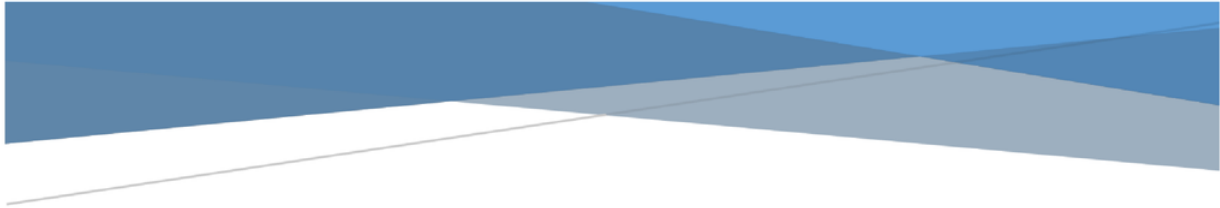


PSDE4005 Assignment 1 - Jordan Gough

by Jordan Gough

FILE	15327_JORDAN_GOUGH_PSDE4005_ASSIGNMENT_1_-_JORDAN_GOUGH_154267_1353706789.DOCX (1.64M)		
TIME SUBMITTED	03-DEC-2017 03:47PM (UTC+0000)	WORD COUNT	4322
SUBMISSION ID	78849073	CHARACTER COUNT	28211



PSDE4005

Programmable Logic Controllers

Assignment 1

03/12/2017

Jordan Gough

4333 Words

Table of Contents

Table of Figures.....	1
1. Research.....	3
2. Hardware Description and I/O List.....	8
3. Program Explanation.....	13
4. System Testing and Evaluation	19
5. Conclusions and Recommendations	25
6. Collaboration.....	26
7. Works Cited.....	27

Table of Figures

Figure 1	3
Figure 2	3
Figure 3	4
Figure 4	5
Figure 5	5
Figure 6	6
Figure 7	6
Figure 8	7
Figure 9	9
Figure 10	10
Figure 11	10
Figure 12	10
Figure 13	11
Figure 14	11
Figure 15	11
Figure 16	12
Figure 17	13
Figure 18	13
Figure 19	13
Figure 20	14
Figure 21	14
Figure 22	15
Figure 23	15
Figure 24	15
Figure 25	16
Figure 26	16
Figure 27	17
Figure 28	17
Figure 29	18
Figure 30	18
Figure 31	23
Figure 32	23

Figure 33	23
Figure 34	23
Figure 35	24

1. Research

Types of PLC

There are three different types of PLC that all have unique applications, due to their differing style of operation. These are unitary, modular and rack mounted.

Unitary – Unitary PLCs are stand-alone units that contain all the PLC's components, such as memory and processor, and have a limited number of inputs and outputs. By holding all the components inside the PLC, it becomes compact and easily mountable on DIN rail. (Crispin, 1997)

An example of a unitary PLC available on the market is a Siemens SIMATIC S7-1200, shown in Figure 1, which can be purchased for £218. Due to the limited number of I/O, these PLCs can be used for small automation uses, for example an automatic barrier. Some unitary PLC may have expansion modules, but for specific applications, such as communication or sensors which would not work otherwise. (Siemens, 2016)



Figure 1
(RS Components, n.d.)

Modular – Modular PLCs have one main CPU which holds the memory, processor and communications modules. The main CPU can then be expanded using different modules to add I/O. These modules connect directly into the adjacent modules to allow the overall system to become more useful for applications that are too large for a unitary PLC. The modules available include I/O modules and communication modules. In some cases, the base CPU does have I/O, so expansion modules will not have to be used if there are enough available. Most Modular PLCs are priced. (Crispin, 1997)

An example of a modular PLC available on the market is Siemens' SIMATIC S7-1500, shown in Figure 2, which can be purchased as part of a starter kit for over £1200. A modular PLC can be used for applications that require more I/O than a unitary PLC will offer, such as running large industrial equipment, for example a boiler. (Siemens, 2017)



Figure 2

(RS Components, n.d.)

Rack Mounted – Rack mounted PLCs consist of modules which plug into a powered back board. This board is powered through a single module, providing power to the other modules that get connected to it. The modules are also all connected onto the same network, so they can communicate with each other. The modules can be added and removed when required. (Crispin, 1997)

An example of a rack mounted PLC available on the market is Siemens' SIMATIC S7-400, shown in Figure 3. The cost of this PLC can be between £1000 and £12000, depending on the component being purchased. Although more expensive than unitary and modular PLCs, they are much more powerful and can have extremely large number of I/O, which can be operated very quickly. Due to the large number of I/O and the power of the system, rack mounted PLCs can be used for large scale applications, such as running an entire unit in an industrial environment. Some rack mounted PLCs can also be used in conjunction with computers, which allow the user to monitor the PLC while it runs. (Siemens, 2013) (Siemens, 2015)



Figure 3
(Siemens, 2015)

Types of IEC Programming Language

There are five different programming languages available for developing PLC programs that are recognised by the IEC61131-3. These languages are Sequential Function Chart (SFC), Ladder Diagram (LD), Structured Text (ST), Instruction List (IL), and Function Block Diagram (FBD). (Rexroth, 2009)

Ladder Diagram – Ladder diagram logic, known as LAD by Siemens, is the main programming language used for small PLC automation applications. An example of ladder logic is shown in Figure 4. Due to the ladder logic's use of symbols, it has become a universally recognised language which can be read and used worldwide. The IEC standard of ladder logic requires the use of set symbols, such as open and closed contacts and output coils. Using these symbols allows the logic to be converted to function block and instruction list if required. An application example of ladder logic is a sorting system that requires outputs to trigger when inputs become true. For example, a conveyor system which pushes a box off the belt when it is detected by a sensor. (Rexroth, 2009) (Crispin, 1997)



```

1: module rtl (
2:     input clk,
3:     input reset,
4:     output [3:0] q
5: );
6:
7:     process (clk)
8:     begin
9:         if (reset == 1)
10:             q <= 0;
11:         else
12:             q <= q + 1;
13:         end if;
14:     end process;
15:
16:     wait_time <= 0;
17:     wait_time_exp <= 0;
18:
19:     process (clk)
20:     begin
21:         if (reset == 1)
22:             wait_time <= 0;
23:             wait_time_exp <= 0;
24:         else
25:             wait_time <= wait_time + 1;
26:             wait_time_exp <= wait_time_exp + 1;
27:         end if;
28:     end process;
29:
30:     q <= q + 1;
31:
32: end module

```

Figure 5
(Rexroth, 2009)

5



Figure 6
(Rexroth, 2009)

Structured Text – Structured text, known as structured control language by Siemens, is another text-based language. An example of structured text is shown in Figure 7. This language is based off the idea that IF <function> THEN. This means that the functions in the logic will only work under the right conditions. The language is easier to develop if there is a lack of knowledge in circuit wiring, as it becomes easier to follow, given that it is written using words. (Rexroth, 2009) (Crispin, 1997)

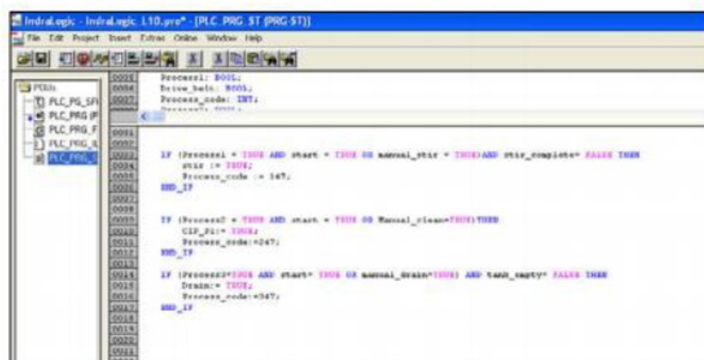
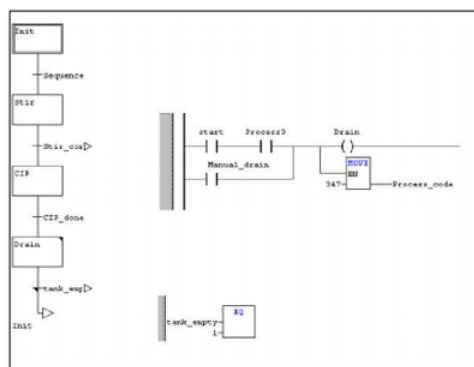


Figure 7
(Rexroth, 2009)

Sequential Function Chart – Sequential function chart (SFC), known as GRAPH by Siemens, utilises a flowchart like structure to automate and control a system. An example of SFC is shown in Figure 8. The flowchart is made up of many blocks, which all contain logic written using any language chosen by the developer. The logic inside the blocks is active until the program moves onto the next step. One application example of SFC is an automated arm that picks up and moves objects to a new location. Like function block diagrams, SFC follows a specific order of operation due to the transition steps between each function. (Rexroth, 2009) (Crispin, 1997)



2. Hardware Description and I/O List

When developing a PLC program, it is vital to be aware of the hardware on each input and output. To maintain the correct tags throughout the development, tables of inputs and outputs were generated.

Inputs	Tag Name	Hardware Description
I0.0	Assembly Area Sensor See Figure 11	Detects the presence of a plastic ring in the assembly area.
I0.1	Sort Area Sensor See Figure 10	Detects the presence of an object inside the sort area.
I0.2	Reject Area Sensor See Figure 15	Detects an object in the reject area, where it will either be rejected by the Reject Solenoid or accepted.
I0.3	Capacitive Area Sensor See Figure 14	Detects the presence of completed assemblies on the belt conveyor, prior to reaching the reject area.
I0.4	Metal Peg Detector See Figure 10	Detects metal pegs on the Chain Conveyor.
I0.5	Infra-red Through Beam Sensor See Figure 12	Detects the presence of objects on the belt conveyor when infra-red beam is broken.
I0.6	Capacitive Sensor See Figure 14	Detects the presence of the plastic rings on the belt conveyor.
I0.7	Inductive Sensor See Figure 12	Detects metal pegs on the belt conveyor.

Outputs	Tag Name	Description
Q0.0	Sort Solenoid See Figure 10	Upon detecting an object in the Sort Area, the solenoid will push the object into the assembly chute. Will not operate when a metal peg is detected.
Q0.1	Rotary Solenoid See Figure 11	The Rotary Solenoid will release the plastic rings from the assembly chute into the assembly area. Will activate after a delay

		of 2 seconds, if the assembly area is empty.
Q0.2	Reject Solenoid See Figure 15	The Reject Solenoid will remove any objects that are not completed assemblies from the belt conveyor.
Q0.3	Chain Conveyor See Figure 9	The Chain Conveyor carries the objects towards the sort area. No inputs used, allowing the output to run once powered.
Q0.4	Belt Conveyor See Figure 9	The Belt Conveyor carries the objects towards the reject area. No inputs used, allowing the output to run once powered.
Q0.5	N/A	Output not used.
Q0.6	N/A	Output not used.
Q0.7	N/A	Output not used.

The conveyor system uses a Siemens S7-1200 PLC, shown in Figure 16, which is an appropriate PLC to use as the number of I/O is sufficient to operate the system. A Siemens HMI is also used to assist with monitoring and maintaining the equipment.



Figure 9



Figure 10



Figure 11

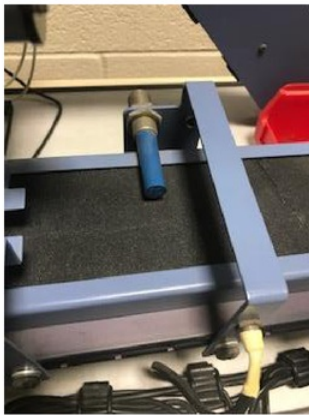


Figure 12



Figure 13



Figure 14



Figure 15



Figure 16

3. Program Explanation

Figure 17 shows the program development software. The program was developed using TIA Portal, which is Siemens' software that is used to design and build PLC programs for S7-1200 PLCs. The software also allows the user to design the Human Machine Interfaces, or HMIs.

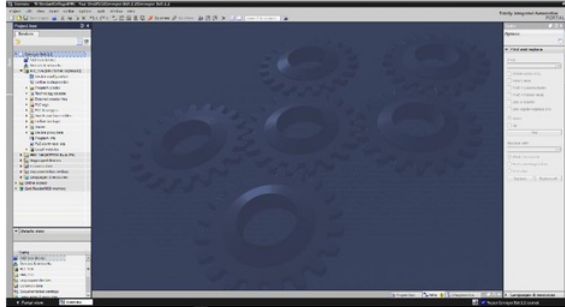


Figure 17

Figure 18 shows the list of I/O tags used to link the inputs and outputs of the PLC to the hardware. There is a total of eight inputs, five outputs and three memory bits. The inputs are addressed %I0.0 to %I0.7, the outputs are %Q0.0 to %Q0.4 and the memory bits are %M0.2.

	Name	Tag table	Data type	Address
1	Assembly Area Sensor	Default tag table	Bool	%I0.0
2	Sort Area sensor	Default tag table	Bool	%I0.1
3	Reject Area Sensor	Default tag table	Bool	%I0.2
4	Capacitive Area Sensor	Default tag table	Bool	%I0.3
5	Metal Peg Detector	Default tag table	Bool	%I0.4
6	Infra-red Detector	Default tag table	Bool	%I0.5
7	Capacitive Sensor	Default tag table	Bool	%I0.6
8	Inductive Sensor	Default tag table	Bool	%I0.7
9	Sort Solenoid	Default tag table	Bool	%Q0.0
10	Rotary Solenoid	Default tag table	Bool	%Q0.1
11	Reject Solenoid	Default tag table	Bool	%Q0.2
12	Chain Conveyor	Default tag table	Bool	%Q0.3
13	Belt Conveyor	Default tag table	Bool	%Q0.4
14	Metal Peg on Belt Conveyor	Default tag table	Bool	%M0.0
15	Assembled Product on Belt	Default tag table	Bool	%M0.1
16	Reject Area Sensor Pulse	Default tag table	Bool	%M0.2
17	<Add new>			

Figure 18

Figure 19 shows Network 1, where there are two rungs with an output on each. The first output is %Q0.4, which is tagged to the Belt Conveyor, and the second is %Q0.3, which is tagged to the Chain Conveyor. There are no inputs on these rungs, meaning that the conveyors will begin to run as soon as the system is powered.

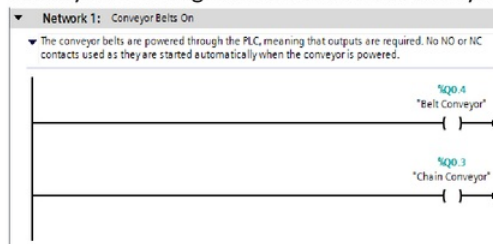


Figure 19

Figure 20 shows Network 2, which has the logic for the Sort Solenoid. A normally open (NO) contact closes when an object is detected by the Sort Area Sensor. The normally closed (NC) Queue Counter contact will open if the counter reaches its set point, thus breaking the signal and preventing operation. The Metal Peg Detector contact also prevents operation. An on timer is used to operate the solenoid for a total of 0.5s. The Sort Solenoid is latched, to hold the signal onto the solenoid until after the time has elapsed. The network then resets.

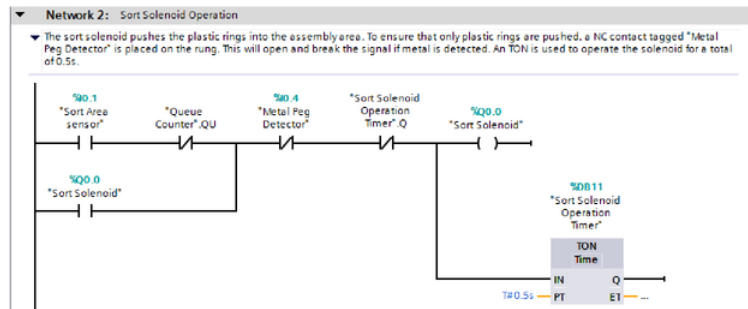


Figure 20

Figure 21 shows Network 3, which contains the up-down counter that monitors the number of plastic rings in the assembly chute. A NO contact will close when the Sort Solenoid operates, thus counting up by one. A NO contact will close when the Rotary Solenoid operates, counting down by one.

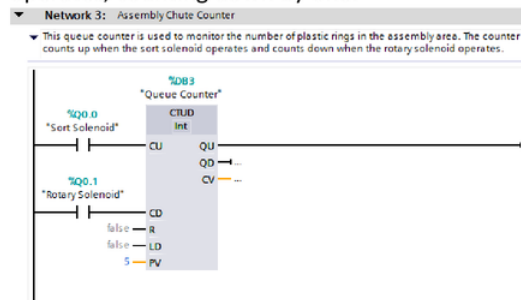


Figure 21

Figure 22 shows Network 4, which is used to operate the Rotary Solenoid. The NC Assembly Area Sensor contact will prevent the solenoid from operating if there is a plastic ring in the assembly area. This prevents the chute from getting blocked. There is also a greater than or equal to contact, which will only operate the solenoid if there are one or more plastic rings in the assembly chute. To ensure that the solenoid allows the assembly area time to empty, an on delay of 2s is used to delay the operation. The output is tagged to a NO contact to latch the signal.

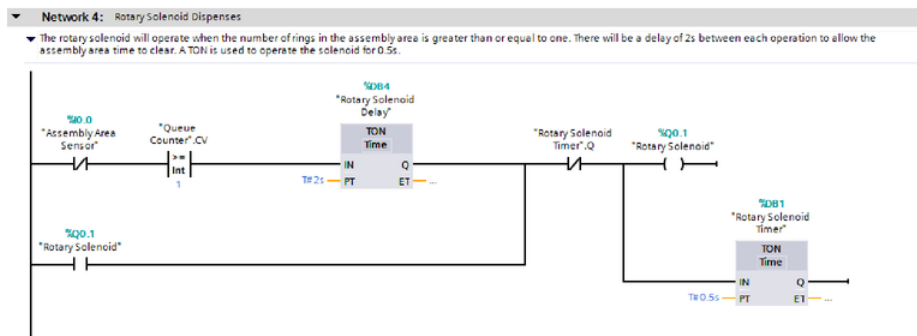


Figure 22

Figure 23 shows Network 5, has two rungs that use set output coils. The first rung contains one of the memory bits, “Metal Peg on Belt Conveyor”. When the Inductive Sensor detects a metal peg on the conveyor, the output will set a bit and keep it set until it is reset. This bit is tagged to a NO contact on the second rung, along with the Capacitive Area Sensor and Capacitive Sensor, to set another memory bit. The memory bit on the second rung is “Assembled Product on Belt”.

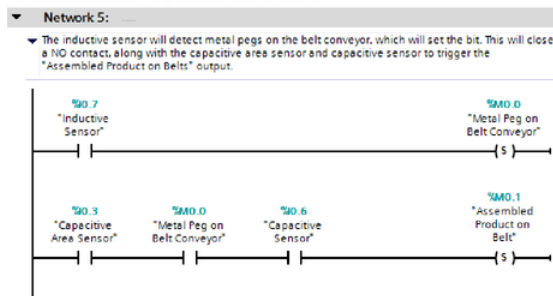


Figure 23

Figure 24 shows Network 6, which is used to operate the Reject Solenoid. A NO contact is linked to the Reject Area Sensor, which will close when an object is detected in the area. There is also a NC contact linked to the set bit “Assembled Product on Belt”. If an assembled product is detected, the contact will open, thus preventing the Reject Solenoid’s operation. Again, the Reject Solenoid output is latched.

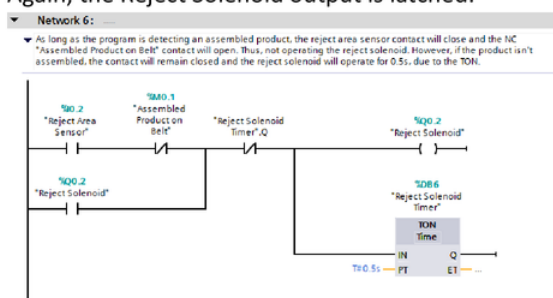


Figure 24

Figure 25 shows Network 7, which is used to reset the set bits, using reset output coils. A negative edge NO contact is tagged to the Reject Area Sensor and to another memory bit, "Reject Area Sensor Pulse". When the reject sensor pulse reaches its negative edge, the contact will close, thus resetting the set bits.

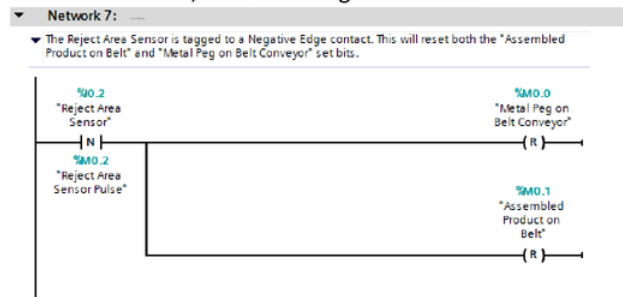


Figure 25

Figure 26 shows Network 8, which has been added for the operator's HMI screen and is used to count the number of completed assemblies. This is done by using an up counter that counts up by one every time the Reject Area Sensor detects an object and the Reject Solenoid does not operate. If the Reject Solenoid does operate, the NC contact will open, breaking the signal.

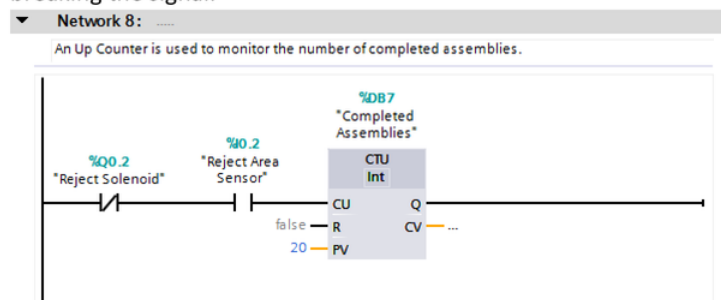


Figure 26

Figure 27 shows Network 9 which is also used to add a counter to the operator's HMI screen. Every time the Reject Solenoid operates, the NO contact will close, allowing the counter to count up by one.

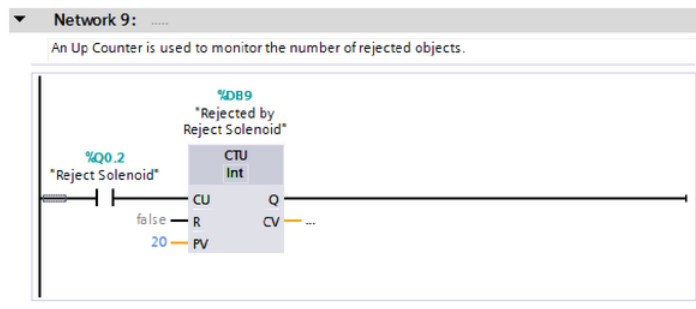


Figure 27

Figure 28 shows Network 10, which has a counter that counts the number of plastic pegs that are not sent into the assembly chute, due to there being 5 plastic rings in the chute. If the sort area sensor detects an object, the NO contact will close. As long as the object isn't metal and the Sort Solenoid does not operate, the counter will count up by one. However, if the solenoid does operate or a metal peg is detected, the NC contact will open, breaking the signal.

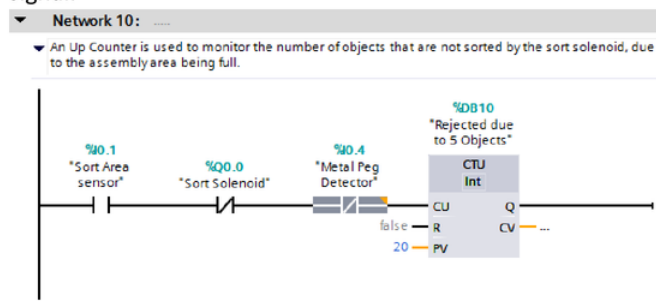


Figure 28

Figure 29 shows the HMI's operator screen. This screen contains four counters and two indicators. The indicators show the running status of the Belt Conveyor and the Chain Conveyor. The first counter monitors the number of plastic rings in the assembly chute, on Network 3. The second counter monitors the number of completed assemblies, found on Network 8. The third shows the number of objects rejected by the Reject Solenoid, found on Network 9. The fourth counter is the number of items rejected by the Sort Solenoid, which is found on Network 10.

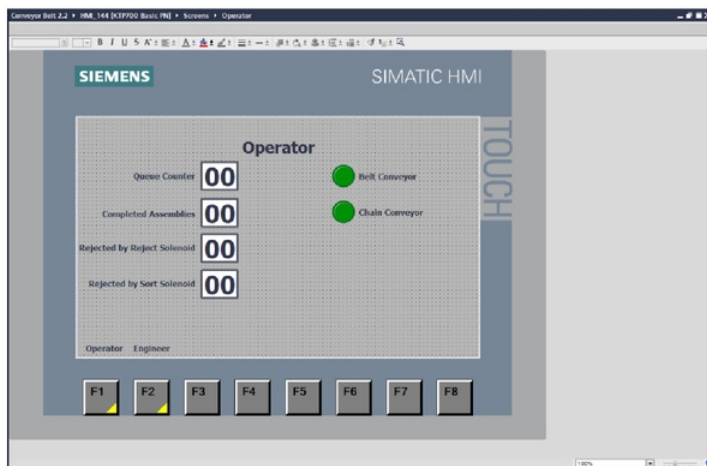


Figure 29

Figure 30 shows the engineer's screen. The engineer's screen has a counter, which can be altered to simulate a value in the assembly chute. This is used to test Network 2. There are also timer displays for the Sort Solenoid operation, Rotary Solenoid operation, Rotary Solenoid delay and Reject Solenoid operation. The green indicators are in place for the engineer to monitor the inputs when they are activated. There are also three buttons on the screen to manually test the outputs. By pushing the buttons, the engineer can either test the Sort Solenoid, the Rotary Solenoid or the Reject Solenoid.

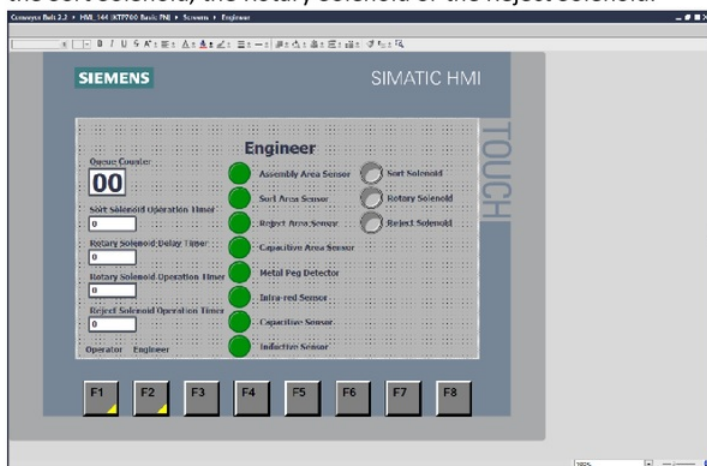


Figure 30

4. System Testing and Evaluation

Conveyor Testing Table

Number	Description	Expected Result	Actual Result
1	Switch on.	Both Conveyors should start.	Both conveyors started running when powered.
2	Place plastic ring on Chain Conveyor.	Rings are pushed down assembly chute by Sort Solenoid.	Chain Conveyor carried plastic ring into sort area, where Sort Solenoid pushed into assembly chute.
3	Metal peg on Chain Conveyor. Figure 31	The metal pegs pass by and do not get pushed down the chute.	Chain Conveyor carried metal peg through sort area and down feeder chute.
4	Put 7 plastic rings on Chain Conveyor.	Only 5 should be pushed down the assembly chute and 2 should pass by.	Chain Conveyor carried the two additional plastic rings through the sort area, without the Sort Solenoid operating.
5	Rotary Solenoid releases plastic ring.	Rotary Solenoid should drop plastic ring into the Assembly Area.	Plastic ring dropped out of the assembly chute and into the assembly area.
6	Metal peg passes through assembly area. Figure 32	Plastic ring should drop onto metal peg.	Metal peg picked up plastic ring as it travelled through assembly area.
7	Assembly passes Inductive Sensor.	Inductive Sensor detects metal peg.	Inductive Sensor LED lit up to prove that input activated.

8	Assembly passes Capacitive Sensor. Figure 33	Capacitive Sensor detects completed assembly.	Capacitive Sensor LED lit up to prove that input activated.
9	Metal Peg passes through sensors without plastic ring. Figure 35	The Capacitive Sensor should detect that there is no plastic ring, triggering the Reject Solenoid.	Metal peg detected by sensor and rejected by the Reject Solenoid.
10	Plastic ring passes through sensors without metal peg. Figure 34	Plastic ring should be rejected by Reject Solenoid.	Plastic ring rejected by Reject Solenoid.
11	Assembled product passes through upside down.	Upside down product should be rejected by Reject Solenoid.	Product rejected.

HMI Testing Table

12.	Queue Counter reaches 5.	Sort Solenoid ignores plastic rings.	Plastic rings passed by Sort Solenoid.
13.	Completed assemblies.	The number of completed assemblies is counted.	HMI counter successfully counted the number of completed assemblies.
14.	Rejected by Solenoid.	This function counts how many objects are rejected by the Reject Solenoid.	HMI counter successfully counted the number of objects rejected by the Reject Solenoid.
15.	Rejected by Sort Solenoid.	When assembly chute is full the Sort Solenoid counter on the HMI Operator display screen starts	HMI correctly counted the number of plastic rings ignored by the Sort Solenoid.

		counting how many plastic rings pass by.	
16.	Chain Conveyor indicator.	When the Chain Conveyor is running, the indicator on the HMI's Operator screen lights up.	HMI indicator changed state when Chain Conveyor was running.
17.	Belt Conveyor indicator.	When the Belt Conveyor is running, the indicator on the HMI's Operator screen lights up.	HMI indicator switched state when Belt Conveyor was running.
18.	Assembly Area Sensor indicator.	Assembly Area Sensor indicator comes on when Assembly Area Sensor detects object.	Assembly Area Sensor indicator on HMI switched when plastic ring entered assembly area.
19.	Sort Area Sensor indicator.	Sort Area Sensor indicator comes on when Sort Area Sensor detects object.	Sort Area Sensor indicator on HMI changed when object passed Sort Area Sensor.
20.	Reject Area Sensor indicator.	Reject Area Sensor indicator comes on when Reject area sensor detects object.	Reject Area Sensor indicator on HMI switched when object passed Reject Area Sensor.
21.	Capacitive Area Sensor indicator.	Capacitive Area Sensor indicator comes on when triggered.	Capacitive Area Sensor indicator on HMI changed when assembly passed by the Capacitive Area Sensor.
22.	Metal Peg Detector indicator.	Metal Peg Detector indicator comes on when triggered.	Metal Peg Detector indicator on HMI changed when metal peg passed detector.
23.	Infra-red Sensor indicator.	Infra-red Sensor indicator comes on when triggered.	HMI indicator changed when infra-red beam was broken.

24.	Capacitive Sensor indicator.	Capacitive Sensor indicator comes on when triggered.	Capacitive Sensor indicator on HMI changed when assembly passed by the Capacitive Sensor.
25.	Inductive Sensor indicator.	Inductive Sensor indicator comes on when triggered.	HMI indicator switched when metal peg passed by the Inductive Sensor.
26.	Sort Solenoid operation timer.	Sort Solenoid operation timer counts correctly.	Timer on HMI did not correctly display the operation time of the Sort Solenoid.
27.	Rotary Solenoid delay timer.	Rotary Solenoid delay timer counts correctly.	Timer on HMI successfully showed 2s delay time between operations.
28.	Rotary Solenoid operation timer.	Rotary Solenoid operation timer counts correctly.	Timer on HMI did not correctly display the operation time of the Rotary Solenoid.
29.	Reject Solenoid operation timer.	Reject Solenoid operation timer counts correctly.	Timer on HMI did not correctly display the operation time of the Reject Solenoid.
30.	Sort Solenoid function test.	When the button on HMI is pressed, solenoid activates.	Sort Solenoid operated when the test button was pressed.
31.	Rotary Solenoid test button.	When the button on HMI is pressed Solenoid activates.	Rotary Solenoid operated when the test button was pressed.
32.	Reject Solenoid Test button.	When the button on HMI is pressed Solenoid activates.	Reject Solenoid operated when the test button was pressed.

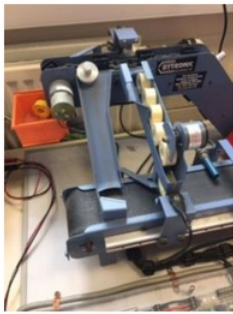


Figure 31



Figure 32



Figure 33

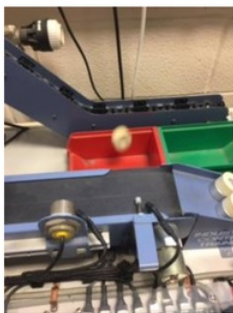


Figure 34



Figure 35

The system failed 3 of the 37 tests that were completed. However, the failed tests were for the timer displays on the HMI's operator screen and not for the program. This is due to the number of figures used in the timer not being able to correctly display 0.5 seconds. Otherwise, the testing of the program was successful, and all the networks worked as intended. The other functions found on the HMI screens, such as the solenoid function test, worked correctly.

5. Conclusions and Recommendations

The objective was to understand the different PLC types available, the IEC programming languages and design and build a PLC program that would operate a conveyor belt assembly line, with rejection elements incorporated, which were successfully completed. A testing table and an I/O list were built to ensure that the program was developed and tested correctly. The program was successfully built using Siemens' TIA Portal software, downloaded to the S7-1200 PLC controller and tested using the conveyor belt assembly and the Siemens HMI. The program operated successfully, with the only tests it failed relating to the HMI Testing. The program's networks contained contacts, timers and counters that were correctly tagged, to make the testing procedure easy to follow but also to ensure that the correct outputs operated when required. The HMI screens were designed with a user-friendly approach, with everything being suitably labelled, providing enough information for fault diagnosis or for standard operation.

In the program, the conveyor belts run continuously until the power is removed. There may be a situation in which the operator may need to stop the conveyors running, for example in case of emergency. A normally closed contact can be added to either both rungs, or on one rung, if one conveyor branches off the other. An emergency stop could also be hardwired into the system, rather than be added to the HMI. This will break the signal and stop the conveyors running. As seen in the testing results, the HMI's operation timers did not function correctly. This may be due to the timer only being for 0.5s. To improve this, the timer should be set to incorporate milliseconds, which can be used to provide an engineer with the information. Another major issue is the engineer's screen being freely accessed by anybody. The engineer's screen should be password protected, to prevent anybody without the correct permissions from accessing the functions.

6. Collaboration

As a group, we analysed the project brief and allocated tasks and responsibilities based on individual's strengths and competence to complete them. This was to ensure that we all contributed to the program design effectively. Although tasks were allocated to individuals, the group contributed by providing support to each other throughout. After completing each task, the group reconvened to evaluate progress and confirm that everybody understood how each network developed and how it operated.

Along with the PLC program, the group also worked to develop operator and engineer screens for the HMIs. The functions and displays present on the screens and how they were arranged was determined through group discussion.

Once the program and HMIs had been developed and downloaded, they needed testing. To monitor all aspects of the system, one person controlled the conveyor belt, a second person monitored the HMI and the third watched the program run on TIA Portal. This allowed full confirmation that all inputs and outputs were programmed correctly. The same steps were taken to test the engineer screen, with one person manually operating the system from the HMI, another watching the inputs and outputs operate on the conveyor and the final person watching each network on the computer, as required.

The group worked well together, all making effective contributions to the design, development and testing of the PLC program and HMI screens.

7. Works Cited

Crispin, A. J., 1997. *Programmable Logic Controllers and their Engineering Applications*. 2nd ed. Maidenhead: McGraw-Hill Publishing Company.

Rexroth, 2009. *Understanding the IEC61131-3 Programming Languages*. [Online]
Available at: https://www.automation.com/pdf_articles/IEC_Programming_Thayer_L.pdf
[Accessed 20 November 2017].

RS Components, n.d. *Siemens SIMATIC S7-1200 PLC CPU*. [Online]
Available at: <https://uk.rs-online.com/web/p/products/6683052/>
[Accessed 28 11 2017].

Siemens, 2013. *SIMATIC Controllers*. [Online]
Available at: http://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_simatic-controller_overview_en.pdf
[Accessed 26 11 2017].

Siemens, 2015. *SIMATIC S7-400 automation systems – always the right choice*. [Online]
Available at:
https://w3app.siemens.com/mcms/infocenter/dokumentencenter/as/InfocenterLanguagePacks/SI-MATIC_S7_400/SIMATIC_S7_400_EN.pdf
[Accessed 26 11 2017].

Siemens, 2016. *Basic Controller SIMATIC S7-1200*. [Online]
Available at: <https://c4b.gss.siemens.com/resources/images/articles/dffa-b10053-02-7600.pdf>
[Accessed 26 11 2017].

Siemens, 2017. *Advanced Controllers SIMATIC S7-1500 with ET 200MP*. [Online]
Available at: <https://www.click4business-supplies.com/resources/articles/dffa-b10140-02-7600.pdf>
[Accessed 26 11 2017].

PSDE4005 Assignment 1 - Jordan Gough

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

72/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

GRADING FORM: CSG-RUBIC

JORDAN GOUGH**72****RESEARCH-INFORMED LITERATURE**

Section 1: Your research - Good evidence of research for PLC types and languages. Further marks were available for more performance data about each PLC together with more details of applications for PLC and languages. (9/15) Section 8: Your use of Harvard for citing and listing - Good range of sources listed correctly. Further marks were available for a great range of sources and more accuracy in citation method. (3/5)

12**KNOWLEDGE AND UNDERSTANDING**

Section 2: Hardware Description and I/O list - A good explanation given of the function of each of the devices in the system together with the PLC supported by photos. Further marks were available

15

for more details of each device and the HMI screen.
(7/10) Section 3: Program Explanation – Detailed explanations given for each network and screen
(8/10)

ANALYSIS



Section 4: System Testing and Evaluation -
Thorough testing carried out with a helpful table and some photos with a brief evaluation. Further marks could have been awarded for linking the photos to each of the tests. (7/10) Section 5: Conclusions and Recommendations - Some good conclusions and recommendations given. However, there were other recommendations that could have been given to improve the perform of the system such as a tracker to allow more than one assembled product on the belt at a time. (8/10)

15

PRACTICAL APPLICATION AND DEP.



Section 3: Program Design and Network Comments
- Good program design with additional features for diagnostics and operator information and some detailed network comments. Further marks could have been awarded if network images weren't so small as not all comments cold be read. (8/10)
Section 3: HMI Screen Designs – Good screen designs. However, size of images meant that t was difficult to see all the detail. (6/10) Section 4: Presentation of Test Results - Well presented in table format with some good photos. (7/10)

21

SKILLS FOR PROFESSIONAL PRAC.

All sections - A well structured report with a helpful



list of figures, most well written and presented.

Further marks were available for larger images that could be seen. (4/5) Section 6: Team working - Detailed account given. (5/5)