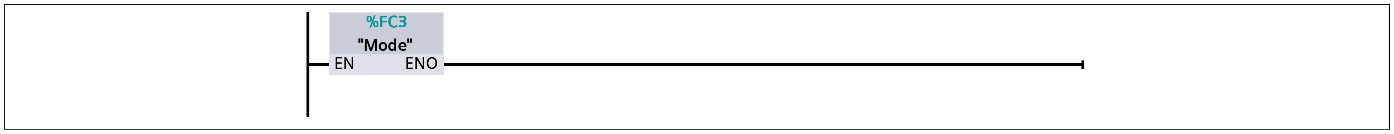
WS-S7-1200-Sort-Weight-HMI-V6

Program blocks

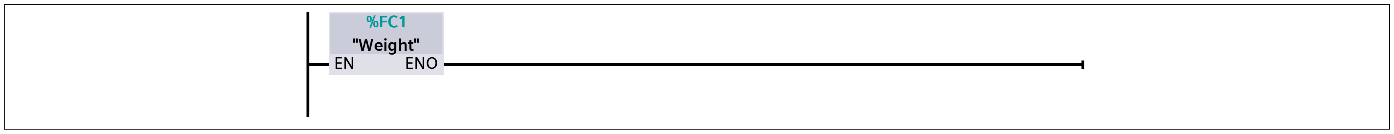
Main [OB1]

Main Properties					
General					
Name	Main	**Number**	1	**Type**	OB
Language	LAD	**Numbering**	Automatic		
Information					
Title	"Main Program"	**Author**		**Comment**	OB1 is the traffic control block. In this case, the main function is to call the program block used in the program. Calls to block can be controlled to increase processing speed or switch between functionality. For example, two palletizer blocks can be created which change the functionality of the palletizer depending on which block is called. Care must be taken when using the same output multiple times as it can create confusion for Trades, Engineers and Technicians troubleshooting the program.
Family		**Version**	0.1	**User-defined ID**	

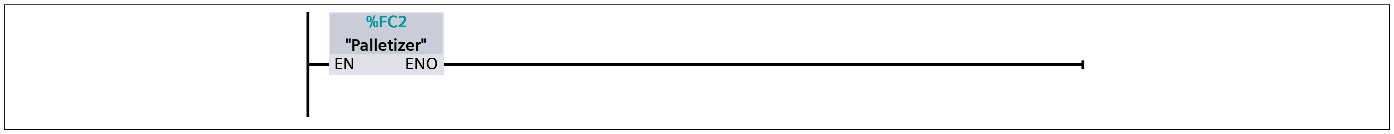
Network 1: Call FC that selects Manual or Automatic Modes.



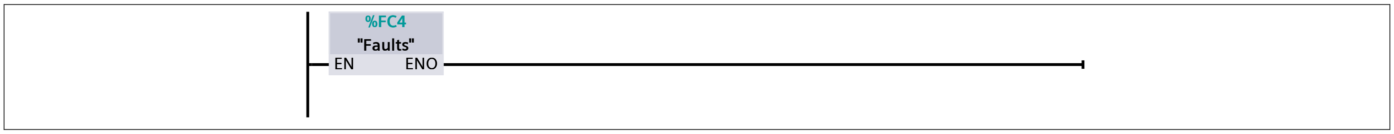
Network 2: Call FC that controls the Weight Scale function.



Network 3: Call FC that controls the Palletizer.

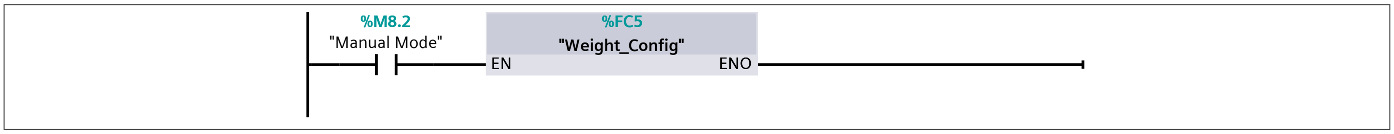


Network 4: Call the Fault FC

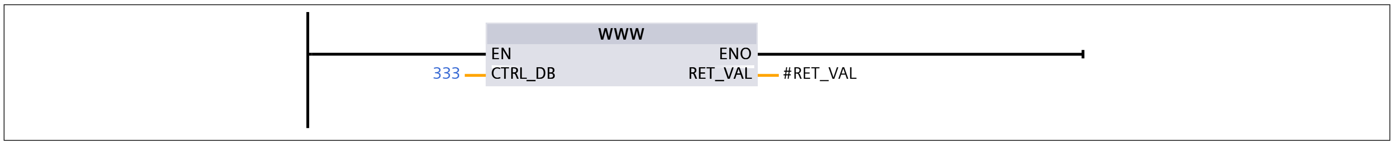


Network 5: Call the Weight Configuration Routine

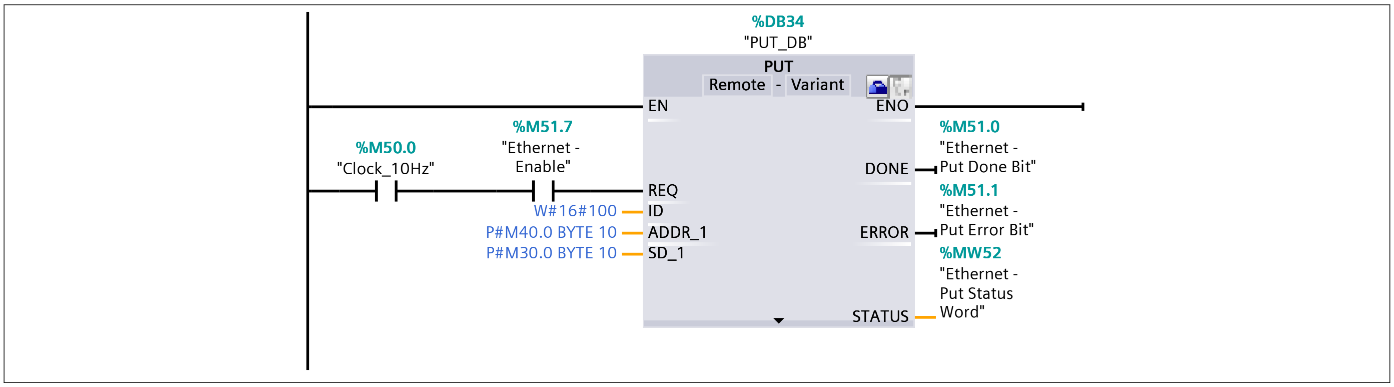
Call weight configuration and calculation program. This is only required in Manual Mode.
For the purpose of this code, FC5 could be called repeatedly without adversely affecting the scan time or communication but it is good practice to only use certain routines when required. This is especially true for routines that are very CPU intensive such as data handling, storage, retrieval or non-critical communications.



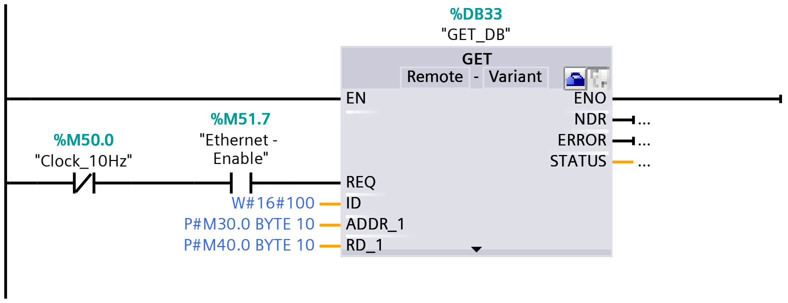
Network 6: Enable Web Service



Network 7: Send Data to PLC_2



Network 8: Pull Data From PLC_2


%DB33
"GET_DB"
GET
Remote - Variant
EN
ENO
NDR ...
ERROR ...
STATUS ...
%M51.7
"Ethernet -
Enable"
%M50.0
"Clock_10Hz"
REQ
W#16#100
ID
P#M30.0 BYTE 10
ADDR_1
P#M40.0 BYTE 10
RD_1

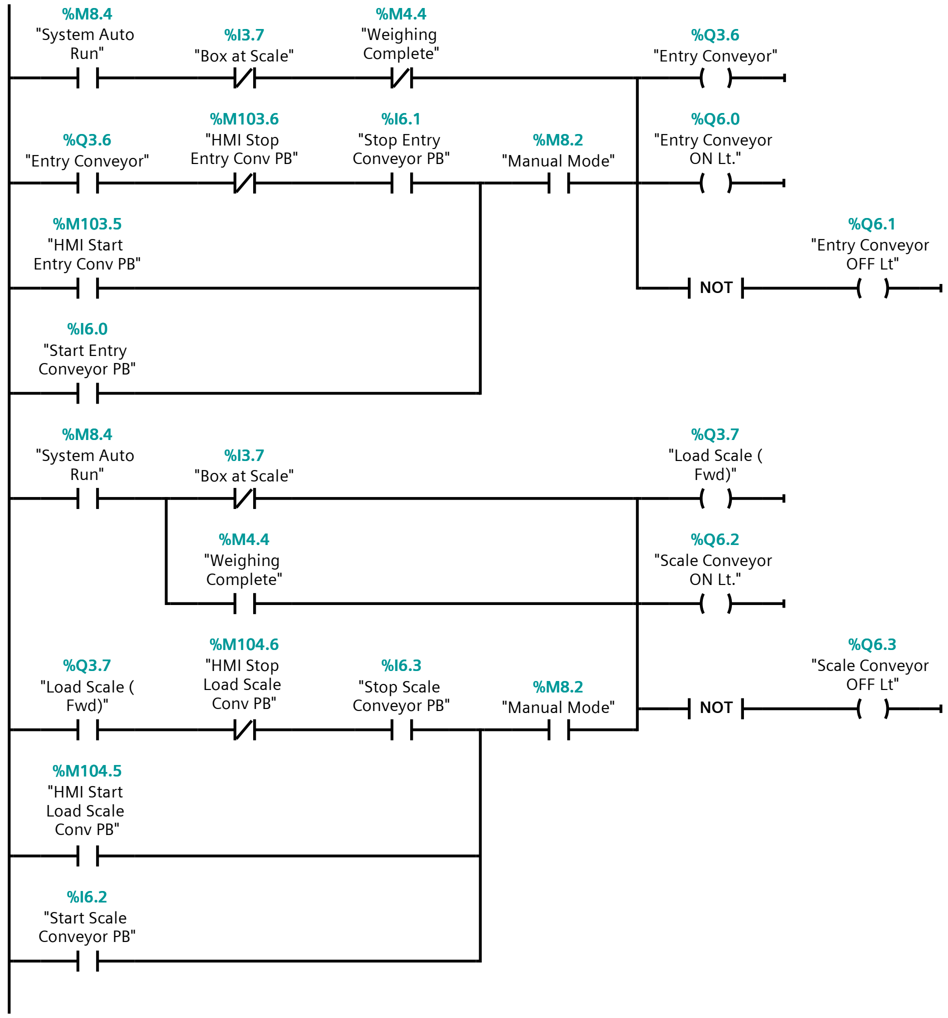
Program blocks

Weight [FC1]

Weight Properties					
General					
Name	Weight	**Number**	1	**Type**	FC
Language	LAD	**Numbering**	Automatic		
Information					
Title	Sort by Weight Station	**Author**	WayneSchaefer	**Comment**	Station 1 is the Sort by Weight Station consisting of a supply conveyor, weighing station, diverter ball table and 3 exit conveyors to take the box in one of 3 directions.
Family	S71200	**Version**	0.1	**User-defined ID**	

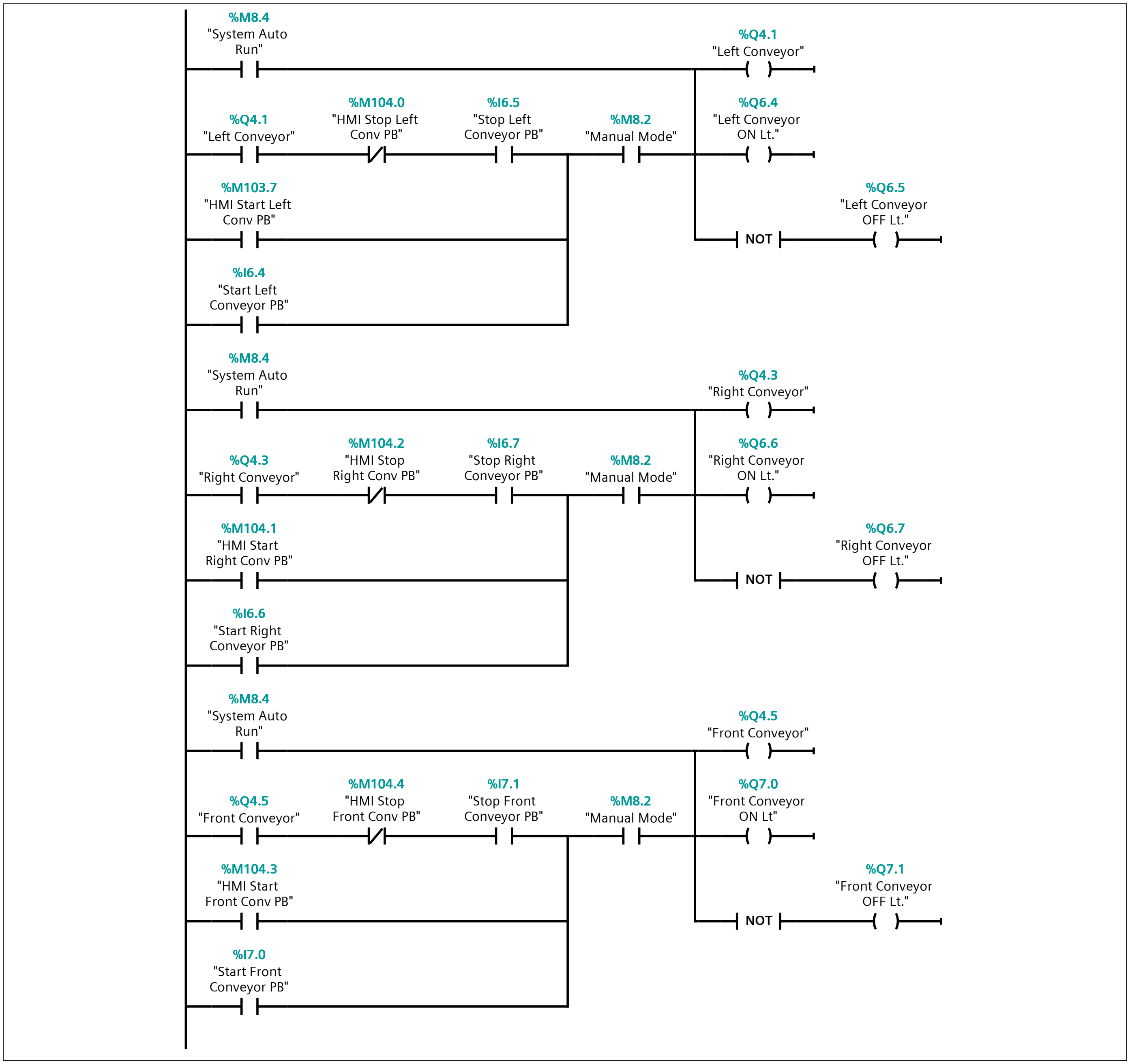
Network 1: Entry and Load (Scale) Conveyor Motor Control.

Conveyor Control. The Entry Conveyor and Load Scale Conveyor are on until a box arrives at the scale and makes the Box at Scale proximity switch. Once weighing is complete, the Load Scale is allowed to run to send the box to the diverter ball table. Weighing Complete is reset when the box reaches one of the 3 entry sensors at which time the Entry Conveyor is turned back on.



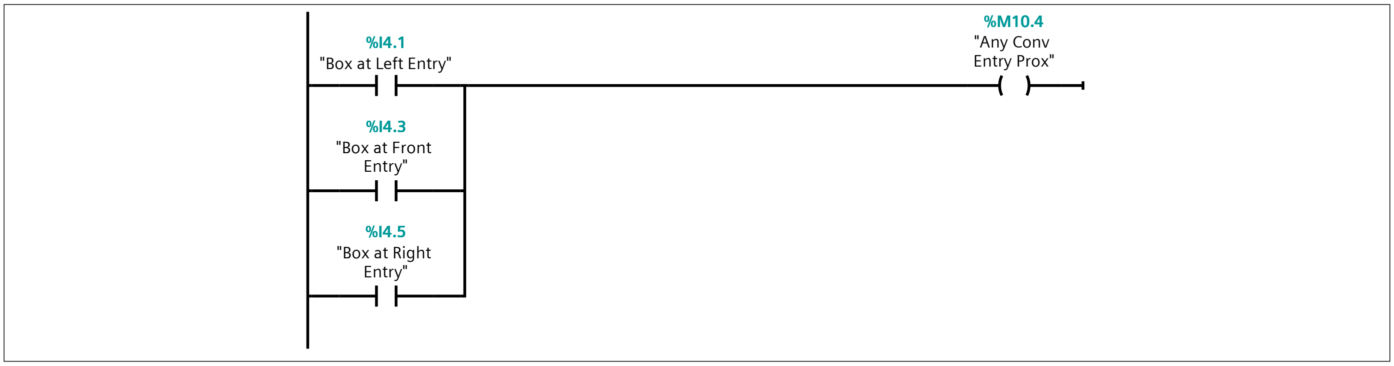
Network 2: Right, Left and Front Conveyor Motor Control.

Conveyor control. The Right, Left and Front Conveyors turn on as soon as Auto Run is initiated. From a power consumption and wear and tare perspective, these conveyors would most likely on run when needed. Once a box exits any conveyor, it would be prudent to turn the conveyor off until the next box arrives. There is always a possibility the starter controlling the motor would have to be resized to handle the number of cycles required for on-off-on-off operation.



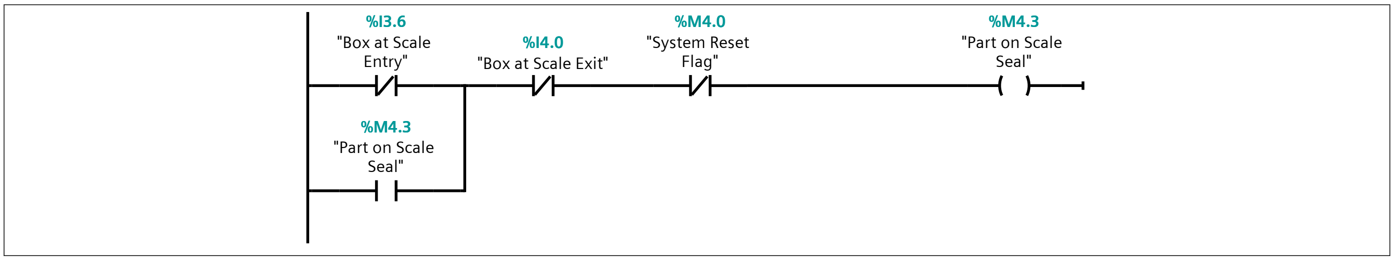
Network 3: Any Entry Prox Made

The Left, Right and Front Entry proximity switches are used to clear the weight scale logic seals.
Individual proximity switches could be used to only clear the seal for that particular direction. However, in the unlikely event a box is sent to the wrong conveyor, it would be much better to clear the seal but capture the fault in another part of the program.



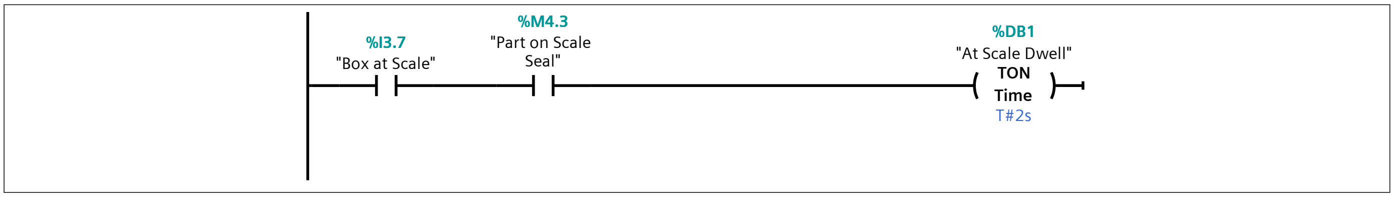
Network 4: At Scale Memory

Memory that a Box is on the Scale (box can be in transit between photo eyes) at this point.



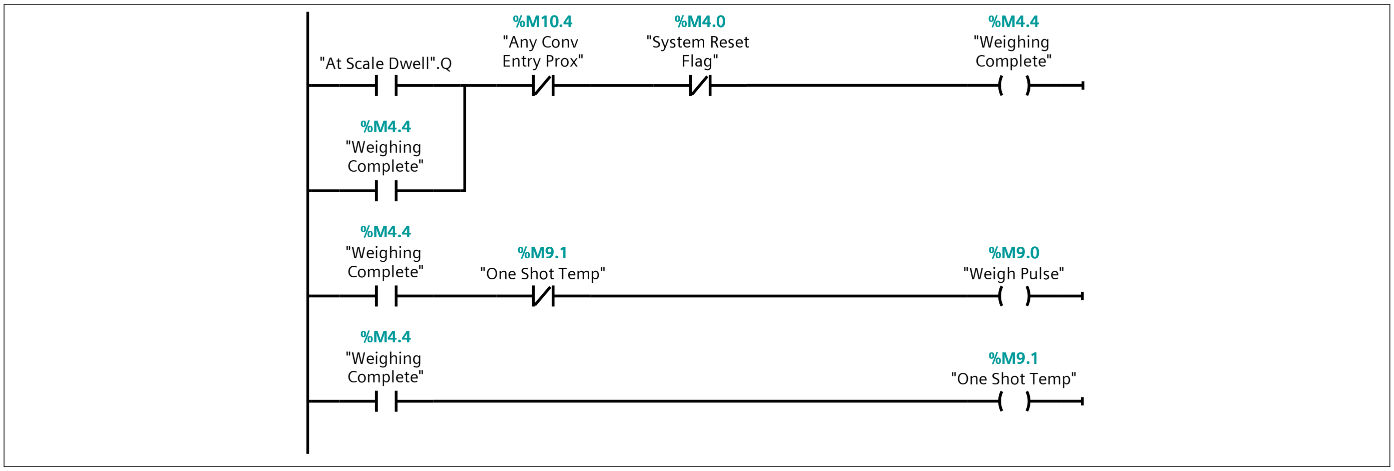
Network 5: Box Detected on Scale Dwell Timer.

Box at scale dwell. Part on Scale Seal is added to ensure multiple conditions are met before the box can be weighed. A box placed directly on the scale will not trigger a weigh cycle.



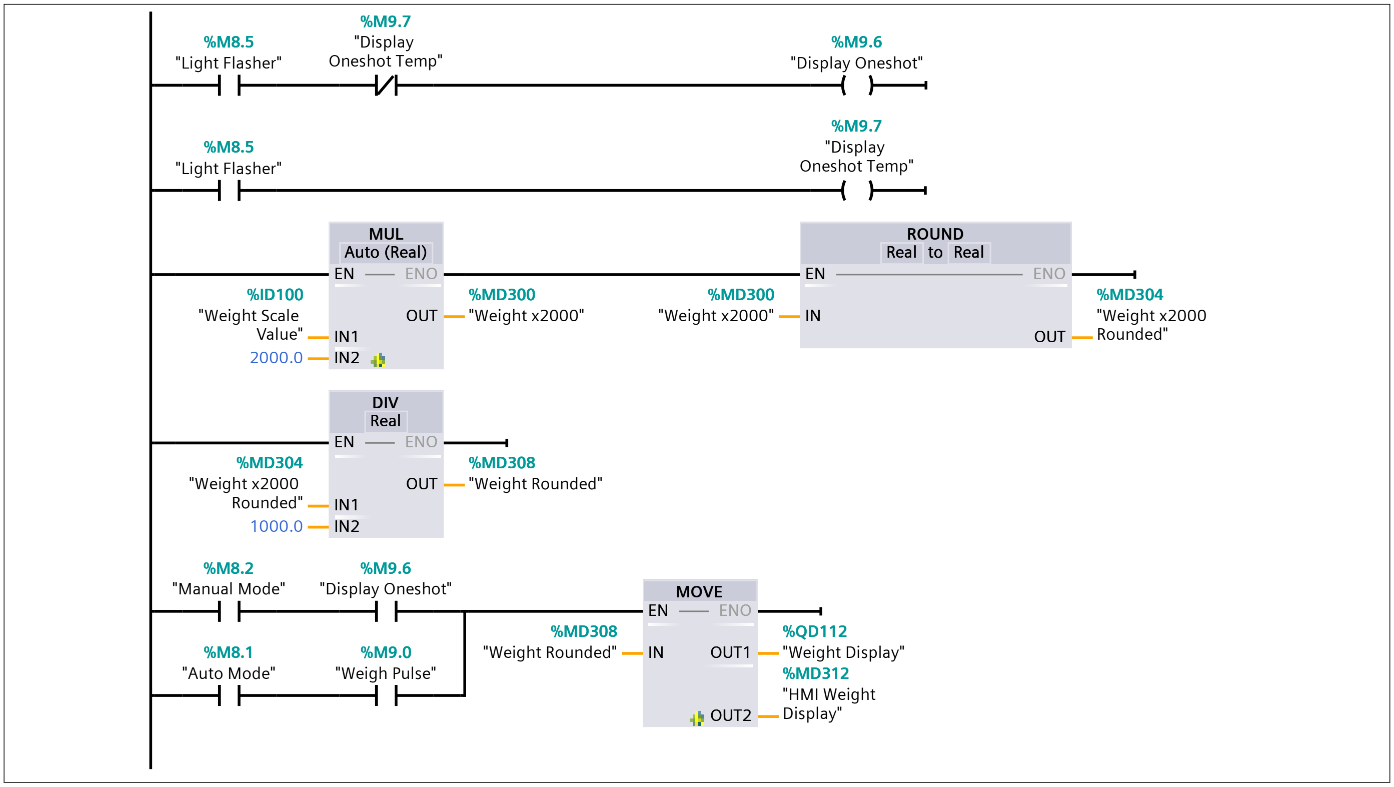
Network 6: Weighing Complete

Demonstration of a One-shot if not built into a ladder program as a specific function. This is a good example to demonstrate have a PLC Traditionally scans logic and updates (left to right, top to bottom). This will not work with some controllers, such as the Siemens Logo, which solve logic differently.



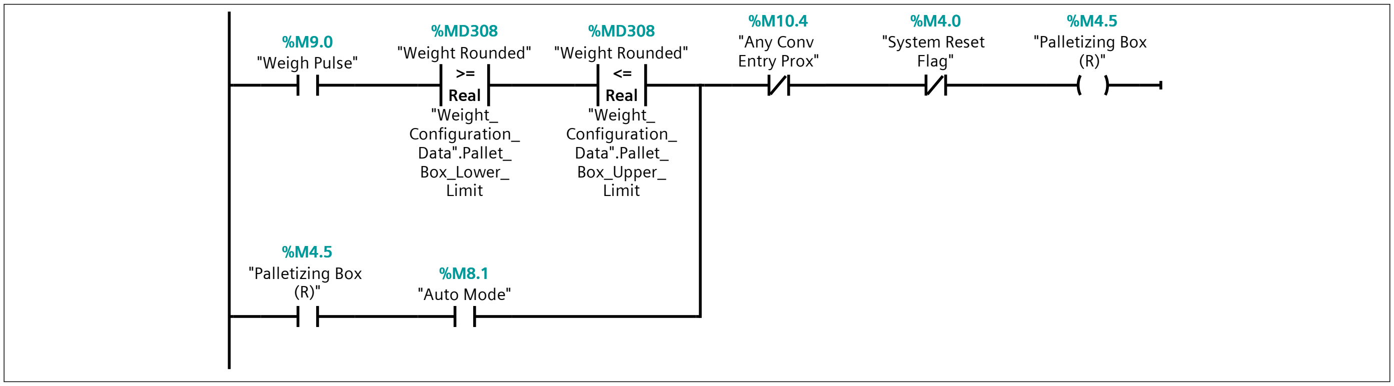
Network 7: Weight Calculation and Display

Using the light flasher bit to slow down the scale readout update in manual mode.
Display only updates once per box in automatic - easier to see what value the decision to transfer was based on.
Using the 20Kg max weight option which mean that 0-10 Volts must be scale to 0-20Kg.
The siemens rounding function block only works on rounding up to the nearest whole number.
For accuracy and scaling, this network, multiplied the voltage received by 2000, rounds and then divides by 1000.



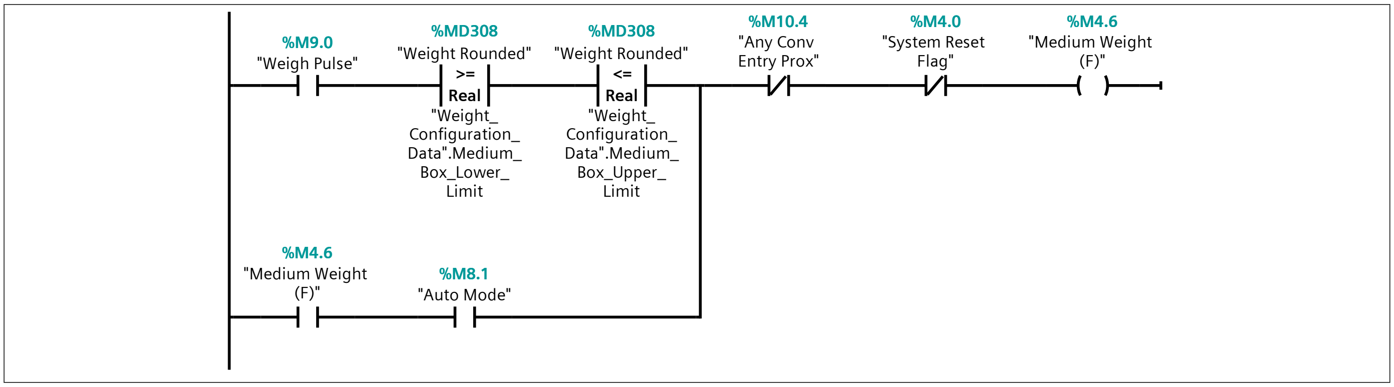
Network 8: Transfer Right Memory

Memory that a Palletizing Box on the Scale is detected.



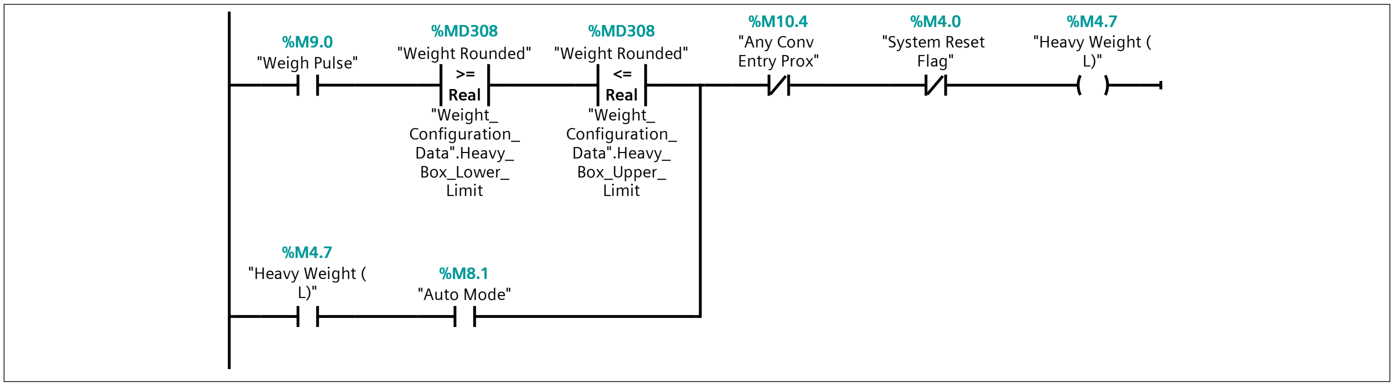
Network 9: Transfer Forward Memory

When a Medium Weight Box is detected, set Transfer Forward Memory.



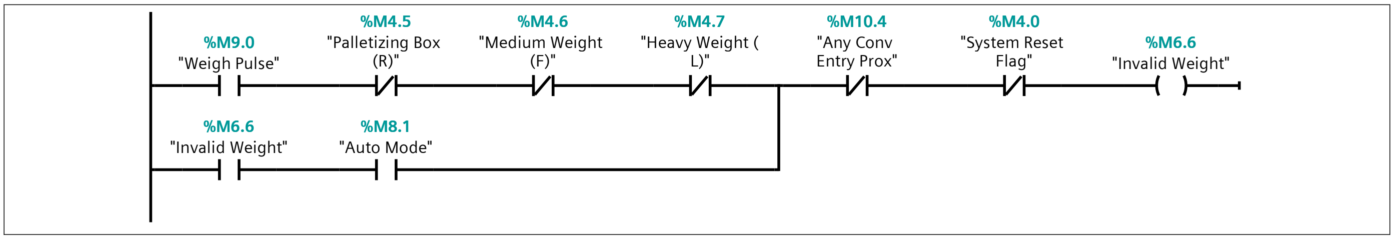
Network 10: Transfer Left Memory

When a Heavy Weight Box is detected, set Transfer Left Memory.



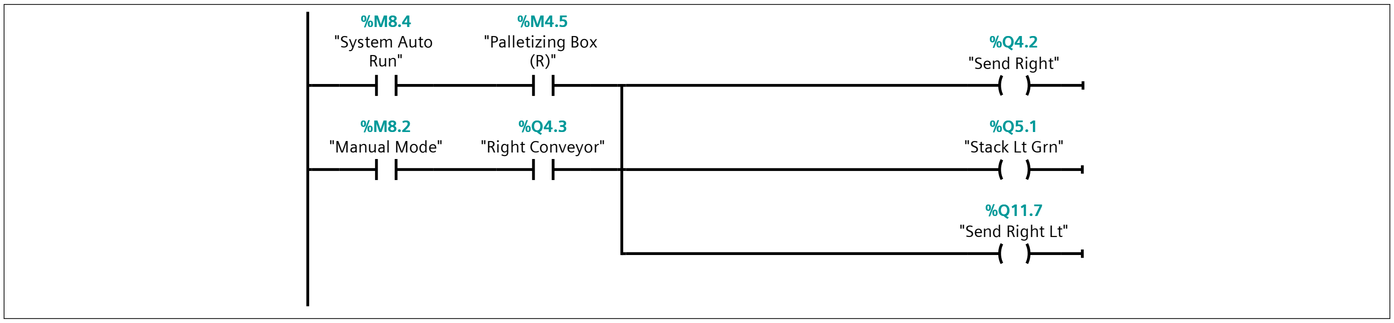
Network 11: Invalid Weight on Scale

If no valid weight is detected, set Invalid Weight Bit. Note that this rung must occur after the logic that calculate what box type is on the scale.



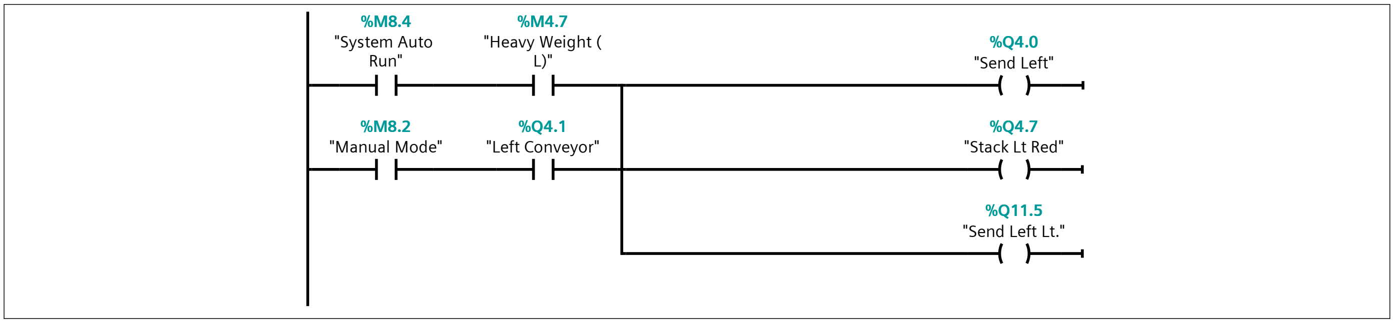
Network 12: Select Direction Right

Memory that a Palletizing (Light) Weight Box was detected.Send Right.



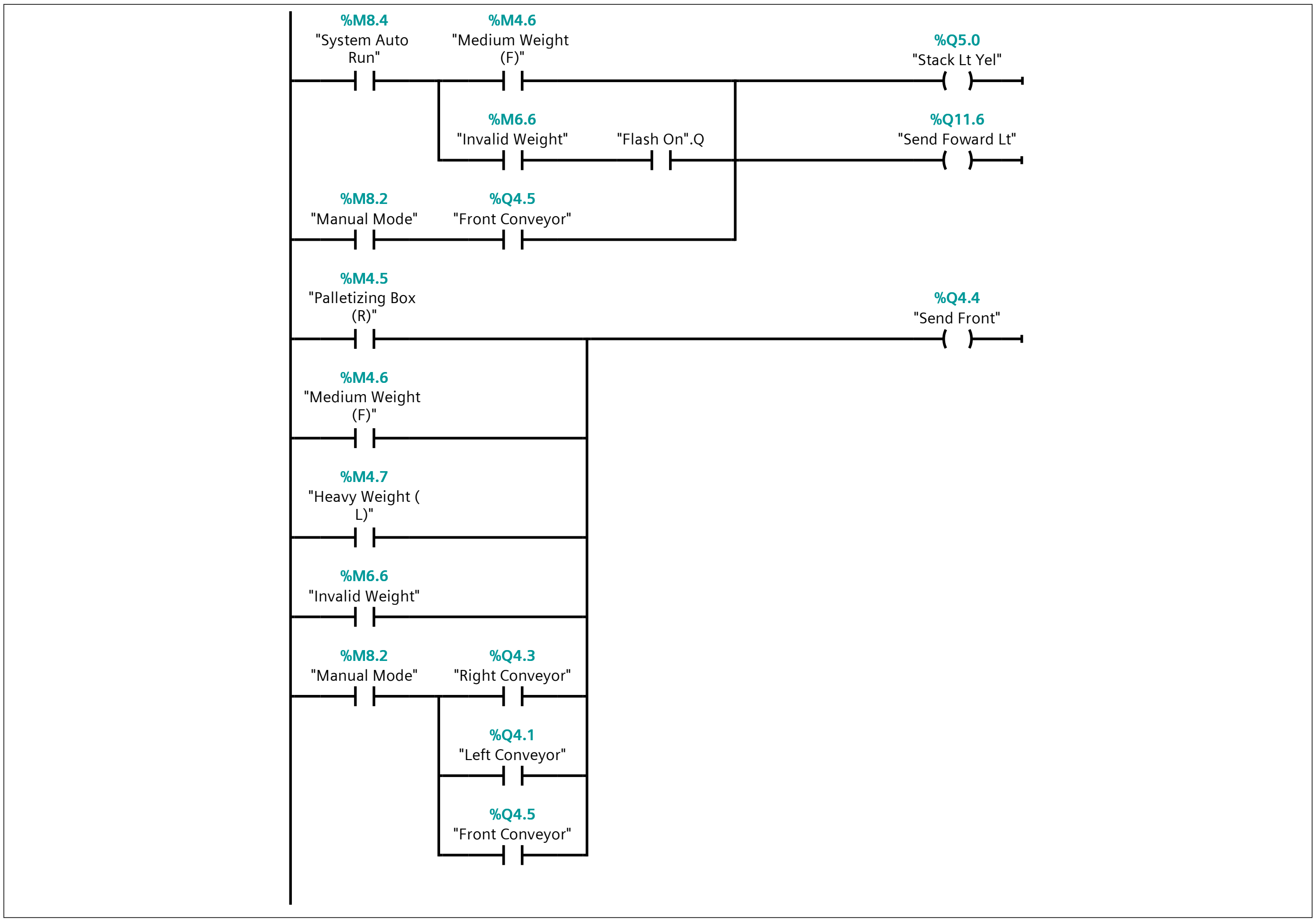
Network 13: Select Direction Left

Memory that a Heavy weight box was detected. Send Left.



Network 14: Select Direction Front

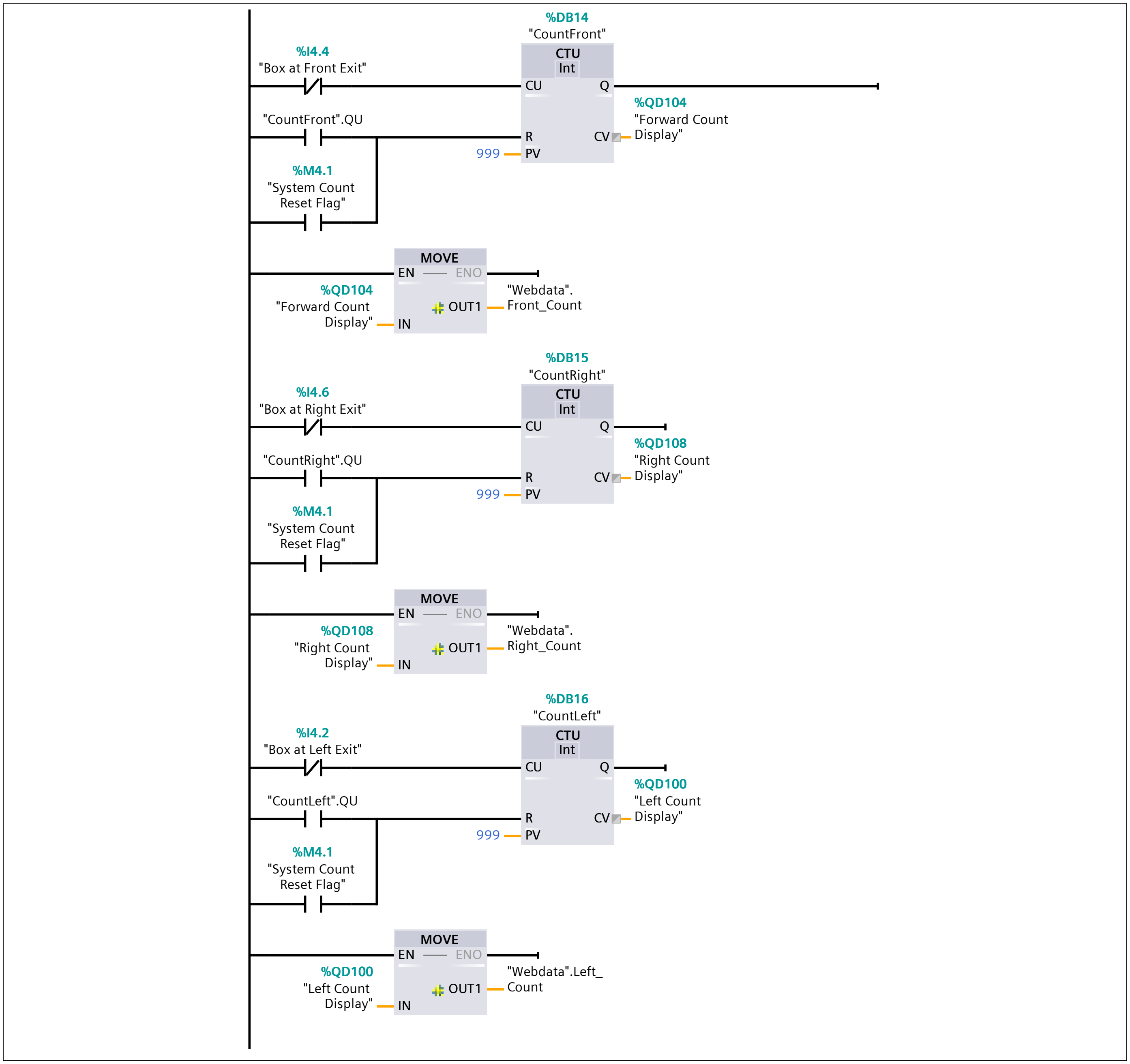
Memory that a Heavy Weight Box was detected. Send Front.
Currently invalid weight boxes get sent Front as well.
Note that the Front Direction Output is required to raise the diverter for Left and Right.



Network 15: Counters

Box counters are simply using the box passing a proximity to trigger the count. Based on the current logic, any disruption of the proximity field (whether the detection method is magnetic, capacitive or light based) a count will occur.

False counting can be mitigated by looking at other factors such as In-Transit memory that could be combined to create a more robust set of conditions.



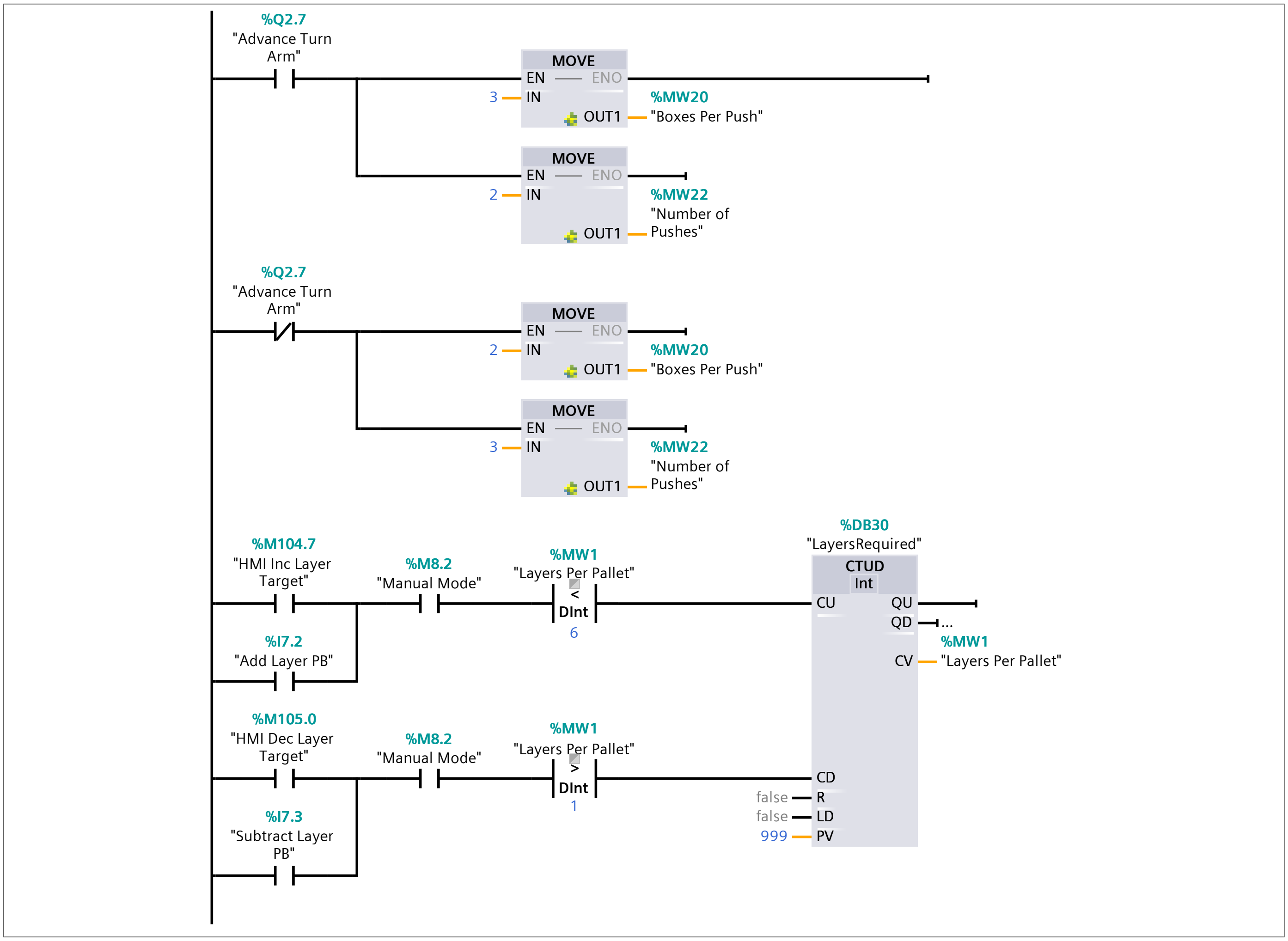
Program blocks

Palletizer [FC2]

Palletizer Properties					
General					
Name	Palletizer	**Number**	2	**Type**	FC
Language	LAD	**Numbering**	Automatic		
Information					
Title	Palletizer	**Author**	WayneSchaefer	**Comment**	
Family	S71200	**Version**	0.1	**User-defined ID**	

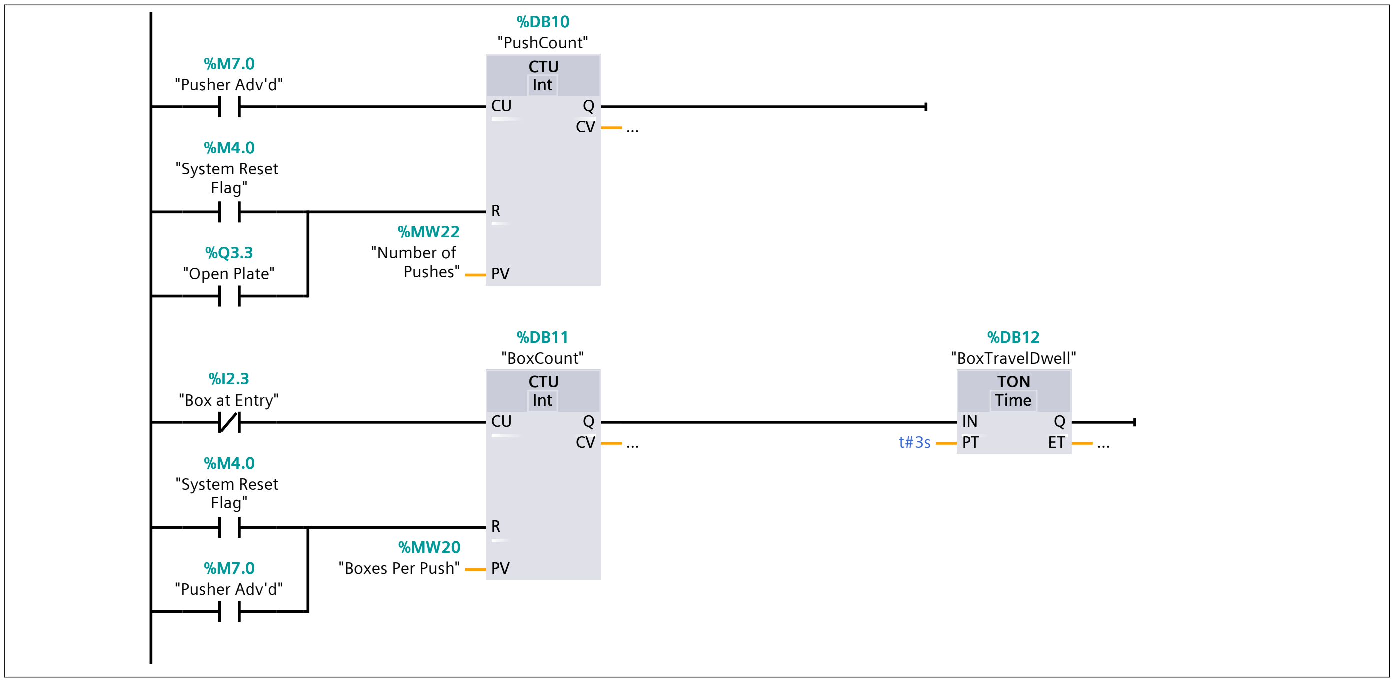
Network 1: Configuration

If the turn arm is not advanced, the boxes are collected such that 2 must stack up before pushing and 3 sets of 2 are required to complete a layer. If the boxes are turned, 2 sets of 3 boxes per layer are required. Number of layers is selected via the HMI screen using push buttons to increment or decrement the number of layers.



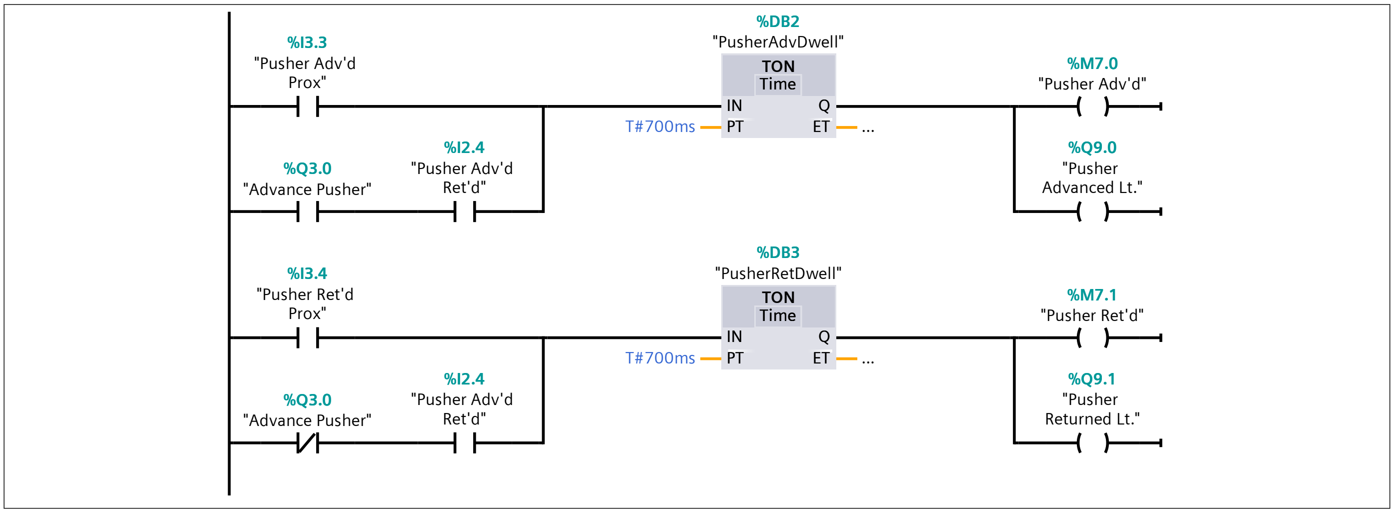
Network 2: Box Counts

Counting the number of boxes accumulated since the last push and how many pushes since the last layer was built. Since the boxes have two different orientations, the number of boxes per row and the number of pushes per layer alternate.



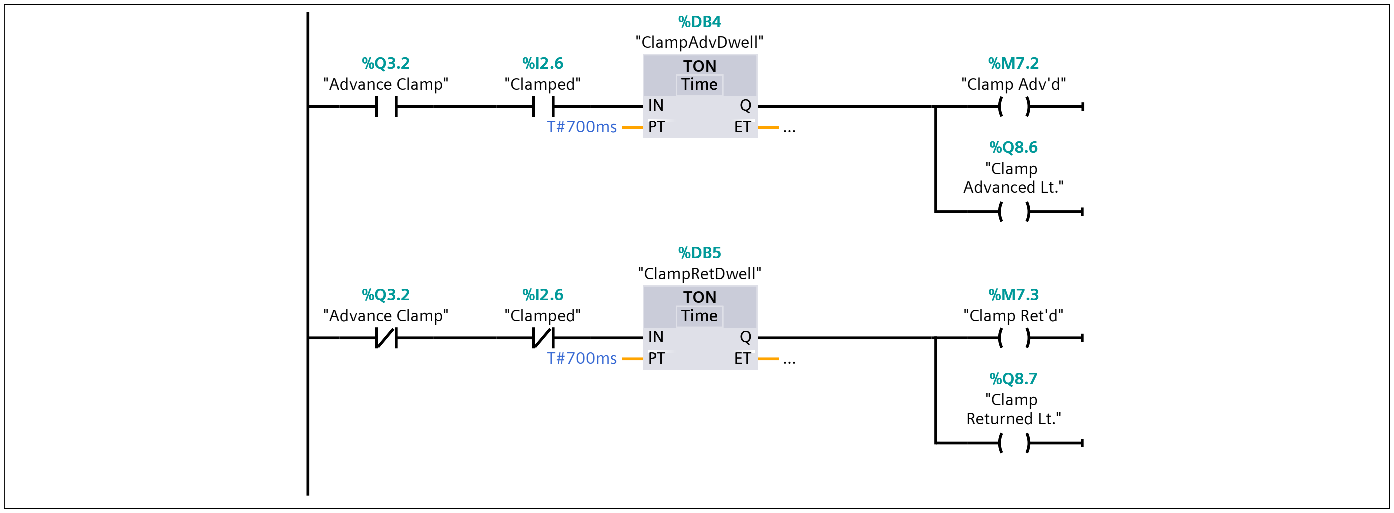
Network 3: Pusher Position

There are two methods to determine if the pusher is advanced or returned.
Option 1:
If the advance pusher output is ON and the pusher In-Position input is ON, the pusher is Adv'd
If the return pusher output is ON and the pusher In-Position input is ON, the pusher is Ret'd
The dwell is required since the Input and Output are briefly ON at the same time at the beginning of the movement.
Option 2:
Requires the addition of two proximity switches to the standard palletizer scene. One for the returned position and one for the advanced position.
Note:
It is always best to directly detect the position of a device. If that is not possible some logical deduction may be required.



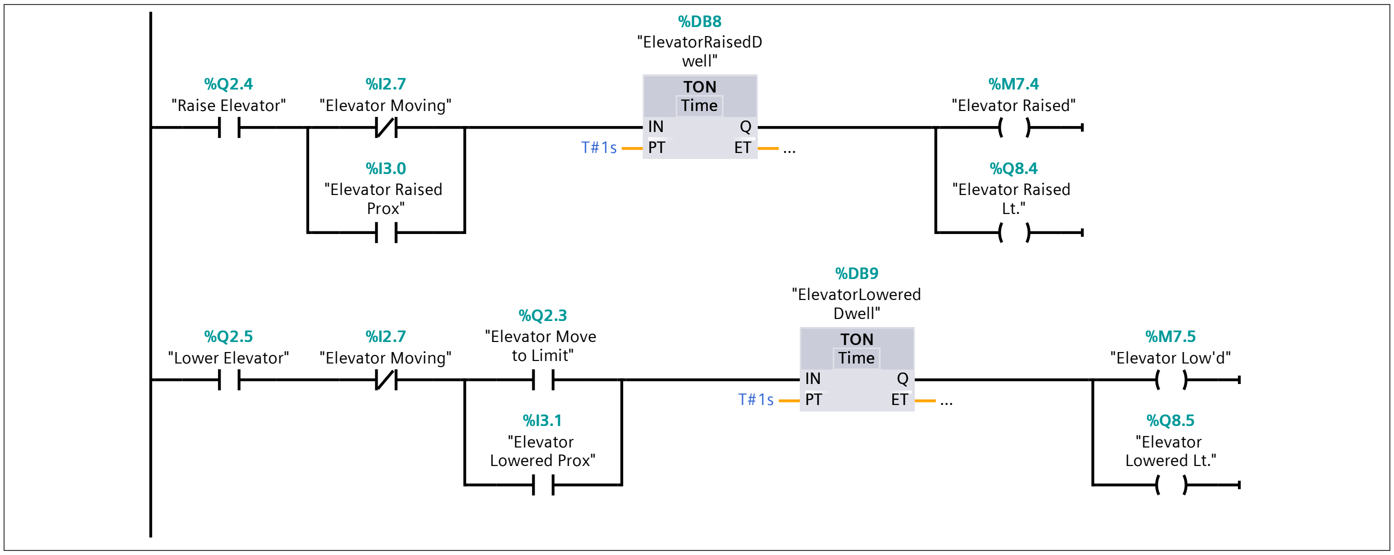
Network 4: Clamp Position

If the advance clamp output is ON and the clamp In-Position input is ON, the clamp is Adv'd
If the return clamp output is ON and the clamp In-Position input is ON, the clamp is Ret'd
The dwell is required since the Input and Output are briefly ON at the same time at the beginning of the movement.



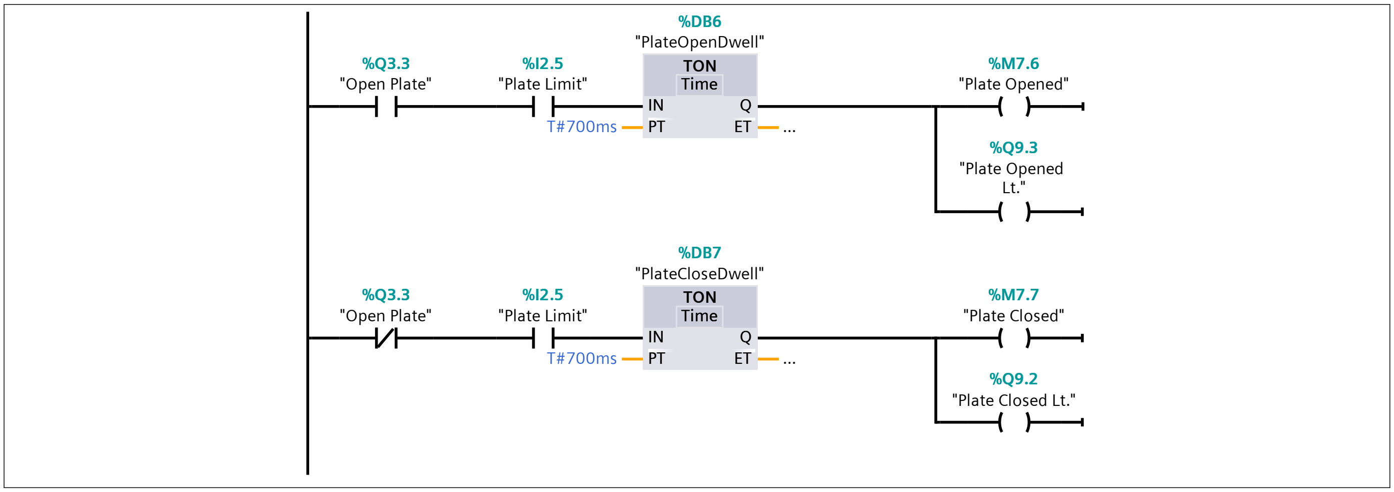
Network 5: Elevator Position

There are two methods to determine if the elevator is raised or lowered.
Option 1:
If the raise elevator output is ON and the elevator moving input is OFF, the elevator is raised
If the lower elevator output is ON and the elevator moving input is OFF, the elevator is lowered
The dwell is required since the Input and Output are briefly ON at the same time at the beginning of the movement.
Option 2:
Requires the addition of two proximity switches to the standard palletizer scene. One for the raised position and one for the lowered position.
Note:
It is always best to directly detect the position of a device. If that is not possible some logical deduction may be required.



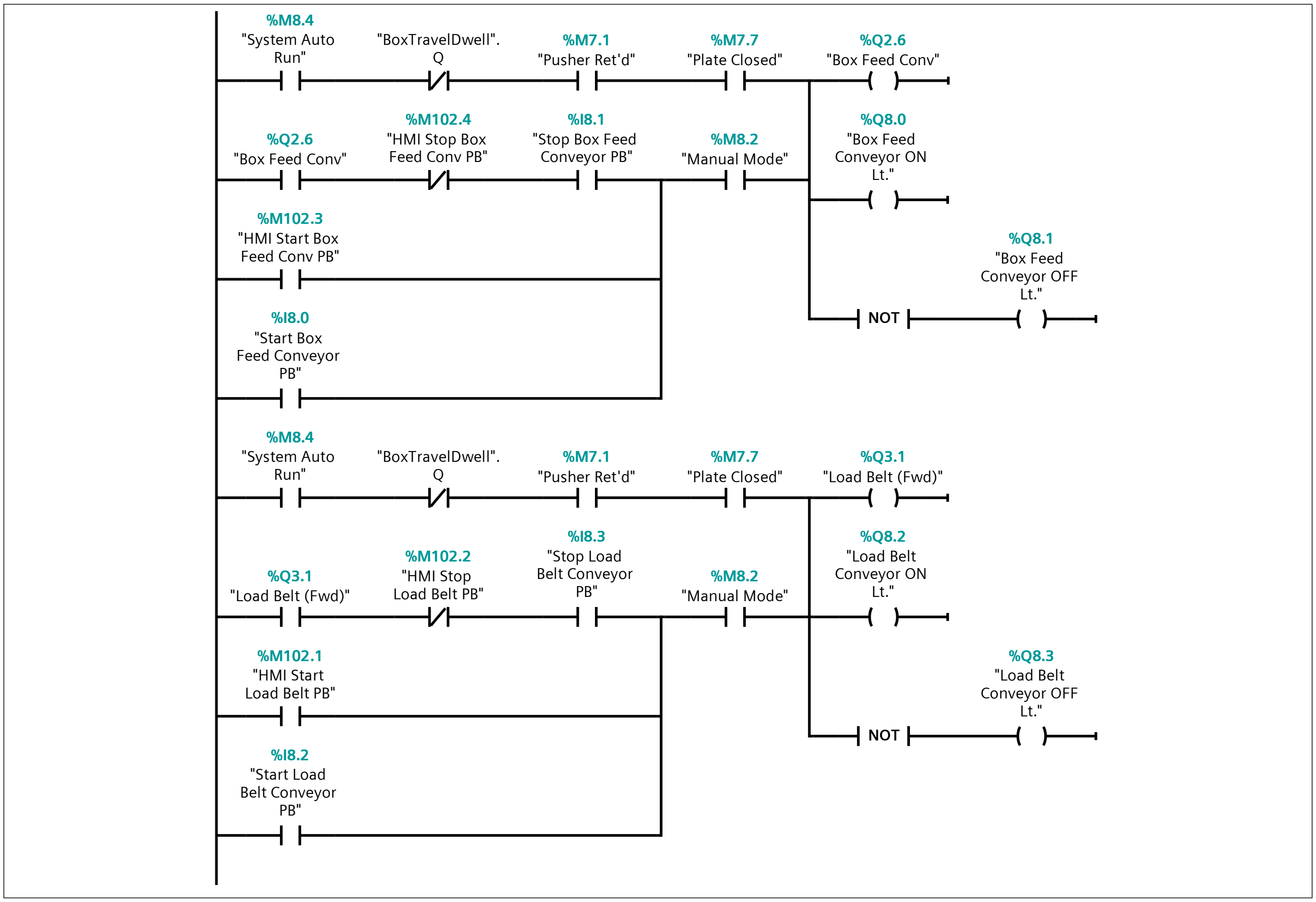
Network 6: Plate Position

If the open plate is ON and the plate limit input is ON, the plate is open.
If the open plate is OFF and the plate limit input is ON, the plate is closed.
The dwell is required since the Input and Output are briefly ON at the same time at the beginning of the movement.



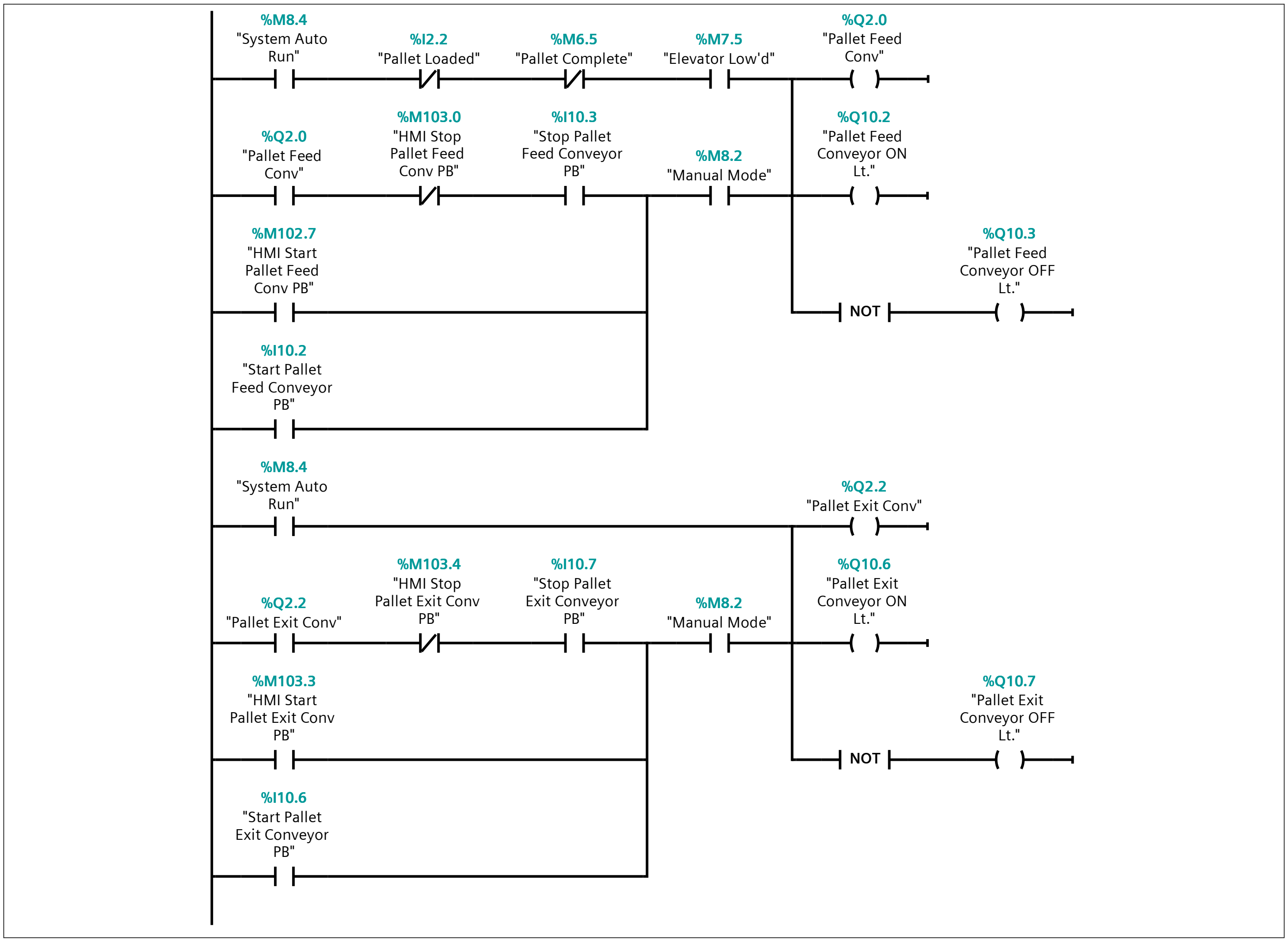
Network 7: Box Feed and Belt Conveyor Control

The operation of the Box Feed Conveyor and Load Belt are similar. Once the required number of boxes has accumulated in front of the pusher, the pusher advanced and the belts are stopped. An alternative is to stop the conveyor before the pusher advances but the logic would have to be modified. Since there is no possibility of collision, either method could be used.



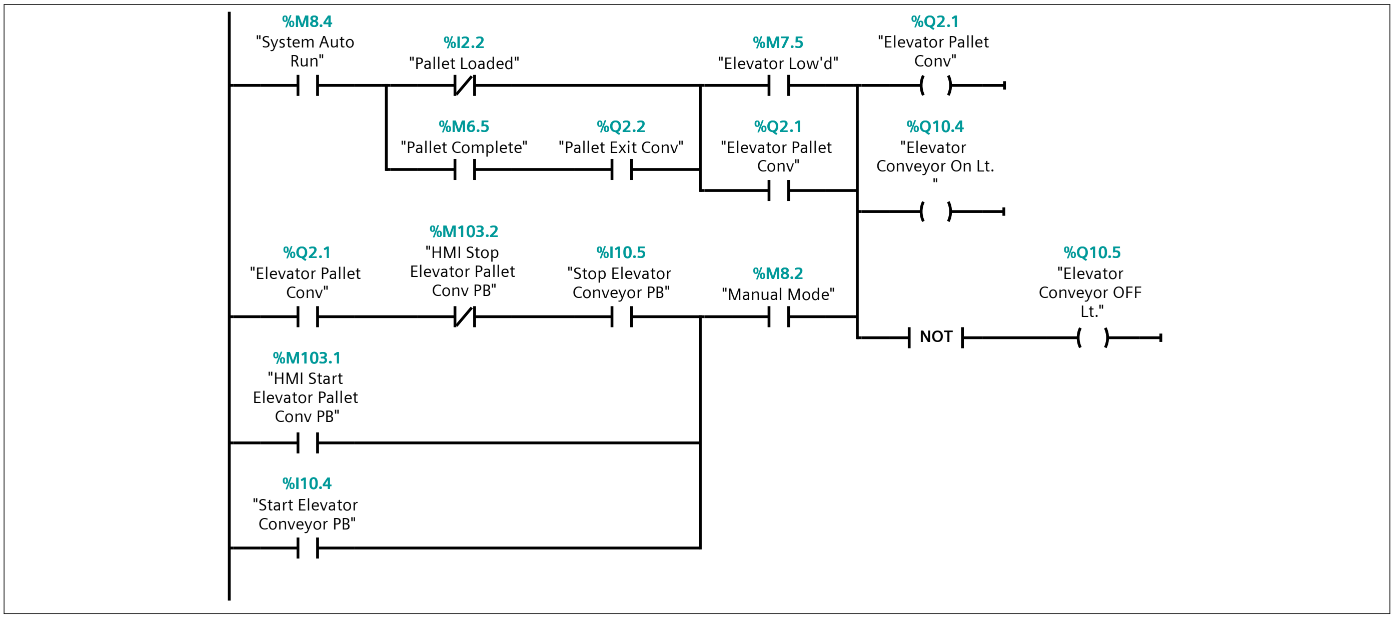
Network 8: Pallet Feed and Exit Conveyor Control

Pallet feed conveyor works in conjunction with the elevator conveyor but only runs when there is not pallet present in the elevator and the elevator is lowered.
The Pallet exit conveyor is always running but could be cycled if needed.


%M8.4
"System Auto Run"
%I2.2
"Pallet Loaded"
%M6.5
"Pallet Complete"
%M7.5
"Elevator Low'd"
%Q2.0
"Pallet Feed Conv"
%Q2.0
"Pallet Feed Conv"
%M103.0
"HMI Stop Pallet Feed Conv PB"
%I10.3
"Stop Pallet Feed Conveyor PB"
%M8.2
"Manual Mode"
%Q10.2
"Pallet Feed Conveyor ON Lt."
%M102.7
"HMI Start Pallet Feed Conv PB"
NOT
%Q10.3
"Pallet Feed Conveyor OFF Lt."
%I10.2
"Start Pallet Feed Conveyor PB"
%M8.4
"System Auto Run"
%Q2.2
"Pallet Exit Conv"
%Q2.2
"Pallet Exit Conv"
%M103.4
"HMI Stop Pallet Exit Conv PB"
%I10.7
"Stop Pallet Exit Conveyor PB"
%M8.2
"Manual Mode"
%Q10.6
"Pallet Exit Conveyor ON Lt."
%M103.3
"HMI Start Pallet Exit Conv PB"
NOT
%Q10.7
"Pallet Exit Conveyor OFF Lt."
%I10.6
"Start Pallet Exit Conveyor PB"

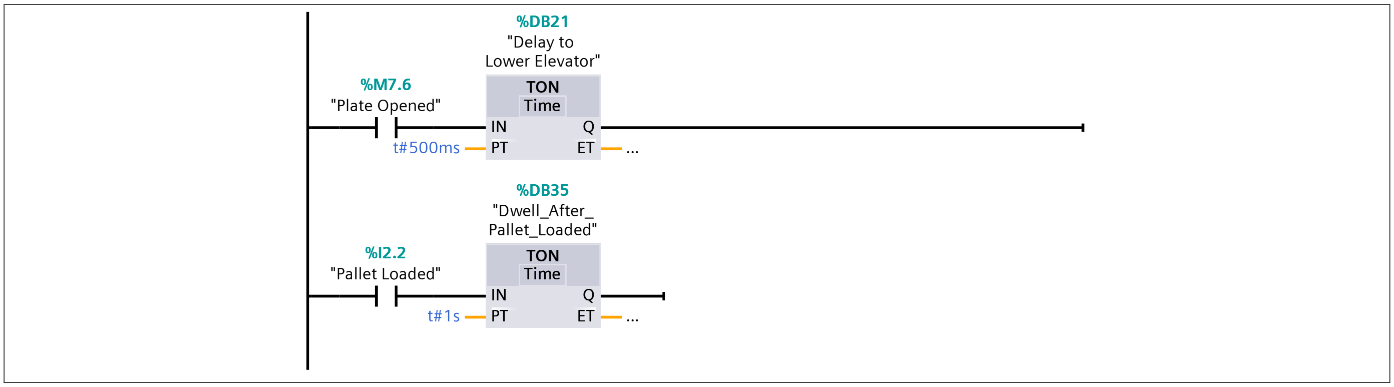
Network 9: Elevator Pallet Conveyor Control

Elevator pallet conveyor can only run if the elevator is lowered. If a pallet is not loaded, the elevator conveyor can run until a pallet is detected. If the pallet is complete, the elevator conveyor can run until the pallet is unloaded as long as the exit conveyor is running. Interlocking the elevator conveyor and the exit conveyor is important because, although the exit conveyor is always running for the purpose of this simulation, it could be programmed to shut down for energy savings or reduce wear and tear in a real factory.



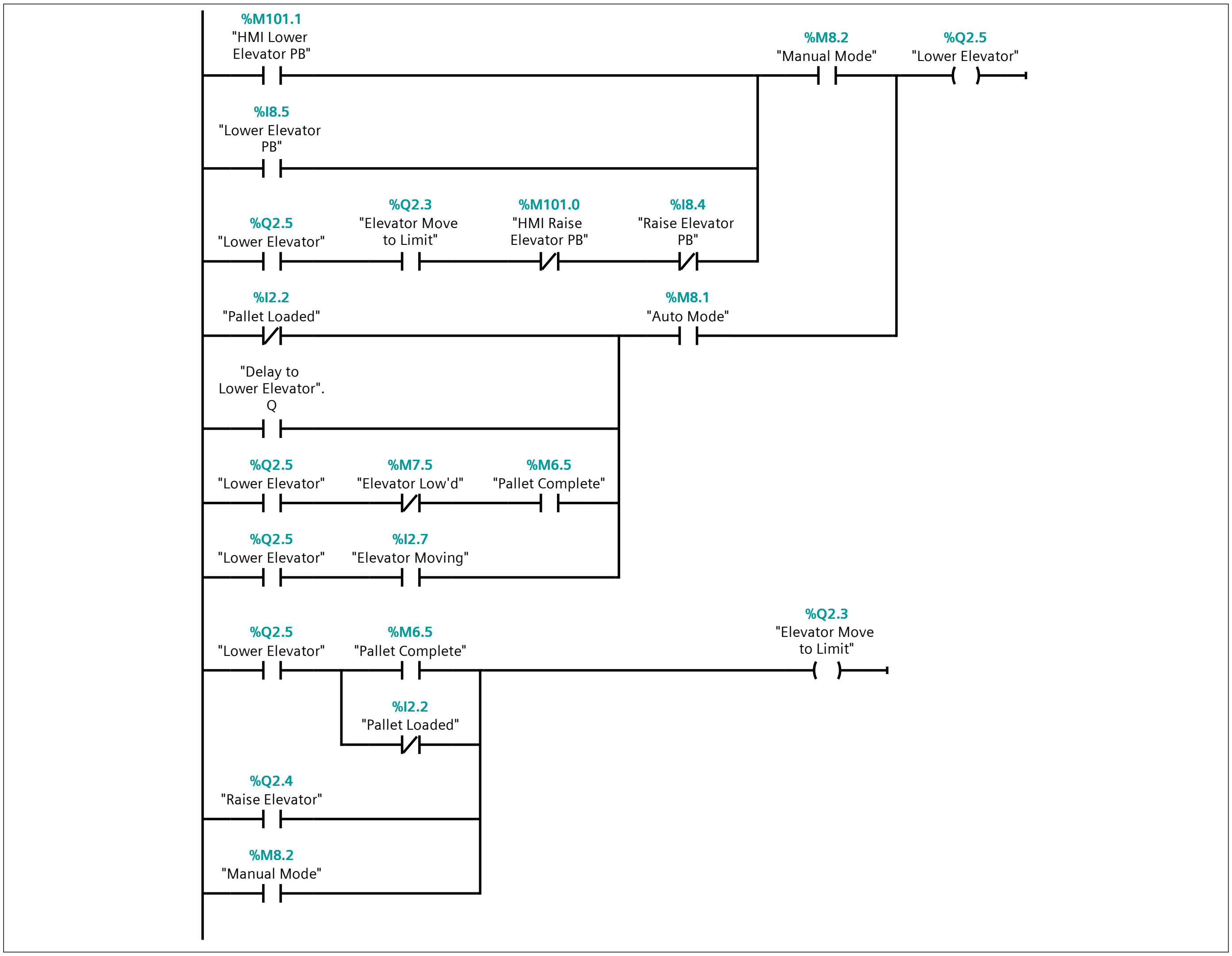
Network 10: Dwells for Elevator

Dwell timers are required for various reasons. In real world applications, sensors rarely detect exact positions of devices and some time is necessary to allow for movement to fully complete or stabilize.



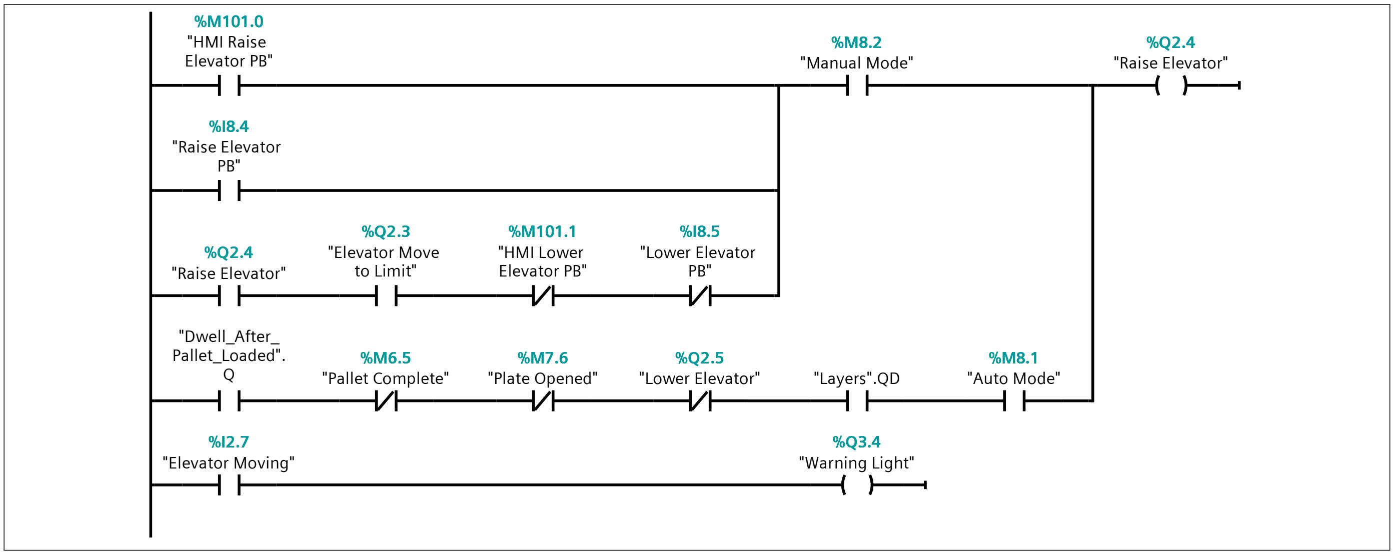
Network 11: Lower Elevator

If there is no pallet detected on the elevator conveyor, the elevator is commanded to lower. This is useful if the elevator is not fully in the lowered position when the program starts. The elevator is commanded to lower each time the plate is opened, after a small stabilization dwell time. Depending on the status of the build, the elevator will move incrementally to the next layer (elevator move to limit = OFF) or return all the way to the lowered position (elevator move to limit = ON).



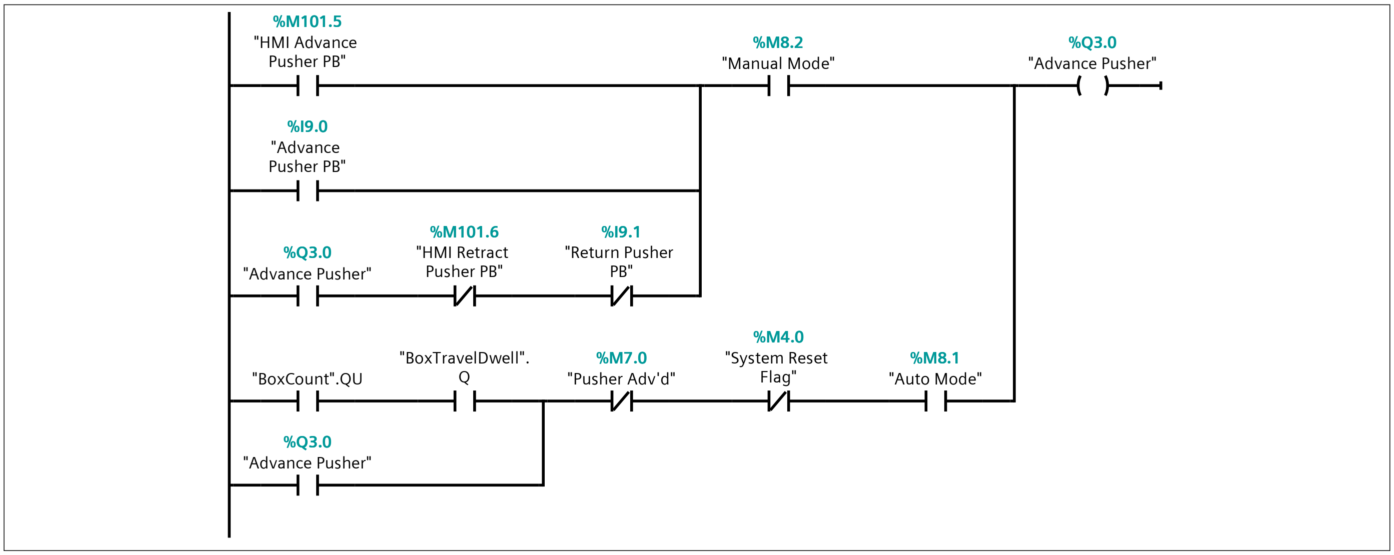
Network 12: Raise Elevator

The elevator will raise once a pallet is detected.



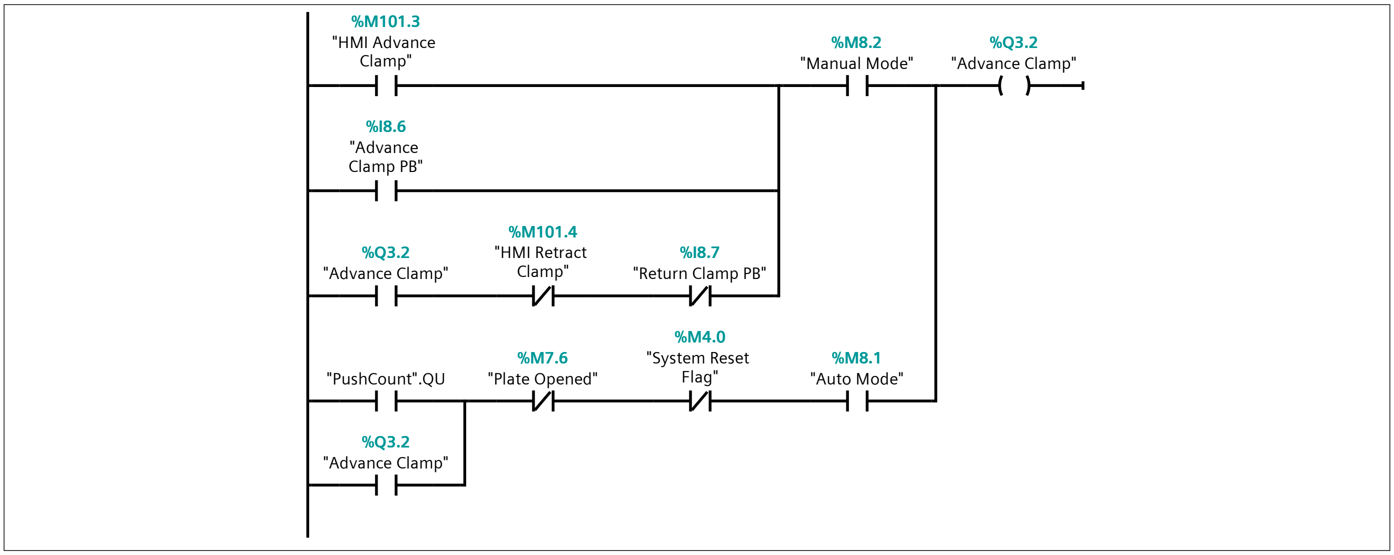
Network 13: Pusher Control

The pusher is advanced when the number of boxes in front of the pusher is equal to the setpoint. In this case the setpoint is either 2 or 3 boxes depending on their orientation. There is a small travel delay between counting of the boxes and allowing the pusher to advance. This is a single sided, spring return valve such that pusher returns as soon as advance pusher turns OFF.



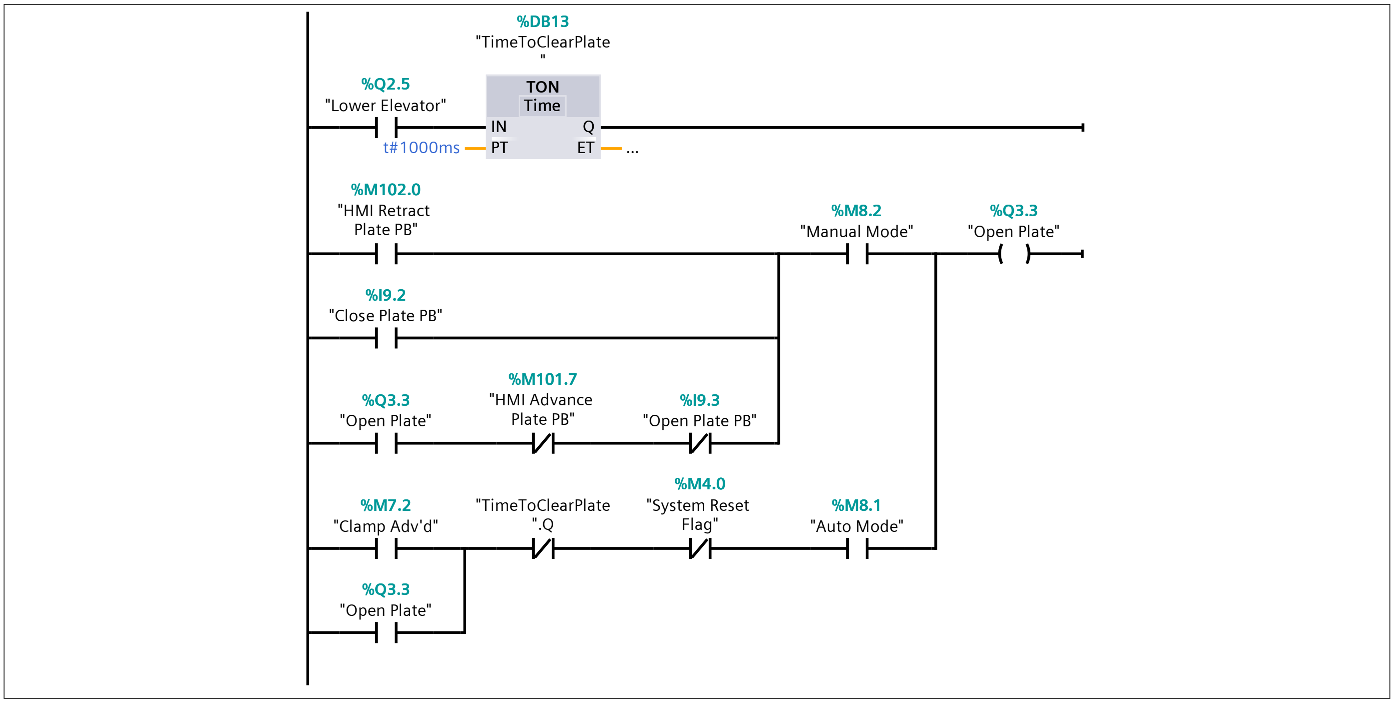
Network 14: Clamp Control

The clamp is advanced when the number of pushes is equal to the setpoint. In this case the setpoint is either 2 or 3 pushes depending on their orientation. This is a single sided, spring return valve such that clamp returns as soon as advance clamp turns OFF.



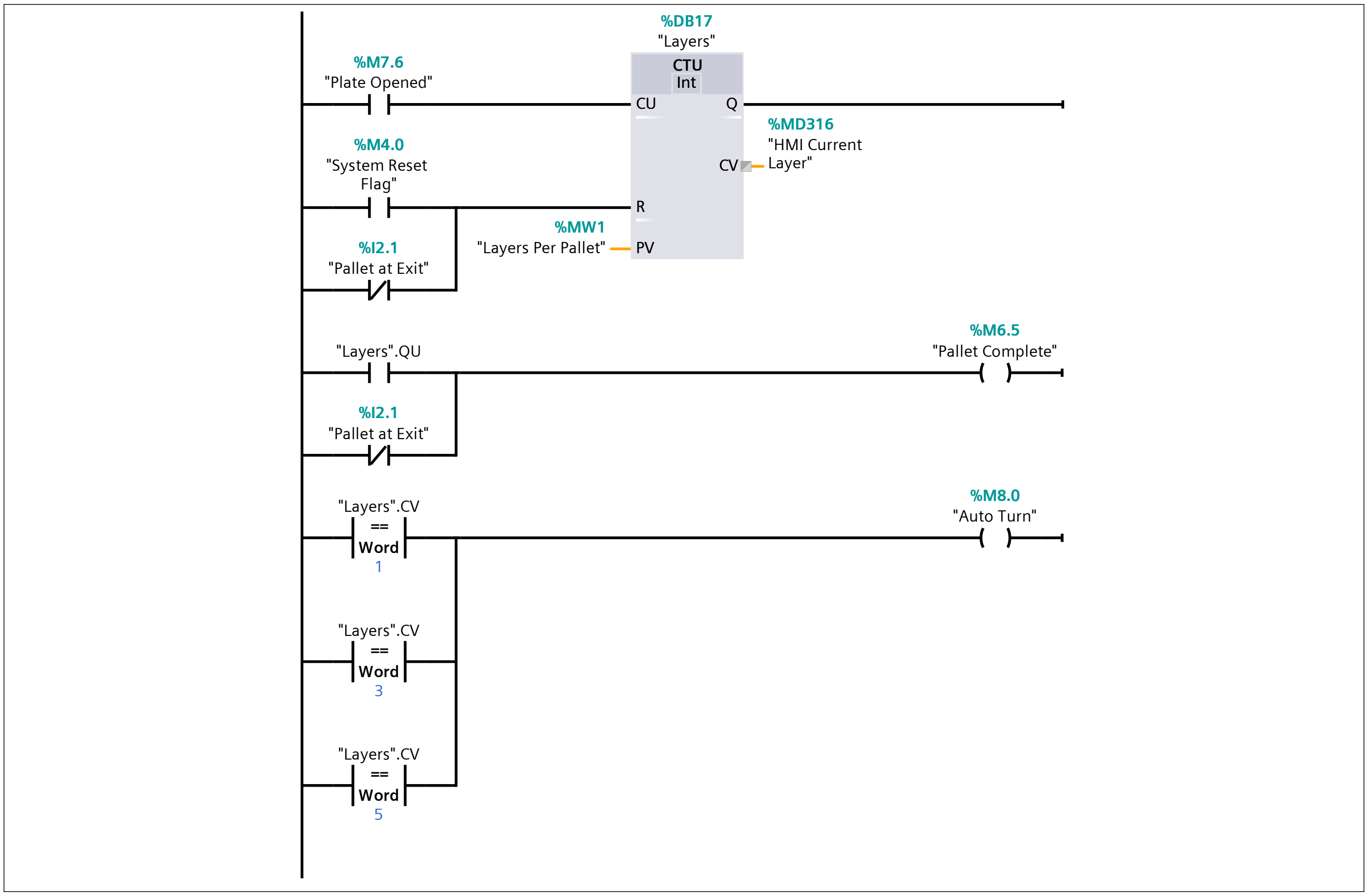
Network 15: Plate Control

The plate (aka shutter) is opened when the clamp is advanced allowing the boxes to drop to the pallet to form a layer. The open plate output is sealed ON and remains on until the timer "TimeToClearPlate" reaches its setpoint. Although this timer starts when the elevator begins to move, another method would be to use the open plate output to start the timer but add the dwell time required in the lower elevator rung. Either method works so long as the timers are coordinated properly.



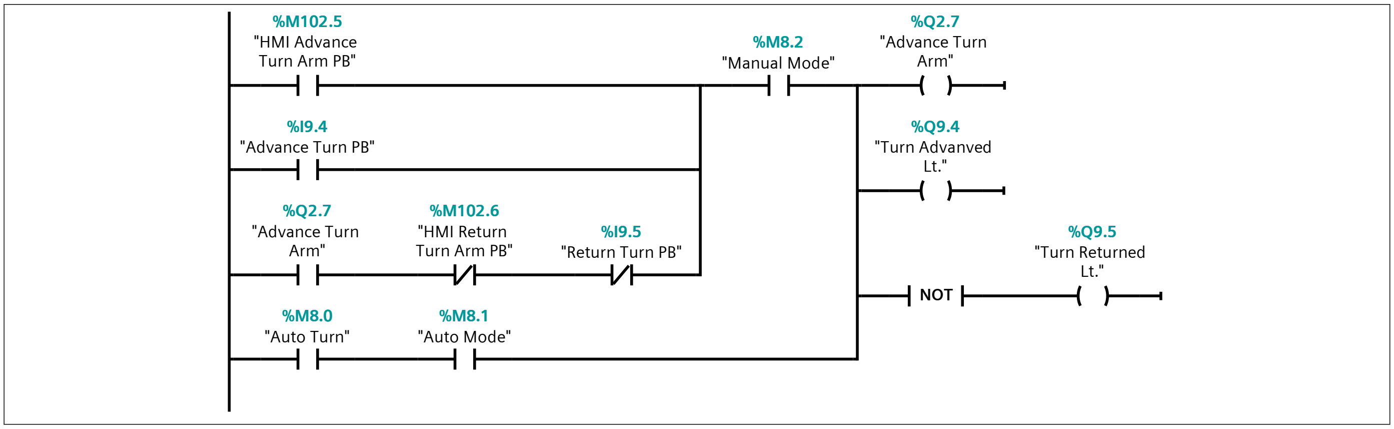
Network 16: Layer Status

Counts the number of layers build using the plate opened status bit. Once this counter reached the setpoint, the "Elevator Move to Limit" output is turned ON allowing the elevator fully lower. There is a delay between opening the plate bing opened and the elevator being commanded to lower which also allows for the "Elevator Move to Limit" output to turn ON without causing a race condition. The output which controls the box turn arm is hard-coded to advance after layers 1, 3 and 5 have been competed.

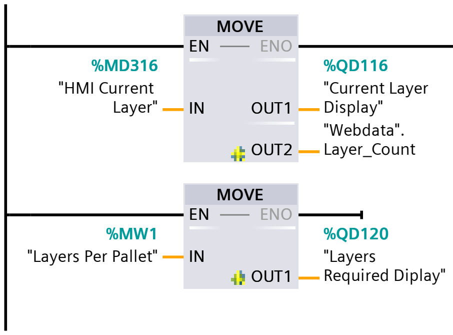


Network 17: Turn Arm

The output which controls the box turn arm is hard-coded to advance after layers 1, 3 and 5 have been competed.



Network 18: Transfer Data to Display


MOVE
EN
ENO
%MD316
"HMI Current
Layer"
IN
OUT1
OUT2
%QD116
"Current Layer
Display"
"Webdata".
Layer_Count
MOVE
EN
ENO
%MW1
"Layers Per Pallet"
IN
OUT1
%QD120
"Layers
Required Diplay"

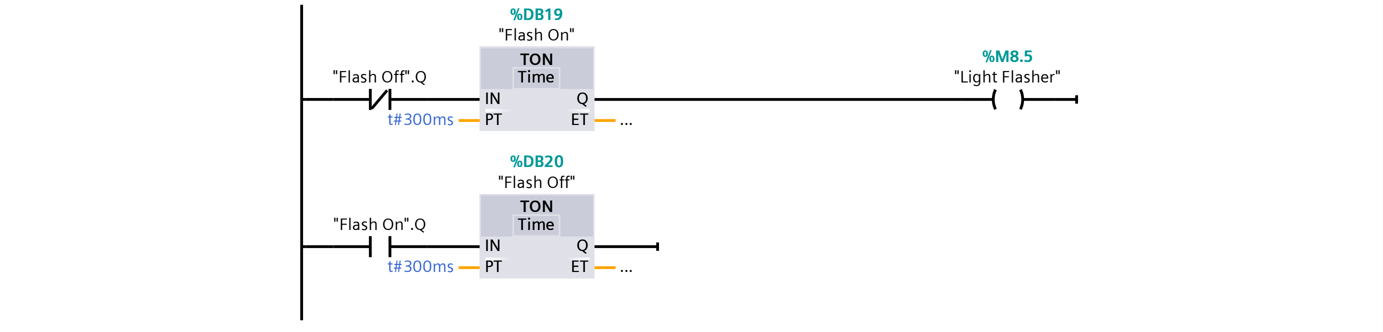
Program blocks

Mode [FC3]

Mode Properties					
General					
Name	Mode	**Number**	3	**Type**	FC
Language	LAD	**Numbering**	Automatic		
Information					
Title	Mode Selection	**Author**		**Comment**	Mode Selection and Start Command.
Family		**Version**	0.1	**User-defined ID**	

Network 1: Flasher

Use this bit to control lamp flashing and display update timing.

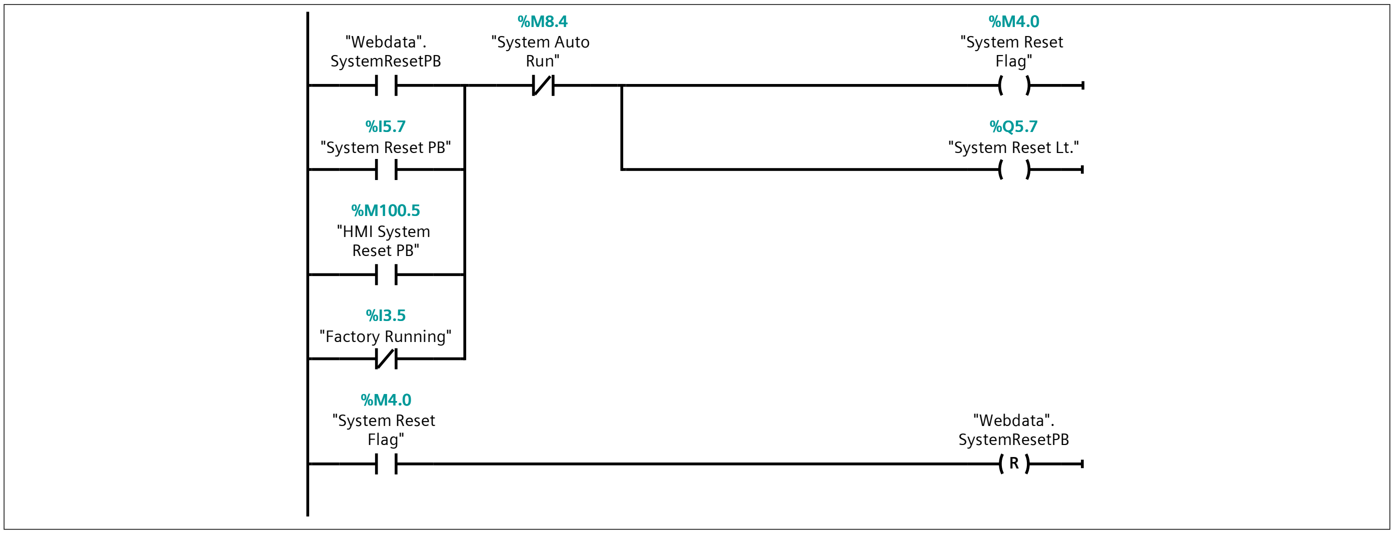


Network 2: System Reset

System can be reset if it is not in Automatic. Reset can be done from inside the Simulation, from the HMI and the Webpage. A reset is also required if the Simulation if turned off in order to clear any active seals or memories.

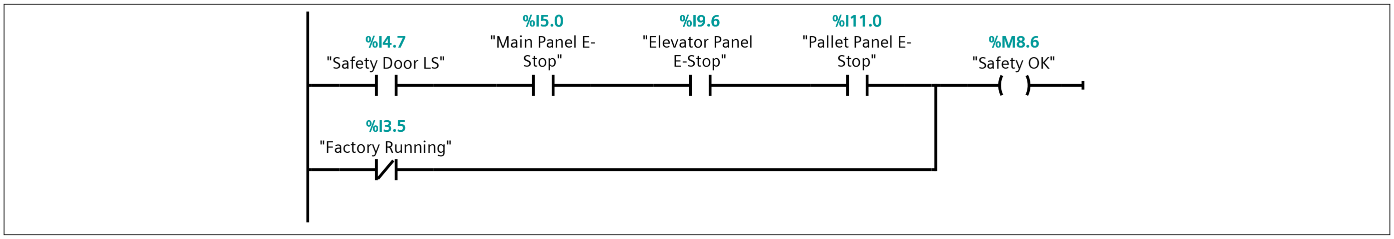
Note that Webdata bit is reset after setting an internal memory. This allows simplification of the webpage by resetting the Webdata directly from the PLC code after confirming receipt. The alternative is to run a script on the webpage that holds the bit on for X seconds and then automatically reset. The problem with the timed method is that there is no confirmation that the PLC actually received the bit from the webpage.

All note that the System Reset Flag is turned ON when the Factory Simulation is Stopped. This is due to the loss of all objects and positions in the simulation. Similar to removing all the parts from a real system and restarting the system.



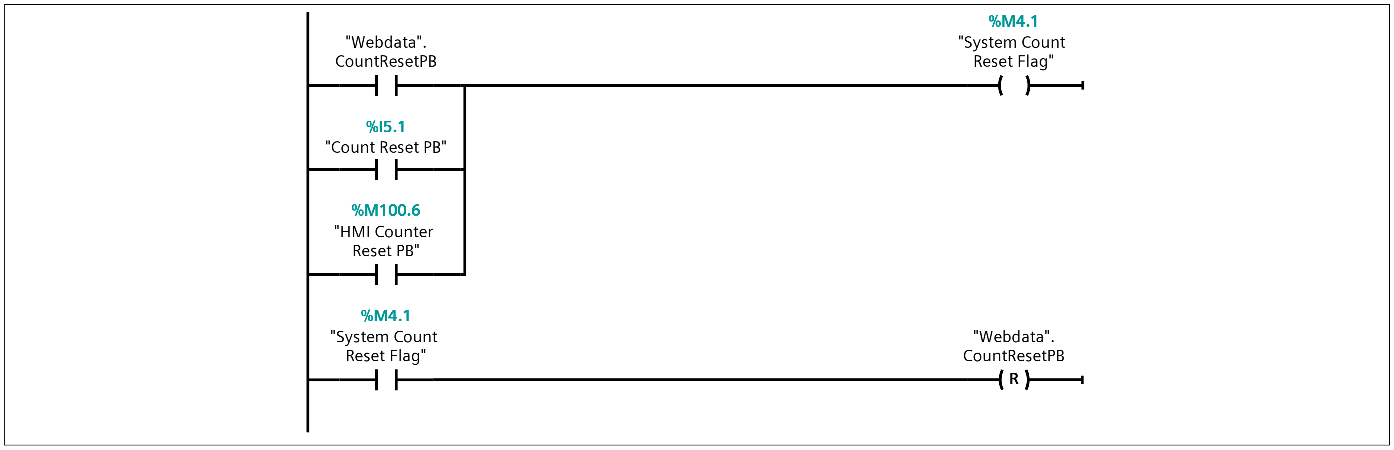
Network 3: Safety Rung

Check safety switches and door. Safety is set to be OK if the Simulation is not running to prevent a fault from being generated on the HMI.



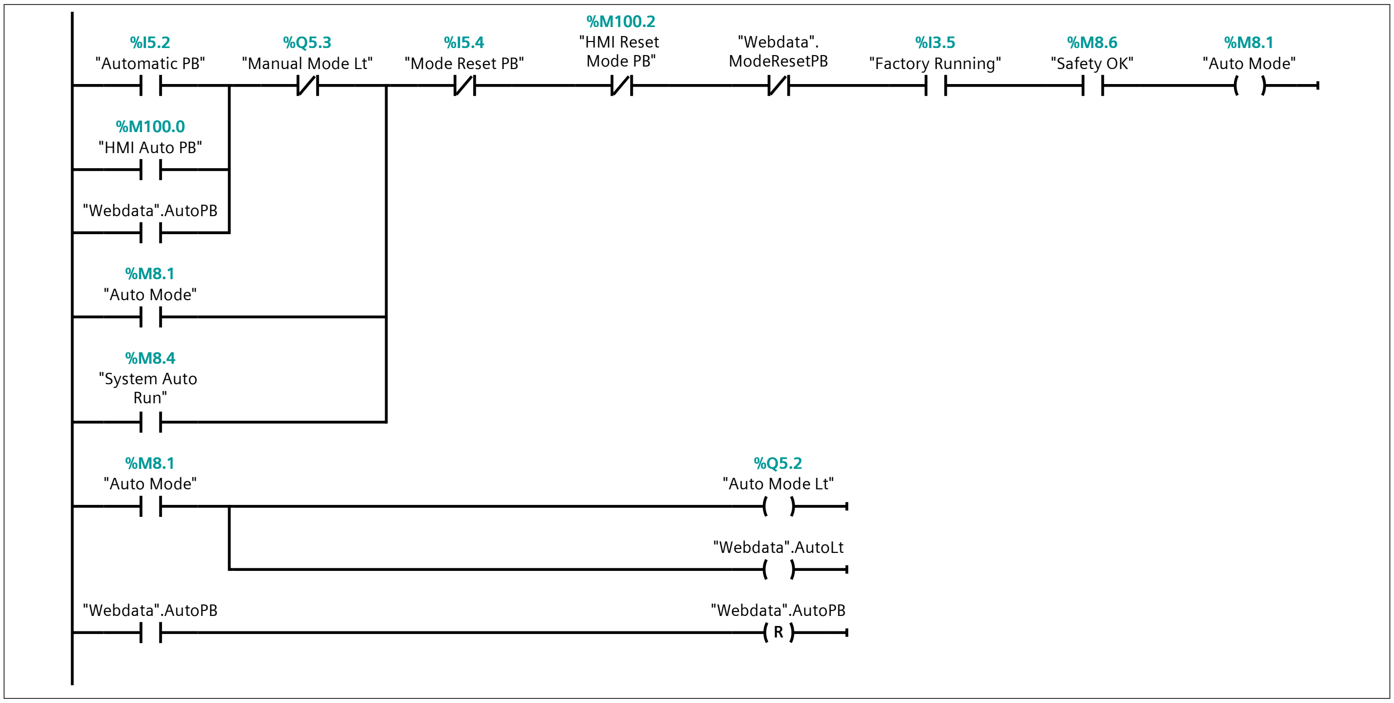
Network 4: System Count Reset

Set a flag to reset all the counters. Uses the same concept to reset the databit associated with the webserver as in the rung above. Count reset can be done at any time. This does not reset the counters that track the statue of the Palletizer. The Palletizer counters are reset with a general "System Reset".



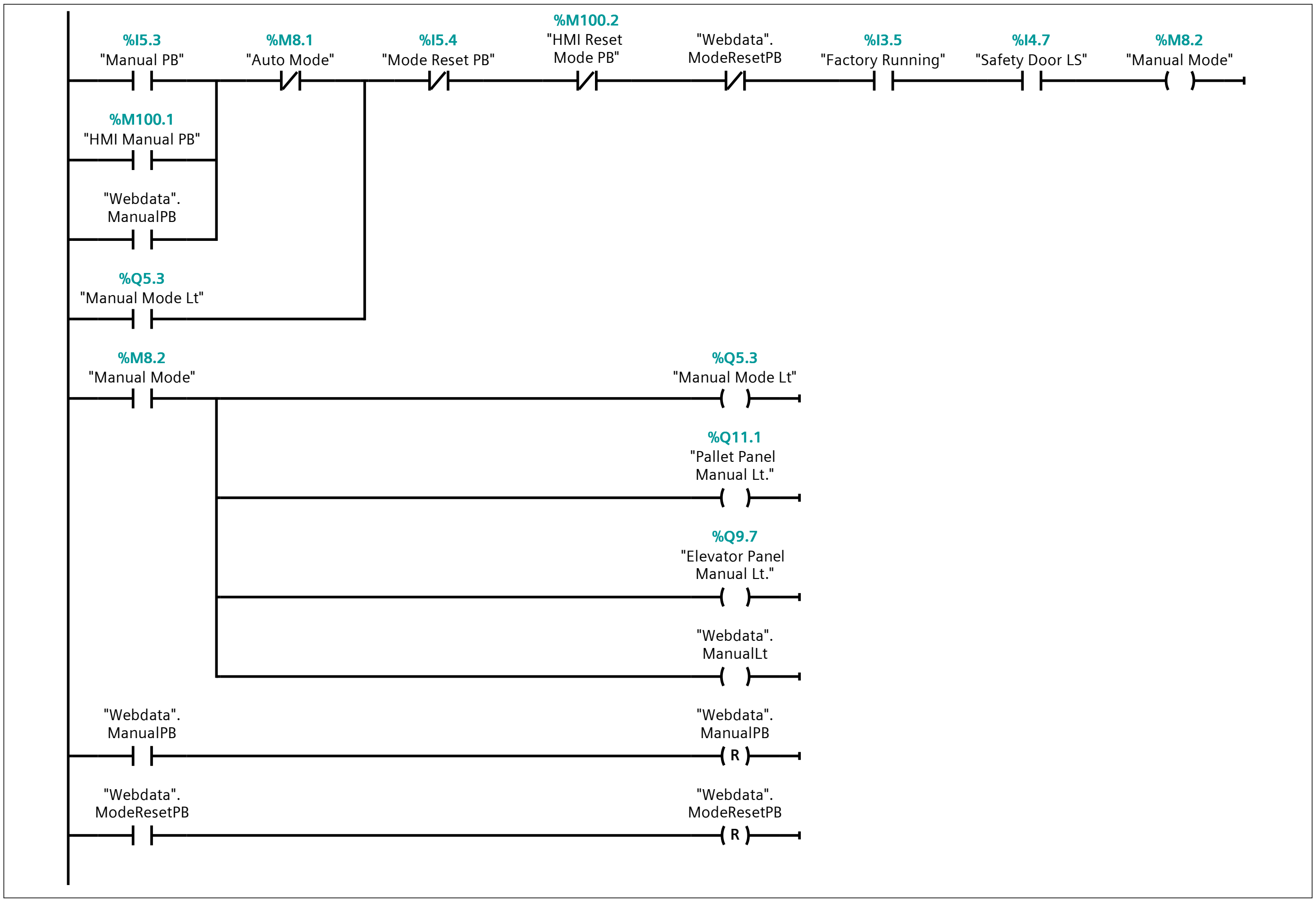
Network 5: Auto Mode

Although the Auto Mode Lt. output can be used in place of an Automatic Mode Flag bit, the style of this network is slightly cleaner as it differentiate between an internal Status Bit and a Functional Output. Using this technic would allow for the Auto Mode Lt to have different functions such as Flashing if Auto Mode is not allowed when the Auto PB is pressed. Another example is demonstrated on the System Start Rung.



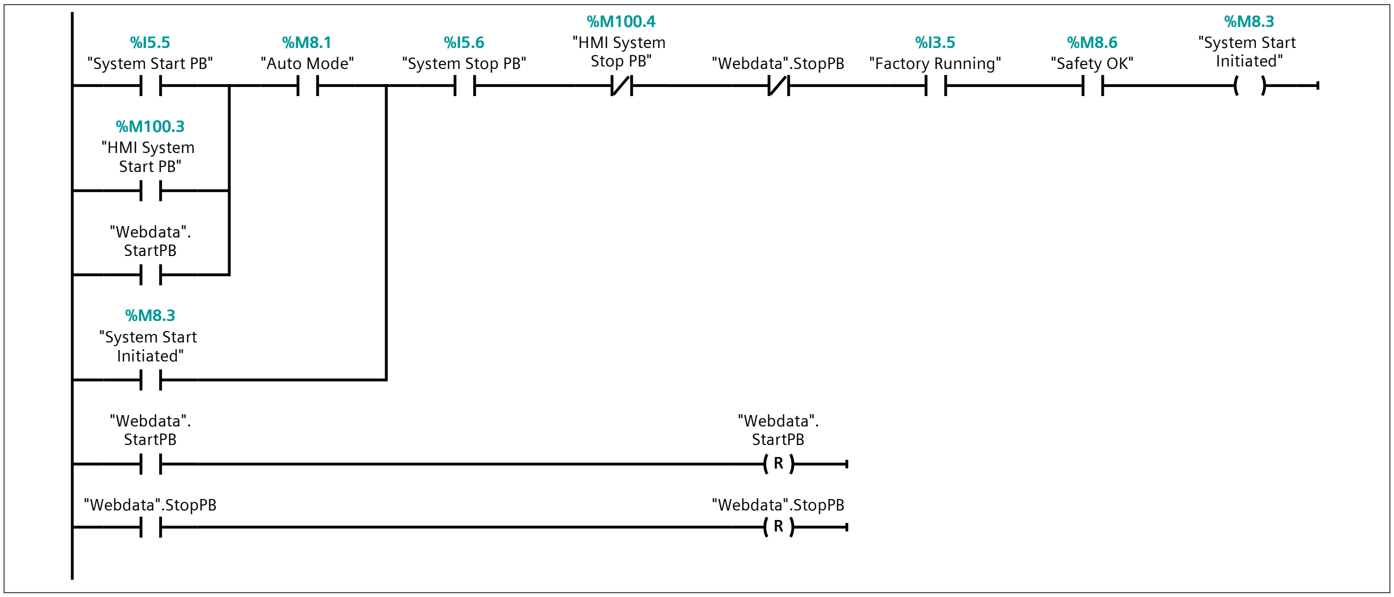
Network 6: Manual Mode

As an alternative to the way the System Reset webdata bit was captured and turns on another internal flag, the Mode Reset is used directly as a normally closed contact to release Auto or Manual Mode. The trick here is to reset the Webdata in the last rung the bit is used. The main difference between Mode Reset and System Reset is that Mode Reset is only used in two rungs whereas System Reset is used throughout the program.

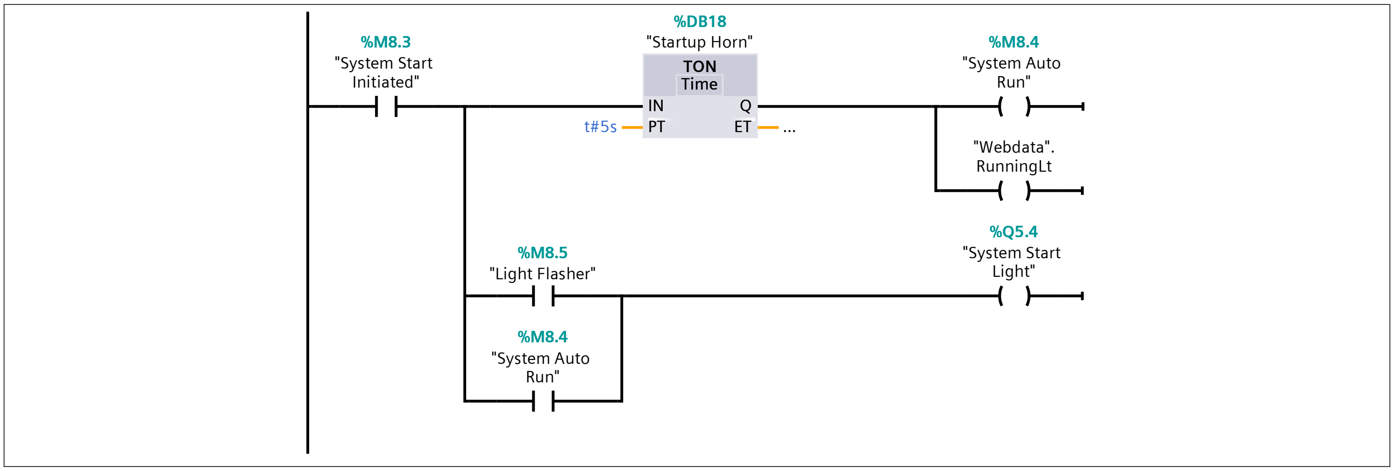


Network 7: System Start Initiated

System start has been initiated. There is no abort for this command, the Startup Horn is only a warning that motion is about to occur, not a delay to allow system start to be terminated prematurely.

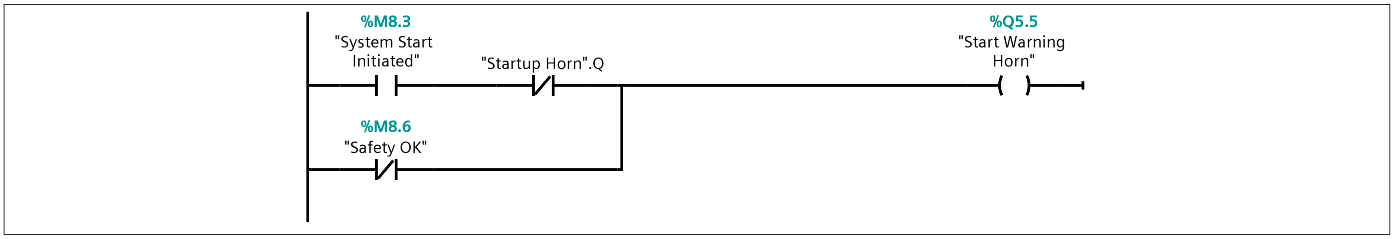


Network 8: System Startup



Network 9: Warning Horn

The warning horn is used for Startup and Safety Gate open warning. Typically a real machine would use diffennent horns or sounds for startup and a critical fault.



Program blocks

Webdata [DB22]

Webdata Properties					
General					
Name	Webdata	**Number**	22	**Type**	DB
Language	DB	**Numbering**	Automatic		
Information					
Title	Webdata	**Author**		**Comment**	Data for the webserver
Family		**Version**	0.1	**User-defined ID**	

Name	Data type	Start value	Retain
▼ Static			
Left_Count	DWord	16#0	False
Right_Count	DWord	16#0	False
Front_Count	DWord	16#0	False
Layer_Count	DWord	16#0	False
AutoPB	Bool	false	False
ManualPB	Bool	false	False
StartPB	Bool	false	False
ModeResetPB	Bool	false	False
SystemResetPB	Bool	false	False
CountResetPB	Bool	false	False
StopPB	Bool	false	False
AutoLt	Bool	false	False
ManualLt	Bool	false	False
RunningLt	Bool	false	False

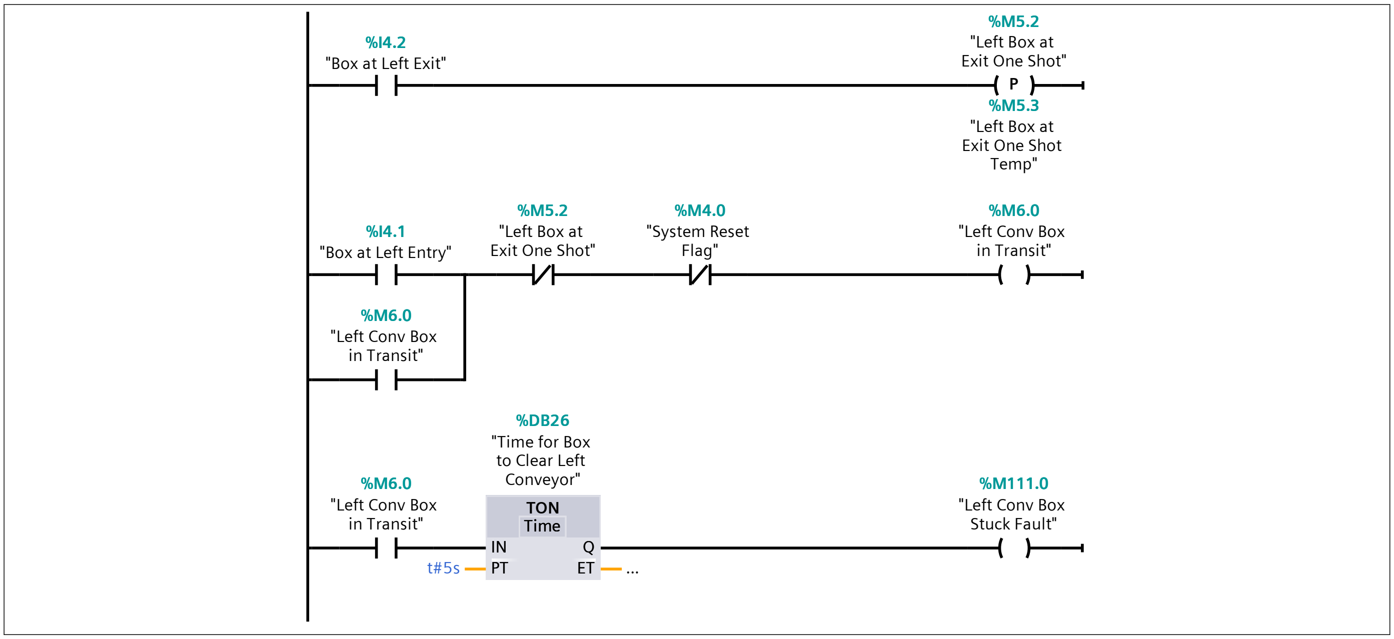
Program blocks

Faults [FC4]

Faults Properties					
General					
Name	Faults	**Number**	4	**Type**	FC
Language	LAD	**Numbering**	Automatic		
Information					
Title		**Author**		**Comment**	Current alarms using Alarm Word 1. Word 2 is being used for Safety Messages. It is common to group alarms as much as possible to make diagnostic easier although grouping requires quite a bit of forward planning or rework. Another common practice is to designate blocks of words with 20% spare in each group. Note the byte swap: For MW110: MB111 is the LSB (A0-A7) MB110 is the USB (A8-A15) For MW112: MB113 is the LSB (A16-A24) MB112 is the USB (A25-A32) Also note that bits start at 0 but alarm ID starts at 1.
Family		**Version**	0.1	**User-defined ID**	

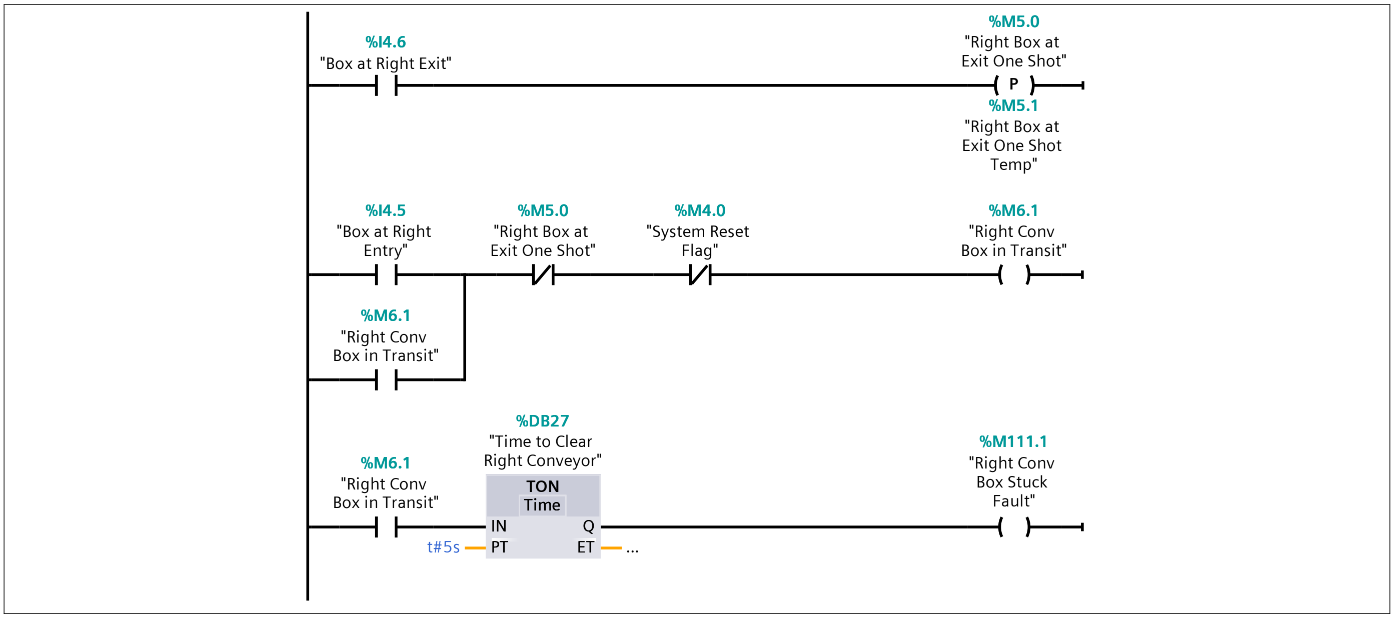
Network 1: (A1) Box Stuck on Left Conveyor

A one shot is used to cancel the Transit flag to capture the cases of a box either not making it to the exit proximity switch or getting stuck in front of the exit proximity switch. A transition is required for correct operation.
Alarm Word 1, Bit 0


%I4.2
"Box at Left Exit"
%M5.2
"Left Box at Exit One Shot"
P
%M5.3
"Left Box at Exit One Shot Temp"
%I4.1
"Box at Left Entry"
%M5.2
"Left Box at Exit One Shot"
%M4.0
"System Reset Flag"
%M6.0
"Left Conv Box in Transit"
%M6.0
"Left Conv Box in Transit"
%DB26
"Time for Box to Clear Left Conveyor"
TON
Time
%M6.0
"Left Conv Box in Transit"
IN
Q
t#5s
PT
ET
...
%M111.0
"Left Conv Box Stuck Fault"

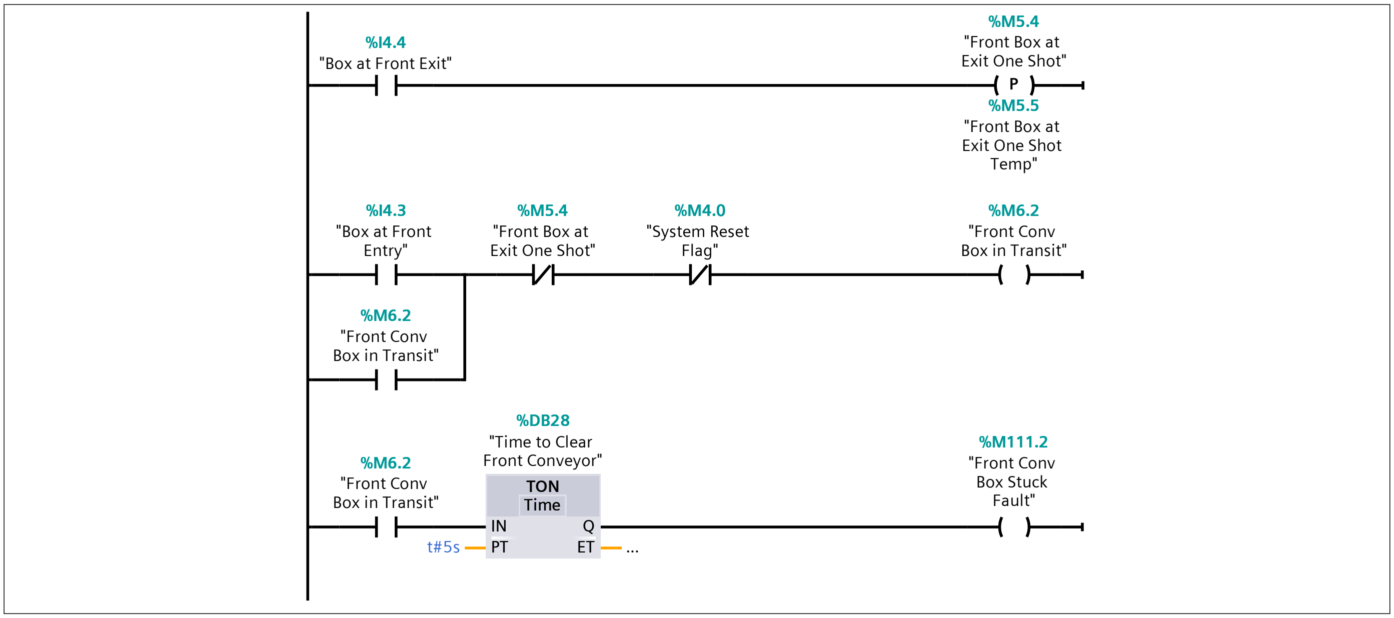
Network 2: (A2) Box Stuck on Right Conveyor

Alarm Word 1, Bit 1


%I4.6
"Box at Right Exit"
%M5.0
"Right Box at Exit One Shot"
P
%M5.1
"Right Box at Exit One Shot Temp"
%I4.5
"Box at Right Entry"
%M5.0
"Right Box at Exit One Shot"
%M4.0
"System Reset Flag"
%M6.1
"Right Conv Box in Transit"
%M6.1
"Right Conv Box in Transit"
%DB27
"Time to Clear Right Conveyor"
TON
Time
%M6.1
"Right Conv Box in Transit"
IN
Q
t#5s
PT
ET
...
%M111.1
"Right Conv Box Stuck Fault"

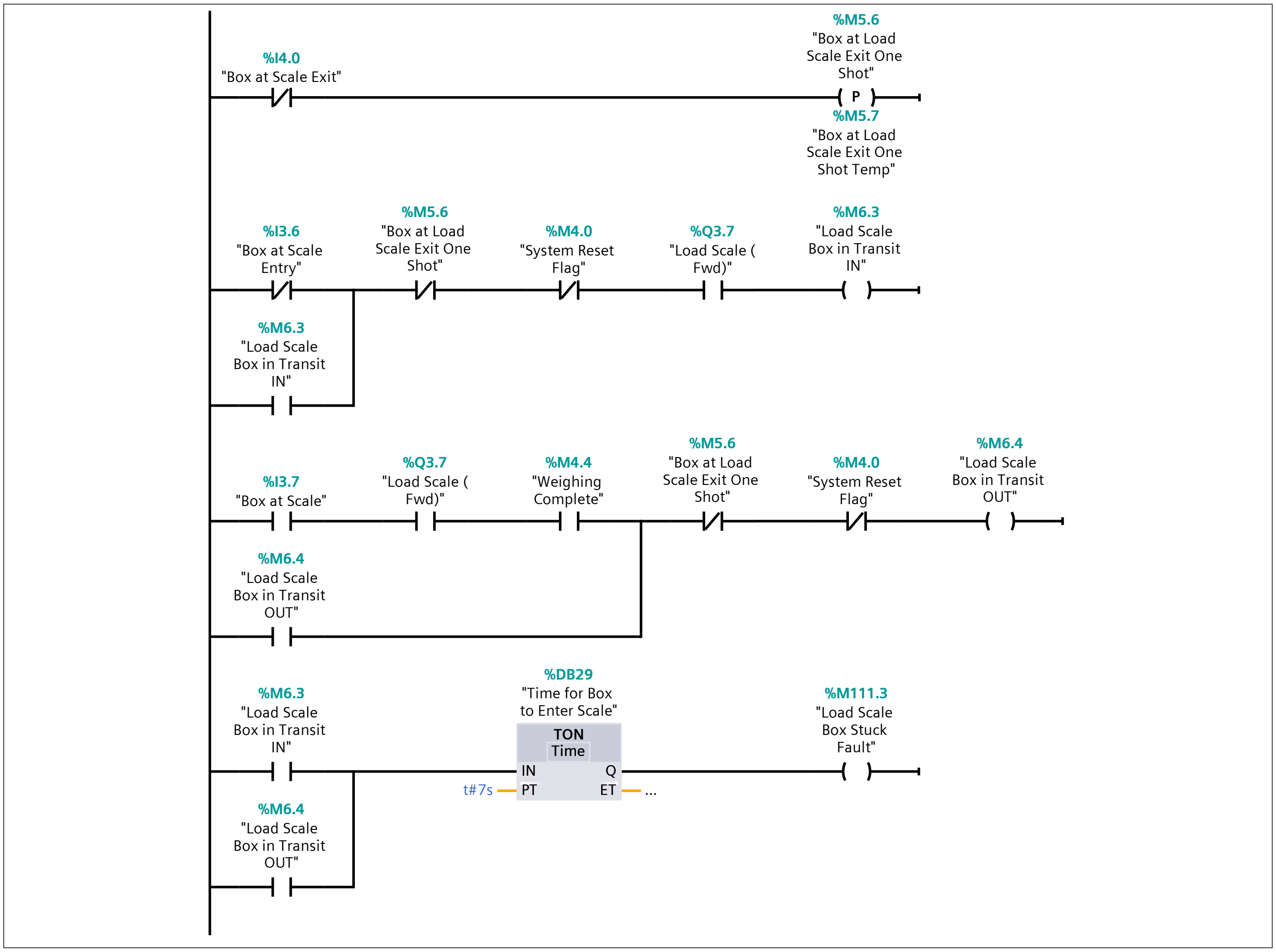
Network 3: (A3) Box Stuck on Front Conveyor

Alarm Word 1, Bit 2


%I4.4
"Box at Front Exit"
%M5.4
"Front Box at Exit One Shot"
P
%M5.5
"Front Box at Exit One Shot Temp"
%I4.3
"Box at Front Entry"
%M5.4
"Front Box at Exit One Shot"
%M4.0
"System Reset Flag"
%M6.2
"Front Conv Box in Transit"
%M6.2
"Front Conv Box in Transit"
%DB28
"Time to Clear Front Conveyor"
TON
Time
%M6.2
"Front Conv Box in Transit"
IN
Q
t#5s
PT
ET
...
%M111.2
"Front Conv Box Stuck Fault"

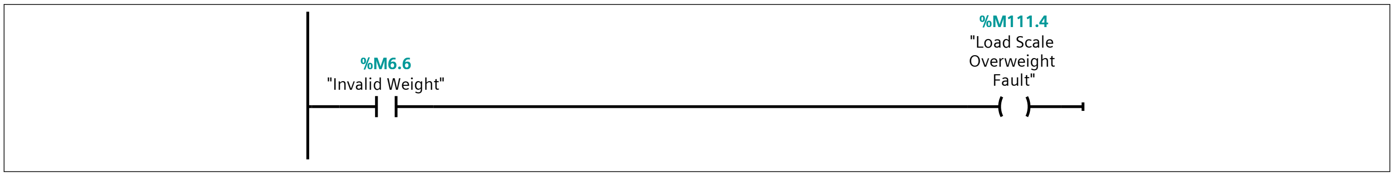
Network 4: (A4) Box Stuck on Entry to Scale

Alarm Word 1, Bit 3


%I4.0
"Box at Scale Exit"
%M5.6
"Box at Load Scale Exit One Shot"
P
%M5.7
"Box at Load Scale Exit One Shot Temp"
%I3.6
"Box at Scale Entry"
%M5.6
"Box at Load Scale Exit One Shot"
%M4.0
"System Reset Flag"
%Q3.7
"Load Scale (Fwd)"
%M6.3
"Load Scale Box in Transit IN"
%M6.3
"Load Scale Box in Transit IN"
%I3.7
"Box at Scale"
%Q3.7
"Load Scale (Fwd)"
%M4.4
"Weighing Complete"
%M5.6
"Box at Load Scale Exit One Shot"
%M4.0
"System Reset Flag"
%M6.4
"Load Scale Box in Transit OUT"
%M6.4
"Load Scale Box in Transit OUT"
%DB29
"Time for Box to Enter Scale"
TON
Time
%M6.3
"Load Scale Box in Transit IN"
IN
Q
t#7s
PT
ET
...
%M111.3
"Load Scale Box Stuck Fault"
%M6.4
"Load Scale Box in Transit OUT"

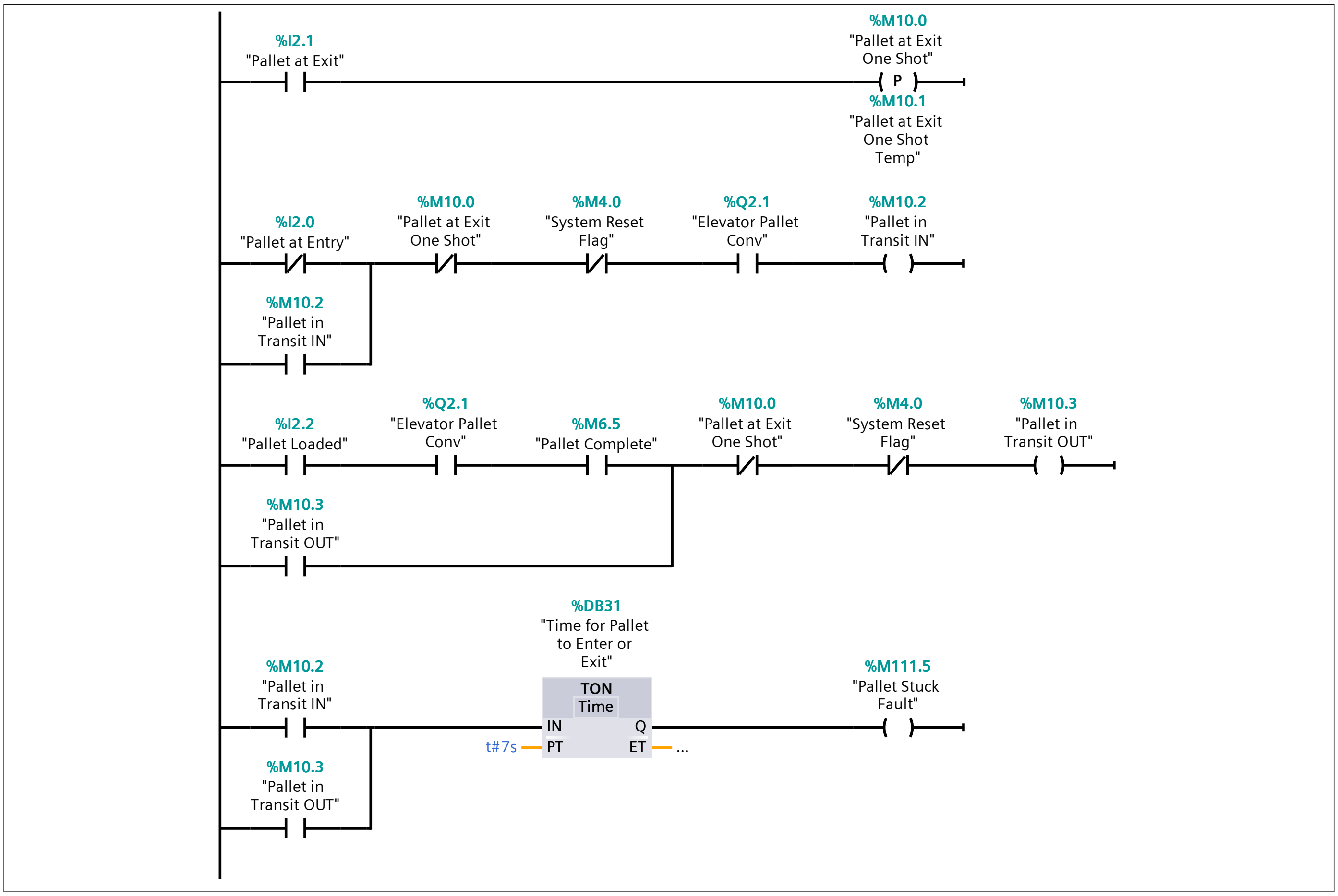
Network 5: (A5) Invalid Weight on Scale

Alarm Word 1, Bit 4



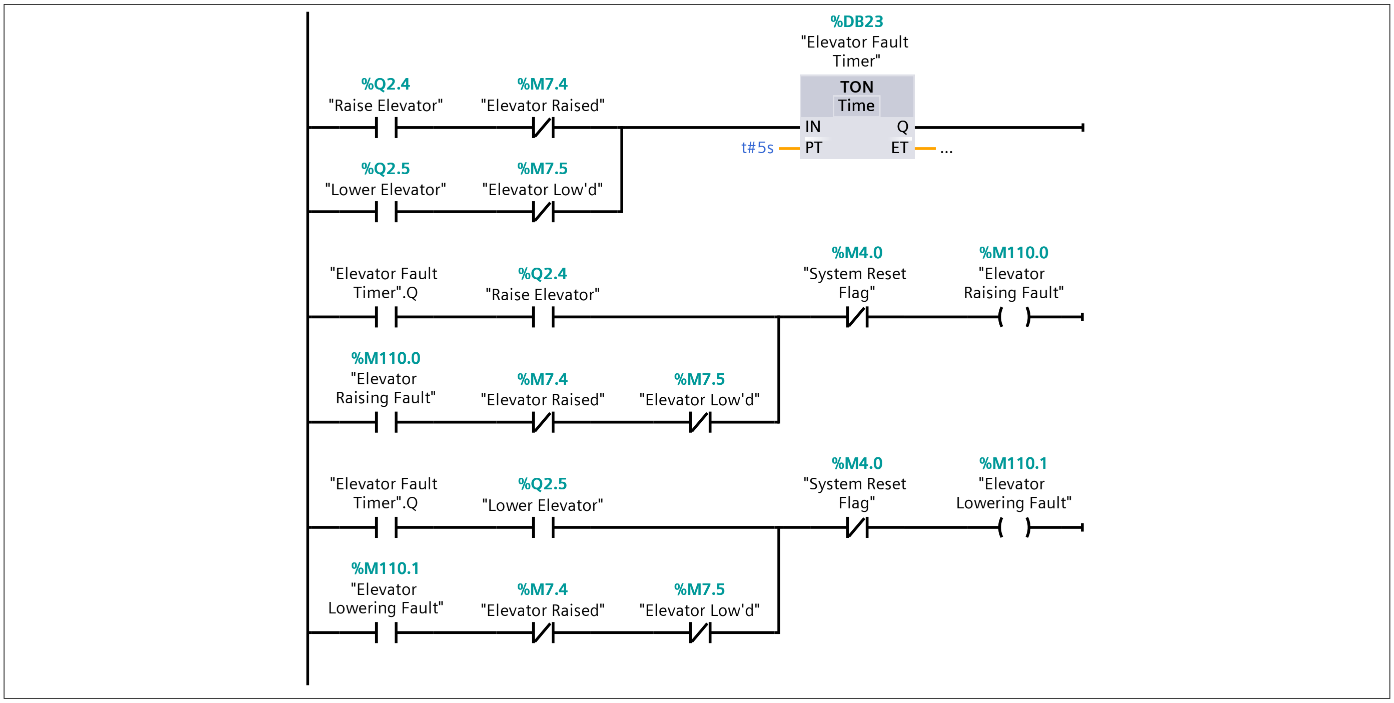
Network 6: (A6) Pallet Stuck in Elevator

Alarm Word 1, Bit 5



Network 7: (A9) Elevator Raising or (A10) Lowering Fault

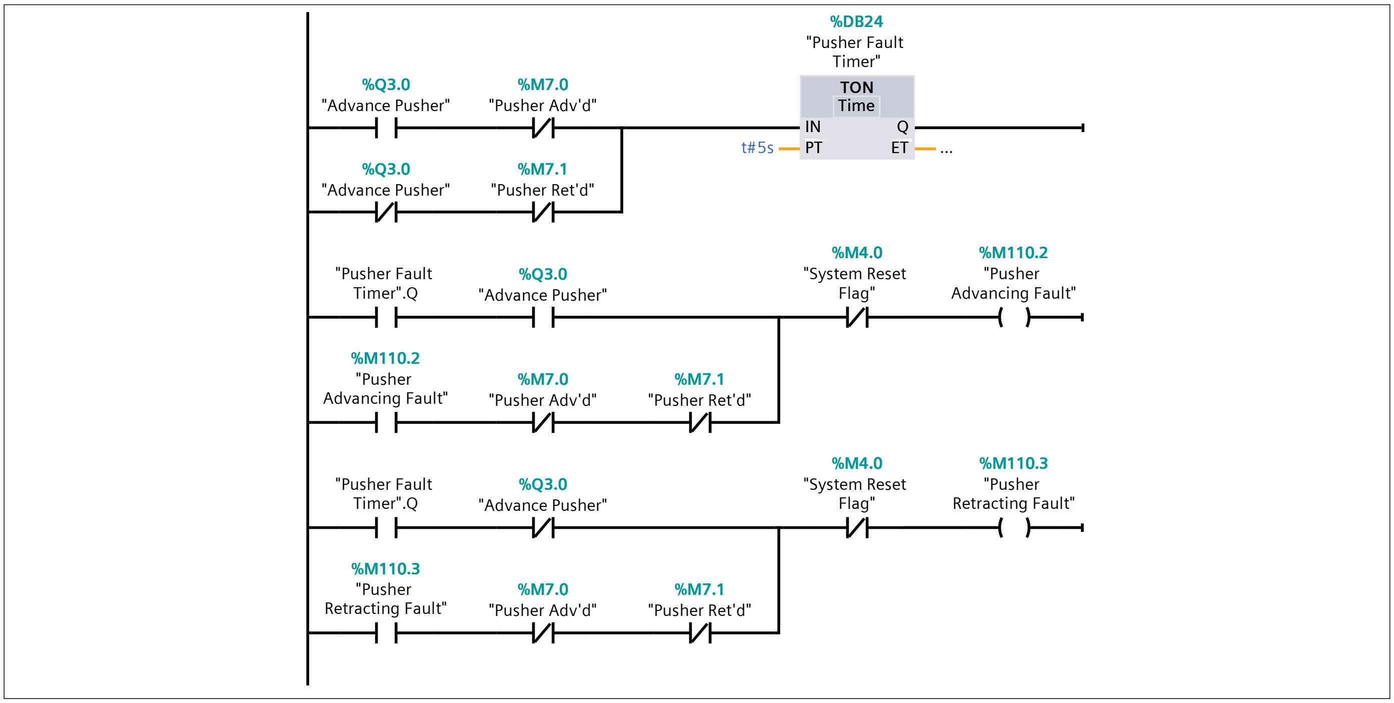
Alarm Word 1, Bits 8 and 9



Network 8: (A11) Pusher Advancing and (A12) Returning Fault

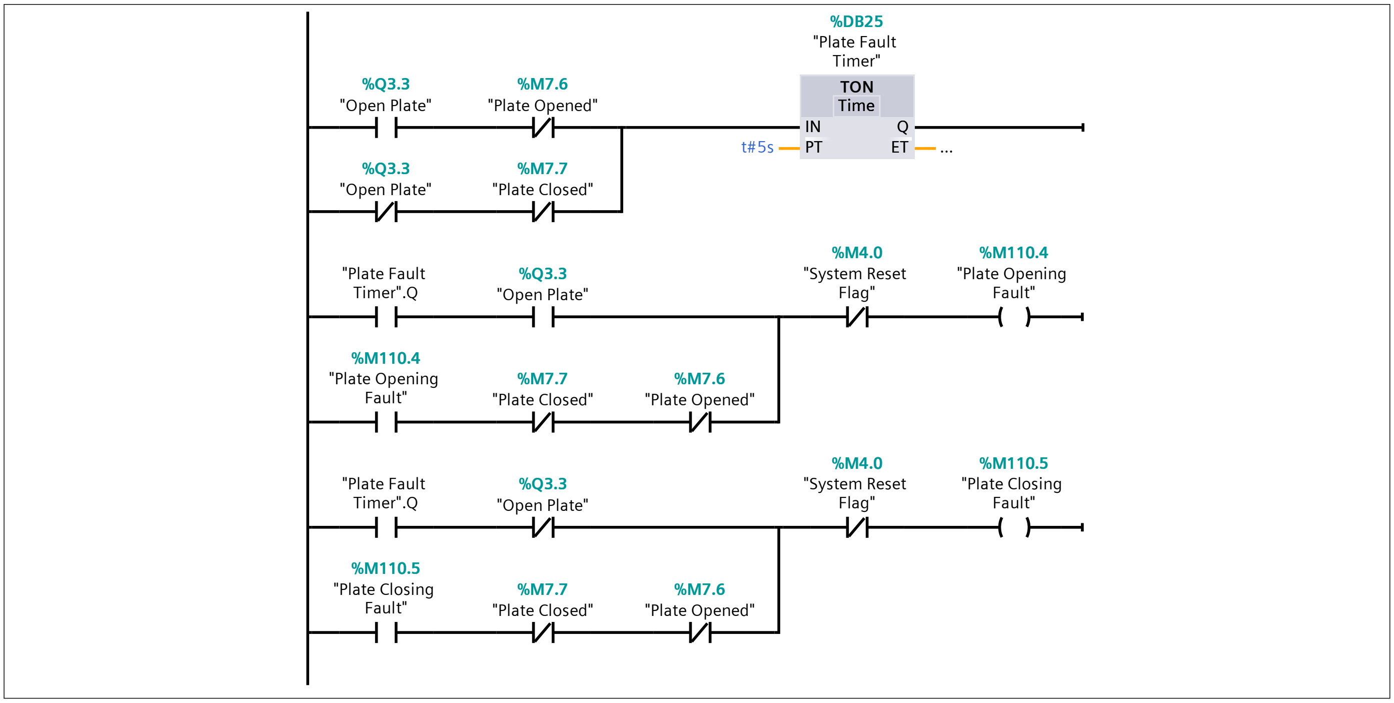
Pusher fault will self reset once pusher is back in the advanced or returned position. Typically jam ups are cleared by putting the system in manual and pressing a fault reset button. For the purposes of demonstration, a box can be cleared using the mouse, causing the pusher to move and reset its own fault.

Alarm Word 1, Bits 10 and 11



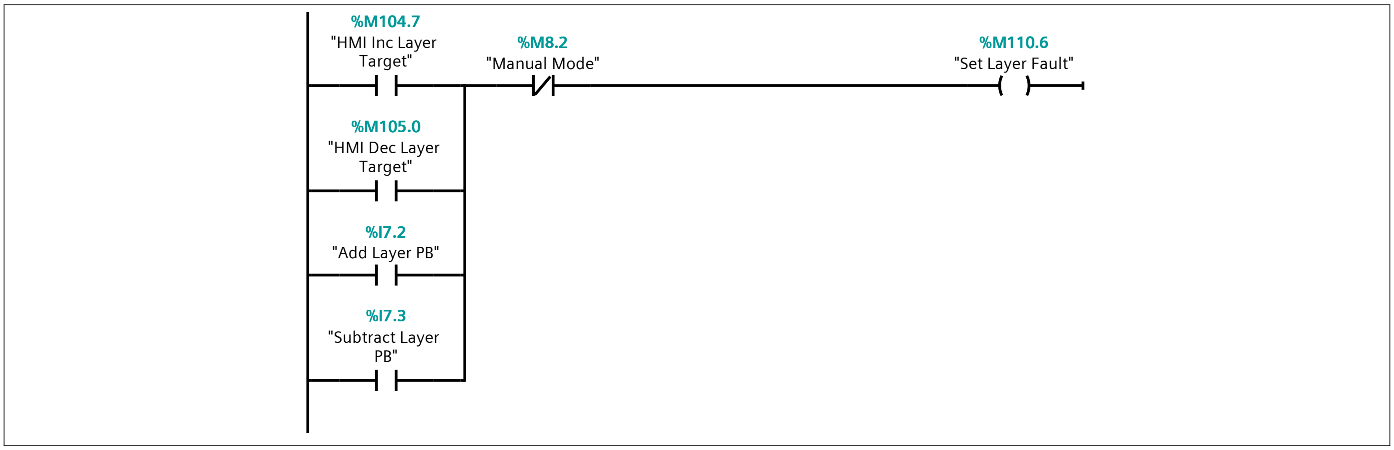
Network 9: (A13) Plate Opening and (A14) Closing Fault

Alarm Word 1, Bits 12 and 13



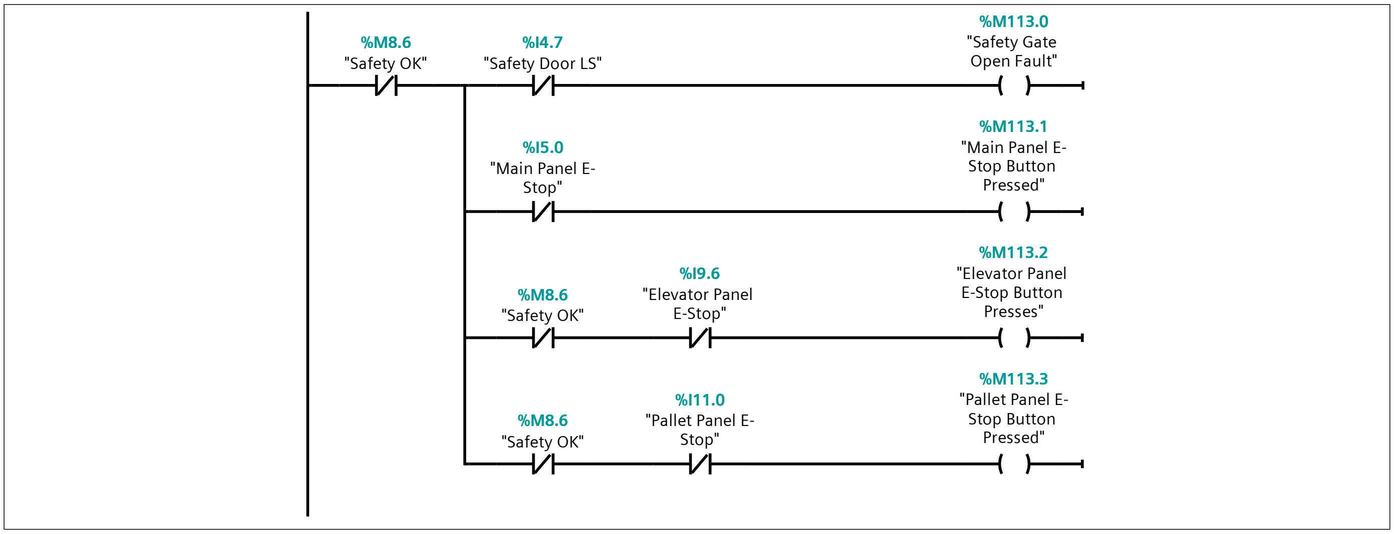
Network 10: (A15) Set Layer Fault

Alarm Word 1, Bit 14



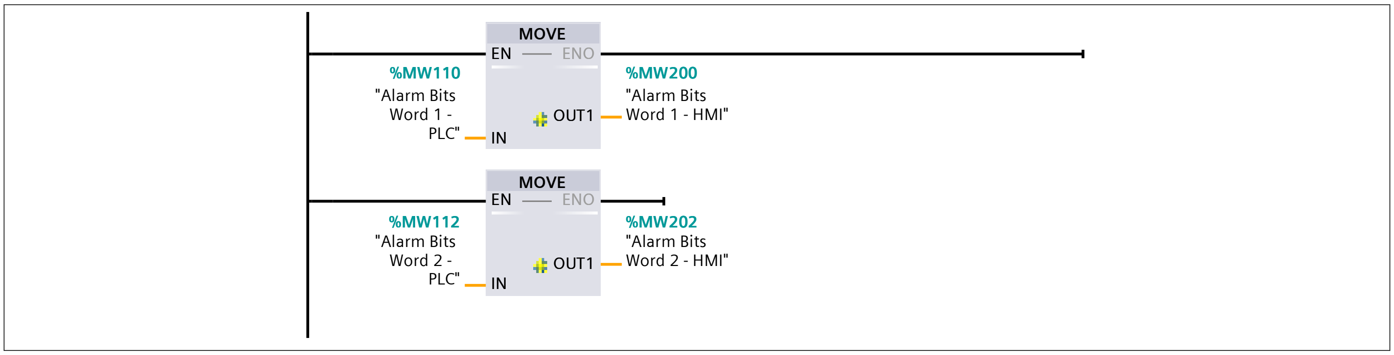
Network 11: (A16) - (A19) Safety Messages

Alarm Word 2, Bits 0 - 3



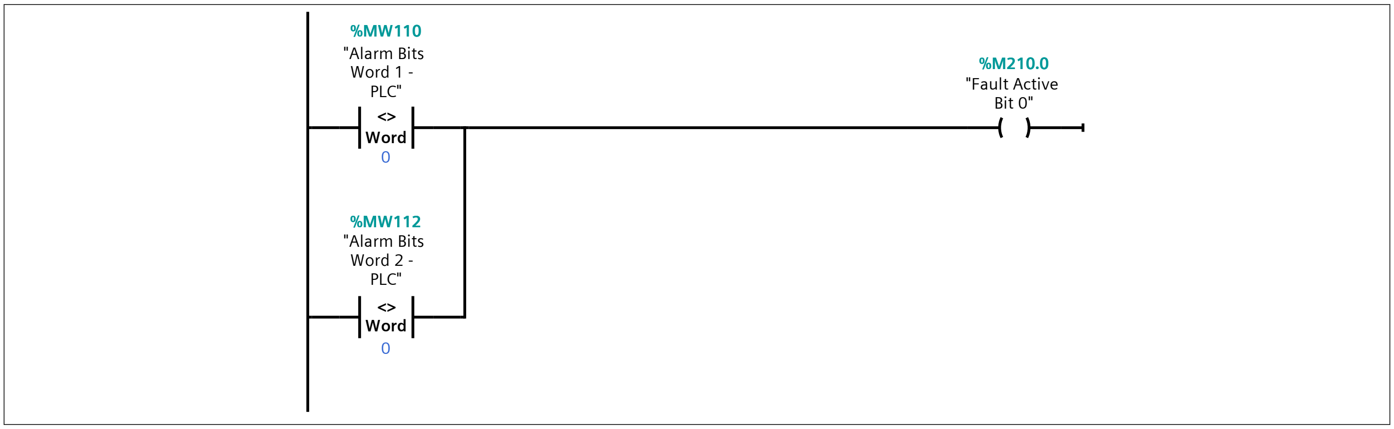
Network 12: Move PLC Fault Bits to HMI for Alarm Words

This method currently only set up for 32 faults.


MOVE
EN
ENO
%MW110
"Alarm Bits Word 1 - PLC"
IN
OUT1
%MW200
"Alarm Bits Word 1 - HMI"
MOVE
EN
ENO
%MW112
"Alarm Bits Word 2 - PLC"
IN
OUT1
%MW202
"Alarm Bits Word 2 - HMI"

Network 13: Set Fault Status Bit Turn on Panel Fault Lights

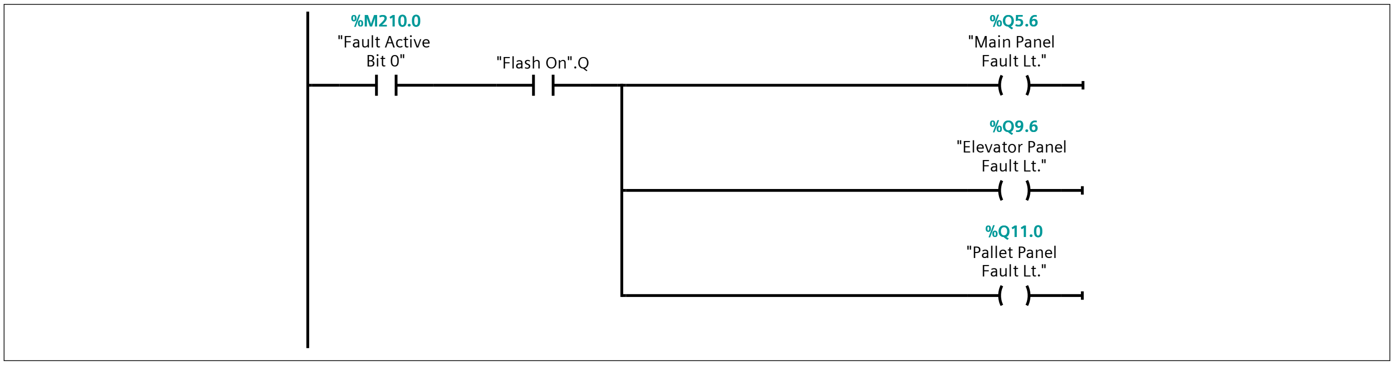
Bit required for HMI Fault Banner Annimation
If Warnings and Faults are separated, another bit could be used for another alarm banner color
Currently all faults change the alarm banner to Red.



Network 14: Turn on Panel Fault Lights

Flash all alarm light regardless of where the alarm occurs.

One enhancement here is to only flash an alarm light in the area that requires operator attention.



Program blocks

Weight_Configuration_Data [DB32]

Weight_Configuration_Data Properties					
General					
Name	Weight_Configuration_Data	**Number**	32	**Type**	DB
Language	DB	**Numbering**	Automatic		
Information					
Title	Weight Limit Storage	**Author**		**Comment**	This is a datablock use to store the configuration data used by the weight scale in determining which conveyor a box is sent to.
Family		**Version**	1.0	**User-defined ID**	

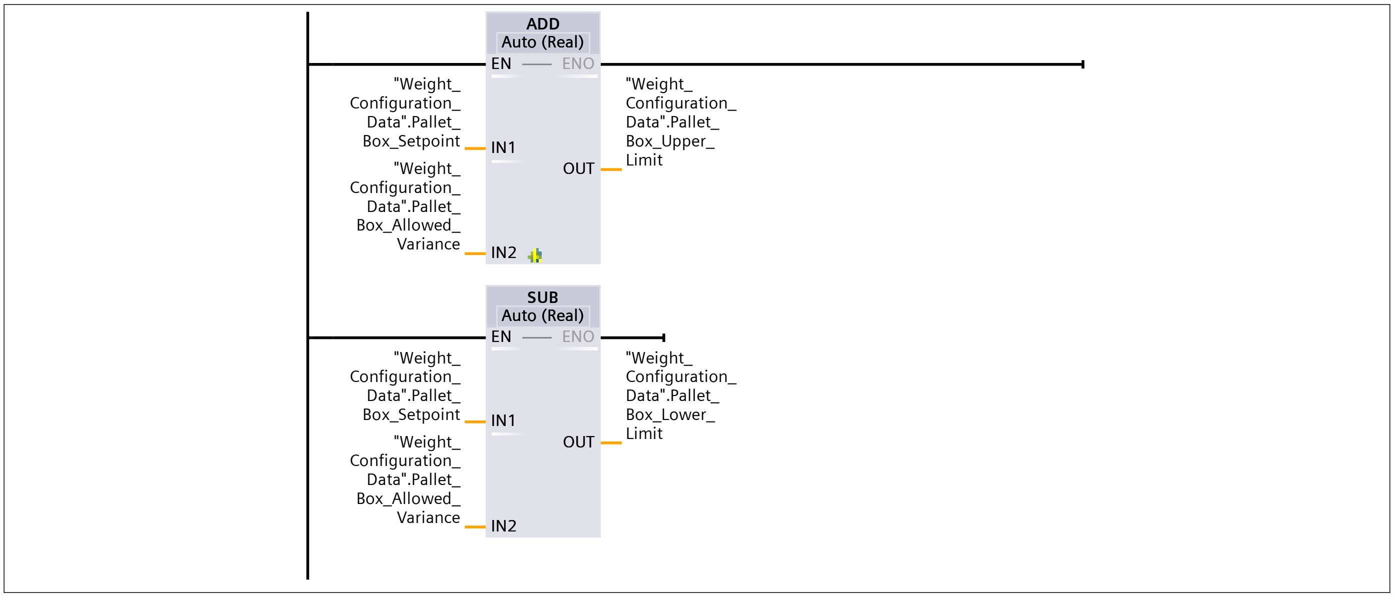
Name	Data type	Start value	Retain
▼ Static			
Pallet_Box_Setpoint	Real	3.0	True
Pallet_Box_Allowed_Variance	Real	0.1	True
Pallet_Box_Upper_Limit	Real	3.1	True
Pallet_Box_Lower_Limit	Real	2.9	True
Medium_Box_Setpoint	Real	8.0	True
Medium_Box_Allowed_Variance	Real	0.1	True
Medium_Box_Upper_Limit	Real	8.1	True
Medium_Box_Lower_Limit	Real	7.9	True
Heavy_Box_Setpoint	Real	10.0	True
Heavy _Box_Allowed_Variance	Real	0.1	True
Heavy_Box_Upper_Limit	Real	10.1	True
Heavy_Box_Lower_Limit	Real	9.9	True

Program blocks

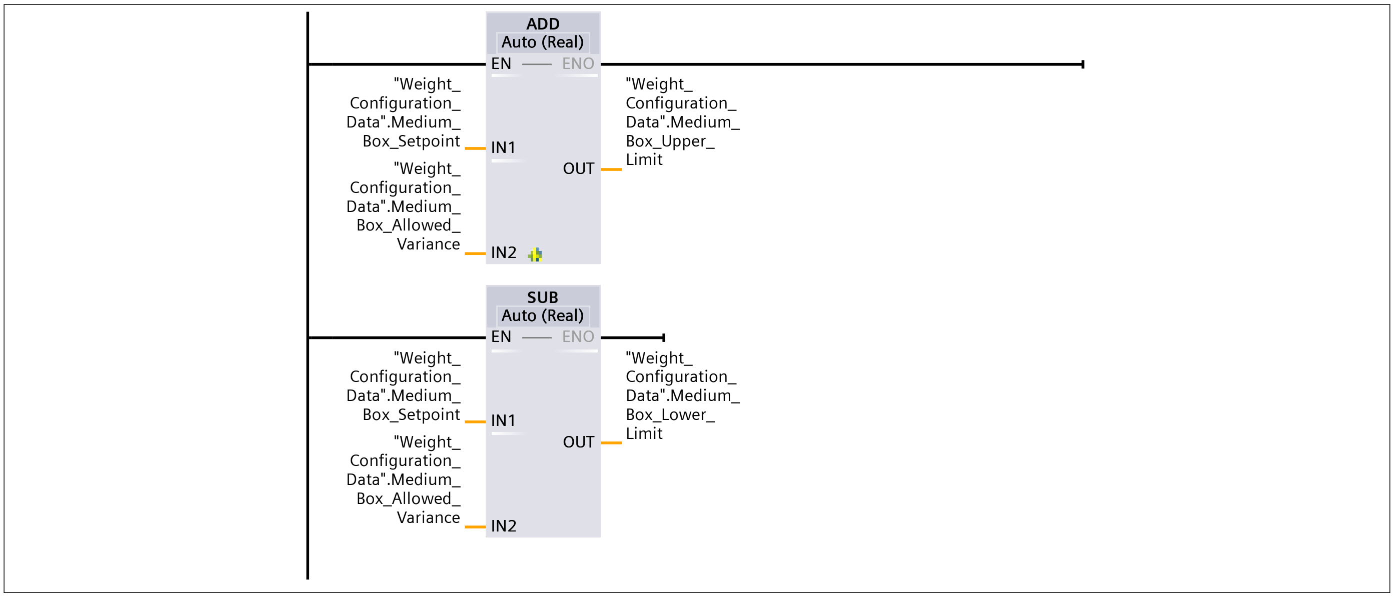
Weight_Config [FC5]

Weight_Config Properties					
General					
Name	Weight_Config	**Number**	5	**Type**	FC
Language	LAD	**Numbering**	Automatic		
Information					
Title	Weight Configuration Program	**Author**		**Comment**	Set weight configuration for scale.
Family		**Version**	0.1	**User-defined ID**	

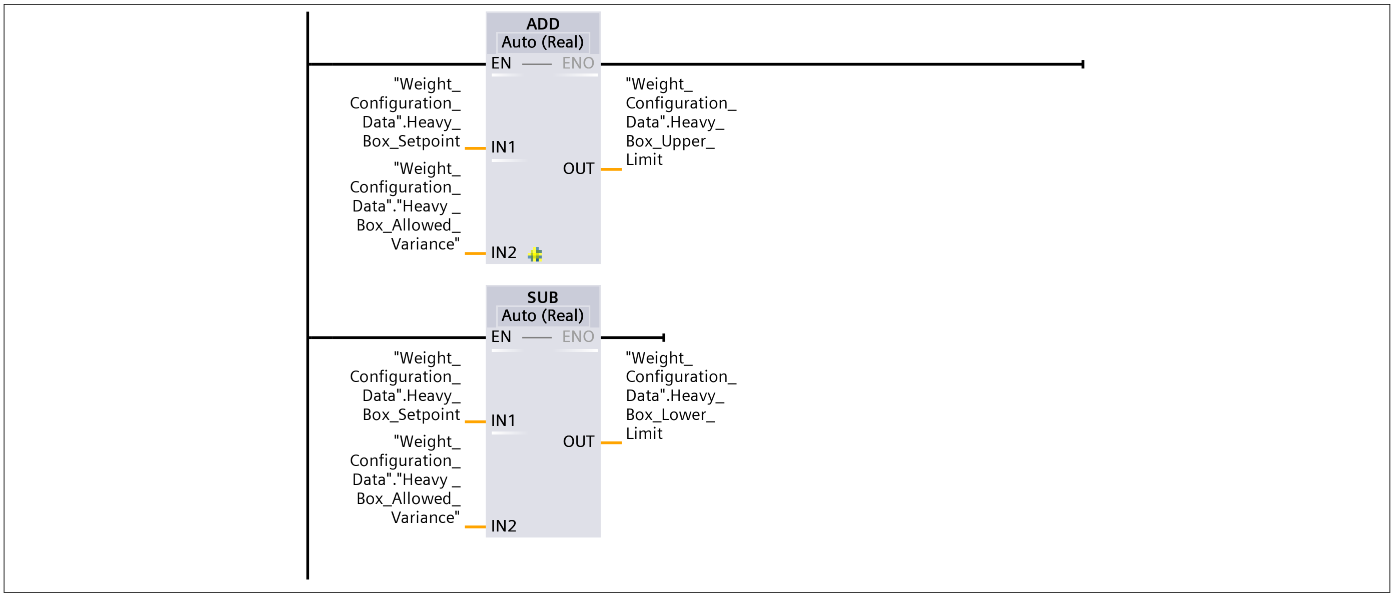
Network 1: Pallet Box Setpoint and Upper and Lower Limits


ADD
Auto (Real)
EN
ENO
"Weight_Configuration_Data".Pallet_Box_Setpoint
IN1
OUT
"Weight_Configuration_Data".Pallet_Box_Upper_Limit
"Weight_Configuration_Data".Pallet_Box_Allowed_Variance
IN2
SUB
Auto (Real)
EN
ENO
"Weight_Configuration_Data".Pallet_Box_Setpoint
IN1
OUT
"Weight_Configuration_Data".Pallet_Box_Lower_Limit
"Weight_Configuration_Data".Pallet_Box_Allowed_Variance
IN2

Network 2: Small Box Setpoint and Upper and Lower Limits


ADD
Auto (Real)
EN
ENO
"Weight_ Configuration_ Data".Medium_ Box_Setpoint
IN1
OUT
"Weight_ Configuration_ Data".Medium_ Box_Upper_ Limit
"Weight_ Configuration_ Data".Medium_ Box_Allowed_ Variance
IN2
SUB
Auto (Real)
EN
ENO
"Weight_ Configuration_ Data".Medium_ Box_Setpoint
IN1
OUT
"Weight_ Configuration_ Data".Medium_ Box_Lower_ Limit
"Weight_ Configuration_ Data".Medium_ Box_Allowed_ Variance
IN2

Network 3: Medium Box Setpoint and Upper and Lower Limits


ADD
Auto (Real)
EN
ENO
"Weight_ Configuration_ Data".Heavy_ Box_Setpoint
IN1
OUT
"Weight_ Configuration_ Data".Heavy_ Box_Upper_ Limit
"Weight_ Configuration_ Data"."Heavy _ Box_Allowed_ Variance"
IN2
SUB
Auto (Real)
EN
ENO
"Weight_ Configuration_ Data".Heavy_ Box_Setpoint
IN1
OUT
"Weight_ Configuration_ Data".Heavy_ Box_Lower_ Limit
"Weight_ Configuration_ Data"."Heavy _ Box_Allowed_ Variance"
IN2

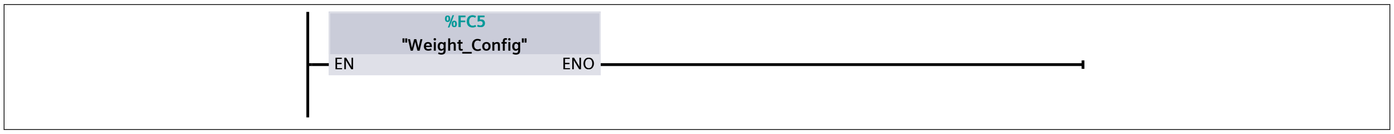
Program blocks

Startup [OB100]

Startup Properties					
General					
Name	Startup	**Number**	100	**Type**	OB
Language	LAD	**Numbering**	Automatic		
Information					
Title	Startup	**Author**		**Comment**	Call any setup block when swtiching from Stop to Run.
Family		**Version**	0.1	**User-defined ID**	

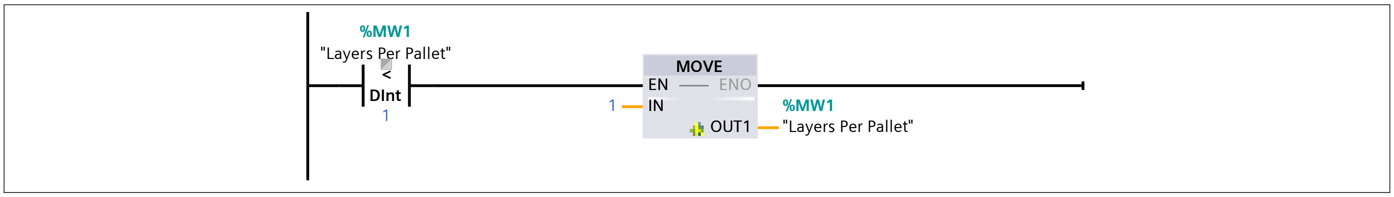
Network 1:

Recalculate weights when the controller is switch from Stop to Run. Allow new data to be entered by manually editing the weight datablock and then cycling the PLC.



Network 2: Initialize number of layers per pallet

The number of layers per pallet can not be less than 1. Although MW1 is set as a retentative word, if the PLC is reset or reloaded, this value may end of being equal to zero and initialization is required.
It should be noted that the weight data is contained in a datablock with intial values preset, which would be a better method of ensuring startup values are correct.


%MW1
"Layers Per Pallet"
<
DInt
1
MOVE
EN
ENO
1
IN
OUT1
%MW1
"Layers Per Pallet"