Samuel Adams

# The Automation of ABB: Breakers and Switches' Spindle Milling Machine

Thesis
Spring 2016
School of Engineering
Automation Engineering

SeAMK
SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

## Thesis abstract

Faculty: School of Engineering

Degree Programme: Automation Engineering

Specialisation: Electrical Automation

Author: Samuel Adams

Title of thesis: Automation of ABB: Breakers and Switches' Spindle Milling Machine

Supervisor: Ismo Tupamäki

Year: 2016 Number of pages: 48

The company Co-Automation was tasked with updating the Spindle Milling Machine for ABB: Breakers and Switches in Vaasa. The old machine was semi-automated and the goal was to make it fully-automated. Co-Automation also wished to use this project as a test project to try out different components and systems for their viability for future use with the company's existing systems that centred on their close co-operation with Omron industrial automation systems and ABB Robotics.

In this thesis the electrical system was designed and schematics drawn, components were chosen and purchased. The installation work was carried out and the machine was programmed and commissioned, also the basic principles of electrical design, component selection and PLC and HMI programming are explored in the theory section of this thesis.

Keywords: semi-automated, fully-automated, schematics, PLC, HMI.

SEINÄJOEN AMMATTIKORKEAKOULU

## Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Tutkinto-ohjelma: Automaatio

Suuntautumisvaihtoehto: Sähköautomaatio

Tekijä: Samuel Adams

Työn nimi: ABB: Breakers and Switches karajyrsinkoneen automatisointi

Ohjaaja: Ismo Tupamäki

Vuosi: 2016 Sivumäärä: 48

Yhtiön Co-Automation saama toimeksianto koski ABB: Breakers and Switches karajyrsinkoneen päivittämistä. Tavoitteena oli tehdä alun perin puoliautomatisoidusta koneesta täysin automatisoitu. Tämän lisäksi Co-Automation tahtoi käyttää kyseistä projektia testatakseen erilaisten komponenttien ja järjestelmien yhteensopivuutta yhtiön jo käyttämien Omron ja ABB Robotics järjestelmien kanssa.

Opinnäytetyöhön kuului sähköjärjestelmän suunnittelu, sähköpiirustusten piirtäminen, komponenttien valinta ja ostaminen, sekä järjestelmän asennus, ohjelmointi ja käyttöönotto. Opinnäytetyön teoriaosa käsittelee sähkösuunnittelun, komponenttien valinnan, sekä PLC- ja HMI-ohjelmoinnin perusperiaatteita.

Asiasanat: puoliautomatisoitu, täysin automatisoitu, sähköpiirustukset, PLC, HMI.

# TABLE OF CONTENTS

## Terms and Abbreviations

| | |
|---|---|
| **CAN Bus** | Controller area network. A bus system designed for use in cars. |
| **CoDeSys** | A development environment for programming controller applications according to the IEC 61131-3 standard. |
| **EDM** | External device monitoring. |
| **Ethernet IP** | Ethernet Industrial Protocol. An industrial Field-Bus. |
| **Ethernet TCP/IP** | Ethernet based transmission control protocol and internet protocol. A method of transmitting packets of data over the internet or local network. |
| **CPX Platform** | An embedded automation platform by Festo. |
| **Ethercat** | An Ethernet based Field-Bus system developed by Beckhoff. |
| **Flash** | An electronic computer storage medium that can be electrically programmed and erased. |
| **HMI** | Human Machine Interface. Any method which allows humans and machines to interact. |
| **I/Os** | Input and outputs. |
| **PLC** | Programmable Logic Controller. A programmable computer used for automation control. |
| **Servo Motor** | A motor that operates with a closed loop feedback control system. |
| **Stepper Motor** | A brushless DC electric motor. |
| **ST-Programming** | A text based automation programming language. |

| | |
|---|---|
| **PC** | Personal Computer. |
| **Rotary Cam Switch** | An industrial switch which can contain many contacts which are switched by a central axel. |
| **Safety PLC** | A Programmable Logic Controller specifically designed for the control and monitoring of safety applications. |
| **MCB** | Miniature Circuit Breaker. An automatic resettable fuse. |
| **RAM** | Random Access Memory. A form of computer data storage. |
| **Runtime** | The time during which a program is running. |

## Tables, Figures and Pictures

# 1 INTRODUCTION

The thesis was undertaken for Co-Automation in Vaasa. They are a company which specialises in the complete design, manufacturing and programming of automated cells and lines on both a mechanical and electrical level. They are also the only company in Finland to be an Authorised Value Provider to ABB Robotics Finland.

The company usually uses Omron automation products. However, over the past four years, a lot of their work has come from ABB in Vaasa and the company wished to explore the viability of using ABB HMIs and other makes of PLCs in case of their customers requesting it in the future. Therefore, this project was the first project undertaken by the company where a Festo PLC, stepper motor and an ABB HMI were used.

The goal of the thesis was to explore these possible future options for the company by electrically re-designing the spindle milling machine, choosing and purchasing components for it and programming and commissioning it for ABB: Breakers and Switches in Vaasa. On a practical level this was undertaken by visiting the ABB premises in Vaasa, and meeting with their representatives. The old machine was then studied and it was mutually agreed upon what had to be done to improve it. An interview with an automation expert from ABB took place and a training day with Festo Oy Ab in Vantaa was undertaken. The electrical schematics were drawn up and the components purchased. The PLC, stepper motor and HMI were programmed and tested in Co-Automation's premises and then later over the course of four days the machine was rebuilt, wired, programmed and commissioned in ABB's premises in Vaasa.

# 2 BASICS

In this section the basic principles of the equipment used in the project are explored so as to make the thesis' later sections more understandable. As such, the reader will be more familiar with problems encountered and overcome.

## 2.1 PLC

In the following section there is given a brief history of the PLC and its general operating principles. The most common types and configurations found in industrial automation are also looked into sufficiently enough to give the reader a basic understanding of what PLCs are.

### 2.1.1 A Brief History

PLC stands for programmable logic controller. They were first developed in the late 1960s to early 1970s. Previously industrial processes were controlled by strictly mechanical means, such as pistons, cogs, levers and gear systems. From the late 19$^{th}$ century up to the mid or late 20$^{th}$ century, as electrical systems developed, a lot of these mechanical process controls were replaced by electrical relays and switches. This had clear advantages over a purely mechanical system yet it was inflexible and very time consuming to reconfigure the system. It was no longer flexible enough to keep up with the constantly changing industrial manufacturing process. PLCs were then developed so that the wiring could be reprogrammed without the need for physical rewiring. (Hackworth 2004, 36.)

The first devices were cumbersome and in essence just switches. The modern PLC, however, with the advancement of computers, has developed into a powerful controller capable of complex mathematical functions and generating and receiving analogue signals. (Hackworth & Hackworth 2004, 36.)

### 2.1.2 Configurations and Operating Principles

PLCs come in many shapes and forms. However, in regards to the control of industrial processes the most common are compact PLCs (Picture 1), modular PLCs (Picture 2) and rack PLCs (Picture 3).


Picture 1. Omron CP1E Compact PLC (Omron 2015)

Picture 2. Siemens SIMATIC S7-300 Modular PLC (Siemens 2015)

Picture 3. Omron CS1G/H Rack PLC (Omron 2015)

Regardless of the type of PLC they almost always share the same system components, those being a processor, a mounting rack, input and output modules and an interface with a programming unit or a PC. (Hackworth 2004, 37-41.)

The operating principles are relatively simple if thought of in terms of the number of steps involved, basically, there are two steps. Firstly, when the PLC is turned on, it solves the PLC code. Secondly it updates the inputs and outputs accordingly. During the start-up process there are more complex processes at work which are hidden from the user, such as diagnostic checks and other checks on the working of various communications with other hardware through field-buses or other means. That being said one will gain a good understanding of the basic operating principle if one just considers the two aforementioned steps. (Hackworth 2004, 43.)

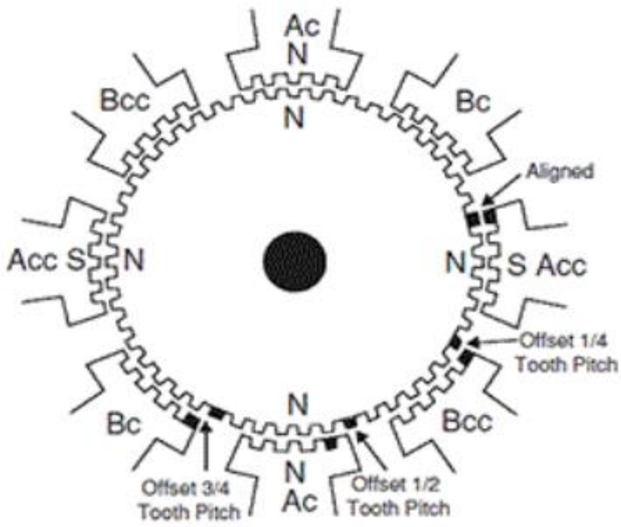
## 2.2 Stepper Motors

The unique and useful aspect of a stepper motor is that its shaft rotates at discrete pre-set angular intervals dictated by the pulses it receives. It turns one interval for every pulse received. This quality means it can be easily controlled by micro-controllers, PLCs and PCs. Its accuracy is dependent on how many pulses it receives per rotation. When it receives one pulse, it will move one step and stay at that position until it receives the next pulse. The direction of movement is defined by the state of the direction line input, it being on or off. (Hughes 2006, 305.)

### 2.2.1 Hybrid Stepper Motors

A hybrid stepper motor is a type commonly used in modern industrial applications and such a type was used in this study. Hybrid motors provide very good torque, speed and step resolution. The rotor of a hybrid stepper motor is axially magnetised meaning that one end is polarized north and the other south. The rotor and the stator have teeth which align with each other in various configurations during the rotation as seen in picture 4. A common rotor cut resolution is 1.8 degrees. The rotor has then two cups whose teeth are at a 3.6 degrees angle of separation. The stator is then comprised of a two phase construction where the poles that are at 180 degree angle to each other are always of the same polarity and the poles which are at 90 degree angle to each other are always of a different polarity. As seen in picture 4, if there are 50 teeth on the rotor then the angular difference between the teeth will be

7.2 degrees. As it moves some teeth will be out of line by ¾, ½ or ¼. When the motor moves, it will move to the next aligned tooth which is ¼ of 7.2 degrees. Therefore, the resolution of the motor is said to be 1.8 degrees. (NMB 2004.)




Picture 4. Internal construction of a hybrid stepper motor (NMB 2014)

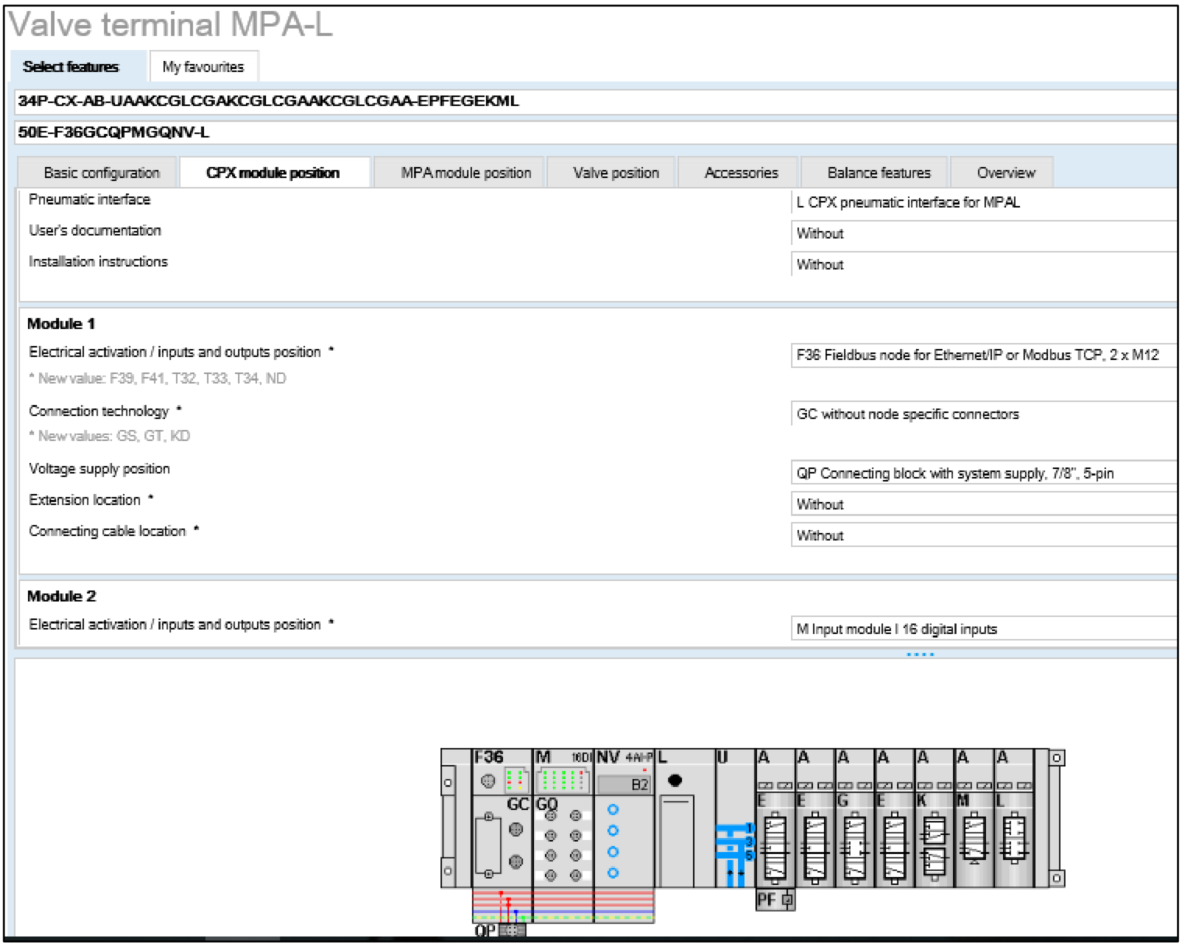
The motor moves when it receives pulses from the controller. The movement is realised by constantly reversing the polarity of the two phases, +A, -B, -A, +B, +A (clockwise) or +A, +B, -A, -B, +A (anticlockwise). (Hughes 2006, 317.)

### 2.3 The Festo CPX System

The idea of the Festo CPX system is to provide a complete automation platform which can be configured as a field IP65-IP67 unit or as the more conventional method of having it mounted in an electrical cabinet. It is fully configurable and can be an economical small embedded I/O system all the way to a fully functional PLC with motion control, temperature inputs, pneumatic valves and many other options all in the one unit that, if designed carefully, can make the need for an electrical cabinet redundant in some applications. Over 90% of the most common field-bus protocols are supported. (Festo 2016, 2-4.)

### 2.3.1 CPX Configuration Options

As stated in the previous section the configuration options are quite large. Festo have made the selection relatively simple through an effective online configurator. The kind used in this project was a CPX-CEC combination. The CEC means there is an embedded PLC included. There are different PLC's available with different protocols and interfaces. The CPX system can then be joined to a number of different valve terminals. A configuration commonly used by Co-Automation is the CPX-MPAL. The MPAL is a pneumatic series of valves with small to medium flow capacity. (Festo 2016, 2-19.)




Picture 5. Festo CPX MPAL Terminal Configurator (Festo 2016)

An example configuration from the online configurator can be seen in picture 5. This gives some idea of the different options available. An explanation of the various terminals is the following: in position 1 there is an F36 block which is a Field-Bus node for Ethernet I/P or Modbus TCP. The second block is a 16 digital input card with an 8x M8-4pin interface. In position three there is a vacuum analogue input card. This measures the amount of vacuum on the output of an ejector by bringing the pneumatic hose directly into the unit. The forth block is an end block for the electrical terminals. The fifth block is where the pneumatic feed is brought in and the rest of the units are different pneumatic valves of various types, some with built in pressure regulators.

## 2.4 Safety PLC

A safety PLC is a PLC specifically designed to control and monitor safety processes in a machine application according to IEC 61508. It shares common attributes with a standard PLC in regards to physical appearance and having inputs and outputs as can be seen in picture 6, but the similarities with PLCs end there. Safety PLCs can come as stand-alone modules as seen in picture 6 or appear alongside normal inputs and outputs in a remote I/0 configuration or connected to the actual PLC itself. (Rockwell Automation 2002, 2-3).

Picture 6. Omron G9SP stand-alone safety PLC (Omron 2016)

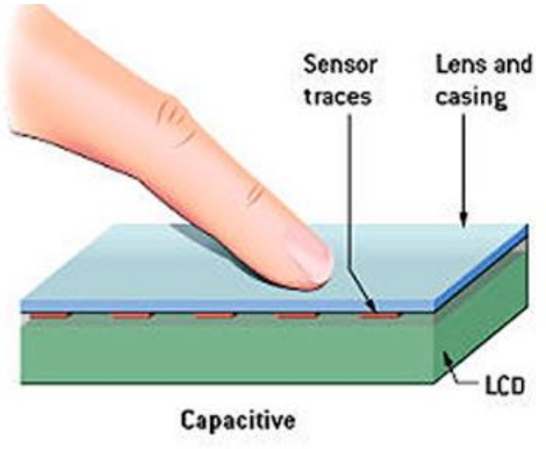
A Safety PLC differs from a regular PLC in its architecture and how its inputs and outputs function and are monitored. In contrast to a regular PLC a safety PLC has redundant microprocessors, Flash and RAM which are monitored by a watchdog and a continuous protection circuit. The functionality of the inputs and outputs are also tested during start up to insure they are working properly. Each output has dedicated safety switches and microprocessors to test the output's integrity and functionality at each stage of the operating process. (Rockwell Automation 2002, 3-4).

## 2.5 HMI Panels and Touch Screen Technology

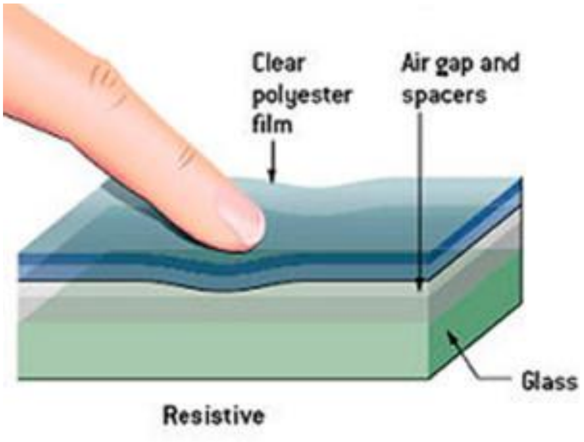
HMI stands for Human Machine Interface. In industrial automation touch screen panels are the most common type of HMI, sometimes with physical buttons working alongside the touch screen and sometimes without. The term HMI is a very broad term that incorporates any medium that allows humans and machines to interface together. (Texas Instruments 2014, 2).

### 2.5.1 Capacitive and Resistive Touch Screen Technology

Capacitive touch screens are the most common touch screens available nowadays. Most smartphones and tablets use capacitive touch screen technology. Capacitive touch screens are made up of layers of a conductive material such as Indium Tin Oxide. A protective cover, such as glass or plastic, isolates the conductive material from the outside environment. When an electrically conductive material such as a human finger touches the screen, an electrical circuit is completed at that location. Sensors in the protective layer register this as a touch and can determine its location through analysing which sensors were activated. Since they have no moving parts they are durable when used gently and are very sensitive to touch. As a result, it is a popular technology used within smartphones and tablets. (Baanto 2015)




Picture 7. A capacitive touch screen (Baanto 2015)




Picture 8. *A resistive touch screen (Baanto 2015).*

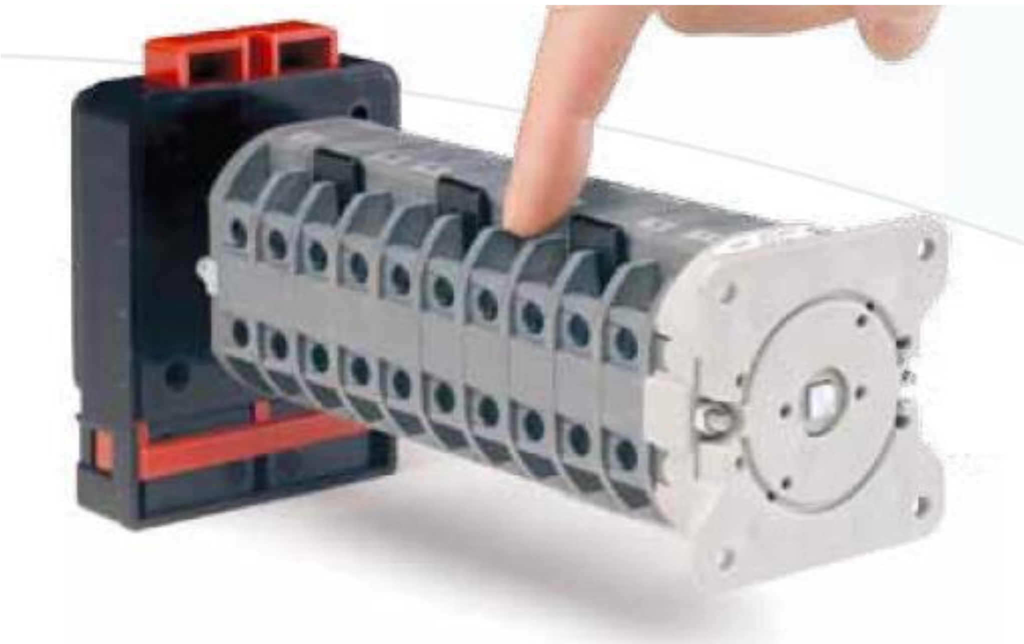
Resistive touch screens are also very popular and used in applications where more robustness required. Two layers of glass or plastic are each coated with a layer of Indium Tin Oxide with the conducting sides facing each other and separated by a small air gap. When pressure is applied, the layers touch each other and a small amount of current flows in that location. Thus, the panel knows the location of the press. As mentioned, resistive touch screens require pressure and because there are moving parts they may not last as long as a capacitive touch screen. On the other hand, they are not affected by dust or water and are therefore better suited for industrial applications where there may be dirt or water and people are wearing gloves. The HMI used in this work was of the resistive type.

# 3 THE ORIGINAL MACHINE

Before going into the actual work undertaken it would be important to explain how the old machine looked and operated along with its various issues and shortcomings. This will give a good background to understanding why the commissioning of an update of this machine was issued by ABB: Breakers and Switches. Also the purpose of the machine and what it makes and where the product is used will be explained in this section.

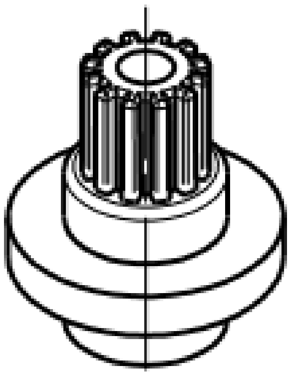
## 3.1 Background of the product and the original spindle milling machine

In 2013 ABB: Breakers and Switches commissioned Co-Automation to make them a semi-automated machine capable of milling cuts into plastic spindles. These spindles are used inside some of ABB's rotary cam switches as seen in Picture 9. The spindles are built together inside the switch to form the shaft which triggers the contacts to open or close.

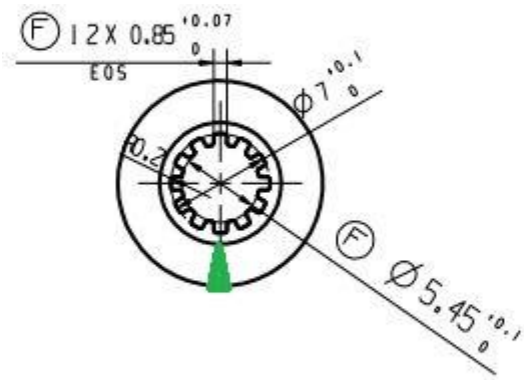


Picture 9. ABB Rotary Cam Switch (ABB 2016)

The cut-outs in the spindle depend upon how many contacts are used in the switch and in what configuration the contacts are needed. Hence there is a very wide range of different types of spindles which can be made. The cuts are made at 15 degree intervals so there are 24 different possible cut-out locations per spindle. Some spindles only require 1 cut-out, others may require as much as 18 cut-outs. An example of the spindle can be seen in Picture 10 and a simulated example of a cut-out made in a spindle highlighted in green in Picture 11.



Picture 10. ABB Spindle (ABB 2014)




Picture 11. ABB Spindle with simulated cut-out (ABB 2014)

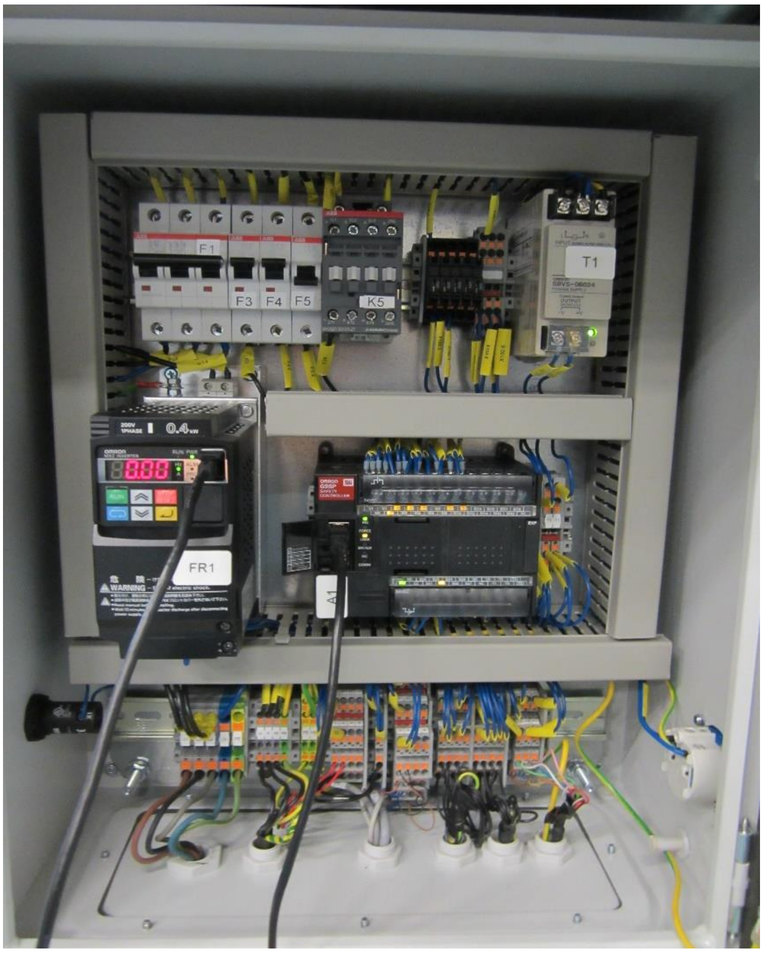
The spindle milling process is a vital part of the switch’s assembly and as order volumes increased over time it was required that a machine be made that enabled ABB to mill their own spindles to order.

### 3.2 The design and operating principle of the original machine

The original machine was built as a table frame using aluminium profile for the legs and an aluminium plate for the table top. On the left of the table there was a pneumatic soft-start valve along with an electro pneumatic valve rack as displayed in picture 12.

Picture 12. Pneumatic soft-start valve (Co-Automation 2013)

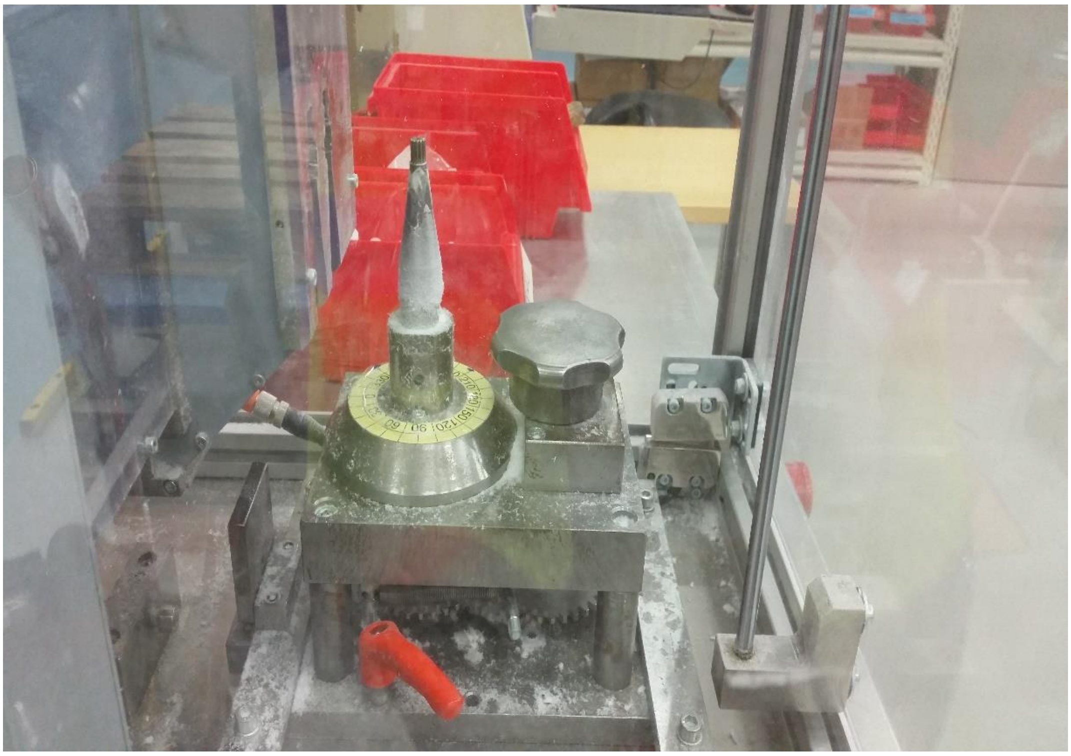
To the right under the table was an electrical cabinet where was located all the control equipment which consisted of MCBs, fuses, a 24VDC power supply, a safety PLC and a frequency convertor used to control the speed of the cutting blades. These are displayed in picture 13.




Picture 13. Control equipment (Co-Automation 2014)

The spindle holder and cutting blade was surrounded by protective plastic sheeting. The door opened and closed using a pneumatic cylinder and the door was confirmed to be in its closed state using a magnetic, contact free, safety switch which was connected to the safety PLC. The spindle was placed inside and attached to the metal holder. This metal holder could be manually turned at 15 degree increments. This was achieved using mechanical cogs under the table and a lever above the table. Using a removable pin it could also be moved to the side. This was necessary because the machine needed to be able to mill two different sized spindles which needed different depths of cuts. Hence there were two blades and sideways movement was possible.

The safety PLC controlled all the safety functions and also controlled the simple automatic sequence of the machine. The operating principle was as follows: when the power was powered on and if the emergency circuit was ok then by pressing the blue reset button the door would open. Then spindle would be placed inside and turned to the desired position as seen in picture 14.



Picture 14. Turning cog (Co-Automation 2015)

The automatic sequence was started by pressing both black buttons at the same time. The door would close, the blade would turn and get up to speed and a second cylinder under the table would drive the cog unit towards the blade and perform the cut. The cylinder would then drive back and when the blade had stopped turning the door would open and the cycle was completed.

### 3.3 Limitations of the original machine

Although the original machine was designed and built as requested by ABB, it proved over time to be too slow as demand for the rotary cam switch grew. Some of the spindles which required many cut outs could take up to 9 minutes to complete.

At first, it was tried to improve the manufacturing time by decreasing the deceleration of the cutting blade in the parameters of the frequency convertor. The momentum of the heavy blade proved too great and even adding an external braking resister to the frequency convertor would not have been enough. As a result a solution to cut more time off the manufacturing process was needed.

# 4 THE UPDATING OF THE MACHINE

In this section the process of training, pricing, redesigning, programming and commissioning the new machine will be laid out. Also, the obstacles encounter along the way will be explained, along with the solutions found to overcome them.

## 4.1 Initial Design Idea

The original machine was delivered in 2013 and since then its problems and limitations had become apparent. ABB had requested that we try and improve some of these problems, the major one being that it took too long to make some of the spindles which had many cut outs.

The initial approach taken in trying to solve this problem was to find a way to speed the ramp-up of the frequency convertor and get the blade up to its cutting speed faster. Also, inversely, to get it to slow down faster. The frequency convertor was and Omron MX 0.4KW model. Therefore, because of the physical mass of the cutting blade, there was a minimum start-up time that could be achieved and this was directly related to the output power of the frequency convertor. If the start-up time was to be decreased then the frequency convertor would have to be changed to a more powerful model. This was impractical because it would be physically too big for the electrical cabinet. The blade could be slowed down faster by adding an external resister to the frequency convertor to dissipate the mechanical energy electrically in the form of heat. This was a viable option and was presented to ABB. It was estimated that about 3 seconds could be cut from the ramp-down (deceleration) phase in this way.

The other option to speed up the process was that the machine cut all the desired cuts in one operation and the motor would slow down and the doors open only when the product was finished. In order to do this it was necessary to change the manual mechanical turning process to an automatic one. With this in mind three different offers were made by Co-Automation to ABB.

**The Three Offers made to ABB**

1. To add a braking resister to the frequency convertor.
2. To make the machine fully automated.
3. To make the machine fully automated and add a braking resister.

ABB decided upon option two. A meeting was arranged with the ABB representatives and it was verbally agreed upon what was required concerning the redesigning of the machine. A mechanical engineer then designed the operating principle using 3D drawing software. The 3D design was submitted to and approved by ABB.

The design work itself was then started. To make the machine fully automated it was necessary to add three main components to the machine.

**The Three Main Additional Components Required**

1. A motor, to turn the spindle.
2. A PLC, to control the logic of the motor and the automation sequence.
3. A HMI, so the user could input what cuts were required.

### 4.1.1 The Motor

The motor had to be capable of driving to pre-set positions to a high degree of accuracy. The two most viable options were to use an AC servo motor or a stepper motor. In conjunction with the other goals of the project it was decided to use something that we thought had future potential for us but that hadn’t been used before.

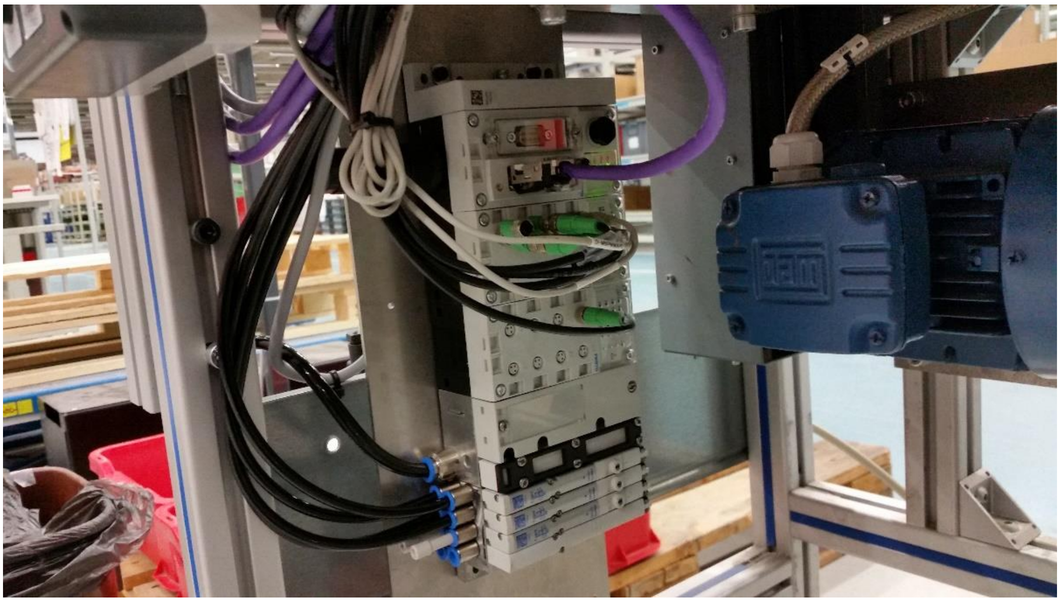
Picture 15. Festo CMMO-ST Controller (Festo 2016)

Picture 16. Festo EMMS-ST Stepper Motor (Festo 2016)

A servo motor would have had a very high degree of positioning accuracy and the motor would have been physically small, however, the servo amplifier was too big and would not have fit into the electrical cabinet. It was also almost twice the price of a stepper motor, hence, a servo motor was decided against. A stepper motor was well suited for the job, the controller was small and the accuracy was good enough for the application. The only downside being that the motor which was needed would have to be physically large in order to have enough holding force to hold the spindle in position as it was being cut. The motor chosen was a Festo EMMS stepper motor, a belted gear unit and a CMMO stepper controller.

### 4.1.2 The PLC

It was necessary to add a PLC because the Omron safety PLC would have been incapable of controlling the stepper motor. Also, there was no more room in the electrical cabinet and considering the other objective of testing future products for their suitability for future use, it was decided to use a Festo field mounted PLC/CPX unit. This enabled the PLC, I/Os and pneumatic valves to be mounted in the field in one unit.



Picture 17. Festo PLC/CPX Unit (Co-Automation 2015)

The PLC module selected was a T07 CoDeSys embedded controller with a M8 4-pin module with 16 inputs, a M8 3-pin module with 8 outputs and then the pneumatic valves. This model of PLC is capable of controlling electric drives and it communicated with the CMMO stepper controller through the CAN Bus protocol and with the ABB HMI through the standard Ethernet TCP/IP. These two different protocols were able to be implemented over the same Ethernet cabling, seen in picture 17 as the purple cable.

### 4.1.3 The HMI

It was clear early on that it would be necessary to add a HMI panel to the machine so that the user could tell the PLC what positions had to be cut out on the spindle. The simplest thing to do would have been to use a Festo HMI, but considering the other goal of the project being to test out ABB HMI panels for potential future usage it was decided to use an ABB CP635 panel with built in Runtime. This panel was economical and well suited to the simple task at hand.


Picture 18. ABB CP635 Panel (Co-Automation 2015)

This panel was, on paper, able to communicate with the Festo PLC as they both use the CoDeSys programming environment. However, when considering their versions of the Ethernet protocol, the ABB HMI uses a CoDeSys v2.3 and a Festo PLC communicates with a Festo HMI using CoDeSys v3 protocol. As a result it was unclear how well their communication would work. This issue however and it's clarification was part of the reason for doing this work in the first place so it was decided to go ahead and try them together anyway.

## 4.2 Training

Once the main components had been chosen and purchased it was necessary for some basic training to be undertaken so as the programming process wouldn't take too long. Both Festo and ABB were eager to provide training as the success of this project would give them both good opportunity for potential future sales with Co-Automation.

### 4.2.1 Training with ABB HMI Panels

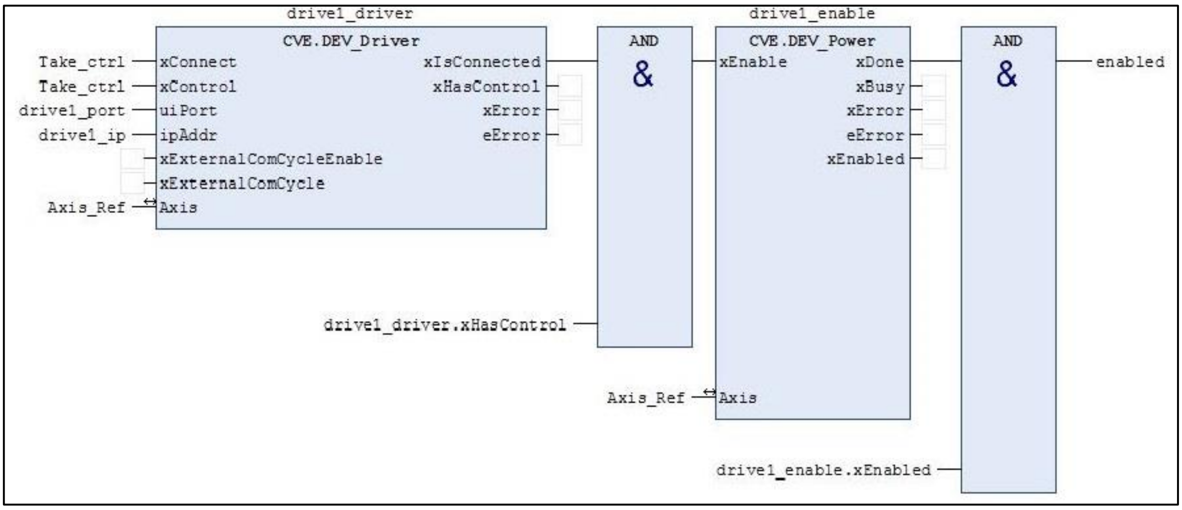
An automation expert from ABB provided an afternoon of training with the Panel Builder software which is used to program ABB HMI panels. The program's basic features and how to build a basic project was looked at.

It was discussed more in depth some potential issues that could arrise relating to this particular project. The expert was not able to say with certainty if the ABB CoDeSys Ethernet protocol would work with the Festo PLC. It was uncertain if the tag import function, which enables the desired variables to be transferred from the PLC to the HMI, would work. It was suggested to try it and if it proved to be unsuccessful then it was a certainty that the Modbus TCP protocol would work. It was explained however that this was more complicated to implement and some basic instruction on how it might work was given. Thus, conformation was given that the communication was nonetheless possible, even though it might be difficult.

### 4.2.2 Training with Festo PLCs and Motors

A training day was provided by Festo at their Finnish headquarters in Vantaa. The agenda for the day was an intensive look at their CoDeSys V3 PLC programming environment and at their FCT (Field Configuration tool) motor configuration software. It was a lot to cover in one day, so only the basic properties of these were explored.

At the time of the training day the writer of this thesis was quite inexperienced with this type of PLC programming software. What was examined was the basic programming structure and how to configure the PLC for first time use. Also it was shown what libraries and function blocks would be needed to establish communication with and control the stepper motor. Some of these control blocks can be seen in picture 19. Other than that it was left to the programmer to figure it out as the project progressed.




Picture 19. Festo’s motor control Function Blocks (Co-Automation 2015)

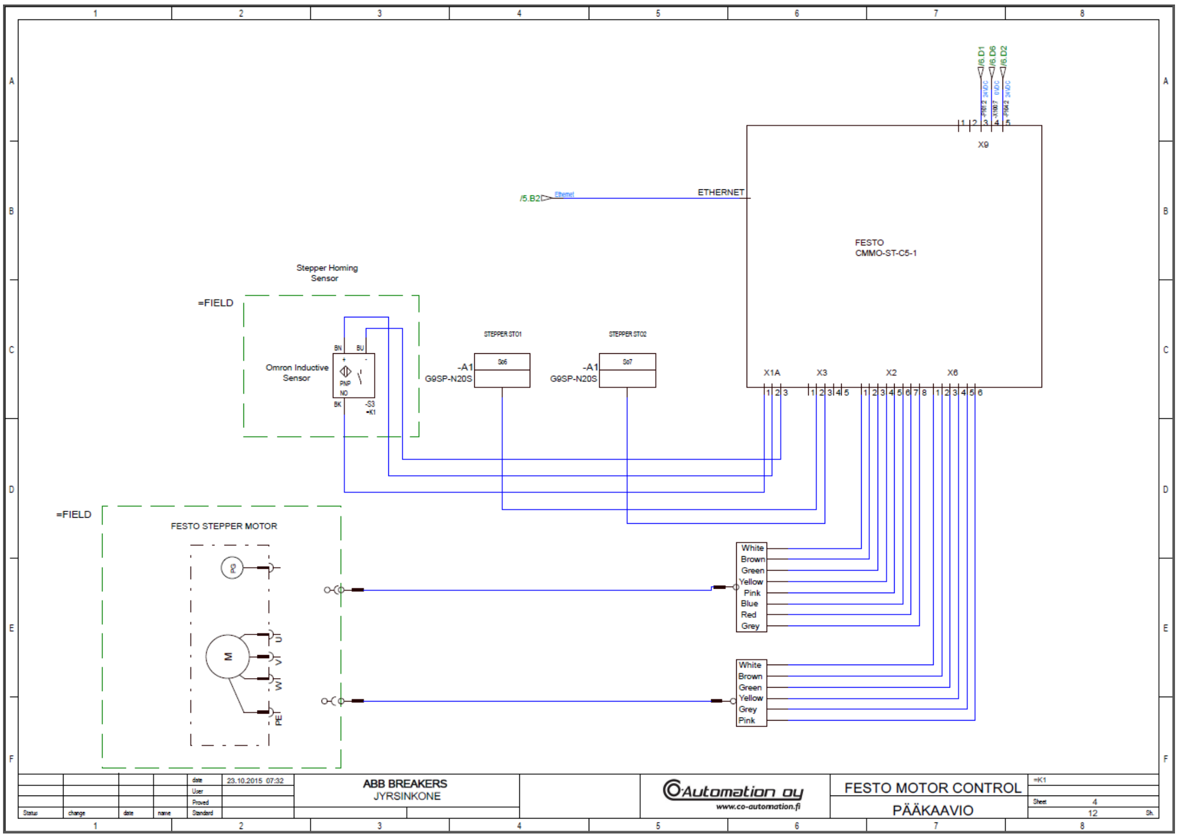
With the motor configuration tool it was explored how to configure the required motor and gearbox. It was also explained how to move to different positions with the stepper motor and after this, because of time constraints, it was left up to the programmer to figure it out as the project progressed.

### 4.3 Electrical Design

Once the basic design of the machine had been settled, the components chosen and the basic training completed, the new electrical schematics were drawn up. The electrical schematics did not require any radical changes, it mainly involved adding the field-bus, moving the main control unit (PLC) into the field, bringing 24VDC to the new components and redesigning the wiring of the safety PLC so it only controlled the safety functions and no longer the automated sequence.

### 4.3.1 The Field-Bus

The Field-Bus had to be added to the machine. This consisted of an Ethernet cable over which the CAN Bus and Ethernet TCP/IP protocols operated.

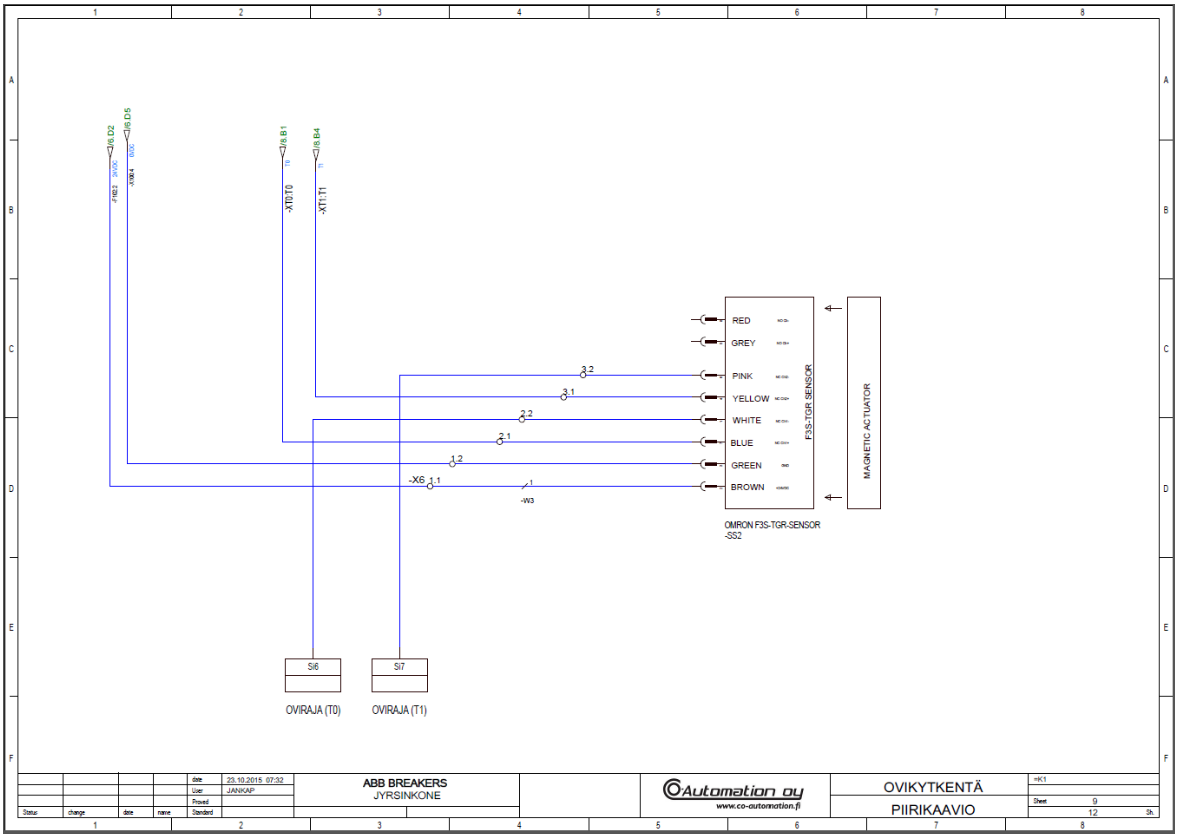



Picture 20. Wiring schematic for the CMMO stepper controller (Co-Automation 2015)

The ABB HMI had two Ethernet ports so this meant there was no need to install a permanent Ethernet switch on the machine. The Ethernet cable left the PLC, went to the HMI and from there to the CMMO stepper motor controller. A temporary Ethernet switch was required when connecting a programming PC to the machine.

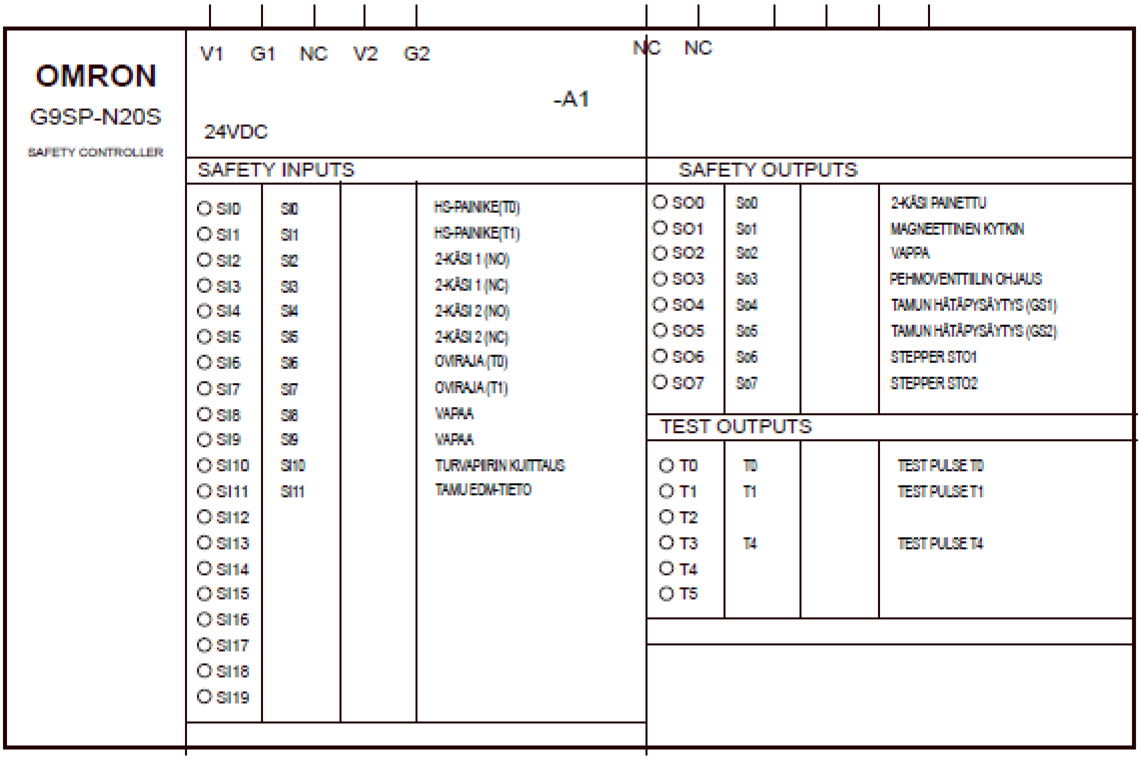
### 4.3.2 Safety

The safety wiring itself needed to be changed from the original machine. Including connecting the safety outputs of the safety PLC to the two safety inputs on the CMMO controller. This ensured that when the door opened or when the emergency stop was activated the motor could not turn. The door position was monitored by a non-contact magnetic safety switch as seen in pictures 21 and 22.




Picture 21. The wiring of the non-contact safety switch (Co-Automation 2015)

Picture 22. Photograph of the non-contact safety switch (Co-Automation 2015)




Picture 23. I/O configuration of the Safety PLC (Co-Automation 2015)

From an electric point of view the safety PLC was rewired and the order and number of inputs and outputs changed as seen in picture 23.

## 4.4 Programming, Installation and Obstacles Encountered

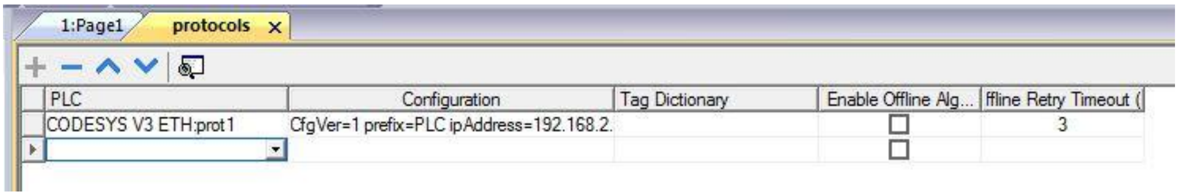
The following section describes the programming, installation on ABB's premises and the final commissioning of the machine. Some of the obstacles encountered during these processes are also described.

Prior to the renovation of the machine in ABB's premises it was possible to test the basic functionality of the PLC, HMI and Stepper Motor and it was during this stage that the communication between the PLC and HMI could be tested.

### 4.4.1 Communication between the PLC and HMI

One of the main goals of this work was to test the Ethernet communication abilities between the ABB CP series HMIs and other PLC systems. This would help determine if Co-Automation could use them in conjunction with Omron's NJ series of PLCs which is generally used as the main PLC by Co-Automation. As mentioned earlier, the ABB automation expert was himself unclear if the tag import function would work with a non-ABB PLC. Since Festo PLCs used the same programming environment as ABB it would be a good litmus test of its other communication capabilities because if it didn't work with a PLC which had the same programming environment then it was unlikely that it would work well with a PLC that had a completely different programming environment.

The basic frame of the PLC program was made and the HMI program was created in ABB's Panel Builder software. As stated earlier it was known that the ABB Panel supported ABB CoDeSys V2.3 Ethernet protocol. It was known that Festo's own HMIs and PLCs used CoDeSys V3 Ethernet protocol. It was advised by Festo to download their own HMI programming software and through Windows File Explorer find the protocols folder and copy the V3 protocol file from there to the ABB Panel Builders protocol folder. When this was done the V3 protocol was able to be added as the protocol in use in the ABB software as seen in picture 24.

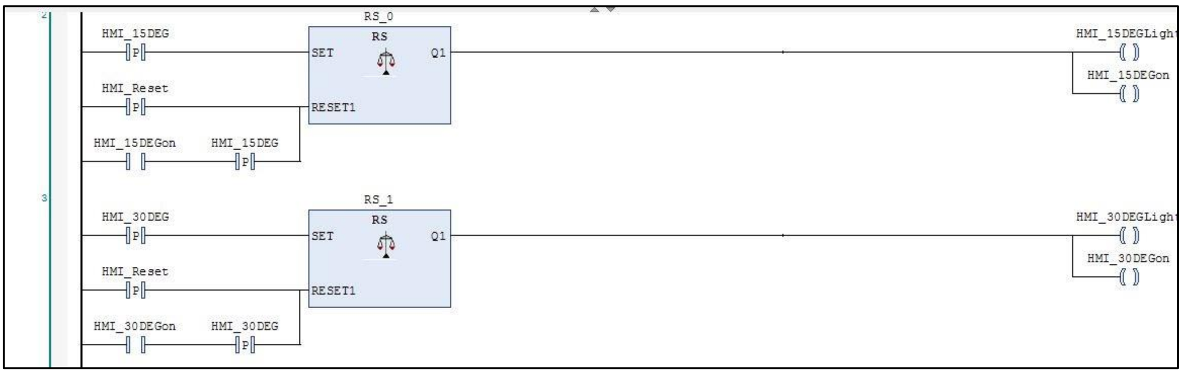



Picture 24. The CoDeSys V3 protocol in ABB's Panel Builder software (Co-Automation 2015)

When this was added and it proved successful the tag import function from the PLC was tested and this proved to be unsuccessful. After much trial and error what eventually proved successful was to manually write the path of every variable in the PLC which was desired to be used in the HMI panel. The variables were written as follows: “192.168.2.10/PLC/Application/GVL/Start:BOOL;”. This way when the PLC’s IP address and the variables location in it was defined, the HMI was then able to control the PLC’s variables. The simulation function in the HMI software didn’t work, so the panel had to be physically connected and the program downloaded to it for it to work. It is also worth adding that the Festo PLC (unlike many other PLCs) required, in a section called “symbol configuration” that every variable had to be given individual read/write permissions. So when a variable was created in the HMI it couldn’t by default be read or written by the PLC until its read/write permissions were enabled.

### 4.4.2 The HMI Program

The HMI program itself was relatively simple fitting onto one page as seen in picture 18. The user had to be able to input which cut configurations he/she wanted to cut out from the spindle. The user would then press the button and a light at the top right-hand corner of the button would confirm that that choice was activated. The button would be pressed again to deactivate the choice.



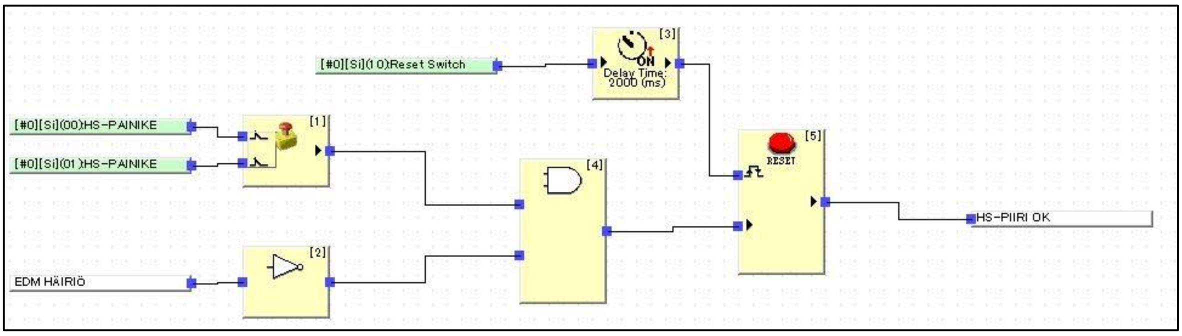

Picture 25. The HMI’s button logic in the PLC (Co-Automation 2015)

Initially the button's set/reset logic was realised in the HMI program itself. However, later on, after commissioning, ABB requested that one button be added that would clear all the selections from the screen. This request revealed a downside to the Panel Builder software in that a set/reset button could not be reset by more than one variable. The ABB automation expert confirmed that the button logic would have to be moved to the PLC. The button logic was then transferred to the PLC and the HMI only read and wrote these values. The logic was realised in the PLC using standard "Reset/Set" function blocks as seen in picture 25 and the extra variables transferred to the HMI.

### 4.4.3 Programming of the Safety PLC

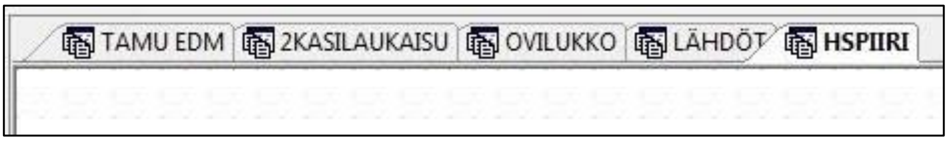
As mentioned earlier, the original machine's sequence and safety functions were controlled from the machine's Omron G9SP-20S safety PLC. Once the Festo PLC was added to the new machine it would have been easy and quite sufficient to just add a safety relay to the machine to control and monitor the safety aspects of the machine. However, in order to simplify the renovation it was easier to leave the Safety PLC in place especially considering it had already been purchased when the old machine was manufactured.

The safety PLC was programmed using Omron's G9SP Configurator. The new program was completely different so the safety PLC had to be programmed again. The safety PLC allows the programmer to program using only certain types of safety function blocks and standard Boolean algebra logic gates. An example from the emergency stop portion of the Safety PLC program can be seen in picture 26.




Picture 26. The Emergency Stop logic from the safety PLC (Co-Automation 2015)

It can be seen form the picture that if both signals from the emergency stop switch were ok and there was no fault from the frequency convertors' EDM, then the emergency stop circuit could be reset when the reset button was pressed for two seconds. The emergency stop signal "ok" would go on and it would enable the motors and turn the machines pneumatic start valve on. A list of the five safety programs can be seen in picture 27.




Picture 27. The five programs in the safety PLC (Co-Automation 2015)

Translated into English from picture 27, the five programs are:

1. The Frequency Convertor's EDM.
2. The two-hand switch.
3. The door lock.
4. The Outputs.
5. The emergency stop circuit.

The Safety program operated as follows: when the electricity was switched on, the machine's blue emergency stop reset button had to be pressed for two seconds. If there was no fault in the frequency convertor and the emergency stop was not pressed down, then the emergency stop circuit was said to be ok. The two motor's safety inputs were enabled and the pneumatic soft-starter valve would switch on and the door would open. The PLC would receive an input that the emergency stop circuit was ok and that next phase of the safety sequence could begin.

Once the user had inputted the desired cut-out positions on the HMI, they needed to press the two-handed switch for 2 seconds and the doors would close. The two handed switch's "ok" signal left from the safety output and went to the PLC input.

Once the door was closed the magnetic non-contact safety switch would confirm the door's closed position. This was realised by the safety PLC sending two test pulse channels through the switch and if they came back to the safety inputs then the safety PLC would send a signal to the PLC confirming the door was closed.

If the emergency stop button was pressed at any time or the EDM signal from the frequency convertor was received then the emergency stop circuit would activate and result in pneumatic pressure being cut off, both motors would shut down and the PLC would receive a signal that the emergency circuit was not ok.

### 4.4.4 The Configuration of the Stepper Motor

Festo's stepper and servo motors are configured using their FCT tool. When the software is opened, the motor type and gearbox/differential (if used) can be entered in. It then limits the configuration options and uses a step by step process until the motor is configured.

Once the motor enable block is configured correctly in the PLC, with the correct IP address and port as shown in picture 19, then an online connection via the CAN bus is possible. From the software it's possible to monitor the different signals, such as safety, that go to the CMMO controller and all the properties can be monitored online there during the operation of the motor. The motor can also be manually driven from the software.

With the version of the CMMO controller used with this project, moving to different positions was accomplished by commanding the motor to move to 24 different recorded positions saved within the CMMO controller itself. Function blocks in the PLC would write the record number to be moved to in a "move to record block". The motor would then move to the record at the acceleration and velocity saved to that particular record number.

The homing type and direction was also configured in the software, the homing type used on this machine was to home to a sensor. This sensor was connected directly to the CMMO controller.

### 4.4.5 The Programming of the PLC

The PLC program in the Festo PLC was programed using the function block and ladder diagram programming languages. The program itself was made up of four separate programs.

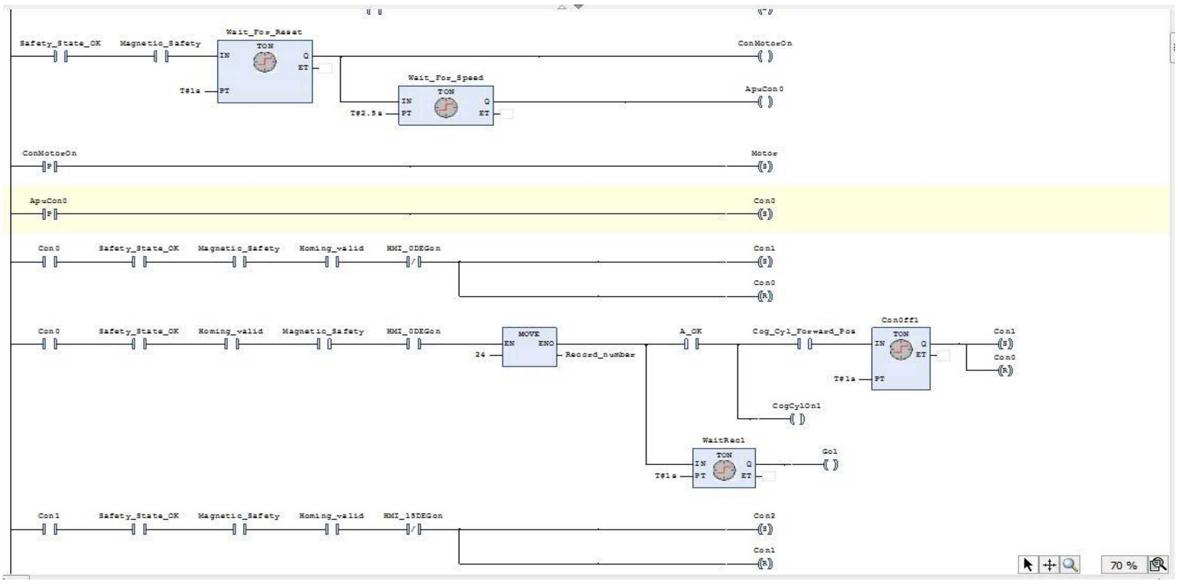
**The Four PLC Programs**

1. Permissions
2. Motor Control
3. Main
4. Outputs

The permissions programme was programmed using ladder diagram. Essentially, it checked if the correct conditions were present to allow the main program to start. It checked if there was at least one button selected on the HMI panel and if the emergency stop signal from the safety PLC was ok. If these two things were ok and there was no fault from the stepper motor or frequency convertor then the start permission value was activated.

The motor control block was programmed using function blocks from the Festo CVE.DEV stepper motor control library. As seen in picture 19, the first block established communication with the CMMO controller and the following block enabled the stepper for use. There was a “write record block” in which the record number which had to be moved to was written in an integer form. A “move to record block” would then insure that the motor moved to the record which was written. There was also a “reset block” which reset any errors in the stepper motor and its control function blocks.

The main program was where the PLC checked which cut-out buttons were activated on the HMI, executed the drive sequence for that cut-out and moved onto the next activated cut-out when it was finished. This was implemented using a ladder sequence program and the steps would be set and reset in sequence. For example, if the button to cut-out at thirty degrees was not pressed it reset that step and enabled the next and so on down the program network till it came to a button that was enabled. This can be seen in picture 28.



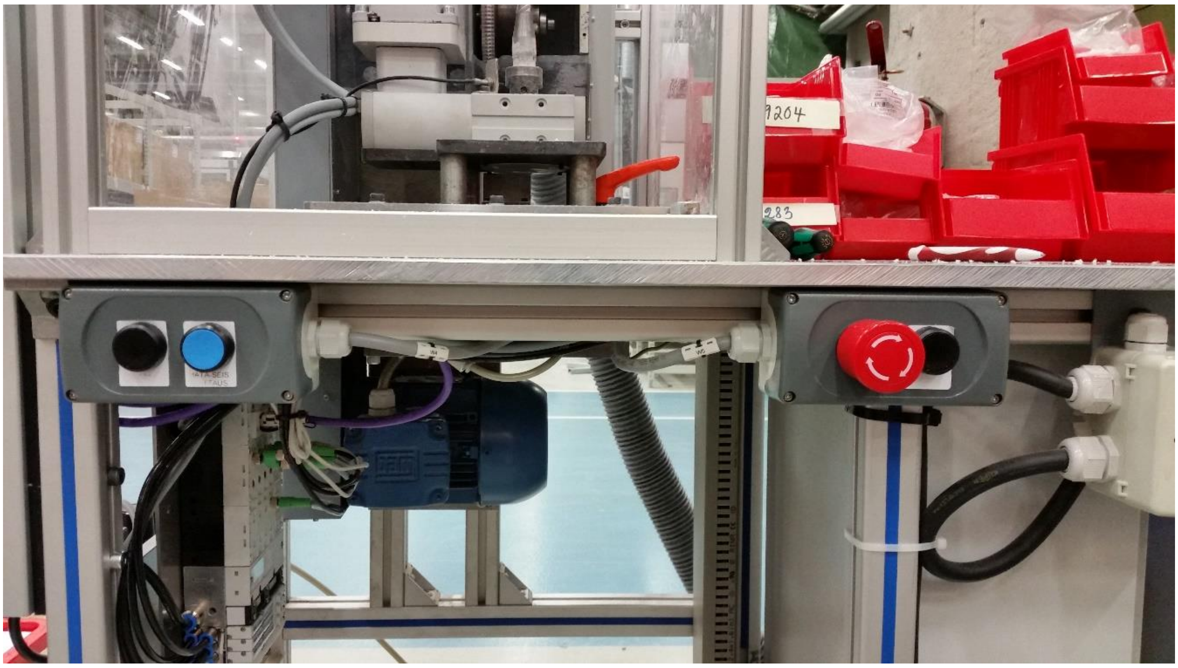

Picture 28. The Main Program (Co-Automation 2015)

When all the activated buttons had been checked and the cut-outs performed, the run signal to the frequency convertor would be turned off and the cutting blade would start to slow-down. An eight second timer was activated and after that the door would open. The same product could then be cut again or new cut-out values could be entered.

The output program was programmed using ladder diagram. Here there were the various different variables and steps that turned on the frequency convertor and the two pneumatic cylinders.

### 4.4.6 Installation

The old machine was a critical part of ABB's production line and they were unwilling to have it shut down for a week. Also the machine couldn't be taken away and built on Co-Automation's premises because of time constraints. It was therefore agreed upon to do the installation and programming starting on Friday morning, work over the weekend (except for Sunday) and have the machine working again on Monday or Tuesday.




Picture 29. The finished machine button controls (Co-Automation 2015)

Picture 30. The finished machine’s HMI and motor (Co-Automation 2015)

On Friday morning the power was turned off and the electrical and mechanical installation work began. This portion of the work was quite straightforward and it was completed by Friday evening. The finished machine can be seen in pictures 29 and 30.

### 4.4.7 Obstacles Encountered

The machine was ready Monday afternoon as had been arranged with ABB. It was at this time it was tested for the first time to see how some of the spindles the machine made, fitted in the rotary cam switches.

The first spindle that was tested didn't fit and one of the workers at the assembly station informed why. It turned out that the cuts could not be cut out starting at 0 degrees from any random point as had been assumed during the whole design and programming and installation process. Rather, if the cut required was at an even number, such as 30 degrees, then the middle of that cut had to be in the middle of where there was a tooth on the inside cog of the spindle. Likewise, if it was an odd number it had to be in the middle of where there was no tooth on the inside of the spindle. This is illustrated by the green mark as a simulated cut-out in picture 11.

This was a major issue because the stepper motor had an incremental encoder, so all the positions to which it had to drive to were calculated from where it stopped last, the last position being its 0 degree point. All cut out positions were absolute positions from this point. Adding to this was the problem that the belted differential was causing the drive positions to lose their accuracy because of mechanical backlash. The motor could not be told go to a certain point via clockwise or anticlockwise directions because of the limited function blocks which were available with this controller, rather it choose for itself the shortest route. All the quick changes in direction caused this mechanical backlash to occur. This wouldn't have been a problem if the cuts could be started from any point. Despite the mechanical backlash the accuracy would've been good enough for it to cut accurately for one or two products.

The machine had to be working right away so a solution had to be found to overcome this problem quickly. If this had been known at the project outset then a motor with an absolute encoder would have been chosen or at least a different version of the

CMMO controller where the direction of turning could be dictated so the motor would always turn in the same direction and therefore suffer no mechanical backlash.

Picture 31. The Homing Sensor (Co-Automation 2015)

What was decided upon was to insert an M5 bolt into a pin hole on the rotation shaft and mount a homing inductive sensor close to this shaft as seen in picture 31 inside the red box. This homing sensor was connected to a dedicated terminal on the CMMO controller. In the motor configuration software the motor was told to home always in one direction. When it came to the bolt it stopped and then moved a little bit to a point in the middle of where no tooth was, this was then the 0 degree point and all other cut-out points could be calculated from there. Homing was then run for every piece. This is unusual for automation applications but was the only solution short of buying a new motor with a different encoder. It was also a good solution because the product cycle time was not lowered in any way because the homing function could be completed when the door shut and the cutting blade was running up to speed.

# 5 Conclusions

As stated in the introduction this thesis had two main goals, one being to test out equipment that Co-Automation had not used before but desired to test in order to ascertain their future usability and suitability with the existing systems which were in use at the company. Secondly it was to complete the updating of the Spindle milling machine. Through this second goal the first goal was accomplished.

## 5.1 The testing of new equipment

When building automation systems, mixing different manufacturers' equipment is generally avoided to prevent communication and compatibility problems. However, in this project it was necessary in order to test the three different systems, these being: Festo PLCs and CPX platforms, Festo's Stepper motors and CMMO controllers and ABB's HMI Panels. All these products are quality products that work well individually and almost seamlessly within their own systems. So rather, this was an individual assessment on behalf of Co-Automation to determine how they would work with their systems, that being Omron PLCs.

### 5.1.1 The Festo PLC and CPX Platform

It was found that the Festo PLC and CPX platform was a good product that could work well with Co-Automation's systems. Omron's NJ PLCs, which use the Ethernet I/P and Ethercat protocols, were able to communicate well with the Festo PLCs, embedded I/Os and pneumatic valve terminals. This was because Festo have a wide range of communication cards available which makes it easy to communicate with their systems. They also have many different I/O and valve options along with motor control cards that can control up to four stepper motors from a small card in the CPX unit. Perhaps their greatest benefit is the ease of configuring a CPX system online using the configuration tools on their website. In a few minutes it is possible to configure a whole automation system, ready with the order code. It will then be delivered completely assembled.

The negative side was that if you wish to program using a ladder diagram then it is not quite as easy and powerful with CoDeSys as it is with Omron's CX-Programmer or Sysmac Studio softwares. Omron appears to design their software around the ladder while CoDeSys appears to be designed around the ST programming language. This is merely a matter of the individual programming preferences of Co-Automation and does not mean one system is inferior to another.

As a result of this trial Co-Automation now sometimes uses Festo PLCs on equipment which has to be able to operate independently but nonetheless needs to be controlled by a higher order PLC, this generally being an Omron PLC. The CPX platform is now used for the most part as the company's field embedded I/O system. This has the convenience of allowing all the field I/Os and valves to be located in one or more packs all connected to a Field-Bus.

### 5.1.2 Festo Stepper Motors

The motors and controllers were found to be good value, flexible and useful. However, care must be taken to choose the correct controller for the application needed as different controllers have different programming features and some are more flexible than others. Different CMMO models also support different communication protocols. These stepper motors and controllers are now widely used by Co-Automation as a result of this study. If a greater lifting force, longer duty cycle or greater holding force is needed then a servo motor system will be used instead.

### 5.1.3 ABB HMI Panels

The ABB HMI was included in this machine because at the time of this thesis a lot of Co-Automation's work came from ABB. Since the HMI is a very visible element of an automation cell or line, the company thought that in the future ABB could insist on Co-Automation using ABB's own HMI brand.

As a result of this study it is unlikely the company will use this HMI on its lines. The ABB HMI uses for its Ethernet based communication the standard Ethernet TCP/IP or Modbus TCP/IP. The Modbus protocol is not used as standard by Omron or ABB Robots which are two companies with which Co-Automation has close co-operation. Communication will work over Ethernet TCP/IP between Omron NJ PLCs and ABB HMIs but it will require opening a socket between them thus making it more time consuming and cumbersome than it would need to be. In general HMIs and PLCs of the same make will naturally work best together thus an ABB HMI will always work best with an ABB PLC.

### 5.2 The Updating of the Spindle Milling Machine

In regards to the updating of the machine, what was set out to be accomplished was accomplished. The machine was updated from a semi-automated machine to a fully automated one. Stipulations about what the machine had to be able to do were given at the beginning of the project, and these were met.

As an example, on the old machine a spindle that required to have many sections cut out could take up to nine minutes to make. Whereas, with the updated machine, it now takes a little over one minute to make the same spindle. ABB are pleased with the result and since the commissioning was completed it has not been necessary to go back and make any adjustments or changes.

# BIBLIOGRAPHY


ABB Rotary Cam Switch. [Online publication]. ABB Oy. [Ref. 14 January 2016]. Available at:https://library.e.abb.com/public/bbdf03de7bf680eac1257d09001db0f4/1SCC302003C0201.pdf

ABB Spindle. [Online publication]. ABB Oy. [Ref. 14 January 2016]. Internal Drawing

ABB Spindle with Simulated Cut-out. [Online publication]. ABB Oy. [Ref. 14 January 2016]. Internal Drawing

A Capacitive Touch Screen. [Online publication]. Baanto International Ltd. [Ref. 07 February 2016]. Available at: http://baanto.com/capacitive-touch-screen

A Resistive Touch Screen. [Online publication]. Baanto International Ltd. [Ref. 11 February 2016]. Available at: http://baanto.com/resistive-touch-screen-technology

Festo. 2016. Modular Electrical Terminal CPX. [Online publication]. Festo AG & Co. KG. [Ref. 22 February 2016]. Available at: https://www.festo.com/cat/fi_fi/data/doc_engb/PDF/EN/CPX_EN.PDF

Festo CMMO-ST Controller. [Online publication]. Festo AG & Co. KG. [Ref. 30 January 2016]. Available at: https://www.festo.com/cat/fi_fi/products_CMMO_ST

Festo EMMS-ST Stepper Motor. [Online publication]. Festo AG & Co. KG. [Ref. 30 January 2016]. Available at: https://www.festo.com/cat/en-gb_gb/data/doc_engb/PDF/EN/EMMS-ST_EN.PDF

Hackworth, J. & Hackworth, F. 2004. Programmable Logic Controllers: Programming methods and applications. New Jersey: Pearson Prentice Hall.

Hughes, A. 2006. Electric Motors and Drives: Fundamentals, Types, and Applications. [E-book]. Ed.: 3rd ed. Amsterdam: Newnes. [Ref. 17 December 2015]. Available at: Needs special user rights.

Omron CP1E Compact PLC. [Online publication]. Omron Corp. [Ref. 17 December 2015]. Available at: https://industrial.omron.fi/fi/products/catalogue/automation_systems/programmable_logic_controllers/compact_plc_series/cp1e/default.html

Omron CS1G/H Rack PLC. [Online publication]. Omron Corp. [Ref. 17 December 2015]. Available at: https://industrial.omron.fi/fi/products/catalogue/automation_systems/programmable_logic_controllers/rack_plc_series/cs1g_h/default.html

Omron G9SP Safety PLC. [Online publication]. Omron Corp. [Ref. 9 January 2016]. Available at: https://industrial.omron.fi/fi/products/catalogue/safety/programmable_safety_system/standalone_safety_controllers/g9sp/default.html

Rockwell Automation. 2002. PLC vs. Safety PLC – Fundamental and Significant Differences. [Online publication]. Rockwell Automation Inc. [Ref. 9 January 2016]. Available at: http://literature.rockwellautomation.com/idc/groups/literature/documents/wp/safety-wp002_-en-e.pdf

Siemens SIMATIC S7-300 Modular PLC. [Online publication]. Siemens AG. [Ref. 17 December 2015]. Available at: https://w3.siemens.com/mcms/programmable-logic-controller/en/advanced-controller/s7-300/Pages/Default.aspx

Texas Instruments. 2014. Human Machine Interface (HMI) Guide. [Online publication]. Texas Instruments Incorporated. [Ref. 07 February 2016]. Available at: http://www.ti.com/lit/sl/slyy049a/slyy049a.pdf