

INDUSTRIAL AUTOMATION USING PROFINET PROTOCOL

by

Roy Abou Samra

Jasmine El Afiouni

Rasha Jabbour

Submitted to the Department of Computer and Electrical Engineering

Faculty of Engineering

University of Balamand

June 2020

ACKNOWLEDGMENTS

This project could not have been accomplished if it were not for the help of multiple people.

Primarily, our sincerest gratitude goes to Dr. Nicholas Haddad who first suggested the idea of the project and helped us all along the way with his insight and knowledge.

We would also like to thank the moderators Mr. Abdel-Monhem Alameddine and Miss Katia Karam for their unwavering support.

And finally, special regards go to our prestigious university, the University of Balamand, for offering us all the tools and facilities necessary to complete this project as it runs.

ABSTRACT

Although more technologically advanced and simpler to configure than Modbus, PROFINET has not been as explored as its predecessor. The purpose of this project is to explore the specifications, capabilities, and applicability of the PROFINET communications protocol when it comes to an industrial and home network. A prototype consisting of a programmable logic controller (PLC), a variable frequency driver (VFD), an AC motor, and a human-machine interface (HMI) was constructed. The PLC, VFD, and HMI communicate together via a common router using PROFINET. Push buttons, LED lights, and roller blinds are installed on the prototype as physical inputs and outputs. Upon programming the prototype completely, PROFINET proved to be flexible and easily configured to be a suitable source of communication between numerous modules. This protocol shows to be capable of both industrial and home network automation as the prototype clearly portrays.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
ABSTRACT	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	vi
LIST OF FIGURES	vii
CHAPTER 1: INTRODUCTION.....	1
1.1 Brief Overview	1
1.2 Objective	1
1.3 Constraints.....	2
1.4 Project Outline.....	2
CHAPTER 2: LITERATURE REVIEW	3
2.1 Modbus.....	3
2.1.1 <i>Introduction on Modbus</i>	3
2.1.2 <i>More on SCADA systems</i>	3
2.1.3 <i>Modbus modes of operation</i>	5
2.1.4 <i>Limitations of Modbus</i>	6
2.1.5 <i>Security Risks of Modbus</i>	7
2.1.6 <i>Conclusion on Modbus</i>	8
2.2 PROFINET.....	9
2.2.1 <i>Introducing PROFINET</i>	9
2.2.2 <i>Why PROFINET</i>	10
2.2.3 <i>System Model of a PROFINET System</i>	12
2.2.4 <i>Device Model of an I/O Device</i>	13
2.2.5 <i>Addressing PROFINET devices</i>	14

2.2.6	<i>Integrating the Web</i>	15
2.2.7	<i>Linking Devices</i>	15
2.2.8	<i>Cabling</i>	15
2.2.9	<i>Advantages over Modbus</i>	18
2.2.10	<i>Introducing PROFINET for Industrial Applications</i>	18
2.2.11	<i>History</i>	19
2.2.12	<i>Wireless Communication</i>	19
2.2.13	<i>Wireless System Architecture</i>	21
2.2.14	<i>Conclusion on PROFINET</i>	22
CHAPTER 3: HARDWARE		23
3.1	Hardware Setup	23
3.1.1	<i>Equipment Overview</i>	24
3.1.2	<i>IP Address Setup</i>	28
3.1.3	<i>Physical Connections</i>	28
3.2	Cost.....	33
CHAPTER 4: SOFTWARE		35
4.1	TIA Configuration.....	35
4.2	Software Setup	36
4.2.1	<i>HMI Root Screen</i>	36
4.2.2	<i>Motor Control</i>	38
4.2.3	<i>Lights</i>	44
4.2.4	<i>Roller Blinds</i>	50
CHAPTER 5: FUTURE WORK		53
CHAPTER 6: CONCLUSION		54
LIST OF REFERENCES		55

LIST OF TABLES

Table 2.1: Advantages of Fiber Optic Cables over copper cables	17
Table 2.2: Comparison between Modbus and PROFINET [11].....	18
Table 2.3: Comparison between Star and Mesh Topologies.....	20
Table 3.1: Connected Devices and their IP Address.....	28
Table 3.2: PLC Inputs with their Physical Push Buttons and their Uses	29
Table 3.3: PLC Outputs with their Position on Kit and the Indication on it	30
Table 3.4: Variable Frequency Driver and its Indication	32
Table 3.5: Total Cost of Kit [7]	33

LIST OF FIGURES

Figure 2.1: Different Modbus Communication Interfaces.....	5
Figure 2.2: Automation Technology.....	9
Figure 2.3: I/O Relationship.....	12
Figure 2.4: Slots and Sub-slots.....	14
Figure 2.5: PROFINET Name Assignment	14
Figure 2.6: 8-wire Cables	16
Figure 2.7: Wireless System Architecture	21
Figure 3.1: Prototype Setup	23
Figure 3.2: PLC.....	24
Figure 3.3: Power Supply	24
Figure 3.4: HMI	25
Figure 3.5: Control Unit.....	26
Figure 3.6: Panel	26
Figure 3.7: Power Module.....	26
Figure 3.8: AC Motor Used.....	27
Figure 3.9: Router Utilized in the Prototype	27
Figure 3.10: Inner Connections of PLC	29
Figure 4.1: Root Screen	37
Figure 4.2: Motor PLC Tags.....	38
Figure 4.3: Motor Activation	39
Figure 4.4: Motor On.....	40
Figure 4.5: Motor Direction	41
Figure 4.6: Motor Direction Indicators	41
Figure 4.7: Motor Set Point Speed	42
Figure 4.8: Motor HMI Tags	42

Figure 4.9: Motor Screen	43
Figure 4.10: Lighting PLC Tags	44
Figure 4.11: Two-way Light.....	45
Figure 4.12: Timed (Stairway) Light	46
Figure 4.13: Sequence Lights On and Off Memory	47
Figure 4.14: Sequencing Lights	48
Figure 4.15: Lighting HMI Tags.....	48
Figure 4.16: Lightings Screen	49
Figure 4.17: Roller Blinds PLC Tags	50
Figure 4.18: Roller Blinds Control	51
Figure 4.19: Roller Blinds HMI Tags.....	51
Figure 4.20: Roller Blinds Screen.....	52

CHAPTER 1: INTRODUCTION

1.1 Brief Overview

An automated industry utilizes control systems, such as a programmable logic control, to handle processes and actions that otherwise a human being conducts. Even though migration to an automated industry might be costly at first, its numerous advantages over manual production are worth it. With faster and safer productivity, automated industries have improved the quality of life globally by meeting ever-growing requirements, amongst other things, in a manner much less error-prone way than human manual production could ever achieve.

With the introduction of PROFINET, an efficient industrial network that can handle a large transfer of data among different devices is easy to achieve. With its seamless reconfiguration and flexibility, PROFINET can be extremely beneficial for industries that need constant upgrading. PROFINET can also be useful for industries that are time-sensitive with its support for real-time transmission.

1.2 Objective

This project aims to design a fully developed industrial network using PROFINET. The major components used in this project include a programmable logic controller (PLC), a variable frequency driver (VFD), a human-machine interface (HMI), roller blinds, and an AC motor. These components communicate with one another through PROFINET to return an industrial network that is seamlessly controlled through an HMI. An important objective of this project is to show how easy it is to develop an industrial network through PROFINET even though it is a powerful and advanced communications protocol.

1.3 Constraints

Due to the COVID-19 pandemic and the resulting strict lockdown measures, the team members were not able to meet up and work on the prototype as planned. As such, original plans of adding a conveyor belt to the programmed motor control and adding weight sensors to that belt were substituted for several home automation functions.

1.4 Project Outline

This project report is divided into six chapters. After this introduction, a literature review of Modbus and PROFINET is conducted in chapter 2 where the concept, specifications, capabilities, and limitations of each protocol are discussed. In chapter 3, there's a brief hardware overview of all the components used in the prototype, as well as the configurations and connections required to replicate this project. In the same chapter, the costs of everything used in and for the prototype are tabulated. In chapter 4, the process of configuring TIA Portal V14 and downloading the necessary add-ons is detailed. Moreover, each function's list of tags, ladder code, and HMI setup are shared and explained in section 4.2. Finally, additional functions that can be added to this prototype in the future are briefly discussed in chapter 5.

CHAPTER 2: LITERATURE REVIEW

2.1 Modbus

2.1.1 Introduction to Modbus

Developed by Modicon Systems in 1979, Modbus is a communication protocol at the application layer that is used for transmitting data over serial lines between field devices. It is an open protocol and is the most widely used network protocol in the industrial manufacturing environment and uses a simple master/slave architecture. Modbus can be used as a transmission medium to send discrete and analog I/O and to send this data between several different devices.

Many modern industrial environments require the use of ICT systems to make life easier in the sense that everything is just more reliable when you can access anything anytime wherever you go. Modbus is an example of a SCADA system, which stands for “supervisory control and data acquisition”, which supports the idea of remote access, such as the internet, whether public or private, from anywhere you go. While it is more efficient to use SCADA protocols as they offer TCP/IP connections, in the sense that you connect anywhere anytime, they do expose vulnerabilities on the security of this system, making them highly vulnerable to security attacks and breaches.

2.1.2 More on SCADA systems

These systems are widely used in industrial plants, with sensors and actuators to collect the data and perform actions over the internet. The SCADA systems contain many components to be able to control, monitor, and send and receive data.

Firstly, the system would need an operator, who is a person in charge of maintaining the system and performing supervisory tasks to keep things in check. Then, the system would

need a human-machine interface (HMI), which is a small screen designed by the people responsible for the project, making work easier for the operator to maintain and create actions to control the process for that specific industrial required task. The HMI gives commands to the system to perform and can be connected to a database to collect data about trends about this system.

In basic SCADA systems, it will implement a master/slave architecture, one called a Master Terminal Unit (MTU), containing the high-level control of the whole system, where it gathers data from remote PLCs and actuators, and sends it to the Remote Terminal Unit (RTU), which is the slave in this architecture. The communication between MTU and RTU goes both directions, but the RTU cannot initiate the communication, it serves as a data collector from devices, and sends this data to the MTU when requested to. An RTU could be the system's main PLC as well.

2.1.3 Modbus modes of operation

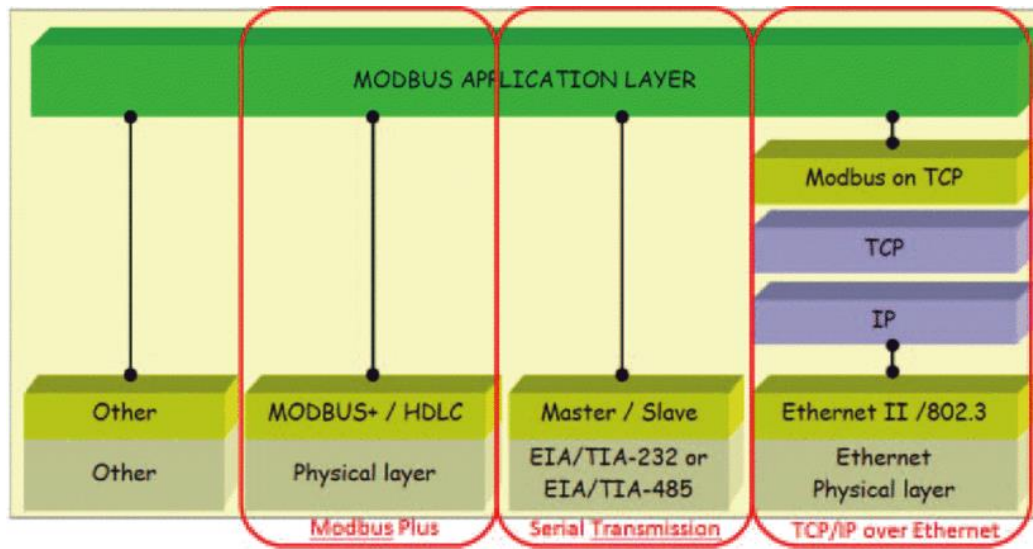


Figure 2.1: Different Modbus Communication Interfaces

As shown in Figure 2.1, different communication interfaces can be used. However, the Modbus protocol is currently implemented using:

1. TCP/IP over Ethernet

In this mode, data packets are encapsulated in TCP (Transmission Control Protocol) specifications and physically transmitted over a standard Ethernet (IEEE 802.3) network. Carrier Sense Multiple Access with Collision Detection (CSMA-CD) is implemented for the medium access control layer.

2. Serial Transmission

This process is carried out using asynchronous serial transmission over a range of physical media (wire: EIA / TIA-232-E, EIA-422, EIA / TIA-485-A; fiber, radio, etc.). The main protocols implemented when wires are employed as the physical medium are EIA/TIA-

232-E (RS232), EIA-422 (RS422) and EIA/TIA-485-A (RS485). We could implement this method of communication by using one of these two message techniques.

- (i) Modbus RTU messages have an basic 16-bit structure with a Cyclic-Redundant Checksum (CRC) that are a simple and reliable way to transmit data
- (ii) Modbus ASCII messages are a 7-bit ASCII format to make them human intelligible although requiring much higher network traffic based on the increase in their size

3. Modbus Plus

Modbus Plus overcomes the ‘single-master’ restriction and is, in fact, a peer-to-peer protocol that runs at 1 Mbps. This mode offers higher transfer rates than other modes with multiple additional advantages. Unlike Modbus, which is open source, Modbus Plus is owned by Schneider Electric and was developed with the aim of overcoming the one master restriction of Modbus Serial communication. Thus, it lacks open specification since it is a proprietary standard.

2.1.4 Limitations of Modbus

- ❑ Modbus is a master/slave protocol which means that it is impossible for a non-master device to report back to the controller about an error or exception (excluding over Ethernet TCP/IP, using open-mpbus) – the master node has to regularly check for changes in every field device in polling mode. This method consumes a great deal of bandwidth and time which might be a hurdle where bandwidth may be expensive, for example in a radio link with limited bitrate.
- ❑ Unless Ethernet TCP/IP is used, it is not possible to address more than 254 devices on the same data link in Modbus. This can be a limitation for a larger-scale project.

- ❑ Modbus transmissions must be uninterrupted which means that only devices that can buffer data can be used for remote transfers in order to avoid gaps in transmission.
- ❑ Since Modbus is an old protocol designed in the 70s to use with programmable logic controllers (PLCs), the versatility in data types is limited by the capabilities of PLCs at that point which means that more complex data structures are not supported.

2.1.5 *Security Risks of Modbus*

Modbus has security attack risks that can be taken advantage of by attackers on these industrial control systems. Some of these attacks on such a system include:

- ❑ **Unauthorized Command Execution:** Since the master and slaves lack authentication, it means that the attacker could send fake messages that are actually malware that could seem like they are coming from a master to the slaves in this system, and the attacker needs to have access to the networking hosting the system in question, or just the network hosting the slaves.
- ❑ **Modbus Denial-of-Service Attacks:** This type of attack, on the other hand, sends senseless messages as if they are being sent by the master to the slaves of this system, the RTUs. An example of this attack involves impersonating the master and sending meaningless messages to RTUs. This can cause the RTUs to fill up this slave with processing actions that are useless to the system, which in turn wastes processing resources as it floods it with useless messages, and that also wastes time for the system.
- ❑ **Man-in-the-Middle Attacks:** Since Modbus protocol lacks integrity checks, this, in turn, enables an attacker to have access to the main network and either change legitimate messages or create fake messages and send them to slave devices to perform some sort of action.

- ❑ **Replay Attacks:** The last type of attack allows the attacker to reuse actual Modbus messages sent by the system sent to or from the slave device to once again waste time and processing resources of the system.

Even though firewalls and intrusion detection systems may stop attackers from exploiting Modbus vulnerabilities, it is always possible to find a way around these security measures. The most ideal way to protect these vulnerabilities is by filling the holes in this protocol. Such a method, however, is difficult to implement as it requires an expert to create significant changes to the architecture and design of this system at the core. While the recent Modbus protocol helps protect against several attacks, there are still the attacks discussed above such as unauthorized command execution or man-in-the-middle attacks that have yet to be resolved.

2.1.6 Conclusion on Modbus

Modbus is a basic protocol that is often praised for its simplicity and reliability. However, it is quite restricting when implemented in large projects such as industrial applications. For example, Modbus only supports one master, as it is a master/slave protocol, which means that one device has to control everything. This is a disadvantage against peer-to-peer protocols where field devices are also allowed to communicate autonomously which could come in hand in decentralization, error detection, and diagnosis. It also does not offer any security measure, which means that it is open to attackers to take advantage of this system.

2.2 PROFINET

2.2.1 Introducing PROFINET

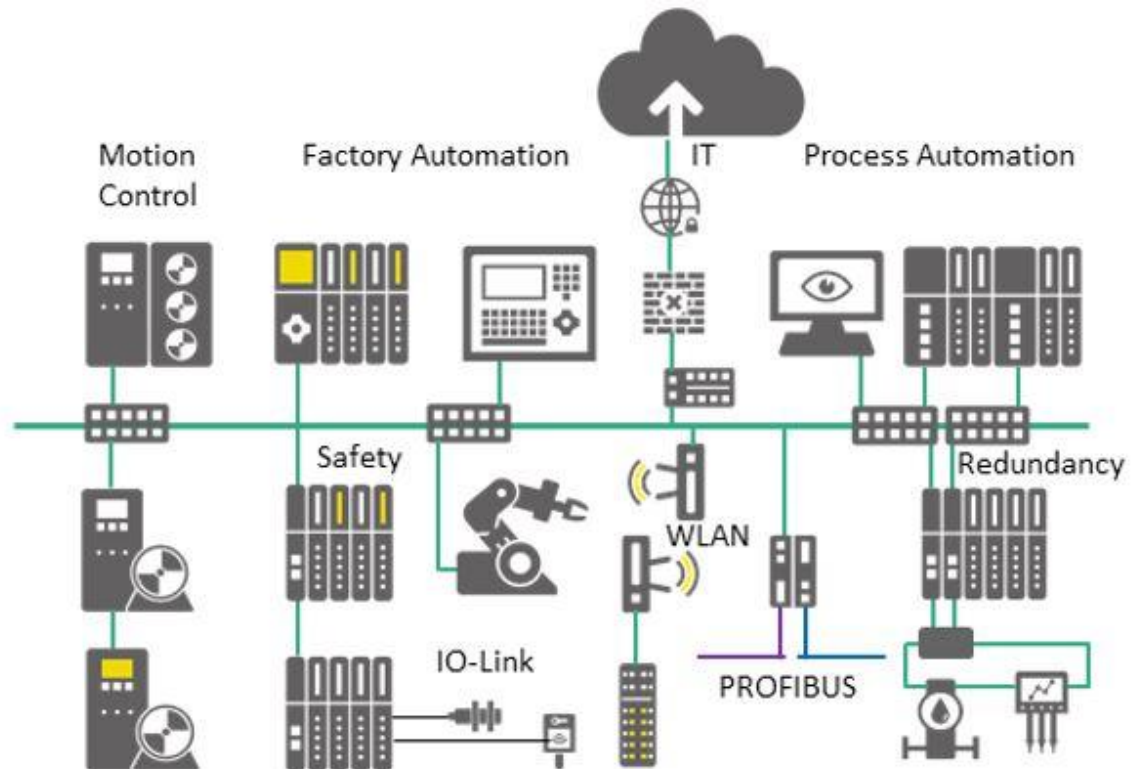


Figure 2.2: Automation Technology

The development cycle of new products is getting shorter and shorter, so continuous development of automation technology is required. Since the early 1990s, the use of Fieldbus technology has made tremendous progress. It makes it possible to move from a centralized automation system to a decentralized automation system. PROFIBUS, which is known as the worldwide market pioneer, has without a doubt set the standard here for over 25 years.

Combining information technology into automation systems can significantly improve communication options between automation systems, a wide range of configuration and diagnostic possibilities, and system-wide service capabilities. The transition from PROFIBUS

to the Ethernet-based PROFINET enables the integration of these functions as an integral part of PROFINET from the very beginning.

All the numerous requirements of automation technology are covered by PROFINET (Figure 2.2). It is the primary decision no matter how you look at it, regardless of whether the application includes production or process automation, or even drives.

The utilization of PROFINET limits the expense of installation for system and machine manufacturers. As for system operators, PROFINET uses autonomously running system units and low upkeep necessities to simplify system expansion and high system availability.

2.2.2 Why *PROFINET*

☐ Sensor/actuator integration

IO-Link, the internationally institutionalized I/O innovation (IEC 61131-9) for speaking with sensors and actuators, can be ideally coordinated into PROFINET. This empowers digital communication up to the sensor/actuator level.

☐ User-friendliness

The ease of use of PROFINET limits the expenses of installation, engineering, and commissioning for machine and system builders. The system owner benefits from the simplicity of system expansion, high system availability, and quick and effective correspondence.

☐ Powerful system base

PROFINET satisfies the most wide-going requirements with its reliable, Ethernet-based correspondence. PROFINET allows automation in real-time and IT incorporation with a single

system, be it very rapid I/O data transmission or data-intensive parameter configuration, making it necessary for Industry 4.0.

☐ Integrated safety

PROFIsafe is a tried-and-tested technology for the functional safety of PROFIBUS that can also be used for PROFINET. The ability to use the same cable for standard and safety-related communications saves equipment, engineering, and setup.

☐ Scalable real-time

Whether it is a simple control task or a demanding motion control application, communication is performed over the same cable in all applications. For high-precision closed-loop control tasks, the deterministic and isochronous transmission of time-based process information can be considered, and its jitter is lower than 1 μ s.

☐ Flexible network topology

Since PROFINET is considered switched Ethernet, many PROFINET devices contain a switch integrated with two or more switches, in order to make them cost-effective and simple to implement. PROFINET is fully Ethernet compatible and therefore compliant with the IEEE standards. In addition, PROFINET utilizes its flexible topology to meet system requirements, as star, linear and ring structures can be easily implemented with fiber and copper cables. Moreover, wireless communication can be enabled by PROFINET using WLAN and Bluetooth.

2.2.3 System Model of a PROFINET System

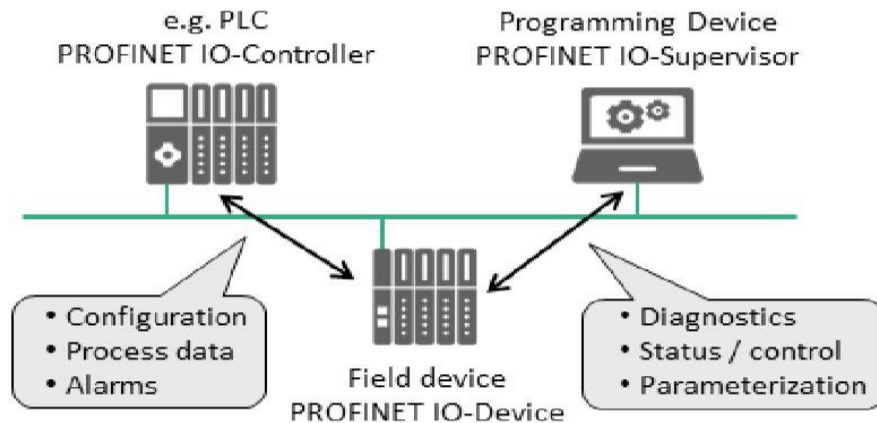


Figure 2.3: I/O Relationship

At least one I/O controller and I/O device need to be present in a system unit. I/O supervisors, on the other hand, can be temporarily integrated for commissioning or troubleshooting purposes.

❑ I/O Controller

Usually, the Programmable Logic Controller (PLC) that runs the automation program. Output data is given to the configured I/O devices by the I/O Controller acting as a provider. I/O controllers can also act as a consumer of input data.

❑ I/O Device

It is connected to at least one I/O controller via PROFINET, working opposite the I/O controller as an output data consumer from the controller and an input data provider.

❑ I/O Supervisor

It is used for either commissioning or diagnostic purposes. An I/O supervisor could be a number of things, like a personal computer, or a programming device, or even a human-machine interface (HMI).

2.2.4 *Device Model of an I/O Device*

An I/O device is typically composed of a communication module with an Ethernet interface along with physical or virtual modules assigned to it. The Ethernet interface with data processing that acts as the access point for communication is also referred to as the Device Access Point (DAP). The assigned modules are usually responsible for the actual process of data communication.

The device model consists of:

- ❑ Modules: The **module's** only task is to provide structuring.
- ❑ Slots: The insertion slots for a module in an I/O field device is designated by the **slot**.

That device typically has at least two slots.

- ❑ Submodules: At least one **submodule** is used to compromise a module.
- ❑ Sub-slots: Submodule can be inserted into available **sub-slots** provided by the module.
- ❑ Channels: The **channels** are the inputs and outputs implemented in the module's submodules.

I/O data in PROFINET can be addressed using slots and sub-slots (Figure 2.4).

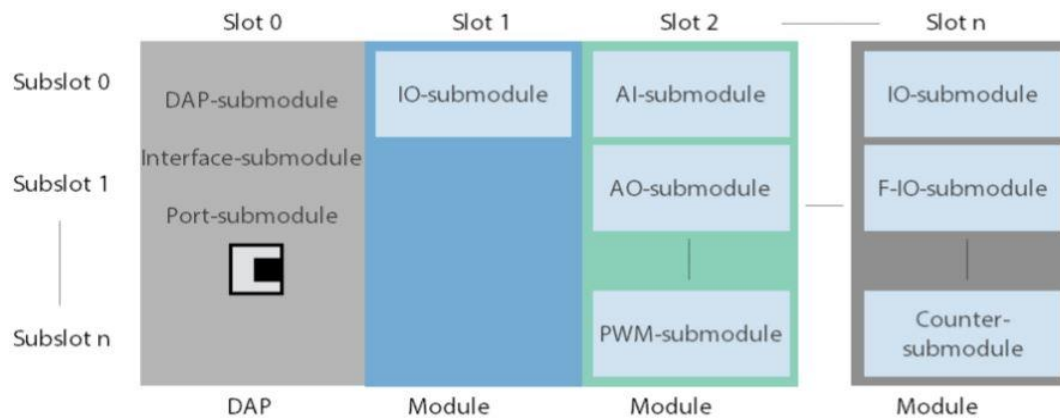


Figure 2.4: Slots and Sub-slots

2.2.5 Addressing PROFINET devices

Unique MAC addresses are used to allow communication between Ethernet devices.

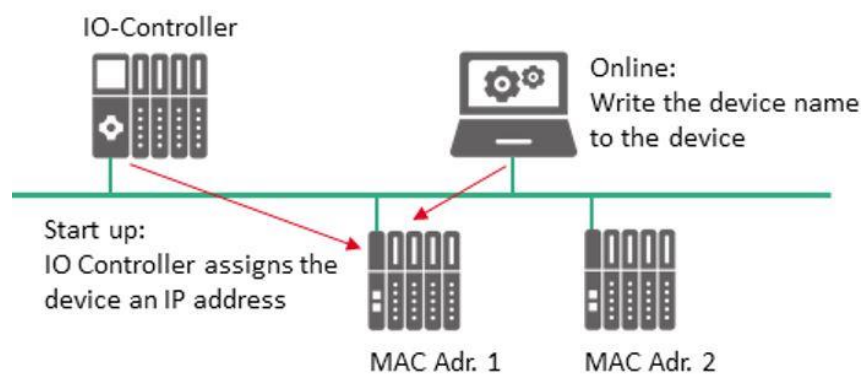


Figure 2.5: PROFINET Name Assignment

The name assignment is frequently used in PROFINET systems to uniquely identify each field device within the I/O system (Figure 2.5). The identification and configuration of the device's name happen within the engineering process; and when the PROFINET application begins, this name is used to resolve the exact IP and MAC addresses.

The name assignment is done using the Discovery and basic Configuration Protocol (DCP Protocol). The protocol issues an engineering tool during device initialization to assign each device name to the individual I/O device, and subsequently to its MAC address.

An automatic assignment of the name by the I/O controller to the I/O device could also be an option, this assignment would be carried out by means of a specified topology based on neighborhood detection. The IP address assignment is carried out in the project through DCP, with the use of commonly used and internationally used Dynamic Host Configuration Protocol (DHCP) or with the use of manufacturer-specific mechanisms. No manual access is required in either of the latter two cases since the IP addresses would be read out of the controller automatically via DCP.

2.2.6 Integrating the Web

Since automation-specific functions are required, PROFINET developed Ethernet to include these functions without confining existing properties. This led to the use of web technologies among other things.

2.2.7 Linking Devices

PROFINET allows other communication systems to be integrated, and this process occurs by linking devices together. This enables the devices to work as PROFINET devices on one hand and to communicate with the Fieldbus system and map it onto PROFINET on the other hand.

2.2.8 Cabling

PROFINET generally uses two types of cables, copper and fiber optics.

❑ PROFINET Copper Cabling

Typically, this type of cable is a 4-wire shielded copper cable. 8-wire cables can be used for higher transmission rates (1000 Mbps).

The cable types could differ in the wire design, whether the wire is solid or flexible, and/or the sheathing.

The wires are also color-labeled. The 4-wire cables consist of two pairs of wires, the first of which has yellow and orange insulation, while the second has blue and white insulation. The wires in these pairs are arranged in such a way that they cross each other.

The colors blue, brown, orange, and green along with the associated white wires are the colors that constitute the four cable pairs of the 8-wire cables (Figure 2.6). These cables abide standard Ethernet applications by having a maximum transmission distance, known as PROFINET end-to-end-link, limited to 100 meters between communication endpoints for copper cables.



Figure 2.6: 8-wire Cables

❑ PROFINET Fiber-Optic Cabling

FOC (fiber-optic cables), utilized for connection of automation islands and systems, should be used in cases of electromagnetic interference fields or when there is an anticipation of high potential differences. The aforementioned interferences are eliminated with the use of FOC.

Table 2.1: Advantages of Fiber Optic Cables over copper cables

Advantages of FOC over copper cables
Span larger distance
Electrical separation enabled between connected plant sections
Immunity to electromagnetic interference

Four different fiber types are possible for PROFINET use:

- ☐ Polymer Clad Fiber (PCF) (Glass fibers)
- ☐ Plastic Optical Fiber (POF) (Plastic fibers)
- ☐ Single-mode Glass Fiber
- ☐ Multimode Glass Fiber

The requirements of the automation project are the main factors in choosing the type of fiber.

2.2.9 Advantages over Modbus

Table 2.2: Comparison between Modbus and PROFINET [11]

Modbus	PROFINET
A simple protocol that only allows for data transmission	A sophisticated protocol that allows for data transmission and network monitoring
$2^{16} = 65,536$ bits or words are permitted for the internal registers	32767 slots with 32767 subslots are permitted for the internal registers, as either bits, bytes, or words
No support for isochronous real-time transportation class	Support for isochronous real-time transportation class

2.2.10 Introducing PROFINET for Industrial Applications

Factory automation is the future for numerous reasons. Factories have grown in complexity exponentially since the 1980s, and only automation can meet the ever-growing requirements. Fewer humans in production results in a lower error and higher efficiency. Wireless communication in factory automation, by translating physical signals to logical representation, means less wear-and-tear and, in hand, less production downtime. However, migrating from a non-automated to an automated factory can prove difficult.

2.2.11 History

The first generation of Fieldbuses was introduced in the early 1990s. It meant fewer physical signals for sensors/actuators and a more logical representation of each signal onto Fieldbuses. Thus, the cost of installation decreased. Early on, the Fieldbuses' speeds were improved and increased from 30 kb/s to around 1 Mb/s for more reliable connection quality. Almost a decade later, the next-generation Fieldbuses focused on Ethernet-based solutions. Since Ethernet has the capability of dividing one network into subnetworks, the unique next-generation Fieldbus protocols were compatible with one another, leading to even better efficiency.

2.2.12 Wireless Communication

Wireless communication in factory automation is very beneficial. Preventive maintenance is less frequent with wireless communication, thus leading to less downtime and more productivity. The market for wireless equipment targeted at factory automation is gradually increasing. Wireless communication can be integrated into factory automation at four different levels:

1. Bridging of network segments
2. Wireless communication to a field device
3. Field device wireless access to sensors and/or actuators
4. Wireless access to field devices used for management and monitoring

At the first level, a high-speed wireless link, like the IEEE 802.11b/g, must be used. Ethernet communication stemming from and to the remote network segment would be bridged across the wireless link. At the second level, a suitable wireless technology, similar to IEEE 802.11b/g, must be chosen to satisfy the requirements needed for maximum latency of 20

milliseconds. At level 3, a low latency must be managed; thus, the high-speed wireless link should be improved as much as possible to transmit small packets of data across a wide range of sensors and/or actuators. At the first three levels, a star network topology is chosen since it provides a reliable low-latency short-range radio communication system for a high number of nodes while at level 4, a mesh network topology would be utilized due to its capabilities in long-range communication. Moreover, since non-real-time data transmission is being handled at level 4, latency is not as prioritized in level 4 as it is in the other three levels.

Table 2.3: Comparison between Star and Mesh Topologies

Comparatives	Star Topology	Mesh Topology
Cost	Cheap	Expensive
Routing	Data is routed from the connection of the central network	Data is routed from one device to another
Scalability	Good	Poor
Complexity	Simplex	Complex
Installation & Reconfiguration	Easy	Difficult

For process automation, similar to the fourth level, even though mesh network topology is typically chosen, both star and mesh network topologies can be used. Ultimately, range and latency requirements determine what topology is more suitable. Failure to differentiate factory automation and process automation and their fundamental requirements will lead to a nonoptimal, subpar performance.

2.2.13 Wireless System Architecture

There are three levels of communication networks in any wireless system architecture: control, field, and sensor levels. These communication networks traditionally have followed a strict hierarchical structure (Figure 2.7). An Ethernet network is usually on the control network. On the level of the field network, a high-speed serial bus is used, similar to RS485. Discrete signals such as 4-20-mA, 0-24-V are used for communication on the sensor network level.

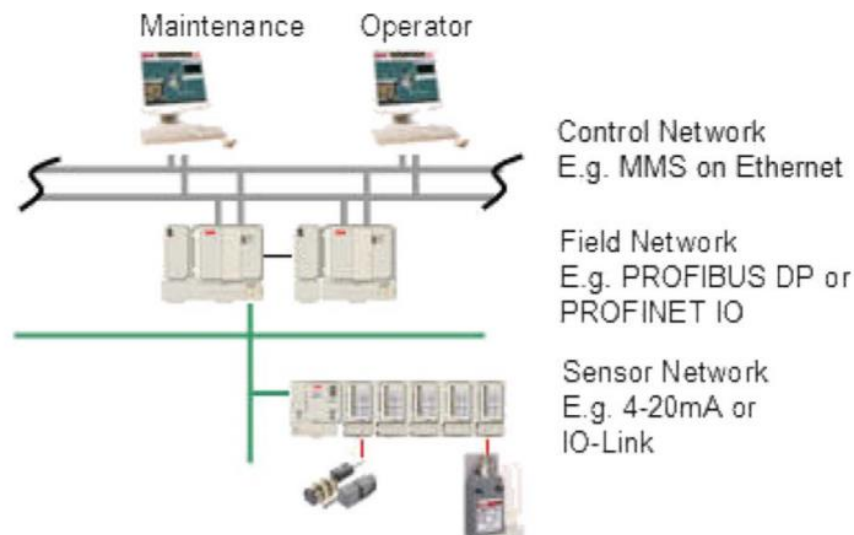


Figure 2.7: Wireless System Architecture

There are two communication structures:

1. Traditional Controller-Centric Architecture

Programmable logic controllers (PLCs) are used in this architecture as gateways between the control network and the field network. Master-slave protocols are used by the Fieldbuses on top of physical layers for communication.

2. Network-Centric Architecture

Since more data is being communicated through networks in automated industries, a more efficient structure was needed. PROFINET I/O allowed for such a structure where programmable logic controllers (PLCs) didn't always have the gateway role as that proved to be computationally heavy on them. Since field and control networks are Ethernet-based, it was possible to join both networks on the same Ethernet network. This led to better flexibility and efficiency because data that isn't required by the PLCs can now be routed between these two networks without having to include PLCs as gateways. Because the network has now become the center in modern automation networks, field devices can be directly accessed making industrial automation that much seamless and easy for configuration and parameterization.

2.2.14 Conclusion on PROFINET

PROFINET is one of the best mechanisms that can be used for industrial automation. Since it is easily customized and re/configured, PROFINET is future proof and seamlessly adapts to the ever-growing requirements of industries today. With the addition of PROFI-safe, wireless automation is less susceptible to third-party intervention.

CHAPTER 3: HARDWARE

3.1 Hardware Setup

The kit, as seen below (Figure 3.1), comprises numerous electrical components that connect to make a prototype of a factory automation system. In this chapter, each component is explored in detail.



Figure 3.1: Prototype Setup

3.1.1 Equipment Overview

1. Programmable Logic Controller:

The programmable logic controller (PLC) is possibly the most critical device in the prototype. It is an advanced industrial automation control-system designed for manufacturing. The PLC constantly tracks the state of input devices and updates the state of output devices accordingly. Input devices can range from push buttons to sensors, while output devices can range from lamps to motors. In this



Figure 3.2: PLC

project, the SIMATIC S7-1200 CPU 1214C 6ES7214-1AG40-0XB0 is used (Figure 3.2) [3]. The project utilized push buttons as the physical inputs to the PLC and LED lights & an AC as physical outputs. The same inputs and outputs are also represented in the HMI discussed below.

2. Power Supply:

Each PLC has a list of compatible power supplies that can be used to power it. In this project, the LOGO!POWER 24 V / 2.5 A 6EP3332-6SB00-0AY0 (Figure 3.3) is used to power the PLC [2].



Figure 3.3: Power Supply

3. Human Machine Interface:

A human-machine interface, commonly referred to as HMI, is an equipment used to control and monitor machines in the industry. It consists of a touchscreen as well as physical buttons for easy control. An HMI



Figure 3.4: HMI

could prove advantageous when there are numerous indicators that must be continuously monitored since every single indicator would be easily viewed remotely on the HMI. An HMI that is used daily on a global scale is an ATM machine. For industrial HMIs, an operator is usually assigned to each HMI to monitor and control a machine. For this project, the SIMATIC HMI KTP400 Basic 6AV2123-2DB03-0AX0 (Figure 3.4) was utilized [1].

Variable Frequency Driver:

Referred to as VFD, variable frequency drivers are used to run and configure AC motors. Through PROFINET, PLCs can connect to VFDs and, in turn, control and monitor the speed of any AC motor. In this project, the SINAMICS G120 CU240E-2 PN 6SL3244-0BB12-1FA0a control unit is used (Figure 3.5) [5] as a



communication link between the PLC and the motor. For easy configuration of the motor through the control unit, the SINAMICS G120 6SL3255-0AA00-4CA1 panel (Figure 3.6) is used [6]. A power module is required

to power the control unit, and in hand, the panel. In the case of a motor, the power module also handles the motor braking function [16]. For the specified control unit and panel mentioned above, the SINAMICS Power Module PM240-2 6SL32101PB130AL0 (Figure 3.7) is used [4].

4. AC Motor:

Through electromagnetic induction, an AC motor, which is an electric motor, converts alternating current into power. In this project, a 3-phase motor (JL 562-4, Figure 3.8) with the following specifications is used as one of the outputs:



Figure 3.5: Control Unit



Figure 3.7: Power Module

- 50 Hz
- 220/380 V
- 0.09 kW
- 0.65/380 A
- 1330 RPM
- $0.67 \cos\phi$



Figure 3.8: AC Motor Used

5. Router:

A router is an electric device used to connect different computers, or equipment, together. It provides some sort of network hub for most of the equipment mentioned above to connect to each other, thus allowing them to communicate with one another. A Thomson TG585 v7 (Figure 3.9).



Figure 3.9: Router Utilized in the Prototype

3.1.2 IP Address Setup

This is how each equipment's IP address should be setup:

Table 3.1: Connected Devices and their IP Address

Device	IP Address
PLC	192.168.0.10
HMI	192.168.0.12
VFD control unit	192.168.0.14
Router	192.168.0.1
All	Subnet mask: 255.255.255.0

3.1.3 Physical Connections

The connections between the physical inputs (push buttons) and the physical outputs (lamp, motor, and roller) and the PLC inputs and outputs (Figure 3.10) are listed respectively in table form below. Every push button is connected to a specific input of the PLC and to the ground from its positive and negative terminals respectively.

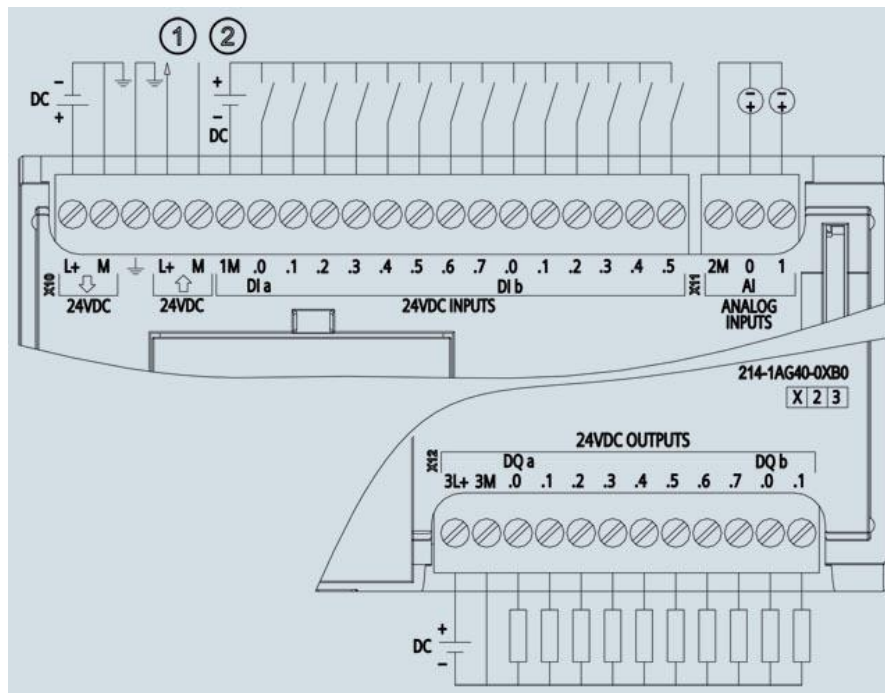


Figure 3.10: Inner Connections of PLC

1. Inputs:

Table 3.2: PLC Inputs with their Physical Push Buttons and their Uses

PLC input	Physical push button	Action
I0.0	DI0	Turns on stairway lamp (timed light)
I0.1	DI1	Rolls the blind up
I0.2	DI2	Rolls the blind down
I0.3	DI3	Turns the motor on if off or off if on

I0.4	DI4	Triggers sequencing lights
I0.5	DI5	The two-directional lamp is triggered either on or off
I0.6	DI6	The two-directional lamp is triggered either on or off

2. Outputs:

Table 3.3: PLC Outputs with their Position on Kit and the Indication on it

PLC output	Position on kit	Indication
Q0.0	First row, the first column (lamp)	Motor is ON
Q0.1	First row, second column (lamp)	The motor is going in the forward direction
Q0.2	First row, the third column (lamp)	The motor is going in the reverse direction
Q0.3	Second row, the first column (lamp)	Light 1 used as: Beginning of sequencing

		Light for the 2 directional lamp
Q0.4	Second row, second column (lamp)	Light 2, used as a second light in sequencing
Q0.5	Second row, the third column (lamp)	Light 3, used as last light in sequencing
Q0.6	Third row, the first column (lamp)	DAMAGED
Q0.7	Third row, second column (lamp)	Timed stairway light is ON
Q1.0	Analog output	Connected to the relay of the roller up function
Q1.1	Analog output	Connected to the relay of the roller down function

Table 3.4: Variable Frequency Driver and its Indication

VFD output	Indication
Q257	Motor is running
Q256.3	The motor is going in the reverse direction

Note: Red light on the kit used as a power indicator. When the kit is on it turns on.

3.2 Cost

Table 3.5: Total Cost of Kit [7]

Hardware or Software	Customer Price
SIMATIC S7-1200 CPU 1214C 6ES7214-1AG40-0XB0	\$536.37
LOGO!POWER 24 V / 2.5 A 6EP3332-6SB00-0AY0	\$69.29
SIMATIC HMI KTP400 Basic 6AV2123-2DB03-0AX0	\$465.33
SINAMICS G120 CU240E-2 PN 6SL3244-0BB12-1FA0	\$481.83
SINAMICS G120 6SL3255-0AA00-4CA1	\$75.22
SINAMICS Power Module PM240-2 6SL32101PB130AL0	\$156.30
JL 562-4	\$70.00
Thomson TG585 v7	\$30.00

Roller blinds	\$250.00
Push button (total of 7)	\$5/piece
Lights (total of 9)	\$2/piece
TIA Portal V14 License (one time purchase)	\$2,412.00 [17]
TOTAL	\$4,599.34

CHAPTER 4: SOFTWARE

4.1 TIA Configuration

Firstly, we downloaded the TIA Portal V14 on our PC and activated it with its required license.

TIA is needed to configure the PLC and HMI and G120 Motor Drive, write the PLC program, design the HMI, and connect them all together through PROFINET. We were able to successfully configure the correct versions of both our PLC S7-1200 and HMI KTP400 Basic Panel, and connect them together. However, there was no available option to configure any Drivers within the software. After some research, we discovered that SIMATIC TIA Portal did not include the Drivers configuration option because it requires an extension, called SINAMICS StartDrive. This extension is available for download on the Support Siemens website, after making an account using a valid university email, otherwise, the extension cannot be downloaded.

After the download was completed, there was a new tab on the TIA Portal start page called Drivers, in which we were able to find the specific G120 Motor Drive required. We were able to configure it but when attempting to download the configurations to the device, we received a warning sign that the firmware in our physical drive is version 4.7.10 while the one in the software is version 4.7.6, and while the download could be successful, it might cause issues. To be safe, we decided to download the latest SINAMICS StartDrive update, Update 8, from the Support Siemens website, along with the most recent version of Hardware Support Package (HSP). The HSP was installed in the TIA's support packages, upgrading the firmware to version 4.7.10 successfully.

For any further complications that could arise from setting up the TIA Portal, checking the Support Siemens Forums is recommended as they are filled with users that went through the same issues and more often than not received a very useful piece of advice that could very well work in your case, as it did in ours.

4.2 Software Setup

4.2.1 *HMI Root Screen*

The HMI is the main source of navigation for our setup. It has two types of screens: a root screen and a screen for every function on the root screen.

To navigate between the different HMI sections, a similar root screen to the one below should be used.

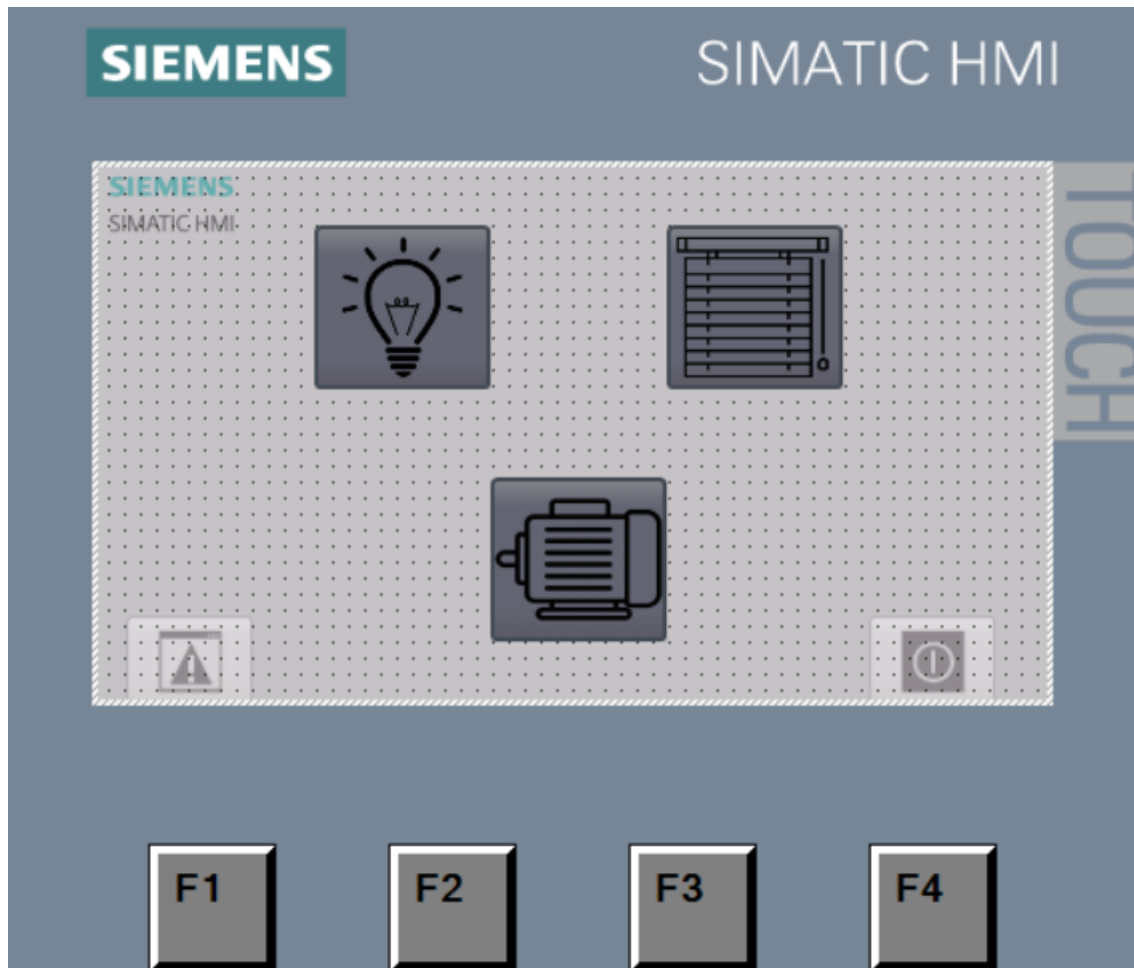


Figure 4.1: Root Screen

Each button is to access a screen. The button that is labeled as the motor, takes you to the motor control function screen on the HMI. The button that is labeled as a light bulb, takes you to the lights function screen, and the final button takes you to the roller blinds function on the HMI.

4.2.2 Motor Control

a. Code:

We need a list of tags to be able to run the motor control, and the below tag list is used for the upcoming motor drive ladder codes.

Note: The forward and reverse pushbuttons are memory inputs, meaning they can only be accessed from the HMI.

motor drive								
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	
1	Active_0	Bool	%Q256.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Active_1	Bool	%Q257.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Active_2	Bool	%Q257.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Active_3	Bool	%Q257.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	Active_4	Bool	%Q257.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Active_5	Bool	%Q257.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Active_6	Bool	%Q257.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Active_7	Bool	%Q257.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Motor Running	Bool	%Q257.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Motor Light	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	Forward Pushbutton	Bool	%M0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	Reverse Motor	Bool	%Q256.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	Forward Light	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	Reverse Light	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	hmi set speed	Int	%MW20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	NormSpeed	Real	%MD22	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	Motor SP speed	Int	%QW258	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	Reverse Pushbutton	Bool	%M0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	ON status motor	Bool	%M1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	OFF status motor	Bool	%M2.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	ON/OFF	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure 4.2: Motor PLC Tags

To start, the motor needs to be turned either ON or off. To do this, the outputs should remain active. Thus, in the ladder code below, M0.0 (or any other memory) is used to activate the outputs, which are required to be active for the motor to run.

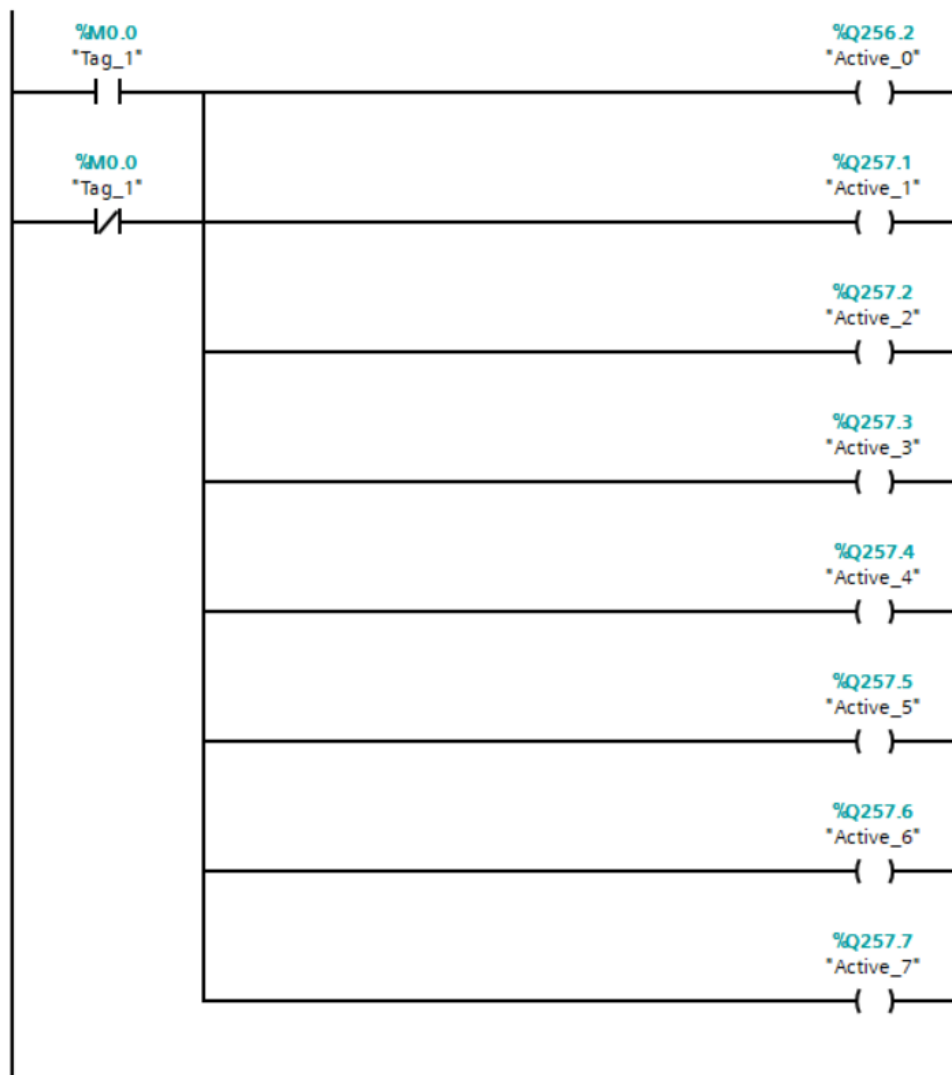


Figure 4.3: Motor Activation

In the following ladder code, a single push button is programmed to either turn the motor ON if the motor is OFF or vice versa. The push button's ON or OFF status is kept in memory, which, in turn, decides if the motor, and its light indicator, will turn ON or OFF.

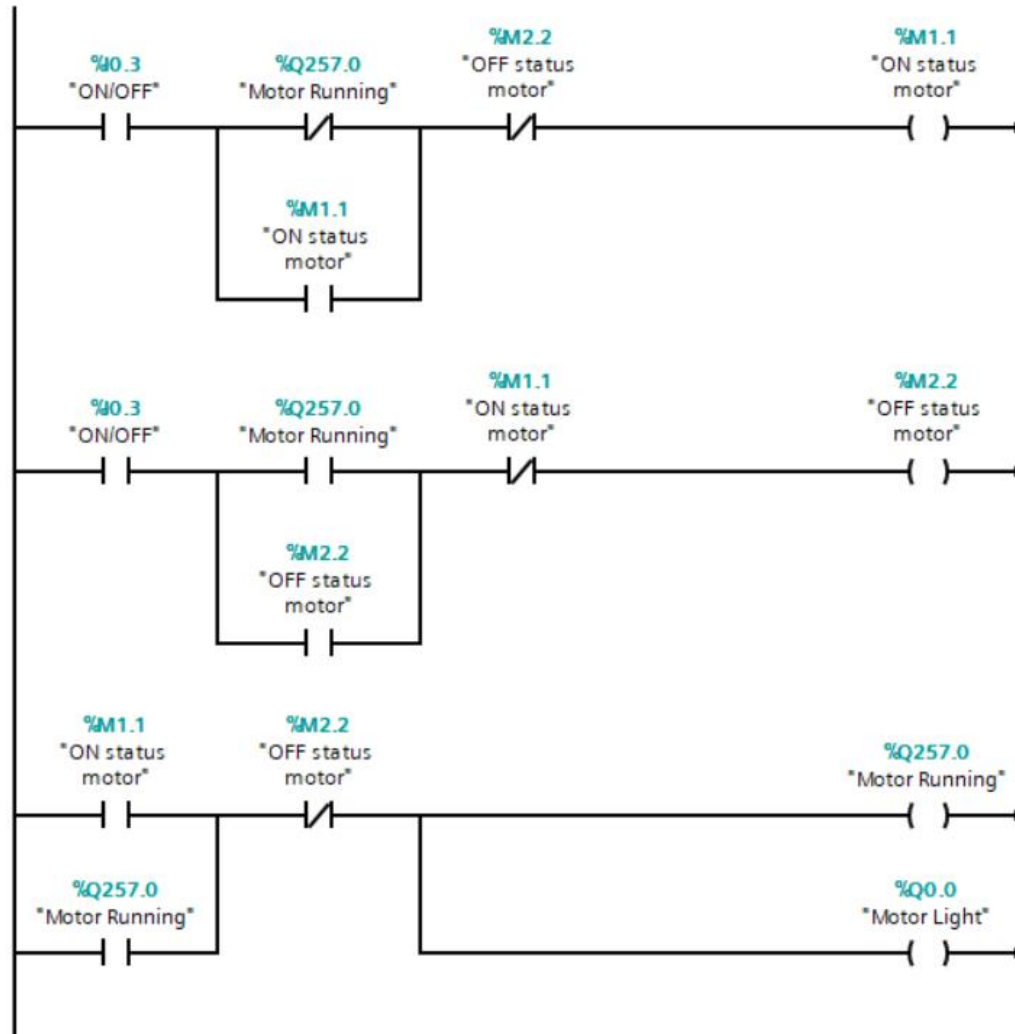


Figure 4.4: Motor On

The ladder code below handles what happens when the forward button is pushed while the motor is on. It also handles what happens when the reverse button is pushed.



Figure 4.5: Motor Direction

The below ladder code handles what happens with the forward and reverse lights when the motor goes in either the forward or reverse direction.

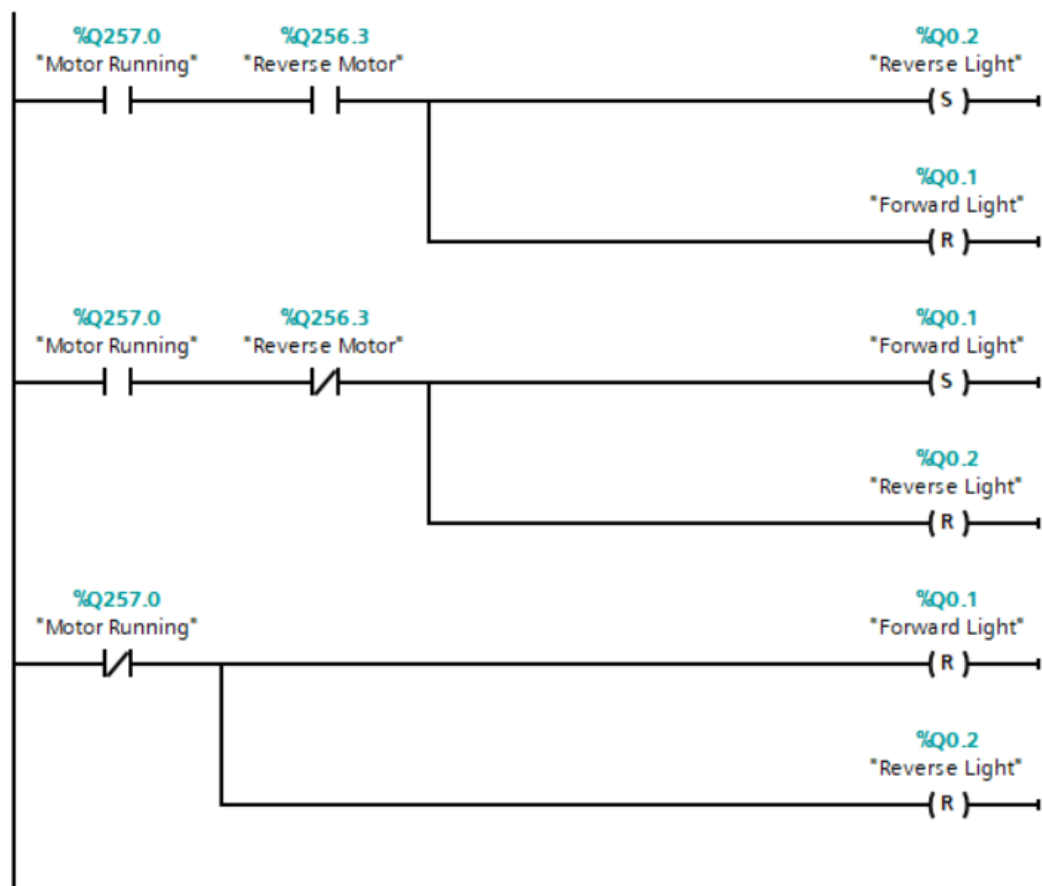


Figure 4.6: Motor Direction Indicators

To be able to manually edit the speed of the motor using the HMI, the following ladder is used. The motor gets its set point from NORM_X, whose job is to normalize the value of the tag at the VALUE input. Scale_X, whose job is to scale the value of the tag at the VALUE input, goes directly to the driver. The normalized speed indicates the percentage. The HMI set speed is the speed that is set through the HMI.

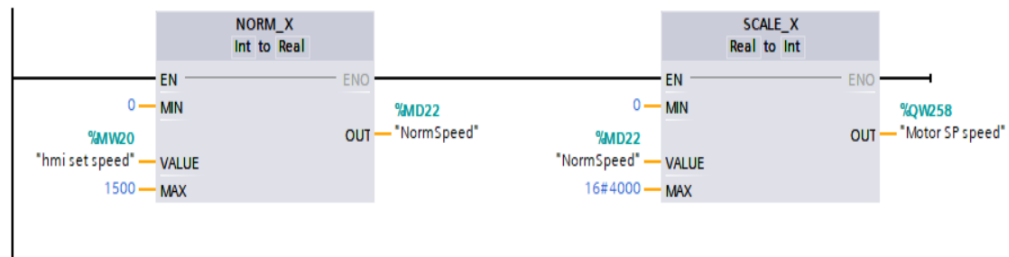


Figure 4.7: Motor Set Point Speed

b. HMI:

The following tag list is used for the specific motor section in the HMI:

hmi motor				
	Name ▲	Data type	Connection	PLC name
DI	Forward Pushbutton	Bool	HMI_Connectio...	PLC_1
DI	hmi set speed	Int	HMI_Connectio...	PLC_1
DI	Motor Running	Bool	HMI_Connectio...	PLC_1
DI	Reverse Motor	Bool	HMI_Connectio...	PLC_1
DI	Reverse Pushbutton	Bool	HMI_Connectio...	PLC_1

Figure 4.8: Motor HMI Tags

These tags are implemented in the buttons shown in the screen below:

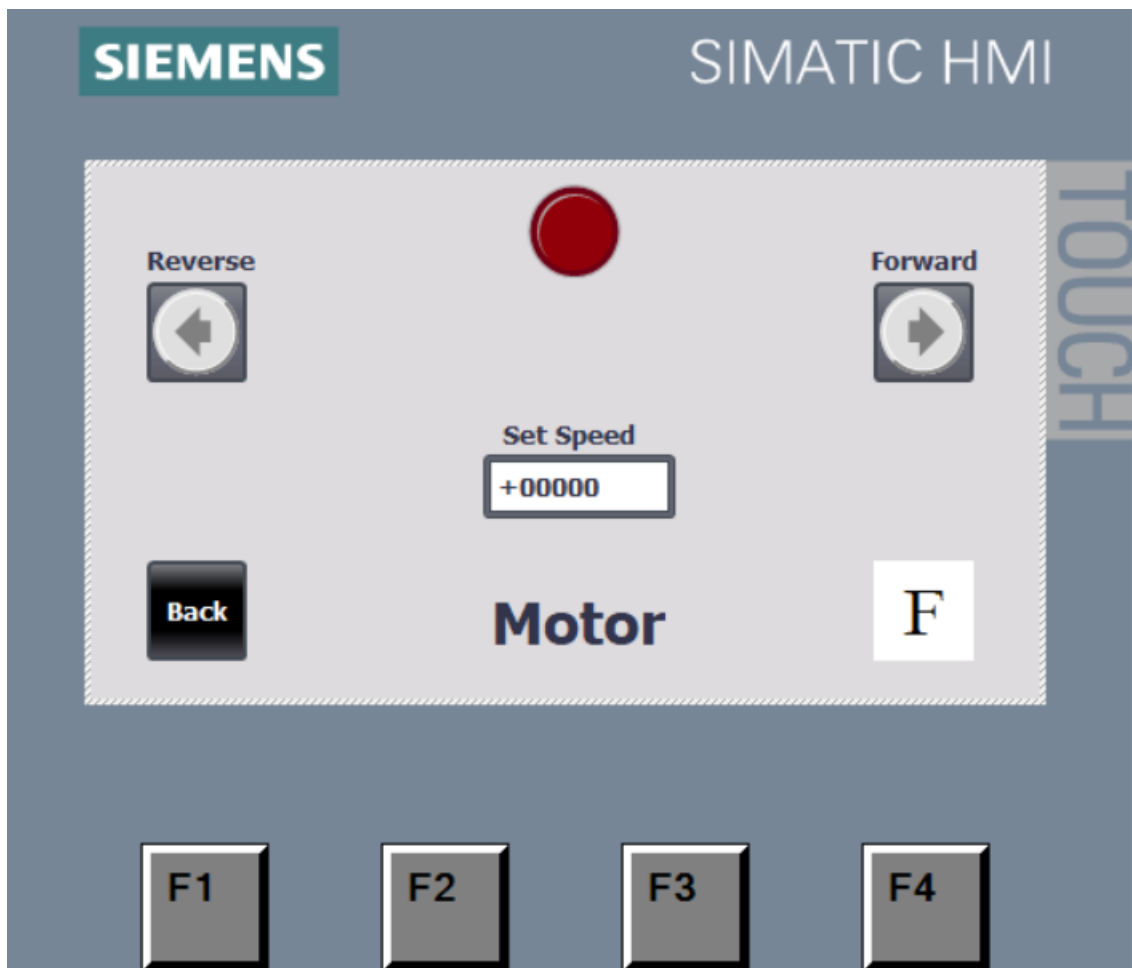


Figure 4.9: Motor Screen

In the HMI's motor control section, the actions and indicators on the screen perform as follows:

- Red Button: It is not a button. It is an indicator that the motor is on. Red when off, green when on.
- Forward: Forward Push button
- Reverse: Reverse Push-button
- Set Speed: HMI set speed

- F: Not a button. An indicator of the motor direction. F when forward, R when backward
- Back: Takes screen to Root Screen

4.2.3 Lights

a. Code:

The tag list below is used for the upcoming light functions ladder codes.

Lighting							
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...
1	Light 1	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	Timed light	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	30s light	Bool	%Q0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	OFF status light	Bool	%M2.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	ON status light	Bool	%M1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	2D-1	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	2D-2	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Light 3	Bool	%Q0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	sequence button	Bool	%I0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	ON1	Bool	%M1.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	ON2	Bool	%M1.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	ON3	Bool	%M1.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	OFF seq	Bool	%M1.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	Light2	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 4.10: Lighting PLC Tags

1. Two-way:

The point of this function is to use two push buttons to turn the same light on and off. Both buttons can turn the light on and off individually, but they also work in sync, meaning that if the first one turns the light on, the second can turn it off. In the ladder code below, the push of any of the buttons is kept in memory as either an ON status or an OFF status. Those statuses, in turn, decide if the light will turn on or off.

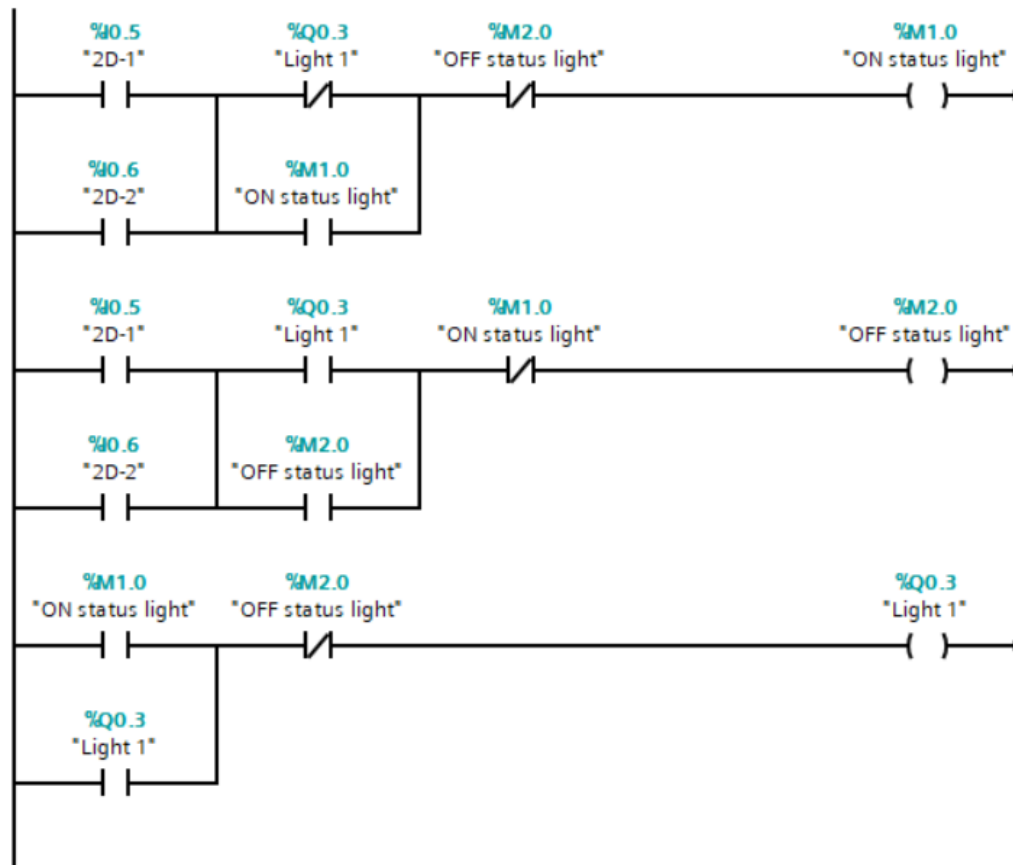


Figure 4.11: Two-way Light

2. Timed (Stairway):

The point of this function is to turn the light on for a given amount of time with the simple push of a button, similar to a stairway light in a building. In the ladder code below, when the button is pushed, the light turns on for 30 seconds and then turns off, as programmed. Pushing the button while the timer is running doesn't reset the timer, thus the button is ignored while the timer is still running.

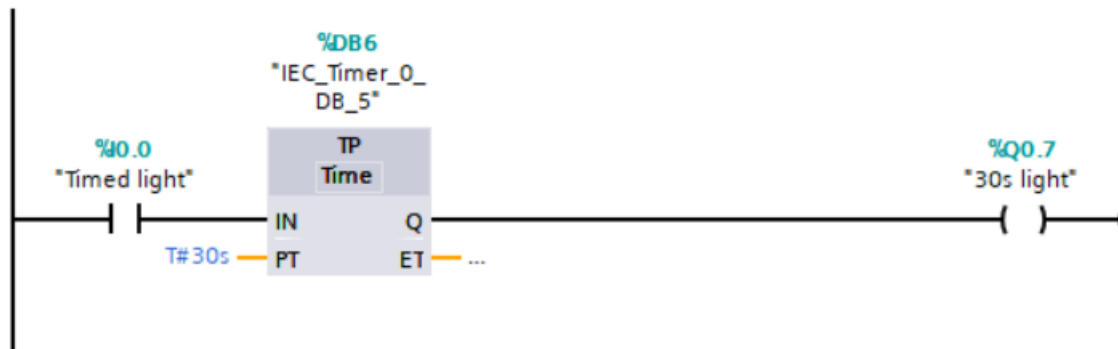


Figure 4.12: Timed (Stairway) Light

3. Sequential:

The point of this function is to turn on three lights, one after the other. So with the push of a button, the first light turns on, and with each push of the same button the next one turns on. After the third light is on the button turns all of the lights off. In the ladder code below, we had to accommodate the fact that we need to turn on three lights before they turn off. Thus, we have three ON status memories, one for each light, and one OFF status memory for when we push the button after all the lights are on.

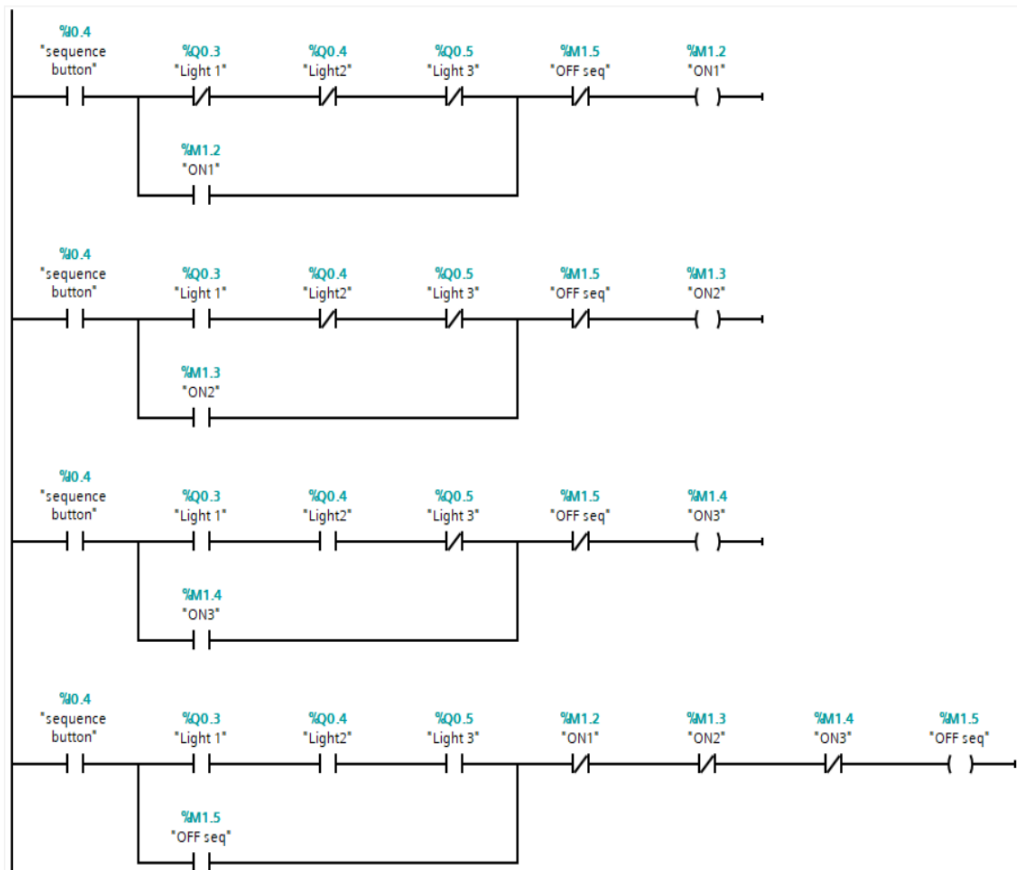


Figure 4.13: Sequence Lights On and Off Memory

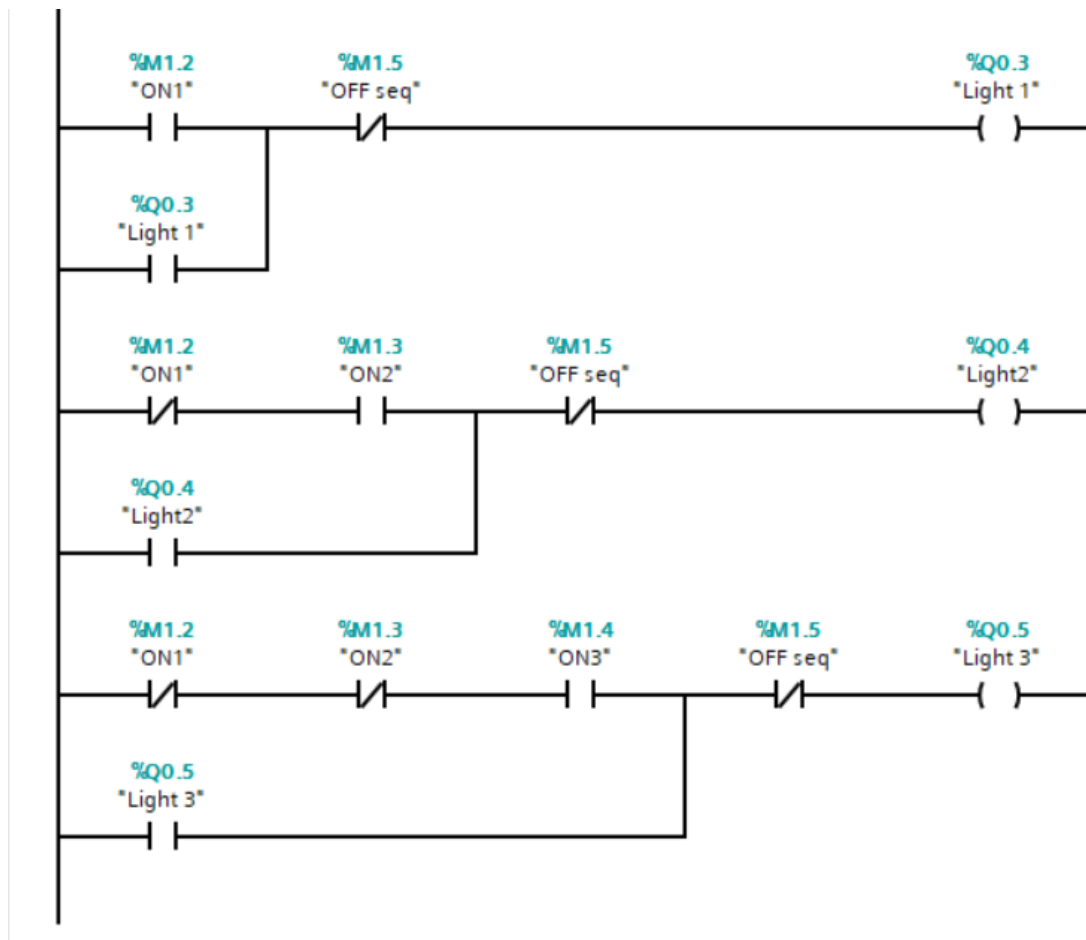


Figure 4.14: Sequencing Lights

b. HMI:

The following tag list is used for the specific lights section in the HMI:

hmi lights				
	Name ▲	Data type	Connection	PLC name
DI	30s light	Bool	HMI_Connectio...	PLC_1
DI	Light 1	Bool	HMI_Connectio...	PLC_1
DI	Light 3	Bool	HMI_Connectio...	PLC_1
DI	Light timer	Time	HMI_Connectio...	PLC_1
DI	Light2	Bool	HMI_Connectio...	PLC_1
DI	sequence button	Bool	HMI_Connectio...	PLC_1
DI	Timed light	Bool	HMI_Connectio...	PLC_1

Figure 4.15: Lighting HMI Tags

These tags are implemented in the buttons shown in the screen below:

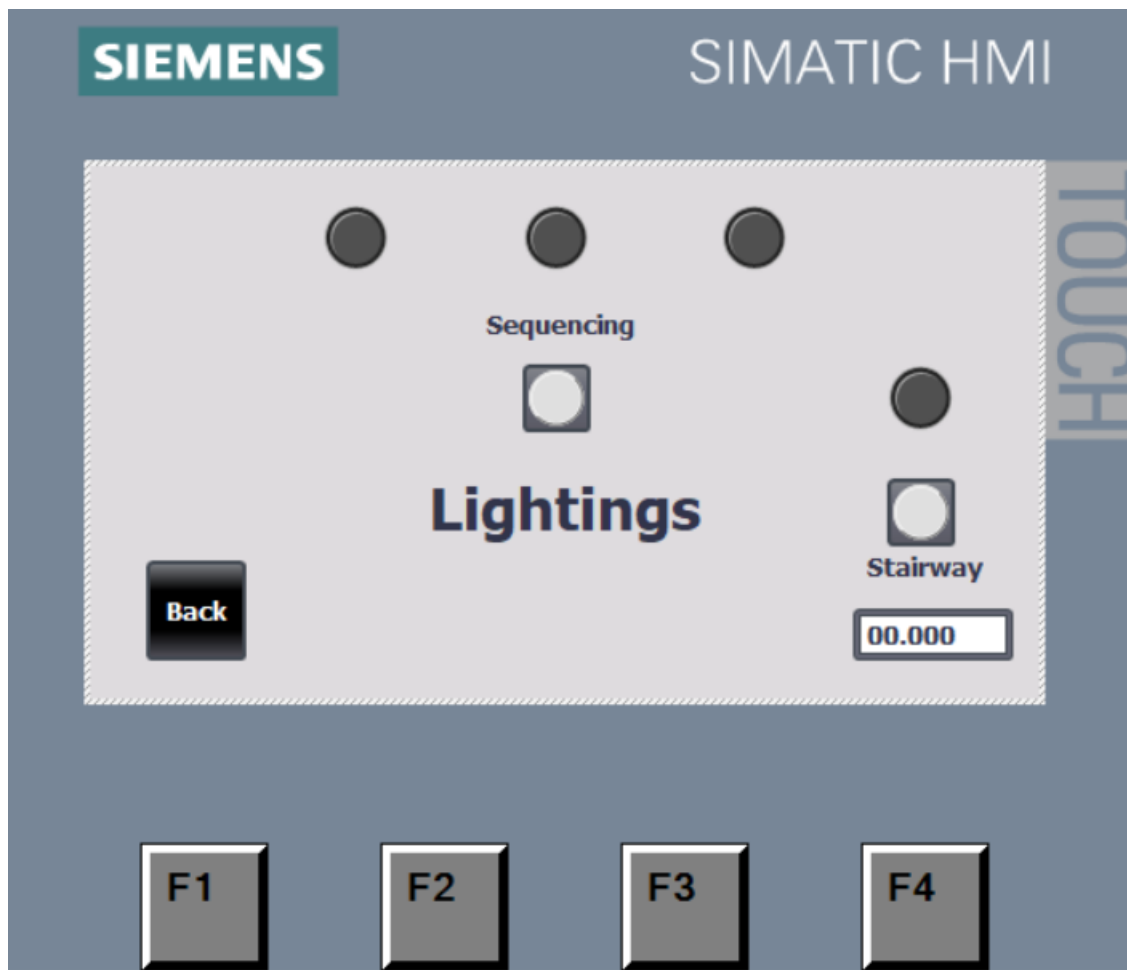


Figure 4.16: Lightings Screen

In the HMI's lights section, the actions and indicators on the screen perform as follows:

- Stairway button: one push of it turns on the timed light.
- Stairway light: Indicates that the timed light is on or off. When off it is black, when on it is green.
- Timer: When the timed light is on, the timer shows us how long it has been on for. The timed light is programmed to be on for 30 seconds, so the timer resets after 30 seconds when the light turns off.

- Lights at the top: These lights indicate light1, light2, and light3 on the kit. These lights are used in the sequencing function and light1 is used in the 2 direction pushbutton function as well. Black when off, white when on.
- Sequencing button: This button triggers the sequencing function explained before.
- Back: Takes screen to Root Screen

4.2.4 Roller Blinds

a. Code:

The below tag list is used for the upcoming roller blinds ladder code. Analog outputs Q1.0 and Q1.1 are used for the roller blinds function, as seen below.





roller							
	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...
1	 Goes up	Bool	%Q1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	 Goes down	Bool	%Q1.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	 UP	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	 Down	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 4.17: Roller Blinds PLC Tags

The roller blinds function is handled in the ladder code below. When the UP button is pushed and the roller blinds aren't down, the roller blinds go up and the function runs for 20 seconds to give enough time for the roller blinds. Same thing happens when the down button is pushed and the roller blinds are not up.

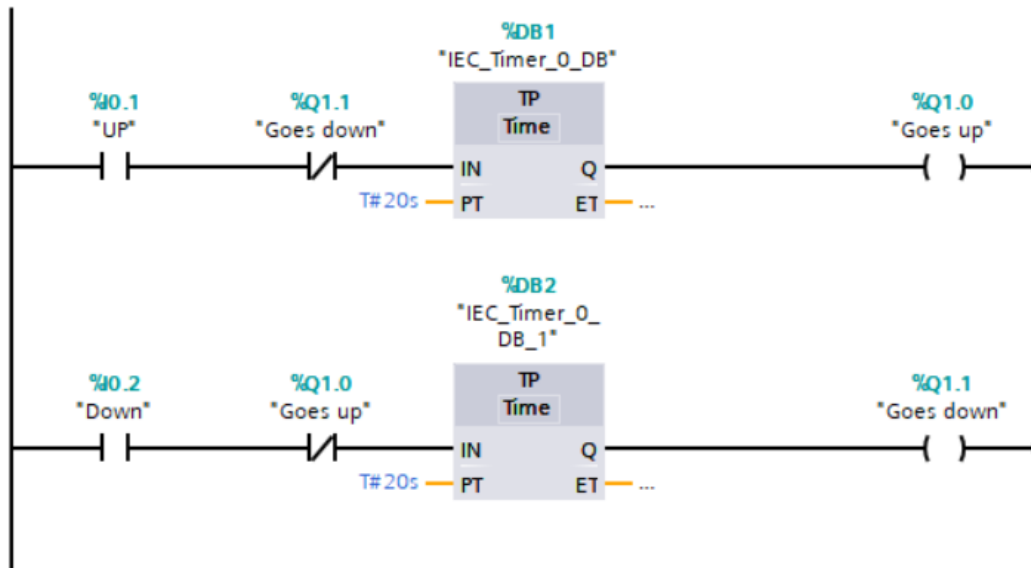


Figure 4.18: Roller Blinds Control

b. HMI:

The following tag list is used for the specific roller blinds section in the HMI:

hmi roller				
	Name ▲	Data type	Connection	PLC name
	Down	Bool	HMI_Connectio...	PLC_1
	Goes down	Bool	HMI_Connectio...	PLC_1
	Goes up	Bool	HMI_Connectio...	PLC_1
	UP	Bool	HMI_Connectio...	PLC_1

Figure 4.19: Roller Blinds HMI Tags

These tags are implemented in the buttons shown in the screen below:

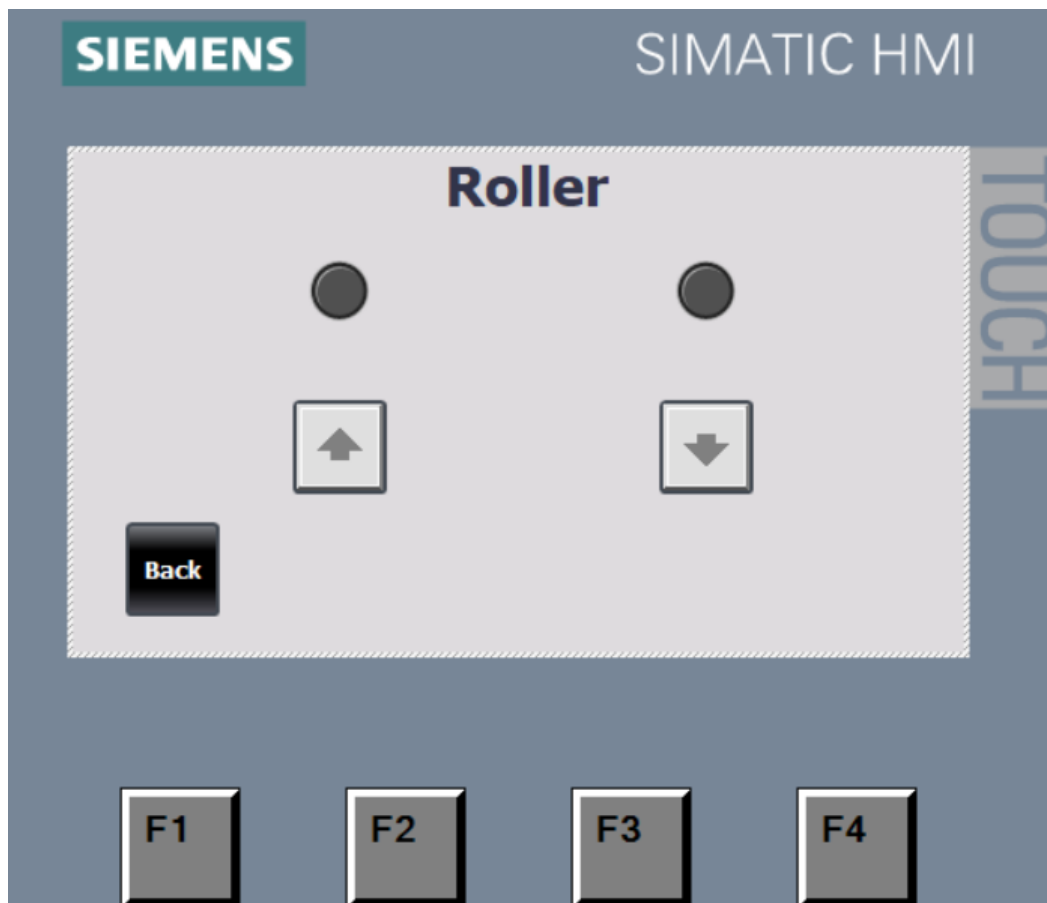


Figure 4.20: Roller Blinds Screen

In the HMI's lights section, the actions and indicators on the screen perform as follows:

- Up Button: Push Button to allow the roller to move up
- Down Button: Push Button to allow the roller to move down
- Black Buttons: Light indicators, which turn white when on, show if the roller is going up or down. Each one represents the button they are above.
- Back: Takes screen to Root Screen

CHAPTER 5: FUTURE WORK

This project allows the control of as many functions as possible using the PROFINET interface of our PLC, the SIMATIC S7-1200. The suggested path for future students who want to continue this work is to utilize the Modbus interface of that same PLC, as we were unable to do that due to the COVID-19 outbreak which significantly shortened our time. This can be done by using Modbus based devices that communicate with the PLC, which in turn would communicate to the HMI through PROFINET. This communication happens through the PLC gateway since the HMI only communicates through PROFINET.

This is done by first establishing Modbus communication using the RS485 interface, which allows the Modbus master, in this case, the S7-1200, to control up to 32 Modbus with the use time-division multiplexing. TIA portal includes an integrated Modbus library that can be used to establish the aforementioned communication between master and slave.

Several modules could be controlled using the Modbus interface, including, and not limited to:

1. Temperature Module
2. Energy Meter
3. Modbus Motor Control
4. Another PLC with Modbus interface

CHAPTER 6: CONCLUSION

The project gave us a look at industrial work while implementing different applications on the TIA Portal V14 program, including motor control for applications such as conveyer belt control and speed control. PROFINET also proved suitable for home automation applications, and not just wider set industrial automation applications. The prototype was implemented on a hardware set of PLC I/O connections, a variable frequency driver, motor control for the first application, a set of lights for the second application, and roller blinds for the roller function. The HMI (Human Machine Interface) gives us the ability to guide through several panels on it to access the different pages of the application.

This application gave us great insight into how it is like to program in the industry, using PROFINET, a very powerful internet protocol that can handle applications to control factory machines and send those machines signals to perform their designated tasks.

LIST OF REFERENCES

- [1] 6AV2123-2DB03-0AX0. (2020). Retrieved 5 June 2020, from <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6AV2123-2DB03-0AX0>

- [2] 6EP3332-6SB00-0AY0. (2020). Retrieved 5 June 2020, from <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6EP3332-6SB00-0AY0>

- [3] 6ES7214-1AG40-0XB0. (2020). Retrieved 5 June 2020, from <https://mall.industry.siemens.com/mall/en/WW/Catalog/Product/6ES7214-1AG40-0XB0>

- [4] 6SL32101PB130AL0. (2020). Retrieved 5 June 2020, from <https://mall.industry.siemens.com/mall/en/us/Catalog/Product/6SL32101PB130AL0>

- [5] 6SL3244-0BB12-1FA0. (2020). Retrieved 5 June 2020, from <https://support.industry.siemens.com/cs/pd/221512?pdte=pi&dl=en&lc=en-DE>

- [6] 6SL3255-0AA00-4CA1. (2020). Retrieved 5 June 2020, from <https://mall.industry.siemens.com/mall/en/ww/Catalog/Product/?mlfb=6SL3255-0AA00-4CA1>

- [7] Allied Electronics & Automation - Industrial Automation & Control Products Distributor. (2020). Retrieved 6 June 2020, from <https://www.alliedelec.com/>

- [8] Fovino I.N., Carcano A., Masera M., Trombetta A. (2009) Design and Implementation of a Secure Modbus Protocol. In: Palmer C., Shenoi S. (eds) Critical Infrastructure Protection III. ICCIP 2009. IFIP Advances in Information and Communication Technology, vol 311. Springer, Berlin, Heidelberg, from https://link.springer.com/chapter/10.1007/978-3-642-04798-5_6
- [9] G. B. M. Guarese, F. G. Sieben, T. Webber, M. R. Dillenburg, and C. Marcon, "Exploiting Modbus Protocol in Wired and Wireless Multilevel Communication Architecture," 2012 Brazilian Symposium on Computing System Engineering, Natal, 2012, pp. 13-18, from <https://ieeexplore-ieee-org.ezsecureaccess.balamand.edu.lb/document/6473625/citations#citations>
- [10] How do you establish MODBUS-RTU communication? . (2017, February 24). Retrieved June 8, 2020, from [https://support.industry.siemens.com/cs/document/47756141/how-do-you-establish-a-modbus-rtu-communication-with-step-7-\(tia-portal\)-for-the-simatic-s7-1200-?dti=0&lc=en-RS](https://support.industry.siemens.com/cs/document/47756141/how-do-you-establish-a-modbus-rtu-communication-with-step-7-(tia-portal)-for-the-simatic-s7-1200-?dti=0&lc=en-RS)
- [11] Kasberger, J. (2011). Advantages of Industrial Ethernet - Comparison of Modbus over TCP/IP and PROFINET (pp. 27 - 30). Retrieved from https://www.auto.tuwien.ac.at/bib/pdf_TR/TR0153.pdf
- [12] Kjellsson, J., Vallestad, A., Steigmann, R., & Dzung, D. (2009). Integration of a Wireless I/O Interface for PROFIBUS and PROFINET for Factory Automation. IEEE Transactions On Industrial Electronics, 56(10), 4279-4287. DOI: 10.1109/tie.2009.2017098
- [13] Modbus-IDA. Modbus application protocol specification v1.1b.3 p. Dec. 2006, from http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf

- [14] PROFINET Field Devices Recommendations for Design and Implementation. (2018, April 18). Retrieved March 28, 2020, from <https://www.profibus.com/index.php?eID=dumpFile&t=f&f=73580&token=eba001cc4c45db026cda45bac8773712309971a7>
- [15] PROFINET System Description Technology and Application. (2018, November 23). Retrieved March 28, 2020, from <https://www.profibus.com/index.php?eID=dumpFile&t=f&f=82430&token=7cbb78f5ba6b3e17762ab594f803f1901eb24fdf>
- [16] Siemens. (2007). Siemens Sinamics G120 PM240 Power Module Manual [Ebook]. Retrieved from <https://inverterdrive.com/file/Siemens-Sinamics-G120-PM240-Power-Module-Manual>
- [17] SIMATIC STEP 7 Prof. V14, floating license download; engineering software in TIA PORTAL. (2020). Retrieved 6 June 2020, from <https://vipausa.com/products/simatic-step-7-prof-v14-floating-license-download-engineering-software-in-tia-portal.html>