



APPLIED PROGRAMMABLE LOGIC CONTROLLERS

ACDA6000

Assignment 1 – May 2024

Tomos Hall

2219188@student.uwtsd.ac.uk

Abstract

A PLC program was designed to operate a factory simulation. Four box sizes were sorted by size and weight into separate sections. Different functions were utilised such as pulse timers, delays, and computational operations. Using sensors and pushers the boxes were sent down separate conveyors depending on their parameters.

A flow chart was produced to outline the programs operation. A HMI screen was designed to allow an engineer to operate the factory. An engineer's screen was designed to set warning and alarm parameters to the program. A warning screen was designed to allow the engineer to determine the location of faults in the system.

Table of Contents

Abstract.....	1
Introduction.....	3
Background research	4
PLC development.....	4
PLC construction.....	5
PLC comparison.....	6
Factory design	11
.....	17
Factory flowchart.....	18
HMI screens	19
Warnings and alarm strategy	23
Ladder Program	27
PLC tags	49
Conclusion.....	53
Bibliography.....	54

Introduction

Throughout industry, automated processes are controlled using computers. Programmable logic controllers (PLCs) are a trusted method of control for countless operations across a range of industries. This report details the design and implementation of an automated box sorting process.

Different types of PLCs are researched and compared, highlighting their pros and cons, and explaining their functions. Using an industrial simulation tool, Factor IO, an automated sorting process was designed. An arrangement of conveyors, sensors, and pushers was constructed to sort four different boxes by size or weight.

Utilising siemens TIA software, a ladder program was written to operate the designed factory, HMI screens were utilised allowing the engineer to input data and control the system.

Background research

PLC development

Programmable logic controllers (PLCs) were first developed in 1969 [1, p. 7]. Since their invention they have been utilised across industry for automation, manufacturing, and control. Using a non-volatile, programmable memory, controlled by a microprocessor controller, instructions are implemented to carry out different functions [2]. These functions control machines and operations using logic, timers, counters, and arithmetic [1, p. 5].

Inputs to PLCs are commonly sensors or switches. When a sensor is input into the PLC it will activate the programmed output, this could be an electric motor or another control device such as a memory coil within the PLC [1, pp. 5,6] [3] [4]. Inputs and outputs are programmed by the engineer. The PLC performs a scanning cycle, this analyses the inputs and outputs implemented by the engineer and stores them in an input image table. The CPU then scans the program, linking the inputs and outputs to the instruction set in the program. Once an input from the table has been detected, the corresponding instruction set is carried out [3] [5]. This is a continuous cycle producing an efficient and reliable automated process [4].

PLCs were designed to replace electromechanical relay circuits used to automate manufacturing. These relay coils were physically wired, producing a complex circuit requiring a relay ladder diagram, detailing their operation [6, pp. 329-333]. For ease of transition, PLC programming was designed to follow the same language, ladder logic. Ladder logic is the main programming language used for PLCs. An example of a ladder program can be found in figure 1 [7]. The first rung of the ladder depicts a normally open input connecting to an output coil. The second rung depicts a logic AND function, where the output will only become energised when both normally open inputs are activated. Rung 3 shows an OR function, the output will be energised when either of the inputs are activated [1, pp. 121,122].

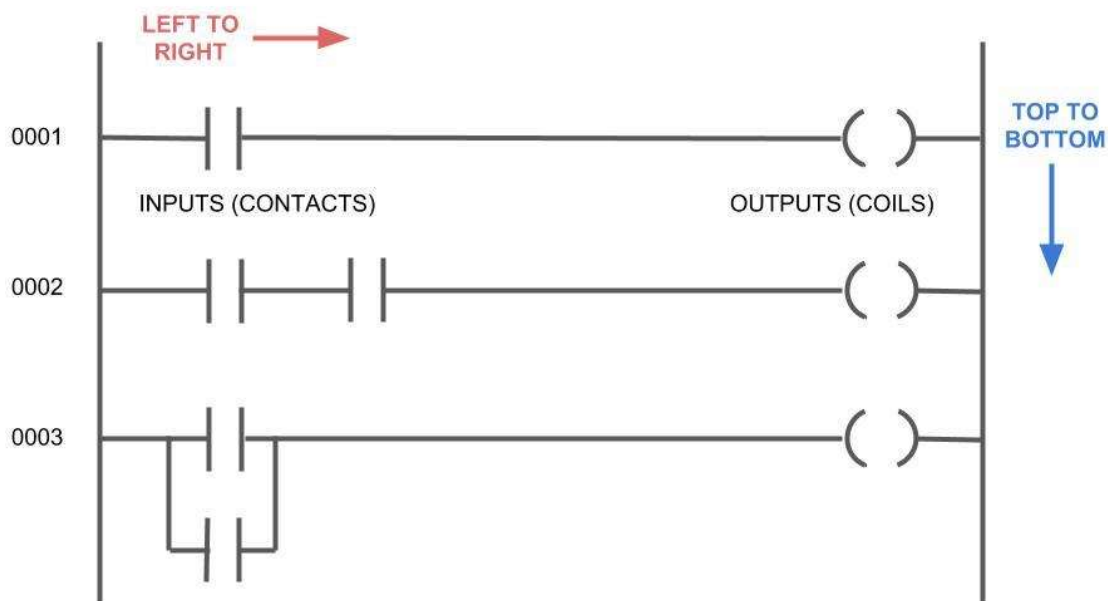


Figure 1 - Ladder program example [7]

PLC construction

There are multiple types of PLC available today, they all follow the same basic construction. Each type of PLC utilises a power supply, an input/output controller, programming device, memory, communication interface and CPU [1, p. 7]. This can be seen in figure 2 [1].

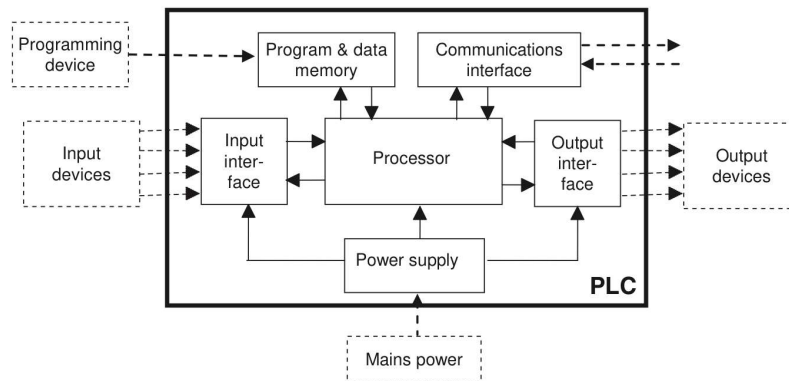


Figure 2 - PLC construction [1].

The two main types of PLC are fixed and modular [4]. A fixed type, or compact PLC is utilised for smaller applications. They offer a low number of fixed inputs and outputs. The compact PLC contains all the required components listed in figure 2 in a single unit. Compact PLCs are often more cost effective and simpler to use. They are low maintenance, utilised for smaller applications [4]. Due to their compact nature this type of PLC they are limited in their capabilities. Figure 3 shows an example of a compact PLC, the Omron CP1L [8].

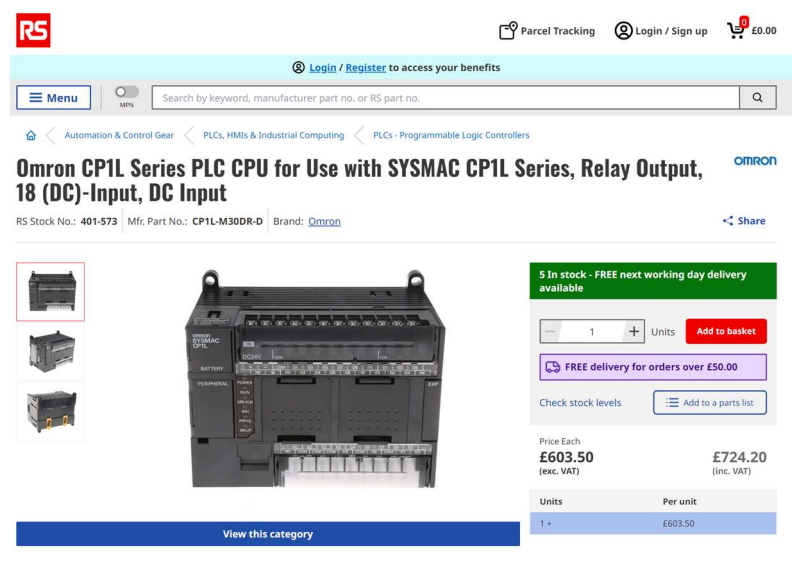


Figure 3 - Compact PLC, Omron CP1L [8] [1]

A modular PLC is made up of multiple modules which fit into a rack. Unlike the fixed PLC which offers screw in terminals for I/O, a modular PLC will require an I/O module which fits in the rack. All the features of a PLC listed in figure 2 are individually added to the rack. This type of PLC can be easily upgraded as any module can be interchanged. More memory, a faster CPU, a bigger power supply or more I/O ports can be added simply by upgrading the corresponding module [1] [9] [10, pp. 14,15]. This makes modular PLCs much more flexible in their abilities. As they can be easily upgraded to handle innovation. As the number of components required is larger the cost can also be expensive. An example of a modular PLC can be seen in figure 4, a Mitsubishi iQ-R series [11]. Figure 5 displays the modules they offer [11].

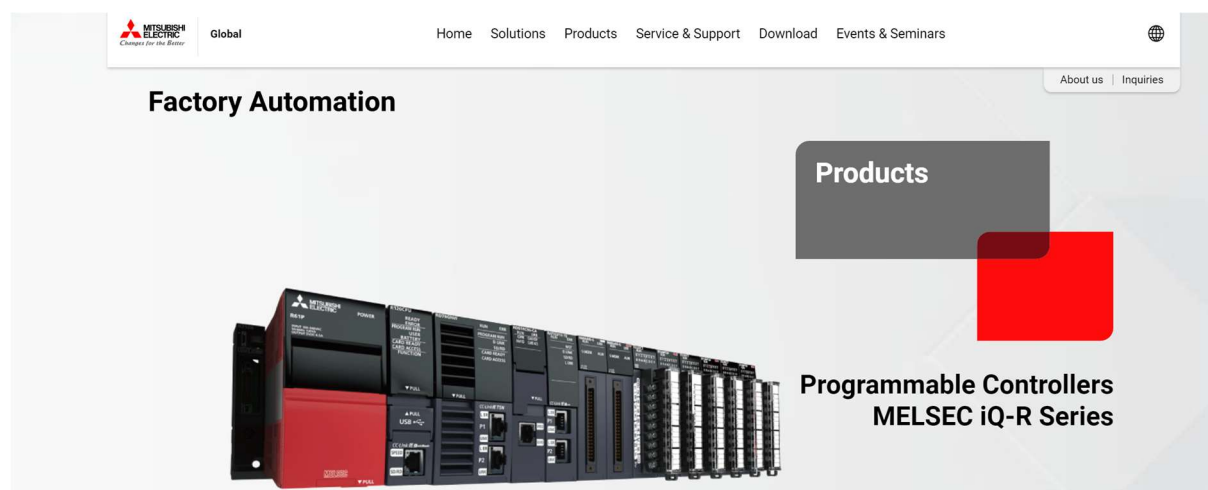


Figure 4 - Modular PLC - Mitsubishi iQ-R [11]

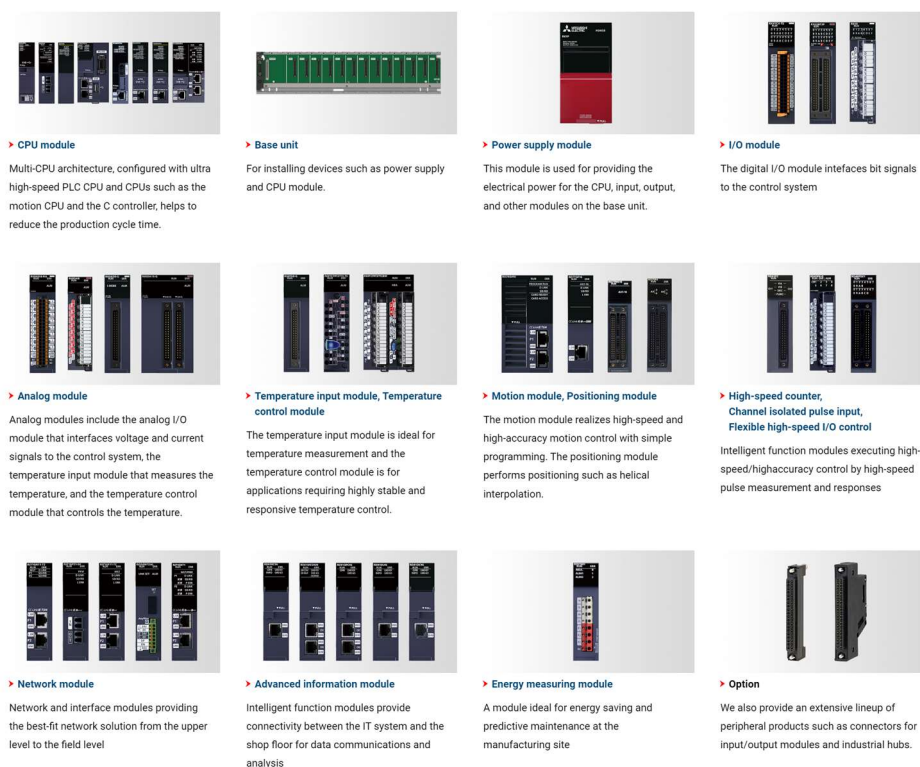


Figure 5 - Mitsubishi modules [11]

PLC comparison

There are many different types of PLC used in industry, as discussed there are compact units and modular units. Each manufacturer offers a different solution to meet the customer's needs. Some of the most popular manufacturers are Siemens, Allen Bradley, Mitsubishi, and Omron [2]. Table 1 below compares four different PLCs from different manufacturers.

Manufacturer	Model	Type	Price (CPU only)	Program capacity	Inputs	Outputs	Supply	Communication ports
Siemens	S7-1200	Expandable	£ 512.40	75kB	14	10	24V	1
Allen-Bradley	Micro870	Expandable	£ 1,393.43	128kB	14	10	24V	3
Mitsubishi	iQ-F	Expandable	£ 925.20	120kB	16	16	100-240V	2
Omron	CP1L	Expandable	£ 724.20	10K steps	18	12	24	1

Table 1 - PLC comparison [12] [14] [14] [8]

The Siemens S7-1200, shown in figure 6, is a favourite for its ease of use. Siemens software, TIA, is simple to grasp and easy to program. The S7-1200 PLC can be used on its own as a standalone unit or expansion modules can be added to cater for industrial requirements [12]. Pros and cons of the S7-1200 can be seen below in table 2.

Siemens S7-1200	
Pros	Cons
Expandable	Only 1 communication port
Can be used without additional modules	Lower program capacity
Competitively priced	No wireless communication
Compact	Lower number of outputs
Low voltage	

Table 2 - Siemens S7-1200 pros and cons [12]

Siemens SIMATIC S7-1200 Series PLC CPU for Use with SIMATIC S7-1200 Series, Digital, Transistor Output, 14 (Digital, 2

RS Stock No.: 862-4471 | Mfr Part No.: 6ES7214-1AG40-0XB0 | Brand: Siemens



S7 In stock - FREE next working day delivery available

98 available from Europe for delivery within 1 working day(s).

1 Units [Add to basket](#)

FREE delivery for orders over £50.00

[Check stock levels](#) [Add to a parts list](#)

Price Each
£427.00
(exc. VAT)

£512.40
(inc. VAT)

Units	Per unit
1 +	£427.00

[View this category](#)

Figure 6 - Siemens S7-1200 PLC [12]

The Allen Bradley micro 870 can be seen in figure 7. Like the S7-1200, the micro 870 series can be used on its own, or as a modular unit depending on the industries requirements. Although Allen-Bradley is commonly used throughout industry, engineers often complain about the short fall of its software [13] [14]. The pros and cons of the micro870 can be seen below in table 3.

Allen Bradley Micro870 Series PLC CPU, 24V Source Output, 14-Input, Voltage Input

RS Stock No.: 183-3562 | Mfr. Part No.: 2080-LC70-24QBB | Brand: [Allen Bradley](#)



Available to back order for despatch
12/06/2024

1 Units [Back order](#)

FREE delivery for orders over £50.00

[Add to a parts list](#)

Price Each
£1,161.19
(exc. VAT)

£1,393.43
(inc. VAT)

Units	Per unit
1 +	£1,161.19

[View this category](#)

Figure 7 - Allen Bradley Micro 870 [14]

Allen-bradley Micro870	
Pros	Cons
Expandable	More expensive
Can be used without addiotional modules	Lower number of outputs
Low voltage	
Multiple communication types	
Multiple language option	

Table 3 - Allen-Bradley Micro870 Pros and cons [14]

The Mitsubishi IQ-F is an older PLC shown in figure 8. The IQ-F offers high processing speeds, designed to keep up with fast manufacturing processes. Mitsubishi software is efficient and easy to grasp, making programming undemanding for engineers. The pros and cons of the IQ-F can be seen below in table 4 [15].

Mitsubishi FX5U Series PLC CPU for Use with MELSEC IQ-F Series IQ Platform-Compatible PLC, Relay, Transistor Output,

RS Stock No.: 875-5672 | Mfr. Part No.: FX5U-32MR/ES | Brand: [Mitsubishi](#)



3 In stock - FREE next working day delivery available

1 Units [Add to basket](#)

[FREE delivery for orders over £50.00](#)

[Check stock levels](#) [Add to a parts list](#)

Price Each
£771.00
 (exc. VAT)

£925.20
 (inc. VAT)

Units	Per unit
1 +	£771.00

[View this category](#)

Figure 8 - Mitsubishi IQ-F PLC [15]

Mitsubishi iQ-F	
Pros	Cons
Expandable	No wireless communication
Built in SD card	Only 2 communication ports
Built in ethernet	Higher voltage
Integrated real time clock	
Large number I/O	

Table 4 - Pros and cons of Mitsubishi iQ-F [15]

The Omron CP1L, figure 9, is a compact PLC. Without expansion It has the most input capabilities of the four PLCs compared. Due to its compact nature, its shortfall is a lack of communication options as it only offers USB communication. The pros and cons of the CP1L can be seen in table 5 [8].

The screenshot shows the RS Components website interface. At the top, there's a navigation bar with the RS logo, links for 'Parcel Tracking', 'Login / Sign up', and a shopping cart icon showing '£0.00'. Below this is a search bar with the text 'Search by keyword, manufacturer part no. or RS part no.' and a 'Menu' button. The main content area features the product title 'Omron CP1L Series PLC CPU for Use with SYSMAC CP1L Series, Relay Output, 18 (DC)-Input, DC Input' with the Omron logo. Below the title, it lists 'RS Stock No.: 401-573', 'Mfr. Part No.: CP1L-M30DR-D', and 'Brand: Omron'. There's a 'Share' button. The product image shows a black PLC unit with various ports and a terminal block. To the right of the image, there's a green banner stating '5 In stock - FREE next working day delivery available'. Below this is a quantity selector set to '1' with an 'Add to basket' button. A purple banner indicates 'FREE delivery for orders over £50.00'. There are also links for 'Check stock levels' and 'Add to a parts list'. The price section shows 'Price Each' as '£603.50 (exc. VAT)' and '£724.20 (inc. VAT)'. A table at the bottom shows 'Units' as '1+' and 'Per unit' as '£603.50'. A blue button at the bottom says 'View this category'.

Figure 9 - Omron CP1L PLC [8]

Omron CP1L	
Pros	Cons
Larger number of inputs	single communication port
Competative price point	Lower program capacity
AC or DC supply	
Battery back up	

Table 5 - Pros and cons of Omron CP1L PLC [8]

Factory design

In order to design a PLC program capable of sorting four boxes, a factory was first designed. Using factory IO simulation software, a conveyor system was developed utilising diffuse sensors, pushers and weighing scales. Four different boxes were emitted onto the conveyor. A palletising box, labelled as small, a large box and two medium sized boxes which were labelled medium light and medium heavy. An overview of the designed factory can be seen below in figure 10.

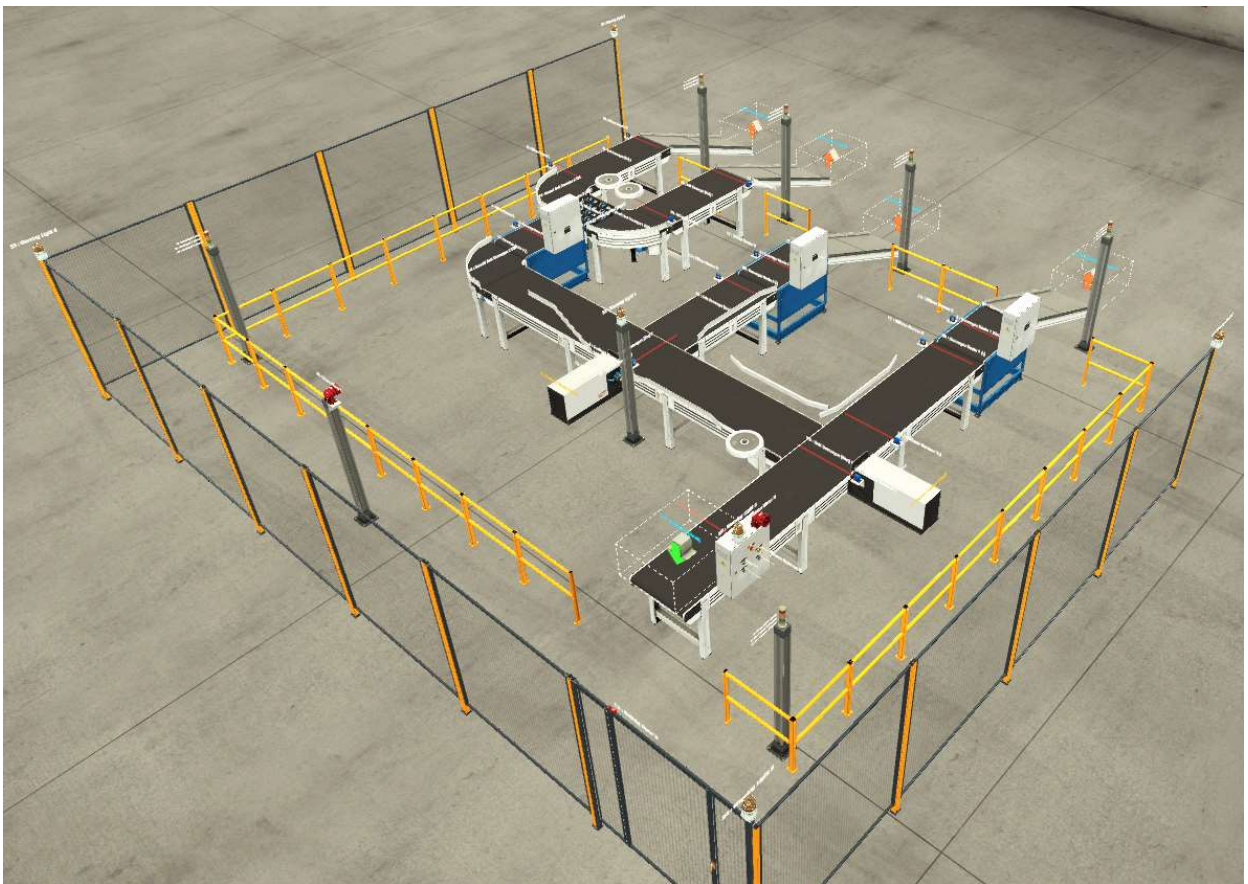


Figure 10 - Factory overview

The green arrow emits the different sized boxes, following that a diffuse sensor is activated by a box. This sensor is used to count the total boxes emitted. Further down the conveyor a second diffuse sensor is located. The sensor is set at a strategic height, all boxes apart from the smallest, palletising box, activates the sensor. When the sensor is activated, after a time delay the first pusher is activated. The pusher sends the larger boxes onto a separate conveyor, the smallest box continues down the original conveyor, this can be seen in figure 11.



Figure 11 - Box separation

Continuing down the original conveyor the small box activates another diffuse sensor, the sensor counts small boxes. At the end of the conveyor another diffuse sensor counts down the small boxes. Discrepancies in this count are used to activate an alarm. The small box is then weighed on a conveyor scale, figure 12. If the box is over a set weight an overweight alarm is triggered.

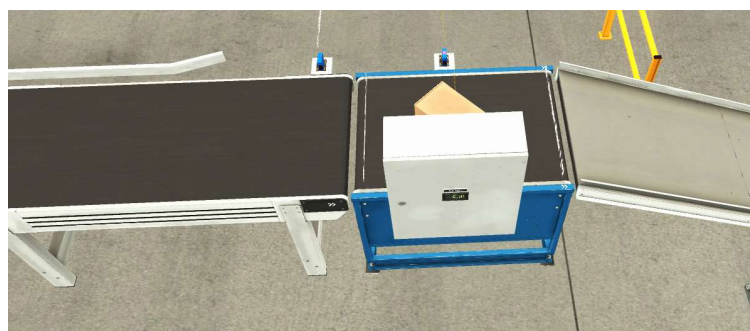


Figure 12 - Small box on scale

Boxes separated onto the secondary conveyor pass through another diffuse sensor. This sensor is positioned to allow medium boxes to pass underneath it. Large boxes activate the sensor which after a delay, activates the second pusher. This can be seen in figure 13 below. The pusher counts the large boxes, the next diffuse sensor counts them down. If a box is missing a warning or alarm is activated. The box is then weighed on a scale, if the box is over a set weight an alarm is flagged.

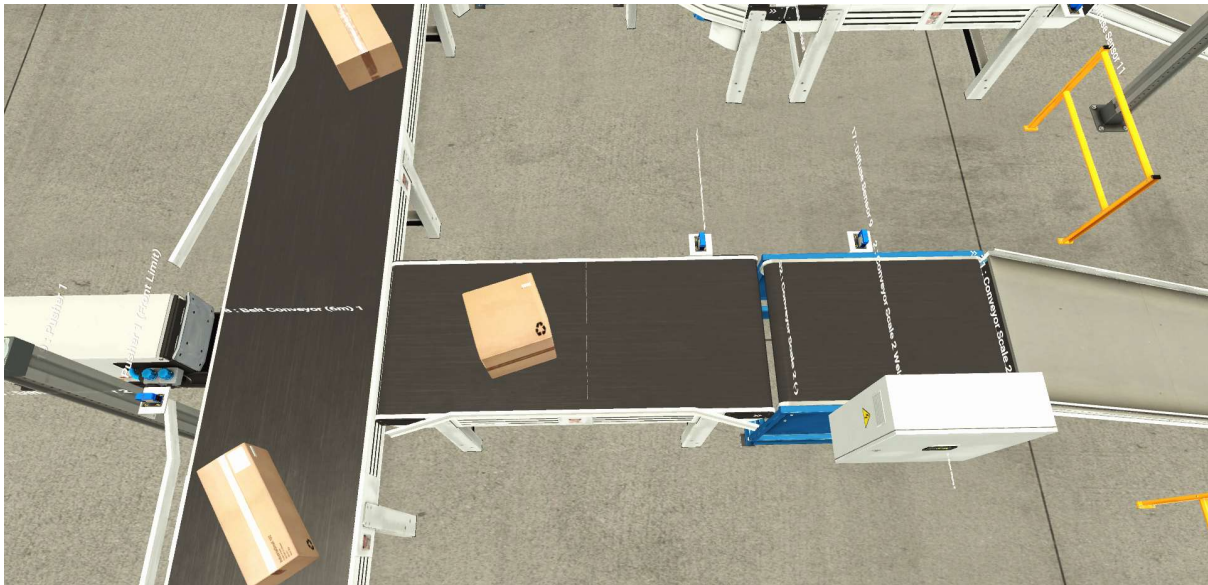


Figure 13 -Large box separated

The remaining boxes are the medium boxes. As they are the same height they can only be sorted by weight. The boxes are weighed on a conveyor scale, boxes over a set weight are sent left by a pop-up sorter. Boxes under a set weight are sent right by the sorter. After sorting, a diffuse sensor on each corner conveyor counts the boxes. Another sensor at the end of the conveyor counts down the boxes. A discrepancy on either triggers an alarm or warning. Boxes found to be over a set weight triggers an overweight alarm. This is shown below in figure 14.

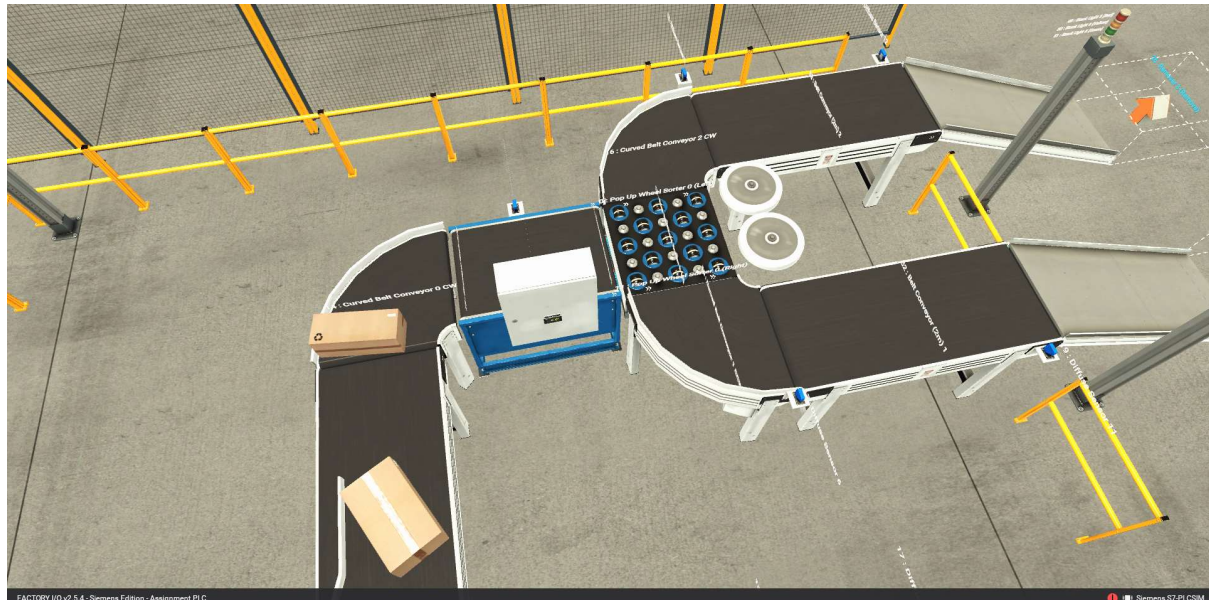


Figure 14 - Medium box sorter

For the warning system, each box chute has an individual stack light. When an alarm is triggered, the corresponding red stack light is activating. If a warning is triggered the yellow stack light will illuminate. When the system is running normally each stack light is green. These stack lights are shown below in figure 15.



Figure 15 - Chute stack lights

In addition to the warning indicator stack lights on each chute, there are also two global stack lights located at the back corners of the factory as seen in figure 16. These global stack lights are green when the system is not running, indicating site safety. When any alarm is triggered the global stack lights illuminate red. When any warning is triggered, they indicate yellow. Figure 16 also shows the alarm siren in the middle. This siren is used to alert the operator of an alarm.

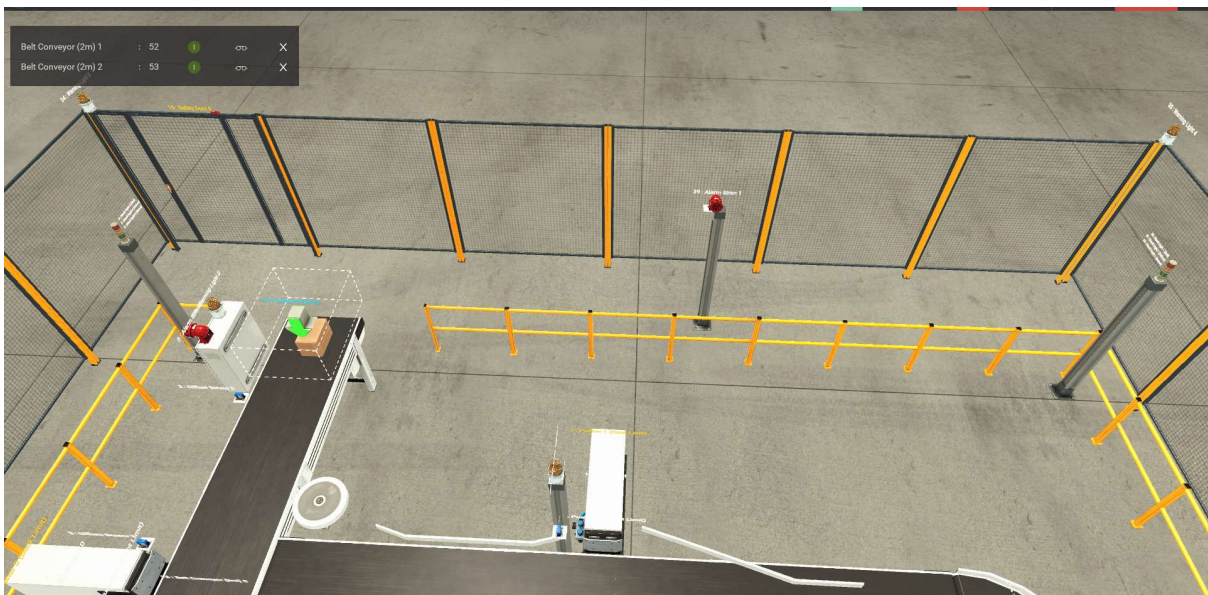


Figure 16 - Global stack lights

The main control panel seen in figure17 includes a start button, with a green indicator light and a stop button, with a red indicator. It includes an emergency stop which is also linked to a door interlock as seen in figure 18. On the control panel is an alarm siren which sounds to alert works of the conveyors starting. This also activates the warning lights which are located throughout the factory.



Figure 17 - Control panel

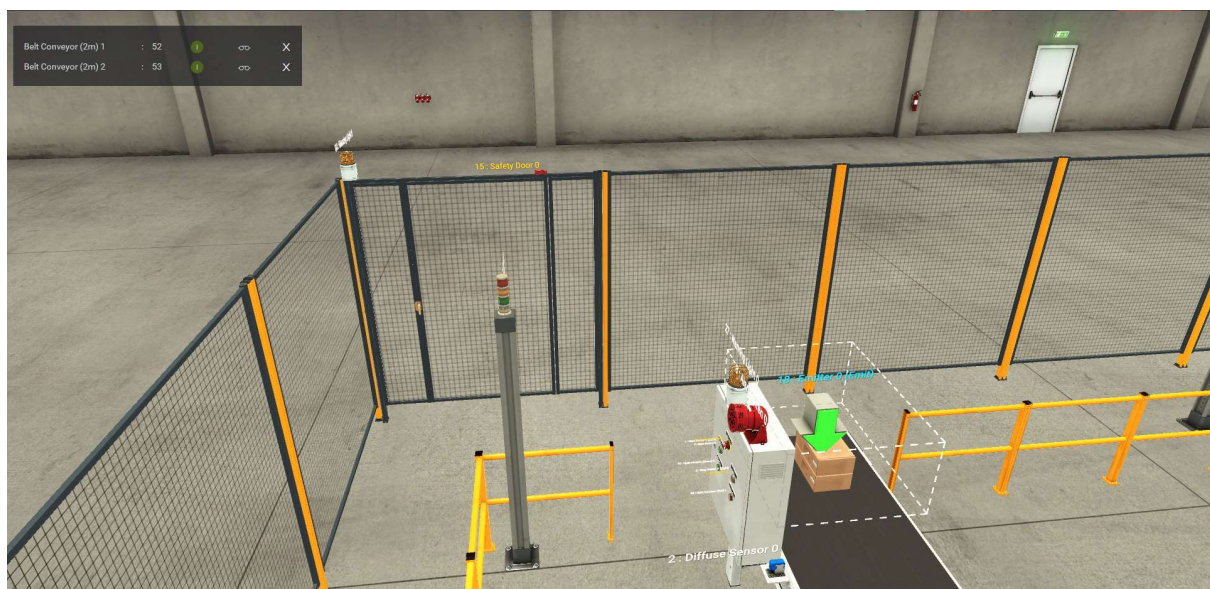


Figure 18 - Door interlock

Each scale conveyor has a control panel which displays the weight of the box as it is weighed, see figure 19.



Figure 19 - Scale conveyor digital displays

Factory flowchart

The basic operation of the factory can be visualised in the flowchart below, figure 20. Inputs and outputs are labelled with the tags assigned in the program.

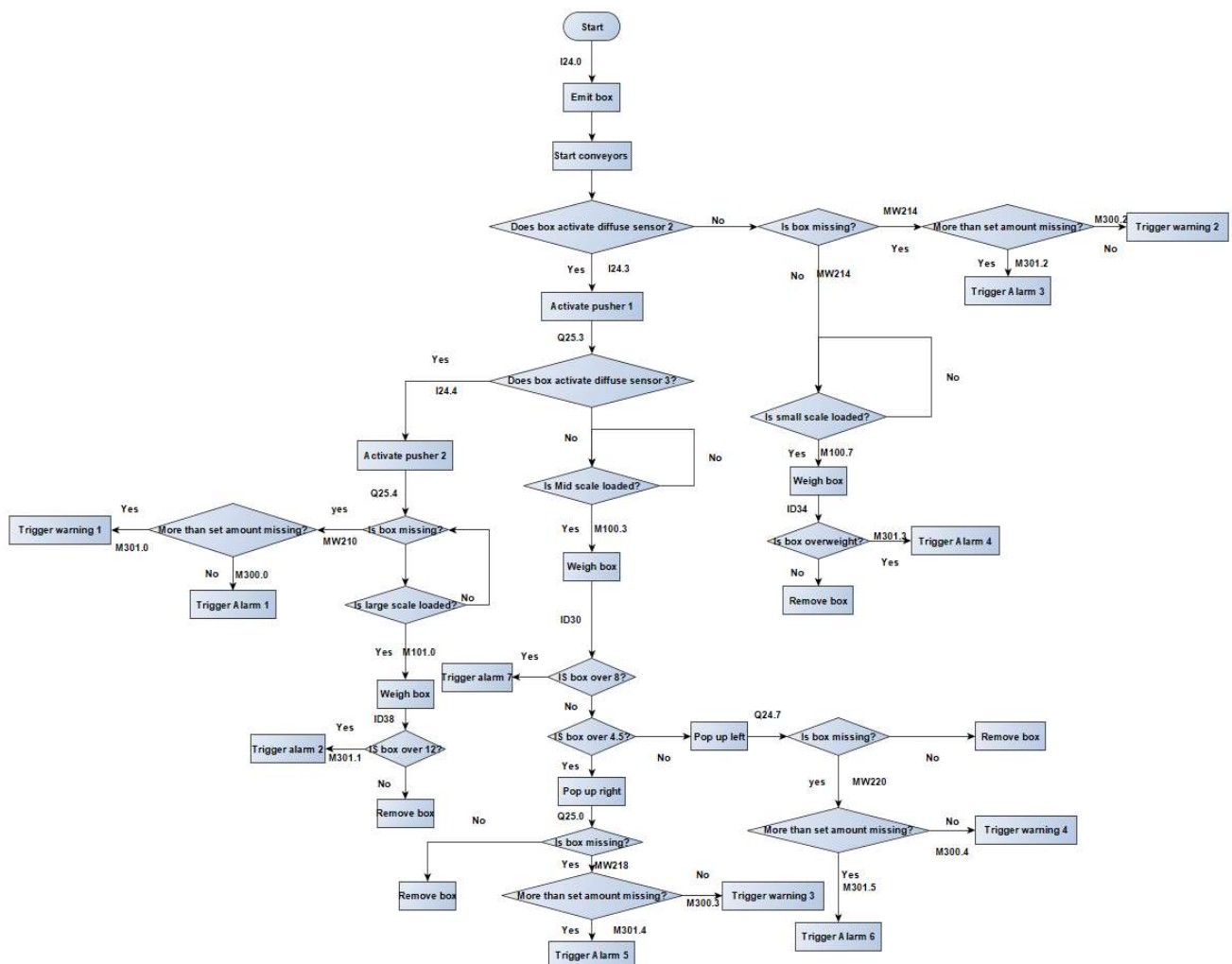


Figure 20 - Program flowchart

HMI screens

To allow the operator to interact with the factory HMI screens were designed. Human machine interface (HMI) screens offer the engineer a graphical means of interacting with the system. A HMI screen must inform the engineer how the system is functioning and allow adjustable parameters to be set [9].

A root screen as shown in figure 21, allows the operator to start and stop the system, an emergency stop button is also included. The counters or entire system can be restarted using the corresponding buttons. Indicator lights are programmed to show whether the system is stopped or running, whether the safety door is open and whether the emergency stop is depressed. The root screen also displays the total box count, and each individual box size count. The weight of box on each scale is also displayed. The global stack lights are replicated on the root screen. The individual stack lights are replicated on a second HMI screen as seen in figure 22. Each screen allows the user to toggle to the desired screen.

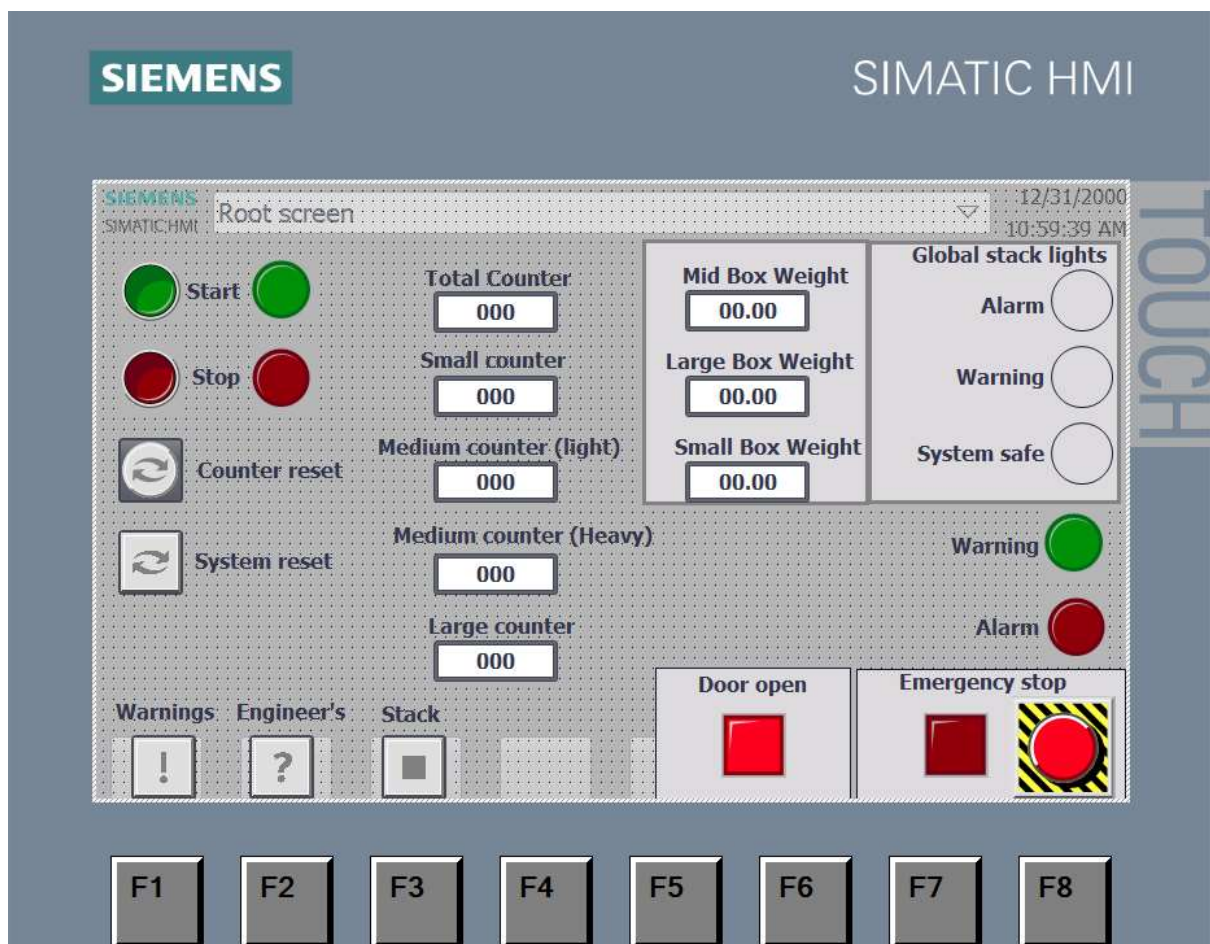


Figure 21 - Root screen

When a stack light in the factory is activated, the corresponding stack light on the HMI screen is also activated. This is linked to the warnings and alarms strategy.

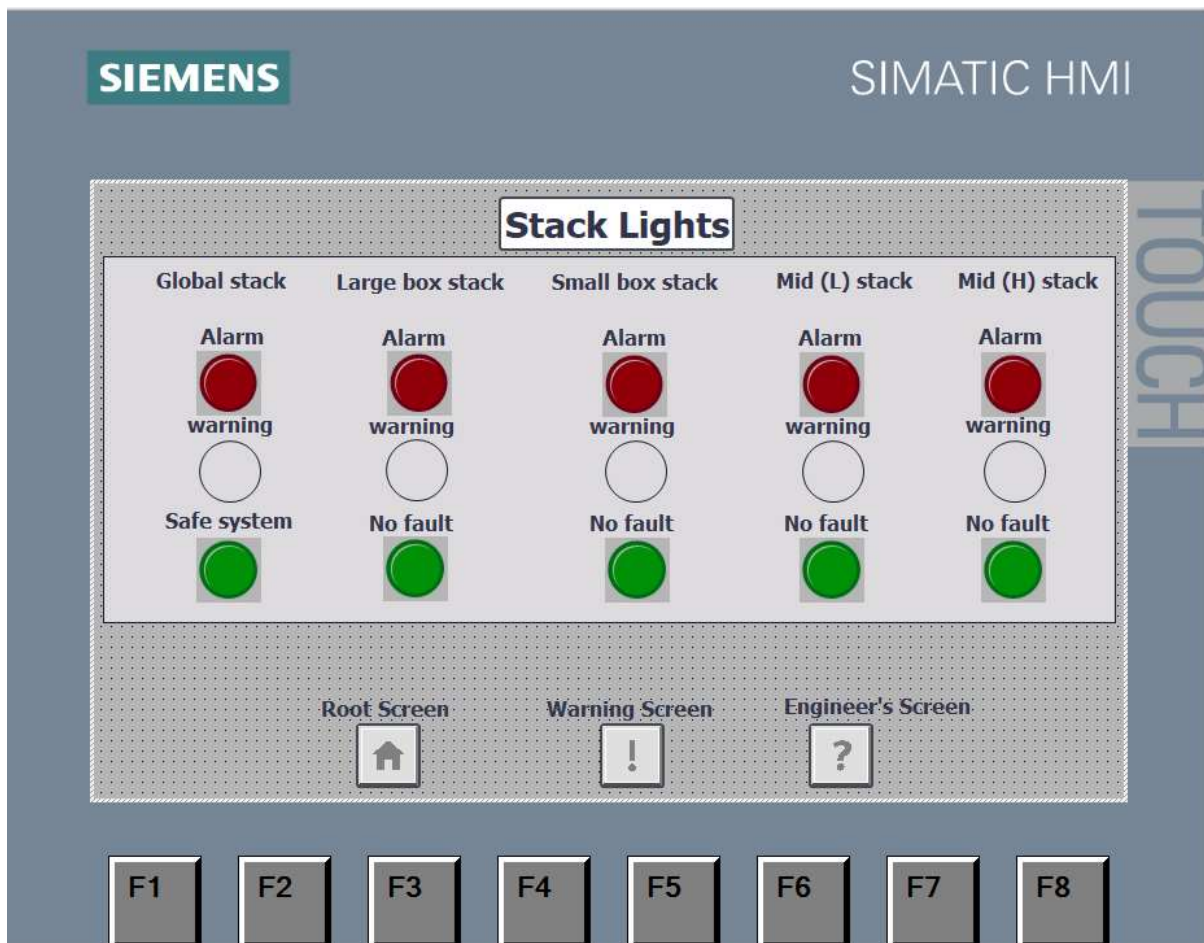


Figure 22 - Stack lights replication

To allow the engineer to input parameters into the program, an engineering screen was designed as shown in figure 23. For the warning and alarm system, the engineer must program how many boxes they require to trigger each output.

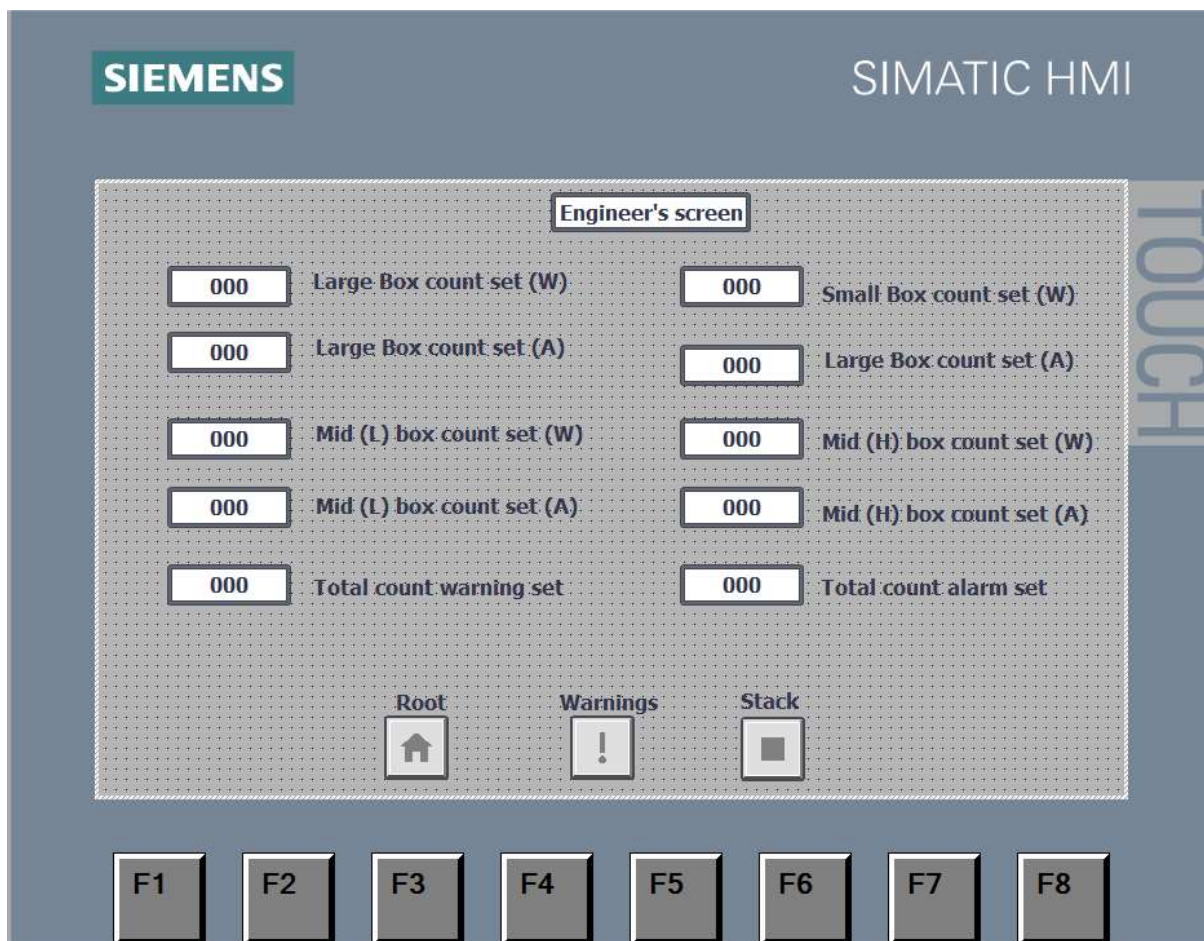


Figure 23 - Engineer's screen

To aid the engineer in visualising which warning, or alarm was triggered, the warning screen was designed. When a warning or alarm is triggered, the corresponding light is activated. This is shown in figure 24.

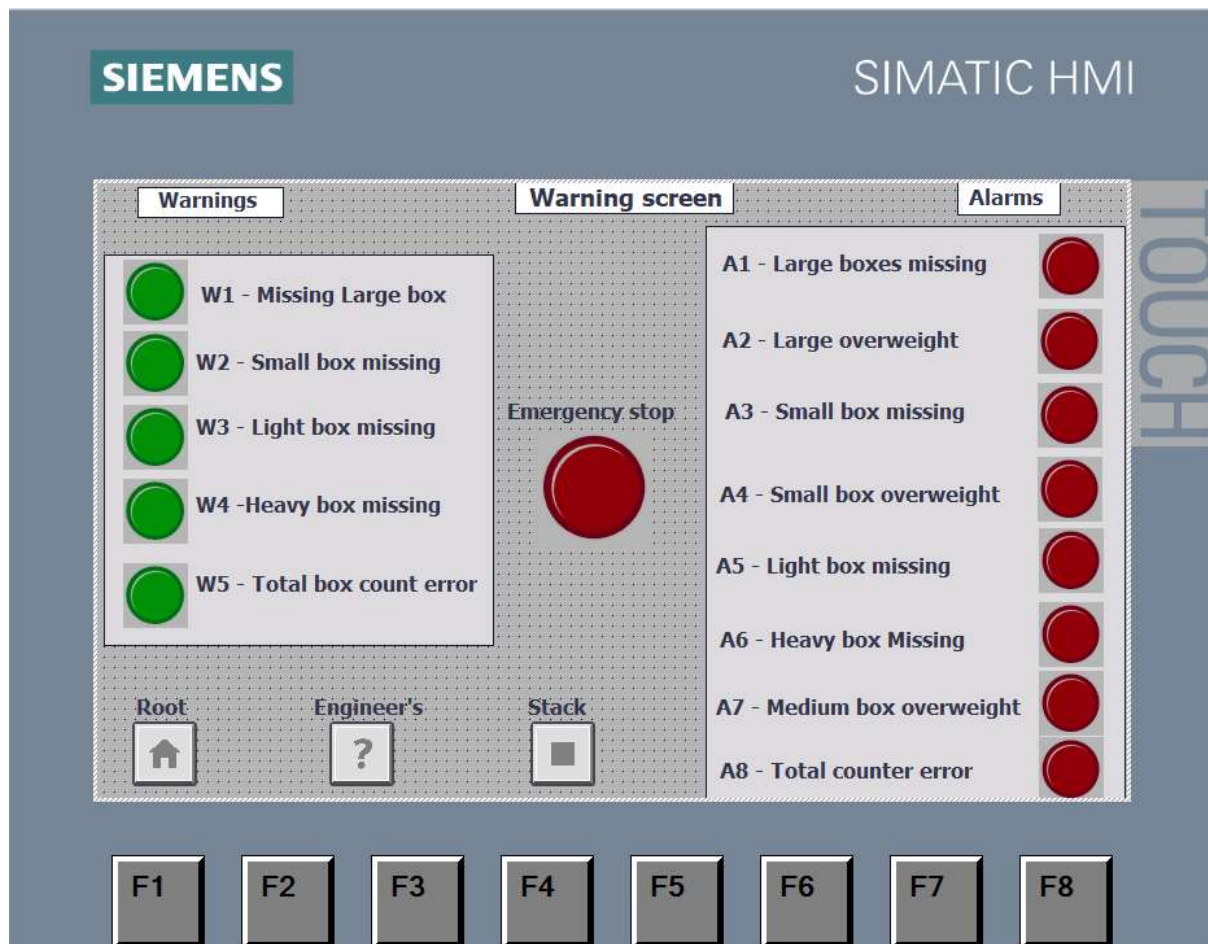


Figure 24 - Warning screen

Warnings and alarm strategy

A warning and alarm system was programmed into the system. Each box size was given its own warning system which was linked to the corresponding stack light. Parameters for each warning are input by the engineer. Warning 1 and alarm 1 are used to determine if a large box has fallen off the conveyor. If a box falls off the conveyor between the pusher and the sensor highlighted in figure 25, the warning or alarm is triggered. Alarm 2 is triggered if the large box is over a set weight on the scale conveyor.

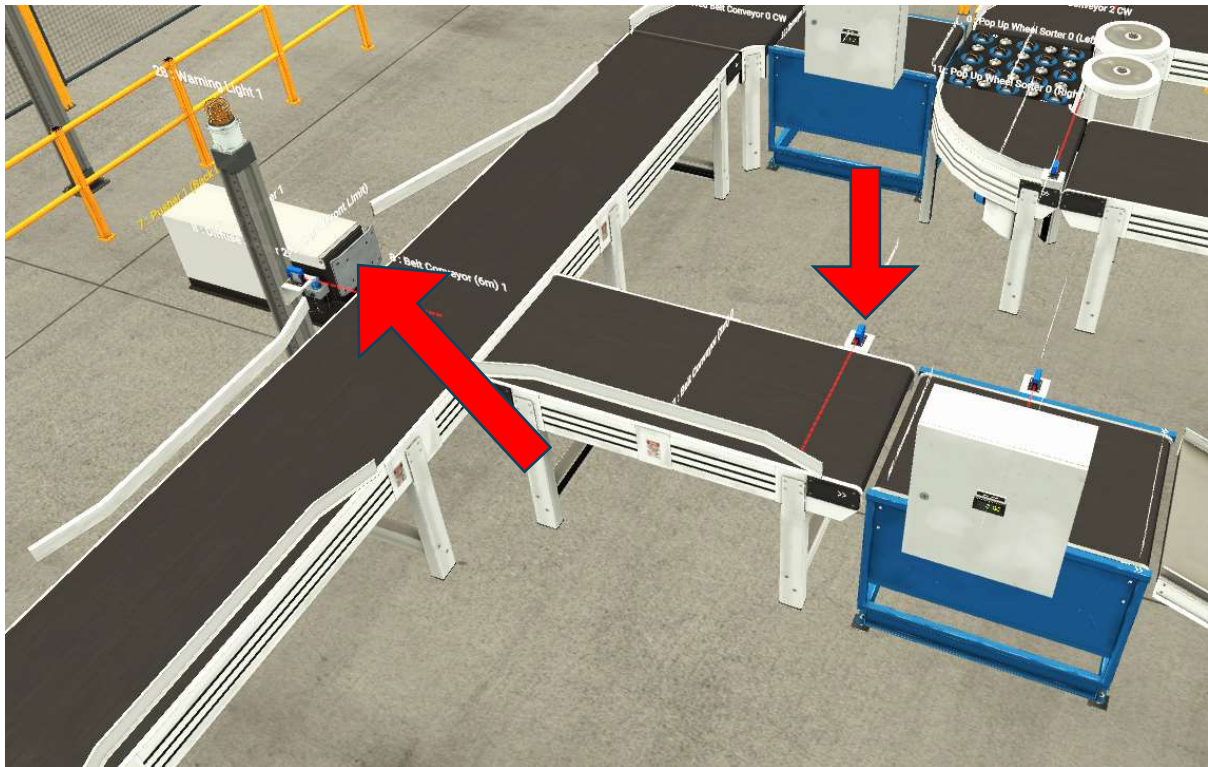


Figure 25 – Alarm 1 + warning 1

Warning 2 and alarm 3 are triggered by a missing small box. The boxes are counted by the sensors highlighted in figure 26. If a box falls off between these sensors the warning or alarm is triggered. Alarm 4 is triggered if a small box is over a set weight on the scale conveyor.

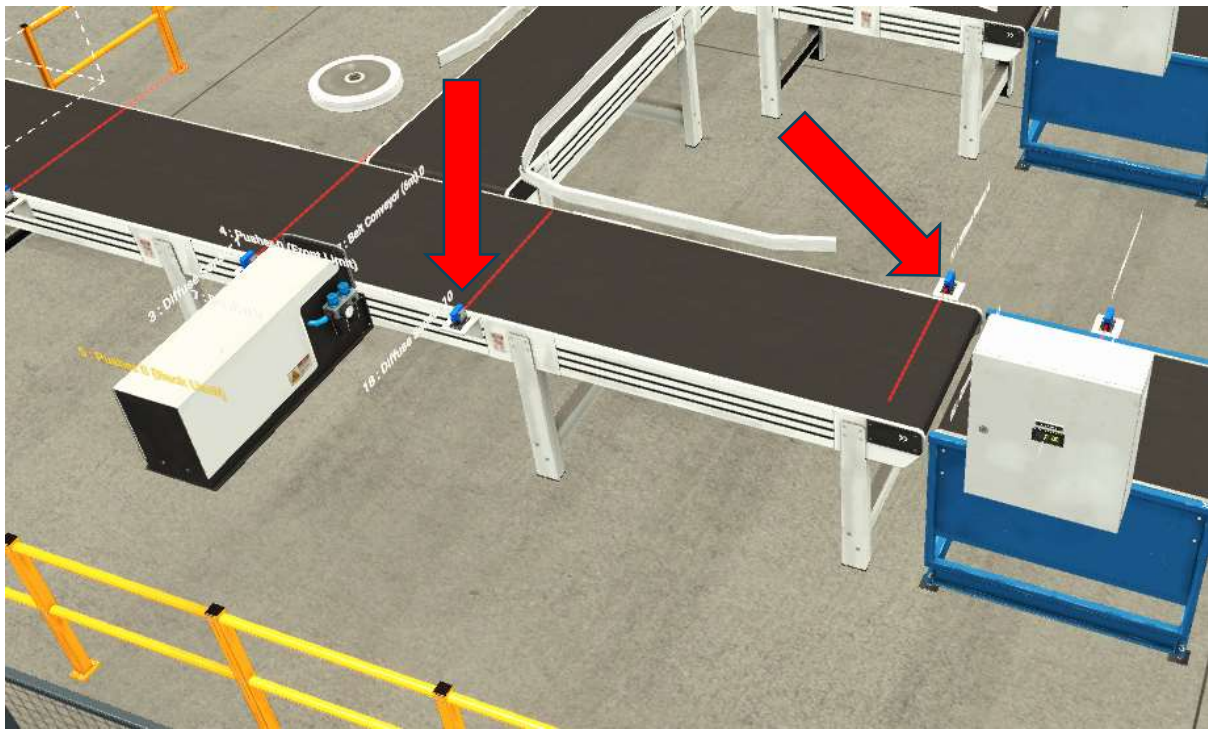
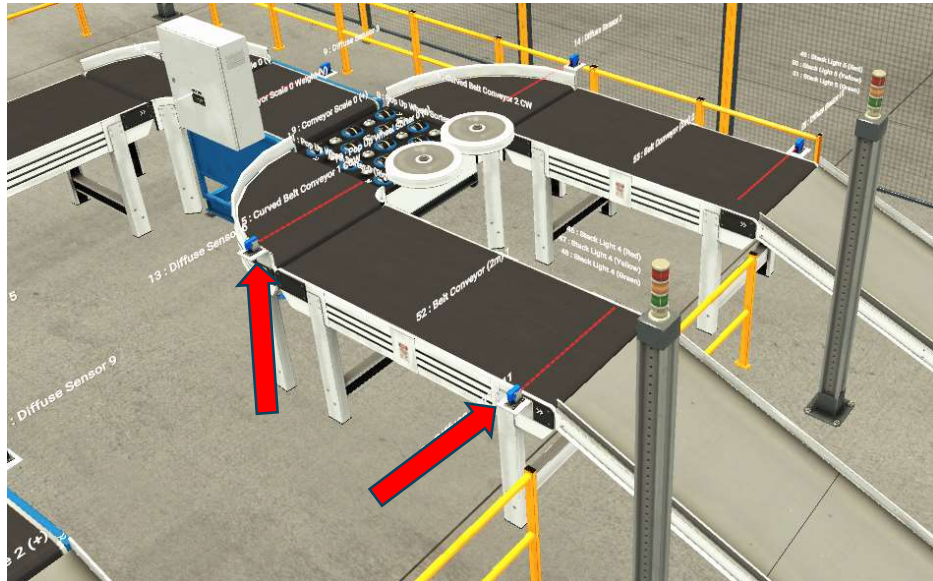


Figure 26 - Warning 2 + Alarm 3 sensors

Tomos Hall (2219188)



Warning 5 and alarm 8 are triggered by the total box count, set by the engineer. The first diffuse sensor counts up and each diffuse sensor labelled in figure 29 counts down.

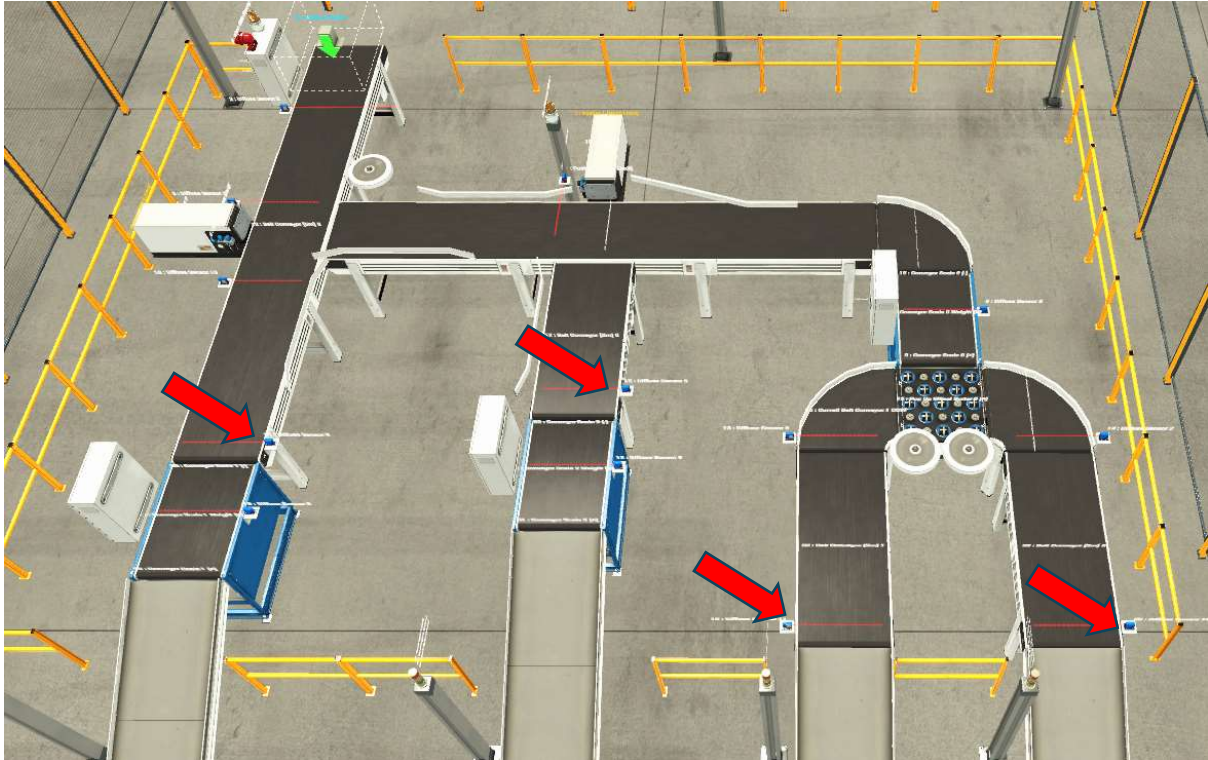


Figure 29 - Warning 5 and alarm 8, total box error

Ladder Program

In order to operate the factory as designed, the PLC must be programmed. A ladder logic program was written using Siemens TIA software. The full program can be found in appendix 1. Figure 30 shows the first two rungs of the ladder. The first rung controls the whole factory operation. A normally open (NO) contact is linked to the start button on the factory IO control panel, when it is pushed the output “factory operate” is activated as well as the green indicator light. When the output is triggered, it is “latched” on, meaning the factory stays running until the signal is interrupted. Both the “stop button” and the “Emergency stop” are capable of interrupting. The second rung links the emergency stop button from the control panel and the emergency stop from the HMI screen to a combined output. The Door interlock is also linked to this output. When either the door, HMI E-stop or control panel E-stop is activated, the emergency stops normally closed (NC) contact in the first rung will stop the factory.

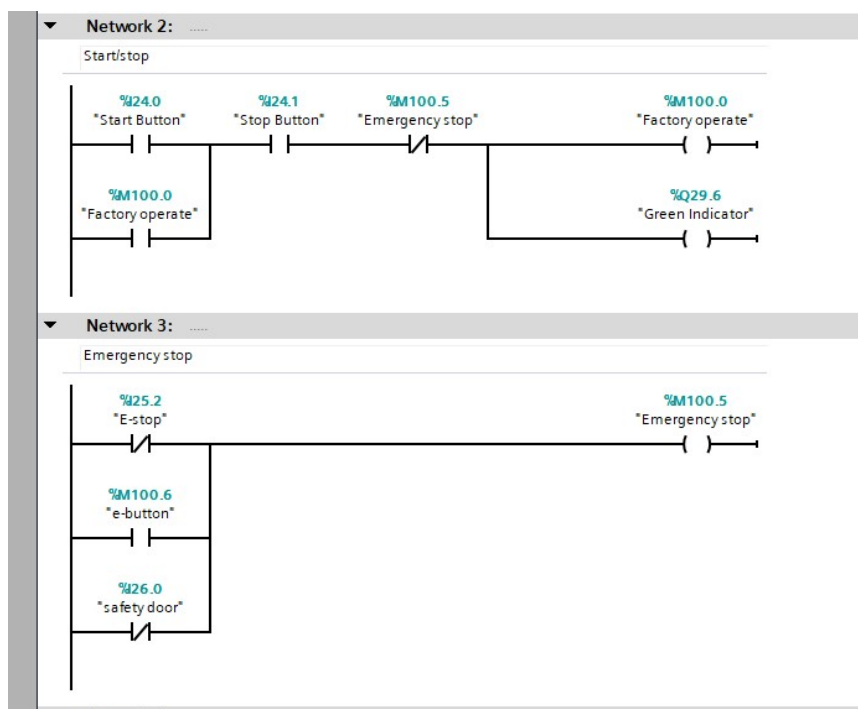


Figure 30 - Rungs 1 and 2 of the ladder program.

The next network of the ladder continues the factory operate function. A TP timer generates a two second pulse which sounds an alarm to announce the movement of the conveyors. Siren lights around the factory are also triggered by the pulse to warn workers. This can be seen in figure 31.

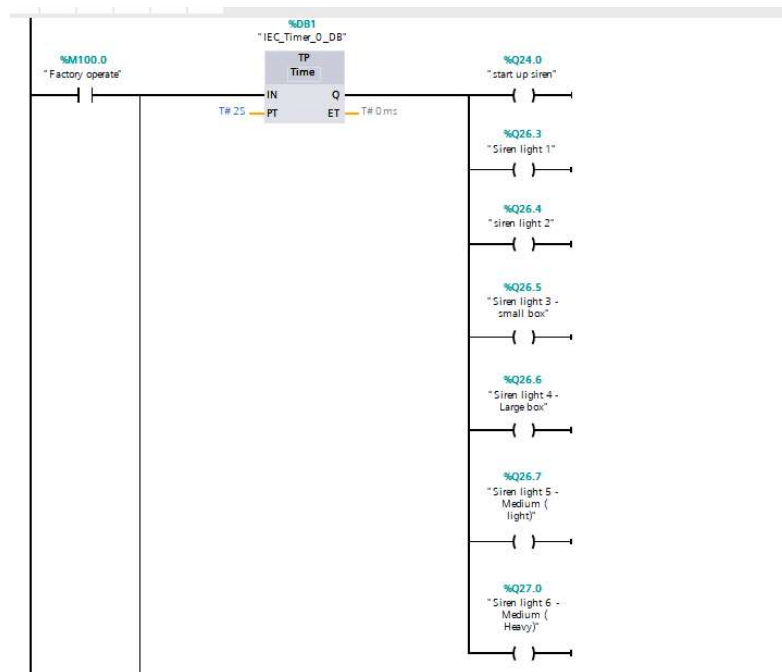


Figure 31 - Rung three

Figures 32-34 show the continued factory operate functions. TON timers are used to input a delay for each conveyor. An increased delay ensures each conveyor starts up at a different time, minimising start up current. Scale conveyors include a normally closed input, this flag is used to stop the conveyor while it is loaded. This prevents additional boxes being added to the scale.

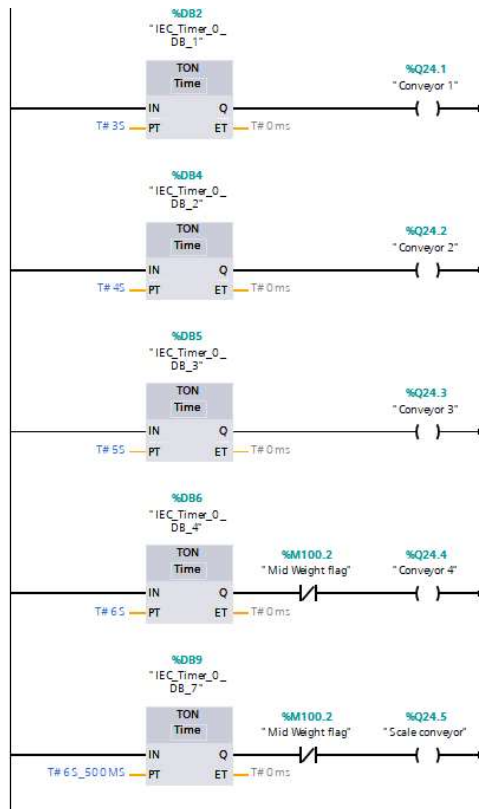


Figure 34 - Factory operate, conveyor functions

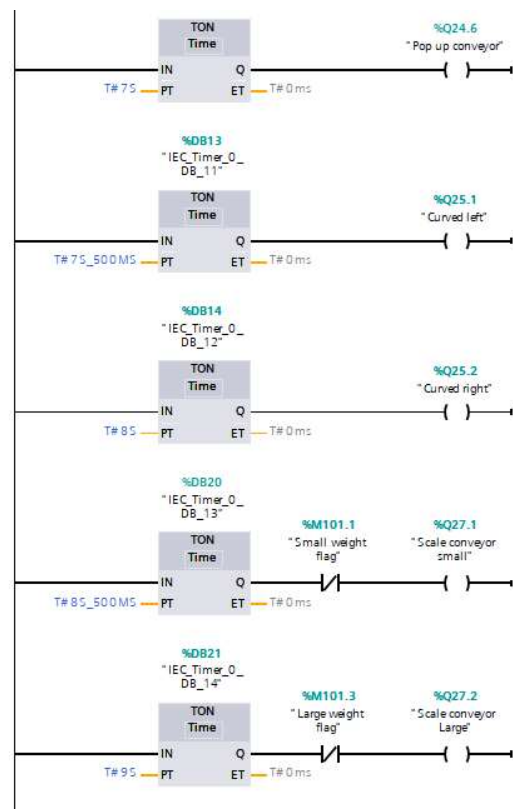


Figure 33 - Factory operate, conveyor functions continued

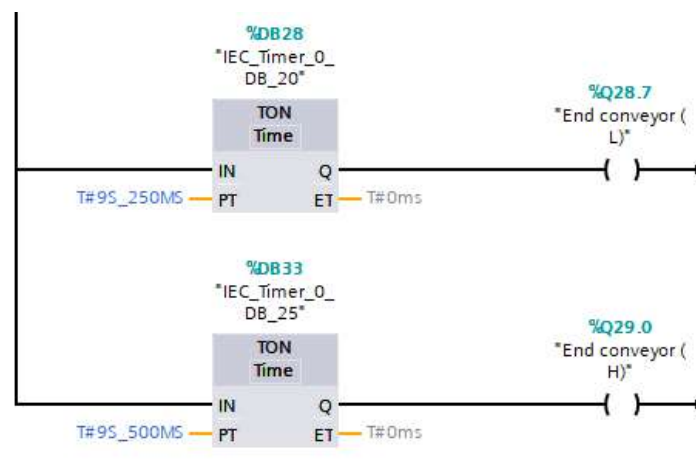


Figure 32 - Factory operate, conveyor functions continued

The next network is used to control the global stack lights, these are the stack lights in the factory that aren't specifically linked to a box size. Both global green stack lights are saved to a single memory core to simplify use later in the program. The global green stack light is programmed to be on when the factory is not running, this shows workers it is safe to enter the conveyor area. The red and yellow global stack lights are linked in the same way as the green. This is shown in figure 35.

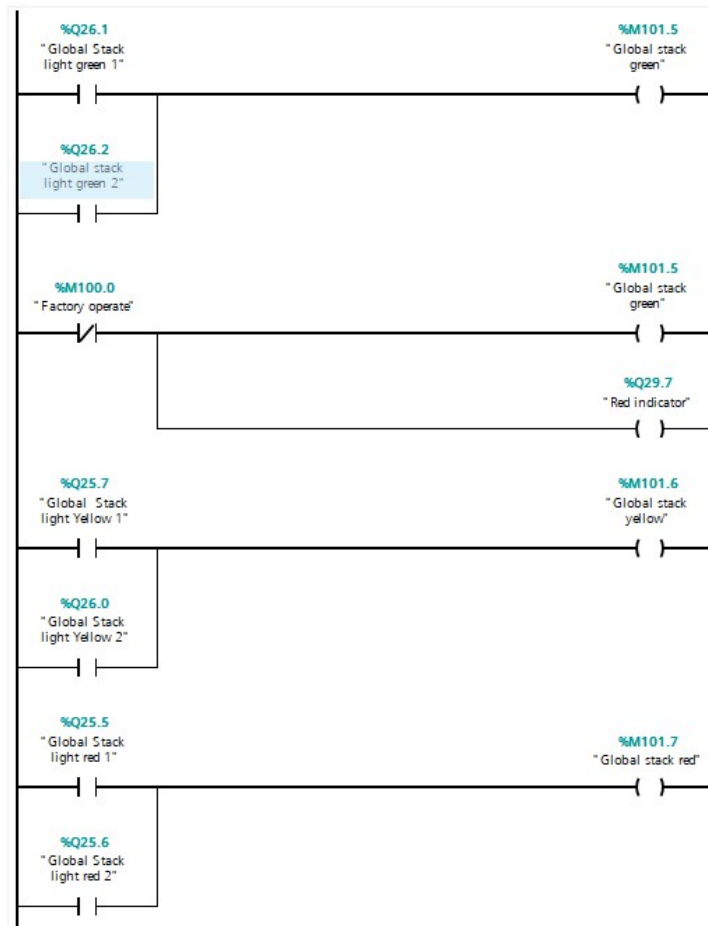


Figure 35 - Global stack set up

The next network controls the system reset button programmed to the HMI screen. This is used to reset all scales and counters in the system. This network is shown in figure 36.

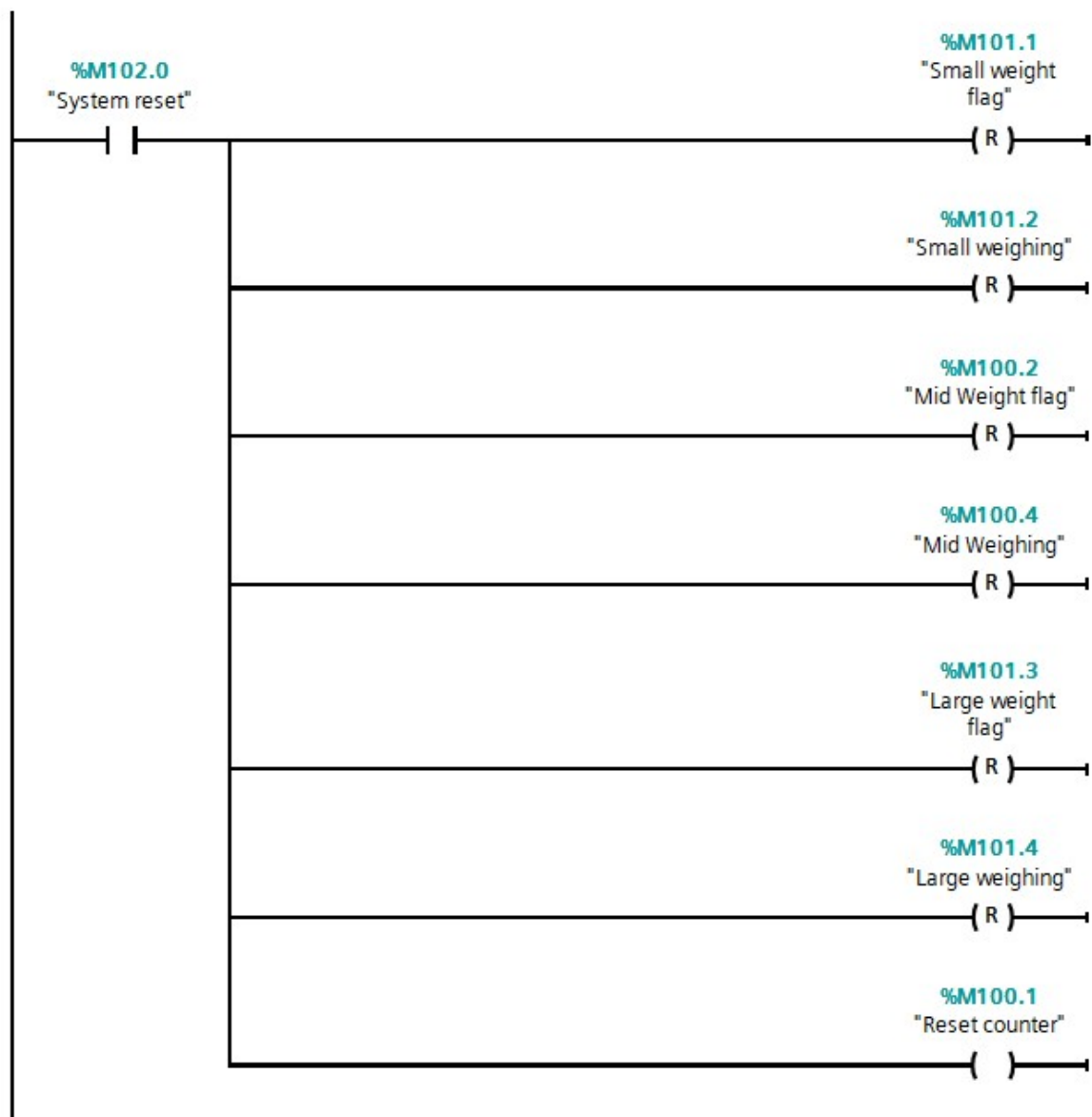


Figure 36 - System reset

The next network, figure 37, controls the total count of the system. It is linked to the first diffuse sensor in the factory and counts every box emitted. The value is displayed on the HMI screen.

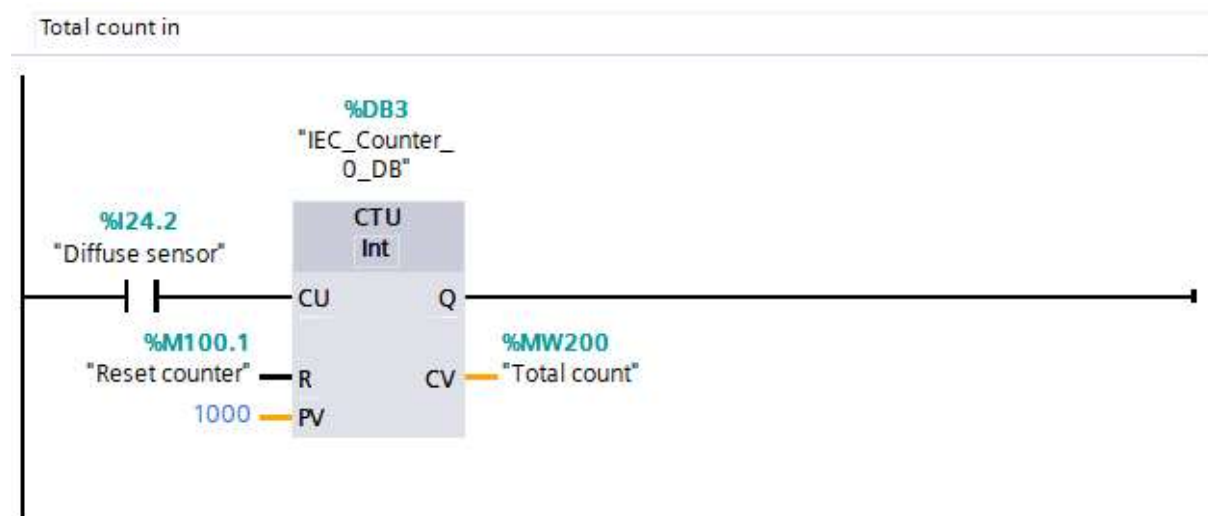


Figure 37 - Total counter

The next network controls the first pusher, pushing large and medium boxes onto the next conveyor. A time delay of 300mS is used to push the box at the required time. See figure 38.

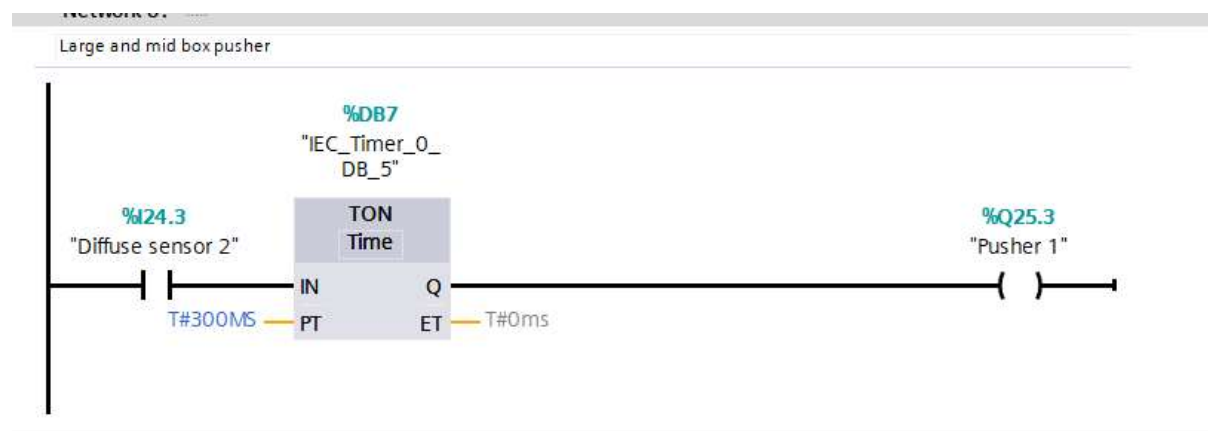


Figure 38 - Pusher 1

The same operation is utilised again to push the large boxes onto the next conveyor, shown in figure 39.



Figure 39 - Large box pusher

The next network controls all three scale conveyors. Each scale uses a diffuse sensor to sense when a box is loaded, this is saved to a memory function, figure 40.



Figure 40 - Scale conveyor sensors (Box loaded)

The next network, figure 41, controls the scale conveyors for each box. Using the “box loaded” memory from the previous network, a flag is set for each box size. A time delay is added to allow the box to be weighed. Another flag is set whilst the box is weighing, after another time delay, both flags are reset allowing the conveyors to restart.

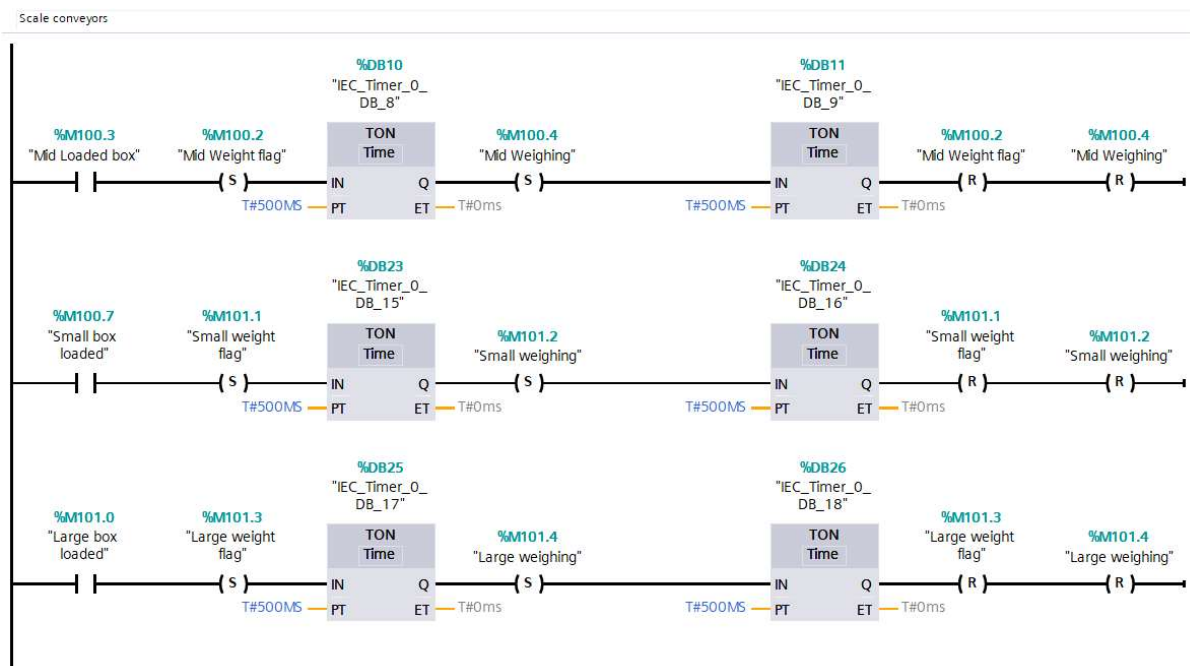


Figure 41 - Scale conveyor flags

The next network, 42, uses a move function to take the data from the scales and display it on the digital display for each box size’s panel.

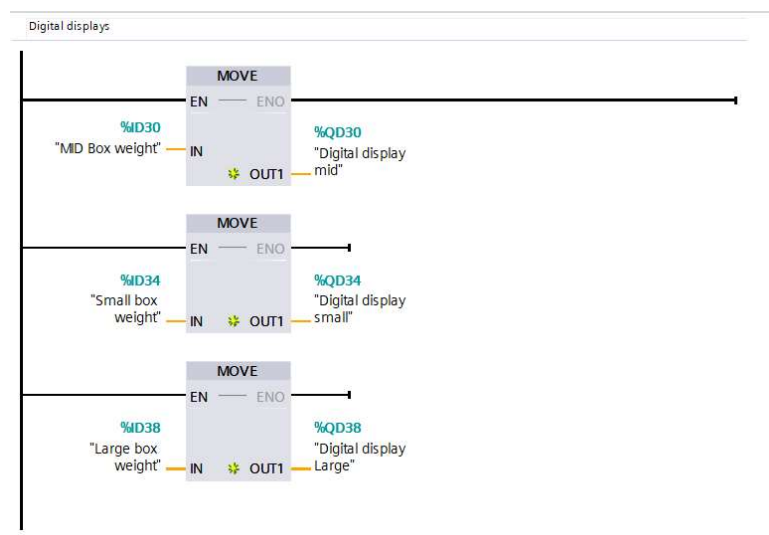


Figure 42 - Scale digital display

The next network, figure 43, controls the pop-up sorter. Using the “weighing” flag from figure 41 to determine when the sorter is required, two comparator functions are used. Each comparator compares the weight of the box with a set value. If the box is greater than or equal to 4.5, the sorter pops up left, resetting any right bias. If the value is less than or equal to 4.4, the sorter sets right, resetting left. The box weights are set as “real” inputs as they represent fractional

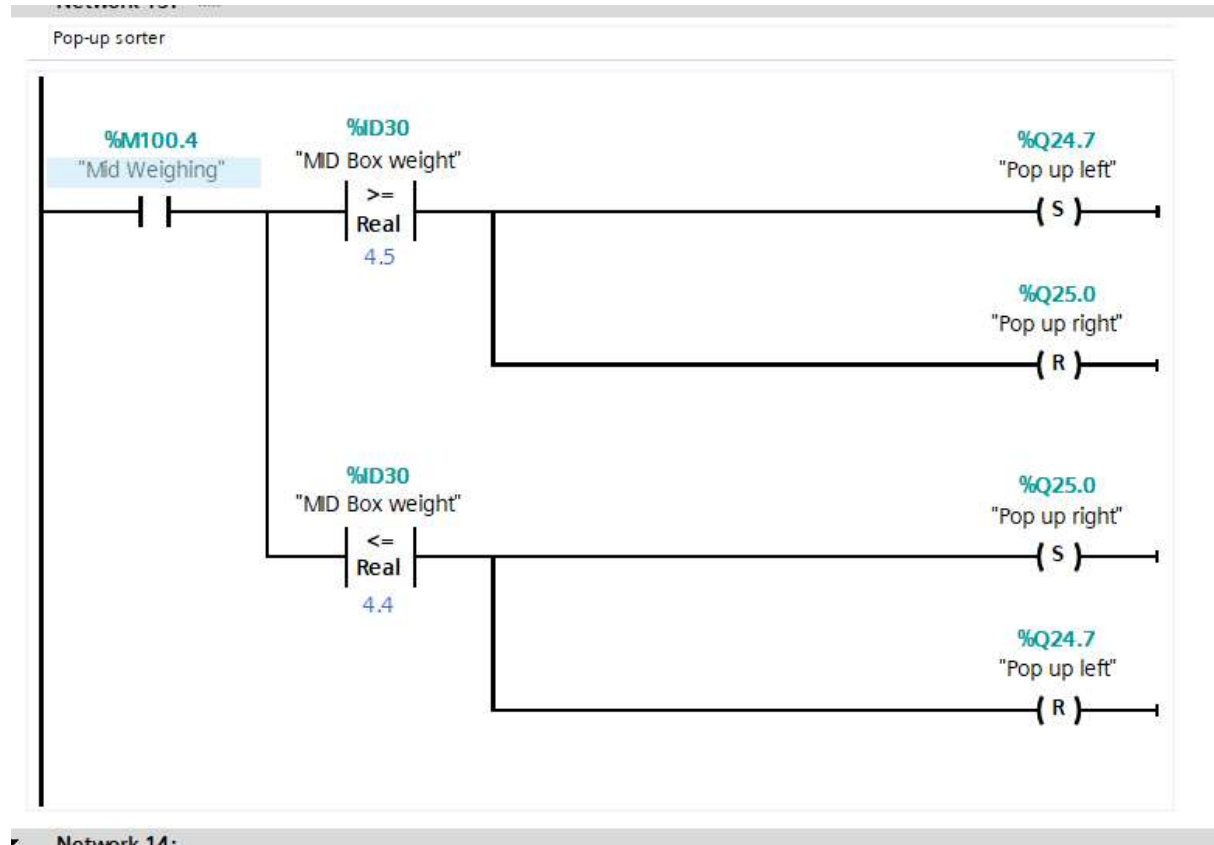


Figure 43 - Pop-up sorter

decimal numbers.

The next four networks control the counters for each box size. A counter is a computing function provided by TIA. They operate using diffuse sensors near the end of each conveyor, providing an accurate count of each box size that leaves the system. This can be seen in figure 44 and 45. Each counter is connected to the counter reset button on the HMI screen.

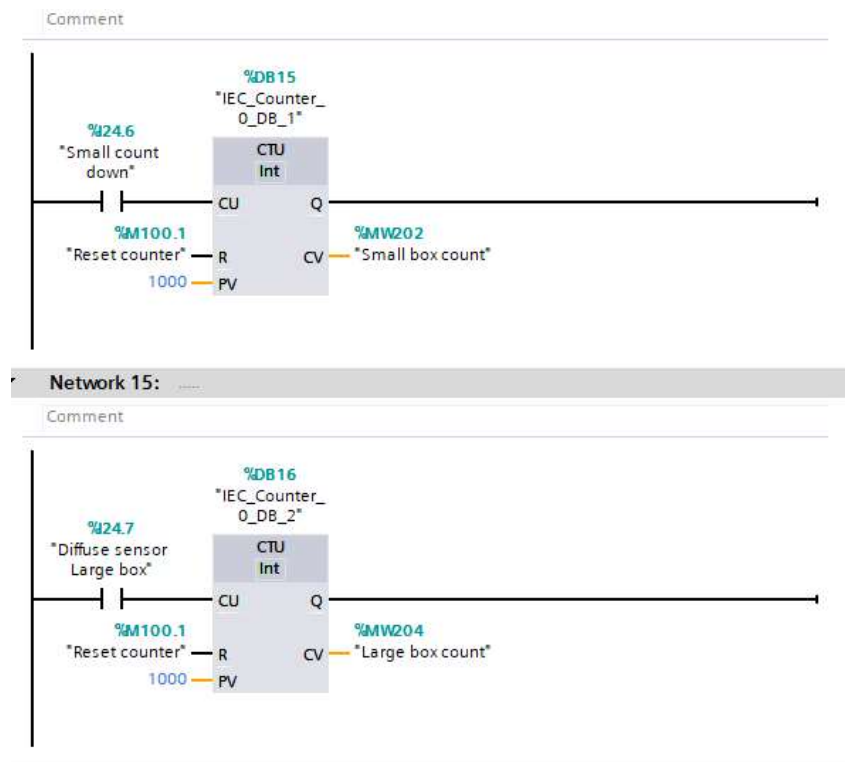


Figure 44 - Small and large box counters

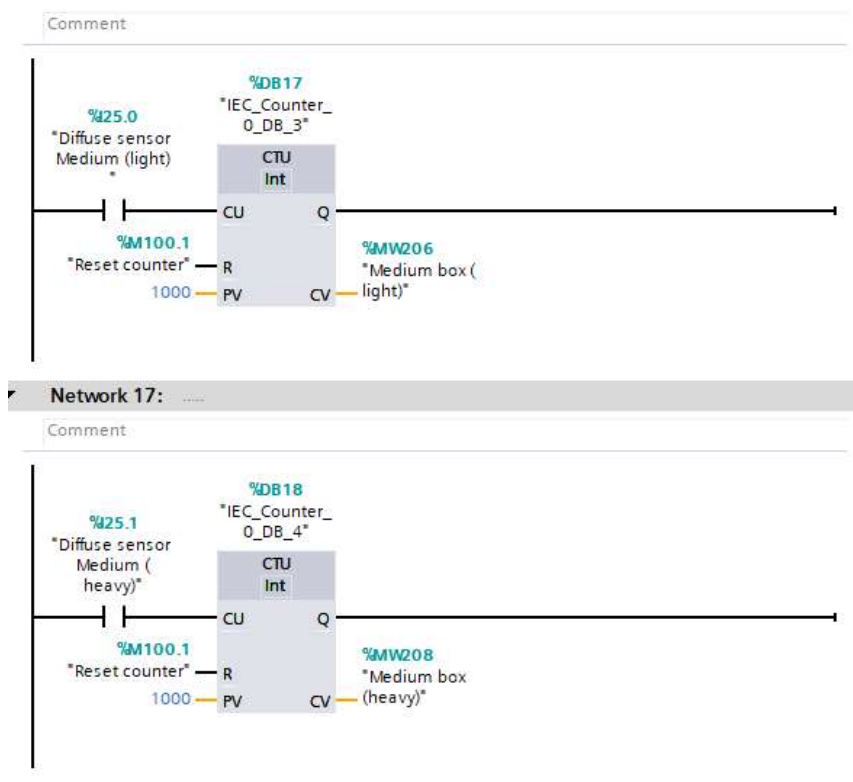


Figure 45 - Medium box counters

The next network, shown in figure 46, controls the warning and alarm system. Each warning and alarm is programmed to a specific byte, this means when any alarm is triggered, the whole byte is triggered. When the warning byte is triggered, it raises a warning “flag” and turns on the yellow global stack lights. When the alarm byte is triggered, it triggers an alarm flag and alarm master, as well as the red global stack lights. Using a pulse timer a siren is triggered by the alarm.

Warnings and alarm bytes

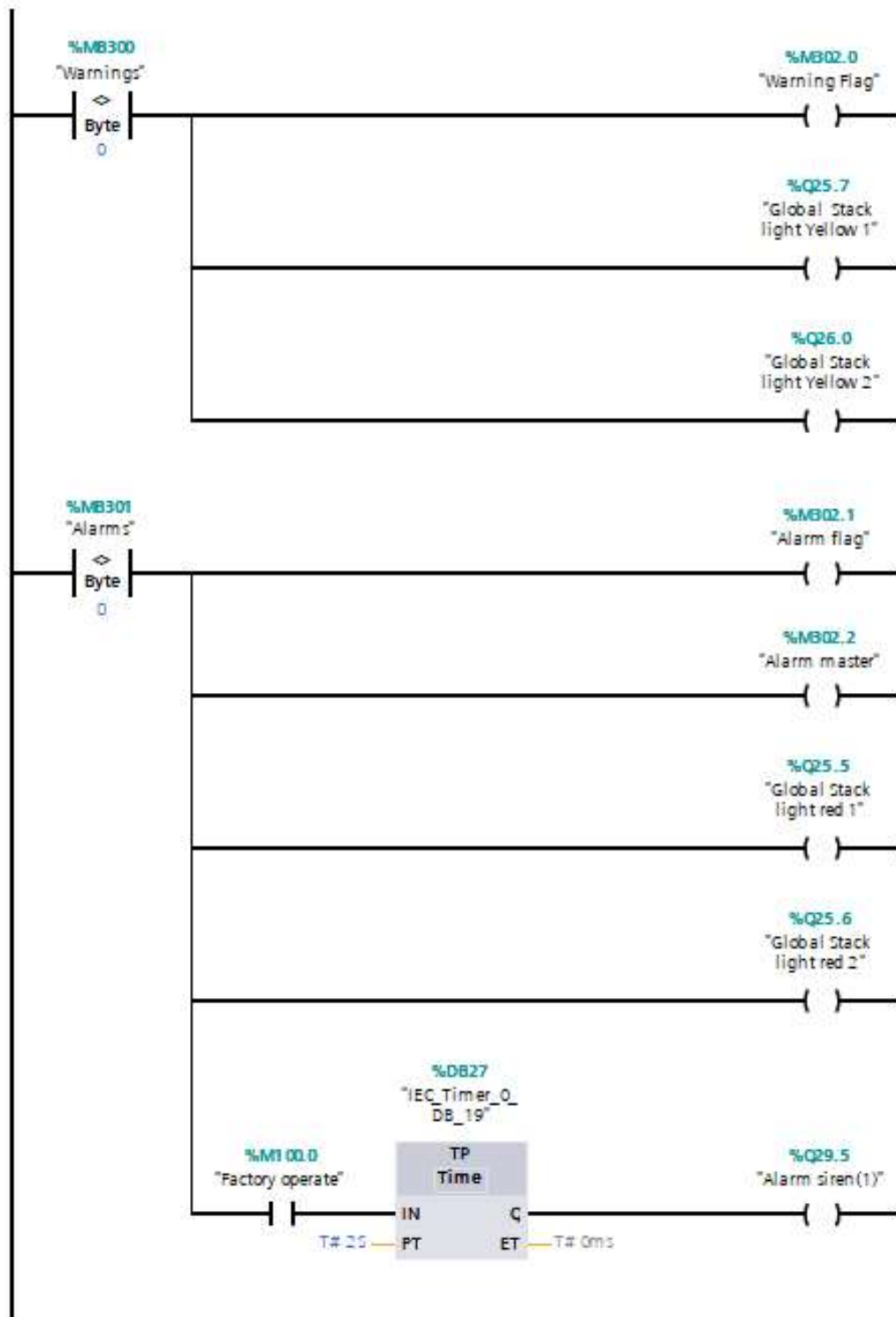


Figure 46 - Warnings and alarms bytes

The next two networks, shown in figures 47 and 48 control the warnings and alarms for the large boxes. Network 19 utilises an up/down counter to determine if a box is missing. The large box pusher counts up the boxes pushed onto the conveyor and then a diffuse sensor at the end of the conveyor counts them down. The output of the counter is stored in a memory write which is then compared to another memory write function set by the engineer's screen. This is done using a greater than or equal to comparator, if the counter value is greater than the engineer's set value the warning or alarm is triggered. A time delay was required to stop the alarm from sounding before the box exited the conveyor. The alarm or warning output was then used to trigger the corresponding stack light.

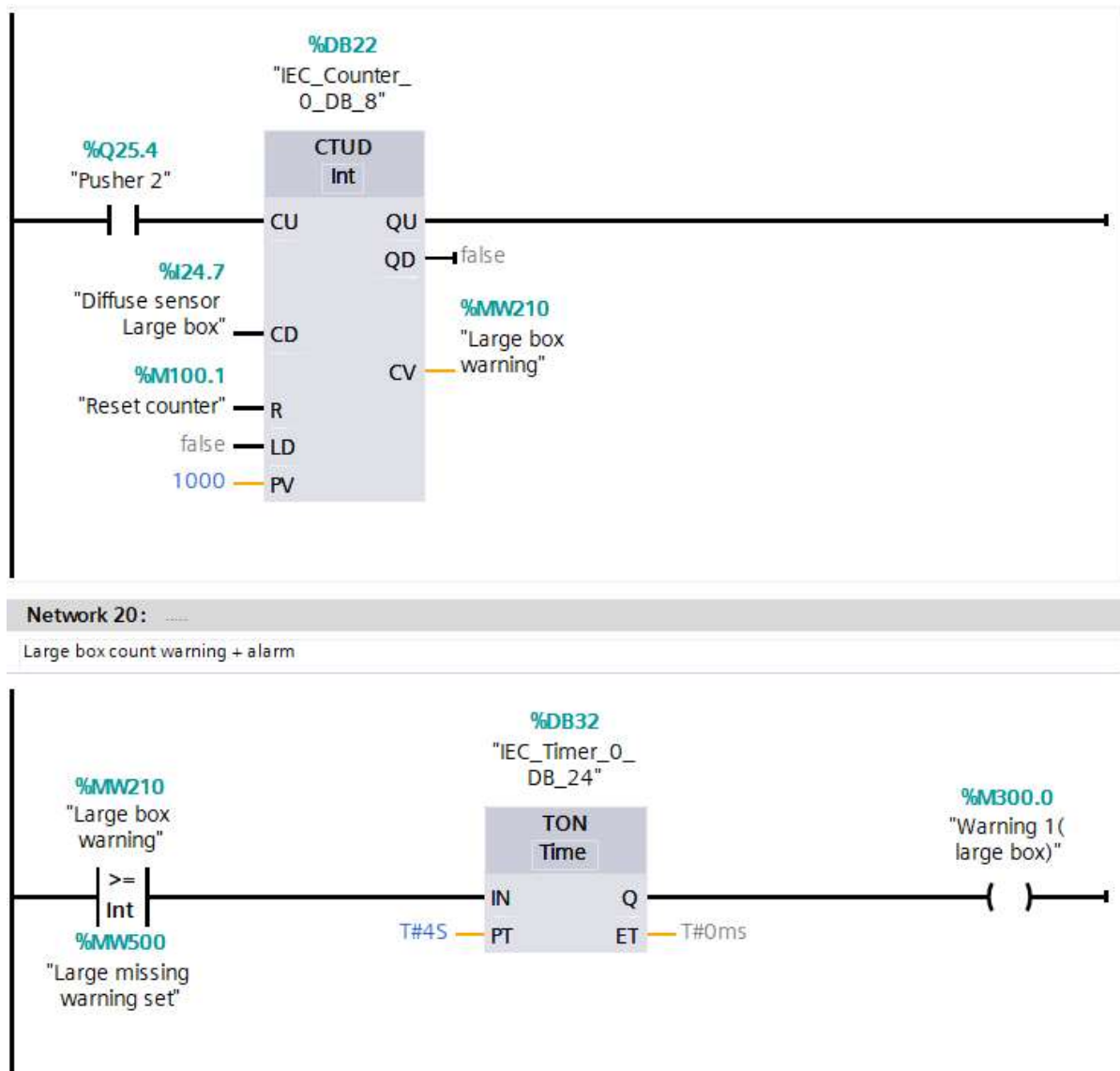


Figure 47 - Large box warnings and alarms

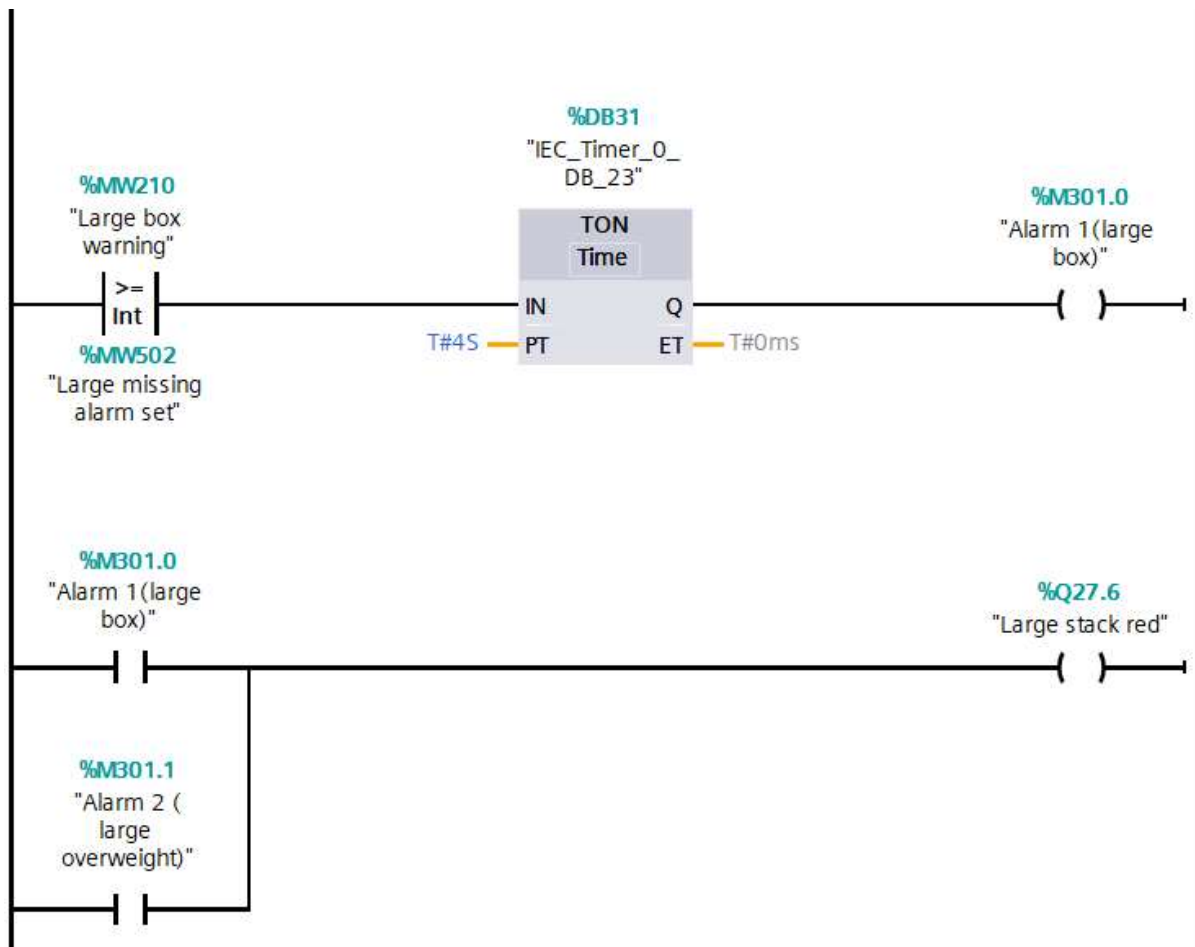


Figure 48 - Large box warning and alarms

Continuing the large box warning system in figure 49, when the red and yellow stack light are not on, the green stack is active. This acts as a visual aid to allow the operator to see which conveyor has a warning or alarm. The large overweight alarm is then activated using a comparator function. When the weight from the scales is present and its over 12 the alarm is triggered. As the weight is only available when the box is on the conveyor, the output is latched on, a NC contact is included to reset the system.

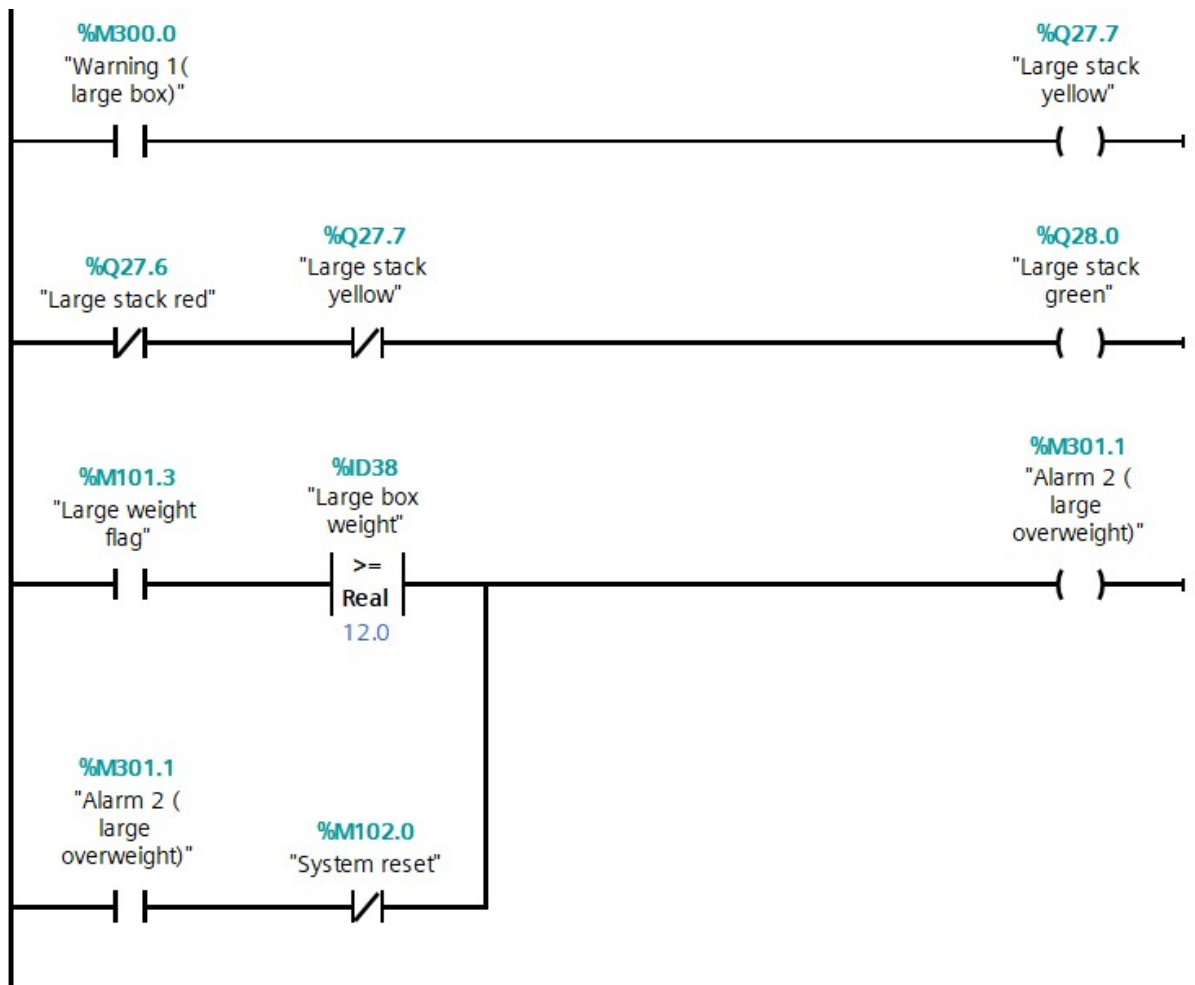


Figure 49 - Large warning and alarm continued

The same warning and alarm system is employed for the small box conveyor as seen below in figures 50 and 51.

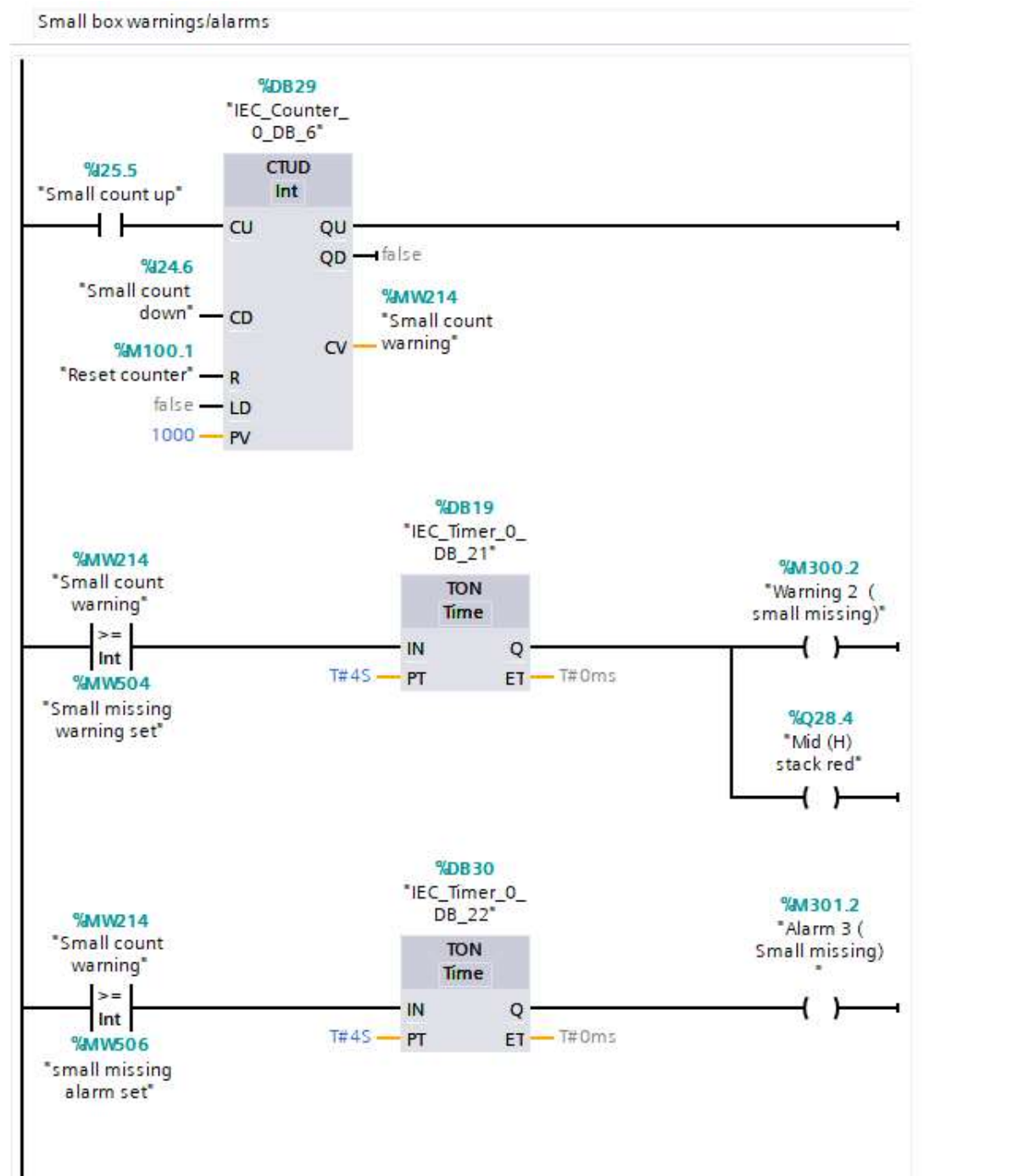


Figure 50 - Small count warning and alarm system part 1

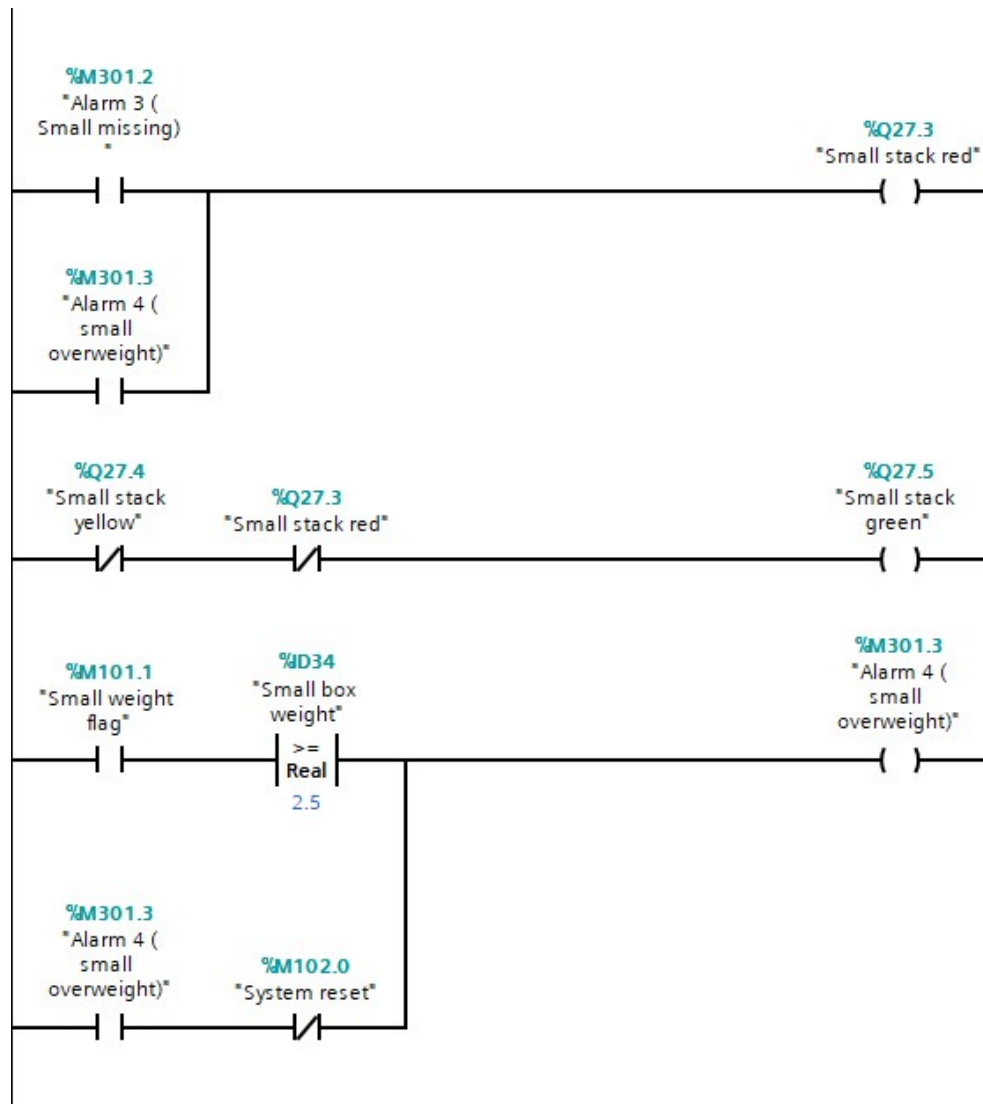


Figure 51 - Small count warning and alarm system part 2

The same system is also employed for the medium box warning and alarm system. The medium systems are split for the lighter medium box and the heavy medium box. The light box system is shown in figures 52 and 53.

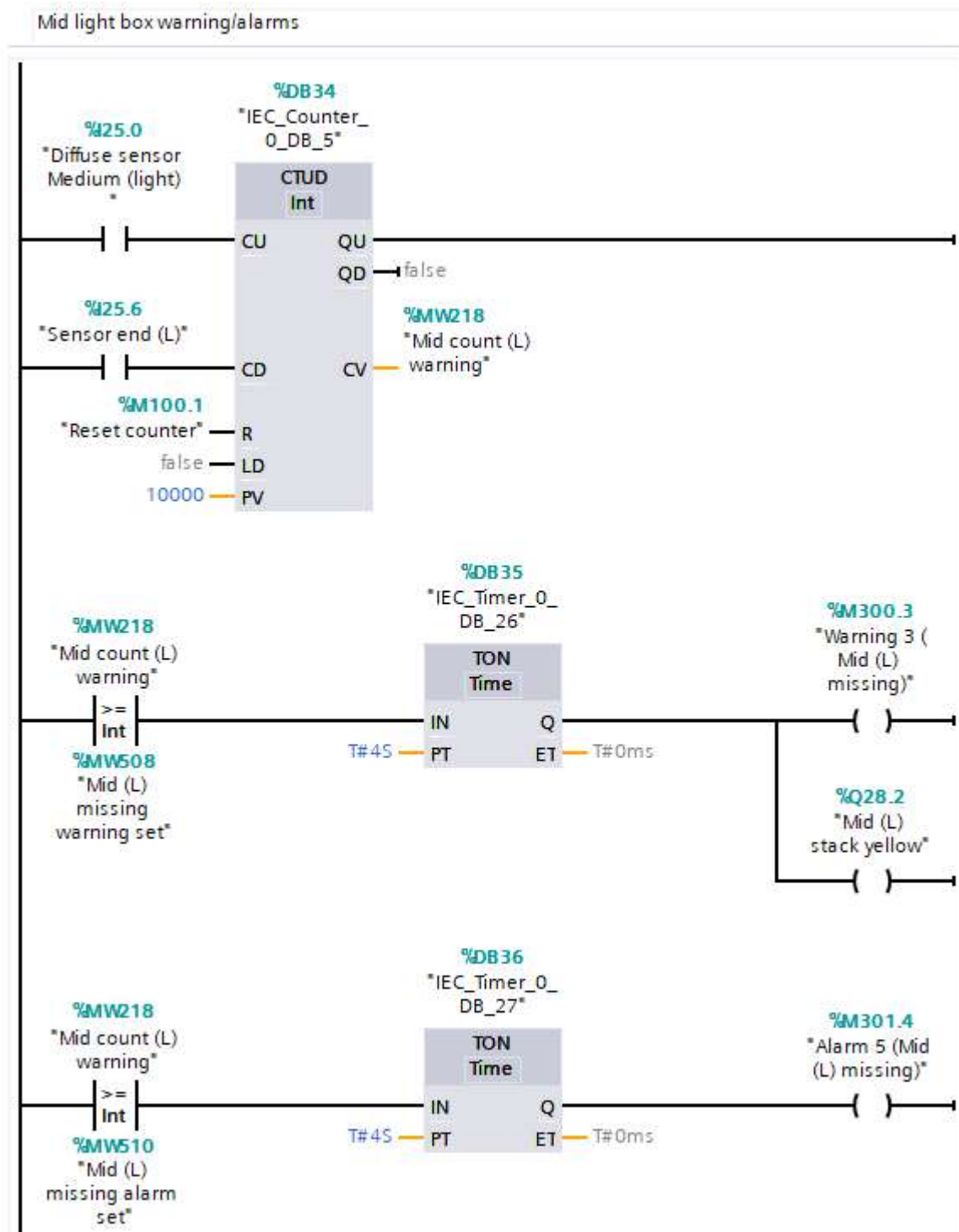


Figure 52 - Light box warning and alarm system part 1

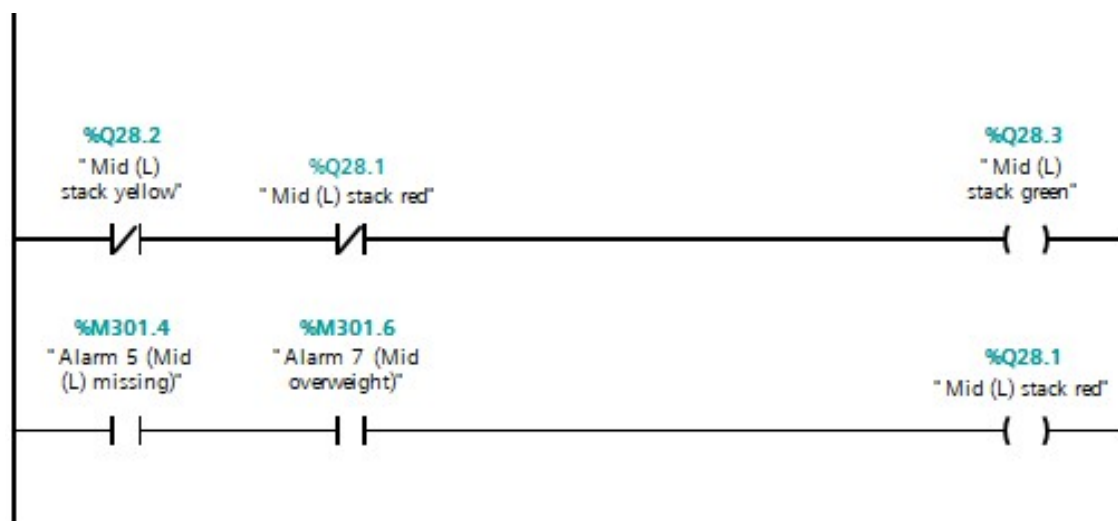


Figure 53 - Light box warning and alarm system part 2

The heavy box warning and alarm system is shown in figures 54 and 55.

▼ Mid heavy box warning/alarms

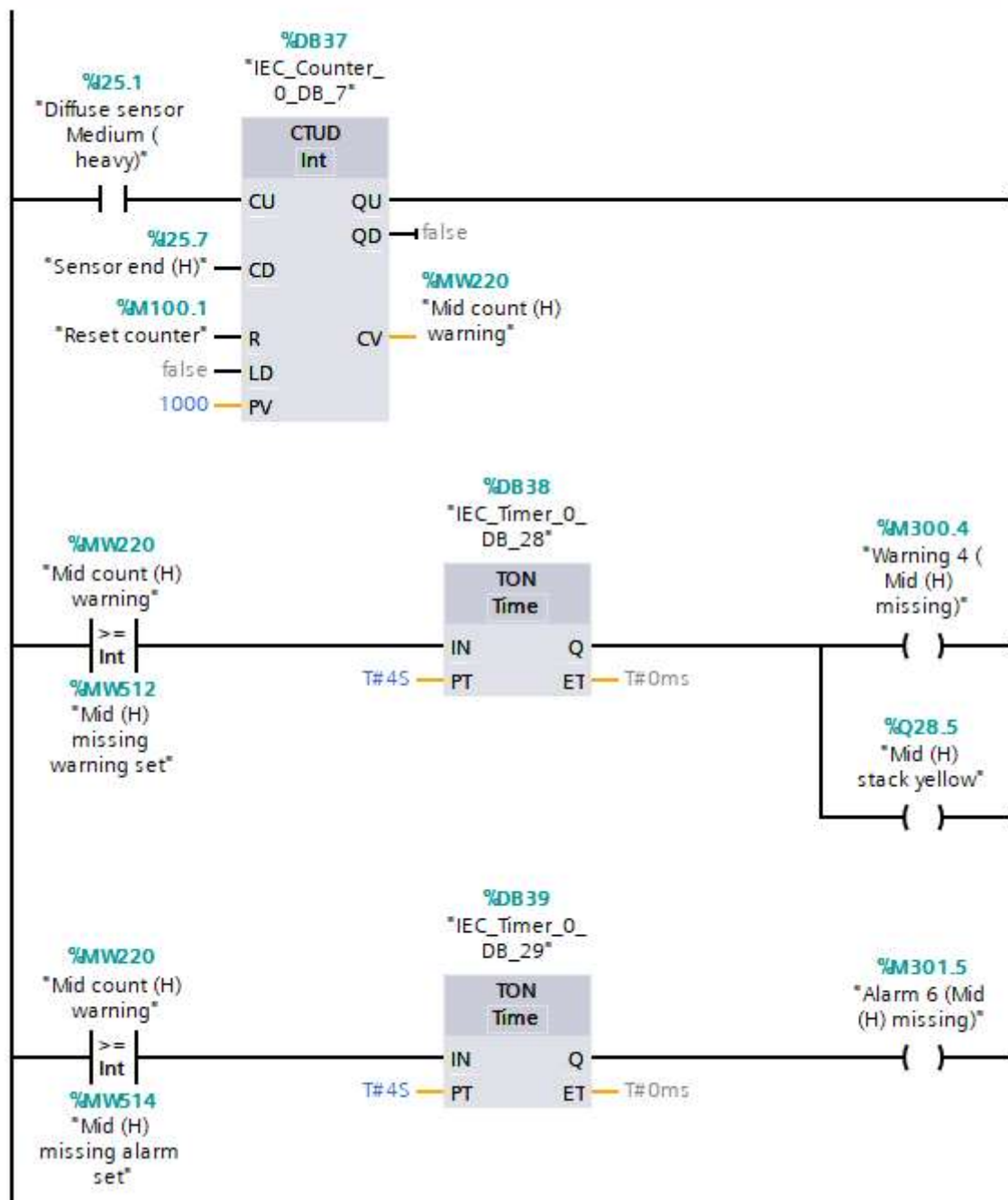


Figure 54 - Heavy box warning and alarm system part 1

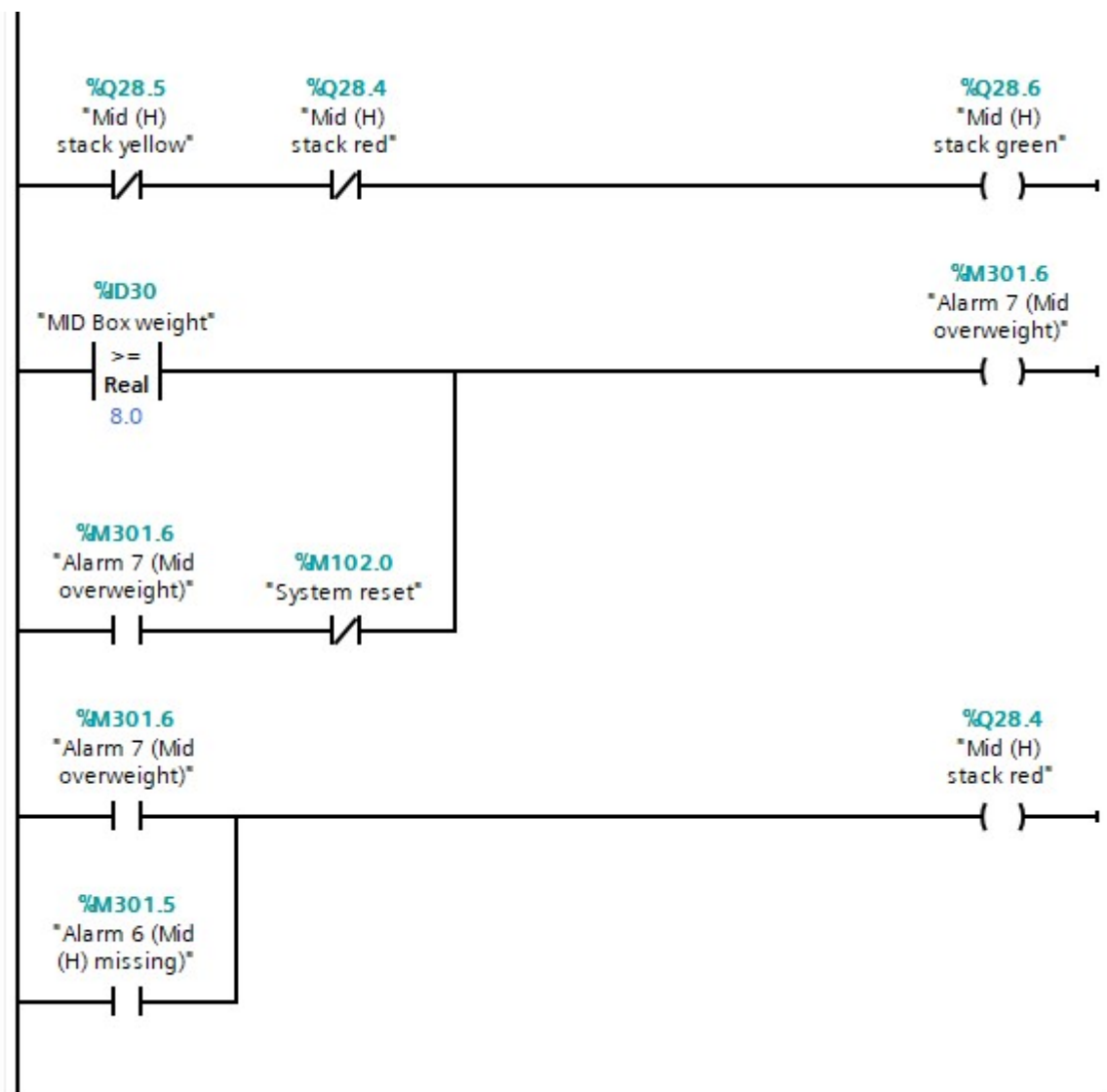


Figure 55 - Heavy box warning and alarm system part 2

The next network, figure 56, controls the total box warnings and alarms. Using an up/down counter, the boxes are counted up using the first diffuse sensor, they're then counted down by each sensor at the end of the conveyors. The value is again compared to a set value from the engineer's screen to trigger the warning. The total box warning must be set to at least 9 boxes to allow for the boxes to reach the end of the conveyor before sounding the warning or alarm.

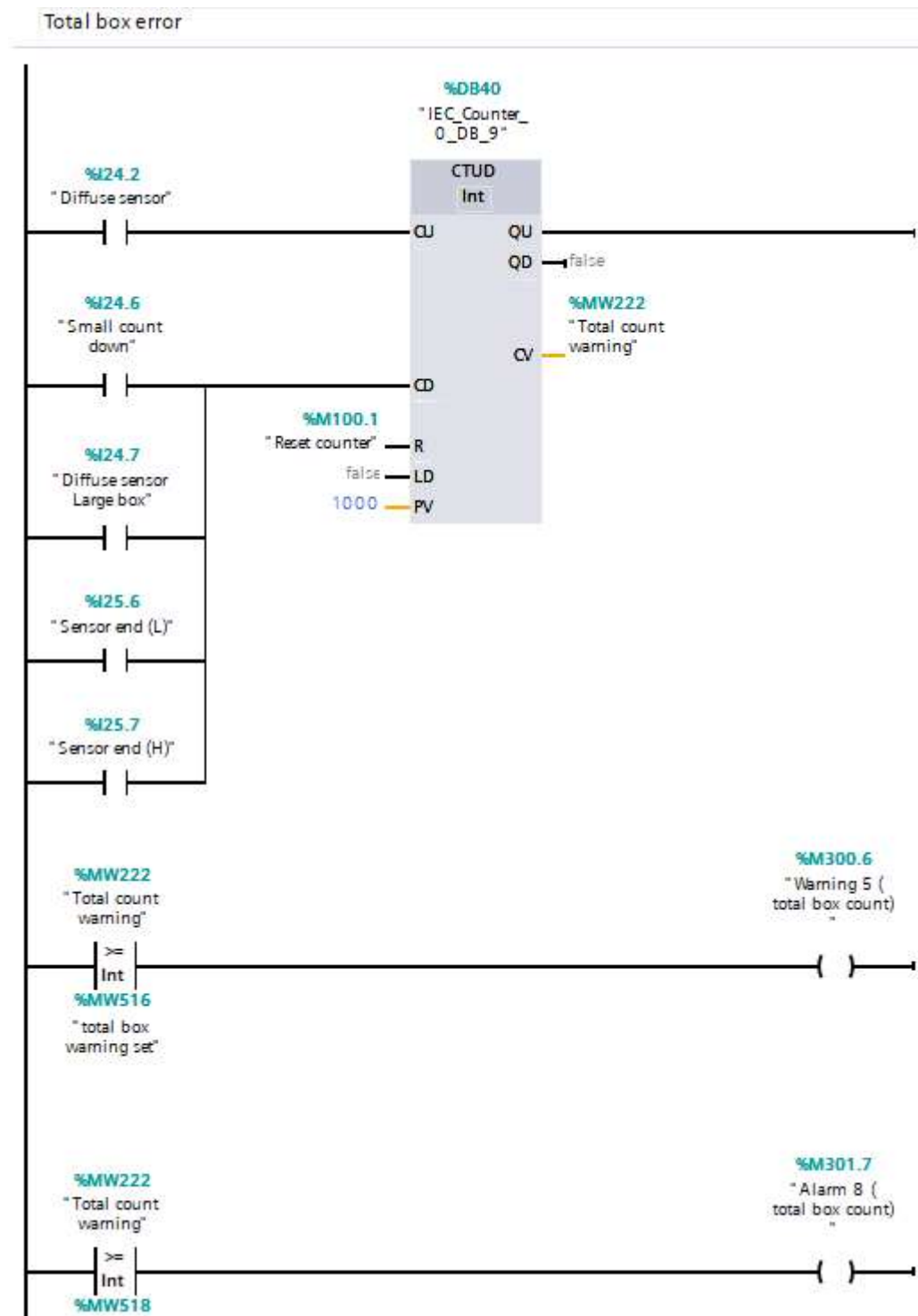


Figure 56 - Total box error warning and alarm system

PLC tags

PLC and HMI tags were labelled to link functions from the factory to the program. Names were given based on the function of the operation [16]. All inputs into the PLC were tagged as shown in figure 57. Inputs are set as a Boolean data type meaning their outputs are either off or on, 0 or 1 [1].

	Name	Tag table	Data type	Address ▲	Retain	Acces...	Writa...	Visibl...
1	Start Button	Default tag table	Bool	%I24.0	☐	☒	☒	☒
2	Stop Button	Default tag table	Bool	%I24.1	☐	☒	☒	☒
3	Diffuse sensor	Default tag table	Bool	%I24.2	☐	☒	☒	☒
4	Diffuse sensor 2	Default tag table	Bool	%I24.3	☐	☒	☒	☒
5	Diffuse sensor 3	Default tag table	Bool	%I24.4	☐	☒	☒	☒
6	Diffuse sensor scale	Default tag table	Bool	%I24.5	☐	☒	☒	☒
7	Small count down	Default tag table	Bool	%I24.6	☐	☒	☒	☒
8	Diffuse sensor Large box	Default tag table	Bool	%I24.7	☐	☒	☒	☒
9	Diffuse sensor Medium (light)	Default tag table	Bool	%I25.0	☐	☒	☒	☒
10	Diffuse sensor Medium (heavy)	Default tag table	Bool	%I25.1	☐	☒	☒	☒
11	E-stop	Default tag table	Bool	%I25.2	☐	☒	☒	☒
12	small scale sensor	Default tag table	Bool	%I25.3	☐	☒	☒	☒
13	Large scale sensor	Default tag table	Bool	%I25.4	☐	☒	☒	☒
14	Small count up	Default tag table	Bool	%I25.5	☐	☒	☒	☒
15	Sensor end (L)	Default tag table	Bool	%I25.6	☐	☒	☒	☒
16	Sensor end (H)	Default tag table	Bool	%I25.7	☐	☒	☒	☒
17	safety door	Default tag table	Bool	%I26.0	☐	☒	☒	☒

Figure 57 - Input tags

Scale weights were tagged as real values as seen in figure 58. A “Real” represents a number as a decimal fraction, in this case it is used as the box weight is not a whole number [1].

18	MID Box weight	Default tag table	Real	%ID30	☐	☒	☒	☒
19	Small box weight	Default tag table	Real	%ID34	☐	☒	☒	☒
20	Large box weight	Default tag table	Real	%ID38	☐	☒	☒	☒

Figure 58 - ID tags

All outputs were tagged as Boolean values as shown in figure 59.

21		start up siren	Default tag table	Bool	%Q24.0				
22		Conveyor 1	Default tag table	Bool	%Q24.1				
23		Conveyor 2	Default tag table	Bool	%Q24.2				
24		Conveyor 3	Default tag table	Bool	%Q24.3				
25		Conveyor 4	Default tag table	Bool	%Q24.4				
26		Scale conveyor	Default tag table	Bool	%Q24.5				
27		Pop up conveyor	Default tag table	Bool	%Q24.6				
28		Pop up left	Default tag table	Bool	%Q24.7				
29		Pop up right	Default tag table	Bool	%Q25.0				
30		Curved left	Default tag table	Bool	%Q25.1				
31		Curved right	Default tag table	Bool	%Q25.2				
32		Pusher 1	Default tag table	Bool	%Q25.3				
33		Pusher 2	Default tag table	Bool	%Q25.4				
34		Global Stack light red 1	Default tag table	Bool	%Q25.5				
35		Global Stack light red 2	Default tag table	Bool	%Q25.6				
36		Global Stack light Yellow 1	Default tag table	Bool	%Q25.7				
37		Global Stack light Yellow 2	Default tag table	Bool	%Q26.0				
38		Global Stack light green 1	Default tag table	Bool	%Q26.1				
39		Global stack light green 2	Default tag table	Bool	%Q26.2				
40		Siren light 1	Default tag table	Bool	%Q26.3				
41		siren light 2	Default tag table	Bool	%Q26.4				
42		Siren light 3 - small box	Default tag table	Bool	%Q26.5				
43		Siren light 4 - Large box	Default tag table	Bool	%Q26.6				
44		Siren light 5 - Medium (light)	Default tag table	Bool	%Q26.7				
45		Siren light 6 - Medium (Heavy)	Default tag table	Bool	%Q27.0				
46		Scale conveyor small	Default tag table	Bool	%Q27.1				
47		Scale conveyor Large	Default tag table	Bool	%Q27.2				
48		Small stack red	Default tag table	Bool	%Q27.3				
49		Small stack yellow	Default tag table	Bool	%Q27.4				
50		Small stack green	Default tag table	Bool	%Q27.5				
51		Large stack red	Default tag table	Bool	%Q27.6				
52		Large stack yellow	Default tag table	Bool	%Q27.7				
53		Large stack green	Default tag table	Bool	%Q28.0				
54		Mid (L) stack red	Default tag table	Bool	%Q28.1				
55		Mid (L) stack yellow	Default tag table	Bool	%Q28.2				
56		Mid (L) stack green	Default tag table	Bool	%Q28.3				
57		Mid (H) stack red	Default tag table	Bool	%Q28.4				
58		Mid (H) stack yellow	Default tag table	Bool	%Q28.5				
59		Mid (H) stack green	Default tag table	Bool	%Q28.6				
60		End conveyor (L)	Default tag table	Bool	%Q28.7				
61		End conveyor (H)	Default tag table	Bool	%Q29.0				
62		Alarm siren(1)	Default tag table	Bool	%Q29.5				
63		Green Indicator	Default tag table	Bool	%Q29.6				
64		Red indicator	Default tag table	Bool	%Q29.7				

Figure 59 - Output tags

Outputs saved to memory locations were tagged as shown in figure 60.

68		Mem weight	Default tag table	Real	%MD34	☐	☒	☒	☒
69		Factory operate	Default tag table	Bool	%M100.0	☐	☒	☒	☒
70		Reset counter	Default tag table	Bool	%M100.1	☐	☒	☒	☒
71		Mid Weight flag	Default tag table	Bool	%M100.2	☐	☒	☒	☒
72		Mid Loaded box	Default tag table	Bool	%M100.3	☐	☒	☒	☒
73		Mid Weighing	Default tag table	Bool	%M100.4	☐	☒	☒	☒
74		Emergency stop	Default tag table	Bool	%M100.5	☐	☒	☒	☒
75		e-button	Default tag table	Bool	%M100.6	☐	☒	☒	☒
76		Small box loaded	Default tag table	Bool	%M100.7	☐	☒	☒	☒
77		Large box loaded	Default tag table	Bool	%M101.0	☐	☒	☒	☒
78		Small weight flag	Default tag table	Bool	%M101.1	☐	☒	☒	☒
79		Small weighing	Default tag table	Bool	%M101.2	☐	☒	☒	☒
80		Large weight flag	Default tag table	Bool	%M101.3	☐	☒	☒	☒
81		Large weighing	Default tag table	Bool	%M101.4	☐	☒	☒	☒
82		Global stack green	Default tag table	Bool	%M101.5	☐	☒	☒	☒
83		Global stack yellow	Default tag table	Bool	%M101.6	☐	☒	☒	☒
84		Global stack red	Default tag table	Bool	%M101.7	☐	☒	☒	☒
85		System reset	Default tag table	Bool	%M102.0	☐	☒	☒	☒

Figure 60 - Memory tags

Operations used to write to memory were saved as integers tagged as shown in figure 61. An integer refers to a signed number from -32768 to +32768, this refers to the number of bits available in the PLC [1].

86		Total count	Default tag table	Int	%MW200	☐	☒	☒	☒
87		Small box count	Default tag table	Int	%MW202	☐	☒	☒	☒
88		Large box count	Default tag table	Int	%MW204	☐	☒	☒	☒
89		Medium box (light)	Default tag table	Int	%MW206	☐	☒	☒	☒
90		Medium box (heavy)	Default tag table	Int	%MW208	☐	☒	☒	☒
91		Large box warning	Default tag table	Int	%MW210	☐	☒	☒	☒
92		Mid Box warning	Default tag table	Int	%MW212	☐	☒	☒	☒
93		Small count warning	Default tag table	Int	%MW214	☐	☒	☒	☒
94		Small count alarm	Default tag table	Int	%MW216	☐	☒	☒	☒
95		Mid count (L) warning	Default tag table	Int	%MW218	☐	☒	☒	☒
96		Mid count (H) warning	Default tag table	Int	%MW220	☐	☒	☒	☒
97		Total count warning	Default tag table	Int	%MW222	☐	☒	☒	☒
116		Large missing warning set	Default tag table	Int	%MW500	☐	☒	☒	☒
117		Large missing alarm set	Default tag table	Int	%MW502	☐	☒	☒	☒
118		Small missing warning set	Default tag table	Int	%MW504	☐	☒	☒	☒
119		small missing alarm set	Default tag table	Int	%MW506	☐	☒	☒	☒
120		Mid (L) missing warning set	Default tag table	Int	%MW508	☐	☒	☒	☒
121		Mid (L) missing alarm set	Default tag table	Int	%MW510	☐	☒	☒	☒
122		Mid (H) missing warning set	Default tag table	Int	%MW512	☐	☒	☒	☒
123		Mid (H) missing alarm set	Default tag table	Int	%MW514	☐	☒	☒	☒
124		total box warning set	Default tag table	Int	%MW516	☐	☒	☒	☒
125		Total box alarm set	Default tag table	Int	%MW518	☐	☒	☒	☒

Figure 61 - Memory write tags

Warnings and alarms were saved to bytes as shown in figure 62. The Boolean outputs are stored to the same byte of memory [1] .

98		Warnings	Default tag table	Byte	%MB300	☐	☒	☒	☒
99		Warning 1 (large box)	Default tag table	Bool	%M300.0	☐	☒	☒	☒
100		Warning 2 (small missing)	Default tag table	Bool	%M300.2	☐	☒	☒	☒
101		Warning 3 (Mid (L) missing)	Default tag table	Bool	%M300.3	☐	☒	☒	☒
102		Warning 4 (Mid (H) missing)	Default tag table	Bool	%M300.4	☐	☒	☒	☒
103		Warning 5 (total box count)	Default tag table	Bool	%M300.6	☐	☒	☒	☒
104		Alarms	Default tag table	Byte	%MB301	☐	☒	☒	☒
105		Alarm 1 (large box)	Default tag table	Bool	%M301.0	☐	☒	☒	☒
106		Alarm 2 (large overweight)	Default tag table	Bool	%M301.1	☐	☒	☒	☒
107		Alarm 3 (Small missing)	Default tag table	Bool	%M301.2	☐	☒	☒	☒
108		Alarm 4 (small overweight)	Default tag table	Bool	%M301.3	☐	☒	☒	☒
109		Alarm 5 (Mid (L) missing)	Default tag table	Bool	%M301.4	☐	☒	☒	☒
110		Alarm 6 (Mid (H) missing)	Default tag table	Bool	%M301.5	☐	☒	☒	☒
111		Alarm 7 (Mid overweight)	Default tag table	Bool	%M301.6	☐	☒	☒	☒
112		Warning Flag	Default tag table	Bool	%M302.0	☐	☒	☒	☒
113		Alarm flag	Default tag table	Bool	%M302.1	☐	☒	☒	☒
114		Alarm master	Default tag table	Bool	%M302.2	☐	☒	☒	☒
115		Alarm 8 (total box count)	Default tag table 	Bool	 %M301.7 	☐	☒	☒	☒

Figure 62 - Warnings and alarm tags

Conclusion

A PLC program was designed to control a factory, sorting four different boxes by size and weight. The program functions by taking inputs from diffuse sensors throughout the factory and performing the required operation. Large and medium boxes activate the sensor, causing a pusher to divert them to a secondary conveyor. Small boxes continue down the main conveyor where they are weighed and then removed. Large boxes then trigger another diffuse sensor, causing them to be pushed off the secondary conveyor. They are then weighed and removed from the system. The medium boxes continue down the secondary conveyor until they are sorted by a pop-up sorter diverting them left or right based on their weight.

If a box of any size goes missing in the system an alarm or warning is triggered controlled by parameters input by the engineer. Boxes that are over a set weight will also trigger an alarm.

The designed system functions as required. Due to the method of design and lack of planning, some PLC tags or function labels may not be in a logical order. To overcome this downfall, the factory and program should have been planned out beforehand. This would ensure all tags and labels would be in a logical order.

Bibliography

- [1] W. Bolton, Programmable logic controllers, Sixth edition ed., Kidlington, Oxford: Elsevier, 2015.
- [2] RS, "PLCs - A Complete Guide," RS, 2024 04 05. [Online]. Available: <https://uk.rs-online.com/web/content/discovery/ideas-and-advice/plcs-programmable-logic-controllers-guide>. [Accessed 14 04 2024].
- [3] A. Rullán, "Programmable logic controllers versus personal computers for process control," *Computers & Industrial Engineering*, vol. 33, no. 1-2, p. 422, 1997.
- [4] E. R. Alphonsus and M. O. Abdullah, "A review on the applications of programmable logic controllers (PLCs)," *Renewable and Sustainable Energy Reviews*, vol. 60, pp. 1185-1205, 2016.
- [5] M. G. Hudedmani, U. R. M, S. K. Kabberalli and R. Hittalamani, "Programmable Logic Controller (PLC) in Automation," *Advance journal of Graduate research*, vol. 2, no. 1, pp. 37-45, 07 2017.
- [6] T. L. Bartelt, Industrial Electronics, Albany, New York: Delmar, 1997.
- [7] L. Miller, "PLC Programming Basics using Ladder Logic," Learn robotics , 11 03 2020. [Online]. Available: <https://www.learnrobotics.org/blog/plc-programming-basics/>. [Accessed 14 04 2024].
- [8] RS, "Omron CP1L Series PLC CPU for Use with SYSMAC CP1L Series, Relay Output, 18 (DC)-Input, DC Input," RS, 14 04 2024. [Online]. Available: <https://uk.rs-online.com/web/p/plcs-programmable-logic-controllers/0401573?gb=s>. [Accessed 14 04 2024].
- [9] H. Jack, Automating manufacturing systems with PLCs, 7.0 ed., Hugh Jack, 2010.
- [10] G. Dunning, Introduction to programmable logic controllers, Albany, New York: Delmar, 1998.
- [11] Mitsubishi electric, "Factory Automation," Mitsubishi electric, 14 04 2024. [Online]. Available: <https://www.mitsubishielectric.com/fa/products/cnt/plcr/items/index.html>. [Accessed 14 04 2024].
- [12] RS, "Siemens SIMATIC S7-1200 Series PLC CPU for Use with SIMATIC S7-1200 Series, Digital, Transistor Output, 14 (Digital, 2," RS, 04 05 2024. [Online]. Available: <https://uk.rs-online.com/web/p/plcs-programmable-logic-controllers/8624471?gb=s>. [Accessed 04 05 2024].
- [13] Frequent-Penalty247, "PLC," 2022. [Online]. Available: https://www.reddit.com/r/PLC/comments/xa7xvo/what_do_you_guys_like_and_dislike_about_micro/. [Accessed 04 05 2024].

- [14 RS, "Allen Bradley Micro870 Series PLC CPU, 24V Source Output, 14-Input, Voltage Input," RS, 04 05 2024. [Online]. Available: <https://uk.rs-online.com/web/p/plcs-programmable-logic-controllers/1833562?gb=s>. [Accessed 04 05 2024].
- [15 RS, "Mitsubishi FX5U Series PLC CPU for Use with MELSEC IQ-F Series IQ Platform-Compatible PLC, Relay, Transistor Output," RS, 04 05 2024. [Online]. Available: <https://uk.rs-online.com/web/p/plcs-programmable-logic-controllers/8755672?gb=s>. [Accessed 04 05 2024].
- [16 R. C. Martin, Clean code, Boston, Massachusetts: Prentice Hall, 2008.
-]