

PSDE4005 Assignment 1

by Daniel Ince

FILE	14401_DANIEL_INCE_PSDE4005_ASSIGNMENT_1_154267_1022610719.DOCX (1.46M)		
TIME SUBMITTED	03-DEC-2017 12:16PM (UTC+0000)	WORD COUNT	3152
SUBMISSION ID	78840208	CHARACTER COUNT	20987

ASSIGNMENT 1

Programmable Logic Controllers

03/12/2017

Word Count: 3257

Daniel Ince
HNC Electrical & Electronic Engineering

Contents

Task 1 – Research.....	3
Unitary	3
Modular.....	4
Rack Mounted.....	5
Languages	6
Task 2 – Hardware Description	9
Task 3 – Input/Output.....	10
Task 5 – System Testing and Evaluation	12
Task 6 – Program Explanation.....	13
Labwork 1 – Sorting Routine.....	13
Labwork 2 – Queue Counting.....	14
Labwork 3 – Dispensing and Operation Timing	15
Labwork 4 – Plastic Ring Detection and Sorting	15
Labwork 5 – Metal Component Detection and Sorting	16
Labwork 6 – Component Assembly Detection and Sorting	19
Task 7 – Conclusions and Recommendations	20
Task 8 – Collaboration.....	21
Works Cited.....	22

Report should be structured according to the brief and not by task number since the marking scheme is linked to section numbers.

Figure 1:

<https://www.999mitsubishi.com/pub/media/catalog/product/cache/image/700x560/e9c3970ab036de70892d86c6d221abfe/a/l/al2-24.jpg>.....

Figure 2: http://2.bp.blogspot.com/-CQ3MhI4oJwo/Ucqt6OA4XCI/AAAAAAAAA3E/Uo8gOrwjHe8/s1600/0001_Compact_Logix.png

Figure 3: <https://www.schneider-electric.com/ww/en/Images/tricon-IC-490x280.jpg>

Figure 4: <http://www.plcademy.com/wp-content/uploads/2017/06/ladder-logic-tutorial-rungs.gif>

Figure 5: <http://www.electronics-micros.com/img/arduino/plc-lib-alarm2.jpg>

Figure 6: <https://gojs.net/latest/assets/images/screenshots/sequentialFunction.png>

Figure 7: http://www.ronbeaufort.com/CLX_Structured_Text.PNG

Figure 8: <http://engineeronadisk.com/V3/engineeronadisk-3073.gif>

Figure 9: PLC tags

Figure 10: Testing Table

Figure 11: Belt start up.....

Figure 12: Sort operation

Figure 13: Queue Counter.....

avoid putting URLs into figure titles

Figure 14: Engineer HMI Screen	14
Figure 15: Dispense delay and operation	15
Figure 16: Memory bit set and reset	16
Figure 17: Reject solenoid operation and count.....	17
Figure 18: Assembled product memory bit set	17
Figure 19: Solenoid Testing Screen	18
Figure 20: HMI manufacturing screen	18
Figure 21:Reject counter and reset	19
Figure 22: Plastic reject counter and reset	19

good range of figures listed

Task 1 – Research

this is section 1 not task 1

A large variety of programmable logic controllers are available to the consumer from many different manufacturers. The performance of each type of PLC varies and depends on the design and brand. The main function of a PLC is to act as a controller for a system with many inputs and outputs, this allows large boards of relay logic and wiring to be replaced by a much smaller PLC. An added benefit of the unit is that they offer increased reliability and versatility with the ability for the program to be altered quickly if necessary without the need to alter any physical hardwiring.

As stated in *Microprocessors and Interfacing*, "Microcomputer-based process-control systems range from a small programmable controller... which might be used to control a machine on a factory floor, to a large minicomputer used to control an entire fractionating column in an oil refinery." (Hall, 1992, p. 320)

old source

An example of a large manufacturer of PLCs is Siemens, they produce a variety of different types, along with other manufacturers. The three main types of PLC are as follows.

Unitary

Unitary PLCs usually consist of one unit with I/O ports, power supply, CPU and communication ports all built into the same unit. This type is typically the cheapest and smallest with a limited number of I/O ports and less processing power than the other types.



Figure 1:

<https://www.999mitsubishi.com/pub/media/catalog/product/cache/image/700x560/e9c3970ab036de70892d86c6d221abfe/a/l/al2-24.jpg>

not Harvard method

Figure 1 shows a Mitsubishi Alpha 2 unitary PLC, all functions are built into this one block including screen and I/O ports. The power is supplied directly to this one unit and is tied into all the I/O without the need for a rack or separate module.

Modular

Explained by Aries, with a modular PLC, the main unit only consists of a power supply, a CPU and maybe a couple of input ports. Any extra functions required such as analogue input ports, outputs and analogue to digital converters can be added if required. (The Types of PLCs)

This means that the modular type of PLC is very customisable to suit the required use without the high cost of a rack. Different units can be swapped out and upgraded if a higher processing power is needed, along with extra I/O ports being added at will in modules of their own.

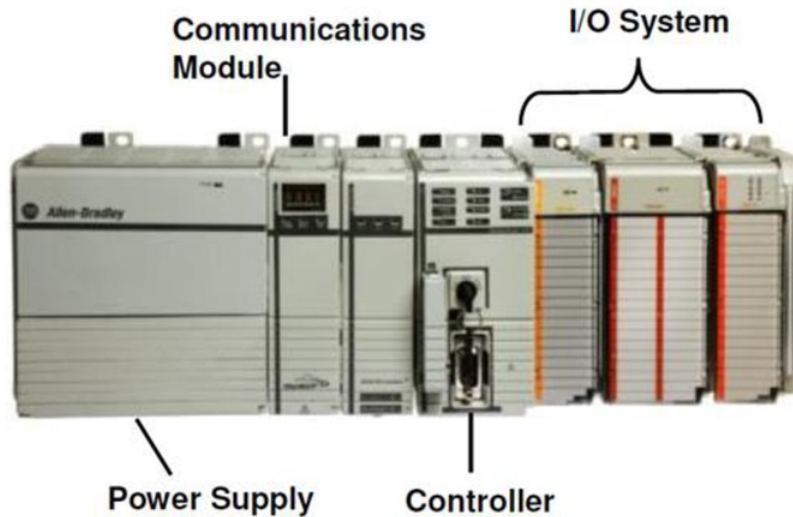


Figure 2: http://2.bp.blogspot.com/-CQ3MhI4oJwo/Ucqt6OA4XCI/AAAAAAAAA3E/Uo8gOrwjHe8/s1600/0001_Compact_Logix.png

Figure 2 displays a modular Allen Bradley PLC, with the different modules labelled and clearly separate from one another.

Rack Mounted

This type of PLC is typically the most expensive but also offers the most customisation and higher processing power with the ability to deal with large amounts of inputs and outputs for heavy industry. Similar to the modular type, the rack PLC is not all in one unit, with the different cards slotting into a wall mounted rack which enables communication and supplies power to everything slotted in.



Figure 3:<https://www.schneider-electric.com/ww/en/Images/tricon-IC-490x280.jpg>

Figure 3 shows a rack mounted Triconex PLC capable of dealing with hundreds of inputs and outputs. The rack houses each of the PLC cards with them being connected at the rear to the board rather than to each other. This system allows for faster communication as the data does not have to travel through each module to reach the CPU, it instead takes a faster route via the rack highway.

Languages

Different languages are used within PLC programming, these allow for different methods of carrying out the program and some are more suited to different tasks as opposed to others.

Ladder

Ladder logic is a form of programming language that involves lines of logic carried out from left to right. The design is similar in composition to how relay logic would work.

As Explained by KronoTech, *"The decision to use the relay logic diagrams was a strategic one. By selecting ladder logic as the main programming method, the amount of retraining needed for engineers and trades people was greatly reduced."* (PLC Languages)

year should be in the citation

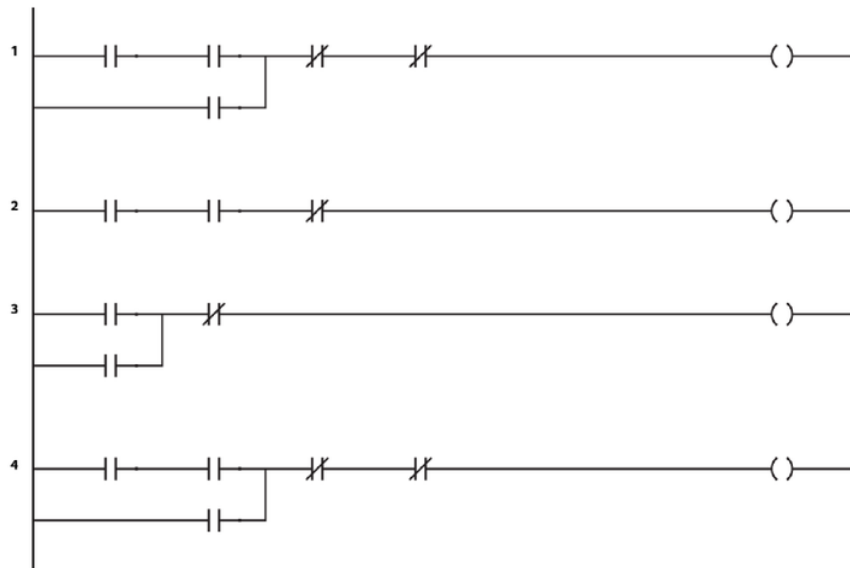


Figure 4: <http://www.plcacademy.com/wp-content/uploads/2017/06/ladder-logic-tutorial-rungs.gif>

Figure 4 shows the inputs on the left, mainly consisting of normally open/closed contacts are arranged in logic gates to provide a path to the output on the right. Other functions such as timers, counters and comparators can be added for added complexity if required.

Function Block

This type of programming language is similar to relay logic in the way it passes from left to right, however instead used logic and function blocks to provide the path to the output.

Figure 5 displays a function block language layout, with the logic gates and function blocks visible without the need for normally open/closed contacts.

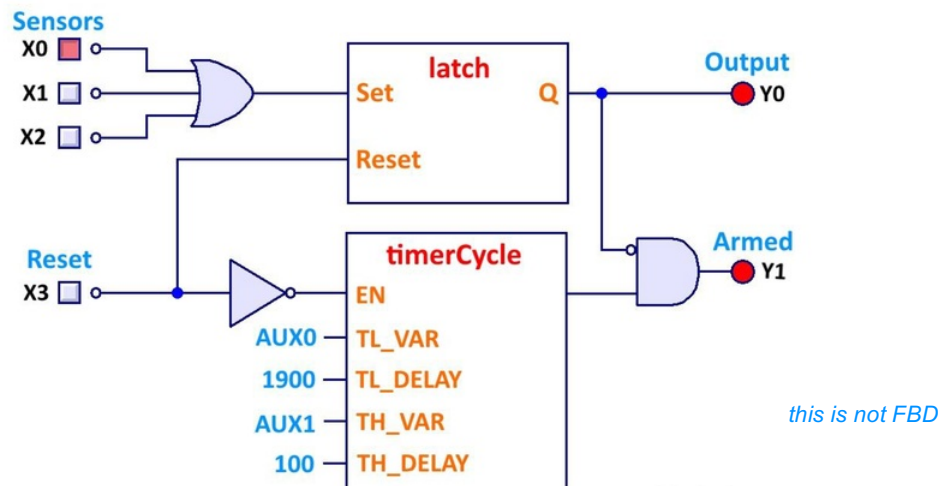


Figure 5: <http://www.electronics-micros.com/img/arduino/plc-lib-alarm2.jpg>

Sequential Function Chart

This language is different from the previous two as it does not carry from left to right it instead processes the data in a sequential manner from top to bottom.

Siemens explains how control performed sequentially allows for parallel and sequential operations to be carried out separate to time or events. (Sequential Function Chart, 2010, p. 1)

This means that certain conditions will need to be met before a step can transition to the next one,

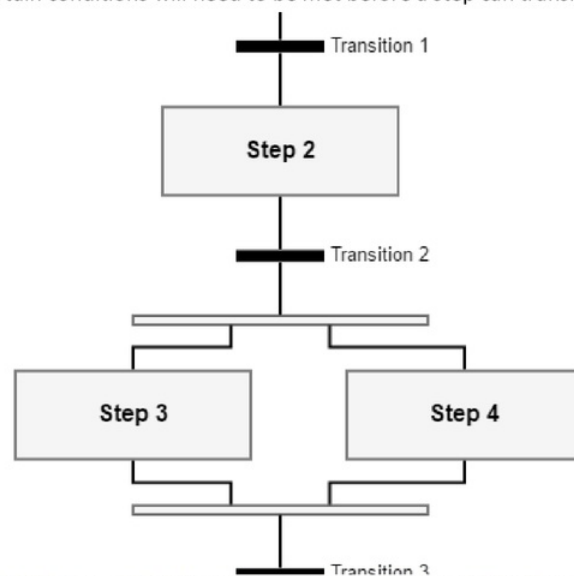


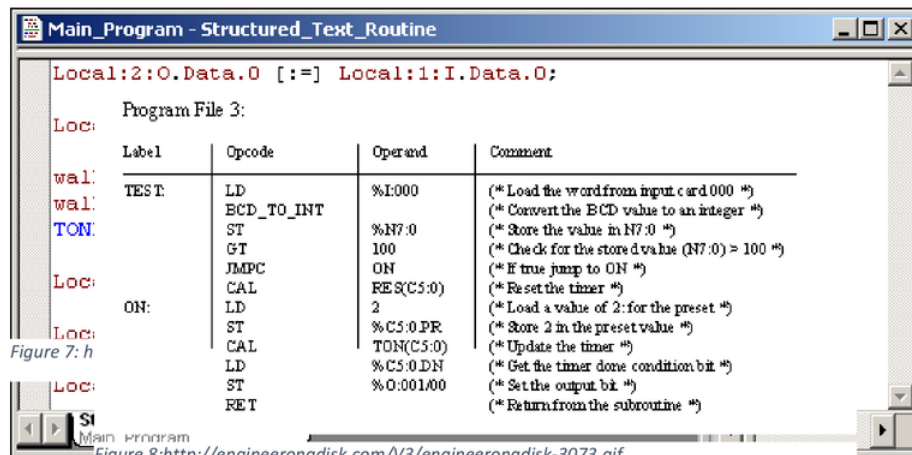
Figure 6: <https://gojs.net/latest/assets/images/screenshots/sequentialFunction.png>

being useful for automatic manufacturing systems that process large amounts of product repeatedly. It also means that the outcome can be achieved via separate paths if designed that way, this is shown in Figure 6.

Structured Text

This type of PLC language is different to the rest as it does not display the logic graphically, instead being lines of text with conditions to be met in order to achieve the desired output.

Figure 7 shows a program written in structured text language, numerical conditions can be seen and the timer functions written in. This type of language can be useful as it can be written quickly and does not take up much memory.



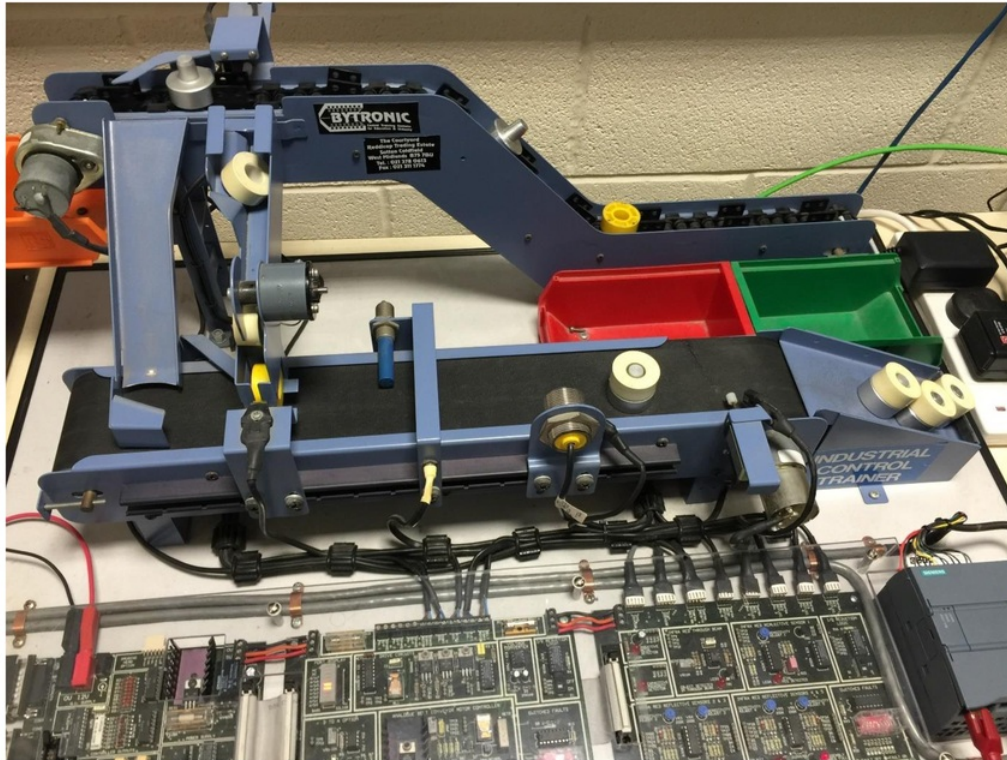
Instruction List

This language is a very simple language used for programming the PLC, as it is such a low-level language, similar to the language such as assembly used to initially program microprocessors and computers. It is a list of instructions to carry out a task under specified operators and conditions, taking up less memory than almost all other programming types.

two images have merged

Task 2 – Hardware Description

The hardware used as part of this assignment included the Siemens PLC and HMI, and industrial control conveyor sorting machine.



The above figure shows the industrial control trainer with all sensors and solenoid outputs. This system is designed to sort the plastic rings and metal pegs, assembling them into the combined finished product. The system is also designed to reject any incomplete products and excess parts into a waste bin.

A good brief description

Both types of parts are sent into the system via the motor operated chain conveyor, it is here where the plastic rings are sorted from the metal pegs. Once on the assembly line, a motor driven belt conveyor moves them along a series of sensors and finally a reject solenoid. The belt and chain conveyors are running constantly as the system is designed to run continuously. Completed products are dropped into the ramp on the right whereas the excess or incomplete parts are rejected into their respective bins ready to be sent back into the system.

The hardware used to control all this is a Siemens PLC 1200 and is built into the industrial control trainer along with a Siemens 14" HMI display. As part of this assignment, the HMI will also be configured to suit the needs of the equipment operator and an engineer for maintenance purposes. The IP address of these machines will need to be used to configure the program on the Siemens Totally Integrated Autonomy software in order to communicate properly.

More details of each device, the PLC and the HMI were expected

This is still section 2

Task 3 – Input/Output

Tag	Name	Description
I0.0	Assembly Area Sensor	Capacitive sensor that detects when a ring is ready to be dropped onto the belt.
I0.1	Sort Area Sensor	Detects when a part is in the region to be pushed into the assembly queue.
I0.2	Reject Area Sensor	Detects when a part or product is in the region to be pushed into the reject bin.
I0.3	Capacitive Area Sensor	Detects if there is a low part present when passing the sensors.
I0.4	Metal Peg Detect	Detects if a metal peg is present to tell the machine to let it pass along the chain.
I0.5	IR Through Beam Sensor	Beam to detect if anything is present mounted on the upper of the product.
I0.6	Capacitive Sensor	Detects if the assembled product is stacked correctly.
I0.7	Inductive Sensor	Ensures there is a metal peg present at this height.

Outputs

*Good
description
given of
the
function of
each
device*

Tag	Name	Description
Q0.0	Sort Solenoid	Pushes plastic rings into the assembly queue.
Q0.1	Rotary Solenoid	Droppes plastic rings into the assembly area.
Q0.2	Reject Solenoid	Pushes rejects into the waste bin.
Q0.3	Chain Conveyor	Moves parts up to the sorting area.
Q0.4	Belt Conveyor	Moves parts along the assembly area.

*This belongs with the program
explanation in section 3*

	Name	Data type	Address	Retain	Acces...	Writa...	Visibl...	C
1	Assembly Area Sensor	Bool	%I0.0	☐	☒	☒	☒	
2	Sort Area Sensor	Bool	%I0.1	☐	☒	☒	☒	
3	Reject Area Sensor	Bool	%I0.2	☐	☒	☒	☒	
4	Capacitive Area Sensor	Bool	%I0.3	☐	☒	☒	☒	
5	Metal Peg Detector	Bool	%I0.4	☐	☒	☒	☒	
6	IR Beam Sensor	Bool	%I0.5	☐	☒	☒	☒	
7	Sort Solenoid	Bool	%Q0.0	☐	☒	☒	☒	
8	Rotary Solenoid	Bool	%Q0.1	☐	☒	☒	☒	
9	Reject Solenoid	Bool	%Q0.2	☐	☒	☒	☒	
10	Chain Conveyor	Bool	%Q0.3	☐	☒	☒	☒	
11	Belt Conveyor	Bool	%Q0.4	☐	☒	☒	☒	
12	Capacitive Sensor	Bool	%I0.6	☐	☒	☒	☒	
13	Inductive Sensor	Bool	%I0.7	☐	☒	☒	☒	
14	Metal Peg on Belt	Bool	%M0.0	☐	☒	☒	☒	
15	Reject Area Sensor Pulse	Bool	%M0.1	☐	☒	☒	☒	
16	Assembled Product	Bool	%M0.2	☐	☒	☒	☒	
17	Counter Reset	Bool	%M0.3	☐	☒	☒	☒	
18	Assembled Counter Reset	Bool	%M0.4	☐	☒	☒	☒	
19	Reject Counter Reset	Bool	%M0.5	☐	☒	☒	☒	
20	Queue Full	Bool	%M0.6	☐	☒	☒	☒	
21	Reject Metal	Bool	%M0.7	☐	☒	☒	☒	
22	Reject Plastic	Bool	%M1.0	☐	☒	☒	☒	
23	Stop Chain	Bool	%M1.1	☐	☒	☒	☒	
24	Stop Belt	Bool	%M1.2	☐	☒	☒	☒	
25	Sort Sol Test	Bool	%M1.3	☐	☒	☒	☒	
26	Rotary Sol Test	Bool	%M1.4	☐	☒	☒	☒	
27	Reject Sol Test	Bool	%M1.5	☐	☒	☒	☒	
28	<Add new>			☐	☒	☒	☒	

Figure 9: PLC tags

Figure 9 shows the PLC tags used as part of this program, there are many memory bits added to allow for internal control on the HMI along with the ability to set and reset bits to remember if a certain part is in the correct place on the conveyor.

This should be section 4

Task 5 – System Testing and Evaluation

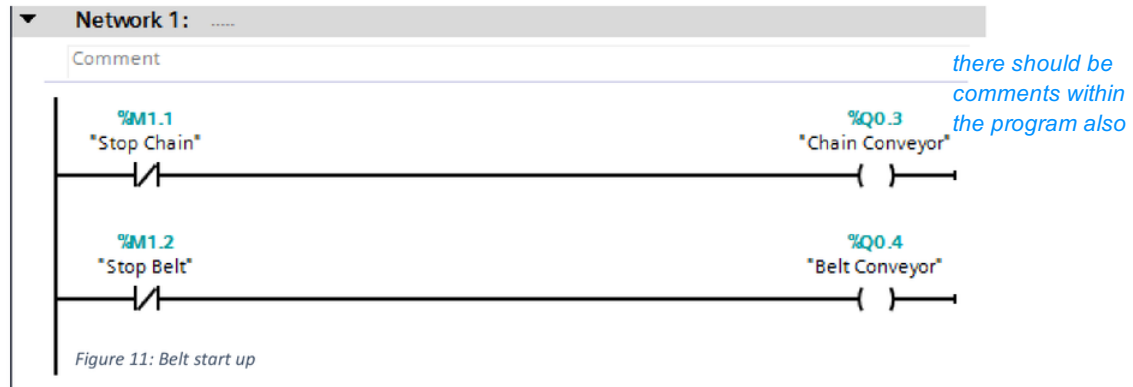
Assembly System Test Table			
Step	Action	Expected	Result
1	Power up system.	Belt and chain conveyors start so long as stop buttons are not pressed.	Conveyors started moving.
2	Load a plastic ring onto chain conveyor.	When ring reaches the sorting area, the sort solenoid should activate for 500ms and push ring into queue.	Plastic ring was pushed into queue by solenoid which activated for half a second when required.
3	Load up chain conveyor with a metal peg	The metal peg should continue along chain conveyor, passing the sort solenoid.	Metal peg passed sort solenoid and dropped into the appropriate chute.
4	Load up chain conveyor with six consecutive plastic rings.	Five plastic rings should be sent into the queue with one being loaded into the assembly area after a two second delay.	Solenoid pushed 5 plastic pegs down the shoot and rejected the 6 th plastic peg
5	Load up chain conveyor with seven plastic rings.	Rotary solenoid should activate 2 seconds after sort solenoid pushes plastic ring into queue with the counter dropping one count upon activation.	Counter counted up with sort solenoid activation and down with rotary activation after two seconds.
6	Fill up queue with plastic rings then load a metal peg onto chain conveyor.	Metal peg should pass sort solenoid onto belt conveyor and accept the plastic ring in assembly area. Two seconds later rotary solenoid should activate.	Assembled product made successfully with the rotary solenoid operating when required.
7	Load a false part (plastic ring) onto belt conveyor.	Plastic ring should be recognised as a foreign object and rejected into waste bin by a 500ms operation of the reject solenoid.	Ring was rejected as required.
8	Place metal peg onto chain conveyor to produce an assembled product.	Allows completed product to pass along belt conveyor until it drops into correct bin.	Assembled product was successfully recognised and was not rejected.
9	Place four metal pegs and 6 plastic rings onto chain conveyor.	Four assembled products should be made, leaving one plastic ring in the queue and one in the assembly area.	Two assembled products were rejected as there were multiple parts on the belt conveyor at the same time.
10	Perform same test as above but separate parts by one on chain conveyor.	No assembled products should be rejected as there will only be one part on the belt conveyor at any one time.	All four assembled products were made successfully and passed into correct bin.
11	Place two plastic rings and four metal pegs onto chain conveyor.	Two products should pass through, with both metal pegs being rejected.	Excess metal pegs rejected, and assembled product allowed to pass successfully. Product counters also incremented.
12	HMI screen test. Stop buttons tested, counter resets tested and solenoid tests operated.	All buttons should perform as required.	Counters all reset when required, solenoids operated under test functions.

Figure 10: Testing Table

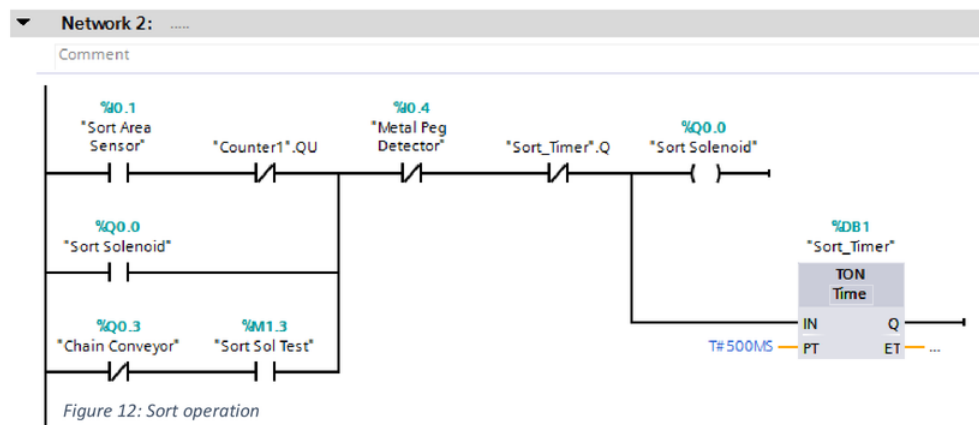
Overall the tests performed were successful as shown in Figure 10, the only issue was with parts being placed close together on the chain conveyor. Once spaced apart appropriately this problem was resolved, with assembled products being produced as required.

Task 6 – Program Explanation

Labwork 1 – Sorting Routine



The above network in Figure 11 is the first step of the sorting routine. It is important to have both conveyors running continually and so outputs for these were built. Normally closed contacts were added later down the line to add a stop function. However, without the stops pressed, both conveyors will run continuously.



Network 2 was constructed to allow the Sort Area Sensor (I0.1), to trigger the Sort Solenoid (Q0.0) when it detected a plastic ring in the vicinity. The OR gate ensured that the solenoid would hold itself energised until the 500ms timer expired, ensuring that the ring was sorted into the queue. The AND gate prevents the solenoid from energising should a metal peg be detected by the (I0.4) detector. The output from the first counter is present, this ensures the solenoid does not energise if the queue has reached its maximum capacity of 5. Any excess plastic rings will continue along and fall into the excess ring bin.

Labwork 2 – Queue Counting

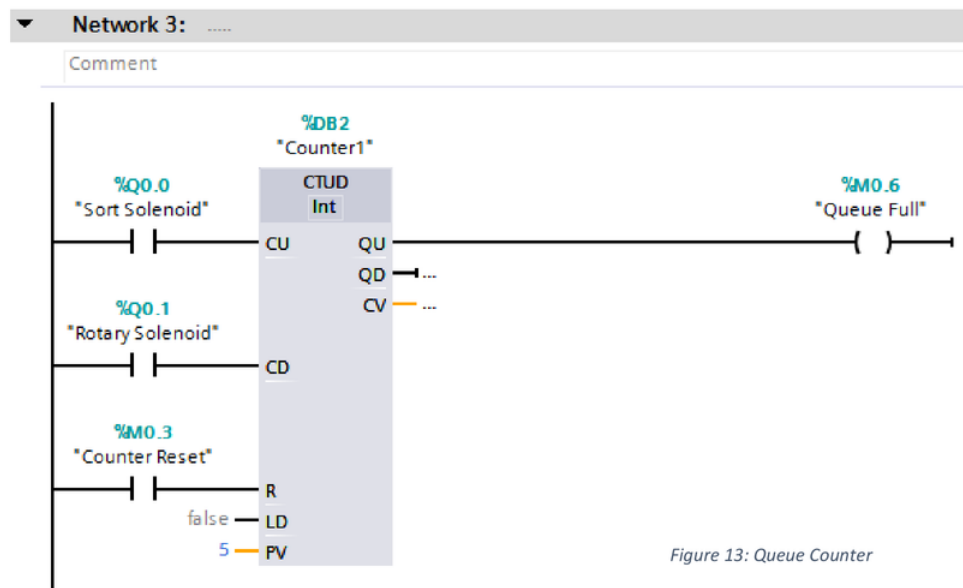


Figure 13: Queue Counter

Figure 13 above shows the up/down counter used for the queue system. Energising the Sort Solenoid increments the counter and energising the Rotary Solenoid (Q0.1) decrements it. This is down to a space being freed up in the queue upon operation of the rotary solenoid. The limit is set to 5 as this is the maximum available space in the queue. A manual reset is also incorporated so that the value can be reset on the HMI. The 'Queue Full' output is stored in memory and used to provide indication on the HMI when the capacity is reached.

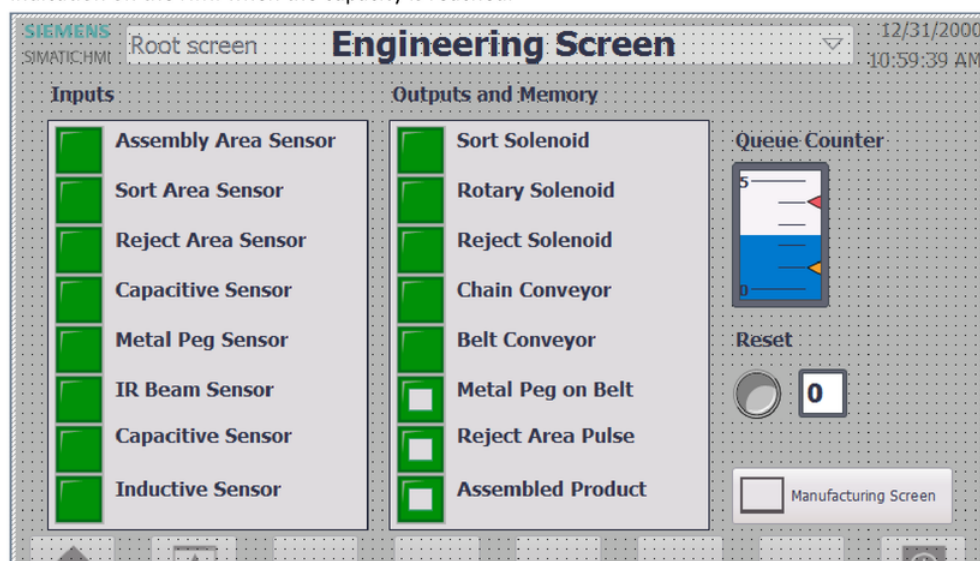


Figure 14: Engineer HMI Screen

The above HMI screen in Figure 14 was developed so all outputs could be viewed live along with the queue counter current value and reset.

Labwork 3 – Dispensing and Operation Timing

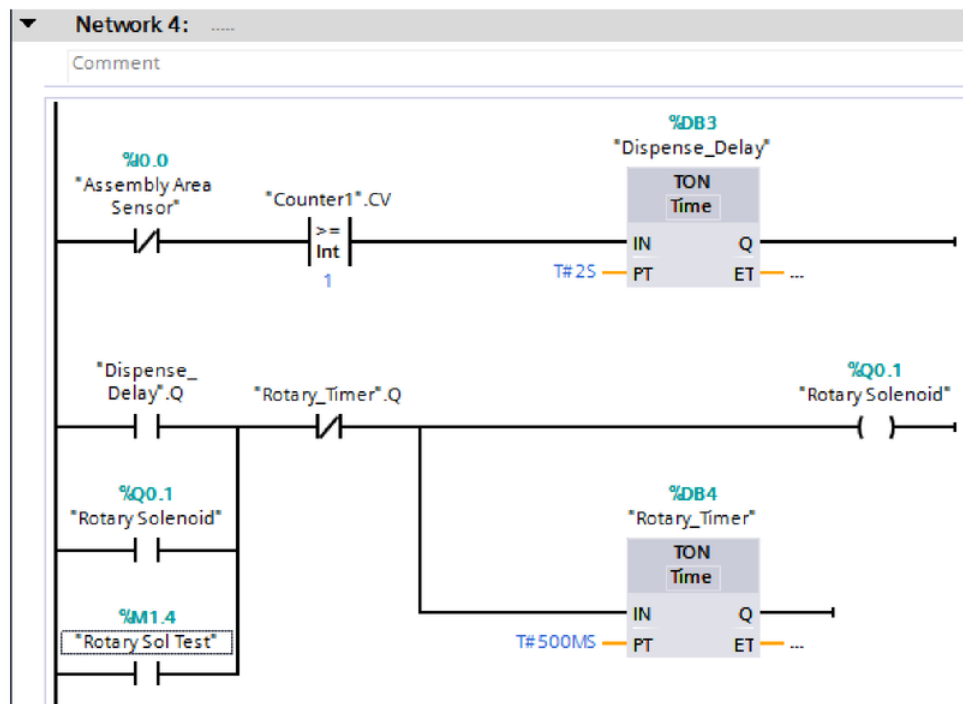


Figure 15: Dispense delay and operation

Figure 15 shown above is responsible for dispensing the plastic peg into the assembly area when one is present in the queue. The delay timer is triggered should there not be a part in the assembly area and there also be a queue count greater than 1. The timer lasts for 2 seconds and provides enough time for the recently dispensed ring to slide to the end of the queue. The rotary solenoid then operates for 500ms and the plastic ring falls into the assembly area before de-energising.

Labwork 4 – Plastic Ring Detection and Sorting

Network 5 in Figure 16 is designed to detect a plastic ring and reject it when necessary. The first rung allows a memory bit to be set if the inductive sensor is triggered. If this is not triggered, it is apparent there is not a metal peg on the belt and so it must be a plastic ring. Once the reject area is reached, the ring is picked up by the sensor and will be rejected if the metal peg on belt bit has not been set. This is apparent in the first rung on Network 6. Where the reject solenoid is also held open by the 500ms timer and once operated, the bit is reset by the Reject Area Sensor Pulse.

A counter is also present to record all rejected parts for the screen on the HMI.

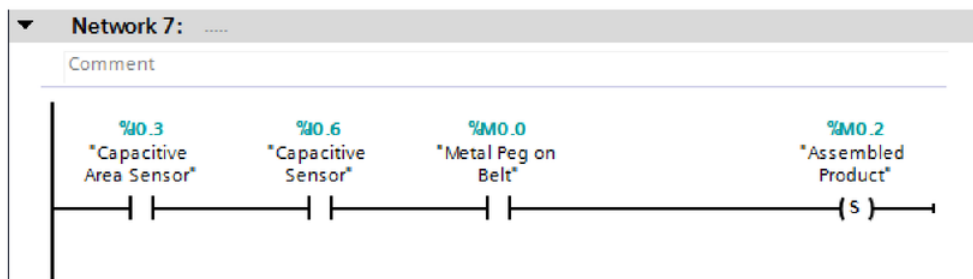
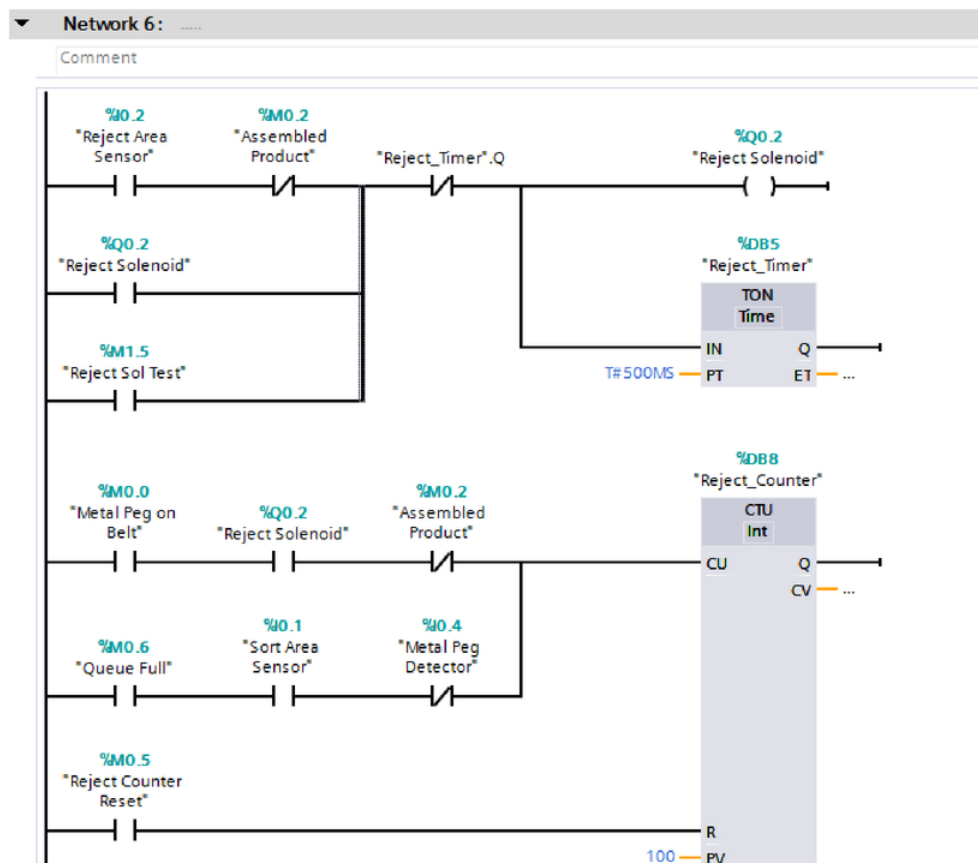


Figure 18: Assembled product memory bit set

The following HMI screen in Figure 19 was developed so that the operator could monitor the total amount of assembled products and rejected items. A reset for the counter is also present on this screen and an indication that both belts are running.

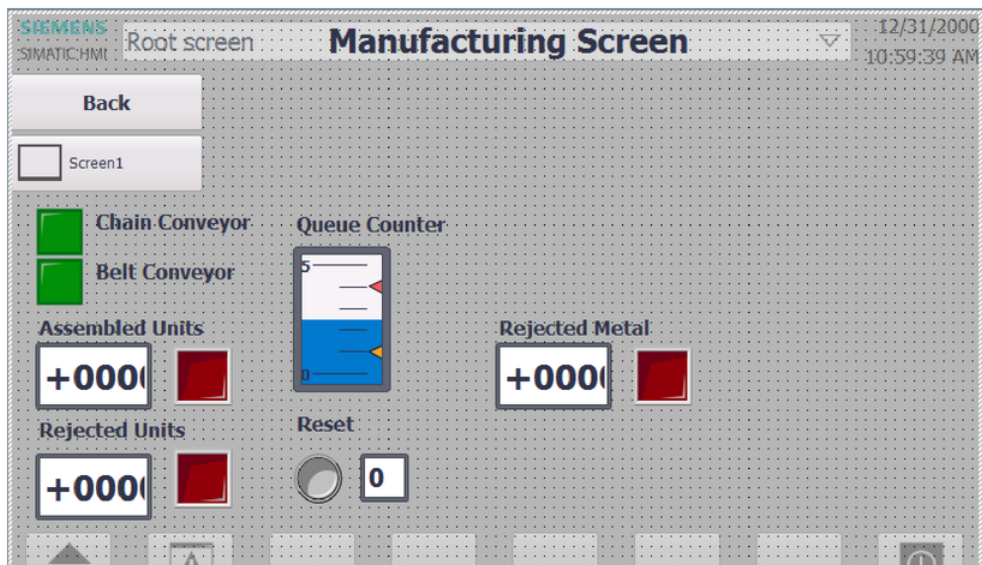


Figure 20: HMI manufacturing screen

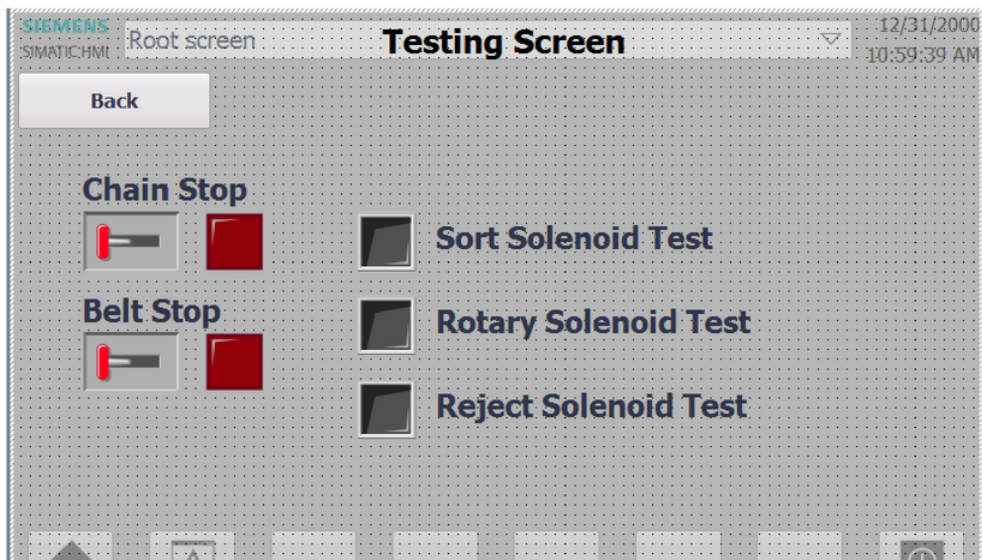


Figure 19: Solenoid Testing Screen

The above HMI testing screen in Figure 20 was developed to allow the belt and chain conveyors to be stopped in order for the solenoids to be tested in situ. They are built into Network 2, 4 and 6 for the sort, rotary and reject solenoids respectively.

Labwork 6 – Component Assembly Detection and Sorting

If an unassembled product reaches the reject area, the Assembled Product memory bit will not have been set, resulting in the reject solenoid being triggered as shown in Network 6 and 7. A correctly assembled component will have set the memory bit due to their being a metal peg on belt bit set and the capacitive sensor picking up the additional plastic ring. This then opens the normally closed contact on the first rung of Network 6 and prevents the reject solenoid from opening, allowing the product to continue into the bin.

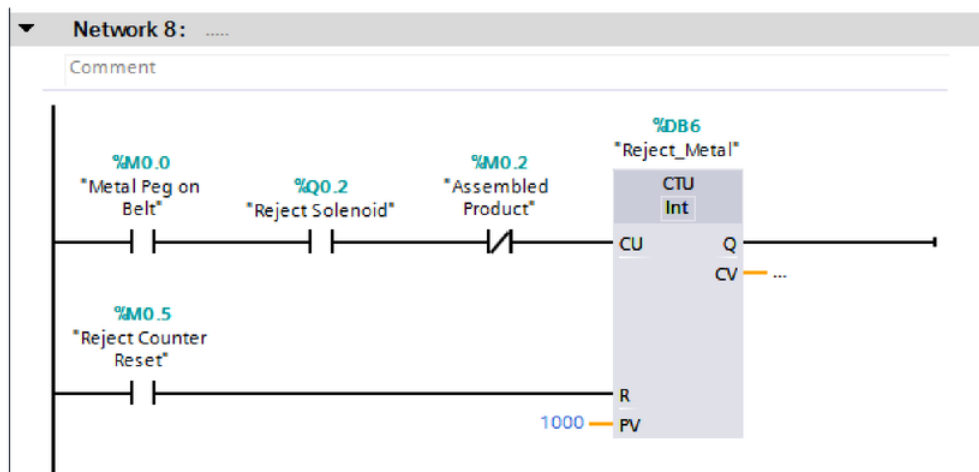


Figure 21: Reject counter and reset

Networks 8 (Figure 21) and 9 (Figure 22) contain resets for the reject counters which are to be activated when starting the equipment up. This is also visible on the HMI and all assembled products are counted along with rejects.

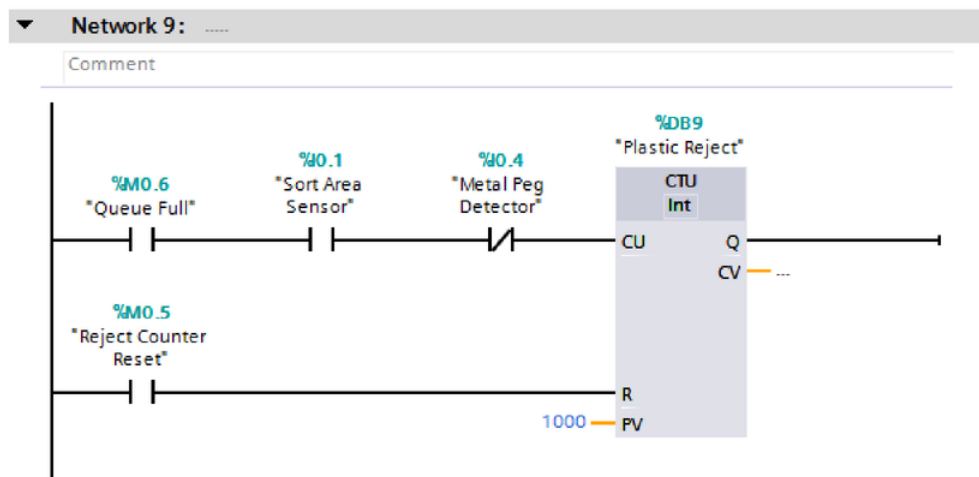


Figure 22: Plastic reject counter and reset

Task 7 – Conclusions and Recommendations

The main recommendation for this equipment and software would be to incorporate the ability to rapidly process and sort the parts one after another in the production line. This became apparent because the machine is unable to sort properly when there is more than one part on the belt conveyor, which happens if parts are placed directly behind one another on the chain conveyor. If this system was to be added, the equipment could effectively process double the number of products within the same time scale than normal. Meaning parts do not have to be placed two spaces apart on the chain conveyor when loading as is currently the case.

The issue happens because the memory bit is reset too soon by the part on the conveyor ahead as it passes the Reject Area Sensor. Meaning the Metal Peg on Belt bit is reset for the part just behind and is ultimately rejected even if it is a correctly assembled product because the Assembled Product bit can no longer be set as it does not currently store memory of the metal peg being present.

A way around this would be to add a sequence which remembers the order of each individual part and what it is, meaning only the memory set and reset by that part would influence the outcome. This would solve any interference from other parts causing unwanted rejections. Upgrading the amount of inputs or changing the programming language to a sequential and memory based program could solve this issue, such as the sequential flow chart language. This would be unaffected by time and the events such as other parts being on the belt conveyor and so their presence would not affect other parts on the belt.

Task 8 – Collaboration

When working as a group the most important aspect was communication between individuals, ensuring this meant that everyone understood what part of the task was to be completed and what others had achieved. In order to maximise efficiency, each member of the team worked to solve the program challenge together while the previous solution was built into the program, the team rotated to ensure no one person performed the same task throughout. Every individual kept an up to date copy of the program to prevent progress being hindered if someone was not present.

When deciding the layout of the HMI and program, each team member put forward their thoughts and recommendations and the universally decided best solution was put into the final product. If at any point a problem was thought to be present by an individual, they would voice their opinion and allow the group to focus on investigating and rectifying the issue.

Works Cited

- Aries, B. (n/a). *The Types of PLCs*. Retrieved December 02, 2017, from Techwalla:
<https://www.techwalla.com/articles/the-types-of-plcs>
- Hall, D. V. (1992). *Microprocessors and Interfacing* (2nd ed.). Retrieved December 02, 2017
- KronoTech. (n.d.). *PLC Languages*. Retrieved December 02, 2017, from Kronotech:
<http://www.kronotech.com/PLC/Languages.htm>
- Siemens. (2010, December). *Sequential Function Chart*. Retrieved December 02, 2017, from Industry Automation and Drive Technologies:
https://w3.siemens.com/mcms/sce/de/fortbildungen/ausbildungsunterlagen/hochschul-module/tabcardseiten/Documents/V7.0/eP01-07_sequential_function_chart_RC1012.pdf

PSDE4005 Assignment 1

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

60/100

PAGE 1

PAGE 2

Text Comment. Report should be structured according to the brief and not by task number since the marking scheme is linked to section numbers.

Text Comment. avoid putting URLs into figure titles

PAGE 3

Text Comment. good range of figures listed

PAGE 4

Text Comment. this is section 1 not task 1

Text Comment. engineer

Text Comment. old source

Text Comment. not Harvard method

PAGE 5

PAGE 6

PAGE 7

Text Comment. year should be in the citation

PAGE 8

Text Comment. this is not FBD

PAGE 9

Text Comment. two images have merged

PAGE 10

Text Comment. A good brief description

Text Comment. More details of each device, the PLC and the HMI were expected

PAGE 11

Text Comment. This is still section 2

Text Comment. Good description given of the function of each device

PAGE 12

Text Comment. This belongs with the program explanation in section 3

PAGE 13

Text Comment. This should be section 4

PAGE 14

Text Comment. This should be section 3

Text Comment. there should be comments within the program also

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

RESEARCH-INFORMED LITERATURE

-  Section 1: Your research - Some evidence of reading for PLCs and languages, explanations are generally very good. Higher marks could have been achieved by giving more PLC performance data and machine applications for languages. (9/15) Section 8: Your use of Harvard for citing and listing - Small range of sources listed and cited. A higher mark would have been given if a larger range had been consulted and listed correctly using the Harvard method. (2/5) **11**

KNOWLEDGE AND UNDERSTANDING

-  Section 2: Hardware Description and I/O list - Brief explanation given with a good I/O list. A higher mark would have been awarded for a more detailed description of each device including the PLC and the HMI. (5/10) Section 3: Program Explanation – A good tag list with detailed explanations of each program rung and HMI screen given. (7/10) **12**

ANALYSIS

-  Section 4: System Testing and Evaluation - Good testing table with a brief evaluation. Further marks were available for a more in-depth evaluation and photographic evidence. (6/10) Section 5: Conclusions and Recommendations - A very good **11**

set of recommendations offered. Further marks were available for a conclusion. (5/10)

PRACTICAL APPLICATION AND DEP.



Section 3: Program Design and Network Comments
- Very good program design. Further marks were available for network comments on each rung. (5/10)
Section 3: HMI Screen Designs – Good clear screen designs. (9/10)
Section 4: Presentation of Test Results - A good table but no photographic evidence provided. (5/10)

19

SKILLS FOR PROFESSIONAL PRAC.



All sections: A well written and presented report. Further marks were available for a more correct structure as stated in the brief. (4/5)
Section 6: Team working - good evidence of team working observed and briefly explained. Further marks were available for explaining your role in the team. (3/5)

7