

Chris James PLC assignment

by Chris James

Submission date: 01-Jan-2025 06:14PM (UTC+0000)

Submission ID: 246987144

File name: 50113_Chris_James_Chris_James_PLC_assignment_629994_1516801865.docx (34.32M)

Word count: 4424

Character count: 21291

Literature review

Abstract

This literature review aims at looking at the different types of plc's that are currently available worldwide. I will look at the pros and cons of multiple different examples and explore where they are most commonly used.

PLC's (programmable logic controllers) are vital in today's process manufacturing and automation. They allow the user to have a much greater control over the process. Programmers are able to create inputs, outputs, flags, timers and other types of control to allow different functions to occur. Ephrem Ryan Alphonsus describes a plc as 'a device that can efficiently control many different types of equipment which can include entire automated systems.' (**Alphonsus, 2016**). Due to the flexibility of a plc. capabilities, they are now found in many different processes with automation today.

Types of PLC's available

The most popular plc manufacturers in today's automation are Siemens and Allen Bradley (**Peko, 2017**), however many other types of plc's are also available. Different parts of the world may have different preferences over the types of PLC's that they prefer. For example, American companies often prefer to use Allen Bradley (an American brand) (**Cheng, 2024**) where as in Europe Siemens is often preferred. (**Cheng, 2024**) This is often due to availability and cost rather than functionality.

Each PLC will use its own software to allow the programmer to create the programs. Most modern programs will also include the options to add HMIs (Human machine interface) to the programs within the same program. This can help to make programming much easier as you are able to program both in the same software instead of trying to communicate between different programs.

Brands of plc include:

- Siemens – S1200 and S1500. Widely used across the world. Popular for their easy user interface and flexibility. Both have good processing speeds and multiple different potential methods of communication (ethernet, Profibus, profinet). The majority of PLCs that Siemens provide are modular. This means that they can add multiple modules as required to expand its capabilities. (**Kishorre, 2019**)
For smaller projects, a plc such as the s1200 range may be suitable. For larger projects, they offer the 1500 range. Each can add different pros and cons to a project such as memory, processing speed, size etc.
Siemens also offers a much smaller type of PLC called the siemens logo. This plc is programmed using logosoft and offers the most basic of features. It is a static plc, meaning that unlike the modular versions, it cannot be expanded. They often only allow a small number of inputs and outputs so are only suitable for very small projects. The siemens logo is most often only suitable for small projects or educational use.
- Allen Bradley – Part of Rockwell automation, offers compactlogix and controllogix. Both widely used in many different types of industry. Both have a good processing speed and a good user interface.
Allen Bradley products are very often more expensive than Siemens products. This can be an issue, especially for smaller companies that are working on a budget. (**Wright, 2020**) Even though they are more costly, Rockwell does offer a good customer support base which can

be useful if you have any issues with your software. They also offer compatibility with many other products such as servo drives/motors etc.

- Mitsubishi – Good for small to medium sized project but struggle with larger programs. Compact but cost effective while powerful enough to handle basic tasks.
- Omron – A less popular option compared to others but can still offer high performance. They use their own software which is not considered as user friendly as some of the others.

The following graph shows that each type of PLC has its own pros and cons. Research would have to be carried out to determine the appropriate PLC for your project such as project size, cost and space available.

Types of plc	Pros	Cons
Siemens S7-1200	• Compact design suitable for small to medium projects. • Compatible with TIA portal • Good quality and reliable • Supports multiple different communication methods.	• Limited IO's compared to some others. Additional modules may be required • Limited scalability for larger programmes. More suited for small to mid sized programs. • Can be more expensive than some other types of plc's
Siemens S7-1500	• High processing power, suitable for large and complex systems. • Upgradeable and great for large scale programs. • Supports multiple different communication methods.	• Overkill for smaller projects. • Can take up a large amount of panel room • More expensive than other types of PLC
Allen Bradley controllogix	• Reliable and Versatile • Modular to allow for larger scale projects • Good support network with Rockwell	• Software licenses make these an expensive choice • Expensive PLCs, new modules can also be costly.
Allen Bradley Compactlogix	• Good plc for small to medium sized projects • Good software to use for programming • Small panel footprint	• Limited expansion possibilities. • Not as powerful as alternatives such as the controllogix
Mitsubishi Q series	• Reliable brand of plc • Flexible and modular, high range of addons available.	• Programming software is not as advanced as allen bradleys or siemens • Limited global support compared to others.

PLC hardware

All PLCs are compatible with 3 main types of logic. These are Ladder, FBD (function block) and STL (Statement list). Ladder logic is the most popular due to its linear programming methods and ease of understanding. For a PLC to function it will require, a power supply, rack mounts, programming software, input/output modules and a CPU.

Conclusion

Programmable logic controllers are indispensable in modern day process manufacturing. They offer a new level of flexibility and control over machinery that was never previously possible. Each type of PLC comes with its own unique strengths and weaknesses. Considerations into such things as cost and performance must be

taken into account when deciding on your PLC for your project. Siemens and Allen Bradley are both considered industry leaders in the field and are known for their performance and user-friendly interfaces. Whereas Siemens will provide modular systems such as the S1200 and the S1500, Allen Bradley will also offer their own alternative in the compact logix and control logix. Both are supported by a useful technical helpline, however in my experience I have found that Allen Bradley comes with a much higher cost.

Other brands such as Omron and Mitsubishi are less popular but can also be more cost-effective options for companies on a budget.

Selecting a PLC means considering project size, budget, space and compatibility with current systems. The logic used, either ladder, FBD, or STL is dependent on the user/programmers' skills. While industries continue to embrace this technology, it will continue to play an increasingly important role in the future of manufacturing and other industries.

The choice of PLC is often reflected by a balance of performance, cost, support network and expansion, ensuring high results for the specific automation requirements.

Factory IO

To create my factory layout simulation, I used FactoryIO. With this software I am able to connect directly to TIA portal and carry out a real time simulation of my project.

My Simulation contained the following:

- Conveyors
- Sensors
- Pushers
- Weighting scales
- Spur Conveyor
- Stack lights
- Siren
- Warning lights
- Counters
- E-stop
- Push buttons

The process is connected to TIA portal and also uses an HMI simulator. The HMI will make the operator aware of any warnings or alarms that will help to diagnose any issues within the system. It will also allow them to reset alarms, reset counters and view the number of boxes processed. An E-stop is added to the main HMI screen so a user can stop the system in an emergency. Replica stack lights have also been created to allow the operator to view the process status via the HMI.

FIGURE 1 - SIMULATION OVERVIEW

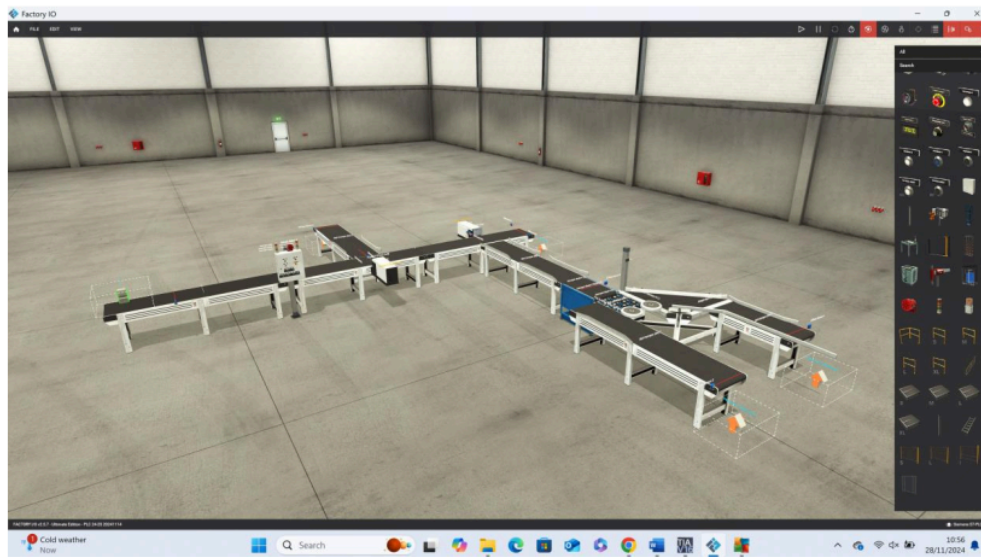


Figure 1 shows an overview of the factoryIO simulation. For this process to begin, the control panel is used. When the start button is pressed, the boxes will begin to move down the conveyor from the emitter.

FIGURE 2 - CONTROL PANEL



The panel contains 3 push buttons, an E-stop, 6 counters, 2 stack lights and a siren.

When the start button is pressed, it will send a signal to the plc to begin operations. As long as all other conditions are met, the plc will begin the startup sequence.

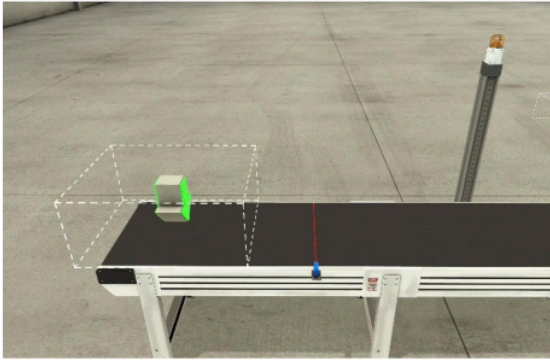
When the sequence begins, the alarm will sound, and the conveyors will begin in sequential order.

In case of emergencies, an E-stop can be pressed. This will stop the whole system, also sending an error message to the HMI. This will require a manual reset on the engineering screen to allow the process to restart. This is done to prevent the system from automatically restarting once the e-stop has been reset.

The 6 counters on the panel all indicate different values. 4 counters will count the number of boxes of each size that has been processed. For each processed box, the corresponding counter will count up by one each time, this is programmed in the PLC. The counter values can be reset at any point by pressing with the reset counter pushbutton on the control panel or alternatively, the reset counter button on the HMI.

The final 2 counters will show a value for the total amount of boxes that have been processed, and the amount of boxes that are remaining in the system. This value is achieved by subtracting the overall amount of boxes that have been processed by the total amount of boxes that has been put into the system.

FIGURE 3 – BEGINNING OF PROCESS



The boxes are inserted into the system using a feature of FactoryIO called the emitter. This will insert random boxes to be sorted at regular intervals (figure 3).

The following sensor will then count the number of boxes that are being introduced into the system. This information is then displayed on the main panel and the HMI (via the PLC). The warning light (pictured) activates whenever the system starts. This activates at the same time as the siren (figure 2)

When starting the process, each conveyor will start sequentially. The 1st conveyor will start after 2 seconds (once the siren and warning lights have finished. This makes everyone aware that the machine is about to start). Each other conveyor will then start in 0.5second intervals. This will help to minimise the initial startup current.

FIGURE 4 - LARGE BOX PUSHER



The large box pusher (figure 4) will push any box that has been detected by the sensor. These boxes are pushed onto a new conveyor that will take the large boxes to their final destination. When they reach the end of this conveyor, another sensor is situated to count the amount of large boxes that have been processed.

The sensor is installed at a height where it will only detect large boxes. All other boxes will pass underneath it. This means that the pusher will only activate at the correct times.

When the sensor has detected a large box, a delay of 430ms has been added into the PLC program before the output is given to activate the pusher. This gives the box enough time to move past the conveyor and into the line of the pusher.

Although I could have used a signal from the pusher to count the amount of large boxes being processed (count up every time the pusher is activated), by adding a sensor on the end of the large box conveyor, I am able to keep track of all of the boxes that are currently in the system and not just what is on the main conveyor. This also standardises the counters as the medium, small and pallet counters are also designed in a similar way.

FIGURE 5 - MEDIUM AND SMALL BOX PUSHER

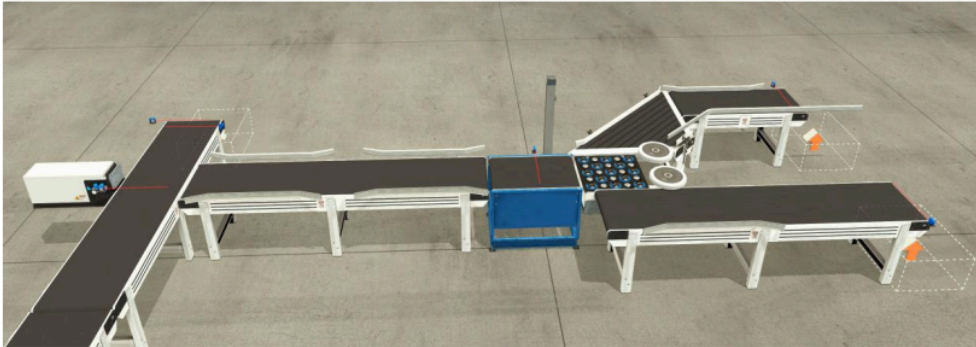
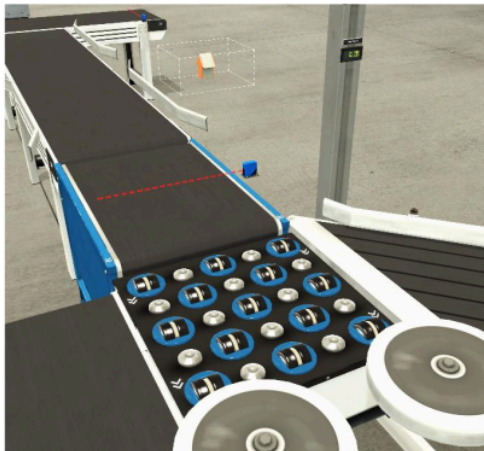


Figure 5 shows that the same pusher is used to push medium and small boxes off the main conveyor. This is because they are both the same height and can only be differentiated by weight.

When the sensor detects either a small or medium box, a delay of 800ms is programmed into the PLC before the signal is sent to the pusher to activate. Again, this gives the box enough time to line up with the pusher so that it can accurately push the box onto the correct conveyor.

This sensor is set in a position that will only detect the medium and small boxes. All pallet boxes will pass underneath it, preventing a signal from being sent to the PLC.

FIGURE 6 - MEDIUM AND SMALL BOXES



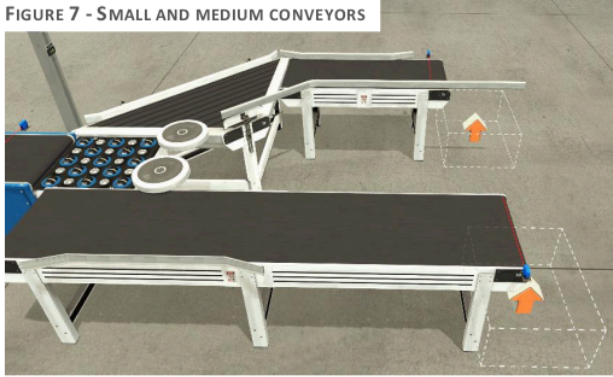
As the medium and small boxes move up the new conveyor, they will move onto the weighing scales (Figure 6).

These weighing scales will send an analog signal to the PLC which will determine the weight of the box. Since small boxes weigh 4kg and medium boxes weigh 5kg. I have set the scales so that anything less than 4.7kg is determined as a small box. Anything greater than this value is determined as a large box.

The wheels on the pop-up roller (pictured) will then turn towards either the left or right conveyor depending on the type of box that has been processed. A medium box will move towards the left conveyor while a small box will move towards the right conveyor.

A weight display is found next to the weight conveyor. This allows an operator to see the value of each box as they are being processed. This is useful when carrying out quality/process checks.

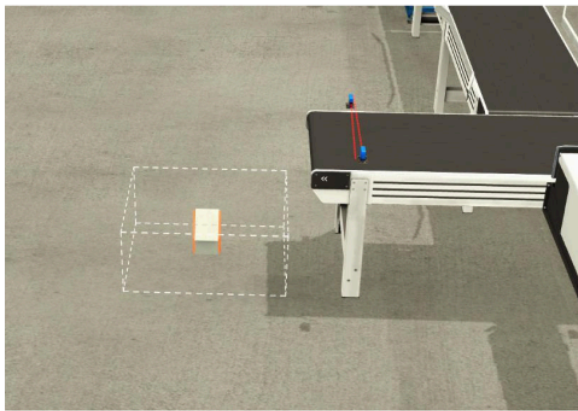
FIGURE 7 - SMALL AND MEDIUM CONVEYORS



As the boxes are moving towards their final positions, a sensor on each conveyor will detect the amount of small and medium boxes. This will allow us to count the amount of small and medium boxes that have been processed.

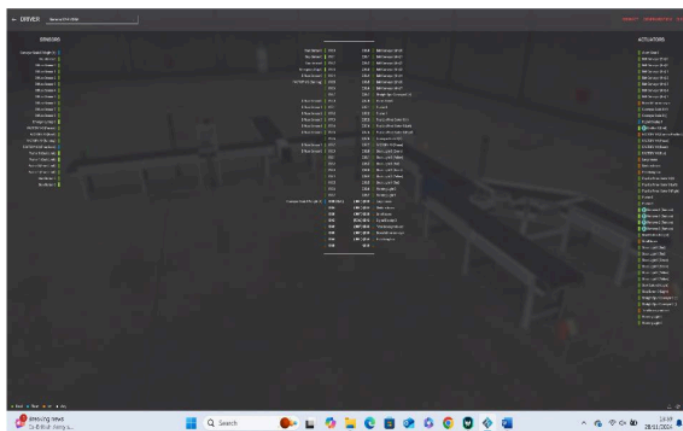
The large rollers that are seen by the pop-up wheel sorter (figure 7) are in place to ensure that the boxes move to the new conveyors smoothly and do not fall off the conveyor.

FIGURE 8 - PALLETISING BOXES



The only boxes that will be left on the main conveyor once all other boxes have been processed are the palletising boxes. There are 2 sensors in this position. One of the sensors will detect the palletising boxes and send a signal to the counter. The top sensor is in place as a failsafe. It is in place to detect any type of box that is NOT a palletising box. If this sensor detects anything then it will send a signal to pause the system, and an alarm will show on the HMI advising the operator to check the area.

FIGURE 9 - DRIVER SCREEN



Each input and output are assigned its own location on the driver screen. This has been setup to simulate a Siemens PLC in the setup screen. This PLC simulator contains both digital and analog inputs. This allowed me to send Boolean signals from sensors as well as send and receive analog signals from such components as weighing scales that are using analog values. A Boolean signal will

only have 2 values – on or off whereas an analog signal may have any value between 2 set limits (e.g. 0-5v or 0-10mA).

PLC Program

To create the PLC program, I used TIA portal V16. Currently, up to V20 is available, however using older versions does have advantages. If this program was created on V20, then it would only be able to be used on machines that has V20 installed. Since v20 has only recently been released, very few companies/machines have an up-to-date version. A program created using V16 can still be read/monitored on higher versions.

For the purposes of this project the program consisted of multiple networks, all under the block OB1 (Main program). To make the program easier to read, we may break the program down into sub-routines (or sub-folders). This would allow us to have different parts of the program in different areas. For example, a sub routine may just contain the alarms, this means that if a user wanted to look at the alarms, then they can find them all in one area. This may also be a DB block (database). This would make it easier to refer to the alarms when trying to read them from external sources such as an HMI.

To create the program, I made use of flags, inputs, outputs, timers and counters. An input is active when the program receives a signal from an external source onto one of its input terminals (E.g. I0.1, I0.2 etc.), An output is a signal that is sent to a specific output terminal (E.g. Q0.1, Q0.2 etc.) once certain conditions in a rung are met.

Counter blocks will allow the user to store integer values and add to them every time a desired signal is received. This also has a reset function so that it can be reset to 0 whenever the required input is received.

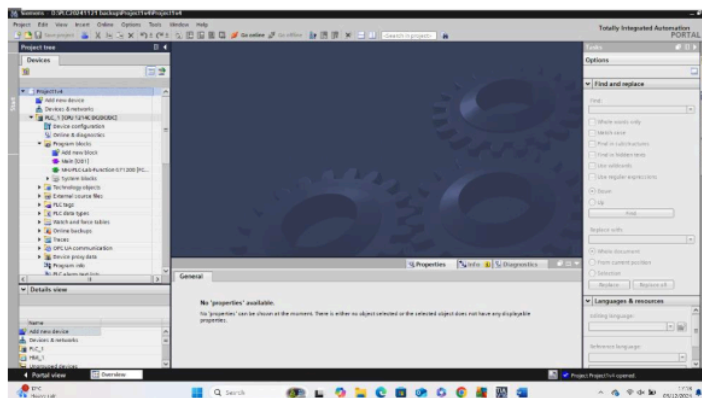
A flag is used to store information within the program. They can act as an 'internal output' that can be called upon at any point in the program. It will always be in one of 2 states – of or on.

Latching circuits were also used within the program. By adding a normally open contact of an output in parallel with a push button, it will hold the output on when the push button has been released. A separate input can then be used to reset this latch. This is how I created the operate function in network 2 (figure 11)

There are multiple different types of timers that are available in TIA portal such as TON and TOFF (timer on and timer off). Timer on will only allow an output after it has received an input for a set amount of time. This is how I created the on delay for the startup on the conveyor (Figure 13).

My program is broken down as follows:

FIGURE 10 - MAIN SCREEN



The program was created in TIA portal. Figure 10 shows the main screen. The program is broken down to the left of the screen, with option to view either the HMI or the PLC program.

When the PLC program is selected, it shows in the centre of the screen. These windows can be modified/moved to suit the user.

FIGURE 11 - NETWORKS 1 AND 2

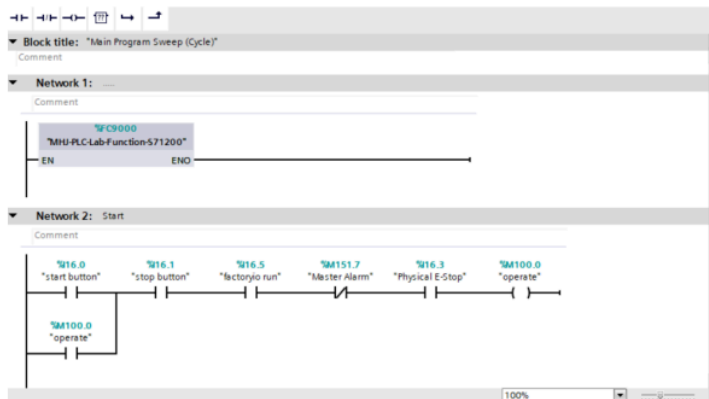
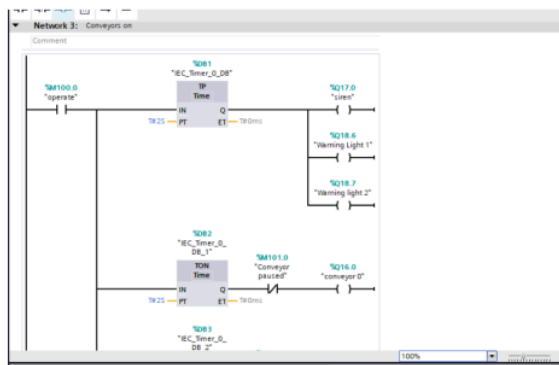


Figure 11 Network 1 – This is the block that must be inserted to any program that you wish to run using Factoryio. If this program was to be simulated elsewhere, or placed into an actual plc, then this block could be removed.

Network 2 – Creates the 'operate' flag. All contacts must be in the correct position for the flag to be made. Once the start button

is pressed, and all other conditions are met, then the operate flag will be activated. This will be latched on using a normally open contact that is in parallel with the start button. If any of the other conditions in the rung change, then it will disengage the operate flag. Due to the programming in the rest of the program, this will stop the program.

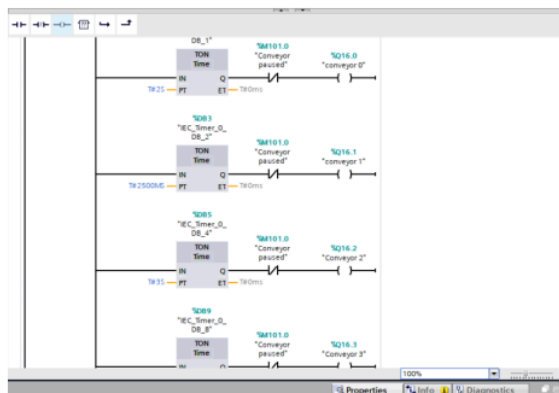
FIGURE 12 - NETWORK 3



Figures 12 – 15 show network 3. This network sets all conveyors to begin in sequence 500ms apart. In a real-world scenario, this would limit the startup current and prevent excess strain to a circuit.

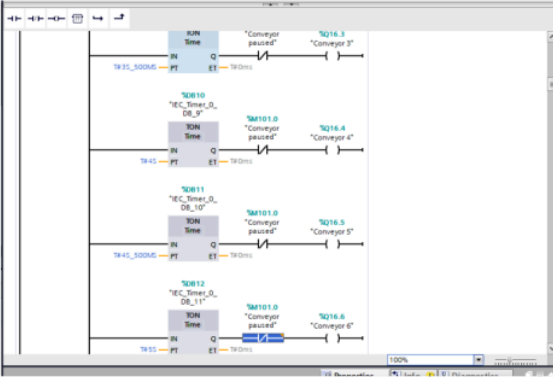
Before any conveyors start, a siren and 2 warning lights are activated to warn operators of the startup procedure. This activates for 500ms before the conveyor sequence begins.

FIGURE 13 - NETWORK 3



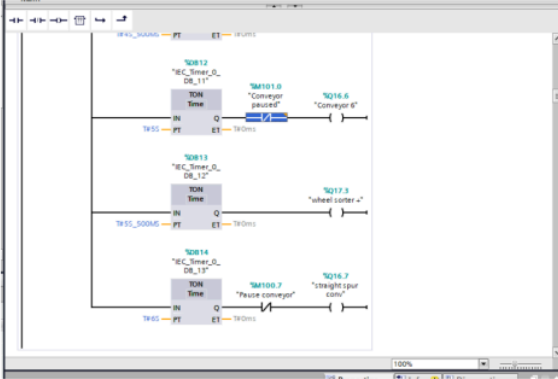
When the conveyors are running, the pause flag will stop the conveyors if there are any issues in the system that may require attention. Once these issues have been rectified, the system will then start back up.

FIGURE 14 - NETWORK 3



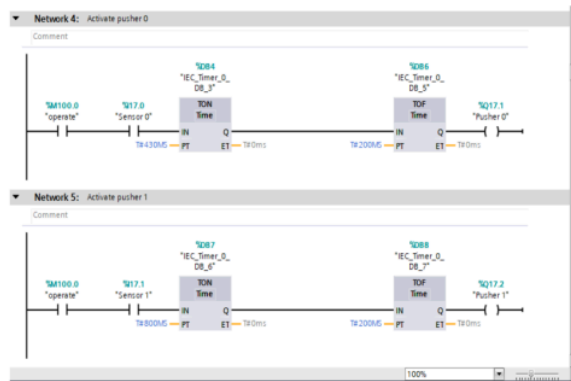
The sirens and warning lights use a 500ms pulse block. This will cause the sirens to pulse on and off. Each conveyor 500ms delay is activated using a TON block. This is an on delay that will only switch on a set period after receiving a signal.

FIGURE 15 - NETWORK 3



The wheel sorter and spur conveyor were also added into the conveyor startup sequence.

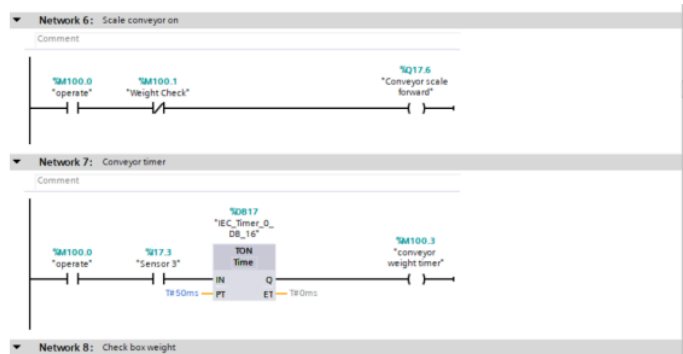
FIGURE 16 - NETWORK 4



Network 4 shows the conditions for pusher 0 to activate (large box pusher). This shows that the pusher will activate 430ms after the sensor has received a signal. After activating for 200ms, it will retract. This process will continue for every time it receives a signal from a large box.

Network 5 shows the same process but for the medium and small boxes.

FIGURE 17 - SCALE CONVEYORS

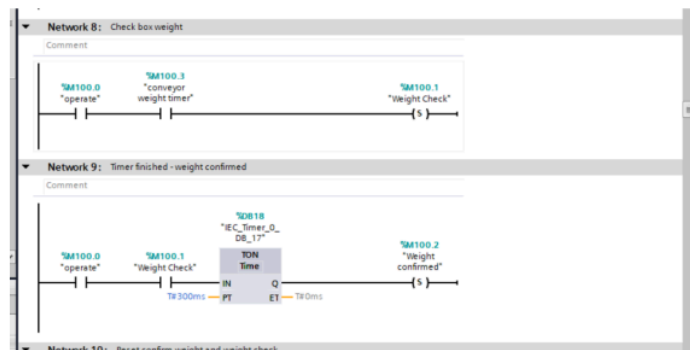


The conveyor scale (Network 6) will send the 'forward' signal as long as there is no weight check being carried out. If there is a signal from the weight check flag, then the conveyor will stop to allow the conveyor to measure the weight.

The Conveyor weight timer will begin 50ms after the sensor

has received a signal (network 7).

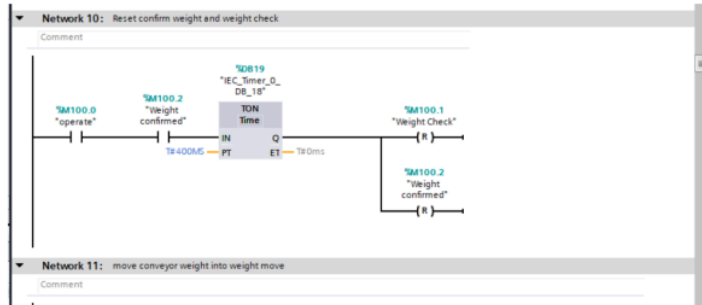
FIGURE 18 - WEIGHT CHECK



Once the weight timer has completed, the weight check flag is set (this stops the conveyor in network 4).

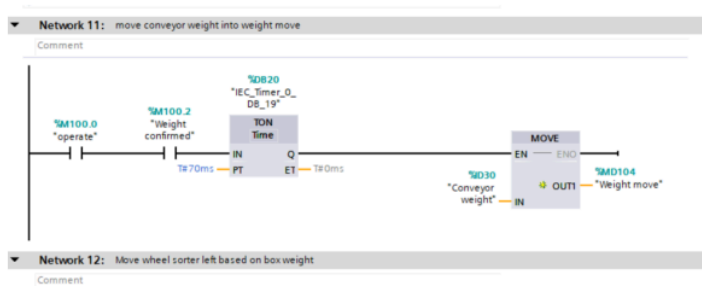
With the weight check set, the value can be taken for the weight of the box. This sets the weight confirmed flag.

FIGURE 19 - RESET FLAGS



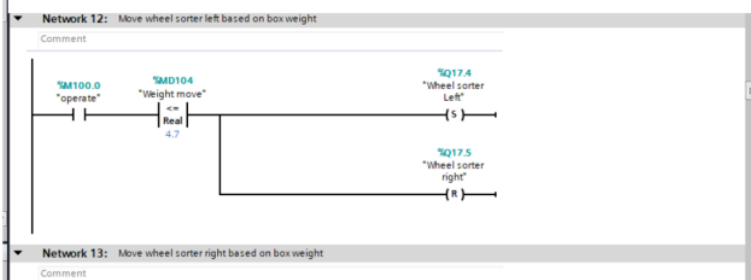
Both the weight check and weight confirmed flags are reset 400ms after the weight confirmed flag has been activated (figure 19).

FIGURE 20 - NETWORK 11, WEIGHT VALUE



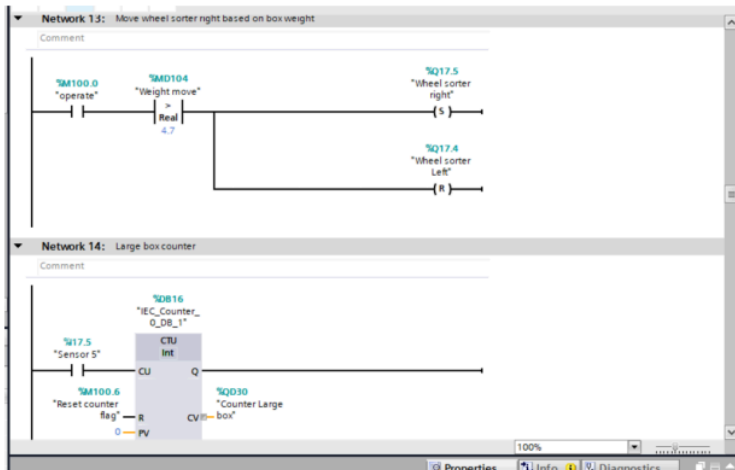
Using the weight confirmed flag, a delay of 70ms is added before the conveyor weight value is moved into an MD block. This value can be used in external features such as HMI's.

FIGURE 21 - WHEEL SORTER



If the value of MD104 is less than or equal to 4.7, then it recognises it as a small box. This will set wheel sorter left and reset wheel sorter right. This will move the boxes to the small box conveyor.

FIGURE 22 - NETWORKS 13 AND 14



flag.

Alternatively, if MD104 value is greater than 4.7, then it will reset the left sorter and set the right wheel sorter. (Network 13)

This will move the boxes to the boxes to the medium conveyor.

Network 14 shows that every time sensor 5 is made, it will use a count up block (CTU) to count up the number of large boxes processed. This value can be reset using the reset counter

FIGURE 23 - MEDIUM BOX COUNT



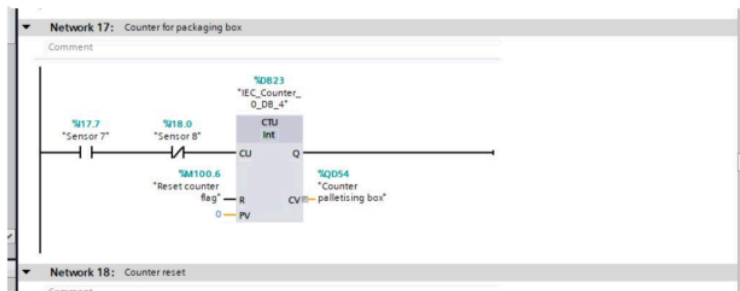
The same process is carried out to detect the amount of medium boxes processed. This sensor is found at the end of the medium box conveyor.

FIGURE 24 - SMALL BOX COUNTER



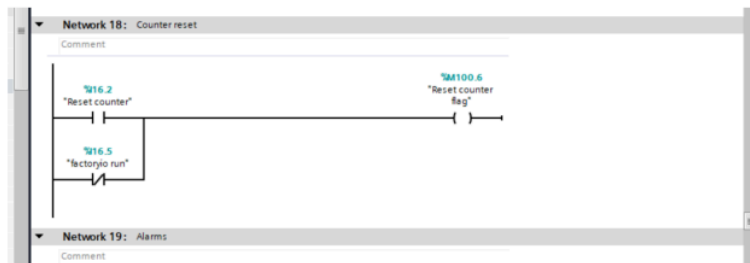
Same process is carried out to measure the amount of small boxes processed. This value is reset using the reset counter flag.

FIGURE 25 - PALLETISING BOX COUNTER



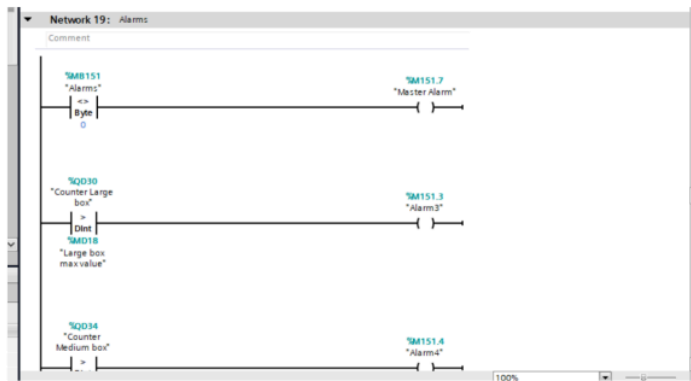
The same process is carried out for the palletising boxes however this also requires no signal from sensor 8. This sensor is in place to ensure that only palletising boxes are processed in this area.

FIGURE 26 - RESET COUNTERS



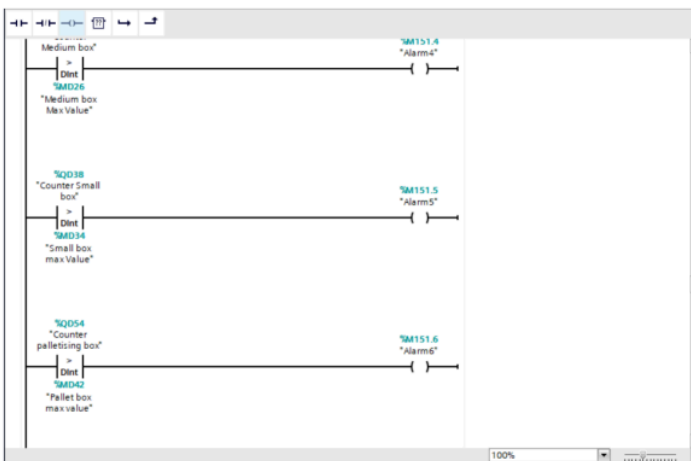
There are 2 conditions that will reset the counters, if the input signal is received from the main panel or if the Factoryio program is stopped (Figure 26).

FIGURE 27 - NETWORK 19



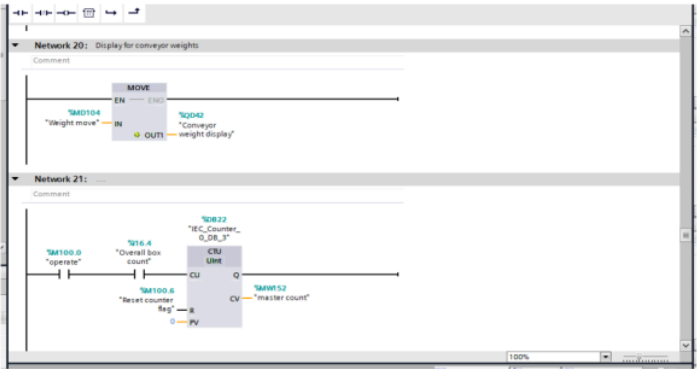
Network 19 (figure 27) contains all of the alarms that can be found on the HMI page. If any alarms are activated, the master alarm will also activate. This is due to the contact looking from any signal from the MB151 byte.

FIGURE 28 - NETWORK 19



The alarms are activated for various reasons. Once the maximum number of boxes has been reached (set on the HMI) then the relevant alarm will activate to make the users aware.

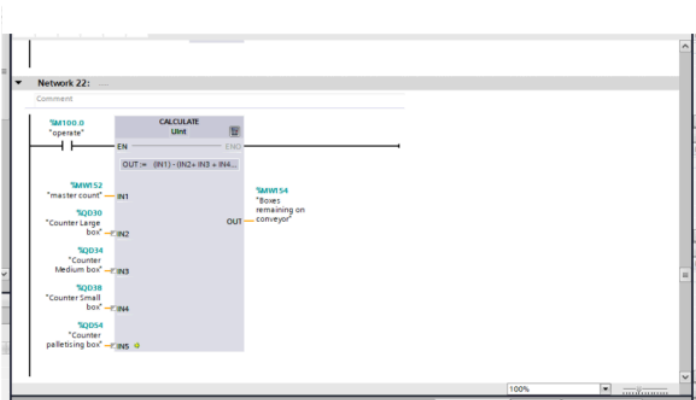
FIGURE 29 - NETWORK 20 AND 21



Network 20 moves the value of the block weight and will move it to a value that will allow the HMI to display it.

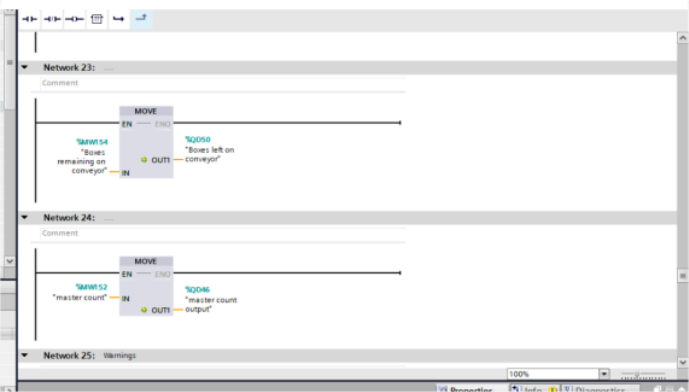
Every time a box is inserted into the system, a sensor will detect it, adding +1 to the 'master count'.

FIGURE 30 - NETWORK 22



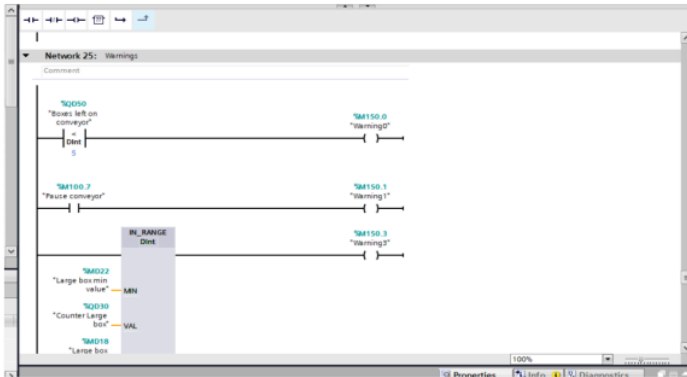
To determine the number of boxes left in the system, The total amount of boxes processed is subtracted from the 'master count' (Figure 30).

FIGURE 31 - NETWORK 23



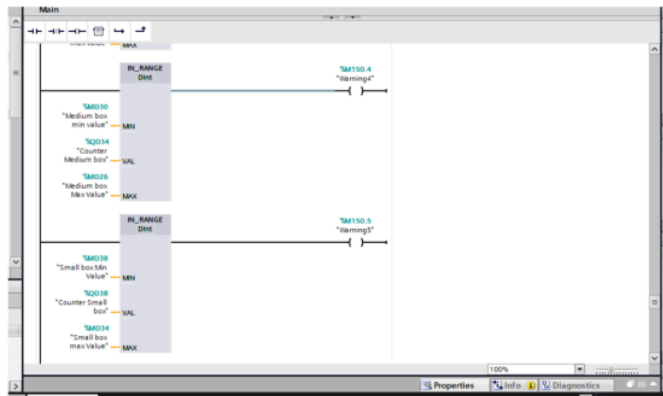
The counter values are moved to the relevant QD output values so that they can be used on the counters in the FactoryIO software.

FIGURE 32 - NETWORK 25



Figures 32 and 33 show the warnings that will appear on the HMI. If there are less than 5 boxes on the conveyor then warning 0 will activate. If the system is paused, warning 1 will activate.

FIGURE 33 - NETWORK 25



A Warning light will appear on the HMI if there are between 10 and 15 boxes of any size processed (this can be set on the HMI – Figures 33 and 34)

FIGURE 34 - NETWORK 25

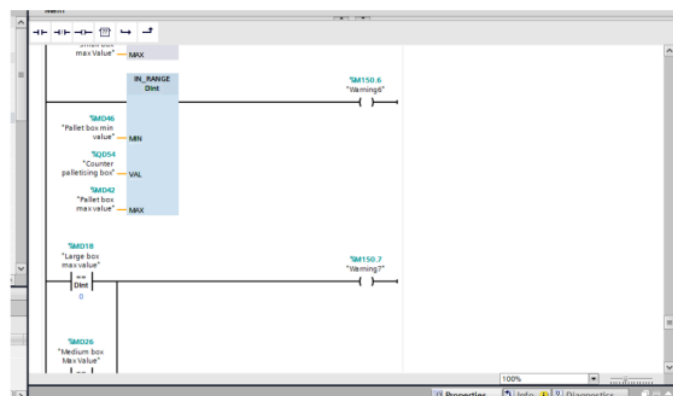


FIGURE 35 - NETWORK 25

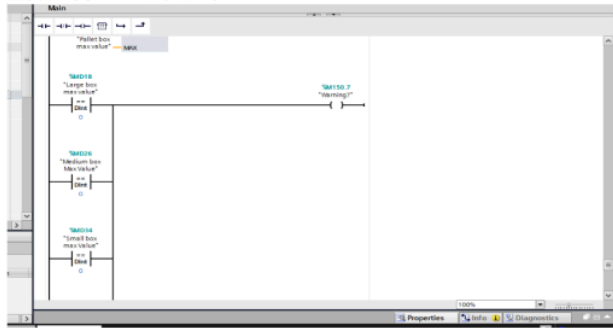
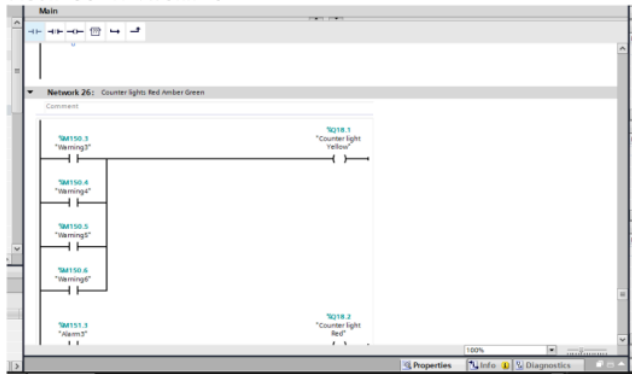
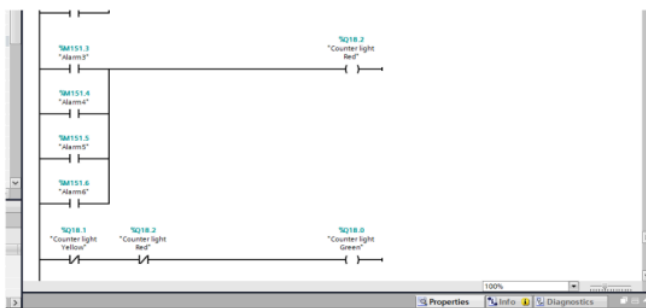


FIGURE 36 - NETWORK 26



If any of the warning activate to say that they are close to the maximum value than the yellow warning light will activate on the stack lights and also on the HMI stack lights.

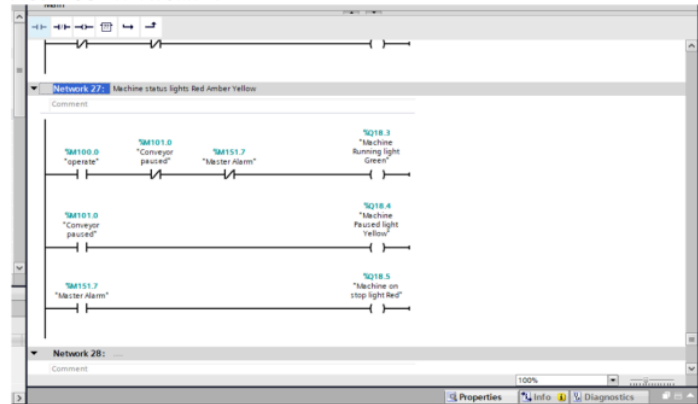
FIGURE 37 - NETWORK 26



An alarm signal will set the counter light output, activating the red light on the stack lights.

If neither red nor amber light is activated, then the green light will show that counters are below required value (Figure 37)

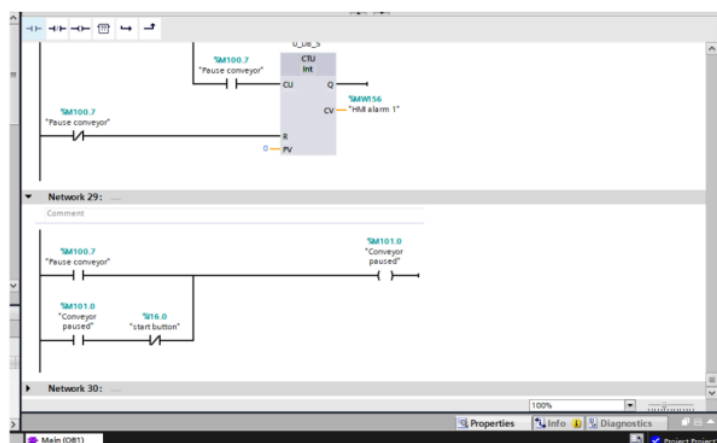
FIGURE 38 - NETWORK 27



Stack lights within the program are used to determine the status of the machine:

- Machine ok – Green
- Machine paused – yellow.
- Machine stopped – red.

FIGURE 39 - NETWORK 29



Network 29 shows a conveyor paused latching circuit. If the machine is paused, then the start button must be pressed to re-start. This prevents an unexpected startup while operators are next to the machine.

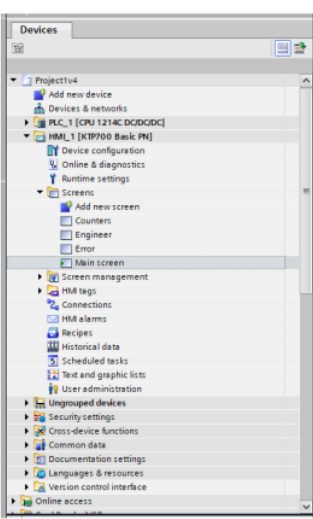
Siemens HMI

I used a simulated version of a Siemens TP700 to create my HMI. This contains the following screens:

- Main screen
- Error screen
- Engineer screen
- Counter screen

Project tree

FIGURE 40 - PROJECT TREE



The Project tree contains all of the HMI pages that are contained within the project. Each page can be found underneath the 'Screens' tab. new screens can be created as required.

FIGURE 41 - MAIN SCREEN

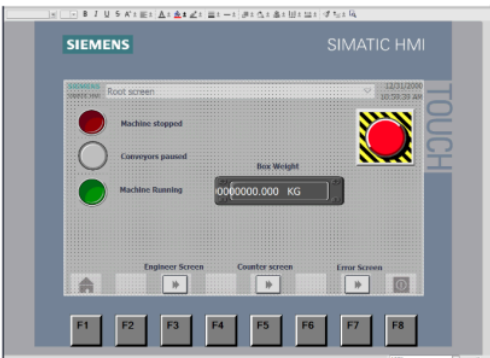
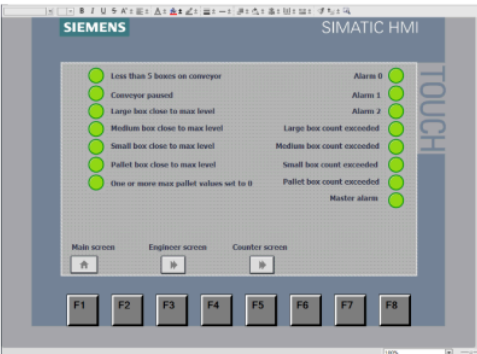


Figure 41 shows the main screen. This contains an E-stop, machine status lights and the box weight.

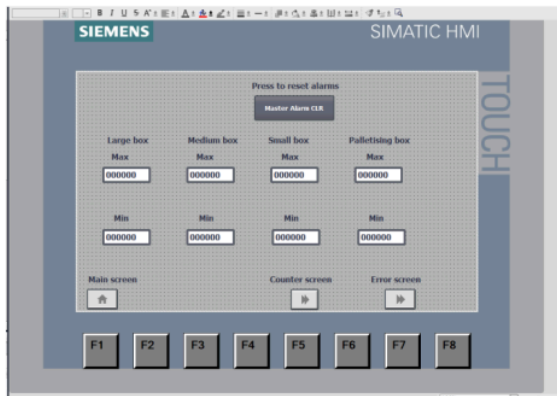
There are also tabs at the bottom of the screen to navigate to different screens as required.

FIGURE 42 - ERROR SCREEN



The error screen contains all errors and warnings within the system. The errors (to the left of the screen) will not stop the system. However, the errors (On the right of the screen) will cause the system to stop. These will have to be reset in the engineering screen before the system will begin again.

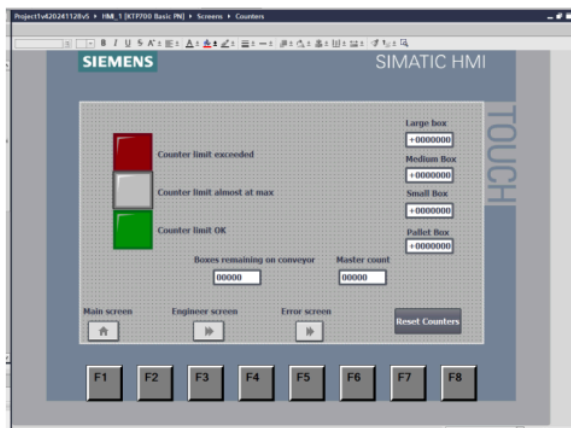
FIGURE 43 - ENGINEERING SCREEN



The engineering screen contains information that can only be modified by skilled personnel. This contains a master alarm clear. This will allow the machine to start back up after a fault, as long as the issue has been rectified.

This screen also contains options to determine the maximum and minimum amount of each type of box to be processed before the alarms activate. This can be changed as required during production.

FIGURE 44 - COUNTERS.



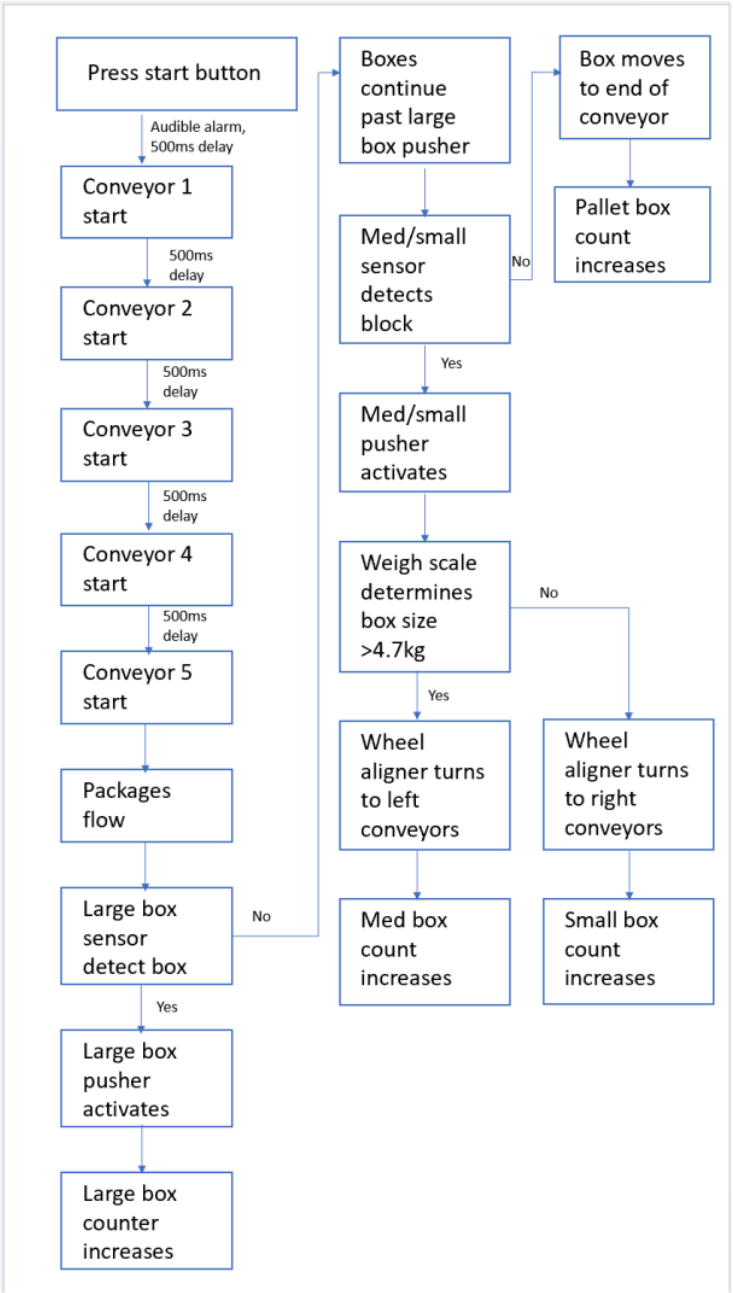
The counters screen contains a separate counter for each type of box. It also contains a counter for how many boxes are currently in the system and how many have been processed overall.

The reset counter button will reset all counts. There is also a push button on the main panel that will reset counters that can be used by operators.

Conclusion

My design includes sensors conveyors, counters and alarms. Being able to incorporate all of this into a plc program shows the flexibility and capabilities of modern PLCs. By understanding the core principles of how to utilise TIA portal, I am able to create more complex programs that can be used in both simulations and also in real world scenarios. For simulation purposes, Factory IO is a powerful software that is capable of replicating many functions of real life without the need to invest large amount of money during the testing phases of a program. Once the program has been proven to work in simulation, we can then look at the benefits of the design to determine the financial and safety benefits of the process.

FIGURE 45 - PROCESS FLOW CHART



References

- Alphonsus, E. R. (2016, 06). *A review on the applications of programmable logic controllers (PLCs)*. Retrieved from Science Direct: <https://www.sciencedirect.com/science/article/abs/pii/S1364032116000551>
- Cheng, K. (2024, 09 10). *Most commonly used plc's in industrialised companies*. Retrieved from mroelectric: <https://www.mroelectric.com/blog/most-common-used-plcs-in-industrialized-countries/#:~:text=In%20the%20United%20States%20and,role%20in%20Japan%20and%20China.>
- Kishorre, V. (2019, 12). *Intelligent manufacturing, a case study of siemens industry*. Retrieved from iopscience: <https://iopscience.iop.org/article/10.1088/1742-6596/1969/1/012019/pdf>
- Peko. (2017, 06). *Most popular plc brands and companies*. Retrieved from PK: <https://www.pekoprecision.com/blog/most-popular-plc-brands/>
- Su, K.-M. (2023). *Risks of control systems in PLC configurations*. Retrieved from <https://ieeexplore.ieee.org/document/9359077>
- Wright, C. (2020, 04). *Siemens vs Allen Bradley*. Retrieved from Automation Blog: <https://theautomationblog.com/siemens-vs-allen-bradley-plcs-one-engineers-perspective/>

Chris James PLC assignment

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

70/100

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

GRADING FORM: SMU GENERIC_GJ

CHRIS JAMES

69.90

RESEARCH-INFORMED LITERATURE



Good literature review of PLCs on the market, could have been improved with a few more devices and some images 70%

7

KNOWLEDGE AND UNDERSTANDING



The factory looks good with all of the required components included plus a few others. Your factory operates well and demos very well. 75%

22.5

ANALYSIS



Your ladder logic program is comprehensive and well documented throughout with an alarm strategy. Although the alarm strategy does not appear to function as expected 66%

19.8

PRACTICAL APPLICATION AND DEP.



Your HMI screens are reasonably laid out and informative for the operator but the error screen . The counters would probably have been more useful on the main screen. 68%

13.6

SKILLS FOR PROFESSIONAL PRAC.



The report is nicely presented with the network operations described.. The illustrations are good and clear. A reasonable selection of references, please note, in this discipline we require the use of IEEE referencing protocol. 70%

7