2224559

by Mohammed Ibrahim

Submission date: 30-Jan-2023 02:19PM (UTC+0000)
Submission ID: 196430305
File name: 19116_Mohammed_Ibrahim_2224559_180905_595680319.pdf (5.61M)
Word count: 6304
Character count: 39144

Course:	HNC Electrical and Electronic Engineering
Module:	SGEE4004 Programmable Logic Controllers
Assignment: 1.	Pick and Place Robot
Name:	Mohammed Zeeshan Ibrahim
University number:	2224559
Date submitted:	30/01/2023
Word count:	2429 Words

Table of Contents

Good Structure

List of Figures 4
1 - Work Plan for Lifelong Learning 6
Assignment Approach: 6
What has been learned?....6
2 - Research 7
Allen Bradley Programmable Logic Controller, MicroLogix DIN Rail Mount, 120 to 240V AC Supply Voltage – 1766-L32BWA 7
Allen Bradley - CompactLogix 5380 Controller, 600KB, 8 I/Os, 16 nodes, Standard - 5069-L306ER 8
Allen Bradley - CompactLogix™ 5370 L2 Programmable Automation Controllers - 1769-L24ER-QB1B....9
Comparison table 10
Best Allen Bradley PLC alternative for the robot 11
3 - Hardware Investigation and Input-Output List 12
4 - Program Design and Explanation....14
Main Program Block: 14
Automatic Sequence Block:....15
Operator's Screen: 15
Engineer's Screen: 15
5 - Simulation Testing 16
6 - Practical testing 17
7 - Discussion, Conclusions and Recommendations 18
Discussion:....18
Conclusion:....18
Recommendations:....19
8 - Team Working 20
9 - Citation List....21
10 - Appendices....23
Appendix A: - Program Design 23

Appendix B: - Program Design, Automatic Control 28

Appendix C: - Program Design, Networks, Including Manual Control and Interlocks.....33

Appendix D: - Simulation Testing Results and Evidence.....40

Appendix E: - Practical Testing Results and Evidence.....56

List of Figures

Figure 1 Allen Bradley Programmable Logic Controller MicroLogix, DIN Rail Mount, 120 to 240V AC Supply Voltage - 1766-L32BWA (RS Components Ltd., n.d.) 7
Figure 2 CompactLogix 5380 Controller, 600KB, 8 I/Os, 16 nodes, Standard (Rockwell Automation Ltd., 2022) 8
Figure 3 Allen Bradley - CompactLogix™ 5370 L2 Programmable Automation Controllers - 1769-L24ER-QB1B (Routeco Limited, 2022) 9
Figure 4 PLC Controlled Robot Hardware 12
Figure 5 Pushbuttons and Rocker Switches 12
Figure 6 Ethernet Address Configuration 23
Figure 7 PLC Tag List 1 - 40 24
Figure 8 PLC Tag List 41 - 59 25
Figure 9 Operator's Screen 26
Figure 10 Engineer's Screen 27
Figure 11 Automatic Sequential Function Chart 28
Figure 12 Automatic Sequence SCL Code 29
Figure 13 Continued SCL Code 30
Figure 14 Completed SCL Code 32
Figure 15 Networks 1 & 2 33
Figure 16 Networks 3 & 4 34
Figure 17 Networks 5 & 6 35
Figure 18 Networks 7 & 8 36
Figure 19 Networks 9 & 10 37
Figure 20 Networks 11 & 12 38
Figure 21 Network 13 39
Figure 22 Test 1- Down Sensor Interlock 42
Figure 23 Test 2 - In Control 43
Figure 24 Test 2 - Out Control 43
Figure 25 Test 2 - Left Control 44
Figure 26 Test 2 - Right Control 44
Figure 27 Test 3 - Out of Home Position 45
Figure 28 Test 3 - Robot in Home Position After Pressing 'Auto' 45
Figure 29 Test 4 - Beginning of Sequence 46

Figure 30 Test 4 - End of Sequence.....47
Figure 31 Test 4 - Robot Returned to Home Position.....47
Figure 32 Test 5 - Beginning of Sequence.....48
Figure 34 Test 5 - Robot Returned to Home Position.....49
Figure 33 Test 5 - End of Sequence.....49
Figure 35 Test 6 - Beginning of Sequence.....50
Figure 37 Test 6 - Robot Returned to Home Position.....51
Figure 36 Test 6 - End of Sequence.....51
Figure 38 Test 7 - Sequence Paused.....52
Figure 39 Test 7 - Sequence Continued.....52
Figure 40 Test 8 - Timer 2 Second Delay.....53
Figure 41 Test 9 - Beginning of Sequence, Counter = 2.....54
Figure 42 Test 9 - Step 8, Counter Decremented to 1.....54
Figure 43 Test 9 - Step 26, Counter Incremented to 2.....55
Figure 44 Test 1 – Testing Down Sensor Interlock.....58
Figure 45 Test 2 - Right Manual Request.....59
Figure 46 Test 3 - Robot Away from Home Position and in Manual Mode.....60
Figure 47 Test 3 - Robot in The Home Position After Pressing 'Auto'.....61
Figure 48 Test 4 - 2 Objects in The Machine Pre-Sequence.....62
Figure 49 Test 4 - 2 Objects in The Machine Post-Sequence.....63
Figure 50 Test 5 - 1 Object in The Machine Pre-Sequence.....64
Figure 51 Test 5 - Two Objects in The Machine Post-Sequence.....65
Figure 52 Test 6 - No Objects in The Machine Pre-Sequence.....66
Figure 53 Test 6 - 1 Object in The Machine Post-Sequence.....67
Figure 54 Test 7 - Sequence Paused After Pressing ‘Pause’ Button.....68
Figure 55 Test 7 - Sequence Continued After Pressing 'Continue' Button.....69
Figure 56 Test 8 - Timer Delay with HMI Displaying Timer.....70
Figure 57 Test 9 - Object Counter Reading 2 Objects Before Step 8.....71
Figure 58 Test 9 - Object Counter Decremented After Step 8.....72
Figure 59 Test 9 - Object Counter Incremented After Step 26.....73

Table 1 PLC Comparison.....10
Table 2 Simulation Testing and Results.....40
Table 3 Practical Testing And Results.....56

1 - Work Plan for Lifelong Learning

Assignment Approach:

The assignment will first be approached by carrying out research. Different types of PLCs will be compared with each other and determined which is most suitable for robot control. A pick and place PLC controlled robot will then be designed and developed in a team. The hardware will then be investigated, distinguishing between inputs and outputs. A program will then be created for the Siemens S7-1214 PLC and an HMI for operator control and engineer control. Manual operation of the robot will be introduced, including safety interlocks. A program will then be written for an automatic sequence. Simulation testing will also be done to acquire test results for each mode. Finally, a practical test will be conducted on the robot itself in a group after discussing the optimum program to use.

already exists

This is a good plan

What has been learned?*Nice idea, but not required here*

Overall, a lot was learned on how user interfaces and ease of program navigation makes it easier for an engineer to do maintenance on the robot and fault find if need be. It also gave an understanding of how testing is conducted to ensure the system meets the need of the customer whilst also being reliable. An understanding was also built on making the HMI for the operator simple to use and observe whilst not overcomplicating it. Research played an important part for obtaining knowledge on different PLCs and their specifications. In an engineering workplace, different PLCs would need to be compared to find which is the most cost effective and which is the most suitable for a certain task.

2 - Research

Allen Bradley Programmable Logic Controller, MicroLogix DIN Rail Mount, 120 to 240V AC Supply Voltage – 1766-L32BWA:

Allen Bradley states that " ……..

The PLC model stated above "*provides fast processing speeds and a large storage capacity for efficient coordination of your industrial machinery*". This would be ideal for robot control in the manufacturing industry, due to the demand of fast processing speeds and expansion modules can also be purchased to increase the number of inputs and output ports to 144 (RS Components Ltd., n.d.).

year?

why no date?

It also features 20 inputs, and 12 relay outputs. This is enough for the robot being used in this assignment and would be ideal in this case. The power supply is also embedded in the controller itself (Rockwell Automation, Inc., 2008, p. 2).

only include page numbers for direct quotes

The MicroLogix series are designed for smaller tasks (Automation IT, 2022).

Correctly cited

This PLC is also easy to mount, with the use of DIN rail. This controller uses the RS Logix 500 software which doesn't support SFC programming (Rockwell Automation, Inc, 2008, p. 1).

only include page numbers for direct quotes



Figure 1 Allen Bradley Programmable Logic Controller MicroLogix, DIN Rail Mount, 120 to 240V AC Supply Voltage - 1766-L32BWA (RS Components Ltd., n.d.)

Allen Bradley - CompactLogix 5380 Controller, 600KB, 8 I/Os, 16 nodes, Standard - 5069-L306ER:

quotes should start sooner

Rockwell Automation. Inc state the CompactLogix 5380 controllers offer "*higher performance and increased capacity*". They also say they are ideal for small and large applications where low axis motion and I/O point counts are required (2022). *page number should be given with direct quotes*

The CompactLogix processors are used where cost is a big consideration and are used in smaller processes (Automation IT, 2022). Correctly cited

High speed I/O can also be enabled. They also have enhanced diagnostics and troubleshooting with fault status and controller display. This is ideal where a robot goes into fault conditions, making it easier to fault find. The power supply is also integrated. Unlike the MicroLogix series controllers, the CompactLogix series do not have I/O embedded into the controller, so expansion modules must be purchased along with it. This makes it less cost effective. The 5069-L306ER model has a maximum expansion module capability of 8 modules. The controller itself along with the expansion modules are mounted with DIN rail (Rockwell Automation Ltd., 2022).

The Compact Logix series controllers also offers Studio Logix 5000 software, according to (Rockwell Automation, Inc., 2019, p. 3). This software supports Sequential Function Chart (SFC) programming, which is ideal for the robot, due to containing an automatic sequence (Rockwell Automation Publication 1756-PM018I-EN-P, 2022, p. 12). *page numbers are for direct quotes*



Figure 2 CompactLogix 5380 Controller, 600KB, 8 I/Os, 16 nodes, Standard (Rockwell Automation Ltd., 2022)

Allen Bradley - CompactLogix™ 5370 L2 Programmable Automation Controllers - 1769-L24ER-QB1B:

The 1769-L24ER-QB1B model of the CompactLogix 5370 series controllers are said to be perfect for small and mid-size applications. The PLC also has 16 digital inputs and 16 digital outputs embedded, with the availability of adding 4 extra modules (Routeco Limited, 2022).

This particular model is ideal for small, stand-alone applications and higher performance applications. Rockwell Automation, Inc states it includes "*assembly machines, hoisting systems, process skids, indexing tables, and packaging*". They also say that is has a small form factor, which would be ideal as an alternative to the current PLC being used for the robot (2019, p.17).

This model has the power supply built in. This makes it more cost effective (Rockwell Automation, Inc., 2019, p. 18).

The controller is also simply mounted with DIN rail (Routeco Limited, 2022).

This PLC also supports SFC programming due to containing Studio Logix 5000 software (Rockwell Automation Publication 1756-PM018I-EN-P, 2022, p. 12).



Figure 3 Allen Bradley - CompactLogix™ 5370 L2 Programmable Automation Controllers - 1769-L24ER-QB1B (Routeco Limited, 2022)

Comparison table:

Table 1 PLC Comparison

part numbers are confusing, stick with the model number

PLC	MicroLogix 1766 – L32BWA	CompactLogix – 5069-L306ER	CompactLogix - 1769-L24ER-QB1B
Digital I/O	20 inputs and 12 relay outputs.	None *you should consider explansion IO instead*	16 digital inputs and 16 digital outputs
Analogue I/O	None	None	None
Expandability	Up to 7 I/O modules	Up to 8 I/O modules	Up to 4 modules
Memory	10KB	600KB	750KB
Mounting	DIN Rail	DIN rail	Din Rail
Cost	£952.82	£1333.19	£1681.80
SFC Language	No	Yes	Yes
Power Supply	Built in 120V/240V AC	Built in 24V DC	Built in 24V DC
Advantages	Comes with embedded I/O with a built-in power supply and has a good amount of expandability. It's also the most cost effective out of the three.	Power supply is built in and contains an adequate amount of user memory. This PLC has the most expandability out of the three, due to being expandable up to 8 modules. It also supports SFC programming.	Includes embedded I/O and the power supply is built in. This PLC also contains the most amount of user memory out of the three, making more room for the program and memory data. It also supports SFC programming.
Disadvantages	Contains the least amount of user memory out of the three which means there is less space for the program and memory data. Analogue I/O expansion modules will need to be purchased separately. It also does not support SFC language.	Does not include embedded I/O, making it less cost effective. Expansion modules will need purchasing with the controller itself. Analogue I/O expansion modules will also need to be purchased separately.	This PLC is the least cost effective out of the three and has the least amount of expandability, only being able to expand up to 4 modules. There are also no embedded analogue modules.
	(Rockwell Automation, Inc., 2008, p. 2) (Rockwell Automation, Inc., 2008, p. 1) (RS Components Ltd., n.d.) (Rockwell Automation, Inc., 2009)	(Rockwell Automation Ltd., 2022) (Routeco Limited, 2022) (Rockwell Automation, Inc., 2022)	(Rockwell Automation, Inc., 2019) (Routeco Limited, 2022)

Best Allen Bradley PLC alternative for the robot:

From the comparison table, the best Allen Bradley alternative would be the MicroLogix – L32BWA. This is because it includes an adequate amount of I/O, contains the most cost effective out of the three and has enough user memory to execute the program due to it being a small program. There is also a respectable amount of expandability, if need be, with up to 7 modules.

The CompactLogix - 1769-L24ER-QB1B is also a good option if cost isn't an issue, as it contains more user memory, some expandability and the ability to program in SFC language.

Goodwin states that "*An oversized controller allows for more flexibility within the cell long-term, thus accommodating any changes",* meaning it is still okay to have an oversized module for a small system (2021). *page number?*

The CompactLogix – 5069-L306ER may be the least cost effective simply due to not having any embedded I/O, however it does offer the best amount of expandability for long term changes to the system and offers SFC programming language.

3 - Hardware Investigation and Input-Output List

Various pieces of hardware were used throughout this task. The PLC itself is a Siemens S7-1214 DC/DC/DC. There is also an HMI on the robot door which was used to control the robot movement manually and automatically. The robot axis is powered with pneumatic pistons that are controlled with solenoid valves. These also control the robot grippers. The robot also features sensors that dictate the position of the robot. There is also a proximity switch that indicates where an object is from the entry chute. This is used to trigger a sequence. There are also 2 rocker switches and 3 pushbuttons that are connected to the PLC, shown in figure 5.



Figure 4 PLC Controlled Robot Hardware



Figure 5 Pushbuttons and Rocker Switches

Inputs and outputs from the robot are integrated in the software with tag tables in order for the robot to be PLC controlled. The inputs and outputs with their addresses and descriptions are as follows:

Robot Input Addresses	Description
%I0.0	UP Sensor
%I0.1	DOWN Sensor
%I0.2	Gripper Open Sensor
%I0.3	Gripper Closed Sensor
%I0.4	IN Sensor
%I0.5	OUT Sensor
%I0.6	RIGHT Sensor
%I1.7	LEFT Sensor
%I1.0	Pushbutton X10
%I1.1	Pushbutton X11
%I1.2	Pushbutton X12
%I1.3	Pushbutton X13
%I1.4	Pushbutton X14
%I1.5	Object Sensor

Robot Output Addresses	Description
%Q0.0	Move DOWN Solenoid
%Q0.1	Move UP Solenoid
%Q0.2	Close Gripper Solenoid
%Q0.3	Open Gripper Solenoid
%Q0.4	Move OUT Solenoid
%Q0.5	Move IN Solenoid
%Q0.6	Move LEFT Solenoid
%Q0.7	Move Right Solenoid

excellent information

4 - Program Design and Explanation

Main Program Block:

The program created in TIA portal contains a main program block and an auto sequence program block. The main program block contains addresses programmed in ladder logic and features addresses for manual control and automatic control, such as memory, input and output addresses. These addresses are written in 'PLC Tags'. The main program contains a set of rungs and a bit, which is used to change between simulation and actual inputs. Most rungs feature a control for a specific movement. It also contains automatic control and manual control with various interlocks in place to detect the position of the robot. This is true from network 2 to 9 which are all the position controls. Each of these (except network 8&9) contain an up-sensor interlock which doesn't allow the robot to move in certain directions unless the robot is in the UP position. Network 3, down control contains a combination of safety interlocks only allowing the robot to move down unless located over a station. Network 10 contains the automatic sequence written in SCL code in the main block in order for the sequence to function correctly.

Networks 11 and 12 contain a delay sequence where the sequence will start after a 2 second delay from the object sensor detecting an object.

Network 13 contains an up and down counter and simply counts the number of objects in the machine.

Good explanation given

(See Appendix C, page 33 for network comments)

Automatic Sequence Block:

The 'auto sequence' program block contains a sequence the robot will operate for picking and placing the objects. It is written in SCL code in which a lot of robots use in industry today. It is placed in the main program block. When 'auto mode' is selected, it triggers a step sequence which sends the robot to the home position. Once it is in the home position, the object sensor will start the sequence or the 'start sequence' button is pressed on the HMI by the engineer. The automatic pick and place sequence the robot follows is shown in the Sequential Function Chart (see Appendix B, page 28).

Good explanation given

Operator's Screen:

The operator's screen was made to be user friendly and simple to use. 'Manual' and 'Auto' buttons were added to allow the operator switch between manual and automatic control. Manual control was also added via various buttons. The operator can also see the number of objects in the machine from the counter display and know the robot position from the sensor indicators. They can also see the step number from the automatic sequence (see Appendix A, page 26).

Good explanation given

Engineer's Screen:

The engineer's screen contains the same elements as the operator's screen, however there are a few additional elements, such as the 'simulate inputs' button which allows the engineer to simulate the robot in operation. The screen also contains solenoid indication which indicates when the solenoid valves are in operation. They can also simulate the object sensor sequence delay via the use of the 'start sequence' and observing the timer display. They can also pause and continue the sequence via the 'pause' and 'continue' buttons. This would be used especially for fault rectification to see which inputs and outputs are active and inactive (see Appendix A, page 27).

Good explanation given

5 - Simulation Testing

From the results seen in the appendix, all tests conducted in simulation mode have passed. The HMI was observed to determine the test results, along with observing the program in monitor mode which showed which contacts were closed, which coils were energised, the state of the counter etc. The actual results were compared with the expected results, then each test was also determined if they passed or failed (see Appendix D, page 40).

6 - Practical testing

Once the simulation testing was complete, a discussion was made on which program to practically test. There were two programs to choose from. Program structure and ease of navigation were assessed and the program in this assignment was chosen. The same tests as the simulation were conducted and all passed except test 8. This was due to a last-minute change on the robot being tested on. Initially, the robot ending with IP address '125' was going to be used, however the robot being tested on ended up ending with the IP address '126'. These robots were slightly different due to their object sensor locations. The first robot would have the object stop on the sensor which would keep the normally open contact in the software energised, hence staring the sequence. The robot being tested on had the object sensor in a location where the object would simply roll past it, not giving the delay timer enough time to initiate the automatic sequence. Therefore, it couldn't be tested. The software would have required alteration in order for the object sensor to work correctly (see Appendix E, page 56).

The brief asked for aa 2 second delay and you should have design the program to start with either robot

You should have used a memorise bit as the object rolls past the sensor

7 - Discussion, Conclusions and Recommendations

Discussion:

Maintenance and reliability of the system was considered by first adding network titles and comments within the main program block. This made it easier for an engineer to understand how the networks operate and how different input, outputs and memory bits interact with each other. It also gave an understanding on how the various interlocks work. An HMI screen was also created for the engineer to assist in fault finding. Being able to pause the sequence to see what inputs and outputs are active at a certain step makes it easier for the engineer to fault find in this case. Having manual control also gave the engineer the ability to test the positional controls after doing maintenance on the robot. The HMI also satisfies the user's needs by displaying how many objects are in the machine and what the step the robot is on for the automatic sequence. The system was also reliable for the operator with the HMI screen. From all the information added to this, the operator should be able to communicate with the engineer stating what the faut is. E.g., the robot does not go into the home position which could be due to a faulty position sensor. The system could've been more secure by adding authentication to enter the program and make any changes, as anyone could access it.

Good explanation given

Conclusion:

Overall, the creation and testing of the program went well, with only one failed test. The program was able to execute all tasks effectively within the simulation along with the robot itself. There were no major problems ran into. The position interlocks worked as they should, closing the solenoid until the robot couldn't reach any further. The operator screen was also made simplistic and easy to use. The engineer screen was made to be easy to fault find on. The program itself was also easy to navigate and the networks were easy to interpret, with the comments making it easier to understand how the inputs and outputs were interacting with each other. The only problem with testing was being unable to test the object sensor timer delay on the robot itself due to the two robots being slightly different. Adequate research was also completed to understand the technical implications of the PLC along with cost.

Good conclusions

Recommendations:

One recommendation that can be made for the reliability of the system is to add a 'maintenance mode'. This would be to add a counter and after a certain number of automatic sequence cycles, the counter will open normally closed memory bit from a comparison block which stops the sequence from starting and can be indicated on the HMI stating maintenance is required. This will make the robot more reliable in that preventative planned maintenance can be completed, such as safety checks, compressor maintenance, solenoid valve maintenance etc. The engineer will then need authorisation to reset the maintenance mode switch once it is complete and the robot can be returned to operation. Another recommendation that can be made is to use a PLC that supports SFC programming, such as the Compact Logix series controllers discussed in the research part of the assignment. This would have been ideal for the automatic sequence.

excellent ideas presented

The Siemens S7-1511 would also allow use of SFC

8 - Team Working

Working in a team allowed different methods of robot control to be discussed. There were many ways to approach writing the program and being able to see how different people completed this give ideas on how to improve, as they may have used a simpler solution to control the robot. This had the possibility of making the program easier to understand, which is required in engineering. Not only with programs, but with engineering as a whole, it is better to have different approaches. Discussing which program to use enabled the ability to find advantages and disadvantages in the software, regarding navigation, usability and ease of control. Working as a team also gave everyone a chance to have an impact on the task, which is good practice for when this needs to be applied in the engineering industry and ensures everyone obtains an equal opportunity.

9 - Citation List

Correct method of listing.

Good range of sources

Automation IT, 2022. *Choosing a PLC: Things to consider.* [Online]
Available at: https://www.automationit.com/blog/80-choosing-a-plc-things-to-consider
[Accessed 8th December 2022].

Goodwin, D., 2021. *Programmable Logic Controller (PLC) Considerations for Robot Cells.* [Online]
Available at: https://control.com/technical-articles/controller-considerations-for-small-assembly-robots/
[Accessed 13th December 2022].

Rockwell Automation Ltd., 2022. *CompactLogix 5380 Controller, 600KB, 8 I/Os, 16 nodes, Standard.* [Online]
Available at: https://www.rockwellautomation.com/en-gb/products/details.5069-L306ER.html
[Accessed 6th December 2022].

Rockwell Automation Publication 1756-PM018I-EN-P, 2022. *Logix 5000 Controllers IEC.* [Online]
Available at: https://literature.rockwellautomation.com/idc/groups/literature/documents/pm/1756-pm018_-en-p.pdf
[Accessed 24th January 2023].

Rockwell Automation, Inc., 2008. *MicroLogix 1400 Product Profile.* [Online]
Available at: https://docs.rs-online.com/675a/0900766b80f10301.pdf
[Accessed 6th December 2022].

Rockwell Automation, Inc., 2009. *MicroLogix Programmable Controllers Selection Guide.* [Online]
Available at: https://docs.rs-online.com/8616/0900766b80dfc363.pdf
[Accessed 8th December 2022].

Rockwell Automation, Inc., 2019. *CompactLogix System Selection Guide.* [Online]
Available at: https://www.routeco.com/en-gb/resourceviewer?resource=RO/1769-sg001_-en-p.pdf?937FB0959B7A8489A28AA7A1A4868C65877B8D72C59E606481A481AAC064A062C0A9BA?desc=Selection%20Guide_1769-L24ER-QB1B
[Accessed 7th December 2022].

Rockwell Automation, Inc., 2022. *CompactLogix 5380 Controllers.* [Online]
Available at: https://www.rockwellautomation.com/en-us/products/hardware/allen-bradley/programmable-controllers/small-controllers/compactlogix-family/compactlogix-5380-controllers.html
[Accessed 6th December 2022].

Rockwell Automation, Inc, 2008. *PRODUCT PROFILE MicroLogix™ 1400 / 1766 Small Programmable Logic Controller.* [Online]
Available at: https://docs.rs-online.com/675a/0900766b80f10301.pdf
[Accessed 24th January 2023].

Routeco Limited, 2022. *CompactLogix - Medium Syetms.* [Online]
Available at: https://www.routeco.com/en-gb/shop/programmable-controllers/compactlogix/5069-l306erm
[Accessed 8th December 2022].

Routeco Limited, 2022. *CompactLogix - Medium Systems.* [Online]
Available at: https://www.routeco.com/en-gb/shop/programmable-controllers/compactlogix/1769-l24er-qb1b
[Accessed 6th December 2022].

RS Components Ltd., n.d. *PLCs - Programmable Logic Controllers.* [Online]
Available at: https://uk.rs-online.com/web/p/plcs-programmable-logic-controllers/7140095
[Accessed 6th December 2022].

10 - Appendices

Appendix A: - Program Design

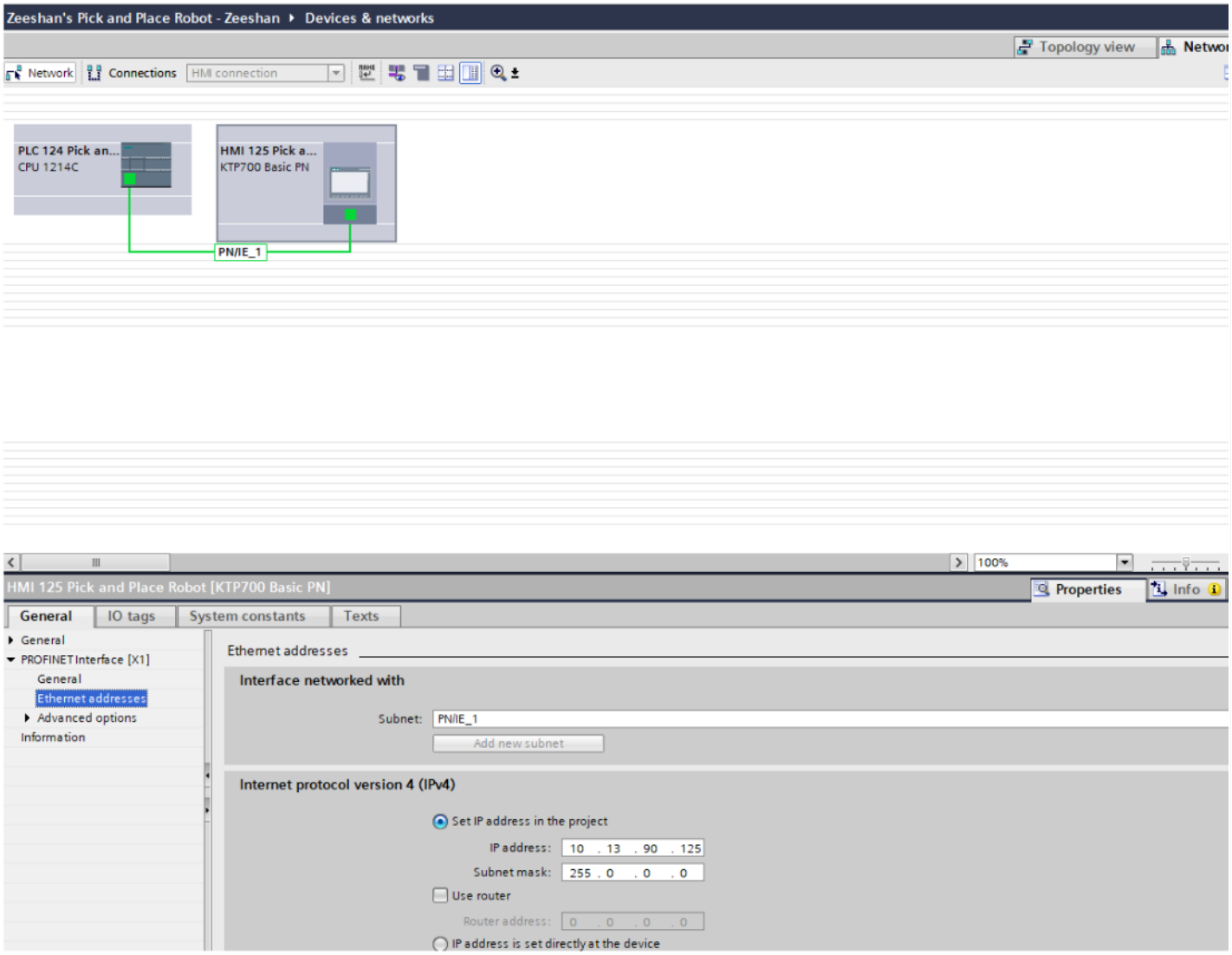


Figure 6 Ethernet Address Configuration

PLC tags

	Name	Tag table	Data type	Address	Retain	Acces...	Writa...	Visibl...
1	UP sensor	IO tags	Bool	%I0.0	☐	☑	☑	☑
2	DOWN sensor	IO tags	Bool	%I0.1	☐	☑	☑	☑
3	OPEN sensor	IO tags	Bool	%I0.2	☐	☑	☑	☑
4	CLOSED sensor	IO tags	Bool	%I0.3	☐	☑	☑	☑
5	IN sensor	IO tags	Bool	%I0.4	☐	☑	☑	☑
6	OUT sensor	IO tags	Bool	%I0.5	☐	☑	☑	☑
7	RIGHT sensor	IO tags	Bool	%I0.6	☐	☑	☑	☑
8	LEFT sensor	IO tags	Bool	%I0.7	☐	☑	☑	☑
9	X10 Pushbutton	IO tags	Bool	%I1.0	☐	☑	☑	☑
10	X11 Pushbutton	IO tags	Bool	%I1.1	☐	☑	☑	☑
11	X12 Pushbutton	IO tags	Bool	%I1.2	☐	☑	☑	☑
12	X13 toggle switch	IO tags	Bool	%I1.3	☐	☑	☑	☑
13	X14 toggle switch	IO tags	Bool	%I1.4	☐	☑	☑	☑
14	OBJECT sensor	IO tags	Bool	%I1.5	☐	☑	☑	☑
15	DOWN solenoid	IO tags	Bool	%Q0.0	☐	☑	☑	☑
16	UP solenoid	IO tags	Bool	%Q0.1	☐	☑	☑	☑
17	CLOSE solenoid	IO tags	Bool	%Q0.2	☐	☑	☑	☑
18	OPEN solenoid	IO tags	Bool	%Q0.3	☐	☑	☑	☑
19	OUT solenoid	IO tags	Bool	%Q0.4	☐	☑	☑	☑
20	IN solenoid	IO tags	Bool	%Q0.5	☐	☑	☑	☑
21	LEFT solenoid	IO tags	Bool	%Q0.6	☐	☑	☑	☑
22	RIGHT solenoid	IO tags	Bool	%Q0.7	☐	☑	☑	☑
23	System_Byte	System Tags	Byte	%MB0	☐	☑	☑	☑
24	FirstScan	System Tags	Bool	%M0.0	☐	☑	☑	☑
25	DiagStatusUpdate	System Tags	Bool	%M0.1	☐	☑	☑	☑
26	AlwaysTRUE	System Tags	Bool	%M0.2	☐	☑	☑	☑
27	AlwaysFALSE	System Tags	Bool	%M0.3	☐	☑	☑	☑
28	Clock_Byte	System Tags	Byte	%MB1	☐	☑	☑	☑
29	Clock_10Hz	System Tags	Bool	%M1.0	☐	☑	☑	☑
30	Clock_5Hz	System Tags	Bool	%M1.1	☐	☑	☑	☑
31	Clock_2.5Hz	System Tags	Bool	%M1.2	☐	☑	☑	☑
32	Clock_2Hz	System Tags	Bool	%M1.3	☐	☑	☑	☑
33	Clock_1.25Hz	System Tags	Bool	%M1.4	☐	☑	☑	☑
34	Clock_1Hz	System Tags	Bool	%M1.5	☐	☑	☑	☑
35	Clock_0.625Hz	System Tags	Bool	%M1.6	☐	☑	☑	☑
36	Clock_0.5Hz	System Tags	Bool	%M1.7	☐	☑	☑	☑
37	UP HMI Request	My Tags	Bool	%M100.0	☐	☑	☑	☑
38	DOWN HMI Request	My Tags	Bool	%M100.1	☐	☑	☑	☑
39	Simulate	My Tags	Bool	%M100.2	☐	☑	☑	☑
40	IN HMI Request	My Tags	Bool	%M100.3	☐	☑	☑	☑

Figure 7 PLC Tag List 1 - 40

41		OUT HMI Request	My Tags	Bool	%M100.4	☐	☑	☑	☑
42		LEFT HMI Request	My Tags	Bool	%M100.5	☐	☑	☑	☑
43		RIGHT HMI Request	My Tags	Bool	%M100.6	☐	☑	☑	☑
44		OPEN HMI Request	My Tags	Bool	%M100.7	☐	☑	☑	☑
45		CLOSE HMI Request	My Tags	Bool	%M101.0	☐	☑	☑	☑
46		Auto Mode	My Tags	Bool	%M110.0	☐	☑	☑	☑
47		UP Auto Request	My Tags	Bool	%M110.1	☐	☑	☑	☑
48		DOWN Auto Request	My Tags	Bool	%M110.2	☐	☑	☑	☑
49		IN Auto Request	My Tags	Bool	%M110.3	☐	☑	☑	☑
50		OUT Auto Request	My Tags	Bool	%M110.4	☐	☑	☑	☑
51		LEFT Auto Request	My Tags	Bool	%M110.5	☐	☑	☑	☑
52		RIGHT Auto Request	My Tags	Bool	%M110.6	☐	☑	☑	☑
53		OPEN Auto Request	My Tags	Bool	%M110.7	☐	☑	☑	☑
54		CLOSE Auto Request	My Tags	Bool	%M111.0	☐	☑	☑	☑
55		Start Sequence	My Tags	Bool	%M150.0	☐	☑	☑	☑
56		Step	My Tags	UInt	%MW200	☐	☑	☑	☑
57		Next Step	My Tags	UInt	%MW250	☐	☑	☑	☑
58		Start Memory	My Tags	Bool	%M200.0	☐	☑	☑	☑
59		Pause	My Tags	Bool	%M250.0	☐	☑	☑	☑

Figure 8 PLC Tag List 41 - 59

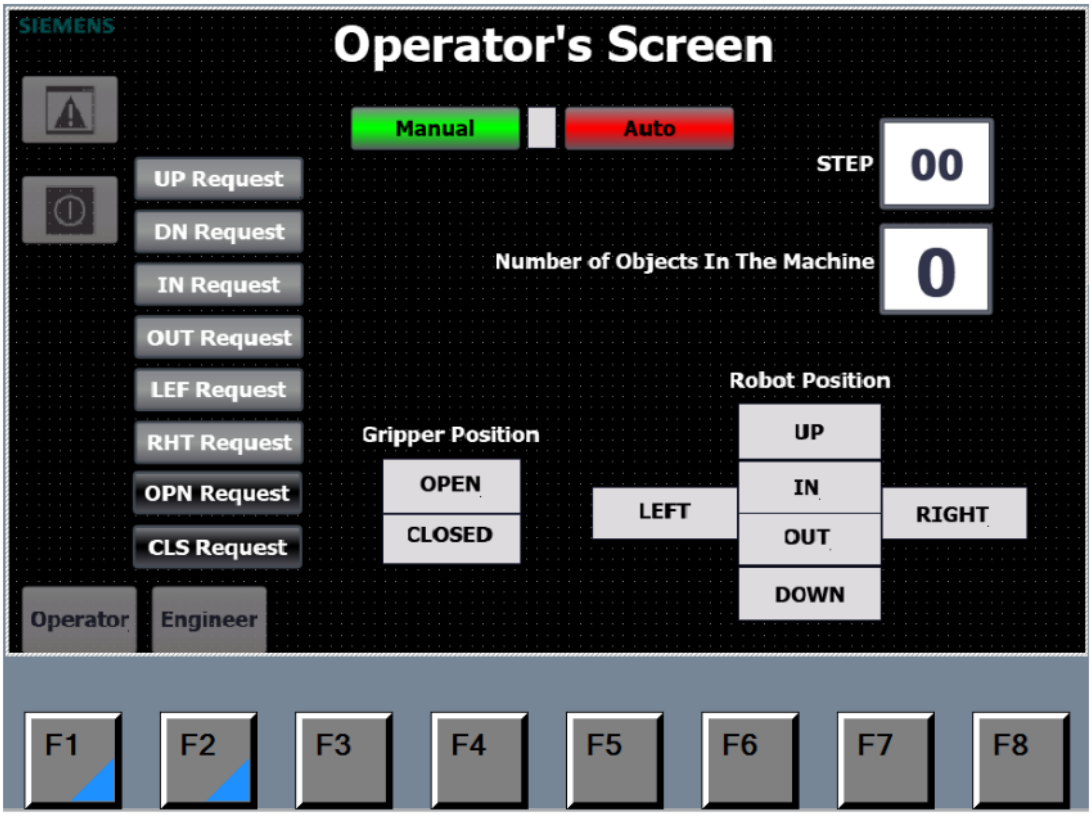


Figure 9 Operator's Screen

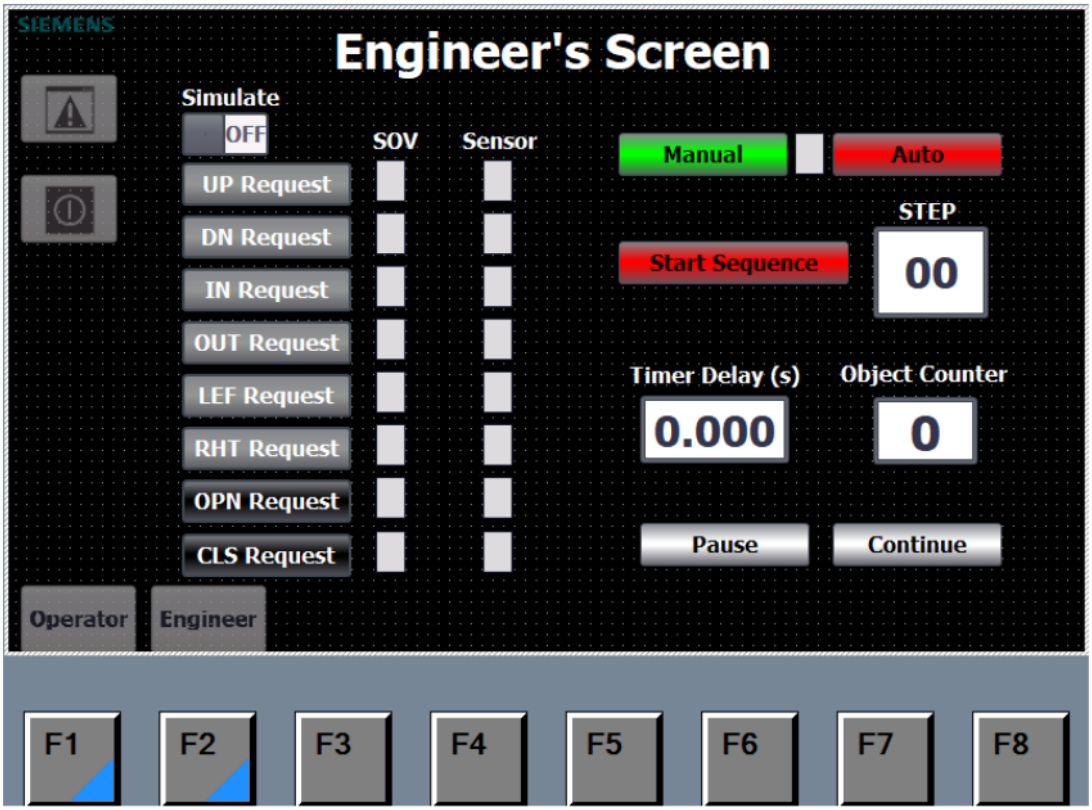


Figure 10 Engineer's Screen

Appendix B: - Program Design, Automatic Control

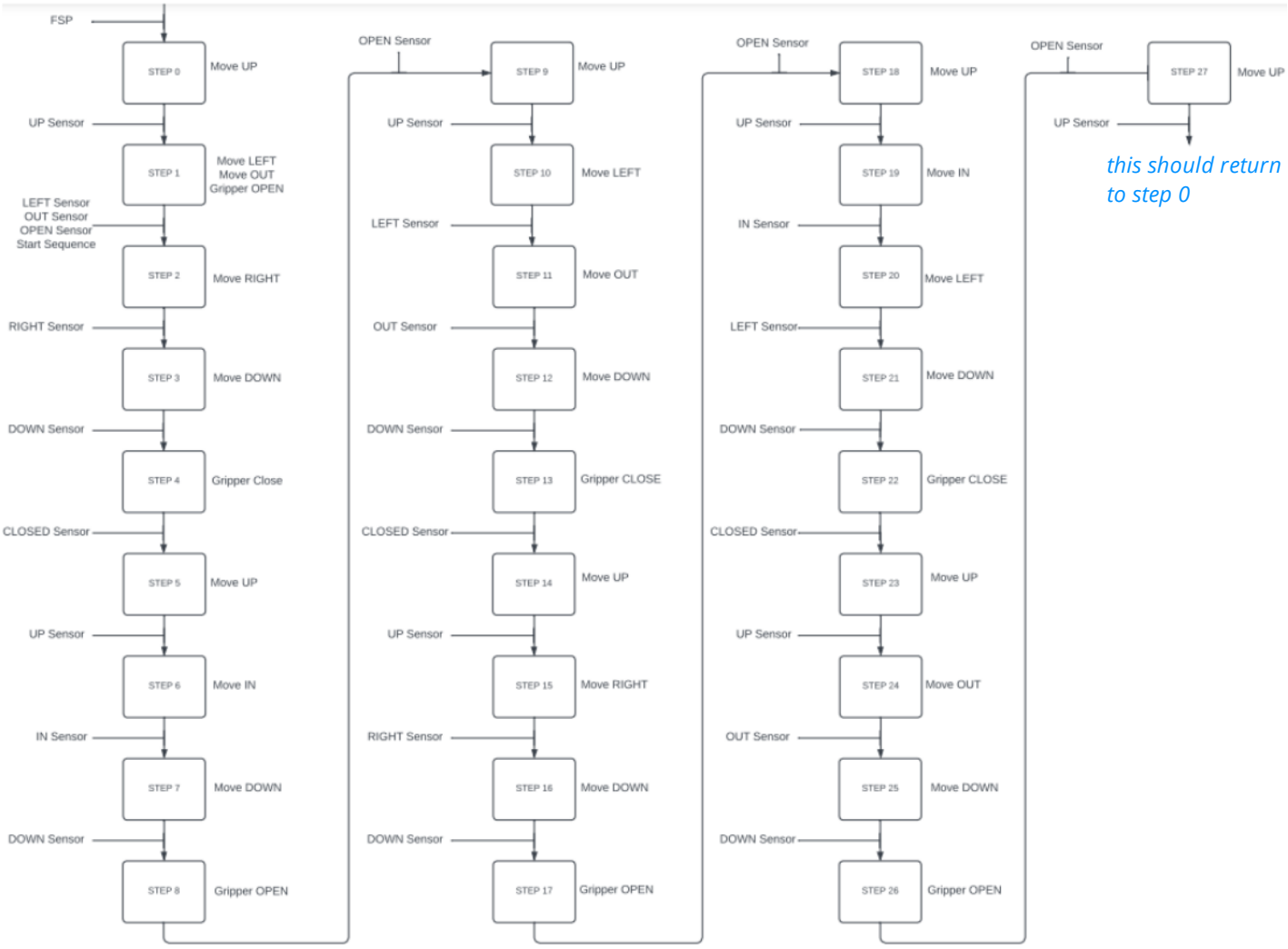


Figure 11 Automatic Sequential Function Chart

```
IF ("Auto Mode" = 0) OR ("FirstScan" = 1) THEN // Auto Mode or FirstScan engages the automatic sequence.
    "Step" := 0; // Sets step to '0'.
    "Next Step" := 0; // Enables next step.
END_IF;

CASE "Step" OF
    0:
        "UP Auto Request" := 1; // Requests the robot to move up, before allowing it to navigate to the home position.
        IF "UP sensor" = 1 THEN // Up sensor engages.
            "Next Step" := 1; // Step 1 begins.
            "UP Auto Request" := 0; // Resets the automatic UP request.
        END_IF;

    1:
        "LEFT Auto Request" := 1; // Requests the robot to move left.
        "OUT Auto Request" := 1; // Requests the robot to move out.
        "OPEN Auto Request" := 1; // Requests robot grippers to open. Once this is complete, the robot is succesfully in the home position.
        IF ("LEFT sensor" = 1) AND ("OUT sensor" = 1) AND ("OPEN sensor" = 1) AND ("Start Sequence" = 1) AND "IEC_Counter_0_DB".CV = 2 AND "Delay Timer".Q = 1 THEN
            "Next Step" := 2; // Step 2 begins if the out sensor, in sensor, open sensor, start sequence and delay timer output are '1' and the counter reads '2', as seen above.
            "LEFT Auto Request" := 0; // Resets the automatic LEFT request.
            "OUT Auto Request" := 0; // Resets the automatic OUT request.
            "OPEN Auto Request" := 0; // Resets the automatic OPEN request.
        END_IF;
        IF ("LEFT sensor" = 1) AND ("OUT sensor" = 1) AND ("OPEN sensor" = 1) AND ("Start Sequence" = 1) AND "IEC_Counter_0_DB".CV = 1  AND "Delay Timer" .Q = 1 THEN
                "Next Step" := 12; // Step 12 begins if the out sensor, in sensor, open sensor, start sequence and delay timer output are '1' and the counter reads '1', as seen above.
                "LEFT Auto Request" := 0; // Resets the automatic LEFT request.
                "OUT Auto Request" := 0; // Resets the automatic OUT request.
                "OPEN Auto Request" := 0; // Resets the automatic OPEN request.
            END_IF;
            IF ("LEFT sensor" = 1) AND ("OUT sensor" = 1) AND ("OPEN sensor" = 1) AND ("Start Sequence" = 1) AND "IEC_Counter_0_DB".CV = 0  AND "Delay Timer" .Q = 1 THEN
                "Next Step" := 19; // Step 19 begins if the out sensor, in sensor, open sensor, start sequence and delay timer output are '1' and the counter reads '0', as seen above.
                "LEFT Auto Request" := 0; // Resets the automatic LEFT request.
                "OUT Auto Request" := 0; // Resets the automatic OUT request.
                "OPEN Auto Request" := 0; // Resets the automatic OPEN request.
            END_IF;

    2:
        "RIGHT Auto Request" := 1; // Sequence begins, starting with the RIGHT automatic requuest.
        IF "RIGHT sensor" = 1 THEN // Once the RIGHT sensor engages, it will move onto the next step.
            "Next Step" := 3; // Requests next step.
            "RIGHT Auto Request" := 0; // Resets RIGHT automatic request. Case number.
        END_IF;

    3:
        "DOWN Auto Request" := 1; // Requests the robot to move down.
        IF "DOWN sensor" = 1 THEN
            "Next Step" := 4; // Next step initiates once the DOWN sensor is engaged.
            "DOWN Auto Request" := 0; // Resets the DOWN request.
        END_IF;

    4:
        "CLOSE Auto Request" := 1; // Requests robot to close its grippers.
        IF "CLOSED sensor" = 1 THEN
            "Next Step" := 5; // Next step initiates once the CLOSED sensor is engaged.
            "CLOSE Auto Request" := 0; // Resets the CLOSE request.
        END_IF;

    5:
        "UP Auto Request" := 1; // Requests robot to move up.
        IF "UP sensor" = 1 THEN
```

Figure 12 Automatic Sequence SCL Code

```
            "Next Step" := 6; // Next step initiates once the UP sensor engages.
            "UP Auto Request" := 0; // Resets the UP request.
        END_IF;

    6:
        "IN Auto Request" := 1; // Requests robot to move in.
        IF "IN sensor" = 1 THEN
            "Next Step" := 7; // Next step initiates once the IN sensor is engaged.
            "IN Auto Request" := 0; // Resets the IN request.
        END_IF;

    7:
        "DOWN Auto Request" := 1; // Request the robot to move down.
        IF "DOWN sensor" = 1 THEN
            "Next Step" := 8; // Next step initiates once the DOWN sensor is engaged.
            "DOWN Auto Request" := 0; // Resets the DOWN request.
        END_IF;

    8:
        "OPEN Auto Request" := 1; // Requests the robot to open its grippers.
        IF "OPEN sensor" = 1 THEN
            "Next Step" := 9; // Next step initiates once the OPEN sensor is engaged.
            "OPEN Auto Request" := 0; // Resets the OPEN request
        END_IF;

    9:
        "UP Auto Request" := 1; // Requests robot to move up.
        IF "UP sensor" = 1 THEN
            "Next Step" := 10; // Next step initiates once the UP sensor engages.
            "UP Auto Request" := 0; // Resets the UP request.
        END_IF;

    10:
        "LEFT Auto Request" := 1; // Requests the robot to move left.
        IF "LEFT sensor" = 1 THEN
            "Next Step" := 11; // Next step initiates once the LEFT sensor is engaged.
            "LEFT Auto Request" := 0; // Resets the LEFT request.
        END_IF;

    11:
        "OUT Auto Request" := 1; // Requests the robot to move OUT.
        IF "OUT sensor" = 1 THEN
            "Next Step" := 12; // Next step initiates once the OUT sensor is engaged.
            "OUT Auto Request" := 0; // Resets the OUT request.
        END_IF;

    12:
        "DOWN Auto Request" := 1; // Request the robot to move down.
        IF "DOWN sensor" = 1 THEN
            "Next Step" := 13; // Next step initiates once the DOWN sensor is engaged.
            "DOWN Auto Request" := 0;  // Resets the DOWN request.
        END_IF;

    13:
        "CLOSE Auto Request" := 1; // Requests robot to close its grippers.
        IF "CLOSED sensor" = 1 THEN
            "Next Step" := 14; // Next step initiates once the CLOSED sensor is engaged.
            "CLOSE Auto Request" := 0; // Resets the CLOSE request.
        END_IF;
```

Figure 13 Continued SCL Code

```
    14:
        "UP Auto Request" := 1; // Requests robot to move up.
        IF "UP sensor" = 1 THEN
            "Next Step" := 15; // Next step initiates once the UP sensor engages.
            "UP Auto Request" := 0; // Resets the UP request.
        END_IF;

    15:
        "RIGHT Auto Request" := 1; // Requests the rboot to move right.
        IF "RIGHT sensor" = 1 THEN
            "Next Step" := 16; // Next step initiates once the RIGHT sensor is engaged.
            "RIGHT Auto Request" := 0; // Resets the RIGHT request.
        END_IF;

    16:
        "DOWN Auto Request" := 1; // Request the robot to move down.
        IF "DOWN sensor" = 1 THEN
            "Next Step" := 17; // Next step initiates once the DOWN sensor is engaged.
            "DOWN Auto Request" := 0; // Resets the DOWN request.
        END_IF;

    17:
        "OPEN Auto Request" := 1; // Requests the robot to open its grippers.
        IF "OPEN sensor" = 1 THEN
            "Next Step" := 18; // Next step initiates once the OPEN sensor is engaged.
            "OPEN Auto Request" := 0; // Resets the OPEN request.
        END_IF;

    18:
        "UP Auto Request" := 1; // Requests robot to move up.
        IF "UP sensor" = 1 THEN
            "Next Step" := 19; // Next step initiates once the UP sensor engages.
            "UP Auto Request" := 0; // Resets the UP request.
        END_IF;

    19:
        "IN Auto Request" := 1; // Requests robot to move in.
        IF "IN sensor" = 1 THEN
            "Next Step" := 20; // Next step initiates once the IN sensor is engaged.
            "IN Auto Request" := 0; // Resets the IN request.
        END_IF;

    20:
        "LEFT Auto Request" := 1; // Requests the robot to move left.
        IF "LEFT sensor" = 1 THEN
            "Next Step" := 21; // Next step initiates once the LEFT sensor is engaged.
            "LEFT Auto Request" := 0; // Resets the LEFT request.
        END_IF;

    21:
        "DOWN Auto Request" := 1; // Request the robot to move down.
        IF "DOWN sensor" = 1 THEN
            "Next Step" := 22; // Next step initiates once the DOWN sensor is engaged.
            "DOWN Auto Request" := 0; // Resets the DOWN request.
        END_IF;

    22:
        "CLOSE Auto Request" := 1; // Requests robot to close its grippers.
        IF "CLOSED sensor" = 1 THEN
            "Next Step" := 23; // Next step initiates once the CLOSED sensor is engaged.
            "CLOSE Auto Request" := 0; // Resets the CLOSE request.
        END_IF;

    23:
        "UP Auto Request" := 1; // Requests robot to move up.
        IF "UP sensor" = 1 THEN
            "Next Step" := 24; // Next step initiates once the UP sensor engages.
            "UP Auto Request" := 0; // Resets the UP request.
        END_IF;

    24:
        "OUT Auto Request" := 1; // Requests the robot to move OUT.
        IF "OUT sensor" = 1 THEN
            "Next Step" := 25; // Next step initiates once the OUT sensor is engaged.
            "OUT Auto Request" := 0; // Resets the OUT request.
        END_IF;

    25:
        "DOWN Auto Request" := 1; // Request the robot to move down.
        IF "DOWN sensor" = 1 THEN
            "Next Step" := 26;  // Next step initiates once the DOWN sensor is engaged.
            "DOWN Auto Request" := 0; // Resets the DOWN request.
        END_IF;

    26:
        "OPEN Auto Request" := 1; // Requests the robot to open its grippers.
        IF "OPEN sensor" = 1 THEN
            "Next Step" := 27; // Next step initiates once the OPEN sensor is engaged.
            "OPEN Auto Request" := 0;  // Resets the OPEN request.
```

```
        END_IF;

    27:
        "UP Auto Request" := 1; // Requests robot to move up.
        IF "UP sensor" = 1 THEN
            "Next Step" := 0; // Next step initiates once the UP sensor engages.
            "UP Auto Request" := 0; // Resets the UP request. The robot is now back in the home position.
        END_IF;
END_CASE;
"Step" := "Next Step"; // Allows the sequence to move onto next steps.
```

Figure 14 Completed SCL Code

Appendix C: - Program Design, Networks, Including Manual Control and Interlocks

[Main Program Block]

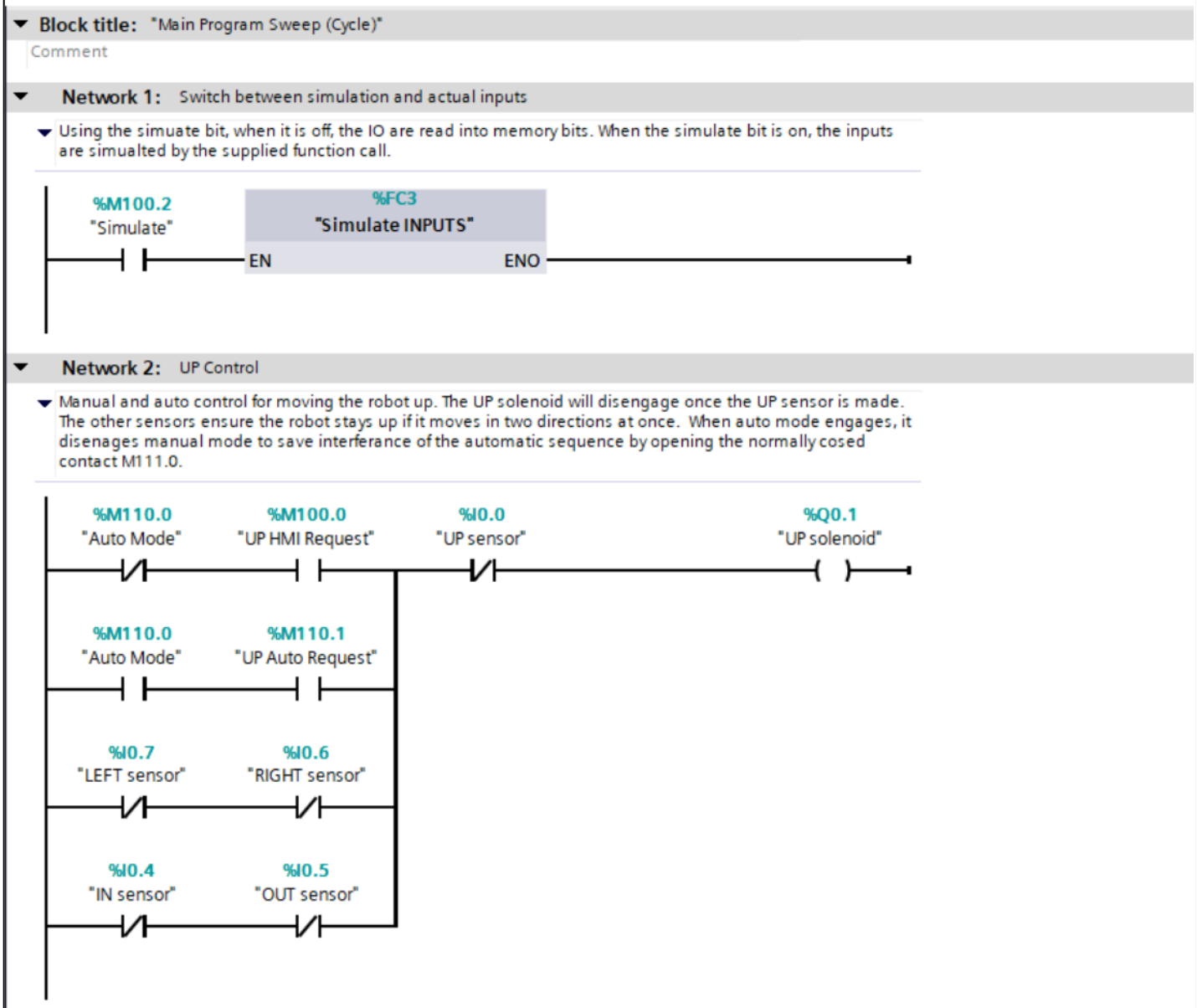
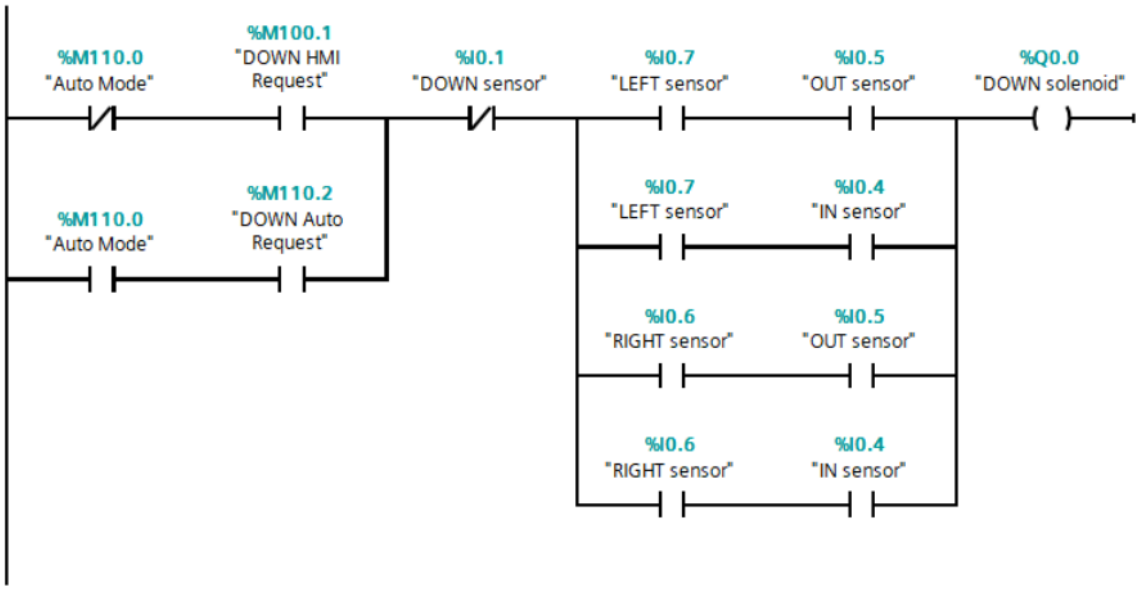


Figure 15 Networks 1 & 2

Network 3: DOWN Control

Manual and auto control for moving the robot down. When auto mode engages, it disenages manual mode to save interferance of the automatic sequence by opening the normally cosed contact M111.0. The DOWN solenoid will disengage once the DOWN sensor is made. Each of the other sensors in parallel indicate that the robot is at a station. Only when the robot is located over a station can it move down.



Network 4: IN Control

Manual and auto control for moving the robot IN. The IN solenoid will disengage once the IN sensor is made. The UP sensor works as an interlock as the robot can only move in when it is in the UP position. When auto mode engages, it disenages manual mode to save interferance of the automatic sequence by opening the normally cosed contact M111.0.

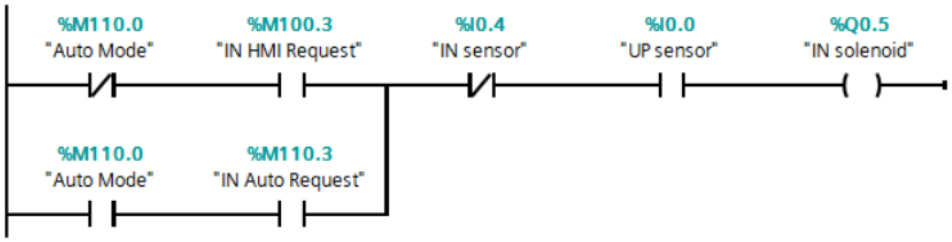


Figure 16 Networks 3 & 4

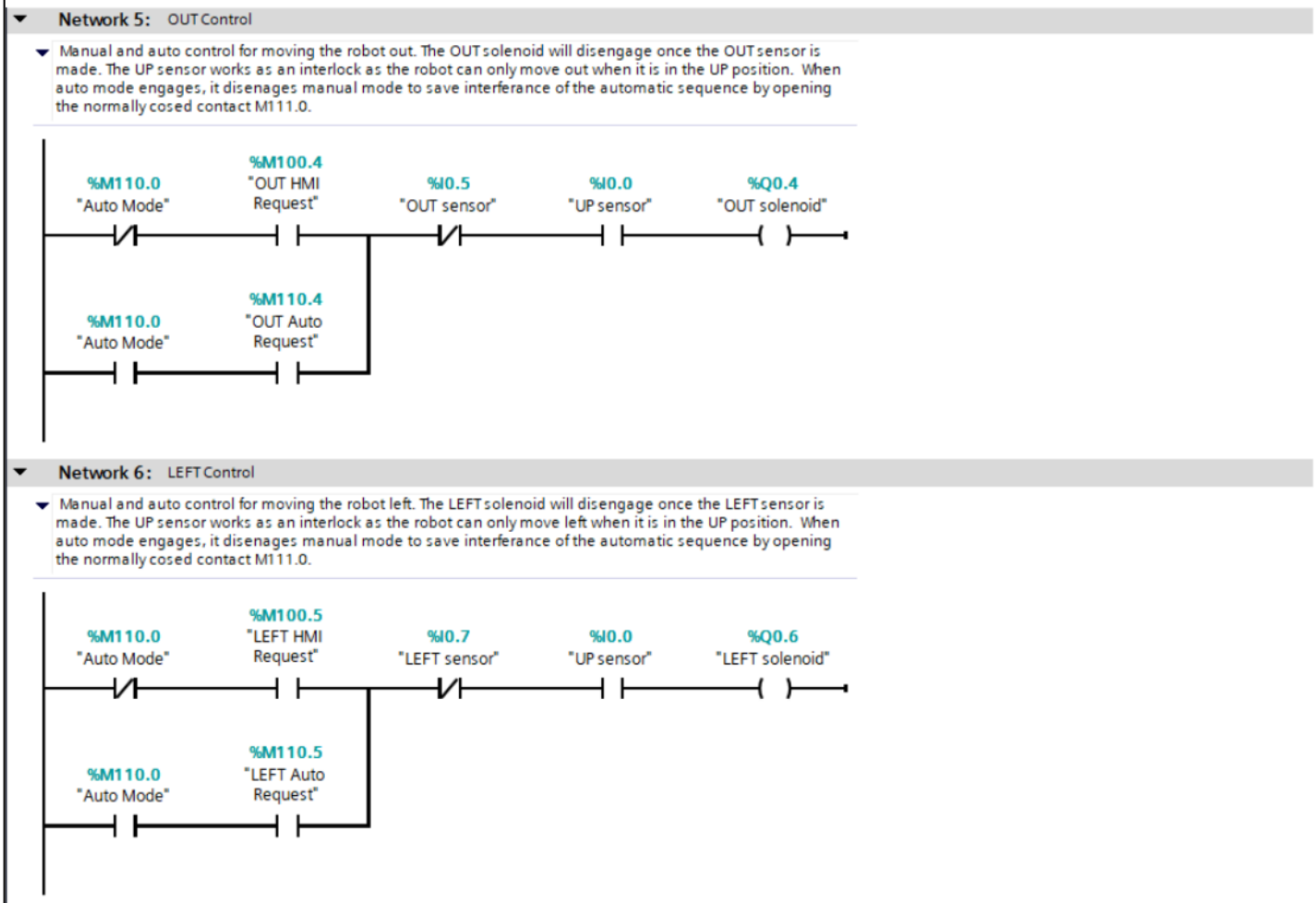
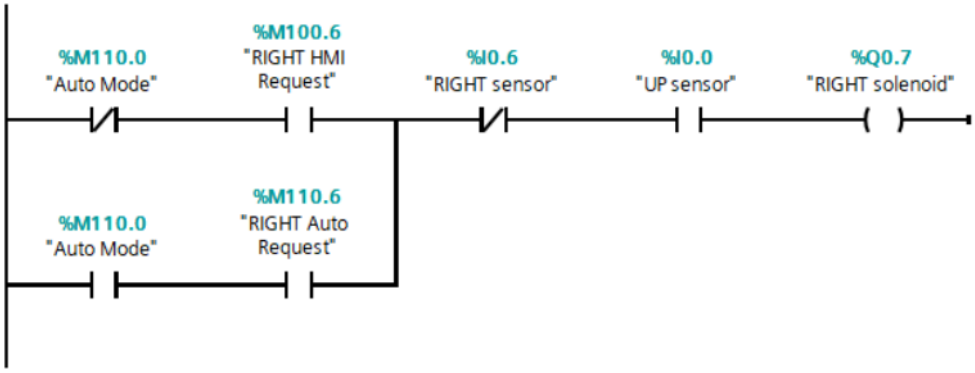


Figure 17 Networks 5 & 6

Network 7: RIGHT Control

Manual and auto control for moving the robot right. The RIGT solenoid will disengage once the RIGHT sensor is made. The UP sensor works as an interlock as the robot can only move right when it is in the UP position. When auto mode engages, it disenages manual mode to save interferance of the automatic sequence by opening the normally cosed contact M111.0.



Network 8: OPEN Jaws Control

Manual and auto control for opening the robot jaws. The OPEN solenoid will disengage once the OPEN sensor is made. When auto mode engages, it disenages manual mode to save interferance of the automatic sequence by opening the normally cosed contact M111.0.

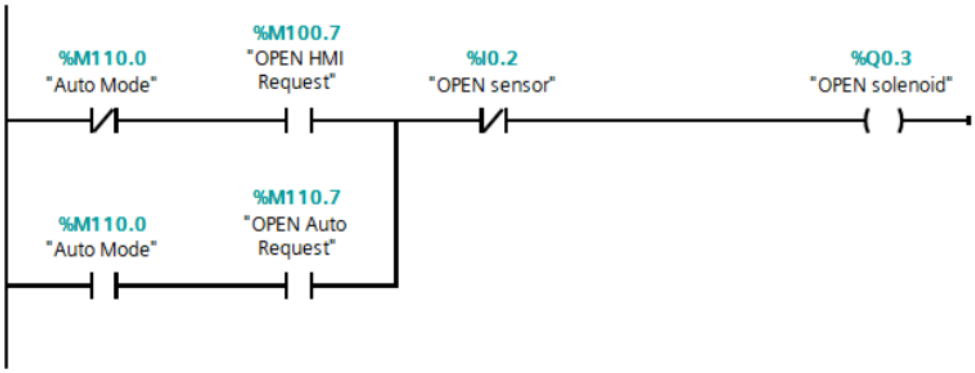


Figure 18 Networks 7 & 8

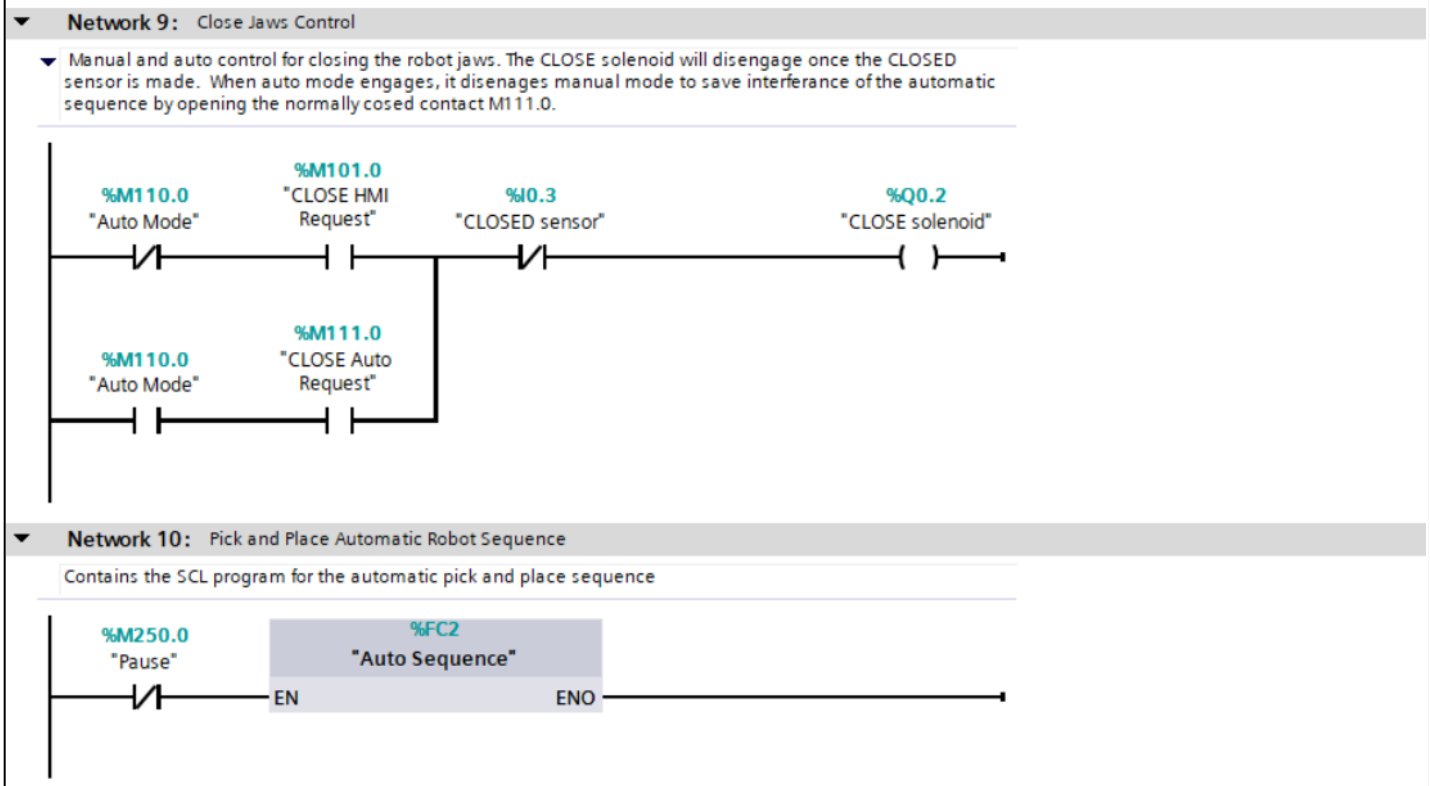


Figure 19 Networks 9 & 10

Network 11: Start Sequence Delay

The object sensor or start sequence memory address outputs a start memory output that engages start memory M200.0 in network 12.



Network 12: Start Sequence Delay

Start Memory M200.0 starts a timer of 2s which will then start the sequence, which also resets the start memory bit when complete.

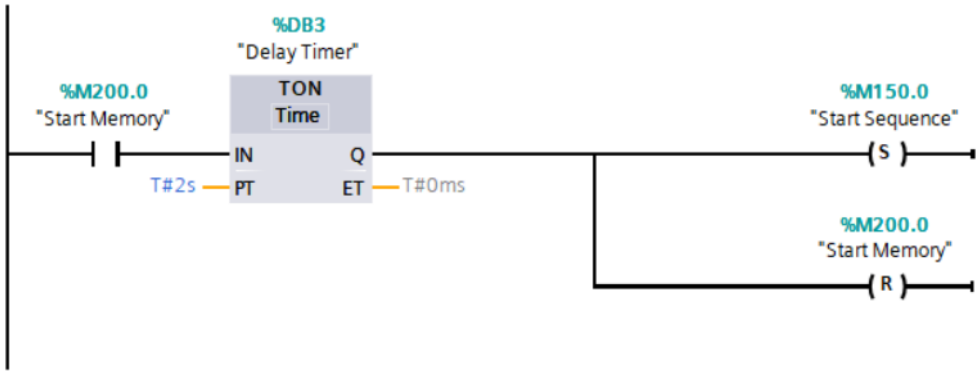


Figure 20 Networks 11 & 12

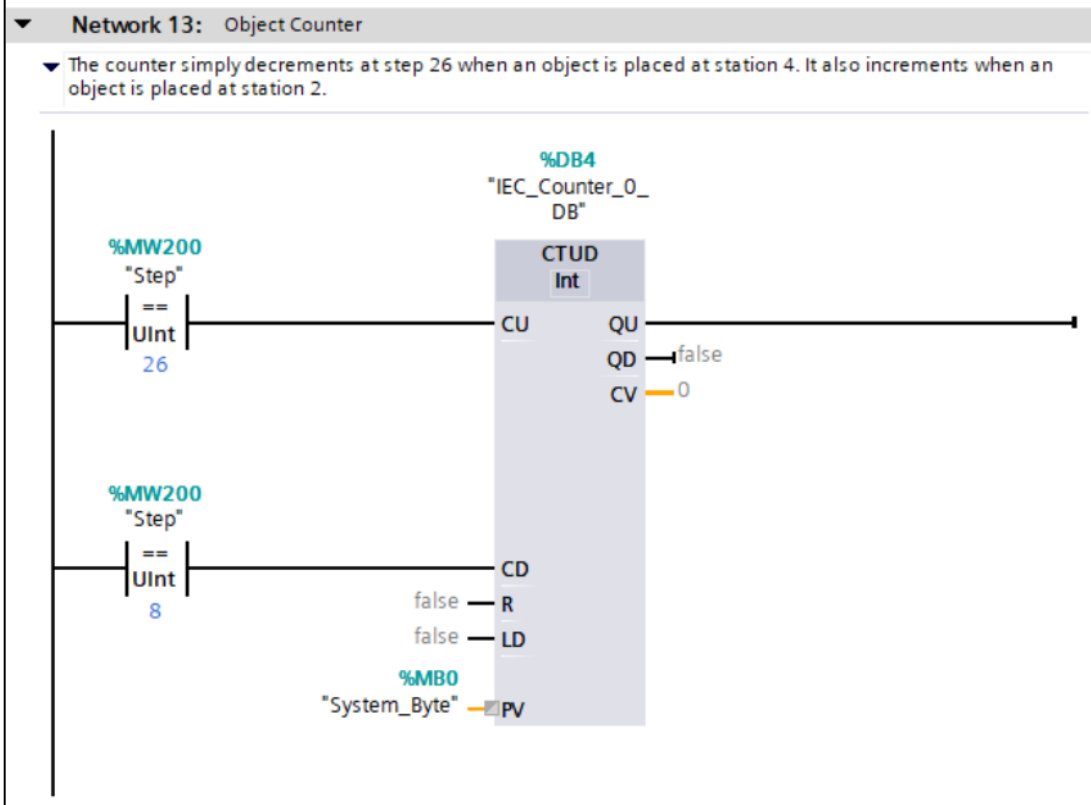


Figure 21 Network 13

Appendix D: - Simulation Testing Results and Evidence

Table 2 Simulation Testing and Results

Test Description	Expected Result	Actual Result	Pass or fail	Reference
1 - All manual controls must be fully functional. All sensors are to work correctly by closing the solenoid valves once the sensor is made. The solenoid valve indicators must illuminate when in operation.	Each sensor normally closed contact should break, stopping the solenoid from opening. The 'SOV' indicators on the HMI should illuminate when the solenoid is operating.	The solenoids for each manual control stopped when the indication lamps came on, meaning the sensors were working correctly as interlocks.	Pass	Figure 22
2 - The robot must not move left, right, in or out unless in the up position and must not move down unless located over a station.	When in manual control, if the robot is not in the up position, there should be no movement when the manual buttons are pressed, which will be seen on the HMI.	There was no movement in the robot when pressing each manual control except the grippers. No activity with the SOV lamps proved this. The evidence for this shown was for the down position. The	Pass	Figures 23, 24, 25 & 26
3 - The robot must move to the home position when auto mode is selected, and the HMI should display step '1'.	The robot must locate over station 1 regardless of any position the robot is currently in.	The solenoid valves operated automatically and the up, out, left and open sensor lamps illuminated, and the step display showed step '1', indicating the robot was in the home position.	Pass	Figures 27 & 28
4 - When there are 2 objects in the machine, the sequence must start at step '2' and go up until step '27', correctly moving the object at station 3 to 4 and moving the object at station 2 to 3 before moving the new object to station 2. The step must be reset to step '1'.	The HMI must display the step incrementing from 2 – 27 without it skipping steps.	The step display on the HMI correctly cycled through steps 2 – 27 without skipping any steps and ended at step 1.	Pass	Figures 29, 30 & 31
5 - When there is 1 object in the machine, the sequence must start at step '12' and go up until step '27', correctly moving the object at station 2 to 3 before moving the new object to station 2. The step must be reset to step '1'.	The HMI must display the step incrementing from 12 – 27 without it skipping steps	The step display on the HMI correctly cycled through steps 12 – 27 without skipping any steps and ended at step 1.	Pass	Figures 32, 33 & 34

6 - When there are 0 objects in the machine, the sequence must start at step '19' and go up until step '27', correctly moving the new object to station 2 The robot must be reset to step '1'.	The HMI must display the step incrementing from 19 – 27 without it skipping steps.	The step display on the HMI correctly cycled through steps 19 – 27 without skipping any steps and ended at step 1.	Pass	Figures 35, 36 & 37
7 - The sequence must be able to be paused and continued at any step.	The step display on the HMI should stop incrementing and there should be indication that the sequence is stopped from the sensors and the solenoid valves should not be in operation.	The sequence safely paused with the number on the step display being constant and there was no solenoid movement. The sensor indicators correctly illuminated their position.	Pass	Figures 38 & 39
8 - There must be a 2 second delay after the sequence is triggered by the object sensor.	The timer display on the HMI should display a 2 second delay and the steps should start incrementing.	The timer display correctly displayed 2 seconds until the sequence begun.	Pass	Figure 40
9 - The counter must be incremented at step '26' and decremented at step '8'.	The counter display on the HMI should increment at step '26' and decrement at step '8'.	The counter display on the HMI correctly incremented at step '26' and correctly decremented at step '8'.	Pass	Figures 41, 42 & 43

[Test 1]

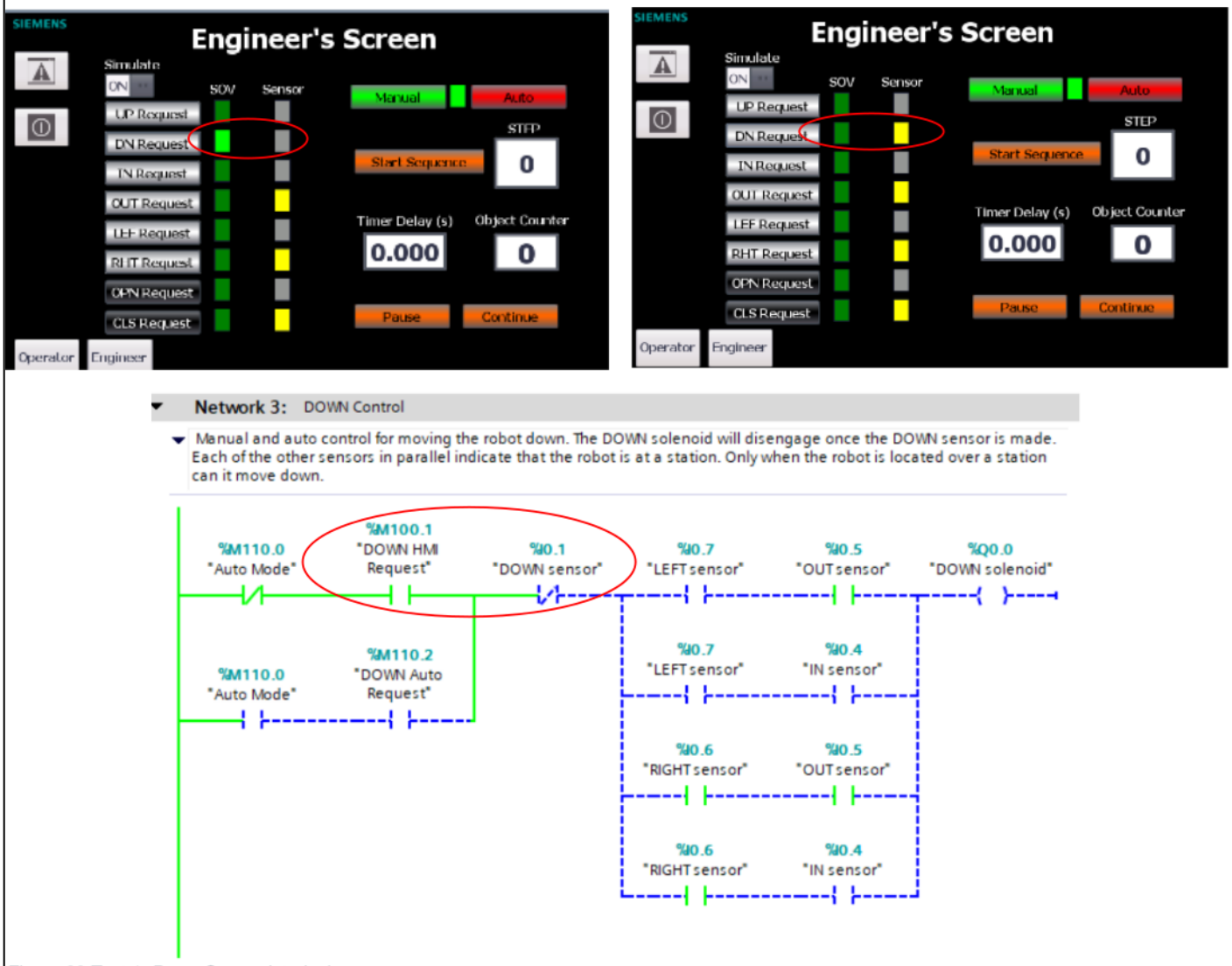


Figure 22 Test 1- Down Sensor Interlock

[Test 2]

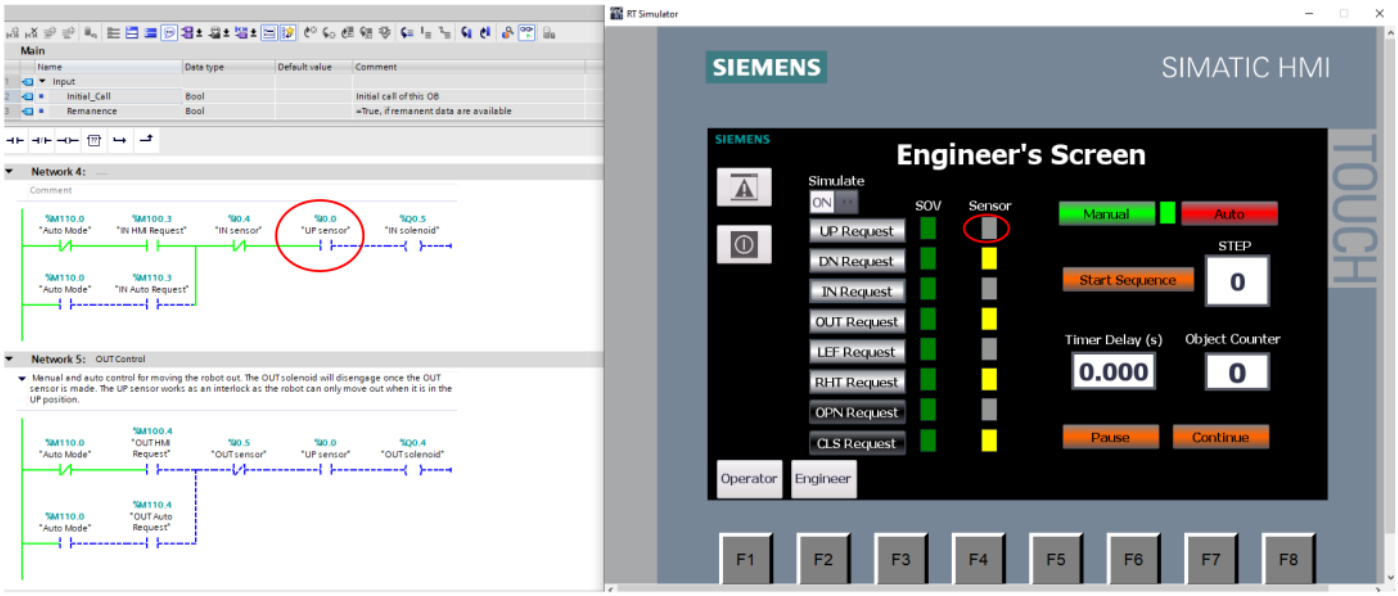


Figure 23 Test 2 - In Control

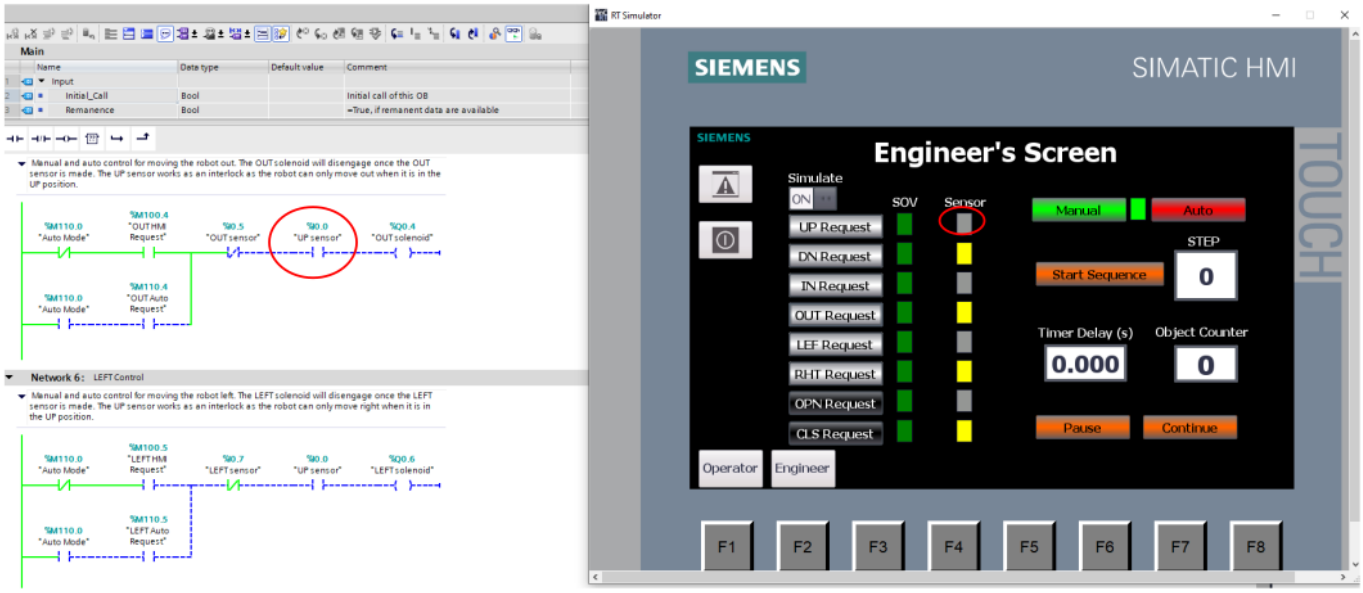


Figure 24 Test 2 - Out Control

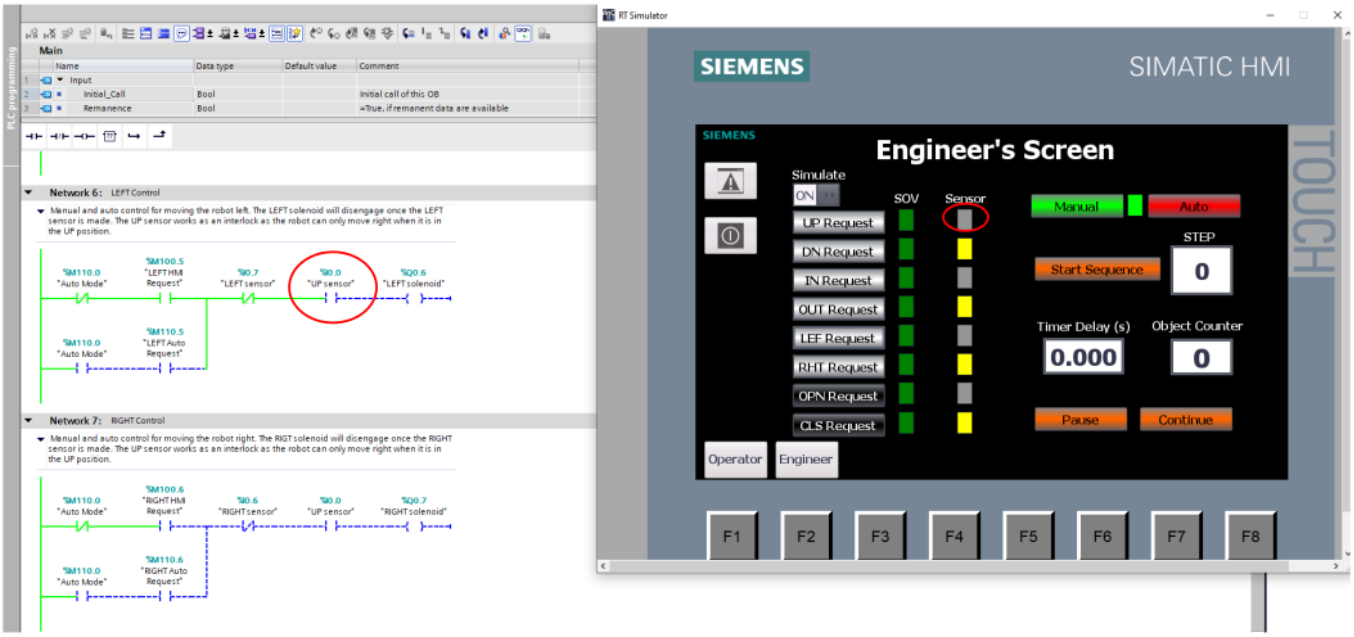


Figure 25 Test 2 - Left Control

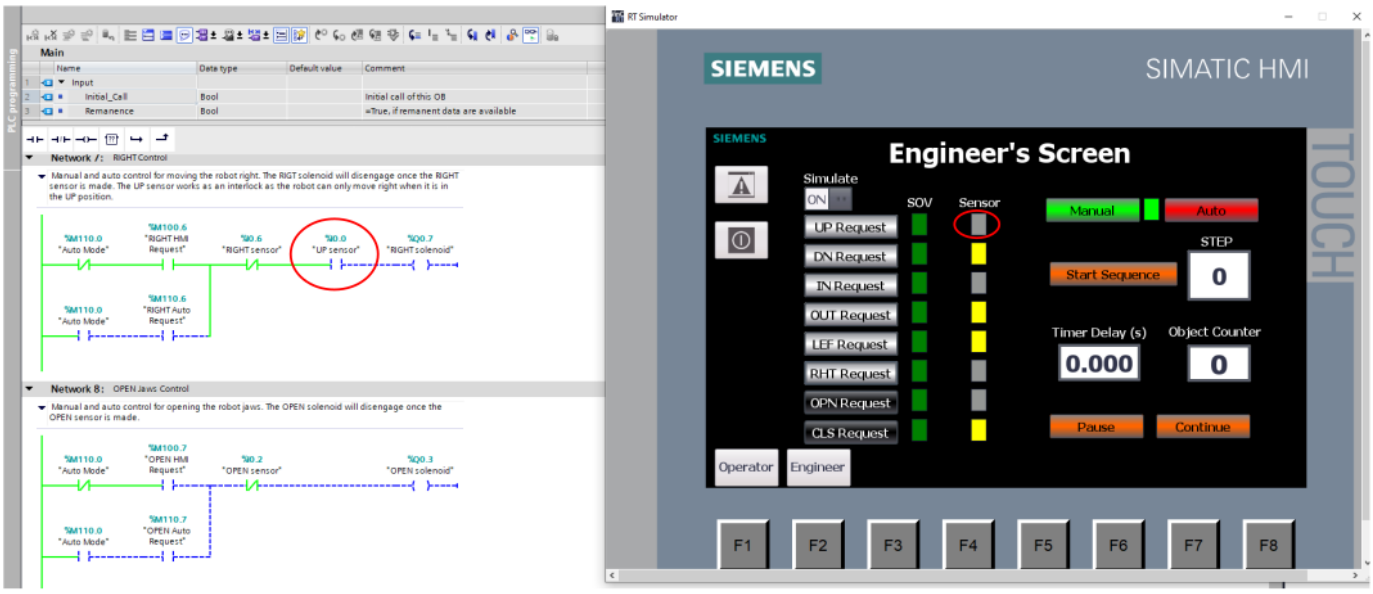


Figure 26 Test 2 - Right Control

[Test 3]

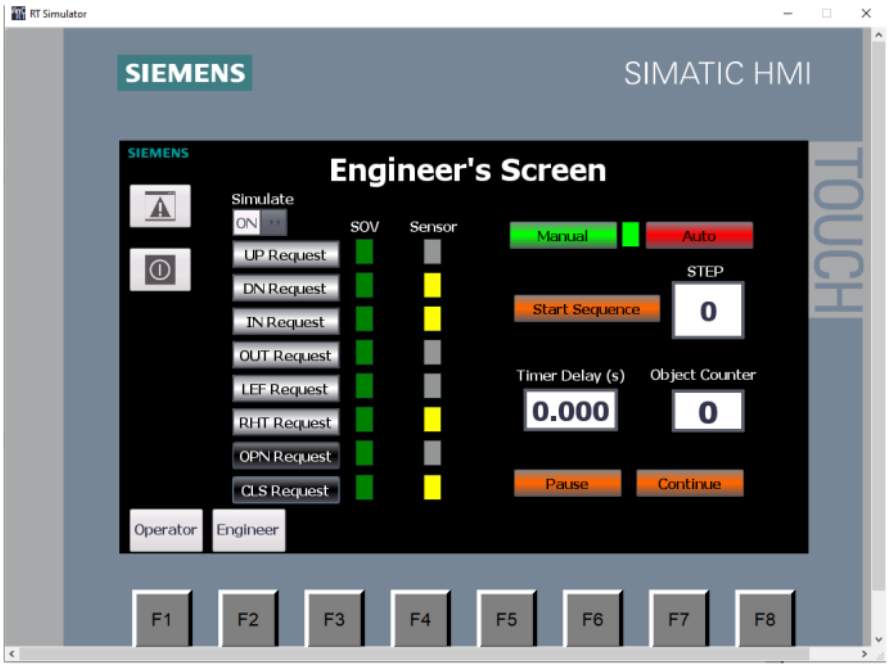


Figure 27 Test 3 - Out of Home Position

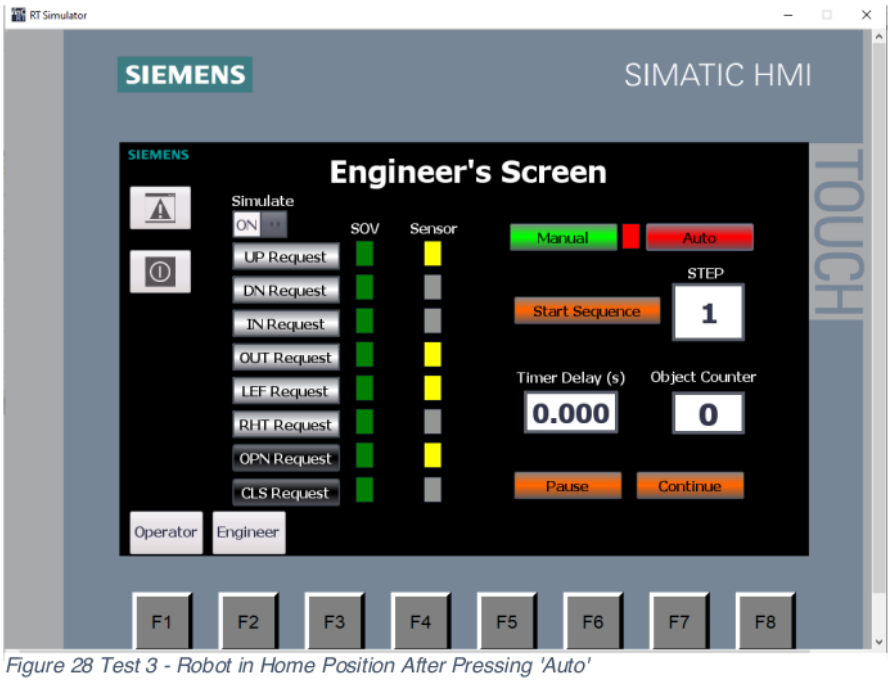


Figure 28 Test 3 - Robot in Home Position After Pressing 'Auto'

[Test 4]

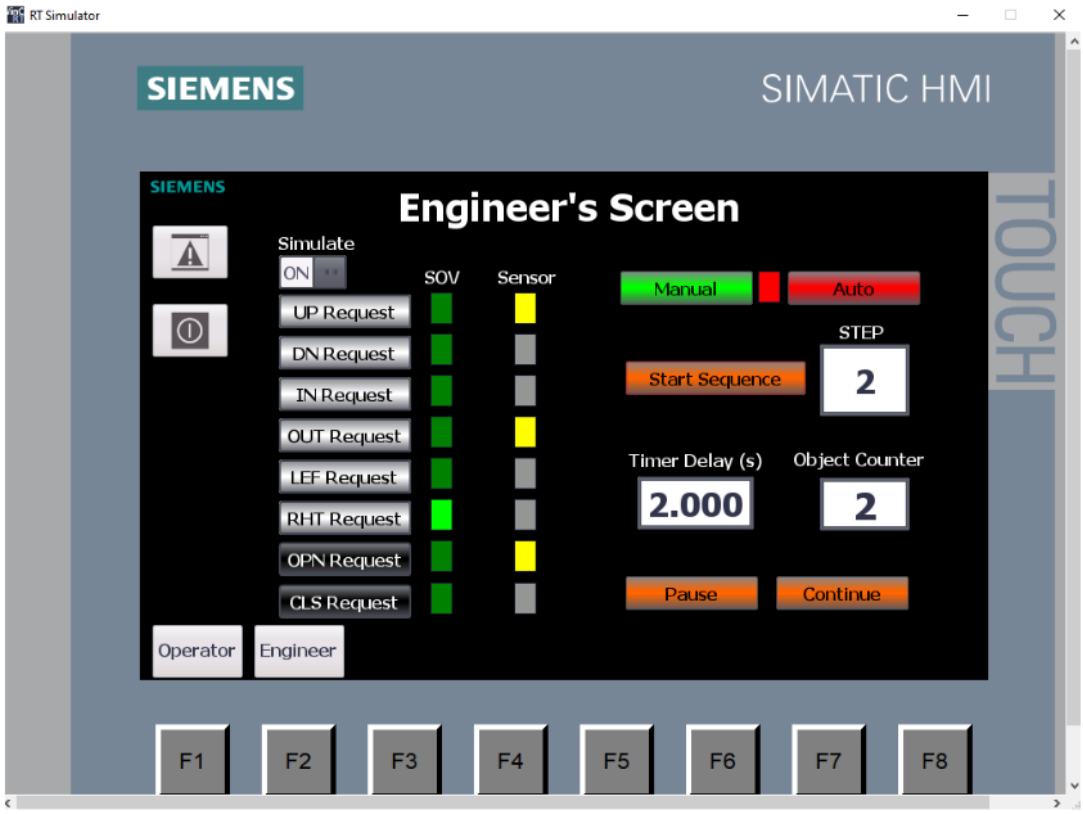


Figure 29 Test 4 - Beginning of Sequence

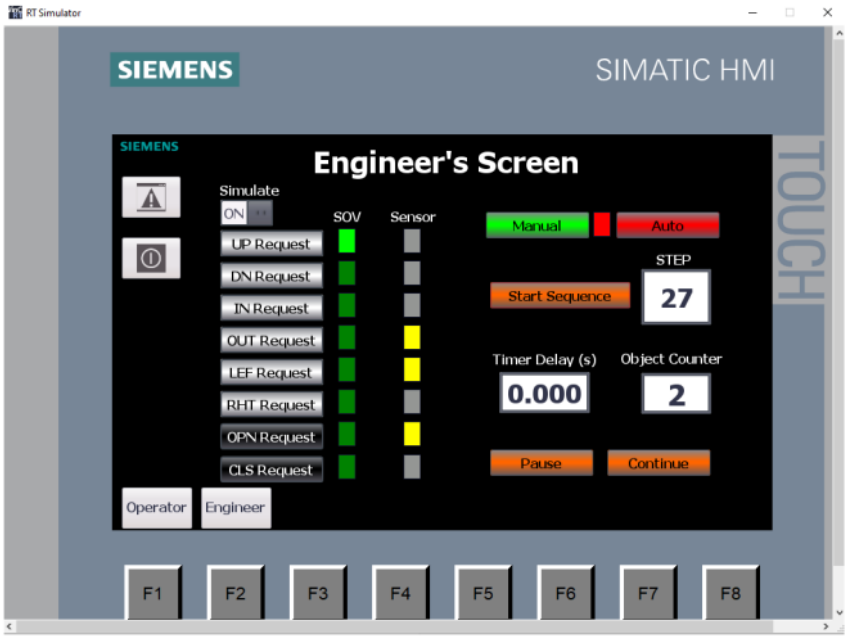


Figure 30 Test 4 - End of Sequence

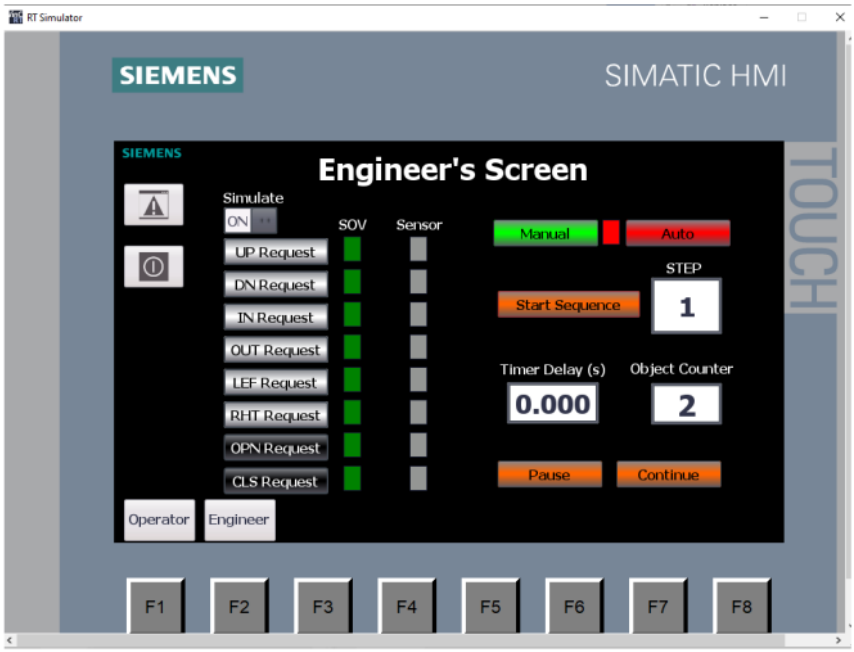


Figure 31 Test 4 - Robot Returned to Home Position

[Test 5]

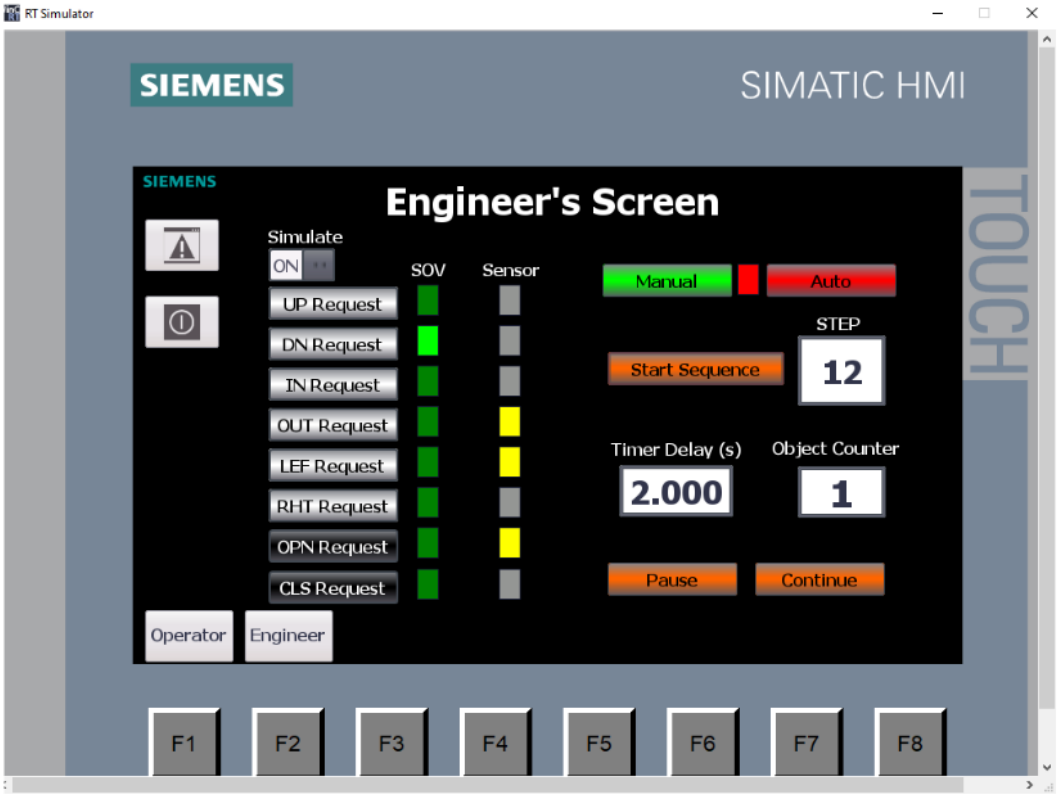


Figure 32 Test 5 - Beginning of Sequence

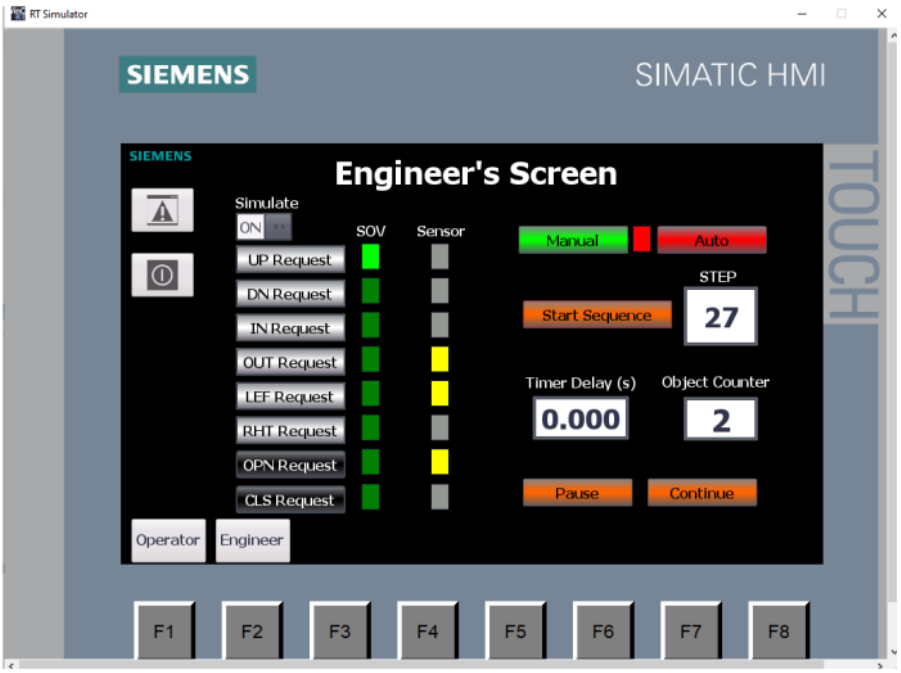


Figure 34 Test 5 - End of Sequence

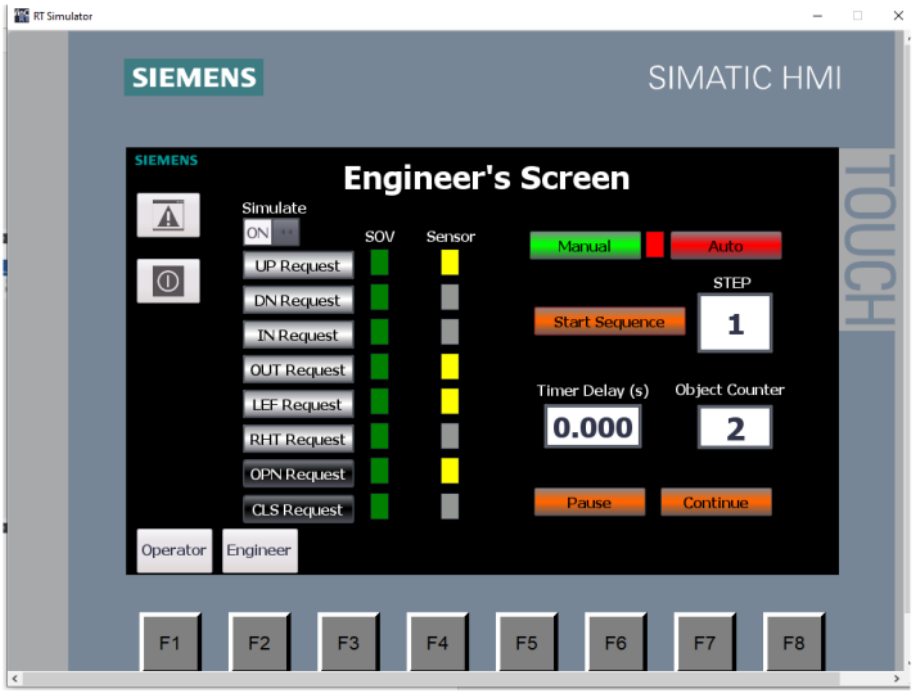


Figure 33 Test 5 - Robot Returned to Home Position

[Test 6]

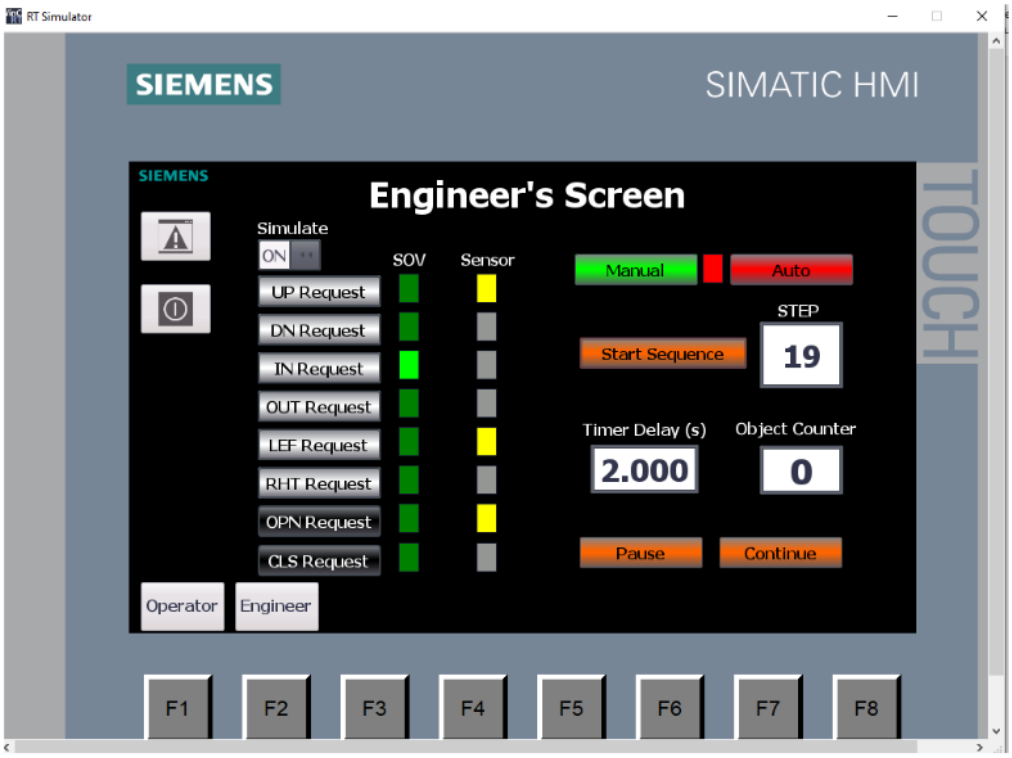


Figure 35 Test 6 - Beginning of Sequence

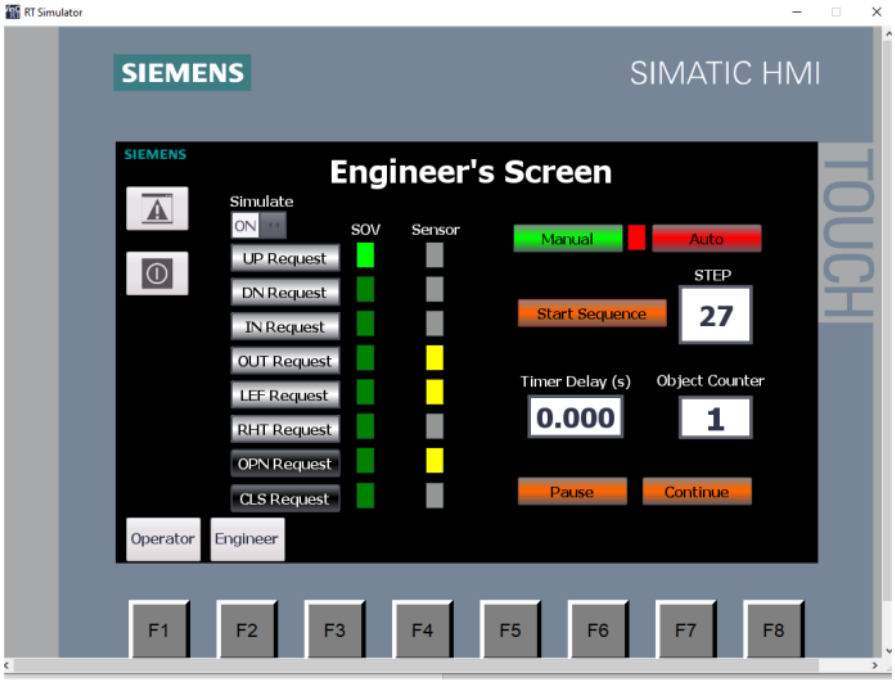


Figure 37 Test 6 - End of Sequence

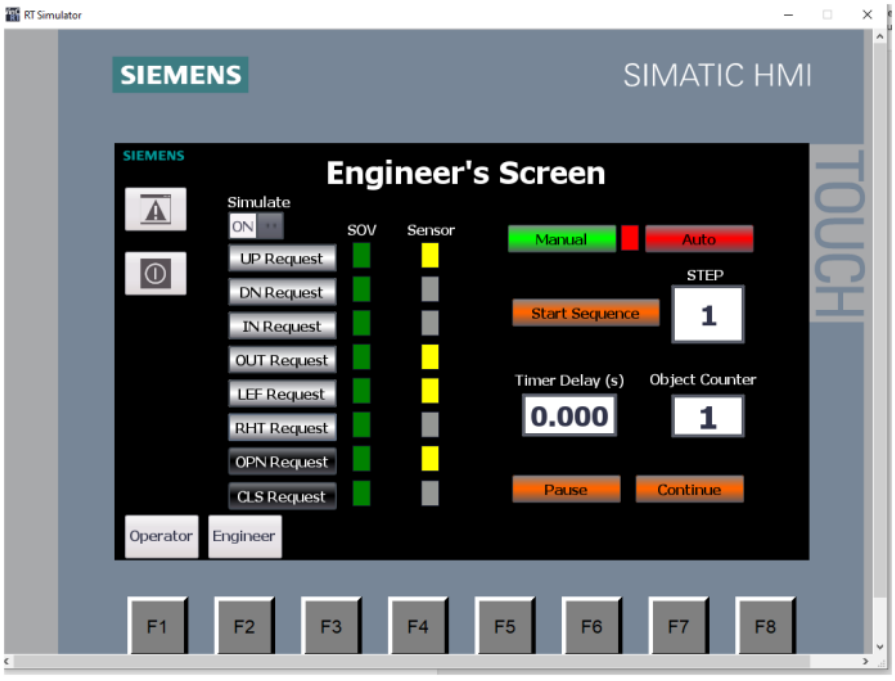


Figure 36 Test 6 - Robot Returned to Home Position

[Test 7]

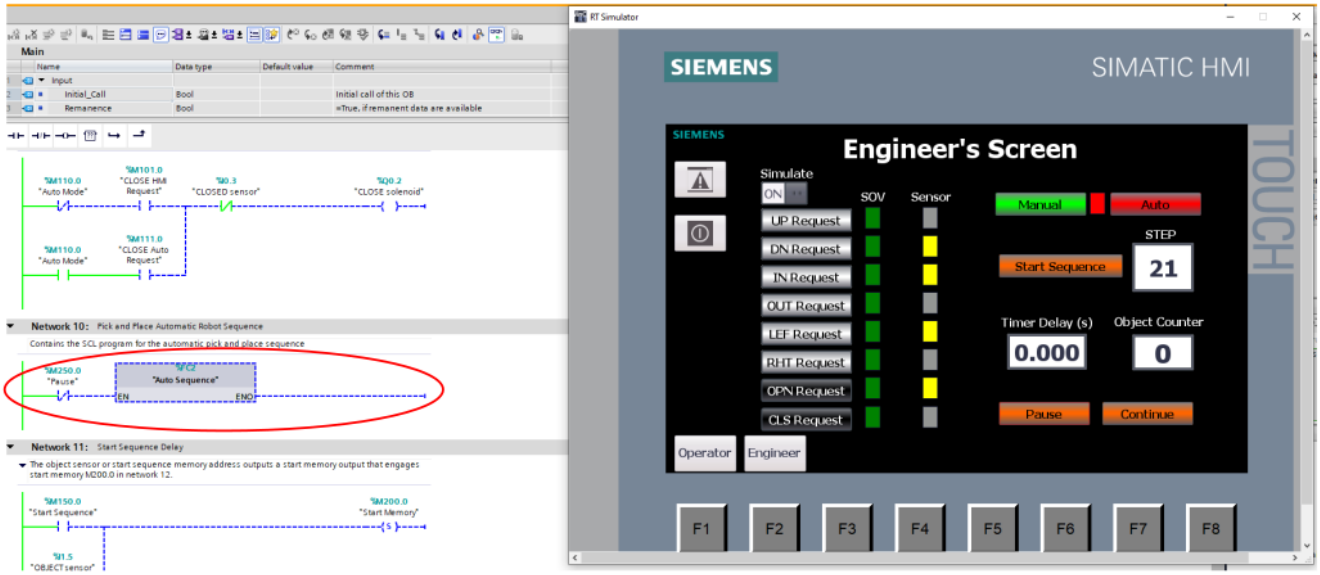


Figure 38 Test 7 - Sequence Paused

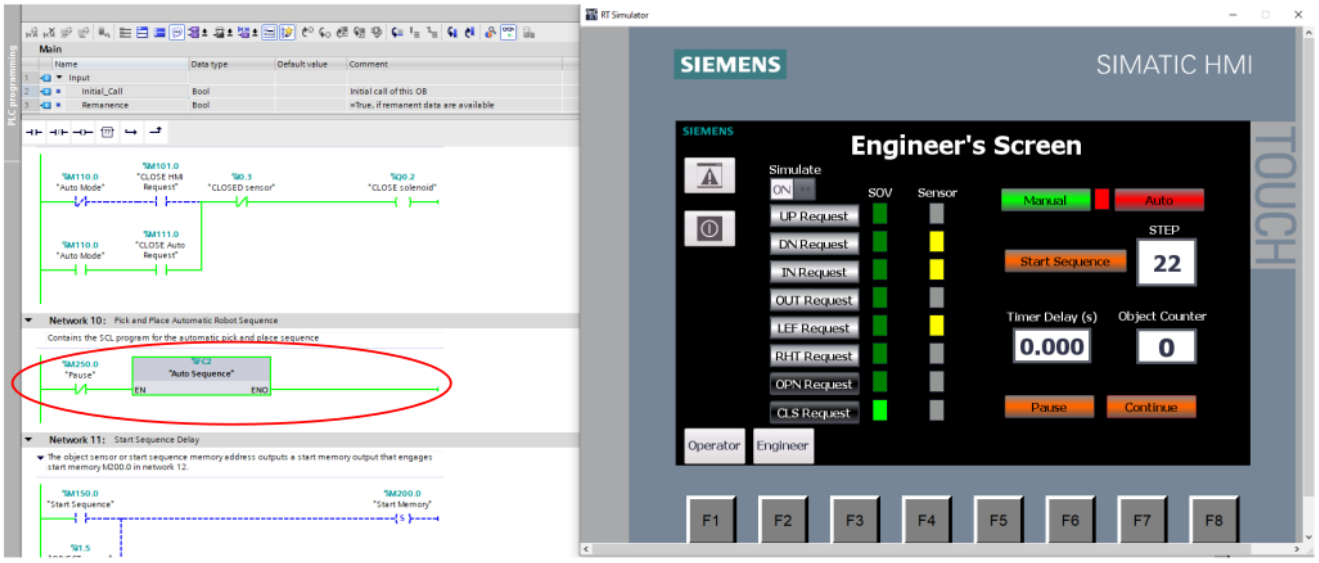


Figure 39 Test 7 - Sequence Continued

[Test 8]

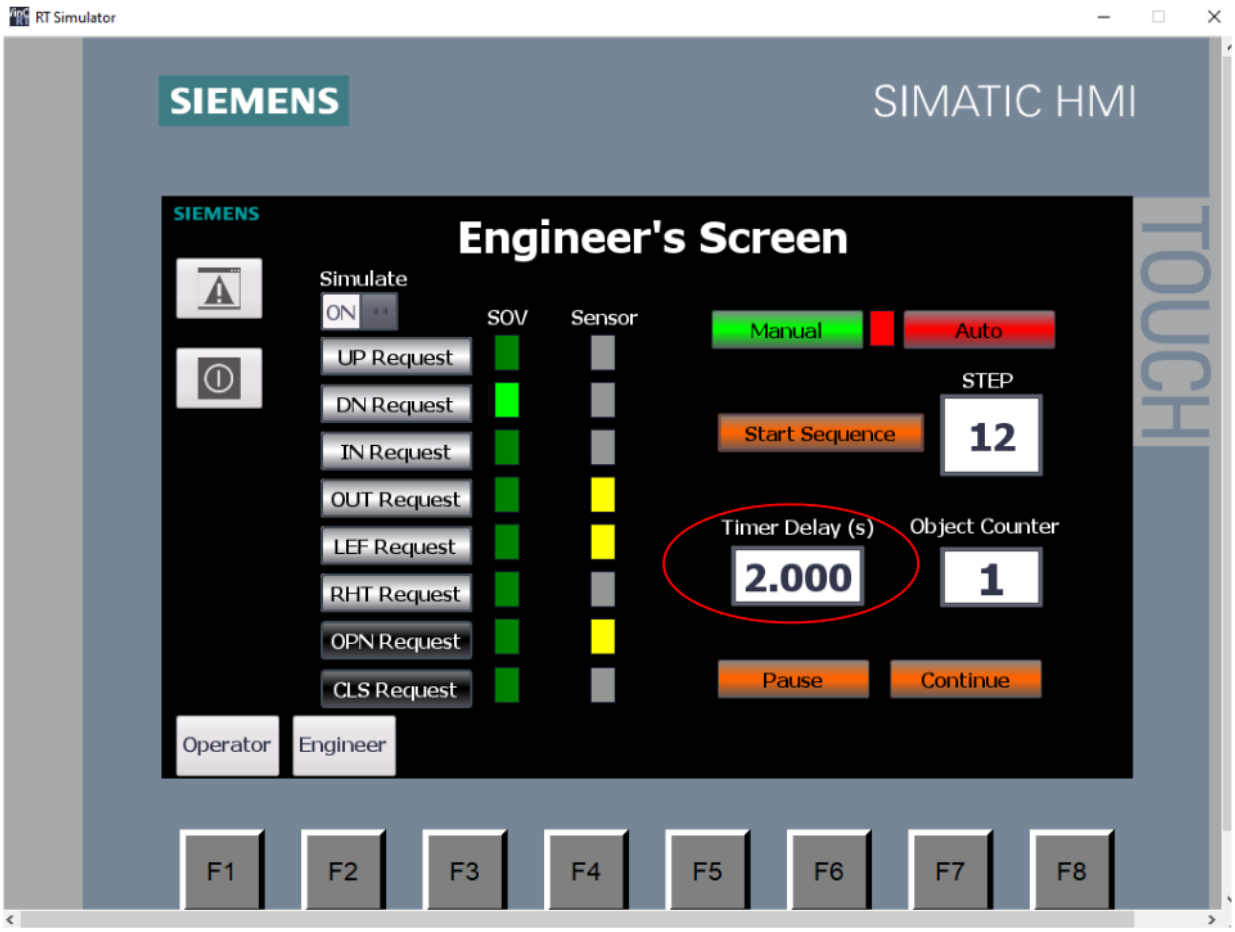


Figure 40 Test 8 - Timer 2 Second Delay

[Test 9]

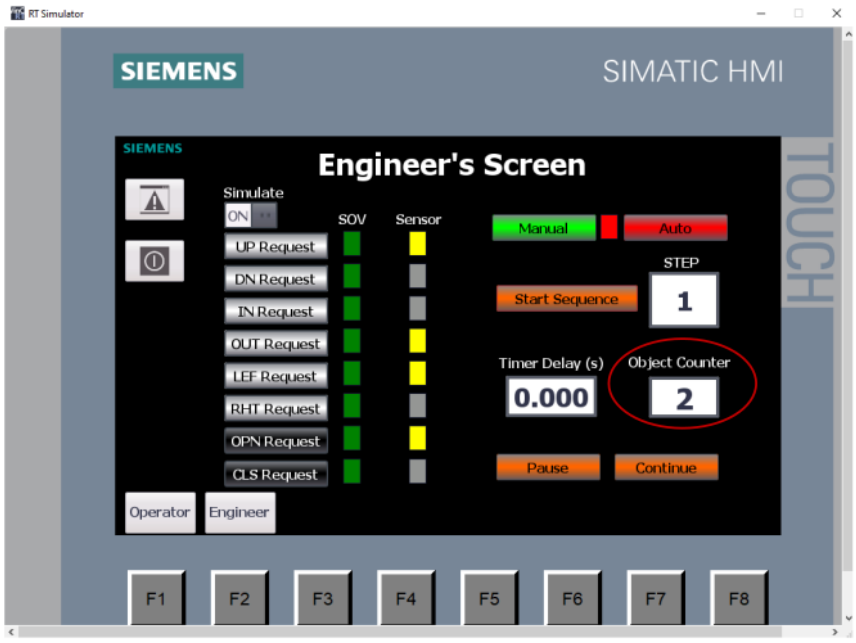


Figure 41 Test 9 - Beginning of Sequence, Counter = 2

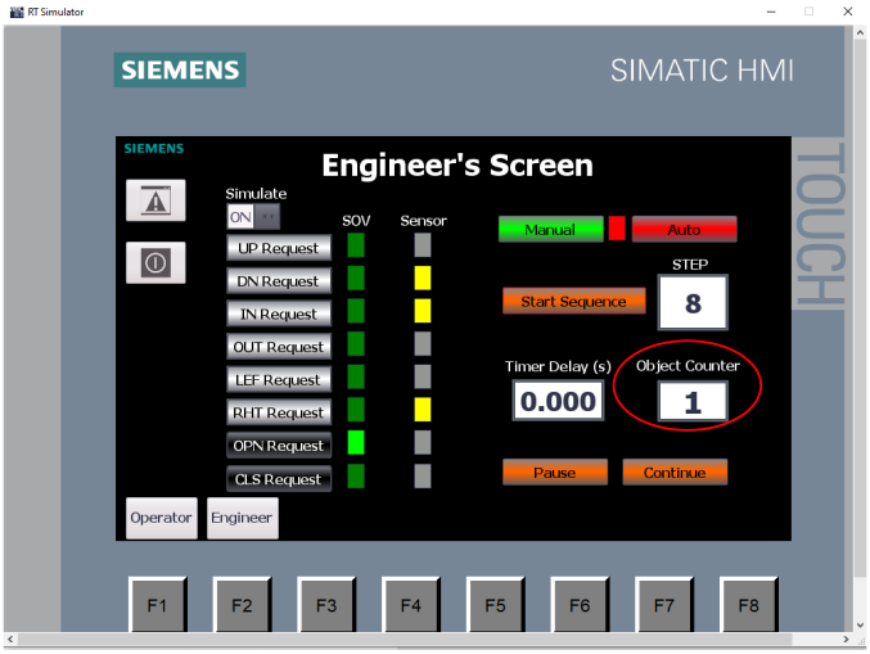


Figure 42 Test 9 - Step 8, Counter Decremented to 1

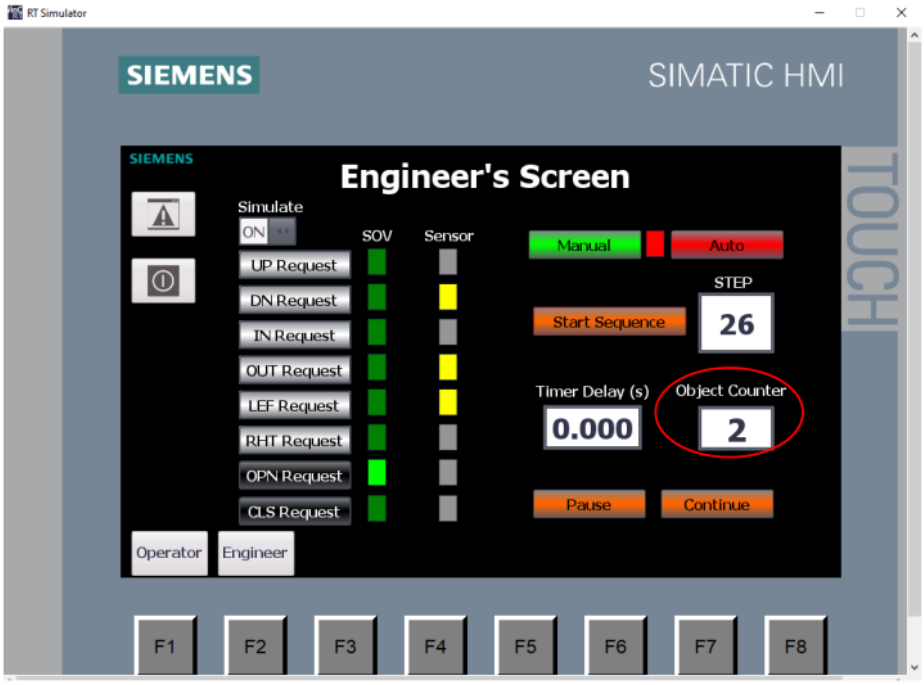


Figure 43 Test 9 - Step 26, Counter Incremented to 2

Appendix E: - Practical Testing Results and Evidence

Table 3 Practical Testing and Results

Test Description	Expected Result	Actual Result	Pass or fail	Reference
1 - All manual controls must be fully functional. All sensors are to work correctly by closing the solenoid valves once the sensor is made. The solenoid valve indicators must illuminate when in operation.	Each sensor normally closed contact should break, stopping the solenoid from opening. The 'SOV' indicators on the HMI should illuminate when the solenoid is operating.	The position sensors on the robot did not allow it to move any further when the sensor detected its position, whilst still pressing the manual requests. The evidence shown was for the down position. This was also true for every other manual request.	Pass	Figures 44
2 - The robot must not move left, right, in or out unless in the up position and must not move down unless located over a station.	When in manual control, if the robot is not in the up position, there should be no movement when the manual buttons are pressed, which will be seen on the HMI.	The robot did not move in the positions stated in the test description when still pressing the HMI manual requests, which means the up sensor was working correctly. The evidence sown demonstrates attempting to move the robot right when the robot is down, however the UP sensor prevented this.	Pass	Figures 45
3 - The robot must move to the home position when auto mode is selected, and the HMI should display step '1'.	The robot must locate over station 1 regardless of any position the robot is currently in.	The robot moved to the home position when the 'auto' mode was pressed, which means the automatic control was working correctly.	Pass	Figures 46 & 47
4 - When there are 2 objects in the machine, the sequence must start at step '2' and go up until step '27', correctly moving the object at station 3 to 4 and moving the object at station 2 to 3 before moving the new object to station 2. The step must be reset to step '1'.	The robot must complete the full sequence from step 2 – 27. There should remain 2 objects in the machine.	The robot completed the full sequence with the maximum number of objects in the machine and picked and placed the objects in their correct locations. The robot also returned to its home position.	Pass	Figures 48 & 49

5 - When there is 1 object in the machine, the sequence must start at step '12' and go up until step '27', correctly moving the object at station 2 to 3 before moving the new object to station 2. The step must be reset to step '1'.	The robot must complete the full sequence from step 12 – 27. There should remain 2 objects in the machine.	The robot completed the sequence from step 12 -27 correctly placing the objects in their assigned locations and returned to its home position.	Pass	Figures 50 & 51
6 - When there are 0 objects in the machine, the sequence must start at step '19' and go up until step '27', correctly moving the new object to station 2 The robot must be reset to step '1'.	The robot must complete the full sequence from step 19 – 27. There should remain 1 object in the machine.	The robot completed the sequence from step 19 – 27 correctly placing the object in its assigned location and returned to its home position.	Pass	Figures 52 & 53
7 - The sequence must be able to be paused and continued at any step.	The step display on the HMI should stop incrementing and there should be indication that the sequence is stopped from the sensors and the solenoid valves should not be in operation. Hence the robot should stop in its current position.	The robot stopped in the sequence once the 'pause' button was pressed and continued once the 'continue' button was pressed.	Pass	Figures 54 & 55
8 - There must be a 2 second delay after the sequence is triggered by the object sensor.	The timer display on the HMI should display a 2 second delay and the steps should start incrementing.	The only way the obtain a two second delay was to manually start the sequence from the HMI screen. The robot being tested had the sensor in a different location where the object would simply roll past the sensor. This means the delay triggered by the sensor could not be fully tested.	Fail	Figure 56
9 - The counter must be incremented at step '26' and decremented at step '8'.	The counter display on the HMI should increment at step '26' and decrement at step '8'.	The counter correctly incremented and decremented at the assigned steps and the robot had the correct number of objects in the machine.	Pass	Figures 57, 58 & 59

[Test 1]



Figure 44 Test 1 – Testing Down Sensor Interlock

[Test 2]

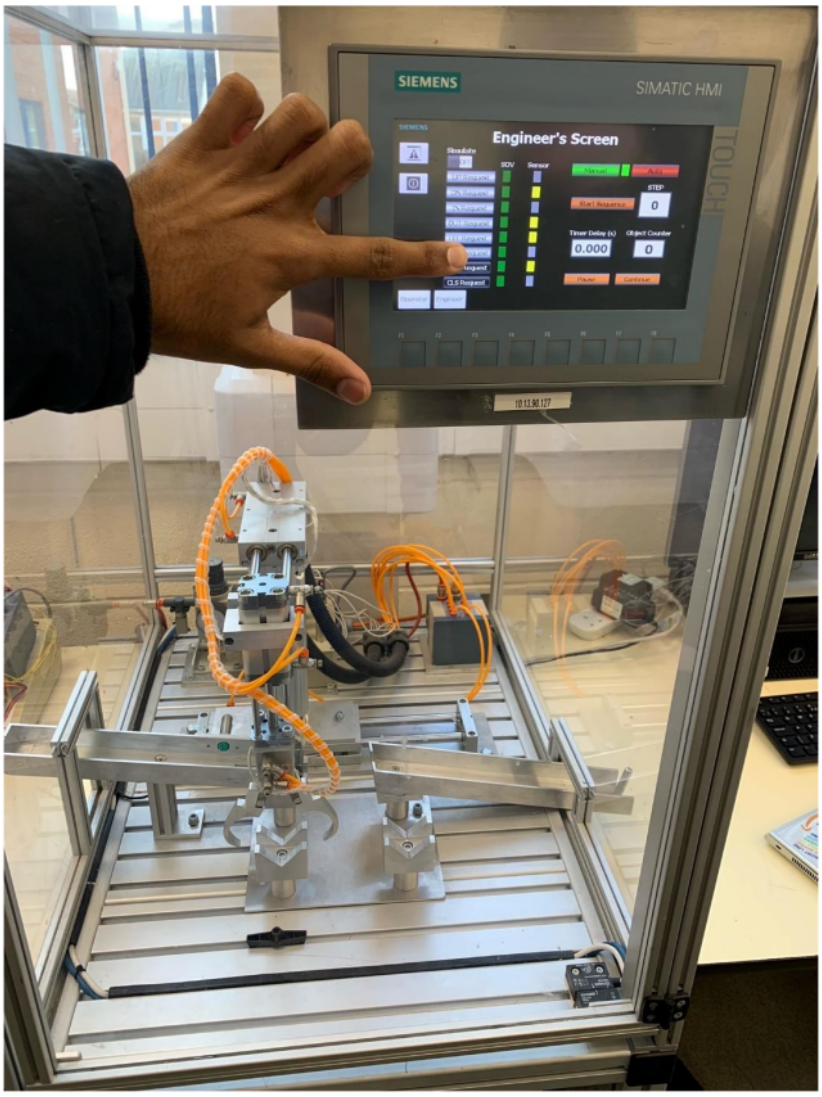


Figure 45 Test 2 - Right Manual Request

[Test 3]



Figure 46 Test 3 - Robot Away from Home Position and in Manual Mode



Figure 47 Test 3 - Robot in The Home Position After Pressing 'Auto'

[Test 4]

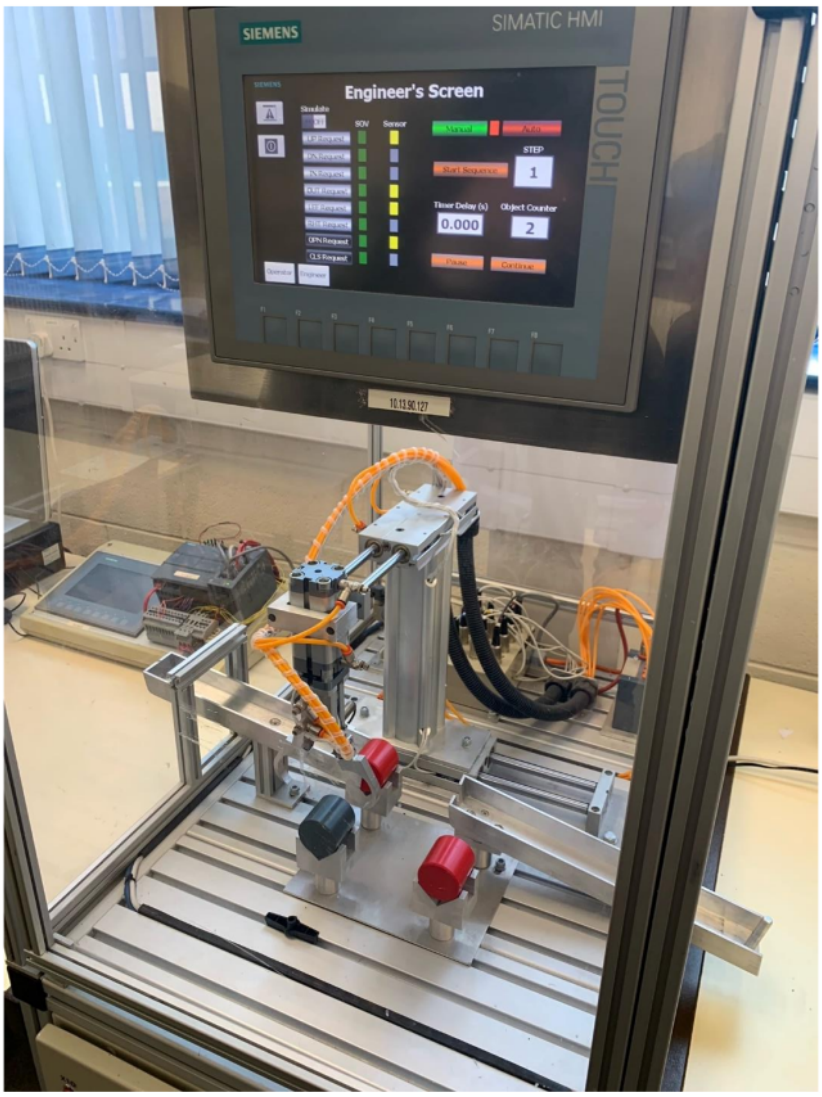


Figure 48 Test 4 - 2 Objects in The Machine Pre-Sequence

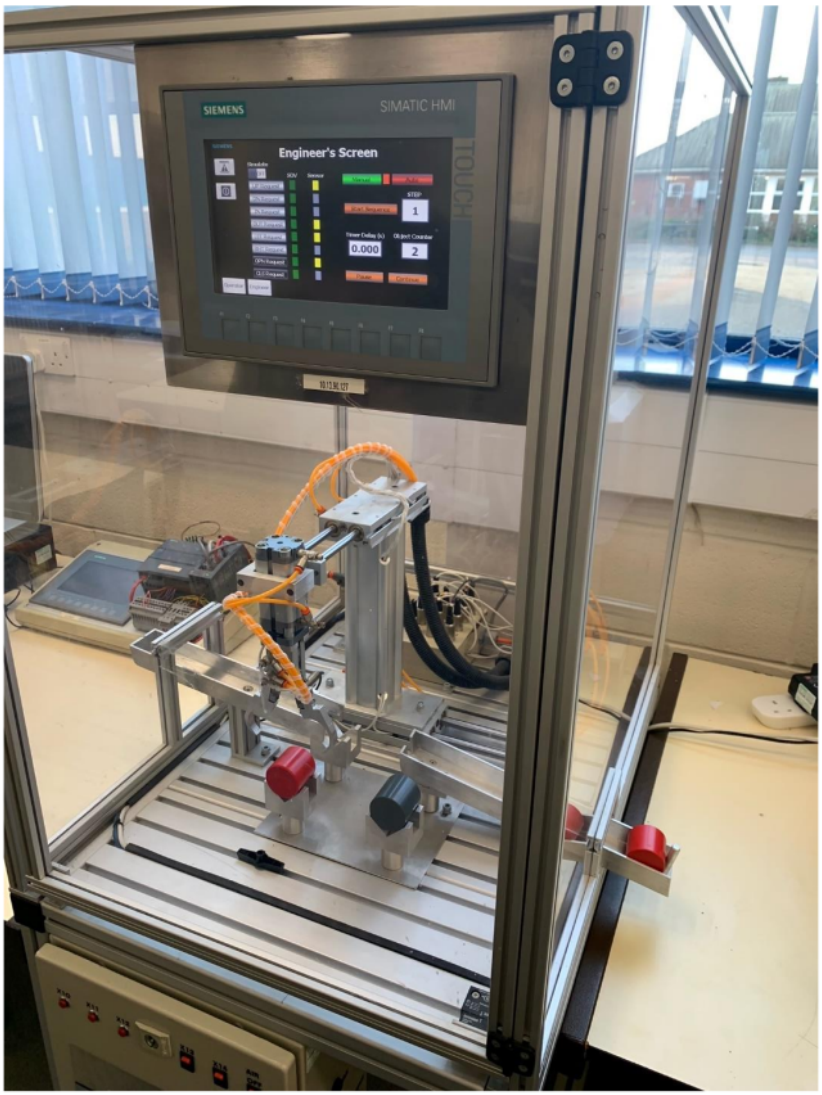


Figure 49 Test 4 - 2 Objects in The Machine Post-Sequence

[Test 5]

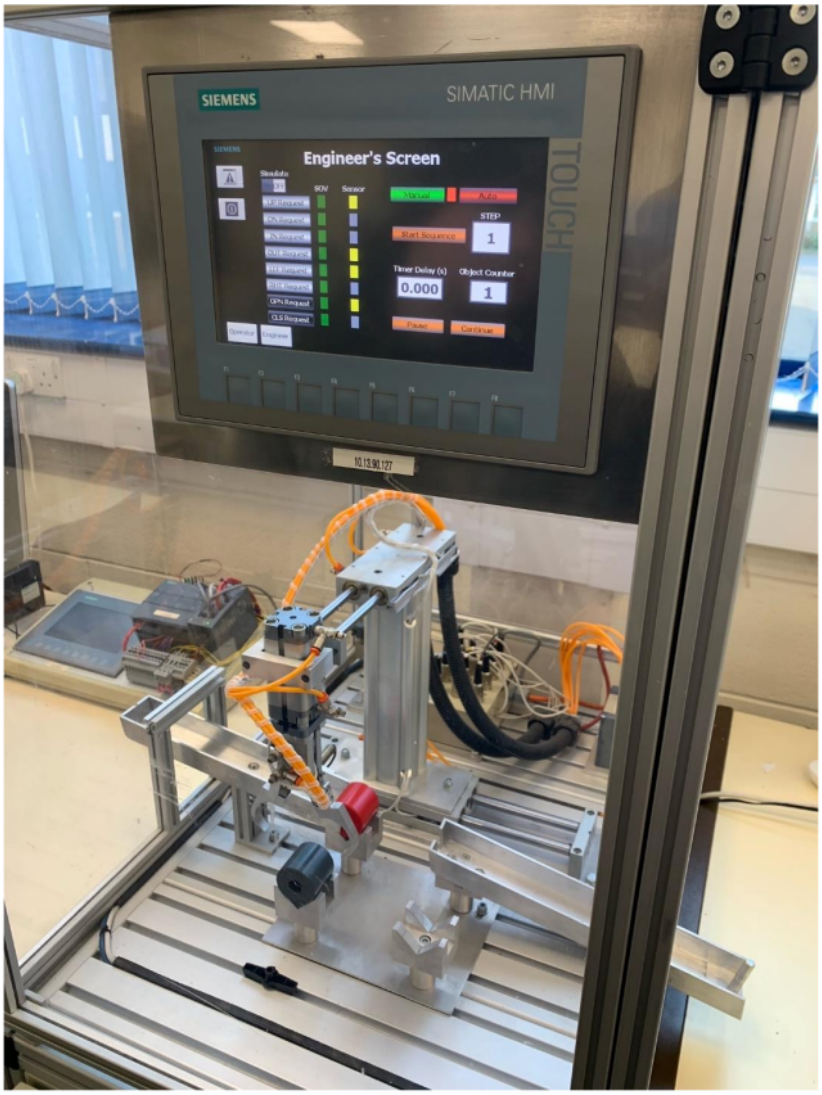


Figure 50 Test 5 - 1 Object in The Machine Pre-Sequence

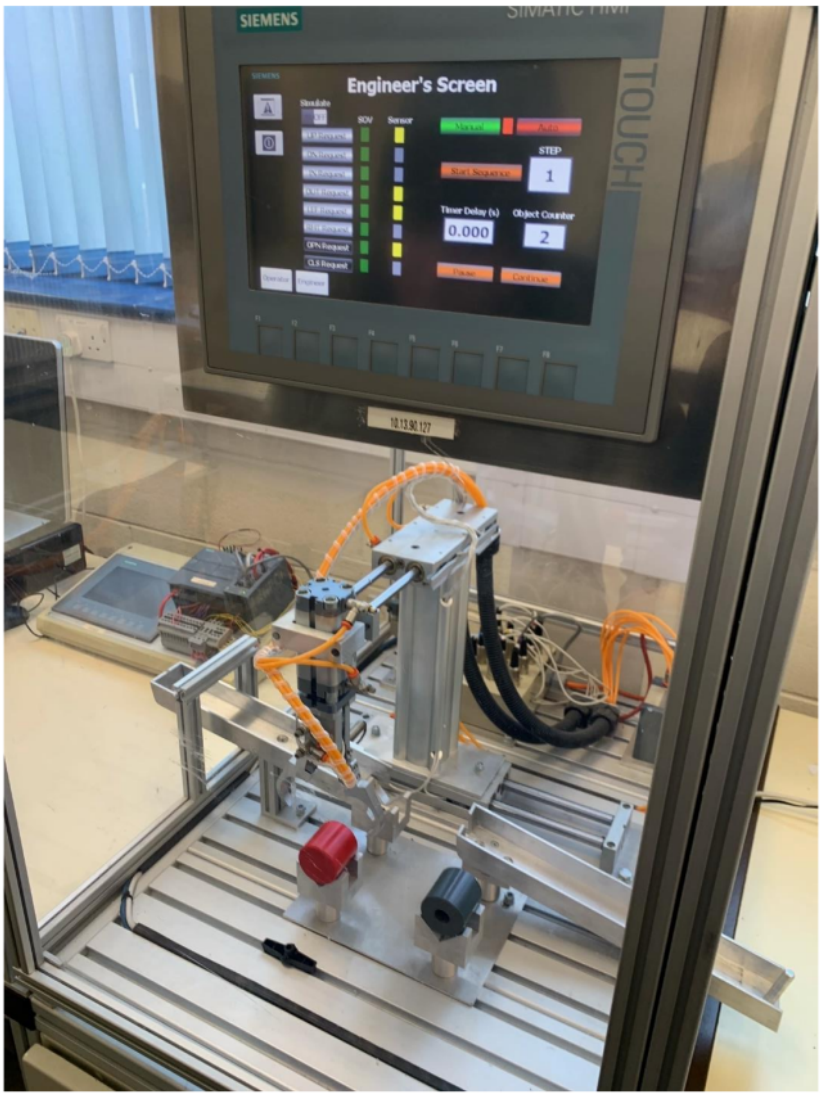


Figure 51 Test 5 - Two Objects in The Machine Post-Sequence

[Test 6]



Figure 52 Test 6 - No Objects in The Machine Pre-Sequence



Figure 53 Test 6 - 1 Object in The Machine Post-Sequence

[Test 7]

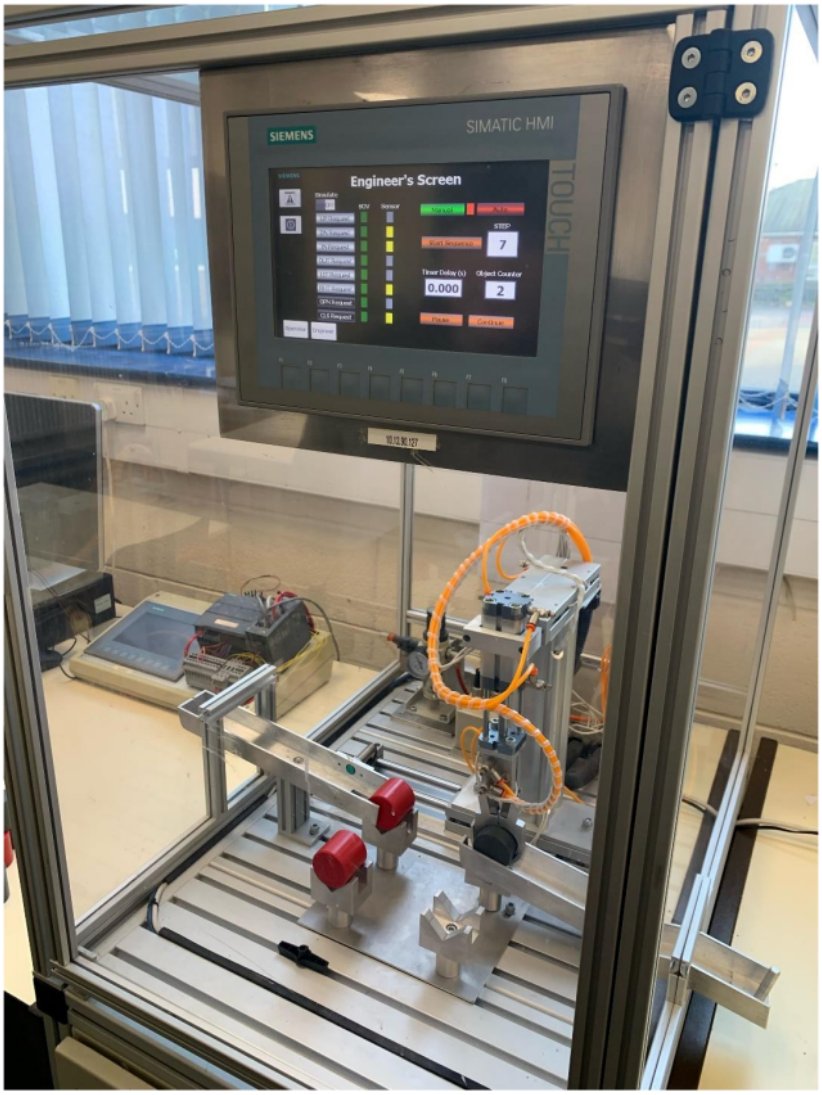


Figure 54 Test 7 - Sequence Paused After Pressing 'Pause' Button



Figure 55 Test 7 - Sequence Continued After Pressing 'Continue' Button

[Test 8]

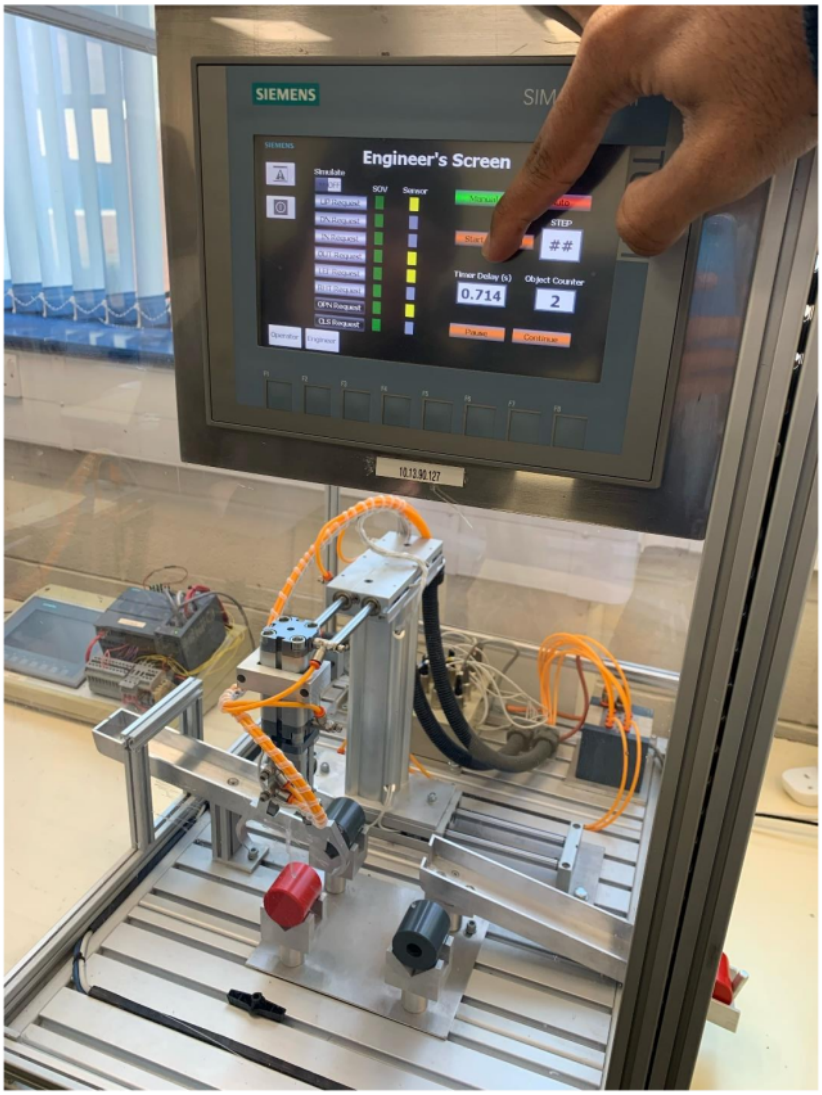


Figure 56 Test 8 - Timer Delay with HMI Displaying Timer

[Test 9]

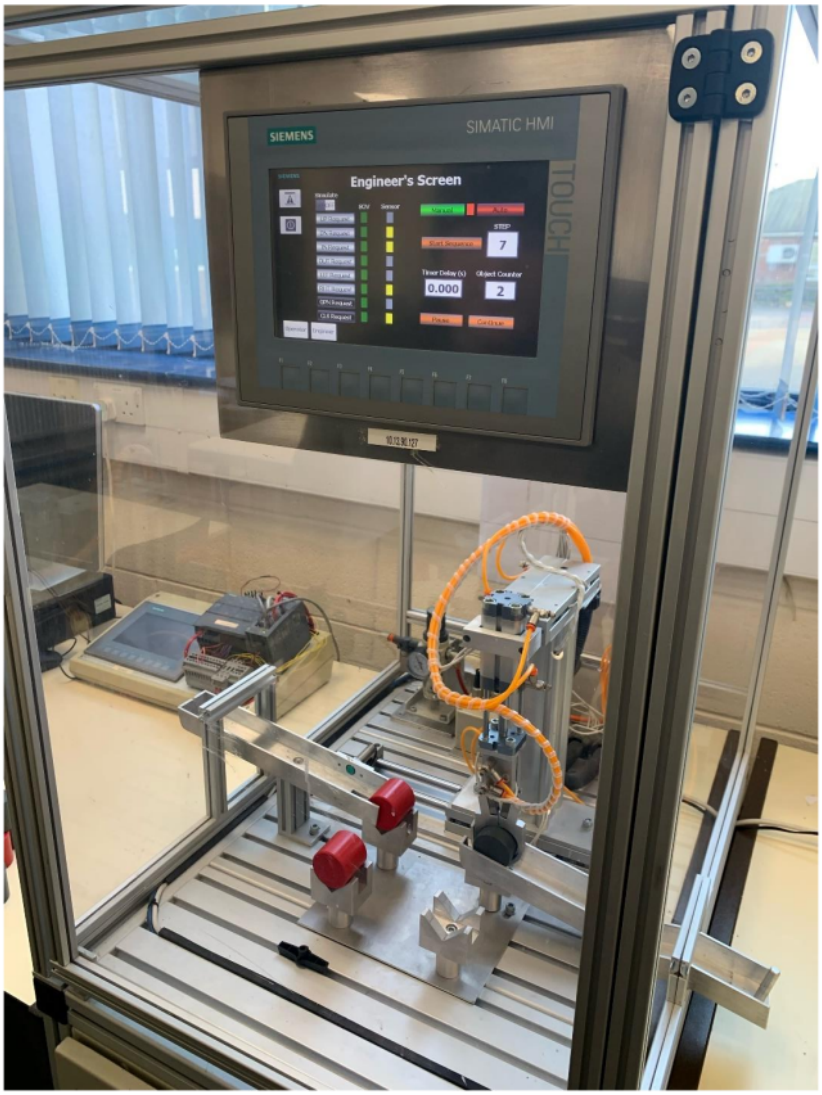


Figure 57 Test 9 - Object Counter Reading 2 Objects Before Step 8



Figure 58 Test 9 - Object Counter Decremented After Step 8



Figure 59 Test 9 - Object Counter Incremented After Step 26

2224559

GRADEMARK REPORT

FINAL GRADE

80/100

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

QM

Good Structure

Report is structured exactly as stated in the assignment brief.

PAGE 3

PAGE 4

PAGE 5

PAGE 6

Text Comment. already exists

Text Comment. This is a good plan

Text Comment. Nice idea, but not required here

PAGE 7

Text Comment. Allen Bradley states that " .....

Text Comment. year?

Text Comment. why no date?

Text Comment. only include page numbers for direct quotes

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

Text Comment. only include page numbers for direct quotes

PAGE 8

Text Comment. quotes should start sooner

Text Comment. page number should be given with direct quotes

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

Text Comment. page numbers are for direct quotes

PAGE 9

PAGE 10

Text Comment. part numbers are confusing, stick with the model number

Text Comment. you should consider explansion IO instead

PAGE 11

Text Comment. page number?

PAGE 12

PAGE 13

Text Comment. excellent information

PAGE 14

QM

Good explanation given

You have clearly put much effort into this explanation, well done.

PAGE 15

QM

Good explanation given

You have clearly put much effort into this explanation, well done.

QM

Good explanation given

You have clearly put much effort into this explanation, well done.

QM

Good explanation given

You have clearly put much effort into this explanation, well done.

PAGE 16

PAGE 17

Text Comment. The brief asked for aa 2 second delay and you should have design the program to start with either robot

Text Comment. You should have used a memorise bit as the object rolls past the sensor

PAGE 18

QM

Good explanation given

You have clearly put much effort into this explanation, well done.

QM

Good conclusions

You have shown understanding here from these observations and deductions

PAGE 19

Text Comment. excellent ideas presented

Text Comment. The Siemens S7-1511 would also allow use of SFC

PAGE 20

PAGE 21

QM

Correct method of listing.

You have listed your sources correctly according to the Harvard method.

QM

Good range of sources

You have demonstrated reading from a good range of sources shown by your list below

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

Text Comment. this should return to step 0

PAGE 29

PAGE 30

PAGE 31

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

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70

GRADING FORM: NEW CSG-RUBRIC

MOHAMMED IBRAHIM	**80**

1. ENGAGEMENT WITH LITERATURE

A excellent research section with evidence of reading from a very wide range of sources cited and listed, mostly but not always, correctly using the Harvard method. 13/20 — **13**

2. KNOWLEDGE AND UNDERSTANDING

Excellent knowledge and understanding demonstrated in your research findings, hardware investigation, IO list and program design explanations 17/20 — **17**

3. COGNITIVE AND INTELLECTUAL

Excellent simulation and practical test results analysis, discussion and conclusions/recommendations 17/20 — **17**

4. PRACTICAL APPLICATION



Excellent program design, simulation and practical testing methods and presentation of results 17/20

17

5. TRANSFERABLE SKILLS



An excellent work plan and explanation of team working. Report is well structured, written and presented with clear diagrams. Word count not exceeded. 16/20

16