**CHAPTER 2**

**LITERATURE REVIEW**

**2.1 Factory I/O**

**2.1.1 Overview**

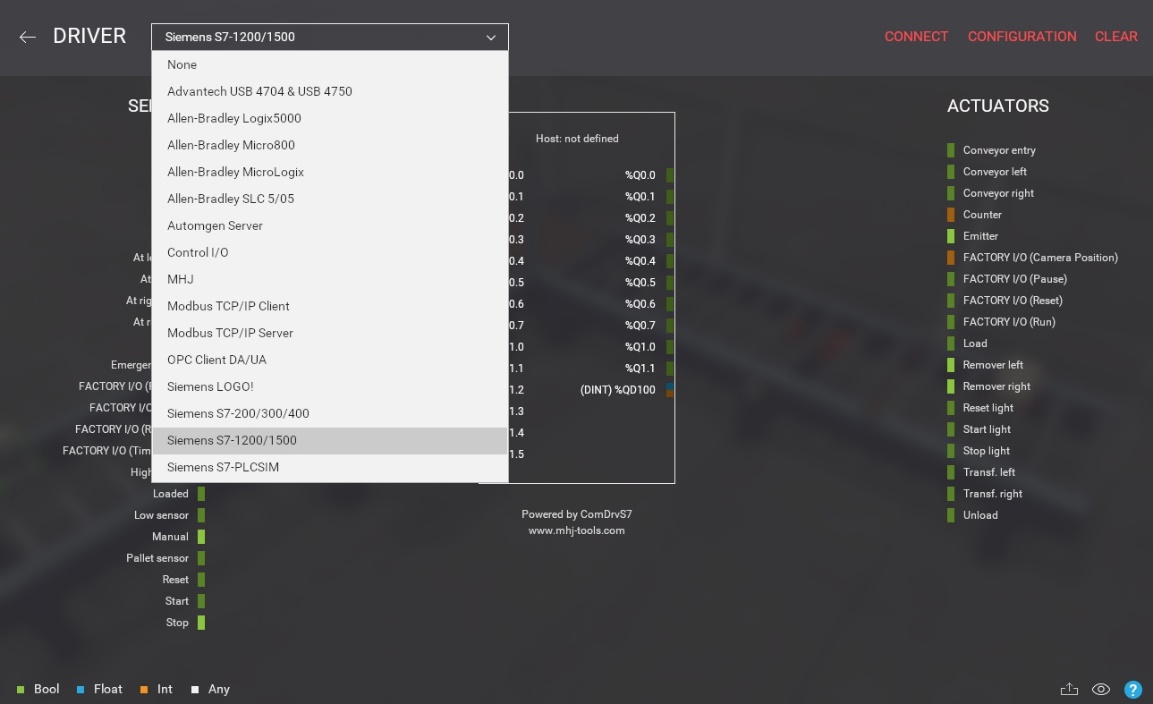
Factory I/O is a 3D factory simulation for learning automation technologies. Designed to be fast and simple to use, allowing you to easily build a virtual factory using a number of different industrial parts. Factory I/O also offers various scenes inspired by typical industrial applications with difficulty level varying from beginner to advanced **[8 Factory I/O ,Factory I/O – Documentation, accessed 30 May 2021,** [**https://docs.factoryio.com/**](https://docs.factoryio.com/) **]**.

Factory I/O is compatible with a variety microcontroller such as softPLC or Modbus, as well as controllers commonly used in industrial applications such as Siemens or Allen-Bradley. Factory I/O is often used as a PLC training platform since PLCs are the most common controllers found in industrial applications **[8 Factory I/O ,Factory I/O – Documentation, accessed 30 May 2021,** [**https://docs.factoryio.com/**](https://docs.factoryio.com/) **].**

**2.1.2 I/O Drivers**

Each one of the controller that compatible with Factory I/O has its own data protocol method. The I/O Driver is a built-in feature of Factory I/O that is responsible for communicating to an external controller with each one on the list is to be used in a specific controller **[8 Factory I/O ,Factory I/O – Documentation, accessed 30 May 2021,** [**https://docs.factoryio.com/**](https://docs.factoryio.com/) **].**

Each communication protocol has their own configuration. In I/O Drivers you can setup the configuration such as define what is the PLC name or IP address at the Host Section, the port number and how many memory address for input or output to be used. I/O Drivers also where sensor and actuator tags are assigned; click and drag the tag to where you desired.



**Figure 2.1** Overview I/O Drivers **[8]**

**2.1.3 Industrial Parts**

Factory I/O provides a collection of parts based on the most common industrial equipment **[8 Factory I/O ,Factory I/O – Documentation, accessed 30 May 2021,** [**https://docs.factoryio.com/**](https://docs.factoryio.com/) **]**. Most of the parts that have been provided by Factory I/O does not have a specific brand. They try to provide the most general shape, function and instruction parts. Factory I/O is not a customizable software. The library for parts could not be added or edited. These parts are organized into eight categories:

|  |  |
| --- | --- |
| **Category** | **Parts** |
| Items | Boxes, Pallets, Stackable Box, Raw Material, Product Lid, Product Base, Final Product |
| Heavy Load | Roller Conveyor, Curved Roller Conveyor, Loading Conveyor,  Roller Stop, Free Roller, Chain Transfer, Turntable, Chute Conveyor Low |
| Light Load | Belt Conveyors, Curved Belt Conveyor, Belt Conveyor Gate,  Inclined Belt Conveyor, Conveyor Scale, Straight Spur Conveyor,  Pivot Arm Sorter, Pop-Up Wheel Sorter, Aligners, Pusher,  Chute Conveyor, Stop Blade, Bracket and Metal Corner, Positioning Bars |
| Sensors | Capacitive Sensor, Diffuse Sensor, Inductive Sensor, Light Array,  Retroreflective Sensor, RFID Reader, Vision Sensor, Incremental Encoder |
| Operators | Emergency Stop, Push Buttons, Light Indicators, Selector,  Potentiometer, Digital Display, Electric Switchboard, Column |
| Stations | Machining Center, Elevator, Pick & Place, Stacker Crane and Rack,  Palletizer, Two-Axis Pick & Place, Tank |
| Warning Devices | Alarm Siren, Stack Light, Warning Light |
| Walkways | Handrails, Platforms, Safeguard, Stairs,Safety Door |

**Table 2.1** List of parts available in Factory I/O

**2.2 Codesys Controller Development System**

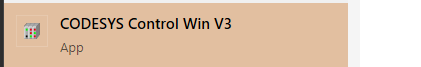
**2.2.1 Overview**

Codesys is an engineering software platform for industrial automation technology. Codesys used the IEC 61131-3 as the core of the platform for programming tool “CODESYS Development System” **[9 Codesys, The Comprehensive Software Suite for Automation Technology, accessed 30 May 2021,** [**https://www.codesys.com/the-system.html**](https://www.codesys.com/the-system.html) **]**. It offers users integrated solutions for the convenient configuration of automatic applications. Codesys is a natural industry 4.0 platform due to its open interfaces, comprehensive security features and easy connection to a cloud based administration platform **[9 Codesys, The Comprehensive Software Suite for Automation Technology, accessed 30 May 2021,** [**https://www.codesys.com/the-system.html**](https://www.codesys.com/the-system.html) **]**.

Codesys has integrated the features for automation components such as integrated products for visualization, coordinate motion control and connection to fieldbus and industrial Ethernet systems.

**2.2.2 Virtual PLC Control Win V3**

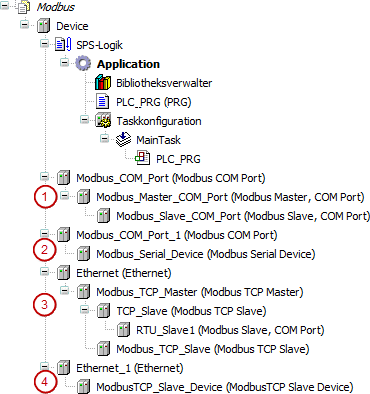
The PLC system in Codesys is embedded in the application. Before the PLC program can running, it is necessary for PCs running windows to start the PLC Program for testing purposes. Using Codesys Control Win V3 as one of the configuration variants of Codesys control **[10 Codesys, Codesys Online Help, version 3.5.17.0, accessed 10 June 2021,** [**https://help.codesys.com/**](https://help.codesys.com/) **]**. The virtual PLC runs as service and can be manually in the computer.



**Figure 2.2** Codesys Control Win V3 application

**2.2.3 Fieldbus Support**

Codesys supports different kind of Fieldbus for the communication protocol, such as: CAN-Based, EtherCAT, EtherNet/IP, Ethernet Adapter, IO-Link, Modbus, PROFIBUS DP, PROFINET, and Sercos. Modbus/TCP is used as the communication protocol for this thesis. The concept of Modbus Master and Slaves can be found in Codesys; one Master and one or more Slaves. Modbus in Codesys can be link into 2 devices, which are Modbus COM Port if through serial port and Ethernet Adapter if through Ethernet Network. Up to 64 Slaves can be connected to the same device **[10 Codesys, Codesys Online Help, version 3.5.17.0, accessed 10 June 2021,** [**https://help.codesys.com/**](https://help.codesys.com/) **]**. The configuration is shown in Figure **2.3:**



**Figure 2.3** 4 different configurations of Modbus in Codesys **[10]**

1. Codesys acts as a Modbus Master
2. Codesys acts as a Modbus Slave
3. Codesys acts as a Modbus Master (client)
4. Codesys acts as a Modbus slave (server)

**2.3 Modbus Protocol**

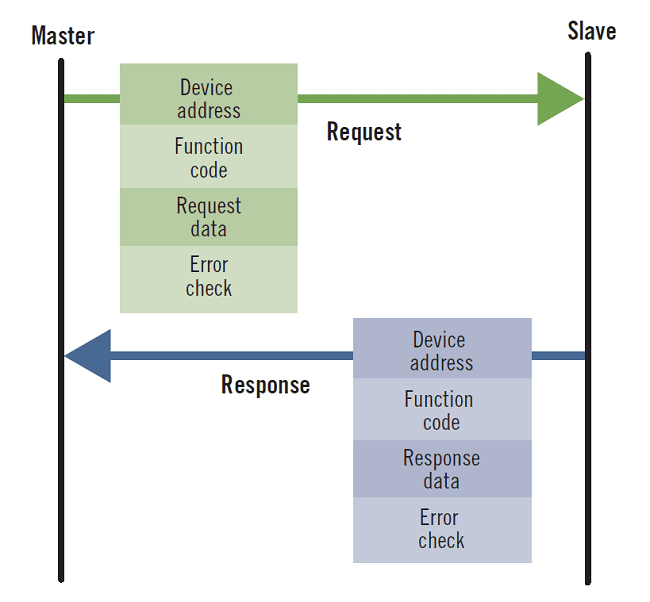
**2.3.1 Overview**

Modbus is a communication protocol for exchanging information between electronic devices through serial lines or via the Ethernet, commonly used in process and industrial automation. Originally, Modbus was published by Modicon in 1979 for its own PLCs **[11 Control Global, January 2019, Introduction to Modbus, accessed 22 June 2021,** [**https://www.controlglobal.com/articles/2019/introduction-to-modbus/**](https://www.controlglobal.com/articles/2019/introduction-to-modbus/) **]**. Currently, the term “Modbus” is a registered trademark of Schneider Electric USA, Inc. Schneider Electric is a partner in the Modbus.org organization, which was formed to increase the development of Modbus.

The Modbus serial protocol is a master/slave communication protocol. One computer is called master and the other one or more devices are the slaves. The master controls the Modbus data transaction with the slaves, which respond to the Master’s requests to read from or write data to slaves and then send it back to the master as its output **[12 Hung (Benjamin) Huang, November 2015, Communication Between Virtual Emulation System and PLC by Modbus/TCP Protocol]**.

**2.3.2 TCP/IP Protocol**

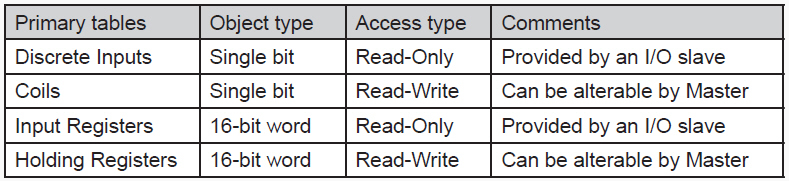
TCP/IP refers to the Transmission Control Protocol and Internet Protocol both of which provide a means of transmission for Modbus TCP/IP **[12 Hung (Benjamin) Huang, November 2015, Communication Between Virtual Emulation System and PLC by Modbus/TCP Protocol]**. Modbus TCP/IP uses between Client-Server or Master-Slave architecture. In a standard Modbus serial network, it has one master and up to 247 slaves, each with its own slave address. Data transfers from a Modbus Client are sent via an IP address to a Modbus Server **[11 Control Global, January 2019, Introduction to Modbus, accessed 22 June 2021,** [**https://www.controlglobal.com/articles/2019/introduction-to-modbus/**](https://www.controlglobal.com/articles/2019/introduction-to-modbus/) **]**. The IP address is to ensure that the data packets are correctly addressed.



**Figure 2.4** Master-Slave Model of Communication **[ 11]**

**2.3.3 Data Model**

Each device must define the objects accessible to the protocol by assigning them a Modbus address and, as a result, must deal with object associations with the effective internal variables in its own memory **[12 Overdigit, June 2017, Modbus protocol over RS485 – Part 4 – Application layer, accessed 22 June 2021,** [**https://web-plc.com/blog/2017/06/01/protocol-data-unit/**](https://web-plc.com/blog/2017/06/01/protocol-data-unit/) **]**. The objects are classified into 4 types, each with a clear connection to the sort of resources that the devices offered during the time when the protocol was defined as it shown in Table **2.2**:



**Table 2.2** Types of data model and its characteristics **[13]**

The term “Discrete Inputs” refers to digital input (ON/OFF) values. The term “Coils” refers to digital output values that can be both writeable and readable. While the term “Input Registers” and “Holding Registers” are analogue versions that are used for analog inputs/outputs or any other numeric value of the device such as a work parameter.

**2.3.4 Function Codes**

The Modbus protocol has various function codes that has been predefined that can perform in specific ways **[13 Codesys, 2006, Modbus Application Protocol, V1b]**. When a Master sends a request to a Slave, the Function Code field instruct the addressed Slave what function to perform depending on the desired function code. These are the function codes that Codesys supports **[10 Codesys, Codesys Online Help, version 3.5.17.0, accessed 10 June 2021,** [**https://help.codesys.com/**](https://help.codesys.com/) **]**:

1. Read Coils (Function Code 01)
2. Read Discrete Inputs (Function Code 02)
3. Read Holding Registers (Function Code 03)
4. Read Input Registers (Function Code 04)
5. Write Single Coil (Function Code 05)
6. Write Single Register (Function Code 06)
7. Write Multiple Coils (Function Code 15)
8. Write Multiple Registers (Function Code 16)
9. Read/Write Multiple Registers (Function Code 23)