

2024421

by Matthew Perry

FILE	16803_MATTHEW_PERRY_2024421_178848_784629063.PDF (5.94M)		
TIME SUBMITTED	08-FEB-2021 10:34AM (UTC+0000)	WORD COUNT	4097
SUBMISSION ID	144388809	CHARACTER COUNT	21575

Course: HNC Electrical and Electronic Engineering

Module: PROGRAMMABLE LOGIC CONTROLLERS

Assignment Title: Pick and Place Robot

Name: Matthew Perry

University number: 2024421

Date Submitted: 08/02/2021

Word Count: 2438

Table of Contents

Good Structure

1. Research.....	3
2. Hardware Investigation, Circuit Diagram and I/O List	6
3. Program Design and Explanation	8
4. Simulation Testing.....	17
5. Practical Testing and Team Working	24
6. Discussion, Conclusions and Recommendations	25
References	26
Appendix	27

Figure 1; Operator HMI screen	8
Figure 2; Engineer HMI screen	9
Figure 3; Object counter	12
Figure 4; Object timer	12
Figure 5; SFC Chart	16
Figure 6; Test table of pause button, and ACT	22
Figure 7; Manual interlock simulation testing	23
Figure 8; PLC wiring diagram	28

Table 1; PLC Comparison of Unitary, Modular and Rack Mounted	4
Table 2; Pick and place robot requirements in comparison to chosen PLC; Siemens S7-1214C.....	5
Table 3; Input list	7
Table 4; Output list.....	7
Table 5; Operator HMI; Start and pause	8
Table 6; Operator HMI; Object counter and step	8
Table 7 ; Engineer HMI ; Buttons	10
Table 8; Engineer HMI; Counter and step.....	10
Table 9 ; Engineer HMI; Switches.....	10
Table 10; Engineer HMI ; Pilot lights	11

1. Research

Use of 11pt Arial, full justification with 1.5 line spacing allows easier reading and

Ensuring that an appropriate PLC is chosen for the robot control panel is important. Siemens has a wide range of PLCs available, so it is essential to investigate into all types.

There are three different types of PLCs. To understand what is meant by 'types', this section will explain each one below.

A unitary PLC is a controller which is small and usually enclosed in a fixed box. They consist of inputs, outputs, power supply connection and CPU. They are unable to have expandability modules added and are best suited for applications which have a low input and outputs. The scan rate may be slower in some cases, meaning it could not complete fast-moving applications. The benefit of unitary PLC's is price, they are usually the cheapest option of the three types. An example of a Unitary PLC is a Siemens S7-1211C (Siemens, 2021).

Correctly cited

Modular PLCs are contained in an enclosed box which is known as a base unit and are fitted onto a DIN rail. They are larger than a unitary PLC and can be expanded on with expansion modules which are slid onto the DIN rail on either side of the base unit. They consist of inputs, outputs, power supply connection and CPU. Modular PLCs have more input and output options as a base module, but also has the option to have extra I/O with expansion modules added (Bolton, 2015). There are more features and expansion modules available which can be viewed in Table 1. This type of PLC is more costly than a unitary, however cheaper than a rack mounted. An example of a Modular PLC is a Siemens S7-1511C (Siemens, 2021).

Correctly cited

Grocer's apostrophe

Correctly cited

Rack mounted PLC's are held together by the rack, the modules such as a power supply, CPU, input/output modules are separate. When the modules are slid into the rack they are connected, the power and data interfaces are supplied by copper conductors in the backplane of the rack. Allowing all the modules to be separate allows the user to define what the PLC will consist of, whether they want more specific types of I/O or memory etc determined by the application (Bolton, 2015). Rack mounted PLCs can be more expensive compared to unitary and modular. An example of a Rack mounted PLC is a Siemens S7-400 (Siemens, 2021).

Correctly cited

Correctly cited

By investigating the three examples stated for each type of PLC, a table of characteristics has been compiled.

<u>Type</u>	<u>Unitary</u>	<u>Modular</u>	<u>Rack-mounted</u>
PLC	Siemens S7-1211C	Siemens S7-1511C	Siemens S7-400 with CPU 412H
Digital inputs	6	16	65 536
Digital outputs	4	16	65 536
Analog inputs	2	5	4 096
Pulse outputs	2	2	4 096
PWM outputs	6, with a frequency up to 100kHz.	4	High, dependent on module
Counters	3 up to 100kHz	6 up to 400kHz	2048
Work memory; Expandable	50kbyte; No	17kbyte; Yes 1 memory card for load memory (32Gbyte).	512kbyte; Yes 64mbyte
Max number of modules	3 communication and 1 signal board.	CPU + 31 modules	21
CPU blocks	DBs, FCs, FBs, counters, and timers.	OB, FB, FC, DB and UDTs	OB, FB, FC, DB
Switching frequency	100kHz	100 kHz; For high-speed output, 100 Hz for standard output	Dependent on module.
Rated supply voltage	24 V DC	24 V DC	120/230 V AC or 24 V DC.
Programming languages	LAD, FBD, SCL	LAD, FBD, STL, SCL, GRAPH	LAD, FBD, STL, SCL, GRAPH, Higraph
Expansion availability	Low	Medium	High
CPU processing times bit operation; word operation; floating point arithmetic; fixed point arithmetic	0.08 µs; / instruction; 1.7 µs; / instruction ;2.3 µs; /; N/A instruction	60ns; 72ns; 384 ns; 96 ns.	21.25 ns; 31.25 ns; 62.5 ns; 31.25 ns.
Number of maximum connections overall	8	96	48

Table 1; PLC Comparison of Unitary, Modular and Rack Mounted

One citation would be enough since these are identical, they are like 4 buses arriving at the same time!
(Siemens, 2021) (Siemens, 2021) (Siemens, 2021) (Siemens, 2021)

It is important to investigate and consider the specifications needs while choosing a PLC. This is because cost is a big factor in a design process therefore, choosing a rack mounted PLC for an application with a small I/O requirement would be a waste of finances. However, it would be irresponsible to choose a PLC which is not capable of carrying out all the requirements needed.

The PLC selected for the robot control panel is a modular Siemens S7-1214C. This is a suitable choice for the application for various reasons. A table below shows a side-by-side comparison of why it is suitable.

Excellent table to help choose the best PLC!

Specification	Pick and Place Robot	Siemens S7-1214C
Digital Input	14	14
Digital Output	8	10
Programming Language	SCL, LAD	SCL, LAD and FBD
Number of Counters	1	6
Interface	RJ45 Ethernet	RJ45 Ethernet
Maintenance Free	Required	Specified
Number of position-controlled positioning axes, max	6	8
Test Commissioning Functions	Inputs/outputs, memory bits, DBs, distributed I/Os, timers, counters	Inputs/outputs, memory bits, DBs, distributed I/Os, timers, counters
Work memory	56kbytes Estimated	100mbytes
Load memory	2.2kbytes Estimated	100kbytes

Table 2; Pick and place robot requirements in comparison to chosen PLC; Siemens S7-1214C

(Siemens, 2021)

All the required specifications will be met by this PLC, with addition for room for expansion.

2. Hardware Investigation, Circuit Diagram and I/O List

To be able to construct a pick and place robot, it requires components to interact with each other to operate accurately. The types of hardware consist of inputs and outputs, meaning where an input is given an output will be carried out.

Reed switch sensors are located on each axes of the robot this allows it to detect which position the of the piston. The reed switch sensor is an electromagnetic switch. When an electromagnetic field is nearby, the magnets repel which breaks the circuit (FIRST4MAGNETS ; A DIVISION OF MAGNET EXPERTS, 2014). This is used by detected the position of the robot, because the piston comes close to the reed switch which creates the magnets to repel, thus breaking the circuit and a signal is sent to the PLC. An example is when the piston goes to the UP position, it reaches the reed switch which opens and sends a signal to the PLC to say it has reached the up position.

A proximity switch is used on the entry point at station 1. The assumed proximity switch used for this application is a capacitive proximity sensor because of the object material is resin. The switch is also a sensor, when an object is placed in front of it, the amount of capacitance changes which then sends a signal to the PLC (OMRON Industrial Automation, n.d.). This is used in the robot, when an object is inserted at station 1 the proximity switch sensor will send a signal to the PLC, which then starts the automatic sequence of the program.

Pneumatic pistons which are controlled by solenoid valves are used to control movement. Compressed air is stored in the pneumatic manifold at approximately 4 bar. The solenoid valve connected to the manifold will open when power is sent to the coil which allows the compressed air to pass. The piston will move due to the compressed air, allowing it to control the axes of the robot, in addition to opening and closing the gripper.

Three pushbuttons and two rocker switches are used. These are normally open contacts, once they are pushed, they are closed which allows the power to flow and sends a signal to the PLC. Whether it is manual controls for the solenoid or starting the automatic sequence.

A KTP700 7 inch landscape HMI is used for the manual controls and display information about the program such as the step numbers, object counter, and pilot lights. This is an interactive device which is suitable for all types of users such as operators or engineers.

Lastly, the PLC, which is a Siemens 1214C, PLC stands for programmable logic controller, this industrial computer like device is sent input from input devices such as sensors and then compares those inputs with program logic data. The PLC is then able to generate outputs to control peripheral output devices such as the ones mentioned above (Kamel & Kamel, 2013).

Table 3 details an input list and table 4 details an output list.

Addresses of inputs should be given a 'I' not 'M' in this table

Inputs	Data type	Address	Signal
UP sensor	Bool	%M0.0	Digital
DOWN sensor	Bool	%M0.1	Digital
OPEN sensor	Bool	%M0.2	Digital
CLOSED sensor	Bool	%M0.3	Digital
IN sensor	Bool	%M0.4	Digital
OUT sensor	Bool	%M0.5	Digital
RIGHT sensor	Bool	%M0.6	Digital
LEFT sensor	Bool	%M0.7	Digital
OBJECT sensor	Bool	%M1.5	Digital
X10 Pushbutton	Bool	%M1.0	Digital
X11 Pushbutton	Bool	%M1.1	Digital
X12 Pushbutton	Bool	%M1.2	Digital
X13 toggle switch	Bool	%M1.3	Digital
X14 toggle switch	Bool	%M1.4	Digital

Table 3; Input list

Outputs	Data type	Address	Signal
move DOWN solenoid	Bool	%Q0.0	Digital
move UP solenoid	Bool	%Q0.1	Digital
CLOSE GRIPPER solenoid	Bool	%Q0.2	Digital
OPEN GRIPPER solenoid	Bool	%Q0.3	Digital
move OUT solenoid	Bool	%Q0.4	Digital
move IN solenoid	Bool	%Q0.5	Digital
move LEFT solenoid	Bool	%Q0.6	Digital
move RIGHT solenoid	Bool	%Q0.7	Digital

Table 4; Output list

Figure 3 which is located at the end of the appendix, shows the wiring diagram.

3. Program Design and Explanation

All program design such as SCL code, main block and PLC tag list can be found in the appendix with supporting comments.

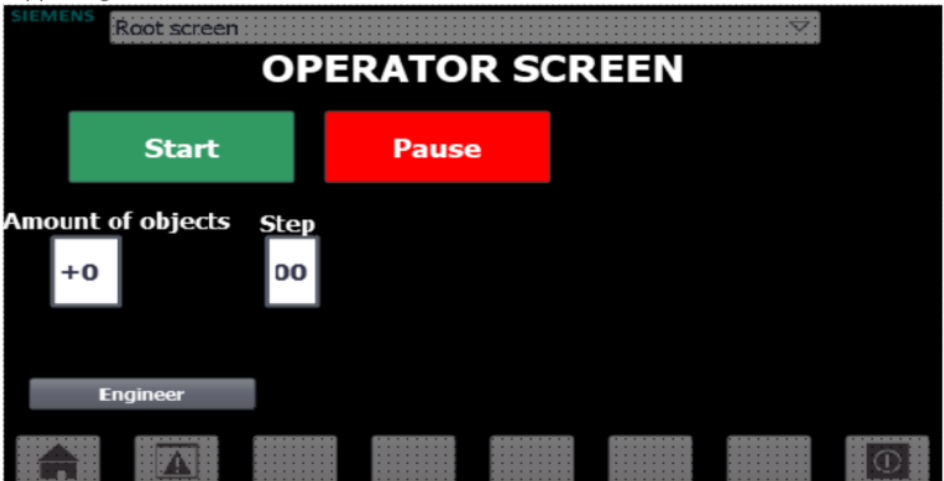


Figure 1; Operator HMI screen

	Pressed	Released
Start	▼ SetBit Tag (Input/output)Start pb ▼ ResetBit Tag (Input/output)Pause pb	▼ ResetBit Tag (Input/output)Start pb
Pause	▼ SetBit Tag (Input/output)Pause pb	▼ SetBit Tag (Input/output)Pause pb

Table 5; Operator HMI; Start and pause

Object counter	General Process Tag: Object counter_CV PLC tag: "Object counter".CV Address: Int Type Mode: Input/output Format Display format: Decimal Decimal places: 0 Field length: 1 Leading zeros: Format pattern: s9
Step number	Process Tag: Step PLC tag: Step Address: UInt Type Mode: Input/output Format Display format: Decimal Decimal places: 0 Field length: 2 Leading zeros: Format pattern: 99

Table 6;

Operator HMI; Object counter and step

The engineer screen can be accessed via the operator screen on the bottom left.

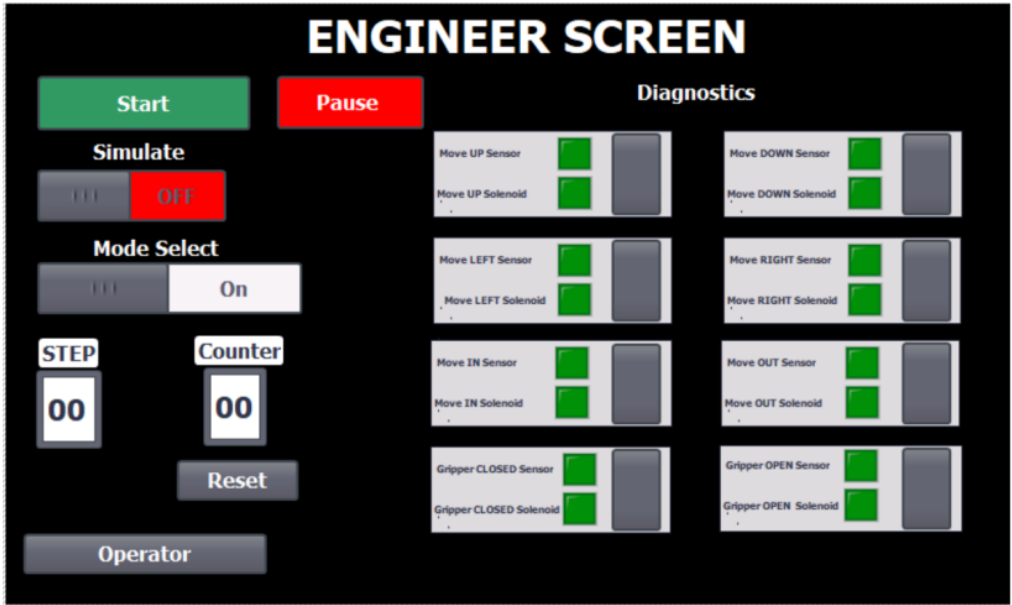


Figure 2; Engineer HMI screen

The engineer screen consists of the same functions as the operator screen, however, includes more options such as simulating inputs, selecting automatic or manual, manual controls for all solenoids and a reset counter. There is a diagnostic panel which shows what is happening in the program or for manual testing by displaying pilot lights for when a solenoid is active and when the sensor is on.

	Pressed	Released
Start	▼ SetBit Tag (Input/output)Start pb ▼ ResetBit Tag (Input/output)Pause pb	▼ ResetBit Tag (Input/output)Start pb
Pause	▼ SetBit Tag (Input/output)Pause pb	▼ SetBit Tag (Input/output)Pause pb
Reset	▼ SetBit Tag (Input/output)reset object counter	▼ ResetBit Tag (Input/output)reset object counter

Manual Up Button	▼ SetBit	
	Tag (Input/output)	Up HMI request
Manual Down Button	▼ SetBit	
	Tag (Input/output)	Down HMI request

Table 7 ; Engineer HMI ; Buttons

The rest of the manual buttons for solenoids are set up the same way, such as

Press – Set bit, Tag (... HMI request)

Release – Set bit, Tag (... HMI request)

The manual control of the solenoids may be called; however, they are in the main interlocks, meaning they must follow the same principles as the auto sequence.

Object counter	General Process Tag: Object counter_CV PLC tag: *Object counter*.CV Address: Int Type Mode: Input/output Format Display format: Decimal Decimal places: 0 Field length: 1 Leading zeros: Format pattern: s9
Step number	Process Tag: Step PLC tag: Step Address: UInt Type Mode: Input/output Format Display format: Decimal Decimal places: 0 Field length: 2 Leading zeros: Format pattern: 99

Table 8; Engineer HMI; Counter and step

Simulate	General Process Tag: Simulate PLC tag: Simulate Address: Bool Value for "ON": 1 Mode Format: Switch Label ON: ON OFF: OFF
Mode Select	Process Tag: AUTO PLC tag: AUTO Address: Bool Value for "ON": 1 Mode Format: Switch Label ON: Off OFF: On

Table 9 ; Engineer HMI; Switches

Up sensor pilot light	Process Tag: UP sensor PLC tag: "UP sensor" Address: Bool Bit number: 0 Mode Mode: Two states	Contents Value for "ON": 1 On: PilotLight_Square_G_On_2 Off: PilotLight_Square_G_Off_2
Move up sensor pilot light	Process Tag: move UP solenoid PLC tag: "move UP solenoid" Address: Bool Bit number: 0 Mode Mode: Two states	Contents Value for "ON": 1 On: PilotLight_Square_G_On_2 Off: PilotLight_Square_G_Off_2
Down sensor pilot light	Process Tag: DOWN sensor PLC tag: "DOWN sensor" Address: Bool Bit number: 0 Mode Mode: Two states	Contents Value for "ON": 1 On: PilotLight_Square_G_On_2 Off: PilotLight_Square_G_Off_2
Move down solenoid pilot light	Process Tag: move DOWN solenoid PLC tag: "move DOWN solenoid" Address: Bool Bit number: 0 Mode Mode: Two states	Contents Value for "ON": 1 On: PilotLight_Square_G_On_2 Off: PilotLight_Square_G_Off_2

Table 10; Engineer HMI ; Pilot lights

The rest of the pilot lights are set up the same way, such as

For sensor lights, Tag - sensor

For solenoids, Tag – Move solenoid

The pilot lights assist engineers in recording the reliability of the program by seeing what function is carried out.

Object counter

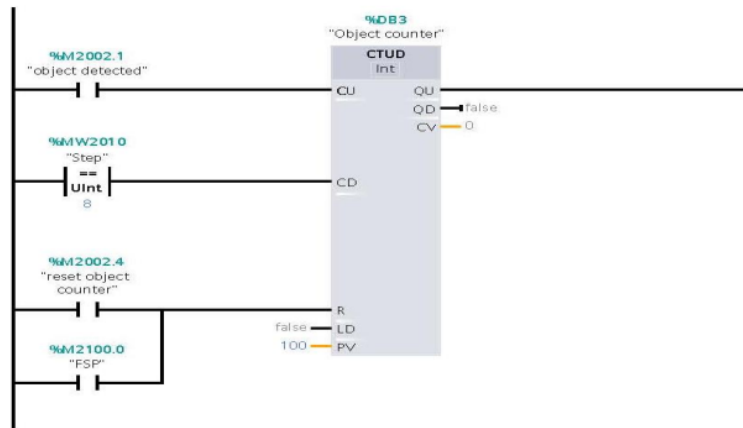


Figure 3; Object counter

This object counter will be displayed on both HMI screens to show how many objects are in the robot. Once object detected sensor is activated it will increase to 1 object in the machine. It will count down 1 on step 8 because this is when the object is released on the exit chute. The counter will reset if the first scan pulse is started or the reset object counter is pressed.

Object timer

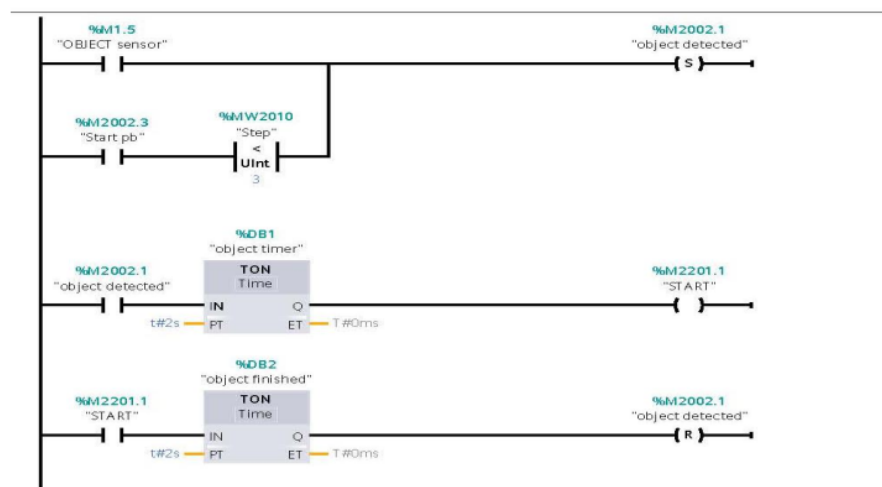


Figure 4; Object timer

DB1% shows if there is an object rolled into station 1 or the start pb button pressed it will wait 2 seconds before starting the automatic sequence. DB2% is used for object finished, if start is pressed it will wait two seconds before resetting the object detected.

Safety interlocks

Safety interlocks have been implemented into the main block, this is shown in the appendix and detailed below. Interlocks stop the machine from causing damage to itself in the automatic sequence and the manual mode. Below will only detail the interlocks, not including the selected mode and calls made by the HMI or automatic sequence.

Network 2, the move up solenoid will only be activated if the up sensor is off. This stops the machine from causing damage if it is already in the up position. It should also be noted in the move up rung, if any sensors are not on then the up solenoid will be activated. This prevents any glitches in the program such as moving left, right, in or out while in the down position.

Network 3, the move down solenoid will only be activated if the left or right and in or out sensors are on. This stops it moving down between stations.

Network 4, the move left solenoid will only be activated if the up sensor is on, and the left sensor is off. Meaning the up sensor must be on because it should not travel when down and the left sensor off otherwise it is already left and cannot go any further.

Network 5, the move right solenoid will activate if the up sensor is on, because it cannot travel when down, and the right sensor is off because if its already right.

Network 6, the move in solenoid will activate if the up sensor is on, because it cannot travel when down and the in sensor is off. If the in sensor is on, it is already in and cannot go any further.

Network 7, the move out solenoid will activate if the up sensor is on and the out sensor is off. This is because the robot should not travel if down, and the out sensor should be off otherwise it would already be in the out position.

Network 8, which is a pause button for the sequence is a safety feature. Once the pause button is pressed it will stop the sequence.

Network 9, the close gripper solenoid will be activated if the closed sensor is off, otherwise it would already be closed.

Network 10, the open gripper solenoid will be activated if the open sensor is off, otherwise it would already be open.

Program explanation

The operation of the program is dependent on the number of objects which are in the robot. If there is one object in the robot it will go from step 1 straight to step 18 and then follow each step till 26. After step 26 it will return to step 0.

If there are 2 objects in the machine, it will start from step 1 to step 10 and then follow each step till 26. After step 26 it will return to 0.

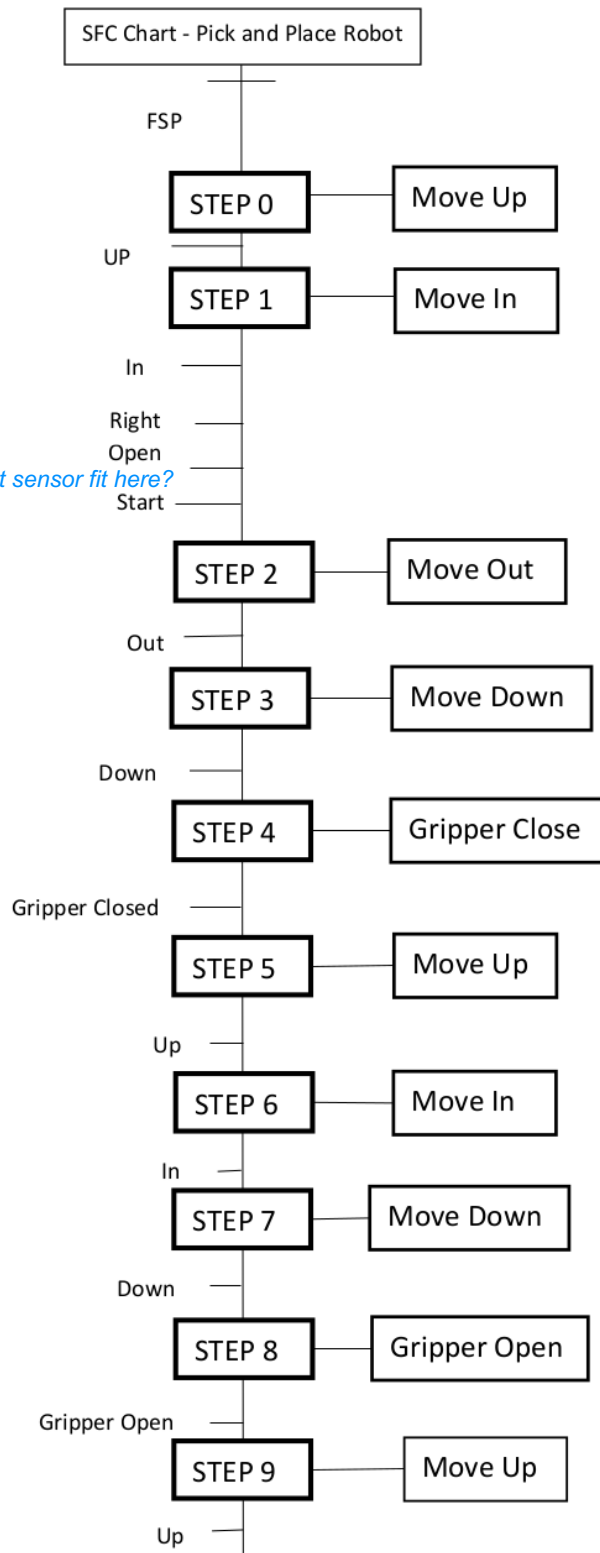
If there are 3 objects in the machine it will start at step 1 and follow till 26. After step 10 an object will be deducted and after step 26 it will return to step 0.

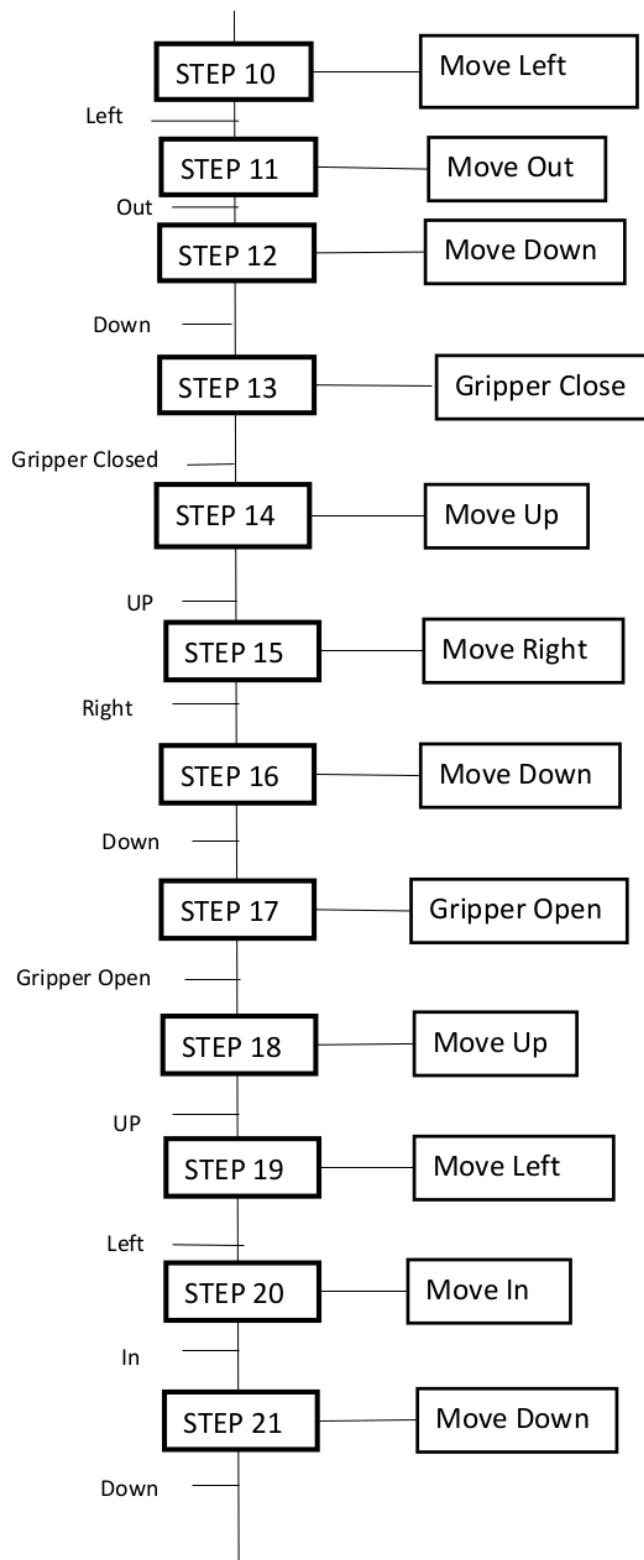
Below shows an SFC chart of the steps and what sensors and movements activate dependent on the step.

A very good SFC chart

NOTE: When automatic sequence is selected. This is the program steps.

where does the object sensor fit here?





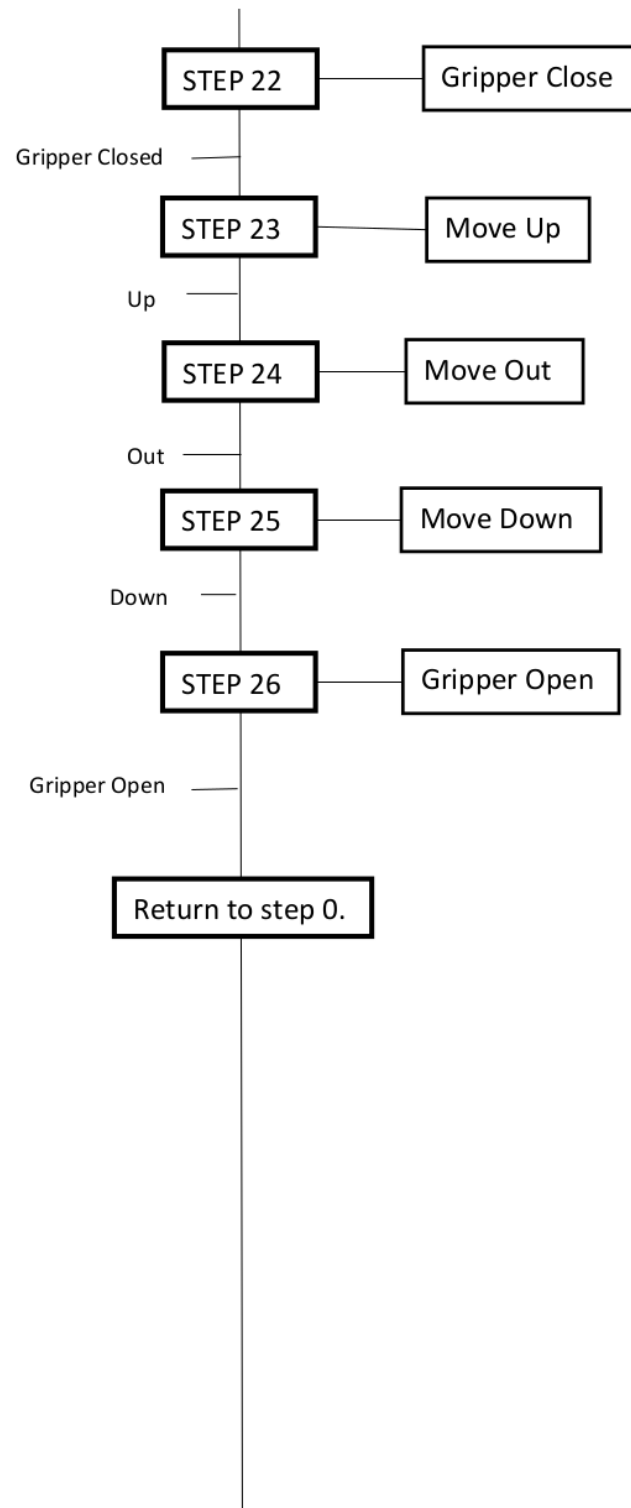


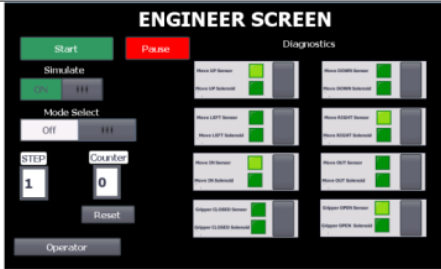
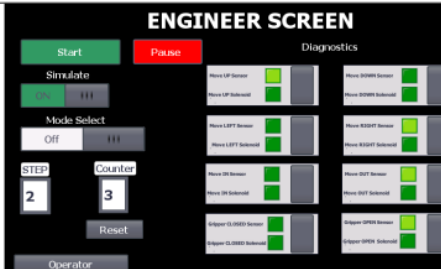
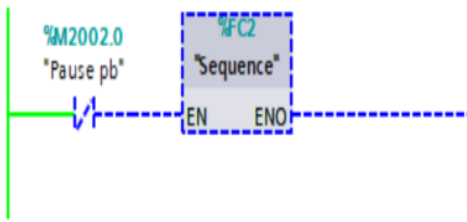
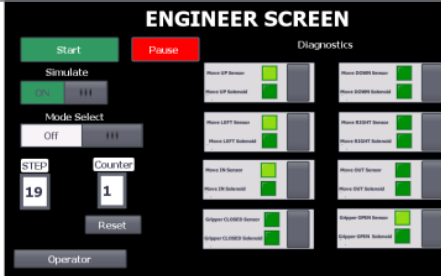
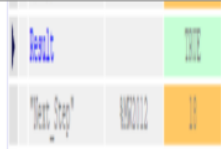
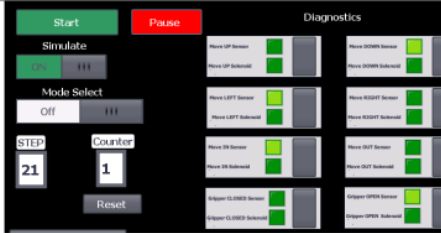
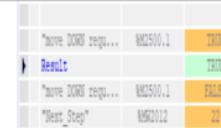
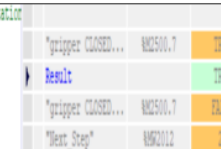
Figure 5; SFC Chart

I like what you have done with this table but it could be better if the evidence was in the appendix and actual results described.

Testing Table could be

used for

4. Simulation Testing

Requirement	HMI Image	Program Image	Result
On power up the robot will move safely to the home position above station 4 and wait.		<pre> Result "move UP request" %M2500.0 TRUE "Next_Step" %MW2012 1 "move IN request" %M2500.4 TRUE "move RIGHT req..." %M2500.3 TRUE "gripper OPEN r..." %M2500.6 TRUE Result "RIGHT sensor" %M0.6 TRUE "OPEN sensor" %M0.2 TRUE "START" %M2201.1 FALSE </pre>	Pass
Pause button to stop sequence. (pressed pause at step 2)			Pass
Sequence for 1 object. -Goes to step 18 up, but already up so then moves left step 19		<pre> IF "Object counter" <= 1 THEN "Next_Step" := 19; </pre> 	Pass
-Goes to step 20 in, but already in so goes to step 21.		<pre> 21: // Move down: lowers onto station 1. "move DOWN request" := 1; IF "DOWN sensor" = 1 THEN "move DOWN request" := 0; "Next_Step" := 22; </pre> 	Pass
Step 22		<pre> 22: // Gripper close: grabs object on station "gripper CLOSED request" := 1; IF "CLOSED sensor" = 1 THEN "gripper CLOSED request" := 0; "Next_Step" := 23; </pre> 	Pass

Requirement	HMI Image	Program Image	Result												
Step 23		<pre>23: // Move up; moves above station 1. "move UP request" := 1; IF "UP sensor" = 1 THEN "move UP request" := 0; "Next_Step" := 24;</pre> <table><tr><td>"move UP request"</td><td>MMS500.0</td><td>TRUE</td></tr><tr><td>Result</td><td></td><td>TRUE</td></tr><tr><td>"move UP request"</td><td>MMS500.0</td><td>FALSE</td></tr><tr><td>"Next_Step"</td><td>MMS2012</td><td>24</td></tr></table>	"move UP request"	MMS500.0	TRUE	Result		TRUE	"move UP request"	MMS500.0	FALSE	"Next_Step"	MMS2012	24	Pass
"move UP request"	MMS500.0	TRUE													
Result		TRUE													
"move UP request"	MMS500.0	FALSE													
"Next_Step"	MMS2012	24													
Step 24		<pre>24: // Move out; moves to station 2. "move OUT request" := 1; IF "OUT sensor" = 1 THEN "move OUT request" := 0; "Next_Step" := 25;</pre> <table><tr><td>"move OUT request"</td><td>MMS500.5</td><td>TRUE</td></tr><tr><td>Result</td><td></td><td>TRUE</td></tr><tr><td>"move OUT request"</td><td>MMS500.5</td><td>FALSE</td></tr><tr><td>"Next_Step"</td><td>MMS2012</td><td>25</td></tr></table>	"move OUT request"	MMS500.5	TRUE	Result		TRUE	"move OUT request"	MMS500.5	FALSE	"Next_Step"	MMS2012	25	Pass
"move OUT request"	MMS500.5	TRUE													
Result		TRUE													
"move OUT request"	MMS500.5	FALSE													
"Next_Step"	MMS2012	25													
Step 25		<pre>25: // Move down; moves down to station 2 "move DOWN request" := 1; IF "DOWN sensor" = 1 THEN "move DOWN request" := 0; "Next_Step" := 26; END IF;</pre> <table><tr><td>"move DOWN request"</td><td>MMS500.1</td><td>TRUE</td></tr><tr><td>Result</td><td></td><td>TRUE</td></tr><tr><td>"move DOWN request"</td><td>MMS500.1</td><td>FALSE</td></tr><tr><td>"Next_Step"</td><td>MMS2012</td><td>26</td></tr></table>	"move DOWN request"	MMS500.1	TRUE	Result		TRUE	"move DOWN request"	MMS500.1	FALSE	"Next_Step"	MMS2012	26	Pass
"move DOWN request"	MMS500.1	TRUE													
Result		TRUE													
"move DOWN request"	MMS500.1	FALSE													
"Next_Step"	MMS2012	26													
Step 26		<pre>26: // Gripper "grripper OPEN request" := 1; IF "OPEN sensor" = 1 THEN "grripper OPEN request" := 0; "Next_Step" := 0; // back to start</pre> <table><tr><td>"grripper OPEN request"</td><td>MMS500.6</td><td>TRUE</td></tr><tr><td>Result</td><td></td><td>TRUE</td></tr><tr><td>"grripper OPEN request"</td><td>MMS500.6</td><td>FALSE</td></tr><tr><td>"Next_Step"</td><td>MMS2012</td><td>0</td></tr></table> Returns to step 0.	"grripper OPEN request"	MMS500.6	TRUE	Result		TRUE	"grripper OPEN request"	MMS500.6	FALSE	"Next_Step"	MMS2012	0	Pass
"grripper OPEN request"	MMS500.6	TRUE													
Result		TRUE													
"grripper OPEN request"	MMS500.6	FALSE													
"Next_Step"	MMS2012	0													
2 objects in on robot stations. Goes from step 1 to step 10.		<pre>10: // 1 "move LEFT request" := 1; IF "LEFT sensor" = 1 THEN "move LEFT request" := 0; "Next_Step" := 11;</pre> <table><tr><td>"move LEFT request"</td><td>MMS500.2</td><td>TRUE</td></tr><tr><td>Result</td><td></td><td>TRUE</td></tr><tr><td>"move LEFT request"</td><td>MMS500.2</td><td>FALSE</td></tr><tr><td>"Next_Step"</td><td>MMS2012</td><td>11</td></tr></table>	"move LEFT request"	MMS500.2	TRUE	Result		TRUE	"move LEFT request"	MMS500.2	FALSE	"Next_Step"	MMS2012	11	Pass
"move LEFT request"	MMS500.2	TRUE													
Result		TRUE													
"move LEFT request"	MMS500.2	FALSE													
"Next_Step"	MMS2012	11													

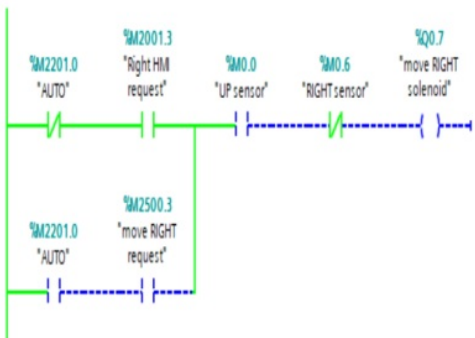
Step 11		<pre> 11: // M "move OUT request" := 1; IF "OUT sensor" = 1 THEN "move OUT request" := 0; "Next_Step" := 12; </pre> <table border="1"> <tr> <td>"move OUT request" := 1;</td> <td>"move OUT request" := 1;</td> <td>4M2500.5</td> <td>TRUE</td> </tr> <tr> <td>IF "OUT sensor" = 1 THEN</td> <td>Result</td> <td></td> <td>TRUE</td> </tr> <tr> <td>"move OUT request" := 0;</td> <td>"move OUT request" := 0;</td> <td>4M2500.5</td> <td>FALSE</td> </tr> <tr> <td>"Next_Step" := 12;</td> <td>"Next_Step" := 12;</td> <td>4M2012</td> <td>12</td> </tr> </table>	"move OUT request" := 1;	"move OUT request" := 1;	4M2500.5	TRUE	IF "OUT sensor" = 1 THEN	Result		TRUE	"move OUT request" := 0;	"move OUT request" := 0;	4M2500.5	FALSE	"Next_Step" := 12;	"Next_Step" := 12;	4M2012	12	Pass
"move OUT request" := 1;	"move OUT request" := 1;	4M2500.5	TRUE																
IF "OUT sensor" = 1 THEN	Result		TRUE																
"move OUT request" := 0;	"move OUT request" := 0;	4M2500.5	FALSE																
"Next_Step" := 12;	"Next_Step" := 12;	4M2012	12																
Step 12		<pre> 12: // Mo "move DOWN request" := 1; IF "DOWN sensor" = 1 THEN "move DOWN request" := 0; "Next_Step" := 13; </pre> <table border="1"> <tr> <td>"move DOWN request" := 1;</td> <td>"move DOWN request" := 1;</td> <td>4M2500.1</td> <td>TRUE</td> </tr> <tr> <td>IF "DOWN sensor" = 1 THEN</td> <td>Result</td> <td></td> <td>TRUE</td> </tr> <tr> <td>"move DOWN request" := 0;</td> <td>"move DOWN request" := 0;</td> <td>4M2500.1</td> <td>FALSE</td> </tr> <tr> <td>"Next_Step" := 13;</td> <td>"Next_Step" := 13;</td> <td>4M2012</td> <td>13</td> </tr> </table>	"move DOWN request" := 1;	"move DOWN request" := 1;	4M2500.1	TRUE	IF "DOWN sensor" = 1 THEN	Result		TRUE	"move DOWN request" := 0;	"move DOWN request" := 0;	4M2500.1	FALSE	"Next_Step" := 13;	"Next_Step" := 13;	4M2012	13	Pass
"move DOWN request" := 1;	"move DOWN request" := 1;	4M2500.1	TRUE																
IF "DOWN sensor" = 1 THEN	Result		TRUE																
"move DOWN request" := 0;	"move DOWN request" := 0;	4M2500.1	FALSE																
"Next_Step" := 13;	"Next_Step" := 13;	4M2012	13																
Step 13		<pre> 13: // Gri "grripper CLOSED request" := 1; IF "CLOSED sensor" = 1 THEN "grripper CLOSED request" := 0; "Next_Step" := 14; </pre> <table border="1"> <tr> <td>"grripper CLOSED request" := 1;</td> <td>"grripper CLOSED request" := 1;</td> <td>4M2500.7</td> <td>TRUE</td> </tr> <tr> <td>IF "CLOSED sensor" = 1 THEN</td> <td>Result</td> <td></td> <td>TRUE</td> </tr> <tr> <td>"grripper CLOSED request" := 0;</td> <td>"grripper CLOSED request" := 0;</td> <td>4M2500.7</td> <td>FALSE</td> </tr> <tr> <td>"Next_Step" := 14;</td> <td>"Next_Step" := 14;</td> <td>4M2012</td> <td></td> </tr> </table>	"grripper CLOSED request" := 1;	"grripper CLOSED request" := 1;	4M2500.7	TRUE	IF "CLOSED sensor" = 1 THEN	Result		TRUE	"grripper CLOSED request" := 0;	"grripper CLOSED request" := 0;	4M2500.7	FALSE	"Next_Step" := 14;	"Next_Step" := 14;	4M2012		Pass
"grripper CLOSED request" := 1;	"grripper CLOSED request" := 1;	4M2500.7	TRUE																
IF "CLOSED sensor" = 1 THEN	Result		TRUE																
"grripper CLOSED request" := 0;	"grripper CLOSED request" := 0;	4M2500.7	FALSE																
"Next_Step" := 14;	"Next_Step" := 14;	4M2012																	
Step 14		<pre> 14: // Mov "move UP request" := 1; IF "UP sensor" = 1 THEN "move UP request" := 0; "Next_Step" := 15; </pre> <table border="1"> <tr> <td>"move UP request" := 1;</td> <td>"move UP request" := 1;</td> <td>4M2500.0</td> <td>TRUE</td> </tr> <tr> <td>IF "UP sensor" = 1 THEN</td> <td>Result</td> <td></td> <td>TRUE</td> </tr> <tr> <td>"move UP request" := 0;</td> <td>"move UP request" := 0;</td> <td>4M2500.0</td> <td>FALSE</td> </tr> <tr> <td>"Next_Step" := 15;</td> <td>"Next_Step" := 15;</td> <td>4M2012</td> <td></td> </tr> </table>	"move UP request" := 1;	"move UP request" := 1;	4M2500.0	TRUE	IF "UP sensor" = 1 THEN	Result		TRUE	"move UP request" := 0;	"move UP request" := 0;	4M2500.0	FALSE	"Next_Step" := 15;	"Next_Step" := 15;	4M2012		Pass
"move UP request" := 1;	"move UP request" := 1;	4M2500.0	TRUE																
IF "UP sensor" = 1 THEN	Result		TRUE																
"move UP request" := 0;	"move UP request" := 0;	4M2500.0	FALSE																
"Next_Step" := 15;	"Next_Step" := 15;	4M2012																	
Step 15		<pre> 15: // Move "move RIGHT request" := 1; IF "RIGHT sensor" = 1 THEN "move RIGHT request" := 0; "Next_Step" := 16; </pre> <table border="1"> <tr> <td>"move RIGHT request" := 1;</td> <td>"move RIGHT request" := 1;</td> <td>4M2500.3</td> <td>TRUE</td> </tr> <tr> <td>IF "RIGHT sensor" = 1 THEN</td> <td>Result</td> <td></td> <td>TRUE</td> </tr> <tr> <td>"move RIGHT request" := 0;</td> <td>"move RIGHT request" := 0;</td> <td>4M2500.3</td> <td>FALSE</td> </tr> <tr> <td>"Next_Step" := 16;</td> <td>"Next_Step" := 16;</td> <td>4M2012</td> <td></td> </tr> </table>	"move RIGHT request" := 1;	"move RIGHT request" := 1;	4M2500.3	TRUE	IF "RIGHT sensor" = 1 THEN	Result		TRUE	"move RIGHT request" := 0;	"move RIGHT request" := 0;	4M2500.3	FALSE	"Next_Step" := 16;	"Next_Step" := 16;	4M2012		Pass
"move RIGHT request" := 1;	"move RIGHT request" := 1;	4M2500.3	TRUE																
IF "RIGHT sensor" = 1 THEN	Result		TRUE																
"move RIGHT request" := 0;	"move RIGHT request" := 0;	4M2500.3	FALSE																
"Next_Step" := 16;	"Next_Step" := 16;	4M2012																	
Step 16		<pre> 16: // Move "move DOWN request" := 1; IF "DOWN sensor" = 1 THEN "move DOWN request" := 0; "Next_Step" := 17; </pre> <table border="1"> <tr> <td>"move DOWN request" := 1;</td> <td>"move DOWN request" := 1;</td> <td>4M2500.1</td> <td>TRUE</td> </tr> <tr> <td>IF "DOWN sensor" = 1 THEN</td> <td>Result</td> <td></td> <td>TRUE</td> </tr> <tr> <td>"move DOWN request" := 0;</td> <td>"move DOWN request" := 0;</td> <td>4M2500.1</td> <td>FALSE</td> </tr> <tr> <td>"Next_Step" := 17;</td> <td>"Next_Step" := 17;</td> <td>4M2012</td> <td>17</td> </tr> </table>	"move DOWN request" := 1;	"move DOWN request" := 1;	4M2500.1	TRUE	IF "DOWN sensor" = 1 THEN	Result		TRUE	"move DOWN request" := 0;	"move DOWN request" := 0;	4M2500.1	FALSE	"Next_Step" := 17;	"Next_Step" := 17;	4M2012	17	Pass
"move DOWN request" := 1;	"move DOWN request" := 1;	4M2500.1	TRUE																
IF "DOWN sensor" = 1 THEN	Result		TRUE																
"move DOWN request" := 0;	"move DOWN request" := 0;	4M2500.1	FALSE																
"Next_Step" := 17;	"Next_Step" := 17;	4M2012	17																

Step 17		<pre> 17: // Gripper "grripper OPEN request" := 1; IF "OPEN sensor" = 1 THEN "grripper OPEN request" := 0; "Next_Step" := 18; </pre> <table border="1"> <tr> <td>"grripper OPEN request" := 1;</td><td>"grripper OPEN ...</td><td>4M2500.6</td><td>TRUE</td></tr> <tr> <td>IF "OPEN sensor" = 1 THEN</td><td>Result</td><td></td><td>TRUE</td></tr> <tr> <td>"grripper OPEN request" := 0;</td><td>"grripper OPEN ...</td><td>4M2500.6</td><td>FALSE</td></tr> <tr> <td>"Next_Step" := 18;</td><td>"Next_Step"</td><td>4MM2012</td><td>18</td></tr> </table>	"grripper OPEN request" := 1;	"grripper OPEN ...	4M2500.6	TRUE	IF "OPEN sensor" = 1 THEN	Result		TRUE	"grripper OPEN request" := 0;	"grripper OPEN ...	4M2500.6	FALSE	"Next_Step" := 18;	"Next_Step"	4MM2012	18	Pass
"grripper OPEN request" := 1;	"grripper OPEN ...	4M2500.6	TRUE																
IF "OPEN sensor" = 1 THEN	Result		TRUE																
"grripper OPEN request" := 0;	"grripper OPEN ...	4M2500.6	FALSE																
"Next_Step" := 18;	"Next_Step"	4MM2012	18																
Step 18		<pre> 18: // Move "move UP request" := 1; IF "UP sensor" = 1 THEN "move UP request" := 0; "Next_Step" := 19; </pre> <table border="1"> <tr> <td>"move UP request" := 1;</td><td>"move UP request"</td><td>4M2500.0</td><td>TRUE</td></tr> <tr> <td>IF "UP sensor" = 1 THEN</td><td>Result</td><td></td><td>TRUE</td></tr> <tr> <td>"move UP request" := 0;</td><td>"move UP request"</td><td>4M2500.0</td><td>FALSE</td></tr> <tr> <td>"Next_Step" := 19;</td><td>"Next_Step"</td><td>4MM2012</td><td>19</td></tr> </table>	"move UP request" := 1;	"move UP request"	4M2500.0	TRUE	IF "UP sensor" = 1 THEN	Result		TRUE	"move UP request" := 0;	"move UP request"	4M2500.0	FALSE	"Next_Step" := 19;	"Next_Step"	4MM2012	19	Pass
"move UP request" := 1;	"move UP request"	4M2500.0	TRUE																
IF "UP sensor" = 1 THEN	Result		TRUE																
"move UP request" := 0;	"move UP request"	4M2500.0	FALSE																
"Next_Step" := 19;	"Next_Step"	4MM2012	19																
Step 19		<pre> 19: // Move "move LEFT request" := 1; IF "LEFT sensor" = 1 THEN "move LEFT request" := 0; "Next_Step" := 20; </pre> <table border="1"> <tr> <td>"move LEFT request" := 1;</td><td>"move LEFT req..."</td><td>4M2500.2</td><td>TRUE</td></tr> <tr> <td>IF "LEFT sensor" = 1 THEN</td><td>Result</td><td></td><td>TRUE</td></tr> <tr> <td>"move LEFT request" := 0;</td><td>"move LEFT req..."</td><td>4M2500.2</td><td>FALSE</td></tr> <tr> <td>"Next_Step" := 20;</td><td>"Next_Step"</td><td>4MM2012</td><td>20</td></tr> </table>	"move LEFT request" := 1;	"move LEFT req..."	4M2500.2	TRUE	IF "LEFT sensor" = 1 THEN	Result		TRUE	"move LEFT request" := 0;	"move LEFT req..."	4M2500.2	FALSE	"Next_Step" := 20;	"Next_Step"	4MM2012	20	Pass
"move LEFT request" := 1;	"move LEFT req..."	4M2500.2	TRUE																
IF "LEFT sensor" = 1 THEN	Result		TRUE																
"move LEFT request" := 0;	"move LEFT req..."	4M2500.2	FALSE																
"Next_Step" := 20;	"Next_Step"	4MM2012	20																
Step 20		<pre> 20: // Move "move IN request" := 1; IF "IN sensor" = 1 THEN "move IN request" := 0; "Next_Step" := 21; </pre> <table border="1"> <tr> <td>"move IN request" := 1;</td><td>"move IN request"</td><td>4M2500.4</td><td>TRUE</td></tr> <tr> <td>IF "IN sensor" = 1 THEN</td><td>Result</td><td></td><td>TRUE</td></tr> <tr> <td>"move IN request" := 0;</td><td>"move IN request"</td><td>4M2500.4</td><td>FALSE</td></tr> <tr> <td>"Next_Step" := 21;</td><td>"Next_Step"</td><td>4MM2012</td><td>21</td></tr> </table>	"move IN request" := 1;	"move IN request"	4M2500.4	TRUE	IF "IN sensor" = 1 THEN	Result		TRUE	"move IN request" := 0;	"move IN request"	4M2500.4	FALSE	"Next_Step" := 21;	"Next_Step"	4MM2012	21	Pass
"move IN request" := 1;	"move IN request"	4M2500.4	TRUE																
IF "IN sensor" = 1 THEN	Result		TRUE																
"move IN request" := 0;	"move IN request"	4M2500.4	FALSE																
"Next_Step" := 21;	"Next_Step"	4MM2012	21																
Step 21		<pre> 21: // Move "move DOWN request" := 1; IF "DOWN sensor" = 1 THEN "move DOWN request" := 0; "Next_Step" := 22; </pre> <table border="1"> <tr> <td>"move DOWN request" := 1;</td><td>"move DOWN req..."</td><td>4M2500.1</td><td>TRUE</td></tr> <tr> <td>IF "DOWN sensor" = 1 THEN</td><td>Result</td><td></td><td>TRUE</td></tr> <tr> <td>"move DOWN request" := 0;</td><td>"move DOWN req..."</td><td>4M2500.1</td><td>FALSE</td></tr> <tr> <td>"Next_Step" := 22;</td><td>"Next_Step"</td><td>4MM2012</td><td>22</td></tr> </table>	"move DOWN request" := 1;	"move DOWN req..."	4M2500.1	TRUE	IF "DOWN sensor" = 1 THEN	Result		TRUE	"move DOWN request" := 0;	"move DOWN req..."	4M2500.1	FALSE	"Next_Step" := 22;	"Next_Step"	4MM2012	22	Pass
"move DOWN request" := 1;	"move DOWN req..."	4M2500.1	TRUE																
IF "DOWN sensor" = 1 THEN	Result		TRUE																
"move DOWN request" := 0;	"move DOWN req..."	4M2500.1	FALSE																
"Next_Step" := 22;	"Next_Step"	4MM2012	22																
Step 22		<pre> 22: // Gripper close: grabs object on station "grripper CLOSED request" := 1; IF "CLOSED sensor" = 1 THEN "grripper CLOSED request" := 0; "Next_Step" := 23; </pre> <table border="1"> <tr> <td>"grripper CLOSED request" := 1;</td><td>"grripper CLOSED..."</td><td>4M2500.7</td><td>TRUE</td></tr> <tr> <td>IF "CLOSED sensor" = 1 THEN</td><td>Result</td><td></td><td>TRUE</td></tr> <tr> <td>"grripper CLOSED request" := 0;</td><td>"grripper CLOSED..."</td><td>4M2500.7</td><td>FALSE</td></tr> <tr> <td>"Next_Step" := 23;</td><td>"Next_Step"</td><td>4MM2012</td><td>23</td></tr> </table>	"grripper CLOSED request" := 1;	"grripper CLOSED..."	4M2500.7	TRUE	IF "CLOSED sensor" = 1 THEN	Result		TRUE	"grripper CLOSED request" := 0;	"grripper CLOSED..."	4M2500.7	FALSE	"Next_Step" := 23;	"Next_Step"	4MM2012	23	Pass
"grripper CLOSED request" := 1;	"grripper CLOSED..."	4M2500.7	TRUE																
IF "CLOSED sensor" = 1 THEN	Result		TRUE																
"grripper CLOSED request" := 0;	"grripper CLOSED..."	4M2500.7	FALSE																
"Next_Step" := 23;	"Next_Step"	4MM2012	23																
Step 23		<pre> 23: // Move "move UP request" := 1; IF "UP sensor" = 1 THEN "move UP request" := 0; "Next_Step" := 24; </pre> <table border="1"> <tr> <td>"move UP request" := 1;</td><td>"move UP request"</td><td>4M2500.0</td><td>TRUE</td></tr> <tr> <td>IF "UP sensor" = 1 THEN</td><td>Result</td><td></td><td>TRUE</td></tr> <tr> <td>"move UP request" := 0;</td><td>"move UP request"</td><td>4M2500.0</td><td>FALSE</td></tr> <tr> <td>"Next_Step" := 24;</td><td>"Next_Step"</td><td>4MM2012</td><td>24</td></tr> </table>	"move UP request" := 1;	"move UP request"	4M2500.0	TRUE	IF "UP sensor" = 1 THEN	Result		TRUE	"move UP request" := 0;	"move UP request"	4M2500.0	FALSE	"Next_Step" := 24;	"Next_Step"	4MM2012	24	Pass
"move UP request" := 1;	"move UP request"	4M2500.0	TRUE																
IF "UP sensor" = 1 THEN	Result		TRUE																
"move UP request" := 0;	"move UP request"	4M2500.0	FALSE																
"Next_Step" := 24;	"Next_Step"	4MM2012	24																

Step 24	Simulate ON OFF Mode Select Off III STEP 24 Counter 2 Reset Move UP Sensor Move UP Subsensor Move DOWN Sensor Move DOWN Subsensor Move LEFT Sensor Move LEFT Subsensor Move IN Sensor Move IN Subsensor Gripper CLOSED Sensor Gripper CLOSED Subsensor Move DOWN Sensor Move DOWN Subsensor Move RIGHT Sensor Move RIGHT Subsensor Move OUT Sensor Move OUT Subsensor Gripper OPEN Sensor Gripper OPEN Subsensor 24: // Move "move OUT request" := 1; IF "OUT sensor" = 1 THEN "move OUT request" := 0; "Next_Step" := 25; "move OUT request" := 1; "move OUT request" := 1; 4M2500.5 TRUE Result TRUE "move OUT request" := 0; "move OUT request" := 0; 4M2500.5 FALSE "Next_Step" := 25; "Next_Step" 4MM2012 25	Pass
Step 25	Simulate ON OFF Mode Select Off III STEP 25 Counter 2 Reset Move UP Sensor Move UP Subsensor Move DOWN Sensor Move DOWN Subsensor Move LEFT Sensor Move LEFT Subsensor Move IN Sensor Move IN Subsensor Gripper CLOSED Sensor Gripper CLOSED Subsensor Move DOWN Sensor Move DOWN Subsensor Move RIGHT Sensor Move RIGHT Subsensor Move OUT Sensor Move OUT Subsensor Gripper OPEN Sensor Gripper OPEN Subsensor 25: // Move "move DOWN request" := 1; IF "DOWN sensor" = 1 THEN "move DOWN request" := 0; "Next_Step" := 26; "move DOWN request" := 1; "move DOWN request" := 1; 4M2500.1 TRUE Result TRUE "move DOWN request" := 0; "move DOWN request" := 0; 4M2500.1 FALSE "Next_Step" := 26; "Next_Step" 4MM2012 26	Pass
Step 26	Simulate ON OFF Mode Select Off III STEP 0 Counter 2 Reset Move UP Sensor Move UP Subsensor Move DOWN Sensor Move DOWN Subsensor Move LEFT Sensor Move LEFT Subsensor Move IN Sensor Move IN Subsensor Gripper CLOSED Sensor Gripper CLOSED Subsensor Move DOWN Sensor Move DOWN Subsensor Move RIGHT Sensor Move RIGHT Subsensor Move OUT Sensor Move OUT Subsensor Gripper OPEN Sensor Gripper OPEN Subsensor 26: // Gripper "grripper OPEN request" := 1; IF "OPEN sensor" = 1 THEN "grripper OPEN request" := 0; "Next_Step" := 0; // back to step 2 "grripper OPEN request" := 1; "grripper OPEN request" := 1; 4M2500.6 TRUE Result TRUE "grripper OPEN request" := 0; "grripper OPEN request" := 0; 4M2500.6 FALSE "Next_Step" := 0; "Next_Step" 4MM2012 0 Returns to step 0.	Pass
3 Objects in the robot. Goes to step 2.	Simulate ON OFF Mode Select Off III STEP 2 Counter 3 Reset Move UP Sensor Move UP Subsensor Move DOWN Sensor Move DOWN Subsensor Move LEFT Sensor Move LEFT Subsensor Move IN Sensor Move IN Subsensor Gripper CLOSED Sensor Gripper CLOSED Subsensor Move DOWN Sensor Move DOWN Subsensor Move RIGHT Sensor Move RIGHT Subsensor Move OUT Sensor Move OUT Subsensor Gripper OPEN Sensor Gripper OPEN Subsensor 2: "move OUT request" := 1; IF "OUT sensor" = 1 THEN "move OUT request" := 0; "Next_Step" := 3; "move OUT request" := 1; "move OUT request" := 1; 4M2500.5 TRUE Result TRUE "move OUT request" := 0; "move OUT request" := 0; 4M2500.5 FALSE "Next_Step" := 3; "Next_Step" 4MM2012 3	Pass
Step 3	Simulate ON OFF Mode Select Off III STEP 3 Counter 3 Reset Move UP Sensor Move UP Subsensor Move DOWN Sensor Move DOWN Subsensor Move LEFT Sensor Move LEFT Subsensor Move IN Sensor Move IN Subsensor Gripper CLOSED Sensor Gripper CLOSED Subsensor Move DOWN Sensor Move DOWN Subsensor Move RIGHT Sensor Move RIGHT Subsensor Move OUT Sensor Move OUT Subsensor Gripper OPEN Sensor Gripper OPEN Subsensor 3: "move DOWN request" := 1; IF "DOWN sensor" = 1 THEN "move DOWN request" := 0; "Next_Step" := 4; "move DOWN request" := 1; "move DOWN request" := 1; 4M2500.1 TRUE Result TRUE "move DOWN request" := 0; "move DOWN request" := 0; 4M2500.1 FALSE "Next_Step" := 4; "Next_Step" 4MM2012 4	Pass
Step 4	Simulate ON OFF Mode Select Off III STEP 4 Counter 3 Reset Move UP Sensor Move UP Subsensor Move DOWN Sensor Move DOWN Subsensor Move LEFT Sensor Move LEFT Subsensor Move IN Sensor Move IN Subsensor Gripper CLOSED Sensor Gripper CLOSED Subsensor Move DOWN Sensor Move DOWN Subsensor Move RIGHT Sensor Move RIGHT Subsensor Move OUT Sensor Move OUT Subsensor Gripper OPEN Sensor Gripper OPEN Subsensor 4: "grripper CLOSED request" := 1; IF "CLOSED sensor" = 1 THEN "grripper CLOSED request" := 0; "Next_Step" := 5; END IF; "grripper CLOSED request" := 1; "grripper CLOSED request" := 1; 4M2500.7 TRUE Result TRUE "grripper CLOSED request" := 0; "grripper CLOSED request" := 0; 4M2500.7 FALSE "Next_Step" := 5; "Next_Step" 4MM2012	Pass
Step 5	Simulate ON OFF Mode Select Off III STEP 5 Counter 3 Reset Move UP Sensor Move UP Subsensor Move DOWN Sensor Move DOWN Subsensor Move LEFT Sensor Move LEFT Subsensor Move IN Sensor Move IN Subsensor Gripper CLOSED Sensor Gripper CLOSED Subsensor Move DOWN Sensor Move DOWN Subsensor Move RIGHT Sensor Move RIGHT Subsensor Move OUT Sensor Move OUT Subsensor Gripper OPEN Sensor Gripper OPEN Subsensor 5: "move UP request" := 1; IF "UP sensor" = 1 THEN "move UP request" := 0; "Next_Step" := 6; "move UP request" := 1; "move UP request" := 1; 4M2500.0 TRUE Result TRUE "move UP request" := 0; "move UP request" := 0; 4M2500.0 FALSE "Next_Step" := 6; "Next_Step" 4MM2012	Pass

Step 6		<pre>6: "move IN request" := 1; IF "IN sensor" = 1 THEN "move IN request" := 0; "Next_Step" := 7;</pre> <table><tr><td>"move IN request"</td><td>%M2500.4</td><td>TRUE</td></tr><tr><td>Result</td><td></td><td>TRUE</td></tr><tr><td>"move IN request"</td><td>%M2500.4</td><td>FALSE</td></tr><tr><td>"Next_Step"</td><td>%M2012</td><td>7</td></tr></table>	"move IN request"	%M2500.4	TRUE	Result		TRUE	"move IN request"	%M2500.4	FALSE	"Next_Step"	%M2012	7	Pass
"move IN request"	%M2500.4	TRUE													
Result		TRUE													
"move IN request"	%M2500.4	FALSE													
"Next_Step"	%M2012	7													
Step 7		<pre>7: "move DOWN request" := 1; IF "DOWN sensor" = 1 THEN "move DOWN request" := 0; "Next_Step" := 8;</pre> <table><tr><td>"move DOWN req..."</td><td>%M2500.1</td><td>TRUE</td></tr><tr><td>Result</td><td></td><td>TRUE</td></tr><tr><td>"move DOWN req..."</td><td>%M2500.1</td><td>FALSE</td></tr><tr><td>"Next_Step"</td><td>%M2012</td><td>8</td></tr></table>	"move DOWN req..."	%M2500.1	TRUE	Result		TRUE	"move DOWN req..."	%M2500.1	FALSE	"Next_Step"	%M2012	8	Pass
"move DOWN req..."	%M2500.1	TRUE													
Result		TRUE													
"move DOWN req..."	%M2500.1	FALSE													
"Next_Step"	%M2012	8													
Step 8		<pre>8: "grripper OPEN request" := 1; IF "OPEN sensor" = 1 THEN "grripper OPEN request" := 0; "Next_Step" := 9;</pre> <table><tr><td>"grripper OPEN ..."</td><td>%M2500.6</td><td>TRUE</td></tr><tr><td>Result</td><td></td><td>TRUE</td></tr><tr><td>"grripper OPEN ..."</td><td>%M2500.6</td><td>FALSE</td></tr><tr><td>"Next_Step"</td><td>%M2012</td><td>9</td></tr></table>	"grripper OPEN ..."	%M2500.6	TRUE	Result		TRUE	"grripper OPEN ..."	%M2500.6	FALSE	"Next_Step"	%M2012	9	Pass
"grripper OPEN ..."	%M2500.6	TRUE													
Result		TRUE													
"grripper OPEN ..."	%M2500.6	FALSE													
"Next_Step"	%M2012	9													
Step 9		<pre>9: "move UP request" := 1; IF "UP sensor" = 1 THEN "move UP request" := 0; "Next_Step" := 10;</pre> <table><tr><td>"move UP request"</td><td>%M2500.0</td><td>TRUE</td></tr><tr><td>Result</td><td></td><td>TRUE</td></tr><tr><td>"move UP request"</td><td>%M2500.0</td><td>FALSE</td></tr><tr><td>"Next_Step"</td><td>%M2012</td><td>10</td></tr></table> Sequence carried on as if there are two objects in the machine. Refer to previous test to visualise the rest of the sequence.	"move UP request"	%M2500.0	TRUE	Result		TRUE	"move UP request"	%M2500.0	FALSE	"Next_Step"	%M2012	10	Pass
"move UP request"	%M2500.0	TRUE													
Result		TRUE													
"move UP request"	%M2500.0	FALSE													
"Next_Step"	%M2012	10													

Figure 6; Test table of pause button, and ACT

Interlock	HMI Image	Program Image	Result
Move right unless the robot is in the up position.			Pass

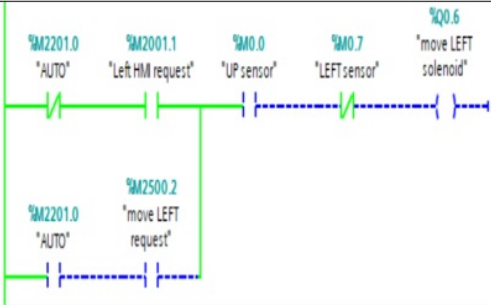
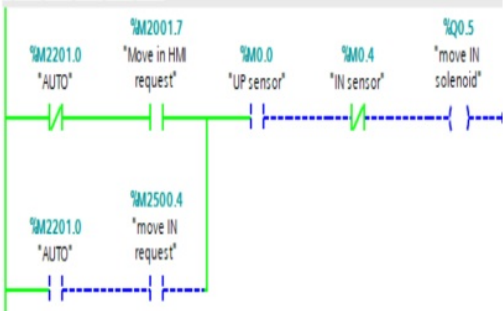
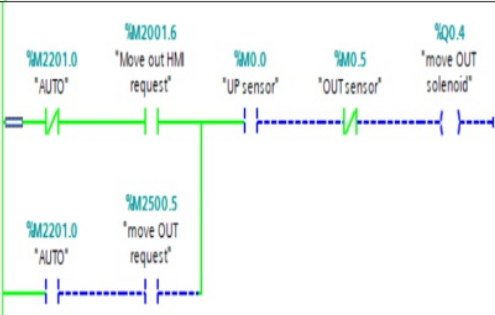
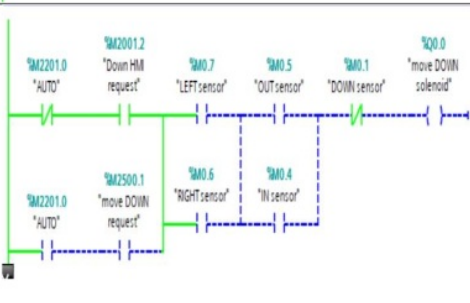
Move left unless the robot is in the up position.			Pass
Move in unless the robot is in the up position.			Pass
Move out unless the robot is in the up position.			Pass
Move down unless it is located over one of the stations.			Pass

Figure 7; Manual interlock simulation testing.

5. Practical Testing and Team Working

Due to the COVID-19 pandemic, practical testing of the program was unable to be carried out as the college was closed. Lack of practical testing means the simulation in the previous section will be used for this section.

Group discussions took place on google meet. In groups of three each student displayed their program simulating the automatic sequence. This ensured that the correct steps were carried out dependent on the number of objects in the machine. While this was being carried out my role was to notify any students if their program was missing any steps in relation to the object number. In addition to this, I checked if the switches for automatic and simulation worked correctly.

The manual interlocks were tested by trying to travel in any direction while down and trying to go down if it was not above a station.

The group discussed appropriate test tables such as the characteristics that should be tested. It was then suggested images of the program and the HMI display would be captured using snipping tool while in monitoring mode.

6. Discussion, Conclusions and Recommendations

The results gathered by practical testing were unable to be attained due to the college being closed. Practical testing was substituted for simulated testing on TIA portal and a small discussion was carried out on Google Meet.

The design of the program for the pick and place robot met all the required specifications such as an auto sequence, interlocks, manual control, and suitable PLC tags.

Maintenance can be carried out easily by replacing components and keeping to I/O addresses, but also if additional analogue inputs or digital outputs are needed, they can be expanded upon. Maintenance can also be carried out by anyone who views the program because of the internal documentation.

The reliability of the system can be documented via automatic control tests and viewing the pilot lights ensuring they are operating functionally.

In conclusion, this assignment details research on PLCs and the hardware selected for this robot. It documents the program and shows suitable testing for all specifications which should be met. Lastly, the practical testing was not carried out which was a big part, but under the current circumstances it is understandable.

This would be a problem that would affect the other functions of the gripper

The recommendations for the future would include a normally closed contact for a close gripper sensor interlock for going up. This would ensure that it picks the object up before going back up. A normally closed contact for an open sensor interlock for going down. This would ensure that the gripper is open before going down, this could prevent the gripper being damaged depending on the size of the object. An overall cycle count could be shown on the engineer screen to track how many cycles have been carried out. This could show how many cycles before a component breaks and introduce preventative maintenance. Lastly, in the test table it should include what is expected to happen in comparison to what happened.

Good range of

second hand

Correct method of

installing

References

Bolton, W., 2015. *Programmable Logic Controllers*. 6th ed. Oxford: Newnes.

FIRST4MAGNETS ; A DIVISION OF MAGNET EXPERTS, 2014. *FIRST4MAGNETS ; A DIVISION OF MAGNET EXPERTS*. [Online]

Available at: <https://www.first4magnets.com/blog/what-is-a-reed-switch-and-which-magnets-operate-them/>

[Accessed 26th January 2021].

Kamel, K. & Kamel, E., 2013. *Programmable Logic Controllers: Industrial Control*. s.l.:McGraw-Hill.

OMRON Industrial Automation, n.d. *Proximity Sensors*. [Online]

Available at: <https://www.ia.omron.com/support/guide/41/introduction.html>

[Accessed 26th January 2021].

Siemens , 2021. *Siemens Industry Mall*. [Online]

Available at:

<https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10046882#Technical%20specifications>

[Accessed 17th January 2021].

Siemens, 2021. *Data sheet 6ES7214-1AG40-0XB0*. [Online]

Available at:

<https://mall.industry.siemens.com/mall/en/ww/Catalog/DatasheetDownload?downloadUrl=tddatasheet%2F%3Fformat%3DPDF%26caller%3DMall%26mlfbs%3D6ES7214-1AG40-0XB0%26language%3Den>

[Accessed 20th January 2021].

Siemens, 2021. *Industry Mall - CPU 412H*. [Online]

Available at:

<https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10121850?tree=CatalogTree#Technical%20specifications>

[Accessed 17th January 2021].

Siemens, 2021. *Siemens Industry Mall*. [Online]

Available at:

<https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/5000014?tree=CatalogTree>

[Accessed 17th January 2021].

Siemens, 2021. *Siemens Industry Mall*. [Online]

Available at: <https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10268309>

[Accessed 17th January 21].

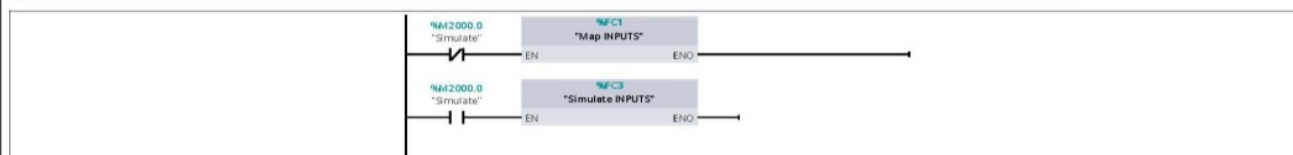
Appendix

Main [OB1]

Main Properties							
General							
Name	Main	Number	1	Type	OB	Language	LAD
Numbering	Automatic						
Information							
Title	"Main Program Sweep (Cycle)"	Author		Comment		Family	
Version	0.1	User-defined ID					
Name		Data type	Default value		Comment		
▼ Input							
Initial_Call		Bool			Initial call of this OB		
Remanence		Bool			=True, if remanent data are available		
Temp							
Constant							

Network 1: Switch between simulation and actual inputs.

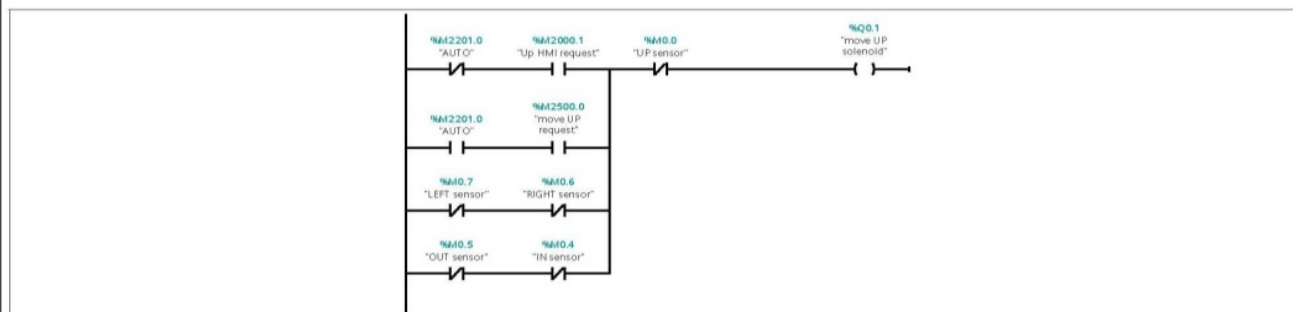
By using the simulate bit, while it is off the I/O is read to memory bits. When the simulate is on, the inputs are simulated by the function call.



Network 2: Move up

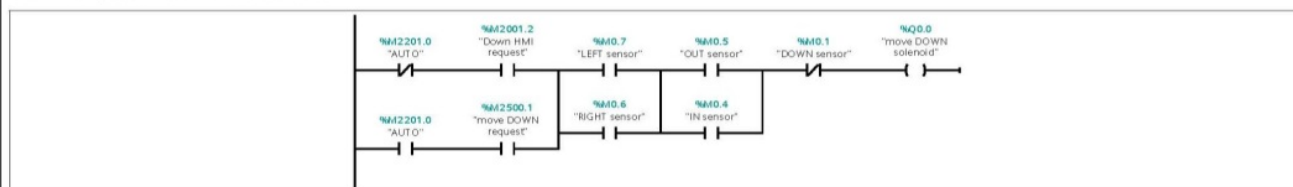
The move up solenoid will be powered dependant on a number of conditions.

1. If not in AUTO, an up HMI manual request must be closed, and the up sensor should be off. All other sensors, Left,right,in and out must be off.
2. If in auto, a move up request called by the automatic sequence, and the up sensor should be off. All other sensors, Left,right,in or out must be off.



Network 3: Move down

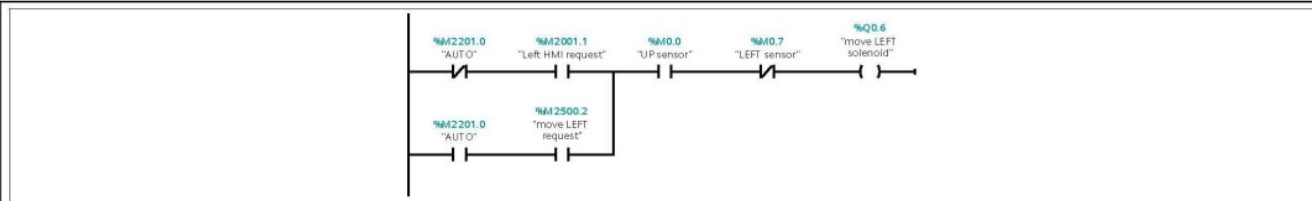
Move down solenoid coil will activate, if in AUTO move down request will have to be called from the sequence. If its not in AUTO then it will have to be manually controlled on the HMI. Then either left OR right in combination with out OR in provided that the down sensor is off. It cannot go down if the down sensor is on because it could cause damage and crash into the stations.



Network 4: Move left

The move left solenoid will be activated, in 1 of 2 ways.

1. Auto is off, then a manual left HMI request called, the up sensor must be on, and the left sensor off.
 2. Auto is on, then a move left request is called from the automatic sequence program, the up sensor must be on and the left sensor must be off.
- NOTE: Up sensor must be on because the robot should not travel when down which could cause damage to the machine. Left sensor must be off, otherwise it will not stop trying to go left, and cannot go any further.



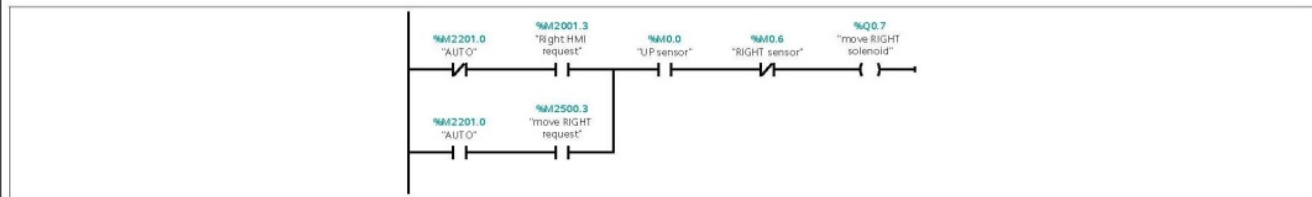
Network 5: Move right

The move right solenoid will be activated, in 1 of 2 ways.

1. Auto is off, then a manual HMI request called, the up sensor must be on, and the right sensor off.

2. Auto is on, then a move right request is called from the automatic sequence program, the up sensor must be on and the right sensor must be off.

NOTE: Up sensor must be on because the robot should not travel when down which could cause damage to the machine. Right sensor must be off, otherwise it will not stop trying to go right, and cannot go any further.



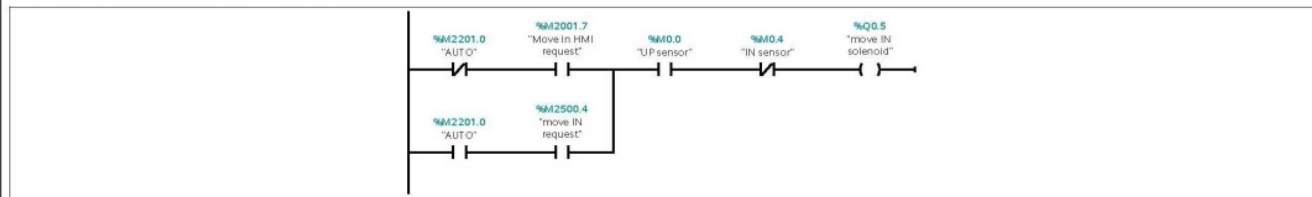
Network 6: Move in

The move in solenoid will be activated, in 1 of 2 ways.

1. Auto is off, then a manual Move in HMI request called, the up sensor must be on, and the in sensor off.

2. Auto is on, then a move in request is called from the automatic sequence program, the up sensor must be on and the in sensor must be off.

NOTE: Up sensor must be on because the robot should not travel when down which could cause damage to the machine. In sensor must be off, otherwise it will not stop trying to go in, and cannot go any further.



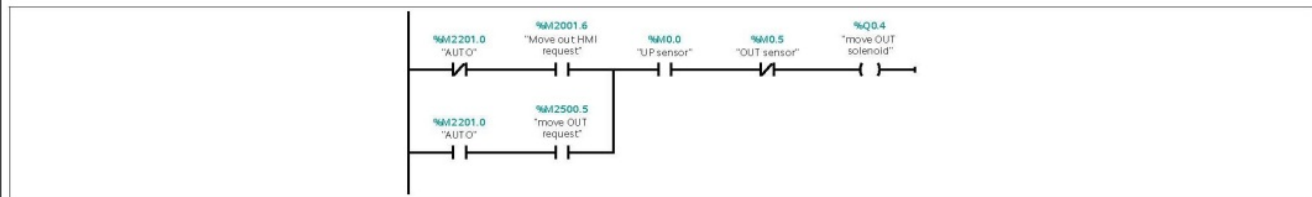
Network 7: Move out

The move out solenoid will be activated, in 1 of 2 ways.

1. Auto is off, then a manual Move out HMI request called, the up sensor must be on, and the out sensor off.

2. Auto is on, then a move out request is called from the automatic sequence program, the up sensor must be on and the out sensor must be off.

NOTE: Up sensor must be on because the robot should not travel when down because it could cause damage to the machine. Out sensor must be off, otherwise it will not stop trying to go out and cannot go any further.



Network 8: Auto sequence pause push button.

This uses a memory bit as a N/C switch, meaning if the push button is pressed it will de energise the rung which will stop the sequence.



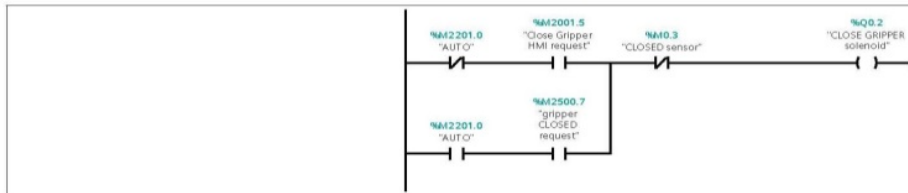
Network 9: Close gripper

The close gripper solenoid will be activated, in 1 of 2 ways.

1. Auto is off, then a manual close gripper HMI request called and the closed sensor must be off.

2. Auto is on, then a gripper closed request is called from the automatic sequence program and the closed sensor must be off.

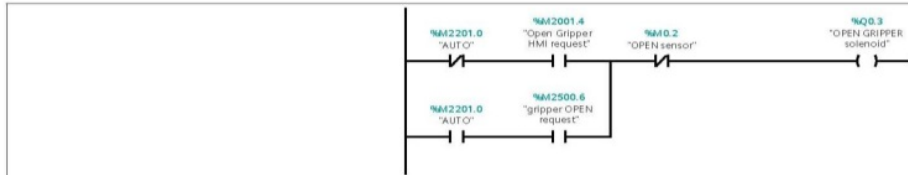
NOTE: Closed sensor must be off, otherwise it will not stop trying to close which could cause damage.



Network 10: Open gripper

The open gripper solenoid will be activated, in 1 of 2 ways.

1. Auto is off, then a manual open gripper HMI request called and the open sensor must be off.
 2. Auto is on, then a gripper open request is called from the automatic sequence program and the open sensor must be off.
- NOTE: Open sensor must be off, otherwise it will not stop trying to open which could cause damage.



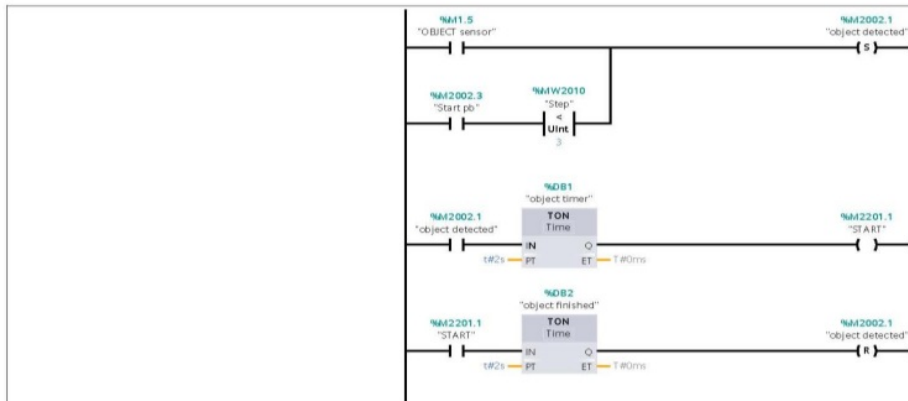
Network 11: Object detected.

If the proximity switch sends a signal, then an object will be detected.

If the start push button is pressed, and it is less than its 3rd step, an object will be detected.

If an object is detected, it will wait 2 seconds, then start the sequence.

If start sequence activated, after 2 seconds an it will reset the coil object detected, meaning that the task has been completed.



Network 12: Object counter

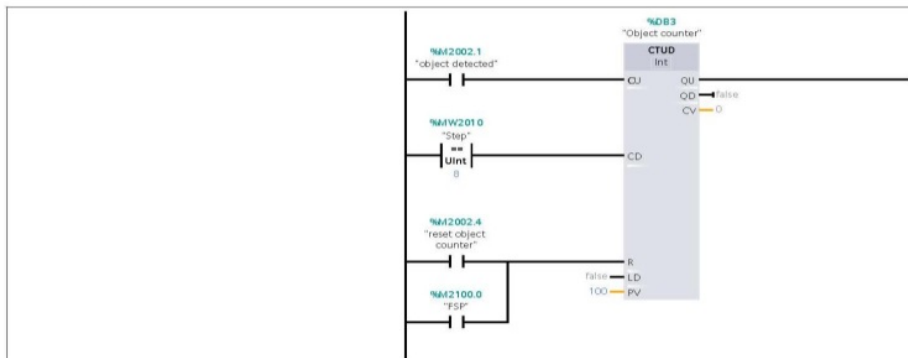
This is the object counter.

If the object detected activates its linked to CU (Count up)

If at step 8, it will count down.

If first scan pulse (FSP) or reset object counter is pressed it will reset the counter.

If a high signal of 100 on LD, it will be copied to CV.



Sequence [FC2]

Sequence Properties

Sequence Properties								
General								
Name	Sequence	Number	2	Type	FC	Language	SCL	
Numbering	Automatic							
Information								
Title				Author			Family	
Version	0.1	User-defined ID						
Name		Data type	Default value		Comment			
Input								
Output								
InOut								
Temp								
Constant								
▼ Return								
Sequence		Void						

```

0001 IF ("FSP" = 1) OR ("AUTO" = 0) THEN
0002     "Step" := 0; // reset all values to 0.
0003     "Next_Step" := 0;
0004     "move LEFT request" := 0;
0005     "move RIGHT request" := 0;
0006     "move IN request" := 0;
0007     "move OUT request" := 0;
0008     "gripper OPEN request" := 0;
0009     "gripper CLOSED request" := 0;
0010 END_IF;
0011 // Step numbers are located before comments past this point.
0012 CASE "Step" OF
0013 0: //This step ensures that the robot is in the up position.
0014     "move UP request" := 1;
0015     IF "UP sensor" = 1 THEN
0016         "move UP request" := 0;
0017         "Next_Step" := 1;
0018     END_IF;
0019 1: // Move home to station 4; if all sensors made and start button engaged;
0020     // 1 object then skip to step 18, 2 object skip to step 10, 3 objects then go to step 2.
0021     "move IN request" := 1;
0022     "move RIGHT request" := 1;
0023     "gripper OPEN request" := 1;
0024     IF ("IN sensor" = 1) AND
0025         ("RIGHT sensor" = 1) AND
0026         ("OPEN sensor" = 1) AND
0027         ("START" = 1) THEN
0028         "move RIGHT request" := 0;
0029         "move IN request" := 0;
0030         "gripper OPEN request" := 0;
0031         IF "Object counter".CV = 1 THEN
0032             "Next_Step" := 18;
0033         END_IF;
0034         IF "Object counter".CV = 2 THEN
0035             "Next_Step" := 10;
0036         END_IF;
0037         IF "Object counter".CV = 3 THEN
0038             "Next_Step" := 2;
0039         END_IF;
0040     END_IF;
0041 2: // Move out; from station 4 to station 3.
0042     "move OUT request" := 1;
0043     IF "OUT sensor" = 1 THEN
0044         "move OUT request" := 0;
0045         "Next_Step" := 3;
0046     END_IF;
0047 3: // Move down; this lowers the grabber on station 3.
0048     "move DOWN request" := 1;
0049     IF "DOWN sensor" = 1 THEN
0050         "move DOWN request" := 0;
0051         "Next_Step" := 4;
0052     END_IF;
0053 4: // Gripper close; closes the gripper to grab the object from station 3.
0054     "gripper CLOSED request" := 1;
0055     IF "CLOSED sensor" = 1 THEN
0056         "gripper CLOSED request" := 0;
0057         "Next_Step" := 5;
0058     END_IF;
0059 5: // Move up; moves arm up above station 3.
0060     "move UP request" := 1;
0061     IF "UP sensor" = 1 THEN
0062         "move UP request" := 0;
0063         "Next_Step" := 6;
0064     END_IF;
0065 6: // Move in ; moves back to station 4.
0066     "move IN request" := 1;

```

```

0067 IF "IN sensor" = 1 THEN
0068     "move IN request" := 0;
0069     "Next_Step" := 7;
0070 END_IF;
0071 7:
0072     "move DOWN request" := 1;
0073     IF "DOWN sensor" = 1 THEN
0074         "move DOWN request" := 0;
0075         "Next_Step" := 8;
0076     END_IF;
0077     8:
0078         "gripper OPEN request" := 1;
0079         IF "OPEN sensor" = 1 THEN
0080             "gripper OPEN request" := 0;
0081             "Next_Step" := 9;
0082         END_IF;
0083         9:
0084             "move UP request" := 1;
0085             IF "UP sensor" = 1 THEN
0086                 "move UP request" := 0;
0087                 "Next_Step" := 10;
0088             END_IF;
0089             10:
0090                 "move LEFT request" := 1;
0091                 IF "LEFT sensor" = 1 THEN
0092                     "move LEFT request" := 0;
0093                     "Next_Step" := 11;
0094                 END_IF;
0095                 11:
0096                     "move OUT request" := 1;
0097                     IF "OUT sensor" = 1 THEN
0098                         "move OUT request" := 0;
0099                         "Next_Step" := 12;
0100                     END_IF;
0101                     12:
0102                         "move DOWN request" := 1;
0103                         IF "DOWN sensor" = 1 THEN
0104                             "move DOWN request" := 0;
0105                             "Next_Step" := 13;
0106                         END_IF;
0107                         13:
0108                             "gripper CLOSED request" := 1;
0109                             IF "CLOSED sensor" = 1 THEN
0110                                 "gripper CLOSED request" := 0;
0111                                 "Next_Step" := 14;
0112                             END_IF;
0113                             14:
0114                                 "move UP request" := 1;
0115                                 IF "UP sensor" = 1 THEN
0116                                     "move UP request" := 0;
0117                                     "Next_Step" := 15;
0118                                 END_IF;
0119                                 15:
0120                                     "move RIGHT request" := 1;
0121                                     IF "RIGHT sensor" = 1 THEN
0122                                         "move RIGHT request" := 0;
0123                                         "Next_Step" := 16;
0124                                     END_IF;
0125                                     16:
0126                                         "move DOWN request" := 1;
0127                                         IF "DOWN sensor" = 1 THEN
0128                                             "move DOWN request" := 0;
0129                                             "Next_Step" := 17;
0130                                         END_IF;
0131                                         17:
0132                                             "gripper OPEN request" := 1;
0133                                             IF "OPEN sensor" = 1 THEN
0134                                                 "gripper OPEN request" := 0;
0135                                                 "Next_Step" := 18;
0136                                             END_IF;
0137                                             18:
0138                                                 "move UP request" := 1;
0139                                                 IF "UP sensor" = 1 THEN
0140                                                     "move UP request" := 0;
0141                                                     "Next_Step" := 19;
0142                                                 END_IF;
0143                                                 19:
0144                                                     "move LEFT request" := 1;
0145                                                     IF "LEFT sensor" = 1 THEN
0146                                                         "move LEFT request" := 0;
0147                                                         "Next_Step" := 20;
0148                                                     END_IF;
0149                                                     20:
0150                                                         "move IN request" := 1;
0151                                                         IF "IN sensor" = 1 THEN
0152                                                             "move IN request" := 0;
0153                                                             "Next_Step" := 21;
0154                                                     END_IF;

```

```

0155 21: // Move down; lowers onto station 1.
0156 "move DOWN request" := 1;
0157 IF "DOWN sensor" = 1 THEN
0158   "move DOWN request" := 0;
0159   "Next_Step" := 22;
0160 END_IF;
0161 22: // Gripper close; grabs object on station 1.
0162 "gripper CLOSED request" := 1;
0163 IF "CLOSED sensor" = 1 THEN
0164   "gripper CLOSED request" := 0;
0165   "Next_Step" := 23;
0166 END_IF;
0167 23: // Move up; moves above station 1.
0168 "move UP request" := 1;
0169 IF "UP sensor" = 1 THEN
0170   "move UP request" := 0;
0171   "Next_Step" := 24;
0172 END_IF;
0173 24: // Move out; moves to station 2.
0174 "move OUT request" := 1;
0175 IF "OUT sensor" = 1 THEN
0176   "move OUT request" := 0;
0177   "Next_Step" := 25;
0178 END_IF;
0179 25: // Move down; moves down to station 2.
0180 "move DOWN request" := 1;
0181 IF "DOWN sensor" = 1 THEN
0182   "move DOWN request" := 0;
0183   "Next_Step" := 26;
0184 END_IF;
0185 26: // Gripper open; releases object on station 2.
0186 "gripper OPEN request" := 1;
0187 IF "OPEN sensor" = 1 THEN
0188   "gripper OPEN request" := 0;
0189   "Next_Step" := 0; // back to step 0, which resets all values.
0190 END_IF;
0191 END_CASE;
0192 "Step" := "Next_Step"; // if step go next step.
0193
0194

```

Symbol	Address	Type	Comment
"AUTO"	%M2201.0	Bool	
"CLOSED sensor"	%M0.3	Bool	
"DOWN sensor"	%M0.1	Bool	
"FSP"	%M2100.0	Bool	
"gripper CLOSED request"	%M2500.7	Bool	
"gripper OPEN request"	%M2500.6	Bool	
"IN sensor"	%M0.4	Bool	
"LEFT sensor"	%M0.7	Bool	
"move DOWN request"	%M2500.1	Bool	
"move IN request"	%M2500.4	Bool	
"move LEFT request"	%M2500.2	Bool	
"move OUT request"	%M2500.5	Bool	
"move RIGHT request"	%M2500.3	Bool	
"move UP request"	%M2500.0	Bool	
"Next_Step"	%MW2012	UInt	
"Object counter".CV		Int	
"OPEN sensor"	%M0.2	Bool	
"OUT sensor"	%M0.5	Bool	
"RIGHT sensor"	%M0.6	Bool	
"START"	%M2201.1	Bool	
"Step"	%MW2010	UInt	
"UP sensor"	%M0.0	Bool	

PLC tags / Assignment tag table [74]

PLC tags

PLC tags									
	Name	Data type	Address	Retain	Accessi- ble from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engi- neering	Supervision	Comment
	Simulate	Bool	%M2000.0	False	True	True	True		
	PLC Base Unit input byte 0	Byte	%IB0	False	True	True	True		
	PLC Base Unit mapped input byte 0	Byte	%MB0	False	True	True	True		
	PLC Base Unit input byte 1	Byte	%IB1	False	True	True	True		
	PLC Base Unit mapped input byte 1	Byte	%MB1	False	True	True	True		
	move UP solenoid	Bool	%Q0.1	False	True	True	True		
	move DOWN solenoid	Bool	%Q0.0	False	True	True	True		
	move LEFT solenoid	Bool	%Q0.6	False	True	True	True		
	move RIGHT solenoid	Bool	%Q0.7	False	True	True	True		
	move IN solenoid	Bool	%Q0.5	False	True	True	True		
	move OUT solenoid	Bool	%Q0.4	False	True	True	True		
	OPEN GRIPPER solenoid	Bool	%Q0.3	False	True	True	True		
	CLOSE GRIPPER solenoid	Bool	%Q0.2	False	True	True	True		
	Step	UInt	%MW2010	False	True	True	True		
	Next_Step	UInt	%MW2012	False	True	True	True		
	move UP request	Bool	%M2500.0	False	True	True	True		
	move DOWN request	Bool	%M2500.1	False	True	True	True		
	move LEFT request	Bool	%M2500.2	False	True	True	True		
	move RIGHT request	Bool	%M2500.3	False	True	True	True		
	move IN request	Bool	%M2500.4	False	True	True	True		
	move OUT request	Bool	%M2500.5	False	True	True	True		
	gripper OPEN request	Bool	%M2500.6	False	True	True	True		
	gripper CLOSED request	Bool	%M2500.7	False	True	True	True		
	AUTO	Bool	%M2201.0	False	True	True	True		
	START	Bool	%M2201.1	False	True	True	True		
	Up HMI request	Bool	%M2000.1	False	True	True	True		
	Left HMI request	Bool	%M2001.1	False	True	True	True		
	Down HMI request	Bool	%M2001.2	False	True	True	True		
	Right HMI request	Bool	%M2001.3	False	True	True	True		
	UP sensor	Bool	%M0.0	False	True	True	True		
	DOWN sensor	Bool	%M0.1	False	True	True	True		
	OPEN sensor	Bool	%M0.2	False	True	True	True		
	CLOSED sensor	Bool	%M0.3	False	True	True	True		
	IN sensor	Bool	%M0.4	False	True	True	True		
	OUT sensor	Bool	%M0.5	False	True	True	True		
	RIGHT sensor	Bool	%M0.6	False	True	True	True		
	LEFT sensor	Bool	%M0.7	False	True	True	True		
	Open Gripper HMI request	Bool	%M2001.4	False	True	True	True		
	Close Gripper HMI request	Bool	%M2001.5	False	True	True	True		
	Move out HMI request	Bool	%M2001.6	False	True	True	True		
	Move in HMI request	Bool	%M2001.7	False	True	True	True		
	Pause pb	Bool	%M2002.0	False	True	True	True		
	object detected	Bool	%M2002.1	False	True	True	True		
	object ready	Bool	%M2002.2	False	True	True	True		
	Start pb	Bool	%M2002.3	False	True	True	True		
	reset object counter	Bool	%M2002.4	False	True	True	True		

PLC tags / Simulation Tags [6]

PLC tags

PLC tags

	Name	Data type	Address	Retain	Accessi- ble from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engi- neering	Supervision	Comment
	X10 Pushbutton	Bool	%M1.0	False	True	True	True		
	X11 Pushbutton	Bool	%M1.1	False	True	True	True		
	X12 Pushbutton	Bool	%M1.2	False	True	True	True		
	X13 toggle switch	Bool	%M1.3	False	True	True	True		
	X14 toggle switch	Bool	%M1.4	False	True	True	True		
	OBJECT sensor	Bool	%M1.5	False	True	True	True		

PLC tags / System Tags [14]

PLC tags

PLC tags

	Name	Data type	Address	Retain	Accessi- ble from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engi- neering	Supervision	Comment
	System_Byte	Byte	%MB2100	False	True	True	True		
	FSP	Bool	%M2100.0	False	True	True	True		
	DiagStatusUpdate	Bool	%M2100.1	False	True	True	True		
	AlwaysTRUE	Bool	%M2100.2	False	True	True	True		
	AlwaysFALSE	Bool	%M2100.3	False	True	True	True		
	Clock_Byte	Byte	%MB2101	False	True	True	True		
	Clock_10Hz	Bool	%M2101.0	False	True	True	True		
	Clock_5Hz	Bool	%M2101.1	False	True	True	True		
	Clock_2.5Hz	Bool	%M2101.2	False	True	True	True		
	Clock_2Hz	Bool	%M2101.3	False	True	True	True		
	Clock_1.25Hz	Bool	%M2101.4	False	True	True	True		
	Clock_1Hz	Bool	%M2101.5	False	True	True	True		
	Clock_0.625Hz	Bool	%M2101.6	False	True	True	True		
	Clock_0.5Hz	Bool	%M2101.7	False	True	True	True		

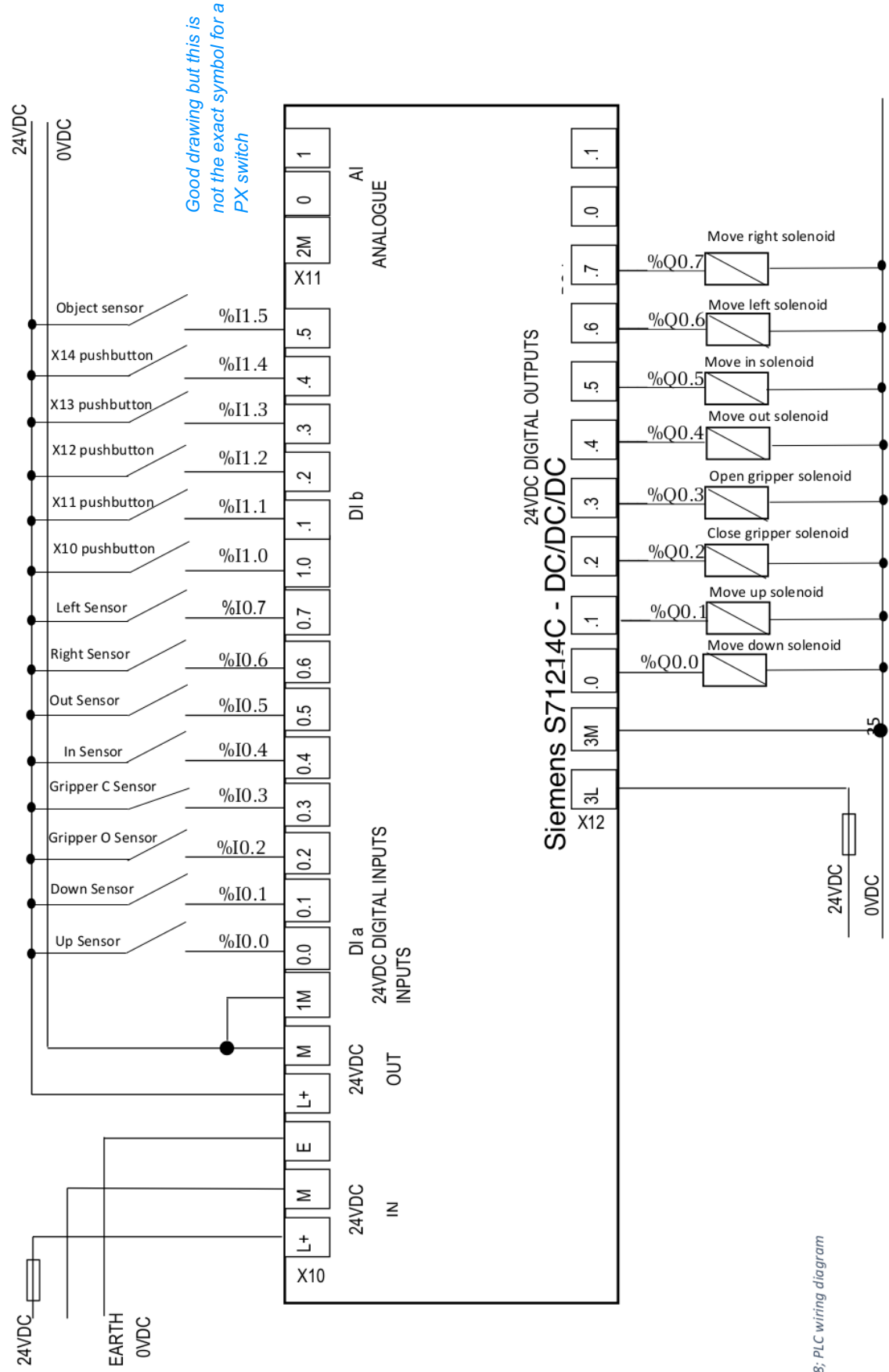


Figure 8: PLC wiring diagram

FINAL GRADE

GENERAL COMMENTS

70/100**Instructor**

PAGE 1

PAGE 2

QM

Good Structure

Report is structured exactly as stated in the assignment brief.

PAGE 3

QM

Use of 11pt Arial, full justification with 1.5 line spacing allows easier reading and marking

Your presentation could be improved making the text easier to read and mark. Your tutor will then be able to add comments between lines

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Grocer's apostrophe

If you meant this as a plural word, don't use an apostrophe. This also applies to acronyms.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

PAGE 4

Text Comment. One citation would be enough since these are identical, they are like 4 buses arriving at the same time!

PAGE 5

Text Comment. Excellent table to help choose the best PLC !

PAGE 6

QM

Use of 11pt Arial, full justification with 1.5 line spacing allows easier reading and marking

Your presentation could be improved making the text easier to read and mark. Your tutor will then be able to add comments between lines

QM

Incorrectly cited, this is not the Harvard Method

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Citation Needed

Cite Source:

Please use the link below to find links to information regarding specific citation styles:

http://www.plagiarism.org/plag_article_citation_styles.html

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

PAGE 7

Text Comment. Addresses of inputs should be given a 'I' not 'M' in this table

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

Text Comment. A very good SFC chart

Text Comment. where does the object sensor fit here?

PAGE 15

PAGE 16

PAGE 17

Text Comment. I like what you have done with this table but it could be better if the evidence was in the appendix and actual results described.

QM

Testing Table could be better

This table should contain five columns: (1) description of test, (2) expected result (3) actual result (4) pass-fail decision (5) reference to evidence as image in the appendix.

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

Text Comment. This would be a problem that would affect the other functions of the gripper

PAGE 26

QM

Good range of sources

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

QM

Correct method of listing.

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

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

Text Comment. Good drawing but this is not the exact symbol for a PX switch

GRADING FORM: NEW CSG-RUBRIC

MATTHEW
PERRY

70

1. ENGAGEMENT WITH LITERATURE

Very good evidence of research from a wide range of



sources correctly cited and listed. Correct conclusion reached with a helpful table of comparison. (14/20)

14

2. KNOWLEDGE AND UNDERSTANDING



Good explanations provided for the hardware and the software giving clear evidence of knowledge and understanding. (15/20)

15

3. COGNITIVE AND INTELLECTUAL



There is good evidence of analysis through testing, discussion and conclusions. A good testing table provided but it could be improved. (14/20)

14

4. PRACTICAL APPLICATION



Program is well designed and internally documented with good drawing and SFC chart. The testing table is good but could be better. It should contain five columns: (1) description of test, (2) expected result (3) actual result (4) pass-fail decision (5) reference to evidence as image in the appendix. (14/20)

14

5. TRANSFERABLE SKILLS



Report is well structured but it would be easier to read and mark if it was set to 1.5 line spacing with full justification of text. Good clear evidence of group working within Covid-19 restrictions. (13/20)

13