

**INTERNATIONAL UNIVERSITY LIAISON INDONESIA**

BACHELOR’S THESIS

**Comparison between Codesys and OpenPLC as a Modbus TCP Protocol Integrated Development Environment (IDE)**

By

**Andika Bramantio Wicaksono**

**11201902005**

In Partial Fulfillment of the Requirements for the Degree of

Sarjana Teknik

In

MECHATRONICS ENGINEERING

FACULTY OF ENGINEERING

BSD City 15310

Indonesia

February 2023

**APPROVAL PAGE**

**Comparison between Codesys and OpenPLC as a Modbus TCP Protocol Integrated Development Environment (IDE)**

**ANDIKA BRAMANTIO WICAKSONO**

11201902005

Presented to the Faculty of Engineering

In Partial Fulfillment of the Requirements for the Degree of

SARJANA TEKNIK

In

MECHATRONICS ENGINEERING

FACULTY OF ENGINEERING

|  |  |  |
| --- | --- | --- |
| Dr. Ir. XXXXXXXXXXXXXXXX  Thesis Advisor |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Date |
| Ir. YYYYYYYYYYYYYYYYYY  Thesis Advisor from Industry |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Date |
| Ir.YXYXYXYXYXYXYXY, M.Sc  Thesis Co-Advisor |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Date |
| Dr. –Ing. ZZZZZZZZZZZZZZZZ  Dean of Faculty of XXXXXXX |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Date |

STATEMENT BY THE AUTHOR

I hereby declare that this submission is my own work and to the best of my knowledge, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at any educational institution, except where due acknowledgment is made in the thesis.

|  |  |  |
| --- | --- | --- |
| Andika Bramantio Wicaksono  Student |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Date |

**ABSTRACT**

Modbus TCP Protocol is one of the protocols it is applicable in many industrial applications systems that make a system communicate with one another. One of the uses of this Modbus TCP protocol is acting as the bridge of communication between a Programmable Logic Controller (PLC), Human Machine Interface (HMI), the controlled machine, and the integrated development environment (IDE).

The term of the PLC IDE that can work with the Industrial Standard is often associated with exclusive or expensive software/equipment. This can be a challenge for the first-time learner for the new generations of automation engineers learning the basics of automation programs.

There are many choices of software that can be selected when choosing which programs that is suitable for learning the PLC. Preferably is the one that can work with many of the current industrial equipment, is easy to use, and one of the most important parts is the accessibility of the software which is at least free-to-use software.

One of the few selected choices that comes into mind when choosing free software that can work with industrial equipment is Codesys and OpenPLC. Both are a powerful program that was designed to be used with many Industrial Equipment and can be equipped with the latest technology in communication and protocol.

*Keywords: Modbus TCP, Industrial Internet of Things, Programmable Logic Controller, Automation, Ladder Logic Diagram, FactoryIO*

© Copyright 2023

by Andika Bramantio Wicaksono

International University Liaison Indonesia

All rights reserved

DEDICATION

I dedicate this thesis to my parents. Without their patience, understanding, support, and most of all love, the completion of this work would not have been possible.

ACKNOWLEDGMENTS

I wish to thank my committee members for their support, patience and good humor. Their gentle but firm direction has been most appreciated. Dr. xxxxxx was particularly helpful in guiding me toward a qualitative methodology. Dr.[[1]](#footnote-1) yyyyyyy interest in sense of competence was the impetus for my proposal.

Finally, I would like to thank my major professor, Dr. zzzzzzzz. From the beginning, he had confidence in my abilities to not only complete a degree but to complete it with excellence.

I have found my coursework throughout the Curriculum and Instruction program to be stimulating and thoughtful, providing me with the tools with which to explore both past and present ideas and issues.

TABLE OF CONTENTS

[APPROVAL PAGE ii](#_Toc34835870)

[STATEMENT BY THE AUTHOR iii](#_Toc34835871)

[ABSTRACT iv](#_Toc34835872)

COPYRIGHT PAGE iv

[DEDICATION vi](#_Toc34835873)

[ACKNOWLEDGMENTS vii](#_Toc34835874)

[TABLE OF CONTENTS viii](#_Toc34835875)

[LIST OF FIGURES x](#_Toc34835876)

[LIST OF ABBREVIATION xi](#_Toc34835877)

[CHAPTER 1 1](#_Toc34835878)

[INTRODUCTION 1](#_Toc34835879)

[1.1. Background 1](#_Toc34835880)

[1.2 *Objective of the thesis* 1](#_Toc34835881)

[1.3 Research Purpose of the Thesis 1](#_Toc34835882)

[1.4 *The Scope of the Thesis* 2](#_Toc34835883)

1.5 *The Limitations of the Thesis* …………..……………………………………………………………… 2

[CHAPTER 2 2](#_Toc34835884)

[LITERATURE REVIEW 2](#_Toc34835885)

[2.3. Xxxxx Yyyyyy Zzzzzz 2](#_Toc34835886)

[2.3.1. Aaaaaa Bbbbbbb Ccccccc 2](#_Toc34835887)

[2.4. Xxxxx Yyyyyy Zzzzzz 2](#_Toc34835888)

[2.4.1. Aaaaaa Bbbbbbb Ccccccc 2](#_Toc34835889)

[CHAPTER 3 3](#_Toc34835890)

[METHODOLOGY 3](#_Toc34835891)

[2.5. Xxxxx Yyyyyy Zzzzzz 3](#_Toc34835892)

[2.5.1. Aaaaaa Bbbbbbb Ccccccc 3](#_Toc34835893)

[2.6. Xxxxx Yyyyyy Zzzzzz 3](#_Toc34835894)

[2.6.1. Aaaaaa Bbbbbbb Ccccccc 3](#_Toc34835895)

[CHAPTER 4 4](#_Toc34835896)

[RESULTS AND DISCUSSION 4](#_Toc34835897)

[5.1. Xxxxx Yyyyyy Zzzzzz 4](#_Toc34835898)

[5.1.1. Xxxxx Yyyyyy Zzzzzz 4](#_Toc34835899)

[5.2. Xxxxx Yyyyyy Zzzzzz 4](#_Toc34835900)

[5.2.1. Xxxxx Yyyyyy Zzzzzz 4](#_Toc34835901)

[CHAPTER 5 5](#_Toc34835902)

[SUMMARY, CONCLUSION, RECOMMENDATION 5](#_Toc34835903)

[5.1. Xxxxx Yyyyyy Zzzzzz 5](#_Toc34835904)

[5.1.1. Xxxxx Yyyyyy Zzzzzz 5](#_Toc34835905)

[5.2. Xxxxx Yyyyyy Zzzzzz 5](#_Toc34835906)

[5.2.1. Xxxxx Yyyyyy Zzzzzz 5](#_Toc34835907)

[REFERENCES 6](#_Toc34835908)

[APPENDICES 7](#_Toc34835909)

[Turnitin Report 7](#_Toc34835910)

[CURRICULUM VITAE 8](#_Toc34835911)

LIST OF FIGURES

Figure Page

1. XXXXXXX………………………

2. XXXXXXX……………………….

3. XXXXXXX ……………….………

4. XXXXXXX….…………….………

5. XXXXXXX…………………………

6. XXXXXXX…………………………

LIST OF ABBREVIATION

LIST OF SYMBOLS

CHAPTER 1

# INTRODUCTION

* 1. **Background**

The Manufacturing Industry in the 21st century relies on certain things. The combination of substantial human resources, computer-oriented integration, and automation. Making the manufacturing operation a sustainable and optimized operation in the company. The main backbone for the machine to be able to be working as it was intended to is the Automation Scripts that run in the background.

There are many varieties of software that can be used in order to make the automation scripts that were used in the industrial-grade standard. Such as the Totally Integrated Automation Portal (TIA Portal), Control Development System (Codesys), CX-Programmer (Omron), etc. However, not all of this software came in handy when it comes to being used by a first-time learner of the programmable logic controller (PLC). It is always for some reasons like it’s not open to the public, or it is way too expensive to be purchased in the first place which is not suitable for learning it the easier way. In this case, an alternative integrated development environment (IDE) is needed to compensate for such a requirement.

This paper aims to compare two of the most popular, free-to-use programmable logic controller IDE that is popularly used to connect with the Modbus TCP Protocol. Codesys and OpenPLC are the two software that is popular in terms of making industrial automation programming system. Both can also establish a Modbus communication that lets IDE, Human Machine Interface (HMI), PLC, and the automated machine communicate with each other using an ethernet-based network.

This paper will provide a comprehensive analysis of the two IDEs, including their advantages and disadvantages, to help readers make an informed decision when selecting an IDE for their automation program and their integrated development through the Modbus protocol.

* 1. **Objective of this Thesis**

The objective of this thesis research is to find out which software is the better alternative IDE for making the automation program. In this case, is a comparison between two free-to-use software. The Control Development System (Codesys) developed by the Codesys Group and OpenPLC developed by the OpenPLC Development Group.

* 1. **Research Purpose of The Thesis**

The purpose of this thesis research is to make a clear difference between free-to-use software that is used in the development of making automation scripts which in this case is the Ladder Diagram/Ladder Logic Diagram (LD/LLD). Although it might be seeming that it has no difference between software usage. But it can be challenging when it is executed, especially when facing difficulties in certain software and its application when applied to real-world manufacturing machinery.

This also aims to help the new engineers/students that might be interested in the automation programming system but might find it challenging when it comes to choosing the first software to be used for first-time learning. Counting from user-friendliness, easiness to use, documentation of the software, and community support.

* 1. **The Scope of the Thesis**

The following is the scope of comparing the Control Development System (Codesys) and the OpenPLC

Codesys and OpenPLC are both IDE for making Ladder Logic Diagrams. This thesis would include the overview perspective from the both IDE and the Modbus TCP protocol. Comparison between their feature

* 1. **The Limitations of The Thesis**
* All of the Factory Simulation would be done in the FactoryIO simulation software and using a custom scene based on the Advanced by Height Scenario.
* The Codesys PLC would be simulated by using the Codesys Control WinV3 x64 virtual PLC and the OpenPLC using the OpenPLC runtime virtual PLC.s
* Communications between the IDE and the FactoryIO would be strictly limited to the Modbus TCP Master/Slave (Server/Client) protocol.

**Chapter 2**

**Literature Review**

**2.1 Modbus Protocol**

Modicon bus communication protocol (Modbus) is a series of communication protocols that were developed by Modicon (now part of Schneider Electric). It has become the standard for the manufacturer to use the Modbus protocol to transmit data from one device to others, for example, PLC, HMI, RTU, and many other smart instruments.

The device can request and transfer data over the serial line of information. In a standard Modbus network, the devices that request the information are called the Modbus Client/Slave, and the device that supplies the information is called the Modbus Server/Master. Some devices also may function as both Client/Slave and Server/Master at the same time.

A Client/Slave device is a device that can process any data or sets of information sent by the Server/Master and later return the output back using Modbus. In the meanwhile, a Server/Master is a device that acts as the host computer running the application software. A Server/Master has the right to assign an address to each individual Client/Slave, initiating broadcast messages to all the slaves in the network. Each Slave is required to respond to all the inquiries and messages that the server has given to individuals. A slave device cannot initiate communication on their own or communicate with each other, they can only respond to inquiries and messages from the Master.

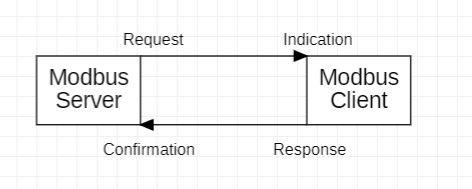
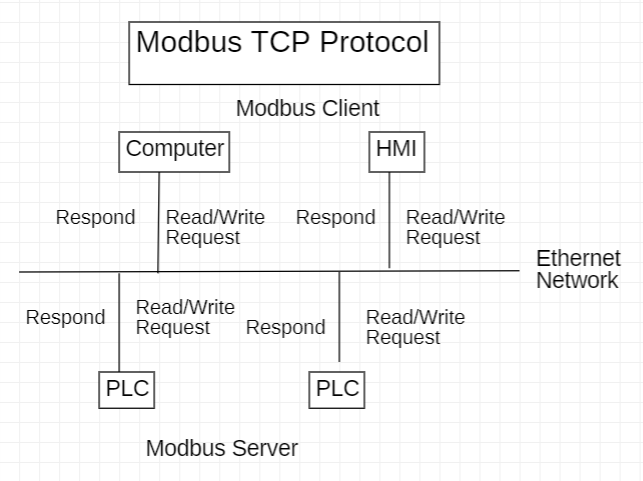


Fig 2.1 Example of Modbus Server and Client Communication

**2.1.1 Modbus TCP/IP Protocol**

The Modbus TCP/IP is a variant of the Modbus Communication Protocol Family that is intended to use as the controller and supervisor over any automation equipment. The Modbus TCP/IP communicates through many of the devices attached by using the Ethernet Network. The most commonly attached device to this Ethernet network is PLC.

The TCP/IP itself refers to Transmission Control Protocol and Internet Protocol. The TCP/IP protocol allows data in form of binary blocks transmitted between devices and ensures that the right device receives the correct all sets of data correctly over the Ethernet network. And it has been the worldwide standard that helps to become the foundation of the Internet or the World Wide Web. However, the Modbus TCP/IP combination is just limited to a transport protocol, which means it does not help define how the data is interpreted, which in this case is the Modbus job as the application protocol.



*Figure 2.2 Modbus TCP Protocol Diagram*

**2.1.2 Usage of Modbus TCP Compared to the OPC UA**

Modbus TCP over the years has been slowly replaced by a newer protocol called Open Platform Communication Unified Architecture or just simply OPC UA. It is a more advanced protocol when compared to only just using Modbus TCP Protocol. It has many advantages that the Modbus didn’t able to do so. Such it can be installed directly on the system or in PLCs, offering more flexibility in terms of device selection, and a more structured data system that allows more efficient and effective data exchanges between one machine to the others.

Although it can offer many strong capabilities Modbus can’t afford to do so. In several cases, selecting the Modbus TCP protocol is still a viable choice. Such as installing OPC UA software and hardware can be very expensive to be installed in the devices. Or for a small manufacturing plant that doesn’t really need expensive equipment yet for their machinery. And compatibility with an older system that might have some problems running OPC UA that might it can be vulnerable since their security system may not have met the modern-day standard for machines security.

**2.1.3 Types of Modbus Protocol Used in the Communication**

Codesys Supports several types of Modbus Configuration:

1. Codesys acting as the Modbus Master
2. Codesys acting as the Modbus Master (Client)
3. Codesys acting as the Modbus Slave
4. Codesys acting as the Modbus Slave (Server)

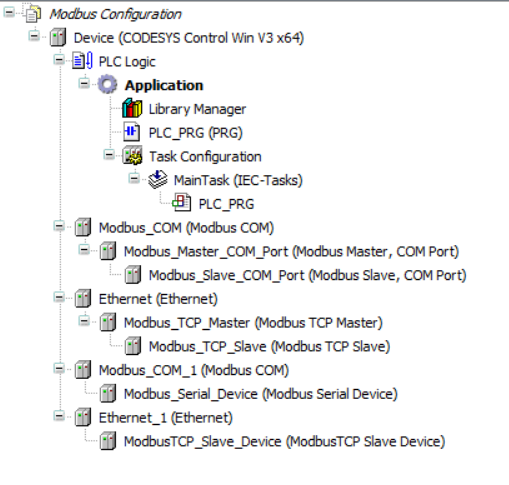


Fig 2.3 Possible Codesys Modbus Configuration

However, for this thesis, Codesys will act as the Modbus Slave (Server)

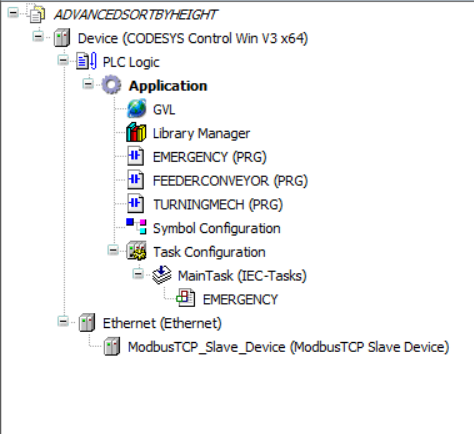
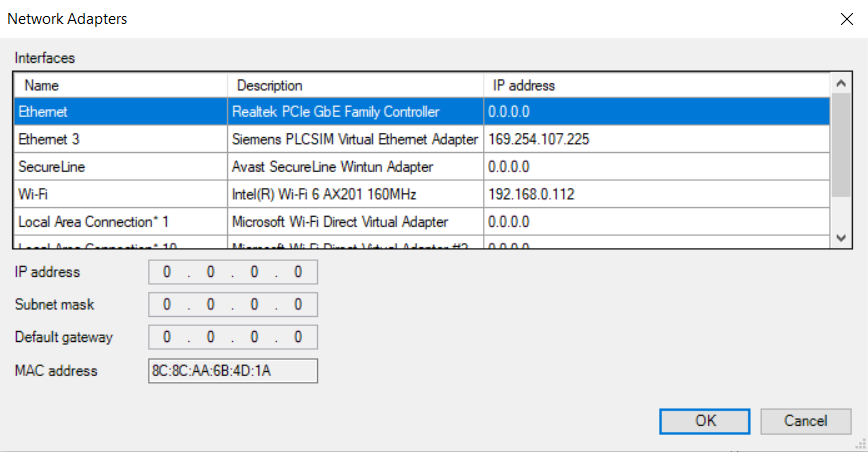


Fig 2.4 Codesys Modbus Slave (Server) Configuration used in the Thesis

The Modbus Slave (Server) configuration is selected because of the simplicity it offers compared to the other configuration available. By only just selecting which network interfaces would be used for the communication and setting up the tags that match the FactoryIO tags.

  
Fig 2.5 Network Adapters Configuration

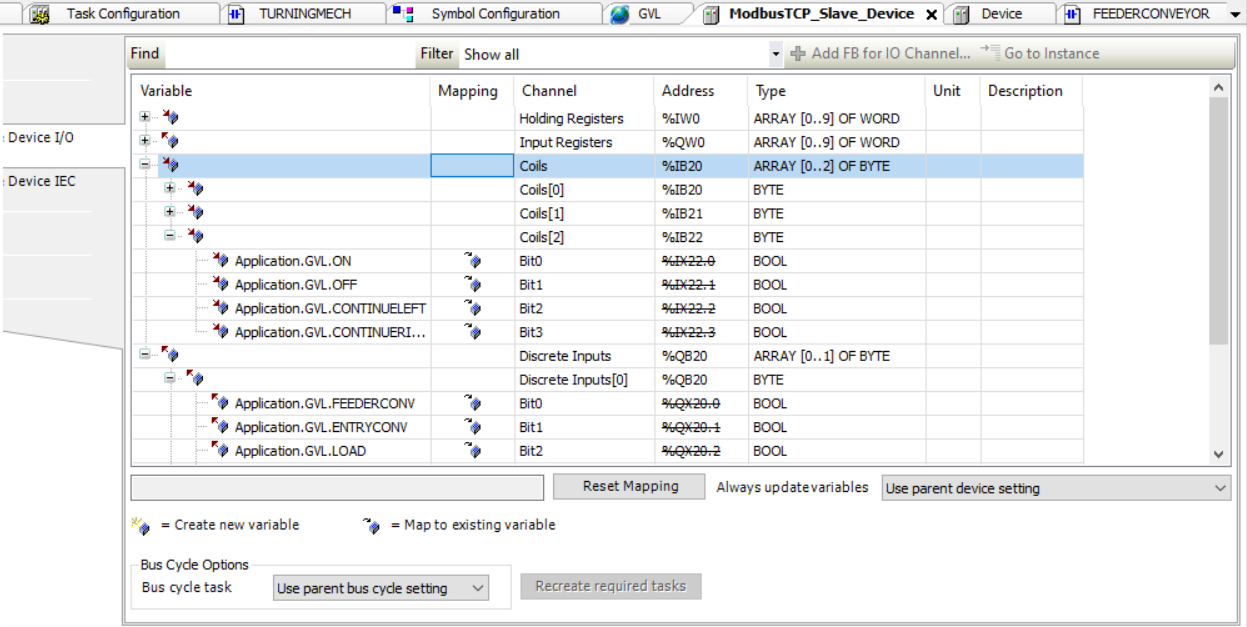


Fig 2.6 Tags Configuration Example

**2.1.4 Modbus Function Codes**

There are 4 types of Objects Type in Modbus Function Codes with each having its own purposes and configuration.

|  |  |  |
| --- | --- | --- |
| **OBJECT TYPE** | **ACCESS** | **SIZE (BIT)** |
| Holding Register | Read-write | 16 |
| Input Register | Read-only | 16 |
| Coil | Read-write | 1 |
| Discrete Input | Read-only | 1 |

Table 2.1 Modbus Function Codes

In the project used for the Thesis, only the Coil and Discrete Input will be used for communication. Since Holding Register and Input Register are only for working for Analog Devices. The Coil represents the value that can be both read or written by the Master device. Meanwhile, Discrete Input represents an either the ON or OFF state of a Sensor

**2.2 Programmable Logic Controller (PLC)**

Programmable Logic Controllers or simply known as PLCs are computer-based single-processor devices that execute an instruction set that is more known as an Electrical Ladder Diagram. It is also capable to take control and managing a wide range of industrial equipment and some of its own type can even take over control of the whole automated systems. PLCs are the main heart and brain of every automatic system in the industry, they are known for being efficient and reliable in industrial applications that involve sequential steps and synchronization between one device with the others.

The Logic in the Programmable Logic Controller is used because the programming language used in the PLC is primarily involving logic and switching operations. The Language that can be implemented in the PLC program is the following:

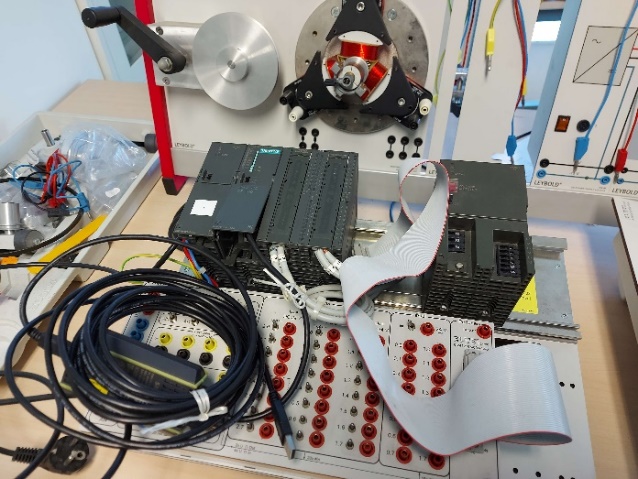
* Ladder Logic Diagram (LLD)
* Structured Text (ST)
* Function Block Diagram (FBD)
* Sequential Function Charts (SFC)
* Instructions List (IL)

PLCs were first used to replace basic mechanical and electrical relay systems in the 1960s. The automation at the time primarily consisted of electromechanical relays and coils that were hardwired on panels. However, this electromechanical relay system can cause a nightmare if just one relay has issues that could affect the whole system’s operation. Nowadays, The Programmable Logic Controller consists of a Power Supply, a CPU, an I/O memory card to store the memories, and a rack where the I/O cards are placed.

**Diagram

Description automatically generated with low confidence**

*Figure 2.7 Example of Basic Ladder Logic Diagram Language for a Basic Roller Conveyor System*



*Figure 2.8 A Siemens S7-300 PLC*

**2.2.1 SoftPLC / VirtualPLC**

A soft PLC is software that can be installed in computer devices that can be used for providing the functionality of a Programmable Logic Controller (PLC). Typically, they’re used to simulate the control and process of an industrial automation system. A soft PLC may also be used as a substitute for physical PLCs. They are also known for versatile and cost-effective solutions for small-scale industrial businesses and for learning material for new students and engineers. Also, SoftPLC offers some advantages over traditional PLC such as flexibility, cost, upgradability, etc.

**2.2.2 Advantages and Restriction of Virtual PLC over Real PLC**

There are several advantages of using a Virtual PLC over a real physical PLC such as:

1. Costs  
   A virtual PLC can be installed in an everyday computer and laptop that people use on a daily basis. Some software even offers free installation (no purchases required beforehand).
2. Flexibility  
   Virtual PLCs tend to be very flexible when compared to their physical counterpart. It can be modified and hacked into the user’s requirement as much as their wish to meet the requirement of the program.

1. Easy to set up and learn

Installing a virtual PLC is as easy as installing regular software on the computer. They also provide a safe environment for learning industrial-grade hardware at zero risking damaging expensive industrial equipment.

Although it has several advantages, Virtual PLC also has several limitations that only Physical PLC possessing it:

1. Simulation Based on Ideal Condition  
   Virtual PLCs simulation is always based on a perfect no error simulated environment. They don’t have the capability of detecting whenever there are physical wiring errors when I/O doesn’t match the assigned address on the computer and the wiring.
2. Performance and I/O limitations  
   When it comes to the performance of Virtual PLC. Their performance can depend on the user’s hardware specification which can lead to results inconsistencies. Also, most of the available Virtual PLCs have limited I/O that they can support at one time. Which can be affected when Virtual PLC tries to communicate with real word devices.

**2.2.3 Ladder Logic Diagram**

Ladder Logic Diagram or Simply LLD is a programming language that instead of using text, uses many combinations of symbolic graphics elements that are called symbols. Originally created to be used by Electricians and Technicians. Since they were made to represent and look like an electrical diagram. Electrical contacts and relays do exist in Ladder Logic (although it is called coil for the relay in ladder logic). The symbol might be different from Electrical Circuit but they have a similar function to do the same job.

Some advantages of Ladder Logic when compared to a similar programming language that is supposed to do the same job:

1. It’s easy to read and understand Ladder Logic Diagram  
   When it comes to readability and the experience during the programming or learning the Ladder Logic. It is easy since the natural reading sequence of most people that goes from the left side to the right side.
2. Structure of the Execution of the Program

The ladder Logic Sequence is going from Left to Right and From Top to Bottom, the order of the programming can make the program precisely determine which instructions to be executed first. From one Ladder to another is separated by a line called rungs that will stack on top of each other that somehow will look like a ladder.

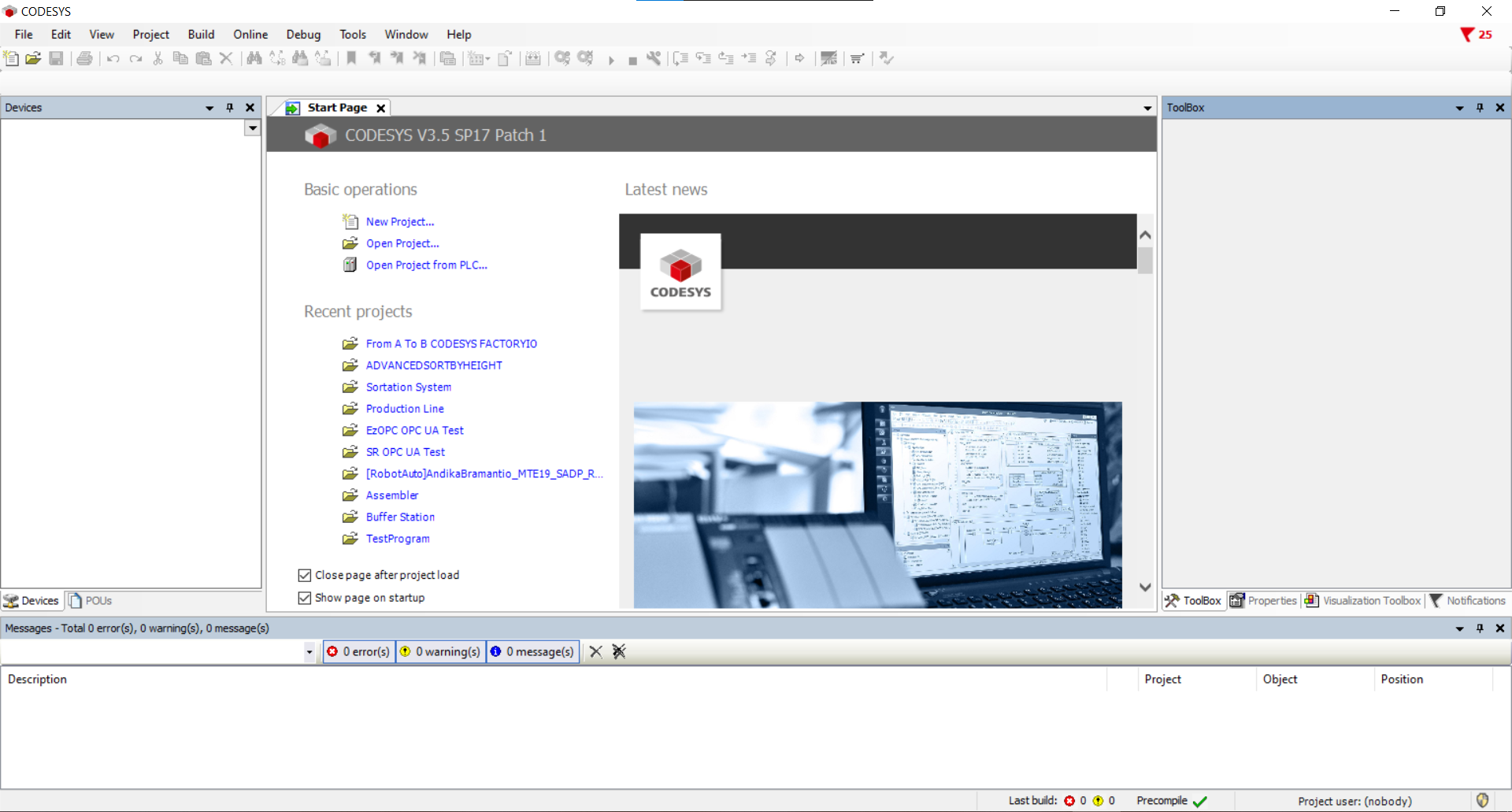
1. Flexibility of Programming  
   Programming Ladder Logic is pretty easy, especially for those who have Electrical Background or ever working with Relay Logic. For example, you can write the program and debug it on a piece of paper first before implementing it on the computer.

**2.3 Software Proposed for the Thesis**

For the current thesis, there are several software that is proposed to be used for the research and development of the current thesis topic.

**2.3.1 Control Development System (Codesys)**

Control Development System or simply known as Codesys is multiple languages, multipurpose integrated development environment (IDE) that is used to program controller and industrial computer programming development. Codesys were developed by 3S-Smart Software Solutions later known as the Codesys Group / Codesys GmbH. This development environment offers user-friendly and free-to-use software to help program the PLCs. However, not all the PLCs can be programmed using this IDE.

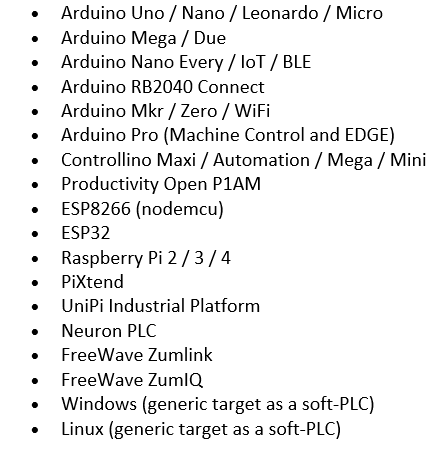
****

*Figure 2.9 The Codesys IDE Home Screen Welcome Page*

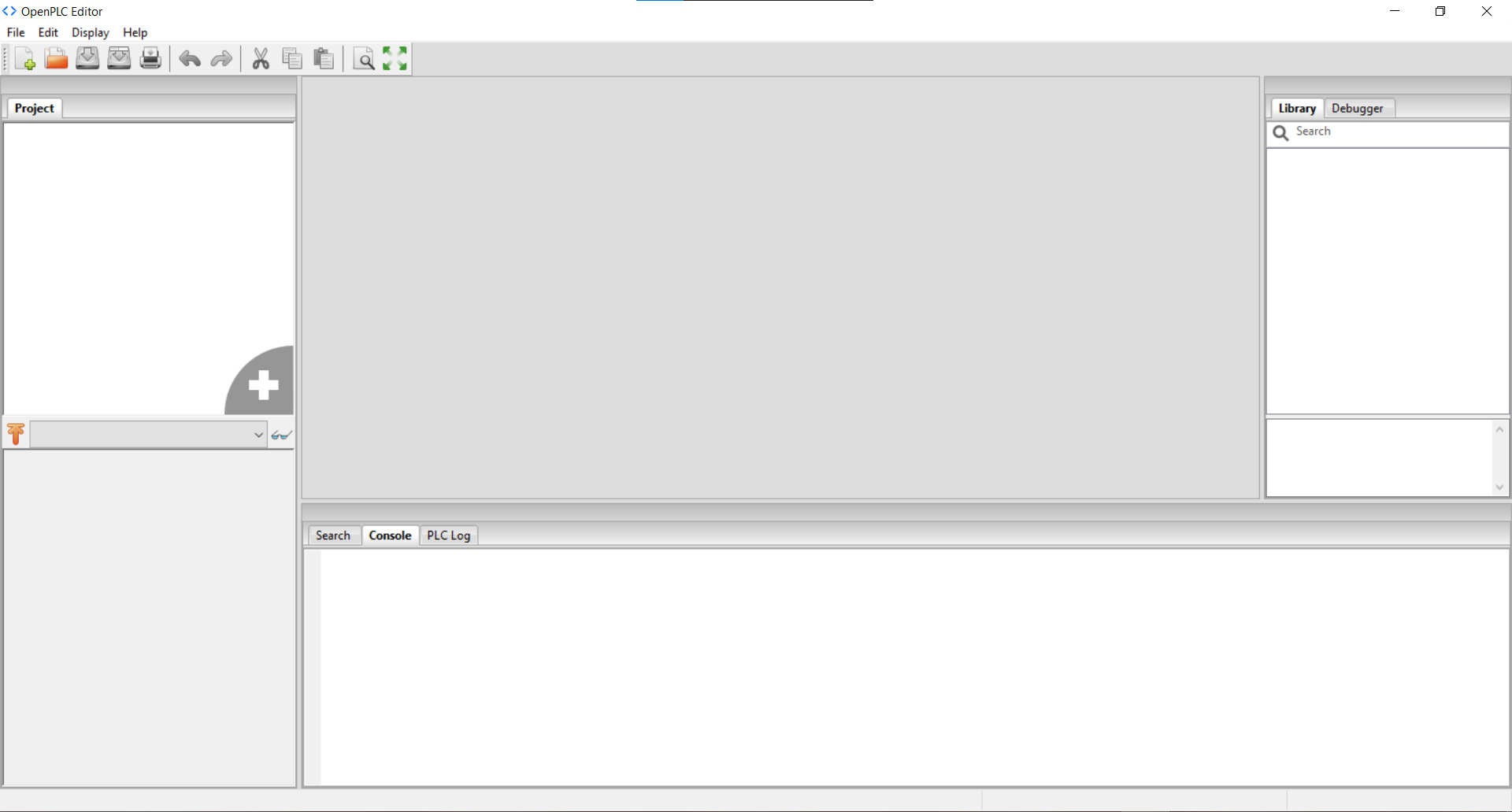
Codesys offers not also provide the IDE environment to program the language to make the automation system. But also provides the virtual PLC that is called Codesys Control Win V3 x64. It simulates and can act like an actual industrial PLC that can be used to run automation simulations.

**2.3.2 OpenPLC**

The OpenPLC is an open-source PLC software that offers multi-language, multi-hardware variations that are based on the Beremiz IDE. It is the first of its kind that is a fully functional and industrial standardized open-source PLC that was created in accordance with the IEC 61131-3 standard that defines basic software architecture and programming languages to the PLC. The Open PLC splits into two parts: The Editor and the Runtime. The Editor is standard software that was used to create the PLC program. Meanwhile, the OpenPLC runtime is a software designed to be able to be executed on many various devices.



*Figure 2.10 List of Devices supported by the OpenPLC Runtime*

**

*Figure 2.11 OpenPLC Editor IDE Home Screen Welcome Page*

For this thesis research project. The Device that will act as the PLC will be the Windows OpenPLC runtime for the simulation and a more realistic case scenario the PLC would be carried out by using a Raspberry Pi 4B.

**2.3.3 Factory IO**

FactoryIO is software that was designed to simulate a 3D environment of a manufacturing plant. It allows the users to create a design, test, and simulate many various industrial processes automated system. The software is designed to be able to work with various devices from different manufacturers such as Siemens, Allen Bradley, MHJ, and even simulated PLC or from a third-party manufacturer through various available protocols.

*s*

*Figure 2.12 A Typical FactoryIO Scene (Production Line)*

**2.3.3.1 Factory IO Drivers**

IO drivers or I/O drivers is a feature of Factory IO that is acting as the relay bridge between the Factory IO and the PLC Controller. Some brands even have built-in driver support like Allen Bradley and Siemens controllers.

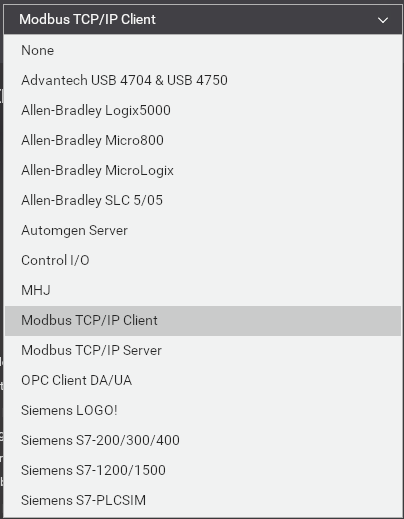


Fig 2.13 List of Drivers supported by FactoryIO version 2.5.2

In order to assign tags to each address in the FactoryIO it is just as simple as Drag and Drop the desired tags to the PLC address available.



Fig 2.14 Tags assignment in FactoryIO

1. [↑](#footnote-ref-1)