

2017198

by Ryan James

FILE	16861_RYAN_JAMES_2017198_178848_2058413246.PDF (2.45M)		
TIME SUBMITTED	08-FEB-2021 09:56AM (UTC+0000)	WORD COUNT	3708
SUBMISSION ID	144385575	CHARACTER COUNT	17291

Course:	HNC Electrical and Electronic Engineering
Module:	SGEE4004 Programmable Logic Controllers
Assignment 1:	Pick and Place Robot
Name:	Ryan James
University number:	2017198
Date submitted:	08/02/2021
Word count:	2641

Contents

Good Structure

1. Research.....	3
2. Hardware Investigation, Circuit Diagram, and I/O List	5
3. Program Design and Explanation.....	7
4. Simulation Testing.....	13
5. Practical Testing and Team Working.....	15
6. Discussion, Conclusions and Recommendations	16
7. Works Cited.....	17
Appendix	18

1. Research

According to Khaled Kamel, a programmable logic controller, also known as a PLC, is a micro-processor-based computer unit that can be used to perform many control instructions. They also go on to say that they are commonly used today in robotic and different control systems. (Kamel, 2014)

Suppress the author's name if you have already

There are three main varieties of PLCs, which are Unitary, Modular and Rack Mounted. A Unitary PLC is considered to be the simplest of all three as it contains all of its components (including inputs and outputs) in one unit and is generally found attached to the equipment being controlled. A Modular PLC is slightly different to a Unitary PLC as it is capable of combining several input and outputs together to create a specialized type. Due to this personal trait, it makes the PLC easy to be customised. Rack Mounted PLCs can be compared to Modular PLC types because they share a similar trait in that they can be adapted to have more digital inputs and outputs, however with a Rack Mounted PLC they are not combined modules they are single modules, resulting in any extra modules required being added via a network. They are also organised on a “rack” hence the name and can be removed and added as required (Aries, 2021).

Correctly cited

Siemens produce a variety of PLCs in the formats of Unitary, Modular and Rack Mounted. Some examples of each of these types of Siemens PLCs can be represented in a comparison table.

	Type of PLC	Cost (£)	Speed per 1K m/s	Memory (kiloByte)	Number of I/O	Net weight (kg)	Program Memory (KB)
S7-1211	Unitary	182	0.08	50.00	6	0.38	50
S7-1212	Modular	224	0.08	75.00	16	0.39	75
S7-1214	Modular	369	0.08	100.00	26	0.49	100
S7-1217	Modular	803	0.08	150.00	28	0.50	150
S7-1500	Modular	3851	0.03	512.00	32	0.94	512
S7-400	Rack Mounted	14691	0.01	64000.00	>32	1.06	64000

(PG, 2021)

Correctly cited

Hans Berger states that the Siemens S7-1200 series is the modular micro-system, with a central processing unit which can be said to be power failure-proof. They go on to say that this means in the event of a power failure, tags which are saved are stored in the retentive memory for safety (Berger, 2013).

Use the third

Suppress the author's name if you have already stated

Within the research task I am also set a question to decide whether the PLC used for the Pick and Place robot is the most suitable; the PLC used for this was the Siemens S7-1214. As we can see in the comparison table above it is a modular PLC. This identifies straight away that a Unitary PLC like the S7-1211 would be inadequate because of its ability not to expand signal modules.

From looking at the comparison table a trend is starting to develop in that as the speed per 1K m/s of the PLC increases so does the cost. Therefore, purely down to financial reasons the rack mounted S7-400 would also be inadequate as its too expensive.

After explaining why, the S7-1211 and the S7-400 would be inappropriate, it is now possible to say that the Modular PLC was the right type to be selected. It is also possible to say that the S7-1214 was the correct PLC to be used because of the number of input and outputs required for the Pick and Place robot. The robot required a total of 22 inputs and outputs, meaning the S7-1212 would not have enough. Therefore, as the S7-1214 is the correct option as it can withhold a total of 26 inputs and outputs which is within the required amount and offers room for some expansion. However, if greater expansion were required then the S7- 1217 would be a better option.

*Good conclusion but
space is a limitation*

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

Throughout the pick and place robot there has been numerous different types of hardware used which can be shown on the drawing of the PLC below in figure 71. The hardware within the pick and place robot consists of push buttons to start and stop the robot, pneumatic pistons, solenoids, reed switch sensors and also proximity sensors. The hardware used can be identified as either being an input or an output as shown in the table below.

You
shoul

	Input / Output (I/O)	Explanation of operation	PLC addresses
UP sensor	I	Indicates when the robot is the "up" position.	%M0.0
DOWN sensor	I	Indicates when the robot is the "down" position.	%M0.1
OPEN sensor	I	Indicates when the gripper of the robot is "open".	%M0.2
CLOSED sensor	I	Indicates when the gripper of the robot is "Closed".	%M0.3
IN sensor	I	This sensor indicates when the robot has moved "in", but only when in the "up" position.	%M0.4
OUT sensor	I	This sensor identifies when the robot has moved "out", but only when in the "up" position.	%M0.5
RIGHT sensor	I	Indicates that the robot has moved "right" when in the "up" position.	%M0.6
LEFT sensor	I	Indicates the robot has moved "left" but only when in the "up" position.	%M0.7
OBJECT sensor	I	Indicates when the robot has detected an "object" in the chute.	%M1.5

Move UP solenoid	O	Controls the movement when the robot needs to move “up”.	%Q0.1
Move DOWN solenoid	O	Controls the movement when the robot needs to move “down”.	%Q0.0
Move LEFT solenoid	O	Controls the movement when the robot needs to move “left”.	%Q0.6
Move RIGHT solenoid	O	Controls the movement when the robot needs to move “right”.	%Q0.7
Move IN solenoid	O	Controls the movement when the robot needs to move “in”.	%Q0.5
Move OUT solenoid	O	Controls the movement when the robot needs to move “out”.	%Q0.4
Gripper OPEN solenoid	O	Controls the movement when the gripper needs to “open”.	%Q0.3
Gripper CLOSE solenoid	O	Controls the movement when the gripper needs to “close”.	%Q0.2

3. Program Design and Explanation

Explain the system or design, don't tell a story of how it was

Unnecessary

was TIA Portal

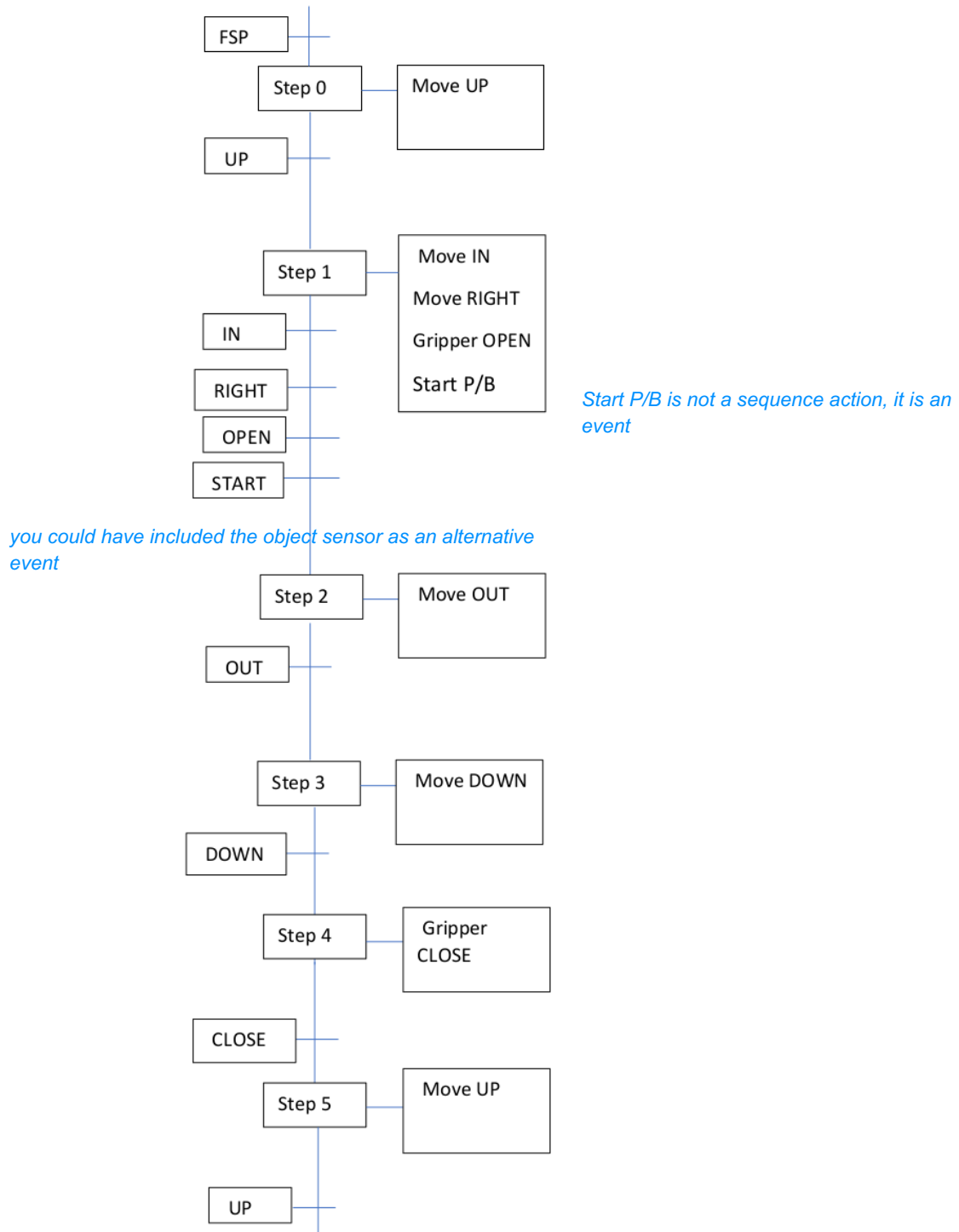
The next stage of this assignment was to use the Siemens TIA Portal to design the pick and place robot project. The project had to be able to run in both manual and automatic modes which would be shown on 2 HMI screens (Operator and Engineer screens). The screens created can be shown below in figures 1 and 2.

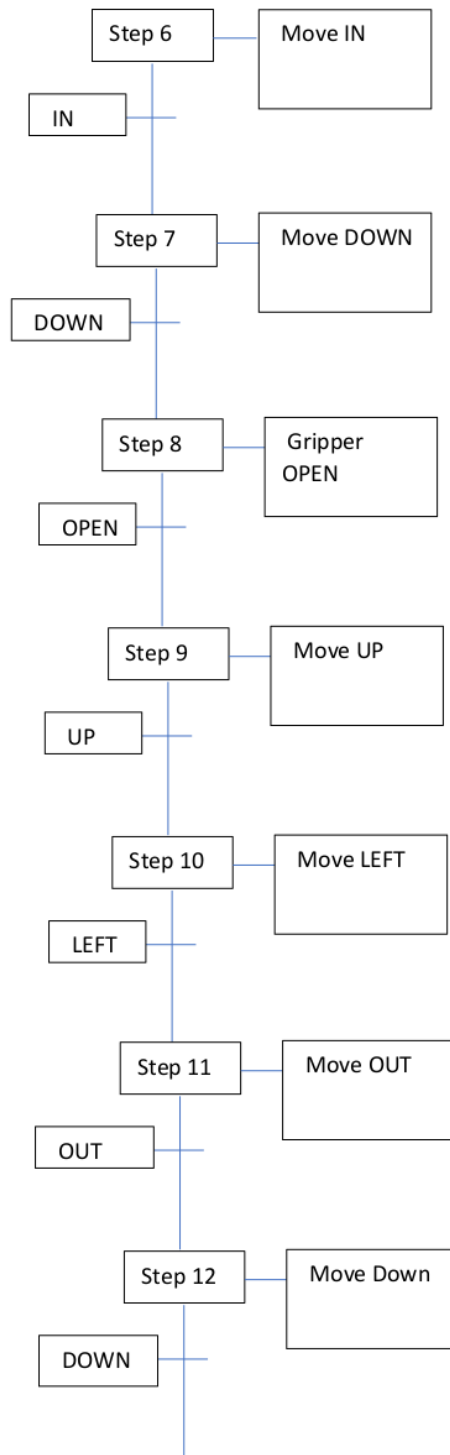
It was essential that the robot had interlocks created for when in manual mode otherwise there is a potential to break the robot. For example, if an object was in station 2 and the robot was moving to the station whilst going down there is a potential for it to crash. The interlocks prevent this by only allowing the robot to move to a position until the reed switch (sensor) is made. From looking at figures 3-9, it is possible to see the interlocks in place. Additional descriptive comments are written in the program to help with possible fault finding. From looking at the figures mentioned the sensors can be seen to be normally closed; this is because once the solenoid has moved the robot to a station or new position there is no need for it to run which as a result preserves energy from being wasted. As a result, the solenoids switch off once the sensor is made.

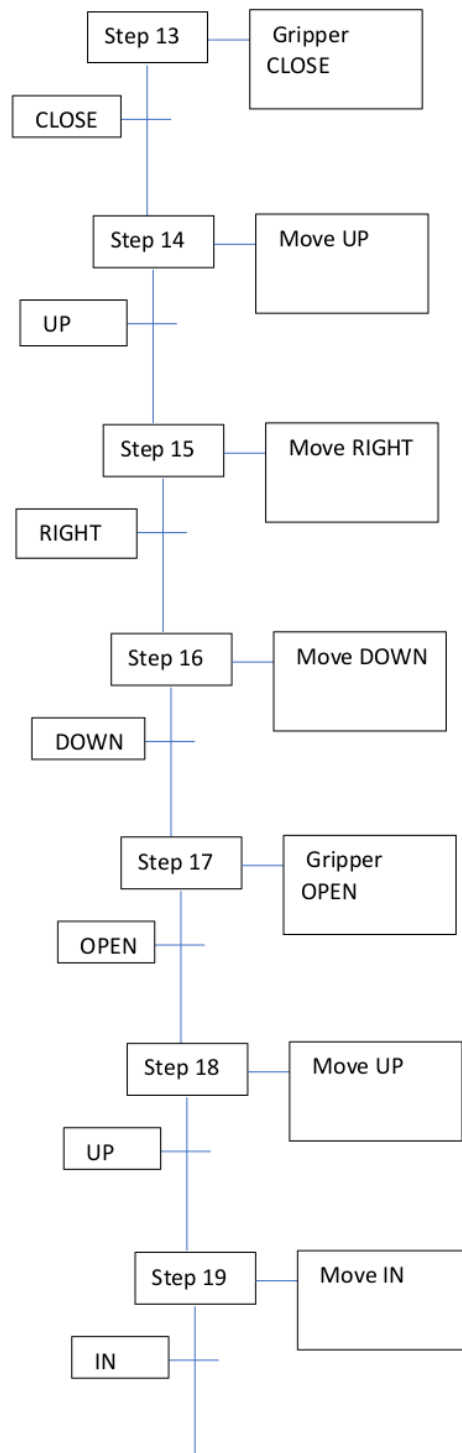
The specification of this assignment required an HMI screen for the operator with a step and object counter which can be shown in figure 1. A pause and start sequence buttons were also added to help the operator along with reset buttons for both counters. A further and final addition to the HMI screen was a lamp that indicates when the sequence is running in automatic mode. Similarly, the HMI of the engineers screen also required a pause and start button to control the sequence along with other essential controls to help with potential maintenance or fault finding as shown in figure 2.

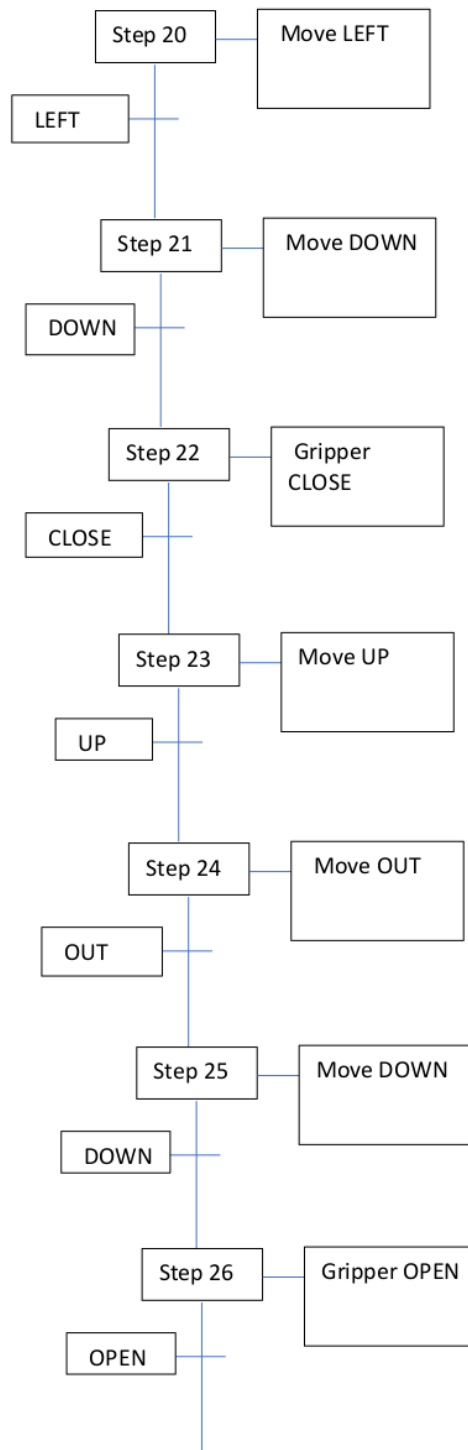
When the robot was running in automatic mode an SFC (Sequential Function Chart) was created to aid with the designing of the program, the SFC can be seen below. This chart helped break down the programming as it breaks down each step into a simpler form. From the SFC chart it is possible to see that before the robot starts any form of sequence it must be up before moving in any direction.

don't use boxes for labels, switch off the outline next time
you label a drawing using a text box









*return to
step 0*

Within figures 14-25 there are additional comments to help understand the SCL auto sequence code. The SFC chart above helped write this code as it forms a step-to-step guide in a simpler form. The way the code works is by the following (take figure 15), when the robot is started the first thing it does is move up because of the safety lock already stated. The robot has requested this movement, so when the "up sensor" is met, the move up request =0 as the instruction is made, therefore it can progress to the next step which is 1.

In order to write the program in the form of networks and also the SCL code there had to be PLC tags. The tags were helped formed by use of the I/O list in task 2, with the memory elements starting from M2000.0 as seen below. The tags of the PLC are shown in figures 26-29, and they are tagged to both the engineer and operator HMI screens where it was appropriate to do so, see figures 1 and 2 for the tags on the HMI screens.

The sequence of the robot works after a delay of 2s at the object sensor, in order for it to detect the object in the entry chute. After detecting the object, the robot moves home and then progresses to the entry chute via the automatic sequence. However, if the robot detects there are more objects in the system by use of the object counter then it must move them; accordingly, if an object is at station 3 then it must be moved to the exit chute station 4, and so if there is an object at station 2 it must then be moved to station 3 to allow the robot to bring in an object to station 2.

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. Some are included.

4. Simulation Testing

Testing Table could be

Auto(A) /Manual (M)	What is the test?	Pass(P)/ Fail(F)	Figure reference
M	Testing “up” interlock	P	30
M	Testing “down” interlock	P	31
M	Testing “right” interlock	P	32
M	Testing “left” interlock	P	33
M	Testing “in” interlock	P	34
M	Testing “out” interlock	P	35
A	Testing step 1 – counter 1	P	36
A	Testing step 20 – counter 1	P	37
A	Testing step 21 – counter 1	P	38
A	Testing step 22 – counter 1	P	39
A	Testing step 23 – counter 1	P	40
A	Testing step 24 – counter 1	P	41
A	Testing step 25 – counter 1	P	42
A	Testing step 26 – counter 1	P	43
A	Testing step 0 – counter 2	P	44
A	Testing step 1 – counter 2	P	45
A	Testing step 10 – counter 2	P	46
A	Testing step 11 – counter 2	P	47
A	Testing step 12 – counter 2	P	48
A	Testing step 13 – counter 2	P	49
A	Testing step 14 – counter 2	P	50
A	Testing step 15 – counter 2	P	51
A	Testing step 16 – counter 2	P	52
A	Testing step 17 – counter 2	P	53
A	Testing step 18 – counter 2	P	54
A	Testing step 19 – counter 2	P	55
A	Testing step 20 – counter 2	P	56
A	Testing step 21 – counter 2	P	57
A	Testing step 22 – counter 2	P	58
A	Testing step 23 – counter 2	P	59

A	Testing step 24 – counter 2	P	60
A	Testing step 25 – counter 2	P	61
A	Testing step 26 – counter 2	P	62
A	Testing step 0 – counter 3	P	63
A	Testing step 1 – counter 3	P	64
A	Testing step 2 – counter 3	P	65
A	Testing step 3 – counter 3	P	66
A	Testing step 4 – counter 3	P	67
A	Testing step 5 – counter 3	P	68
A	Testing step 6 – counter 3	P	69
A	Testing step 7 – counter 3	P	70

After this step, the counter reverts back to step 8 counter 2 and continues for the rest of the sequence in this manner completing the testing results.

5. Practical Testing and Team Working

This section was a group task, consisting of 3 people to practically test the program on the PLC pick and place robot, however due to COVID-19 lockdown restrictions this was not possible and as a result had to be done on a video call.

The program tested worked well and passed all the required tests in order to do so. The group was responsible for observing and recording any necessary results however to do this the group was split up into individual roles. As the program within this report was used to be tested the other two members of the group were responsible for recording the manual and automatic results, which can also be found within the assignment in relevant sections (see section 4 for tabulated results).

6. Discussion, Conclusions and Recommendations

As previously stated, due to COVID 19 lockdown restrictions the program was unable to be tested physically on the PLC pick and place robot. However, as a group in section 5 the program was able to be tested via the simulation methods available on TIA portal. As a recommendation for the future a practical test where lockdown restrictions are no longer an issue would be advisable.

With the results obtained from the testing of the program, its is possible to say that the consumer needs were met as the specification was met. The robot had to move up before being allowed to move in any other direction and in order to move down it had to be located over a station which the results are showing with the safety locks in place. When the program was run in the auto sequence it was possible to witness that the safety interlocks were in place via the engineer's screen.

Maintenance efficiency needs have been met when the program was written in the main block networks. For example, as stated already when the solenoid has moved after being energised it turns off when it hits the sensor which makes it energy saving as it is not running. It is possible to say that by it not running it is prolonging its life expectancy as a component since it is not energised and working all the time.

With the overall testing of the program done on TIA portal it was possible to obtain results of the program working efficiently, however something that was noticed was with network 8 (figure 10) in that it does not have a safety interlock to prevent the gripper opening when it is not down. It would therefore be recommended that an interlock is to be put in this network which would only allow the gripper to open when down on a station. This would prevent any harm to the user and of course the object itself.

Furthermore, it is possible to conclude that with a practical test to visually see the robot moving efficiently and accurately, along with an additional safety interlock on network 8 the open gripper solenoid, that this would greatly improve the design of the pick and place robot.

7. Works Cited

Correct method of
citing eg.

Aries, B., 2021. *Techwalla*. [Online]
Available at: <https://www.techwalla.com/articles/the-types-of-plcs>
[Accessed 05 January 2021].

Berger, H., 2013. Automating with Simatic S7-1200: configuring. In: s.l.:Publicis, p. 1.

Kamel, K., 2014. Programmable Logic Controllers: Industrial Control. In: *Industrial Control*. Maidenhead: McGraw-Hill, p. 2.

PG, 2021. *Parmley Graham*. [Online]
Available at: <https://www.parmley-graham.co.uk/automation/siemens/plc-s/>
[Accessed 06 January 2020].

A small range of source given. For more marks, you need a lot more sources than this to show evidence of wide reading.

Appendix

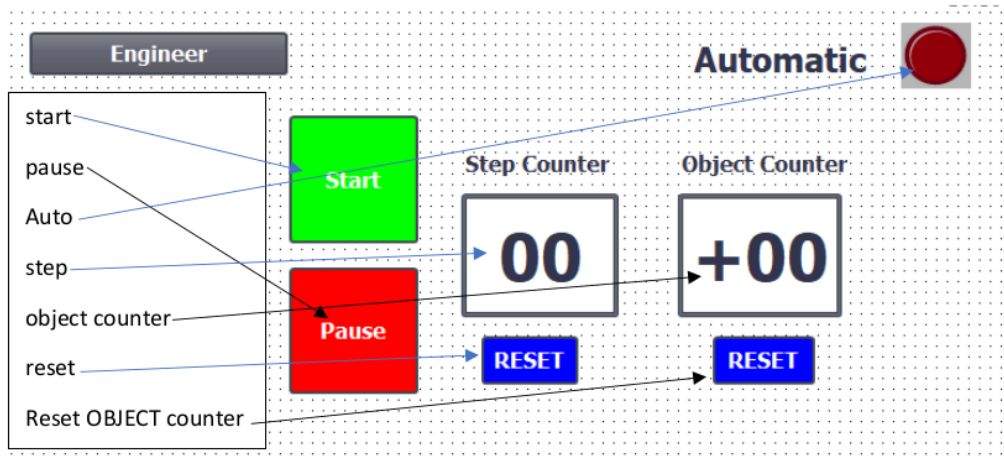


Figure 1: Image of Operators screen, with indication of PLC tags.

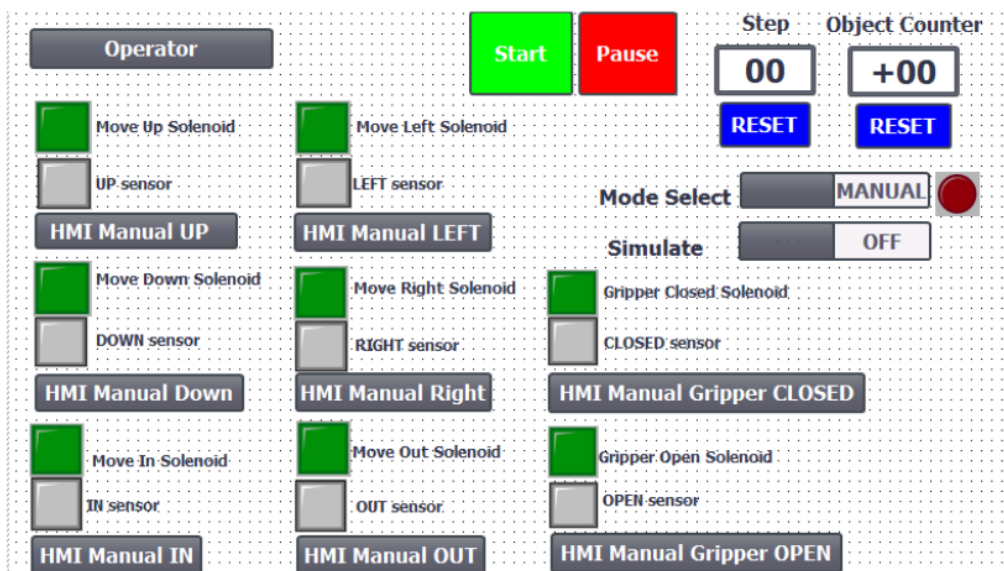


Figure 2: Image of Engineers screen, with indication of PLC tags.

Good comments, well explained. The problem with putting the network number in the comment is that if you were to insert a network (which is often the case), you would have to edit all the comments as they would not match up with the actual network number anymore.

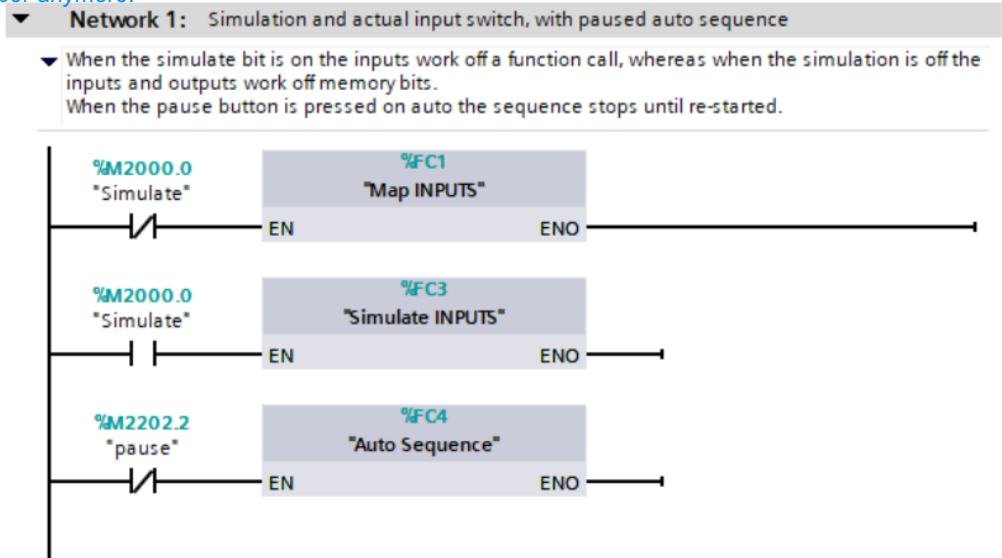


Figure 3: Inputs and paused sequence.

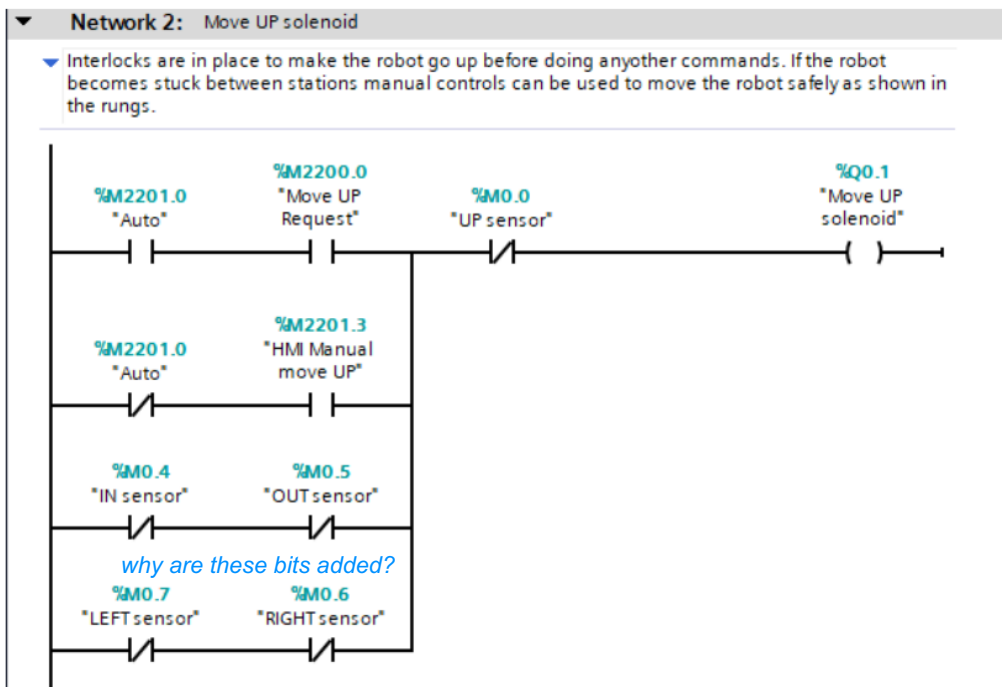


Figure 4: Safety interlocks for moving "up" with additional comments.

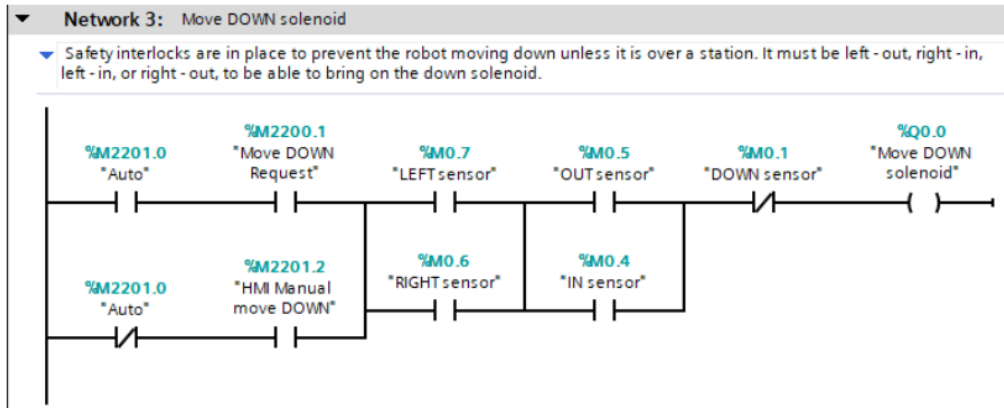


Figure 5: Safety interlocks for moving "down" with additional comments.

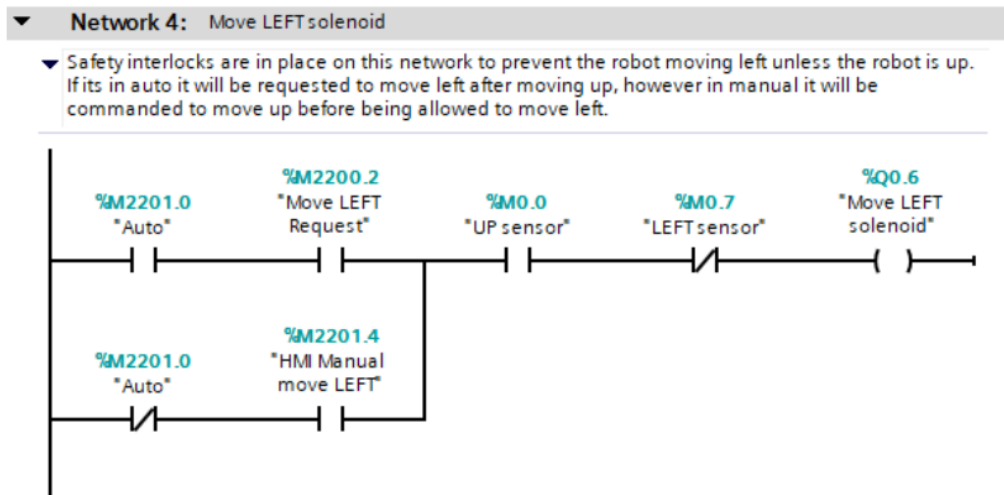


Figure 6: Safety interlocks for moving "left" with additional comments.

▼ **Network 5:** Move RIGHT solenoid

- ▼ Network 5 has safety interlocks to prevent the robot from moving right unless the robot is up. When in manual it must move up before being allowed to move right, whereas in auto it will be requested for the robot to move up before being able to move right.

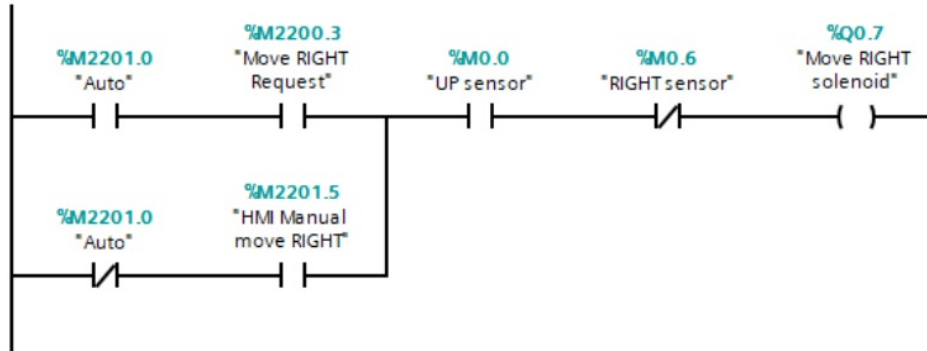


Figure 7: Safety interlocks for moving "right" with additional comments.

▼ **Network 6:** Move IN solenoid

- ▼ Network 6 has safety interlocks to prevent the robot moving in unless it is in the up position. The robot when in auto will be requested to move up before moving in to enable the solenoid, whereas when in manual the robot will be commanded to move up before being able to energise the solenoid.

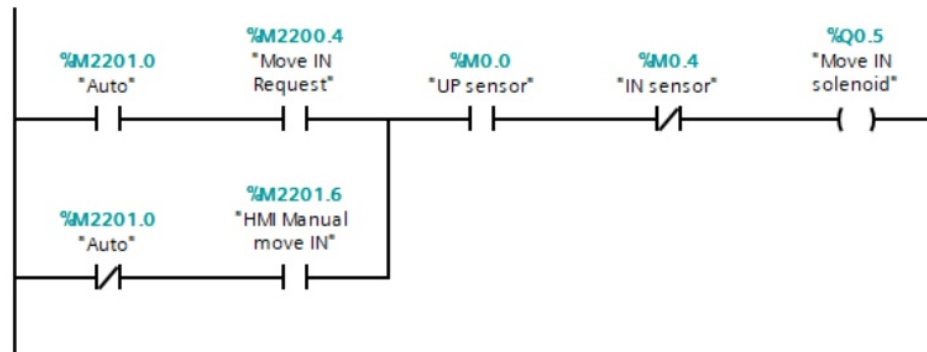


Figure 8: Safety interlocks for moving "in" with additional comments.

Network 7: Move OUT solenoid

- Network 7 has safety interlocks to prevent the robot moving out unless it is in the up position. For the solenoid to energise the up sensor must be met before the auto out request can be allowed, and when in manual the up sensor must be met before the out solenoid can be energised.

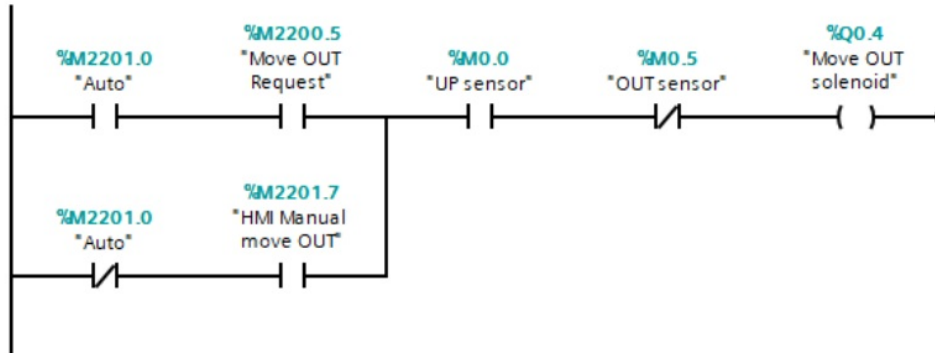


Figure 9: Safety interlocks for moving "out" with additional comments.

Network 8: Gripper OPEN solenoid

- When in auto the gripper will be requested to open when it is at its turn of the sequence, whereas in manual it is commanded to open by energising the solenoid. This is because if its stuck in auto it can be manually overridden.

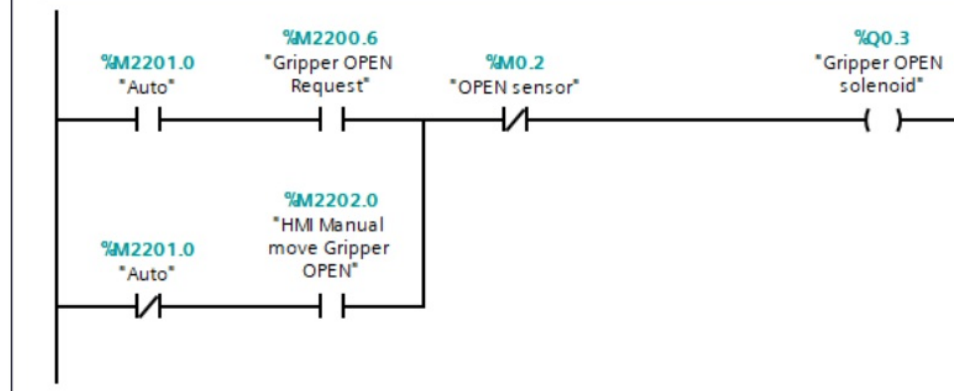


Figure 10: Network rungs for "gripper open" with additional comments.

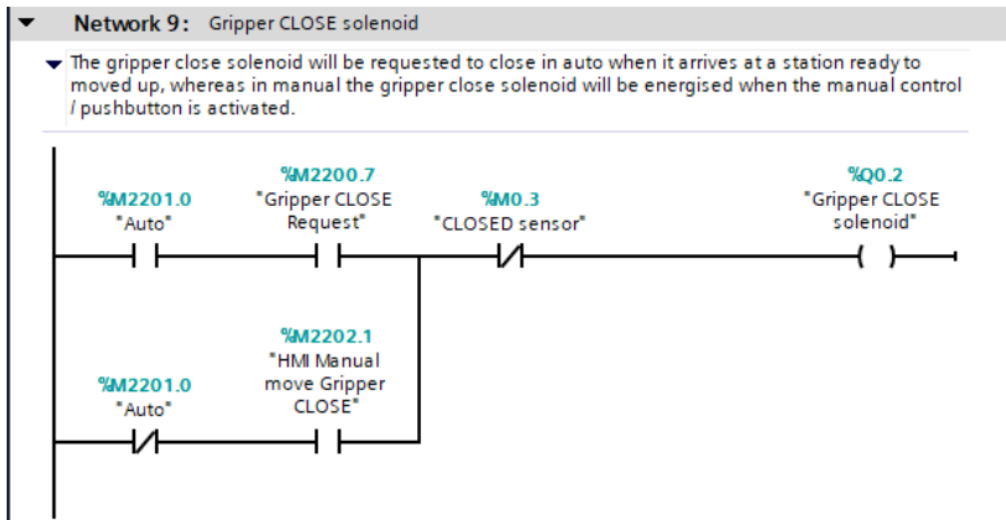


Figure 11: Network rungs for "grripper close" with additional comments.

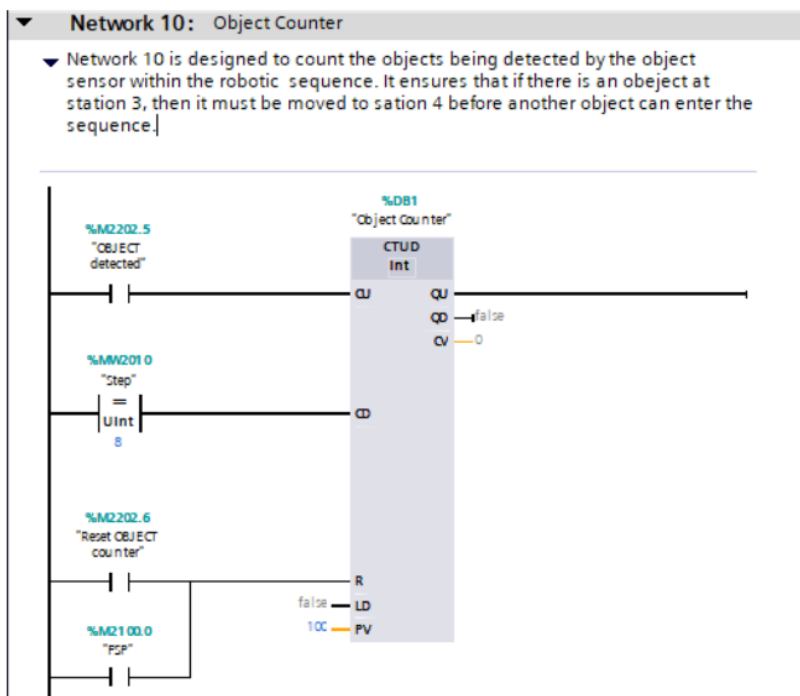


Figure 12: Object counter network with additional comments.

Network 11: Object Timer

- Network 11 is responsible for the object timer. It must delay movement of the robot by 2 seconds when an object is first rolled in. Once it has counted 2 seconds of detecting the object, the sequen...

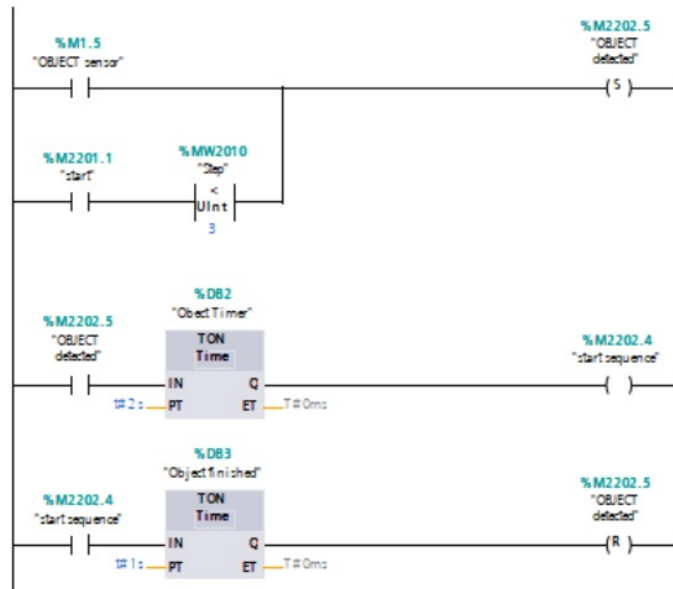


Figure 13: Object timer network with additional comments.

```

1 IF ("FSP" = 1) OR ("Auto" = 0) OR ("reset" = 1) THEN
2   "Step" := 0;
3   "Next_Step" := 0;
4   "Move IN Request" := 0;
5   "Move OUT Request" := 0;
6   "Move LEFT Request" := 0;
7   "Move RIGHT Request" := 0;
8   "Gripper OPEN Request" := 0;
9   "Gripper CLOSE Request" := 0;
10  END_IF;
11

```

Figure 14: SCL Auto sequence program FSP.

```

13 CASE "Step" OF
14   0: //move UP                                     //This step indicates the robot moving up because of the safety interlock
15     "Move UP Request" := 1;
16   IF "UP sensor" = 1 THEN
17     "Move UP Request" := 0;
18     "Next_Step" := 1;                               //Next step is 1
19   END_IF;

```

Figure 15: SCL Auto sequence step 0 - move UP with additional comments.

```

20 1: //move home //Indication of the robot moving to its home position
21 "Move IN Request" := 1;
22 "Move RIGHT Request" := 1;
23 "Gripper OPEN Request" := 1;
24 IF "IN sensor" = 1 AND
25     "RIGHT sensor" = 1 AND
26     "OPEN sensor" = 1 AND
27     "start sequence" = 1 THEN
28     "Move IN Request" := 0;
29     "Move RIGHT Request" := 0;
30     "Gripper OPEN Request" := 0;
31 IF "Object Counter".CV = 1 THEN //If the object counter is at station 1 then move to step 20
32     "Next_Step" := 20;
33 END_IF;
34 IF "Object Counter".CV = 2 THEN //If the object counter is at station 2 then move to step 10
35     "Next_Step" := 10;
36 END_IF;
37 IF "Object Counter".CV = 3 THEN //If the object counter is at station 3 then move to step 2
38     "Next_Step" := 2;
39 END_IF;
40 END_IF;

```

Figure 16: SCL Auto sequence step 1 - with additional comments.

```

41 2: //move object from station 3 to station 4 //This step shows the object moving out from station 3 to 4
42 //move OUT
43 "Move OUT Request" := 1;
44 IF "OUT sensor" = 1 THEN
45     "Move OUT Request" := 0;
46     "Next_Step" := 3; //Next step is 3
47 END_IF;
48 3: //move DOWN //This step indicates the robot is moving down
49 "Move DOWN Request" := 1;
50 IF "DOWN sensor" = 1 THEN
51     "Move DOWN Request" := 0;
52     "Next_Step" := 4; //Next step is 4
53 END_IF;
54 4: //gripper CLOSE //The gripper closes on the object
55 "Gripper CLOSE Request" := 1;
56 IF "CLOSED sensor" = 1 THEN
57     "Gripper CLOSE Request" := 0;
58     "Next_Step" := 5; //Next step is 5
59 END_IF;

```

Figure 17: SCL Auto sequence step 2, 3 & 4 – with additional comments.

```

60 5: //move UP //The gripper is now moving into the up position
61 "Move UP Request" := 1;
62 IF "UP sensor" = 1 THEN
63     "Move UP Request" := 0;
64     "Next_Step" := 6; //Next step is 6
65 END_IF;
66 6: //move IN //The gripper is moving in
67 "Move IN Request" := 1;
68 IF "IN sensor" = 1 THEN
69     "Move IN Request" := 0;
70     "Next_Step" := 7; //Next step is 7
71 END_IF;
72 7: //move DOWN //The gripper moving down to station 4
73 "Move DOWN Request" := 1;
74 IF "DOWN sensor" = 1 THEN
75     "Move DOWN Request" := 0;
76     "Next_Step" := 8; //Next step is 8
77 END_IF;

```

Figure 18: SCL Auto sequence steps 5,6 & 7 - with additional comments.

```

78 8: //gripper OPEN //The gripper opening to release the object
79 "Gripper OPEN Request" := 1;
80 IF "OPEN sensor" = 1 THEN
81 "Gripper OPEN Request" := 0;
82 "Next_Step" := 9; //Next step is 9
83 END_IF;
84 9: //move UP //After releasing the object the gripper moves up
85 "Move UP Request" := 1;
86 IF "UP sensor" = 1 THEN
87 "Move UP Request" := 0;
88 "Next_Step" := 10; //Next step is 10
89 END_IF;
90 10: //move object from station 2 to station 3 //The robot will move left,moving an object from station 2 to 3
91 //move LEFT
92 "Move LEFT Request" := 1;
93 IF "LEFT sensor" = 1 THEN
94 "Move LEFT Request" := 0;
95 "Next_Step" := 11; //Next step is 11
96 END_IF;

```

Figure 19: SCL Auto sequence steps 8, 9 & 10 - with additional comments.

```

97 11: //move OUT //The robot is moving out
98 "Move OUT Request" := 1;
99 IF "OUT sensor" = 1 THEN
100 "Move OUT Request" := 0;
101 "Next_Step" := 12; //Next step is 12
102 END_IF;
103 12: //move DOWN //The robot has moved out and is now moving down
104 "Move DOWN Request" := 1;
105 IF "DOWN sensor" = 1 THEN
106 "Move DOWN Request" := 0;
107 "Next_Step" := 13; //Next step is 13
108 END_IF;
109 13: //gripper CLOSE //Once moved down, the gripper closes on the object
110 "Gripper CLOSE Request" := 1;
111 IF "CLOSED sensor" = 1 THEN
112 "Gripper CLOSE Request" := 0;
113 "Next_Step" := 14; //Next step is 14
114 END_IF;

```

Figure 20: SCL Auto sequence steps 11, 12 & 13 - with additional comments.

```

115 14: // move UP //After closing the robot moves to its up position
116 "Move UP Request" := 1;
117 IF "UP sensor" = 1 THEN
118 "Move UP Request" := 0;
119 "Next_Step" := 15; //Next step is 15
120 END_IF;
121 15: //move RIGHT //Robot moving right to station 3
122 "Move RIGHT Request" := 1;
123 IF "RIGHT sensor" = 1 THEN
124 "Move RIGHT Request" := 0;
125 "Next_Step" := 16; //Next step is 16
126 END_IF;
127 16: //move DOWN //Once right the robot moves down
128 "Move DOWN Request" := 1;
129 IF "DOWN sensor" = 1 THEN
130 "Move DOWN Request" := 0;
131 "Next_Step" := 17; //Next step is 17
132 END_IF;

```

Figure 21: SCL Auto sequence steps 14, 15 & 16 - with additional comments.

```

133 | 17: //gripper OPEN                                //Once down the gripper opens releasing the object
134 |     "Gripper OPEN Request" := 1;
135 | IF "OPEN sensor" = 1 THEN
136 |     "Gripper OPEN Request" := 0;
137 |     "Next_Step" := 18;                                //Next step is 18
138 | END_IF;
139 | 18: //move UP                                    //Robot moving into the up position
140 |     "Move UP Request" := 1;
141 | IF "UP sensor" = 1 THEN
142 |     "Move UP Request" := 0;
143 |     "Next_Step" := 19;                                //Next step is 19
144 | END_IF;
145 | 19: //move IN                                    //Robot moving from the out porstion to the in position
146 |     "Move IN Request" := 1;
147 | IF "IN sensor" = 1 THEN
148 |     "Move IN Request" := 0;
149 |     "Next_Step" := 20;                                //Next step is 20
150 | END_IF;

```

Figure 22: SCL Auto sequence steps 17, 18 & 19 - with additional comments.

```

151 | 20: //move object from station 1 to station 2      //Moving object from station 1 to 2 after moving left
152 |     //move LEFT
153 |     "Move LEFT Request" := 1;
154 | IF "LEFT sensor" = 1 THEN
155 |     "Move LEFT Request" := 0;
156 |     "Next_Step" := 21;                                //Next step is 21
157 | END_IF;
158 | 21: //move DOWN                                //Once left the gripper goes down
159 |     "Move DOWN Request" := 1;
160 | IF "DOWN sensor" = 1 THEN
161 |     "Move DOWN Request" := 0;
162 |     "Next_Step" := 22;                                //Next step is 22
163 | END_IF;
164 | 22: //gripper CLOSE                            //Gripper closes on the object
165 |     "Gripper CLOSE Request" := 1;
166 | IF "CLOSED sensor" = 1 THEN
167 |     "Gripper CLOSE Request" := 0;
168 |     "Next_Step" := 23;                                //Next step is 23
169 | END_IF;

```

Figure 23: SCL Auto sequence steps 20, 21 & 22 - with additional comments.

```

170 | 23: //move UP                                    //Gripper moves into the up postion
171 |     "Move UP Request" := 1;
172 | IF "UP sensor" = 1 THEN
173 |     "Move UP Request" := 0;
174 |     "Next_Step" := 24;                                //Next step is 24
175 | END_IF;
176 | 24: //move OUT                                //Once up the gripper moves out
177 |     "Move OUT Request" := 1;
178 | IF "OUT sensor" = 1 THEN
179 |     "Move OUT Request" := 0;
180 |     "Next_Step" := 25;                                //Next step is 25
181 | END_IF;
182 | 25: //move DOWN                                //Gripper moves into the down postion
183 |     "Move DOWN Request" := 1;
184 | IF "DOWN sensor" = 1 THEN
185 |     "Move DOWN Request" := 0;
186 |     "Next_Step" := 26;                                //Next step is 26
187 | END_IF;

```

Figure 24: SCL Auto sequence steps 23, 24 & 25 - with additional comments.

```

188 | 26: //gripper OPEN                                //Gripper opens releasing the object
189 |     "Gripper OPEN Request" := 1;
190 | IF "OPEN sensor" = 1 THEN
191 |     "Gripper OPEN Request" := 0;
192 |     "Next_Step" := 0;                                //Next step is 0
193 | END_IF;
194 |
195 | END_CASE;
196 | "Step" := "Next_Step";                                //Indicates the gripper returns to next step which is home
197 |

```

Figure 25: SCL Auto sequence step 26 with additional comments.

1		UP sensor	Simulation Tags	Bool	%M0.0
2		DOWN sensor	Simulation Tags	Bool	%M0.1
3		OPEN sensor	Simulation Tags	Bool	%M0.2
4		CLOSED sensor	Simulation Tags	Bool	%M0.3
5		IN sensor	Simulation Tags	Bool	%M0.4
6		OUT sensor	Simulation Tags	Bool	%M0.5
7		RIGHT sensor	Simulation Tags	Bool	%M0.6
8		LEFT sensor	Simulation Tags	Bool	%M0.7
9		X10 Pushbutton	Simulation Tags	Bool	%M1.0
10		X11 Pushbutton	Simulation Tags	Bool	%M1.1
11		X12 Pushbutton	Simulation Tags	Bool	%M1.2
12		X13 toggle switch	Simulation Tags	Bool	%M1.3
13		X14 toggle switch	Simulation Tags	Bool	%M1.4
14		OBJECT sensor	Simulation Tags	Bool	%M1.5
15		Simulate	Assignment tag table	Bool	%M2000.0
16		PLC Base Unit input byte 0	Assignment tag table	Byte	%IB0
17		PLC Base Unit mapped input by..	Assignment tag table	Byte	%MB0
18		PLC Base Unit input byte 1	Assignment tag table	Byte	%IB1
19		PLC Base Unit mapped input by..	Assignment tag table	Byte	%MB1

Figure 26: PLC tag table.

20		System_Byte	System Tags	Byte	%MB2100
21		FSP	System Tags	Bool	%M2100.0
22		DiagStatusUpdate	System Tags	Bool	%M2100.1
23		Always TRUE	System Tags	Bool	%M2100.2
24		Always FALSE	System Tags	Bool	%M2100.3
25		Clock_Byte	System Tags	Byte	%MB2101
26		Clock_10Hz	System Tags	Bool	%M2101.0
27		Clock_5Hz	System Tags	Bool	%M2101.1
28		Clock_2.5Hz	System Tags	Bool	%M2101.2
29		Clock_2Hz	System Tags	Bool	%M2101.3
30		Clock_1.25Hz	System Tags	Bool	%M2101.4
31		Clock_1Hz	System Tags	Bool	%M2101.5
32		Clock_0.625Hz	System Tags	Bool	%M2101.6
33		Clock_0.5Hz	System Tags	Bool	%M2101.7
34		Move UP solenoid	Assignment tag table	Bool	%Q0.1
35		Move DOWN solenoid	Assignment tag table	Bool	%Q0.0
36		Move LEFT solenoid	Assignment tag table	Bool	%Q0.6
37		Move RIGHT solenoid	Assignment tag table	Bool	%Q0.7
38		Move IN solenoid	Assignment tag table	Bool	%Q0.5

Figure 27: PLC tag table

39		Move OUT solenoid	Assignment tag table Bool	%Q0.4
40		Gripper OPEN solenoid	Assignment tag table Bool	%Q0.3
41		Gripper CLOSE solenoid	Assignment tag table Bool	%Q0.2
42		Step	Assignment tag table UInt	%MW2010
43		Next_Step	Assignment tag table UInt	%MW2012
44		Move UP Request	Assignment tag table Bool	%M2200.0
45		Move DOWN Request	Assignment tag table Bool	%M2200.1
46		Move LEFT Request	Assignment tag table Bool	%M2200.2
47		Move RIGHT Request	Assignment tag table Bool	%M2200.3
48		Move IN Request	Assignment tag table Bool	%M2200.4
49		Move OUT Request	Assignment tag table Bool	%M2200.5
50		Gripper OPEN Request	Assignment tag table Bool	%M2200.6
51		Gripper CLOSE Request	Assignment tag table Bool	%M2200.7
52		Auto	Assignment tag table Bool	%M2201.0
53		start	Assignment tag table Bool	%M2201.1
54		HMI Manual move DOWN	Assignment tag table Bool	%M2201.2
55		HMI Manual move UP	Assignment tag table Bool	%M2201.3
56		HMI Manual move LEFT	Assignment tag table Bool	%M2201.4
57		HMI Manual move RIGHT	Assignment tag table Bool	%M2201.5

Figure 28: PLC tag table

57		HMI Manual move RIGHT	Assignment tag table Bool	%M2201.5
58		HMI Manual move IN	Assignment tag table Bool	%M2201.6
59		HMI Manual move OUT	Assignment tag table Bool	%M2201.7
60		HMI Manual move Gripper OPEN	Assignment tag table Bool	%M2202.0
61		HMI Manual move Gripper CLO...	Assignment tag table Bool	%M2202.1
62		pause	Assignment tag table Bool	%M2202.2
63		reset	Assignment tag table Bool	%M2202.3
64		start sequence	Assignment tag table Bool	%M2202.4
65		OBJECT detected	Assignment tag table Bool	%M2202.5
66		Reset OBJECT counter	Assignment tag table Bool	%M2202.6

Figure 29: PLC tag table.

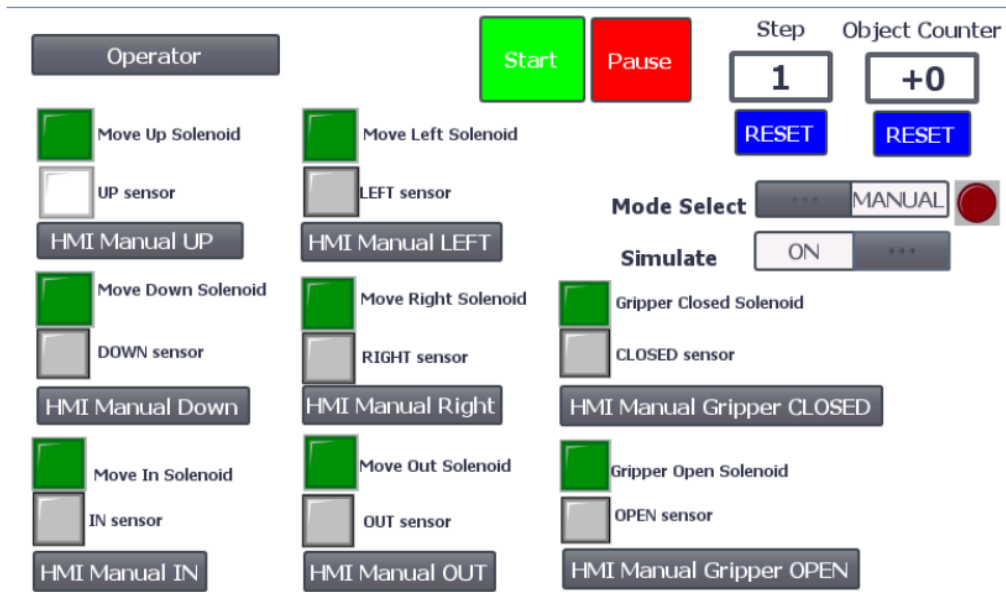


Figure 30: Engineer screen up interlock.

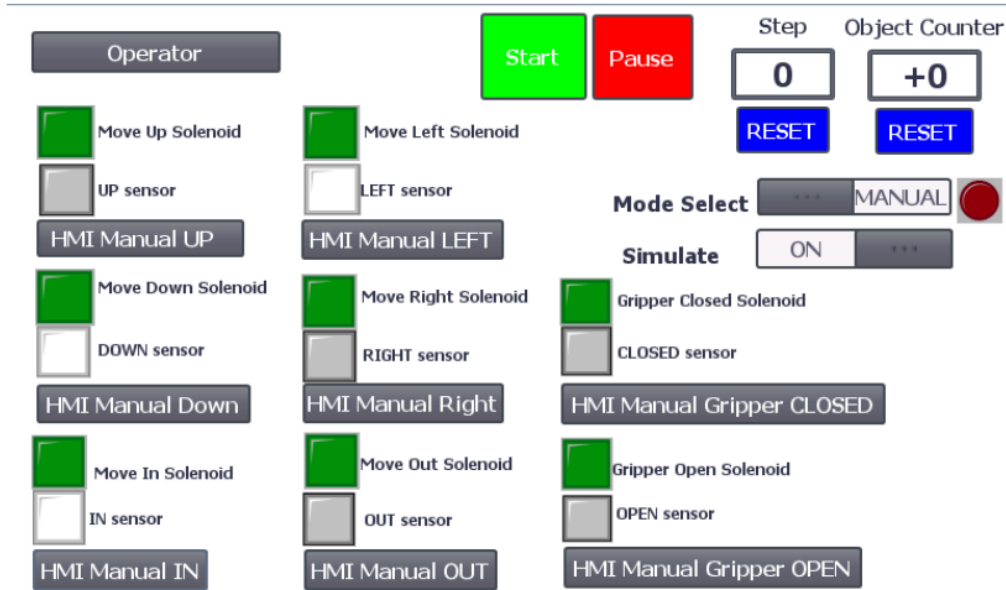


Figure 31: Testing down interlock.

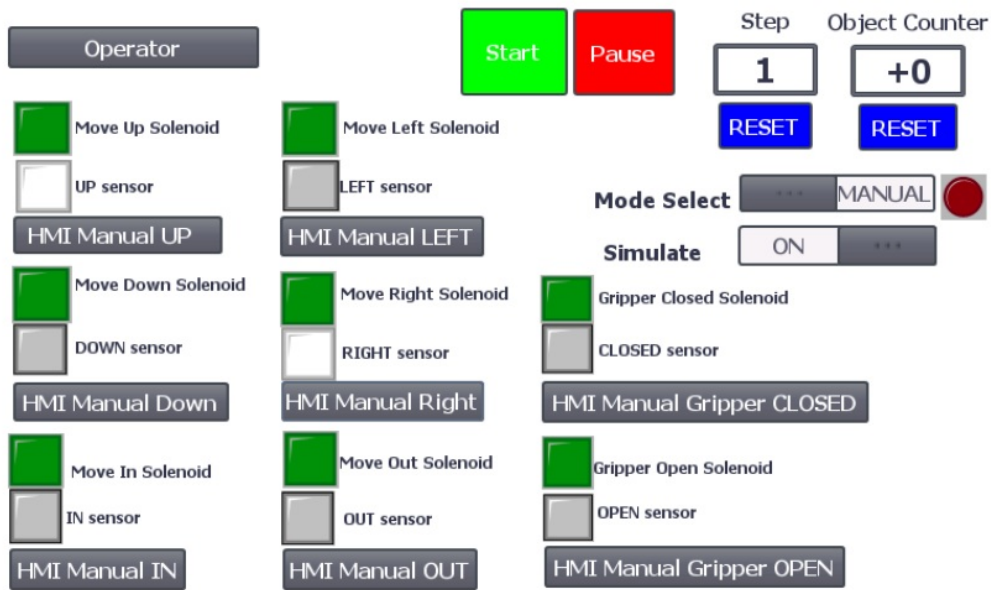


Figure 32: Testing right interlock.

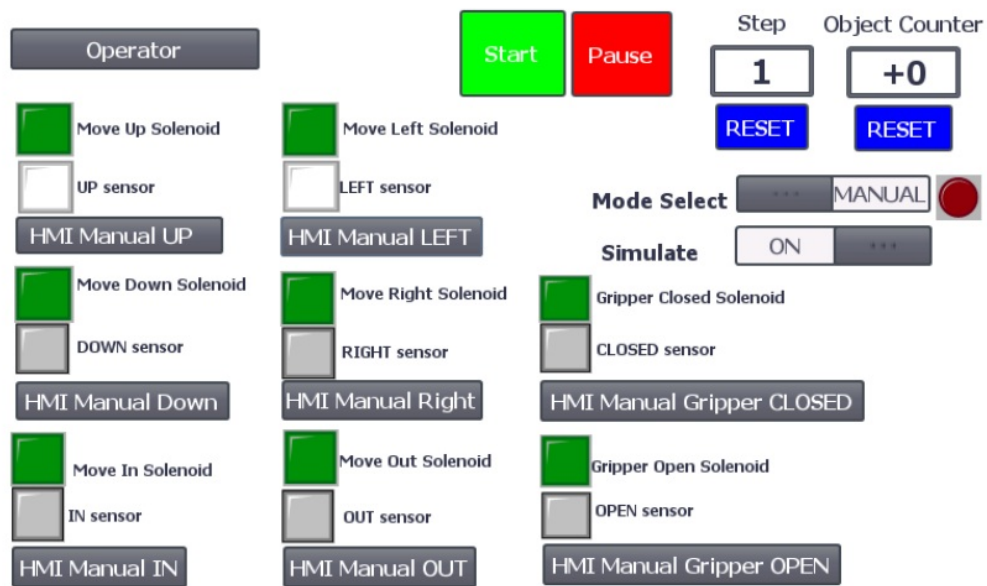


Figure 33: Testing left interlock.

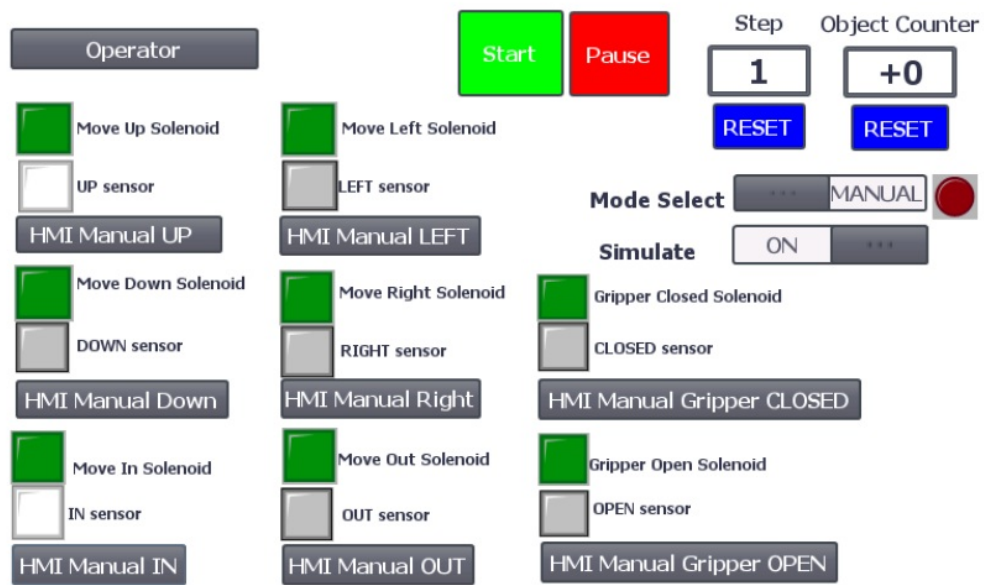


Figure 34: Testing in interlock.

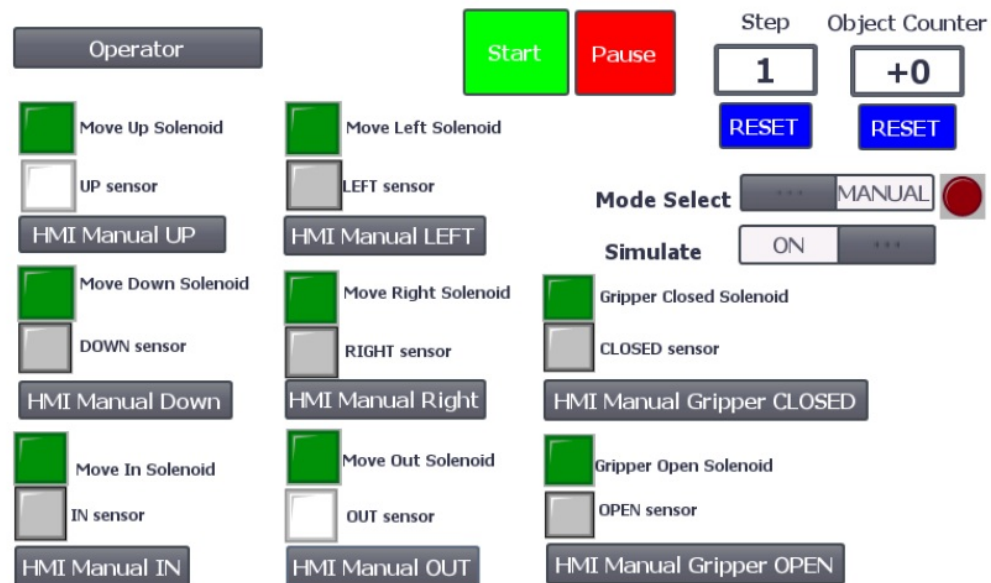


Figure 35: Testing out interlock.

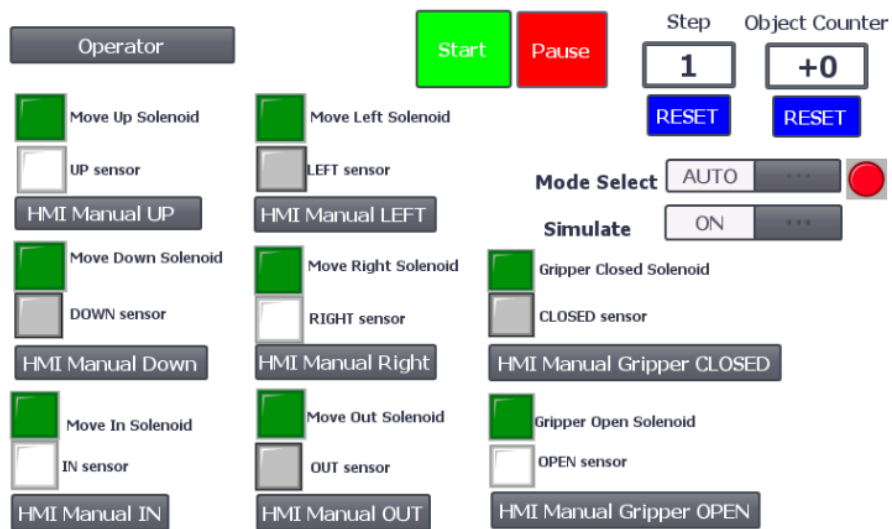


Figure 36: Testing auto sequence step 1 – counter 1.

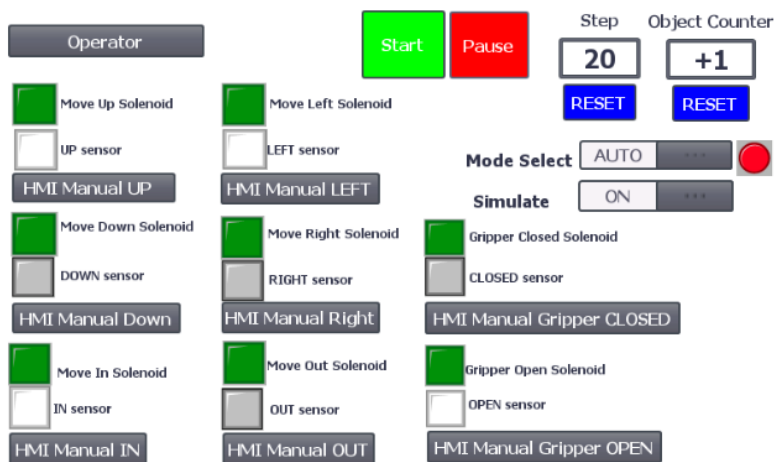


Figure 37: Testing auto sequence step 20 - counter 1.

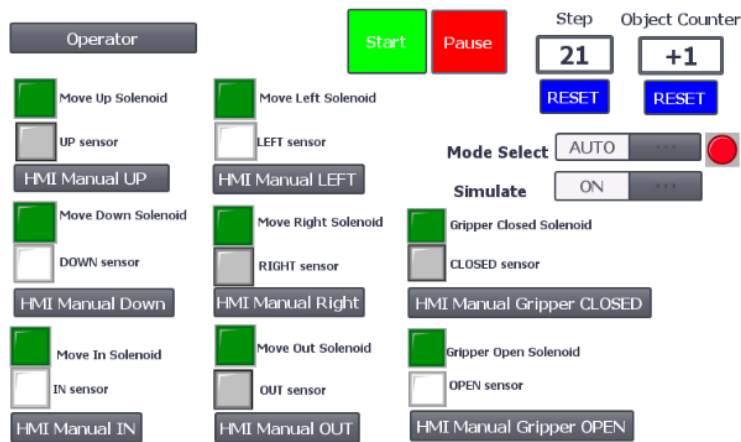


Figure 38: Testing auto sequence step 21 - counter 1.

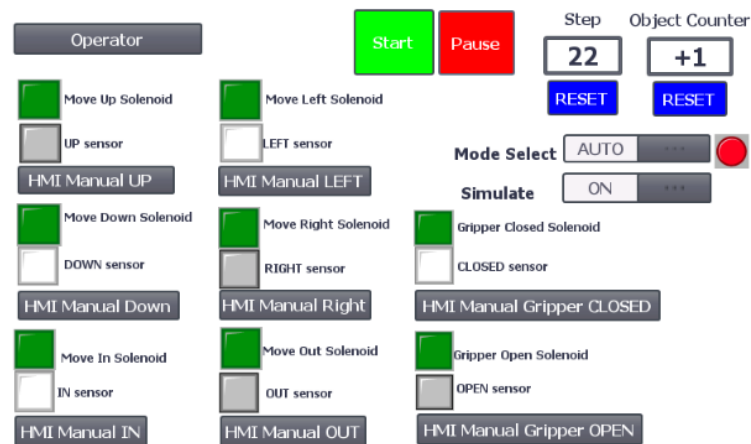


Figure 39: Testing auto sequence step 22 - counter 1.

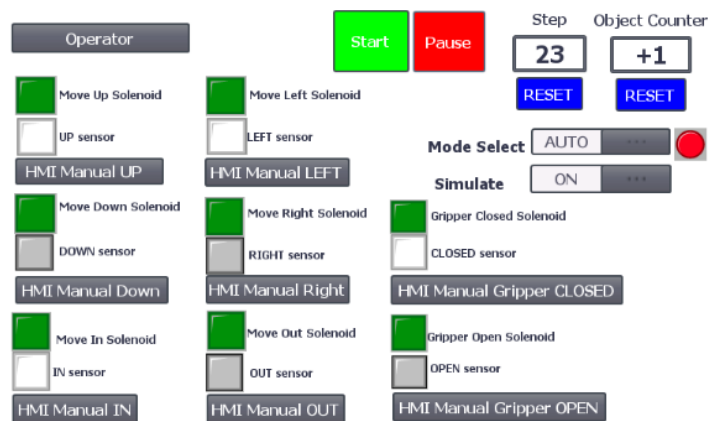


Figure 40: Testing auto sequence step 23 - counter 1.

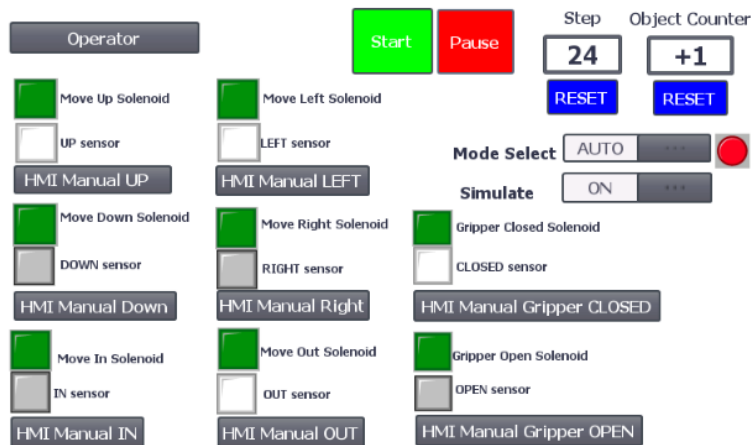


Figure 41: Testing auto sequence step 24 - counter 1.

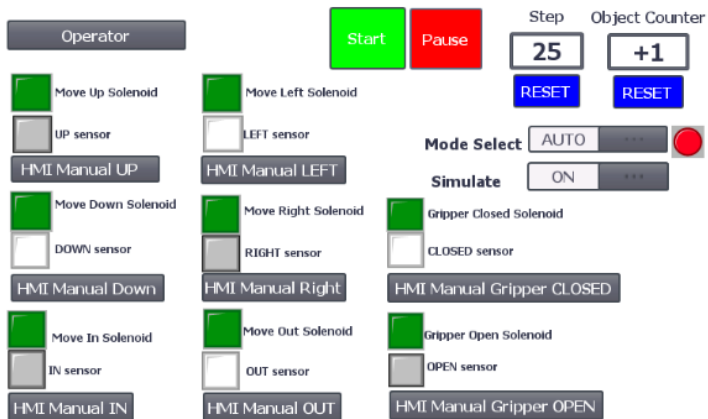


Figure 42: Testing auto sequence step 25 - counter 1.

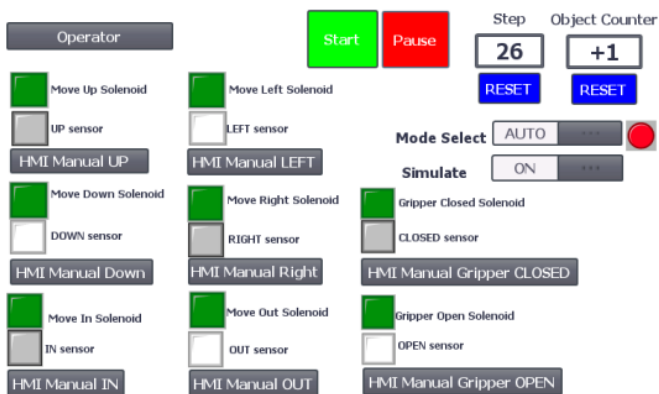


Figure 43: Testing auto sequence step 26 - counter 1.

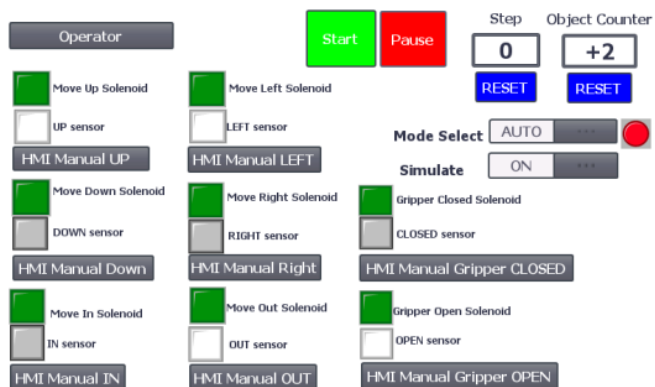


Figure 44: Testing auto sequence step 0 - counter 2.

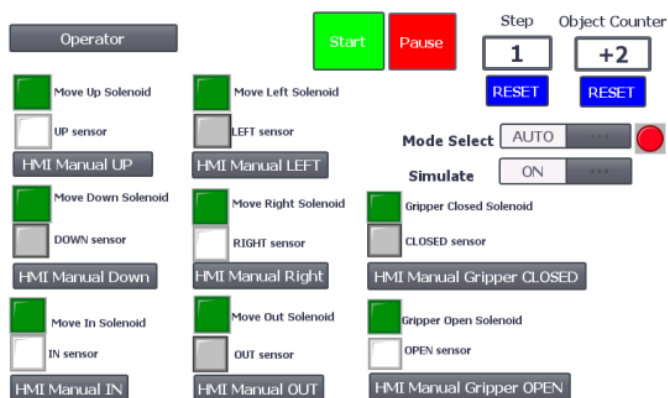


Figure 45: Testing auto sequence step 1 - counter 2.

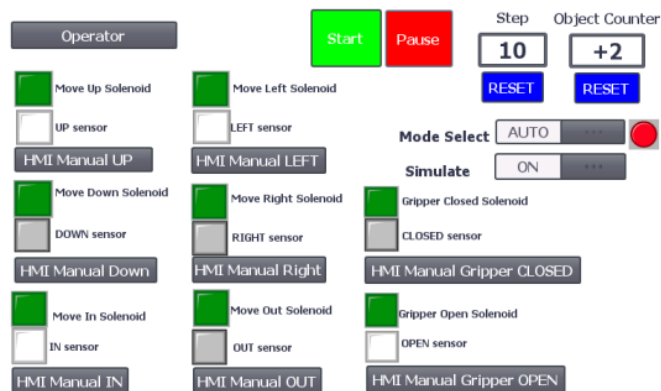


Figure 46: Testing auto sequence step 10 - counter 2.

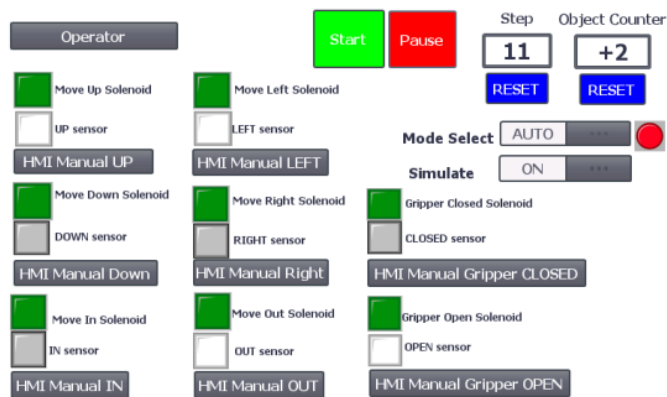


Figure 47: Testing auto sequence step 11 - counter 2.

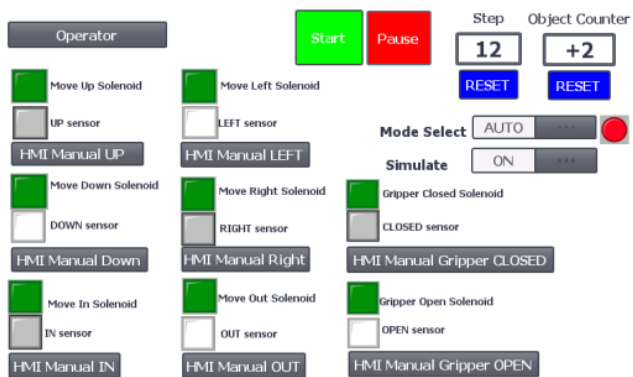


Figure 48: Testing auto sequence step 12 - counter 2.

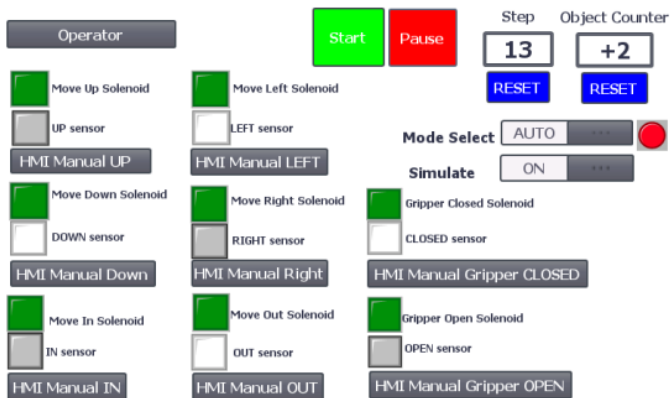


Figure 49: Testing auto sequence step 13 - counter 2.

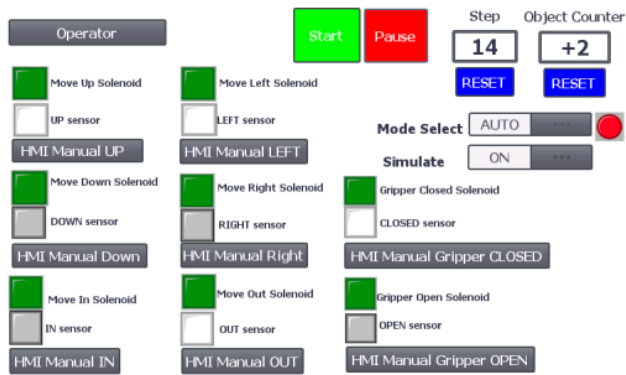


Figure 50: Testing auto sequence step 14 - counter 2.

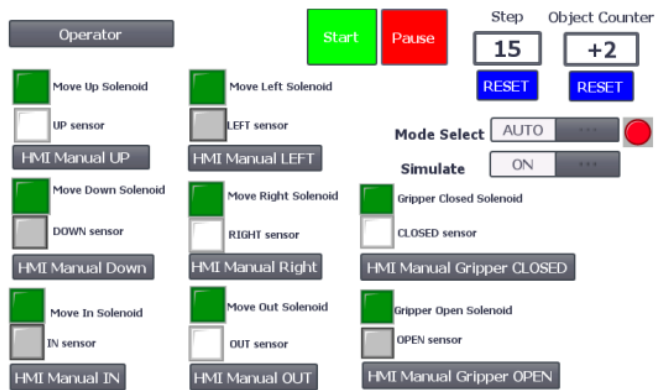


Figure 51: Testing auto sequence step 15 - counter 2.

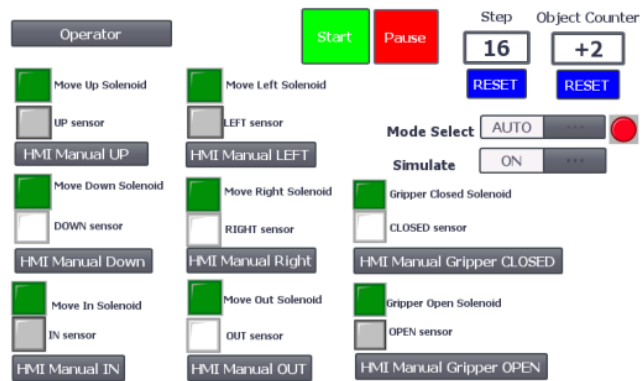


Figure 52: Testing auto sequence step 16 - counter 2.

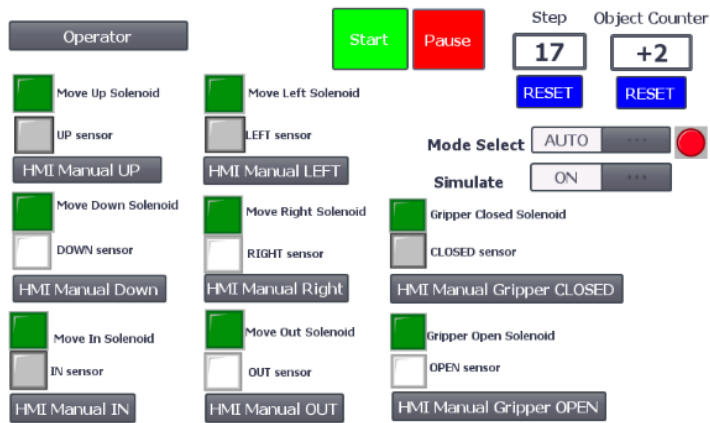


Figure 53: Testing auto sequence step 17 - counter 2.

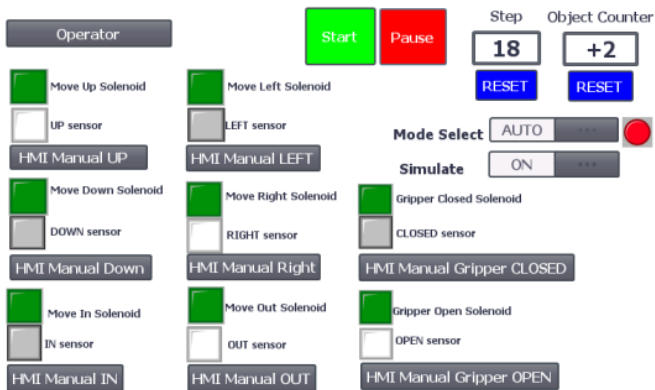


Figure 54: Testing auto sequence step 18 - counter 2.

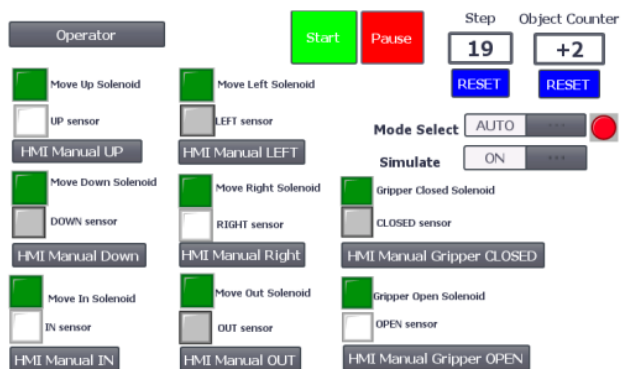


Figure 55: Testing auto sequence step 19 - counter 2.

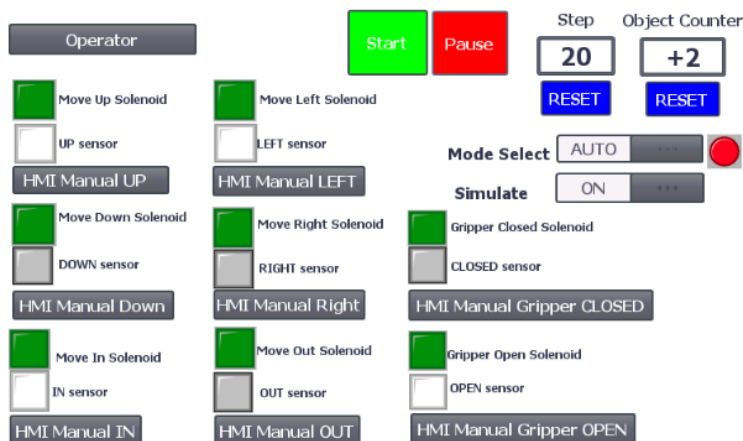


Figure 56: Testing auto sequence step 20 - counter 2.

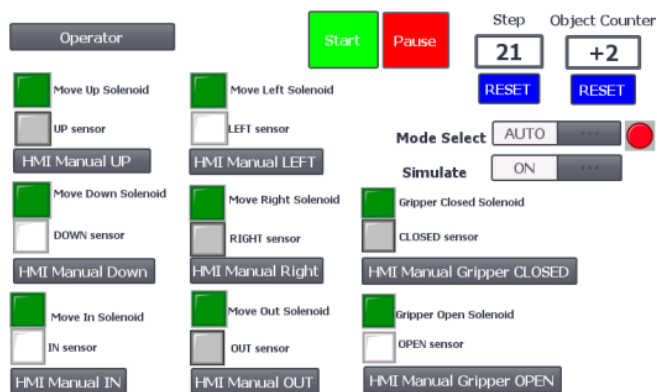


Figure 57: Testing auto sequence step 21 - counter 2.

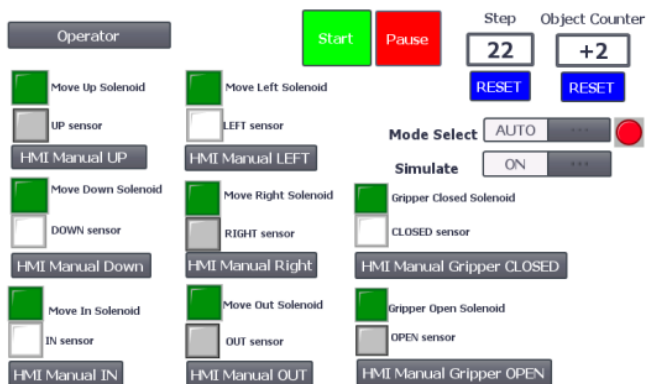


Figure 58: Testing auto sequence step 22 - counter 2.

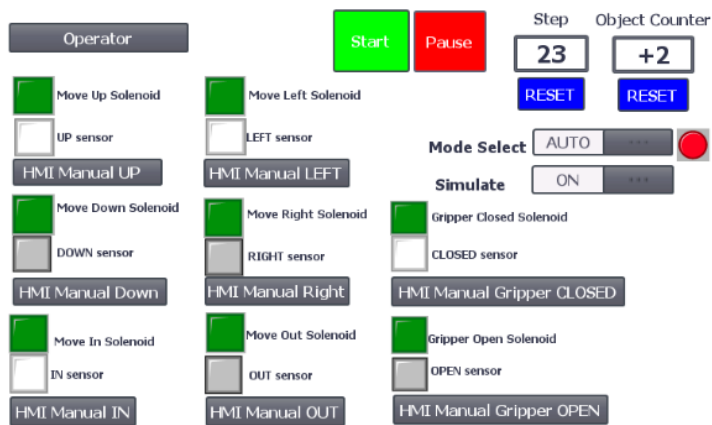


Figure 59: Testing auto sequence step 23 - counter 2.

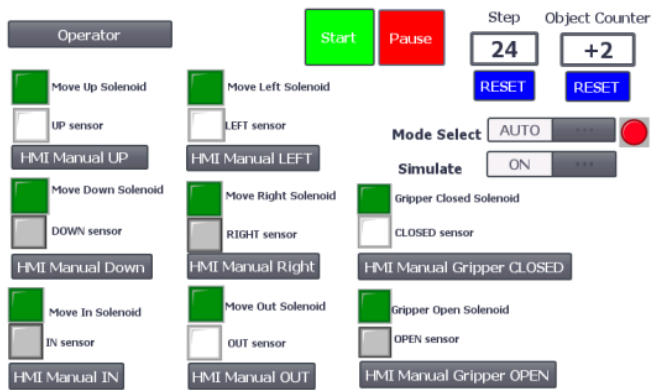


Figure 60: Testing auto sequence step 24 - counter 2.

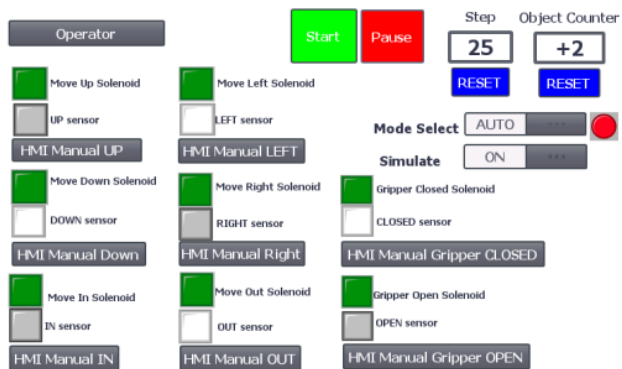


Figure 61: Testing auto sequence step 25 - counter 2.

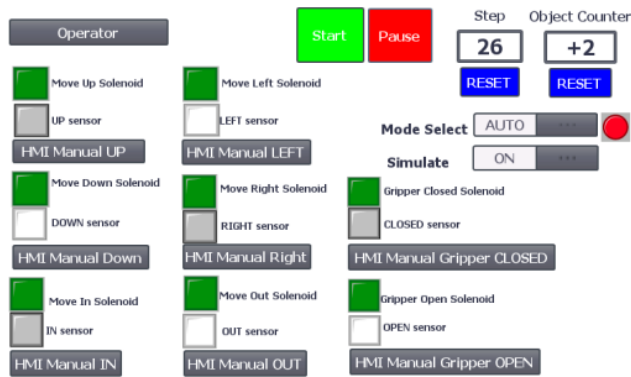


Figure 62: Testing auto sequence step 26 - counter 2.

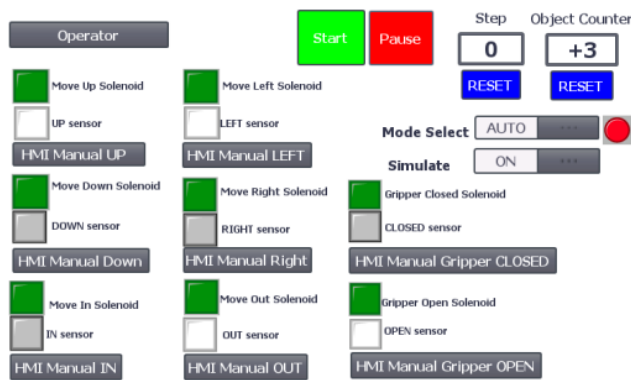


Figure 63: Testing auto sequence step 0 - counter 3.

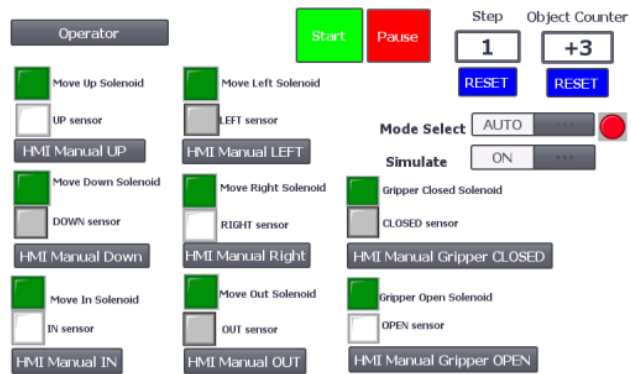


Figure 64: Testing auto sequence step 1 - counter 3.

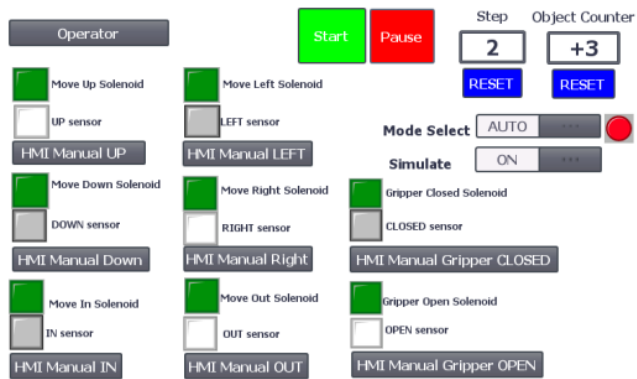


Figure 65: Testing auto sequence step 2 - counter 3.

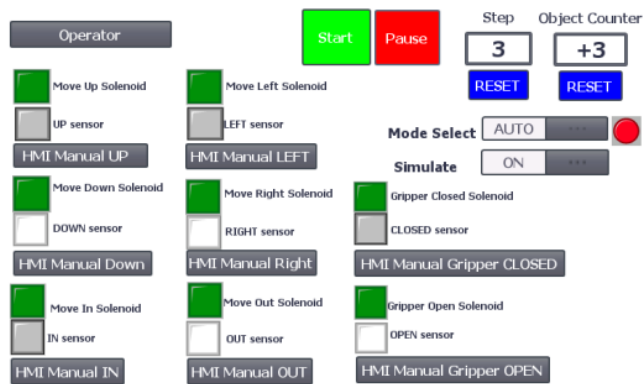


Figure 66: Testing auto sequence step 3 - counter 3.

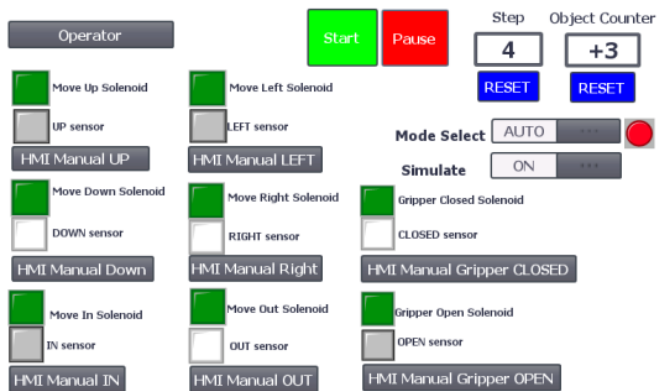


Figure 67: Testing auto sequence step 4 - counter 3.

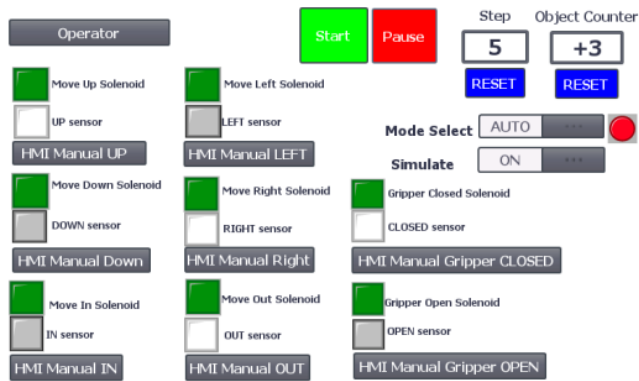


Figure 68: Testing auto sequence step 5 - counter 3.

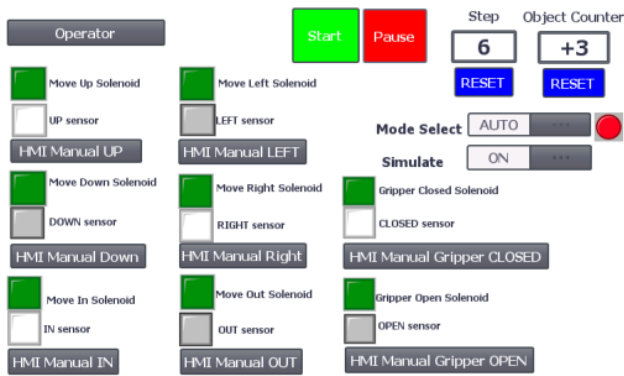


Figure 69: Testing auto sequence step 6 - counter 3.

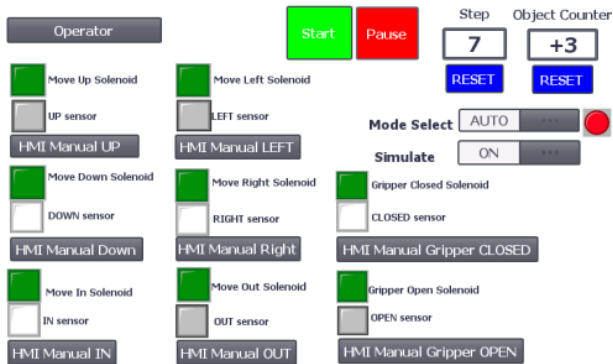
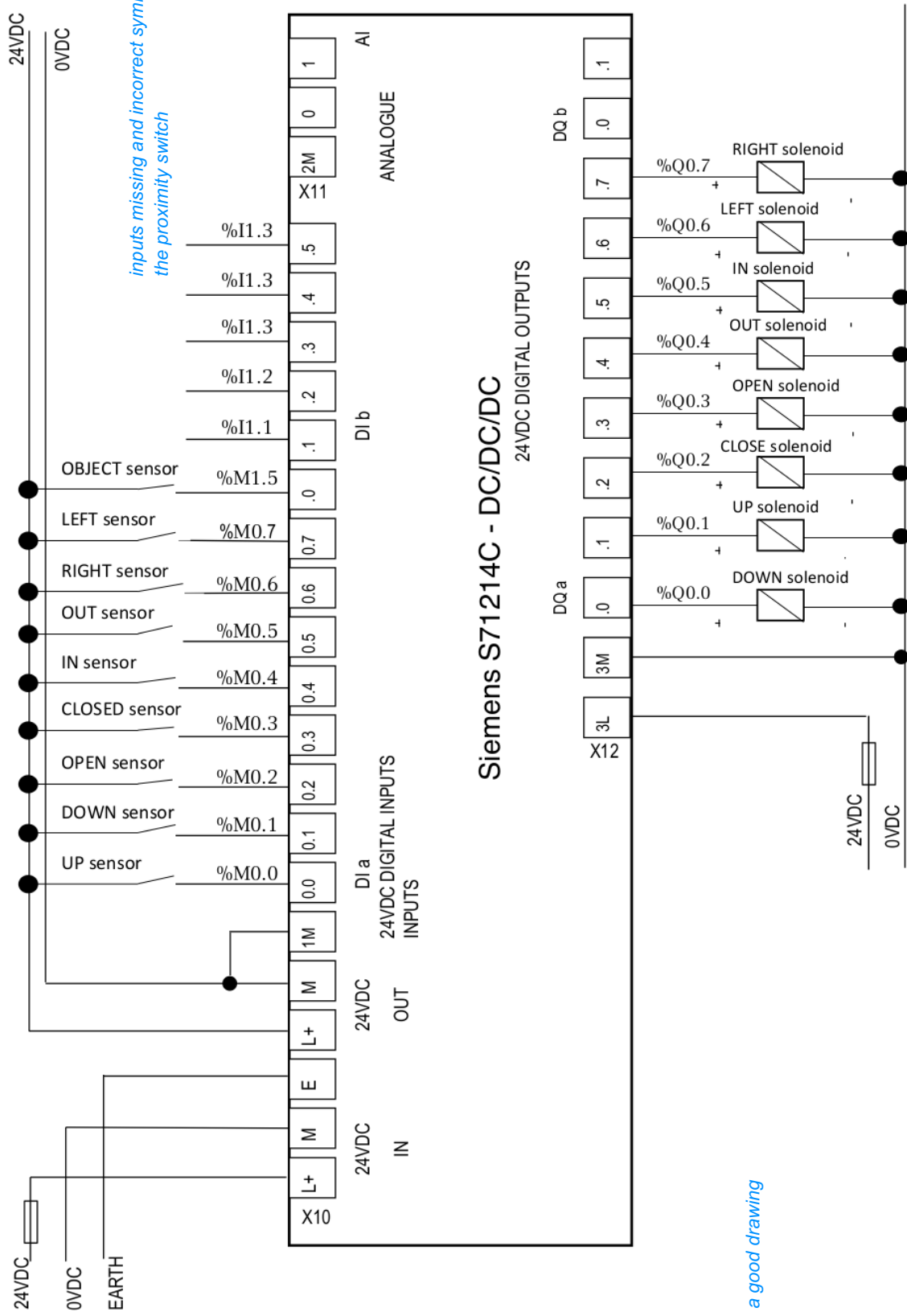


Figure 70: Testing auto sequence step 7 - counter 3.



inputs missing and incorrect symbol used for the proximity switch

a good drawing

Figure 71: PLC drawing

FINAL GRADE

GENERAL COMMENTS

Instructor

62/100

PAGE 1

PAGE 2

QM

Good Structure

Report is structured exactly as stated in the assignment brief.

PAGE 3

QM

Suppress the author's name if you have already stated it.

You don't need the author's name since your sentence began with it, just suppress it to satisfy the fully Harvard method

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

PAGE 4

QM

Suppress the author's name if you have already stated it.

You don't need the author's name since your sentence began with it, just suppress it to satisfy the fully Harvard method

QM

Use the third Person

You are referring to first or second persons (I, we, you, us etc) whereas you should use the third person thus depersonalising it.

Text Comment. Good conclusion but space is a limitation

PAGE 5

Text Comment. You should have given the I address here not the M address

PAGE 6

PAGE 7

QM

Explain the system or design, don't tell a story of how it was made

You are wasting words telling your reader a story of how you made the system or program. Focus instead the finished system and explain that instead.

Text Comment. Unnecessary waste of words

PAGE 8

Text Comment. don't use boxes for labels, switch off the outline next time you label a drawing using a text box

Text Comment. Start P/B is not a sequence action, it is an event

Text Comment. you could have included the object sensor as an alternative event

PAGE 9

PAGE 10

PAGE 11

Text Comment. return to step 0

PAGE 12

PAGE 13

Text Comment. 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. Some are included.

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 14

PAGE 15

PAGE 16

PAGE 17



Correct method of listing.

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

Text Comment. A small range of source given. For more marks, you need a lot more sources than this to show evidence of wide reading.

PAGE 18

PAGE 19

Text Comment. Good comments, well explained. The problem with putting the network number in the comment is that if you were to insert a network (which is often the case), you would have to edit all the comments as they would not match up with the actual network number anymore.

Text Comment. why are these bits added?

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

Text Comment. inputs missing and incorrect symbol used for the proximity switch

Text Comment. a good drawing

GRADING FORM: NEW CSG-RUBRIC

RYAN
JAMES

62

1. ENGAGEMENT WITH LITERATURE

- | | | |
|---|--|----------|
|  | Evidence provided of research from a small range of sources listed but not always cited correctly using the Harvard method. (8/20) | 8 |
|---|--|----------|

2. KNOWLEDGE AND UNDERSTANDING

- | | | |
|---|--|-----------|
|  | Good evidence of knowledge and understanding provided through system explanation and program comments. (15/20) | 15 |
|---|--|-----------|

3. COGNITIVE AND INTELLECTUAL

- | | | |
|---|--|-----------|
|  | Clear evidence of these skills displayed in testing, discussion , conclusions and recommendations. (15/20) | 15 |
|---|--|-----------|

4. PRACTICAL APPLICATION

- | | | |
|---|---|-----------|
|  | A well designed program, well comments. However, see my comments on your SFC and drawing. Well presented results but see my comments on your table. (14/20) | 14 |
|---|---|-----------|

5. TRANSFERABLE SKILLS

- | | | |
|---|--|-----------|
|  | Use the third person, never refer to I, we, you in your report. Instead of "I did this" say, "this was done". Well structured but word count exceeded though not | 10 |
|---|--|-----------|

declared so. (10/20)