MECHATRONICS SYSTEM INTEGRATION (MCTA 3203)

SEMESTER 2 2024/2025

WEEK 5: PLC INTERFACING WITH MICROCONTROLLER

SECTION 1

GROUP 5

LECTURER: ZULKIFLI BIN ZAINAL ABIDIN & WAHJU SEDIONO

Date of Experiment: Monday, 7 April 2025

Date of Submission: Monday, 12 April 2025

NO.	GROUP MEMBERS	MATRIC NO.
1.	BUSHRA BINTI A. RAHIM	2318514
2.	AUFA SIDQY BINIT MOHD SIDQY	2310542
3.	NUR HAMIZAH BINTI MUHAMAD HAZMI	2319132

ABSTRACT

This experiment explores the fundamentals of Programmable Logic Controller (PLC) interfacing with microcontrollers using OpenPLC software and an Arduino board. The objective was to simulate and implement a basic Start-Stop control circuit and LED blinking system using ladder logic diagrams. OpenPLC Editor was used to create, compile, and simulate the ladder diagrams before transferring them to the Arduino microcontroller for physical testing. The successful implementation of both the blinking LED and the Start-Stop control circuit demonstrates the feasibility of using open-source PLC environments with low-cost microcontrollers. This lab reinforces key concepts in industrial automation and control logic, while providing hands-on experience in ladder diagram programming, variable configuration, and hardware-software integration.

TABLE OF CONTENTS

1.0 INTRODUCTION

Programmable Logic Controllers (PLCs) are widely used in industrial automation to control machinery and processes through logical operations. They offer reliability, modularity, and ease of programming using ladder logic, a graphical language resembling electrical relay logic diagrams. In this lab, OpenPLC, a free and open-source platform for PLC programming was introduced to simulate and upload control logic directly onto a microcontroller.

The primary objective was to interface the OpenPLC Editor with an Arduino board to implement two key tasks: a blinking LED circuit and a Start-Stop control circuit using push buttons. The experiment required creating ladder diagrams, defining variables, simulating logic, associating I/O pins, and transferring the compiled program to the microcontroller via a USB interface. This exercise bridges the gap between industrial control systems and embedded microcontroller platforms, providing students with foundational skills in PLC programming and real-time control system integration.

2.0 MATERIALS AND EQUIPMENT

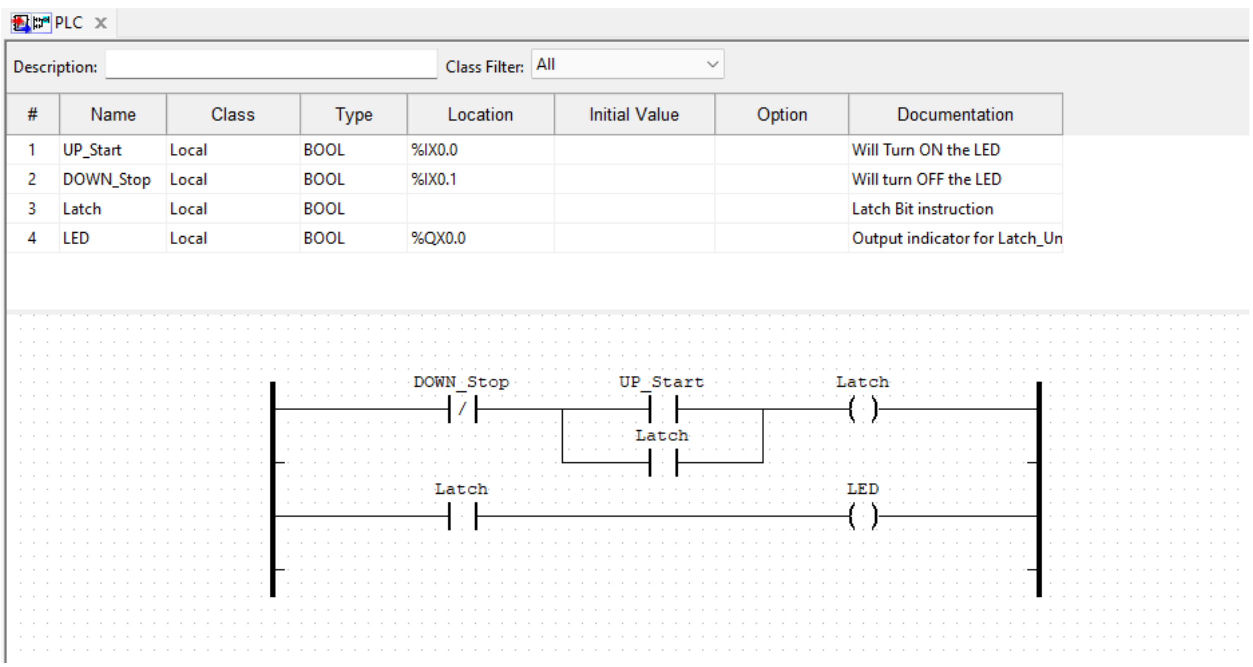
Hardware Components:

- Arduino board (e.g., Arduino UNO or Arduino Mega)
- LED (1 unit, for output indication)
- Push Button Switches (2 units, used for Start and Stop control)
- Resistors (typically 220Ω or 330Ω for LED current limiting, and 10kΩ for pull-down resistors on buttons)
- Breadboard (for prototyping and wiring connections)
- Jumper wires (male-to-male, for circuit connections)
- USB cable (to connect Arduino board to the computer)
- Computer/Laptop (with USB ports for uploading ladder diagram code)

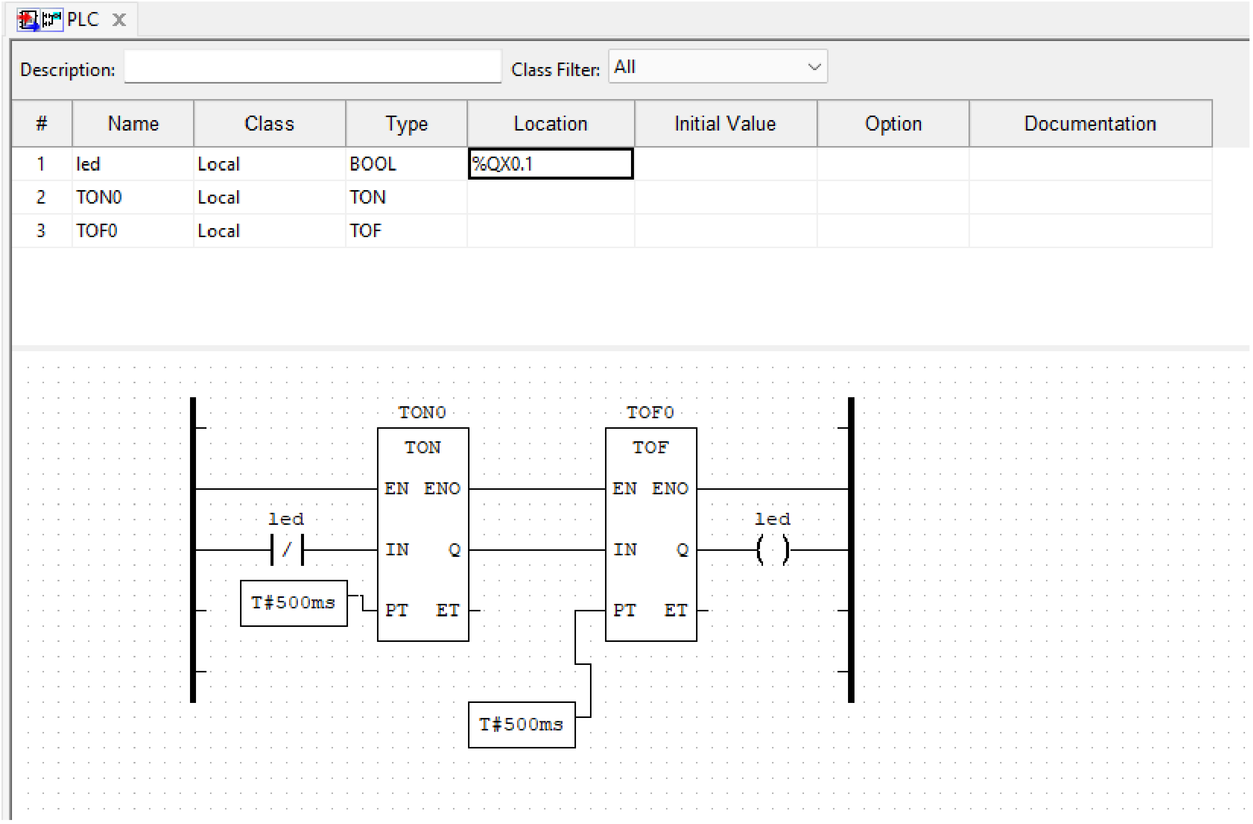
Software Tools:

- OpenPLC Editor (for designing, simulating, and compiling ladder logic diagrams)
- OpenPLC Runtime (for uploading and running PLC programs on Arduino)
- Arduino drivers (pre-installed with Arduino IDE or OpenPLC Runtime setup)

3.0 EXPERIMENTAL SETUP



#	Name	Class	Type	Location	Initial Value	Option	Documentation
1	UP_Start	Local	BOOL	%IX0.0			Will Turn ON the LED
2	DOWN_Stop	Local	BOOL	%IX0.1			Will turn OFF the LED
3	Latch	Local	BOOL				Latch Bit instruction
4	LED	Local	BOOL	%QX0.0			Output indicator for Latch_Un



Software Setup:

- Installed OpenPLC Editor f
- Created a new project using Ladder Diagram (LD) programming language.
- Designed two ladder diagrams:
  - A blinking LED circuit using a contact and coil.
  - A Start-Stop control circuit using two push buttons and an output coil.
- Defined variables for inputs and outputs (e.g., %IX0.0 for inputs, %QX0.0 for outputs).
- Assigned variables to appropriate Arduino digital pins based on physical addressing documentation.

Hardware Setup:

- Used an Arduino board (e.g., UNO or Mega) connected to a PC via USB cable.
- Built the circuits on a breadboard using jumper wires.
- LED connected to an output pin (e.g., pin 14), with a current-limiting resistor.
- Push Button 1 (Start) connected to a digital input pin with a pull-down resistor.
- Push Button 2 (Stop) connected to another input pin with a pull-down resistor.
- Ensured correct COM port was selected during the upload process in OpenPLC Editor.

Execution and Testing:

- Compiled and simulated the ladder diagrams within OpenPLC Editor.
- Uploaded the compiled programs to the Arduino using “Transfer to PLC”.
- Verified that the LED:
    - Blinks at a set interval for the blinking LED task.
    - Turns on/off according to the state of the push buttons in the Start-Stop control task.

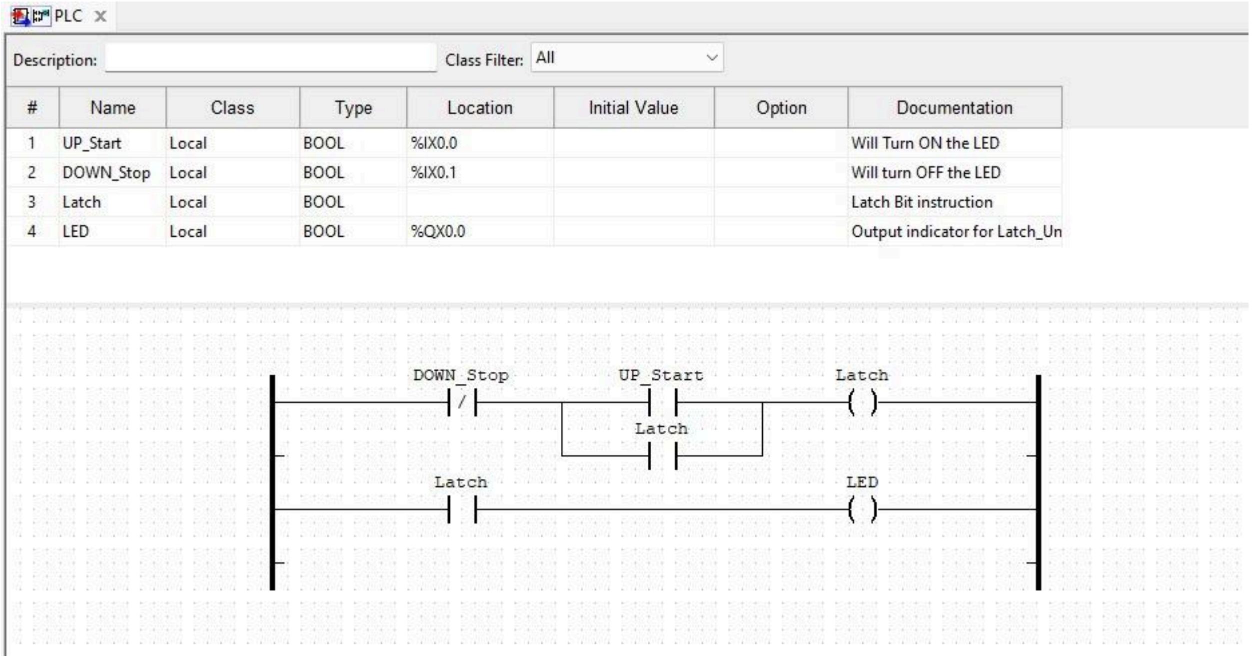
4.0 METHODOLOGY

Step-by-Step Procedure (Past Tense)

1. **Ladder Diagram Design:**
    - The Start-Stop circuit was designed using the **OpenPLC Editor**.
    - A **Normally Open (NO)** contact was assigned for the Start button, a **Normally Closed (NC)** contact for the Stop button, and an **output coil** for the load (LED).

2. **PLC Variable Assignment:**
    - The following I/O variables were assigned in the OpenPLC environment:
        - `%IX0.0` → Start Button (Digital Input Pin 2)
        - `%IX0.1` → Stop Button (Digital Input Pin 3)
        - `%QX0.0` → Output (LED - Digital Output Pin 13)

3. **Ladder Logic Implementation:**

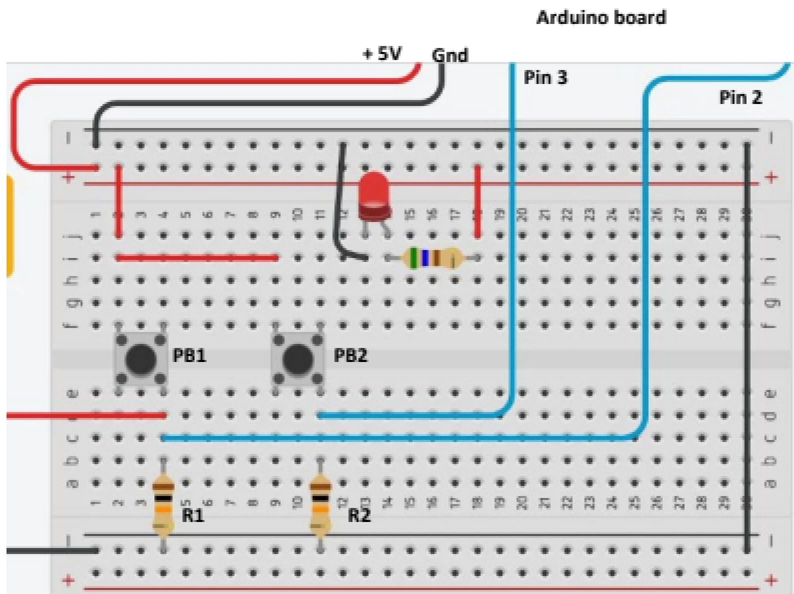


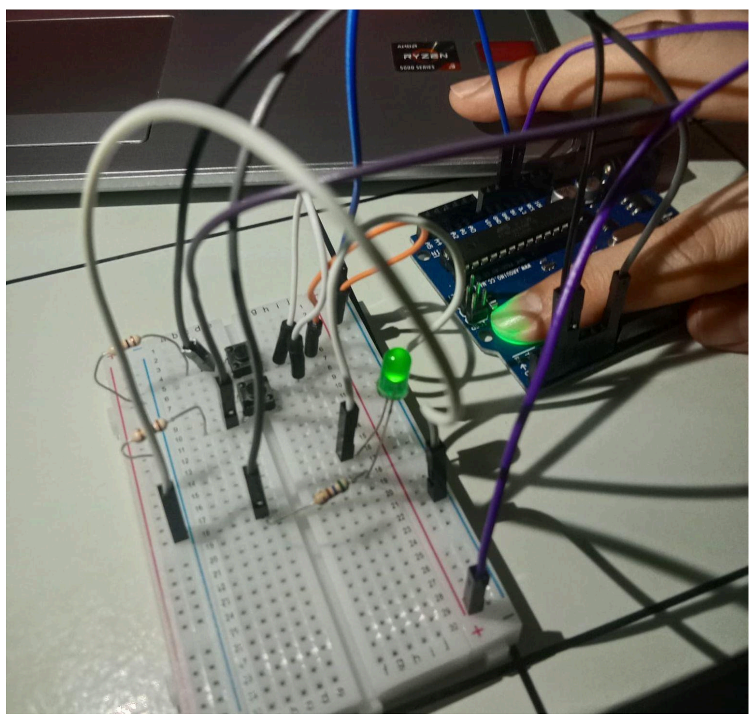
PLC

Description: | Class Filter: All

#	Name	Class	Type	Location	Initial Value	Option	Documentation
1	UP_Start	Local	BOOL	%IX0.0			Will Turn ON the LED
2	DOWN_Stop	Local	BOOL	%IX0.1			Will turn OFF the LED
3	Latch	Local	BOOL				Latch Bit instruction
4	LED	Local	BOOL	%QX0.0			Output indicator for Latch_Un

4. **Compilation and Export:**
   - The ladder logic was compiled using the "Build" function in the OpenPLC Editor.
   - The program was then saved and prepared for upload.
5. **Hardware Setup:**
   - The physical components were arranged on a breadboard.
   - The Start button (NO) was connected to **Pin 2**, and the Stop button (NC) to **Pin 3**.
   - An LED, in series with a 220Ω resistor, was connected to **Pin 13**.
   - 10kΩ pull-down resistors were added to the input pins to ensure signal stability.


Arduino board
+ 5V
Gnd
Pin 3
Pin 2
PB1
PB2
R1
R2



6. **Arduino Configuration:**
   - The **OpenPLC runtime firmware** was uploaded to the Arduino Uno using the Arduino IDE.
   - The correct COM port was selected based on the device manager.
7. **Program Upload:**
   - The compiled ladder logic was uploaded to the Arduino via the **OpenPLC Web Interface**.
   - The runtime was started from the web dashboard.
8. **Testing:**
   - The Start button was pressed to activate the LED.
   - The LED remained ON after releasing the Start button (latched).
   - The Stop button was pressed to turn the LED OFF.

5.0 DATA COLLECTION

Test scenario	Start button	Stop button	Output (LEC)
Press start only	Pressed	Released	ON
Hold start	Held	Released	ON
Release start	Released	Released	ON
Press Stop	Released	Pressed	OFF
Press start again	Pressed	Released	ON

6.0 DATA ANALYSIS

The data shows that the ladder logic functions as expected:

- Pressing the Start button energizes the output coil (LED turns on).
- The circuit maintains the ON state (latched) even after releasing Start.
- Pressing the Stop button de-energizes the coil (LED turns off).
- The logic mimics a standard industrial Start-Stop motor control setup.

7.0 RESULT

The Start-Stop control system using OpenPLC was successfully implemented:

- The ladder diagram was created and compiled without error.
- The program was deployed to the Arduino board.
- Hardware interaction behaved as expected, proving the correctness of the control logic.

8.0 DISCUSSION

The experiment effectively demonstrated the use of ladder diagrams to implement simple automation logic on microcontrollers. The OpenPLC platform proved to be an efficient tool for bridging industrial PLC logic and Arduino hardware. One notable benefit is the flexibility to simulate and test before hardware deployment. Potential improvements include:

- Implementing debounce logic to prevent multiple triggers from mechanical switches.
- Using relays or transistors for higher power loads instead of just LEDs.

9.0 CONCLUSIONS

The experiment demonstrated that a Start-Stop Control Circuit can be effectively implemented using ladder diagram program on OpenPLC and executed through Arduino board. The results show that the system correctly responded to the Start-Stop Control inputs, latching the output when started and deactivating it when stopped, same as how the standard industrial motor control logic system works.

These results also highlight the value of OpenPLC as a bridge between theoretical PLC programming and practical hardware control. Such systems are widely used in automation across manufacturing, robotics, and smart home applications. Understanding how to design and implement control logic using affordable platforms like Arduino expands access to industrial-level automation for education, prototyping, and small-scale operations.

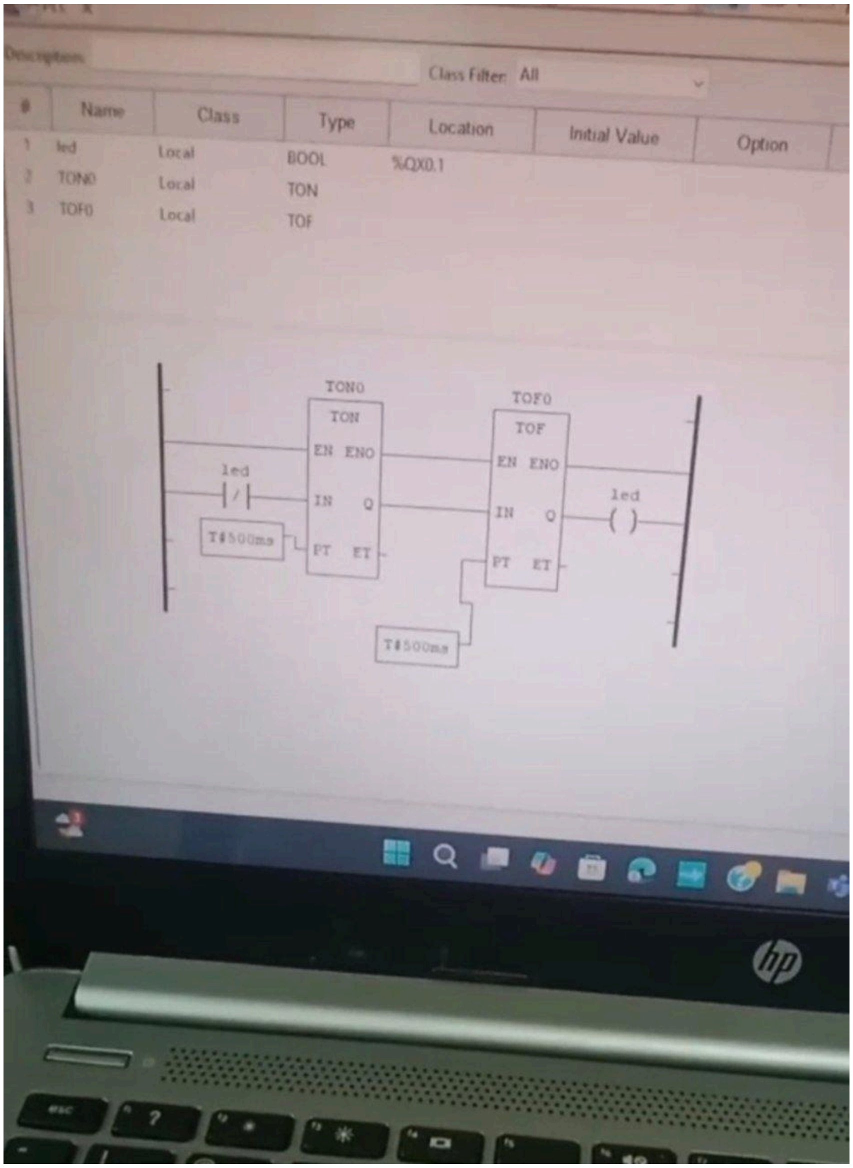
10.0 RECOMMENDATIONS

Recommendations for future iterations of the experiment include implementing debounce logic to avoid multiple triggers from mechanical switches, ensuring more reliable control, and using relays or transistors for handling higher power loads instead of just LEDs. This will allow for safer and more practical simulations of real-world industrial control systems, as well as better prepare students for dealing with higher-voltage applications.

11.0 REFERENCES

1. Storr, W. (2022, August 3). *Transistor as a switch - using transistor switching*. Basic Electronics Tutorials. https://www.electronics-tutorials.ws/transistor/tran_4.html
2. Arshad, B. (2025, January 12). *Different types of relays and their applications*. Engineers Guidebook. https://engineersguidebook.com/different-types-of-relays-their-application/
3.

12.0 APPENDICES



13.0 ACKNOWLEDGEMENT

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 qno 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: — Read [/]
Name: BUSHRA BINTI A. RAHIM — Understand [/]
Matric Number: 2318514 — Agree [/]

Signature: aufa — Read [/]
Name: AUFA SIDQY BINTI MOHD SIDQY — Understand [/]
Matric Number: 2310542 — Agree [/]

Signature: — Read [/]
Name: NUR HAMIZAH BINTI MUHAMAD HAZMI — Understand [/]
Matric Number:2319132 — Agree [/]