Lab Report: Sensor Interfacing with Arduino

Name: Venkata Siva Sai Krishna Kumar Reddyboina
Student ID: 12503460
Course: Embedded Systems
Instructor: Prof. Tobias Schaffer
Date: 25-06-2025

1. Introduction

This lab aims to familiarize students with the basics of sensor integration using the **Arduino Nano 33 BLE Sense Rev2**. It emphasizes reading and interpreting data from the onboard IMU and temperature/humidity sensors. Through this hands-on activity, students will gain practical experience in collecting, analyzing, and displaying real-time sensor data using the Arduino IDE. Understanding these concepts is fundamental to embedded systems development and is especially important in fields like IoT, robotics, and smart technology, where sensor data is crucial for intelligent operation.

2. Methodology

The experiment involved writing and uploading Arduino code to read and display data from onboard sensors using the Arduino IDE. The steps included initializing serial communication, setting up the sensors, acquiring sensor data, and displaying the output in the Serial Monitor.

2.1. Software and Hardware Used

- **Programming Language:** C++ (Arduino framework)
- **Libraries Used:**
  - `Arduino_BMI270_BMM150` (for IMU sensor)
  - `Arduino_HS300x` (for temperature and humidity sensor)
- **Hardware:**
  - Arduino Nano 33 BLE Sense Rev2
  - Micro-USB cable
  - Computer running Arduino IDE

2.2. Code Repository

The full source code for this project is available on GitHub at: `https://github.com/saikumar374/Embedded_Systems`

2.3. Code Implementation

2.3.1 Reading IMU Sensor Data

```
#include "Arduino_BMI270_BMM150.h"

void setup() {
    Serial.begin(9600);
    while (!Serial);
    if (!IMU.begin()) {
        Serial.println("IMU␣initialization␣failed!");
        while (1);
    }

}

void loop() {
    float x, y, z;
    if (IMU.accelerationAvailable()) {
        IMU.readAcceleration(x, y, z);
        Serial.print("Acceleration:␣X="); Serial.print(x);
        Serial.print("␣Y="); Serial.print(y);
        Serial.print("␣Z="); Serial.println(z);
    }
    delay(500);
}
```

Figure 1:

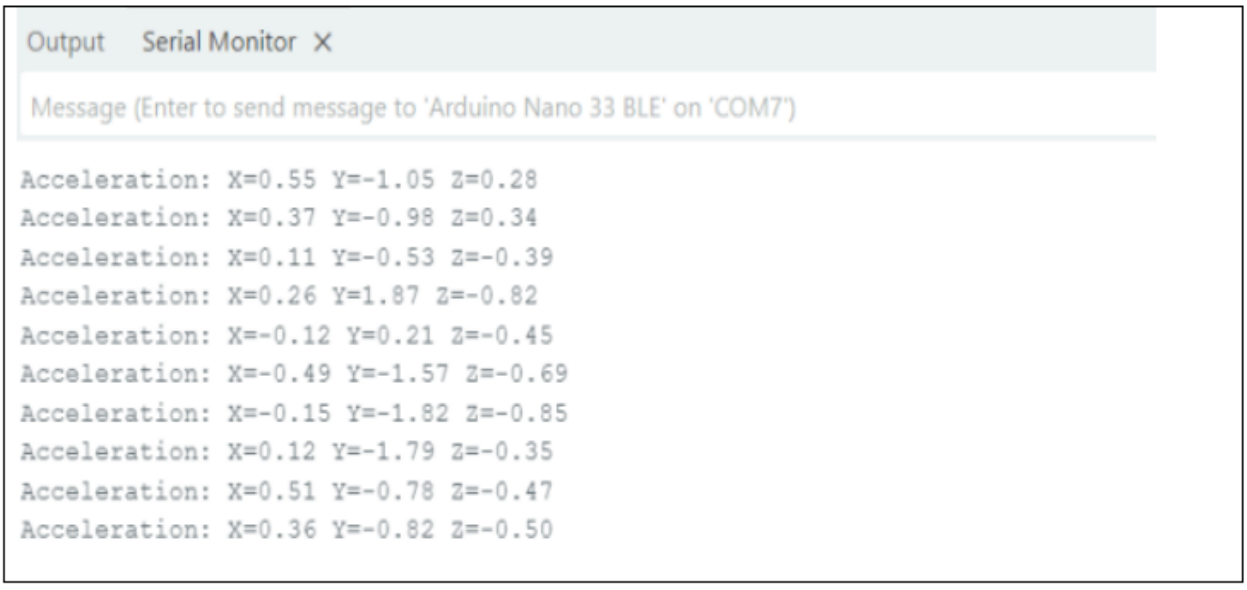
2.3.2 Reading Temperature and Humidity Data

```
#include <Arduino_HS300x.h>

void setup() {
    Serial.begin(9600);
    while (!Serial);
    if (!HS300x.begin()) {
        Serial.println("HS300␣sensor␣initialization␣failed!");
        while (1);
    }
}

void loop() {
    float temp = HS300x.readTemperature();
    float humidity = HS300x.readHumidity();
    Serial.print("Temp:␣"); Serial.print(temp); Serial.print("C␣␣");
    Serial.print("Humidity:␣"); Serial.print(humidity); Serial.println("%");
    delay(2000);
}
```

Figure 2:

3. Results

The images display the results recorded during the code execution, providing a visual representation of the sensor outputs collected in real time over the course of the experiment.



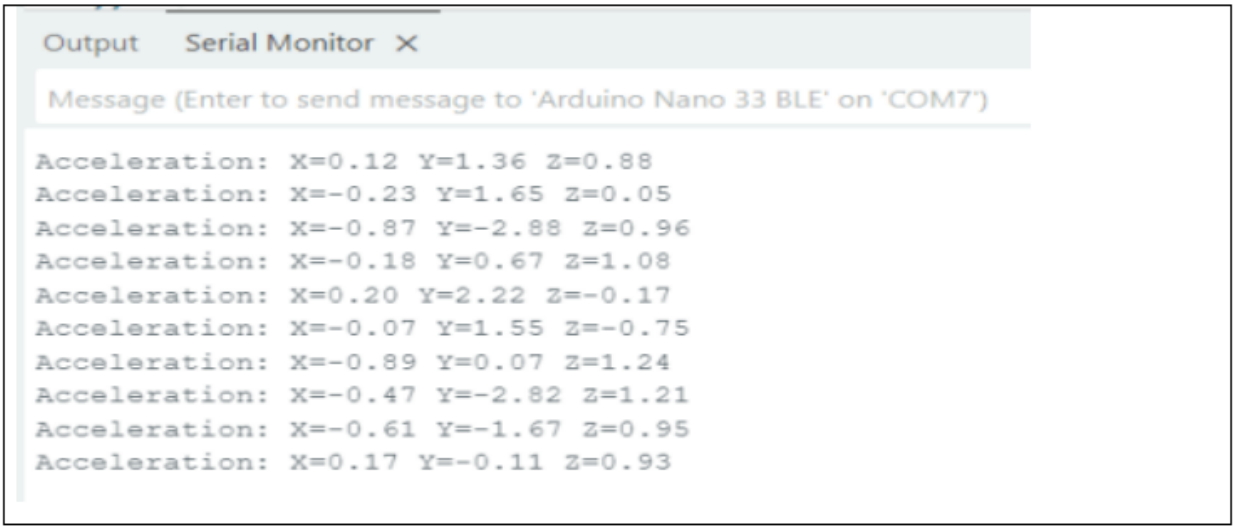


Figure 3: Serial monitor displaying real-time acceleration readings (X, Y, Z) from the IMU sensor on the Arduino Nano 33 BLE Sense Rev2.

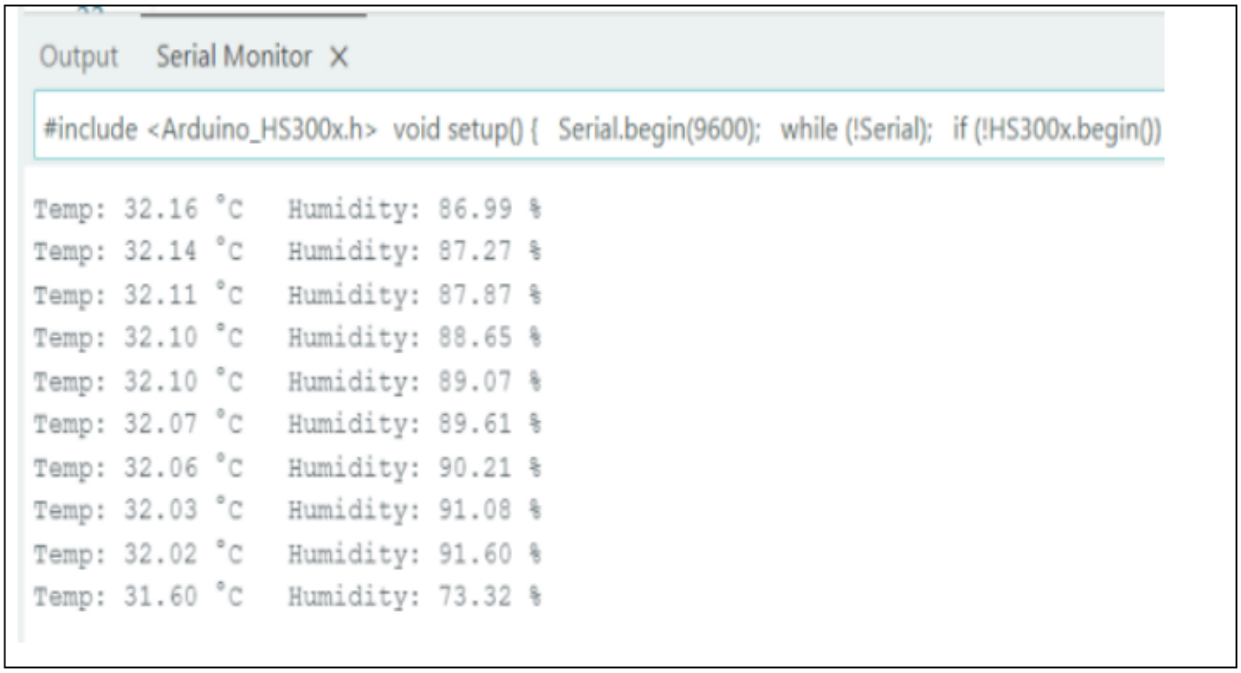


Figure 4: Serial monitor displaying temperature and humidity readings.

4. Challenges, Limitations, and Error Analysis

4.1. Challenges Faced

- Understanding the correct initialization process for the IMU and sensor libraries.
- Ensuring proper installation and inclusion of the required libraries in the Arduino IDE.
- Interpreting raw sensor data and formatting it for meaningful output.

4.2. Error Analysis

- Occasional communication delays caused incomplete or missing sensor data on the Serial Monitor.
- Incorrect library references initially caused compilation errors during upload.
- Logical error in reading sensor data before checking its availability led to runtime issues.

4.3. Limitations of the Implementation

- The current code only reads acceleration; other IMU features like gyroscope or magnetometer are not utilized.
- Temperature and humidity readings were not integrated in this implementation.
- No data filtering or averaging was applied, which may cause noise in real-time readings.

5. Discussion

The lab successfully demonstrated how to interface and read data from onboard sensors of the Arduino Nano 33 BLE Sense Rev2. It provided practical exposure to embedded programming and the importance of real-time sensor monitoring. The results were consistent with expectations, showing meaningful data variation when the device was moved or exposed to different temperatures.

6. Conclusion

This lab provided foundational knowledge in sensor interfacing with embedded systems. Students learned how to program and interact with real-world data using the Arduino platform. Future improvements could include data logging, wireless transmission, and integrating more complex processing algorithms.