REPORT 4: PLC INTERFACING

GROUP 2

MCTA 3203

SEMESTER 1 2024/2025

MECHATRONICS SYSTEM INTEGRATION

DATE OF SUBMISSION: 13/11/2024

	NAME	MATRIC NUMBER
1.	AKMAL ZARIF BIN NAJIB	2211971
2.	AMIR FARHAN BIN GHAFFAR	2115617
3.	AMIRAH HUDA BINTI JAMALULAIL ASRI	2210776
4.	AMIRUL AIZAD BIN AMIR	2211263
5.	ANAS BIN BADRULZAMAN	2219945

TABLE OF CONTENTS

ABSTRACT

This experiment explores the integration of a Programmable Logic Controller (PLC) with a microcontroller to control a Start-Stop circuit, using the OpenPLC Editor Software, ladder diagram are created, simulated and transferred to the arduino microcontroller, the logical interface of the software with the physical hardware component. This experiment demonstrates a simple but effective control mechanism commonly used in industrial automation. The objective is to bridge the gap between software based PLC control and hardware implementation, highlighting key aspects of PLC programming and microcontroller interfacing.

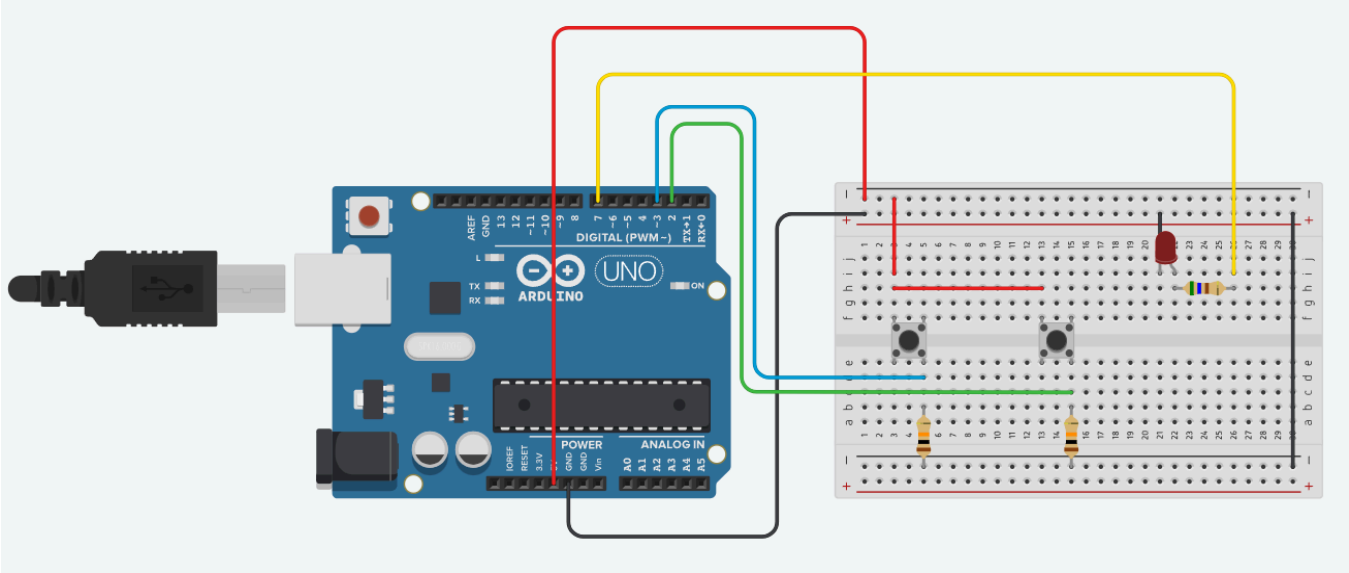
INTRODUCTION

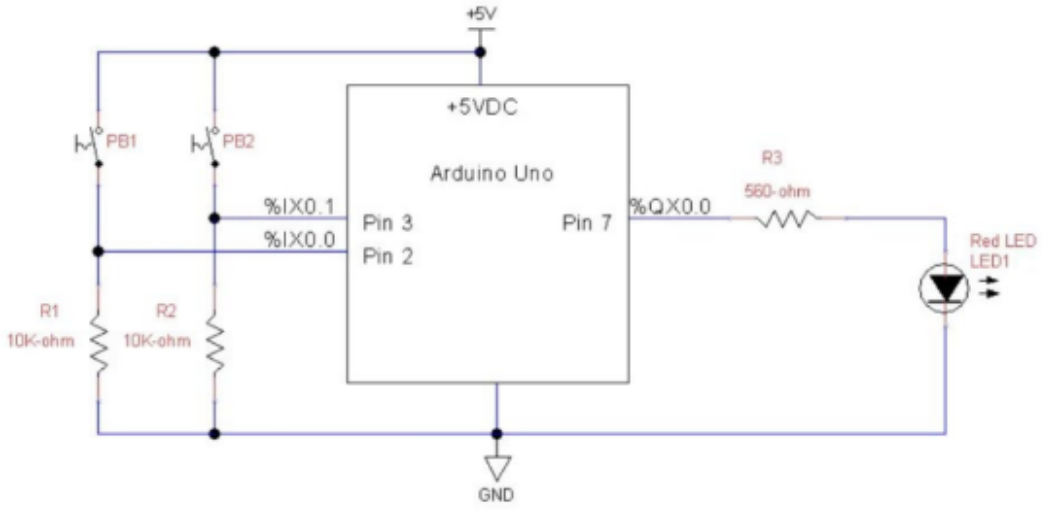
In Industrial automation, PLC plays an important role in controlling electromechanical processes across a wide range of applications. By using a logic-based control system, PLCs ensure accurate and reliable operation of machines and processes. However, understanding both software programming and hardware interface aspects is essential to creating an efficient control circuit. This experiment demonstrates the use of OpenPLC Editor which is a widely used PLCprogramming tool, to design and simulate a Start-Stop control circuit, which is then transferred to an Arduino microcontroller for real-world application. The Arduino microcontroller serves as a bridge between the digital logic in the ladder diagram and the physical components on the breadboard, providing insight into integration of PLCs with microcontrollers for practical applications in automation. Through these hands-on experiments, the basic principles of PLCprograming, simulation and hardware interfaces are illustrated, emphasizing the importance of combining software and hardware elements in a control system.

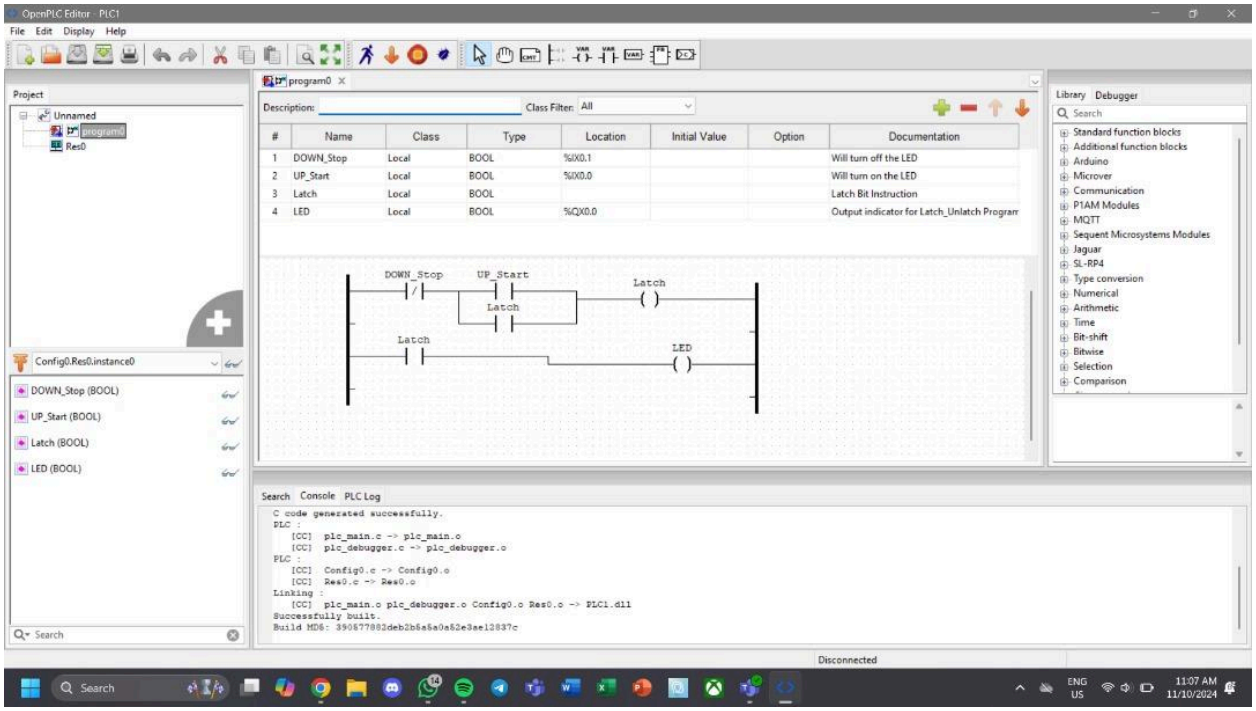
MATERIALS AND EQUIPMENT

- OpenPLC Editor Software
- Arduino board
- 2 push buttons
- Jumper wires
- LED
- Resistors
- Breadboard

EXPERIMENTAL SETUP




+5V
+5VDC
Arduino Uno
PB1
PB2
%IX0.1
%IX0.0
Pin 3
Pin 2
Pin 7
%QX0.0
R3
560-ohm
Red LED
LED1
R1
10K-ohm
R2
10K-ohm
GND


OpenPLC Editor - PLC1
File Edit Display Help
Project
Unnamed
program0
Res0
program0
Description:
Class Filter: All
Name Class Type Location Initial Value Option Documentation
1 DOWN_Stop Local BOOL %IX0.1 Will turn off the LED
2 UP_Start Local BOOL %IX0.0 Will turn on the LED
3 Latch Local BOOL Latch Bit Instruction
4 LED Local BOOL %QX0.0 Output indicator for Latch_Unlatch Program
DOWN_Stop
UP_Start
Latch
Latch
Latch
LED
Library Debugger
Standard function blocks
Additional function blocks
Arduino
Microver
Communication
P1AM Modules
MQTT
Sequent Microsystems Modules
Jaguar
SL-RP4
Type conversion
Numerical
Arithmetic
Time
Bit-shift
Bitwise
Selection
Comparison
Config0.Res0.instance0
DOWN_Stop (BOOL)
UP_Start (BOOL)
Latch (BOOL)
LED (BOOL)
Search Console PLC Log
C code generated successfully.
PLC :
[CC] plc_main.c -> plc_main.o
[CC] plc_debugger.c -> plc_debugger.o
PLC :
[CC] Config0.c -> Config0.o
[CC] Res0.c -> Res0.o
Linking :
[CC] plc_main.o plc_debugger.o Config0.o Res0.o -> PLC1.dll
Successfully built.
Build MD5: 390877883deb2b6a5a0a82e3ee12837c
Disconnected
Search
ENG US
11:07 AM 11/10/2024

METHODOLOGY

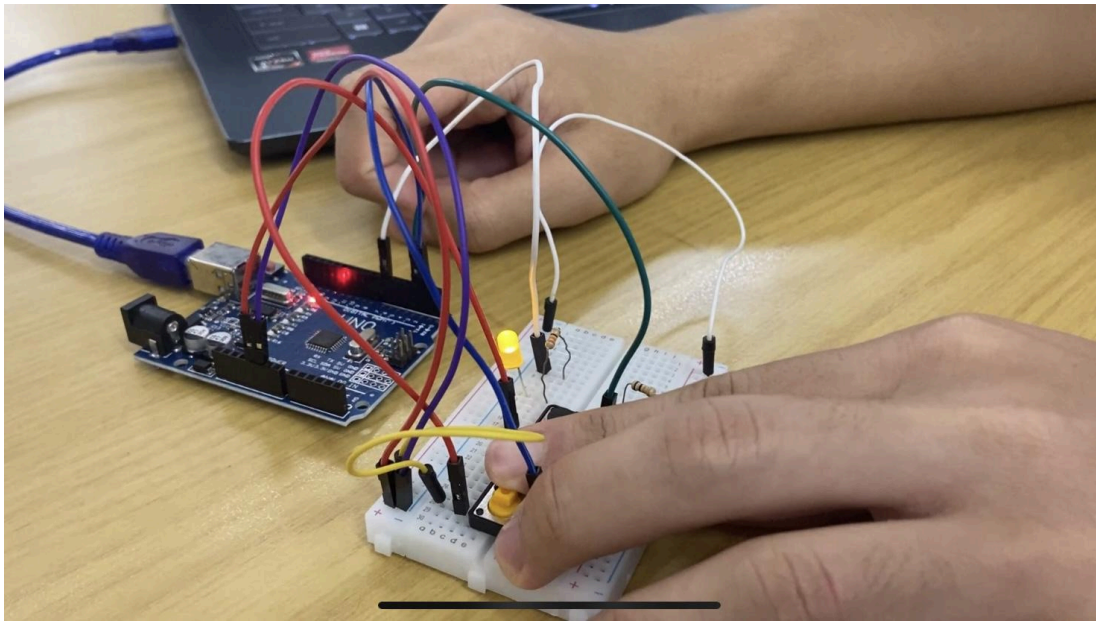
The methodology of this experiment focuses on interfacing OpenPLC Editor with an Arduino Uno board. The OpenPLC Editor software is used to create, compile, simulate and transfer the ladder diagram program to Arduino Uno board. The ladder diagram represents the Start-Stop Control Circuit. The steps to create the Start-Stop Control Circuit involves drawing the ladder diagram in the software, wiring and testing the hardware components.

PROCEDURE

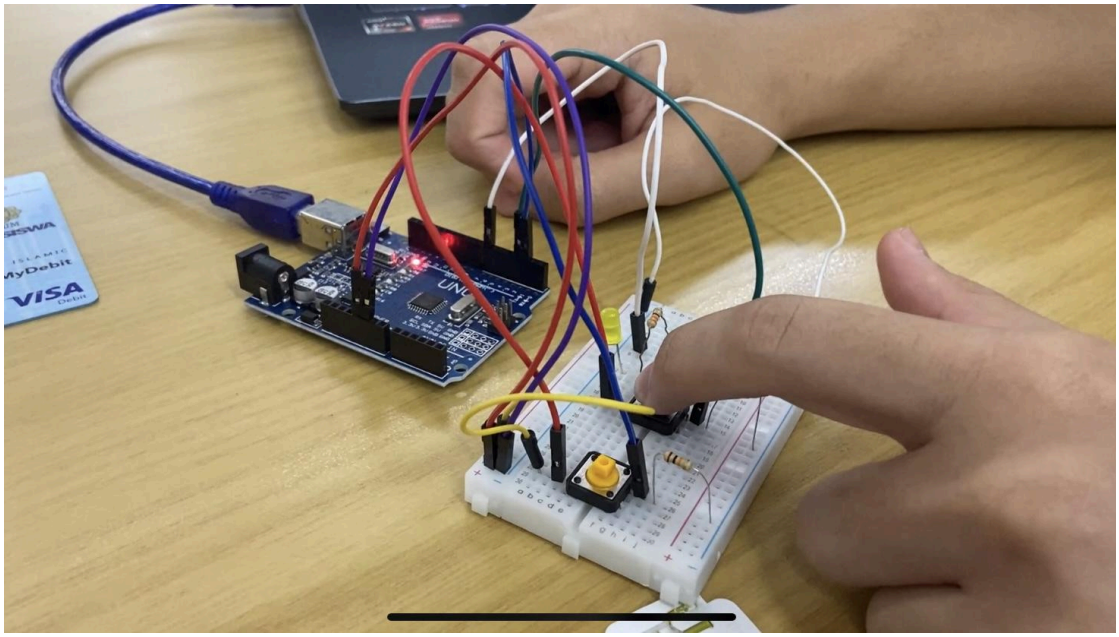
1. Start a new project in the OpenPLC Editor by creating a new file.
2. Name the file accordingly and select the language of LD (ladder diagram).
3. Create the variables and specify them. Refer the associated pin between OpenPLC variables and Arduino Uno from the OpenPLC Editor website.
4. Start creating the ladder diagram by right-clicking and choosing Add.
5. Compile the ladder diagram.
6. Choose the correct board and COM port number.
7. Press the Transfer to PLC button.
8. Test the functionality and do troubleshooting.

RESULTS

The output of the LED is turned on as we push PB1, which relates to the “UP_Start” variable in the OpenPLC Editor.



The output of the LED is turned off as we push PB2, which relates to the “DOWN_Stop” variable in the OpenPLC Editor.



DISCUSSION

This experiment demonstrates the basic principles of a Programmable Logic Controller (PLC) and how to interface it to an Arduino Uno 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**

  OpenPLC provides a user-friendly interface for ladder logic programming which is a widely used language in PLC applications. Firstly, when adding the components to a new sketch, we start with power rails which in this case have 4 pins for both sides. Next, we add the functioning variables: turn off led, turn on led, latch bit instruction, and output indicator. For “Down_Stop”, its component in the ladder diagram is a negated contact component. As for “UP_Start” and “Latch”, they are both normal contact components. Lastly, we have the “Latch” and “LED” which are normal coil components. These variables are arranged according to the start-stop control circuit. Once compiled, we choose the right board and com port before uploading it to the Arduino.

- **Electrical**

  The circuit design follows basic electrical principles to ensure proper current flow. A 10 kΩ resistor is connected to both push buttons to stabilize its input signal. The LED is protected by a 560 Ω current-limiting resistor, preventing excessive current from damaging it when it lights up. 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**

  The hardware setup includes an Arduino Uno, two standard push buttons, an LED, and a few resistors. The Arduino functions as a PLC, running the ladder logic program uploaded through OpenPLC Editor. The push button serves as the digital input device while the LED acts as the digital output device. Proper pin connections are essential for correct operations, accurately interpreting button presses and managing LED output.

CONCLUSION

Through this hands-on experience, participants gained insight into the relationship between hardware, software, and electrical principles that are the basis of PLC-based control systems. The experiment serves as an excellent gateway to industrial automation and provides the foundation for more sophisticated control applications and system design.

In summary, the experiment strengthened the theoretical knowledge of PLC programming and also equipped participants with practical abilities in building and diagnosing basic control systems, leading to a firm grasp of the functions of programmable logic controllers in practical automation contexts.

RECOMMENDATIONS

Based on the observation of the experiment, there are many recommendations that can enhance our understanding and practical skills in PLC programming such as handling a more complex circuit that can help in gaining experience with different types of PLC instructions and handling more complex automation tasks. Furthermore, we also 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.

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.

ACKNOWLEDGEMENT

We would like to express our gratitude to Dr. Wahju Sediono, Dr. Ali Sophian, Dr. Zulkifli bin Zainal Abidin, my teaching assistant, and my peers for their invaluable help and support in finishing this project. Their advice, feedback, and experience have greatly influenced the level of quality and understanding of this work.

STUDENT'S DECLARATION

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 has 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 that no further improvement on the reports is needed from any of the individual 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: AKMAL ZARIF BIN NAJIB	**Understand**	/
Matric No: 2211971	**Agree**	/

Signature: 	**Read**	/
Name: AMIR FARHAN BIN GHAFFAR	**Understand**	/
Matric No: 2115617	**Agree**	/

Signature: 	**Read**	/
Name: AMIRAH HUDA BINTI JAMALULAIL ASRI	**Understand**	/
Matric No: 2210776	**Agree**	/

Signature: 	**Read**	/
Name: AMIRUL AIZAD BIN AMIR	**Understand**	/
Matric No: 2211263	**Agree**	/

Signature: 	**Read**	/
Name: ANAS BIN BADRULZAMAN	**Understand**	/
Matric No: 2219945	**Agree**	/