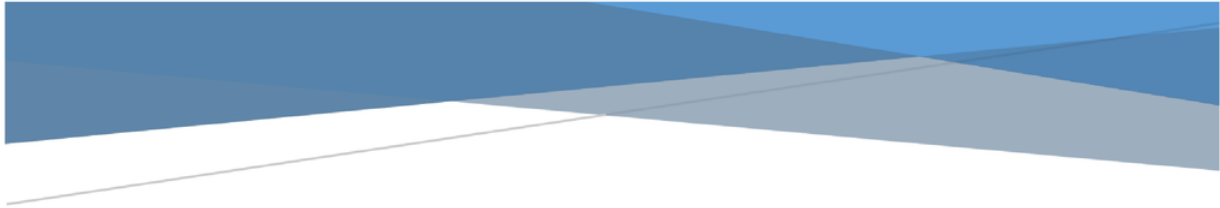


PSDE - 4005 Assignment 2

by Jac Williams

FILE	11945_JAC_WILLIAMS_PSDE_-_4005_ASSIGNMENT_2_156961_476529233.DOCX		
TIME SUBMITTED	12-MAR-2018 11:11AM (UTC+0000)	WORD COUNT	5479
SUBMISSION ID	84511676	CHARACTER COUNT	37805



HNC ELECTRICAL AND ELECTRONIC ENGINEERING PSDE 4005 - PLCs

Assignment 2: Pick and Place Robot

Submission Date: 12/03/18
Name: Jac Williams
Company: TATA Steel

Table of Contents

Table of Tables	1
Table of Figures.....	2
1 - Hardware Investigation, System design and I/O List	4
I/O List:.....	8
2 - Program Documentation and Explanation	10
SFC:.....	10
Program explanation.....	14
Main:	14
Outputs:	16
Y-Axis solenoids:	16
Gripper Solenoids:.....	18
Z-Axis Solenoids:	19
X-Axis Solenoids:	20
Sequence:.....	21
Counters:.....	39
Timers:	41
HMI SCREENS:	42
Operator Screen:.....	42
Engineering Screen:	43
PLC Tags:	45
3 – System Testing, maintenance and Reliability	47
Testing Table:	47
4 – Conclusions and Recommendations	63

Table of Tables

Table 1 - I/O list.....	8
-------------------------	---

Table of Figures

Figure 1 - Pick and Place Robot	4
Figure 2 - Siemens PLC CPU	4
Figure 3 - Power Supply	5
Figure 4 - Electrical Panel	5
Figure 5 - Door Interlock	6
Figure 6 - Air Regulator	6
Figure 7 - Manifold	6
Figure 8 - Reed Switches	7
Figure 9 - Object Sensor	7
Figure 10 - Main: Network 1 "Outputs"	14
Figure 11 - Main: Network 2 "Timers"	14
Figure 12 - Main: Network 3 "Sequence"	14
Figure 13 - Main: Network 4 "Counters"	15
Figure 14 - Main: Network 5 "FSP"	15
Figure 15 - Outputs: Network 1 "Move Down"	16
Figure 16 - Outputs: Network 2 "Move Up"	17
Figure 17 - Outputs: Network 3 "Close Gripper"	18
Figure 18 - Outputs: Network 4 "Open Gripper"	18
Figure 19 - Outputs: Network 5 "Move Out"	19
Figure 20 - Outputs: Network 6 "Move In"	19
Figure 21 - Outputs: Network 7 "Move Left"	20
Figure 22 - Outputs: Network 8 "Move Right"	20
Figure 23 - Sequence: FSP	21
Figure 24 - Sequence: Step 0	21
Figure 25 - Sequence: Step 1	22
Figure 26 - Sequence: Steps 2 & 3	23
Figure 27 - Sequence: Steps 4, 5, 6	24
Figure 28 - Sequence: Steps 7, 8, 9	25
Figure 29 - Sequence: Steps 10, 11, 12	26
Figure 30 - Sequence: Steps 13, 14, 15	27
Figure 31 - Sequence: Steps 16, 17, 18	28
Figure 32 - Sequence: Steps 19, 20, 21	29
Figure 33 - Sequence: Steps 22, 23, 30	30
Figure 34 - Sequence: Steps 31, 32, 33	31
Figure 35- Sequence: Steps 34, 35, 36	32
Figure 36 - Sequence: Steps 37, 38, 39	33
Figure 37- Sequence: Steps 40, 41, 42	34
Figure 38- Sequence: Steps 43, 44, 45	35
Figure 39- Sequence: Steps 46, 47, 48	36
Figure 40- Sequence: Steps 49, 50, 51	37
Figure 41- Sequence: Steps 52, 53, & "Step := Next Step"	38
Figure 42 - Counters: Network 1 "Object Count"	39
Figure 43 - Counters: Network 2 "Object Sensor De-Bouncing"	39
Figure 44 - De-Bouncing Illustration	40
Figure 45 - Counters: Network 3 "Objects Processed"	40
Figure 46 - Timers: Network 1 "5 Second Timer"	41

Figure 47 - HMI: Operator Screen	42
Figure 48 - HMI: Engineering Screen	43
Figure 49 - HMI: Login prompt.....	44
Figure 50 HMI: Administrator credentials	44
Figure 51 - HMI: logoff event	44
Figure 52 - PLC Taglist: 1-32	45
Figure 53 - PLC Taglist: 33-48.....	45
Figure 54 - Loading Seq Test: Step 4	50
Figure 55 - Loading Seq Test: Step 1	50
Figure 56 - Loading Seq Test: Step 12	51
Figure 57 - Loading Seq Test: Step 9	51
Figure 58 - Loading Seq Test: Step 17	52
Figure 59 - Loading Seq Test: Step 14	52
Figure 60 - Loading Seq Test: Returned Home	53
Figure 61 - Loading Seq Test: Step 22	53
Figure 62 - Unloading Seq test: Step 32.....	54
Figure 63 - Unloading Seq test: Step 30.....	54
Figure 64 - Unloading Seq test: Step 38.....	55
Figure 65 - Unloading Seq test: Step 34.....	55
Figure 66 - Unloading Seq test: Step 47.....	56
Figure 67 - Unloading Seq test: Step 44.....	56
Figure 68 - Unloading Seq test: Step 52.....	57
Figure 69 - Unloading Seq test: Step 50.....	57
Figure 70 - Unloading Seq test: Returned Home	58
Figure 71 - Manual Test: Down movement	59
Figure 72 - Manual Test: Up movement	59
Figure 73 - Manual Test: Out Movement & Close Grip	60
Figure 74 - Manual Test: In movement & Open Gripper	60
Figure 75 - Manual Test: Right Movement	61
Figure 76 - HMI: Objects processed count test.....	62

1 - Hardware Investigation, System design and I/O List

In this assignment we are to create a programme and HMI control for the pick and place robot. This must feature manual control, an automatic loading sequence, and an unloading sequence as specified in the brief.

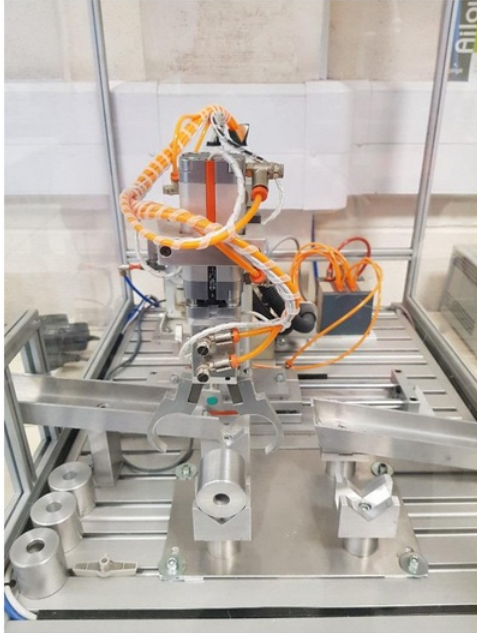


Figure 1 - Pick and Place Robot

The pick and place robot is controlled by a Siemens PLC, namely a CPU1214C as can be seen in Figure 2.

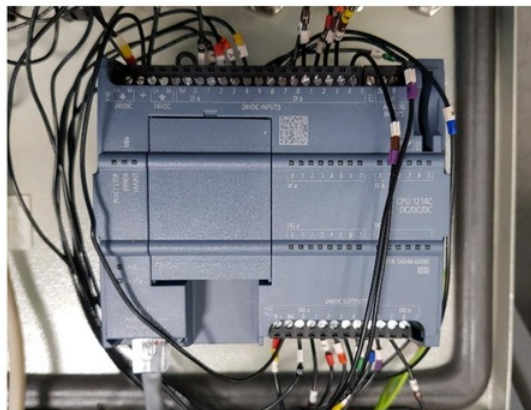


Figure 2 - Siemens PLC CPU

This supply for this CPU is the 240VAC to 24DC power supply shown in Figure 3.



Figure 3 - Power Supply

These are safely housed in a panel situated beneath the pick and place robot (Figure 4).

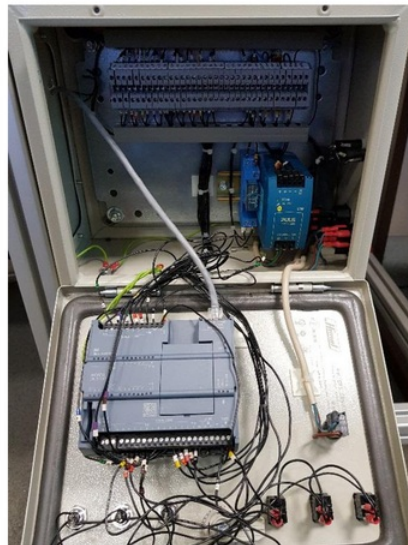


Figure 4 - Electrical Panel

The robot is pneumatically controlled, with an air regulator (which allows control of pressure and can restrict the air flow as necessary) as seen in Figure 6, being brought to a manifold (Figure 7) with

solenoid valves which allow air to flow to the desired pistons, thus moving the robot. The air regulator is interlocked with a magnetic safety switch (Figure 5) situated on the door, so that if at any point the door is opened the air supply to the robot is restricted. This stops the robot from being able to physically move (although electrically the robot is not isolated) whilst the door is open, ensuring no one can be physically injure themselves or tamper with the robot during operation.



Figure 6 - Air Regulator



Figure 5 - Door Interlock



Figure 7 - Manifold

The manifold/solenoid block then supplies air to the pneumatic pistons as determined by the programme. The solenoids valves used are memory solenoid valves and retain their state, and therefore do not need to be latched in the program.

In order to track the position of the robot and limit its movement appropriately reed switches are employed, these can be seen highlighted in Figure 8. Reed switches are non-contact switches that switch states when a magnet is nearby. Magnets are present on the pistons themselves, and thus as the piston moves the reed switches change states allowing us to track movement of the pistons.

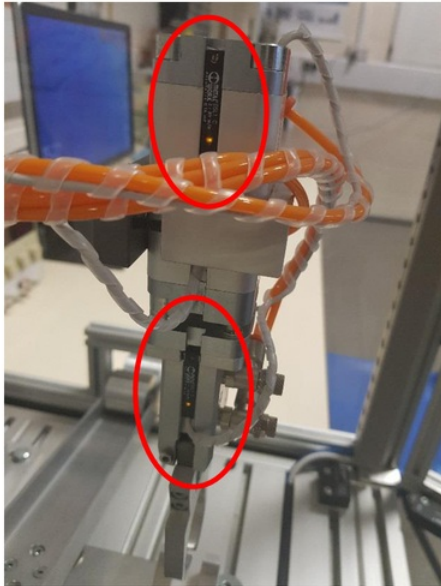


Figure 8 - Reed Switches

The object sensor is situated at station 1 as displayed in Figure 9, and is an inductive proximity detector, meaning only metal objects will be picked up.

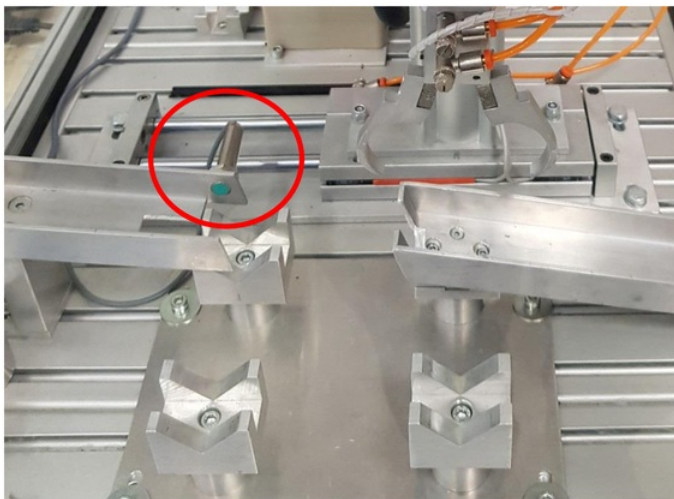
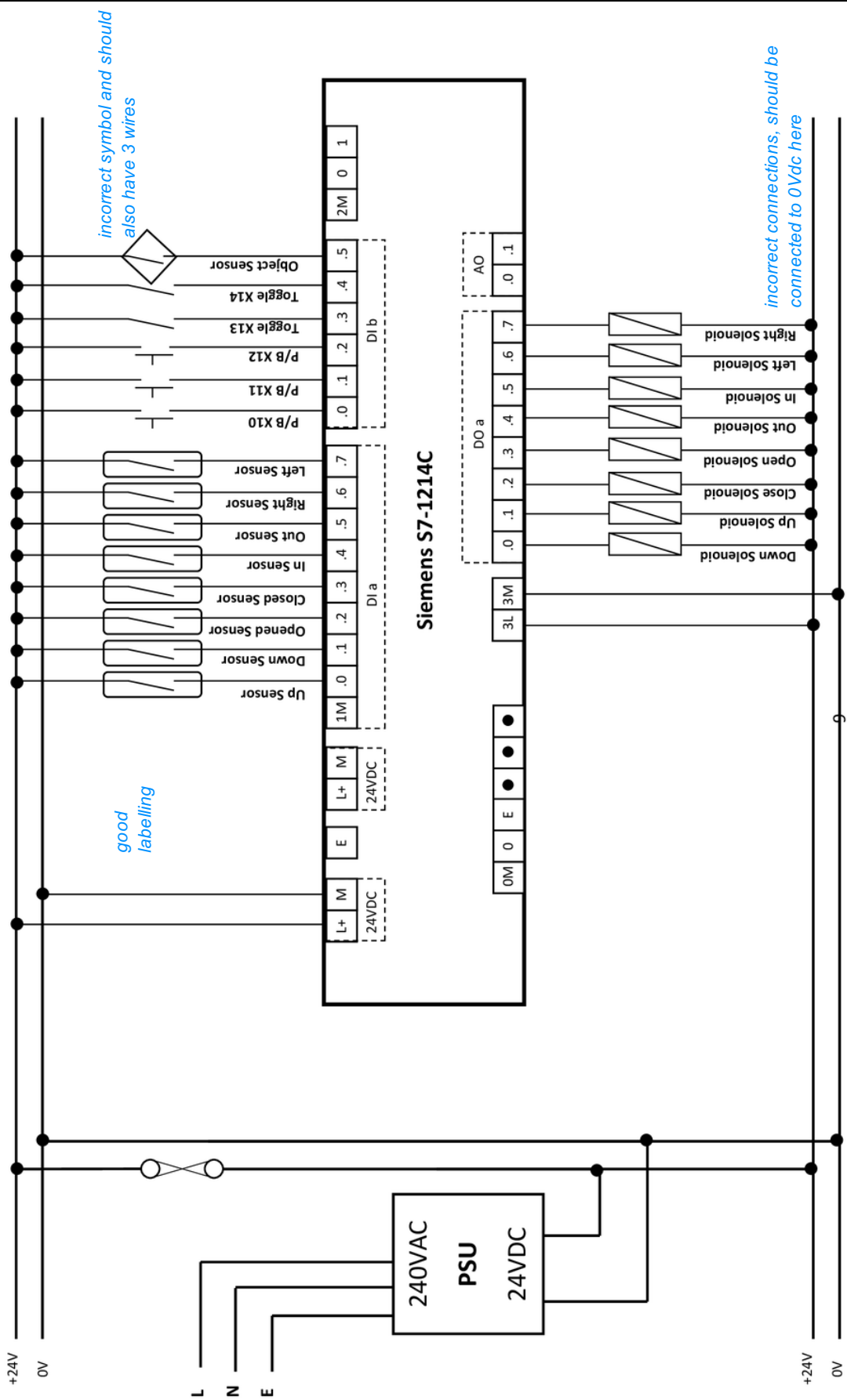


Figure 9 - Object Sensor

I/O List:

Inputs		Outputs	
I0.0	Up sensor	Q0.0	Move Down solenoid
I0.1	Down sensor	Q0.1	Move Up solenoid
I0.2	Gripper Open sensor	Q0.2	Close Gripper solenoid
I0.3	Gripper Closed sensor	Q0.3	Open Gripper solenoid
I0.4	In sensor	Q0.4	Move Out solenoid
I0.5	Out sensor	Q0.5	Move In solenoid
I0.6	Right sensor	Q0.6	Move Left solenoid
I0.7	Left sensor	Q0.7	Move Right solenoid
I1.0	PB X10		
I1.1	PB X11		
I1.2	PB X12		
I1.3	Toggle X13		
I1.4	Toggle X14		
I1.5	Object sensor		

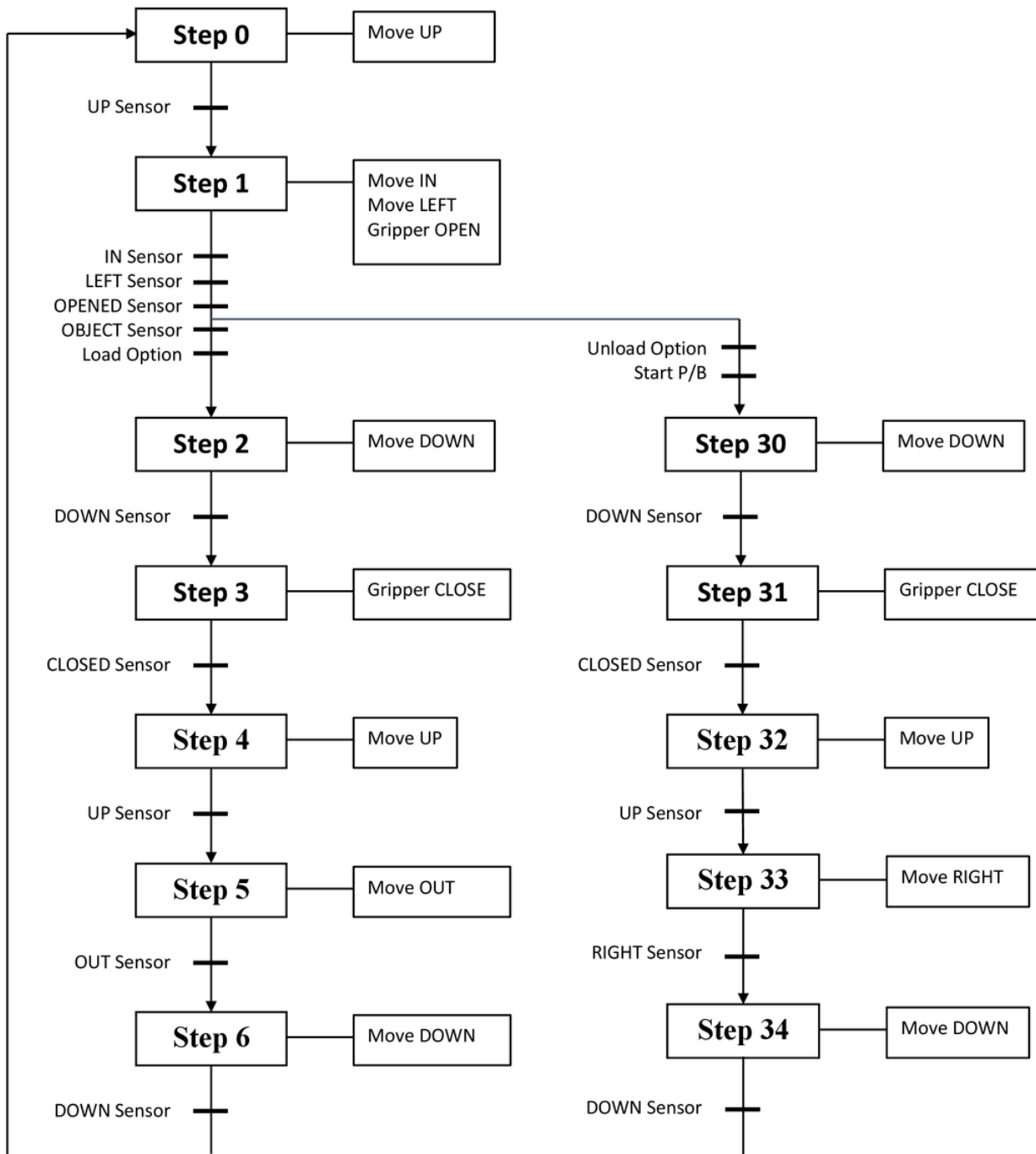
Table 1 - I/O list

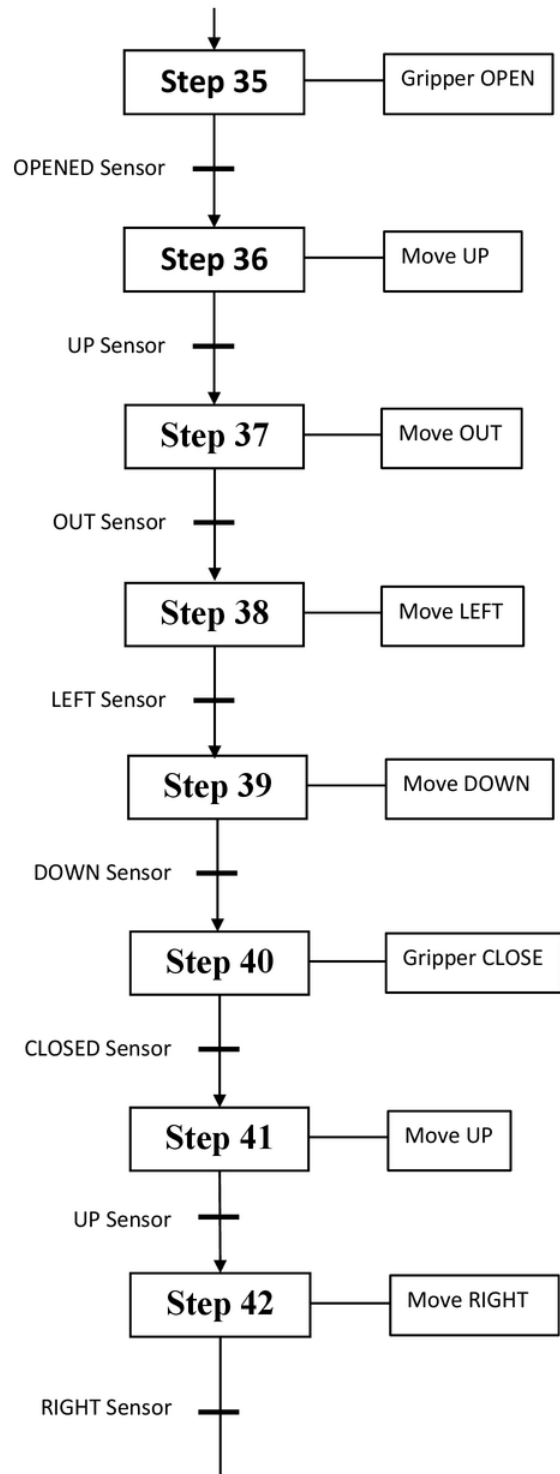
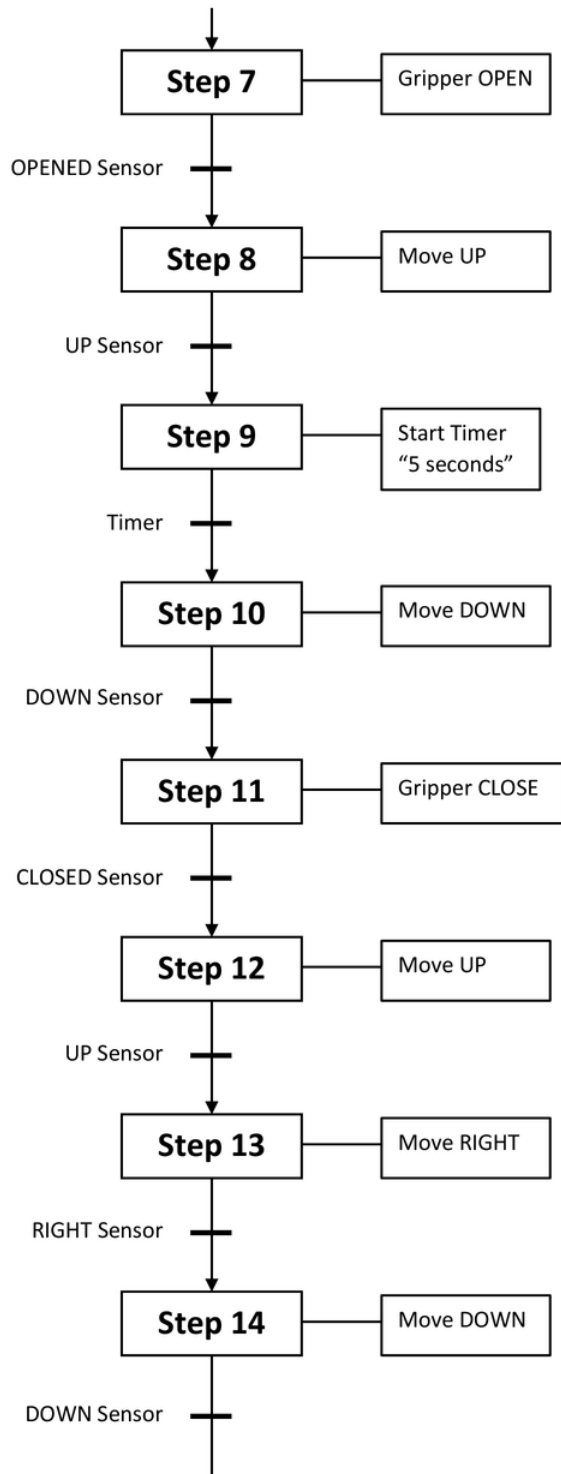


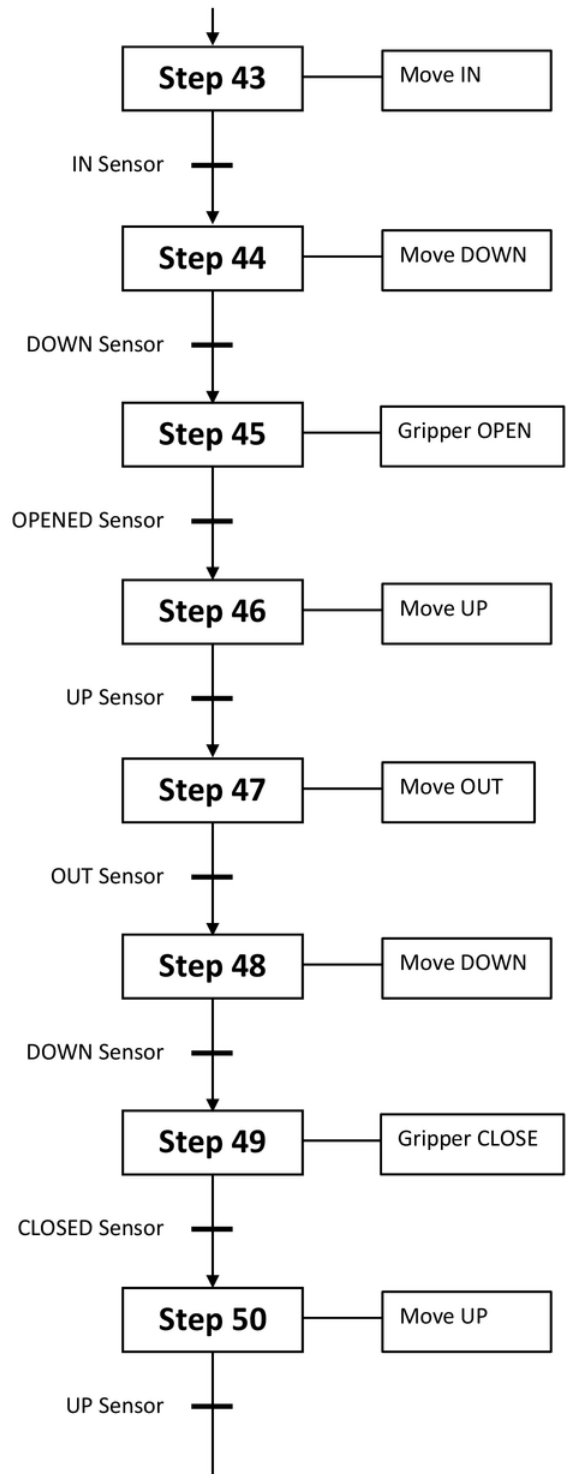
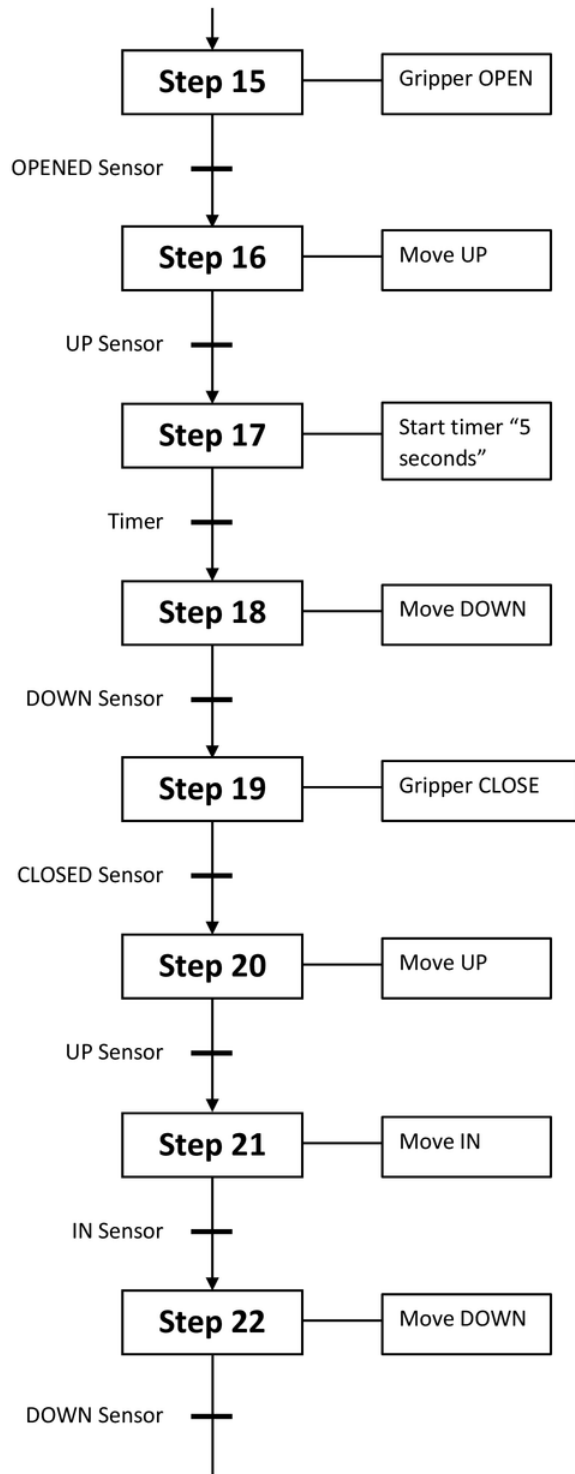
a helpful wiring diagram containing lots of useful

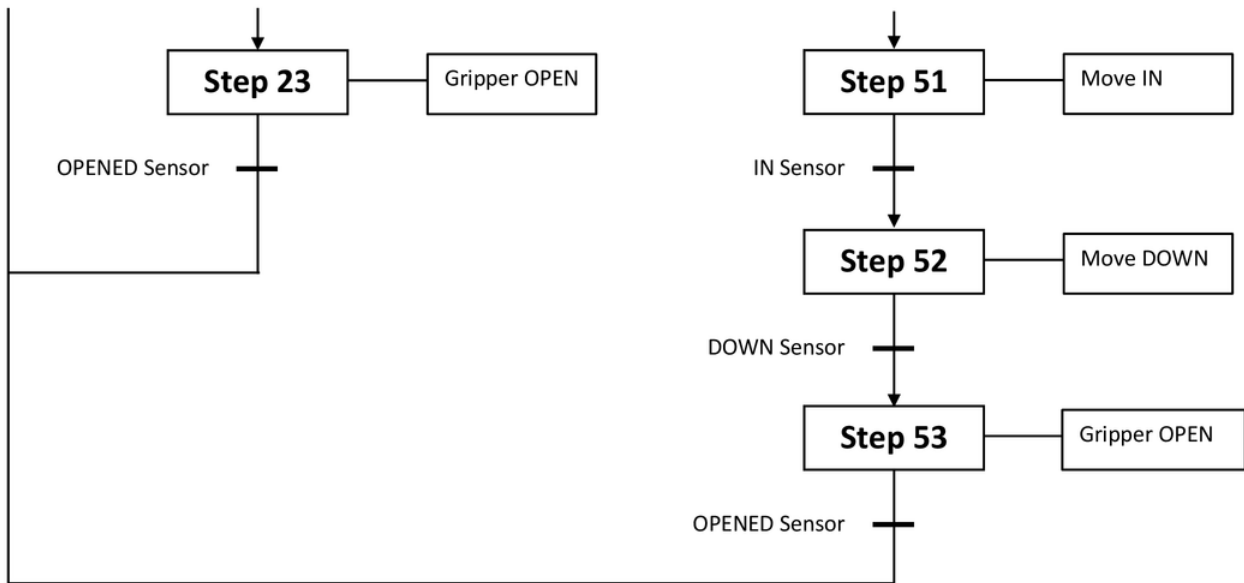
2 - Program Documentation and Explanation

SFC:









Program explanation

Main:

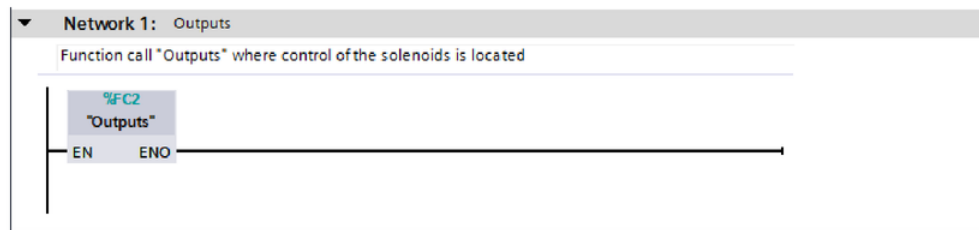


Figure 10 - Main: Network 1 "Outputs"

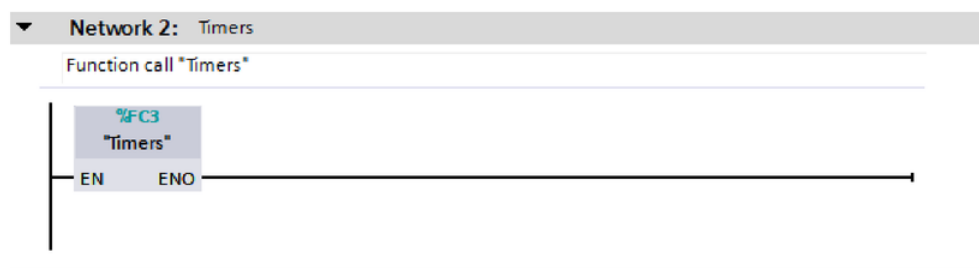


Figure 11 - Main: Network 2 "Timers"

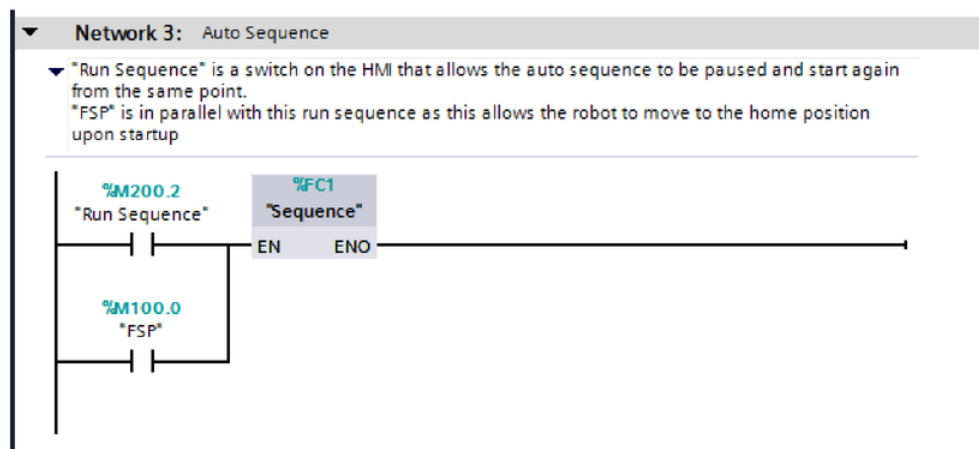


Figure 12 - Main: Network 3 "Sequence"

The network displayed in Figure 12 shows that the "sequence" function call is brought into main and executed under two conditions, when the run sequence is true or FSP is true. The run sequence is

controlled by a button on the HMI which allows the SCL programme in Figure 24 - Figure 41, to be paused and resumed from the same point as specified by the brief. The conditions for the FSP can be seen in Figure 14, this allows the sequences function all to be executed immediately upon start up, if this parallel FSP contact was not present then during the first scan the "run sequence" tag might not immediately be scanned and thus might affect operation for the sequence FC.

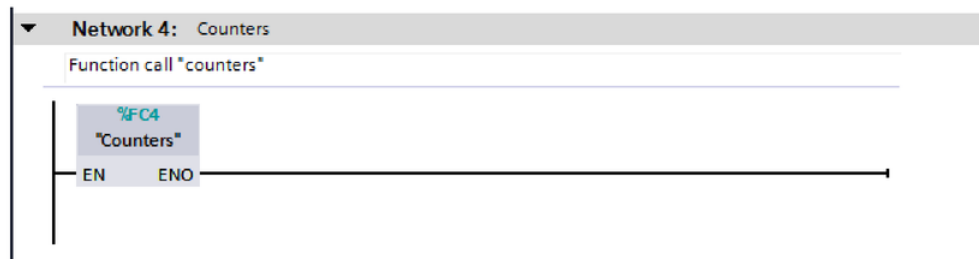


Figure 13 - Main: Network 4 "Counters"

The networks displayed in; Figure 10, Figure 11, Figure 12, and Figure 13 are all situated in the "Main" block OB1. This is the block which the PLC scans and executes. Function calls and data blocks can then be called into main by placing them in a network in main, and are then executed as the PLC scans. If a function call is not called into main then it is not executed. Essentially all of the aforementioned figures display the function calls being brought into "Main" so that they are executed.

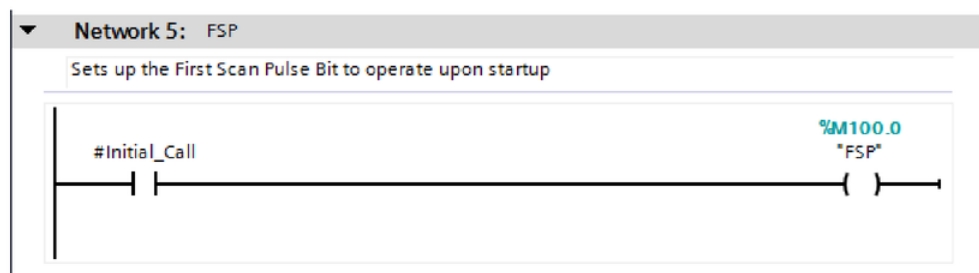


Figure 14 - Main: Network 5 "FSP"

The network in Figure 14 activates the FSP memory bit when the "#Initial_Call" executes upon start-up. That means that the FSP is only true for the first scan of the program and then turns off.

Outputs:

The networks in the function call “outputs” (Figure 15 - Figure 22) control the solenoids of the robot. They feature auto and manual control as required by the brief.

In the auto position the solenoids are controlled by a request from the SCL from the “Sequence” function call, and when not in auto (manual) the solenoids are controlled by a manual request from a HMI button on the engineering screen displayed in Figure 48. Upon reaching the sensor for the matching solenoid (i.e. DOWN sensor and DOWN solenoid) the solenoid de-energises as it has reached its limit, and prevents it from being energised if already at its limit.

The solenoids are not directly driven by the SCL or the manual buttons as this would circumvent any interlocks or safety features. By using a request we can then control the conditions of when these requests are accepted, and when they are denied for safety reasons.

The solenoids don’t need to be latched as they are memory valves

Y-Axis solenoids:

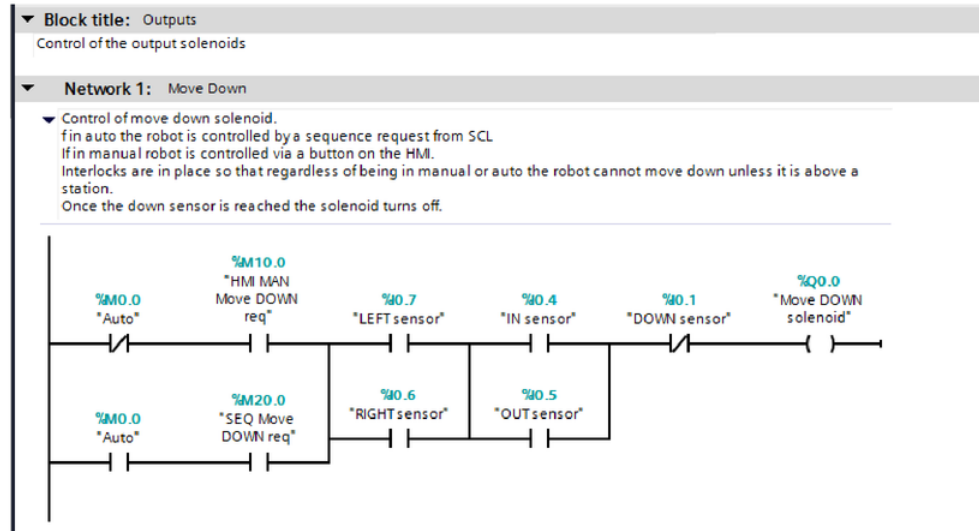


Figure 15 - Outputs: Network 1 "Move Down"

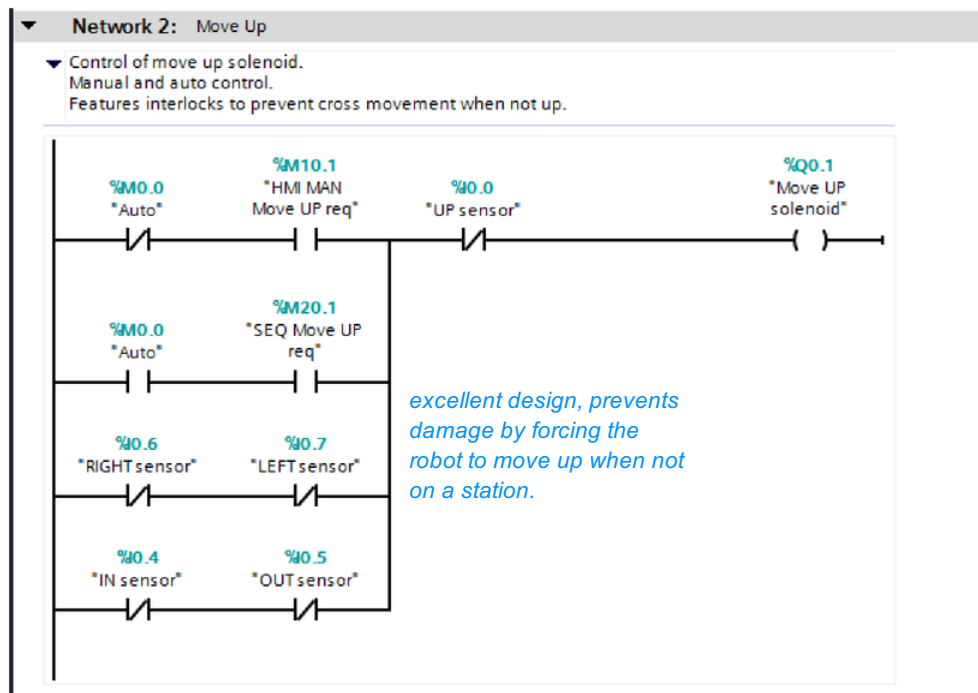


Figure 16 - Outputs: Network 2 "Move Up"

The Y Axis solenoid control illustrated in Figure 15 and Figure 16 have extra safety interlocks.

In Figure 15 the control for the down solenoid can be seen. The H shaped combination of the Left, Right, In, and Out sensor normally open contacts, in series with the auto/manual control, prevent the robot from moving down unless it is directly above one of the 4 stations. This prevents a down request from being accepted whilst the robot is in transit between two stations, ensuring it does not crash into one of the stations.

Whereas the Up solenoid control network in Figure 16 has two paths in parallel with the auto/manual control, each made up of either the X-Axis sensors (Right and Left) or Z-Axis (In and Out) normally closed contacts. These essentially do the opposite of the combination seen in the Figure 15 by forcing the robot to move up if it is not currently in one of the 4 stations, preventing movement in the down state. This acts as another layer of protection working in tandem with the interlock in the down solenoid control to prevent lateral movement in the down position.

Gripper Solenoids:

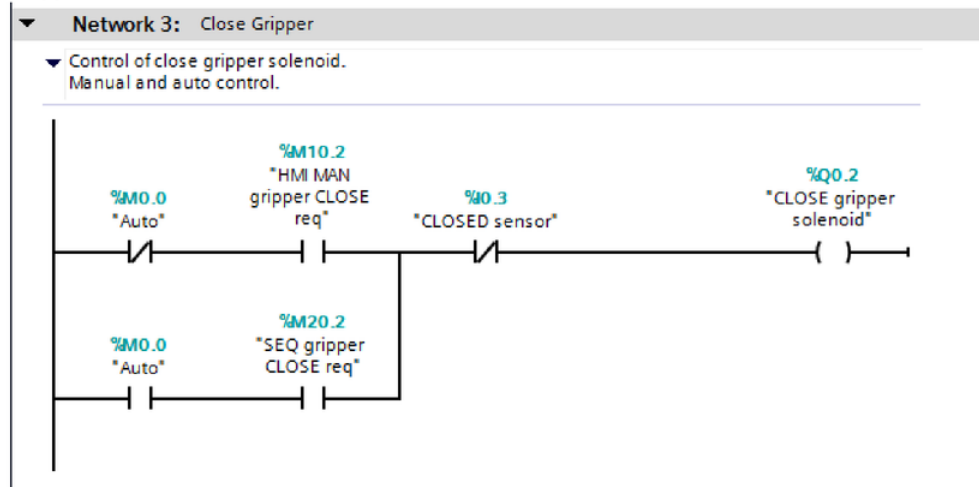


Figure 17 - Outputs: Network 3 "Close Gripper"

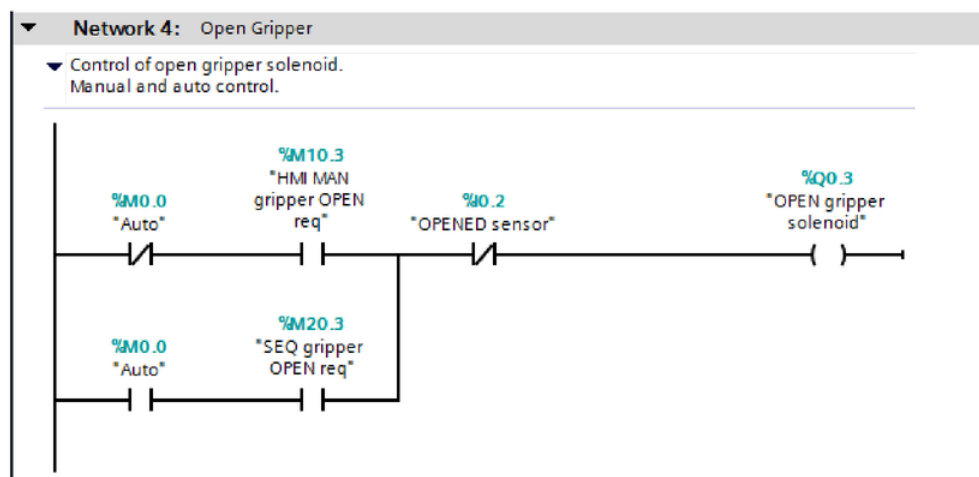


Figure 18 - Outputs: Network 4 "Open Gripper"

The gripper solenoid control networks can be seen in Figure 17 and Figure 18. These feature no interlocks save the sensor pertaining to the solenoid to prevent them from being energised when already at their limit. This is because there is no need to interlock them as the grippers can safely open or close in any state.

Z-Axis Solenoids:

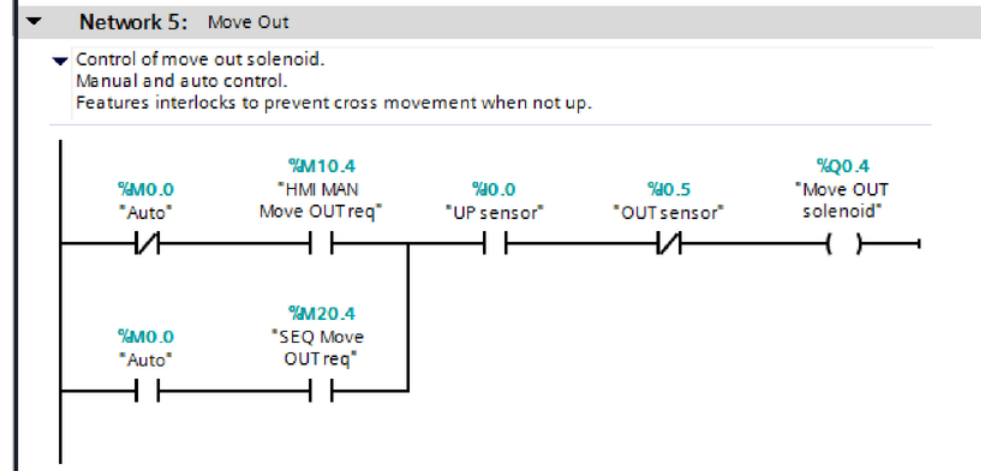


Figure 19 - Outputs: Network 5 "Move Out"

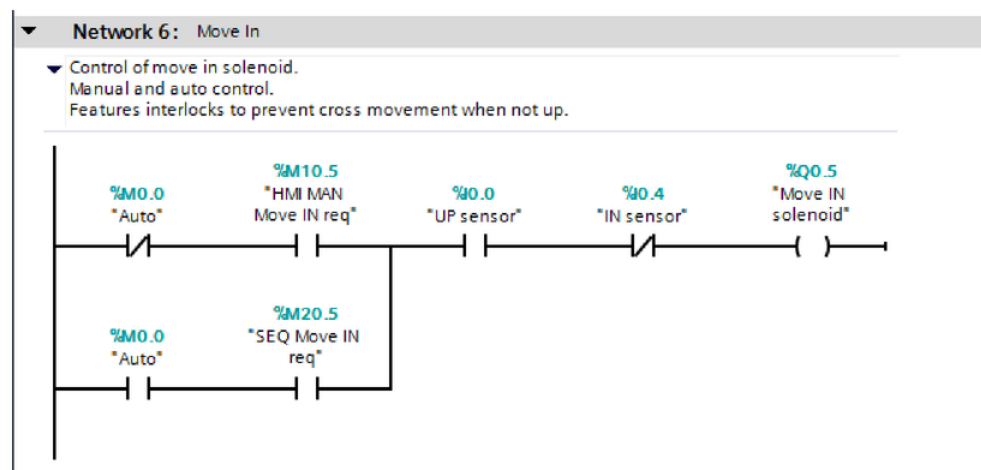


Figure 20 - Outputs: Network 6 "Move In"

Figure 19 and Figure 20 depict the Z-Axis movement solenoids (Out and In respectively). The normally open "Up sensor" contact in series with the coil prevents Z-Axis movement unless the robot is in the up position. This ensures the robot does not damage itself or the product by crashing into a station by moving in the down position.

X-Axis Solenoids:

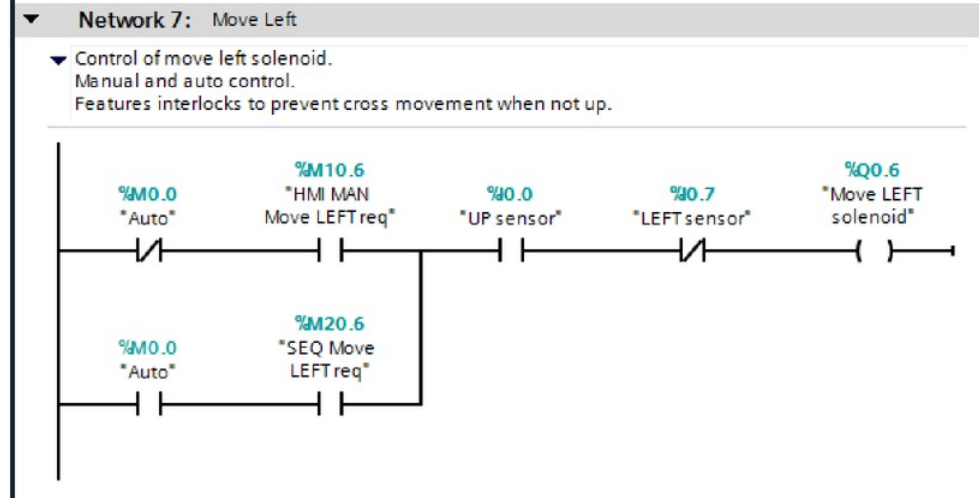


Figure 21 - Outputs: Network 7 "Move Left"

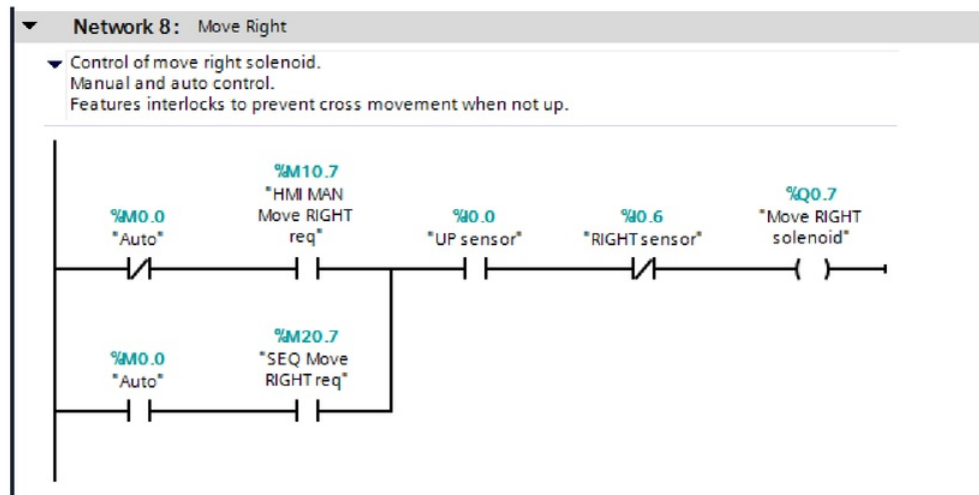


Figure 22 - Outputs: Network 8 "Move Right"

The X-Axis movement networks (Figure 21 & Figure 22) have the exact same structure as the Z-Axis movement networks (Figure 19 & Figure 20), but pertain to the left and right solenoids.

Sequence:

```
1 IF "FSP" THEN           //When the PLC first starts the first scan pulse (FSP) occurs and sets the step to "0", which sends the robot to "home" position
2   "STEP" := 0;
3   "NEXT STEP" := 0;
4 END_IF;
5
6 //All Movements are done via a Sequence Request, as opposed to directly driving the outputs from the sequence
7 //This allows for interlocks and safety features to be implemented in LAD prior to writing sequence to ensure no damage to machine
8 //This can be seen in the function call "Outputs"
9
```

Figure 23 - Sequence: FSP

In Figure 23 the "Step" and "Next Step" memory words are reset to 0 upon startup of the PLC, as FSP is only true on the initial call as seen in Figure 14. Step 0 and 1 put the robot in the home position, illustrated in Figure 24 and Figure 25 respectively, which means that whenever the PLC is booted and the robot is put into auto it will default to the home position.

The reason for not resetting the step count upon entering manual is that only competent and authorised personnel will have access to the engineering screen (where the manual controls are situated) seen in Figure 48, and therefore should be able to put the robot back into the correct position for the step or reset the step by entering a "0" into the step count to take it back home.

```
10 CASE "STEP" OF
11   0: //MOVE UP FOR HOME POSITION
12     "SEQ Move DOWN req" := 0;
13     "SEQ Move UP req" := 1;
14     "SEQ gripper CLOSE req" := 0;
15     "SEQ gripper OPEN req" := 0;
16     "SEQ Move OUT req" := 0;
17     "SEQ Move IN req" := 0;
18     "SEQ Move LEFT req" := 0;
19     "SEQ Move RIGHT req" := 0;
20   IF "UP sensor" = 1 THEN
21     "NEXT STEP" := 1;
22   END_IF;
```

Figure 24 - Sequence: Step 0

The robot is moved up in step 0 of the SCL Programme (Figure 24), and all of the move requests are reset, so that no matter what state the robot was in beforehand in the sequence it will reset previous movement requests. Once the Up sensor has become true denoting that the robot has successfully reached the up position the next step is set to "1". Which as seen in Figure 41 "Step = Next Step", moving the step number ahead one.

```

23
24
25 1: //MOVE IN, LEFT, AND OPEN GRIPPERS FOR HOME POSITION
26 //If "load" option selected and object present then loading sequence will begin
27 //If "Unload" is selected and a pushbutton is pressed then unload sequence will begin
28 "SEQ Move DOWN req" := 0;
29 "SEQ Move UP req" := 0;
30 "SEQ gripper CLOSE req" := 0;
31 "SEQ gripper OPEN req" := 1;
32 "SEQ Move OUT req" := 0;
33 "SEQ Move IN req" := 1;
34 "SEQ Move LEFT req" := 1;
35 "SEQ Move RIGHT req" := 0;
36 IF "OPENED sensor" = 1 AND //Load" seq
37 "IN sensor" = 1 AND
38 "LEFT sensor" = 1 AND
39 "Load" = 1 AND //For "load" seq
40 ("OBJECT sensor" OR //For Load seq
41 "PB X10") THEN //Or" to allow testing of sequence with no object (in brackets otherwise the only thing that would be needed would be PB x10)
42 "NEXT STEP" := 2;
43 END_IF;
44
45 IF "OPENED sensor" = 1 AND //Unload" seq
46 "IN sensor" = 1 AND
47 "LEFT sensor" = 1 AND
48 "Load" = 0 AND //Load = 0 means the program is in unload
49 "Unload Start Button" = 1 THEN //Button Located on HMI
50 "NEXT STEP" := 30;
51 END_IF;

```

Figure 25 - Sequence: Step 1

The SCL code in Figure 25 shows step 1, which moves the robot to the home position by moving in, left and opening the gripper (the robot will already be in the up position from step 0). Then two separate if statements control whether the robot runs through the loading or unloading sequences (both require for it to have reached the home position successfully by activating the relevant sensors in order to continue). The state of the "load" memory bit is altered by a button on the HMI screens as seen in figures Figure 47 & Figure 48. When in load (Load = 1) the loading sequence starts as soon as an object is detected by the object sensor which can be seen in Figure 9, or via the pushbutton "X10" which allows the robots auto load sequence to be tested without the need for a product to be present. Only when all of the prerequisites are met will the robot enter its loading sequence by making "next step" = 2, running through steps 2 - 23 shown in Figure 26 - Figure 33 at which point it will reset to step 0 and cycle through the program again.

The load sequence has the robot pick up an object from station 1, place it in station 2 and wait above it for 5 seconds, pick it up and place it in station 3 and then wait above it for 5 seconds again, and finally place it in station 4 and return to the home position to await another object.

Whereas if the robot is in unload (Load = 0) and the "start unload" button on the operator HMI (Figure 47) , is pressed then the robot will jump to step 30 which is the beginning of the unload sequence (Figure 33) and continue until step 53 (Figure 41) wherein it returns to step 0.

The unload sequence assumes an object is present in all of the stations and will unload station 1, 2, and 3 and place the objects into station 4 in order to empty the machine.

```

52
53     2:      //Move Down
54         "SEQ Move DOWN req" := 1;
55         "SEQ Move UP req" := 0;
56         "SEQ gripper CLOSE req" := 0;
57         "SEQ gripper OPEN req" := 0;
58         "SEQ Move OUT req" := 0;
59         "SEQ Move IN req" := 0;
60         "SEQ Move LEFT req" := 0;
61         "SEQ Move RIGHT req" := 0;
62     IF "DOWN sensor" = 1 THEN
63         "NEXT STEP" := 3;
64     END_IF;
65
66     3:      //Close Grippers
67         "SEQ Move DOWN req" := 0;
68         "SEQ Move UP req" := 0;
69         "SEQ gripper CLOSE req" := 1;
70         "SEQ gripper OPEN req" := 0;
71         "SEQ Move OUT req" := 0;
72         "SEQ Move IN req" := 0;
73         "SEQ Move LEFT req" := 0;
74         "SEQ Move RIGHT req" := 0;
75     IF "CLOSED sensor" = 1 THEN
76         "NEXT STEP" := 4;
77     END_IF;

```

Figure 26 - Sequence: Steps 2 & 3

In order to ensure clarity a note is made next to each step number that describes what action is taken in that step. As an example in step 2 and 3 (Figure 26) the former moves the robot down, and the latter closes the gripper. This means that in step 2 a request for the robot to move down occurs and every other request is reset, and the next step does not occur until the down sensor has been reached.

All of the steps follow this format except with a different action, with the only exceptions being steps 9 & 17 (Figure 28 and Figure 31 respectively) where a timer is incorporated. In these steps the robot must wait above station 3 for step 9 and station 4 for step 17 for 5 seconds. The timers start in their own steps as opposed to being incorporated in the previous steps in order to ensure that the timer does not start until the robot has reached the up position. This prevents the timer from beginning before being fully up meaning that 5 full seconds would not elapse in the up position, and the specification calls for it to be above the station for 5 seconds. The Ladder logic for the timers can be seen in Figure 46.

```

78
79 4:      //Move Up
80      "SEQ Move DOWN req" := 0;
81      "SEQ Move UP req" := 1;
82      "SEQ gripper CLOSE req" := 0;
83      "SEQ gripper OPEN req" := 0;
84      "SEQ Move OUT req" := 0;
85      "SEQ Move IN req" := 0;
86      "SEQ Move LEFT req" := 0;
87      "SEQ Move RIGHT req" := 0;
88 □ IF "UP sensor" = 1 THEN
89      "NEXT STEP" := 5;
90      END_IF;
91
92 5:      //Move Out
93      "SEQ Move DOWN req" := 0;
94      "SEQ Move UP req" := 0;
95      "SEQ gripper CLOSE req" := 0;
96      "SEQ gripper OPEN req" := 0;
97      "SEQ Move OUT req" := 1;
98      "SEQ Move IN req" := 0;
99      "SEQ Move LEFT req" := 0;
100     "SEQ Move RIGHT req" := 0;
101 □ IF "OUT sensor" = 1 THEN
102     "NEXT STEP" := 6;
103     END_IF;
104
105 6:      //Move Down
106      "SEQ Move DOWN req" := 1;
107      "SEQ Move UP req" := 0;
108      "SEQ gripper CLOSE req" := 0;
109      "SEQ gripper OPEN req" := 0;
110      "SEQ Move OUT req" := 0;
111      "SEQ Move IN req" := 0;
112      "SEQ Move LEFT req" := 0;
113      "SEQ Move RIGHT req" := 0;
114 □ IF "DOWN sensor" = 1 THEN
115      "NEXT STEP" := 7;
116      END_IF;

```

Figure 27 - Sequence: Steps 4, 5, 6

```

117 [
118 7:      //Open Grippers
119      "SEQ Move DOWN req" := 0;
120      "SEQ Move UP req" := 0;
121      "SEQ gripper CLOSE req" := 0;
122      "SEQ gripper OPEN req" := 1;
123      "SEQ Move OUT req" := 0;
124      "SEQ Move IN req" := 0;
125      "SEQ Move LEFT req" := 0;
126      "SEQ Move RIGHT req" := 0;
127 □ IF "OPENED sensor" = 1 THEN
128      "NEXT STEP" := 8;
129      END_IF;
130
131 8:      //Move Up
132      "SEQ Move DOWN req" := 0;
133      "SEQ Move UP req" := 1;
134      "SEQ gripper CLOSE req" := 0;
135      "SEQ gripper OPEN req" := 0;
136      "SEQ Move OUT req" := 0;
137      "SEQ Move IN req" := 0;
138      "SEQ Move LEFT req" := 0;
139      "SEQ Move RIGHT req" := 0;
140 □ IF "UP sensor" = 1 THEN
141      "NEXT STEP" := 9;
142      END_IF;
143
144 9:      //Wait 5 Seconds
145      "SEQ Move DOWN req" := 0;
146      "SEQ Move UP req" := 0;
147      "SEQ gripper CLOSE req" := 0;
148      "SEQ gripper OPEN req" := 0;
149      "SEQ Move OUT req" := 0;
150      "SEQ Move IN req" := 0;
151      "SEQ Move LEFT req" := 0;
152      "SEQ Move RIGHT req" := 0;
153 □ IF "5 Second Timer".Q = 1 THEN      //Timer Is activated in Function Call "Timers" Using LAD
154      "NEXT STEP" := 10;
155      END_IF;

```

Figure 28 - Sequence: Steps 7, 8, 9

```

156
157 10:    //Move Down
158     "SEQ Move DOWN req" := 1;
159     "SEQ Move UP req" := 0;
160     "SEQ gripper CLOSE req" := 0;
161     "SEQ gripper OPEN req" := 0;
162     "SEQ Move OUT req" := 0;
163     "SEQ Move IN req" := 0;
164     "SEQ Move LEFT req" := 0;
165     "SEQ Move RIGHT req" := 0;
166     IF "DOWN sensor" = 1 THEN
167         "NEXT STEP" := 11;
168     END_IF;
169
170 11:    //Close Grippers
171     "SEQ Move DOWN req" := 0;
172     "SEQ Move UP req" := 0;
173     "SEQ gripper CLOSE req" := 1;
174     "SEQ gripper OPEN req" := 0;
175     "SEQ Move OUT req" := 0;
176     "SEQ Move IN req" := 0;
177     "SEQ Move LEFT req" := 0;
178     "SEQ Move RIGHT req" := 0;
179     IF "CLOSED sensor" = 1 THEN
180         "NEXT STEP" := 12;
181     END_IF;
182
183 12:    //Move Up
184     "SEQ Move DOWN req" := 0;
185     "SEQ Move UP req" := 1;
186     "SEQ gripper CLOSE req" := 0;
187     "SEQ gripper OPEN req" := 0;
188     "SEQ Move OUT req" := 0;
189     "SEQ Move IN req" := 0;
190     "SEQ Move LEFT req" := 0;
191     "SEQ Move RIGHT req" := 0;
192     IF "UP sensor" = 1 THEN
193         "NEXT STEP" := 13;
194     END_IF;
195

```

Figure 29 - Sequence: Steps 10, 11, 12

```

195
196 13:      //Move Right
197      "SEQ Move DOWN req" := 0;
198      "SEQ Move UP req" := 0;
199      "SEQ gripper CLOSE req" := 0;
200      "SEQ gripper OPEN req" := 0;
201      "SEQ Move OUT req" := 0;
202      "SEQ Move IN req" := 0;
203      "SEQ Move LEFT req" := 0;
204      "SEQ Move RIGHT req" := 1;
205  IF "RIGHT sensor" = 1 THEN
206      "NEXT STEP" := 14;
207  END_IF;
208
209 14:      //Move Down
210      "SEQ Move DOWN req" := 1;
211      "SEQ Move UP req" := 0;
212      "SEQ gripper CLOSE req" := 0;
213      "SEQ gripper OPEN req" := 0;
214      "SEQ Move OUT req" := 0;
215      "SEQ Move IN req" := 0;
216      "SEQ Move LEFT req" := 0;
217      "SEQ Move RIGHT req" := 0;
218  IF "DOWN sensor" = 1 THEN
219      "NEXT STEP" := 15;
220  END_IF;
221
222 15:      //Open Grippers
223      "SEQ Move DOWN req" := 0;
224      "SEQ Move UP req" := 0;
225      "SEQ gripper CLOSE req" := 0;
226      "SEQ gripper OPEN req" := 1;
227      "SEQ Move OUT req" := 0;
228      "SEQ Move IN req" := 0;
229      "SEQ Move LEFT req" := 0;
230      "SEQ Move RIGHT req" := 0;
231  IF "OPENED sensor" = 1 THEN
232      "NEXT STEP" := 16;
233  END_IF;

```

Figure 30 - Sequence: Steps 13, 14, 15

```

234
235 16:    //Move Up
236     "SEQ Move DOWN req" := 0;
237     "SEQ Move UP req" := 1;
238     "SEQ gripper CLOSE req" := 0;
239     "SEQ gripper OPEN req" := 0;
240     "SEQ Move OUT req" := 0;
241     "SEQ Move IN req" := 0;
242     "SEQ Move LEFT req" := 0;
243     "SEQ Move RIGHT req" := 0;
244     IF "UP sensor" = 1 THEN
245         "NEXT STEP" := 17;
246     END_IF;
247
248 17:    //Wait 5 Seconds
249     "SEQ Move DOWN req" := 0;
250     "SEQ Move UP req" := 0;
251     "SEQ gripper CLOSE req" := 0;
252     "SEQ gripper OPEN req" := 0;
253     "SEQ Move OUT req" := 0;
254     "SEQ Move IN req" := 0;
255     "SEQ Move LEFT req" := 0;
256     "SEQ Move RIGHT req" := 0;
257     IF "5 Second Timer".Q = 1 THEN    //Timer Is activated in Function Call "Timers" Using LAD
258         "NEXT STEP" := 18;
259     END_IF;
260
261 18:    //Move Down
262     "SEQ Move DOWN req" := 1;
263     "SEQ Move UP req" := 0;
264     "SEQ gripper CLOSE req" := 0;
265     "SEQ gripper OPEN req" := 0;
266     "SEQ Move OUT req" := 0;
267     "SEQ Move IN req" := 0;
268     "SEQ Move LEFT req" := 0;
269     "SEQ Move RIGHT req" := 0;
270     IF "DOWN sensor" = 1 THEN
271         "NEXT STEP" := 19;
272     END_IF;

```

Figure 31 - Sequence: Steps 16, 17, 18

```

273
274 19:    //Close Grippers
275     "SEQ Move DOWN req" := 0;
276     "SEQ Move UP req" := 0;
277     "SEQ gripper CLOSE req" := 1;
278     "SEQ gripper OPEN req" := 0;
279     "SEQ Move OUT req" := 0;
280     "SEQ Move IN req" := 0;
281     "SEQ Move LEFT req" := 0;
282     "SEQ Move RIGHT req" := 0;
283 IF "CLOSED sensor" = 1 THEN
284     "NEXT STEP" := 20;
285 END_IF;
286
287 20:    //Move Up
288     "SEQ Move DOWN req" := 0;
289     "SEQ Move UP req" := 1;
290     "SEQ gripper CLOSE req" := 0;
291     "SEQ gripper OPEN req" := 0;
292     "SEQ Move OUT req" := 0;
293     "SEQ Move IN req" := 0;
294     "SEQ Move LEFT req" := 0;
295     "SEQ Move RIGHT req" := 0;
296 IF "UP sensor" = 1 THEN
297     "NEXT STEP" := 21;
298 END_IF;
299
300 21:    //Move In
301     "SEQ Move DOWN req" := 0;
302     "SEQ Move UP req" := 0;
303     "SEQ gripper CLOSE req" := 0;
304     "SEQ gripper OPEN req" := 0;
305     "SEQ Move OUT req" := 0;
306     "SEQ Move IN req" := 1;
307     "SEQ Move LEFT req" := 0;
308     "SEQ Move RIGHT req" := 0;
309 IF "IN sensor" = 1 THEN
310     "NEXT STEP" := 22;
311 END_IF;

```

Figure 32 - Sequence: Steps 19, 20, 21

```

312
313 22: //Move Down
314 "SEQ Move DOWN req" := 1;
315 "SEQ Move UP req" := 0;
316 "SEQ gripper CLOSE req" := 0;
317 "SEQ gripper OPEN req" := 0;
318 "SEQ Move OUT req" := 0;
319 "SEQ Move IN req" := 0;
320 "SEQ Move LEFT req" := 0;
321 "SEQ Move RIGHT req" := 0;
322 IF "DOWN sensor" = 1 THEN
323 "NEXT STEP" := 23;
324 END_IF;
325
326 23: //Gripper Open
327 "SEQ Move DOWN req" := 0;
328 "SEQ Move UP req" := 0;
329 "SEQ gripper CLOSE req" := 0;
330 "SEQ gripper OPEN req" := 1;
331 "SEQ Move OUT req" := 0;
332 "SEQ Move IN req" := 0;
333 "SEQ Move LEFT req" := 0;
334 "SEQ Move RIGHT req" := 0;
335 IF "OPENED sensor" = 1 THEN
336 "NEXT STEP" := 0; //Once the object has been placed in station 4 the SCL will load back to "step 0" returning home
337 END_IF;
338
339
340 // "Unload" seq
341 30: //Move Down
342 "SEQ Move DOWN req" := 1;
343 "SEQ Move UP req" := 0;
344 "SEQ gripper CLOSE req" := 0;
345 "SEQ gripper OPEN req" := 0;
346 "SEQ Move OUT req" := 0;
347 "SEQ Move IN req" := 0;
348 "SEQ Move LEFT req" := 0;
349 "SEQ Move RIGHT req" := 0;
350 IF "DOWN sensor" = 1 THEN
351 "NEXT STEP" := 31;
352 END_IF;

```

Figure 33 - Sequence: Steps 22, 23, 30

```

353
354 31:      //Gripper Close
355      "SEQ Move DOWN req" := 0;
356      "SEQ Move UP req" := 0;
357      "SEQ gripper CLOSE req" := 1;
358      "SEQ gripper OPEN req" := 0;
359      "SEQ Move OUT req" := 0;
360      "SEQ Move IN req" := 0;
361      "SEQ Move LEFT req" := 0;
362      "SEQ Move RIGHT req" := 0;
363      IF "CLOSED sensor" = 1 THEN
364          "NEXT STEP" := 32;
365      END_IF;
366
367 32:      //Move Up
368      "SEQ Move DOWN req" := 0;
369      "SEQ Move UP req" := 01;
370      "SEQ gripper CLOSE req" := 0;
371      "SEQ gripper OPEN req" := 0;
372      "SEQ Move OUT req" := 0;
373      "SEQ Move IN req" := 0;
374      "SEQ Move LEFT req" := 0;
375      "SEQ Move RIGHT req" := 0;
376      IF "UP sensor" = 1 THEN
377          "NEXT STEP" := 33;
378      END_IF;
379
380 33:      //Move Right
381      "SEQ Move DOWN req" := 0;
382      "SEQ Move UP req" := 0;
383      "SEQ gripper CLOSE req" := 0;
384      "SEQ gripper OPEN req" := 0;
385      "SEQ Move OUT req" := 0;
386      "SEQ Move IN req" := 0;
387      "SEQ Move LEFT req" := 0;
388      "SEQ Move RIGHT req" := 1;
389      IF "RIGHT sensor" = 1 THEN
390          "NEXT STEP" := 34;
391      END_IF;
---
```

Figure 34 - Sequence: Steps 31, 32, 33

```

392      -
393      34:      //Move Down
394      "SEQ Move DOWN req" := 1;
395      "SEQ Move UP req" := 0;
396      "SEQ gripper CLOSE req" := 0;
397      "SEQ gripper OPEN req" := 0;
398      "SEQ Move OUT req" := 0;
399      "SEQ Move IN req" := 0;
400      "SEQ Move LEFT req" := 0;
401      "SEQ Move RIGHT req" := 0;
402      IF "DOWN sensor" = 1 THEN
403          "NEXT STEP" := 35;
404      END_IF;
405
406      35:      //Gripper Open
407      "SEQ Move DOWN req" := 0;
408      "SEQ Move UP req" := 0;
409      "SEQ gripper CLOSE req" := 0;
410      "SEQ gripper OPEN req" := 1;
411      "SEQ Move OUT req" := 0;
412      "SEQ Move IN req" := 0;
413      "SEQ Move LEFT req" := 0;
414      "SEQ Move RIGHT req" := 0;
415      IF "OPENED sensor" = 1 THEN
416          "NEXT STEP" := 36;
417      END_IF;
418
419      36:      //Move Up
420      "SEQ Move DOWN req" := 0;
421      "SEQ Move UP req" := 1;
422      "SEQ gripper CLOSE req" := 0;
423      "SEQ gripper OPEN req" := 0;
424      "SEQ Move OUT req" := 0;
425      "SEQ Move IN req" := 0;
426      "SEQ Move LEFT req" := 0;
427      "SEQ Move RIGHT req" := 0;
428      IF "UP sensor" = 1 THEN
429          "NEXT STEP" := 37;
430      END_IF;

```

Figure 35- Sequence: Steps 34, 35, 36

```

431
432      37:      //Move Out
433      "SEQ Move DOWN req" := 0;
434      "SEQ Move UP req" := 0;
435      "SEQ gripper CLOSE req" := 0;
436      "SEQ gripper OPEN req" := 0;
437      "SEQ Move OUT req" := 1;
438      "SEQ Move IN req" := 0;
439      "SEQ Move LEFT req" := 0;
440      "SEQ Move RIGHT req" := 0;
441      IF "OUT sensor" = 1 THEN
442          "NEXT STEP" := 38;
443      END_IF;
444
445      38:      //Move Left
446      "SEQ Move DOWN req" := 0;
447      "SEQ Move UP req" := 0;
448      "SEQ gripper CLOSE req" := 0;
449      "SEQ gripper OPEN req" := 0;
450      "SEQ Move OUT req" := 0;
451      "SEQ Move IN req" := 0;
452      "SEQ Move LEFT req" := 1;
453      "SEQ Move RIGHT req" := 0;
454      IF "LEFT sensor" = 1 THEN
455          "NEXT STEP" := 39;
456      END_IF;
457
458      39:      //Move Down
459      "SEQ Move DOWN req" := 1;
460      "SEQ Move UP req" := 0;
461      "SEQ gripper CLOSE req" := 0;
462      "SEQ gripper OPEN req" := 0;
463      "SEQ Move OUT req" := 0;
464      "SEQ Move IN req" := 0;
465      "SEQ Move LEFT req" := 0;
466      "SEQ Move RIGHT req" := 0;
467      IF "DOWN sensor" = 1 THEN
468          "NEXT STEP" := 40;
469      END_IF;

```

Figure 36 - Sequence: Steps 37, 38, 39

```

470
471      40:      //Close Grippers
472      "SEQ Move DOWN req" := 0;
473      "SEQ Move UP req" := 0;
474      "SEQ gripper CLOSE req" := 1;
475      "SEQ gripper OPEN req" := 0;
476      "SEQ Move OUT req" := 0;
477      "SEQ Move IN req" := 0;
478      "SEQ Move LEFT req" := 0;
479      "SEQ Move RIGHT req" := 0;
480      IF "CLOSED sensor" = 1 THEN
481          "NEXT STEP" := 41;
482      END_IF;
483
484      41:      //Move Up
485      "SEQ Move DOWN req" := 0;
486      "SEQ Move UP req" := 1;
487      "SEQ gripper CLOSE req" := 0;
488      "SEQ gripper OPEN req" := 0;
489      "SEQ Move OUT req" := 0;
490      "SEQ Move IN req" := 0;
491      "SEQ Move LEFT req" := 0;
492      "SEQ Move RIGHT req" := 0;
493      IF "UP sensor" = 1 THEN
494          "NEXT STEP" := 42;
495      END_IF;
496
497      42:      //Move Right
498      "SEQ Move DOWN req" := 0;
499      "SEQ Move UP req" := 0;
500      "SEQ gripper CLOSE req" := 0;
501      "SEQ gripper OPEN req" := 0;
502      "SEQ Move OUT req" := 0;
503      "SEQ Move IN req" := 0;
504      "SEQ Move LEFT req" := 0;
505      "SEQ Move RIGHT req" := 1;
506      IF "RIGHT sensor" = 1 THEN
507          "NEXT STEP" := 43;
508      END_IF;

```

Figure 37- Sequence: Steps 40, 41, 42

```

509
510      43:      //Move In
511      "SEQ Move DOWN req" := 0;
512      "SEQ Move UP req" := 0;
513      "SEQ gripper CLOSE req" := 0;
514      "SEQ gripper OPEN req" := 0;
515      "SEQ Move OUT req" := 0;
516      "SEQ Move IN req" := 1;
517      "SEQ Move LEFT req" := 0;
518      "SEQ Move RIGHT req" := 0;
519      IF "IN sensor" = 1 THEN
520          "NEXT STEP" := 44;
521      END_IF;
522
523      44:      //Move Down
524      "SEQ Move DOWN req" := 1;
525      "SEQ Move UP req" := 0;
526      "SEQ gripper CLOSE req" := 0;
527      "SEQ gripper OPEN req" := 0;
528      "SEQ Move OUT req" := 0;
529      "SEQ Move IN req" := 0;
530      "SEQ Move LEFT req" := 0;
531      "SEQ Move RIGHT req" := 0;
532      IF "DOWN sensor" = 1 THEN
533          "NEXT STEP" := 45;
534      END_IF;
535
536      45:      //Gripper Open
537      "SEQ Move DOWN req" := 0;
538      "SEQ Move UP req" := 0;
539      "SEQ gripper CLOSE req" := 0;
540      "SEQ gripper OPEN req" := 1;
541      "SEQ Move OUT req" := 0;
542      "SEQ Move IN req" := 0;
543      "SEQ Move LEFT req" := 0;
544      "SEQ Move RIGHT req" := 0;
545      IF "OPENED sensor" = 1 THEN
546          "NEXT STEP" := 46;
547      END_IF;

```

Figure 38- Sequence: Steps 43, 44, 45

```

548
549      46:      //Move Up
550      "SEQ Move DOWN req" := 0;
551      "SEQ Move UP req" := 1;
552      "SEQ gripper CLOSE req" := 0;
553      "SEQ gripper OPEN req" := 0;
554      "SEQ Move OUT req" := 0;
555      "SEQ Move IN req" := 0;
556      "SEQ Move LEFT req" := 0;
557      "SEQ Move RIGHT req" := 0;
558      IF "UP sensor" = 1 THEN
559          "NEXT STEP" := 47;
560      END_IF;
561
562      47:      //Move Out
563      "SEQ Move DOWN req" := 0;
564      "SEQ Move UP req" := 0;
565      "SEQ gripper CLOSE req" := 0;
566      "SEQ gripper OPEN req" := 0;
567      "SEQ Move OUT req" := 1;
568      "SEQ Move IN req" := 0;
569      "SEQ Move LEFT req" := 0;
570      "SEQ Move RIGHT req" := 0;
571      IF "OUT sensor" = 1 THEN
572          "NEXT STEP" := 48;
573      END_IF;
574
575      48:      //Move Down
576      "SEQ Move DOWN req" := 1;
577      "SEQ Move UP req" := 0;
578      "SEQ gripper CLOSE req" := 0;
579      "SEQ gripper OPEN req" := 0;
580      "SEQ Move OUT req" := 0;
581      "SEQ Move IN req" := 0;
582      "SEQ Move LEFT req" := 0;
583      "SEQ Move RIGHT req" := 0;
584      IF "DOWN sensor" = 1 THEN
585          "NEXT STEP" := 49;
586      END_IF;

```

Figure 39- Sequence: Steps 46, 47, 48

```

587
588      49:      //Gripper Close
589      "SEQ Move DOWN req" := 0;
590      "SEQ Move UP req" := 0;
591      "SEQ gripper CLOSE req" := 1;
592      "SEQ gripper OPEN req" := 0;
593      "SEQ Move OUT req" := 0;
594      "SEQ Move IN req" := 0;
595      "SEQ Move LEFT req" := 0;
596      "SEQ Move RIGHT req" := 0;
597      IF "CLOSED sensor" = 1 THEN
598          "NEXT STEP" := 50;
599      END_IF;
600
601      50:      //Move Up
602      "SEQ Move DOWN req" := 0;
603      "SEQ Move UP req" := 1;
604      "SEQ gripper CLOSE req" := 0;
605      "SEQ gripper OPEN req" := 0;
606      "SEQ Move OUT req" := 0;
607      "SEQ Move IN req" := 0;
608      "SEQ Move LEFT req" := 0;
609      "SEQ Move RIGHT req" := 0;
610      IF "UP sensor" = 1 THEN
611          "NEXT STEP" := 51;
612      END_IF;
613
614      51:      //Move In
615      "SEQ Move DOWN req" := 0;
616      "SEQ Move UP req" := 0;
617      "SEQ gripper CLOSE req" := 0;
618      "SEQ gripper OPEN req" := 0;
619      "SEQ Move OUT req" := 0;
620      "SEQ Move IN req" := 1;
621      "SEQ Move LEFT req" := 0;
622      "SEQ Move RIGHT req" := 0;
623      IF "IN sensor" = 1 THEN
624          "NEXT STEP" := 52;
625      END_IF;

```

Figure 40- Sequence: Steps 49, 50, 51

```

626
627
628      52:      //Move Down
629      "SEQ Move DOWN req" := 1;
630      "SEQ Move UP req" := 0;
631      "SEQ gripper CLOSE req" := 0;
632      "SEQ gripper OPEN req" := 0;
633      "SEQ Move OUT req" := 0;
634      "SEQ Move IN req" := 0;
635      "SEQ Move LEFT req" := 0;
636      "SEQ Move RIGHT req" := 0;
637      IF "DOWN sensor" = 1 THEN
638          "NEXT STEP" := 53;
639      END_IF;
640
641      53:      //Gripper Open
642      "SEQ Move DOWN req" := 0;
643      "SEQ Move UP req" := 0;
644      "SEQ gripper CLOSE req" := 0;
645      "SEQ gripper OPEN req" := 1;
646      "SEQ Move OUT req" := 0;
647      "SEQ Move IN req" := 0;
648      "SEQ Move LEFT req" := 0;
649      "SEQ Move RIGHT req" := 0;
650      IF "OPENED sensor" = 1 THEN
651          "NEXT STEP" := 0;
652      END_IF;
653  END_CASE;
654  "STEP" := "NEXT STEP";      //Linking "Step" to "Next Step"

```

Figure 41- Sequence: Steps 52, 53, & "Step := Next Step"

Counters:

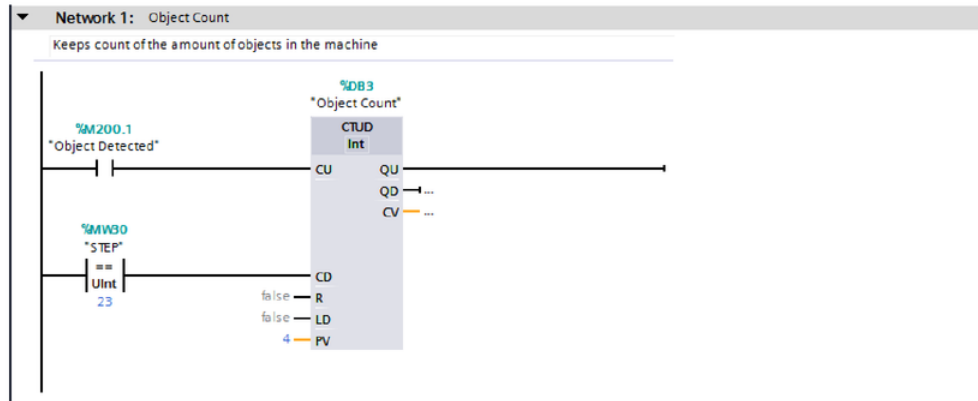


Figure 42 - Counters: Network 1 "Object Count"

The network displayed in Figure 42 is responsible for the object count, which is the amount of objects currently in the machine. This count increases whenever the object detected bit goes high (the state of object detected is controlled by the network in Figure 43), and counts down when step 23 occurs which is the final step in the automatic load sequence (Figure 33) as this is when the object is placed in station 4 and exits the machine.

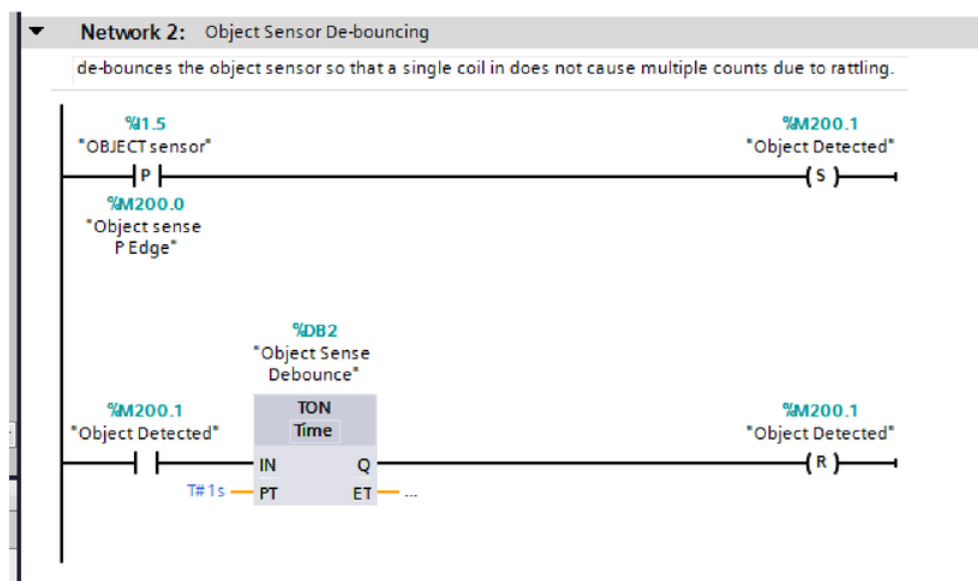


Figure 43 - Counters: Network 2 "Object Sensor De-Bouncing"

When an object rolls in to station 1 it can rattle due to being brought to a stop suddenly, which occurs in front of the object sensor. This rattling can cause the object sensor to be triggered multiple

times, causing an artificially inflated count. To prevent this a de-bouncer is utilised which can be seen in Figure 43.

On the positive edge of an object being sensed the object detected bit is set which latches the bit until it is reset. The object detected bit will remain set until the de-bounce timer has elapsed and then it is reset. This effectively masks the multiple triggers of the object sensor by taking the positive edge of the first trigger and then latching it on for a second, which prevents it from triggering any more times in that 1 second period, which should be ample time for the object to finish rattling. An illustration of this can be seen in Figure 44.

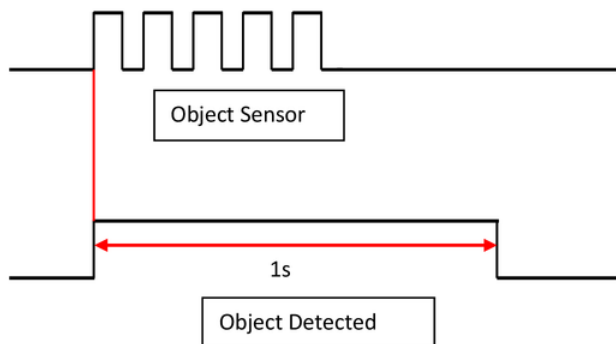


Figure 44 - De-Bouncing Illustration

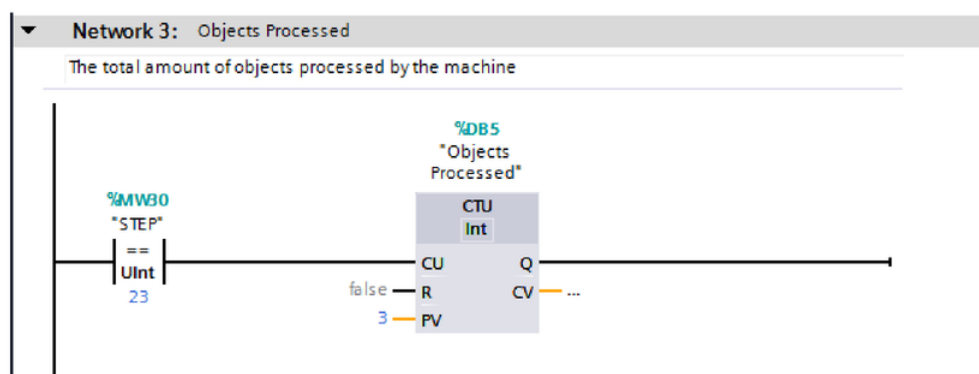


Figure 45 - Counters: Network 3 "Objects Processed"

The count of total objects processed by the machine in the load sequence is controlled by the network depicted in Figure 45.

Every time step 23 (the final step of the load sequence in Figure 33) occurs the count increases, as this is when an object exits the system after being processed.

The PV is set to 3 so that an alarm can come up on the operator HMI (Figure 47) that the next time the system stops it should undergo a maintenance. In reality this would be a much higher number of operations, but it has been set low so that it can be easily tested.

Timers:

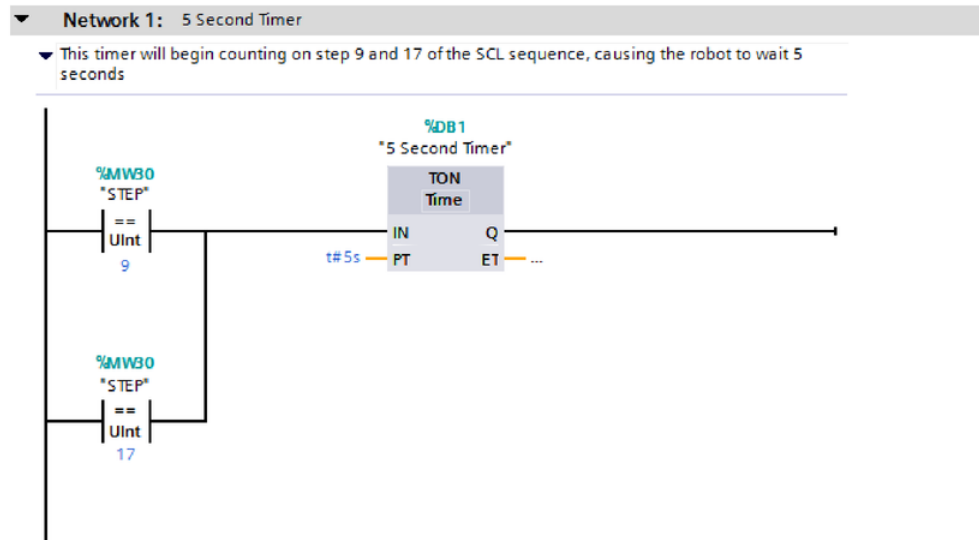


Figure 46 - Timers: Network 1 "5 Second Timer"

The 5 second timer shown in Figure 46 is responsible for the 5 second pause that occurs in step 9 & 17 of the auto load sequence (Figure 28 and Figure 31). The PT of the timer has been set to 5s which means that once a signal is detected in the "In" input of the timer then after 5s has elapsed the Q becomes true. The "In" input only becomes true when "Step" is equal to either 9 or 17.

HMI SCREENS:

Operator is "root screen" so that the machine boots on that screen, preventing accidental access to the engineering screen.

Operator Screen:

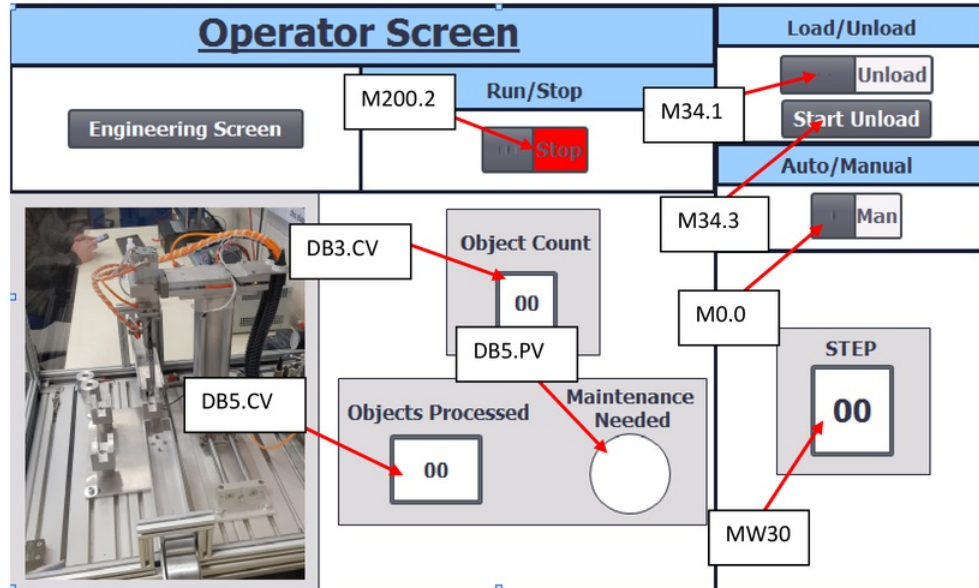


Figure 47 - HMI: Operator Screen

The screen displayed in Figure 47 is the screen that is available to operators as well as to engineering, and is the root screen, which means that upon start-up the HMO defaults to this screen. This prevents accidental access to the engineering screen (Figure 48) and the manual controls of the robot by unauthorised people.

Engineering Screen:

Engineering Screen			
Operator Screen STEP 00		Run/Stop Run Stop	
Sensors		Solenoids	
Object Open Right Close Left Down Out Up In		Open Right Close Left Down Out Up In	
		Load/Unload Unload Start Unload	
		Auto/Manual Auto Man	
		Manual Controls Up Down In Out Left Right Open Close	

Figure 48 - HMI: Engineering Screen

The engineering screen, Figure 48, is only accessible to authorised personnel. This is due to it containing manual controls and diagnostic tools which are helpful to engineering craft and should only be used by competent people in order to ensure safety.

The manual controls are addressed to the HMI man requests which appear in the “Outputs” function call, Figure 15 - Figure 22.

The solenoid indication lights are addressed to the solenoids themselves, and the sensor indications to the reed switch sensors, all of these addresses can be seen in the I/O list in Table 1.

To that end the screen has been password protected, so that only authorised people can access it. When the “engineering screen” button displayed on the operator screen (Figure 47) is pressed then a prompt appears to input username and password (Figure 49). Once the correct credentials have been entered then access to the engineering screen is permitted.

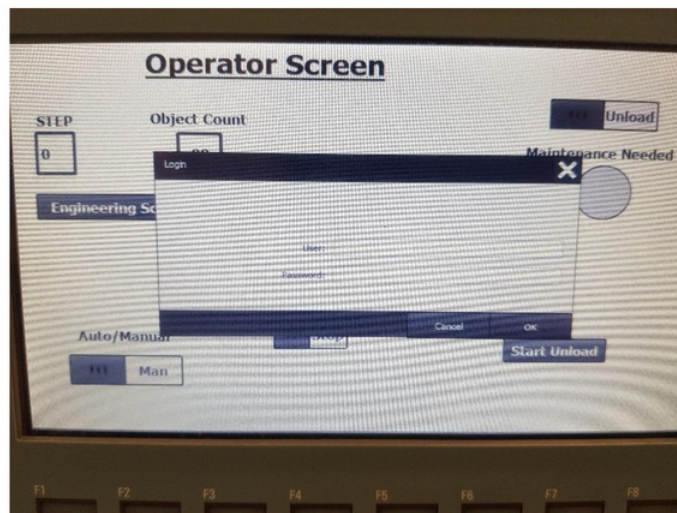


Figure 49 - HMI: Login prompt

The credentials for the engineering screen can be altered under the users tab, and by changing the information in the relevant user boxes. The ones used can be seen in Figure 50.

Users						
Name	Password	Automatic logoff	Logoff time	Number	Comment	
Administrator	*****	☒	5	1	The user 'Administrator' is as..	

Figure 50 HMI: Administrator credentials

Crucially as soon as the user exits the engineering screen and returns to the operator screen the user is logged out, removing engineering screen privileges and preventing the user from re-entering the engineering screen without inputting the username and password. This is achieved through the button event tied to the "Operator Screen" button on the Engineering HMI, the event can be seen in Figure 51.

Properties	Animations	Events	Texts
Click Press Release Activate Deactivate Change			
ActivateScreen Screen name Object number Logoff <Add function>			
			Operator Screen
			0

Figure 51 - HMI: logoff event

PLC Tags:

	Name	Tag table	Data type	Address	Retain	Acces...	Writa...	Visibl...	Comment
1	FSP	Default tag table	Bool	%M100.0	☐	☒	☒	☒	
2	UP sensor	Default tag table	Bool	%I0.0	☐	☒	☒	☒	
3	DOWN sensor	Default tag table	Bool	%I0.1	☐	☒	☒	☒	
4	OPENED sensor	Default tag table	Bool	%I0.2	☐	☒	☒	☒	
5	CLOSED sensor	Default tag table	Bool	%I0.3	☐	☒	☒	☒	
6	IN sensor	Default tag table	Bool	%I0.4	☐	☒	☒	☒	
7	OUT sensor	Default tag table	Bool	%I0.5	☐	☒	☒	☒	
8	RIGHT sensor	Default tag table	Bool	%I0.6	☐	☒	☒	☒	
9	LEFT sensor	Default tag table	Bool	%I0.7	☐	☒	☒	☒	
10	PB X10	Default tag table	Bool	%I1.0	☐	☒	☒	☒	
11	PB X11	Default tag table	Bool	%I1.1	☐	☒	☒	☒	
12	PB X12	Default tag table	Bool	%I1.2	☐	☒	☒	☒	
13	PB X13	Default tag table	Bool	%I1.3	☐	☒	☒	☒	
14	PB X14	Default tag table	Bool	%I1.4	☐	☒	☒	☒	
15	OBJECT sensor	Default tag table	Bool	%I1.5	☐	☒	☒	☒	
16	Move DOWN solenoid	Default tag table	Bool	%Q0.0	☐	☒	☒	☒	
17	Move UP solenoid	Default tag table	Bool	%Q0.1	☐	☒	☒	☒	
18	CLOSE gripper solenoid	Default tag table	Bool	%Q0.2	☐	☒	☒	☒	
19	OPEN gripper solenoid	Default tag table	Bool	%Q0.3	☐	☒	☒	☒	
20	Move OUT solenoid	Default tag table	Bool	%Q0.4	☐	☒	☒	☒	
21	Move IN solenoid	Default tag table	Bool	%Q0.5	☐	☒	☒	☒	
22	Move LEFT solenoid	Default tag table	Bool	%Q0.6	☐	☒	☒	☒	
23	Move RIGHT solenoid	Default tag table	Bool	%Q0.7	☐	☒	☒	☒	
24	Auto	Default tag table	Bool	%M0.0	☐	☒	☒	☒	
25	HMI MAN Move UP req	Default tag table	Bool	%M10.1	☐	☒	☒	☒	
26	HMI MAN Move DOWN req	Default tag table	Bool	%M10.0	☐	☒	☒	☒	
27	HMI MAN gripper OPEN req	Default tag table	Bool	%M10.3	☐	☒	☒	☒	
28	HMI MAN gripper CLOSE req	Default tag table	Bool	%M10.2	☐	☒	☒	☒	
29	HMI MAN Move OUT req	Default tag table	Bool	%M10.4	☐	☒	☒	☒	
30	HMI MAN Move IN req	Default tag table	Bool	%M10.5	☐	☒	☒	☒	
31	HMI MAN Move LEFT req	Default tag table	Bool	%M10.6	☐	☒	☒	☒	
32	HMI MAN Move RIGHT req	Default tag table	Bool	%M10.7	☐	☒	☒	☒	

Figure 52 - PLC Taglist: 1-32

33	SEQ Move DOWN req	Default tag table	Bool	%M20.0	☐	☒	☒	☒	
34	SEQ Move UP req	Default tag table	Bool	%M20.1	☐	☒	☒	☒	
35	SEQ gripper OPEN req	Default tag table	Bool	%M20.3	☐	☒	☒	☒	
36	SEQ gripper CLOSE req	Default tag table	Bool	%M20.2	☐	☒	☒	☒	
37	SEQ Move OUT req	Default tag table	Bool	%M20.4	☐	☒	☒	☒	
38	SEQ Move IN req	Default tag table	Bool	%M20.5	☐	☒	☒	☒	
39	SEQ Move LEFT req	Default tag table	Bool	%M20.6	☐	☒	☒	☒	
40	SEQ Move RIGHT req	Default tag table	Bool	%M20.7	☐	☒	☒	☒	
41	STEP	Default tag table	UInt	%MW30	☐	☒	☒	☒	
42	NEXTSTEP	Default tag table	UInt	%MW32	☐	☒	☒	☒	
43	Load	Default tag table	Bool	%M34.1	☐	☒	☒	☒	
44	Unload Start Button	Default tag table	Bool	%M34.3	☐	☒	☒	☒	
45	Home Button	Default tag table	Bool	%M34.4	☐	☒	☒	☒	
46	Object sense P Edge	Default tag table	Bool	%M200.0	☐	☒	☒	☒	
47	Object Detected	Default tag table	Bool	%M200.1	☐	☒	☒	☒	
48	Run Sequence	Default tag table	Bool	%M200.2	☐	☒	☒	☒	
49	<Add new>				☐	☒	☒	☒	

Figure 53 - PLC Taglist: 33-48

The PLC tag lists visible in Figure 52 and Figure 53 contain all of the tags used in this PLC programme. Several different address types were employed; Inputs (with the prefix “I”), Outputs (Prefix “Q”), and internal memory addresses (Prefix “M”). Memory words (Prefix “MW”) were also used for “Step” and “Next Step” tags as memory bits would not suffice and a larger binary storage was necessary as they had more than two states, So whilst every other tag is a bool unit, these two memory words were Uint (Unsigned Integer).

3 – System Testing, maintenance and Reliability

Testing Table:

Test No.	Task Description	Action	Expected Results	Actual Results	Success/Fail
MANUAL					
	Test manual control in auto	Put robot into auto and try to issue a manual move command	Robot should not move	Robot did not move	✓
	Test Sensors	Load program into PLC and view sensors on engineering HMI and manually move	Sensors for the current location of the robot should be lit, and should update as robot moves	The correct sensors were lit for the location of the robot (Figure 71 - Figure 75)	✓
	Test Up Movement	Press Up manual control	Robot should move up and up sensor should be made	The robot moved up and up sensor was made (Figure 72)	✓
	Test Down Movement	Press Down manual control	Robot should move down and down sensor should be made	The robot moved down and down sensor was made (Figure 71)	✓
	X-Axis Movement	Drive robot left and right using manual control	Robot should move right and left as corresponding buttons are pressed, and corresponding sensor indication should light.	The robot moved as instructed and the correct sensor indication illuminated (Figure 75)	✓
	Y-Axis Movement	Drive robot in and out using manual control	Robot should move in and out as corresponding buttons are pressed, and corresponding sensor indication should light.	The robot moved as instructed and the correct sensor indication illuminated (Figure 74 & Figure 73)	✓
	Gripper Functions	Open and close grippers using manual control	Grippers should open and close as commanded and corresponding sensor indication should light	Grippers functioned as instructed and the correct sensor indication illuminated (Figure 73 & Figure 74)	✓
	Check that solenoid indication is working on HMI	Watch the solenoid indication whilst manually controlling	Indication should display currently operated solenoids	Indication worked as intended (Figure 71 - Figure 75)	✓
	Test movement of robot in down position	Drive robot down then attempt to	Robot should not move	Robot started to move, at which point the e/stop was pressed to prevent any damage	✗

		move in the X and Z axes.			
	Re-Test movement of robot in down position with new interlocks in place (Figure 19 - Figure 22)	Drive robot down then attempt to move in the X and Z axes.	Robot should not move	Robot did not move due to interlocks	✓
	Check interlocks in ladder	Try to move in/out and left/right in down position whilst monitoring in ladder	Should be visible in ladder that the solenoid coil does not activate	Coil did not activate	✓
AUTO					
	Test home movement upon program start	Start programme and put into auto	Robot should move to home position and be in step 1.	Robot moved home (Figure 55)	✓
	Check to see that home condition sensors are met	Start programme and monitor SCL	In, Left, and Open sensors should be true	The sensors were true	✓
	Check step counter updates with steps	Run through auto sequence	Step should display correct current step number for the action being performed	Displayed correct step number (Figure 55 - Figure 70)	✓
	Test Load Sequence	Place object in incoming ramp whilst robot in home position	Auto sequence should start	Auto sequence ran till completion then returned home (Figure 55 - Figure 60)	✓
	Check timer on step 9 and 17	Run through auto sequence and time step 9 and 17	Both step 9 and 17 should stay on for 5 seconds each	Stayed on for the correct amount of time	✓
	Test unload	Press start unload with robot in unload and home position	Unload sequence should be carried out	Unload sequence ran till completion then returned home (Figure 63 - Figure 70)	✓
	Test unload does not work under load conditions	Roll object in with "unload" selected	Auto sequence should not start	Unload did not start	✓
	Test sequence run/stop switch (freeze)	Run the load sequence and toggle switch as it runs	Robot should stop after finishing its current action and not restart until put back into run	Robot stopped as intended and continued once put back into run	✓

HMI					
	Test objects processed count and "maintenance needed" indication	Put robot in auto and load, then put 3 objects into the ramp	Load sequence should run through 3 times, each time an object is placed in station 4 count should increase. And once 3 have been placed then light should turn green.	Counted as intended and light illuminated (Figure 76)	✓
	Test engineering screen password protection	Try to access engineering screen from operator screen	Prompt should appear asking for username and password, once enter should allow access to engineering screen	Prompt appeared and when credentials entered engineering screen was accessed (Figure 49)	✓
	Logoff feature	Exit engineering screen and try to return to it	Prompt should reappear upon trying to regain access to engineering screen	Prompt reappeared	✓
	Test Object Count	Place object into ramp	Object should roll in front of object sensor and count should increase by 1	Object rolled in front of object sensor, but rattled and counted up 3 times for the one object.	✗
	Re-Test Object Count after adding de-bouncer	Place object into ramp	Object should roll in front of object sensor and count should increase by 1	Count only increased by 1	✓

Considered maintenance of the system by including a "maintenance needed" light on the operator HMI in Figure 47, which lights on a pre-set value of objects processed (Automatic sequence operations) set by the PV of the objects processed counter Figure 45.

Also included a pushbutton alternative to start the load sequence in order to test it without the need for a product to be present (Figure 25).

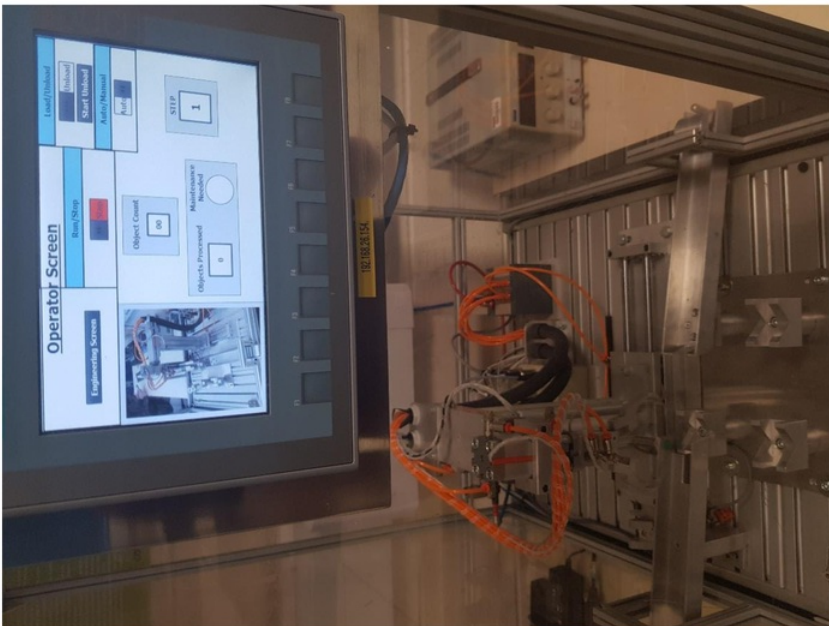


Figure 55 - Loading Seq Test: Step 1

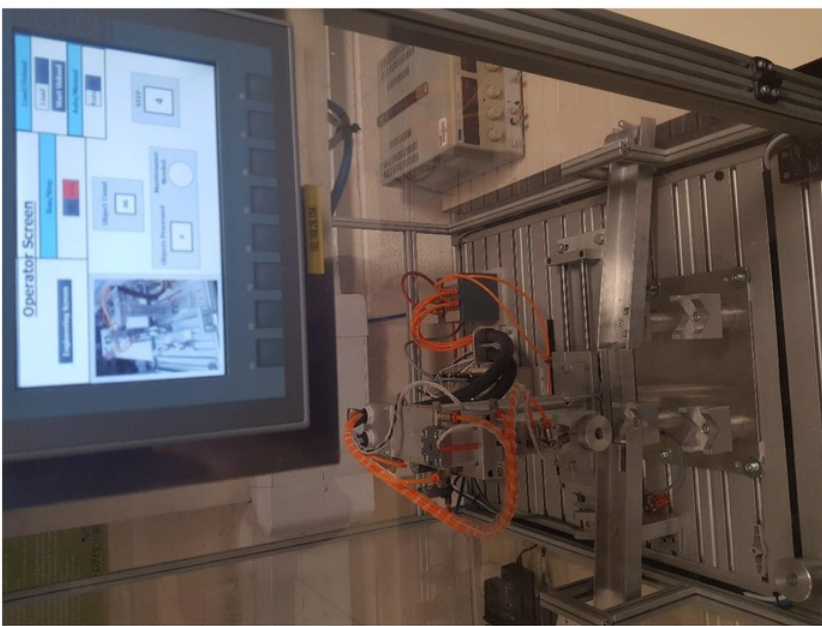


Figure 54 - Loading Seq Test: Step 4

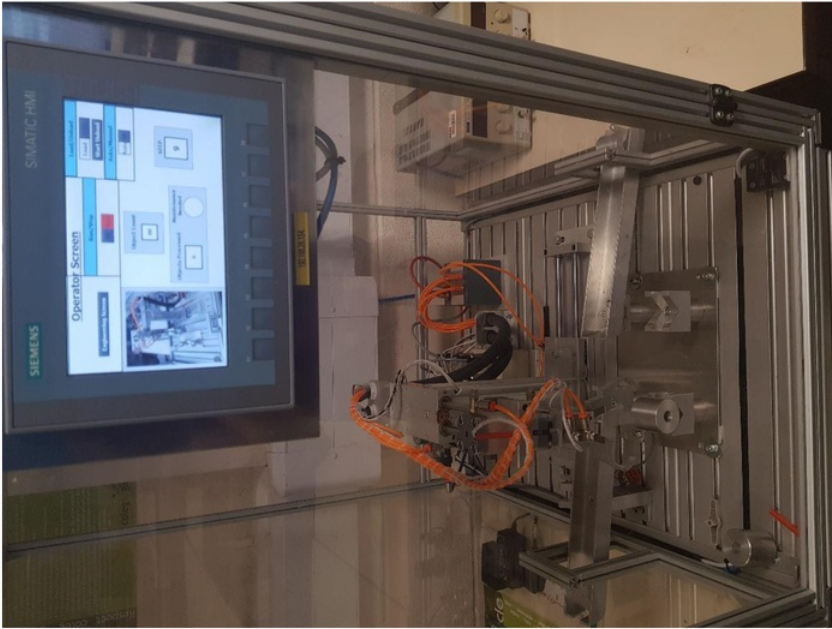


Figure 57 - Loading Seq Test: Step 9

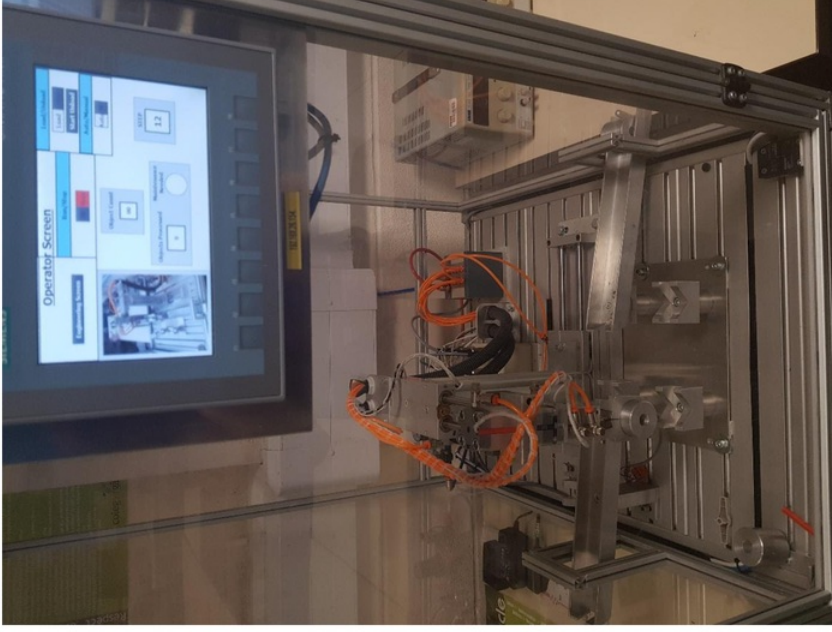


Figure 56 - Loading Seq Test: Step 12

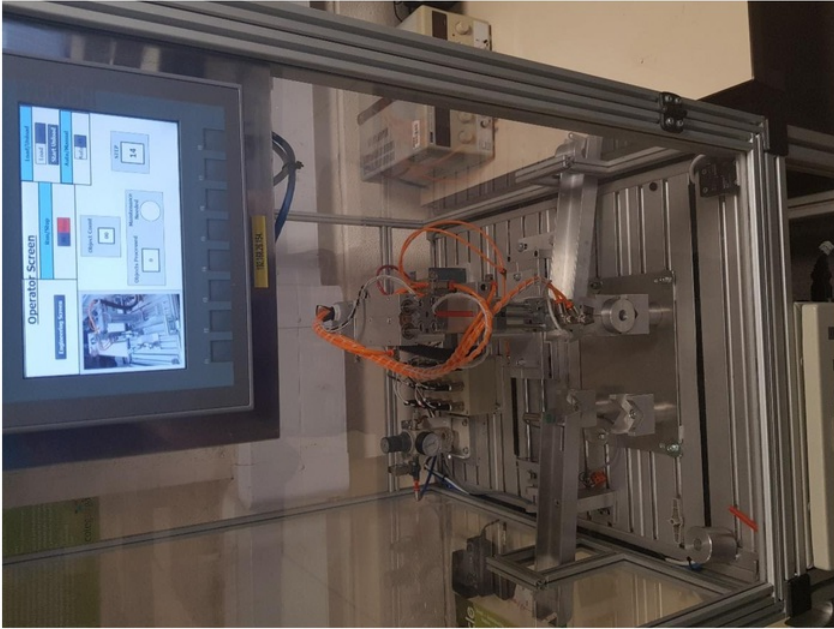


Figure 59 - Loading Seq Test: Step 14

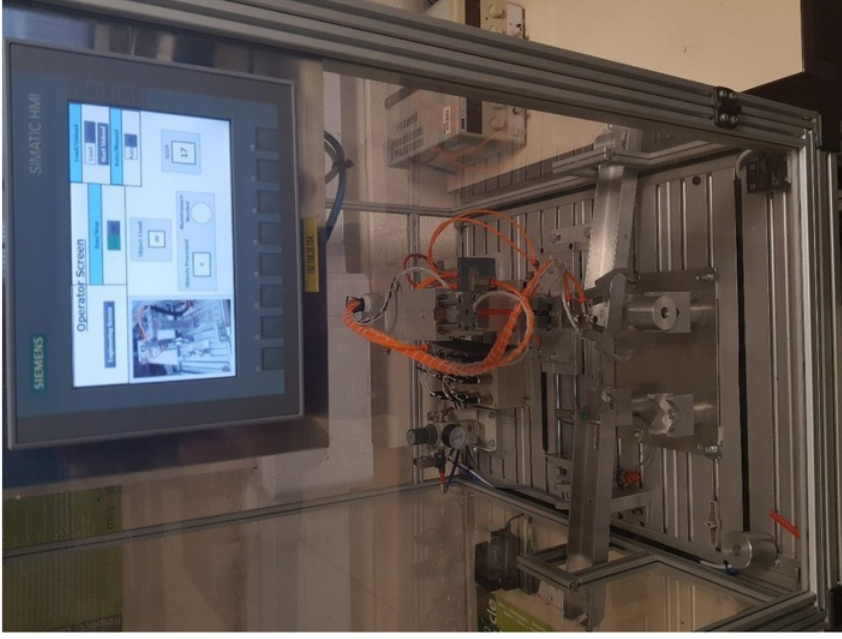


Figure 58 - Loading Seq Test: Step 17

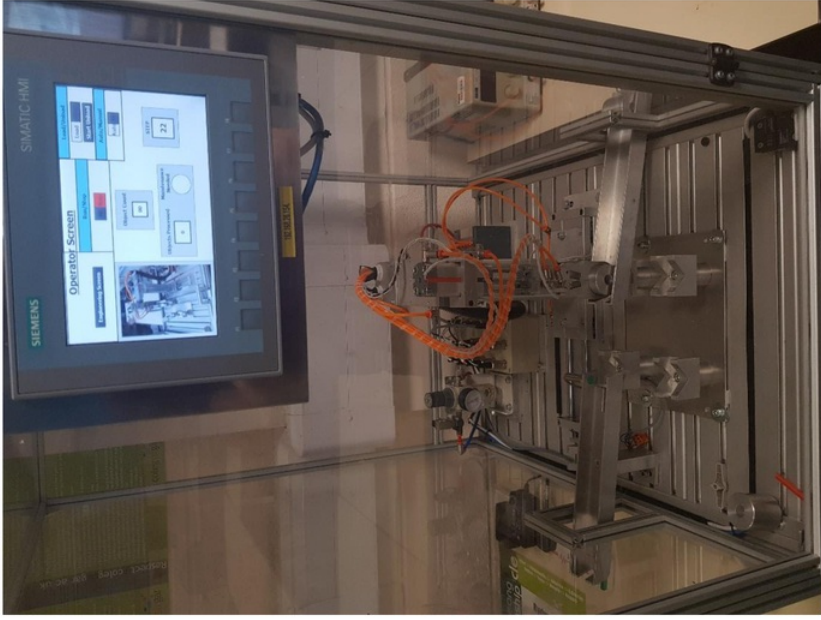


Figure 61 - Loading Seq Test: Step 22

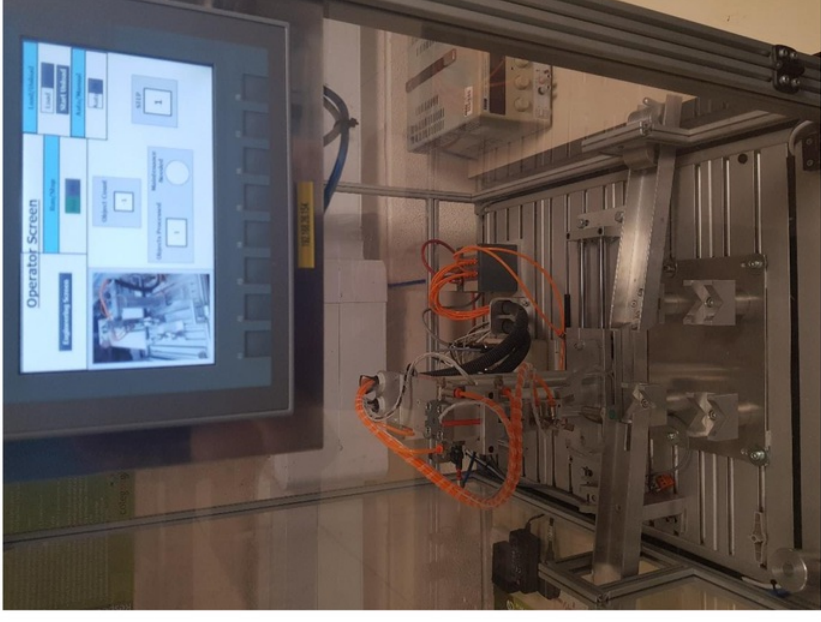


Figure 60 - Loading Seq Test: Returned Home

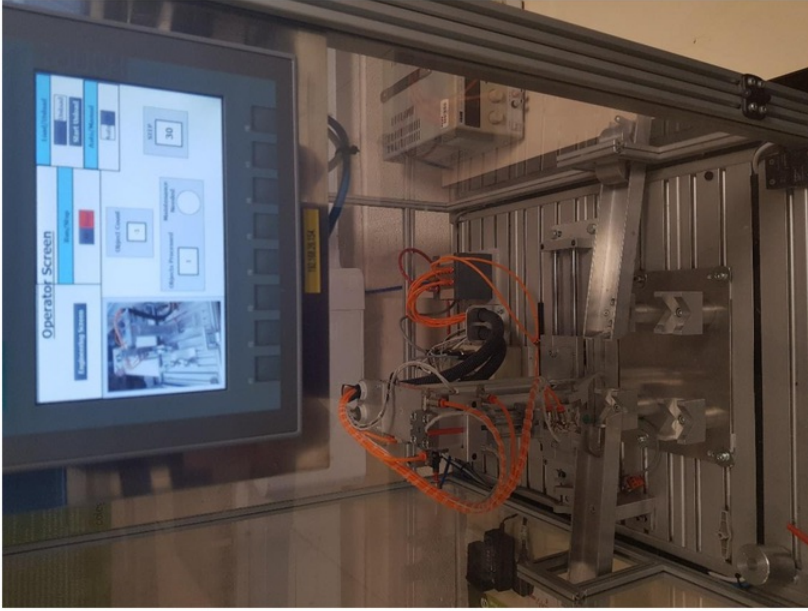


Figure 63 - Unloading Seq test: Step 30

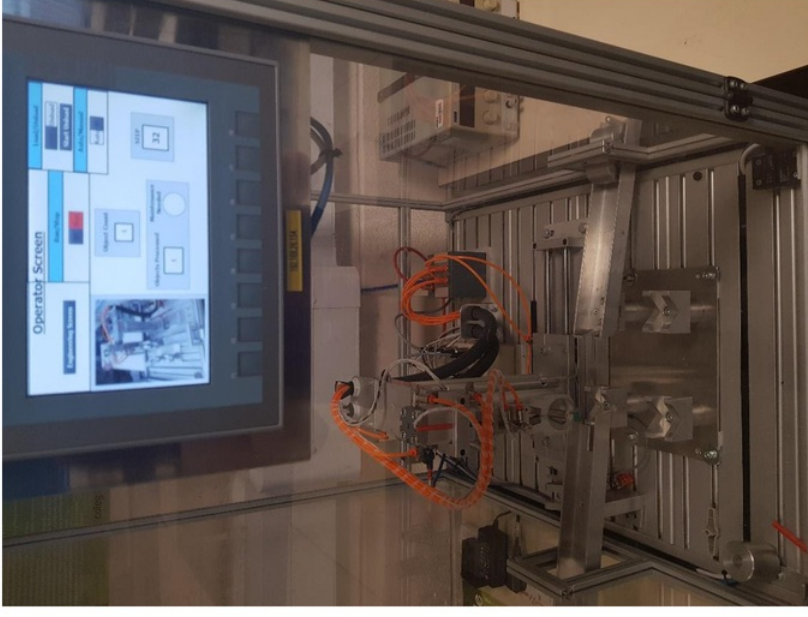


Figure 62 - Unloading Seq test: Step 32

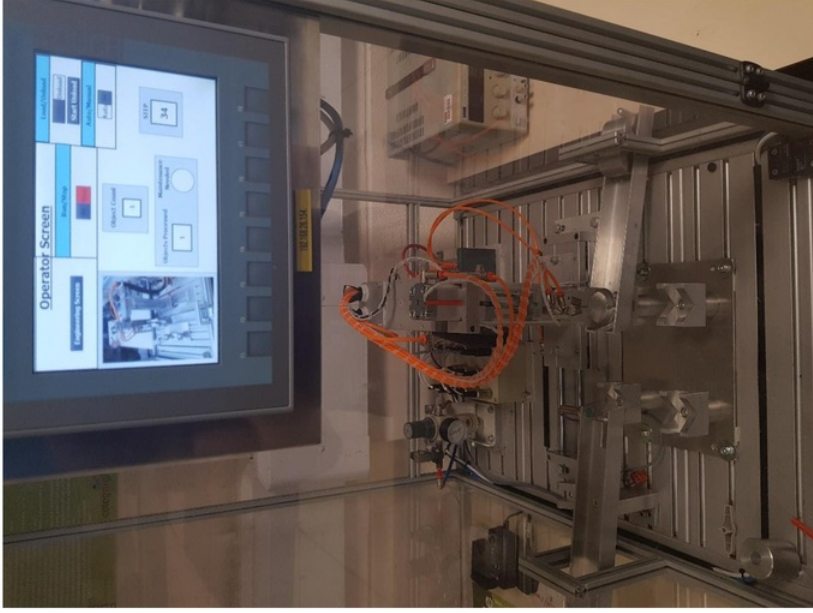


Figure 65 - Unloading Seq test: Step 34

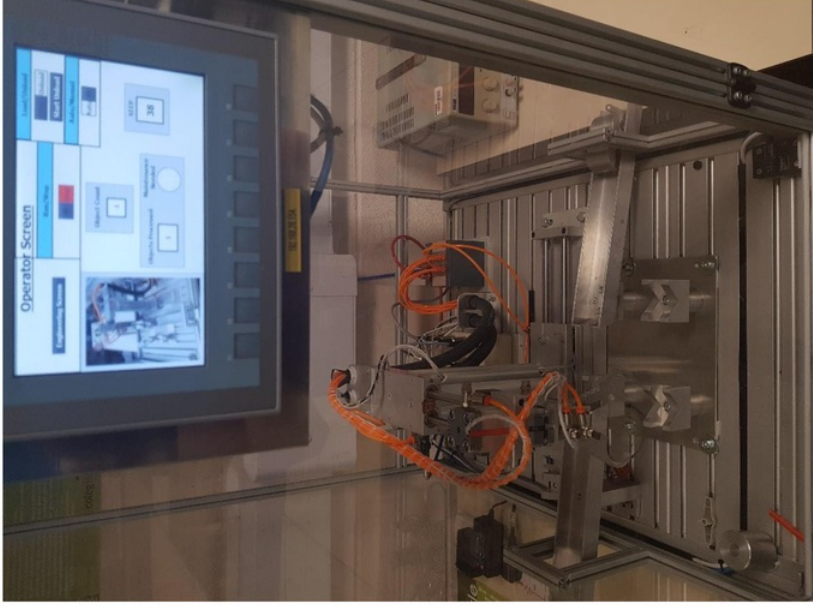


Figure 64 - Unloading Seq test: Step 38

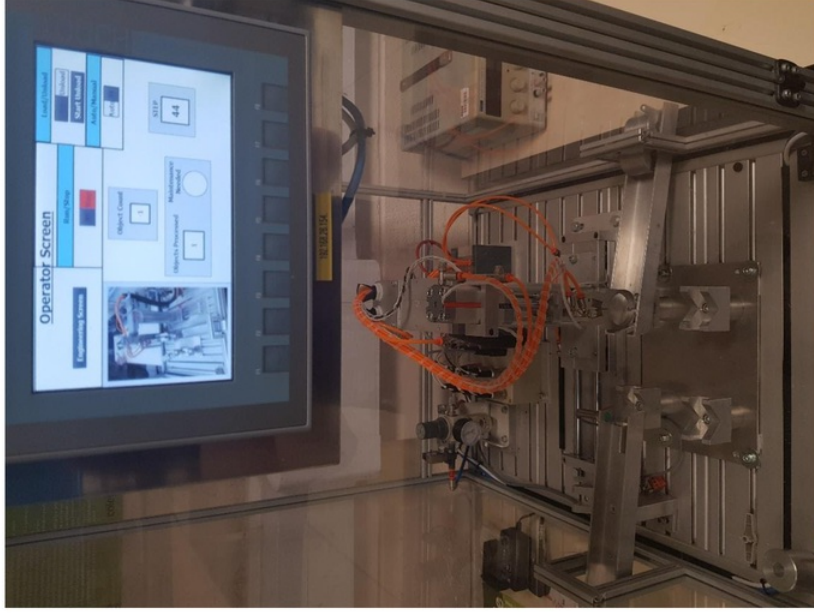


Figure 67 - Unloading Seq test: Step 44

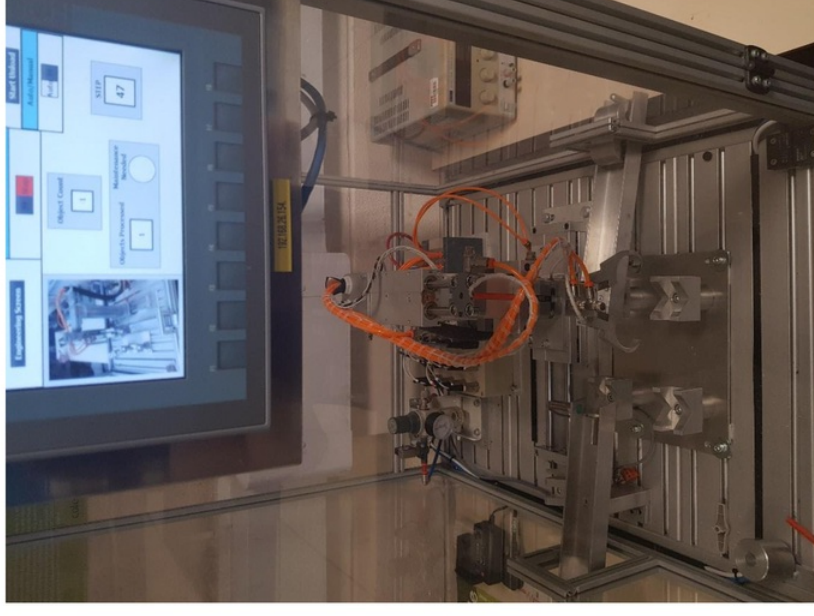


Figure 66 - Unloading Seq test: Step 47

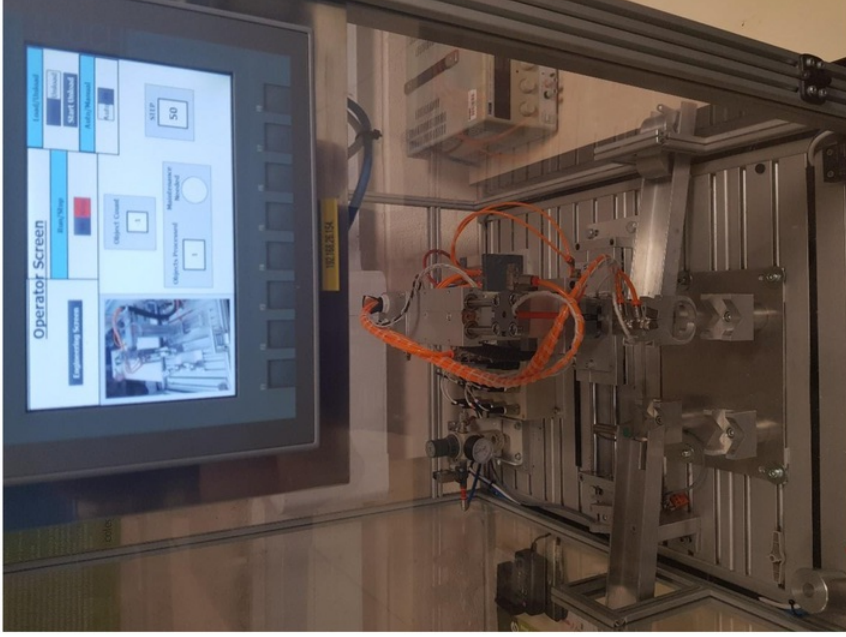


Figure 69 - Unloading Seq test: Step 50

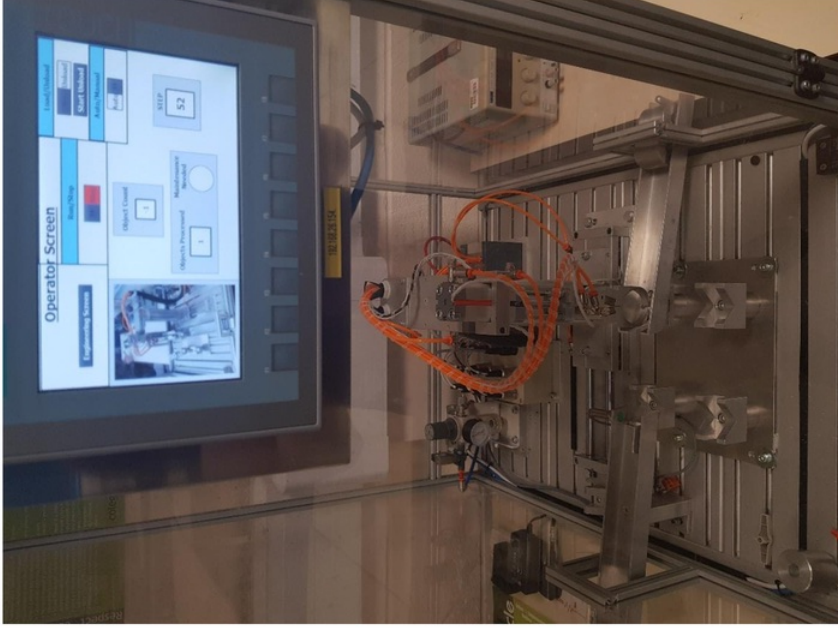


Figure 68 - Unloading Seq test: Step 52

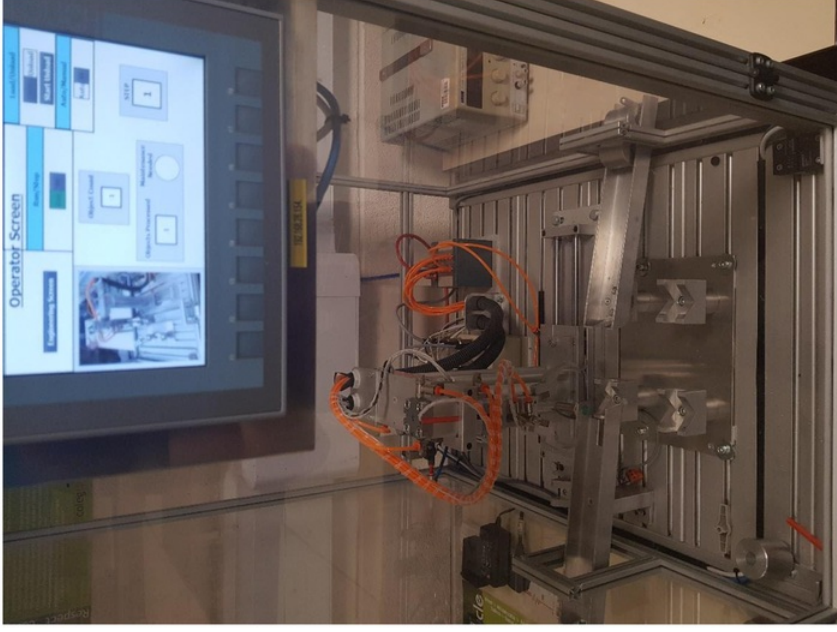


Figure 70 - Unloading Seq test: Returned Home



Figure 72 - Manual Test: Up movement



Figure 71 - Manual Test: Down movement



Figure 74 - Manual Test: In movement & Open Gripper



Figure 73 - Manual Test: Out Movement & Close Grip

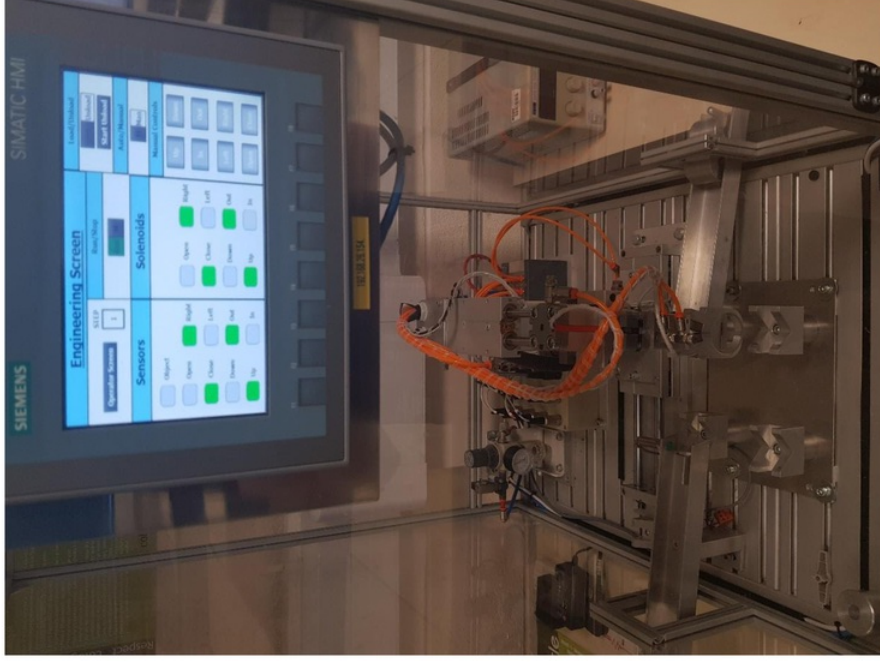


Figure 75 - Manual Test: Right Movement

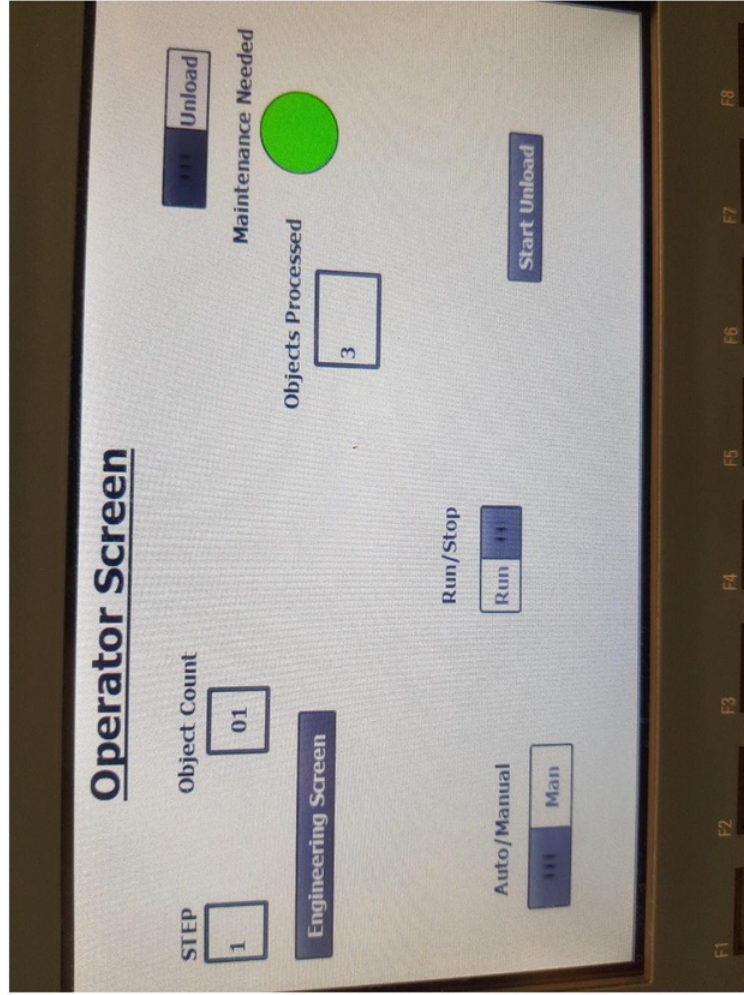


Figure 76 – HMI: Objects processed count test

4 – Conclusions and Recommendations

Avoid using the first person and expressing

I feel as though I have met the aims and objectives put forth by the brief, as well as the learning outcomes of the module.

Throughout the process I kept a testing table, which allows me to compare my theoretical expectations with the practical results and if any discrepancies occur I have adapted the program in order to overcome them and ensure correct operation of the pick and place robot according to the specifications.

I have displayed an ability to use ladder logic (LAD), and Siemens Structured Control Language (SCL) and Sequential flow chart (SFC) as a means of planning my SCL.

I have successfully met all of the objectives specified having created manual control, automatic control (featuring a loading and unloading sequence). I have also created two HMI screens, one for the operators and the other for engineering, with the latter full of useful maintenance and fault finding aids and is password protected to ensure only authorised and competent engineering personnel gain access to it.

I have also considered maintenance aspects for the robot throughout, one such example being the maintenance needed indication in Figure 47 on the operator screen that recommends maintenance be undertaken every set amount of cycles of the robot. I have ensured reliability of the object counter by creating a de-bouncer to ensure that a single object does not trigger multiple counts.

Whilst my program has met the specifications some further improvements could be made to it and the pick and place robot.

Inductive proximity detectors could be placed on station 2 and 3 in order to verify that there are objects there, allowing us to create a more accurate and efficient unloading sequence by only unloading the stations with objects present. Another object sensor could also be put into station 4 in order to create a more accurate object counter, allowing us to precisely tell when an object has exited the system and not relying on using the final step of the load sequence. This would allow the object counter to work in load, unload, and manual.

A longer range inductive proximity sensor could be used in place of the current object sensor, that way the de-bouncer would not be necessary and the objects would be picked up more reliably.

A safety interlock could be implemented that prevents the grippers from opening whilst in the up position, as opening the grippers whilst up and holding an object would drop it from a height. Whilst this does not present much danger in this small scale application, in an industrial scale application this could be catastrophic and dangerous.

Could put a means of inputting a new PV to the objects processed counter on the engineering, so that the amount of operations needed to light the “maintenance needed” light can altered more intuitively, as opposed to rewriting it in the programme.

Have the objects processed counter reset every week at the end of the final shift, and write the value to a database. This can then be used to create a trend of productivity.

PSDE - 4005 Assignment 2

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

92/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

Text Comment. incorrect symbol and should also have 3 wires

Text Comment. good labelling

Text Comment. incorrect connections, should be connected to 0Vdc here

Text Comment. a helpful wiring diagram containing lots of useful information

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

Text Comment. excellent design, prevents damage by forcing the robot to move up when not on a station.

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

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

Text Comment. Avoid using the first person and expressing feelings. Be more objective.

GRADING FORM: CSG-RUBIC

RESEARCH-INFORMED LITERATURE

 n/a

KNOWLEDGE AND UNDERSTANDING



Section 1: Hardware Investigation - This is a good section contains lots of helpful information for your reader and system builder/maintenance engineer. One incorrect symbol on your drawing and some incorrect connections. (8/10) Section 1: System Design and I/O list - A well laid out I/O list, very good SFC chart sequence design and well designed ladder networks and SCL code. (10/10) Section 2: Program Explanation – Each ladder network has been explained and each block of SCL has also been explained. (9/10)

27

ANALYSIS



Section 3: System Testing – This section contains a detailed table of results linked to supporting photographic evidence. (10/10) Section 3: Maintenance and Reliability - . Excellent evidence in the report through design and testing with some comments made. A subsection could have been included which specifically targets these issues. (9/10) Section 4: Conclusions and recommendations - Excellent but don't write in the

28

first person. (9/10)

PRACTICAL APPLICATION AND DEP.



Section 2: Program Network Comments - Very good ladder network comments and many in-program comments contained in the SCL code. (9/10)

Section 2: HMI Screen Designs – Excellent designs with password protection for the engineer. (10/10)

Section 3: Presentation of Test Results - . Excellent (9/10)

28

SKILLS FOR PROFESSIONAL PRAC.



All sections - A well structured, present and written report at a professional standard except for your use of the first person in the last section. You must avoid been too subjective. (9/10)

9