**Industrial Automation (Programmable Logic Controller)**

**Lovely Professional University**

**A Training Report**

Submitted in partial fulfillment of the requirements for the award of degree of

**B. Tech ECE**

**(IOT)**

**Submitted to**

**Lovely Professional University**

**Phagwara, Punjab.**



**From 05/06/2024 to 17/07/2024**

**Submitted by**

Name of the student: Shaik Mohammad Amzad

Registration number: 12207131

Signature of the student:

**To whom so ever it may concern**

I, **Shaik Mohammad Amzad, 12207131**, hereby declare that the work done by me on “**Industrial Automation (Programmable Logic Controller)**” from **June, 2024** to **July, 2024**, is a record of original work for the partial fulfillment of the requirements for the award of degree, **B. Tech ECE**.

**Name of the Student (Registration Number)**

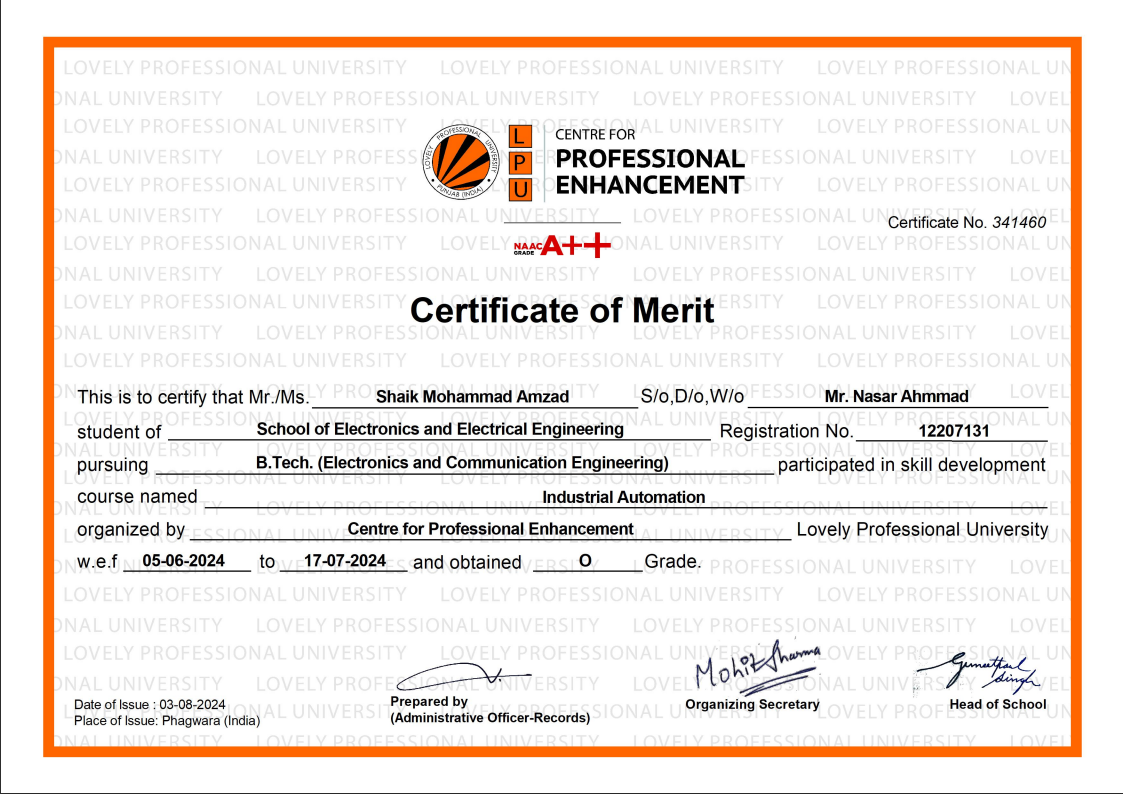
Shaik Mohammad Amzad (12207131)

**Signature of the student**

**Dated:**

**17/07/2024**

# **Training certificate from organization/ Company**

****

# **Acknowledgement**

I extend my sincere gratitude to Lovely Professional University, Department of Electronics and Communication Engineering, for providing me with the opportunity to undertake a summer internship and for fostering an environment that encouraged independent project-based learning.

I am deeply indebted to Dr. Harpreet Singh Bedi (17377) for his exceptional mentorship and unwavering support throughout my internship. His unique teaching approach, emphasizing real-world applications and hands-on experience, was instrumental in inspiring the development of my automatic visitor guided vehicle project. Dr. Bedi's guidance in utilizing Logix Pro software and his encouragement to parallel simulation and practical implementation were invaluable to the project's success.

I would also like to express my heartfelt thanks to my family for their constant encouragement and support throughout my academic journey. Their belief in my abilities has been a constant source of motivation. Additionally, I am grateful to my friends for their companionship and assistance during challenging times.

**Name:** Shaik Mohammad Amzad

**Regd. No:** 12207131

**Degree:** B. Tech ECE

**Internship Period:** 05/06/2024 – 17/07/2024

**List of contents**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Title** | **Page Number** |
| **1.** | **Declaration by student** | **1** |
| **2.** | **Training Certificate from organization** | **2** |
| **3.** | **Acknowledgement** | **3** |
| **4.** | **List of contents** | **4** |
| **5.** | **List of Figures** | **5** |
| **6.** | **List of Abbreviations** | **6** |
| **7.** | **Chapter-1 Industrial Automation** | **7-11** |
| **8.** | **Chapter-2 PLC (Programmable Logic Controller)** | **12-15** |
| **9.** | **Chapter-3 Programming in PLC** | **16-23** |
| **10.** | **Chapter-4 Timers in PLC Programming** | **24-26** |
| **11.** | **Chapter-5 Counters in PLC Programming** | **27-28** |
| **12.** | **Chapter-6 Project** | **29-45** |
| **13.** | **Chapter-7 Conclusion** | **46-49** |
| **14.** | **Chapter-8 References** | **50** |

**List of Figures:**

1. **Chapter-1**
2. Figure 1.1 Early Water Wheels
3. Figure 1.2 Early Windmills
4. Figure 1.3 Early manufacturing factories
5. Figure 1.4 Steam Engines for Locomotive
6. Figure 1.5 Early Control Stations
7. Figure 1.6 Robotics in manufacturing factories
8. **Chapter-2**
9. Figure 2.1 PLC (Programmable Logic Controller)
10. Figure 2.2 PLC Communication
11. **Chapter-3**
12. Figure 3.1 Rung and Rail in PLC
13. Figure 3.2 Ladder diagram for Start/Stop Circuit
14. Figure 3.3 NOT Gate in PLC
15. Figure 3.4 AND Gate in PLC
16. Figure 3.5 OR Gate in PLC
17. Figure 3.6 NAND Gate in PLC
18. Figure 3.7 NOR Gate in PLC
19. Figure 3.8 EX-OR Gate in PLC
20. Figure 3.9 EX-NOR Gate in PLC
21. **Chapter-4**
22. Figure 4.1 On Delay Timer
23. Figure 4.2 Off Delay Timer
24. Figure 4.3 Pulse Timer
25. **Chapter-5**
26. Figure 5.1 Up-Counter
27. Figure 5.2 Down-Counter
28. **Chapter-6**
29. Figure 6.1 Illustration of Automatic Visitor Guided Vehicle
30. Figure 6.2 Obstacle Detection by AVGV
31. Figure 6.3 Connections between the output devices and sensors
32. Figure 6.4 Ladder diagram of AVGV
33. **Chapter-7**
34. **Chapter-8**

**List of Abbreviations:**

* **PLC- Programmable Logic Controller**
* **ACC - Accumulator**
* **PRE - Preset Value**
* **LD- Ladder Diagram**
* **FBD- Function Block Diagram**
* **ST- Structured Text**
* **IL- Instruction List**
* **SFC- Sequential Function Chart**
* **EX-OR- Exclusive OR gate**
* **TON- On Delay Timer**
* **TOF- Off Delay Timer**
* **TP- Pulse Timer**
* **CTU- Up Counter**
* **CTD- Down Counter**
* **AVGV- Automatic Visitor Guided Vehicle**

**Chapter 1: Industrial Automation**

The word ‘Automation’ is derived from Greek words “Auto” (self) and “Matos” (moving). Automation therefore is the mechanism for systems that “move by itself”. However, apart from this original sense of the word, automated systems also achieve significantly superior performance than what is possible with manual systems, in terms of power, precision and speed of operation.

**Definition:** Automation is a set of technologies that results in operation of machines and systems without significant human intervention and achieves performance superior to manual operation.

## **The Evolution of Automation**

Since the beginning, everyday people and business owners have always looked for different ways to improve efficiency in day-to-day tasks. In ancient times, our ancestors did most of their work manually with the assistance of simple machines, pulleys, levers, hand tools, and sometimes working animals. From a manufacturing standpoint, production processes are typically categorized as:

1. Fully manual
2. Semi-automated
3. Fully automated

By reducing the amount of human intervention required to perform a task, processes such as manufacturing become much more efficient through automation. The history of automation’s timeline covers how technology has advanced through the evolution of automation.

### **1st Century BC: Water Wheels**

The true date of origin for water wheels is difficult to confirm, however, they became more common use by the Greeks and Romans for grinding grain into flour around the 1st century BC. Other simpler machines which date back much earlier still required a fair amount of human intervention or animal labor to work. The utilization of falling water from watermills to drive a mechanical process can be viewed as the beginning of “semi-automation” and the early stages of automation evolution.



Figure 1.1 Early Water Wheels

### **9th Century: Mill Machinery Advancements**

Watermills and windmills are both a type of mill machinery that uses renewable energy to drive a mechanical process. The earliest recorded design of a windmill for practical use originates around the 7-9th century and was made by the Persians. These windmills were also used to grind grain and then later developed for many other mechanical processes.

Although watermills can offer more power for its size, windmills were usable in regions which had no flowing water available. Both technologies continued improving and were being adopted across the globe to reduce the manual labor required in:

1. Hammermills
2. Sawmills
3. Paper mills
4. Ore-crushing mills
5. Tool-sharpening mills

Figure 1.2 Early Windmills

### **17th to 18th Century: Industrial Revolution**

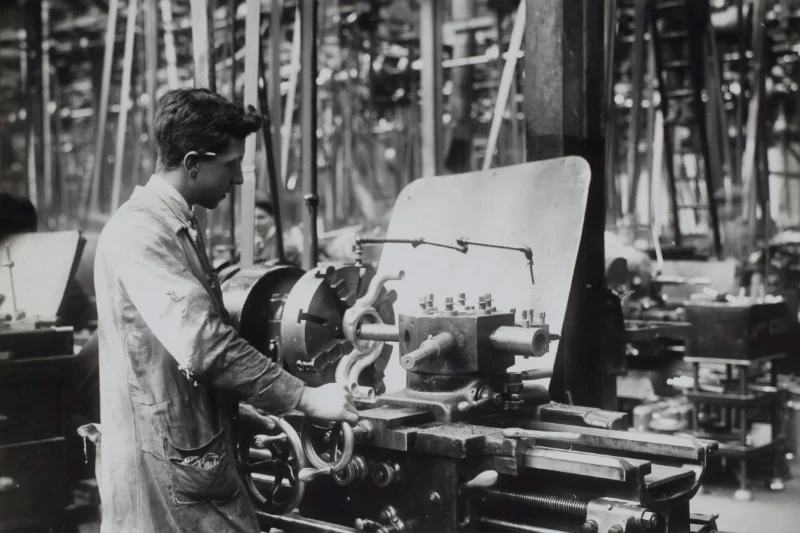
****Originating in Western Europe, the 17th century industrial revolution was a major turning point in the evolution of industrial automation. During this era, the invention of steam engines, steam mills and internal combustion engines had mostly replaced the need for watermills and windmills. In 1785, Oliver Evans had also developed an automatic flour mill which was history’s first completely automated industrial process by being able to have continuous production without any human intervention.

Figure 1.3Early manufacturing factories

In 1867, James Clerk Maxwell published a paper establishing the beginnings of a theoretical basis for understanding control theory. Industrialized factories continued being adopted to mass produce materials such as cotton, paper, plastics, glass, and metals in much higher volumes with greater efficiency. As the technology and processes of the industrial revolution had spread across the globe, the economy, transportation, health, and medicines worldwide were all growing exponentially.

Figure 1.4 Steam Engines for Locomotive

### **1900 to 1950s: Electrification & Industrial Controllers**

Around the 1920s, the evolution of industrial automation accelerated rapidly as factories began making use of relay logic and underwent electrification - the process of powering by electricity. Expanding use of central electric power stations combined with the operation of new high-pressure boilers, electrical substations and steam turbines resulted in a growing demand for instruments and controls.

Figure 1.5 Early Control Stations

Color-coded lights from control rooms were required to send signals for factory workers to make manual changes such as opening or closing valves and turning switches on or off. This is a type of process control known as “on-off”. In the 1930s, controllers were introduced into the industry to enable calculated changes as a response to disturbances from the set point. Solid-state digital logic modules for hard-wired programmed logic controllers were being adopted by industrial control systems for process control and automation in 1958. As the predecessors of programmable logic controllers (PLC) used today, they gradually replaced most of our needs for electro-mechanical relay logic.

### **20th to 21st Century: Computers & Robotics**

In 1971, the invention of microprocessors resulted in large price drops for computer hardware and allowed the rapid growth of digital controls in the manufacturing industry. Our constant advancements in computer technology up till this day continues to advance the evolution of industrial automation. With digital computers, manufacturing facilities were now able to have controllers which can perform more complex tasks at faster speeds and greater efficiency.

As technology continued in advancements, the evolution of robotic process automation was becoming more prominent in manufacturing facilities. Victor Scheinman, an American pioneer of the robotics field had invented the “Stanford arm” in 1969. It was designed to permit an arm solution as a 6-axis articulated all-electric robot. This shaped a path for robots to have the potential of performing more complex tasks such as welding and assembly. In 1973, Europe was making huge advancements in industrial robotics by bringing robots to the market through ABB Robotics and KUKA Robotics.

The robots in today’s factories are now used for almost every existing assembly and manufacturing process. Not only do robots remove humans from hazardous environments, but they also help lower costs for business owners to stay competitive by increasing energy efficiency, productivity, accuracy, and precision for better production quality. We can now see the evolution of robotic process automation in processes such as:

1. Glass manufacturing
2. Pulp and paper mills
3. Food and beverage processing
4. Automotive assembly
5. Natural gas separation
6. Electrical power generation
7. Electronics manufacturing
8. Canning and bottling

Figure 1.6 Robotics in manufacturing factories

**Chapter 2: PLC (Programmable Logic Controller)**

Programmable Logic Controllers (PLCs) are industrial computers, with various inputs and outputs, used to control and monitor industrial equipment based on custom programming.

PLCs come in many different sizes and form factors. Some are small enough to fit in your pocket, while others are large enough to require their own heavy-duty racks to mount. Some PLCs are more modular, with only basic I/O (Inputs and Outputs), but can be customized with additional back planes and functional modules (such as analog I/O, communications modules, or display modules) to fit different types of industrial applications.

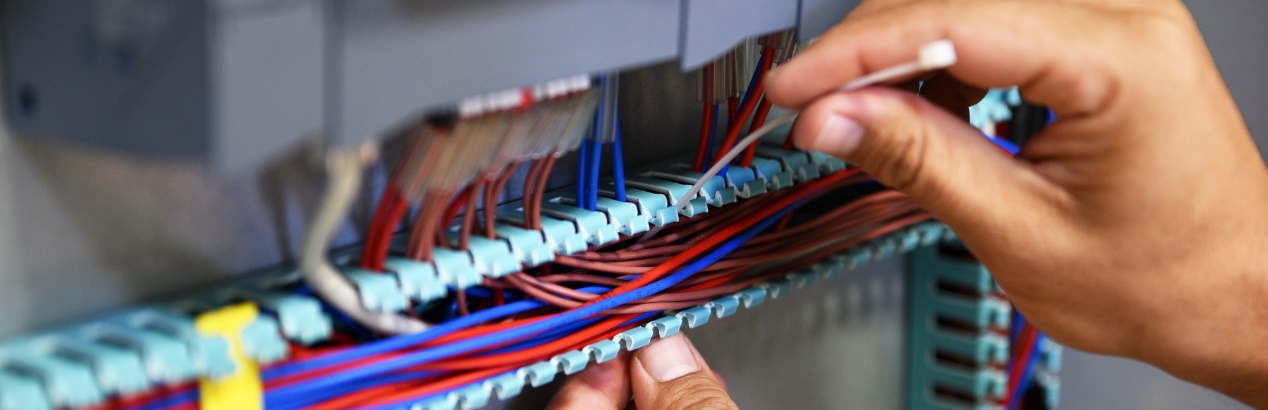


Figure 2.1 PLC (Programmable Logic Controller)

**Main Types of PLCs**

There are many kinds of PLCs but most fall into one of two main categories:

1. Fixed
2. Modular

Fixed PLCs have a pre-built number of inputs and outputs, and modular PLCs are much more adaptable. Fixed and modular PLCs are useful for different purposes, with benefits ranging from portability and affordability to scalability and customizability.

The most common type of PLC is the fixed PLC. Fixed PLCs are smaller and more affordable than modular PLCs, which makes fixed PLCs a good choice for smaller or portable control systems or for performing standalone tasks. However, fixed PLCs are made with the processing unit, terminals, and input and output components wired internally, and they do not have as much memory as modular PLCs. This makes fixed PLCs harder to repair and modify, which could lead to increased downtime if you do not have backup PLCs.

Modular PLCs are more scalable and customizable, and easier to troubleshoot than fixed PLCs, but modular PLCs are also larger and more expensive. Modular PLCs are best for expanding large-scale operations because you can easily add more input and output devices, improve processing units, upgrade memory, and more. This allows you to customize a modular PLC to increase functionality and do complex operations that would be impossible for a fixed PLC. Modular PLCs also reduce downtime because each module does a specific job, which makes it easier to isolate and fix faults while the working PLC modules continue to operate.

**5 Standard PLC Programming Languages**

1. **Ladder Diagram (LD)**
   * Resembles electrical relay logic diagrams.
   * Widely used in the automation industry.
   * Graphical representation of control logic.
2. **Function Block Diagram (FBD)**
   * Uses blocks to represent functions.
   * Easy to visualize and understand.
   * Suitable for complex control systems.
3. **Structured Text (ST)**
   * High-level textual programming language.
   * Similar to Pascal or C.
   * Suitable for complex mathematical algorithms and data handling.
4. **Instruction List (IL)**
   * Low-level textual language.
   * Resembles assembly language.
   * Efficient but less readable and being phased out in favor of ST.
5. **Sequential Function Chart (SFC)**
   * Graphical programming language.
   * Represents the sequence of operations.
   * Suitable for sequential and state-based control systems.

Each of these languages has its strengths and is suited to different types of control tasks in industrial automation.

**Working of PLC**

A PLC’s operation is broken down into three stages:

1. Inputs
2. Program Execution
3. Outputs

PLCs capture data from the plant floor by monitoring inputs from any connected machines or devices. These inputs are checked against the program logic, which changes the outputs to any connected output devices. It is possible to have the same machine connected to both inputs and outputs on the same PLC, such as a valve position sensor connected to the inputs with the control of that valve position connected to the outputs. A program could read the current position of that valve, check to see if it needs to move, then move the valve position with the output.

PLCs often make a distinction between Digital and Analog I/O. Digital I/O acts like a standard light switch where the state is either on or off, with no states between. Analog I/O acts like a dimmer switch, where the state can be anywhere between on and off.

It is easy to think of there being two sources of input data for PLCs: Device input data, automatically generated by a machine or sensor, or User input data, generated by a human operator using an HMI or SCADA system.

The Device input data comes from sensors and machines that send information to the PLC. This can include:

1. On/Off states for things like mechanical switches and buttons
2. Analog readings for things like speed, pressure, and temperature
3. Opened/Closed states for things like pumps and valves

Human-facilitated inputs can include button pushes, switches, sensors from devices like keyboards, touch screens, remotes, or card readers.

PLC outputs are very similar to inputs, but can also include audible or visual indicators for the user, such as turning on a warning light, or sounding an alarm beacon. Other outputs can include:

1. Opening or closing a valve
2. Adjusting the speed on a motor
3. Turning a heater On or Off

PLC programs operate in cycles. First, the PLC detects the state of all input devices that are connected to it. The PLC executes the user-created program, using the state of the inputs to determine the state that the outputs should be changed to. The PLC then changes the output signals to each corresponding device. After completing all these steps, the PLC then does a housekeeping step, which includes an internal diagnostic safety check to ensure that everything is within normal operating conditions. The PLC restarts the cycle each time the process is completed, starting again by checking inputs.

**Future of PLCs**

****

Figure 2.2 PLC Communication

The industry continues to see new products entering the market ranging from devices like Programmable Automation Controllers (PACs), which combine the functionality of PLCs with higher-level PC functionality, all the way to industrial embedded hardware.

Even with these new products, PLCs remain popular because of their simplicity, affordability, and usefulness. And software like Ignition will enable organizations to maximize their usefulness for many years to come.

**Chapter 3: Programming in PLC**

 **Basic Elements**

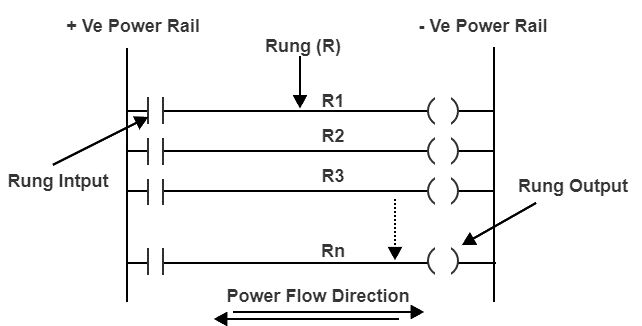
1. **Rungs:** Horizontal lines of logic.
2. **Rails:** Vertical lines of logic.
3. **Contacts:** Represent conditions (like switches or sensors).
4. **Coils:** Represent outputs (like motors or lights).
5. **Timers and Counters:** Special instructions for timing events and counting occurrences.

Figure 3.1 Rung and Rail in PLC

 **Creating a Simple Program**

* **Step 1:** Identify the control requirements.
* **Step 2:** Draw the ladder diagram with the necessary contacts and coils.
* **Step 3:** Enter the program using PLC programming software.
* **Step 4:** Test and debug the program.
* **Step 5:** Load the program into the PLC.

**Example: Simple Start/Stop Circuit**

1. **Requirement:** Control a motor using a start and stop button.
2. **Components:**
   * **Start Button (Input)**
   * **Stop Button (Input)**
   * **Motor (Output)**
3. **Ladder Diagram**

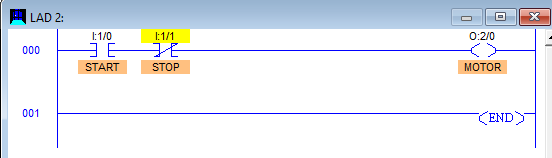
****

Figure 3.2 Ladder diagram for Start/Stop Circuit

1. When you press the start button, the NO switch closes.
2. This completes the circuit because the NC stop switch is already closed.
3. Current flows through the circuit, energizing the motor (Q0.0), and the motor starts running.
4. When you press the stop button, the NC switch opens.
5. This interrupts the current flow through the circuit.
6. The motor (Q0.0) de-energizes and stops running.

* **Logic Gates in PLC**

1. NOT GATE

In electronics, NOT GATE is also called an ‘**Inverter’** or ‘**Buffer’**.

**Working:** NOT gate works as inversion. It takes one input and gives one output. When the input is high then the output is low and vice-versa.

**Logic Gate Truth Table for NOT Gate:**

|  |  |
| --- | --- |
| **Input (I1)** | **Output (Q1)** |
| 0 | 1 |
| 1 | 0 |

**NOT Gate in PLC programming:**

In the case of PLC ladder, there will be a push button to provide input. When (I1) is pressed then the coil (Q1) is on. And when Input (I1) is released then coil (Q1) is off.

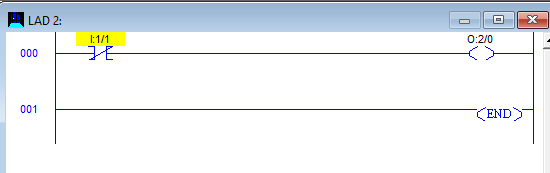


Figure 3.3 NOT Gate in PLC

**2. AND GATE**

**Working**: In AND Gate, when both inputs (I1 and I2) are high then the output (Q1) will be high.  For all other inputs, output (Q1) will be the low.

**Logic Gate Truth Table for AND Gate:**

|  |  |  |
| --- | --- | --- |
| **Input (I1)** | **Input (I2)** | **Output (Q1)** |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**AND Gate in PLC programming:**

Using Ladder diagram programming, we are connecting two switches (I1 and I2) as input and coil/lamp (Q1) as output.

In the case of both switches (I1 and I2) are closed, the lamp (Q1) will glow. In another case, if any of the switches (I1 or I2) are open then lamp (Q1) will not glow.

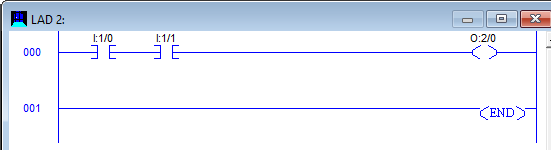


Figure 3.4 AND Gate in PLC

**3. OR GATE**

**Working:** If both inputs are low in the OR gate, then the output will be low. For all other cases, the output will be high.

**Logic Gate Truth Table for OR Gate:**

|  |  |  |
| --- | --- | --- |
| **Input (I1)** | **Input (I2)** | **Output (Q1)** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**OR Gate in PLC programming:**

In case both or anyone inputs (I1 and I2) are closed then coil (Q1) will on.

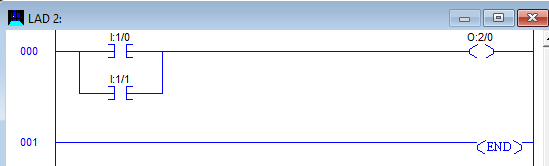
****

Figure 3.5 OR Gate in PLC

4. NAND GATE

NAND gate is operated as an AND gate and followed by the inverter.

**Working:** In NAND Gate, the output will be low when both inputs are high. For all other cases, the output will be high.

**Logic Gate Truth Table for NAND Gate:**

|  |  |  |
| --- | --- | --- |
| **Input (I1)** | **Input (I2)** | **Output (Q1)** |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**NAND Gate in PLC programming:**

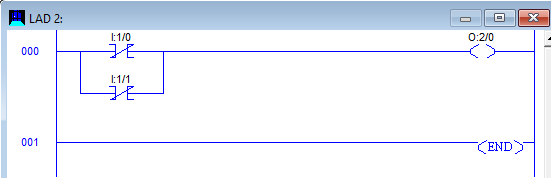
If both switches (I1 and I2) or anyone switch (I1 or I2) are closed, the lamp will be glow. In the case, both switches are open then the lamp will not be glow.

Figure 3.6 NAND Gate in PLC

5. NOR GATE

NOR Gate is operated OR Gate followed by the NOT Gate.

When both inputs are low then the output will be high. Otherwise, the low output will occur if both inputs are high.

**Logic Gate Truth Table for NOR Gate:**

|  |  |  |
| --- | --- | --- |
| **Input (I1)** | **Input (I2)** | **Output (Q1)** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

**NOR Gate in PLC programming:**

The Coil (Q1) will be activated if both inputs are closed. Coil (Q1) will be deactivated if any one or both the inputs are open.

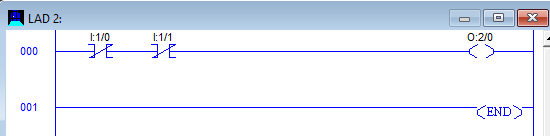


Figure 3.7 NOR Gate in PLC

6. EX-OR GATE

**Working:** If both inputs are high or low, the output will become low. For any other input condition, the output will be high.

**Logic Gate Truth Table for EX-OR Gate:**

|  |  |  |
| --- | --- | --- |
| **Input (I1)** | **Input (I2)** | **Output (Q1)** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**EX-OR Gate in PLC programming:**

In the function of EX-OR Gate, the lamp will be on if one switch is closed and another switch is opened.

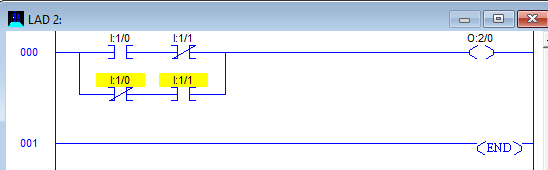


Figure 3.8 EX-OR Gate in PLC

7. EX-NOR GATE

**Working:** When both inputs (I1 and I2) are high or low then the output will high. If anyone input is high or low then the output will become low.

**Logic Gate Truth Table for EX-NOR Gate:**

|  |  |  |
| --- | --- | --- |
| **Input (I1)** | **Input (I2)** | **Output (Q1)** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**EX\_NOR gate in PLC programming:**

The function of EX-NOR Gate, the lamp (Q1) will be on if both switches (I1 and I2) are open or closed. The lamp (Q1) will not be on if anyone switches (I1) is activated and another switch (I2) is deactivated.

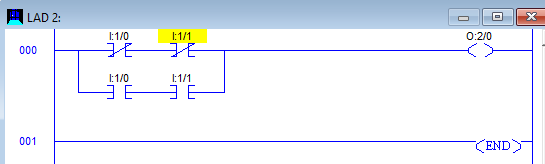


Figure 3.9 EX-NOR Gate in PLC

Programmable Logic Controllers (PLCs) have revolutionized industrial automation by providing a flexible, reliable, and efficient platform for controlling complex processes. Their user-friendly ladder logic programming interface, combined with robust hardware capabilities, enables engineers to design and implement sophisticated control systems with relative ease. The ability to handle a wide range of inputs and outputs, coupled with their inherent fault tolerance, makes PLCs indispensable in modern manufacturing and industrial operations. As technology continues to advance, PLCs are likely to become even more sophisticated, integrating seamlessly with other automation technologies and expanding their applications across various industries.

**Chapter-4 Timers in PLC Programming**

Timers are key in PLC programming, helping control everything from when a motor starts to how long a signal lasts. They might seem tricky at first, but they’re straightforward once you get the hang of them.

Different PLC platforms offer various timer functions, and many are unique to a specific platform. For instance, Siemens PLC has its own set of timer functions exclusively for their systems. It’s important to know this as the timer functions you can use might vary with the platform you choose.

Timers can be used not only in ladder logic but also as functions blocks in function block diagram or functions in structured text. They can even be used to check how long an actuator has been running and then for alarms in a SCADA system.

Three standard timer functions: -

1. On Delay Timer (TON)
2. Off Delay Timer (TOF)
3. Pulse Timer (TP)

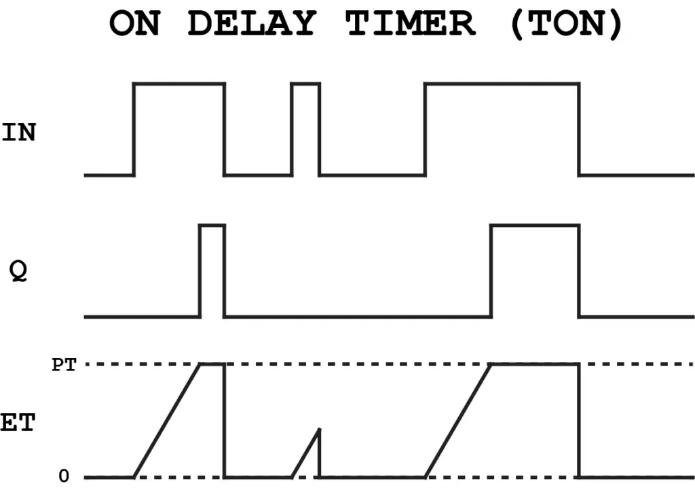
First one of the standard timers is the on delay timer also known as just TON. This is by far the most used timer in PLC programming. You will find this in any platform and it is in fact so useful that you can build the other timer functions with the on delay timer. The functionality of the on delay timer (TON) can be described like this: Output is turned ON after a delay Below here you can see the timing diagram of the on delay timer. You can see that the output is turned on after a delay.

Figure 4.1 On Delay Timer

This delay is called the preset time (PT). The delay said in another way, is how long you want the timer to be turned on. When you turn on the input (IN) the timer will start timing (turning on the timer). Elapsed time (ET) is the current time of the timer. Here you can always see how long the timer has been turned on.

As soon as you activate the timer by turning on the input the timer will start counting. After a certain delay the output will be turned on.

The second standard PLC timer is the off delay timer or just TOF. My best way to remember how it works is again by its name.

It is called an off delay timer because it works like this:

Output is turned OFF after a delay

One of the biggest differences between this and the on delay timer is how you activate it.

As soon as you turn on the input of this timer, the output is also turned on. This is because in order for the output to be turned OFF (after a delay) it needs to be turned on in the beginning.



Figure 4.2 Off Delay Timer

The timer will not be activated before you turn the input off again. When you do that the timer will start counting and after the delay, the output will be turned off.

The final one of the 3 standard timers is called the Pulse Timer or PT.

Although this timer is not so commonly used it is still a very useful timer function.

This timer is a little different than the two others, since this one is used to generate pulses. Yes, that is also how we can describe its functionality:

Generates a pulse of a specific length

You can activate the timer by turning on the input. When that happens, the timer will start counting time. As a parameter for the pulse timer the time for the pulse is defined.

After the output has been on for that amount of time it will be turned off again.



Figure 4.3 Pulse Timer

The difference from the two other timers here is that this pulse will happen no matter what the state of the input will be in the meantime. You will always get a pulse of that certain length after the timer has been activated.

If you look closer at the timing diagram you can also see that Q always has the same length.

Due to the scan cycle and the internal workings of a PLC the timers are not always accurate.

**Chapter-5 Counters in PLC Programming**

A PLC counter is a function block that counts up or down until it reaches a limit. When the limit is reached the output is set.

Counting plays a crucial role in PLC programming. It’s all about tracking numbers—like how many times a process completes or the quantity of products produced. Think of it as a digital tally counter that helps manage industrial tasks efficiently.

PLC counters are also used to assist logging to SCADA systems by counting the number of times these events have happened or setting alarms when an event has happened a certain number of times.

With all that said timers are very useful and it is crucial for every PLC programmer to know the basics of counting in a PLC program. In this article I will explain how counters work, and how you can use them in your PLC programs.

**Basics of Counting in a PLC**

Before you start counting in any PLC program there are some basics you should know first. These are basic information about the counter instructions and the PLC itself.

Understanding a PLC’s counting limits is vital. It’s like knowing the maximum number your digital counter can reach before it resets or overflows.

**Up Counter (CTU)**

The first counter instruction I will introduce you to is the up counter, also known as just CTU. As the name implies, this PLC counter is used for counting up.

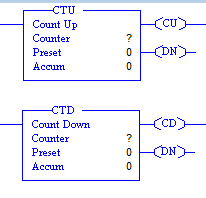
You can see the up-counter function block illustrated below:

Figure 5.1 Up-Counter

An Up Counter sets an output after counting to a specified number. It’s straightforward: each incoming pulse adds one to the count until the set limit is reached.

Each pulse on the count input (CU) will increase the current counter value (CV) by 1. When CV is greater than or equal to the counter limit (PV) the output (Q) is set. A pulse on the reset input (R) will reset the value of CV to 0.

Up counters are usually used to keep track of how many times an event has happened. Let’s say you want a process to complete 10 times before cleaning needs to happen.

For this you have to set the counter limit (PV) to 10. Each time the process has completed you will give a pulse on the count input (CU). When the process has completed 10 times, the output (Q) will be set. Now you can use that output to for example set an alarm that the system needs cleaning.

When cleaning is done you can give a pulse on the reset input (R) and you can now start over again.

**Down Counter (CTD)**

Counting down is another operation that is widely used in PLC programming. In some cases, you want to know how many counts are remaining before the limit is reached. With the up counter you can use some math to do it. But you can do it easily with a down counter.

The Down Counter works by counting backwards from a set number down to zero, offering a clear countdown for tracking remaining processes.

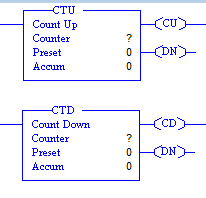


Figure 5.2 Down-Counter

As you can see the down counter has a pin called LD instead of the reset. It is called load and is used for loading a value into the current counter value. Because when you count down to 0 you will need some initial value of the counter. Each pulse on the count input (CD) will decrement current counter value (CV) by 1. When CV is less than or equal to 0 the output (Q) is set. A pulse on the load input (LD) will assign the value of counter limit (PV) to CV.

**Chapter-6 Project**

* **Automatic Visitor Guided Vehicle**

****

Figure 6.1 Illustration of Automatic Visitor Guided Vehicle

**Introduction**

Enhancing Visitor Experiences with Automatic Guided Vehicles (AGVs)

In today's world, museums, exhibitions, and large facilities are constantly seeking innovative ways to improve visitor experiences. Traditional guided tours, while informative, can be inflexible and require dedicated staff. This is where Automatic Visitor Guided Vehicles (AGVs) come in. AGVs are self-driving robots programmed to navigate a pre-defined path within a designated environment. Imagine a miniature, automated tour guide that effortlessly carries visitors through a facility, providing them with information and freedom to explore at their own pace. This project delves into the development and functionalities of an AGV simulation using LogixPro software, showcasing the potential of AGVs to revolutionize visitor experiences. Key benefits of AGVs for visitors: Self-Guided Exploration: Visitors can explore the facility independently, tailoring their tour to their interests and pace. Informative Audio Guidance: Integrated speaker systems can deliver pre-recorded messages or live announcements about points of interest. Enhanced Engagement: AGVs can add a touch of interactivity and novelty to the visitor experience. This project will explore the functionalities of an AGV simulation, including: Automated Path Following: The AGV will navigate a pre-programmed path within a simulated environment. Obstacle Detection: Sensors will detect and avoid obstacles in the path, prioritizing visitor safety. Manual Control Features: Emergency stop and technician control buttons will be implemented for safety and maintenance purposes.

**Abstract**

This system takes visitors on a self-guided tour of our facility, showing them around automatically and providing information along the way.

When the start button is pressed, the PLC activates the two motors, and the vehicle begins moving forward. It travels for 10 seconds and then stops for 5 seconds, repeating this cycle up to 8 times, creating a total of 9 stops, including the starting and ending points.

At every third stop, the vehicle takes a right turn to navigate through the predetermined path, ensuring it follows the desired route throughout the facility.

To ensure the safety of our visitors, I've integrated ultrasonic sensors into the vehicle. These sensors continuously monitor the path ahead, and if an obstacle is detected while the vehicle is in motion, the PLC program automatically stops the vehicle to avoid a collision. Once the obstacle is removed, the vehicle resumes its normal operation.

In addition to the autonomous features, I've also implemented an emergency stop button that can turn off all the output devices of the vehicle in case of an emergency or malfunction. Furthermore, I've included two manual buttons that allow our technicians to control the two motors individually, enabling them to manually move or position the vehicle as needed.

One of the key features of my project is the integration of a speaker system. As the vehicle navigates through the facility, the speaker provides verbal instructions and information to the visitors. When the vehicle reaches each stop location, the speaker delivers pre-recorded messages or real-time announcements, informing the passengers of their current location and any relevant details.

This audio guidance complements the vehicle's autonomous movement, creating a more immersive and informative experience for our visitors.

By automating the guidance process, we can improve visitor safety, reduce the workload on our staff, and provide a more consistent and efficient visitor experience. The inclusion of obstacle detection, emergency stop, manual control, and audio guidance features further enhances the system's capabilities, making it a valuable asset for our facility.

**Project Objectives**

Developing a comprehensive simulation of an AGV using LogixPro software that effectively guides visitors through a pre-programmed path within a virtual environment. **Functionality:**

Implementing automated path following control logic in the simulation, ensuring the AGV navigates the desired route accurately. Integrating obstacle detection capabilities using virtual sensors to prioritize visitor safety by stopping the AGV if obstacles appear on the path. Including manual control features like an emergency stop button for immediate shutdown and technician control buttons for maneuvering the AGV during maintenance or testing. Developing a speaker system simulation that delivers pre-recorded or real-time audio messages to visitors at designated stops, enhancing the tour experience.

**Performance:**

Optimizing the simulation's performance to ensure smooth and realistic movement of the AGV within the virtual environment. Testing and refining the AGV's behavior to guarantee reliable and consistent operation throughout the simulated tour.

**Evaluation:**

Evaluating the effectiveness of the AGV simulation in mimicking realworld AGV behavior, focusing on aspects like path following, obstacle avoidance, and visitor guidance. Analyzing the potential benefits of implementing such an AGV system in a real-world facility, considering factors like visitor experience, staff workload, and safety improvements.

**System Overview**

**1. Automated Path Following:**

The AGV navigates a pre-defined path within a simulated environment. This path represents a visitor tour route within our facility. The LogixPro program controls the AGV's movement along this path, including: Forward motion for a specified duration (10 seconds). Pauses at designated stops to allow visitors to explore points of interest (5 Seconds). Making turns (as per the route) at specific points along the path to ensure the AGV follows the desired route.

**2. Obstacle Detection:**

This functionality simulates the AGV's ability to detect obstacles in its path, prioritizing visitor safety. Virtual sensors (like ultrasonic sensors) are incorporated within the simulation. The program continuously monitors the sensor data. If an obstacle is detected while the AGV is moving, the program triggers an immediate stop to prevent a collision.



Figure 6.2 Obstacle Detection by AVGV

**3. Manual Control Features:**

These features provide control options for technicians or emergency situations. Emergency Stop Button: This button triggers a complete shutdown of all the AGV's output devices in case of emergencies or malfunctions. Technician Control Buttons: These buttons allow for manual control of each motor individually. This enables technicians to maneuver the AGV during maintenance or testing phases, facilitating setup, calibration, or troubleshooting.

**4. Speaker System Simulation:**

This feature enhances the visitor experience by providing audio guidance throughout the tour. The simulation replicates a speaker system that delivers pre-recorded messages or real-time announcements to visitors at designated stops. This audio information can include: Descriptions of points of interest along the tour route. Additional information or instructions for visitors.

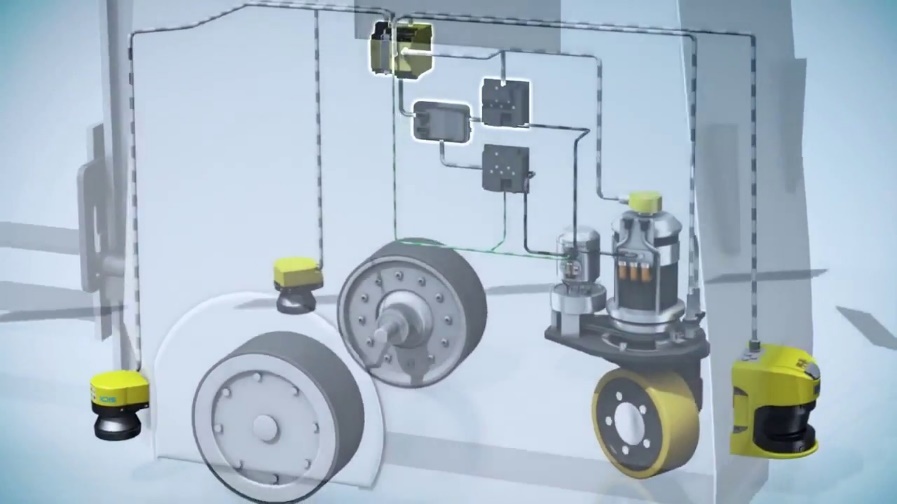


Figure 6.3 Connections between the output devices and sensors

**Ladder Diagram:**

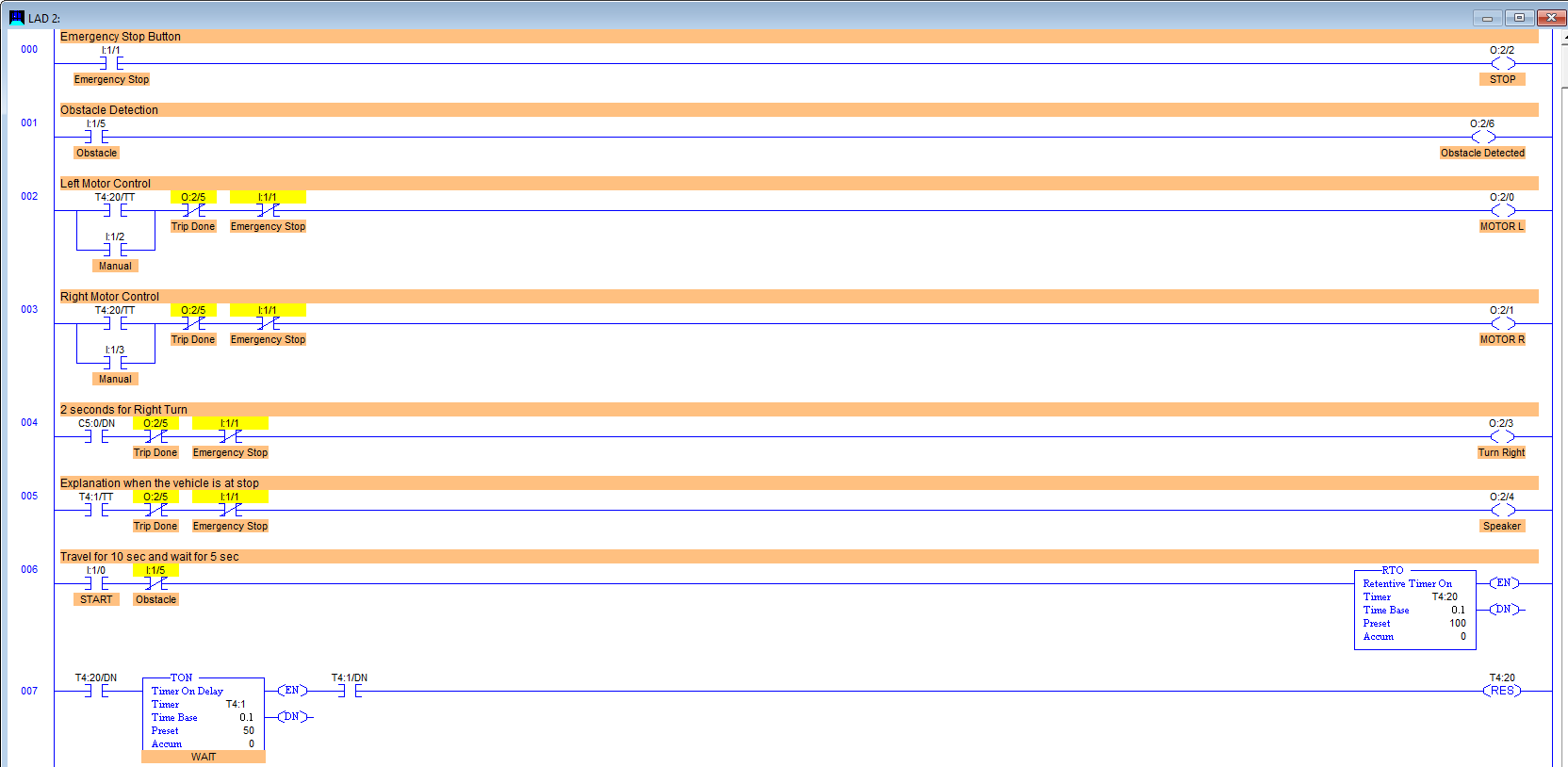
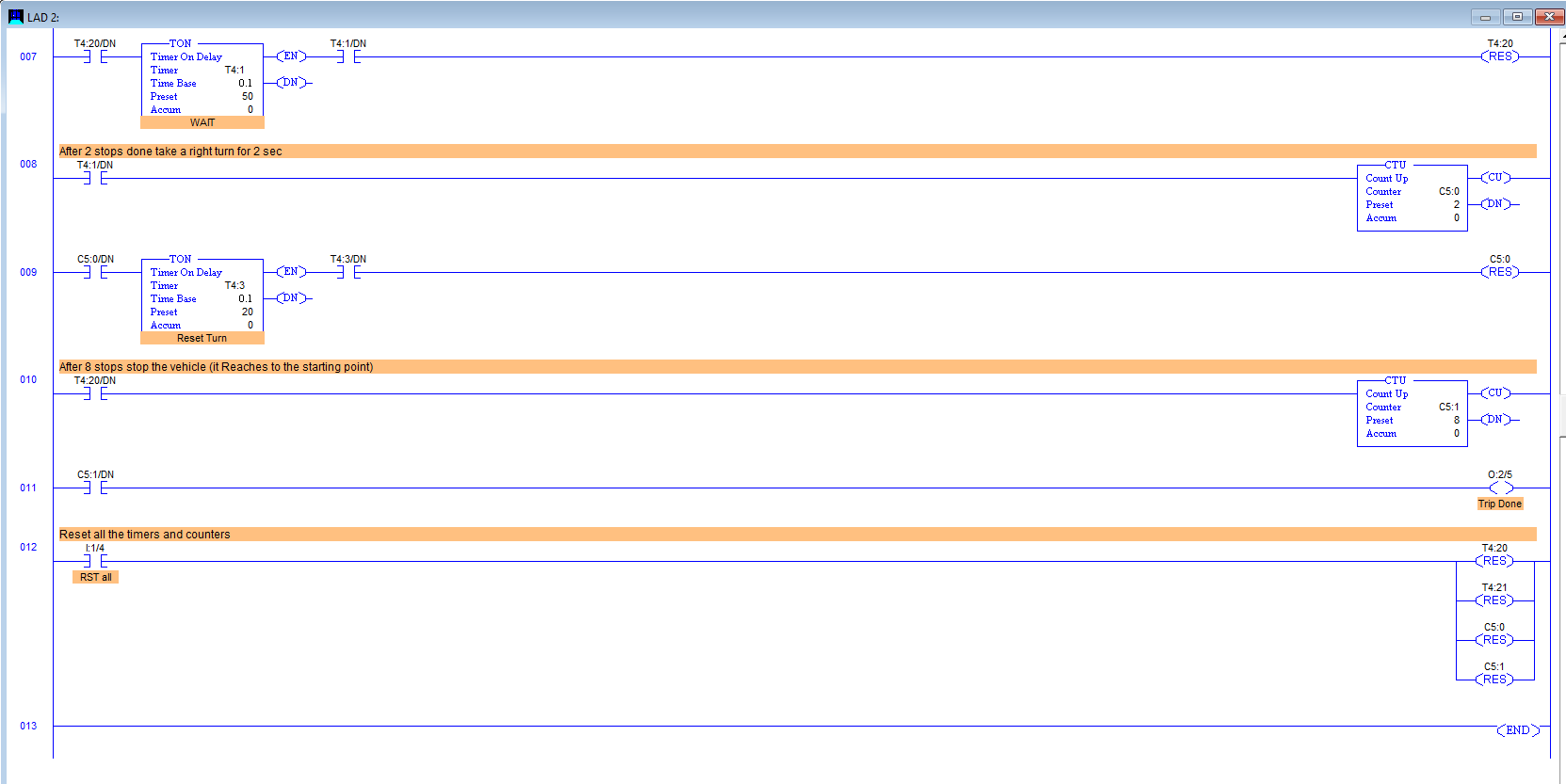


Figure 6.4 Ladder diagram of AVGV

**Steering control**

To help an AGV navigate it can use three different steer control systems. The differential speed control is the most common. In this method there are two independent drive wheels. Each drive is driven at different speeds in order to turn or the same speed to allow the AGV to go forwards or backwards. The AGV turns in a similar fashion to a tank. This method of steering is the simplest as it does not require additional steering motors and mechanism. More often than not, this is seen on an AGV that is used to transport and turn in tight spaces or when the AGV is working near machines. This setup for the wheels is not used in towing applications because the AGV would cause the trailer to jackknife when it turned.

The second type of steering used is steered wheel control AGV. This type of steering can be similar to a car's steering. But this is not very maneuverable. It is more common to use a three-wheeled vehicle similar to a conventional three wheeled forklift. The drive wheel is the turning wheel. It is more precise in following the programmed path than the differential speed-controlled method. This type of AGV has smoother turning. Steered wheel control AGV can be used in all applications; unlike the differential controlled. Steered wheel control is used for towing and can also at times have an operator control it.

The third type is a combination of differential and steered. Two independent steer/drive motors are placed on diagonal corners of the AGV and swiveling castors are placed on the other corners. It can turn like a car (rotating in an arc) in any direction. It can crab in any direction and it can drive in differential mode in any direction.

**Path Decision**

AGVs have to make decisions on path selection. This is done through different methods: frequency select mode (wired navigation only), and path select mode (wireless navigation only) or via a magnetic tape on the floor not only to guide the AGV but also to issue steering commands and speed commands.

**Frequency select mode:**

Frequency select mode bases its decision on the frequencies being emitted from the floor. When an AGV approaches a point on the wire which splits the AGV detects the two frequencies and through a table stored in its memory decides on the best path.

The different frequencies are required only at the decision point for the AGV. The frequencies can change back to one set signal after this point. This method is not easily expandable and requires extra cutting, which makes the process more expensive.

**Path selects mode**:

An AGV using the path select mode chooses a path based on preprogrammed paths. It uses the measurements taken from the sensors and compares them to values given to them by programmers. When an AGV approaches a decision point it only has to decide whether to follow path 1, 2, 3, etc. This decision is rather simple since it already knows its path from its programming. This method can increase the cost of an AGV because it is required to have a team of programmers to program the AGV with the correct paths and change the paths when necessary. This method is easy to change and set up.

**Magnetic tape mode:**

The magnetic tape is laid on the surface of the floor or buried in a 10mm channel; not only does it provide the path for the AGV to follow but also strips of the tape in different combinations of polarity, sequence, and distance laid alongside the track tell the AGV to change lane, speed up, slow down, and stop.

**Traffic control**

Flexible manufacturing systems containing more than one AGV may require it to have traffic control so the AGV's will not run into one another. Traffic control can be carried out locally or by software running on a fixed computer elsewhere in the facility. Local methods include zone control, forward sensing control, and combination control. Each method has its advantages and disadvantages.

**Zone control**

Zone control is the favorite of most environments because it is simple to install and easy to expand. Zone control uses a wireless transmitter to transmit a signal in a fixed area. Each AGV contains a sensing device to receive this signal and transmit back to the transmitter. If the area is clear the signal is set at "clear" allowing any AGV to enter and pass through the area. When an AGV is in the area the "stop" signal is sent and all AGV attempting to enter the area stop and wait for their turn. Once the AGV in the zone has moved out beyond the zone the "clear" signal is sent to one of the waiting AGVs. Another way to set up zone control traffic management is to equip each individual robot with its own small transmitter/receiver. The individual AGV then sends its own "do not enter" message to all the AGVs getting too close to its zone in the area. A problem with this method is if one zone goes down all the AGV's are at risk to collide with any other AGV. Zone control is a cost-efficient way to control the AGV in an area.

**Forward Sensing Control**

Forward sensing control uses collision avoidance sensors to avoid collisions with other AGV in the area. These sensors include: sonic, which work like radar; optical, which uses an infrared sensor; and bumper, physical contact sensor. Most AGV's are equipped with a bumper sensor of some sort as a fail-safe. Sonic sensors send a "chirp" or high frequency signal out and then wait for a reply from the outline of the reply the AGV can determine if an object is ahead of it and take the necessary actions to avoid collision. The optical uses an infrared transmitter/receiver and sends an infrared signal which then gets reflected back; working on a similar concept as the sonic sensor. The problems with these are they can only protect the AGV from so many sides. They are relatively hard to install and work with as well.

**Combination control**

Combination control sensing is using collision avoidance sensors as well as the zone control sensors. The combination of the two helps to prevent collisions in any situation. For normal operation the zone control is used with the collision avoidance as a fail safe. For example, if the zone control system is down, the collision avoidance system would prevent the AGV from colliding.

**Common applications**

Automated Guided Vehicles can be used in a wide variety of applications to transport many different types of material including pallets, rolls, racks, carts, and containers. AGVs excel in applications with the following characteristics:

1. Repetitive movement of materials over a distance
2. Regular delivery of stable loads
3. Medium throughput/volume
4. When on-time delivery is critical and late deliveries are causing inefficiency
5. Operations with at least two shifts
6. Processes where tracking material is important.

**Primary application industries**

Efficient, cost-effective movement of materials is an important, and common element in improving operations in many manufacturing plants and warehouses. Because automatic guided vehicles (AGVs) can deliver efficient, cost-effective movement of materials, AGVs can be applied to various industries in standard or customized designs to best suit an industry's requirements. Industries currently utilizing AGVs include (but are not limited to):

**Pharmaceutical**

AGVs are a preferred method of moving materials in the pharmaceutical industry. Because an AGV system tracks all movement provided by the AGVs, it supports process validation and cGMP (current Good Manufacturing Practice).

**Chemical**

AGVs deliver raw materials, move materials to curing storage warehouses, and provide transportation to other processing cells and stations. Common industries include rubber, plastics, and specialty chemicals.

**Manufacturing**

AGVs are often used in general manufacturing of products. AGVs can typically be found delivering raw materials, transporting work-in process, moving finished goods, removing scrap materials, and supplying packaging materials.

**Automotive**

AGV installations are found in Stamping Plants, Power Train (Engine and Transmission) Plants, and Assembly Plants delivering raw materials, transporting work-in process, and moving finished goods. AGVs are also used to supply specialized tooling which must be changed.

**Paper and print**

AGVs can move paper rolls, pallets, and waste bins to provide all routine material movement in the production and warehousing (storage/retrieval) of paper, newspaper, printing, corrugating, converting, and plastic film.

**Food and beverage**

AGVs can be applied to move materials in food processing (such as the loading of food or trays into sterilizers) and at the "end of line," linking the palletizer, stretch wrapper, and the warehouse. AGVs can load standard, over-the-road trailers with finished goods, and unload trailers to supply raw materials or packaging materials to the plant. AGVs can also store and retrieve pallets in the warehouse.

**Hospital**

AGVs are becoming increasingly popular in the healthcare industry for efficient transport, and are programmed to be fully integrated to automatically operate doors, elevators/lifts, cart washers, trash dumpers, etc. AGVs typically move linens, trash, regulated medical waste, patient meals, soiled food trays, and surgical case carts.

**Warehousing**

AGVs used in Warehouses and Distribution Centers logically move loads around the warehouses and prepare them for shipping/loading or receiving or move them from an induction conveyor to logical storage locations within the warehouse. Often, this type of use is accompanied by customized warehouse management software.[[19]](https://en.wikipedia.org/wiki/Automated_guided_vehicle#cite_note-19) To avoid damage to fragile goods, AGVs are preferred in warehouses that handle fragile items since human errors are reduced to almost zero. Warehouses with hazardous goods have primarily adopted this technology as they can operate in extreme conditions like passing through freezers.

**Theme parks**

In recent years, the theme park industry has begun using AGVs for rides. One of the earliest AGV ride systems was for Epcot's Universe of Energy, opened in 1982. The ride used wired navigation to drive the 'Traveling Theatre' through the ride. Many rides use wired navigation, especially when employees must frequently walk over the ride path such as at (the now-closed attraction) The Great Movie Ride at Disney's Hollywood Studios. Another ride at Hollywood Studios that uses wired navigation is The Twilight Zone Tower of Terror, a combined drop tower/dark ride. The elevator cars are AGVs that lock into place inside separate vertical motion cabs to move vertically. When it reaches a floor requiring horizontal movement, the AGV unlocks from the vertical cab and drives itself out of the elevator.

A recent trend in theme parks is a so-called trackless ride system, AGV rides that use LPS, Wi-Fi, or RFID to move around. The advantage of this system is that the ride can execute seemingly random movements, giving a different ride experience each time.

**Output Devices:**

1. Left Motor
2. Right Motor
3. Speaker
4. Right Turn

**Indicators:**

1. Obstacle Detected
2. Emergency Stop
3. Trip done

**Normally Open (NO) Switch:**

* Default state: Open
* Actuated state: Closed

**Normally Closed (NC) Switch:**

* Default state: Closed
* Actuated state: Open

**Retentive Timers:**

Retentive timers maintain their accumulated time value even when the input condition that started the timer is no longer true. When the input condition is true, the timer starts accumulating time. Once the preset time is reached, the timer's output is energized. Even if the input condition goes false, the accumulated time is retained, and the output remains energized until the timer is reset. Retentive timers are useful when you need to maintain a timed action or delay, even if the initiating condition is no longer present.

**Non-Retentive Timers:**

Non-retentive timers, also known as "one-shot" timers, only accumulate time while the input condition is true. When the input condition becomes false, the timer's accumulated time is reset to zero, and the output is de-energized. Non-retentive timers are useful when you need a timed action or delay that is directly tied to the presence of the input condition.

**Up Counter:**

The counter's input condition becomes true, incrementing the count value by 1. The current count value is compared to the preset value. If the count value is less than the preset value, the counter's output remains de-energized. If the count value reaches the preset value, the counter's output is energized, indicating the target count has been reached. The counter can then be reset, either manually or automatically, to prepare for the next counting cycle.

**System Integration & Testing**

The PLC-based Automatic Visitor Guided Vehicle project involved the integration of several hardware components, including the Programmable Logic Controller (PLC), motors, ultrasonic sensors, a speaker, and various input buttons (start, emergency stop, manual control). The integration process began with the physical mounting and connection of these components within the vehicle's structure. The PLC was securely installed, and all input and output devices were wired to the appropriate PLC terminals. Special care was taken to ensure proper grounding and shielding to minimize electrical interference. The integration of the PLC was a crucial step, as it served as the central control unit for the entire system. The PLC was programmed using ladder logic to coordinate the operation of the motors, sensors, and speaker. Configuring the PLC's input and output assignments, as well as the control logic, was essential for achieving the desired functionality.

**Sensor Integration**

The ultrasonic sensors were strategically placed on the front of the vehicle to detect obstacles in the path. These sensors were connected to the PLC's analog input channels, allowing the PLC to continuously monitor the sensor readings and calculate the distance to any detected objects. Careful calibration of the ultrasonic sensors was necessary to ensure accurate distance measurements. This involved adjusting the sensor parameters, such as the detection range and sensitivity, to optimize their performance within the specific operating environment.

**Motor Integration**

The two motors responsible for the vehicle's movement were connected to the PLC's output channels. The PLC program implemented motor control algorithms to manage the speed, direction, and timing of the vehicle's motion, enabling the desired forward movement, stopping, and turning behavior. Integration of the motor control involved configuring the PLC's output signals, ensuring proper wiring and power connections, and tuning the control parameters to achieve smooth and reliable vehicle navigation.

**Audio Guidance Integration**

The integration of the speaker system was crucial for providing audio guidance to visitors during the vehicle's operation. The speaker was connected to the PLC's output channels, allowing the PLC program to trigger pre-recorded audio messages at the appropriate stop locations. Synchronizing the audio guidance with the vehicle's movement was a key aspect of the integration process. The PLC program was designed to coordinate the timing and triggering of the audio messages to align with the vehicle's navigation through the facility.

**Testing Procedures and Validation**

Individual component testing:

Verifying the proper operation of the sensors, motors, and speaker in isolation. Ensuring each component met its specified performance requirements.

Integrated system testing:

Evaluating the vehicle's navigation, including the start, stop, and turning behavior. Validating the obstacle detection and avoidance functionality. Confirming the synchronization of the audio guidance with the vehicle's movement. Assessing the emergency stop and manual control features.

Stress testing and edge case scenarios:

Subjecting the system to various challenging conditions, such as multiple obstacles. Verifying the system's robustness and ability to handle unexpected situations.

**Result & Observation**

The vehicle was able to successfully navigate the predetermined path, following the programmed sequence of forward movement, stopping, and turning at the specified intervals. The vehicle's movement was smooth and consistent, demonstrating the effectiveness of the PLC's motor control algorithms. The vehicle was able to complete the full 9-stop route without any issues, showcasing the reliability of the navigation logic. The ultrasonic sensors were able to accurately detect the presence of obstacles in the vehicle's path. The PLC's obstacle detection logic successfully interpreted the sensor data and triggered the vehicle to stop before colliding with the obstacles. The vehicle was able to resume its normal operation once the obstacles were removed from its path, demonstrating the effectiveness of the obstacle avoidance functionality. The audio guidance system provided clear and informative instructions to the visitors at each stop location. The synchronization between the vehicle's movement and the audio messages was seamless, enhancing the overall visitor experience. Visitors reported that the audio guidance was helpful in understanding their current location and the next steps in the tour. The emergency stop button functioned as expected, immediately deactivating all the vehicle's outputs and bringing it to a complete stop. The manual control buttons allowed technicians to easily override the automatic movement and control the vehicle's motors individually, which was useful for troubleshooting and maintenance.

**Project Future Improvements and Enhancements**

**Expanded Navigation Capabilities:**

Implement more advanced path planning algorithms to enable the vehicle to navigate more complex routes and adapt to changes in the facility layout. Integrate additional sensors, such as laser scanners or camera-based vision systems, to enhance the vehicle's environmental awareness and obstacle detection capabilities.

**Improved Visitor Experience:**

Develop a touchscreen interface or mobile app that allows visitors to interact with the vehicle, access facility information, and provide feedback. Integrate the vehicle's audio guidance with a multilingual support system, enabling the delivery of instructions in multiple languages.

**Maintenance and Diagnostics:** Develop a comprehensive maintenance and diagnostics system, including automated self-checks and fault reporting, to ensure the vehicle's reliable and continuous operation. Implement predictive maintenance algorithms that can analyze sensor data and usage patterns to anticipate potential issues and schedule proactive maintenance. Provide a user-friendly interface or dashboard for facility staff to monitor the vehicle's health, access diagnostic information, and perform maintenance tasks.

**Energy Efficiency and Sustainability:**

Investigate the integration of renewable energy sources, such as solar panels or regenerative braking, to improve the vehicle's energy efficiency and reduce its environmental impact. Explore the possibility of implementing autonomous charging stations that can automatically recharge the vehicle's batteries, minimizing the need for manual intervention. Analyze the vehicle's power consumption and energy usage patterns to identify opportunities for optimization and improved sustainability.

**Advantages of Automatic Visitor Guided Vehicles (AGVs)**

* Enhanced Visitor Experience: AGVs offer a unique and engaging experience for visitors by providing self-guided tours with informative content.
* Increased Efficiency: AGVs can optimize visitor flow, reducing congestion and wait times, especially in large facilities.
* Improved Accessibility: AGVs can be designed to accommodate visitors with disabilities, providing equal access to information and facilities.
* Staff Optimization: By automating visitor guidance, staff can focus on providing more personalized assistance or handling other tasks.
* Data Collection: AGVs can gather valuable data on visitor behavior and preferences, aiding in facility improvement and marketing strategies.
* Cost-Effective: In the long term, AGVs can reduce labor costs associated with manual visitor guidance.
* Flexibility: AGVs can be easily reprogrammed to accommodate changes in facility layout or tour routes.

**Disadvantages of Automatic Visitor Guided Vehicles (AGVs)**

* High Initial Investment: The development and implementation of an AGV system can be costly, including hardware, software, and infrastructure.
* Technical Complexity: AGVs require sophisticated technology and skilled personnel for operation and maintenance.
* Dependence on Technology: System failures or malfunctions can disrupt visitor experience and facility operations.
* Safety Concerns: Ensuring the safety of visitors and facility personnel is crucial and requires careful planning and implementation.
* Limited Flexibility: In some cases, AGVs might lack the adaptability of human guides, especially in unexpected situations or complex environments.
* Public Acceptance: There might be initial resistance or concerns from visitors about using automated systems for guidance.

**Project Conclusion**

The PLC-based Automatic Visitor Guided Vehicle project has successfully demonstrated the integration of Programmable Logic Controller (PLC) technology, sensor integration, and autonomous control to create a comprehensive solution for guiding visitors through a facility. Through the course of this project, I have designed and implemented a vehicle that can autonomously navigate a predetermined path, detect and avoid obstacles, and provide audio guidance to visitors. The core of the system is the PLC, which orchestrates the various hardware components, including motors, ultrasonic sensors, and a speaker, to achieve the desired functionality. The vehicle's ability to move forward, stop, and turn at specific intervals, as well as its effective obstacle detection and avoidance capabilities, have been thoroughly tested and validated. The integration of the audio guidance system has further enhanced the visitor experience, providing clear and informative instructions to guide the participants through the facility. The inclusion of emergency stop and manual control features has also ensured the safety and flexibility of the system, allowing for immediate intervention and technician-controlled operation when necessary. Overall, this project has showcased my expertise in PLC programming, sensor integration, motor control, and audio system implementation. The successful integration and testing of the various components have demonstrated the reliability and effectiveness of the PLC based Automatic Visitor Guided Vehicle system. As I look into the future, there are several potential enhancements and improvements that could be explored, such as expanded navigation capabilities, improved visitor experience features, connectivity and integration with facility management systems, and enhanced maintenance and diagnostics capabilities. By continuously exploring these avenues, we can further refine and evolve this solution to meet the growing demands of visitor guidance and automation in various industrial and commercial settings. In conclusion, the PLC-based Automatic Visitor Guided Vehicle project has been a valuable learning experience, showcasing my ability to design and implement a complex, integrated system that enhances visitor safety, efficiency, and engagement.

**Chapter-7 Conclusion**

The industrial automation course has been instrumental in advancing my understanding of the principles and practices that define modern industrial systems. Through a comprehensive curriculum, I have gained a solid foundation in the various types of industrial production systems and the ways in which automation can optimize these processes. This knowledge is crucial in today's manufacturing landscape, where efficiency, precision, and reliability are paramount.

One of the key aspects of this course was the in-depth study of Programmable Logic Controllers (PLCs). I learned to program these essential devices, which serve as the brains behind many automated systems. The course provided a detailed exploration of the different programming languages and techniques used in PLCs, enabling me to develop and troubleshoot automation programs with confidence. This skill is particularly valuable in industries where downtime can be costly, and precise control over machinery and processes is essential.

The course also expanded my understanding of sensors and actuators, which are vital components in any automated system. I learned how different types of sensors, such as proximity, temperature, and ultrasonic sensors, can be used to monitor and control industrial processes. Similarly, I studied the various actuators that convert electrical signals into physical action, playing a crucial role in tasks ranging from simple on/off commands to complex motion control. Understanding how to select and implement these components has given me the ability to design systems that are both efficient and effective.

Furthermore, the course introduced me to modern automation techniques and tools that are shaping the future of manufacturing. Concepts such as Industry 4.0, smart factories, and the Internet of Things (IoT) were covered, providing a glimpse into the direction that industrial automation is heading. This forward-looking approach ensures that I am not only equipped with current knowledge but am also prepared to adapt to future technological advancements.

The hands-on component of the course, particularly the use of PLC Logix Pro software, was invaluable in bridging the gap between theory and practice. By simulating real-world scenarios and programming automated systems, I was able to apply the concepts learned in a controlled environment. This experience has built my confidence in working with industrial automation tools and has prepared me to handle complex projects in a professional setting.

Overall, the industrial automation course has been a transformative learning experience. It has equipped me with the technical expertise and practical skills needed to excel in the field of industrial automation. I am now better prepared to contribute to the design, implementation, and maintenance of automated systems that drive efficiency and innovation in the manufacturing sector.

The automatic visitor guided vehicle project, developed using LogixPro software, represents a significant achievement in applying industrial automation concepts within a simulated environment. The project's goal was to create a self-guided tour system that improves visitor safety and engagement by automating the navigation process. By utilizing PLC programming, sensor integration, and audio systems, I was able to design a system that not only moves autonomously but also interacts with its environment and provides valuable information to its users.

During the course of the project, one of the most critical challenges I faced was ensuring effective communication between the ultrasonic sensors and the PLC within the LogixPro simulation. These sensors were tasked with detecting obstacles in the vehicle's path, and their data needed to be accurately processed by the PLC to ensure timely and appropriate responses. In the initial phases of testing, I encountered inconsistencies in how the simulated sensors communicated with the PLC, leading to unreliable obstacle detection. This was a crucial issue, as it directly impacted the safety and functionality of the system.

To overcome this challenge, I employed a rigorous approach to troubleshooting and refining the PLC program. Since the project was developed in a simulated environment, I focused on optimizing the logic within the PLC to ensure accurate and real-time processing of sensor data. This involved adjusting the sensor parameters within the LogixPro environment to simulate more accurate and reliable behavior. I also refined the PLC program to better handle the data being received from the sensors, ensuring that the vehicle responded correctly to any detected obstacles.

Another key aspect of the project was ensuring the system could handle unexpected scenarios. I implemented robust error-handling routines within the PLC program, which allowed the system to respond gracefully to anomalies, such as false positives or missed detections from the sensors. This approach not only improved the reliability of the system but also enhanced its overall resilience in a simulated industrial setting.

In addition to solving the technical challenges, the project also highlighted the importance of user-centered design in automation systems. The inclusion of an emergency stop function and manual control options within the simulation provided an added layer of safety and operational flexibility. These features allowed for manual intervention when needed, ensuring that the system could be controlled directly by a technician in case of any issues.

The integration of an audio guidance system added an interactive element to the project, providing visitors with verbal instructions and information as the vehicle navigated through the simulated facility. This feature demonstrated the potential for combining automation with user engagement tools, creating a more immersive and informative experience.

Overall, the project was a successful demonstration of my ability to apply theoretical knowledge in a practical setting using LogixPro software. It required a deep understanding of PLC programming, sensor integration, and system design, all within a simulated environment that mimicked real-world industrial automation challenges. By addressing the issues related to sensor communication and refining the PLC logic, I was able to create a reliable and effective system that met the project’s objectives.

This experience has significantly enhanced my technical skills, particularly in the areas of simulation and programming within an industrial context. It has also underscored the importance of iterative development and problem-solving in engineering projects. The knowledge and skills I gained through this project will be invaluable as I continue to explore and contribute to the field of industrial automation.

**Chspter-8 References:**

* **https://www.progressiveautomations.com/blogs/news/the-evolution-of-automation**
* **https://inductiveautomation.com/resources/article/what-is-a-PLC**
* **https://www.plcacademy.com/plc-timers/**
* **https://www.plcacademy.com/plc-counters/**
* **https://en.wikipedia.org/wiki/Automated\_guided\_vehicle**