MECHATRONICS SYSTEM INTEGRATION

MCTA 3203

LAB 6:

DAQ INTERFACING WITH MICROCONTROLLERS

SECTION 1

SEMESTER 2, 2024/2025

INSTRUCTOR:

ASSOC. PROF. EUR. ING. IR. TS. GS. INV. DR. ZULKIFLI BIN ZAINAL ABIDIN

DR. WAHJU SEDIONO

GROUP 9

NAME	MATRIC NO
AMER WAFDI BIN ZAINIZAM	2228703
AMALIN BALQIS BINTI RUYUSNI	2314684
FARHA QAISARA BINTI MOHD ZAHIDI	2315994

DATE OF SUBMISSION: 21 APRIL 2025

ABSTRACT

This experiment explores the use of an Arduino Uno as a simple data acquisition (DAQ) system for real-time environmental measurement of temperature and light intensity. Analog measurements were successfully read using an LM35 temperature sensor and a light-dependent resistor (LDR) connected to the Arduino, which were then digitized and captured using PLX-DAQ software with Microsoft Excel.

The readings over time indicated clear variations in temperature, ranging from 17.11°C to 55.23°C, and light intensity, ranging from 55% to 82%. Sudden changes in the readings of temperature were indicated, with suggestions that the environment is sensitive or perhaps susceptible to external influences, while the light readings remained constant. Line graphs were used to graph and establish trends and correlations for these. Regardless of minor inconsistencies which may have arisen due to hardware constraints or noise in sensors, the Arduino setup displayed stable logging and interfacing of sensors.

In addition to proving the functional capabilities of sensors and the logging device, the experiment proved beyond doubt that the Arduino-based setup holds tremendous value in its usability for environmental applications, smart automation, and research work. This hands-on experience gave us valuable insights into embedded system design, data analysis, and the real-world practicalities of data collection.

TABLE OF CONTENT

1.0 INTRODUCTION

The purpose of this experiment is to design and implement a simple data acquisition (DAQ) system using an Arduino microcontroller to interface with analog sensors, specifically, a temperature sensor (LM35) and a light-dependent resistor (LDR). The primary objective is to collect real-time environmental data, including temperature and ambient light levels, and transmit this data to a computer for visualization and analysis using PLX-DAQ, an Excel-based data logging tool. This hands-on task to understand sensor interfacing, analog-to-digital conversion, and serial communication protocols essential in embedded systems and IoT applications.

The LM35 is a precision temperature sensor with an analog output proportional to the temperature in degrees Celsius. It is widely used due to its linearity and ease of use with microcontrollers. The LDR, on the other hand, is a passive optical component whose resistance decreases with increasing incident light intensity. When connected in a voltage divider configuration, it allows light intensity to be read as a variable voltage, which can then be interpreted by the Arduino's analog input.

This experiment is grounded in fundamental principles of electronics and microcontroller programming. It highlights key concepts such as signal conditioning, mapping analog sensor values into meaningful physical units such as °C or percentage, and data logging techniques. The use of PLX-DAQ provides an accessible method to visualize the data and draw conclusions about environmental changes over time.

It is hypothesized that the LM35 will provide stable, linearly increasing voltage with temperature, enabling accurate temperature measurement. Similarly, the LDR is expected to show a decrease in resistance with increased light levels, reflected as a higher percentage reading in the processed data and vice versa. Successful implementation will be evident through the smooth, consistent trends observed in Excel plots of temperature and light against time.

2.0 MATERIALS AND EQUIPMENT

The materials and equipment that we used in this experiment are:

a) PLX‑DAQ
b) Arduino Uno board
c) LDR
d) LM35
e) 10k-ohm resistor
f) Breadboard
g) Jumper wires

3.0 EXPERIMENTAL SETUP

Here are the steps of equipment and components that were set up for the experiment:

1) Connect the LM35 temperature sensor to the Arduino by connecting its VCC pin to the 5V pin, GND pin to the GND pin, and the output pin to analog pin A0 of the Arduino.
2) Connect the LDR sensor in a voltage divider configuration by connecting one leg of the LDR to the 5V pin, and the other leg to analog pin A1.
3) A 10K-ohm resistor was used with the LDR by connecting one end of the resistor to analog pin A1 and the other end to the GND pin to complete the voltage divider circuit.
4) Jumper wires were used to establish all connections between the Arduino board, sensors, and the breadboard.
5) The Arduino board was connected to the PC via USB cable to power the board and enable serial communication.

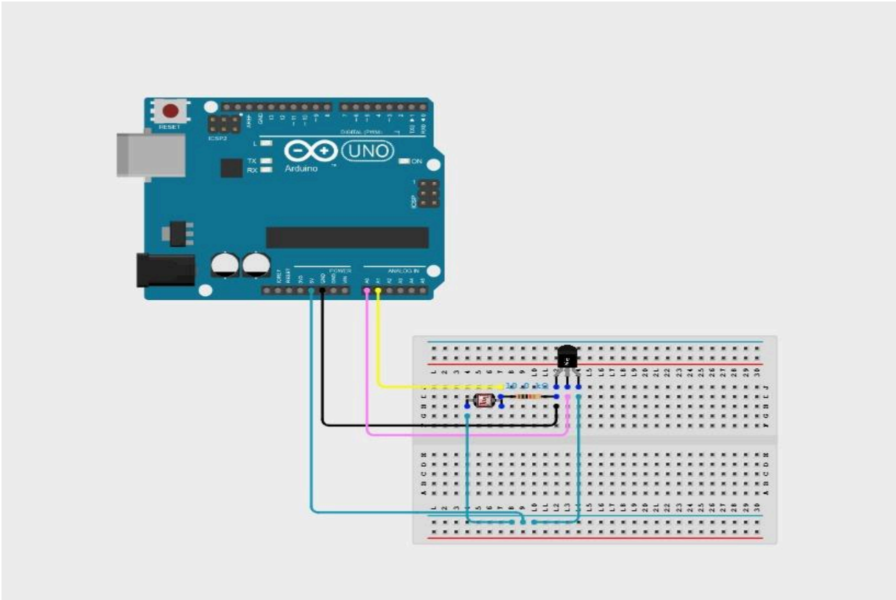


Figure 1: Schematic Diagram

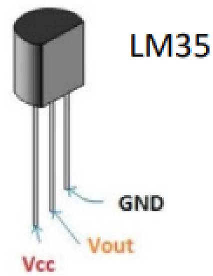


Figure 2: LM35 Connection

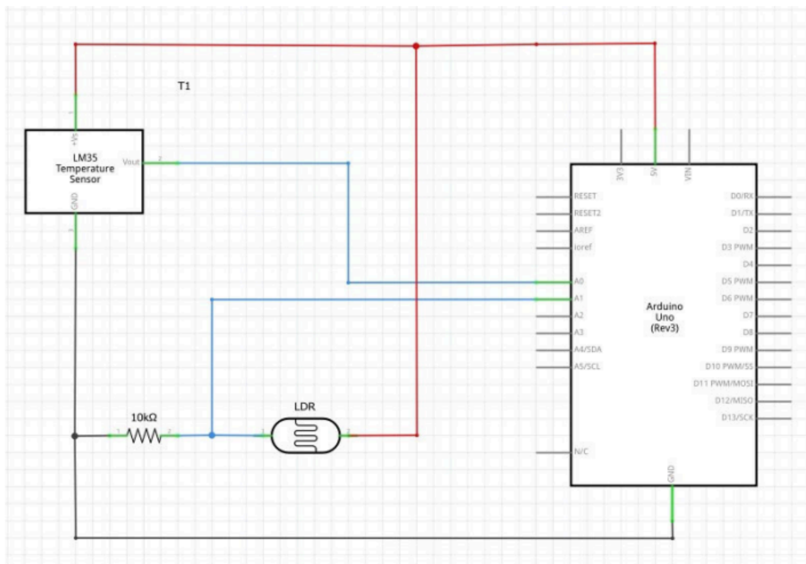


Figure 3: Circuit Diagram

4.0 METHODOLOGY

Here are the steps followed during the experiment:

1) The circuit was built as shown in **Figure 1** and the Arduino Uno was set up.
2) The Arduino IDE was launched. A code was written to allow the Arduino to read the analog signals from the LM35 and LDR sensors, convert them to digital values, and then process them into meaningful physical quantities such as temperature in Celsius and light level in percentage.
3) The code was verified and uploaded to the Arduino board.
4) The PLX‑DAQ was launched in a spreadsheet. Correct com port was selected and the baud rate was the same as the one written in the code.
5) The connect tab was pressed and the output from the sensors was generated in the spreadsheet.
6) The graph of temperature and light vs clock was produced based on data received from LM35 and LDR.

The Arduino code used:

```
// Sensor pins
const int LM_PIN = A0;    // Pin A0 for the temperature sensor (LM35)
const int LDR_PIN = A1;   // pin A1 for the light sensor (LDR)

// Variables for sensor readings
float lm_value;           // Temperature sensor reading
float temp_celsius;       // Temperature in Celsius
int ldr_value;            // Light sensor reading (0-1023)
int ldr_percentage;       // Light level as percentage (0-100)

void setup() {
    Serial.begin(9600);             // Initialize serial communication at
9600 baud rate
    Serial.println("CLEARDATA");    // Clears any existing data
    Serial.println("LABEL,CLOCK,TEMPERATURE,LIGHT");  // Set column
headers in Excel
}

void loop() {
    // Read and process temperature sensor
    lm_value = analogRead(LM_PIN);                 // Read analog voltage
from LM35
    temp_celsius = (lm_value / 1023.0) * 500.0;  // Convert analog value
to Celsius

    // Read and process light sensor
```

```
    ldr_value = analogRead(LDR_PIN);                   // Read analog
voltage from LDR

    ldr_percentage = map(ldr_value, 0, 1023, 0, 100);  // Convert raw
value to percentage



    // Send data to serial in CSV format

    Serial.print("DATA,TIME,");     // 'TIME' will be automatically
replaced by the current time in Excel logger

    Serial.print(temp_celsius);     // Print temperature value

    Serial.print(",");              // Using comma as separator instead of
period

    Serial.println(ldr_percentage); // Print light level percentage



    delay(1500);  // Wait for 1.5 seconds before next reading

}
```

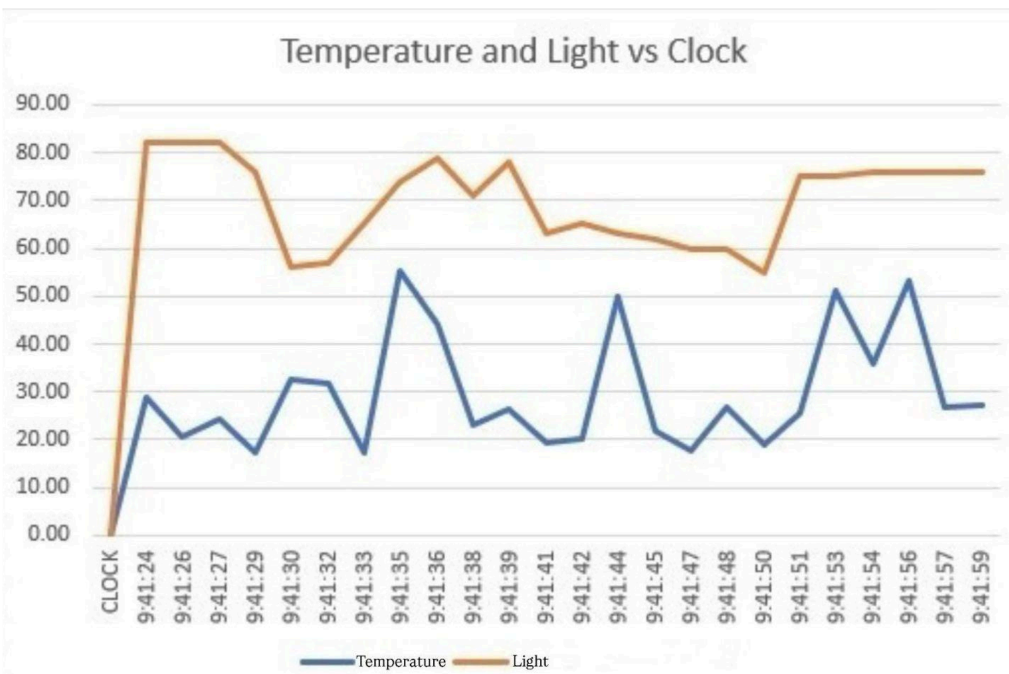
5.0 DATA COLLECTION

During the experiment, data was collected using the PLX-DAQ tool, which interfaces Arduino-based sensors with Microsoft Excel in real-time. The experimental setup includes temperature and light sensors connected to a microcontroller, with the readings transmitted and logged into the spreadsheet. Data acquisition was set at a consistent interval, capturing time (CLOCK), temperature (°C), and light intensity values (%).

CLOCK	TEMPERATURE (°C)	LIGHT (%)
9:41:24	28.84	82
9:41:26	20.53	82
9:41:27	24.44	82
9:41:29	17.11	76
9:41:30	32.75	56
9:41:32	31.77	57
9:41:33	17.11	65
9:41:35	55.23	74
9:41:36	43.99	79
9:41:38	22.97	71
9:41:39	26.39	78
9:41:41	19.55	63
9:41:42	20.04	65
9:41:44	49.85	63
9:41:45	21.99	62
9:41:47	17.60	60

9:41:48	25.88	60
9:41:50	19.06	55
9:41:51	25.42	76
9:41:53	51.32	75
9:41:54	35.68	76
9:41:56	53.27	76
9:41:57	26.88	76
9:41:59	27.37	76

Table 1: Logged Sensor Data from Arduino (Time, Temperature in °C, and Light Intensity in %)

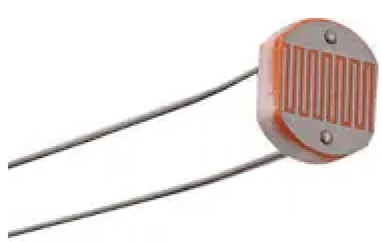


Graph 1: Temperature (°C) and Light Intensity (%) vs Clock

Instruments used for data acquisition:

1) PLX-DAQ: A Microsoft Excel add-in used to capture and log real-time data sent from the Arduino via serial communication. In this experiment, PLX-DAQ enabled direct monitoring and recording of sensor readings (light and temperature) into Excel for analysis.
2) Arduino Uno board: A microcontroller board based on the ATmega328P, used to read analog signals from the LDR and LM35 sensors and transmit the data over a serial connection to a computer running PLX-DAQ.

3) LDR : Used to measure ambient light intensity. It changes its electrical resistance based on the amount of light it receives. Its resistance decreases with increasing light, providing an analog voltage that the Arduino reads to monitor brightness levels.



4) LM35: A precision integrated-circuit temperature sensor with a linear output voltage proportional to the Celsius temperature. It was used in the experiment to measure environmental temperature and send real-time data to PLX-DAQ.

5) 10k ohm Resistor: Used in series with the LDR to form a voltage divider circuit, enabling accurate analog voltage readings corresponding to changes in light intensity.

6) Jumper wires: Used to establish electrical connections between Arduino and components on the breadboard.
7) Breadboard: A prototyping platform used to connect components without soldering, allowing for easy circuit modifications.

6.0 DATA ANALYSIS

The data collected throughout the experiment shows variations in both temperature and light intensity readings over time. These values were recorded using temperature and light sensors connected to a data acquisition system that logged the data at frequent time intervals. The purpose of this analysis is to identify trends, outliers, and correlations that may offer insight into the environmental conditions during the test and the behavior of the sensors involved.

The temperature values range significantly, from as low as 17.11°C to as high as 55.23°C, indicating fluctuations that may be influenced by changing environmental factors or sensor responses to different heat sources. Such extreme variations in short intervals, like the sharp increase from 17.11°C at 9:41:33 to 55.23°C at 9:41:35, may suggest the sensor was exposed to an external heat source or there were inconsistencies in the measurement process. Meanwhile, the light intensity values are recorded as percentages and range between 55% and 82%. Although there is variation in light levels, the changes are more gradual and less erratic compared to temperature. This may imply a more stable lighting environment or a sensor with higher resistance to environmental noise.

To further interpret the data, a line graph was plotted with time on the x-axis and both temperature and light intensity on the y-axis. The graph highlights that while light intensity remains relatively consistent, temperature experiences sharper rises and falls. A correlation between light and temperature was not clearly established based on this data alone, as several instances show temperature spikes without corresponding changes in light levels. For example, the temperature rise to 53.27°C at 9:41:56 occurred while the light remained steady at 76%.

In a broader context, this data helps validate the responsiveness and accuracy of the sensors used. It also illustrates environmental variability over time, which is important for systems where environmental monitoring or adaptive control is required. For more robust conclusions, statistical analysis such as calculating the mean temperature (≈30.11°C) and mean light intensity (≈69.7%), standard deviation, and correlation coefficients could be applied. These would help confirm if patterns exist or if readings are purely coincidental or due to sensor noise. Ultimately, the analysis contributes to understanding how real-world conditions impact sensor data and provides a foundation for future improvements in sensor placement, data filtering, or experimental design.

The data collected is significant as it directly supports the experiment's objective of monitoring environmental conditions, specifically temperature and light intensity, using real-time data acquisition through PLX-DAQ. The wide fluctuations in temperature readings demonstrate the sensor's responsiveness to environmental changes, while the relatively stable light intensity values indicate consistent lighting during the experiment. These results validate the effectiveness and reliability of the sensor system and the data logging setup, confirming that the system can successfully capture, transmit, and visualize environmental data in real-time. This is essential for applications that require continuous monitoring and data-driven responses, such as automated environmental control, smart systems, or further scientific analysis.

7.0 RESULTS

The results of the experiment confirm the successful demonstrated capability of an Arduino Uno to acquire and log analog data from temperature and light sensors in real time. The LM35 temperature sensor provided consistent voltage readings that were accurately converted to temperature values in degrees Celsius. Similarly, the LDR, when used in a voltage divider configuration, generated light intensity readings that were mapped to percentage values.

As recorded in **Table 1: Logged Sensor Data from Arduino (Time, Temperature in °C, and Light Intensity in %)**, temperature values ranged from approximately 17.11°C to 55.23°C, while light intensity values varied between 55% and 82%. These readings were plotted using Excel to generate line graphs, which clearly displayed the variations over time. **Graph 1** shows a line graph of temperature and light intensity vs. time. The trends observed in both graphs reflect real environmental fluctuations, such as changes in light exposure or proximity to heat sources.

The Arduino system responded reliably to sensor inputs and transmitted the data effectively to PLX-DAQ. The real-time data logging and visualization confirmed that the setup functioned as intended, meeting the primary objectives of the experiment. The results validate the Arduino-based DAQ system as a useful tool for simple environmental monitoring applications.

As shown in the **Figure 4** below, the data is systematically organized in tabular format with clear columns for each parameter. To enhance readability and interpretability, a line graph titled "Temperature and Light vs Clock" is also included. This graph visually represents the fluctuations in temperature and light over time, offering a straightforward way to observe trends and anomalies.

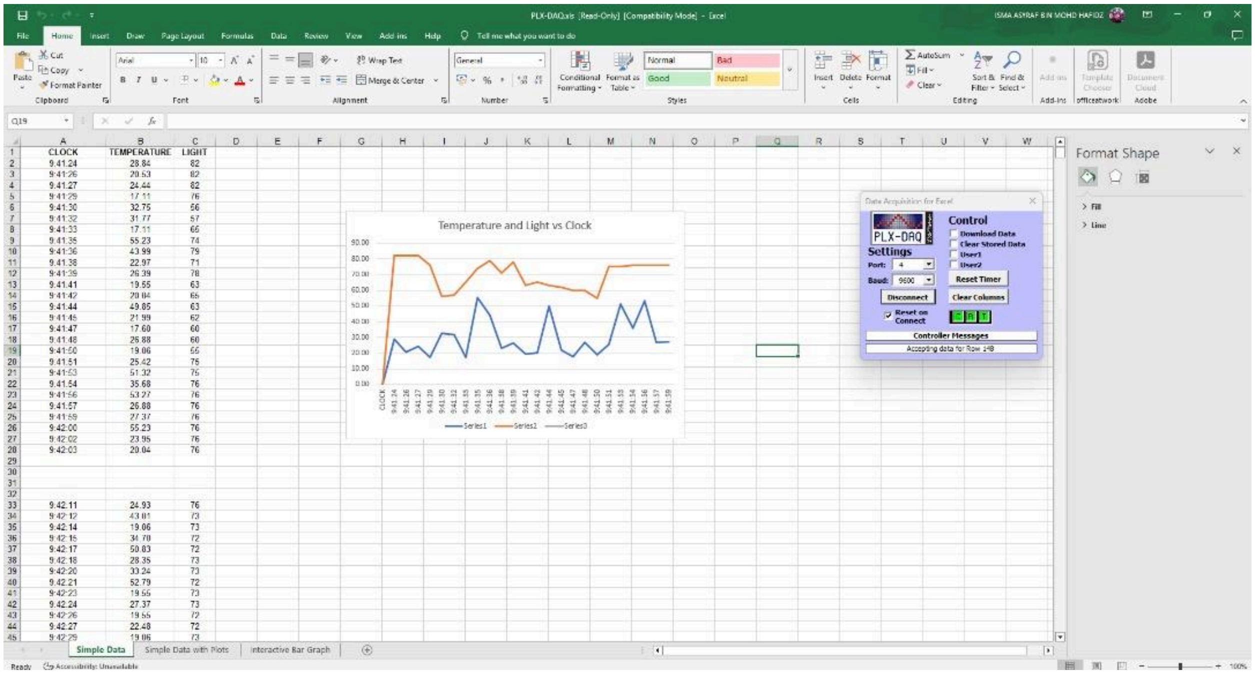


Figure 4: Spreadsheet displaying real-time data acquisition using PLX-DAQ.xls

Video for Task 6:

https://github.com/amalinblqs/MCTA3203_GROUP_9/blob/main/Lab%206/video_task.mp4

8.0 DISCUSSION

The results of this experiment successfully show that the Arduino Uno was effective in collecting and transmitting real-time sensor data from the LM35 temperature sensor and the LDR. The data collected closely followed expected trends, with temperature readings rising or falling in response to environmental changes and light intensity values varying with ambient lighting. These outcomes affirm the expected behavior of both sensors and demonstrate the Arduino's capability to function as a simple data acquisition system.

The implications of the results highlight the practical application and potential of using Arduino-based systems for real-time environmental monitoring. The accurate and consistent readings from both the LM35 and LDR sensors show that simple, low-cost hardware can be effectively used to measure and track temperature and light conditions in a given environment. This kind of setup is particularly useful in educational, experimental, and even commercial scenarios where data logging and environmental feedback are essential, such as in smart home systems, agricultural monitoring, and classroom learning projects.

The ability to visualize sensor data through PLX-DAQ in real time provides a clear advantage in understanding environmental dynamics and responding accordingly. For instance, observing a sharp increase in temperature or drop in light could trigger automated actions in a more advanced setup, like turning on fans or lights. Thus, the experiment not only confirms the reliability of the sensors but also underlines the potential for developing more complex systems using this foundational knowledge.

However, there were some minor discrepancies between expected and observed outcomes. For instance, occasional sudden spikes or dips in temperature readings may not have accurately reflected actual changes in room temperature. These inconsistencies could be attributed to sensor noise, ADC resolution limits, or brief fluctuations in power supply.

Several potential sources of error were identified in this experiment. These include electrical noise, loose jumper wire connections, variations in sensor calibration, and limitations in the analog-to-digital conversion process of the Arduino, which has a 10-bit resolution, mapping values from 0–1023. Additionally, the LM35 sensor can be sensitive to heat from nearby electronic components, which may have influenced its accuracy. Another limitation was the sampling interval of 1.5 seconds; faster or slower sampling could provide different insights depending on the use case.

Despite these minor issues, the experiment was successful in demonstrating key concepts of sensor interfacing, analog signal conversion, and real-time data logging. The results align well with the objectives and offer a solid foundation for understanding basic environmental monitoring using embedded systems.

9.0 CONCLUSION

This experiment successfully demonstrated the use of an Arduino Uno as a simple data acquisition (DAQ) system for collecting and logging temperature and light intensity data using LM35 and LDR sensors. The main findings show that the Arduino accurately read and converted analog sensor signals into digital data, which was then transmitted and recorded in real time using PLX-DAQ. The recorded data showed clear patterns corresponding to actual changes in the environment, confirming the proper functionality and responsiveness of the sensor setup. The temperature readings, expressed in degrees Celsius, and the light intensity, measured as a percentage, showed consistent variations corresponding to actual changes in the surroundings. These results confirmed that the sensor setup was functioning correctly and that the Arduino could serve as an effective data acquisition system.

The results supported the initial hypothesis that Arduino could be effectively used to capture and log sensor data in a structured and meaningful format. The expected trends in temperature and light readings were observed, with only minor fluctuations that were likely due to environmental noise or hardware limitations. These variations did not significantly affect the overall accuracy of the system, reinforcing its reliability for basic monitoring tasks.

Beyond the immediate outcomes, the experiment highlights the broader applicability of Arduino-based systems in real-world scenarios such as home automation, weather monitoring, smart farming, and educational tools for teaching electronics and data analysis. The insights gained from this experiment can serve as a foundation for more advanced projects involving multiple sensors, wireless communication, or control systems, making it a valuable learning experience in embedded systems and sensor interfacing.

10.0 RECOMMENDATIONS

Based entirely on the experiment results and the challenges encountered, several recommendations can be made to improve future versions of this Arduino-based data acquisition system. One key issue was the problem with PLX-DAQ integration, where the expected pop-out GUI window for the spreadsheet failed to appear. This caused delays and made data logging frustrating. It is recommended to ensure that Excel macros are enabled, PLX-DAQ is properly installed, and that security settings in Excel are correctly configured before starting the experiment. Creating a simple step-by-step PLX-DAQ setup guide would greatly benefit future users.

Calibrating the sensors is also an important step that should not be overlooked. Pre-calibrating the LM35 and LDR sensors can significantly improve accuracy, as some readings were inconsistent during testing. Additionally, better sensor placement such as keeping sensors away from direct heat or interference can help reduce sudden temperature spikes. One key insight gained was that sensor readings can fluctuate unexpectedly if influenced by nearby electronics or inconsistent lighting.

Also, adjusting the sampling interval based on the experiment's sensitivity can help capture more relevant data. For instance, quick changes in light or temperature may go unnoticed if readings are too slow. On the software side, incorporating filtering methods like a moving average can help smooth out noisy data. For more critical or sensitive applications, using higher-resolution ADCs or upgrading to microcontrollers with better analog precision could enhance reliability.

These recommendations are intended to improve the accuracy, usability, and overall experience of working with Arduino-based sensor systems. The lessons learned from this experiment, especially regarding setup challenges and sensor behavior, should be helpful to future students tackling similar projects.

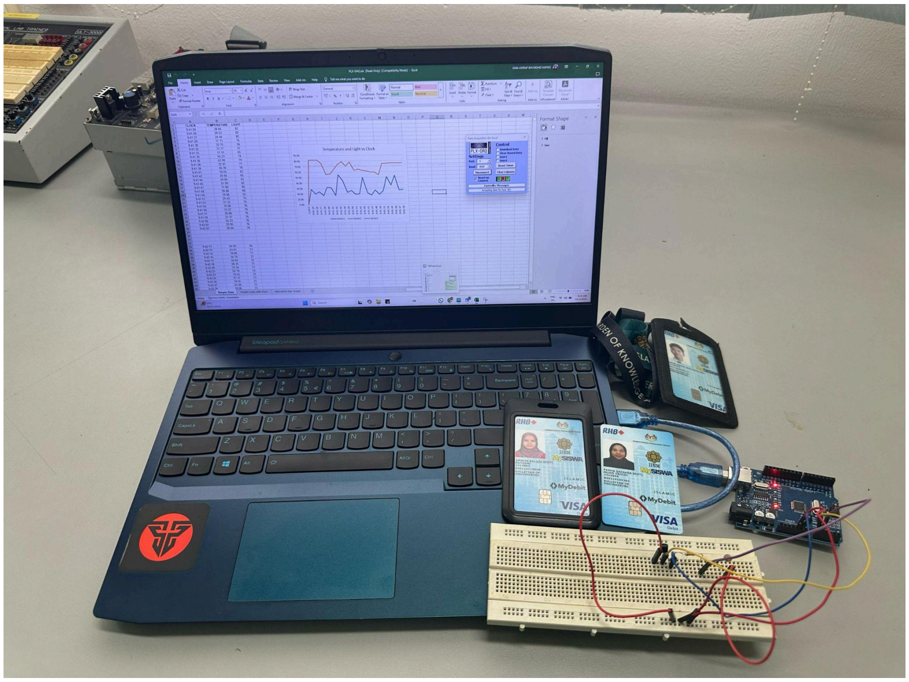
11.0 REFERENCES

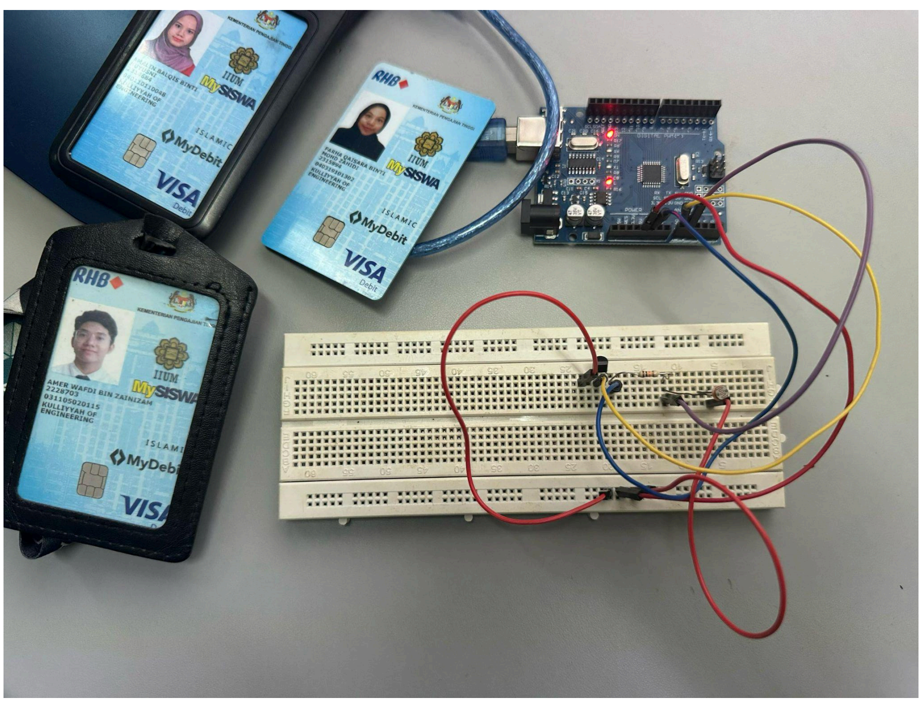
1. Maharmi, B. (2022). Water flow measurement-based data acquisition using Arduino microcontroller and PLX-DAQ software. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, *8*(1), 107–118. https://www.researchgate.net/publication/362989044_Water_Flow_Measurement-Based_Data_Acquisition_Using_Arduino_Microcontroller_and_PLX-DAQ_Software_Corresponding_Author

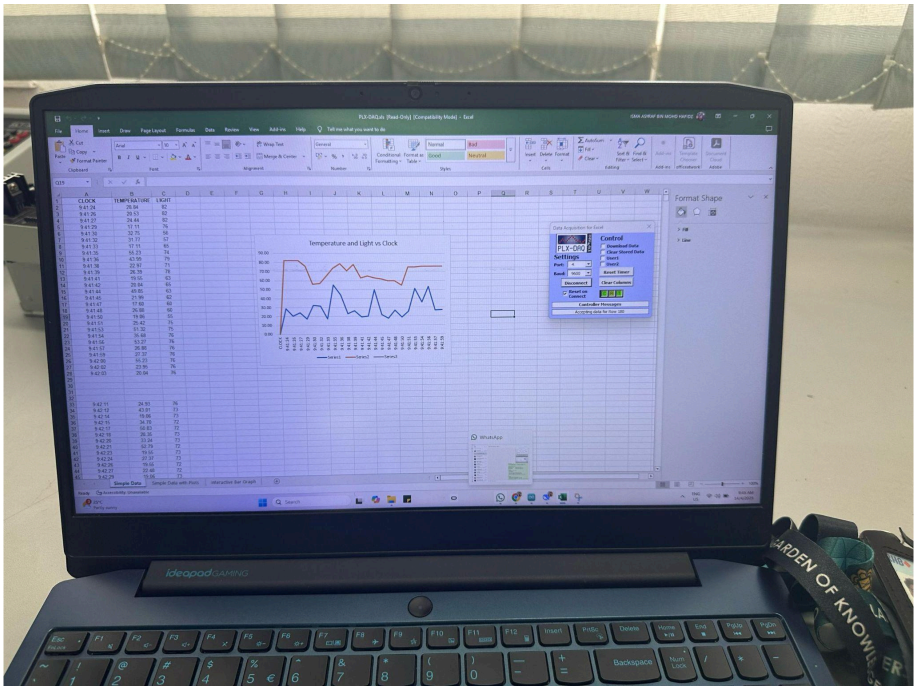
2. Josephine, A. R., Liem, F. S., Santoso, G. A., & Prabowo, N. K. (2021). ChemDuino and PLX-DAQ integration for affordable temperature data acquisition: Health and school laboratory monitoring system. *Research Archive of Rising Scholars*. https://research-archive.org/index.php/rars/preprint/view/771

3. SDE Electronics. (2020, March). PLX-DAQ Arduino tutorial. *SDE Electronics*. https://sdevelectronics.blogspot.com/2020/03/plx-daq-arduino-tutorial.html

APPENDICES






Temperature and Light vs Clock
PLX-DAQ
ideapadGAMING

ACKNOWLEDGMENTS

First of all, we would like to express our deepest appreciation to all those who provided us the possibility to complete this report. We would like to express our sincere gratitude to our lecturer, Assoc. Prof. Eur. Ing. Ir. Ts. Gs. Inv. Dr. Zulkifli Bin Zainal Abidin and Dr. Wahju Sediono for their invaluable guidance and support throughout this lab. Their insightful explanations and encouragement greatly enhanced our understanding of the concepts covered.

Next, a special thanks goes to our own teammates, Wafdi, Amalin and Farha, who helped to build up the circuit and set up the programming code using Arduino IDE, and also successfully completed this experiment. Last but not the least, we would like to thank everyone, especially our classmates who are willingly helping us out in the lab experiment directly or indirectly.

STUDENT'S DECLARATION

<u>Certificate of Originality and Authenticity</u>

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: *Wafdi* Read [/]
Name: Amer Wafdi bin Zainizam Understand [/]
Matric Number: 2228703 Agree [/]
Contribution: Abstract, materials and equipment & methodology

Signature: *Amalin* Read [/]
Name: Amalin Balqis binti Ruyusni Understand [/]
Matric Number: 2314684 Agree [/]
Contribution: Introduction, data collection, data analysis, results & discussion

Signature: *farha*

Read [/]

Name: Farha Qaisara binti Mohd Zahidi

Understand [/]

Matric Number: 2315994

Agree [/]

Contribution: Experimental setup, conclusion & recommendations