iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** III **Month of publication:** March 2025

DOI: https://doi.org/10.22214/ijraset.2025.66974

PLC Development with Arduino Uno

Jogi Naidu Kuriti[1], Suneel Yarnagula[2], Leekesh Kumar[3], Tharun Gurana[4], Taj Kumari Kanala[5], Perumal Srivaths Karanam[6]
Centurion University of Technology and Management, Vizianagaram

***Abstract:** This paper explores the use of the Arduino Uno microcontroller as a cost-effective and flexible alternative to traditional Programmable Logic Controllers (PLCs) for small-scale industrial automation. It outlines the core concepts of PLCs and examines the Arduino Uno's suitability, emphasizing its digital and analog I/O capabilities, communication interfaces, and interrupt handling. The study demonstrates programming the Arduino Uno with PLC-like functions using the Arduino IDE and custom libraries. Practical examples, such as motor control and sensor integration, are provided. Performance and scalability are evaluated, showing the Arduino Uno as a viable PLC solution for education, prototyping, and small-scale automation. The paper concludes with potential enhancements and future applications for the Arduino-based PLC.*
***Keywords:** Programmable Logic Controllers (PLCs), Industrial automation, communication interfaces, Arduino Uno*

I. INTRODUCTION

In the ever-evolving field of industrial automation, the integration of innovative technologies is crucial for optimizing control systems and enhancing operational efficiency. The project, titled "PLC using Arduino Uno," explores the potential synergy between traditional Programmable Logic Controllers (PLCs) and the versatile Arduino Uno platform. This initiative aims to combine the robustness and reliability of PLCs, well-regarded in industrial environments, with the flexibility and programmability of the Arduino Uno. The objective is to create an integrated system that leverages the strengths of both technologies, offering a solution adaptable to various automation needs. By undertaking this project, we aim to bridge the gap between conventional industrial control systems and the dynamic field of embedded electronics, contributing to the development of smarter and more efficient automation solutions. In a time marked by rapid technological progress, the "PLC using Arduino Uno" project stands as a blend of advanced automation technologies. This endeavor combines the dependable nature of PLCs with the adaptability of the Arduino Uno microcontroller, envisioning a union of reliability and flexibility. The backdrop of this project is the ongoing demand for automation systems that adhere to industry standards while being agile enough to meet the diverse and changing needs of modern applications. This project explores how the traditional capabilities of PLCs can be enhanced by the innovation found in the Arduino Uno, fostering a significant shift in industrial control and automation. This introduction sets the stage for a thorough examination of the project's objectives, methodologies, and outcomes, highlighting its potential impact on embedded systems and industrial automation. Additionally, it considers the cost-effectiveness and scalability of this integrated approach, aiming to provide insights into practical implementations and future advancements in the automation industry.

II. BACKGROUND

Programmable Logic Controllers (PLCs) have been pivotal in transforming industrial automation, revolutionizing control system operations. Emerging from the need for adaptable and flexible control in manufacturing, PLCs have evolved into sophisticated devices that streamline various automation tasks. Their history dates back to the late 1960s, and since then, PLCs have become integral to industrial applications, offering unmatched reliability and efficiency. This section examines the historical development and evolution of PLCs, emphasizing their transformative impact on industries and setting the stage for exploring a modern application: integrating a PLC with the versatile Arduino Uno platform. A Programmable Logic Controller (PLC) is a specialized digital computer used in industrial control systems to automate processes. Designed to withstand harsh environments, PLCs are programmed to execute a wide range of tasks, including controlling machinery, processes, and other automated systems. They are known for their robustness, flexibility, and ability to handle complex control functions reliably.

A. Historical Overview of PLC Technology

The development of PLC technology began in the late 1960s, driven by the need for automated control in industrial settings and the limitations of traditional relay-based systems. Before PLCs, industries relied on complex relay systems for automation, which were cumbersome and difficult to modify. The introduction of the first PLC by Bedford Associates in 1969 marked a significant milestone. This innovation provided a more flexible and programmable solution, replacing hard-wired relay systems with software-based control.

Over the years, PLCs have undergone substantial advancements in size, functionality, and integration capabilities. Early PLCs were relatively large and limited in functionality compared to modern versions. Today's PLCs are compact, powerful, and capable of integrating with a wide range of sensors and actuators. They support advanced communication protocols, enabling seamless integration into complex industrial networks. The evolution of PLCs has also seen improvements in programming languages and user interfaces, making them more accessible and easier to use.

The continuous development of PLC technology has expanded its application across various industries, from manufacturing and processing to energy management and infrastructure. Modern PLCs offer enhanced processing power, increased memory, and greater connectivity, supporting the demands of Industry 4.0 and the Internet of Things (IoT).

In this context, the integration of PLCs with the Arduino Uno platform represents a forward-looking approach to industrial automation. Combining the robustness of traditional PLCs with the flexibility and programmability of Arduino, this project aims to create a versatile and cost-effective solution for contemporary automation challenges. By exploring this integration, we aim to bridge the gap between conventional industrial control systems and the innovative capabilities of embedded electronics, fostering the development of smarter and more efficient automation solutions.

III. METHODOLOGY

The methodology for developing a Programmable Logic Controller (PLC) using the Arduino Uno involves several key steps, each designed to leverage the unique capabilities of both technologies. The process is divided into hardware setup, software development, and testing phases to ensure a comprehensive and robust solution.

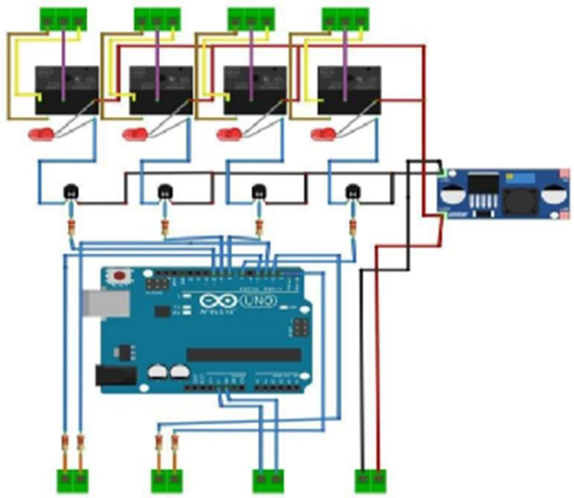
A. Hardware Setup

1) *Component Selection:* Begin by selecting the necessary hardware components, Arduino Uno will serve as the central processing unit. Select appropriate input devices such as switches, sensors, and push buttons, along with output devices like relays, motors, and LEDs. Choose a suitable power supply to ensure all components receive the required voltage and current.

Here are the specifications of the Arduino Uno

Microcontroller	ATmega328
Clock Speed	16MHz
Operating Voltage	5V
Maximum supply Voltage	20V
Supply Voltage	7-12V
Analog Input Pins	6
Digital Input/Output Pins	14
DC Current per Input/Output Pin	40mA
DC Current in 3.3V Pin	50mA
SRAM	2KB
EEPROM	1 KB
Flash Memory	32KB

2) *Wiring and Assembly:* The circuit diagram demonstrates the schematic interconnections that need to be established between the various hardware components of the PLC system described above, in order to implement the required load switching functionality.Connect the input devices (such as switches and sensors) and output devices (such as relays, motors, and LEDs) to the Arduino Uno. Ensure proper power supply connections and secure all components on a stable platform, such as a breadboard or custom PCB.



Circuit diagram of PLC

B. Software Development

1) Programming Environment

Use the Arduino Integrated Development Environment (IDE) for coding the Arduino Uno. The IDE is intuitive and facilitates writing, compiling, and uploading programs to the microcontroller. It supports numerous libraries that are crucial for implementing PLC functions, such as handling I/O operations, timing, and communication protocols. The IDE's straightforward interface and extensive documentation make it accessible for both beginners and experienced developers, streamlining the development process and ensuring efficient code management for the PLC system. Here`s the code for the PLC Using Arduino Uno consisting 4 Inputs and 4 Outputs

PLC Using Arduino Uno 4-Inputs And 4-Outputs

int pbuttonPin1 = 2; // connect output to push button 1 int pbuttonPin2 = 3; // connect output to push button 2 int pbuttonPin3 = 4; // connect output to push button 3 int pbuttonPin4 = 6; // connect output to push button 4
int relayPin1 = 7; // Connected to relay 1 (LED) int relayPin2 = 8; // Connected to relay 2 (LED) int relayPin3 = 9; // Connected to relay 3 (LED) int relayPin4 = 10; // Connected to relay 4 (LED)
int val1 = 0; // push value from pin 2 int val2 = 0; // push value from pin 3 int val3 = 0; // push value from pin 4 int val4 = 0; / / push value from pin 6
int lightON1 = 0; // light 1 status int lightON2 = 0; // light 2 status int lightON3 = 0; // light 3 status int lightON4 = 0; // light 4 status
int pushed1 = 0; // push 1 status int pushed2 = 0; // push 2 status int pushed3 = 0; // push 3 status int pushed4 = 0; // push 4 status
void setup() {
Serial.begin(9600); pinMode(pbuttonPin1,INPUT_PULLUP); pinMode(pbuttonPin2,INPUT_PULLUP); pinMode(pbuttonPin3,INPUT_PULLUP); pinMode(pbuttonPin4,INPUT_PULLUP);

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

```
pinMode(relayPin1,OUTPUT);
pinMode(relayPin2,OUTPUT);
pinMode(relayPin3,OUTPUT);
pinMode(relayPin4, OUTPUT);
}
void loop() {
val1 = digitalRead(pbuttonPin1); // read push button 1 value val2 = digitalRead(pbuttonPin2); // read push button 2 value val3 = digitalRead(pbuttonPin3); // read push button 3 value val4 = digitalRead(pbuttonPin4); // read push button 4 value
// Start of Relay 1
if (val1 == HIGH && lightON1 == LOW) {
    pushed1 = 1 - pushed1;
    delay(10);
}
lightON1 = val1;
if (pushed1 == HIGH) {
    Serial.println("Relay 1 ON");
    digitalWrite(relayPin1, LOW);
} else {
    Serial.println("Relay 1 OFF");
    digitalWrite(relayPin1, HIGH);
    }
 // End of Relay 1
// Start of Relay 2

if (val2 == HIGH && lightON2 == LOW) {
    pushed2 = 1 - pushed2;
     delay(10);
}
lightON2 = val2;
if (pushed2 == HIGH) {
    Serial.println("Relay 2 ON");
    digitalWrite(relayPin2,LOW);
} else {
    Serial.println("Relay 2 OFF");
    digitalWrite(relayPin2, HIGH);
}
    // End of Relay 2
 // Start of Relay 3

if (val3 == HIGH && lightON3 == LOW) {
    pushed3 = 1 - pushed3;
    delay(10);
}
lightON3 = val3;
if (pushed3 == HIGH) {
    Serial.println("Relay 3 ON");
    digitalWrite(relayPin3, LOW);
} else {
    Serial.println("Relay 3 OFF");
    digitalWrite(relayPin3, HIGH);
```

```
}
 // End of Relay 3
// Start of Relay 4

if (val4 == HIGH && lightON4 == LOW) {
    pushed4 = 1 - pushed4;
    delay(10);
}
lightON4 = val4;
if (pushed4 == HIGH) {
    Serial.println("Relay 4 ON");
    digitalWrite(relayPin4, LOW);
} else {
    Serial.println("Relay 4 OFF");
    digitalWrite(relayPin4, HIGH);
    delay(10);
    }
}
 // End of Relay 4
```

IV. TESTING & RESULTS

Once the hardware connections are completed and Arduino program is uploaded, thorough testing needs to be performed to verify the correct working of the PLC system.
The PLC system developed using the Arduino Uno is expected to demonstrate reliable and efficient control over various industrial automation tasks. It should effectively handle inputs from sensors and switches, process these inputs according to programmed logic, and control outputs such as motors, relays, and actuators with precision. The system should exhibit robust performance in real-time scenarios, maintaining stable operations even in demanding industrial environments. Additionally, the Arduino-based PLC should offer flexibility in reprogramming and customization, allowing for easy modifications to adapt to different automation needs. The user interface, whether via serial communication, LCD display, or web-based control, should provide clear, real-time feedback on system status and operational metrics. Overall, the system is expected to be a cost-effective alternative to traditional PLCs, delivering comparable functionality with added versatility and ease of use. The successful implementation and testing of this PLC system should validate its practical applicability in small to medium-sized industrial automation projects.

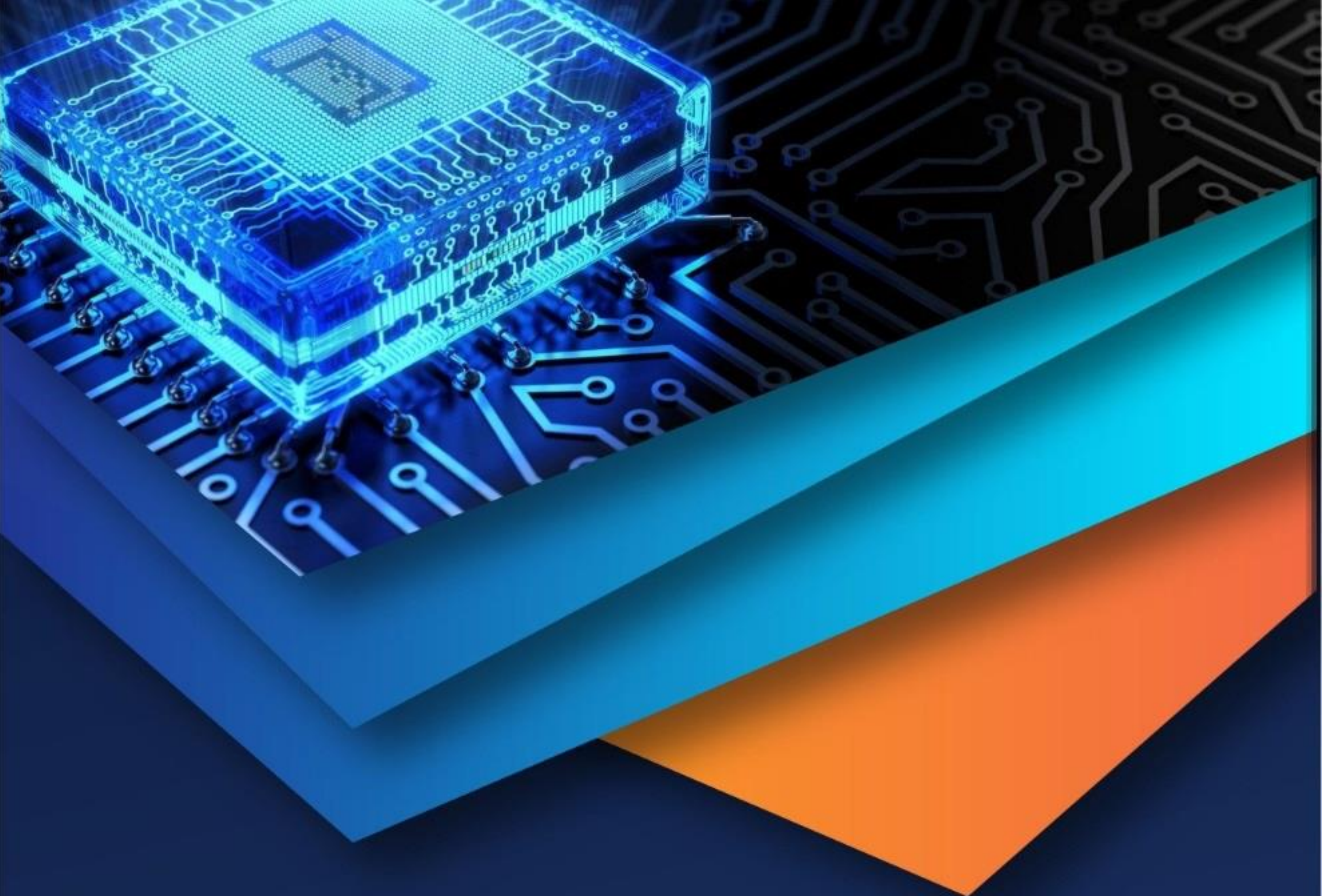
V. FUTURE WORK

Future work on the Arduino Uno-based PLC system can focus on enhancing its capabilities and expanding its applications. One extension could involve integrating advanced communication protocols such as Modbus, CAN, or Ethernet to enable seamless connectivity with other industrial devices and systems, facilitating more complex and distributed automation tasks. Another area for improvement is increasing the system's I/O capacity by adding expansion modules, thus broadening its applicability to larger and more intricate automation projects. Additionally, developing a more sophisticated and user-friendly programming interface, possibly through graphical programming tools, can make the system more accessible to users with limited coding experience. Future work could also explore incorporating advanced features like real-time data logging, remote monitoring, and control via mobile apps or cloud platforms.
Overall, these extensions aim to enhance the functionality, scalability, and usability of the Arduino-based PLC system, making it a more versatile and powerful solution for modern industrial automation challenges.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.
[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY