MECHATRONICS SYSTEM INTEGRATION

MCTA 3203

LAB 5:

PLC INTERFACING

SECTION 1

SEMESTER 2, 2024/2025

INSTRUCTOR:

ASSOC. PROF. EUR. ING. IR. TS. GS. INV. DR. ZULKIFLI BIN ZAINAL ABIDIN

DR. WAHJU SEDIONO

GROUP 9

NAME	MATRIC NO
AMER WAFDI BIN ZAINIZAM	2228703
AMALIN BALQIS BINTI RUYUSNI	2314684
FARHA QAISARA BINTI MOHD ZAHIDI	2315994

DATE OF SUBMISSION: 14 APRIL 2025

ABSTRACT

This experiment explores the design and implementation of a basic Start-Stop Control Circuit using ladder logic created in the OpenPLC Editor and executed on an Arduino Mega 2560 board. The objective was to simulate a fundamental industrial automation function, starting and stopping a process using momentary push-button switches, while demonstrating the practical application of PLC programming concepts. The circuit comprises two push buttons (Start and Stop), an LED indicator, resistors, and an Arduino, connected on a breadboard. The ladder diagram was developed using OpenPLC and included a latching mechanism that kept the LED on after the Start button was pressed, and turned it off when the Stop button was activated. The program was compiled, simulated, and successfully transferred to the Arduino, which interpreted input signals and controlled the LED output. Results confirmed the correct operation of the logic and hardware integration, with consistent performance and no unexpected behavior. The experiment validated the use of ladder logic for simple control systems and demonstrated its potential for real-world applications, especially in educational and prototyping contexts. This hands-on exercise provided foundational experience in digital control, automation, and embedded systems, paving the way for more complex PLC-based automation solutions.

1.0 INTRODUCTION

The purpose of this experiment is to design and implement a Start-Stop Control Circuit using a ladder diagram developed in OpenPLC software environment, which is then compiled, simulated, and transferred to an Arduino board for execution. This circuit represents a fundamental concept used in industrial automation systems, where processes are started and stopped using simple push-button inputs. By creating this system, the experiment aims to demonstrate the basic principles of PLC programming and digital control using Arduino hardware.

Ladder logic, commonly used in Programmable Logic Controllers (PLCs), is a graphical programming language that mimics traditional relay logic used in electrical control systems. It uses symbolic representations of electrical components such as contacts, coils, and timers to define control behavior in software. The Start-Stop control system is one of the most basic and essential forms of logic control, where a "Start" button energizes a coil to keep an output such as LED on, and a "Stop" button interrupts the power to de-energize the coil, stopping the output.

In this experiment, OpenPLC Editor is used to construct the ladder diagram, while the Arduino board acts as the controller executing the logic. The control circuit involves two push-button switches, one for starting and one for stopping, a latch to maintain the output state, and an LED representing the actuator. The circuit operates on a simple principle: when the Start button is pressed, the LED turns on and remains on, thanks to the latching logic. Pressing the Stop button breaks the latch and turns off the LED. This type of control logic is frequently used in real-world industrial applications to control motors, pumps, and other machinery.

The expected outcome is that pressing the Start button will turn on the LED and keep it on using a latching mechanism, while pressing the Stop button will turn the LED off. This reinforces the understanding of control logic and provides foundational knowledge for more advanced automation systems.

2.0 MATERIALS AND EQUIPMENT

The materials and equipment that we used in this experiment are:

a) OpenPLC Editor software
b) Arduino Mega 2560 board
c) Breadboard
d) Jumper wires
e) 2 Push buttons
f) LED
g) 560-ohm resistors
h) 10k-ohm resistors

3.0 EXPERIMENTAL SETUP

Here are the steps of equipment and components that were set up for the experiment:

1. Connect the pushbuttons to the Arduino by connecting one leg of each pushbutton to a separate digital pins D2 and D3, and the other leg of each pushbutton to the 5V pin.
2. Use 10K-ohm resistors for each pushbutton by connecting one end of each resistor to the digital pin and the other end to the GND pin of the Arduino.
3. The LED was connected to the Arduino by attaching the longer leg (anode, +) of the LED to one end of the resistor, and the shorter leg (cathode, -) of LED to the GND pin.
4. A 560-ohm resistor was used for the LED by connecting one end of the resistor to digital pin D7, and the other end to the longer leg (anode, +) of LED.
5. The Arduino board was connected to PC via USB.

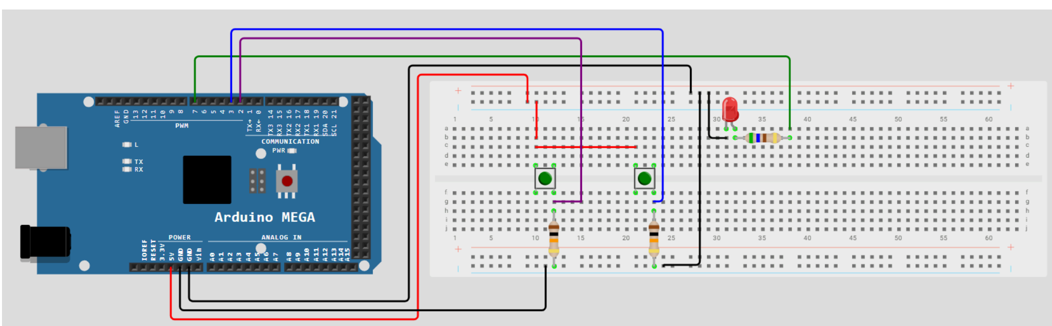


Figure 1: Schematic Diagram

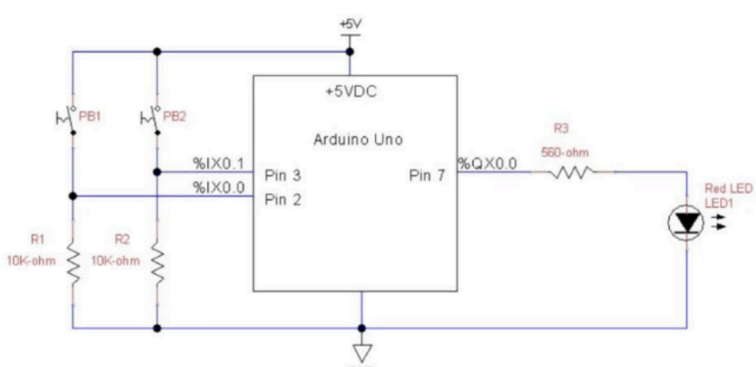


Figure 2: Start-Stop Control Circuit

4.0 METHODOLOGY

Here are the steps followed during the experiment:

1) The circuit was built as shown in **Figure 1**, and the Arduino Mega 2560 was set up.
2) The ladder diagram shown in **Figure 3** was created.
3) All variables used in the ladder diagram were specified.
4) The ladder diagram was compiled and simulated in OpenPLC Editor.
5) The ladder diagram was uploaded to the Arduino board.
6) The correct COM port number and all pin association was selected between the OpenPLC variables and Arduino board.
7) The functionality was tested. The circuit tested on the breadboard must be the same as the simulated circuit in the PLC.
8) The circuit was observed by pressing and releasing the UP_Start and DOWN_Stop pushbuttons.

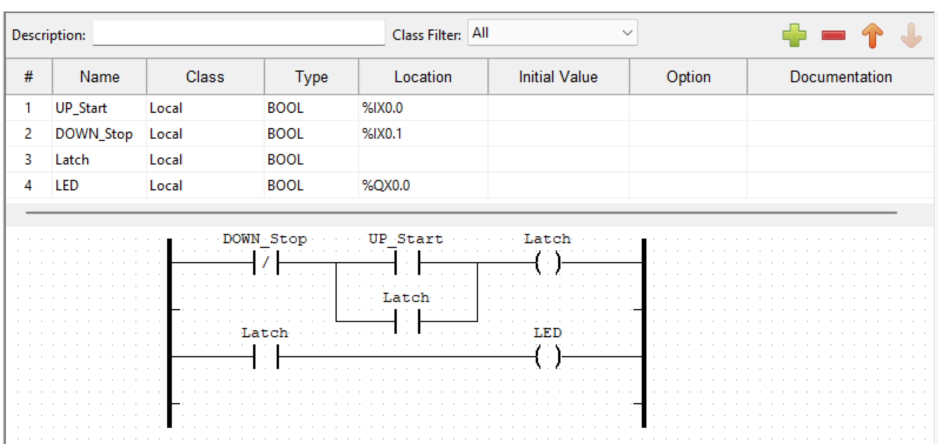
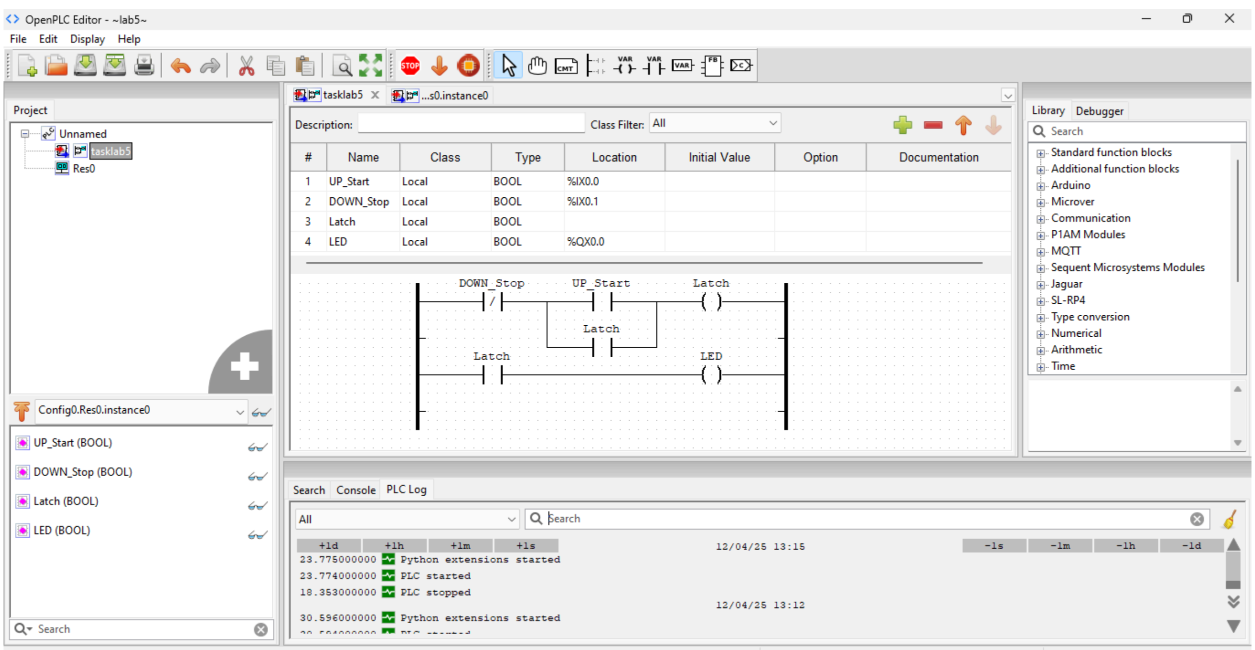


Figure 3: Ladder Diagram

Coding:

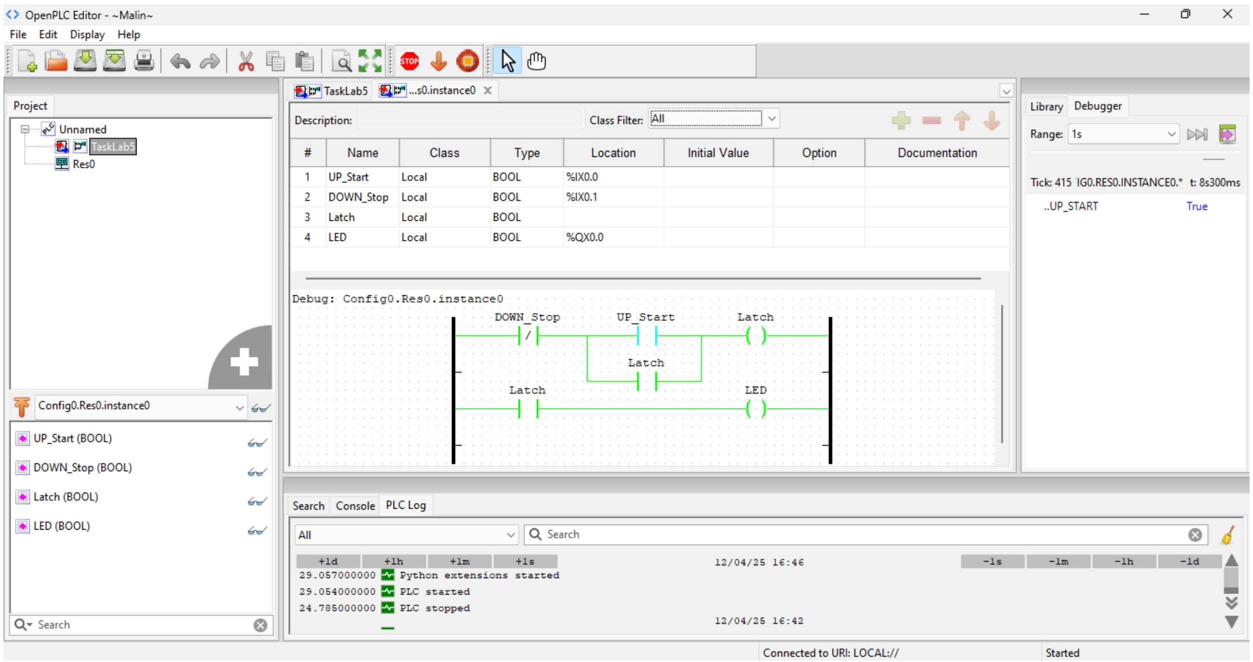
```
PROGRAM tasklab5
  VAR
    UP_Start AT %IX0.0 : BOOL;      // Push button input to start (Set)
    DOWN_Stop AT %IX0.1 : BOOL;     // Push button input to stop (Reset)
  END_VAR
  VAR
    Latch : BOOL;                   // Internal memory for latching
  END_VAR
  VAR
    LED AT %QX0.0 : BOOL;           // Output to LED
  END_VAR

  Latch := (Latch OR UP_Start) AND NOT(DOWN_Stop); // Set/Reset logic
  LED := Latch;                     // LED follows latch
END_PROGRAM
```

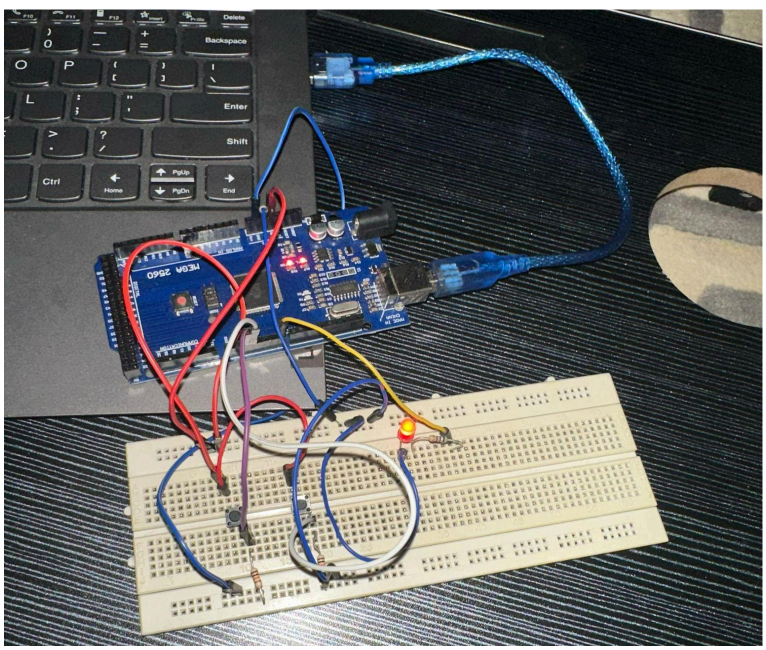


5.0 DATA COLLECTION

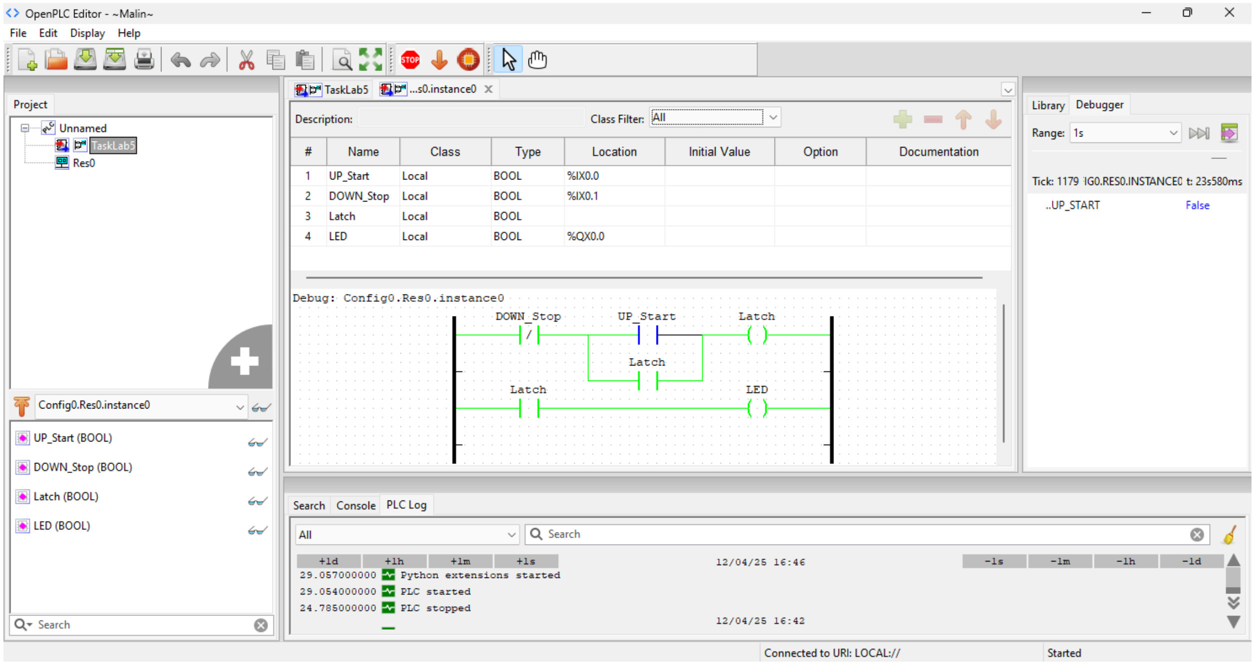
During the experiment, data collected includes:

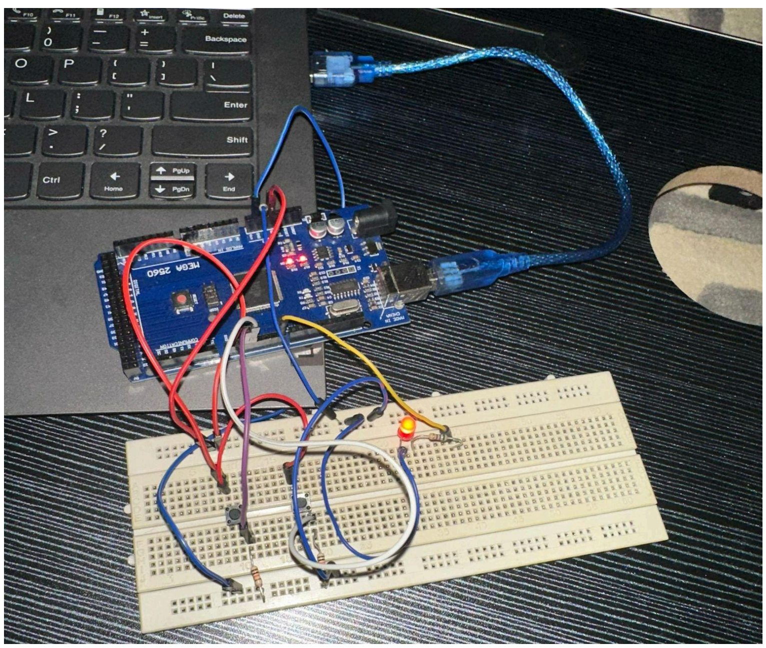
1) When the UP_Start button is pressed, the LED will turn on.



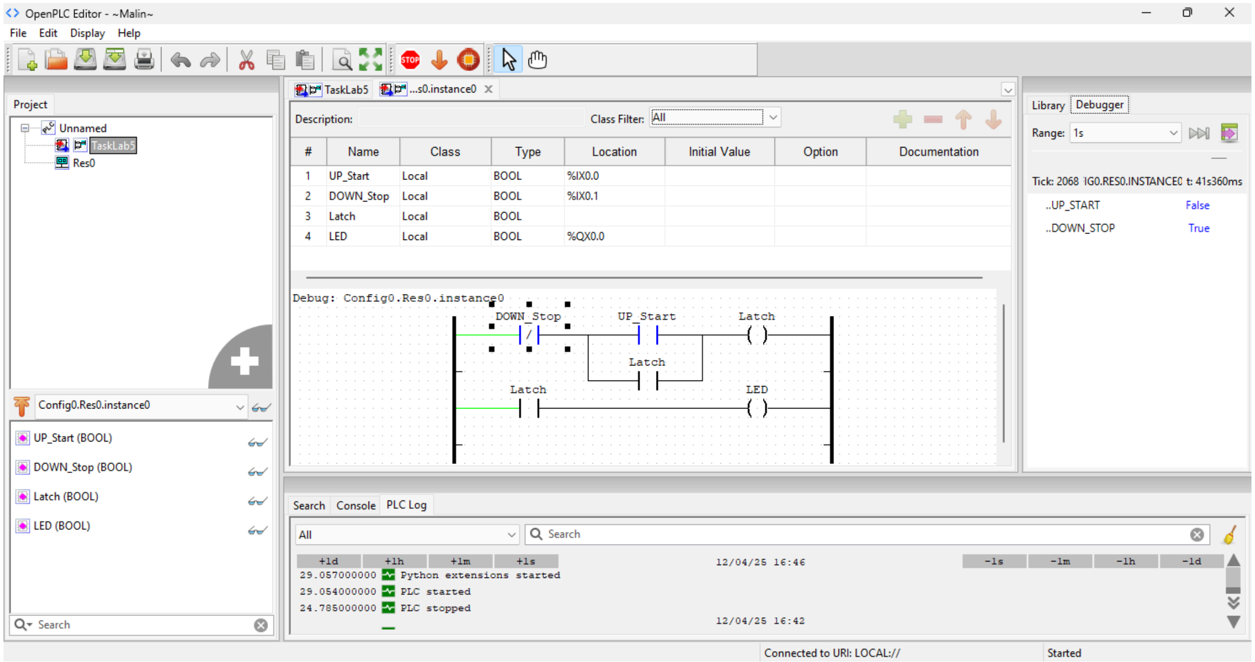


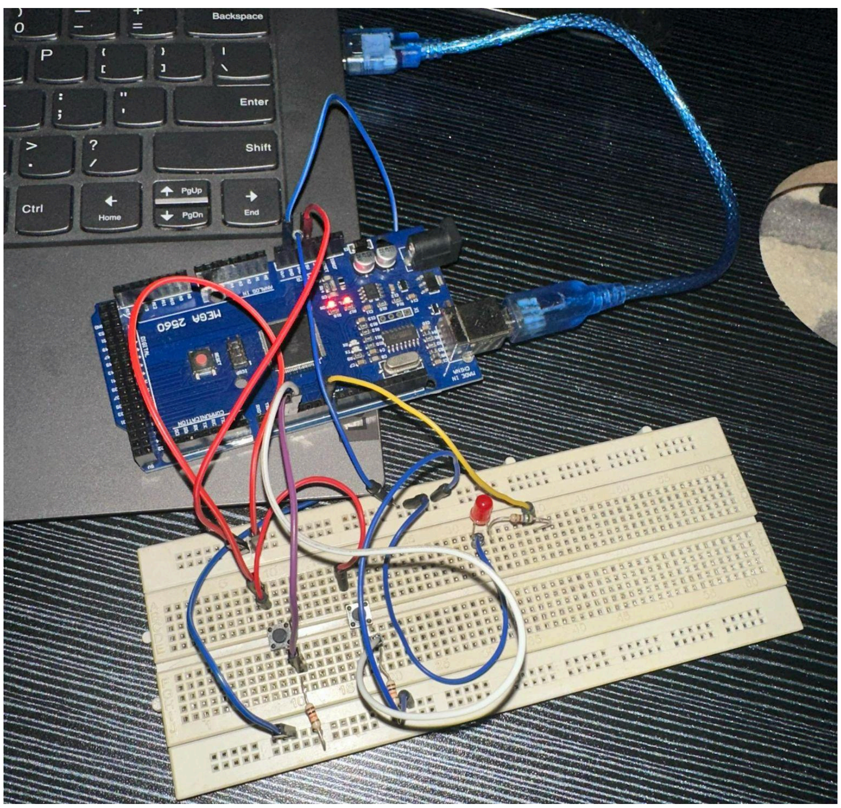
2) When the UP_Start button is released and pressed again, the LED will still turn on.



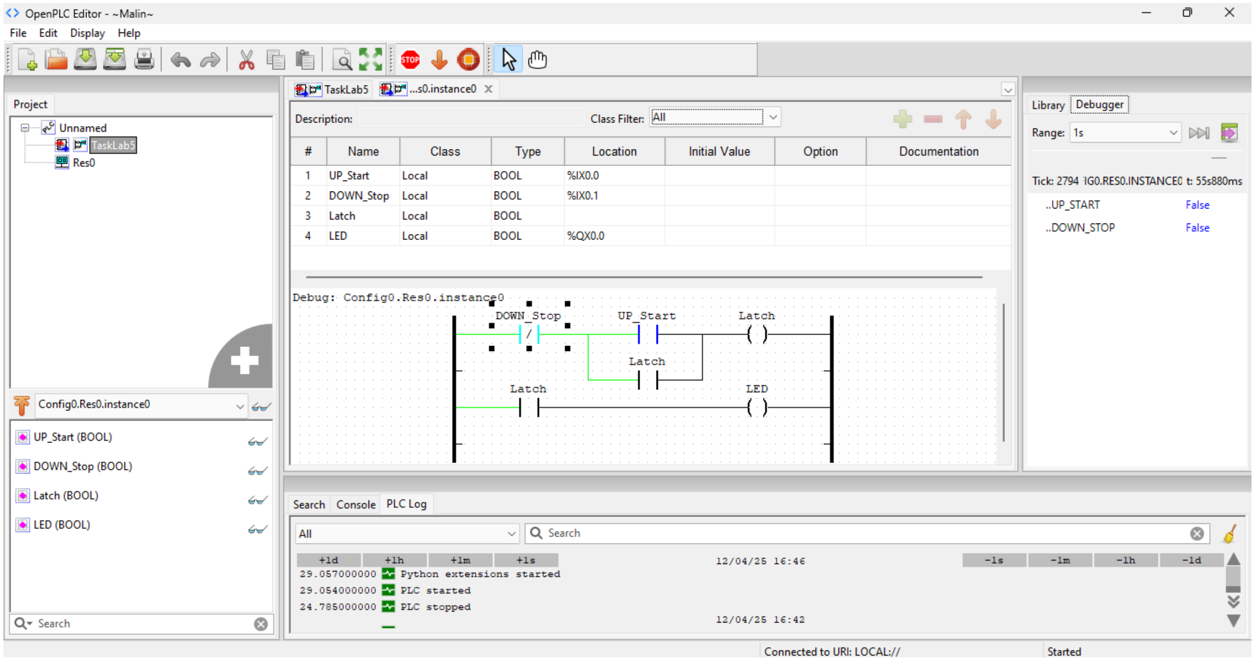


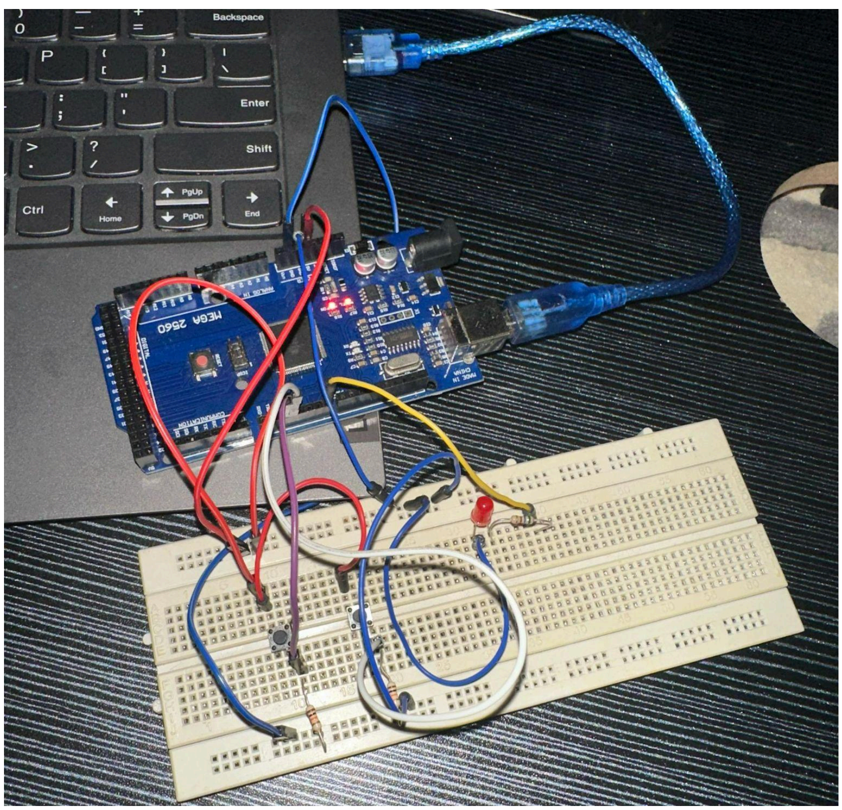
3) When the DOWN_Stop button is pressed, the LED will turn off.





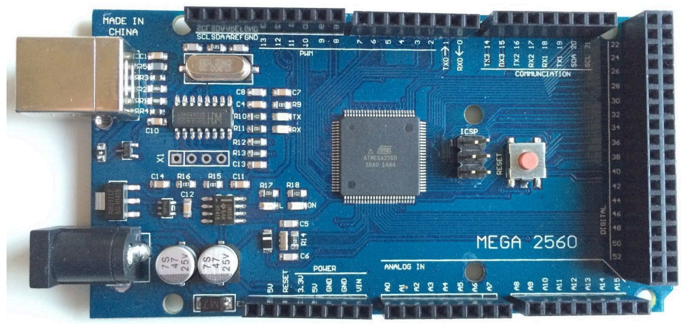
4) When the DOWN_Stop button is released and pressed again , the LED will still turn off.





Instruments used for data acquisition:

1) Arduino Mega 2560: A microcontroller board based on the ATmega2560, used to process input signals from pushbuttons and control the 7-segment display.



2) Push buttons: The count-ip button increments the displayed number each time it is pressed from 0 to 9 and the reset button resets the display to zero.

3) Light Emitting Diode (LED) : LED used to produce light up to 90% more efficiently than incandescent light bulbs.

4) Resistors: 560-ohm resistor used to limit the current flowing through it, preventing excessive current that could damage the LED. While 10k ohms resistors used for the pushbuttons to ensure stable HIGH signals when the buttons are not pressed.
5) Jumper wires: Used to establish electrical connections between components on the breadboard.
6) Breadboard: A prototyping platform used to connect components without soldering, allowing for easy circuit modifications.

6.0 DATA ANALYSIS

During the experiment, the ladder diagram designed in the OpenPLC Editor was successfully compiled and uploaded to the Arduino board. The inputs and outputs were tested through manual interaction with the push-button switches and observation of the LED behavior. The system responded as expected, pressing the "Start" button activated the LED, which remained on due to the latch logic implemented in the ladder diagram. Pressing the "Stop" button deactivated the LED, breaking the latch and returning the system to its initial state.

The correct operation of the circuit validated the logic design and hardware implementation. The input pins that connected to the push buttons provided digital HIGH or LOW signals depending on whether the buttons were pressed or released. These signals were accurately interpreted by the Arduino running the ladder program. The output pin controlled the LED based on the internal state of the latch, proving that the logic was properly latched and unlatched by the corresponding inputs.

No complex mathematical models were required for this experiment, but digital logic analysis confirmed that the expected truth table for the system was satisfied. For example, with the system in the OFF state, the LED remained off until the Start input was momentarily set HIGH. Once HIGH, the latch logic held the output ON until the Stop input was triggered. This behavior aligns with theoretical expectations for a latched Start-Stop control system, confirming the accuracy of the ladder logic program.

The significance of this data lies in its demonstration of fundamental industrial control principles. The successful implementation of latching logic using software-based PLC programming on physical hardware (Arduino) showcases the integration of software and hardware in automation systems. It also reinforces the reliability of ladder logic for implementing real-time control processes. This experiment serves as a foundational exercise in understanding how PLCs manage sequential control tasks in industrial settings.

7.0 RESULTS

The Start-Stop Control Circuit was successfully implemented and tested using the Arduino board and the ladder logic program developed in the OpenPLC Editor. Upon uploading the compiled ladder diagram to the Arduino, the system functioned as intended. When the "Start" push button was pressed, the LED connected to Pin 7 turned on and and remained on even after the button was released, indicating that the latching logic was correctly maintaining the output state. Pressing the "Stop" button successfully turned the LED off, breaking the latch and returning the system to its initial state.

These results confirm the correct operation of both the hardware setup as shown in Figure 1 and the ladder logic program shown in Figure 3. The LED’s state consistently matched the expected logic behavior throughout multiple test cycles. The system demonstrated reliable performance without any erratic behavior, delays, or failed transitions between states.

The behavior of the system closely matched the expected results based on the logic design. The LED reliably turned on and off in response to the respective inputs, confirming the correct implementation of the latching mechanism in the ladder logic. All components, including the resistors, push buttons, and LED, functioned as expected throughout the test. No anomalies or deviations from the expected control behavior were observed during the experiment.

Here’s the link of video for the result:

https://github.com/amalinblqs/MCTA3203_GROUP_9/blob/main/Lab%205/Video%20Task.mp4

8.0 DISCUSSION

The observed results validate the correct implementation of the basic Start-Stop control system using ladder logic and an Arduino. The circuit behaved as intended: the LED remained on, allowing the system to maintain its output state after pressing the Start button and turned off only when the Stop button was pressed, indicating that the latching logic was functioning correctly. This accurately simulates a basic industrial control circuit where continuous operation is needed after a momentary start input. This confirms that the OpenPLC software can effectively simulate and transfer ladder logic to an Arduino, bridging the gap between traditional PLC systems and low-cost microcontroller platforms.

The implications of this result are significant, especially in educational and prototyping environments, where Arduino-based systems can be used to model industrial automation concepts at a lower cost. Additionally, the experiment reinforces key automation principles such as momentary inputs, output latching, and digital control, all essential for designing larger and more complex control systems. These results also highlight the practicality of using OpenPLC for real-world control applications and serve as a foundational step toward developing more advanced automation solutions.

There were no significant discrepancies between expected and observed outcomes in this experiment. However, minor delays in the system's response could occur if button presses were too brief or inconsistent, which may be attributed to mechanical bounce in the push buttons. While not significant in this experiment, such issues could be mitigated with debouncing techniques in future iterations.

One limitation of the experiment is the lack of feedback or fault handling, which are typically present in industrial systems for added reliability. Furthermore, the experiment was conducted using a single output (LED), which limits the complexity of the control logic. Expanding the system to include multiple outputs or interlocks would provide a more comprehensive understanding of PLC capabilities. Another limitation of the experiment is the potential for incorrect resistor values affecting input signal reliability. Although pull-down or pull-up resistors are used to stabilize the input signal from the push buttons, if the resistor values are too high or too low, they might cause floating inputs or excessive current draw, respectively. This could lead to inconsistent behavior such as false triggering or failure to detect button presses, especially in noisy environments. In more robust industrial systems, precise resistor values and additional filtering are often implemented to ensure signal integrity.

Question:

Develop a Start-Stop Control Circuit by using ladder diagram created in OpenPLC, compile, simulate and transfer the ladder diagram program to Arduino Board. The circuit with pin configuration is shown in Fig.4 below.

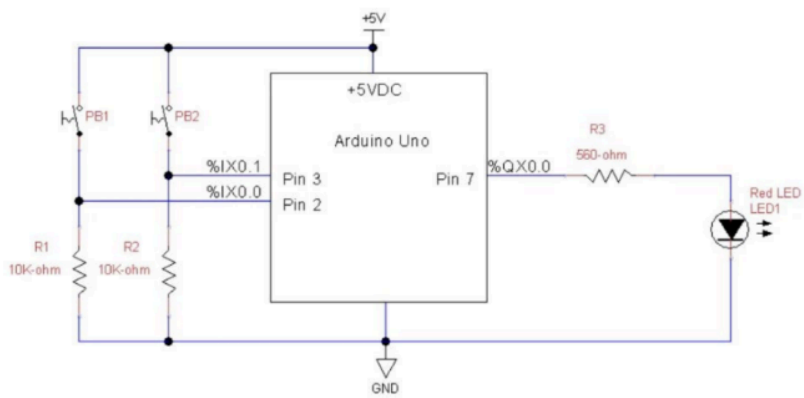


Fig. 4: Start-Stop Control Circuit

Answers:

Firstly, we assembled the Start-Stop Control Ciruit, following the layout in Fig. 4 that connects various components like push-buttons, resistors, and an LED to an Arduino Uno microcontroller. The circuit setup included connecting two push-button switches: one is configured as the start button to light up the LED while the other as the stop button to break the circuit which turns off the LED. The LED is an indicator if the circuit is active or inactive, and the Arduino Uno serves as the controller, coordinating the start-stop actions based on the ladder logic program from OpenPLC editor. The resistors were used to prevent any fluctuations of current flowing through the circuit.

9.0 CONCLUSION

This experiment successfully demonstrated the design and implementation of a basic Start-Stop Control Circuit using ladder logic programmed in the OpenPLC Editor and executed on an Arduino Mega 2560 board. The main findings confirmed that the system performed reliably, pressing the Start button activated the LED and latched the output, while pressing the Stop button effectively deactivated it. These results validate the hypothesis that a simple Start-Stop logic could be implemented using PLC-style programming on Arduino hardware to mimic industrial control systems.

The significance of this experiment lies in its ability to bridge theoretical control concepts with practical hands-on experience using accessible components. By using ladder logic to program the Start-Stop function and deploying it on an Arduino, students gain valuable insight into how industrial processes are automated using PLCs. The successful operation of the circuit further reinforces fundamental automation principles such as latching, momentary inputs, and digital output control.

Additionally, the experiment highlights the broader applications of Arduino-based control systems in educational, prototyping, and small-scale industrial environments. It also serves as a stepping stone toward more complex control logic and real-world automation systems, encouraging future development in areas such as interlocking, sensor integration, or feedback mechanisms. Overall, the results supported the experiment's objectives and provided a strong foundation for understanding programmable logic control in both software and hardware contexts.

10.0 RECOMMENDATIONS

Improvements for future iterations of this experiment, it would be beneficial to incorporate more advanced features to enhance the system's complexity and realism. For instance, integrating fault detection mechanisms could improve the system's robustness and reliability. In industrial systems, feedback and fault management are crucial for detecting and responding to issues that could arise during operation, such as a malfunctioning input device or unexpected system behavior. Implementing software-based debouncing for the pushbuttons would also address any mechanical bounce issues that might cause unreliable inputs. In addition, expanding the system to control multiple outputs or include interlocks would demonstrate the versatility of PLC-based control systems and allow students to explore more sophisticated control logic.

Another potential modification could be to integrate sensors that provide feedback to the system, allowing for automated decision-making or conditional control. For example, using a temperature sensor to control the Start-Stop system based on environmental conditions could simulate real-world industrial processes, where external factors influence the operation of machinery. Additionally, adding a display such as an LCD or 7-segment display to provide real-time status feedback would enhance the interactivity and usability of the system, making it more user-friendly for troubleshooting and monitoring.

One lesson learned from this experiment is the importance of clear communication between software and hardware components. The process of designing and implementing ladder logic in OpenPLC and transferring it to Arduino illustrates the need for a solid understanding of both digital control principles and the underlying hardware. It also emphasizes the value of simulation tools in verifying control logic before deployment to physical systems. The successful operation of the Start-Stop control circuit is a testament to the importance of careful testing and validation at every stage of the design process.

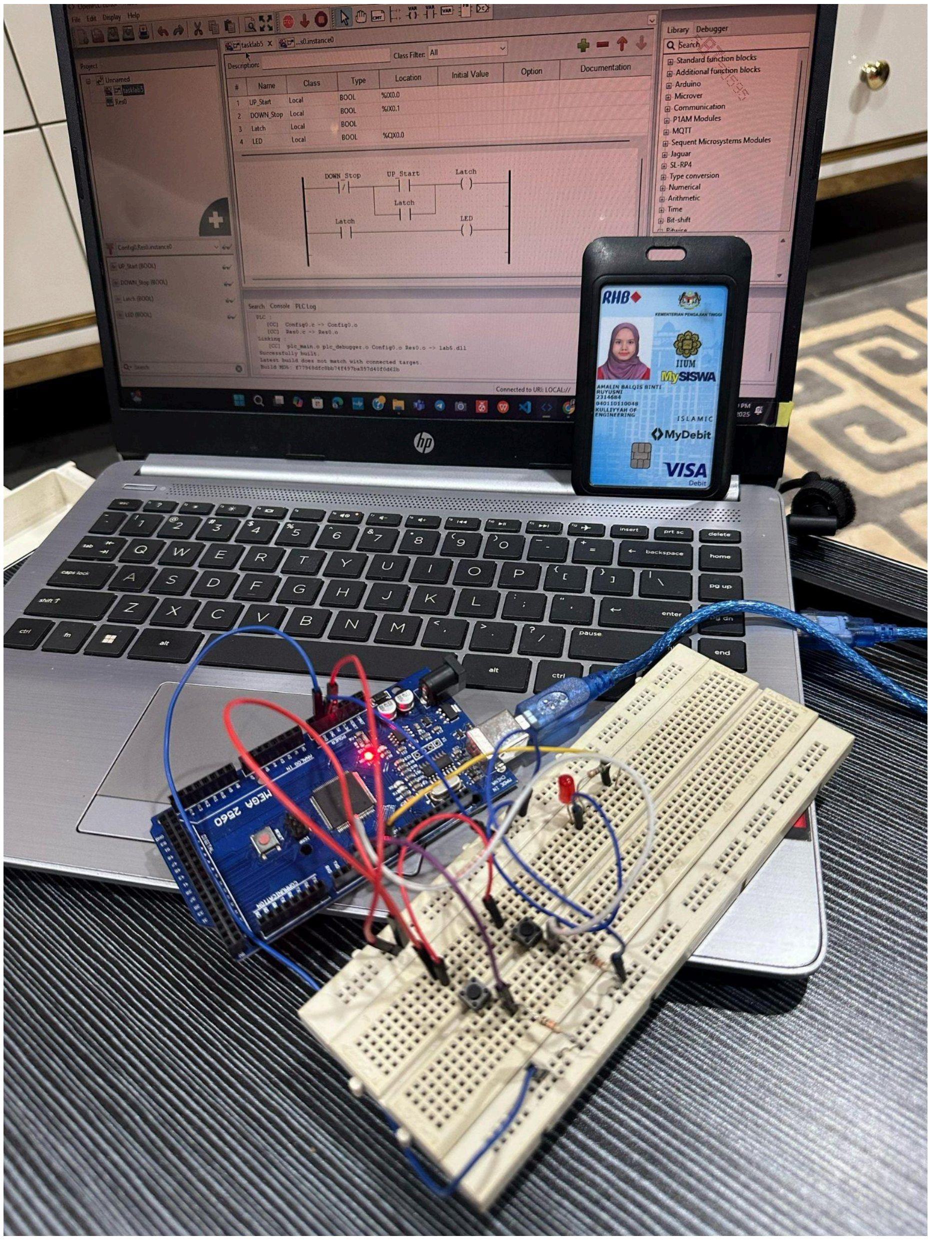
Another important lesson is the significance of addressing small but impactful factors like mechanical bounce in pushbuttons. While this was not a major issue in this experiment, in real-world applications, such issues can lead to unreliable operation and need to be mitigated. Incorporating debouncing techniques, either through hardware or software, ensures the system responds consistently to user inputs.

Lastly, it is essential to remember that the simplicity of the circuit in this experiment is only a starting point. As students progress in automation and control systems, they will need to build on this foundation and address more complex real-world challenges, such as handling multiple inputs, ensuring system safety, and managing power efficiently. Expanding the scope of the experiment to include multiple outputs, fault handling, or communication with external devices would provide a deeper understanding of how industrial control systems operate in practice.

11.0 REFERENCES

Wilcher, D. 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-stop-circuit/

APPENDICES


DOWN_Stop
UP_Start
Latch
Latch
Latch
LED
RHB
IIUM
MySISWA
ISLAMIC
MyDebit
VISA
Debit
hp
MEGA 2560

ACKNOWLEDGMENTS

First of all, we would like to express our deepest appreciation to all those who provided us the possibility to complete this report. We would like to express our sincere gratitude to our lecturer, Assoc. Prof. Eur. Ing. Ir. Ts. Gs. Inv. Dr. Zulkifli Bin Zainal Abidin and Dr. Wahju Sediono for their invaluable guidance and support throughout this lab. Their insightful explanations and encouragement greatly enhanced our understanding of the concepts covered.

Next, a special thanks goes to our own teammates, Wafdi, Amalin and Farha, who helped to build up the circuit and set up the programming code using Arduino IDE, and also successfully completed this experiment. Last but not the least, we would like to thank everyone, especially our classmates who are willingly helping us out in the lab experiment directly or indirectly.

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 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: *Wafdi* Read [/]
Name: Amer Wafdi bin Zainizam Understand [/]
Matric Number: 2228703 Agree [/]
Contribution: Abstract, materials and equipment, experimental setup & discussion

Signature: *Amalin* Read [/]
Name: Amalin Balqis binti Ruyusni Understand [/]
Matric Number: 2314684 Agree [/]
Contribution: Introduction, data collection, data analysis & results

Signature: *farha* Read [/]
Name: Farha Qaisara binti Mohd Zahidi Understand [/]
Matric Number: 2315994 Agree [/]
Contribution: Methodology, conclusion & recommendations