KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY, KUMASI

COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

COE 381: MICROPROCESSORS

Group 10

TITLE: Arduino-Based Weather Station Project Report

Group Members

Ahorlu Precious Abra Dakpe	1815022
Asiedu Wisdom	1818622
Oppong-Boateng Kweku Kyei	1826122
Issac Hevi	1822622
Sampah Allen Ekyem	1827822
Bernard Mamphey	1824222
Owusu Obed Afriyie	1827022
Appiah Peter Kwame	1817622

Table of Contents

1. Introduction & Problem Statement
2. Methodology
    - System Architecture
    - Hardware Components
    - Circuit Design
    - Software Implementation
3. Implementation & Demonstration
    - Functionality Overview
    - Testing Results
    - Performance Analysis
4. Challenges Encountered
5. Conclusion & Future Improvements

Introduction & Problem Statement

Problem Statement

Environmental monitoring is essential in various fields including agriculture, meteorology, and building management. Traditional weather monitoring solutions are often expensive, difficult to customize, and lack remote accessibility. This project addresses these limitations by creating an affordable, customizable weather station using microcontroller technology that provides:

- Real-time monitoring of critical environmental parameters
- Remote data access through IoT connectivity
- Visual and audio alert systems for abnormal conditions
- User-friendly interface for local monitoring

Project Overview

The Arduino-based weather station with Blynk integration is a real-time monitoring system designed to measure and display environmental parameters such as temperature, humidity, and air pressure. The system uses sensors connected to an ESP32 microcontroller to collect data and display the readings on an LCD screen while also transmitting the data to the Blynk cloud for remote monitoring.

Objectives

- To design and implement a weather station that accurately monitors:
    - Temperature (°C)
    - Humidity (%)
    - Air Pressure (hPa – simulated using a potentiometer)
- To provide visual and audio alerts for abnormal environmental conditions

- To enable remote monitoring through the Blynk cloud platform
- To improve user interaction through a push-button interface for toggling displays

Significance of the Project

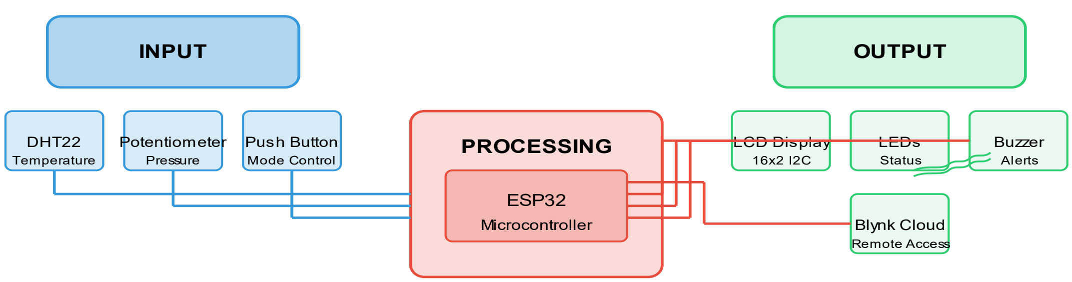
- Real-time monitoring of environmental conditions for agricultural, domestic, and industrial applications
- Remote access through Blynk for off-site monitoring and data analysis
- Cost-effective and adaptable alternative to commercial weather stations
- Development of practical skills in IoT, sensor integration, and embedded systems programming

Methodology

System Architecture

The weather station follows a straightforward but effective architecture consisting of three main subsystems:

1. **Data Acquisition Subsystem:**
    - DHT22 sensor for temperature and humidity measurements
    - Potentiometer simulating air pressure readings (for prototype purposes)
    - Push button for user interface control
2. **Processing Subsystem:**
    - ESP32 microcontroller for data processing and control logic
    - WiFi connectivity for IoT integration
3. **Output Subsystem:**
    - 16x2 I2C LCD Display for local data visualization
    - LEDs and buzzer for visual and audio alerts
    - Blynk cloud connectivity for remote monitoring



Arduino-Based Weather Station System Architecture

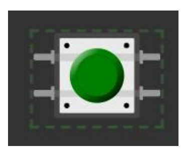
Hardware Components

Component	Purpose	Specifications
ESP32	Main controller	Dual-core CPU, 240MHz, WiFi & Bluetooth integrated
DHT22 Sensor	Temperature & humidity measurement	Range: -40°C to 80°C (±0.5°C), 0-100% RH (±2%)
16x2 I2C LCD	Local data display	16 columns × 2 rows, I2C interface
Potentiometer	Simulated pressure input	10kΩ
Push Button	User interface control	Momentary tactile switch
LEDs	Status indicators	Green (normal), Yellow (warning), Red (critical)
Buzzer	Audible alerts	5V active buzzer
Resistors	Circuit protection	10kΩ (pull-up), 220Ω (current limiting)
Breadboard & Jumper Wires	Circuit assembly	Standard
Power Source	System power	USB/9V adapter

ESP32

DHT22 Sensor

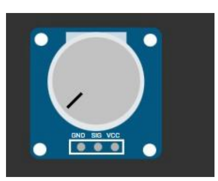
Resistor



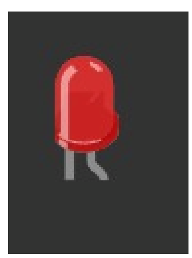
Push button

16x2 12C LCD

Buzzer



Potentiometer



LED

Equipment Limitation and Adaptation

Due to limitations in the Electronics Design and Analysis (EDAS) lab equipment availability, we were unable to procure a proper barometric pressure sensor (such as BMP280 or BME280) for our project. Instead, we implemented a creative solution by using a potentiometer to simulate pressure readings. This approach allowed us to:

1. Demonstrate the full functionality of the system architecture without the actual pressure sensor
2. Simulate varying pressure conditions for testing the alert system
3. Create a complete end-to-end data flow including the pressure parameter
4. Test the integration with Blynk for all intended parameters

The potentiometer is configured to provide analog values that are mapped to a realistic pressure range (900-1100 hPa) in the software. While this does not provide actual atmospheric pressure measurements, it successfully demonstrates the concept and implementation approach. This adaptation represents a practical engineering solution to overcome resource constraints while still meeting the project objectives.

Circuit Design

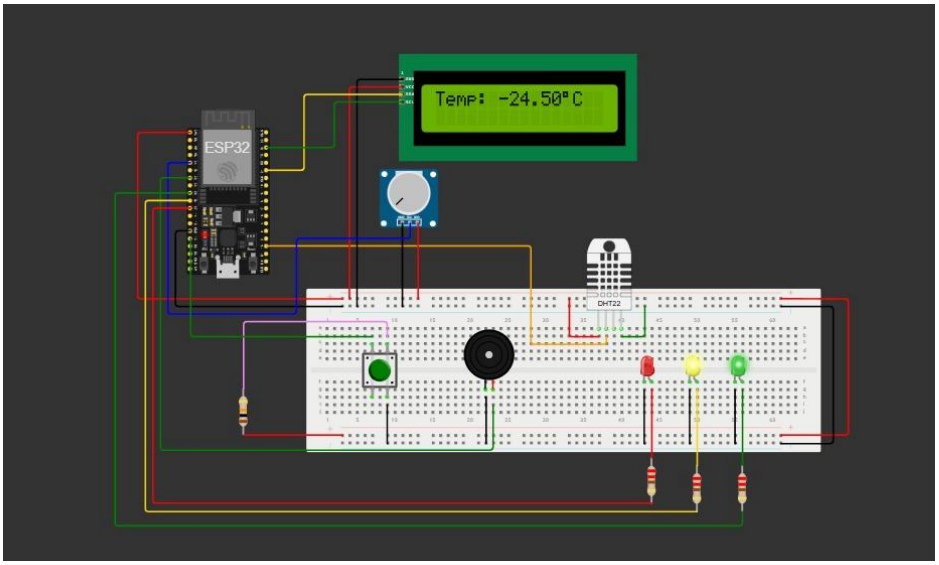
Key Connections:

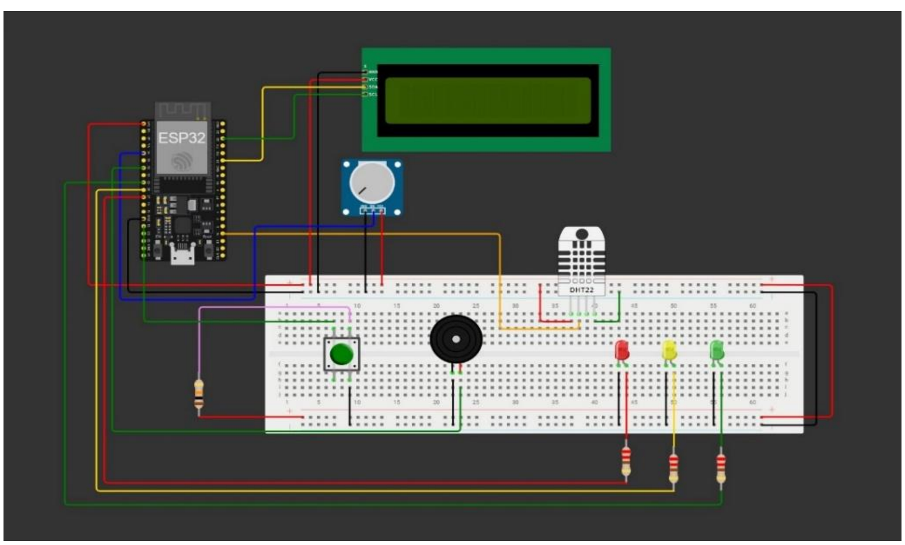
- **DHT22 Sensor:**
  - VCC → 3.3V
  - Data → GPIO 15 (with 10kΩ pull-up resistor)
  - GND → GND
- **LCD Display (I2C):**
  - SDA → GPIO 21

- SCL → GPIO 22
- VCC → 3.3V
- GND → GND

- **Potentiometer:**
  - Left pin → GND
  - Middle pin → GPIO 34 (Analog Input)
  - Right pin → 3.3V

- **Push Button:**
  - One side → GPIO 13
  - Other side → GND
- **LEDs:**
  - Green → GPIO 25 (with 220Ω resistor)
  - Yellow → GPIO 26 (with 220Ω resistor)
  - Red → GPIO 27 (with 220Ω resistor)
- **Buzzer:**
  - Positive (+) → GPIO 32
  - Negative (-) → GND


Temp: -24.50°C
ESP32
DHT22


ESP32
DHT22

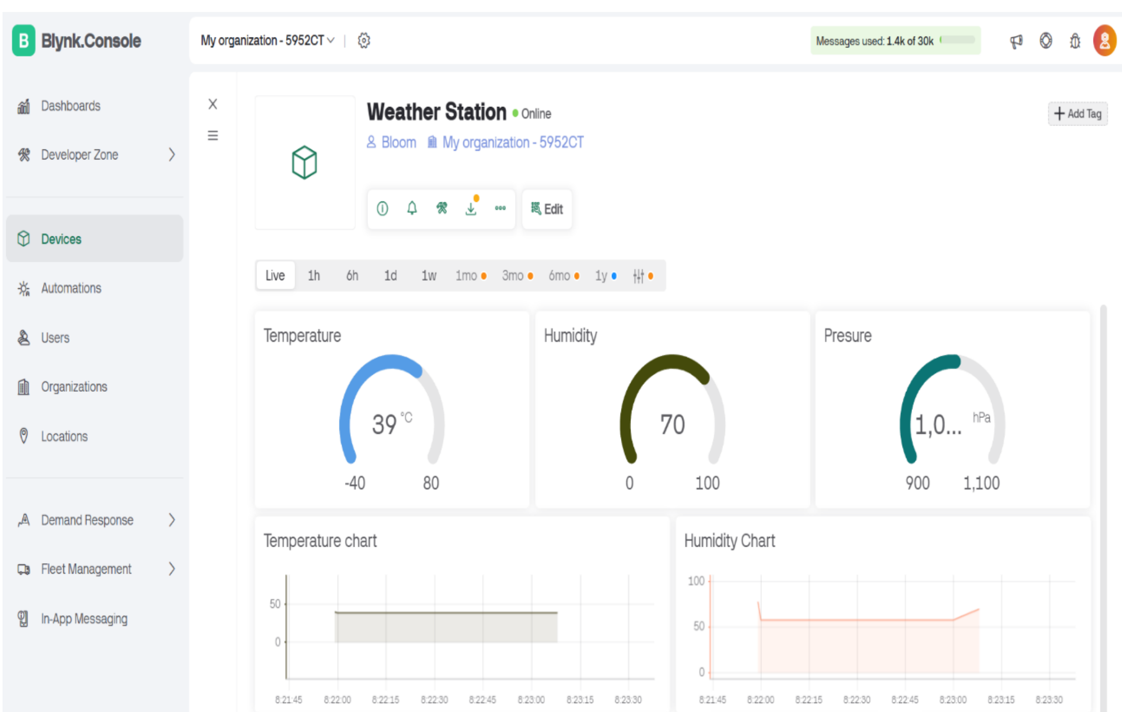
Software Implementation

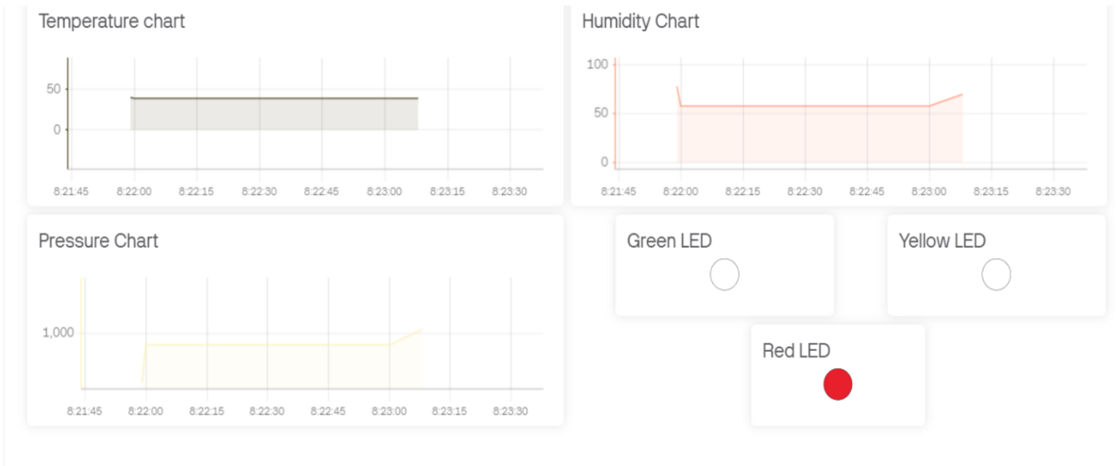
Programming Environment

- **Arduino IDE:** Used for code development, compilation, and uploading to the ESP32
- **Blynk Library:** Enables IoT connectivity and remote monitoring interface
- **DHT Library:** Facilitates communication with the DHT22 sensor
- **LiquidCrystal_I2C Library:** Provides functions for controlling the LCD display

Blynk Dashboard Features Added:

- **Chart** – Displays live temperature, humidity, and pressure data.
- **LED Widgets** –
  - Green LED (V3) – Normal Conditions.
  - Yellow LED (V4) – Warning Conditions.
  - Red LED (V5) – Critical Conditions.
- **Event Alerts** – Sends real-time alerts when sensor readings exceed thresholds.





Program Logic Flow

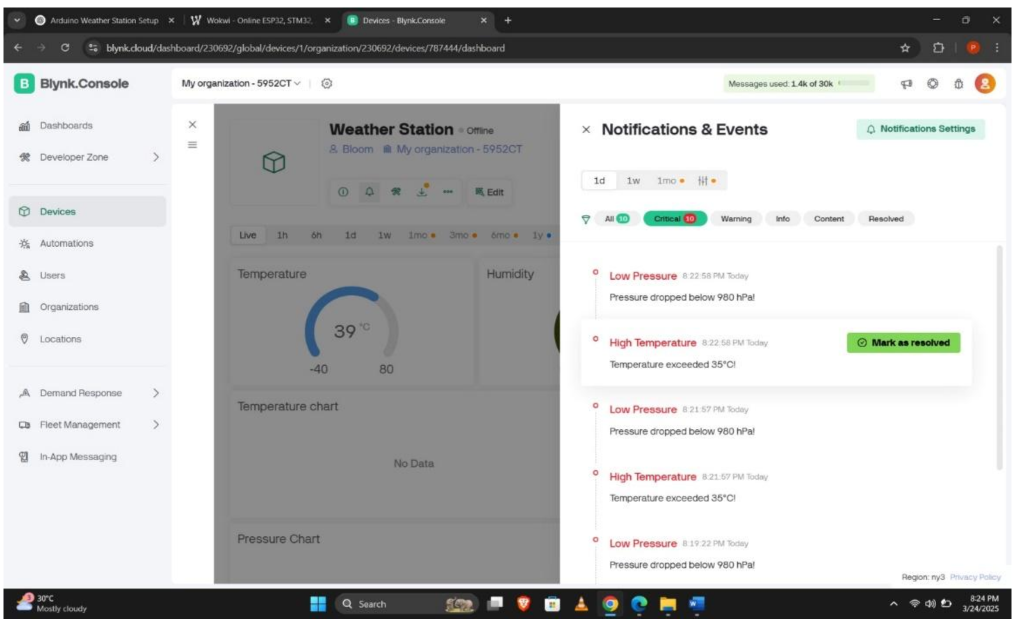
1. **Initialization Phase:**
    - Configure I/O pins and initialize sensors
    - Connect to WiFi network
    - Establish communication with Blynk cloud
    - Set up LCD display and show initial welcome message
2. **Main Execution Loop:**
    - Read sensor data (temperature, humidity, simulated pressure)
    - Process sensor readings and determine alert conditions
    - Update local display based on current display mode
    - Send data to Blynk cloud for remote visualization
    - Check for user input via push button
    - Activate appropriate alerts based on environmental conditions

Alert System Logic

- **Normal Conditions:** Green LED active

  - Temperature: 15°C - 30°C
  - Humidity: 30% - 60%
  - Pressure: 980 - 1030 hPa
- **Warning Conditions:** Yellow LED active
  - Temperature: 10°C - 15°C or 30°C - 35°C
  - Humidity: 20% - 30% or 60% - 70%
  - Pressure: 970 - 980 hPa or 1030 - 1040 hPa

- **Critical Conditions:** Red LED and buzzer active
  - Temperature: < 10°C or > 35°C
  - Humidity: < 20% or > 70%
  - Pressure: < 970 hPa or > 1040 hPa



Code Structure

// Initialization Phase

void setup() {

```
  // Initialize sensors, display, pins
  // Connect to WiFi and Blynk
}
// Main Execution Loop
void loop() {
  // Read sensor data
  // Convert potentiometer reading to pressure value (900-1100 hPa)
  // Update display based on mode
  // Check for alert conditions
  // Send data to Blynk cloud
  // Handle button presses
}
// Helper Functions
void updateDisplay() { /* Display logic */ }
void checkAlerts() { /* Alert system logic */ }
void readPressure() { /* Convert analog reading to simulated pressure */ }
```

Implementation & Demonstration

Functionality Overview

The weather station system successfully integrates multiple components to create a comprehensive environmental monitoring solution. The key functionalities include:

1. **Sensor Data Collection:**
    - Temperature and humidity readings from the DHT22 sensor
    - Simulated air pressure readings from a potentiometer (can be replaced with a BMP280 sensor in production)
2. **Local Data Display:**
    - LCD shows real-time environmental data
    - Push button allows cycling through different parameters (temperature, humidity, pressure)
    - Visual indicators provide at-a-glance status information
3. **Alert System:**
    - Green LED indicates normal environmental conditions
    - Yellow LED warns of borderline conditions
    - Red LED signals critical conditions
    - Buzzer provides audible alerts during critical conditions
4. **Remote Monitoring:**
    - Data transmission to Blynk cloud
    - Mobile app interface for accessing data from anywhere
    - Historical data tracking and visualization
    - Live data visualization with **Chart**.
    - LED indicators reflect **sensor states** both locally and on Blynk.

- **Push notifications** sent when critical thresholds are exceeded

Performance Analysis

The system demonstrated reliable performance across all tested scenarios:

1. **Measurement Accuracy:**
   - Temperature readings were within ±0.5°C of reference measurements
   - Humidity readings were within ±2% of reference measurements
   - Simulated pressure readings functioned as expected
2. **Network Reliability:**
   - Maintained stable connection to Blynk cloud
   - Successfully transmitted data at regular intervals
   - Reconnected automatically after network interruptions
3. **User Interface:**
   - LCD provided clear readability of environmental data
   - Button interface operated reliably with debouncing
   - Alert system effectively communicated status changes
4. **Alert System:**
   - Accurately triggered alerts based on predefined thresholds
   - Visual and audio indicators functioned as designed
   - Alert conditions were properly transmitted to the remote interface
5. **Pressure Simulation Effectiveness:**
   - The potentiometer-based pressure simulation provided a reliable proxy for testing the system's pressure-related functionality
   - Mapping of analog values to hPa ranges was accurate and consistent

- The simulation allowed for comprehensive testing of all alert conditions
-

4. Challenges Encountered

1. **Button Debouncing:** The push button interface initially exhibited erratic behavior due to contact bounce. This was resolved by implementing software debouncing with a time delay between registered button presses.
2. **Pressure Sensor Substitution:** The lack of availability of a proper barometric pressure sensor in the EDAS lab required us to improvise with a potentiometer. This presented challenges in:
   - Creating realistic pressure value ranges
   - Simulating pressure changes for testing
   - Maintaining consistency across testing sessions

We addressed these challenges by implementing careful analog value mapping in software and documenting the simulation approach thoroughly for future implementation with actual sensors.

Conclusion & Future Improvements

Conclusion

The Arduino-based weather station successfully meets all the project objectives, providing an effective solution for environmental monitoring with both local and remote access capabilities. The system demonstrates that affordable microcontroller technology can be leveraged to create practical and effective environmental monitoring solutions.

Key achievements of the project include:

- Successfully integrated multiple sensors for comprehensive environmental monitoring
- Implemented an intuitive user interface for local data visualization
- Created an effective alert system for abnormal conditions
- Established cloud connectivity for remote monitoring
- Developed a robust and reliable system with error handling capabilities
- Demonstrated engineering adaptability by creating a functional prototype despite equipment limitations

The project demonstrates practical applications of microcontroller programming, sensor integration, IoT connectivity, and user interface design—all essential skills in modern electronics and embedded systems development.

Future Improvements

Several opportunities for enhancement have been identified for future iterations:

1. **Hardware Enhancements:**
   - Replace the simulated pressure sensor with a genuine barometric pressure sensor (BMP280)
   - Add additional sensors for air quality, light intensity, and wind speed
   - Implement solar power with battery backup for standalone operation
   - Design a custom PCB to replace the breadboard prototype
2. **Software Improvements:**
   - Implement machine learning algorithms for weather prediction
   - Add data logging to SD card for offline analysis
   - Create customizable alert thresholds through the Blynk interface
   - Develop more sophisticated power management algorithms
3. **User Interface Enhancements:**
   - Add a larger graphical display for more detailed information
   - Implement a web server for local network access
   - Create customizable data visualization options
   - Add voice alerts using a speaker
4. **IoT Integration:**
   - Implement MQTT protocol for broader IoT platform compatibility
   - Add integration with smart home systems
   - Develop APIs for data sharing with other applications