

1502997 - Final

by David Harries

FILE	7309_DAVID_HARRIES_1502997_-_FINAL_177909_116346150.PDF (10.66M)		
TIME SUBMITTED	08-MAY-2020 10:46PM (UTC+0100)	WORD COUNT	7177
SUBMISSION ID	126805534	CHARACTER COUNT	44414

Course: BSc (Hons) Electrical and Electronic Engineering
Module: SGEE6003 Major Project
Assignment: 2. Project Design, Implementation and Evaluation
Name: Sam Harries
University Number: 1502997
Date Submitted: 8th May 2020
Word Count: 4626

Table of Contents

1	- Executive Summary.....	6
2	- Introduction	7
3	- Design and Implementation.....	8
3.1	- Replacing Existing Hardware	9
3.2	- Integrating into Existing Network and Modifying Program.....	11
3.3	- Ladder Program	15
3.4	- Configuration and Commissioning of V90 Servo Drive	28
3.5	- V90 TIA Program	34
4	- Testing and Results.....	44
4.1	- Replacing Existing Hardware	44
4.2	- V90 Tests	48
5	- Evaluation and the Wider Issues	50
6	- Conclusions and Recommendations.....	52
7	- Citation List.....	54
8	- Bibliography	55
9	- Appendix	56

Table of Figures

Figure 3.1 - Work Breakdown Structure.....	8
Figure 3.2 - Keyence PLC I/O Diagram	9
Figure 3.3 - Keyence PLC	10
Figure 3.4 - Teach Box	10
Figure 3.5 - Siemens PLC I/O Diagram.....	11
Figure 3.6 - PLC 106 IP Configuration	12
Figure 3.7 - Linear Drive Block Diagram.....	12
Figure 3.8 - I/O Tags in PLC Hardware Configuration.....	13
Figure 3.9 - HMI IP Configuration.....	13
Figure 3.10 - Linear Drive HMI Screen.....	14
Figure 3.11 - Control Data Block	14
Figure 3.12 - Position Data Block.....	14
Figure 3.13 - Teach Box Data Block	15
Figure 3.14 - Move Left OUTPUT Ladder Logic	15
Figure 3.15 – Initialised	15
Figure 3.16 - HSC Decrementing Moving LEFT	16
Figure 3.17 - HSC Incrementing Moving RIGHT	16
Figure 3.18 - Dint to Real Conversion.....	17
Figure 3.19 - Encoder Relative Count	17
Figure 3.20 - Scaled Distance Calculation.....	17
Figure 3.21 – Move Right OUTPUT Ladder Logic	18
Figure 3.22 - Load Mitsubishi Program Function Call.....	18
Figure 3.23 - RS232 Message Handler	18
Figure 3.24 - Function Call Blocks.....	18
Figure 3.25 - In Position Signal	19
Figure 3.26 - Initial Position Values	20
Figure 3.27 - Manual Distance Setpoint	21
Figure 3.28 – Automatic Distance Setpoint.....	22
Figure 3.29 - Manual Position Teaching.....	22
Figure 3.30 - Distance Error Calculation	23
Figure 3.31 - Direction Decision Logic	23
Figure 3.32 - Run Command Control Logic.....	24

Figure 3.33 - Speed Control	24
Figure 3.34 - Speed Control Logic for RSN	25
Figure 3.35 - GET and PUT Function Block	25
Figure 3.36 - GET and PUT Function Block and Data Block	26
Figure 3.37 - PLC 106 DB in Master PLC 101	26
Figure 3.38 - Encoder Scaled Distance MOVE to PLC 101	27
Figure 3.39 - Scaled Distance Converted to DWORD	27
Figure 3.40 - Horizontal Animation	27
Figure 3.41 - V90 Single Phase Connection Diagram (Siemens, 2017)	28
Figure 3.42 - V90 Connection Block Diagram (SIEMENS, 2016)	29
Figure 3.43 - Physical System	29
Figure 3.44 - Network Switch	30
Figure 3.45 - Drive and Motor Selection in V-Assistant	31
Figure 3.46 - PROFINET Setup	31
Figure 3.47 - Ramp Function Configuration	32
Figure 3.48 - Torque and Speed Limits	33
Figure 3.49 - 3000rpm CW Test	33
Figure 3.50 - 3000rpm CCW Test	34
Figure 3.51 - 5000rpm Test	34
Figure 3.52 - 3-Bit Logic	35
Figure 3.53 - Basic Parameters (General)	35
Figure 3.54 - Network View	36
Figure 3.55 - Telegram Selection	36
Figure 3.56 - Drive Parameters in TIA	37
Figure 3.57 - Encoder Parameters 1	37
Figure 3.58 - Encoder Parameters 2	38
Figure 3.59 – Mechanics	38
Figure 3.60 - Position Monitoring	39
Figure 3.61 - MC_Power	39
Figure 3.62 - MC_Reset	40
Figure 3.63 - V90 Position 1	41
Figure 3.64 - V90 Position 2	41
Figure 3.65 - V90 Position 3	41
Figure 3.66 - V90 Position 4	42

Figure 3.67 - V90 Position 5.....	42
Figure 3.68 - Test HMI	43
Figure 4.1 – Mimic Home Position (0.00mm).....	45
Figure 4.2 - Mimic Position 1 (1000.00mm)	45
Figure 4.3 - Mimic Position 2 (300.00mm)	46
Figure 4.4 - Mimic Position 3 (500.00mm)	46
Figure 4.5 - Mimic Position 4 (1200.00mm)	47
Figure 4.6 - Mimic Position 5 (0.00mm)	47
Figure 6.1 - Odrive Board.....	52
Figure 9.1 - Appendix 1 – Parameters	56
Figure 9.2 - Appendix 2 - Parameters.....	56
Figure 9.3 - Appendix 3 – Parameters	57
Figure 9.4 - Appendix 4 - Parameters.....	57
Figure 9.5 - Appendix 5 – Parameters	58
Figure 9.6 - Appendix 6 – Parameters	58
Figure 9.7 - Appendix 7 – Parameters	58
Figure 9.8 - Appendix 8 - Mitsi Program 1	59
Figure 9.9 - Appendix 9 - Mitsi Program 2	59
Figure 9.10 - Appendix 9 - Mitsi Program 3	60
Figure 9.11 - Appendix 10 - Mitsi Program 4	60
Figure 9.12 - Appendix 11 - Mitsi Program 5	61
Figure 9.13 - Appendix 12 - Mitsi Program 6	61
Figure 9.14 - Appendix 13 - Mitsi Program 7	62
Figure 9.15 - Appendix 14 - RS232 Program 1	62
Figure 9.16 - Appendix 15 - RS232 Program 2.....	63
Figure 9.17 - Appendix 16 - RS232 Program 3.....	63
Figure 9.18 - Appendix 17 - RS232 Program 4.....	64
Figure 9.19 - Appendix 18 - RS232 Program 5.....	64
Figure 9.20 - Appendix 19 - RS232 Program 6.....	65
Figure 9.21 - Appendix 20 - RS232 Program 7.....	65
Figure 9.22 - Appendix 21 - RS232 Program 8.....	66
Figure 9.23 - Appendix 22 - RS232 Program 9.....	66
Figure 9.24 - Appendix 23 - RS232 Program 9.....	67
Figure 9.25 - Appendix 24 - RS232 Program 10.....	67

Figure 9.26 - Appendix 25 - RS232 Program 11	68
Figure 9.27 - Appendix 26 - RS232 Program 12	68
Figure 9.28 - Appendix 27 - GET and PUT Program 1	69
Figure 9.29 - Appendix 28 - GET and PUT Program 2	69
Figure 9.30 - Appendix 29 - GET and PUT Program 3	70
Figure 9.31 - Appendix 30 - GET and PUT Program 4	71
Figure 9.32 - Appendix 31 - GET and PUT Program 5	71
Figure 9.33 - Appendix 32 - GET and PUT Program 6	72
Figure 9.34 - Appendix 33 - GET and PUT Program 7	72
Figure 9.35 - Appendix 34 - GET and PUT Program 8	73

*You are summarising the report rather than
the project.*

1 - Executive Summary

This report is a continuation of assignment 1 in which a problem with the Finishing Robot on the CIM system required a solution. The problem was that the linear drive was outdated and needed bringing up to Industry 4.0 standards while also improving its mechanical performance.

To do this, the SIMATIC V90 Servo Drive by Siemens was selected as the most optimal solution. This report details the design, implementation and testing of the V90 to the CIM system. However, due to the national lockdown, the implementation and testing aspects of the project were cut short.

The design process includes replacing the outdated Keyence PLC with a Siemens S7-1212 AC/DC/RLY PLC, modifying the program to have the Teach Box functions available on a HMI, and to allow for manual and automatic control of the linear drive. The V90 is then configured and commissioned. All these are then tested in preparation for the integration to the CIM system.

The conclusion to the project is that most of the design work is completed and the V90 drive is ready to be integrated to the CIM system and the existing PROFINET network. However, if the project had gone as planned then the budget would have been exceeded with the additional cost of panel components.

*The panel already existed so no
further cost was needed.*

2 - Introduction

You should be introducing the project not the report. Aims and Objectives?

The contents of this report will detail the design, implementation and testing of the improved linear drive system used to move the Finishing Robot within the CIM system in G8. Within the previous report, a comparison of three solutions was done and the most suitable solution, a SIMATIC V90 Servo Drive, will be used to fulfil the project's design specifications.

The desired outcome of the project is a fully functioning Linear Drive System which meets the FDS requirements, meets "Industry 4.0" standards, and provides a valuable educational tool to be used to teach Servo Motor and PLC modules at the college.

However, due to the COVID-19 global pandemic, the implementation of the V90 Servo Drive and Motor to the CIM system has been made impossible. Therefore, the implementation and testing of the system has been inhibited. As a compromise, the design and commissioning of the system will be tested as if it is theoretically part of the CIM system.

You did not need to include this last paragraph in the introduction

3 - Design and Implementation

The work breakdown structure (WBS) created in assignment 1 will utilized to ensure the design process is done correctly. The WBS can be seen in *Figure 3.1*.

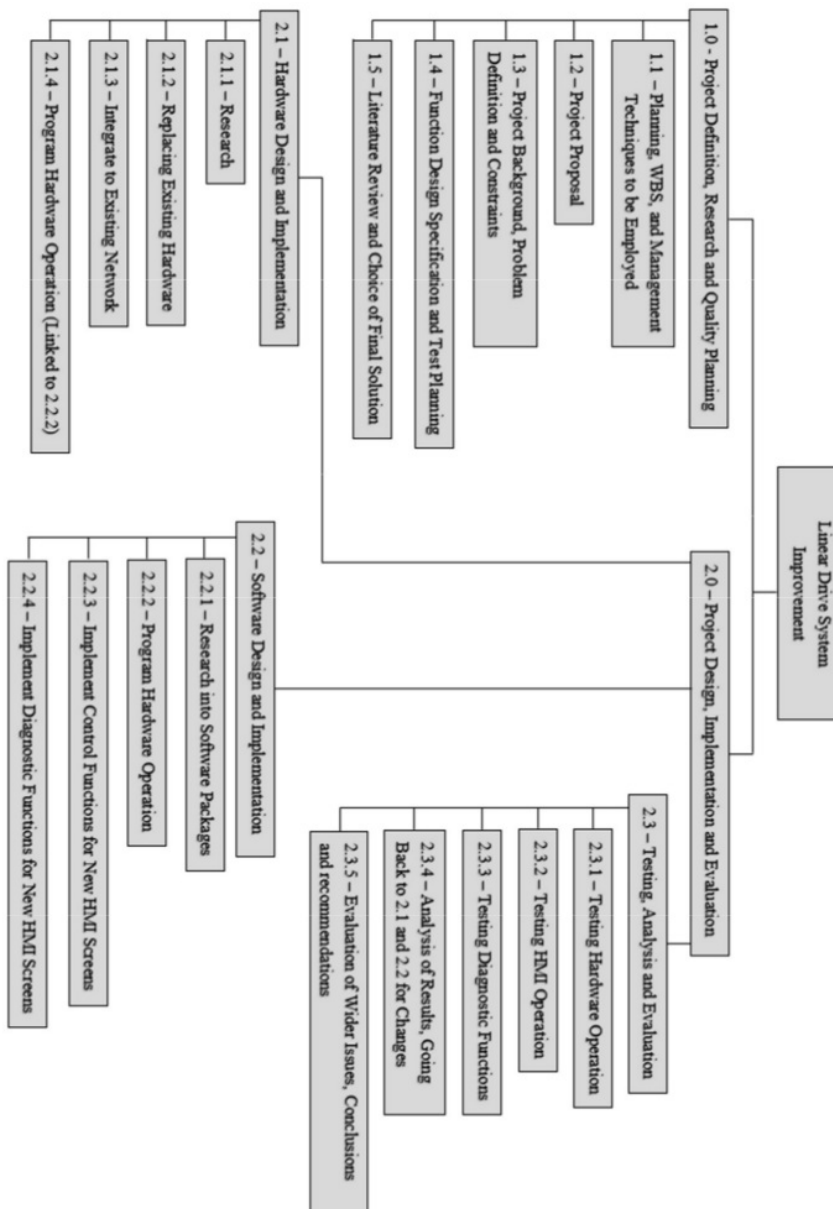


Figure 3.1 - Work Breakdown Structure

3.1 - Replacing Existing Hardware

The first stage of the design process was to replace the existing Keyence PLC with a Siemens S7-1200 AC/DC/RLY PLC. The inputs and outputs of the Keyence PLC were determined by manually controlling the Finishing Station to see each I/O's function, which were then recorded using a diagram. This can be seen in **Figure 3.2**.

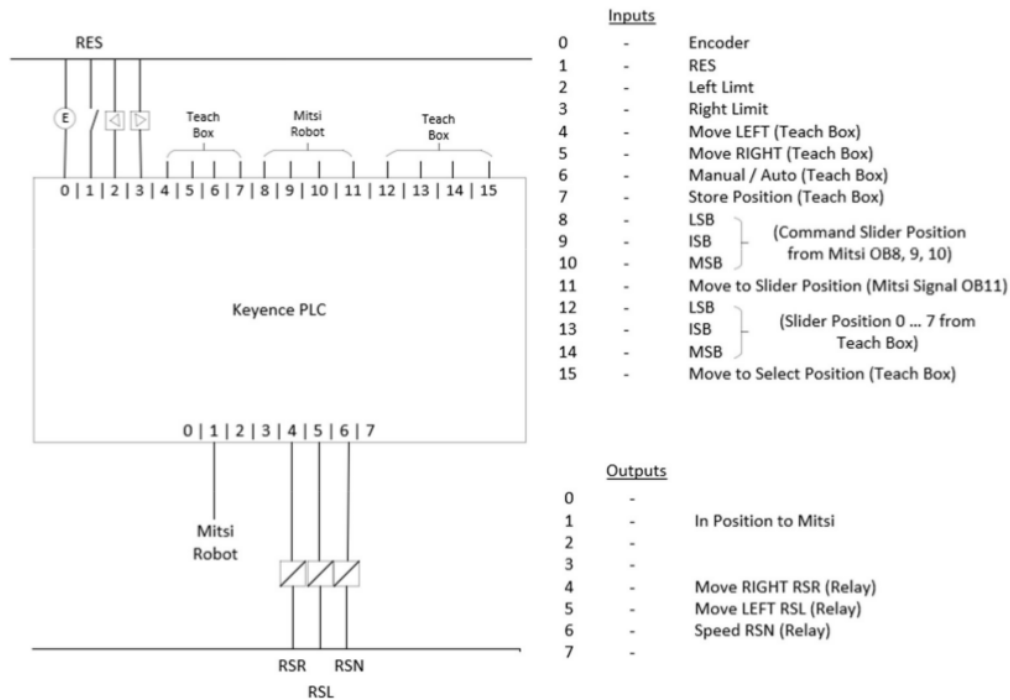


Figure 3.2 - Keyence PLC I/O Diagram

The Teach Box was used to manually control the position of the Finishing Robot across the actuator, store up to 8 positions, and switch between auto and manual modes. These functions are easily transferred to an HMI screen making the Teach Box redundant, while also reducing the required amount of inputs by 8. This was an important design decision as the Siemens S7-1200 only has 8 inputs, whereas the Keyence uses all 15 to control the system. Therefore, the number of required inputs must be reduced. **Figure 3.3** shows the Keyence PLC and **Figure 3.4** shows the Teach Box.



Figure 3.3 - Keyence PLC



Figure 3.4 - Teach Box

By reducing the number of inputs required, a diagram was then made detailing the I/Os of the Siemens PLC.

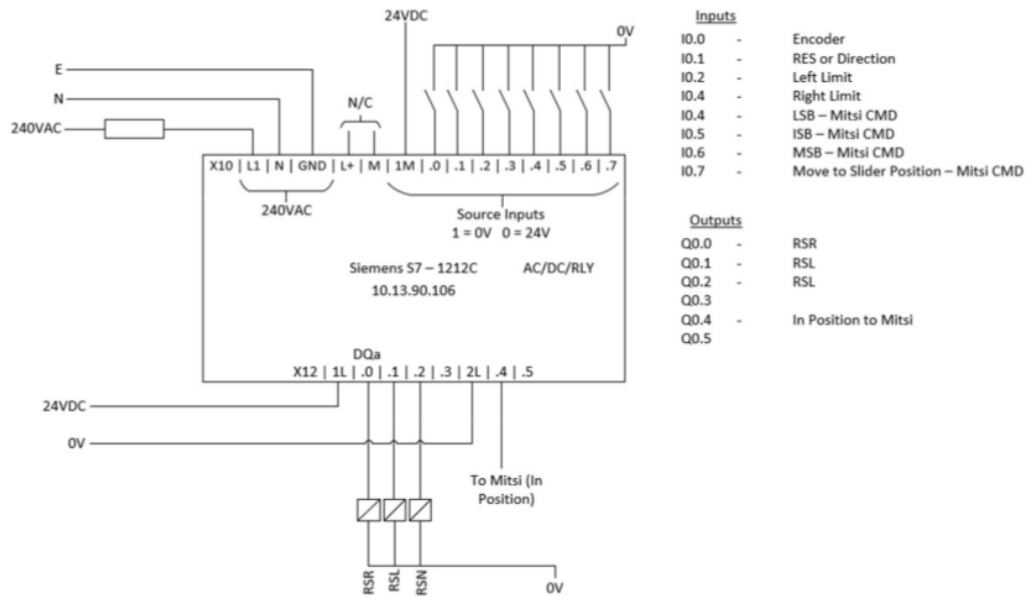


Figure 3.5 - Siemens PLC I/O Diagram

Due to the lockdown, a photo of the Siemens PLC taking the Keyence's place is unavailable. After switching the PLCs, the program had to be modified to allow the Siemens PLC to control the Linear Drive, and for the Teach Box's functions to be implemented to a HMI. *Photo could have been taken back in January at the time of build, should be in the log book.*

3.2 - Integrating into Existing Network and Modifying Program

The Mitsubishi robot (Mitsi) controls its linear position by sending signals to its controller. These signals, as shown in **Figure 3.5**, are a 'move to slider position' command, and three bits that combine to make a 3-bit binary number. These bits represent the position number that the Mitsi should move to. As it sends out a 3-bit number, a total of 8 positions are available (0 to 7). The Mitsi uses the 'move to slider position' command to tell the controller when it needs to move to the next position.

Furthermore, the teach box inputs; manual move LEFT, manual move RIGHT, manual / auto mode switch, store position, LSB, ISB and MSB had to be integrated into the program.

The LSB, ISB and MSB bit inputs are used in conjunction with the store position command to tell the PLC to store the encoder reading for positions 0 to 7.

Knowing this information, the Teach Box can be implemented to the PLC program, and changes can be made to the existing controls. Firstly, the PLC must be added to the master TIA project with the IP address: 10.13.90.106.

You should be writing about what was done rather than instructing your reader in what they should do. "has been added" is better than "must be added"

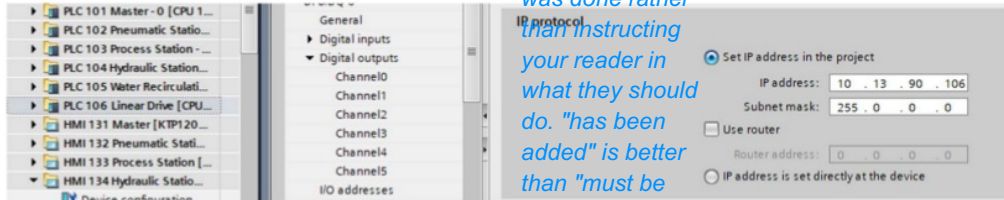


Figure 3.6 - PLC 106 IP Configuration

Using the block diagram shown in **Figure 3.7**, the PLC is integrated to the existing network via a network switch using PROFINET Industrial Ethernet (PN/IE). This allows the PLC to communicate with all the other devices on the CIM system network.

There are other discrete signals between these two.

RS232 comms is in the other direction

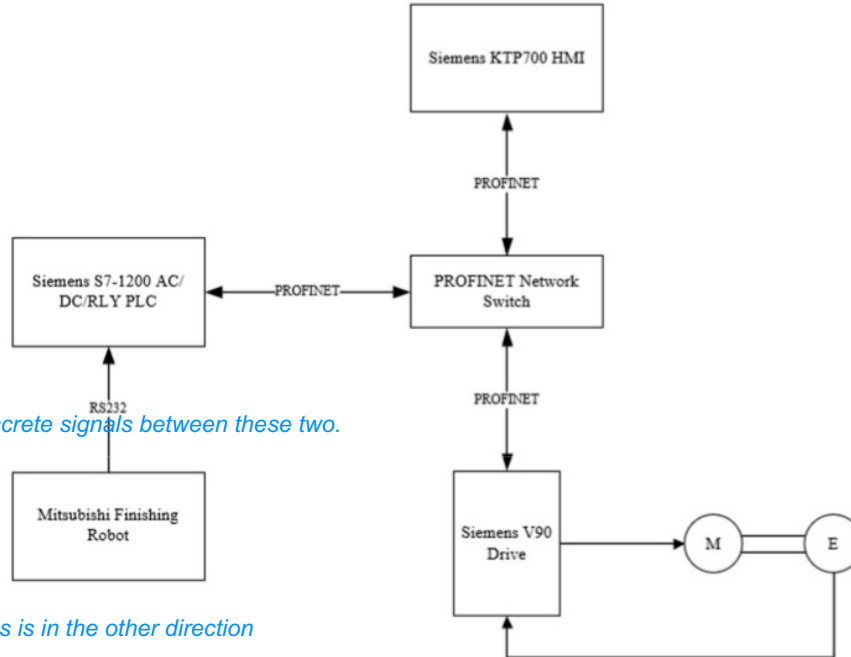


Figure 3.7 - Linear Drive Block Diagram

Next, the input and output tags were assigned in the PLC's hardware configuration settings.

General	IO tags	System constants	Texts
Name	Type	Address	Tag table
Encoder	Bool	%I0.0	Default tag table
RES	Bool	%I0.1	Default tag table
Left_Limit	Bool	%I0.2	Default tag table
Right_Limit	Bool	%I0.3	Default tag table
CMD_Slider_Po...	Bool	%I0.4	Default tag table
CMD_Slider_Po...	Bool	%I0.5	Default tag table
CMD_Slider_Po...	Bool	%I0.6	Default tag table
Move_To_Slider...	Bool	%I0.7	Default tag table
Move_Right_RSR	Bool	%Q0.0	Default tag table
Move_Left_RSL	Bool	%Q0.1	Default tag table
Speed_RSN	Bool	%Q0.2	Default tag table
	Bool	%Q0.3	
In_Position to ...	Bool	%Q0.4	Default tag table
	Bool	%Q0.5	

Figure 3.8 - I/O Tags in PLC Hardware Configuration

The HMI that is used is the Hydraulic Station HMI with an IP address of: 10.13.90.134.

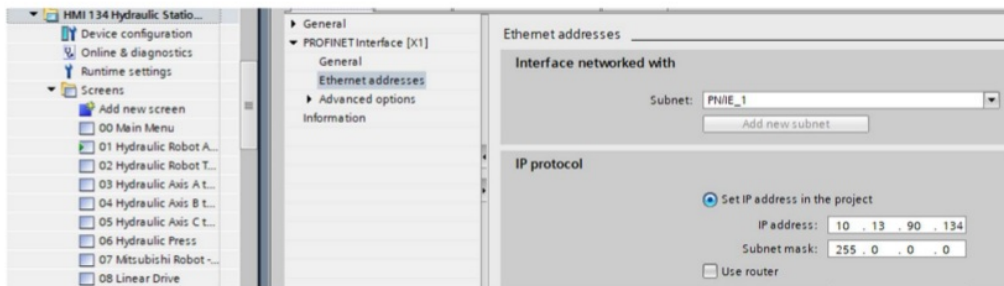


Figure 3.9 - HMI IP Configuration

Figure 3.9 shows the HMI screen that is used to control the Linear Drive system. It contains the functions that were originally from the Teach Box, diagnostic indicators for the PLC I/Os, and fields that display data such as encoder distance etc. This screen can be used to manually control the position of the robot, teach the PLC positions and start the automatic process.

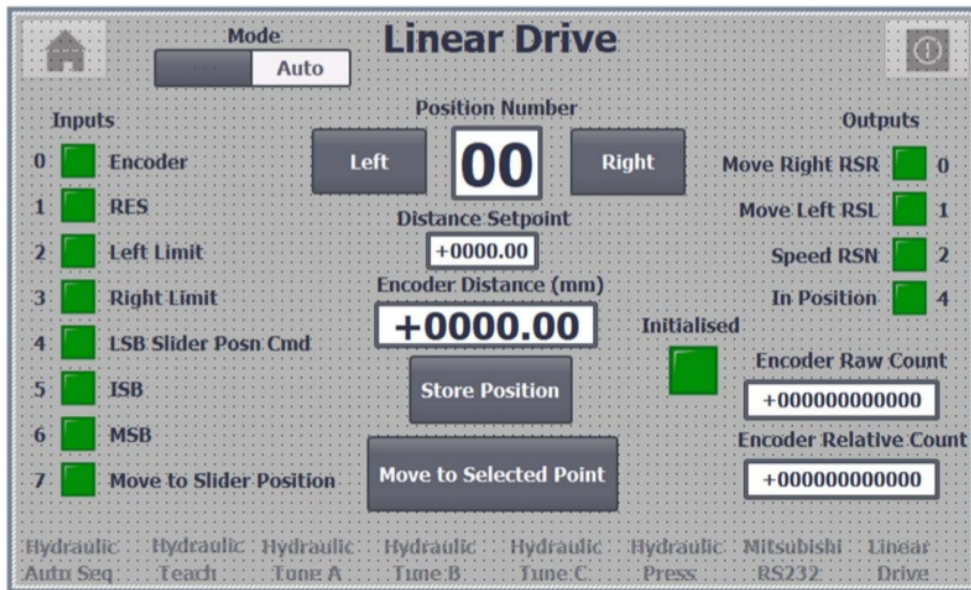


Figure 3.10 - Linear Drive HMI Screen

For this HMI to function, the main Organization Block required new Ladder to be written. Additionally, 'Control', 'Teach Box' and 'Position' data blocks were created to store control variables.

Control									
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static								
2	Distance Error	Real	0.0		☒	☒	☒		
3	Distance Slowdown	Real	0.0		☒	☒	☒		
4	Distance Deadband	Real	0.0		☒	☒	☒		
5	Distance Setpoint	Real	0.0		☒	☒	☒		
6	Move Left	Bool	false		☒	☒	☒		
7	Move Right	Bool	false		☒	☒	☒		
8	Absolute Distance Error	Real	0.0		☒	☒	☒		
9	Run	Bool	false		☒	☒	☒		
10	High Speed	Bool	false		☒	☒	☒		

Figure 3.11 - Control Data Block

Position									
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static								
2	0	Real	0.0		☒	☒	☒		
3	1	Real	0.0		☒	☒	☒		
4	2	Real	0.0		☒	☒	☒		
5	3	Real	0.0		☒	☒	☒		
6	4	Real	0.0		☒	☒	☒		
7	5	Real	0.0		☒	☒	☒		
8	6	Real	0.0		☒	☒	☒		
9	7	Real	0.0		☒	☒	☒		

Figure 3.12 - Position Data Block

Teach Box									
	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static								
2	Teach Position	UInt	0		☒	☒	☒	☐	
3	Move Left	Bool	false		☒	☒	☒	☐	
4	Move Right	Bool	false		☒	☒	☒	☐	
5	Move to Selected Point	Bool	false		☒	☒	☒	☐	
6	Manual	Bool	false		☒	☒	☒	☐	
7	Store Position	Bool	false		☒	☒	☒	☐	

Figure 3.13 - Teach Box Data Block

3.3 - Ladder Program

The first network controls the leftward movement of the robotic arm. The conditions in which the robot moves left are:

- In manual mode and the move left button on the HMI is pressed,
- In automatic mode and the left command is issued,
- In automatic mode and the arm has not initialised yet.

All these conditions are stopped by the robot reaching the left HW limit switch.

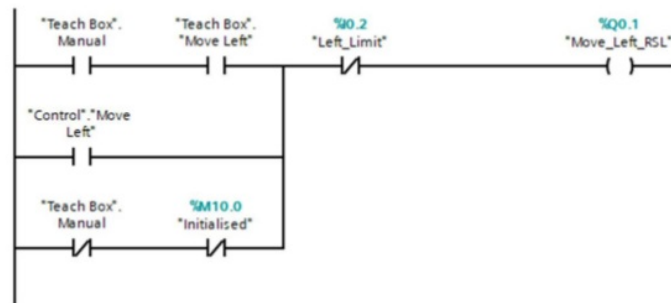


Figure 3.14 - Move Left OUTPUT Ladder Logic

The initialised bit is only set once the left HW limit switch is reached. Therefore, as soon as auto mode is activated, the robot should move left to its 'initialised' position ready to receive commands to move to other positions.

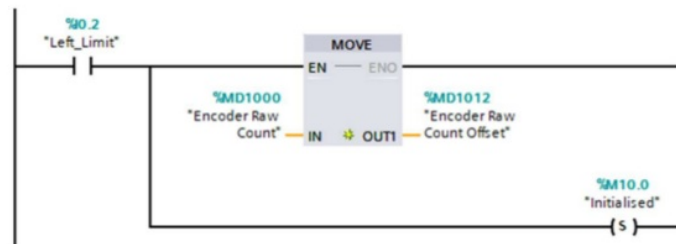


Figure 3.15 – Initialised

Additionally, the 'Encoder Raw Count' value is moved to 'Encoder Raw Count Offset'. When the left HW limit switch is reached, the raw count value should be 0. The raw encoder value is obtained by a High-Speed Counter (HSC) incrementing or decrementing depending on if the motor is moving right or left respectively. The directional outputs %Q0.1 and %Q0.0 enable the direction on each HSC, and the value on 'NEW_DIR' is added to the existing value.

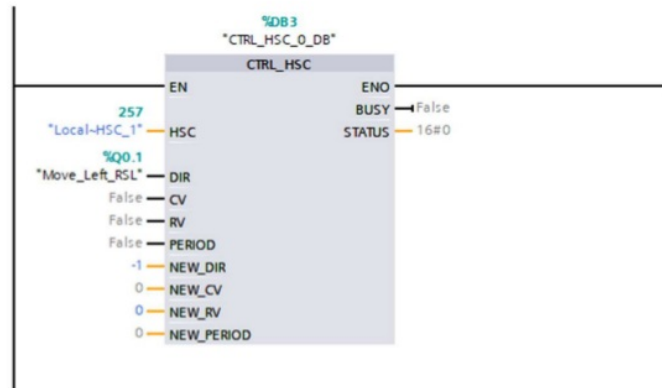


Figure 3.16 - HSC Decrementing Moving LEFT

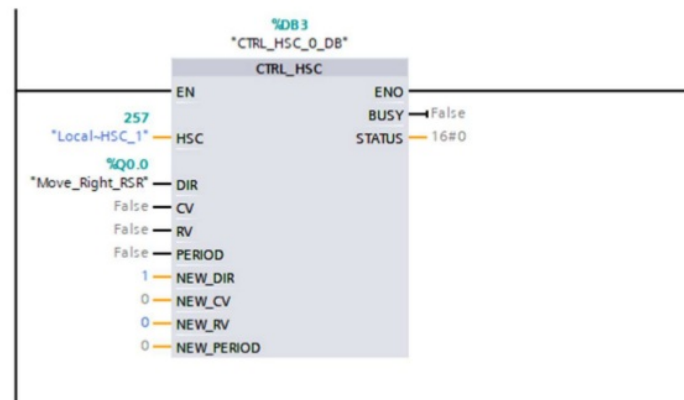


Figure 3.17 - HSC Incrementing Moving RIGHT

The HSC output value is first converted from Dint to Real. Then by moving the raw count offset value, a relative encoder count can be calculated. Without the conversion, these calculations would not be possible.

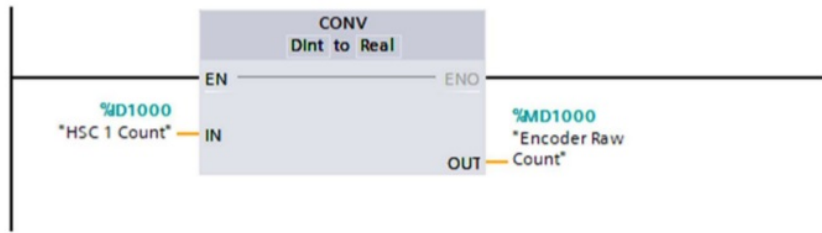


Figure 3.18 - Dint to Real Conversion

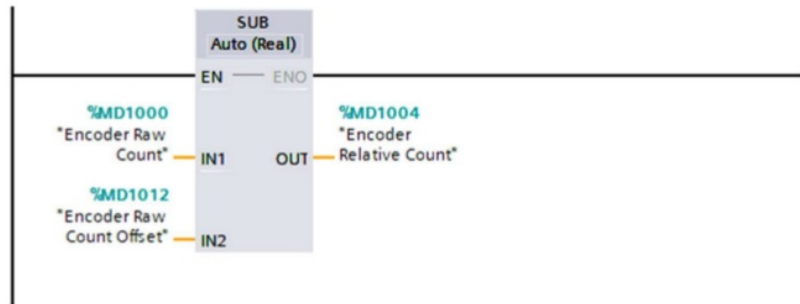


Figure 3.19 - Encoder Relative Count

Therefore, a scaled distance value can be calculated using:

$$\frac{\text{encoder relative count} * \text{actuator leng (mm)}}{\text{maximum encoder count}}$$

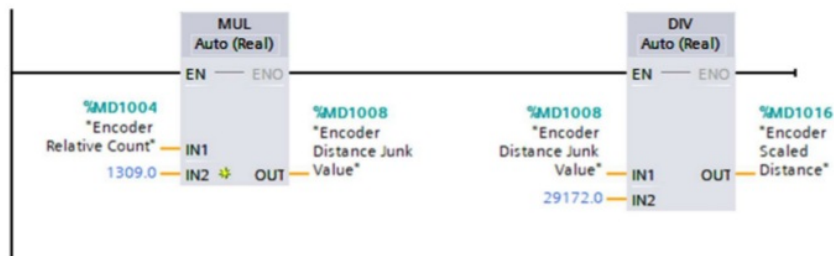


Figure 3.20 - Scaled Distance Calculation

The network controlling the rightward movements contains the conditions:

- In manual mode and the move right HMI button is pressed,
- In auto mode and the move right command is issued,

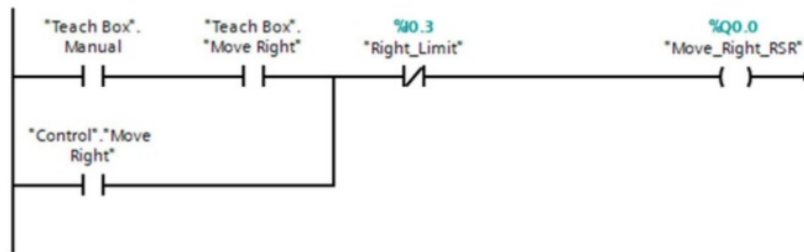


Figure 3.21 – Move Right OUTPUT Ladder Logic

Just like the ‘left’ network, once the right HW limit switch is reached, the output becomes inactive. For the Mitsi Robot to communicate with the PLC, two Function Calls are made.

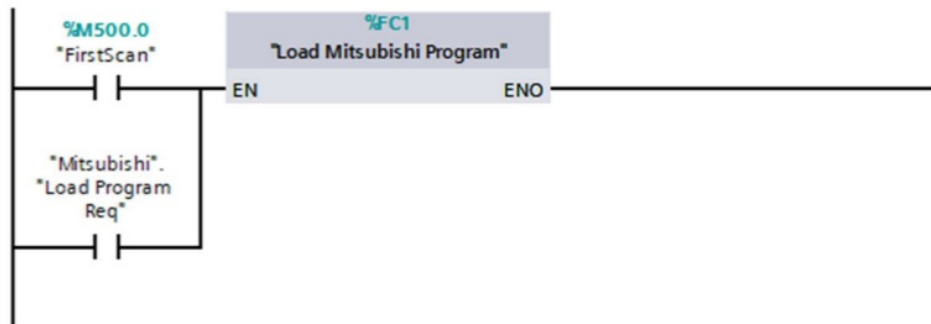


Figure 3.22 - Load Mitsubishi Program Function Call

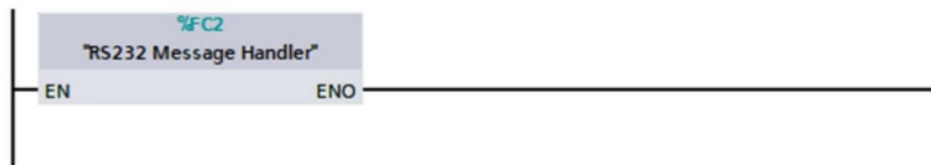


Figure 3.23 - RS232 Message Handler

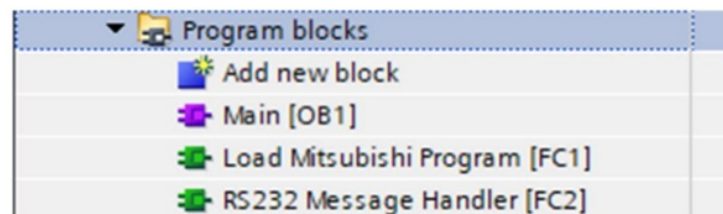


Figure 3.24 - Function Call Blocks

The code within these two Function calls can be found in the appendix. The Mitsi program is sent via RS232 when the FirstScan pulse is generated by the PLC on startup, or when a load program request is sent. The contents of the Mitsi program control the movement of the robotic arm, and the positioning of the Linear Drive System. The most important parts of the program are the lines of code that send the 3-bit position number, and the move to slider position requests. Once a request and position number has been sent, the Mitsi robot will wait for an 'In Position' signal from the PLC before it continues executing the rest of its program.

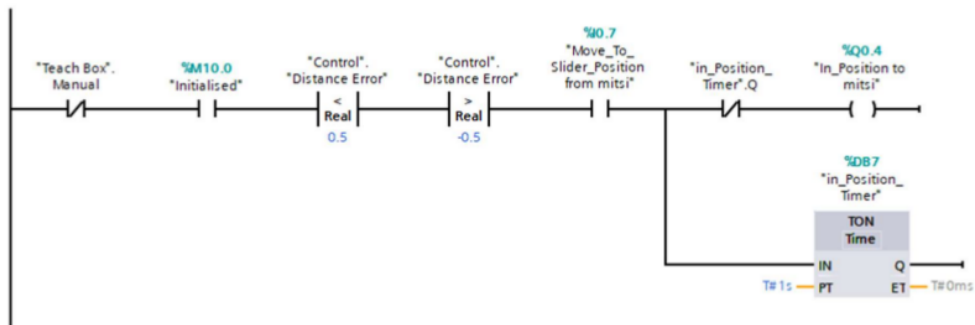


Figure 3.25 - In Position Signal

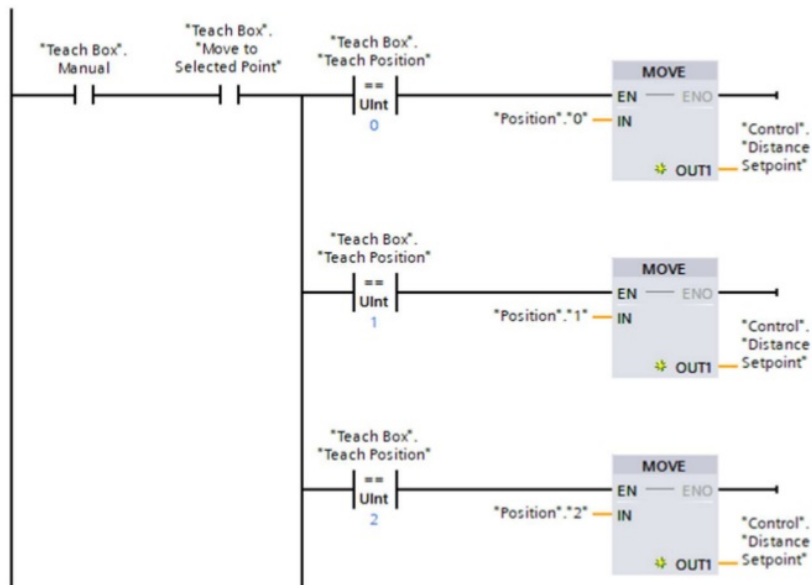
The PLC will send the 'In_Position' signal to the Mitsi when auto mode is active, the robot has been initialised, the distance error is within 0.5mm of the set point and 'Move_To_Slider_Position' signal has been sent from the Mitsi. If all these are true, a 1 second timer starts. This is to give the Linear Drive time to stop before it is given another position to move to. After 1 second, the 'In_Position' signal is sent allowing the Mitsu program to continue.

The automatic sequence dictated by the Mitsi program requires 5 predetermined positions, excluding the home / initialised position. On #initial_Call, 5 values are moved into the 'Position' data blocks 1-5. This allows the automatic sequence to be executed on startup without having to manually set positions each time.



Figure 3.26 - Initial Position Values

The distance setpoint is set in similar ways for both automatic and manual modes. Both use MOVE functions to move values stored in each position's respective 'Position' data block into the 'Control.'Distance_Setpoint' data block.



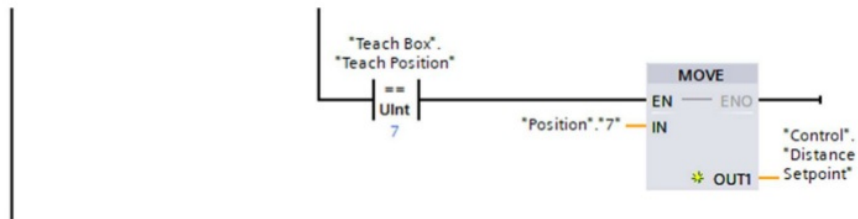
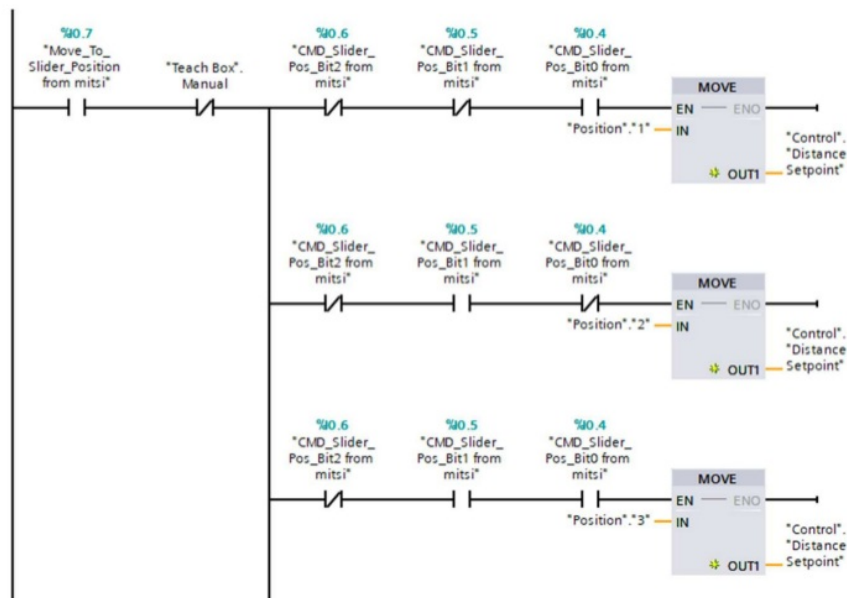


Figure 3.27 - Manual Distance Setpoint

The manual network sets the setpoint based on the position number selected on the HMI, so when the 'Move to Selected Point' button is pressed, the robot moves to the correct position.

The automatic network sets the setpoint based on the 3-bit number received from the Mitsi Robot when the 'Move to Slider Position' signal is received.



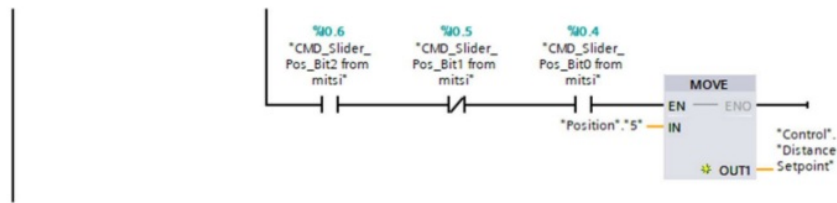


Figure 3.28 – Automatic Distance Setpoint

If the user wishes to change any of the positions along the actuator, they can by selecting a position number on the HMI, and press ‘Store Position’ when the robot is at the desired location.

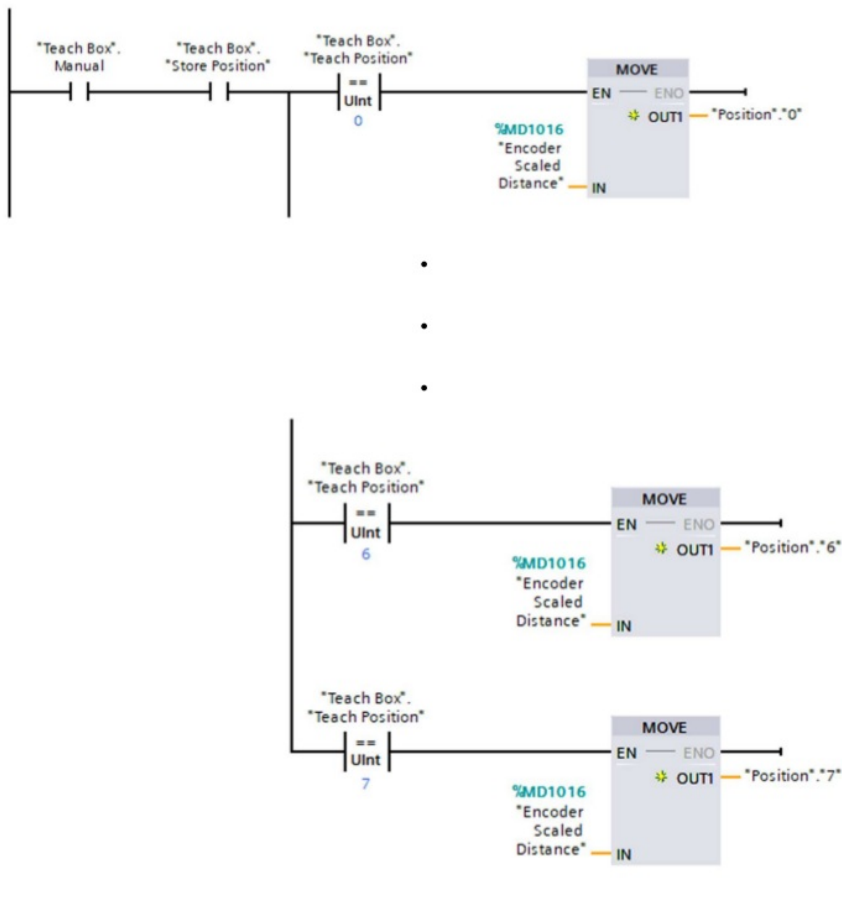


Figure 3.29 - Manual Position Teaching

With the setpoint distances stored, the program needed a way to determine how far away the robot was from the setpoint. This “error” is calculated by:

$$\text{"Control"."Distance Setpoint"} - \text{"Encoder Scaled Distance"} = \text{"Control"."Distance Error"}$$

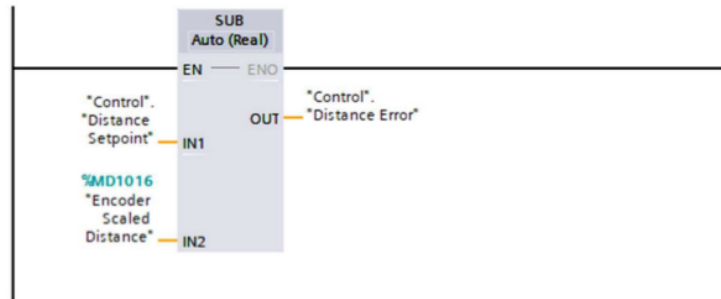


Figure 3.30 - Distance Error Calculation

With the error calculated, the program can then determine which directional command to issue. For example, if the robot were positioned to the right of the setpoint, the robot would have to move left to reach it. Assuming the robot is positioned at 1000mm, and the set point is set to 500mm, the error would be -500. If the situation were changed and the robot was positioned to the left, it would need to move right, resulting in a positive error instead. The following network uses this logic to determine which way the robot needs to move in order to reach the setpoint. A band of 1mm around the setpoint is created to ensure that the logic can determine the correct direction, and not send the robot in the opposite direction once it reaches the setpoint.

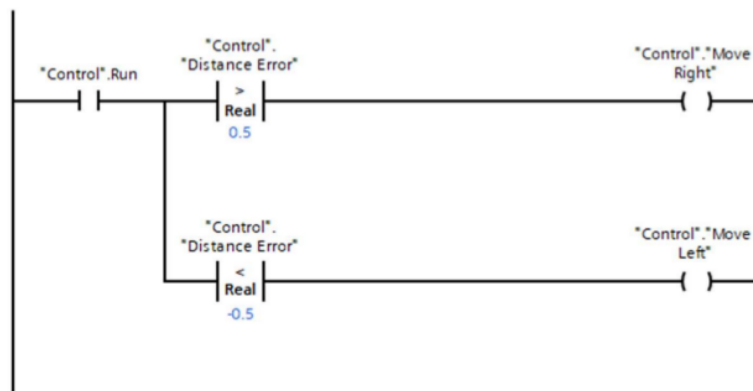


Figure 3.31 - Direction Decision Logic

The “Control”.Run command bit is set when the PLC receives the ‘Move to Selected Point’ command from the HMI while in manual mode, or when the Mitsi robot issues the ‘Move to Slider Position’ command while in automatic mode. The command bit is reset when the error is calculated to be within the 1mm acceptance band.

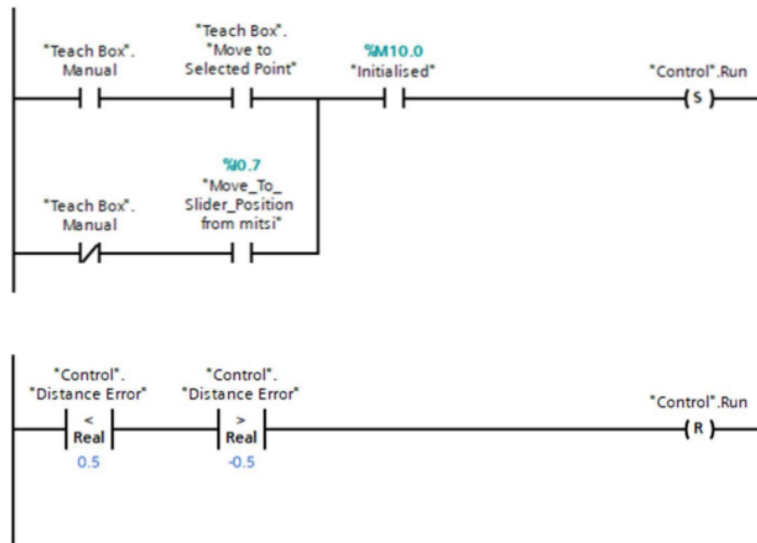


Figure 3.32 - Run Command Control Logic

Additionally, the system has two speed modes, meaning that when the error is greater than a variable amount, the robot should move in high speed mode, then slow down once this value is reached before reaching the setpoint. The same logic can be utilized here.

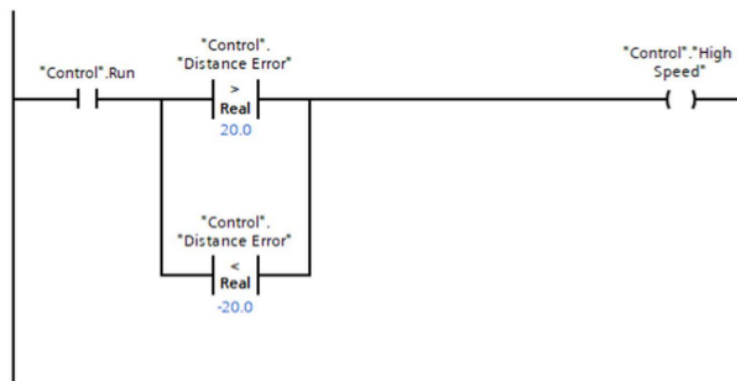


Figure 3.33 - Speed Control

This speed control command is used to control the PLC output which is connected to resistor RSN. Additionally, when the robot is told to move left or right manually using the teach box, and during the initialisation sequence, the robot also moves in high speed mode. When each of these conditions are met, a 1s timer is started. This acts as a 'ramp up' for the motor's speed to stop too much current being drawn due to the load which could damage the motor.

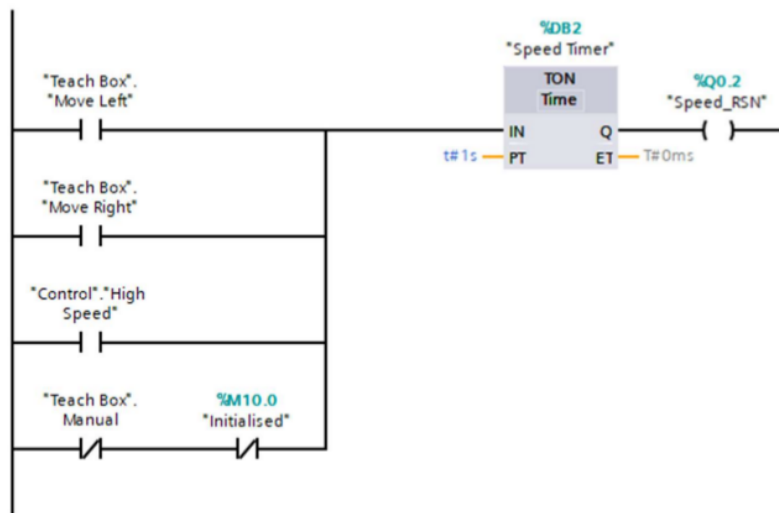


Figure 3.34 - Speed Control Logic for RSN

Finally, a GET and PUT function block is called so that data can be sent to the master PLC 101 to be used in the program, and on the HMI. The ladder for this code can be found in the appendix.

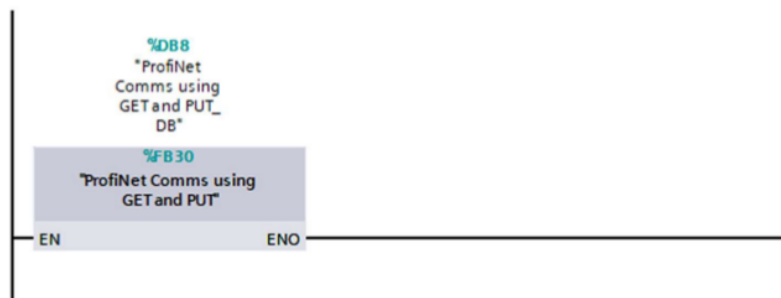


Figure 3.35 - GET and PUT Function Block



Figure 3.36 - GET and PUT Function Block and Data Block

This FB operates under the principle that the data stored within the GET and PUT DB is transferred to a DB on the master PLC.

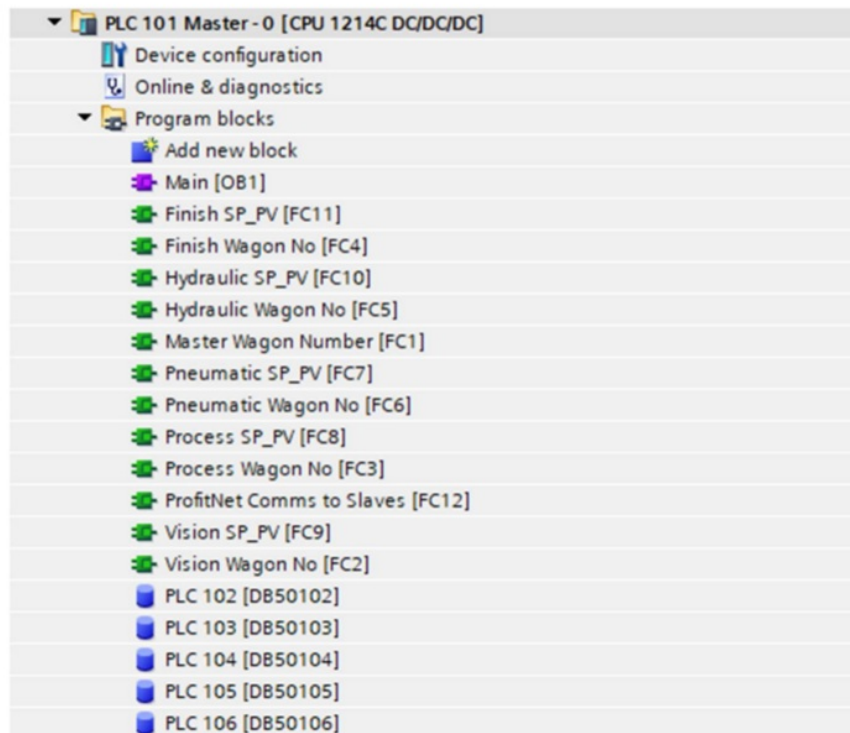


Figure 3.37 - PLC 106 DB in Master PLC 101

The data that is required to be sent to the Master PLC is the Encoder Scale Distance. This is used to create the mimic on the HMI screen which mirrors the robot's position along the actuator. This data is moved into the "PLC 101"."Put Real 1" data block.

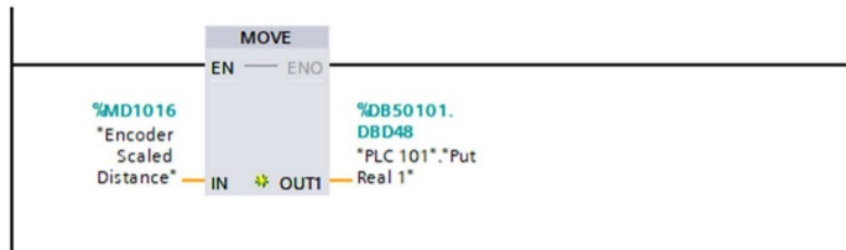


Figure 3.38 - Encoder Scaled Distance MOVE to PLC 101

The mimic is then created by using the animation function in the HMI. To do this the encoder scaled distance must be converted to a DWORD datatype.

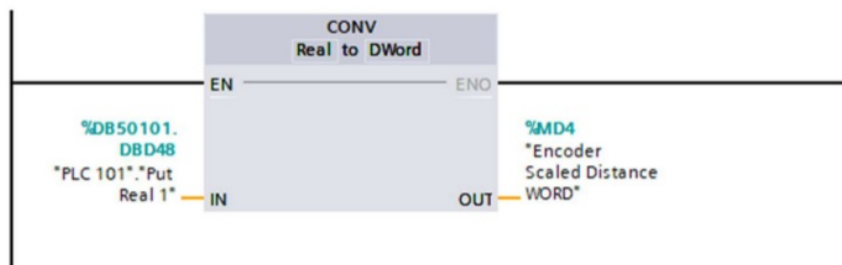


Figure 3.39 - Scaled Distance Converted to DWORD

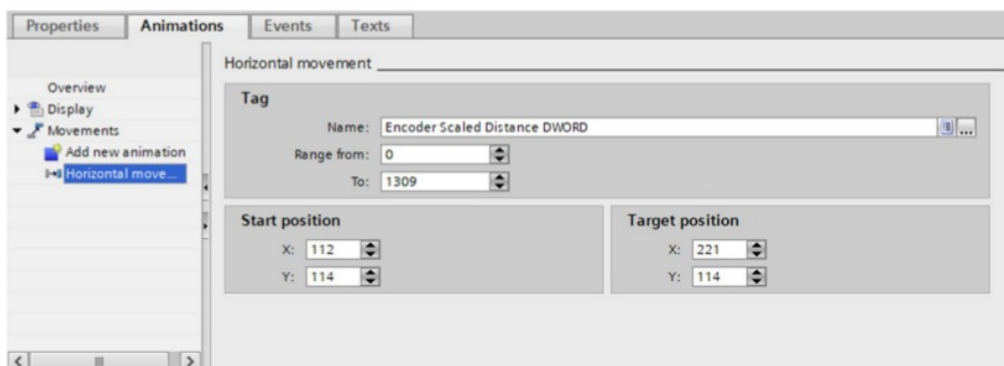


Figure 3.40 - Horizontal Animation

The range for the animation is 0 to 1309, which is the total length of the actuator in mm. The testing for the ladder program and HMI can be found in **Section 4.1**.

3.4 - Configuration and Commissioning of V90 Servo Drive

By using the Siemens SINAMICS V90 / SIMOTICS S-1FL6 operating manual, a diagram showing the required connections was found for single phase operation.

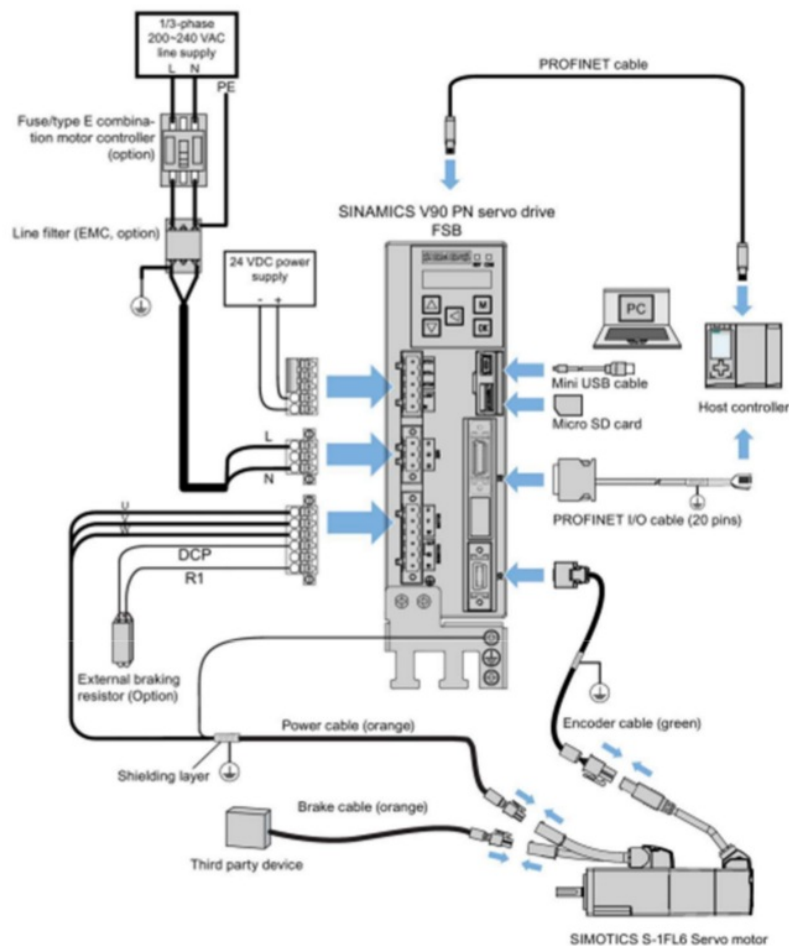


Figure 3.41 - V90 Single Phase Connection Diagram (Siemens, 2017)

Furthermore, a block diagram from the Siemens V90 brochure can be used.

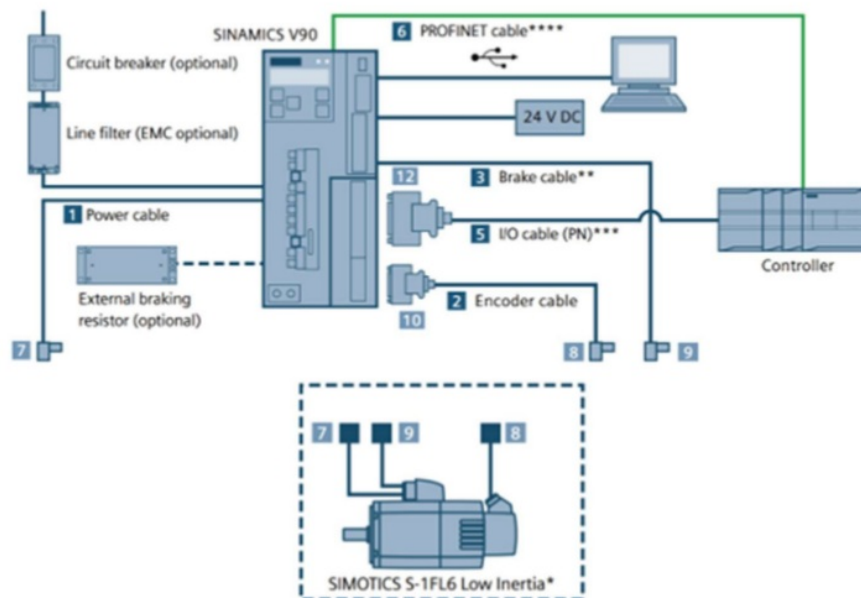


Figure 3.42 - V90 Connection Block Diagram (SIEMENS, 2016)

However, neither of these diagrams include an HMI, therefore a network switch will be used just as **Figure 3.7** shows. Below shows the physical connections and the network switch.

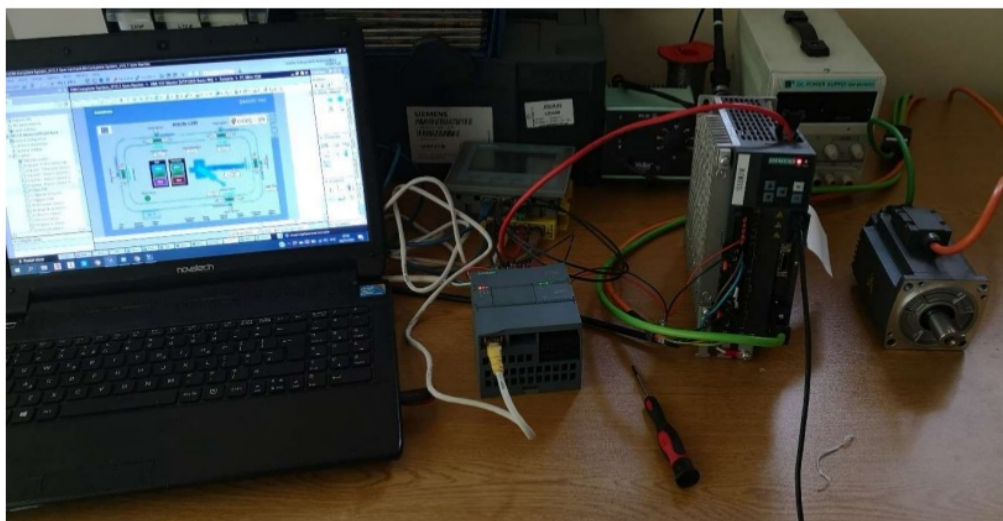


Figure 3.43 - Physical System



Figure 3.44 - Network Switch

- Blue – HMI
- Yellow – PLC
- Red – V90 Drive
- Grey – Laptop

To begin configuring the drive, the Siemens V-Assistant software was downloaded. This is a commissioning tool designed specifically for the V90 series. This software makes configuring parameters, tuning and diagnostics of the drive much easier than using the Basic Operator Panel (BOP) found on the front of the drive.

When the software is first opened, it asks the user to work in Online or Offline mode. Write in the third person In Offline mode you cannot write to the drive, so therefore the Online mode was selected. This can only be done when the USB interface between the PC and Drive is connected.

Once Online, the software automatically reads the drive and motor models from the drive and selects them as shown below.



Figure 3.45 - Drive and Motor Selection in V-Assistant

Firstly, speed control or epos control had to be selected. Epos did not seem to work with the HW configuration, meaning speed control was defaulted to. The next step was to set up PROFINET communications.

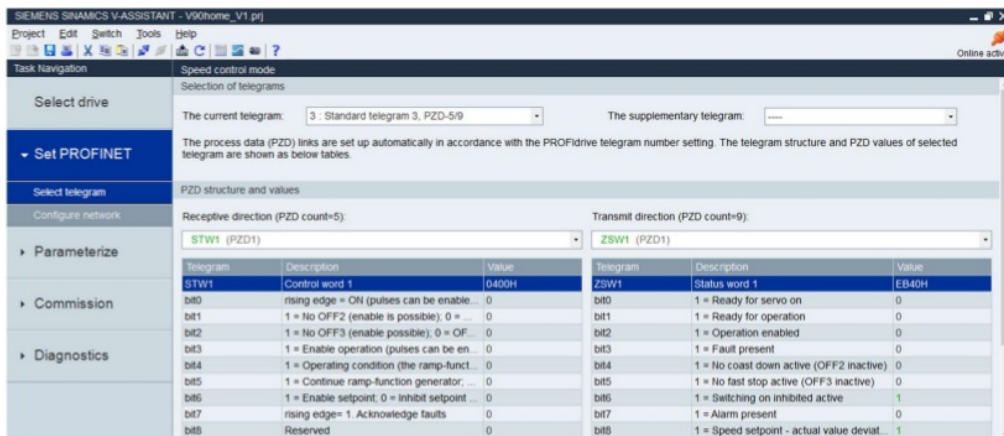


Figure 3.46 - PROFINET Setup

The Standard Telegram 3 was selected here, and each bit is described in the table underneath. The telegram selected is stored in parameter p0922. The PZD structure is STW1. A PZD is the German word initials for “Process Data”. PZD is an area of the PROFINET telegram used to exchange data between the drive and the controller cyclically: control word, status word, speed setpoint, speed status, torque, etc. (Siemens, 2006).

Telegram 3 with STW1 is the recommended selection for initial configuration of the V90 drive according to the V90 operating manual as Telegram 3 is the only one that supports the rotary incremental encoder type. (Siemens, 2017)

The network configuration was skipped following this as the network can be configured within TIA portal. The next step was to set the parameters.

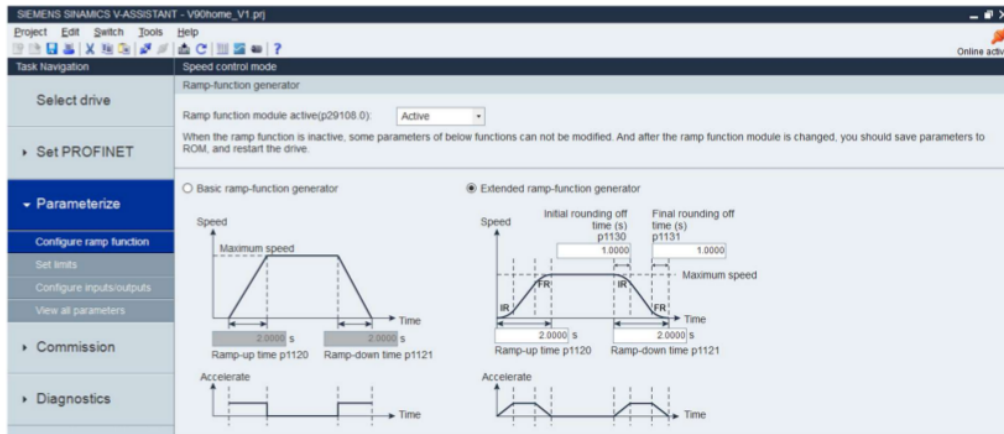


Figure 3.47 - Ramp Function Configuration

The parameter p29108.0 determines if the ramp function module is active or not. By setting this to 1, it allows for basic, and extended ramp-function generation. To make the operation of the motor extra smooth, the extended ramp-function generator was selected. The existing system had a 1s timer acting as a makeshift ramp up. However, due to the V90 servo having much higher speeds available, a 2s ramp up (p1120) and down (p1121) window was given. Additionally, the initial rounding off time (p1130) and final rounding off time (p1131) were set to 1 second making the ramping much smoother.

The next parameters to be set were the torque and speed limits. As stated in the previous assignment and in the FDS, the torque required to move the 29kg load at 2113rpm was 7.65Nm. One of the characteristics of this motor is that it could provide up to 300% torque for short periods if a resting period follows to reduce thermal stress. Therefore, this parameter is set to 300% in both directions.

Furthermore, the speed limit was set to 3000rpm in both directions. This was because the motor was rated for this speed. However, only 2113rpm is required to meet the FDS, so with the 3000rpm limits in place, it stops the motor from going above its rated speed, while allowing the required speed to meet the specification.

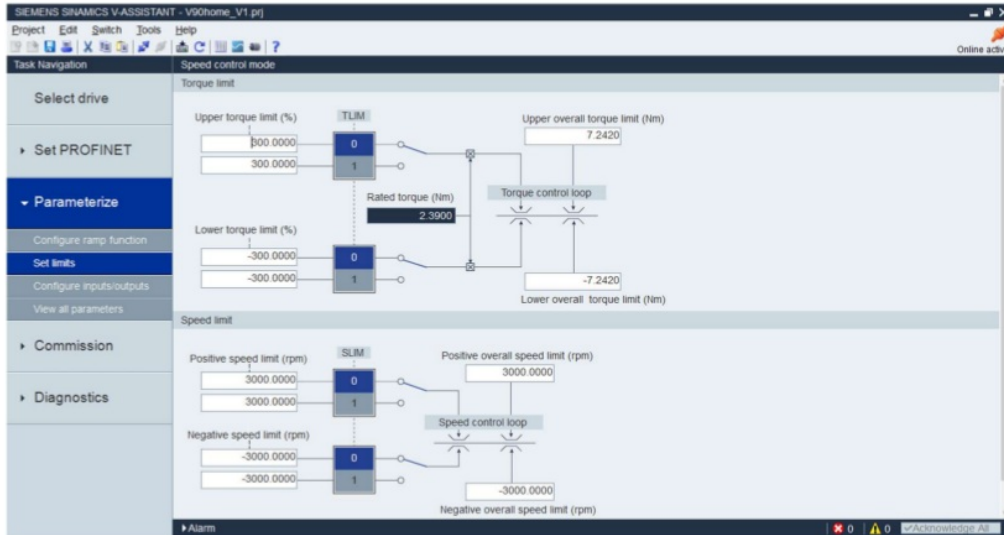


Figure 3.48 - Torque and Speed Limits

The inputs / outputs on the drive itself are not being used for this project and are therefore left alone. Finally, a table with all of the parameters can be found in the 'View All Parameters' section. These can all be found in the appendix.

With the initial configuration of the parameters done, it was time to commission the motor and test it by using the jog function. The first test was to run the motor in both directions at 3000rpm.



Figure 3.49 - 3000rpm CW Test



Figure 3.50 - 3000rpm CCW Test

Next, the speed limit was tested by inputting 5000rpm. The expected result is to see the speed only reach 3000rpm.



Figure 3.51 - 5000rpm Test

Finally, the drive was optimised by enabling One-Button Auto Tuning. The one-button auto tuning function allows the servo drive to automatically estimate the ratio of load moment of inertia. However, this requires the servo to be connected to the mechanical load of the Finishing Robot. Due to the pandemic this is not possible.

3.5 - V90 TIA Program

Another side effect of the corona virus meant that integrating the V90 hardware into the CIM system was not possible, which also made altering the CIM system program to control the motor much more difficult.

Therefore, a separate TIA project was made to simulate the operation of the motor as if it were connected to the finishing robot. To do this, an input switch module was connected to the inputs of the PLC to act as the signals coming from the Mitsi robot. Additionally, the only HMI available was a KTP400, meaning that only a basic HMI display could be made.

This separate program simply moves the motor to various positions depending on the 3-bit inputs from the switches. This logic can be seen below.

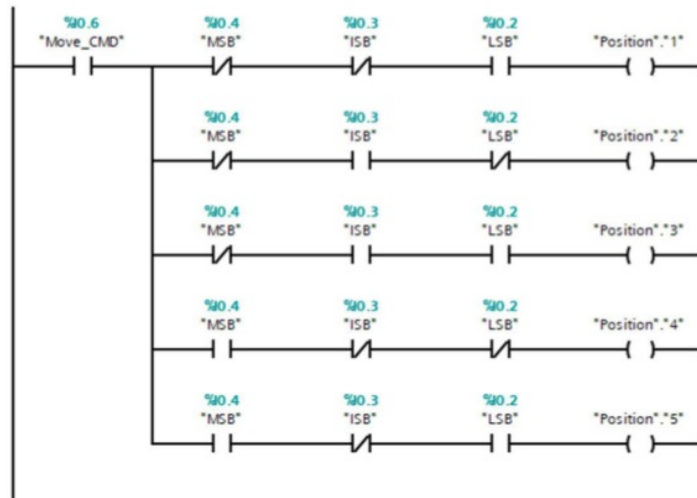


Figure 3.52 - 3-Bit Logic

The method used to control movement of the motor is using the Motion Control (MC) functions within TIA portal. To use these, a technology object must be made to control the axis. The first options are to select the type of Drive (PROFIdrive), and the unit of measurements (mm).

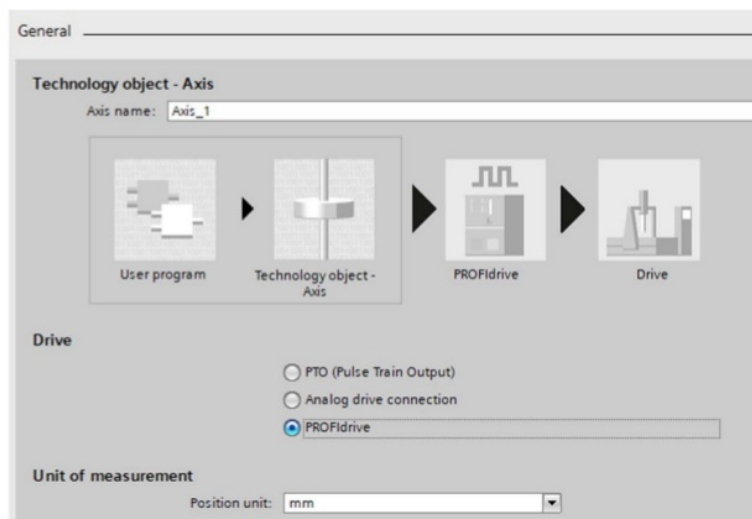


Figure 3.53 - Basic Parameters (General)

Next is the Drive Parameters, which includes selecting the drive in TIA portal, and which Telegram the drive will use to communicate to the PLC. To see the drive in these options, it must first be added to the project via the device Network View.

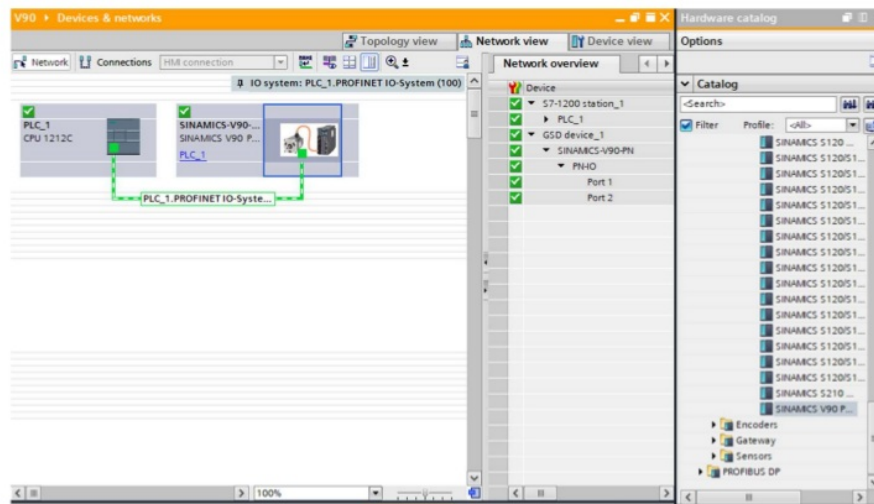


Figure 3.54 - Network View

To select the Telegram, the V90 must be selected in Device View, then searching through the Submodules for the correct Telegram. In V-Assistant, Standard Telegram 3 was selected, so the same was done in TIA.

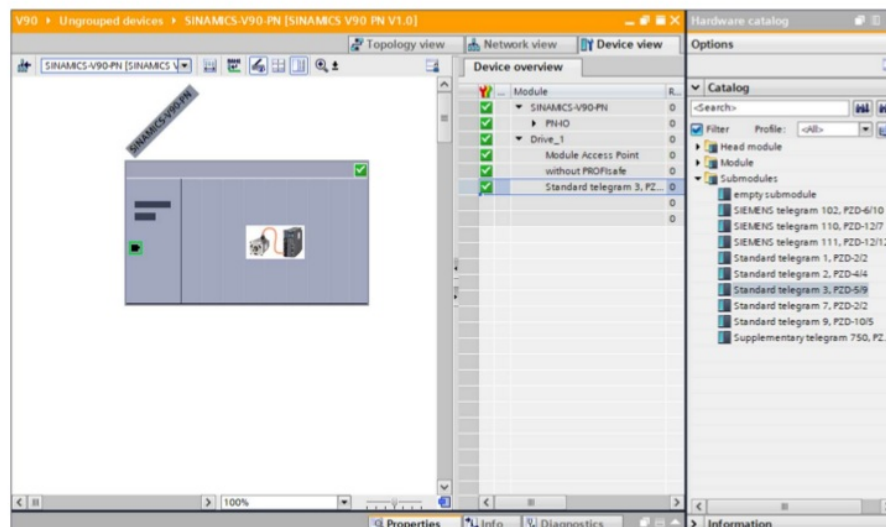


Figure 3.55 - Telegram Selection

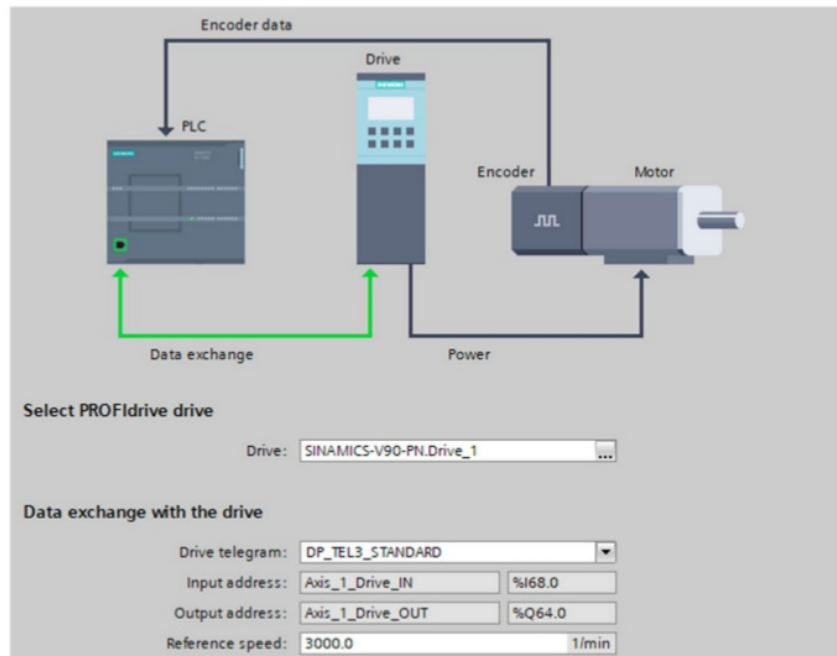


Figure 3.56 - Drive Parameters in TIA

Next, the encoder parameters are selected. These include encoder connection, encoder telegram, encoder type and resolution.

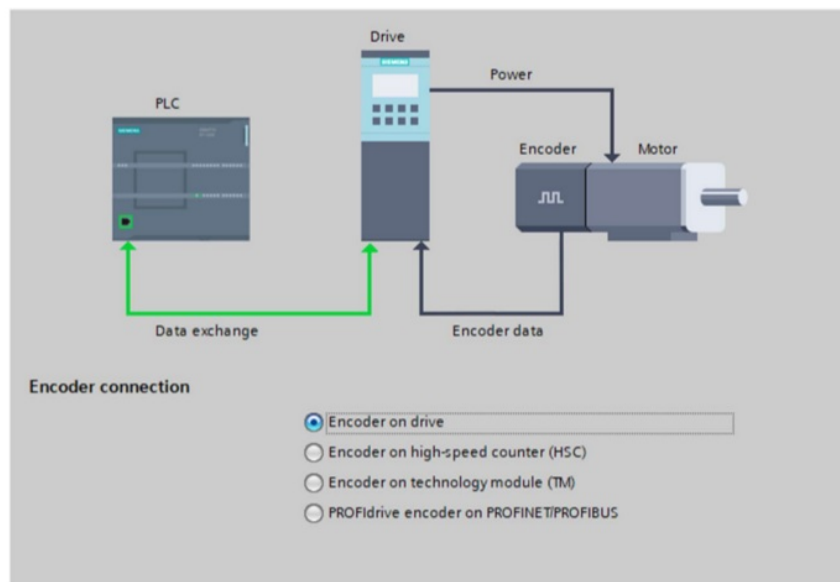


Figure 3.57 - Encoder Parameters 1

Data exchange with encoder

Encoder telegram:

Input address:

Output address:

Encoder type

Encoder type:

Steps per revolution:

Fine resolution

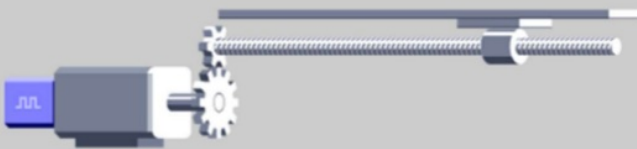
Bits in incr. actual value (GN_XIST1): Bits

☐ Invert encoder direction

Figure 3.58 - Encoder Parameters 2

Following the encoder parameters are the mechanics of the system. Here the mounting type and pitch are input. The modulo and position limits were ignored as they are only relevant after implementing the drive to the CIM system.

Encoder mounting type



Encoder mounting type:

Position parameters

Load movement per motor revolution: mm

Figure 3.59 – Mechanics

The final changes to the axis settings are the position monitoring parameters. These dictate how much tolerance time and dwelling time the motor is allowed before an error is sent.

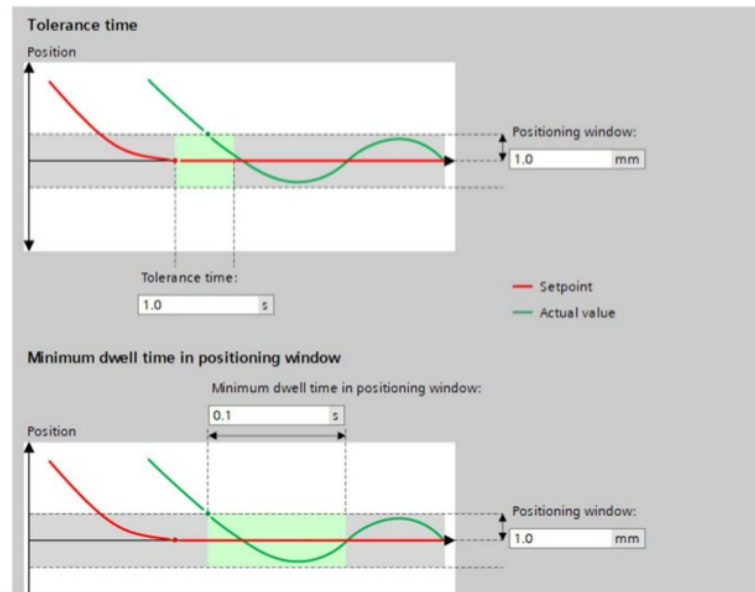


Figure 3.60 - Position Monitoring

For the PLC to control the V90 drive, the MC_Power function had to be used. This acts as an enable for the motor. This would be executed using an internal memory bit in the CIM system's PLC program, however for proof of operation, this is controlled by one of the input switches (%I0.0).

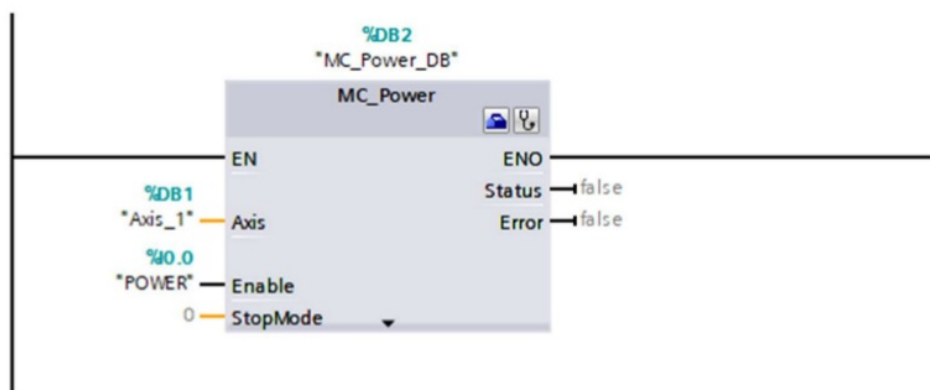


Figure 3.61 - MC_Power

Furthermore, if an error were to occur due to incorrect parameters, speed settings or following error then a MC_Reset function must be used to clear errors, which allows the motor to be re-enabled. Again, this would normally be controlled by an internal memory bit, which could be a HMI button. For now, it is controlled by %I0.7.

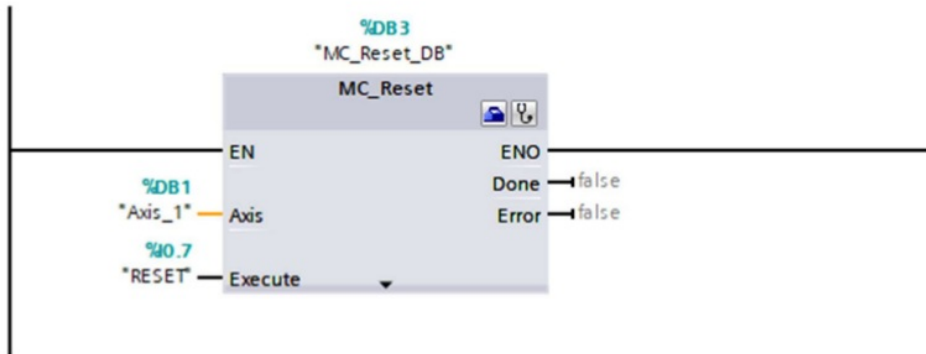
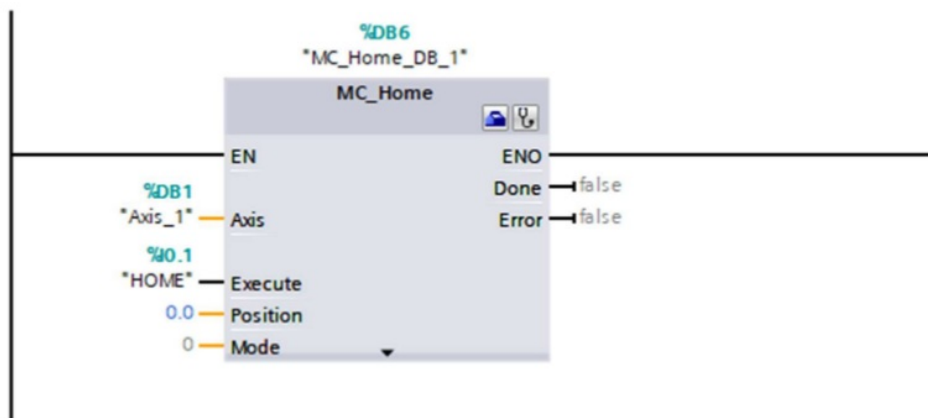


Figure 3.62 - MC_Reset

Before the motor can move, it must be homed which is done using the MC_Home function.



Finally, by using MC_MoveAbsolute, the motor can be told where to move to by the previously shown logic. To mimic the CIM system, 5 positions are selected.

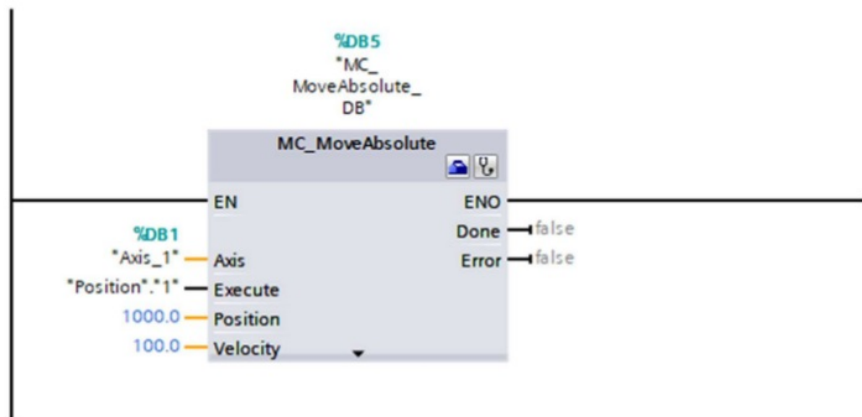


Figure 3.63 - V90 Position 1

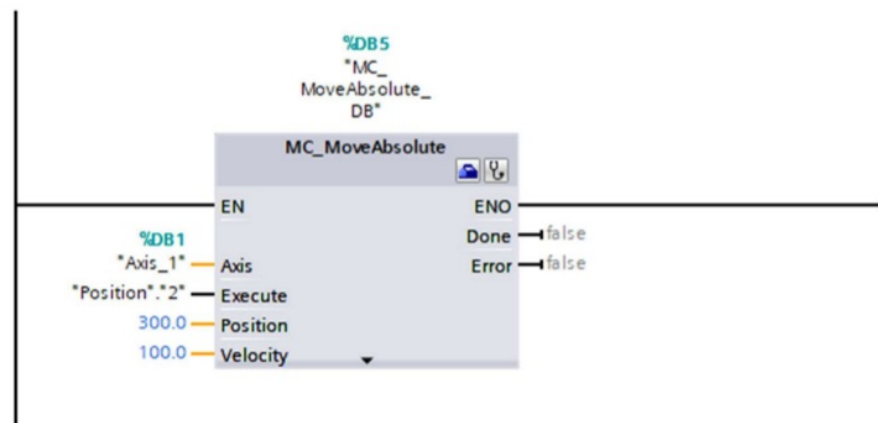


Figure 3.64 - V90 Position 2

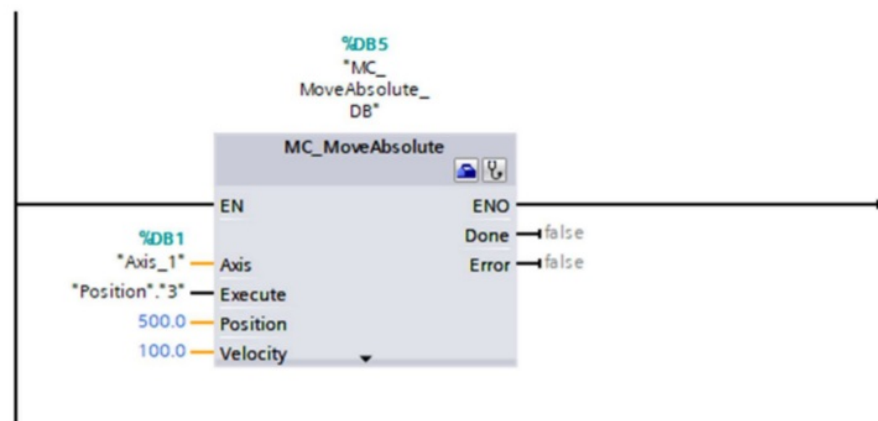


Figure 3.65 - V90 Position 3

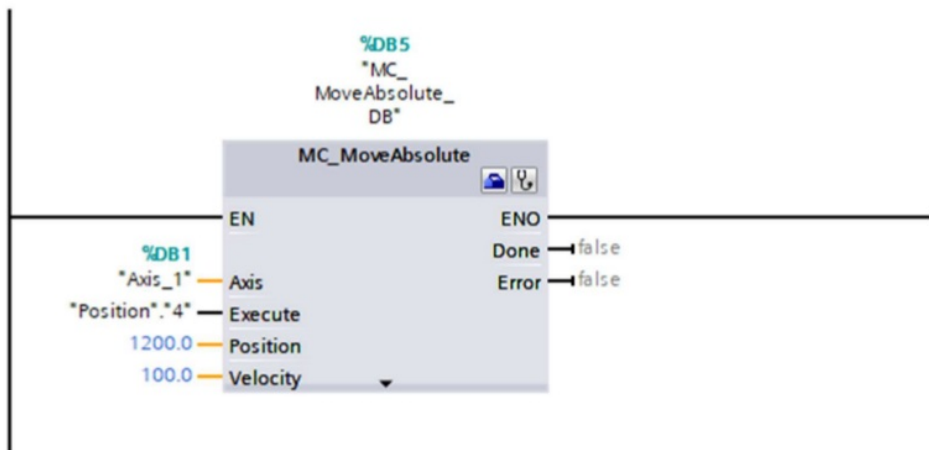


Figure 3.66 - V90 Position 4

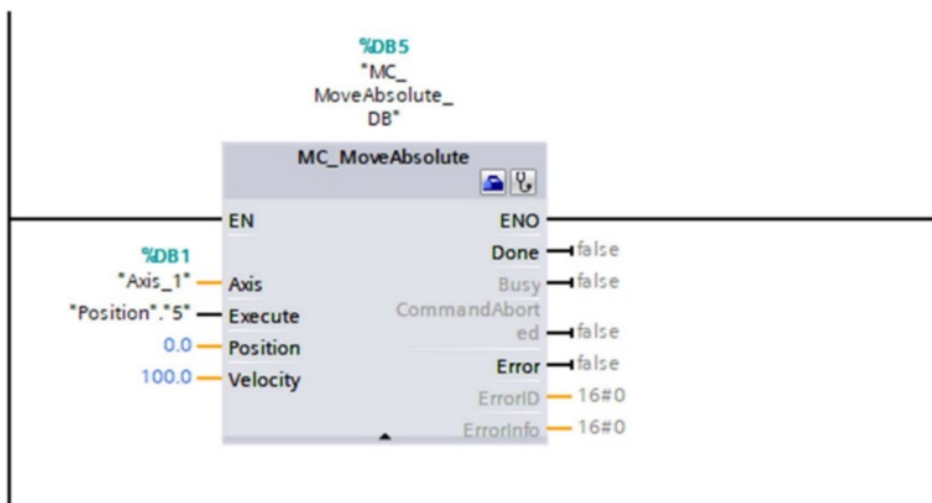


Figure 3.67 - V90 Position 5

The final part of creating this replacement setup for the CIM system was to create a HMI screen to display the position relative to the home position, the setpoint, and the position number. With the program written and HMI made, it was time to test the system.

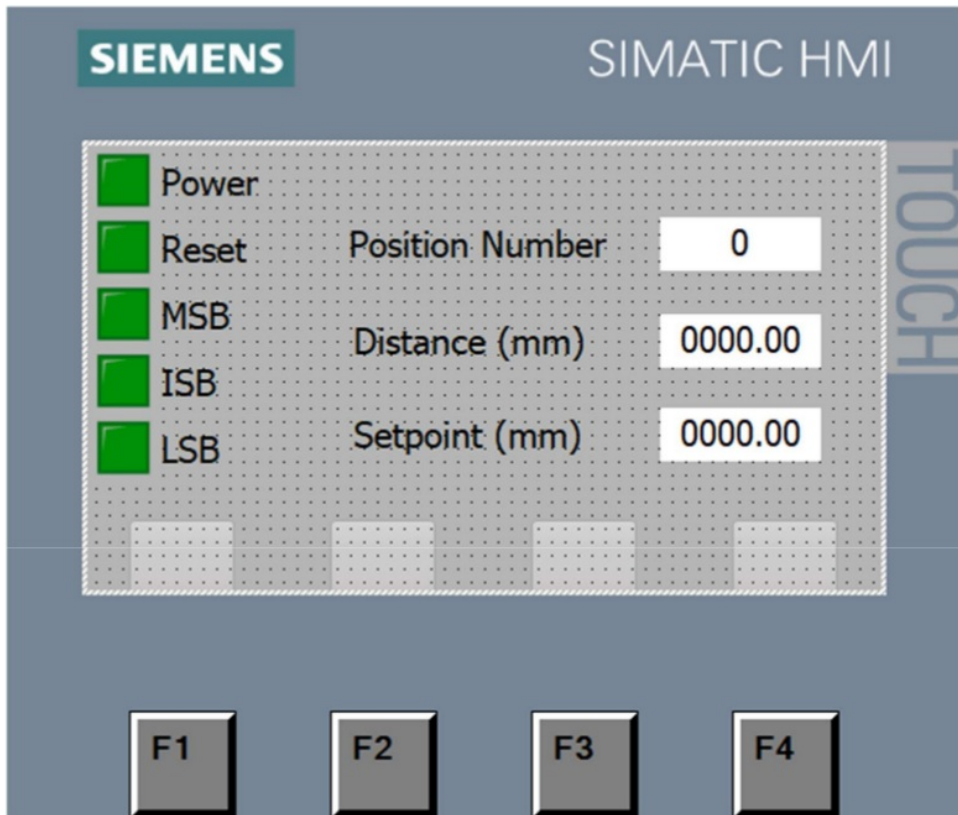


Figure 3.68 - Test HMI

4 - Testing and Results

4.1 - Replacing Existing Hardware

All the changes made during the process of replacing the Keyence PLC with the Siemens PLC were tested on 17/01/2020. The changes made were:

Manual Mode:

- The storing of 7 linear positions using HMI buttons.
- A routine to go to selected positions when the 'Move to Slider Position' command is given from the HMI was been created.
- Updated the screen to indicate when the robot is in each position.

Auto Mode:

- A routine to go to selected positions when told to by Mitsi Robot was created.
- Send signals back to Mitsi Robot about its status.
- Made it so 'In_Position_to_Mitsi' signal does not last as long.
- Tested auto sequence and adjusted stored positions.

Communication between PLC106 and PLC101 using GET and PUT so the position of the robot is stored on the master PLC was added. This allows the HMI screen to have an animated mimic of the linear drive.

The results of the tests were that the whole system worked as intended. Due to the pandemic, there is no evidence available to prove the success of these tests. However, all tests were done in the presence of Rob Davies who can confirm that the system operated as intended. The reason for making these changes was to ensure the system was still usable as an educational tool while the project progresses, and to use as a side by side comparison to the V90. Once the V90 could mimic the operation of the CIM system, it would only then be integrated to the system.

The log book should have been kept up to date at the time of testing, then the evidence would have been available

The HMI mimic could still be proved as a working concept. The test involves moving the position of the motor to 5 locations which will be displayed beside the mimic. When the motor moves, the arm graphic should also move to mirror it. This can be seen in the following **Figures**.

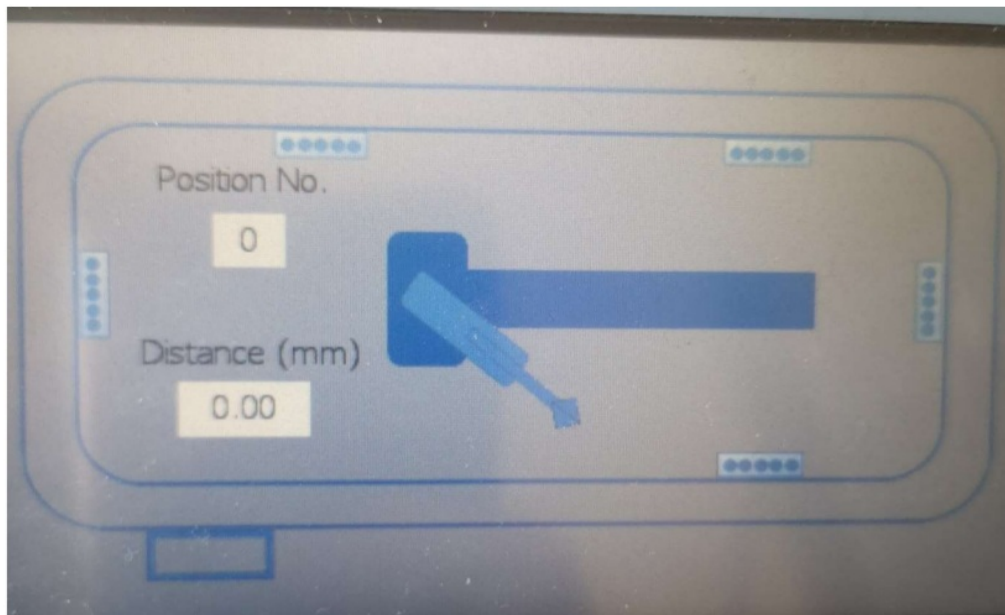


Figure 4.1 – Mimic Home Position (0.00mm)

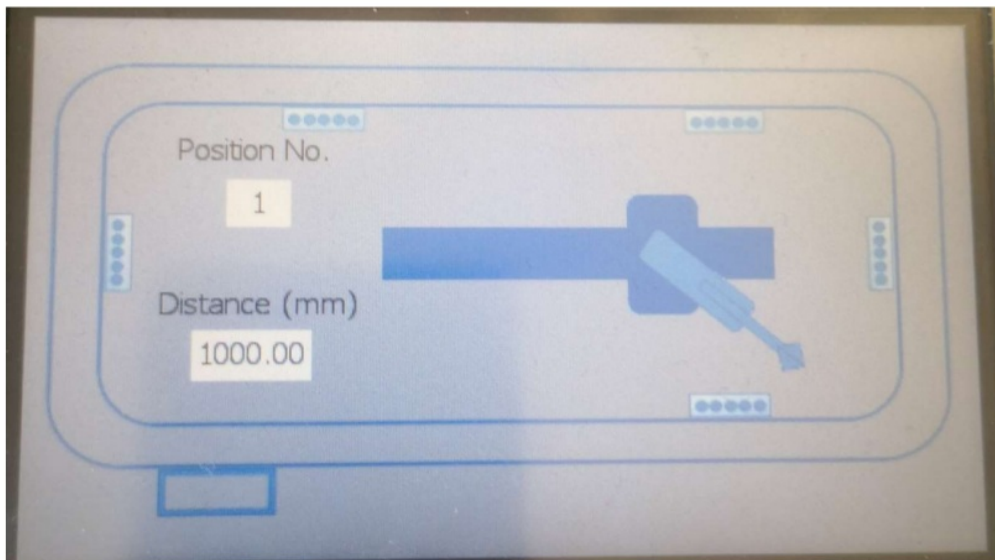


Figure 4.2 - Mimic Position 1 (1000.00mm)

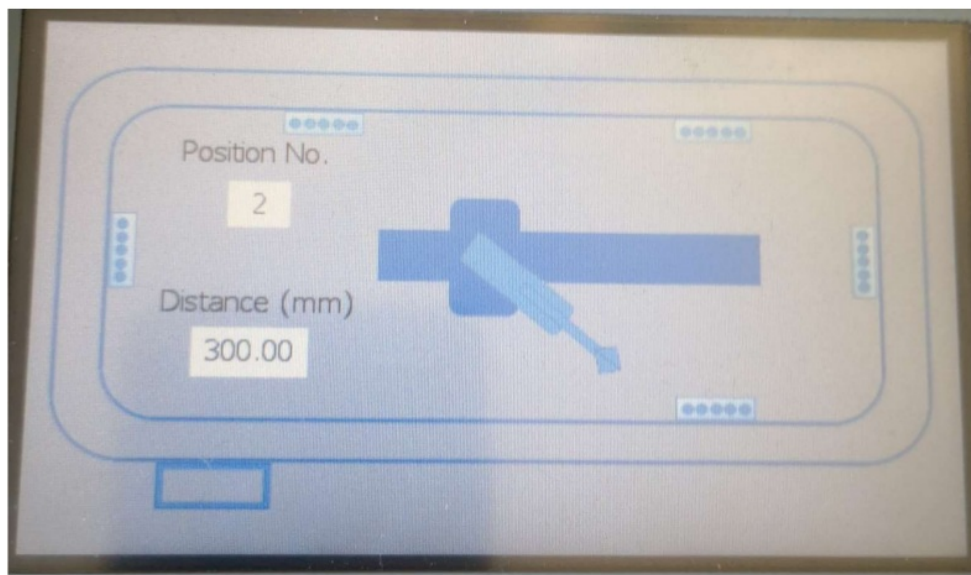


Figure 4.3 - Mimic Position 2 (300.00mm)

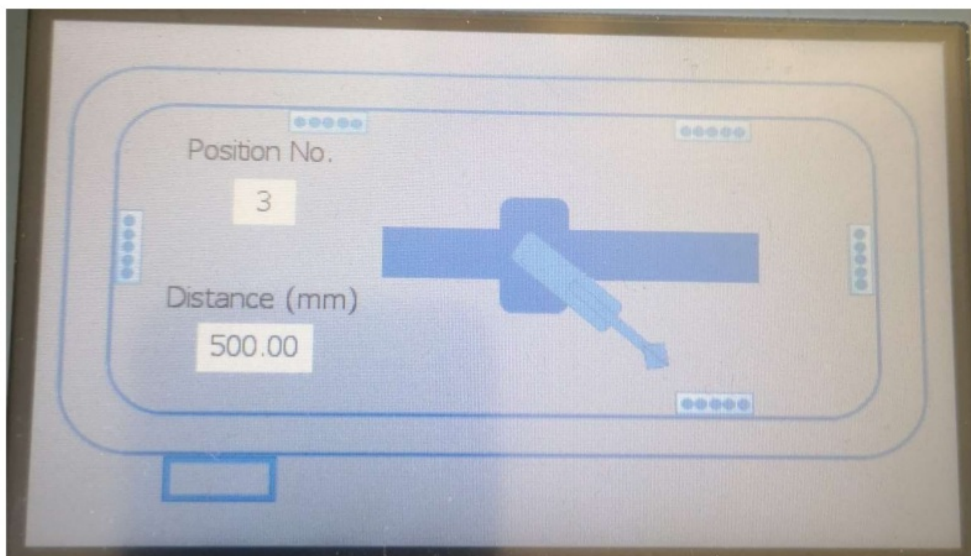


Figure 4.4 - Mimic Position 3 (500.00mm)

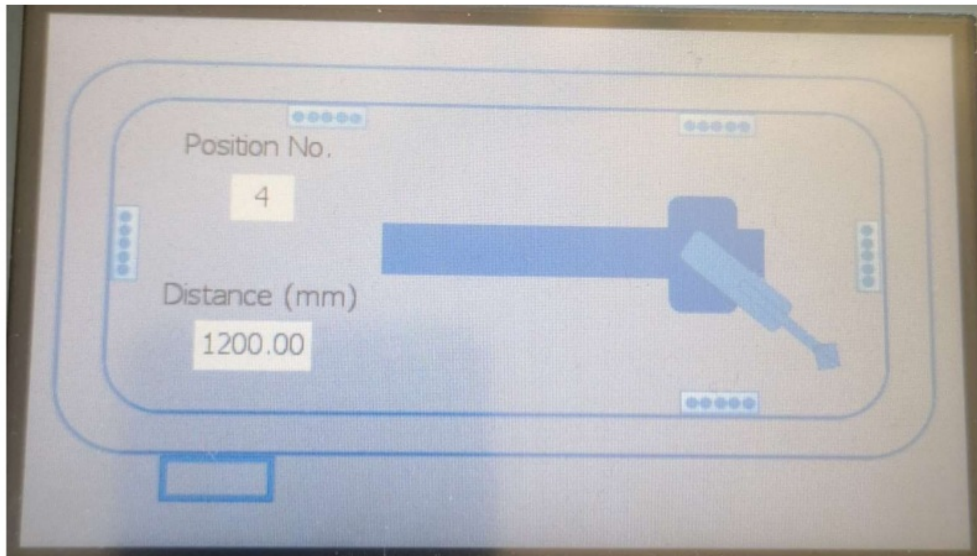


Figure 4.5 - Mimic Position 4 (1200.00mm)

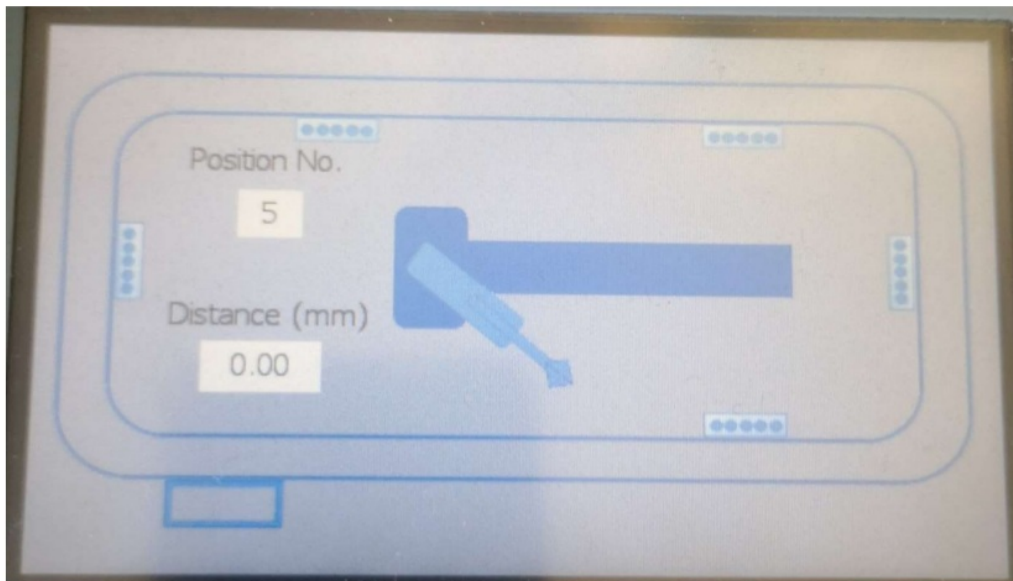


Figure 4.6 - Mimic Position 5 (0.00mm)

As shown, using the animation function in conjunction with an Int tag converted from the scaled distance, the graphic moves accordingly.

4.2 - V90 Tests

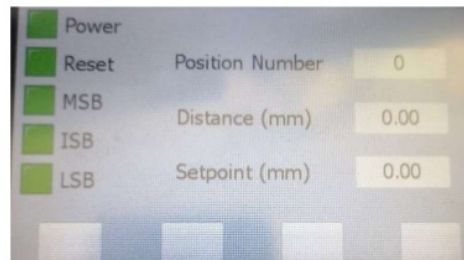
As previously mentioned, the V90 was set up in a test rig to mimic how it would operate within the CIM system. By receiving a 3-Bit number and a move command as inputs from switches, rather than the Mitsi robot, it should operate in the same manner.

Test
No.

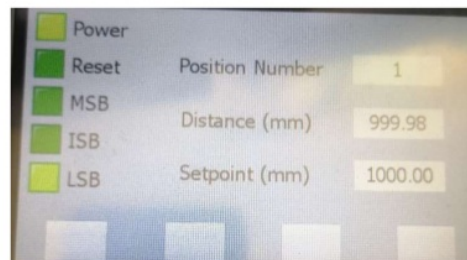
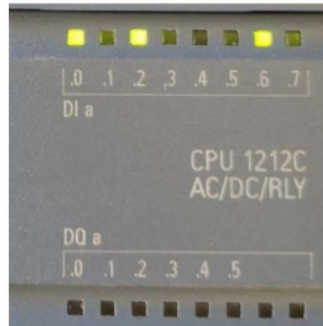
PLC Input Lights

HMI Screens

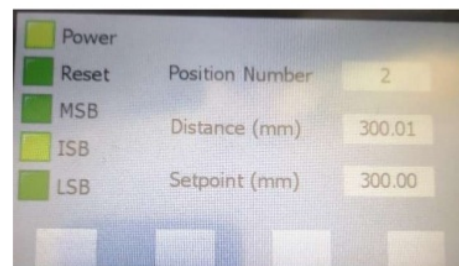
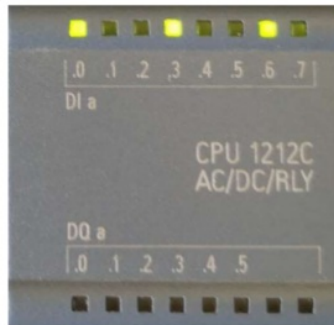
1



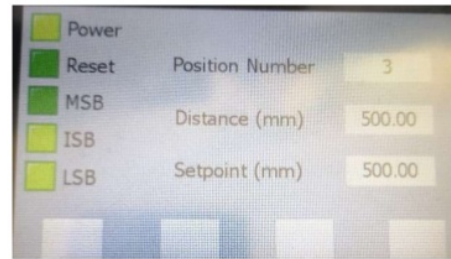
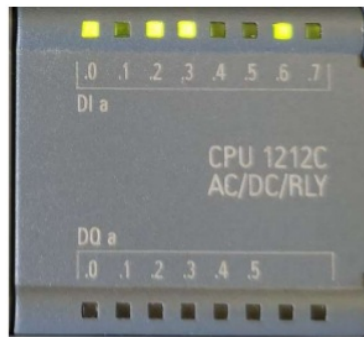
2



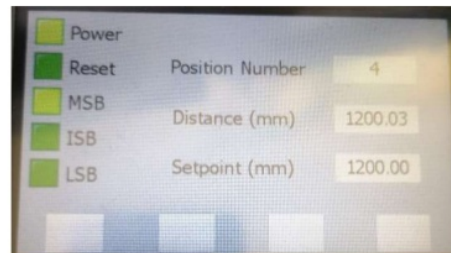
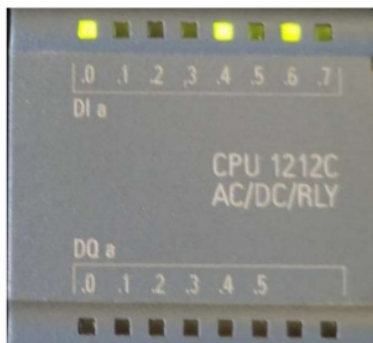
3



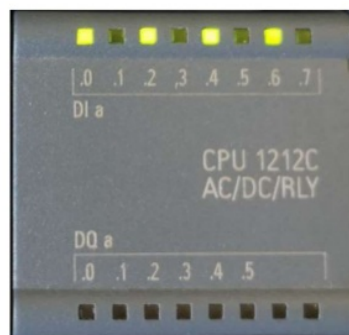
4



5



6



The tests each show the PLC input LEDs, HMI indicator lamps for the inputs, and data fields showing the position number, actual distance and setpoint. As shown, the motor's accuracy is within a maximum of 0.03mm error. This is very accurate but could probably be improved by playing with the motor configuration, specifically the motion tracking.

Additionally, the ramp-function operates correctly with a gradual increase to acceleration and deceleration over a 2 second ramp. This can be seen in the video accompanying the project presentation. Without actual implementation to the CIM system, no more tests can be done with the V90.

5 - Evaluation and the Wider Issues

*A number of
important
goals have
been met.*

Due to the project not being fully completed, the majority of the FDS criteria were not met. Theoretically, the V90 servo drive can achieve the following criteria from the FDS:

- Move a load of 19kg with a maximum torque of 1.834Nm @ 500rpm.
- Move the entire length of the actuator (132cm) within 31 seconds, giving the robot a minimum linear velocity of 43mm/s.
- Position control so that the position of the robot is always fed back to the controller allowing manual, and automatic position control.
- Speed control so that the robot can move at a range of speeds, rather than two. These speeds must be able to be manually or automatically adjusted within limits. This can be done by having variables that can be changed on the HMI, attached to the velocity input to MC_MoveAbsolute functions.
- Match the speed of the conveyor and follow the location of wagons so that it can pick up and place products on and off the wagons as they move. The conveyor moves at 176mm/s meaning that the motor must run at 2200rpm.

Additionally, there are a few criteria that had already been met:

- The system can instantly be interfaced to the PROFINET network by connecting each device an existing network switch that is part of the network. By using the GET and PUT functions, the system will communicate with PLC 101 making these criteria complete.
- The motor and drive have an efficiency (η) greater than 0.8 and is capable of four-quadrant operation with regenerative braking.
- The HMI has an animated mimic of the robot's position.
- Diagnostic tools such as indicators were implemented to the CIM system's HMI, along with data being displayed on the test rig's HMI such as the motor's position, which was obtained via PROFINET.
- The HMI screen has automatic and manual control options.

Due to the COVID-19 pandemic, a panel was unable to be built to house the system. However, the additional cost of the housing, trunking, circuit breakers, safety circuits and peripherals would drive the overall cost above the £1100 budget. The cost of these should have been considered when looking for solutions.

With the system easily integrated to the existing PROFINET network, the system is that much closer to “Industry 4.0” standards as data is much more accessible throughout the network from the Finishing Station.

Since the motor has regenerative braking capabilities and high efficiency the environmental consideration mentioned in the initial project report have been fulfilled. Additionally, the motor and drive bodies are made of majority recyclable parts, and the internal electronics can easily be repurposed or re-used. Thus, potential waste is reduced once this system is made obsolete.

Furthermore, the use of PROFIenergy was not implemented to the test rig as it would not save energy from extended downtime, maintenance or inactivity. This would further reduce energy consumption after implementation to the CIM system. The same can be said for safety peripherals and circuitry. These were not available for the test rig and would only be practical on the CIM system itself where a real border could be made with light curtains or an enclosure.

6 - Conclusions and Recommendations

Since the drive was not implemented to the CIM system and the FDS criteria are incomplete, the overall project has not been a success. However, many conclusions can be made. Firstly, like many companies worldwide, the possibility of a global pandemic causing a global lockdown was not considered. The risk assessment done for the initial report did not include obstacles of this magnitude, not exclusive to pandemics. In future projects, contingency plans should always be made to ensure that even if a major obstacle were to arise, the project could still be completed. Even though, project completion may not have been possible due to the lockdown, additional progress could probably have been made if preparations were made.

Luckily, the V90 drive and servo motor were taken off site to be worked on before the lockdown occurred. If that hadn't been the case then a lot less progress would have been made as the configuration and commissioning of the drive would have been impossible.

Secondly, the budget of the project would have been exceeded if the panel were built. This highlights two potential problems; too little of a budget or poor selection of a solution. Since budget is not always negotiable, and this project was given £1100 to work with, the latter problem is most likely the one that could have been solved. A fourth solution has recently arisen that could have potentially outshone the V90 drive.

This belongs in assignment 1, your literature review. Your solution is good because it is easy to integrate into the CIM and provides the required performance.

The ODrive is a cheaper alternative to industrial servo motor systems. Servo motors are dominant in industry because they do not lose steps, can be very powerful, efficient and silent. The ODrive is opensource hardware and software that drives BLDC motors and converts them into what is essentially a servo motor by simply attaching a high-resolution encoder to the shaft.

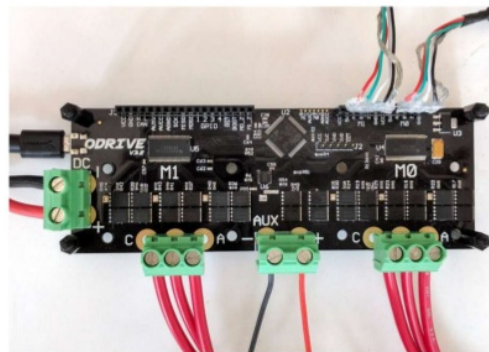


Figure 6.1 - Odrive Board

The result of this conversion is a very precise, affordable and powerful alternative to industrial Servo motors. The board supports regenerative braking and can be interfaced with a PLC using step/direction, or Servo PWM/PPM. The downside is that it cannot be directly interfaced to the PROFINET network. However, for the much lower price of £138 for board and encoder, to be used with any BLDC motor it could have been a potentially strong solution if budget absolutely could not be exceeded. (Weigl, 2018)

Overall, the project could have gone a lot better and more progress could have been made towards the end goal if better planning had been done, and precautions had been put in place so that situations like the lockdown would have minimal impact.

7 - Citation List

Jens Weidauer, R. M., 2014. *Electrical Drives*. Erlangen: Publicis Publishing.

Kendrick, T., 2015. *Identifying and Managing Project Risk*. New York: AMACOM.

PI, 2010. *The PROFienergy Profile*, Karlsruhe: PROFIBUS Nutzerorganisation e.V.(PNO).

Siemens, 2006. *What terminology is used in cyclic PROFIdrive communication?*. [Online]

Available at: <https://support.industry.siemens.com/cs/document/24184004/welche-terminologie-wird-bei-der-zyklischen-profidrive-kommunikation-verwendet?dti=0&lc=de-WW>

[Accessed 3 April 2020].

SIEMENS, 2016. *SINAMICS V90*. Erlangen: Siemens AG.

Siemens, 2016. *Speed Control with SINAMICS V90 and SIMATIC S7-1200 via PROFINET*.

[Online]

Available at: [https://cache.industry.siemens.com/dl/files/222/109739222/att_887931/v2/109739222_Speed Control_V90_S7-1200_DOC_en.pdf](https://cache.industry.siemens.com/dl/files/222/109739222/att_887931/v2/109739222_Speed_Control_V90_S7-1200_DOC_en.pdf)

[Accessed 29 March 2020].

Siemens, 2017. *SINAMICS V90, SIMOTICS S-1FL6 Operating Instructions*. [Online]

Available at: https://cache.industry.siemens.com/dl/files/475/109748475/att_923327/v1/V90_PN_1FL6_op_instr_0417_en-US.pdf

[Accessed 17 May 2020].

Weigl, O., 2018. *ODrive - High performance motor control*. [Online]

Available at: <https://odriverobotics.com/>

[Accessed 14 April 2020].

8 - Bibliography

*This list cannot contain the
same sources as the
Creative list.*

Jens Weidauer, R. M., 2014. *Electrical Drives*. Erlangen: Publicis Publishing.

Kendrick, T., 2015. *Identifying and Managing Project Risk*. New York: AMACOM.

PI, 2010. *The PROFienergy Profile*, Karlsruhe: PROFIBUS Nutzerorganisation e.V.(PNO).

Siemens, 2006. *What terminology is used in cyclic PROFIdrive communication?*. [Online]

Available at: <https://support.industry.siemens.com/cs/document/24184004/welche-terminologie-wird-bei-der-zyklischen-profidrive-kommunikation-verwendet?dti=0&lc=de-WW>

[Accessed 3 April 2020].

SIEMENS, 2016. *SINAMICS V90*. Erlangen: Siemens AG.

Siemens, 2016. *Speed Control with SINAMICS V90 and SIMATIC S7-1200 via PROFINET*.

[Online]

Available at:

https://cache.industry.siemens.com/dl/files/222/109739222/att_887931/v2/109739222_Speed_Control_V90_S7-1200_DOC_en.pdf

[Accessed 29 March 2020].

Siemens, 2017. *SINAMICS V90, SIMOTICS S-1FL6 Operating Instructions*. [Online]

Available at:

https://cache.industry.siemens.com/dl/files/475/109748475/att_923327/v1/V90_PN_1FL6_op_instr_0417_en-US.pdf

[Accessed 17 May 2020].

Weigl, O., 2018. *ODrive - High performance motor control*. [Online]

Available at: <https://odriverobotics.com/>

[Accessed 14 April 2020].

9 - Appendix

Group	Parameter No.	Name	Value	Unit	Range	Factory setting	Effect type
App	p29000	Motor ID	58	N.A.	[0, 65535]	0	immediately
App	p29001	Reversal of motor direction	0 : Direction n...	N.A.	--	0	immediately
App	p29002	BOP display selection	0 : Speed	N.A.	--	0	immediately
App	p29003	Control mode	2 : S	N.A.	--	2	reset
App	p29005	Brake resistor capacity percenta...	100.0000	%	[1, 100]	100.0000	immediately
App	p29006	Line supply voltage	230	V	[200, 480]	400	immediately
App	p29020[0]	► Tuning: Dynamic factor : One-bu...	18	N.A.	[1, 35]	18	immediately
App	p29021	Tuning: Mode selection	0 : Disable	N.A.	--	0	immediately
App	p29022	Tuning: Ratio of total inertia mo...	1.0000	N.A.	[1, 10000]	1.0000	immediately
App	p29023	Tuning: One-button auto tuning ...	0007H	N.A.	--	0007H	immediately
App	p29024	Tuning: Real-time auto tuning co...	004CH	N.A.	--	004CH	immediately
App	p29025	Tuning: Configuration overall	001CH	N.A.	--	0004H	immediately
App	p29026	Tuning: Test signal duration	2000	ms	[0, 5000]	2000	immediately
App	p29027	Tuning: Limit rotation of motor	100	°	[0, 30000]	0	immediately
App	p29028	Tuning: Pre-control time constant	7.5000	ms	[0, 60]	7.5000	immediately
App	p29035	VIBSUP activation	0 : Disable	N.A.	--	0	immediately
App	p29050[0]	► Torque limit upper : Torque limit	300.0000	%	[-150, 300]	300.0000	immediately
App	p29051[0]	► Torque limit lower : Torque limit L...	-300.0000	%	[-300, 150]	-300.0000	immediately
App	p29070[0]	► Speed limit positive : Speed limit	3000.0000	rpm	[0, 210000]	210000.0000	immediately
App	p29071[0]	► Speed limit negative : Speed lim...	-3000.0000	rpm	[-210000, 0]	-210000.0000	immediately
App	p29080	Overload threshold for output si...	100.0000	%	[10, 300]	100.0000	immediately
App	p29108	Function module activate	00000001H	N.A.	--	00000000H	immediately
App	p29110	Position loop gain	1.8000	1000/min	[0, 300]	1.8000	immediately
App	p29111	Speed pre-control factor (feed fo...	0.0000	%	[0, 200]	0.0000	immediately
App	p29120	Speed loop gain	0.0264	Nms/rad	[0, 999999]	0.3000	immediately

Figure 9.1 - Appendix 1 – Parameters

Group	Parameter No.	Name	Value	Unit	Range	Factory setting	Effect type
App	p29121	Speed loop integral time	17.2682	ms	[0, 100000]	15.0000	immediately
App	p29150	User defined PZD receive	0 : No function	N.A.	--	0	immediately
App	p29151	User defined PZD send	0 : No function	N.A.	--	0	immediately
App	p29230	MDI direction selection	0 : MDI shorte...	N.A.	--	0	immediately
App	p29231	MDI positioning type	0 : MDI relativ...	N.A.	--	0	immediately
App	p29240	Select referencing mode	1 : External re...	N.A.	--	1	immediately
App	p29243	Position tracking activate	0 : Position tr...	N.A.	--	0	immediately
App	p29244	Absolute encoder virtual rotary r...	0	N.A.	[0, 4096]	0	immediately
App	p29245	Axis mode state	0	N.A.	[0, 1]	0	immediately
App	p29246	Modulo correction range	360000	N.A.	[1, 2147482647]	360000	immediately
App	p29247	Mechanical gear: LU per revoluti...	10000	N.A.	[1, 2147483647]	10000	immediately
App	p29248	Mechanical gear: Numerator	1	N.A.	[1, 1048576]	1	immediately
App	p29249	Mechanical gear: Denominator	1	N.A.	[1, 1048576]	1	immediately
App	p29301	Digital input 1 assignment	2 : DI_RESET	N.A.	--	2	immediately
App	p29302	Digital input 2 assignment	11 : DI_TLIM	N.A.	--	11	immediately
App	p29303	Digital input 3 assignment	0 : DI_NA	N.A.	--	0	immediately
App	p29304	Digital input 4 assignment	0 : DI_NA	N.A.	--	0	immediately
App	p29330	Digital output 1 assignment	2 : DO_FAULT	N.A.	--	2	immediately
App	p29331	Digital output 2 assignment	9 : DO_OLL	N.A.	--	9	immediately
App	p29360	Brake resistor alarm active	1 : activate m...	N.A.	--	1	immediately
App	p29418	Fine resolution G1_XIST1 (in bits)	2	N.A.	[2, 18]	11	immediately
App	p29419	Fine resolution G1_XIST2 (in bits)	9	N.A.	[2, 18]	9	immediately
Data	r311	Rated motor speed	3000.0000	rpm	[0, 210000]	0.0000	immediately
Base	p748	CU invert digital outputs	00000000H	N.A.	--	00000000H	immediately
Com	p922	IF1 PROFIdrive PZD telegram se...	3 : Standard t...	N.A.	--	105	immediately

Figure 9.2 - Appendix 2 - Parameters

Group	Parameter No	Name	Value	Unit	Range	Factory setting	Effect type
Com	p925	PROFIdrive clock synchronous s...	1	N.A.	[0, 65535]	1	immediately
Com	p927	Param authority	0003H	N.A.	[0, 3]	0003H	immediately
Base	p1058	Jog 1 speed setpoint	100.0000	rpm	[0, 210000]	100.0000	immediately
Base	p1082	Maximum speed	3000.0000	rpm	[0, 210000]	1500.0000	immediately
Base	p1083	Speed limit in positive direction o...	3000.0000	rpm	[0, 210000]	210000.0000	immediately
Base	p1086	Speed limit in negative direction ...	-3000.0000	rpm	[-210000, 0]	-210000.0000	immediately
Base	p1115	Ramp-function generator selection	1 : Extended r...	N.A.	--	0	immediately
Base	p1120	Ramp-function generator ramp-u...	2.0000	s	[0, 999999]	1.0000	immediately
Base	p1121	Ramp-function generator ramp-d...	2.0000	s	[0, 999999]	1.0000	immediately
Base	p1130	Ramp-function generator initial r...	1.0000	s	[0, 30]	0.0000	immediately
Base	p1131	Ramp-function generator final ro...	1.0000	s	[0, 30]	0.0000	immediately
Base	p1135	OFF3 ramp-down time	0.0000	s	[0, 600]	0.0000	immediately
Base	p1215	Motor holding brake configuration	0 : No motor ...	N.A.	--	0	immediately
Base	p1216	Motor holding brake opening time	105.0000	ms	[0, 10000]	100.0000	immediately
Base	p1217	Motor holding brake closing time	15.0000	ms	[0, 10000]	100.0000	immediately
Base	p1226	Threshold for zero speed detecti...	20.0000	rpm	[0, 210000]	20.0000	immediately
Base	p1227	Zero speed detection monitoring...	300.0000	s	[0, 300]	4.0000	immediately
Base	p1228	Pulse suppression delay time	0.0000	s	[0, 299]	0.0000	immediately
Base	p1414	Speed setpoint filter activation	0001H	N.A.	--	0000H	immediately
Base	p1415	Speed setpoint filter 1 type	2 : General 2...	N.A.	--	0	immediately
Base	p1416	Speed setpoint filter 1 time const.	0.0000	ms	[0, 5000]	0.0000	immediately
Base	p1417	Speed setpoint filter 1 denomina...	100.0000	Hz	[0.5, 16000]	1999.0000	immediately
Base	p1418	Speed setpoint filter 1 denomina...	0.9000	N.A.	[0.001, 10]	0.7000	immediately
Base	p1419	Speed setpoint filter 1 numerator...	100.0000	Hz	[0.5, 16000]	1999.0000	immediately
Base	p1420	Speed setpoint filter 1 numerator...	0.9000	N.A.	[0, 10]	0.7000	immediately

Figure 9.3 - Appendix 3 – Parameters

Group	Parameter No	Name	Value	Unit	Range	Factory setting	Effect type
Base	p1421	Speed setpoint filter 2 type	0 : Low pass...	N.A.	--	0	immediately
Base	p1422	Speed setpoint filter 2 time const.	0.0000	ms	[0, 5000]	0.0000	immediately
Base	p1423	Speed setpoint filter 2 denomina...	1999.0000	Hz	[0.5, 16000]	1999.0000	immediately
Base	p1424	Speed setpoint filter 2 denomina...	0.7000	N.A.	[0.001, 10]	0.7000	immediately
Base	p1425	Speed setpoint filter 2 numerator...	1999.0000	Hz	[0.5, 16000]	1999.0000	immediately
Base	p1426	Speed setpoint filter 2 numerator...	0.7000	N.A.	[0, 10]	0.7000	immediately
Base	p1441	Actual speed smoothing time	1.0000	ms	[0, 50]	0.0000	immediately
Base	p1520	Torque limit upper/motoring	7.2420	Nm	[-1000000, 2E...	0.0000	immediately
Base	p1521	Torque limit lower/regenerative	-7.2420	Nm	[-2E+07, 1000...	0.0000	immediately
Base	p1656	Activates current setpoint filter	0001H	N.A.	--	0001H	immediately
Base	p1658	Current setpoint filter 1 denomin...	1999.0000	Hz	[0.5, 16000]	1999.0000	immediately
Base	p1659	Current setpoint filter 1 denomin...	0.7000	N.A.	[0.001, 10]	0.7000	immediately
Base	p1663	Current setpoint filter 2 denomin...	974.2245	Hz	[0.5, 16000]	1000.0000	immediately
Base	p1664	Current setpoint filter 2 denomin...	0.2000	N.A.	[0.001, 10]	0.3000	immediately
Base	p1665	Current setpoint filter 2 numerat...	974.2245	Hz	[0.5, 16000]	1000.0000	immediately
Base	p1666	Current setpoint filter 2 numerat...	0.0000	N.A.	[0, 10]	0.0100	immediately
Base	p1668	Current setpoint filter 3 denomin...	1000.0000	Hz	[0.5, 16000]	1000.0000	immediately
Base	p1669	Current setpoint filter 3 denomin...	0.3000	N.A.	[0.001, 10]	0.3000	immediately
Base	p1670	Current setpoint filter 3 numerat...	1000.0000	Hz	[0.5, 16000]	1000.0000	immediately
Base	p1671	Current setpoint filter 3 numerat...	0.0100	N.A.	[0, 10]	0.0100	immediately
Base	p1673	Current setpoint filter 4 denomin...	1000.0000	Hz	[0.5, 16000]	1000.0000	immediately
Base	p1674	Current setpoint filter 4 denomin...	0.3000	N.A.	[0.001, 10]	0.3000	immediately
Base	p1675	Current setpoint filter 4 numerat...	1000.0000	Hz	[0.5, 16000]	1000.0000	immediately
Base	p1676	Current setpoint filter 4 numerat...	0.0100	N.A.	[0, 10]	0.0100	immediately
Base	p2000	Reference speed reference freq...	3000.0000	rpm	[5, 210000]	3000.0000	immediately

Figure 9.4 - Appendix 4 - Parameters

Group	Parameter No	Name	Value	Unit	Range	Factory setting	Effect type
Base	p2002	Reference current	14.2000	Arms	[0.1, 100000]	100.0000	immediately
Base	p2003	Reference torque	7.2420	Nm	[0.01, 2E+07]	1.0000	immediately
Base	p2118[0]	Change message type message ...	6310	N.A.	[0, 65535]	6310	immediately
Base	p2119[0]	Change message type type	2 : Alarm (A)	N.A.	--	2	immediately
Base	p2153	Speed actual value filter time co...	0.0000	ms	[0, 1000000]	0.0000	immediately
Base	p2161	Speed threshold 3	10.0000	rpm	[0, 210000]	10.0000	immediately
Base	p2162	Hysteresis speed n_act > n_max	250.0000	rpm	[0, 60000]	0.0000	immediately
Base	p2175	Motor blocked speed threshold	210000.0000	rpm	[0, 210000]	210000.0000	immediately
Base	p2177	Motor blocked delay time	0.5000	s	[0, 65]	0.5000	immediately
EPOS	p2525	LR encoder adjustment offset	0	LU	[0, 4294967295]	0	immediately
EPOS	p2533	LR position setpoint filter time co...	0.0000	ms	[0, 1000]	0.0000	immediately
EPOS	p2542	LR standstill window	1000	LU	[0, 2147483647]	1000	immediately
EPOS	p2543	LR standstill monitoring time	200.0000	ms	[0, 100000]	200.0000	immediately
EPOS	p2544	LR positioning window	40	LU	[0, 2147483647]	40	immediately
EPOS	p2545	LR positioning monitoring time	1000.0000	ms	[0, 100000]	1000.0000	immediately
EPOS	p2546	LR dynamic following error monit...	3000	LU	[0, 2147483647]	3000	immediately
EPOS	p2571	EPOS maximum velocity	30000	1000 LU/min	[1, 40000000]	30000	immediately
EPOS	p2572	EPOS maximum acceleration	100	1000 LU/s²	[1, 2000000]	100	immediately
EPOS	p2573	EPOS maximum deceleration	100	1000 LU/s²	[1, 2000000]	100	immediately
EPOS	p2574	EPOS jerk limiting	2000000	1000 LU/s³	[1, 100000000]	2000000	immediately
EPOS	p2575	EPOS jerk limiting activation	0	N.A.	[0, 1]	0	immediately
EPOS	p2580	EPOS software limit switch minus	-2147482648	LU	[-2147482648, ...]	-2147482648	immediately
EPOS	p2581	EPOS software limit switch plus	2147482647	LU	[-2147482648, ...]	2147482647	immediately
EPOS	p2582	EPOS software limit switch activa...	0	N.A.	[0, 1]	0	immediately
EPOS	p2583	EPOS backlash compensation	0	LU	[-200000, 2000...	0	immediately

Figure 9.5 - Appendix 5 – Parameters

Group	Parameter No	Name	Value	Unit	Range	Factory setting	Effect type
EPOS	p2585	EPOS jog 1 setpoint velocity	-300	1000 LU/min	[-40000000, 40...	-300	immediately
EPOS	p2586	EPOS jog 2 setpoint velocity	300	1000 LU/min	[-40000000, 40...	300	immediately
EPOS	p2587	EPOS jog 1 traversing distance	1000	LU	[0, 2147482647]	1000	immediately
EPOS	p2588	EPOS jog 2 traversing distance	1000	LU	[0, 2147482647]	1000	immediately
EPOS	p2599	EPOS reference point coordinat...	0	LU	[-2147482648, ...]	0	immediately
EPOS	p2600	EPOS search for reference refer...	0	LU	[-2147482648, ...]	0	immediately
EPOS	p2604	EPOS search for reference start	0	N.A.	[0, 1]	0	immediately
EPOS	p2605	EPOS search for reference appr...	5000	1000 LU/min	[1, 40000000]	5000	immediately
EPOS	p2606	EPOS search for reference refer...	2147482647	LU	[0, 2147482647]	2147482647	immediately
EPOS	p2608	EPOS search for reference appr...	300	1000 LU/min	[1, 40000000]	300	immediately
EPOS	p2609	EPOS search for reference max	20000	LU	[0, 2147482647]	20000	immediately
EPOS	p2611	EPOS search for reference appr...	300	1000 LU/min	[1, 40000000]	300	immediately
EPOS	p2617[0]	EPOS traversing block position	0	LU	[-2147482648, ...]	0	immediately
EPOS	p2618[0]	EPOS traversing block velocity	600	1000 LU/min	[1, 40000000]	600	immediately
EPOS	p2619[0]	EPOS traversing block accelerati...	100.0000	%	[1, 100]	100.0000	immediately
EPOS	p2620[0]	EPOS traversing deceleration ov...	100.0000	%	[1, 100]	100.0000	immediately
EPOS	p2621[0]	EPOS traversing block task	1 : POSITIONI...	N.A.	--	1	immediately
EPOS	p2622[0]	EPOS traversing block task para...	0	N.A.	[-2147483648, ...]	0	immediately
EPOS	p2623[0]	EPOS traversing block task mode	0	N.A.	[0, 65535]	0	immediately
EPOS	p2634	EPOS fixed stop maximum followi...	1000	LU	[0, 2147482647]	1000	immediately
EPOS	p2635	EPOS fixed stop monitoring wind...	100	LU	[0, 2147482647]	100	immediately
EPOS	p2690	MDI position fixed setpoint	0	LU	[-2147482648, ...]	0	immediately
EPOS	p2691	MDI velocity fixed setpoint	600	1000 LU/min	[1, 40000000]	600	immediately
EPOS	p2692	MDI acceleration override, fixed ...	100.0000	%	[0.1, 100]	100.0000	immediately
EPOS	p2693	MDI deceleration override, fixed ...	100.0000	%	[0.1, 100]	100.0000	immediately

Figure 9.6 - Appendix 6 – Parameters

Com	p8864	PROFIdrive supplementary teleg...	999 : Free tel...	N.A.	--	999	immediately
Com	p8921[0]	PROFIdrive: IP address of station	PROFIdrive supplementary telegram selection		[0, 255]	0	immediately
Com	p8922[0]	PROFIdrive: Default gateway of ...		N.A.	[0, 255]	0	immediately
Com	p8923[0]	PROFIdrive: Subnet mask of stat...		N.A.	[0, 255]	0	immediately
Com	p8925	PROFIdrive: Interface configurati...	0 : No function	N.A.	--	0	reset
Data	p8940	CBE2x Name of Station	0	N.A.	[0, 255]	0	immediately
App	p31581	VIBSUP filter type	0 : Rugged	N.A.	--	0	immediately
App	p31585	VIBSUP frequency fd	1.0000	Hz	[0.5, 3.40282E...	1.0000	immediately
App	p31586	VIBSUP damping	0.0300	N.A.	[0, 0.99]	0.0300	immediately

Figure 9.7 - Appendix 7 – Parameters

1	///FINISHING ROBOT PROGRAM - MITSUBISHI		
2	"Mitsubishi".Program Length := 0;	Mitsubishi	ADB38
3	"Mitsubishi".Load Program Done := 0;	Mitsubishi	ADB38
4	#i := -1;		
5			
6	///MOVE TO HOME POSITION WHEN CIM IS ONLINE AND OUTPUT 'ONLINE' AND 'CAN GET'		
7	#i := #i + 1;"Mitsubishi".Line[#i] := 'DL 1, 2048R';	//delete all line	Mitsubishi
8	#i := #i + 1;"Mitsubishi".Line[#i] := '10 NTSR';	//RETURN TO ORIGIN	Mitsubishi
9	#i := #i + 1;"Mitsubishi".Line[#i] := '15 IDGR';	//INPUT DATA	Mitsubishi
10	#i := #i + 1;"Mitsubishi".Line[#i] := '20 TB +0, 30GR';	//TEST BIT '0' ON	Mitsubishi
11	#i := #i + 1;"Mitsubishi".Line[#i] := '25 GT 15GR';	//LOOP TO LINE 15	Mitsubishi
12			
13	////////////////////////////////////		
14			
15	#i := #i + 1;"Mitsubishi".Line[#i] := '30 SP 5GR';	//SET SPEED TO 50	Mitsubishi
16	#i := #i + 1;"Mitsubishi".Line[#i] := '35 MD 10GR';	//MOVE TO ROBOT P	Mitsubishi
17	#i := #i + 1;"Mitsubishi".Line[#i] := '40 OB -5GR';	//SET UP SLIDE PO	Mitsubishi
18	#i := #i + 1;"Mitsubishi".Line[#i] := '45 OB -9GR';	//POSITION = 000	Mitsubishi
19	#i := #i + 1;"Mitsubishi".Line[#i] := '50 OB -10GR';	//	Mitsubishi
20	#i := #i + 1;"Mitsubishi".Line[#i] := '55 OB -11GR';	//MOVE TO SLIDE P	Mitsubishi
21	#i := #i + 1;"Mitsubishi".Line[#i] := '60 TI 5GR';	//WAIT FOR 0.5 SE	Mitsubishi
22	#i := #i + 1;"Mitsubishi".Line[#i] := '65 IDGR';	//INPUT DATA	Mitsubishi
23	#i := #i + 1;"Mitsubishi".Line[#i] := '70 TB +7, 80GR';	//TEST IF ROBOT I	Mitsubishi
24	#i := #i + 1;"Mitsubishi".Line[#i] := '75 GT 65GR';	//LOOP TO LINE 10	Mitsubishi
25	#i := #i + 1;"Mitsubishi".Line[#i] := '80 OB -11GR';	//RESET MOVE TO S	Mitsubishi
26	#i := #i + 1;"Mitsubishi".Line[#i] := '85 OB +0GR';	//ROBOT "Online"	Mitsubishi
27	#i := #i + 1;"Mitsubishi".Line[#i] := '90 OB +1GR';	//CAN Pick	Mitsubishi
28			
29	///CHECK IF WORK IS REQUIRED VIA INPUT BITS '1','2','4' & '6'		
30			
31	#i := #i + 1;"Mitsubishi".Line[#i] := '100 IDGR';	//INPUT DATA	Mitsubishi
32	#i := #i + 1;"Mitsubishi".Line[#i] := '105 TB +1, 130GR';	//TEST 'Pick FRI	Mitsubishi
33	#i := #i + 1;"Mitsubishi".Line[#i] := '110 TB +2, 145GR';	//TEST 'Pick ROU	Mitsubishi
34	#i := #i + 1;"Mitsubishi".Line[#i] := '115 TB +4, 160GR';	//TEST 'Pick FAL	Mitsubishi
35	#i := #i + 1;"Mitsubishi".Line[#i] := '120 TB +6, 175GR';	//TEST 'Pick FAL	Mitsubishi

Figure 9.8 - Appendix 8 - Mitsi Program 1

36	#i := #i + 1;"Mitsubishi".Line[#i] := '120 ID +0, 170GR';	//TEST 'Pick FAL	Mitsubishi
37	#i := #i + 1;"Mitsubishi".Line[#i] := '125 GT 100GR';	//LOOP BACK TO 1	Mitsubishi
38	#i := #i + 1;"Mitsubishi".Line[#i] := '130 IDGR';	//	Mitsubishi
39	#i := #i + 1;"Mitsubishi".Line[#i] := '135 TB +2, 400GR';	//FINISHED PROCO	Mitsubishi
40	#i := #i + 1;"Mitsubishi".Line[#i] := '140 TB -2, 400GR';	//PRISMATIC PART	Mitsubishi
41	#i := #i + 1;"Mitsubishi".Line[#i] := '145 IDGR';	//	Mitsubishi
42	#i := #i + 1;"Mitsubishi".Line[#i] := '150 TB +1, 600GR';	//FINISHED PROCO	Mitsubishi
43	#i := #i + 1;"Mitsubishi".Line[#i] := '155 TB -1, 200GR';	//ROUND CH1/CH2	Mitsubishi
44	#i := #i + 1;"Mitsubishi".Line[#i] := '160 IDGR';	//	Mitsubishi
45	#i := #i + 1;"Mitsubishi".Line[#i] := '165 TB +6, 1000GR';	//FAIL PRISMATIC	Mitsubishi
46	#i := #i + 1;"Mitsubishi".Line[#i] := '170 TB -6, 800GR';	//FAIL ROUND CH1	Mitsubishi
47	#i := #i + 1;"Mitsubishi".Line[#i] := '175 IDGR';	//	Mitsubishi
48	#i := #i + 1;"Mitsubishi".Line[#i] := '180 TB +4, 1000GR';	//FAIL PRISMATIC	Mitsubishi
49	#i := #i + 1;"Mitsubishi".Line[#i] := '185 TB -4, 800GR';	//FAIL ROUND CH2	Mitsubishi
50			
51	///IF Pick ROUND CH1/CH2		
52	#i := #i + 1;"Mitsubishi".Line[#i] := '200 OB -1GR';	//RESET 'CAN Pic	Mitsubishi
53	#i := #i + 1;"Mitsubishi".Line[#i] := '205 OB +2GR';	//FINISH BUSY	Mitsubishi
54	#i := #i + 1;"Mitsubishi".Line[#i] := '210 OB +8GR';	//SET UP SLIDE P	Mitsubishi
55	#i := #i + 1;"Mitsubishi".Line[#i] := '215 OB -9GR';	//POSITION = 001	Mitsubishi
56	#i := #i + 1;"Mitsubishi".Line[#i] := '220 OB -10GR';	//	Mitsubishi
57	#i := #i + 1;"Mitsubishi".Line[#i] := '225 OB +1GR';	//MOVE TO SLIDE	Mitsubishi
58	#i := #i + 1;"Mitsubishi".Line[#i] := '230 TI 5GR';	//WAIT FOR 0.5 S	Mitsubishi
59	#i := #i + 1;"Mitsubishi".Line[#i] := '235 IDGR';	//INPUT DATA	Mitsubishi
60	#i := #i + 1;"Mitsubishi".Line[#i] := '240 TB +7, 250GR';	//TEST IF ROBOT	Mitsubishi
61	#i := #i + 1;"Mitsubishi".Line[#i] := '245 GT 235GR';	//LOOP TO LINE 2	Mitsubishi
62	#i := #i + 1;"Mitsubishi".Line[#i] := '250 OB -11GR';	//RESET MOVE TO	Mitsubishi
63	#i := #i + 1;"Mitsubishi".Line[#i] := '255 MD 20GR';	//PICK UP ROUND	Mitsubishi
64	#i := #i + 1;"Mitsubishi".Line[#i] := '260 MD 30GR';	//	Mitsubishi
65	#i := #i + 1;"Mitsubishi".Line[#i] := '265 MD 40GR';	//	Mitsubishi
66	#i := #i + 1;"Mitsubishi".Line[#i] := '270 MD 50GR';	//	Mitsubishi
67	#i := #i + 1;"Mitsubishi".Line[#i] := '275 MD 60GR';	//	Mitsubishi
68	#i := #i + 1;"Mitsubishi".Line[#i] := '280 OB -2GR';	//FINISH BUSY	Mitsubishi
69	#i := #i + 1;"Mitsubishi".Line[#i] := '285 OB -8GR';	//SET UP SLIDE P	Mitsubishi

Figure 9.9 - Appendix 9 - Mitsi Program 2

70	## := #1 + 1:"Mitsubishi".Line[#1] := '290 OB +90R';	//POSITION = 010	▶	"Mitsubishi"		ADB30
71	## := #1 + 1:"Mitsubishi".Line[#1] := '295 OB -100R';	//	▶	"Mitsubishi".Line		
72	## := #1 + 1:"Mitsubishi".Line[#1] := '300 OB +110R';	//MOVE TO SLIDE	▶	"Mitsubishi"		ADB30
73	## := #1 + 1:"Mitsubishi".Line[#1] := '305 TI 50R';	//	▶	"Mitsubishi"		ADB30
74	## := #1 + 1:"Mitsubishi".Line[#1] := '310 IDGR';	//INPUT DATA	▶	"Mitsubishi"		ADB30
75	## := #1 + 1:"Mitsubishi".Line[#1] := '315 TB +7, 3250R';	//TEST IF ROBOT	▶	"Mitsubishi"		ADB30
76	## := #1 + 1:"Mitsubishi".Line[#1] := '320 GT 3100R';	//LOOP TO LINE 3	▶	"Mitsubishi"		ADB30
77	## := #1 + 1:"Mitsubishi".Line[#1] := '325 OB -110R';	//RESET MOVE TO	▶	"Mitsubishi"		ADB30
78	## := #1 + 1:"Mitsubishi".Line[#1] := '330 MD 700R';	//PUT DOWN ROUND	▶	"Mitsubishi"		ADB30
79	## := #1 + 1:"Mitsubishi".Line[#1] := '335 MD 000R';	//	▶	"Mitsubishi"		ADB30
80	## := #1 + 1:"Mitsubishi".Line[#1] := '340 MD 900R';	//	▶	"Mitsubishi"		ADB30
81	## := #1 + 1:"Mitsubishi".Line[#1] := '345 MD 1000R';	//	▶	"Mitsubishi"		ADB30
82	## := #1 + 1:"Mitsubishi".Line[#1] := '350 MD 1100R';	//	▶	"Mitsubishi"		ADB30
83	## := #1 + 1:"Mitsubishi".Line[#1] := '355 MD 1200R';	//	▶	"Mitsubishi"		ADB30
84	## := #1 + 1:"Mitsubishi".Line[#1] := '360 OB +60R';	//TELL CIM TO PR	▶	"Mitsubishi"		ADB30
85	## := #1 + 1:"Mitsubishi".Line[#1] := '365 GT 150R';	//	▶	"Mitsubishi"		ADB30
86						
87	//IF Pick PRISMATIC PART					
88						
89	## := #1 + 1:"Mitsubishi".Line[#1] := '400 OB -10R';	//RESET 'CAN Pic	▶	"Mitsubishi"		ADB30
90	## := #1 + 1:"Mitsubishi".Line[#1] := '405 OB +20R';	//FINISH BUSY	▶	"Mitsubishi"		ADB30
91	## := #1 + 1:"Mitsubishi".Line[#1] := '410 OB +80R';	//SET UP SLIDE P	▶	"Mitsubishi"		ADB30
92	## := #1 + 1:"Mitsubishi".Line[#1] := '415 OB +90R';	//POSITION = 011	▶	"Mitsubishi"		ADB30
93	## := #1 + 1:"Mitsubishi".Line[#1] := '420 OB -100R';	//	▶	"Mitsubishi"		ADB30
94	## := #1 + 1:"Mitsubishi".Line[#1] := '425 OB +110R';	//MOVE TO SLIDE	▶	"Mitsubishi"		ADB30
95	## := #1 + 1:"Mitsubishi".Line[#1] := '430 TI 50R';	//	▶	"Mitsubishi"		ADB30
96	## := #1 + 1:"Mitsubishi".Line[#1] := '435 IDGR';	//INPUT DATA	▶	"Mitsubishi"		ADB30
97	## := #1 + 1:"Mitsubishi".Line[#1] := '440 TB +7, 4500R';	//TEST IF ROBOT	▶	"Mitsubishi"		ADB30
98	## := #1 + 1:"Mitsubishi".Line[#1] := '445 GT 4350R';	//LOOP TO LINE 4	▶	"Mitsubishi"		ADB30
99	## := #1 + 1:"Mitsubishi".Line[#1] := '450 OB -110R';	//RESET MOVE TO	▶	"Mitsubishi"		ADB30
100	## := #1 + 1:"Mitsubishi".Line[#1] := '455 MD 2000R';	//PICK UP PRISMA	▶	"Mitsubishi"		ADB30
101	## := #1 + 1:"Mitsubishi".Line[#1] := '460 MD 2100R';	//	▶	"Mitsubishi"		ADB30
102	## := #1 + 1:"Mitsubishi".Line[#1] := '465 MD 2200R';	//	▶	"Mitsubishi"		ADB30
103	## := #1 + 1:"Mitsubishi".Line[#1] := '470 MD 2300R';	//	▶	"Mitsubishi"		ADB30
104	## := #1 + 1:"Mitsubishi".Line[#1] := '475 MD 2400R';	//	▶	"Mitsubishi"		ADB30

Figure 9.10 - Appendix 9 - Mitsi Program 3

105	## := #1 + 1:"Mitsubishi".Line[#1] := '480 OB -20R';	//FINISH "Busy"	▶	"Mitsubishi"		ADB30
106	## := #1 + 1:"Mitsubishi".Line[#1] := '485 OB -80R';	//SET UP SLIDE "	▶	"Mitsubishi"		ADB30
107	## := #1 + 1:"Mitsubishi".Line[#1] := '490 OB +90R';	//	▶	"Mitsubishi"		ADB30
108	## := #1 + 1:"Mitsubishi".Line[#1] := '495 OB -100R';	//	▶	"Mitsubishi".Line		ADB30
109	## := #1 + 1:"Mitsubishi".Line[#1] := '500 OB +110R';	//MOVE TO SLIDE	▶	"Mitsubishi"		ADB30
110	## := #1 + 1:"Mitsubishi".Line[#1] := '505 TI 50R';	//	▶	"Mitsubishi"		ADB30
111	## := #1 + 1:"Mitsubishi".Line[#1] := '510 IDGR';	//INPUT DATA	▶	"Mitsubishi"		ADB30
112	## := #1 + 1:"Mitsubishi".Line[#1] := '515 TB +7, 5250R';	//TEST IF ROBOT	▶	"Mitsubishi"		ADB30
113	## := #1 + 1:"Mitsubishi".Line[#1] := '520 GT 5100R';	//LOOP TO LINE 5	▶	"Mitsubishi"		ADB30
114	## := #1 + 1:"Mitsubishi".Line[#1] := '525 OB -110R';	//RESET MOVE TO	▶	"Mitsubishi"		ADB30
115	## := #1 + 1:"Mitsubishi".Line[#1] := '530 MD 2500R';	//PUT DOWN PRISM	▶	"Mitsubishi"		ADB30
116	## := #1 + 1:"Mitsubishi".Line[#1] := '535 MD 2600R';	//	▶	"Mitsubishi"		ADB30
117	## := #1 + 1:"Mitsubishi".Line[#1] := '540 MD 2700R';	//	▶	"Mitsubishi"		ADB30
118	## := #1 + 1:"Mitsubishi".Line[#1] := '545 MD 2800R';	//	▶	"Mitsubishi"		ADB30
119	## := #1 + 1:"Mitsubishi".Line[#1] := '550 MD 2900R';	//	▶	"Mitsubishi"		ADB30
120	## := #1 + 1:"Mitsubishi".Line[#1] := '555 OB +40R';	//TELL CIM STAMP	▶	"Mitsubishi"		ADB30
121	## := #1 + 1:"Mitsubishi".Line[#1] := '560 GT 150R';	//	▶	"Mitsubishi"		ADB30
122						
123	//IF Pick FINISHED PRODUCT					
124						
125	## := #1 + 1:"Mitsubishi".Line[#1] := '600 OB -10R';	//RESET 'CAN Pic	▶	"Mitsubishi"		ADB30
126	## := #1 + 1:"Mitsubishi".Line[#1] := '605 OB +20R';	//FINISH BUSY	▶	"Mitsubishi"		ADB30
127	## := #1 + 1:"Mitsubishi".Line[#1] := '610 OB -80R';	//SET UP SLIDE "	▶	"Mitsubishi"		ADB30
128	## := #1 + 1:"Mitsubishi".Line[#1] := '615 OB +90R';	//POSITION = 010	▶	"Mitsubishi"		ADB30
129	## := #1 + 1:"Mitsubishi".Line[#1] := '620 OB -100R';	//	▶	"Mitsubishi"		ADB30
130	## := #1 + 1:"Mitsubishi".Line[#1] := '625 OB +110R';	//MOVE TO SLIDE	▶	"Mitsubishi"		ADB30
131	## := #1 + 1:"Mitsubishi".Line[#1] := '630 TI 50R';	//	▶	"Mitsubishi"		ADB30
132	## := #1 + 1:"Mitsubishi".Line[#1] := '635 IDGR';	//INPUT DATA	▶	"Mitsubishi"		ADB30
133	## := #1 + 1:"Mitsubishi".Line[#1] := '640 TB +7, 6500R';	//TEST IF ROBOT	▶	"Mitsubishi"		ADB30
134	## := #1 + 1:"Mitsubishi".Line[#1] := '645 GT 6350R';	//LOOP TO LINE 6	▶	"Mitsubishi"		ADB30
135	## := #1 + 1:"Mitsubishi".Line[#1] := '650 OB -110R';	//	▶	"Mitsubishi"		ADB30
136	## := #1 + 1:"Mitsubishi".Line[#1] := '655 MD 3200R';	//PICK UP FINISH	▶	"Mitsubishi"		ADB30
137	## := #1 + 1:"Mitsubishi".Line[#1] := '660 MD 3300R';	//	▶	"Mitsubishi"		ADB30
138	## := #1 + 1:"Mitsubishi".Line[#1] := '665 MD 3400R';	//	▶	"Mitsubishi"		ADB30
139	## := #1 + 1:"Mitsubishi".Line[#1] := '670 MD 3500R';	//	▶	"Mitsubishi"		ADB30

Figure 9.11 - Appendix 10 - Mitsi Program 4


```

206 #I := #I + 1; "Mitsubishi".Line[#I] := '1025 OB +110R'; //MOVE TO SLIDE  "Mitsubishi"  VDB38
207 #I := #I + 1; "Mitsubishi".Line[#I] := '1030 TI 50R'; // "Mitsubishi"  VDB38
208 #I := #I + 1; "Mitsubishi".Line[#I] := '1035 IDGR'; //INPUT DATA  "Mitsubishi"  VDB38
209 #I := #I + 1; "Mitsubishi".Line[#I] := '1040 TB +7, 1050GR'; //TEST IF ROBOT  "Mitsubishi"  VDB38
210 #I := #I + 1; "Mitsubishi".Line[#I] := '1045 GT 1035GR'; //LOOP TO LINE 1  "Mitsubishi"  VDB38
211 #I := #I + 1; "Mitsubishi".Line[#I] := '1050 OB -110R'; //RESET MOVE TO  "Mitsubishi"  VDB38
212 #I := #I + 1; "Mitsubishi".Line[#I] := '1055 MD 200GR'; //PICK UP FAILED  "Mitsubishi"  VDB38
213 #I := #I + 1; "Mitsubishi".Line[#I] := '1060 MD 210GR'; // "Mitsubishi"  VDB38
214 #I := #I + 1; "Mitsubishi".Line[#I] := '1065 MD 220GR'; // "Mitsubishi"  VDB38
215 #I := #I + 1; "Mitsubishi".Line[#I] := '1070 MD 230GR'; // "Mitsubishi"  VDB38
216 #I := #I + 1; "Mitsubishi".Line[#I] := '1075 MD 240GR'; // "Mitsubishi"  VDB38
217 #I := #I + 1; "Mitsubishi".Line[#I] := '1080 OB +80R'; //SET UP SLIDE P  "Mitsubishi"  VDB38
218 #I := #I + 1; "Mitsubishi".Line[#I] := '1085 OB -90R'; // "Mitsubishi"  VDB38
219 #I := #I + 1; "Mitsubishi".Line[#I] := '1090 OB +10GR'; // "Mitsubishi"  VDB38
220 #I := #I + 1; "Mitsubishi".Line[#I] := '1095 OB +110R'; //MOVE TO SLIDE  "Mitsubishi"  VDB38
221 #I := #I + 1; "Mitsubishi".Line[#I] := '1100 TI 50R'; //WAIT FOR 0.5 S  "Mitsubishi"  VDB38
222 #I := #I + 1; "Mitsubishi".Line[#I] := '1105 IDGR'; //INPUT DATA  "Mitsubishi"  VDB38
223 #I := #I + 1; "Mitsubishi".Line[#I] := '1110 TB +7, 1120GR'; //TEST IF ROBOT  "Mitsubishi"  VDB38
224 #I := #I + 1; "Mitsubishi".Line[#I] := '1115 GT 1105GR'; //LOOP TO LINE 1  "Mitsubishi"  VDB38
225 #I := #I + 1; "Mitsubishi".Line[#I] := '1120 OB -110R'; //RESET MOVE TO  "Mitsubishi"  VDB38
226 #I := #I + 1; "Mitsubishi".Line[#I] := '1125 MD 450GR'; // "Mitsubishi"  VDB38
227 #I := #I + 1; "Mitsubishi".Line[#I] := '1130 MD 460GR'; // "Mitsubishi"  VDB38
228 #I := #I + 1; "Mitsubishi".Line[#I] := '1135 OB -20R'; //FINISH BUSY  "Mitsubishi"  VDB38
229 #I := #I + 1; "Mitsubishi".Line[#I] := '1140 GT 150R'; // "Mitsubishi"  VDB38
230
231 #I := #I + 1; "Mitsubishi".Line[#I] := 'RGR'; // "Mitsubishi"  VDB38
232
233
234 "Mitsubishi".Program Length := #I; // "Mitsubishi"  VDB38
235 "Mitsubishi".Load Program Done := 1; // "Mitsubishi"  VDB38
236
237 //////////////////////////////////////////////////
238

```

Figure 9.14 - Appendix 13 - Mitsi Program 7

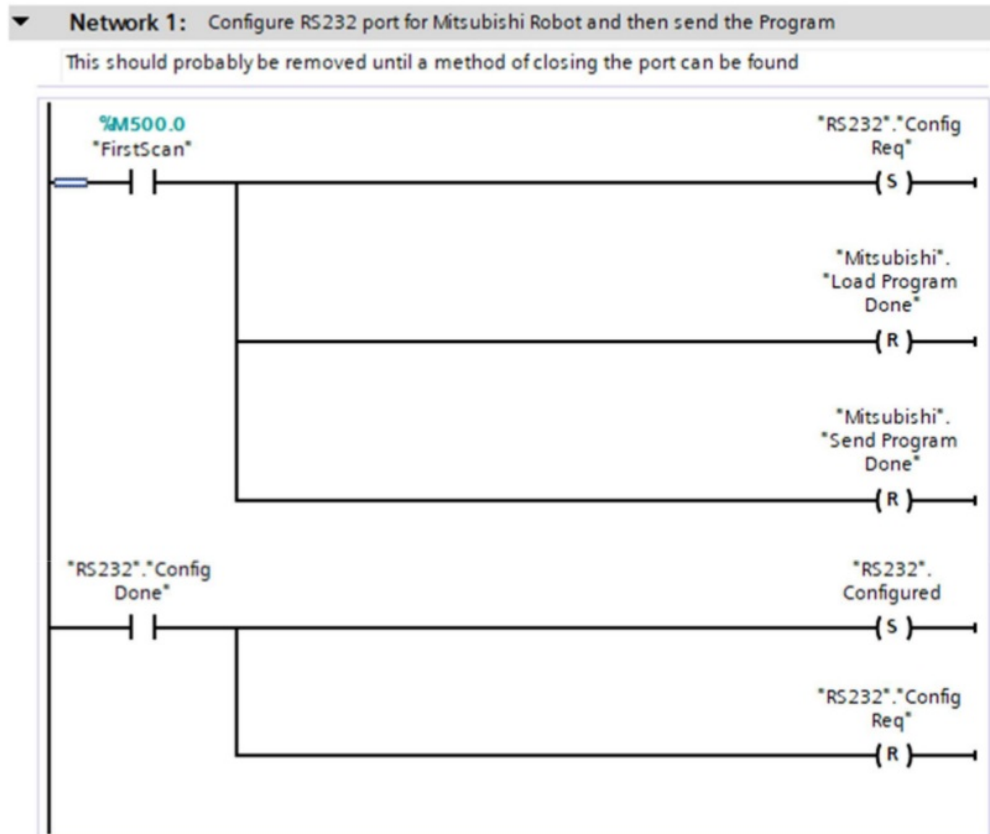


Figure 9.15 - Appendix 14 - RS232 Program 1

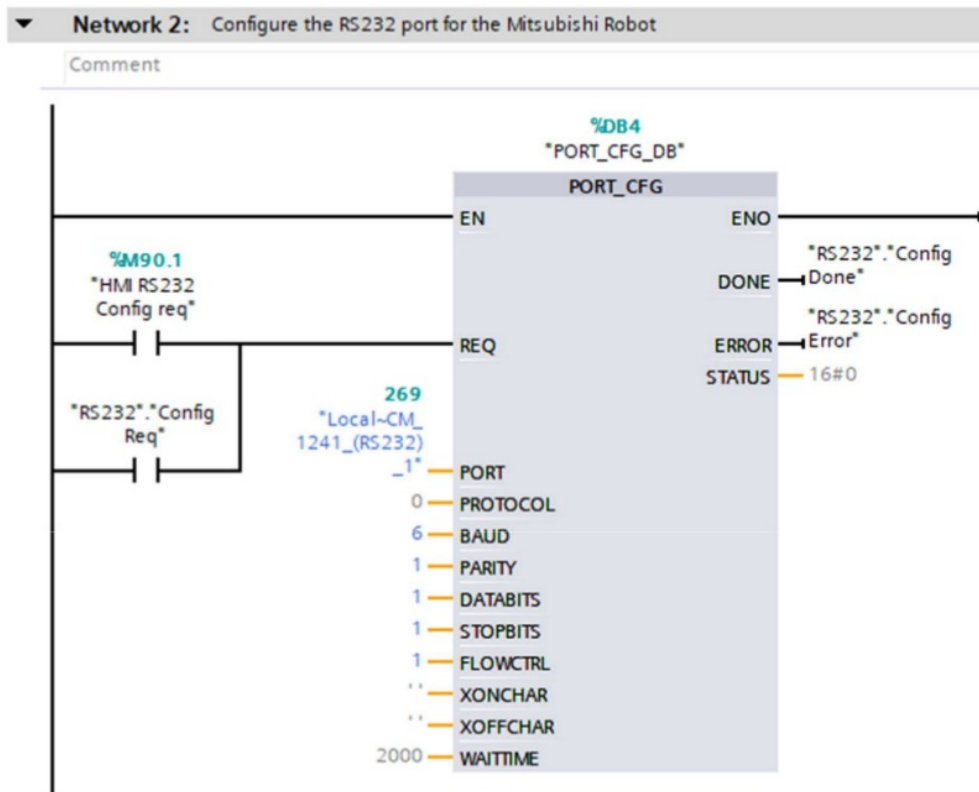


Figure 9.16 - Appendix 15 - RS232 Program 2

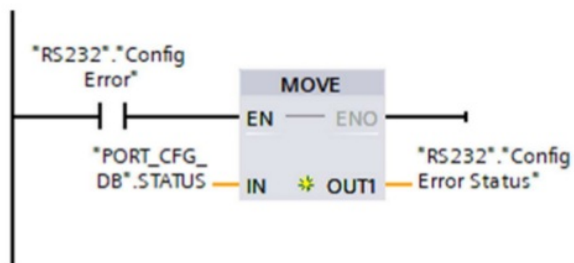


Figure 9.17 - Appendix 16 - RS232 Program 3

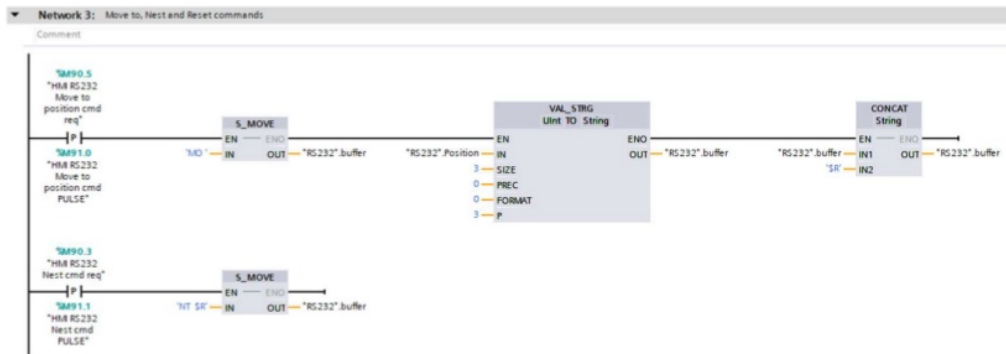


Figure 9.18 - Appendix 17 - RS232 Program 4

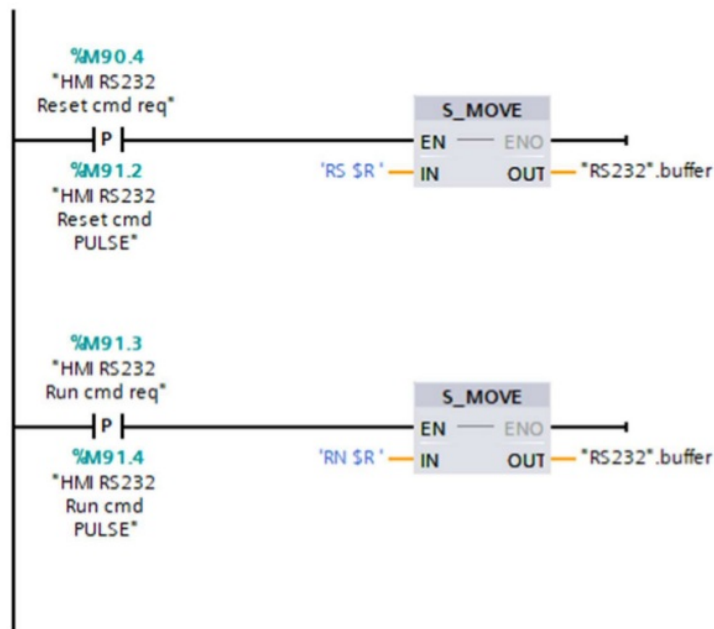


Figure 9.19 - Appendix 18 - RS232 Program 5

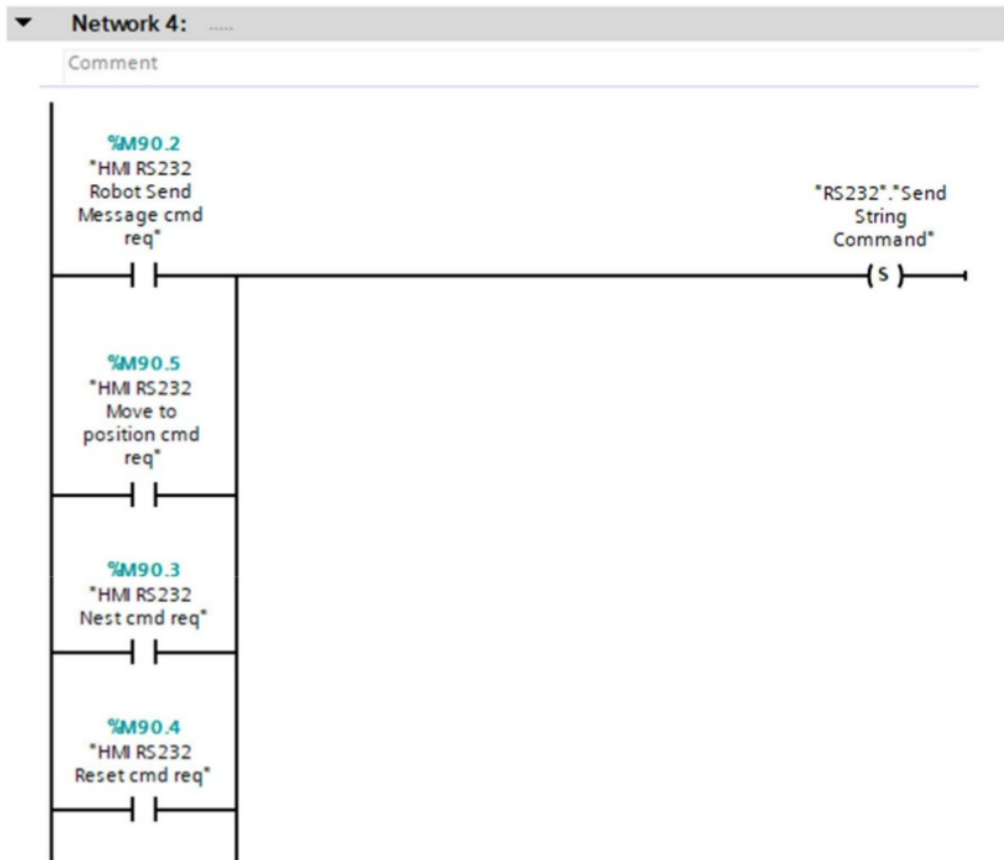


Figure 9.20 - Appendix 19 - RS232 Program 6



Figure 9.21 - Appendix 20 - RS232 Program 7

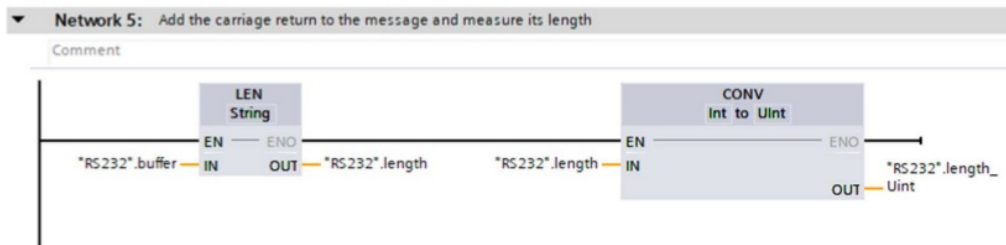


Figure 9.22 - Appendix 21 - RS232 Program 8

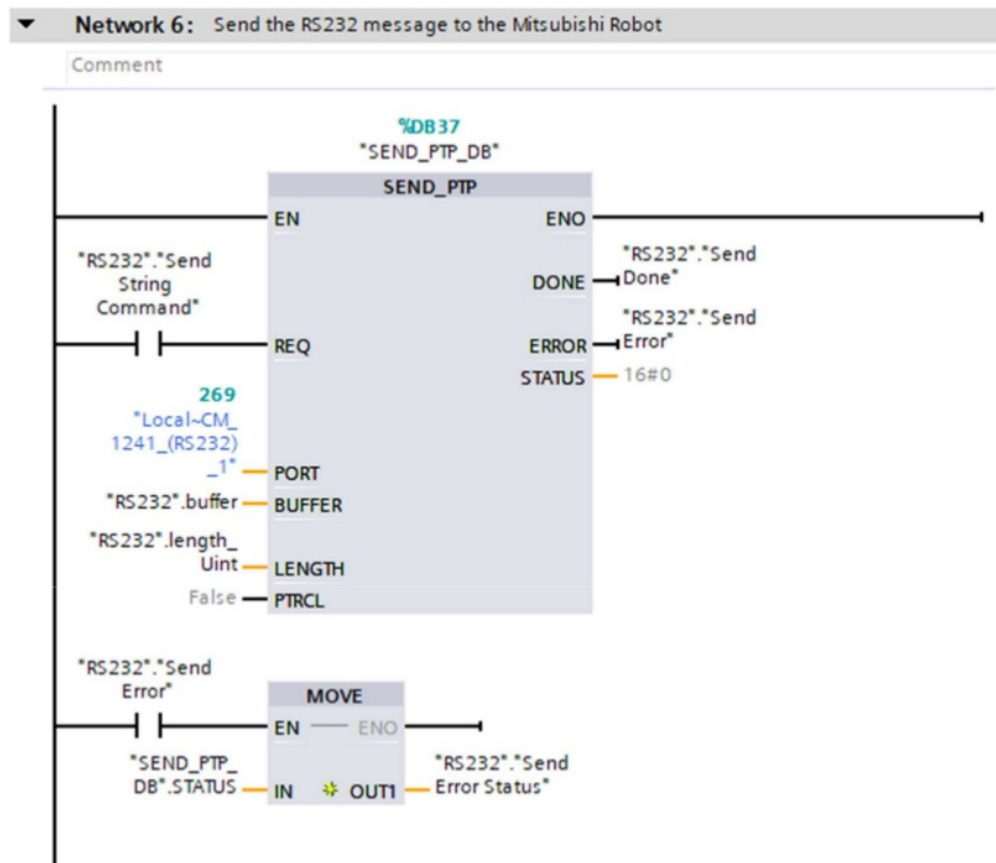


Figure 9.23 - Appendix 22 - RS232 Program 9

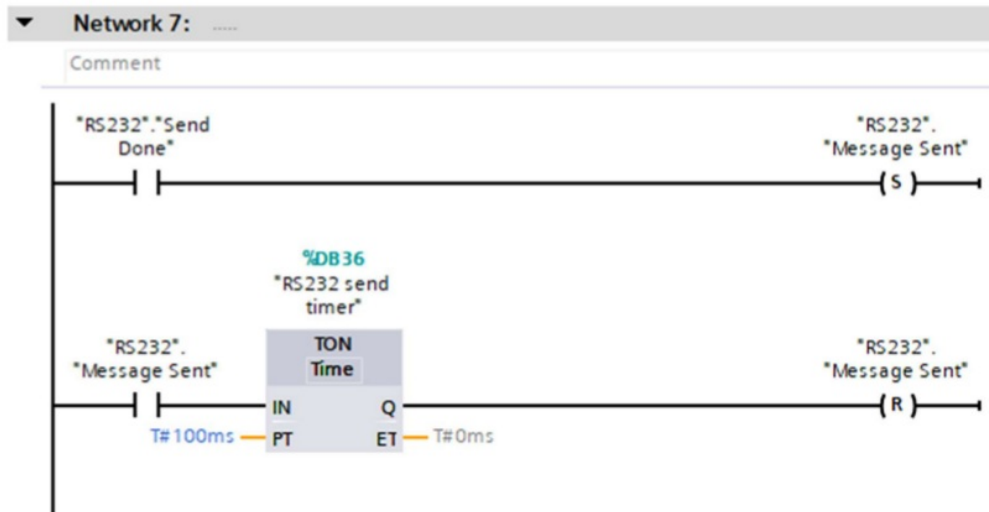


Figure 9.24 - Appendix 23 - RS232 Program 9

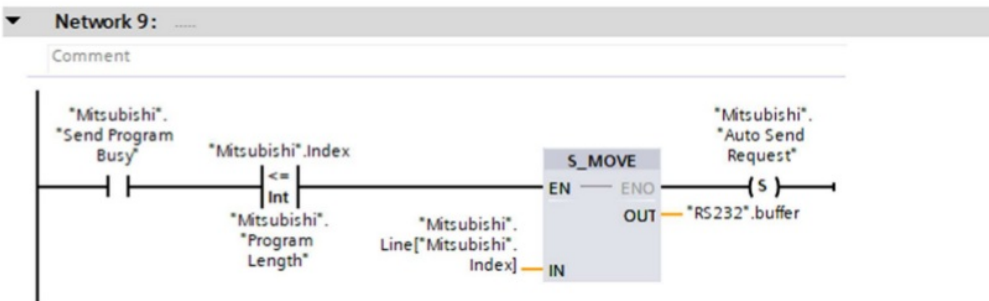
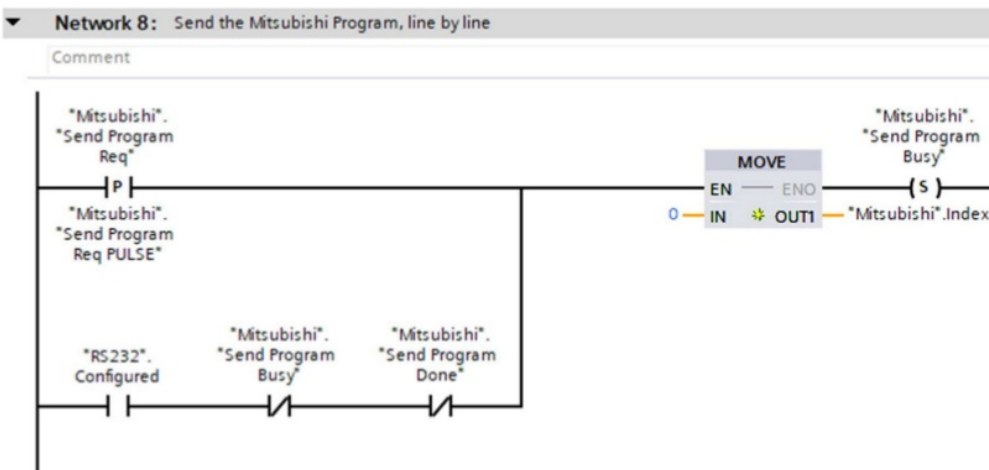


Figure 9.25 - Appendix 24 - RS232 Program 10

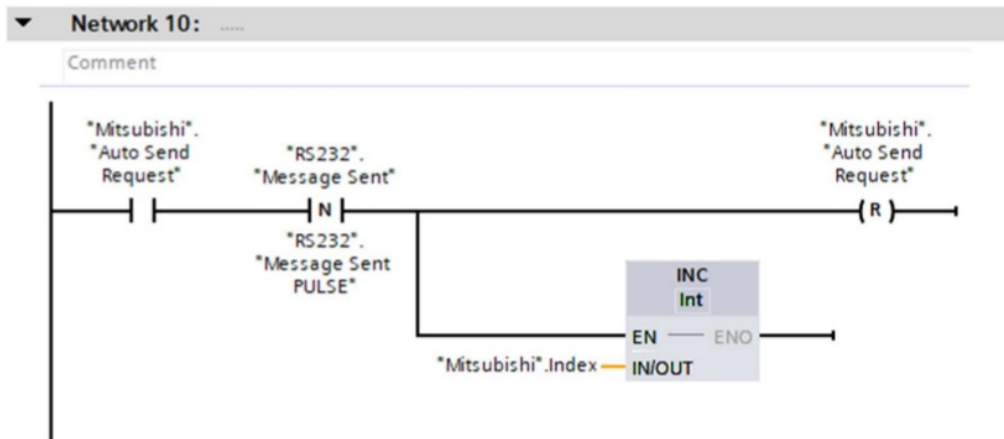


Figure 9.26 - Appendix 25 - RS232 Program 11

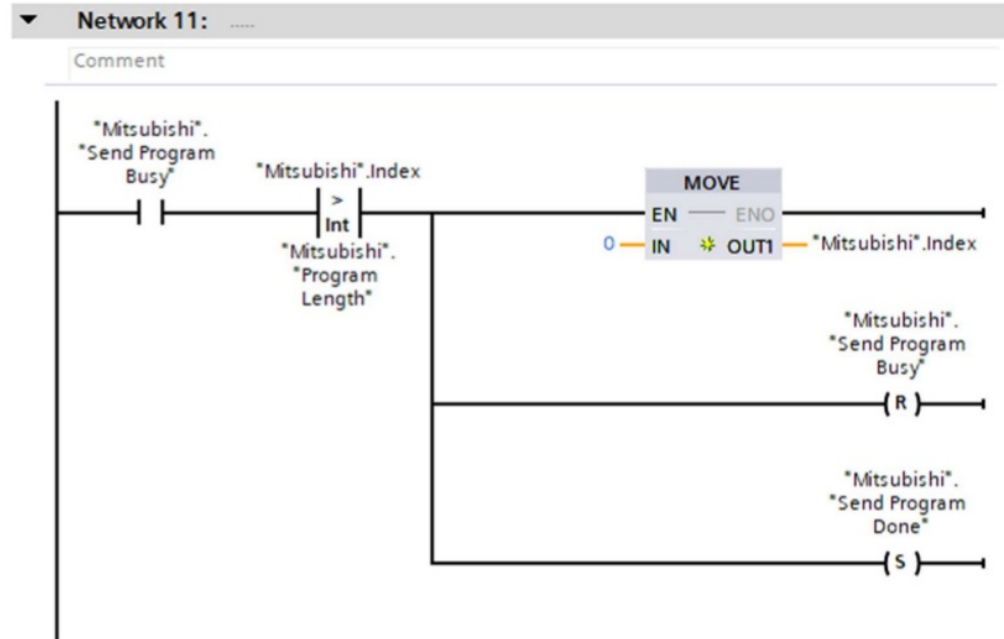


Figure 9.27 - Appendix 26 - RS232 Program 12

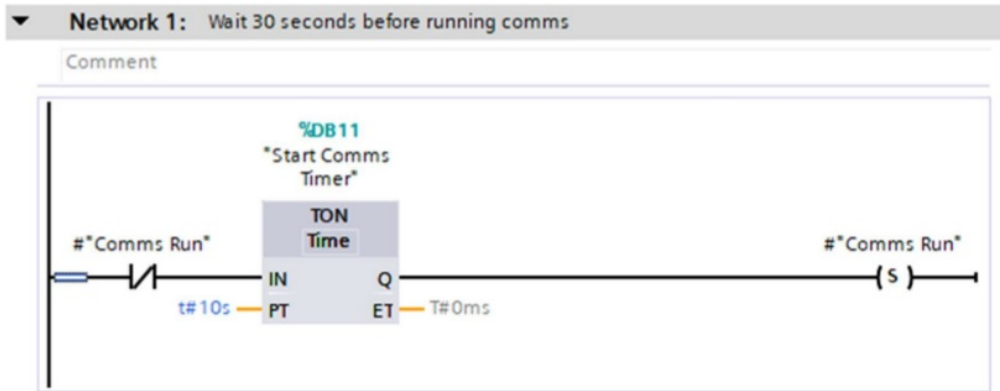


Figure 9.28 - Appendix 27 - GET and PUT Program 1

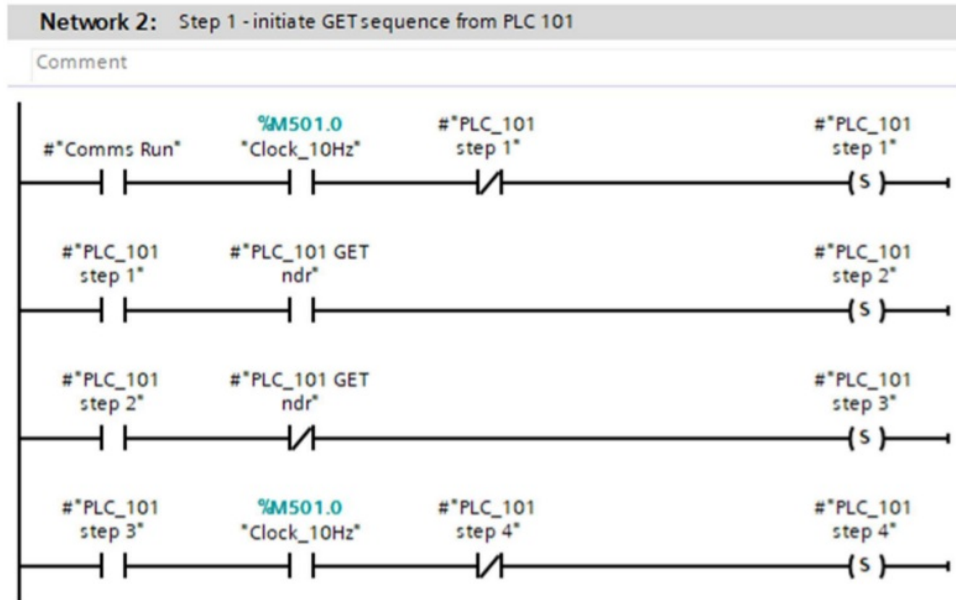


Figure 9.29 - Appendix 28 - GET and PUT Program 2

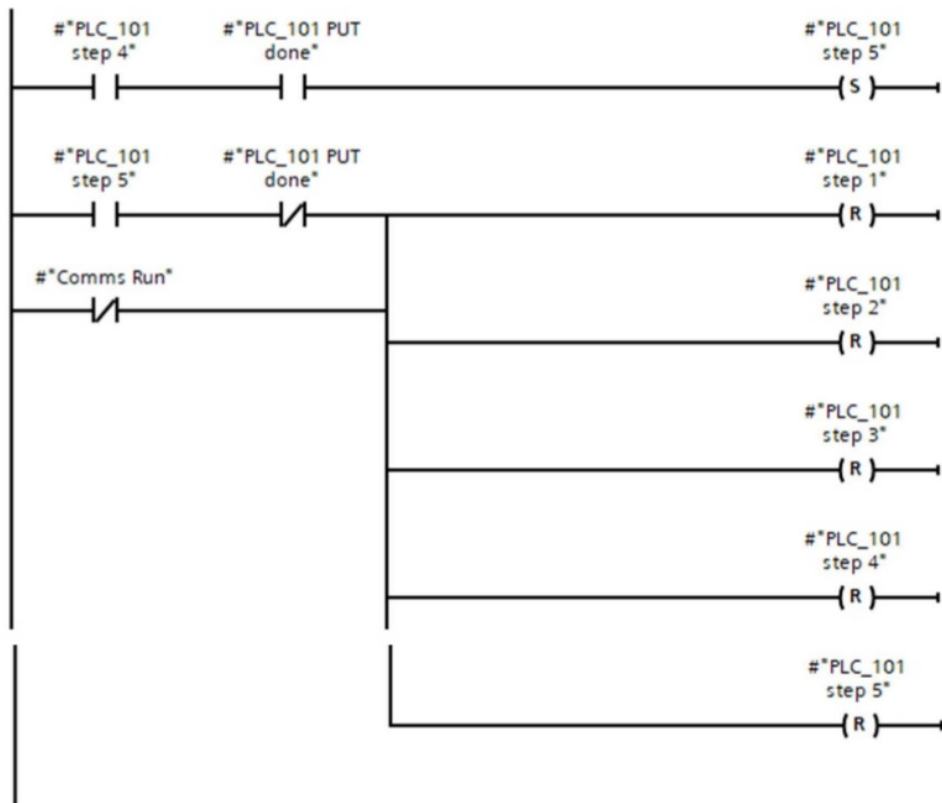


Figure 9.30 - Appendix 29 - GET and PUT Program 3

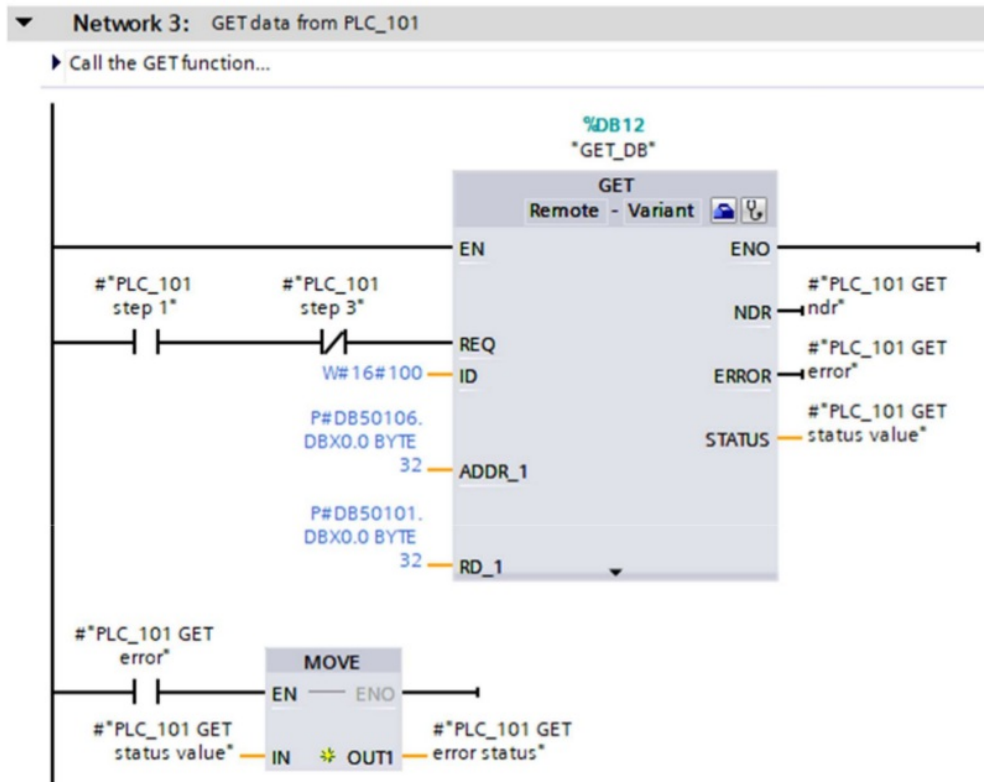


Figure 9.31 - Appendix 30 - GET and PUT Program 4

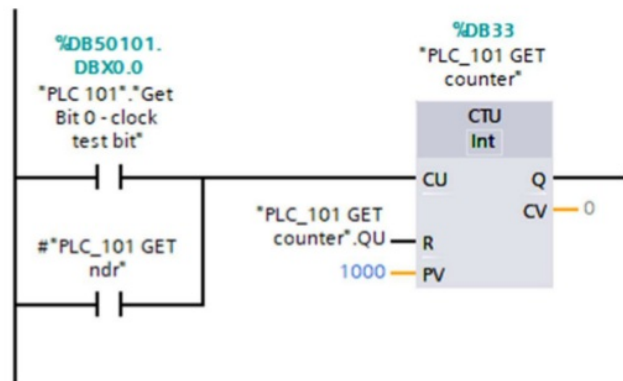


Figure 9.32 - Appendix 31 - GET and PUT Program 5

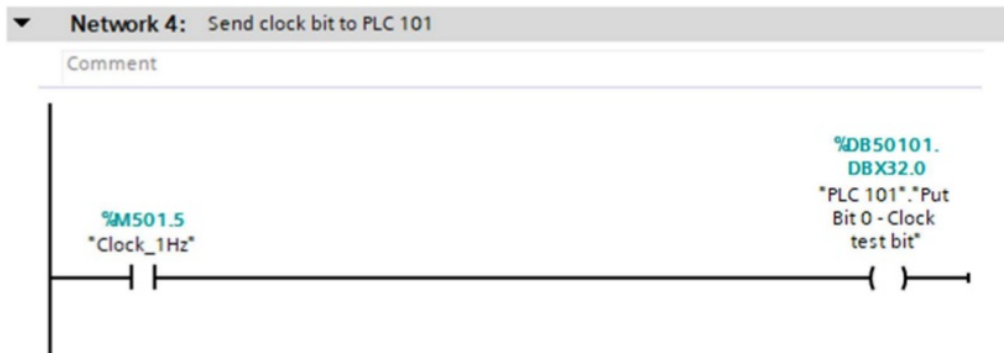


Figure 9.33 - Appendix 32 - GET and PUT Program 6

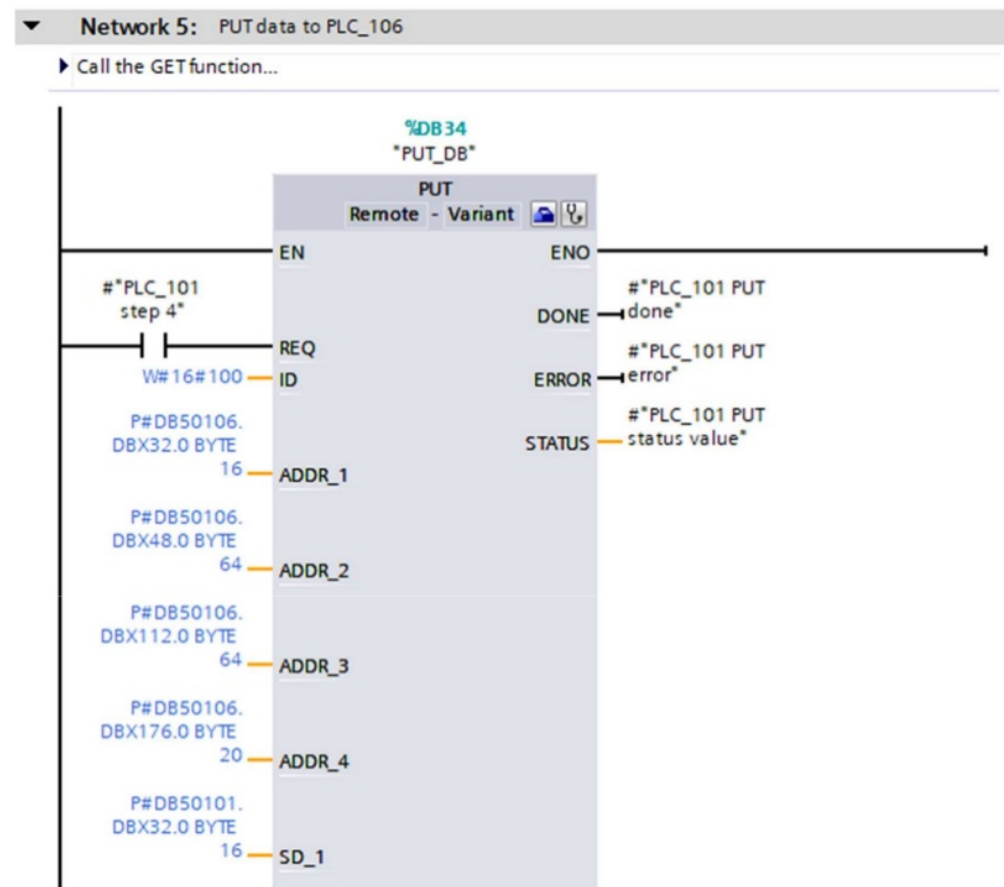


Figure 9.34 - Appendix 33 - GET and PUT Program 7

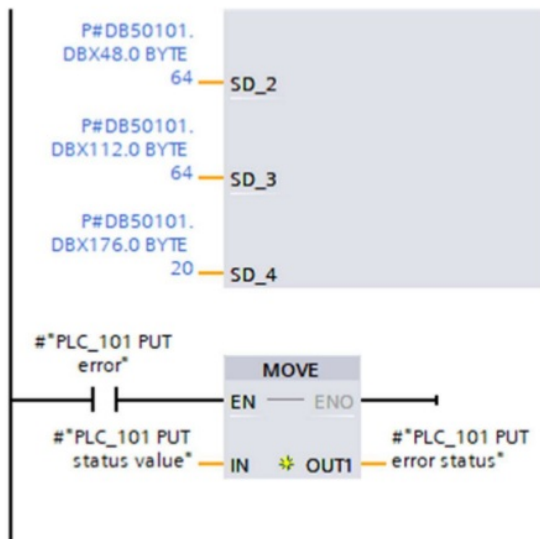


Figure 9.35 - Appendix 34 - GET and PUT Program 8

FINAL GRADE

GENERAL COMMENTS

Instructor

66/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

Text Comment. You are summarising the report rather than the project.

Text Comment. The panel already existed so no further cost was needed.

PAGE 8

Text Comment. You should be introducing the project not the report. Aims and Objectives?

Text Comment. You did not need to include this last paragraph in the introduction

PAGE 9

PAGE 10

PAGE 11

PAGE 12

Text Comment. Photo could have been taken back in January at the time of build, should be in the log book.

Text Comment. You should be writing about what was done rather than instructing your reader in what they should do. "has been added" is better than "must be added".

Text Comment. There are other discrete signals between these two.

Text Comment. RS232 comms is in the other direction

necessary, refer to self as "the author".

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

Text Comment. Your program has 5 positions

Text Comment. The log book should have been kept up to date at the time of testing, then the evidence would have been available

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

Text Comment. A number of important goals have been met.

Text Comment. This belongs in assignment 1, your literature review. Your solution is good because it is easy to integrate into the CIM and provides the required performance.

Text Comment. This list cannot contain the same sources as the Citation list

GRADING FORM: NEW CSG-RUBRIC

**DAVID
HARRIES****66****1. ENGAGEMENT WITH LITERATURE**

Judicial use and critical evaluation of technical literature in the design of the project: Clear use has been made of literature in the development of the solution to a high technical standard. Extensive use of Siemens' online technical literature and resources. (4/5) Extent of reading, range of credible sources and use of Harvard method: A small range of credible and appropriate sources has been consulted with correct use of Harvard. Good use of credible and reliable sources throughout the report. Citation list and bibliography appear to be identical? (2/5)
Total = 6/10

6**2. KNOWLEDGE AND UNDERSTANDING**

Advanced knowledge and critical understanding demonstrated in the design of the project: The solution, though not complete due to restrictions, is excellent and could be easily integrated into the CIM

8

system. The programs are well designed and explained. The report demonstrates a very good critical understanding of the principles required to implement the project successfully. Total = 8/10

3. COGNITIVE AND INTELLECTUAL



Testing and analysis of result: Good evidence of testing in some parts under the circumstances, particularly between the servo and the PLC. However, some parts were tested before lockdown and records should have been kept up to date in the log book. Testing appears to be incomplete, primarily due to Covid-19 lockdown restrictions. (6/10) Critical evaluation of the project outcomes: A good honest evaluation has been carried out what has been achieved is more significant than the report suggests. Outcomes that were achieved were compared against the FDS, other outcomes were theoretically assessed. Budget was forecast to be exceeded. (6/10) Good conclusions presented with consideration given to the original plan. However, the recommendation of using a different drive belongs to the literature review in assignment 1 and it is not appropriate to mention it here. Consideration was given for “always expect the unexpected” with regards to risks to a project and the need f (7/10)
Total = 19/30

19

4. PRACTICAL APPLICATION



Formulation of innovative and creative design solution to the project problem: Excellent solution

23

developed that suits the application perfectly though not fully developed due to circumstances. (8/10) Deployment method, use of material, tools, equipment and techniques in the construction of the project: Good use of software and hardware to develop the solution though the panel was not built due to constraints of Covid-19. (4/5) Quality of the final product or system: The final product is of a high quality – hardware is industrial standard, software screens are easy to read and understand. No panel built. (6/10) Use of test equipment and presentation of results of the project : Good use of the available equipment, results are well presented. (3/5) Management of the design process. Clear evidence presented of the design process that has been managed effectively but the final Gantt chart is missing that would show a comparison between planned and actual performance. (2/5) Total = 23/35

5. TRANSFERABLE SKILLS



Executive summary, acknowledgements and introduction: The summary is more of an introduction to the report, no acknowledgements given and the introduction should be more clear about the aims and objectives of the project. (1/5) Quality of the report: High quality report, well written and presented. (4/5) Word count: Word count declared as 4626, Within limits, no penalty. (5/5) Total = 10/15

10