MECHATRONICS SYSTEM INTEGRATION MCTA3203

PLC INTERFACING WITH

MICROCONTROLLER

EXPERIMENT 7

DATE: 17TH November 2025
SECTION: 2
GROUP: 18
SEMESTER 1, 2025/2026

NO	NAME	MATRIC NO
1.	SHAMSUL HAKIMEE BIN SHAMSUL HAIREE	2315027
2.	QASHRINA AISYA BINTI MOHD NAZARUDIN	2315184
3.	NUREL HUMAIRAH BINTI MUHAMMAD FIRDAUS	2315680

DATE OF SUBMISSION :
Monday, 24TH November 2025

Abstract

This lab report details the process of interfacing a Programmable Logic Controller (PLC) program with an Arduino microcontroller using the OpenPLC Editor and an Ethernet/IP connection. The OpenPLC Editor, an IEC 61131-3 compliant, free, and open-source PC editor, was used to create and simulate Ladder Diagrams (LD) before uploading the code directly to the Arduino board running the OpenPLC Runtime.

Table of Contents

1.0 Introduction

Programmable Logic Controllers (PLCs) are essential devices in modern industrial automation, having evolved from their original role of replacing complicated relay circuits in manufacturing. Today, they are the core technology for managing and automating various electromechanical processes across diverse applications, such as controlling machinery in production facilities.

This laboratory session focuses on connecting a PLC-based control system, which is programmed using the OpenPLC Editor, with an Arduino microcontroller. The OpenPLC Editor is a free, open-source PC editor that is compliant with the IEC 61131-3 standard , allowing users to create PLC code in languages like Ladder Diagram (LD). This code can then be uploaded directly to any board running the OpenPLC Runtime, such as an Arduino.

The practical segment of this experiment includes two primary objectives:

1. Basic Control: Developing, simulating, and uploading a fundamental ladder diagram to the Arduino to achieve a simple task: making an LED blink.
2. Industrial Simulation: Creating and testing a more complex Start-Stop Control Circuit using an LED and two push-button switches to simulate a core function of industrial control systems.

By executing these tasks, students will gain practical experience in translating theoretical control logic into a functioning physical system using electronic components, effectively bridging the gap between abstract control system design and hardware implementation

2.0 Materials and Equipment

- Software:

  ● OpenPLC Editor: For designing ladder logic, simulating it, and compiling code for Arduino microcontroller.
  ● Arduino IDE (optional): Used for board communication troubleshooting and verification.
  ● Device Manager (Windows): To confirm the active COM port used by Arduino.

- Hardware:

  ● Arduino Mega 2560
  ● 2 Push Button Switches
  ● LED
  ● Breadboard
  ● Resistors (220Ω – 10kΩ)
  ● Jumper wires
  ● USB Cable

3.0 Experimental Setup

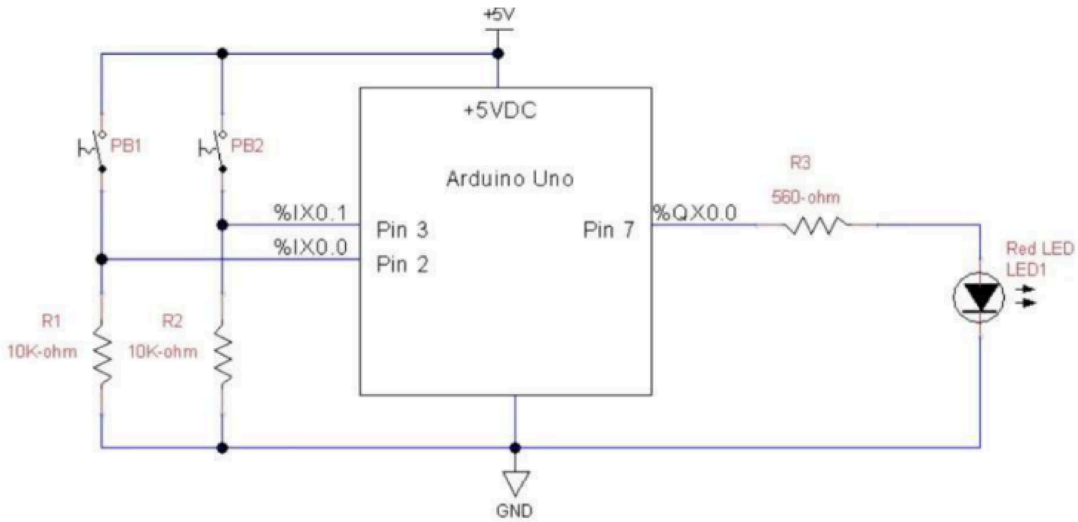

Fig 1. Start-Stop Control Circuit

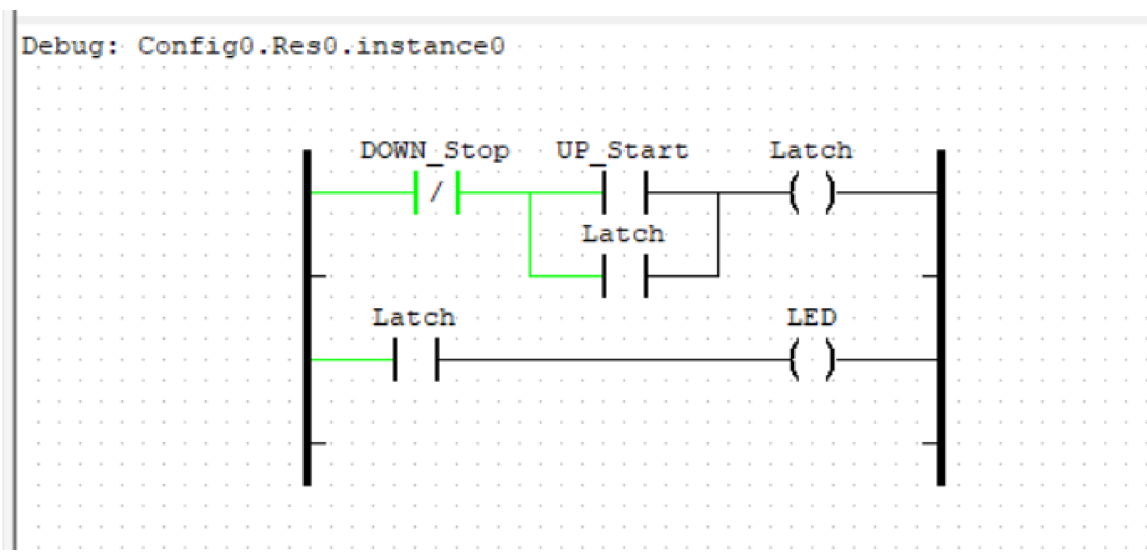

Fig 2. Ladder Diagram for the Start-Stop Control Circuit

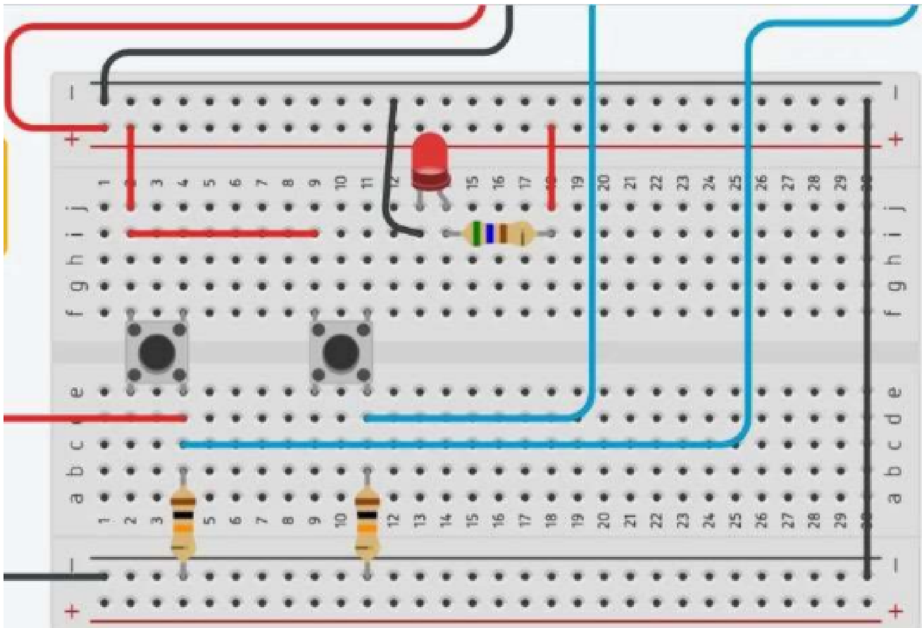

Fig 3. Circuit Diagram

3.1 Circuit Setup

1. The Start-Stop Control Circuit ladder diagram (Figure 1) was first developed in the software.
2. All necessary variables for the diagram were defined and configured.
3. The completed ladder logic program was then compiled and tested through simulation 4 within the OpenPLC Editor.
4. Following successful simulation, the program was transferred to the Arduino board.
5. Crucially, the correct COM port and the mappings (pin associations) between the OpenPLC variables and the physical Arduino pins were confirmed before transfer.
6. The physical circuit corresponding to the design (Figure 3) was built.
7. Finally, the operational functionality of the built circuit and uploaded program was verified

4.0 Methodology

4.1 LED Blinking Using Ladder Logic

The experiment followed these steps to achieve the blinking LED functionality:

1. Software Setup: The OpenPLC Editor software was downloaded and installed.
2. Project Initiation: A new project was created (File $\rightarrow$ New) and named, with the language set to LD (Ladder Diagram).
3. Variable Definition: A local BOOL variable named led was created. Its location was set to the physical output pin of the microcontroller, such as %QX0.0 for digital pin 14 on an Arduino Mega board.
4. Logic Diagram Creation: The basic ladder diagram was constructed by adding left and right power rails, a negated contact (representing the LED status), and a coil.
5. Simulation: The logic was compiled by clicking the build icon, and the blinking LED operation was observed in the editor's simulation.
6. Deployment: The compiled program was uploaded to the Arduino board. This required connecting the board, noting the COM port number, selecting the correct board type and COM port in the transfer window, and pressing "Transfer to PLC." The physical LED then began to blink.
7. Time Delay Modification (Exercise): To control the blinking time, the ladder diagram was modified to include a timer block (e.g., TON/TOF). The time delay was set using a variable expression like T#500ms. This updated logic was then simulated and transferred to the Arduino.

4.2 Control Algorithm

The control algorithm uses two push buttons to manage an output (in this case, an LED):

- PB1 (Start): Pressing this button activates the output, turning the LED ON.
- PB2 (Stop): Pressing this button deactivates the output, turning the LED OFF.
- Latching Functionality: The LED maintains its current state (ON or OFF) until it receives an explicit signal from either the Start or Stop button. This holding behavior is achieved using a latch instruction.

5.0 Result

The results demonstrated the successful implementation of both simple and complex Ladder Diagram (LD) logic onto an Arduino microcontroller using the OpenPLC Editor. For the initial task, a basic LD program was created, simulated, and uploaded, causing a physical LED connected to the Arduino output pin (e.g., %QX0.0) to blink continuously. This was extended by integrating TON and TOF timer function blocks to accurately control the blinking interval (e.g., T#500ms), validating the use of IEC 61131-3 standard function blocks. For the more advanced Start-Stop Control Circuit, a ladder diagram featuring a latch instruction was successfully deployed, where pressing the Start button (%IX0.0) latched the output LED ON, and pressing the Stop button (%IX0.1) safely unlatched it OFF, accurately mimicking fundamental industrial control behavior. These outcomes confirmed that the OpenPLC platform effectively bridges standard industrial control programming with low-cost microcontroller hardware.

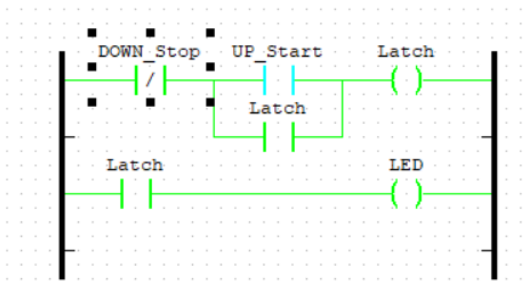

Figure 5.1 showing the simulation when the UP_Start button is in TRUE state during simulation.

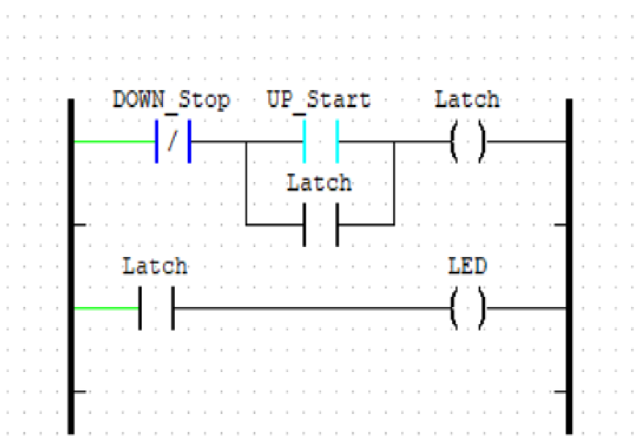

Figure 5.2 Shows DOWN_Stop button in TRUE during simulation.

6.0 Discussion

The timer block integration exercise successfully extended the fundamental LED blinking program, demonstrating the practical application of IEC 61131-3 standard function blocks in the OpenPLC environment. By replacing the basic logic with a combination of TON (Timer ON-delay) and TOF (Timer OFF-delay) blocks, the blinking rate of the LED was precisely controlled and modified. The assignment of a time variable expression, such as T#500ms, to the timer's PT (Preset Time) input, provided a direct and quantifiable method for adjusting the ON and OFF intervals, which is essential for synchronization and sequencing in industrial automation. This exercise confirmed that the OpenPLC Editor can handle and execute standard PLC timing instructions, validating its use as a viable platform for developing and testing time-based control algorithms before deploying them to the Arduino board. The ability to simulate and transfer these complex blocks underscores the power of using a standardized PLC language (Ladder Diagram) on a versatile, low-cost microcontroller.

7.0 Conclusion

This laboratory session successfully demonstrated the fundamental principles and practical steps required for implementing industrial control logic on a microcontroller platform using the OpenPLC Editor and an Arduino board. By utilizing the IEC 61131-3 compliant Ladder Diagram (LD) language, two key applications were developed and deployed. First, the basic blinking LED circuit provided an accessible introduction to the compile, simulate, and transfer workflow, with the subsequent exercise validating the crucial functionality of TON and TOF timer blocks for precise time-based control. Second, the development and testing of the Start-Stop Control Circuit using two push buttons and a latch instruction successfully replicated a core electromechanical control pattern found in automation. The ability to map standard PLC variables (inputs `%IX0.0`, `%IX0.1,` and output `%QX0.0`) to the physical pins of the Arduino confirms the OpenPLC platform's effectiveness in bridging standardized industrial programming practices with low-cost, open-source hardware, ultimately achieving PLC-like control functionality over an Ethernet/IP concept.

8.0 Recommendation

The results demonstrated the successful implementation of both simple and complex Ladder Diagram (LD) logic onto an Arduino microcontroller using the OpenPLC Editor. For the initial task, a basic LD program was created, simulated, and uploaded, causing a physical LED connected to the Arduino output pin (e.g., %QX0.0) to blink continuously, which was further extended by integrating TON and TOF timer function blocks with a preset time of, for example, T#500ms to accurately control the blinking interval. For the more advanced Start-Stop Control Circuit, a ladder diagram featuring a latch instruction was successfully deployed, where pressing the Start button (UP_Start, mapped to %IX0.0) latched the output LED ON, and pressing the Stop button (DOWN_Stop, mapped to %IX0.1) safely unlatched it OFF, accurately mimicking fundamental industrial control behavior. These outcomes confirmed that the OpenPLC platform effectively bridges standardized industrial control programming (IEC 61131-3) with low-cost microcontroller hardware.

9.0 References

1. *Autonomy – open-source PLC software*. (n.d.). https://autonomylogic.com/

2. Wilcher, D. (2022, August 30). *PLC Ladder Logic on an Arduino: Introduction to OpenPLC.* Control.com. https://control.com/technical-articles/plc-ladder-logic-on-an-arduino-introduction-to-openplc/

3. Wilcher, D. (2022, September 28). *PLC Ladder Logic on an Arduino: Building a Start-Stop Circuit.* Control.com. https://control.com/technical-articles/plc-ladder-logic-on-an-arduino-building-a-start-stopcircuit/ (control.com)

4. Mechatronics System Integration Lab Module - Google Drive https://drive.google.com/drive/folders/1QecGGN96D76UL2gwBjevFwiEXv2UEniO

10.0 Acknowledgement

We would like to express my sincere gratitude to Dr. Wahju Sediono, Dr. Zulkifli Bin Zainal Abidin, our teaching assistant, and our peers for their invaluable guidance and support throughout the completion of this report. Their insights, feedback, and expertise greatly contributed to the depth and quality of this work. We truly appreciate their time, patience, and commitment to our academic growth.

11.0 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.

Name: Shamsul Hakimee Bin Shamsul Hairee	Read	[/]
Matric Number: 2315027	Understand	[/]
Signature:	Agree	[/]

Name: Qashrina Aisya binti Mohd Nazarudin	Read	[/]
Matric Number: 2315184	Understand	[/]
Signature:	Agree	[/]

Name: Nurel Humairah Binti Muhammad Firdaus	Read	[/]
Matric Number: 2315680	Understand	[/]
Signature:	Agree	[/]