ME 308

- Industrial Control -

Term Project Report

Group 1

Sabanci University
Faculty of Engineering and Natural Sciences

This document is Project Report for ME 308 Industrial Control course.
All figures are captured by the Siemens Simatic Manager, Beckhoff Twincat 2 and the plants.

Mechatronics Engineering
Sabanci University
https://me.sabanciuniv.edu/en

Title:
PLC Application on Production Line

Project Period:
Fall Semester 2019

Project Group:
Group 1

Participant(s):
Ahmet Guclu 17494
Demir Demirel 20586
Edin Guso 23435
Emre Erdinc 20809
Gunes Basak Ozgun 23521

Instructor(s):
Kemalettin Erbatur

Supervisor(s):
Gokhan Calbaz
Kartal Ucar
Tibet İskesen

Copies: 1

Page Numbers: 58

Date of Completion:
January 3, 2019

Abstract:

The project based on Siemens S7-300 family PLC control and Beckhoff Industrial PC implementations.

Sabancı Üniversitesi

Mekatronik Mühendisliği
Sabancı Üniversitesi
https://me.sabanciuniv.edu/

Başlık:
Üretim Bandları için PLC Uygulamaları

Proje Periyodu:
2019 Güz Dönemi

Proje Grubu:
Grup 1

Grup Üyeleri:
Ahmet Guclu 17494
Demir Demirel 20586
Edin Guso 23435
Emre Erdinc 20809
Gunes Basak Ozgun 23521

Öğretim Üyesi:
Kemalettin Erbatur

Asistanlar:
Gokhan Calbaz
Kartal Ucar
Tibet İskesen

Kopya: 1

Sayfa Sayısı: 58

Bitirme Tarihi:
January 3, 2019

Abstract:

Projemiz Siemens S7-300 ailesi PLC kontrolü ve Beckhoff Endüstriyel Bilgisayar uygulamalarından oluşmaktadır.

Contents

Preface **vii**

1 Problem Definition and Solution Proposal **1**
1.1 Problem Definition 1
1.2 Solution Proposal 2

2 Wiring Diagrams **3**
2.0.1 Title Page 4
2.0.2 Content 5
2.0.3 Main Connections And Power Distribution 6
2.0.4 CPU Power Connections 7
2.0.5 Digital Input Connections 8
2.0.6 Digital Output Connections 10
2.0.7 Conveyor Motor Connections 12
2.0.8 List of Devices 13
2.0.9 Terminal Connections 15

3 Plant Photos **17**

4 Siemens S7 300 Family PLC Application **21**
4.1 Project Description 21
4.2 Hardware Configurations 22
4.3 Symbol Table 24
4.4 Ladder Logic Programming Codes 26
4.4.1 Toggle Operations 26
4.4.2 Mode Selection 27
4.4.3 Light Operations 28
4.4.4 Conveyor Belt 1 Enable 28
4.4.5 Yellow Mode 29
4.4.6 Red Mode 34
4.4.7 Green Mode 37
4.4.8 Conveyor Belt Return 42

5 Beckhoff Industrial PC Application **43**
5.1 Project Description 43
5.2 Variable Declarations and Main Program Cycle 44
5.2.1 Global Variables 44
5.2.2 Main Program and Local Variables 45
5.3 Actions 45
5.3.1 Action 1: h001_lock_button_on_handling 45
5.3.2 Action 2: h001_lock_button_off_handling 47
5.3.3 Action 3: h003_soft_button_on_handling 48
5.3.4 Action 4: h004_signal_lamp 49
5.4 Visualization of Touch panel 50
5.5 The Touch Panel Setting and Tests 52

6 Discussion **57**
6.1 PLC 57
6.2 Selecting PLC and Sensors 57

Preface

As a part of the Industrial Control course, model production line is assembled by group members and write the code for achieve the given tasks. For that purpose, designing of each components and writing corresponding codes have been done by the group members. As we are going to complete project, the report for better understanding of project is written. Every detail can be found in the report's corresponding sections.

Sabanci University, January 3, 2019

Ahmet Guclu
<ahmetguclu@sabanciuniv.edu>

Demir Demirel
<demirdemirel@sabanciuniv.edu>

Edin Guso
<edinguso@sabanciuniv.edu>

Emre Erdinc
<emreerdinc@sabanciuniv.edu>

Gunes Basak Ozgun
<gunesozgun@sabanciuniv.edu>

Chapter 1

Problem Definition and Solution Proposal

1.1 Problem Definition

In this Project we are asked to design a conveyor belt system which is packaging yellow and red balls in different combinations.The system is able to work in 3 different modes. Each mode has its own requirements that our system should meet. The system starts running with a start button. The first mode is yellow mode. If the system is in yellow mode, conveyor belt's car should first move to the red container. After the red ball dropped in to the car, the car should move to the yellow container then again to the red container. After getting balls from each container the car with 2 red and 1 yellow balls should move all the way to the left. When the car comes to the left end, it is pushed to the turn table and from the turn table to another conveyor belt.As the final position, the box should move to the right end of this conveyor belt. The second mode is red mode.If the system is in red mode, conveyor belt's car should first move to the yellow container. After the yellow ball dropped in to the car, the car should move to the red container then again to the yellow container. In the end with 2 yellow and 1 red balls the car should move to the left end of the conveyor belt where it is shifted in to the turn table. As the final position the box should slide down from the left wedge.These two modes only work if the box carried by the car is steel. If a wooden box is put on the car, there is another mode. If the system is in this mode, the car first moves beneath the yellow container and wait s for 2 yellow balls to drop in. Afterwards the car moves to red container and waits for 2 red balls to drop in. Then again the car moves to the left end with 2 red and 2 yellow balls and shifts in to the turn table. As the final position the box slides down from the right wedge.

1.2 Solution Proposal

The whole system is controlled by a Siemens S7-300 family PLC(see Figure: 1.1) to meet with the Project requirements. A Ladder logic program is used to adjust each mode of operation of the system. As mentioned in the problem definition, there were 3 modes of operation of the system. We named these modes as Red,Yellow and Green modes. For each mode there is a lamp showing which mode is the system in at that moment. The purpose of that is following the current mode of operation as well as tracing the last mode of operation. The mode change is done by a push button and the system starts running with another push button which we called the start button. We designed this start button to allow running only if there is a box on the car to avoid misuse of the system. The containers are designed to store balls and drop them one by one. A piston attached to the one end of the container is used to move the balls to the pipe and the pipe at the other end is used to carry the balls to the box waiting on the conveyor belts car.The cars position is controlled by a metal ruler and metal sensors in the system. The conveyor belts and turn table are run by motors and the pistons are used to move the box from one section of the system to another section of the system.

Figure 1.1

Chapter 2

Wiring Diagrams

2.0.1 Title Page

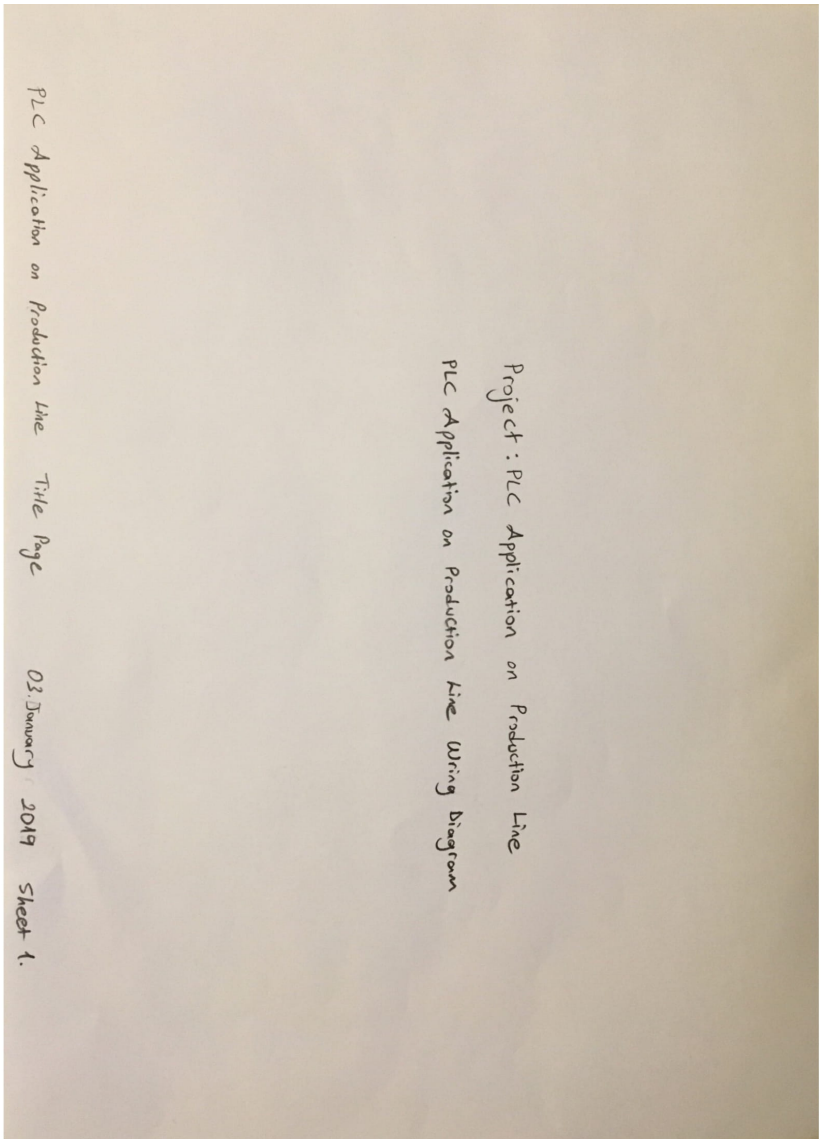


Figure 2.1: Title Page

2.0.2 Content

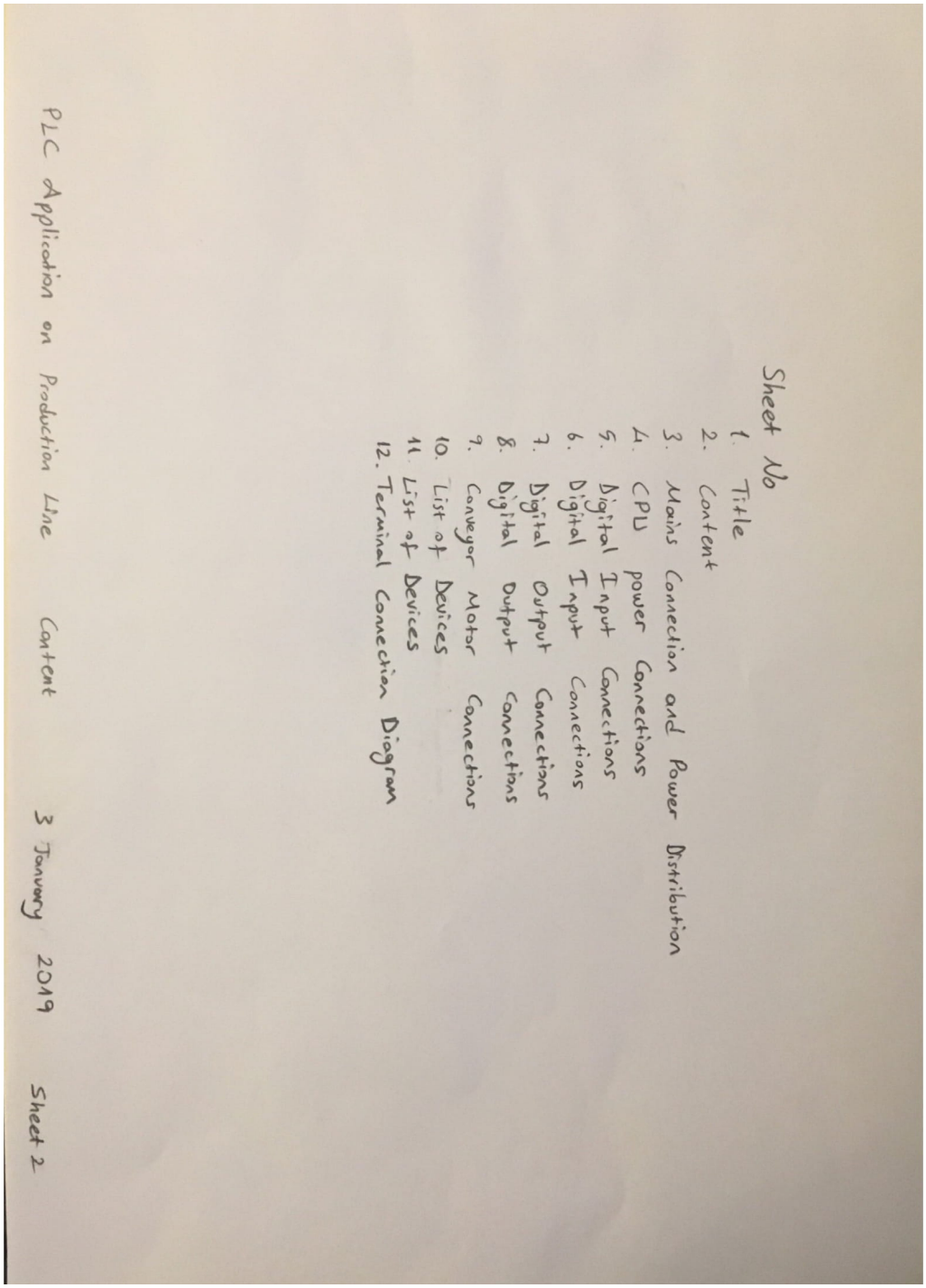


Figure 2.2: Content

2.0.3 Main Connections And Power Distribution

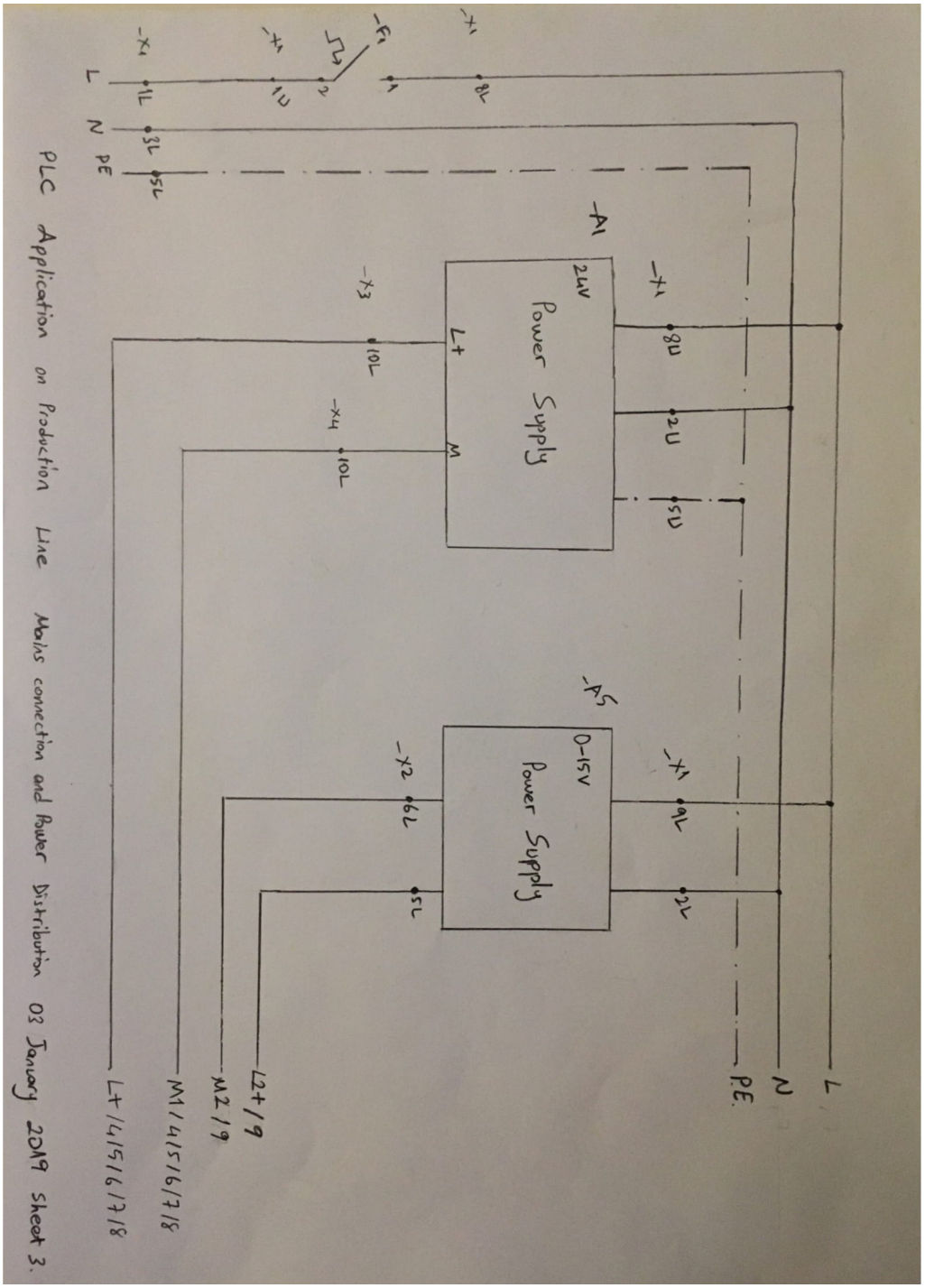


Figure 2.3: Main Connections And Power Distribution

2.0.4 CPU Power Connections

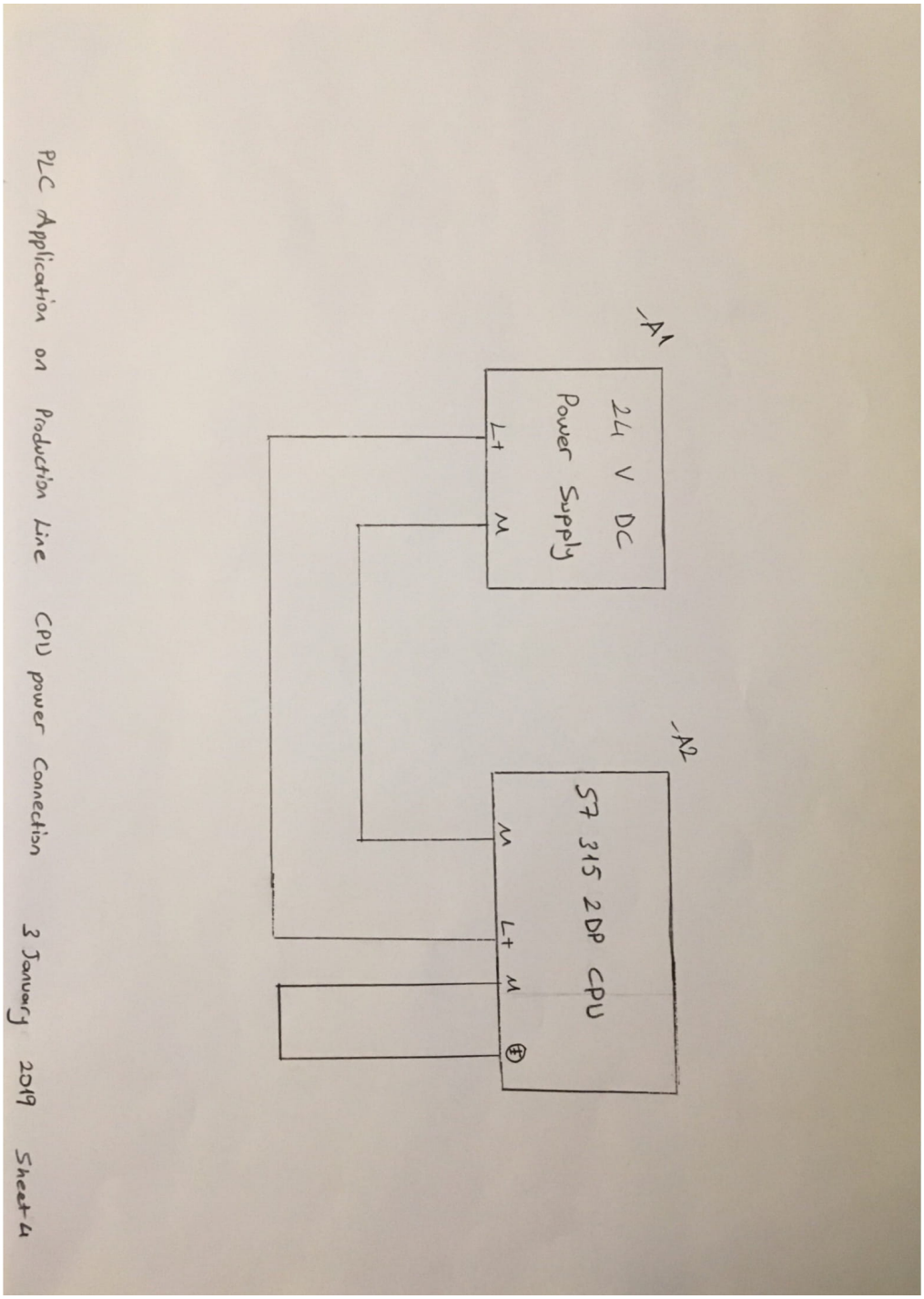


Figure 2.4: CPU Power Connections

2.0.5 Digital Input Connections

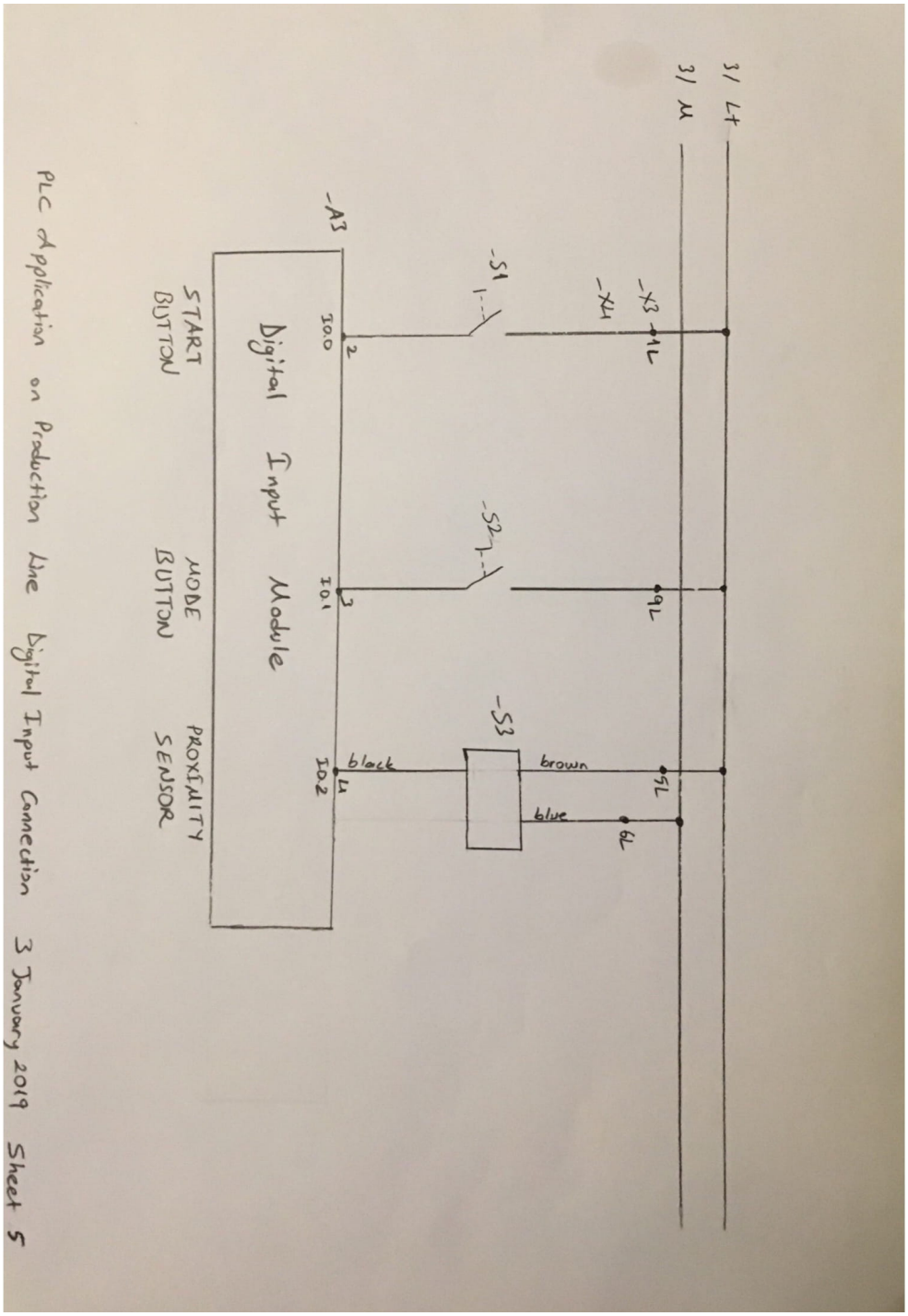


Figure 2.5: Digital Input Connections 1

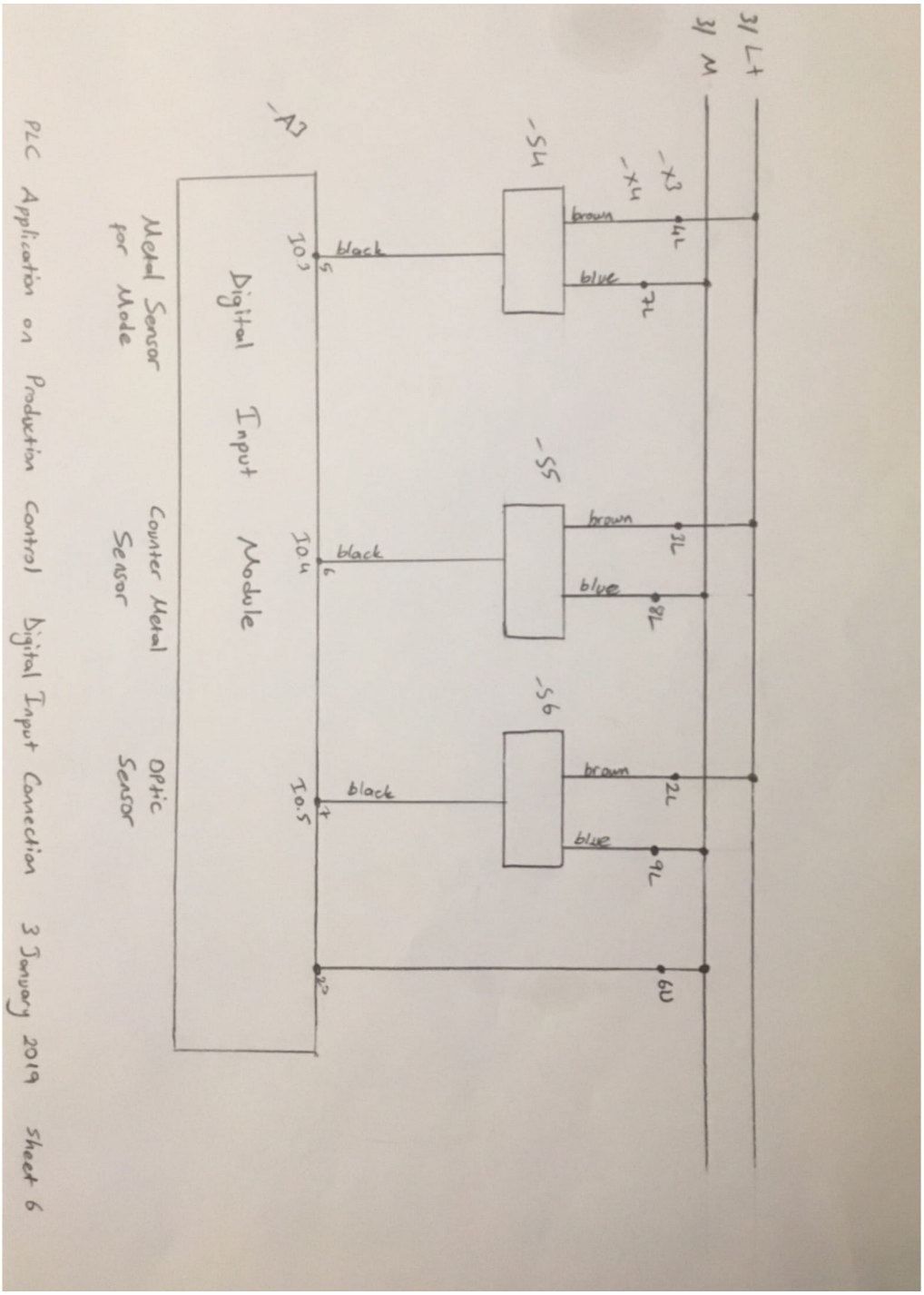


Figure 2.6: Digital Input Connections 2

2.0.6 Digital Output Connections

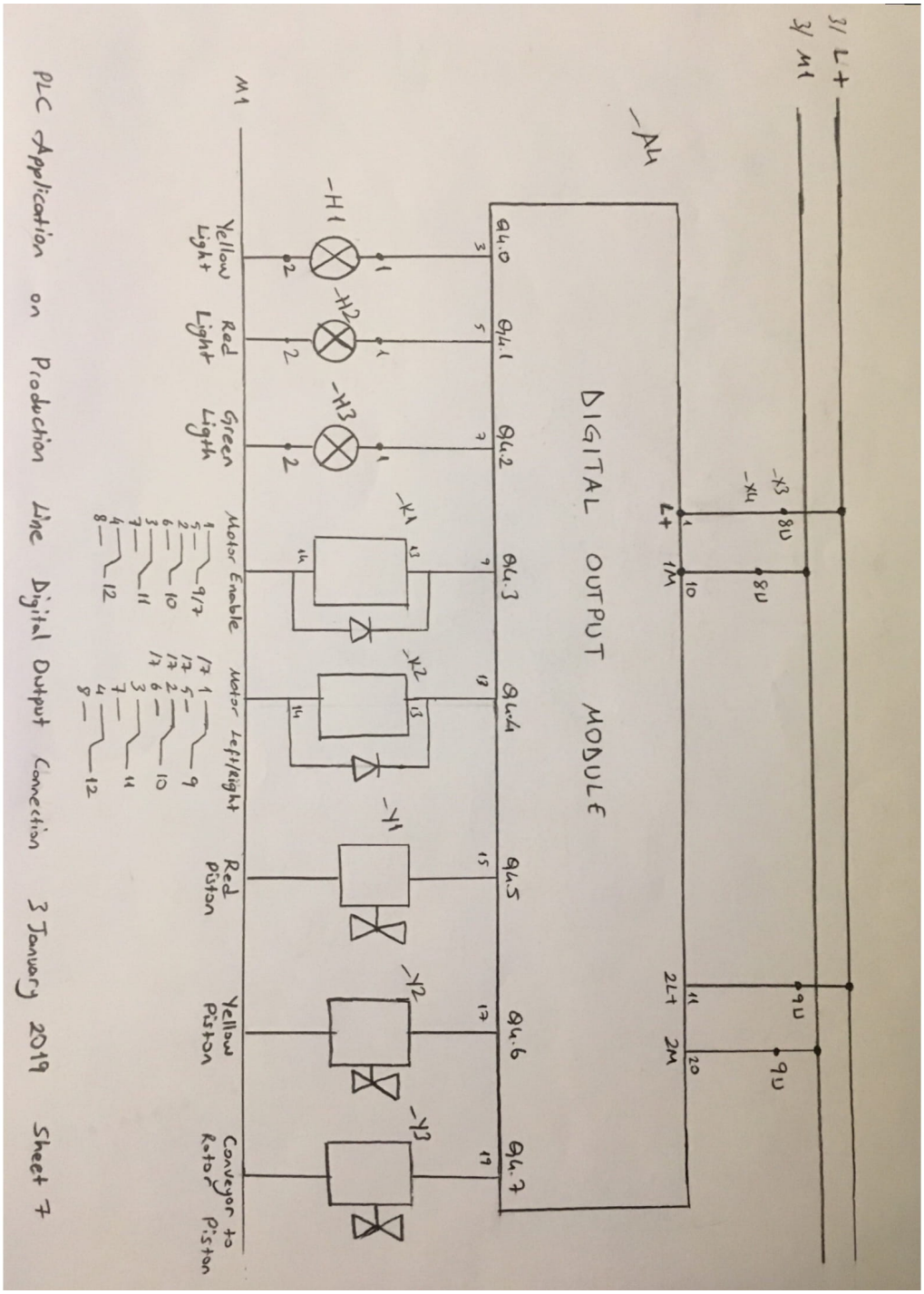


Figure 2.7: Digital Output Connections 1

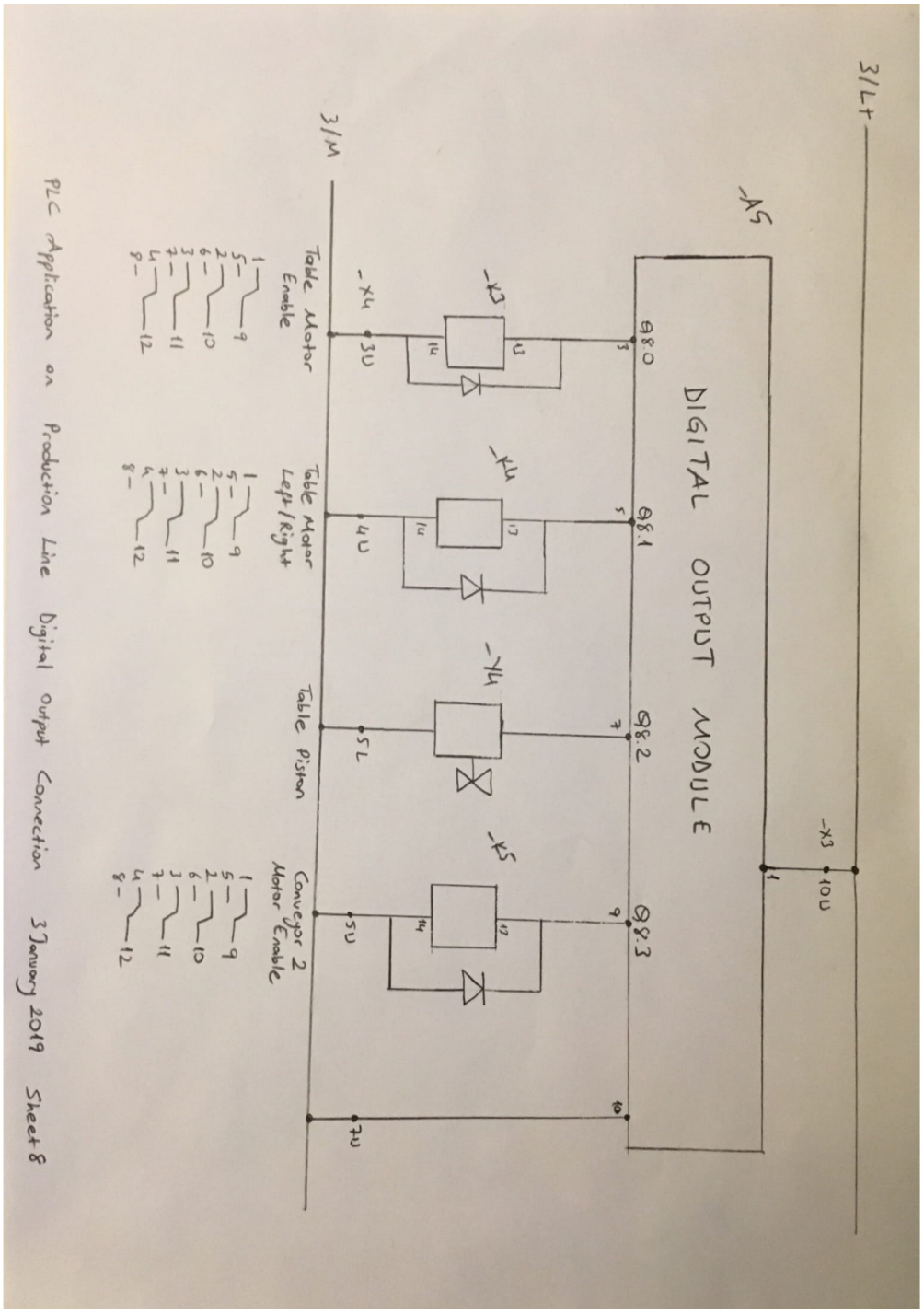


Figure 2.8: Digital Output Connections 2

2.0.7 Conveyor Motor Connections

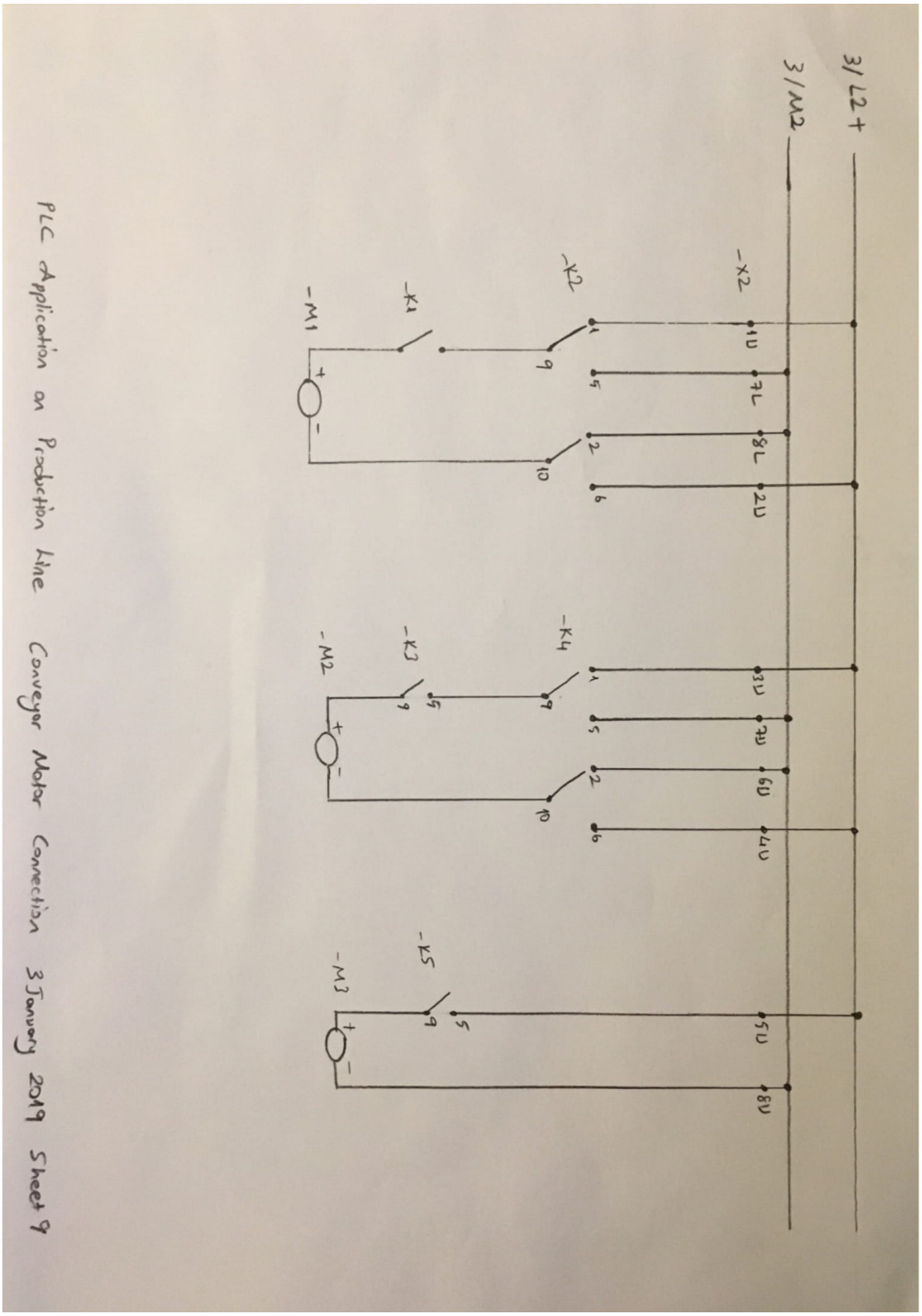
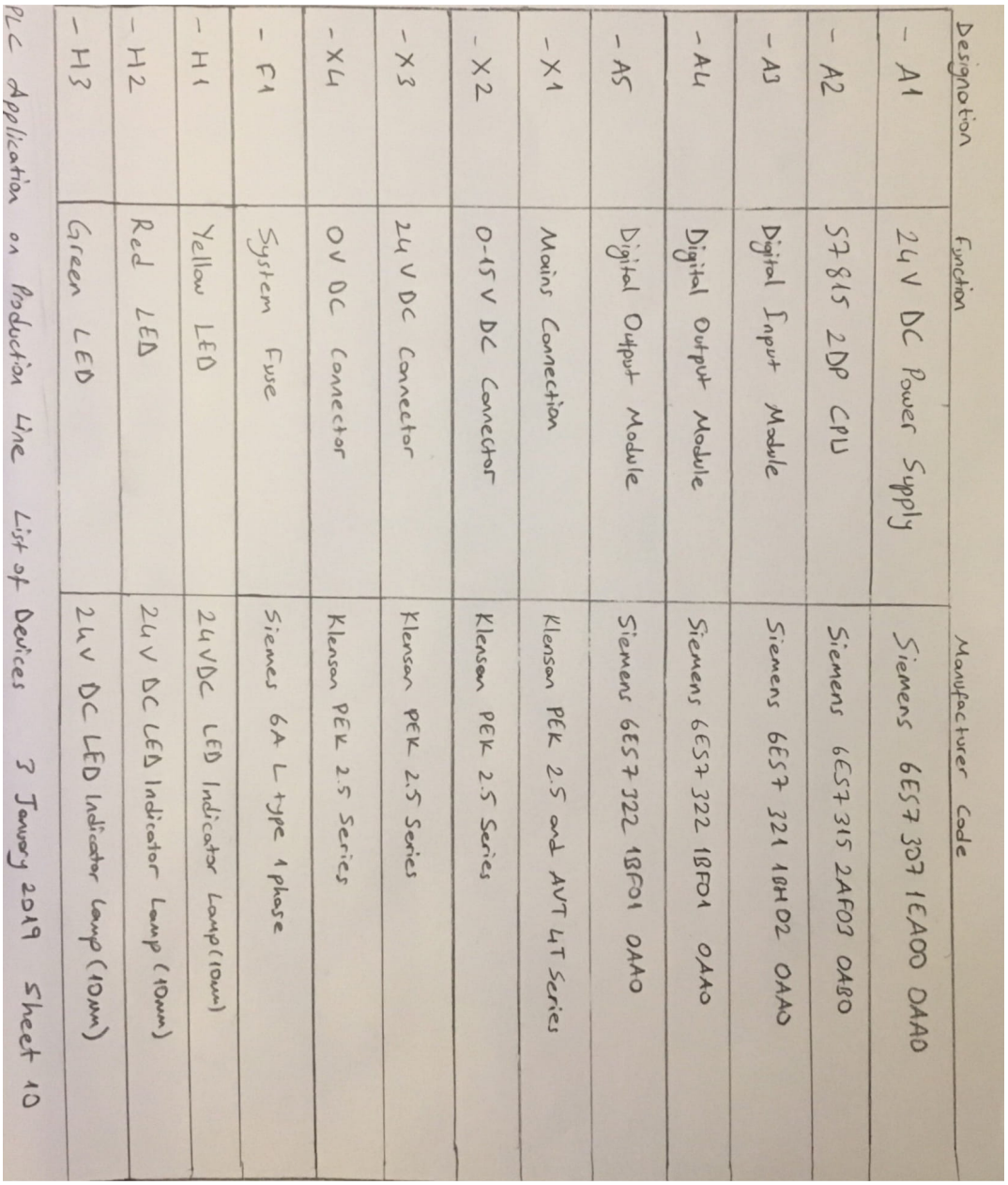


Figure 2.9: Conveyor Motor Connections

2.0.8 List of Devices



Designation	Function	Manufacturer Code
- A1	24 V DC Power Supply	Siemens 6ES7 307 1EA00 0AA0
- A2	S7 815 2DP CPU	Siemens 6ES7 315 2AF03 0AB0
- A3	Digital Input Module	Siemens 6ES7 321 1BH02 0AA0
- A4	Digital Output Module	Siemens 6ES7 322 1BF01 0AA0
- A5	Digital Output Module	Siemens 6ES7 322 1BF01 0AA0
- X1	Mains Connection	Klenson PEK 2.5 and AVT 4T Series
- X2	0-15 V DC Connector	Klenson PEK 2.5 Series
- X3	24 V DC Connector	Klenson PEK 2.5 Series
- X4	0V DC Connector	Klenson PEK 2.5 Series
- F1	System Fuse	Siemes 6A L type 1 phase
- H1	Yellow LED	24VDC LED Indicator Lamp(10mm)
- H2	Red LED	24 V DC LED Indicator Lamp (10mm)
- H3	Green LED	24V DC LED Indicator Lamp (10mm)

PLC Application on Production Line List of Devices 3 January 2019 Sheet 10

Figure 2.10: List of Devices 1

Designation	Function	Manufacturer Code
-S1	Start Button	Telemecanique ZBE101
-S2	Mode Button	Telemecanique ZBE101
-S3	Proximity Sensor	Omron E3F2-R2B4
-S4	Metal Sensor For Mode	Omron E3F2-R2B4
-S5	Counter Metal Sensor	Omron E3F2-R2B4
-S6	Optic Sensor	Omron E3F2-R2B4
-Y1	Red Piston	Festo MEBH-5/2-1/8F
-Y2	Yellow Piston	Festo MEBH-5/2-1/8B
-Y3	Conveyor to Rotor Piston	Festo MEBH-5/2-1/8F
-Y4	Table Piston	Festo MEBH-5/2-1/8F
-K1	Motor Enable Relay	Omron MY4
-K2	Motor Left/Right Relay	Omron MY4
-K3	Table Motor Enable Relay	Omron MY4
-K4	Table Motor Left/Right	Omron MY4
-K5	Conveyor 2 Motor Enable	Omron MY4

PLC Application on Production Line List of Devices 3 January 2019 Sheet 11

Figure 2.11: List of Devices 2

2.0.9 Terminal Connections

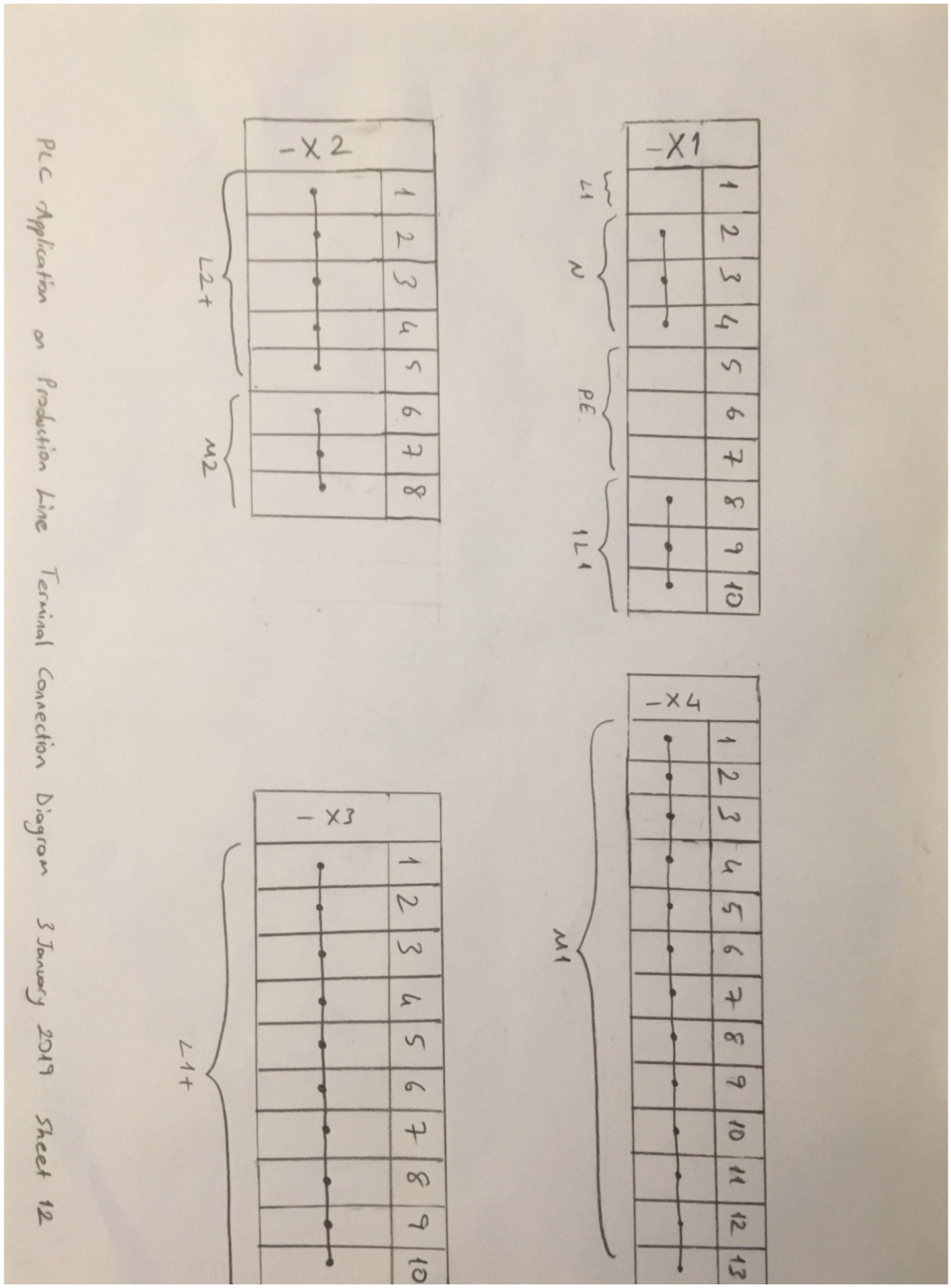
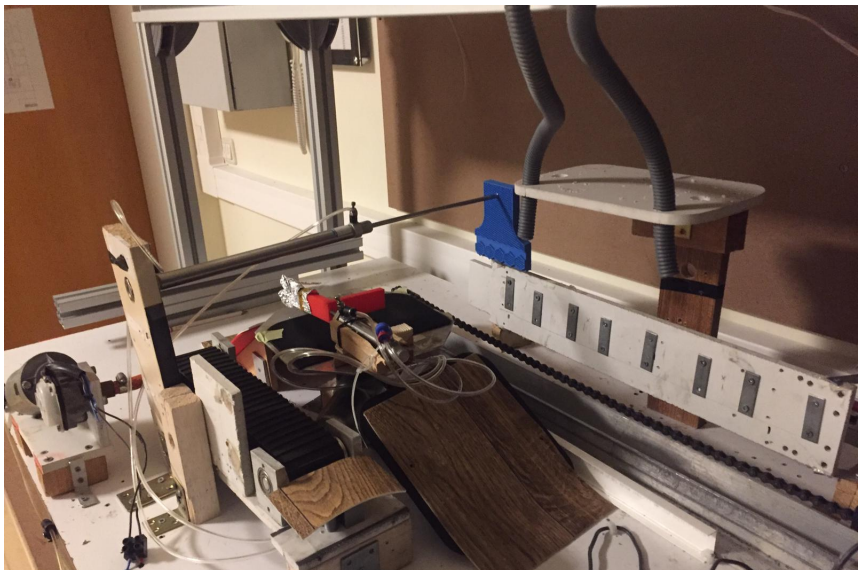
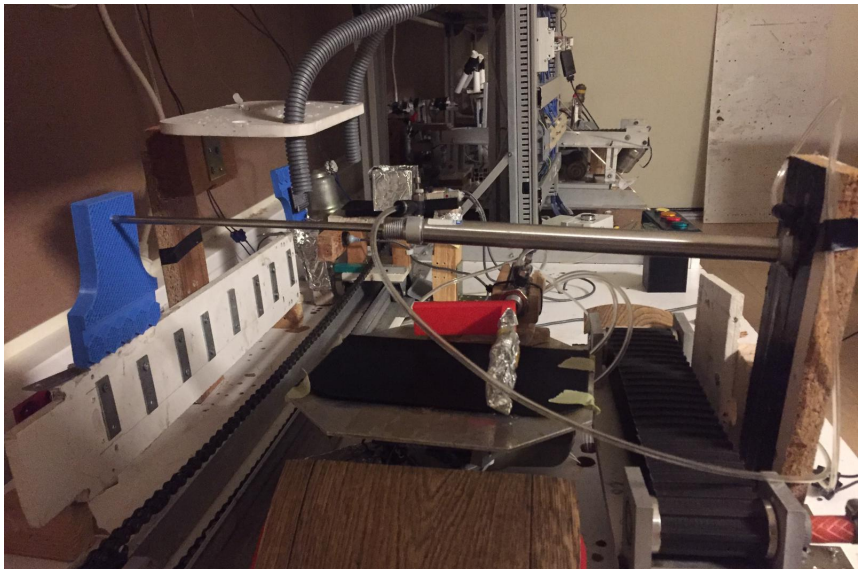


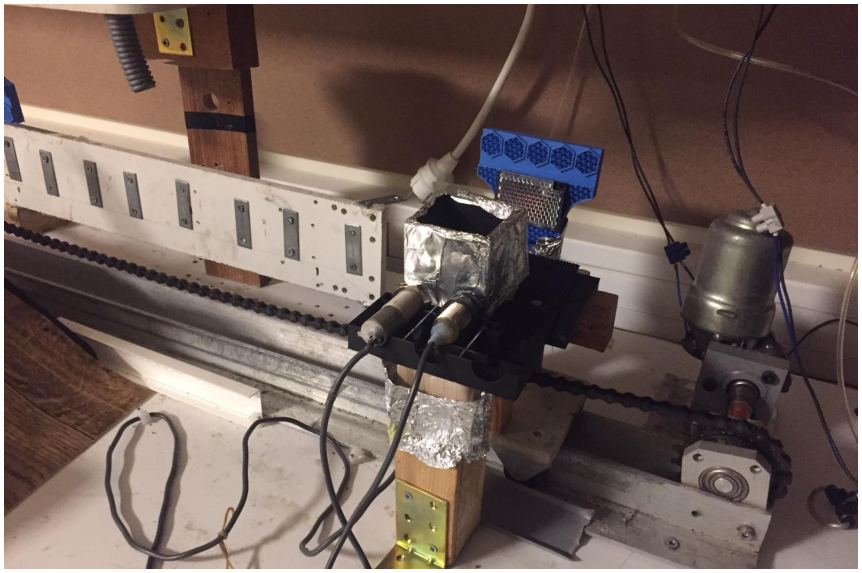
Figure 2.12: Terminal Connections

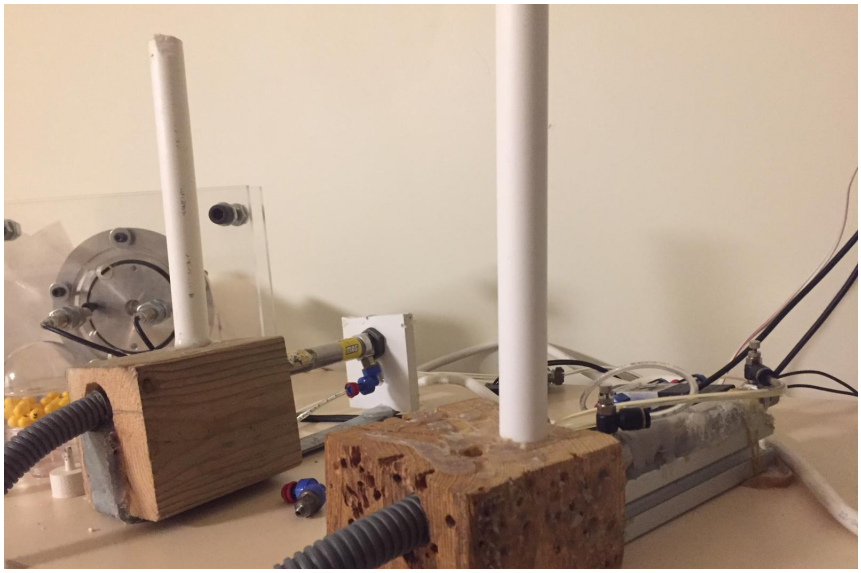
Chapter 3

Plant Photos









Chapter 4

Siemens S7 300 Family PLC Application

4.1 Project Description

In the first part of our term project we were given a task to control a system consisting of two conveyor belts, a turn table, several pistons and sensors. The image of the system that we had to create was as following:

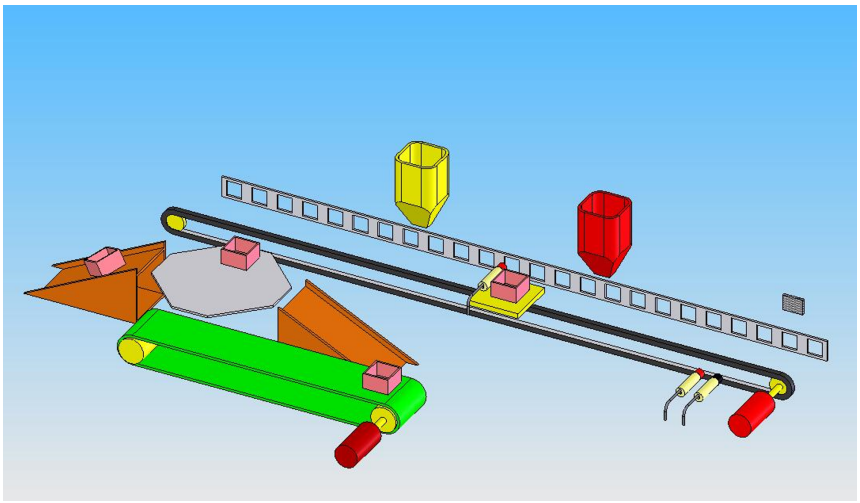


Figure 4.1: Project Layout

There were three different tasks this system had to perform. But they all had to start at the same position and move from the first conveyor belt to the turn table. Some other conditions of the tasks were different. These conditions were explained as:

1. Put the box to the start position (right on the chain).
2. Use a push button to toggle signal “yellow” or “red” modes.
3. If the yellow mode is active and if the box is a steel box fill the box with one red + one yellow + one red balls from the containers.
4. If the red mode is active and if the box is a steel box fill the box with one yellow + one red + one yellow balls from the containers.
5. If the box is a wooden box fill the box with two yellow + two red balls from the containers.
6. Move the box to the turn table.
7. Move wooden boxes down the wedge on the right.
8. If the yellow mode is active and if the box is a steel box move the box to the conveyor and pile them at the right end of the conveyor.
9. If the red mode is active and if the box is a steel box move the box down the wedge on the left.

4.2 Hardware Configurations

For this project, we used a Siemens Programmable Logic Controller. We used a PS 307 5A power supply, CPU 315-2 DP controller, DI16xDC24V input module and two DO8xDC24V/2A output modules. In the hardware configuration part of our program we clicked the print option and saved the hardware configuration. It looked as following (see Figure: 4.2)

```
SIMATIC 300(1)


UR - Rack 0
Rack 0, Slot 1
  Short designation:                PS 307 5A
  Order no.:                        6ES7 307-1EA00-0AA0
  Designation:                      PS 307 5A
  Width:                            1
  Comment:
    - - -

Rack 0, Slot 2
  Short designation:                CPU 315-2 DP
  Order no.:                        6ES7 315-2AF03-0AB0
  Designation:                      CPU 315-2 DP
  Width:                            1
  MPI address:                      2
  Highest MPI address:              31
  Baud rate:                        187.5 Kbps
  Comment:
    - - -

Rack 0, Slot 2, Interface X2
  Short designation:                DP
  Order no.:                        - - -
  Designation:                      DP
  Width:                            1
  PROFIBUS address:                 2
  Comment:
    - - -
  Addresses
    Inputs
      Start:                        1023
      End:                          1023


  Synchronization mode:             None
  Time interval:                    None

Rack 0, Slot 4
  Short designation:                DI16xDC24V
  Order no.:                        6ES7 321-1BH02-0AA0
  Designation:                      DI16xDC24V
  Digital channels:                 16 Inputs
  Width:                            1
  Comment:
    - - -
  Addresses
    Inputs
      Start:                        0
      End:                          1

Rack 0, Slot 5
  Short designation:                DO8xDC24V/2A
  Order no.:                        6ES7 322-1BF01-0AA0
  Designation:                      DO8xDC24V/2A
  Digital channels:                 8 Outputs
  Width:                            1
  Comment:
    - - -
  Addresses
    Outputs
      Start:                        4
      End:                          4

Rack 0, Slot 6
  Short designation:                DO8xDC24V/2A
  Order no.:                        6ES7 322-1BF01-0AA0
  Designation:                      DO8xDC24V/2A
  Digital channels:                 8 Outputs
  Width:                            1
  Comment:
    - - -
  Addresses
    Outputs
      Start:                        8
      End:                          8
```

Figure 4.2: Hardware configuration

4.3 Symbol Table

The symbol table of our program is below. As we didn't have comments on the symbols, they will be explained in the next page.

SIMATIC LAB5_2018\SIMATIC 300(1)\CPU 315-2 DP\S7 Program(1)\Symbols 12/26/2018 19:43:17

Properties of symbol table

Name:	Symbols
Comment:	
Created on	08.11.2018 17:12:36
Last modified on:	11.12.2018 09:47:23
Last filter criterion:	Alle Symbole
Number of symbols:	53/ 53
Last Sorting:	Symbol Ascending

Symbol	Address	Data type	Comment
Conveyor Belt 1 Dir	Q 4.4	BOOL	
Conveyor Belt 1 Enable	Q 4.3	BOOL	
Conveyor Belt 1 Stop	M 20.6	BOOL	
Conveyor Belt 2 Enable	Q 8.3	BOOL	
Conveyor Belt Return	M 30.0	BOOL	
Conveyor to Table Piston	Q 4.7	BOOL	
Counter Sensor	I 0.4	BOOL	
fall-rise status	M 20.0	BOOL	
Green light	Q 4.2	BOOL	
Green Mode	M 20.4	BOOL	
Green Mode Light Control	M 22.4	BOOL	
Green Mode Part 2	M 24.0	BOOL	
Green Mode Part 3	M 24.1	BOOL	
Green Mode Part 4	M 24.2	BOOL	
Green Mode Part 5	M 24.3	BOOL	
Metal sensor	I 0.3	BOOL	
Optic Sensor	I 0.5	BOOL	
Proximity Sensor	I 0.2	BOOL	
Red Container	Q 4.5	BOOL	
Red light	Q 4.1	BOOL	
Red Mode	M 20.3	BOOL	
Red Mode Part 2	M 22.0	BOOL	
Red Mode Part 3	M 22.1	BOOL	
Red Mode Part 4	M 22.2	BOOL	
Red Mode Part 5	M 22.3	BOOL	
Rotor Table Direction	Q 8.1	BOOL	
Rotor Table Enable	Q 8.0	BOOL	
Start Button	I 0.0	BOOL	
System Active	M 20.5	BOOL	
Table Piston	Q 8.2	BOOL	
Toggle Button Red/Yellow	I 0.1	BOOL	
Yellow Container	Q 4.6	BOOL	
Yellow light	Q 4.0	BOOL	
Yellow Mode	M 20.2	BOOL	
Yellow Mode Part 2	M 20.7	BOOL	
Yellow Mode Part 3	M 21.0	BOOL	
Yellow Mode Part 4	M 21.1	BOOL	
Yellow Mode Part 5	M 21.2	BOOL	
Yellow Mode Part 6	M 21.3	BOOL	
Yellow On/Red Off	M 20.1	BOOL	

Page 1

Figure 4.3: Symbol Table

- I0.0 Start Button is a NO push button used as the main starter of the system.
- I0.1 Toggle Button RedYellow is a NO push button and it switches between red and yellow modes when pressed.
- I0.2 Proximity Sensor is a NC optic switch at the beginning of the system which checks whether a box is placed or not.
- I0.3 Metal Sensor is a NO inductive switch at the beginning of the system which checks whether the box placed is a metal one or not.
- I0.4 Counter Sensor is a NO inductive switch fixed on the cart which counts the number of metal plates passed and determines the position of the cart.
- I0.5 Optic Sensor is a NC optic switch under the turn table which determines the angle of the turn table.
- Q4.0 Yellow Light is indicating whether the toggle button is set to yellow mode.
- Q4.1 Red Light is indicating whether the toggle button is set to red mode.
- Q4.2 Green Light is indicating whether the system was started with a wooden box.
- Q4.3 Conveyor Belt 1 Enable controls whether the first conveyor belt is on or off.
- Q4.4 Conveyor Belt 1 Dir controls whether the conveyor belt moves left(off) or right(on).
- Q4.5 Red Container controls the piston which is inside the red container to drop red balls.
- Q4.6 Yellow Container controls the piston which is inside the yellow container to drop yellow balls.
- Q4.7 Conveyor to Table Piston controls the piston which moves the box from the first conveyor belt to the turn table.
- Q8.0 Rotor Table Enable controls whether the turn table is turning or not.
- Q8.1 Rotor Table Direction controls whether the turn table is turning counter-clockwise(off) or clockwise(on).
- Q8.2 Table Piston controls the piston which is fixed on the turn table and is used for moving the box from the turn table to their destination.

- Q8.3 Conveyor Belt 2 Enable controls whether the second conveyor belt is on or off.
- M20.0 and M20.1 are used for the toggle operation.
- M20.2, M20.7, M21.0, M21.1, M21.2 and M21.3 are used for consecutive execution of the yellow mode operations.
- M20.3, M22.0, M22.1, M22.2 and M22.3 are used for consecutive execution of the red mode operations.
- M20.4, M24.0, M24.1, M24.2 and M24.3 are used for consecutive execution of the green mode operations.
- M20.5 handles many things such as the movement of the conveyor belt 1 and the light management.
- M20.6 is used to stop the conveyor belt 1 at desired locations.
- M22.4 is used in light management.
- M30.0 is used to return the cart to the initial position.

4.4 Ladder Logic Programming Codes

All the code will be explained network by network below.

4.4.1 Toggle Operations

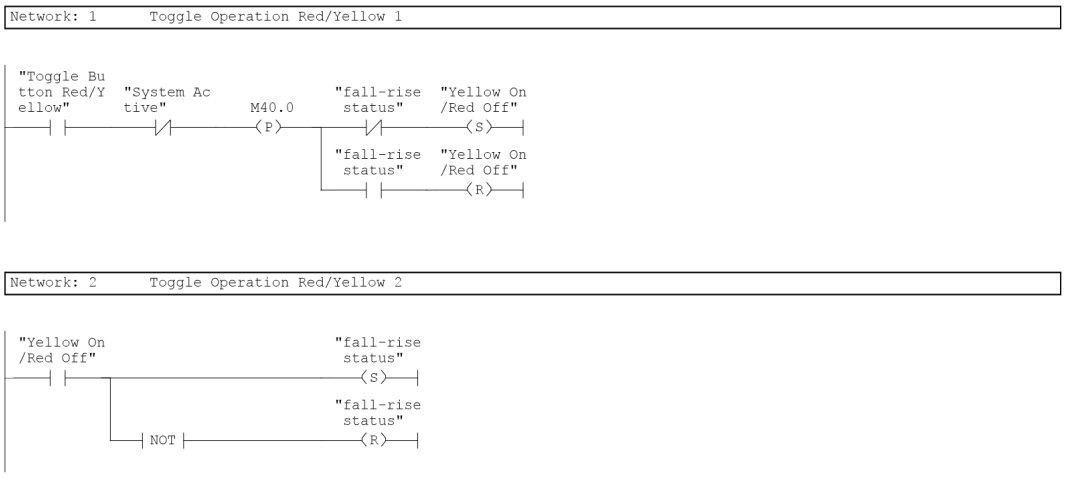


Figure 4.4: Network 1,2

In Network 1: “Toggle Operation RedYellow 1” and Network 2: “Toggle Operation RedYellow 2”(see Figure: 4.4), we implement a simple toggle operation used for switching between yellow and red modes. Whenever the “Toggle Button RedYellow” is pressed, the “Yellow OnRed Off” memory bit switches. This memory bit indicates that yellow mode is active when it is on and the red mode is active when it is off. We also put a normally closed “System Active” memory bit which prevents the mode from changing if a chain of operations for any mode has started.

4.4.2 Mode Selection

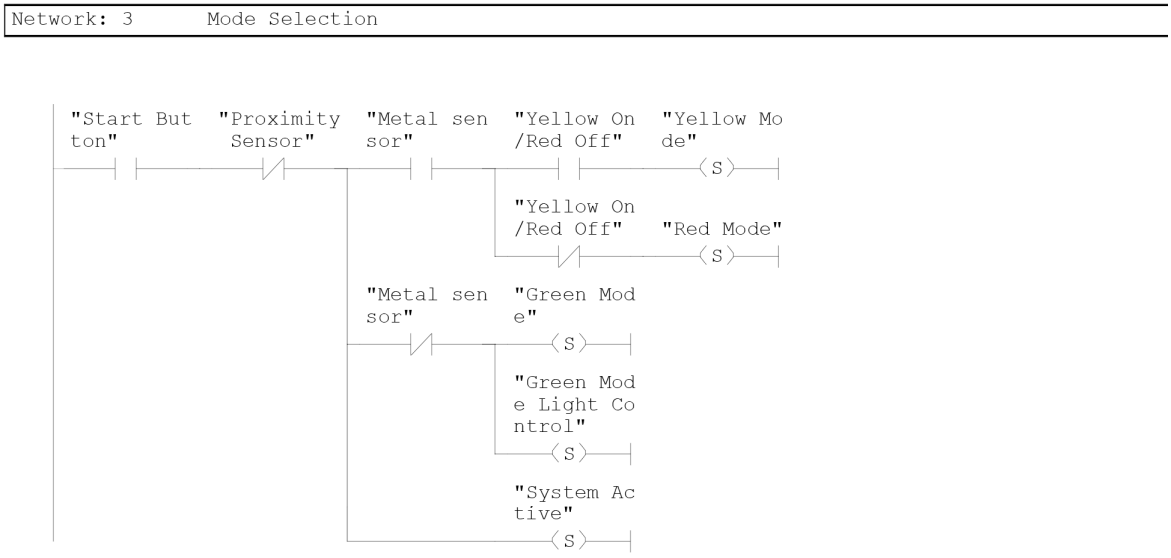


Figure 4.5: Network 3

In Network 3: “Mode Selection”(see Figure: 4.5), we handle in which mode the system should start operating, and whether it should start operating at all. First two conditions for every mode is that the “Start Button” should be pressed and that there is an object in front of the “Proximity Sensor” setting it to off. Then, no matter what mode is selected, “System Active” memory bit which controls the conveyor belt movement is set to on.
If the “Metal Sensor” is off, meaning the box is a wooden box, then the “Green Mode” memory bit which starts the whole process of green mode and the “Green Mode Light Control” memory bit which handles the light management are set to on. On the other hand, if the “Metal Sensor” is on, meaning the box is a metal box, depending on which mode was selected by the toggle operation, either “Yellow Mode” or “Red Mode” starts the chain of operations for the respective mode.

4.4.3 Light Operations

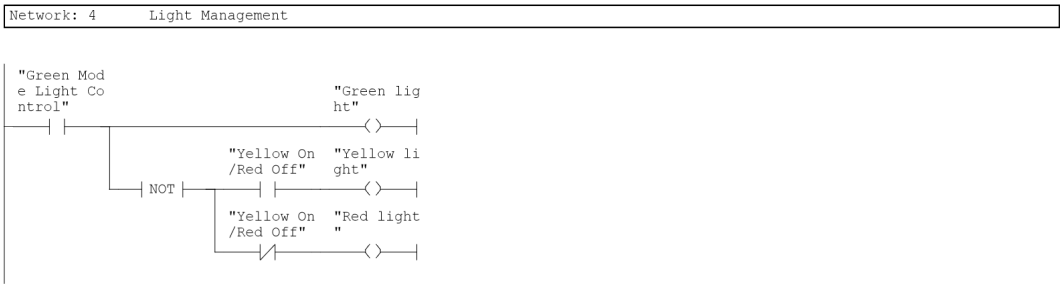


Figure 4.6: Network 4

In Network 4: "Light Management", we handle which light should be on at what time. If we have entered the green mode, which means that "Green Mode Light Control" is also turned on, then the "Green light" will stay on until the green mode operation is finished. If we are not in the green mode, then depending on which mode we are in, "Yellow light" or "Red light" is on. Also these lights cannot be switched during an operation thanks to the condition in Network 1(see Figure: 4.4).

4.4.4 Conveyor Belt 1 Enable

Figure 4.7: Network 5

In Network 5: "Conveyor Belt 1 Enable"(see Figure: 4.7), we handle all of the conveyor belt 1 turning on and off other than returning the cart to the initial position, which is handled in Network 23. Whenever a rising edge comes from the condition "System Active" memory bit on and "Conveyor Belt 1 Stop" memory bit off, the "Conveyor Belt 1 Enable" is set to on. Similarly, whenever a falling edge comes from the same condition, it is set to off. The "Conveyor Belt 1 Stop" memory bit is used in many networks to stop the movement of the conveyor belt 1.

4.4.5 Yellow Mode

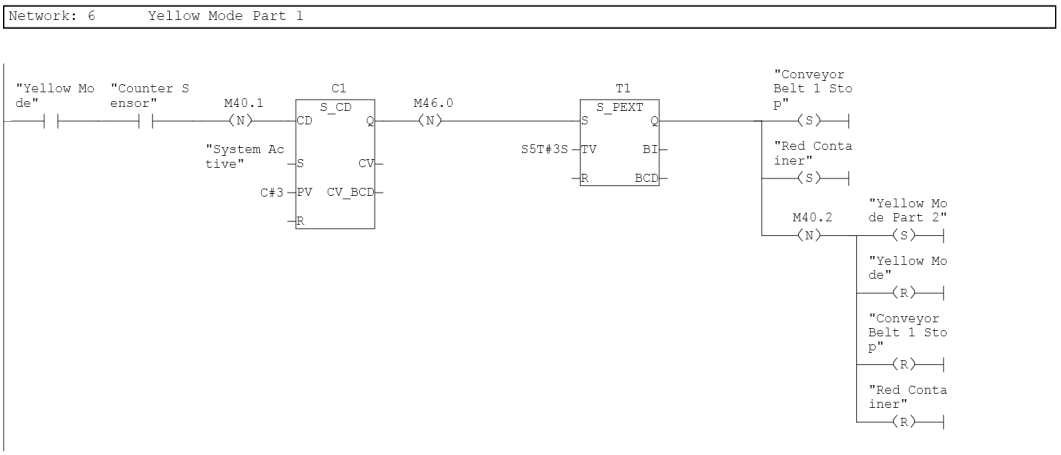


Figure 4.8: Network 6

In Network 6: "Yellow Mode Part 1"(see Figure: 4.8), the cart is moving due to network 1 towards left and the system has started operating in yellow mode. We start counting the metal plates our cart has passed in order to know the position of the cart. With every metal plate passed, a falling edge is sensed, and the counter counts down. When three plates are passed, meaning we are under the red container, the counter reaches the value 0 and it stops outputting a signal, which is sensed as a falling edge and that falling edge activates a timer for 3 seconds.

When this timer becomes on, the "Conveyor Belt 1 Stop" memory bit is set to on in order to stop the conveyor belt 1 in Network 5. Also, "Red Container" which is the piston inside the red container is set to on in order to drop a red ball into our box. When the 3 seconds of the timer is finished a falling edge is sensed. This causes the "Conveyor Belt 1 Stop" memory bit to be set to off and let the cart moving again. The "Red Container" is set to off. Also, we switch in which mode we are. As we have finished the first part of yellow mode, we set the "Yellow Mode" memory bit to off and set the "Yellow Mode Part 2" memory bit to on.

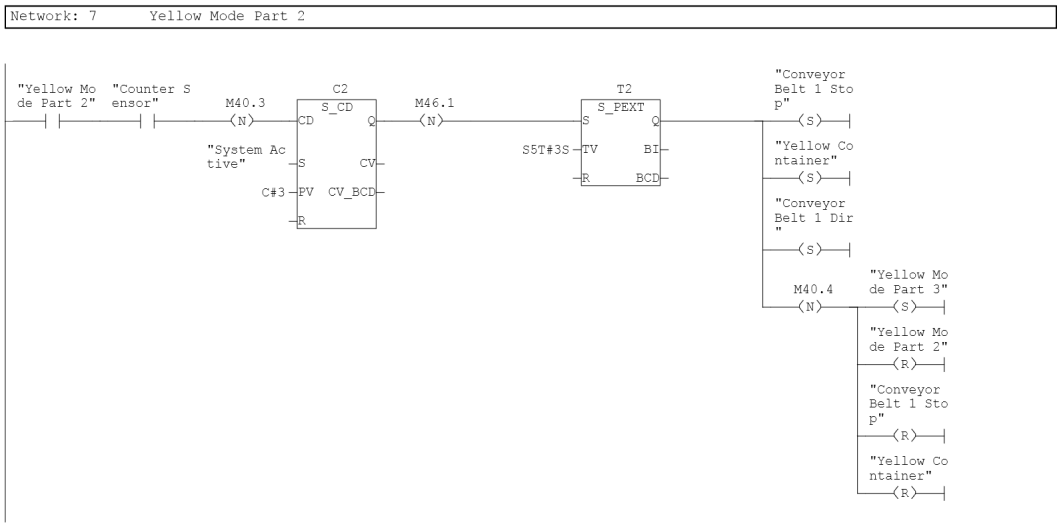


Figure 4.9: Network 7

In Network 7: "Yellow Mode Part 2"(see Figure: 4.9), we count the number of metal plates after moving from the red container. We use the same method of detecting the falling edges of the "Counter Sensor". When we count 3 metal plates, meaning we are under the yellow container, a falling edge is received and starts a timer for 3 seconds.

When this timer becomes on, the "Conveyor Belt 1 Stop" memory bit is set to on in order to stop the conveyor belt1. The "Conveyor Belt 1 Dir" is set to on, too, because the cart will have to move towards right this time to reach the red container another time. Also, "Yellow Container" which is the piston inside the yellow container is set to on in order to drop a yellow ball into our box. When the three seconds of the timer finish and a falling edge is sensed, we reset the "Conveyor Belt 1 Stop" to allow the cart to move. We also reset the "Yellow Container" to allow the piston to retract. And finally, we switch the mode from "Yellow Mode Part 2" to "Yellow Mode Part 3".

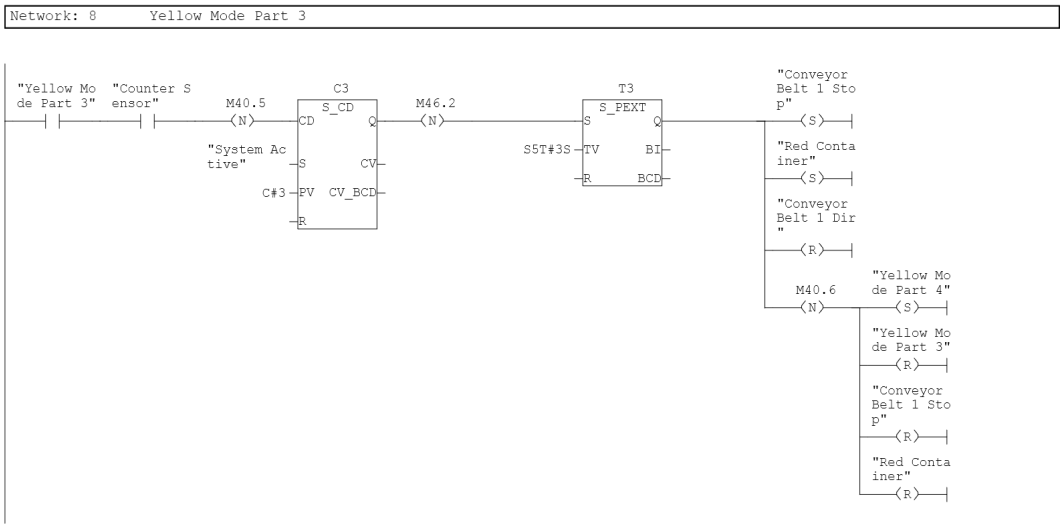


Figure 4.10: Network 8

In Network 8: "Yellow Mode Part 3"(see Figure: 4.10), we have started moving towards right in order to get another red ball. When we count the metal plates and know that we are under the red container, the timer starts for 3 seconds. When these three seconds start, the cart is stopped, the direction is changed, and a red ball is dropped. When the three seconds end, the cart starts moving towards left, piston inside the red container retracts and the mode is switched from "Yellow Mode Part 3" to "Yellow Mode Part 4".

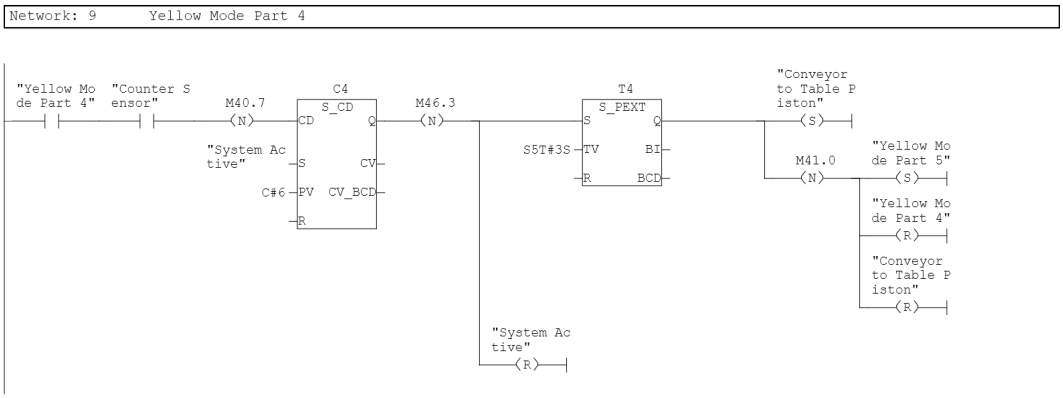


Figure 4.11: Network 9

In Network 9: "Yellow Mode Part 4"(see Figure: 4.11), we are aiming to reach the position where our box will move from the cart to the turn table. We need to count 6 metal plates beginning from the red container in order to reach that position. When we reach that position, we reset the "System Active" memory bit

in order to stop the cart from moving and start a 3 second timer. When this 3 second timer starts, “Conveyor to Table Piston” is set to on in order to move the box from the cart to the turn table. When the 3 second timer ends, the piston retracts and the mode switches from “Yellow Mode Part 4” to “Yellow Mode Part 5”.

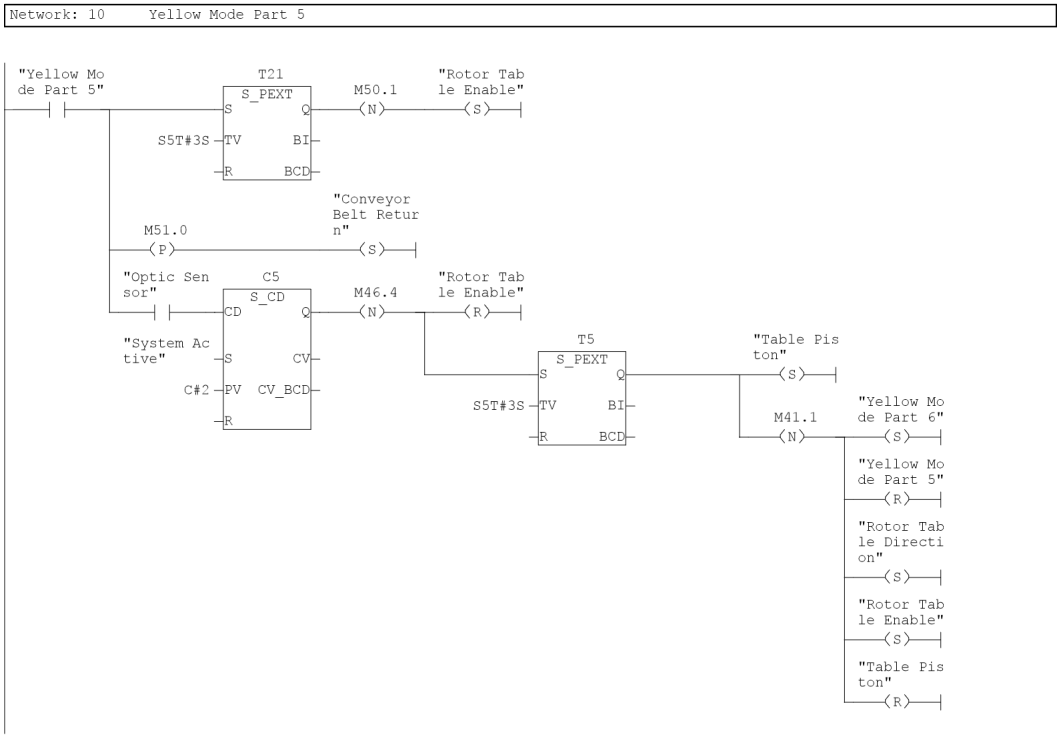


Figure 4.12: Network 10

In Network 10: “Yellow Mode Part 5”(see Figure: 4.12), we first detect a rising edge from the “Yellow Mode Part 5” and set the “Conveyor Belt Return” memory bit to on, which makes the cart return to its initial position. This operation will be explained in Network 23. Then we need to wait for three seconds for the cart to move towards its initial position, because otherwise it would be in the way of the table piston which turns together with the turn table. When the 3 seconds end, “Rotor Table Enable” is set to on which allows the turn table to turn.

Then we start counting how many times the “Optic Sensor” has been turned on. There are reflectors at every 90 degrees under the turn table, and we want to turn 90 degrees in the yellow mode, so the counter should count only one “Optic Sensor” output. But in the initial position of the turn table, the “Optic Sensor” already sees a reflector, so when “Yellow Mode Part 5” is turned on, a signal comes to the counter input and decreases the counter by one. Therefore, we needed to set the counter to two.

When the turn table has been turned by 90 degrees and the counter is set to off, a falling edge is detected and “Rotor Table Enable” is set to off, stopping the turn table. Also, a 3 second timer is started. When this timer starts, “Table Piston” is set to on and the box is moved onto the second conveyor belt. When the timer ends, “Table Piston” retracts, “Rotor Table Direction” and “Rotor Table Enable” are set to on so that the turn table can return to its initial position. Also, the mode is switched from “Yellow Mode Part 5” to “Yellow Mode Part 6”(see Figures: **??**-4.15).

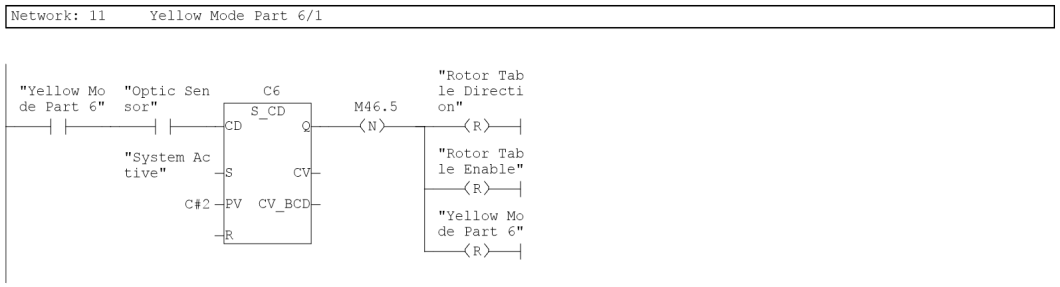


Figure 4.13: Network 11

In Network 11: “Yellow Mode Part 61”(see Figure:4.13), we handle the returning of the turn table to its initial position. Again the “Optic Sensor” counts the number of reflectors in order to know the position of the turn table. When it counts twice, a falling edge is detected at the output of the counter, setting “Rotor Table Direction”, “Rotor Table Enable” and “Yellow Mode Part 6” to off. This signals the ending of the yellow mode chain of operations.

Figure 4.14: Network 12

In Network 12: "Yellow Mode Part 62"(see Figure: 4.14), we start a timer for 3 seconds which turns on the "Conveyor Belt 2 Enable". This moves the box to its destination.

4.4.6 Red Mode

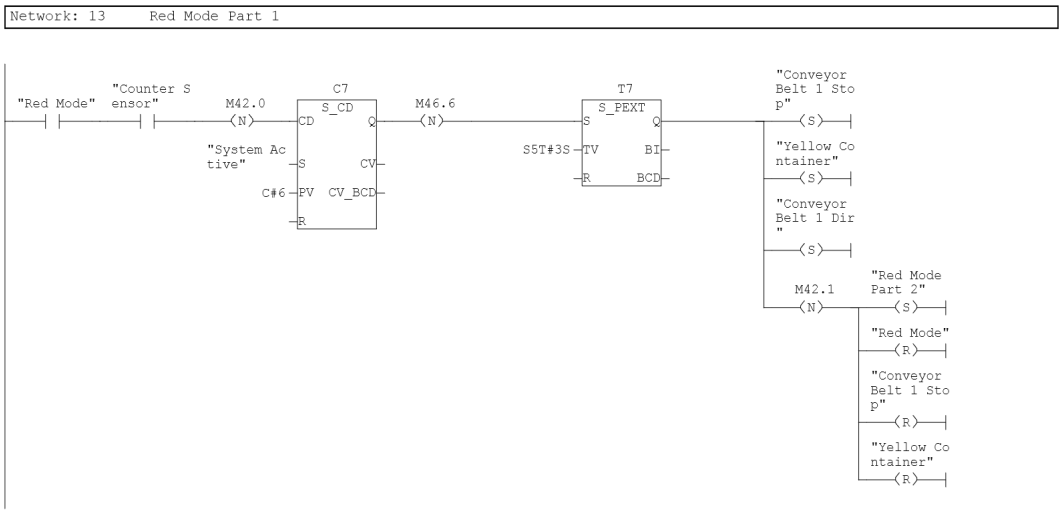


Figure 4.15: Network 13

In Network 13: "Red Mode Part 1", we do a very similar operation to Network 6(see Figure: 4.8). The only differences are that the counter counts 6 times instead of three as we want to get to the yellow container first and the directions is changed because we would like to move towards the red container afterwards.

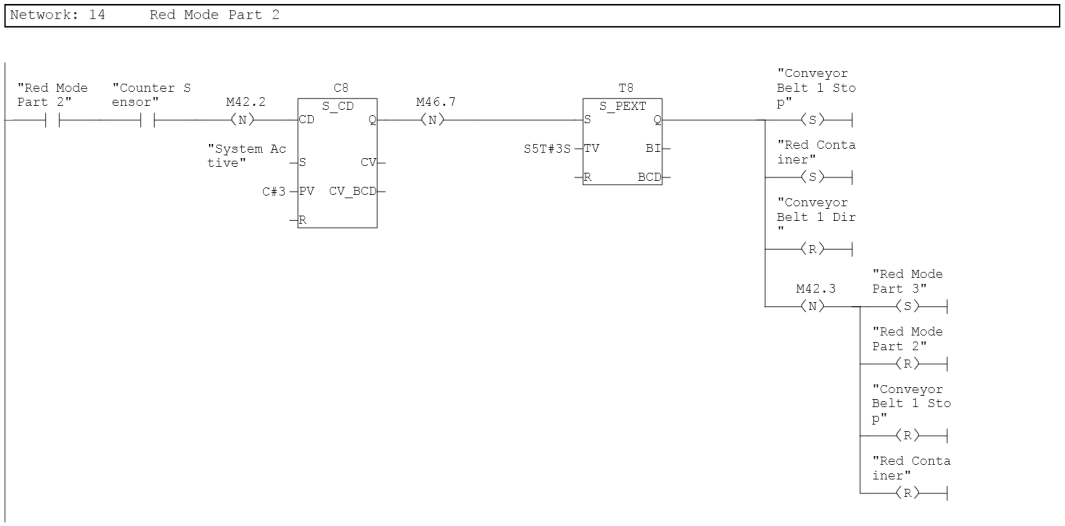


Figure 4.16: Network 14

Network 14: "Red Mode Part 2"(see Figure: 4.16), is a very similar operation to Network 7(see Figure: 4.9), the only difference being that we move from yellow container to red container instead of the opposite.

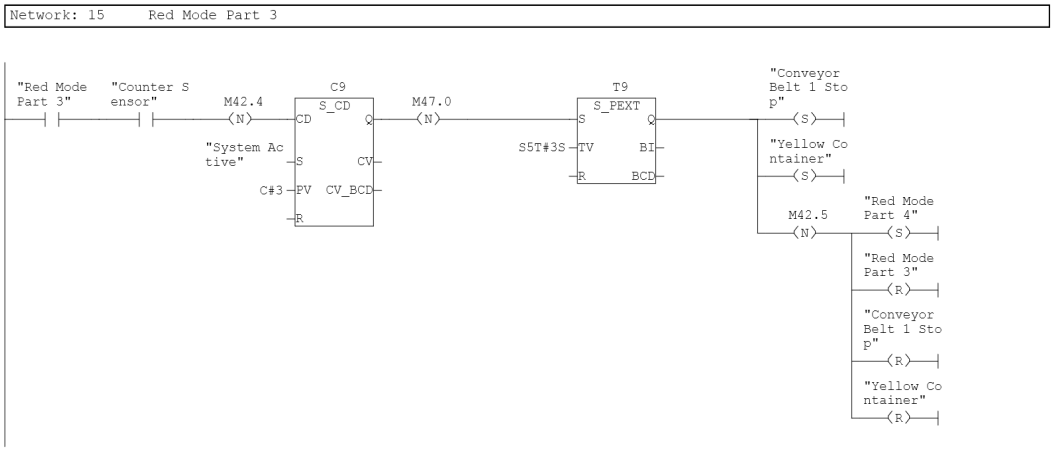


Figure 4.17: Network 15

Network 15: "Red Mode Part 3"(see Figure: 4.17), is a very similar operation to Network 8(see Figure: 4.10), the only difference being that we move from red container to yellow container instead of the opposite.

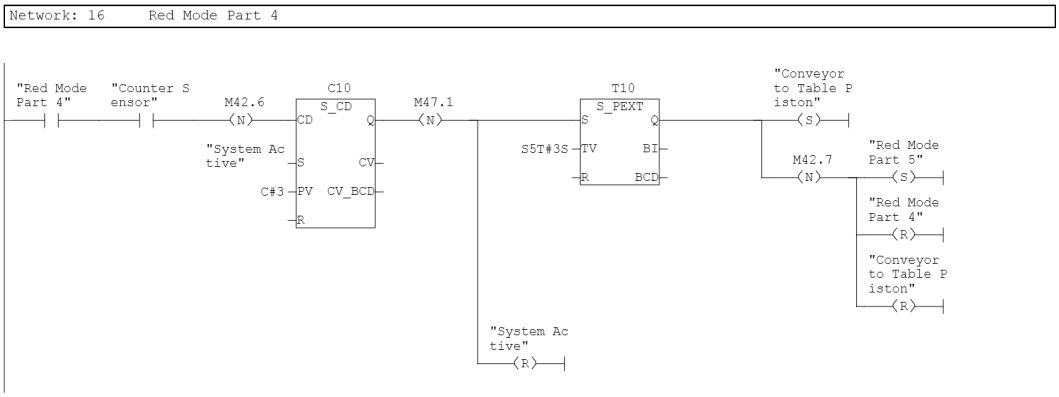


Figure 4.18: Network 16

Network 16: "Red Mode Part 4"(see Figure: 4.18), is a very similar operation to Network 9(see Figure: 4.11), the only difference being that we start moving from the yellow container instead of red container to the end position and therefore we need to count 3 metal plates instead of 6.

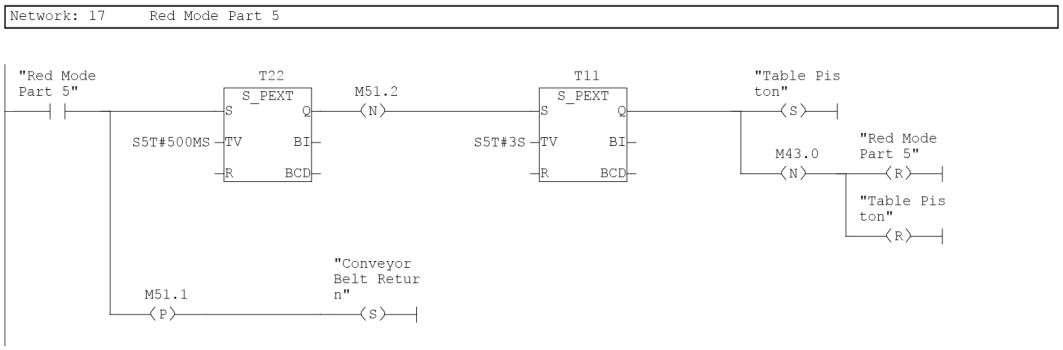


Figure 4.19: Network 17

In Network 17: "Red Mode Part 5"(see Figure: 4.19), we first set the "Conveyor Belt Return" memory bit to on and start the operation of returning the cart to the initial position. In red mode, we are not supposed to turn the turn table at all, we just have to activate the table piston. But we had to set a 500ms timer anyway to avoid the "Table Piston" and "Conveyor to Table Piston" from colliding. Then, when this 500ms timer ends, a 3 second timer starts. When this timer starts, "Table Piston" is set to on, and the box is pushed to the left of the turn table to its destination. When the timer ends, the piston retracts and the "Red Mode Part 5" is set to off, concluding the red mode chain of operations.

4.4.7 Green Mode

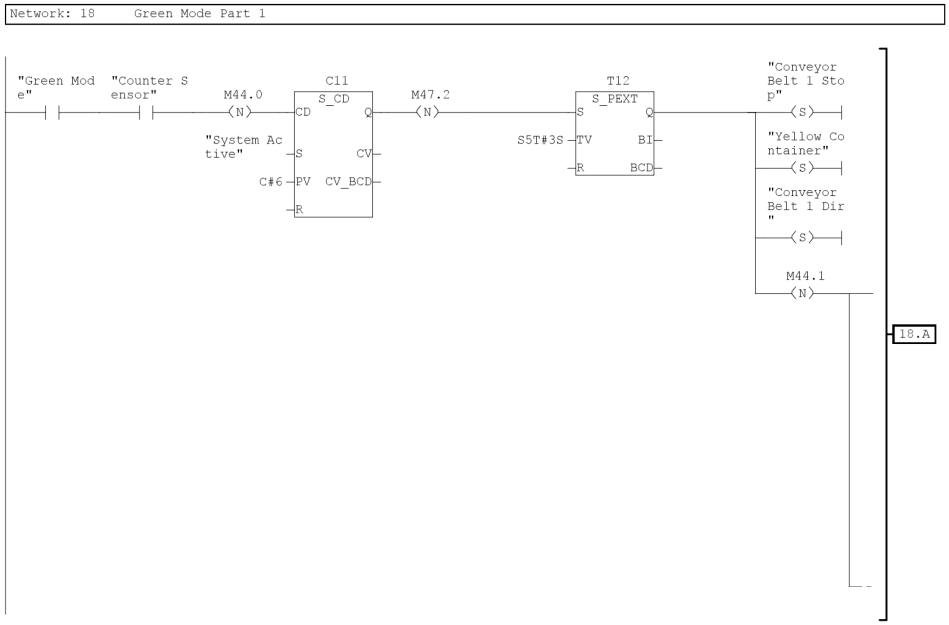


Figure 4.20: Network 18.1

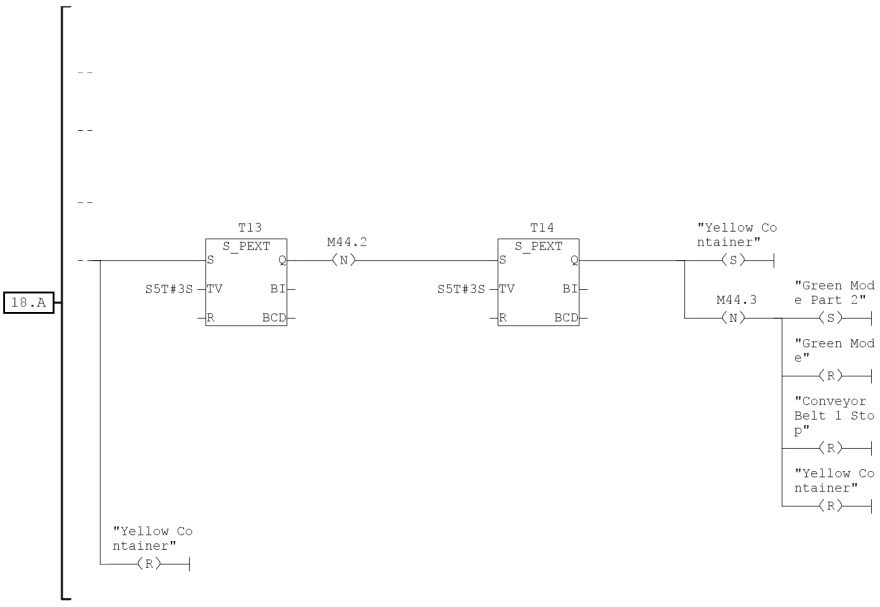


Figure 4.21: Network 18.2

In Network 18: “Green Mode Part 1”(see Figure: 4.20), the task is different then the tasks in Networks 6(see Figure: 4.8) and 13(see Figure: 4.15). We have to reach yellow container first, just like in Network 13(see Figure: 4.15), but then we have to drop two red balls instead of one. This is achieved by first, when the counter reaches 0, we stop the conveyor belt, change its direction and set the “Yellow Container” to on just like the other networks. But when the timer ends, we do not move the cart, we just retract the “Yellow Container” piston so that another ball can be loaded. Then when the next timer ends, we start yet another timer and set “Yellow Container” to on again. When the third timer ends, we set the “Yellow Container” to off, start moving the cart towards the red container and switch the mode from “Green Mode” to “Green Mode Part 2”.

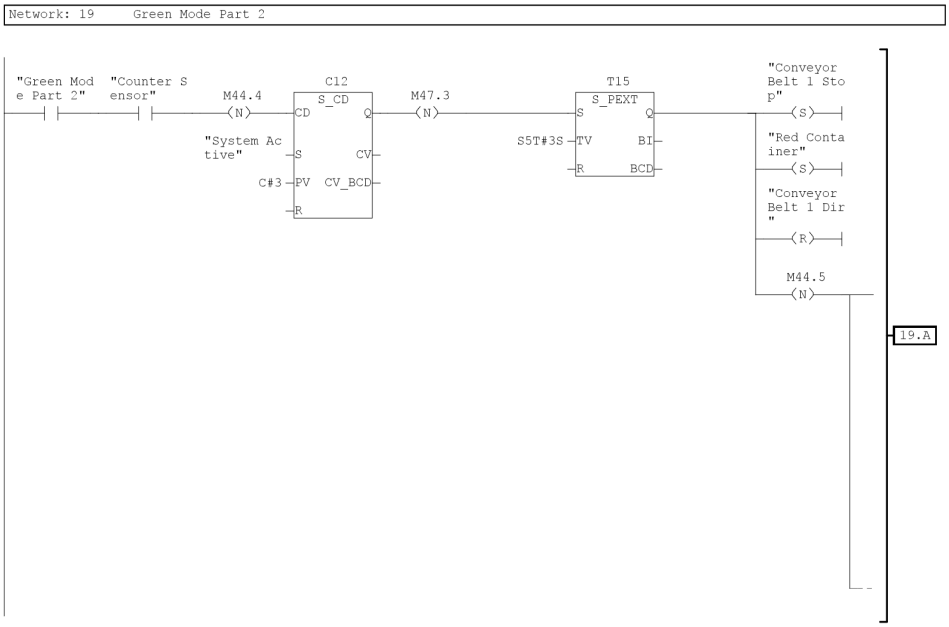


Figure 4.22: Network 19.1

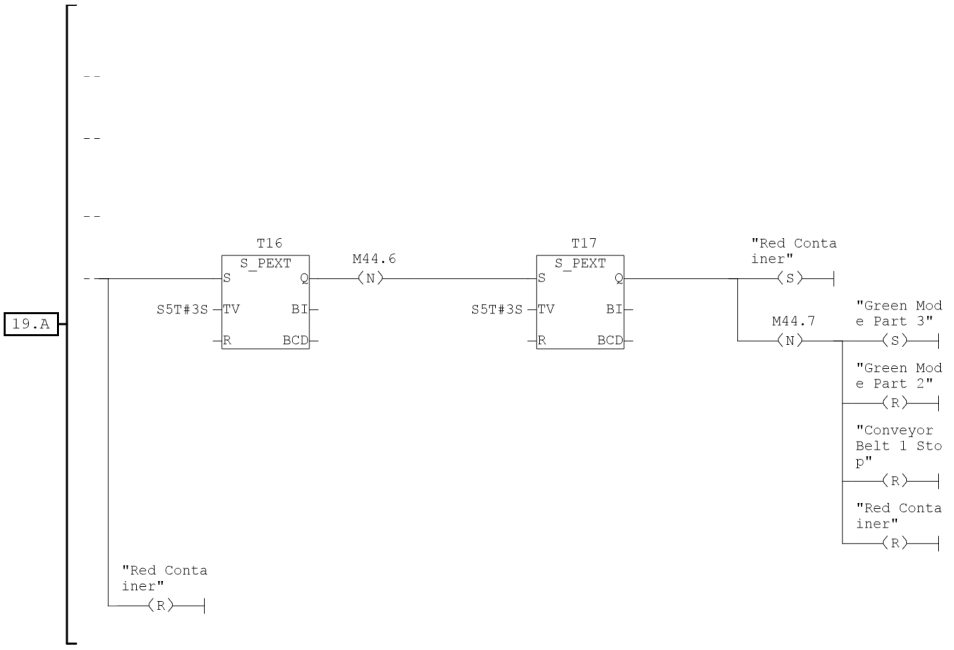


Figure 4.23: Network 19.2

In Network 19: “Green Mode Part 2”(see Figure: 4.22), we have the exact same operation just as in the previous network. Just instead of moving from the initial position to the yellow container and drop yellow balls, we move from the yellow container to the red container and drop red balls. At the end of this network, we start moving towards the turn table and switch the mode from “Green Mode Part 2” to “Green Mode Part 3”.

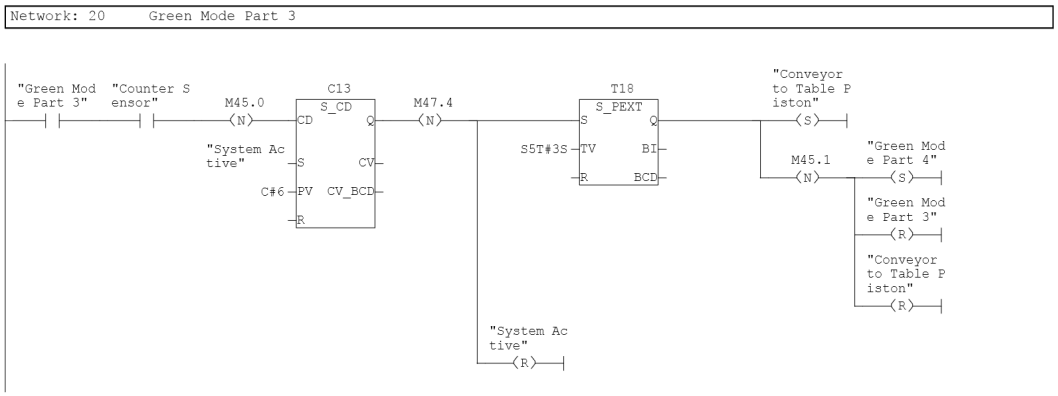


Figure 4.24: Network 20

In Network 20: “Green Mode Part 3”(see Figure: 4.24), we do the exact same operation as in the Network 9(see Figure: 4.11). We mode the cart from the red container to the conveyor to table transfer point.

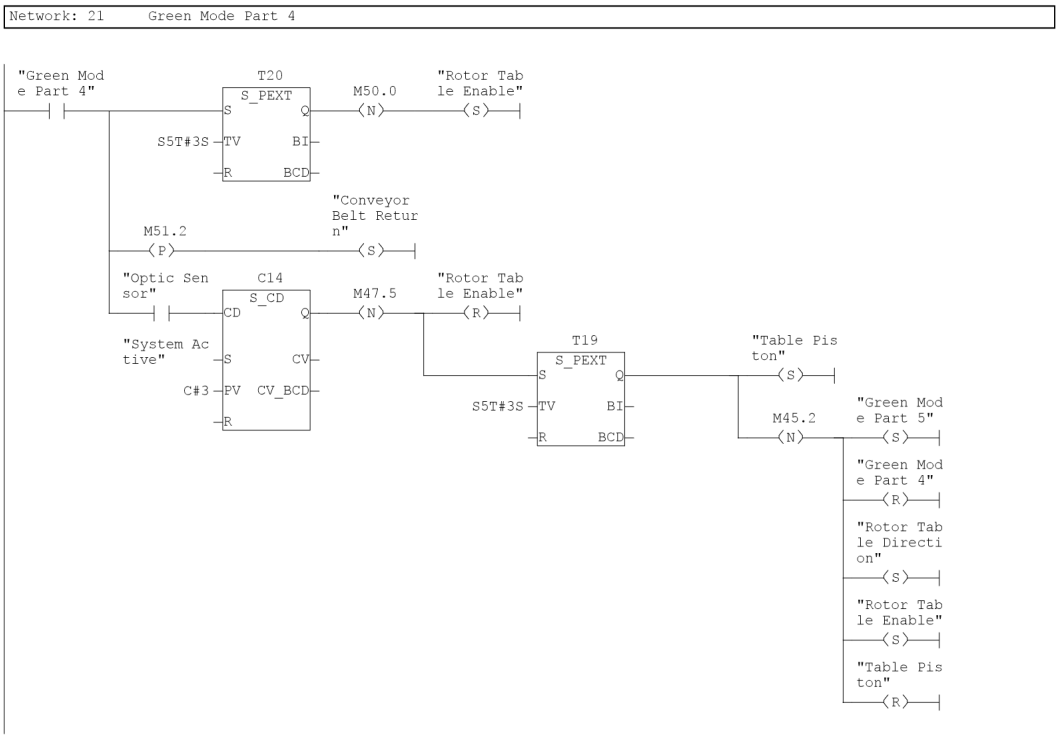


Figure 4.25: Network 21

Network 21: "Green Mode Part 4"(see Figure: 4.25), is very similar o Network 10(see Figure: 4.12). The only difference is that the counter value is set to 3 instead of 2, because we want to turn the turn table by 180 degrees instead of 90 degrees. And by doing that, the box is pushed towards the right of the turn table, to reach its destination.

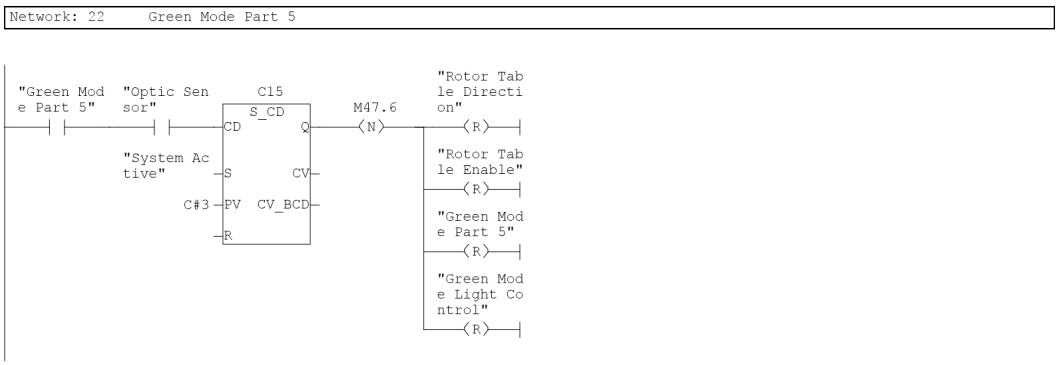


Figure 4.26: Network 22

In Network 22: "Green Mode Part 5"(see Figure: 4.26), just like in Network 11(see Figure: 4.13), we return the turn table to its initial position. In addition to Network 11(see Figure: 4.13), we set the "Green Mode Light Control" to off, so that the green light is not on anymore and the lights are switchable between red and yellow.

4.4.8 Conveyor Belt Return

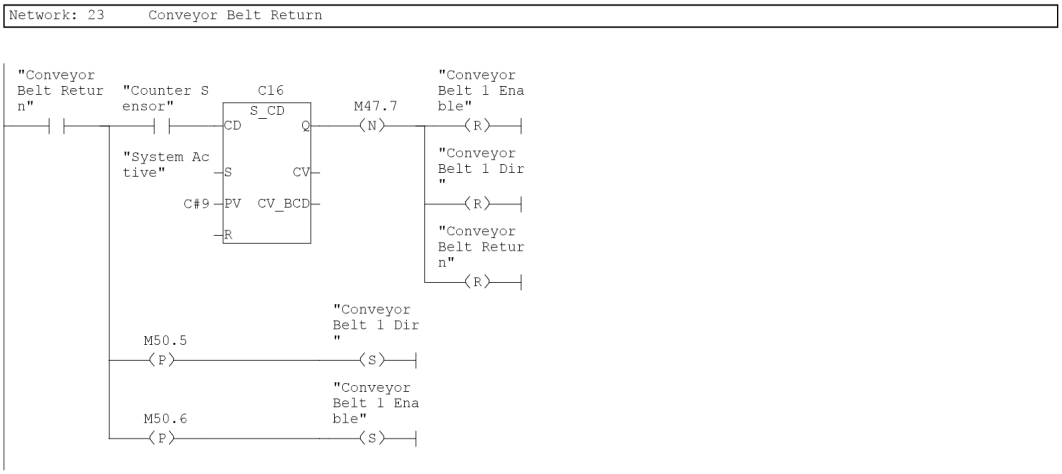


Figure 4.27: Network 23

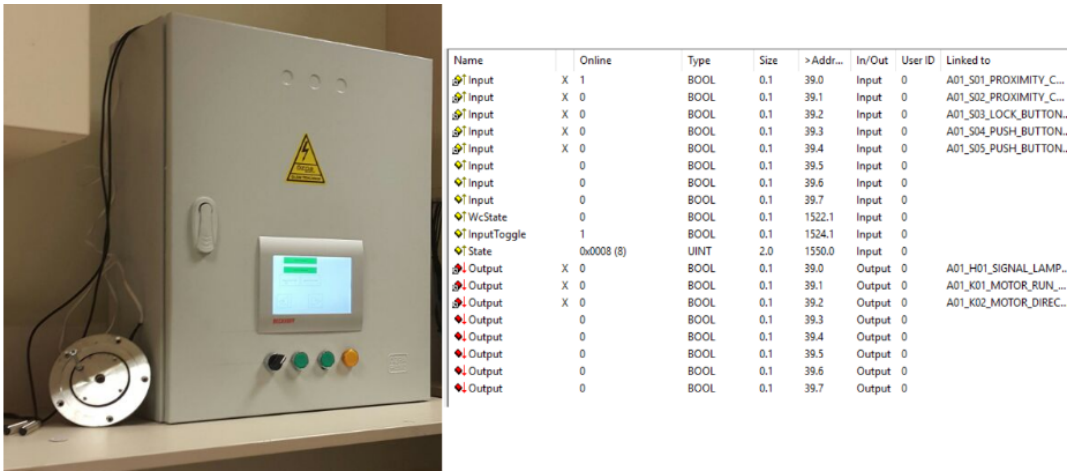
In Network 23: "Conveyor Belt Return"(see Figure: 4.27), we move the cart to its initial position. Firstly, we set both the "Conveyor Belt 1 Dir" and "Conveyor Belt 1 Enable" to on so that the cart starts returning. We count the number of metal plates the cart has passed. When the cart reaches the 9^{th} metal plate, it means that it reached the initial position. Therefore we set the "Conveyor Belt 1 Enable", "Conveyor Belt 1 Dir" and "Conveyor Belt Return" memory bit to 0.

Chapter 5

Beckhoff Industrial PC Application

5.1 Project Description

In this project we used the Beckhoff PLC system to create the system explained below. We used Beckhoff Panel PC with touch panel, a DC motor, DC motor "run" relay and DC motor "direction" relay. To know about the position of the motor, we used two inductive NO PNP proximity switches and to change and define the functionality, we used a lock type button NO, two push buttons NO and a signal lamp. The setup of the system and input-output configuration can be seen in the figures(see Figure 5.1).



Name		Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Input	X	1	BOOL	0.1	39.0	Input	0	A01_S01_PROXIMITY_C...
Input	X	0	BOOL	0.1	39.1	Input	0	A01_S02_PROXIMITY_C...
Input	X	0	BOOL	0.1	39.2	Input	0	A01_S03_LOCK_BUTTON...
Input	X	0	BOOL	0.1	39.3	Input	0	A01_S04_PUSH_BUTTON...
Input	X	0	BOOL	0.1	39.4	Input	0	A01_S05_PUSH_BUTTON...
Input		0	BOOL	0.1	39.5	Input	0	
Input		0	BOOL	0.1	39.6	Input	0	
Input		0	BOOL	0.1	39.7	Input	0	
WcState		0	BOOL	0.1	1522.1	Input	0	
InputToggle		1	BOOL	0.1	1524.1	Input	0	
State		0x0008 (8)	UINT	2.0	1550.0	Input	0	
Output	X	0	BOOL	0.1	39.0	Output	0	A01_H01_SIGNAL_LAMP...
Output	X	0	BOOL	0.1	39.1	Output	0	A01_K01_MOTOR_RUN_...
Output	X	0	BOOL	0.1	39.2	Output	0	A01_K02_MOTOR_DIREC...
Output		0	BOOL	0.1	39.3	Output	0	
Output		0	BOOL	0.1	39.4	Output	0	
Output		0	BOOL	0.1	39.5	Output	0	
Output		0	BOOL	0.1	39.6	Output	0	
Output		0	BOOL	0.1	39.7	Output	0	

Figure 5.1: System setup an I/O configurations

The two push buttons have two different of functionality depending on the mode. 1. When the lock button is on, push button 1 and 2 turn the motor counter-clockwise and clockwise, respectively.

2. When the lock button is off, the two push buttons increase and decrease an integer which will be displayed on the touch screen of the Industrial PC.
a. The number should not allowed to be smaller than 0 and larger than 10.
b. The signal lamp should shine 3 times when the number is tried to go below 0 or above 10.
c. This number is called "the number of repetitions".

When a soft button on the touch screen is clicked the motor should move between the two proximity switches, number of repetitions times.

5.2 Variable Declarations and Main Program Cycle

5.2.1 Global Variables

The global variable table shows the inputs and outputs used in the program. The hardware of our system constructs our global variable table(see Figure 5.2).

```
0001 VAR_GLOBAL
0002 (*INPUTS*)
0003     A01_S01_PROXIMITY_CCW_NO_DI AT %I*:BOOL;
0004     A01_S02_PROXIMITY_CCW_NO_DI AT %I*:BOOL;
0005     A01_S03_LOCK_BUTTON_NO_DI AT %I*:BOOL;
0006     A01_S04_PUSH_BUTTON_1_NO_DI AT %I*:BOOL;
0007     A01_S05_PUSH_BUTTON_2_NO_DI AT %I*:BOOL;
0008
0009 (*OUTPUTS*)
0010     A01_H01_SIGNAL_LAMP_DO AT %Q*:BOOL;
0011     A01_K01_MOTOR_RUN_COIL_DO AT %Q*:BOOL;
0012     A01_K02_MOTOR_DIRECTION_COIL_DO AT %Q*:BOOL;
0013
0014 END_VAR
0015
```

Figure 5.2: Global Variable Declaration

5.2.2 Main Program and Local Variables

The main program and local variables can be seen below. We divided our program into 4 actions as shown in the main program organization.

Listing 5.1: Main Program Cycle

```
h001_lock_button_on_handling();
h002_lock_button_off_handling();
h003_soft_button_handling();
h004_signal_lamp();
```

Each will handle different tasks which will be explained detailed in the following sections(see Figure 5.3).

5.3 Actions

5.3.1 Action 1: h001_lock_button_on_handling

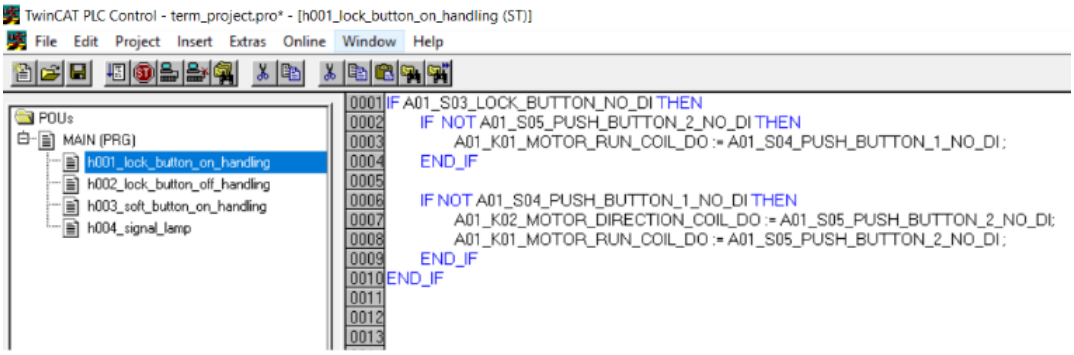


Figure 5.4: h001_lock_button_on_handling

In this action, we handle the cases which the lock button *(A01_S03_LOCK _BUTTON _NO_DI)* is on. In that case the two push buttons will be used to determine the motor turning direction. The motor only works when the any of the push buttons is kept pressed. So, we wrote the motor on command inside the both if statements *(A01_K01 _MOTOR _RUN _COIL_DO)*. When push button 1 *(A01_S04 _PUSH_BUTTON _1_NO_DI)* is pressed the motor should turn in counter clockwise direction. So we set *A01_K01 _MOTOR _RUN_COIL _DO* to the button that is pressed. When push button 2 (*A01_S05_PUSH_BUTTON_2_NO_DI*) is pressed the motor should turn in the clockwise direction. So, the *A01_K02 _MOTOR _DIRECTION _COIL _DO* should be on since the default behavior of the motor is to turn in CCW direction(see Figure 5.4).

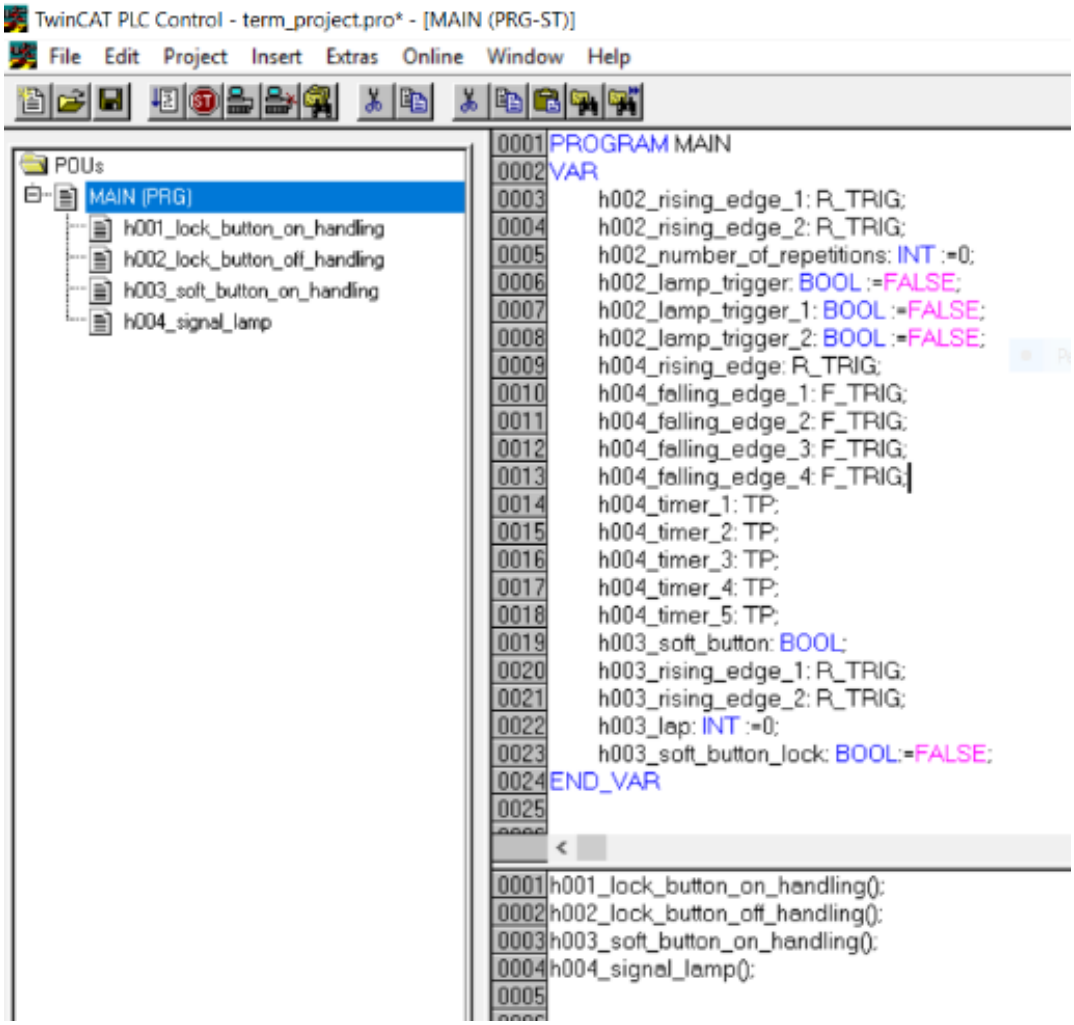


Figure 5.3: Main Cylcle and Local Variables

5.3.2 Action 2: h001_lock_button_off_handling

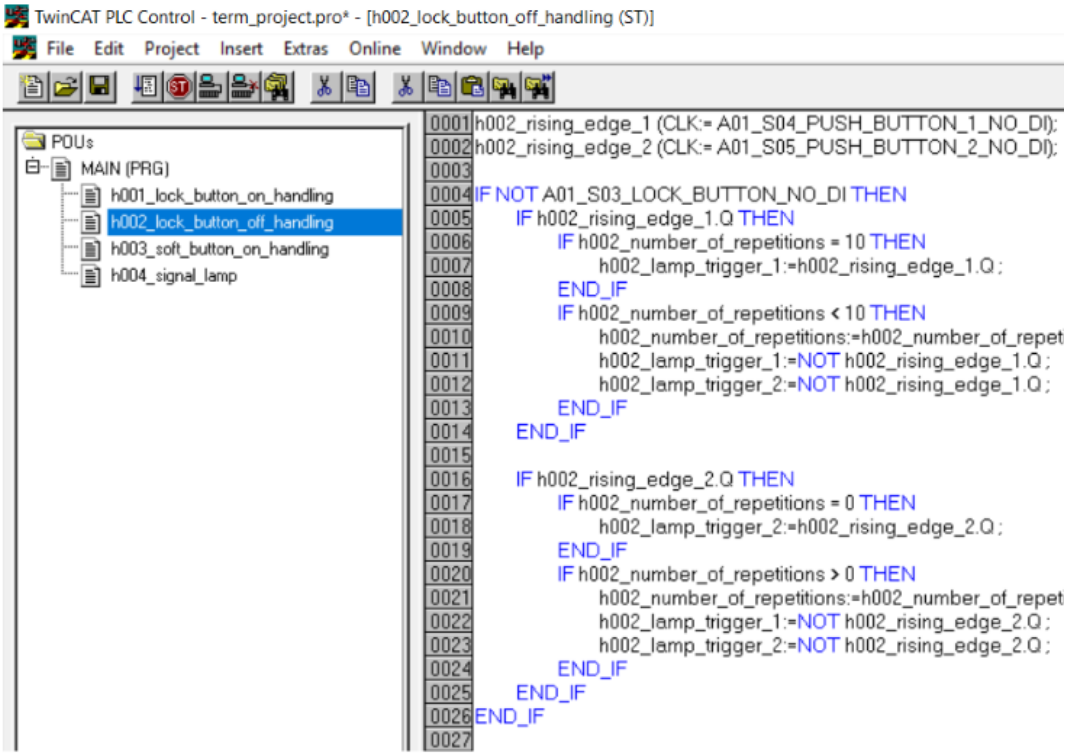


Figure 5.5: h002_lock_button_off_handling

In this action, we handle the cases which the lock button *(A01_S03 _ LOCK _BUTTON _NO _DI)* is off. In that case the two push buttons will be used to determine the number of repetitions of the motor. The default value of the number of repetitions *(h002 _number _of_repetitions)* is zero. When push button 1 *(A01 _S04 _PUSH _BUTTON _1 _NO_DI)* is pressed, number of repetitions is increased by 1 and when push button 2 *(A01 _S05 _PUSH _BUTTON _2 _NO _DI)* is pressed, it will be decreased by 1. But if the push button 1 is pressed when the number of repetitions reached its maximum value, 10, the system should not allow it to be 11 and the signal lamp should shine 3 times. Also if the button 2 is pressed when the number of repetitions reached its minimum value, 0, the system should not allow it to be -1 and the signal lamp should shine 3 times. The signal system works as a domino. First timer is triggered by the *h002 _lamp _trigger _1* and *h002 _lamp _trigger _2*. And the timers trigger each other. The signal system will be explained in detail in *h004 _signal _lamp* action. We defined two rising edge variables *h002 _rising _edge _1* and *h002 _rising _edge _2* to detect the push button 1 and 2 pressings respectively. The first if handles when the push button 1 pressed. If the number of repetitions

has reached its maximum value, *h002 _lamp _trigger _1* is on to activate the signal lamp system. If the number is between 0 and 10, the number of repetitions will be incremented by 1 and neither *h002 _lamp _trigger _1* nor *h002 _lamp _trigger _2* will be active. The second if handles the cases where the number of repetitions is zero and is tried to drop below zero. *h002 _lamp _trigger _2* is activated when this case happens, and it will be an intermediate factor to activate the signal system. If the number is between 0 and 10, the number of repetitions will be decremented by 1 and neither *h002_lamp _trigger _1* nor *h002 _lamp _trigger _2* will be active (see Figure 5.5).

5.3.3 Action 3: h003_soft_button_on_handling

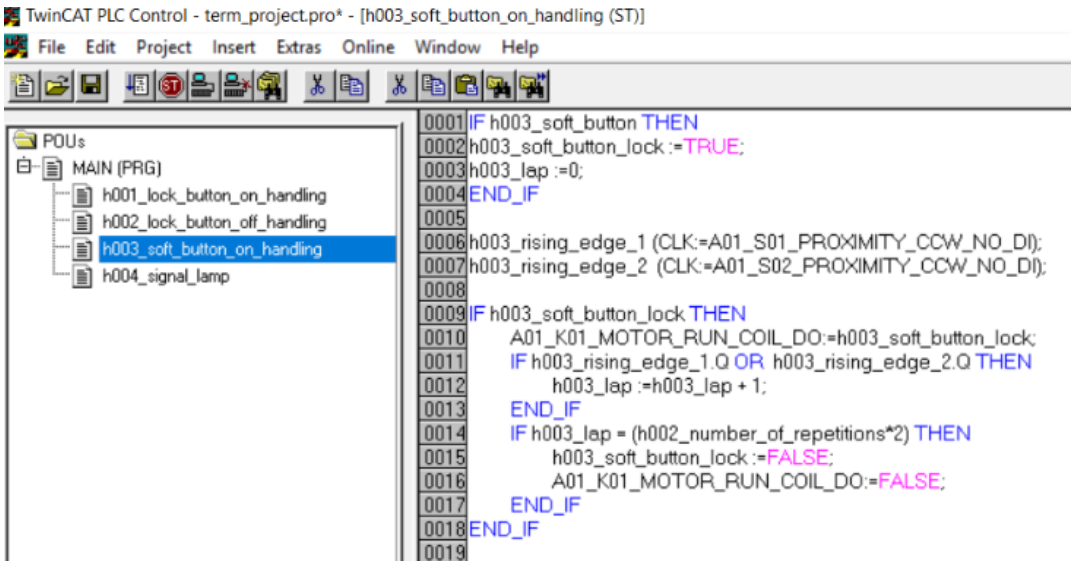


Figure 5.6: h003_soft_button_on_handling

On the touch screen of the control panel, we have a soft button *(h003_soft_button)*. When this button is pressed the motor should turn number of repetitions times. We keep the track of the laps by the two inductive NO PNP proximity switches. When the soft button is pressed, the laps the motor completed so far *(h003_lap)* is reset to zero and the variable *(h003 _soft _button _lock)* which we use to initiate the actions and keep the motor turning is set to true. We count the laps by the rising edges of the switches *(h003 _rising _edge _1 and h003 _rising _edge _2)*. To make the motor complete the laps with no error, we made use of both switches. If we have used only one of the switches, in some cases the motor would turn number of repetitions – 0.5 times. If *h003 _rising _edge _1* or *h003 _rising _edge _2* is on, the lap is incremented by one. And the motor stops when lap is equal to the twice of the *h002 _number _of _repetitions*. We set the *h003 _soft _button _lock* to false to prevent

it from getting in the loop and make it reusable every time(see Figure 5.6).

5.3.4 Action 4: h004_signal_lamp

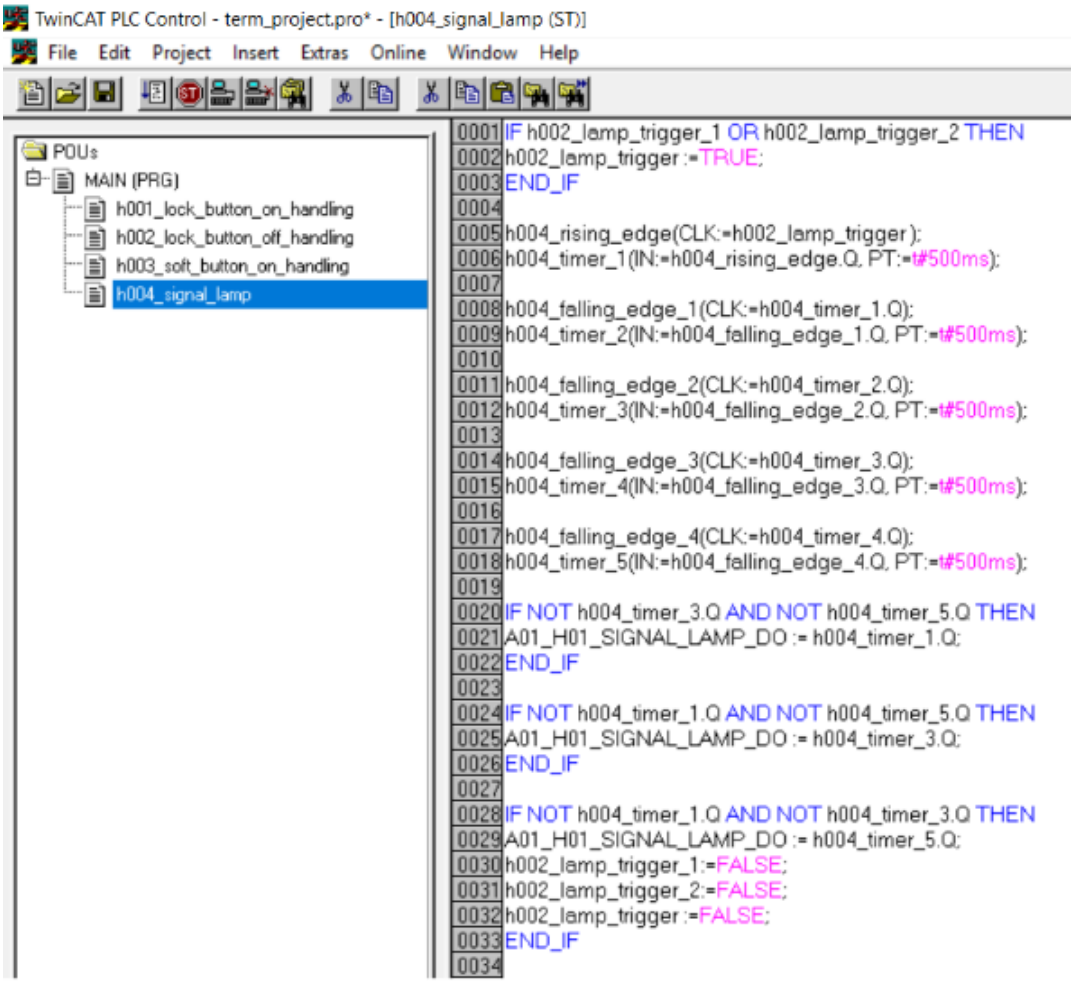


Figure 5.7: h004_signal_lamp

In this action, we handled the signal lamp shining operation. The trigger that is used to start the timer sequence is *h002 _lamp _trigger*. And it is activated as a result the limitations of number of repetitions explained in action *h002 _lock _button _off _handling*. We programmed the signal lamp *(A01 _H01 _SIGNAL _LAMP _DO)* shining such that it will shine for 0.5 seconds and it will go off for 0.5 seconds. *h004 _timer _1*, *h004 _timer _3* and *h004 _timer _5* are used for flashing the signal and *h004 _timer _2* and *h004 _timer _4* are used to turn the signal lamp off. The first

two if statements are designed to turn the lamp on by using the timer that is on at that instant. The last if statement is the condition which the lamp will shine for the last time, so we added some extra statements as well. When it shines for the third time, we must set each Boolean variable to its default value so that it can function properly. The timers are triggered by falling edge variables *h004 _falling _edge _1, h004 _falling _edge _2, h004 _falling _edge _3* and *h004 _falling _edge _4*. except the first timer (*h004 _time _1*). The rising edge of the *h002 _lamp _trigger* turns *h004 _rising _edge* on. And the rising edge of this variable is the input of the *h004 _timer _1*. The falling edge of this timer will set *h004 _falling _edge _1* to true and the input of the *h004 _timer _2* is on, which will turn the *h004 _timer _2* on. And the same operation is done for the other two timers as well.

5.4 Visualization of Touch panel

In the touch screen panel, we need to display the number of repetitions and the soft button which will activate the action *h003_soft_button_handling*. The graphical view of the touch screen can be seen in the figure below(see Figure 5.8).

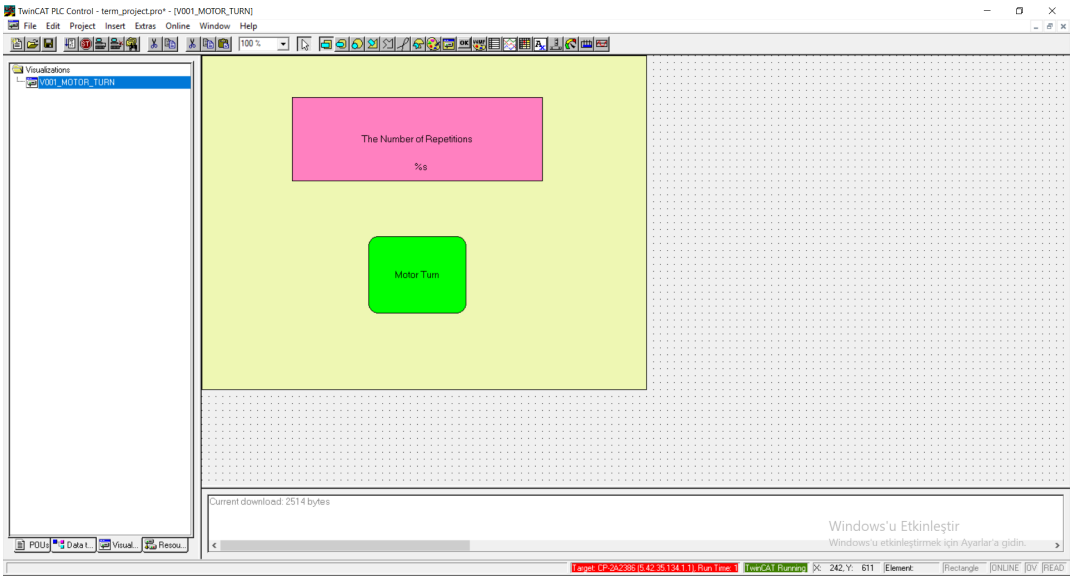


Figure 5.8: Layout of Touch Panel

We created this setup by configuring each box to a variable and a functionality as can be seen in the following figures. For The Number of Repetitions, we wrote the variables which represents the number into the Text display section.

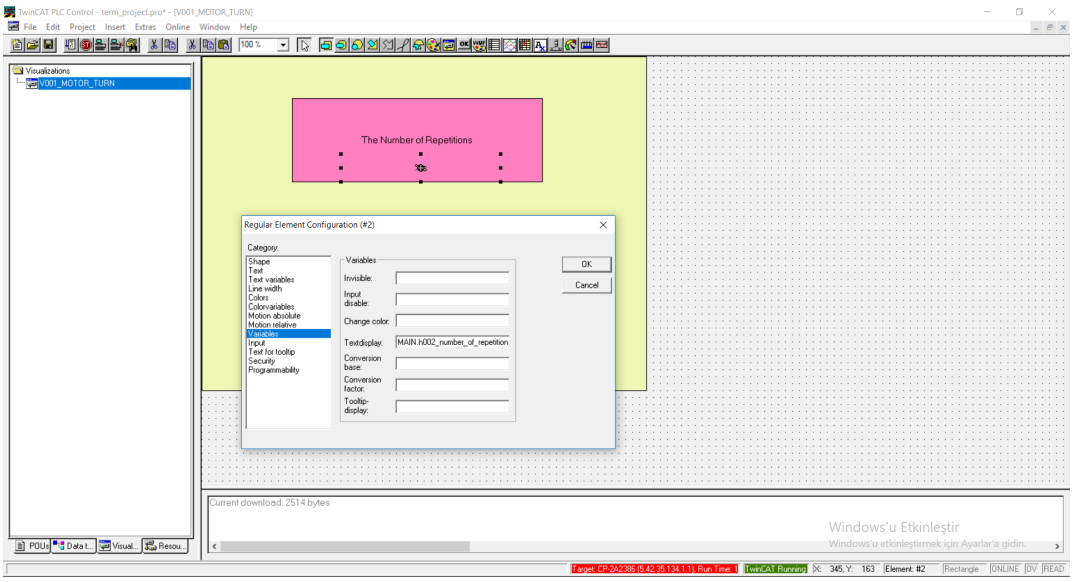


Figure 5.9: Printing Number of Repetitions

For the Motor Turn soft button, we set the Input as the *h003_soft_button* and a Tap Variable. Which means the soft button has only one functionality.

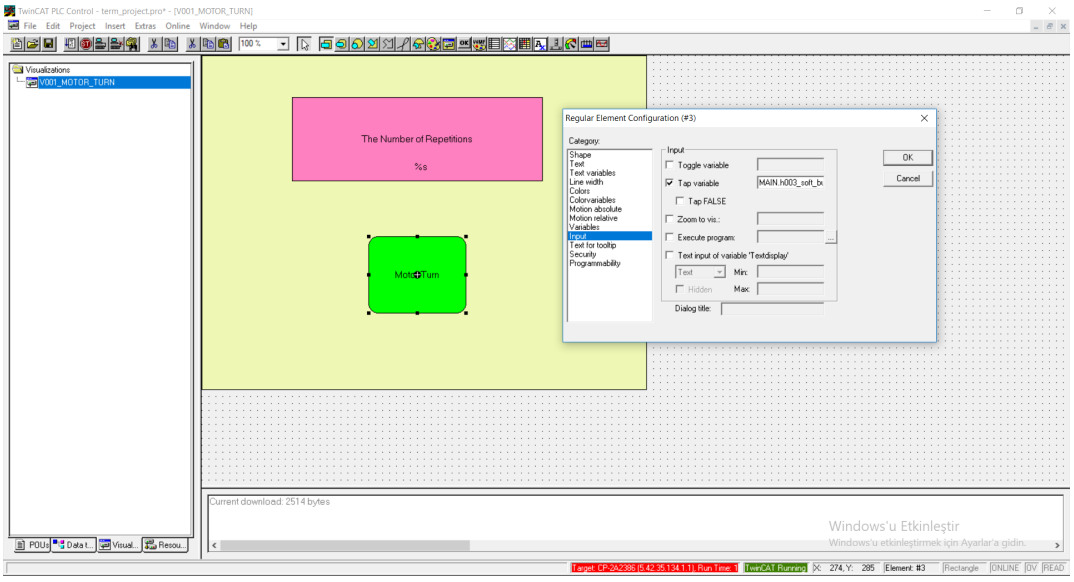


Figure 5.10: Configuration of Motor Turn Button

5.5 The Touch Panel Setting and Tests

When we run the code on our Industrial PC, the touch screen setting looked as in the figure below(see Figure 5.11).

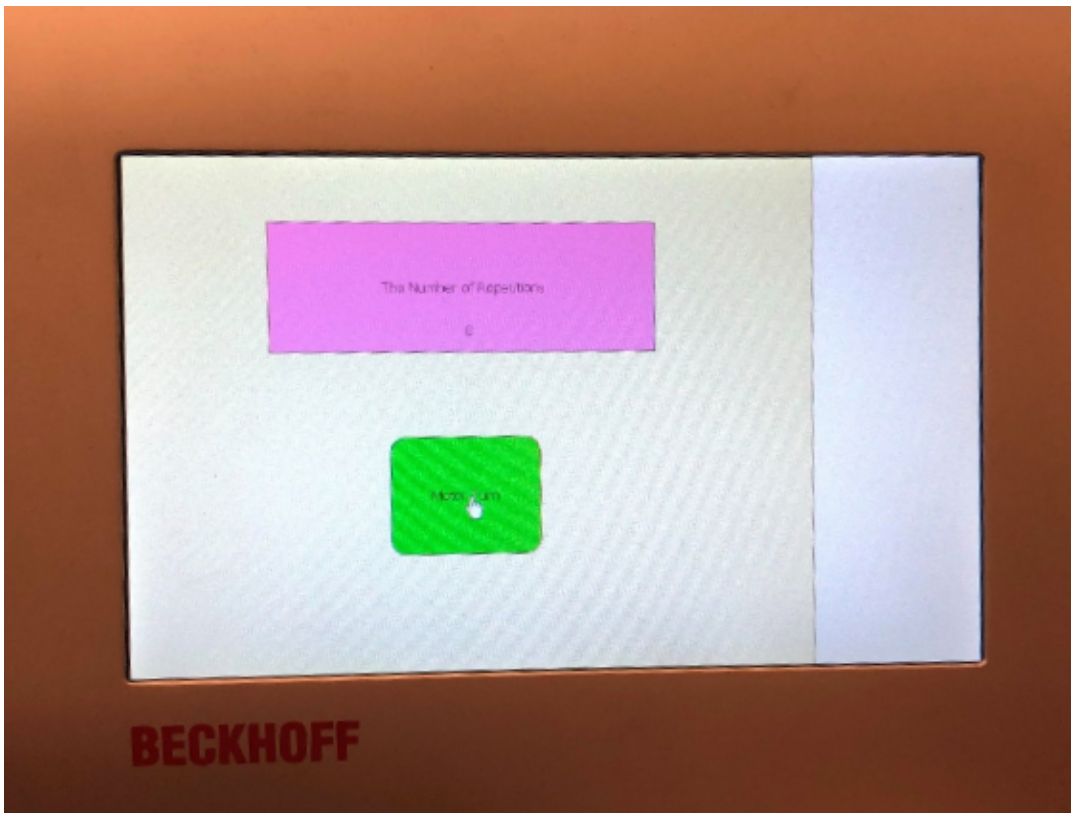


Figure 5.11

We tested our system with 3 cases;

1. When the number of repetitions is 0 and the push button 2 is pressed. (We expect that the signal lamp would shine)

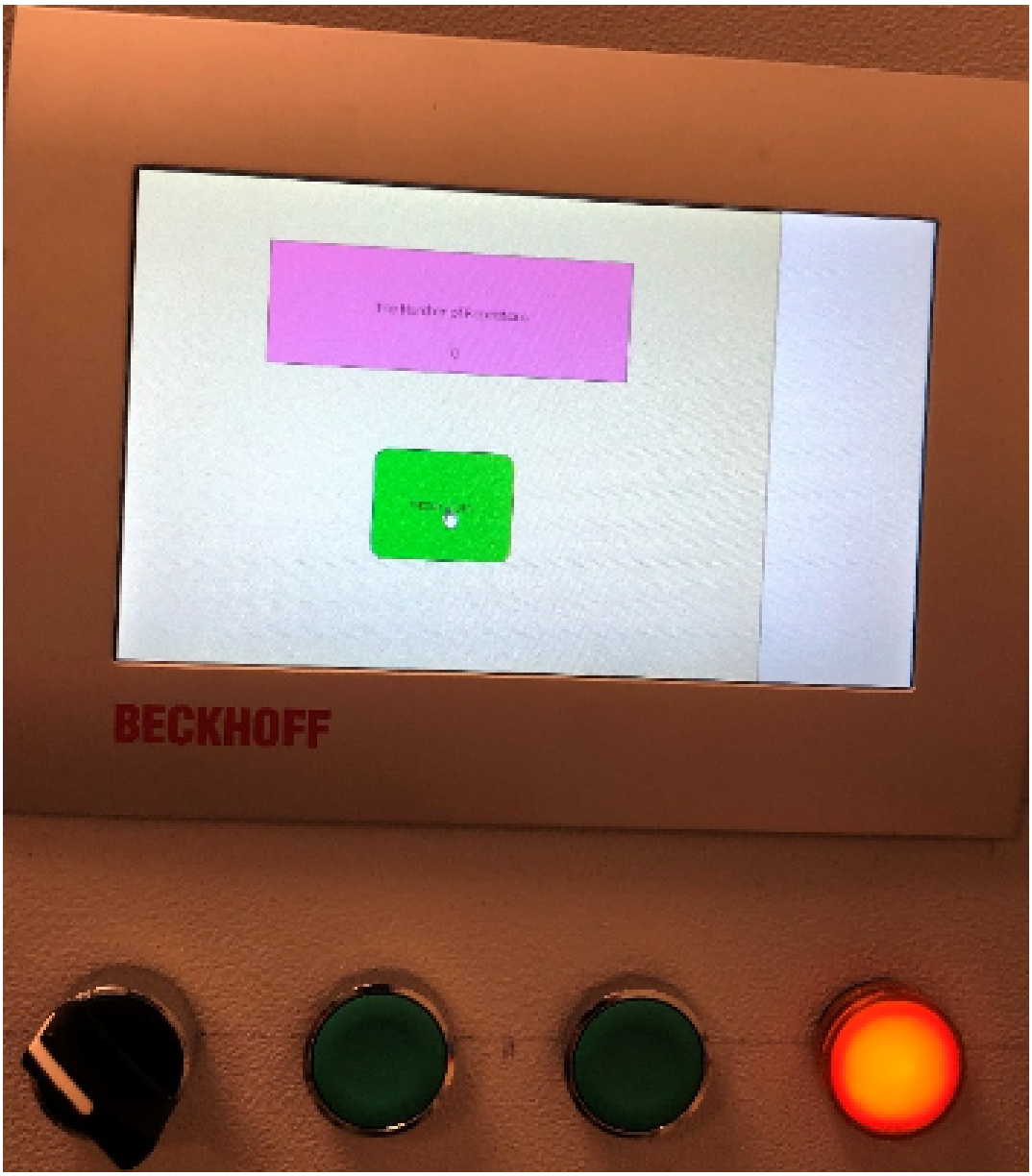


Figure 5.12

2. When the number of repetitions is 10 and the push button 1 is pressed. (We expect that the signal lamp would shine)

Figure 5.13

3. When the number of repetitions is between 0 and 10 (in the figure it is 1), and the any of the push button is pressed. (We expect that the signal lamp would not shine)

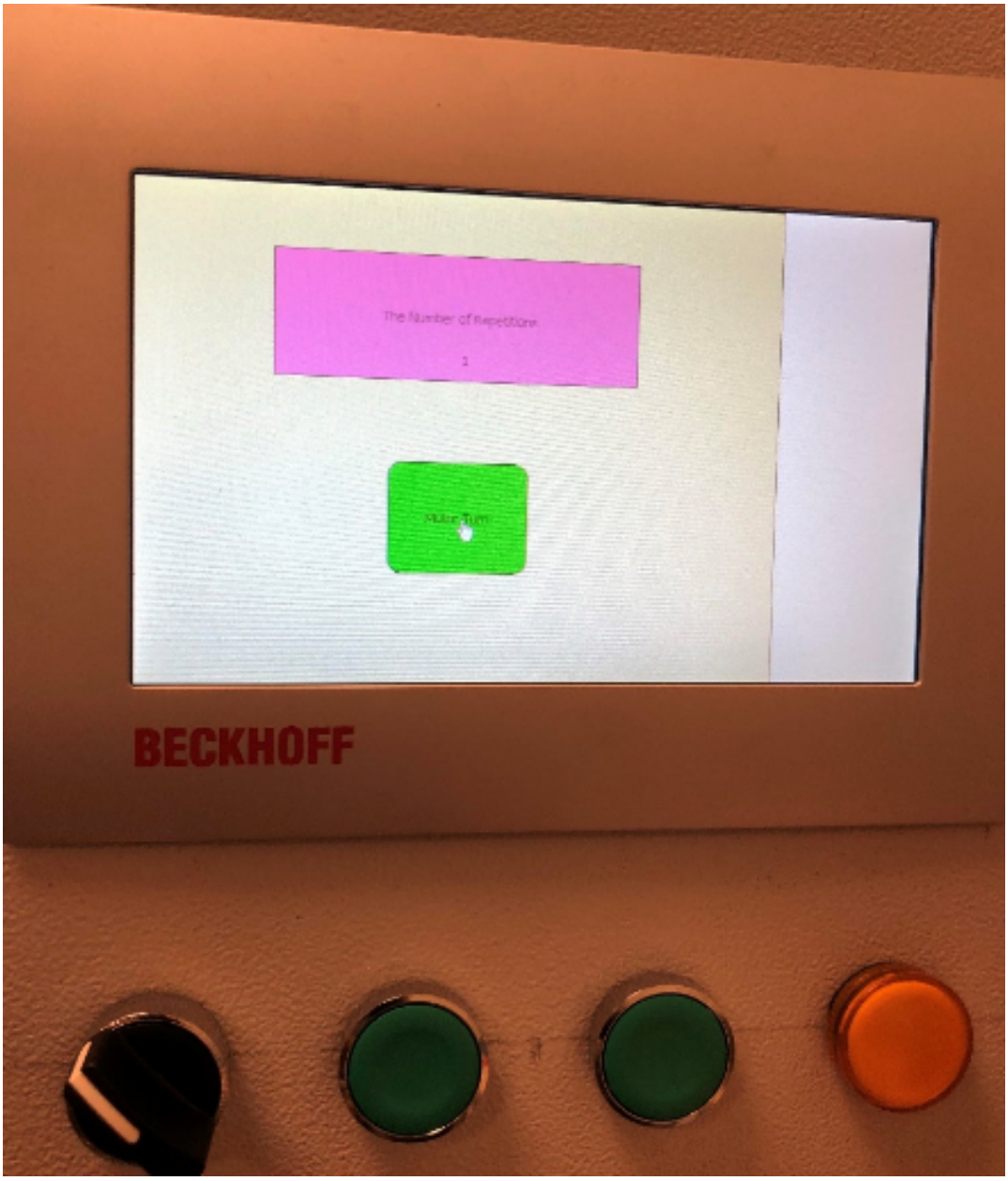


Figure 5.14

Chapter 6

Discussion

6.1 PLC

PLC is very commonly used industrial automation controller, which is programmed to do same specific processes. These are commonly used in car factories, food factories, etc. User could change the process by some minor changes in the system, which could be in the code or in wiring of PLC.

Variety of the sensors and other hardware configurations leads to change the system very much. There is wide range of sensor in the markets with wide range of price tags. Even the Industrial PC has very different types. So the question is "What is the perfect configuration for me ?"

6.2 Selecting PLC and Sensors

Selecting PLC can be differ by the sensitivity of the application. By the sensitivity, it could be any movement sensitivity ,pneumatic piston sensitivity, sensor sensitivity or any time sensitive applications. Some PLC's has faster running speed and some of them not. If you need any sensitive applications, you need to use those have faster running speed.

Also the compatibility with the other automation systems or any other devices could be one of the matters. For instance if the factory manager or operators need monitor the systems status, the system should compatible with the monitoring systems. Or the factory needs Human Machine Interface (HMI) the rest of the system should be compatible with that.

Input and Output module compatibility is another concern. Depending of the applications, number of inputs and outputs could be differ. Also when buying

that modules, the future needs should be thought, since these modules are significantly expensive. Another dependency is the input and output voltages. There are several options for that purpose too.

As written before there is a wide range of sensors in market. So when you buy randomly one of them, it wouldn't fit with your application. To encounter that kinds of problems, reading the data-sheets of the sensor could be good start. In data-sheets the values of accuracy, working distance, working voltages could be found. With that values the perfect sensor could be found.

The other considerable point is the physical limits. For example if you planning to drop an object to the certain point, you need to calculate how much time does it take between drop point and arrival point. If the system drops object when the cart or any mechanism is in the arrival point object may not catch the cart since the travel time is very long.
On the other hand this problem could be related with the temperature. The systems that need to stay cold needs to produced by lower heat capable materials. Basic mechanics knowledge is could encounter that kinds of problems.