FRENCH UNIVERSITY IN ARMENIA

Faculty of Computer science and Applied mathematics

Subject Project S1

Student Gor BABAYAN **Student Number** 202305013

Teacher Narek MARGARYAN

December 19, 2024

1. Summary

1. Brief overview of the project objectives, scope, and significance

This project aims to design and implement a simple wind gauge system to measure wind speed and direction. The objective is to create a functional device that provides accurate measurements for practical use in small-scale settings, such as local weather monitoring or hobby projects. The scope of the project includes building a basic prototype and testing it to ensure reliable performance. The significance of this project lies in its ability to offer an affordable and straightforward solution for individuals or communities interested in understanding wind conditions in their area.

2. Key outcomes or current status of the project

The project is currently in progress. A prototype wind gauge has been constructed using readily available materials and is undergoing initial testing to check its accuracy and functionality. The results so far are encouraging, showing that the device is capable of delivering reliable wind measurements for its intended purpose.

2. Introduction

2.1 Project background and context

Wind is a critical environmental factor that influences various activities, including weather forecasting, outdoor planning, and small-scale agricultural practices. Understanding wind speed and direction can provide valuable insights for everyday decision-making. This project explores the creation of a basic wind gauge system, focusing on practicality and ease of use for individuals or small groups.

2.2 Problem statement and objectives

Currently, accessible and cost-effective solutions for measuring wind speed and direction are limited, especially for individuals or small-scale users. The primary objective of this project is to design and build a wind gauge that is simple, affordable, and reliable for nonindustrial purposes. The device aims to fill the gap between expensive professional tools and impractical manual methods.

2.3 Description of target audience and end users

The target audience for this project includes hobbyists, small-scale farmers, students, and anyone interested in understanding local wind conditions. The end users are individuals who require an easy-to-operate and budget-friendly device for personal or educational use.

2.4 Outline of report structure

This report is divided into the following sections:

- **Section 1: Summary** – Offers a concise explanation of the project's goals, scope, importance, and current progress, including key findings or results.

- **Section 2: Introduction** – Covers the project's background, problem statement, aims, target audience, and a brief description of the report's layout.
- **Section 3: Methodology** – Details the steps taken during project planning and execution, including tools, technologies, and considerations for accessibility compliance.
- **Section 4: Research and Analysis** – Reviews relevant literature, compares similar systems, and highlights insights gathered during the research phase.
- **Section 5: Work in Progress** – Summarizes tasks completed so far, challenges encountered, and notable code features linked to the Git repository.
- **Section 6: Lessons Learned** – Discusses key takeaways from each task, strategies used to overcome obstacles, and evaluates the efficiency of tools and approaches applied.
- **Section 7: Final Deliverables and Achievements** – Highlights the final product, instructions for usage, and includes a guide for users to operate the developed system.
- **Section 8: Critical Self-Reflection** – Provides personal reflections on individual contributions and evaluates the group's effectiveness in collaboration.
- **Section 9: Future Work** – Proposes enhancements, features to be added, and suggestions for refining the development process in subsequent iterations.
- **Section 10: References** – Lists all sources, articles, and materials that were referred to during the course of the project.

3. Methodology

3.1 Project planning and development process

The project followed an **Agile methodology**, breaking down the development process into smaller, iterative cycles or sprints. Each sprint focused on completing specific tasks, such as sensor integration, coding, and testing the wind gauge. Regular evaluations after each sprint allowed adjustments and improvements based on testing outcomes and feedback. This flexible approach ensured continuous progress and adaptability throughout the project.

3.2 Tools and technologies used

- **Hardware:**
    - **Arduino Uno:** The microcontroller used to manage and process data from the wind sensor.
    - **YGC-FS-5V-V Sensor:** A sensor used to measure wind speed.
    - **Wires:** Used to connect the components and build the circuit.
- **Software:**
    - **Arduino IDE:** The primary software used for programming the Arduino Uno and integrating the hardware components with the system. The Arduino IDE was used to write the code that allows the microcontroller to collect data from the wind sensor and send it to the laptop for display.

3.3 ADA compliance considerations and implementation

Although the project is on a smaller scale, ADA compliance was considered to make the wind gauge more accessible:

- The results are displayed on a laptop screen, which ensures the text is large and easy to read.
- The setup is designed for ease of use, with simple connections and an intuitive interface, allowing users with varying technical skills to operate the system.
- While there was no physical LCD display, the laptop screen provides clear visibility, and the interface can be adjusted for better readability, including font size and color contrast.

4. Research and Analysis

4.1 Literature review or research conducted to support the project

To support the development of the wind gauge system, research was conducted on existing wind measurement technologies and sensors. Literature sources included articles on anemometer designs, microcontroller integration, and sensor calibration. Key insights highlighted the importance of accuracy in wind speed measurement and the use of low-cost, widely available sensors like the YGC-FS-5V-V for small-scale applications. Research also indicated that integrating simple, off-the-shelf components such as the Arduino Uno could significantly lower costs without compromising functionality.

4.2 Comparative analysis of similar systems or solutions

Several wind gauge systems were reviewed to understand the range of available solutions. Most commercial devices, such as digital anemometers, rely on proprietary sensors and are relatively expensive. In comparison, the system developed in this project uses affordable and widely accessible components, making it suitable for individual or small-scale use. Other similar systems that integrate Arduino with wind sensors were also considered, but many require more complex setups or additional components for data visualization. Our approach, using a laptop screen for displaying results, ensures simplicity and cost-effectiveness.

4.3 Challenges and insights gained from the research phase

During the research phase, one challenge was ensuring accurate calibration of the YGC-FS5V-V sensor, as variations in wind conditions and sensor quality could affect results. However, the simplicity of the Arduino Uno and its compatibility with the sensor allowed for straightforward testing and adjustments. Another key insight was the value of opensource platforms like Arduino, which offer extensive documentation and community support, making the development process smoother. Additionally, the research highlighted the potential for future improvements, such as adding wireless data transmission or using a more advanced display system for real-time monitoring.

5. Work in Progress

1. Detailed Description of Tasks Completed So Far:

- **Task 1: Hardware Setup and Integration** ○ **Objective:** Connect the YGC-FS-5V-V wind sensor to the Arduino Uno for wind speed measurement.
    - **Implementation Details:**
        - Wired the sensor to the Arduino Uno using appropriate connections.
        - Verified sensor output by running test code to ensure data was transmitted correctly to the Arduino.
- **Task 2: Programming and Data Visualization** ○ **Objective:** Write Arduino code to read wind speed data from the sensor and display it on the serial monitor.
    - **Implementation Details:**
        - Programmed the Arduino using the Arduino IDE.
        - Implemented a loop to process sensor data and display real-time wind speed values in the serial monitor.
- **Task 3: Testing and Calibration** ○ **Objective:** Ensure accurate wind speed readings under different airflow conditions.
    - **Implementation Details:**
        - Conducted multiple tests by exposing the sensor to varying airflow levels.
        - Adjusted calibration parameters in the code to improve the accuracy of readings.

2. Challenges Faced and Solutions Applied:

- **Challenge 1:** Fluctuating sensor readings in unstable airflow environments.
    - **Solution:** Added filtering logic in the Arduino code to smooth data output and improve stability.

- **Challenge 2:** Limited data visualization options without additional hardware.
    - **Solution:** Leveraged the serial monitor in the Arduino IDE to display wind speed data effectively.

6. Lessons Learned

6.1 Insights gained from each completed task:

- **Sensor Integration and Setup:**

  Integrating the YGC-FS-5V-V sensor with the Arduino Uno highlighted the importance of understanding sensor specifications and ensuring proper connections for accurate data collection. Thorough testing of connections and code logic was essential to avoid initial setup errors.

- **Data Display on Laptop:**

  Using the laptop as the primary display underscored the value of simplicity in visualization for small-scale projects. Formatting the output on the serial monitor effectively enhanced readability and user-friendliness without additional hardware.

- **Testing and Calibration:**

  Testing the emphasized the significance of calibration to achieve consistent results.

6.2 How challenges were addressed and lessons that can be applied to future tasks:

- **Addressing Data Inconsistencies:**

  Noise in sensor readings was mitigated by incorporating filtering techniques into the code. This reinforced the need for robust data handling methods, which will be crucial for any future projects involving sensors or real-time data collection.

- **Managing Limited Visualization Options:**

  Leveraging existing tools, like the laptop and Arduino IDE serial monitor, showcased how creative solutions can overcome resource constraints. This adaptability is a valuable skill for addressing similar challenges in future projects.

- **Calibrating for Accuracy:**

Iterative testing and adjustment of the sensor readings provided a deeper understanding of calibration processes. Future tasks can benefit from establishing a systematic approach to calibration early in the development phase.

6.3 Reflections on the effectiveness of chosen tools and methods:

- **Arduino Uno and YGC-FS-5V-V Sensor:**

  The Arduino Uno proved to be a reliable and versatile platform for integrating and processing data from the wind sensor. The YGC-FS-5V-V sensor was affordable and effective for the project's scope, despite requiring additional calibration efforts.

- **Arduino IDE:**

  The IDE provided a straightforward environment for coding and debugging. Its serial monitor feature allowed seamless data visualization without requiring additional software or hardware, making it a practical choice for this project.

- **Project Approach:**

  The step-by-step approach to sensor integration, data visualization, and testing demonstrated the importance of structured project development. Each stage provided insights that informed subsequent tasks, highlighting the value of iterative learning and adaptation.

- **Lessons for Future Work:**
  - Plan for potential challenges during the initial design phase, such as sensor calibration or data visualization needs.
  - Explore alternative methods for data presentation, including wireless transmission or software dashboards, to expand the project's functionality.
  - Allocate time for iterative testing and debugging to ensure system reliability and accuracy.

7. Final Deliverables and Achievements (if applicable)

7.1 Overview of the Developed System:

The wind gauge system integrates a YGC-FS-5V-V sensor with an Arduino Uno to measure wind speed and display the readings in real-time on a laptop screen using the Arduino IDE serial monitor. The system successfully collects, processes, and visualizes wind speed data, making it suitable for small-scale applications or educational purposes.

7.2 Instructions for Running the Code and Accessing the Repository:

- **Code Setup:**
  - Install the Arduino IDE on your laptop.
  - Connect the YGC-FS-5V-V sensor to the Arduino Uno following the specified wiring configuration.
  - Upload the provided code to the Arduino Uno via the Arduino IDE.
- **Running the System:**
  - Open the serial monitor in the Arduino IDE after uploading the code.
  - Set the baud rate to 9600 to view real-time wind speed data on the screen. ○ Expose the sensor to airflow to see variations in wind speed readings.
- **Repository Access:**
  - The code is available in the following GitHub repository: [GitHub Link to Project Code].

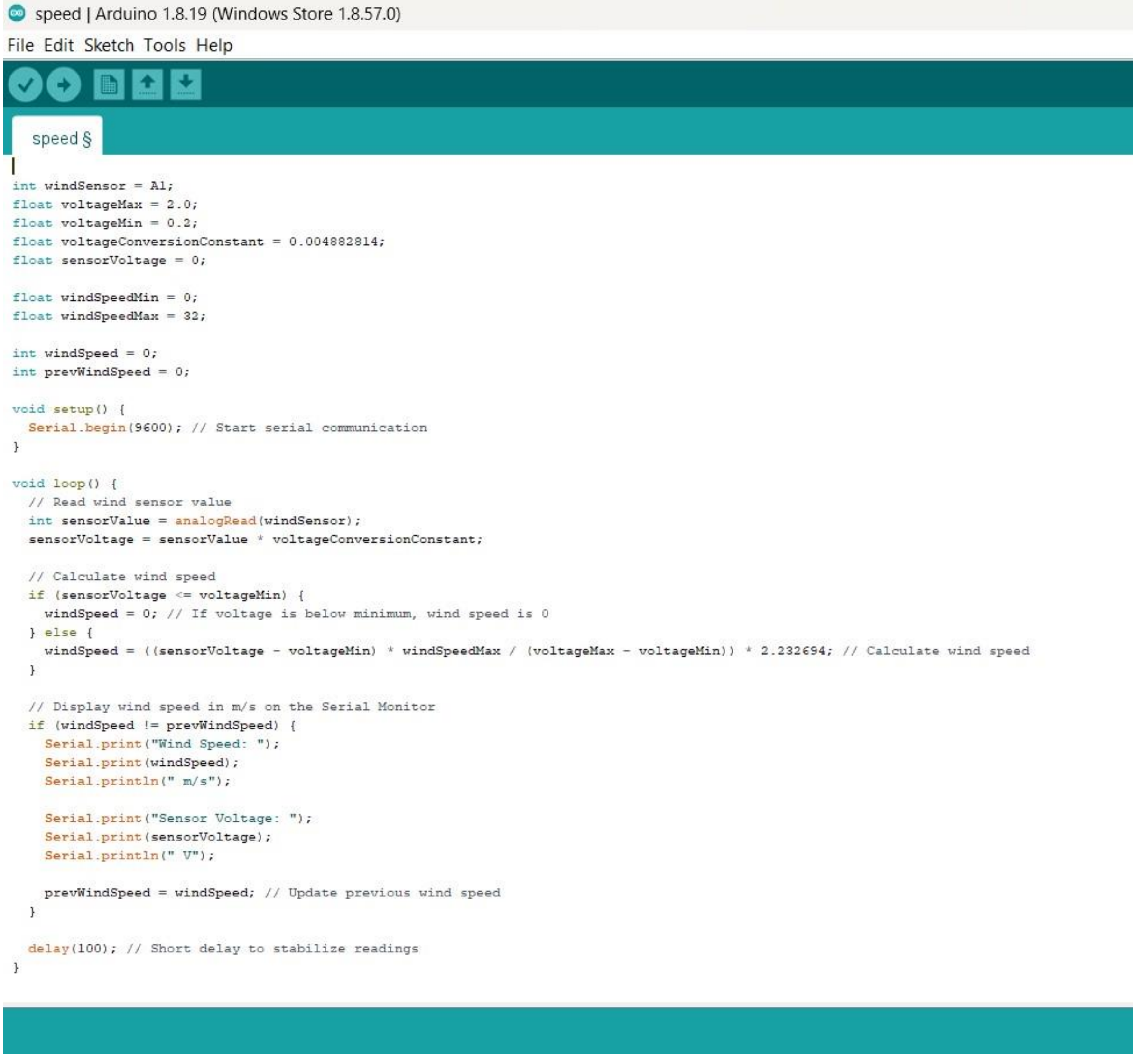
7.3 User Guide and Demonstration:

- A **video demonstration** has been created to showcase the system in action, providing a clear understanding of how the sensor works and how the wind speed data is displayed in real time.

Video link:

https://youtu.be/lGtEI1mUnTw?si=rHov2VXGoE_pkujk

Here is a screenshot of the code used in video:




```
int windSensor = A1;
float voltageMax = 2.0;
float voltageMin = 0.2;
float voltageConversionConstant = 0.004882814;
float sensorVoltage = 0;

float windSpeedMin = 0;
float windSpeedMax = 32;

int windSpeed = 0;
int prevWindSpeed = 0;

void setup() {
  Serial.begin(9600); // Start serial communication
}

void loop() {
  // Read wind sensor value
  int sensorValue = analogRead(windSensor);
  sensorVoltage = sensorValue * voltageConversionConstant;

  // Calculate wind speed
  if (sensorVoltage <= voltageMin) {
    windSpeed = 0; // If voltage is below minimum, wind speed is 0
  } else {
    windSpeed = ((sensorVoltage - voltageMin) * windSpeedMax / (voltageMax - voltageMin)) * 2.232694; // Calculate wind speed
  }

  // Display wind speed in m/s on the Serial Monitor
  if (windSpeed != prevWindSpeed) {
    Serial.print("Wind Speed: ");
    Serial.print(windSpeed);
    Serial.println(" m/s");

    Serial.print("Sensor Voltage: ");
    Serial.print(sensorVoltage);
    Serial.println(" V");

    prevWindSpeed = windSpeed; // Update previous wind speed
  }

  delay(100); // Short delay to stabilize readings
}
```

8. Critical Self-Reflection

Critical Self-Reflection

8.1 Individual Reflections on Task Contributions:

- **What went well?**
  - Successfully integrating the YGC-FS-5V-V sensor with the Arduino Uno was a significant accomplishment. The step-by-step approach to wiring and coding minimized errors during the development process.
  - The use of the Arduino IDE for real-time data display on the serial monitor provided a simple and effective solution for visualization without requiring additional hardware.
  - Testing and calibrating the sensor helped improve the accuracy and reliability of the wind speed measurements.
- **What could have been improved?**
  - More time could have been allocated to exploring additional features, such as wireless data transmission or integrating a user-friendly graphical interface.
  - Better preparation for sensor calibration might have saved time during the testing phase. A more systematic calibration approach could have streamlined the process. o Documenting the project during each stage could have made it easier to track progress and identify areas for improvement earlier.
- **Skills gained or developed during the project:**
  - Improved understanding of sensor integration and calibration techniques.
  - Gained experience in troubleshooting hardware and software issues, which will be valuable for future projects.

    - Developed the ability to present technical results clearly, as demonstrated in the accompanying video.

8.2 Reflection on the Group's Collaborative Efforts and Outcomes:

- Although the project was conducted individually, support from peer and other groups played a role in addressing challenges effectively.
- Collaborative feedback helped refine the approach to sensor calibration and testing, ensuring better results.
- The project highlights the importance of collaboration for brainstorming ideas and solving problems efficiently, even when working independently.
- The outcome demonstrates that clear objectives, steady progress, and leveraging available resources can lead to a successful project despite limitations in scale and scope.

This reflection underscores the value of iterative learning, adaptability, and seeking feedback to achieve project goals while gaining new skills and insights.

9. Future Work

9.1 Unresolved Issues:

- Sensor accuracy in unstable environments remains a challenge.
- Limited data visualization, as output is restricted to the serial monitor.

9.2 Planned Features:

- Wireless data transmission via Bluetooth or Wi-Fi for remote monitoring.
- Development of a graphical interface for real-time trend visualization.
- Data logging for long-term analysis and advanced calibration tools.

9.3 Suggestions for Improvement:

- Allocate more time for planning and systematic testing.
- Explore alternative sensors or microcontrollers with enhanced precision.
- Use an iterative development process for better feature integration.
- Collaborate with peers for diverse ideas and solutions.

References

- Arduino Documentation – Official Guide for Arduino Uno and Arduino IDE. Available at: https://www.arduino.cc/

- Tutorials on Sensor Calibration – General resources for sensor accuracy improvement. Example source: https://www.sparkfun.com/