiving commands and data from the microcontroller. An embedded controller within the LCD module interprets and processes this data to control the display, including showing characters, numbers, and symbols while efficiently using minimal

hardware pins. This streamlined communication method simplifies wiring and hardware requirements, making it popular in various embedded systems.



**Figure 6.7.1** Pin configuration

**Table 6.7** LCD pin Description:

|  |  |
| --- | --- |
| **Pin Name** | **Description** |
| VCC | Power supply (usually +5V) |
| GND | Ground |
| SDA | I2C Serial Data Line |
| SCA | I2C Serial Clock Line |

**Features of 16×2 LCD module**

* Operating Voltage is 4.7V to 5.3V
* Current consumption is 1mA without backlight
* Alphanumeric LCD display module, meaning can display alphabets and numbers.
* Consists of two rows and each row can print 16 characters.
* Each character is built by a 5×8-pixel box
* Can work on both 8-bit and 4-bit mode
* It can also display any custom generated characters
* Available in Green and Blue Backlight.

**Advantage**

* LCDs are consisting of some microwatts for display in comparison to some mill watts for LED’s.
* LCDs are of low cost
* LCDs consumes less amount of power compared to CRT and LED
* LCDs are thinner and lighter when compared to cathode ray tube and LED

**Applications**

* Liquid crystal technology has major applications in the field of science and engineering as well on electronic devices.
* Liquid crystal thermometer
* Optical imaging
* The liquid crystal display technique is also applicable in visualization of the radio frequency waves in the waveguide.
* Used in the medical applications.

**6.1.8 EMERGENCY ALERT SYSTEM**

The Smart E-Uniform is equipped with a DHT11 temperature and humidity sensor, which continuously monitors the soldier's body temperature. When the temperature readings cross pre-defined threshold limits, an emergency alert is automatically triggered. These thresholds are:

* High Temperature Threshold: Above 35°C
* Low Temperature Threshold: Below 29°C

Once the temperature moves beyond either of these thresholds, the system initiates a multi-channel emergency alert mechanism to notify the base station and concerned authorities**.**

The E-uniform triggers three types of emergency alerts based on the distance between the soldier and the base station:

**Short Range:**

* MQTT (Message Queuing Telemetry Transport) is a lightweight messaging protocol used for real-time communication.
* Sensor data (temperature, gas, distance) is sent instantly to the base station via MQTT for immediate monitoring.

**Medium Range:**

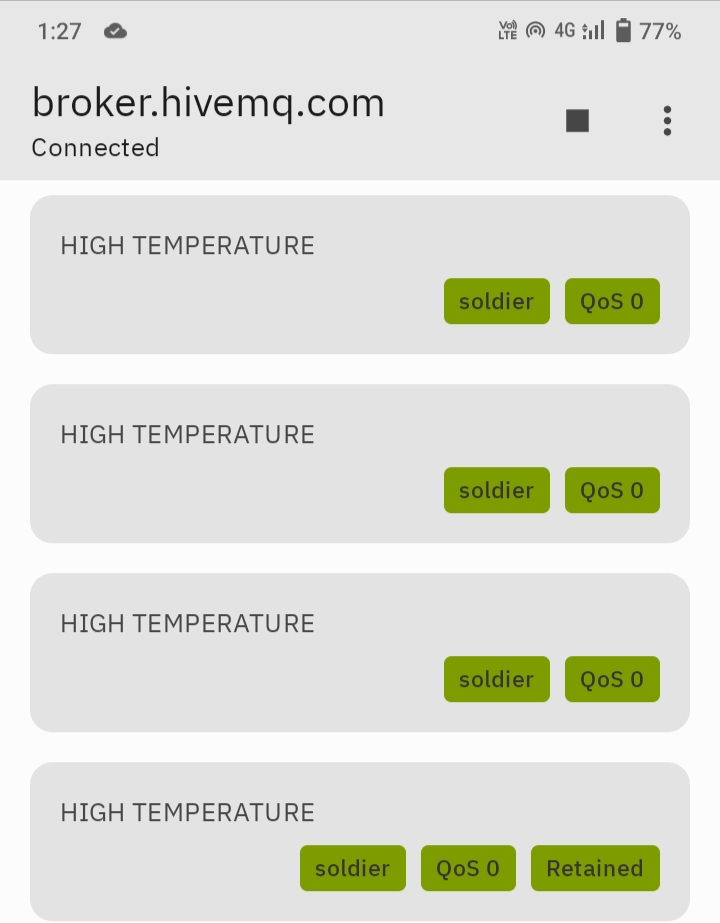
* Email Alert is sent.
* A complete status report is emailed to the control center using internet-based SMTP services.

**Long Range:**

* SMS Alert is triggered via GSM network using Twilio.
* A short, urgent text message is sent to base station for ensure timely action.
* The base station can forward these alerts to rescue teams for ensure the soldier’s safety.

**MQTT ALERT**

MQTT is a lightweight publish/subscribe messaging protocol. It is useful for use with low power sensors, but is applicable to many scenarios.



**Figure 6.8** MYMQTT APP

**Publish/Subscribe**

The MQTT protocol is based on the principle of publishing messages and subscribing to topics, or "pub/sub". Multiple clients connect to a broker and subscribe to topics that they are interested in. Clients also connect to the broker and publish messages to topics. Many clients may subscribe to the same topics and do with the information as they please. The broker and MQTT act as a simple, common interface for everything to connect to.

**Topics/Subscriptions**

Messages in MQTT are published on topics. There is no need to configure a topic, publishing on it is enough. Topics are treated as a hierarchy, using a slash (/) as a separator. This allows sensible arrangement of common themes to be created, much in the same way as a filesystem. For example, multiple computers may all publish their hard drive temperature information on the following topic, with their own computer and hard drive name being replaced as appropriate:

**Client**

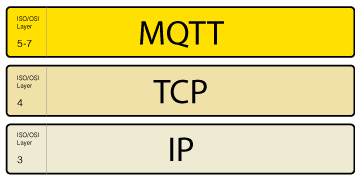
When we talk about a client, we almost always mean an [MQTT client](https://www.hivemq.com/blog/seven-best-mqtt-client-tools/). Both publishers and subscribers are MQTT clients. The publisher and subscriber labels refer to whether the client is currently publishing messages or subscribing to messages (publish and subscribe functionality can also be implemented in the same MQTT client). An MQTT client is any device (from a micro controller up to a full-fledged server) that runs an MQTT library and connects to an [MQTT broker](https://www.hivemq.com/hivemq/) over a network.

**Broker**

The counterpart of the MQTT client is the MQTT broker. The broker is at the heart of any publish/subscribe protocol. Depending on the implementation, a broker can handle up to thousands of concurrently connected MQTT clients. The broker is responsible for receiving all messages, filtering the messages, determining who is subscribed to each message, and sending the message to these subscribed clients.

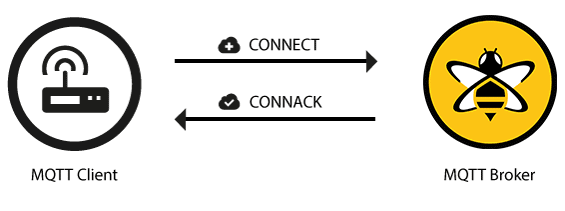
**Working of MQTT**

The MQTT protocol is based on TCP/IP. Both the client and the broker need to have a TCP/IP stack.



**Figure 6.8.1** MQTT Layer

The MQTT connection is always between one client and the broker. Clients never connect to each other directly. To initiate a connection, the client sends a CONNECT message to the broker. The broker responds with a CONNACK message and a status code. Once the connection is established, the broker keeps it open until the client sends a disconnect command or the connection breaks.



**Figure 6.8.2** MQTT Connection

**Client initiates connection with the CONNECT message**

To initiate a connection, the client sends a command message to the broker. If this CONNECT message is malformed (according to the MQTT specification) or too much time passes between opening a network socket and sending the connect message, the broker closes the connection. This behavior deters malicious clients that can slow the broker down. A good-natured client sends a connect message with the following content (among other things): For all the details, have a look at the [MQTT 3.1.1 specification](http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html).

**ClientID**

The client identifier (ClientID) identifies each MQTT client that connects to an MQTT broker. The broker uses the ClientID to identify the client and the current state of the client. Therefore, this ID should be unique per client and broker. In MQTT 3.1.1 (the current standard), you can send an empty ClientID.

**Username/Password**

MQTT can send a user name and password for client authentication and authorization. However, if this information isn’t encrypted or hashed (either by implementation or TLS), the password is sent in plain text. We highly recommend the use of user names and passwords together with a secure transport. Brokers like HiveQL can authenticate clients with an SSL certificate, so no username and password is needed.

**Will Message**

The last will message is part of the Last Will and Testament (LWT) feature of MQTT. This message notifies other clients when a client disconnects ungracefully. When a client connects, it can provide the broker with a last will in the form of an MQTT message and topic within the CONNECT message.

**MQTT in IoT**

MQTT is one of the most commonly used protocols concerning IoT. MQTT enables resource-constrained IoT devices to send, or publish, information about a given topic to a server that functions as an MQTT message broker. The broker then pushes the information out to those clients that have previously subscribed to the topic. To a human, a topic looks like a hierarchical file path. Clients can subscribe to a specific level of a topic's hierarchy or use a wild-card character to subscribe to multiple levels.

**Pros and cons of MQTT**

MQTT has a few distinct advantages and disadvantages when compared to competing protocols. Advantages include the following:

* efficient data transmission and quick to implement due to its being a lightweight protocol;
* low network usage, due to minimized data packet.
* efficient distribution of data;
* successful implementation of remote sensing and control;
* fast and efficient message delivery;
* usage of small amounts of power, which is good for the connected devices
* reduction of network bandwidth

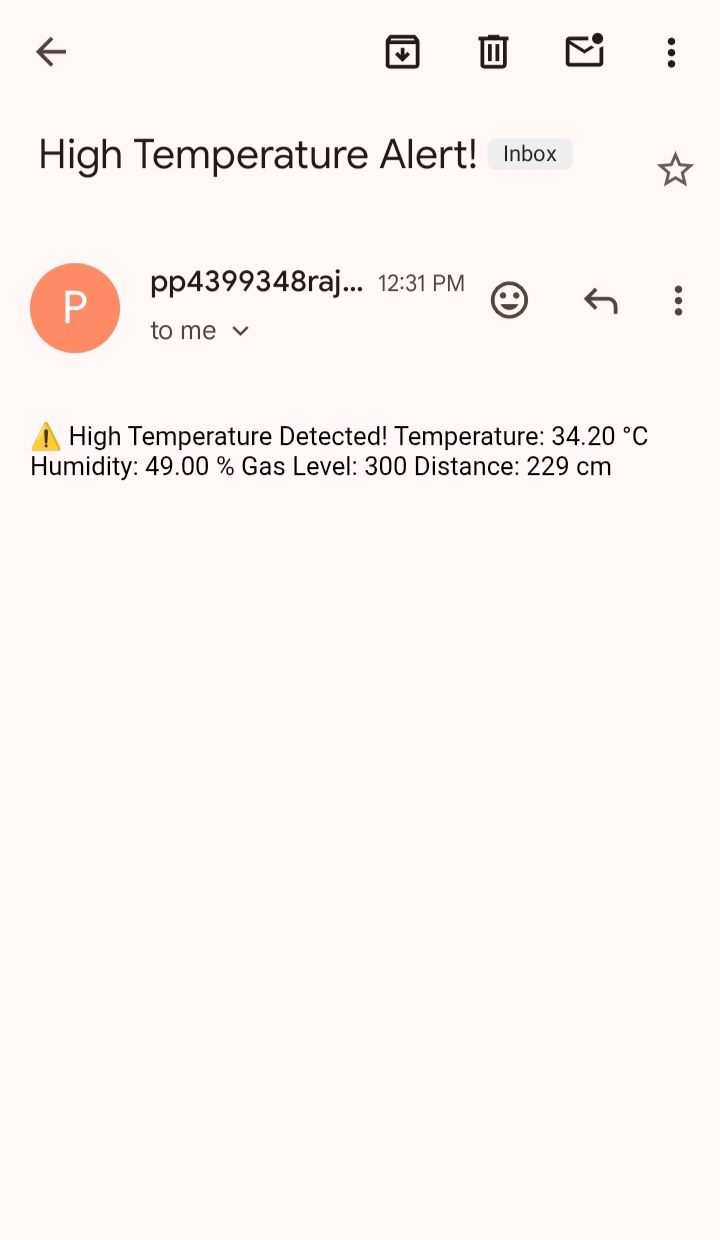
**Specifications**

MQTT has different specifications depending on the specific version. Version 5.0 superseded the last version of MQTT, version 3.1.1. Some newer specifications, as defined by OASIS, include the following:

* the use of publish/subscribe message patterns;
* a mechanism that can notify users when abnormal disconnections occur;
* the three levels of message delivery: at most once, at least once and exactly once;
* the minimization of transport overhead and protocol exchanges to reduce network traffic; and
* an agnostic messaging transport referring to the content of the payload.

**E-MAIL ALERT**

One of the primary alert mechanisms is **Email Notification** using the **SMTP (Simple Mail Transfer Protocol)**. This is implemented to ensure that detailed and documented alerts reach the base station even in low-bandwidth conditions.



**Figure 6.8.3** E-mail Alert

**WORKING MECHANISM**

**Data Collection**  
The DHT11 sensor detects the real-time temperature and humidity values.

**Threshold Detection Logic**  
Microcontroller (e.g., ESP8266) evaluates the temperature against the defined high and low thresholds.

**Alert Trigger**  
Upon detecting a critical temperature condition:

* The microcontroller composes an email with all relevant sensor data (temperature, humidity, soldier ID, location if available).
* An SMTP connection is established with a secure mail server (e.g., smtp.gmail.com).
* The email is sent automatically to the pre-configured base station address.

**Message Format**  
The email includes a clearly structured message such as:

**Subject:** Emergency Alert - Abnormal Temperature Detected

**Body:**  
Temperature: 35.6°C  
Humidity: 55%  
Status: High Temperature Alert

**Redundancy & Frequency Control**

* The system ensures that alerts are not repeatedly sent for the same condition unless the temperature returns to normal and crosses the threshold again.
* This prevents spam and ensures relevant alerts only.

#### **Advantages of SMTP Email Alerts:**

* Ensures **traceable and documented** communication
* Can be accessed on any device (laptop, smartphone)
* Integrates easily with existing military communication systems
* Secure and reliable delivery using encrypted SMTP protocols.

**SMS ALERT**

As part of the **Emergency Alert System** in the Smart E-Uniform for Soldiers Using IoT, **Twilio API** is integrated to send **real-time SMS alerts** directly to the base station or authorized personnel when abnormal temperature conditions are detected. This ensures prompt response even in environments where internet-based notifications (like email or MQTT) might be delayed or unavailable.

**Purpose**

The purpose of integrating Twilio SMS alerts is to provide a fast, reliable, and direct method of communication using mobile networks. This helps safeguard the health and safety of soldiers by enabling instant alert transmission, especially in critical field conditions.

**Working Mechanism**

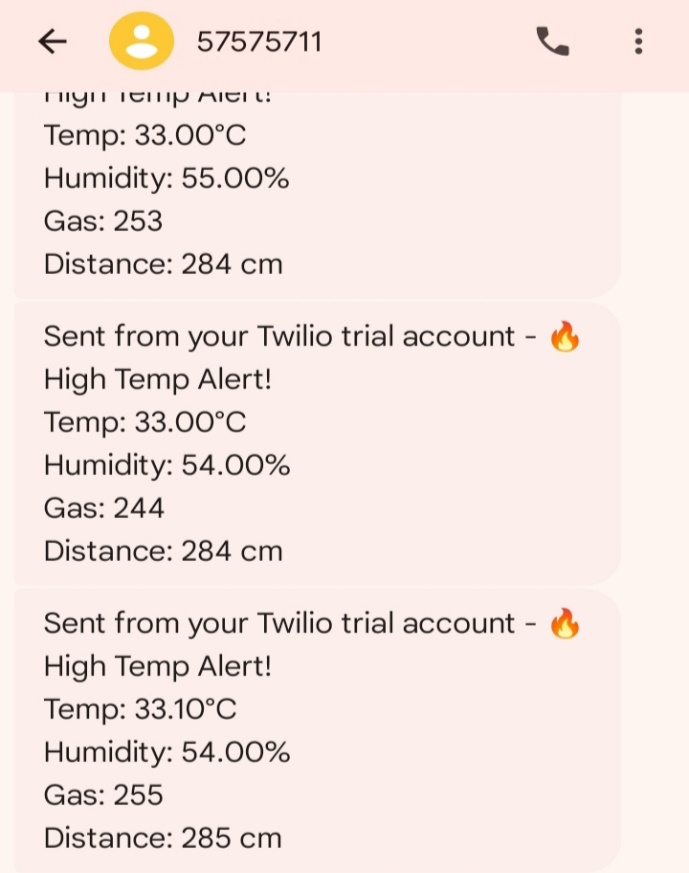
**Temperature Monitoring:**  
The system uses a **DHT11 sensor** to monitor the soldier’s body temperature in real time.

**Threshold Evaluation**

* + If **temperature > 33°C**, it is considered **high**.
  + If **temperature < 28°C**, it is considered **low.**

**Triggering SMS Alert**  
Upon detecting a high or low temperature:

* + An alert message is constructed containing:
    - Temperature and Humidity values
    - Type of alert (High/Low)
    - Timestamp
  + The system initiates an HTTPS request to the **Twilio REST API** using:
    - **Account SID**
    - **Auth Token**
    - **Twilio phone number**
    - **Recipient's mobile number**
  + The alert SMS is sent instantly to the designated recipient(s).



**Figure 6.8.4** SMS Alert

#### **Advantages of Twilio SMS Integration**

* **Instant Delivery:** Sends alerts within seconds to base station or supervisors.
* **Works in Low Connectivity Areas:** Only mobile network is needed, no internet.
* **Scalable:** Multiple recipient numbers can be configured.
* **Reliable:** Twilio provides delivery confirmation and supports global SMS delivery.

**6.1.9 REAL-TIME MONITORING VIA THING SPEAK**

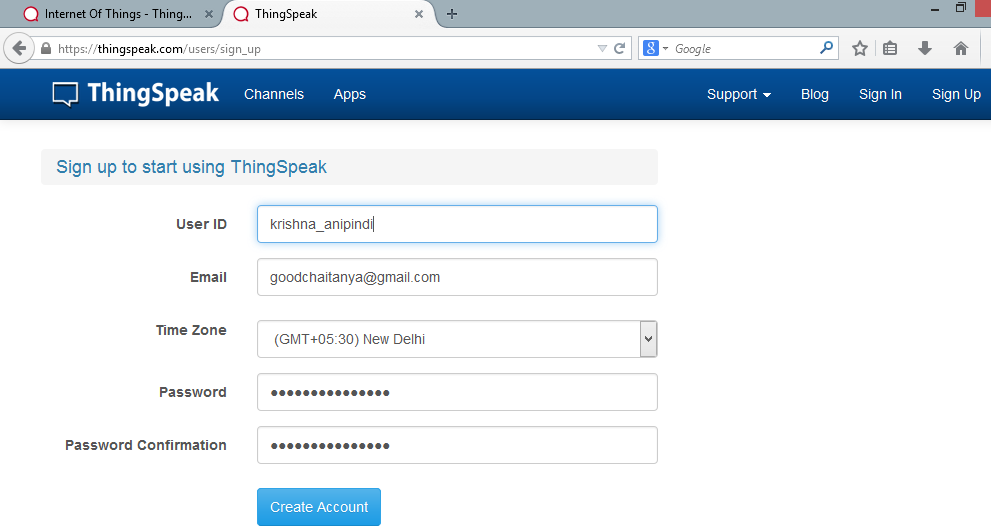
Thing Speak is a platform providing various services exclusively targeted for building IoT applications. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, ability to create plugins and apps for collaborating with web services, social network and other APIs. We will consider each of these features in detail below.

The core element of Thing Speak is a ‘Thing Speak Channel’. A channel stores the data that we send to Thing Speak and comprises of the below elements:

* 8 fields for storing data of any type - These can be used to store the data from a sensor or from an embedded device.
* 3 location fields - Can be used to store the latitude, longitude and the elevation. These are very useful for tracking a moving device.
* 1 status field - A short message to describe the data stored in the channel.

**Getting Started**

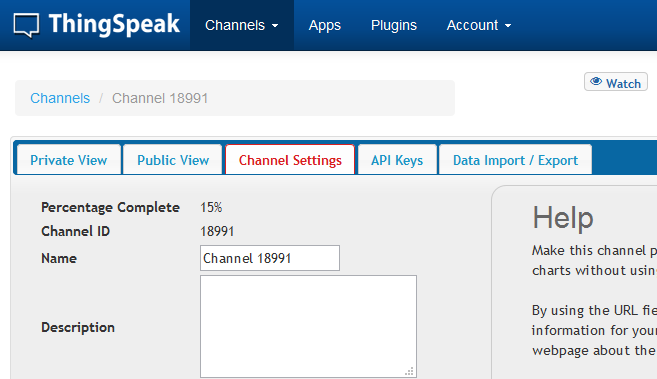
* Open <https://thingspeak.com/>and click on the ‘Get Started Now’ button on the center of the page and you will be redirected to the sign-up page(you will reach the same page when you click the ‘Sign Up’ button on the extreme right). Fill out the required details and click on the ‘Create Account’ button.



6.9 Thing speak Account

Now you should see a page with a confirmation that the account was successfully created.

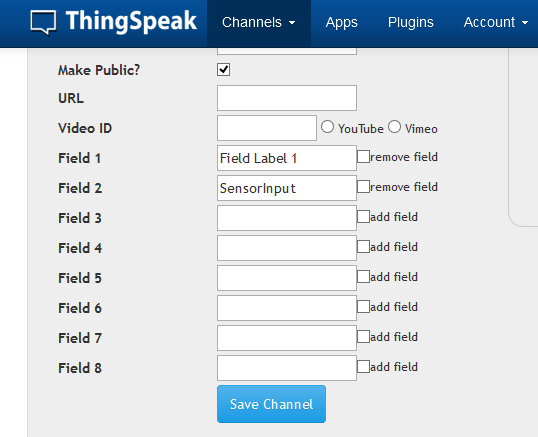
Go ahead and click on ‘New Channel’. You should see a page like the below

**Figure 6.9.1** Thing speak channel

You can change the name to fit your need and you can add a description corresponding to the channel.

* Fields 1 to 8 - These are the fields which correspond to the data sent by a sensor or a thing.

.In case you try posting to fields that you have not added, your request will still be successful, but you will not be able to see the field in the charts and the corresponding data. You can click on the small box before the ‘add field’ text corresponding to each field to add it. Once you click the ‘add field’ box.

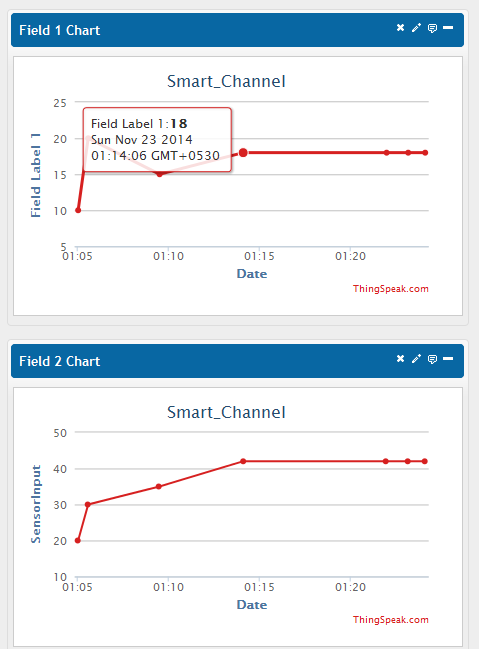


**Figure 6.9.2** Field creation

Once you have edited the fields, click on ‘Save Channel’ button.

A throwing exception and displaying the resultant message in a label.

After a series of updates, the charts in the private view tab for each of the fields will look like the below.



**Figure 6.9.3** Data in Graph

Each of the dots correspond to the value and the time at which the value was posted to the channel. Place the mouse over a dot to get more details on the exact date and the GMT offset from which the value was posted.

**Thing Speak Apps**

Thing Speak provides apps that allow us for an easier integration with the web services, social networks and other APIs. Below are some of the apps provided by Thing Speak:

* Thing Tweet - This allows you to post messages to twitter via Thing Speak. In essence, this is a Twitter Proxy which re-directs your posts to twitter.
* Thing HTTP - This allows you to connect to web services and supports GET, PUT, POST and DELETE methods of HTTP.
* Tweet Control - Using this, you can monitor your Twitter feeds for a specific key word and then process the request. Once the specific keyword is found in the twitter feed, you can then use Thing HTTP to connect to a different web service or execute a specific action.
* React - Send a tweet or trigger a Thing HTTP request when the Channel meets a certain condition.

**6.2 PROJECT CODE IMPLEMENTATION**

#### **6.2.1 Introduction**

This topic presents the complete source code developed for the Smart E-Uniform for Soldiers Using IoT. The code is written in C++ and executed using the Arduino IDE. It integrates multiple sensors and communication modules to enable real-time monitoring and alerting functionalities, including email and SMS-based emergency notifications.

**Libraries Used**:

* ESP8266WiFi.h
* DHT.h
* WiFiClientSecure.h
* LiquidCrystal\_I2C.h
* base64.h
* Custom Libraries for ThingSpeak / Twilio / SMTP

**6.2.2 Complete Source Code:**

#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include <LiquidCrystal\_I2C.h>

#include <DHT.h>

#include <EMailSender.h>

#include <base64.h>

#include <WiFiClientSecure.h>

// LCD Setup

LiquidCrystal\_I2C lcd (0x27, 16, 2);

// Pin Definitions

#define TRIG\_PIN D3

#define ECHO\_PIN D4

#define GAS\_SENSOR A0

#define RELAY D5

#define RELAY1 D6

#define DHTPIN D7

#define DHTTYPE DHT11

// Email Sender

EMailSender emailSend ("pp4399348raj@gmail.com", "yaoh maap cwob ldwo");

// Twilio credentials

const char\* accountSID = "AC3196a13378a5b0de7af86bf5de03538d";

const char\* authToken = "f37edf78846e37f7d5b57e410e2cffec";

const char\* fromNumber = "+12298505618"; // Twilio

const char\* toNumber = "+919361208320"; // Your number

const char\* twilioHost = "api.twilio.com";

const int httpsPort = 443;

// Network + MQTT

const char\* ssid = "vivo Y15s";

const char\* password = "prakash29";

const char\* mqtt\_server = "broker.hivemq.com";

// ThingSpeak

const char\* TS\_SERVER = "api.thingspeak.com";

String TS\_API\_KEY = "KV6LEKXTDZGLWJLX";

// Objects

DHT dht (DHTPIN, DHTTYPE);

WiFiClient espClient;

PubSubClient client(espClient);

// MQTT

void callback (char\* topic, byte\* payload, unsigned int length) {}

void reconnect () {

while (! client. connected ()) {

String clientId = "ESP8266-" + String(random(0xffff), HEX);

if (client. connect (clientId.c\_str ())) {

client. subscribe("soldier");

} else {

delay (5000);

}

}

}

void mqtt () {

client. setCallback(callback);

if (! client. connected ()) reconnect ();

client. loop ();

}

// WiFi

void connectwifi () {

WiFi.begin(ssid, password);

while (WiFi.status()! = WL\_CONNECTED) delay (500);

Serial.println("WiFi connected");

}

int entryCount=0;

//Ultrasonic Distance Funtion

long readUltrasonicDistance () {

digitalWrite (TRIG\_PIN, LOW); delayMicroseconds (2);

digitalWrite (TRIG\_PIN, HIGH); delayMicroseconds (10);

digitalWrite (TRIG\_PIN, LOW)

long duration = pulseIn (ECHO\_PIN, HIGH);

return duration \* 0.034 / 2;

}

// Send data to ThingSpeak

void sendToThingSpeak (float temperature, float humidity, int gasValue, long distance) {

if (WiFi.status() == WL\_CONNECTED) {

WiFiClient client;

const int httpPort = 80;

if (! client. connect (TS\_SERVER, httpPort)) {

Serial.println("Connection to ThingSpeak failed");

return;

}

String url = "/update?api\_key=" + TS\_API\_KEY;

url += "&field1=" + String(temperature);

url += "&field2=" + String(humidity);

url += "&field3=" + String(gasValue);

url += "&field4=" + String(distance);

client.print (String ("GET ") + url + " HTTP/1.1\r\n" +

"Host: " + TS\_SERVER + "\r\n" +

"Connection: close\r\n\r\n");

Serial.println("Sent data to ThingSpeak:");

Serial.println(url);

} else {

Serial.println("WiFi not connected");

}

}

// URL Encoding helper

String urlencode (String str) {

String encodedString = "";

char c, code0, code1;

for (int i = 0; i < str. length (); i++) {

c = str. charAt (i)

if (isalnum(c)) {

encodedString += c;

} else {

code1 = (c & 0xf) + '0'; if ((c & 0xf) > 9) code1 = (c & 0xf) - 10 + 'A';

c = (c >> 4) & 0xf; code0 = c + '0'; if (c > 9) code0 = c - 10 + 'A';

encodedString += '%'; encodedString += code0; encodedString += code1;

}

}

return encodedString;

}

// Twilio SMS Function

void sendSMS (String messageBody) {

WiFiClientSecure client;

client. setInsecure ();

if (! client. connect (twilioHost, httpsPort)) {

Serial.println("❌ Twilio connection failed");

return;

}

// Use raw phone number directly without variables

String postData = "To=%2B919361208320" // "+91..." properly encoded

"&From=%2B12298505618" // Twilio number

"&Body=" + urlencode(messageBody);

String credentials = String(accountSID) + ":" + String(authToken);

String encodedCreds = base64::encode(credentials);

String url = "/2010-04-01/Accounts/" + String(accountSID) + "/Messages.json";

client.print (String ("POST ") + url + " HTTP/1.1\r\n" +

"Host: " + twilioHost + "\r\n" +

"Authorization: Basic " + encodedCreds + "\r\n" +

"Content-Type: application/x-www-form-urlencode\r\n" +

"Content-Length: " + postData. Length () + "\r\n\r\n" +

postData);

Serial.println("📡 SMS Request Sent. Awaiting Response:");

while (client. connected ()) {

String line = client. readStringUntil ('\n');

if (line == "\r") break;

Serial.println(line);

}

String responseBody = client. readString ();

Serial.println("📨 SMS Response:");

Serial.println(responseBody);

}

void setup () {

Serial.begin(9600);

//pin setup

pinMode (TRIG\_PIN, OUTPUT);

pinMode (ECHO\_PIN, INPUT);

pinMode (GAS\_SENSOR, INPUT);

pinMode (RELAY, OUTPUT);

pinMode (RELAY1, OUTPUT);

digitalWrite (RELAY, LOW); // Peltier OFF

digitalWrite (RELAY1, LOW); // Fan OFF

dht. begin ();

delay (1000); // Let DHT settle

dht. read Temperature (); // dummy read to stabilize

lcd. init ();

lcd. backlight ();

//connect wifi&MQTT

connectwifi ();

client. setServer (mqtt\_server, 1883);

}

void loop () {

mqtt ();

// Read sensors

float temperature = dht. read Temperature ();

float humidity = dht. read Humidity ();

// Validate DHT sensor data

if (isnan(temperature) || isnan(humidity)) {

Serial.println("⚠️ Invalid DHT11 reading. Skipping.");

return;

}

int gasValue = analogRead (GAS\_SENSOR);

long distance = readUltrasonicDistance ();

//output to serial monitor

entryCount++;

Serial.println("\n--- Entry #" + String(entryCount) + " ---");

Serial.println("Temp: " + String(temperature) + "°C");

Serial.println("Humidity: " + String(humidity) + " %");

Serial.println("Gas Level: " + String(gasValue)+"%");

Serial.println("Distance: " + String(distance) + " cm");

// Display Smart E-uniform

lcd. clear ();

lcd. setCursor (0, 0); lcd.print ("SMART E-UNIFORM");

lcd. setCursor (0, 1); lcd.print("FOR SOLDIERS");

delay (2000);

//LCD: Temperature and Humidity

lcd. clear ();

lcd. setCursor (0, 0); lcd.print ("Temp:" + String(temperature) +" C ");

lcd. setCursor (0, 1); lcd.print ("Humidity:" + String(humidity) + "%");

delay (3000);

// LCD: Gas and Distance

lcd. clear ();

lcd. setCursor (0, 0); lcd.print ("Gas level:" + String(gasValue)+"%");

lcd. setCursor (0, 1); lcd.print ("Distance:" + String(distance) + "cm");

delay (3000):

// 🔁 Hybrid Alert Flags

static bool highTempAlertSent = false;

static bool lowTempAlertSent = false;

static int highTempAlertCount = 0;

static int lowTempAlertCount = 0;

static int lastHighAlertTemp = 0;

static int lastLowAlertTemp = 0;

// ⚠️ Check for high temperature

if (temperature >= 34.0) {

digitalWrite (RELAY1, LOW); // Fan ON

digitalWrite (RELAY, LOW) ;// Peltier OFF

client. publish ("soldier", "HIGH TEMPERATURE", true);

Serial.println("HIGH TEMP: Hydrate Mode on");

lcd. clear ();

lcd. setCursor (0, 0); lcd.print ("HIGH TEMP");

lcd. setCursor (0, 1); lcd.print ("Hydrate Mode ON");

if (! highTempAlertSent ||

(highTempAlertCount < 4 && temperature >= lastHighAlertTemp + 2)) {

String sms = "🔥 High Temp Alert! \nTemp: " + String(temperature) +

"°C\nHumidity: " + String(humidity) +

"%\nGas: " + String(gasValue) +

"%\nDistance: " + String(distance) + " cm";

sendSMS(sms);

// Send Email

EMailSender::EMailMessage message;

message. subject = "Smart Uniform Alert!";

message. message = "⚠️ High Temperature Detected! \n\n" +

String ("Temperature: ") + temperature + " °C\n" +

String ("Humidity: ") + humidity + " %\n" +

String ("Gas Level: ") + gasValue + "%\n" +

String ("Distance: ") + distance + " cm\n";

EMailSender::Response resp = emailSend.send("p.prakash292002@gmail.com", message);

if (resp. status) {

Serial.println("Email sent successfully.");

} else {

Serial.println("❌ Email failed:");

Serial.println("Error code: " + String (resp. code));

Serial.println("Desc: " + resp. desc);

}

lcd. clear ();

lcd. setCursor (0, 0);

lcd.print (resp. status? "Email Sent”: "Email Failed");

// Send SMS

//sendSMS(sms);

highTempAlertSent = true;

lastHighAlertTemp = (int)temperature;

highTempAlertCount++;

delay (3000);

}

// Reset low side

lowTempAlertSent = false;

lowTempAlertCount = 0;

lastLowAlertTemp = 0;

}

else if (temperature <= 28.0) {

digital Write (RELAY1, LOW); // Fan OFF

digital Write (RELAY, HIGH); // Peltier ON

client. Publish ("soldier", "LOW TEMPERATURE", true);

Serial.println("LOW TEMP: Warm Mode");

lcd. clear ();

lcd. setCursor (0, 0); lcd.print ("LOW TEMP");

lcd. setCursor (0, 1); lcd.print ("Warm Mode ON");

if (! lowTempAlertSent ||

(lowTempAlertCount < 4 && temperature <= lastLowAlertTemp - 2)) {

String sms = "❄️ Low Temp Alert! \nTemp: " + String(temperature) +

"°C\nHumidity: " + String(humidity) +

"%\nGas: " + String(gasValue) +

"%\nDistance: " + String(distance) + " cm";

sendSMS(sms);

// Send Email

EMailSender::EMailMessage message;

message. Subject = "Smart Uniform Alert!";

message. Message = "⚠️ Low Temperature Detected! \n\n" +

String ("Temperature: ") + temperature + " °C\n" +

String ("Humidity: ") + humidity + " %\n" +

String ("Gas Level: ") + gasValue + "%\n" +

String ("Distance: ") + distance + " cm\n";

EMailSender:: Response resp = emailSend.send("p.prakash292002@gmail.com", message);

if (resp. status) {

Serial.println("Email sent successfully.");

} else {

Serial.println("❌ Email failed:");

Serial.println("Error code: " + String (resp. code));

Serial.println("Desc: " + resp. desc);

}

lcd. setCursor (0,0);

lcd.print (resp. status? "Email Sent OK”: "Email Failed");

// Send SMS

//sendSMS(sms);

lowTempAlertSent = true;

lastLowAlertTemp = (int)temperature;

lowTempAlertCount++;

delay (3000);

}

// Reset high side

highTempAlertSent = false;

highTempAlertCount = 0;

lastHighAlertTemp = 0;

}

else {

// Normal temperature — reset everything

highTempAlertSent = false;

lowTempAlertSent = false;

highTempAlertCount = 0;

lowTempAlertCount = 0;

lastHighAlertTemp = 0;

lastLowAlertTemp = 0;

digital Write (RELAY1, HIGH); // Fan OFF

digital Write (RELAY, LOW); // Peltier OFF

}

// Gas alert

if (gas Value > 500) {

client. Publish ("soldier", "HIGH GAS LEVEL", true);

lcd. clear (); lcd. setCursor (0, 0); lcd.print ("GAS LEVEL HIGH");

}

// Send to Thing Speak

SendToThing Speak (temperature, humidity, gas Value, distance);

delay (15000) ;// Minimum delay for Thing Speak

}

**CHAPTER 7**

**TESTING**

**7.1 TESTING METHODOLOGY**

Testing methodology refers to the standardized practices and processes followed during the validation of the Smart E-Uniform system to ensure its accuracy, stability, and responsiveness in real-time field conditions. In IoT-based systems like this, where hardware and software interact continuously, structured testing documentation plays a vital role in tracking system performance, ensuring interoperability, and enabling future debugging.

Testing documentation includes all artifacts created during the verification and validation stages ranging from test plans, test data, and test cases, to traceability matrices and execution logs. These documents are crucial not just for the development team, but also for stakeholders and end-users, as they reflect the quality and reliability of the system.

In the context of this project, the testing methodology combines both **manual and automated testing techniques,** using real sensor inputs, simulated data, and various threshold-trigger scenarios (such as high/low temperature and gas detection).

**Benefits of Testing Documentation**

* **Ensures Quality and Clarity**  
  Documents clarify system behaviour, sensor responses, and alert outputs under different environmental conditions.
* **Facilitates Team Coordination**  
  With multiple modules (sensors, GSM, Wi-Fi, LCD, etc.), documentation helps coordinate efforts between developers, testers, and network engineers.
* **Enhances Maintainability and Debugging**  
  Future teams can trace issues using detailed logs, test cases, and requirement mappings without having to repeat the entire testing cycle.
* **supports Stakeholder Communication**  
  Provides a transparent overview of which features have been validated and how performance matches expectations.

**Test Strategy**

The test strategy defines the scope, objectives, and overall approach to testing. This IoT-based system integrates multiple sensors (DHT11, gas sensor, ultrasonic sensor), output modules (LCD, fan, Peltier), and communication protocols (Wi-Fi, SMTP, Twilio, MQTT). Testing focuses on:

* Functional testing of sensor inputs and system responses
* Integration testing of communication modules
* Interface testing of the LCD and ThingSpeak dashboard
* Reliability testing of the alert system (under simulated temperature and gas events).

### **Test Data**

Test data was created to simulate real-world operational scenarios. Since physical sensor readings can vary, controlled and dummy values were introduced to check system behaviour and trigger alerts. Some examples include:

**TABLE** 7.1 Test Data

| **Scenario** | **Test Data** |
| --- | --- |
| High Temperature | 36°C input from DHT11 |
| Low Temperature | 25°C input from DHT11 |
| Gas Detection | Simulated LPG presence via analog value > threshold |
| Distance Alert | Ultrasonic sensor reading < 50 cm |
| SMTP Email Test | Email triggered on 35°C |
| Twilio SMS Test | Alert message with full sensor data sent to verified number |
| MQTT Test | Temperature published on HiveQL Broker |

**Test Plan**

The test plan covers resource allocation, schedule, environments, and limitations:

* **Testing Resources**: ESP8266 module, sensors, Wi-Fi access, mobile phone (for SMS), Gmail SMTP account
* **Environment**: Lab environment with stable power, secured Wi-Fi, controlled heat/gas for testing
* **Tools**: Arduino IDE, Serial Monitor, ThingSpeak Dashboard, Twilio Console, HiveQL Broker
* **Test Duration**: 2 weeks of modular testing + 1 week of full-system validation.

**Test Scenario**

* The Smart E-Uniform was tested through multiple scenarios to ensure reliable performance of each module. For temperature monitoring, the DHT11 sensor was tested to detect values beyond the threshold (above 34°C or below 28°C) and trigger appropriate alerts while updating the LCD display. Gas detection was validated using the MQ-2 sensor, which successfully identified gas presence and initiated emergency alerts.
* Distance monitoring was tested with the ultrasonic sensor to detect when a soldier moves beyond 150 cm from the base station, activating a proximity alert.
* Alert mechanisms were tested independently. MQTT alerts were published to HiveQL, email alerts were sent via Gmail SMTP with sensor values, and Twilio was used to send SMS alerts during emergencies. Sensor failure handling was tested by simulating missing data, where the system correctly displayed an error without crashing.
* LCD functionality was validated to display updated sensor data every 15 seconds. Finally, internet disconnection scenarios confirmed that the system retries sending alerts when the connection is restored, ensuring reliable and continuous operation under real-world conditions.

**Test Case**

Test cases were written for each scenario with detailed steps, expected results, and actual results. Sample test case:

#### **Test Case: High Temperature Alert Trigger**

* **Objective**: Verify that when temperature exceeds 34°C, MQTT, Email, and SMS alerts are sent
* **Input**: Simulate temperature = 35.5°C
* **Steps**:
  1. Power on system
  2. Set temperature input in code (or heat sensor manually)
  3. Observe LCD, Serial Monitor, and alert logs
* **Expected Output**:
  + MQTT data sent to HiveQL
  + Email received in inbox
  + SMS received with correct format
* **Actual Output**: All alerts triggered successfully

**Traceability Matrix**

A traceability matrix mapped each system requirement to a corresponding test case. For example, high temperature detection was linked to an alert test case validating MQTT, email, and SMS delivery. Each test confirmed the correct implementation of system features, with results showing all modules functioned as expected. This ensured all functional and safety goals of the smart uniform were fully met before deployment.

**CHAPTER 8**

**PERFORMANCE AND LIMITATION**

**8.1 PERFORMANCE**

* The Smart E-Uniform accurately monitors vital environmental parameters such as temperature, gas presence, and proximity, ensuring real-time situational awareness for soldiers.
* Emergency alert mechanisms, including MQTT publishing, email alerts via SMTP, and SMS alerts using Twilio, function reliably during abnormal conditions.
* The system is capable of dynamically controlling cooling and heating modules (fan and peltier panel) based on the soldier’s body temperature, maintaining comfort and safety.
* The LCD display provides real-time updates of sensor values, enhancing user interaction and field readability.
* Integration with cloud platforms like ThingSpeak allows remote data visualization and performance tracking for base station monitoring.
* The project ensures that only one actuator (fan or peltier) operates at a time, optimizing power usage and thermal efficiency.
* The uniform is suitable for deployment in training simulations and field environments, acting as a proactive health and safety system.

**8.2 LIMITATION**

* The system relies on continuous internet connectivity for cloud-based alerts; disruptions may delay notifications.
* Sensor accuracy can be affected by extreme environmental conditions, dust, or prolonged exposure to harsh weather.
* Twilio SMS service may be limited by geographic or network restrictions in remote areas.
* The prototype is limited to basic temperature and gas detection and does not include advanced biomedical monitoring.
* Battery-powered operation may require regular charging or power optimization for extended use in the field.

**8.3 CONCLUSION AND FUTURE ENHANCEMENT**

The **Smart E-Uniform for Soldiers using IoT** was successfully designed and implemented to monitor key environmental and physical parameters essential for soldier safety and performance in real-time. The system integrates temperature, gas, and distance sensors with wireless communication protocols to detect emergency conditions and trigger alerts through MQTT, Email, and SMS. The use of IoT technologies provides enhanced situational awareness and immediate feedback to both the soldier and the command base station. Overall, the system demonstrates reliability, real-time responsiveness, and practical applicability for military use. It also has the potential to be extended to other high-risk professions and industries where environmental monitoring is critical.

In future, it may include real-time video monitoring using an ESP-32 camera and precise location tracking with a NEO-M9N GPS module. i**ntegration of biomedical sensors** such as heart rate, ECG, and hydration level monitoring for comprehensive soldier health tracking. **Encrypted communication protocols** to enhance the security of transmitted data in military operations. **Mobile app development** for field commanders to receive live alerts and data visualization directly on smartphones or tablets. **Voice alert system** for soldiers to receive immediate warnings even without looking at the LCD.

**CHAPTER 9**

**APPENDICES**

**9.1 SAMPLE SCREENS**

****

**Figure 9.1 WELCOME SCREEN**

****

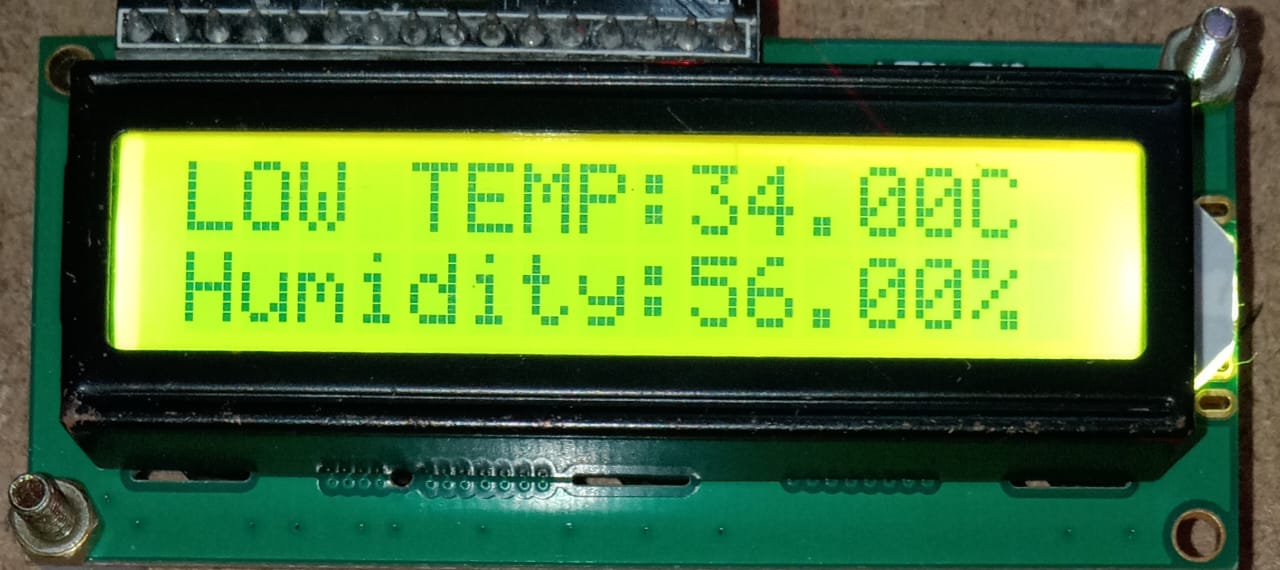
**Figure 9.2 Real-time distance data**

****

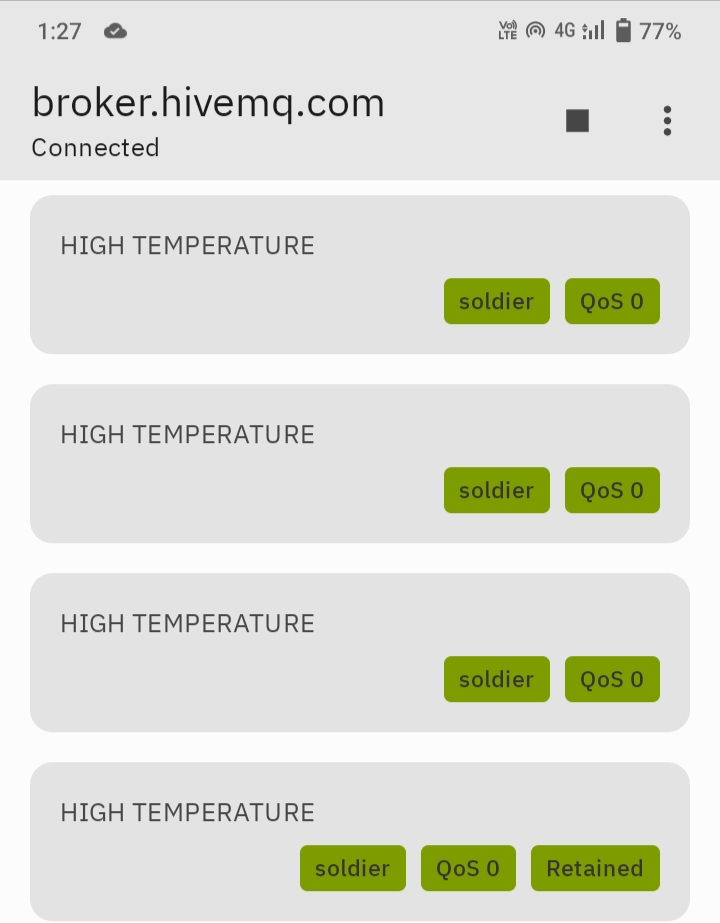
**Figure 9.3 Toxic gas detection screen**

****

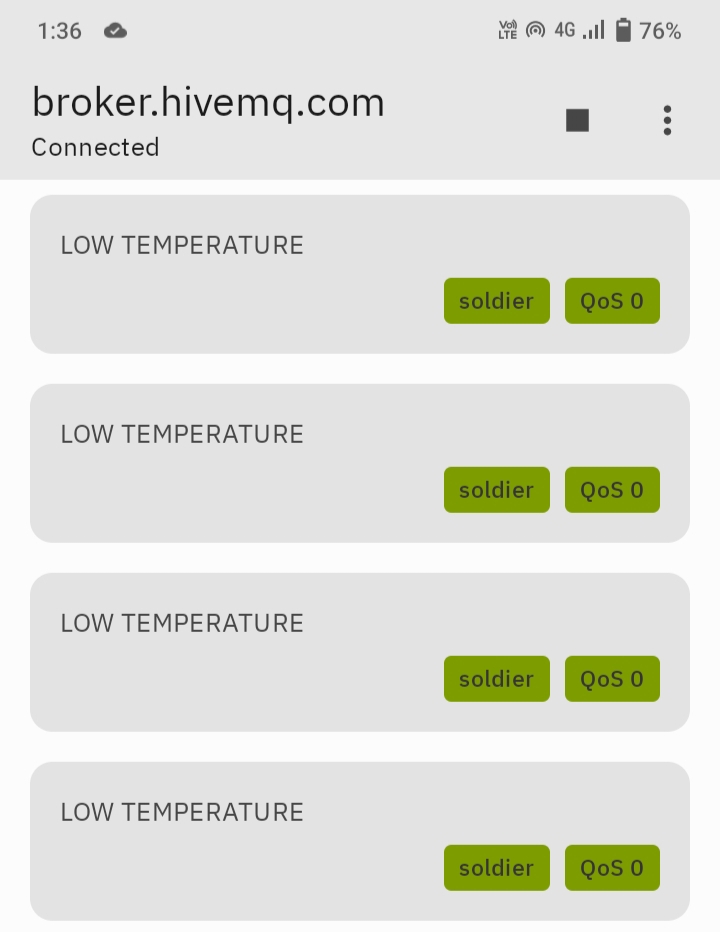
**Figure 9.4 Real-time Temp & Humidity data (High)**

****

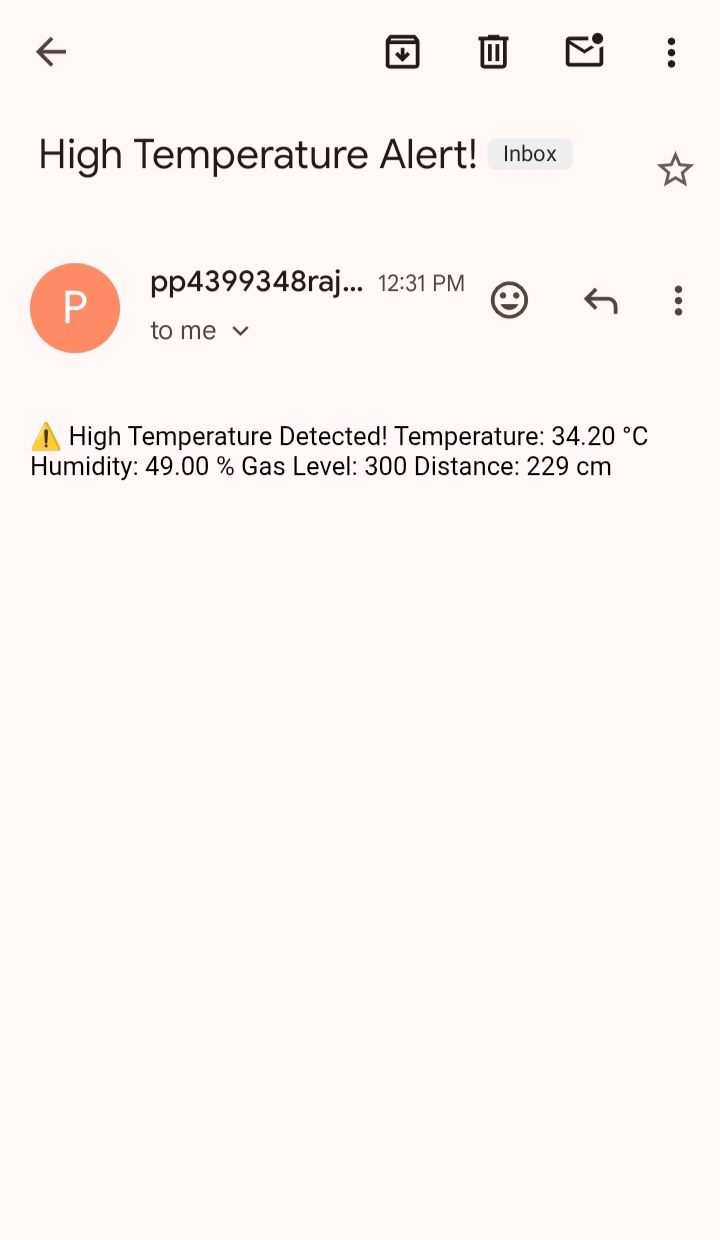
**Figure 9.5 Real-time Temp & Humidity data (Low)**

****

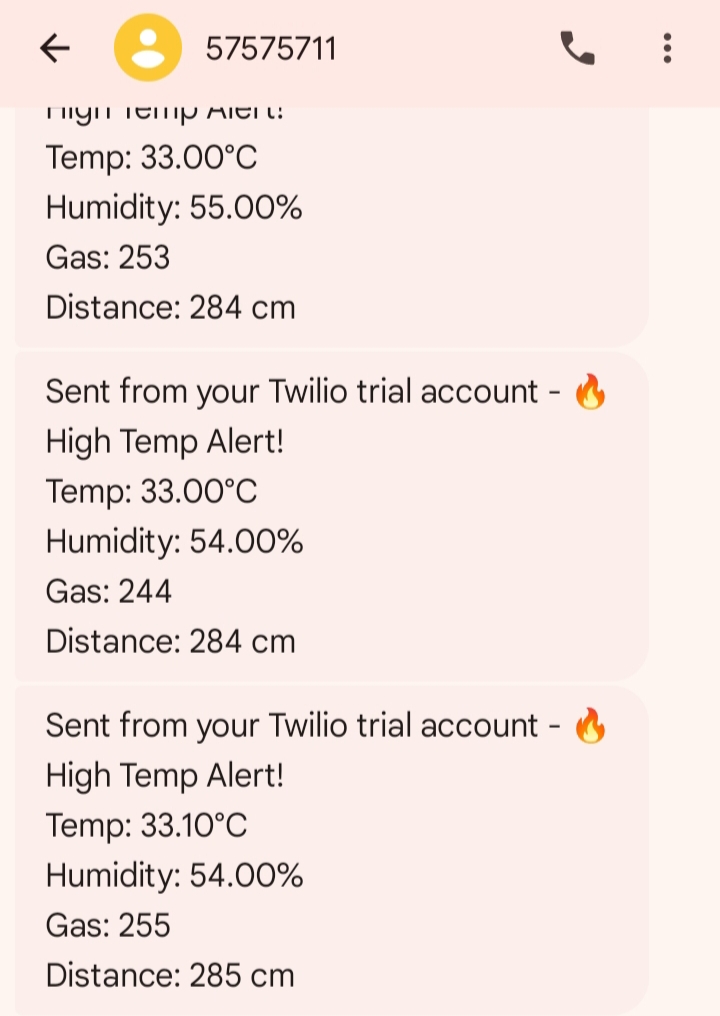
**Figure 9.6 MQTT Alert for High Temperature**

****

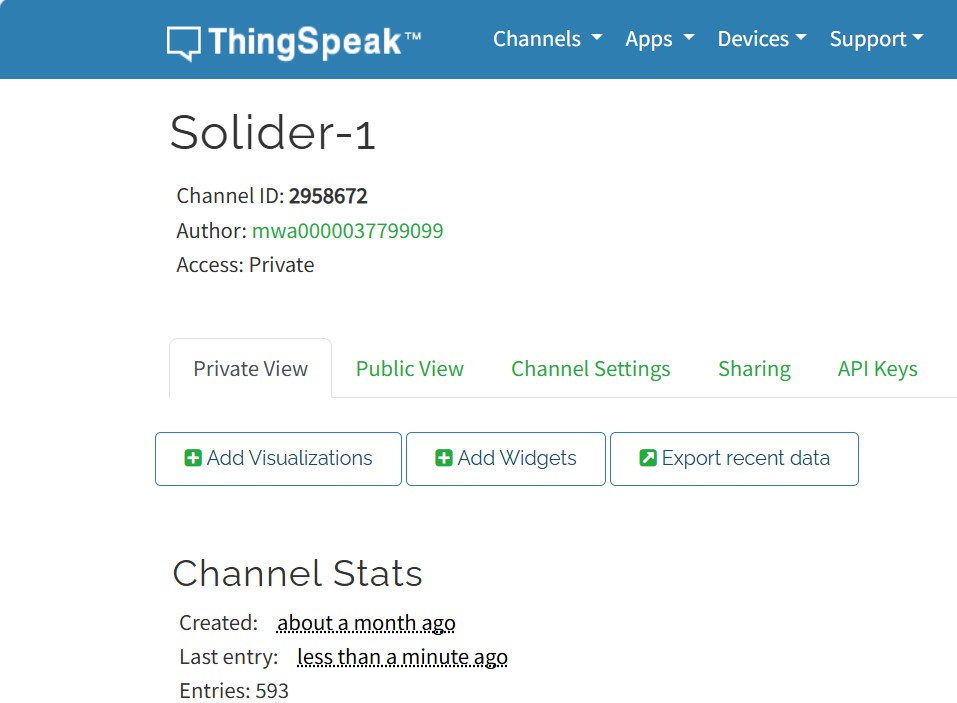
**Figure 9.7 MQTT Alert for Low Temperature**

****

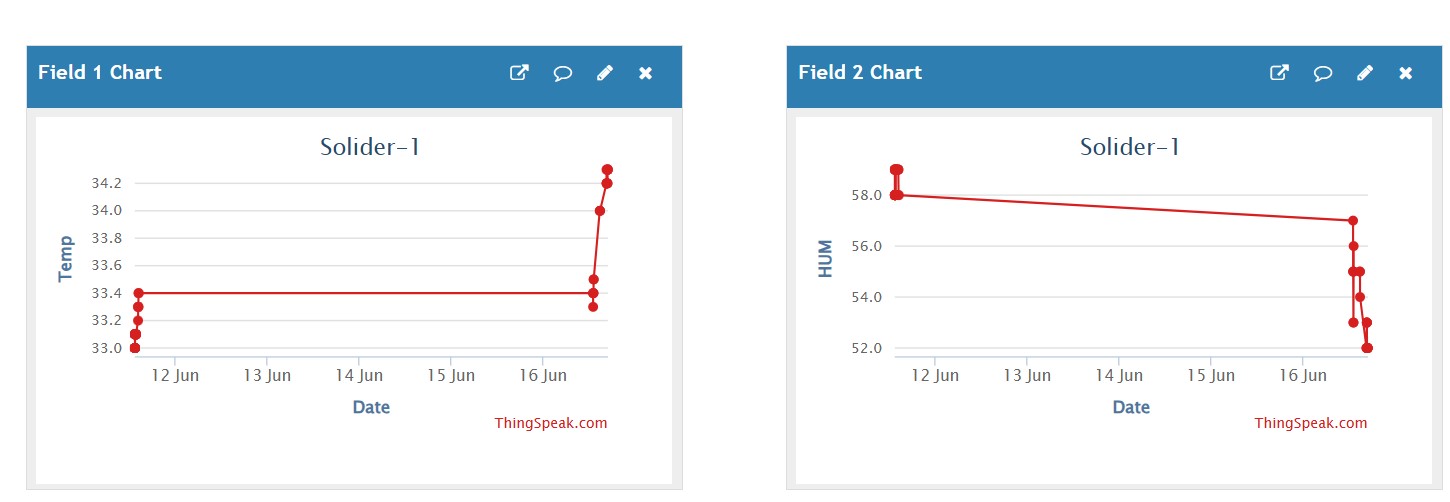
**Figure 9.8 E-mail Alert**

****

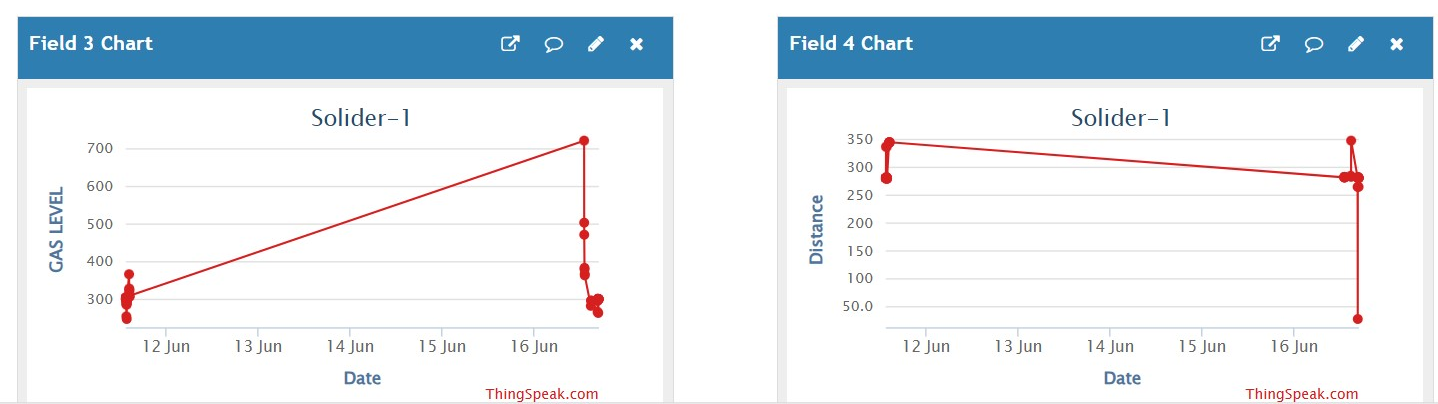
**Figure 9.9 SMS Alert**

****

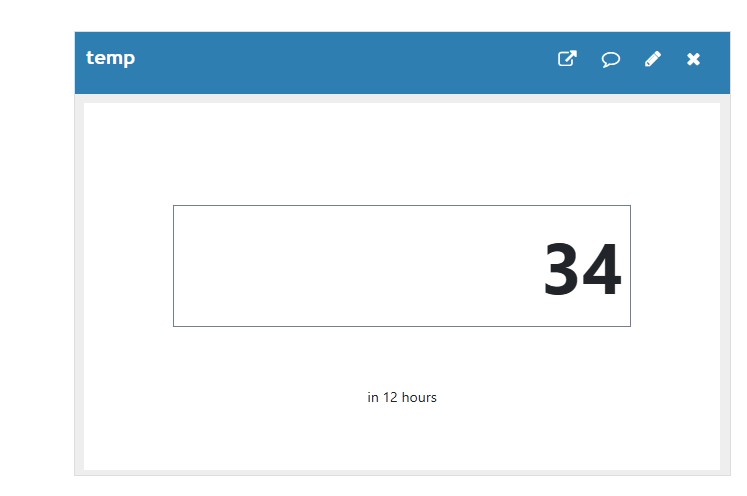
**Figure 9.10 ThingSpeak dashboard of soldier**

****

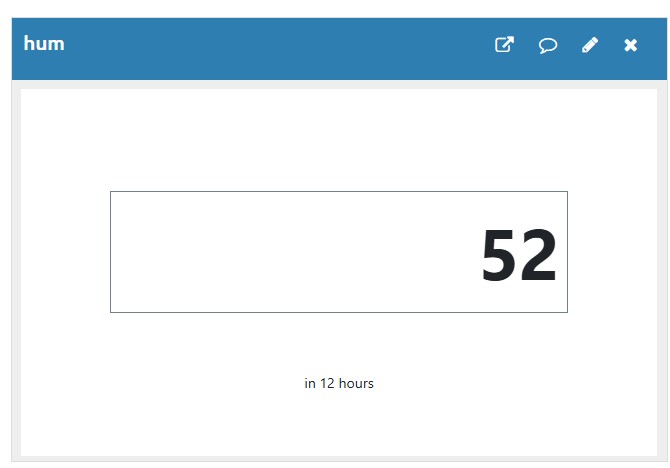
**Figure 9.11 ThingSpeak-based visualization for Temp & Hum**

****

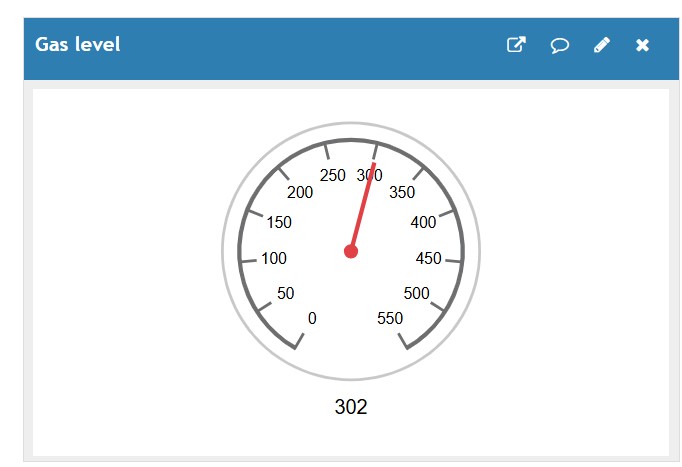
**Figure 9.12 ThingSpeak-based visualization for Gas level & Distance**

****

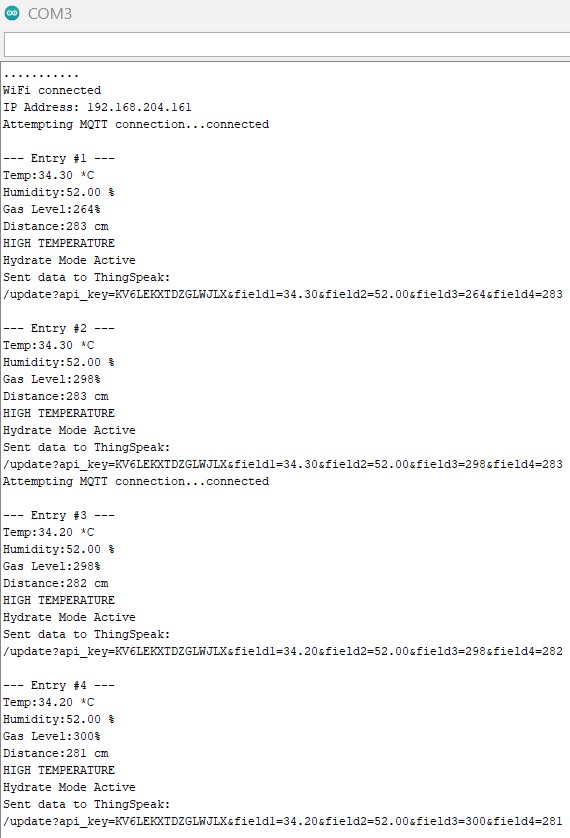
**Figure 9.13 ThingSpeak widgets of Temperature sensor value**

****

**Figure 9.14 ThingSpeak widgets of Humidity value**

****

**Figure 9.15 ThingSpeak widgets of Gas Sensor**

**Figure 9.16 Serial Monitor Output**

**REFERENCES**

[1] Vignesh. A; C. Bhuvaneswari; Shiva M; W. Abitha Mamala; M. Pushpavalli; M. Kavitha, Intelligent Solar Based Climate Adjustable E-Uniform for Soldiers, 2023 International Research journal on Advances in Computing, Communication and Applied Informatics.

[2] Jeeva B; Bangi Karuna Sree; Basineni Udaya Priyanka; Divyadarshini S; Hema. S, Smart E-uniform for Soldiers, 2023 7th International Research Journal on Design Innovation for 3 Cs Compute Communicate Control.

[3] S. Ramesh, Dr R. Sankar Ganesh, N. Dharani and P. Kandeepan, "Climate Adjustable E- Military Suit", International Research Journal of Modernization in Engineering Technology and Science, March 2023.

[4] Chowdam Anila; Ashok Kumar. N; Eduru Joel Vittan; Bheemaneni UmaMaheswara Chowdary; Dhanya Sree Bala Subramani; Balamanikandan. A, Sun-Powered Tactical Uniforms for Soldiers in Military Applications, 2024 10th International Journal on Advanced Computing and Communication Systems.

[5] Rahul Khairamode, Rahul Khangouka and Rutik Patil, "Design and Fabrication of Solar Based E-Jacket for Soldiers", International Journal of Engineering Technology and Management Sciences, July 2022.

[6] Dipali H. Kale, Yogita P. Phapale, S. Kamble Shivani and S. L. Shirke, "Solar Based E-Uniform for Soldiers", International Research Journal of Engineering and Technology (IRJET), May 2022.

[7] Phase Change Materials for Thermal Regulation in Military Clothing Cho, Y., Kim, H., & Lee, J International Research Journal of Engineering and Technology (IRJET) (2023).

[8] Gas Sensor-Based Environmental Monitoring for Soldier Safety by Patel, V., Sharma, B., & Khan, An International Research Journal of Engineering and Technology (IRJET) (2022).

[9] Infrared Sensor Applications in Low-Visibility Environments. Journal of Sensor Technology Kim by D., Park, S., & Choi, M. International Research Journal of Engineering and Technology (IRJET) (2022).

[10] M. Vasanthi, M. Suganya, S. Anitha, S. Ashma, K. Janani, M. Kaviyasree, E-Smart Army Jacket Based on Climate Condition Using Arm Microcontroller 2022 International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET).

[11] Mrs. Geetha B1, Shreya B M2, Sushma R3, Jayashree D P4, Soundarya K R5 “Implementation of solar based e uniform for soldiers” International Journal of Advanced Research in Computer and Communication Engineering (2024).

[11] A. Abi Zeid Daou, C. Haddad and R. Abi Zeid Daou, "Design and Implementation of a Smart Soldier Uniform",2022 International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET).

[12] Prof. J.V. Satre, Sudarshan B. Masure, Sujit N. Bhand: dept. Assistant Professor, Department of Electrical Engineering) KJ’s Trinity College of Engineering & Research Pune, India IOT and solar based E-uniform for soldiers with real time tracking International Journal of Creative Research Thoughts (IJCRT) 2023.

[13] Ms. Sumalata Kadire, Spandana, J. Shravya, K. Sravani Smart Solar-driven E-uniform with climate control for soldiers International Journal of Advanced Research in Computer and Communication Engineering (2022).

[14] Rakshitha M, Shreyas R, Yeshas C Balaji, Rakshitha Mrs. S Smart Soldier Jacket Using Internet of Things (IoT) International Research Journal of Engineering and Technology (IRJET) 2022.

[15] “IOT based health and position tracking system for soldier security system” rajitha m, s. madhav rao 1. vol 13, issue 06, june,2022.

[16] Jasvinder Singh Chhabra1, Akshay Chhajed1, Shamlee Pandita1, Suchita Wagh2 “GPS And IoT Based Soldier Tracking & Health Indication System” International Research Journal of Engineering and Technology Volume: 04 Issue: 06 | June-2021.

[17] Mrs. Pallavi Kulkarni and Mrs. Tripti Kulkarni, “Secure health monitoring of soldiers with tracking system using IOT”, International Journal of Trend in Scientific Research and Development (IJTSRD), Volume No:3, Issue: 4, pp:693-696, May June-2021.

[18] E. J. Shanko and M. G. Papoutsidakis, “Real time health monitoring and wireless transmission: A controller application to improve human medical needs”, International Research Journal of Engineering and Technology (IRJET) 2021.