SEM 1 25/26

SECTION 1

COURSE CODE: MCTA 3202

LECTURER NAME:
ZULKIFLI BIN ZAINAL ABIDIN

WEEK 7: PLC INTERFACING WITH MICROCONTROLLER AND PC OVER ETHERNET/IP

PREPARED BY:
GROUP 1

NAME	MATRIC NUMBER
AMIR MUIZZUDDIN BIN NORAIDIL RAZMAN	2129579
MUHAMMAD HAIKAL HANIF BIN ABDUL RAZAK	2213297
AHMAD ILHAM BIN ABDUL AZIZ	2112109

INTRODUCTION

Programmable Logic Controller (PLC) is a specialized industrial computer used for controlling and automating electromechanical processes in manufacturing plants, machinery, and various other applications. Today's purpose is to introduce us to PLC and how to make a circuit using an app called OpenPLC. By the end of this, we are able to make a proper and functional PLC program through OpenPLC and integrate into Arduino.

MATERIALS

The following hardware and software were used in the experiment (as listed in the module document):

Hardware

- Arduino Board (e.g., Arduino Mega/Uno)
- 2 Push Button Switches
- LED
- Resistors
- Jumper wires
- Breadboard

Software

- OpenPLC Editor
- Arduino USB driver/software for uploading programs

EXPERIMENT SETUP

The experiment setup consists of creating a ladder diagram in OpenPLC Editor, simulating it, assigning variables to Arduino pins, and physically wiring the control circuit.

3.1 Software Setup

1. OpenPLC Editor is installed and launched.
2. A new project is created using Ladder Diagram (LD) language.
3. Variables are defined with appropriate data types (e.g., BOOL).
4. The ladder diagram is designed using power rails, contacts, coils, and timer blocks.

3.2 Hardware Setup

- A Start-Stop control circuit is assembled as shown in the file (Fig. 6).
- Push buttons are connected for Start and Stop functions.
- An LED is used as an actuator output.
- Resistors are added to protect the circuit.
- Arduino is connected via USB to the PC, and the correct COM port is selected.

3.3 Pin Configuration

OpenPLC variables are mapped to Arduino digital pins following the Physical Addressing table from OpenPLC Runtime documentation. For example, an LED connected to Arduino pin 14 corresponds to %QX0.0 in OpenPLC.

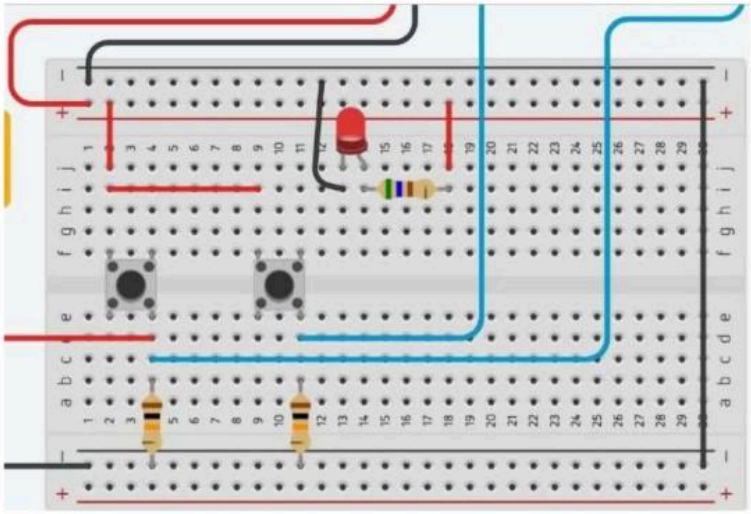

Fig. 6

METHODOLOGY

This experiment investigates the integration of a Programmable Logic Controller (PLC) environment with a microcontroller (Arduino) using the OpenPLC Editor. The methodology consists of two major parts: (1) configuring and simulating a ladder diagram in OpenPLC, and (2) uploading and testing the program on physical hardware.

SOFTWARE PREPARATION AND PROJECT SETUP

1. **Download and Install OpenPLC Editor**
   The OpenPLC Editor was obtained from the official website and installed on a PC as instructed in the file (p.1) .

2. **Create a New PLC Project**
   A new project folder was created, followed by generating a new Program Organisation Unit (POU) set to *Ladder Diagram (LD)*.
   This established the working environment for designing the control logic.

3. **Define Variables**
   Local Boolean variables were added to represent inputs, outputs, timers, and internal coils.

Variable classes and types were defined consistent with PLC addressing syntax (e.g., %IX0.0, %QX0.0).

LADDER DIAGRAM DEVELOPMENT AND SIMULATION

1. A Start–Stop latch circuit was constructed using:

- **UP_Start** pushbutton (Start)
- **DOWN_Stop** pushbutton (Stop)
- **Latch** internal memory coil
- **LED** output coil

2.The latch logic was implemented to ensure the LED remains ON after the Start button is pressed, and turns OFF only when the Stop button is pressed.

3. The program was compiled and simulated inside OpenPLC to confirm that:

- The start button energises the latch.
- The stop button breaks the latch and de-energises the LED.

UPLOADING LADDER LOGIC TO ARDUINO

1. Assign Physical Addresses
   PLC variables were mapped to Arduino digital pins based on the OpenPLC "Physical Addressing" documentation.
   Example: if the LED is connected to Arduino pin 7, it is assigned to %QX0.0 as shown in Fig. 4 (p.4) .

2. Compile and Transfer Program
   The Arduino board was connected to the PC, and the correct COM port was selected.
   Using the Transfer to PLC function (Fig. 3, p.3) , the compiled PLC logic was uploaded into the Arduino runtime.

HARDWARE IMPLEMENTATION

1. Circuit Assembly
   Following Fig. 4 and Fig. 6 (p.4–5) :

- Two pushbutton switches (Start and Stop) were connected with pull-down resistors.
- The LED and limiting resistor were connected to the assigned output pin.
- All components shared a common ground with the Arduino board.

2. Testing and Verification
   The system was powered, and the Start–Stop sequence was tested:

- Pressing Start energised the latch and turned ON the LED.
- Pressing Stop disengaged the latch and turned OFF the LED.
- System response was compared to the simulation to ensure correctness.

CONCLUSION

This experiment successfully demonstrated how a PLC-based control system can be designed, simulated, and deployed onto a microcontroller using OpenPLC. By creating both a blinking LED program and a Start–Stop control circuit, the experiment proved that ladder logic can be seamlessly integrated with Arduino hardware through Ethernet-based PLC workflow. The simulation validated the logic prior to deployment, while the hardware test confirmed that the Start button correctly latched the output and the Stop button reliably reset the circuit.

Overall, the experiment strengthened understanding of PLC programming fundamentals, real-time I/O mapping, and the practical interfacing of industrial-style control logic with embedded microcontroller systems.