



Murdoch
UNIVERSITY

ENG470 Engineering Industrial Project Final Report

Mine site Commissioning with Programmable Logic
Controllers

Ben Pattimore

1/27/2016

Honours Thesis submitted to the school of Engineering and Information Technology, Murdoch
University, in partial completion of Electrical Engineering and Industrial Computer Systems.

Unit Co-ordinator:

Professor Parisa A. Bahri

Supervisors:

Associate Professor Graeme Cole
Chris Colson

Declaration

All of the work discussed in this report is the work of the author unless otherwise referenced.

I declare the following to be my own work, unless otherwise referenced, as defined by Murdoch University's policy on plagiarism.

.....

Ben Pattimore

December 2016

Terms and Abbreviations

- HMI: Human Machine Interface
- Interlock: a condition that must be satisfied for an event to occur. The event will become unhealthy and stop if the interlock becomes unhealthy during operation.
- Permissive: a condition that must be satisfied for an event to occur. The event will remain healthy if the permissive becomes unhealthy during operation.
- PLC: Programmable Logic Controller, a specialised industrial computer that monitors inputs and energises outputs based on the code that has been written and the inputs energised.
- Engineering Workstation: The main computer in which the required software for which HMI development and alteration can be found
- PF & ID: Process Flow and Instrumentation diagram (P&ID or PI&D)
- PFD: Process Flow Diagram
- Control Philosophy: A document providing an extremely detailed description of how the site is to run from a controls perspective.
- I/O: Input / Output
- MCC: Main Control Centre
- Control Room: room where the system is operated from containing PLC and HMI software/operators.
- Electrical Drawings: Electrician's and draftman's drawings showing the wiring of field and MCC instruments and hardware.

Abstract

This report outlines the industrial control system project of implementing an Allen-Bradley Programmable Logic Controller (PLC) into an alluvial gold mine site. The report will cover the steps taken for the PLC to be integrated into the site based on both the hardware supplied by the client and the control guidelines specified by the client and system owner.

The steps taken to integrate the PLC into three main sections covering the PLC being bench tested to full installation on site, this structure also follows the timeline of the project, first reporting on work completed at the office then moves into a software/hardware test at the warehouse and finally full installation on site.

Note, this report does not detail the code utilised for the inner workings of the live mine site as it is against company policy and intellectual property laws, however it will go in-depth into the communications between the system owner, control engineer and technicians, the procedure taken to get the PLC up to a standard for site installation where all parties write off on the final product and the steps done to troubleshoot and communicate any problems found in any of the stages throughout this project.

Table of Contents

Declaration.....	iii
Terms and Abbreviations.....	iii
Abstract.....	iv
Table of Contents.....	v
Acknowledgements.....	vii
0.0 Introduction	1
1.1.0 Client Readings.....	2
1.1.1 PLC Input Output list	3
1.1.2 Instrument List	3
1.1.3 Process and Instrument Diagram (PID).....	4
1.1.4 Process Flow Diagram (PFD)	5
1.1.5 Control Philosophy.....	7
1.1.6 Error checking documents	8
1.2.0 Hardware/Software	10
1.2.1 Hardware	10
1.2.2 Software.....	11
1.3.0 Networking.....	12
1.3.1 Network Connection	13
1.3.2 Setting PLC IP Address.....	14
1.3.3 Software License	17
1.3.4 Final Network design	19
1.4.0 PLC coding.....	20
1.4.1 Ladder Logic	20
1.4.2 Function Block Design.....	22
1.4.3 Organisation.....	24
1.4.4 Functions.....	26
1.4.5 Testing.....	29
1.4.6 PLC and HMI Roles	30
2.0 FAT (Factory Acceptance Test)/Black-box Test.....	31
2.1.0 Warehouse Purpose.....	31
2.1.1 F.A.T (Factory Acceptance Test).....	31
2.1.2 Black-box Test	32

2.2.0 Testing.....	33
2.2.1 Test Points.....	34
2.2.2 Results Software	36
2.2.3 Results Hardware	40
2.3 Summary Black-Box Test.....	45
3.0 Clampton Commissioning	47
3.1.0 Pre-Commissioning	48
3.1.1 Pre-Commissioning	48
3.1.2 Error Checking.....	50
3.1.3 Clampton Mining Error Check.....	57
3.2.0 Commissioning.....	59
3.2.1 Boundaries	60
3.2.2 Field Safety Mechanism Response and Commissioning.....	66
3.2.3 Sequencing.....	70
4.0 Conclusion.....	73
Bibliography	74
Appendix	76

Acknowledgements

I want to thank Power and Control WA for the opportunity of control system engineer intern, giving me a small glance of the real world of engineering where I have been exposed to a great deal of both theoretical and practical engineering problems ranging from simulations and discussion of how components and software may interact and affect each other to being able to see these components be tested live and installed on site based on these discussions.

I give my greatest appreciation to my project supervisors Chris Colson and Graeme Cole, Chris for teaching me every step of the way what to look and where when facing not just control system problems and tasks but also touching on other engineering professions and how it affects the project as a whole and Graeme Cole who without fail have guided me through the problems faced in my project making this an eye opening and enjoyable experience. I would also like to give a big thanks to Adam Foster whose time, effort and guidance on site did not go to waste adding to my experience of being able to read/write drafts of electrical drawings explaining the practical hardware components making up a gold mine, making sure I was safe and exposing me to a very small part of a technician's/electrician's big responsibilities on site.

I want to also thank friends and family for the support they have given me throughout my education supporting me through stressful times and making me appreciate that putting time and effort aside to pursue hobbies and relax is just as important as a strong work ethic.

0.0 Introduction

The purpose of the report is to inform reader of the progress and objectives that were achieved by the intern during the employment as a Process and Control intern. The focus of the project is to show how an Allen-Bradly Programmable Logic Controller is implemented and commissioned into a green field (new) alluvial gold mine. The project took place at Power and Control Australia between August 2015 and December 2015 and the objectives that will be discussed are listed below.

- Client Document readings
- PLC benchtop testing (hardware/software/networking/code)
- Factory Acceptance test / Black Box Test
- Pre-Commissioning
- Commissioning

First Chapter 1.0 includes documentation of what the client wants the system to achieve, equipment provided with a description of what is integrated into building the system and PLC coding which explains objectives and important functions used for this project.

Next Chapter 2.0 explains what a factory acceptance test and black box test are giving a foundation of goals are to be reached. An explanation of the errors in the hardware, software and documentation are handled and ending the section with problems the author of this report faced when doing the Black-box Test.

Finally, Chapter 3.0 describes pre-commissioning where a description by what method the hardware and software was individually tested and by what means errors were narrowed down.

Among the points listed above there will be focus on the methods of solving problems at each stage of the project by taking into account factors such as time management, cost and effectiveness of the

solution. Another focus throughout the report will be the communication between all parties from different fields (system owner, technicians of different background and control engineers) to make affective decision to compensate for future and current problems during each stage of the project.

(Power Control WA office)

Before starting any coding, researching the design objectives for the system based on the client's wants, research what topologies the network between devices can be used and what software and hardware provided by the client can be utilised to reach these objectives. The hardware that was provided would be the hardware that is implemented in the control room onsite, the purpose of the hardware was to setup how the devices in the control room communicated to one another and the PLC in the Mill Control Centre. Using the four documents provided by the client a design of the PLC code was created.

1.1.0 Client Readings

Understanding the software and hardware used for the project creates guidelines for both the PLC and HMI operation as this thesis is based on the PLC's integration into the system the first objective was to understand the goals for the control system that was specified by the client. Using the four documents together that were provided by the client the basic logic for the control code could be created. These documents were created by the client who is not a Control Engineer, this must be kept in mind during the reading of the documentation as there may be enquiries on how operation of parts of the control system may not be possible without certain I/O or possibly the hardware/software provided does not have the capabilities needed for their requirements.

1.1.1 PLC Input Output list

The PLC I/O list contains all tags that represent all of the analogue and digital I/O in the system, providing the I/O address, description and type (analogue, digital, direct online). It is important to understand that tags are not variables, when creating a tag in the PLC it should be a summation of where and what that tag represents. A tag name may be called "FIT001" this is more than just a name this tag represents a physical piece of hardware out in the field ("Flow indicator transmitter one"). Having appropriate tag names is important for technicians and maintenance so they can easily follow the code and relate each tag to its designated instrument. The tag can then be cross referenced with the PF/ID document to show where in the system flow metre one is installed. (WA C. S., PLC I/O list, 2015)

1.1.2 Instrument List

This document contains all hardware out in the field; the hardware includes all devices that have signals going to and from the PLC such as valves, sensors, pumps and motors and also the hardware that do not have the I/O like hoppers, piping, and cable types. Similar to the I/O list where a name of the equipment is given and if possible a PLC name and address, the instrument list also includes a brief description, cost, amount and brand of the PLC.

It is important to make sure that all equipment required to create a system included in this list and acts as more of a reference to double check the instruments and properties of parts when reading the other three documents. This becomes useful when more information is needed on equipment that is not on the I/O list example. The shape of a hopper is important to calibrate a level sensor for accurate readings or diameter of water piping affects the pressure through the pumps and thus affecting the icon pressure. (WA C. S., Instrument List, 2015)

1.1.3 Process and Instrument Diagram (PID)

The Process and Instrument Diagram is a visual representation of the Instrument list showing all drives, valves, sensors, piping (pneumatic, water and process slurry) and dams with labels used from the tag list. The tag list shows where the equipment is installed in relation to each other. Example, if the drawing shows a water pipe manual valve connected to the sprinklers, which go into the intake of the mill, the valve and water pipe will be orientated in this way onsite. This document is very helpful to quickly see where the surrounding equipment is in reference to each other.

This document is referred to frequently with the Control Philosophy and the I/O list for quick referencing. When reading the I/O list to find the purpose of a particular I/O e.g. level sensor four can be found on the PID and a quick understanding of its operation in context of the surrounding equipment can be made. When the Control Philosophy talks about the permissive and interlocks of a particular drive the PID can be used to identify where these permissive and interlocks are making the Control Philosophy easier to understand and an understanding of how the system will operate on sight. (WA C. s., 2015)

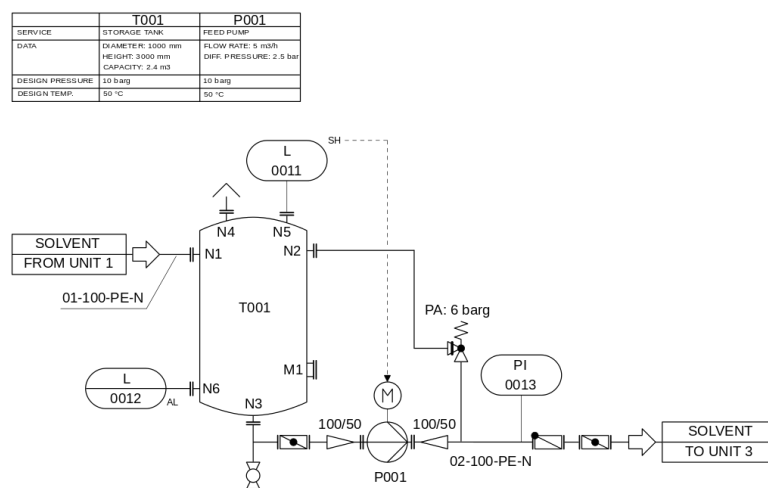


Figure 1: (Con-struct, 2012) – Process Instrument Diagram Example. Diagram represents I/O tags and instruments in relation to each other.

1.1.4 Process Flow Diagram (PFD)

The PFD show the process and the interactions between larger components in the system such as drives, tanks, piping with units representing the state in each step in the system such as mass flow rate, pressure, temperature and density.

When reading this diagram an understanding of each step in the system moves into the next can be followed and is very helpful in creating sequences. The PFD does not show the same level of detail as the PID, since it only draws the connection and direction from one-unit operation to another the sensors and the names/details of these instruments cannot be seen in a PFD as shown in Figure 2.

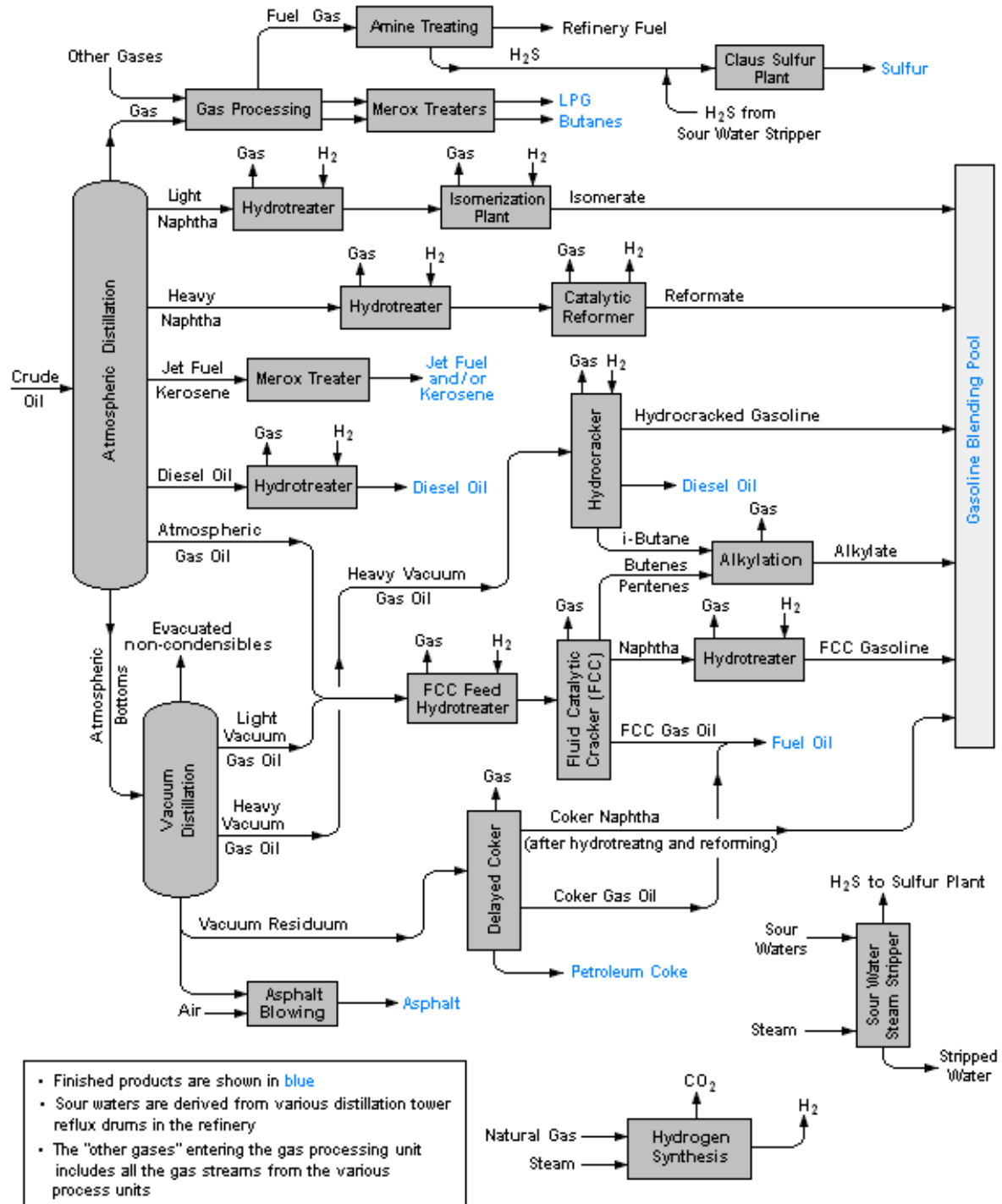


Figure 2: (Mbeychok, 2007) – Process Flow Diagram example. Representing the control flow of the system, showing each state and its relationship to each other state and system as a whole; and the process to transition to the next state.

1.1.5 Control Philosophy

The Control Philosophy is considered one of the most important documents as it contains the client's control objectives for the control system. This document contains all drives with a brief description of the drive's purpose and place in the system, also the permissive and interlocks of that drive. The PF/ID is used to gain a better understanding of instruments in reference with the entire system, while the I/O list is used for more information on the properties of the instrument. The Control Philosophy purpose is to use these references and explain how the drive will operate and is the base of which the code will be built on.

The Control Philosophy represents the entire system and cannot be created by one engineering profession, when designing the system multiple disciplines must work together to create an optimal system. The opinions and boundaries provided by each engineering field must be compensated in the design. This document lays the foundation that this system was built on, explaining how it will operate and what the purpose of each stage its operation is to achieve.

The engineers base their own design for the site on requirements set by the Control System engineer such as amount of power required, motor purpose, container dimensions and pipe requirements. The Control System engineers would then receive the boundaries and requirements of the instruments and devices chosen by these engineers and use the boundaries to build the PLC code that integrates these parts into a working system.

The information that the Control Philosophy aims to provide is the operation of the plant, sequences, interlocks and permissive, control loops, different types of alarms, alarm handling, and functions available to operators, maintenance, references to documents mentioned in this section.

The PLC contains the rules the site will follow. This is important to keep in mind when reading the Control Philosophy. As mentioned the Control Philosophy contains important information in regards to coding the PLC such as the purpose of the control loops, the interlocks and permissive of drives, operation of each I/O in the system and is heavily referred to when creating the control code.

A second Control Philosophy is created after the commissioning has been completed. This document will present the optimal condition for the best performance for the system and a detailed explanation of how the system runs. The document is a manual for operators and maintenance for use of the system once the Control System engineer has completed and left the project.

The Clampton control philosophy could not be included in this paper due to the confidentiality policy of the company. (WA C. S., Control Philosophy, 2015)

1.1.6 Error checking documents

All documents are related to each other so when reading it is important to check for errors made by the client. This is done by cross referencing the PLC I/O list with the PF/ID to make sure that the correct assignment of tag names is used in the drawing. I/O devices on the PLC I/O list should be listed on the Instrument list and all names in the Control Philosophy must be those in the I/O list.

When an error is found or in any aspect when the control engineer does not agree, a document is created listing the errors, enquiries and possible solutions, this list will be sent back to the client to review. The client will be in communication with the control system engineer negotiating a solution that meets both party's needs.

Two examples of errors found in the Clampton documents are shown below:

Example one: it was found that a tag in the PF/ID was different to the associated tag given in the I/O list; this was then further cross referenced with the Control Philosophy indicating that the tag being

false. Finally, an electrical wiring diagram was checked, it was found that the tag was incorrectly used by the drive. If this error was not picked up it could create the wrong logic for the wrong drive.

Example two: when reading the PF/ID it was noted that one of the pumps did not have inputs back to the PLC but did have outputs to control the speed of the pump. This meant that the speed the pump was actually no known, only the speed sent to the pump.

The earlier the errors are detected the faster they can be fixed because it is still in the building stages of the project. This also means that it costs less for the client to implement any changes in the Mill Control Centre. If errors are solved early enough it does not add to the cost, though if left until the factory acceptance test it is not just time consuming for programmers but also for electricians. This is due to the electricians laying and rewiring cables.

Another altercation could occur if the errors were not discovered until implementation on site, which will result in metres of cables that need to be re-arranged and rewired by electricians (as was found on site) costing the client in time and money.

1.2.0 Hardware/Software

Once the documents have been read, an understanding of what the client goals were outlined and any changes to the documents were recorded and sent back to the client, the next step was to check the hardware and software that would be used for the project, such as the software vendor used to code the PLC, which PLC would be programmed on and what hardware would be needed that was provided by the client.

1.2.1 Hardware

List of hardware provided by client:

- Dell PowerEdge server
- Server rack cabinet
- Rack mounted network switch
- Allen Bradley PLC 1756
- Three client computers
- Engineering workstation
- Multiple monitors
- Multiple keyboards
- Multiple mouse

Various problems were found with the hardware. Some of the computers were unable to boot or unable to find the hard drives. Meanwhile the server had a username and password that was not provided by the client and had preconfigured settings on the server. These issues were documented and the document was sent to the client as this was more an IT problem than a PLC problem.

With the hardware provided a simulated room was set up that would replicate the environment of the PLC communicating with all devices in the control room on site.

1.2.2 Software

List of Software provided by the client:

- Rslinx
- Factory talk view site edition
- Factory talk view machine edition
- Rslogix 5000

Factory talk view site edition and machine edition are both used for the human machine interface creation but they will be only referred to not be discussed in detail. Rslogix 5000 was the software used to create the PLC code. The version of Rslogix that was preinstalled on the PC was Version 17. The purpose of Rslogix 5000 in the design phase of this project was to create PLC code to control the given system and to distribute data to the HMI so that it can be displayed to operators. The decision was made for all manipulation and scaling of data from the field to occur within the PLC and not shared between HMI and PLC. This makes future error checking easier e.g. if an unexpected output or input is occurring the engineer will know that only manipulation of data is within the PLC as the HMI only receives and displays data hence the error must be within the PLC code.

Rslinx is stand-alone software but is packaged with Rslogix 5000 and acts as the communication between Rslogix and the PLC. When Rslogix is correctly configured it allows Rslogix to see and use values coming in from all I/O on the PLC and even though it is separated to PLC programming it is just as important in understanding how it is used and how to setup the network between PLC and Rslogix.

1.3.0 Networking

Designing a network is a necessary step while still in the design phase of the control system. The network defines where values will be held from the PLC and what PCs have priority of software and access to administrative settings. The server would hold the PLC values, software keys, alarms, trends and HMI pages that would be called upon by the three PCs where each would have their own allowance and access to different data and software keys depending on the setup. Figure 3 shows the Clampton network topology where distribution of data is controlled by a network switch (star topology). (Rockwell Automation literature library, 2012)

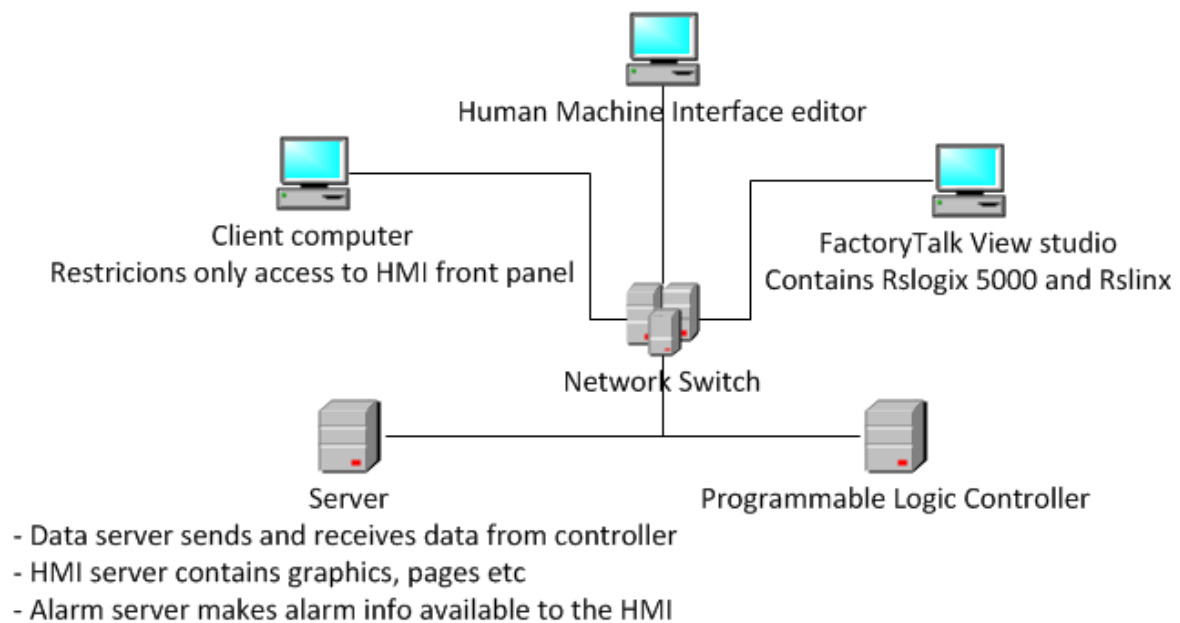


Figure 3: Network design of Clampton Control room. As shown above the client computers communicate through the network switch that allows access to the PLC and server to request or write data.

1.3.1 Network Connection

Before the network could be setup the control system engineers needed to check if all devices could be connected to the PLC and one another. By not setting up the network each device could be isolated and individually connected to each other via the Ethernet port on the back of each PC and the Server. As the IP address of each PC was known and because all PCs had compatible IP addresses they should be able to ping each other. By using the cmd.exe software on all windows computers and the ping function, each computer was pinged to check that all Ethernet cards and cables were functional.

It was noted the IP address of the PLC was displayed on the LCD of the PLC and was on a different network to all other IP address found on the PCs and server.

An example of the IP address found for the devices is:

Client computer1: 192.154.13.1

Client computer2: 192.154.542.13

Client computer3: 192.154.78.3

Server: 192.154.95.44

PLC: 195.155.45.78

As shown in the above the example, network address for the PLC was different, so without pinging the PLC it was known that the PLC would not be found on the network. Instead of changing each device to the PLC IP address it was opted to change the PLC to be on the same network as all other devices, the software used to change the PLC was software provided free by Rockwell called BOOTP DHCP Server Software.

1.3.2 Setting PLC IP Address

Connecting the PLC to the network is an important step in the design phase because a mimic simulation of the control room can be created for testing and optimising before going on site. Due to the importance of communication the steps of changes the PLC IP address are explained below. This documented procedure was one of the activities of the author during the project.

Because the PLC had a different IP address, this meant that it was on a different network to all other devices, although being connected to the network switch, it would seem invisible to the other PCs and the software Rslogix 5000 as well this would result in no data transfer between PLC and the software, even though the code could be created in Rslogix. As a result, the HMI would receive no values as to display to operators. With BOOTP DHCP Server Software it is possible to change the IP of a PLC so that it can included in a specified network. The following steps have been done to change the IP address.

Step 1: power on PLC and connect the communication card of the PLC to the Ethernet port of the PC using an Ethernet cross over cable. An Ethernet crossover cable is used when devices are to be connected directly and these devices are of the same type. This is because if it is a straight through cable the pins for connecting the transmit signal should be connected to the transmit pin on the other device and similarly for the receive signal. When using a crossover signal, the transmit and receive pins are switched over so that the receive pin on one device will line up with the transmit of the other device resulting in successful transfer of data.

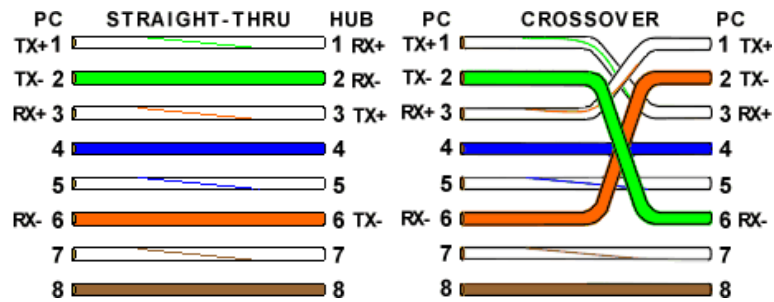


Figure 4: (The Internet Centre, 2015) – *Straight through vs. Crossover diagrams. If the two devices have the same port structure connecting them directly will not work as the TX will connect to the other devices TX not allowing communication. This is why a crossover cable which is required, allowing the TX connect to the appropriate RX.*

Step 2: Open up BOOTP DHCP Server Software (Figure 5) it will take some time for the “request history” registry to be filled up with requests of all devices connected to the PC. Because the PC has been isolated, the only device connected is the PLC which will be the only device being continuously requested by the software. The software will show the time of request, the type of data transferred which in this case is a BOOTP data packet, the MAC address and the IP of the device which in this example is 195.155.45.78.

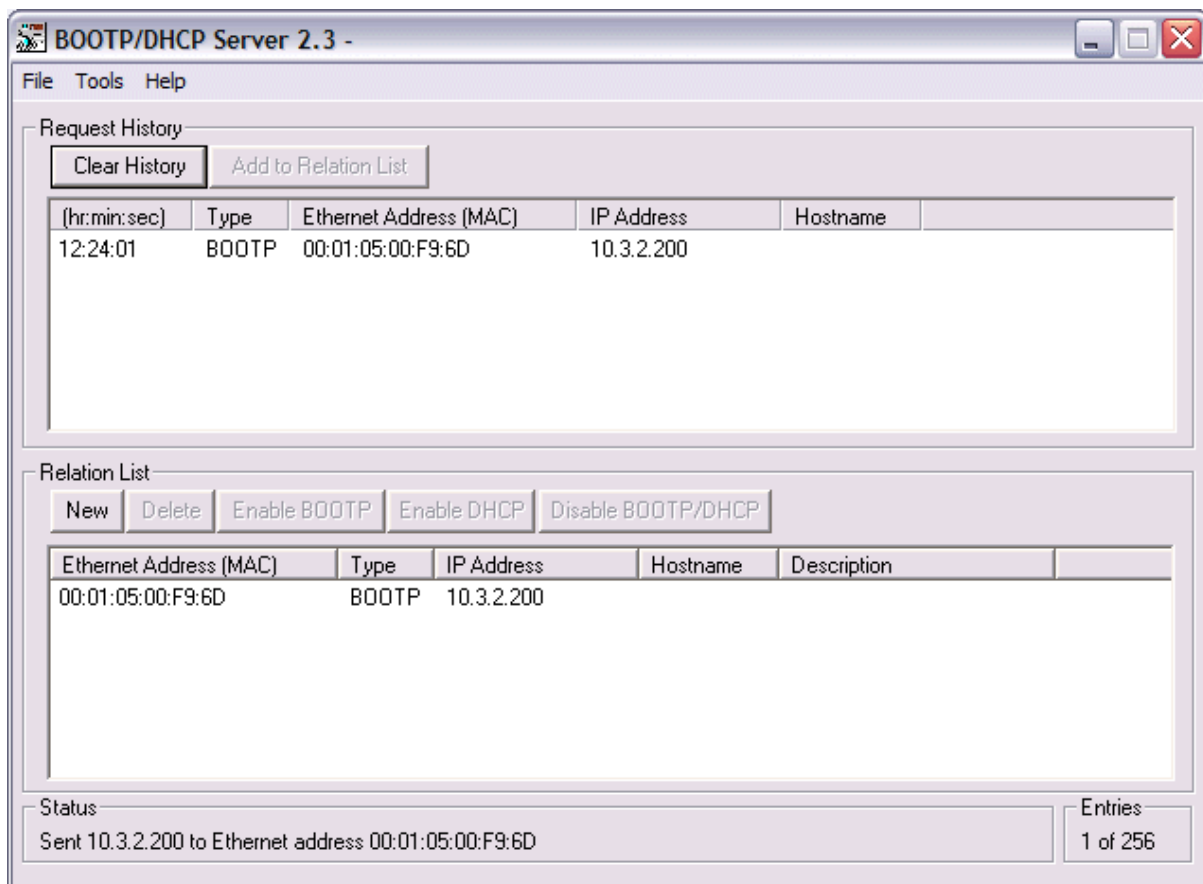


Figure 5:(Addressing via Rockwell BOOTP/DHCP Server 2.3, 2015) – “BOOTP/DHCP Server front panel”. Displays all available devices connected to the network giving their IP address and the MAC address.

Step 3: Choose the packet from the “request history” registry, which has the same IP address as the PLC to be changed and add to “relation list”. This will give the option to change the IP address of the PLC.

Step 4: Now the PLC has a temporary new IP address. When the PLC is turned off and turned back on again either a default IP address or an IP address that had been given to it by a server or host. To avoid this change, in other word to keep the temporary IP address permanently choose the new PLC properties found in the “relation list” and click “Disable BOOTP/DHCP” (Allen Bradley . Rockwell Automation, 2015)

Contextual information:

- BOOTP (Bootstrap Protocol) is a protocol for IP networks where a network address will change its IP address to one provided by a server automatically (Allen Bradley . Rockwell Automation, 2015)
- DHCP (Dynamic Host Configuration Protocol) this is when a device automatically requests IP addresses and other network settings from a configuration server (Allen Bradley . Rockwell Automation, 2015)
- MAC address (Media Access Control Address) also goes by physical address is a unique address given to the physical hardware of a network interface, all MAC addresses are unique. (Allen Bradley . Rockwell Automation, 2015)

When choosing the option to disable BOOTP/DHCP, this means to remove the option for the PLC to change or accept a new IP address. Because the MAC address is unique to the hardware unlike the IP address which can be changed this means if there were multiple devices on the network the MAC address of the PLC can be used to find the right device in the “request history” when using BOOTP DHCP Server Software.

1.3.3 Software License

After all of the IP addresses of the devices are on the same network they can communicate to each other. The designed a network chosen has been as a star topology, including a network switch, a server, four computers and PLC. Later, in the commissioning stage this network design can be changed, still having a star topology but with only two computers to make a much simpler, robust and suitable network for the small mine site and small control room.

When all of the devices were connected it was found that the license keys for the software were not stored in the client computers but were held externally on the server. As the client did not provide a password or username there was no access to the server settings, it was assumed that the server was used to just store data for the HMI and PLC such as trending graphs, alarm warnings, variables

etc. it was soon discovered that the server distributed computers specific licensing depending on the settings of the server and that allowed the client computer to borrow the license for a set amount of time. While the engineering work station had all licenses needed for the software installed it also had licenses configured to unknown PCs leading to believe there were originally more PCs in the network. (Allen Bradley . Rockwell Automation, 2015)

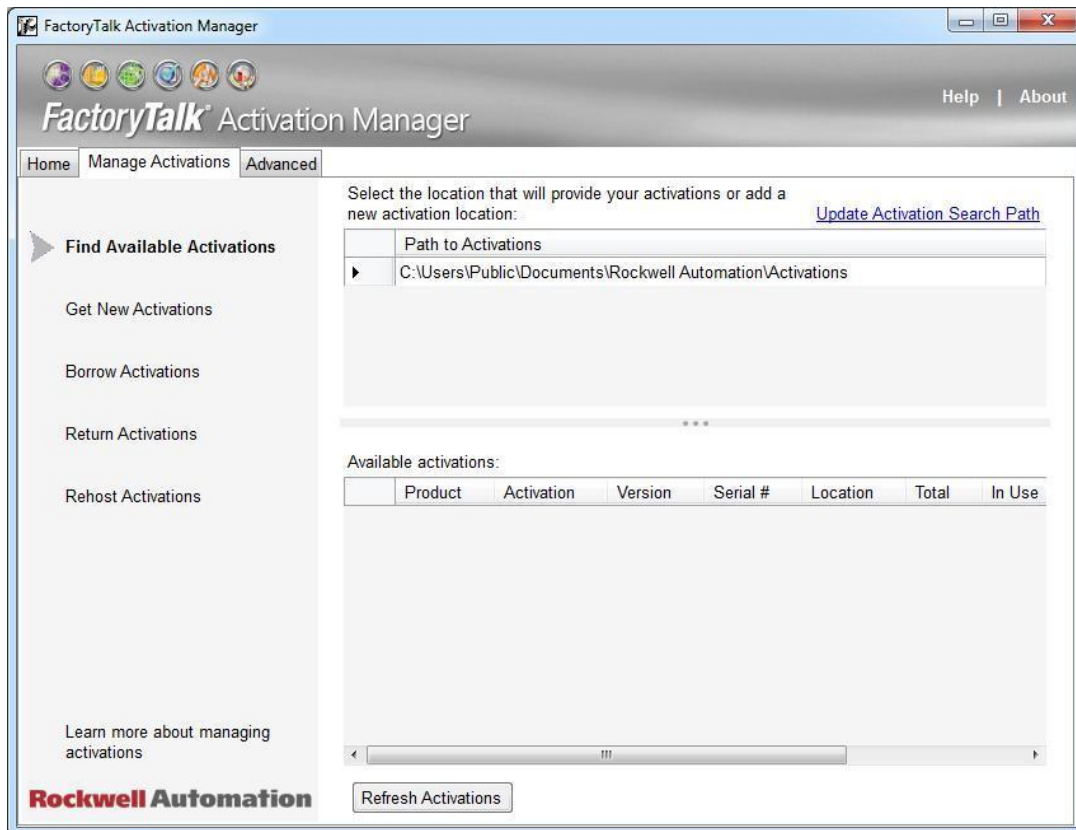


Figure 6: “Activations Window”: This software is used to manage license keys. Checking for available licenses on the PC and connected dongles, which can then be distributed to other computers or used by the managing PC.

The major problem occurred was the server did not allow access to keys to be used on other computers. Attempted to remove licenses from the server through an Ethernet connection to another computer was unsuccessful. Until the client provided the username and password PLC coding could only be done on trial versions still had all the capabilities of Rslogix 5000.

Once the Client provided the password to the server it was decided to move all HMI licenses to the respective computers and Rslogix 5000 to the removable USB dongle. This was done by re-hosting all licenses from the server to the engineering work station using the "FactoryTalk Activation Manager" re-hosts capabilities. It is important to read all documentation before re-hosting licensing when using FactoryTalk Activation Manager. Once the license keys to be re-hosted have been selected it will then be deleted from the local computer and a FactoryTalk Activation data file will be created with a pop up of the host ID of the local computer, serial number of key, the re-host code will be displayed. The next step is to obtain the activation file from the offline computer and go to a computer with internet access to upload the file and input the host ID of the local computer, serial number and re-host code to the Rockwell Automation activation website for verification.

Once these steps have been achieved, Rockwell will confirm that the license is not bound to the initial computer and can now be re-installed, reused and bound to the MAC address on another computer or dongle. (Allen Bradley . Rockwell Software, 2010)

1.3.4 Final Network design

The password and username for the server was provided by the client and became apparent when looking through the server settings that the server used internet explorer to communicate between devices. Internet explorer is used in these sites so that the operators or maintenance can access HMIs from anywhere that has internet access. The simulated network was not given internet access because the software pre-installed on the computers was dated. In other words, once connected to the internet the software would try to update firmware or drives. Having inconsistent software and hardware firmware versions in a network could lead to the software and PLC incompatible communication or between PLC and HMI.

This would explain why the server did not allow access to license keys as it was trying to find an internet connection for verification within the isolated simulated network. This was a problem as the

control room on site will not have access to the internet as it was a small site with only two computers making the server unnecessary.

The star topology was kept but after the removal of the server and because of the size of the control room, number of technicians on site and size of the system there was only need for two computers and PLC with a network switch to communicate between devices. One computer would have the HMI and the software to edit the HMI while the other would hold Rslogix 5000 where the code can be run and changed from.

1.4.0 PLC coding

The organisation of how the code is presented and arranged makes troubleshooting easier for the control engineer. Also when the site is commissioned the code would be easier to navigate for technicians allowing easier editing of the code. It was decided for this project that the program will be written in Function Block Design as there was a lot of repetitive code, a rundown of the main functions used for coding the PLC in regards to the Clampton mine site will be explained in the following section.

Note: The specific code and structure used for the Clampton site will not be shown because of company confidential policy.

1.4.1 Ladder Logic

PLC programming is very unique compared to other types of coding as it is a visual based code using graphics that represent switches, relays, contacts buses, rungs to manipulate values. The design is to make troubleshooting easier for users who do not have a strong background in coding such as electricians or technicians; often these programs are used in association with a HMI program which was the setup for this project.

It is easier to imagine ladder logic as a rule based language unlike most other languages which can be seen as procedural (BASIC, C and Pascal). A line or rung in ladder logic which may have inputs, timers, switches etc. represents a rule. Example (Figure 7) only after Pushbutton I:1 (input one) goes high will Light O:0 (Output zero) go high where by a procedural program could call a routine or function at any time during the reading of this rung.

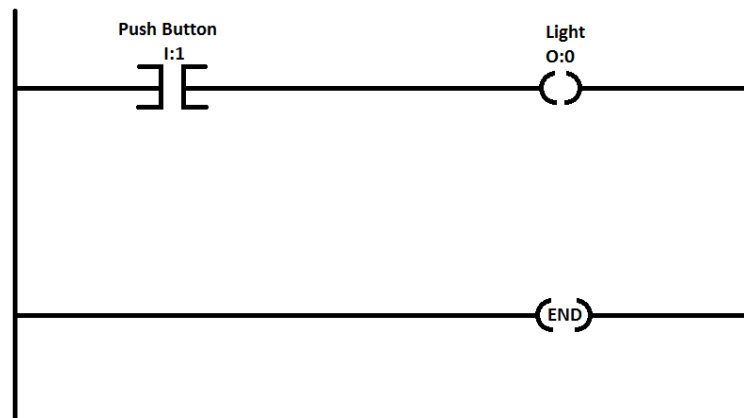


Figure 7: Rule based code: Only if “Pushbutton” is energised will “Light” energise, END signifying the code has finished.

With this in mind it is easy to understand the use for the PLCs when creating a system that operates based on rules. Using ladder logic makes these procedural rules easier to follow. Adding, removing and editing logic in the code are made simplistic compared to other languages because of its visual design. It is much easier to trouble-shoot problems when errors in the system occur as everything is sequential, which is a key point. Since as once the code has been created by the control engineer usually a technician or electrician may be left to fix or maintain the system an easy to follow language is important.

1.4.2 Function Block Design

Function block diagram helps to understand the stages that a system takes and the effect and relationship these stages affects and relates other stages in the process as the part of a whole. It can be seen as a layer on top of ladder logic where by each function block has ladder logic written inside.

There are two parts to a function block, what is happening outside and inside of the. For example, the code of a function block for a level switch includes input names to a block such as alarm high, alarm high high, ready signal, alarm low, alarm low low, alarm delay and output names from the block like status, alarm delay, ready. These names are not bound to any variable and are considered open. Tags/values are wired to will send values into these function blocks or tags that the function block will send to. The tags used may have data coming from the field such as a drive sending a ready status to the associated tag or maybe an absolute value like an integer to set the parameter for a high alarm.

I/O from a function have associated variables inside written in ladder logic. For the input side the variables such as alarm high or ready signal are waiting for tags or values to be written into, simultaneously the outputs are waiting for variables status, alarm delay etc. to send signals to the outputs associated to them.

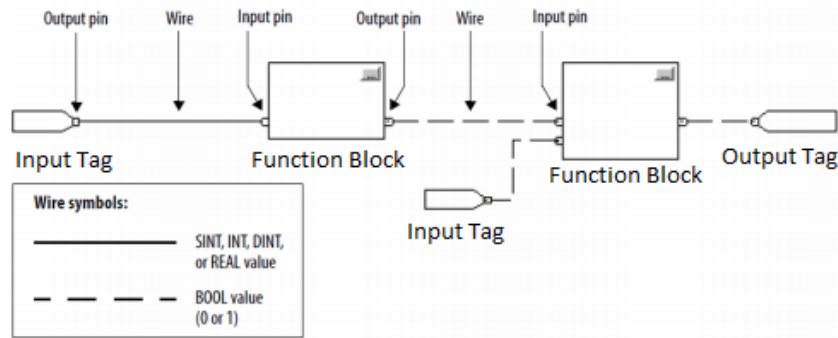


Figure 8: (Stienecker, 2011) – Example of what could be found inside a function block where “SENSOR I:0” on rung 0000 would be reading values from the input pin named “SENSOR” and “OUTPUT1/2 O:0” would send to an output pin which could lead to an output tag or directly to another function block.

As a result, the function blocks can be cloned to make systems easier to navigate as each instrument has its own block and a visual display of the flow of the system, which can be seen as one function block directly moves into another. For example, the outputs of a flow metre will go to the inputs of a pump so that it can control the speed of a pump. This is more organised than reading hundreds of rungs of ladder logic as shown in the above diagram. Looking at a flow metre differentiating which part of the code is data processing, and which parts send data to the next step in the sequence is much more difficult to navigate in ladder than having separate function blocks grouping these pieces of code separately. (Pattimore, Move Function Block, 2015)

1.4.3 Organisation

Arranging the code so that it is easy to follow is important when writing any type of code, it is far from ideal to create one program to be uploaded to the PLC that has only rungs and rungs of ladder logic. This makes troubleshooting difficult as there was no documentation or separation between instruments making navigation slow. Adding code would be difficult because none of the code is isolated so the new code or re-arranged code could affect operation somewhere else in the program and adding code in the wrong section may change the sequence for the operation of a device.

This was one of the reasons to choose function block design was chosen, the operation of an instrument can be summarised in a subroutine by a single block which would then process the tags internally. Folders created with headings to group similar types of subroutines so that navigation was made even simpler to find the associated blocks meaning valves, drives, analogue and digital blocks were all stored in their own folder. This would be helpful when in the commissioning and Factory Acceptance Test stages of the project as finding specific tags; rungs or entire processes would be quick and easy.

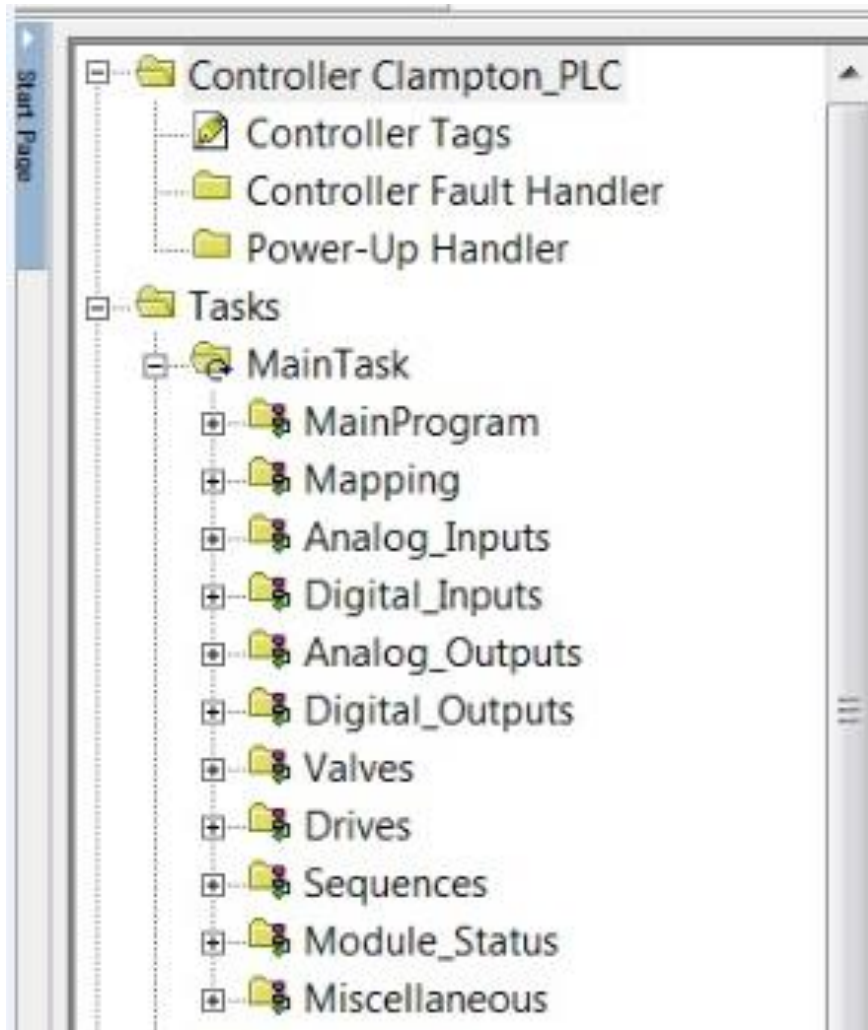


Figure 9: Rslogix 5000 “MainTask” file contain “Drives”, “Pumps”, “Digital Output” etc. folders that are used to group each function into its appropriate file containing other subroutines and functions for each drive, pump and digital output in the Clampton project.

1.4.4 Functions

Important ladder logic functions related to the coding of the project will be explained with relation to how it works within this project. Keep in mind company policy does not allow the use of specific code to be used in the report.

Move (Mov) function: allows an integer to be sent to a destination tag from a source tag. This was used repeatedly for fault control of drives. For example, if Pump one has an input of ten percent and suddenly faults and shuts down, it is necessary to start backup pump two immediately so that the system stays in control without having to shut down the entire sequence. When pump one shuts down it may not be running but there is still an input signal of ten percent of this value to be sent to the move function and directly to pump two (Figure 10). This results in pump two instantly having the value of pump one, rather than starting at zero and ramping up to ten percent. Having the move function removes the initial start-up value of zero on pump two keeping all interlocks healthy. (Jackson, n.d.) (Pattimore, Move Function Block, 2015)

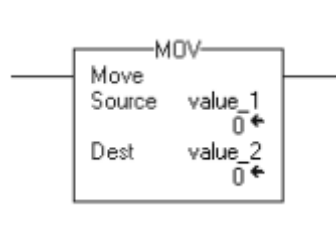


Figure 10: Once a true is sent to the block the source "value_1" currently 0 would be sent to "value_2", these could be static variables or constantly changing tags.

Jump to Subroutine (JSR) function: Because ladder logic is a sequential rule based language all rungs must be read before moving onto the next sequence. It is not possible to stop half way through a set of code and jump to the end without using the Jump function. When receiving and manipulating data from the field the JSR function is rarely used as the processing speed is within microseconds while the usable values out in the field can be seconds minutes even hours.

For this project, when opening a file with function blocks each function block is stored within a subroutine that is called upon to be executed, a default “main program” or subroutine is chosen that automatically executes on running the program. It is common to use this “main program” with jumps to jump to each subroutine. The reason why having one subroutine per drive was done is for easy navigation through code. If there are no jumps the organisation of files explained above is not possible and all function blocks for example digital signals must be created in one subroutine, this would be a problem for future trouble shooting when trying to find specific function blocks. (Jackson, n.d.) (Pattimore, Move Function Block, 2015)

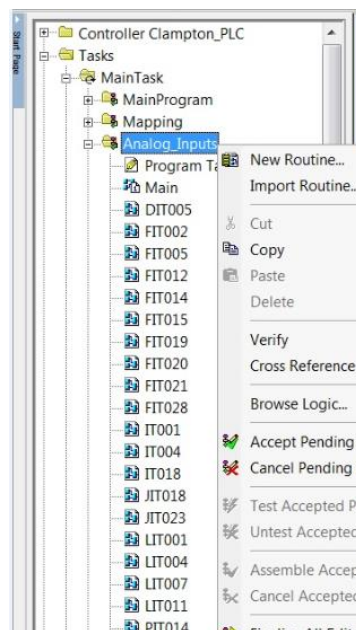


Figure 11: The “Main” subroutine would contain one JSR block for every routine within that task folder so when a specific subroutine is called upon i.e. FIT (Flow Indicator Transmitter) this routine can be used immediately.

Boolean: switches that when placed correctly can represent AND, OR, XOR, NOT, NAND, NOR and XNOR logic. These are the most common and basic logic found in the PLC code that. In combinations single or multiple inputs and can be used and manipulated to produce a desired output.

An example of using a basic Boolean function was the NOT function. When the drive tag is sent a one (rising edge) it would invert the one to a zero (falling edge). This was for intrinsic safety so that if there was a power surge or a circuit were to short and force a rising edge to the PLC this would instead turn off the drive. Keep in mind if the power were to turn off shutting down the PLC it also turns off all drives as on the Clampton site the PLC and drives are on the same grid. (Jackson, n.d.) (Pattimore, Move Function Block, 2015)

Timer function (TON/TOFF): the two most commonly used types of timers are timer delay on which after a true has been sent to the input of the timer block it will wait a pre-defined amount of milliseconds before sending a true to the output. Timer delay off allows a true to be sent from the input through to the output of the timer block but once the true on the input stops sending the signal it will latch the output as true until a pre-defined amount of milliseconds is complete where it will then send a false.

Timers are used mostly in sequential steps and alarms; an example would be warning workers and giving them time to clear away from a conveyor by having a siren sound for twenty seconds before turning on a conveyor belt. Another example would be to have a three second delay before showing a high high alarm on a level sensor, this is if the height of the liquid is just at the edge of the high high alarm limit it may go above and below this limit several times before settling, if there were no delay several high high alarm warnings would appear on the HMI. (Melore, 2011) (Pattimore, Move Function Block, 2015)

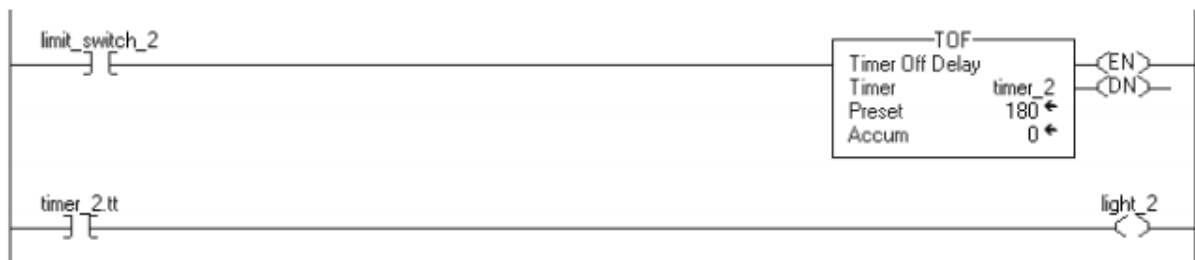


Figure 12: “Timer” is the name of the switch that will energise once TON finishes, “Preset” the time delay in milliseconds until the block sends a true, “Accum” shows a live value of how many milliseconds the timer has counted once activated. TOFF is similar except only activates on a falling edge.

1.4.5 Testing

Once the function blocks for instruments have been programmed into the PLC, bench testing was done by sending a digital or analogue signal from the HMI by the HMI programmer to the PLC to test how the code reacted. For example, when the HMI sends a ten percent speed signal to the conveyor tag in the PLC the tag would visibly change when monitoring the tag on Rslogix 5000 from zero to ten. This tag which is connected to the speed input of the conveyor function block will then go through the function block and be scaled from percentage to hertz for that particular motor and send a signal to the analogue output channel on the PLC rack that is designated to the conveyor according to the PLC I/O list.

Because the simulated network made in the office did not include the external instruments coming from the field there were no inputs going into the PLC or outputs going to any instruments. As a result, many of the interlocks and permissive written in the function blocks were unhealthy. Consequently, the testing becomes difficult as functions did not process data when interlocks were unhealthy. All interlocks were temporarily removed so that the function blocks were healthy for the bench testing.

1.4.6 PLC and HMI Roles

“Roles” does not mean the definition of what is a PLC or HMI but the purpose the PLC and HMI have in regards to this specific project and how they will be designed to interact with one another.

The HMI is a user interface but has the capabilities to manipulate data, such as inverting signals and creating basic Boolean logic and rules. However, the purpose for the HMI in this project is to display appropriate information to operators and allow the operators some limited access to control particular parts of the system while the PLC will implement boundaries, logic and manipulate data from the signals received from the instruments in the field. Having all data processing done on the PLC is also for safety if the HMI did control parts of the system and the PC with the HMI were to crash or suddenly shutdown then some of the logic controlling the system will also stop, sending false values back to the PLC potentially damaging the system but if all control were in the PLC and power were cut the control centre cabinet breakers will trip cutting power to the entire system not just the shutdown of the PLC.

Having all code that alters data on the PLC made problem solving easier. If an error occurred with the code i.e. the HMI displayed the wrong value or PLC did not respond as expected, it would be difficult to find and time consuming to switch between the PLC and HMI instead of checking just the PLC. Editing rungs on the PLC would become more difficult as the programmer must also compensate for how the new code may affect the code written in the HMI.

2.0 FAT (Factory Acceptance Test)/Black-box Test

The purpose of a FAT is to demonstrate the operation of the equipment requested by the client working within boundaries outlined within the operational procedure contract given to the vendor. When designing and constructing parts requested by a client, an operational procedure is created to be given to vendors, this operational procedure acts as guidelines outlining the goals that the system owner has in mind.

2.1.0 Warehouse Purpose

A description of a Factory Acceptance Test and Black-box Test is explained in this section. This summarises the goals to be achieved Section 2.2.0 Testing. It is important to differentiate between the F.A.T and Black-box Tests as the author of this report did a Black-Box test which may have similar goals to a Factory Acceptance Test but differ in what is being tested.

2.1.1 F.A.T (Factory Acceptance Test)

Before the vendor begins working on the design of the project a list of enquiries are made based on the operational procedure given by the system owner. These enquiries may include raising design flaws, Cost of parts or change of time to completion. Constant communication must be kept between vendor and system owners so both parties have an understanding of changes to the equipment and contract to speed up completion of the project.

When organising a F.A.T it is recommended to test as much hardware as practically possible. For this project the vendor was asked to design the Mill Control Centre based on an operational specification provided. Some of the hardware installed in the cabinet include the PLC rack with all

necessary inputs and outputs for digital and analogue signals shown on the I/O list, uninterruptible power supply, breakers, busses for various power supplies/earth, relays, thermal overloads, auxiliary relays and AC to DC converters.

When testing the product, the operational procedure is used as a final check list to show that the product meets the system owner's goals, creating a checklist of pass and fail criteria of the system that is tested. An affective F.A.T will result in finding faults in the system so that these faults can be fixed before being sent to the site. Having faults that occur during a F.A.T is expected and is more cost effective to fix after the test, in the planning stage than having them fix the faults on site.

Once the FAT has been completed any complications or failed criteria during the test are listed in a report to be given to the vendor. A discussion between the vendor, control engineer and system owner to confirm the faults to be fixed and items to be added to meet the system owner's goals.

(Hedberg, 2006) (Southern testing, 2013)

2.1.2 Black-box Test

Black-box testing is a sub-category of a Factory Acceptance Test that focuses on software testing. Black-box testing examines the functionality of software and how it reacts to its hardware counterparts without having to examine the internal workings and intricacies of the program.

An in-depth understanding of programming or how the code is designed is not needed for a Black-box Test. The tester should have a basic understanding of the whether an analogue or digital signal and what instrument a tag represents it is not required to know how the program works. For example, the tester should know the input/outputs used; variables and instrument the software presented by the tester does not need to know how the software chooses or creates these output/input.

In respect to this project the Black-box Test aimed to assess the response of the Mill Control Centre Cabinet (MCC). The criteria that were tested were inputs and outputs both analogue and digital of the PLC. These I/O were energised through either Rslogix 5000 tag library for outputs or injecting 4-20mA to inputs and monitor in the tag library, doing continuity tests on the wiring of relays and thermal overloads through the energizing of outputs and inputs. (Williams, 2006)

2.2.0 Testing

Separating between hardware and software this section aims to explain how the equipment was tested during the Black-box test so that it coincides with the I/O list created in Rslogix 5000. The section also presents some of the problems encountered during the test and how they were solved. Checking the I/O list that had been created in Rslogix 5000 with the hardwired I/O within the MCC would also simultaneously check the continuity of the wiring between the PLC I/O and the terminal blocks that connected the field instruments to the MCC. The Black-box test for this project can be broken down into three parts software testing, PLC setting and relay/wiring, once these three sections are confirmed to be working the Mill Control Centre was ready for site installation.



Figure 13: *The Mill Control Centre cabinet to be tested in the Black Box Test. Testing the digital and analogue I/O for consistency with the electrical drawings and the thermal overload signals which can be seen on the left half of the cabinet.*

2.2.1 Test Points

Any change or problem found during the test was documented for the system owner to look over but no fixes happened until the system owner accepted the proposed changes, this was required so that all participants in the integration of the system from warehouse to site had the same information during all stages of commissioning. This was to reduce future miscommunication between participants and less explanation of why and what was happening for each party before each stage making the commissioning progress faster. These tests were done to check the I/O list provided by the client with the hardwired I/O within the MCC and also the functionality of the hardware installed in the MCC as there had been changes to the system after receiving the documents.

Rslogix 5000: using the I/O list (described in Section 1.1 Client Readings, mentioned in Section 1.4.3 Organisation) a tag library was created representing all inputs and outputs of the PLC. When testing these I/O the technician had his own copy of electrical drawings showing the tag name coming from the physical cards on the PLC rack, cross referencing the tag library in Rslogix. The electrical drawings should show that each tag would go to the same channel. This was the first step to check if there was any inconsistency between the hardware and the software.

Programmable Logic Controller (PLC): the second test point was energising inputs and outputs, this was done by either turning on bits in the tag library to test in the Rslogix (digital output), varying a number most likely between 0-100(percentage) or 4-20 (representing mA) in Rslogix (analogue output), using a 4-20mA injector to energize analogue inputs or the test button on thermal overloads (digital inputs).

When testing digital and analogue outputs in Rslogix the outputs could be toggled from the tag list, on each I/O slot on the PLC rack there is an associated light that represents when that channel is energized. Toggling the tag on and off in Rslogix the technician is looking at the light on the slot to see if the correct I/O number is lighting up in accordance to the electrical drawings.

When testing the analogue inputs, the technician uses a device that injects 4-20mA into the channel. Once the channel is receiving a signal, the control engineer will check the channels labelled tag in the tag library to see if there is a response. Possible reasons for no response would be the wrong channel is used or the physical tag address is different to the tag address in the tag library. The technician will incrementally increase the current up to 20mA, if the tag in the library is scaled then the tag should display a proportional response e.g 4mA – 0, 8ma – 25, 12ma – 50, 16ma – 75, 20ma – 100. If the response is not proportional this is an indicator of faulty equipment or if raw current readings are displayed this means the tag has not been scaled. (Fluke Corporation, 2015)

Continuity Test: the purpose of the continuity test is to check the wiring and relays inside the MCC to see if they are consistent with the electrical drawings and that they are operational. Inspecting the digital and analogue inputs to the PLC channels is done by energizing the relays and testing thermal overloads (digital signal) or injecting current (analogue signal), once the relays are tripped or current is injected into the terminal block they are then sent to the specific channel on the PLC. Only if the wiring is correct and equipment is functioning will the PLC inputs receive the signal.

The same goes for outputs during the PLC test points, when testing output channels, the PLC will send a signal which will either trip a relay (digital output) or send a varying current to an address on a terminal block (analogue). Once these signals are confirmed in the PLC test it does not only confirm that the address is the same on the PLC as it is in the physical wiring of the MCC but also checks continuity of hardware and wiring. (Fluke Corporation, 2015)

2.2.2 Results Software

This section aims to show the procedure taken to check the tag list for this control system with errors, faults or proposed changes to the system that were documented and sent to the system owner.

Software Error 1: The client provided the control engineers an identical model Allen-Bradley PLC that was used for table testing in the office pre-installed in the MCC. Before checking the I/O ensure that the card revision in the software is the same as the card given in the PLC hardware. If major revisions are not the same between the PLC and Rslogix then Rslogix will not be able to communicate with the PLC. In this system it was found the PLC major revision was older than the revision set up in the software. This was easily changed in the software by downgrading the revision to the same in the PLC rather than upgrading the PLC to the newer revision, which would require the internet access or installation CD.

Software Error 2: Was found during the I/O testing that some of the labels on the wires did not match their associated tag in the PLC. The electrical drawings were then used to confirm if either the I/O list had been edited since being given to the control engineer or the cables had been labelled incorrectly. The electrical drawings showed that the cables should have the same tag names as the PLC tag library. This indicates that the cables had been labelled wrong. It was decided to change the tag name in the PLC to comply with the cable tags rather than keep the I/O list names. This was proposed so the cabinet makers did not have to individually change each cable tag resulting in prolonging the completion of the cabinet which would increase the cost of labour for the system owner. A list of all wrongly created labels was created and notes of the changes to the wiring diagram for submission to the system owner.

Software Error 3: Was found there was no feedback fault signal installed for the Mill. The electrical drawings showed that this was done on purpose. This was documented to propose to the system

owner to have a feedback signal installed, because of no feedback the nature of a malfunction would not be. Consequently it would make troubleshooting for the malfunction more difficult.

Software Error 4: It was noted that the Thermal Overloads (TOL) did not have a feedback fault signal to the PLC input card. The client deemed this feedback unnecessary since TOL would only be reset manually within the cabinet and not by the HMI. Having a feedback would make troubleshooting easier for example if a drive did not start because of a TOL become unhealthy, the HMI would only display the drive as not running.

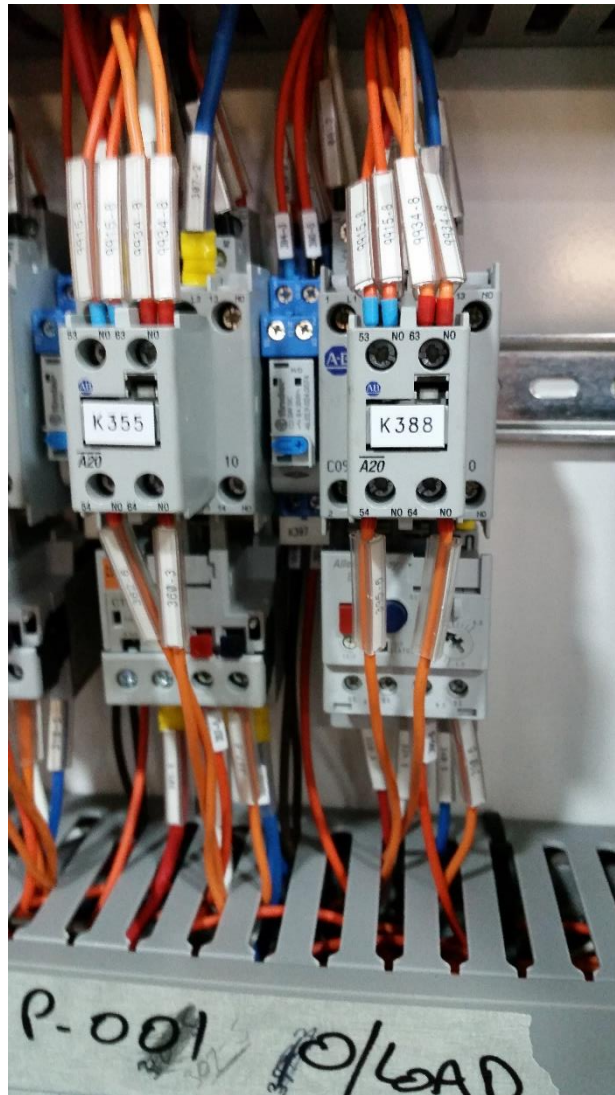


Figure 14: (Clampton, 2015) – E.g. K388 auxiliary relay mounted on the primary relay. Thermal Overload is attached below the primary relay.

Software Error 5: It was found one of the analogue input card slots on the PLC rack was set to differential mode rather than single ended mode. The differential mode compared two voltages on the analogue inputs, but this Mill Control Centre did not have a second voltage input for comparison. Meanwhile the single ended mode compares the incoming signal with the grounding of the input. Therefore the setup should be in single-ended mode. By checking the properties of the card slot this fault was found. Before this was misinterpreted as a faulty card. If this incorrect setting was not found, a new card slot would have been ordered for replacement.



Figure 15: (Clampton, 2015) – the Allen-Bradley PLC rack showing all card attachments DC and Analogue I/O cards, communication card (Ethernet) and logix5000 processor.

In summary when finding a fault, first check the electrical drawings match the wiring within the cabinet, once confirmed check communication between wiring and PLC. That is analogue input may be hardwired but the PLC expects a digital input. When these three checks are complete but the fault is still unknown, the cause may be either faulty hardware or PLC communication/manipulation or incorrect programming.

2.2.3 Results Hardware

Hardware problems found during the test in relation to the operation of the system and the PLC are documented and sent to the system owner where these solutions and proposed changes are looked over by the owner. The changing of or adding of hardware can cost the system owner time and money. Therefore documentation of any changes to the Mill Control Centre must be made. Software changes differ from hardware changes. Software changes do not necessarily mean it will cost more time and money because there are no physical hardware components or hours of labour for installation (depending on the type of software malfunction). However it still needs to be noted to keep the system owner up to date with how the operation of the software and cabinet. Contrarily hardware errors may result in the re-design of a section of the system. There is possibility of purchasing more expensive parts if the current ones cannot achieve the goals and time to completion set by the client.

Hardware Error 1: It was found that when a drives fault signal turned into a ready signal the fault signal was still being received. Checking the PLC code showed the fault tag for the input was receiving a live fault signal from somewhere else before going into the PLC in other words it was not the PLC code wrongly manipulating the tag. The electrical wiring schematic showed that the wiring was done correctly but the electrician found the drawing was incorrect. Then the drawing should be sent to a draftsman to re-draw and re-check the wiring before being sent back to the cabinet makers to re-wire.

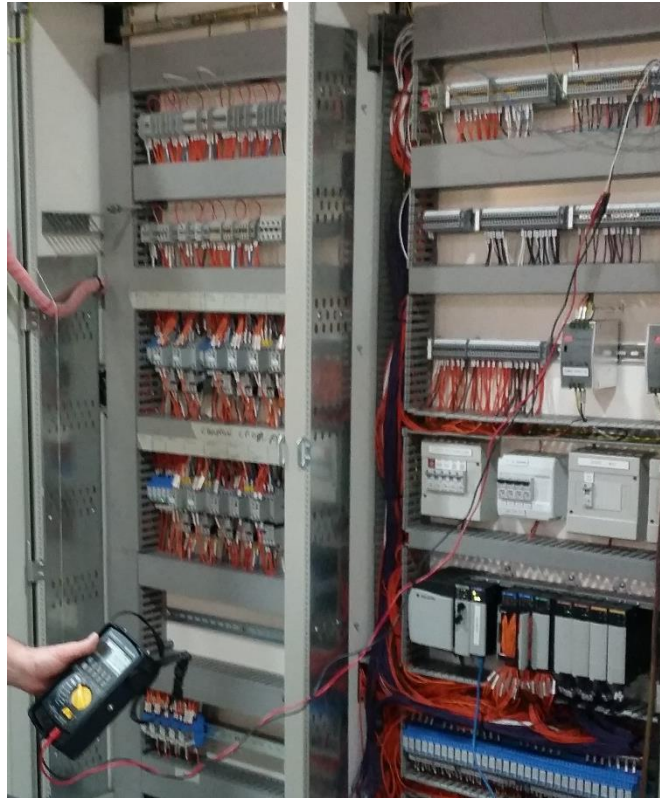


Figure 16: (Clampton, 2015) – *Technician sending a ready signal using the field terminal blocks. This plus checking the PLC library and monitoring the tag confirmed the drive showed a fault and ready signal simultaneously.*

Hardware Error 2: Some of the Thermal Overloads in the cabinet were set to auto while others were in manual mode. This was a concern as there were no feedbacks to the PLC, so it was reported to the system owner on for advice regarding system operation. When TOL is set to manual mode, once it is tripped, it breaks the circuit and can only be reset manually from inside the Mill Control Centre. An automatic TOL after tripping would be healthy again once the drive's temperature came back within the TOL boundaries. Then drive could start back up, this might be a problem if the PLC did not have a fault signal from the TOL to compensate. This was dangerous if a technician working on the drive not knowing that the drive automatically turned on again when the drive temperature goes down.

With the implemented design the settings for the TOL were inconsistent with some being automatic and other manual. This was dangerous because of lack of fault signal feedback if a drive were to trip

the operator would not know why and if they did not know if the TOL has to be manually reset or the drive could start up at any moment.

Hardware Error 4: Auxiliary contact blocks mounted over the drive relays were set to normally close instead of normally open giving the PLC input a constant positive. According to the control philosophy these contact blocks were supposed to send an open circuit signal. This error also occurred in one of the Thermal Overloads. This was checked with a multimeter using the continuity option.

Hardware Error 5: Some of the drives had no fault detection feedback to the PLC. One of the options was by daisy chaining all drives to one input, in doing so if any drives became unhealthy a single fault signal representing fault detection of all drives shown, rather than no fault detection. However the drives and auxiliary contactors were from different brands, making daisy chaining as different archetypes could cause conflict. To change all brands to a unique brand could increase the cost for parts and labour and delay the completion time of the Mill Control Centre. As a result daisy chaining not a feasible option. The problem however was documented to be discussed with the system owner.

Hardware Error 6: A problem that was encountered multiple times was loose contactors or wrong wiring. For examples being the technician could not send signals during the Black-Box Test because of loose connections. The vibrating hopper indicator light flickered on and off as cabinet door opened and closed as if shorting. The electrical drawings showed that if a drive were in a ready state a ready light on the cabinet should light up but if it faulted the ready light would turn off. This was not the case as even when faulted the ready light still remained on. It was important to fix these connections not just for consistency with the signal but also to reduce the possibility of sparking.



Figure 17: (Clampton, 2015) – *Technician tracing cable for loose connections, electrical drawing is consistent or tags labelled on the wire are correct.*

Hardware Error 7: New outputs were found according to the electrical drawing that was not updated in the I/O list. Following electrical drawing show the sirens. These sirens were unique in the way that they need two 240V to run unlike all other digital signals which use 24V relays. The wiring showed that the 24V relays fed into a 240V relay that then powered the siren. The I/O list was updated and this finding was included in the report to the system owner for advice regarding the unknown I/O and its desired operation.

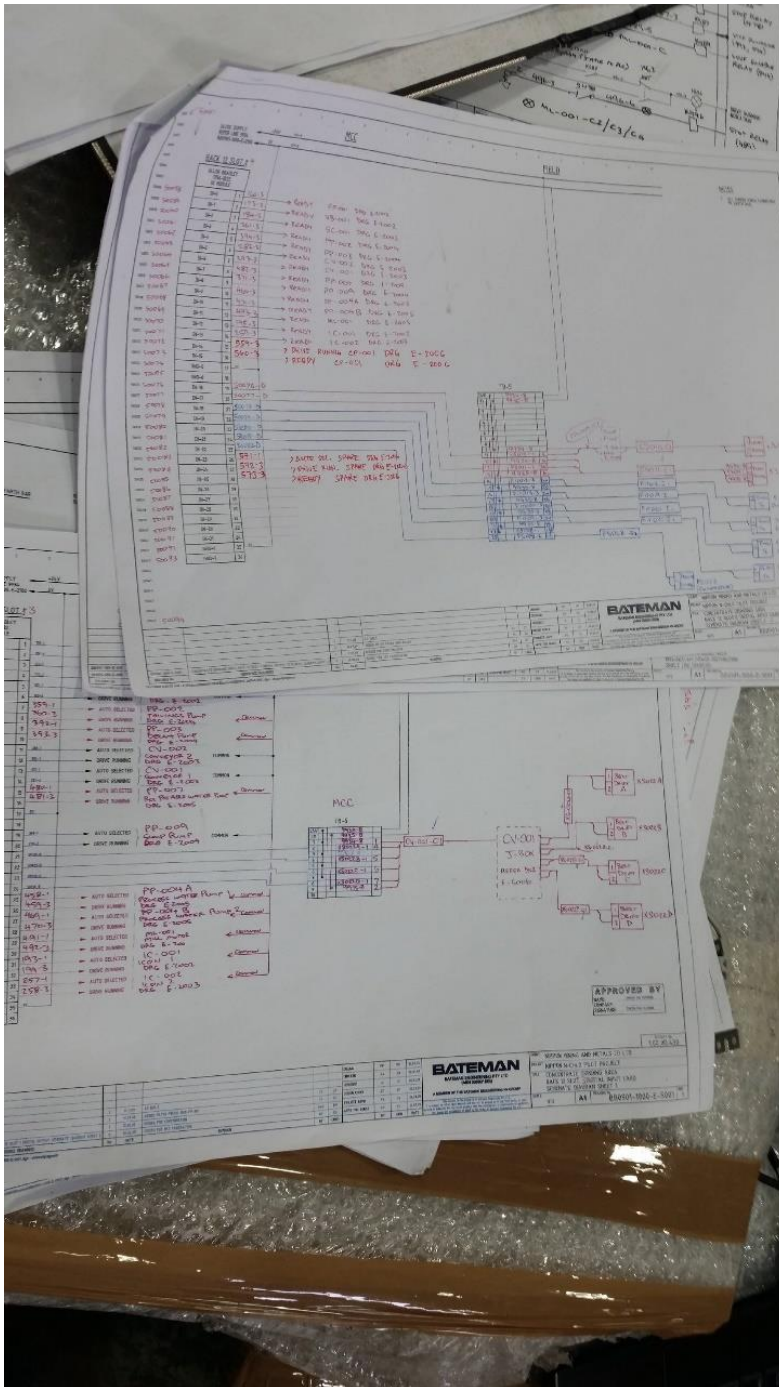


Figure 18: (Clampton, 2015) – Electrical drawings with edits to the design which include errors found or new design objectives to be added to the MCC.

2.3 Summary Black-Box Test

A summary of the procedure for checking the software within the PLC with the hardware of the Mill Control Centre will be explained. Using the tag library in Rslogix the tags were arranged in numerical order.

Step1: find the input or output to be tested. Meanwhile the technician finds the channel, relay, terminal or thermal overload that is associated with the tag to be tested according to the wiring diagram.

Step2: Highlight the tag in the monitor tab in Rslogix allow the tag to be modified if it is an output or to observe if it is an input. An Excel spread sheet is brought to the Black Box Test containing all I/O with their addresses, tag names and descriptions that make up the tag library in Rslogix. If any of the I/O given in the list has different or conflicting detail (address, names, description) to the detail is found in the hardware or if any new tags is found, a record of these changes is created and the spread sheet is updated.

Step3: Create a checklist including any questions that need to be asked to the system owner or errors occur such as not having feedback fault signals or manual and auto settings for the thermal overloads. This checklist contains errors, changes and problems with possible solutions to the problems. If possible an explanation of what would happen if this solution is chosen such as time to completion or cost.

The steps taken during the Black Box Test if an unanticipated response from the I/O or software takes place will be explained. An unanticipated response can be no response, wrong signal or wrong I/O reacting to the signal, these responses can represent multiple problems in different parts of the Mill Control Centre where these parts are the software (Rslogix, Rslinx), hardware(PLC, TOL, relay)

and wiring. The software problem can be the properties of the I/O card on the PLC rack not correctly been configured, or faulty wiring, or faulty equipment or electrical drawings.

To troubleshoot an I/O problem, first check the wiring diagram to confirm the I/O being tested in the program and in the Mill Control Centre being the same. This eliminates the possibility of wrongly labelled cables. Once confirmed, if the error persists, check the Excel spread sheet for correct I/O name and channel, cross referencing the tag library on the PLC. With these checks done it rules out the possibility of the error being from the software or discrepancies between documents and hardware/software.

If all the above steps are completed and the problem is still not resolved, trace the cables using the wiring diagram to check where the connection of the cables to rule out incorrect wiring in the cabinet. Checking many metres of cable takes time. If the wiring error is serious it may require to re-cable large parts of the cabinet which is a undesirable solution costing more money and delaying the completion of the product and Black Box Test, while a software error can be quickly fixed. If wiring is correct then check code looking through driver properties, tag properties and cross-reference that tag to check if is used somewhere else in the program as it might conflict with other operations. Once all these steps are taken, the faulty equipment be considered as the reason for the problem.

3.0 Clampton Commissioning

The final stage of this project, once the Black Box Test has been completed, is commissioning onsite. This section will present work that was accomplished on site and some of the problems came across. Many of these problems included technicians or electricians as all problems related to the control engineer was interrelated to communication between devices and installation and operation of equipment.

Commissioning on site is very similar to a Factory Acceptance Test, where each installed component is checked for operational use. The difference is the equipment being setup on site as described in the control philosophy, ready to be used in correlation with each other.



Figure 19: (Clampton, Site - Commission, 2015) – *System as found first day, still in process of installation. Throughout commissioning the control team will be updated on changes or completion of systems so that they can be tested for compatibility with the control system.*

3.1.0 Pre-Commissioning

Pre-commissioning is the time to individually check all I/O, hardware and software being correctly installed on site. This stage was done before creating sequences its only purpose is to ready all parts of the system so they can begin to work together. First the hardware and software are tested and checked to be able to work individually, once they are functional can they be integrated to work with each other. To achieve this the steps taken are to systematically check all equipment and systems within a project. Then test and design their functions based on the control philosophy provided by the system owner.

3.1.1 Pre-Commissioning

Once the Black Box test has been completed and any necessary changes have been made the next step is installation of the control program on site. The aim of commissioning on site is to test the PLC code, making sure all boundaries, sequences and responses from the field are as expected according to the control philosophy. Since the objective of this thesis is to show the purpose of the PLC from table-top simulation to Black Box Test the implementation and errors related to the human machine interface will not be looked into but only mentioned when related to the PLC.

Before the implementation of sequences or the creation/modification of the HMI can take place, a number of objectives must be checked off for the site to be ready for commissioning.

Testing of individual components that have been installed through the human machine interface:

When the control engineer arrives on a site to commission most of the system has been installed and the finishing stages are commencing. There may be only a few components in the system left to be installed before the building of the system is complete.

At this stage, as in a FAT, the control engineer will test each analogue and digital input /output, checking for the response of the component, the responses of the sensors controlling the component, how the PLC code reacting to the sensors data and how the HMI displaying important information relevant to the system operators.

An example is the drive representing the tailings pump. The purpose of this pump is to push the tailings or excess slurry that has been processed through the system from the tailings vat to the tailings dam so that it can be recycled back through the system. Before commissioning/sequencing the pump will be implemented in the start-up/shutdown sequence and control loops. The control engineer must check the response of the pump in the system. Firstly turn the pump on and off from the HMI proves the connections from the Mill Control Centre out to the field and correct code uploading to the PLC. Secondly control the pump by ramping the pump up and down through the PLC and then through the HMI to check if the code calibrated, in this case converting a percentage zero to one hundred to a four to twenty milliamp signal.

Finally the high, high high, low and low low limits for the tailings vat is tested. This is done by stepping through each alarm, energizing each fault into the Rslogix 5000 pump function block. If programmed correctly the HMI will display the level limit to be reached and the output of the function block will also show which limit has been exceeded.

Now that it is known that the code is correct and the PLC to pump wiring is correct the final test is to simulate the ultrasonic sensor that will indicate the level of the tailings vat. The electrician on site will inject a four to twenty milliamp signal into the terminal block which will simulate the ultrasonic level sensor. If the sensor terminal block is wired correctly the function block and consequently the HMI for the upper and lower boundaries will change and affect the status of the pump.



Figure 20: (Clampton, Site - Commission, 2015) – *Tailings vat showing sump water output and ultrasonic level sensor.*

3.1.2 Error Checking

The steps taken during pre-commissioning are to test each of the components in the system. The test confirms that each part is individually functional, so when commissioning begins they can then be integrated into the system. This is an important step for when an error occurs during commissioning where multiple parts are interacting and affecting each other's operation, because each individual component has been isolated and tested the error must be in how they are interacting.

When an error does take place a step by step process is done to find the source of the problem much like during the Black Box Test. This is not just to find a solution problem but also systematically checks the scale of the problem starting from a wrongly labelled tag to ordering new parts. The steps that were taken on the Clampton site to solve all problems will be explained below.

Potential errors can be grouped into:

- Software/code
- Wiring/connections
- Field instruments

The priority for finding which category the error is located in is software, wiring and field instrument, the order of priority is based on how demanding the solution is to fix the problem. If it is a software or code problem the labour required to change the code and calibrate the settings may cost time but no parts would need to be replaced or physical components fixed. If it is a wiring or communications problem it may be found between PC to PLC or PLC to instrument. Most problems in this category are a continuity error where the cable does not match the wiring diagram or the electrician's wiring diagram has not been updated for changes made to the cabinet since the Black Box Test or Factory Acceptance Test. The last possibility where the error can be located is the field instrument being tested. The common errors that can occur are wrong installation, the faulty instrument or wrong configuration of a drive or a sensor one example is that for a ultrasonic level sensor the upper/lower boundary or margin of error for turbulent liquid may not match the size/shape of container or possibly the type of liquid to be measured giving false readings back to the PLC.

Finding the section where the error occurs can narrow down what the error is and will be explained this below.

Software/code error check:

When checking an error in an analogue or digital output, firstly check the output of the function block that is to be sent to the field from the PLC. This will either show an expected value proving that the code is correct and the error is somewhere further down the line or a unexpected value meaning the logic is incorrectly manipulated somewhere in the code or conflicting software properties before it is sent to the instrument out in the field.

Whether the output from the code is correct or incorrect it is important to check the properties of the software as the code may be correct but the communications between the PC and PLC may not be configured correctly. An examples of incorrect properties would be converting percentage zero to one hundred to a four to twenty milliamps signal. Without proper conversions the component out in the field will not be able to be controlled by the PLC.

Regarding the input signals a technician with a device that injects an analogue signal or simulates a digital signal will control the input tag by injecting a controlled signal into the terminal block to which the tag is assigned to. Once again monitoring the input tag in the tag library will show what happens in the field. If the signal from the field is identical to the signal being injected into the system then this rules out incorrect wiring/communication and field instruments, leaving software properties and the logic manipulating the data to be the cause of the error.

When all settings have been checked for consistency such as scaling, communication properties and software revision, the final setup is checking each rung inside the function block and watching which bits become healthy and unhealthy until the error is found. In the commissioning stage the control engineer only looks at isolated components. The task of following each rung and tracing the logic the signal is being processed through. This is why pre-commissioning exists as the instrument is isolated from other components, it is much easier to follow the code and tags in ladder logic as each instrument/function block would affect and control other instruments/function blocks in the system if the instrument being tested were already integrated with other instruments.

Wiring/connections error check:

Once it is confirmed that the code/software is correct the next step is checking the Mill Control Centre. Firstly if this is an output signal from the PLC to the field instrument the electrician will find the terminal port coming from the field into the cabinet and isolate the instrument. Using a multimeter the technician will check if the signal created by the control engineer is being sent to the field from the cabinet. If the signal from the output of the terminal is not consistent with the signal sent then the problem can be identified to be within the cabinet as the data being correctly processed and sent from the PLC but corrupted before leaving the cabinet to the field instrument.

Similarly this is done for the inputs but in reverse. The technician will inject an analogue or digital signal from the terminal into the cabinet. The PLC input card will read this value and send the reading to the control room, where it is monitored by the control engineer. If the signal created from the technician does not match the signal being received by the engineer it can be determined the problem being isolated to within the cabinet.

Now that it is confirmed that the error is within the cabinet there are a number of categories within this section that the problem can be under:

- Wrong wiring/tags
- Outdated/changed/wrong wiring diagram
- Faulty parts

These categories are done in the order of priority where each category takes more time and effort than the previous. When checking wrong wiring and tags the electrical drawings are needed to trace the wires. The wires should be correctly labelled, they should go to the correct components within the cabinet and the wiring diagram.

If no faults are found, a continuity test will be required to check that all components and connections within the circuit are healthy. This is done using a function on the multimeter that sends a small current from one node point to the other. If the circuit is completed, the multimeter will indicate a closed circuit. The continuity test will be done by isolating each component in the circuit and testing the hardware and also checking each wire segment. Wires should be labelled correctly, all components such as terminal blocks, auxiliary drives, thermal overloads, timers and relays should be functional. If any parts do not pass the continuity test the electrician may have to replace them, if wires do not pass the test and it is not a faulty connection then the wire will be replaced.

Finally the electrician will proof read the electrical drawings again to make sure the signal is correctly sent to the right components. This is similar to the taste in which the control engineer proof reads the ladder logic within the function blocks to check if inputs and outputs being correctly manipulated by the control logic. Errors found in the electrical drawings must be brought to the system owner and control engineer to discuss changes. These changes could affect the operation of the PLC and postpone other stages in commissioning as the new design should be documented, edited into the electrical drawings and installed. Once all these categories are ticked off the possible errors must be the field instrument.

Field Instrument Error check:

There are thousands of different types of devices that make up a system. Four main groups are panels (containing emergency stops, potentiometers, touch screens, breakers switches etc.), sensors, drives and alarms. It is not the control engineer's job to install and configure the settings of each of these devices. That job is for the technician, electricians, mechanics etc. This section will explain the involvement of the control engineer and PLC in the general problem solving process of a field instrument. Every instrument is unique to problem solve from the point of view of a technician as each device has its own manual and operational procedure. The control engineer however uses the same solving methodology to troubleshoot field instrument errors. All types of instrument and

how to find their errors will not be explained in this section but the general steps taken by the control engineer and PLC to help the technician find the problem will be presented.

The sensors such as level switch, ultrasonic level, flow rate, and pressure sensors are installed by the technician. Almost all of these sensors have front panels or dual in-line package (DIP) switches to be able to configure for the specific use of the sensor for example if a level switch sends a normally closed or open signal back to the Mill Control Centre, flow rate and pressure sensors must have boundary settings input and frequency of data sending back to the Mill Control Centre so that the PLC will not be overloaded with unnecessary data.

If these settings are not correctly configured or the physical sensor is wrongly installed are sent back false readings to the field. False readings will wrongly represent the state of the system is doing consequently the PLC will try to control this incorrect system.

Finding the error in the field instrument is much easier than in the software/code or wiring/connections in regards to the control engineer and PLC and can be cut down into three categories for errors:

- Instrument has not been correctly configured
- Instrument has not been correctly installed
- Instrument is faulty

The technician should be able to check If the instrument has not been correctly configured this by using a multimeter and testing the signal cable. Using controlled responses with definite values the technician will monitor the readings from the instrument on a multimeter while the control engineer will monitor the address in the tag library of the instrument for correct values. An example would be an ultrasonic level sensor where there are settings for the boundaries of measurement. If the level is below or above certain distances, the sensor will read the value as empty or full for the container so that the PLC does not have to have these boundaries coded into the program. The control engineer

will monitor the response of the tag as the technician toggles between the low and high boundaries. If the response is below the limit the sensor should send four milliamps and if it is above the limit the sensor should send twenty milliamps. If the sensor does not send these it means the instrument has not been configured correctly, so these settings must be changed to produce the correct boundaries.

Once the boundary values are being read correctly for the analogue sensor, the response of the sensor within the upper and lower limit must be tested. The control engineer will communicate with the technician to report the PLC reading, meanwhile while the technician is out in the field checking if these values are a good representation of the sensor reading, so then he will confirm the reading with the control engineer. If there are discrepancies between the PLC readings and the technician's readings the advanced settings on the sensors front panel might not be properly configured for this instrument under this specific circumstance. These settings are more commonly found on analogue sensors rather than digital as a varying signal needs more boundaries to give accurate information than a basic on/off digital sensor.

Some of the settings are time delay when limits are reached, so valves are not continuously closing and opening. Margin of error ignorance which is important for sensors that have problems with noise such as level sensors, flow sensors and pressure sensors usually have frequency of data to be sent back to the control room. If the frequency is not high enough there will be a delay in the response of the PLC to control the system. If the frequency is too high, it may send values faster than the PLC can read creating a buffer overflow that is if the PLC can only read one packet per second and the instrument sends three packets per second in five seconds, the PLC buffer which stores values for the next iteration of the loop in the PLC code will now have ten packets and will infinitely increase until the PLC memory is completely used causing an error.

In summary the objective for pre-commissioning is to confirm that all components in the system are checked for functionality before they are integrated into an interactive system. In doing so an error

occurs during the testing of a sequence or interaction between instruments the possibility of faulty equipment, installation and wiring can be ruled out.

3.1.3 Clampton Mining Error Check

Using the problem solving steps from 3.1.2 Error Checking this section will list some problems found on site that from software, wiring to field errors:

- All function blocks on drives have interlock properties set to zero or unhealthy either these interlocks have not been created yet or because of isolated inputs causing the drives not be able to start by the PLC. It is necessary to go in block properties to switch all unused interlocks to 1 for default value. (Software error).
- Siren was not turned on when wiring was checked. It was found that the earth wire of the siren was put into the twenty volt ground rather than the two hundred and forty volt ground. (Wiring)
- Multiple address errors were found with different revisions of the Control Philosophy, either the wrong drive giving the wrong input back to the PLC or the wrong address in the PLC tag library for the drive being tested. (Documentation error)
- Field sensor was set up expecting the PLC to act like a source instead of a sink, meaning the PLC was expected to supply varying current to the sensor instead of act as a ground to which the sensor would send current. (Field instrument error)
- Flow indicator transmitter was wired into the correct channel and slot on the PLC rack and correctly implemented in the code but the function was called FIT012 when the previous flow meter was labelled FIT010 meaning a “spare” analogue digital input/output was not included in the wiring

diagram. It was found that the PLC had one unused spare I/O because of the miss-labelled document. (Documentation both wiring diagram and I/O list)



Figure 21: (Clampton, Site - Commission, 2015) – *After reading the control philosophy and wiring diagram the spare I/O was found in the MCC.*

- Normally open level switch for sump gave constant signal. The signal was checked with multimeter. It gave back fifteen milliamps instead of twenty four milliamps or four milliamps. Because it was a digital on/off signal any value other than twenty four and four should not be possible, indicating either the connection is faulty or the level switch is faulty. Doing a continuity test showed the wiring to be correct and connections were tightened leading to the conclusion that the sensor was faulty and was replaced. (Field)



Figure 22: (Clampton, Site - Commission, 2015) – *Sump pump (left), digital level switch (right). Tailings pit fills when the system overflows, the level switch activates the sump pump to recycle the water back into the tailings damn to be used again.*

These faults that were found using error checking techniques were then brought to a meeting with the control engineer, technicians and system owner weighing the pros and cons of cost, time, purpose and efficiency of possible solutions.

3.2.0 Commissioning

Now that all components in the system have been checked for functionality such as communication between the instrument and the PLC, the communication between the PLC and control room and appropriate responses from isolated instrument when being controlled the next step is commissioning.

Commissioning and pre-commissioning are very much like building a car. In pre-commissioning each component of the car is checked for functionality ticking off check boxes so that the components are

up to a standard and are ready to be assembled. When commissioning once these components have been tested, each is integrated into the system based on the design of the car by the system owner in the same manner as the control philosophy is used as the base design for mine site.

The three major steps taken in commissioning the Clampton mine site for the PLC are boundary implementation on drives and valves, field safety mechanism response/testing (how does hitting an emergency stop or conveyor drift switch affect the system both before it is tripped and after) and sequencing.

3.2.1 Boundaries

As explained in Section 1.1.4 the Control Philosophy acts as a set of operational guidelines for each instrument in the field, explaining how it should function and what interlocks are needed for the instrument to be healthy. The boundaries defined by the control philosophy on how the instrument should operate is based not only on the end product of how much gold can be produced in the shortest time but also other factors. Some of these factors are tonnes of material that can be processed per hour, maintaining controlled water pressure, maintaining container level, viscosity of the slurry and safe operation of equipment. These points act as boundaries that restrict the amount of gold that can be collected and these boundaries are used to design a system that is optimised to produce as much product while still maintaining a controlled system.

The control engineer first applies boundaries into the PLC for each drive and valve, before creating a system where each drive/valve works together to produce a desired outcome. The individual parts have boundaries placed so that they do not get damaged, do not hinder the effectiveness of its purpose in the system, do not damage the parts around it and do not endanger personnel working near the part.

Two examples of boundaries created by the control engineer and implemented into the PLC for the operation of isolated parts of the system shown about below:

Example 1) Boundaries vary depending on the purpose of the part. For example if it is a motor that drives the mill, an upper and lower limit for the motor frequency will be placed both to protect the motor driving the mill and safety for people working around the mill. These lower and upper limits under normal circumstances should never be reached but if a malfunction occurred these boundaries act as fail safes. As mentioned, the engineer does not want to hinder the effectiveness of the mills purpose of grinding dirt to a fine enough slurry to be processed and also to produce the grind as quickly as possible. If the mill spins to slow the amount of slurry per hour coming out of the mill will be far less than if the drive has had a higher frequency which is undesired. However a higher frequency means that the dirt does not get a thorough grind in the mill with the slurry turning from the desired size of smaller than sand to as large as pebbles which is also unwanted.

To help narrow down the operating frequency, the engineer knows the type of Mill is a Semi-Autogenous Grinding Mill rather than a Ball Mill. That means instead of the bearings in the mill to be dropped onto the dirt by the momentum of the mill spinning at high frequency the bearings are supposed to be grinding together producing a much finer slurry on the output as the mill rotates slowly. (SATYENDRA, 2015)



Figure 23: (Clampton, Site - Commission, 2015) – *Semi-Autogenous Grinding Mill.*

Example 2) the mill water sprayer purpose is to spray water into the intake of the mill where the dirt from the conveyor belt goes in. The water and dirt are mixed together to create a slurry that once grinded can be pumped throughout the system so that the gold in the dirt can be processed. The valve that controls the amount of water per second may not be a hazard for personnel as much as the mill but it has its own boundaries to maintain.

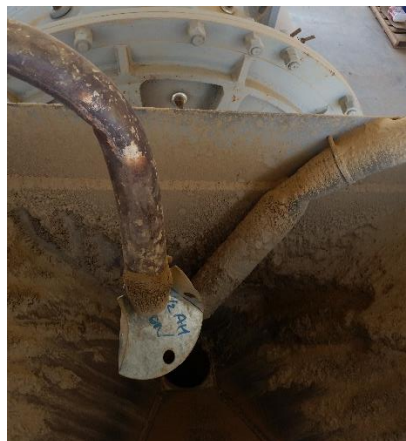


Figure 24: (Clampton, Site - Commission, 2015) – *Intake of Mill with water sprayer nozzle.*

The valve controls the flow rate of the water and also directly affects the pressure in the pipes. It is necessary to keep a constant pressure through the pipe. If the valve is nearly fully open, too much water flows through. Consequently there is not enough pressure to maintain a constant flowrate. Meanwhile if there is not enough water mixing with the dirt, the viscosity of the slurry becomes too thick, which could slow the flow through the pipes, forcing the pumps to work harder and possibly over heat or pressure rising to dangerous levels. The control engineer uses the readings from the flow sensor and pressure sensor installed around this valve as boundaries to control the valve while aiming for a preferred ratio of water to dirt from the output of the mill as defined by the system owner.



Figure 25: (Clampton, Site - Commission, 2015) – Mill Feed Flow meter + Auto controlled valve.

Example 3) Tailings pump is run at a constant speed to prevent blocking of the tailings line and will auto start pumping only when the level indicator transmitter is above the pre-set Hi limit that was

programmed into the sensor by technicians. This pump will stop if the level indicator transmitter reads a low low for more than thirty seconds, emergency stop is pressed or a drive fault such as thermal overload is read back to the PLC. This is an example of how two devices interact to produce a wanted outcome, a sensor's data is used to control the level of a tank by placing boundaries that are used by the pump. Without this level sensor the tank would be constantly dry eventually damaging the pump or overflowing the tank.

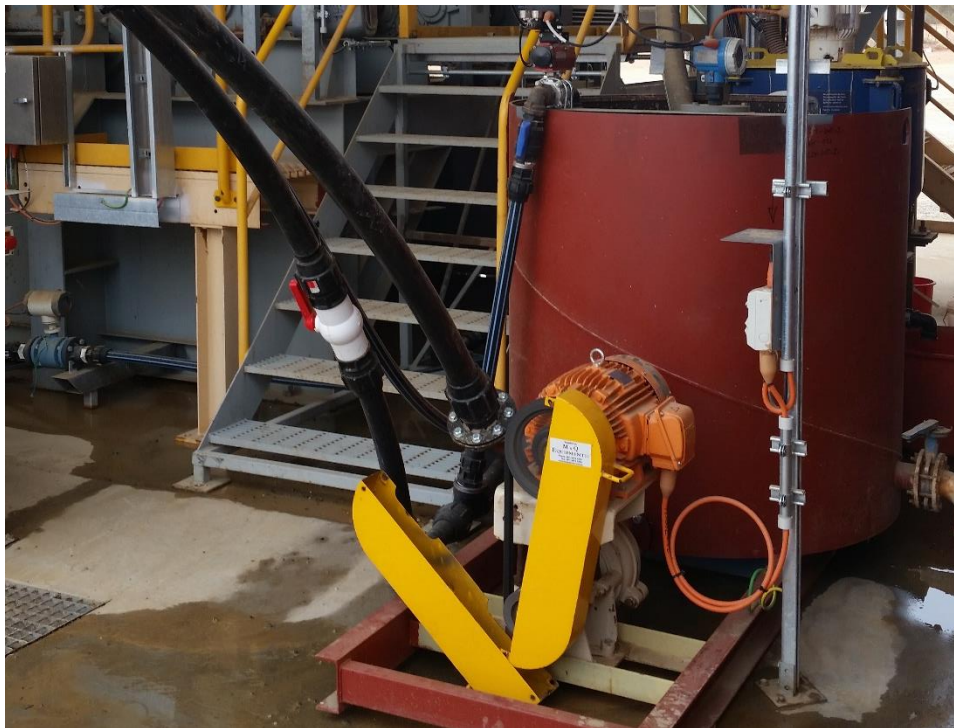


Figure 26: (Clampton, Site - Commission, 2015) – Tailings pump with ultrasonic level sensor.

In summary as explained in the second and third example, there are two layers of boundaries that are placed on every drive and valve. The first are the boundaries created by the physical limitations of the drive/valves and its actions such as max and min rpm, number of times valve opens and closes per minute etc. The second layer is where the sensors surrounding the component add new boundaries to the drive giving new inputs and information that further improve the productivity, consistency and safety of the instrument. Once all drives and valves have been programmed with

appropriate boundaries the control engineer can proceed to the next step of testing the response of manual safety switches out in the field.

3.2.2 Field Safety Mechanism Response and Commissioning.

The field safety devices installed at the Clampton site are different types of emergency stops such as conveyor belt tilt switch, emergency stop button and pulley switches. Machinery in the field malfunctions could result in damage to machinery, harm the work in progress or becomes a hazard to personnel on site the emergency stop can be pressed to manually cut the power to that drive, all drives have an emergency stop and all emergency stops are installed near the drives for easy access. Once the emergency stop has been pressed and the power to the drive is disconnected the only way to restore power is to manually reset the emergency stop out in the field.



Figure 27: (Clampton, Site - Commission, 2015) – *Emergency stop for sump pump.*

The purpose for testing the emergency stops after implementing the boundaries for each drive have been programmed in but before sequencing is that the technicians and control engineer still work on isolated instruments. The aim is to test the response of isolated instruments when emergency

buttons are pressed. From a hardware perspective if the sequencing had been completed and the instrument were to malfunction when the E-stop is pressed it could damage other instruments in the sequence. From a software point of view if tested after sequencing is completed and an emergency stop wiring error was found or code was not implemented correctly to compensate for emergency stops the software solutions to fix the problem may affect other steps in the sequence forcing the Control Engineer to re-write large parts of the code.

The procedure for a Clampton emergency stop is that once the button is pressed, the drive's power is cut off which forces the drive to shut down then an unhealthy signal is sent to the PLC indicating the E-stop has been pressed. After resetting the emergency stop, the drive will not start until the operator in the control room starts the drive again through the human machine interface. Because coding involves in the procedure, this code needs to be tested while at the same time the wiring of the emergency stop switches would be simultaneously tested.

Testing is done by turning on each drive, monitoring the responses of the input and output tags of the drive in the PLC library, and having the technician press the emergency stop out in the field. If done correctly the emergency stop should cut the power to the drive immediately sending a signal back to the PLC that the drive has stopped, then even after the emergency stop has been reset the drive will not start until the control room sends a run signal back to the drive.

Below is an example of an emergency button error that was found on site:

Found an error in the wiring of the emergency stop for the conveyor belt. The wiring for the conveyor belts are designed to include physical timers so once the E-stop is reset and PLC sent the on signal to the conveyor this signal would go through a physical latching on delay relay. The goal for this delay was that during the count down a siren would sound warning all personnel near the conveyor that it would start moving shortly.



Figure 28: (Clampton, Site - Commission, 2015) – *Emergency stops for conveyor one and two.*

It was found that when the emergency stop was pressed the conveyor would stop but once the emergency stop was reset the conveyor without the warning siren or delay would start immediately. Checking the wiring diagram the emergency stops wiring was installed after the timers which were time delay on and would latch holding the ready signal. Because the emergency stop was wired after the latched timers there was no way to de-energize the timers resulting in the conveyor to immediately start without a time delay as according to the time delay relay it had already counted down. The solution to this problem was to re-wire the cabinet moving the emergency stop wiring to before the timer on delay latching relay. With this new configuration the emergency stop would not only cut power to the conveyor belt immediately stopping it but would also reset the latched timer ready to be started by the operator.



Figure 29: (Clampton, Site - Commission, 2015) – *Tilt switch and lanyard emergency stops for conveyor belt.*

Because the control engineer and technician checked the emergency stops for each drive systematically they were able to find this error. As turning on the conveyor is one of the last steps in the start-up sequence, this E-stop would have taken the longest to check. If the sequencing had already been implemented and this error is a direct hazard to personnel working on site showing the importance of having a procedure to check for these errors.

3.2.3 Sequencing

After the components have been tested, wiring and emergency stops have been checked, and the boundaries have been implemented, the next step is to integrate the parts using the control philosophy provided by the system owner. Sequencing is integrating each component in the field to work together to create a desired outcome. In this project using the components tested in pre-

commissioning, the final product would be a functional alluvial gold mine site. Because of the company policy, this report will not show the specific design of sequences which may show how the site runs. However the report will use similar examples to demonstrate the steps that the control engineer should take using the PLC and control philosophy to make a site that adheres to the system owner's operational standards.

When creating a sequence using the control philosophy the first information to be checked is the interlocks and permissive, which act as a second layer of the boundaries for each drive. The difference between a permissive and interlock is that a permissive is optional being allowed to occur but is not compulsory, while an interlock is required to happen for the drive to be ready to run. These interlocks and permissive determine the different states of the drive and how it will be displayed on the HMI and how the PLC will read/respond to these states. An example of an interlock is that conveyor one drive will not start until conveyor two drive starts and after a ten second delay, while a permissive would be a siren may sounds for the ten second delay.

The boundaries explained before act as operational guidelines for each drive. These boundaries are created based on the purpose and limitation of the drive, for which it is being created. This is different to the interlocks and permissive boundaries in sequencing as these boundaries are created by the operational guidelines of the system and how the system as a whole will function ignoring the individual boundaries of each drive and instead seeing how each drive will affect the operation of the drives around it in the sequence.

For examples of operation boundaries for sequences are below:

Example 1) The mill will continue to run for thirty seconds after the shutdown sequence has begun. This is to grind out any leftover media in the mill, If there was still material in the mill the next time the start-up sequence is initiated the motor will need a much larger amount of torque because of the left over slurry. This may seem like a normal operational boundary but it takes into account the

devices around it checking the discharge vibrating screen and discharge pump are running first for this step to be active.

Example 2) The drive that runs conveyor belt two feeding into the mill in accordance to the control philosophy can only start once the mill turns (interlock). If the mill does not spin but the conveyor turns the will can build up within the motionless mill and increasing the torque required for start-up. Also conveyor belt one can only start after conveyor belt two turns, otherwise the dirt cannot move.

When creating sequences, the boundaries to be created only monitor the following step in the sequence. For example conveyor belt one can only be turned on when conveyor belt two turns. Meanwhile the code for conveyor two states that it can only run when the mill turns. Each step in the sequence only checks the following step but does not try to compensate for all steps before and after the following step. That means that conveyor one only monitors conveyor two to be able to become healthy. It does not need to compensate by checking if the mill turns as this will be monitored in conveyor two's sequence code. If each step needed to be coded to compensate for every other step, the amount of extra code needed per step would be extremely large.

There are a limited amount of tags given per Rslogix license and a limited amount of memory held in the Allen-Bradley PLC to hold the code that controls the site. Having efficient code that is not just for processing speed and easy to follow for technicians after commissioning is completed but also because of these limited resources provided by the client.

4.0 Conclusion

The project was a success with the complete and successful commissioning of the Clampton Alluvial gold mine site allowing the intern to gain an understanding of the responsibilities of a Process Control Engineer. The exposure to the mineral processing and mining sector of industry has helped the intern gain a deeper understanding of what is expected of engineers and the role of an engineer in a team. To not just be problem solvers, but also give well thought out logical opinions for solutions to any problem in the project to all participants in the project whether it be engineers of different backgrounds, technicians, operators or clients where communication, hard work and co-operation are most important to a projects completion.

The project on how a PLC is incorporated into a mine site has exposed the intern to how time management, work ethic, scope/time changes and unplanned engineering tasks are handled with support of the team and initiative.



Figure 30: (Pattimore, 2015) – *First handful of processed dirt hand sifted after commissioning complete, a thin ring of gold is visible*

Bibliography

- Rockwell Automation literature library. (2012, October). Retrieved December 14, 2015, from Rockwell Automation:
http://literature.rockwellautomation.com/idc/groups/literature/documents/qr/iasimp-qr010_-en-p.pdf
- Addressing via Rockwell BOOTP/DHCP Server 2.3. (2015, 12 4). Retrieved 10 2015, from Beckhoff Information System:
https://infosys.beckhoff.com/content/1033/bk9055_bk9105/images/BootPServer_2.gif
- Allen Bradley . Rockwell Automation. (2015, December 28). *Using the BOOTP Utility to Configure an IP Address*. Retrieved December 2015, from Rockwell Automation:
<http://www.rockwellautomation.com/global/relianceelectricdrives/bootp-utility.page>
- Allen Bradley . Rockwell Software. (2010, September). *Rockwell Literature FactoryTalk® Activation Manager*. Retrieved December 12, 2015, from Rockwell Automation:
http://literature.rockwellautomation.com/idc/groups/literature/documents/pp/ftalk-pp025_-en-p.pdf
- Allen Bradley. Rockwell Automation. (2010, 9). *Logix5000 Controllers Advanced Process Control and Drives Instructions*. Retrieved 10 2015, from Rockwell Automation:
http://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1756-rm006_-en-p.pdf
- Clampton. (2015, 09 28). Black-box Test.
- Clampton. (2015, 08). Site - Commission.
- Con-struct. (2012, 02 23). *Piping and instrumentation diagram of pump with storage tank*. Retrieved from Wikimedia:
https://en.wikipedia.org/wiki/Piping_and_instrumentation_diagram#/media/File:Pump_with_tank_pid_en.svg
- Fluke Corporation. (2015). *How to test for continuity*. Retrieved from FLUKE: <http://en-us.fluke.com/training/training-library/test-tools/digital-multimeters/how-to-test-for-continuity-with-a-digital-multimeter.html>
- Hedberg, J. (2006, 08 25). *Factory Acceptance Test Guidelines*. Retrieved from Science Partner (SP):
<https://www.sp.se/sv/index/services/functionalsafety/Documents/Factory%20acceptance%20testing%20guideline%20Process.pdf>
- Jackson, d. D. (n.d.). *Programmable Logic Controllers -- Lectures*. Retrieved from the university of Alabama: <http://jjackson.eng.ua.edu/courses/ece485/lectures/LECT08.pdf>
- Mbeychok. (2007, 01 01). *RefineryFlow*. Retrieved from wikimedia:
<https://commons.wikimedia.org/wiki/File:RefineryFlow.png>
- Melore, P. (2011). *Timers*. Retrieved from PLC.net: <http://www.plcs.net/chapters/time16.htm>

- Pattimore, B. (2015, 10). Clampton PLC Project. Perth, Western Australia, Australia.
- Pattimore, B. (2015). Move Function Block.
- plcdev.com. (2015). *Using Ladder Logic for Gray Code Conversion*. Retrieved 10 2015, from PLCdev:
<http://www.plcdev.com/files/plcdev/images/Gray%20code%20conversion%20ladder%20logic%20rslogix.png>
- SATYENDRA. (2015, 04 09). *GRINDING MILLS AND THEIR TYPES*. Retrieved from ispatguru.com/:
<http://ispaguru.com/grinding-mills-and-their-types/>
- Southern testing. (2013). *F.A.T. testing*. Retrieved from Electrical Testing Safety: <http://electrical-testing-safety.co.uk/electrical-testing/f-a-t-testing>
- Stienecker, D. A. (2011). *Ladder Logic*. Retrieved 10 2015, from Dr. Stienecker's TECH site:
<https://adamwsonu.files.wordpress.com/2010/03/countersapplication.png>
- The Internet Centre. (2015). *Ethernet Cable – Color Coding Diagram*. Retrieved 10 2015, from The Internet Centre Inc.: <http://www.incentre.net/wp-content/uploads/2015/02/ethcable03.gif>
- WA, C. S. (2015). *Control Philosophy*. Perth.
- WA, C. S. (2015). *Instrument List*. Perth.
- WA, C. s. (2015). *PF/ID process flow and instrumentatin drawing*. Perth.
- WA, C. S. (2015). *PLC I/O list*. Perth.
- Williams, L. (2006). *Testing Overview and Black-Box Testing Techniques*. Retrieved from Realsearch:
<http://agile.csc.ncsu.edu/SEMaterials/BlackBox.pdf>

Appendix

- FactoryTalk® Activation Manager
http://literature.rockwellautomation.com/idc/groups/literature/documents/pp/ftalk-pp025_-en-p.pdf
Manual for the Activation Manager Software use to remove license keys from MAC address and re-activate on different MAC address/computer.
- BOOTP/DHCP manual
https://www.google.com.au/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwiB8PH66MnKAhVQ4mMKHZJeAGQQFggcMAA&url=https%3A%2F%2Frockwellautomation.custhelp.com%2Fci%2Fattach%2Fget%2F8311%2F1164808462&usg=AFQjCNEVWnUb19a3Uu_HQOwczzxYd1gwpQ&sig2=E5kT0ko75AAMQsDurvzc5A&cad=rja
Direct download link for the word document by Rockwell Automation explaining how the IP address of the PLC is changed from
- ICONs Manual/Specification
<http://www.iconconcentrator.com/espanol/images/stories/documents/i150-i350-Installation-Operation-Manual-EN.pdf>
The operation of the ICONs installed on site is important to understand as operating the ICONs at their full potential will create maximum yield of product, there for the boundaries in the Control Philosophy such as water pressure and drive operation are based off producing optimal conditions for these gravity concentrators.
- Rockwell Automation FactoryTalk Network Design
http://literature.rockwellautomation.com/idc/groups/literature/documents/gr/iasimp-gr010_-en-p.pdf
The original network was based of this manual that was then modified to an architecture that is better suited to the hardware and setup up provided by the client. This gives insight on how communication is setup showing that the software used to monitor and change the PLC code can work independently of the execution of the PLC code on the PLC.
- RSLinx® Classic Getting Results Guide
http://literature.rockwellautomation.com/idc/groups/literature/documents/gr/linx-gr001_-en-e.pdf
Software that is installed with RSLogix and FactoryTalk that establishes communication between RSLogix AND FactoryTalk with the PLC. Allowing the monitoring of live data from the PLC and the upload/download of code to and from the PLC to RSLogix5000
- Logix5000 Controllers General Instructions Reference Manual
http://literature.rockwellautomation.com/idc/groups/literature/documents/rm/1756-rm003_-en-p.pdf
General function blocks and Ladder Logic used to create the rules controlling the system in context with the Rockwell software to be programmed on (Logix).

- Logix5000 Controllers Add On Instructions
http://literature.rockwellautomation.com/idc/groups/literature/documents/pm/1756-pm010_-en-p.pdf
 Add On Instructions (AOI) used repeatedly to create function blocks throughout the coding the PCL. This document was referenced a lot to get an understanding of how AOI work and how to create one in context of the Rockwell Automation Software used.
 - 1756 ControlLogix Controllers, Revision 17
http://literature.rockwellautomation.com/idc/groups/literature/documents/rn/1756-rn017_-en-e.pdf
 This document explains the differences between Logix versions, this was needed to understand the difference in capabilities between the software installed on the office PLC and the PLC at the Factory Acceptance Test.
 - Dell PowerEdge R730 Spec Sheet
http://www.dell.com/learn/us/en/04/shared-content~data-sheets~en/documents~dell_poweredge_r730_spec_sheet.pdf
 The specifications of the rack mounted server that was to be used in the initial design of the network. This was later removed but understanding the limitations of the hardware was important to the PLC as to determine where data should be stored or access to license keys would be held and how they would be distributed.
 - FactoryTalk® View Site Edition User's Guide
http://literature.rockwellautomation.com/idc/groups/literature/documents/um/viewse-um006_-en-e.pdf
 The initial software used to create the HMI that was later changed to Citect. This document was used to gain an understanding of how the HMI and PLC interact with the transfer and manipulation of data from the PLC.
- AND the output will only be true if all inputs are true otherwise false.
 - OR the output sends a true as long as one of the inputs sends a true if all inputs are false output a false.
 - XOR exclusive or only sends a true if one of the signals is true if both signals are true the output will be false unlike the OR logic
 - NOT used to reverse the input logic, input is true then output is false.
 - NAND considered a NOT for the AND logic where it sends a false if inputs one and two are true but otherwise true.

- XNOR sending a true when either both inputs are true or false otherwise neither, this differs from and as and does not allow both false to be true.
- NOR only outputs true when inputs are both false.