5-2018

SCADA and PLC Systems Configuration for the NCREPT Test Facility

Arman Ahmed

University of Arkansas, Fayetteville

SCADA and PLC Systems Configuration for the NCREPT Test Facility

An undergraduate Honors thesis submitted in partial fulfilment
of the requirements for the degree of
Bachelor of Science in Electrical Engineering

by

Arman Ahmed

May 2018
University of Arkansas

Abstract

This thesis details the project to update the control and interface system of the National Center for Reliable Electric Power Transmission (NCREPT) testing facility. The need for this project arose from the 2017-2018 expansion of the facility, which included some modifications in the layout of electrical equipment used for testing purposes. These modifications necessitated the update of the control and interface system. Additionally, the old system was implemented a decade ago and is nearing obsolescence, so the facility's expansion served as an opportune time for an upgrade.

There were two main parts to the scope of this project, which were developed in parallel. The first part involved the programming and configuration of the new programmable logic controller (PLC) from Bedrock Automation. The Bedrock PLC is a more powerful device possessing advanced cybersecurity features, as well as support for multiple communication protocols which allows greater flexibility in implementation. The added features will prove to be advantageous as the new PLC can be integrated into a testbed for ongoing cybersecurity research that the university is taking part in. The second part involved the layout and configuration of Ignition from Inductive Automation, which is the supervisory control and data acquisition (SCADA) software platform that will serve as the human-machine interface (HMI). The Ignition software is an affordable, highly modular, and user-friendly option for the SCADA interface that will make changes to the NCREPT system simpler in the future. The project culminated in a successful test of both the PLC and the interface to control one breaker in the facility. This project is a step towards keeping expertise within NCREPT and reducing the need for costly outside consultants or contractors for future improvements.

Acknowledgements

I would like to thank Dr. Alan Mantooth for first giving me the opportunity to work in his lab two years ago as an REU student, and for continuing to give me opportunities to learn from his program throughout the rest of my undergraduate career. His leadership and vision are qualities that I certainly look up to and admire.

I would like to thank Chris Farnell for being an unmatched source of knowledge on the projects that I have worked on under his tutelage. I am immensely grateful for his patience in answering my questions and always finding time to help even with his very busy schedule. His encouragement as I dug into these projects went a long way. His passion for the work that he does shows, and it is definitely contagious.

I would like to thank Nicholas Blair for being my partner as I did this project. I certainly could not have done it without his help. I am thankful for our (occasional) goofing off in the office. I truly believe it boosted morale. I would like to thank all of the undergraduate and graduate students in the lab that I have been able to learn from.

I would like to thank my girlfriend Mary Helen for giving me support during this project and always being a kind ear to my worries and gripes.

I would like to thank my parents and my brother, without whom I could not have even completed this degree. It is hard to imagine where I would be without their unwavering foundation of support, certainly not here.

Table of Contents

List of Figures

I. Introduction

The NCREPT facility recently underwent an expansion and improvements. This included some changes to parts of the electrical layout of the equipment. As a result, the aging control and interface for the test equipment needed to be changed. The old programmable logic controller (PLC) was upgraded to a Bedrock Automation PLC. The main benefit of the Bedrock PLC is that it is more cybersecure, which means that it is safer from unauthorized tampering. The cybersecurity features will also make it useful for future research of cybersecure power systems. It also allows for multiple communication protocols, the most important of which is the OPC-UA protocol. OPC-UA is a flexible and easy-to-use protocol. The supervisory control and data acquisition (SCADA) interface was upgraded to Ignition, from Inductive Automation, to serve as the human-machine interface (HMI). The main benefit of Ignition is that it is very flexible and easy to use. It can seamlessly connect to the PLC through OPC-UA, however it has multiple avenues for communication as well. The elements of the HMI are easy to modify as needed, and there are many options and tools for designing the HMI. The background, design process, and testing of the new system are discussed in more detail in the following sections.

II. Background

A. The NCREPT Facility

The NCREPT facility was established at the University of Arkansas in 2005 for the advancement of research in power electronics [1]. Power electronic devices have applications in modern grid solutions as well as in electric vehicle technology. These devices, which are pushing the limits of power density and operating temperatures, of course need to be tested as they are developed. The testing of these devices, therefore, often demands large voltages and currents. NCREPT has the equipment necessary to provide these high power values up to 6 MVA.

The testing equipment in the facility includes nineteen low voltage (480 V) breakers, which are named F1 through F18 and one main breaker UM1 from the electric utility transformer. There are fourteen medium voltage (13.8 kV) breakers, which are named MV1 through MV14. In addition to the breakers, there are three 2 MVA regenerative electronic load banks and one 750 kVA variable-voltage/variable-frequency (VVVF) drive. The regenerative load banks are essentially large back-to-back converters that are able to provide dynamic load profiles for testing while recirculating the power sans the losses, thus increasing energy efficiency as well pushing up the ceiling for maximum testing power. The VVVF drive can output AC power at different voltages and frequencies, which can be used for a variety of tests including emulating different grid conditions. All of this large equipment is housed in an area known as the "bay", and some of it can be seen in Figure 1.

This equipment must be controlled and monitored from the safety of the control room during testing. This is where the PLC and SCADA interface come in. The PLC handles the control logic, sending and receiving signals to and from the test equipment and SCADA interface. The

SCADA interface allows human control input to the PLC and monitoring of equipment status signals sent from the PLC. The control room and the old HMI can be seen in Figure 2.

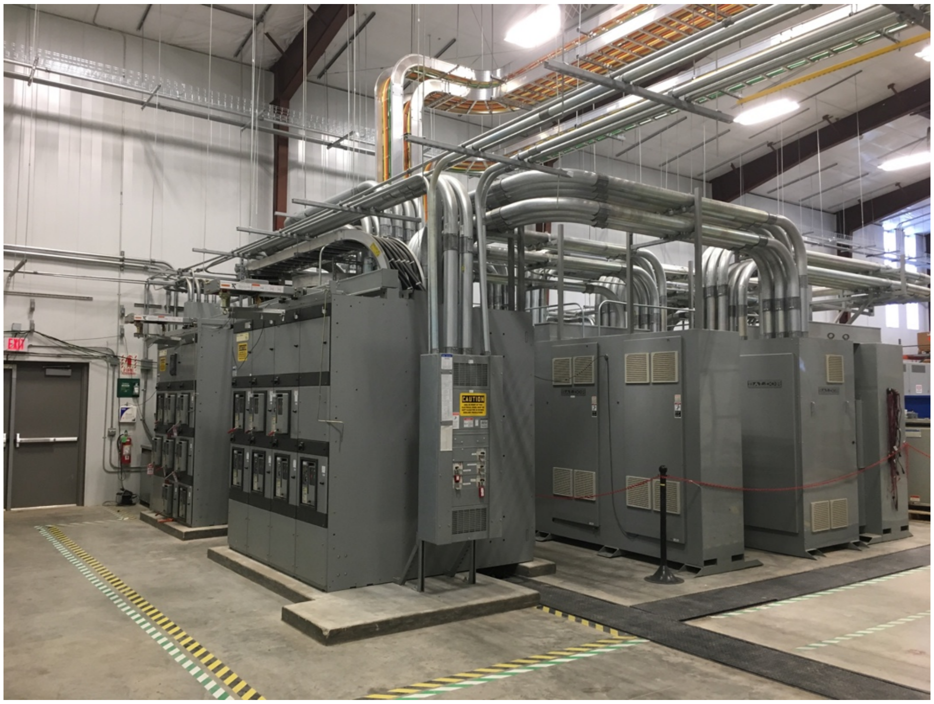


Figure 1: Test equipment in bay.

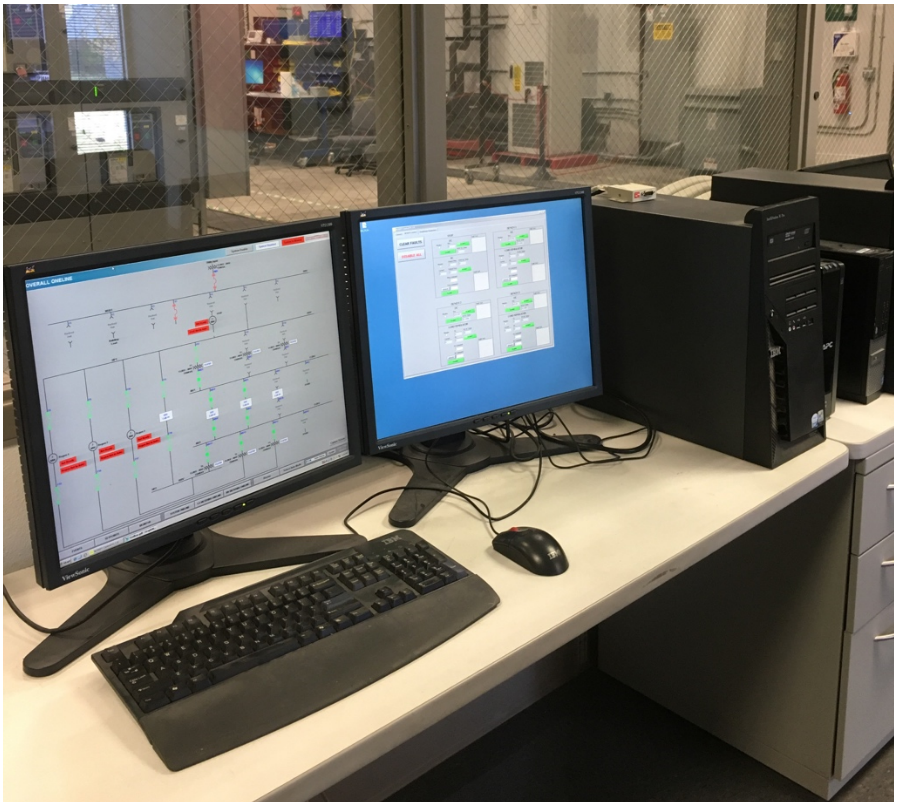


Figure 2: Control room with old SCADA interface visible.

The recent expansion of the facility required that the one-line diagram in the SCADA interface be changed. The facility's house power (lights, wall outlets, etc.) is now fed from a separate utility transformer than the test equipment; previously house power was fed from breaker F2. Breaker F2 now feeds a new AC busway that will be used for testing. There is also a new DC busway that will be fed from a 2 MW DC supply, which is powered from breaker F3.

B. PLC Primer: IEC 61131-3

PLCs are programmed in much the same way as microcontrollers and other embedded systems. The program is written in an integrated development environment (IDE) and loaded onto the PLC. Since PLCs are nearly ubiquitous in industrial control systems, a standard for the languages used to program these devices was developed by the International Electrotechnical Commission (IEC). This standard is known as IEC 61131-3. The standard contains five languages: Ladder Diagram (LD), Function Block Diagram (FBD), Structured Text (ST), Instruction List (IL), and Sequential Function Chart (SFC) [2].

An important element in this standard is the idea of the program organization unit (POU) [2]. POUs are the blocks from which a program is built. Each of these blocks is something like a box that a certain portion of code with a specific purpose can go into. The blocks can be standalone meaning that they can be compiled independently of other POUs, thus allowing for easy modularity. POUs can talk to each other by making calls to other POUs, allowing the overall program to work together. A major advantage is that a POU can be written in any one of the five languages, which allows each POU to be written in the best language for its specific job.

C. Cybersecurity

The NCREPT facility is fast becoming a major site for cybersecurity research as it relates to the energy sector. The University of Arkansas is a member of the Cybersecurity Center for Secure Evolvable Energy Delivery Systems (SEEDS), which is a multi-industry and university partnership to develop cybersecurity technologies [3]. The goal of these technologies is to make the increasingly network integrated and automated energy infrastructure more robust against cyber-attack by those wishing to do harm.

PLCs are often used to control and monitor parts of the energy infrastructure such as the equipment inside substations. Older PLC technology is more vulnerable to cyber-attack. The cybersecurity features were a major point of motivation for selecting the Bedrock PLC for controlling NCREPT. There is a “White Paper Series” document titled “Intrinsic Cybersecurity

Fundamentals" on the Bedrock website that details the cybersecurity features available on the PLC [4]. Figure 3 shows a visual representation of how Bedrock has implemented the features. This main idea of their approach is that the security is built into the devices on every level of their construction.

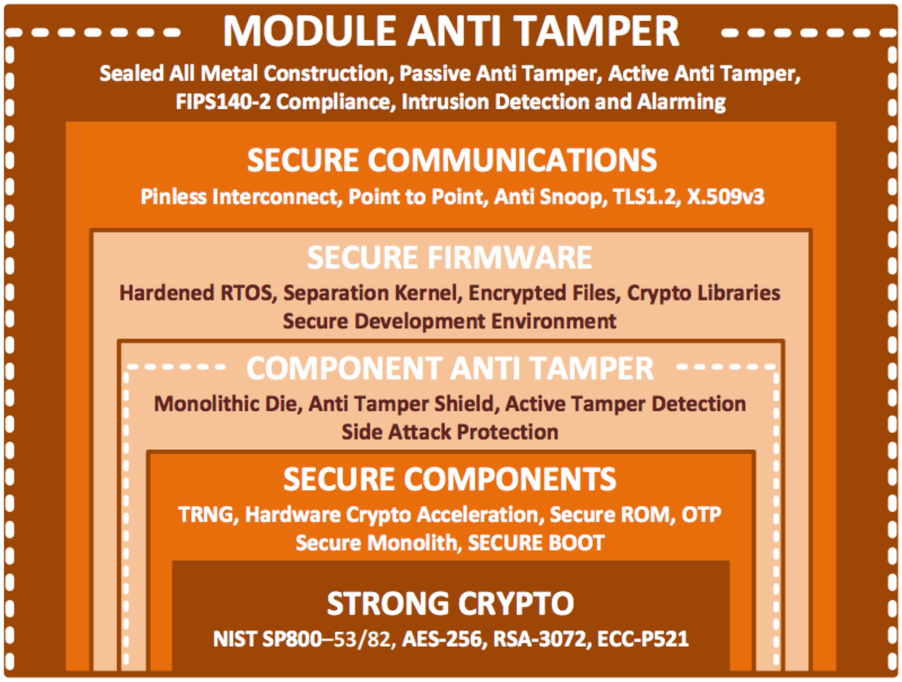

MODULE ANTI TAMPER
Sealed All Metal Construction, Passive Anti Tamper, Active Anti Tamper,
FIPS140-2 Compliance, Intrusion Detection and Alarming
SECURE COMMUNICATIONS
Pinless Interconnect, Point to Point, Anti Snoop, TLS1.2, X.509v3
SECURE FIRMWARE
Hardened RTOS, Separation Kernel, Encrypted Files, Crypto Libraries
Secure Development Environment
COMPONENT ANTI TAMPER
Monolithic Die, Anti Tamper Shield, Active Tamper Detection
Side Attack Protection
SECURE COMPONENTS
TRNG, Hardware Crypto Acceleration, Secure ROM, OTP
Secure Monolith, SECURE BOOT
STRONG CRYPTO
NIST SP800–53/82, AES-256, RSA-3072, ECC-P521

Figure 3: Intrinsic cybersecurity [4].

The powerful cybersecurity features of the Bedrock PLC not only make NCREPT safer from attack by external forces, but can also be of use in the cybersecurity research being done at NCREPT as part of the SEEDS center. The PLC has the potential to be integrated into a testbed with other grid control and automation equipment that will act as targets for emulated cyber-attacks.

D. Cost and Flexibility

Another major motivation for selecting the Bedrock PLC with the Ignition software was cost. An initial consultation with an external engineering contractor deemed that it would cost significantly more than if it were to be handled internally. Since it is handled internally, any future modifications that need to be made can also be done internally, given that a thorough documentation practice is maintained. This is perhaps more important than the lower cost; keeping the expertise within the lab makes the system very accessible and flexible, as well as bolsters students' knowledge of the facility.

III. Design Process

A. PLC Programming

The PLC was programmed using Bedrock's integrated development environment (IDE), which was downloaded from the Bedrock website. The program was written with a combination of Ladder Diagram (LD), Function Block Diagram (FBD), and Structured Text (ST).The old PLC code, which was written in the Unity Pro XL environment, was used as a template to write the new code and served as a starting point. The old code was commented and documented by a past student, which also proved to be helpful [5]. It is assumed the reader has a basic understanding of Boolean and ladder logic.

The old code was first examined closely to gain a sufficient understanding of its structure and function. Writing the new code involved preserving all of the necessary core functionality features and interlocks that were present in the old code, with some changes and simplifications. An interlock is a piece of logic in the code that allows certain actions to be executed only if certain conditions are met. Interlocks are for the safety of equipment and personnel. Some additional interlocks were added to the code that involved the emergency stop (E-stop) buttons that are located at the entrances to the bay. The function of the E-stop buttons is to open breakers during a failure or other undesirable/unsafe event during testing. In the old code, breakers UM1 and F7 are excluded from the coverage of the E-stop buttons. In the new code, every breaker will be opened by the press of an E-stop button. Supervisory control of breakers F1, F2, and F3 were not included in the old code; coverage of these breakers was added in the new code.

Some features of the old code were also removed. The old code had some auto mode features that allowed some breakers to be automatically opened or closed in a timed sequence based on configuration input from the SCADA interface. For increased safety and simplification of operation, all of the breakers are manually operated in the new system.

Other than adding and removing some features, the structure of the code was somewhat simplified. Each breaker was put into its own program organization unit (POU). This way each breaker can be seen as a distinct item in the program, making the code easier to understand. In the old code, several breakers were placed into a single POU. Some of the breaker POUs in the device tree can be seen in Figure 4.

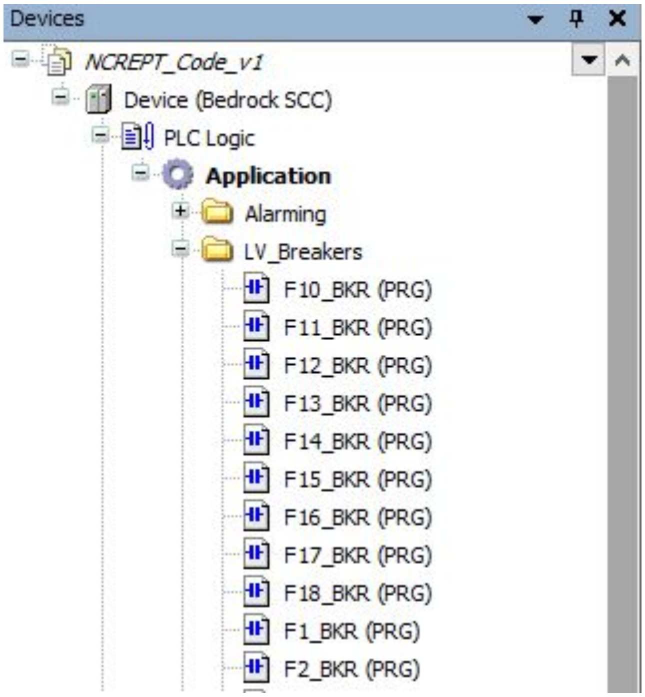


Figure 4: Hierarchical view of breaker POUs.

The operational logic of a breaker in general was put into its own POU named BKR_OPR, so that it could be called as a function block in individual breaker POUs. The BKR_OPR function block is written with Ladder Diagram. This idea was present in the old code. An example of a call of the BKR_OPR function block can be seen in Figure 5, where it is being called within the F18_BKR POU, also written with Ladder Diagram.

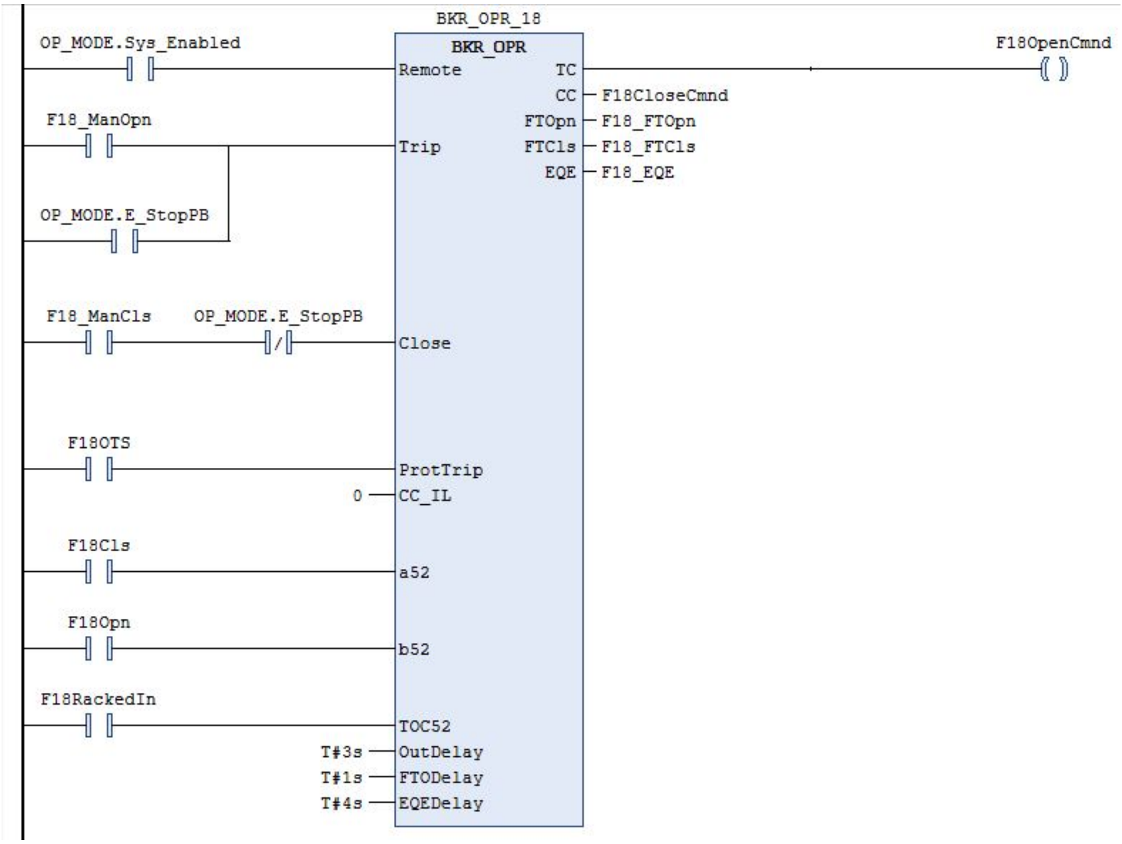


Figure 5: BKR_OPR function block called in F18 POU.

The lines going in on the left side are inputs to the block and the lines coming out on the right are outputs. The 'Remote' input checks for a true signal from the 'Sys_Enabled' variable, which is called from the OP_MODE POU. The OP_MODE POU will be discussed in more detail later on. The system must be enabled via a pushbutton on the SCADA interface for the BKR_OPR block to perform any opening or closing action. However, the Sys_Enabled variable does not need to be true for the breaker to trip due to an overcurrent. The system just has to be enabled for manual control of the breakers. The 'Trip' input and the 'Close' input are checking for the 'F18_ManOpn' and 'F18_ManCls' variables to become true to either open or close the breaker, respectively. The E-stop button input is ORed with 'F18_ManOpn' to open the breaker when the E-stop button is pressed. The negation of the E-stop button input is ANDed with 'F18_ManCls' to prevent closing the breaker when the E-stop input is true. The ‘F18_ManOpn’ and ‘F18_ManCls’ variables are set from two ladder rungs above the BKR_OPR block shown in Figure 6, based on open and close pushbutton input from the SCADA interface. The ‘Trip’ and ‘Close’ rungs *within* the BKR_OPR block are shown in Figure 7. Some conditions have to be met, for example, to set the ‘F18_ManOpn’ variable: the breaker has to be closed (F18Cls = true), the breaker must not already be open (F18Opn = false), the breaker must be racked in (F18Racked_In = true), the breaker must not have a fail to open alarm active (F18_FTOpn = false), the manual close pushbutton must not be pressed (F18ManualClsPB = false), and the E-stop button must not be pressed (E_StopPB = false). The same idea applies to the ‘F18_ManCls’.

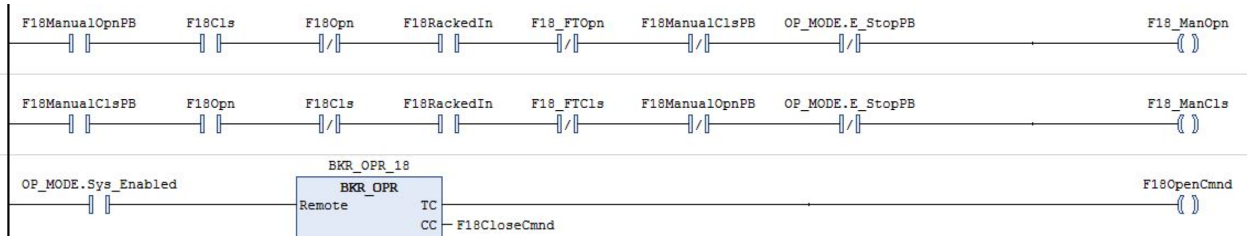


Figure 6: Manual open and close rungs.

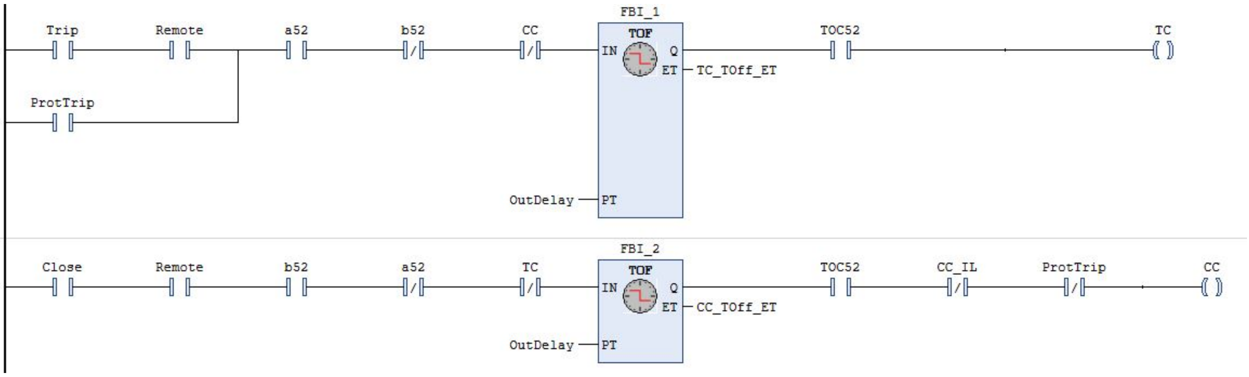


Figure 7: Trip and Close rungs in BKR_OPR.

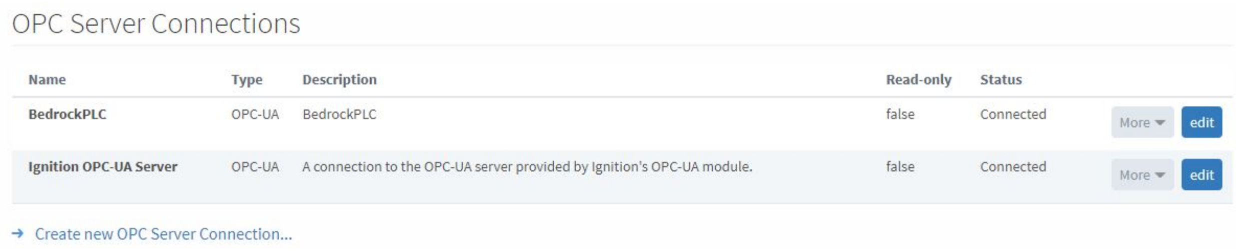
The ‘ProtTrip’ input is checking for a true signal from the ‘F18OTS’ variable to cause a protective tripping of the breaker (sends an open command). The ‘F18OTS’ variable is tied directly to a PLC input coming directly from the breaker that indicates overcurrent trip status as well as providing the stimulus to the PLC for tripping. The ‘CC_IL’ input is an interlock for the closing command; if it is true, the breaker cannot close. However, it is not used for the F18 breaker, so it is hard set to 0. The ‘CC_IL’ variable is only used for the breakers that are tied to the regenerative loads. The ‘a52’ input is checking to see if the closed status of the breaker is true. The ‘b52’ input is checking to see if the opened status of the breaker is true. The ‘TOC52’ input is checking to see if the breaker is racked in. These three status inputs are coming directly from the breaker. The ‘F18OpenCmnd’ and ‘F18CloseCmnd’ variables are tied directly to the output of the PLC and send the signals to the breaker to open and close, respectively. The ‘F18_FTOpn’ and ‘F18_FTCls’ output variables are used to signal that the breaker has failed to open or failed to close, respectively. The ‘F18_EQE’ variable is true when the open and closed status variables from the breaker are in the same state, which indicates the breaker status has failed. These three signals are sent to the PLC for alarming purposes.

The OP_MODE POU contains part of the interlock logic for the regenerative loads and the VVVF drive. The other part of the interlock logic for these devices is in the respective POUs of the breakers that are associated with these devices (F4, F5, F8, F9, F10, F15, F16, and F17). This POU also contains the logic enabling the system. When the system is enabled, the building door locks are activated, a strobe light turns on, and a siren sounds for five seconds.

B. SCADA Interface Design

The SCADA interface was created using the Ignition software platform. Ignition is installed on the computer as server software. It is entirely web-based, and the Ignition Designer and Ignition Clients are web-launched on the computer that has the license installed. The Designer is the graphical development environment for creating the interface. The Clients are the interface windows.

Ignition uses the OPC-UA protocol to communicate with the Bedrock PLC. An OPC server connection was created in the Ignition Gateway, shown in Figure 8. The Gateway is the web browser-based control panel of Ignition, where many different things can be configured. The Gateway is where the Designer and Clients are launched from. Within the settings of the "BedrockPLC" OPC server, the endpoint was set to the PLC's IP address. This established the connection between Ignition and the PLC.



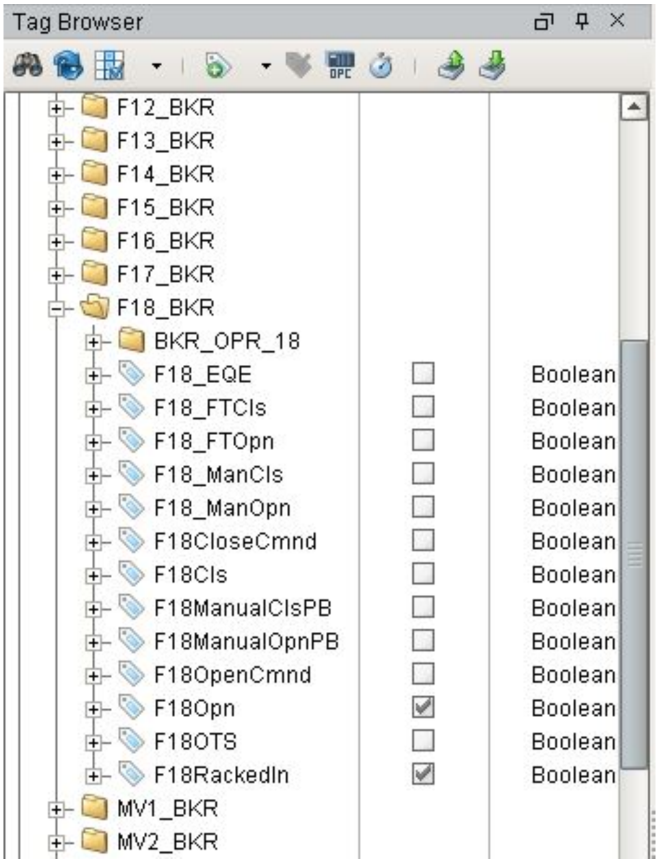
OPC Server Connections

Name	Type	Description	Read-only	Status	
BedrockPLC	OPC-UA	BedrockPLC	false	Connected	More ▾ edit
Ignition OPC-UA Server	OPC-UA	A connection to the OPC-UA server provided by Ignition's OPC-UA module.	false	Connected	More ▾ edit

→ Create new OPC Server Connection...

Figure 8: OPC server connections.

Within the Bedrock IDE, a Symbol Configuration was created before loading the program onto the PLC, which creates OPC tags from all of the selected variables in the program. These OPC tags can be read and written to by Ignition. Figure 9 shows the OPC tag browser in Ignition with the some variables shown as tags.



F12_BKR		
F13_BKR		
F14_BKR		
F15_BKR		
F16_BKR		
F17_BKR		
F18_BKR		
BKR_OPR_18		
F18_EQE	□	Boolean
F18_FTCls	□	Boolean
F18_FTOpn	□	Boolean
F18_ManCls	□	Boolean
F18_ManOpn	□	Boolean
F18CloseCmnd	□	Boolean
F18Cls	□	Boolean
F18ManualClsPB	□	Boolean
F18ManualOpnPB	□	Boolean
F18OpenCmnd	□	Boolean
F18Opn	☑	Boolean
F18OTS	□	Boolean
F18RackedIn	☑	Boolean
MV1_BKR		
MV2_BKR		

Figure 9: Tag Browser showing F18 variables.

The new one-line diagram in the Ignition Client is shown in Figure 10. There are still some details that need to be added to it, however. All of the breaker statuses are showing racked out because only breaker F18 was connected to the PLC for testing purposes.

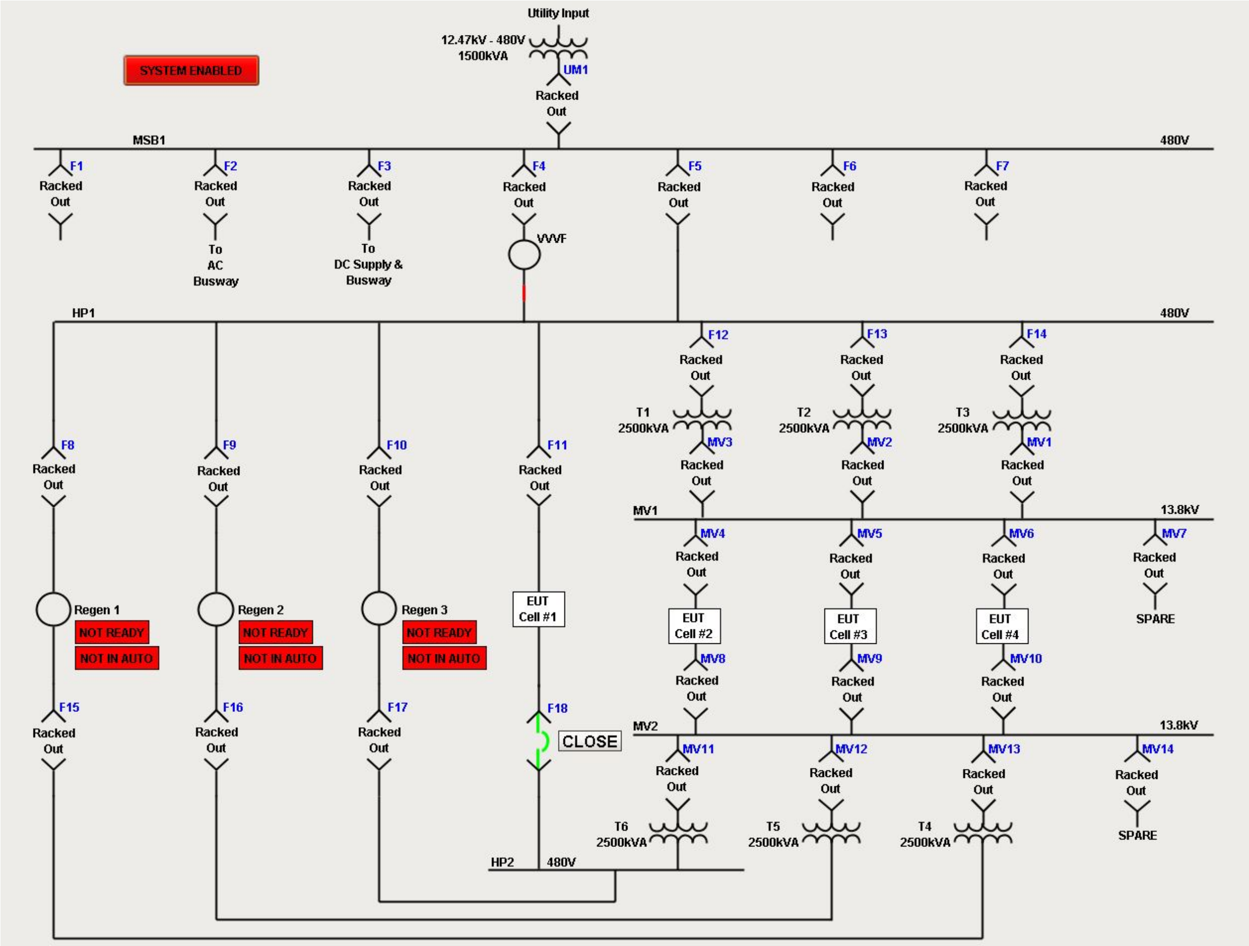


Figure 10: Ignition HMI showing one-line.

IV. Testing

The Bedrock PLC was tested throughout the code writing process, but nothing was connected to the I/O cards during these small incremental software tests. Much of the code was also tested within the IDE using the simulation mode. There were two tests that involved physically connecting the PLC I/O cards with cables.

The first test was a simple loopback test in which an output channel on an output card was connected to an input channel on an input card. This test was done during the initial stages of the project to learn how the PLC and development environment works, and also to verify that the PLC hardware functions properly. Since the output card utilizes an open drain, it does not output a voltage signal. The output channel simply acts like a controlled switch, and an external 24 V DC power supply must be used in series. A simple program to toggle a Boolean true value was loaded onto the PLC, and the output card with the DC supply was connected to the input channel. The true/false status of the input channel was observed in the IDE and changed with the output as expected.

The second test involved connecting the new PLC to breaker F18 to test the open and close commands as well as the breaker status signals (open, closed, tripped, racked in). The emergency buttons were also tested. Only one breaker was tested because if one breaker works, the rest should also work when the new PLC is fully installed. The Bedrock PLC was set up on a table placed in front of the original PLC cabinet in the bay, shown in Figure 11. The old PLC was powered off. The new PLC was plugged into a 120 V socket in the cabinet. An ethernet cable connected to the UARK network was run to the PLC to communicate with the SCADA interface The PLC will be on its own dedicated network during the full installation.

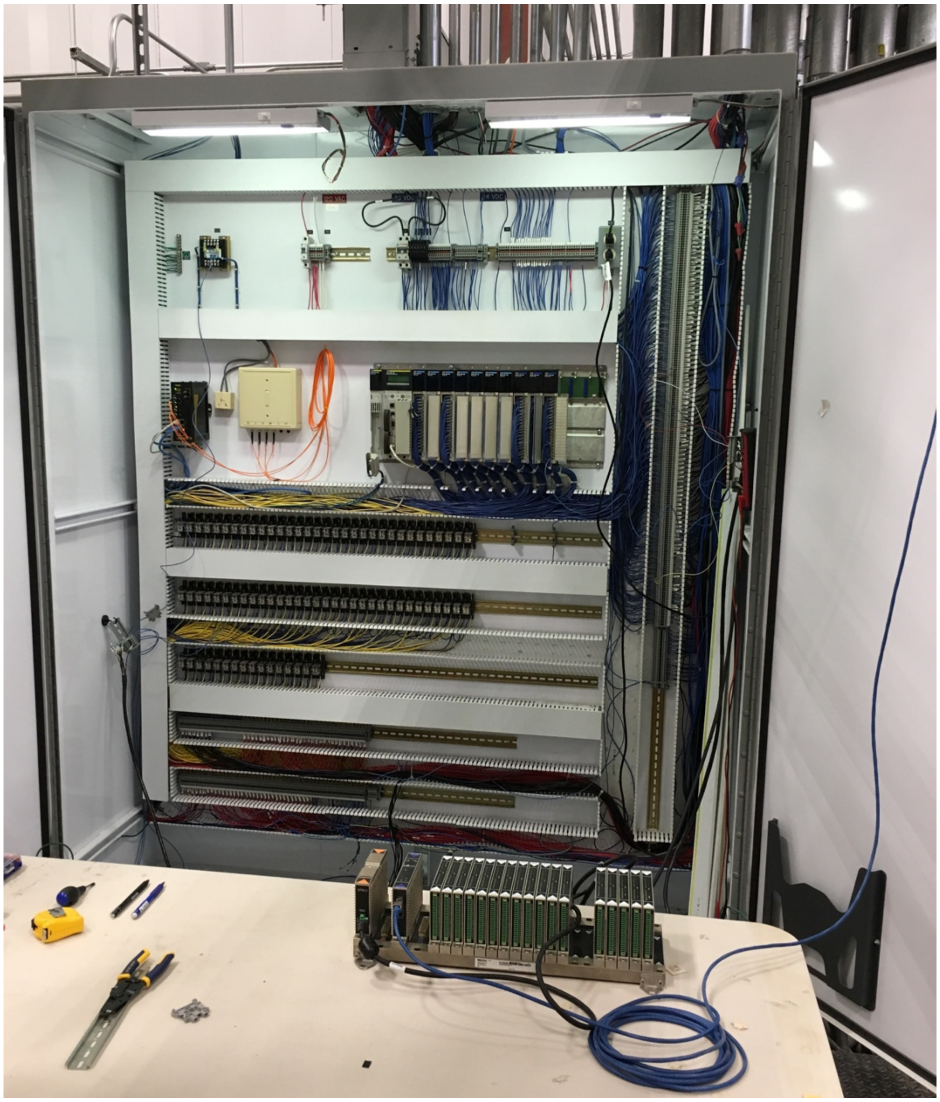


Figure 11: Test setup in front of PLC cabinet.

The input connections to the new PLC were connected to the original terminal blocks, shown in Figure 12. The original input connections were removed. The output connections were connected to the original relays for the open and close commands, with the old PLC connections removed, shown in Figure 13. Because of the open drain configuration of the output card, the 24 V supply of the PLC cabinet was tapped into to actuate the relays. The old PLC output cards actually put out 24 V to drive the relays.

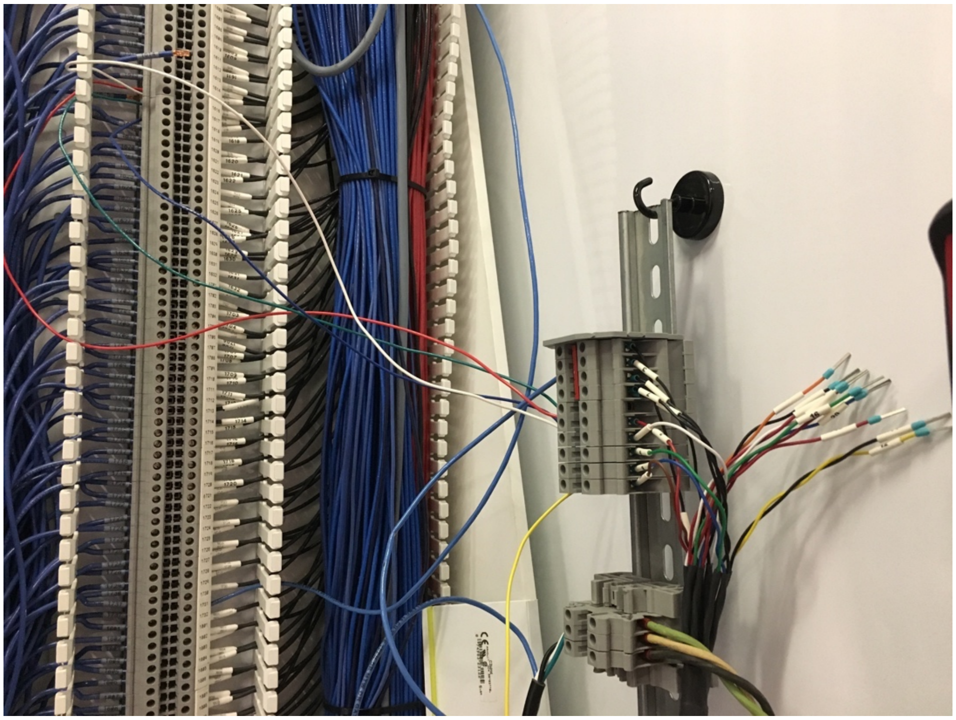


Figure 12: Inputs connected to original terminal blocks.

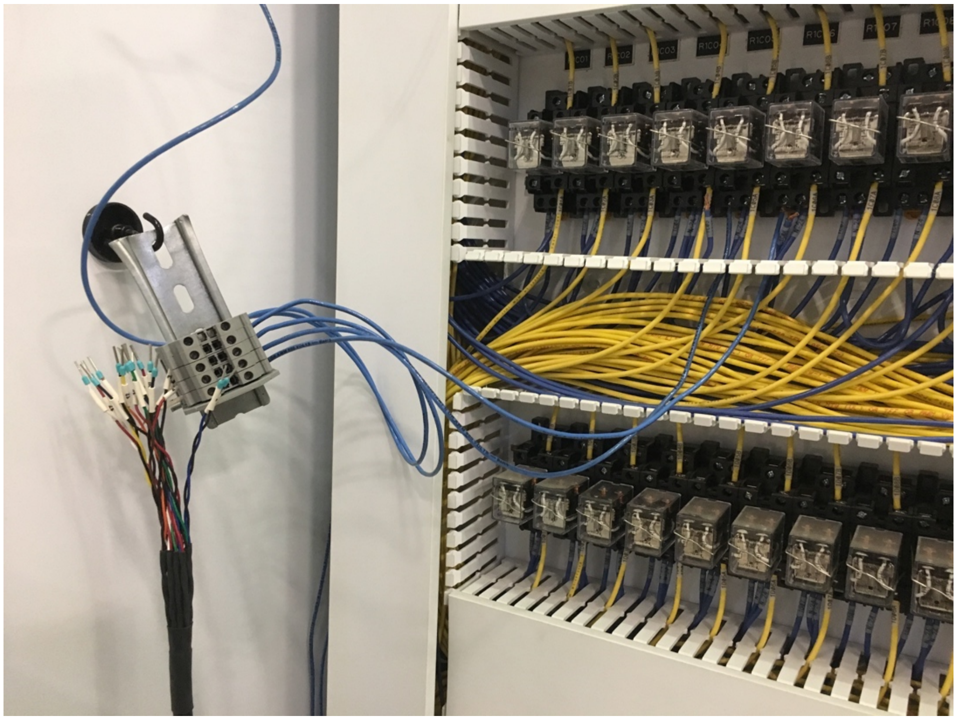


Figure 13: Outputs connected to original relays.

To test this setup, breaker F18 was first racked in. The status on the SCADA interface was observed to change from “Racked Out” (Figure 14) to showing the open breaker symbol (Figure 15). The system enable button was pressed on the interface. This allowed the close button to become visible as well. The close button for the breaker was then pressed on the interface. The breaker was observed to close, and the status on the interface changed to the closed breaker symbol (Figure 16). This also prompted the close button to turn to an open button. One of the E-stop buttons was then pressed. The breaker was observed to open, and the status of the breaker changed to open. Next, the overcurrent tripping function was tested. To do this, the breaker was again closed. The front panel controls on the breaker’s trip unit were used to perform a test trip of the breaker. The breaker was observed to open with the trip alarm flashing on the breaker, and the status on the interface was observed to turn to the tripped breaker symbol (Figure 17). The open and close buttons are inaccessible until the trip alarm is manually cleared on the breaker. The status then resolves to open.

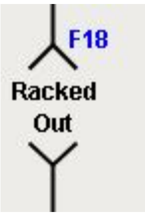


Figure 15: Racked out status.

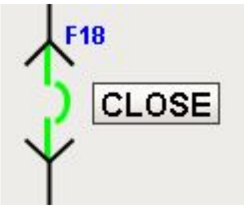


Figure 14: Open status.

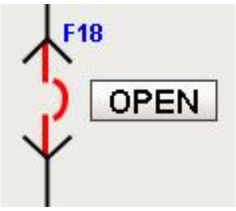


Figure 17: Closed status.

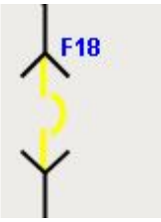


Figure 16: Overcurrent tripped status.

V. Conclusions and Future Work

The conclusion of this project saw the successful test of breaker F18 controlled with the new Bedrock PLC and Ignition SCADA interface. Since it has been demonstrated that the new system can control and monitor one breaker, the next step in the process is to complete the full installation, and tidy up the details of the interface. After it is installed, all of the alarm handling must be configured in Ignition. The SCADA interface is easily accessible by staff and students of NCREPT to make any necessary changes and improvements. This project was one small part in the modernization of NCREPT to further aid in the research of new power and cybersecurity technologies.

VI. References

[1] "NCREPT | National Center for Reliable Electric Power Transmission | University of Arkansas", ncrept.uark.edu. [Online]. Available: https://ncrept.uark.edu/. [Accessed: 23- Apr- 2018]

[2] K. John and M. Tiegelkamp, IEC 61131-3: Programming Industrial Automation Systems. Springer, 2010.

[3] "SEEDS | A Multi Industry & University Partnership", seedscenter.uark.edu. [Online]. Available: https://seedscenter.uark.edu/. [Accessed: 23- Apr- 2018]

[4] A. Rooyakkers, Chapter Three: Intrinsic Cybersecurity Fundamentals. 2016 [Online]. Available: https://bedrockautomation.com/wp-content/uploads/2016/04/Revolution-Chapter-Three-Intrinsic-Cyber-Security-Fundamentals.pdf. [Accessed: 23- Apr- 2018]

[5] Patrick Arreaga, Summer 2014 Research Experience for Undergraduates.