ME 308 INDUSTRIAL CONTROL PROJECT REPORT

SIEMENS PLC AND BECKHOFF INDUSTRIAL PC PROJECTS

SPRING 2019

GROUP 6

Instructor

Kemalettin Erbatur

Group Members

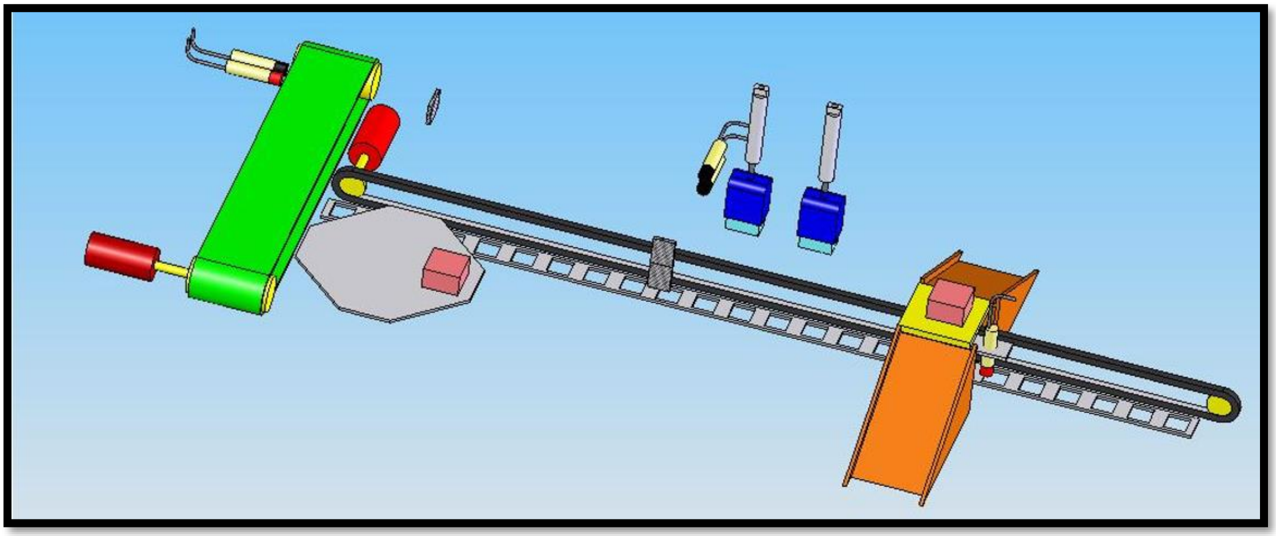
Cagatay Yagiz Uysal

Mustafa Gokturk Yandim 24134

Sarp Konrapa

Problem Definition

Part 1



1. Wooden or Metal boxes are placed at the end of the conveyor belt
2. Wooden boxes are loaded to chain platform
3. Pile tall wooden boxes down the wedge on the left
4. Pile short wooden boxes down the wedge on the right
5. Load metal boxes to the chain platform
6. Stamp tall metal boxes (press for 1 seconds, stamp four times). Move them to the conveyor belt and to the turn table. Turn the table 90 degrees and pile them there.
7. Short metal boxes should be stamped (press for 2 seconds). Move them to the conveyor belt and drop them from the front end of the conveyor belt.

Part 2

Use the Beckhoff PLC system program to do the following:

Use the two push buttons in a tapping fashion to turn the motor counterclockwise and clockwise:
Press and keep push button 1 and the motor should turn counterclockwise.
Release push button 1 and the motor should stop.
Likewise, push button 2 should command in a tapping mode the clockwise motion.

Count and display the inductive switch crossings of the two inductive switches in two text boxes on the touch screen.

The lamp should shine when the second inductive switch detects the bolt for more than 4 seconds.

When the lock button is off, the first inductive switch's crossings should not be counted.

Add a soft button on the touch screen. When the soft button is clicked the counter numbers should be incremented by 10.

Solution Proposal

Part 1

- Three sensors will be put at the end of the conveyor belt (2 optic and 1 metal sensor). Optic sensors will be used for Box-In and for detecting taller boxes. Metal sensor detects metal boxes.
- Conveyor belt will start running after detecting signal from Box-In sensor. There will be another sensor on the conveyor belt; which is placed before chain platform. When a box passes this sensor, conveyor will stop, and a piston will push the box on the chain platform.
- After the box is pushed on the chain platform; platform will start to move. If box is metal, it will stop at the stamp station and depending on their size, different stamp procedure will be applied (See problem definition for stamp operation requirements). However, if the box is wood, it will not stop at the stamp station and continue to move until ramps at the end of the chain rail. Depending on the size of the box, they will be pushed down from one of the ramps. There is a metal sensor on the chain platform cart, and it is connected to a long metal plate with square holes on it. We will count number of on to off changes in sensor output to stop the cart at the desired position (ie at the stamp station or ramp).
- Tall metal boxes will be pushed to the belt with a piston after stamp application. Then it will be pushed to the rotating belt, it will be dropped after 90 degree of rotation.
- Short metal boxes will be piled at the end of the conveyor belt.

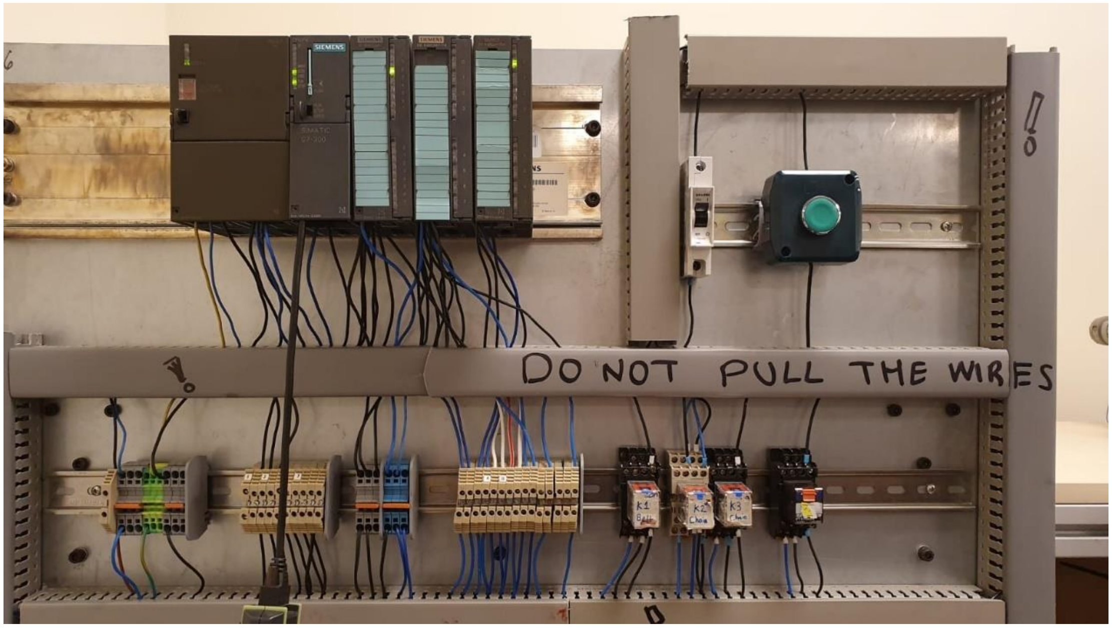
Part 2

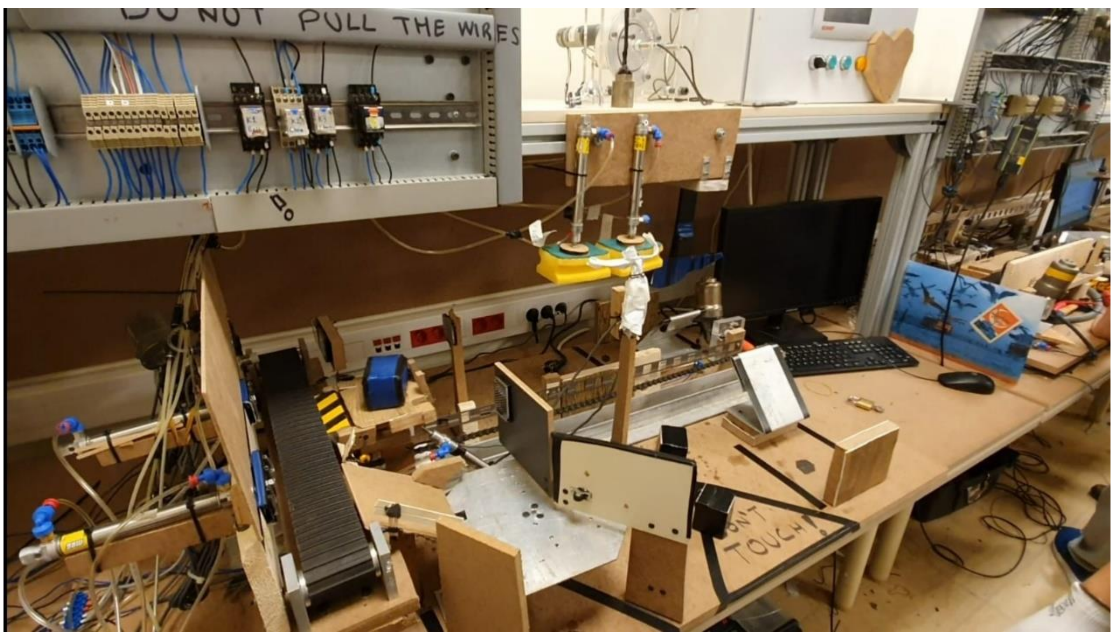
- We can solve the given problem definition using ST programming is TwinCat PLC Control system.
- We will use timer for 4 seconds counting.
- Negative and falling edge detectors will be used for switch crossings and counting's.

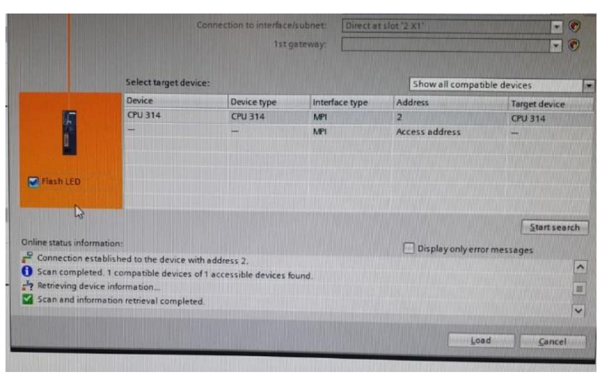
Wiring Diagram

Wiring diagram is attached at the end of the report.

Photos & Snapshots







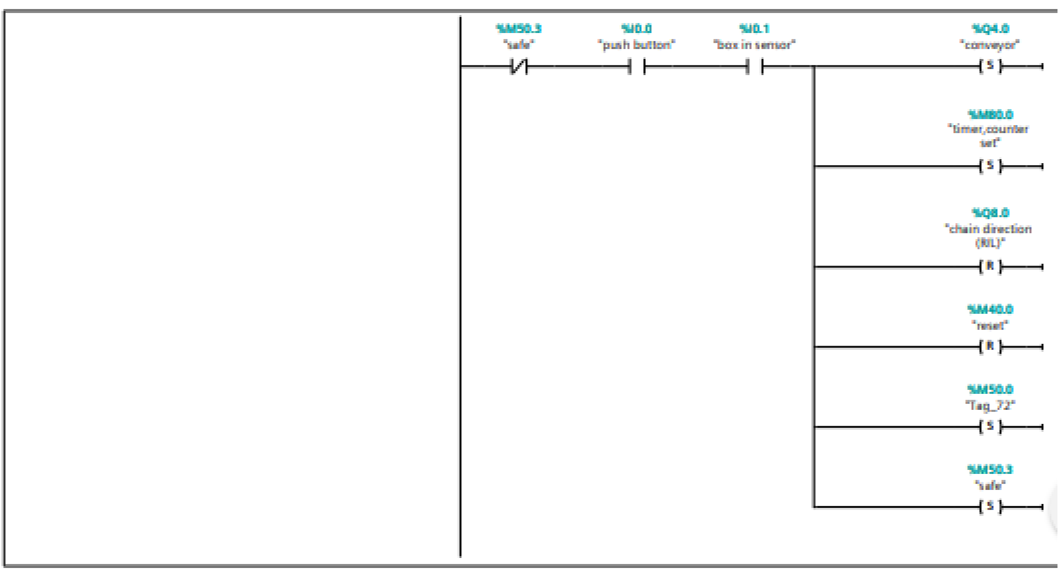
Symbol Table

Symbol	Address	Symbol	Address
Start Button	I0.0	Belt on/off	Q4.0
Box in Sensor	I0.1	Belt to Cart Piston	Q4.1
Rail counting sensor on the cart	I0.2	Wood distributing Piston	Q4.2
Metal Check Sensor	I0.3	Chain on/off	Q4.3
Belt to Chain Sensor	I0.4	First Stamp	Q4.4
Tall Check Sensor	I0.5	Second Stamp	Q4.5
Turn Table in Sensor	I0.7	Cart to Belt Piston	Q4.6
		Turn Table Piston	Q4.7
		Chain Direction (L/R)	Q8.0
		Turn Table on/off	Q8.1

Ladder Logic Networks:

Network 1:

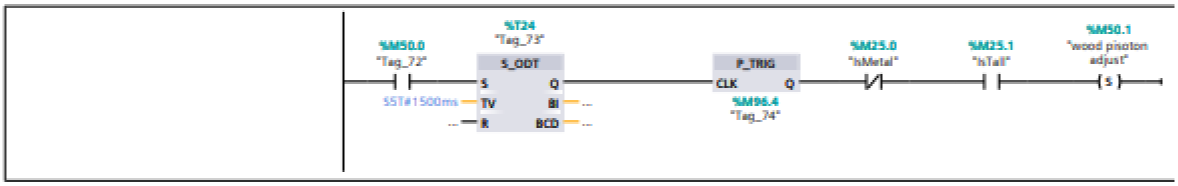
Push button and belt start



This network is the starting point of the system. The system runs only when box is placed in front of box in sensor and the starting button is pressed. If those conditions were met, conveyor belt starts and the box moves forward. By pressing the button, we also adjust some variables which we will use later such as: set of counters, chain direction (Right) and resetting memory. Also, we have a safe memory which will break the network when the box moves forward. So, during the process the system won't accept new boxes in the process.

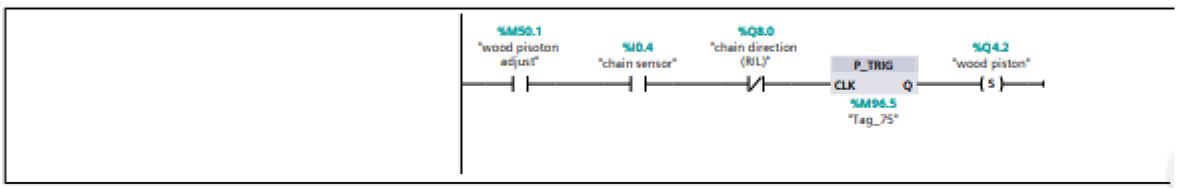
Network 2:

wood piston adjust



Network 3:

wood piston adjust2

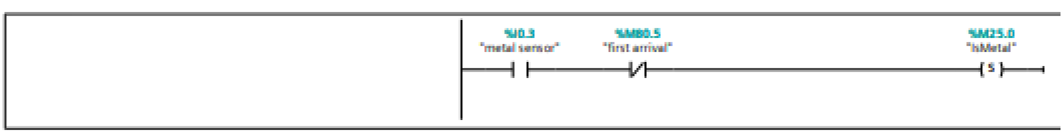


Network 2 and **Network 3** are for the adjustment of the wood distribution piston. M50.0 memory is activated from **Network 1** and system waits for 1.5 seconds. During the wait box is going through sensors on the belt. This time is enough for the system to gather required information about the boxes material and length. If the placed box is wood and tall, system energizes M50.1 and **Network 3** begins.

The logic behind Network 3 is the placement of the wood piston. If the memory M50.1 is activated, this means the incoming box is wood and tall. So, the network adjusts the piston when the box reaches chain sensor (I0.4). Normally closed chain direction input is there for preventing unnecessary activation when the box is coming back.

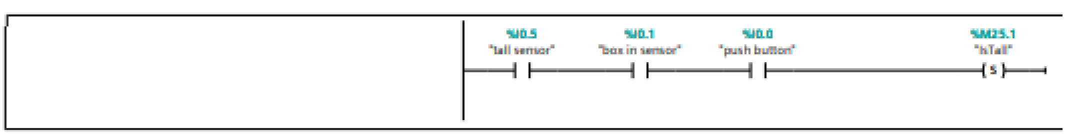
Network 4:

metal check



Network 5:

wood piston adjust and tall check



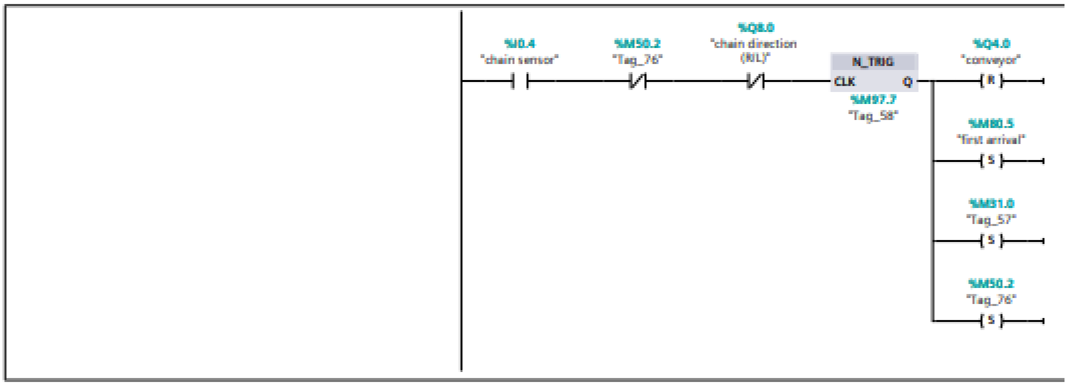
Network 4 and **Network 5** is for gathering information about the material and the length of the box.

Network 4 looks for the material of the box. If the box is metal activates the M25.0 and uses that information for the upcoming networks. M80.5 memory is for the setting of information. When the box is pushed into the cart, we cannot change the material type until the next process.

Network 5 looks for the length of the box. If both box in sensor (I0.1) and tall sensor (I0.5) is active, this means the box is long enough to cover them. So, network classifies the box as tall. This information is taken only at the beginning. After the button is pushed this information cannot be changed.

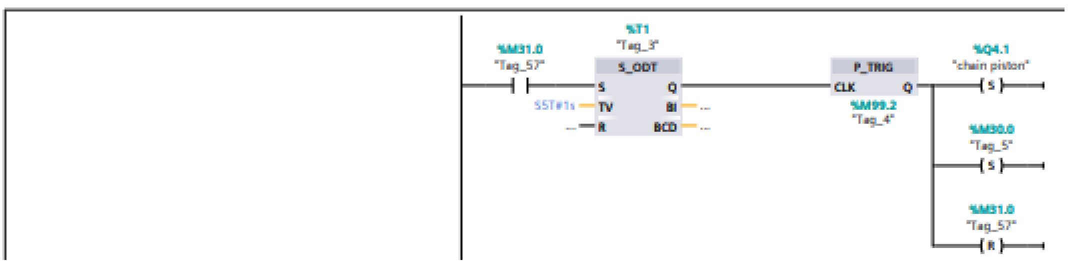
Network 6:

chain enterance



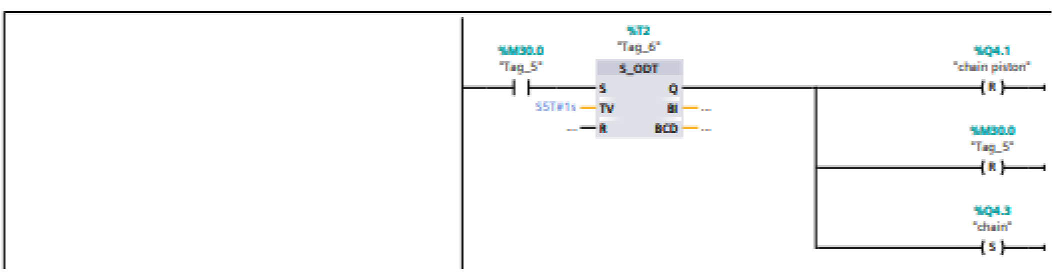
Network 7:

chain enterance contd



Network 8:

chain pison off



Network 6 and **Network 7** is for the transfer of the box, from belt to the cart. **Network 8** is for the adjustment of the piston that pushes the box to the cart (Q4.1) and starts the chain. These networks are for all four box types.

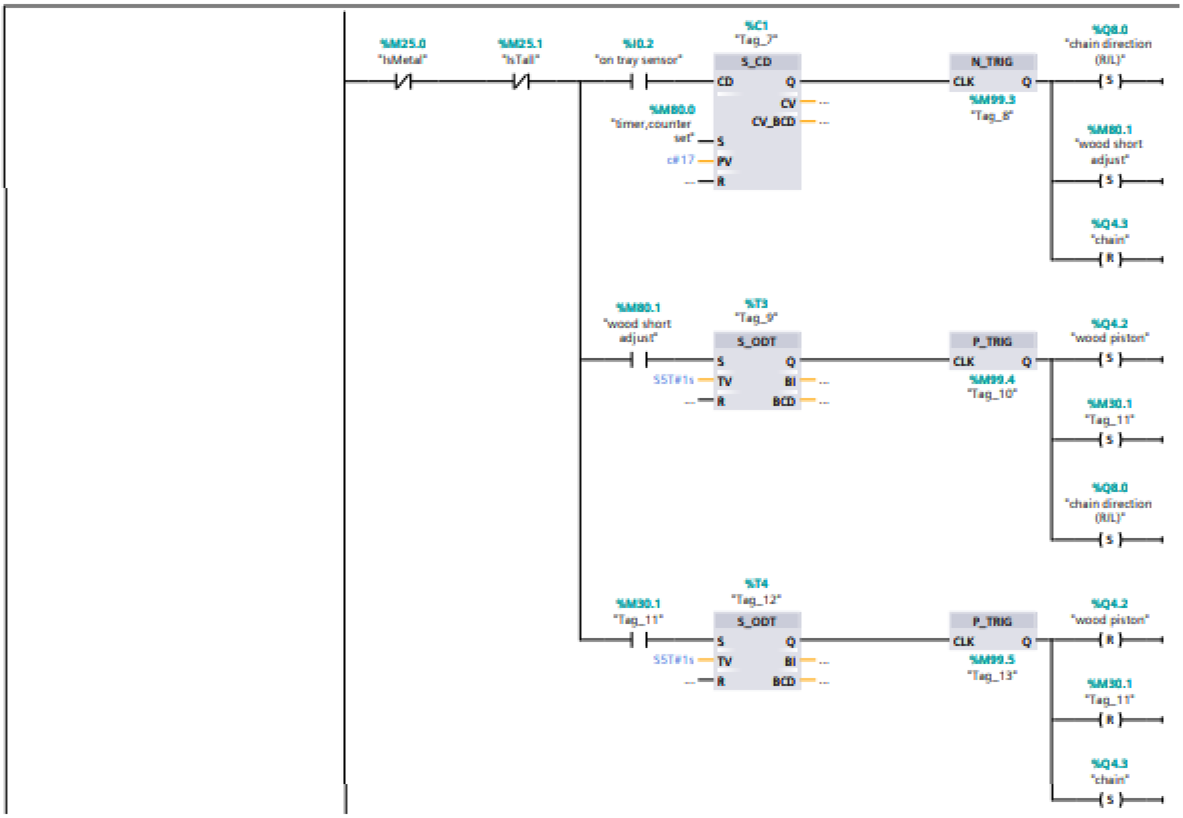
Network 6 activates when the box arrives to the chain sensor (I0.4). After box leaves the sensor belt stops and declares its arrival by setting M80.5. Also, to avoid unnecessary activation M50.2 breaks the network until the process ends. At the end of this network sets M31.0, which will start **Network 7**.

Network 7 waits for 1 second and after that wait activates the piston that pushes the box to the cart (Q4.1). Also, M31.0 resets itself so it is reusable.

Network 8 waits for 1 second after the activation of M30.0 from **Network 7**, then takes piston back and starts the chain to move boxes to desired locations. Also, M30.0 resets itself so it is reusable.

Network 9:

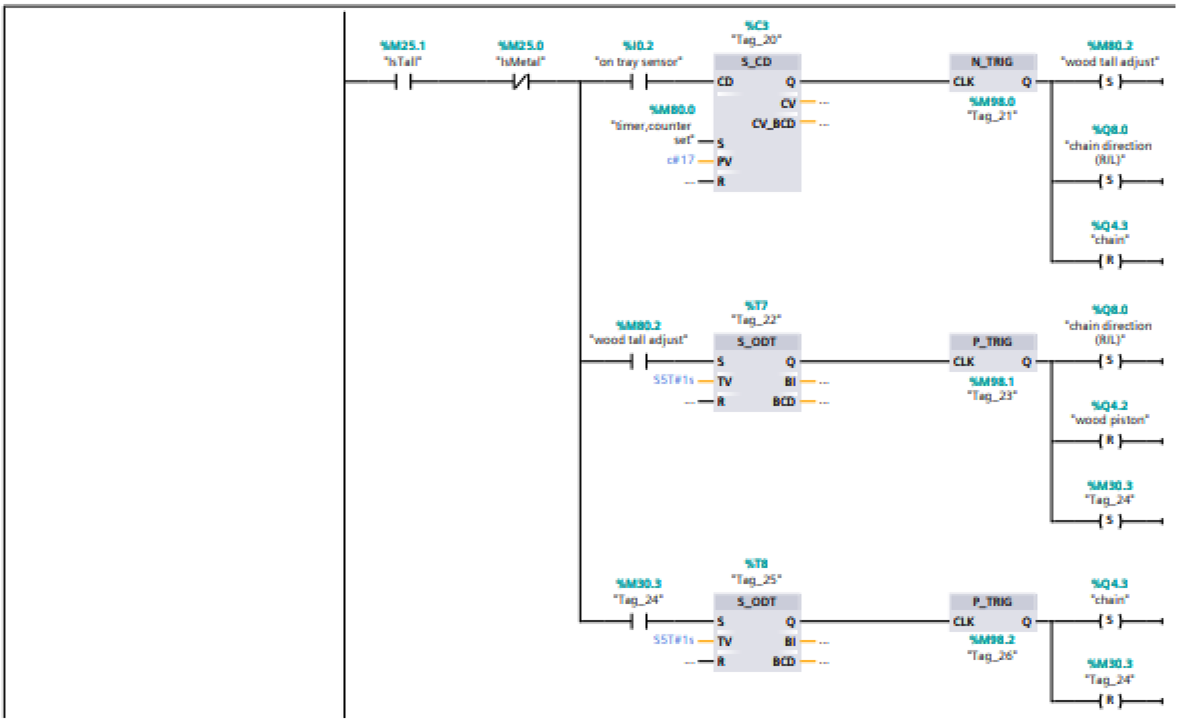
wood short



This network is the place which manages the process for the short wood boxes. On tray sensor (I0.2) counts the rail to detect the desired stop, for this process 17 metal counts are needed. When the cart reaches its destination, chain (Q4.3) stops and its direction is changes for the upcoming action. M80.1 activates the second branch, which waits for 1 second and activates the wood piston (Q4.2). This piston will push the short wood box to the desired location and the process is completed. M30.1 activates the third branch, which waits for 1 second and takes back the wood piston to its initial position. After everything is done chain starts again and takes the cart back to the its initial position. This action will be explained in **Network 13**.

Network 10:

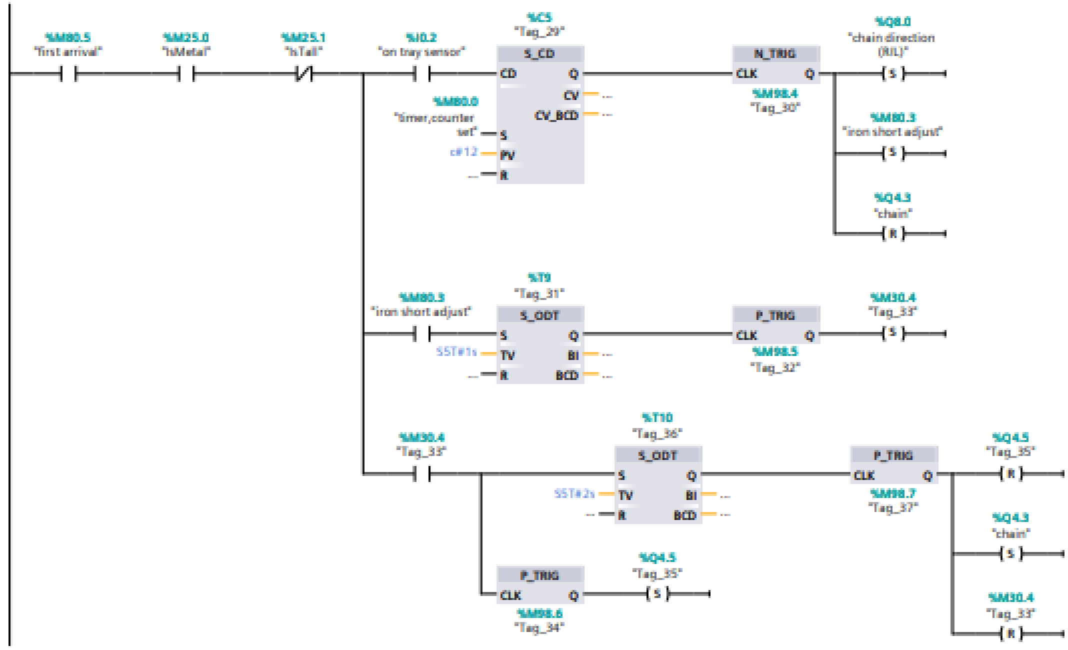
wood tall



This network is the place which manages the process for the long wood boxes. On tray sensor (I0.2) counts the rail to detect the desired stop, for this process 17 metal counts are needed. When the cart reaches its destination, chain (Q4.3) stops and its direction is changes for the upcoming action. M80.2 activates the second branch, which waits for 1 second and deactivates the wood piston (Q4.2). This piston will push the long wood box to the desired location and the process is completed. M30.3 activates the third branch, which waits for 1 second and takes the cart back to the its initial position. This action will be explained in **Network 13**.

Network 11:

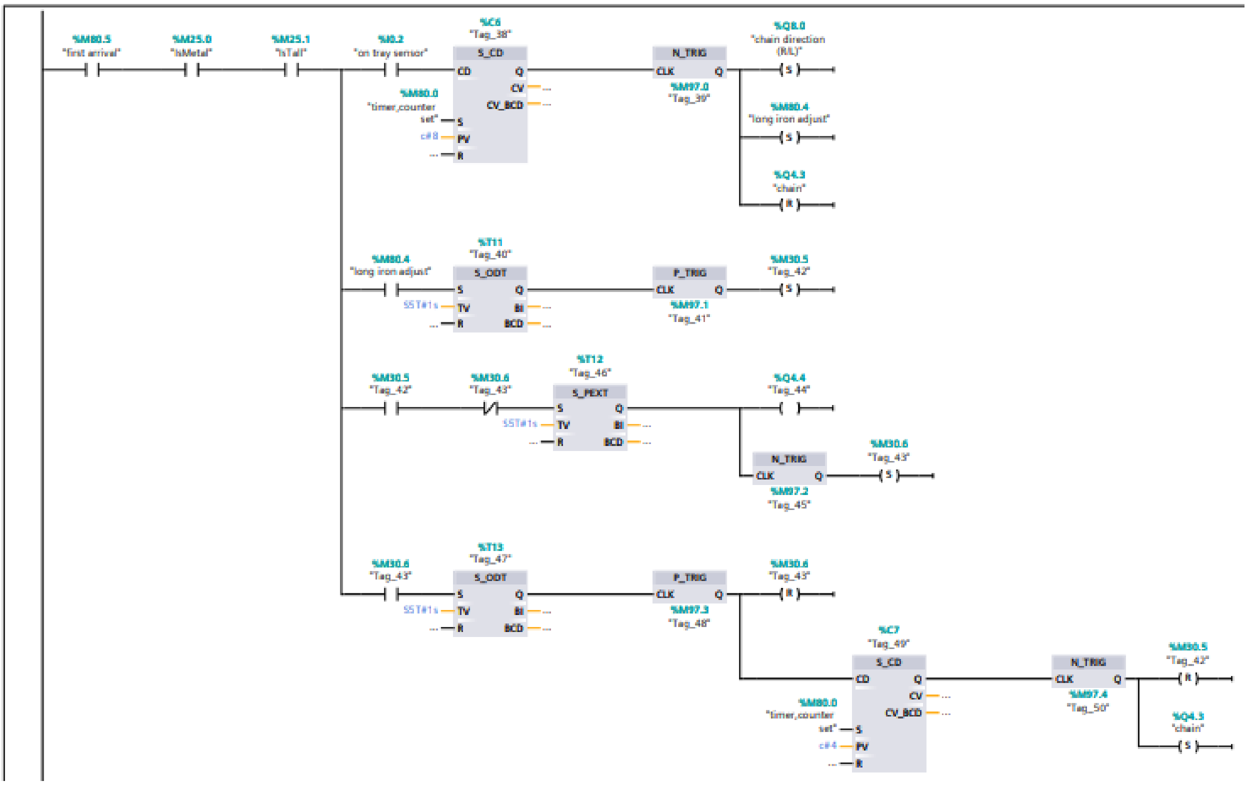
iron short



This network is the place which manages the process for the short metal boxes. On tray sensor (I0.2) counts the rail to detect the desired stop, for this process 12 metal counts are needed. When the cart reaches its destination, chain (Q4.3) stops and its direction is changes for the upcoming action. M80.3 activates the second branch, which waits for 1 second and activates M30.4. This activates the third branch. When the third branch starts, second stamp (Q4.5) activates. After 2 seconds stamping is completed and stamp deactivates. Also, M30.4 resets itself so it is reusable. Afterwards chain starts again and takes the cart back. Transfer of the short metal box to the belt will be explained in **Network 14**.

Network 12:

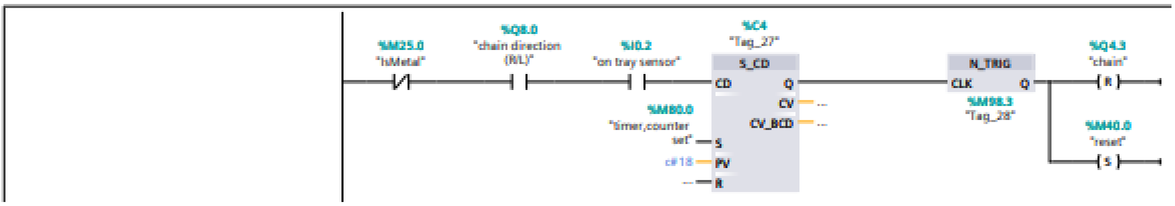
iron tall



This network is the place which manages the process for the long metal boxes. On tray sensor (I0.2) counts the rail to detect the desired stop, for this process 8 metal counts are needed. When the cart reaches its destination, chain (Q4.3) stops and its direction is changes for the upcoming action. M80.4 activates the second branch, which waits for 1 second and activates M30.5. This activates the third branch. When the third branch starts, first stamp (Q4.4) activates. After 1 second M30.6 sets, this unlocks the fourth branch and breaks the third branch. So, stamp deactivates and cannot be activated. In fourth branch, the system waits for 1 second and sets M30.6. This creates a loop between third and fourth branch. Counter #7 counts those loops and when it reaches 4, system resets M30.5 and breaks third branch. This cuts the loop and stamping for 4 times is completed. Afterwards chain starts again and takes the cart back. Transfer of the short metal box to the belt will be explained in **Network 16**.

Network 13:

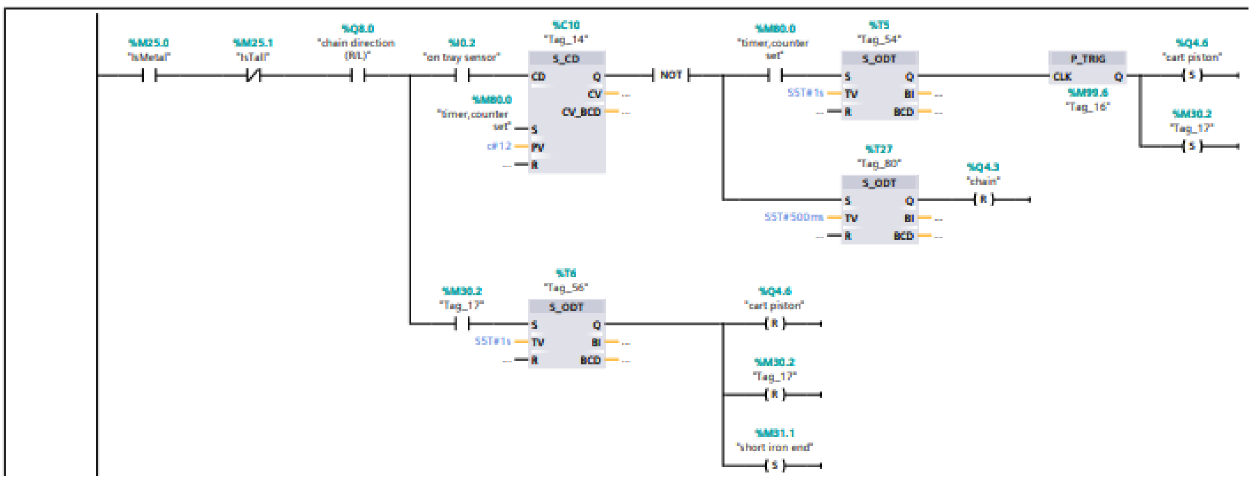
wood cart to conveyor



This network is the follow-up to both **Network 9** and **Network 10**. Since both wood processes ends at the same place, their return is also same. On tray sensor (I0.2) counts for 18 when the chain direction is left. When it reaches 18 chain stops and the whole process for tall and short wood box is ended. At the end, system activates reset network (M40.0) which will be explained in **Network 18**.

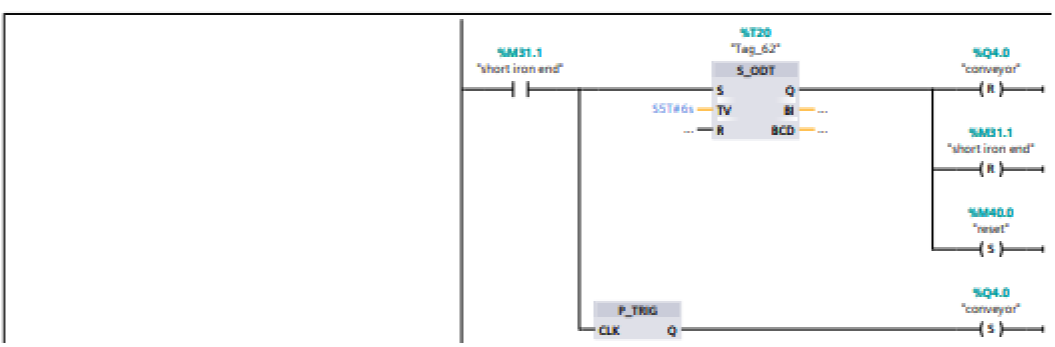
Network 14:

metal short cart to conveyor



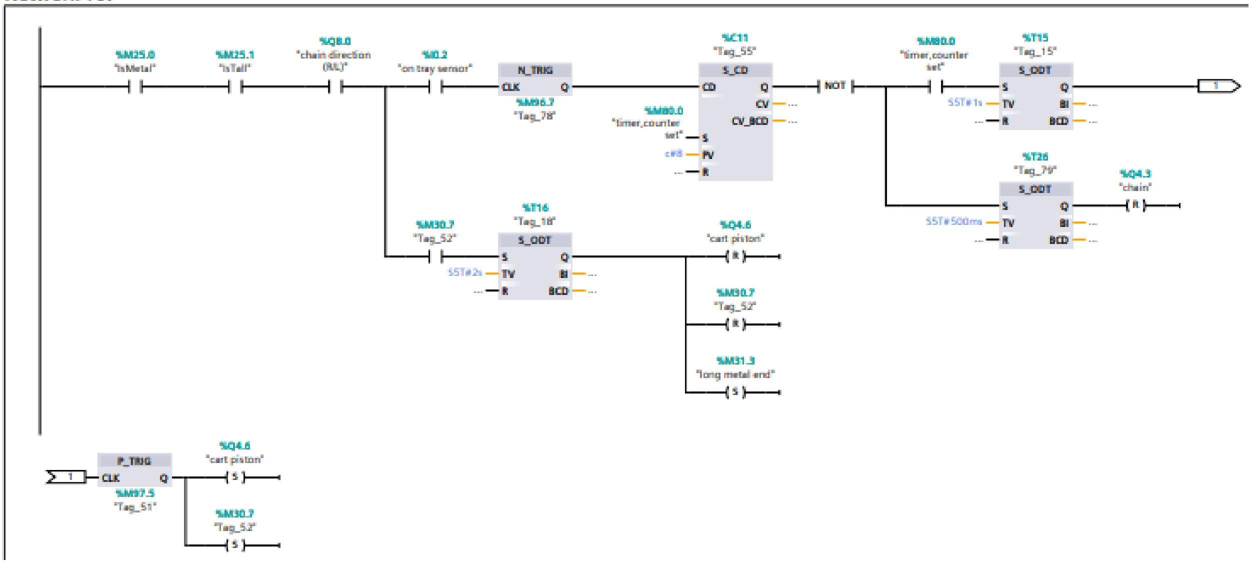
Network 15:

short metal contd



These networks are follow-up to **Network 11**. On tray sensor (I0.2) counts for 12 when the chain direction is left. When it reaches 12 chain stops. After 1 second, cart piston (Q4.6) and M30.2 activates. Cart piston transfers the box to the belt. M30.2 unlocks the second branch. In the second branch, system waits for 1 second and deactivates the cart piston. M31.1 activates **Network 15**. In Network 15, belt will be working for 6 seconds to send the box to desired location. At the end, system activates reset network (M40.0) which will be explained in **Network 18**.

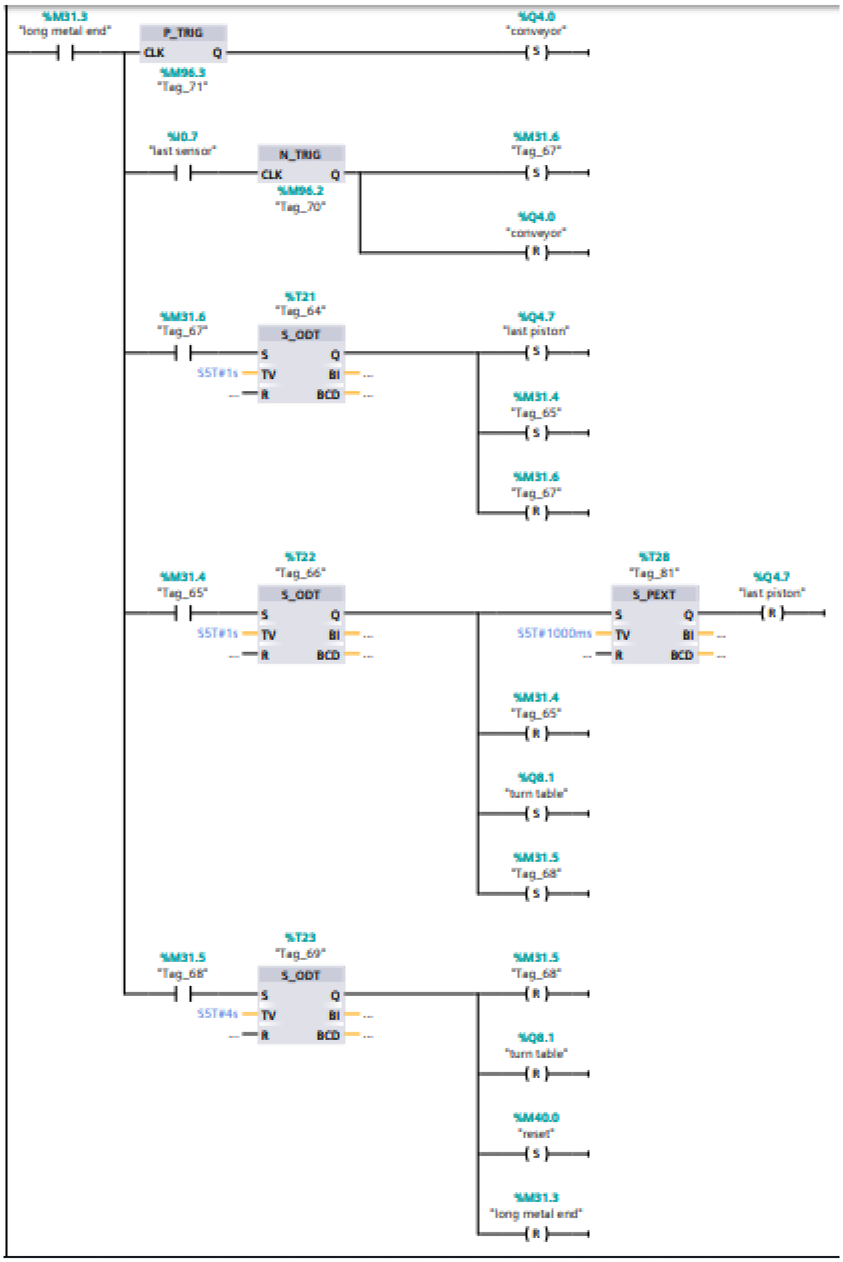
Network 16:



These networks are follow-up to **Network 12**. On tray sensor (I0.2) counts for 8 when the chain direction is left. When it reaches 8 chain stops. After 1 second, cart piston (Q4.6) and M30.7 activates. Cart piston transfers the box to the belt. M30.7 unlocks the second branch. In the second branch, system waits for 2 seconds and deactivates the cart piston. M31.3 activates **Network 17**.

Network 17:

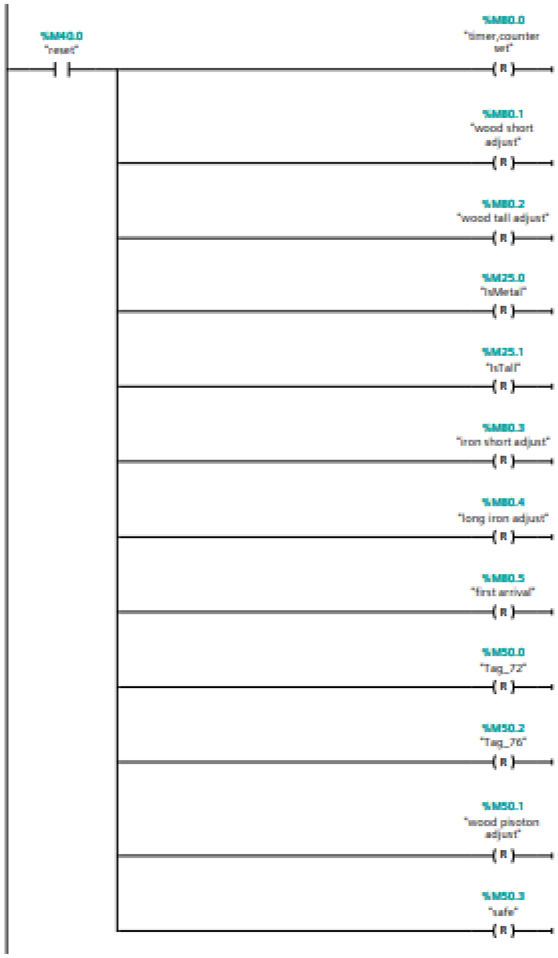
long metal contd



This network is the follow-up of **Network 16**. When it activates, belt starts moving. Belt will be on until the box passes from last sensor (I0.7). When it passes from the sensor, belt stops. As shown in the third branch, system waits for 1 second and pushes the box to the turning table by activating last piston (Q4.7). In the fourth branch, system waits for 1 second and resets the last piston. Also, starts the turning table. Turning table turns for 4 seconds and stops. At the end, system activates reset network (M40.0) which will be explained in **Network 18**.

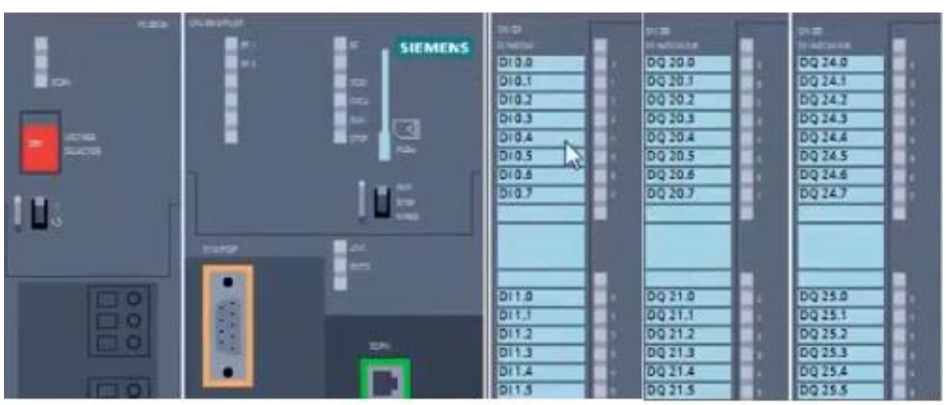
Network 18:

reset



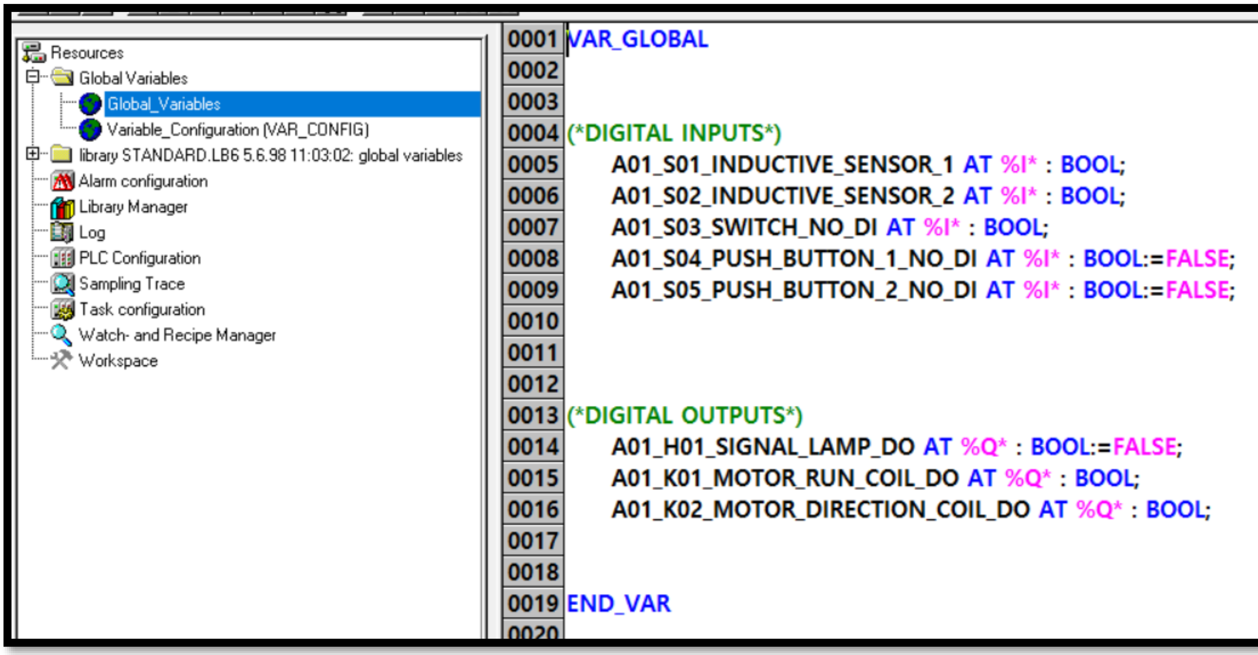
This network is the reset network. It resets all critical variables assigned during the processes, so that new process won't be affected from the previous one.

PLC Hardware Configuration

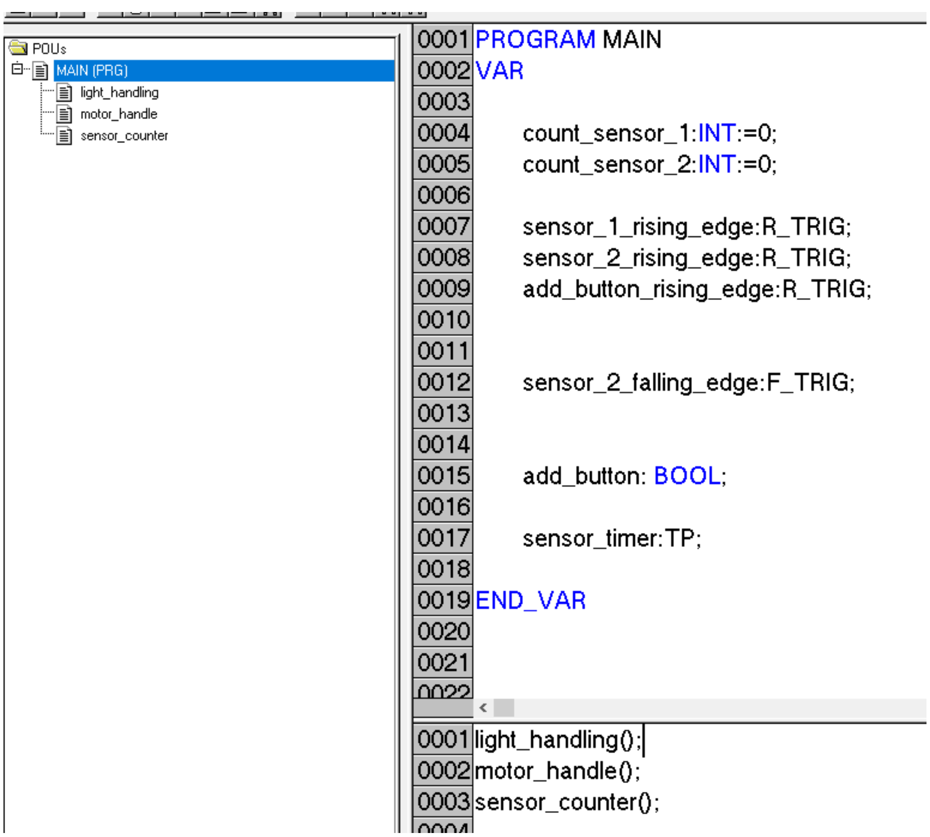


Industrial PC Code

Global Variables



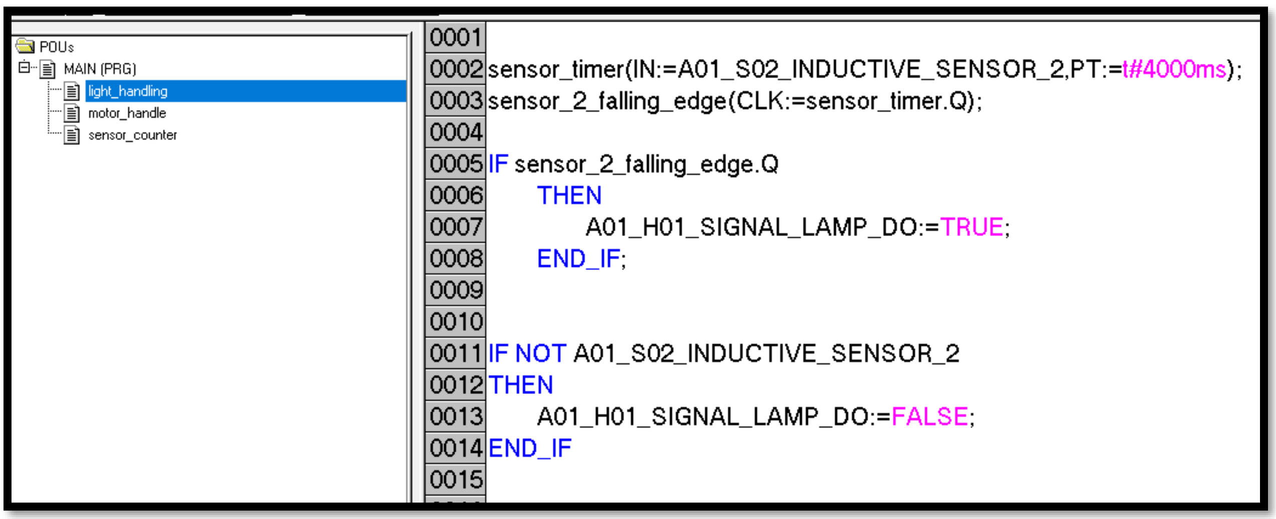
Program Main



- We have defined the input sensors (two Inductive sensors, a switch and two push buttons) inside the Global Variables Tab.
- We have defined the outputs (Signal lamp, Motor on-off relay and motor direction relay) inside the Global Variables Tab.
- We also defined some variables (Integer, rising edge trigger, Falling edge trigger, Bool and Pulse Timer) inside the Program Main Tab.

Light_handling

This part controls the light operation.



At 0002: sensor_timer is a Pulse Timer and it starts when Inductive sensor 2 detects the bolt; It counts for 4 seconds.

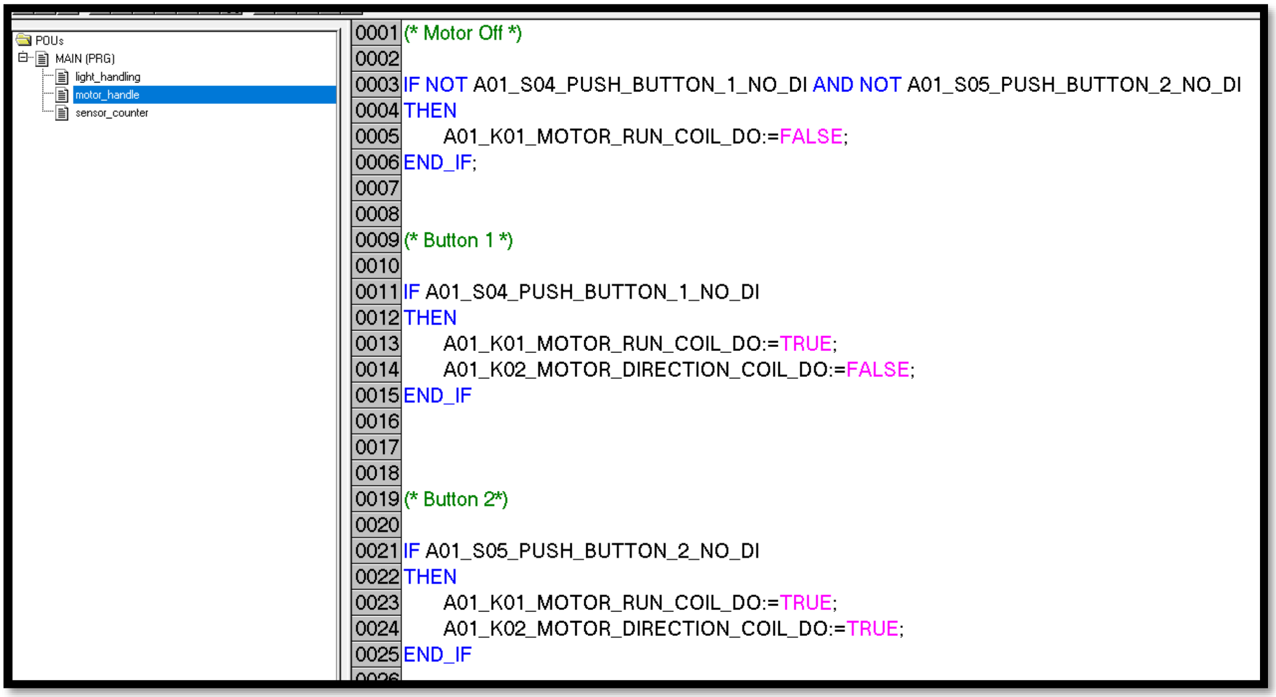
At 0003: sensor_2_falling_edge becomes TRUE when 4 seconds passes in Pulse Timer (Pulse Timer gives “1” output while it is counting and gives “0” when counting ends).

At 0005: When a faliing edge is detected, program goes into the IF condition and lights the signal lamp.

At 0011: If bolt is in front of the inductive sensor, If condition sets the signal lamp to FALSE; it stops shining.

Motor_handle

This part controls motor on/off and rotation direction.




```
0001 (* Motor Off *)
0002
0003 IF NOT A01_S04_PUSH_BUTTON_1_NO_DI AND NOT A01_S05_PUSH_BUTTON_2_NO_DI
0004 THEN
0005       A01_K01_MOTOR_RUN_COIL_DO:=FALSE;
0006 END_IF;
0007
0008
0009 (* Button 1 *)
0010
0011 IF A01_S04_PUSH_BUTTON_1_NO_DI
0012 THEN
0013       A01_K01_MOTOR_RUN_COIL_DO:=TRUE;
0014       A01_K02_MOTOR_DIRECTION_COIL_DO:=FALSE;
0015 END_IF
0016
0017
0018
0019 (* Button 2*)
0020
0021 IF A01_S05_PUSH_BUTTON_2_NO_DI
0022 THEN
0023       A01_K01_MOTOR_RUN_COIL_DO:=TRUE;
0024       A01_K02_MOTOR_DIRECTION_COIL_DO:=TRUE;
0025 END_IF
```

At 0003: Program enters to the IF condition if none of the push buttons are pressed; then motor on-off relay is set to FALSE (Motor is inactive).

At 0011: If push button 1 is pressed, Motor is enabled, and direction is set to FALSE (=CCW direction).

At 0021: If push button 2 is pressed, Motor is enabled, and direction is set to TRUE (=CW direction).

Therefore, Motor is working only when user presses and holds any of the push buttons. Otherwise, First IF condition works and sets the motor to off.

Sensor_Counter

Our problem definition suggests that the screen on the Beckhoff should display the number of crossings for each inductive sensor. Also, we should use a soft switch on the screen. When clicked, it should increment the number of crossings counters by 10 (For both inductive switches). Lastly, first inductive switch should count the crossings when lock switch is at the on position.

```
POUs
  MAIN (PRG)
    light_handling
    motor_handle
    sensor_counter
```

```
0001 (*Sensor 1*)
0002 IF A01_S03_SWITCH_NO_DI
0003 THEN
0004     sensor_1_rising_edge(CLK:=A01_S01_INDUCTIVE_SENSOR_1);
0005     IF sensor_1_rising_edge.Q
0006     THEN
0007         count_sensor_1:= count_sensor_1+1;
0008     END_IF;
0009 END_IF;
0010
0011
0012 (*Sensor 2*)
0013 sensor_2_rising_edge(CLK:=A01_S02_INDUCTIVE_SENSOR_2);
0014 IF sensor_2_rising_edge.Q
0015 THEN
0016     count_sensor_2:= count_sensor_2+1;
0017 END_IF;
0018
0019
0020 (*Add button*)
0021 add_button_rising_edge(CLK:=add_button);
0022
0023 IF add_button_rising_edge.Q
0024 THEN
0025     count_sensor_1:= count_sensor_1+10;
0026     count_sensor_2:= count_sensor_2+10;
0027 END_IF;
0028
0029
```

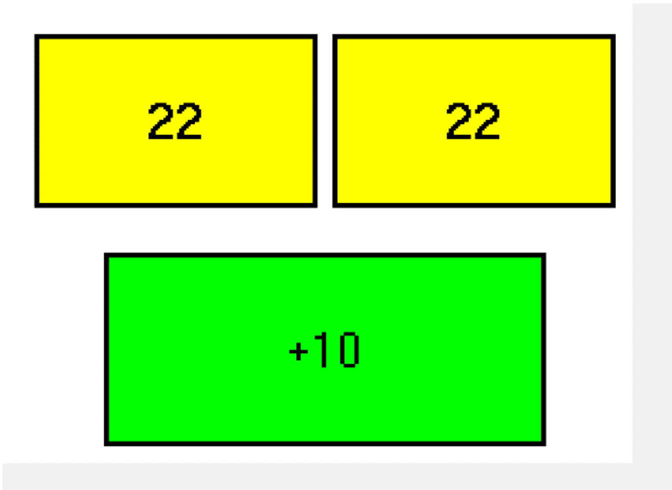
At 0002: If lock switch is at the on position, program goes into first IF condition. Sensor_1_rising_edge detects a rising edge at the first inductive sensor (**At 0004**) and becomes "1" if signal (bolt) is detected. If it detects a rising edge, count_sensor_1 increase by 1 and if condition ends (**At 0007**).

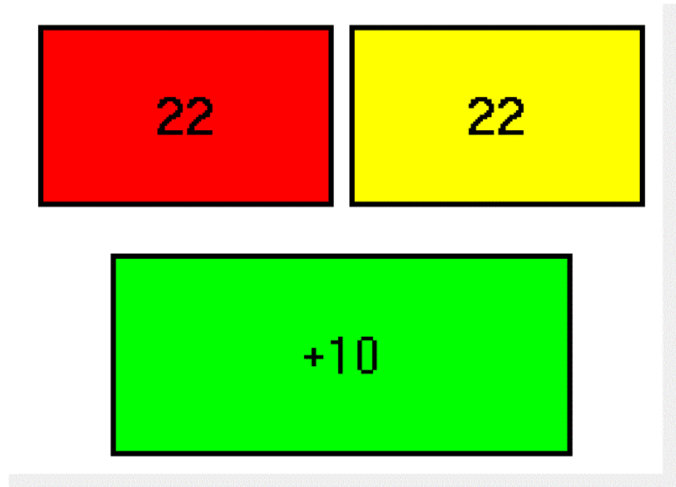
At 0012: sensor_2_rising_edge detects at the inductive sensor 2. If it detects a positive edge, count_sensor_2 increase by 1.

At 0021: add_button_rising_edge detects a signal if soft button on the Beckhoff screen is clicked (we called it as add-button). If there is a rising edge (soft button is clicked), both counters for inductive sensors gets increased by 10.

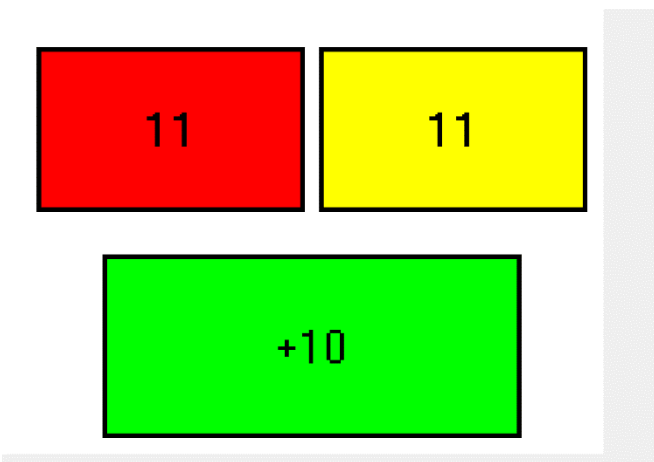
Industrial PC Touch Panel

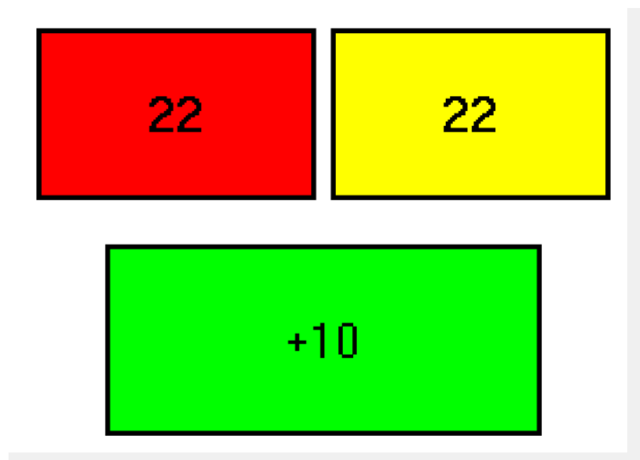
Counter box turns to red when sensor detects the Bolt.

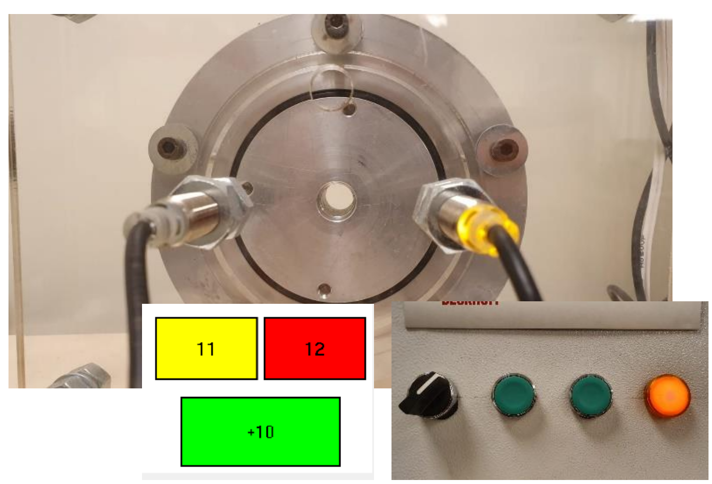


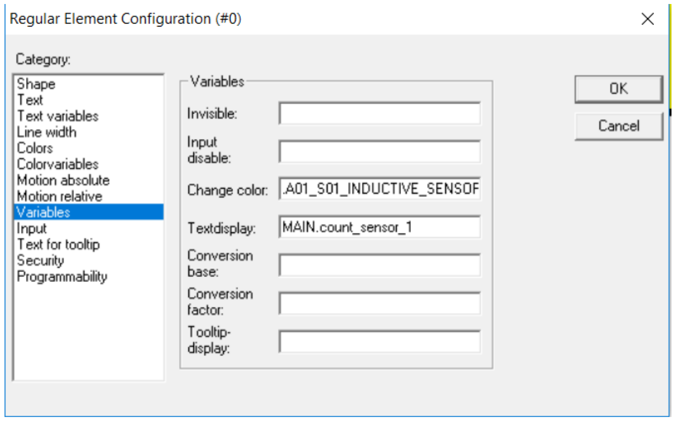


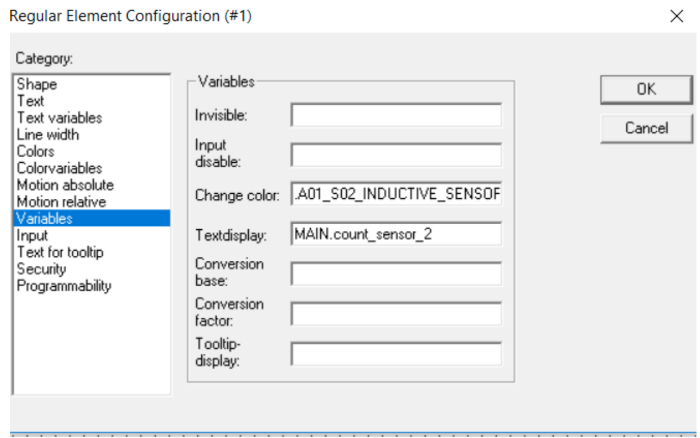
Counters increase by 10 when pressed to +10 soft button

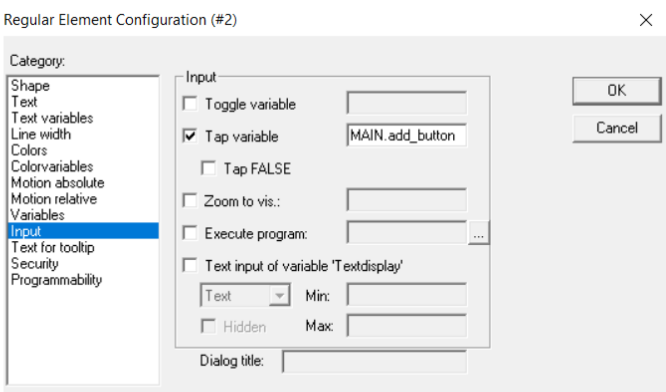












Element 0 = Counter 1 (for switch on the left)

Element 1 = Counter 2 (for switch on the right)

Element 2 = Add Button operation (+10).

Discussion

We built our system as a most effective and in cool fashion. While we are doing it, we did not use piston on the cart to move the boxes from cart to belt back. We made a rotating part on the cart and we insert a piston on the ground and made it stable. When we need to move boxes from cart, piston works, and part turns then boxes are put on the belt

We had some difficulties while finding necessary part or tools. But Luckily, one of our friends had access to the mechatronics lab that we were able to use tools and parts in there.

We built our connections by ourselves, at first, we have used some already existing cables, but their ferrules were not stable.