



## MECHATRONICS SYSTEM INTEGRATION (MCTA 3203)

SECTION 2, SEMESTER 1, 2024/2025

### ACTIVITY REPORT

#### Week 5:

Understanding both software and hardware aspects of PLC interfacing with  
Microcontrollers

No	Name (Group 6)	Matric No.
1	MUHAMMAD BASIL BIN ABDUL HAKIM	2215315
2	MUHAMMAD SYAFIQ BIN NOR AZMAN	2213187
3	MUHAMMAD HAFIZ HAMZAH BIN FANSURI	2212803
4	MUHAMMAD AMMAR ZUHAIR BIN NOR AZMAN SHAH	2110259
5	MUHAMMAD RAZIQ BIN KAHARUDDIN	2120225

# Table of content

<b>Table of content.....</b>	<b>2</b>
<b>Abstract.....</b>	<b>3</b>
<b>Material and equipment.....</b>	<b>4</b>
<b>Experimental Setup and Methodology.....</b>	<b>6</b>
<b>Results.....</b>	<b>7</b>
<b>Discussions.....</b>	<b>8</b>
<b>Question.....</b>	<b>15</b>
<b>Recommendation.....</b>	<b>17</b>
<b>Conclusion.....</b>	<b>17</b>
<b>Student Declaration.....</b>	<b>18</b>

# Abstract

A Programmable Logic Controller (PLC) is a dedicated computing device designed for automating industrial operations. PLCs observe inputs, decide based on their stored logic, and manage outputs to automate machinery or processes. They are built to be durable, dependable, and able to manage various control and automation requirements across multiple sectors.

PLCs are extensively utilized in multiple sectors especially the industrial sector because of their affordability. Because they can be reused and reprogrammed, they help businesses save considerable expenses in creating and executing control systems. PLCs provide adaptability and programmability; in contrast to hard-wired control systems, they can be reconfigured to respond to alterations in production lines or processes, providing considerable flexibility for companies. Moreover, they are flexible and applicable in both straightforward and intricate applications, varying from an individual machine to a complete production system.

Furthermore, PLCs also minimize downtime by enabling rapid diagnosis of problems and swift adjustments, resulting in increased productivity. They also provide maintenance through built-in diagnostics that make troubleshooting simpler, reducing the need for extensive staff retraining.

Last but not least, PLCs promote standardization. They are founded on well-known programming languages (like Ladder Logic), which makes them approachable for engineers in various sectors and companies.

## Introduction

This project aims to develop a Start-Stop control circuit using a ladder diagram, which will be created in the OpenPLC editor. The ladder diagram is designed to implement a latch system for controlling the on/off operation of an LED using two push buttons. Once the ladder diagram is developed, it will be compiled, simulated, and then transferred to an Arduino Uno microcontroller. The Arduino board, connected to the circuit components, will execute the control logic, allowing the Start-Stop function to control the LED based on the ladder diagram's programming. This setup provides a practical demonstration of using programmable logic in simple automation tasks.

## Material and equipment

1. 1x Arduino UNO board
2. 2x Push button
3. Jumper wires
4. 1x LED
5. Resistors
6. Breadboard
7. OpenPLC Editor software

## Experimental Setup and Methodology

1. The ladder diagram is created in the OpenPLC Editor.
2. The variables used are specified in the ladder diagram.
3. Compile the ladder diagram and simulate it in the OpenPLC Editor.
4. Once no error occurs, the ladder diagram is uploaded to the Arduino board.
5. Ensure the COM port number of the Arduino board connected to the computer is correct before being uploaded.
6. The circuit is built as shown in Figure 1.

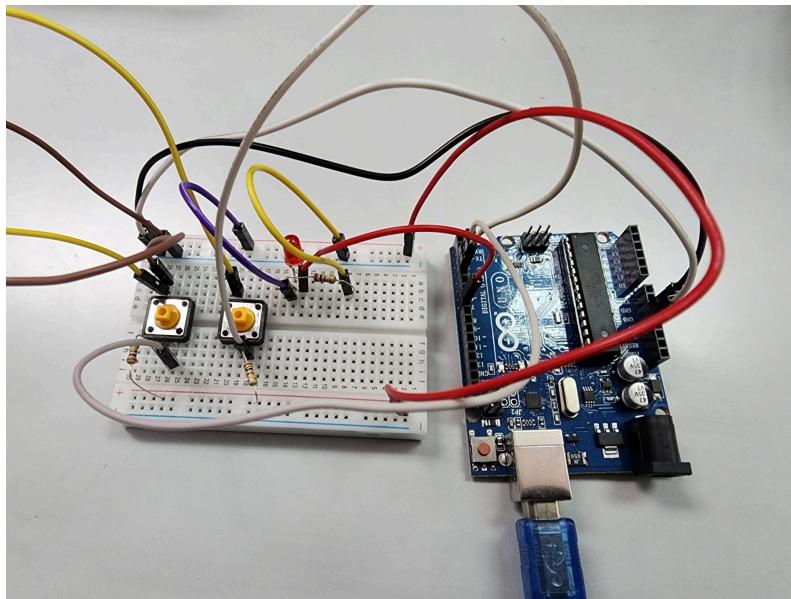


Figure 1: Experiment setup

## Results

The experiment effectively showcased the regulation of an LED through OpenPLC and an Arduino. As soon as the push-button 1 was pressed, the LED lit up instantly and stayed on even after the button was let go. This behaviour validated that the ladder logic utilised a latching mechanism, indicating that the LED stayed in the "on" state until the subsequent action took place. The system was engineered to maintain the LED's illumination after push button 1 was activated, and this was confirmed in practice as the LED remained on without needing the button to be pressed continuously.

When push button 2 was activated, the LED turned off as anticipated, and it stayed off even after the button was let go. This action showcased the unlatching feature of the system, in which pressing push button 2 directly disabled the LED. The system accurately demonstrated the desired functionality: the LED remained illuminated after push button 1 was pressed, and only turned off when push button 2 was activated. The findings validated that the ladder logic created in OpenPLC was effectively executed, with the Arduino correctly managing the LED in accordance with the push buttons' states.

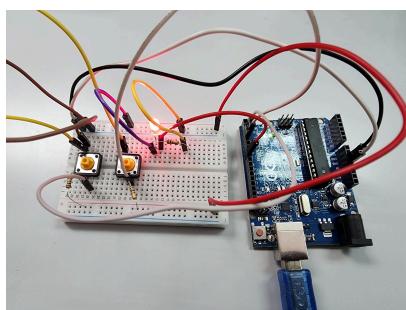


Figure 2.1: LED when push button 1 is pressed (ON).

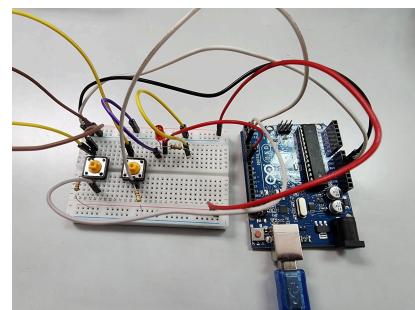


Figure 2.2: LED when push button 2 is pressed (OFF)

[Link for the video: click here](#)

# Discussions

## Hardware Discussion

### **1. Arduino**

Arduino is a type of microcontroller that can be used to control electrical circuits. It acts as the brain of the circuit based on the codes that have been uploaded into it. Usually, Arduino IDE is used as the medium to code for Arduino. But in this experiment, we are exploring other alternatives, such as OpenPLC in this experiment, to learn how to use different programmes to control an Arduino.

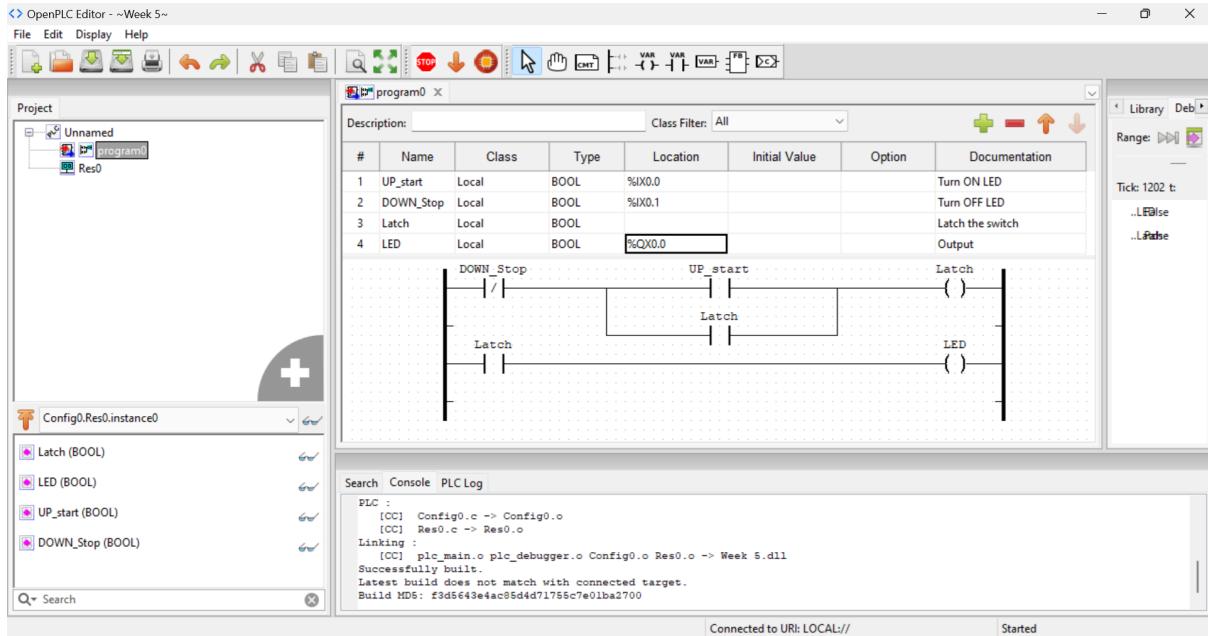
### **2. Light-emitting Diode**

A light-emitting diode(LED) is a semiconductor that emits light when a current flows through it.

### **3. Push-button**

A push-button is a mechanical switch that can open or close a current flow by pressing its button.

## Software Discussion



OpenPLC is a high-language programming language that is used to create programmable logic controllers (PLC). Unlike the conventional method of writing codes in Arduino IDE, we can use OpenPLC to build a programme by simply dragging and dropping to build the circuit that we want before uploading the results into an Arduino.

In this experiment, we built a latching circuit that will turn on an LED when a push-button is pressed and will still be turned on even after letting go of the button. To switch off the LED, another button is served to cut the electrical connection to the LED.

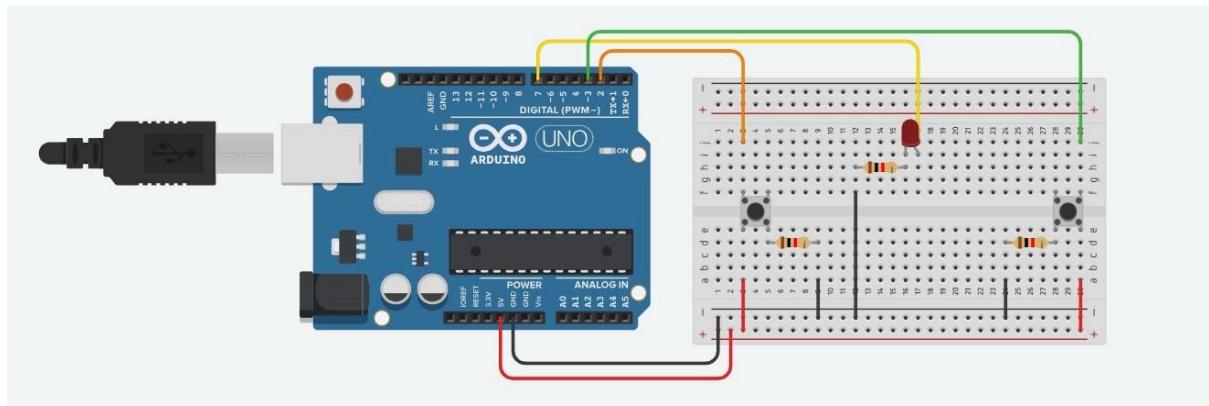
Initially, a current is flowing through a normally closed *DOWN\_stop* switch. When a normally open switch, *UP\_start* button is pressed, it will have a boolean value TRUE, thus current will flow through it to the *latch* coil. When the *latch* coil is TRUE, the remaining components that have the same variable name *latch* will be TRUE.

The *latch* power contact on the second level of the power rail will be TRUE and current will flow through it to an LED output, thus turning it on.

At the same time, the *latch* power contact on the first level will also be TRUE, acting as a latch to allow the current to flow.

To switch off the circuit, we simply need to press the *DOWN\_stop* switch which will become TRUE, opening the switch, and cutting off the power from the circuit.

## Electrical discussion



## 1. Light-emitting Diode

The LED's anode is connected to pin 7 on the Arduino, corresponding to the physical addressing initialise in OpenPLC of digital out: %QX0.0 while the anode is connected to the ground through a resistor.

## 2. Push-button

On one side of the push-button, one leg is connected to the Ground pin on the Arduino through a resistor while the other leg is on 5V. On the other side of the push-button, the connection depends on the function of the button:

- a)** For the ON button, either leg is connected to pin 2, corresponding to the physical addressing initialise in OpenPLC of digital in: %IX0.0.
  - b)** For the OFF button, either leg is connected to pin 3, corresponding to the physical addressing initialise in OpenPLC of digital in: %IX0.1.

## Recommendation

In this experiment, optimising the ladder diagram for efficiency and clarity is essential. It is recommended to eliminate redundant contacts and simplify the logic wherever possible to enhance readability and minimise errors during implementation. Additionally, careful planning of the I/O assignments on the Arduino is crucial to ensure the system works as expected. Using external relays or transistors is advised when controlling devices that require more current or voltage than the Arduino pins can supply, thus protecting the microcontroller from potential damage.

Testing the ladder diagram in a simulation environment before uploading it to Arduino is highly recommended. This approach helps identify and resolve logic errors without risking hardware failure. It is also beneficial to keep the program structure modular and well-documented, which will make troubleshooting easier and improve the maintainability of the system. Using consistent naming conventions and commenting on the code can help future modifications or additions to the project.

Finally, increasing system reliability through the inclusion of fail-safes, error-handling routines, and feedback systems should be a priority. These measures can help detect issues early, preventing system downtime and ensuring smoother operation in real-world applications. Proper documentation of both the ladder diagram and the Arduino code is crucial for future users to understand, modify, or troubleshoot the system. Looking ahead, exploring more advanced PLC platforms or expanding the system with additional sensors and

actuators may provide further opportunities for improvement and growth in more complex control environments.

## Conclusion

To conclude, this project demonstrated the integration of software and hardware elements to manage a latch system. Using the OpenPLC Editor, ladder logic programming was utilized to create the system logic, which was subsequently implemented on an Arduino microcontroller. This integration enabled effective management of the latch system via accurate programming (ladder diagram design) and circuit design.

During the entire process, from setup and configuration to testing and troubleshooting, it was crucial to guarantee the system's reliability and compatibility. By following a structured methodology, problems that have been found during trial and error were pinpointed and addressed, until it finally worked, allowing the system to operate as intended. The indicator of our (project) system's success must be followed by the process of monitoring the LED function and checking circuit parameters to confirm the proper operation of the logic.

In summary, this project emphasizes the practicality and efficiency of integrating Arduino with PLC logic in embedded system applications. The findings highlight that, through suitable programming and circuit design, even complex control systems can be reliably developed, providing an understanding of the wider potential of combining PLCs with microcontrollers like an Arduino for more diverse automation and control functions' applications.

## Student Declaration

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 taken or done by unspecified sources or person. 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 have been verified by us.

NAME: Muhammad Basil bin Abdul Hakim	READ	✓
MATRIC NO: 2215315	UNDERSTAND	✓
SIGNATURE: 	AGREE	✓

NAME: Muhammad Syafiq Bin Nor Azman	READ	✓
MATRIC NO: 2213187	UNDERSTAND	✓
SIGNATURE 	AGREE	✓

NAME:Muhammad Hafiz Hamzah Bin Fansuri	READ	<input checked="" type="checkbox"/>
MATRIC NO:2212803	UNDERSTAND	<input checked="" type="checkbox"/>
SIGNATURE : 	AGREE	<input checked="" type="checkbox"/>

NAME: Muhammad Ammar Zuhair Bon Nor Azman Shah	READ	<input checked="" type="checkbox"/>
MATRIC NO: 2110259	UNDERSTAND	<input checked="" type="checkbox"/>
SIGNATURE: 	AGREE	<input checked="" type="checkbox"/>

NAME: MUHAMMAD RAZIQ BIN KAHARUDDIN	READ	<input checked="" type="checkbox"/>
MATRIC NO: 2120225	UNDERSTAND	<input checked="" type="checkbox"/>
SIGNATURE 	AGREE	<input checked="" type="checkbox"/>