

Assignment 1 PSDE 4005

by Thomas Wood

FILE	15329_THOMAS_WOOD_ASSIGNMENT_1_PSDE_4005_154267_2061473 30.DOCX (1.77M)		
TIME SUBMITTED	03-DEC-2017 02:18PM (UTC+0000)	WORD COUNT	7766
SUBMISSION ID	78844636	CHARACTER COUNT	41519



HNC ELECTRICAL AND ELECTRONIC ENGINEERING

PSDE4005 Programmable logic controllers - Assignment 1



NOVEMBER 6, 2017

THOMAS WOOD

Word count: 7662

Contents

Table of figures	2
1. Research.....	3
2. Hardware description and I/O list.....	8
3. Program explanation	15
4. System testing and evaluation	23
5. Conclusions and recommendations	27
6. Collaboration.....	28
7. Citation list.....	29
Works Cited	29

Table of figures

Figure 1 - Basic ladder diagram language (Peter, 2015).....	4
Figure 2 - Basic instruction list language (Anon., n.d.)	5
Figure 3 - Function block diagram language (Anon., n.d.)	5
Figure 4 - Structured text language (Anon., n.d.)	6
Figure 5 - Sequential function chart language example (Hechtman, 2014)	7
Figure 6 - PLC utilised on the trainer	8
Figure 7 - Sort area of the trainer	9
Figure 8 - Assembly area of trainer	10
Figure 9 - Inductive and Infra-red sensors on trainer	10
Figure 10 - Capacitive sensor and are sensor.....	11
Figure 11 - Reject components of trainer	12
Figure 12 - Both conveyors of the trainer	12
Figure 13 - PLC tag table	15
Figure 14 - Network 1.....	15
Figure 15 - Network 2.....	16
Figure 16 - Network 3.....	17
Figure 17 - Network 4.....	17
Figure 18 - Network 5.....	18
Figure 19 - Network 6.....	19
Figure 20 - Network 7.....	19
Figure 21 - The engineer HMI design	20
Figure 22 - The operator HMI design	20
Figure 23 - Network 8.....	21
Figure 24 - Network 9.....	21
Figure 25 - Network 10.....	22

1. Research

Many different forms of PLC (programmable logic controllers) are available, this is due to the many different manufacturers and the differing demand of the performance of the PLC's. A well-known PLC manufacturer is Siemens; Siemens have an array of different PLC's suited to different applications.

There are 3 different types of PLC; unitary, modular and rack mounted. Siemens offer each of these types within their products, it is therefore important that each type of PLC is compared in order to decide the most suitable type for a certain application.

Siemens offer a unitary PLC; this PLC is named the Siemens S7-1211c.

Performance

This PLC has a limited amount of inputs and outputs, this is because the PLC is a unitary type, which means that no extension modules can be installed to the rail. This particular PLC is small in size and can therefore be mounted in confined spaces such as small panels. A signal board can be added to the front of the PLC, this allows a limited amount of inputs and outputs to be added to the module. Cooper states that the 'S7-1211c is ideal for open-loop and closed-loop control tasks in mechanical equipment manufacture and plant construction' (n.d.). This PLC has a slow processor speed and also low memory, this means that complex programs could not be saved to the PLC and also the time taken for actions to be completed is relatively long.

Cost

The S7-1211c is available for purchase for £222 from rapid electronics (Cooper, n.d.), this price is relatively cheap compared to the other types of PLC, however with the capabilities of this PLC it can be deemed to be a very good price for a high quality PLC.

Siemens offer a modular PLC; this PLC is named the Siemens S7-1500.

Performance

This PLC theoretically can have a large amount of input and outputs, this is due to the capabilities of this PLC to have expansion modules added. This PLC is also relatively small in size however after adding some additional modules can become extensively larger, therefore the mounting places should be rather spacious to allow future additions to be made. Gilbert states that 'a modular PLC has the capacity to accomplish more complex processes making it advantageous.' (2014). This is possible due to the PLC's respectable memory size and processor speed. Modular PLC's are commonly used within the manufacturing industry.

Cost

The simplest form of the Siemens S7-1500 which has 16 inputs and 16 outputs, is available for purchase from 'Parmley Graham' at a price of £1,155. This price is almost 6 times the price of the unitary PLC which shows the capabilities of this PLC is much larger.

Siemens also offer a rack mounted PLC; this PLC is named the Siemens S7-400.

Performance

This PLC is formed on a single rack; each module is plugged into the rack. The PLC itself has a high amount of inputs and outputs and even has the opportunity for more racks to be added if desirable. This PLC is also very easy to customise this is because of how simple it is to remove and replace a card from the module (Anon., 2015). Processor speed and memory are extremely good, therefore the rack mounted PLC can carry out very complex programs. This PLC is large in size which makes it unsuitable for most applications therefore this PLC is normally only used within large industries such as oil refineries.

Cost

The S7-400 is currently for sale on 'Parmley Graham' for £12,691, although this price seems excessive compared to the other PLC's it is normally justified within certain applications due to the PLC's workability and features.

There are 5 different languages that can be utilised in order to write a program for a PLC, these 5 languages are (as stated by the IEC Standard 61131-3) (Anon., 2013);

- Ladder Diagram (LD)
- Instruction List (IL)
- Function Block Diagram (FBD)
- Structured Text (ST)
- Sequential Function Chart (SFC)

In order to better understand each language, an example of each language and a suitable application for that particular language is going to be produced.

Ladder Diagram (LD)

Ladder diagrams are a sort of language that allows the coder to physically draw components into the language. Ladder logic uses an array of different components such as relays, N/O (normally open) contacts and timers. By the utilisation of many rungs containing components logic can be built. An example of how a basic stop/start system would be coded with LD can be seen below;

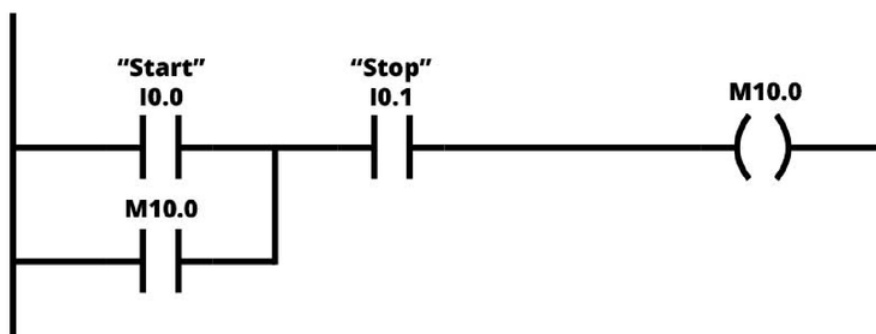


Figure 1 - Basic ladder diagram language (Peter, 2015)

LD is most commonly used within industries; this is normally because of the simplicity of the language due to the fact that it is a graphical language. Although LD is considered to be relatively simple to understand however LD is also reliable enough to be utilised to control critical systems.

Instruction List (IL)

Instruction list is a language that utilises different Boolean operations in order to perform a task. A Boolean can either be true or false, by utilising many different of these Boolean commands programs can be written. An example of a piece of IL coding can be seen below;

LD	A	(*Load A*)
AND	B	(*AND B*)
ST	Q	(*Store result in Q, i.e. output to Q*)

Figure 2 - Basic instruction list language (Anon., n.d.)

IL is a language that the PLC can complete extremely quickly therefore this makes this language perfect for use in extremely fast control loops.

Function Block Diagram (FBD)

A function block diagram is a program which is formed by utilising an array of blocks. A block is formed through combinations of different functions or function blocks. Formed blocks are then connected to allow certain blocks to effect other blocks. An example of a short piece of an FBD program can be seen below;

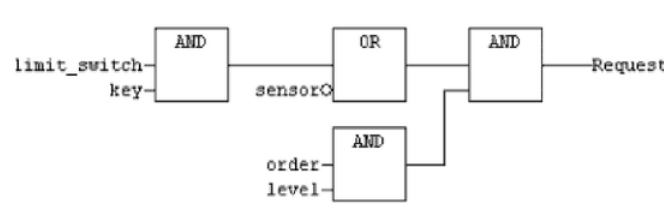


Figure 3 - Function block diagram language (Anon., n.d.)

FBD is a language that is common to LD, FBD is normally utilised after a program has been designed in LD. The LD language is then converted to FBD so that the PLC can execute the program quicker than in LD. FBD is widely used within industries.

Structured Text (ST)

Structured Text is a text-based programming language, the code itself is formed by utilising an array of different instructions. Within these instruction high level language is utilised, HLL is an instruction set that can be understood by a computer but is also easily identifiable to a human who is coding or analysing the code.

```
(* TEST CYCLE SETUP *)
cycle_In1 := tb_In1[testCycleNum];
cycle_In2 := tb_In2[testCycleNum];
cycle_In3 := tb_In3[testCycleNum];
cycle_Out := tb_Out[testCycleNum];
IF testCycleNum = 0 THEN
  (* INIT *)
  PID_Subsystem(i0_PID_Subsystem, 0, cycle_In1, cycle_In2, cycle_In3, out_Out);
END_IF;
(* STEP *)
PID_Subsystem(i0_PID_Subsystem, 1, cycle_In1, cycle_In2, cycle_In3, out_Out);
(* VERIFY *)
IF testVerify THEN
  IF cycle_Out = 0.0 THEN
    IF ABS(out_Out) > 9.9999997473787516E-5 THEN
      testVerify := 0;
    END_IF;
  ELSEIF ABS(out_Out - cycle_Out) > (9.9999997473787516E-5 * ABS(cycle_Out)) THEN
    testVerify := 0;
  END_IF;
END_IF;
testCycleNum := testCycleNum + 1;
END_IF;
END_IF;
```

Figure 4 - Structured text language (Anon., n.d.)

ST is a very modern programming language; it is starting to become one of the most desirable programming language due to its user friendly HLL language. ST is currently utilised in order to produce control algorithms.

Sequential Function Chart (SFC)

Sequential function charts are a graphical programming language that utilises algorithm based logic in order to program. Budimir states 'The key concepts underlying SFCs are steps and transitions' (2017). A step is considered to be a form of function such as a process, and a transition is a state that must be met in order for the next step to be executed.

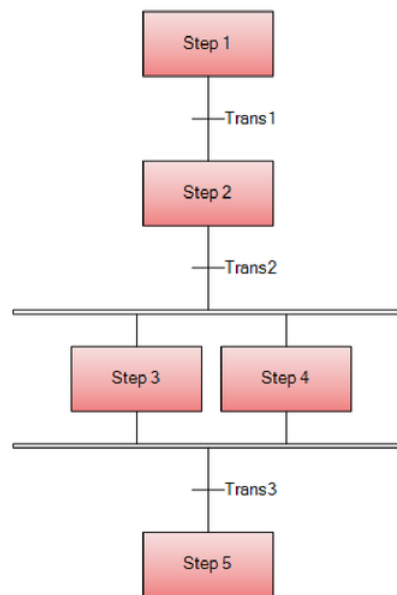


Figure 5 - Sequential function chart language example (Hechtman, 2014)

SFC is a language that is not normally utilised as much as the other PLC languages, this is because this language is widely utilised within advanced systems and therefore doesn't normally get utilised for every day programming.

2. Hardware description and I/O list

On the 'industrial control trainer' there are many different pieces of hardware. Hardware is any physical component that is utilised on the system. An example of a piece of hardware on this trainer is a solenoid. In all there is 14 pieces of hardware, these are;

- PLC
- Metal Peg Detector
- Sort Solenoid
- Sort Area Sensor
- Rotary Solenoid
- Assembly Area Sensor
- Inductive Sensor
- Infra-red Through Beam Sensor
- Capacitive Sensor
- Capacitive Area Sensor
- Reject Solenoid
- Reject Area Sensor
- Belt Conveyor
- Chain Conveyor



Figure 6 - PLC utilised on the trainer

- PLC (Figure 6)

The PLC (programmable logic controller) is an assembly of many different pieces of hardware. By utilising the correct combinations of hardware a PLC can be utilised in order to control a circuit in a way that it is programmed to do so. The usual hardware that is found within a PLC is; Memory, I/O, CPU (central processing unit), PSU (power supply unit) and a programming device. The PLC within the trainer is utilised as the main controller of the circuit, by connecting the relevant I/O hardware to the PLC each individual piece of hardware can be monitored and controlled.



Figure 7 - Sort area of the trainer

- Metal peg detector (Figure 7)

The metal peg detector is a piece of hardware that is utilised within the trainer as an input to the PLC; the detector is a form of metal sensor. Whenever metal moves past the detector it will trigger the sensor and therefore give a pulse to the PLC. The metal sensor will only pick up the metal pegs moving past it, and therefore will not notice a plastic piece when it goes past the sensor.

- Sort solenoid (Figure 7)

The sort solenoid is a piece of hardware that is an output from the PLC, this therefore means that in order for the hardware to operate, the PLC has to tell it to do so. When a signal is sent to the solenoid it pushes a lever out, this lever is also then told when to return back to its normal position by the PLC. This sort solenoid is utilised in order to put the plastic parts into position for assembly by pushing them off the conveyor and down the feeder chute.

- Sort area sensor (Figure 7)

The sort area sensor is a piece of hardware that is utilised as an input, this particular sensor provides the PLC with a signal whenever the sensor is triggered. This means that whenever an assembly piece goes past the sensor then a signal will be given to the PLC. The sort area sensor is utilised together with the metal peg detector in order to allow the PLC to understand whether to operate the sort solenoid or not.



Figure 8 - Assembly area of trainer

- Rotary solenoid (Figure 8)

The rotary solenoid is a piece of hardware that is an output from the PLC, the sort solenoid is utilised in order to ensure that only one plastic piece is dispensed into the assembly area at any given time. In order for this to happen many different piece of hardware is analysed by the PLC and then at the correct time a signal is sent from the PLC to the rotary solenoid. When the rotary solenoid operates it utilises a cleverly designed CAM in order to ensure that only one plastic piece from the chute is allowed to pass.

- Assembly area sensor (Figure 8)

The assembly area sensor is an input piece of hardware that provides the PLC with a signal whenever there is a plastic piece in the assembly area, whilst there is a plastic piece in the assembly area this sensor will be constantly triggered and will therefore provide a constant signal to the PLC, as soon as this signal is broken then it is known that there is now a need for one of the queuing plastic pieces to be dropped into the assembly area.

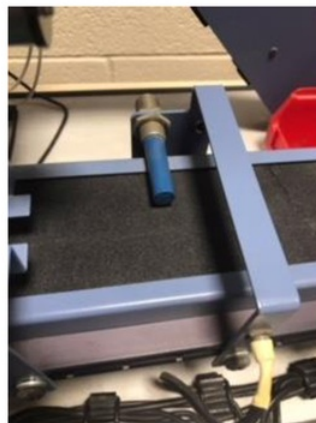


Figure 9 - Inductive and Infra-red sensors on trainer

- Inductive sensor (Figure 9)

The inductive sensor is a piece of hardware that is utilised as an input to the PLC, the sole purpose of the sensor is to provide a signal to the PLC when it detects anything metallic (metal peg) moving past it. This ensures that the metal peg is present within the finished assembly.

- Infra-red through beam sensor (Figure 9)

The infra-red through beam sensor is a piece of hardware that is utilised as an input to the PLC, the through beam sensor ensures that the metal peg has not been installed on top of the plastic piece, it does this as the plastic and the metal peg would have a different effect on the infra-red sensor. This therefore allows the sensor to differentiate between the two materials. Therefore, if the infra-red is different then a signal can be sent to the PLC.

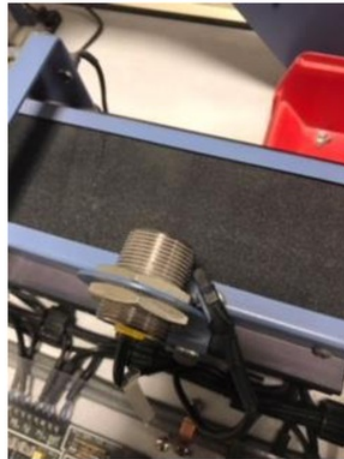


Figure 10 - Capacitive sensor and are sensor

- Capacitive sensor (Figure 10)

The capacitive sensor is a piece of hardware that is utilised as an input to the PLC, the purpose of the capacitive sensor is to ensure that the plastic piece has been installed correctly to the metal peg after the final assembly. The capacitive sensor works by providing the PLC with a signal when it detects a plastic piece.

- Capacitive area sensor (Figure 10)

The capacitive area sensor is a piece of hardware that provides an input to the PLC, the sensor will provide a signal to the PLC whenever it is triggered. This particular sensor is triggered whenever something on the conveyor moves past the sensor, this therefore allows the PLC to utilise the capacitive sensor in line with this sensor in order to conclude whether the plastic piece has been installed correctly.



Figure 11 - Reject components of trainer

- Reject solenoid (Figure 11)

The reject solenoid is a piece of hardware that is utilised as an output from the PLC, when a signal is sent from the PLC the solenoid will engage and push a bar out of itself, in this case the bar is utilised in order to knock faulty components off the conveyor and thus not allowing incomplete products from entering the completed area. A signal is sent from the PLC through the use of logic, this logic utilises many different inputs to determine whether the assembled product is correct or not.

- Reject area sensor (Figure 11)

The reject area sensor is a piece of hardware that is utilised as an input to the PLC, the sensor will provide an input signal to the PLC whenever an assembly is detected on the conveyor belt in the reject area. This signal can be utilised in line with other given signals to the PLC in order to decide whether to accept or reject that particular assembly.

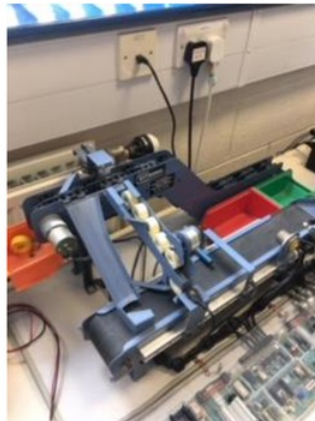


Figure 12 - Both conveyors of the trainer

- Belt conveyor (Figure 12)

The belt conveyor is a piece of hardware that is utilised as an output from the PLC, the conveyor works by utilising a motor. This motor is utilised in order to turn a rubber belt around an assortment of rollers. This belt has grooves upon it that allow for the movement of items that are placed upon the belt. The belt's motor will only turn when given a signal from the PLC.

- Chain conveyor (Figure 12)

The chain conveyor is a piece of hardware that is utilised as an output from the PLC, the chain conveyor is relatively similar to the belt conveyor and also works by utilising a motor in order to turn the chain around rollers. The chain conveyor has grooves upon it that allow for the movement of items that are placed upon the conveyor. This also allows inclines and declines to be utilised in the conveyor.

I/O List

In order to draw up a completed I/O table, the first thing that is to be done is to identify all Inputs and Outputs that will be utilised within the 'control trainer'. In order to identify each, I/O a diagram was utilised. From using the diagram, the following table was produced;

Inputs	Outputs
Metal peg detector	Chain conveyor
Sort area sensor	Belt conveyor
Assembly area sensor	Reject solenoid
Infra-red through beam sensor	Rotary solenoid
Capacitive sensor	Sort solenoid
Inductive sensor	
Capacitive area sensor	
Reject area sensor	

Following the identification of all I/O in the circuit the next step was to tag each I/O with the suitable identification tags for the PLC program. In order to identify each input and output the trainer was utilised. By manually making inputs and outputs the tags of each I/O were identified from looking at the PLC's identification lights. The I/O tag format for Siemens is shown below;

- Outputs - Q0.1
- Input - I0.1

By utilising the above format, the table of I/O tags to be used within the program were produced, the tables can be seen below;

OUTPUTS	TAG NAME
Q0.0	Sort Solenoid
Q0.1	Rotary Solenoid
Q0.2	Reject Solenoid
Q0.3	Chain Conveyor
Q0.4	Belt Conveyor

INPUTS	TAG NAME
I0.0	Assembly Area Sensor
I0.1	Sort Area Sensor
I0.2	Reject Area Sensor
I0.3	Capacitive Area Sensor
I0.4	Metal Peg Detector
I0.5	Infra-red Detector
I0.6	Capacitive Sensor
I0.7	Inductive Sensor

3. Program explanation

The final program itself was developed in six major steps, each of these steps were known as different Labworks. In all there was six different labworks, each labwork had several different requirements that the program needed to meet. By following this structure of writing the program it ensured that nothing was overlooked and therefore that everything that the program needed to do was included within the program. In all the program had 10 network's (rungs), each one of these networks are important to the successful completion of the trainer.

The first step that was undergone when beginning to program the trainer, this step was to add the I/O tags into the program. In order to do this the PLC tags within the program was opened and then the I/O tags were physically input into the program. Figure 13 illustrates the PLC tag screen after the addition of the relevant I/O tags. Three memory bits were also added to the tag list.

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	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 13 - PLC tag table

The very first requirement for the program was for both conveyors to operate whenever the trainer was switched on. This was achieved by adding the output tags of both conveyors to coils and then connecting them to rungs. Figure 14 shows how this was achieved within network 1. There was no need for any NO (Normally open) or NC (Normally closed) contacts to be added to the rungs as the PLC itself has a latching on button wired into it that therefore means that it must be engaged for the conveyors to work.

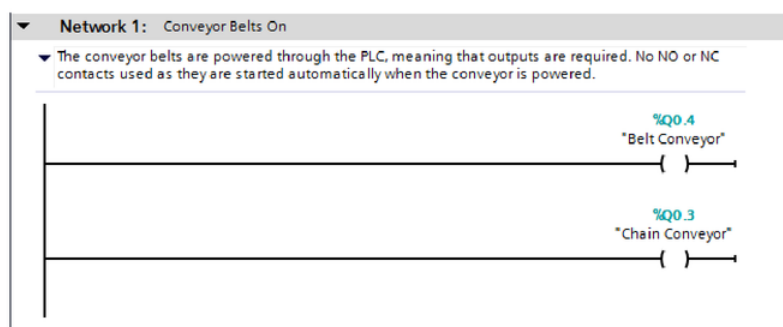


Figure 14 - Network 1

The next requirement for the program was for the plastic pieces to be moved into the assembly chute. In order for this to take place there was need for two sensors and a solenoid. The sensors that were needed were the sort area sensor and the metal peg detector. These sensors were needed in conjunction with each other in order to determine whether the component passing was plastic or not. If the sort area sensor is triggered but the metal peg detector is not, then the PLC can conclude that the component is plastic and therefore send a signal to the solenoid to knock the plastic component off the conveyor and down the assembly chute.

Figure 15 shows how the above logic was achieved, a counter was also added to this network so that no more than five plastic pieces are put down the chute at once, this stops any overfilling of the assembly chute to occur and therefore prevents blockades. As the plastic piece is moved away from the sort area sensor the signal from the sensor to the PLC is lost, in order to prevent this from ensuring that the solenoid engages sufficiently to push the plastic piece down the chute, a link is utilised that bypasses the need to have the sensor signal constantly on. By doing this with a NO of the solenoid itself it means that the solenoid can stay energised for 0.5 seconds even after the sensor loses its detection of the plastic piece.

After the solenoid engages it must return to its normal state before the next groove of the conveyor goes past, in order for this to be achieved a TON timer 'Sort solenoid operation timer' was utilised, this particular timer allows the solenoid to stay engaged for half a second and then breaks its NC contact and therefore removes signal from the solenoid and draws the solenoid back into position. this network is also connected so that whenever a metal peg goes past the solenoid it is allowed to the pass, this is achieved by breaking the contact of the metal peg detector.

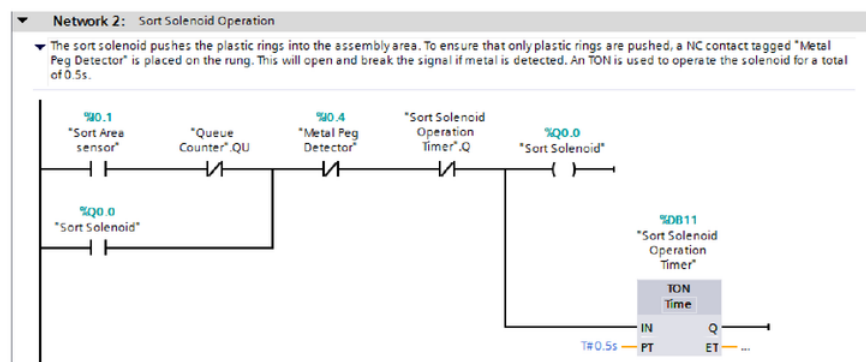


Figure 15 - Network 2

The queue counter itself works by adding 1 to its subtotal every time that the sort solenoid is engaged. Once the subtotal of the queue counter reaches 5 the PLC is then told to stop allowing plastic pieces to be moved down the assembly chute, this is achieved by putting a normally closed contact into the rung in Figure 15 and labelling it with the queue counter. The queue counters subtotal can also be reduced when the rotary solenoid is energised, this works by allowing one of the plastic pieces from the assembly chute into the assembly area. Whenever this occurs 1 is subtracted from the subtotal of the queue counter and therefore a plastic piece can then be put down the chute by the sort solenoid, this can be seen in Figure 16.

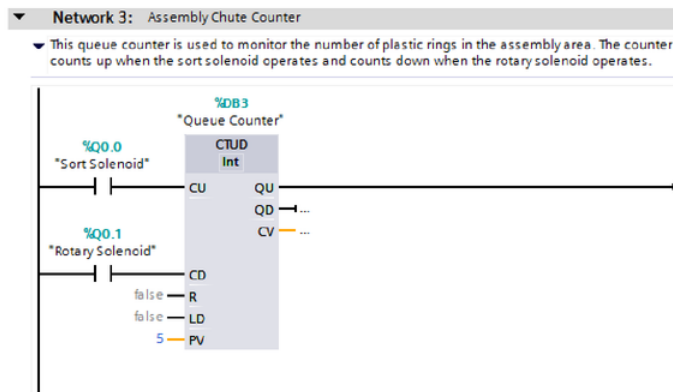


Figure 16 - Network 3

The following part of the program is to allow a single plastic piece to be put into the assembly area as long as there is not already one in the assembly area. In order to do this a sensor must be used in order to ensure that there is room in the assembly area for plastic piece, therefore this sensor is utilised in the program as a normally closed contact as it will only break the circuit when it is triggered and therefore there must be a plastic piece in the assembly area. The next thing to check would be the number of plastic pieces in the assembly chute is sufficient (over 1), this is achieved by utilising a function, this function is; if the queue counter is equal to or greater then 1 allow the signal to pass. This is better represented within Figure 17. A delay timer for the rotary solenoid is also incorporated into the rung, the delay is two seconds.

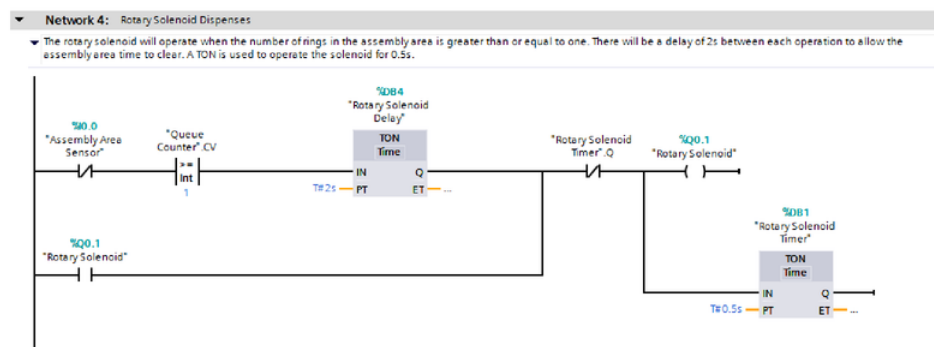


Figure 17 - Network 4

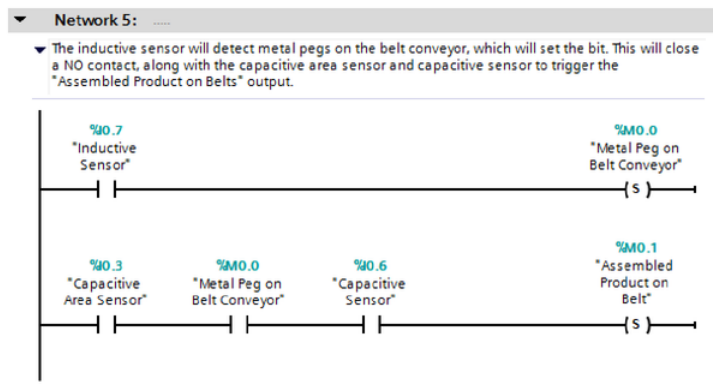


Figure 18 - Network 5

Figure 18 represents the next network within the program, this part of the program is the assortment of different sensors on the belt conveyor. These sensors need to be utilised with appropriate logic in order to determine whether an assembled product is correct or not. The first rung on the network is the very first sensor on the conveyor, the inductive sensor is utilised in order to determine whether a metal peg is in the assembly, a normally open contact is utilised as if a metal peg is detected then it will close the contact and therefore set the memory bit 'metal peg on belt conveyor' to on.

This memory bit is then also utilised on the next rung. It is utilised in line with the capacitive area sensor and the capacitive sensor. When an assembly is seen by the capacitive area sensor it will close its NO contact and therefore allow the signal to travel. If the memory bit of the metal peg is also on then the signal will then travel to the capacitive sensor. The capacitive sensor is also a normally opened contact because if it detects a plastic piece then it will close that contact and set the memory bit 'assembled product on belt' set to on. This therefore means that in order for the memory bit to be set all three conditions must be in the on state.

The assembled product memory bit is now utilised within the next network. Figure 19 shows the next network; this network is utilised in order to determine whether the assembly is to be rejected by the reject solenoid or not. In order for this to be achieved the reject area sensor is first of all utilised, this sensor is utilised as a NO contact, by doing this it will close when it is triggered by something on the belt conveyor. The next part of the rung is the memory bit, by utilising a normally closed contact and tagging it with the memory bit it means that if the assembly is satisfactory then it will break the signal and therefore the reject solenoid will not engage.

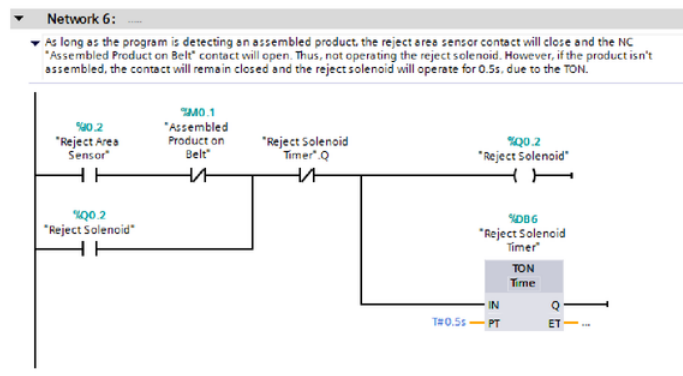


Figure 19 - Network 6

However, if the two contacts are in the correct state then the reject solenoid will be engaged and therefore push the product off the belt conveyor and into the reject box.



Figure 20 - Network 7

The following network (Figure 20) is utilised in order to reset the two memory bits; this is done so that the memory bits can be set again for the next product that is coming on the belt conveyor. In order to reset the bits a reset contact is utilised for both memory bits, however a negative edge contact is utilised before the memory bits are reset. This negative edge contact will engage when the signal from the reject area sensor is lost. This therefore means that the sensor must first of all come on and then go off for the contact to close and reset the memory bits. This particular network is the last of the networks needed for the trainer to operate however three more networks were utilised in order for the HMI (Human Machine Interface) screens.

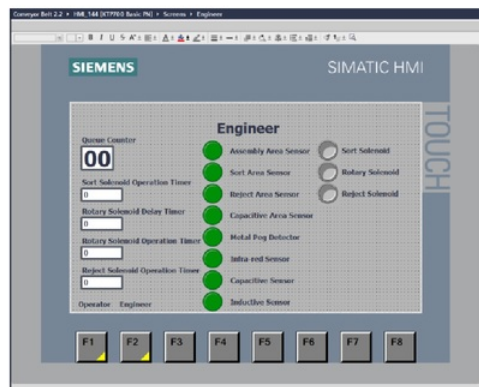


Figure 21 - The engineer HMI design

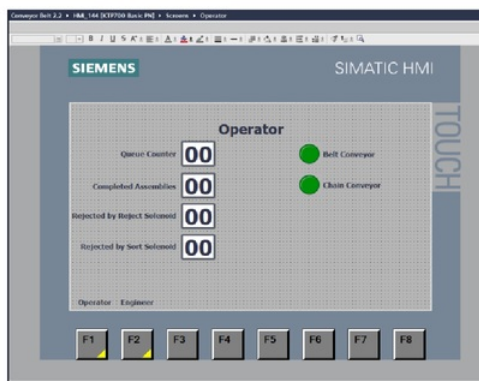


Figure 22 - The operator HMI design

Figure 21 & 22 show the two HMI screens that were produced for this trainer. There are two different trainers so that the relevant people are allowed to see the relevant information for their particular job. The operator screen (Figure 22) allows the operator to determine whether both conveyors are running, this is helpful as it allows him to ensure that there is no fault with the conveyors. There are also four different numerical values that the operator is allowed to see. The numerical values are;

- Queue counter value
- Total completed assemblies
- Total products rejected by the reject solenoid
- Total plastic pieces rejected by the sort solenoid

It is believed that for the operator to go about his daily job this information is completely sufficient.

Secondly, Figure 21 is the engineer HMI screen, the engineers daily work can see him conducting an array of different tasks that provide a lot of different details. It is therefore important that a comprehensive amount of details and information is provided to the engineer on the HMI screen. The numerical information that can be obtained from the engineer screen is as follows;

- Queue counter value
- The value of the sort solenoid operation timer

- The value of the rotary solenoid delay timer
- The value of the rotary solenoid operation timer
- The value of the reject solenoid operation timer

The engineer screen also has a visual representation of the current state (On or Off) of the following inputs;

- Assembly area sensor
- Sort area sensor
- Reject area sensor
- Capacitive area sensor
- Metal peg detector
- Infra-red sensor
- Capacitive sensor
- Inductive sensor

Alongside this information, the engineer also has remote control of all three of the solenoid from the engineer screen, this can be helpful in determining whether the fault is with the solenoids or something else.

It is believed that the array of different information and capabilities that the engineer has been given on this HMI screen should be sufficient in aiding the engineer with any possible job that they may come across on the trainer.

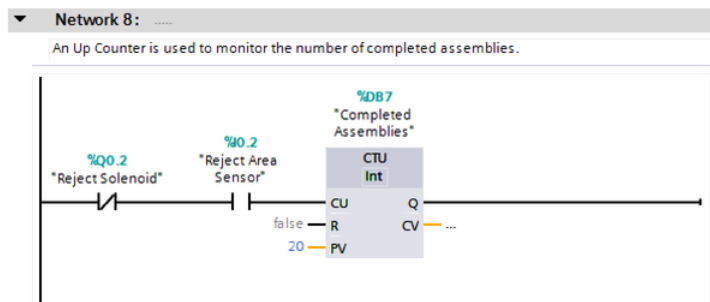


Figure 23 - Network 8

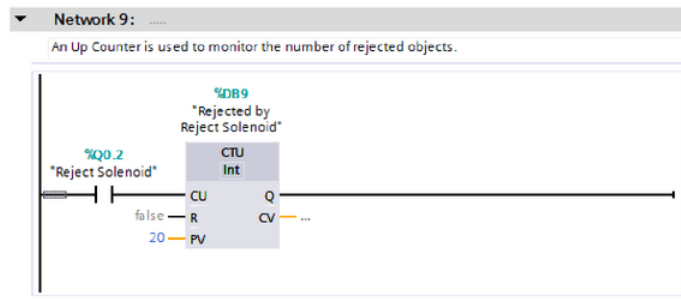


Figure 24 - Network 9

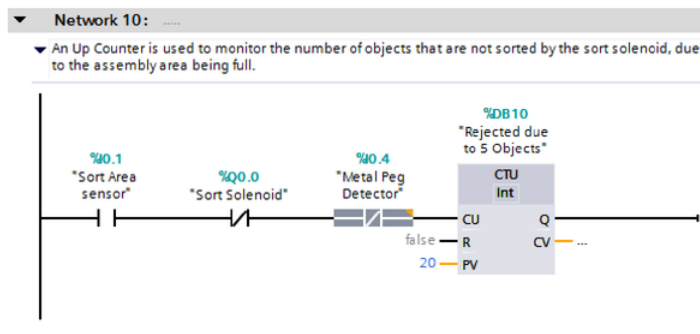


Figure 25 - Network 10

Figure 23, 24 & 25 depict networks 8, 9 and 10, each one of these networks were built in order to aid with certain parts of the HMLs. For example, Figure 23 was constructed to allow the operator to keep track of the amount of completed assemblies, this was achieved by adding one to the subtotal of the counter every time that a product went past the reject area sensor but did not get rejected by the reject solenoid.

Figure 24 is a counter that is utilised in order to identify the amount of products that are rejected by the reject solenoid. It works by adding one to the subtotal of the counter every time that the reject solenoid engages and therefore closes the NO contact.

Figure 25 is another counter that is utilised in order to keep track of any plastic pieces that are rejected on the chain conveyor due to the assembly chute being full. This is achieved by ensuring that a product is seen by the sort area sensor, but is not rejected by the sort solenoid and also not detected by the metal peg detector. Every time that this particular process occurs one is added to the rejected by 5 object counter.

4. System testing and evaluation

Two testing tables were developed for the actual testing of the program and HMI operations. It was decided that the testing table be split into two major parts, the Conveyor section and the HMI section. In all there were 32 tests to carry out, a description of each test an expected results and an actual result of each test can be seen in the two testing tables below.

Conveyor Test Table

Test Number	Test Description	Expected Result	Actual Result
1.	Switch the on button to the on position.	Both Conveyors should start.	Both conveyors started.
2.	Place plastic ring on Chain Conveyor.	Ring are pushed down assembly chute by Sort Solenoid.	The ring was pushed down the assembly chute by the sort solenoid.
3.	Metal peg on Chain Conveyor.	The metal pegs pass by and do not get pushed down the chute.	The metal peg was not pushed by the sort solenoid.
4.	Put 8 plastic rings on Chain Conveyor.	Only 5 should be pushed down the assembly chute (one plastic piece will be allowed down into the assembly area) and 3 should pass by.	6 plastic pieces were pushed down the assembly chute and 2 were rejected into the reject box at the end of the conveyor.
5.	Place ring on chain conveyor.	Rotary Solenoid should drop plastic ring into the Assembly Area where it will get picked up by a metal peg.	The plastic piece was pushed down the chute and then allowed to enter the assembly area by the rotary solenoid.
6.	Metal peg passes through Assembly Area.	Plastic ring should drop onto metal peg.	The plastic piece dropped onto the metal peg.
7.	Assembly passes Inductive Sensor.	Inductive Sensor detects metal peg.	The inductive sensor detected the peg
8.	Assembly passes Capacitive Sensor.	Capacitive Sensor detects completed assembly. Completed assembly also detected by Reject Area Sensor, allowing it to pass into the finished assembly area.	The assembled product was allowed to pass through to the completed zone.
9.	Metal Peg passes through sensors without plastic ring.	The Capacitive Sensor should detect that there is no plastic ring, triggering the Reject Solenoid pushing the peg off the conveyor.	Metal peg was rejected by the reject solenoid.

10.	Plastic ring passes through sensors without metal peg.	The Inductive Sensor should not detect the plastic ring passing. When the ring passes the Capacitive Sensor it is pushed off the conveyor by the Reject Solenoid.	Plastic ring was rejected by the reject solenoid.
11.	Assembled product passes through upside down.	Inductive Sensor does not detect the metal ring due to it being above the sensor, this then causes it to be kicked out when it passes the capacitive sensor.	The upside down assembled product was rejected by the reject solenoid.

HMI Testing Table

Test Number	Test Description	Expected Result	Actual Result
1.	Queue Counter reaches 5.	Queue counter on HMI counts up to five. After doing so it does not allow any more components in the assembly chute.	Queue counter on HMI counted up to 5.
2.	Completed assemblies.	The number of completed assemblies is counted by counting up when an object is detected by the Reject Area Sensor, without the Reject Solenoid energising.	Number of completed assemblies increased every time a successful assembly was completed.
3.	Rejected by Solenoid.	This function counts how many objects are rejected. This is done by counting up every time the Reject Solenoid activates.	Every time a assembly was rejected the counter increased by 1.
4.	Rejected by Sort Solenoid.	When assembly chute is full the Sort Solenoid counter on the HMI Operator display screen starts counting how many plastic rings pass by. This is done by the Sort Area Sensor detecting non-metal objects when the assembly chute is full.	Every time a plastic piece was not allowed down the chute, the counter went up by 1.
5.	Chain Conveyor indicator.	When the Chain Conveyor is running, the indicator on the HMI's Operator screen lights up.	The indication light came on.
6.	Belt Conveyor indicator.	When the Belt Conveyor is running, the indicator on the HMI's Operator screen lights up.	The indication light came on.
7.	Assembly Area Sensor indicator.	Assembly Area Sensor indicator comes on when Assembly Area Sensor detects object.	The indication light came on.

8.	Sort Area Sensor indicator.	Sort Area Sensor indicator comes on when Sort Area Sensor detects object.	The indication light came on.
9.	Reject Area Sensor indicator.	Reject Area Sensor indicator comes on when Reject area sensor detects object.	The indication light came on.
10.	Capacitive Area Sensor indicator.	Capacitive Area Sensor indicator comes on when Capacitive area sensor detects object.	The indication light came on.
11.	Metal Peg Detector indicator.	Metal Peg Detector indicator comes on when Metal peg detector detects object.	The indication light came on.
12.	Infra-red Sensor indicator.	Infra-red Sensor indicator comes on when Infra-red sensor detects object.	The indication light came on.
13.	Capacitive Sensor indicator.	Capacitive Sensor indicator comes on when Capacitive sensor detects object.	The indication light came on.
14.	Inductive Sensor indicator.	Inductive Sensor indicator comes on when Inductive sensor detects object.	The indication light came on.
15.	Sort Solenoid operation timer.	Sort Solenoid operation timer counts correctly.	The timer counted incorrectly, as it only showed the digit '1'.
16.	Rotary Solenoid delay timer.	Rotary Solenoid delay timer counts correctly.	The timer counted correctly.
17.	Rotary Solenoid operation timer.	Rotary Solenoid operation timer counts correctly.	The timer counted incorrectly, as it only showed the digit '1'.
18.	Reject Solenoid operation timer.	Reject Solenoid operation timer counts correctly.	The timer counted incorrectly, as it only showed the digit '1'.
19.	Sort Solenoid function test.	When the button on HMI is pressed, solenoid activates.	The solenoid activated.
20.	Rotary Solenoid test button.	When the button on HMI is pressed Solenoid activates.	The solenoid activated.
21.	Reject Solenoid Test button.	When the button on HMI is pressed Solenoid activates.	The solenoid activated.

After carrying out all the tests that were decided upon, it can be evaluated that although 3 tests found slight malfunctions with the timers on the HMI, and one test result was satisfactory but slightly different to the expected result, that the overall program and HMIs operation was of good quality.

At no point whilst testing was there any need to conduct a test an unreasonable amount of time to ensure that it definitely did meet specification, there may have been a need for this if a test was satisfactory at one point but when testing again was un-satisfactory. There was no need for this because almost every test that was carried out, met the expected result. It was ensured that each test was carried out twice to ensure that the first outcome was not down to complete chance. This also ensured that full confidence could be had when writing the test results into the test table. This is important as it ensures that a comprehensive quality control test was carried out and that confidence can be had in the results of the test, and that the results are valid.

The three tests that did not meet the expected result were that of the operation timers of all three solenoids. The operation counters of the three solenoids are represented on the engineer's screen. It was found that because the timers were counting up to half a second the timers on the engineer screen only represented the digit 1 and then returned to 0. This is believed to be because either the layout of the counter utilised within the HMI or down to the fact that the timer is of such short period that the counter was not able to keep up.

Another test result did not meet the expected result however it was not concluded to be a failure; this is because it when the expected result was noted the programs working was slightly overlooked. The test was to ensure that when 8 plastic pieces were put onto the chain conveyor only 5 of them were pushed down the assembly chute by the sort solenoid. However, whilst testing it was found that 6 plastic pieces were pushed down the assembly chute and only 2 were rejected. After having a quick reflection on why this could be it was remembered that one of the plastic pieces would actually be pushed down into the assembly area which therefore provided the assembly chute with one more space for another plastic piece. This therefore meant that the sort solenoid would be engaged again and push another plastic piece down the chute due to the queue counters subtotal only equalling the value of 4.

5. Conclusions and recommendations

Conclusion

The brief states that a metal peg must be allowed past the sort area and down the feeder chute and onto the belt conveyor. In conclusion this was achieved within the program by ensuring that the sort solenoid only operated when the sort area sensor was triggered but the metal peg detector was not. This therefore meant that every time a metal peg went past the peg detector would pick it up and therefore stop the solenoid from engaging and allow it to pass.

The brief stated that the sort solenoid must push the plastic pieces into the assembly chute ready for assembly. In conclusion this was achieved within the program by ensuring that plastic components were distinguished by using the metal peg detector and the sort area sensor together a counter was also introduced to ensure that a maximum of 5 plastic pieces were in the chute at any given time. This therefore meant that when the metal peg detector didn't see anything but the sort area sensor did the component must be plastic and was then pushed down the chute by the solenoid as long as the counters value was less than 5.

When a metal peg passes the assembly area a plastic piece must be placed on top of it. in conclusion this was achieved by ensuring that one plastic piece is always in the assembly area by utilising a rotary solenoid. This rotary solenoid was operated in the program whenever the assembly area sensor was not triggered.

If a fault was to occur on the assembly and a metal peg was allowed to progress past the assembly area without having a plastic piece upon it, it must be rejected. In conclusion, this was met by using an assortment of different sensors and if the signals of the sensors did not meet the requirements the assembly was rejected by the reject solenoid. This was achieved within the program by ensuring that the final product had a metallic and plastic part to it and that they were of correct orientation, if these requirements were not met then the product was rejected by the rejected solenoid before the completed zone.

A suitable HMI that can be utilised for diagnostic use must be produced. In conclusion, this was met by producing two different HMIs, for use by two different workers at a production industry. These HMIs provided the two different workers (operators and engineers) with all the relevant information for them to carry out their jobs. The HMIs allowed them to keep track of sensors by showing if their on or not, it also allowed remote control of solenoids and finally allowed a numerical representation of rejected components, completed assemblies and the queue counter.

Recommendations

In order to address the problem with the timers on the HMIs, the layout of the counters utilised upon the HMI should be changed in order to allow decimals to be represented because the timers only go up to half a second and therefore does not become a whole number.

In order to address the problem of extensive amounts of metal pegs being rejected, another solenoid could be introduced. This solenoid would prevent the metal pegs from traveling into the assembly area without a plastic piece being present, by doing this the amount of metal pegs being rejected and therefore wasted is decreased and therefore increases the productivity of the trainer.

Having looked at both HMIs in detail it is believed that 2 additional features could be made. It is believed that the engineer should have remote control of the queue counter so therefore be able to change the value of the queue counter from the HMI, and also it is believed to be desirable for the operator to have a emergency stop button upon his HMI for quick reactions to a dangerous situation.

6. Collaboration

There were many aspects to this task where team work was needed, as a team and as an individual it is believed that we performed well together and were therefore successful in producing a pleasing program. Whilst creating the program all 3 members of the team alternated in the programming, this allowed all team members to ensure that our programming skills were up to scratch and also distributed the work fairly.

When producing the I/O list we all work as a team, one of the team members had a piece of paper ready to write down the I/Os, the other team member was manually triggering I/O, and as the team member was manually making a I/O, the final team member then called out which I/O that particular sensor/solenoid was connected to. The first team member was then able to write down all the I/O and therefore produce the I/O list for the program.

Whilst creating the testing table the whole team collaborated and discussed the most important tests that were needed to be carried out in order for us to have complete confidence in saying that the program works. This was achieved by discussing the most important tests and then noting the ones that we decided were most important.

Whenever a team member was programming other team members ensured to be assisting them, most of the time it would be as simple as instructing them of the next contact needed upon a rung, however by doing this it allowed the programmer to keep their thoughts on the program and therefore not make any simple mistakes.

As a team member a list of things that were believed desirable upon the HMIs was produced, and after consulting heavily as a team on the points the team then proceeded to produce both HMIs with the desired points upon them. Once again the team ensured to rotate the team member that was doing the coding of the HMIs to ensure that all team members were aware of what was going to be on the HMIs and how they worked. The initial points that were drew up can be seen below;

Operator

- Completed assemblies
- Rejected due to 6 rings
- Rejected by solenoid
- Queue counter value
- Belt and Chain running indication

Engineer

- Queue counter
- Allow operation of all three solenoids
- Indication of all inputs
- Representation of all timers

When testing we once again worked as a team, one team member was doing the physical tests on the conveyor, and one team member was next to that team member in order to tell them the tests to carry out and also noting whether a test was satisfactory. The other team member was at the HMI in order to ensure that it was operating as it should whilst carrying out the different tests. The first team member would then carry out a test 2 times wait for the thumbs up from both team members and then move on to the next test.

7. Citation list

Works Cited

Anon., 2013. *PLCedge*. [Online]

Available at: <http://www.plcedge.com/plc-programming.html>

[Accessed 26 November 2017].

Anon., 2015. *UKessays*. [Online]

Available at: <https://www.ukessays.com/essays/information-technology/an-introduction-to-the-plc-information-technology-essay.php>

[Accessed 24 November 2017].

Anon., n.d. *BECKHOFF*. [Online]

Available at:

https://infosys.beckhoff.com/english.php?content=../content/1033/tcquickstart/html/tcquickstart_iec.htm&id=

[Accessed 26 November 2017].

Anon., n.d. *Simulink*. [Online]

Available at: <https://uk.mathworks.com/products/sl-plc-coder/features.html>

[Accessed 26 November 2017].

Anon., n.d. *Wisdomjobs*. [Online]

Available at: <https://www.wisdomjobs.com/e-university/programmable-logic-controllers-tutorial-523/instruction-lists-14693.html>

[Accessed 26 November 2017].

Budimir, M., 2017. *MCT*. [Online]

Available at: <http://www.motioncontroltips.com/sequential-function-charts-sfcs-plcs/>

[Accessed 27 November 2017].

Cooper, M., n.d. *ReviewsYouTrust*. [Online]

Available at: <https://www.reviewsyoutrust.co.uk/siemens-6es7211-1ae31-0xb0-simatic-s7-1200-cpu-1211c-compact-cpu-review/>

[Accessed 22 11 2017].

Gilbert, C., 2014. *Idaho State University*. [Online]

Available at: <https://blog.cetrain.isu.edu/blog/bid/353287/Advantages-of-a-Modular-PLC-Over-a-Fixed-PLC>

[Accessed 24 November 2017].

Hechtman, S., 2014. *Computing without boundaries*. [Online]

Available at: <http://computingwithoutboundaries.blogspot.co.uk/2013/07/sequential-function-charts-in-ignition.html>

[Accessed 26 November 2017].

Lachner, L., 2009. *Techwalla*. [Online]

Available at: <https://www.techwalla.com/articles/five-types-of-plc-language>

[Accessed 20 November 2017].

Peter, 2015. *PLC Programming & Automation Online*. [Online]

Available at: <http://www.plcacademy.com/ladder-logic-examples/>

[Accessed 26 November 2017].

Assignment 1 PSDE 4005

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

78/100

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

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

GRADING FORM: CSG-RUBIC

THOMAS WOOD

78

RESEARCH-INFORMED LITERATURE



Section 1: Your research - Very good evidence of reading. Further marks were available for further performance data and more detailed explanations of applications. (11/15) Section 8: Your use of Harvard for citing and listing - Comprehensive range of sources listed and cited correctly. (4/5)

15

KNOWLEDGE AND UNDERSTANDING



Section 2: Hardware Description and I/O list - Excellent explanations given of PLC input/output devices with a good I/O list. More marks were available for a fuller description of the actual PLC

17

used and the HMI screen hardware itself and some of your descriptions are unnecessary. (8/10)

Section 3: Program Explanation – A very detailed explanation of the program networks and screen explanations given. However, this section has been written in the style of a journal whereas you should focus your comments on how it works, not how or why you developed it. (9/10)

ANALYSIS



Section 4: System Testing and Evaluation - Evidence of thorough testing with a very detailed table and evaluation. Further marks were available where testing is supported by photographic evidence. (8/10) Section 5: Conclusions and Recommendations - Excellent work (10/10)

18

PRACTICAL APPLICATION AND DEP.



Section 3: Program Design and Network Comments - A well designed program with additional diagnostic counters and good in-program comments. (7/10) Section 3: HMI Screen Designs – Good screen designs. (7/10) Section 4: Presentation of Test Results - A detailed testing table but no photographic evidence to support it. (5/10)

19

SKILLS FOR PROFESSIONAL PRAC.



All sections - A well structured, presented and generally well written report. (4/5) Section 6: Team working - Excellent explanation given (5/5)

9