MECHATRONICS SYSTEM INTEGRATION (MCTA 3203)

SEMESTER 2 2024/2025

WEEK 5: PLC 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, 7 April 2025

Date of Submission: Monday, 14 April2025

ABSTRACT

This experiment explores the integration of PLCs with microcontrollers using OpenPLC Editor and an Arduino board. A basic LED blinking program is created using ladder logic, simulated, and uploaded to the Arduino. The circuit is then modified using a timer block to control the blink interval. Finally, a Start-Stop control circuit is developed, simulated, and tested with physical components, demonstrating practical PLC-microcontroller interfacing.

TABLE OF CONTENTS

INTRODUCTION

Programmable Logic Controllers (PLCs) play a key role in industrial automation by providing reliable control over electrical and mechanical systems. In this lab, OpenPLC Editor is used to create and simulate ladder logic programs, which are then uploaded to an Arduino microcontroller to control external components. The first task involves programming a basic LED blink sequence, followed by modifying the program to include a timer for adjustable blink intervals. The final task involves designing a Start-Stop control circuit using push buttons and an LED. Through these exercises, students gain hands-on experience in PLC programming, simulation, and hardware integration, building foundational skills for real-world automation applications.

MATERIALS AND EQUIPMENTS

1. Arduino Mega 2560
2. Push Buttons
3. LEDs
4. Resistors
5. Jumpers
6. Breadboard

EXPERIMENTAL SETUP

For the button and LED connections:

1. Connect the anode of the LED to a resistor, and then to pin D14 on the Arduino.
2. Connect the cathode of the LED to the GND pin on the Arduino.
3. Connect one leg of the first button to pin A13 on the Arduino.
4. Connect one leg of the second button to pin A14 on the Arduino.

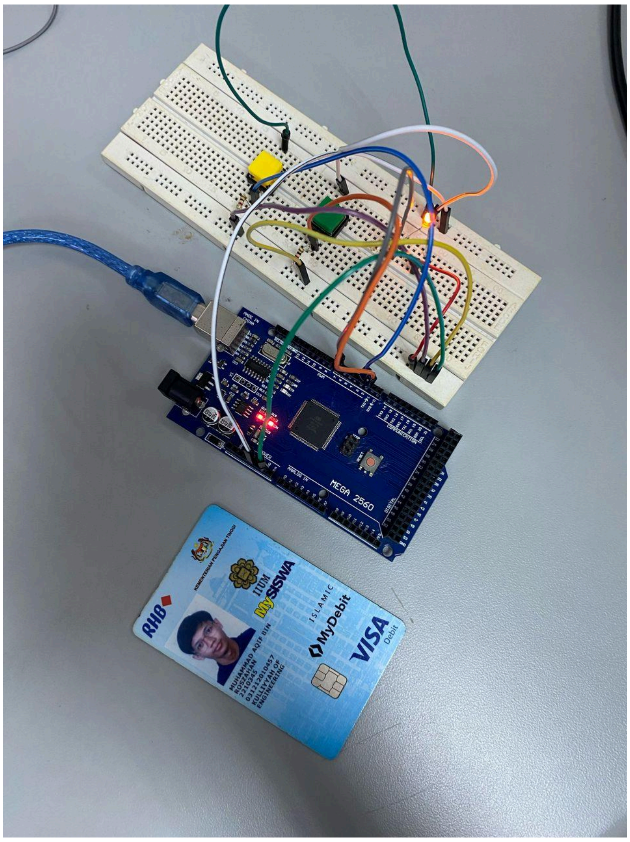


Figure 1: Hardware Setup

METHODOLOGY

1. Setup the Arduino Mega 2560 with the LEDs, resistors and push buttons
2. Implement the PLC in OpenPLC
3. UP_Start will turn on the LED
4. DOWN_Stop will turn off the LED

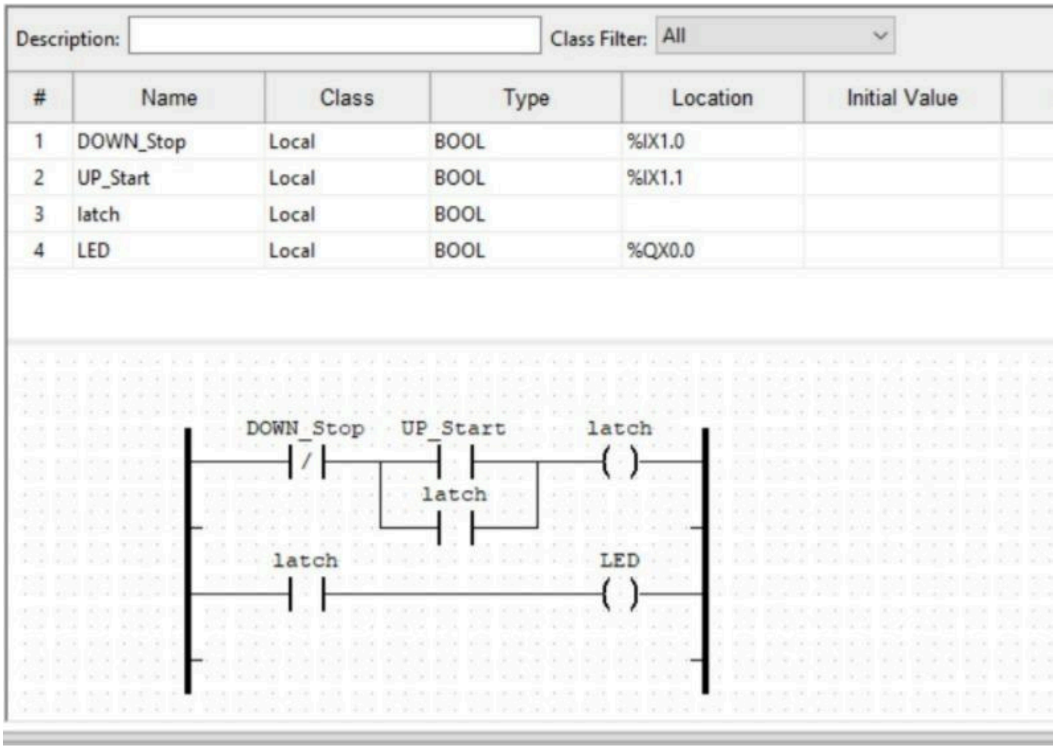


Figure 2: OpenPLC Implementation

DATA COLLECTION

UP_Start	DOWN_Stop	Led
-	-	off
pushed	-	on
released	-	on
-	pushed	off

Table 1: OpenPLC Implementation

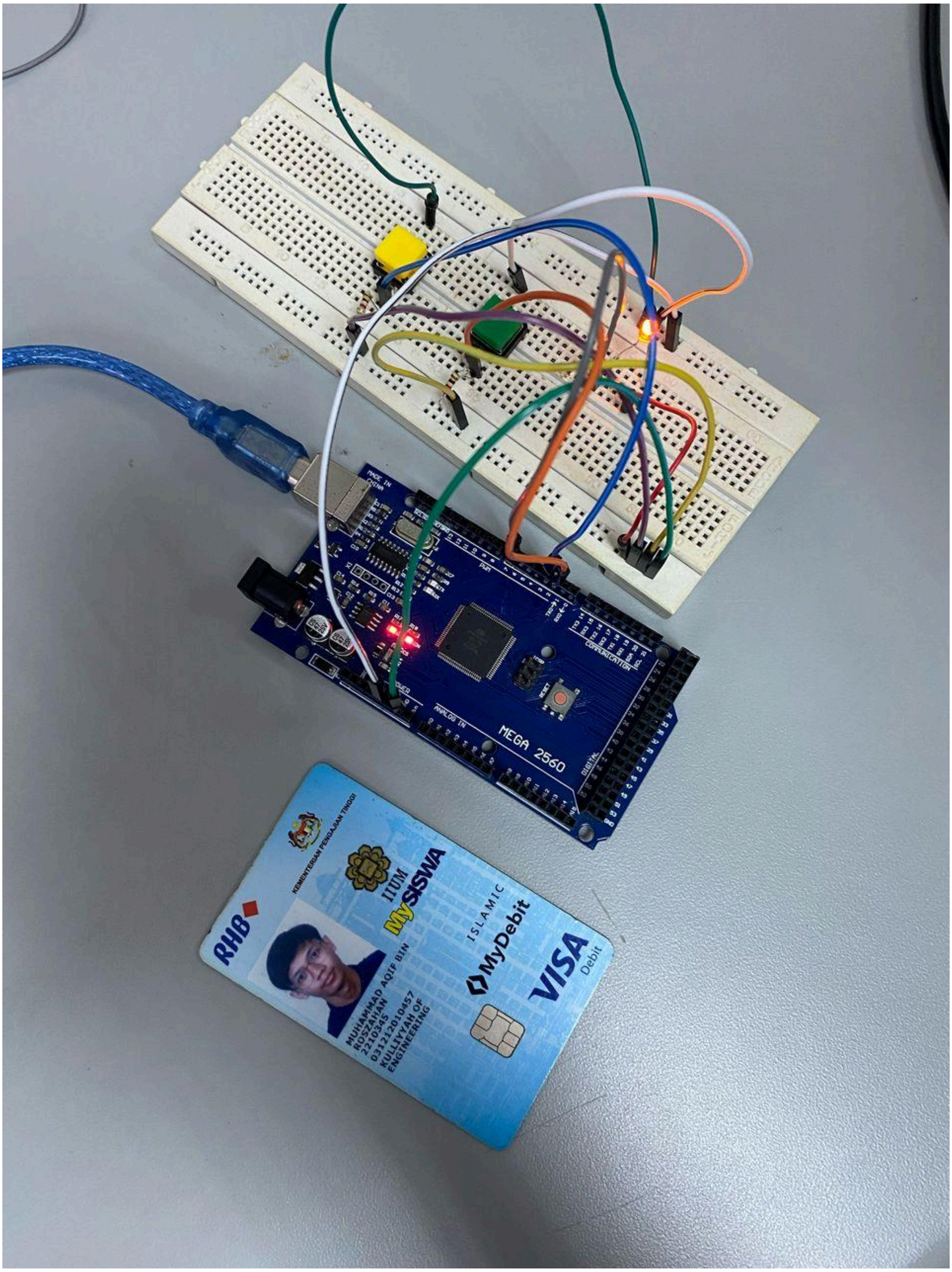
DATA ANALYSIS

Based on the collected data:

1. Pressing the Start Push Button (UP_Start) sends a TRUE signal to the circuit, allowing current to pass through and activating the latch, which turns the LED ON.
2. Once the Start button is released, it stops sending the TRUE signal, but the LED stays ON due to the latching mechanism. The latch provides an alternate path for current, keeping the circuit active even after the button is no longer pressed.
3. Pressing the Stop Push Button (DOWN_Stop) interrupts the latch, stopping the current flow and turning the LED OFF, effectively resetting the circuit.

This operation is known as latching mode, where the circuit maintains its current state (ON or OFF) until a deliberate action (like pressing the Stop button) changes it. This type of control is commonly used in systems like motor starters or toggle-based switches.

RESULT


MEGA 2560
KEMENTERIAN PENGAJIAN TINGGI
RHB
IIUM
MySISWA
ISLAMIC
MyDebit
VISA
Debit
MUHAMMAD AQIF BIN
ROSZAIAN
2210345
031212010457
KULLIYYAH OF
ENGINEERING

DISCUSSION

This experiment demonstrates the basic principles of a Programmable Logic Controller (PLC) and how to interface it to an Arduino Mega controlled by OpenPLC Editor software. The objective was to create a simple start-stop circuit where the input device is a push-button that controls the on/off state of the output device, LED. By integrating the software part with basic electric and hardware components, this experiment introduces the key concept of ladder logic programming.

- **Software**

Ladder logic is a commonly used programming language in PLC applications, and OpenPLC provides an easy-to-use environment for doing so. Adding power rails, each with four pins on the left and right sides, is the first step in making a new schematic. The functional variables—"Turn Off LED," "Turn On LED," "Latch Bit Instruction," and the output indicator—are then added. The ladder diagram shows "UP_Start" and "Latch" as normally open (NO) contacts, while the "Down_Stop" input is represented by a negated contact. The "LED" and "Latch" are set up as standard coil parts. A typical start-stop control circuit is replicated by the arrangement of these components. After completing the diagram, the code is compiled, the appropriate board and COM port are selected, and the sketch is uploaded to the Arduino.

- **Electrical**

The circuit design follows basic electrical principles to ensure proper current flow. A 10kΩ resistor is connected to both push buttons to stabilize its input signal. The LED is connected to Pin 7 and PB1 and PB2 are connected to Pin 3 and Pin 2 respectively to ensure their different functions. Lastly, 5V and GND are connected across all the components.

- **Hardware**

A few resistors, an LED, two conventional push buttons, and an Arduino Uno make up the hardware configuration. The ladder logic program uploaded using the OpenPLC Editor is executed by the Arduino, which functions as a PLC. The LED is a digital output, while the push buttons are digital input devices. To guarantee optimal operation, precise pin connections are essential. This allows the system to recognize button inputs and adjust the LED appropriately.

CONCLUSION

This project used ladder logic programming with the OpenPLC Editor and Arduino Uno to successfully illustrate the basic workings of a start-stop control circuit. The research demonstrated how a PLC-based system can efficiently control digital outputs (LED) using digital inputs (push buttons) through a latching mechanism by combining hardware, software, and electrical components.

By using common ladder logic components like coils, negated contacts, and normally open contacts, the circuit was able to hold onto its state until a certain input altered it. This is similar to industrial automation systems in the real world, where control that is dependable and consistent is essential. All things considered, the project laid a strong basis for future, more complex control systems by offering hands-on experience with PLC programming, fundamental circuit design, and real-time hardware interfacing.

RECOMMENDATIONS

To further enhance this PLC-based control system, we can explore different inputs and outputs devices so that this will provide a deeper understanding of how PLCs interact with a variety of field devices commonly used in industrial automation. It is recommended to expand the project by integrating additional input and output devices, such as sensors, buzzers, or relays. This would allow for the simulation of more complex automation scenarios, such as emergency stops, delayed starts, or multi-stage processes By following these recommendations, learners can expand their knowledge and practical experience with PLCs and acquire the skills needed to design and implement advanced control systems used in a wide range of industries.

REFERENCES

Rehg, J. A., & Sartori, G. J. (2009). *Programmable Logic Controllers* (4th ed.). Pearson Education.

ACKNOWLEDGEMENTS

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

<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: darwish Read [/]
Name: AHMAD DARWISH BIN AHMAD TERMIZI Understand [/]
Matric Number: 2212089 Agree [/]
Contribution: Experimental Setup, Data Collection, Data Analysis

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

Signature: Read [/]
Name: AKMAL ZARIF BIN NAJIB Understand [/]
Matric Number: 2211971 Agree [/]
Contribution: Discussion, Materials & Equipment

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