RIT

Proposal for Primary Packaging of Bottles

Purpose, environment, configuration, devices and safety measures

by

Matheus Tararam de Laurentys

Final Report
MFET0340 - Automation Control Systems

Directed to: Prefessor Michael Slifka

Report submitted in partial fulfillment of the
requirements for the final project

at

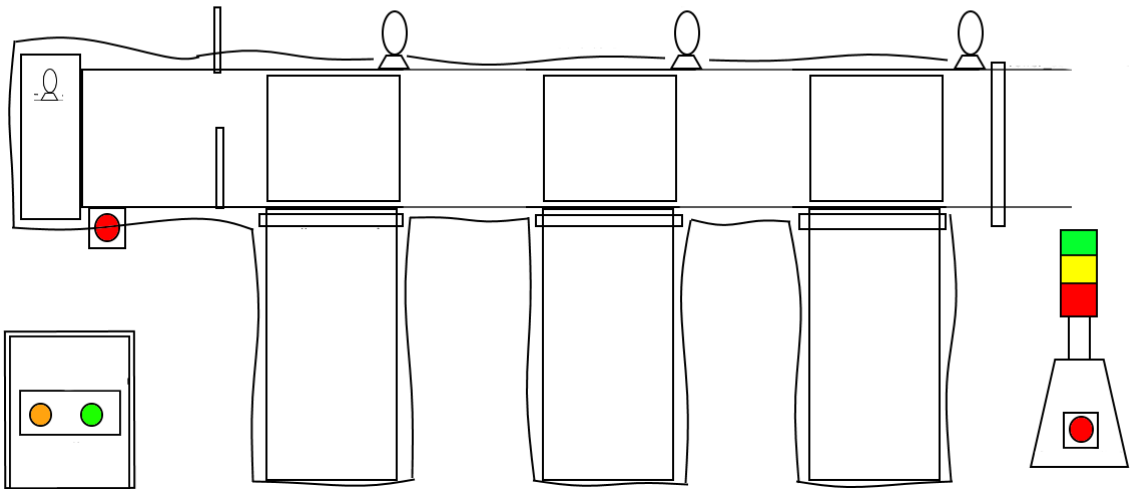
Rochester Institute of Technology
Department of Manufacturing and Mechanical Engineering Technology
December 2021

1 Executive Summary

The operation of filling bottles, from those of drinkable waters to containers of washing machine detergent, is highly automated. The high volume and low production cost requirements of these products make automation the only viable option. This report features a system capable of operating with different types of containers and varied liquid materials. The report also dives deep into a program that simulates the production line of bottled liquids. With some adaptations the program can be used to operate on the physical implementation of the system.

The system uses two feeders, one for bottles and one for liquids. When operating, the bottles are moved into a container and get rinsed, filled and caped. Finally the bottles of the same type and liquid, grouped together, are made available to packaging. To operate, the system uses a PLC program that verifies the count of bottles in the feeder to coordinate with moving of products to the conveyor and verifies volume of liquid available to coordinate with filling of bottles.

Other than the two feeders, the system requires a set of colored button devices, a bottle filling machine, a bottle conveyor and the PLC itself. The PLC used is Rockwell's model 5069-L306ER, with models 5069-IB6F and 5069-OB16 as input and output modules.



2 Full System Description

The proposed bottle filling system ("System") solves the problem of rinsing, filling and packaging bottles in high speed with low operation cost. When compared to manual processes the advantage is clear in terms of economy but the System also meets more complex business requirements. In a paint manufacture, for example, several colors of paint are produced, bottled and packaged, but each paint cannot get mixed with the others.

Not only that, but typically black and white paint ship in larger bottles, while colored paint sells is smaller containers. The feeder of bottles of such machines must be able to comport containers of different sizes. The feeder of paint must also be able to be filled with different colors of paint. In the case of paint, the feeder must be cleaned so colors do not get mixed whenever the paint is changed.Other liquid have different requirements, such as toxicity removal or cleaning through chemical reactions. See the following graphic of the System:

Bottles flow from the feeder onto the conveyor. Once on the conveyor, the bottles are rinsed and proceed to the liquid filler. Once a predetermined amount of bottles reach the liquid filler, the liquid flows from the liquid feeder to the bottles. After it is filled the bottle must be caped and moved to a predetermined area. The process of moving may involve sorting or could be directly connected to the startup configuration of the feeders. The conveyor part System featured only fills the bottle.

3 Logic Flow Chart

The flow and logic of the System that is implemented in the form of a PLC program. This flow simulates the behavior of the proposed sorting conveyor.

4 Development Spreadsheet

	TYPE	Logic Function	Name	I/O	DATATYPE	BaseTag
1	ALIAS	There is room for more A items on collection point	CONVEYOR_A_GREEN	OUTPUT	DIGITAL	Local:6:O.Pt05.Data
2	ALIAS	Collection point A is full	CONVEYOR_A_RED	OUTPUT	DIGITAL	Local:6:O.Pt06.Data
3	ALIAS	There is room for more C items on collection point	CONVEYOR_C_GREEN	OUTPUT	DIGITAL	Local:6:O.Pt07.Data
4	ALIAS	Collection point C is full	CONVEYOR_C_RED	OUTPUT	DIGITAL	Local:6:O.Pt08.Data
5	ALIAS	Indicates sorter found an A product	CONVEYOR_SOL_A	OUTPUT	DIGITAL	Local:6:O.Pt02.Data
6	ALIAS	Indicates sorter found an C product	CONVEYOR_SOL_B	OUTPUT	DIGITAL	Local:6:O.Pt03.Data
7	ALIAS	Feeder is full, system can start	feeder_green	OUTPUT	DIGITAL	Local:6:O.Pt00.Data
8	ALIAS	Operation stopped due to state of feeder. Overfilling or starting with feeder not full cause this light to blink.	feeder_red	OUTPUT	DIGITAL	Local:6:O.Pt01.Data
9	ALIAS	Collected total amount of C products	TRAFFIC_A_GREEN	OUTPUT	DIGITAL	Local:5:O.Pt07.Data
10	ALIAS	Collected total amount of B products	TRAFFIC_A_RED	OUTPUT	DIGITAL	Local:5:O.Pt05.Data
11	ALIAS	Collected total amount of A products	TRAFFIC_A_YELLOW	OUTPUT	DIGITAL	Local:5:O.Pt06.Data
12	ALIAS	System is off, personnel can interact	tower_green	OUTPUT	DIGITAL	Local:6:O.Pt15.Data
13	ALIAS	System is on, personnel should pay attention	tower_red	OUTPUT	DIGITAL	Local:6:O.Pt13.Data
14	ALIAS	Either a collection point is full or feeder is empty	tower_yellow	OUTPUT	DIGITAL	Local:6:O.Pt14.Data
15	ALIAS	Indicates the main system is operating	motor_1	OUTPUT	DIGITAL	Local:4:O.Pt14.Data
16	ALIAS	Indicates shutting off or recover system are operating	motor_2	OUTPUT	DIGITAL	Local:4:O.Pt15.Data
17	ALIAS	Immediately shuts off the system	e_stop_1	INPUT	DIGITAL	Local:1:I.Pt01.Data
18	ALIAS	Immediately shuts off the system	e_stop_2	INPUT	DIGITAL	Local:2:I.Pt01.Data
19	ALIAS		PB1_NO	INPUT	DIGITAL	Local:1:I.Pt03.Data
20	ALIAS	When system is operating, triggers the system to sort next item in the conveyor	PB3_NO	INPUT	DIGITAL	Local:1:I.Pt07.Data
21	ALIAS	Shuts off the main system and starts shuttof system	PB4_NO	INPUT	DIGITAL	Local:1:I.Pt09.Data
22	ALIAS	Triggers system boot	PB5_NO	INPUT	DIGITAL	Local:1:I.Pt11.Data
23	ALIAS		SW3_NO_left	INPUT	DIGITAL	Local:2:I.Pt07.Data
24	ALIAS	Indicates next item in the conveyor is of type C	SW3_NO_right	INPUT	DIGITAL	Local:2:I.Pt09.Data
25	ALIAS	Helper variable used to avoid direct access to NC input. NC is necessary because there is NO for center position	SW4_aux1	INPUT	DIGITAL	Local:3:I.Pt05.Data
26	ALIAS	Indicates next item in the conveyor is of type A	SW4_left	INPUT	DIGITAL	Local:3:I.Pt01.Data
27	ALIAS	Indicates next item in the conveyor is of type C	SW4_right	INPUT	DIGITAL	Local:3:I.Pt00.Data
28	TAG	Indicates collection point A is full. True if collection point is full or total amount of product As were sorted sorted.	A_full		BOOL	
29	TAG	Indicates collection point C is full. True if collection point is full or total amount of product As were sorted sorted.	C_full		BOOL	
30	TAG	Couter for how many products are in line A	A_in_collection_point		COUNTER	
31	TAG	Couter for how many products are in line C	C_in_collection_point		COUNTER	
32	TAG	Counter of total sorted products A	A_total		COUNTER	
33	TAG	Counter of total sorted products B	B_total		COUNTER	
34	TAG	Counter of total sorted products C	C_total		COUNTER	
35	TAG	Counter of how many products were sorted A but did not go through to line A	A_in_D		COUNTER	
36	TAG	Counter of how many products were sorted B but did not go through to line B	B_in_D		COUNTER	
37	TAG	Counter of how many products were sorted C but did not go through to line C	C_in_D		COUNTER	
38	TAG	Counter of how many items are in the conveyor	conveyor		COUNTER	
39	TAG	Counter for the maximum capacity of the feeder	feeder_limit		COUNTER	
40	TAG	Counter to verify whether the feeder became empty	feeder_not_empty		COUNTER	
41	TAG	Timer used to control when feeder should put products in the container	feed_timer		TIMER	
42	TAG	Prevents the feeder from counting multiple products when only one was added	fill_feeder_ons		BOOL	
43	TAG	Timer used to delay system activation to give personnel time to react to red lights and horns	motor_start_delay		TIMER	
44	TAG	Auxiliary variable true when state is correct and operator triggers a sorf of an A type product	operate_SOL_A		BOOL	
45	TAG	Auxiliary variable true when state is correct and operator triggers a sorf of an B type product	operate_SOL_B		BOOL	
46	TAG	Auxiliary variable true when state is correct and operator triggers a sorf of an C type product	operate_SOL_C		BOOL	
47	TAG	Auxiliary variable true when system is on. Note that when system is shutting down, this is false	system_on		BOOL	
48	TAG	Auxiliary variable true when shutting is powering off	shutting_off		BOOL	
49	TAG	Auxiliary variable true when system is a recovery process	recover		BOOL	
50	TAG	Delay used to simulate an actual recovery process	recover_simulator		TIMER	
51	TAG	Delay used to simulate an actual shutoff process	shutoff_simulator		TIMER	
52	TAG	Timer used to delay recover system activation to give personnel time to react to red lights and horns	recover_start_delay		TIMER	
53	TAG	Auxiliary variable used to control indicator light FEEDER_RED	show_feeder_problem		TIMER	
54	TAG	Auxiliary variable true when SW4 is center	SW4_center		BOOL	

5 PLC definition and specification

Components

Controller selected:

- 5069-L306ER
- 8 I/Os
- https://configurator.rockwellautomation.com/#/configurator/E6650EFBABB645899D338EDD0D78A095/summary

Input Module selected:

- 5069-IB16F
- 16 points
- https://configurator.rockwellautomation.com/#/configurator/5069-IB16F/summary

Output Module selected:

- 5069-OB16
- 16 points
- https://configurator.rockwellautomation.com/#/configurator/5069-OB16/summary

Connections

Find in the table below a possible way to connect the devices listed to the PLC points. Not that only one input and output module are required.

Alias Tag (1)	Base Tag (10)	I/O (3)	Slot (9)	Device Type (4)	Manufacturer (5)	Device PN (6)	Contact(s) (7)	Notes (8)
estop_conveyor	Local:1:I:Pt01.Data	Input	1	Estop rope/PB	Banner Engineering	RP-RM83F	1	Both the rope and PB use the same tag
estop_tower	Local:1:I:Pt02.Data	Input	1	Estop button	Banner Engineering	SSA-EB	1	
power_on	Local:1:I:Pt03.Data	Input	1	Momentary PB	Banner Engineering	K50LYXXPPB2Q	1	
power_off	Local:1:I:Pt04.Data	Input	1	Momentary PB	Banner Engineering	K50LYXXPPB2Q	1	
Sorter	Local:1:I:Pt05.Data	Input	1	Sensor for Object detection	Banner Engineering	LX3EQ	1	
roller_sensorA	Local:1:I:Pt06.Data	input	1	Proximity Sensor	Banner Engineering	TTR1219RPSD7T	1	
roller_sensorB	Local:1:I:Pt07.Data							
roller_sensorC	Local:1:I:Pt08.Data							
roller_sensor_over	Local:1:I:Pt09.Data							
DiverterA	Local:2:O:Pt01.Data	output	2	Steerable wheel sorter	Nido Automation	DENEB-PU	1	
DiverterB	Local:2:O:Pt02.Data							
DiverterC	Local:2:O:Pt03.Data							
A_full_indicator	Local:2:O:Pt04.Data	output	2	Multi-color Indicator light	Banner Engineering	K50LGRY2NQ	1	Multi-color indicator lights seem to require a special cable
B_full_indicator	Local:2:O:Pt05.Data							
C_full_indicator	Local:2:O:Pt06.Data							
feeder_indicator	Local:2:O:Pt07.Data							
tower_green	Local:2:O:Pt08.Data	output	2	Indicator Tower Light	Banner Engineering	TL70GYRQ	1	These three indicator lights are located in the same device
tower_yellow	Local:2:O:Pt09.Data	output	2					
tower_red	Local:2:O:Pt10.Data	output	2					
Feeder	Local:2:O:Pt11.Data	input/output	2	Feeder	Rovibec Agrisolutions	CC718	2	
	Local:1:I:Pt10.Data		1					
gravity_rollerA	Local:2:O:Pt12.Data	output	2	Automated Gravity Roller with breaks	Materials Handling	QTY 10	1	Prove the products to the end of queue avoiding collision
gravity_rollerB	Local:2:O:Pt13.Data	output	2					
gravity_rollerC	Local:2:O:Pt14.Data	output	2					

6 Specification of input and output devices

E-stop Rope:

- Qty: 1
- PN: RP-RM83F
- https://www.bannerengineering.com/sg/en/products/part.82128.html

E-stop button:

- Qty: 1
- PN: SSA-EB1PLXR-12ECQ8
- https://www.bannerengineering.com/us/en/products/part.25305.html

Illuminated push buttons:

- Qty: 2
- PN: K50LYXXPPB2Q
- https://www.bannerengineering.com/us/en/products/part.32021.html

Sorter:

- Qty: 1
- PN: LX3EQ
- https://www.bannerengineering.com/us/en/products/part.02661.html

Proximity sensor:

- Qty: 4
- PN: TTR1219RPSD7T
- https://www.bannerengineering.com/us/en/products/part.812487.html

Steerable wheel sorter:

- Qty: 3
- PN: DENEB-PU
- https://nidoworld.com/automation/sortation-systems/popup-sorter/

Multi-color Indicator light:

- Qty: 4
- PN: K50LGRY2NQ
- https://www.bannerengineering.com/br/pt/products/part.75668.html

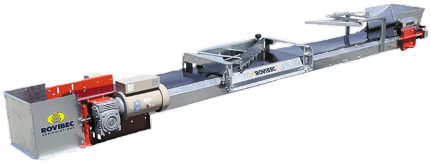
Indicator Tower Light:

- Qty: 3
- PN: TL70GYRQ
- https://www.bannerengineering.com/us/en/products/part.92336.html

Feeder:

- Qty: 1

- PN: CC718
- https://rovibecagrisolutions.com/en/produit/feeder-conveyor



Automated Gravity Roller with breaks:

- Qty: 3
- PN: QTY 10
- https://www.materialshandling.com.au/products/gravity-roller-conveyors/

7 Logic of the system

Find the code for the simulation system in the next few pages.

The next 6 rungs define auxiliary variable used in the remainer of the program. They serve the purpose of not repeating checks.

SW_center is the negation of 3:Pt05. This is necessary because no data point is true for the center position and false for both left and right.

0
SW4_aux1 <Local:3:I.Pt05.Data> / SW4_center

A_full indicates that no more products can go through line A. It is just a semantic name that also helps reduce code repetition.

1
A_in_collection_point.DN
A_total.DN
A_full

C_full indicates that no more products can go through line C. It is just a semantic name that also helps reduce code repetition.

2
C_in_collection_point.DN
C_total.DN
C_full

operate_SOL_A indicates that SOLA is activated. The ONS prevents feeding line A multiple times.

3
SW4_left <Local:3:I.Pt01.Data> PB3_NO <Local:1:I.Pt07.Data> conveyor.DN operate_SOL_A

operate_SOL_B indicates that SOLB is activated. The ONS prevents feeding line B multiple times.

4
SW4_center PB3_NO <Local:1:I.Pt07.Data> conveyor.DN operate_SOL_B

operate_SOL_C indicates that SOLC is activated. The ONS prevents feeding line C multiple times.

5
SW4_right <Local:3:I.Pt00.Data> PB3_NO <Local:1:I.Pt07.Data> conveyor.DN operate_SOL_C

The code that de facto operates the system starts at this point.

Controlls the feeder system. It does not operate when the system is running and does not allow the operator to overfill the feeder. There are two counters for products because sometimes it is needed to check wether the feeder is empty and others check wether it is full.

6
PB1_NO <Local:1:I.Pt03.Data> system_on / shutting_off / feeder_limit.DN / fill_feeder_ons ONS

CTU
Counter feeder_not_empty
Preset 1
Accum 0
CU
DN

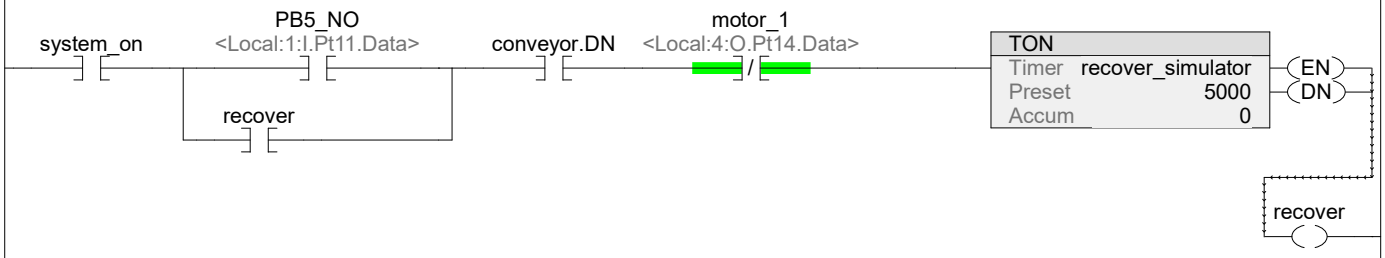
CTU
Counter feeder_limit
Preset 12
Accum 0
CU
DN

System starter rung. System will only power on if both ESTOPS are reset and the feeder is full.

7
PB4_NO <Local:1:I.Pt09.Data> / PB5_NO <Local:1:I.Pt11.Data> feeder_limit.DN e_stop_1 <Local:1:I.Pt01.Data> / e_stop_2 <Local:2:I.Pt01.Data> / system_on
system_on

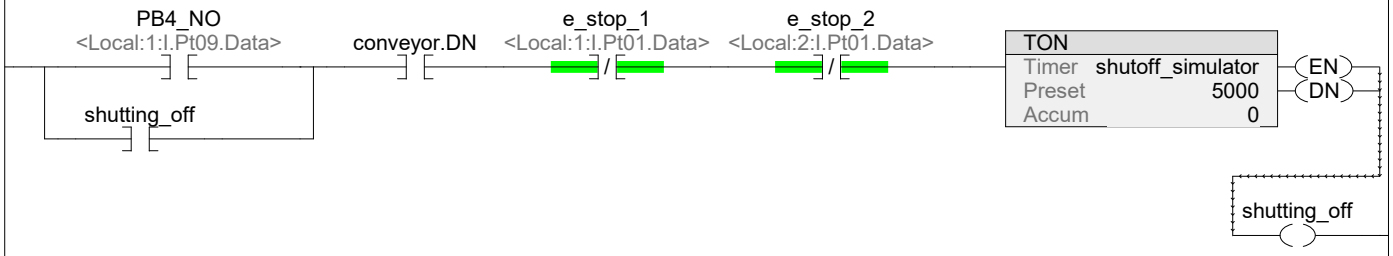
When starting off the system, if problems were detected, turn on a recover module.
In this case, we detect problems by verifying if there were items left in the conveyor. Note items might be on the conveyor during startup due to a previous ESTOP or a power outage.

8



When shutting off the system (normally, instead of using a ESTOP) it will run a module to make sure the system ends is a safe state.
In the simulation, it would only happen if there items in the conveyor, so that is the only check necessary before shutting off.

9



The following two rungs simullate personnel collecting the producs on A and C's collection points.

A is reset by flipping the momentary switch (SW3) to the left.

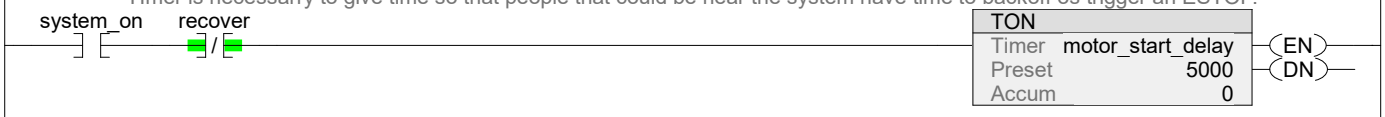
10

B is reset by flipping the momentary switch (SW3) to the right.

11

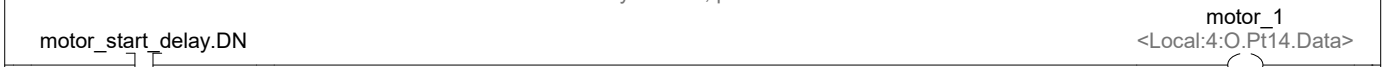
motor_1 indicates the main routine of the system is running.
Starts timer when the recover procedure is done and stops when system starts shutting off.
TImer is necessarry to give time so that people that could be near the system have time to backoff os trigger an ESTOP.

12



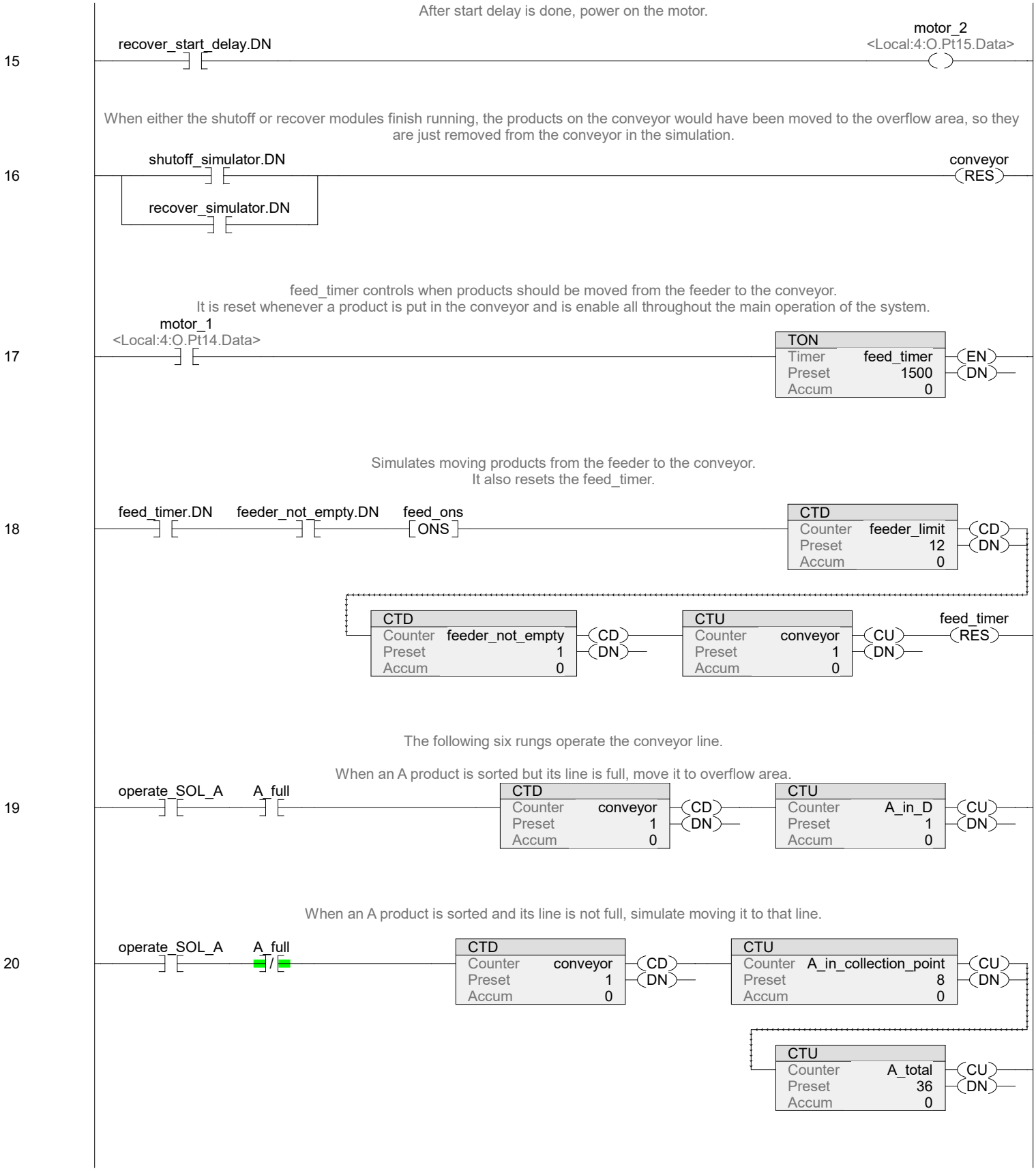
After start delay is done, power on the motor.

13



motor_2 indicates the recovery module is running.
In the case of this program, since both shutoff and recover consist of moving the products in the conveyor to an overflow area, they will trigger the same module, so the same motor.
TImer is necessary in both cases, because it may run on startup (like motor 1), but also it may run on diffent modules so making a delay trigerring shutoff seems adequate for the simulation.

14


After start delay is done, power on the motor.
recover_start_delay.DN
motor_2
<Local:4:O.Pt15.Data>
15
When either the shutoff or recover modules finish running, the products on the conveyor would have been moved to the overflow area, so they are just removed from the conveyor in the simulation.
shutoff_simulator.DN
recover_simulator.DN
conveyor
RES
16
feed_timer controls when products should be moved from the feeder to the conveyor.
It is reset whenever a product is put in the conveyor and is enable all throughout the main operation of the system.
motor_1
<Local:4:O.Pt14.Data>
TON
Timer feed_timer
Preset 1500
Accum 0
EN
DN
17
Simulates moving products from the feeder to the conveyor.
It also resets the feed_timer.
feed_timer.DN
feeder_not_empty.DN
feed_ons
ONS
CTD
Counter feeder_limit
Preset 12
Accum 0
CD
DN
CTD
Counter feeder_not_empty
Preset 1
Accum 0
CTU
Counter conveyor
Preset 1
Accum 0
feed_timer
RES
18
The following six rungs operate the conveyor line.
When an A product is sorted but its line is full, move it to overflow area.
operate_SOL_A
A_full
CTD
Counter conveyor
Preset 1
Accum 0
CTU
Counter A_in_D
Preset 1
Accum 0
19
When an A product is sorted and its line is not full, simulate moving it to that line.
operate_SOL_A
A_full
CTD
Counter conveyor
Preset 1
Accum 0
CTU
Counter A_in_collection_point
Preset 8
Accum 0
CTU
Counter A_total
Preset 36
Accum 0
20

When a B product is sorted but its line is full, move it to overflow area.

21 operate_SOL_B B_total.DN

CTD: Counter conveyor, Preset 1, Accum 0 — CD, DN

CTU: Counter B_in_D, Preset 1, Accum 0 — CU, DN

When a B product is sorted and its line is not full, simulate moving it to that line.

22 operate_SOL_B B_total.DN /

CTD: Counter conveyor, Preset 1, Accum 0 — CD, DN

CTU: Counter B_total, Preset 24, Accum 0 — CU, DN

When an C product is sorted but its line is full, move it to overflow area.

23 operate_SOL_C C_full

CTD: Counter conveyor, Preset 1, Accum 0 — CD, DN

CTU: Counter C_in_D, Preset 1, Accum 0 — CU, DN

When a C product is sorted and its line is not full, simulate moving it to that line.

24 operate_SOL_C C_full /

CTD: Counter conveyor, Preset 1, Accum 0 — CD, DN

CTU: Counter C_in_collection_point, Preset 6, Accum 0 — CU, DN

CTU: Counter C_total, Preset 18, Accum 0 — CU, DN

Starting from this point the rungs will only concern with displaying visual cues to allow operators visualizing the simulation. There is no added manipalition of the conveyor.

The show feeder problem is useful during the simulation as it coordinate an indicator light for problems related to the state of the feeder.

25 system_on / PB1_NO <Local:1:I.Pt03.Data> feeder_limit.DN

PB5_NO <Local:1:I.Pt11.Data> feeder_limit.DN /

system_on feeder_not_empty.DN /

TOF: Timer show_feeder_problem, Preset 1000, Accum 0 — EN, DN

Powers on the red light when the state of the feeder might impact the behavior of the simulation.
Timer is used because it improves the visualization (which is the main part of the simulation).

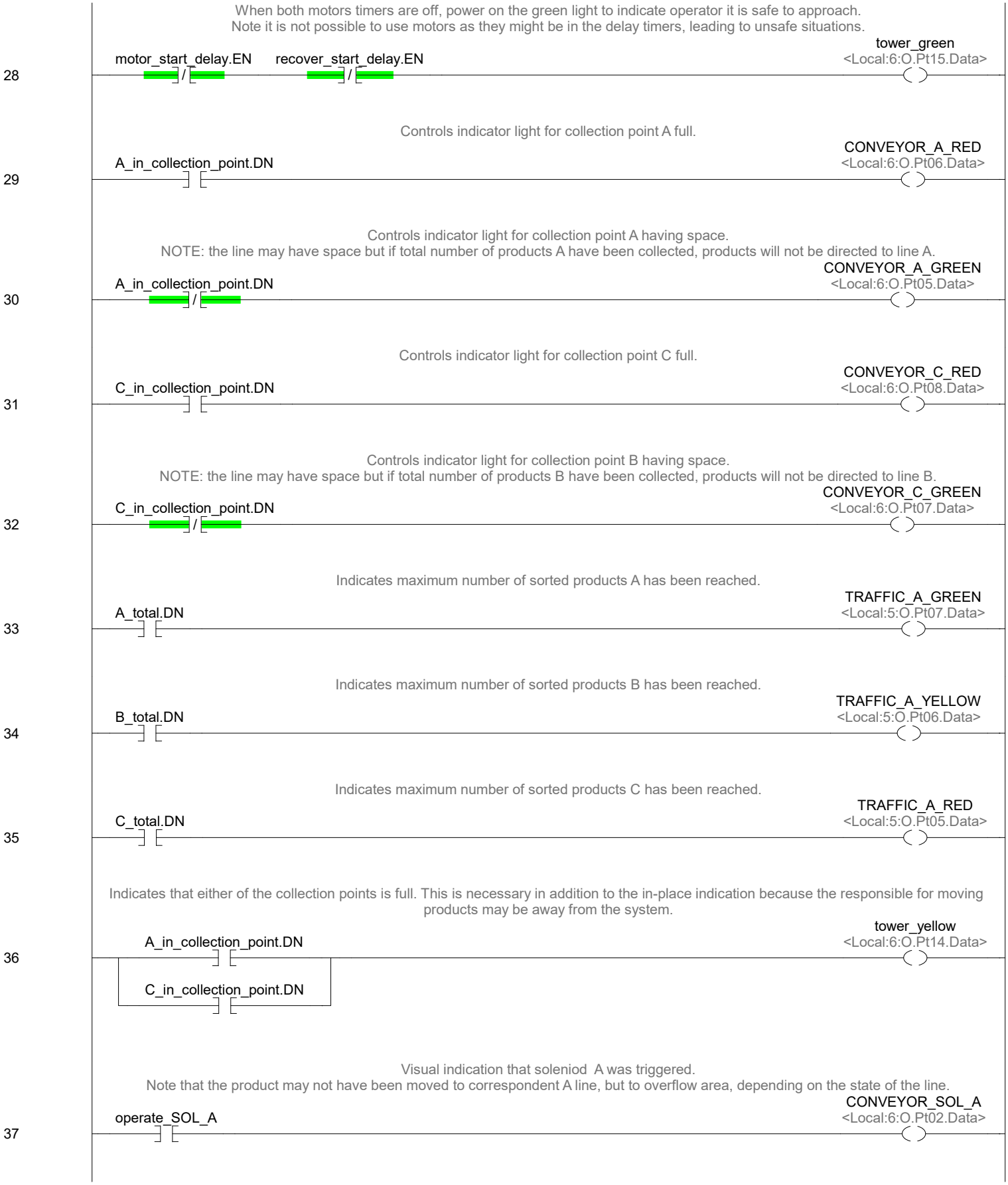
26 show_feeder_problem.DN — feeder_red <Local:6:O.Pt01.Data>

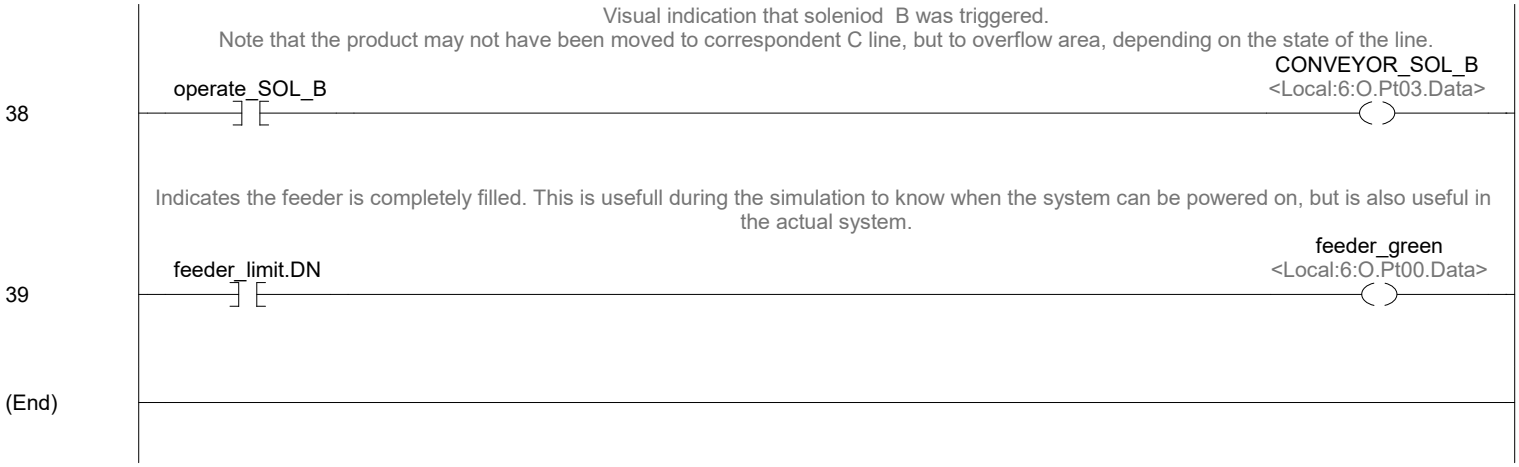
Whenever a motor is running (regardless of running the main routine or recovery module) keeps the red light on.
Note that the light is powerd on before the motor starts as it needs to warn possible operators before powering on the motor.

27 motor_start_delay.EN

recover_start_delay.EN

tower_red <Local:6:O.Pt13.Data>


When both motors timers are off, power on the green light to indicate operator it is safe to approach.
Note it is not possible to use motors as they might be in the delay timers, leading to unsafe situations.
motor_start_delay.EN
recover_start_delay.EN
tower_green
<Local:6:O.Pt15.Data>
28
Controls indicator light for collection point A full.
A_in_collection_point.DN
CONVEYOR_A_RED
<Local:6:O.Pt06.Data>
29
Controls indicator light for collection point A having space.
NOTE: the line may have space but if total number of products A have been collected, products will not be directed to line A.
A_in_collection_point.DN
CONVEYOR_A_GREEN
<Local:6:O.Pt05.Data>
30
Controls indicator light for collection point C full.
C_in_collection_point.DN
CONVEYOR_C_RED
<Local:6:O.Pt08.Data>
31
Controls indicator light for collection point B having space.
NOTE: the line may have space but if total number of products B have been collected, products will not be directed to line B.
C_in_collection_point.DN
CONVEYOR_C_GREEN
<Local:6:O.Pt07.Data>
32
Indicates maximum number of sorted products A has been reached.
A_total.DN
TRAFFIC_A_GREEN
<Local:5:O.Pt07.Data>
33
Indicates maximum number of sorted products B has been reached.
B_total.DN
TRAFFIC_A_YELLOW
<Local:5:O.Pt06.Data>
34
Indicates maximum number of sorted products C has been reached.
C_total.DN
TRAFFIC_A_RED
<Local:5:O.Pt05.Data>
35
Indicates that either of the collection points is full. This is necessary in addition to the in-place indication because the responsible for moving products may be away from the system.
A_in_collection_point.DN
C_in_collection_point.DN
tower_yellow
<Local:6:O.Pt14.Data>
36
Visual indication that soleniod A was triggered.
Note that the product may not have been moved to correspondent A line, but to overflow area, depending on the state of the line.
operate_SOL_A
CONVEYOR_SOL_A
<Local:6:O.Pt02.Data>
37


Visual indication that soleniod B was triggered.
Note that the product may not have been moved to correspondent C line, but to overflow area, depending on the state of the line.
operate_SOL_B
CONVEYOR_SOL_B
<Local:6:O.Pt03.Data>
38
Indicates the feeder is completely filled. This is usefull during the simulation to know when the system can be powered on, but is also useful in the actual system.
feeder_limit.DN
feeder_green
<Local:6:O.Pt00.Data>
39
(End)

8 Startup Configuration

The first requirement for system startup is that emergency stops are reset to their original position. This is a common and necessary procedure.

The other two startup requirements are the feeder being full and the belts being empty. While the former is not a necessity, the second one is because the system relies on constant speed and timers to calculate the position of products. Once there is a reset, like with a power loss, necessary information is lost and there is no way to recover the position of the product. If not emptied the system would end up moving products to wrong lines, to state of the possibilities.

9 Potential System Issues

The system as implemented in the simulation and described section 2 will resist some errors resulting prom emergency stops and power loss. The current state of the system, however, is not safe for all scenarios, including resistance to personnel mistakes, power loss and package jams. This section brings some of them into light.

The conveyor is intended to operate in accessible places as it does not present any major or fatal injury risk. Adding physical barriers might reduce the frequency of injuries but that happens in exchange of productivity gains. It is a decision that may vary depending on the specification of sizes of products or speed of conveyors. However, when looking at production environments, which typically do not have barriers, a decision to not use them was made. The main risk is when a product is redirected by the belt diverter, a PLC or device errors happens, and the product ends up falling, maybe in some employee.

However, the current version of the system does not take into account misuse of the equipment. As from the PLC program, the collection points cannot be accessed when the machine is on due to the risk of having other products coming down from the conveyor in high speed. However, since there are no physical blocks that becomes a risk.

A second issue, that was brought to light since section two is that not all problems related to power loss were handled. If an item is high speed is coming down a line and the automated gravity rollers used do not have power to break their movement, the product may reach the end of the line, falling in the ground and breaking or maybe even hitting someone.

The third problem is that the system does not handle hardware issues, by using additional sensors for example, nor does it fail safe. When a jam happens, for example, it goes unnoticed by the system and this can lead to multiple issues, from braking products, to breaking sensors and even hurting people. While there were attempts to solve safety issues, it is very clear they persist.