

1302736

by Jason Booth

FILE	19735_JASON_BOOTH_1302736_178848_2003636376.PDF (1.95M)		
TIME SUBMITTED	08-FEB-2021 10:54AM (UTC+0000)	WORD COUNT	4555
SUBMISSION ID	144391166	CHARACTER COUNT	24352

Course: HNC Electrical and Electronic Engineering

Module: SGEE4004 Programmable Logic
Controllers

Assignment: 1. Pick and Place Robot

Name: Jason Booth

University number: 1302736

Date Submitted: 08/02/2021

Word Count: 3201

Contents

*Sections should be numbered
according to the brief*

Introduction	1
Research	1
S7 1200.....	1
S7 1500.....	2
S7 300.....	2
S7-400	3
Which Siemens controller is suitable for the pick and place robot design?	3
Hardware Investigation, Circuit Diagram, and I/O List	3
Inputs	3
Pushbuttons	4
Proximity Sensor (Reed switches)	4
Outputs	4
Program Design and Explanation	6
PLC Tags list.....	7
Manual Control	7
Automatic Control.....	12
Simulation Testing.....	13
Practical Testing and Team Working	15
Discussion, Conclusion and Recommendations.....	28
Bibliography	29
Appendix	30
PLC Tag List.....	30
Program Main Block.....	31
Automatic Sequence	37
Engineer's Screen.....	41
Operators screen	41
S7-1200 Data Sheet.....	42
S7-1500 Data Sheet.....	43

Introduction

The purpose of this assignment is to give an insight into PLC design, programming and testing by building a pick and place robot using programmable software to write directly to the PLC controller. The hardware and software used is by Siemens, so TIA portal is used to write function calls to the S7-1214C controller. In the report there also contains research into various types of controllers made by Siemens and to discuss whether the controller chosen to build the pick and place robot is a suitable choice.

Research

The following research has been obtained mostly through the Siemens website and Viktori-automation website.

S7 1200



The SIMATIC S7 1200 series is a modular PLC and is Siemens basic controller, which is recommended for projects scaling from small to medium. It has 5 different models with each one having an extra benefit to support the demands of a project. It can have multiple expansions of digital input and output modules. Hence the term 'modular' PLC. The controller is quite compact so is space saving and is mountable on DIN rail, which can be situated in a control panel.

It has integrated digital inputs and outputs ranging from 6 DI and 4 DO, to 14 DI and 10 DO. It can also be purchased with additional AI and AO. The 1215C controller for example, has 2 AI and 2 AO that can be used to send feedback to the controller and continue the process. It supports basic programming languages such as LAD, FBD, SCL.

There are various communication standards onboard by means of integrated functions such as PROFINET, Modbus, etc. This means you can communicate to a whole network of PLCs via an ethernet cable, and build a network topology with multiple PLCs on the network and have it connected to a server, where it can be monitored. Although this is not needed as all the controllers can be used in stand-alone mode.

S7 1500



Incorrectly cited, this is not the Harvard Method

The following information was obtained from the S7-1500 data sheet through the manufacture's website and a snippet of the data sheet is attached in the appendix of this report. The SIMATIC S7-1500 series is Siemens advanced controller and the successor to the S7-300. It is recommended for small to large scale project. There are 7 different models, giving either better or additional features such as analogue inputs and outputs, as well as relay outputs for switching with higher voltage and current. The integrated memory for program ranges from 150 KB to 6MB, and for data ranges 1 MB to 20 MB. Up to 32 GB memory card can be installed for load memory/mass storage. It has 2 PROFINET ports as standard on all S7-1500 CPUs, although more can be selected when ordering. Onboard it comes with 16 DI & DO up to 32 DI & DO depending on which model is selected.

The S7-1500 has a standard built in display and keypad that allows basic configuration and diagnostics without the need for an external screen. It can also come in three versions, standard version for typical applications, compact version for when space is a key factor, and fail-safe for safety critical applications.

Citation

Like the S7-1200, this PLC is built around a modular expandable design. I/O modules, communication modules can be expanded to the PLC via the backplane bus. It can be mounted on a back plate supplied by Siemens which can either be situated in a control panel or in a control cabinet. The S7-1500 can be expanded to a maximum of 32 modules.

Citation

Pricing is around £1000 for a starter kit.

S7 300

The SIMATIC S7-300 is much like the S7-1500 but is older and therefore lacking some performances and features that the S7-1500 controller has. But is still proven to be an excellent controller that can handle small to large projects. The controller is modular in design and can support expansion up to 32 modules. The S7-300 come in three categories. Standard, compact, and fail-safe. It does not have integrated inputs and outputs like the S7-1200.

S7-400

Incorrectly cited, this is not the Harvard Method

According to PT. VIKTORI PROFINDO AUTOMATION, the S7-400 is the most powerful PLC in the family of SIMATIC controllers. It enables successful automation solutions with TIA which is the software used to write code to the PLC. It is recommended for large scale projects where there is a big demand of a lot of data being processed at one time. Average price for this controller is around £3,250.

Which Siemens controller is suitable for the pick and place robot design?

Analyzing the program requirements and the hardware used in the system, there are in total 14 digital inputs and 8 digital outputs. So, this is the bare minimum IO required.

Citation
Needed

When looking at what is involved in the process and what the program is doing, it is not a complex system. The inputs are digital switches with just two states, on or off, 1 or 0, and the outputs are 24V solenoids memory valves that allow a valve to open or close, which will allow compressed air to extend the piston for the robot arm to be in position.

PLC Controller	S7-1200	S7-300	S7-1500
Onboard DI	14	24	32
Onboard DO	10	16	32
Work Memory	150KB	N/A	6MB
Load memory	4MB	4MB (Card)	20MB
Price	£454	£854	£550 (No IO)

Citation
Needed

Spelling

All of the PLC controllers listed by Siemens is capable of delivering the required functions for the pick and place robot design. However the pricing on some of these controllers are astronomical in terms of what is actually wanted. Choosing, say a S7-400, would be far overkill for the pick and place robot, as this is an advanced controller design for very complicated and large systems. Also it is quite large in size and does not fit onto regular DIN rail, and comes with a hefty price tag.

The S7-1200 would be a far suitable choice. Onboard the controller, it comes with DI and DO, so eliminates the need in purchasing separate DI and DO modules like you would with other controllers like the S7-1500. It is very small and compact in design and can fit onto regular DIN rail which makes installation an ease. It uses The cost for this controller is quoted at £434.40 on RS Components website.

Good conclusions

Citation
Needed

Hardware Investigation, Circuit Diagram, and I/O List

Inputs

Altogether there are 14 digital inputs being used to initiate the program, to stop the program and to give indication of the positioning of the robots. These inputs are also used as interlocks, so that the robot can only move when certain conditions (inputs) are met. The difference between digital and analogue is that digital has only two states, on or off, 1 or 0. Whereas analogue inputs have a range of values it can be. For example, a thermocouple measuring temperature will give an input signal of between 4-20mA. This is known as an analogue input.

*There are no analogue
inputs in this system.*

Pushbuttons

A pushbutton is basically a switch that is operated by a human. They have two positions, either normally open (NO) or normally closed (NC). They can also come with a latch, so when pressed it will remain either closed or open until pressed again.

Proximity Sensor (Reed switches)

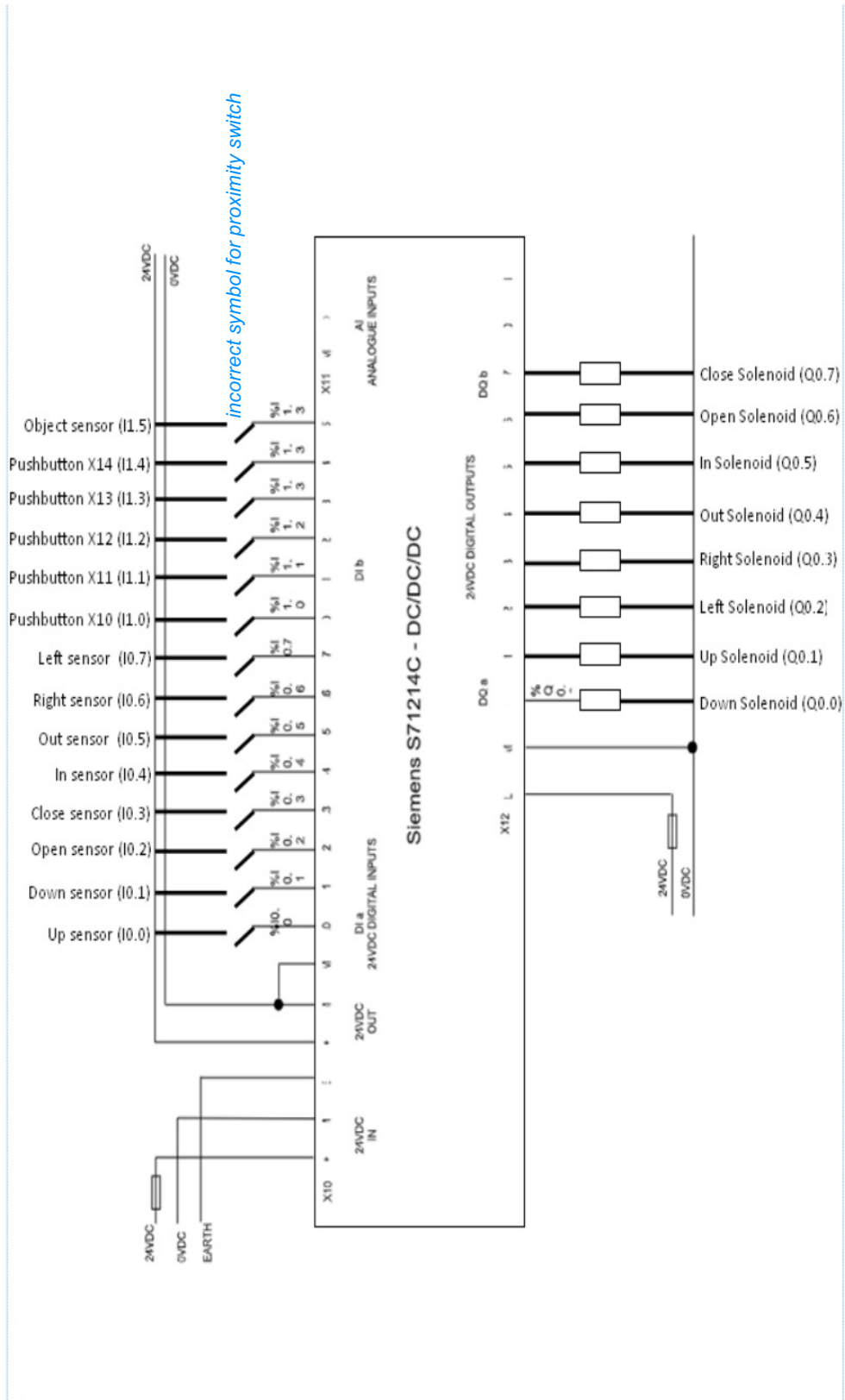
A proximity sensor detects an object when it is in close contact to the device. Their primary purpose is to give indication of position. Most proxies have 2 LED's, one to indicate if there is power to the proxy and the other indicates detection. They are a solid state device which means there are no moving parts, so therefore is quite reliable and robust. They can come in different detection ranges i.e. 2mm, 10mm, 20mm. Proxies are used in closed loop systems to provide feedback to the controller and operator. For example, a valve opens to allow compressed air to extend the piston rod until it reaches the proxy sensor. When the piston rod contacts the proxy sensor it can either close or open the circuit to tell the PLC to close the valve.

Outputs

For the pick and place robot, there are 4 pneumatic DCVs (directional control valves) that extend and retract the piston rods to form 3-axis control for the robot arm (up/down, in/out, left/right) and to open and close the robot gripper. Each DCV has 2 solenoids, one to retract and one to extend. These solenoids are wired to the PLC output terminals so that when conditions are met to energize the output in the network rungs, 24 volts is sent out of the PLC to the solenoid, this will open the valve and allow the piston rod to either extend or retract.

The following page contains an electrical drawing of the PLC with all inputs and outputs connected to the appropriate terminals.

Where is your IO list?

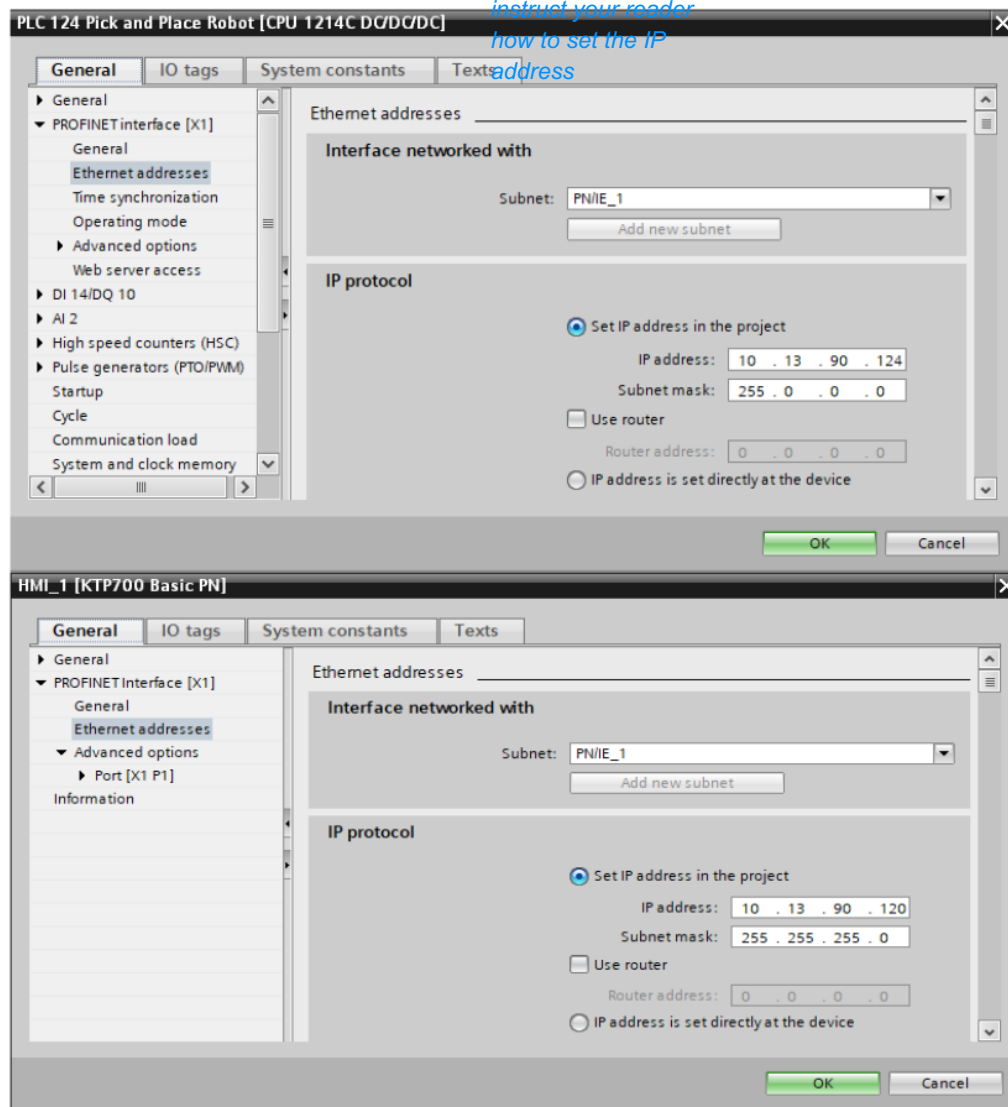


Program Design and Explanation

This section of the report contains the program design for the pick and place robot, which was created using TIA portal. It also contains an HMI with two screens selectable, 'Engineer's screen' and 'Operator's screen'. Memory elements was used as a substitute for the actual inputs so that a simulation block can be used to test the program without having the robot hardware. The program also has two running modes, manual and automatic.

For the PLC controller and HMI to communicate, the IP addresses of both need to be in the same network range. The IP address was set by clicking the PLC and HMI properties on TIA portal and changing the IP address, as in the images below.

It is unnecessary to instruct your reader how to set the IP address



PLC Tags list

A PLC tag list was created based on the I/O list for the robot. The images below show the tag list for the inputs and outputs. A bigger tag list can be viewed in the appendix section of this report. In the bigger version there are additional tags used for function calls for the automatic sequence and manual control, and tags for the HMI switches and counters.

Input Tag List							
	Name	Path	Data Type	Logical Address	Hmi Visible	Hmi Accessible	Hmi Writeable
Reed switches	UP sensor	Simulation Tags	Bool	%M0.0	True	True	True
	DOWN sensor	Simulation Tags	Bool	%M0.1	True	True	True
	OPEN sensor	Simulation Tags	Bool	%M0.2	True	True	True
	CLOSED sensor	Simulation Tags	Bool	%M0.3	True	True	True
	IN sensor	Simulation Tags	Bool	%M0.4	True	True	True
	OUT sensor	Simulation Tags	Bool	%M0.5	True	True	True
	RIGHT sensor	Simulation Tags	Bool	%M0.6	True	True	True
Push Buttons	LEFT sensor	Simulation Tags	Bool	%M0.7	True	True	True
	X10 Pushbutton	Simulation Tags	Bool	%M1.0	True	True	True
	X11 Pushbutton	Simulation Tags	Bool	%M1.1	True	True	True
Rotable Switches	X12 Pushbutton	Simulation Tags	Bool	%M1.2	True	True	True
	X13 toggle switch	Simulation Tags	Bool	%M1.3	True	True	True
	X14 toggle switch	Simulation Tags	Bool	%M1.4	True	True	True
Proxy Sensor	OBJECT sensor	Simulation Tags	Bool	%M1.5	True	True	True

Outputs Tag List							
	Name	Path	Data Type	Logical Address	Hmi Visible	Hmi Accessible	Hmi Writeable
Solenoid Valves	UP solenoid	Assignment tag table	Bool	%Q0.1	True	True	True
	DOWN solenoid	Assignment tag table	Bool	%Q0.0	True	True	True
	OPEN solenoid	Assignment tag table	Bool	%Q0.6	True	True	True
	CLOSE solenoid	Assignment tag table	Bool	%Q0.7	True	True	True
	IN solenoid	Assignment tag table	Bool	%Q0.5	True	True	True
	OUT solenoid	Assignment tag table	Bool	%Q0.4	True	True	True
	RIGHT solenoid	Assignment tag table	Bool	%Q0.3	True	True	True
	LEFT solenoid	Assignment tag table	Bool	%Q0.2	True	True	True

Write in the third person and concentrate on explaining the system.

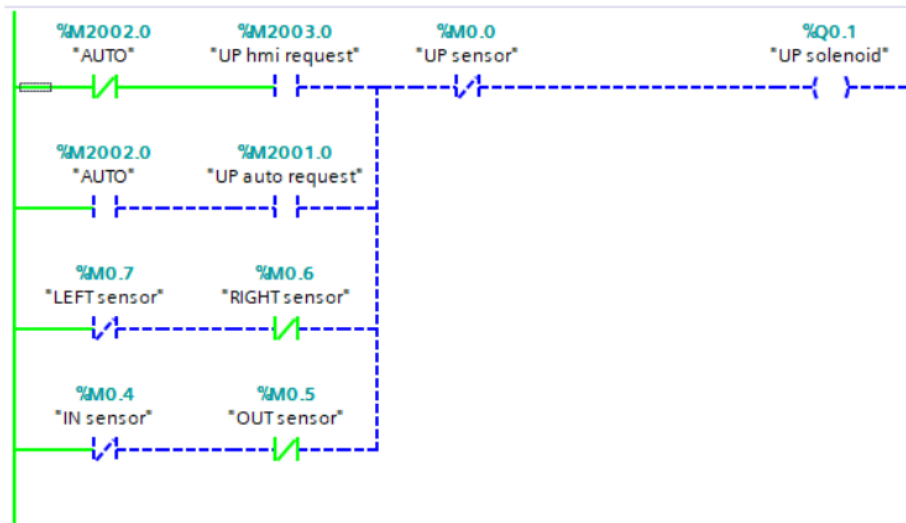
The purpose of the Tag list is to have all the data of the inputs and outputs together in one window. It is also very beneficial when it comes to programming the robot. For example, when labeling a contact in the network rungs, it will automatically pick up the tag name when you enter the first few letters of the name, and the corresponding address to the tag list will be generated. This is also true when entering the address of the contact rather than the name.

Manual Control *Should be "the program contains" rather than*

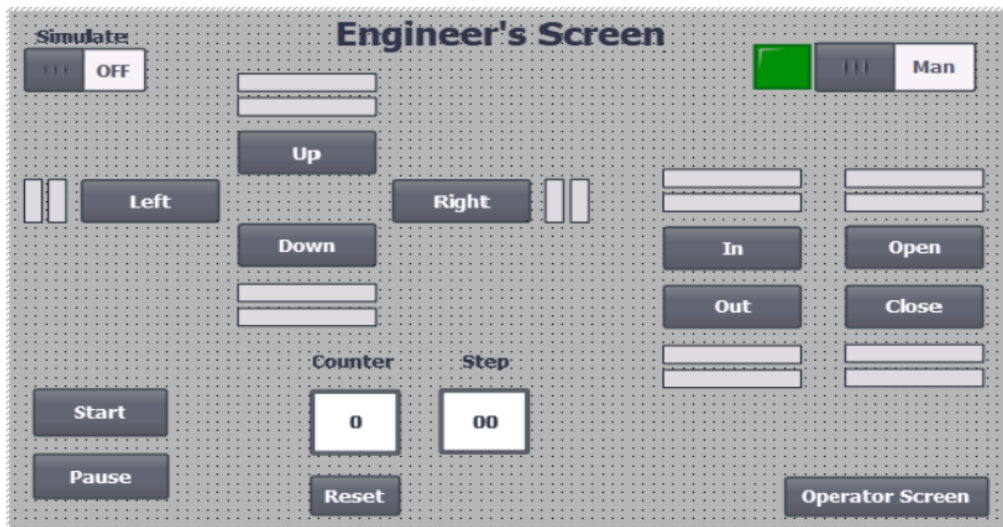
"contains a manual override state in the machine"
The program must contain a manual override state in the machine robot is stuck between positions or stuck somewhere in the sequence and needs positioning by manual means. It must have full axis control for the operator as well as OPEN and CLOSE gripper jaws. The AUTO button on the HMI screen switches opens/closes M2002.0 contact to either allow requests from the HMI screen (manual control) or requests from the auto sequence (automatic control). These statements are true for each network in the program that allows control of a solenoid.

Network 4: Move up

- UP solenoid can be controlled manually or automatically via auto switch and request from HMI screen. The robot arm will move up and stop when it reaches the UP sensor. If the robot is stuck inbetween left and right, or in and out, the robot will move up. This is a safety feature for the robot.



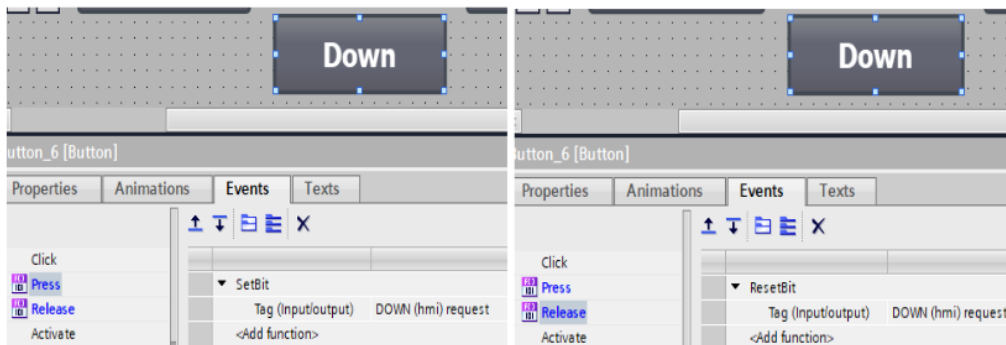
When memory contact M2002.0 (AUTO) is closed and the UP button is pressed on the HMI screen to make M2003.0 (UP HMI request) close, output Q0.1 (UP solenoid) will energize and drive the robot arm up until the UP sensor detects the robot arm and opens contact M0.0 (UP sensor). This is the case throughout all the network rungs that is controlling the solenoids.



all figures should be numbered or simply

This Engineer's screen on the HMI is where only authorized people may use. Here is where the operator may move/position the robot if it were to occur a fault and lose its positioning. The operator has full control of the machine/solenoids. These buttons are tagged to the memory bits

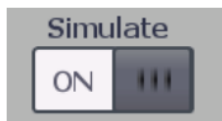
that are named as HMI requests, in the main program (OB1). So, when pressed they will either close or open the contact.



The Down button is tagged to activate DOWN (hmi) request – bit M2003.1 when pressed and to reset on release. As can be seen in the above images.

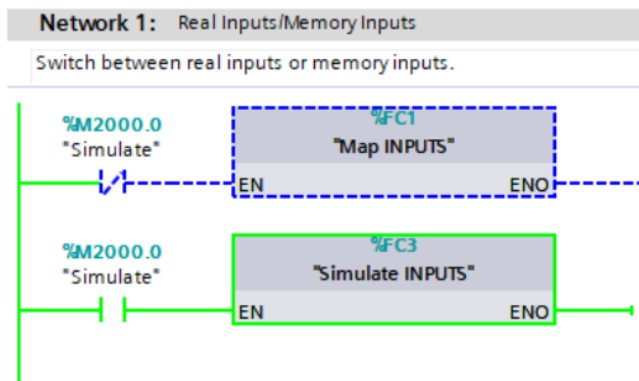


Next to each button are two lights that give indication. The first light closest to the button is tagged to the solenoid and the second light furthest from the button is tagged to the sensor. When the sensor has been made the second light will turn green and when the solenoid is energized the first light will turn green.



The simulation button is tagged to the Simulate contact – M2000.0, and when pressed it switches between MAP INPUTS and Simulated INPUTS. This allows to test the program without having the hardware available and wired to the PLC. When FC1 is chosen, the memory elements for the inputs are swapped to real inputs.

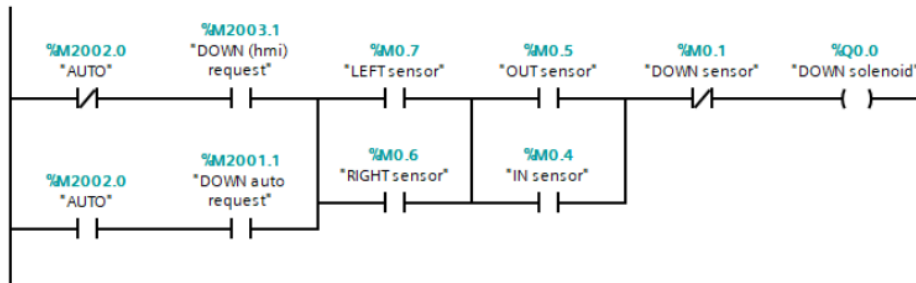
Inside FC3 are timers that simulate the sensors. For example, when the down solenoid is energized a normally open contact (Q0.0) closes and resets the UP sensor and activates the DOWN sensor after 2 seconds.



These next screenshot are from the main (OB1) section in the program, showing each solenoid in its network rung with all the sensors and buttons required for the output to be energized. Move up network has not been added here as it has previously been added and mentioned in the report.

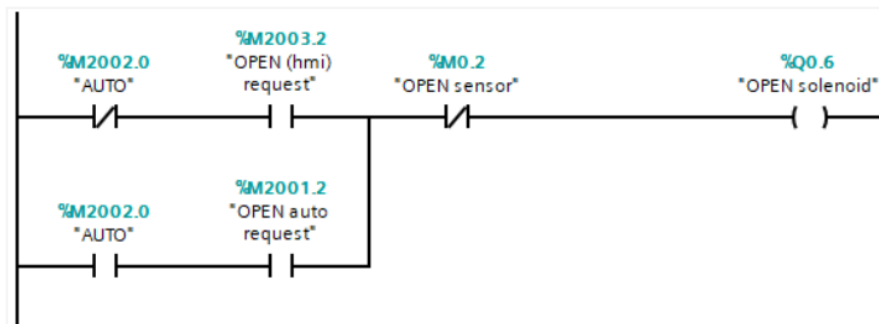
▼ Network 5: Move down

- ▼ DOWN solenoid can be controlled manually or automatically via auto switch and request from HMI screen. The robot arm will move down and stop when it reaches the DOWN sensor. It can only move down when it is above a station which are the LEFT, OUT, RIGHT, IN sensors. These are also safety interlocks.



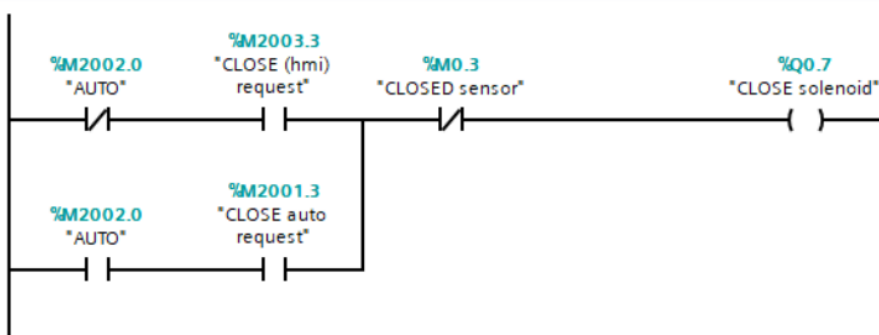
▼ Network 6: Open gripper

- ▼ Open gripper can be controlled manually or automatically via auto switch and request from HMI screen. The gripper will open until it hits the open sensor.



▼ Network 7: Close gripper

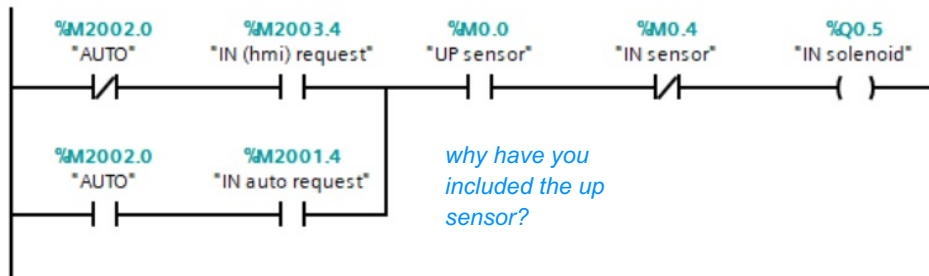
- ▼ Close gripper can be controlled manually or automatically via auto switch and request from HMI screen. The gripper will close until it hits the open sensor *wrong comment*



all figures should be numbered or simply placed in the appendix with a title.

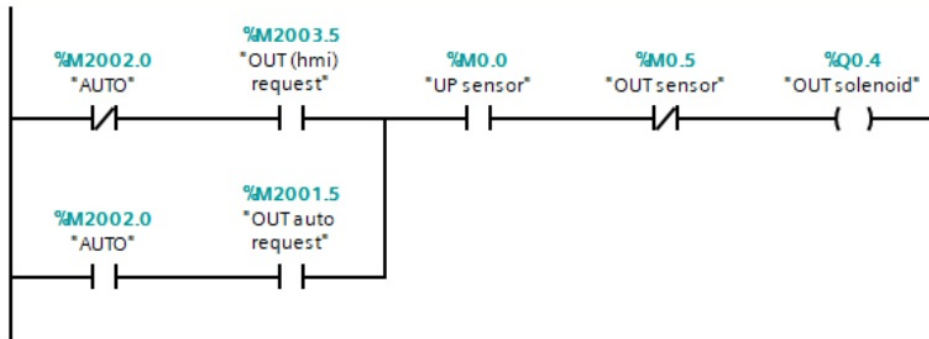
Network 8: Move in

- IN solenoid can be controlled manually or automatically via auto switch and request from HMI screen. The robot arm will extend in and will stop when it hits the IN sensor.



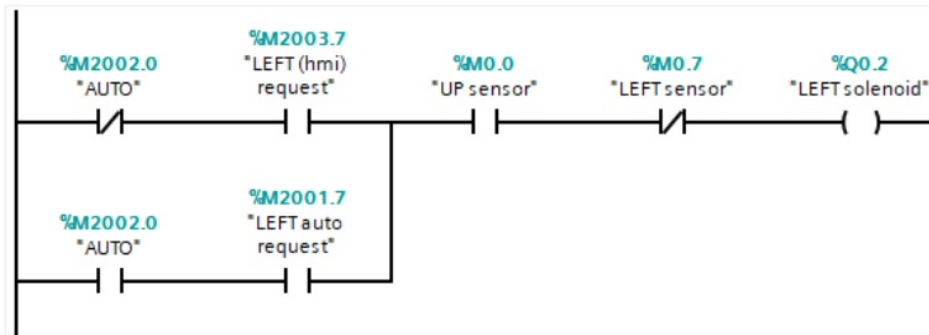
Network 9: Move out

- OUT solenoid can be controlled manually or automatically via auto switch and request from HMI screen. The robot arm will extend outwards and stop when it reaches the OUT sensor.



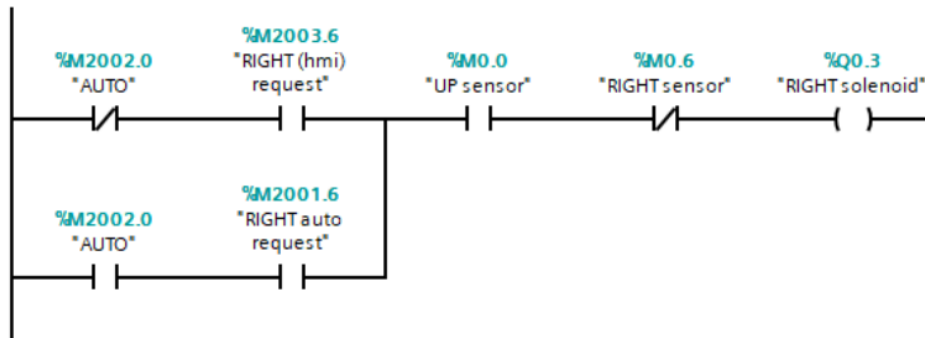
Network 10: Move left

- LEFT solenoid can be controlled manually or automatically via auto switch and request from HMI screen. The robot arm will move left and stop when it reaches the LEFT sensor.



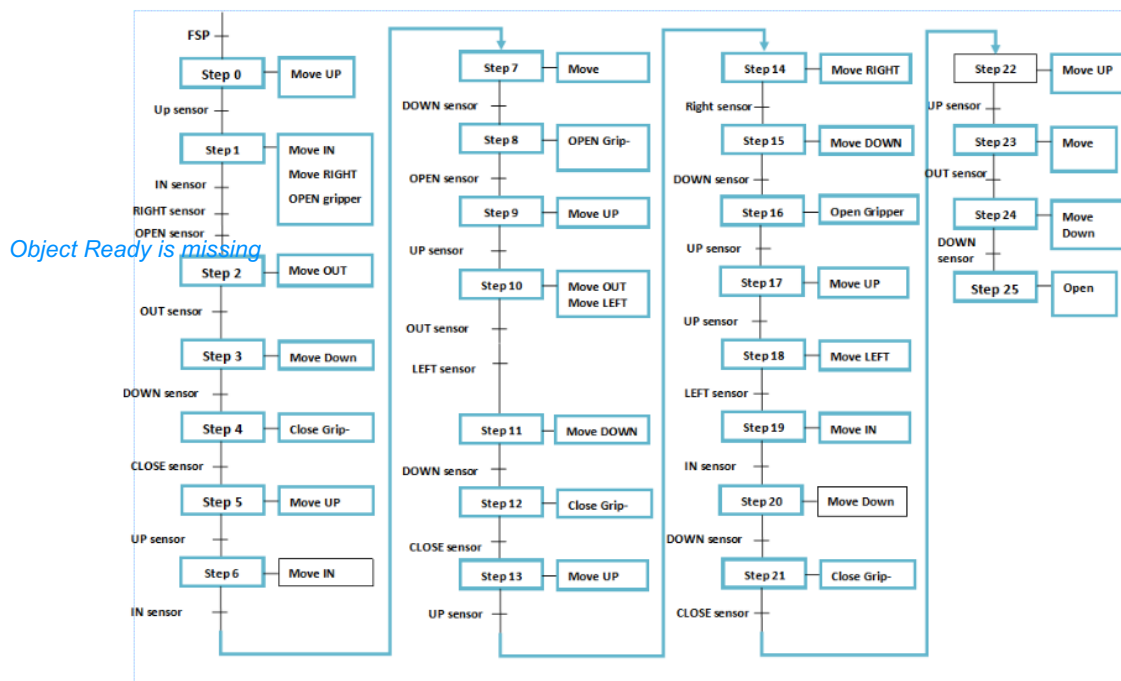
Network 11: Move right

- RIGHT solenoid can be controlled manually or automatically via auto switch and request from HMI screen. The robot arm will move right and stop when it reaches the RIGHT sensor.



Automatic Control

Before attempting to write the automatic sequence, a sequential function chart is created according to the specifications of the robot. This provides a step by step analysis of what should happen in each step and what conditions needs to be met before progressing to the next step. The PLC will only run this sequence when in automatic mode and the start button has been pressed.



all figures should be numbered or simply placed in the appendix with a title.

Before the robot can move positions, it must be in the UP position. This is because it would damage the machine if it were to move IN, OUT, LEFT or RIGHT. Therefore the first step 'step 0' is to move up. For the sequence to progress to step 1 conditions must be met, this is the UP sensor and must read true. Only when the UP sensor is true is that the sequence progresses to step 1.

1: //move home			
"IN auto request" := 1;	"IN auto r..."	%M2001.4	TRUE
"RIGHT auto request" := 1;	"RIGHT aut..."	%M2001.6	TRUE
"OPEN auto request" := 1;	"OPEN auto..."	%M2001.2	TRUE
IF ("IN sensor" = 1) AND	Result		FALSE
("RIGHT sensor" = 1) AND	"RIGHT sen..."	%M0.6	TRUE
("OPEN sensor" = 1) AND	"OPEN sensor"	%M0.2	TRUE
("Start PB" = 1) THEN	"Start PB"	%M2002.1	FALSE
"RIGHT auto request" := 0;	"RIGHT aut..."	%M2001.6	FALSE
"IN auto request" := 0;	"IN auto r..."	%M2001.4	FALSE
"OPEN auto request" := 0;	"OPEN auto..."	%M2001.2	FALSE
IF ("Object counter".CV = 1) THEN	Result		
"NEXT STEP" := 18;	"NEXT STEP"	%MW130	
END_IF;			
IF ("Object counter".CV = 2) THEN	Result		
"NEXT STEP" := 10;	"NEXT STEP"	%MW130	
END_IF;			
IF ("Object counter".CV = 3) THEN	Result		
"NEXT STEP" := 2;	"NEXT STEP"	%MW130	
END_IF;			

You should have used the Object Ready tag here instead of the Start PB

The image above is a screenshot of step 1 in the automatic sequence and is written in SCL (Structured Control Language). The condition to progress into step 2 is false as seen in the image. To get into step 2, the sensors IN, RIGHT, OPEN must be true and the Start PB also true. Looking closely we see that the sensors are true but the Start PB is false. This indicates that the start button the HMI screen needs to be pressed to start the sequence. When pressed the condition will be true and open the auto request contacts on the main OB1 program block and continue to the next step. However if there is one object in the machine it will skip to step 19, two objects skip to step 10 and 3 objects skip to step 2.

Simulation Testing

Every aspect of the program was tested to prove it is safe to operate and it works as intended. The first test carried out involved critical safety interlocks that prevented the robot from crashing into one of the stations, so it was very important this test was completed and correct before continuing. How the robot was to crash was if the robot were to move freely whilst being down, or if the robot were to move down whilst not over hovering above a station – in between a station.

Safety Interlock Testing Table				
Test Number	Description	Expected Result	Actual Result	Pass/Fail
1	Attempt to move left while arm is up	Free movement	Free movement	Y
2	Attempt to move right while arm is up	Free movement	Free movement	Y
3	Attempt to move in while arm is up	Free movement	Free movement	Y
4	Attempt to move out while arm is up	Free movement	Free movement	Y
5	Attempt to move left while arm is down	No movement	No movement	Y
6	Attempt to move right while arm is down	No movement	No movement	Y
7	Attempt to move in while arm is down	No movement	No movement	Y
8	Attempt to move out while arm is down	No movement	No movement	Y
9	Attempt to move down while arm is between left and right	No movement	No movement	Y
10	Attempt to move down while arm is between in and out	No movement	No movement	Y

As can be seen in the above table, the actual result matched as intended with the expected result, therefore the test was a pass, and the robot is not able to crash into a station and cause damage to the equipment.

The first tests produced was an HMI function test and the HMI indication lights to prove the buttons, counter, sensor and solenoid lights were working as intended.

HMI Function Testing			HMI Indication Lights Testing	
Button	Description	Passed (Y/N)	Lights	Passed (Y/N)
Up	Move up	Y	Up solenoid	Y
Down	Move down	Y	Up sensor	Y
Left	Move left	Y	Down solenoid	Y
Right	Move right	Y	Down sensor	Y
In	Move in	Y	Left solenoid	Y
Out	Move out	Y	Left sensor	Y
Open	Open the gripper	Y	Right solenoid	Y
Close	Close the gripper	Y	Right sensor	Y
Start	Start the sequence	Y	In solenoid	Y
Pause	Pause the sequence	Y	In sensor	Y
Simulate	Switch between real inputs/memory inputs	Y	Out solenoid	Y
Auto/Manual	Switch between automatic/manual mode	Y	Out sensor	Y
Operator Screen	Opens the operator screen	Y	Close solenoid	Y
Counter reset	Resets the object counter	Y	Close sensor	Y
Step reset	Resets the step counter and robot returns to home	Y		
Engineer's Screen	Opens the engineers screen	Y		

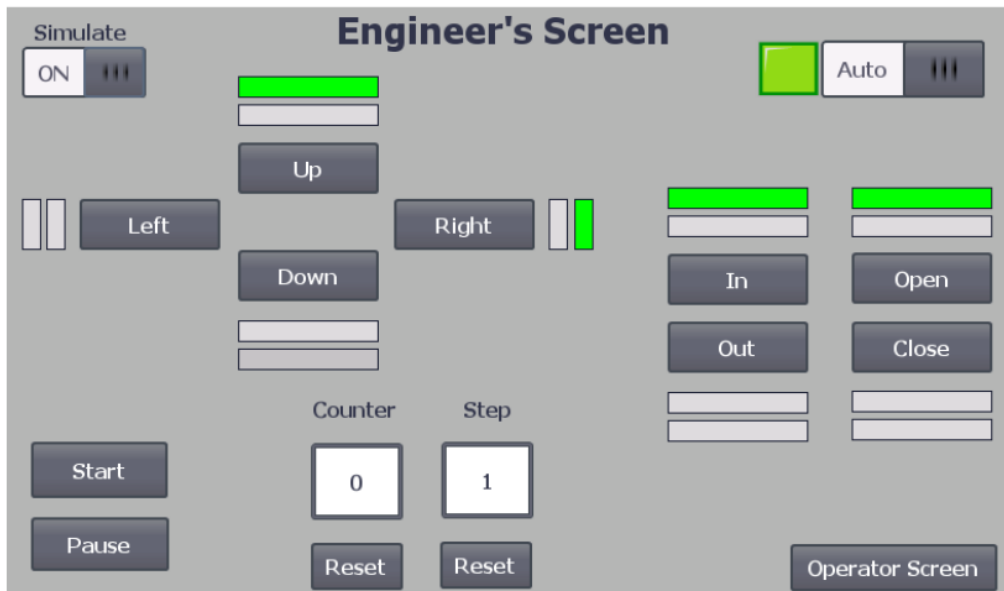
This test also proved that full manual control for the operator was achieved. The automatic sequence was tested next by observing what the robot did at each step of the sequence. For each step, if the robot behaved as expected, that would constitute as a pass.

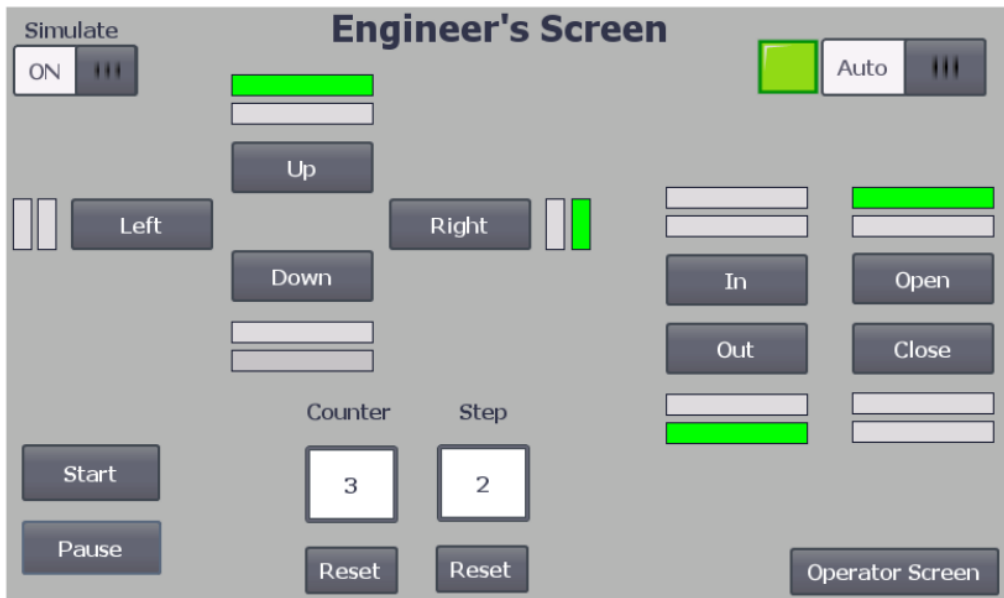
Automatic Sequence Testing		
Step No.	Description	Passed (Y/N)
0	Move up	Y
1	Move robot in, right and open	Y
2	Move out	Y
3	Move down	Y
4	Close the gripper	Y
5	Move up	Y
6	Move in	Y
7	Move down	Y
8	Open the gripper	Y
9	Move up	Y
10	Move out and left	Y
11	Move down	Y
12	Close the gripper	Y
13	Move up	Y
14	Move right	Y
15	Move down	Y
16	Open the gripper	Y
17	Move up	Y
18	Move left	Y
19	Move in	Y
20	Move down	Y
21	Close the gripper	Y
22	Move up	Y
23	Move out	Y
24	Move down	Y
25	Open the gripper & loop back to step 0	Y

From the above testing table, it is gathered that each step of the sequence performed as intended and passed the test. Therefore, it is established that the automatic sequence is fully correct and is ready to be downloaded to the PLC.

Practical Testing and Team Working

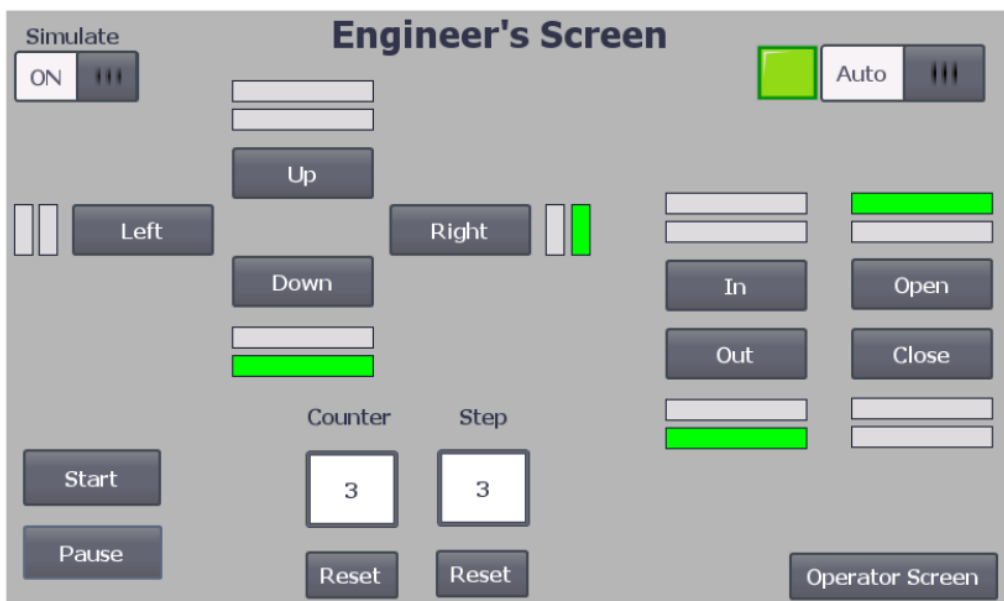
When part of a team, the results and program was compared, and the best program was selected. To prove the automatic sequence is working, the sequence was paused at each step in order to take a screenshot of the HMI screen showing what sensors were made. This gave indication the position of the robot, and was then compared with the SFC to prove the sequence is correct and doing what it should be. As a quick reference – the first light (closest to the buttons) indicates the solenoid, and the second light (furthest from the button) indicates the state of the sensor. When It turns green it indicates that the sensor/solenoid is made.

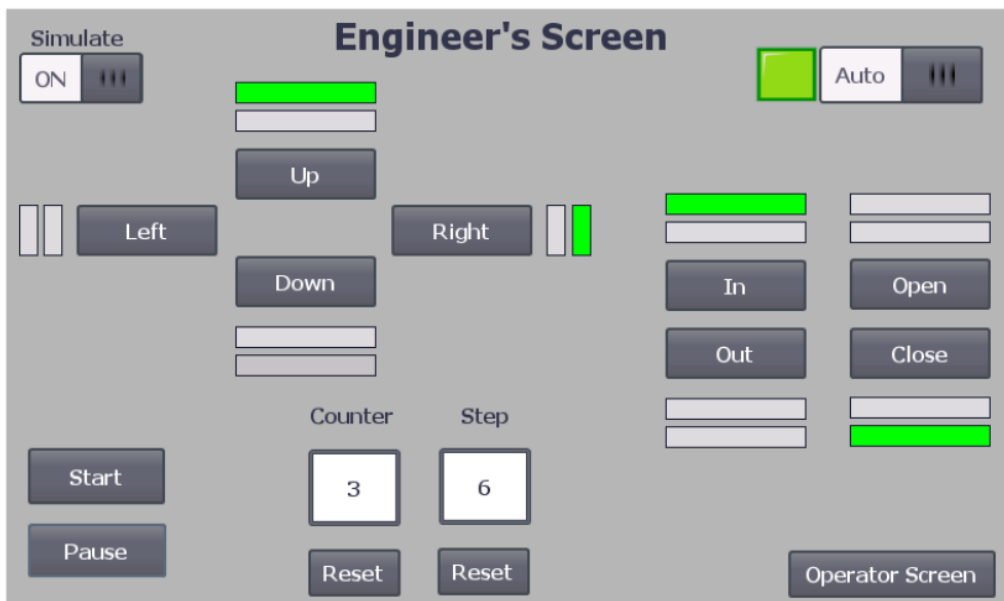
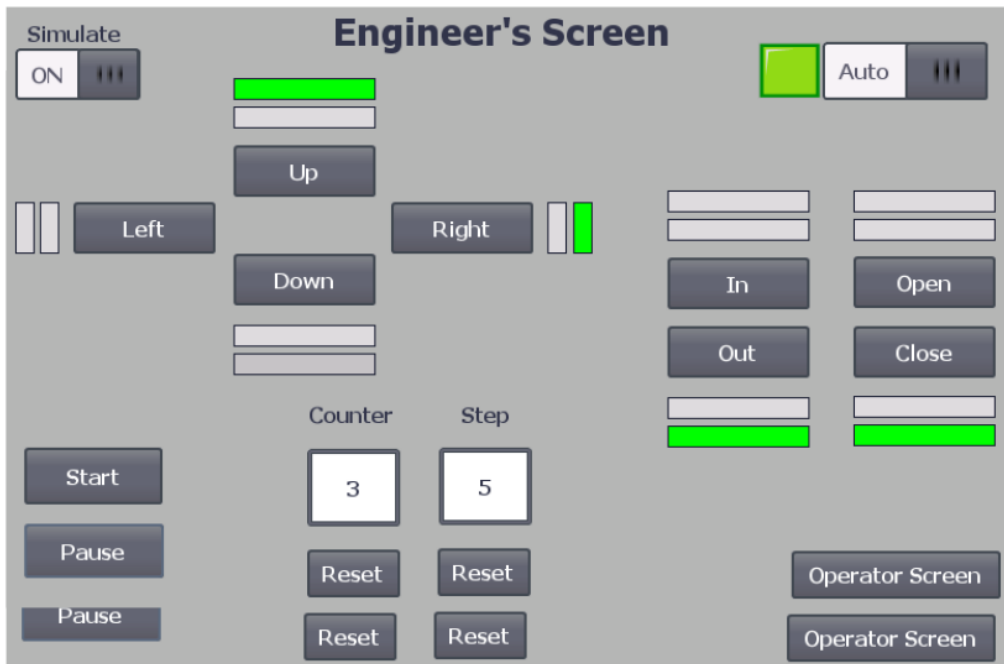


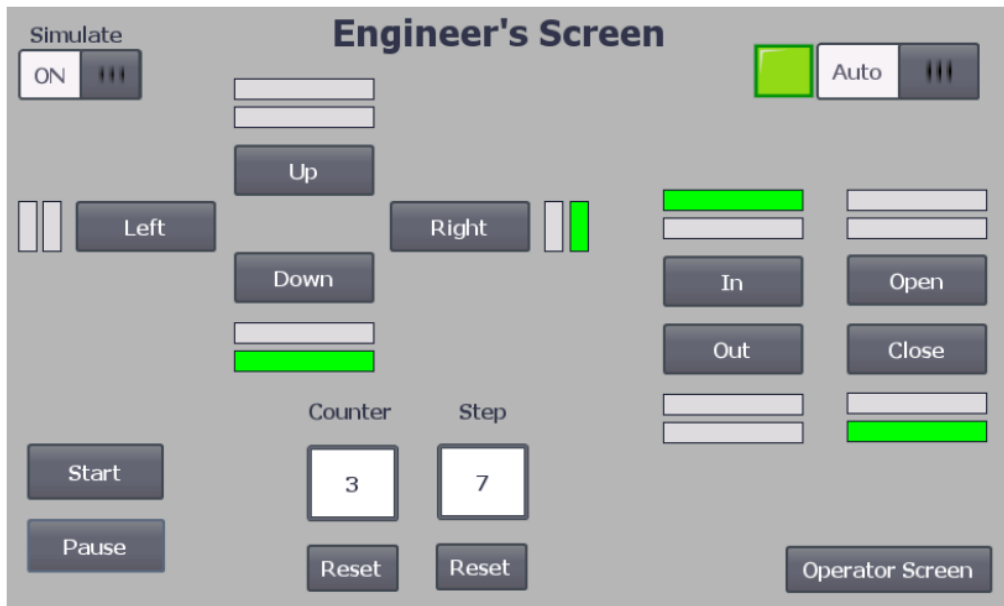


The reason why the counter has jumped to 3 is that when there is 1 object detected in the machine, the automatic sequence will jump to step 18, and if there are 2 objects detected, the sequence will jump to step 10. So in purposes of proving the automatic sequence, the program was forced to see 3 objects in the machine to test steps 2 to 7.

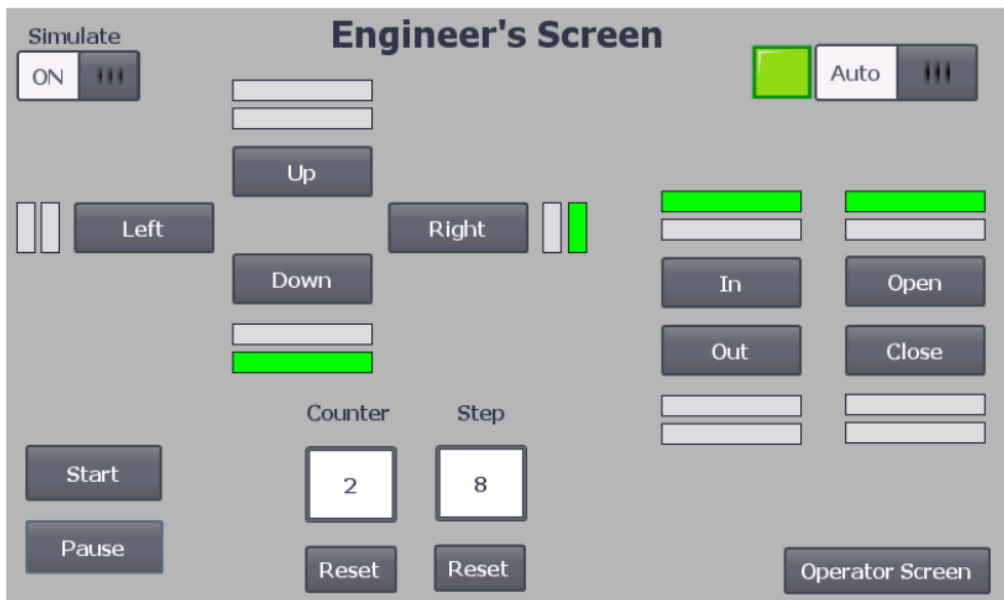
These screen belong in the appendix



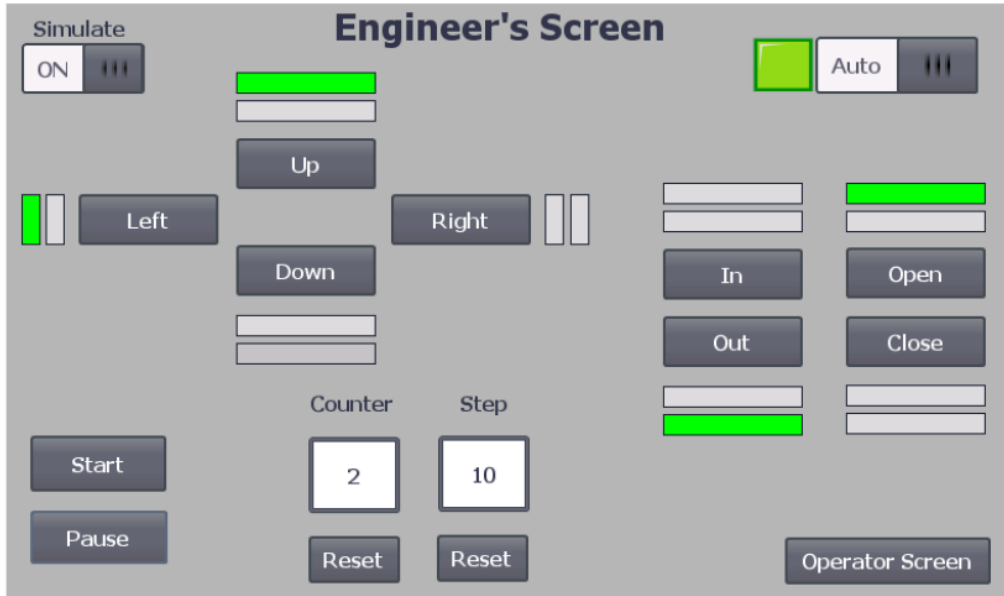
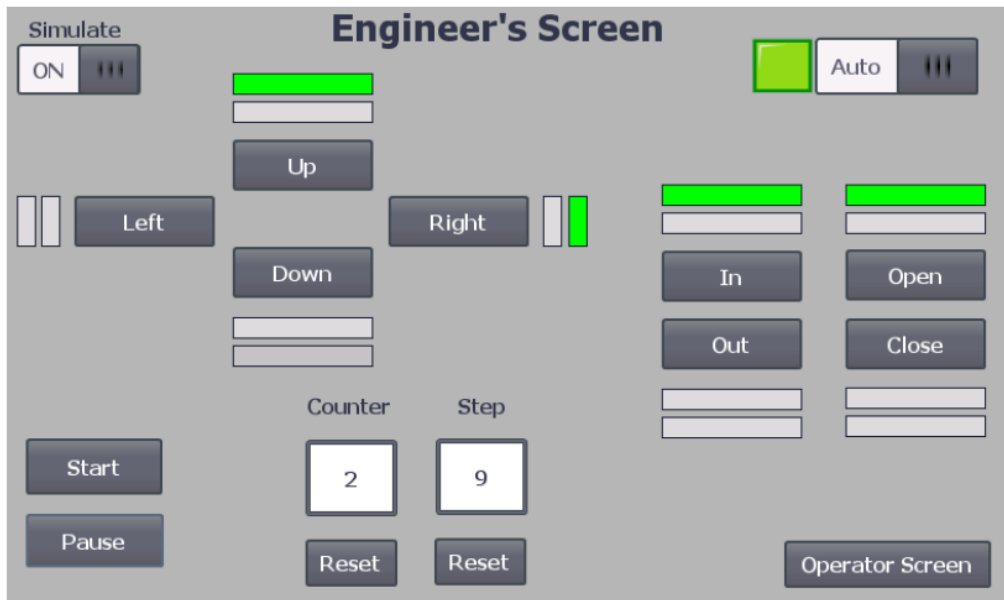


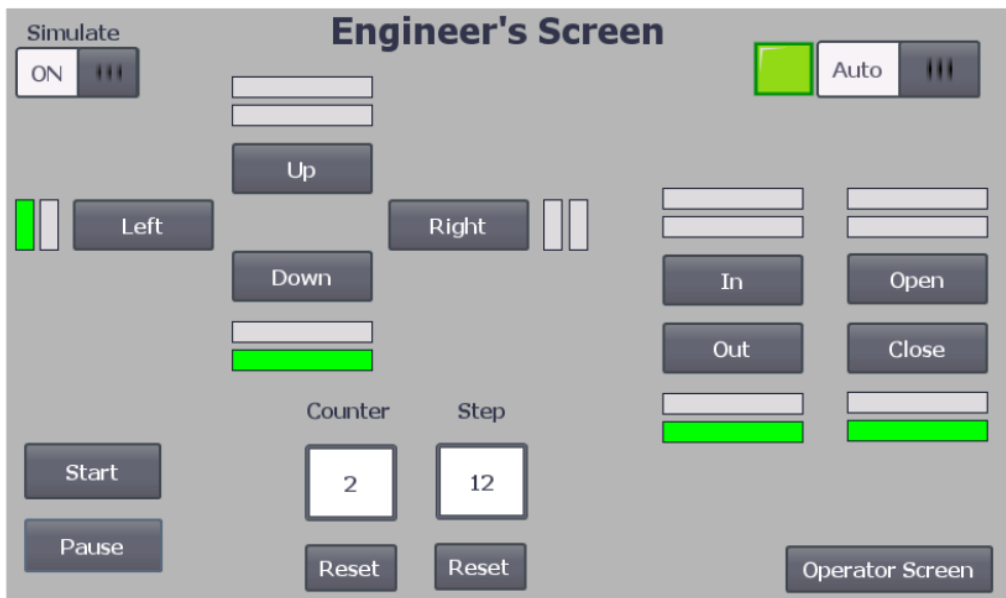
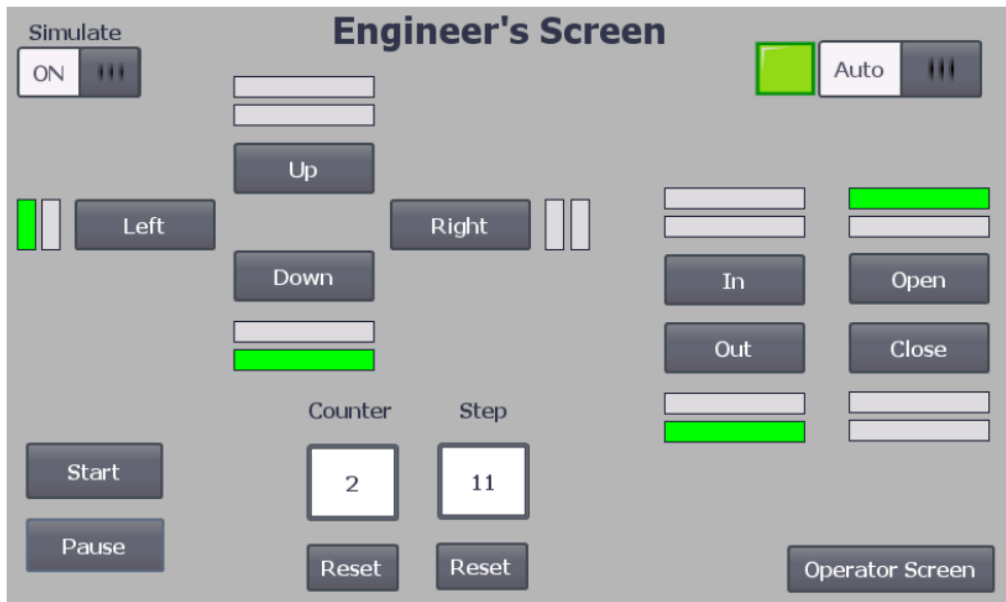


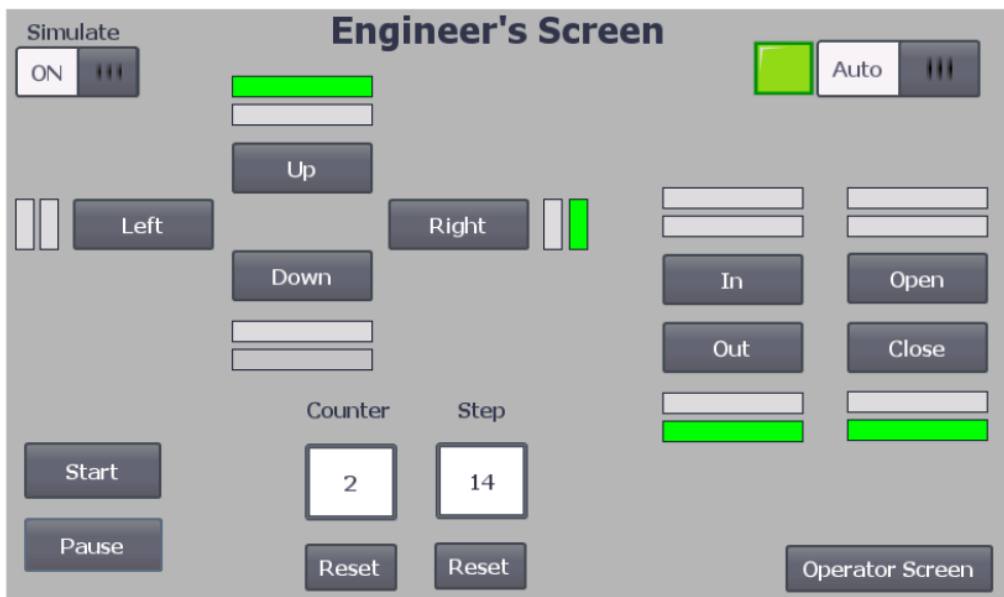
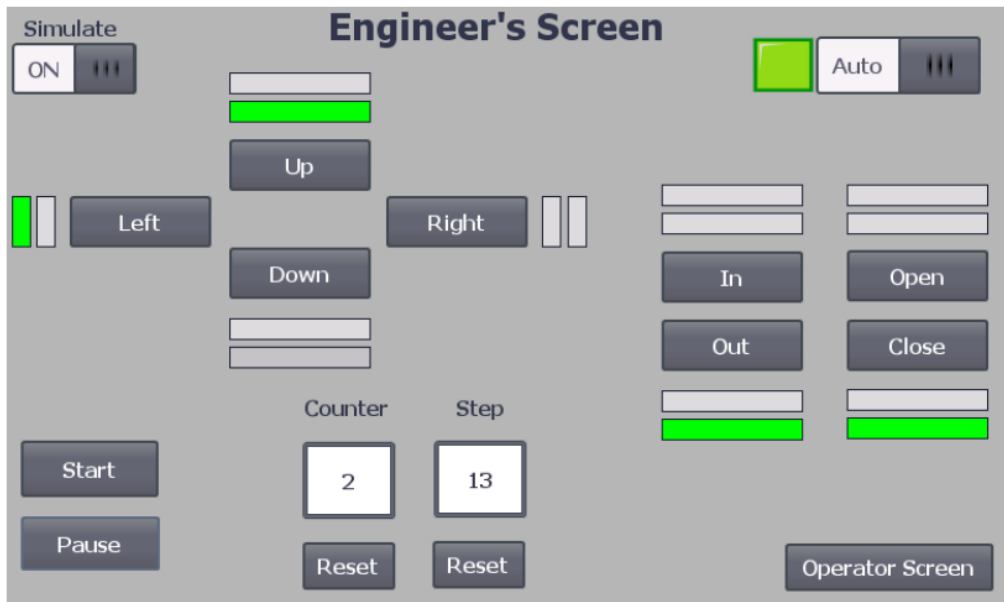
all figures should be numbered or simply placed in the appendix with a title.

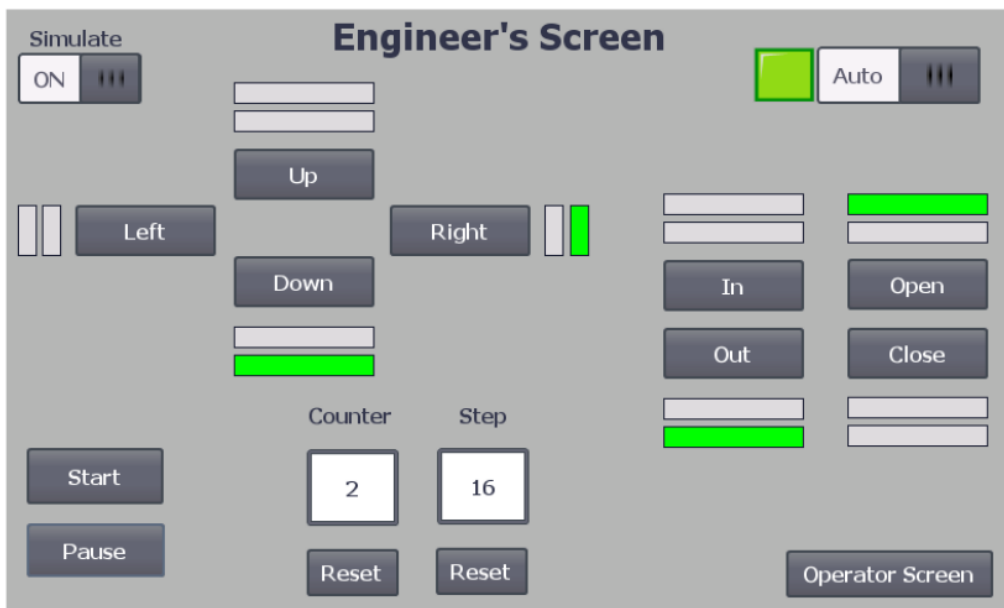
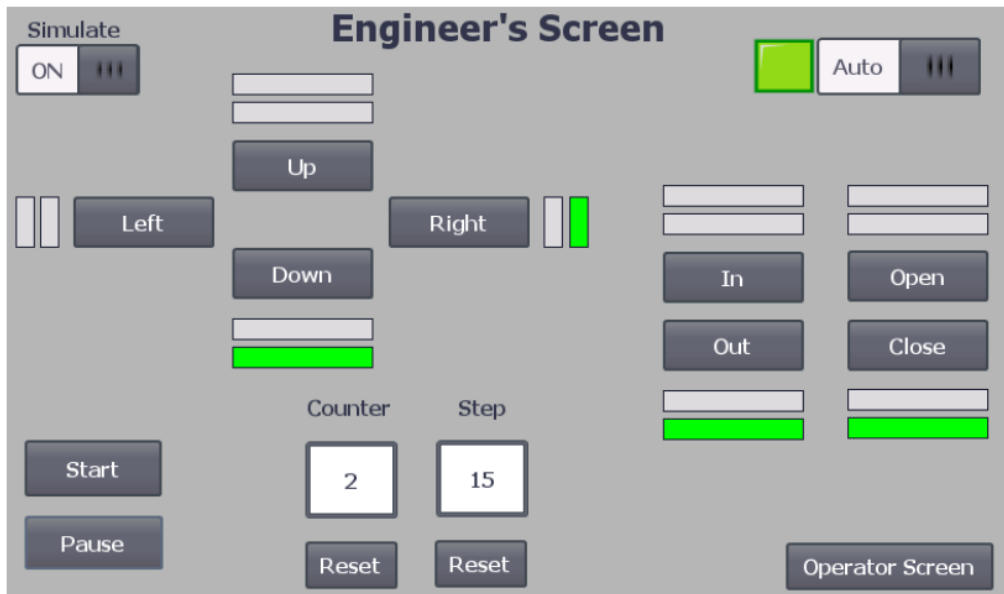


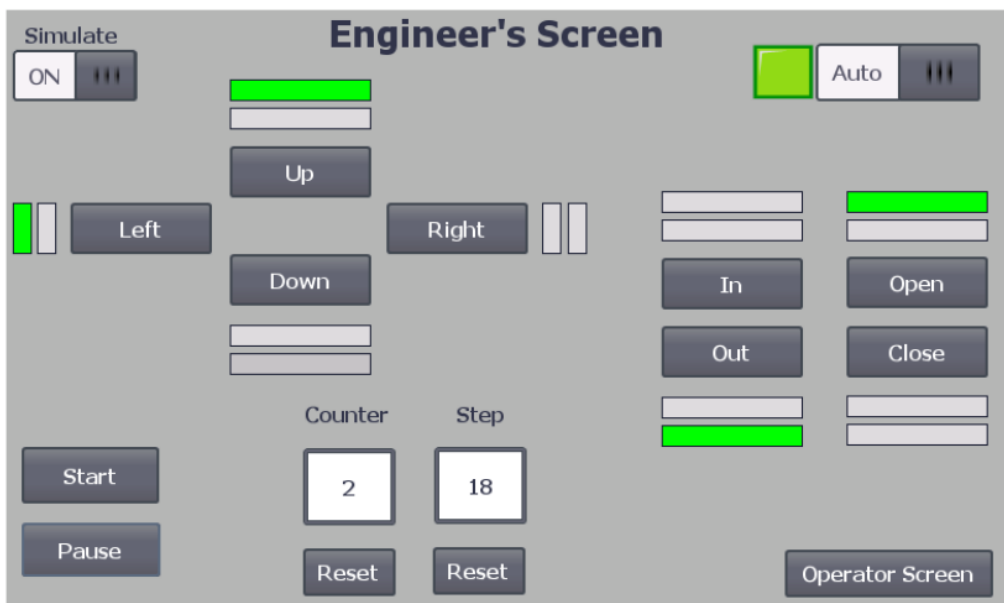
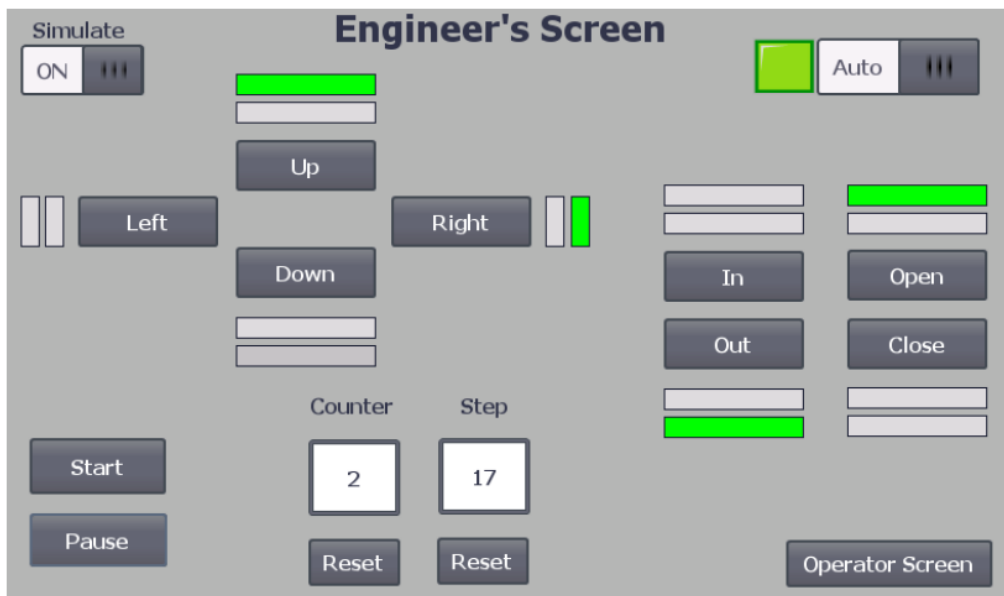
Step 8 of the automatic sequence opens the gripper arm and releases the object through the exit chute and the object counter responds with minus 1 object in the machine.

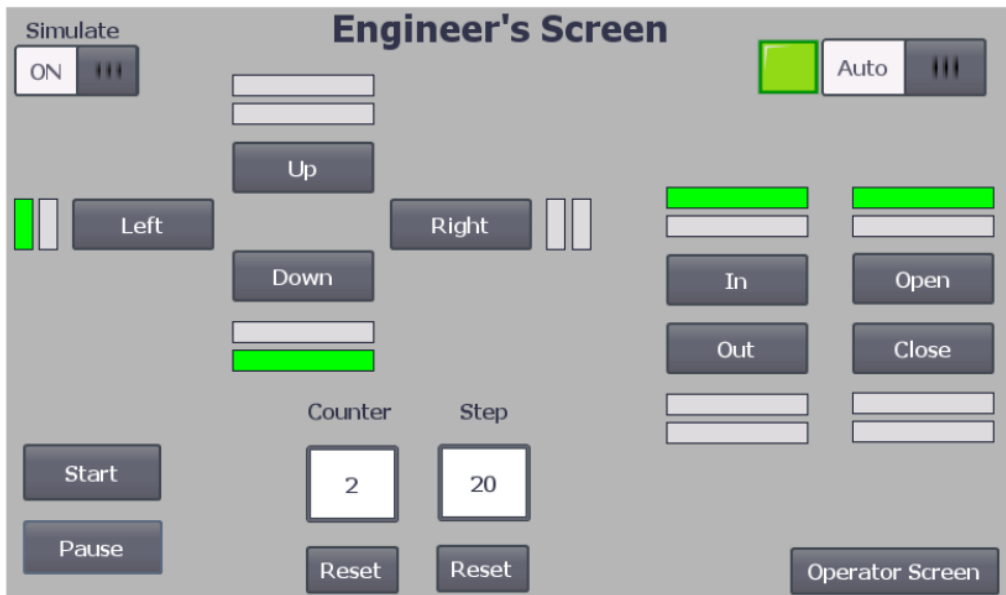
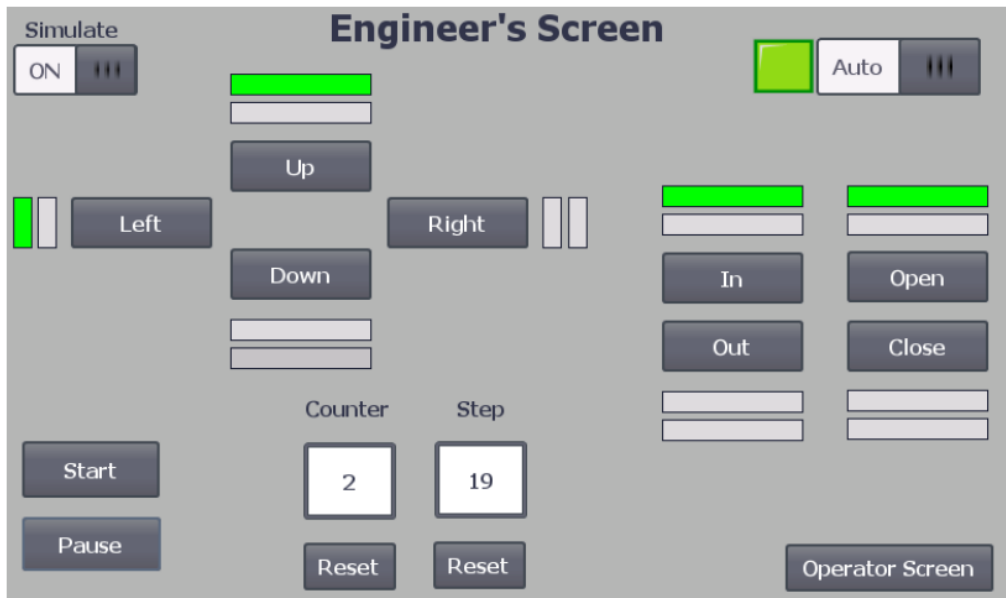


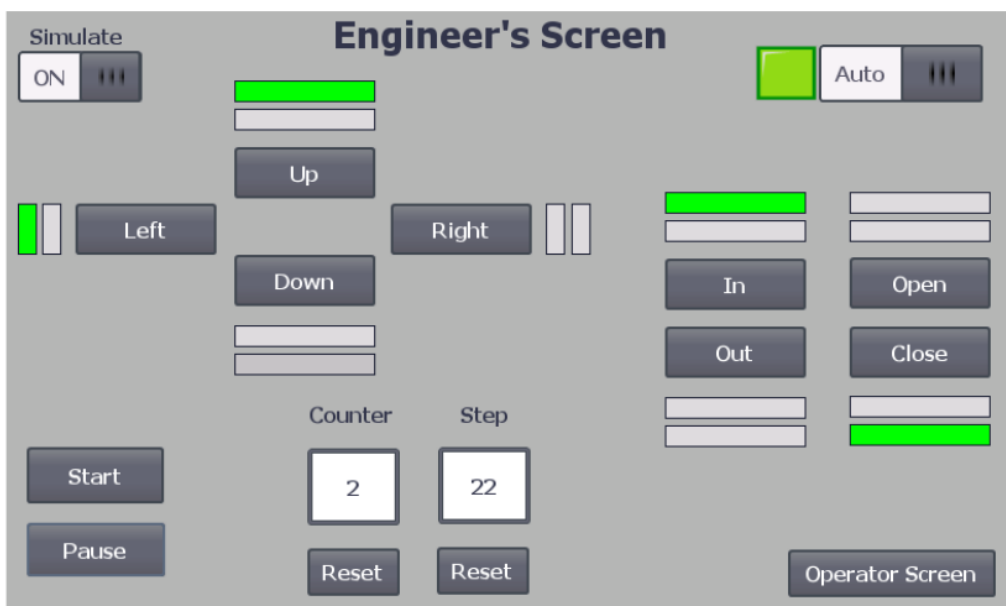
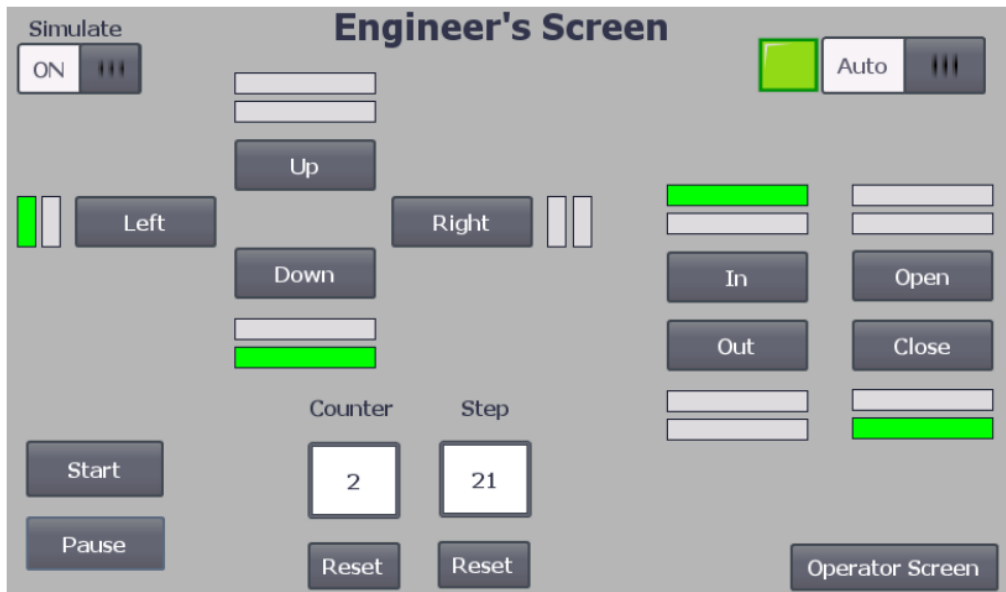


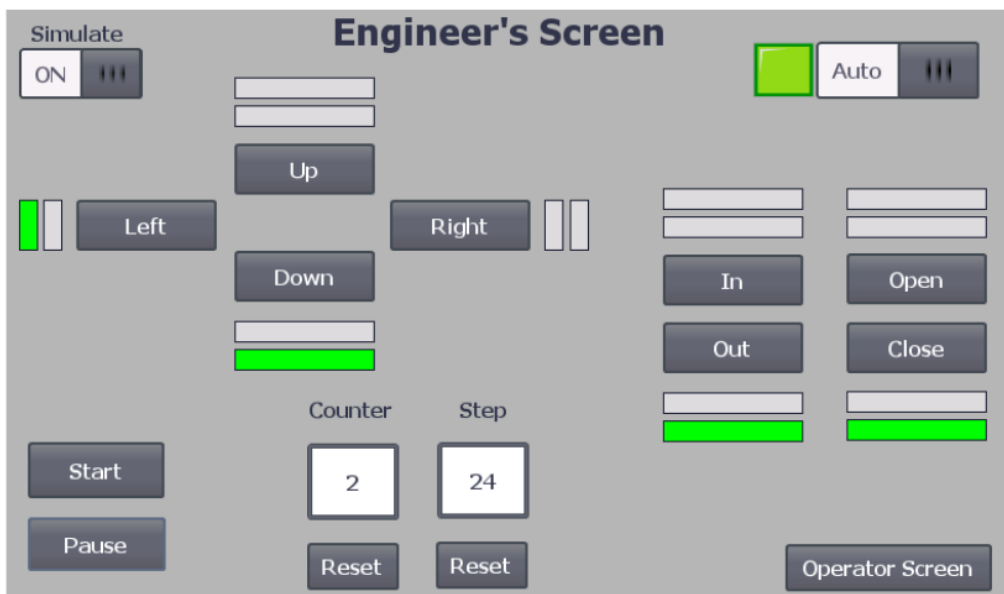
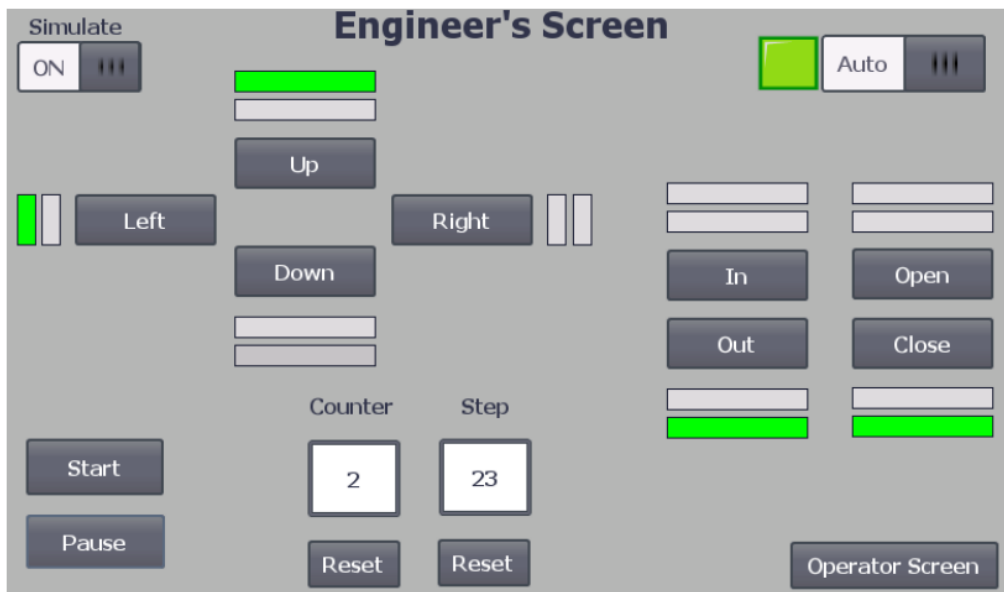


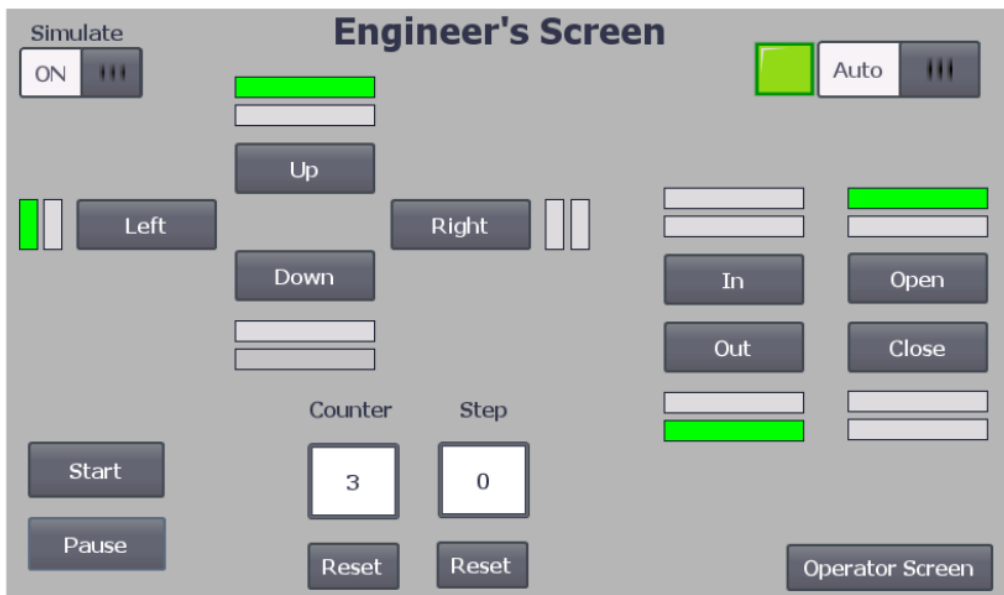
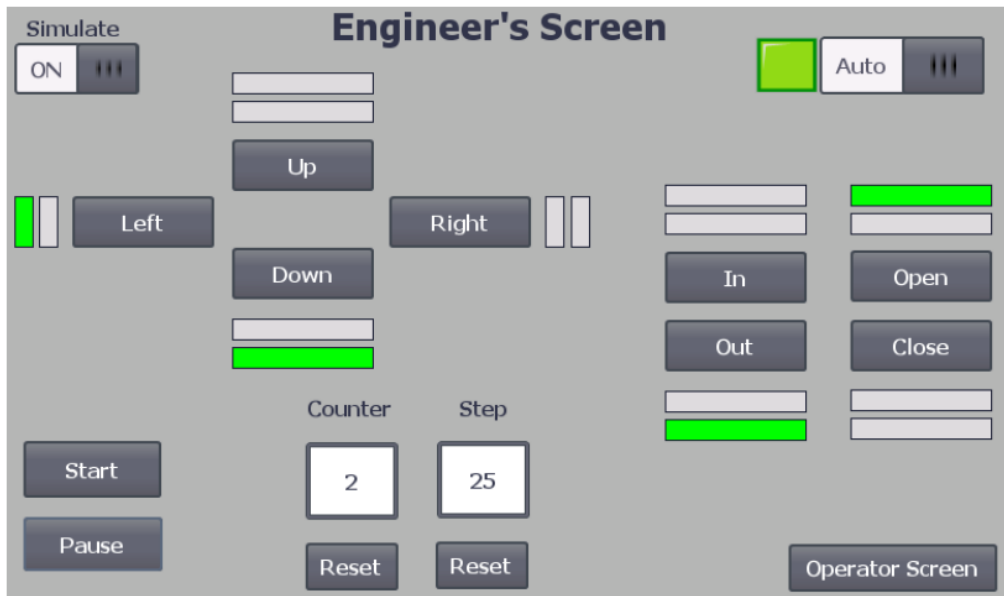












Each step was checked to see if the sensors corresponded with the sequential flow chart to determine if it was correct.

Discussion, Conclusion and Recommendations

From the simulation testing results and the practical results, it is concluded that the system is working as intended and is fully operational for the customer and user.

The manual overrides were written to the main block (OB1) of the program and performed as they should be, which can be seen in the testing table produced earlier in the report. This mode can be used to reset the positioning of the robot if it was stuck somewhere in the automatic sequence. It can also be used for maintenance purposes. For example, you can test each function of the robot via manual mode to check it is working, or if you had to carry out maintenance on the robot but could not access due to the robot arm being up, then simply press the down button on the HMI to gain access. Also providing the step number display can be helpful to view if attending on a machine breakdown as you can see at what part of the sequence the breakdown occurred.

As the solenoids only having one network run each and nowhere else in the program. Also, in the automatic and manual control, only requests were used to bring on the solenoids. This meant that in the event of a fault, the engineer would only have to look at these network rungs as these are the only possible places where the solenoids are energized. This adds reliability to the program. *good comment*

The main priority in the program was to make sure the robot cannot crash into one of the stations and cause damage to the equipment. It was concluded, the only way the robot was to crash, was if it could move freely whilst the robot was down. The solution to this was done by adding safety interlocks in the network rungs to only allow movement when certain conditions were met. These conditions were sensors to indicate correct positioning of the robot before it is able to move down. The safety interlock testing table provided in the Simulation Testing section, confirms the safety interlocks are working as intended and did not allow the robot to move left, right, in and out whilst the robot was down. This enables the user, that he or she can operate the machine safely.

Observing the practical results which is included in the previous section, we see that throughout each step of the sequence that it corresponds exactly to the SFC in the report. Also, when the object counter displays 1 object in the system, the sequence jumps to step 18, if 2 objects are detected, the sequence jumps to 11, and when 3 objects are detected, the sequence will continue from step 1. These behaviors correspond correctly to what is written to the automatic sequence; therefore the automatic sequence is correct and is ready for operation.

It can be pointed out that the HMI screens are not very user friendly as there is not a lot of text information to go with the indication lights. So, this is relying on the user to have been shown the screen by someone that knows the system, or read the handover manuals. The recommendation for this is to have further work done to the HMI to add in text information to each of the indication lights, which will involve a re-design of the layout structure.

THIS IS NOT A CITATION LIST

Bibliography

Although there is only one direct citations in the body of the report, a lot of the research was obtained through the Siemens website and suppliers website RS Components.

Wrong method of listing, this is not

Harvard

RS Components, S7-1200 Datasheet <https://docs.rs-online.com/56ac/0900766b814a40bc.pdf>

S7-1500 <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500.html>

S7-300 - <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-300.html>

S7-400 - <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-400.html>

S7-1200 - <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/s7-1200.html>

PT. VIKTORI PROFINDO AUTOMATION - <https://www.viktori-automation.com/id/produk/s7-400-2/>

Appendix

PLC Tag List

PLC tags									
	Name	Tag table	Data type	Address	Retain	Access...	Writa...	Visibl...	
1	PLC Base Unit input byte 0	Assignment tag...	Byte	%IB0	☐	☒	☒	☒	
2	PLC Base Unit input byte 1	Assignment tag table	Byte	%IB1	☐	☒	☒	☒	
3	DOWN solenoid	Assignment tag table	Bool	%Q0.0	☐	☒	☒	☒	
4	UP solenoid	Assignment tag table	Bool	%Q0.1	☐	☒	☒	☒	
5	LEFT solenoid	Assignment tag table	Bool	%Q0.2	☐	☒	☒	☒	
6	RIGHT solenoid	Assignment tag table	Bool	%Q0.3	☐	☒	☒	☒	
7	OUT solenoid	Assignment tag table	Bool	%Q0.4	☐	☒	☒	☒	
8	IN solenoid	Assignment tag table	Bool	%Q0.5	☐	☒	☒	☒	
9	OPEN solenoid	Assignment tag table	Bool	%Q0.6	☐	☒	☒	☒	
10	CLOSE solenoid	Assignment tag table	Bool	%Q0.7	☐	☒	☒	☒	
11	PLC Base Unit mapped input by..	Assignment tag table	Byte	%MB0	☐	☒	☒	☒	
12	UP sensor	Simulation Tags	Bool	%M0.0	☐	☒	☒	☒	
13	DOWN sensor	Simulation Tags	Bool	%M0.1	☐	☒	☒	☒	
14	OPEN sensor	Simulation Tags	Bool	%M0.2	☐	☒	☒	☒	
15	CLOSED sensor	Simulation Tags	Bool	%M0.3	☐	☒	☒	☒	
16	IN sensor	Simulation Tags	Bool	%M0.4	☐	☒	☒	☒	
17	OUT sensor	Simulation Tags	Bool	%M0.5	☐	☒	☒	☒	
18	RIGHT sensor	Simulation Tags	Bool	%M0.6	☐	☒	☒	☒	
19	LEFT sensor	Simulation Tags	Bool	%M0.7	☐	☒	☒	☒	
20	PLC Base Unit mapped input by..	Assignment tag table	Byte	%MB1	☐	☒	☒	☒	
21	X10 Pushbutton	Simulation Tags	Bool	%M1.0	☐	☒	☒	☒	
22	X11 Pushbutton	Simulation Tags	Bool	%M1.1	☐	☒	☒	☒	
23	X12 Pushbutton	Simulation Tags	Bool	%M1.2	☐	☒	☒	☒	
24	X13 toggle switch	Simulation Tags	Bool	%M1.3	☐	☒	☒	☒	
25	X14 toggle switch	Simulation Tags	Bool	%M1.4	☐	☒	☒	☒	
26	OBJECT sensor	Simulation Tags	Bool	%M1.5	☐	☒	☒	☒	
27	STEP	Assignment tag table	UInt	%MW120	☐	☒	☒	☒	
28	NEXT STEP	Assignment tag table	UInt	%MW130	☐	☒	☒	☒	
29	Simulate	Assignment tag table	Bool	%M2000.0	☐	☒	☒	☒	
30	UP auto request	Assignment tag table	Bool	%M2001.0	☐	☒	☒	☒	
31	DOWN auto request	Assignment tag table	Bool	%M2001.1	☐	☒	☒	☒	
32	OPEN auto request	Assignment tag table	Bool	%M2001.2	☐	☒	☒	☒	
33	CLOSE auto request	Assignment tag table	Bool	%M2001.3	☐	☒	☒	☒	
34	IN auto request	Assignment tag table	Bool	%M2001.4	☐	☒	☒	☒	
35	OUT auto request	Assignment tag table	Bool	%M2001.5	☐	☒	☒	☒	
36	RIGHT auto request	Assignment tag table	Bool	%M2001.6	☐	☒	☒	☒	
37	LEFT auto request	Assignment tag table	Bool	%M2001.7	☐	☒	☒	☒	
38	AUTO	Assignment tag table	Bool	%M2002.0	☐	☒	☒	☒	
39	Start PB	Assignment tag table	Bool	%M2002.1	☐	☒	☒	☒	
40	Stop pb	Assignment tag table	Bool	%M2002.2	☐	☒	☒	☒	
41	Object detected	Assignment tag table	Bool	%M2002.3	☐	☒	☒	☒	
42	Object ready	Assignment tag table	Bool	%M2002.4	☐	☒	☒	☒	

42		Object ready	Assignment tag table Bool	%M2002.4	☐	☒	☒	☒
43		reset object counter	Assignment tag table Bool	%M2002.5	☐	☒	☒	☒
44		RESET STEP	Assignment tag table Bool	%M2002.6	☐	☒	☒	☒
45		UP hmi request	Assignment tag table Bool	%M2003.0	☐	☒	☒	☒
46		DOWN (hmi) request	Assignment tag table Bool	%M2003.1	☐	☒	☒	☒
47		OPEN (hmi) request	Assignment tag table Bool	%M2003.2	☐	☒	☒	☒
48		CLOSE (hmi) request	Assignment tag table Bool	%M2003.3	☐	☒	☒	☒
49		IN (hmi) request	Assignment tag table Bool	%M2003.4	☐	☒	☒	☒
50		OUT (hmi) request	Assignment tag table Bool	%M2003.5	☐	☒	☒	☒
51		RIGHT (hmi) request	Assignment tag table Bool	%M2003.6	☐	☒	☒	☒
52		LEFT (hmi) request	Assignment tag table Bool	%M2003.7	☐	☒	☒	☒
53		System_Byte	System Tags Byte	%MB2100	☐	☒	☒	☒
54		FSP	System Tags Bool	%M2100.0	☐	☒	☒	☒
55		DiagStatusUpdate	System Tags Bool	%M2100.1	☐	☒	☒	☒
56		AlwaysTRUE	System Tags Bool	%M2100.2	☐	☒	☒	☒
57		AlwaysFALSE	System Tags Bool	%M2100.3	☐	☒	☒	☒
58		Clock_Byte	System Tags Byte	%MB2101	☐	☒	☒	☒
59		Clock_10Hz	System Tags Bool	%M2101.0	☐	☒	☒	☒
60		Clock_5Hz	System Tags Bool	%M2101.1	☐	☒	☒	☒
61		Clock_2.5Hz	System Tags Bool	%M2101.2	☐	☒	☒	☒
62		Clock_2Hz	System Tags Bool	%M2101.3	☐	☒	☒	☒
63		Clock_1.25Hz	System Tags Bool	%M2101.4	☐	☒	☒	☒
64		Clock_1Hz	System Tags Bool	%M2101.5	☐	☒	☒	☒
65		Clock_0.625Hz	System Tags Bool	%M2101.6	☐	☒	☒	☒
66		Clock_0.5Hz	System Tags Bool	%M2101.7	☐	☒	☒	☒
67		<Add new>			☐	☒	☒	☒

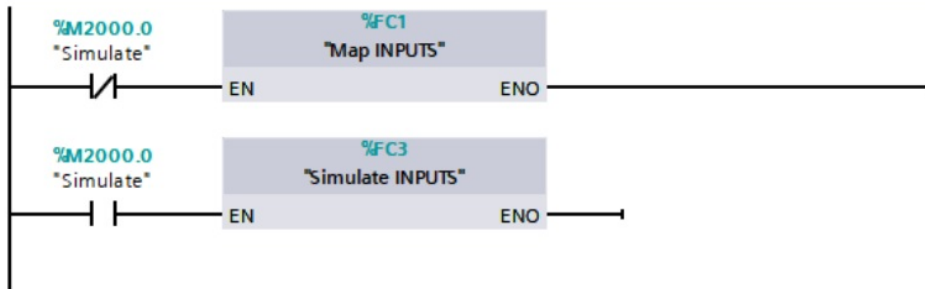
Program Main Block

▼ Block title: "Main Program Sweep (Cycle)"

Comment

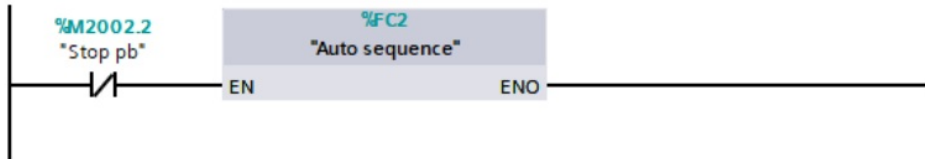
▼ Network 1: Real Inputs/Memory Inputs

Switch between real inputs or memory inputs. Via simulate switch on HMI engineer's screen.



▼ Network 2:

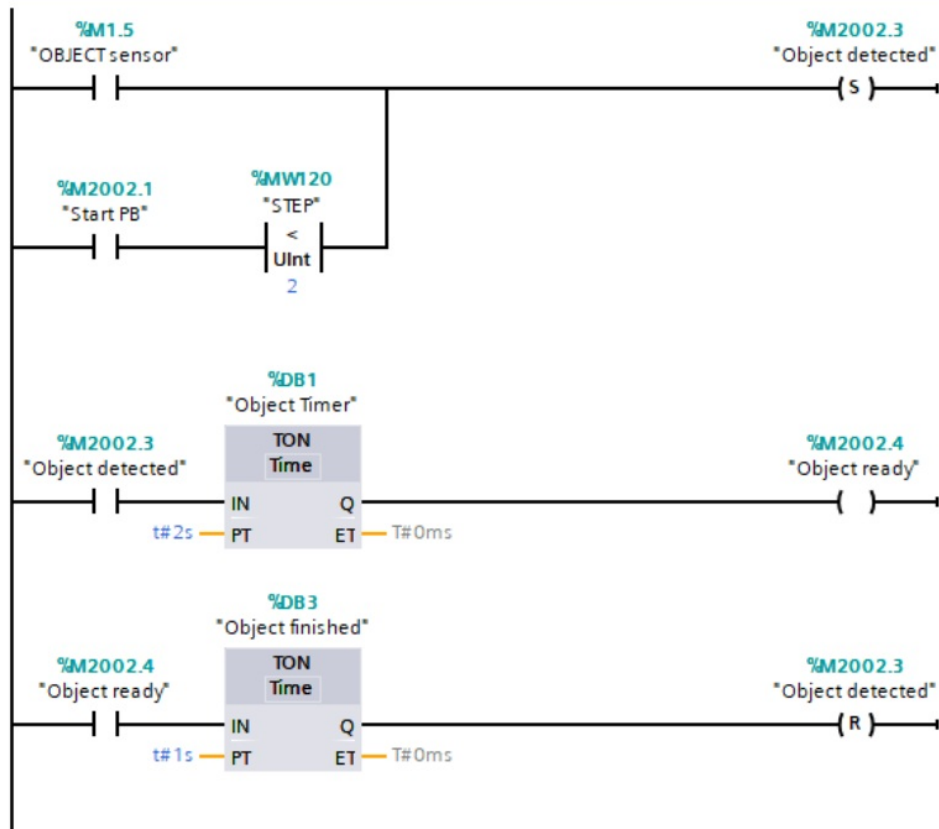
When start button is pressed on the HMI interface, the auto sequence will start.



Network 3: Object detection

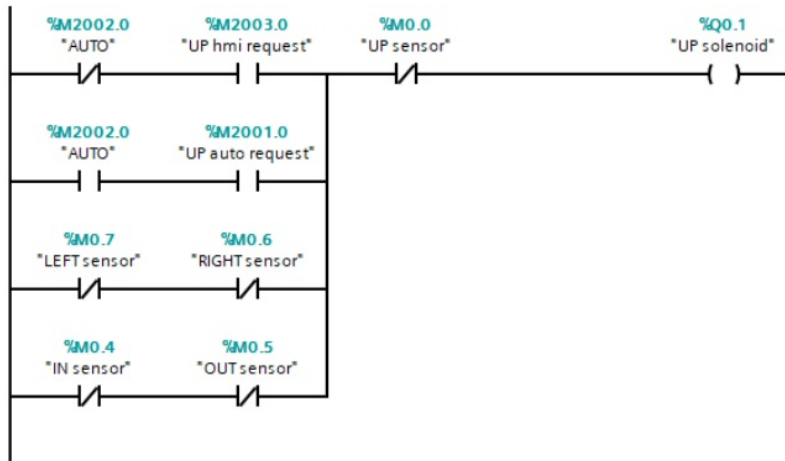
- When an object passes the object sensor, the contact will close and the object detected will be set on. It can also be set by pushing the start p/b when the sequence step is less than 2.

When the object detected is set, this will bring a timer on with a 2 second delay before bringing the object ready output on. When this output is on, the object ready contact closes and brings the object finished timer on with a 1 second delay until it resets the object detected.



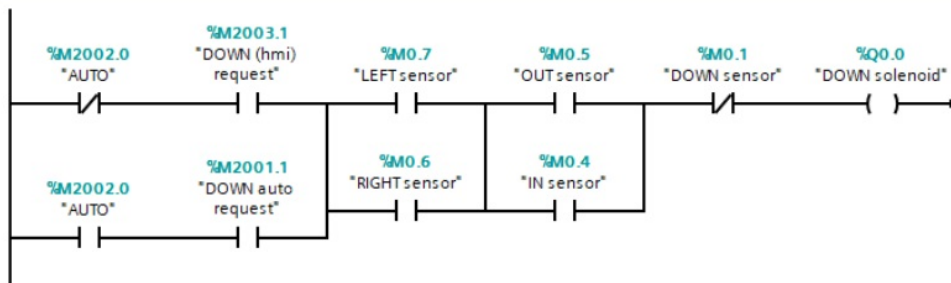
▼ Network 4: Move up

- UP solenoid can be controlled manually or automatically via auto switch and request from HMI screen. The robot arm will move up and stop when it reaches the UP sensor. If the robot is stuck inbetween left and right, or in and out, the robot will move up. This is a safety feature for the robot.



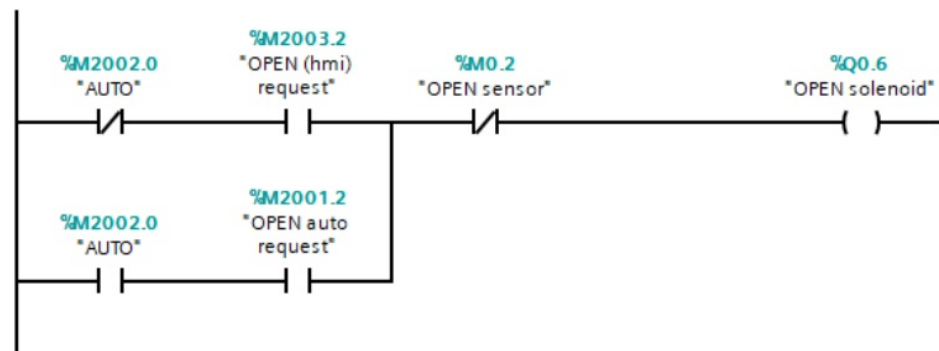
▼ Network 5: Move down

- DOWN solenoid can be controlled manually or automatically via auto switch and request from HMI screen. The robot arm will move down and stop when it reaches the DOWN sensor. It can only move down when it is above a station which are the LEFT, OUT, RIGHT, IN sensors. These are also safety interlocks.



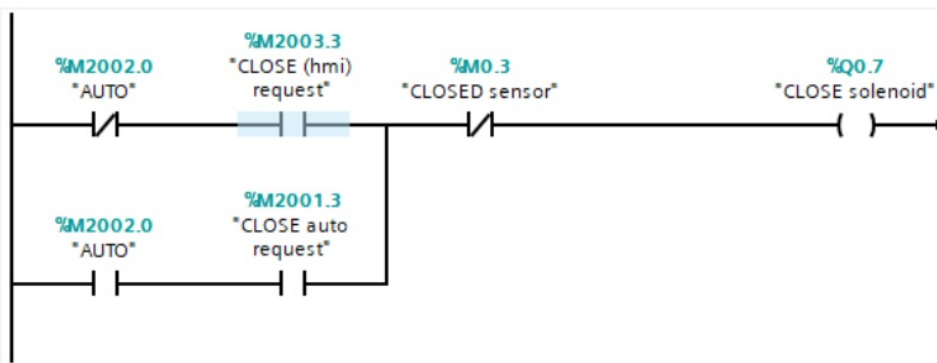
▼ Network 6: Open gripper

- Open gripper can be controlled manually or automatically via auto switch and request from HMI screen. The gripper will open until it hits the open sensor.



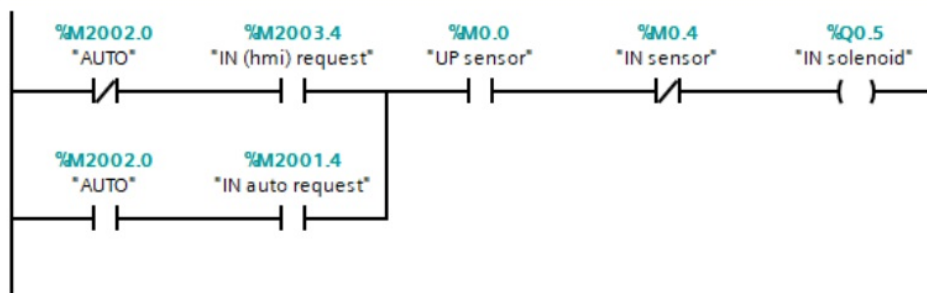
Network 7: Close gripper

- Close gripper can be controlled manually or automatically via auto switch and request from HMI screen. The gripper will close until it hits the open sensor.



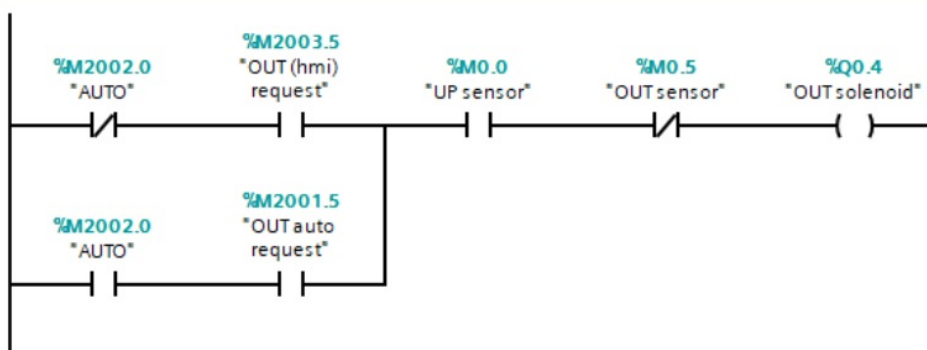
Network 8: Move in

- IN solenoid can be controlled manually or automatically via auto switch and request from HMI screen. The robot arm will extend in and will stop when it hits the IN sensor.



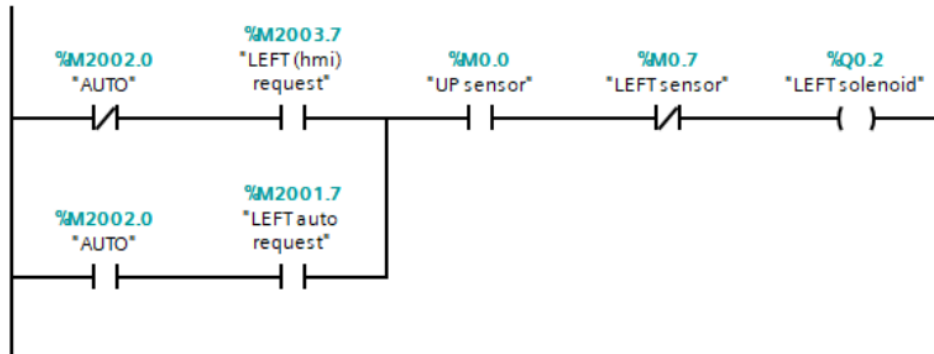
Network 9: Move out

- OUT solenoid can be controlled manually or automatically via auto switch and request from HMI screen. The robot arm will extend outwards and stop when it reaches the OUT sensor.



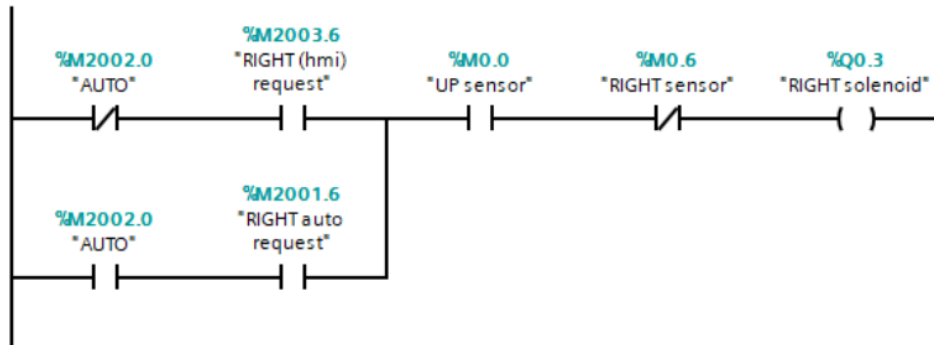
▼ **Network 10:** Move left

- ▼ LEFT solenoid can be controlled manually or automatically via auto switch and request from HMI screen. The robot arm will move left and stop when it reaches the LEFT sensor.



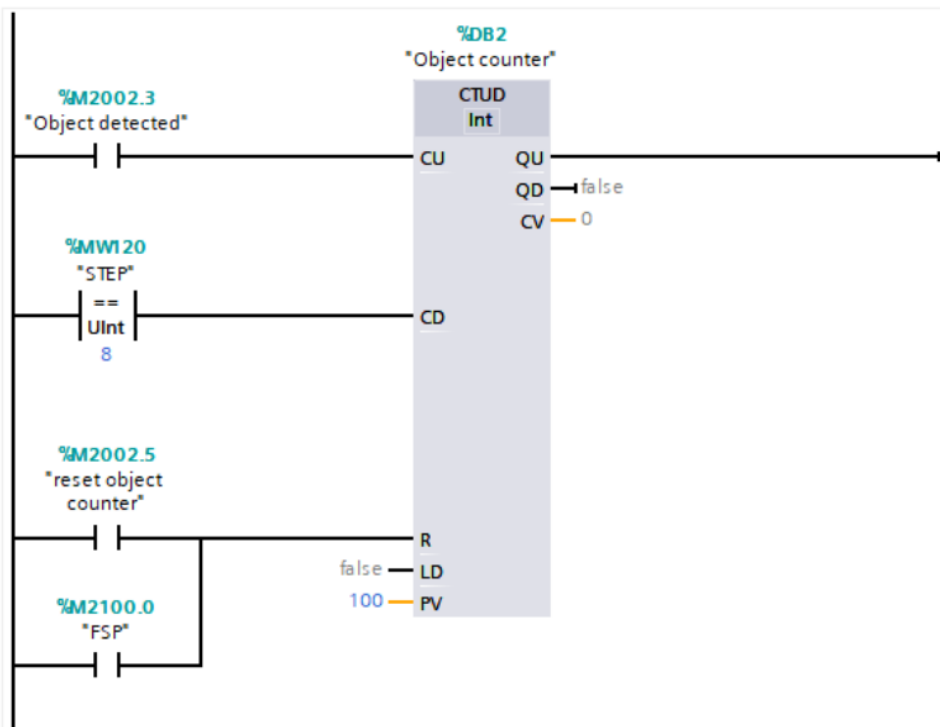
▼ **Network 11:** Move right

- ▼ RIGHT solenoid can be controlled manually or automatically via auto switch and request from HMI screen. The robot arm will move right and stop when it reaches the RIGHT sensor.



Network 12: Object counter

- When an object is detected via object sensor, data will be sent to DB2 "Object counter" to count up. When reset object counter is pressed on the HMI screen or FSP is true, the object counter will reset to zero. When step number equals 8, the counter will decrement by one.



Automatic Sequence

1	IF ("FSP"=1) OR ("AUTO"=0) OR ("RESET STEP"=1) THEN	▶	"FSP"	%M2100.0
2	"STEP" := 0;		"STEP"	%MW120
3	"NEXT STEP" := 0;		"NEXT STEP"	%MW130
4	"LEFT auto request" := 0;		"LEFT auto..."	%M2001.7
5	"RIGHT auto request" := 0;		"RIGHT aut..."	%M2001.6
6	"IN auto request" := 0;		"IN auto r..."	%M2001.4
7	"OUT auto request" := 0;		"OUT auto ..."	%M2001.5
8	"OPEN auto request" := 0;		"OPEN auto..."	%M2001.2
9	"CLOSE auto request" := 0;		"CLOSE aut..."	%M2001.3
10	END_IF;			
11				
12	CASE "STEP" OF		"STEP"	%MW120
13	0: //Robot must be UP before it does anything else.			
14	//This is for the safety of the robot.			
15	"UP auto request" := 1;		"UP auto r..."	%M2001.0
16	IF "UP sensor" = 1 THEN		"UP sensor"	%M0.0
17	"UP auto request" := 0;		"UP auto r..."	%M2001.0
18	"NEXT STEP" := 1;		"NEXT STEP"	%MW130
19	END_IF;			
20	1: //move home (IN, RIGHT & OPEN)			
21	"IN auto request" := 1;		"IN auto r..."	%M2001.4
22	"RIGHT auto request" := 1;		"RIGHT aut..."	%M2001.6
23	"OPEN auto request" := 1;		"OPEN auto..."	%M2001.2
24	IF ("IN sensor" = 1) AND		"IN sensor"	%M0.4
25	("RIGHT sensor" = 1) AND		"RIGHT sen..."	%M0.6
26	("OPEN sensor" = 1) AND		"OPEN sensor"	%M0.2
27	("Start PB" = 1) THEN		"Start PB"	%M2002.1
28	"RIGHT auto request" := 0;		"RIGHT aut..."	%M2001.6
29	"IN auto request" := 0;		"IN auto r..."	%M2001.4
30	"OPEN auto request" := 0;		"OPEN auto..."	%M2001.2
31	IF ("Object counter".CV = 1) THEN	▶	"Object co..."	%DB2
32	"NEXT STEP" := 18;		"NEXT STEP"	%MW130
33	END_IF;			
34	IF ("Object counter".CV = 2) THEN	▶	"Object co..."	%DB2
35	"NEXT STEP" := 10;		"NEXT STEP"	%MW130
36	END_IF;			
37	IF ("Object counter".CV = 3) THEN	▶	"Object co..."	%DB2
38	"NEXT STEP" := 2;		"NEXT STEP"	%MW130
39	END_IF;			
40	END_IF;			
41	2: //move object from station 3 to station 4			
42	//move out			
43	"OUT auto request" := 1;		"OUT auto ..."	%M2001.5
44	IF "OUT sensor" = 1 THEN		"OUT sensor"	%M0.5

```

45         "OUT auto request" := 0;
46         "NEXT STEP" := 3;
47     END_IF;
48     3: //move down
49         "DOWN auto request" := 1;
50     IF "DOWN sensor" = 1 THEN
51         "DOWN auto request" := 0;
52         "NEXT STEP" := 4;
53     END_IF;
54     4: //gripper close
55         "CLOSE auto request" := 1;
56     IF "CLOSED sensor" = 1 THEN
57         "CLOSE auto request" := 0;
58         "NEXT STEP" := 5;
59     END_IF;
60     5: //move up
61         "UP auto request" := 1;
62     IF "UP sensor" = 1 THEN
63         "UP auto request" := 0;
64         "NEXT STEP" := 6;
65     END_IF;
66     6: //move in
67         "IN auto request" := 1;
68     IF "IN sensor" = 1 THEN
69         "IN auto request" := 0;
70         "NEXT STEP" := 7;
71     END_IF;
72     7: //move down
73         "DOWN auto request" := 1;
74     IF "DOWN sensor" = 1 THEN
75         "DOWN auto request" := 0;
76         "NEXT STEP" := 8;
77     END_IF;
78     8: //gripper open
79         "OPEN auto request" := 1;
80     IF "OPEN sensor" = 1 THEN
81         "OPEN auto request" := 0;
82         "NEXT STEP" := 9;
83     END_IF;
84     9: //move up
85         "UP auto request" := 1;
86     IF "UP sensor" = 1 THEN
87         "UP auto request" := 0;
88         "NEXT STEP" := 10;

```

"OUT auto ...	%M2001.5
"NEXT STEP"	%MW130
"DOWN auto...	%M2001.1
"DOWN sensor"	%M0.1
"DOWN auto...	%M2001.1
"NEXT STEP"	%MW130
"CLOSE aut...	%M2001.3
"CLOSED se...	%M0.3
"CLOSE aut...	%M2001.3
"NEXT STEP"	%MW130
"UP auto r...	%M2001.0
"UP sensor"	%M0.0
"UP auto r...	%M2001.0
"NEXT STEP"	%MW130
"IN auto r...	%M2001.4
"IN sensor"	%M0.4
"IN auto r...	%M2001.4
"NEXT STEP"	%MW130
"DOWN auto...	%M2001.1
"DOWN sensor"	%M0.1
"DOWN auto...	%M2001.1
"NEXT STEP"	%MW130
"OPEN auto...	%M2001.2
"OPEN sensor"	%M0.2
"OPEN auto...	%M2001.2
"NEXT STEP"	%MW130
"UP auto r...	%M2001.0
"UP sensor"	%M0.0
"UP auto r...	%M2001.0
"NEXT STEP"	%MW130

89	END_IF;				
90	10: //move left and out				
91	"LEFT auto request" := 1;		"LEFT auto..."	%M2001.7	
92	"OUT auto request" := 1;		"OUT auto ..."	%M2001.5	
93	IF "LEFT sensor" = 1 AND "OUT sensor" = 1 THEN		"LEFT sensor"	%M0.7	
94	"LEFT auto request" := 0;		"LEFT auto..."	%M2001.7	
95	"OUT auto request" := 0;		"OUT auto ..."	%M2001.5	
96	"NEXT STEP" := 11;		"NEXT STEP"	%MW130	
97	END_IF;				
98	11: //move down				
99	"DOWN auto request" := 1;		"DOWN auto..."	%M2001.1	
100	IF "DOWN sensor" = 1 THEN		"DOWN sensor"	%M0.1	
101	"DOWN auto request" := 0;		"DOWN auto..."	%M2001.1	
102	"NEXT STEP" := 12;		"NEXT STEP"	%MW130	
103	END_IF;				
104	12: //gripper close				
105	"CLOSE auto request" := 1;		"CLOSE aut..."	%M2001.3	
106	IF "CLOSED sensor" = 1 THEN		"CLOSED se..."	%M0.3	
107	"CLOSE auto request" := 0;		"CLOSE aut..."	%M2001.3	
108	"NEXT STEP" := 13;		"NEXT STEP"	%MW130	
109	END_IF;				
110	13: //move up				
111	"UP auto request" := 1;		"UP auto r..."	%M2001.0	
112	IF "UP sensor" = 1 THEN		"UP sensor"	%M0.0	
113	"UP auto request" := 0;		"UP auto r..."	%M2001.0	
114	"NEXT STEP" := 14;		"NEXT STEP"	%MW130	
115	END_IF;				
116	14: //move right				
117	"RIGHT auto request" := 1;		"RIGHT aut..."	%M2001.6	
118	IF "RIGHT sensor" = 1 THEN		"RIGHT sen..."	%M0.6	
119	"RIGHT auto request" := 0;		"RIGHT aut..."	%M2001.6	
120	"NEXT STEP" := 15;		"NEXT STEP"	%MW130	
121	END_IF;				
122	15: //move down				
123	"DOWN auto request" := 1;		"DOWN auto..."	%M2001.1	
124	IF "DOWN sensor" = 1 THEN		"DOWN sensor"	%M0.1	
125	"DOWN auto request" := 0;		"DOWN auto..."	%M2001.1	
126	"NEXT STEP" := 16;		"NEXT STEP"	%MW130	
127	END_IF;				
128	16: //gripper open				
129	"OPEN auto request" := 1;		"OPEN auto..."	%M2001.2	
130	IF "OPEN sensor" = 1 THEN		"OPEN sensor"	%M0.2	
131	"OPEN auto request" := 0;		"OPEN auto..."	%M2001.2	
132	"NEXT STEP" := 17;		"NEXT STEP"	%MW130	

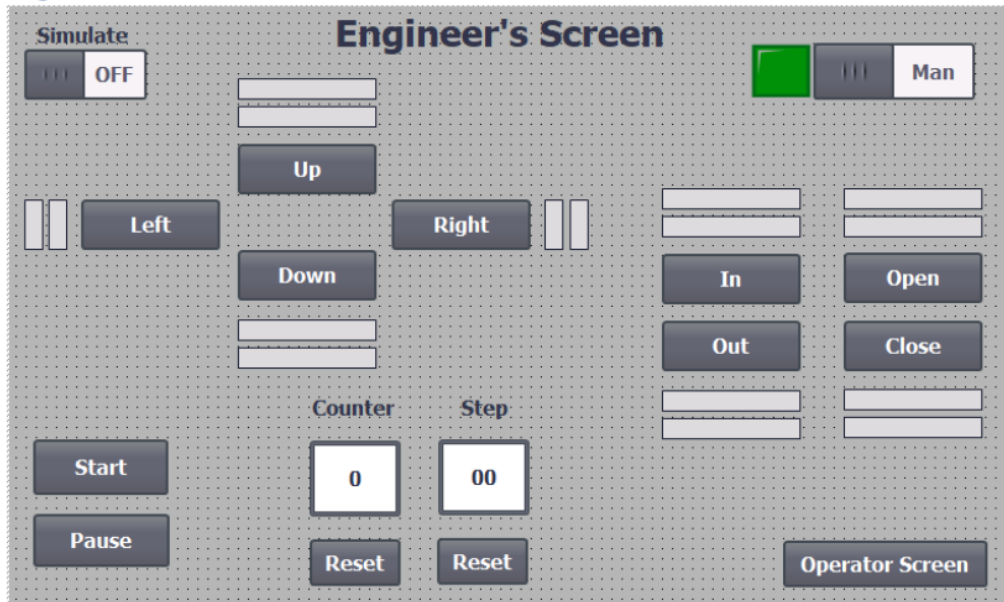
```

133     END_IF;
134 17: //move up
135     "UP auto request" := 1;
136 IF "UP sensor" = 1 THEN
137     "UP auto request" := 0;
138     "NEXT STEP" := 18;
139 END_IF;
140 18: //move object from station 1 to staion 2
141 //
142 "LEFT auto request" := 1;
143 IF "LEFT sensor" = 1 THEN
144     "LEFT auto request" := 0;
145     "NEXT STEP" := 19;
146 END_IF;
147 19: //move in
148 "IN auto request" := 1;
149 IF "IN sensor" = 1 THEN
150     "IN auto request" := 0;
151     "NEXT STEP" := 20;
152 END_IF;
153 20: //move down
154 "DOWN auto request" := 1;
155 IF "DOWN sensor" = 1 THEN
156     "DOWN auto request" := 0;
157     "NEXT STEP" := 21;
158 END_IF;
159 21: //gripper close
160 "CLOSE auto request" := 1;
161 IF "CLOSED sensor" = 1 THEN
162     "CLOSE auto request" := 0;
163     "NEXT STEP" := 22;
164 END_IF;
165 22: //move up
166 "UP auto request" := 1;
167 IF "UP sensor" = 1 THEN
168     "UP auto request" := 0;
169     "NEXT STEP" := 23;
170 END_IF;
171 23: //move out
172 "OUT auto request" := 1;
173 IF "OUT sensor" = 1 THEN
174     "OUT auto request" := 0;
175     "NEXT STEP" := 24;
176 END_IF;
177 24: //move down
178 "DOWN auto request" := 1;
179 IF "DOWN sensor" = 1 THEN
180     "DOWN auto request" := 0;
181     "NEXT STEP" := 25;
182 END_IF;
183 25: //gripper open
184 "OPEN auto request" := 1;
185 IF "OPEN sensor" = 1 THEN
186     "OPEN auto request" := 0;
187     "NEXT STEP" := 0;
188 END_IF;
189 END_CASE;
190 "STEP" := "NEXT STEP";
191

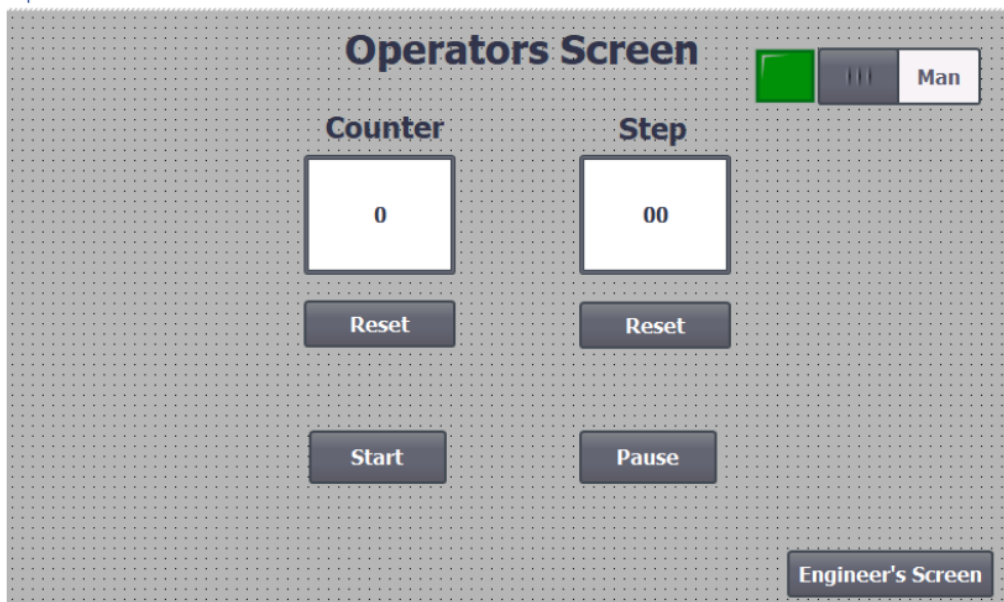
```

	"UP auto r...	%M2001.0
	"UP sensor"	%M0.0
	"UP auto r...	%M2001.0
	"NEXT STEP"	%MW130
	"LEFT auto...	%M2001.7
	"LEFT sensor"	%M0.7
	"LEFT auto...	%M2001.7
	"NEXT STEP"	%MW130
	"IN auto r...	%M2001.4
	"IN sensor"	%M0.4
	"IN auto r...	%M2001.4
	"NEXT STEP"	%MW130
	"DOWN auto...	%M2001.1
	"DOWN sensor"	%M0.1
	"DOWN auto...	%M2001.1
	"NEXT STEP"	%MW130
	"CLOSE aut...	%M2001.3
	"CLOSED se...	%M0.3
	"CLOSE aut...	%M2001.3
	"NEXT STEP"	%MW130
	"UP auto r...	%M2001.0
	"UP sensor"	%M0.0
	"UP auto r...	%M2001.0
	"NEXT STEP"	%MW130
	"OUT auto ...	%M2001.5
	"OUT sensor"	%M0.5
	"OUT auto ...	%M2001.5
	"NEXT STEP"	%MW130
	"DOWN auto...	%M2001.1
	"DOWN sensor"	%M0.1
	"DOWN auto...	%M2001.1
	"NEXT STEP"	%MW130
	"OPEN auto...	%M2001.2
	"OPEN sensor"	%M0.2
	"OPEN auto...	%M2001.2
	"NEXT STEP"	%MW130
	"STEP"	%MW120

Engineer's Screen



Operators screen



S7-1200 Data Sheet

Product overview

1.1 Introducing the S7-1200 PLC

Table 1- 1 Comparing the CPU models

Feature		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C
Physical size (mm)		90 x 100 x 75	90 x 100 x 75	110 x 100 x 75	130 x 100 x 75
User memory	Work	30 Kbytes	50 Kbytes	75 Kbytes	100 Kbytes
	Load	1 Mbyte	1 Mbyte	4 Mbytes	4 Mbytes
	Retentive	10 Kbytes	10 Kbytes	10 Kbytes	10 Kbytes
Local on-board I/O	Digital	6 inputs/4 outputs	8 inputs/6 outputs	14 inputs/10 outputs	14 inputs/10 outputs
	Analog	2 inputs	2 inputs	2 inputs	2 inputs / 2 outputs
Process image size	Inputs (I)	1024 bytes	1024 bytes	1024 bytes	1024 bytes
	Outputs (Q)	1024 bytes	1024 bytes	1024 bytes	1024 bytes
Bit memory (M)		4096 bytes	4096 bytes	8192 bytes	8192 bytes
Signal module (SM) expansion		None	2	8	8
Signal board (SB), Battery board (BB), or communication board (CB)		1	1	1	1
Communication module (CM) (left-side expansion)		3	3	3	3
High-speed counters	Total	3 built-in I/O, 5 with SB	4 built-in I/O, 6 with SB	6	6
	Single phase	3 at 100 kHz	3 at 100 kHz	3 at 100 kHz	3 at 100 kHz
		SB: 2 at 30 kHz	1 at 30 kHz SB: 2 at 30 kHz	3 at 30 kHz	3 at 30 kHz
	Quadrature phase	3 at 80 kHz SB: 2 at 20 kHz	3 at 80 kHz 1 at 20 kHz SB: 2 at 20 kHz	3 at 80 kHz 3 at 20 kHz	3 at 80 kHz 3 at 20 kHz
Pulse outputs ¹		4	4	4	4
Memory card		SIMATIC Memory card (optional)			
Real time clock retention time		20 days, typ. / 12 day min. at 40 degrees C (maintenance-free Super Capacitor)			
PROFINET		1 Ethernet communication port			2 Ethernet communication ports
Real math execution speed		2.3 µs/instruction			
Boolean execution speed		0.08 µs/instruction			

Feature		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Physical size (mm)		90 x 100 x 75		110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
User memory	Work	50 Kbytes	75 Kbytes	100 Kbytes	125 Kbytes	150 Kbytes
	Load	1 Mbyte	2 Mbytes	4 Mbytes		
	Retentive	10 Kbytes				
Local onboard I/O	Digital	6 inputs/ 4 outputs	8 inputs/ 6 outputs	14 inputs/ 10 outputs		
	Analog	2 inputs			2 inputs/2 outputs	
Process image size	Inputs (I)	1024 bytes				
	Outputs (Q)	1024 bytes				
Bit memory (M)		4096 bytes			8192 bytes	
Signal module (SM) expansion		None	2	8		
Signal board (SB), Battery board (BB), or communication board (CB)		1				
Communication module (CM) (left-side expansion)		3				
High-speed counters	Total	Up to 6 configured to use any built-in or SB inputs				
	1 MHz	-				1b.2 to 1b.5

Feature		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
	100/180 kHz	Ia.0 to Ia.5				
	30/120 kHz	--	Ia.6 to Ia.7	Ia.6 to Ib.5	Ia.6 to Ib.1	
	200 kHz ³					
Pulse outputs ²	Total	Up to 4 configured to use any built-in or SB outputs				
	1 MHz	--				Qa.0 to Qa.3
	100 kHz	Qa.0 to Qa.3				Qa.4 to Qb.1
	20 kHz	--	Qa.4 to Qa.5	Qa.4 to Qb.1	--	
Memory card		SIMATIC memory card (optional)				
Data logs	Number	Maximum 8 open at one time				
	Size	500 MB per data log or as limited by maximum available load memory				
Real time clock retention time		20 days, typ./12 day min. at 40 degrees C (maintenance-free Super Capacitor)				
PROFINET Ethernet communication port		1			2	
Real math execution speed		2.3 µs/instruction				
Boolean execution speed		0.08 µs/instruction				

S7-1500 Data Sheet

Supply voltage, permissible range of all the CPUs 19.2 V DC ... 28.8 V DC						
Code work memory						
Standard CPU:	150 KB	300 KB	500 KB	1 MB	2 MB	4 MB
T-CPU	225 KB	---	750 KB	---	3 MB	---
F-CPU	225 KB	450 KB	750 KB	1.5 MB	3 MB	6 MB
Data work memory	1 MB	1.5 MB	3 MB	5 MB	8 MB	20 MB
Processing times						
Bit operations	0.06 µs	0.04 µs	0.03 µs	0.01 µs	0.002 µs	0.001 µs
Word operations	0.072 µs	0.048 µs	0.036 µs	0.012 µs	0.003 µs	0.002 µs
Interfaces						
PROFINET IO	1	1	1	1	1	1
PROFINET	---	---	1	1	1	2
Number of PROFINET ports	2	2	3	3	3	4

CPU	1511-1 PN 1511F-1 PN 1511T-1 PN	1513-1 PN 1513F-1 PN	1515-2 PN 1515F-2 PN 1515T-2 PN	1516- 3 PN/DP 1516F- 3 PN/DP	1517- 3 PN/DP 1517F- 3 PN/DP 1517T- 3 PN/DP 1517TF- 3 PN/DP	1518- 4 PN/DP 1518F- 4 PN/DP 1518-4 PN/DP ODK 1518F-4 PN/DP ODK
PROFIBUS DP	---	---	---	1	1	1
Technology						
Motion Control resources	800	800	2400	2400	10240	10240
Isochronous mode	X	X	X	X	X	X
Web server	X	X	X	X	X	X

Compact CPUs

Controller	1511C-1 PN	1512C-1 PN
Article number	6ES7511-1CK00-0AB0	6ES7512-1CK00-0AB0
Manual		
Supply voltage, permissible range	19.2 V DC to 28.8 V DC	19.2 V DC to 28.8 V DC
Code work memory	175 KB	250 KB
Data work memory	1 MB	1 MB
Processing time for bit operations	60 ns	48 ns
PROFINET interfaces	1	1
Number of PROFINET ports	2	2
Integrated analog inputs/outputs	5 inputs/2 outputs	5 inputs/2 outputs
Integrated digital inputs/outputs	16 inputs/16 outputs	32 inputs/32 outputs
Technology		
Motion Control resources	800	800
Isochronous mode	X	X
High-speed counters	6	6
Pulse generators (pulse width modulation, Pulse Train Output, frequency output)	4 (PTOx/PWMx)	4 (PTOx/PWMx)
Web server	X	X

FINAL GRADE

GENERAL COMMENTS

Instructor**58**/100

PAGE 1

PAGE 2

Text Comment. Sections should be numbered according to the briefPAGE 3

Text Comment. In future justify your text as full with 1.5 line spacing**Text Comment.** A good introduction**Text Comment.** This is an unnecessary statement as you should be using the Harvard method for citing sources.

QM

Citation Needed

Cite Source:

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

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

QM

Citation Needed

Cite Source:

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

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

QM

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.htmlPAGE 4

QM

Incorrectly cited, this is not the Harvard Method

QM

Citation Needed

Cite Source:

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

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

QM

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 5

QM

Incorrectly cited, this is not the Harvard Method

QM

Citation Needed

Cite Source:

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

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

QM

Citation Needed

Cite Source:

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

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

QM

Spelling

Spelling error

QM

Good conclusions

You have shown understanding here from these observations and deductions

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

Text Comment. There are no analogue inputs in this system.

PAGE 6

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

Text Comment. Where is your IO list?

PAGE 7

Text Comment. incorrect symbol for proximity switch

PAGE 8

Text Comment. It is unnecessary to instruct your reader how to set the IP address

PAGE 9

Text Comment. Write in the third person and concentrate on explaining the system.

Text Comment. Should be "the program contains" rather than "must contain" as this is stated in the brief

PAGE 10

Text Comment. all figures should be numbered or simply placed in the appendix with a title.

PAGE 11

PAGE 12

Text Comment. wrong comment

Text Comment. all figures should be numbered or simply placed in the appendix with a title.

PAGE 13

Text Comment. why have you included the up sensor?

PAGE 14

Text Comment. Object Ready is missing

Text Comment. all figures should be numbered or simply placed in the appendix with a title.

PAGE 15

Text Comment. You should have used the Object Ready tag here instead of the Start PB

PAGE 16

PAGE 17

PAGE 18

Text Comment. These screen belong in the appendix

PAGE 19

PAGE 20

Text Comment. all figures should be numbered or simply placed in the appendix with a title.

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

Text Comment. good comment

PAGE 31

Text Comment. THIS IS NOT A CITATION LIST



Wrong method of listing, this is not Harvard

Web sources should have date accessed.

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

GRADING FORM: NEW CSG-RUBRIC

JASON BOOTH

58

1. ENGAGEMENT WITH LITERATURE



This section contains good information and is generally well written and interesting. However, you have not cited nor listed your sources correctly using

8

the Harvard method. There is good evidence of research from a small range of sources. (8/20)

2. KNOWLEDGE AND UNDERSTANDING

 Knowledge is generally good with clear evidence of understanding with some minor misconceptions e.g. the use of the start PB instead of the Object Ready tag in the sequence step 1. (14/20) **14**

3. COGNITIVE AND INTELLECTUAL

 Good evidence of analysis through testing with evidence supplied as screen dumps with a useful conclusion at the end. (12/20) **12**

4. PRACTICAL APPLICATION

 Program is well designed and documented internally with good presentation of results. (14/20) **14**

5. TRANSFERABLE SKILLS

 A well structure and mostly well written report. However, you have not numbered your sections according to the brief and it would help your reader if the text was fully justified with 1.5 line spacing. Word count limit exceeded, -5 marks penalty applied. All figures in the main report should be numbered or simply placed in the appendix with a title. Most of the images belong in the appendix. (10/20) **10**

