INFOKUM
Volume 13, Number 04, 2025, DOI 10.58471/infokum.v13i03
ESSN 2722-4635 (Online)
https://infor.seaninstitute.org/index.php/infokum

Implementation of Low-Cost Programmable Logic Controller (PLC) Based on Arduino Uno as a Magnetic Relay Control System

Sutan Aradji Larby Siregar[1], Ahmad Dani[2], Solly Aryza[3]
[1,2,3]Universitas Pembangunan Panca Budi, Medan, North Sumatera, Indonesia

Article Info
Keywords:
Arduino-based PLC,
LDmicro,
Ladder diagram,
Magnetic relay,
Arduino Uno

ABSTRACT
This research aims to develop and implement a magnetic relay control system based on a low-cost Programmable Logic Controller (PLC) using Arduino Uno and LDmicro programming. The main focus of this research is to create a reliable, efficient, and affordable control system that can be easily integrated into industrial and household applications. The hardware design was carried out systematically, involving the selection of main components, electronic circuit design, and physical arrangement of components, considering efficiency and safety. Software testing was conducted using LDmicro to design and simulate ladder diagrams, and XLoader to upload the program to the Arduino Uno microcontroller. The test results show that the control system based on Arduino Uno and LDmicro functions well in controlling the magnetic relay according to the designed logic. This ladder diagram-based approach proves to be simple, efficient, and economical for developing control systems. The implementation of an Arduino Uno-based PLC offers an affordable alternative to conventional automation solutions, making it suitable for applications in education, prototype development, and small-scale industries. This research is expected to contribute significantly to the development of low-cost automation technology and open up broader opportunities for implementation in various sectors.

Corresponding Author:
Sutan Aradji Larby Siregar
Universitas Pembangunan Panca Budi, Medan, North Sumatera, Indonesia
sutanaradji50@gmail.com

INTRODUCTION

The Fourth Industrial Revolution (Industry 4.0) has brought significant changes to the world of industrial automation, where Programmable Logic Controller (PLC)-based control systems have become one of the main components supporting efficiency and productivity. PLCs are used to control various automation processes in industries such as manufacturing, material processing, and automated transportation systems (Masco, 2022). However, one of the main obstacles in implementing PLCs is the high cost, both in terms of hardware and software required for operation. This high cost often becomes a barrier for educational institutions, startups, and small and medium enterprises (SMEs) in adopting automation technology(Agbomemewa et al., 2024).

To address this issue, research and development of low-cost microcontroller-based control systems such as Arduino have gained attention. Arduino Uno, as one of the most popular variants of this microcontroller platform, offers a low-cost solution that can be implemented in various automation applications (Sinaga et al., 2024). Arduino excels in terms of ease of programming, flexibility in integrating with various sensors and actuators, and the availability of a large and active community that continuously develops various add-ons and programming libraries (Minchala et al., 2020). With its affordable price, Arduino becomes an ideal choice for building simple control systems that can be customized to meet specific user needs (Khan et al., 2020).

As supporting software, LDmicro is an open-source programming tool specifically designed for microcontroller-based systems such as Arduino. LDmicro allows users to program Arduino using *Ladder Diagram* (LD), a standard programming language in the industrial automation sector. The LD language provides an easy way to represent control logic in graphical diagrams resembling electrical circuits, facilitating technicians and engineers familiar with conventional PLC-based control systems (Wardhana et al., 2021). By utilizing LDmicro, Arduino can function like a conventional PLC capable of controlling various automation processes, such as controlling relays, sensors, electric motors, and other actuator systems (Dani & Erivianto, 2023; Hidayati et al., 2017).

Previous studies have shown significant potential for Arduino-based PLC applications in various fields. (Anugrah, 2016) developed a heart rate measurement system based on micro-PLC for healthcare applications, while Aryza et al., 2020 successfully implemented an automatic sliding door system controlled by Arduino Uno and GSM SIM900A module. Another study by Sarifuddin et al., 2021 developed a miniature public fuel station (SPBU) controlled using a PLC-based Human-Machine Interface (HMI). These applications demonstrate that, despite being low-cost, Arduino-based control systems can be reliable in various practical applications.

One interesting application of Arduino-based PLC is in controlling magnetic relays, an essential component in electrical control systems. Magnetic relays allow the control of large electrical currents using small control currents, thereby increasing system efficiency and safety. In this context, the combination of Arduino Uno and LDmicro offers an efficient solution for controlling magnetic relays with complex control logic that is still easy to program and implement (Suyetno, 2022)

Furthermore, the development of Arduino-based control systems offers significant opportunities in education. With a much lower cost than conventional PLCs, Arduino can be used as a learning module for students to understand the basics of industrial automation. (Hidayati et al., 2017) noted that using microcontroller-based PLCs in learning can enhance students' understanding of control logic and programming concepts, while providing practical experience relevant to modern industrial needs.(Dani & Erivianto, 2024)

This research aims to develop and implement a magnetic relay control system using a low-cost PLC based on Arduino Uno, programmed with LDmicro. The focus of this research is on developing a reliable, efficient, and affordable control system that can be easily integrated into industrial and household applications. With this approach, it is hoped that this research can make a tangible contribution to the development of low-cost automation

INFOKUM
Volume 13, Number 04, 2025, DOI 10.58471/infokum.v13i03
ESSN 2722-4635 (Online)
https://infor.seaninstitute.org/index.php/infokum

technology and open up broader implementation opportunities in education, startups, and UKM.

METHOD

The hardware design in the Arduino Uno-based control system using magnetic relays requires a systematic approach to ensure all components function according to system requirements. This process involves the identification and selection of main components, electronic circuit design, and physical arrangement of components, considering efficiency and safety. The following explanation presents the technical details of the hardware design with an academic approach to describe the implementation details of the Arduino Uno-based magnetic relay control system.

Main System Components

a. Arduino Uno

Arduino Uno is a microcontroller based on ATmega328 with sufficient capacity to run simple to intermediate control logic. Arduino Uno has 14 digital pins (6 of which can be used as PWM) and 6 analog pins. In this system, Arduino Uno functions as the main controller that receives input from digital switches or sensors and sends output signals to activate the relay module. The advantages of Arduino Uno include flexibility, extensive documentation availability, and the ability to integrate with LDmicro software that supports ladder diagram programming.

Figure 1. Arduino uno board

b. Modul Relay

Relay modules are used to control electrical devices that operate at high voltages or large currents, such as lights, electric motors, or other household appliances. The module has two main parts, namely the control part (which receives the signal from the Arduino) and the load part (which is connected to the electrical device). In general, the relay modules used have the ability to handle currents up to 10 A at a voltage of 250 VAC or 30 VDC.

Figure 2. Relay module

c. Switches as Digital Inputs
   The digital switches and sensors serve as input devices that provide signals to the Arduino Uno. Switches are used to manually enable or disable certain functions, while digital sensors such as proximity sensors, photodiodes, or limit switches are used to provide automatic inputs based on environmental conditions.
d. Power Supply
   The power supply is a critical element to maintain the stable operation of the system. The Arduino Uno can be powered via a USB cable or a DC adapter with a voltage range of 7-12 V. However, for efficiency, the system is designed using a separate power supply for the Arduino Uno and the relay module. Relays often require additional power (usually 5 V) to activate their coils. Thus, a separate power source helps to reduce interference between the low-voltage control circuit and the high-voltage load circuit.
e. Connecting Cables and Breadboards
   At the prototype stage, a breadboard and jumper cable are used to connect all components flexibly. The use of breadboards allows for easy repair or alteration of the series.

Electronic Circuit Design

The design of the electronic circuit aims to ensure the integration of components runs well and the control logic can be applied as needed. Here are the steps of network planning:

1. Input to Arduino Uno
   The digital switch or sensor is connected to the Arduino Uno digital input pin. Each input pin is equipped with a pull-down resistor to ensure that when the switch or sensor is inactive, the input signal is at a stable low logic level (LOW). Pull-down resistors typically have a value of 10 kΩ to limit the current flowing to the input pins.
2. Arduino Output to Relay Module
   The Arduino digital output is connected to the input terminal of the relay module. When the Arduino sends a high logic signal (HIGH), the transistor on the relay module will conduct current to the relay coil, thus activating the relay contacts. These contacts can be used to control electrical devices operating at high voltages.
3. Power Supply
   In order to maintain the stability of the system's operation, the power supply for the Arduino and the relay module are isolated from each other but still share the same

ground reference. This approach reduces the risk of electromagnetic interference (EMI) originating from inductive loads on the relay.

Physical Arrangement of Components

Once the circuit design is complete, the next step is to physically assemble the components. This arrangement pays attention to a neat and efficient layout, with sufficient spacing between components to prevent inter-circuit interference. In the prototype stage, breadboards are used to facilitate testing and modification of the network. At this stage, all components are assembled on the breadboard using jumper cables. This arrangement is temporary and flexible, allowing for design changes if needed. Components such as resistors and protection diodes are placed as close as possible to the relay to maximize the protection function.

LDMicro Programming

Programming is a crucial step in the design of Arduino Uno-based control systems, especially when used to control magnetic relays. In this study, LDmicro is used as the main programming tool to design ladder diagram-based control logic. LDmicro was chosen because it was able to translate ladder logic into code compatible with Arduino Uno microcontrollers in HEX file format, which can then be uploaded to hardware using software such as XLoader. The following are the stages of programming with LDmicro applied in this study:

1. Creation of Ladder Diagram-Based Control Programs
   LDmicro is designed to make it easier to create control programs with a ladder diagram approach. A ladder diagram is a graphical representation of the control logic that resembles an electrical circuit. In this diagram, elements such as normally open (NO) contacts, normally closed (NC) contacts, and coils are used to represent inputs, processes, and outputs.
2. Export Ladder Programs to HEX Files
   Once the ladder diagram is finished being designed and verified using the simulation feature in LDmicro, the next step is to export the program into a code format that can be run by the Arduino Uno. This process is done through the *compile* function in LDmicro, which generates a file with the . HEX. This HEX file is a digital representation of ladder logic that has been designed and adapted to the ATmega328 microcontroller architecture, which is the core of the Arduino Uno. With this HEX file, the code can be uploaded directly to the hardware without the need to manually write programs in C or C++.

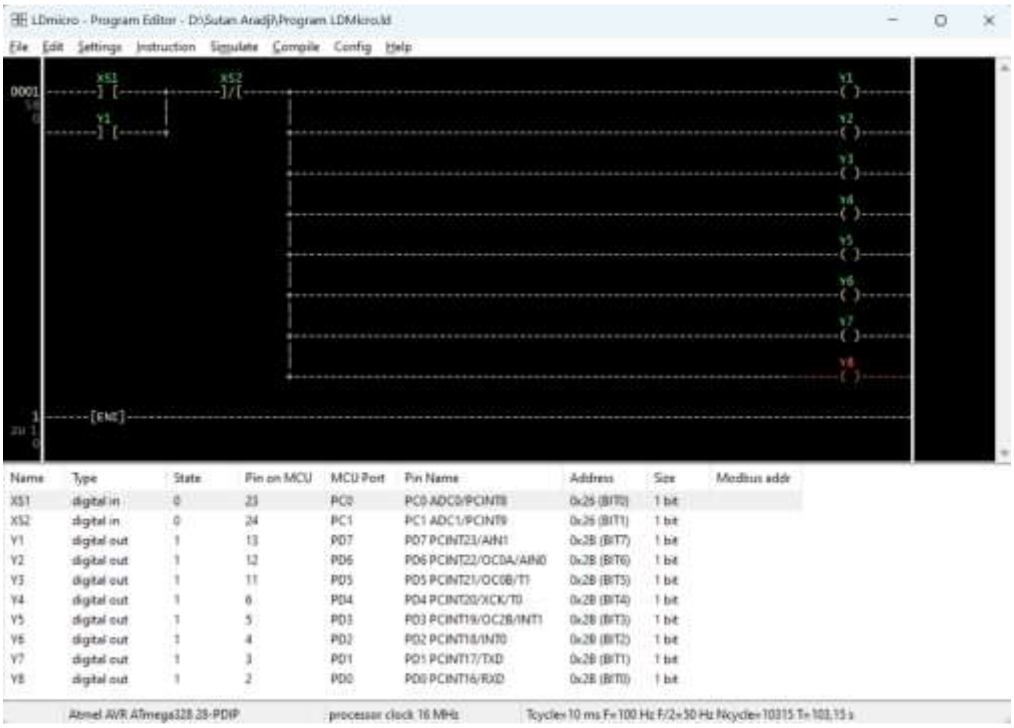

Figure 3. Program LDMicro

3. Uploading Code to Arduino Using XLoader

   The final stage is to upload the HEX file to the Arduino Uno. This process involves software such as XLoader, which allows communication between the computer and the microcontroller via a USB port.

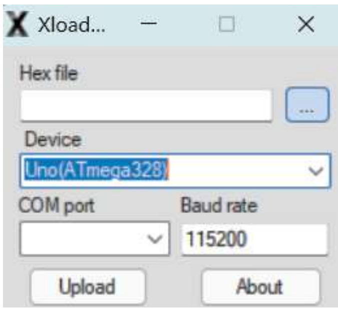

Figure 4. Software Xloader

The implementation process involves designing the control logic with a ladder diagram, converting the program into a file that can be run by the microcontroller, to testing to ensure that the system works according to the design. This article discusses the stages of implementation and testing in a structured manner within a scientific framework. Tests are carried out to validate that the magnetic relay control system works according to the specifications of the control logic for which it has been designed. The testing process includes the following steps:

1. Basic Functionality Testing

   The initial step of testing is to verify the basic functionality of the system. The magnetic relay is tested to ensure that the Relay is active when the switch is pressed, the Relay is off when the switch is released, the Timer works correctly to provide a delay before the relay is active. Each scenario is tested by monitoring the system's response to the given input.

2. System Reliability Testing
The reliability of the system is tested by providing repeated inputs to ensure that the relay responds consistently. Testing is performed over multiple operating cycles to identify potential errors such as delayed response or relay failure to turn on/off.

RESULTS AND DISCUSSION

Result Testing Configuration.

Software testing was carried out to ensure that the ladder diagram-based control system designed using LDmicro can function optimally, from the simulation stage to implementation into the Arduino Uno hardware. This process includes testing the LDmicro software to design and simulate ladder diagrams, as well as testing program uploader software, such as XLoader, to integrate the program into the microcontroller. Here are the testing steps carried out in this study:

1. Microcontroller Parameter Configuration
The first step in testing the LDmicro is to set the cycle time and crystal frequency parameters on the "setting" menu according to the technical specifications of the Arduino Uno. The microcontroller used in this study is an ATmega328 with a 28-PDIP configuration, which is selected through the "setting" menu. This adjustment aims to ensure that the ladder diagram simulation is designed according to the capabilities and characteristics of the microcontroller used.
2. Ladder Diagram Design
The ladder diagram design begins by entering the basic elements of control, namely one coil and eight contacts, through the "instruction" menu on the LDmicro. These elements are then named to represent the system's inputs and outputs. The naming process is carried out by double-clicking each coil and contact, then giving a label through the *toolbox name*. In this study, the contact is named S1, which represents the switch as the input (switch1), while the coil is named L1 to L8, which represents the lamp as the output.
3. Input and Output Pin Configuration

Table 1. Input and Output Pin Configuration

Input	PIN	Input	PIN
S1 (Input)	Pin 23 (PC0)	L4 (Output)	Pin 6 (PD4)
S2 (Input)	Pin 24 (PC1)	L5 (Output)	Pin 5 (PD3)
L1 (Output)	Pin 13 (PD7)	L6 (Output)	Pin 4 (PD2)
L2 (Output)	Pin 12 (PD6)	L7 (Output)	Pin 3 (PD1)
L3 (Output)	Pin 11 (PD5)	L8 (Output)	Pin 2 (PD0)

Once the ladder diagram elements have been designed and named, the next step is to set up the input and output pin configurations. This configuration includes the adjustment of each of the ladder diagram elements with the I/O ports on the Arduino

Uno can be seen in table 1. This configuration is done to ensure that the inputs and outputs on the ladder diagram correspond to the physical ports on the Arduino Uno that will be used to control the relay and other devices.

4. Ladder Diagram Simulation

Simulation is done by accessing the simulation feature on the "simulate" menu and selecting the real-time simulation start option. In this simulation mode, the user can validate the control logic designed on the ladder diagram. Contact activation is done by double-clicking on the S1 element. If the simulation runs correctly, a red dotted line will appear connecting S1 with L1, indicating that the S1 input has been activated and produces outputs in the form of L1 to L8 activations.

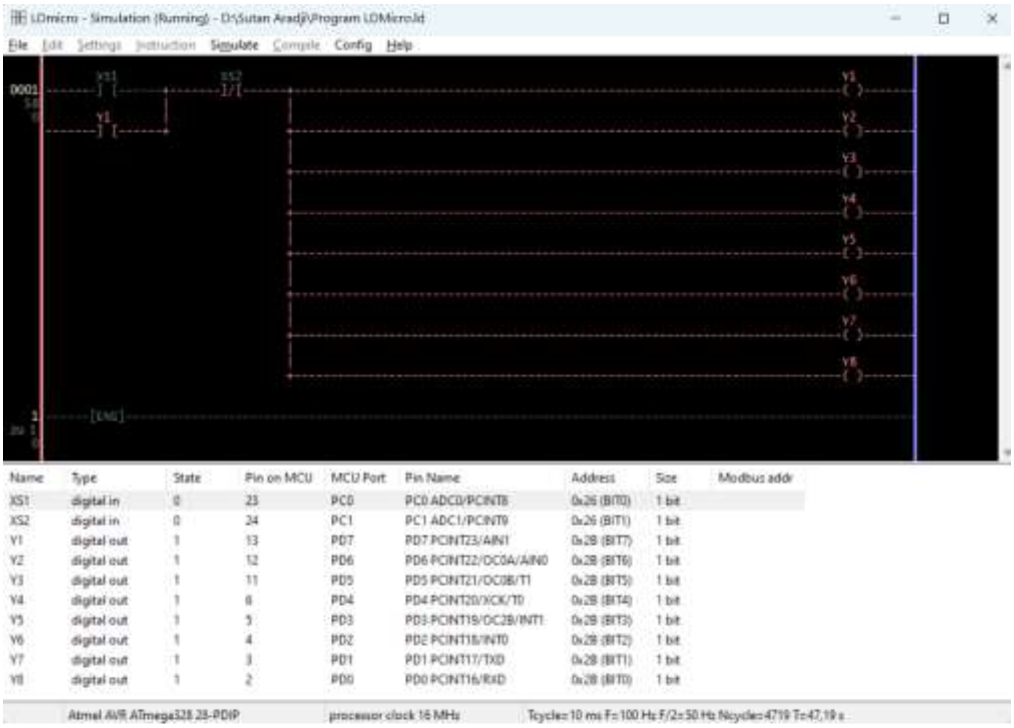

Figure 5. Program Simulation

5. Saving Ladder Diagrams in HEX Format

After the simulation is successfully run and the validation of the control logic is completed, the ladder diagram is saved by compiling it through the "compile" menu. This process produces a file in . HEX, which is compatible with Arduino Uno. This HEX file is a digital representation of a pre-designed ladder diagram, and is used as input for uploader software such as XLoader.

Result Software

Testing of the XLoader software was carried out to ensure that it was able to upload the compiled ladder diagram program to the Arduino Uno effectively. This process is a critical step in integrating the designed control logic into the hardware. Testing is carried out with the following systematic steps:

1. Connecting the Arduino Uno to a Computer

INFOKUM
Volume 13, Number 04, 2025, DOI 10.58471/infokum.v13i03
ESSN 2722-4635 (Online)
https://infor.seaninstitute.org/index.php/infokum

The first step in testing is to connect the Arduino Uno to a Personal Computer (PC) using a USB cable. This connection allows data transfer between the PC and the microcontroller, so that compiled programs can be uploaded easily.

2. Select Program Files in Format. HEX

   After a successful connection, the precompiled ladder diagram file is converted into a . HEX is selected through the "open file" option on the XLoader interface. Format. HEX is a digital representation of the control logic that is designed, so that it can be directly translated by the microcontroller.

3. Adjusting Device Settings

   In this step, the type of microcontroller device is adjusted by selecting Arduino Uno as the type of microcontroller that will receive the program. Once the setup is complete, the program upload process begins by pressing the upload button.

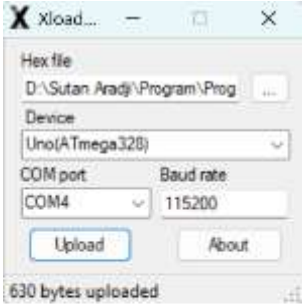

Figure 6. Upload programs with XLoader

4. Verify the Upload Process

   After the upload process is complete, XLoader provides a notification in the form of information on the number of bytes of data that has been successfully uploaded (example: "630 bytes uploaded"). This message indicates that the ladder diagram program has been successfully uploaded to the Arduino Uno microcontroller, so the system is ready to be implemented on the hardware.

Hardware Testing

Hardware testing to ensure that the input and output modules on the Arduino Uno-based PLC function according to the control logic that has been designed in the ladder diagram using LDmicro. This test is carried out systematically by connecting the hardware according to the predetermined configuration, then validating its function through the implementation of the program uploaded to the Arduino Uno. Here are the testing steps performed:

a. Input and Output Connections According to Ladder Diagram Configuration

   After the program is successfully uploaded to the PLC, the first step is to connect the input and output devices on the Arduino Uno according to the configuration that has been specified in the ladder diagram.

b. Input Connection on Module

   The PLC inputs Pin 23 (PC0) and Pin 24 (PC1) on the Arduino Uno, which are configured as inputs in the ladder diagram, are connected to the normally open (NO) terminals for Pin 23 (PC0) and normally close (NC) for Pin 24 (PC1) on the push

INFOKUM
Volume 13, Number 04, 2025, DOI 10.58471/infokum.v13i03
ESSN 2722-4635 (Online)
https://infor.seaninstitute.org/index.php/infokum

button switch. Meanwhile, the common terminal (COM) on the switch is connected to ground (GND) on the Arduino Uno. This configuration ensures that the switch can serve as an input logic trigger for the control system.

c. Output Connection on PLC Output Module
Each output of the ladder diagram is connected to a digital pin on the Arduino Uno according to the configuration L1: Pin 13 (PD7), L2: Pin 12 (PD6), L3: Pin 11 (PD5), L4: Pin 6 (PD4), L5: Pin 5 (PD3), L6: Pin 4 (PD2), L7: Pin 3 (PD1), L8: Pin 2 (PD0). Each Output Pin on the Arduino is connected to a *5 Volt Module Relay Board*. This module relay is also connected as a 24 Volt magnetic relay drive to be connected with larger loads. This connection uses jumper cables to ensure that the signal from the Areduino can be accurately forwarded to the output device.

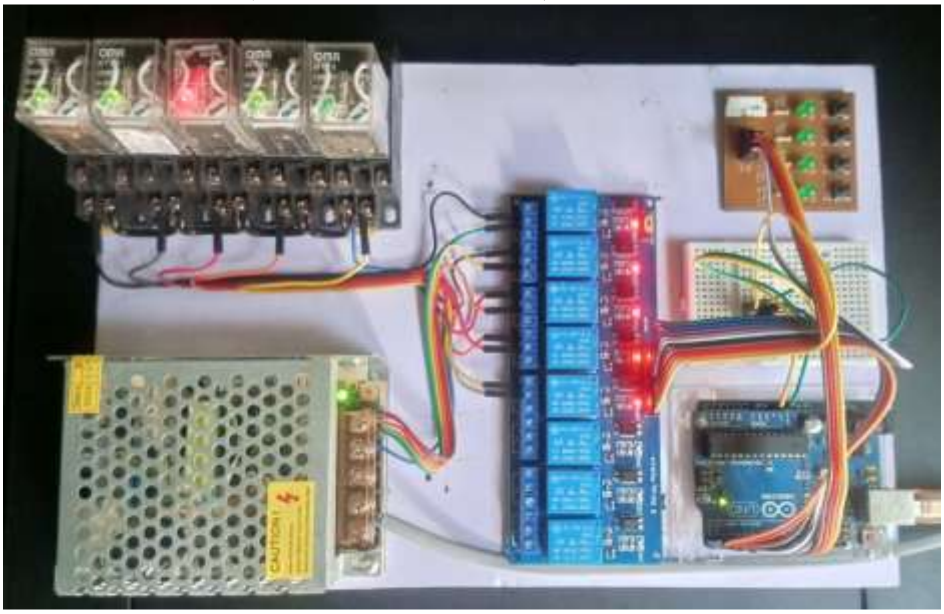

Figure 7. Testing of Magnetic Relay control system

Table 2. Identification in testing

No.	Component	Condition		The Pin I/O	Information
1	*Push Button*	*ON*	*OFF*	Pin 23 (PC0)	*Input*
2	*Push Button*	*OFF*	*ON*	Pin 24 (PC1)	*Input*
3	*Relay Magnetic*	*ON*	*OFF*	Pin 13 (PD7)	*Out Put*
4	*Relay Magnetic*	*ON*	*OFF*	Pin 12 (PD6)	*Out Put*
5	*Relay Magnetic*	*ON*	*OFF*	Pin 11 (PD5)	*Out Put*
6	*Relay Magnetic*	*ON*	*OFF*	Pin 6 (PD4)	*Out Put*
7	*Relay Magnetic*	*ON*	*OFF*	Pin 5 (PD3)	*Out Put*
8	*Relay Magnetic*	*ON*	*OFF*	Pin 4 (PD2)	*Out Put*
9	*Relay Magnetic*	*ON*	*OFF*	Pin 3 (PD1)	*Out Put*
10	*Relay Magnetic*	*ON*	*OFF*	Pin 2 (PD0)	*Out Put*

CONCLUSION

The test results show that the Arduino Uno and LDmicro-based control systems are able to function well in controlling the magnetic relay. Programming with LDmicro provides a simple and efficient solution for developing control logic on Arduino Uno-based control systems. In this study, LDmicro was used to design and implement the control logic of the magnetic relay system with a ladder diagram approach. The implementation of Arduino Uno-based PLC with LDmicro programming is an effective and economical solution for magnetic relay control systems. This approach is well-suited for use in education, prototyping, and small industrial applications, providing an affordable alternative to conventional automation solutions.

REFERENCE

Agbomemewa, L., Foletti, M., Corradini, F., Confalonieri, M., & Pedrazzoli, P. (2024). Low-cost automation solutions for startups: a distributed logic control and low-cost hardware approach. *IET Conference Proceedings*, *2024*(11), 153–158. https://doi.org/10.1049/icp.2024.3500

Anugrah, D. (2016). Rancang Bangun Pengukur Laju Detak Jantung Berbasis PLC Mikro. *Elinvo (Electronics, Informatics, and Vocational Education)*, *1*(3), 163–170. https://doi.org/10.21831/ELINVO.V1I3.10857

Aryza, S., Lubis, Z., & Annisa Lubis, S. (2020). Penguatan Industri 4.0 Berbasiskan Arduino Uno Dan GSM SIM900A DiDalam Pintu Geser. In *Journal of Electrical Technology* (Vol. 5, Issue 2)..

Aryza, S., Efendi, S., & Sihombing, P. (2024). A ROBUST OPTIMIZATION TO DYNAMIC SUPPLIER DECISIONS AND SUPPLY ALLOCATION PROBLEMS IN THE MULTI-RETAIL INDUSTRY. *Eastern-European Journal of Enterprise Technologies*, (3).

Dani, A., & Erivianto, D. (2023). Studi Perbandingan Arus Start Motor Induksi Sistem Pengasutan Dol Dan Star Delta Menggunakan Automation Studio. *Jurnal Indonesia Sosial Teknologi*, *4*(4), 413. https://doi.org/10.59141/JIST.V4I4.602

Dani, A., & Erivianto, D. (2024). Evaluasi Kinerja Motor Induksi Tiga Fasa dengan Modifikasi Resistansi Stator: Pendekatan Simulasi dengan Automation Studio. *Innovative: Journal Of Social Science Research*, *4*(3), 17604–17618. https://doi.org/10.31004/INNOVATIVE.V4I3.12650

Hidayati, Q., Zaini Rachman, F., Yanti, N., & Jamal, N. (2017). *Desain Model dan Simulasi PLC-Mikrokontroler sebagai Modul Pembelajaran Berbasis PLC*. *2*(2).

Khan, H. K., Humayun, M., & Khan, K. (2020). Design and Implementation of Transformer Protection Scheme Based on Micro PLC. *Journal*, *5*(1), 28–36. https://www.researchgate.net/publication/358964630

Masco, J. F. P. (2022). Development and Validation of a Less Expensive and Portable PLC Module for Students Training in Industrial Automation. *Electronics*, *26*(2), 39–45. https://doi.org/10.53314/ELS2226039P

Minchala, L. I., Peralta, J., Mata-Quevedo, P., & Rojas, J. (2020). An approach to industrial automation based on low-cost embedded platforms and open software. *Applied Sciences (Switzerland)*, *10*(14). https://doi.org/10.3390/app10144696

Sarifuddin, Aulia Rahman, M., Andhika Suryo Arisanto, S., & Rizal Wahyudi, M. (2021). MINIATUR SPBU DENGAN KENDALI HMI (HUMAN MACHINE INTERFACE) BERBASIS PLC (PROGRAMMABLE LOGIC CONTROL). *Print) Jurnal POROS TEKNIK*, *13*(1), 42–51.

Sinaga, Y. A., Aryza, S., & Darma Tarigan, A. (2024). An Analysis Of Aircraft Exterior Lightin Indicator Control System Based On Arduino Atmega 256. *Jurnal Scientia*, *13*(04), 1393–1402. https://doi.org/10.58471/scientia.v13i04

Suyetno, A. (2022). TEKNO Jurnal Teknologi Elektro dan Kejuruan Pengembangan Palang Pintu Perlintasan Kereta Api dengan Kendali Remot dan Sensor Getar dengan LDmicro. In *Jurusan Teknik Elektro* (Vol. 32). http://journal2.um.ac.id/index.php/tekno

Wardhana, F., Pradana, S., & Karim, K. (2021). Pemrograman Sistem Arduino Nano dan Arduino Mega Menggunakan Ladder Logic. *PoliGrid*, *2*(2), 35. https://doi.org/10.46964/poligrid.v2i2.704