

1702304

by Benjamin Morgan

FILE	19736_BENJAMIN_MORGAN_1702304_178848_224166469.DOCX (4.05M)		
TIME SUBMITTED	07-FEB-2021 12:31PM (UTC+0000)	WORD COUNT	3034
SUBMISSION ID	144329264	CHARACTER COUNT	14944

Course: HNC Electrical and Electronic Engineering

Module: SGEE4004 Programmable Logic Controllers

Assignment: 1. Pick and Place Robot

Name: Benjamin Morgan

University number: 1702304

Date submitted: 08/01/2021

Word count: 2544

Table of Contents

Good Structure

1. Research.....	3
2. Hardware Investigation, Circuit Diagram, and I/O List	6
3. Program Design and Explanation	8
4. Simulation Testing.....	10
5. Practical Testing and Team Working.....	12
6. Discussion, Conclusions and Recommendations	13
7. Citation List	14
Annexe	15

For a direct quotation, you must say - Siemens states that " " (2020, page ?).

This is done by suppressing the author and including the page number.

1. Research

Siemens offer a wide range of PLC modules; each have a different specification and fit different use cases. The main branch of their PLC's is the S7 range, there is the S7-1200 and S7-1500 range. The 1200 range is a basic range controller that is the "smart choice for compact automation systems" (Siemens, 2020). The 1500 range is more advanced and powerful and is capable of "automating whole plants and other applications that require better performance, flexibility and networking" (Siemens, 2020). They also have distributed controller options which perform the same as the other PLC's but do so in a compact form factor that are able to be situated away from the process, should it be in a hazardous area.

Both the 1200 and 1500 range of PLC's have different models that are able to perform at different levels, the main difference between the PLC models is how many inputs and outputs the CPU can handle. The most basic 1200 series CPU, the '1211C', is capable of operating 6 digital inputs and 4 digital outputs without needing an expansion card, whereas the most basic 1500 series CPU '1511-1PN' requires separate I/O modules to be purchased with it, the smallest digital input module can handle 16 inputs and the smallest output module can handle 8 digital outputs. There are different modules for analogue I/O and fail safe I/O. Due to the size of the pick and place robot and how many inputs and outputs there are, the S7-1500 range can be ruled out as the optimal PLC as it is too advanced and due to the fact the input and output modules have to be purchased separately it means it would not be cost effective for this application. The cost for a 'starter pack' for a S7-1500 system using the basic 1511-1PN CPU is £1461 (Parmley Graham, 2020). This includes all of the required add-ons to make the system the work, such as the CPU, the mounting DIN rail, the front connectors, the power supply and a 4MB SIMATIC memory card. Comparatively a 'starter pack' using the 1212C costs £413 (Parmley Graham, 2020). For these reasons, only the S7-1200 range will be investigated further.

The 1200 range has 5 different CPU options, these are: 1211C, 1212C, 1214C, 1215C and the 1217C. Each of these CPU's other than the 1217C come in 3 variations: DC/DC/DC, DC/DC/RLY and, AC/DC/RLY, the 1217C is only available in a DC/DC/DC configuration. The first section of each type describes the power supply for the CPU (24V DC or 120/230V AC), the middle section describes the operating voltage of the

onboard inputs (24V DC) and the last section describes what type of digital out put it supports (DC = 24V DC, RLY = up to 30V DC or 250V AC with relay) (Berger, 2013). The main differences between the CPU's are how many integrated I/O there are, the memory size and how many expansion modules they are compatible with. Below is a table showing some of the differences between the CPUs.

	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
User memory					
Internal load memory	1MB	2MB	4MB	4MB	4MB
Work memory	50KB	75KB	100KB	125KB	150KB
Integrated I/Os					
Digital inputs (DI)	6 DI, 24V DC	8 DI, 24V DC	14 DI, 24V DC	14 DI, 24V DC	14 DI, 24V DC
Digital outputs (DO)	4 DO, 24V DC or relay	6 DO, 24V DC or relay	10 DO, 24V DC or relay	10 DO, 24V DC or relay	10 DO, 24V DC or relay
Analog inputs (AI)	N/A	2 AI (10 bit)	2 AI (10 bit)	2 AI (10 bit)	2 AI (10 bit)
Analog outputs (AO)	N/A	N/A	N/A	2 AO (10 bit)	2 AO (10 bit)
Counters					
	3 fast (100kHz)	4 fast (3 w/ 100kHz, 1 w/ 30kHz)	6 fast (3 w/ 100kHz, 3 w/ 30kHz)	6 fast (3 w/ 100kHz, 3 w/ 30kHz)	6 fast (1Mhz)
Expansion with					
A board (SB, CB, BB*)	1	1	1	1	1
Signal Modules (SM)	None	2	8	8	8
Communication modules (CM)	None	3	3	3	3

*(SB = signal board, CB = Communication board, BB = battery board) (Siemens, 2020) Correctly cited

All of the CPU's have some degree of expansion, this can be through signal boards, communication boards, battery boards, signal modules and communication modules. The signal boards, communication boards and battery boards are modules that add to the PLC without changing the dimensions of the board, these are useful in cases where space is limited. Whereas the modules are separate modules that are mounted alongside the CPU. Both the signal board and the signal modules are used to add I/O to the CPU, they can add

DI, AI, DO, AO, or a combination of DI/DO. The main difference between them is how much I/O they can add. The signal boards offer less expandability as they are smaller. The communication board expands the communication connections of the CPU whereas the module supports the CPU with communication tasks. The module will become responsible for data transport, establishing connections, and providing the necessary communication services for the CPU and program. The battery board is used to extend the run time of the real-time clock (Berger, 2013). Now the CPU's can be compared, the pick and place robot being worked on has 14 digital inputs and 6 digital outputs. Straight away the 1211C and 1212C can be ruled out as they cannot handle this much I/O without the use of an expansion card. If a 1214C or above CPU were to be used this would allow some expansion in the future, all of the CPU's can handle the same amount of I/O without requiring an expansion card. The 3 higher end CPU's all have very similar specifications, the differences are only in the memory and speed of the counters. Due to this fact the cost will be the deciding factor between the CPU's, the 1214C CPU can be bought for £369, the 1215C for £556 and the 1217C for £803. Also, a 1212C costs £244, although it does not have enough integrated I/O, an I/O signal module with 8 DI and 8 DO can be purchased for £170, this brings the total to £414, this configuration would give you slightly more I/O than any of the other CPU's at a lower cost.

From looking at the technical data and the cost implications, the 1214C appears to be the best CPU for this application, it has the correct amount of integrated I/O without the need for an expansion card and is the best value for this instance. While the 1212C with the added signal module is a viable and cost-effective option, it would only be best if the system were to be expanded slightly. The higher end 1215C and 1217C CPU's are too powerful and have features that are not required for this application. These are better suited to a different application rather than the pick and place robot.

Correct conclusion

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

The PLC used for this project was the S7 1214C (DC/DC/DC), this PLC can handle all the I/O without any expansion. This was linked to a 7 inch HMI screen; this was used to control the robot. There are 2 screens within the HMI, one for the operator and one for the engineer. These have buttons on screen that control the robot and lights that indicate the status of the sensors.

The robot system consists of multiple inputs and outputs, all of the I/O used are digital. The majority of the inputs are limit switches, specifically reed switches. Reed switches are switches that close when a magnetic force acts upon them. These were used as the limit switches to alert the PLC when the robot has moved as far as required. A reed switch is made up of 2 small, magnetic, springy wires, these 2 wires will come together when a magnetic force is applied. This will make the contact and complete the circuit. When the magnetic force is removed, the springy wires will return to their original shape and the circuit will be broken. Another input used was an inductive proximity switch. This was used to detect when an object rolls down the chute into the system. Inductive sensors work by detecting changes in the magnetic field produced by the sensor, caused by a metallic object passing in front of them. There were also 5 physical switches connected to the PLC, although these were not used.

The outputs for the robot system are memory solenoids, responsible for moving the robot. They are pneumatically operated solenoids, the term 'memory' denotes that the solenoids will remain in the same position once the air supply is removed. The solenoids act in 2 directions, the movement depends upon what inlet the air is sent. Each solenoid was responsible for each movement axis and control of the gripper, resulting in a total of 4.

The I/O list for this system is below:

Inputs		Outputs	
Component	PLC Address	Component	PLC Address
UP sensor	%I0.0	DOWN solenoid	%Q0.0
DOWN sensor	%I0.1	UP solenoid	%Q0.1
OPEN sensor	%I0.2	CLOSE gripper solenoid	%Q0.2

CLOSED sensor	%I0.3	OPEN gripper solenoid	%Q0.3
IN sensor	%I0.4	OUT solenoid	%Q0.4
OUT sensor	%I0.5	IN solenoid	%Q0.5
RIGHT sensor	%I0.6	LEFT solenoid	%Q0.6
LEFT sensor	%I0.7	RIGHT solenoid	%Q0.7
Object sensor	%I1.0		
X10 push button	%I1.1		
X11 push button	%I1.2		
X12 push button	%I1.3		
X13 toggle switch	%I1.4		
X14 toggle switch	%I1.5		

Address of last 6 inputs are incorrect

The wiring diagram for the PLC is in the annexe.

3. Program Design and Explanation

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

Now the hardware has been identified, and an I/O list created, the program can be written. A 'skeleton' program was provided, this contained function calls that are responsible for mapping the physical inputs to memory bits within the PLC (FC1), and for simulating the inputs and outputs (FC3). Memory bits are internal connections within the PLC software that do not require wiring.

The first step was to create the full tag list required for the system. This included the memory bits that were used for the on-screen buttons on the HMI, the step and object counters and move requests. The move requests are used instead of the solenoid outputs to ensure the robot does not move when it is not meant to. Each move request has an auto and manual variant. Memory bits were created that were responsible for deciding between real or simulated inputs and for choosing between auto and manual modes. The full tag list can be seen in the annexe.

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

Now the manual controls can be programmed. The manual program was written in the main organisational block (OB1) and was programmed in ladder. Each network was responsible for a single direction of movement; therefore, the manual controls consist of 8 networks. The brief gives safety instructions that must be incorporated to the program. These were implemented through the use of interlocks. The interlocks ensure a signal is only sent to the solenoid when the conditions are safe. The interlocks programmed here will work for both the manual controls and the auto sequence. Once the manual program had been written, the engineers HMI screen could be made. This screen has buttons that are linked with the move requests and lights that indicate if the solenoid is in use and when the limits have been met. OB1 has been documented fully and can be seen in the annexe alongside the HMI screens. The selection between simulated/real inputs and auto/manual mode, the logic for object detection and the object counter are also located within OB1. The start push button on the HMI screen will be used to simulate an object rolling down the chute.

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

Next the auto sequence had to be designed, the first step of this process was to read the brief and create a flow chart that contained every step required. The flow chart showed that 26 steps were required to be able to move the object between all 4 stations. The flow chart is structured so that the command is linked with a step, and then the criteria for the next step to start is linked between the steps. The auto sequence was written in a function call (FC2)

and was programmed in SCL, this is a text based programming language. The auto sequence contains many IF statements, these ensure that commands are only given when the described criteria is met. Initially the reset conditions were written, these were used to reset any move requests and step numbers whenever the program is switched from auto, the reset button is pressed, or the first scan pulse occurs. Next the auto sequence was written. The first step, step '0' ensures that the robot is in the UP position, this is done so that the robot cannot collide with anything before it moves to the home position. At the end of each step the associated move request(s) are reset. Step 1 is responsible for moving the robot to the home position, this being: UP, IN, RIGHT, grippers OPEN. The robot will remain in this step until all of the corresponding limits are made, and the start button on the HMI is pressed. Once these criteria have been met, the program will identify how many objects are within the system. It will then jump to the correct step required. The auto sequence will run after a 2 second delay from when an object is detected.

4. Simulation Testing

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

Once the program had been built, simulation testing was carried out. Firstly, the interlocks were tested and then the auto sequence was tested. To test that the interlocks are functioning correctly, a test table was produced, this described the tests required and what results are expected. All tests were performed through the engineer's HMI screen. The test table for the first set of interlock testing is below.

Interlock safety testing

Machine should not be able to move left, right, in or out unless the robot is in the up position

TEST NUMBER	TEST DESCRIPTION	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1	Attempt to move left while arm is down	No movement	No movement	Pass
2	Attempt to move right while arm is down	No movement	No movement	Pass
3	Attempt to move in while arm is down	No movement	No movement	Pass
4	Attempt to move out while arm is down	No movement	No movement	Pass
5	Attempt to move left while arm is up	Free movement	Free movement	Pass
6	Attempt to move right while arm is up	Free movement	Free movement	Pass
7	Attempt to move in while arm is up	Free movement	Free movement	Pass
8	Attempt to move out while arm is up	Free movement	Free movement	Pass

Figure 1 - Manual Interlock Test Table 1

A good testing table but you could have linked to the image evidence of the results

This table shows that the interlocks that stop the robot from moving while the arm is down are functioning as required. The results for the next set of interlock testing is below.

Interlock safety testing

Machine should not be able to move down unless it is above one of the stations

TEST NUMBER	TEST DESCRIPTION	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
1	Attempt to move down over station 1	Free movement	Free movement	Pass
2	Attempt to move down over station 2	Free movement	Free movement	Pass
3	Attempt to move down over station 3	Free movement	Free movement	Pass
4	Attempt to move down over station 4	Free movement	Free movement	Pass
5	Attempt to move down when robot is not all the way left	No movement	No movement	Pass
6	Attempt to move down when robot is not all the way right	No movement	No movement	Pass
7	Attempt to move down when robot is not all the way in	No movement	No movement	Pass
8	Attempt to move down when robot is not all the way out	No movement	No movement	Pass

Figure 2 - Manual Interlock Test Table 2

These 2 sets of tests prove that the interlocks are providing the necessary protection. This concluded the testing in manual mode, next the auto sequence was tested.

To test the auto sequence, the sequence was started and at each step the program was paused, and the robot position was recorded. The results were compared to the flow chart to ensure the robot is in the correct position for each step. The screenshots of these tests are in the annexe. The testing also checks that the program jumps to the correct step, depending on how many objects are in the system. The testing showed that the robot worked exactly as it should. The robot was in the correct position for each step, as according to the flow chart. The test results have been tabulated below.

Auto Sequence Test Table

Step Number	Expected Position	Actual Position	Pass/Fail
0	UP	UP	Pass
1	UP, IN, RIGHT, OPEN	UP, IN, RIGHT, OPEN	Pass
2	UP, OUT, RIGHT, OPEN	UP, OUT, RIGHT, OPEN	Pass
3	DOWN, OUT, RIGHT, OPEN	DOWN, OUT, RIGHT, OPEN	Pass
4	DOWN, OUT, RIGHT, CLOSED	DOWN, OUT, RIGHT, CLOSED	Pass
5	UP, OUT, RIGHT, CLOSED	UP, OUT, RIGHT, CLOSED	Pass
6	UP, IN, RIGHT, CLOSED	UP, IN, RIGHT, CLOSED	Pass
7	DOWN, IN, RIGHT, CLOSED	DOWN, IN, RIGHT, CLOSED	Pass
8	DOWN, IN, RIGHT, OPEN	DOWN, IN, RIGHT, OPEN	Pass
9	UP, IN, RIGHT, OPEN	UP, IN, RIGHT, OPEN	Pass
10	UP, OUT, RIGHT, OPEN	UP, OUT, RIGHT, OPEN	Pass
11	UP, OUT, LEFT, OPEN	UP, OUT, LEFT, OPEN	Pass
12	DOWN, OUT, LEFT, OPEN	DOWN, OUT, LEFT, OPEN	Pass
13	DOWN, OUT, LEFT, CLOSED	DOWN, OUT, LEFT, CLOSED	Pass
14	UP, OUT, LEFT, CLOSED	UP, OUT, LEFT, CLOSED	Pass
15	UP, OUT, RIGHT, CLOSED	UP, OUT, RIGHT, CLOSED	Pass
16	DOWN, OUT, RIGHT, CLOSED	DOWN, OUT, RIGHT, CLOSED	Pass
17	DOWN, OUT, RIGHT, OPEN	DOWN, OUT, RIGHT, OPEN	Pass
18	UP, OUT, RIGHT, OPEN	UP, OUT, RIGHT, OPEN	Pass
19	UP, IN, LEFT, OPEN	UP, IN, LEFT, OPEN	Pass
20	UP, IN, LEFT, OPEN	UP, IN, LEFT, OPEN	Pass
21	DOWN, IN, LEFT, OPEN	DOWN, IN, LEFT, OPEN	Pass
22	DOWN, IN, LEFT, CLOSED	DOWN, IN, LEFT, CLOSED	Pass
23	UP, IN, LEFT, CLOSED	UP, IN, LEFT, CLOSED	Pass
24	UP, OUT, LEFT, CLOSED	UP, OUT, LEFT, CLOSED	Pass
25	DOWN, OUT, LEFT, CLOSED	DOWN, OUT, LEFT, CLOSED	Pass
26	DOWN, OUT, LEFT, OPEN	DOWN, OUT, LEFT, OPEN	Pass

Figure 3 - Auto Sequence Test Table

5. Practical Testing and Team Working

Unfortunately, due to Covid-19 restrictions practical testing couldn't be conducted. Instead the methods for simulation testing were discussed within a group. The group discussed the testing criteria and the methods of testing. By comparing programs and test tables, the group confirmed that they used similar methods of testing. All members agreed that the basis of their tests were derived from the criteria in the brief. The group also took time to review the designs of their HMI screens, they commented on the intuitiveness of their designs and also how easy to understand the button/light layout was. Through peer reviewing, a member of the group was able to fix an issue regarding the object counter failing to count down once the object had been released at station 4.

6. Discussion, Conclusions and Recommendations

To conclude, through rigorous testing the pick and place program was found to work exactly to the specification, and in an efficient manner. The auto testing sequence showed that the robot followed the flow chart design, and that the object(s) were moved between stations efficiently. The program has been internally documented thoroughly. This means that if the user ever needed to perform testing or maintenance, they would easily be able to find the corresponding network(s). The HMI screens also contain the tools to stop the machine as it works, so that maintenance can occur. Throughout the design process the program was run many times, and any faults found during these tests were rectified. This resulted in a robust program that should not fail. The PLC chosen was better suited to this task than the alternatives, it had the correct amount of integrated I/O and offered good value for money. If this project was to be undertaken again, it would be best to make the engineer's HMI screen password protected. This would prevent someone who is untrained from controlling the robot when it is not safe to do so, for example, if work was being carried out in the vicinity of the robot. Improper use could also damage it. Although, this scenario should be prevented through the protection provided by the interlocks.

7. Citation List

Correct method of

writing

Good range of

examples

Berger, H., 2013. Automating with SIAMATIC S7-1200. In: *Automating with SIAMATIC S7-1200*. Erlangen: Publicis Publishing, p. 44.

Parmley Graham, 2020. *S7-1200 Starter Pack including CPU1212C ac/dc/relay, software & doc*. [Online]

Available at: <https://www.parmley-graham.co.uk/automation/siemens/plc-s/s7-1200-basic-controllers/6es72121be344yb0>

[Accessed 10 12 2020].

Parmley Graham, 2020. *Siemens S7-1200 basic controllers*. [Online]

Available at: <https://www.parmley-graham.co.uk/automation/siemens/plc-s/s7-1200-basic-controllers>

[Accessed 2 January 2020].

Parmley Graham, 2020. *SIAMATIC S7-1500 Starter Kit, comprising: CPU 1511C-1 PN, 4 MB MMC, 160 mm DIN rail, front connector, STEP 7 Professional 365-day license, PM 70 W 120/230 V AC PSU, Ethernet cable, doc..* [Online]

Available at: <https://www.parmley-graham.co.uk/automation/siemens/plc-s/s7-1500-advanced-controllers/6es75111ck034yb5>

[Accessed 9 12 2020].

Siemens, 2020. *SIAMATIC S7-1200*. [Online]

Available at: <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/s7-1200.html>

[Accessed 9 12 2020].

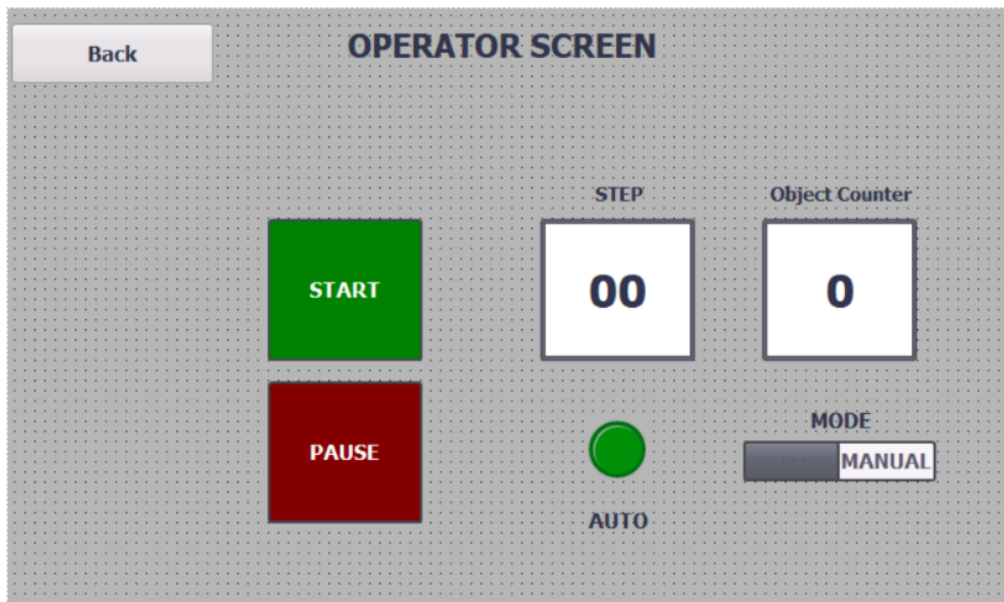
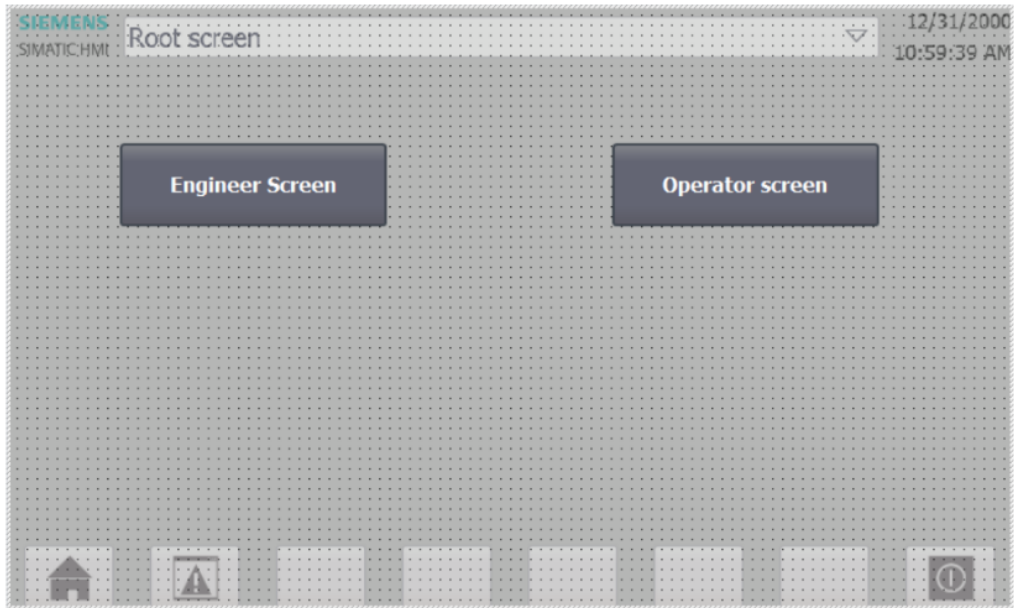
Siemens, 2020. *SIAMATIC S7-1500*. [Online]

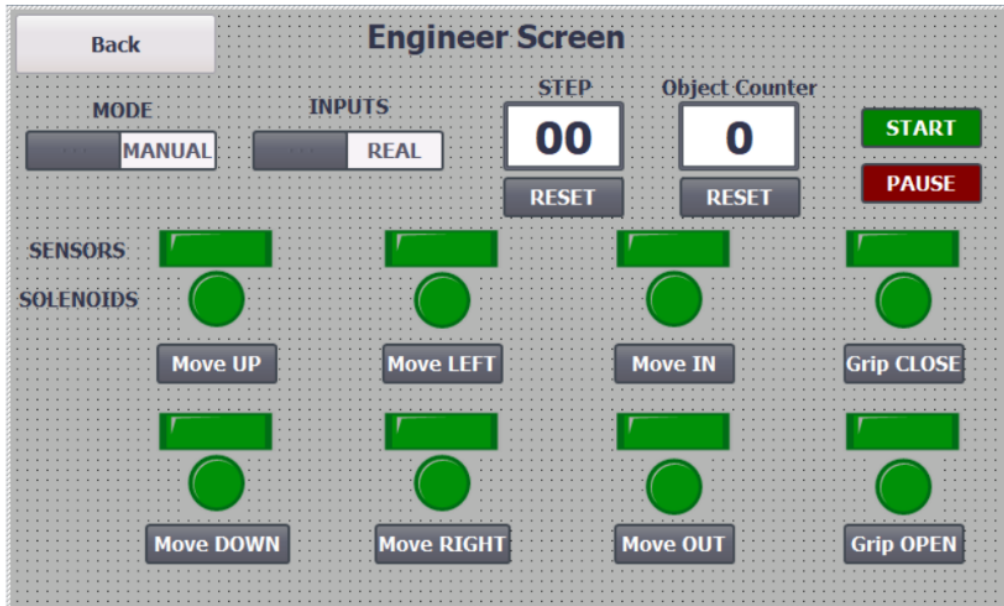
Available at:

<https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500.html>

[Accessed 9 12 2020].

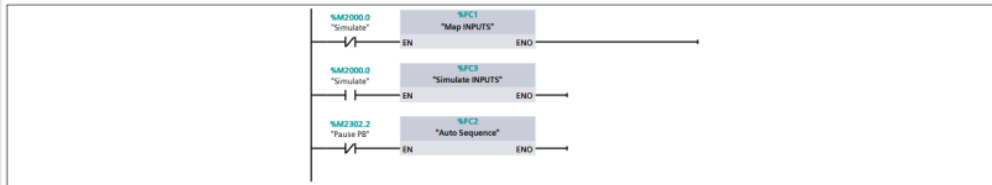
Annexe





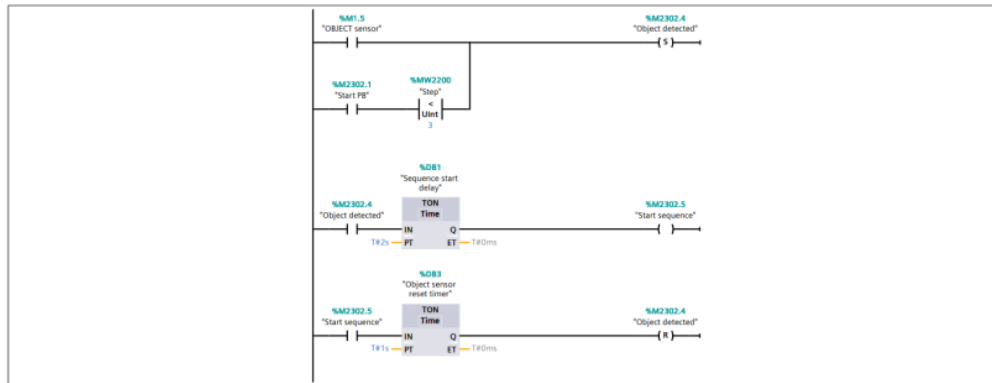
Network 1: Selector between simulated inputs or the real inputs, also has selector between auto mode or manual.

M2000.0 is a memory bit that will determine whether the inputs of the system are to be simulated or if the physical inputs are to be used. This bit will be tagged to a switch on the HMI screen so that the user can decide which mode they want. M2302.2 is used to pause the auto sequence if there is an issue, or testing needs to be done.



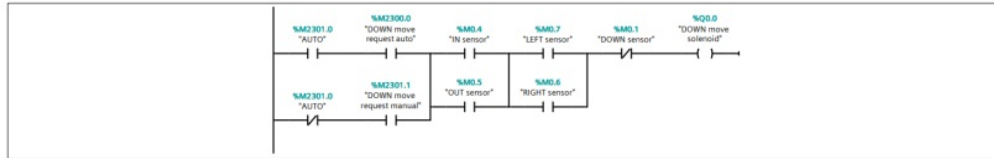
Network 2: Object sensor logic and timer for the start delay

When an object rolls down the chute a memory bit will be set to show that an object has been detected, this can also be done by pressing the start button, as long as the auto sequence has not gone past step 3. Once the object has been detected a 2s ON delay timer will start, this will allow the sequence to start. 1s after the sequence has started the object sensor memory bit will be reset.



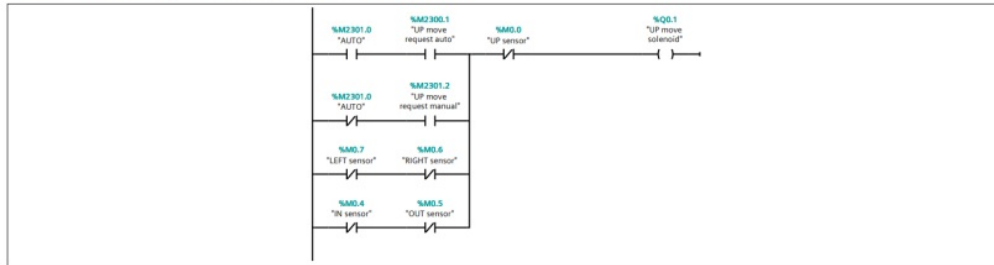
Network 3: Move down controls with interlocks

This network controls the down movement of the robot. First the robot must be in auto or manual, then there must be a move request sent. Next are the safety interlocks that prevent the robot from moving down unless it is all the way OUT / IN or all the way LEFT / RIGHT. The normally closed contact for the down sensor will stop the robot moving down once it has reached the limit.



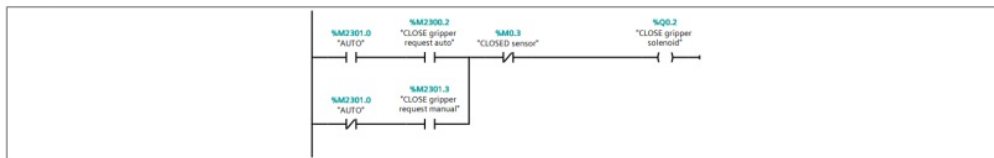
Network 4: Move up controls with interlocks

This network controls the up movement of the robot. The 4 rungs of the program act as OR logic, this means that for the robot to move UP it must reach 1 of 4 criteria. These are, 1) in AUTO and UP move request, 2) in MANUAL and UP move request, 3) The robot is NOT all the way LEFT or RIGHT, 4) The robot is NOT all the way IN or OUT. The UP sensor is used to stop the robot moving UP once it has reached the limit.



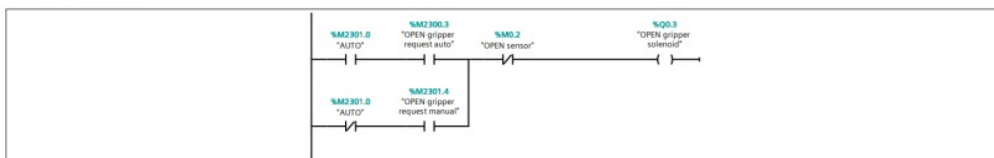
Network 5: Close gripper

This network controls when the grippers CLOSE, for them to close the closed sensor must NOT be made, the program must also be in AUTO or MANUAL with the corresponding request sent.



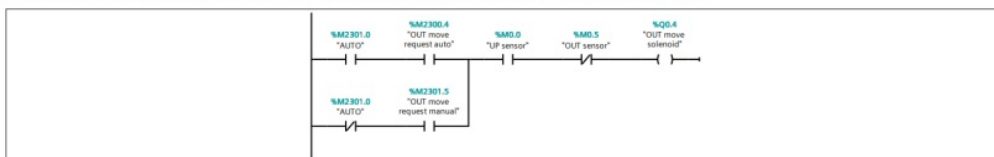
Network 6: Open gripper

This network controls when the grippers OPEN, for them to open the OPEN sensor must NOT be made, the program must also be in AUTO or MANUAL with the corresponding request sent.



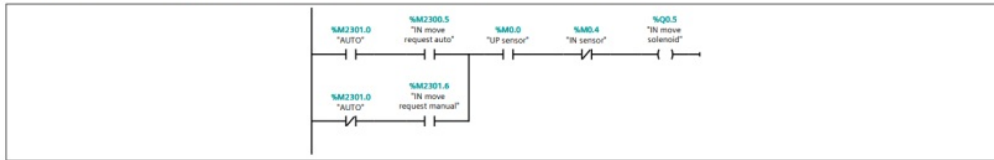
Network 7: Move out

This network controls the OUT movement of the robot. First the robot must be in AUTO or MANUAL and the corresponding request must be sent. Then the robot will move OUT only if the robot is in the UP position, the OUT sensor will stop the robot moving once the limit has been reached.



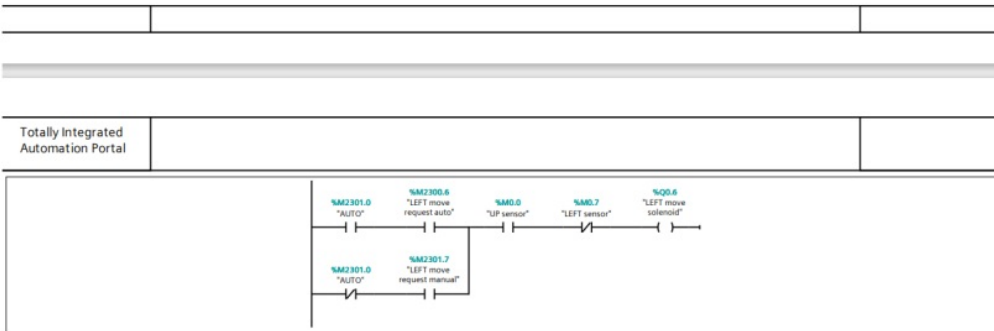
Network 8: Move in

This network controls the IN movement of the robot. First the robot must be in AUTO or MANUAL and the corresponding request must be sent. Then the robot will move IN only if the robot is in the UP position, the IN sensor will stop the robot moving once the limit has been reached.



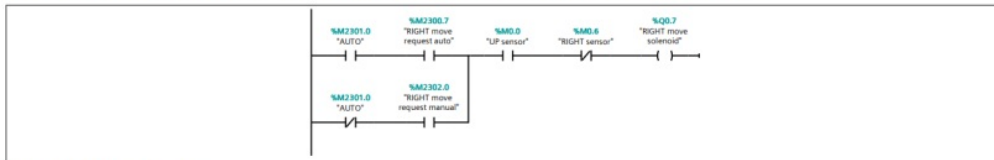
Network 9: Move left

This network controls the LEFT movement of the robot. First the robot must be in AUTO or MANUAL and the corresponding request must be sent. Then the robot will move LEFT only if the robot is in the UP position, the LEFT sensor will stop the robot moving once the limit has been reached.



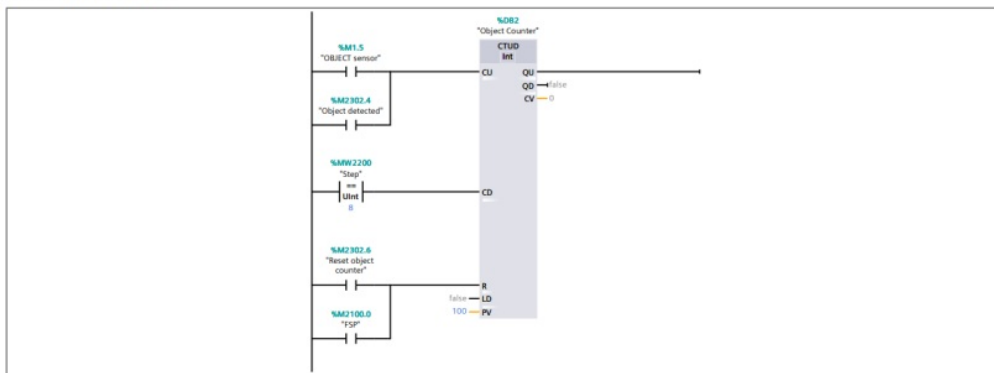
Network 10: Move right

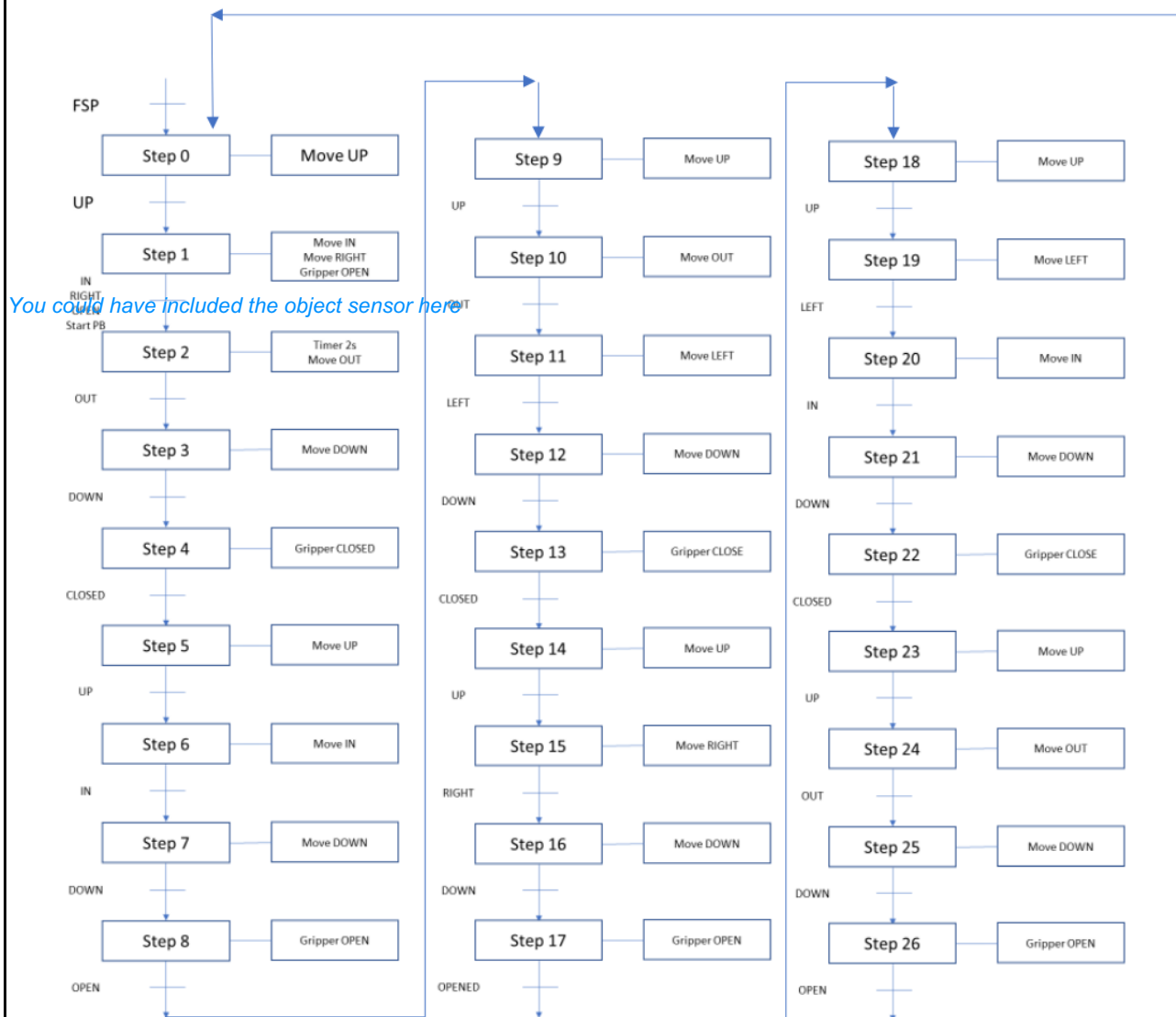
This network controls the RIGHT movement of the robot. First the robot must be in AUTO or MANUAL and the corresponding request must be sent. Then the robot will move RIGHT only if the robot is in the UP position, the RIGHT sensor will stop the robot moving once the limit has been reached.



Network 11: Object Counter

This network is responsible for keeping count of how many objects are within the robot system. A count up/down counter is used for this task. The counter will count up either when the object detects an object rolling past or whenever the M bit for 'object detected' is set. The counter will count down whenever the step number equals 8, this is the step that the object is removed from the system. The counter will be reset whenever the 'object counter reset' button is pressed or whenever the program carries out a first scan pulse.





Good comments

```
1 IF ("FSP" = 1) OR ("AUTO" = 0) OR ("Reset PB" = 1) THEN //these are the initial conditions for
2   "DOWN move request auto" := 0; //the auto sequence, it will reset all
3   "UP move request auto" := 0; //the move requests and the step whenever
4   "CLOSE gripper request auto" := 0; //the FSP occurs, the auto sequence isn't selected or when
5   "OPEN gripper request auto" := 0; //the reset button is pressed
6   "OUT move request auto" := 0;
7   "IN move request auto" := 0;
8   "LEFT move request auto" := 0;
9   "RIGHT move request auto" := 0;
10  "Step" := 0;
11  "Next Step" := 0;
12 END_IF;

15 IF "AUTO" = 1 THEN
16   CASE "Step" OF
17     0://the first step ensures the robot is UP before any other movements occur.
18     "UP move request auto" := 1;
19     IF "UP sensor" = 1 THEN
20       "UP move request auto" := 0;
21       "Next Step" := 1;
22     END_IF;
23     1://move home
24     "IN move request auto" := 1; //this step moves the robot to its home position
25     "RIGHT move request auto" := 1;
26     "OPEN gripper request auto" := 1;
27     IF ("IN sensor" AND "RIGHT sensor" AND "OPEN sensor" AND "Start sequence") = 1 THEN
28       "IN move request auto" := 0; //Once the limits are made for each direction, the robot will stop moving
29       "RIGHT move request auto" := 0;
30       "OPEN gripper request auto" := 0;
31       IF ("Object Counter".CV = 1) THEN //these 3 IF statements decide what station the robot will move to
32         "Next Step" := 18; //initially, the station depends how many objects are in the system.
33       END_IF;
34       IF ("Object Counter".CV = 2) THEN
35         "Next Step" := 9;
36       END_IF;
37       IF ("Object Counter".CV = 3) THEN
38         "Next Step" := 2;
39       END_IF;
40     END_IF;

41     2://move object from station 3 to station 4
42     //move out
43     "OUT move request auto" := 1; //if there are 3 objects in the system, this will be the first set of movements
44     IF "OUT sensor" = 1 THEN //to move the object from station 3 to 4
45       "OUT move request auto" := 0;
46       "Next Step" := 3;
47     END_IF;
48     3://move down
49     "DOWN move request auto" := 1;
50     IF "DOWN sensor" = 1 THEN
51       "DOWN move request auto" := 0;
52       "Next Step" := 4;
53     END_IF;
54     4://close gripper
55     "CLOSE gripper request auto" := 1;
56     IF "CLOSED sensor" = 1 THEN
57       "CLOSE gripper request auto" := 0;
58       "Next Step" := 5;
59     END_IF;
60     5://move up
61     "UP move request auto" := 1;
62     IF "UP sensor" = 1 THEN
63       "UP move request auto" := 0;
64       "Next Step" := 6;
65     END_IF;
66     6://move in
67     "IN move request auto" := 1;
68     IF "IN sensor" = 1 THEN
69       "IN move request auto" := 0;
70       "Next Step" := 7;
71     END_IF;
```

```

72 7://move down
73  "DOWN move request auto" := 1;
74  IF "DOWN sensor" = 1 THEN
75      "DOWN move request auto" := 0;
76      "Next Step" := 8;
77  END_IF;
78 8://open gripper
79  "OPEN gripper request auto" := 1;  //at this point an object has been removed from the system.
80  IF "OPEN sensor" = 1 THEN
81      "OPEN gripper request auto" := 0;
82      "Next Step" := 9;
83  END_IF;
84 9://move object from station 2 to station 3
85  //move up
86  "UP move request auto" := 1;  //if there are 2 objects in the system, this will be the first set of movements
87  IF "UP sensor" = 1 THEN  //to move the object from station 2 to 3
88      "UP move request auto" := 0;
89      "Next Step" := 10;
90  END_IF;
91 10://move out
92  "OUT move request auto" := 1;
93  IF "OUT sensor" = 1 THEN
94      "OUT move request auto" := 0;
95      "Next Step" := 11;
96  END_IF;
97 11://move left
98  "LEFT move request auto" := 1;
99  IF "LEFT sensor" = 1 THEN
100      "LEFT move request auto" := 0;
101      "Next Step" := 12;
102  END_IF;
103 12://move down
104  "DOWN move request auto" := 1;
105  IF "DOWN sensor" = 1 THEN
106      "DOWN move request auto" := 0;
107      "Next Step" := 13;
108  END_IF;
109 13://close gripper
110  "CLOSE gripper request auto" := 1;
111  IF "CLOSED sensor" = 1 THEN
112      "CLOSE gripper request auto" := 0;
113      "Next Step" := 14;
114  END_IF;
115 14://move up
116  "UP move request auto" := 1;
117  IF "UP sensor" = 1 THEN
118      "UP move request auto" := 0;
119      "Next Step" := 15;
120  END_IF;
121 15://move right
122  "RIGHT move request auto" := 1;
123  IF "RIGHT sensor" = 1 THEN
124      "RIGHT move request auto" := 0;
125      "Next Step" := 16;
126  END_IF;
127 16://move down
128  "DOWN move request auto" := 1;
129  IF "DOWN sensor" = 1 THEN
130      "DOWN move request auto" := 0;
131      "Next Step" := 17;
132  END_IF;

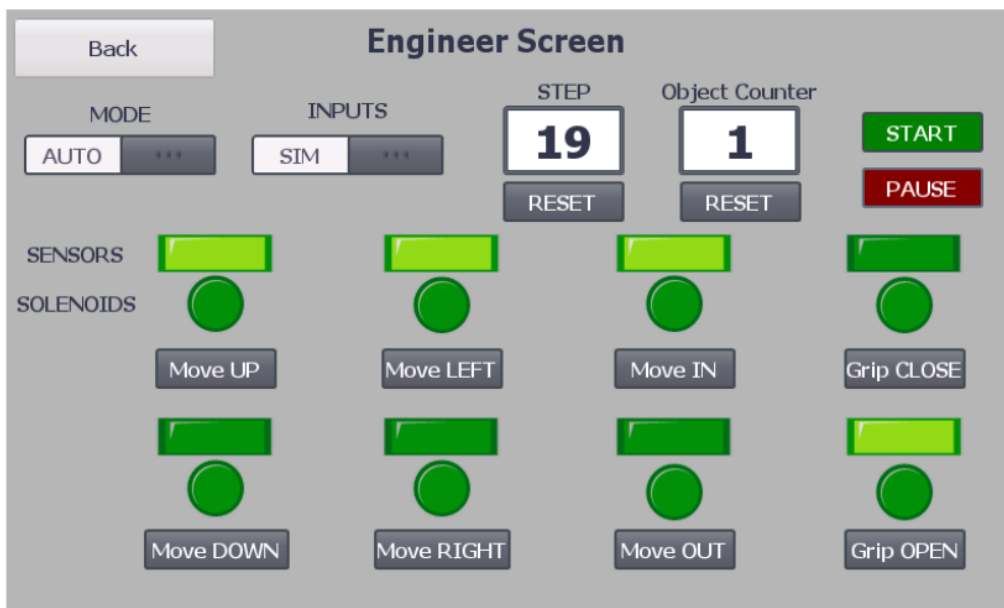
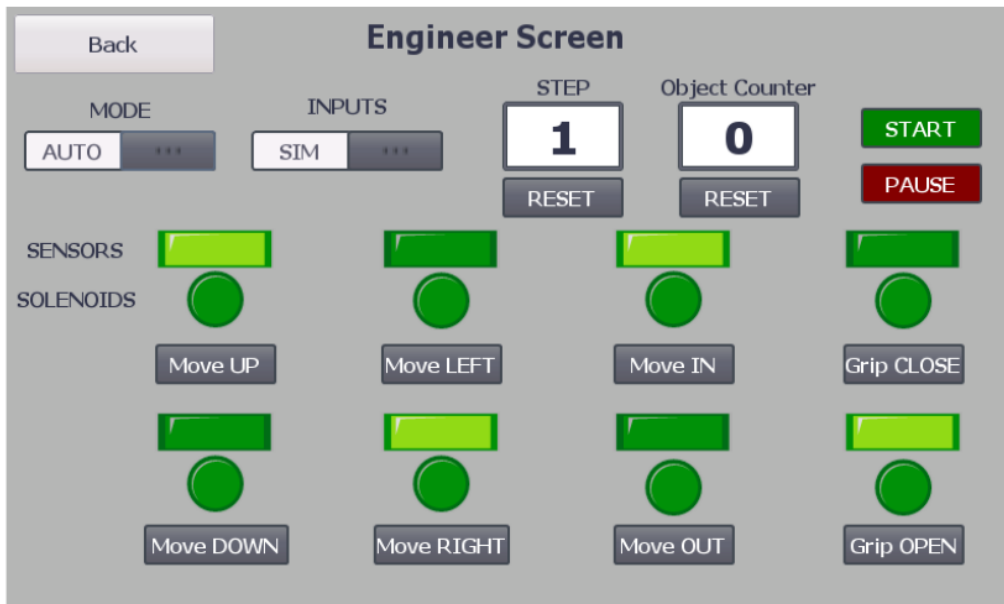
```

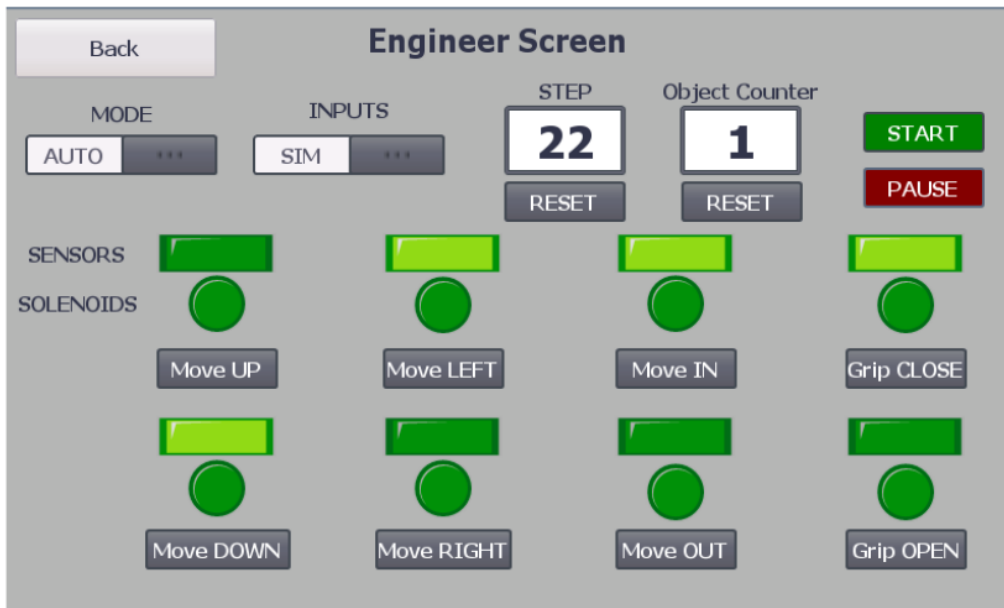
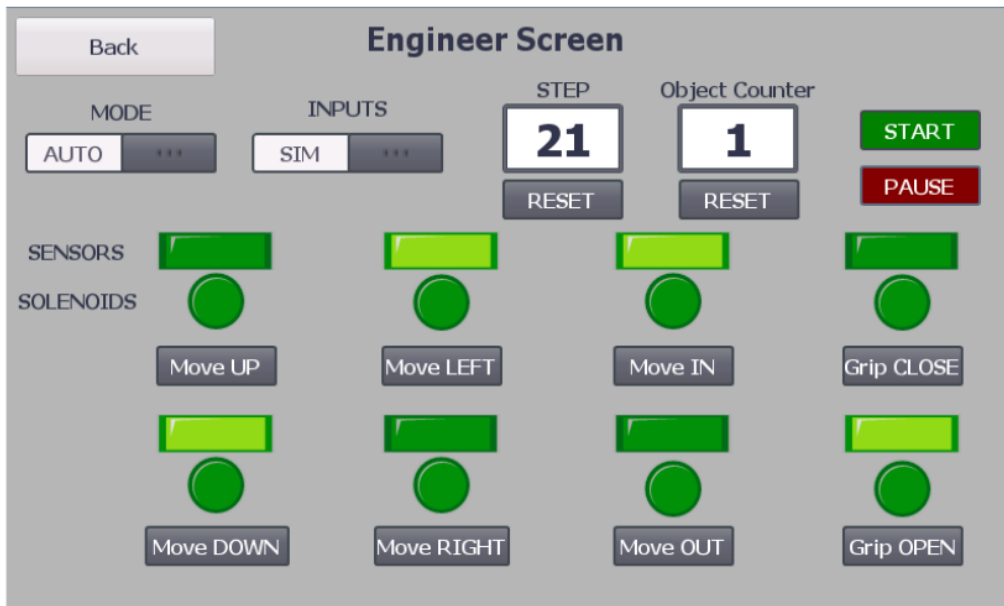
```

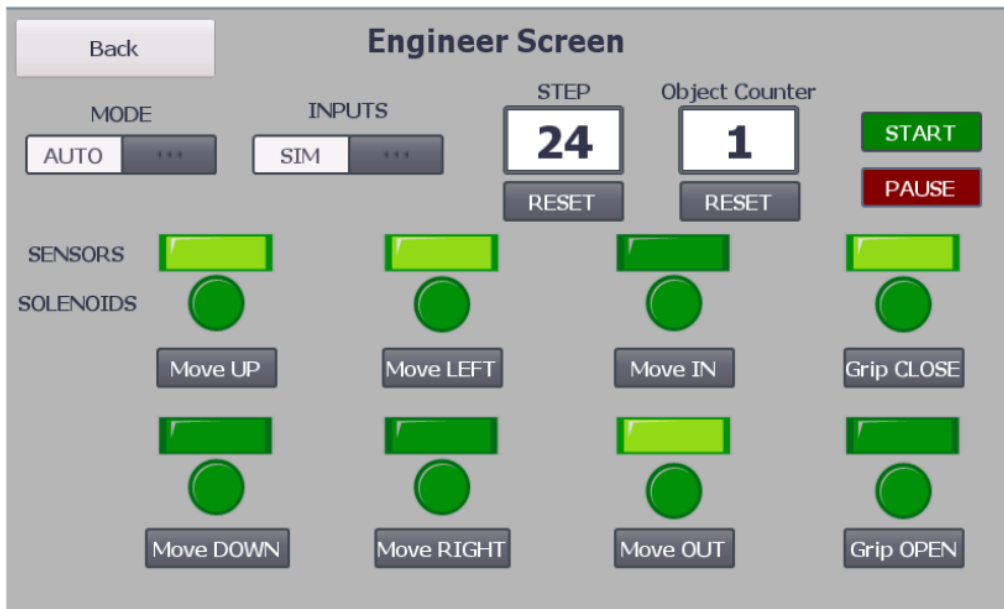
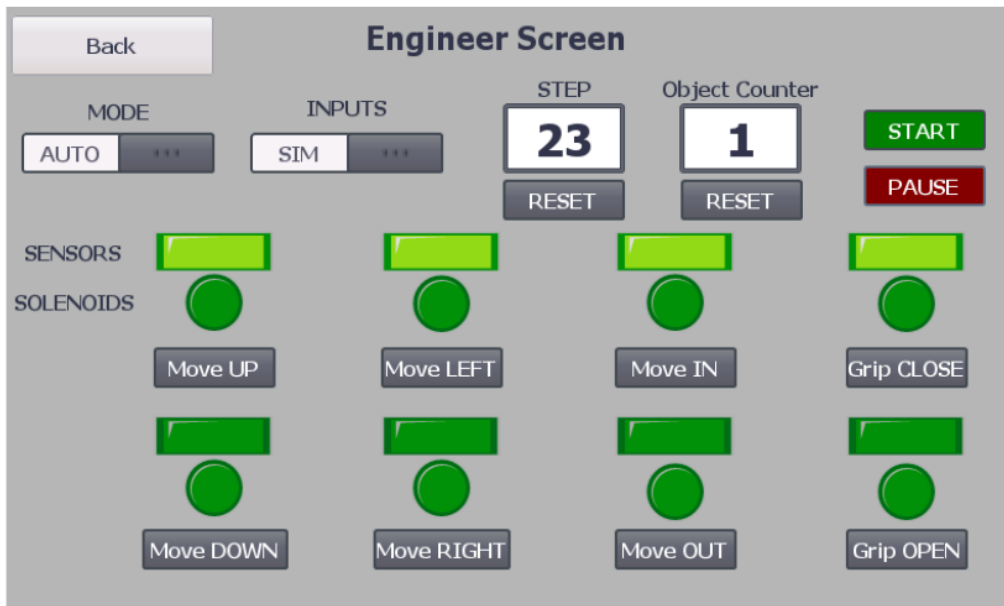
133 | 17://open gripper
134 | "OPEN gripper request auto" := 1; //the object is now at station 3
135 | IF "OPEN sensor" = 1 THEN
136 |     "OPEN gripper request auto" := 0;
137 |     "Next Step" := 18;
138 | END_IF;
139 | 18://move object from station 1 to station 2
140 | //move up
141 | "UP move request auto" := 1; //if there is 1 object in the system, this will be the first set of movements
142 | IF "UP sensor" = 1 THEN //to move the object from station 1 to 2
143 |     "UP move request auto" := 0;
144 |     "Next Step" := 19;
145 | END_IF;
146 | 19://move left
147 | "LEFT move request auto" := 1;
148 | IF "LEFT sensor" = 1 THEN
149 |     "LEFT move request auto" := 0;
150 |     "Next Step" := 20;
151 | END_IF;
152 | 20://move in
153 | "IN move request auto" := 1;
154 | IF "IN sensor" = 1 THEN
155 |     "IN move request auto" := 0;
156 |     "Next Step" := 21;
157 | END_IF;
158 | 21://move down
159 | "DOWN move request auto" := 1;
160 | IF "DOWN sensor" = 1 THEN
161 |     "DOWN move request auto" := 0;
162 |     "Next Step" := 22;
163 | END_IF;
164 | 22://close gripper
165 | "CLOSE gripper request auto" := 1;
166 | IF "CLOSED sensor" = 1 THEN
167 |     "CLOSE gripper request auto" := 0;
168 |     "Next Step" := 23;
169 | END_IF;
170 | 23://move up
171 | "UP move request auto" := 1;
172 | IF "UP sensor" = 1 THEN
173 |     "UP move request auto" := 0;
174 |     "Next Step" := 24;
175 | END_IF;
176 | 24://move out
177 | "OUT move request auto" := 1;
178 | IF "OUT sensor" = 1 THEN
179 |     "OUT move request auto" := 0;
180 |     "Next Step" := 25;
181 | END_IF;
182 | 25://move down
183 | "DOWN move request auto" := 1;
184 | IF "DOWN sensor" = 1 THEN
185 |     "DOWN move request auto" := 0;
186 |     "Next Step" := 26;
187 | END_IF;
188 | 26://open gripper
189 | "OPEN gripper request auto" := 1; //at this point the object is at station 2
190 | IF "OPEN sensor" = 1 THEN
191 |     "OPEN gripper request auto" := 0;
192 |     "Next Step" := 0;
193 | END_IF;
194 |
195 | END_CASE;
196 | "Step" := "Next Step";
197 | END_IF;
198 |

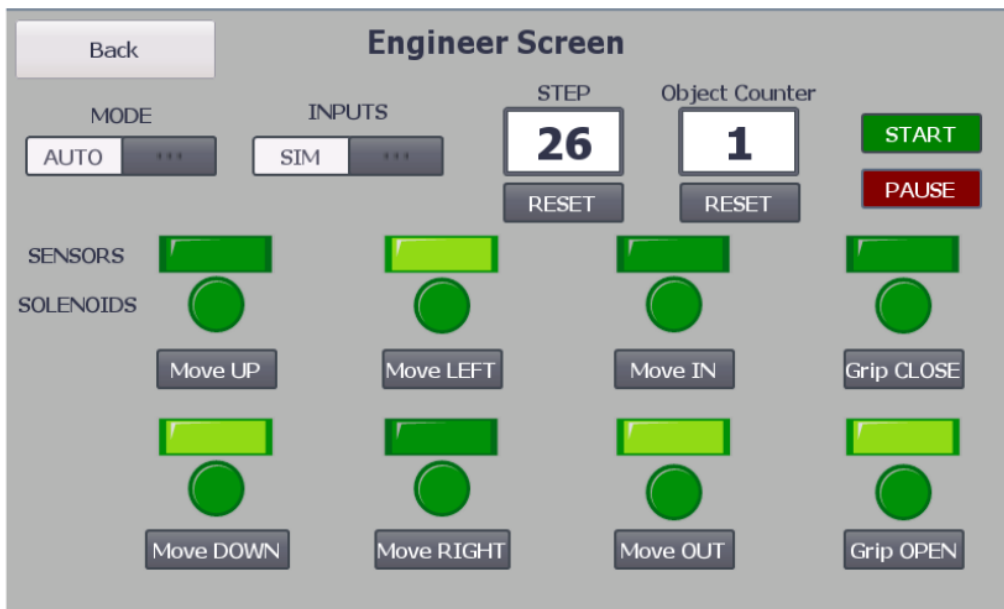
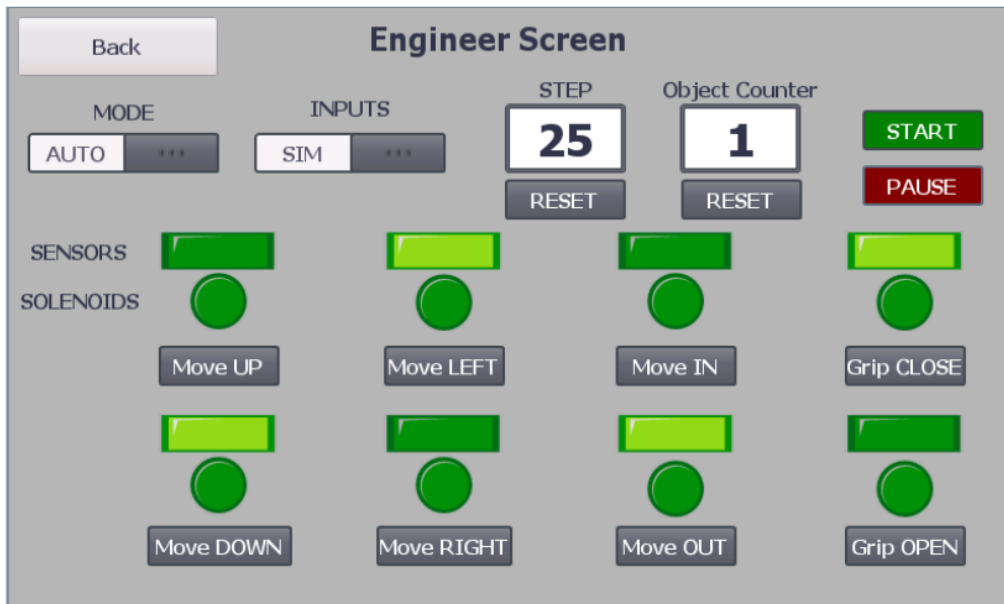
```

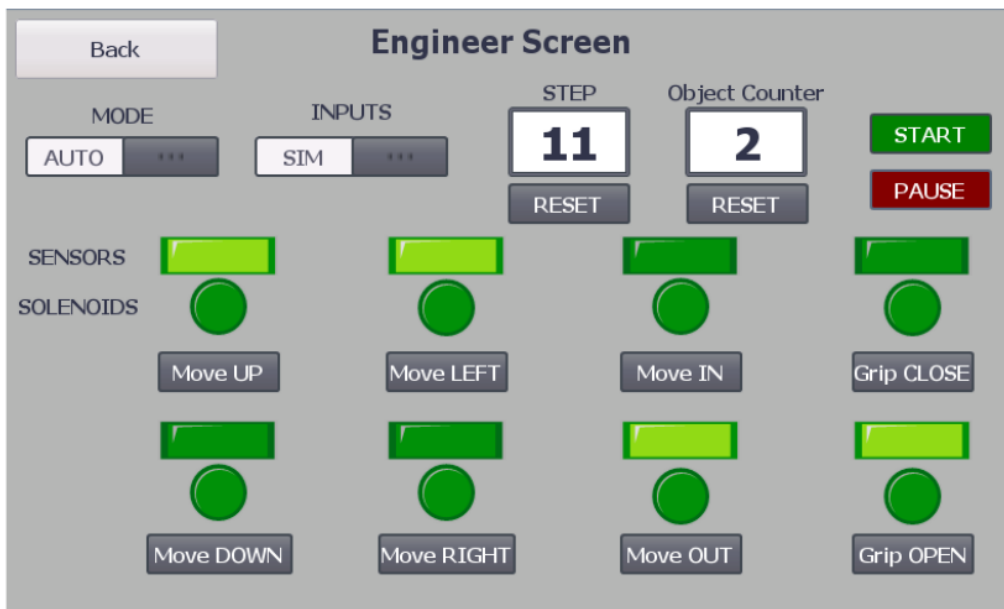
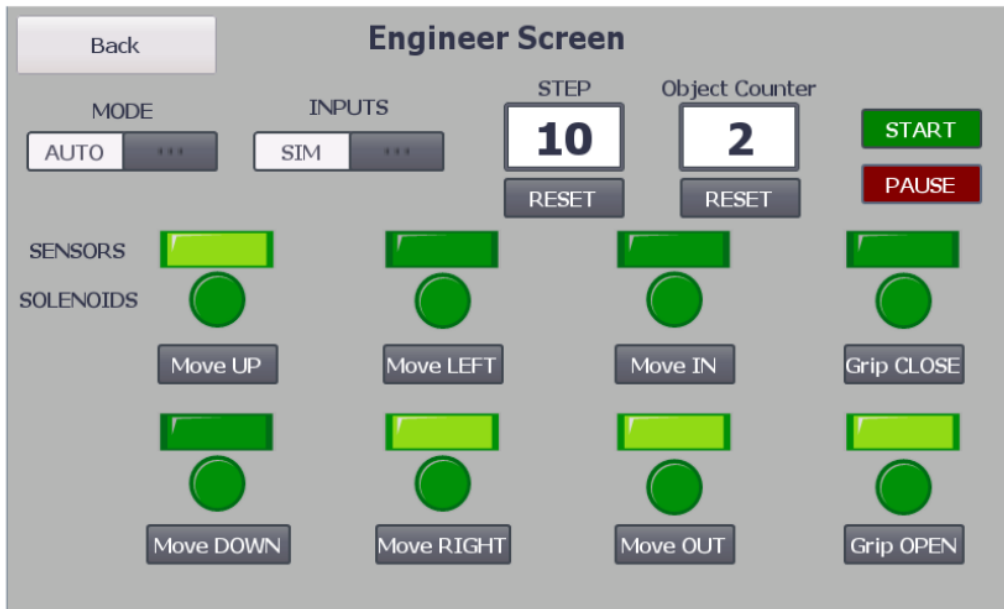
*Each image should be numbered so they
could be linked to the testing tables*

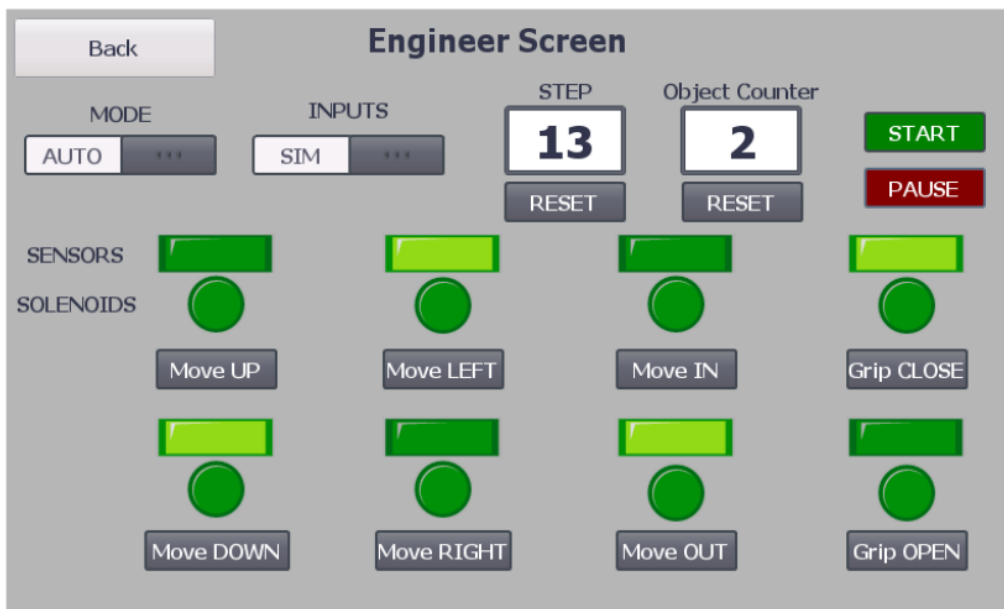
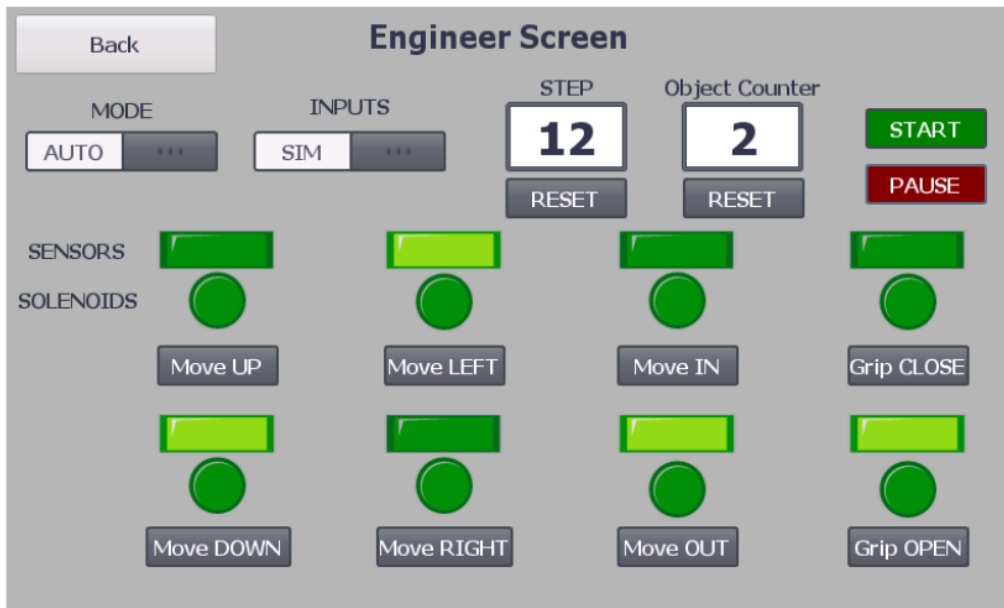


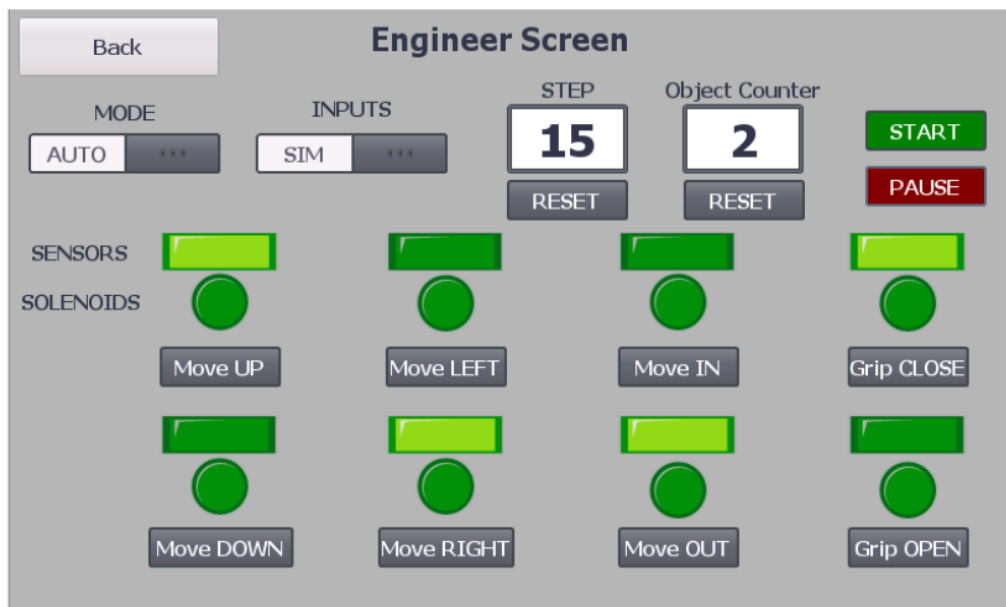
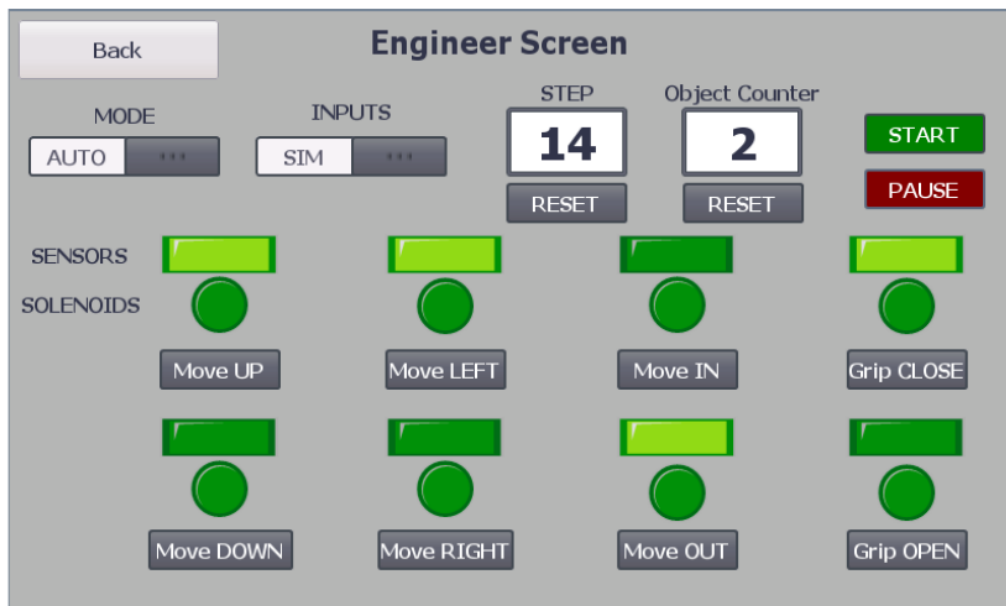


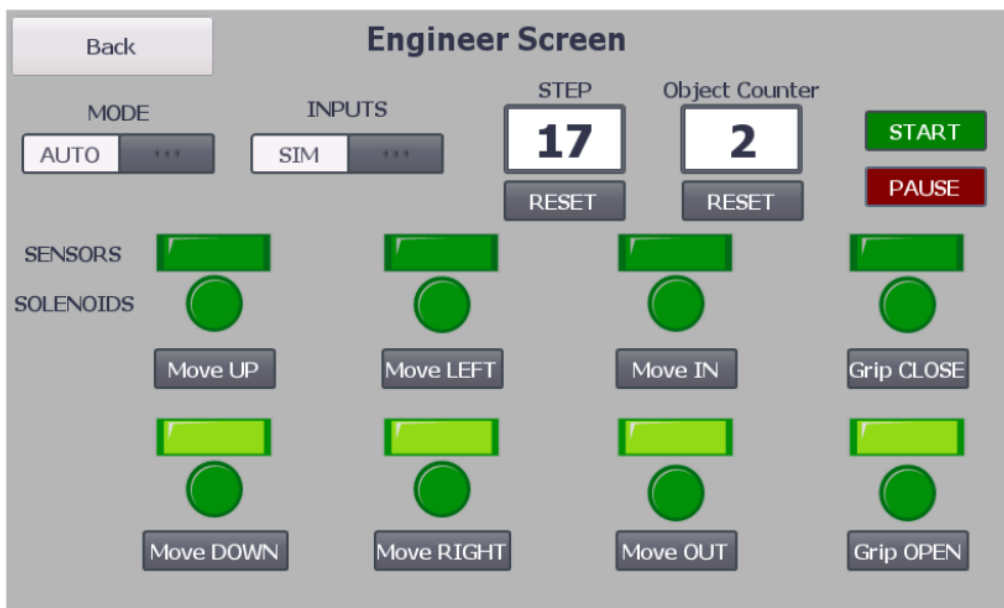
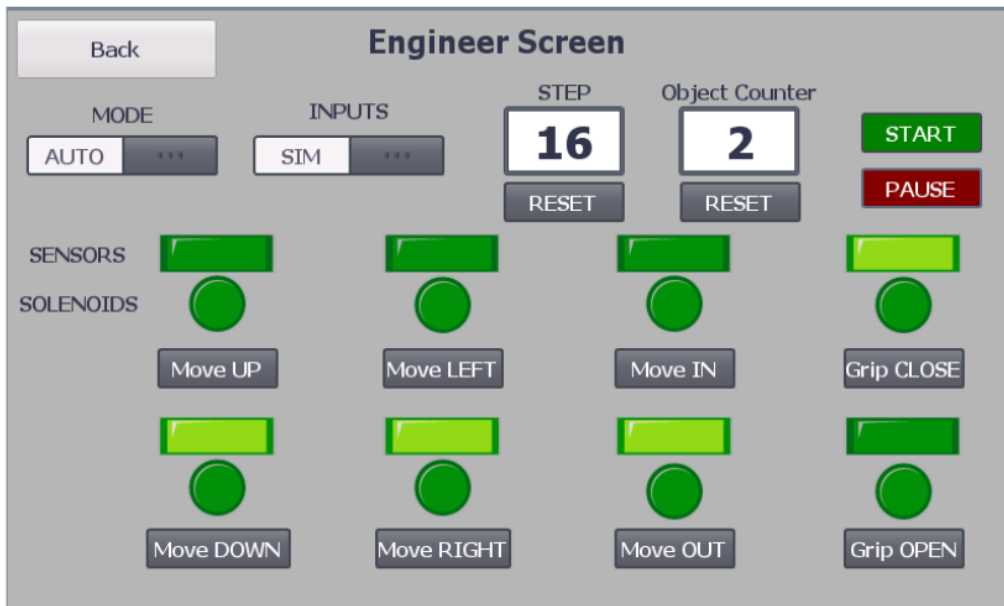


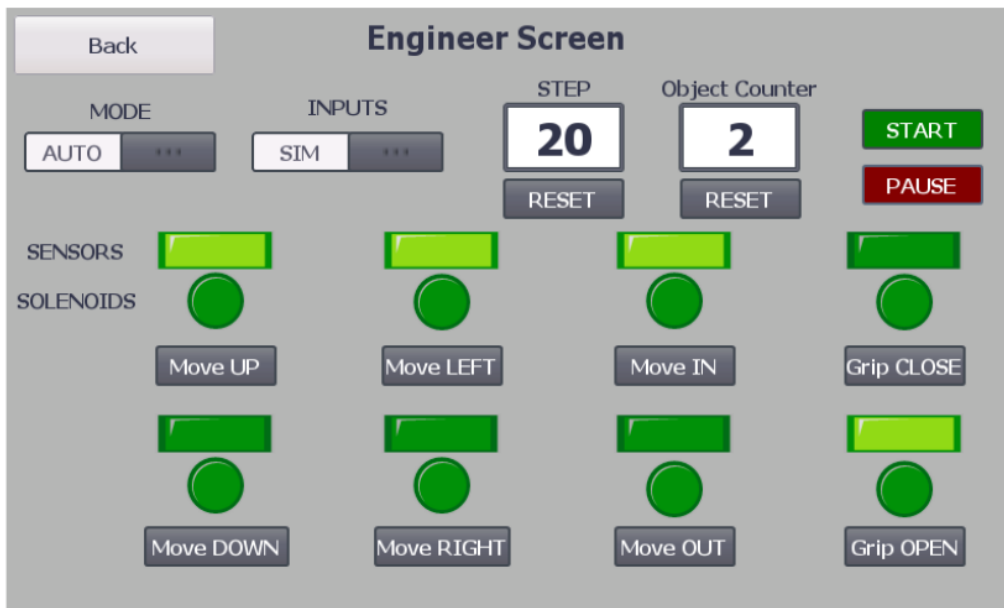
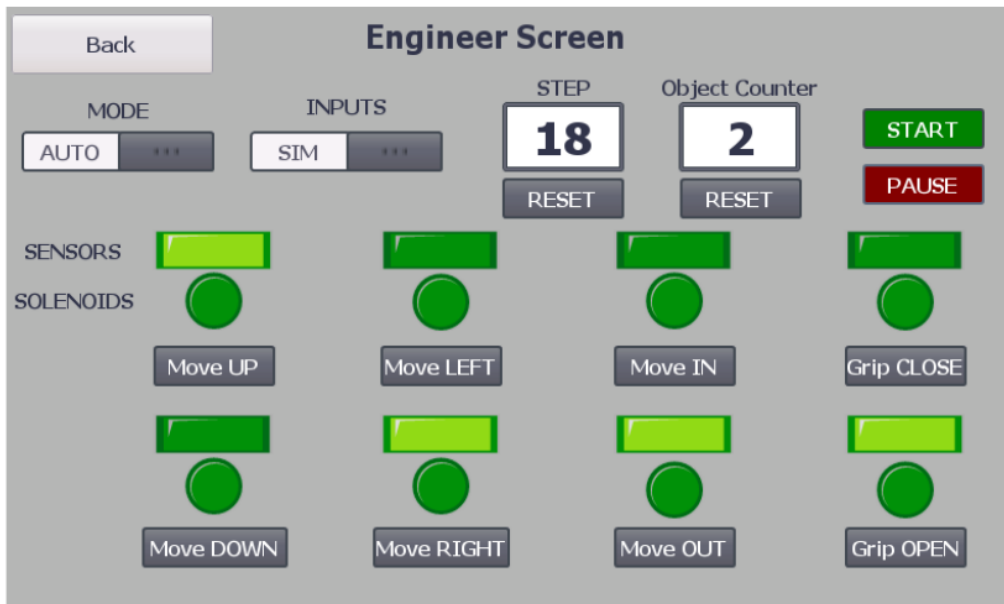


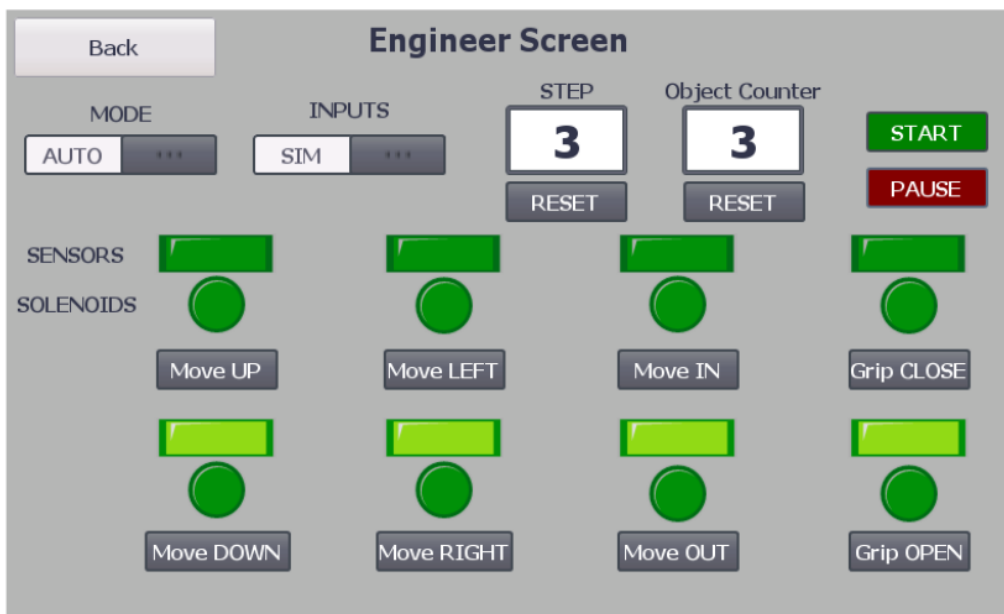
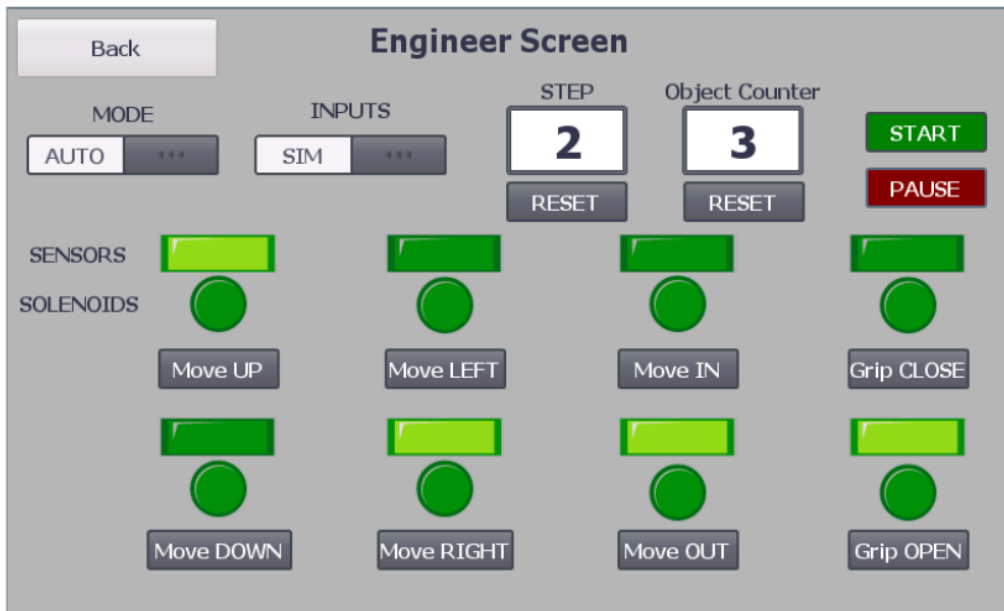


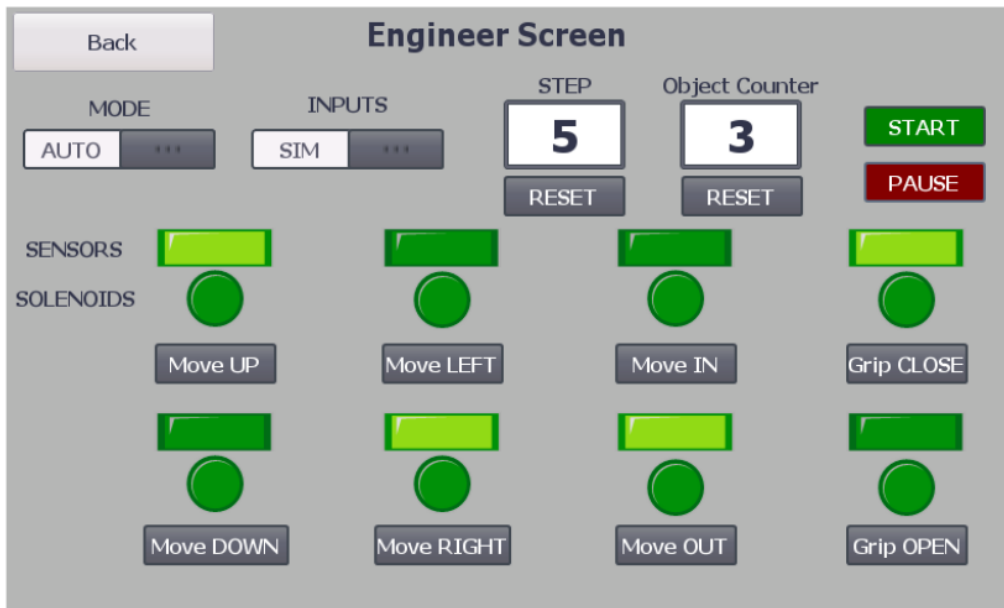
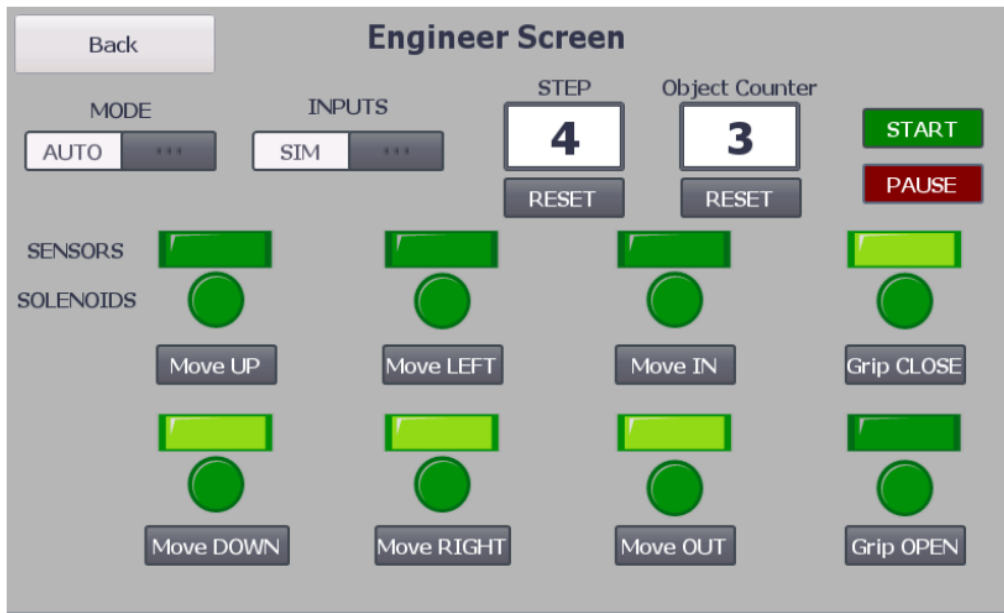


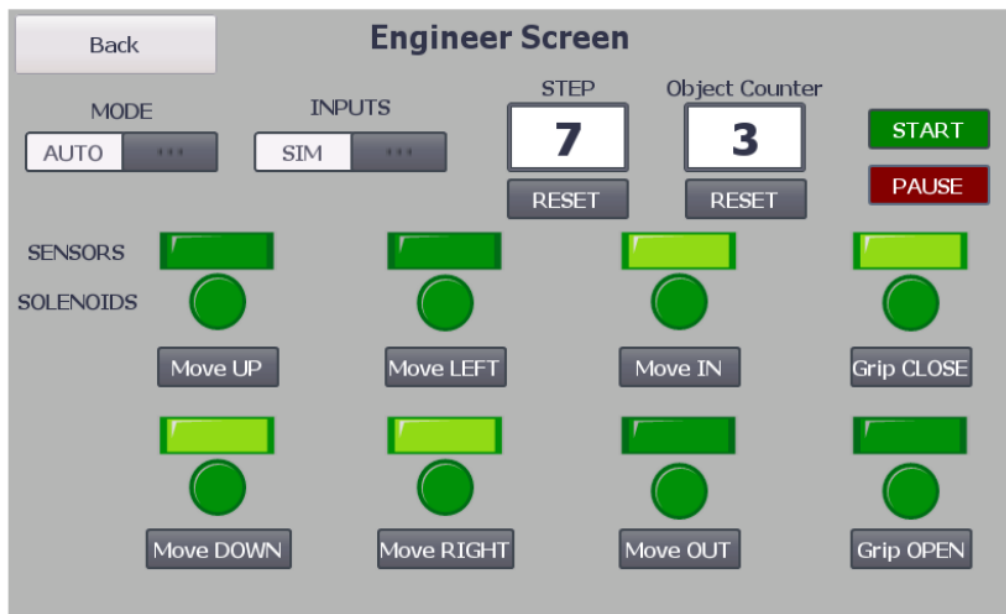
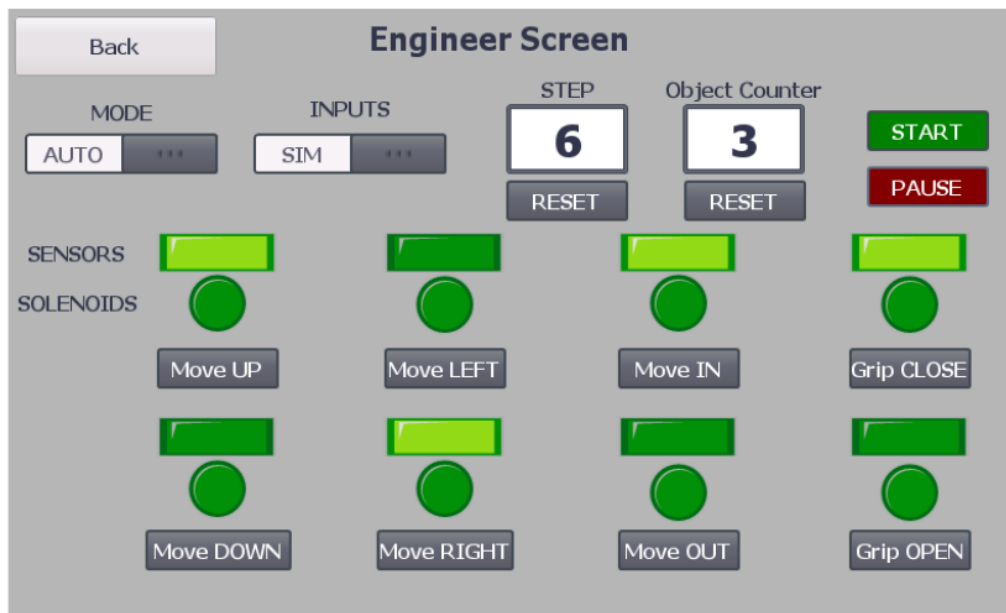


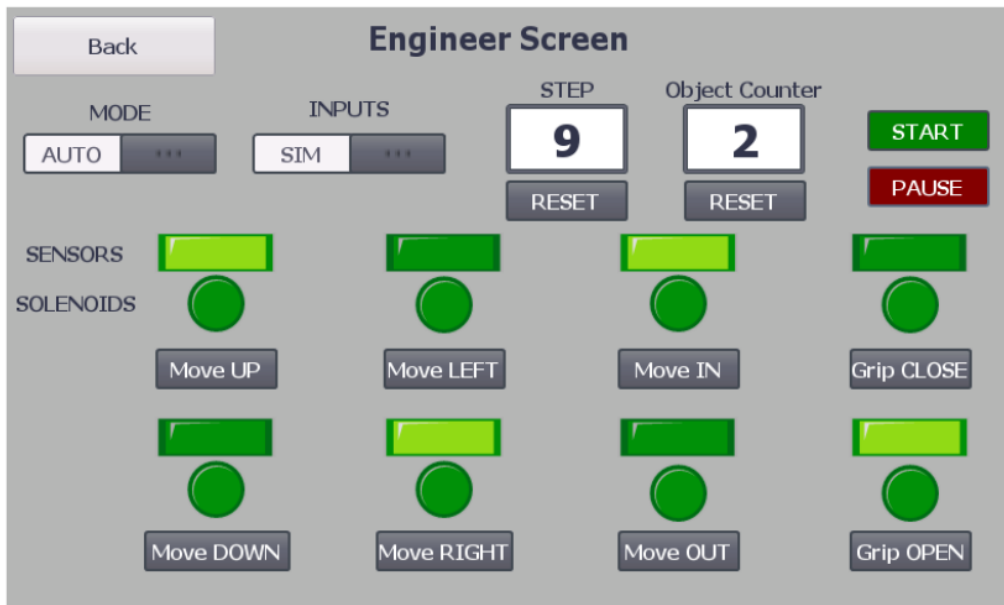
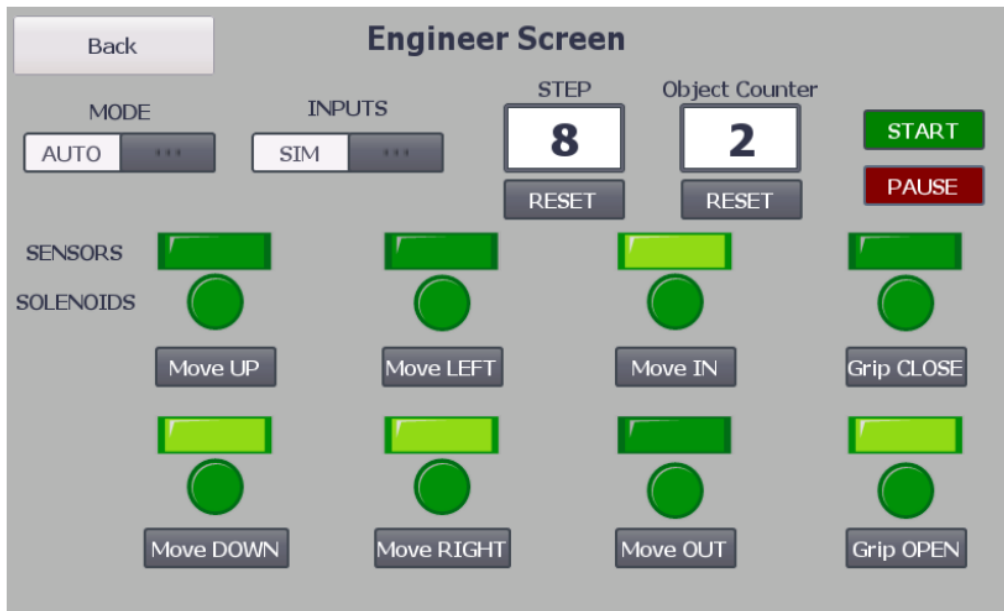






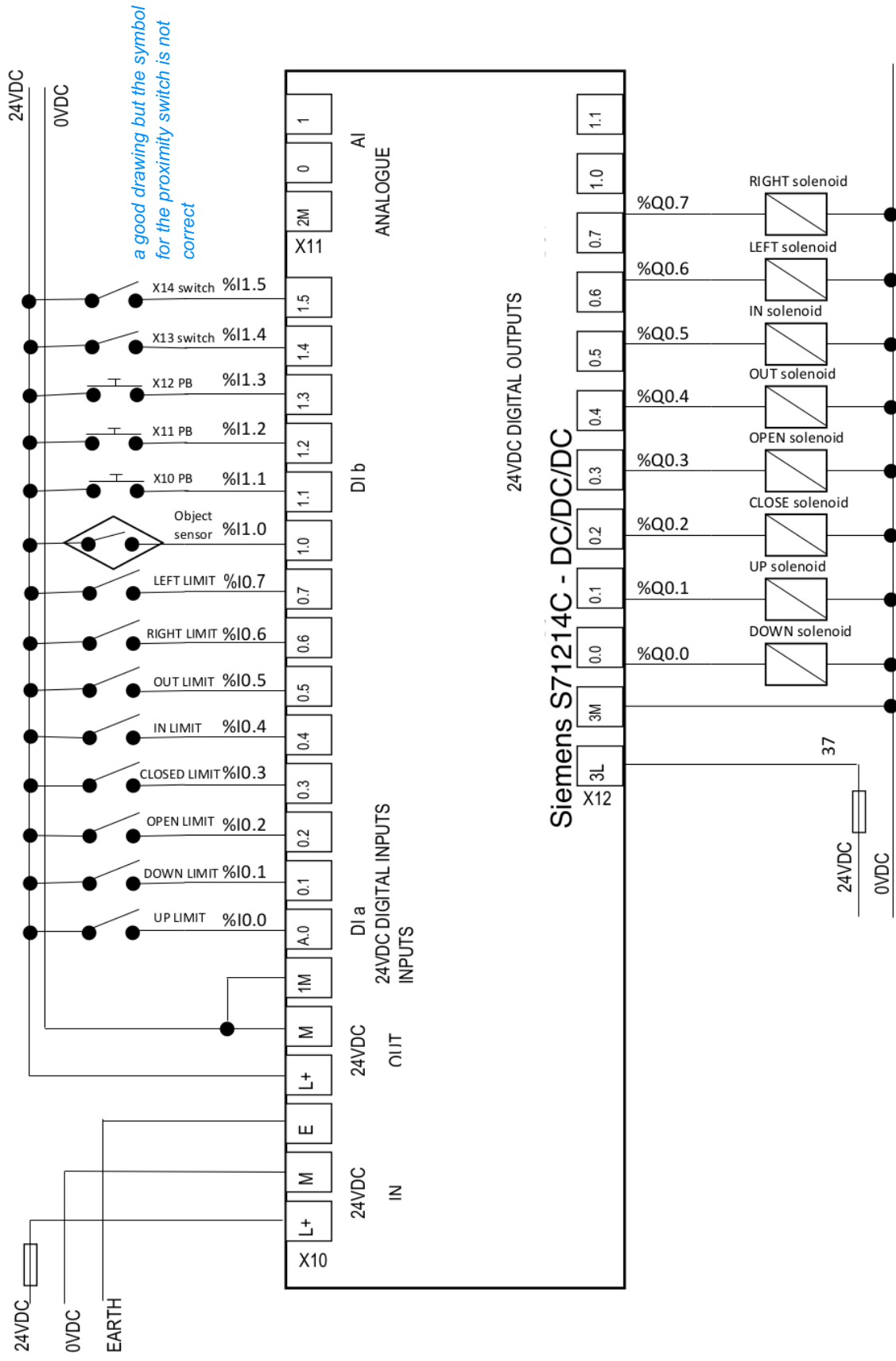






PLC tags								
	Name	Tag table	Data type	Address	Retain	Acces...	Writa...	Visibl...
1	UP sensor	Simulation Tags	Bool	%M0.0	☐	☒	☒	☒
2	DOWN sensor	Simulation Tags	Bool	%M0.1	☐	☒	☒	☒
3	OPEN sensor	Simulation Tags	Bool	%M0.2	☐	☒	☒	☒
4	CLOSED sensor	Simulation Tags	Bool	%M0.3	☐	☒	☒	☒
5	IN sensor	Simulation 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	%IB0	☐	☒	☒	☒
18	PLC Base Unit input byte 1	Assignment tag table	Byte	%IB1	☐	☒	☒	☒
19	PLC Base Unit mapped input by...	Assignment tag table	Byte	%IB1	☐	☒	☒	☒
20	System_Byte	System Tags	Byte	%MB2100	☐	☒	☒	☒
21	FSP	System Tags	Bool	%M2100.0	☐	☒	☒	☒
22	DiagStatusUpdate	System Tags	Bool	%M2100.1	☐	☒	☒	☒
23	AlwaysTRUE	System Tags	Bool	%M2100.2	☐	☒	☒	☒
24	AlwaysFALSE	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	☐	☒	☒	☒

PLC tags								
	Name	Tag table	Data type	Address	Retain	Acces...	Writa...	Visibl...
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	UP move solenoid	Assignment tag table	Bool	%Q0.1	☐	☒	☒	☒
35	DOWN move solenoid	Assignment tag table	Bool	%Q0.0	☐	☒	☒	☒
36	LEFT move solenoid	Assignment tag table	Bool	%Q0.6	☐	☒	☒	☒
37	RIGHT move solenoid	Assignment tag table	Bool	%Q0.7	☐	☒	☒	☒
38	IN move solenoid	Assignment tag table	Bool	%Q0.5	☐	☒	☒	☒
39	OUT move solenoid	Assignment tag table	Bool	%Q0.4	☐	☒	☒	☒
40	OPEN gripper solenoid	Assignment tag table	Bool	%Q0.3	☐	☒	☒	☒
41	CLOSE gripper solenoid	Assignment tag table	Bool	%Q0.2	☐	☒	☒	☒
42	Step	Assignment tag table	UInt	%MW2200	☐	☒	☒	☒
43	Next Step	Assignment tag table	UInt	%MW2202	☐	☒	☒	☒
44	DOWN move request auto	Assignment tag table	Bool	%M2300.0	☐	☒	☒	☒
45	UP move request auto	Assignment tag table	Bool	%M2300.1	☐	☒	☒	☒
46	CLOSE gripper request auto	Assignment tag table	Bool	%M2300.2	☐	☒	☒	☒
47	OPEN gripper request auto	Assignment tag table	Bool	%M2300.3	☐	☒	☒	☒
48	OUT move request auto	Assignment tag table	Bool	%M2300.4	☐	☒	☒	☒
49	IN move request auto	Assignment tag table	Bool	%M2300.5	☐	☒	☒	☒
50	LEFT move request auto	Assignment tag table	Bool	%M2300.6	☐	☒	☒	☒
51	RIGHT move request auto	Assignment tag table	Bool	%M2300.7	☐	☒	☒	☒
52	AUTO	Assignment tag table	Bool	%M2301.0	☐	☒	☒	☒
53	DOWN move request manual	Assignment tag table	Bool	%M2301.1	☐	☒	☒	☒
54	UP move request manual	Assignment tag table	Bool	%M2301.2	☐	☒	☒	☒
55	CLOSE gripper request manual	Assignment tag table	Bool	%M2301.3	☐	☒	☒	☒
56	OPEN gripper request manual	Assignment tag table	Bool	%M2301.4	☐	☒	☒	☒
57	OUT move request manual	Assignment tag table	Bool	%M2301.5	☐	☒	☒	☒
58	IN move request manual	Assignment tag table	Bool	%M2301.6	☐	☒	☒	☒
59	LEFT move request manual	Assignment tag table	Bool	%M2301.7	☐	☒	☒	☒
60	RIGHT move request manual	Assignment tag table	Bool	%M2302.0	☐	☒	☒	☒
61	Start PB	Assignment tag table	Bool	%M2302.1	☐	☒	☒	☒
62	Pause PB	Assignment tag table	Bool	%M2302.2	☐	☒	☒	☒
63	Reset PB	Assignment tag table	Bool	%M2302.3	☐	☒	☒	☒
64	Object detected	Assignment tag table	Bool	%M2302.4	☐	☒	☒	☒
65	Start sequence	Assignment tag table	Bool	%M2302.5	☐	☒	☒	☒
66	Reset object counter	Assignment tag table	Bool	%M2302.6	☐	☒	☒	☒



FINAL GRADE

GENERAL COMMENTS

Instructor**72**/100

PAGE 1

PAGE 2

QM

Good Structure

Report is structured exactly as stated in the assignment brief.

PAGE 3

Text Comment. For a direct quotation, you must say - Siemens states that " " (2020, page ?). This is done by suppressing the author and including the page number.

QM

Incorrectly cited, this is not the Harvard Method

QM

Grocer's apostrophe

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

QM

Grocer's apostrophe

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

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

PAGE 4

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

PAGE 5

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

Text Comment. You could have put the costs in the table and not wasted precious words here

Text Comment. You have not considered the 400 series rack mounted PLCs

Text Comment. Correct conclusion

PAGE 6

PAGE 7

Text Comment. Address of last 6 inputs are incorrect

PAGE 8

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.

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.

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.

PAGE 9

PAGE 10

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. A good testing table but you could have linked to the image evidence of the results

PAGE 11

PAGE 12

PAGE 13

PAGE 14

QM

Correct method of listing.

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

QM

Good range of sources

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

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

Text Comment. You could have included the object sensor here

PAGE 20

Text Comment. Good comments

PAGE 21

PAGE 22

PAGE 23

Text Comment. Each image should be numbered so they could be linked to the testing tables

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

Text Comment. a good drawing but the symbol for the proximity switch is not correct

GRADING FORM: NEW CSG-RUBRIC

BENJAMIN MORGAN

72

1. ENGAGEMENT WITH LITERATURE



Good evidence of research from a wide range of sources with both correct method of citation and listing applied. Correct conclusion to research but you did not consider the 400 series rack mounted PLC. (14/20)

14

2. KNOWLEDGE AND UNDERSTANDING



Clear evidence provided for knowledge and understanding, program is well explained. (15/20)

15

3. COGNITIVE AND INTELLECTUAL



Very good evidence of analysis through testing with good discussion, conclusion and recommendations. (15/20)

15

4. PRACTICAL APPLICATION



A well designed and documented system. However, test results could have been linked with table. (13/20)

13

5. TRANSFERABLE SKILLS



A well structured and presented report. Generally well written with good images. (15/20)

15