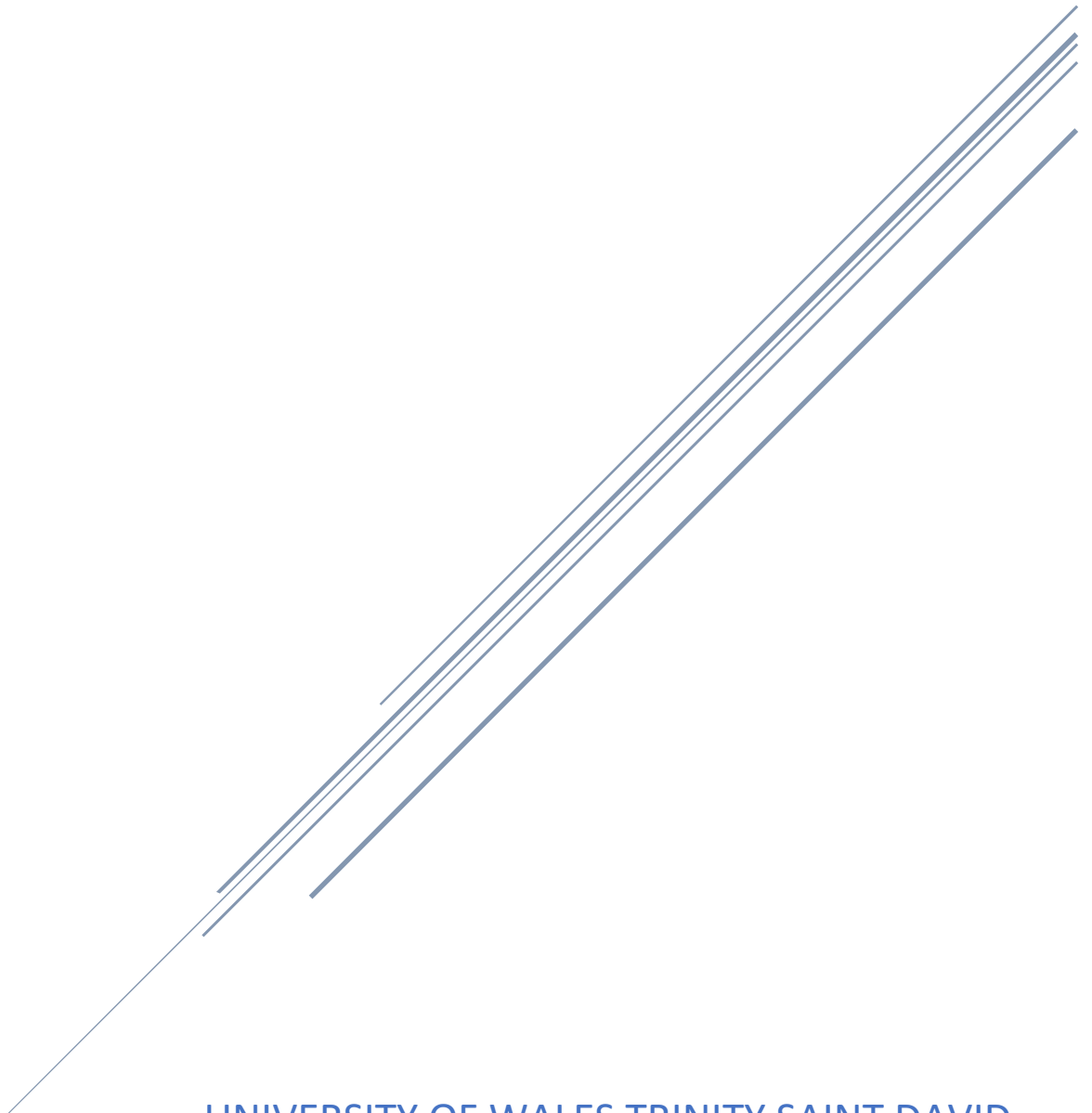


FACTORY CONTROL

ASSIGNMENT 1

BY MICHAEL HOWELLS



UNIVERSITY OF WALES TRINITY SAINT DAVID
APPLIED PROGRAMMABLE LOGIC CONTROLLERS - ACDA6000

CONTENTS

1	ABSTRACT.....	3
2	DESIGN BRIEF	4
3	LITERATURE REVIEW	5
3.1	WHAT IS A PLC?.....	5
3.2	PROGRAMMING PLCS	6
3.3	PLC INTERNAL ARCHITECTURE	7
3.4	PLC SYSTEM ARCHITECTURE	8
3.4.1	FIXED	8
3.4.2	MODULAR	9
3.4.3	DISTRIBUTED	10
3.5	WHAT IS A HMI?.....	11
4	FACTORY DESIGN	12
4.1	PACKAGE VARIANTS.....	12
4.2	CORE COMPONENTS	14
4.2.1	CONVEYOR	14
4.2.2	DIFFUSE SENSOR	15
4.2.3	PUSHER	16
4.2.4	CONVEYOR SCALE	17
4.2.5	POP UP WHEEL SORTER	17
4.3	DESIGN OVERVIEW	18
4.4	SORTING STRATEGY	21
4.5	CONTROL PANEL AND INDICATORS	23
5	ALARM/WARNING STRATEGY	25
5.1	IO ERROR.....	25
5.2	WEIGHT ERROR	25
5.3	IO LIST	26
6	PLC PROGRAM	27
6.1	FLOW CHART.....	27
6.2	LADDER DIAGRAM	28
6.2.1	NETWORK 1:.....	28
6.2.2	NETWORK 2: START OPERATION	28
6.2.3	NETWORK 3: RESET CONVEYOR SCALE	29

6.2.4	NETWORK 4: CONVEYORS.....	30
6.2.5	NETWORK 5: INDICATOR CONTROL.....	32
6.2.6	NETWORK 6: DIFF TRIGGER	32
6.2.7	NETWORK 7: PUSHER CONTROL	33
6.2.8	NETWORK 8: WARNING AND ALARM CONTROL	33
6.2.9	NETWORK 9: CONVEYOR SCALE CONTROL	34
6.2.10	NETWORK 10: WEIGH CONTROL.....	34
6.2.11	NETWORK 11: POP UP SORTER CONTROL	34
6.2.12	NETWORK 12: WEIGHT ERROR STATE	35
6.2.13	NETWORK 13: CONVEYOR 1 IO ERROR COUNTER	35
6.2.14	NETWORK 14: CONVEYOR 2 IO ERROR COUNTER	36
6.2.15	NETWORK 15: CONVEYOR 3 IO ERROR COUNTER	36
6.2.16	NETWORK 16: CONVEYOR 4 IO ERROR COUNTER	37
6.2.17	NETWORK 17: CONVEYOR 5 IO ERROR COUNTER	37
6.2.18	NETWORK 18: CONVEYOR 6 IO ERROR COUNTER	38
6.2.19	NETWORK 19: IO ALARM	39
6.2.20	NETWORK 20: ALARM BEACON	40
6.2.21	NETWORK 21: OUTPUT COUNTS	40
6.2.22	NETWORK 22: STACK LIGHT CONTROL	41
6.3	TAG LIST	42
6.4	HMI.....	45
6.4.1	ROOT SCREEN.....	45
6.4.2	ENGINEER SCREEN	46
6.4.3	ALARM SCREEN	47
6.4.4	GRAPHICAL ALARM SCREEN.....	48
7	DISCUSSION.....	49
8	CONCLUSION.....	49
9	REFERENCES	50
10	APPENDICES	51
10.1	LINKS	51
10.2	FIGURES.....	51

1 ABSTRACT

This document details the design process for an automated production packing line. This packing process is controlled by a PLC and has a HMI for users. TIA Portal V16 was used to design the PLC program and Factory IO was used to simulate the factory environment.

Research of PLCs was also undertaken to help understand what they are and the different types that are available.

2 DESIGN BRIEF

The design brief is as follows:

"You are required to design and control a factory system which sorts 4 different packages based on size and weight. The factory is to be designed in Factory I/O and must contain and control the following as a minimum:-

4 conveyor belts

2 pushers

1 conveyor scale

2 Diffuse sensors

1 Emergency Stop

2 Buttons

1 Siren.

1 Warning light

2 Stack lights

To limit the current draw at start up, each conveyor should be started in sequence. A manual reset of alarm signals once the alarm issue is corrected.

The system will require a HMI for operator interaction. The HMI must have the following controls and displays available on the root screen: -

- Buttons for starting and stopping the system*
- An emergency stop button*
- A button to reset all counters*
- counters*
- A replication of the stack lights*
- Fault and warning indicators*

- A fault and warning screen which displays alarm and warnings.*

Warning and alarm must include: -

- Counter warning number exceeded for each counter*

An engineering screen which has data entry points for:

- Pre-setting counter Warning numbers*

The factory must also have the following which may be manually controlled

- 1 package Emitter*
- package Removers"*

3 LITERATURE REVIEW

3.1 WHAT IS A PLC?

A Programmable Logic Controller (PLC) is a type of microprocessor-based device that can be programmed to operate in a multitude of ways so that it can control machines and processes. These operations include:

- Counting
- Sequencing
- Timing
- Logic
- Mathematic operations.

PLCs can operate in tougher conditions than some other control devices such as standard PCs which make them suitable for industrial environments. This could be places with increased noise, vibrations, heat and moisture [1].

PLCs are programmed in a way that makes it accessible for people with limited knowledge of programming and computers [1].

The first PLC was developed in 1969 by General Motors as they were interested in replacing their relay control system with computers for their automated car manufacturing plants. Two other companies at the time also developed their own controllers in response: Allen Bradley and Bedford Associates. These devices were called programmable controllers and were designed to survive in industrial environments. They were programmed using a language based on relay schematic diagrams and produced documents called Ladder Diagrams which are still used today [2]. Early programs were written using a terminal with keys showing the different symbols, shown in Figure 1.

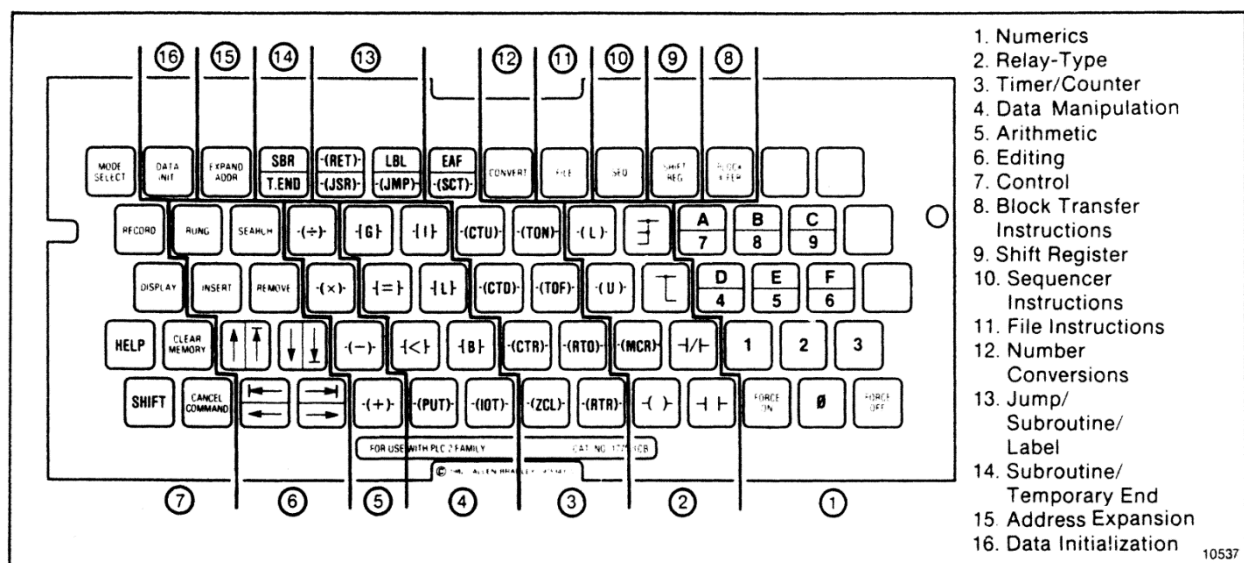


Figure 1 – Early PLC Programming keyboard [3]

3.2 PROGRAMMING PLCs

PLCs programs are written for numerous applications but all will refer to inputs and outputs that are on the system. They are referenced in different ways depending on the manufacturer of the system being used, two examples are shown in Table 1.

Manufacturer	Input Notation	Output Notation
Allen Bradley	I:Rack Slot/Bit	O:Rack Slot/Bit
Siemens	I<byte>/<bit>	Q<byte>/<bit>

Table 1 - Example IO Notation [3]

One of the most common programming languages for PLCs is ladder logic. This was developed as PLCs were replacing relay control schemes. The ladder logic schematics used the same symbols and therefore was helpful in getting the new technology adopted as people were familiar with the diagrams [4]. An example can be seen in Figure 2.

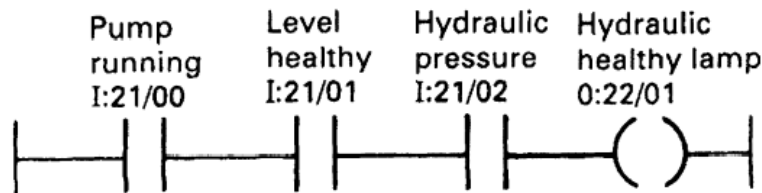


Figure 2 - Ladder Logic Example [4]

More complex operators can be included in the ladder logic rungs. An example can be seen in Figure 3.

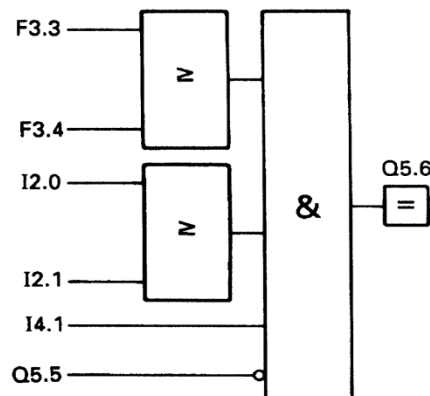


Figure 3 – Ladder logic example with complex [4]

Other programming languages are available such as structured text based ones.

3.3 PLC INTERNAL ARCHITECTURE

The internal architecture of a PLC can be seen in Figure 4. It consists of the following parts:

- Central Processing unit (CPU): This section carries out any operational actions according to the program in memory. It manages any mathematical, logic and timing operations. They also have internal memory that is required to operate the PLC program.
- Clock: The clock determines the frequency of the CPU. It is generally at a frequency between 1-8MHz.
- Buses: The data buses are the paths that the signals are carried on. There are busses for different functions in the PLC. These include, the data bus, address bus, control bus and system bus.
- Memory: There are different types of memory in the PLC. These include: system read-only-memory, user random access memory, data random access memory and also any memory required for add-on modules if required.
- Input/Output (IO) unit: This part interfaces with the real-world inputs and outputs. Each IO will have a unique address which can be referenced within the program.
- Power Supply: The PLC power supply regulates the mains AC to what is required for the PLC to operate. This is usually 5V DC [1].

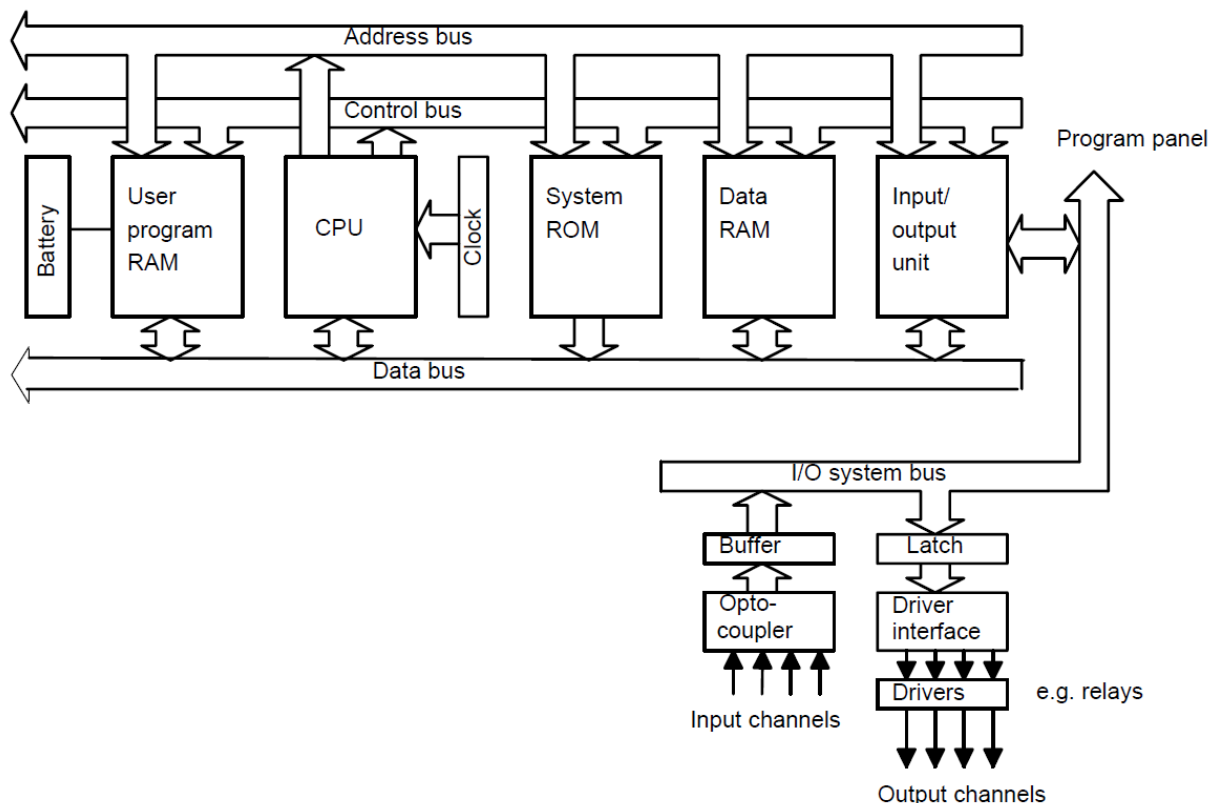


Figure 4 - PLC Architecture [1]

3.4 PLC SYSTEM ARCHITECTURE

The architecture of a PLC defines how the software and hardware interface with each other to make the overall system. There are 3 main types of PLCs architecture which are fixed, modular and distributed [5].

The advantages and disadvantages of each will be shown in subsequent sections.

3.4.1 FIXED

The fixed PLC contains all hardware in a single unit. An example of a fixed PLC is the “Siemens LOGO!” (Figure 5). Fixed PLCs are the simplest architecture type and are used when the application is relatively small scale where the number of IOs required is quite small [5]. Advantages and disadvantages of fixed PLCs can be seen in Table 2.



Figure 5 - Fixed PLC Example (Siemens LOGO!) [9]

Advantage	Disadvantage
Small size	Not customizable
Low cost	Limited number of IOs, therefore not suitable for large applications.
Simpler design means implementation time could be reduced compared to other larger systems.	Lowest performing PLC based on CPU and memory size.

Table 2 - Advantages and Disadvantages of Fixed PLCs [5]

3.4.2 MODULAR

A modular PLC contains separate modules for each part of its hardware. For example, the system could have modules for IO interfaces, memory and power supplies. This means that the systems can be heavily customized for bespoke applications. For example, if the application requires many IOs then several IO modules can be added to meet the requirement. An example of a modular PLC is the Omron CJ2M (Figure 6). Advantages and disadvantages of modular PLCs can be seen in Table 3.



Figure 6 - Modular PLC (Omron CJ2M) [10]

Advantage	Disadvantage
Functionality is extremely customisable due to modules that are available. For example, if more IOs are required, another IO module could be installed.	Potentially complex system design could mean development time is significant.
Medium to high performing CPU	High Cost
Large amounts of memory available.	Not suitable for small applications.

Table 3 - Advantages and Disadvantages of Modular PLCs [5]

3.4.3 DISTRIBUTED

A distributed PLC is similar to a modular PLC in that it has separate modules for each part of its hardware, however the difference is that the modules can be installed in different locations. An example of a distributed PLC system is the SIMATIC ET 200SP series (Figure 7Figure 6). Advantages and disadvantages of distributed PLCs can be seen in Table 4.



Figure 7 - Distributed PLC Example (SIMATIC ET 200SP) [11]

Advantage	Disadvantage
Functionality is extremely customisable due to modules that are available. For example, if more IOs are required, another IO module could be installed. The system can be even more customisable as modules can be installed in different locations when compared to the modular PLC.	Potentially complex system design could mean development time is significant.
High performing CPU	High Cost
Large amounts of memory available.	Not suitable for small applications.

Table 4 - Advantages and Disadvantages of Distributed PLCs [5]

3.5 WHAT IS A HMI?

A human machine interface (HMI) is a screen that is designed and programmed to allow a user to interface with a PLC program and display key parts within it, such as warnings and alarms. The Siemens Simatic series is an example of a HMI (Figure 8).



Figure 8 - HMI Example [12]

4 FACTORY DESIGN

A software package called Factory IO was used to create a virtual factory which included all required elements specified in the brief. Factory IO was linked to the TIA portal PLC software which enabled PLC programs to control the simulated factory. This section will explore the design process for the virtual factory and the parts that were included in the design.

4.1 PACKAGE VARIANTS

The 4 packages that must be sorted are:

- Large boxes (Figure 9)

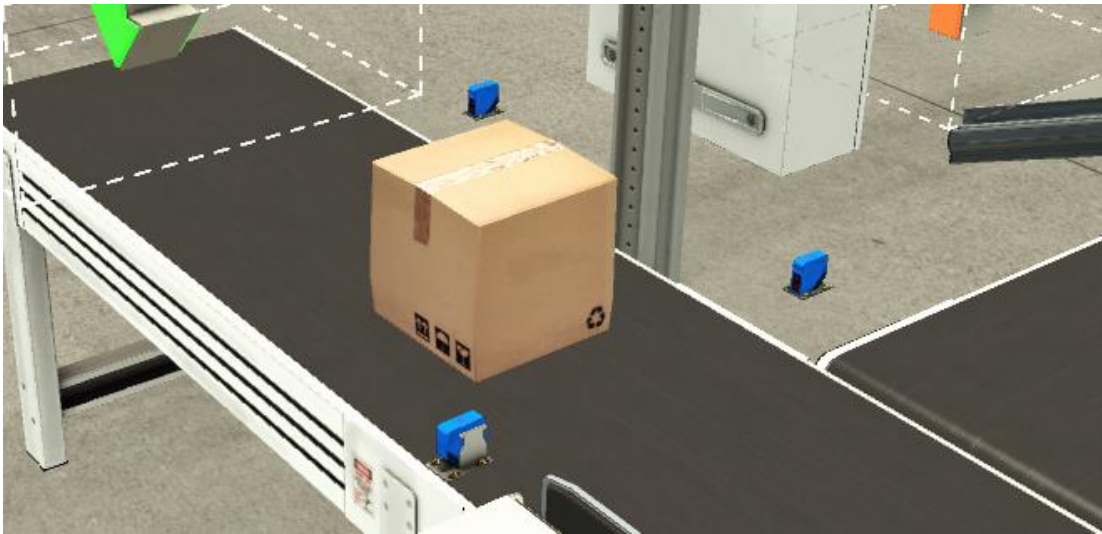


Figure 9 - Large Box

- Medium boxes (Figure 10)



Figure 10 - Medium Box

- Small boxes (Figure 11)

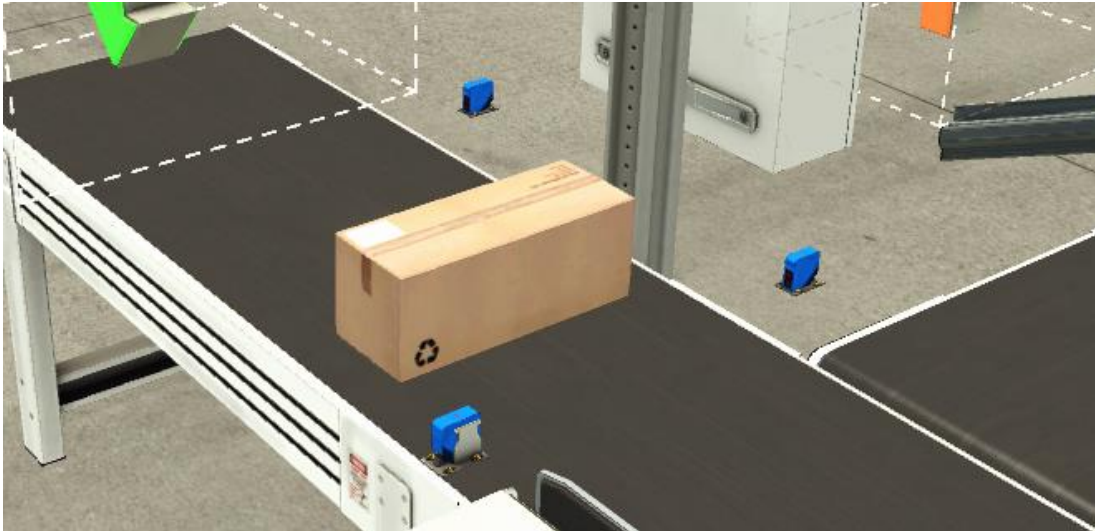


Figure 11 - Small Box

- Palletising boxes (Figure 12)

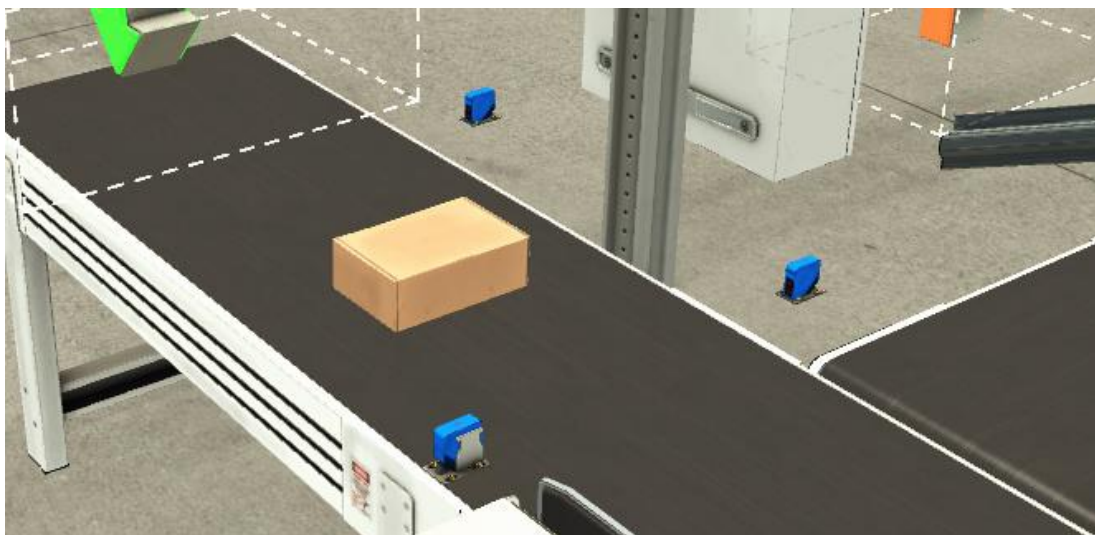


Figure 12 - Palletising Box

4.2 CORE COMPONENTS

In this section, the main components used in Factory IO will be briefly explained as it will aid in understanding the virtual factory. Screenshots in this section have been taken directly from Factory IO.

4.2.1 CONVEYOR

The conveyor is the main method for moving the packages around the virtual factory. Both straight (Figure 14) and curved (Figure 13) conveyors were used.



Figure 14 - Factory IO Straight Conveyor



Figure 13 - Factory IO Curved Conveyor

4.2.2 DIFFUSE SENSOR

The diffuse sensor (Figure 15) is used to detect the presence of something in front of it. Real ones do this by emitting light in front of it and when something is in front of it, the light back into its receive, detecting the object [6].

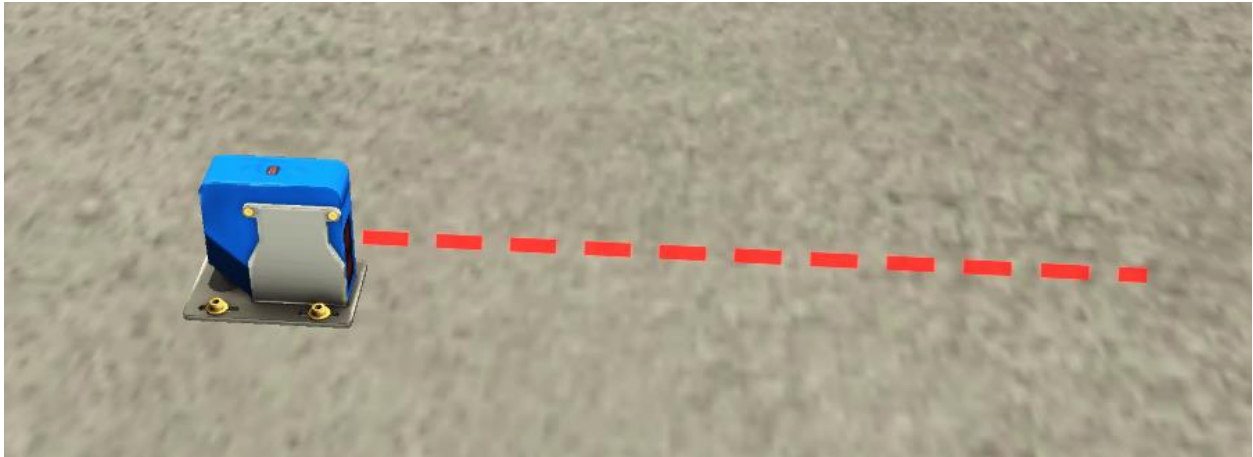


Figure 15- Factory IO Diffuse Sensor

4.2.3 PUSHER

The pusher extends itself when it is triggered. This is used to divert packages onto different parts of the process. The pusher can be seen in Figure 16 and Figure 17.

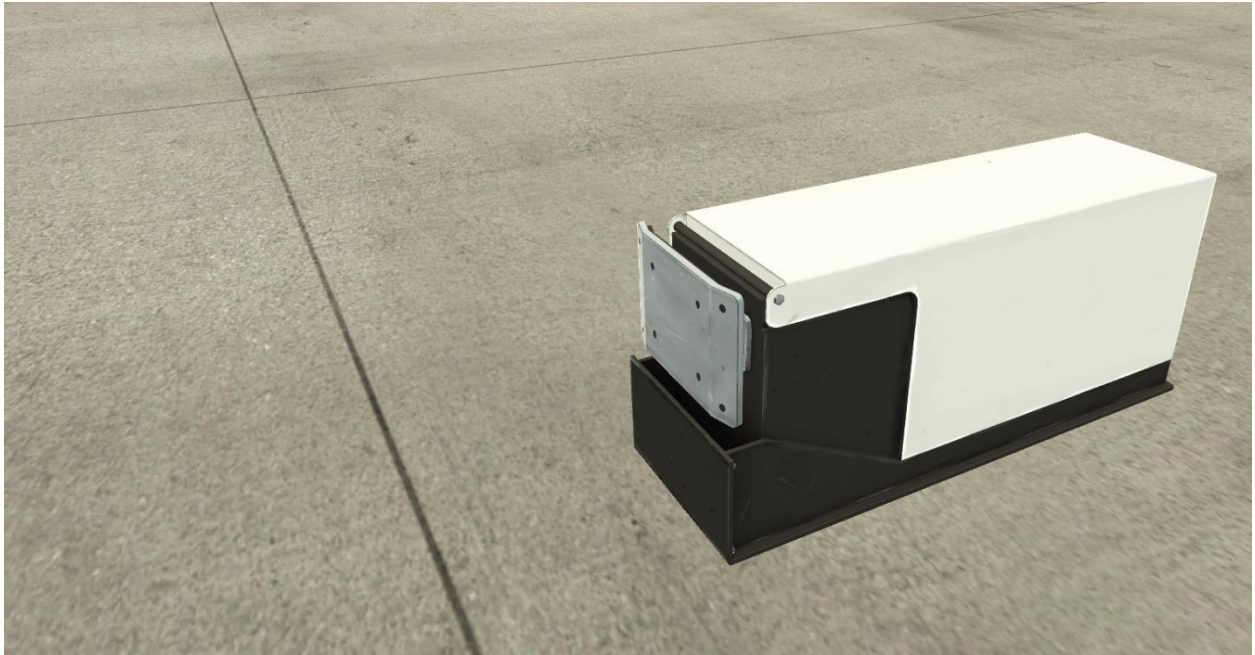


Figure 16 - Factory IO Pusher (in)

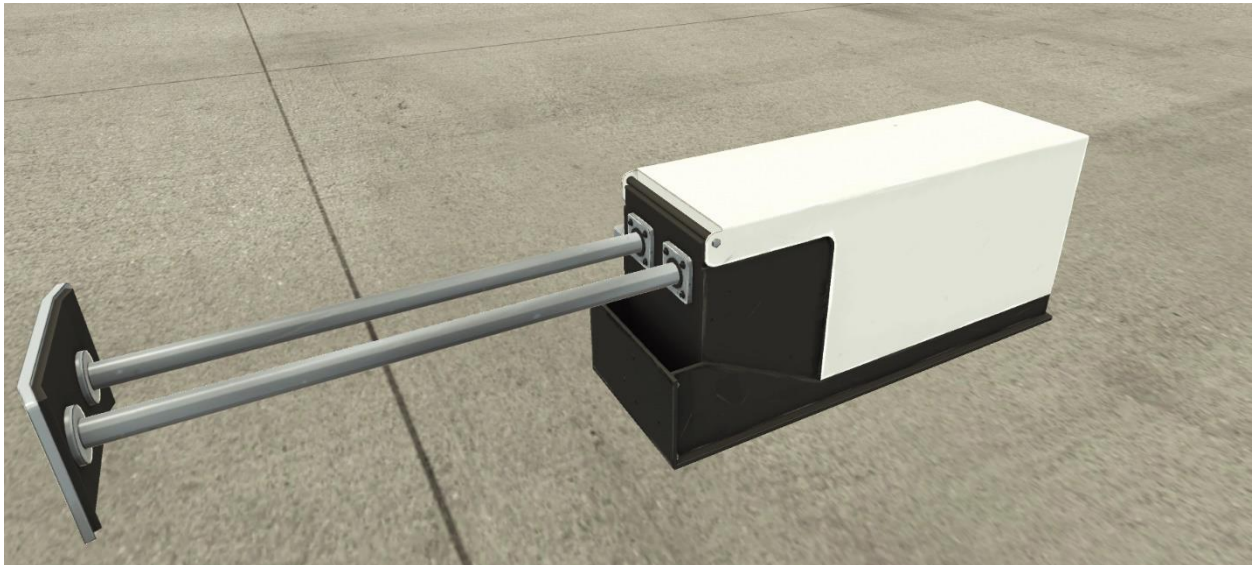


Figure 17 - Factory IO Pusher (out)

4.2.4 CONVEYOR SCALE

The conveyor scale (Figure 18) is used to weigh the boxes. It has a built-in conveyor to move the product across itself.



Figure 18 - Factory IO Weighing Scales

4.2.5 POP UP WHEEL SORTER

The pop-up wheel sorter is another method used to divert the boxes. It can either divert left or right depending on its current state, shown in Figure 19.

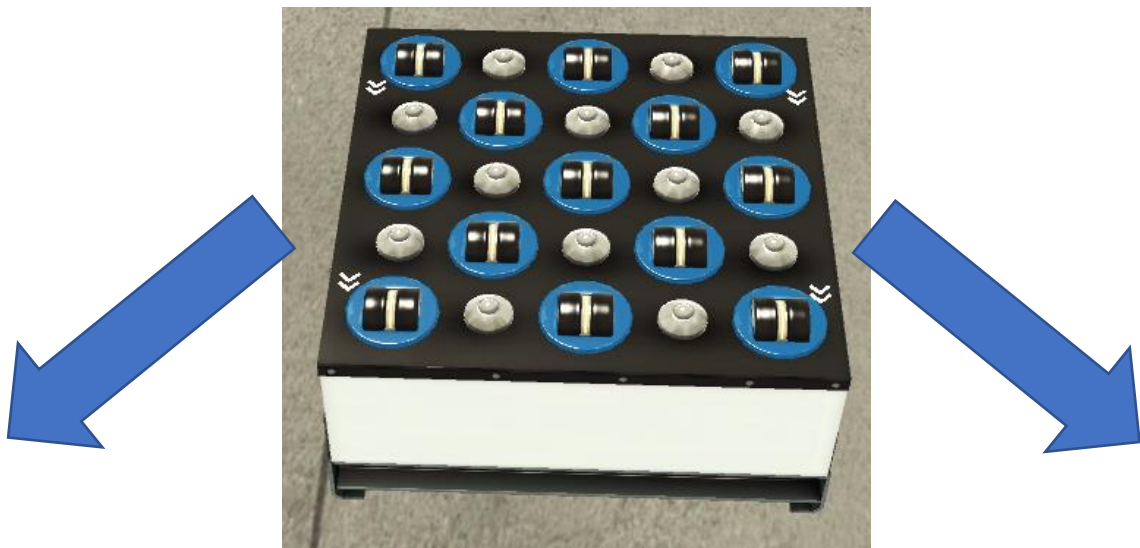


Figure 19 – Factory IO Pop Up Wheel Sorter and Output Directions

4.3 DESIGN OVERVIEW

Figure 20 shows a block diagram view of the factory IO Design. All box types enter the system at input 1 and are then sorted so that they get to their desired output.

Large boxes → Output 1

Small boxes → Output 2

Medium → Output 3

Palletiser boxes → Output 4

The sorting strategy is explained in 4.4 SORTING STRATEGY. The actual layout can be seen in Figure 21 and Figure 22.

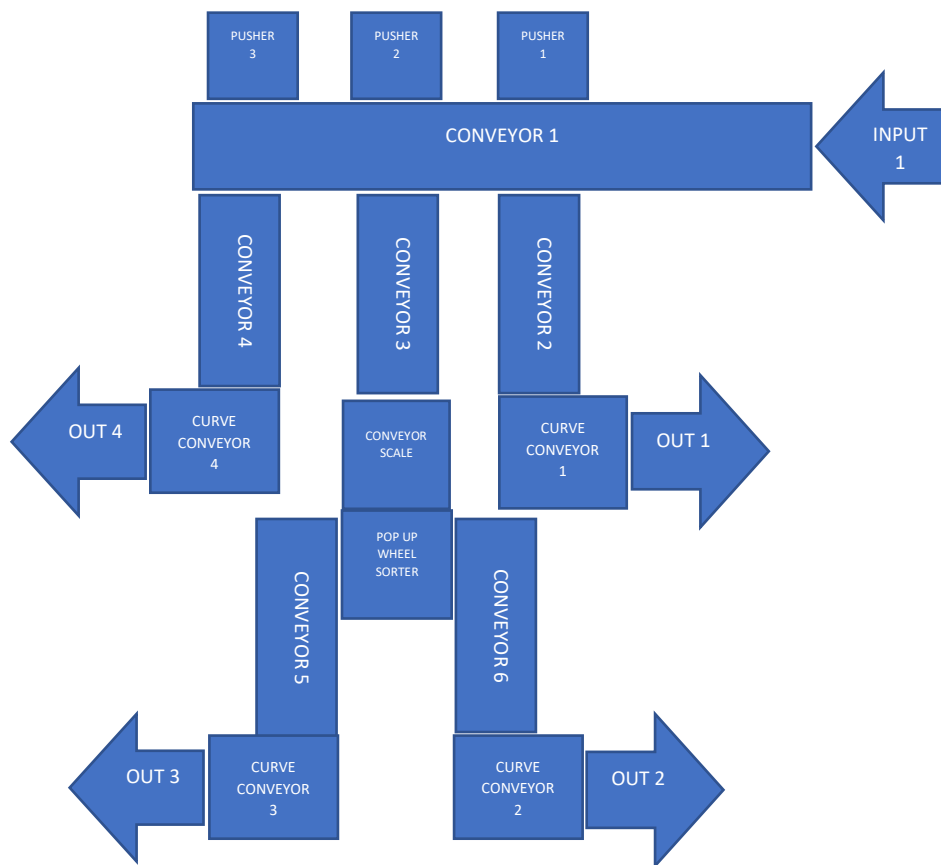


Figure 20 - Factory IO Overview - Annotated Blocks

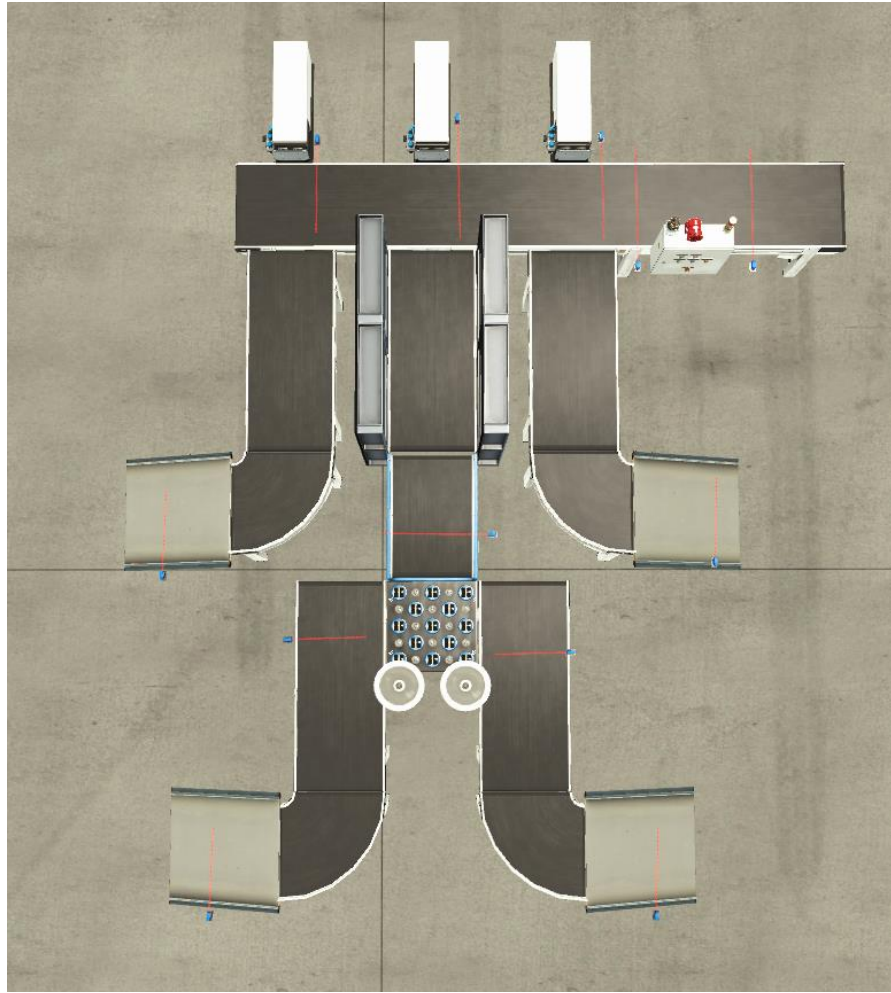


Figure 21 - Factory IO Overview – Top View– Actual Design

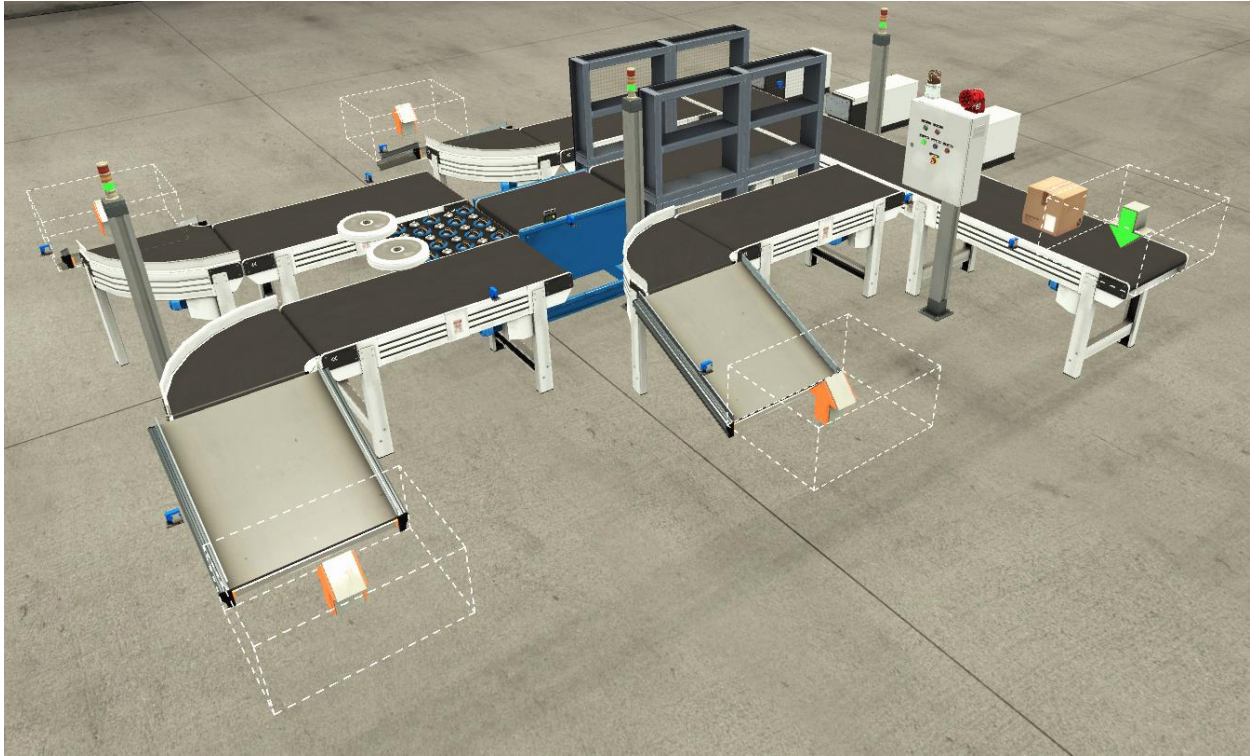


Figure 22 - Factory IO Overview – Side View – Actual Design

4.4 SORTING STRATEGY

To sort the boxes, the unique features of each box were exploited. For example, the difference in heights were used to sort some of the boxes. A diffuse sensor at different heights could be used to detect which box is present and divert accordingly. This will be used to sort the palletising boxes and large boxes to their output lanes.

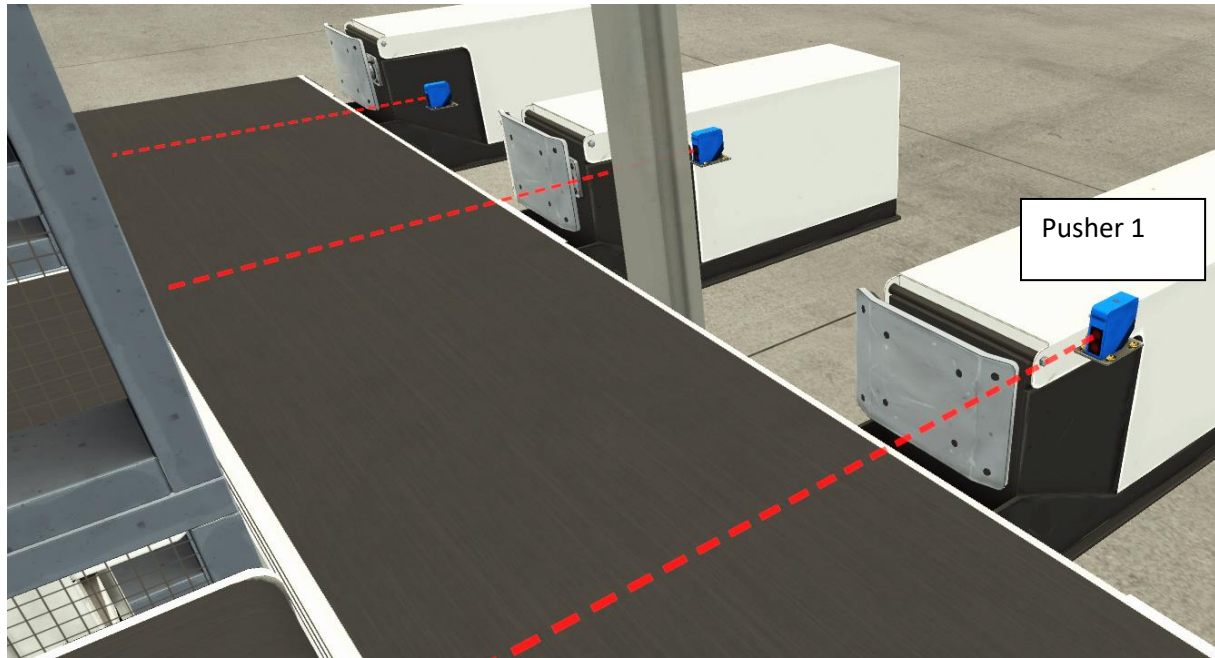


Figure 23 - Diffuse Sensors at different heights

Figure 22 shows the diffuse sensors at different heights. The diffuse sensor next to pusher 1 is higher than the others. Large boxes will trigger it and will cause them to be diverted first. Smaller boxes will move under it and will not trigger the sensor.

Small and medium boxes have the same height and therefore will be sorted but require a second sorting stage to separate them. As the small and medium boxes have different weights, they can be sorted by a scale on the line, that diverts them depending on the weight that was measured.

Figure 24 shows the where the conveyor scale is used to determine the weight of the boxes. This information is used to decide where the pop-up wheel sorter will divert the small and medium boxes.



Figure 24 – Conveyor Scale and pop up wheel sorter

4.5 CONTROL PANEL AND INDICATORS

There are several indicators included in the factory IO simulation.

A siren triggers on startup to alert anyone near the process that it is about to start. A red warning light will turn on when the process is in an alarm state. Other indicators include, a light to show when the process is running and another to show when it is stopped. The siren, warning light and indicators can be seen in Figure 25.



Figure 25 - Control Panel with indicators and buttons

The remaining indicators are the beacons. These show the state of the line by illuminating one of their three lights (green, yellow, and red). Green indicates there are no alarms or errors. Yellow indicates a warning has been triggered. And red means there is an alarm. There are 3 positioned across the production line so that the state of the process can be determined from a distance. The beacon states can be seen in Figure 26.

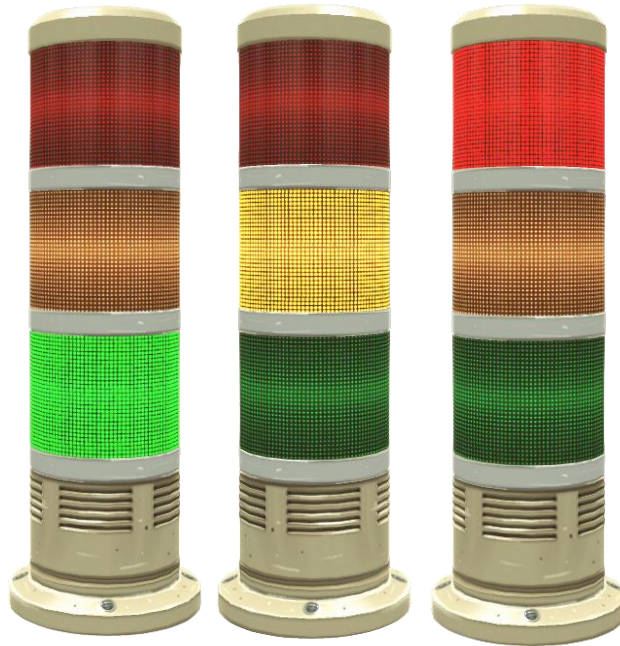


Figure 26 - Beacon States (green, yellow, and red)

There are 3 physical buttons in the simulation. One button is used to start the process and another is used to stop it. The remaining button is an E-stop that if pressed will trigger the alarm flag and will stop the process. They can be seen in Figure 25.

5 ALARM/WARNING STRATEGY

Two methods for determining alarm/warning conditions were implemented into the PLC program.

5.1 IO ERROR

There are 5 places in the process where boxes are monitored to check if any are going missing from the production line. This is done by incrementing a count up at the start of the check point and then incrementing a count down at the end of the check point. If a box passes the line correctly, then the count should not increase over time. However, if boxes are entering a check point and then going missing, the count will only increment up. The value will increase to a specified threshold, indicating an error. The program is written with a warning and alarm threshold. If too many boxes go missing, the line will stop.

There are warning flags for each check point section, however there is only 1 alarm flag for the IO errors. The reason for this design choice was because if the alarm flag is triggered, then there must be a warning flag also triggered as it must have incremented beyond the threshold. The warning light for the specific section will be lit and the error location can be identified from that. A helpful HMI screen was added so the user can quickly see where the error has occurred.

5.2 WEIGHT ERROR

The second error state is triggered if an unknown weight is detected on the scale. This is because only two weights should be detected at this stage (small and medium boxes). If some unknown weight is detected then the process will stop as something incorrect has gone down the line.

5.3 IO LIST

The IO list for all parts used in factory io can be seen in Figure 27.

Start Button 0	%I6.0	%Q16.0	Light Indicator (Green) 1
Stop Button 0	%I6.1	%Q16.1	Light Indicator (Red) 2
	%I6.2	%Q16.2	Light Indicator (Blue) 0
Diffuse Sensor 0	%I6.3	%Q16.3	Alarm Siren 0
Diffuse Sensor 1	%I6.4	%Q16.4	Pusher 0
Diffuse Sensor 3	%I6.5	%Q16.5	Pusher 1
Diffuse Sensor 2	%I6.6	%Q16.6	Pusher 2
	%I6.7	%Q16.7	
Emergency Stop 0	%I7.0	%Q17.0	Belt Conveyor (Bm) 0
	%I7.1	%Q17.1	Belt Conveyor (Zm) 0
Diffuse Sensor 8	%I7.2	%Q17.2	Belt Conveyor (Zm) 1
Diffuse Sensor 11	%I7.3	%Q17.3	Belt Conveyor (Zm) 2
Diffuse Sensor 4	%I7.4	%Q17.4	Curved Belt Conveyor 0 CCW
Diffuse Sensor 5	%I7.5	%Q17.5	Curved Belt Conveyor 0 CCW
Diffuse Sensor 9	%I7.6	%Q17.6	
Diffuse Sensor 6	%I7.7	%Q17.7	Conveyor Scale 0 ()
Diffuse Sensor 7	%I8.0	%Q18.0	
Diffuse Sensor 10	%I8.1	%Q18.1	Pop Up Wheel Starter 0 (+)
	%I8.2	%Q18.2	Pop Up Wheel Starter 0 (Left)
	%I8.3	%Q18.3	Pop Up Wheel Starter 0 (Right)
	%I8.4	%Q18.4	
	%I8.5	%Q18.5	
	%I8.6	%Q18.6	
	%I8.7	%Q18.7	Warning Light 0
	%I9.0	%Q19.0	
	%I9.1	%Q19.1	Belt Conveyor (Zm) 3
	%I9.2	%Q19.2	Belt Conveyor (Zm) 4
	%I9.3	%Q19.3	Curved Belt Conveyor 2 CCW
	%I9.4	%Q19.4	Curved Belt Conveyor 3 CCW
	%I9.5	%Q19.5	
	%I9.6	%Q19.6	Stack Light 0 (Green)
	%I9.7	%Q19.7	Stack Light 0 (Yellow)
	%Q0.0	%Q00.0	Stack Light 0 (Red)
	%Q0.1	%Q00.1	Stack Light 1 (Green)
	%Q0.2	%Q00.2	Stack Light 1 (Yellow)
	%Q0.3	%Q00.3	Stack Light 1 (Red)
	%Q0.4	%Q00.4	Stack Light 2 (Green)
	%Q0.5	%Q00.5	Stack Light 2 (Yellow)
	%Q0.6	%Q00.6	Stack Light 2 (Red)
	%Q0.7	%Q00.7	
Conveyor Scale 0 Weight (V)	%Q30 (REAL)	(REAL) %Q030	Digital Display 0
	%Q34	%Q034	
	%Q38	%Q038	
	%Q42	%Q042	
	%Q46	%Q046	
	%Q50	%Q050	
	%Q54	%Q054	
	%Q58	%Q058	

Figure 27 - IO list used in Factory IO

6 PLC PROGRAM

6.1 FLOW CHART

A link to the full pdf of the flow chart can be found in the APPENDICES section.

6.2 LADDER DIAGRAM

The PLC program was written in TIA Portal using ladder logic.

An explanation of each rung in the ladder diagram is included in this section. A full link to the ladder diagram PDF can be found in the APPENDICES section.

6.2.1 NETWORK 1:

The first network contains a function block that is required to get TIA portal to communicate with Factory IO.

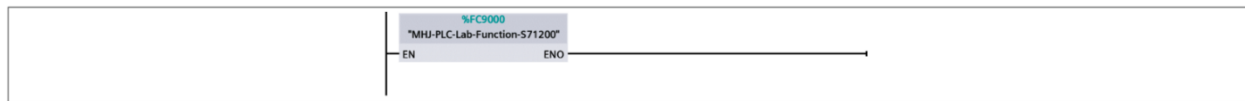


Figure 28 - Network 1

6.2.2 NETWORK 2: START OPERATION

When the start button is pressed, the operate memory bool is latched to start the system.

This will not occur if there are any alarms in the system or if the stop button is pressed.

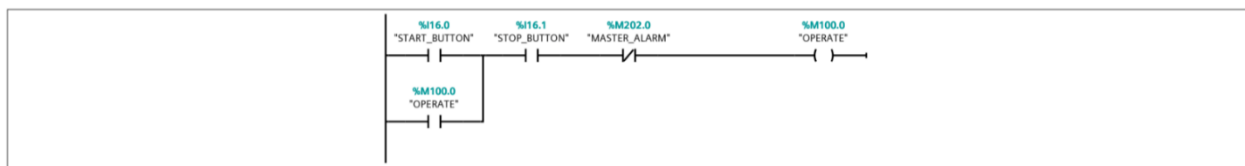


Figure 29 - Network 2

6.2.3 NETWORK 3: RESET CONVEYOR SCALE

The conveyor scale is reset by the general reset memory bool and the stop button.

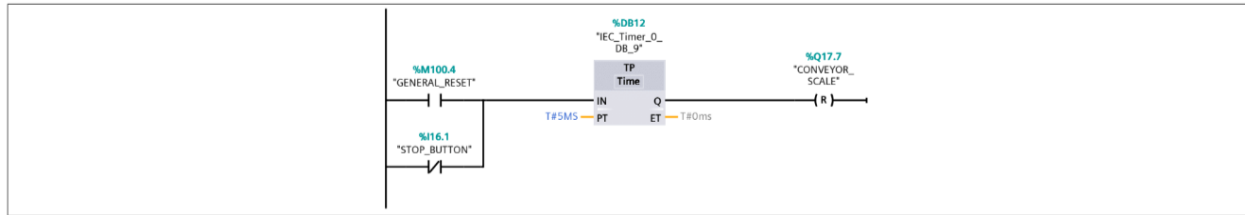


Figure 30 - Network 3

6.2.4 NETWORK 4: CONVEYORS

Operate bool will start to turn on the conveyors. The turning on of each conveyor is staggered to prevent current surges triggering any MCBs on the production line.

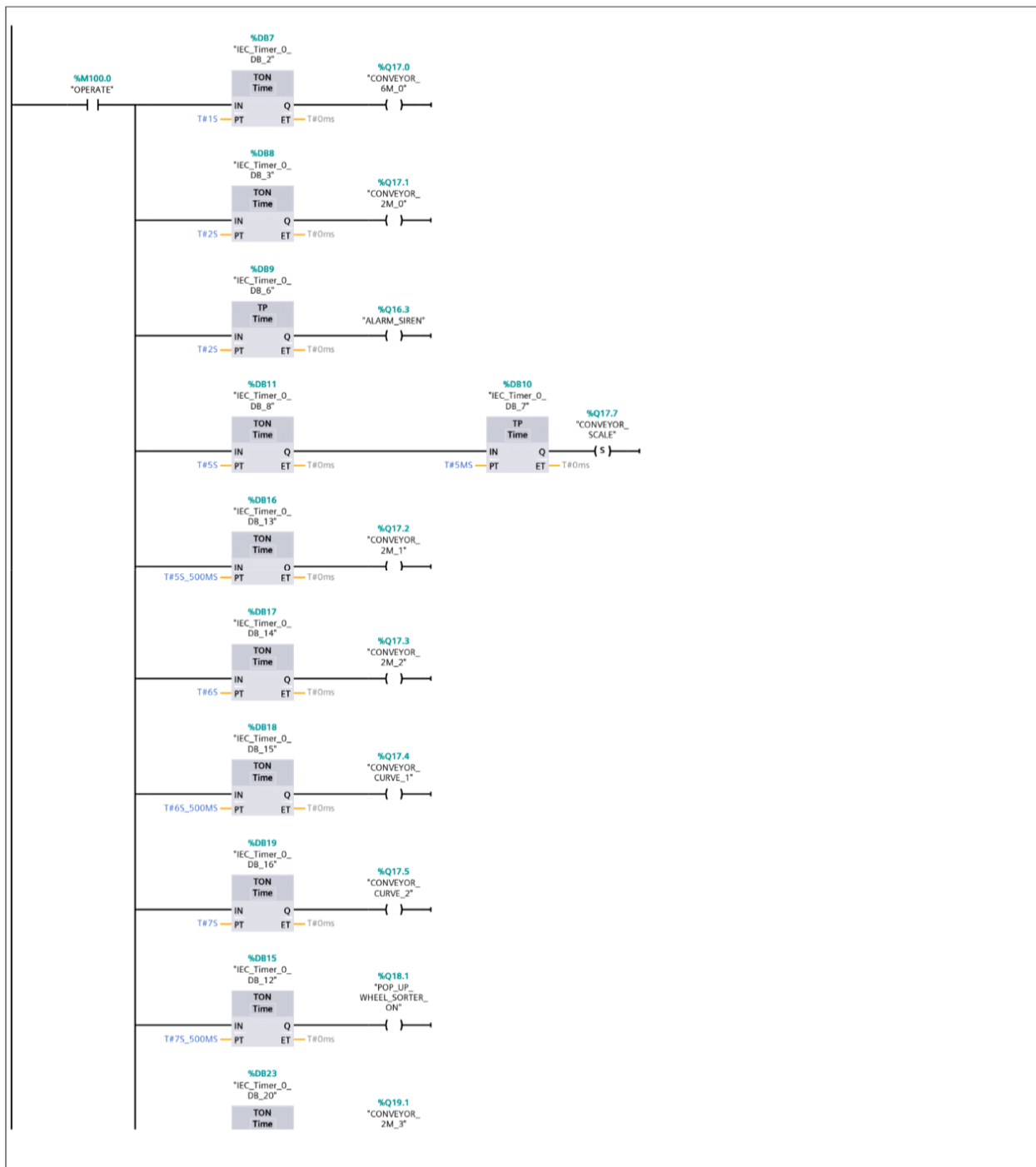


Figure 31 - Network 4 Part 1

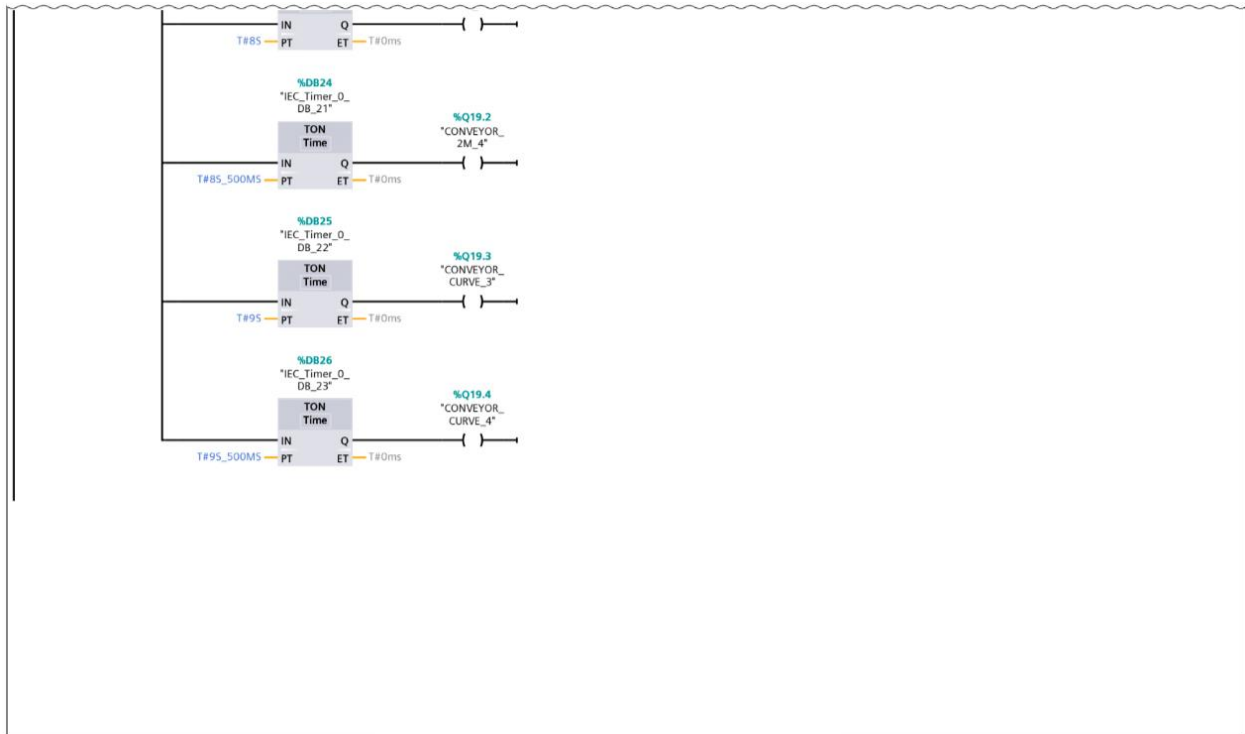


Figure 32 - Network 4 Part 2

6.2.5 NETWORK 5: INDICATOR CONTROL

The green indicator is only on when the operate bool is 1. The red indicator is only on when the operate bool is 0.

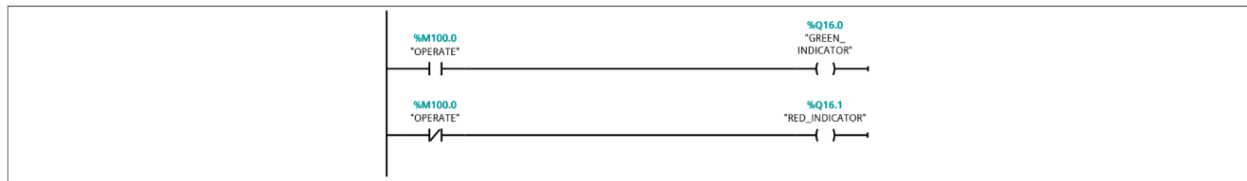


Figure 33 - Network 5

6.2.6 NETWORK 6: DIFF TRIGGER

Diffuse sensor 1, 2 and 3 toggle their respective memory booleans to 1. These are used to trigger the pushers.

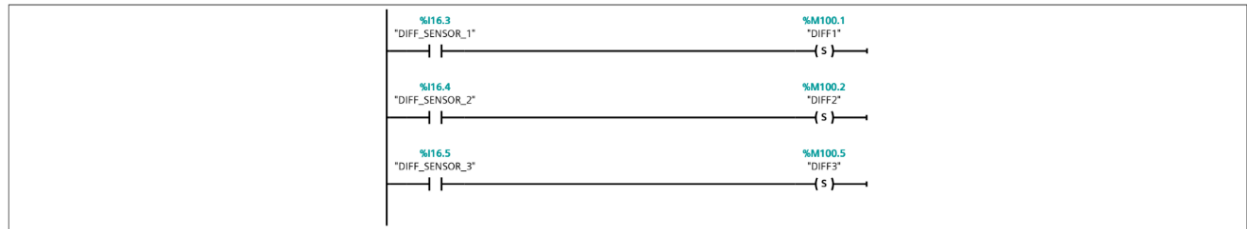


Figure 34 - Network 6

6.2.7 NETWORK 7: PUSHER CONTROL

Each pusher is triggered by the booleans set by the diff sensors in network 6. After a set time the toggled diff sensor bools is reset. This resets the pusher ready for the next box.

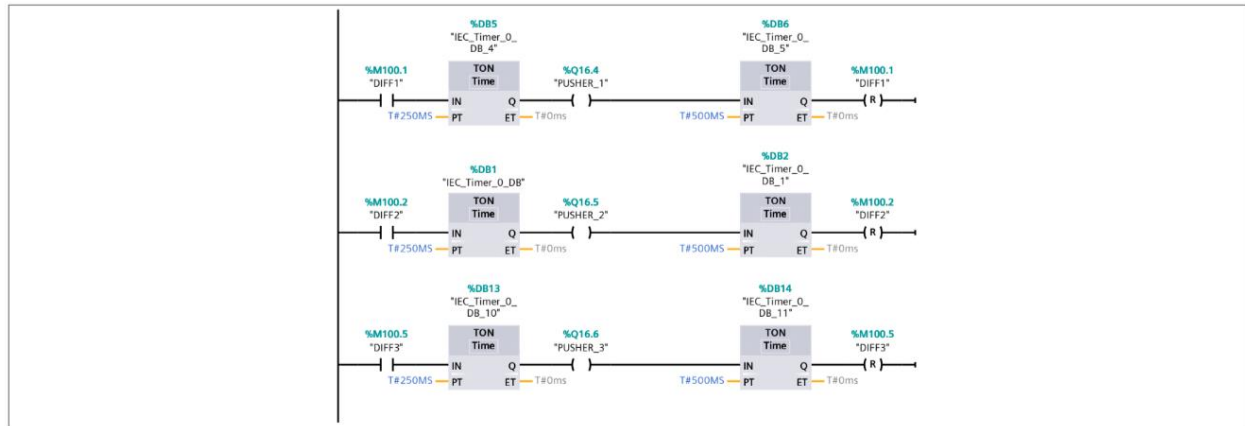


Figure 35 - Network 7

6.2.8 NETWORK 8: WARNING AND ALARM CONTROL

If any warning bits are set, the warning flag is triggered.

If any alarm bits are set, the alarm flag is triggered.

The master alarm bool is set if the alarm flag is triggered or if the E-stop is pressed. It is toggled when set.

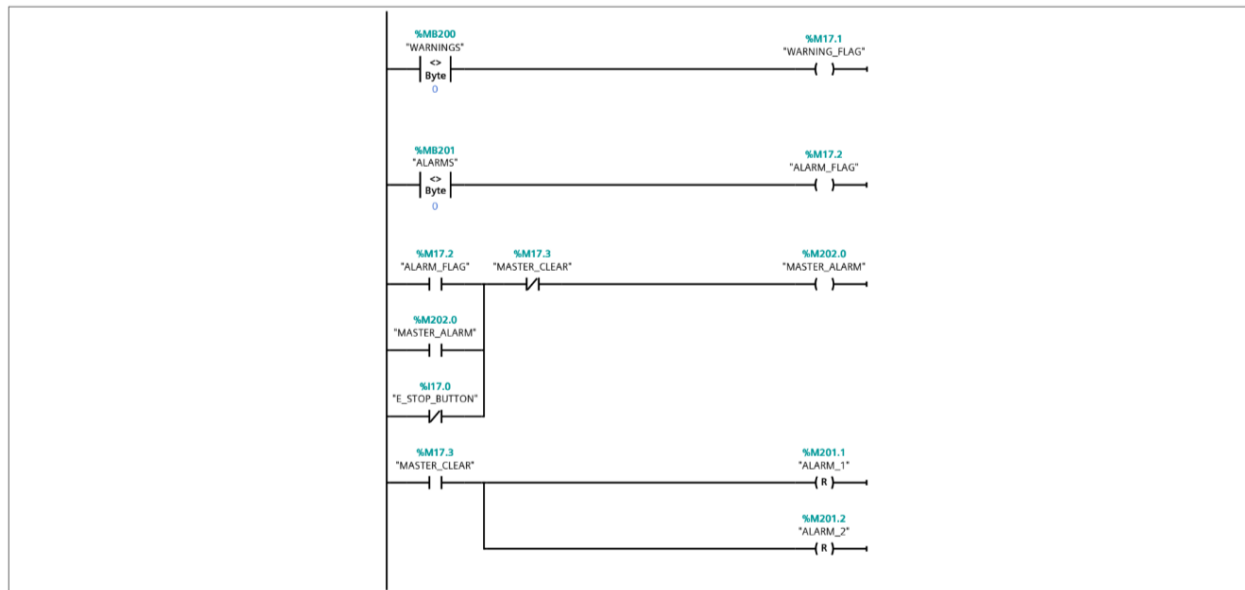


Figure 36 - Network 8

6.2.12 NETWORK 12: WEIGHT ERROR STATE

The scales cause an alarm if:

The weight is not measured as 0 - 0.1kg. This is the weight stored when the scales has nothing on it.

The weight is not measured as 4kg +/-5% (small box).

The weight is not measured as 5kg +/-5% (medium box).

The weight at the scales should be a known value and if this is not the case then the alarm must be triggered as something unknown is going down the line.

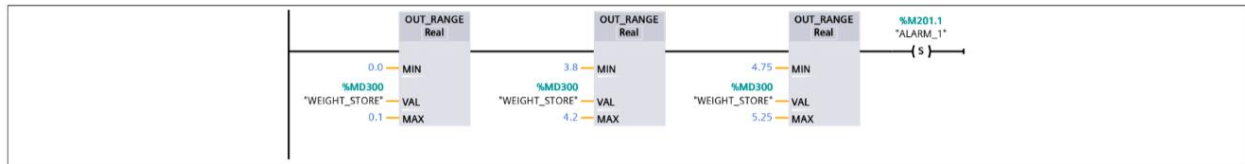


Figure 40 - Network 12

6.2.13 NETWORK 13: CONVEYOR 1 IO ERROR COUNTER

The output and input boxes on conveyor 1 are counted. If balance is not maintained, it indicates boxes going missing on the line. A warning is then triggered if the count is above the set value on the HMI.

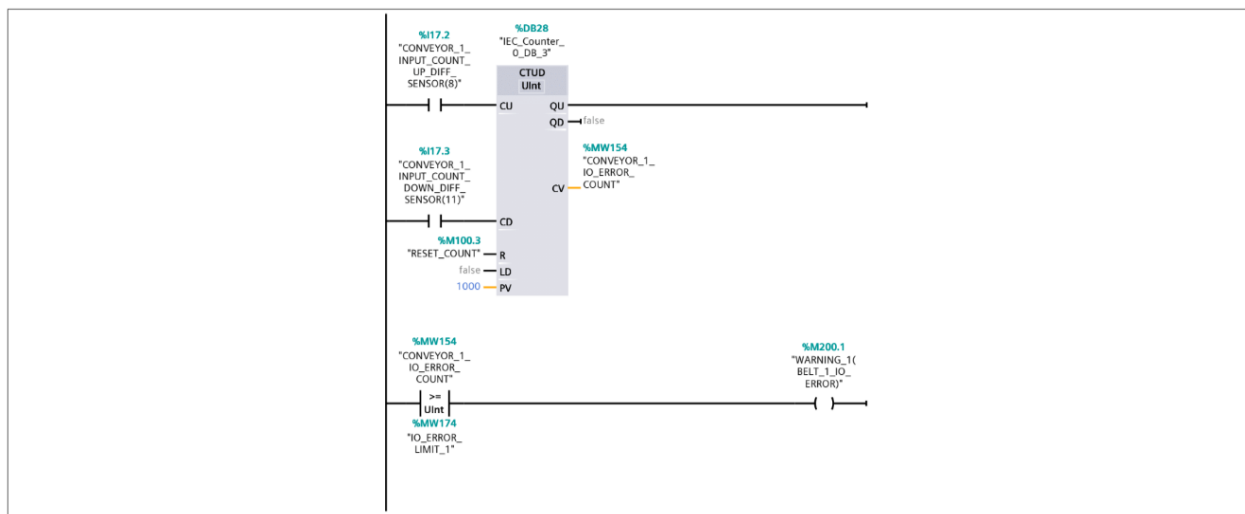


Figure 41 - Network 13

6.2.14 NETWORK 14: CONVEYOR 2 IO ERROR COUNTER

The output and input boxes on conveyor 2 are counted. If balance is not maintained, it indicates boxes going missing on the line. A warning is then triggered if the count is above the set value on the HMI.

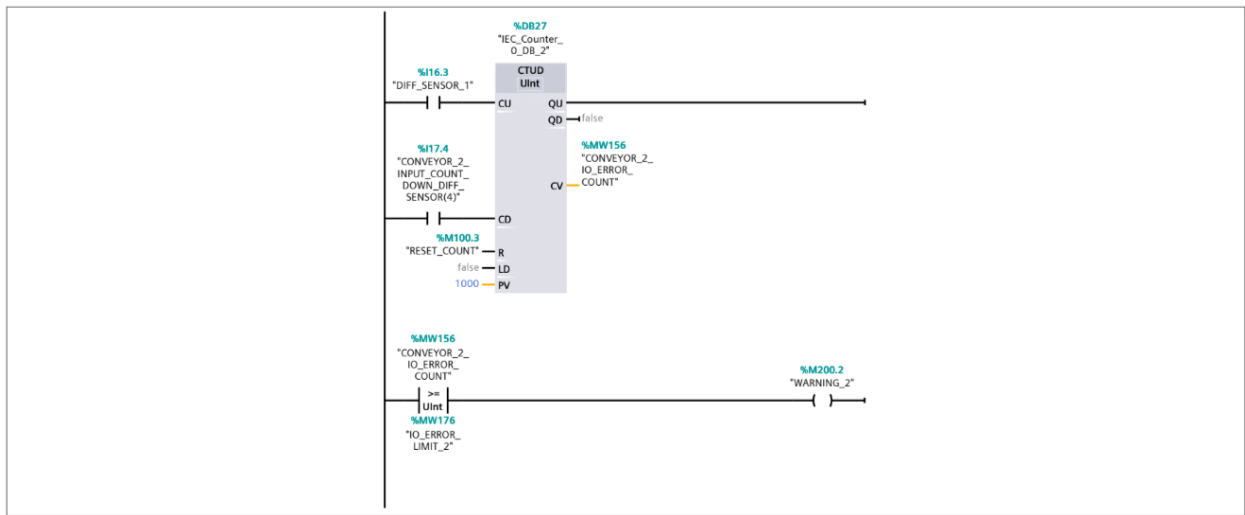


Figure 42 - Network 14

6.2.15 NETWORK 15: CONVEYOR 3 IO ERROR COUNTER

The output and input boxes on conveyor 3 are counted. If balance is not maintained, it indicates boxes going missing on the line. A warning is then triggered if the count is above the set value on the HMI.

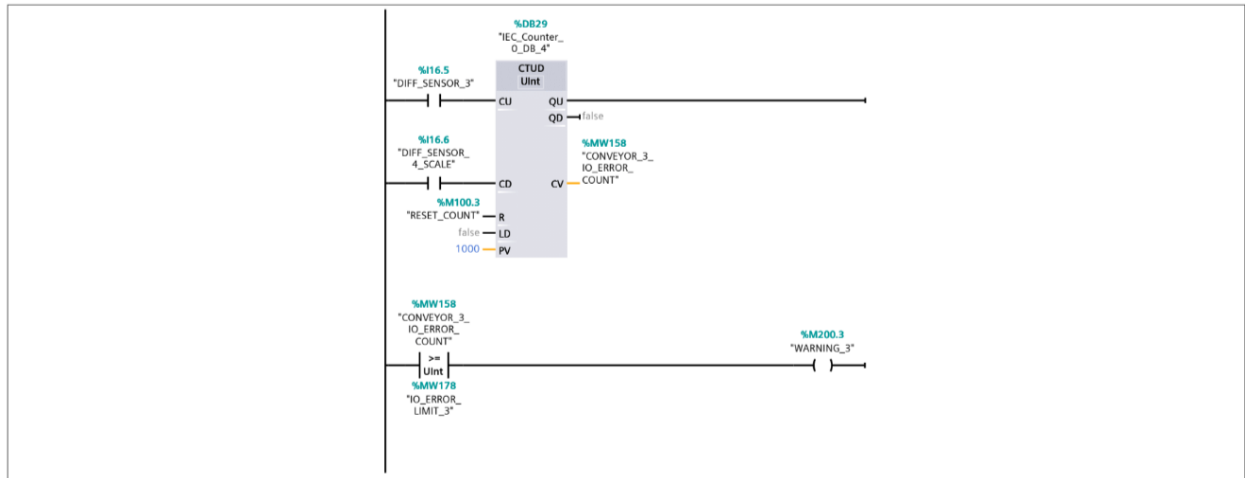


Figure 43 - Network 15

6.2.16 NETWORK 16: CONVEYOR 4 IO ERROR COUNTER

The output and input boxes on conveyor 4 are counted. If balance is not maintained, it indicates boxes going missing on the line. A warning is then triggered if the count is above the set value on the HMI.

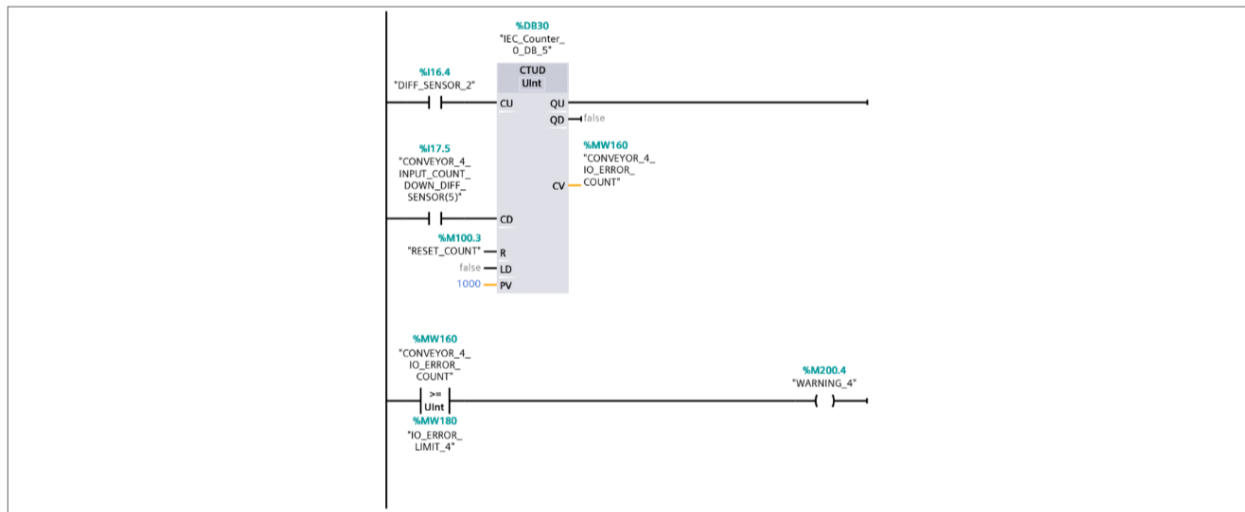


Figure 44 - Network 16

6.2.17 NETWORK 17: CONVEYOR 5 IO ERROR COUNTER

The output and input boxes on conveyor 5 are counted. If balance is not maintained, it indicates boxes going missing on the line. A warning is then triggered if the count is above the set value on the HMI.

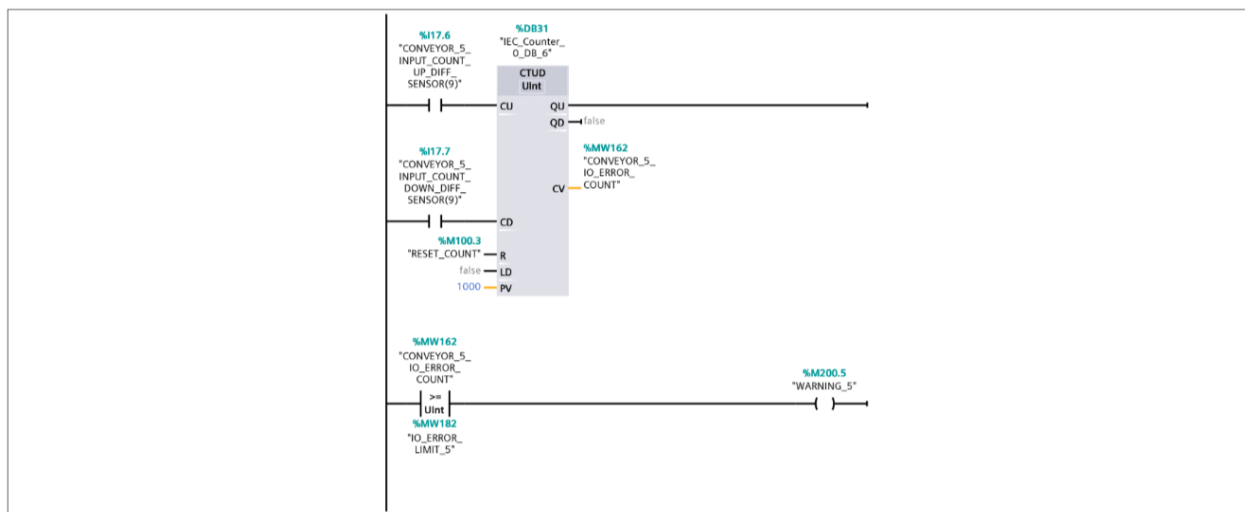


Figure 45 - Network 17

6.2.18 NETWORK 18: CONVEYOR 6 IO ERROR COUNTER

The output and input boxes on conveyor 6 are counted. If balance is not maintained, it indicates boxes going missing on the line. A warning is then triggered if the count is above the set value on the HMI.

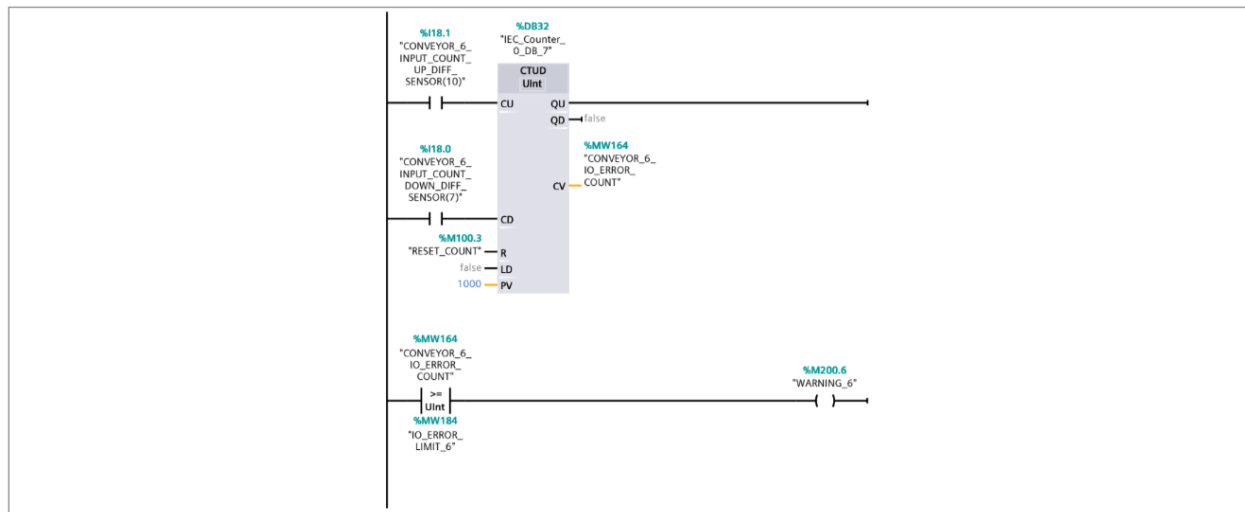


Figure 46 - Network 18

6.2.19 NETWORK 19: IO ALARM

If any of the IO counts are greater than 4. Then alarm 2 is triggered. The warning code can be used to find more detail of which one caused the error.

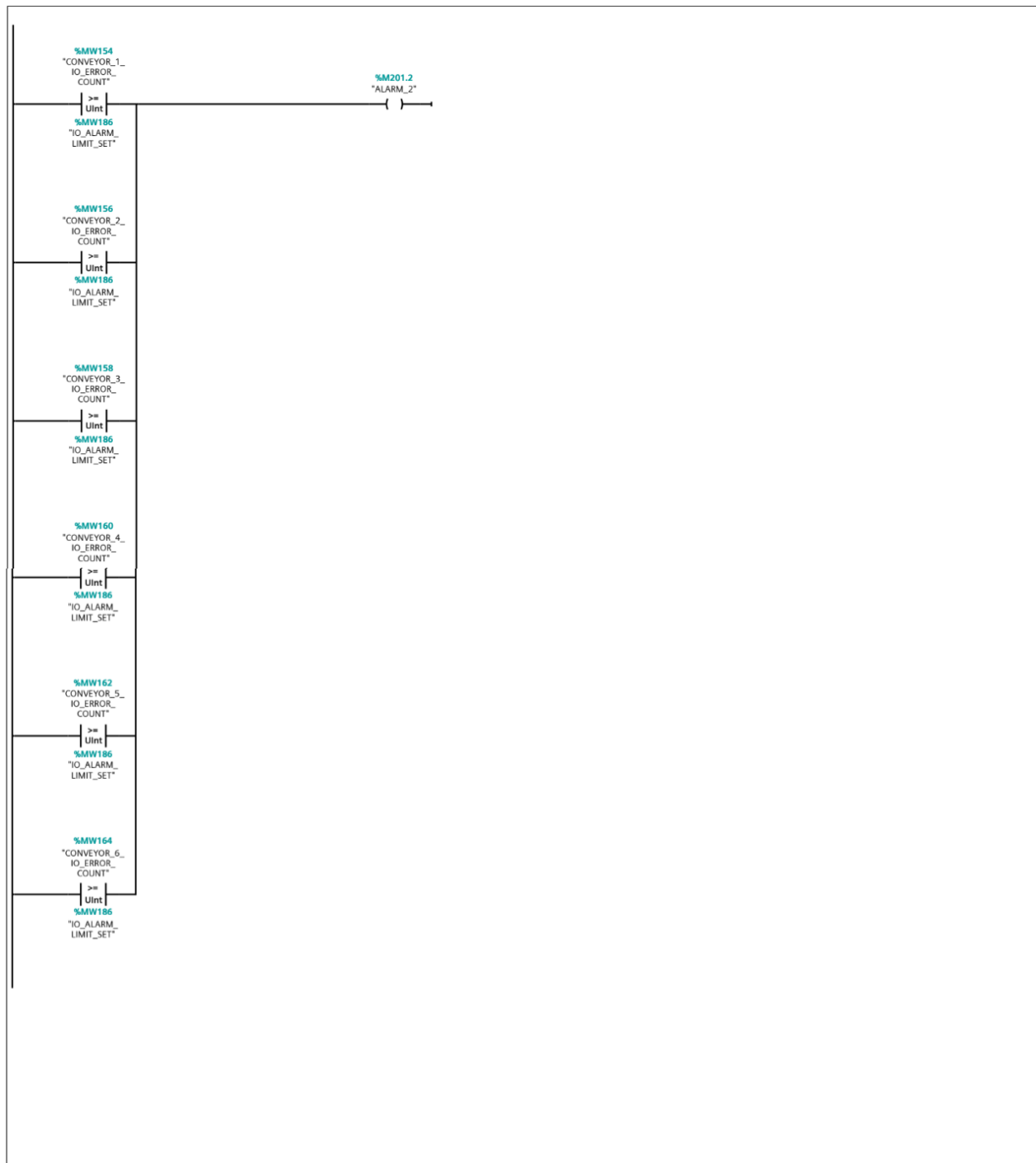


Figure 47 - Network 21

6.2.20 NETWORK 20: ALARM BEACON

The alarm beacon is triggered by the master alarm.



Figure 48 - Network 20

6.2.21 NETWORK 21: OUTPUT COUNTS

The output ramp of each path has a diff sensor. This is used to count the output of each product. This count is displayed on the HMI screen.

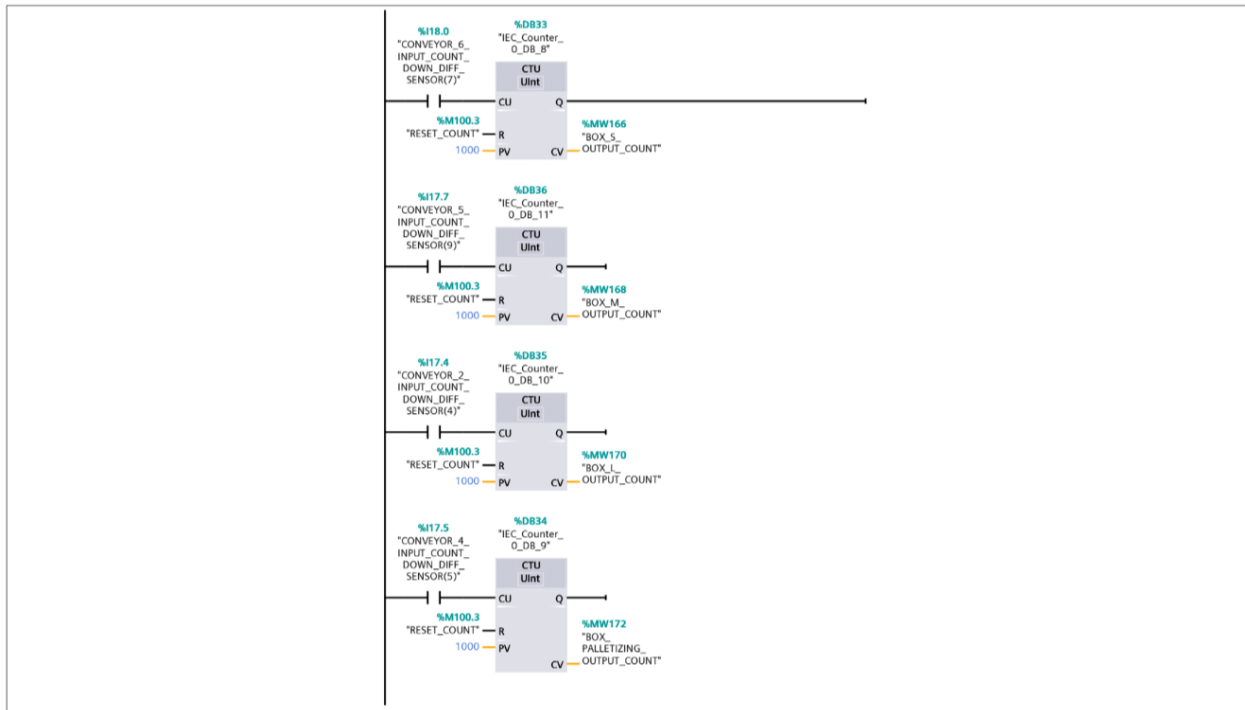


Figure 49 - Network 21

6.2.22 NETWORK 22: STACK LIGHT CONTROL

Stack lights and virtual stack lights are controlled in this section.

Green stack lights turn on only when the system is operating and there are no warnings or alarms set.

Yellow stack lights turn on only when the system is operating and there are warnings set but no alarms.

Red stack lights turn on only when there are alarms triggered by the system.

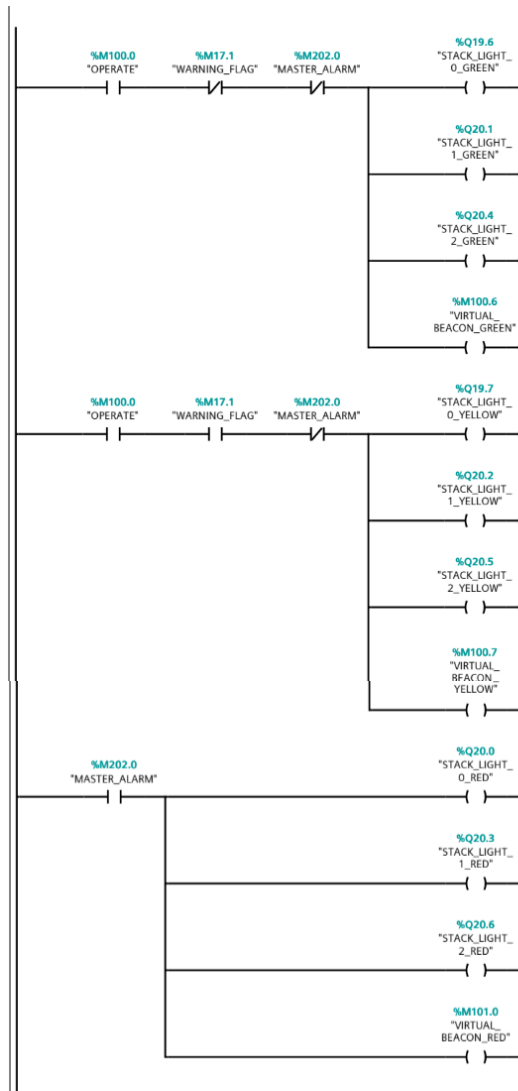


Figure 50 - Network 22

6.3 TAG LIST

A link to the full tag list pdf can be found in the APPENDICES section. The tag list can also be seen in Figure 51, Figure 52, Figure 53, Figure 54, and Figure 55.

	Name	Data type	Address	Retain	Access- ible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engi- neering	Supervision	Comment
	START_BUTTON	Bool	%I16.0	False	True	True	True		
	STOP_BUTTON	Bool	%I16.1	False	True	True	True		
	DIFF_SENSOR_1	Bool	%I16.3	False	True	True	True		
	DIFF_SENSOR_2	Bool	%I16.4	False	True	True	True		
	DIFF_SENSOR_3	Bool	%I16.5	False	True	True	True		
	DIFF_SENSOR_4_SCALE	Bool	%I16.6	False	True	True	True		
	E_STOP_BUTTON	Bool	%I17.0	False	True	True	True		
	CONVEYOR_1_IN- PUT_COUNT_UP_DIFF_SENSOR(8)	Bool	%I17.2	False	True	True	True		
	CONVEYOR_1_IN- PUT_COUNT_DOWN_DIFF_SEN- SOR(11)	Bool	%I17.3	False	True	True	True		
	CONVEYOR_2_IN- PUT_COUNT_DOWN_DIFF_SEN- SOR(4)	Bool	%I17.4	False	True	True	True		
	CONVEYOR_4_IN- PUT_COUNT_DOWN_DIFF_SEN- SOR(5)	Bool	%I17.5	False	True	True	True		
	CONVEYOR_5_IN- PUT_COUNT_UP_DIFF_SENSOR(9)	Bool	%I17.6	False	True	True	True		
	CONVEYOR_5_IN- PUT_COUNT_DOWN_DIFF_SEN- SOR(9)	Bool	%I17.7	False	True	True	True		
	CONVEYOR_6_IN- PUT_COUNT_DOWN_DIFF_SEN- SOR(7)	Bool	%I18.0	False	True	True	True		

Figure 51 - Tag list - Page 1

	Name	Data type	Address	Retain	Access- ible from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engi- neering	Supervision	Comment
	STACK_LIGHT_0_GREEN	Bool	%Q19.6	False	True	True	True		
	STACK_LIGHT_0_YELLOW	Bool	%Q19.7	False	True	True	True		
	STACK_LIGHT_0_RED	Bool	%Q20.0	False	True	True	True		
	STACK_LIGHT_1_GREEN	Bool	%Q20.1	False	True	True	True		
	STACK_LIGHT_1_YELLOW	Bool	%Q20.2	False	True	True	True		
	STACK_LIGHT_1_RED	Bool	%Q20.3	False	True	True	True		
	STACK_LIGHT_2_GREEN	Bool	%Q20.4	False	True	True	True		
	STACK_LIGHT_2_YELLOW	Bool	%Q20.5	False	True	True	True		
	STACK_LIGHT_2_RED	Bool	%Q20.6	False	True	True	True		
	SCALES_DISPLAY	Real	%QD30	False	True	True	True		
	WARNING_FLAG	Bool	%M17.1	False	True	True	True		
	ALARM_FLAG	Bool	%M17.2	False	True	True	True		
	MASTER_CLEAR	Bool	%M17.3	False	True	True	True		
	OPERATE	Bool	%M100.0	False	True	True	True		
	DIFF1	Bool	%M100.1	False	True	True	True		
	DIFF2	Bool	%M100.2	False	True	True	True		
	RESET_COUNT	Bool	%M100.3	False	True	True	True		
	GENERAL_RESET	Bool	%M100.4	False	True	True	True		
	DIFF3	Bool	%M100.5	False	True	True	True		
	VIRTUAL_BEACON_GREEN	Bool	%M100.6	False	True	True	True		
	VIRTUAL_BEACON_YELLOW	Bool	%M100.7	False	True	True	True		
	VIRTUAL_BEACON_RED	Bool	%M101.0	False	True	True	True		
	PUSHER_1_COUNT	UInt	%MW150	False	True	True	True		
	PUSHER_2_COUNT	UInt	%MW152	False	True	True	True		

Figure 53 - Tag list - Page 3

	Name	Data type	Address	Retain	Accessi- ble from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engi- neering	Supervision	Comment
	CONVEYOR_1_IO_ERROR_COUNT	UInt	%MW154	False	True	True	True		
	CONVEYOR_2_IO_ERROR_COUNT	UInt	%MW156	False	True	True	True		
	CONVEYOR_3_IO_ERROR_COUNT	UInt	%MW158	False	True	True	True		
	CONVEYOR_4_IO_ERROR_COUNT	UInt	%MW160	False	True	True	True		
	CONVEYOR_5_IO_ERROR_COUNT	UInt	%MW162	False	True	True	True		
	CONVEYOR_6_IO_ERROR_COUNT	UInt	%MW164	False	True	True	True		
	BOX_S_OUTPUT_COUNT	UInt	%MW166	False	True	True	True		
	BOX_M_OUTPUT_COUNT	UInt	%MW168	False	True	True	True		
	BOX_L_OUTPUT_COUNT	UInt	%MW170	False	True	True	True		
	BOX_PALLETIZING_OUT- PUT_COUNT	UInt	%MW172	False	True	True	True		
	IO_ERROR_LIMIT_1	UInt	%MW174	False	True	True	True		
	IO_ERROR_LIMIT_2	UInt	%MW176	False	True	True	True		
	IO_ERROR_LIMIT_3	UInt	%MW178	False	True	True	True		
	IO_ERROR_LIMIT_4	UInt	%MW180	False	True	True	True		
	IO_ERROR_LIMIT_5	UInt	%MW182	False	True	True	True		
	IO_ERROR_LIMIT_6	UInt	%MW184	False	True	True	True		
	IO_ALARM_LIMIT_SET	UInt	%MW186	False	True	True	True		
	WARNINGS	Byte	%MB200	False	True	True	True		
	WARNING_1(BELT_1_IO_ERROR)	Bool	%M200.1	False	True	True	True		
	WARNING_2	Bool	%M200.2	False	True	True	True		
	WARNING_3	Bool	%M200.3	False	True	True	True		
	WARNING_4	Bool	%M200.4	False	True	True	True		
	WARNING_5	Bool	%M200.5	False	True	True	True		
	WARNING_6	Bool	%M200.6	False	True	True	True		

Figure 54 - Tag list - Page 4

	Name	Data type	Address	Retain	Accessi- ble from HMI/OPC UA/Web API	Writable from HMI/OPC UA/Web API	Visible in HMI engi- neering	Supervision	Comment
	WARNING_7	Bool	%M200.7	False	True	True	True		
	ALARMS	Byte	%MB201	False	True	True	True		
	ALARM_1	Bool	%M201.1	False	True	True	True		
	ALARM_2	Bool	%M201.2	False	True	True	True		
	ALARM_3	Bool	%M201.3	False	True	True	True		
	ALARM_4	Bool	%M201.4	False	True	True	True		
	ALARM_5	Bool	%M201.5	False	True	True	True		
	ALARM_6	Bool	%M201.6	False	True	True	True		
	MASTER_ALARM	Bool	%M202.0	False	True	True	True		
	READY_TO_WEIGH	Bool	%M202.1	False	True	True	True		
	WEIGHT_STORE	Real	%MD300	False	True	True	True		

Figure 55 - Tag list - Page 5

6.4 HMI

Links to full documentation of HMI screens can be found in the APPENDICES section.

6.4.1 ROOT SCREEN

The root screen (Figure 56) is the default screen displayed when the PLC program is running.

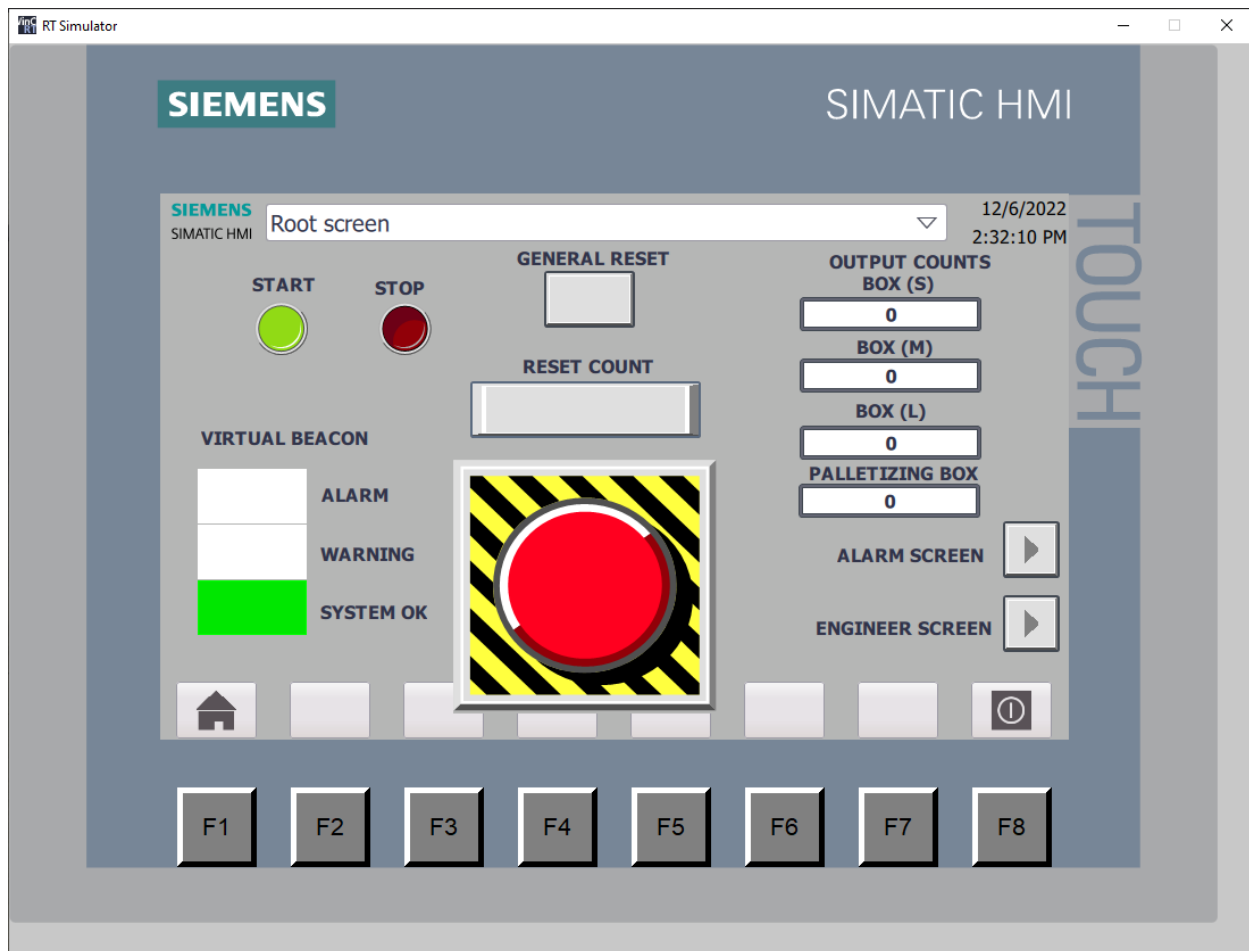


Figure 56 - HMI Root Screen

A large E-stop button is provided which will stop the line in the case of an emergency. Start and stop buttons are available to start and stop the process and are an alternative to the physical buttons on the control panel. The general reset button can be used to stop the conveyors on the line.

A count for each type of box that has reached the output is displayed so users can easily track the output of the line. The reset count button is used to reset all counts on the program.

The virtual beacon will behave exactly the same as the beacons on the line.

Buttons are provided to change to the alarm and engineering screen.

6.4.2 ENGINEER SCREEN

The engineering screen allows a user to adjust some key parameters in the program and also provides a method to reset any toggled error states if they are not being held on. This is required for resetting the machine after an alarm or warning has occurred.

Buttons are provided to change to both the home and alarm screen.

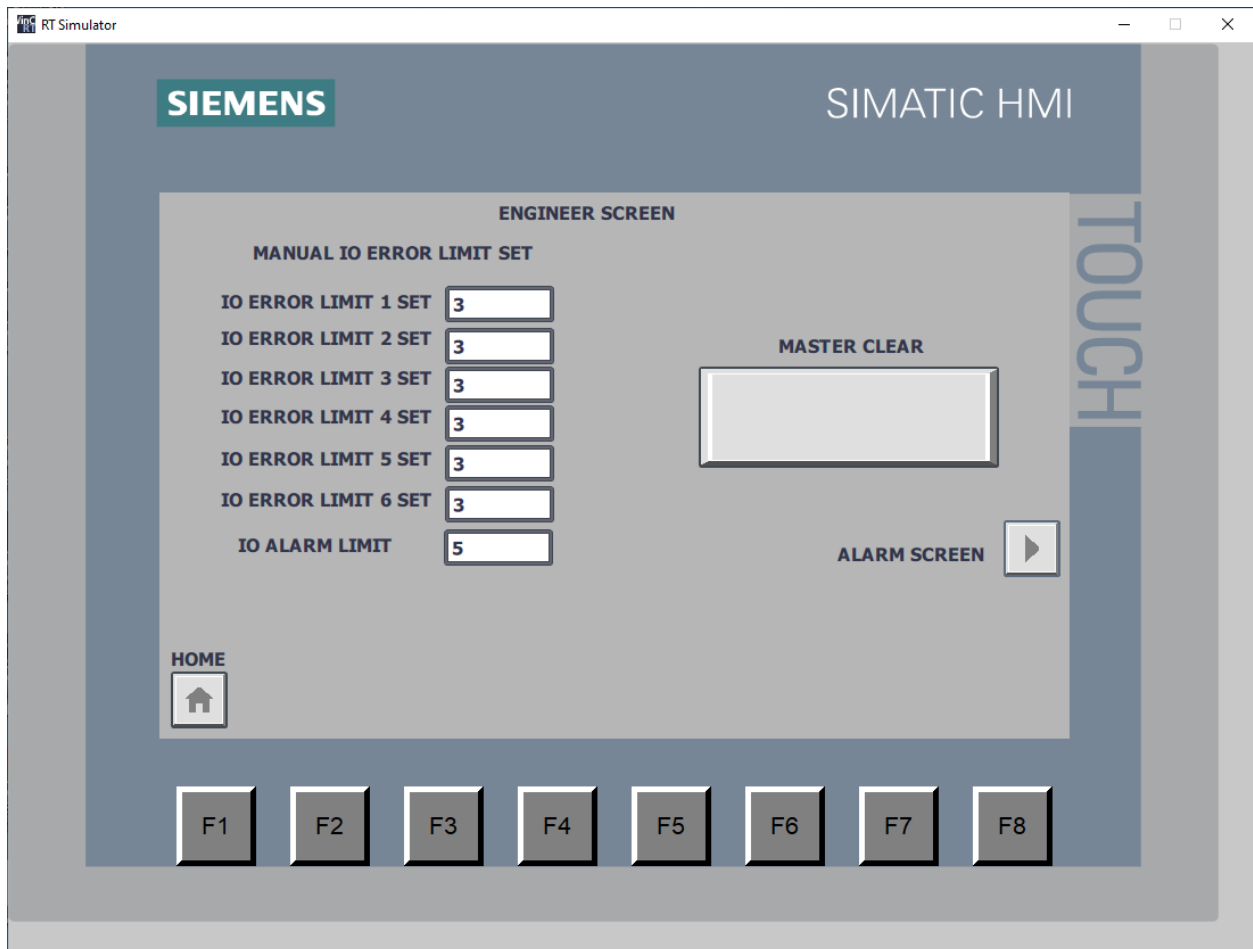


Figure 57 - HMI Engineer Screen

6.4.3 ALARM SCREEN

The alarm screen is used to display any error states within the machine. Green indicates no error and red indicates an error state. If any warnings or alarms have been detected, the relevant light will change state.

Buttons can be used to return to the root screen or go to the engineering screen. There is also a button which can change the HMI over to a graphical alarm screen.

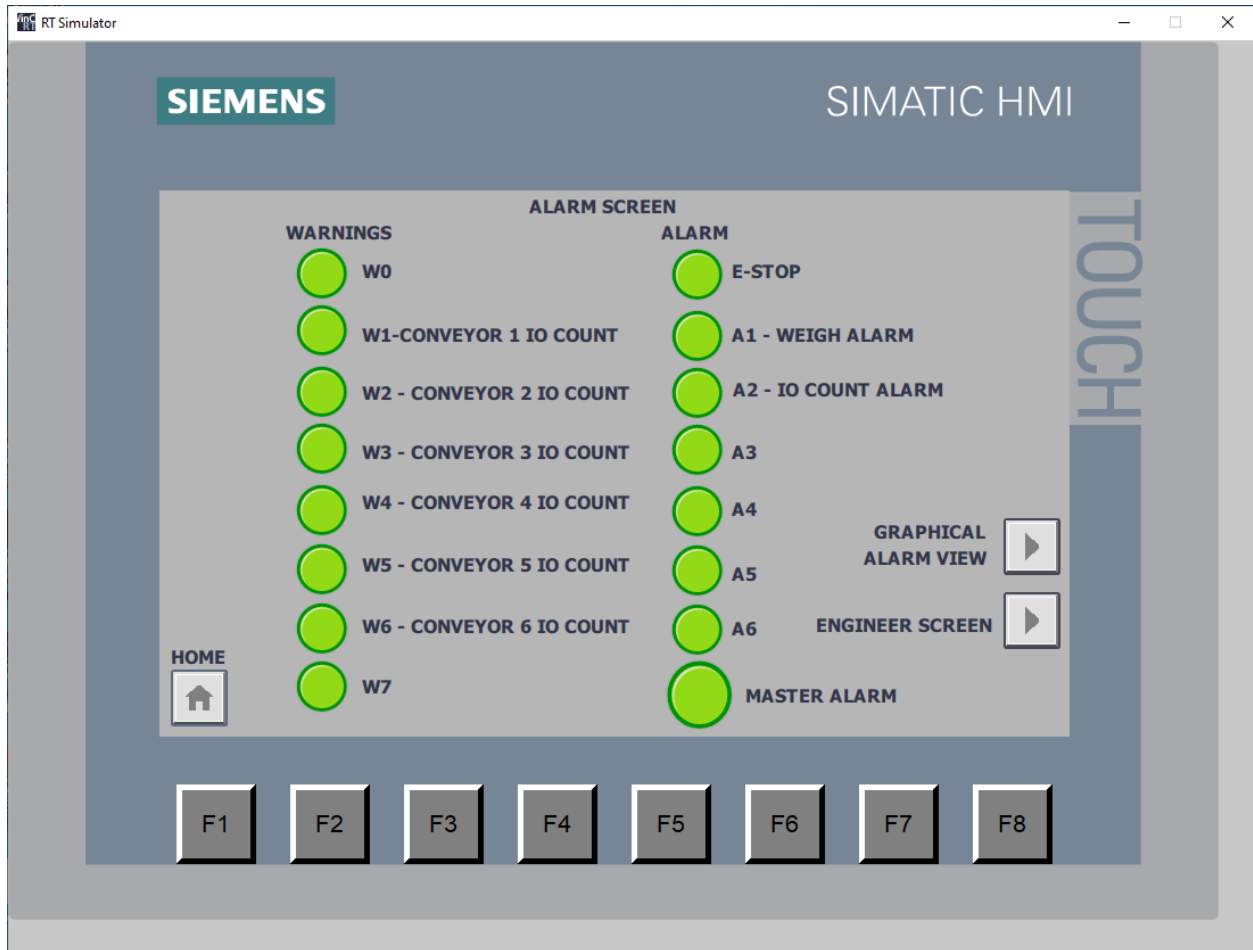


Figure 58 - Alarm Screen

6.4.4 GRAPHICAL ALARM SCREEN

The purpose of the graphical alarm screen is to allow a user to easily detect where the IO warnings have occurred at a glance. If the alarm threshold has been met, the alarm indicator will go red next to the part of the line that caused the error. The state of the warnings can be used to determine where the errors have occurred. The error counts can also be viewed from this screen.

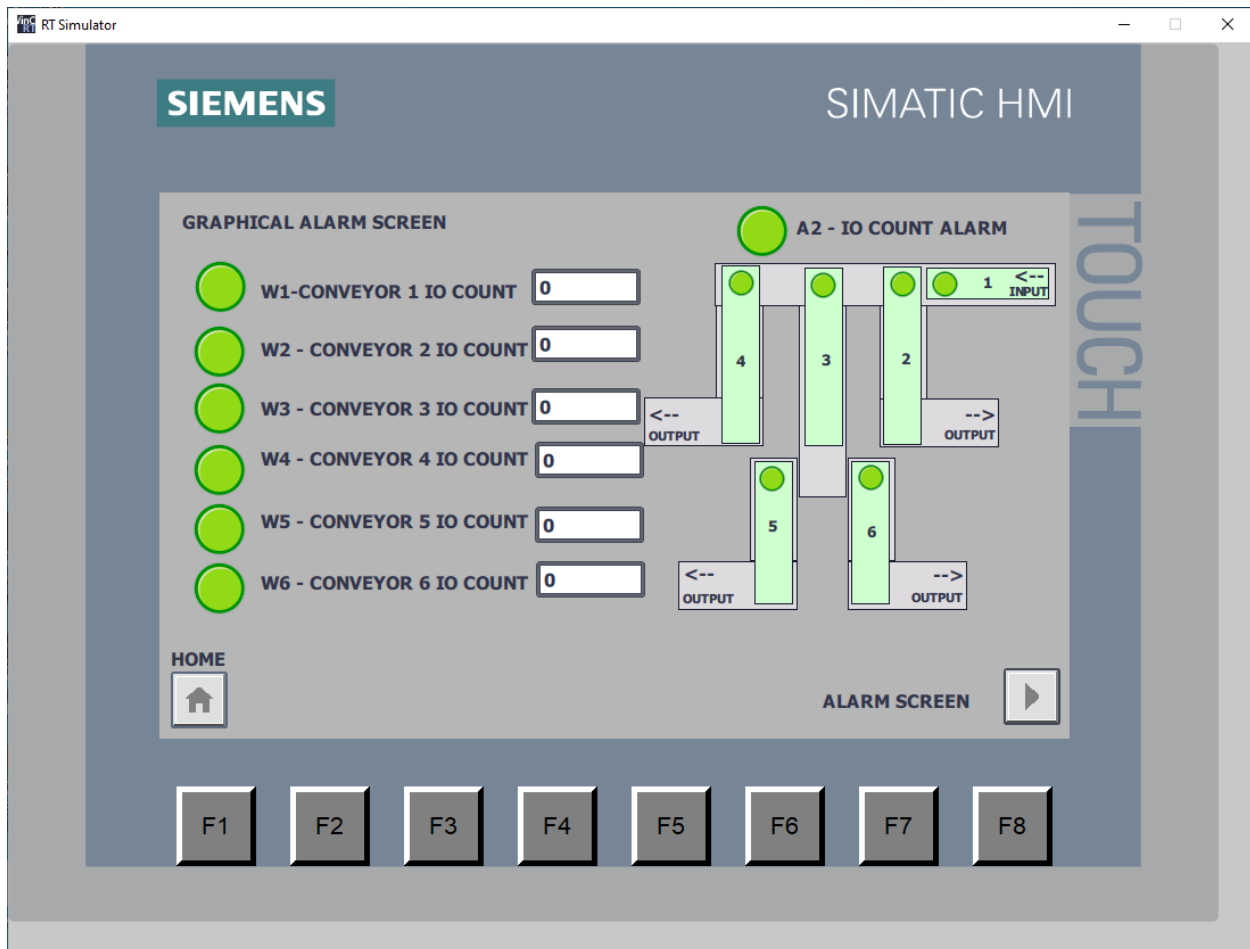


Figure 59 - Graphical Alarm Screen

7 DISCUSSION

One improvement that could be made to the current factory design would be more e-stops. The current factory has only one physical one on the control panel. This could be problematic if the line is required to stop due to an emergency in a location that is not close to the e-stop. Multiple e-stops should be placed in locations that are easily accessible and where people would be working. This would be determined after a full risk assessment of the process was carried out in accordance with HSE guidelines [7].

E-stops should not be included in the HMI. Firstly, the ISO standards state the e-stop should be easily activated by the palm of a hand. If the HMI was on a different screen, then it would compromise the e-stop, also it may be difficult to press the HMI screen accurately with the palm of your hand. The ISO standards also state that "The emergency stop function shall be available and operational at all times". If there was a problem with the HMI screen, then the e-stop would not be available, however the process could still function. This may be unlikely, but a more robust solution would be to hardwire all e-stops into a safety box that would isolate power to any motors used across the process. This would improve the stop category of the system to a type 0 [8].

Currently the engineering HMI screen is accessible to anyone. Anyone can change the error threshold values that are in the program. This should have some kind of security built in to prevent unauthorized users from accessing it. One solution would be to add a password access the screen.

8 CONCLUSION

Overall, the factory design was successful as all of the requirements specified in the brief were met. Some improvements could be made however to make the process more secure and user friendly.

9 REFERENCES

- [1] W. Bolton, "1 Programmable Logic Controllers," in *Programmable Logic Controllers - Fourth Edition*, Elsevier Newnes, 2006, pp. 3-13.
- [2] M. A. L. CEng and D. J. W. CEng, "Programmable Controllers | 16.1 Introduction," in *Electrical Engineer's Reference Book - Sixteenth edition*, Newnes, 2003, pp. 16/4-16/5.
- [3] M. A. L. CEng and D. J. W. CEng, "Programmable Controllers | 16.1 Introduction," in *Electrical Engineer's Reference Book - Sixteenth edition*, Newnes, 2003, p. 16/7.
- [4] M. A. L. CEng and D. J. W. CEng, "Programmable Controllers | 16.3 Programming Methods," in *Electrical Engineer's Reference Book - Sixteenth edition*, Newnes, 2003, pp. 16/13 - 16/27.
- [5] Ladder Logic World, "PLC Architecture and Types: With Comparison Table," Ladder Logic World, [Online]. Available: <https://ladderlogicworld.com/plc-architecture/#:~:text=PLC%20Types%20Comparison%20Table%20%20%20PLC%20TYPES,%20%20Modules%20%208%20more%20rows%20>. [Accessed 13 12 2022].
- [6] Autosen, "Diffuse reflection sensors - autosen.com," Autosen, [Online]. Available: <https://autosen.com/en/Position-Sensors/Photoelectric-sensors/Diffuse-reflection-sensor#:~:text=Diffuse%20reflection%20sensors%20use%20reflection,sensor%20and%20the%20sensor%20switches..> [Accessed 16 12 2022].
- [7] N. Worsell, "HSL/2000/18 - MACHINERY RISK ASSESSMENT VALIDATION LITERATURE REVIEW," [Online]. Available: https://www.hse.gov.uk/research/hsl_pdf/2000/hsl00-18.pdf. [Accessed 17 12 2022].
- [8] ISO, "ISO - ISO 13850:2015 - Safety of machinery — Emergency stop function — Principles for design," 11 2015. [Online]. Available: <https://www.iso.org/standard/59970.html>. [Accessed 17 12 2022].
- [9] RS Components Ltd., "6ED1052-2CC08-0BA1 | Siemens LOGO! PLC CPU - 12 Inputs, 4 Outputs, Transistor, For Use With LOGO! 8.3, Ethernet Networking, Ethernet | RS," RS Components Ltd., [Online]. Available: https://uk.rs-online.com/web/p/plcs-programmable-logic-controllers/2097105?cm_mmc=UK-PLA-DS3A_-google_-CSS_UK_EN_Automation_%26_Control_Gear_Whoop_-PLCs+-+Programmable+Logic+Controllers_Whoop_-2097105&matchtype=&pla-334669696251&cq_src=google_ads&cq_c. [Accessed 13 12 2022].
- [10] RS Components Ltd., "CJ2M-CPU33 | Omron CJ2M PLC CPU, Pulse, For Use With CJ2M Series, Ethernet Networking, Computer Interface | RS," RS Components Ltd., [Online]. Available: https://uk.rs-online.com/web/p/plcs-programmable-logic-controllers/7462884?cm_mmc=UK-

PPC-DS3A-_-bing-_-DSA_UK_EN_Automation_%26_Control_Gear_Index-_-PLCs+-+Programmable+Logic+Controllers%7C+Products-_-DYNAMIC+SEARCH+ADS&matchtype=b&dat-2323130630686154:lo. [Accessed 13 12 2022].

- [11] Siemens, "SIMATIC ET 200SP | Distributed IO | Siemens Global," [Online]. Available: <https://new.siemens.com/global/en/products/automation/systems/industrial/io-systems/et-200sp.html#MultiFieldbusInterfaceModule>. [Accessed 13 12 2022].
- [12] Siemens , "SIMATIC HMI Basic Panels | SIMATIC HMI Panels | Siemens Global," Siemens , [Online]. Available: <https://new.siemens.com/global/en/products/automation/simatic-hmi/panels/basic-panels.html>. [Accessed 13 12 2022].

10 APPENDICES

10.1 LINKS

As some documents are very large in size, links to pdfs uploaded to Dropbox are provided so the reader can zoom in for a closer look.

DOCUMENT NAME	LINK
PLC ASSIGNMENT 1 - FLOW CHART.pdf	link
PLC_ASSIGNMENT_MAIN_PROGRAM_BLOCK.pdf	link
PLC_ASSIGNMENT_TAG_LIST.pdf	link
PLC_ASSIGNMENT_HMI_ROOT_SCREEN.pdf	link
PLC_ASSIGNMENT_HMI_ALARM_SCREEN.pdf	link
PLC_ASSIGNMENT_HMI_ENGINEER_SCREEN.pdf	link
PLC_ASSIGNMENT_HMI_IO_ALARM_SCREEN.pdf	link
Project 4.1.factoryio	link
project 4.zip	link

10.2 FIGURES

Figure 1 – Early PLC Programming keyboard [3].....	5
Figure 2 - Ladder Logic Example [4]	6
Figure 3 – Ladder logic example with complex [4] statements [4].....	6
Figure 4 - PLC Architecture [1]	7

Figure 5 - Fixed PLC Example (Siemens LOGO!) [9].....	8
Figure 6 - Modular PLC (Omron CJ2M) [10].....	9
Figure 7 - Distributed PLC Example (SIMATIC ET 200SP) [11].....	10
Figure 8 - HMI Example [12]	11
Figure 9 - Large Box.....	12
Figure 10 - Medium Box.....	12
Figure 11 - Small Box.....	13
Figure 12 - Palletising Box	13
Figure 13 - Factory IO Curved Conveyor	14
Figure 14 - Factory IO Straight Conveyor	14
Figure 15- Factory IO Diffuse Sensor.....	15
Figure 16 - Factory IO Pusher (in)	16
Figure 17 - Factory IO Pusher (out).....	16
Figure 18 - Factory IO Weighing Scales.....	17
Figure 19 – Factory IO Pop Up Wheel Sorter and Output Directions	17
Figure 20 - Factory IO Overview - Annotated Blocks	18
Figure 21 - Factory IO Overview – Top View– Actual Design.....	19
Figure 22 - Factory IO Overview – Side View – Actual Design	20
Figure 23 - Diffuse Sensors at different heights	21
Figure 24 – Conveyor Scale and pop up wheel sorter	22
Figure 25 - Control Panel with indicators and buttons.....	23
Figure 26 - Beacon States (green, yellow, and red)	24
Figure 27 - IO list used in Factory IO	26
Figure 28 - Network 1	28
Figure 29 - Network 2	28
Figure 30 - Network 3	29
Figure 31 - Network 4 Part 1	30
Figure 32 - Network 4 Part 2	31
Figure 33 - Network 5	32
Figure 34 - Network 6	32
Figure 35 - Network 7	33
Figure 36 - Network 8	33
Figure 37 - Network 9	34
Figure 38 - Network 10	34
Figure 39 - Network 11	34
Figure 40 - Network 12	35
Figure 41 - Network 13	35
Figure 42 - Network 14	36
Figure 43 - Network 15	36
Figure 44 - Network 16	37
Figure 45 - Network 17	37
Figure 46 - Network 18	38
Figure 47 - Network 21	39
Figure 48 - Network 20	40

Figure 49 - Network 21	40
Figure 50 - Network 22	41
Figure 51 - Tag list - Page 1	42
Figure 52 - Tag list - Page 2	43
Figure 53 - Tag list - Page 3	43
Figure 54 - Tag list - Page 4	44
Figure 55 - Tag list - Page 5	44
Figure 56 - HMI Root Screen	45
Figure 57 - HMI Engineer Screen	46
Figure 58 - Alarm Screen	47
Figure 59 - Graphical Alarm Screen	48