

PSDE 4005 Assignment 1

by Callum Thornhill

FILE	15408_CALLUM_THORNHILL_PSDE_4005_ASSIGNMENT_1_154267_1489801934.DOCX (906.24K)		
TIME SUBMITTED	02-DEC-2017 03:38PM (UTC+0000)	WORD COUNT	2756
SUBMISSION ID	78822113	CHARACTER COUNT	16164

HNC Electrical and Electronic Engineering

PSDE 4005 PLC's

Assignment 1

Callum Thornhill

Date: 30/11/2017

Word Count: 2756

Contents

1. Research.....	3
2. Hardware Description	5
3. Input/Output List	7
4. System Testing and Evaluation	8
5. Programme Explanation	11
6. Conclusions and Recommendations	21
7. Collaboration.....	22
Works Cited.....	23

Figure 1, LAD Example	3
Figure 2, FBD Example	4
Figure 3, SCL Example	4
Figure 4, STL Example	4
Figure 5, GRAPH Example	4
Figure 6, Industrial Control Trainer	5
Figure 7, Rung 1	11
Figure 8, Rung 2	11
Figure 9, Rung 3	12
Figure 10, Rung 4	12
Figure 11, Rung 5	13
Figure 12, Rung 6	13
Figure 13, Rung 7	14
Figure 14, Rung 8	14
Figure 15, Rung 9	15
Figure 16, Rung 10	16
Figure 17, Rung 11	17
Figure 18, Rung 12	17
Figure 19, Tag address list.....	18
Figure 20, HMI Maintenance screen.....	19
Figure 21, HMI Operator screen	20

1. Research

Logic programmes are used with programmable logic controllers known as PLC's. In this software, the programmes manage inputs and outputs to operate in a way that the programmer chooses. Inputs can be either normally open or normally closed. The programming of contacts as normally open or normally closed depends on how they are required to operate in the logic program. (Csanyi, 2012). What this means is that an input can be normally open until a signal has been received which will close the contact or a normally closed contact will open when a signal is received.

There are three main types of Siemens plc hardware that are as follows: Unitary, Modular and Rack mounted. These three pieces of hardware have different features and costs which suit different applications.

A unitary model of plc is a standard piece that cannot be expanded, the hardware is limited to its defined number of inputs and outputs which may be a hindrance in certain applications. However, due to the simplicity of this model, the cost is kept low which may be ideal for applications with a low amount of I/O.

A modular model of plc has more I/O than a unitary model and can be expanded with extra modules alongside along the din rail. A modular piece is more costly than a unitary model but the ability to expand the I/O is good for expansive applications where additional I/O may be required in the future. The main module itself however may or may not contain any I/O, if the latter is the case then extra modules will be required for use.

A Rack mounted model of plc is the most expansive of the models. It is mounted on a rack and additional modules can be added to the rack itself. The amount of I/O means that very large amounts of I/O can be used.

Programming languages come in a variety of flavours. Three of these are known as LAD (ladder), FBD (Function block diagram), STL (Statement list), SCL (Structured control language) and GRAPH.

LAD is a basic use of I/O which is easily understandable which is beneficial for technicians to understand how the program operates. Example shown in Figure 1.

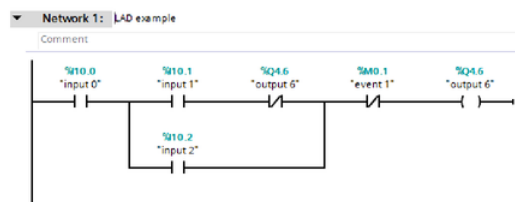


Figure 1, LAD Example

FBD looks similar to LAD but each function block is assigned a behaviour which allows complex routines to look tidier. Example shown in Figure 2.



Figure 2, FBD Example

SCL is a scripted language which is typed out, each I/O is used in the language to create the routines manually. Example shown in Figure 3.

```

1 "BASIC".C := "BASIC".A AND "BASIC".B;
2 "BASIC".D := "BASIC".A OR "BASIC".B;
3 "BASIC".E := NOT "BASIC".A;
4 "BASIC".F := NOT "BASIC".B;
5 "BASIC".G := NOT ("BASIC".A AND "BASIC".B);
6 "BASIC".H := NOT ("BASIC".A OR "BASIC".B);
7 "BASIC".I := ("BASIC".A AND (NOT "BASIC".B)) OR ((NOT "BASIC".A) AND "BASIC".B);
8 "BASIC".J := NOT "BASIC".I;
9

```

Figure 3, SCL Example

STL is similar to SCL although it is displayed differently because it is run sequentially which is easier to diagnose when something isn't working correctly. Example shown in Figure 4.

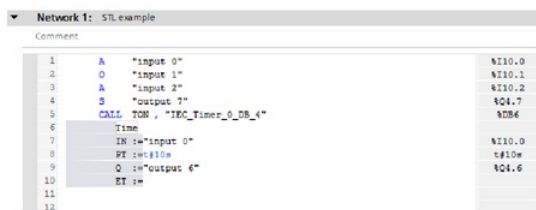


Figure 4, STL Example

GRAPH is a mixture between SCL and FBD, it is sequential but with boxes that have assigned behaviours. Example shown in Figure 5.

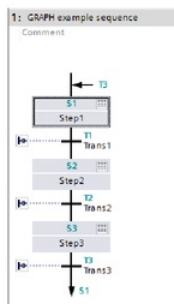


Figure 5, GRAPH Example

2. Hardware Description

The Industrial control trainer used can be seen in Figure 6.

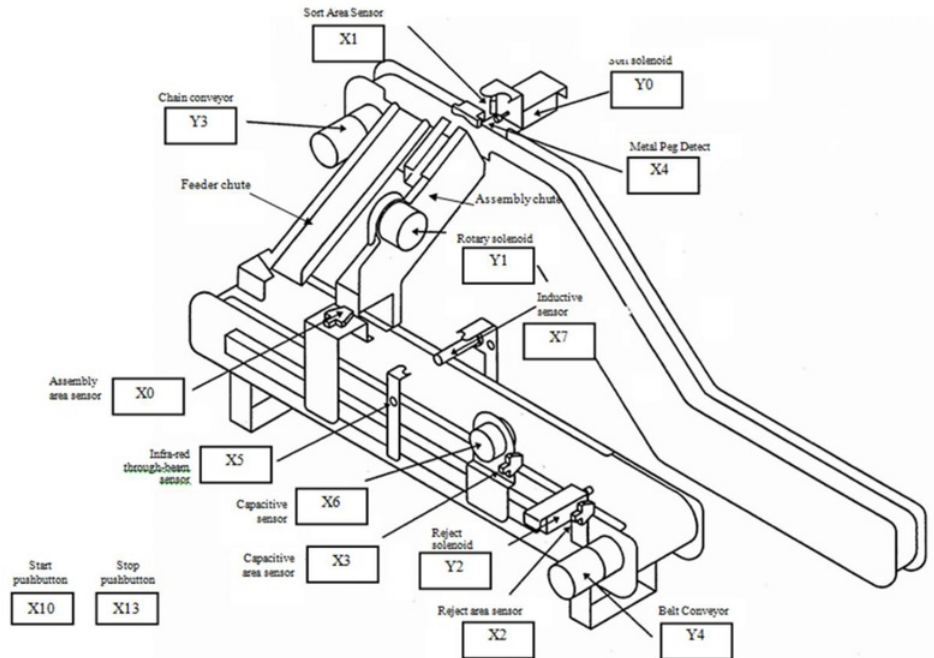


Figure 6, Industrial Control Trainer

The Conveyor system shown has 5 outputs and 8 inputs. 2 of the outputs are conveyor belts to transport objects around the machine. 2 of the outputs are solenoids used to push objects. 1 of the outputs is a rotary solenoid used to allow an object to pass it. The inputs are made up of area sensors and metal sensors. They are used to determine which of the objects is present.

When the machine is turned on, 2 outputs which are responsible for the conveyor belts are switched on to begin moving the objects up the first belt. Once at the top, the objects are determined to be either plastic or metallic using the two sensors. One of the sensors looks for metal while the other looks to see if there is an object there. If there is an object there and it is metallic, the solenoid won't activate, allowing the metal peg to pass. If the metal sensor doesn't detect anything but the sort area sensor does, the solenoid is activated which pushes the plastic peg down the assembly chute shown.

The rotary solenoid will allow one plastic peg to be dispensed to the assembly area and will only allow another one through when the assembly area is empty. When a metal peg passes the top section it slides down onto the bottom conveyor and passes the assembly area to pick up a plastic ring. The inductive sensor detects when a metal peg has passed the assembly area. The capacitive sensor and capacitive area sensor determine whether the product has been assembled correctly. If the product hasn't been assembled correctly the reject solenoid will activate which will push the

rejected product into a bin. If the product has been assembled properly it will be allowed to pass and the process will continue.

3. Input/Output List

The inputs and outputs used are shown below.

Inputs		Outputs	
Input 0.0	Assembly Area Sensor	Output 0.0	Sort Solenoid
Input 0.1	Sort Area Sensor	Output 0.1	Rotary Solenoid
Input 0.2	Reject Area Sensor	Output 0.2	Reject Solenoid
Input 0.3	Capacitive Area Sensor	Output 0.3	Chain Conveyor
Input 0.4	Metal Peg Detect	Output 0.4	Belt Conveyor
Input 0.5	Infra-red Through-beam Sensor		
Input 0.6	Capacitive sensor		

These inputs and outputs represent the numbers in Figure 6, for example, Y0 is output 0.0 and X0 is input 0.0. These inputs and outputs are used in the program in different ways to manage the product correctly.

4. System Testing and Evaluation

Number	Description	Expected Result	Actual Result	Successful?
#1	Use the on and off switches for the belt conveyors.	The belt conveyors should start when start is pressed and remain on until the stop button is pressed.	Both starting and stopping occurred when prompted.	Indeed.
#2	Move plastic ring into assembly chute.	The solenoid should push the plastic peg into the assembly chute.	The ring was pushed into the assembly chute.	Indeed.
#3	Sort a metal peg to the lower conveyor.	The metal peg should pass the solenoid without being pushed, allowing it to slide down onto the lower conveyor.	The metal peg passed the solenoid and slid down to the lower conveyor belt.	Indeed.
#4	To place 7 plastic rings onto the conveyor and see how many the machine allows to be pushed into the assembly chute.	The assembly chute should hold 5 plastic rings at a time and the solenoid shouldn't push more pegs into there if it's full.	4 plastic pegs were pushed into the assembly chute but 3 plastic pegs were allowed to pass the solenoid unhindered. Changes to the programme needed.	No, Changes will be made to the assembly chute counter.
#5	To dispense a single plastic ring from the assembly chute to the assembly area.	If there is at least 1 plastic ring in the assembly chute and no ring in the assembly area, the rotary solenoid should let one drop into there. When a ring is removed from the assembly area, another plastic ring should take its place.	A plastic ring was allowed to pass down into the assembly area from the chute. The ring was removed and the solenoid didn't open as there wasn't a ring in the chute. A ring was pushed into the chute from the conveyor and then the rotary solenoid allowed it to pass into the assembly area.	Indeed.
#6	To place a metal peg and plastic ring on the conveyor to test if the product will assemble correctly.	Knowing that the plastic ring is dispensed into the assembly area correctly, now the metal peg must take the ring from the assembly area.	The metal peg passed and picked up the plastic ring.	Indeed.

#7	Is the PLC recognising an assembled product?	A correctly assembled product should be allowed to pass by unhindered and the counter on the HMI should increase by 1 for the counter of assembled products.	The correctly assembled product made it to the end of the conveyor belt and the value of assembled products on the HMI was updated.	Indeed.
#8	Is there only a ring on the conveyor?	If there is only a ring on the conveyor it should be rejected and the counter for rejected ring only should be increased by 1.	The plastic ring was rejected by the solenoid and the counter on the HMI was updated.	Indeed.
#9	Is there only a peg on the conveyor?	If there is only a peg on the conveyor it should be rejected and the counter for rejected peg only should be increased by 1.	The metal peg was rejected by the solenoid, but the HMI counter didn't increase by 1.	No. The problem was fixed by changing a normally closed contact to a normally open contact.
#10	Is there an inverted product on the conveyor?	If there is an inverted product on the conveyor it should be rejected and the counter for rejected inverted products should be increased by 1.	The inverted product was rejected by the solenoid and the counter on the HMI was updated.	Indeed.
#11	Does the maintenance function on HMI work?	The solenoid outputs should be useable by pressing buttons on the maintenance screen of the HMI.	The solenoids were able to be used from the HMI.	Indeed.

5. Programme Explanation

The rung shown in Figure 7 is responsible for switching the program into test mode. This was added at the start of the program but wasn't used until further on. The purpose is to be able to activate the test mode which will allow the operation of outputs through the HMI.

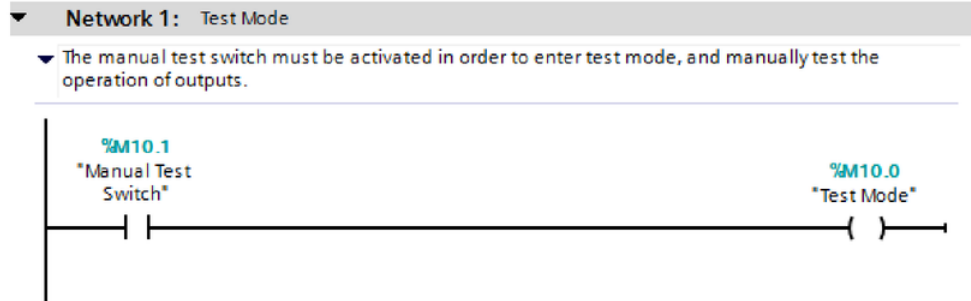


Figure 7, Rung 1

The rung shown in Figure 8 is used to enable the chain conveyor and belt conveyor which are essential to the movement of the plastic rings, metal pegs and assembled products on the line.

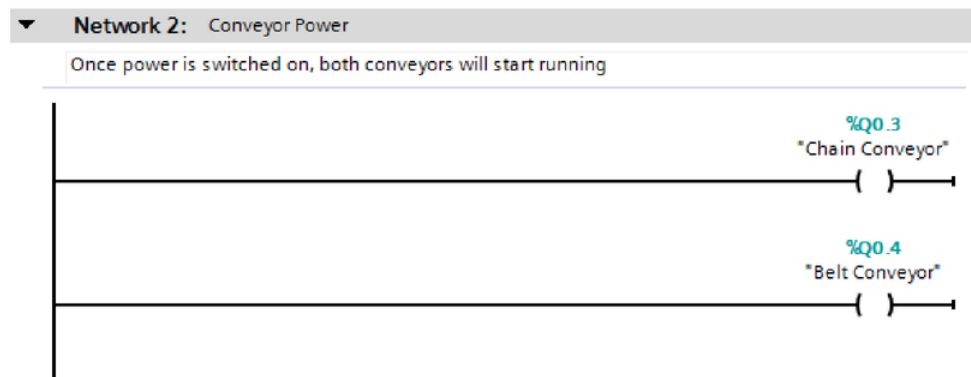


Figure 8, Rung 2

The rung shown in Figure 9 is responsible for separating the plastic rings and metal pegs. A latch and timer is used to keep the solenoid activated for enough time so that it will work.

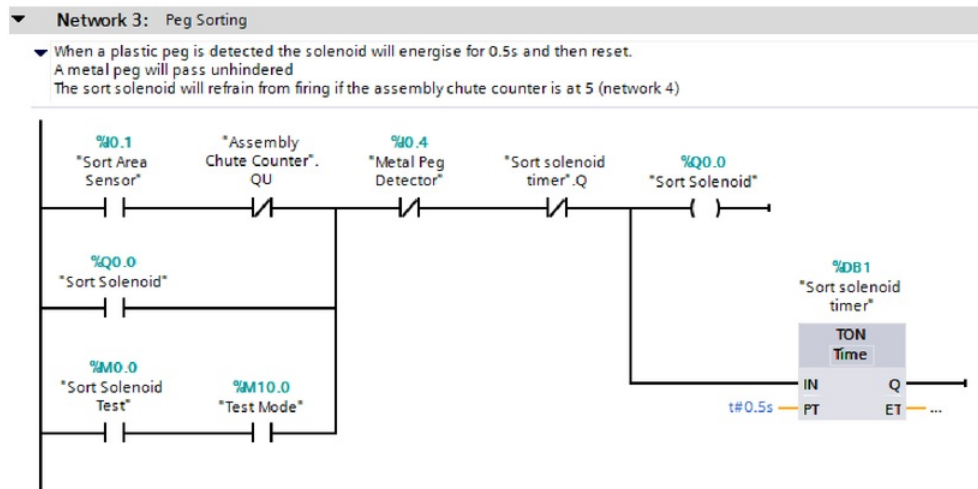


Figure 9, Rung 3

The rung shown in Figure 10 counts how many plastic rings are in the chute and is used later to allow or disallow the rotary solenoid from actuating.

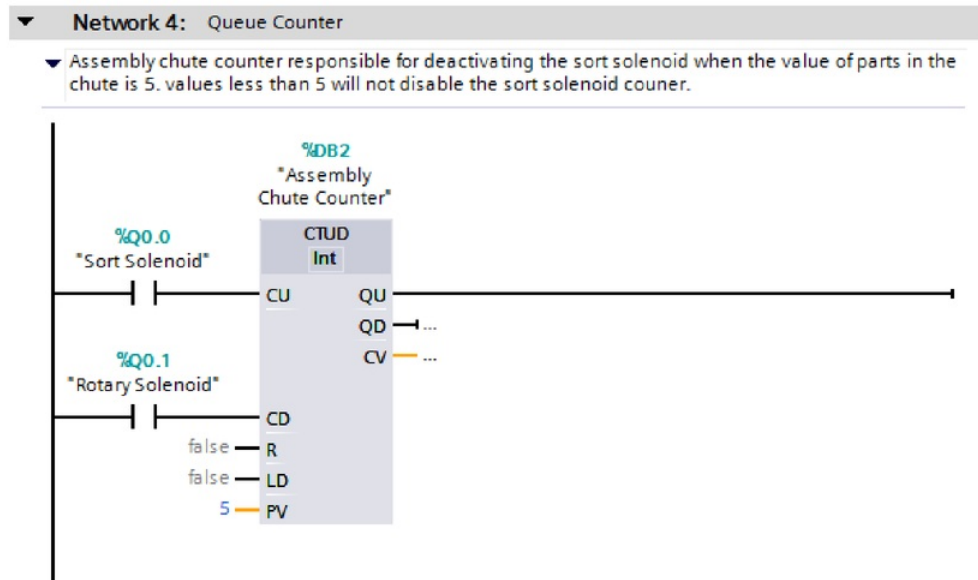


Figure 10, Rung 4

The rung shown in Figure 11 sets the behaviour of the rotary solenoid and when it is allowed to activate.

Network 5: Rotary Solenoid Action

- Once the ring has entered the chute, 2 seconds must elapse before the rotary solenoid can act. When there is at least one ring in the chute the rotary solenoid can be energised and the ring can be dispensed.

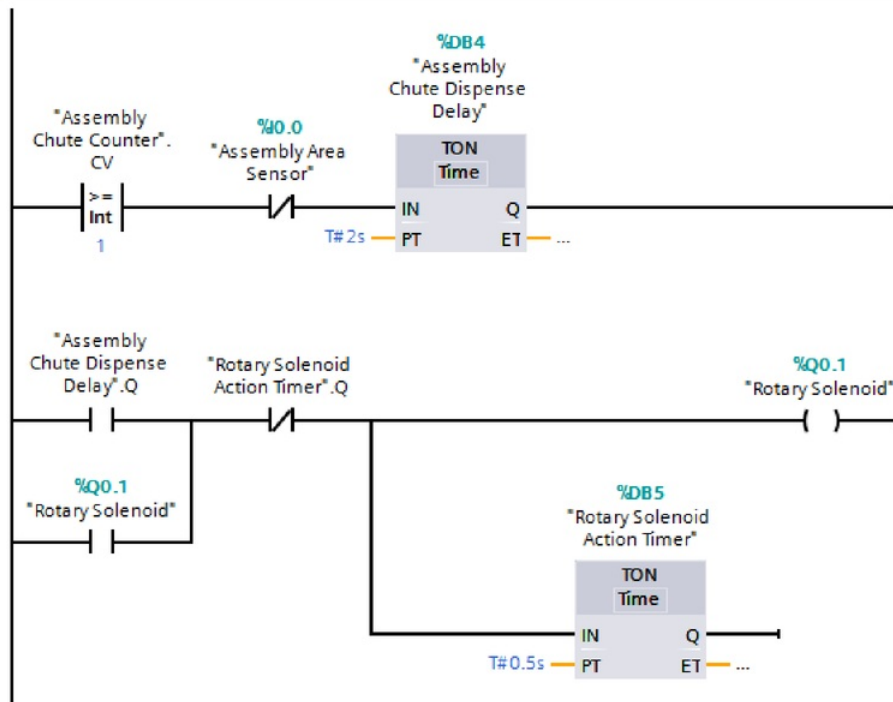


Figure 11, Rung 5

The rung shown in Figure 12 sets bits which can hold the set information to be used later in the program.

Network 6: Assembled product conditions

- When the necessary prerequisites are met for an assemble product, we set bits m30.0 and m30.1 so that they are latched until the product is either rejected or accepted.

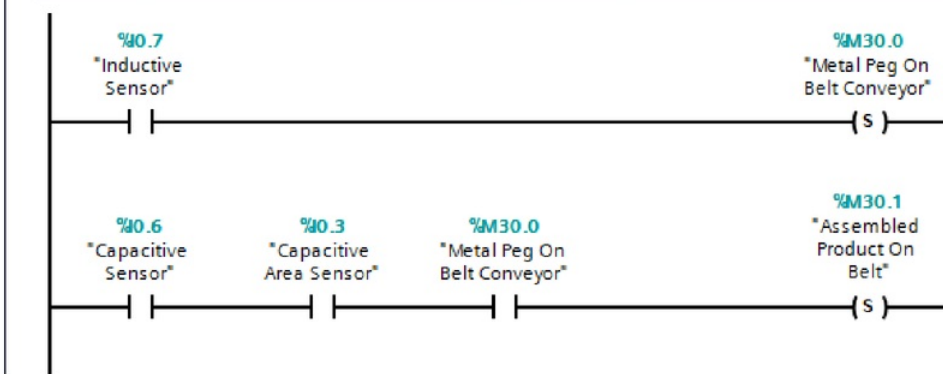


Figure 12, Rung 6

The rung shown in Figure 13 sets the behaviour for the rejection solenoid which includes the latching on of it for enough time for it to be effective.

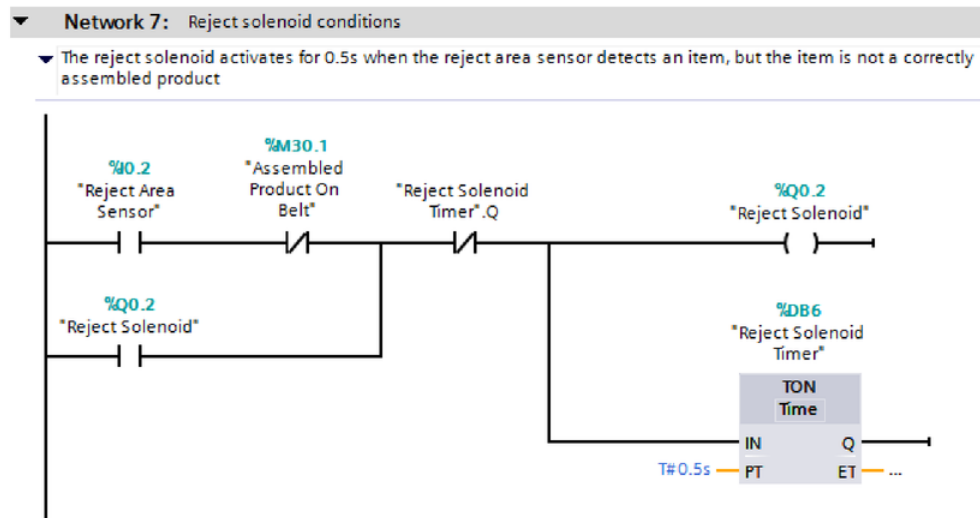


Figure 13, Rung 7

The rung shown in Figure 14 resets memory bits when something has reached the end of the belt whether it has been accepted or rejected.

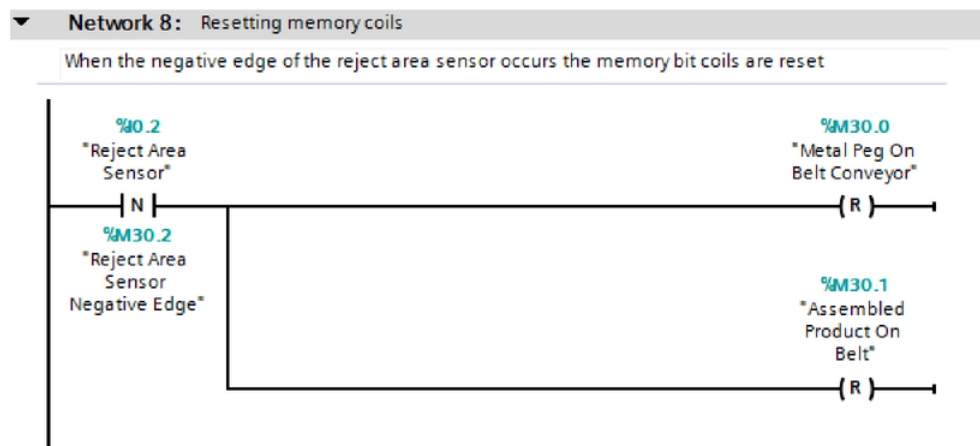


Figure 14, Rung 8

The rung shown in Figure 15 counts how many assembled products have been produced which can be seen on the HMI.

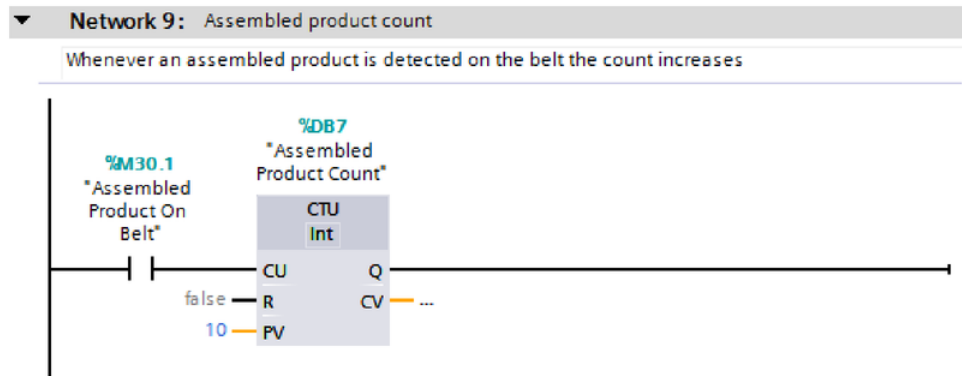


Figure 15, Rung 9

The rung shown in Figure 16 is responsible for counting which type of rejection has occurred and displaying the data on the HMI.

Network 10: Faulty Assemblies count

- These are the conditions necessary to activate the count for the 3 different types of incorrectly assembled product

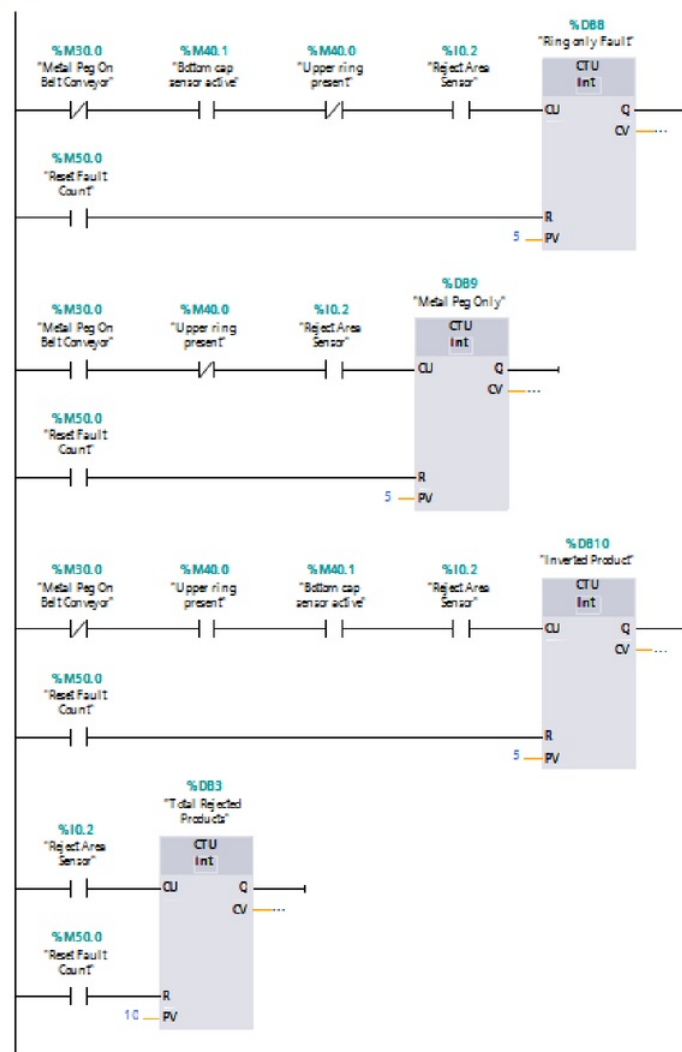


Figure 16, Rung 10

The rung shown in Figure 17 sets a bit which is related to counting for a plastic ring and resets it when something is rejected.

▼ **Network 11:** Upper ring present retain

- ▼ If an upper ring is detected by the capacitive sensor it sets the bit, and is reset when the reject solenoid operates.

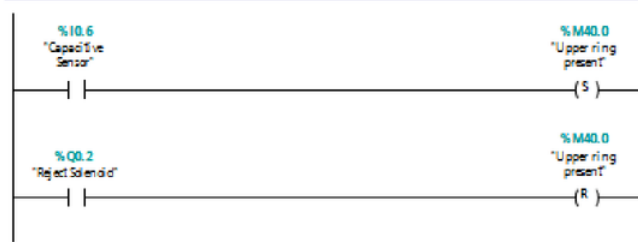


Figure 17, Rung 11

The rung shown in Figure 18 does the same as rung 11 but for the metal pegs.

▼ **Network 12:** Bottom Capacitive sensor active

- ▼ The capacitive area sensor sets the bit, and is reset when the reject solenoid fires.

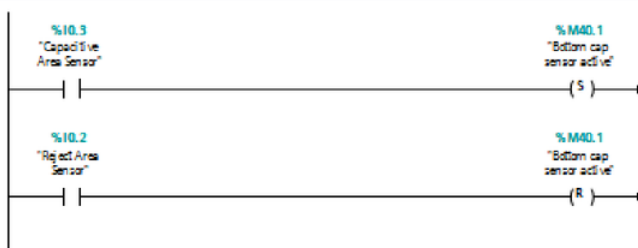


Figure 18, Rung 12

The tag address list for the program is shown in Figure 19. This list includes all the inputs, outputs and memory bits that are used.

PLC tags				
	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	IR Through-beam Sensor	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	Sort Solenoid Test	Default tag table	Bool	%M0.0
15	Test Mode	Default tag table	Bool	%M10.0
16	Manual Test Switch	Default tag table	Bool	%M10.1
17	Rotary Solenoid Test	Default tag table	Bool	%M0.1
18	Reject Solenoid Test	Default tag table	Bool	%M0.2
19	Peg/Ring collision Prevention	Default tag table	Bool	%M20.0
20	Metal Peg On Belt Conveyor	Default tag table	Bool	%M30.0
21	Assembled Product On Belt	Default tag table	Bool	%M30.1
22	Reject Area Sensor Negative Edg	Default tag table	Bool	%M30.2
23	Negative edge Assembled prod.	Default tag table	Bool	%M30.3
24	Upper ring present	Default tag table	Bool	%M40.0
25	Bottom cap sensor active	Default tag table	Bool	%M40.1
26	Reset Fault Count	Default tag table	Bool	%M50.0

Figure 19, Tag address list

The HMI screens that allow users to check progress and interact with the program can be seen in Figure 20 and Figure 21. The maintenance screen has functions which are as follows. The first rung in the program is switched on using the test mode button, this allows the user to test the three solenoids to see if they work, this is useful for testing when the line isn't running. The two circles labelled chain conveyor and belt conveyor light up when they are active to show the user that they are in operation. The reset count button will reset all counting values on the operator screen.

The operator screen has functions which are as follows. The chain and belt conveyor lights appear in the operator screen as well because the operator may need to know if they are active or not. A counter for the assembly chute shows the operator how many plastic rings are in the chute which can be used by an operator to keep an eye on the levels of product available. The total assembled products will show how many products have been successfully assembled. In the red box, there are counters for each type of rejected product and an overall counter for how many times it has occurred.

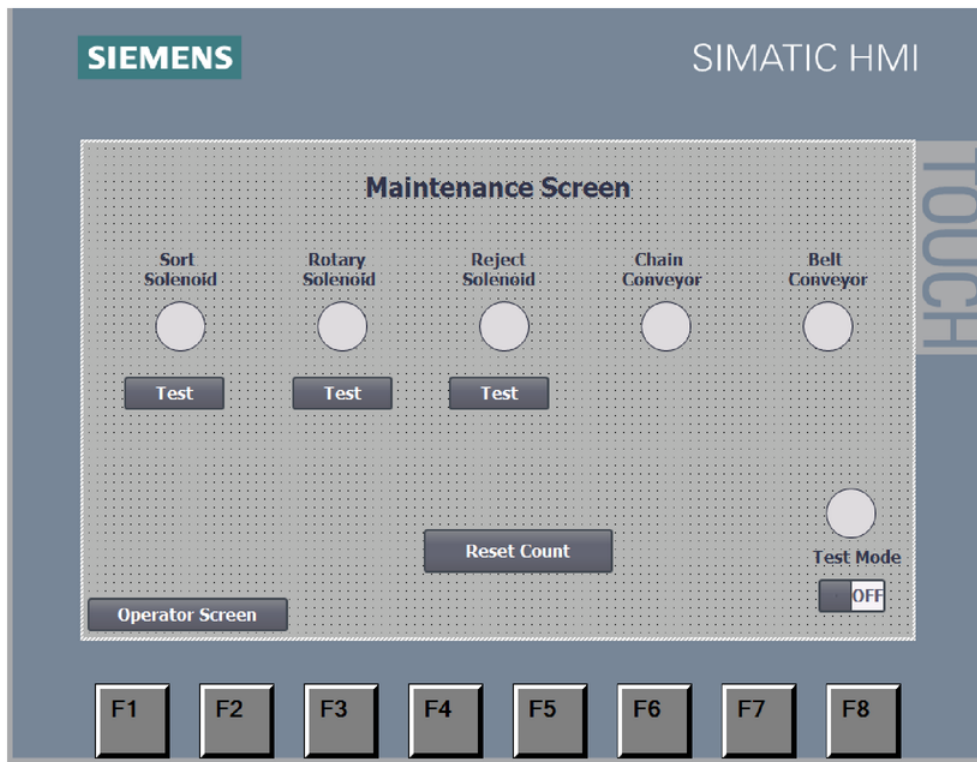


Figure 20, HMI Maintenance screen

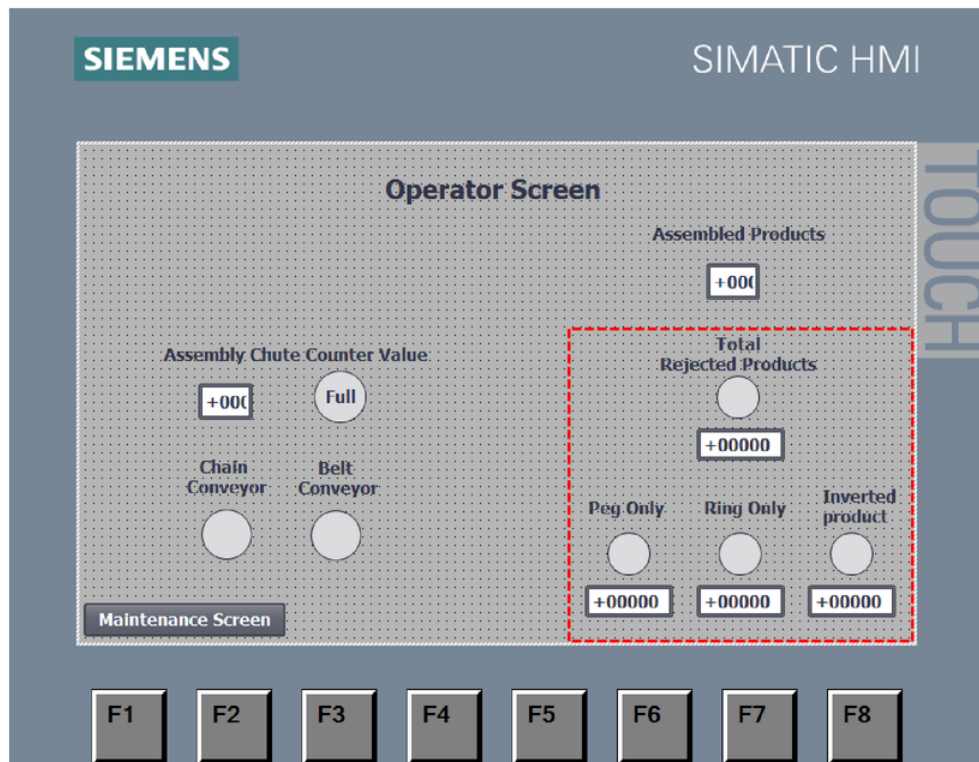


Figure 21, HMI Operator screen

6. Conclusions and Recommendations

Since being provided the brief, the project has progressed very well as part of a team and as an individual. The inputs and outputs have been worked together well to create a well working program which will operate as was requested. However, this machine and its program can be improved to increase performance and accuracy.

To increase accuracy, the program could be changed to allow multiple products to be on the lower belt at any time. Currently the program will set a bit when a metal peg has passed a certain point and then will reset the bit when the peg has either been rejected on its own or accepted as a complete product along with a plastic ring. This means that if two or more metal pegs are present on the belt then the memory bit will be reset before the second peg makes it to the end. This error won't reject the peg if it's a complete product but it will in fact limit the functionality of the counters that keep track of rejected products and assembled products.

To improve performance after the accuracy has been increased, the timer can be adjusted on the rotary solenoid to exactly the right amount of time that would be needed to dispense a plastic ring on arrival of a metal peg. This can be done by timing how long it takes a metal peg to pass the sensors at the top and slide down onto the lower conveyor. Having a tighter tolerance of time here may reduce the amount of metal pegs that pass the assembly point without picking up a plastic ring.

7. Collaboration

As part of the team, I had taken turns with my teammates to design parts of the program and aspects of the HMI. As the project progressed, ideas were provided by myself for the program and HMI, some of them were implemented permanently and others were later decided to not be necessary. As probably one of the most essential parts of the programming is knowing the inputs and outputs, I was responsible for the naming and insertion of the inputs and outputs into the program. The HMI had been changed quite a lot by each of us as time went on which allowed us all to have a go at suggesting changes to the design.

Works Cited

Csanyi, E., 2012. *Basic Steps In PLC Programming For Beginners*. [Online]
Available at: <http://electrical-engineering-portal.com/author/edvard>
[Accessed 20 11 2017].

PSDE 4005 Assignment 1

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

60/100

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

CALLUM THORNHILL**60****RESEARCH-INFORMED LITERATURE**

Section 1: Your research - Evidence of background research for PLC types and programming languages. Further marks were available for a more detailed explanation that included manufacturer's part numbers, performance data, costs and applications together with a more accurate explanation of languages and their applications. (7/15) Section 8: Your use of Harvard for citing and listing - One source cited and listed. Further marks would have been awarded had you consulted a greater range of sources and cited them in the text. (1/5)

8**KNOWLEDGE AND UNDERSTANDING**

Section 2: Hardware Description and I/O list -A brief I/O list given and a description of the operation of the machine. However, marks were available for an explanation of the devices in the system and actual I/O addresses used in the Siemens PLC together with photos and a description of the actual PLC and HMI used. (3/10) Section 3: Program Explanation – Good explanations given for program networks and

11

a good tag list which includes the I/O addresses together with an explanation of the HMI screens created. (8/10)

ANALYSIS

 Section 4: System Testing and Evaluation - A detailed table of result show thorough testing. Further marks were available for photographic evidence linked to the tests together with an overall evaluation. (6/10) Section 5: Conclusions and Recommendations - Some good comments made. (8/10)

14

PRACTICAL APPLICATION AND DEP.

 Section 3: Program Design and Network Comments - Excellent program design, with good additional networks for maintenance and fault finding, and suitable in-program comments. (10/10) Section 3: HMI Screen Designs – Good screen design. Further marks were available for additional diagnostic features. (7/10) Section 4: Presentation of Test Results - Well presented in table form. Further marks were available for photographic evidence linked to each test. (5/10)

22

SKILLS FOR PROFESSIONAL PRAC.

 All sections -A good structure with clear images listed at the front. Further marks were available for a more accurate structure according to the brief. (3/5) Section 6: Team working - Briefly explained. (2/5)

5