

2017197

by Aeron Smitham

FILE	16679_AERON_SMITHAM_2017197_178848_1857222489.PDF (2.29M)		
TIME SUBMITTED	08-FEB-2021 10:59AM (UTC+0000)	WORD COUNT	5027
SUBMISSION ID	144391675	CHARACTER COUNT	32837

Course: HNC Electrical and Electronic Engineering
Module: SGEE4004 Programmable Logic Controllers
Assignment: 1. Pick and Place Robot
Name: Aeron Smitham
University number: 2017197
Date submitted: 08/02/2021
Word count: 2737

Table of Contents

Good Structure

Table of Figures	3
1. Research	5
2. Hardware Investigation, Circuit Diagram, and I/O List	9
3. Program Design and Explanation	11
PLC tags	11
Main Program	12
HMI Screens	14
SFC	16
SCL Program	17
4. Simulation Testing	18
Manual Testing	18
Auto Results	19
5. Team Working	21
6. Discussion, Conclusions and Recommendations	22
Bibliography	24
Appendix	25

Table of Figures

Figure 1 – PLC tag table.....	11
Figure 2 – Engineer Screen	14
Figure 3 – Password protection	14
Figure 4 – Operators screen.....	15
Figure 5 - SFC.....	16
Figure 6 – Network 1 Mapped or Simulate Inputs.....	25
Figure 7 – Network 2 Auto Sequence	25
Figure 8 – Network 3 Safe to Move Down	25
Figure 9 – Network 4 Move Up	26
Figure 10 – Network 5 Move Down	26
Figure 11 – Network 6 Move Left	27
Figure 12 – Network 7 Move Right	27
Figure 13 – Network 8 Move In	28
Figure 14 – Network 9 Move Out	28
Figure 15 – Network 10 Gripper Open.....	29
Figure 16 – Network 11 Gripper Close	29
Figure 17 – Network 12 Object Detected	30
Figure 18 – Network 13 Start Sequence	30
Figure 19 – Network 14 Reset Object Detected	30
Figure 20 – Network 15 Object Counter	31
Figure 21 – Reset All Bits.....	31
Figure 22 – Step 0 to 1	32
Figure 23 – Step 2 to 6	32
Figure 24 – Step 7 to 9	33
Figure 25 – Step 10 to 14	33
Figure 26 – Step 15 to 18	33
Figure 27 – Step 19 to 23	34
Figure 28 – Step 24 to 26	34
Figure 29 – Set Start PB and Reset Pause PB	34
Figure 30 – Move right when down.....	35
Figure 31 – Move out when down	35
Figure 32 – Move left when down	36
Figure 33 – Move in when down	36
Figure 34 – Move left.....	37
Figure 35 – Move right.....	37
Figure 36 – Move in	38
Figure 37 – Move out.....	38
Figure 38 – Move down when unsafe.....	39
Figure 39 – Gripper close	39
Figure 40 – Gripper open	40
Figure 41 – Step 0	40
Figure 42 – Step 1	41
Figure 43 – Step 2	41
Figure 44 – Step 3	42
Figure 45 – Step 4	42
Figure 46 – Step 5	43

Figure 47 – Step 6	43
Figure 48 – Step 7	44
Figure 49 – Step 8	44
Figure 50 – Step 9	45
Figure 51 – Step 10	45
Figure 52 – Step 11	46
Figure 53 – Step 12	46
Figure 54 – Step 13	47
Figure 55 – Step 14	47
Figure 56 – Step 15	48
Figure 57 – Step 16	48
Figure 58 – Step 17	49
Figure 59 – Step 18	49
Figure 60 – Step 19	50
Figure 61 – Step 20	50
Figure 62 – Step 21	51
Figure 63 – Step 22	51
Figure 64 – Step 23	52
Figure 65 – Step 24	52
Figure 66 – Step 25	53
Figure 67 – Step 26	53
Figure 68 - SFC.....	54
Figure 69 - SFC.....	55
Figure 70 - SFC.....	56

1. Research

Siemens have a variety of PLCs, ranging from small-scale unitary type PLCs to large-scale rack-mounted ones. Unitary PLCs are generally used for basic automation tasks that require limited I/O and no expansion. Modular PLCs have a central unit that can be expanded upon. Rack-mounted PLCs have a back-plane where modules are mounted on.

The smallest PLC is the LOGO! Unit which is generally *Not quite so, there are expansion modules available* not expandable with 8 digital inputs and 4 digital outputs, making it a unitary PLC. But newer releases have allowed expansion and cloud connectivity, making it compatible with all devices with Modbus TCP (Siemens, 2020). Correctly cited

The 1211 is a unitary PLC in this range, the rest are modular The S7-1200 range is considered a basic modular controller. These controllers have a set amount of I/O in the base unit and then a limited amount of expansion modules to increase the amount of I/O. The 1200 range is the best example of a modular PLC as it shows it can be expanded upon, but it has limits on the number of expansion module. Kamel states that, "The compact design, flexible configuration, and powerful instruction set combine to make the S7-1200 a perfect solution for controlling a wide variety of applications" (2014, p. 44). Correctly cited

The S7-1500 range of PLCs are also the modular type but are advanced controllers. The 1500 can do everything the 1200 can, and more. Increased number of I/O on the base unit, faster scan time and better communication abilities. It has ultra-fast response times which benefits the reading and writing of data, integrated system diagnostics to help troubleshoot (an operator interface on the front of the PLC) and has the ability for seamless drive integration through motion control integration. It is also a more expensive option, compared with the 1200 range, therefore many engineers use a 1200 PLC with additional distributed I/O in their processes rather than spending more on a 1500 PLC (Pegaia, 2014). Correctly cited A distributed I/O module such as the ET200SP can be placed in the field and then connected via PROFINET to the master PLC. The Parmley Graham website states that "Their high speed and transmission rate ensure significantly greater performance than conventional systems" (2020). Correctly cited

Incorrect, the 300 is modular mounted on a metal rail, this is a common misconception as racks have electronic connectors

The S7-300 range PLC is a rack-mounted type, all additional modules are mounted on the rack to communicate with the central controller. This is an advanced PLC for low to mid-range applications that require simple automation tasks. The controller can have one central controller, or master controller, and three expansion units, or slaves. 32 modules can be added to the 4 controllers, with each controller having a maximum of 8 additional modules each. The controllers can come with in-built communication capabilities or have modules added to the rack to allow communication through various protocols such as PROFINET and PROFIBUS (RS-Online, 2002). Correctly cited

This is the only truly rack mounted in the range

The most powerful PLC in the Siemens family is the Simatic S7-400 rack-mounted range. The 400 PLC is for mid to high end applications that require high speed signal processing, almost unlimited number of I/O, fast communication and large memory storage. The 400's backplane allows multiple communication interfaces to be connected to the CPU for separate channels to ensure fast communication, rather than a lot of traffic through one channel. Another benefit of the 400 range is the ability to have multiple CPUs in one central unit, integrating various processes. Nanping, et al., state that, "complicated multi-node and multi-site data acquisition and control systems can be closely incorporated into SIEMENS S7-400 and simply and conveniently used to create high-performance process windows" (2009). Correctly cited

Table 1 compares the Siemens S7 range of PLCs.

PLC	Type	Cost	Max I/O Available	Processing Time	Languages
LOGO! (Siemens, 2020)	Unitary	Cheapest option: £102 Most expensive: £118	8 inputs 4 outputs Max I/O = 12	N/A	LAD & FBD
S7-1200 (Siemens, 2015)	Modular	Cheapest option: £182 Most expensive: £816	Base unit has 14 inputs and 10 outputs 2 analogue inputs 8 signal modules can be added with 16 I/O each Max I/O = 154	• Bit - 0.08 • Word - 1.7 • Real - 2.3	LAD, FBD & SCL
S7-1500 (Siemens, 2015)	Modular	Cheapest option: £661 Most expensive: £5,430	Modules allowed is 32 (including CPU) so 31 possible I/O modules with 32 I/O each Max I/O = 992	• Bit - 10ns • Word - 12ns • Integer - 16ns • Real - 64ns	LAD, FBD, SCL, STL & GRAPH
S7-300 (Siemens, 2005)	Rack-mounted	Cheapest option: £497 Most expensive: £5,122	Digital Inputs = 992 Digital Outputs = 992 Analogue I/O = 372 Max I/O = 2356	• Bit - 0.06µs • Word - 0.12µs • Integer - 0.16µs • Real - 0.59µs	LAD, FBD, SCL, STL, GRAPH & HiGRAPH
S7-400 (Siemens, 2017)	Rack-mounted	Cheapest option: £1438 Most expensive: £14,691	Inputs = 4Kbyte or 32,000 bits Outputs = 4Kbyte or 32,000 bits Max I/O = 64,000	• Bit - 31.25ns • Word - 31.25ns • Integer - 31.25ns • Real - 62.5ns	LAD, FBD, SCL, STL, GRAPH & HiGRAPH

Table 1

Citation
Siemens AG

Criteria for the PLC used in the “Pick and Place Robot” assignment:

- I/O = 22 (14 inputs and 8 outputs)
- Expandability = Not required but would be useful for future improvements
- Processing time = does not require ultra-fast response times
- Languages = LAD and SCL (Also FBD for timers and counters)
- Cost = Cheapest option that meets the criteria

Table 2 shows which PLCs meet the criteria. The findings from Table 1 have been used to complete this task.

Incorrect, the Logo is expandable

PLC	I/O	Expandability	Processing Time	Languages	Cost
LOGO!	✗	✗	✗	✗	✗
S7-1200	✓	✓	✓	✓	✓
S7-1500	✓	✓	✓	✓	✗
S7-300	✓	✓	✓	✓	✗
S7-400	✓	✓	✓	✓	✗

Table 2

Four out of the five ranges meet the first four criteria. All four have the capabilities for this assignment but the most cost-effective option is the S7-1200 range.

There are multiple options in the 1200 range such as: 1211C, 1214C and 1217C. The 1211C is a unitary PLC with 6 inputs and 4 outputs – not meeting the criteria. The 1214C and 1217C are modular PLCs with 24 digital I/O, while the 1217C has additional analogue I/O. The 1214C is the cheaper option, with the 1217C being needlessly expensive for the task as its additional functions are not needed. The 1214C is the best option when compared against the criteria, with it also having the option of expansion if needed for future improvements.

*Correct
conclusion*

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

Digital Inputs	Tag Address
Up Sensor	"%I0.0"
Down Sensor	"%I0.1"
Gripper Open	"%I0.2"
Gripper Closed	"%I0.3"
In Sensor	"%I0.4"
Out Sensor	"%I0.5"
Right Sensor	"%I0.6"
Left Sensor	"%I0.7"
X10 PB	"%I1.0"
X11 PB	"%I1.1"
X12 PB	"%I1.2"
X13 Toggle Switch	"%I1.3"
X14 Toggle Switch	"%I1.4"
Object Sensor	"%I1.5"

Table 3

Digital Outputs	Tag Address
Down Solenoid	"%Q0.0"
Up Solenoid	"%Q0.1"
Close Solenoid	"%Q0.2"
Open Solenoid	"%Q0.3"
Out Solenoid	"%Q0.4"
In Solenoid	"%Q0.5"
Left Solenoid	"%Q0.6"
Right Solenoid	"%Q0.7"

Table 4

Inputs %I0.0 - %I0.7 are sensors that are located at positions on the robot. The sensors are reed switches, which close when the piston actuates, and a magnet energises the switch. These sensors tell the PLC program what position the robot is in. Without these sensors the robot would have to be moved manually by an operator.

The object sensor, %I1.5, is a proximity sensor. The other inputs were previously used but are now spares if needed.

All outputs %Q0.0 - %Q0.7 are solenoids that energise two-position valves that determine the robots direction of movement. The valves are memory blocks, so when it is fired in one direction it will stay there until being fired back.

The PLC used is a S7-1214C DC/DC/DC. It has a central processing unit (CPU), a power supply, I/O wiring connectors, I/O LEDs, a memory card slot and a PROFINET port.

The HMI used is a KTP700 7" display with PROFINET and USB connectors. Connection between the PLC and HMI is made through PROFINET.

A circuit diagram of the PLC wiring can be found in the appendix.

3. Program Design and Explanation

PLC tags

PLC tags							
	Name	Data type	Address	Retain	Accessi- ble from HMI/OPC UA	Writable from HMI/OPC UA	Visible in HMI engi- neering
	Simulate	Bool	%M2000.0	False	True	True	True
	PLC Base Unit input byte 0	Byte	%IB0	False	True	True	True
	PLC Base Unit mapped input byte 0	Byte	%MB0	False	True	True	True
	PLC Base Unit input byte 1	Byte	%IB1	False	True	True	True
	PLC Base Unit mapped input byte 1	Byte	%MB1	False	True	True	True
	UP Solenoid	Bool	%Q0.1	False	True	True	True
	DOWN Solenoid	Bool	%Q0.0	False	True	True	True
	LEFT Solenoid	Bool	%Q0.6	False	True	True	True
	RIGHT Solenoid	Bool	%Q0.7	False	True	True	True
	IN Solenoid	Bool	%Q0.5	False	True	True	True
	OUT Solenoid	Bool	%Q0.4	False	True	True	True
	OPEN Solenoid	Bool	%Q0.3	False	True	True	True
	CLOSE Solenoid	Bool	%Q0.2	False	True	True	True
	Step	UInt	%MW2010	False	True	True	True
	Next_Step	UInt	%MW2012	False	True	True	True
	UP auto request	Bool	%M2200.0	False	True	True	True
	DOWN auto request	Bool	%M2200.1	False	True	True	True
	LEFT auto request	Bool	%M2200.2	False	True	True	True
	RIGHT auto request	Bool	%M2200.3	False	True	True	True
	IN auto request	Bool	%M2200.4	False	True	True	True
	OUT auto request	Bool	%M2200.5	False	True	True	True
	OPEN auto request	Bool	%M2200.6	False	True	True	True
	CLOSE auto request	Bool	%M2200.7	False	True	True	True
	Auto	Bool	%M2201.0	False	True	True	True
	Start PB	Bool	%M2201.1	False	True	True	True
	UP HMI request	Bool	%M2202.0	False	True	True	True
	DOWN HMI request	Bool	%M2202.1	False	True	True	True
	LEFT HMI request	Bool	%M2202.2	False	True	True	True
	RIGHT HMI request	Bool	%M2202.3	False	True	True	True
	IN HMI request	Bool	%M2202.4	False	True	True	True
	OUT HMI request	Bool	%M2202.5	False	True	True	True
	OPEN HMI request	Bool	%M2202.6	False	True	True	True
	CLOSE HMI request	Bool	%M2202.7	False	True	True	True
	Pause PB	Bool	%M2201.2	False	True	True	True
	OBJECT detected	Bool	%M2201.3	False	True	True	True
	Start Sequence	Bool	%M2201.4	False	True	True	True
	Reset Counter	Bool	%M2201.5	False	True	True	True
	Safe to Move Down	Bool	%M2201.6	False	True	True	True

Figure 1 – PLC tag table

Main Program

Network 1 has two rungs that control whether the inputs will be simulated or if the actual inputs will be mapped to memory bits. This is controlled by a "Simulate" bit. When the bit is on, the simulated inputs are used. When it is off, the actual inputs are used.

Network 2 is the automatic sequence function call. When it is called, the SCL code will run and the sequence will transition through the steps. The "Pause PB" can be used to interrupt the sequence if the operator requires the machine to pause.

Network 3 is an improvement added – "Safe to Move Down". This bit is utilised to make the program look cleaner and more efficient.

Network 4 is "Move Up". The output can be energised in automatic or manual mode, while it can also be energised if it is not safe to move down. The output is turned off once the "Up sensor" is energised.

Network 5 is "Move Down". The output can be energised in automatic or manual mode and it must be safe to move down. The output is turned off once the "Down sensor" is energised.

Network 6 is "Move Left". The output can be energised in automatic or manual mode and it must be in the up position. The output is turned off once the "Left sensor" is energised.

Network 7 is "Move Right". The output can be energised in automatic or manual mode and it must be in the up position. The output is turned off once the "Right sensor" is energised.

Network 8 is "Move In". The output can be energised in automatic or manual mode and it must be in the up position. The output is turned off once the "In sensor" is energised.

Network 9 is "Move Out". The output can be energised in automatic or manual mode and it must be in the up position. The output is turned off once the "Out sensor" is energised.

Network 10 is "Gripper Open". The output can be energised in automatic or manual mode. The output is turned off once the "Open sensor" is energised.

Network 11 is "Gripper Closed". The output can be energised in automatic or manual mode. The output is turned off once the "Closed sensor" is energised.

Network 12 is "Object Detected". The output will be set on by "Object sensor" or "Start PB" and if the step number is less than 3.

Network 13 is "Start Sequence". The output will be energised, after a delay of 2 seconds, if an object is detected.

Network 14 is "Reset Object Detected". "Object Detected" will be reset, after a delay of 1 second, once "Start Sequence" has energised.

Network 15 is "Object Counter". The counter will count up every time an object is detected and will count down when the step number is 8. The count can be reset by the "FSP" or by the "Reset Counter" bit.

In-depth internal documentation can be viewed in figures 6 - 20 in the appendix.

HMI Screens

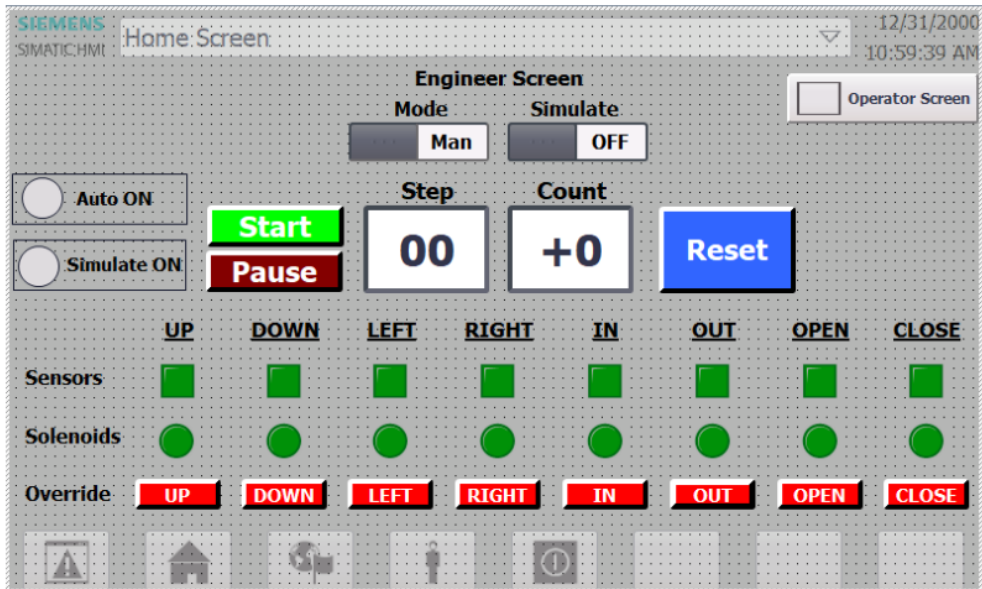


Figure 2 – Engineer Screen

The engineers screen is password protected to inhibit an operator from gaining access to the overrides. When the engineer screen button is pressed a box will appear like the one below and ask for a username and password, which has been set as: User = Engineer and Password = 1234. The manual overrides allow the engineer to check if the robot is functioning correctly. If the solenoid does fire, the sensor light should come on when reached, if not then it shows the sensor is faulty and needs replacing.

A mode can be selected, auto or manual, and simulate can be turned on or off, a light comes on to indicate auto and simulate being on. The step and count number are shown, with a reset button to zero the count. The auto sequence begins when the 'Start' button is pressed and can be paused. To restart the sequence after pause has been pressed the start button must be pressed, this resets the pause bit and sets the start bit as seen in figure 29.

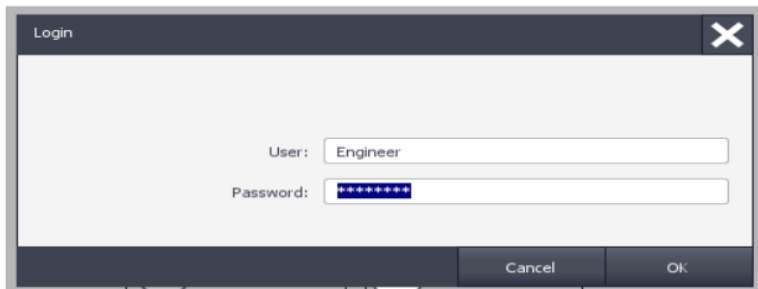


Figure 3 – Password protection

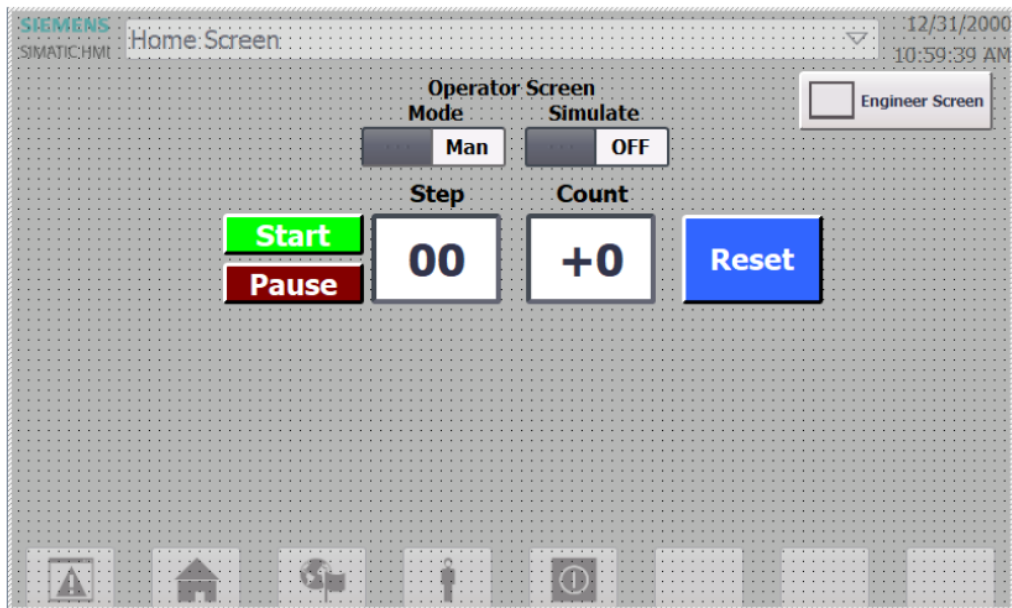


Figure 4 – Operators screen

This screen allows the operator to select the mode and simulation. The screen has start and pause features and the step and count numbers displayed. There is no need for sensor and solenoid indicators on this screen as it could encourage the operator to try fix a fault themselves.

SFC

Excellent SFC

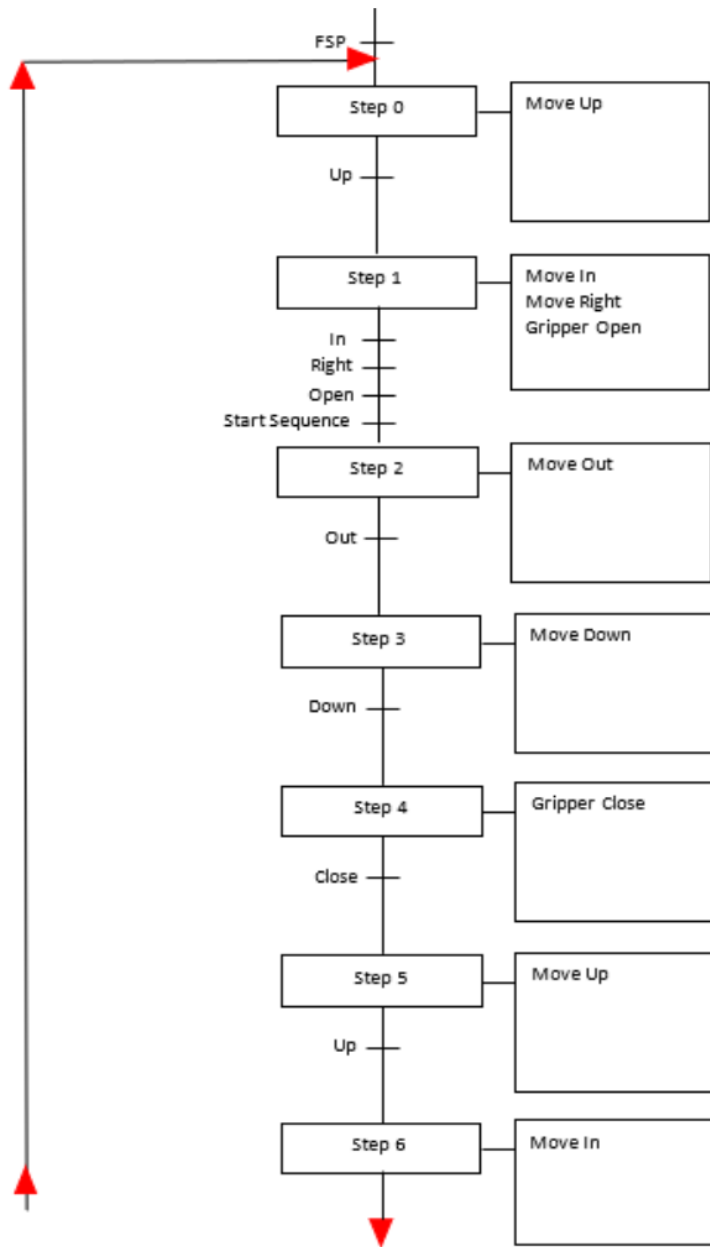


Figure 5 - SFC

Figure 5 shows flow chart was created to outline the steps needed to write the auto sequence. The box in the centre is the step number, with the box on the right being the operation requested and the dash between steps showing what the robot will do for the above step. The complete flow chart can be found in the appendix.

SCL Program

The first "IF" statement compromises the resetting of the step number and all requests, if it is the first scan pulse (FSP) or if auto is off.

The first two steps of the auto sequence are the robot moving to a safe and then home position, the robot is moved up so it cannot cause any damage to the machine and is then moved to its home position. The next step will then be selected by the number of objects in the machine.

If there are 3 objects in the machine, the next step will be step 2. Steps 2 to 9 move the object from station 3 to station 4.

If there are 2 objects in the machine, the next step will be step 10. Steps 10 to 18 move the object from station 2 to station 3.

If there is 1 object in the machine, the next step will be step 19. Steps 19 to 26 move the object from station 1 to station 2.

In-depth internal documentation can be viewed in figures 21 - 28 in the appendix.

4. Simulation Testing

*Excellent
table*

Manual Testing

Test Number	Operation	Expected Results	Actual Results	Pass/Fail	Figure Number
1	In Down, Left & In Position – Move Right	Stay in same position	Did not move	Pass	30
2	In Down, Left & In Position – Move Out	Stay in same position	Did not move	Pass	31
3	In Down, Right & Out Position - Move Left	Stay in same position	Did not move	Pass	32
4	In Down, Right & Out Position - Move In	Stay in same position	Did not move	Pass	33
5	In Up Position – Move Left	Move Left	Moved left	Pass	34
6	In Up Position – Move Right	Move Right	Moved right	Pass	35
7	In Up Position – Move In	Move In	Moved in	Pass	36
8	In Up Position – Move Out	Move Out	Moved out	Pass	37
9	In only Up Position – Move Down	Stay in same position	Did not move	Pass	38
15	Gripper to Close	Gripper Close	Gripper closed	Pass	39
16	Gripper to Open	Gripper Open	Gripper opened	Pass	40

Table 5

All tests passed – showing the interlocks worked. When in the down position the robot was not able to move, and the robot was not able to move down unless it was in the left or right, and in or out position.

Auto Results

Step Number	Expected Result	Actual Result	Passed/Failed	Figure
0	Expected to move Up	Moved up	Passed	41
1	Expected to move in, right and open grippers	Moved: In & Right Grippers opened	Passed	42
2	Expected to move out	Moved out	Passed	43
3	Expected to move down	Moved down	Passed	44
4	Gripper is expected to close	Grippers closed	Passed	45
5	Expected to move up	Moved up	Passed	46
6	Expected to move in	Moved in	Passed	47
7	Expected to move down	Moved down	Passed	48
8	Gripper is expected to open	Gripper opened	Passed	49
9	Expected to move up	Moved up	Passed	50
10	Expected to move left	Moved left	Passed	51
11	Expected to move out	Moved out	Passed	52
12	Expected to move down	Moved down	Passed	53
13	Gripper is expected to close	Grippers closed	Passed	54
14	Expected to move up	Moved up	Passed	55
15	Expected to move right	Moved right	Passed	56
16	Expected to move down	Moved down	Passed	57
17	Gripper is expected to open	Grippers opened	Passed	58
18	Expected to move up	Moved up	Passed	59
19	Expected to move in	Moved in	Passed	60
20	Expected to move left	Moved left	Passed	61
21	Expected to move down	Moved down	Passed	62
22	Gripper is expected to close	Gripper closed	Passed	63

23	Expected to move up	Moved up	Passed	64
24	Expected to move out	Moved out	Passed	65
25	Expected to move down	Moved down	Passed	66
26	Gripper is expected to open	Gripper opened	Passed	67

Table 6

5. Team Working

In a group of 3, we presented our programs to each other. The main program, auto sequence, HMI screens and tag tables were compared. Constructive criticism was given to each member to help improve their program.

The first member of the group did not have a “Reset Count” button on their HMI screens and did not have it in their tag table. This improved their program design and allowed the counter to be reset, as they did not have the option previously.

The second member required extensive help in their program. They were having numerous errors when compiling that were due to incorrect tagging on the HMI. Tags had to be added and then the correct tags chosen for the HMI. Buttons and indicators had to be added while changes had to be made to the main program and auto sequence. This gave them a good grounding to go on and complete the assignment.

On advice from the team members, I decided to remove some indicators from the operator screen as they were not required for the operator to see. This improved the layout of the screen.

An excellent section

*Full justification of text would
improve the presentation of your
paragraphs*

6. Discussion, Conclusions and Recommendations

The project met the specification and included additional benefits such as, added safety and security measures, clear operator interfaces and fault-finding aides. The results of the simulation were as expected; each step performed with no faults. The simulation function allowed the program to be tested, while the practical testing was not able to be completed. The results of the simulation show that the program should run practically without any faults. The simulation function is a good addition to any project, it allows the engineer to be confident their program will perform the required tasks and saves them from carrying out modifications at the commissioning stage, when the machine should be running.

Such problems were encountered when first writing the program as when in simulation the step number was jumping from 18 to 20. On investigation the fault was the robot being asked to move 'In' on step 19 when it was already in the 'In' position. The use of the SFC allows you to logically map out each step in a concise way to meet the specification, doing the work before you transfer it into SCL. The above fault may have occurred due to a typo in either the SFC or SCL code.

The program has been thoroughly tested and set up to be as reliable as possible. The sequence allows no room for mistakes with the test results showing that the robot cannot make any unsafe movements and any possible fault occurrences have been accounted for and interlocks have been added to counter this.

The maintenance of the robot should be relatively simple with the use of the HMI 'Engineers screen'. While monitoring the sequence and watching the indicators on the HMI of the sensors and solenoids, the engineer can see any step being missed, a faulty sensor not working or a solenoid not actuating. There has also been interlocks added which will help the engineer see why the robot is not moving down for instance.

The robot meets the customer needs as it performs the functions needed and has been tested rigorously to ensure it is acceptable for a customer. The specification for the robot is to move objects around the stations as they enter and exit the machine. The program can be operated in automatic and manual modes and the user has all information needed to operate the machine on the operator HMI screen, showing the current step number and the number of objects in the machine. The operator can start and pause the sequence, in the event of a fault for example. The operator does not require the fault-finding aides, as this could cause the operator to try fixing the fault when they do not have the required training.

An improvement that was made to the program was to include the "Safe to Move Down" bit. This improved the design of the program, giving a cleaner and more professional look. This bit replaced the need for several rungs on some networks. Another alteration to the program could improve the reliability of the machine. The robot must be in the "In", "Right" and "Open" positions before "Start Sequence" comes on, as it only energises for 1 second. If "Start Sequence" is energised before the robot is in all three positions the sequence will stay at step 1 instead of progressing. This could be fixed by having all three conditions in series added into the sequence for "Start Sequence".

A third improvement could be the addition of an object sensor on the exit chute to perform the count down function instead of a step number. A sensor at the bottom of the chute would remove all doubt that the object has left the machine.

Bibliography

Kamel, K. & Kamel, E., 2014. *Programmable Logic Controllers*. 1st ed. New York: McGraw-Hill.

Nanping, D., Wei, S. & Guanling, Z., 2009. Application of S7-400 and WinCC in Dense Phase System. *Third International Symposium on Intelligent Information Technology Application Workshops*, Volume III, pp. 352-355. DOI: 10.1109/IITAW.2009.120.

Parmley Graham, 2020. *Parmley Graham Web-store*. [Online]

Available at: <https://www.parmley-graham.co.uk/automation/siemens/io-systems-for-in-cabinets/et200sp>

[Accessed 17 12 2020].

Pegaia, D., 2014. *Siemens Industry Online Support*. [Online]

Available at: <https://support.industry.siemens.com/tf/WW/en/posts/what-is-the-different-s7-1200-and-s7-1500/117918?page=0&pageSize=10>

[Accessed 17 12 2020].

RS-Online, 2002. *RS-Online*. [Online]

Available at: <https://docs.rs-online.com/f87c/0900766b8036eb75.pdf>

[Accessed 05 01 2021].

Siemens, 2005. *Farnell*. [Online]

Available at: <http://www.farnell.com/datasheets/49433.pdf>

[Accessed 20 01 2021].

Siemens, 2015. *Farnell*. [Online]

Available at: <http://www.farnell.com/datasheets/1937472.pdf>

[Accessed 10 01 2021].

Siemens, 2015. *RS Online*. [Online]

Available at: <https://docs.rs-online.com/4ed5/0900766b81397276.pdf>

[Accessed 15 01 2021].

Siemens, 2015. *Sermax*. [Online]

Available at: <http://sermax.my/img/products/industrial-products/ProgrammableLogicControllers/download/01-SIEMENS-S7-1500-Datasheet.pdf>

[Accessed 13 01 2021].

Siemens, 2017. *Automation Siemens*. [Online]

Available at:

https://www.automation.siemens.com/w2/efiles/pics7/pdf/76/ST400_2017_en_Web.pdf

[Accessed 14 01 2021].

Siemens, 2020. *Automation24*. [Online]

Available at: https://media.automation24.com/datasheet/en/6ED10521MD080BA0_en.pdf

[Accessed 13 01 2021].

Siemens, 2020. *LOGO! Manual*. [Online]

Available at: https://cache.industry.siemens.com/dl/dl-media/041/109741041/att_1035691/v5/138938428427/en-US/index.html#treeId=74dd537fe16916bed60e5524f9c52379

[Accessed 16 12 2020].

Appendix

▼ Network 1: Switching between actual and simulated inputs

- ▼ The simulate bit (%M2000.0) allows the operator to switch between simulated and actual inputs. When the simulate bit is OFF, %FC1 is called and the actual inputs are read and mapped to memory bits. When the simulate bit is ON, %FC3 is called and the simulated inputs are read.

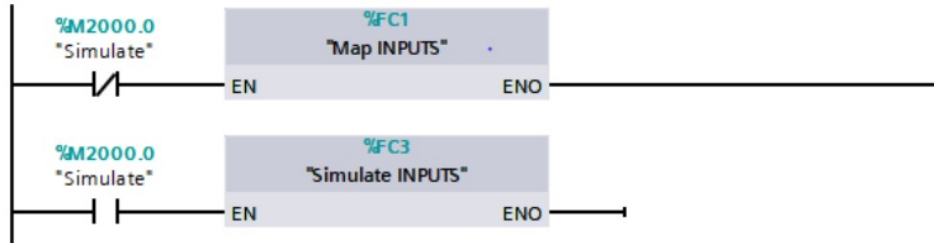


Figure 6 – Network 1 Mapped or Simulate Inputs

▼ Network 2: Automatic sequence function call

- ▼ %FC2 is the auto sequence program written in SCL. The auto sequence can be paused as it is moving through steps when the pause PB bit (%M2201.2) is OFF. When the pause PB bit is ON the auto sequence will continue.

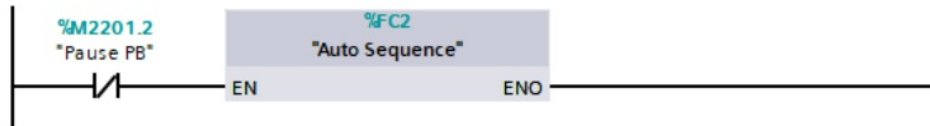


Figure 7 – Network 2 Auto Sequence

▼ Network 3: Safe to Move Down

- ▼ Before the robot can move down it needs to be in a safe position. It must either be in the left OR right position AND the in OR out position before it can move down. The safe to move down bit (%M2201.6) is used in network 4 (Move Up) and network 5 (Move Down).

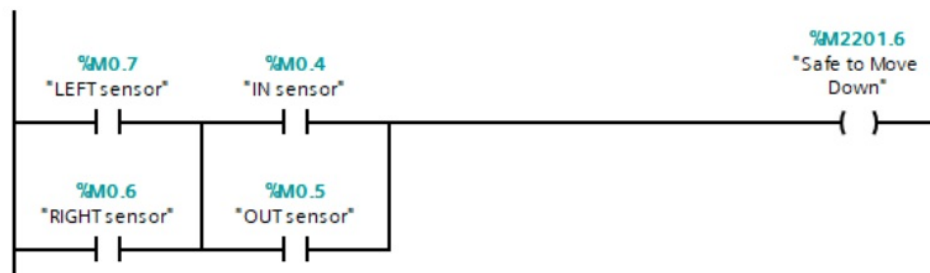


Figure 8 – Network 3 Safe to Move Down

Network 4: Move Up Solenoid

- ▼ The up solenoid (%Q0.1) can be energised by 3 different rungs. If the robot is in auto mode (%M2201.0) AND the up auto request (%M2200.0) is on AND the up sensor (%M0.0) is off the output will energise.
If the robot is in manual (auto off) AND the up HMI button (%M2202.0) has been pressed and %M0.0 is off the output will energise.
If the robot is not safe to move down (%M2201.6) AND is not already in the up position (%M0.0) then the output will energise.

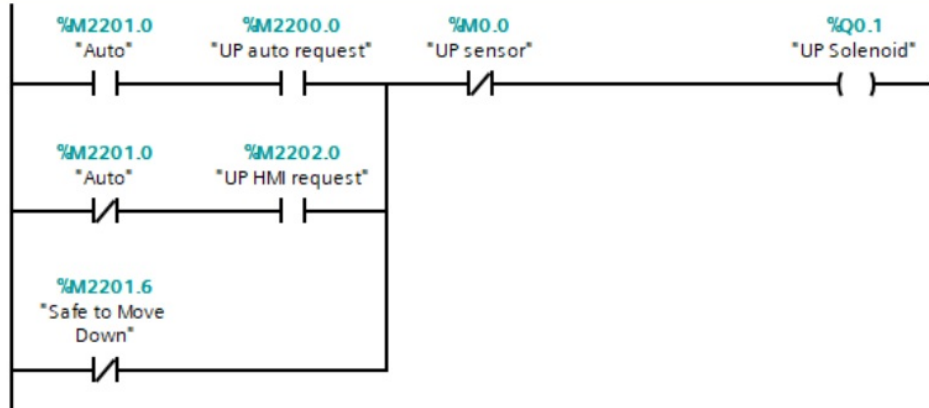


Figure 9 – Network 4 Move Up

Network 5: Move Down Solenoid

- ▼ The down solenoid (%Q0.0) can be energised by 2 rungs. If auto (%M2201.0) is selected AND the down auto request (%M2200.1) is on AND it is safe to move down (%M2201.6) AND the robot is not in the down position (%M0.1) the output will energise.
If manual is selected (not %M2201.0) AND the down HMI button (%M2202.1) has been pressed AND %M2201.6 is on AND %M0.1 is off the output will energise.

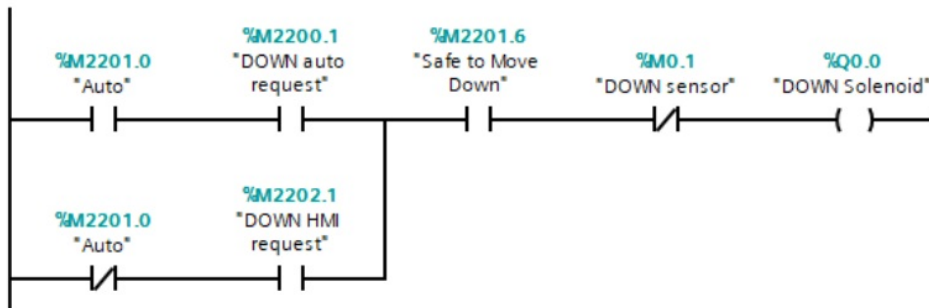


Figure 10 – Network 5 Move Down

▼ Network 6: Move Left Solenoid

- ▼ The left solenoid (%Q0.6) can be energised by 2 rungs. If auto (%M2201.0) is selected AND the left auto request (%M2200.2) is on AND the robot is in the up position (%M0.0) AND the robot is not in the left position (%M0.7) the output will energise.
If manual is selected (not %M2201.0) AND the left HMI button (%M2202.2) has been pressed AND %M0.0 is on AND %M0.7 is off the output will energise.
The Up Sensor needs to be energised so that the robot will only move left if it is in the up position.

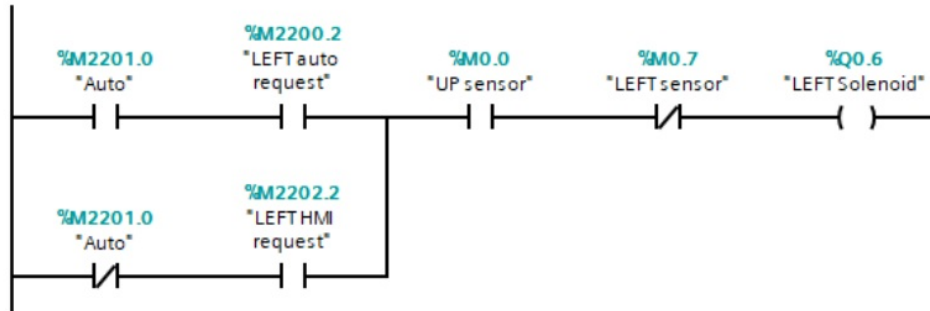


Figure 11 – Network 6 Move Left

▼ Network 7: Move Right Solenoid

- ▼ The right solenoid (%Q0.7) can be energised by 2 rungs. If auto (%M2201.0) is selected AND the right auto request (%M2200.3) is on AND the robot is in the up position (%M0.0) AND the robot is not in the right position (%M0.6) the output will energise.
If manual is selected (not %M2201.0) AND the right HMI button (%M2202.3) has been pressed AND %M0.0 is on AND %M0.6 is off the output will energise.
The Up Sensor needs to be energised so that the robot will only move right if it is in the up position.

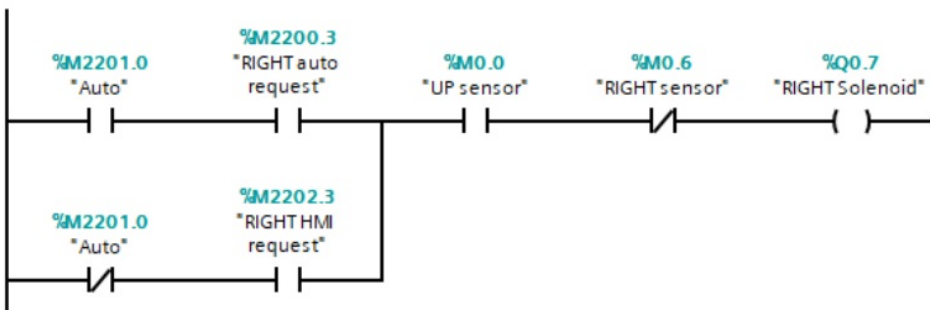


Figure 12 – Network 7 Move Right

Network 8: Move In Solenoid

- ▼ The in solenoid (%Q0.5) can be energised by 2 rungs. If auto (%M2201.0) is selected AND the in auto request (%M2200.2) is on AND the robot is in the up position (%M0.0) AND the robot is not in the in position (%M0.4) the output will energise.
If manual is selected (not %M2201.0) AND the in HMI button (%M2202.4) has been pressed AND %M0.0 is on AND %M0.4 is off the output will energise.
The Up Sensor needs to be energised so that the robot will only move in, if it is in the up position.

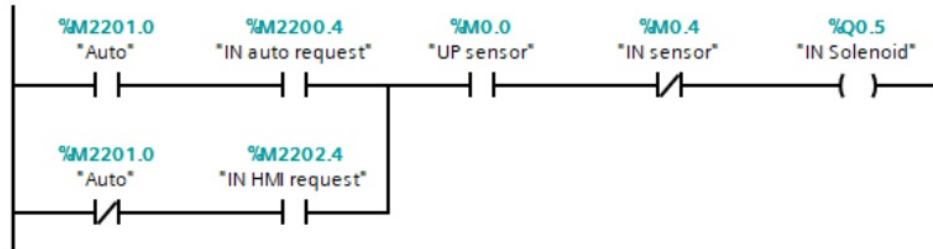


Figure 13 – Network 8 Move In

Network 9: Move Out Solenoid

- ▼ The out solenoid (%Q0.4) can be energised by 2 rungs. If auto (%M2201.0) is selected AND the out auto request (%M2200.5) is on AND the robot is in the up position (%M0.0) AND the robot is not in the out position (%M0.5) the output will energise.
If manual is selected (not %M2201.0) AND the out HMI button (%M2202.5) has been pressed AND %M0.0 is on AND %M0.5 is off the output will energise.
The Up Sensor needs to be energised so that the robot will only move out if it is in the up position.

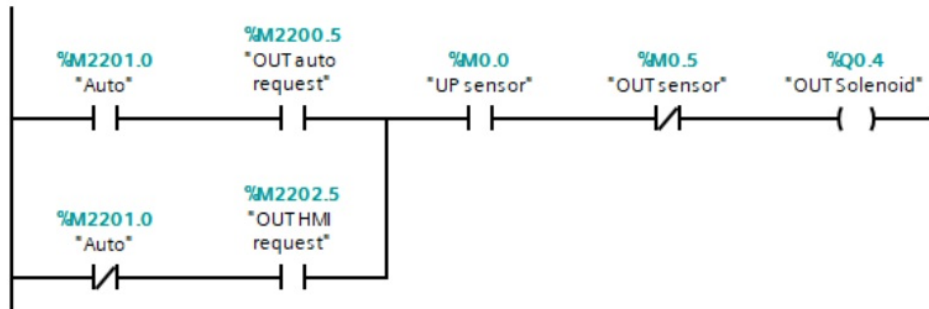


Figure 14 – Network 9 Move Out

Network 10: Gripper Open Solenoid

- ▼ The open solenoid (%Q0.3) can be energised by 2 rungs. If auto (%M2201.0) is selected AND the open auto request (%M2200.6) is on AND the gripper is not open (%M0.2) the output will energise. If manual is selected (not %M2201.0) AND the open HMI button (%M2202.6) has been pressed AND %M0.2 is off the output will energise. Once the gripper is open, the open sensor will energise and break the circuit, turning the output off. The open solenoid opens the grippers claws to allow it to drop the objects at the stations.

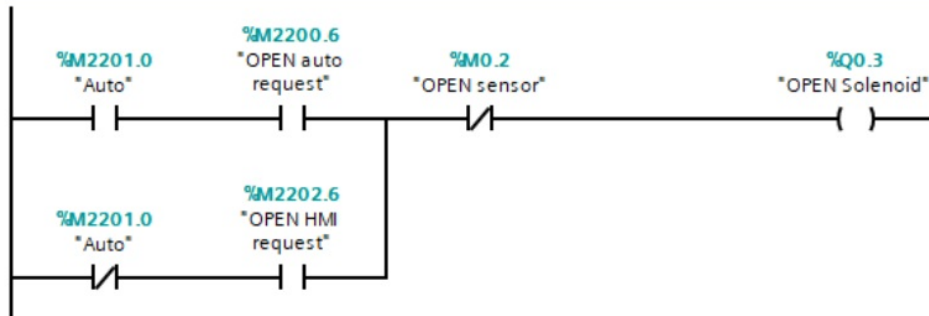


Figure 15 – Network 10 Gripper Open

Network 11: Gripper Close Solenoid

- ▼ The close solenoid (%Q0.2) can be energised by 2 rungs. If auto (%M2201.0) is selected AND the close auto request (%M2200.7) is on AND the gripper is not closed (%M0.3) the output will energise. If manual is selected (not %M2201.0) AND the close HMI button (%M2202.7) has been pressed AND %M0.3 is off the output will energise. Once the gripper is closed, the closed sensor will energise and break the circuit, turning the output off. The close solenoid closes the grippers claws to allow it to pick up the objects at the stations.

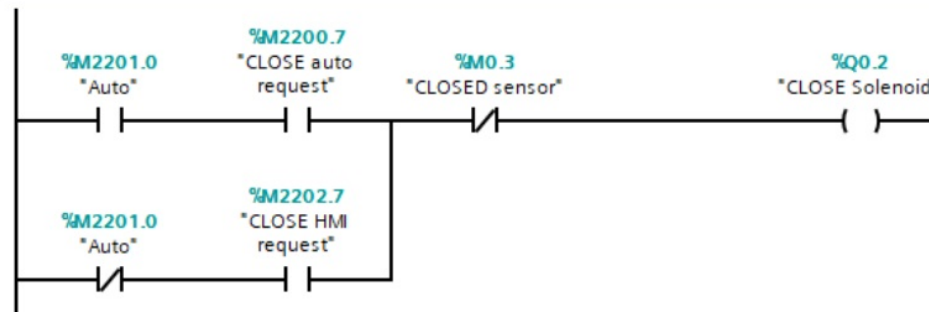


Figure 16 – Network 11 Gripper Close

Network 12: Object Detected

- ▼ The object detected bit (%M2201.3) can be SET on by 2 rungs. If an object is sensed by the object sensor (%M1.5) then %M2201.3 will be SET on. If the start PB (%M2201.1) has been pressed AND the step number (%MW2010) is less than 3 then %M2201.3 will be SET on. If the step number is 4 or more, the bit %M2201.3 will not be SET, even if the start PB is pressed. When the bit %M2201.3 is SET on, it can only turn off by being RESET, in network 14. The bit %M1.5 may only pulse on for 1 second as the object rolls past but the bit %M2201.3 will still be SET.

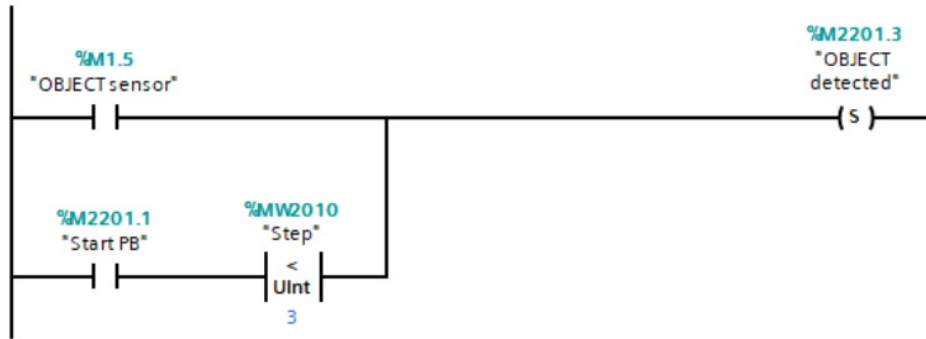


Figure 17 – Network 12 Object Detected

Network 13: Start counter sequence

- ▼ If an object is detected (%M2201.3) an on-delay timer (%DB2) will energise. After 2 seconds has past the timer will output and energise the start sequence bit (%M2201.4). %M2201.4 will stay energised until bit %M2201.3 is RESET, in network 14.



Figure 18 – Network 13 Start Sequence

Network 14: Reset object detected

- ▼ If start sequence (%M2201.4) is on, the on-delay timer (%DB3) will energise. After 1 second has past the timer will output and RESET the object detected bit (%M2201.3). %M2201.3 will stay off until it is SET on in network 12. When %M2201.3 is RESET, %M2201.4 will turn off.



Figure 19 – Network 14 Reset Object Detected

Network 15: Object counter

- ▼ The object counter (%DB1) is an up-down counter. The counter will count up in increments of 1 when an object is detected (%M2201.3), as this is when an object enters the machine into station 1. The counter will count down in decrements of 1 when the step number (%MW2010) is equal to 8, as this is when an object is dropped onto station 4 and down the exit chute. The counter is reset on the first scan pulse (%M2100.0) of the PLC program OR by the reset counter button (%M2201.5) on the HMI.

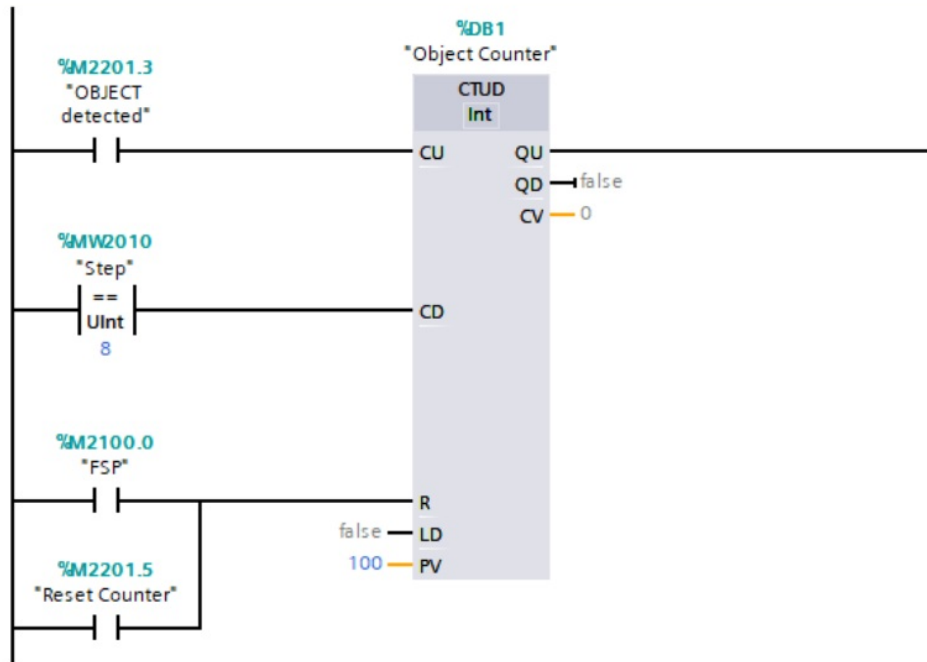


Figure 20 – Network 15 Object Counter

```

1 IF "FSP" = 1 OR "Auto" = 0 THEN                                //if it is the FSP or Auto is off then:
2     "Step" := 0;                                              //reset all values to zero
3     "Next_Step" := 0;                                         //this includes the step number and auto requests
4     "DOWN auto request" := 0;                                 //this ensures the robot doesn't move unexpectedly
5     "UP auto request" := 0;
6     "IN auto request" := 0;
7     "OUT auto request" := 0;
8     "LEFT auto request" := 0;
9     "RIGHT auto request" := 0;
10    "OPEN auto request" := 0;
11    "CLOSE auto request" := 0;
12 END_IF;

```

Figure 21 – Reset All Bits

```

14 IF "Auto" = 1 THEN                                     //auto can be selected on the HMI screens
15     CASE "Step" OF
16         0: //move up                                     //on initial start-up OR when the sequence has finished
17             "UP auto request" := 1;                     //step 0 will move the robot to the UP position
18             IF "UP sensor" = 1 THEN                     //Up sensor = 1 once robot has moved to UP position
19                 "UP auto request" := 0;                 //when up sensor is on, up request reset to 0
20                 "Next_Step" := 1;
21             END_IF;
22         1: //move to home position of station 4           //once the robot has moved UP, it will move HOME
23             "IN auto request" := 1;                     //before carrying on through the rest of sequence
24             "RIGHT auto request" := 1;
25             "OPEN auto request" := 1;
26             IF ("IN sensor" = 1) AND                     //In sensor = 1 once robot has moved to IN position
27                 ("RIGHT sensor" = 1) AND                 //Right sensor = 1 once robot has moved to RIGHT position
28                 ("OPEN sensor" = 1) AND                 //Open sensor = 1 when the gripper is OPEN
29                 ("Start Sequence" = 1) THEN             //Start sequence = 1 when network 13 in [OB1] is on
30                 "IN auto request" := 0;
31                 "RIGHT auto request" := 0;               //step 1 requests reset to 0 once conditions are met
32                 "OPEN auto request" := 0;
33                 IF ("Object Counter".CV = 1) THEN        //if there is 1 object in the machine
34                     "Next_Step" := 20;                 //the sequence will start at step 20
35                 END_IF;
36                 IF ("Object Counter".CV = 2) THEN        //if there is 2 objects in the machine
37                     "Next_Step" := 10;                 //the sequence will start at step 10
38                 END_IF;
39                 IF ("Object Counter".CV = 3) THEN        //if there is 3 objects in the machine
40                     "Next_Step" := 2;                 //the sequence will start at step 2
41                 END_IF;
42             END_IF;

```

Figure 22 – Step 0 to 1

```

43         2: //move object from station 3 to station 4     //there will be 3 objects in the machine on step 2
44             //move out                                    //robot will move out to above station 3
45             "OUT auto request" := 1;                     //out request is energised and is part of network 9 sequence in [OB1]
46             IF ("OUT sensor" = 1) THEN                   //out sensor = 1 once robot has moved to OUT position
47                 "OUT auto request" := 0;                 //when out sensor is on, out request reset to 0
48                 "Next_Step" := 3;
49             END_IF;
50         3: //move down                                    //robot will move down to station 3
51             "DOWN auto request" := 1;                     //down request is energised and is part of network 5 sequence in [OB1]
52             IF ("DOWN sensor" = 1) THEN                   //down sensor = 1 once robot has moved to DOWN position
53                 "DOWN auto request" := 0;                 //when down sensor is on, down request reset to 0
54                 "Next_Step" := 4;
55             END_IF;
56         4: //grripper close                                //grripper will pick up object from station 3
57             "CLOSE auto request" := 1;                     //close request is energised and is part of network 11 sequence in [OB1]
58             IF ("CLOSED sensor" = 1) THEN                 //closed sensor = 1 once gripper has CLOSED
59                 "CLOSE auto request" := 0;                 //when closed sensor is on, close request reset to 0
60                 "Next_Step" := 5;
61             END_IF;
62         5: //move up                                       //robot will move up above station 3
63             "UP auto request" := 1;                     //up request is energised and is part of network 4 sequence in [OB1]
64             IF ("UP sensor" = 1) THEN                     //up sensor = 1 once robot has moved to UP position
65                 "UP auto request" := 0;                 //when up sensor is on, up request reset to 0
66                 "Next_Step" := 6;
67             END_IF;
68         6: //move in                                       //robot will move in to above station 4
69             "IN auto request" := 1;                     //in request is energised and is part of network 8 sequence in [OB1]
70             IF ("IN sensor" = 1) THEN                     //in sensor = 1 once robot has moved to IN position
71                 "IN auto request" := 0;                 //when in sensor is on, in request reset to 0
72                 "Next_Step" := 7;
73             END_IF;

```

Figure 23 – Step 2 to 6

```

74 | 7: //move down                                //robot will move down to station 4
75 |   "DOWN auto request" := 1;                    //down request is energised and is part of network 5 sequence in [OB1]
76 |   IF ("DOWN sensor" = 1) THEN                  //down sensor = 1 once robot has moved to DOWN position
77 |     "DOWN auto request" := 0;                  //when down sensor is on, down request reset to 0
78 |     "Next_Step" := 8;
79 |   END_IF;
80 | 8: //gripper open                              //gripper will drop object on station 4 and down exit chute
81 |   "OPEN auto request" := 1;                    //open request is energised and is part of network 10 sequence in [OB1]
82 |   IF ("OPEN sensor" = 1) THEN                  //open sensor = 1 when the gripper is OPEN
83 |     "OPEN auto request" := 0;                  //when open sensor is on, open request reset to 0
84 |     "Next_Step" := 9;
85 |   END_IF;
86 | 9: //move up                                    //robot will move up above station 4 before next set of actions
87 |   "UP auto request" := 1;                      //up request is energised and is part of network 4 sequence in [OB1]
88 |   IF ("UP sensor" = 1) THEN                    //up sensor = 1 once robot has moved to UP position
89 |     "UP auto request" := 0;                    //when up sensor is on, up request reset to 0
90 |     "Next_Step" := 10;
91 |   END_IF;

```

Figure 24 – Step 7 to 9

```

92 | 10: //move object from station 2 to station 3 //there will be 2 objects in the machine at step 10
93 |   //move left                                //robot will move left to station 1
94 |   "LEFT auto request" := 1;                   //left request is energised and is part of network 6 sequence in [OB1]
95 |   IF ("LEFT sensor" = 1) THEN                 //left sensor = 1 once robot has moved to LEFT position
96 |     "LEFT auto request" := 0;                 //when left sensor is on, left request reset to 0
97 |     "Next_Step" := 11;
98 |   END_IF;
99 | 11: //move out                                //robot will move out to station 2
100 |   "OUT auto request" := 1;                    //out request is energised and is part of network 9 sequence in [OB1]
101 |   IF ("OUT sensor" = 1) THEN                  //out sensor = 1 once robot has moved to OUT position
102 |     "OUT auto request" := 0;                  //when out sensor is on, out request reset to 0
103 |     "Next_Step" := 12;
104 |   END_IF;
105 | 12: //move down                              //robot will move down to station 2
106 |   "DOWN auto request" := 1;                   //down request is energised and is part of network 5 sequence in [OB1]
107 |   IF ("DOWN sensor" = 1) THEN                 //down sensor = 1 once robot has moved to DOWN position
108 |     "DOWN auto request" := 0;                 //when down sensor is on, down request reset to 0
109 |     "Next_Step" := 13;
110 |   END_IF;
111 | 13: //gripper close                          //gripper will pick up object from station 2
112 |   "CLOSE auto request" := 1;                  //close request is energised and is part of network 11 sequence in [OB1]
113 |   IF ("CLOSE sensor" = 1) THEN                //closed sensor = 1 once gripper has CLOSED
114 |     "CLOSE auto request" := 0;                //when closed sensor is on, close request reset to 0
115 |     "Next_Step" := 14;
116 |   END_IF;
117 | 14: //move up                                //robot will move up above station 2
118 |   "UP auto request" := 1;                      //up request is energised and is part of network 4 sequence in [OB1]
119 |   IF ("UP sensor" = 1) THEN                    //up sensor = 1 once robot has moved to UP position
120 |     "UP auto request" := 0;                    //when up sensor is on, up request reset to 0
121 |     "Next_Step" := 15;
122 |   END_IF;

```

Figure 25 – Step 10 to 14

```

123 | 15: //move right                             //robot will move right to station 3
124 |   "RIGHT auto request" := 1;                  //right request is energised and is part of network 7 sequence in [OB1]
125 |   IF ("RIGHT sensor" = 1) THEN                //right sensor = 1 once robot has moved to RIGHT position
126 |     "RIGHT auto request" := 0;                //when right sensor is on, right request reset to 0
127 |     "Next_Step" := 16;
128 |   END_IF;
129 | 16: //move down                              //robot will move down to station 3
130 |   "DOWN auto request" := 1;                   //down request is energised and is part of network 5 sequence in [OB1]
131 |   IF ("DOWN sensor" = 1) THEN                 //down sensor = 1 once robot has moved to DOWN position
132 |     "DOWN auto request" := 0;                 //when down sensor is on, down request reset to 0
133 |     "Next_Step" := 17;
134 |   END_IF;
135 | 17: //gripper open                          //gripper will drop object on station 3
136 |   "OPEN auto request" := 1;                   //open request is energised and is part of network 10 sequence in [OB1]
137 |   IF ("OPEN sensor" = 1) THEN                 //open sensor = 1 when the gripper is OPEN
138 |     "OPEN auto request" := 0;                 //when open sensor is on, open request reset to 0
139 |     "Next_Step" := 18;
140 |   END_IF;
141 | 18: //move up                                //robot will move up above station 3 before next set of actions
142 |   "UP auto request" := 1;                      //up request is energised and is part of network 4 sequence in [OB1]
143 |   IF ("UP sensor" = 1) THEN                    //up sensor = 1 once robot has moved to UP position
144 |     "UP auto request" := 0;                    //when up sensor is on, up request reset to 0
145 |     "Next_Step" := 19;
146 |   END_IF;

```

Figure 26 – Step 15 to 18

```

147 19: //move object from station 1 to station 2 //there will be 1 object in the machine at step 19
148 //move in //robot will move in to above station 4
149 "IN auto request" := 1; //in request is energised and is part of network 8 sequence in [OB1]
150 IF ("IN sensor" = 1) THEN //in sensor = 1 once robot has moved to IN position
151 "IN auto request" := 0; //when in sensor is on, in request reset to 0
152 "Next_Step" := 20;
153 END_IF;
154 20: //move left //robot will move left to station 1
155 "LEFT auto request" := 1; //left request is energised and is part of network 6 sequence in [OB1]
156 IF ("LEFT sensor" = 1) THEN //left sensor = 1 once robot has moved to LEFT position
157 "LEFT auto request" := 0; //when left sensor is on, left request reset to 0
158 "Next_Step" := 21;
159 END_IF;
160 21: //move down //robot will move down to station 1
161 "DOWN auto request" := 1; //down request is energised and is part of network 5 sequence in [OB1]
162 IF ("DOWN sensor" = 1) THEN //down sensor = 1 once robot has moved to DOWN position
163 "DOWN auto request" := 0; //when down sensor is on, down request reset to 0
164 "Next_Step" := 22;
165 END_IF;
166 22: //gripper close //gripper will pick up object from station 1
167 "CLOSE auto request" := 1; //close request is energised and is part of network 11 sequence in [OB1]
168 IF ("CLOSED sensor" = 1) THEN //closed sensor = 1 once gripper has CLOSED
169 "CLOSE auto request" := 0; //when closed sensor is on, close request reset to 0
170 "Next_Step" := 23;
171 END_IF;
172 23: //move up //robot will move up above station 1
173 "UP auto request" := 1; //up request is energised and is part of network 4 sequence in [OB1]
174 IF ("UP sensor" = 1) THEN //up sensor = 1 once robot has moved to UP position
175 "UP auto request" := 0; //when up sensor is on, up request reset to 0
176 "Next_Step" := 24;
177 END_IF;

```

Figure 27 – Step 19 to 23

```

178 24: //move out //robot will move out to station 2
179 "OUT auto request" := 1; //out request is energised and is part of network 9 sequence in [OB1]
180 IF ("OUT sensor" = 1) THEN //out sensor = 1 once robot has moved to OUT position
181 "OUT auto request" := 0; //when out sensor is on, out request reset to 0
182 "Next_Step" := 25;
183 END_IF;
184 25: //move down //robot will move down to station 2
185 "DOWN auto request" := 1; //down request is energised and is part of network 5 sequence in [OB1]
186 IF ("DOWN sensor" = 1) THEN //down sensor = 1 once robot has moved to DOWN position
187 "DOWN auto request" := 0; //when down sensor is on, down request reset to 0
188 "Next_Step" := 26;
189 END_IF;
190 26: //gripper open //gripper will drop object on station 2
191 "OPEN auto request" := 1; //open request is energised and is part of network 10 sequence in [OB1]
192 IF ("OPEN sensor" = 1) THEN //open sensor = 1 when the gripper is OPEN
193 "OPEN auto request" := 0; //when open sensor is on, open request reset to 0
194 "Next_Step" := 0; //sequence restarts at step 0 and skips to a section
195 END_IF; //depending on number of objects in the machine
196
197 END_CASE; //auto sequence is written inside this CASE statement
198 "Step" := "Next_Step";
199 END_IF;

```

Figure 28 – Step 24 to 26

Press	SetBit	
Release	Tag (Input/output)	Start PB
Activate	ResetBit	
Deactivate	Tag (Input/output)	Pause PB

Figure 29 – Set Start PB and Reset Pause PB

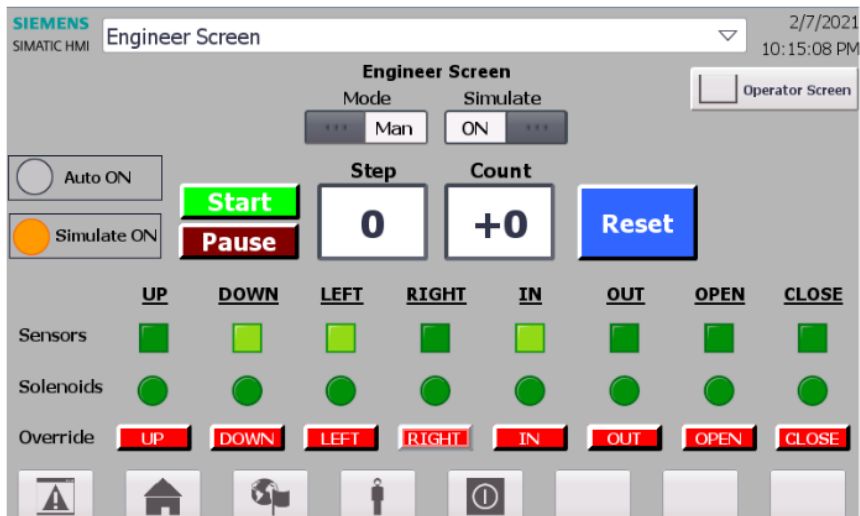


Figure 30 – Move right when down

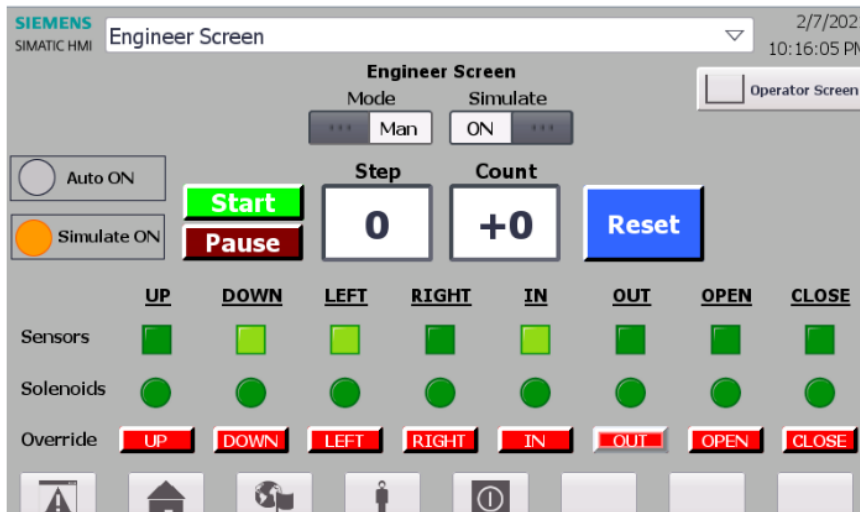


Figure 31 – Move out when down

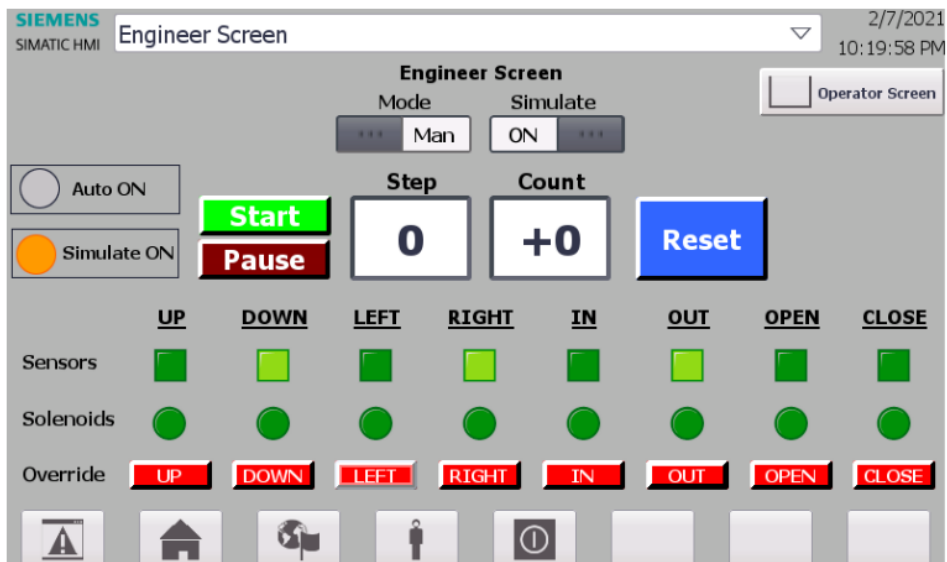


Figure 32 – Move left when down

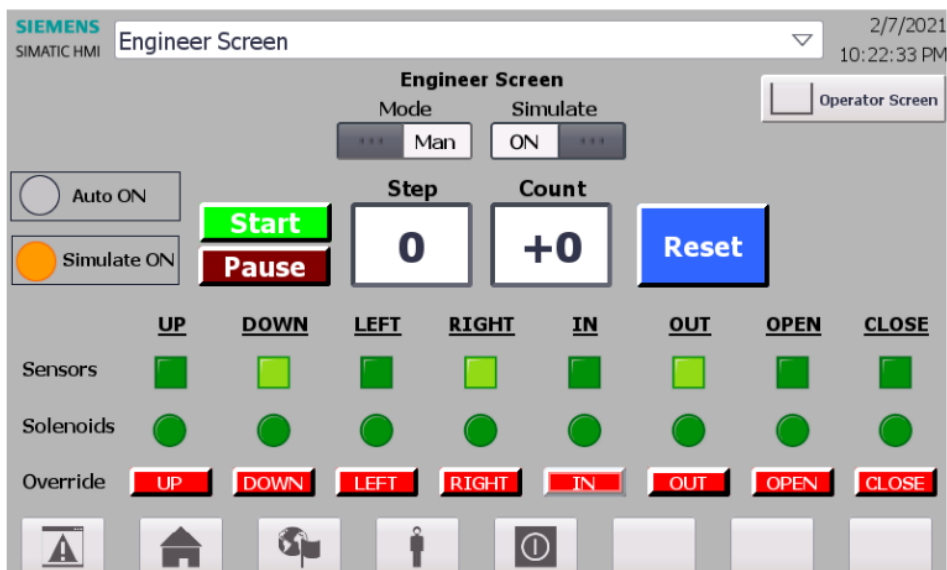


Figure 33 – Move in when down

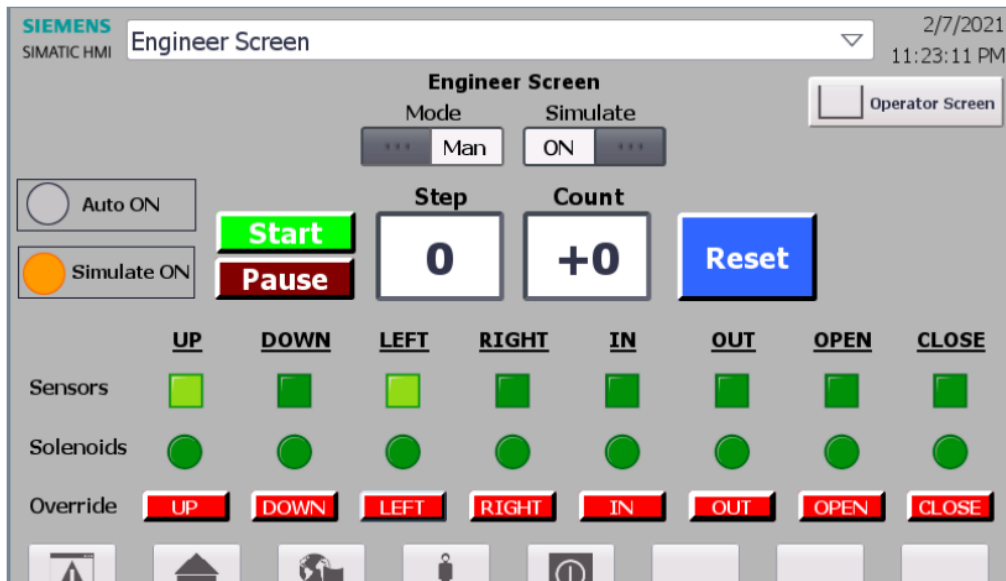


Figure 34 – Move left

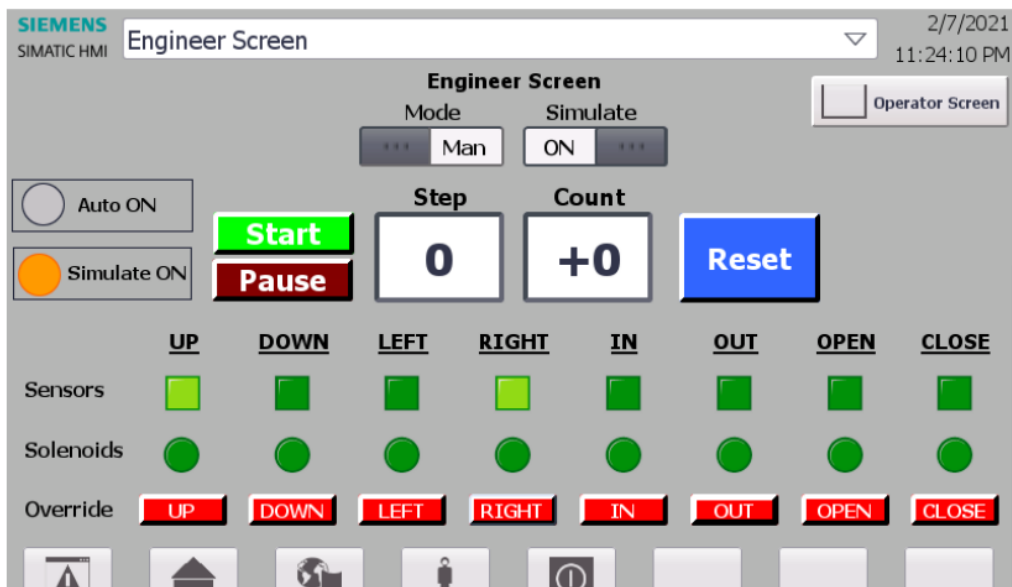


Figure 35 – Move right

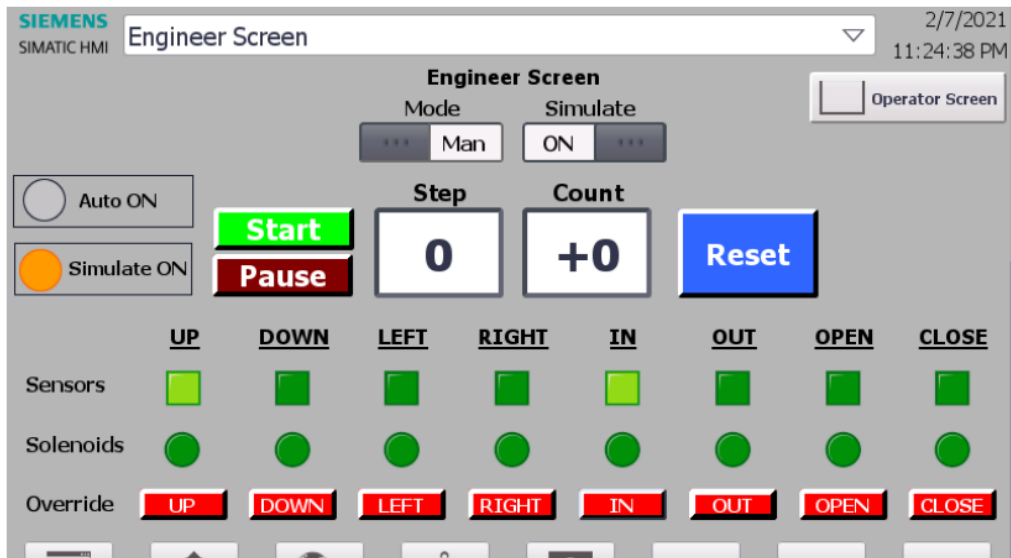


Figure 36 – Move in

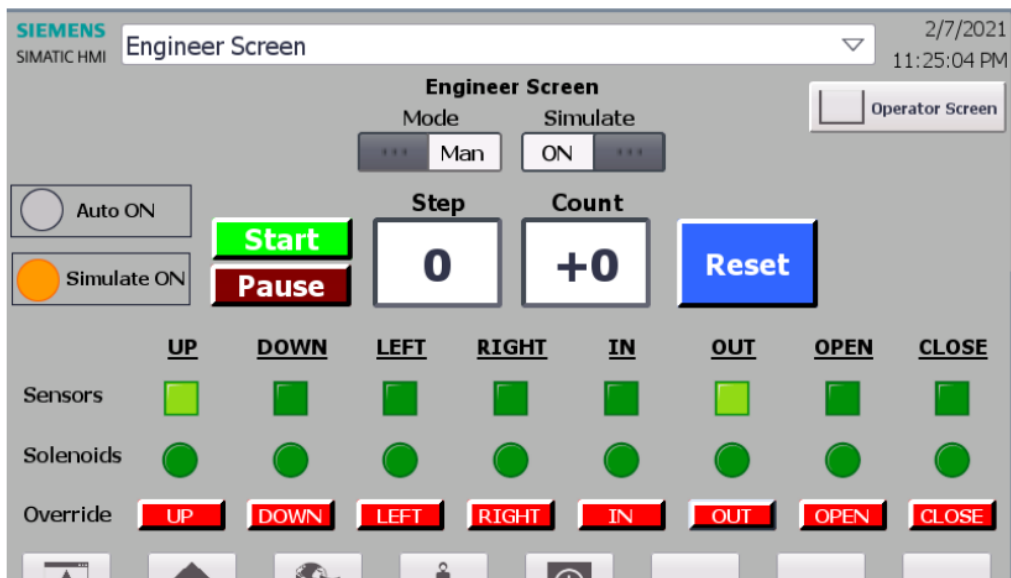


Figure 37 – Move out

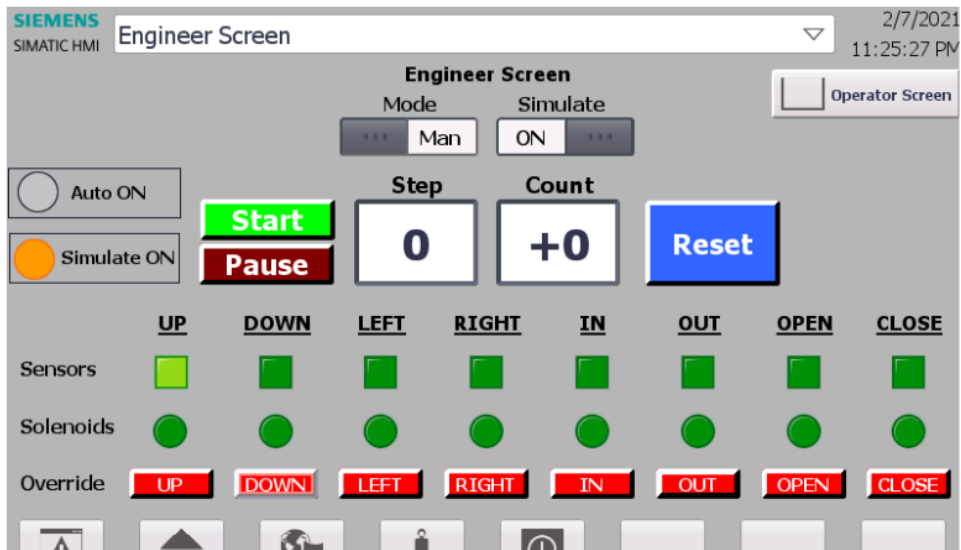


Figure 38 – Move down when unsafe

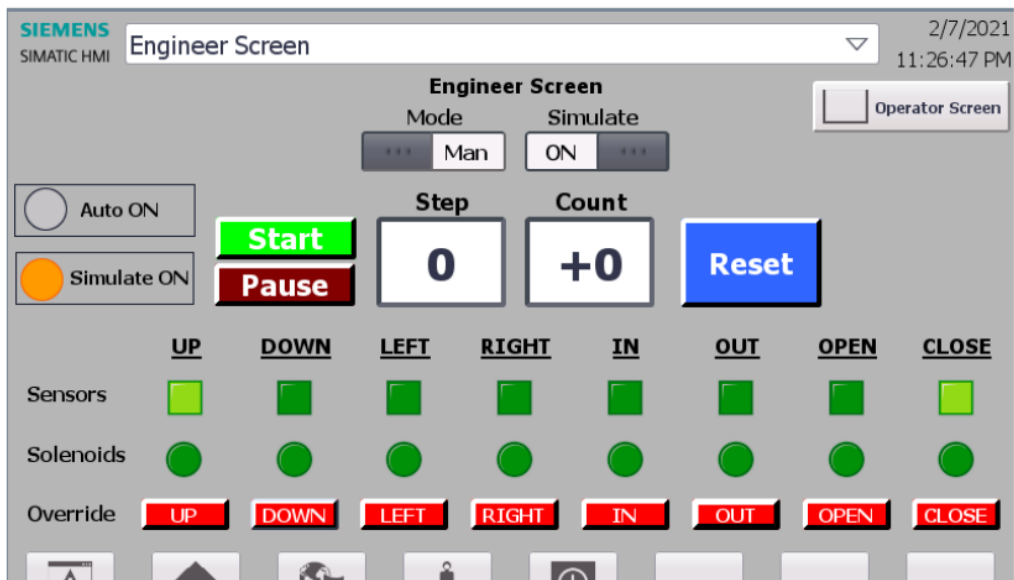


Figure 39 – Gripper close

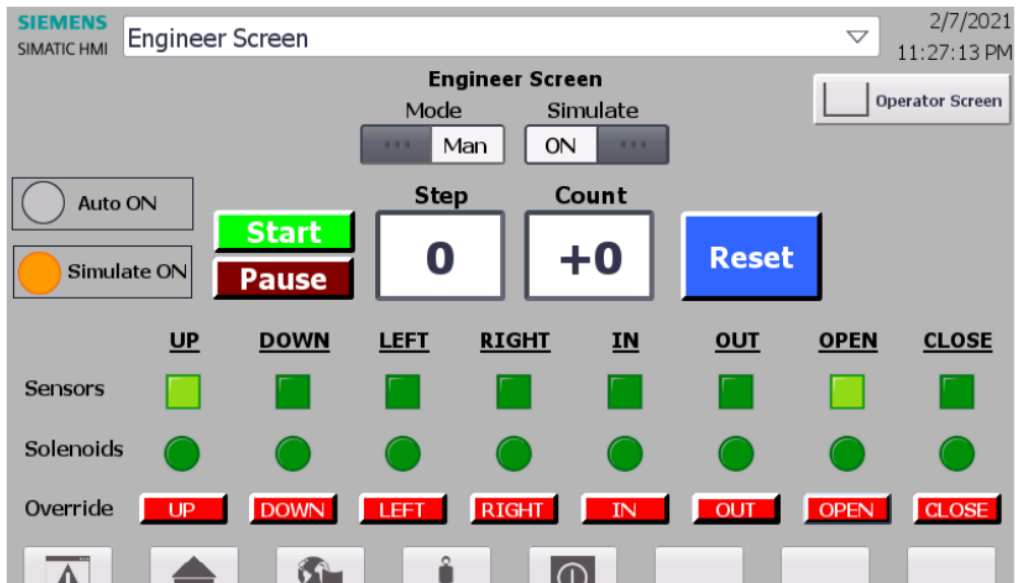


Figure 40 – Gripper open

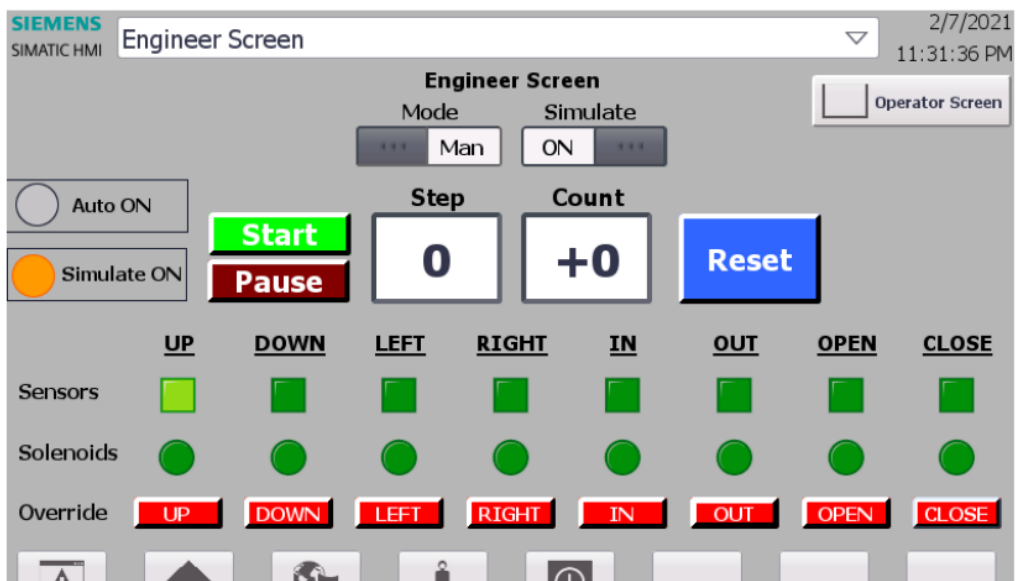


Figure 41 – Step 0

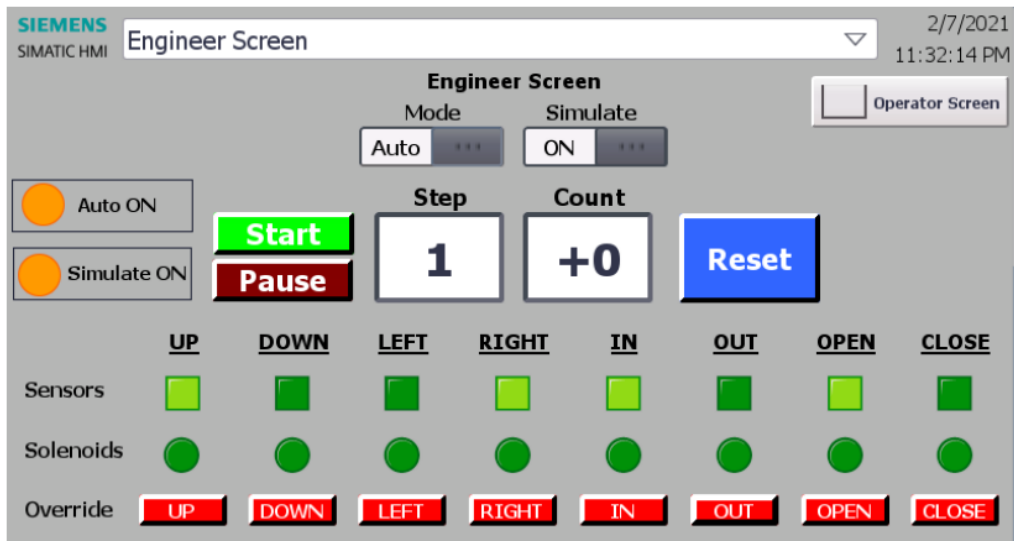


Figure 42 – Step 1

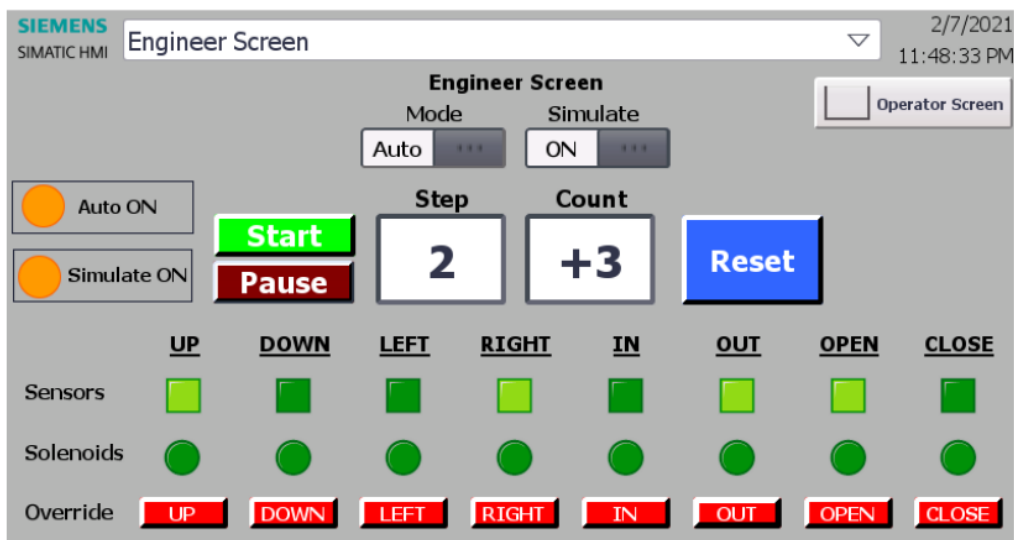


Figure 43 – Step 2

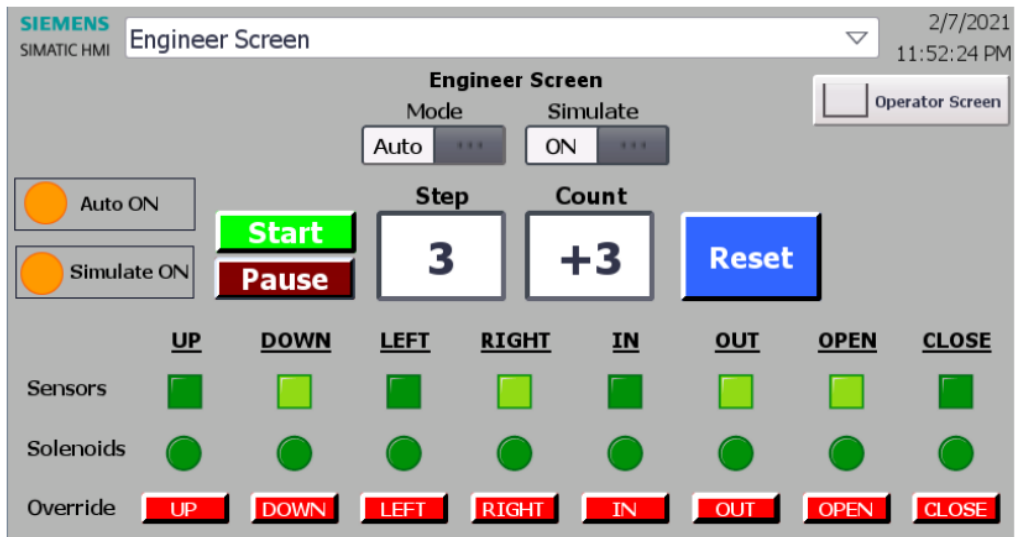


Figure 44 – Step 3

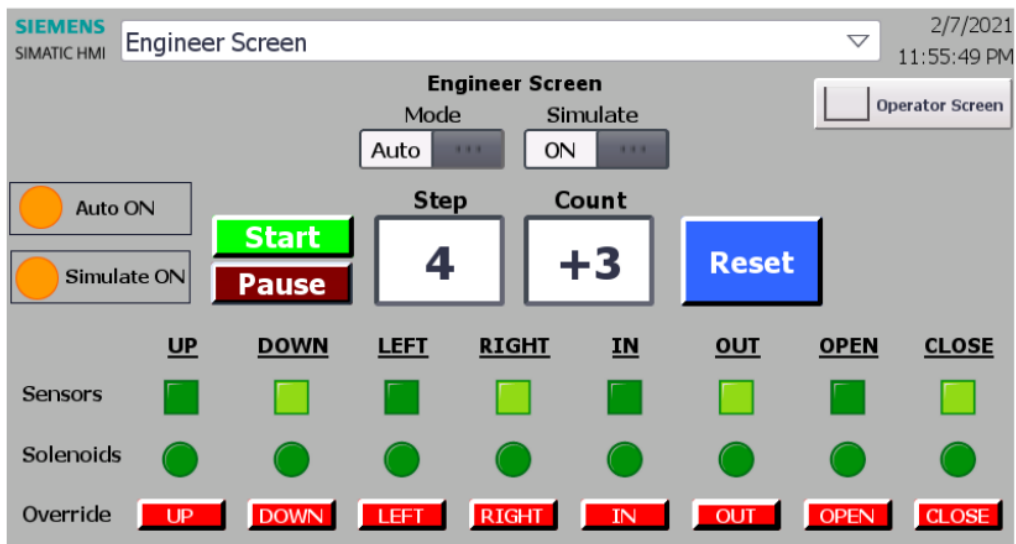


Figure 45 – Step 4

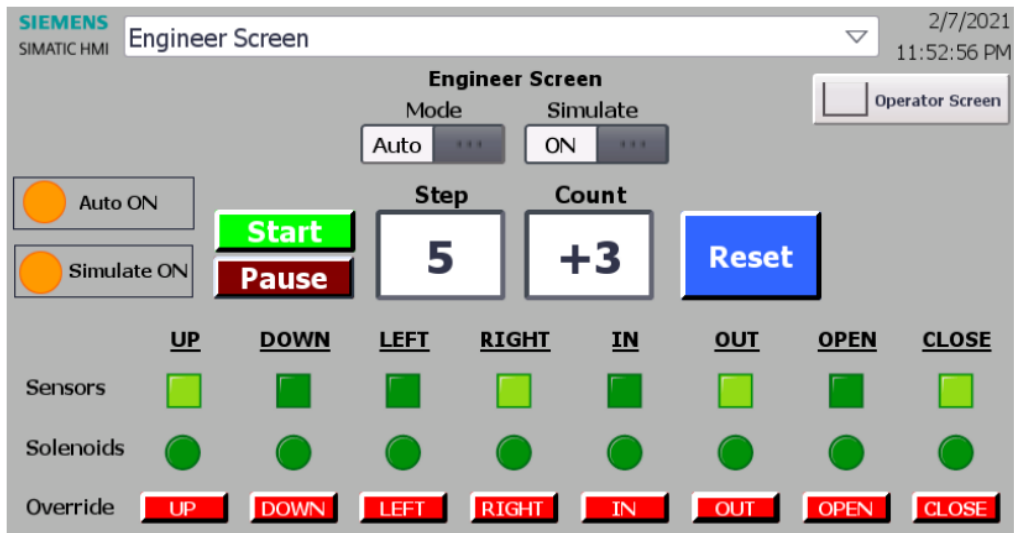


Figure 46 – Step 5

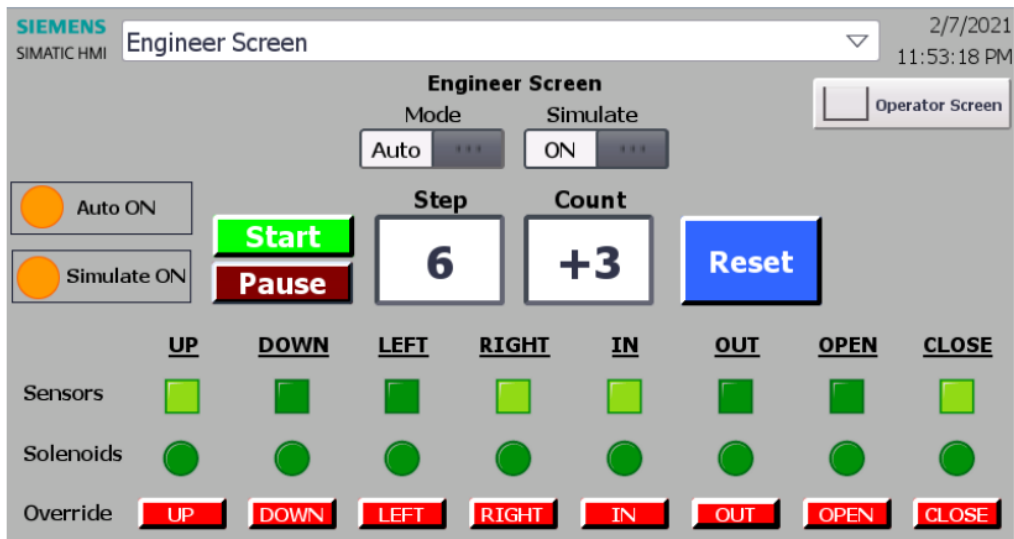


Figure 47 – Step 6

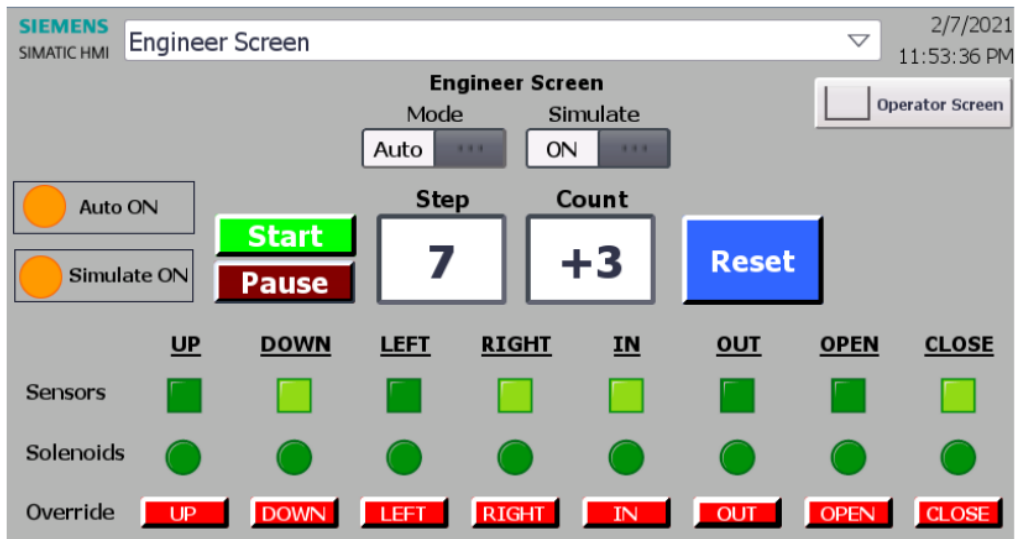


Figure 48 – Step 7

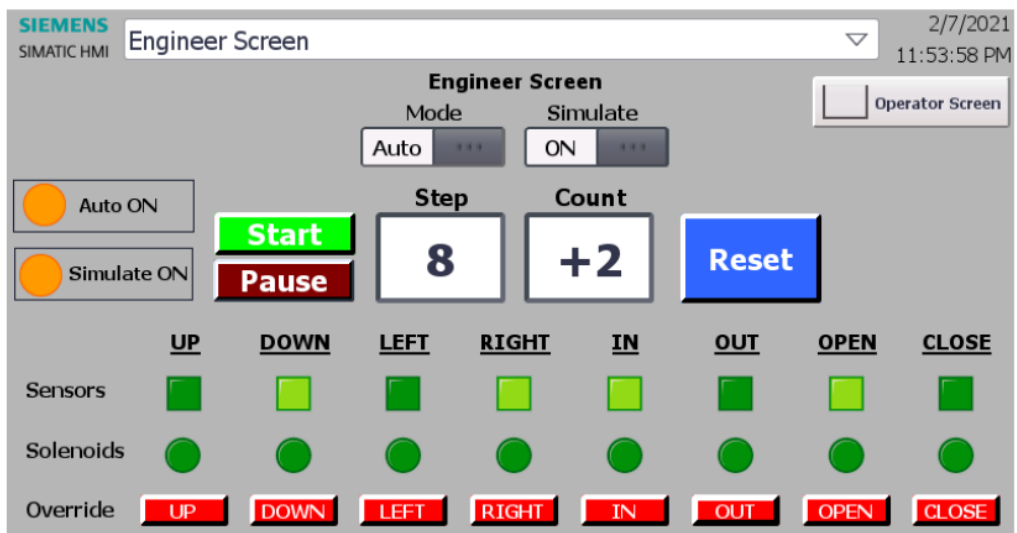


Figure 49 – Step 8

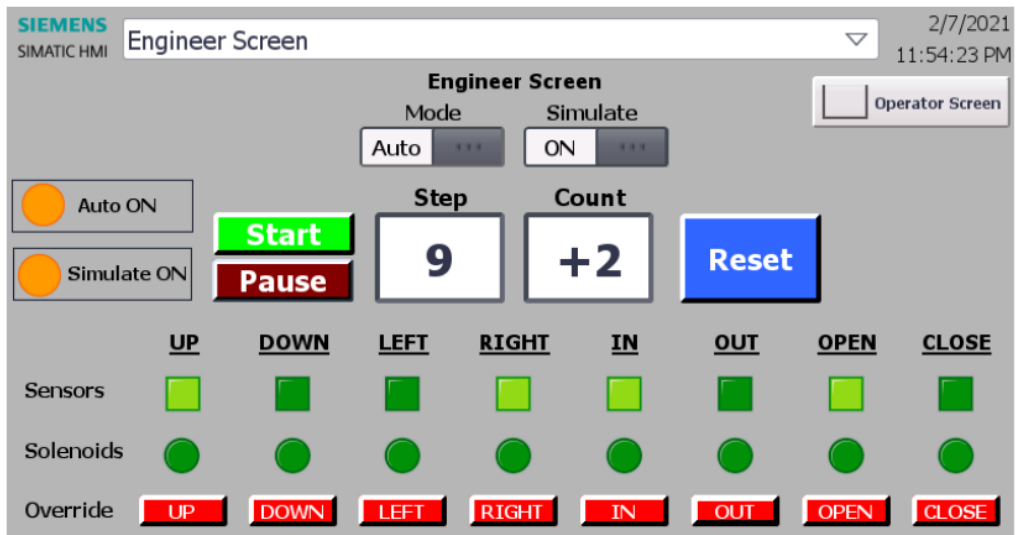


Figure 50 – Step 9

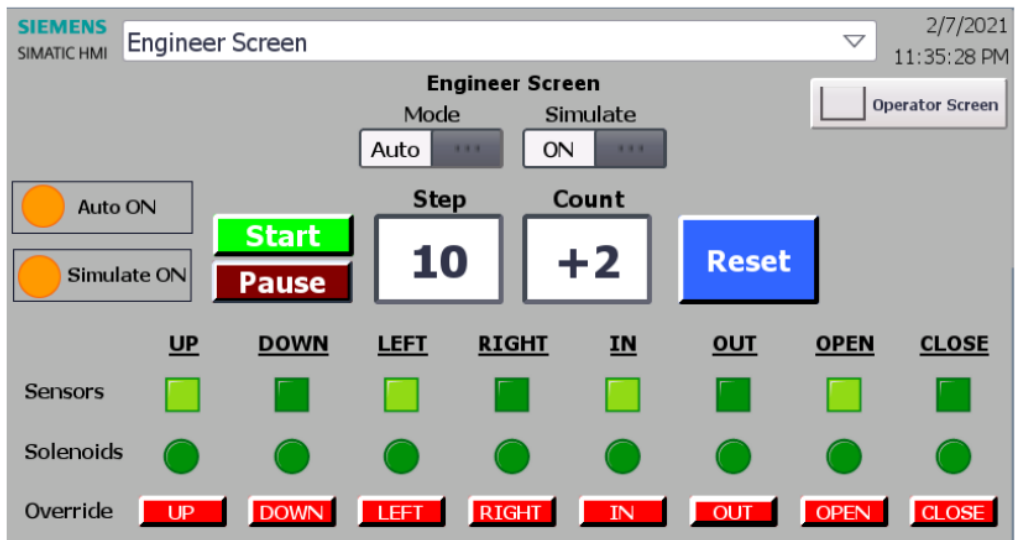


Figure 51 – Step 10

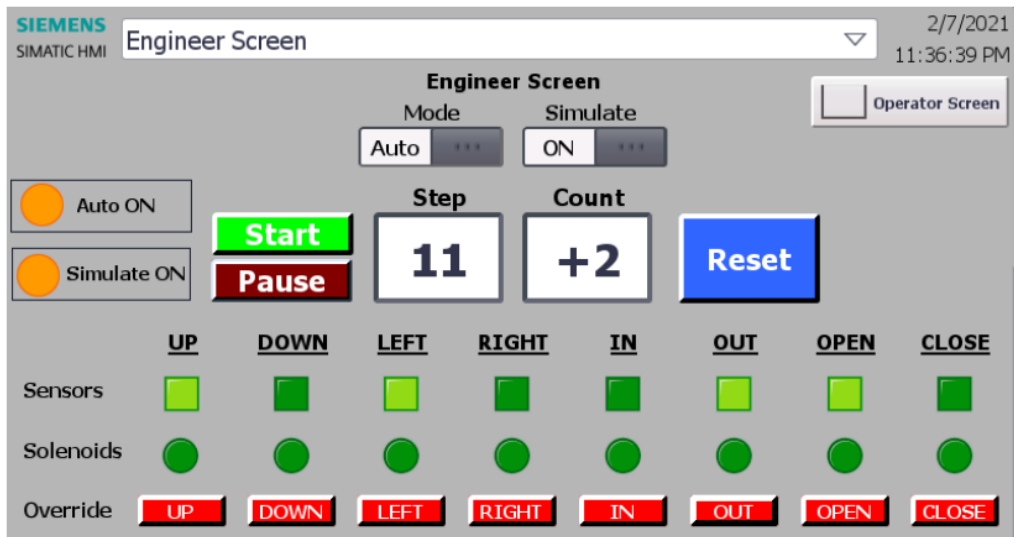


Figure 52 – Step 11

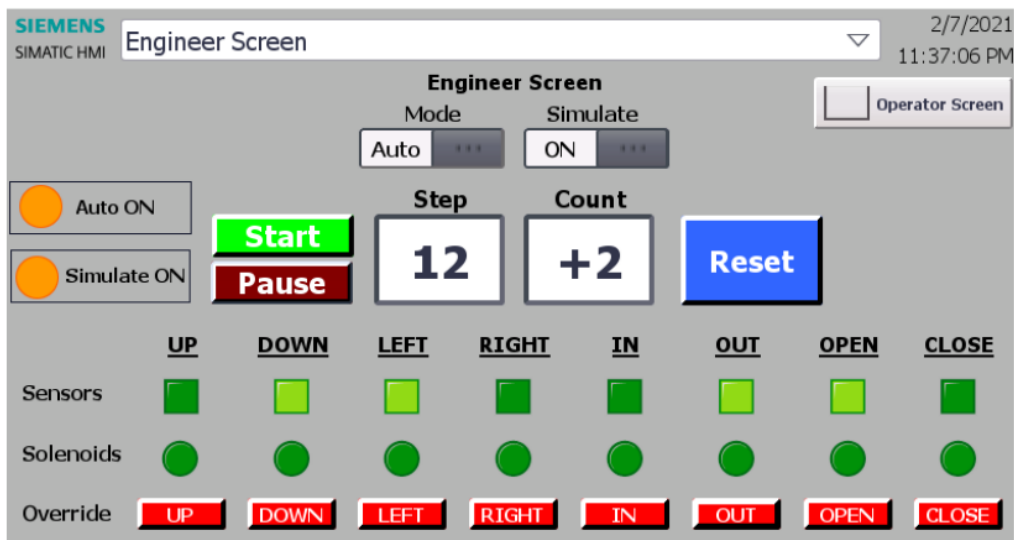


Figure 53 – Step 12

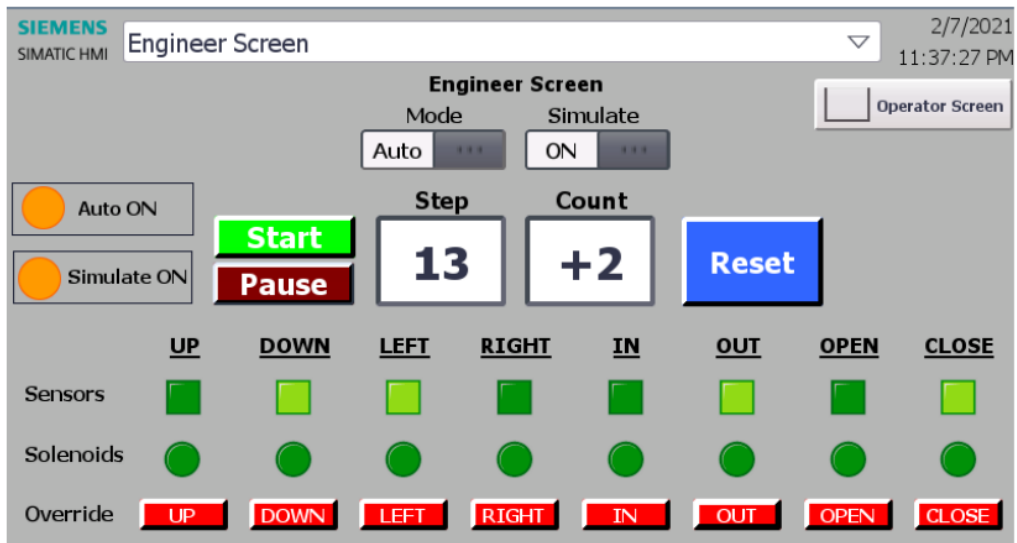


Figure 54 – Step 13

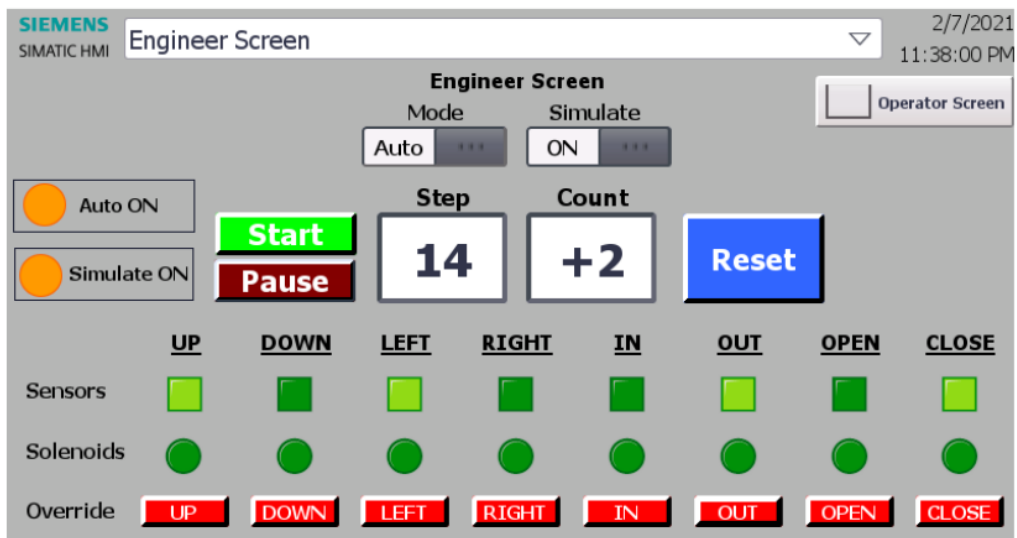


Figure 55 – Step 14

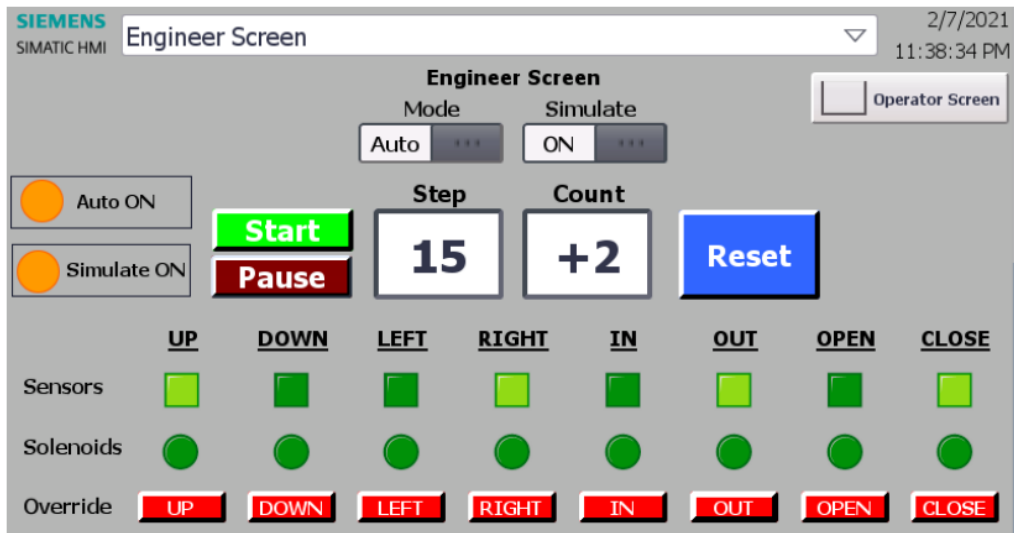


Figure 56 – Step 15

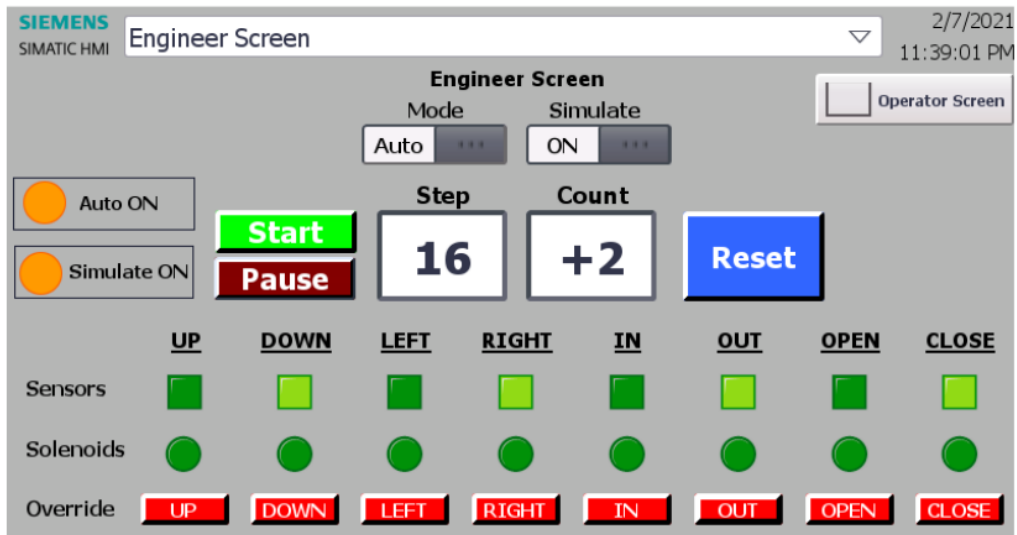


Figure 57 – Step 16

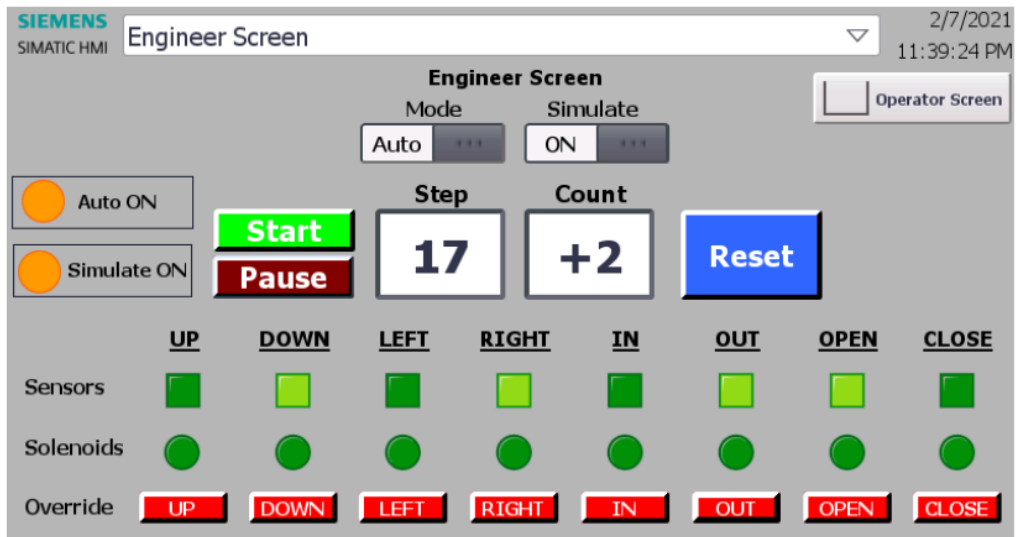


Figure 58 – Step 17

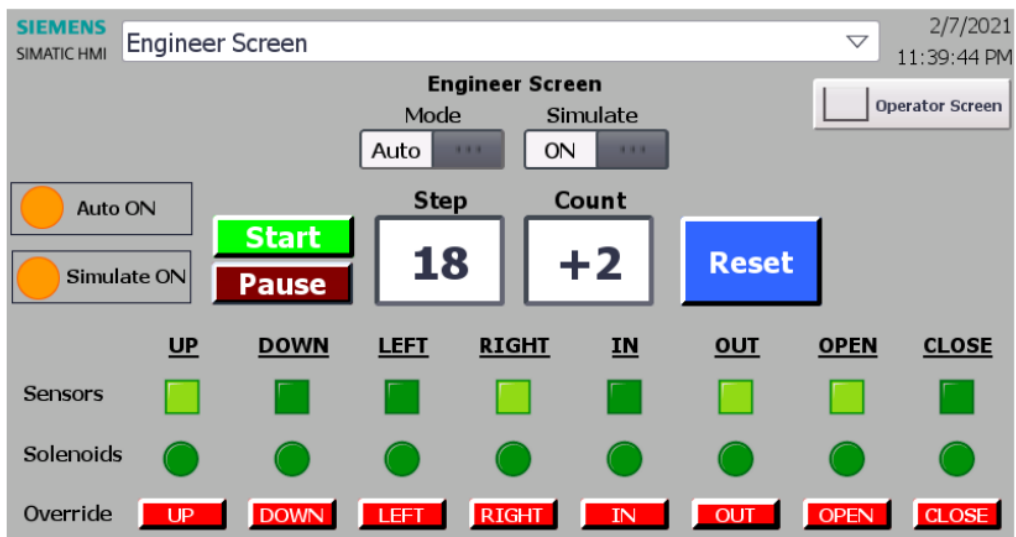


Figure 59 – Step 18

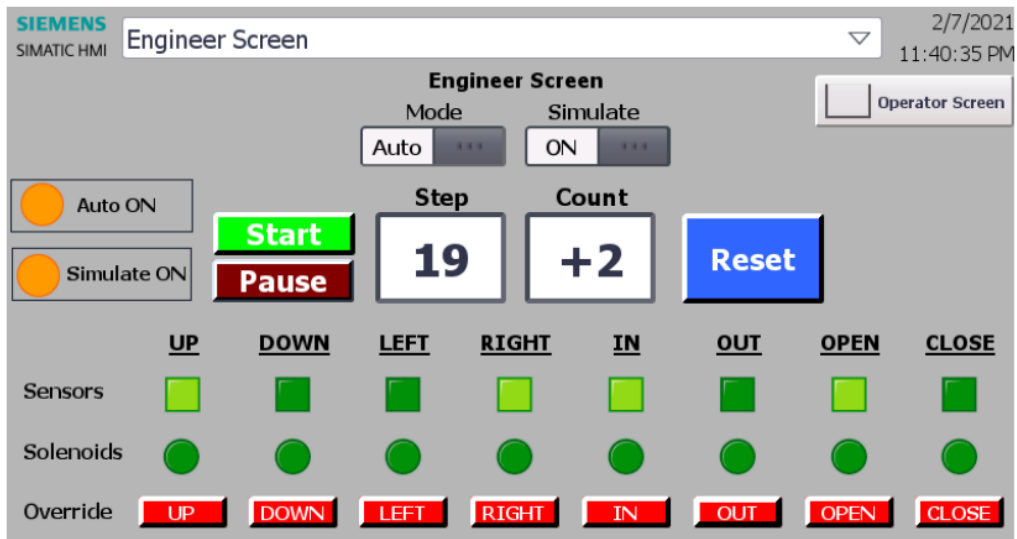


Figure 60 – Step 19

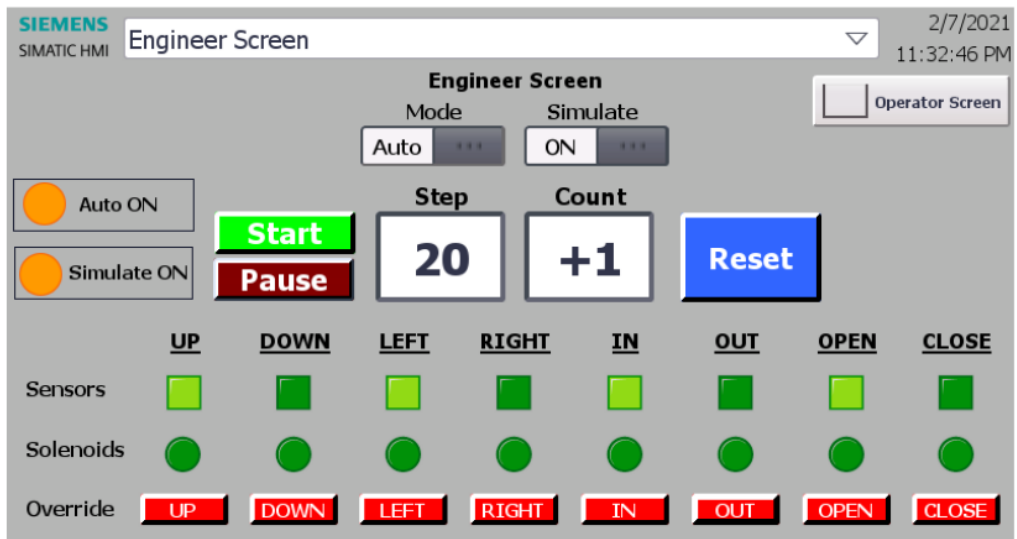


Figure 61 – Step 20

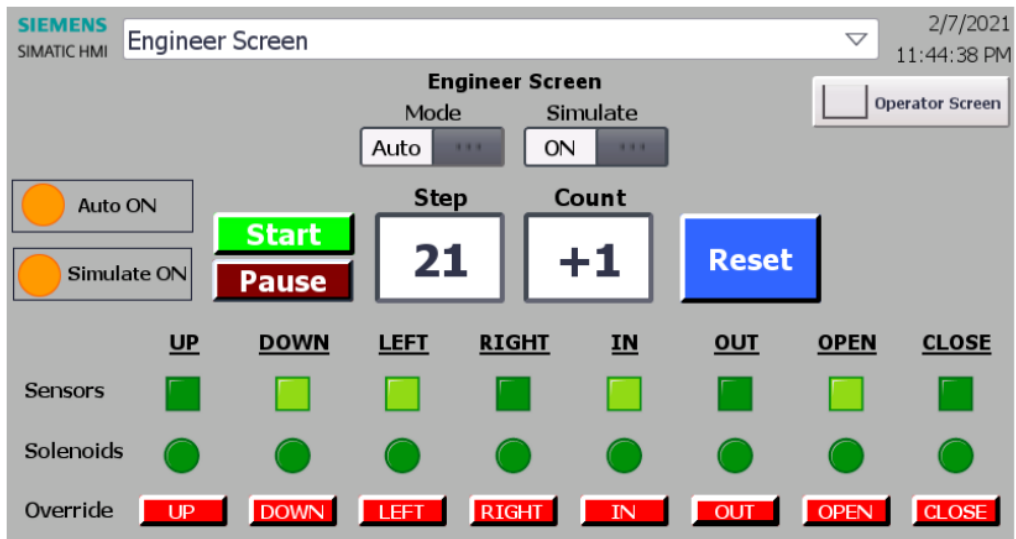


Figure 62 – Step 21

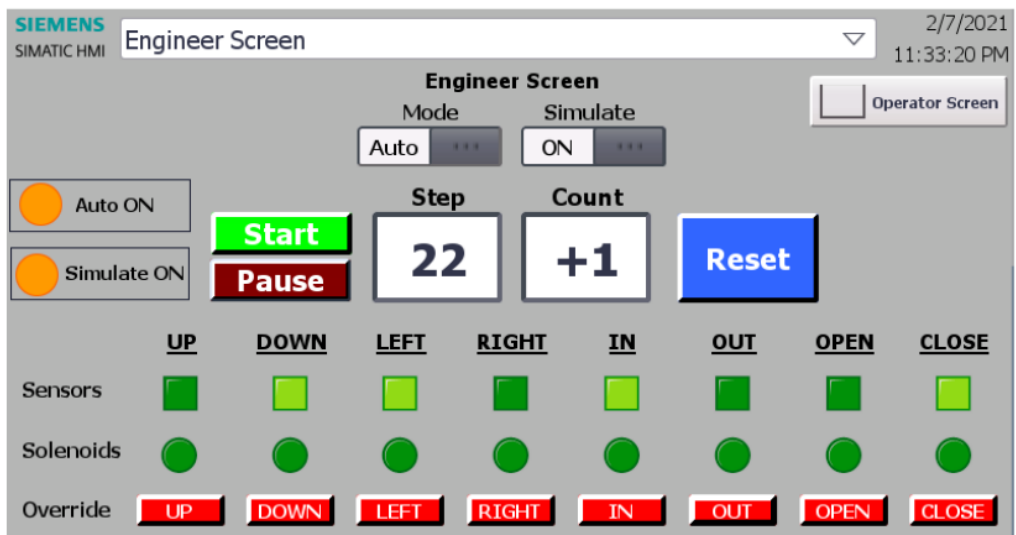


Figure 63 – Step 22

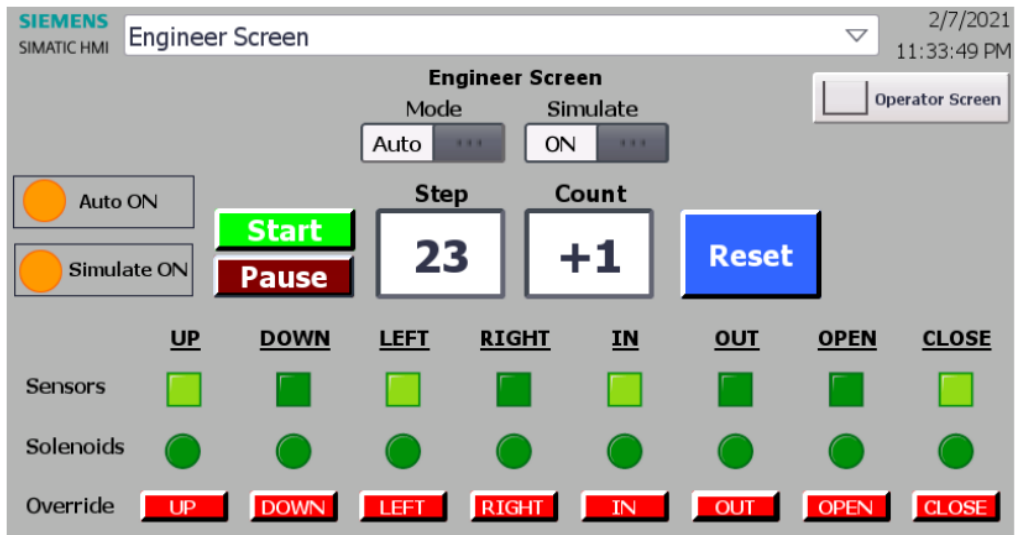


Figure 64 – Step 23

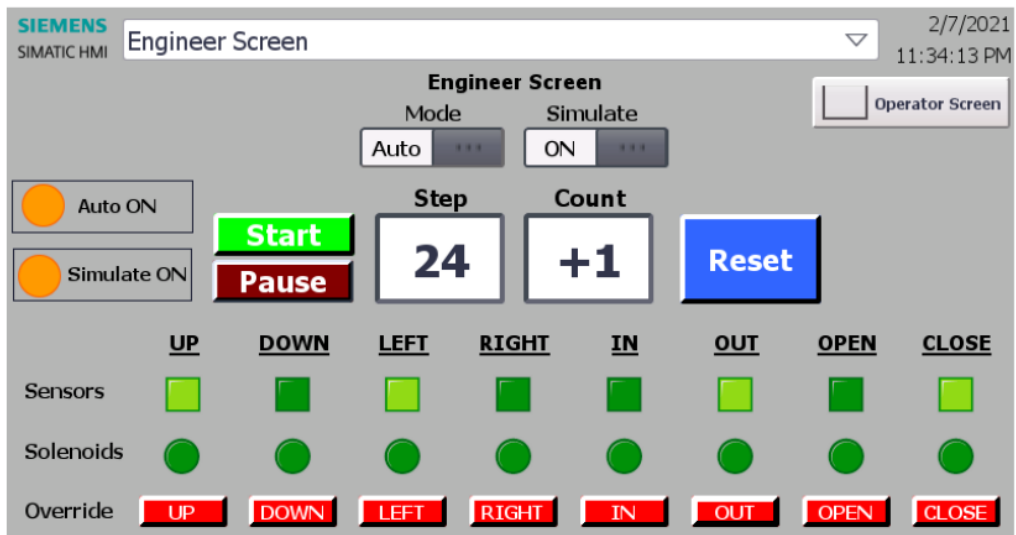


Figure 65 – Step 24

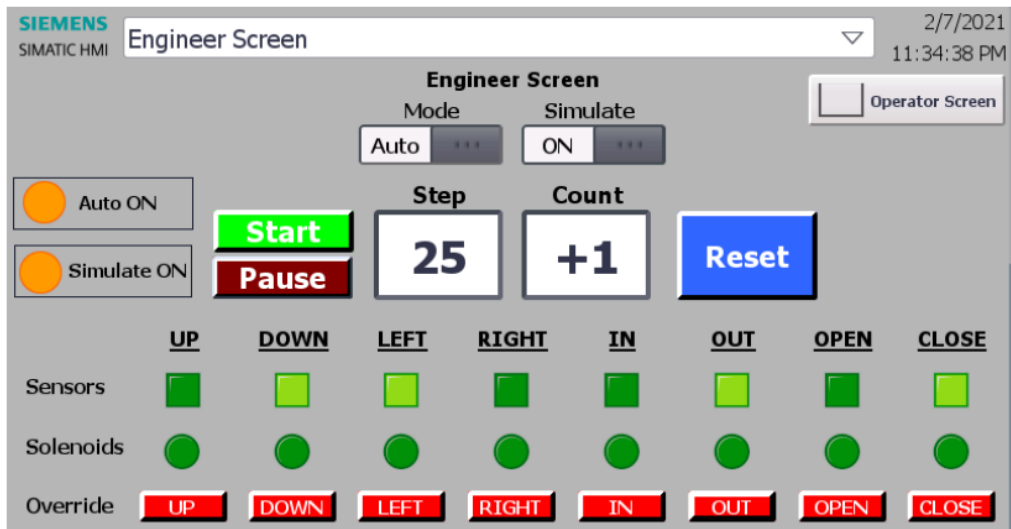


Figure 66 – Step 25

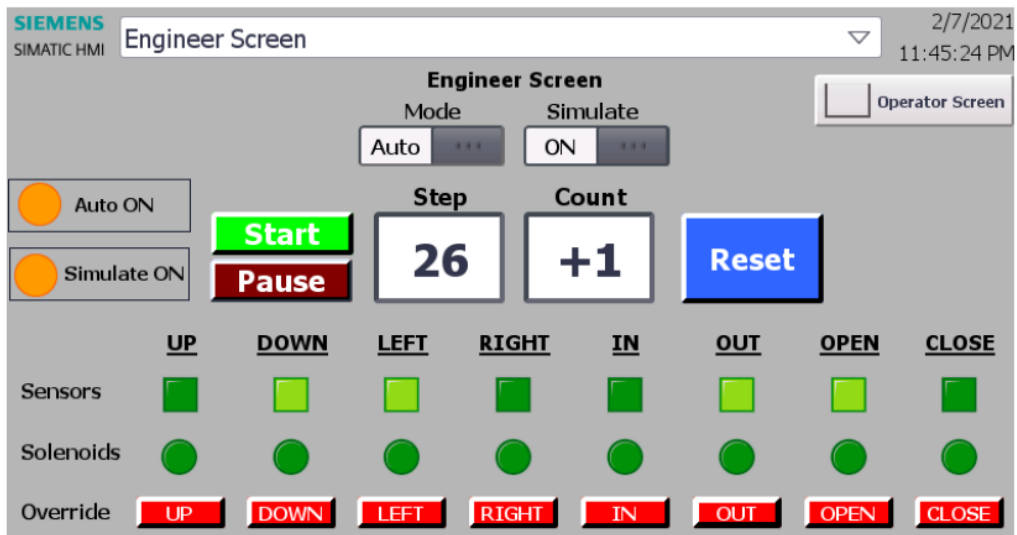


Figure 67 – Step 26

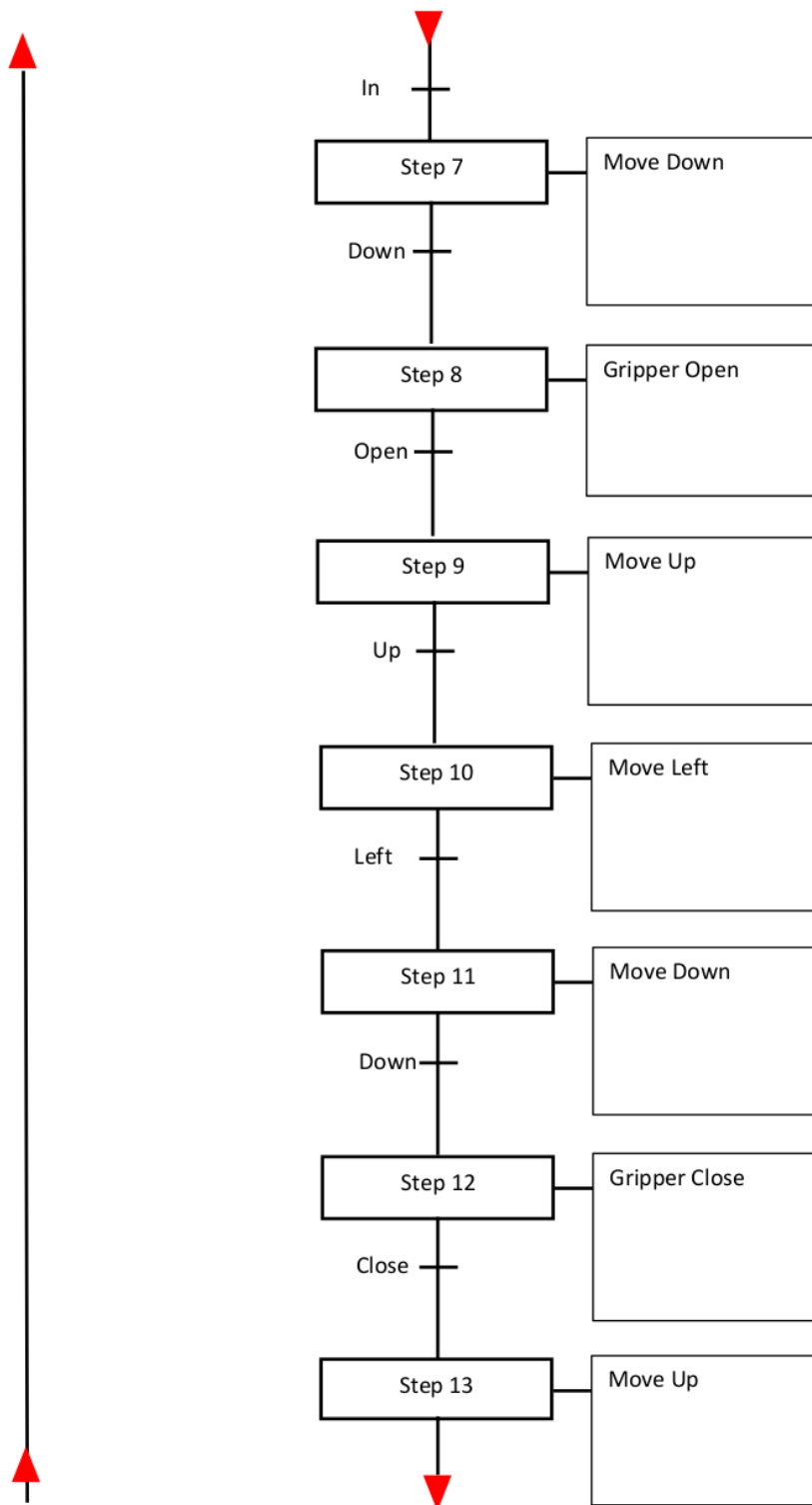


Figure 68 - SFC

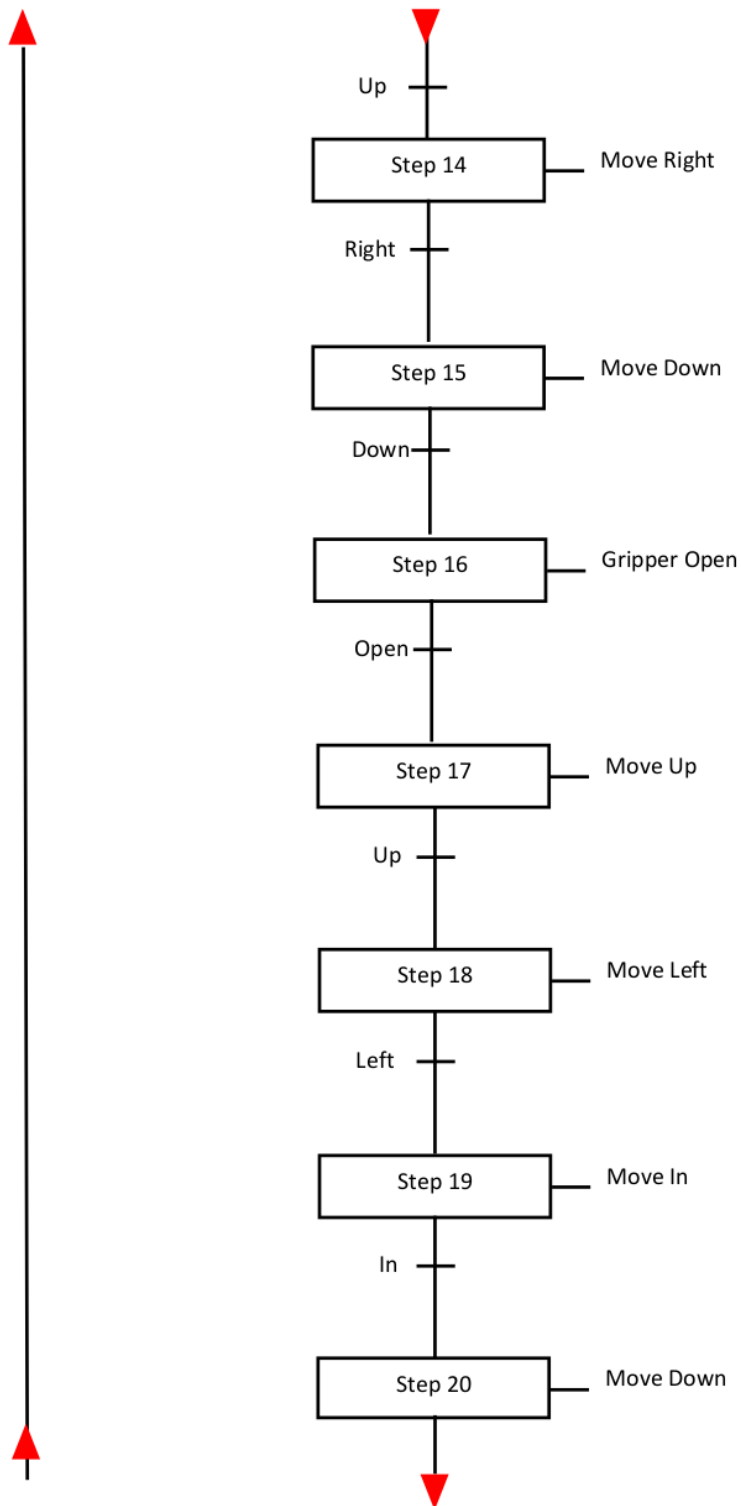


Figure 69 - SFC

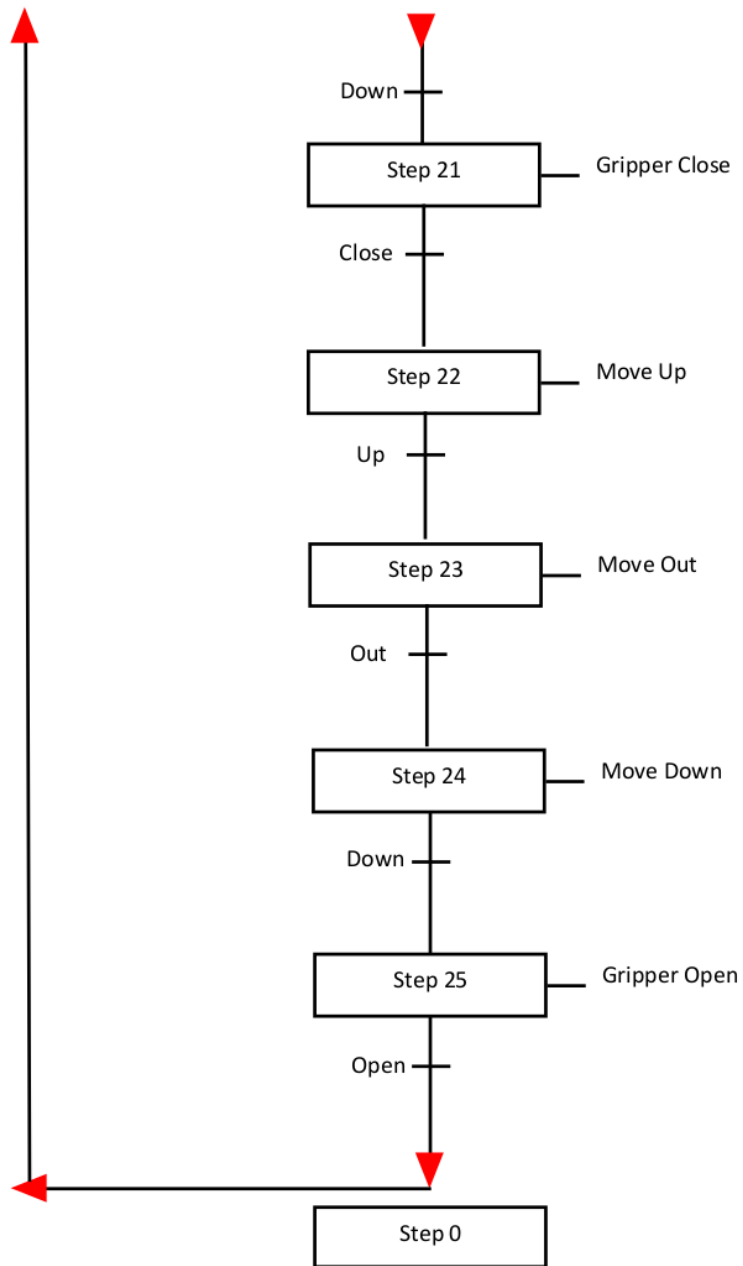
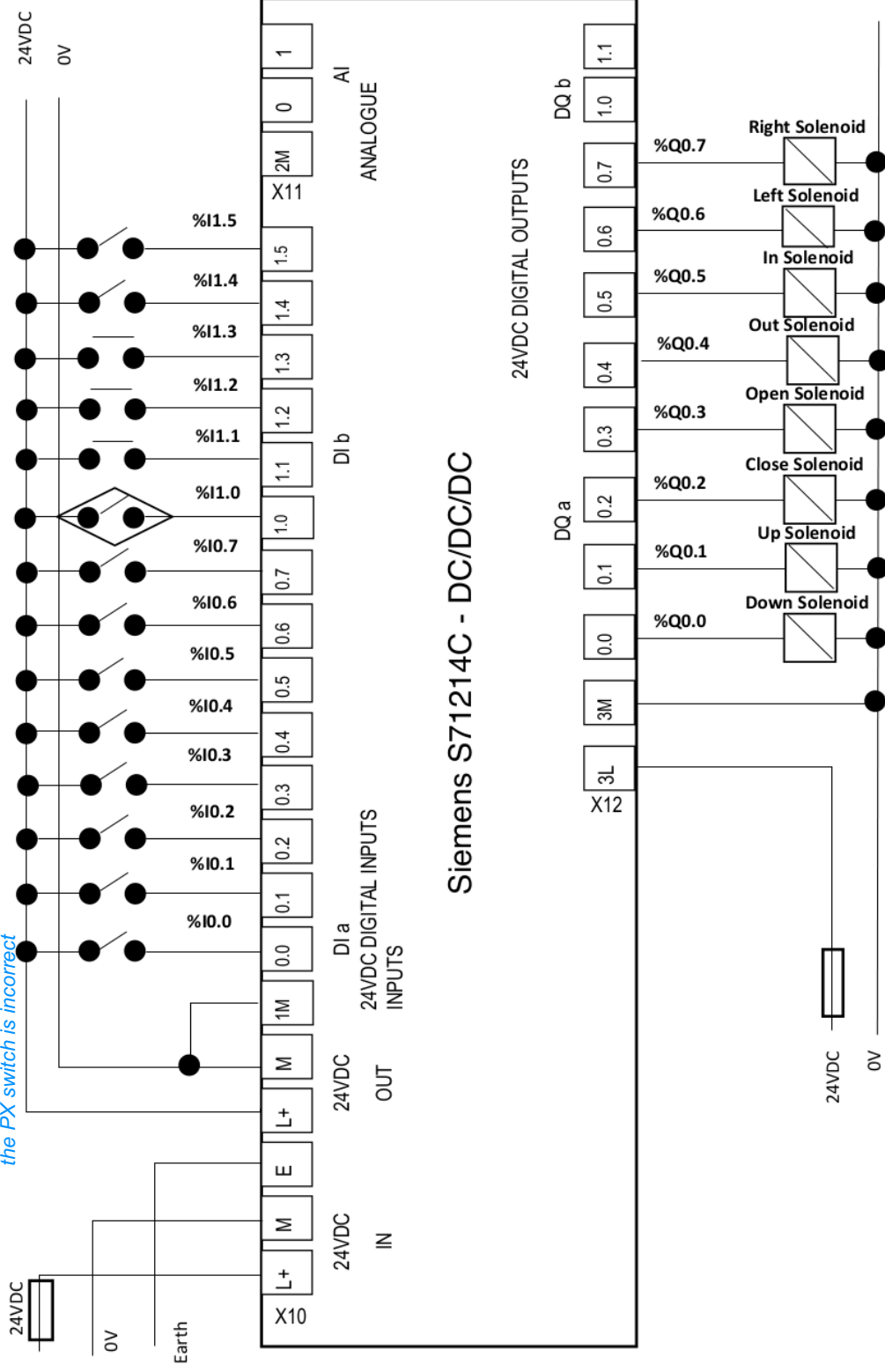


Figure 70 - SFC

Labels are missing for the component, the
black blobs are too large and the symbol for
the PX switch is incorrect



FINAL GRADE

GENERAL COMMENTS

Instructor

78/100

PAGE 1

PAGE 2

QM

Good Structure

Report is structured exactly as stated in the assignment brief.

PAGE 3

PAGE 4

PAGE 5

Text Comment. Not quite so, there are expansion modules available

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

Text Comment. The 1211 is a unitary PLC in this range, the rest are modular

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

Text Comment. Incorrect, the 300 is modular mounted on a metal rail, this is a common misconception as racks have electronic connectors

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

PAGE 6

Text Comment. This is the only truly rack mounted in the range

QM

Correctly cited

Citation method conforms to Harvard standard, well done.

PAGE 7

QM

Citation Needed

Cite Source:

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

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

PAGE 8

Text Comment. Incorrect, the Logo is expandable

Text Comment. Correct conclusion

PAGE 9

Text Comment. Excellent table

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

Text Comment. Excellent SFC

PAGE 17

PAGE 18

Text Comment. Excellent table

PAGE 19

PAGE 20

PAGE 21

PAGE 22

Text Comment. An excellent section

Text Comment. Full justification of text would improve the presentation of your paragraphs

PAGE 23

PAGE 24

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 25

Text Comment. A well designed and documented program

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

Text Comment. Labels are missing for the component, the black blobs are too large and the symbol for the PX switch is incorrect

GRADING FORM: NEW CSG-RUBRIC

AERON SMITHAM

78

1. ENGAGEMENT WITH LITERATURE



There is excellent evidence of research from a wide range of sources cited and listed correctly using the Harvard method. (16/20)

16

2. KNOWLEDGE AND UNDERSTANDING



Knowledge and understanding are excellent, clearly evidenced through your explanations of the hardware and software. However, there are a few misconceptions in your research section. (13/20)

13

3. COGNITIVE AND INTELLECTUAL



Excellent evidence of analysis through testing, discussion, conclusions and recommendations. (16/20)

16

4. PRACTICAL APPLICATION



Excellent program design through SFC chart, ladder and SCL coding with full internal documentation. Hardware drawing could be improved with labels for inputs and correct symbol for PX switch. (16/20)

16

5. TRANSFERABLE SKILLS



Excellent transferrable skills demonstrated through report quality and team working. Next time, set the justification of your paragraphs to FULL. (17/20)

17