Senior Project Report:
Plant Growing Control with Modicon M580

June 10, 2020

Authors
Eileen Tran
Brittany Won

Advisor
Professor Taufik
Table of Contents

List of Figures ................................................................................................................................. 2
List of Tables ......................................................................................................................................... 2
Acknowledgment ................................................................................................................................. 3
Abstract .................................................................................................................................................. 4
I. Introduction ........................................................................................................................................ 5
II. Background ......................................................................................................................................... 9
III. Design Requirements ....................................................................................................................... 13
IV. Design ............................................................................................................................................... 18
V. Simulation Results and Analysis ....................................................................................................... 24
VI. Conclusion ......................................................................................................................................... 35
Appendix A. How to Use Unity Operator Screens ................................................................................ 37
Appendix B. Scheduling ......................................................................................................................... 41
Appendix C. Bill of Materials ................................................................................................................. 42
Appendix D. ABET Senior Project Analysis .......................................................................................... 43
References ............................................................................................................................................... 48
List of Figures

Figure 1-1. Basics of a PLC [3] ............................................................................................................. 5
Figure 1-2. HMI Depicting Start/Stope Button and Water Level Controls for the Tank [8] ........... 7
Figure 3-1. Level 0 Block Diagram .................................................................................................... 13
Figure 4-1. Components Interacting with Input Module .................................................................. 18
Figure 4-2. Components Interacting with PLC ................................................................................ 21
Figure 4-3. Components Interacting with Output Module ............................................................... 22
Figure 5-1. Calculating Temperature (K) From Resistance (Ω) ...................................................... 25
Figure 5-2. FBD of Sprinkler Conditions for Plant 1 ...................................................................... 25
Figure 5-3. FBD of Sprinkler Conditions for Plant 2 ...................................................................... 26
Figure 5-4. Soil Moisture as a Percentage ......................................................................................... 27
Figure 5-5. Temperature Conversion to Degrees Fahrenheit ......................................................... 27
Figure 5-6. Extended Daylight Hours Toggling for Plant 1 ............................................................ 28
Figure 5-7. HMI Display [19] [20] .................................................................................................. 29
Figure 5-8. Testing Sprinklers (Part 1) ............................................................................................ 31
Figure 5-9. Testing Sprinklers (Part 2) ............................................................................................ 32
Figure 5-10. Testing Extended Daylight Hours ................................................................................ 34
Figure A-1. Operator Screen Object Toolbox ................................................................................ 37
Figure A-2. Operator Screen Library Examples .............................................................................. 38
Figure A-3. Text Display Object Properties Example .................................................................. 39
Figure A-4. Spin Box Control Object Properties Example ............................................................ 40
Figure B-1. Fall Quarter 2019 Gantt Chart .................................................................................... 41
Figure B-2. Winter Quarter 2020 Gantt Chart ................................................................................ 41
Figure B-3. Spring Quarter 2020 Gantt Chart ................................................................................ 41

List of Tables

Table 3-1: Summary of Design Requirements ................................................................................. 16
Table 3-2: Summary of Previous Design Requirements ................................................................. 17
Table 4-1: Bill of Materials ................................................................................................................ 23
Table C-1: BOM for Current Project ................................................................................................. 42
Table C-2: BOM for Previous Project [2] ....................................................................................... 42
Acknowledgment

Although many people are deserving of our thanks in completing this project and cannot all be listed, a few people stand out the most to us. Thank you to Professor Taufik, for guiding us in our studies and giving us the freedom to design our own experiment for future generations of Cal Poly electrical engineering students. Thank you to Schneider Electric for donating the equipment that will become invaluable tools of learning, and to Darrick Baker for making the trips to Cal Poly to host the PLC workshops for us. Thank you to Kevin Shipp and Anthony Tyler, our brother team in this PLC lab project, for undertaking this adventure with us. And finally, we want to thank our families for all the love and support throughout our college journey.
Abstract

Currently, the Industrial Power and Controls lab is being developed by the Cal Poly Electrical Engineering department. The lab is equipped with recently donated Modicon M580 Programmable Logic Controllers (PLCs) from Schneider Electric. In addition to PLCs, Schneider Electric also donated Human Machine Interfaces (HMI) and input/output modules. This project focuses on the creation of a lab experiment that will give future Cal Poly electrical engineering students hands-on experience with this equipment. To show students the broad range of PLCs and the role that electrical engineering plays in various industries, this experiment focuses on the agriculture industry, but more specifically, an automated indoor gardening system. Originally, the main goals of this project were the complete simulation and hardware test of a possible lab system, along with a corresponding instructional manual for the usage of the HMI in conjunction with the PLCs and input/output modules. Due to COVID-19, the hardware portion of this project was put on hold, along with the full instructional manual. However, a fully functional simulation using one of the PLC programming languages, function block diagram, allows for straightforward hardware testing in the future, when this project is turned into a full lab experiment. Although the hardware components have already been chosen and will appear in the bill of materials, they must still be tested before being put into use for the experiment. In addition, instructions on the HMI setup were also detailed, and will be important for the official lab manual.
I. Introduction

Living in the information age means that technology is evolving and moving us closer towards fully robotic automation. A key component in automation processes is the use of programmable logic controllers. In 1968, Richard E. Morley founded Bedford Associates and invented the first programmable logic controller (PLC). He established the Modicon company, with its name being a portmanteau of “Modular Digital Control” [1].

A PLC is essentially a computer that monitors inputs and outputs, and makes logic-based decisions for automated processes or machines. The CPU within a PLC system is a microprocessor which directs the PLC to execute instructions, communicate with other devices, carry out logic and arithmetic operations, and perform internal diagnostics. PLCs require a programming device, such as a computer, to upload data onto the CPU [2].

![PLC System Diagram](image)

Figure 1-1. Basics of a PLC [3]

In 1975, the first PLC with a microprocessor and distributed control was introduced, along with the first PLC with digitized process algorithms for continuous control. The year of
1979 saw the first industrial communications network, called Modbus, which allowed users to interface computers to controllers. By 1990, industrial control platforms were implemented on the factory floors themselves. They monitored, data logged, reported and triggered alarms when necessary. Within the next two years, Modbus was the industry standard. The ModConnect Partners Program was implemented to formally cooperate on hardware, software and integration services that would interface with all Modicon products.

In 1996, Schneider Groupe, soon to be Schneider Electric, became the sole owner of Modicon and its technology [4]. That very year, they introduced the Modicon Premium—the predecessor of the Programmable Automation Controller (PAC). It had a much higher capacity and higher performance, allowing for bigger and better applications. By 2007, Schneider released the first all-in-one PAC—the Modicon M340. It had its own USB port, SD card, Ethernet port and did not require batteries. Schneider continued improving on its products and in 2015, came out with the Modicon M580. The M580 is the first PAC to have built in ethernet capabilities and is used globally in today’s automation processes [5].

Another key development in the automation process is the integration of Human Machine Interfaces (HMI). HMI products originated from the need to make machinery easier to operate. Predecessors of HMI include the Batch Interface (1945-1968), the Command-Line User Interface (1969-Present), and the Graphical User Interface (1981-Present). HMI stems directly from the Graphical User Interface, and is now the most commonly used in the control of manufacturing processes due to its user-friendly interface [6].

HMIs work together with PLCs to monitor and control machines, and are often in the form of a touch screen. The screen may include information like temperature, pressure, and exact positioning of machines, among others. What the user can see on the screen, what they can
monitor, the “buttons” that can be pushed, and how the operator can manipulate the machine are all determined by the software written. Engineers can program an HMI to perform almost any function that can be controlled or monitored using a PLC. Each indicator and button must be programmed to a specific input or output address of a PLC. HMIs are often used as a centralized control unit for automation lines, since multiple machines can be monitored and controlled on a single HMI display.

A common HMI that most people encounter on a regular basis would be an ATM machine: the screen and buttons allow the user to operate the machine to dispense or deposit a certain amount of money. In a more industrial example, an HMI may have a large tank displayed on the screen with the level of a liquid displayed. Next to the tank is a pump to lower the liquid level. An HMI can also have a start and stop button displayed, which would be able to actually turn the pump on and off [7].

![Figure 1-2. HMI Depicting Start/Stop Button and Water Level Controls for the Tank [8]](image)

Now, PLC and HMI integrated systems are widely used in automation processes due to their efficiency and ease of use. In general, PLCs are typically used to drive output devices based on their inputs, which are usually analog or digital signals from sensors. HMIs can be programmed to allow the user to easily monitor and control the output devices as well. Since
their development in the late 90s, this system has been increasing in popularity. Automation processes now are easily monitored and controlled with the use of programmable logic controllers.
II. Background

Programmable logic controllers are used in many industries to monitor and control production processes. They are also present in our everyday lives, such as in road traffic signals, automatic car washes, elevators, automatic doors, and more [7]. Despite their usage in so many different applications that span across a wide range of fields, there is no course on this topic in the electrical engineering department at California Polytechnic State University, San Luis Obispo. There is only one course offered at Cal Poly that focuses on PLC and HMI use in automation, but it is within the IME department and is a more general overview of the topic. It is also not geared toward the electrical engineering majors, which means the course content is minimally relevant to electrical engineering related applications. This is a hole in the electrical engineering curriculum that needs to be addressed, especially considering that Cal Poly is known for preparing its graduates with hands-on skills in many industrial applications. Offering such a course would give the students a broader hands-on experience in the field of electrical engineering; and thus, would strengthen our “Learn By Doing” motto.

This project is a continuation of a previous senior project [8], with the same intention to introduce students to the importance and functionality of programmable logic controllers using equipment donated by Schneider Electric (PLCs, input/output modules, and HMIs). The previous project mainly consisted of setting up the PLCs, HMI, and input/output modules and establishing communication between them. This project, along with another concurrent one, furthers the progress by developing laboratory experiments designed to be completed within a typical three-hour lab period. Each of the labs to be created will be focused on various PLC applications, with this specific project focusing on the agriculture sector. A possible issue that is prevalent in any engineering/technological lab is that the experiments can easily become obsolete and no longer
relevant. To combat this, the experiments for this lab can be periodically updated to reflect new technical concepts that arise with the continuous development of PLC technology and current issues. One of the goals of this project is to introduce students to the use and applications of PLCs in industry, and the only way to do that is if the coursework is kept up to date with industry standards.

For a more in-depth experience, there are also numerous introductory PLC courses offered at colleges and universities in California using different equipment models and software, as well as certification courses, offered both online and in person. Schneider Electric even offers online, local, and in-house workshops [9]. The concept of a PLC course is not new and there are many other ways to gain a more thorough knowledge. But, having a course specifically for electrical engineers at Cal Poly will provide them with enough introductory knowledge to prepare them for working in automation and manufacturing industry.

Outside of California, schools such as Tennessee Tech University, Miami University (in Ohio), and University of Wyoming all have their own PLC labs [10, 11, 12]. Tennessee Tech’s PLC lab was established to support their industrial electronics courses, and uses Allen-Bradley SLC 503 and ControlLogix simulation software as well as some automatic control applications developed by their students [10]. Miami University focuses more on aspects of control systems by integrating concepts of transfer functions, system response, offset error, and stability into common practice, which provides an opportunity for students to learn about methods to measure and analyze control variables in manufacturing and mechanical processes [11]. Naturally, each university has their own take on the PLC labs, but the University of Wyoming, which has documented their process of creating a PLC lab, most closely resembles our project and its objectives. In creating the lab, a graduate student worked closely with an industrial control
company to develop a series of practical, hands on laboratory exercises. Their course was officially launched in Spring 2010, and was evenly split between the instruction of controlling industrial processes using microcontrollers and programmable logic controllers [12]. Similar to Tennessee Tech’s course, the University of Wyoming uses Allen-Bradley PLCs [10, 12]. This project mimics ours in that senior-level electrical engineering students at Cal Poly will be working with Schneider Electric to develop the lab and educate students on the fundamentals of PLC application design and implementation. Taking notes from the University of Wyoming, it would be beneficial to have students who took the course provide feedback so the lab can be adjusted accordingly, once the course becomes officially offered.

In summary, the goal of this project is to develop a laboratory course for electrical engineers at Cal Poly to give them experience working with programmable logic controllers, specifically the Modicon M580 PLCs from Schneider Electric. Each experiment will present the students with different challenges and will expose them to PLC applications in different fields, from heating, ventilation, and air conditioning (HVAC) to agriculture and to many other relevant future fields. With this experiment being a continuation of a previous project to introduce PLCs to students, it will not focus on setting up the various modules or establishing communication, but rather a specific application. This specific experiment relates to the agriculture sector and focuses on using a human machine interface (HMI) as a centralized device to monitor and control various aspects of the system. This experiment is based on the functionality of an AeroGarden, which tells the user when the plants need water and nutrients, and automatically turns grow lights on and off to simulate the sun [13]. Of course, in a laboratory setting, the parameters of the experiment are constrained, so not all of the aspects of an AeroGarden can be replicated. This experiment will have the students monitor the moisture level of the “soil” and
the temperature of the surrounding environment on the HMI and automatically turn on a “sprinkler” when the values cross a certain threshold. They will also be able to control the “sunlight” level (LEDs) via the HMI. In order to successfully complete this experiment, the students will need to prove their understanding of:

1. PLC specifications and architecture (input/output modules, CPU, power supply, and communication ports) [8].

2. How to properly wire various components to the correct input/output addresses.

3. How to program the PLC and HMI for the desired functions using Programming Standard IEC 61131-3 [14].

In addition to the hardware setup for the experiment, another outcome of this project is also the corresponding laboratory manual that details the procedure to conduct the laboratory experiment.
III. Design Requirements

Figure 3-1 shows the Level 0 block diagram of the project. The 120V AC input is provided by the wall outlets available in the lab, which supplies power to the various components in the PLC system. The student control input is how the students interact with the system, while the moisture level and temperature are measured inputs to the system and are monitored and controlled by the students. Based on the program written, the hardware driven at the output are the sprinkler, the light, and the HMI display showing the conditions for each plant.
Figure 3-2 shows the Level 1 block diagram. The student control is inputted into the HMI and PLC through a touchscreen and software programs, respectively. This allows the students to monitor, control, and modify the system as necessary throughout the experiment. The moisture levels and temperature are monitored by their respective sensors. This data is transmitted to the input module which is then fed into the PLC for processing by the students’ program. The program turns on the “sprinkler” (utilizing a small DC fan) at the output if the moisture level drops below a certain amount or the temperature increases above a certain level. The students alter the temperature by changing the voltage inputted into the nichrome wire. The HMI displays the moisture level and temperature, along with the light levels for each plant. The students control the amount of light (in the form of LEDs) each plant receives via the HMI. The light levels are controlled manually through the touch screen as opposed to automatically in the
program as in the case with the sprinkler. The 120V AC from the wall outlet provides power to the power supply, which in turn provides DC voltages to each of the subcomponents of the PLC system.

Table 3-1 details the new materials to be used in this experiment, while Table 3-2 lists the equipment and materials for the PLC system and setup, taken from the previous year’s report [2].
Table 3-1: Summary of Design Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Engineering Requirement</th>
</tr>
</thead>
</table>
| Cylewet Soil Moisture Sensor Soil Humidity Detection Module for Arduino [9] | 20 x 60 mm  
Voltage: 3.3V/ 5V  
Current: <20mA |
| Plastic Pot for Plants, Cuttings & Seedlings [10] | 100 x 70 mm  
Material: plastic |
| Fake Plant | 50 x 50 mm  
Material: plastic |
| LED [11] | 5mm RGB wavelengths:  
• R: 620nm-625nm  
• G: 515nm-520nm  
• B: 450nm-455nm  
4 pin  
Common anode  
Luminous intensity:  
• R: 2000-3000 mcd  
• G: 15,000-18,000 mcd  
• B: 7000-8000 mcd |
| Panaflo FBA08T12L Fan [12] | 80 x 80 x 15 mm  
Voltage: 7-13.8V  
Current: 79 mA  
Nominal speed: 2000 rpm |
| Temperature Sensor (NTC Thermistor) [13] | Resistance @ 25°C: 47kΩ  
Thermal dissipation constant: 1.5mW/°C  
Maximum power: 7.5mW |
| Nichrome Wire [14] | Diameter: 24 Gauge / 0.51mm  
Resistance: 1.6089 ohms/ ft  
Melting temperature 1400 °C |
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Engineering Requirement</th>
</tr>
</thead>
</table>
| Lab bench PLC mounting            | 35mm x 7.5mm x 2m symmetrical DIN rail for mounting components  
DIN rail length for lab bench - Machine cut to 6 feet  
16 AWG stranded wire                                                                                   |
| Modicon X80 I/O Platform          | IP degree of protection: IP20  
Product Compatibility:  
• BMXCPS power supply  
BMXP34 processor  
I/O Channels:  
• Discrete I/O 4096  
• Analog I/O 1024  
• Expert 144                                                                                           |
| 8-slot Ethernet + X-bus rack       | Power consumption: 2W  
Fixing Mode:  
• By 4 M6 screws plate  
• By clips 35mm symmetrical DIN rail  
• By 4 screws 4.32...6.35 mm panel                                                                       |
| Circuit Breaker (BMXCPS3500)      | Typical PSU input current:  
• 1.104Arms @115Vrms  
It (for rating external breaker)  
• 0.1As @115Vrms  
• 0.15As @230Vrms  
Breaker rating: 1P, 1.6A, ~277V, 50/60Hz                                                                |
| Circuit Breaker (ABL8REM24950)     | Mounting support: 35 x 7.5 mm symmetrical DIN rail  
Max input current: 2.8A  
Breaker: 1P, 3A, ~277V, 50/60Hz                                                                         |
| Touch Panel HMI (HMISTU855)        | Display resolution: 320 x 240 pixels QVGA  
Power consumption - 6.8 W  
Integrated connection USB 2.0 type A  
Supply voltage limits 20.4...28.8 V                                                                   |
| Unity Pro                          | Programming methods:  
• Structured Text (ST)  
• Function Block Diagram (FBD)  
• Ladder Logic (LL)                                                   |
IV. Design

Breaking the Level 1 Block Diagram in Figure 3-2 into three sections yields: the inputs, the PLC system, and the outputs. The input side (i.e. the components connected to the input module) consists of the moisture sensor and temperature sensor, with the temperature sensor also connected to a nichrome wire and power supply. A more detailed schematic of the input side is shown in Figure 4-1, along with justification for the chosen components.

![Diagram of components interacting with input module](image)

**Figure 4-1. Components Interacting with Input Module**

The Rigol DP832 DC Power Supply is a triple output power supply, with two channels supporting up to 30V/3A and the third channel supporting up to 5V/3A. This power supply is
commonly found in various Electrical Engineering labs (Building 20), and was chosen due to its multiple channels that will be used to power numerous components.

In order to simulate the temperature of different environments for each plant, nichrome wire was chosen due to its high resistivity, which makes it a decent heating element. The nichrome 80 wire is 80% nickel and 20% chromium, with a gauge of 24. It has $1.61\Omega/\text{ft}$ and can be used at temperatures up to $1180^\circ\text{C}$ (~2150°F), which is more than enough to simulate the plants’ changing environments.

To measure the temperature, a thermistor is used, which is essentially a resistor whose resistance is dependent on temperature [15]. Measuring the resistance and working backwards will produce the temperature of the nichrome wire (which is being used to represent the plants’ potential environments). This experiment uses an NTC 47kΩ 4050K bead thermistor. NTC stands for “Negative Temperature Coefficient,” which means that the resistance decreases as temperature increases. NTC thermistors are more common than its counterpart, PTC thermistors (“Positive Temperature Coefficient”) [15]. NTC thermistors also have a much higher temperature sensitivity coefficient than other sensors, such as silicon temperature sensors (silistors) and resistance temperature detectors (RTDs), at 5x and 10x greater, respectively [16]. The 47kΩ refers to its resistance at room temperature ($25^\circ\text{C}$), and the 4050K refers to the nominal B-constant ($25^\circ\text{C}$-$50^\circ\text{C}$) [17]. From this information, the resistance of the thermistor can be used to calculate the measured temperature using the following equation [17]:

$$R = R_o e^{B \left(\frac{1}{T} - \frac{1}{T_o}\right)}$$  \hspace{1cm} (4-1)

where  \( R = \text{resistance in ambient temperature } T \text{ (K)} \)

\( R_o = \text{resistance in ambient temperature } T_o \text{ (K)} \)

\( B = \text{B-constant of thermistor} \)
Rearranging the equation in (1) to solve for $T$ results in:

$$T = \left( \frac{1}{B} \ln \left( \frac{R}{R_o} \right) + \frac{1}{T_o} \right)^{-1}$$  \hspace{1cm} (4-2)

Plugging in the appropriate values for the NTC 47kΩ 4050K bead thermistor into (2) results in:

$$T = \left( \frac{1}{4050k} \ln \left( \frac{R}{47kΩ} \right) + \frac{1}{298.15k} \right)^{-1}$$  \hspace{1cm} (4-3)

This calculation will constantly be performed internally by the PLC to continuously monitor the temperature emitted by the nichrome wire.

The Cylewet soil moisture sensor has three pins: VCC, GND, and SIG. Power is connected to the VCC pin, and although this sensor can handle a range of 3.3V-5V, since the analog value returned depends on the provided voltage, 5V will be used to maximize the resolution. The ground of the power supply will be connected to GND on the sensor, and SIG on the moisture sensor will be connected to the input module in the PLC system. The SIG pin returns an analog value of ~0 when the sensor is dry and a value of ~880 when the sensor is completely saturated with water [18]. From this data, the analog values that correspond to the necessary soil moisture levels for each plant can be estimated.

The Modicon X80 Input Module was not a design factor, as it was donated by Schneider Electric. The input module will take in two values continuously: the resistance from the thermistor and the analog reading from the moisture sensor. These values are constantly being monitored by the PLC, which will perform certain actions at specific values according to the code written.

The next section is the PLC system, which interacts with the input and output modules, the HMI, the power supply, and the corresponding software. A more detailed schematic of the PLC section is shown in Figure 4-2, along with justification for the chosen components.
The new components to be discussed are the Modicon M580 ePAC, Harmony STU HMI, Unity Pro, and Modicon X80 Output Module. These components were provided by Schneider Electric, and therefore were not included in the design process. The code and calculations will be done in the Unity Pro software, which will provide the Modicon M580 ePAC with instructions to drive the outputs and map certain outputs to “buttons” on the Harmony STU HMI. The HMI will also display the soil moisture levels and temperature of each plant on the display for easy monitoring.

The last section covers the output side (i.e. the components connected to the output module). The output module interacts with the HMI, “sprinkler”, and LEDs. A more detailed
schematic of the input side is shown in Figure 4-3, along with justification for the chosen components.

Figure 4-3. Components Interacting with Output Module

The new components to be discussed on the output side are the RGB LED and the Panaflo FBA08T12L fan. The LED is generic and is used to simulate sunlight. Similar to typical LEDs, a voltage of ~3V is connected to the anode (longer leg), and the cathode (shorter leg) is connected to ground [11]. The Panaflo FBA08T12L fan is used to simulate a sprinkler, and will be turned on when the soil moisture level drops below a certain threshold or the temperature rises above a certain degree, as specified in the code. This fan has a rating of 12V/79mA, and was chosen because there were multiple spare fans readily available for use.
Figures 4-1 through 4-3 only show the connections for one plant, but since this lab uses two plants for a more diverse experiment, certain sections (i.e. nichrome wire, thermistor, soil moisture sensor, etc.) will need to be duplicated.

Table 4-1 details the bill of materials for the components used. Additional components that were not previously discussed are the fake succulents, plastic pots, and paper towels. Due to lab constraints and maintenance, fake succulents will be used in place of real plants. Similarly, paper towels will be used to simulate soil. The specific plants, pots, and paper towels were chosen for this lab due to their low price and ease of access.

Table 4-1: Bill of Materials

<table>
<thead>
<tr>
<th>Reference Name</th>
<th>Description</th>
<th>Count</th>
<th>Manufacturer</th>
<th>Part Number</th>
<th>Per Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moisture Sensor</td>
<td>Soil Humidity Detection Modules</td>
<td>10</td>
<td>Cylewet</td>
<td>CYT1033</td>
<td>$2.99</td>
</tr>
<tr>
<td>Fan</td>
<td>PC Fans</td>
<td>10</td>
<td>Panaflo</td>
<td>FBA08T12L</td>
<td>$0.00</td>
</tr>
<tr>
<td>Thermistor</td>
<td>NTC 47KOhm 4050K Bead Thermistors</td>
<td>10</td>
<td>Murata Electronics</td>
<td>NXRT15WB473FA3A016</td>
<td>$0.45</td>
</tr>
<tr>
<td>LED</td>
<td>Clear, 5mm, 4 pin, RGB Multicolor LEDs (100 pack)</td>
<td>1</td>
<td>Chanzon</td>
<td>100F5T-YT-RGB-CC</td>
<td>$8.96</td>
</tr>
<tr>
<td>Nichrome Wire</td>
<td>24 Gauge 106 Nichrome 80 Wire</td>
<td>1</td>
<td>Master Wire Supply</td>
<td>NiCr-24-0100</td>
<td>$8.49</td>
</tr>
<tr>
<td>Plant</td>
<td>Fake Succulents (14 pack)</td>
<td>2</td>
<td>Bioexcel</td>
<td>52140000</td>
<td>$19.39</td>
</tr>
<tr>
<td>Pot</td>
<td>4&quot; Plastic Pots (30 pack)</td>
<td>1</td>
<td>9GreenBox</td>
<td>NP-001</td>
<td>$11.99</td>
</tr>
<tr>
<td>Soil</td>
<td>Paper Towels (2 pack)</td>
<td>1</td>
<td>Bounty</td>
<td>N/A</td>
<td>$7.59</td>
</tr>
</tbody>
</table>

**Total Price:** $110.21
V. Simulation Results and Analysis

The software used for this project is Unity Pro XLS, provided by Schneider Electric to accompany the input/output modules, PLC, and HMI systems. Unity Pro XLS supports all of the PLC programming languages: structured text (ST), instruction list (IL), function block diagram (FBD), sequential function chart (SFC), and ladder diagram (LD). For this project, the simulation will be done using FBD. Although the software allows the user to connect to the PLC system and upload their code, due to connection issues, this project is done in “Simulation Mode” instead of the typical “Standard Mode.”

Since this project is based on an AeroGarden, which operates indoors, there is very little chance of the plants receiving too much water from external sources. To mirror this, the simulation is only set to consider low moisture levels and not levels above the needed amount. Similarly, the simulation assumes that the surrounding temperature of the system does not get low enough to be an issue and only considers high temperatures. In hardware, the temperature would be changed by varying the voltage to the nichrome wire, but since the relationship between voltage and temperature are unknown for this component, the simulation does not take voltage as an input. Instead, since the relationship between the resistance and temperature for the thermistor is known, one input to the simulation is resistance. The simulation calculates the resulting temperature from equation (4-3). This calculation in FBD form is shown in Figure 5-1.
Figure 5-1. Calculating Temperature (K) From Resistance (Ω)

The output of this section becomes one of the inputs to the sprinkler section, shown in Figure 5-2. The other input is “moisture_level1,” which comes from the analog values that are continuously read by the soil moisture level sensor. If the moisture level drops below a certain value or the temperature rises past a certain value, the sprinkler will turn on until the issue is resolved. These thresholds are determined arbitrarily based on different types of plants.

Figure 5-2. FBD of Sprinkler Conditions for Plant 1

These FBDs show variable names with a “1” at the end to denote that these are for the first plant. Since there are two plants, all the diagrams are duplicated, and the second set is
denoted with a “2.” The first plant is designed to simulate the optimal conditions for a desert plant: low moisture levels and high temperatures. Figure 5-3 depicts the conditions for the second plant, which simulates the optimal conditions for a woodland plant: higher moisture levels and lower temperatures, relative to the first plant.

![FBD of Sprinkler Conditions for Plant 2](image)

Figure 5-3. FBD of Sprinkler Conditions for Plant 2

To monitor the plants’ conditions, the HMI is set to display the soil moisture level and temperature for each plant, as well as the status of each sprinkler (on or off). Since the current FBDs have moisture level as an analog value from 0 to 880 and the temperature in Kelvin, conversions are done to make the values more intuitive to the user. The soil moisture level becomes a percentage, with 0% being completely dry (~0 in the analog value) and 100% being completely saturated with water (~880 in the analog value). The temperature is converted from Kelvin to degrees Fahrenheit. The calculations are shown in Figures 5-4 and 5-5, respectively.
In addition to displaying the plant and sprinkler conditions, the HMI will also have an option to turn on “Extended Daylight Hours.” This function is in the form of a checkbox that the user can manually turn on and off. Leaving the switch in the off-position defaults to leaving the LEDs (that simulate sunlight) on for 15 hours a day, while turning on the “Extended Daylight Hours” provides light for 18 hours a day. In simulation, these values will be reduced drastically for the sake of testing. In this specific example, the standard time period for the LEDs to be on is 10 seconds, and the extended time is 20 seconds. These times were chosen so that users would be able to visibly see the difference in the two light cycles without wasting too much time. The logic to achieve this is seen in Figure 5-6.
The checkbox on the HMI is a Boolean operator connected to the variable “extended_daylight1”, with an empty box appearing as a 0 and a checked box appearing as a 1. The first row in Figure 5-6 shows the logic for an unchecked box (standard time), which confirms that an input of 0 results in the pulse timer turning on the LED at the output for 10 seconds. Similarly, the second row shows the logic for a checked box (extended time), which verifies that an input of 1 results in the pulse timer turning on the LED at the output for 20 seconds. Although seemingly unused, the “ET” pins on the timers display the elapsed time during simulation and have no need to be connected. Another point of concern is that the outputs (“LED1a” and “LED1b”) are different in simulation but are technically the same LED. However, this software compromise to allow the simulation to run does not affect the hardware test. In hardware, extra wires can be soldered to the LED leg to allow it to be connected to two different outputs in the output module, which correspond to the two different outputs in simulation.

The HMI screen displaying soil moisture level as a percentage, temperature in degrees Fahrenheit, status of sprinklers, and the option for Extended Daylight Hours for each plant is shown in Figure 5-7. Inputs were chosen arbitrarily to showcase various conditions.
In Figure 5-7, the numbers and switches are called display objects. They display the value or condition of the variable that is connected to them. The checkbox is an example of a control object. These types of objects affect the value or condition of the variable associated with them. A more detailed explanation of how these objects are created and used can be found in Appendix A [21]. In this HMI display, the numbers next to “Moisture Level” are connected to the “percent_ML” variable respective to each plant. Similarly, the numbers next to “Temperature” are connected to their respective “temp_F” variable. The switches next to “Sprinkler” correspond to whether the “sprinkler” variable is true or false for each plant, with true being “On” and false being “Off.” Lastly, the checkboxes next to the “Extended Daylight Hours” toggle the respective “extended_daylight” variable.

To show all the results of simulation, multiple scenarios with varying conditions were run. Figures 5-8 and 5-9 consider scenarios with varying soil moisture levels and temperatures to
trigger the sprinklers, and Figure 5-10 shows the results of the Extended Daylight Hours. All of the figures follow the same format, with the top left section displaying all the variables and their corresponding values, the top right section displaying the FBDs of interest to show the effect of the inputs with the logic, and the HMI screen at the bottom to display the conditions. The table of variables and values is organized so that the variables that appear in the FBD are located at the top (temperature, moisture level, and sprinkler), while the other variables follow afterwards. The human inputs for simulation are the resistances (which are used to calculate temperature), the soil moisture levels, and the option for Extended Daylight Hours.

For Figure 5-8 Plant 1, the soil moisture level is higher than needed and the temperature has not crossed the threshold, so neither of the conditions are met (indicated by the red lines in the FBD), and the sprinkler does not turn on. Plant 2 is the opposite; both the low soil moisture level and high temperature (both crossing their respective thresholds) satisfy the conditions (indicated by the green lines in the FBD), which turn the sprinkler on.
In Figure 5-9, Plant 1 has a low soil moisture level, while the temperature is still within the acceptable range. One of the conditions is met (indicated by the green line for the soil moisture and red line for temperature in the FBD), and the sprinkler turns on. Plant 2 is the opposite; the soil moisture level is higher than needed, while the temperature has crossed the
threshold. Since one of the conditions is satisfied (indicated by the red line for the soil moisture and green line for temperature in the FBD), the sprinkler is turned on.

Figure 5-9. Testing Sprinklers (Part 2)
Figure 5-10 shows Plant 1 getting the extended time and Plant 2 getting the standard time. The time delay between changing the two “extended_daylight” inputs caused the timers to have different starting points, which is why the elapsed time shown for each plant is different. The FBD verifies that Plant 1 receives 20 seconds of light, while Plant 2 only receives 10 seconds. Although not shown, after the timers reach their preset limits, the LEDs turn off.
From the collected data shown in Figures 5-8 to 5-10, the simulation works as intended. Ideally, when the hardware aspect is added, the user will just have to map the inputs and outputs from simulation to ports in the input/output modules, and connect the remainder of the circuit.
VI. Conclusion

The goal of this project was to design an experiment that incorporated both the PLC and HMI for a future laboratory course at Cal Poly. The proposed Indoor Automated Gardening System is a relatively straightforward experiment that students should be able to complete in a typical lab period. Because this experiment is designed for later in the course, students should already have experienced programming using function block diagrams. So, rather than having this experiment be FBD-oriented, more emphasis is placed on introducing the HMI and its setup. Overall, this experiment covers all the intended objectives that students should learn by the end of the course. These objectives, also stated earlier in Chapter II are reproduced below:

1. PLC specifications and architecture (input/output modules, CPU, power supply, and communication ports) [2].
2. How to properly wire various components to the correct input/output addresses.
3. How to program the PLC and HMI for the desired functions using Programming Standard IEC 61131-3 [8].

Although this experiment covers the basics, there are a lot of ways to improve. Due to campus being closed as a cautionary measure to the COVID-19 pandemic, no hardware results were obtained. Therefore, it is possible that the original components chosen for this project do not work together as ideally as planned and would need to be replaced with different models. In terms of simulation, as with all programming languages, there are numerous solutions to achieve the same result. The FBD design in this report, although it gets the job done, could be improved upon, and made more efficient. It was also mentioned in Chapter V that the simulation does not consider cases where the soil moisture level is too high or the temperature is too low, due to the constrained indoor environment. However, to be a fully complete system, it must consider all the
possible cases. Further work could be done to add these conditions and their solutions (i.e. if the temperature is too low, a pop-up on the HMI could tell the user to increase the voltage to the nichrome wire).

Furthermore, this experiment can be easily modified to fit different situations, environments, teaching points, or material needs. The stimuli, responses, and modules are all subject to change for specific needs. For example, although this project used moisture levels and temperature as the stimuli, other reasonable stimuli are light levels, air humidity levels, pH levels, or nutrient levels. To match these, other modules such as light sensors, humidity sensors, misters, soil pH sensors, pH dosers, and nutrient dosers will have to be redesigned into the system.

A note to consider is that the suggestions presented above are not required to cover the objectives but serve to add more variety and possibility. Due to extenuating circumstances, a lab manual for this experiment was not created. Therefore, the next step would be to create an accompanying lab manual for the current experiment. Revisions to the manual could be made in the future if the experiment is changed. With all aspects considered, although this experiment meets the intended objectives, it still has a lot of potential for future growth 🌱.
Appendix A. How to Use Unity Operator Screens

1. To access the operator screens, first go to the Project Browser window, right-click the Operator Screen folder, and select the New Screen option. Give the screen an appropriate name and then press “OK.”

2. To turn the grid on, go to the Tools drop down menu and select Options. Under the subcategory of Operator Screens, click on the Grid option. Check the “Display grid” box. 
(Note: The “Alignment of objects on grid” box can be clicked to snap objects to the grid. This can be helpful in formatting)

3. There are many ways to design the operator screen. A toolbox at the top of the Unity window contains lines, squares, circles, arches, textboxes, images, etc. Figure A-1 shows the toolbox.

![Operator Screen Object Toolbox](image)

Figure A-1. Operator Screen Object Toolbox

Alternatively, right clicking the screen will produce a drop-down menu with a “New” option and produce the same results. For more specific types of objects or symbols, go to the Tools drop down menu and select the “Operator Screen Library.” Below is an example of some useful symbols.
4. An important consideration to take in when designing the screen is the variable’s purpose on the screen, whether it is displaying or altering. There are two types of objects: display and control.

   i. Display objects only present the value or condition of the variable connected with it. All objects can function as a display object.

   ii. Control objects can alter the value or condition of its variable. Examples of control objects are entry fields, push buttons, checkboxes, spin boxes, and scale indicators.

5. To link an object to a variable, double-click on the object. Switch to the Animation tab and check the “Animated Object:” box. Fill in the variable name by either typing it in or selecting it in the Instance Selection window, accessed by clicking the small box with three dots on it. Next, fill out the display conditions. These will be dependent on the variable types. For example, Boolean variables will only allow the first three options of conditions,
“Continuous display”, “Bit = 0”, and “Bit = 1.” Figure A-3 shows an example of a properly set up text display object.

Figure A-3. Text Display Object Properties Example

(Note: It is important to understand that when setting up a BOOL type display object, Continuous display will leave the object up no matter the condition of the variable. “Bit = 0” and “Bit = 1” will only display the object when the variable is set to zero or one, respectively)

6. Setting up a control object is very similar. Instead of switching to the Animation tab, use the Control tab. Input the variable name and then fill out any other necessary information. For BOOL types this is unnecessary but value types will prompt for a range. There are also some Control element style choices that are optional. Figure A-4 shows an example of a filled-out spin box control object screen.
Figure A-4. Spin Box Control Object Properties Example

7. Images can be imported as display objects as well. Unity’s access of these images depends on their file path in the computer. This means that if the images are deleted or moved on the computer, they will disappear from the operator screen as well.
Appendix B. Scheduling

Figure B-1. Fall Quarter 2019 Gantt Chart

Figure B-2. Winter Quarter 2020 Gantt Chart

Figure B-3. Spring Quarter 2020 Gantt Chart
Appendix C. Bill of Materials

Table C-1: BOM for Current Project

<table>
<thead>
<tr>
<th>Reference Name</th>
<th>Description</th>
<th>Count</th>
<th>Manufacturer</th>
<th>Part Number</th>
<th>Per Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moisture Sensor</td>
<td>Soil Humidity Detection Modules</td>
<td>10</td>
<td>Cylewet</td>
<td>CYT1033</td>
<td>$2.99</td>
</tr>
<tr>
<td>Fan</td>
<td>PC Fans</td>
<td>10</td>
<td>Panaflo</td>
<td>FBA08T12L</td>
<td>$0.00</td>
</tr>
<tr>
<td>Thermistor</td>
<td>NTC 47K Ohm 4050K Bead Thermistors</td>
<td>10</td>
<td>Murata Electronics</td>
<td>NXRT15WB473FA3A016</td>
<td>$0.45</td>
</tr>
<tr>
<td>LED</td>
<td>Clear, 5mm, 4 pin, RGB Multicolor LEDs (100 pack)</td>
<td>1</td>
<td>Chanzon</td>
<td>100F5T-YT-RGB-CC</td>
<td>$8.96</td>
</tr>
<tr>
<td>Nichrome Wire</td>
<td>24 Gauge 100' Nichrome 80 Wire</td>
<td>1</td>
<td>Master Wire Supply</td>
<td>NiCr-24-0100</td>
<td>$8.49</td>
</tr>
<tr>
<td>Plant</td>
<td>Fake Succulents (14 pack)</td>
<td>2</td>
<td>Bioexcel</td>
<td>52140000</td>
<td>$19.39</td>
</tr>
<tr>
<td>Pot</td>
<td>4&quot; Plastic Pots (30 pack)</td>
<td>1</td>
<td>9GreenBox</td>
<td>NP-001</td>
<td>$11.99</td>
</tr>
<tr>
<td>Soil</td>
<td>Paper Towels (2 pack)</td>
<td>1</td>
<td>Bounty</td>
<td>N/A</td>
<td>$7.59</td>
</tr>
<tr>
<td><strong>Total Price:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>$10.21</strong></td>
</tr>
</tbody>
</table>

Table C-2: BOM for Previous Project [2]

<table>
<thead>
<tr>
<th>Product Code</th>
<th>Description</th>
<th>Supplier</th>
<th>Cost Per Unit ($)</th>
<th>Quantity</th>
<th>Total ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS84040</td>
<td>Modicon MS80 PS84040</td>
<td>Schneider Electric</td>
<td>6</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>CPS3500</td>
<td>CPU PSU</td>
<td>Schneider Electric</td>
<td>6</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Phaseo ABL8REM24050</td>
<td>HMI PSU</td>
<td>Schneider Electric</td>
<td>6</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>DD1602</td>
<td>Dig 16 24 Vdc Sink</td>
<td>Schneider Electric</td>
<td>6</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>DRA805</td>
<td>Dig 8O Isolated Relays</td>
<td>Schneider Electric</td>
<td>6</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AMC0410</td>
<td>Ana 4 U1 Inputs Isol High Speed</td>
<td>Schneider Electric</td>
<td>6</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>AMIO410</td>
<td>Ana 4 U1 Out Isolated</td>
<td>Schneider Electric</td>
<td>6</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>HMI855/HM855</td>
<td>Touchscreen Monitor</td>
<td>Schneider Electric</td>
<td>6</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>TC8EU053FNI0</td>
<td>Ethernet Switch</td>
<td>Schneider Electric</td>
<td>6</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>B01AUG33LG</td>
<td>Tyreome Gooseneck Tablet Stand</td>
<td>Amazon</td>
<td>6</td>
<td>125.94</td>
<td></td>
</tr>
<tr>
<td>AB 1489-M1C030</td>
<td>1P 3A Circuit Breaker</td>
<td>Royal Industrial Solutions</td>
<td>55.84</td>
<td>6</td>
<td>335.04</td>
</tr>
<tr>
<td>AB 1489-M1C016</td>
<td>1P 1.6A Circuit Breaker</td>
<td>Royal Industrial Solutions</td>
<td>55.84</td>
<td>6</td>
<td>335.04</td>
</tr>
<tr>
<td>-008190046019</td>
<td>DIN rail Weidmüller 35X7.5 LL 2M</td>
<td>Automation 24</td>
<td>7</td>
<td>106.47</td>
<td></td>
</tr>
<tr>
<td>AB 1492-J3</td>
<td>Terminal Block</td>
<td>Royal Industrial Solutions</td>
<td>0.78</td>
<td>140</td>
<td>109.20</td>
</tr>
<tr>
<td>AB 1492-EBJ3</td>
<td>Terminal Block End Barrier</td>
<td>Royal Industrial Solutions</td>
<td>0.56</td>
<td>14</td>
<td>7.84</td>
</tr>
<tr>
<td>AB 1492-EAJ3</td>
<td>Terminal Block End Anchor</td>
<td>Royal Industrial Solutions</td>
<td>1.50</td>
<td>14</td>
<td>21.00</td>
</tr>
<tr>
<td>AB 1492-JG3</td>
<td>Grid Terminal Block</td>
<td>Royal Industrial Solutions</td>
<td>3.92</td>
<td>12</td>
<td>47.04</td>
</tr>
<tr>
<td>FRT FREIGHT IN AB</td>
<td>Shipping and Handling</td>
<td>Royal Industrial Solutions</td>
<td>8.95</td>
<td>1</td>
<td>8.95</td>
</tr>
</tbody>
</table>

**Note:** Donated parts not included in final cost

**Grand Total:** $1,086.52
Appendix D. ABET Senior Project Analysis

**Project Title:** Plant Growing Control Using Modicon M580

**Students:** Eileen Tran, Brittany Won

**Advisor:** Taufik

1. **Summary of Functional Requirements:**
   a. Modicon M580 Programmable Logic Controllers (PLCs), Human Machine Interfaces (HMI), and input/output modules, all donated by Schneider Electric, are used in an experiment to create an automated indoor gardening system. The HMI displays the soil moisture level for each plant, the surrounding temperatures, the status of the sprinklers (on or off), and a switch that toggles the amount of sunlight for each plant.

2. **Primary Constraints:**
   a. The biggest challenge of this project is that none of the members have prior experience working with PLCs or HMIs, either theoretically or physically. Not only that, but also there are minimal available online resources, presumably due to Schneider Electric keeping a good amount of their tutorials limited to employees only.
   b. This project is also constrained due to its nature of being an experiment for a standalone lab. Since there is no separate lecture, students will learn all the material at the beginning of the lab period, which reduces the time they have left to complete the experiment. To account for this, the experiment must be doable in two and a half hours max out of the total three hours, which means that the experiment itself cannot be too complicated.

3. **Economic:**
   a. Economic impacts as a result of this project:
i. Human Capital: People are needed to set up communications between all the modules, program the necessary commands, and test the system.

ii. Financial Capital: Between 2018 and 2023, the PLC market is expected to grow at a compound annual growth rate of 3.7 percent [22].

iii. Manufactured or Real Capital: All the equipment used for this experiment is manufactured.

iv. Natural Capital: PLCs typically contain a plastic casing, resin for the circuit boards, metal for various components, etc. These are limited to what can be produced using the planet’s resources. This experiment also relies on electricity to power all the equipment.

b. The upfront costs include all the new components for the experiment, while the cost of labor is spread out over the project’s life cycle. For the students involved in the project, the benefits come from gaining working knowledge of PLCs and HMIs and their roles in monitoring and controlling various machines. For students who take this lab in the future, the benefits come from having the option to learn about PLCs while still in school.

c. This project is not expected to create a monetary profit; the students involved profit in the form of knowledge, future students profit in the form of an additional class they can take, and Schneider Electric profits from students using their equipment and potentially working for them in the future.

d. This experiment will exist as long as the information is correct, and the equipment remains the same. Since Schneider Electric donated very recent models, it is expected that the experiment will not change significantly for years. In terms of maintenance, as with all lab equipment, the PLCs and HMI must be kept in working conditions. If issues
were to occur that could not be fixed easily, either new parts must be purchased or Schneider would need to be contacted to potentially have someone be sent over to do repairs.

4. Commercial Manufacturing:
   a. Since the project is to create a lab experiment specifically for the Industrial Power and Controls lab at Cal Poly, it will not be commercially manufactured.

5. Environmental:
   a. The main environmental impact from this project is the electricity needed to power all the equipment. This project does not significantly improve or harm the natural resources and ecosystem services, or impact other species in a direct way.

6. Manufacturability:
   a. This project will not be manufactured since the experiment is intended to be used for the lab at Cal Poly only.

7. Sustainability:
   a. There should not be any issues with maintaining the completed system, as it will be stored indoors away from outside weather conditions. Once connected, the individual modules (PLCs, HMIs, input/output modules) should not need to be disconnected, and can be left alone. The components for the experiment itself (sensors, fans, etc.) are easily disconnected and can be reused.
   b. The general goal of the project was to design an experiment that focused on the use of HMIs in PLC systems. Although an agriculture-themed approach was chosen, PLCs are used in so many different industries that this experiment could be changed completely in the future. Assuming this version of the experiment is kept, possible upgrades could
include different sensors to monitor different conditions or different user inputs for the HMI. A possible challenge with the upgrade is that new components must be tested and additional code will need to be written, although this is more of a time issue than a challenge.

8. Ethical:

a. The experiment design must be difficult enough to challenge the students and provide enough learning opportunities, while also being simple enough for an introductory lab.

b. This project is in accordance with points five and six of the IEEE Code of Ethics, and the points are listed below, respectively:

   i. to improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;

   ii. to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;

   c. With this project, future EE students will have the opportunity to be introduced to PLC and HMI systems (point 5), and this can only be achieved if the current students are proficient enough in understanding the equipment that the project can be completed successfully (point 6). This also follows the concept of utilitarianism: the greatest good for the greatest number. The current students benefit from learning about PLC and HMI systems and being able to be part of creating a lab, future students benefit from the opportunity to take the lab, and Schneider Electric benefits from having students know how to use their equipment.
9. Health and Safety:
   a. A potential danger in this project, as with all EE labs, is the possibility of electric shock from the wiring between components.

10. Social and Political:
   a. Students who take this laboratory course in the future will have an advantage over other students in the Industrial Controls and Automation industry, since they will have experience working with PLC systems. Schneider Electric will possibly have an advantage over their competitors in hiring employees, since there will be students who are familiar with their equipment because of this project.

11. Development:
   a. During this project, a representative from Schneider Electric, Darrick Baker, hosted workshops at Cal Poly for the senior project team to become familiar with the equipment, communications, and operating procedures of PLCs. In addition to that, the team also researched several articles and datasheets (listed under References) to get a better sense on the capabilities and limitations of the equipment, before designing the experiment.
   b. During the course of this project, the team became familiar with Unity Pro XLS, the accompanying software provided by Schneider Electric. Although the software supports all the PLC programming languages, the experiment focuses on function block diagrams specifically. The team also learned how to use the operator screens in Unity Pro XLS to program the HMI “buttons” to specific inputs or outputs in the FBD.
References


