VIRTUAL COMMISSIONING
FOR INDUSTRIAL AUTOMATION

A Thesis

Presented to

the Faculty of the College of Business and Technology

Morehead State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Saihiranmitra Mudiki

November 7, 2017

Accepted by the faculty of the College of Business and Technology, Morehead State University, in partial fulfillment of the requirements for the Master of Science degree.

____

Dr. Jorge Alberto Ortega-Moody
Director of Thesis

Master’s Committee: ___________________________________, Chair
Dr. Ahmad Zargari

____
Dr. William R. Grise

____
Date

VIRTUAL COMMISSIONING
FOR INDUSTRIAL AUTOMATION

Saihiranmitra Mudiki
Morehead State University, 2017

Director of Thesis: ____
Dr. Jorge Alberto Ortega-Moody

Most of the recent international manufacturing enterprises have recurred in the last decade to have fully automated and flexible plants to achieve their expected results. However, before the fully operational plant starts working, the process of commissioning represents a very complex step in many aspects, like how incorrect machinery handling, installation, and operation can directly generate a negative effect on the efficiency of the plant; since this equipment is composed of expensive, fragile devices, a wrong implementation of them can significantly delay the ongoing process. Virtual commissioning has been a favorable solution to these issues, saving a high amount of economic resources for both the commissioner and the enterprise.

This research covers the development of Virtual Commissioning with the Virtual Reality scenario along with physical interface by using communication protocol compatible with any class of PLCs.

This study will undertake using these following steps:

Firstly, designed a PCB board communication protocol in a logical layer using Eagle 8.2.2 premium software. This will make it easier to work as a Data Acquisition to transfer data in the same communication layer.

Secondly, making PLC connections by taking power supply 24VDC, PLC inputs/outputs to the microcontroller Digital Outputs and Digital Inputs respectively. The PLC controller uses RSLogix 500 software from Rockwell Automation to create a program and RSLinx Classic Lite to develop a communication with RS-232 in physical layer by selecting the controller model to obtain the program routine to create program and download program to the PLC to turn on Run mode.

Thirdly, Virtual Reality automation plant consists of the Box packaging factory designed in Unity game engine. Also, using 3DS Max to clean all the meshes to easily export as FBX file to Unity will be built in the lab, and an attempt will be made to run and see the whole line of automation in virtual reality using Oculus Rift hardware and controllers with all the functionality.

Fourthly, interfacing between PLC to the Virtual Reality environment by using Microcontroller Serial Communication Port. Upload and Run the programs in all the software's simultaneously in RSLogix 500 PLC programming and play in Unity 3D to observe the functionality of the system.

Finally, test and performance analysis of the developed system. It works according to the desired functionality. So, Implementation of Virtual Commissioning will be achieved. Implementation of Virtual Commissioning of the Box packaging factory line, benefits, economic justification will be demonstrated, and suggestions for designing a PCB board more efficiently in

a way to organize all the components altogether without merging vias with the drills of the components to minimize the size of the board.

Accepted by: ________________________________, Chair
Dr. Ahmad Zargari

____
Dr. Jorge Alberto Ortega-Moody

____
Dr. William R. Grise

ACKNOWLEDGEMENT

This project and the accordingly research was done at the facilities 21st Century Center for Manufacturing Systems, that belong to the Department of Engineering & Technology Management located inside the Morehead State University, Morehead, Kentucky. This research was advised by Dr. Jorge Ortega Moody.

I would like to acknowledge and thank those individuals who aided and supported in completing this thesis. First, I would like to thank my thesis director Dr. Jorge Alberto Ortega-Moody who assisted with the detailed guidance on the design, development of the methodologies, and clarified theory involved in my thesis topic. Thesis committee chairman: Dr. Ahmad Zargari and committee member Dr. William Grise who provided general guidance, and the direction in preparing and completing my thesis, I would like to thank my friends Andres, Enrique, and Tyler to help me throughout the project whenever support needed.

I would like to thank my parents and Linya who always supported me when I had to spend long hours working in the lab. Both were extremely understanding making my work much easier to complete.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 General Background

Modern automation systems are highly integrated and consist of fully automated workstations. A workstation may have robots with tool-changing capabilities, handling systems, storage systems, and computer control system. Since automation systems require a heavy investment, a manufacturing system should be designed so that the long-term profits should remain positive (Chryssolouris, 2013).

Generally, manufacturing systems are dynamic systems and the state changes coincide with the occurrence of various events, thereby exhibiting the characteristics of a discrete event system. The discrete event simulation is among the most popular approaches to the verification of a manufacturing system and has been a powerful tool for calculating utilization statistics, identifying bottlenecks, tracking scheduling errors, and even for creating manufacturing schedules (Schriber, Brunner, & Smith, 2012). If manufacturers are to remain competitive in an ever-changing marketplace, they must continuously improve both the products and the production systems. Thus, an efficient prototyping environment for production systems is crucial, which leads to the notion of the virtual automation system (virtual commissioning), a computer-based environment to simulate individual automation processes (Liu, Suchold, & Diedrich, 2012).

Virtual commissioning enables the full verification of a manufacturing system by performing a simulation involving a virtual plant and a real hardware controller (PLC) (Lee & Park, 2014). This requires the virtual plant model to be fully described at the level of sensors and actuators. Therefore, virtual commissioning is to identify and address design flaws and operational faults without real plant nor controllers so that significant savings can be achieved in the actual implementation of the manufacturing system. A recent study showed the positive effect of working with virtual commissioning, reducing costs and adding weight to value creation in comparison to traditional commissioning (Shahim & Møller, 2016).

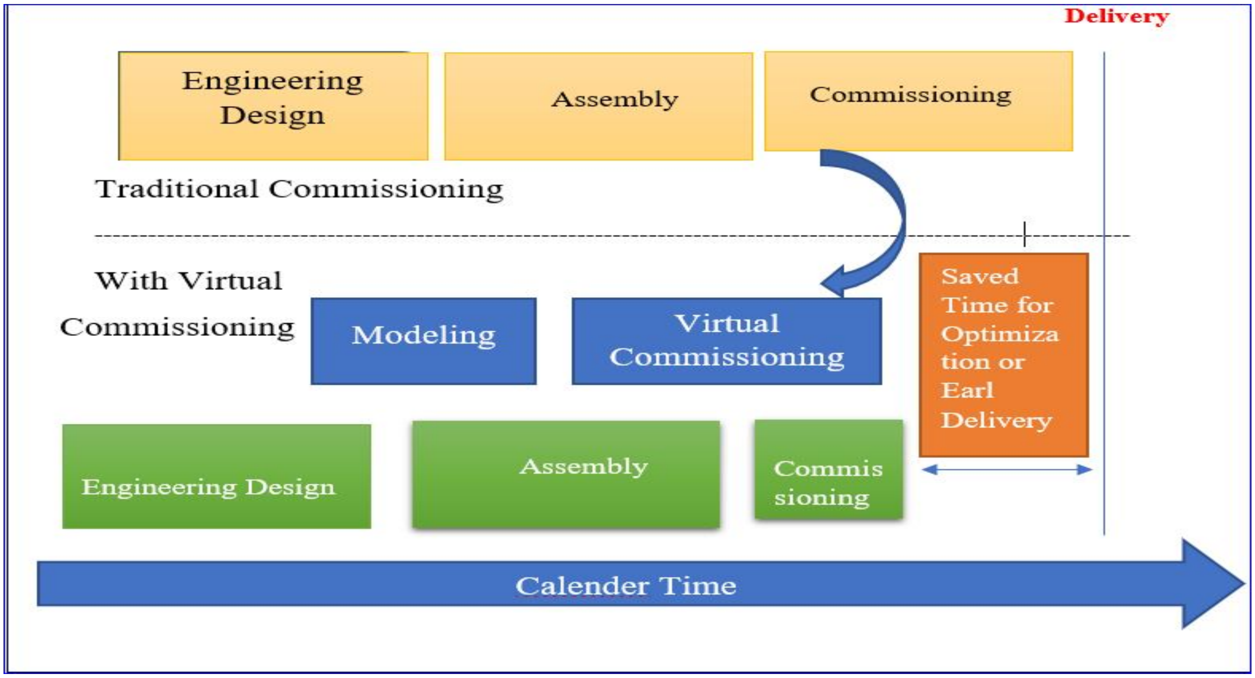


Figure 1. Engineering Project with or without virtual commissioning

(Liu, Suchold, & Diedrich, 2012)

In the last five years, a wide variety of projects have been realized, in which the creation and implementation of virtual reality technologies have helped to improve the plan commissioning process (Süß, Hauf, Strahilov, & Diedrich, 2016). In the same way, some activities like training and industrial machinery maintenance have been replicated in an optimal way thanks to the creation of virtual scenarios (Ortega-Moody, Sánchez-Alonso, González-Barbosa, & Reyes-Morales, 2016).

The way that a physical component is modeled inside a virtual reality scenario is achieved by a very cautious methodology. Not only each element of the plant must be represented as a virtual object, but it must follow a behavior like its real-life counterpart (Süß, Strahilov, & Diedrich, 2015). For this to be possible, computational tools like physics and game engines have started to become the standard requirement in virtual commissioning (Strahilov & Damrath, 2015), since they provide the right environment to simulate a wide range of dynamic properties.

Most of automated manufacturing systems are controlled by a PLC (Programmable Logic Controller), which is currently the most suitable and widely employed industrial control technology. PLCs emulate the behavior of an electric ladder logic diagram (Gonzalez, 2015). PLCs require a programming device, either a computer or console, to upload data on to the CPU. PLCs read signals from different sensors and input devices. These input devices can be keyboards, switches, or sensors. Inputs can be either in digital or analog signal form. Robots and visual systems are intelligent devices that can send signals to PLC input modules. Output devices such as motors and solenoid valves complete the automated system. Nowadays, PLCs have played an important role as part of the plant commissioning process, and as such, they are considered one of the modular elements of virtual commissioning interfaces (Markis, Michalos, & Chryssolouris, 2012).

This thesis details the development of each of these function steps and addresses issues associated with controller system. The effectiveness of the developed Virtual commissioning system is demonstrated to conclude the thesis.

1.2 Problem Statement

In the 21st century, educational institutions and automation industries, there has been a lot of challenges facing particularly in the educational automation laboratories in schools and universities and commissioning time in the industries. A recent study showed the positive effect of working with virtual commissioning, reducing costs and adding weight to value creation in comparison to traditional commissioning (Shahim & Møller, 2016)

1.3 Objectives:

The main purpose of this research has these objectives:

i) To design a PCB and communication protocol to interface between PLC to the Virtual Reality environment.
ii) To design a virtual Reality scenario of the industrial automation plant using Unity 3D 5.5
iii) Interfacing between real control system(PLCs) with the Virtual automated plant (Virtual scenario) using DAQ board(PCB).
iv) Programming in Unity, Arduino, RSLogix 500.
v) Testing and Performance analysis.
vi) Implementation of Virtual Commissioning.

1.4 Definition of Terms:

Commissioning - Commissioning usually the last step in the engineering process, can take up to 15-20% of the total delivery time of an automation system project. Unfortunately, nearly 2/3 of the time spent in commissioning is spent on fixing software errors*, since the control software usually goes through proper integration testing only after all the hardware has been procured and assembled. Commissioning engineers ensure that all aspects of a plant or construction project are properly designed, installed, tested and maintained. They perform troubleshooting tasks, monitor progress, perform tests, conduct audits, assist in financial improvements, write reports and assist clients.

Virtual Commissioning - Emulation for Logic Validation also referred to as virtual commissioning. This process involves replicating the behavior of one or more pieces of hardware with a software environment (typically for a system under design). The goal of the emulation engineer is to create an environment that mimics the real automation hardware. The ultimate goal of emulation is to provide an environment for the manufacturing automation controls engineer to validate their PLC (Programmable Logic Controller) ladder logic and HMI (Human-Machine Interface) files prior to system debug in the plant environment, therefore, improving quality and enabling a seamless transition from the virtual to the physical environment. Another benefit is to deliver plant maintenance operators and machine conductors with realistic virtual environments for training themselves in safe and optimum conditions.

Virtual Reality - Virtual reality is a computer-generated environment that lets you experience a different reality. A VR headset fits around your head and over your eyes and visually separates you from whatever space you're physically occupying. Images are fed to your eyes from two small lenses. Through VR you can virtually hike the Grand Canyon, tour the Louvre, experience a movie as if you are part of it, and immerse yourself in a video game

PLC - A Programmable Logic Controller, or PLC, is an industrial digital computer with a built-in operating system (OS). This OS is highly specialized and optimized to handle incoming events in real time, i.e., at the time of their occurrence.

The PLC has input lines, to which sensors are connected to notify of events (such as temperature above/below a certain level, liquid level reached, etc.), and output lines, to which actuators are connected to effect or signal reactions to the incoming events (such as start an engine, open/close a valve, and so on).

PCB - A printed circuit board (PCB) is the board base for physically supporting and wiring the surface-mounted and socketed components in most electronics.

A printed circuit board (PCB) mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. Components (e.g. capacitors, resistors or active devices) are generally soldered on the PCB. Advanced PCBs may contain components embedded in the substrate (Margaret Rouse, 2013).

PCBs can be single sided (one copper layer), double sided (two copper layers) or multi-layer (outer and inner layers). Conductors on different layers relate to vias. Multi-layer PCBs allow for much higher component density.

Digital Signals **-** A digital signal is a way of transmitting data that converts the data to discrete values, usually based on the binary code that computer systems work upon, which consists of packets of information coded as strings of ones and zeros (wiseGEEK, 2014). Digital signals are not continuous. They use specific values to represent information.

Analog Signals **-** Analog signals are a representation of time-varying quantities in a continuous signal. Basically, a time variance is presented in a way some sort of information is passed using various types of methods, including electrical, mechanical, hydraulic, or pneumatic systems. Unlike digital signals, which use a numeric method of transmitting information, analog signals use small fluctuations in the signal itself to pass information (wiseGEEK, 2014).

Protocol - A set of rules and regulations is called a protocol.

Communication - Exchange of information from one system to another system with a medium is called a communication.

Communication Protocol - A set of rules and regulations that allow two electronic devices to connect to exchange the data with one and another (Agarwal, 2015).

DAQ - Data acquisition (DAQ) is the process of measuring an electrical or physical phenomenon such as voltage, current, temperature, pressure, or sound with a computer. A DAQ system consists of sensors, DAQ measurement hardware, and a computer with programmable software. Compared to traditional measurement systems, PC-based DAQ systems exploit the processing power, productivity, display, and connectivity capabilities of industry-standard computers providing a more powerful, flexible, and cost-effective measurement solution (National Instruments, n.d.).

PWM- Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off.

Vias-In printed circuit board design, a via consists of two pads in corresponding positions on different layers of the board, that are electrically connected by a hole through the board.

Virtual Reality- Virtual reality is the term used to describe a three-dimensional, computer-generated environment which can be explored and interacted with by a person. That person becomes part of this virtual world or is immersed in this environment and whilst there, can manipulate objects or perform a series of action (Rouse, Virtual Reality, 2015).

Augmented Reality-Augmented reality is the integration of digital information with the user's environment in real time. Unlike virtual reality, which creates an artificial environment, augmented reality uses the existing environment and overlays new information on top of it (Rouse, Augmented Reality, 2016).

Chapter 2 Review of Literature

2.1 A Brief Historical Review

Emulation for logical validation also referred to as Virtual Commissioning. This process involves replicating the behavior of one or more pieces of hardware with the software environment (typically for a system under design). The goal of the emulation engineer is to create an environment that mimics the real automation hardware.

It was first pioneered by General Motors in the early 90's, emulation for logic validation has now begun to spawn an industry including software created for the sole purpose of emulation. Companies such as machineering, ISG Virtuos, Emulate3D, Xcelgo, Siemens PLM Software, Visual Components, Dassault Systemes or Irai have developed advanced platforms for virtual commissioning of many different manufacturing and warehouse systems. (Aufderheide, 2015)

2.2 What is Commissioning?

Commissioning is usually the last step in the engineering process before the plant starts production. Commissioning Engineers ensures that all aspects of a building or construction project are properly designed, installed, tested and maintained. They perform troubleshooting tasks, monitor progress, perform tests, conduct audits, assist in financial improvements, write reports and assist clients. (COMMISSIONING SOLUTIONS, n.d.)

2.2.1 Types of Commissioning

i) Retro-Commissioning – Retro-commissioning is a study that seeks to determine how to improve the way building equipment and systems function together. It is the application of the commissioning process to existing plants. Depending on the age of the plant, Retro-Commissioning can often uncover problems which occurred during design or construction. It could also find problems that have developed throughout the plant’s life. In all, Retro-Commissioning allows professionals to offer the best advice on specific ways to improve a plant’s operations and maintenance (O&M) procedures to enhance the building ‘s overall performance. (COMMISSIONING SOLUTIONS, n.d.)

ii) Re-Commissioning - Re-Commissioning is the application of the Commissioning Process applied to projects that have previously been commissioning. During building operations, systems may become out of balance or may be adjusted. As a result, the building may not operate in an efficient manner. Additionally, the needs of a facility may change as technologies change. The Re-Commissioning Process brings a high quality delivered the project to the standards set by either the original or revised Owner’s Project Requirements. A Re-Commissioning Process can typically be applied relatively inexpensively since documentation is available from the original commissioning for the plant. (COMMISSIONING SOLUTIONS, n.d.)

Re-Commissioning entails the examination of actual building system operation and maintenance procedures for compassion to intended or design operation and maintenance procedures.

iii) Continuous Commissioning - Continuous commissioning assures that building systems remain optimized and opportunities for additional improvements are realized. (COMMISSIONING SOLUTIONS, n.d.)

2.3 What is Virtual Commissioning?

Commissioning usually the last step in the engineering process can take up to 15-20% of the total delivery time of an automation system project. Unfortunately, nearly 2/3 of the time spent in commissioning is spent on fixing software errors, since the control software usually goes through proper integration testing only after all the hardware has been procured and assembled. Because commissioning is usually carried out under nearly impossible deadlines, the development of custom control software is often rushed, even so, that debugging it at this stage may cause hardware damage. Resolving these problems earlier on in the process can thus save a great deal of time and effort, which in turn could be spent on optimization or earlier delivery of the project.

Virtual commissioning provides a solution for moving a significant portion of commissioning tasks to an earlier phase of the project, away from the critical path. In virtual commissioning, a simulation model of the system is created to replace the real factory. The virtual factory is then connected to the real control system so that the simulation can be used simultaneously with the procurement and assembly to verify the design and test the control system. This allows for quicker detection of possible errors (VISUAL COMPONENTS, 2015).

2.4 The Benefits of Virtual Commissioning:

There are many benefits to using virtual commissioning throughout the entire engineering process: the total engineering time and prototype waste are reduced; the errors are less expensive to correct since they are detected earlier and the software quality is greatly increased. As an example, in a field study conducted with a small PLC controlled system with 17 sensors and 10 actuators (active components), virtual commissioning was found to increase quality (defined as fulfilment of requirements) from 37% to 84%, while simultaneously reducing the real commissioning time by 75% and total time-to-market by 15%. (Liu, Suchold, & Diedrich, 2012)

By creating visualizations of the proposed system already in the quotation phase, and then building a progressively more comprehensive simulation model during design, you can also achieve far greater synergy between both engineering teams and project stakeholders simply because an interactive 3D simulation conveys the ideas and behavior much better than a stack of design documents ever can. This will help to reduce miscommunication of design goals, desired system behavior, and information requirements between engineering teams as well as the customer. (VISUAL COMPONENTS, 2015)

2.5 What is a control system PLC?

Most of automated manufacturing systems are controlled by a PLC (Programmable Logic Controller), which is currently the most suitable and widely employed industrial control technology. PLCs emulate the behavior of an electric ladder logic diagram (Gonzalez, 2015). PLCs require a programming device, either a computer or console, to upload data on to the CPU. PLCs read signals from different sensors and input devices. These input devices can be keyboards, switches, or sensors. Inputs can be either in digital or analog signal form. Robots and visual systems are intelligent devices that can send signals to PLC input modules. Output devices such as motors and solenoid valves complete the automated system. Nowadays, PLCs have played an important role as part of the plant commissioning process, and as such, they are considered one of the modular elements of virtual commissioning interfaces (Markis, Michalos, & Chryssolouris, 2012).

The researcher used the Allen Bradley PLC Micrologix 1500 LSP C series with one input and one output module.

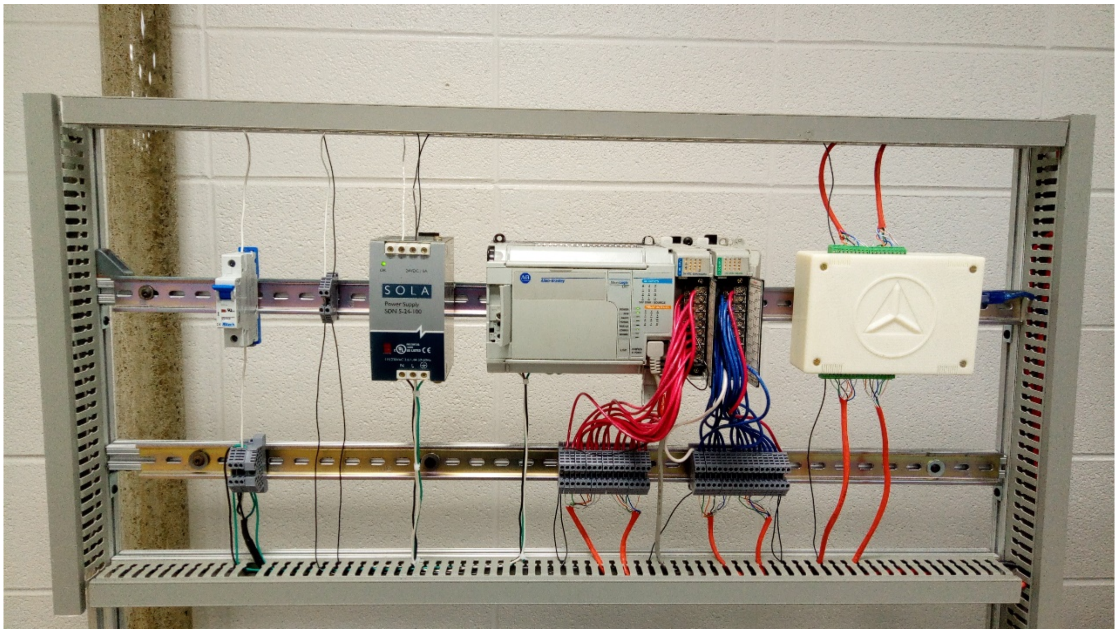


Figure 2. Allen Bradley Micrologix 1500 PLC (MSU Reed Hall Lab)

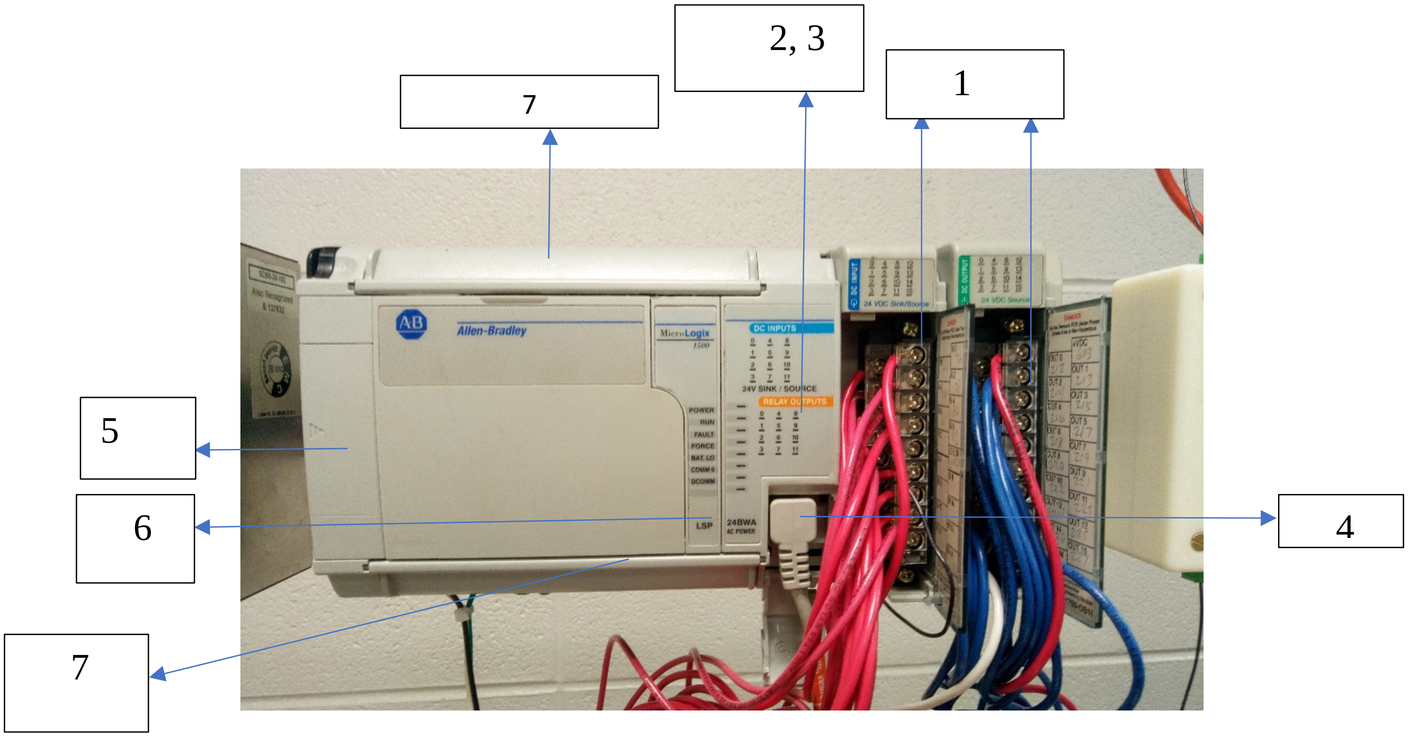


Figure 3. Allen Bradley MicroLogix 1500: Input and output connection diagram

2.5.1 Hardware Overview:

The MicroLogix 1500 programmable controller is composed of a base unit, which contains a power supply, input and output circuits, and a processor. The controller is available with 12 points of embedded I/O including 24VDC power supply, DC Com. Additional I/O may be added using expansion Compact I/O. It consists of LEDs to display activated inputs and outputs. It has a communication port to communicate with the computer, check communication connections in the RSLinx software.

The hardware features of the controller is described below

Table 1. Hardware Features of the Controller

Feature	Description
1	Removable Terminal Blocks
2	Input LEDs
3	Output LEDs
4	Communication Port
5	Remote Run, Program Mode
6	Model 24BWA
7	I/O Slots

Table 2. Micrologix 1500 Component description:

Catalog Number	Line Power	Inputs	Outputs	High Speed I/O
1764-24BWA	120/240 VAC	(8) Standard 24Vdc (4) Fast 24vdc	(12) Relays, 2 Isolated relays per unit	(4) 20 KHZ input

2.5.2 Circuit Breaker

A circuit breaker is an automatically operated electrical switch designed to protect an electrical circuit from damage caused by excess current, typically resulting from an overload or short circuit. Its basic function is to interrupt current flow after a fault is detected.

Figure 4. Circuit breaker

2.5.3 Programming:

Programming the MicroLogix 1500 programmable controller is done using RSLogix 500, Rev. 4.0 or later. Certain features are only available when using the most current version of the software, as noted in System Requirements.

2.5.4 Compact expansion I/O:

In this research, the researcher used Micrologix 1500, LSP 24BWA model. Compact expansion I/O can be connected to the MicroLogix 1500 Controller. A maximum of 16 expansion I/O modules can be used, depending upon system model.

2.5.5 Controller Installation:

This section shows how to install the controller system. The only tools required are a Flat or Phillips head screwdriver and drill to mount on the aluminum din rail (Allen Bradley, 2015).

2.5.6 Installation Considerations:

Most applications require installation in an industrial enclosure to reduce the effects of electrical interference and environmental exposure. Locate your controller as far as possible from power lines, load lines, and other sources of electrical noise such as hard-contact switches, relays, and AC motor drives (Allen Bradley, 2015)

2.5.7 Power Distribution:

There are some points about power distribution to know:

The master control relay must be able to inhibit all machine motion by removing power to the machine I/O devices when the relay is de-energized. It is recommended that the controller remain powered even when the master control relay is de-energized.

When using a DC power supply, interrupt the load side rather than the ac line power. This avoids the additional delay of power supply turn-off. The DC power supply should be powered directly from the fused secondary of the transformer. Power to the DC input and output circuits should be connected through a set of master control relay contacts.

2.5.8 Controller Wiring:

This section describes how to wire the controller (see Table 3. Below)

sinking and sourcing circuits

wiring diagrams, see Figures (5, 6, 7, 8)

2.5.9 Grounding the Controller

In solid-state control systems, grounding and wire routing helps limit the effects of noise due to electromagnetic interference (EMI). Run the ground connection from the ground screw of the base unit to the electrical panel's ground bus prior to connecting any devices. Use AWG #14 wire. This connection must be made for safety purposes (Allen Bradley, 2015).

2.5.10 Sinking and Sourcing Input Circuits

Any of the MicroLogix 1500 DC embedded input groups can be configured as sinking or sourcing depending on how the DC COM is wired to the group.

Table 3. Sinking and Sourcing Input Circuits

Type	**Definition**
Sinking Input: The connection of a PNP sourcing device	The input energizes when a high-level voltage is applied to the input terminal (active high). Connect the power supply VDC (-) to the DC COM terminal
Sourcing Input: The connection of an NPN sinking device	The input energizes when a low-level voltage is applied to the input terminal (active low). Connect the power supply VDC (+) to the DC COM terminal.

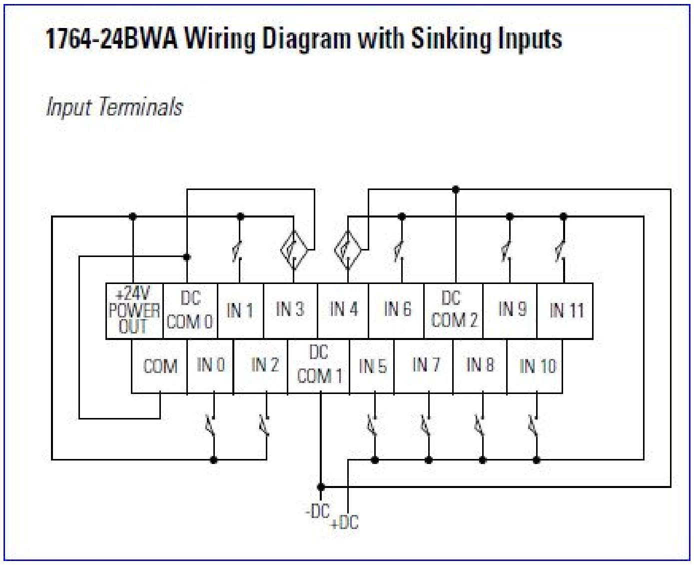


Figure 5. 1764-24BWA Wiring Diagram with Sinking Inputs

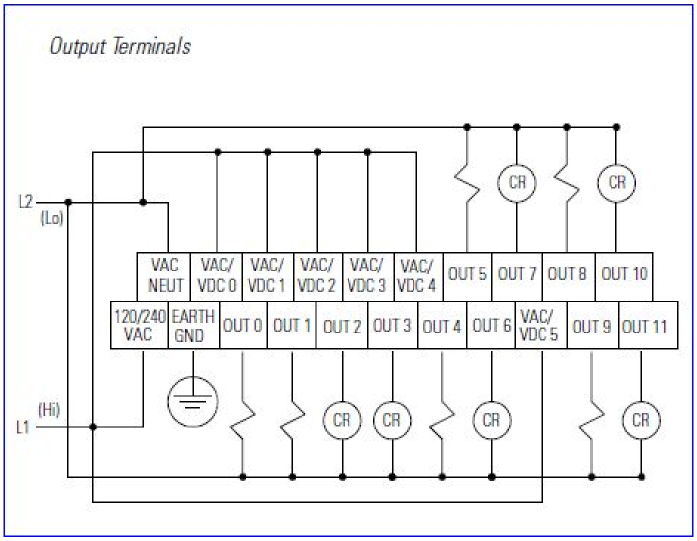


Figure 6. 1764-24BWA Wiring Diagram with Sinking Inputs

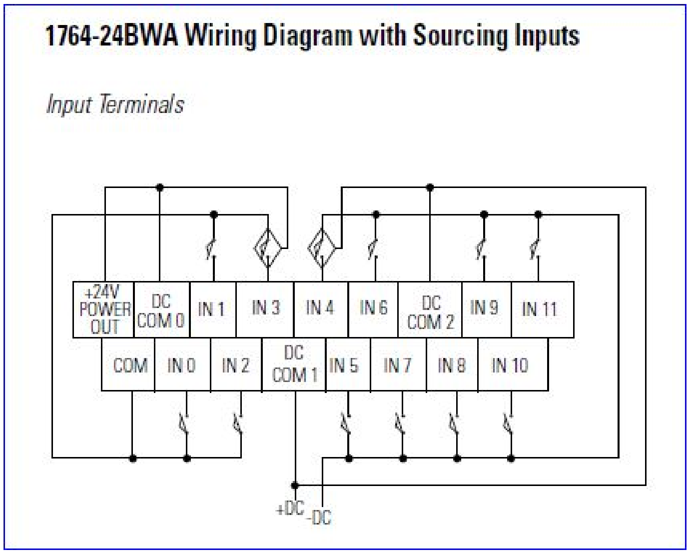


Figure 7. 1764-24BWA Wiring Diagram with Sourcing Inputs

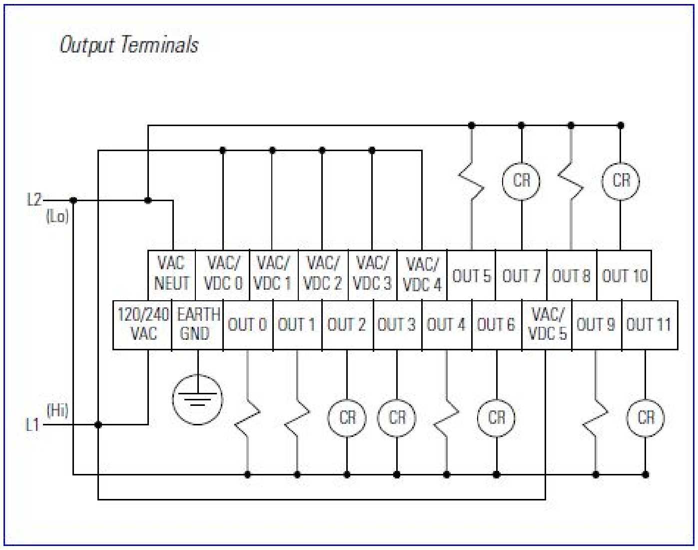


Figure 8. 1764-24BWA Wiring Diagram with Sourcing outputs

2.5.11 Minimizing Electrical Noise

Because of the variety of applications and environments where controllers are installed and operating, it is impossible to ensure that all environmental noise will be removed by input filters. To help reduce the effects of environmental noise, install the MicroLogix 1500 system in a properly rated (i.e. NEMA) enclosure. Make sure that the MicroLogix 1500 system is properly grounded.

A system malfunction may occur due to a change in the operating environment after a period. We recommend periodically checking system operation, particularly when new machinery or other noise sources are installed near the Micrologix 1500 system.

2.5.12 Transistor Output Transient Pulses

The transient energy is dissipated in the load, and the pulse duration is longer for loads with high impedance. The graph below illustrates the relationship between pulse duration and load current. Power-up transients will not exceed the times shown in the graph. For most applications, the pulse energy is not sufficient to energize the load.

To reduce the possibility of inadvertent operation of devices connected to transistor outputs, consider adding an external resistor in parallel to the load to increase the on-state load current. The duration of the transient pulse is reduced when the on-state load current is increased or the load impedance is decreased.

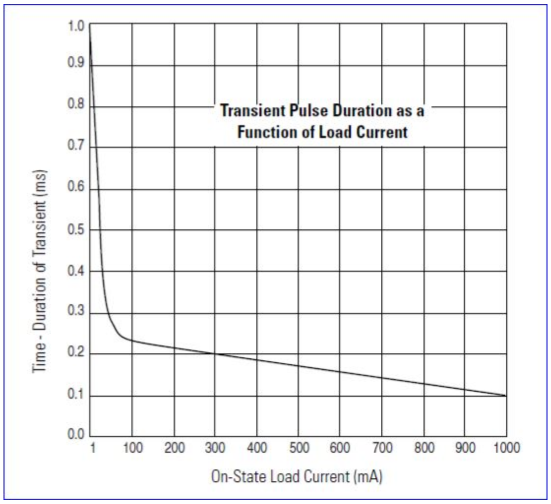


Figure 9. Transistor output Transient Pulse

2.5.13 Communication Connections

This section describes how to set up communications for the control system. The method used in this and cabling required depending on the application. This also describes how the controller establishes communication with the appropriate network.

2.5.14 Connecting to the RS-232 Port

Port 1 DB-9 RS-232 Connections

Port 1 of the DB-9 RS-232, is a 9 pin port that provides connection to port 2 RS-232 8 pin.

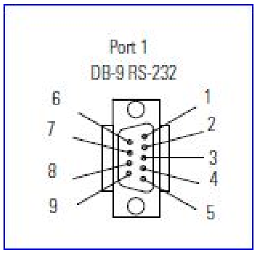


Figure 10. Port1 DB9-RS-232

Table 4. Pins and Ports of DB9-RS-232

Pin	Port 1
1	Received line signal detector (DCD)
2	Received Data (RxD)
3	Transmitted data (TxD)
4	DTE ready (DTR)
5	Signal common (GND)
6	DCE ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	Not applicable

Port 2 of the ENI is an 8-pin mini-DIN RS-232 port that provides connection to DF1 compatible RS-232 devices. The connector pin assignments are shown below.

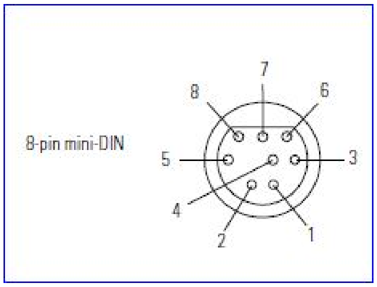


Figure 11. 8-Pin mini-DIN

Table 5. 8 pins and ports of Mini-DIN

Pin	Port 2
1	24V dc
2	ground (GND)
3	no connection
4	ENI Input data, RxD
5	no connection
6	no connection
7	ENI Input data, TxD
8	ground (GND)

2.5.15 Identifying Controller Faults:

While a program is executing, a fault may occur within the operating system of the program. When a fault occurs, you have various options to determine what the fault is and how to correct it.

2.5.16 Automatically Clearing Faults:

Faults can automatically clear by cycling power to the controller when the Fault Override at Power-up bit (S:1/8) is set in the status file. Also, configure the controller to clear faults and press RUN every time the controller is power cycled. This is a feature that OEMs can build into their equipment to allow end users to reset the controller. If the controller faults, it can be reset by simply cycling power to the machine. To accomplish this, set the following bits in the status file:

S2:1/8 - Fault Override at Power-up

S2:1/12 - Mode Behavior

If the fault condition still exists after cycling power, the controller re-enters the fault mode.

2.5.17 Manually Clearing Faults Using the Fault Routine

The occurrence of recoverable or non-recoverable user faults can cause the user fault subroutine to be executed. If the fault is recoverable, the subroutine can be used to correct the problem and clear the fault bit S:1/13. The controller then continues in the Run or test mode.

2.5.18 Specifications

Controller specifications with descriptions along with the model are described in the table below.

Table 6. General Specifications

Description	**1764-BWA**
Number of I/O	12 inputs 12 outputs
Line Power	85 to 265V ac at 47 to 63 Hz
Power Supply Usage	88 VA
User Power Output	24V dc at 400 mA, 400 µF max.

Input Circuit Type	24V dc, sink/source
Output Circuit Type	Relay
Operating Temperature	+0◦C to +55◦C (+32◦F to +131◦F)
Storage Temperature	-40◦C to +85◦C (-40◦F to +185◦F)

2.5.19 Choosing a power supply

This section explains selecting a power supply for applications using a 1764-28BWB base unit. Use the tables in Appendix F to calculate the total power (Watts) consumed by the system. With that information, use the graphs below to choose a power supply. You can use either current or power, depending on how the power supply is rated.

Figure 12. Power Supply 24VDC

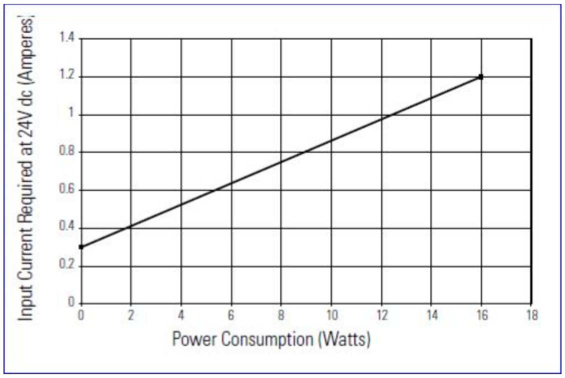


Figure 13. Input Current Required

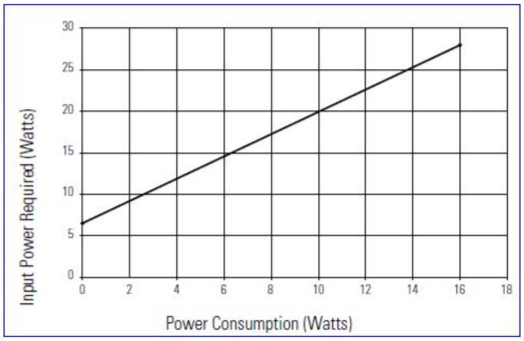


Figure 14. Input Power Required

Table 7. Power Supply Description

Description	**1764 BWA**
On-State Voltage Range	14 to 30.0V dc at 30°C (86°F) 14 to 26.4V dc at 55°C (131°F)
Off-State Voltage Range	0 to 5V dc
Operating Frequency	0 to 20 kHz
On-State Current:	
minimum	2.5 mA at 14V dc
nominal	7.3 mA at 24V dc
maximum	12.0 mA at 30V dc

Off-State Leakage Current	1.5 mA minimum
Nominal Impedance	3.3k ohms

2.6 Data Acquisition Board

Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. Data acquisition systems, abbreviated by the acronyms DAS or DAQ, typically convert analog waveforms into digital values for processing. The components of data acquisition systems include: (Simon S young, 2014)

i) Sensors, to convert physical parameters into electrical signals.

ii)6Signal conditioning circuitry, to convert sensor signals into a form that can be converted to digital values.

ii) Analog-to-digital converters, to convert conditioned sensor signals to digital values.

Data acquisition applications are usually controlled by software programs developed using various general purpose programming languages such as Assembly, BASIC, C, C++, C#, LabVIEW etc.

2.7 Digital Signals:

A digital signal refers to an electrical signal that is converted into a pattern of bits. Unlike an analog signal, which is a continuous signal that contains time-varying quantities, a digital signal has a discrete value at each sampling point. The precision of the signal is determined by how many samples are recorded per unit of time. For example, the illustration below shows an analog pattern (represented as the curve) alongside a digital pattern (represented as the discrete lines).

A digital signal is easily represented by a computer because each sample can be defined by a series of bits that are either in the state 1 (on) or 0 (off). Digital signals can be compressed and can include additional information for error correction.

Figure 15. Digital Signal (learnsparkfun.com)

In this research, digital input signals are formed by using Phototransistor Output Optocouplers with 1K ohm resistor at the input side and 560 Ohm resistor at the output side of the optocouplers connected from PLC outputs.

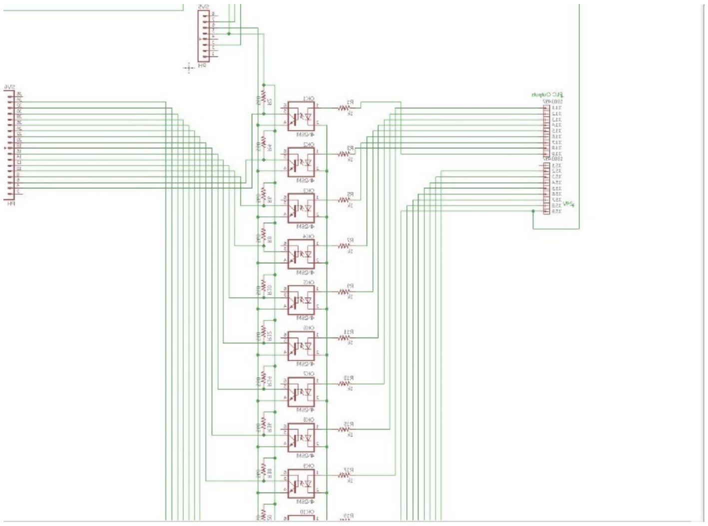


Figure 16. Digital Inputs(Eagle Schematic)

The above figure shows the Digital inputs circuit design in the Eagle software. It has terminal pins in which PLC output signals are connected then it goes to resistor of 1Kohm is connected to the photoresistor input pin (see pin description above) and -24V from PLC connected to pin 2 of the optocoupler, it gets 24V in the inputs, in the output pin 4 it is connected to GND and pin 5 is connected to 560Kohm resistor to go the input pins of the microcontroller.

Photoresistor has 6 pins:

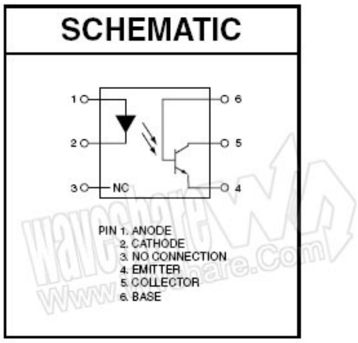


Figure 17. Optocoupler 4N26 (waveshare.com)

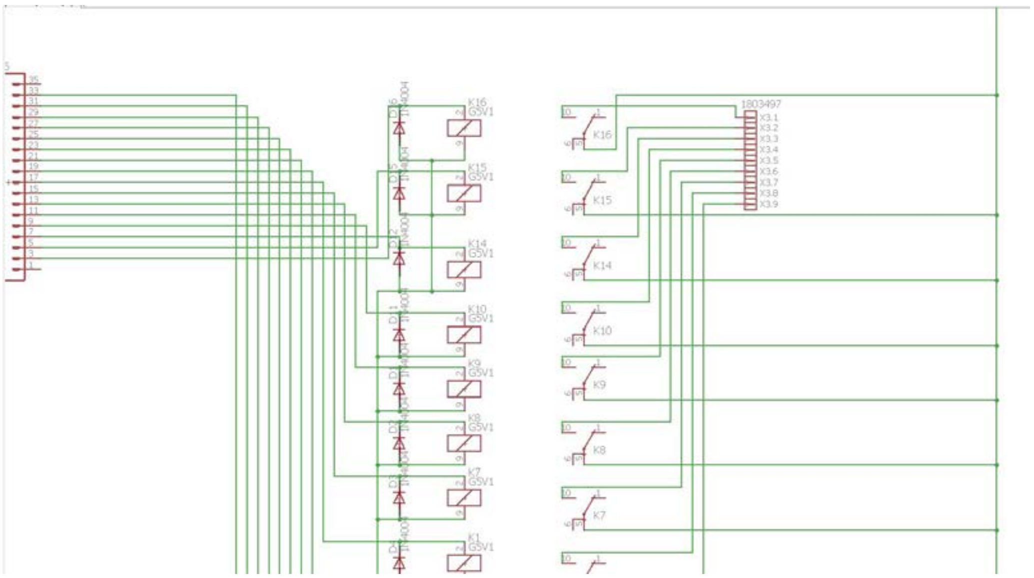


Figure 18. Digital Outputs (Eagle Schematic)

The above figure shows the Digital Outputs in it has 5V Relay coil, each of the microcontroller's output is connected to each 5V Relay (See pin diagram below), the input of the relay has 5V and it sends out the 24V to the PLC inputs. All these communications work in the logical layer.

The microcontroller is then connected to the computer in it has Unity 3D interface for communication with the PLC.

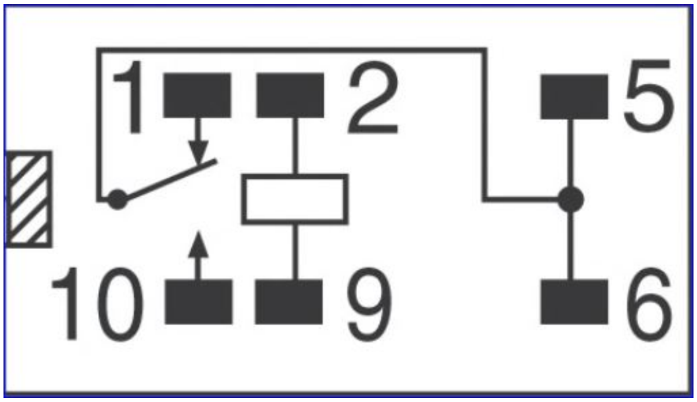


Figure 19. Relay 5V (waveshare.com)

2.8 Analog Signal:

An analog signal is a continuous signal that contains time-varying quantities. Unlike a digital signal, which has a discrete value at each sampling point, an analog signal has constant fluctuations. The illustration below shows an analog pattern (represented as the curve) alongside a digital pattern (represented as the discrete lines).

An analog signal can be used to measure changes in some physical phenomena such as light, sound, pressure, or temperature. For instance, an analog microphone can convert sound waves into an analog signal. Even in digital devices, there is typically some analog component that is used to take in information from the external world, which will then get translated into digital form (using an analog-to-digital converter).

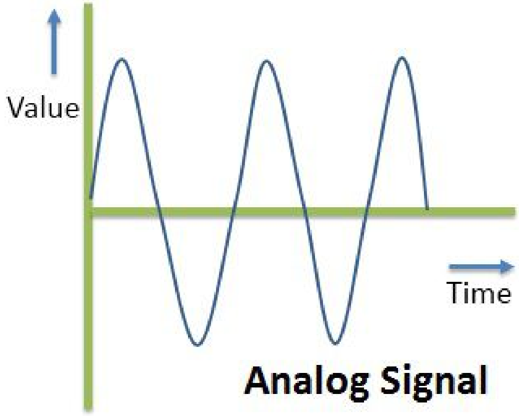


Figure 20. Analog Signal(learnsparkfun.com)

2.8.1 Active Low Pass Filter:

The main disadvantage of passive filters is that the amplitude of the output signal is less than that of the input signal, i.e., the gain is never greater than unity and that the load impedance affects the characteristics of the filter.

Active Filters contain active components such as operational amplifiers, transistors or FET's within their circuit design. They draw their power from an external power source and use it to boost or amplify the output signal.

Filter amplification can also be used to either shape or alter the frequency response of the filter circuit by producing a more selective output response, making the output bandwidth of the filter narrower or even wider. Then the main difference between a “passive filter” and an “active filter” is amplification. An active filter generally uses an operational amplifier (op-amp) within its design and in the Operational Amplifier has a high input impedance, a low output impedance and a voltage gain determined by the resistor network within its feedback loop. The simplest form of a low pass active filter is to connect an inverting or non-inverting amplifier, to the basic RC low pass filter circuit as shown.

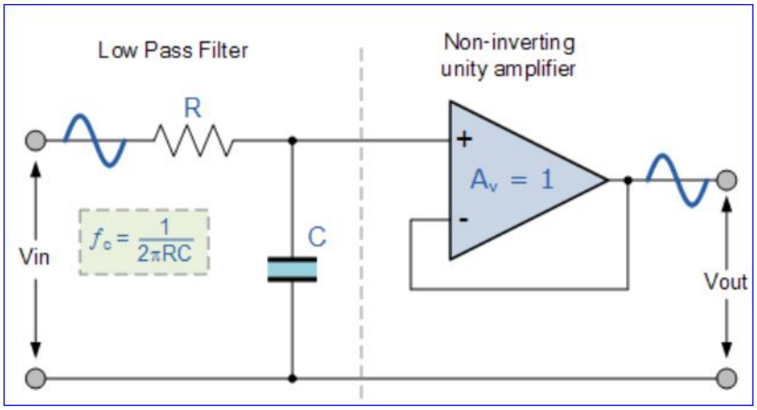


Figure 21. Active Low Pass Filter(learnsparkfun.com)

$$Input\ =\ 24V;\ Output\ should\ be\ able\ to\ get\ 5V.$$

$$Input\ =\ Output$$

$$24V = 5V$$

$$For\ 1V\ =?$$

$$24V * X = 5V * 1$$

$$X = \frac{5V}{24V} = 0.2$$

$$Vo = \frac{Vin\ (R1 + R2)}{R1}$$

$$R1 + \frac{250K}{R1} = 0.2$$

$$1 + \frac{250K}{R1} = 0.2$$

$$Vout = Vin\left(\frac{63k}{303k}\right) = 0.20$$

$$Vout = 24v * 0.2 = 4.8v \cong 5V$$

$$Av = \frac{Vout}{Vin} \quad [\,Av = Voltage\ Gain]$$

$$Av = 1 + \frac{R2}{R1} \quad [R1 = 250kohm, R2 = 250kohm]$$

$$= 1 + \frac{1}{1} = 1 + 1 = 2$$

$$Voltage\ Gain\ Av = 2$$

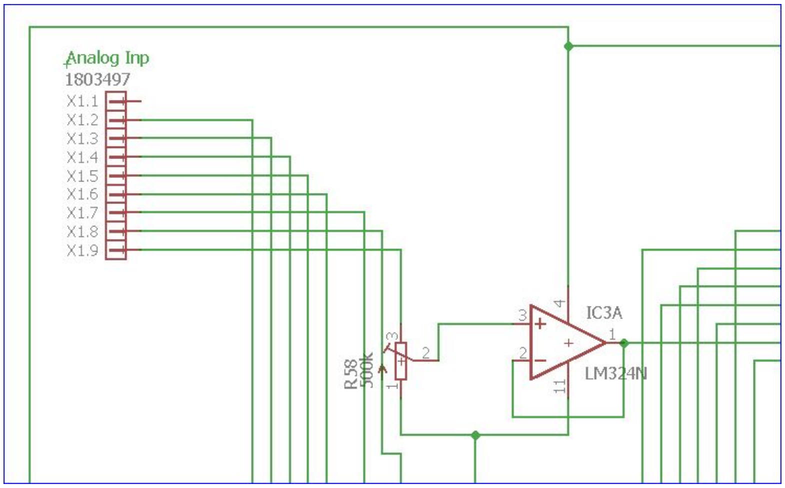


Figure 22. Analog Input (Eagle Schematic)

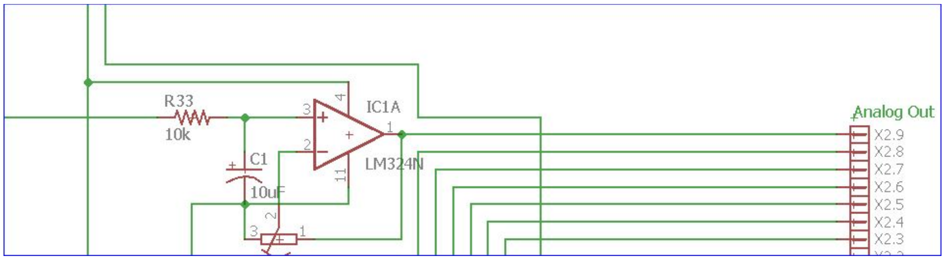


Figure 23. Analog Output Signals (Eagle Schematic)

2.9 Previous Studies

Virtual Commissioning of automated systems Concepts for the digitalization of products and all the production related tasks in the manufacturing and process industry have been developed for several decades. The development began with the introduction of computer-based 2D and 3D.

The Researcher describes below are the previous research studies related to Virtual Commissioning.

Table 8. Previous Research Studies

Researcher	**Area of Study**	**Results**
(Makris, Michalos, and Chryssolouris 2012b)	VC of an assembly Cell with cooperating Robots	Ramp-up time reduction, affecting the total installation time by 15–25%. Reduction in investment costs by decreasing to the minimum. The cost is reduced even more through the reduction by up to 15% of the human resources, required for troubleshooting during the ramp-up process. Enhancement of the reconfigurability of assembly

(Hloska and Kubín 2014)	VC of mechatronic systems with the use of simulation	Shortening of the process of Standard Operating Procedure (SOP). Possibility to carry out part of the SOP in a more convenient environment (not necessarily on site) combined with the opportunity to use the emulation model for training of workers. Parallel development and optimization of mechanical parts, especially mechatronic mechanisms. Simultaneous programming and debugging of the control software (of MFC or individual PLCs). (Hloska, 2014)
(Reinhart and Wünsch 2007)	VC to mechatronic production systems	A significant amount of time and cost can is saved in the production ramp-up process. Delivering a system with a higher software quality for startup.
(Møller, Chaudhry, and Jørgensen 2008)	A virtual enterprise architecture for logistics service	Testing and bug finding under controlled testing environments. Validation of controls software for automation, early identification of errors and actions taken early in the development process. High visibility of quality state of the software during the development process-reveal problems and opens for action-taking in the early process. Continuous test in parallel with installation while the real system is occupied by other activities.

		Low cost of future developments and changes.
(Seidel, Donath, and Haufe 2012)	Simulation and VC environment for controls of material handling systems	The additional work of modeling the material handling process in the simulator was compensated with a reduced commissioning stage on-site. Intense testing of PLC and Material Flow Controller (MFC) programs significantly boosted the software's maturity and reduced the required time on-site to 25% of the planned time.
(Drath, Weber, and Mauser 2008)	VC for Engineering Design Life Cycle	A seamless re-usability of VC models through the complete engineering life cycle.
(Ko, Ahn, and Park 2013)	VC between a real controller and a virtual plant in a production system	Saving of the delays in time to market.

Chapter 3 Methodology

The proposed methodology (see figure 24) is composed of the stages described below:

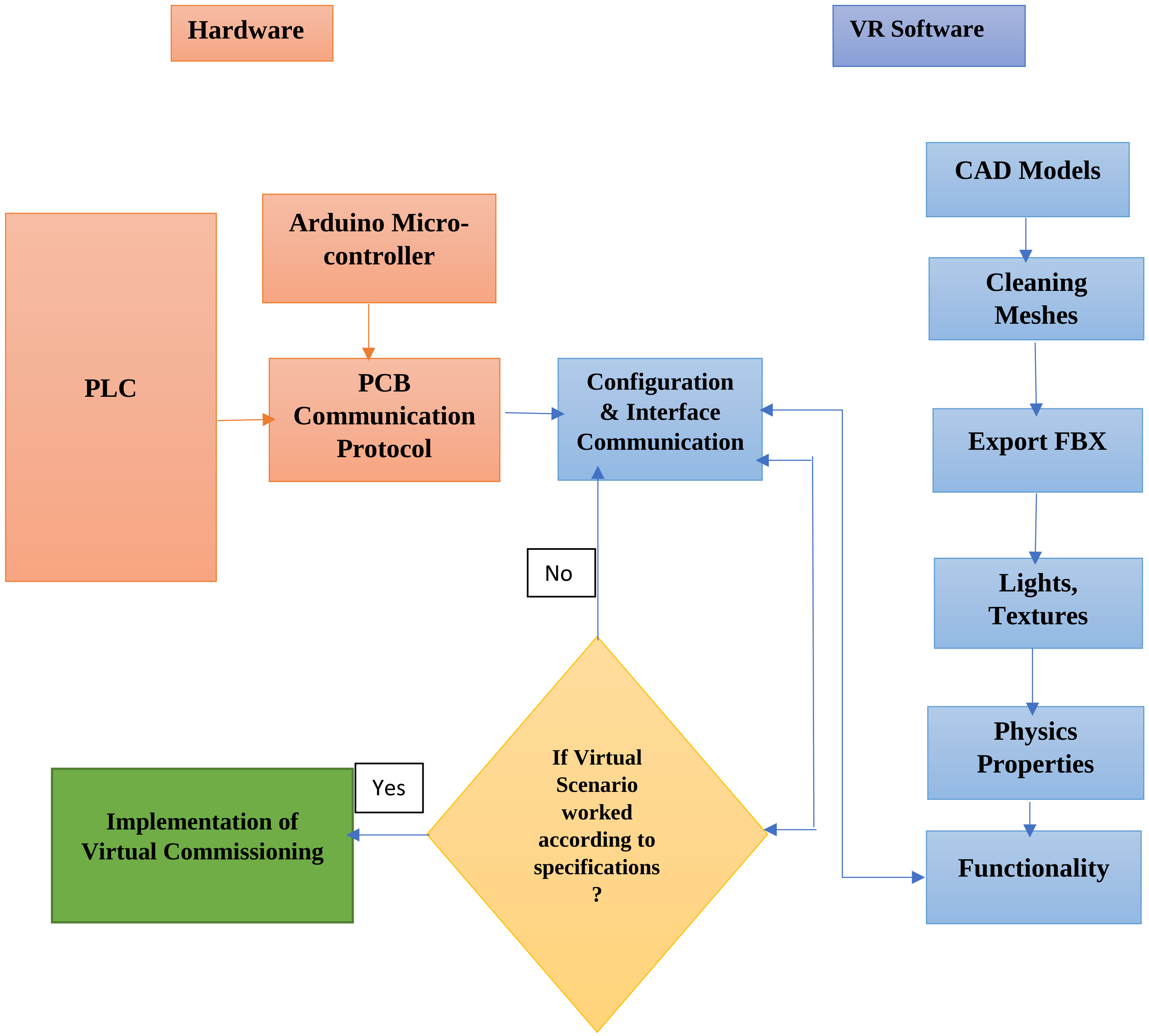


Figure 24. Proposed Methodology

3.1 To design a PCB and communication protocol to interface between PLC to the Virtual Reality System.

Eagle's UI is designed with what is called a modal interface. That is, you select one mode, perform it a bunch of times, as opposed to selecting an object and applying a single operation at a time. When used properly, this allows you to work very rapidly, but it can also be a major source of aggravation if you are used to the Windows-y way of doing things.

Eagle has four basic views: Library, Schematic, Board, and Control Panel.

Control Panel is the main window, it launches everything else and when you close it, all subordinate windows get closed.

Library - Allows you to manage and edit parts. Advanced usage of this will not be covered in this tutorial

Schematic - This is where you draw the schematic for your project. It defines the parts you have in your project, and which pins on the parts should be connected.

Board - This is where you lay out the pieces of your project and physically connect the correct pins as defined in the Schematic.

3.2 To design a virtual Reality Scenario of the industrial automation plant using Unity 3D 5.5

Unity allows you to interact with them via not only code, but also visual components, and export them to every major mobile platform and a whole lot more for free. (There’s also a pro version that’s very nice, but it isn’t free. You can do an impressive amount with the free version.) Unity supports all major 3D applications and many audio formats, and even understands the Photoshop .psd format so you can just drop a .psd file into a Unity project. Unity allows you to import FBX files and assemble assets, write C#, or JavaScript code to interact with your objects, create or import animations for use with an advanced animation system, and much more.

3.3 Configuration and Interfacing between real control system(PLCs) with the Virtual automated plant (Virtual scenario) using DAQ(PCB).

The Real control system is called PLC, a digital industrial computer control system that continuously monitors the state of input devices and makes decisions based upon a custom program to control the state of output devices.

In this stage, the real control system (PLC) is connected to the virtual plant model with a DAQ interface, internally converting the physical signals to virtual signals. Afterwards, the PLC and the simulation systems are interconnected via the communication system or direct wiring. The I/O lists of the simulation system and the I/O lists of the control program must be properly connected with each other. The information which is extracted from existing data sources by using the virtual plant model can represent the structure of the workstation (3D dynamic system and behavioral simulation, PLC program, link-up between PLC and virtual simulation model).

3.4 Programming in unity, Arduino, and RSLogix 500:

After configuration and interfacing, programming software should be done. Firstly, write programming driver for Digital I/O in Arduino IDE software using C++ language. Then check program on the serial monitor to verify the required results. Secondly, write programming in Unity Microsoft Visual Studio using C# for all the functionality in the Virtual Reality Scenario and write communication program to interface PCB with the Virtual Reality environment to communicate the functionality according to the scenario.

PLC Ladder Logic Program using RSLogix 500 Software

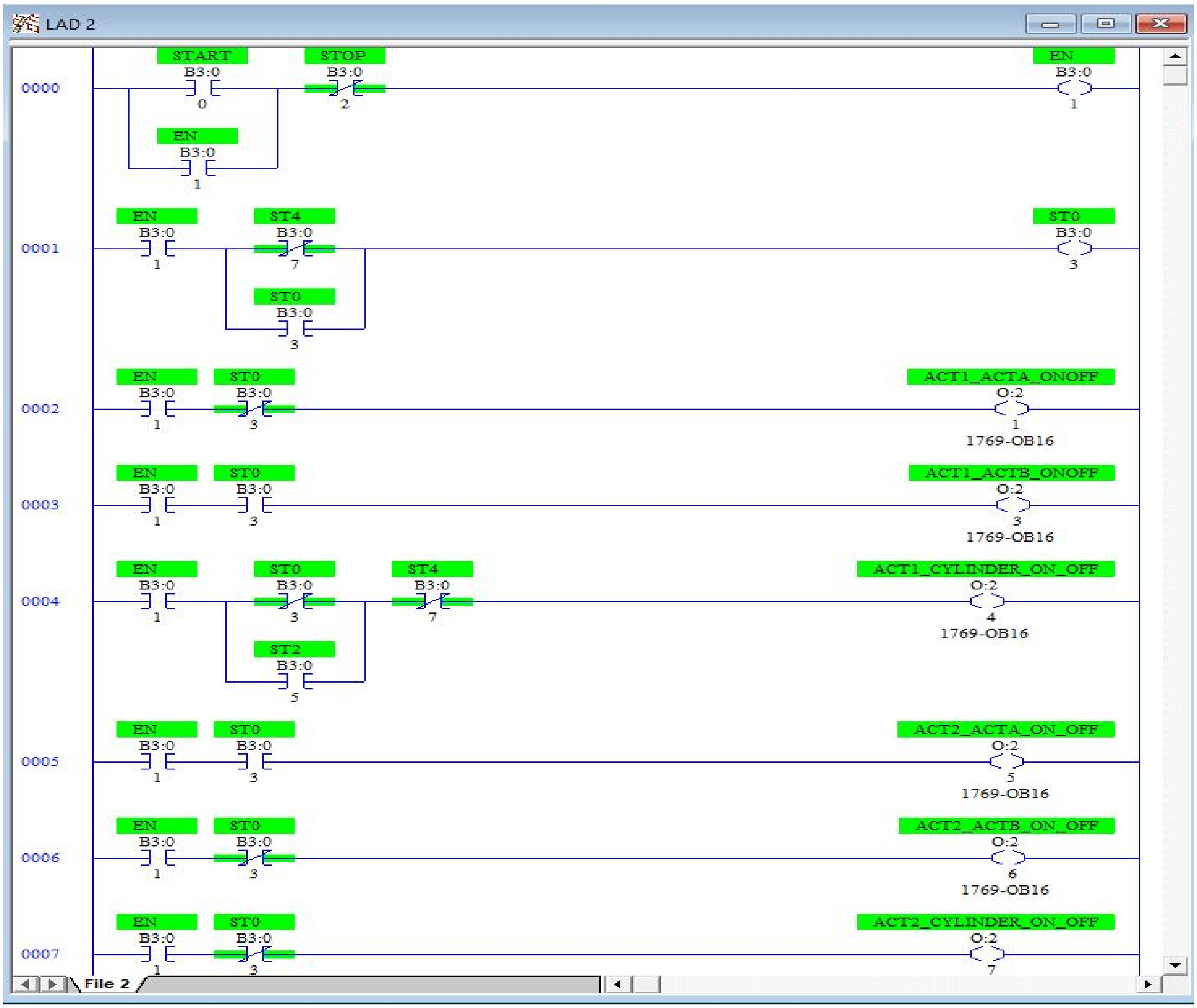
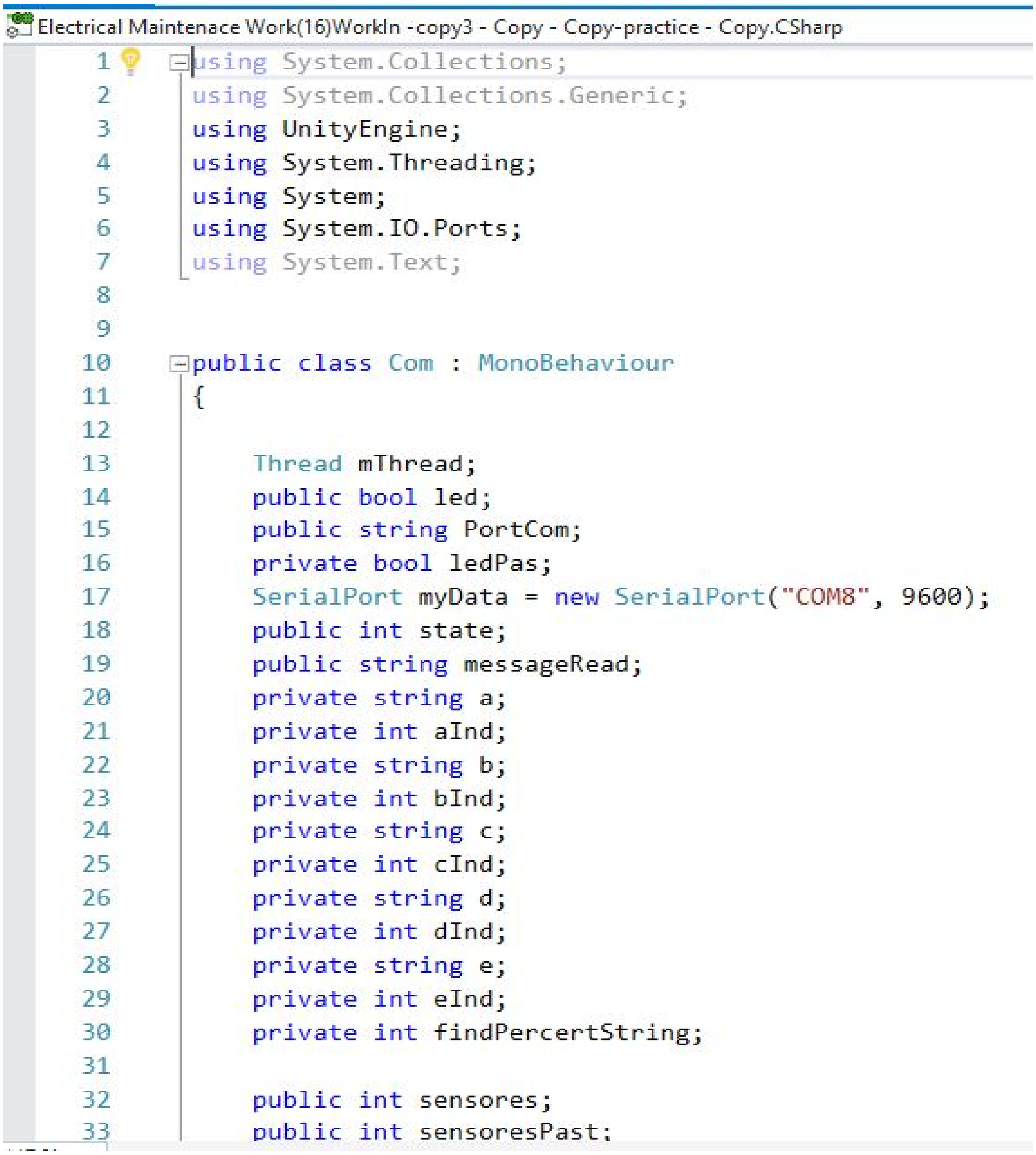


Figure 25. Ladder Logic Program (RSLogix 500)



Electrical Maintenace Work(16)WorkIn -copy3 - Copy - Copy-practice - Copy.CSharp

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System.Threading;
using System;
using System.IO.Ports;
using System.Text;


public class Com : MonoBehaviour
{

    Thread mThread;
    public bool led;
    public string PortCom;
    private bool ledPas;
    SerialPort myData = new SerialPort("COM8", 9600);
    public int state;
    public string messageRead;
    private string a;
    private int aInd;
    private string b;
    private int bInd;
    private string c;
    private int cInd;
    private string d;
    private int dInd;
    private string e;
    private int eInd;
    private int findPercertString;

    public int sensores;
    public int sensoresPast;
```

Figure 26. C# Programming (Microsoft Visual Studio)

```
int state = 1;

const char c1 = 'D';
 const char c11 = 'A';
  const char C2 = 'I';
  const char c22 = 'I';
   const char C3 = '=';
   const char c33 = '0';
   const char c4 = '0';
   const char c44 = '1';

void setup() {

      Serial.begin(9600);
      pinMode(2, INPUT);
      pinMode(3, INPUT);
      pinMode(4, INPUT);
      pinMode(5, OUTPUT);
      pinMode(6, OUTPUT);
      pinMode(7, INPUT);
      pinMode(8, INPUT);
      pinMode(9, INPUT);
      pinMode(10, OUTPUT);
      pinMode(11, OUTPUT);
      pinMode(12, OUTPUT);
      pinMode(13, OUTPUT);

      pinMode(A0, INPUT);
      pinMode(A1, INPUT);
      pinMode(A2, INPUT);
      pinMode(A3, INPUT);
      pinMode(A4, INPUT);
      pinMode(A5, INPUT);

      pinMode(0, INPUT);              // rx
      pinMode(1, OUTPUT);                 // tx
      digitalWrite(1, HIGH);
      delay(10);
      digitalWrite(10, HIGH);
      SWprint('d');                          // data
      SWprint('0');
      SWprint('=');
      SWprint(10);
      state = 1;
```

Figure 27. C++ Programming (Ardunio)

3.5 Testing and Performance Analysis:

After configuration and interfacing between real control system (PLCs) with the Virtual Automated plant (Virtual scenario) using DAQ, it should work according to the functionality. First starting with Arduino execute and compile after finishing open PLC software RSLinx to make communication with RS-232 and RSLogix to go online, download the program to the computer. Here in the RSLogix, programming will do with ladder logic by giving the address of the bits and create new addresses for input and outputs. You can give the same addresses for different output commands. Then, open in Unity 3D virtual scenario, open the Microsoft visual programming software and program for the functionality of the created objects in the unity. Save and build the solution and go to unity play the game, the whole line of automation and sequences will work according to the sensors, actuators according to functionality. If any error occurs in the play mode because of errors in the code or in the communication connections. Do the troubleshooting by each line where it shows error and modifies the code then compile again. At the last stage, the performance of the automation and control system will be verified.

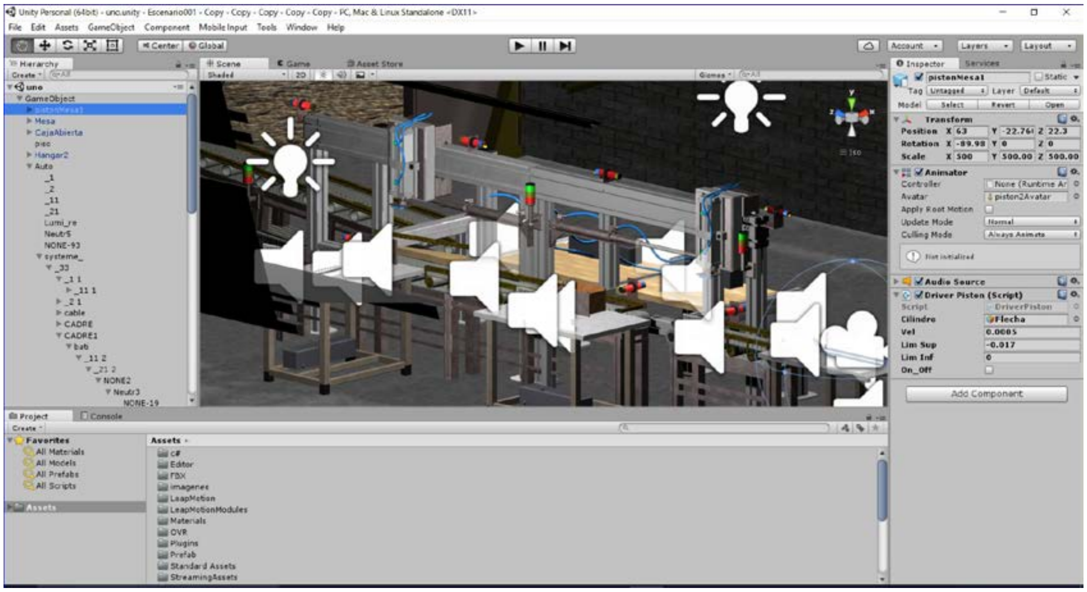
3.6 Implementation of the Virtual Commissioning.

After the interfacing of the real control system and virtual plant model, it worked according to the design specifications, virtual commissioning can be implemented.

Chapter 4 Design and Specifications

4.1 System requirements.

For this investigation, the proposed system consists of a box packaging and assembling automated plant, in which the main purpose is to have wooden boxes go through a process in which pistons will move the box when sensor is activated and detected; then, the boxes will go across a conveyor in which a pneumatic actuator will put and packing on top on each one. The system will not only contain the boxes packaging processes, as a dynamic system will also run alongside. Each one of the I/O that are part of the systems will interact all the time, to a communication protocol in which the virtual world interacts with the physical world. To make this a more practical application, and interface will be constructed so that a programmable logic controller (PLC) can interact with the virtual world.



pistonMesa1 ☐ Static
Tag Untagged Layer Default
Model Select Revert Open
Transform
Position X 63 Y -22.76 Z 22.3
Rotation X -89.98 Y 0 Z 0
Scale X 500 Y 500.00 Z 500.00
☑ Animator
Controller None (Runtime Ar
Avatar piston2Avatar
Apply Root Motion ☐
Update Mode Normal
Culling Mode Always Animate
Not initialized
☑ Audio Source
☑ Driver Piston (Script)
Script DriverPiston
Cilindro Flecha
Vel 0.0005
Lim Sup -0.017
Lim Inf 0
On_Off ☐

Figure 28. Automated Box Packaging process (Unity)

4.2 3D CAD Modeling.

The process to create any virtual object must begin by obtaining all the physical dimensions and properties that are present in its real-life counterpart. After acquiring the dimensions of the desired object, the next step translated them into a CAD software that would allow to creating a three-dimensional model, along with properties like local positions and rotation according to a frame in a virtual space, as well as a mesh.

One of the approaches in creating the 3D models for this scenario was to delete elements that could be easily interchangeable with much simpler geometrical forms inside the game engine, mainly because the lasts don't have a very detailed mesh. Once the 3D models were accordingly constructed like in they are exported into the game engine Unity 3D in an FBX (Film Box) format

4.3 Electronic and PCB Design

Recent research has been able to present practical cases in which the game engine is able to interact with different hardware and communication protocol that can be shared with other devices like PLCs. The work on this research further expands in this capability and established a way to not only interact between digital inputs and outputs but also analog ones as well. For this, an electronic interface was designed and implemented to fully take advantage of the PLC functionality.

The proposed interface schematic, represented in the figure. 32, is composed of an array of relays that are used to send digital signals coming from a microcontroller towards the PLC, as well as optocouplers that can provide electrical isolation for incoming digital signals of the PLC. It also contains many operational amplifiers that allow to send and receive analog signals into the microcontroller.

Figures 29, 30 and 31 are the PCB Standard, Top, and Bottom Layers. It is a 2-layer design. In figure 30 red color routing nets are the top layer and in figure 31 blue color routing nets are the bottom layer. Both layers are separated by Vias from top to bottom. It does everything in the eagle software with an option of Auto Routing with net different values of net size in this PCB design net class is 22 mil. It should do routing 100% and check vias do not overlap or nearness with the component pins.

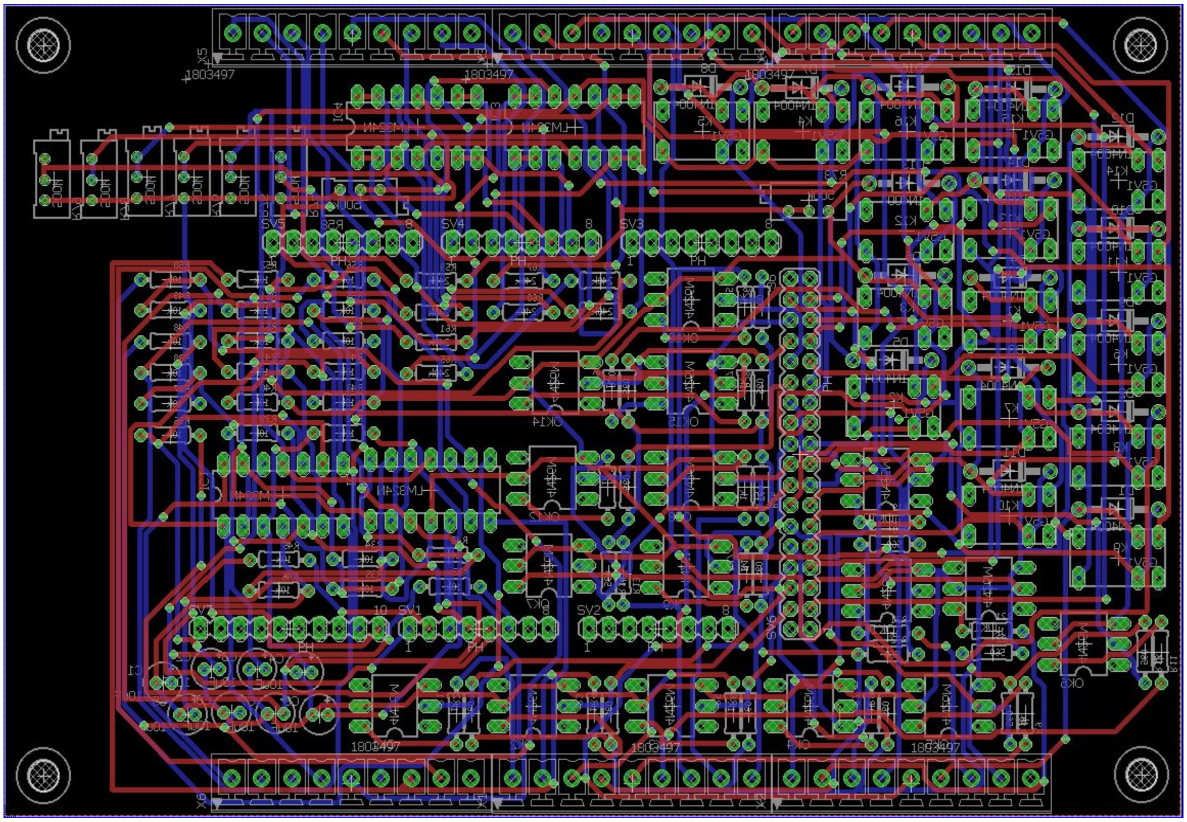


Figure 29. Electronic PCB design and tracing (Eagle Schematic)

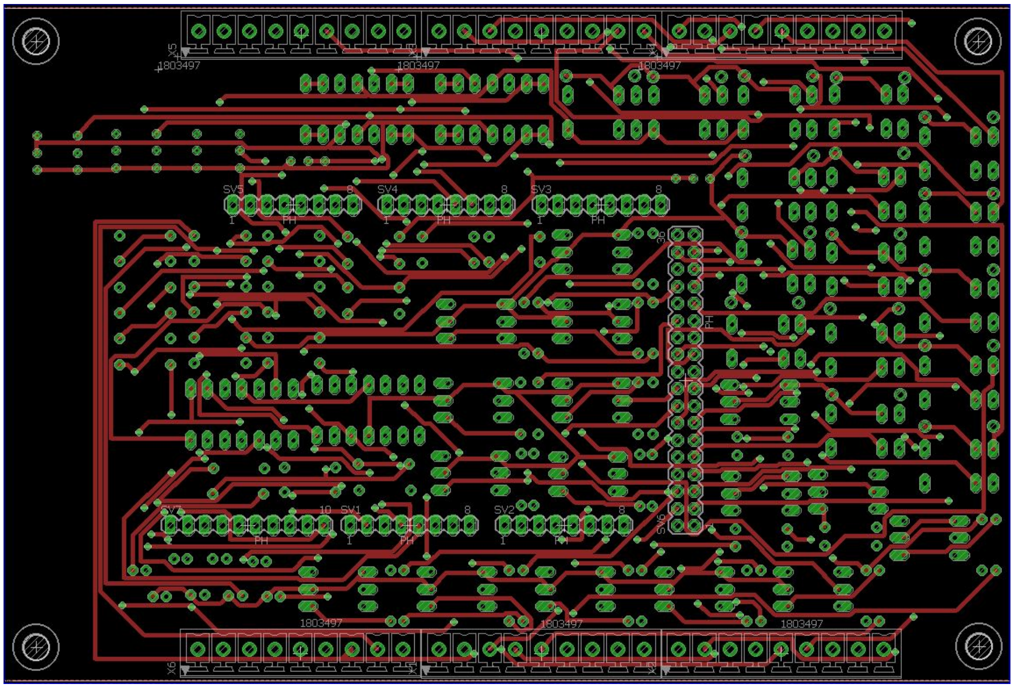


Figure 30. PCB Top Layer Tracing (Eagle Schematic)

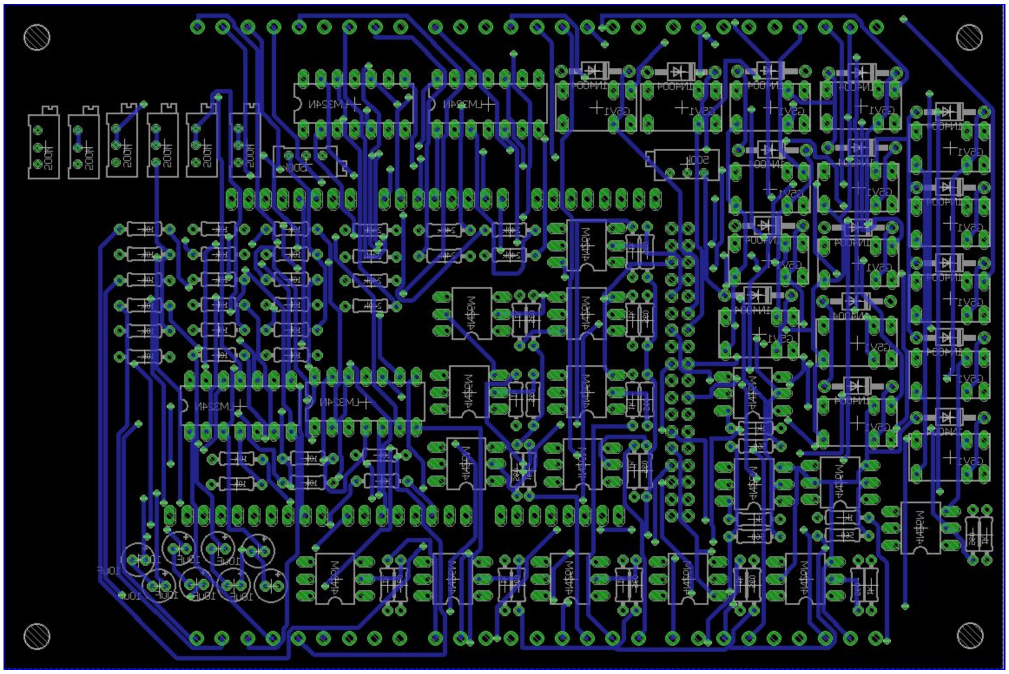


Figure 31. PCB Bottom Layer Tracing (Eagle Schematic)

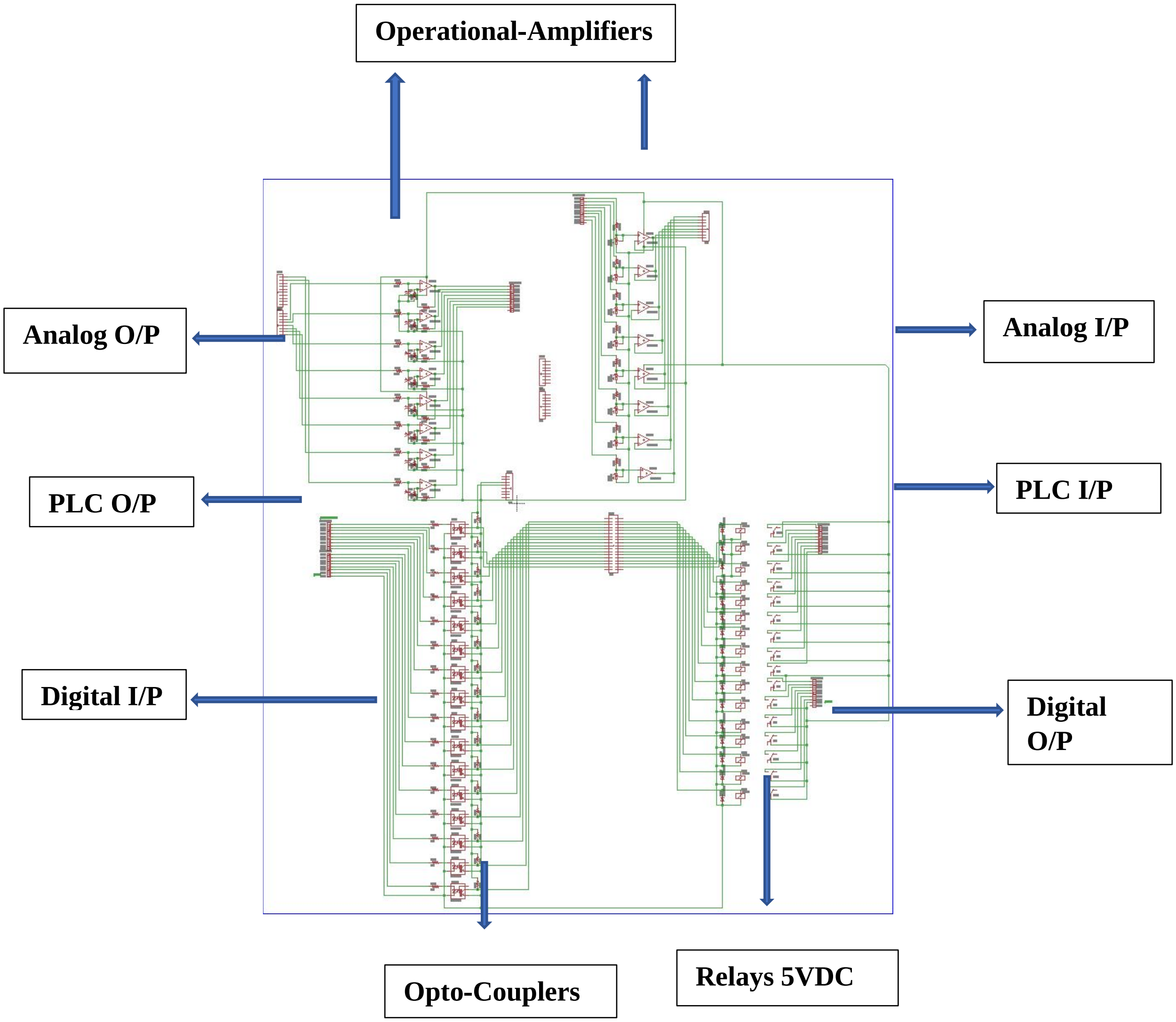


Figure 32. PCB-Schematic (Eagle Schematic)

4.4 Virtual Scenario setup.

Each object in the Unity 3D game engine is configured as “Game Objects”, which are the basic elements of the software. A “Game Object” has a “Transform” component, which determines its three-dimensional position, rotation and scaling. For objects to be animated, one can simply modify these values according to the desired movement. However, for objects that also respond to a specific dynamic interaction, there needs to be another component called “Rigid Body”. This will allow the object to be affected by other physical properties like gravity and joints. In the case of the Box Packaging plant, an example of this would be the pistons that control the movement of the box package with double effect actuator valves showed in fig. 33

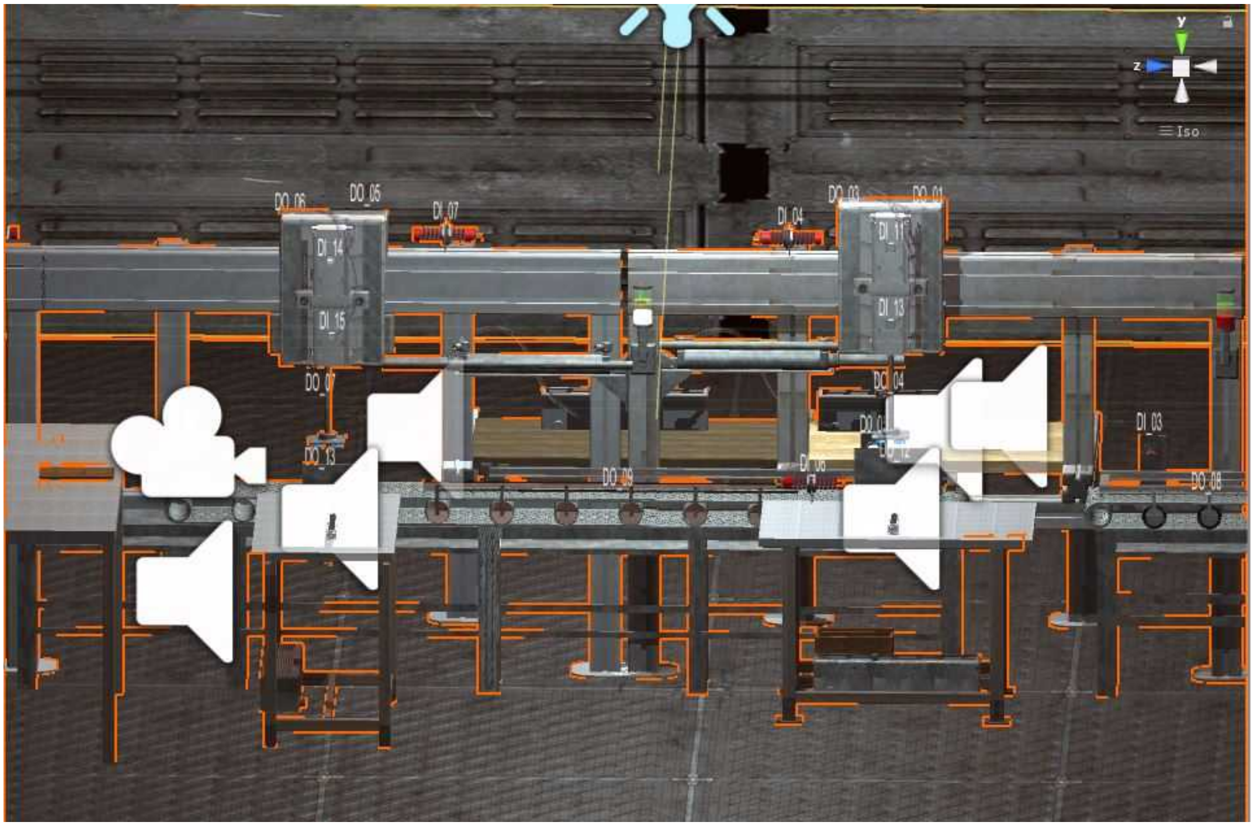


Figure 33. Double Effect Actuators

A collider allows the shape of the element to interact with all the elements of the scenario by limiting a covering surface around the desired object. This surface can take the shape of different geometrical figures like spheres and cubes, or even take the full mesh shape of an object. Another important process inside the virtual scenario step-up is the creation of scripts. These are programming instructions that can be assigned to Game Objects taking full advantage of object-oriented programming languages like C#. Multiple scripts can be assigned to the same object, and the fact that they use object-oriented programming makes the interaction between multiple elements at the same time relatively easy.

4.5 Oculus Rift Hardware.

The ever-increasing field of VR has demanded the development of specific hardware that allows the user to easily interact with these scenarios. One of the most potent and accessible devices is the Oculus Rift, which consists of headset and hand controllers that are linked to fixed-position sensors. Another advantage of the use of Oculus Rift is the compatibility with game engine software like Unity 3D thanks to open access drivers that can be easily installed. For the process described, the user can fully operate inside the scenario by inserting a first-person object that is linked with the headset and the hand controllers, making familiar hand movements easily executable.

Figure 34. Oculus Rift Hardware set(oculusrift.com)

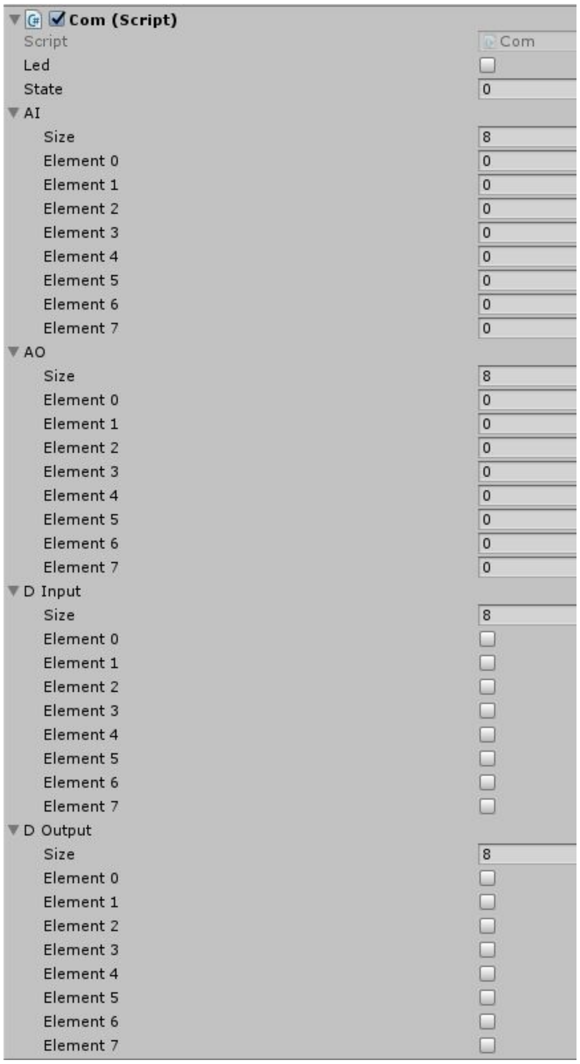
Chapter 5 Findings

5.1 Implementation Analysis

As it was mentioned earlier, one of the main purposes of this paper is to show the results of the interaction between a virtual reality scenario setup and how it can be linked with real-life electronic circuits that can handle the reception and emission of both digital and analog I/O. Digital input signals are sent by activating relays which in turn can let pass the required voltage to the PLC inputs. Digital output signals are received through an opto-isolated circuit, in which the output signal coming from the PLC is transmitted thanks to an optocoupler. As for the analog input and output signals that the microcontroller receives and send respectively, they go through an operational amplifier that can reduce or increase the voltage at its output.

On the other side, the VR scenario is configured so that it can access each single I/O that is going to interact with the system. By sending the data through the serial port, the type of signal is then deciphered by the script, which then will be assigned to a different element on the arrays. The number of digital I/O will depend on the number of inputs and outputs inside the microcontroller. This is seen in fig. 35.

This research project will be able to overcome the challenges facing the educational institutions and automation industries for training applications, saving more time, resources and maintenance.



☑ **Com (Script)**	
Script	Com
Led	☐
State	0
AI	
Size	8
Element 0	0
Element 1	0
Element 2	0
Element 3	0
Element 4	0
Element 5	0
Element 6	0
Element 7	0
AO	
Size	8
Element 0	0
Element 1	0
Element 2	0
Element 3	0
Element 4	0
Element 5	0
Element 6	0
Element 7	0
D Input	
Size	8
Element 0	☐
Element 1	☐
Element 2	☐
Element 3	☐
Element 4	☐
Element 5	☐
Element 6	☐
Element 7	☐
D Output	
Size	8
Element 0	☐
Element 1	☐
Element 2	☐
Element 3	☐
Element 4	☐
Element 5	☐
Element 6	☐
Element 7	☐

Figure 35. I/O table of the electronic-VR interface.

Chapter 6 Results and Conclusions

Results

In this research, the results found are the interaction between the real control system and virtual automation plant. The real control system inputs send physical signals to the Microcontroller (DAQ board) internally converts into the virtual signals and read by the Unity 3D interface. Likewise, Unity 3D writes a virtual signal to the Microcontroller (DAQ) converts virtual digital signals to the real physical signal (PLC). Design and Developed PCB consists of the microcontroller, relays, operational amplifiers for PWM to Analog outputs (4mA – 20mA) or (0-10V) is used to communicate the inputs, outputs from PLC to Unity and Vice-Versa.

Conclusions

A systematic methodology was presented for the implementation of virtual commissioning to reduce the commissioning time, improving the coding in PLCs without damaging the hardware, interactive education training to teach the students how the line of production works in the automation industry, and the most important is safety maintenance. The methodology is based on the characterization of the physical properties of all the elements embedded in the scenario from the real control system to the dynamic model of the sensors, actuators used for the control and operation of the whole process. Visualizing the important system variables, test the behavior of different actuators when introducing their actual technical specifications, performing advanced control tests such as improving the high-level programming languages. So, after successfully interfacing the real control system to the virtual automation plant, it worked according to the dynamic system design specifications.

Augmented Reality can also be implemented by using same virtual scenario by implementing the change of view it appears to be real-world environment whose elements are augmented by sensory input such as audio, graphics video data. Virtual reality is better than augmented reality since the field of view is better in VR. For educational purposes, VR is better used to see the whole line of production and interaction. AR is better for seeing the scene in the real environment but cannot see the full automation process since it has a low processor.

References

Agarwal, T. (2015, January). *Overview of Electronics Communication Protocol*. Retrieved from elprocus.com: https://www.elprocus.com/communication-protocols/

Allen Bradley. (2015, June). *Industrial Automation Wiring and Grounding Guidelines publication.* Retrieved from Rockwell automation: http://literature.rockwellautomation.com/idc/groups/literature/documents/um/1764-um001_-en-p.pdf

Allen Bradley. (2015, June). *Micrologix 1500 Programmable Controllers.* Retrieved from Rockwell Automation: http://literature.rockwellautomation.com/idc/groups/literature/documents/um/1764-um001_-en-p.pdf

Aufderheide, D. C. (2015). METHOD AND SYSTEM FOR TESTING SAFETY AUTOMATION LOGIC OF A MANUFACTURING CELL.

Chryssolouris, G. (2013). *Manufacturing Systems: Theory and Practice* (Segunda ed.). New York, United States of America: Springer-Verlag New York.

COMMISSIONING SOLUTIONS. (n.d.). *Commissioning Solutions*. Retrieved from CxS: http://www.commissioningsolutions.com/retro-commissioning/

Gonzalez, C. (2015, June 1). *Engineering Essentials: What Is a Programmable Logic Controller?* (Machine Design) Retrieved from Machine Design: http://www.machinedesign.com/engineering-essentials/engineering-essentials-what-programmable-logic-controller

Hloska, k. 2. (2014). VC of Mechatronic Systems with the use of Simulation. *Virtual Commissioning*.

Lee, C. G., & Park, S. C. (2014, June 2). Survey on the virtual commissioning of manufacturing systems. *Journal of Computational Design and Engineering, I*(3), 213-222.

lesd, y. e. (2017). asdfgfh. *bgdhhjeg*, 12.

Liu, Z., Suchold, N., & Diedrich, C. (2012). Virtual commissioning of automated systems. In InTech, & F. Kongoli (Ed.), *Automation* (pp. 131-148). Magdeburg, Germany: Intech.

Margaret Rouse. (2013, September). *WhatIs.com*. Retrieved from whatis.techtarget.com: http://whatis.techtarget.com/definition/printed-circuit-board-PCB

Markis, S., Michalos, G., & Chryssolouris, G. (2012, August 25). Virtual Commissioning of an Assembly Cell with Cooperating Robots. *Advances in Decision Sciences, 2012*, 11.

National Instruments. (n.d.). *NI*. Retrieved from ni.com.

Ortega-Moody, J. A., Sánchez-Alonso, R. E., González-Barbosa, J. J., & Reyes-Morales, G. (2016, February). Virtual Laboratories for Training in Industrial Robotics. *IEEE Latin America Transactions, XIV*(2), 665-672.

Rouse, M. (2015). Virtual Reality. *A look at the cutting-edge tech and the enterprise*.

Rouse, M. (2016, April). Augmented Reality. *A look at the cutting-edge tech and the enterprise*.

Schriber, T. J., Brunner, D. T., & Smith, J. S. (2012). How discrete-event simulation software works and why it matters. *Proceedings of the Winter Simulation Conference* (p. 3). Winter Simulation Conference.

Shahim, N., & Møller, C. (2016). Economic justification of virtual commissioning in automation industry. *Proceedings of the 2016 Winter Simulation Conference* (pp. 2430-2441). Arlington: IEEE.

Simon S young. (2014). Computerized Data Acquisition and Analysis for the Life Sciences. Retrieved from Byte Paradigm.

Solutions, C. (n.d.). *Commissioning Solutions*. Retrieved from CxS: http://www.commissioningsolutions.com/retro-commissioning/

Strahilov, A., & Damrath, F. (2015, December 1). Simulation of the behavior of pneumatic drives for virtual commissioning of automated assembly systems. *Robotics and Computer-Integrated Manufacturing, XXXVI*(C), 101-108.

Süß, S., Hauf, D., Strahilov, A., & Diedrich, C. (2016, September 1). Standardized Classification and Interfaces of complex Behaviour Models in Virtual Commissioning. (A. Nassehi, & S. Newman, Eds.) *Procedia CIRP, 52*, 24-29.

Süß, S., Strahilov, A., & Diedrich, C. (2015). Behaviour simulation for virtual commissioning using co-simulation. *2015 IEEE 20th Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 1-8). IEEE.

VISUAL COMPONENTS. (2015, July 13). Retrieved from visualcomponents.com: http://www.visualcomponents.com/insights/articles/increasing-control-software-quality-with-virtual-commissioning/

wiseGEEK. (2014, July). Retrieved from wisegeek.com: http://www.wisegeek.com/what-is-a-digital-signal.htm *wiseGEEK*. (2014, July). Retrieved from wisegeek.com: http://www.wisegeek.org/what-is-an-analog-signal.htm