REPORT 5: DAQ INTERFACING WITH MICROCONTROLLERS.

GROUP 4

MCTA 3203

SEMESTER 1 2024/2025

MECHATRONICS SYSTEM INTEGRATION

DATE OF SUBMISSION: 20 NOVEMBER 2024

NO	NAME	MATRIC NUMBER
1.	IRDINA NABIHAH BINTI MOHD NAZRI	2214772
2.	KHALISAH AMANI BINTI YAHYA AZMI	2218184
3.	LUQMAN AZFAR BIN AZMI	2219857
4.	MOHAMAD NASRI BIN MOHAMAD NAZRI	2219879
5.	MUHAMAD NURHAKIMIE THAQIF BIN ABDULLAH	2213217

TABLE OF CONTENT

INTRODUCTION

Data acquisition (DAQ) systems are an important component in measuring and recording the physical world electronically that help engineers with experiments before they release a product into the market. A simple DAQ system consists of three basic elements which are sensors to translate physical measurements into electrical signals ,a DAQ device to acquire, condition and digitize the signals; and a computer to collect and analyze the data.

In this laboratory experiment, we implement a simple DAQ system using an Arduino microcontroller interfaced with two sensors,a Light Dependent Resistor (LDR) and an LM35 temperature sensor. The system utilizes Parallax Data Acquisition (PLX-DAQ), an excel used to enable real-time data logging and analysis of the sensor readings. This setup demonstrates the practical application of DAQ principles in a mechatronic system, allowing us to collect and analyze environmental data through computer-based measurements.

ABSTRACT

The experiment demonstrates the successful integration of hardware and software components in a basic DAQ system, allowing for real-time monitoring of environmental parameters. The Arduino board acts as the DAQ hardware, converting analog sensor signals to digital data through its built-in ADC capabilities. The system utilizes serial communication to transmit the processed data to PLX-DAQ, where it is automatically organized into Excel spreadsheets for further analysis.

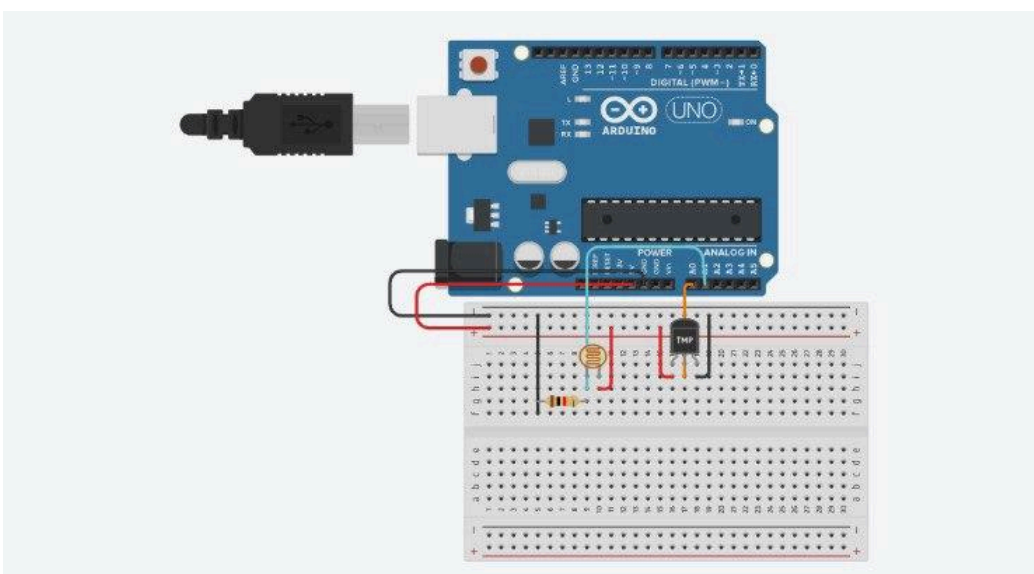
Through this we learn practical sensor interfacing, analog to digital conversion and data logging techniques used in mechatronic systems. How to capture, process and analyze physical parameters (light intensity and temperature) in real-time using a low cost DAQ solution. We will see how to integrate different hardware components (Arduino, LDR and LM35 sensors) with software tools (PLX-DAQ Excel add-in) to get a reliable data acquisition system. PLX-DAQ with Microsoft Excel is a powerful tool to visualize and analyze sensor measurements in real applications.

EXPERIMENT 5: **Collecting data from sensors by using Arduino as a simple DAQ device and transfering the data to the PLX‑DAQ for data logging and analysis.**

MATERIALS AND EQUIPMENT

- PLX‑DAQ
- Arduino Board
- LDR
- LM35
- Jumper Wires
- Resistor
- Breadboard

EXPERIMENTAL SETUP



1. After constructing the circuit, launch Arduino IDE and write code that allows Arduino to read analog signals from the LM35 and LDR and convert it to digital.

2. Complete the example code below. (Alternatively, you may write your own code from the start).

3. Verify the code and upload it to the Arduino board.

4. Launch the PLX‑DAQ spreadsheet. Ensure correct com port is selected and generate the output from the sensors in the spreadsheet.

5. In your report, write the comments to explain each line of the codes and produce meaningful excel plots from the sensors' data.

METHODOLOGY

This experiment will utilize an Arduino as a DAQ device that acquires data coming from sensors. Sensors to be used during the experiment are the LM35 temperature sensor and LDR, or Light Dependent Resistor. The steps are as follows:

Construction of Circuit: Constructing the circuit in Fig. 3 where the LM35 and LDR are connected to the Arduino.

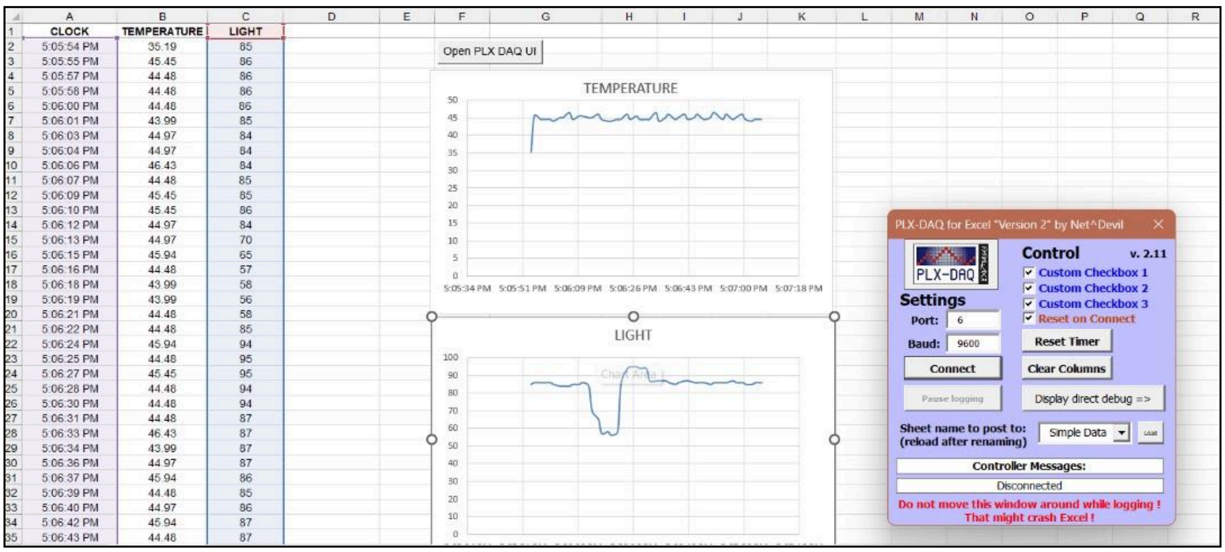
Programming Arduino: Open the Arduino IDE and compose or complete the example code enabling the Arduino to read analog signals from the sensors and convert them to digital format.

Data Logging Setup: The working software for data logging is PLX-DAQ, which works within Microsoft Excel. Start PLX-DAQ and make sure the proper COM port is selected, with the baud rate set to match the settings in the Arduino code.

Data Collection: Once the setup is done, click on the connect tab in PLX-DAQ to display the Arduino data into the Excel spreadsheet.

Data Analysis: After data collection, analyze the data using tools provided by Excel and plot meaningful plots.

RESULTS



Excel View Using Open PLX DAQ

```
float tempcelc;
int lm_value;
int ldr_value;
int ldr_percent;

void setup() {
  Serial.begin(9600);
  Serial.println("CLEARDATA");
  Serial.println("LABEL,CLOCK,TEMPERATURE,LIGHT"); // Labels for
PLX-DAQ columns
}

void loop() {
```

```
  // Read temperature from LM35 sensor
  lm_value = analogRead(A0);
  //tempcelc = ((5 * lm_value* 100)/1024);
  tempcelc = (lm_value / 1023.0) * 5000; // Convert to millivolts
  tempcelc = tempcelc / 10; // Convert to degrees Celsius

  // Read LDR value and map it to a percentage
  ldr_value = analogRead(A1);
  ldr_percent = map(ldr_value, 0, 1023, 0, 100);

  // Print both temperature and light on the same line for PLX-DAQ
  Serial.print("DATA,TIME,");
  Serial.print(tempcelc); // Print temperature value
  Serial.print(",");      // Separator for next data
  Serial.println(ldr_percent); // Print light percentage value

  delay(1500); // Delay for 1.5 seconds
}
```

This code reads temperature and light intensity data and sends it to PLX-DAQ software for real-time visualization in Excel. The LM35 temperature sensor's analog value is read from pin A0 which is then converted to millivolts, and then to degrees Celsius. Similarly, the LDR's light intensity is read from pin A1, and the value is mapped to a percentage (0–100%). These values are then sent to the Serial Monitor in a PLX-DAQ-compatible format, including column labels ("CLOCK," "TEMPERATURE," "LIGHT"). The loop repeats every 1.5 seconds, continuously updating the data.

DISCUSSION

This experiment successfully demonstrates the functionality of a simple data acquisition (DAQ) system for monitoring temperature and light intensity in real-time. By using an Arduino microcontroller, LM35 temperature sensor, and an LDR (Light Dependent Resistor), environmental parameters were measured, processed, and logged into PLX-DAQ for visualization in Microsoft Excel.

Challenges Encountered:

- Noise and Signal Stability:
  Analog signals from the LM35 and LDR were occasionally susceptible to fluctuations, possibly due to electrical noise or environmental interference. This could be mitigated by implementing signal filtering or averaging techniques.
- Resolution Limitation:
  The Arduino's 10-bit ADC resolution limited the precision of sensor readings, particularly for temperature data. Employing higher-resolution ADCs or external conditioning circuits could improve measurement accuracy.

CONCLUSION

This experiment successfully demonstrates how to interface sensors to a microcontroller in order to acquire data for analysis. Employing the Arduino as a DAQ device and PLX-DAQ for logging data, the experiment will be able to monitor and analyze the sensor data in real time. The acquired data can be represented efficiently in Excel to analyze further. This methodology can be applied in a number of ways in such applications as environmental monitoring and equipment diagnostics, thus showing the broadness of practical utilization of DAQ systems.

RECOMMENDATION

To make the experiment smoother and reduce errors, start by ensuring all sensors, such as the LDR and LM35, are properly calibrated to guarantee accurate data readings. Adjusting the data sampling rate can help strike a balance between responsiveness and data noise; an excessively high rate may produce redundant data points, while a low rate risks missing important changes. Additionally, maintaining a stable power supply is essential, as fluctuations can cause data inconsistencies or device malfunctions. Checking all hardware connections, including wiring, jumpers, and resistors, for stability will prevent intermittent data loss or errors. Ensure proper configuration of communication settings when using PLX-DAQ with Excel, as incorrect settings can lead to data loss or communication issues. Implementing robust error-handling routines in the Arduino code can help catch and manage unexpected sensor values or communication errors. Finally, applying statistical methods such as moving averages or median filters in data analysis will further smooth out the data and reduce the impact of noise or anomalies, leading to more reliable experimental outcomes.

ACKNOWLEDGEMENTS

A special thanks goes out to Dr. Wahju Sediono and Dr. Zulkifli Bin Zainal Abidin, my teaching assistant, and my peers for their invaluable help and support in finishing this report. Their advice, feedback, and experience have greatly influenced the level of quality and understanding of this work. Their time, patience, and commitment to supporting my academic success are greatly appreciated.

STUDENT'S DECLARATION

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:
Name: IRDINA NABIHAH BINTI MOHD NAZRI
Matric Number: 2214772

Read ☑
Understand ☑
Agree ☑

Signature:
Name: KHALISAH AMANI BINTI YAHYA AZMI
Matric Number: 2218184

Read ☑
Understand ☑
Agree ☑

Signature:
Name: LUQMAN AZFAR BIN AZMI
Matric Number: 2219857

Read ☑
Understand ☑
Agree ☑

Signature:
Name: MOHAMAD NASRI BIN MOHAMAD NAZRI
Matric Number: 2219879

Read ☑
Understand ☑
Agree ☑

Signature:
Name: MUHAMAS NURHAKIMIE THAQIF BIN ABDULLAH
Matric Number: 2213217

Read ☑
Understand ☑
Agree ☑