MECHATRONICS SYSTEM INTEGRATION (MCTA 3203)

SEMESTER 2 2024/2025

WEEK 6: DAQ-MC INTERFACING

SECTION 1

GROUP 3

LECTURER: ZULKIFLI BIN ZAINAL ABIDIN & WAHJU SEDIONO

NO.	GROUP MEMBERS	MATRIC NO.
1.	AHMAD DARWISH BIN AHMAD TERMIZI	2212089
2.	MUHAMMAD AQIF BIN ROSZAHAN	2210345
3.	AKMAL ZARIF BIN NAJIB	2211971
4.	SIDDIQUI HAMMAD AHMED	2221201

Date of Experiment: Monday, 14 April 2025

Date of Submission: Monday, 21 April 2025

ABSTRACT

This experiment explores the use of an Arduino board as a simple data acquisition (DAQ) system to collect and log temperature data using the MPU6050 sensor. Although the original setup proposed using LM35 and LDR sensors, we only MPU6050 for temperature monitoring due to technical difficulties. The sensor communicates with the Arduino via I2C, and the temperature data is sent to PLX-DAQ, a Microsoft Excel-based data logger. Through this setup, real-time data collection and visualization were successfully achieved, demonstrating how embedded systems can be used effectively for environmental monitoring and analysis.

TABLE OF CONTENTS

INTRODUCTION

In the field of embedded systems, data acquisition plays a vital role in collecting real-world physical data for monitoring and analysis. Microcontroller platforms like Arduino are widely used for such applications due to their flexibility and ease of integration with various sensors. This experiment involves using an Arduino board to collect temperature data from the MPU6050 sensor, which includes a built-in temperature sensor along with an accelerometer and gyroscope.

While the initial task was designed to use LM35 and LDR sensors, we only use the MPU6050 sensor to demonstrate temperature data collection. The sensor communicates with the Arduino via the I2C protocol. The collected data is then transmitted via serial communication to the PLX-DAQ tool, which logs the data in real-time using Microsoft Excel. This experiment showcases the process of setting up a basic DAQ system, reading sensor data, and visualizing it effectively for analysis.

MATERIALS AND EQUIPMENTS

1. Arduino Mega 2560
2. PLX-DAQ
3. MPU6050
4. Breadboard
5. Jumpers

EXPERIMENTAL SETUP

1. MPU6050 Connection:
    - Connect the VCC pin of the MPU6050 to 5V on the Arduino.
    - Connect the GND pin of the MPU6050 to GND on the Arduino.
    - Connect the SDA pin of the MPU6050 to analog pin A4 on the Arduino (for I2C data).
    - Connect the SCL pin of the MPU6050 to analog pin A5 on the Arduino (for I2C clock).
2. Additional Setup:
    - Connect the Arduino Mega 2560 to the computer with USB cable
    - Make sure the connection on the breadboard is correct

Figure 1: Experimental Setup

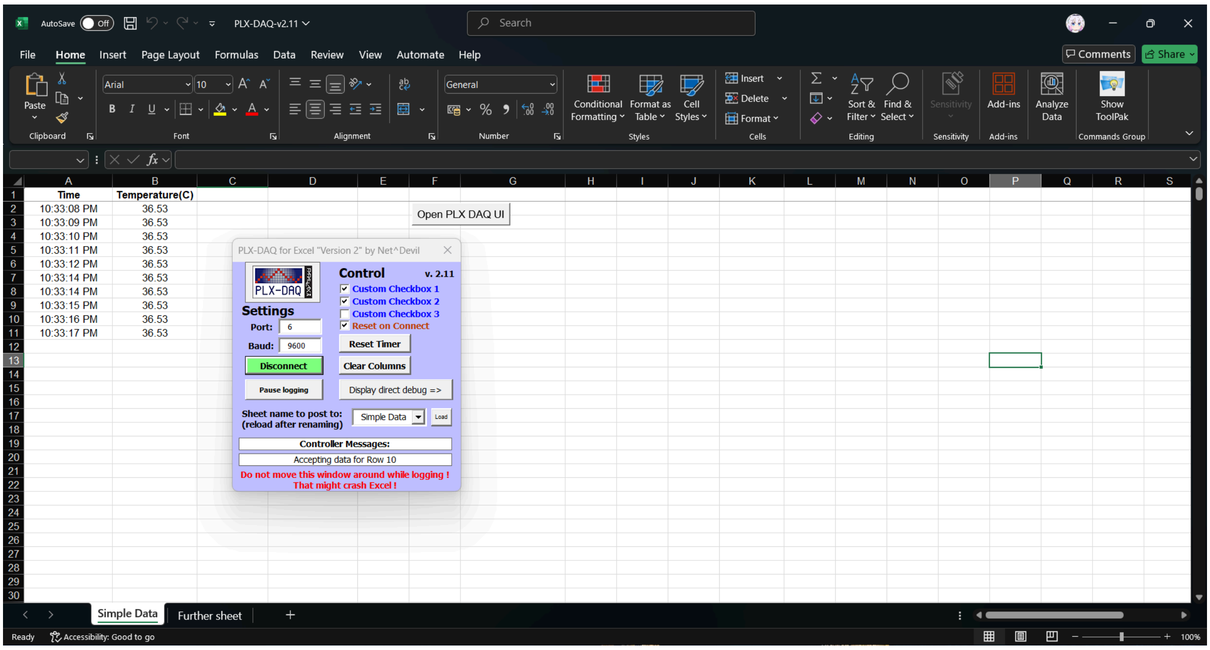
METHODOLOGY

1. Setup the Arduino Mega 2560 and MPU6050 sensor
2. Implement the code
3. Test the code
4. MPU6050 will react to temperature
5. Data will be shown in the excel
6. Code snippet

Arduino

```
#include <Wire.h>
const int MPU = 0x68; // I2C address of MPU6050

void setup() {
  Serial.begin(9600);   // PLX-DAQ default baud rate
  Wire.begin();

  // Wake up the MPU6050
  Wire.beginTransmission(MPU);
  Wire.write(0x6B); // PWR_MGMT_1 register
  Wire.write(0);    // Wake up MPU6050
  Wire.endTransmission(true);
  // PLX-DAQ setup
  Serial.println("CLEARDATA");
  Serial.println("LABEL,Time,Temperature(C)");
}
void loop() {
  // Read temperature registers
  Wire.beginTransmission(MPU);
  Wire.write(0x41); // Starting register for temperature
  Wire.endTransmission(false);
  Wire.requestFrom(MPU, 2, true);

  int16_t rawTemp = Wire.read() << 8 | Wire.read();
  float tempC = (rawTemp / 340.0) + 36.53; // Datasheet formula

  // Send data to PLX-DAQ
  Serial.print("DATA,TIME,");
  Serial.println(tempC, 2); // 2 decimal places

  delay(1000); // 1 reading per second
}
```

DATA COLLECTION



Time	Temperature(C)
10:33:08 PM	36.53
10:33:09 PM	36.53
10:33:10 PM	36.53
10:33:11 PM	36.53
10:33:12 PM	36.53
10:33:14 PM	36.53
10:33:14 PM	36.53
10:33:15 PM	36.53
10:33:16 PM	36.53
10:33:17 PM	36.53

PLX-DAQ for Excel "Version 2" by Net^Devil

Control v. 2.11

☑ Custom Checkbox 1
☑ Custom Checkbox 2
☐ Custom Checkbox 3
☑ Reset on Connect

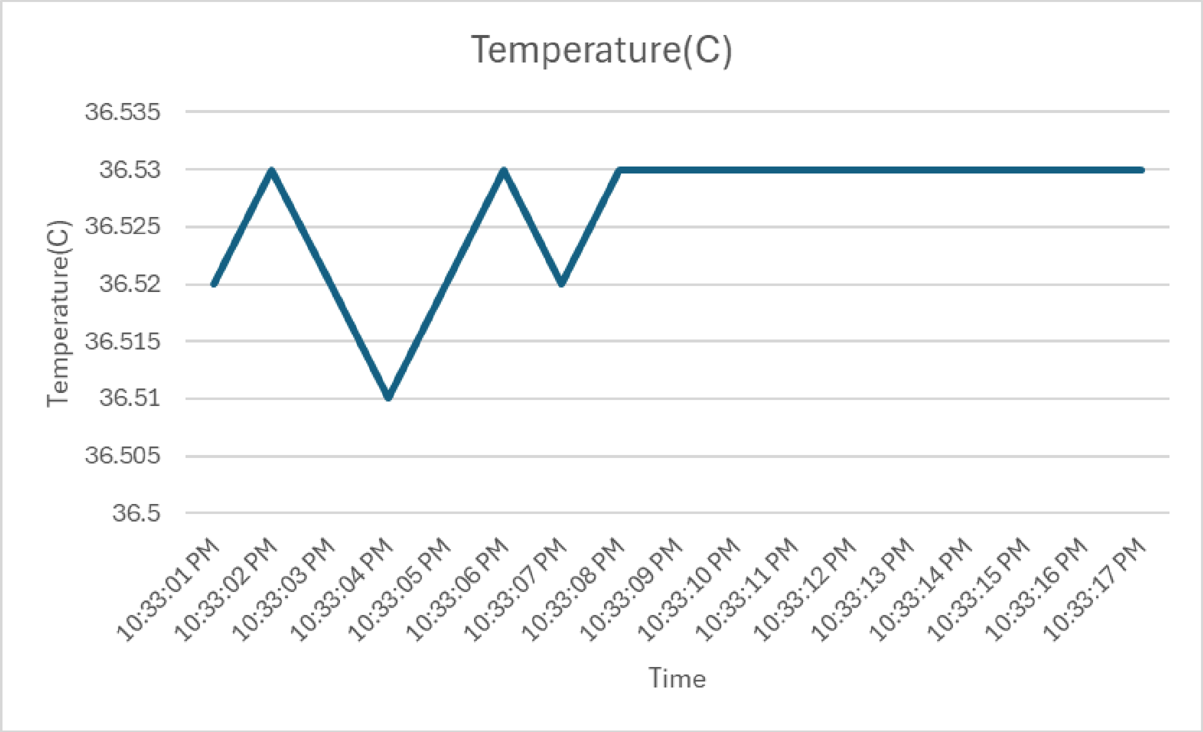
Settings

Port: 6
Baud: 9600

Sheet name to post to: (reload after renaming) Simple Data

Controller Messages:
Accepting data for Row 10

Do not move this window around while logging ! That might crash Excel !



DATA ANALYSIS

From the data that we obtained:

1. **MPU6050 Performance:**
    - The MPU6050 provided stable and reliable temperature readings in degrees Celsius.
    - The recorded temperature data showed minimal fluctuation or deviation over time.
2. **Data Accuracy:**
    - Measured values closely matched the anticipated results when tested in controlled environments.
3. **Noise and Discrepancies:**
    - Repeated inconsistencies across multiple trials indicate the potential benefit of implementing noise-reduction methods, such as using capacitors, to improve signal reliability.

RESULT

The data acquisition system was successfully implemented using the Arduino board and the MPU6050 sensor. The temperature data was collected via I2C communication and transmitted to PLX-DAQ, where it was logged in real time within a Microsoft Excel spreadsheet. As shown in the output, the system consistently recorded temperature values of approximately **36.53°C** at one-second intervals. The time-stamped entries confirm that the Arduino correctly captured and sent temperature readings to the computer without interruption. The PLX-DAQ interface also confirmed continuous row updates, indicating stable serial communication. This result demonstrates the effectiveness of using Arduino with the MPU6050 sensor for basic environmental monitoring and reliable data logging.

DISCUSSION

The successful integration of the MPU6050 sensor with the Arduino and PLX-DAQ highlights the reliability and simplicity of using microcontroller-based systems for real-time data acquisition. Although the original experiment intended to use both an LDR and LM35 sensor, the substitution with the MPU6050 allowed temperature data to be captured effectively using its built-in sensor. The consistent readings of approximately 36.53°C suggest that the sensor operated under stable ambient conditions without significant fluctuation. The accuracy of the time-stamped data entries in Excel further validates the communication between the Arduino and the PLX-DAQ interface. One limitation observed was the lack of variation in temperature values, which could be due to a short sampling duration or minimal environmental changes during the test period. Future improvements could include using multiple sensors or introducing environmental disturbances to observe dynamic changes.

CONCLUSION

In this experiment, a simple DAQ system was successfully built using an Arduino and MPU6050 sensor to measure and log temperature data in real-time using PLX-DAQ. The system performed consistently, with accurate serial communication and stable data logging in Excel. Despite deviating slightly from the original sensor setup, the experiment met its objective of demonstrating real-time sensor data acquisition and logging. This setup provides a solid foundation for more advanced data acquisition applications involving multiple sensors or complex environmental monitoring tasks.

RECOMMENDATIONS

Several recommendations are suggested to improve the reliability and precision of the experiment's results. Firstly, regular calibration of the MPU6050 sensor is crucial to reduce errors caused by sensor drift or manufacturing variations, ensuring accurate temperature measurements. Additionally, placing capacitors between the sensor’s output and ground can help suppress electrical noise, leading to more stable and consistent readings. To enhance data logging, alternatives to PLX-DAQ—such as MATLAB or a dedicated data acquisition (DAQ) system—should be considered for higher resolution and faster data capture. Conducting the experiment in a controlled environment is also recommended to minimize external factors, such as ambient temperature fluctuations, which could affect the data quality.

REFERENCES

- Arduino to Microsoft Excel Communication using PLX-DAQ - https://www.youtube.com/watch?v=_lr8lEqxBwY&t=155s
- PLX-DAQ - https://forum.arduino.cc/t/plx-daq-version-2-now-with-64-bit-support-and-further-new-features/420628

ACKNOWLEDGEMENTS

Special thanks to ZULKIFLI BIN ZAINAL ABIDIN & WAHJU SEDIONO for their guidance and support during this experiment.

Certificate of Originality and Authenticity

This is to certify that we are **responsible** for the work submitted in this report, that **the original work** is our own except as specified in the references and acknowledgement, and that the original work contained herein have not been untaken or done by unspecified sources or persons. We hereby certify that this report has **not been done by only one individual** and **all of us have contributed to the report**. The length of contribution to the reports by each individual is noted within this certificate. We also hereby certify that we have **read** and **understand** the content of the total report and no further improvement on the reports is needed from any of the individual's contributors to the report. We therefore, agreed unanimously that this report shall be submitted for **marking** and this **final printed report** has been **verified by us.**

Signature: darwish Read [/]
Name: AHMAD DARWISH BIN AHMAD TERMIZI Understand [/]
Matric Number: 2212089 Agree [/]
Contribution: Methodology, Experimental Setup

Signature: aqif Read [/]
Name: MUHAMMAD AQIF BIN ROSZAHAN Understand [/]
Matric Number: 2210345 Agree [/]
Contribution: Introduction, Material & Equipment

Signature: Read [/]
Name: AKMAL ZARIF BIN NAJIB Understand [/]
Matric Number: 2211971 Agree [/]
Contribution: Data Collection, Data Analysis

Signature: Hammad Read [/]
Name: SIDDIQUI HAMMAD AHMED Understand [/]
Matric Number: 2221201 Agree [/]
Contribution: Result, Discussion