School of Engineering

MRes General Engineering

PROJECT REPORT

Title:	**Centralized Control System for Cement Plant**
Author:	**MUHAMMAD FURQAN**
Academic session:	**2021 – 2022**

Supervisor:	**Dr. Perry Xiao** **Engineering, Science and the Built Environment** **London South Bank University**
Course Title:	**MRes General Engineering**
Mode of Study:	**Full Time**

Title:	**CENTRALIZED CONTROL SYSTEM FOR CEMENT PLANT**
Advisor:	**DR PERRY XIAO**
Author:	**MUHAMMAD FURQAN**
ID:	**2929244**
Course:	**MRES GENERAL ENGINEERING**
Academic Session:	**2021 – 2022**
Mode of Study:	**FULL TIME**
Submission Date:	**13 June 2022**

This report has been submitted for the assessment towards a Master of Science degree in the school of Engineering, London South Bank University.
This report is written in the author’s own words and all sources have been properly cited.

TABLE OF CONTENT

INDEX	PAGE NO.
Acknowledgement	
Abstract	
Preface	
Chapter 1…Theoretical Background	
1.1 Introduction 1.2 System Background 1.3 System Objective 1.4 Reasons behind project selection 1.5 Key Features 1.6 Block Diagram 1.7 Project Duration	01 03 03 04 04 05 05
Chapter 2... Hardware Narration	
2.1 Simulator	07
Chapter 3… PLC Programming (Simatic Step 7)	
3.1 Introduction 3.2 Hardware 3.3 Software 3.4 Group Motors 3.5 Alarm	11 11 12 20 20
Chapter 4… PLC Communication	
4.1 Communication Hardware Configuration 4.2 Communication Software Configuration 4.3 Data Exchange	22 24 27
Chapter 5… WinCC	
5.1 Introduction 5.2 Screen Configuration 5.3 FCS01 Screen 5.4 FCS02 Screen 5.5 FCS03 Screen 5.6 System Validation 5.7 Future Enhancement	29 29 31 33 34 35 35

Chapter 6… Tag & Alarm Logging	
6.1 Tag Logging 6.2 Alarm Logging	36 38
List Of Figures References Appendices	

Abstract

Centralised Control System for Cement Plant is a control system with multiple Siemens PLC CPUs, each CPU has its own control station called Field Control Station (FCS). There are 3 FCS altogether communicating each other and sharing data through Ethernet communication system as defined in its sharing Data Block (DB). Each FCS has multiple I/O cards (Digital Input, Digital Output, Analog Input, Analog Output) and CPU.

Cement plant has different manufacturing stages, first one is Limestone Crusher which is getting controlled by FCS named FCS01, second is Raw Mill Grinding (RMG) for grinding limestone with different impurities and its using FCS02. Third stage 5 is Kiln system for burning the grinded feed to make the Clinker (Cement) and its using FCS03. Each FCS is at some meter distance from each other.

All the FCS are communicating through Ethernet and fibre optics communication. It can share any data up to 400 bytes at a time. In this project I am sharing 20 bytes to demonstrate the communication. Some of the data are binary control signal and some of them are analogue parameters, PLCs will take decision depends on the data they receive from other PLC.

Due to unavailability of physical PLC, I am simulating my project on Siemens simulator, here I have used Siemens PLC CPU S7 -400 CPUs with each FCS and each FCS has 32 input DI card, 32 output DO card and different AI/AO card with respect to the requirement. I have used Simatic Step7 to program the multiple PLCs and using WinCC to design mimic on the screen. I have made one screen with multiple tabs to access particular FCS data. Operator who is sitting in Common Control Room (CCR) can see all the data (Analog, Digital) in the CCR.

Each Motor circuit has multiple parameters and control signals that operator can see in CCR and can give start/stop command to any motor from the mimic in the CCR. These control signals come from the electrical cabinets to the PLC and I have designed the PLC in a way that if all control signals fulfil its condition then PLC with accept operator command or else PLC will notify operator about any particular signal missing or not available. After taking command, if any control signal malfunctioned then PLC will trip the motor and will generate an alarm to notify operator why it happened.

This control system is also logging all the analogue parameters for nearly up to 3 months and if operator wanted to see any analogue parameters activity can see by going into the trend window and there, operator needs to put the particular date and time to see the graphical representation. This system also logs all the alarms for up to 1000 alarms and if operator want then he can go in alarm list and see how many times alarm occurred and even he can search any particular alarm in the list with alarm number.

ACKNOWLEDGEMENT

I began with the name of **Almighty Allah (GOD)** the ever beneficent, the ever merciful. It is **HE** who gave me the strength and courage to walk through times of utter desperation and stand up against immense odds, without **THEE** I would have been lost.

I express my deep sense of indebt ness with sincere gratitude to my internal advisor Dr Perry Xiao, London South Bank University, from whom I sought invaluable advice and encouragement to complete the research project. He provided every possible guidance in designing this control system and channelized all my efforts to achieve the cherished goal. Dr Perry Xiao is my former teacher in SCADA SYSTEM and DISTRIBUTED CONTROL SYSTEM in Msc Embedded & Distributed System. My previous education with my professional 7-year experience provide a foundation to develop my knowledge in developing control system.

I also would like to acknowledge the tedious hard work of my colleagues and ex-boss who backed me to learned PLC automation and when I was coming to UK for this research course. I learnt the true value of my mettle and short coming. The successful compeltition of this project is an outcome of effort co-operation of advisor Dr Perry Xiao.

I would also like to express my gratefulness to Siemens to produce a simulator with the Step7 software to demonstrate the project or otherwise I wouldn't be able to work or trouble shoot my projects.

Last but not the least I acknowledge the backing lent to me by my parents who gave me the courage to stand aloft in times of fear and encouraged us to go beyond the expectations of all my seniors making me soar towards heights I never thought possible.

PREFACE

One feels that a text should serve not only for the benefits of the reader but also as a pedagogically sound outline for a course of instruction. A text should be sufficiently clear to enable the reader to understand the material well by its reading, with a realism that approaches hands-on experience. I also think that the text should be more comprehensive than the course for which it is used; thus some material can be used both as a reference source and as a source of further examples and illustrations by the student and engineer. Finally, I feel that the text should be able to stand alone, with minimal need of supplement documentation and references. We hope that each student and instructor finds that all these objectives have been met in this text. The Project report of "Centralized Control System for Cement Plant" includes complete knowledge of the project which describes the devices used in the formation and also associated with it. The text also elaborates the terminologies and factors related to the project. Calculations have been kept to the minimum so the reader will learn the central concept being presented rather than be intimated by a torrent of math. The distribution of chapters is organized in a way to make it easier for the reader to become well at home with the basic idea of the project.

Chapter 1 gives sufficient amount of introduction and theoretical background of the project and reflects the discussion regarding the overall working of the project and the task performed by different hardware and software components

Chapter 2 describes the hardware configuration, hardware and its components details that are being used in the project.

Chapter 3 gives comprehensive information of software programming of PLC, function, function block, data blocks and its configurations.

Chapter 4 gives general information of the communication being used in the project between all the CPUs and its configuration.

Chapter 5 describes the WinCC screen configuration, its settings and brief overview of WinCC screen.

Chapter 6 gives the brief summary of the Tag logging and Alarm logging in the WinCC project.

Chapter 1
Theoretical Background

1.1 Introduction

This project is a practical implementation of PLC automation with Siemens S7-400 CPU on modern technologies. It is a project which provides complete automated plant into small screen where operator can see and analyze all the data. The implementation of this project replaces the hardwired relay logic that have been previous used in many process plants. It avoids all the manual control panels that operator had to operate every single equipment.

In this project I have worked on how to use the modern technologies in the most efficient manner and can produce sent percent desired result. The basic purpose of this project is to avoid hardwired relay logic on which multiple relay used to make 1 single input which was very hard to troubleshoot or diagnose the problem. With the growing era where modern technology has taken place to replace the traditional system in all aspects of life, this is the high time to use the modern technologies for monitoring and controlling applications as well.

The basic idea of the project is to provide automated control system using S7-400 CPU which is a Siemens PLC CPU. Due to the unavailability of expensive PLC CPU and its I/o card, I am going to simulate this project in simulator on Siemens Simatic simulator and to program this CPU, I am going to use Simatic Step7. In this software there is a section called “Hardware” in which I can define the CPU and its hardware card. Suppose if I am going to use 1 digital input (DI) card then I need to define it in the hardware so CPU could recognize it and expect to get data compatible with this DI card. All the hardware I am going to use in project I need to define it in its hardware section or else CPU will go on stop mode from run mode.

My project is designed on the thought of achieving the target of providing complete automation and controlling in cement plant or any other process industry. Operator can control the whole plant from his computer screen and will be analyzing all the plant data in the blink of an eye.

The whole project is a PLC based prototype in which CPU is the central controlling component of the project. I am going to use DI card for digital input, DO card for digital output, AI card for analog input and AO card for analog output since I am using simulator to simulate the software I can use any hardware from the library list of I/O cards. I have configured all hardware in the hardware section and download it into the simulator. After this I used "Block" section to do the PLC programming, there I created multiple functions and data blocks for programming purpose. My project is divided in to two modules,

- Simatic Step 7 for PLC Programming
- Simatic WinCC for virtual screen (Mimic) to analyze data in CCR.

Simatic WinCC is also a Siemens software which actually communicate directly to the PLC without any involvement of Step 7 and in my case WinCC will directly communicate to the simulator. Simatic WinCC communicate to the PLC through tags that I need to make of all the addresses in step7 which are used in programming in CPU. To make any mimic, I needed to create a screen and put some motors or graphical pictures from its library to show some process on the screen. Tags usually link all addresses that are being downloaded in the CPU. Apart from this, I can log all the analog parameters in tag logging for as long as I can store it in my hard drive with 0.1 sec acquisition time. I can log all the alarms or faults in alarm logging as well.

1.2 System Background

The oldest and traditional control system is cascade relay based control system which includes multiple relays for a single binary input and it was very hard to diagnose or troubleshoot. If operator want to start any motor, then operator needs to start it through the mechanical switch from the electrical cabinet. For analog parameter, mechanical mechanism sensor has always been used to see the analog readings and operator has to go on multiple places to see the sensor readings. Everything was manual to operate or start any motor and on top of it operator was unable to see the 10% of the parameters. I have put references of some examples explaining hardwired relay logic and it used to have relay ladder logic to perform the task.

The concept of introducing Centralized control system with PLC automation is the solution to the problems which produces by the old relay logic control systems. Though human being is superior than machine still machine is more accurate since it doesn't have mood characteristic as human do.

The "Centralized Control System for Cement Plant" is the implementation of modern technologies for monitoring and controlling purpose and replaces the traditional system of analyzing and controlling system. After completing this project, operator don't need to run in different places to see the parameters because all the parameters of plant and all control signals would be coming in one single screen and operator can give command to any motor from the CCR, doesn't matters if this motor is 2 KM far away from him.

1.3 System Objective

The main objective of Centralized Control System is to provide an automated control system that uses modern technologies and provides a system in which a good monitoring and controlling of cement process. A system which asks for the less availability of the operator and must have the ability to takes decision depending on operating mode. This system will log all the analog parameters for past parameters analyzing and can also log all the alarm/fault in running plant.

1.4 Reasons behind Project Selection

Cement plant is usually based on wide area and it got different stages to turn limestone with different impurities into cement clinker. Traditionally there used to be more than 5 operators to run only one section of the plant because all the places are far from each other. This project replaces the whole scenario by putting PLC station in every section of the plant and we named it FCS (Field Control Station). PLC Systems is more reliable with their solid-state components tend to last longer than the moving parts of electromechanical relays. This project help in diagnosis of fault as there are less wires comparatively than relay system and if I want to add some functionality in the project I can do it easily in programming, I don’t need to add physical component and its wiring into the project. And if we talk about space required for PLC system vs the cabinet needed for a relay logic circuitry is much smaller.

1.5 Key Features

- ➢ It will have three field control stations (FCS) to automate three different stages of cement plant.
- ➢ All three FCS will communicate each other through Ethernet/Fibre communication to share data.
- ➢ It will collect all the data from all three stages of cement plant and will show it on WinCC screen where operator can operate the plant.
- ➢ It will show the complete running status running analog parameters of the plant
- ➢ This WinCC software will store all the analog parameters on the sampling period of 1 sec and can see all the data for previous dates.
- ➢ This WinCC software will also log all the alarm that will generate through the process and it can be seen in alarm list for up to 1000 alarms.

- CPU of each station will take its own decision depending upon analog parameters and binary control signals, if something goes wrong then it will securely trip the system without any failure.

1.6 Block Diagram

Figure 1.1 shows the block diagram of all three FCS communicating each other and how each FCS has it's IOs (inputs/outputs). All three FCS are communicating through Ethernet communication.

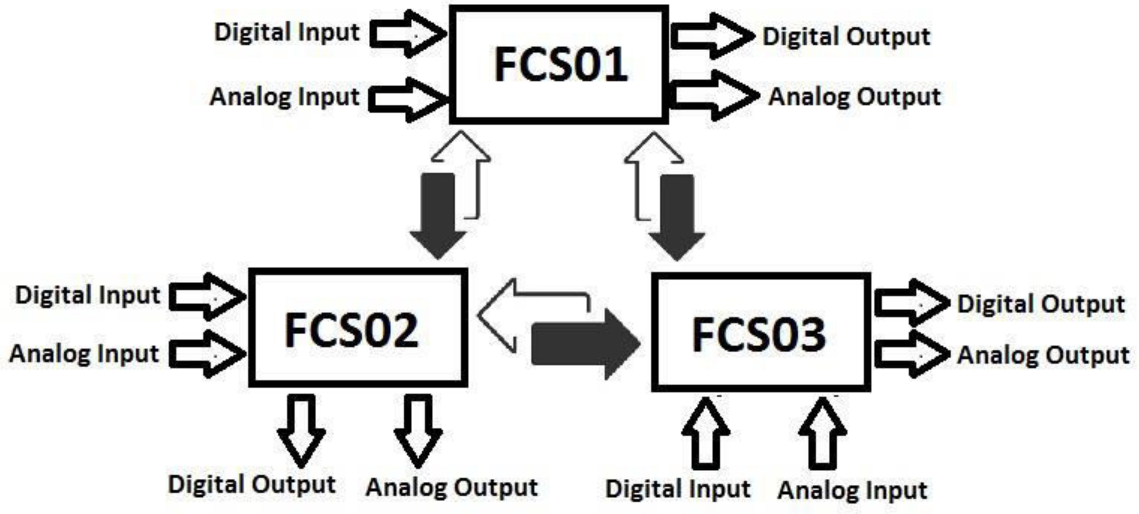


Figure 1.1 (System Block Diagram)

1.7 Project Duration

I started working on this project from the month of October 2021. In the month of September 2021 I did background study of this project, I decided what components I am going to use in this project and decided what software I would use to automate the cement plant. In the month of November 2021, I started working on basic of step7 and WinCC software and design some small project as a practice. In the month of December 2021, I completed working on logic in step

7 for FCS01. After testing complete logic, started working in WinCC for FCS01 by the end of December 2021 and completed WinCC screen work of FCS01 by the end of January 2022.

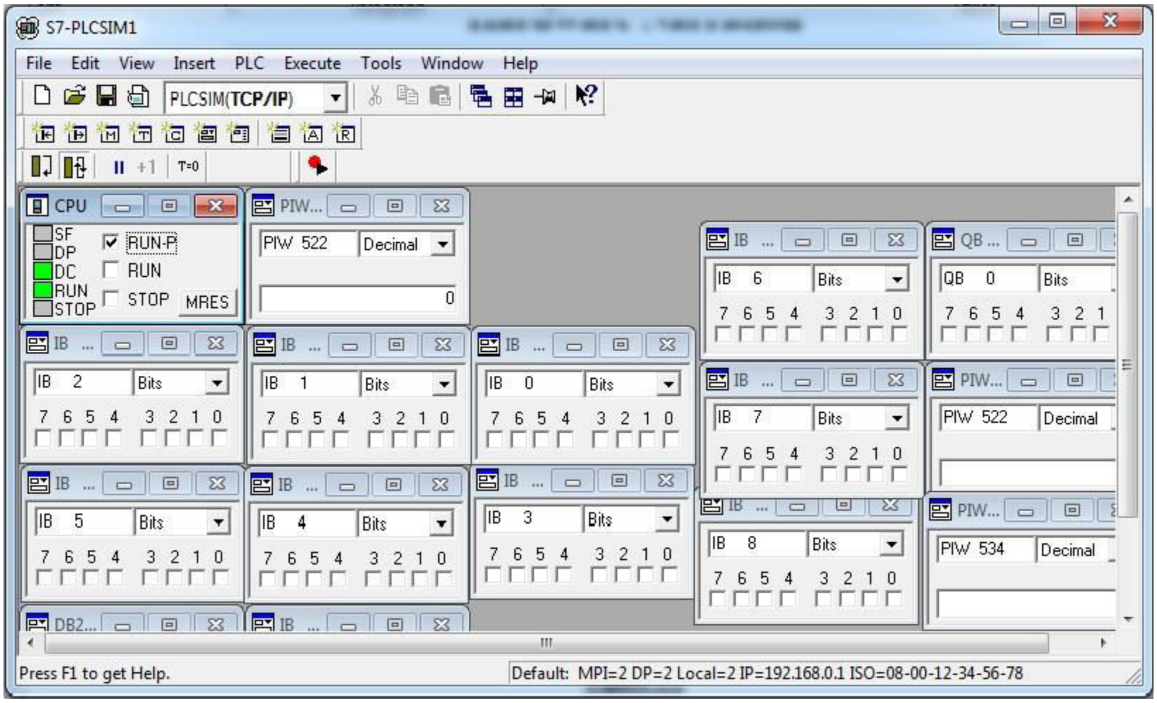
In the beginning of February 2022, started working on Step7 logic in FCS02 and after completing it, started working on WinCC screen of FCS02 which was completed by the end of March 2022. In early April 2022, started working on FCS03 step7 logic which took quite a lot of time as it has a lot of analog parameters and motors so took many days to complete FCS03 Step 7 including WinCC screen of FCS03.

In the end of may2022, I started working on my report and it was also completed in 20 days including notebook which I was writing by the time with my project.

Chapter 2
Hardware Narration

2.1 Simulator

The basic tool I am going to use to demonstrate or run my project is Simatic Step7 Simulator, without this tool I would need to spend thousands of Pound to buy multiple PLCs and their I/O cards. As it can be seen in figure 2.1 of simulator, it demonstrates the actual appearance of one simulator and I can download any Siemens PLC I want. I can run online or check ay code by running it online. It has unlimited I/O both analog and digital and once I download my step7 code in it, it behaves exactly like normal PLC. I can open more than one simulator to download multiple projects in different PLCs and communicate them in through different media. As we can see in figure, there are two ways of putting the PLC in RUN mode (RUN & RUN-P), with RUN mode PLC will run as normal but we can only upload the code from the PLC but with RUN-P, we can download and upload the PLC



Press F1 to get Help.

Default: MPI=2 DP=2 Local=2 IP=192.168.0.1 ISO=08-00-12-34-56-78

<u>Figure 2.1 (Simulator)</u>

The reason of choosing different CPUs is to show communication with different CPUs, First CPU I have used in FCS1 is CPU416F-3 PN/DP, here PN/DP means it can communicate through three modes; Ethernet, DP (Decentralised Port) communication and MPI (multiple purpose interface) communication. Second CPU I have used in FCS02 is 414F-3 PN/DP which quite different from first CPU in terms of code memory and data memory. Third CPU I have used in FCS03 is CPU 412-2 PN which is different in terms of access ports (Ethernet & MPI) and memory. Here is the list of components that I am using in this project with their properties and limitations.

These are the details of CPU 1(416-3 PN/DP) which is being used in FCS01

Communication Mode	Yes; Via PROFIBUS DP or PROFINET interface
Programming package	STEP 7 V5.5 or higher
Work Memory	● integrated (for program) : 8 Mbyte ● integrated (for data): 8 Mbyte ● expandable : No
CPU Processing time	for bit operations, typ. 12.5 ns for word operations, typ. 12.5 ns for fixed point arithmetic, typ. 12.5 ns for floating point arithmetic, typ. 25 ns
CPU Blocks	FC: 5000 FB:5000 DB:10000 OB: depends upon the process
S7 Counters	Numbers: 2048 Count range : 0~999
S7 Timers	Numbers: 2048 Timer Range: 10 mS ~ 9990 S

These are the details of CPU 2(414-3 PN/DP) which is being used in FCS02

Communication Mode	Yes; Via PROFIBUS DP or PROFINET interface
Programming package	STEP 7 V5.5 or higher
Work Memory	● integrated (for program): 2 Mbyte ● integrated (for data): 2 Mbyte ● expandable : No
CPU Processing time	for bit operations, typ. 18.75 ns for word operations, typ. 18.75 ns for fixed point arithmetic, typ. 18.75 ns for floating point arithmetic, typ. 37.5 ns
CPU Blocks	FC: 3000 FB:3000

	DB:6000 OB: depends upon the process
S7 Counters	Numbers: 2048 Count range : 0~999
S7 Timers	Numbers: 2048 Timer Range: 10 mS ~ 9990 S

These are the details of CPU 3(412-2 PN) which is being used in FCS03

Communication Mode	PROFIBUS DP or PROFINET interface
Programming package	STEP 7 V5.5 or higher
Work Memory	● integrated (for program): 0.5 Mbyte ● integrated (for data): 0.5 Mbyte ● expandable : No
CPU Processing time	for bit operations, typ. 75 ns for word operations, typ. 75 ns for fixed point arithmetic, typ. 75 ns for floating point arithmetic, typ. 225 ns
CPU Blocks	FC: 500 FB:500 DB:3000 OB: depends upon the process
S7 Counters	Numbers: 2048 Count range : 0~999
S7 Timers	Numbers: 2048 Timer Range: 10 mS ~ 9990 S

DI (SIMATIC -400, digital input SM 421, isolated 32 DI; 120 V DC/AC)

Number of digital inputs	32
Number of simultaneously controllable inputs	32
Power Loss	6.5W
Input Voltage	AC/DC
Transition time delay	5mS~25mS for 0 to 1

DO (SIMATIC -400, digital output SM 422, isolated 32 DO; 24 V DC)

Number of digital outputs	32
Number of simultaneously controllable	32

outputs	
Supply Voltage	(20.4V~28.8V) average is 24V
Input Voltage	AC/DC
Power Loss	8W

AI (analog input SM 431, isolated 16 AI; resolution 16 bit)

Number of Analog inputs for voltage and current measurement	16
Number of Analog inputs for resistance measurement	8
Supply Voltage	24 V; Only required for supplying 2-wire transmitters
Input Ranges	Current, Voltage, thermocouple, RTD, resistence
Power Loss	4.5W

AO (analog output SM 432, isolated 8 AO; resolution 13 bit)

Number of Analog outputs	8
Output supply	Voltage & Current
Supply Voltage	24 V
Output Ranges	● 0 to 10 V ● 1 V to 5 V ● -10 V to +10 V ● 0 to 20 mA ● -20 mA to +20 mA ● 4 mA to 20 mA
Power Loss	9 W

List of acronyms:

FB	Function block, this blocks are for logical connections between signals and variables. You need a instance DB, where you can storage internal variables as static
FC	Function, this block type is the same like FB only without the instance DB. You have only local variables
DB	Data block, this block is for storing variables
OB	Organisation block, this blocks are for different tasks, program organisation, hardware faults, software faults, cyclic interrupt and a lot more. You need OB1 as Main OB for the program organisation.

Chapter 3

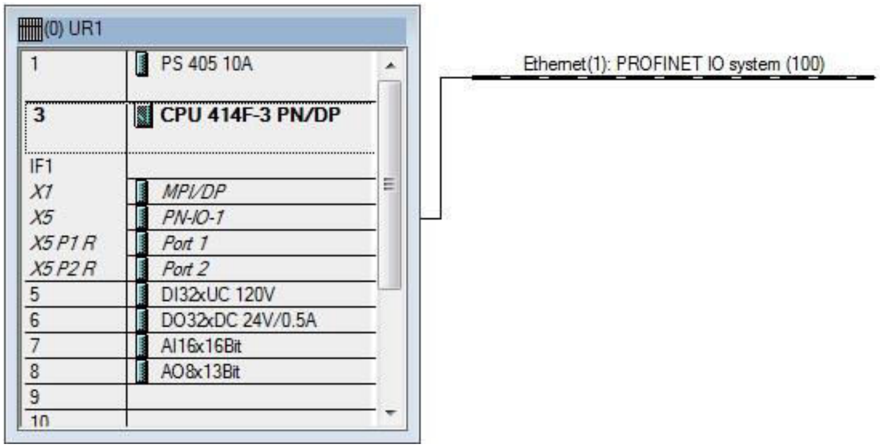
PLC Programming (Simatic Step7)

3.1 Introduction

Siemens PLC are mostly programmed by their own software which is Simatic Step7, there is one more Siemens software TIA Portal which is new software platform for configure and program S7-300/400/1200/1500 whereas STEP 7 is only applicable to S7-300/400. TIA Portal doesn't support all hardware in range of STEP 7 specifically old S7-300/400 modules. In this project I have used step7 with its simulator and I am using three main control stations (FCS01, FCS02, FCS03) so I have made a project in STEP7 which has three different control stations in it and all of them are controlling their own control systems. On top of it, all the control stations are communicating each other and sharing data as well. I will discuss communication system in next chapter, I will discuss about step7 and its programing in this chapter.

3.2 Hardware

When I created this project, first I needed to configure hardware in its hardware section where I needed to tell the software what hardware I would be using in the field. In its hardware section I selected PLC rack, CPU, I/O cards etc as you see in the picture.



(0) UR1

1	PS 405 10A
3	**CPU 414F-3 PN/DP**
IF1	
X1	*MPI/DP*
X5	*PN-IO-1*
X5 P1 R	*Port 1*
X5 P2 R	*Port 2*
5	DI32xUC 120V
6	DO32xDC 24V/0.5A
7	AI16x16Bit
8	AO8x13Bit
9	
10	

Figure3.1 (Hardware Configuration)

Siemens S7-400 rack consist of 18 slots altogether on which I can attached any hardware card to linked it with PLC. This rack actually helps every module to communicate each other. In FCS01 slot 1&2, I have connected power supply of PLC, which is associated with power supply only and I can only connect power supply to these slots. From slot three, I can connect any module I want, it can be any I/O card, CP card (Communication Processor), CPU etc. I have connected CPU 416-3 PN/DP to slot 3 & 4, rest of the slots I have used for I/O cards.

On Slot 5 & 6, I have used two DI cards with 32 inputs each and I will operate on 120V AC/DC input. On slot 7, DO card is connected and it will generate 24V DC from its port to operate any motor. On slot 8 & 9, AI and AO card are connected to take analog input to PLC and take PLC set point to the field. AI can be configured in milli ampere, milli volts, resistance and thermocouple. I have created other FCS hardware same way as I created this FCS hardware.

Next step to program the PLC using FC and FB, these are functions in PLC programming which I can use it just by calling it in the programing. In my project, there are many motors that I need to operate and this function ease my work by calling it each time when I need to program any motor logic. Design a motor circuit is a set of multiple circuits which I don’t need to make it again and again in every motor logic. If I need to create a motor logic with memory block which can store parameters on memory, then I have used FB and if I don’t need to use memory then I have FC which is function without memory block.

3.3 Software

When turn on the PLC CPU, CPU executes is startup routine and this routine perform three task,

- Clear input area (I Memory) of process image
- Initialized last value of outputs
- Any interrupt waiting in the queue during run mode.

After startup, CPU executes startup OB called OB100, to initialize registers and DBs, set control bits, reset past alarm, and so on before letting the program start. OB100 executes only once since it's a startup OB, there are two more startup OBs (OB101, OB102) for hot and cold restart. Once CPU executes startup OB, its ready to execute or run the PLC in normal mode to run program and read I/O for normal process.

This figure 3.2 shows the complete scan cycle of the CPU.

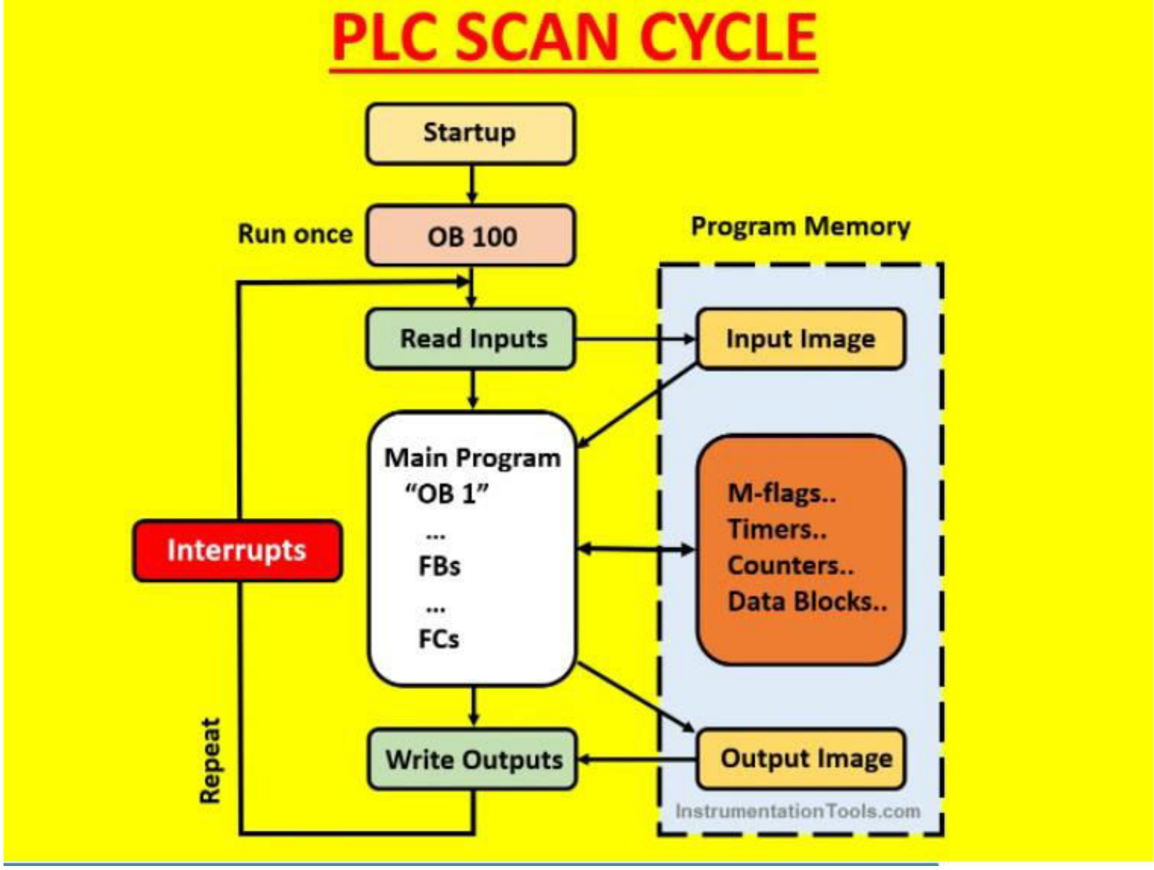


Figure 3.2 (PLC Scan Cycle)

To program the PLC, I needed to add OB1, FC, FB and so on in the block section. For the motor, I created a FB112 with the name of "Interlock Motor Control" to operate in multiple mode. I can start motor from common control room (CCR) on group mode, manual mode and it can also be started from field as local mode. I am using FBD (functional block diagram) to design any PLC logic.

As figure 3.3 shows the appearance of the motor block I created for motor control, it has multiple inputs output for controlling and group controlling. This block will operate one motor on one time call and it can operate in three different modes. Here is the list of inputs/outputs and their details below;

- EN: this input is block enable input which allow to operate this block if its connected and input is logic 1, if it is not connected then it will have no use and block will work normally. This is an automatic input generated from step7 which I cannot changed.
- Unlink: this input changes the modes of operation, if put is logic 0 it will operate at group mode and single control input (start_in, stop_in) will be unable to start the motor. Motor block can only be operated in group mode from group control inputs (L_start, L_stop), there are some more inputs that should be coming to operate in group mode. If unlink is logic 1 then it will operate in single mode.
- Start_in: when Unlink is logic 1 then start_in would be able to start motor control block in single mode. This is actually the single mode start command which only needs ready signal as logic 1 to operate the motor.
- Stop_in: when Unlink is logic 1 and motor is running in single mode then stop_in would be able to stop motor control block in single mode. This is actually the single mode stop command.
- L_start: this is group start command and it operate the motor block in group command if “unlink” input is logic 0.
- L_Stop: this is group stop command and it stops the motor block in group command if “unlink” input is logic 0.
- Remote/Local: this input also define the operating mode of the motor block, if this input is logic 0 which means it can only be started from CCR only either single or group. If this input is logic 1 which means it can now only be started from local button from the field.

- Local_start: this input will be effective on local start when remote/local input is logic 1 which means it's a local start input for local mode operation. It doesn't need any other input to operate the motor block.
- Local_stop: this input will be effective on local stop when remote/local input is logic 1 which means it's a local stop input for local mode operation. It doesn't need any other input to operate the motor block.
- Sta_admit: this block input takes input from other block to start the motor in group mode and without this input I cannot operate the motors in group mode.
- Stp_admit: this block input takes input from other block to stop the motor in group mode and without this input I cannot operate the motors in group mode.
- Ready_in: this is the control signal which I will get it from electrical cabinet which tells me that it's all good to operate this motor from electrical side.
- State_in: this is the control signal which I will get it from electrical cabinet which tells me that motor I started is still running from electrical side.
- Start_T: in motor block there is tripping on ready_in and state_in signal, if on running status any of the signal missed then PLC will trip this motor but Start_T bypass this tripping on startup for whatever the time I put and after that time it will resume its tripping.
- Rst: this is the group reset command and will reset all the alarm in the block.
- GESTP: group emergency stop will trip the motor without any delay on emergency basis.
- Drive_out: this is an output for the motor command, and if this is high which means motor is running.
- Next_ad: this is an output for this block which is going to be an input for other block to start in group mode.
- Last_ad: this is an output for this block which is going to be an input for other block to stop in group mode.
- Motor: this is a status byte output which I can used this status to show it on WinCC screen.

- Now question mark on the top of the block, here is have to put any instant DB number as I have created FB with function with data block. Here I can put any spare DB number that is not being used.

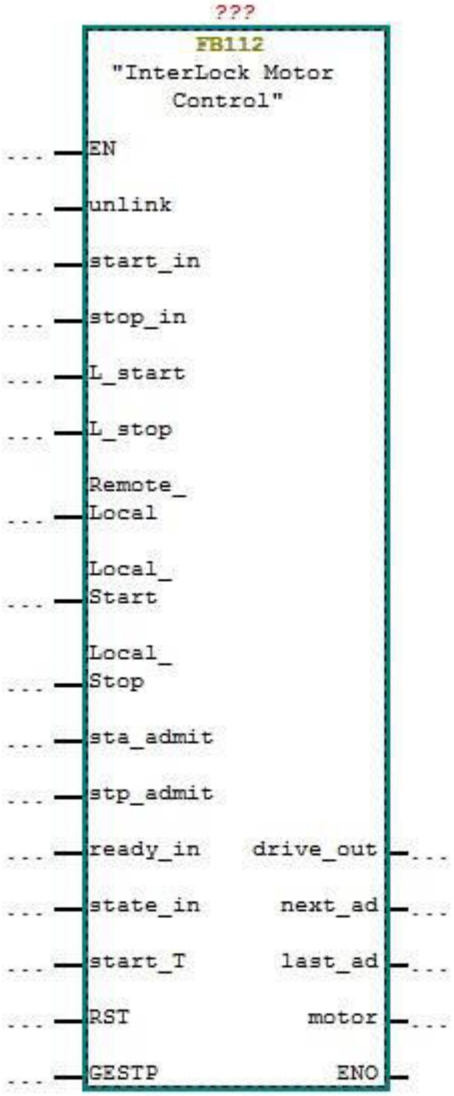


Figure 3.3 (Group Motor Block)

Next motor block I worked on is FB114 with name of “Conditional MCC”. This motor block is for single operating mode only. It will operate motor if its condition on start_ad is getting completed as shown in figure 3.4. I have used this block to operate single motor which are not in any group. As you can see in figure 3.4, it has 6 Boolean inputs and 2 outputs, the description of its inputs are as follows;

- Start_in; This is start motor command and I can use this input to operate the motors when its start_ad condition
- Stop_in: this is motor stop command
- Start_ad: this is condition which will stop motor to operate if it's not fulfilled
- Ready_in: this is the control signal that usually will get it from electrical cabinet in the field. It tells the operator that electrical circuit is ready to take command.
- State_in: this is also a control signal which indicates that motor is running from electrical side in the field.
- Start_T; this is the startup timer which bypass control signal (ready, state) tripping for the time period I choose.
- Drive_out: this is an output to operate the motor and will go directly to the electrical cabinet.
- Motor: this is status 1 byte output, it has multiple indications to show it on the WinCC screen

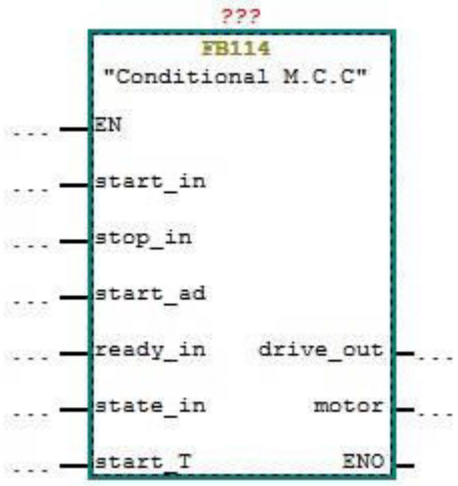


Figure 3.4 (Single Motor Block)

Next block I am going to describe is scaling block that I created in this project is FC101 as shown in figure 3.5, this is function without memory block and it doesn't need to create any DB when using it in programming. We use FB if we need to store the parameters being used in running the motor but block like scaling doesn't need any storage space.

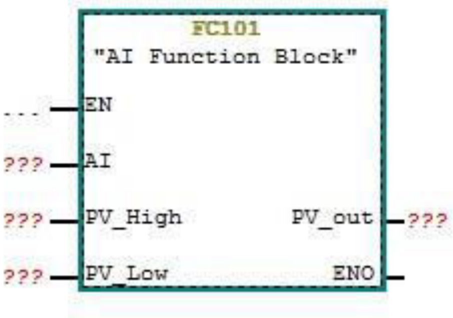


Figure 3.5 (Analog Input Scaling Block)

CPU accept analog parameters in the form of milli amps (4~20mA) which CPU converts this to (0~27648) comparatively which means if 4 mA is coming to the analog channel of PLC then in software it will show 0 decimal value and if 20mA are coming on analog channel of PLC then it will show 27648 decimal value in software. But to scale this 0~27648 value, I have to use this formula in FC101;

$$\text{Scaled value} = \left\{\frac{\text{PLC input}}{27648} \text{ x Engineering Range}\right\} + \text{PV Low}$$

Engineering range = PV_High – PV_Low
AI = PLC input

PV_out = it's a scaled value output relative to the field value.

Next block I am going to describe is scaling input/output block FC102, this can also be used to scale value from PLC decimal values to the field actual values but it can reverse the process as well. It can scale and descale both at the same time and I have this block where I am sending and receiving analog parameters like actuators, weigh feeders etc. This is the formula that I have added in FC102 to descale the value to the field.

$$\text{PLC Output} = \frac{\text{PLC SP} - \text{PV Low}}{\text{Engineering Range}} \times 27648$$

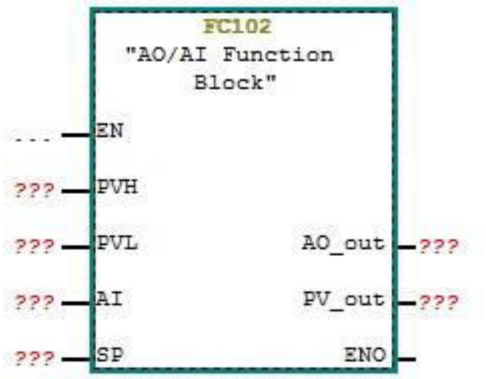


Figure 3.6 (Analog I/O Scaling Block)

Engineering range = PVH-PVL

SP = PLC SP

PLC Output = AO_out

AI = PLC input for scaling

After creating all the blocks, I created symbol table in S7 program block to symbolize all the I/O. It helps diagnose or read the program because every input can be seen with its name that I defined in symbol table. I created DB10 with the name of "Control" and I have defined memory addresses that I will use to control the motors directly from WinCC screen. Now I created FC1 for the use of communication between FCS. Apart from that I created 4 shared DBs in each FCS to send and receive data form all the FCS and will explain it in next chapter.

Created new FC2 for the programming of all the motors and here I called FB112 and FB114 for each motors. I have used FB112 for group motors and FB114 for single start motors. Created FC3 for analog parameters where I am scaling all the analog parameters and saving them in DB9 which I created to save all the analog parameters. Created FC4 for Boolean alarms and in that FC I created three alarms for single motor which is ready missed alarm, state missed alarm and electrical cabinet fault. All three alarm of each motor have been configured as tripping of any motor.

3.4 Group Motors

After calling all the FB112 interlock motor control in FC2, I needed to link them all the block to each other. First motor in the group start directly from the group start command and that command latched in the block itself. Once first motor started, it generates an output called "Next_ad" and block generate this output once he turned on the "Drive_out" and he received that control signal "State". This output "Next_ad" get connected to the input "start_ad" to the next motor of the group and next motor gets start. Once this block gets its state signal after starting the motor, it will also generate the "next_ad" for the next block and that's how it will keep start the whole group with 5 sec delay on each motor.

For stopping the group command, first motor that group is going to stop is the one which started in the end of the sequence. As soon as motor drops it drive_out and it stop receiving "state" signal from the field, this block will generate an output called "Last_ad" Which will connect to the input "stop_ad" of the previous motor block which is next to stop in the sequence. Once next motor stops, this motor will also generate last_ad which will connect to the previous motor input "stop_ad" and that's how complete group will take stop command.

3.5 Alarm

To generate the alarm in step7, I have used FC4 to create three alarms for single motor. In figure 3.7 shows only one circuit of ready missed alarm of apron conveyor of FCS01. In this one circuit, I have used three logic blocks to design one alarm of single motor. First block with "N: written on it is a negative edge block, which only pass single pulse when its input gets falling edge at input. Second block with sign "&" is an ordinary and gate with three inputs and third block is actually a flip flop which is buffering 1-bit memory of alarm and it will hold this memory bit unless reset it from it reset input.

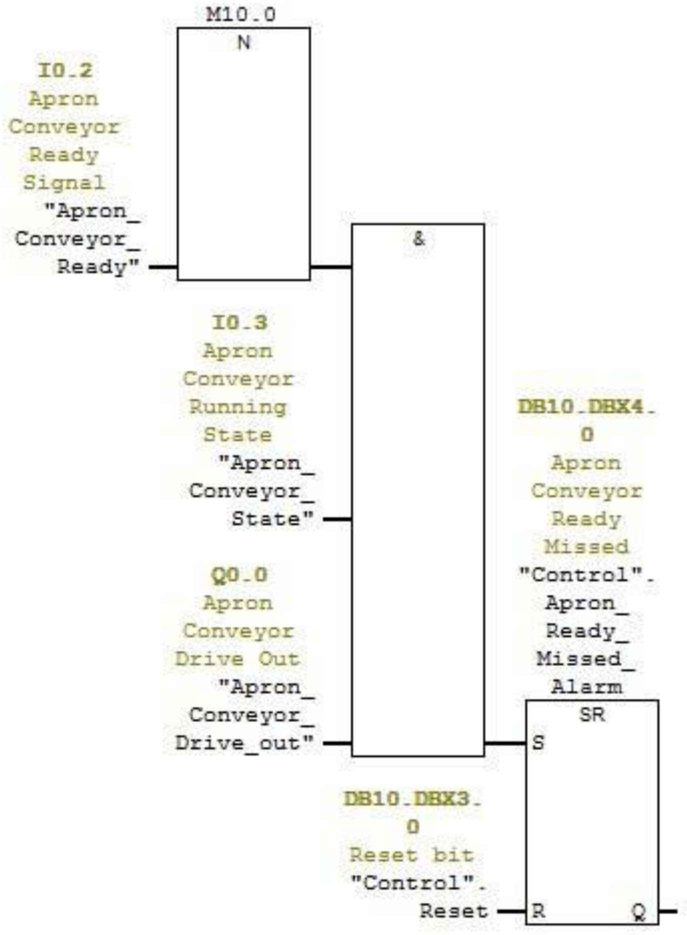

M10.0
N
I0.2
Apron Conveyor Ready Signal
"Apron_Conveyor_Ready"
&
I0.3
Apron Conveyor Running State
"Apron_Conveyor_State"
Q0.0
Apron Conveyor Drive Out
"Apron_Conveyor_Drive_out"
DB10.DBX4.0
Apron Conveyor Ready Missed
"Control".Apron_Ready_Missed_Alarm
SR
S
DB10.DBX3.0
Reset bit
"Control".Reset
R
Q

Figure 3.7 Alarm Generating Circuit

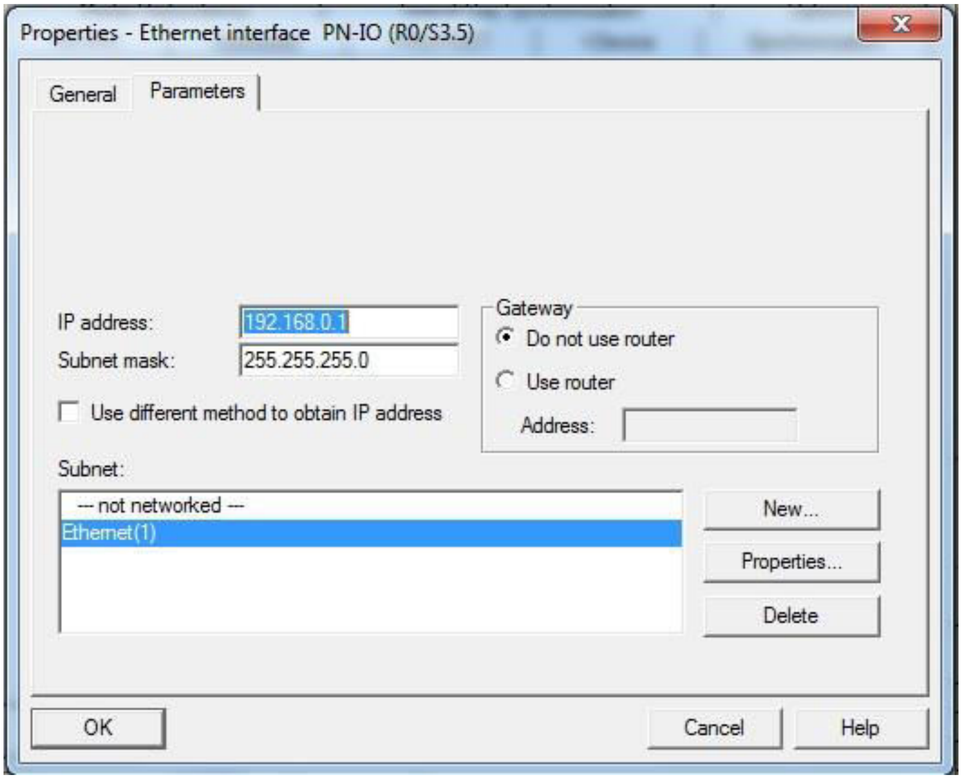
Chapter 4
PLC Communication

There are multiple ways of communicating PLC like DP-DP communication using profibus, profinet communication using Ethernet, wireless communication using wireless PLC module and so on. For long distance communication we can use fibre cable with OLM or Siemens scalance to convert Ethernet to fibre optics signal. In this project I have used Ethernet communication on simulator.

I am using three different master PLC communicating through profinet. Its needs to configure in hardware of Simatic step 7. I am going to use GET/PUT communication block to send and receive data and I just need to use these communication blocks in any one PLC for communication between two CPUs and I can send and receive four slots of 100 bytes which means I can send and receive 400 bytes with one block.

4.1 Communication Hardware Configuration

To communicate two CPU in step7 through profinet, its needs to be configured in step 7 hardware where it creates connection between different CPUs. When configuring hardware of all the FCS in hardware section, I added network as Ethernet in that as it shows in figure 4.1 and I used the same network ID and subnet mask to keep all the FCS on same network. Once I done that all the FCS are connected to each other on Ethernet. I have used network ID 192.168.0.0 and addresses are 1, 2, 3 respectively. So the IP addresses of FCS are 192.168.0.1, 192.168.0.2, 192.168.0.3. Figure 4.1 shows the IP address configuration of FCS01.



Properties - Ethernet interface PN-IO (R0/S3.5)

IP address: 192.168.0.1

Subnet mask: 255.255.255.0

□ Use different method to obtain IP address

Gateway

● Do not use router

○ Use router

Address:

Subnet:

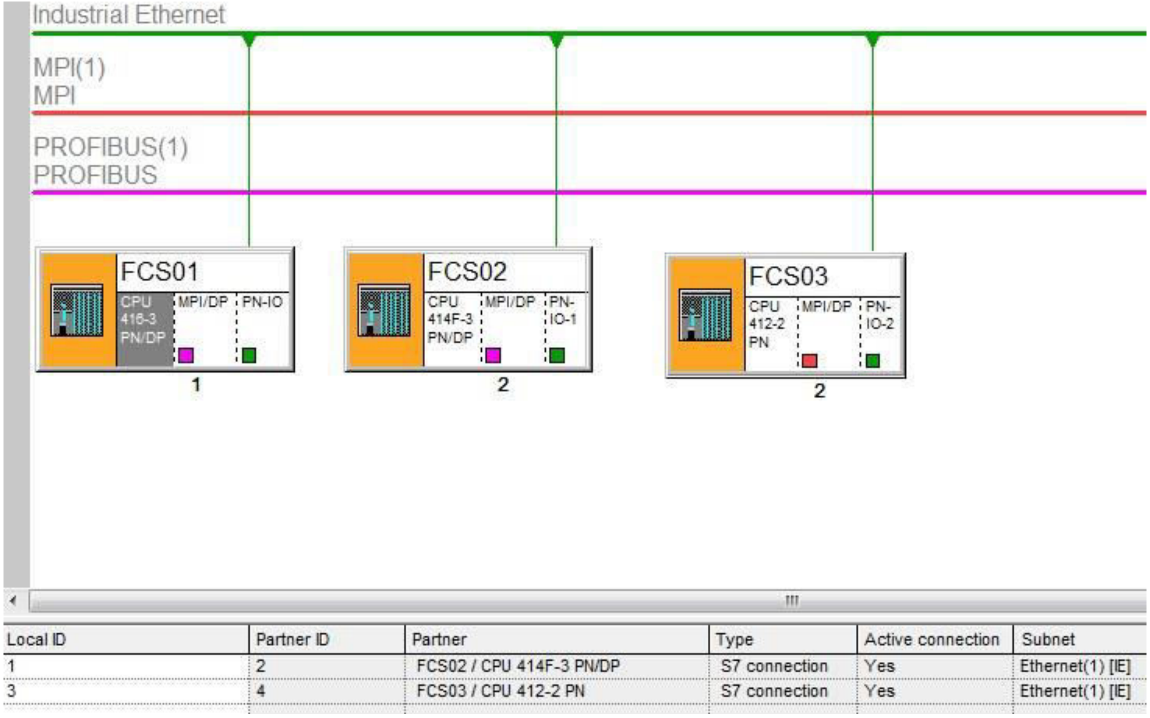
--- not networked ---

Ethernet(1)

Figure 4.1 (CPU Ethernet Configuration

Next step is to created connection in the configure network section to send or receive data between the connections and after opening the configure network section it can be seen that all FCS are connected to each other through Ethernet in green color as shown in figure 4.2. To activate the communication between FCS, I needed to create connection between the FCS and to create connections between FCS, clicked on FCS01 CPU and it will show the table on the bottom of the page. Right clicked on 1st row and select insert new connection. Once clicked on it, a new window will pop up will show the list of FCS that I can select any FCS to create the connection. I will have to give connection ID as well and PLC will communicate through this connection ID in programming. If there is no connection ID I would be unable to communicate even though all the FCS are connected to each other through Ethernet. So I created two connections for FCS01, one is between FCS01 & FCS02 and second connection is between FCS01 & FCS03. Since one connection of FCS02 & FCS03 with FCS01 have been made and I can see that as well when I click on FCS02 or FCS03. Now I need to create new connection between FCS02

& FCS03 by clicking on FCS02 and create connection between FCS02 & FCS03. Now all the connections have been made and can see in figure 3.2.



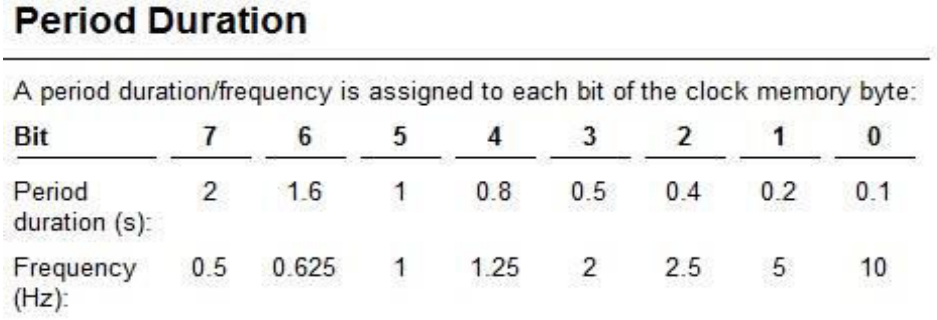
Local ID	Partner ID	Partner	Type	Active connection	Subnet
1	2	FCS02 / CPU 414F-3 PN/DP	S7 connection	Yes	Ethernet(1) [IE]
3	4	FCS03 / CPU 412-2 PN	S7 connection	Yes	Ethernet(1) [IE]

Figure 4.2 (CPU Connection Configuration)

When I created the connection between FCS01 & FCS02, this is first connection for FCS01 & FCS02 and FCS01 & FCS02 are ready to communicate. When I created the connection between FCS01 & FCS03, this is the second connection for FCS01 but first connection of FCS03 and FCS01 & FCS03 are ready to communicate. Only FCS02 & FCS03 are not communicating since there is no connection between them so I needed to create connection between FCS02 & FCS03 and after creating the connection between them all FCS have two connections with each FCS and all FCS are communicating each other.

4.2 Communication Software Configuration

After doing all hardware configuration, I need to configure its software and need to use PUT/GET block to send and receive data. PUT/GET block use connection ID to send and receive data on different band width. Here I am using CPU clock to send and receive data, CPU clock use memory block and can be defined addresses in properties of CPU hardware. I am using CPU clock memory address byte 0. Every bit of this byte has different time period and I can use any bit to transfer data as seen in the figure 4.3.



Period Duration

A period duration/frequency is assigned to each bit of the clock memory byte:

Bit	7	6	5	4	3	2	1	0
Period duration (s):	2	1.6	1	0.8	0.5	0.4	0.2	0.1
Frequency (Hz):	0.5	0.625	1	1.25	2	2.5	5	10

Figure 4.3 (Frequency Table For Sending Data)

I have created FC1 in each FCS for the communication, all the data being send or receive is getting transfer to its original addresses in FC1. Then I created 4 shared DB in each FCS to put addresses on PUT/GET block that means whatever data I will put in these DBs will be send or receive respectively from or to their addresses.

In this project, I am communicating FCS01 with FCS02 and FCS02 with FCS03 with respect to this project even though I have created connection between FCS01 & FCS03 but there is no application in this project of communicating FCS01 & FCS03. Now for the communication of FCS01/FCS02, I have used GET/PUT command in FCS02 and for the communication of FCS02/FCS03, I have used GET/PUT command in FCS03. Figure 4.7 shows the GET block that is being used in OB1 of FCS02 for reading data from FCS01. This GET command is reading 20 bytes in 0.1 sec

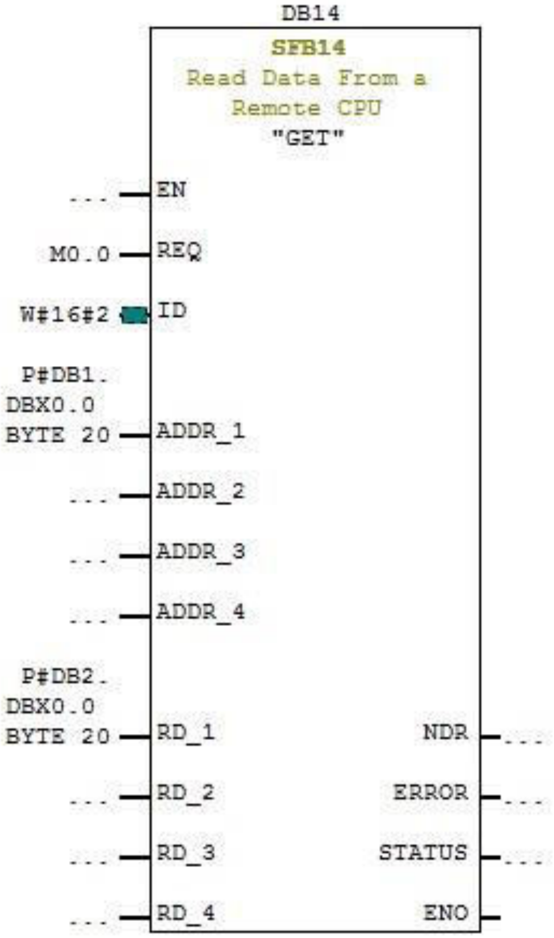


Figure 4.4 (GET Block for Data Receiving)

On input REQ, is a CPU clock memory bit, its giving pulses on 0.1 sec/ pulse or its giving 10 pulses in 1 sec, which also means that this block is reading data 10 times from FCS01 in 1 sec. Here input ID is a connection ID that needs to be mention from which connection it needs to read data. Connection ID can be find out by right click on this input and choose Connection in that and it will show all the connections ID and their name on it and we can choose it from there. ADDR_1 is the source address of FCS01 from where data is going to read so here I have to assign this address as a pointer and I am reading 20bytes from this address. RD_1 is the destination address where collected data is going to write in FCS02. Since I have read 20 bytes from FCS01 which means I need to write 20 bytes in FCS02 or else CPU will create an error and will go on stop mode.

Second block to write data from FCS02 to FCS01 is PUT and figure 4.5 show the image

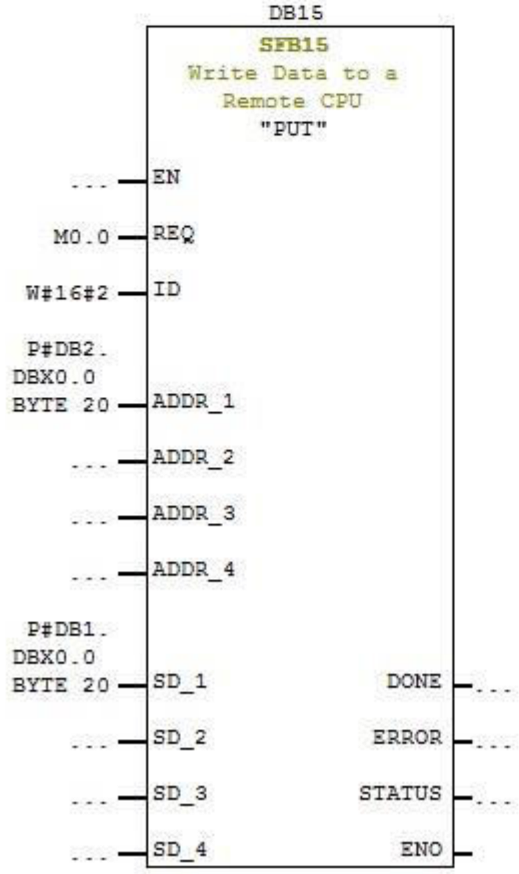


Figure 4.5 (PUT Block for Sending Data)

of PUT block that I used in OB1 of FCS02. In this block, ADDR_1 is the detination address of FCS01 on which data is going to write and SD_1 is the address from where data is going to send. I am sending 20 bytes data again from with the help of pointer address.

4.3 Data Exchange

For the communication between FCS01 and FCS02, I have shared 20 bytes on both sides of PLC which FCS02 is reading 20 bytes from FCS01 in DB2 and saving it in DB10 and sending 20 bytes from DB1 to DB2 of FCS01 where FCS01 CPU is moving DB2 addresses to DB10. In Crusher area, limestone reclaimer and its feeding belt is on FCS01 area but it should get command from FCS02 because it is the part of Raw mill process which is getting operated by FCS02 so here logically reclaimer and its belt is getting operating from FCS01 but its start stop command is coming from FCS02 through Ethernet communication and FCS01 sending all the control signals (Ready, State, Alarm) to FCS02. On FCS02, I am copying 4 Boolean bits to the 20 bytes' addresses which are being send to FCS01 and copying 6 Boolean bits from the 20 bytes' addresses that are being received from FCS01. I can increase this data communication to 400 bytes if its needed in the project.

Same as in FCS01 and FCS02, I am doing same practice for FCS02 and FCS03, here I am also sending and receiving 20 bytes from FCS02 and FCS03. There are two big HT fans (VRM BH Fan & ID fan) in which VRM BH Fan operates from FCS02 while ID Fan operates from FCS03 now the process requirement is if I want to operate ID Fan then VRM BH fan should be in running state or else ID Fan won't start and if both fans are running and VRM BH fan trips on any alarm then ID fan will also trip due to BH fan failure. So here I am reading BH Fan running state from FCS02 and using it on ID Fan logic block in FCS03. Using this technique, I can send or receive 400 bytes with 1 block of GET/PUT and I can increase the number of blocks as well.

Chapter 5
WinCC

5.1 Introduction

To design virtual HMI for the project, I have used Simatic WinCC. This is a Siemens software which can directly link to the Siemens PLCs and can show online data from the PLC directly to the HMI. To create a screen, I created the project with the name “MRes” in which I have created three main screens for three FCS i.e. FCS01.pdl, FCS02.pdl, FCS03.pdl. One main screen will be used which is actually a startup screen with name of “Main”. This is the first screen which WinCC will open when it will run its project. This main screen will have multiple tabs to change between different FCS and I have linked two extra screen which will be used for trends and alarm. This main screen will have tabs for all the screen and I can switch the screen while running the project.

5.2 Screen Configuration

After creating the project, I created the startup screen named with “Main” and in this screen I added two picture windows in which I can call any screen in it. Created one more screen with the name of “Header” in which I have created all the buttons to switch between the screens with the project headings and clock on it. On every button on header screen, I have done scripting on its button press properties, and its working with respect to the scripting.

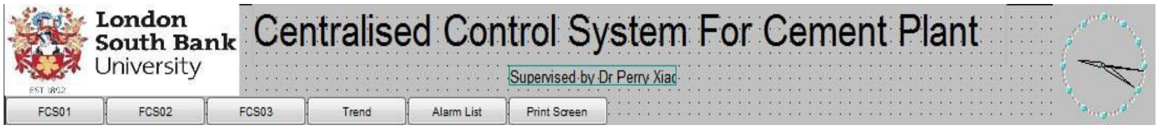
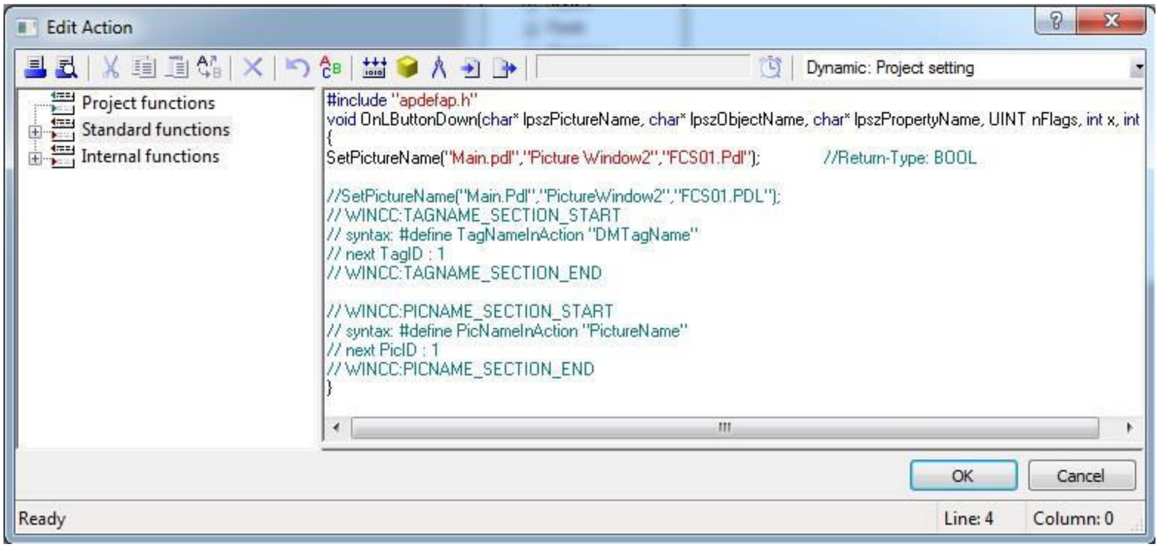

London South Bank University
EST 1892
Centralised Control System For Cement Plant
Supervised by Dr Perry Xiac
FCS01
FCS02
FCS03
Trend
Alarm List
Print Screen

Figure 5.1 (Header Screen)

This is the header screen that I am calling in picture window 1 of startup screen. As it can be seen in figure 5.1 that this screen has many buttons like FCS01, FCS02, FCS03 and so on and it will switch screen in picture window 2 of Main screen. The scripting I have done on button in header screen is based on C language and it has command set that I have used to define screen name on picture window2.



Dynamic: Project setting

```
#include "apdefap.h"
void OnLButtonDown(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName, UINT nFlags, int x, int
{
SetPictureName("Main.pdl","Picture Window2","FCS01.Pdl");          //Return-Type: BOOL

//SetPictureName("Main.Pdl","PictureWindow2","FCS01.PDL");
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction "PictureName"
// next PicID : 1
// WINCC:PICNAME_SECTION_END
}
```

Figure 5.2 (Screen Switching Configuration)

The function I am using here is "SetPictureName" and its actually defining picture name to picture window2 in main screen and that's how it switches its screen on every tab that used in header screen. I have used same technique for the trend window as well in which there are three trend screens that I need to switch and it will switch between trend_FCS01, trend_FCS02, trend_FCS03 as it can be seen in figure 5.3.

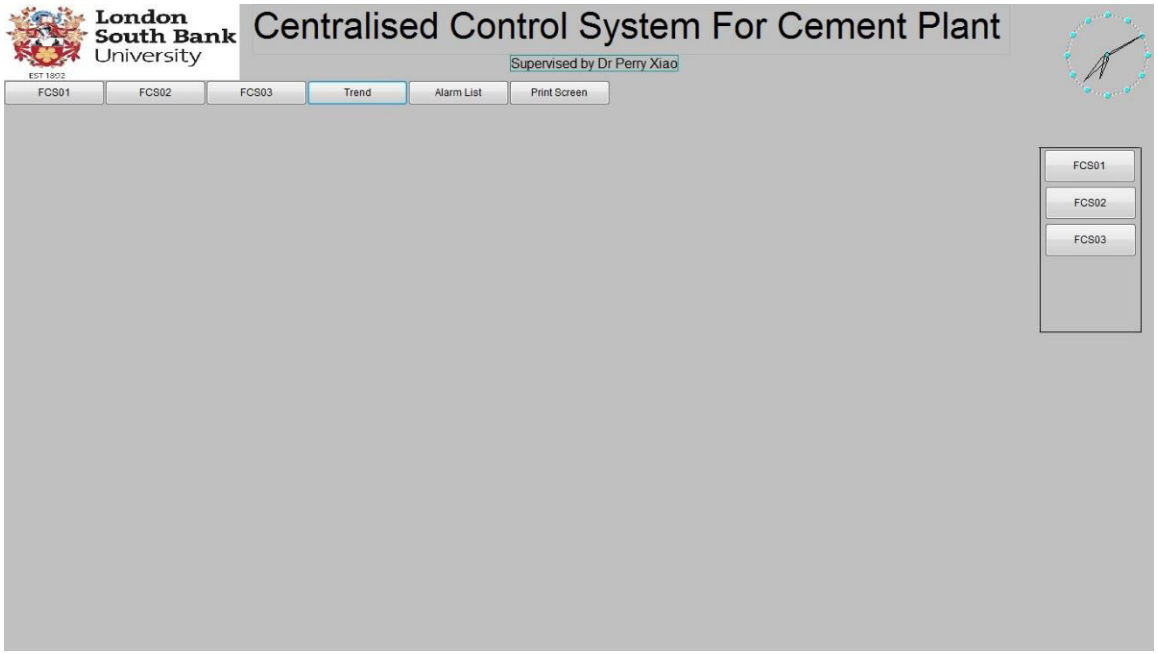
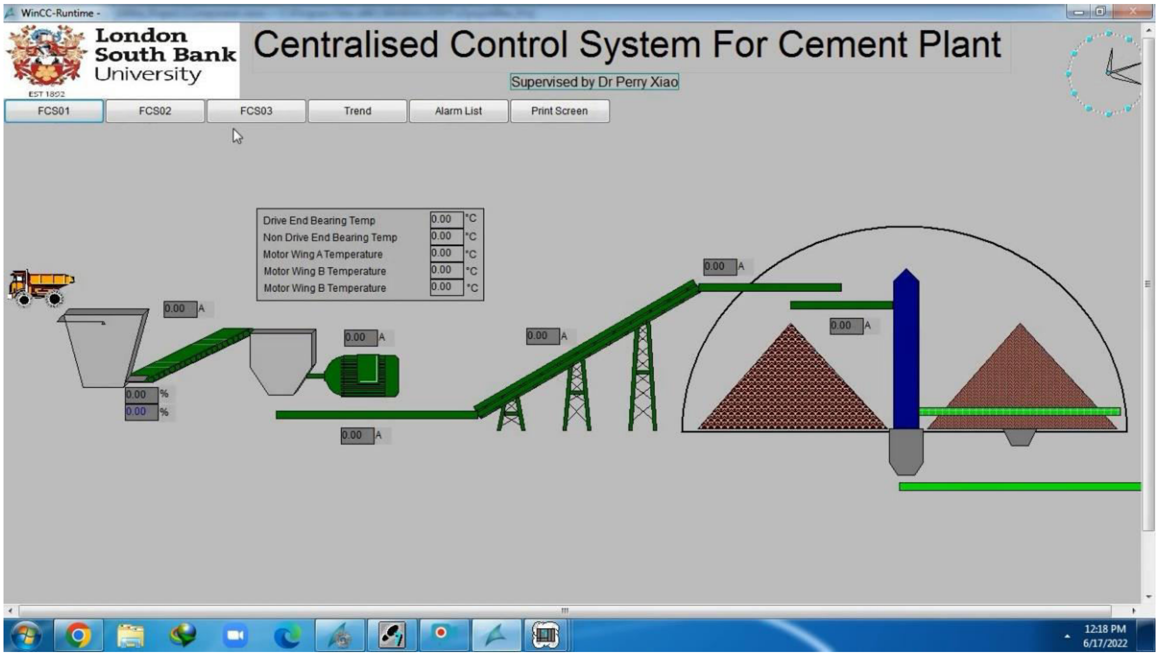


Figure 5.3 (WinCC Main Screen Appearance)

5.3 FCS01 Screen

After working on main appearance of the project, I started working on FCS01 screen and multiple motors on the screen and tried to make a similar process cycle that cement crusher system uses it. I have already done step7 PLC programming of this system. I used WinCC library to select motors, conveyor belts or hoppers even I added one loading dumper to show that all raw limestone is dropping off from this side. After arranging all the symbols in a pattern, FCS01 screens appears in running state as shown in figure 5.4.



Drive End Bearing Temp	0.00	°C
Non Drive End Bearing Temp	0.00	°C
Motor Wing A Temperature	0.00	°C
Motor Wing B Temperature	0.00	°C
Motor Wing B Temperature	0.00	°C

Figure 5.4 (FCS01 Running Screen)

Next step I took is working on control screen of this crusher process. On this control screen, I added group control buttons (group start, group stop) and added individual control buttons to operate it individually. Added some indications for ready state alarm signals as well as added some reset and emergency stop button to give command to the PLC. After arranging all the button in groups, it appears as it shows in figure 5.5 and since it has been completed so I started making some tag in Tag management section. Tag is the component of WinCC on which WinCC perform its task, tag is actually giving name of PLC addresses because WinCC only recognized tags on which I can configure properties so I created all the tags of memory and I/O addresses of PLC and there were nearly 103 tags in FCS01 that I had to create.

After creating all the tags, I linked all the indication tags that where changing pictures when any tag changes and then added all the tags on command button in WinCC scripting based on C language. There are build-in functions that I have used to script the command button on control screen. Function I have used for scripting is `SetTagBit(" ",)` where

I can put any tag that I need to change and after comma I can put 0 or 1 as a required output of that tag example SetTagBit("Apron_Start",1)

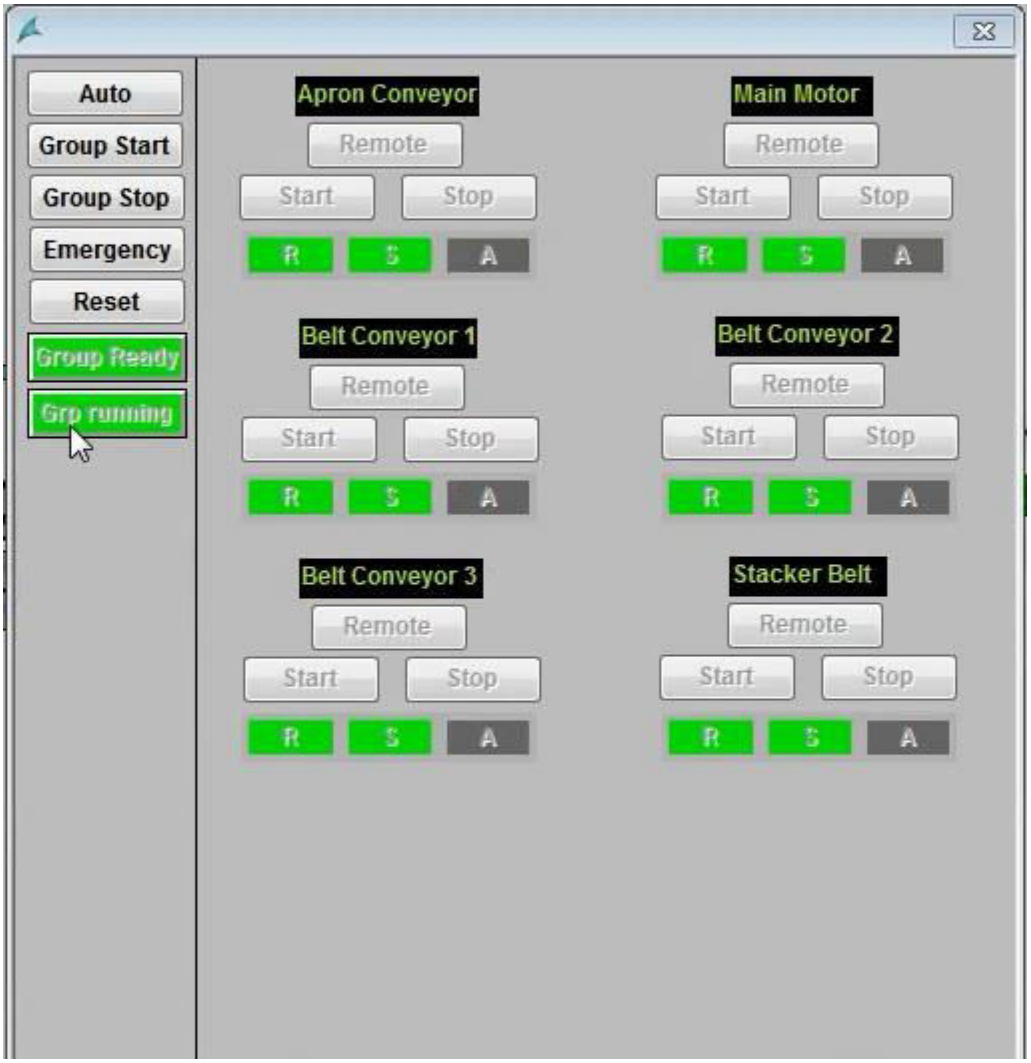


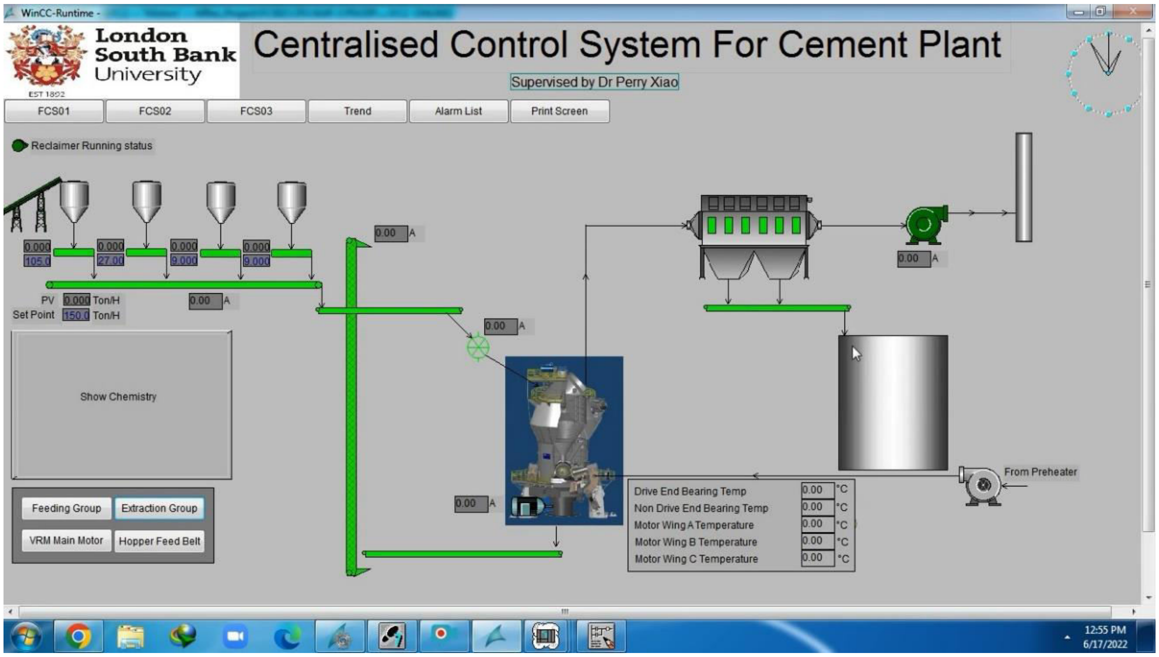
Figure 5.5 (Control Faceplate)

This function will change tag "Apron_Start" value from 0 to 1 when button pressed on runtime. I have used this same function with another function to make a toggle button on the screen example SetTagBit("Apron_Local",!(GetTagBit("Apron_Local"))). In this function GetTagBit is a function which is actually reading the value of tag "Apron_Local" and inverting it with the help of exclamation mark and changing its value with the function SetTagBit so after this scripting this button is working as a toggle button and its changing value from 0 to 1 and 1 to 0 every time this button pressed. This is how I done scripting on all the button of FCS01 control screen.

After this work, I started working on analog parameters and added 13 I/O field on FCS01 screen for motor amperes, control temperatures, and set point & PV for motors. Since I created tags already so I added tags directly to the properties of I/O fields and that's how FCS01 has reached to it ending point.

5.4 FCS02 Screen

FCS02 is actually a Raw Mill Grinding where all the raw material gets grind and gets store in silo to get ready for the next cement process. Here I used the same WinCC library to draw motors, fans etc. Raw mill picture itself was not in the library so I had to get it from other resources. After arranging all the components on FCS02 in process cycle, I created 2 control screen (faceplate) because it has two control groups and both of them working independently. First group I have made is a mill feeding group and second group is material extraction group from the mill. I created 2 more control screen for individual motors of VRM main motor and VRM system fan. All the control screens I have created is similar to what I have created in FCS01 so after adding all the buttons in all four screens of FCS02, I again used same function for the scripting on buttons of control screens. Again after completing all the scripting, I added 20 I/O fields for the analog parameters and linked tags directly to them. Once it's done, my FCS02 screen appearance with running state is shown in figure 5.6



0.000	0.000	0.000	0.000
105.0	27.00	9.000	9.000

PV 0.000 Ton/H
Set Point 150.0 Ton/H

0.00 A
0.00 A
0.00 A
0.00 A
0.00 A

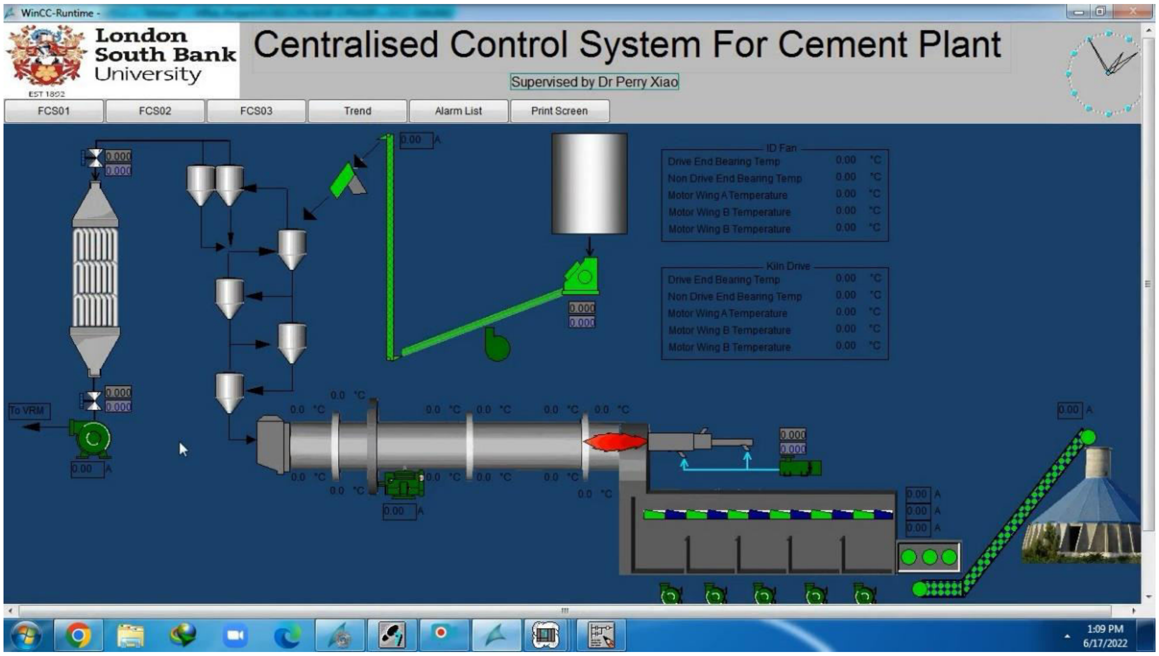
Drive End Bearing Temp	0.00	°C
Non Drive End Bearing Temp	0.00	°C
Motor Wing A Temperature	0.00	°C
Motor Wing B Temperature	0.00	°C
Motor Wing C Temperature	0.00	°C

Figure 5.6 (FCS02 Running Screen)

For the chemistry of raw material, I have added percentage system in VRM. I need to put percentages of different material on one time after that whatever set point I will give it will automatically divide into different weigh feeders of VRM and will appear on the screen as well. Now because this chemistry is very crucial data so for its safety I placed a rectangle screen to hide the chemistry and I created an extra tag "Hide/appear" and put mouse click action on it that if mouse click on it then it turns that tag from 0 to 1 and put one more property on its display that if that tag is 1 then this screen will disappear and if this tag is 0 then it will appear in front of chemistry screen.

5.5 FCS03 Screen

FCS03 is actually a Burning furnace system where all the grinded raw material gets melted and change its shape from raw material to small balls of cement called clinker. Here I used the same WinCC library to draw motors, fans etc. Kiln itself was not in the library so I had to get it from other resources. After arranging all the components on FCS03 in process cycle, I created 2 control screen because it has two control groups and both of them working independently. First group I have made is a kiln feeding group and second group is cooling system group. I created 3 more control screen for individual motors of ID fan, Kiln DC Drive and Main burner. All the control screens I have created is similar to what I have created in FCS01 & FCS2 so after adding all the buttons in all five screens of FCS03, I again used same function for the scripting on buttons of control screens. After completing all the scripting, I added 33 I/O fields for the analog parameters and linked tags directly to them. Once it's done, my FCS03 screen appearance with running state is shown in figure 5.7


WinCC-Runtime -
London South Bank University
EST 1892
Centralised Control System For Cement Plant
Supervised by Dr Perry Xiao
FCS01
FCS02
FCS03
Trend
Alarm List
Print Screen
To VRM
1:09 PM
6/17/2022

ID Fan

Drive End Bearing Temp	0.00	°C
Non Drive End Bearing Temp	0.00	°C
Motor Wing A Temperature	0.00	°C
Motor Wing B Temperature	0.00	°C
Motor Wing B Temperature	0.00	°C

Kiln Drive

Drive End Bearing Temp	0.00	°C
Non Drive End Bearing Temp	0.00	°C
Motor Wing A Temperature	0.00	°C
Motor Wing B Temperature	0.00	°C
Motor Wing B Temperature	0.00	°C

Figure 5.7 (FCS03 Running Screenshot)

5.6 SYSTEM VALIDATON

This system is actually an industrial system which requires hardware and software to run the system. There are certain requirements which this system fulfil it with complete protection. Cement plant process is based on wide area with multiple HT motors and instruments with large amount of parameters and control signals. With this system, operator is able to see and control all motors, instrument and its parameters in one screen of the computer. When there is human interaction there will be an error which can leads to major accidents that's why this system checks motors & instruments safety, plant safety and it helps prevent the accident.

The objective of this project is to fulfill all operator needs for more effective and efficient plant operation which allow significant improvement. It's always been a human factor which has the highest potential for increasing plant unplanned shutdown and slowdown but this system provides smooth operation in day-to-day business and enables the operator to manage unexpected challenges. It is design to make the operator life much easier and after running this system they can't even imagine to operate a cement plant without this system. This system helps the operator stay capable and perform with the cool head even in critical situations.

5.7 FUTURE ENHANCEMENT

This project has very scope in the future, modern technologies has overtaken the standard PLC control systems. Siemens latest technology for control system is PCS7, this system not only help the operator in smooth plant operation but this also helps an engineer to design the system easily. This is more efficient and effective control system which we can use in this project as future enhancement. Instead of Ethernet and fibre cable, we can use Siemens latest wireless module which will enhance its communication system. We can use latest CPUs to enhance the processing speed of the control system. If there is more requirement of I/O then we can extend our I/O by adding more I/O cards.

Chapter 6
Tag & Alarm Logging

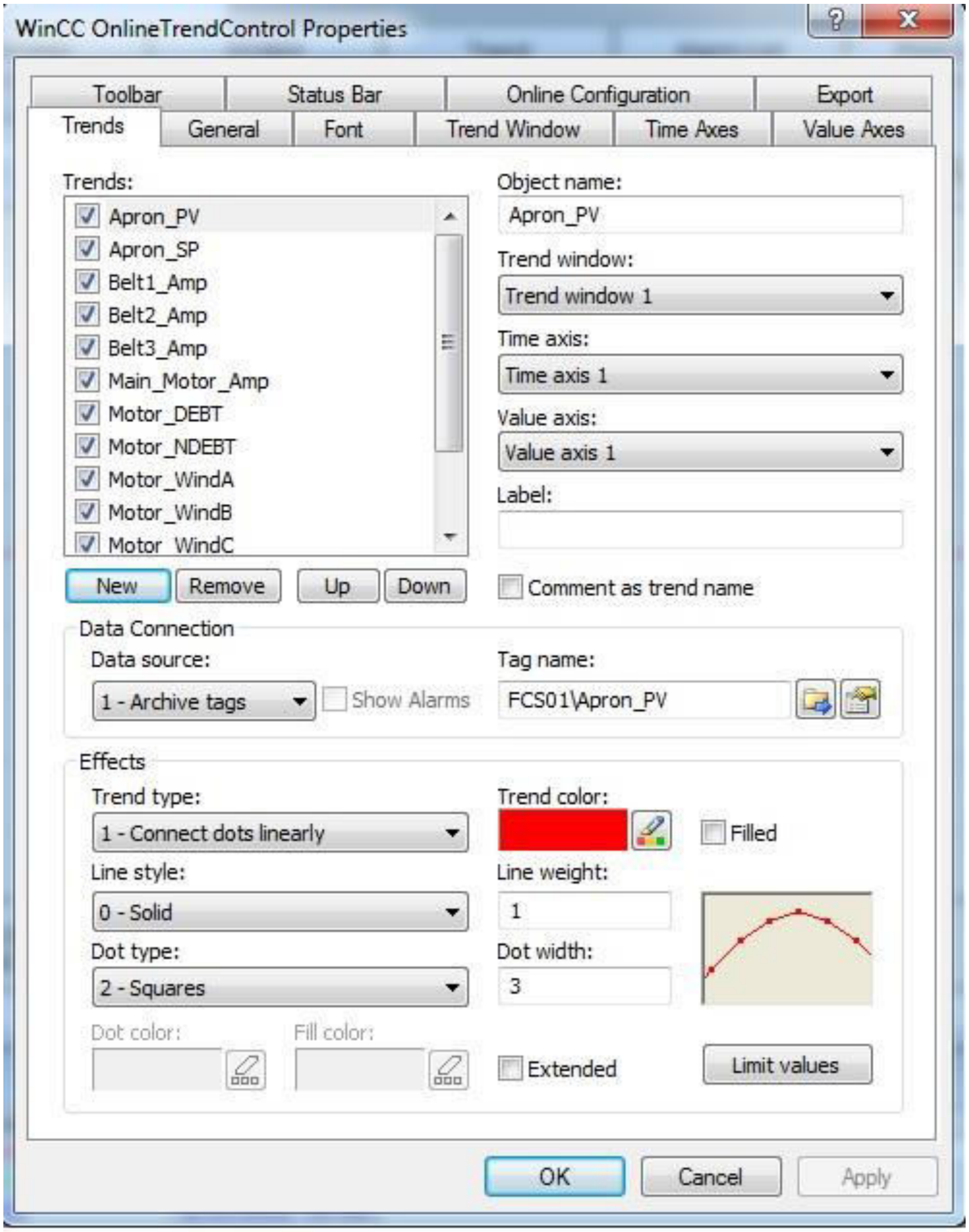
6.1 Tag logging

WinCC has the ability to log the analog parameters and their runtime generated alarms in WinCC. It uses tag logging and alarm logging section in it to log the parameters. To log the tag of analog parameters I created another screen named "Trend.pdl" and added the scripting on header screen to call "trend.pdl" in "main" screen of picture window2. In "Trend.pdl", I created again two picture windows in which picture window 2 has the button to switch the screens in picture window 1. For picture window 2, I created another three screens for the trend windows of different FCS named "Trend_FCS01, Trend_FCS02, Trend_FCS03". These screens will be called on picture windows 1 from the tabs of picture window 2.

Now in "Trend_FCS01" which is the trend picture of FCS01, added WinCC online trend control which helps WinCC to show graphical representation of logged analog parameters. This online trend control accesses the past logged parameters and display it in graphical representation on demand. I added a ruler control as well with online trend control and needed to link the ruler control to the online trend control by selecting source control1 in configuration properties of ruler control. Now trend graphical representation and physical work is complete, next step would be adding tags in tag logging.

Adding tags in tag logging is the basic practice to see trend in the trend window. For doing this, I created three different archiving group for different FCS, and every parameter acquisition time is 500mS. After adding all parameters in tag logging in their respective archiving group, I just needed to add all the archive tags in online trend control which will display all the archived sample of analog parameters in a graphical representation. Here I have configured time period for all segments is one month and maximum size of all segments is 1 giga byte which means it will store analog parameters for one month with maximum data storage of HDD is 1 GB.

Archived tags can be added by going in the configuration dialog of trend window, add multiple trend with respect to the analog parameters and give tag address in source window of all the trends, see figure 6.1



Trends:

☑ Apron_PV
☑ Apron_SP
☑ Belt1_Amp
☑ Belt2_Amp
☑ Belt3_Amp
☑ Main_Motor_Amp
☑ Motor_DEBT
☑ Motor_NDEBT
☑ Motor_WindA
☑ Motor_WindB
☑ Motor_WindC

Object name: Apron_PV

Trend window: Trend window 1

Time axis: Time axis 1

Value axis: Value axis 1

Label:

□ Comment as trend name

Data Connection

Data source: 1 - Archive tags □ Show Alarms

Tag name: FCS01\Apron_PV

Effects

Trend type: 1 - Connect dots linearly

Trend color: 🟥 □ Filled

Line style: 0 - Solid

Line weight: 1

Dot type: 2 - Squares

Dot width: 3

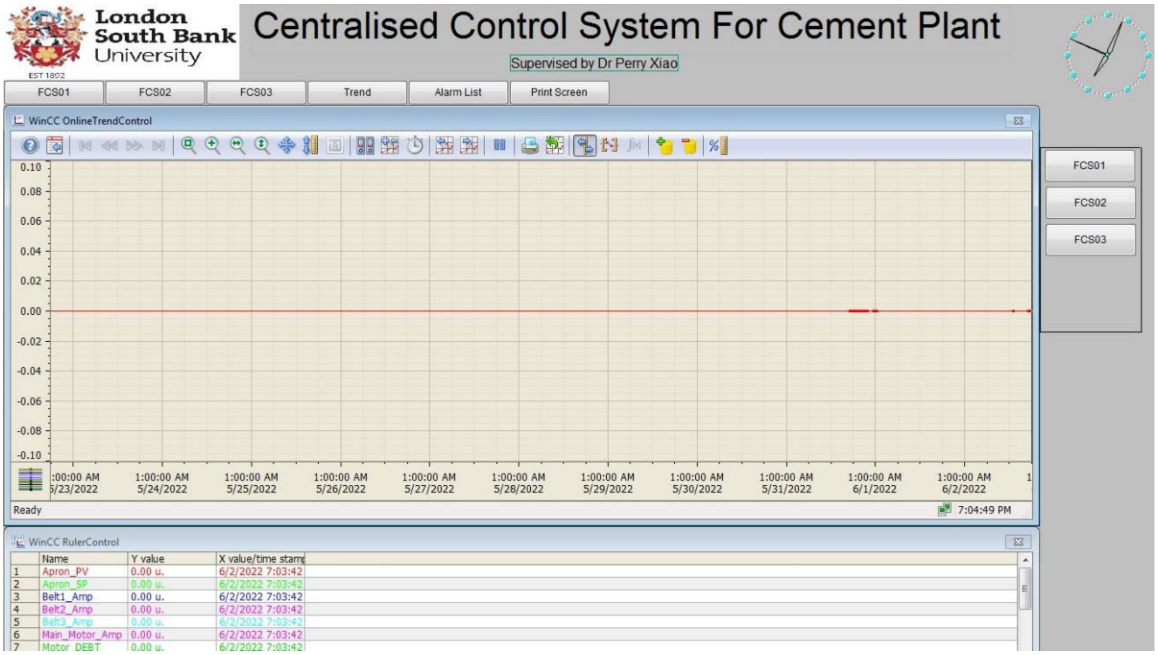
Dot color:

Fill color:

□ Extended

Figure 6.1 (Trend Configuration Window)

After adding all the trend in trend configuration dialog, I am able to see all the tags in the online trend control and ruler control is showing current value wherever I take my ruler on the trend, final appearance of trend window is shown in figure 6.2

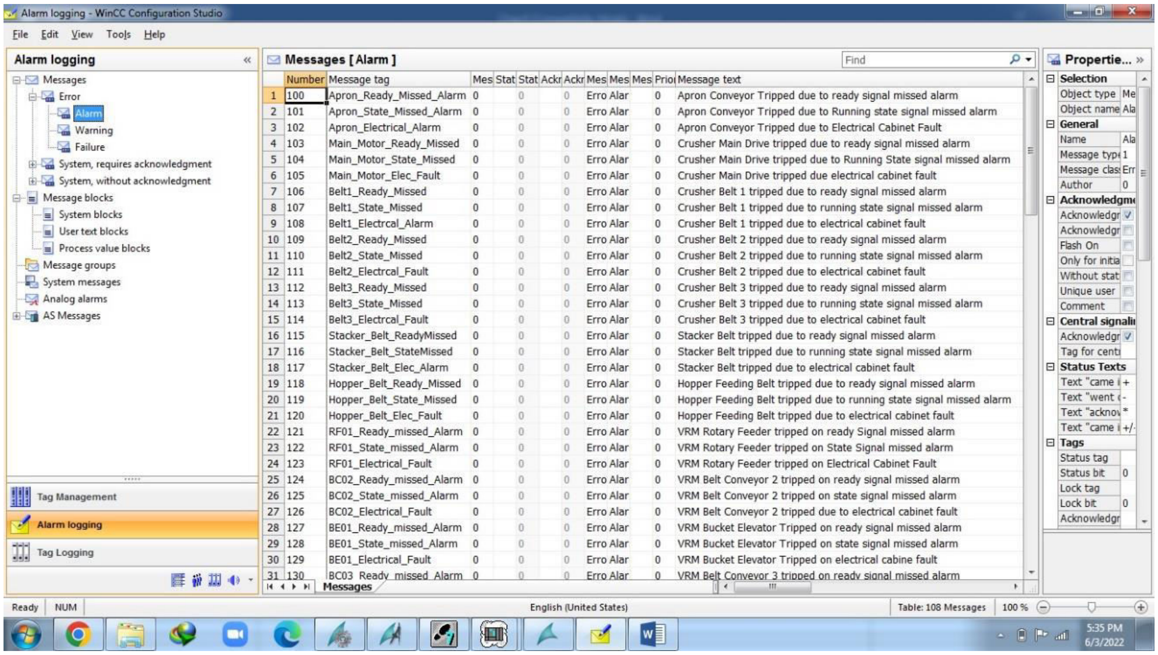


	Name	Y value	X value/time stamp
1	Apron_PV	0.00 u.	6/2/2022 7:03:42
2	Apron_SP	0.00 u.	6/2/2022 7:03:42
3	Belt1_Amp	0.00 u.	6/2/2022 7:03:42
4	Belt2_Amp	0.00 u.	6/2/2022 7:03:42
5	Belt3_Amp	0.00 u.	6/2/2022 7:03:42
6	Main_Motor_Amp	0.00 u.	6/2/2022 7:03:42
7	Motor_DEBT	0.00 u.	6/2/2022 7:03:42

Figure 6.2 (WinCC Trend Picture)

6.2 Alarm logging

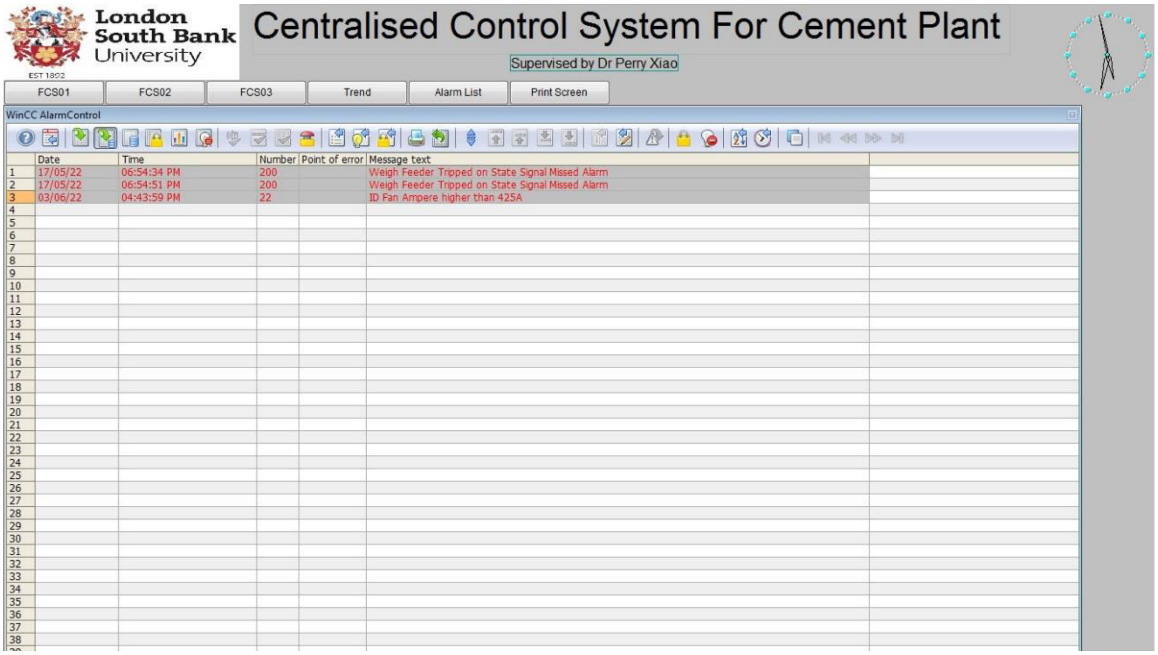
Alarm logging also works on same principle as tag logging, WinCC logged alarm up to 1000 alarms in list. WinCC uses alarm logging section in it to log the alarm in runtime. I created another screen named “Alarm_List.pdl” in graphic designer for the alarm list and in this picture I used alarm control from the WinCC library. After giving it a screen size I called this screen by adding scripting on button named “Alarm List”. I am using same technique here to call the screen in picture window2 of screen “Main.pdl”. After doing this, Alarm_List.pdl was appearing in main screen when press button “Alarm List”.



	Number	Message tag	Mes	Stat	Stat	Ackr	Ackr	Mes	Mes	Mes	Prio	Message text
1	100	Apron_Ready_Missed_Alarm	0		0		0	Erro	Alar		0	Apron Conveyor Tripped due to ready signal missed alarm
2	101	Apron_State_Missed_Alarm	0		0		0	Erro	Alar		0	Apron Conveyor Tripped due to Running state signal missed alarm
3	102	Apron_Electrical_Alarm	0		0		0	Erro	Alar		0	Apron Conveyor Tripped due to Electrical Cabinet Fault
4	103	Main_Motor_Ready_Missed	0		0		0	Erro	Alar		0	Crusher Main Drive tripped due to ready signal missed alarm
5	104	Main_Motor_State_Missed	0		0		0	Erro	Alar		0	Crusher Main Drive tripped due to Running State signal missed alarm
6	105	Main_Motor_Elec_Fault	0		0		0	Erro	Alar		0	Crusher Main Drive tripped due electrical cabinet fault
7	106	Belt1_Ready_Missed	0		0		0	Erro	Alar		0	Crusher Belt 1 tripped due to ready signal missed alarm
8	107	Belt1_State_Missed	0		0		0	Erro	Alar		0	Crusher Belt 1 tripped due to running state signal missed alarm
9	108	Belt1_Electrcal_Alarm	0		0		0	Erro	Alar		0	Crusher Belt 1 tripped due to electrical cabinet fault
10	109	Belt2_Ready_Missed	0		0		0	Erro	Alar		0	Crusher Belt 2 tripped due to ready signal missed alarm
11	110	Belt2_State_Missed	0		0		0	Erro	Alar		0	Crusher Belt 2 tripped due to running state signal missed alarm
12	111	Belt2_Electrcal_Fault	0		0		0	Erro	Alar		0	Crusher Belt 2 tripped due to electrical cabinet fault
13	112	Belt3_Ready_Missed	0		0		0	Erro	Alar		0	Crusher Belt 3 tripped due to ready signal missed alarm
14	113	Belt3_State_Missed	0		0		0	Erro	Alar		0	Crusher Belt 3 tripped due to running state signal missed alarm
15	114	Belt3_Electrcal_Fault	0		0		0	Erro	Alar		0	Crusher Belt 3 tripped due to electrical cabinet fault
16	115	Stacker_Belt_ReadyMissed	0		0		0	Erro	Alar		0	Stacker Belt tripped due to ready signal missed alarm
17	116	Stacker_Belt_StateMissed	0		0		0	Erro	Alar		0	Stacker Belt tripped due to running state signal missed alarm
18	117	Stacker_Belt_Elec_Alarm	0		0		0	Erro	Alar		0	Stacker Belt tripped due to electrical cabinet fault
19	118	Hopper_Belt_Ready_Missed	0		0		0	Erro	Alar		0	Hopper Feeding Belt tripped due to ready signal missed alarm
20	119	Hopper_Belt_State_Missed	0		0		0	Erro	Alar		0	Hopper Feeding Belt tripped due to running state signal missed alarm
21	120	Hopper_Belt_Elec_Fault	0		0		0	Erro	Alar		0	Hopper Feeding Belt tripped due to electrical cabinet fault
22	121	RF01_Ready_missed_Alarm	0		0		0	Erro	Alar		0	VRM Rotary Feeder tripped on ready Signal missed alarm
23	122	RF01_State_missed_Alarm	0		0		0	Erro	Alar		0	VRM Rotary Feeder tripped on State Signal missed alarm
24	123	RF01_Electrical_Fault	0		0		0	Erro	Alar		0	VRM Rotary Feeder tripped on Electrical Cabinet Fault
25	124	BC02_Ready_missed_Alarm	0		0		0	Erro	Alar		0	VRM Belt Conveyor 2 tripped on ready signal missed alarm
26	125	BC02_State_missed_Alarm	0		0		0	Erro	Alar		0	VRM Belt Conveyor 2 tripped on state signal missed alarm
27	126	BC02_Electrical_Fault	0		0		0	Erro	Alar		0	VRM Belt Conveyor 2 tripped due to electrical cabinet fault
28	127	BE01_Ready_missed_Alarm	0		0		0	Erro	Alar		0	VRM Bucket Elevator Tripped on ready signal missed alarm
29	128	BE01_State_missed_Alarm	0		0		0	Erro	Alar		0	VRM Bucket Elevator Tripped on state signal missed alarm
30	129	BE01_Electrical_Fault	0		0		0	Erro	Alar		0	VRM Bucket Elevator Tripped on electrical cabine fault
31	130	BC03_Ready_missed_Alarm	0		0		0	Erro	Alar		0	VRM Belt Conveyor 3 tripped on ready signal missed alarm

Figure 6.3 (WinCC Alarm Logging Configuration)

When I opened the alarm logging section, I see multiple different sections for different applications. Here I am using "Analog Alarm" section for analog alarm where I can add tags, its alarm limits and alarm number in "Limit Values" section while in "Messages" section, I can write all the messages that supposed to appear in Alarm list screen with its alarm number. When I put analog value in alarm, it asked me for upper limit or lower limit which means it asking for alarm value greater than or smaller than I put as alarm value. If I choose upper limit it will generate alarm when analog value go above this limit and I choose lower value it will generate alarm when analog value go below this value. For the Boolean alarm, I used "Alarm" section in "Error" section, in this section I can add respective tag, its message that should appear in alarm list and its number. I have created 55 analog alarm and 108 Boolean messages every detail on it. After finalizing the alarm screen, figure 6.3 shows its final appearance. Alarm number is very important in configuration of alarm so for the analog alarm, I have used message number starting from 1 to 99 and for Boolean alarm I started using the number from 100 to onwards.



London South Bank University
EST 1892
Centralised Control System For Cement Plant
Supervised by Dr Perry Xiao
FCS01
FCS02
FCS03
Trend
Alarm List
Print Screen
WinCC AlarmControl

	Date	Time	Number	Point of error	Message text
1	17/05/22	06:54:34 PM	200		Weigh Feeder Tripped on State Signal Missed Alarm
2	17/05/22	06:54:51 PM	200		Weigh Feeder Tripped on State Signal Missed Alarm
3	03/06/22	04:43:59 PM	22		ID Fan Ampere higher than 425A
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					
31					
32					
33					
34					
35					
36					
37					
38					

Figure 6.4 (WinCC Alarm Picture)

LIST OF FIGURES

INDEX	PAGE NO.
Figure 1.1 System Block Diagram	5
Figure 2.1 Simulator	7
Figure 3.1 Hardware Configuration	11
Figure 3.2 PLC Scan Cycle	13
Figure 3.3 Group Motor Block	16
Figure 3.4 Single Motor Block	17
Figure 3.5 Analog Input Scaling Block	18
Figure 3.6 Analog I/O Scaling Block	19
Figure 3.7 Alarm Generating Circuit	21
Figure 4.1 CPU Ethernet Configuration	23
Figure 4.2 CPU Connection Configuration	24
Figure 4.3 Frequency Table for Sending Data	25

Figure 4.4	GET Block For Data Receiving	26
Figure 4.5	PUT Block For Data Sending	27
Figure 5.1	Header Screen	29
Figure 5.2	Screen Switching Configuration	30
Figure 5.3	WinCC Main Screen Appearance	30
Figure 5.4	FCS01 Running Screen	31
Figure 5.5	Control Faceplate	32
Figure 5.6	FCS02 Running Screen	33
Figure 5.7	FCS03 Running Screen	34
Figure 6.1	Trend Configuration Window	37
Figure 6.2	WinCC Trend Picture	38
Figure 6.3	WinCC Alarm Logging Configuration	39
Figure 6.4	WinCC Alarm Picture	40

References

PLC startup references:

https://support.industry.siemens.com/cs/document/34053758/what-is-the-difference-between-a-restart-(warm-restart)-cold-restart-and-hot-restart-of-a-s7-400-cpu-?dti=0&lc=en-WW

PLC Scan Cycle:

https://instrumentationtools.com/understanding-the-scan-cycle-of-siemens-plc/

Simatic Step7 Help

WinCC Help

Component Datasheets

Project simulation youtube link:

https://youtu.be/FzP5DXTu99g

Hardwired relay logic example:

https://electrical-engineering-portal.com/modernizing-hardwired-relay-logic-with-plcs

Reason to choose the project:

https://easwaikato.co.nz/relays-vs-plcs/

Difference between Step7 & TIA portal:

https://support.industry.siemens.com/forum/WW/en/posts/difference-between-tia-portal-simatic-manager-and-pcs7/132924#:~:text=TIA%20Portal%20is%20new%20software,supported%20by%20current%20TIA%20versions.

PLC analog scaling formula:

https://engineerscommunity.com/t/scale-analog-values-using-siemens-plc/7064

FB112 - <offline>

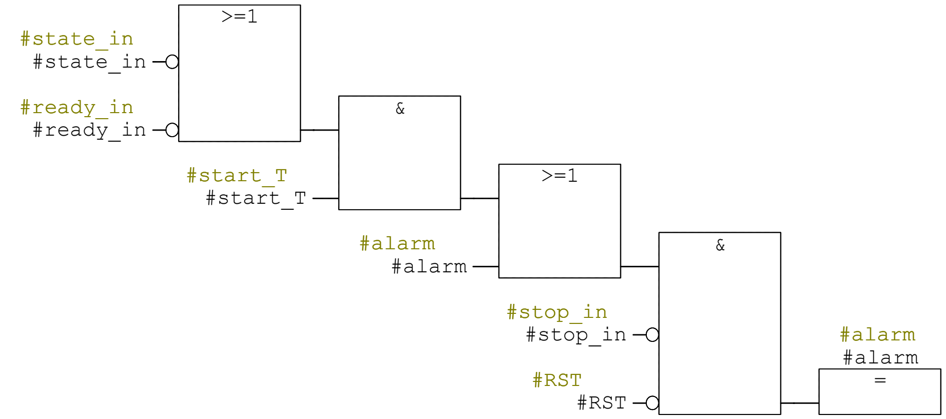
"InterLock Motor Control"

Name: **Family:**
Author: **Version:** 1.0
Block version: 2
Time stamp Code: 03/12/2022 01:45:06 AM
Interface: 01/28/2022 04:19:12 PM
Lengths (block/logic/data): 00488 00324 00000

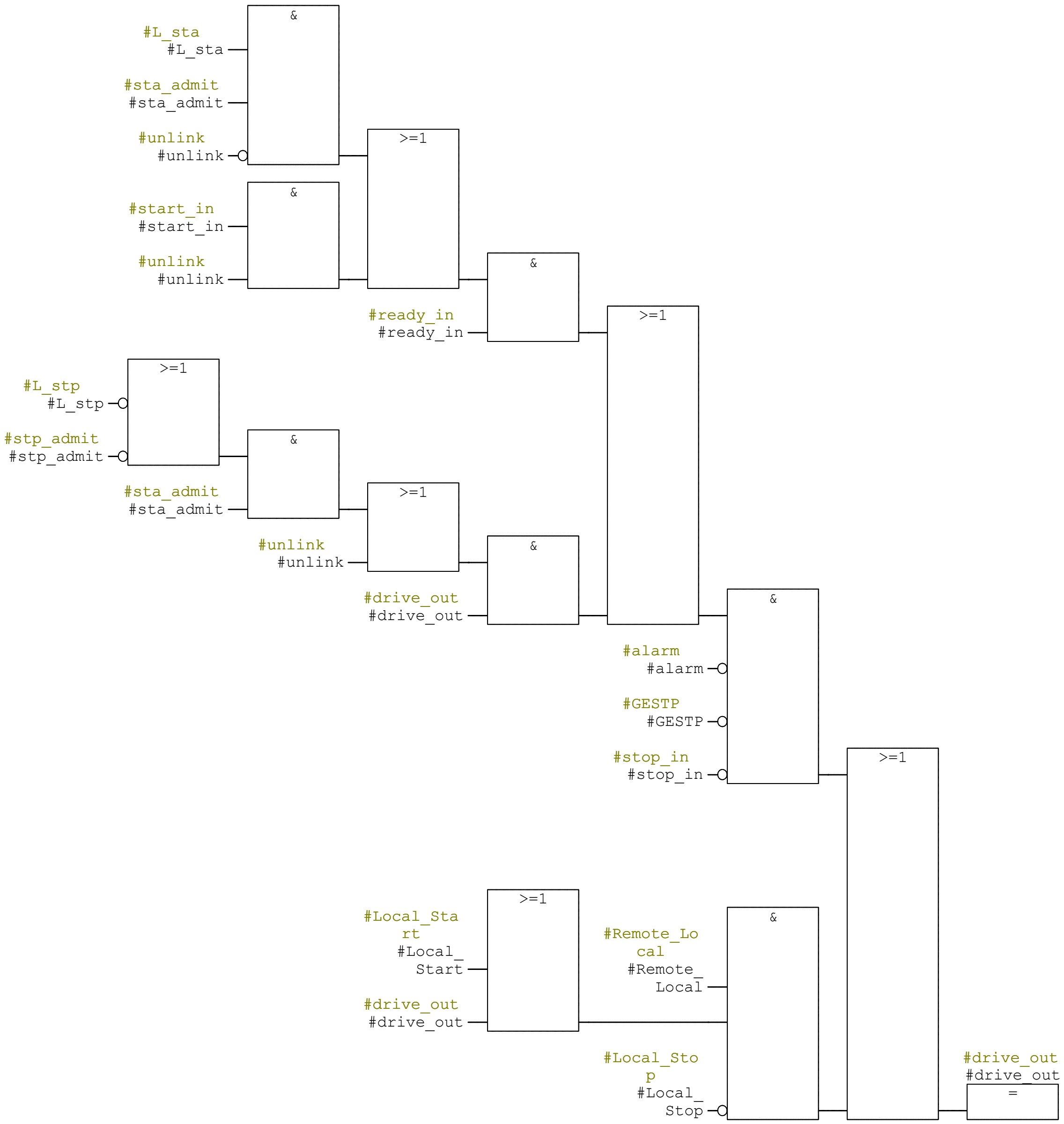
Name	Data Type	Address	Initial Value	Comment
IN		0.0		
unlink	Bool	0.0	FALSE	
start_in	Bool	0.1	FALSE	
stop_in	Bool	0.2	FALSE	
L_start	Bool	0.3	FALSE	
L_stop	Bool	0.4	FALSE	
Remote_Local	Bool	0.5	FALSE	
Local_Start	Bool	0.6	FALSE	
Local_Stop	Bool	0.7	FALSE	
sta_admit	Bool	1.0	FALSE	
stp_admit	Bool	1.1	FALSE	
ready_in	Bool	1.2	FALSE	
state_in	Bool	1.3	FALSE	
start_T	Bool	1.4	FALSE	
RST	Bool	1.5	FALSE	
GESTP	Bool	1.6	FALSE	
OUT		0.0		
drive_out	Bool	2.0	FALSE	
next_ad	Bool	2.1	FALSE	
last_ad	Bool	2.2	FALSE	
motor	Byte	3.0	B#16#0	
IN_OUT		0.0		
STAT		0.0		
alarm	Bool	4.0	FALSE	
L_sta	Bool	4.1	FALSE	
L_stp	Bool	4.2	FALSE	
temp	Int	6.0	0	
TEMP		0.0		

Block: FB112

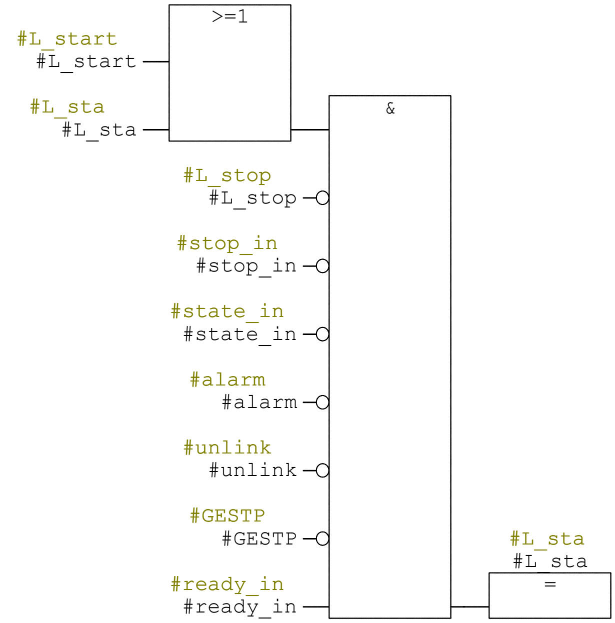
Network: 1



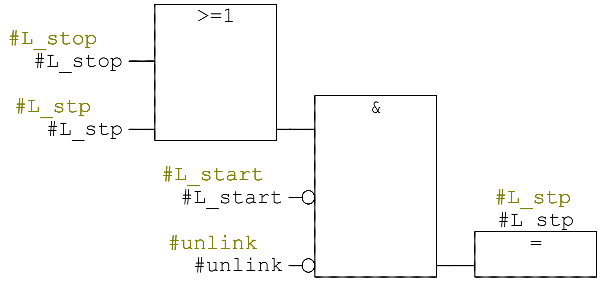
Network: 2



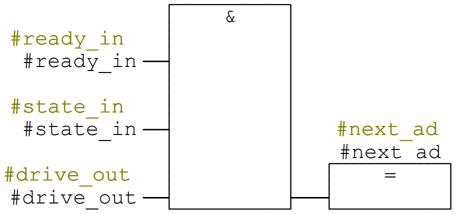
Network: 3



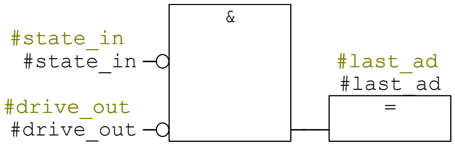
Network: 4



Network: 5



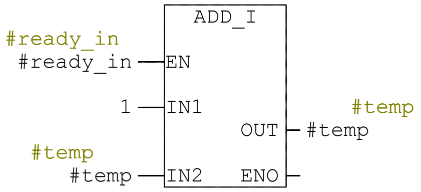
Network: 6



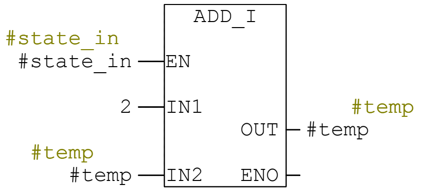
Network: 7

```
      L     0
      T     #temp  #temp
```

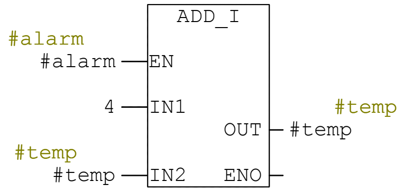
Network: 8



Network: 9



Network: 10



Network: 11

```
      L     #temp   #temp
      T     #motor  #motor
```

FB114 - <offline>

"Conditional M.C.C"

```
Name:                          Family:
Author:                        Version: 0.1
                               Block version: 2
Time stamp Code:               04/08/2022 03:47:42 PM
           Interface:          06/18/2004 05:17:51 PM
Lengths (block/logic/data):    00284  00154  00000
```

Name	Data Type	Address	Initial Value	Comment
IN		0.0		
start_in	Bool	0.0	FALSE	
stop_in	Bool	0.1	FALSE	
start_ad	Bool	0.2	FALSE	
ready_in	Bool	0.3	FALSE	
state_in	Bool	0.4	FALSE	
start_T	Bool	0.5	FALSE	
OUT		0.0		
drive_out	Bool	2.0	FALSE	
motor	Byte	3.0	B#16#0	
IN_OUT		0.0		
STAT		0.0		
alarm	Bool	4.0	FALSE	
temp	Int	6.0	0	
TEMP		0.0		

Block: FB114

Network: 1

```
A(
A(
ON    #state_in  #state_in
ON    #ready_in  #ready_in
)
A     #start_T   #start_T
O     #alarm     #alarm
)
AN    #stop_in   #stop_in
=     #alarm     #alarm
```

Network: 2

```
A(
A     #start_in   #start_in
A     #ready_in   #ready_in
O     #drive_out  #drive_out
)
A     #start_ad   #start_ad
AN    #alarm      #alarm
AN    #stop_in    #stop_in
=     #drive_out  #drive_out
```

Network: 3

```
      L     0
      T     #temp  #temp
```

Network: 4

```
      A     #ready_in  #ready_in
      JNB   _001
      L     1
      L     #temp      #temp
      +I
      T     #temp      #temp
_001: NOP   0
```

Network: 5

```
      A     #state_in  #state_in
      JNB   _002
      L     2
      L     #temp      #temp
      +I
      T     #temp      #temp
_002: NOP   0
```

Network: 6

```
      A     #alarm  #alarm
      JNB   _003
      L     4
      L     #temp   #temp
      +I
      T     #temp   #temp
_003: NOP   0
```

Network: 7

```
      L     #temp   #temp
      T     #motor  #motor
```

FC101 - <offline>

"AI Function Block"
Name: **Family:** Standard
Author: **Version:** 1.0
Block version: 2
Time stamp Code: 12/10/2021 02:53:52 PM
Interface: 03/01/2004 04:55:47 PM
Lengths (block/logic/data): 00166 00056 00008

Name	Data Type	Address	Comment
IN		0.0	
AI	Int	0.0	
PV_High	Real	2.0	
PV_Low	Real	6.0	
OUT		0.0	
PV_out	Real	10.0	
IN_OUT		0.0	
TEMP		0.0	
PV_Real	Real	0.0	
Eng_Range	Real	4.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC101

Network: 1

Ä£ÄâÁ¿ÊäÈë

```
      L     #AI       #AI
      ITD
      DTR
      T     #PV_Real  #PV_Real
```

Network: 2

```
      L     #PV_High    #PV_High
      L     #PV_Low     #PV_Low
      -R
      T     #Eng_Range  #Eng_Range
```

Network: 3

```
      L     #PV_Real       #PV_Real
      L     2.764800e+004
      /R
      L     #Eng_Range     #Eng_Range
      *R
      L     #PV_Low        #PV_Low
      +R
      T     #PV_out        #PV_out
```

FC102 - <offline>

"AO/AI Function Block"

Name: **Family:** Standard
Author: **Version:** 1.0
Block version: 2
Time stamp Code: 03/12/2022 01:44:02 AM
Interface: 03/01/2004 05:13:39 PM
Lengths (block/logic/data): 00316 00182 00012

Name	Data Type	Address	Comment
IN		0.0	
PVH	Real	0.0	
PVL	Real	4.0	
AI	Int	8.0	
OUT		0.0	
AO_out	Word	10.0	
PV_out	Real	12.0	
IN_OUT		0.0	
SP	Real	16.0	
TEMP		0.0	
PV_real	Real	0.0	
range	Real	4.0	
MR3	Real	8.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC102

Network: 1

A/D START

```
      L     #AI       #AI
      ITD
      DTR
      T     #PV_real  #PV_real
```

Network: 2

```
      L     #PVH    #PVH
      L     #PVL    #PVL
      -R
      T     #range  #range
```

Network: 3

```
      L     #PV_real  #PV_real
      L     #range    #range
      *R
      T     #MR3      #MR3
```

Network: 4

```
      L     #MR3           #MR3
      L     2.764800e+004
      /R
      T     #MR3           #MR3
```

Network: 5

```
      L     #MR3     #MR3
      L     #PVL     #PVL
      +R
      T     #PV_out  #PV_out
```

Network: 6

```
      A(
      L     #SP   #SP
      L     #PVH  #PVH
      >R
      )
      JNB   _002
      L     #PVH  #PVH
      T     #SP   #SP
_002: NOP   0
```

Network: 7

```
      A(
      L     #SP   #SP
      L     #PVL  #PVL
      <R
      )
      JNB   _003
      L     #PVL  #PVL
      T     #SP   #SP
_003: NOP   0
```

Network: 8

```
      L     #SP   #SP
      L     #PVL  #PVL
      -R
      T     #MR3  #MR3
```

Network: 9

```
      L     #MR3    #MR3
      L     #range  #range
      /R
      T     #MR3    #MR3
```

Network: 10

```
      L     #MR3           #MR3
      L     2.764800e+004
      *R
      T     #MR3           #MR3
```

Network: 11

```
      L     #MR3     #MR3
      RND
      T     #AO_out  #AO_out
```

FC1 - <offline>

"Data Sharing"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 06/07/2022 02:58:12 PM
Interface: 11/14/2021 01:41:44 AM
Lengths (block/logic/data): 00228 00130 00000

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC1

Network: 1   Life bit for FCS02

```
// send life bit to all FCS
      A     M      0.5
      =     "Send to FCS02".Life_bit_FCS01       DB1.DBX0.0         -- Life bit for FCS02
      A     M      0.5
      =     "Send to FCS03".life_bit_FCS01       DB3.DBX0.0         -- Life bit for FCS03

//  Receive Life bit from all FCS
      A     "Receive From FCS02".Life_bit_FCS02  DB2.DBX0.0         -- Life bit FCS02
      =     M    100.0
      A     "Receive From FCS03".Life_bit_FCS03  DB4.DBX0.0         -- Life bit FCS03
      =     M    100.1
```

Network: 2   Sharing Signals

```
      A     "Hopper Feeding Belt R"                        I4.3                -- Hopper Feeding Belt Ready
      =     "Send to FCS02".Feeding_Belt_R                 DB1.DBX0.1          -- Hopper Feeding Belt Ready
      A     "Hopper Feeding Belt S"                        I4.4                -- Hopper Feeding Belt State
      =     "Send to FCS02".Feeding_Belt_S                 DB1.DBX0.2          -- Hopper Feeding Belt State
      A     "Hopper Feeding Belt A"                        I4.5                -- Hopper Feeding Belt Alarm
      =     "Send to FCS02".Feeding_Belt_A                 DB1.DBX0.3          -- Hopper Feeding Belt Alarm
      A     "Receive From FCS02".Hopper_Feeding_Belt_Star  DB2.DBX0.1          -- Hopper Feeding Belt Start Command
      =     "Control".Hopper_FeedingBelt_Start             DB10.DBX3.6         -- Hopper Feeding Belt Start cmd
      A     "Receive From FCS02".Hopper_Feeding_Belt_Stop  DB2.DBX0.2          -- Hopper Feeding Belt Stop Command
      =     "Control".Hopper_Feeding_Belt_Stop             DB10.DBX3.7         -- Hopper Feeding Belt Stop cmd
      A     "Reclaimer R"                                  I4.6                -- Reclaimer Ready Signal
      =     "Send to FCS02".Reclaimer_R                    DB1.DBX0.4          -- Reclaimer Ready Signal
      A     "Reclaimer S"                                  I4.7                -- Reclaimer State Signal
      =     "Send to FCS02".Reclaimer_S                    DB1.DBX0.5          -- Reclaimer State Signal
      A     "Reclaimer A"                                  I5.0                -- Reclaimer Alarm Signal
      =     "Send to FCS02".Reclaimer_A                    DB1.DBX0.6          -- Reclaimer Alarm Signal
      A     "Receive From FCS02".Reclaimer_start           DB2.DBX0.3          -- Reclaimer Start Cmd
      =     "Control".Reclaimer_Start                      DB10.DBX6.5         -- Reclaimer Start Command
      A     "Receive From FCS02".Reclaimer_Stop            DB2.DBX0.4          -- Reclaimer Stop Cmd
      =     "Control".Reclaimer_Stop                       DB10.DBX6.6         -- Reclaimer Stop Command
```

FC2 - <offline>

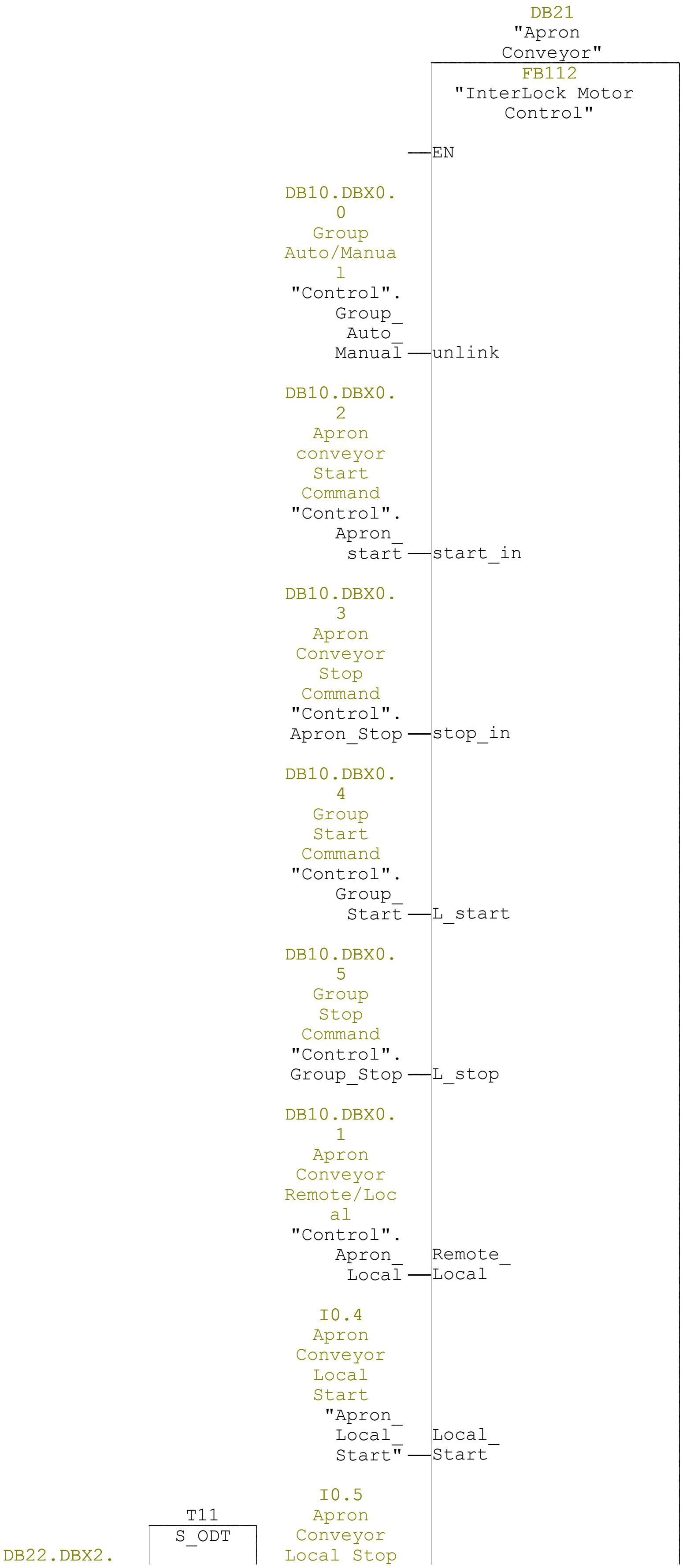
"Motors"

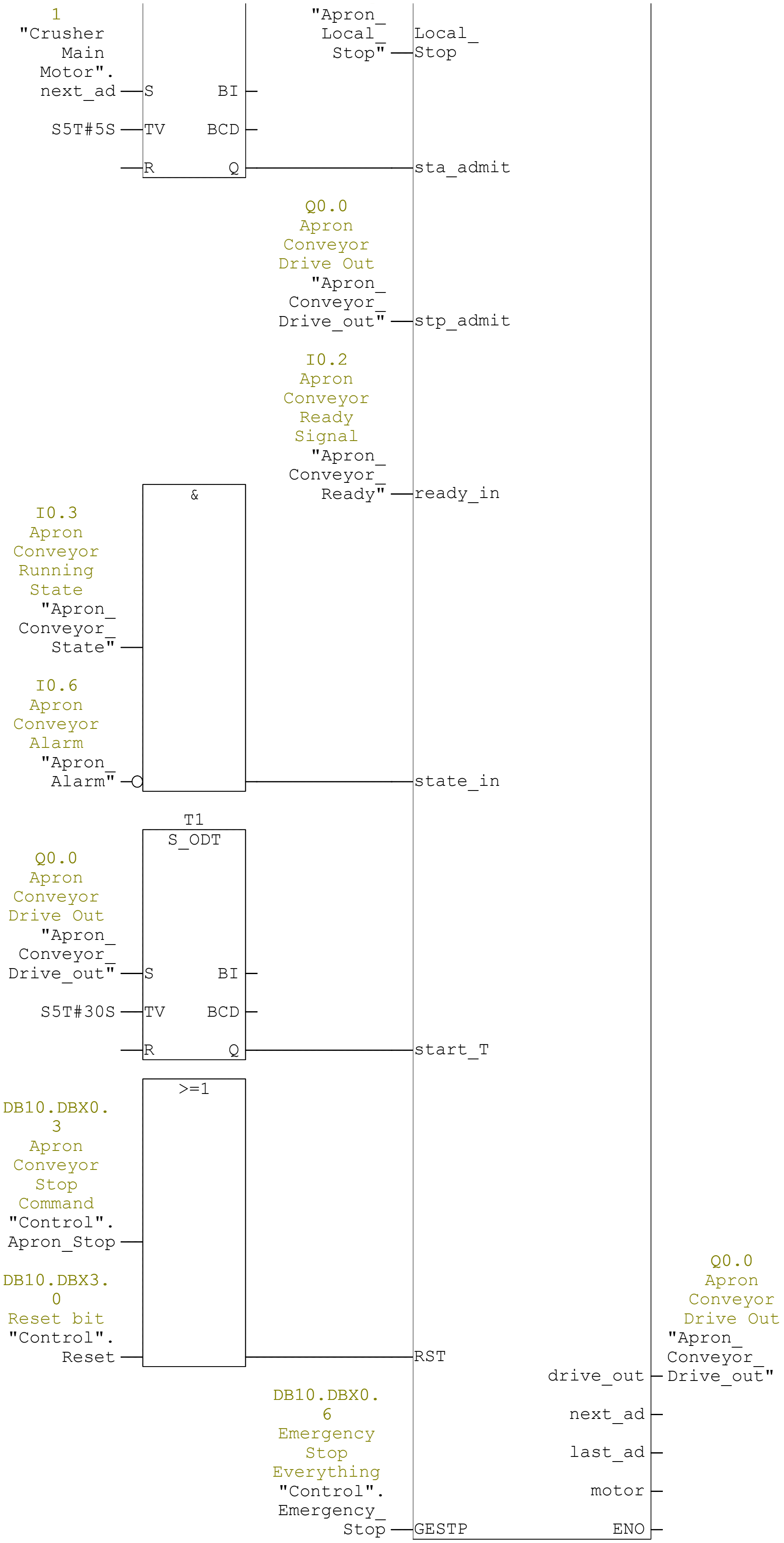
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 06/07/2022 05:14:09 PM
Interface: 12/03/2021 02:15:53 PM
Lengths (block/logic/data): 02774 02648 00008

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

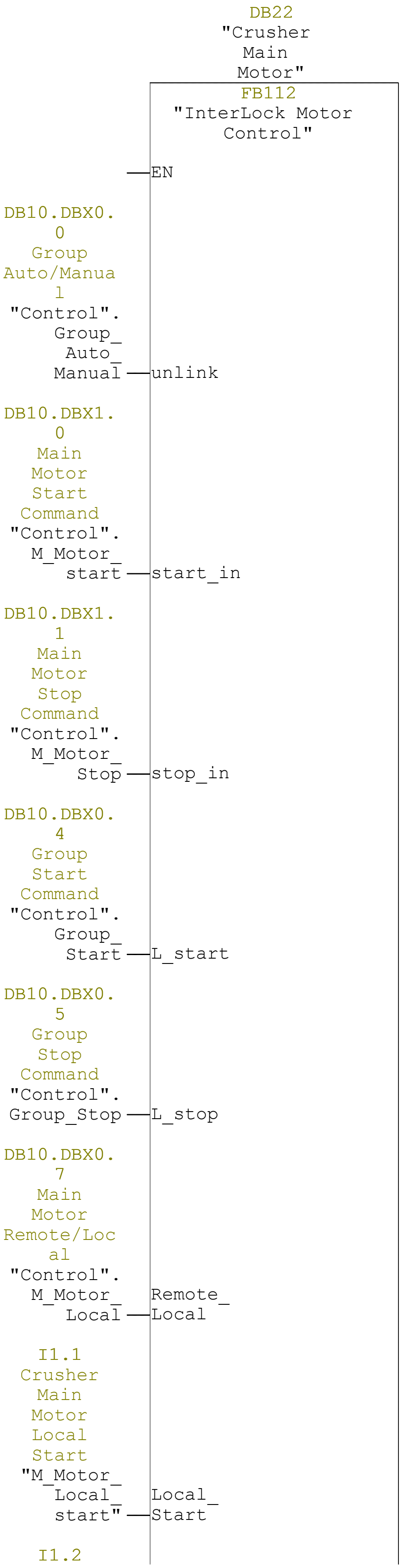
Block: FC2

Network: 1

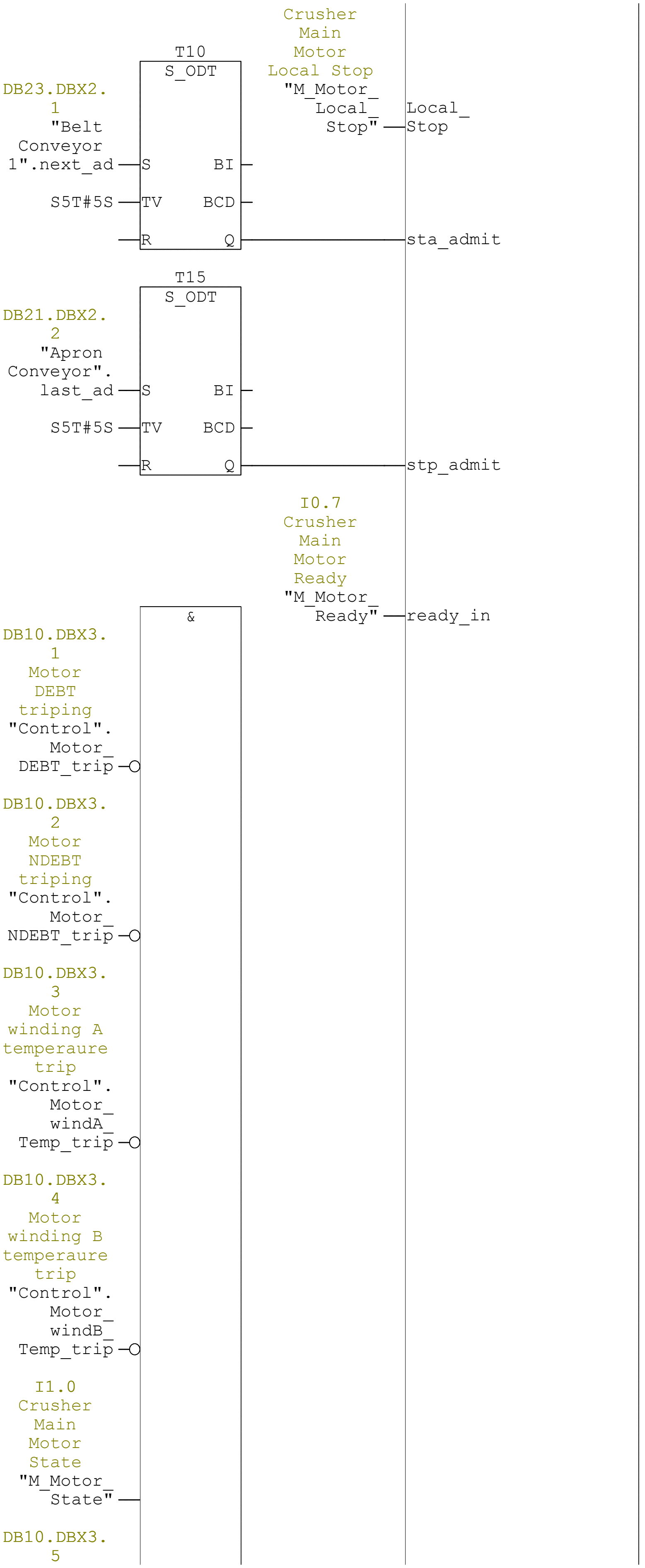

DB21
"Apron
Conveyor"
FB112
"InterLock Motor
Control"
EN
DB10.DBX0.0
Group Auto/Manual
"Control".Group_Auto_Manual
unlink
DB10.DBX0.2
Apron conveyor Start Command
"Control".Apron_start
start_in
DB10.DBX0.3
Apron Conveyor Stop Command
"Control".Apron_Stop
stop_in
DB10.DBX0.4
Group Start Command
"Control".Group_Start
L_start
DB10.DBX0.5
Group Stop Command
"Control".Group_Stop
L_stop
DB10.DBX0.1
Apron Conveyor Remote/Local
"Control".Apron_Local
Remote_Local
I0.4
Apron Conveyor Local Start
"Apron_Local_Start"
Local_Start
I0.5
Apron Conveyor Local Stop
T11
S_ODT
DB22.DBX2.

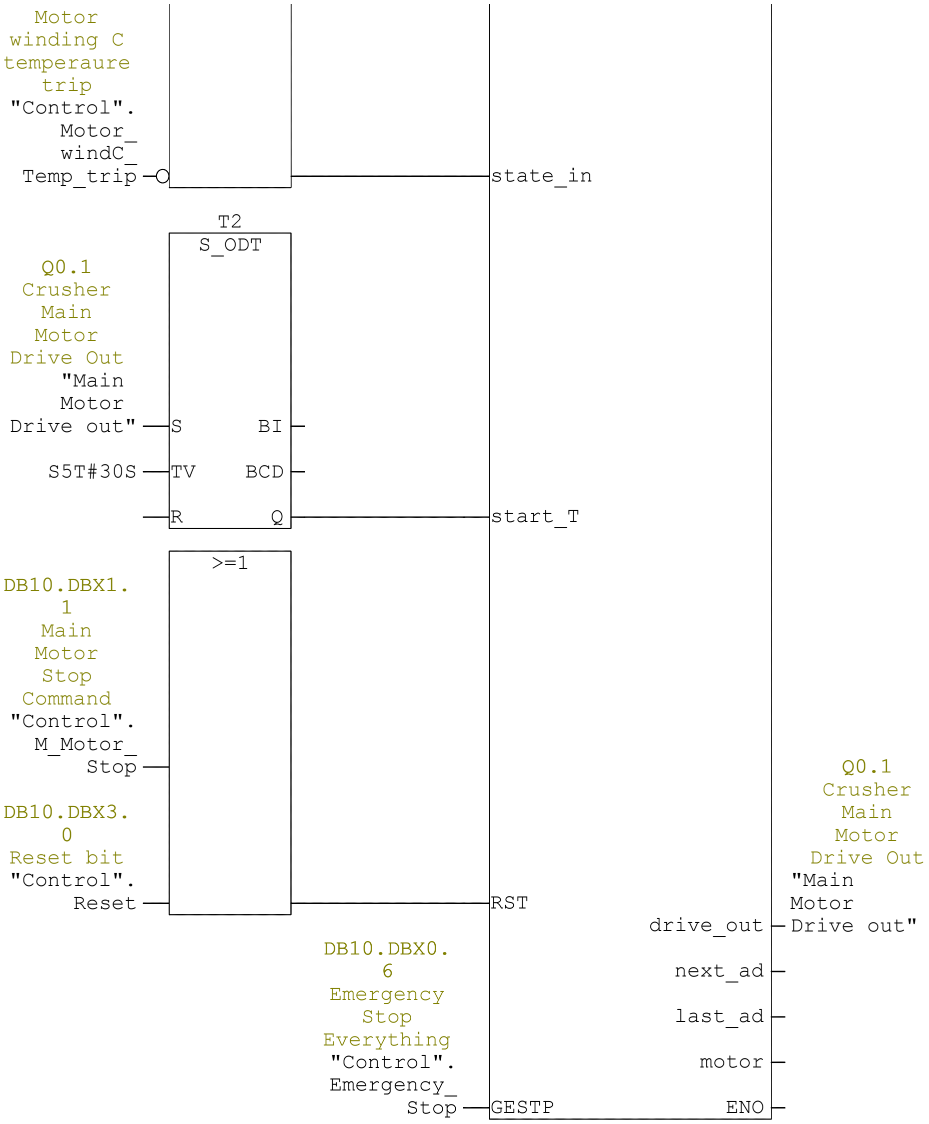

1
"Crusher Main Motor".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
"Apron_Local_Stop"
Local_Stop
sta_admit
Q0.0
Apron Conveyor Drive Out
"Apron_Conveyor_Drive_out"
stp_admit
I0.2
Apron Conveyor Ready Signal
"Apron_Conveyor_Ready"
ready_in
&
I0.3
Apron Conveyor Running State
"Apron_Conveyor_State"
I0.6
Apron Conveyor Alarm
"Apron_Alarm"
state_in
T1
S_ODT
Q0.0
Apron Conveyor Drive Out
"Apron_Conveyor_Drive_out"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
>=1
DB10.DBX0.3
Apron Conveyor Stop Command
"Control".Apron_Stop
DB10.DBX3.0
Reset bit
"Control".Reset
RST
Q0.0
Apron Conveyor Drive Out
drive_out
"Apron_Conveyor_Drive_out"
next_ad
last_ad
DB10.DBX0.6
Emergency Stop Everything
"Control".Emergency_Stop
motor
GESTP
ENO

Network: 2

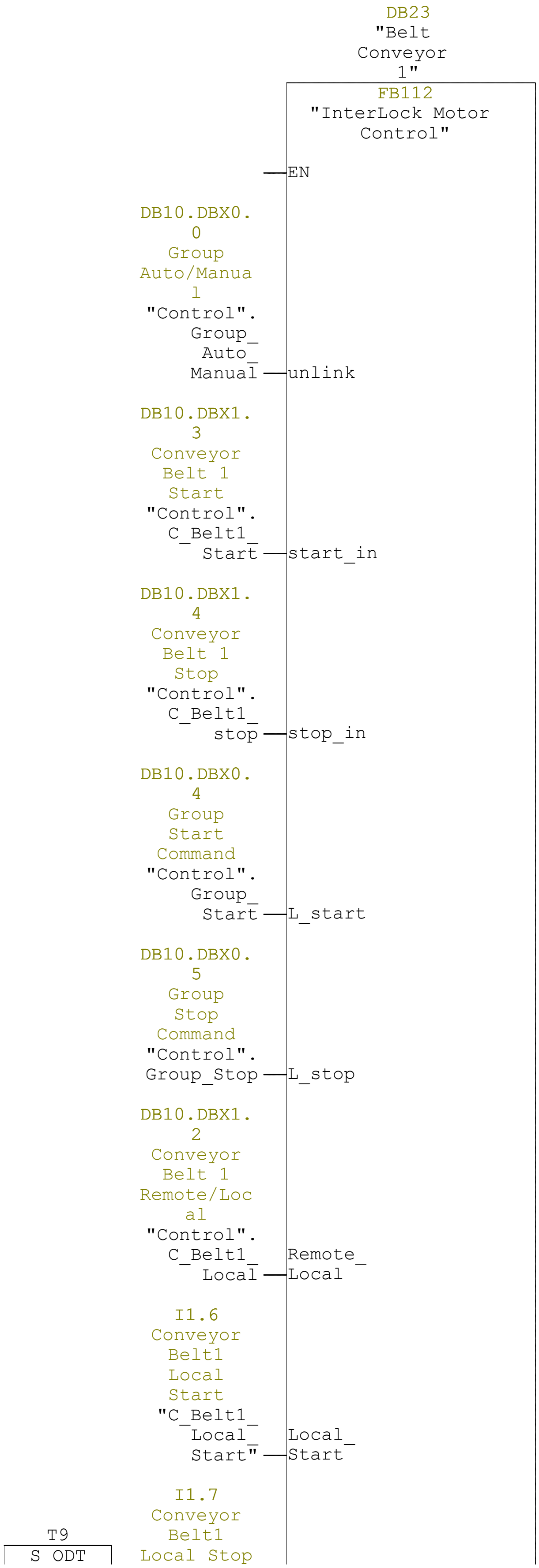


```
                         DB22
                       "Crusher
                         Main
                        Motor"
                 +-------FB112-------+
                 | "InterLock Motor  |
                 |     Control"      |
                 |                   |
               --|EN                 |
                 |                   |
 DB10.DBX0.0     |                   |
 Group           |                   |
 Auto/Manual     |                   |
 "Control".      |                   |
 Group_          |                   |
 Auto_           |                   |
 Manual        --|unlink             |
                 |                   |
 DB10.DBX1.0     |                   |
 Main            |                   |
 Motor           |                   |
 Start           |                   |
 Command         |                   |
 "Control".      |                   |
 M_Motor_        |                   |
 start         --|start_in           |
                 |                   |
 DB10.DBX1.1     |                   |
 Main            |                   |
 Motor           |                   |
 Stop            |                   |
 Command         |                   |
 "Control".      |                   |
 M_Motor_        |                   |
 Stop          --|stop_in            |
                 |                   |
 DB10.DBX0.4     |                   |
 Group           |                   |
 Start           |                   |
 Command         |                   |
 "Control".      |                   |
 Group_          |                   |
 Start         --|L_start            |
                 |                   |
 DB10.DBX0.5     |                   |
 Group           |                   |
 Stop            |                   |
 Command         |                   |
 "Control".      |                   |
 Group_Stop    --|L_stop             |
                 |                   |
 DB10.DBX0.7     |                   |
 Main            |                   |
 Motor           |                   |
 Remote/Local    |                   |
 "Control".      |                   |
 M_Motor_        |Remote_            |
 Local         --|Local              |
                 |                   |
 I1.1            |                   |
 Crusher         |                   |
 Main            |                   |
 Motor           |                   |
 Local           |                   |
 Start           |                   |
 "M_Motor_       |                   |
 Local_          |Local_             |
 start"        --|Start              |
                 |                   |
 I1.2            |                   |
```

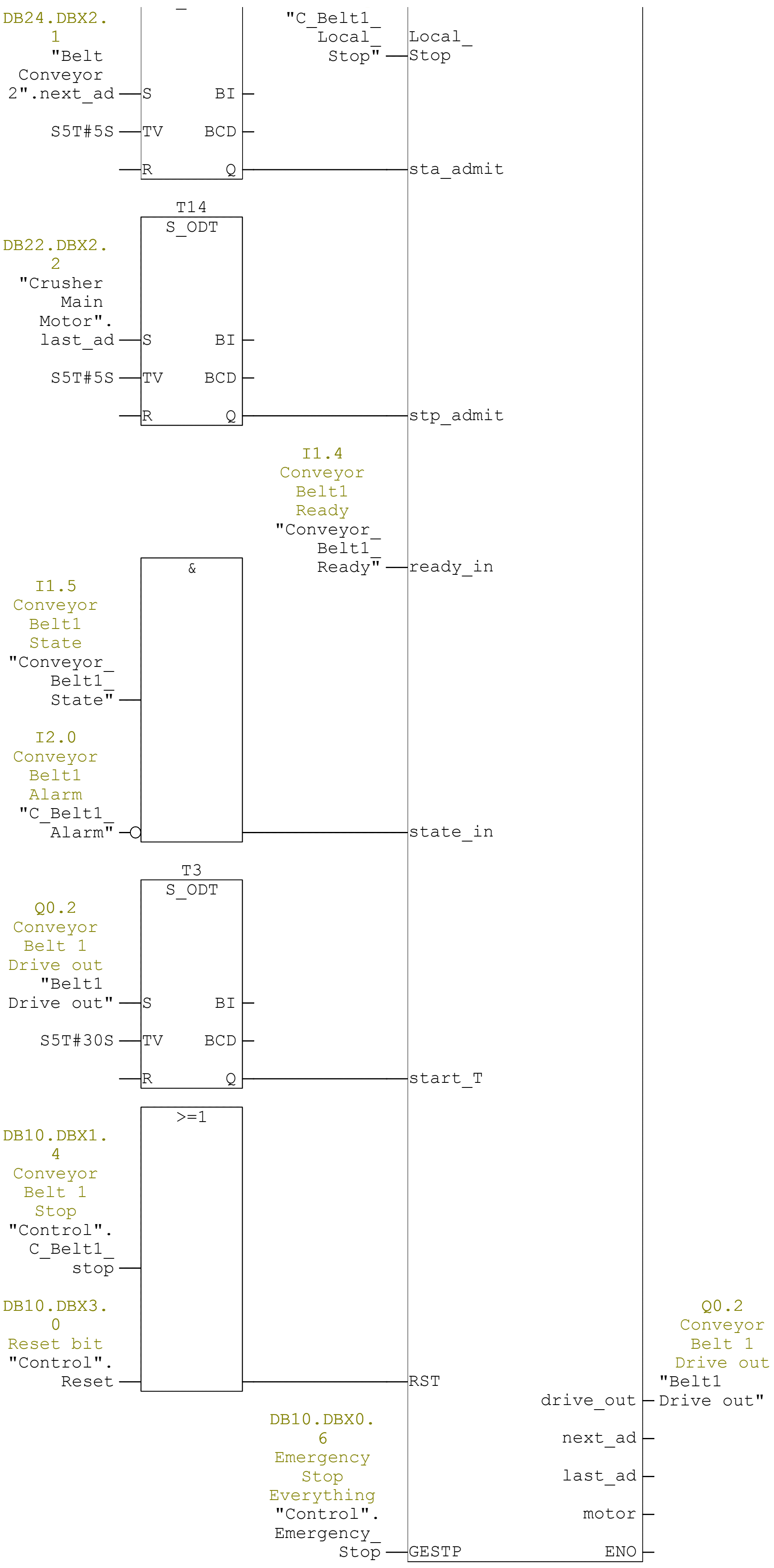

Crusher Main Motor Local Stop
"M_Motor_Local_Stop"
Local_Stop
T10
S_ODT
DB23.DBX2.1
"Belt Conveyor 1".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
sta_admit
T15
S_ODT
DB21.DBX2.2
"Apron Conveyor".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I0.7
Crusher Main Motor Ready
"M_Motor_Ready"
ready_in
&
DB10.DBX3.1
Motor DEBT triping
"Control".Motor_DEBT_trip
DB10.DBX3.2
Motor NDEBT triping
"Control".Motor_NDEBT_trip
DB10.DBX3.3
Motor winding A temperaure trip
"Control".Motor_windA_Temp_trip
DB10.DBX3.4
Motor winding B temperaure trip
"Control".Motor_windB_Temp_trip
I1.0
Crusher Main Motor State
"M_Motor_State"
DB10.DBX3.5


Motor winding C temperaure trip
"Control". Motor_ windC_ Temp_trip
state_in
T2
S_ODT
Q0.1 Crusher Main Motor Drive Out
"Main Motor Drive out"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
>=1
DB10.DBX1. 1 Main Motor Stop Command
"Control". M_Motor_ Stop
DB10.DBX3. 0 Reset bit
"Control". Reset
RST
Q0.1 Crusher Main Motor Drive Out
"Main Motor Drive out"
drive_out
DB10.DBX0. 6 Emergency Stop Everything
next_ad
last_ad
"Control". Emergency_ Stop
motor
GESTP
ENO

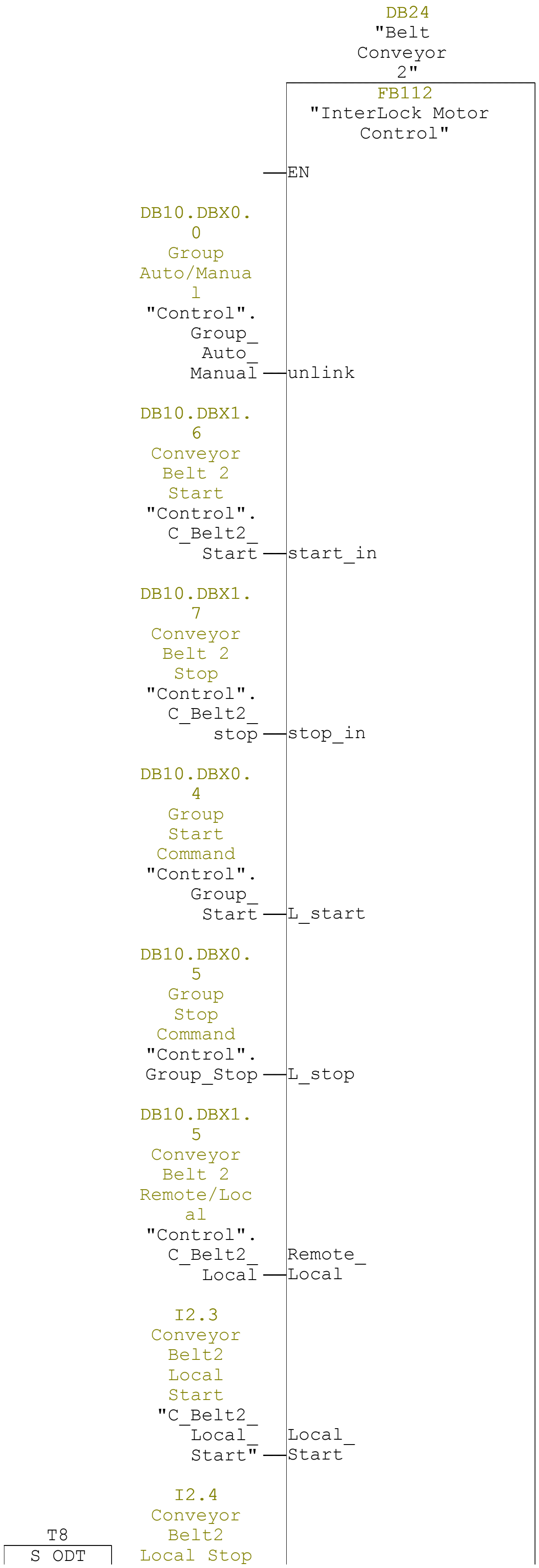
Network: 3

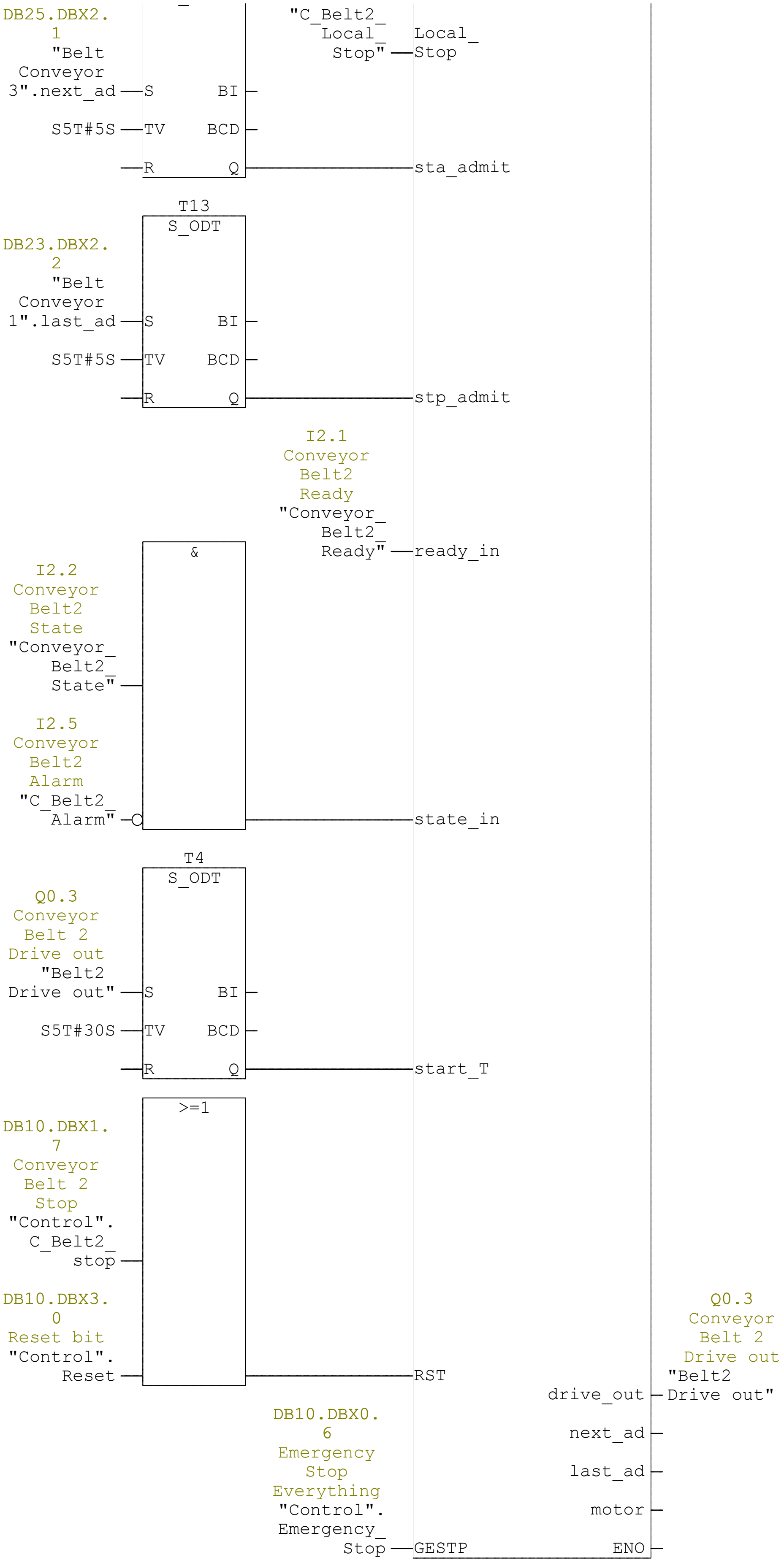


```
                                   DB23
                                  "Belt
                                 Conveyor
                                    1"
                              FB112
                          "InterLock Motor
                              Control"

                        —EN
DB10.DBX0.0
Group Auto/Manual
"Control".Group_Auto_Manual —unlink
DB10.DBX1.3
Conveyor Belt 1 Start
"Control".C_Belt1_Start —start_in
DB10.DBX1.4
Conveyor Belt 1 Stop
"Control".C_Belt1_stop —stop_in
DB10.DBX0.4
Group Start Command
"Control".Group_Start —L_start
DB10.DBX0.5
Group Stop Command
"Control".Group_Stop —L_stop
DB10.DBX1.2
Conveyor Belt 1 Remote/Local
"Control".C_Belt1_Local —Remote_Local
I1.6
Conveyor Belt1 Local Start
"C_Belt1_Local_Start" —Local_Start
I1.7
Conveyor Belt1 Local Stop
T9
S ODT
```

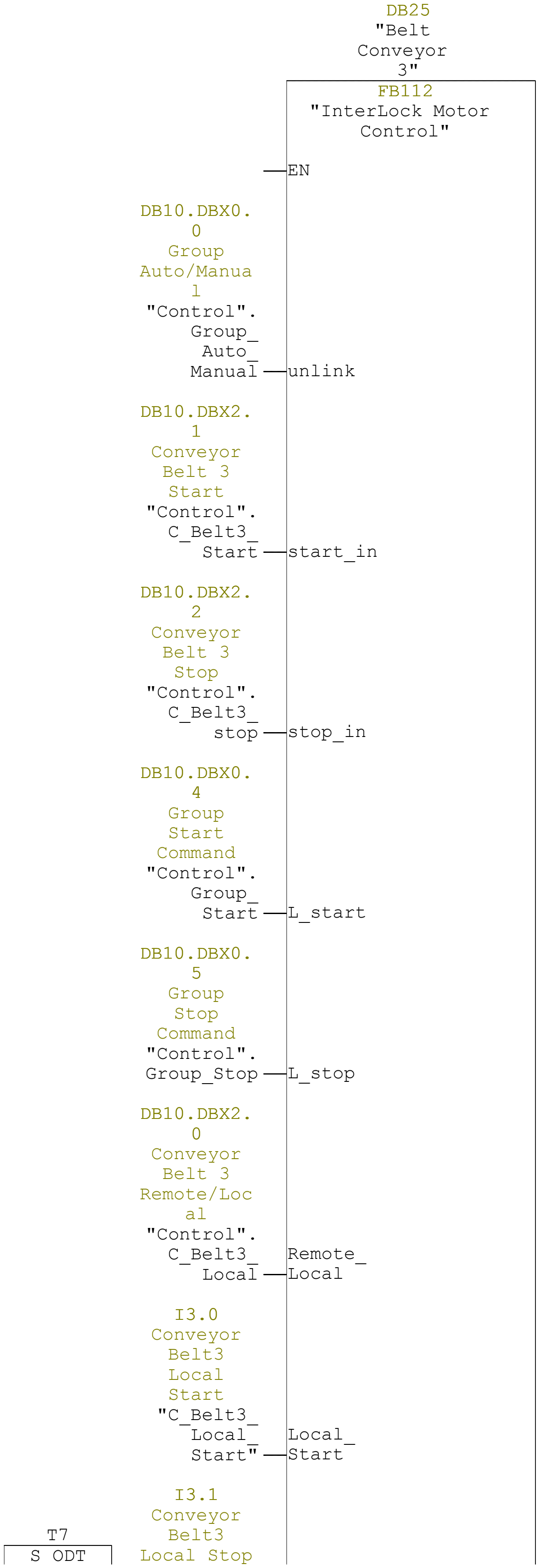

DB24.DBX2.
1
"Belt
Conveyor
2".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
"C_Belt1_
Local_
Stop"
Local_
Stop
sta_admit
T14
S_ODT
DB22.DBX2.
2
"Crusher
Main
Motor".
last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I1.4
Conveyor
Belt1
Ready
"Conveyor_
Belt1_
Ready"
ready_in
&
I1.5
Conveyor
Belt1
State
"Conveyor_
Belt1_
State"
I2.0
Conveyor
Belt1
Alarm
"C_Belt1_
Alarm"
state_in
T3
S_ODT
Q0.2
Conveyor
Belt 1
Drive out
"Belt1
Drive out"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
>=1
DB10.DBX1.
4
Conveyor
Belt 1
Stop
"Control".
C_Belt1_
stop
DB10.DBX3.
0
Reset bit
"Control".
Reset
RST
Q0.2
Conveyor
Belt 1
Drive out
"Belt1
Drive out"
drive_out
next_ad
last_ad
motor
DB10.DBX0.
6
Emergency
Stop
Everything
"Control".
Emergency_
Stop
GESTP
ENO

Network: 4

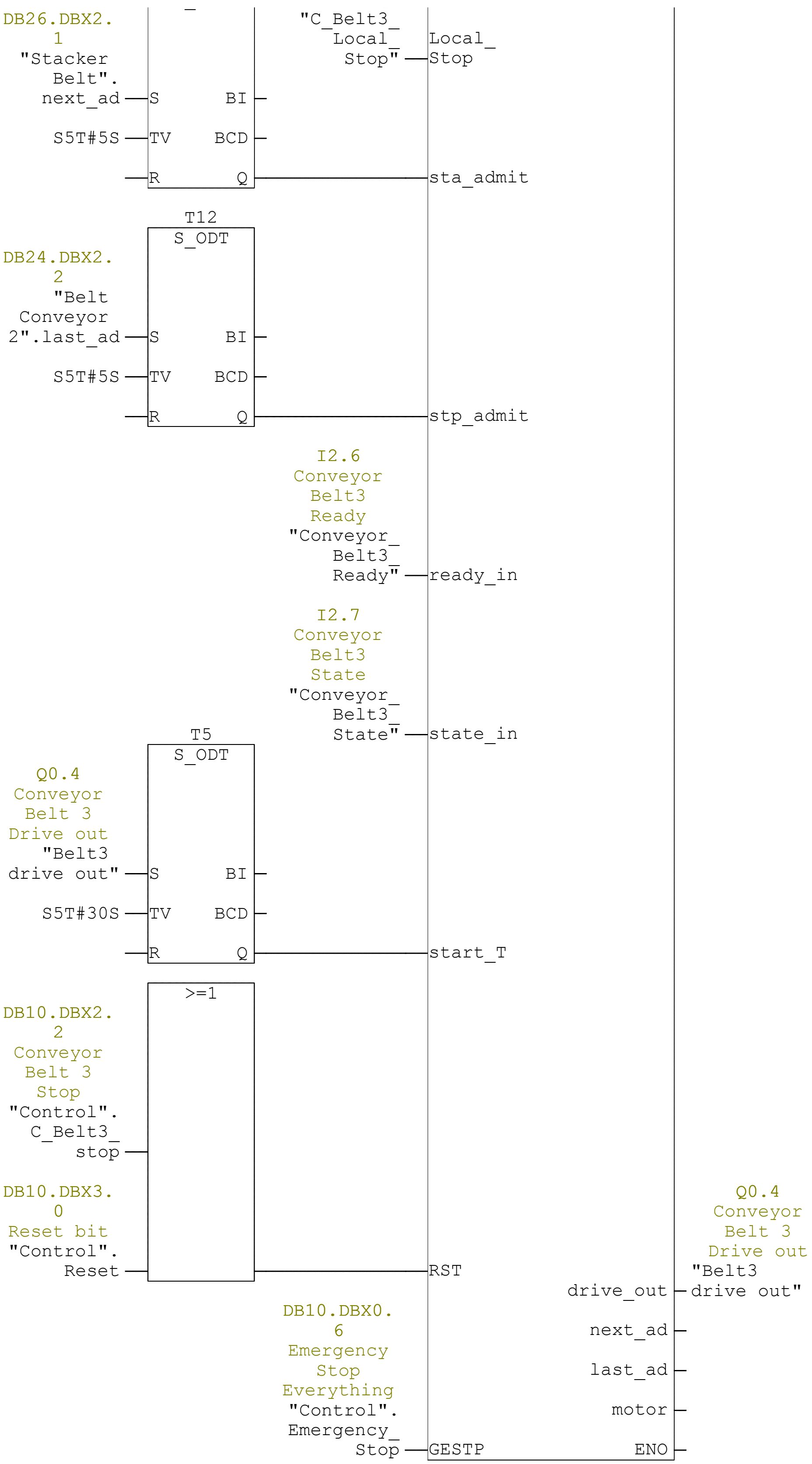

DB24
"Belt Conveyor 2"
FB112
"InterLock Motor Control"
EN
DB10.DBX0.0
Group Auto/Manual
"Control".Group_Auto_Manual
unlink
DB10.DBX1.6
Conveyor Belt 2 Start
"Control".C_Belt2_Start
start_in
DB10.DBX1.7
Conveyor Belt 2 Stop
"Control".C_Belt2_stop
stop_in
DB10.DBX0.4
Group Start Command
"Control".Group_Start
L_start
DB10.DBX0.5
Group Stop Command
"Control".Group_Stop
L_stop
DB10.DBX1.5
Conveyor Belt 2 Remote/Local
"Control".C_Belt2_Local
Remote_Local
I2.3
Conveyor Belt2 Local Start
"C_Belt2_Local_Start"
Local_Start
I2.4
Conveyor Belt2 Local Stop
T8
S_ODT


DB25.DBX2.1
"Belt Conveyor 3".next_ad
S
S5T#5S
TV
R
BI
BCD
Q
"C_Belt2_Local_Stop"
Local_Stop
sta_admit
T13
S_ODT
DB23.DBX2.2
"Belt Conveyor 1".last_ad
S
S5T#5S
TV
R
BI
BCD
Q
stp_admit
I2.1
Conveyor Belt2 Ready
"Conveyor_Belt2_Ready"
ready_in
&
I2.2
Conveyor Belt2 State
"Conveyor_Belt2_State"
I2.5
Conveyor Belt2 Alarm
"C_Belt2_Alarm"
state_in
T4
S_ODT
Q0.3
Conveyor Belt 2 Drive out
"Belt2 Drive out"
S
S5T#30S
TV
R
BI
BCD
Q
start_T
>=1
DB10.DBX1.7
Conveyor Belt 2 Stop
"Control".C_Belt2_stop
DB10.DBX3.0
Reset bit
"Control".Reset
RST
drive_out
Q0.3
Conveyor Belt 2 Drive out
"Belt2 Drive out"
next_ad
last_ad
motor
DB10.DBX0.6
Emergency Stop Everything
"Control".Emergency_Stop
GESTP
ENO

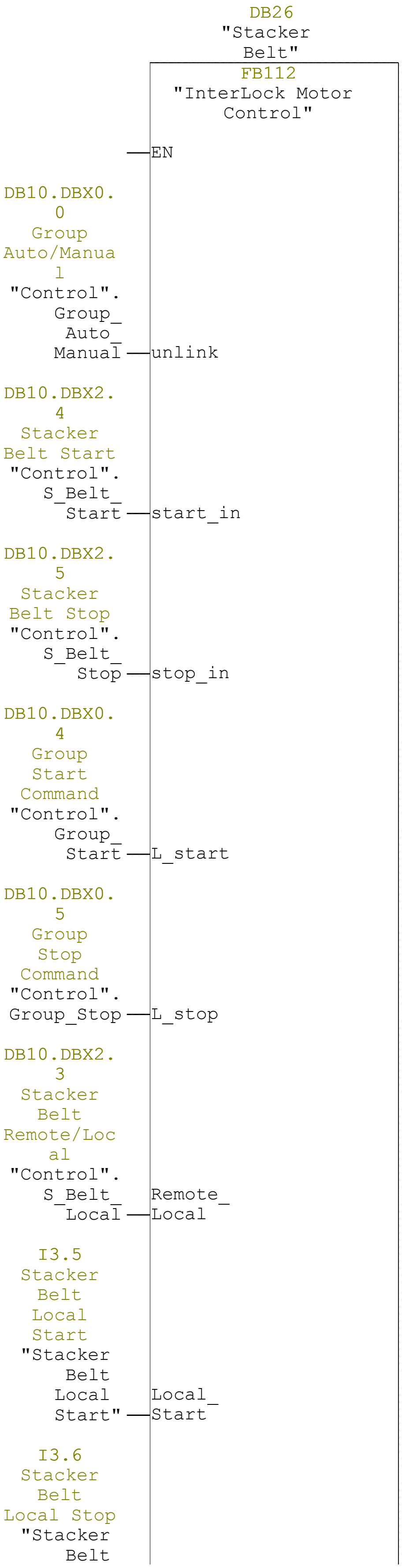
Network: 5



```
                                DB25
                               "Belt
                              Conveyor
                                3"
                          +-----FB112-------------+
                          | "InterLock Motor      |
                          |     Control"          |
                        --|EN                     |
         DB10.DBX0.       |                       |
             0            |                       |
           Group          |                       |
        Auto/Manua        |                       |
             l            |                       |
         "Control".       |                       |
             Group_       |                       |
              Auto_       |                       |
             Manual ------|unlink                 |
         DB10.DBX2.       |                       |
             1            |                       |
          Conveyor        |                       |
           Belt 3         |                       |
           Start          |                       |
         "Control".       |                       |
           C_Belt3_       |                       |
              Start ------|start_in               |
         DB10.DBX2.       |                       |
             2            |                       |
          Conveyor        |                       |
           Belt 3         |                       |
            Stop          |                       |
         "Control".       |                       |
           C_Belt3_       |                       |
               stop ------|stop_in                |
         DB10.DBX0.       |                       |
             4            |                       |
           Group          |                       |
           Start          |                       |
          Command         |                       |
         "Control".       |                       |
             Group_       |                       |
              Start ------|L_start                |
         DB10.DBX0.       |                       |
             5            |                       |
           Group          |                       |
            Stop          |                       |
          Command         |                       |
         "Control".       |                       |
         Group_Stop ------|L_stop                 |
         DB10.DBX2.       |                       |
             0            |                       |
          Conveyor        |                       |
           Belt 3         |                       |
        Remote/Loc        |                       |
             al           |                       |
         "Control".       |                       |
           C_Belt3_       |Remote_                |
              Local ------|Local                  |
            I3.0          |                       |
          Conveyor        |                       |
           Belt3          |                       |
           Local          |                       |
           Start          |                       |
          "C_Belt3_       |                       |
             Local_       |Local_                 |
             Start" ------|Start                  |
            I3.1          |                       |
          Conveyor        |                       |
    T7     Belt3          |                       |
 +-S ODT-+ Local Stop     |                       |
```


DB26.DBX2.1
"Stacker Belt".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
"C_Belt3_Local_Stop"
Local_Stop
sta_admit
T12
S_ODT
DB24.DBX2.2
"Belt Conveyor 2".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I2.6
Conveyor Belt3 Ready
"Conveyor_Belt3_Ready"
ready_in
I2.7
Conveyor Belt3 State
"Conveyor_Belt3_State"
state_in
T5
S_ODT
Q0.4
Conveyor Belt 3 Drive out
"Belt3 drive out"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
>=1
DB10.DBX2.2
Conveyor Belt 3 Stop
"Control".C_Belt3_stop
DB10.DBX3.0
Reset bit
"Control".Reset
RST
Q0.4
Conveyor Belt 3 Drive out
"Belt3 drive out"
drive_out
next_ad
last_ad
motor
DB10.DBX0.6
Emergency Stop Everything
"Control".Emergency_Stop
GESTP
ENO

Network: 6



DB26
"Stacker Belt"
FB112
"InterLock Motor Control"

EN

DB10.DBX0.0
Group Auto/Manual
"Control".Group_Auto_Manual — unlink

DB10.DBX2.4
Stacker Belt Start
"Control".S_Belt_Start — start_in

DB10.DBX2.5
Stacker Belt Stop
"Control".S_Belt_Stop — stop_in

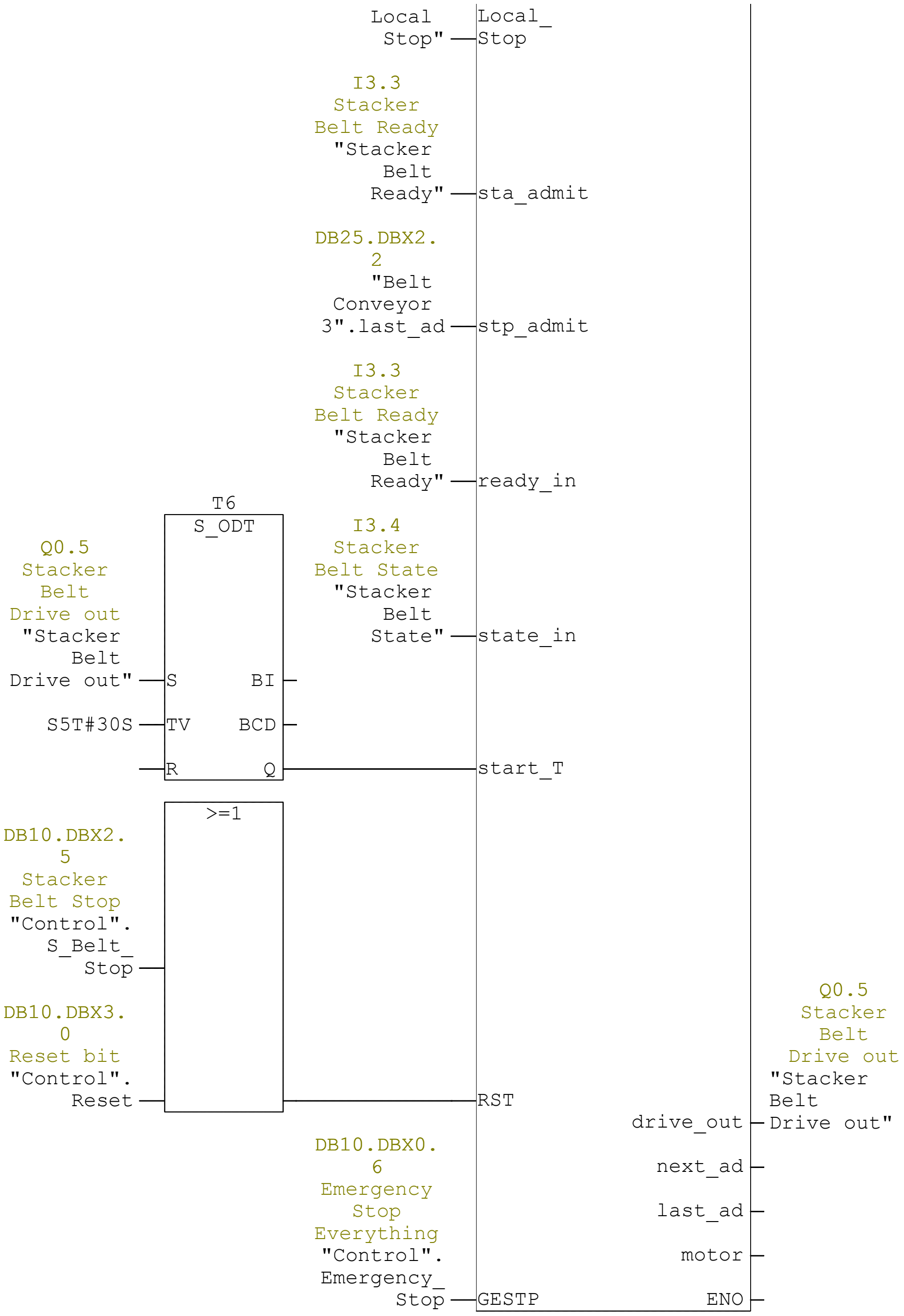
DB10.DBX0.4
Group Start Command
"Control".Group_Start — L_start

DB10.DBX0.5
Group Stop Command
"Control".Group_Stop — L_stop

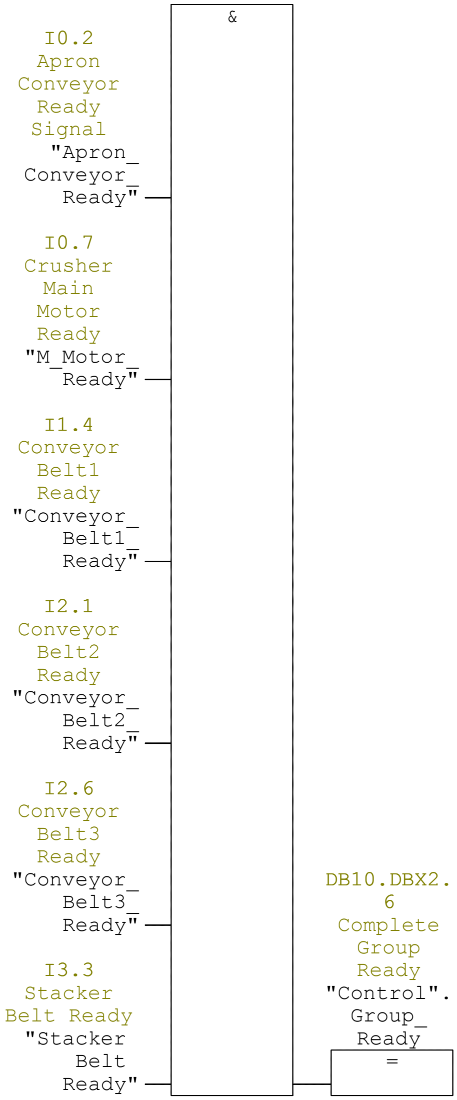
DB10.DBX2.3
Stacker Belt Remote/Local
"Control".S_Belt_Local — Remote_Local

I3.5
Stacker Belt Local Start
"Stacker Belt Local Start" — Local_Start

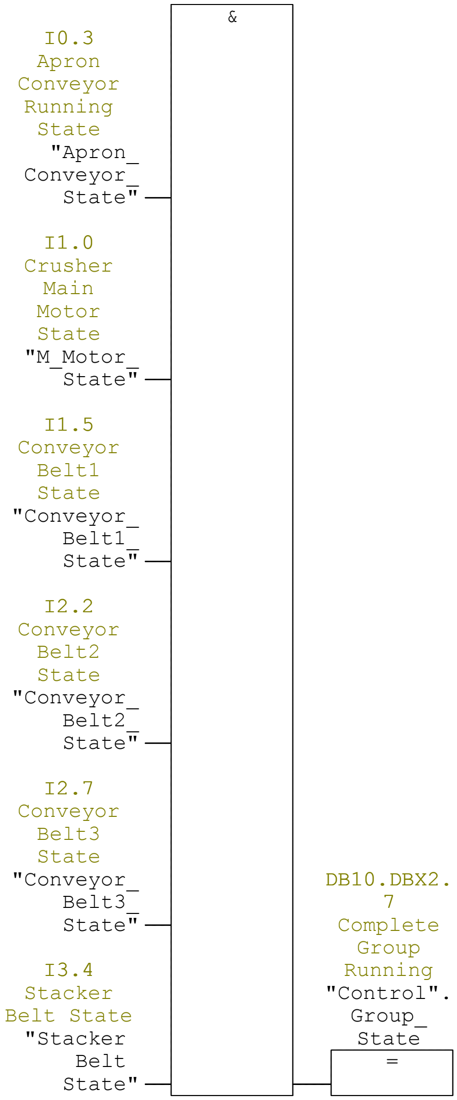
I3.6
Stacker Belt Local Stop
"Stacker Belt


Local
Stop"
Local_
Stop
I3.3
Stacker
Belt Ready
"Stacker
Belt
Ready"
sta_admit
DB25.DBX2.
2
"Belt
Conveyor
3".last_ad
stp_admit
I3.3
Stacker
Belt Ready
"Stacker
Belt
Ready"
ready_in
T6
S_ODT
Q0.5
Stacker
Belt
Drive out
"Stacker
Belt
Drive out"
S
BI
S5T#30S
TV
BCD
R
Q
I3.4
Stacker
Belt State
"Stacker
Belt
State"
state_in
start_T
>=1
DB10.DBX2.
5
Stacker
Belt Stop
"Control".
S_Belt_
Stop
DB10.DBX3.
0
Reset bit
"Control".
Reset
RST
Q0.5
Stacker
Belt
Drive out
"Stacker
Belt
Drive out"
drive_out
DB10.DBX0.
6
Emergency
Stop
Everything
"Control".
Emergency_
Stop
GESTP
next_ad
last_ad
motor
ENO

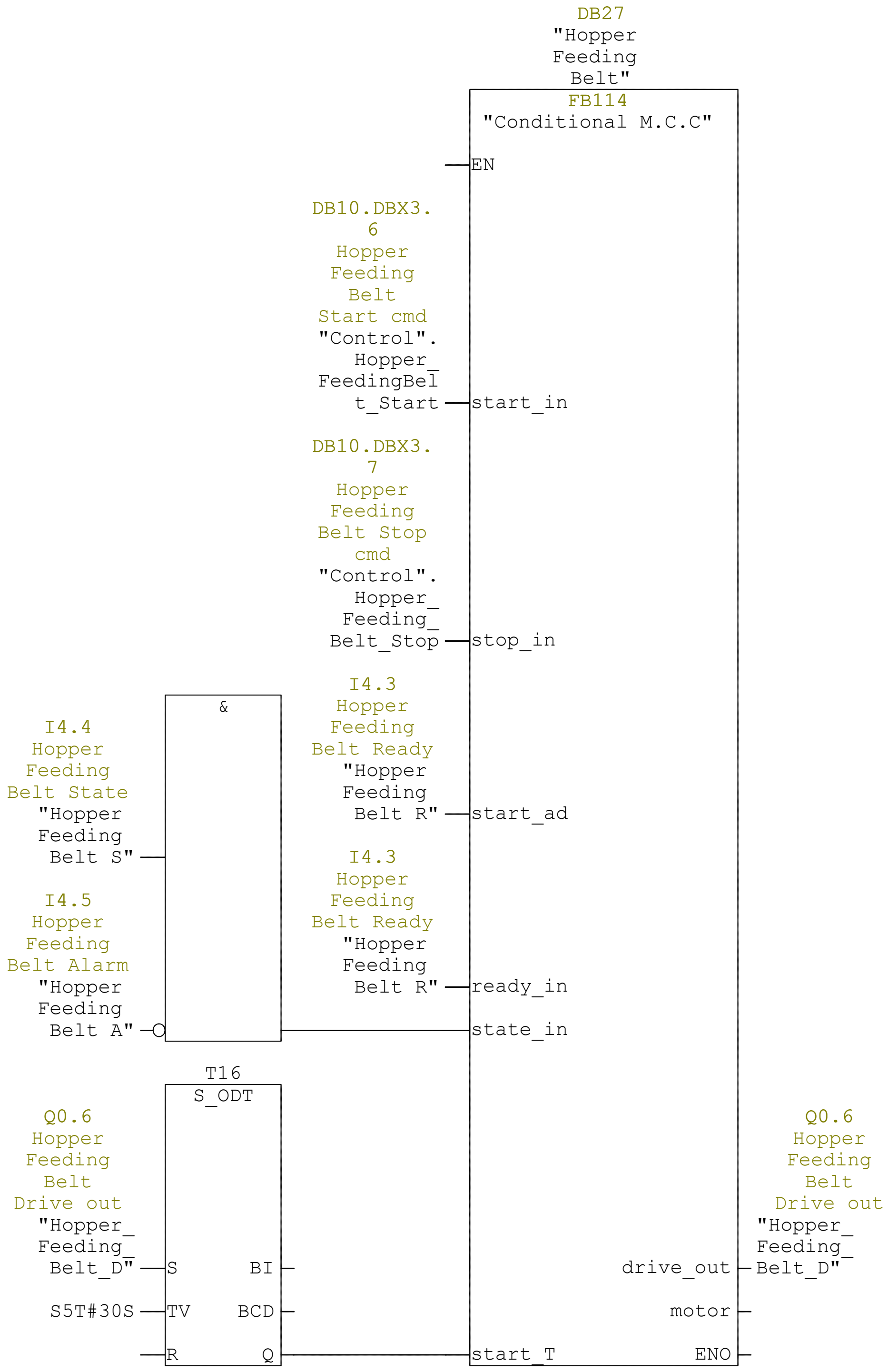
Network: 7 Complete Group Ready



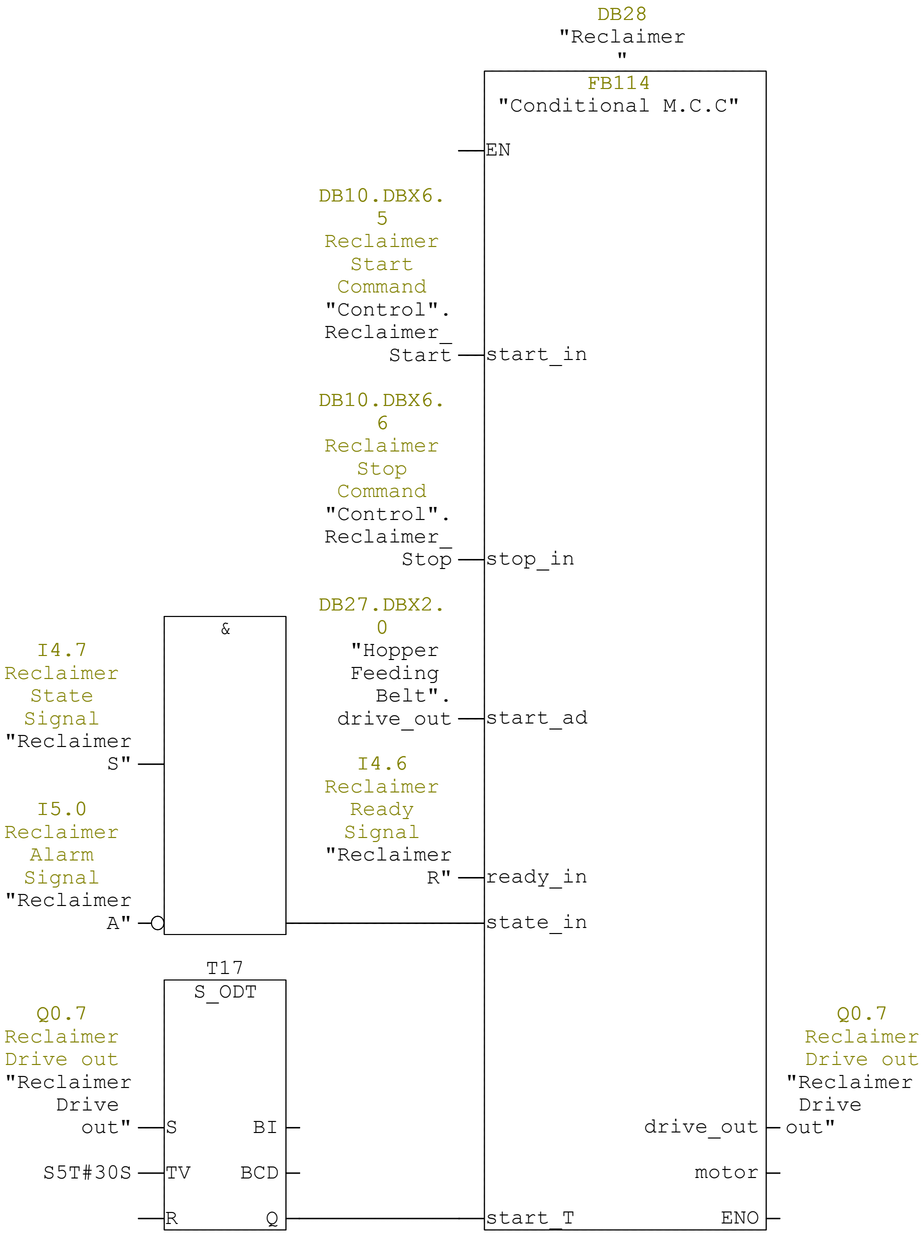
Network: 8 Complete Group Running



Network: 9


DB27
"Hopper Feeding Belt"
FB114
"Conditional M.C.C"
EN
DB10.DBX3.6
Hopper Feeding Belt Start cmd
"Control".Hopper_FeedingBelt_Start
start_in
DB10.DBX3.7
Hopper Feeding Belt Stop cmd
"Control".Hopper_Feeding_Belt_Stop
stop_in
&
I4.4
Hopper Feeding Belt State
"Hopper Feeding Belt S"
I4.5
Hopper Feeding Belt Alarm
"Hopper Feeding Belt A"
I4.3
Hopper Feeding Belt Ready
"Hopper Feeding Belt R"
start_ad
I4.3
Hopper Feeding Belt Ready
"Hopper Feeding Belt R"
ready_in
state_in
T16
S_ODT
Q0.6
Hopper Feeding Belt Drive out
"Hopper_Feeding_Belt_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
drive_out
motor
ENO
Q0.6
Hopper Feeding Belt Drive out
"Hopper_Feeding_Belt_D"

Network: 10


DB28
"Reclaimer
"
FB114
"Conditional M.C.C"
EN
DB10.DBX6.
5
Reclaimer
Start
Command
"Control".
Reclaimer_
Start
start_in
DB10.DBX6.
6
Reclaimer
Stop
Command
"Control".
Reclaimer_
Stop
stop_in
DB27.DBX2.
0
"Hopper
Feeding
Belt".
drive_out
start_ad
I4.6
Reclaimer
Ready
Signal
"Reclaimer
R"
ready_in
&
I4.7
Reclaimer
State
Signal
"Reclaimer
S"
I5.0
Reclaimer
Alarm
Signal
"Reclaimer
A"
state_in
T17
S_ODT
Q0.7
Reclaimer
Drive out
"Reclaimer
Drive
out"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
drive_out
motor
ENO
Q0.7
Reclaimer
Drive out
"Reclaimer
Drive
out"

FC3 - <offline>

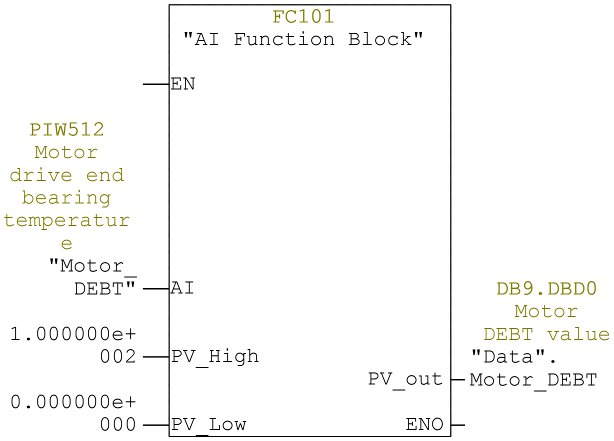
"Analog Parameters"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 03/12/2022 01:56:13 AM
Interface: 03/12/2022 01:19:21 AM
Lengths (block/logic/data): 00998 00874 00018

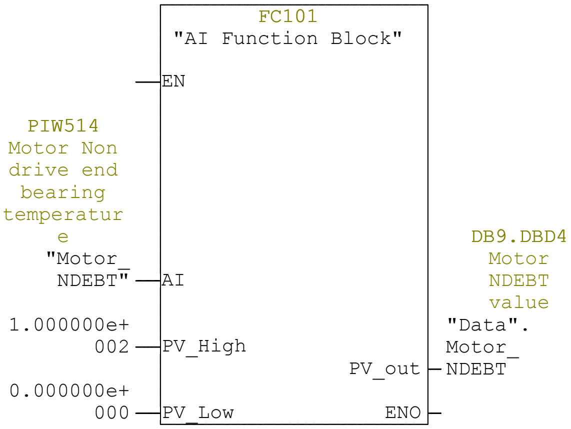
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC3

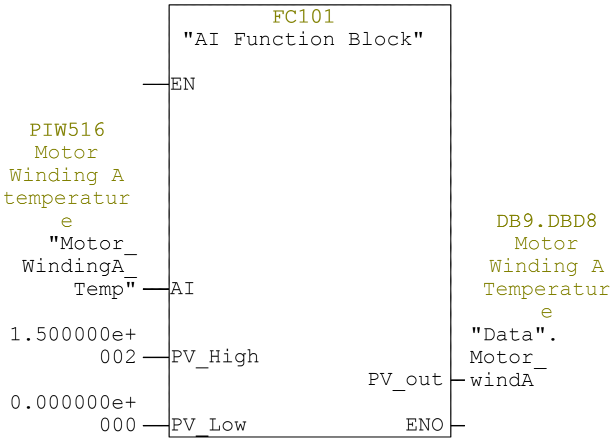
Network: 1



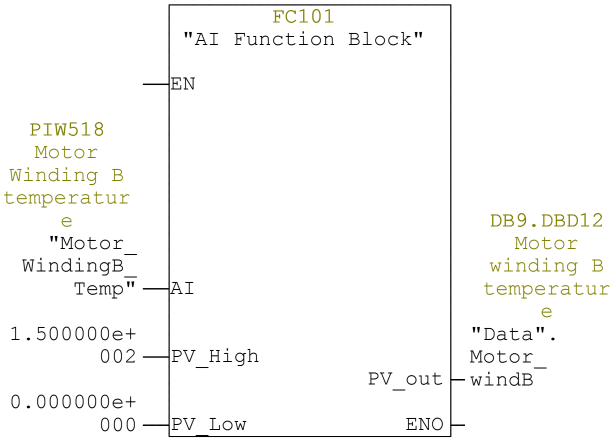
Network: 2



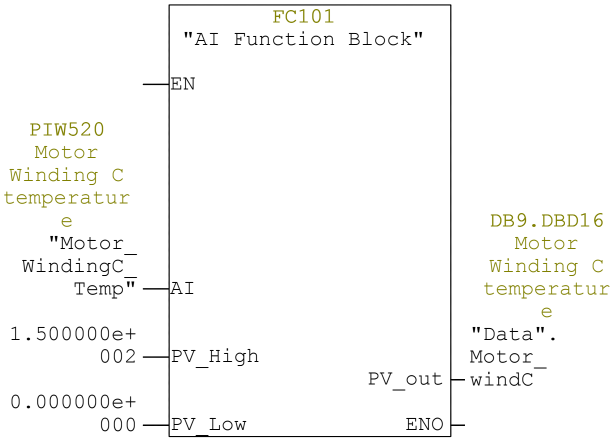
Network: 3



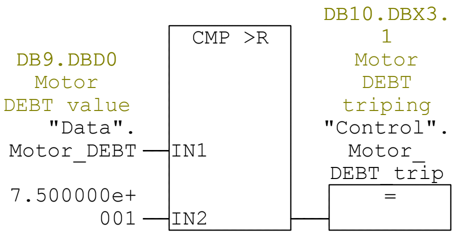
Network: 4



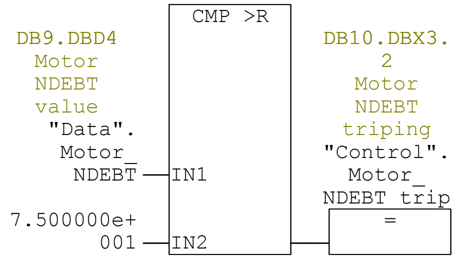
Network: 5



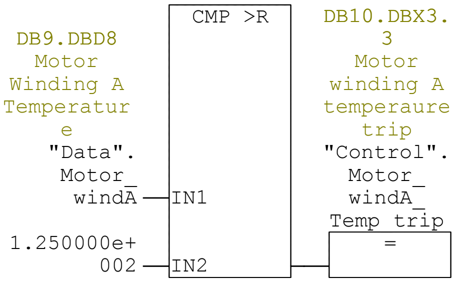
Network: 6



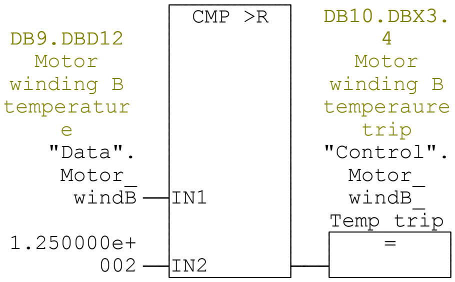
Network: 7



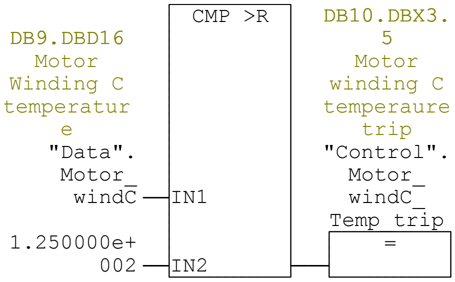
Network: 8



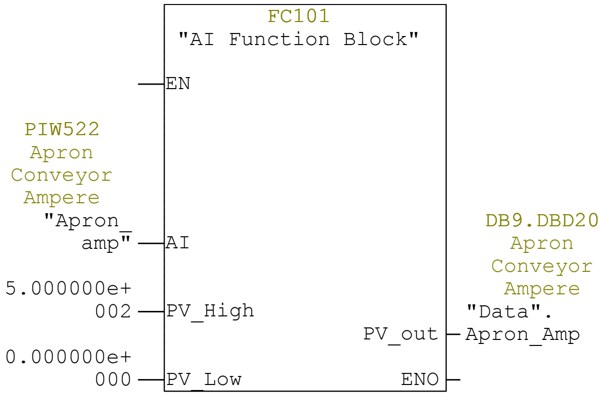
Network: 9 Motor winding B temperaure trip



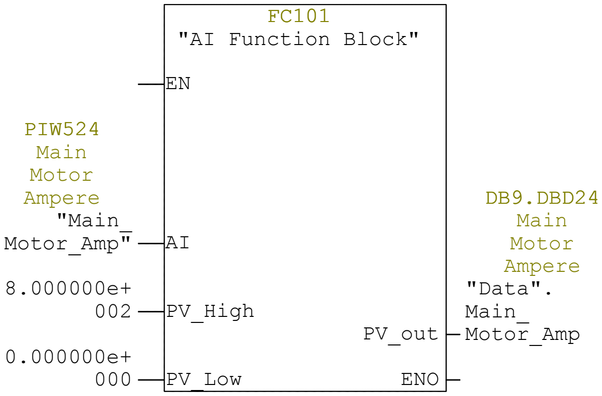
Network: 10 Motor winding C temperaure trip



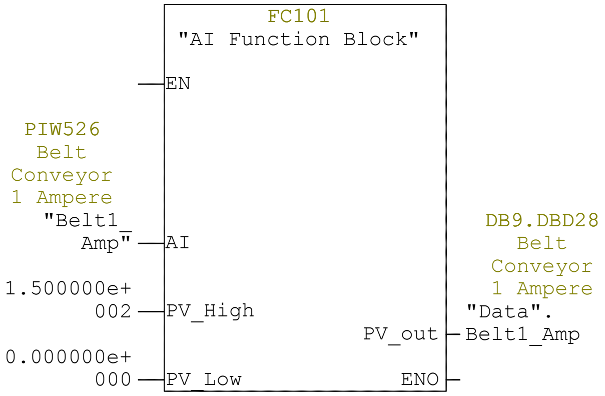
Network: 11



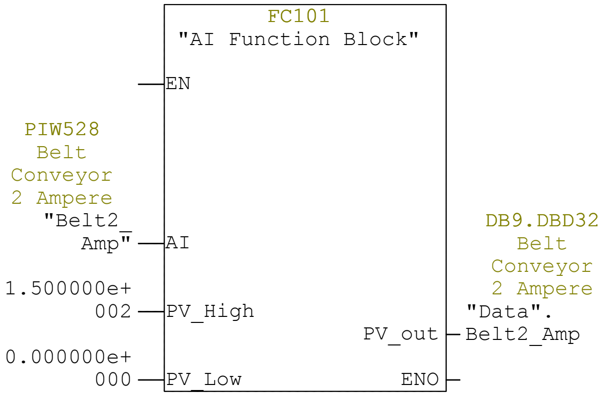
Network: 12



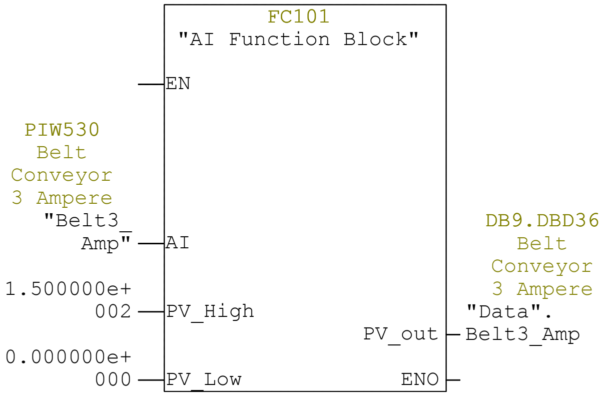
Network: 13



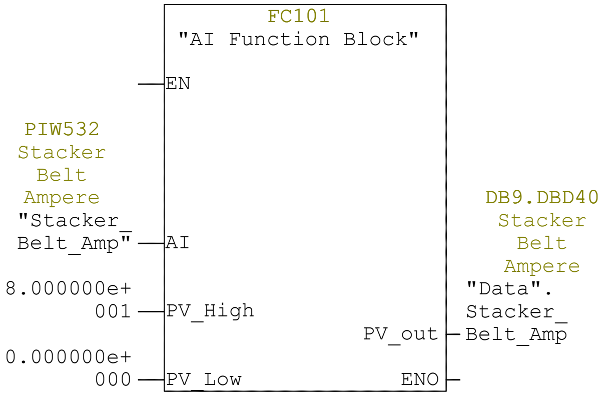
Network: 14



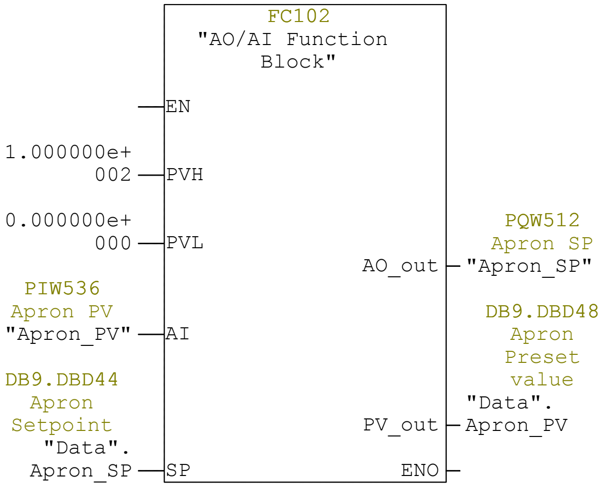
Network: 15



Network: 16



Network: 17



FC4 - <offline>

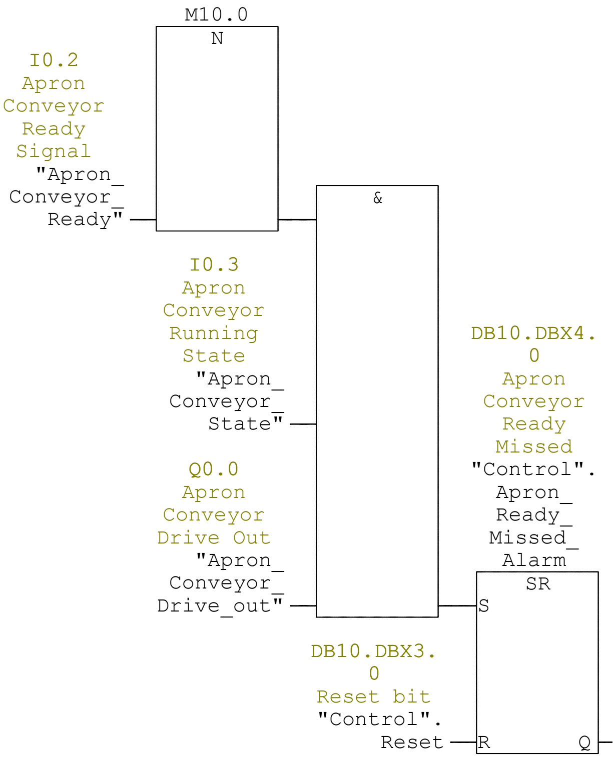
"Alarm"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/01/2022 02:45:20 AM
Interface: 04/30/2022 03:51:12 PM
Lengths (block/logic/data): 00598 00478 00000

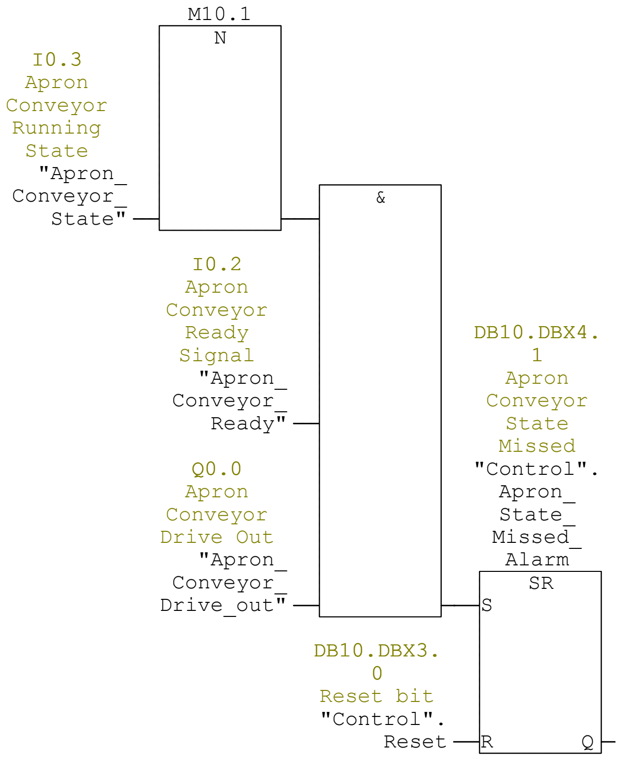
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC4

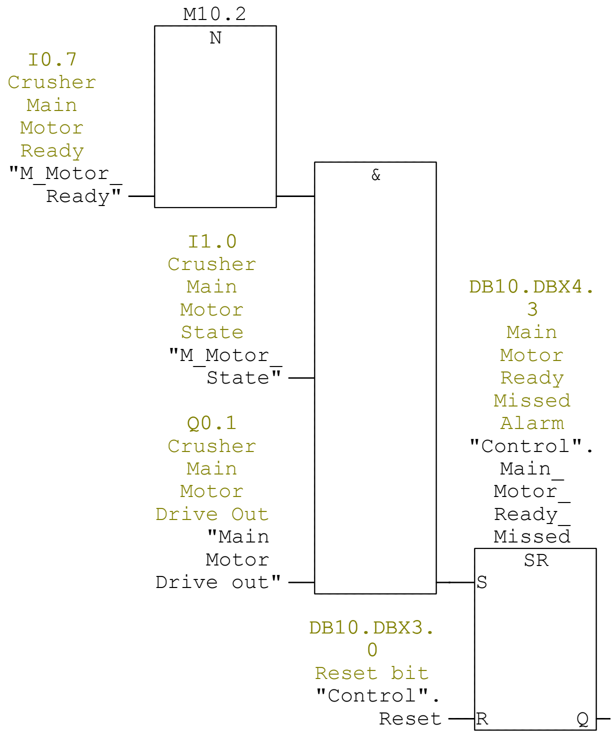
Network: 1 Apron Conveyor Ready Missed



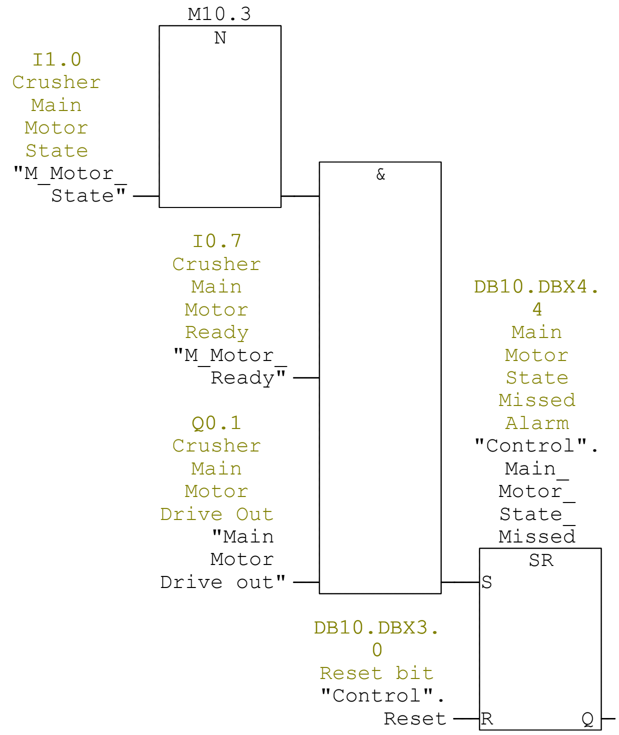
Network: 2 Apron Conveyor State Missed



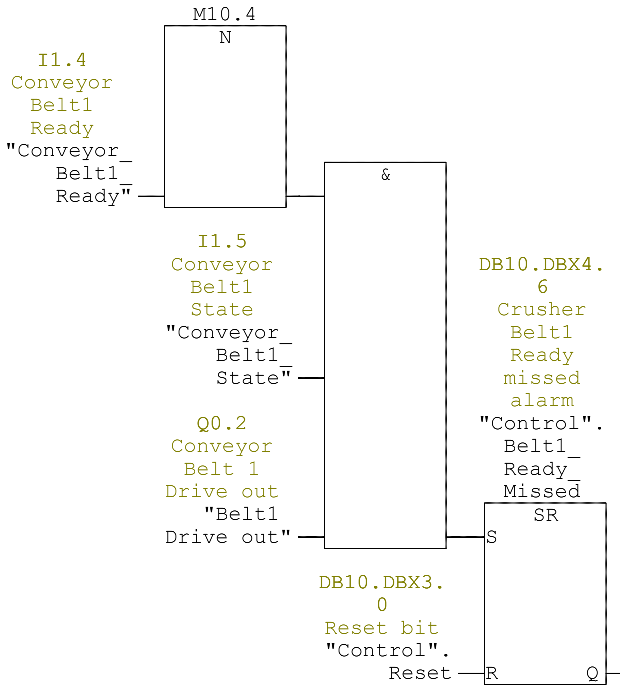
Network: 3 Main Motor Ready Missed Alarm



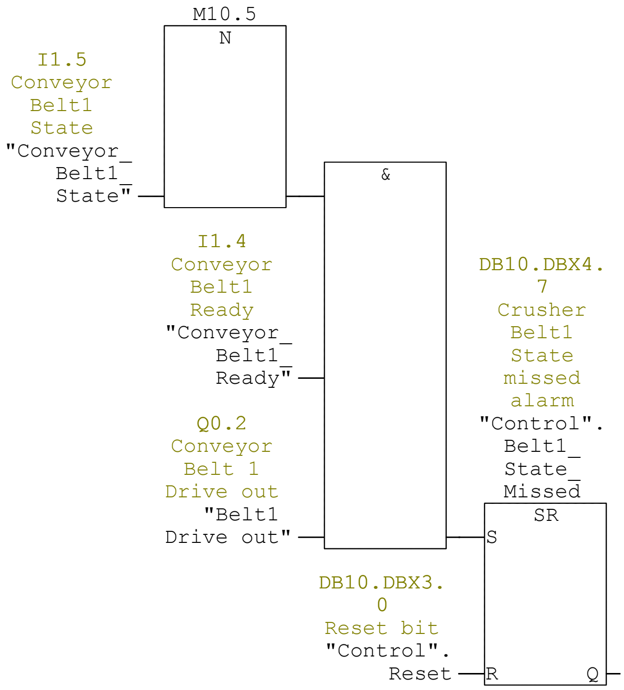
Network: 4 Main Motor State Missed Alarm



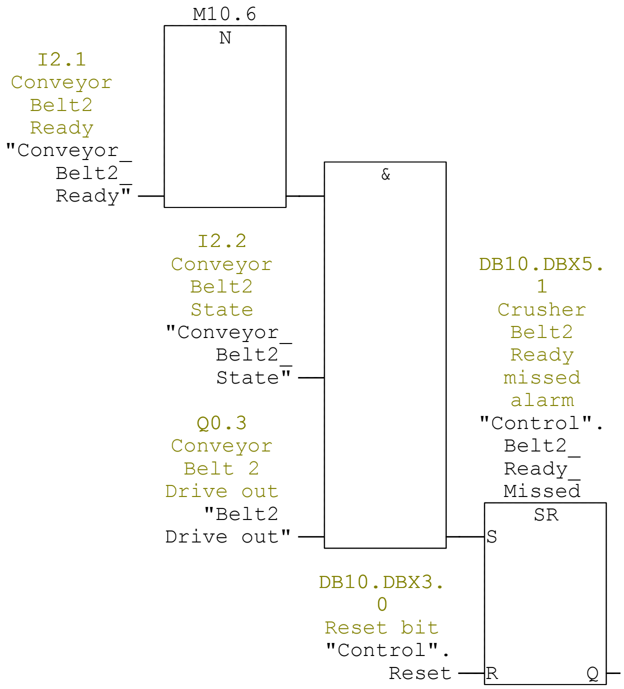
Network: 5 Crusher Belt1 Ready missed alarm



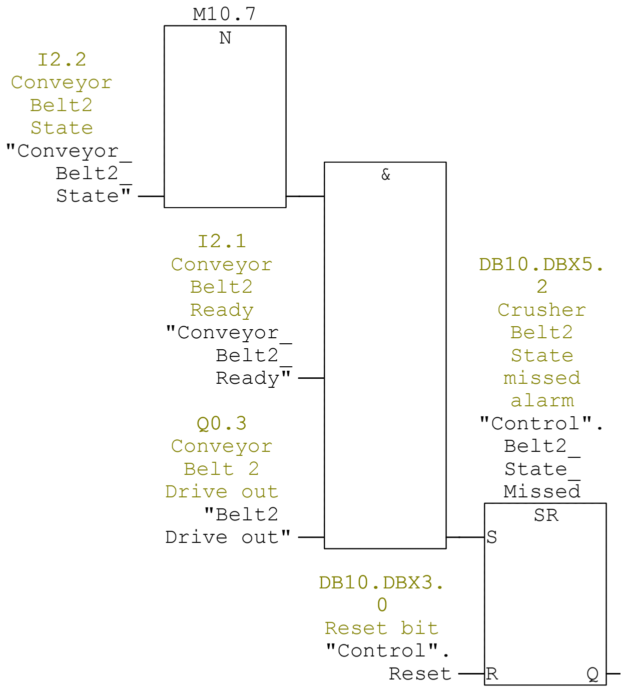
Network: 6 Crusher Belt1 State missed alarm



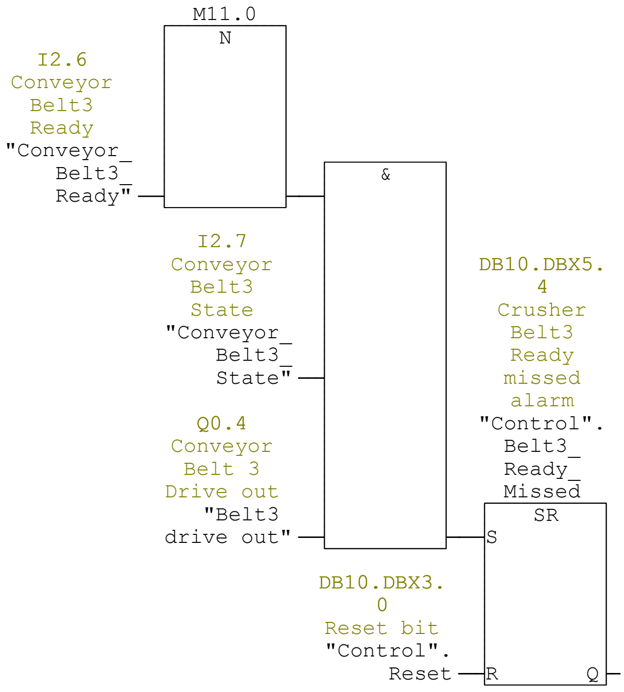
Network: 7 Crusher Belt2 Ready missed alarm



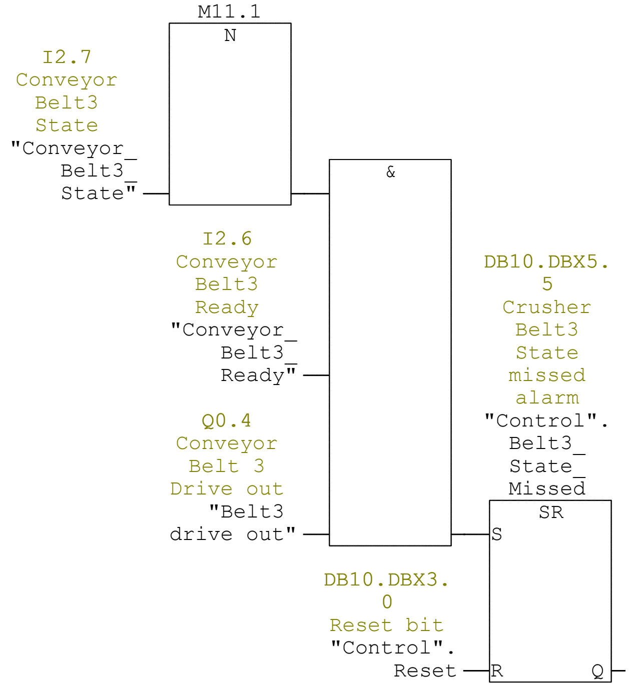
Network: 8 Crusher Belt2 State missed alarm



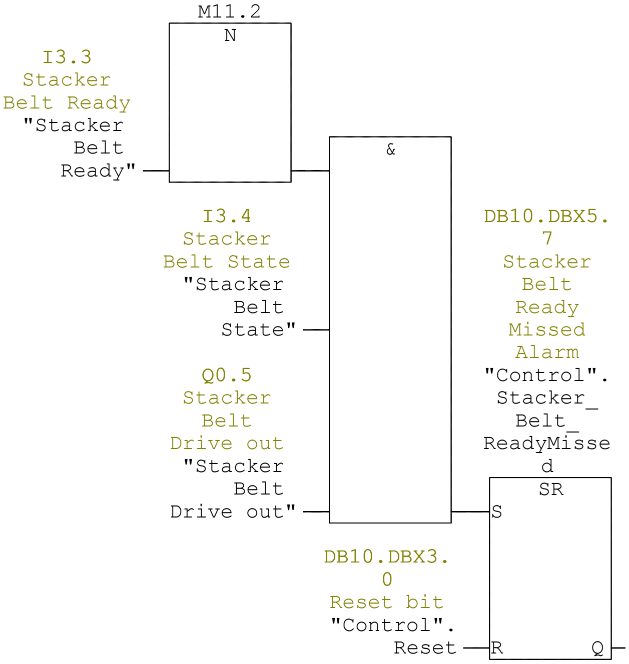
Network: 9 Crusher Belt3 Ready missed alarm



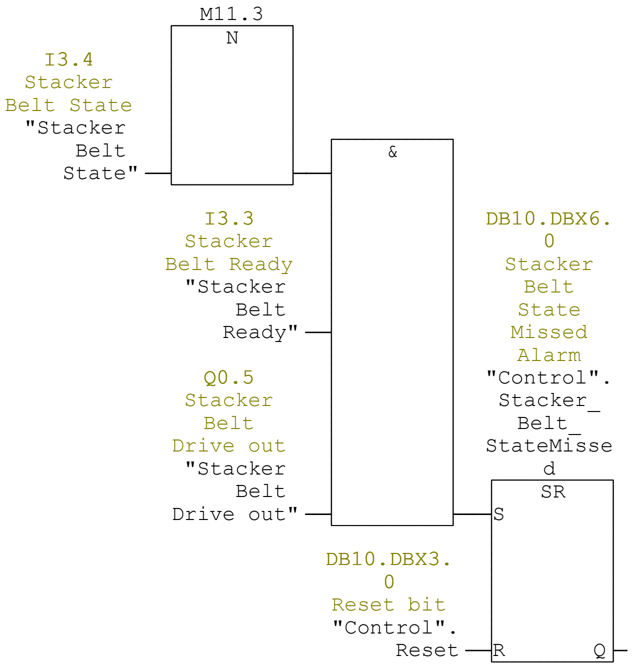
Network: 10 Crusher Belt3 State missed alarm



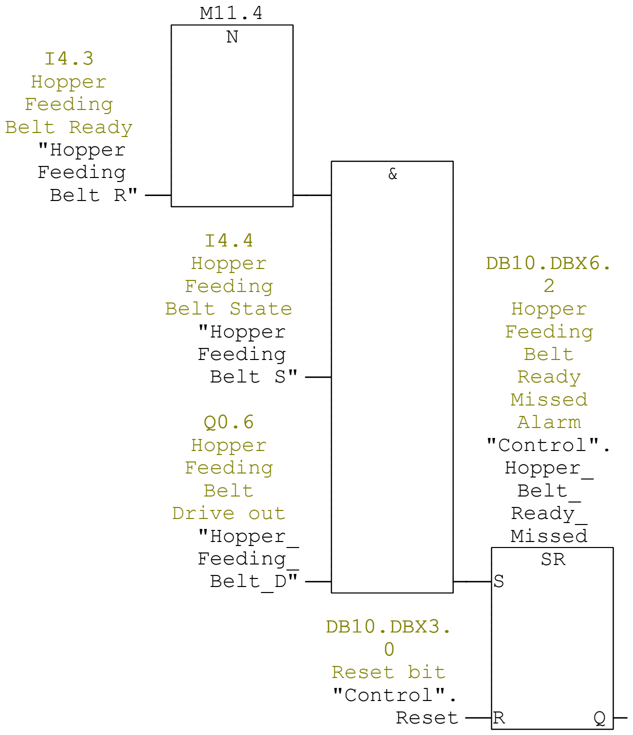
Network: 11 Stacker Belt Ready Missed Alarm



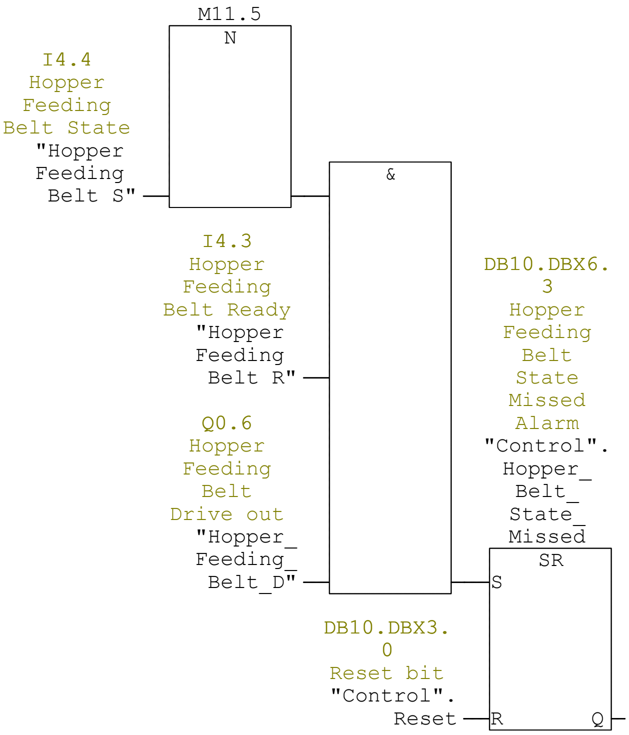
Network: 12 Stacker Belt State Missed Alarm



Network: 13 Hopper Feeding Belt Ready Missed Alarm



Network: 14 Hopper Feeding Belt State Missed Alarm



Network: 15 Apron Electrical Cabinet Alarm

```
A     "Apron_Alarm"                    I0.6            -- Apron Conveyor Alarm
=     "Control".Apron_Electrical_Alarm  DB10.DBX4.2    -- Apron Electrical Cabinet Alarm
A     "M_Motor_Alarm"                  I1.3            -- Crusher Main Motor Alarm
=     "Control".Main_Motor_Elec_Fault  DB10.DBX4.5     -- Main Motor Electrical CAbinet Alarm
A     "C_Belt1_Alarm"                  I2.0            -- Conveyor Belt1 Alarm
=     "Control".Belt1_Electrcal_Alarm  DB10.DBX5.0     -- Crusher Belt1 Electrical Cabinet Fault
A     "C_Belt2_Alarm"                  I2.5            -- Conveyor Belt2 Alarm
=     "Control".Belt2_Electrcal_Fault  DB10.DBX5.3     -- Crusher Belt2 Electrical Cabinet Fault
A     "C_Belt3_Akarm"                  I3.2            -- Conveyor Belt3 Alarm
=     "Control".Belt3_Electrcal_Fault  DB10.DBX5.6     -- Crusher Belt3 Electrical Cabinet Fault
A     "Stacker_Belt_Alarm"             I3.7            -- Stacker Belt Alarm
=     "Control".Stacker_Belt_Elec_Alarm DB10.DBX6.1    -- Stacker Belt Electrical Cabinet Fault
A     "Hopper Feeding Belt A"          I4.5            -- Hopper Feeding Belt Alarm
=     "Control".Hopper_Belt_Elec_Fault DB10.DBX6.4     -- Hopper Feeding Belt Electrical Cabinet Fault
```

FC1 - <offline>

"Data Sharing"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 06/08/2022 10:41:27 PM
Interface: 01/28/2022 03:25:59 PM
Lengths (block/logic/data): 00222 00130 00000

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC1

Network: 1   Silo Feeding Belt Ready Signal

```
A     "Receive From FCS01".Hopper_Feeding_Belt_R  DB2.DBX0.1         -- Hopper Feeding Belt Ready
=     "Control".Hoper_Feeding_Belt_Ready          DB10.DBX6.6        -- Hopper Feeding Belt Ready Signal
A     "Receive From FCS01".Hopper_Feeding_Belt_S  DB2.DBX0.2         -- Hopper Feeding Belt State
=     "Control".Hoper_Feeding_Belt_State          DB10.DBX6.7        -- Hopper Feeding Belt State Signal
A     "Receive From FCS01".Hopper_Feeding_Belt_A  DB2.DBX0.3         -- Hopper Feeding Belt Alarm
=     "Control".Hoper_Feeding_Belt_Alarm          DB10.DBX7.0        -- Hopper Feeding Belt Alarm Signal
A     "Control".Hoper_Feeding_Belt_Start          DB10.DBX6.4        -- Hopper Feeding Belt Start cmd
=     "Send to FCS01".Hopper_feeding_belt_Star    DB1.DBX0.1         -- Hopper feeding belt start command
A     "Control".Hoper_Feeding_Belt_Stop           DB10.DBX6.5        -- Hopper Feeding Belt Stop Cmd
=     "Send to FCS01".hopper_feeding_belt_stop    DB1.DBX0.2         -- Hopper feeding belt stop command
A     "Receive From FCS01".Reclaimer_R            DB2.DBX0.4         -- Reclaimer Ready
=     "Control".Reclaimer_Ready                   DB10.DBX12.4       -- Reclaimer Ready Signal
A     "Receive From FCS01".Reclaimer_S            DB2.DBX0.5         -- Reclaimer State
=     "Control".Reclaimer_State                   DB10.DBX12.5       -- Reclaimer State Signal
A     "Receive From FCS01".Reclaimer_A            DB2.DBX0.6         -- Reclaimer Alarm
=     "Control".Reclaimer_Alarm                   DB10.DBX12.6       -- Reclaimer Alarm Signal
A     "Control".Reclaimer_Start                   DB10.DBX12.2       -- Reclaimer Start Cmd
=     "Send to FCS01".Reclaimer_Start             DB1.DBX0.3         -- Reclaimer Start Command
A     "Control".Reclaimer_Stop                    DB10.DBX12.3       -- Reclaimer Stop Cmd
=     "Send to FCS01".Reclaimer_Stop              DB1.DBX0.4         -- Reclaimer Stop Command
A     "BH_Fan_D"                                  Q1.3               -- Baghouse fan drive out
=     "Send to FCS03".BH_Fan_Running_Cmd          DB3.DBX0.1         -- BH Fan running Command send to FCS03
```

FC2 - <offline>

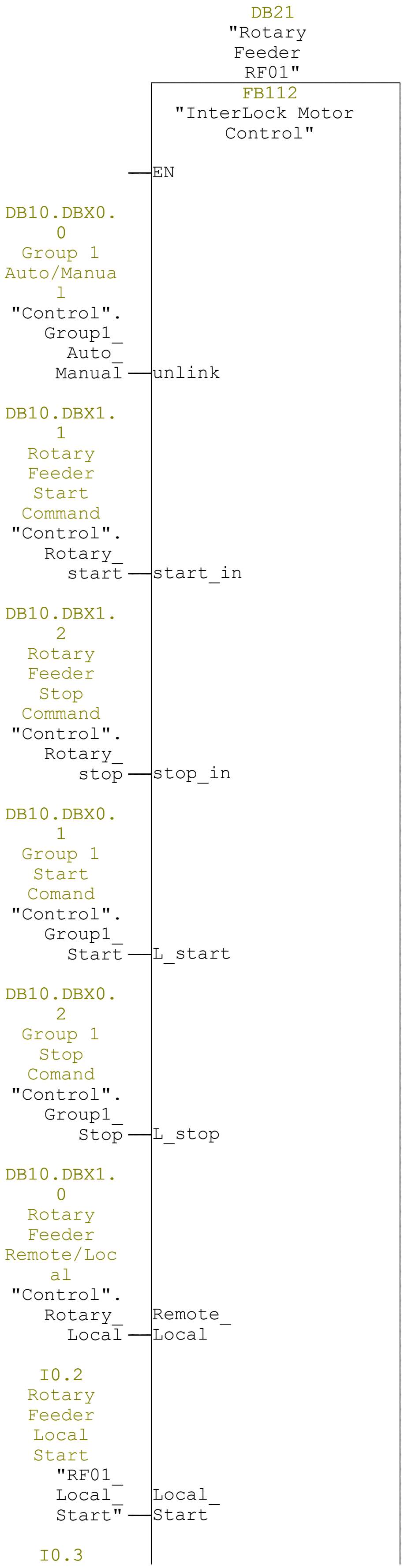
"Motors"

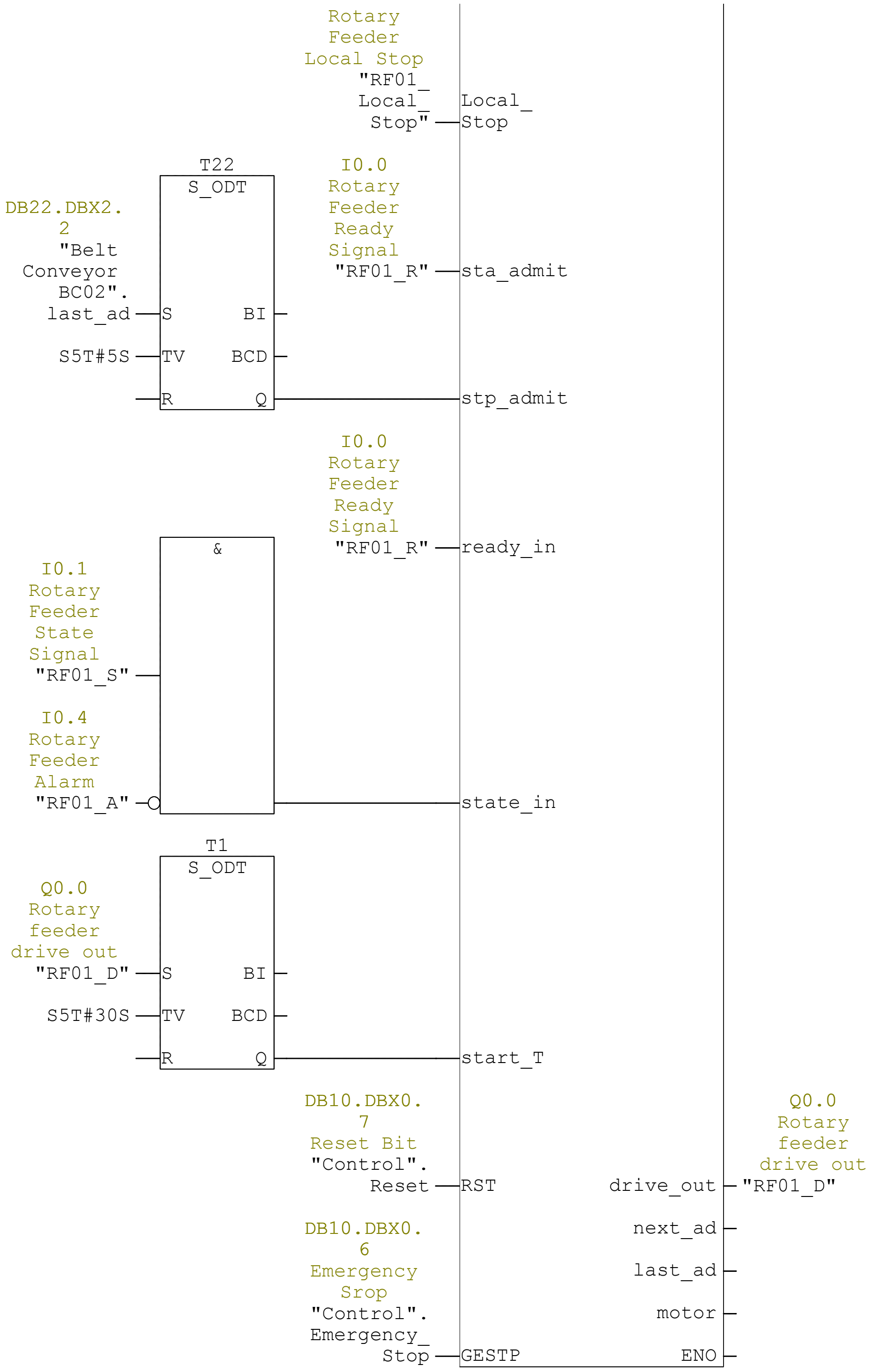
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 04/08/2022 04:24:32 PM
Interface: 03/23/2022 03:10:53 AM
Lengths (block/logic/data): 04730 04606 00008

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC2

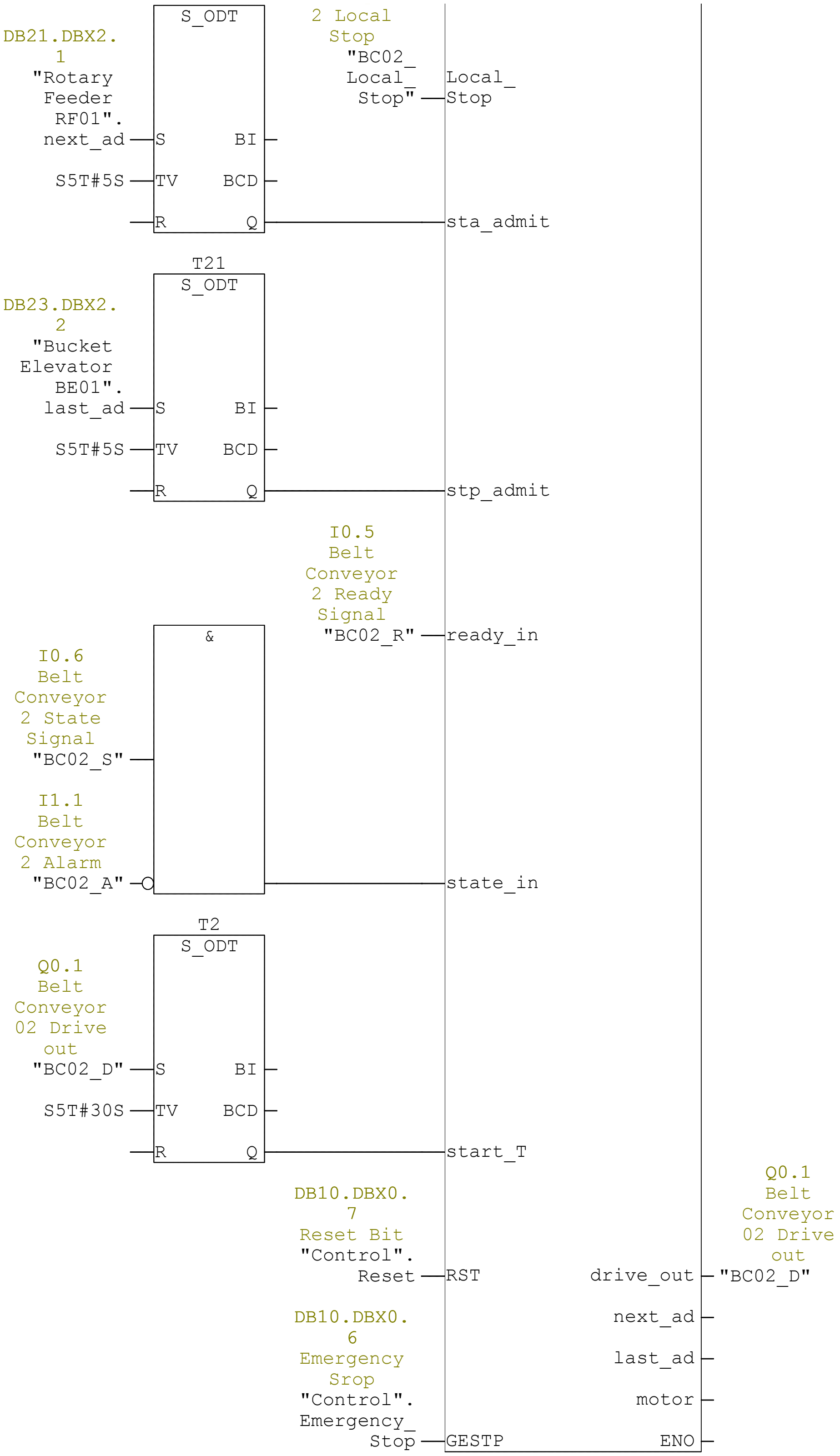
Network: 1


DB21
"Rotary Feeder RF01"
FB112
"InterLock Motor Control"
EN
DB10.DBX0.0
Group 1 Auto/Manual
"Control".Group1_Auto_Manual
unlink
DB10.DBX1.1
Rotary Feeder Start Command
"Control".Rotary_start
start_in
DB10.DBX1.2
Rotary Feeder Stop Command
"Control".Rotary_stop
stop_in
DB10.DBX0.1
Group 1 Start Comand
"Control".Group1_Start
L_start
DB10.DBX0.2
Group 1 Stop Comand
"Control".Group1_Stop
L_stop
DB10.DBX1.0
Rotary Feeder Remote/Local
"Control".Rotary_Local
Remote_Local
I0.2
Rotary Feeder Local Start
"RF01_Local_Start"
Local_Start
I0.3

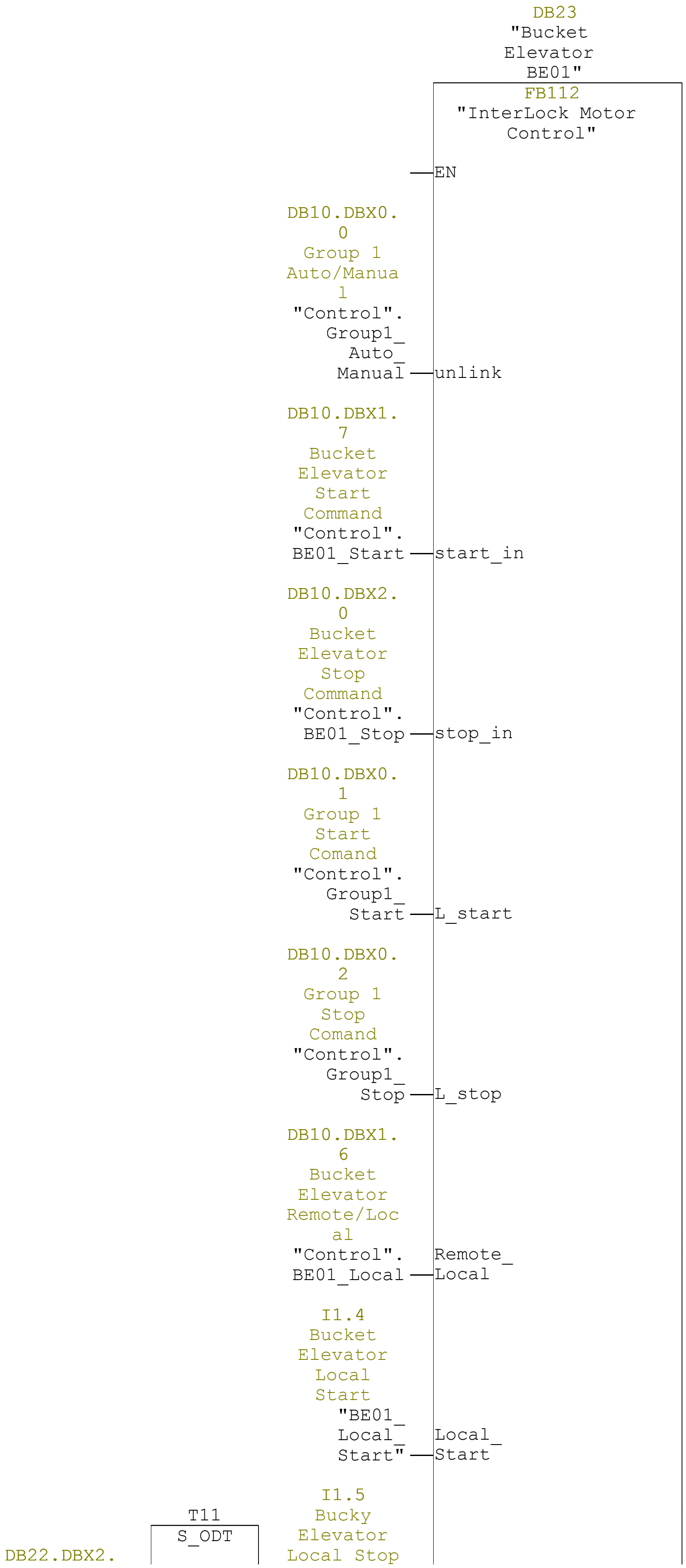

Rotary
Feeder
Local Stop
"RF01_
Local_
Stop"
Local_
Stop
T22
S_ODT
DB22.DBX2.
2
"Belt
Conveyor
BC02".
last_ad
S
BI
S5T#5S
TV
BCD
R
Q
I0.0
Rotary
Feeder
Ready
Signal
"RF01_R"
sta_admit
stp_admit
I0.0
Rotary
Feeder
Ready
Signal
"RF01_R"
ready_in
&
I0.1
Rotary
Feeder
State
Signal
"RF01_S"
I0.4
Rotary
Feeder
Alarm
"RF01_A"
state_in
T1
S_ODT
Q0.0
Rotary
feeder
drive out
"RF01_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q0.0
Rotary
feeder
drive out
"RF01_D"
next_ad
last_ad
motor
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
ENO

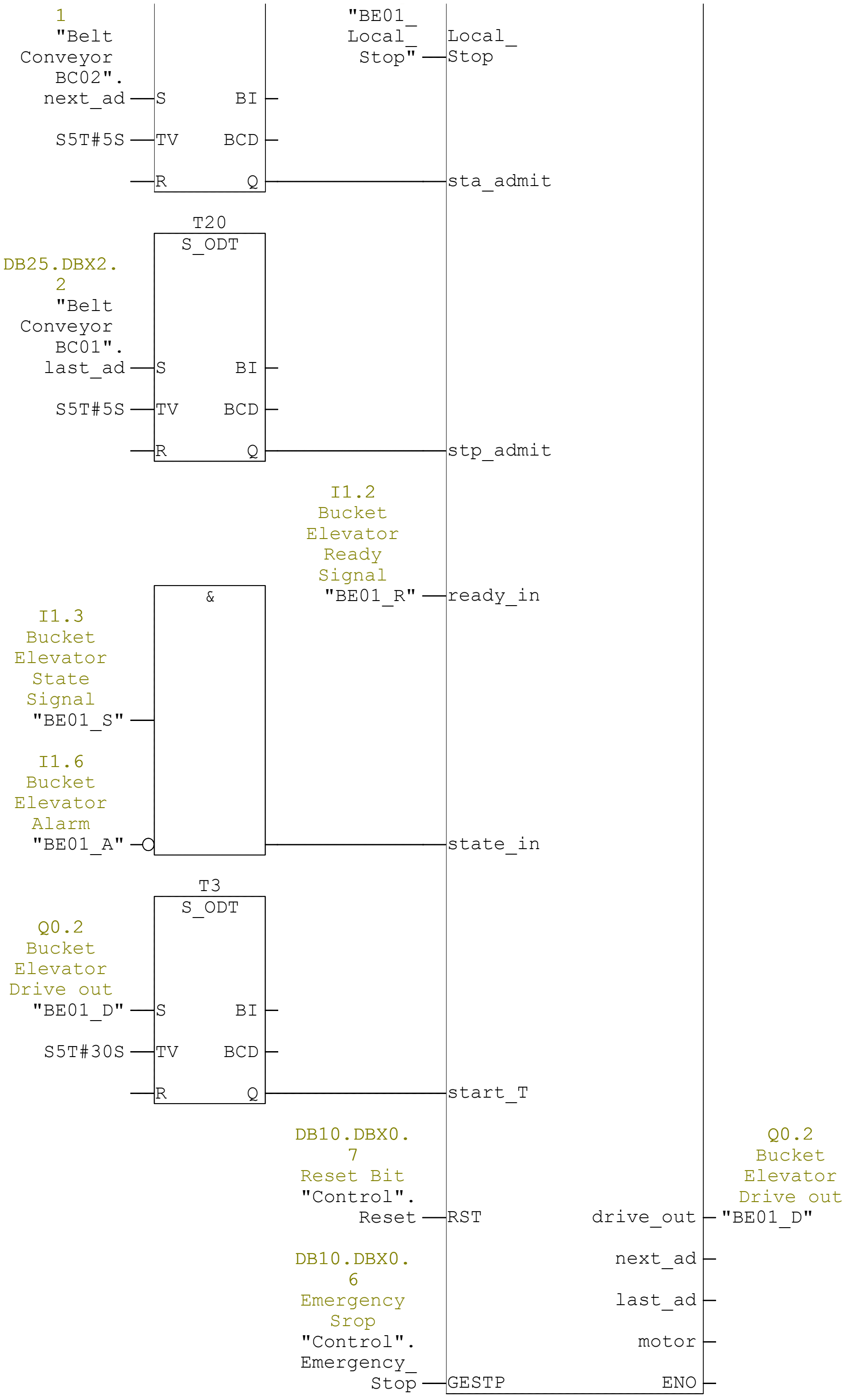
Network: 2

```
                                    DB22
                                   "Belt
                                  Conveyor
                                   BC02"
                           +-------FB112--------+
                           | "InterLock Motor   |
                           |      Control"      |
                           |                    |
                        ---|EN                  |
          DB10.DBX0.       |                    |
               0           |                    |
            Group 1        |                    |
          Auto/Manua       |                    |
               l           |                    |
           "Control".      |                    |
              Group1_      |                    |
                Auto_      |                    |
               Manual -----|unlink              |
                           |                    |
          DB10.DBX1.       |                    |
               4           |                    |
             Belt          |                    |
           Conveyor        |                    |
            2 Start        |                    |
            Command        |                    |
           "Control".      |                    |
           BC02_start -----|start_in            |
                           |                    |
          DB10.DBX1.       |                    |
               5           |                    |
             Belt          |                    |
           Conveyor        |                    |
            2 Stop         |                    |
            Command        |                    |
           "Control".      |                    |
            BC02_stop -----|stop_in             |
                           |                    |
          DB10.DBX0.       |                    |
               1           |                    |
            Group 1        |                    |
             Start         |                    |
            Comand         |                    |
           "Control".      |                    |
              Group1_      |                    |
                Start -----|L_start             |
                           |                    |
          DB10.DBX0.       |                    |
               2           |                    |
            Group 1        |                    |
             Stop          |                    |
            Comand         |                    |
           "Control".      |                    |
              Group1_      |                    |
                 Stop -----|L_stop              |
                           |                    |
          DB10.DBX1.       |                    |
               3           |                    |
             Belt          |                    |
           Conveyor        |                    |
               2           |                    |
          Remote/Loc       |                    |
              al           |                    |
           "Control".      |Remote_             |
           BC02_Local -----|Local               |
                           |                    |
             I0.7          |                    |
             Belt          |                    |
           Conveyor        |                    |
            2 Local        |                    |
             Start         |                    |
               "BC02_      |                    |
               Local_      |Local_              |
               Start" -----|Start               |
                           |                    |
             I1.0          |                    |
             Belt          |                    |
  T10      Conveyor        |                    |
```


S_ODT
DB21.DBX2.1
"Rotary Feeder RF01".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
2 Local Stop
"BC02_Local_Stop"
Local_Stop
sta_admit
T21
S_ODT
DB23.DBX2.2
"Bucket Elevator BE01".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I0.5
Belt Conveyor 2 Ready Signal
"BC02_R"
ready_in
&
I0.6
Belt Conveyor 2 State Signal
"BC02_S"
I1.1
Belt Conveyor 2 Alarm
"BC02_A"
state_in
T2
S_ODT
Q0.1
Belt Conveyor 02 Drive out
"BC02_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.7
Reset Bit
"Control".Reset
RST
drive_out
Q0.1
Belt Conveyor 02 Drive out
"BC02_D"
next_ad
DB10.DBX0.6
Emergency Srop
last_ad
"Control".Emergency_Stop
motor
GESTP
ENO

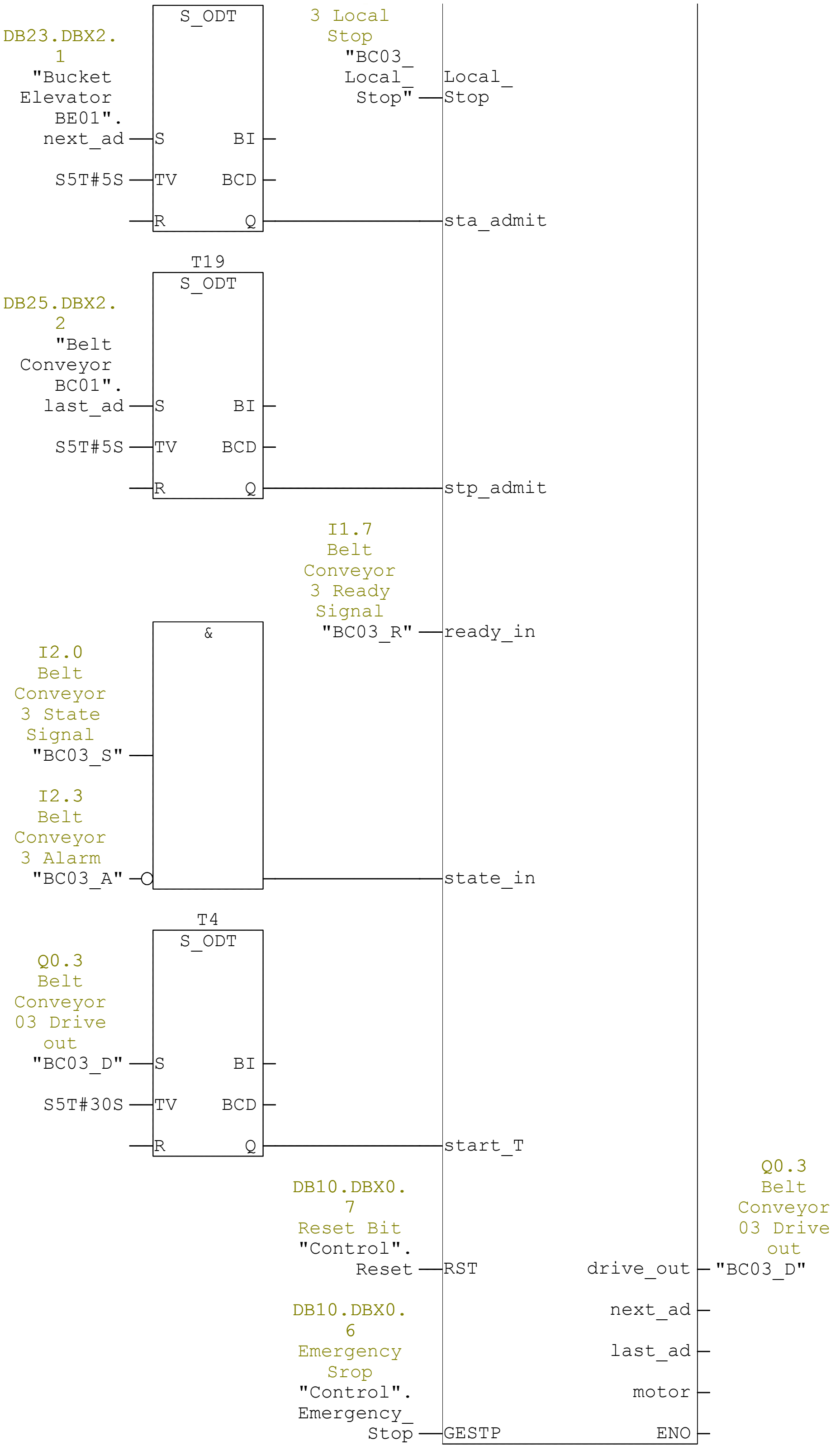
Network: 3


DB23
"Bucket Elevator BE01"
FB112
"InterLock Motor Control"
EN
DB10.DBX0.0
Group 1 Auto/Manual
"Control".Group1_Auto_Manual
unlink
DB10.DBX1.7
Bucket Elevator Start Command
"Control".BE01_Start
start_in
DB10.DBX2.0
Bucket Elevator Stop Command
"Control".BE01_Stop
stop_in
DB10.DBX0.1
Group 1 Start Comand
"Control".Group1_Start
L_start
DB10.DBX0.2
Group 1 Stop Comand
"Control".Group1_Stop
L_stop
DB10.DBX1.6
Bucket Elevator Remote/Local
"Control".BE01_Local
Remote_Local
I1.4
Bucket Elevator Local Start
"BE01_Local_Start"
Local_Start
T11
S_ODT
DB22.DBX2.
I1.5
Bucky Elevator Local Stop

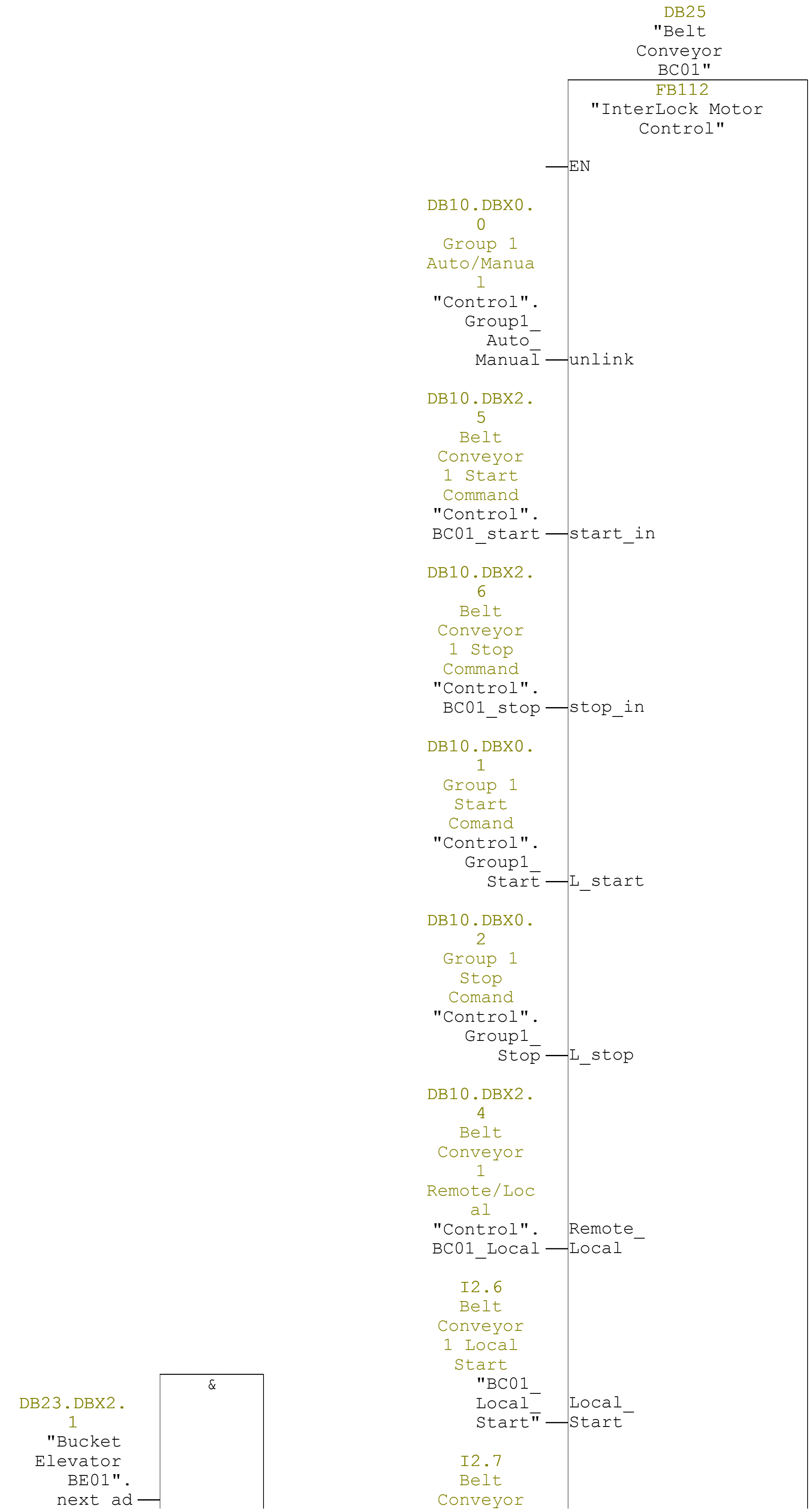

1
"Belt
Conveyor
BC02".
next_ad
S
BI
S5T#5S
TV
BCD
R
Q
"BE01_
Local_
Stop"
Local_
Stop
sta_admit
T20
S_ODT
DB25.DBX2.
2
"Belt
Conveyor
BC01".
last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I1.2
Bucket
Elevator
Ready
Signal
"BE01_R"
ready_in
&
I1.3
Bucket
Elevator
State
Signal
"BE01_S"
I1.6
Bucket
Elevator
Alarm
"BE01_A"
state_in
T3
S_ODT
Q0.2
Bucket
Elevator
Drive out
"BE01_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q0.2
Bucket
Elevator
Drive out
"BE01_D"
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
next_ad
last_ad
motor
ENO

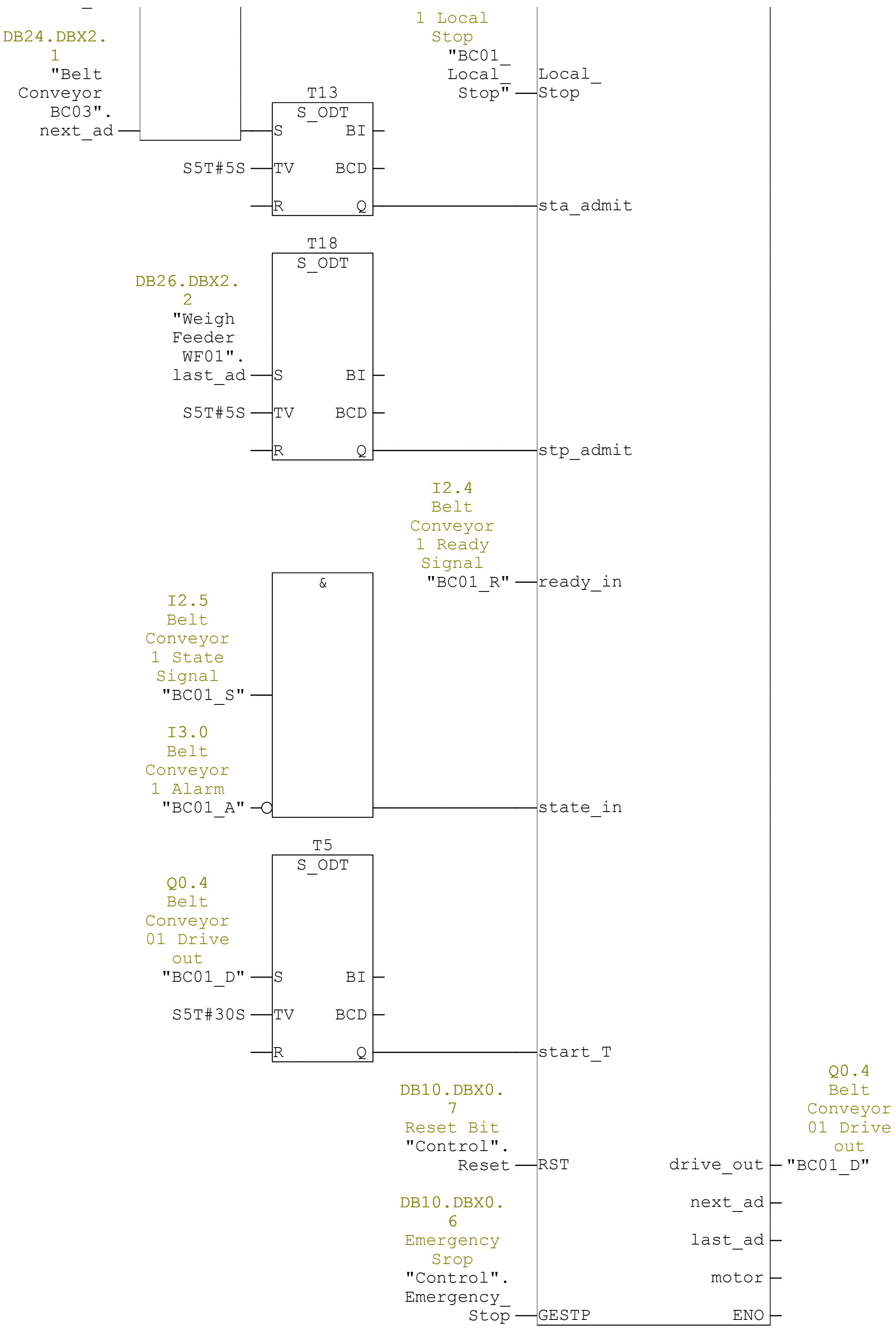
Network: 4

```
                                    DB24
                                   "Belt
                                  Conveyor
                                   BC03"
                              +---------------+
                              |    FB112      |
                              |"InterLock Motor
                              |   Control"    |
                              |               |
                           ---|EN             |
                              |               |
              DB10.DBX0.      |               |
                  0           |               |
               Group 1        |               |
             Auto/Manua       |               |
                  l           |               |
             "Control".       |               |
                Group1_       |               |
                  Auto_       |               |
                 Manual ------|unlink         |
                              |               |
              DB10.DBX2.      |               |
                  2           |               |
                 Belt         |               |
               Conveyor       |               |
               3 Start        |               |
               Command        |               |
             "Control".       |               |
             BC03_Start ------|start_in       |
                              |               |
              DB10.DBX2.      |               |
                  3           |               |
                 Belt         |               |
               Conveyor       |               |
                3 Stop        |               |
               Command        |               |
             "Control".       |               |
              BC03_Stop ------|stop_in        |
                              |               |
              DB10.DBX0.      |               |
                  1           |               |
               Group 1        |               |
                Start         |               |
                Comand        |               |
             "Control".       |               |
                Group1_       |               |
                  Start ------|L_start        |
                              |               |
              DB10.DBX0.      |               |
                  2           |               |
               Group 1        |               |
                 Stop         |               |
                Comand        |               |
             "Control".       |               |
                Group1_       |               |
                   Stop ------|L_stop         |
                              |               |
              DB10.DBX2.      |               |
                  1           |               |
                 Belt         |               |
               Conveyor       |               |
                  3           |               |
             Remote/Loc       |               |
                  al          |               |
             "Control".       |Remote_        |
             BC03_Local ------|Local          |
                              |               |
                 I2.1         |               |
                 Belt         |               |
               Conveyor       |               |
               3 Local        |               |
                Start         |               |
                 "BC03_       |               |
                 Local_       |Local_         |
                 Start" ------|Start          |
                              |               |
                 I2.2         |               |
                 Belt         |               |
    T12        Conveyor       |               |
```

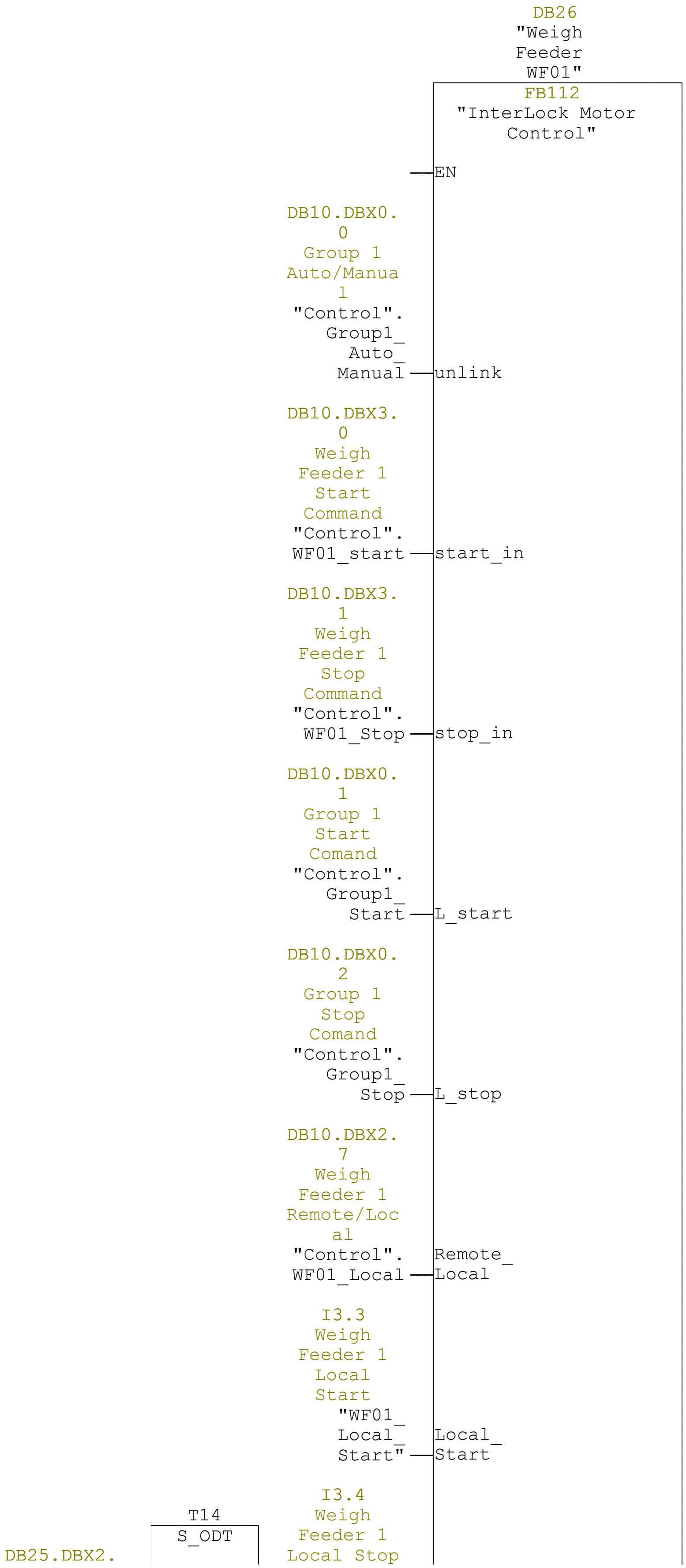

S_ODT
DB23.DBX2.1
"Bucket Elevator BE01".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
3 Local Stop
"BC03_Local_Stop"
Local_Stop
sta_admit
T19
S_ODT
DB25.DBX2.2
"Belt Conveyor BC01".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I1.7
Belt Conveyor 3 Ready Signal
"BC03_R"
ready_in
&
I2.0
Belt Conveyor 3 State Signal
"BC03_S"
I2.3
Belt Conveyor 3 Alarm
"BC03_A"
state_in
T4
S_ODT
Q0.3
Belt Conveyor 03 Drive out
"BC03_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.7
Reset Bit
"Control".Reset
RST
drive_out
Q0.3
Belt Conveyor 03 Drive out
"BC03_D"
next_ad
DB10.DBX0.6
Emergency Srop
last_ad
motor
"Control".Emergency_Stop
GESTP
ENO

Network: 5


DB25
"Belt
Conveyor
BC01"
FB112
"InterLock Motor
Control"
EN
DB10.DBX0.
0
Group 1
Auto/Manua
l
"Control".
Group1_
Auto_
Manual
unlink
DB10.DBX2.
5
Belt
Conveyor
1 Start
Command
"Control".
BC01_start
start_in
DB10.DBX2.
6
Belt
Conveyor
1 Stop
Command
"Control".
BC01_stop
stop_in
DB10.DBX0.
1
Group 1
Start
Comand
"Control".
Group1_
Start
L_start
DB10.DBX0.
2
Group 1
Stop
Comand
"Control".
Group1_
Stop
L_stop
DB10.DBX2.
4
Belt
Conveyor
1
Remote/Loc
al
"Control".
BC01_Local
Remote_
Local
I2.6
Belt
Conveyor
1 Local
Start
"BC01_
Local_
Start"
Local_
Start
&
DB23.DBX2.
1
"Bucket
Elevator
BE01".
next ad
I2.7
Belt
Conveyor


_
DB24.DBX2.
1
"Belt
Conveyor
BC03".
next_ad
1 Local
Stop
"BC01_
Local_
Stop"
Local_
Stop
T13
S_ODT
S
BI
S5T#5S
TV
BCD
R
Q
sta_admit
T18
S_ODT
DB26.DBX2.
2
"Weigh
Feeder
WF01".
last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I2.4
Belt
Conveyor
1 Ready
Signal
"BC01_R"
ready_in
&
I2.5
Belt
Conveyor
1 State
Signal
"BC01_S"
I3.0
Belt
Conveyor
1 Alarm
"BC01_A"
state_in
T5
S_ODT
Q0.4
Belt
Conveyor
01 Drive
out
"BC01_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q0.4
Belt
Conveyor
01 Drive
out
"BC01_D"
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
next_ad
last_ad
motor
ENO

Network: 6



DB26
"Weigh Feeder WF01"
FB112
"InterLock Motor Control"

EN

DB10.DBX0.0
Group 1 Auto/Manual
"Control".Group1_Auto_Manual — unlink

DB10.DBX3.0
Weigh Feeder 1 Start Command
"Control".WF01_start — start_in

DB10.DBX3.1
Weigh Feeder 1 Stop Command
"Control".WF01_Stop — stop_in

DB10.DBX0.1
Group 1 Start Comand
"Control".Group1_Start — L_start

DB10.DBX0.2
Group 1 Stop Comand
"Control".Group1_Stop — L_stop

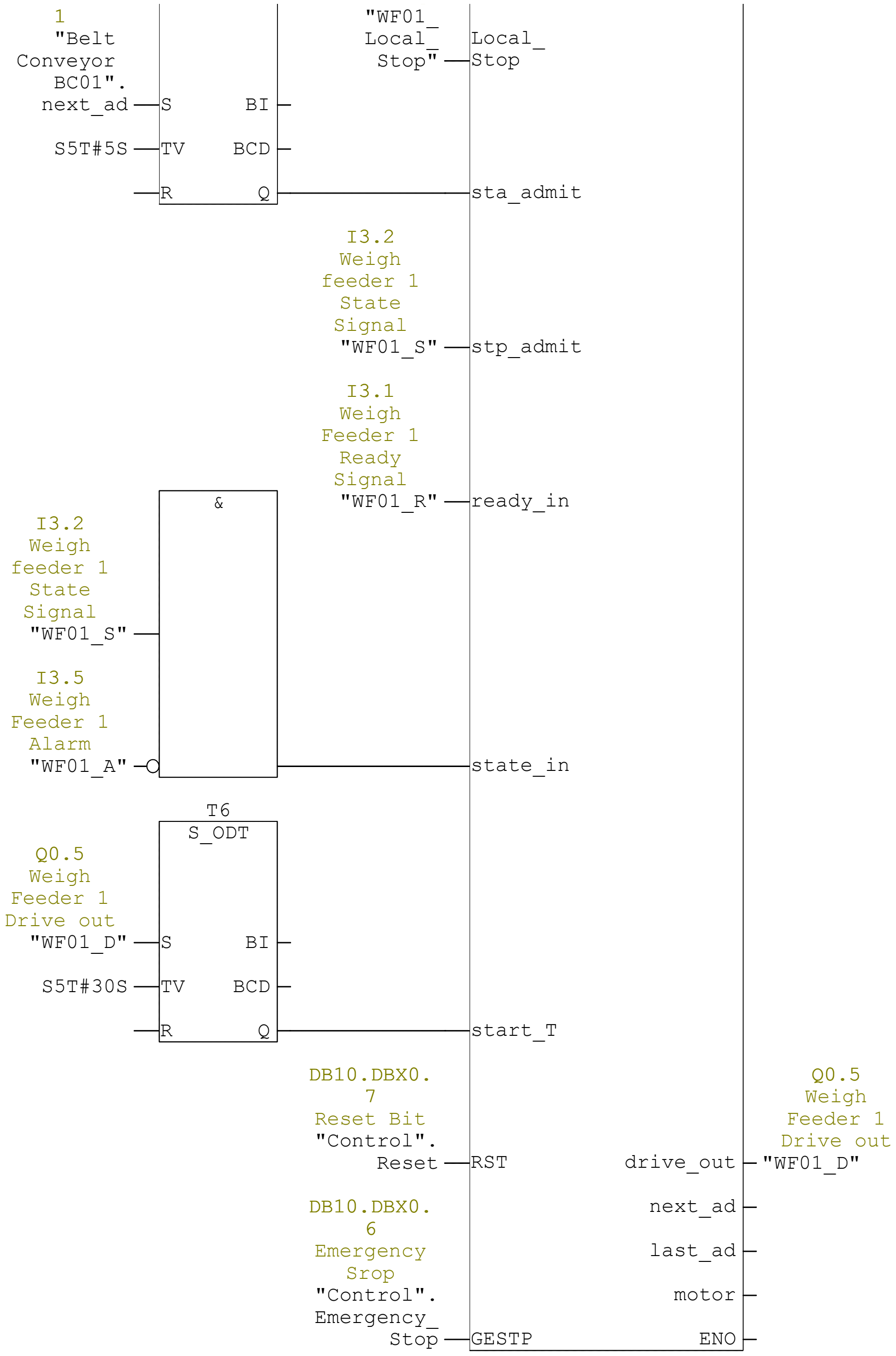
DB10.DBX2.7
Weigh Feeder 1 Remote/Local
"Control".WF01_Local — Remote_Local

I3.3
Weigh Feeder 1 Local Start
"WF01_Local_Start" — Local_Start

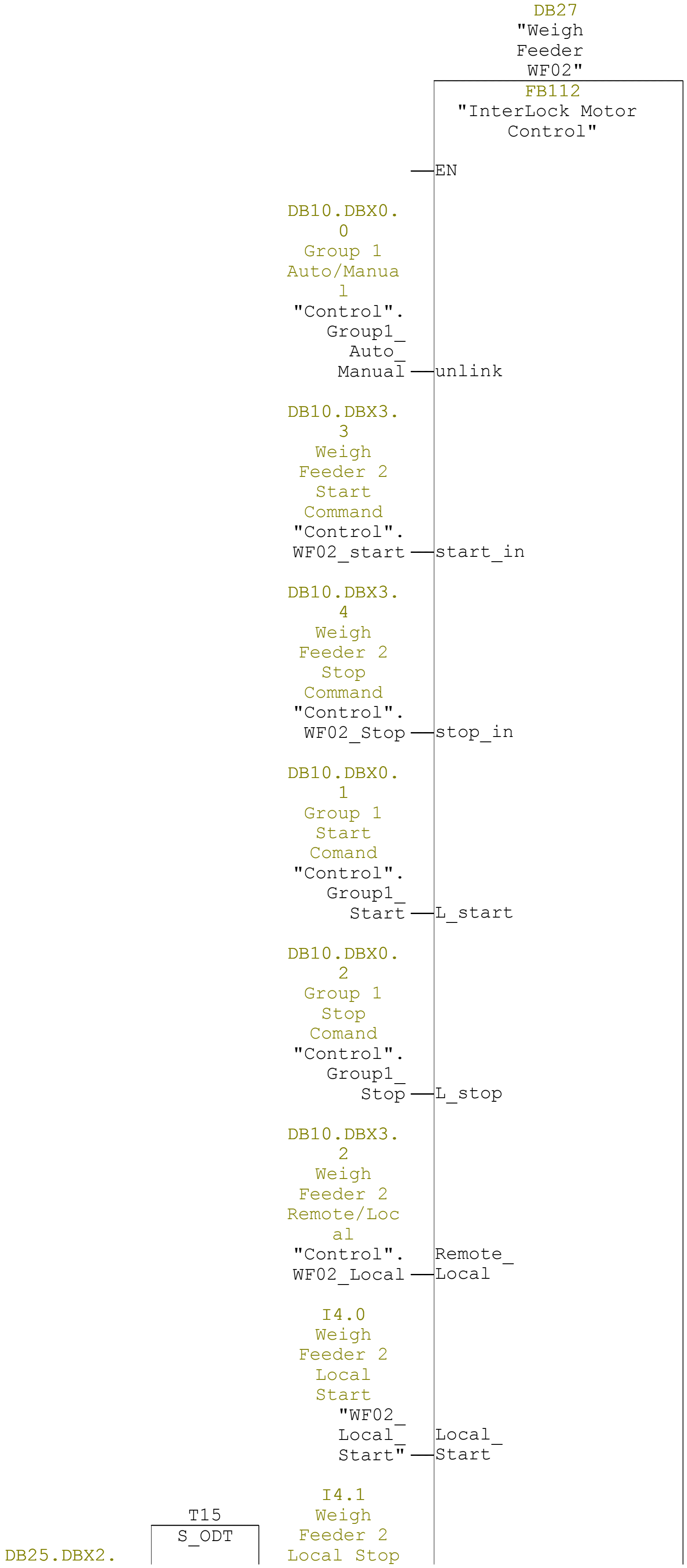
I3.4
Weigh Feeder 1 Local Stop

T14
S_ODT

DB25.DBX2.


1
"Belt
Conveyor
BC01".
next_ad
S
BI
S5T#5S
TV
BCD
R
Q
"WF01_
Local_
Stop"
Local_
Stop
sta_admit
I3.2
Weigh
feeder 1
State
Signal
"WF01_S"
stp_admit
I3.1
Weigh
Feeder 1
Ready
Signal
"WF01_R"
ready_in
&
I3.2
Weigh
feeder 1
State
Signal
"WF01_S"
I3.5
Weigh
Feeder 1
Alarm
"WF01_A"
state_in
T6
S_ODT
Q0.5
Weigh
Feeder 1
Drive out
"WF01_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q0.5
Weigh
Feeder 1
Drive out
"WF01_D"
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
next_ad
last_ad
motor
ENO

Network: 7



DB27
"Weigh Feeder WF02"
FB112
"InterLock Motor Control"

EN

DB10.DBX0.0
Group 1 Auto/Manual
"Control".Group1_Auto_Manual — unlink

DB10.DBX3.3
Weigh Feeder 2 Start Command
"Control".WF02_start — start_in

DB10.DBX3.4
Weigh Feeder 2 Stop Command
"Control".WF02_Stop — stop_in

DB10.DBX0.1
Group 1 Start Comand
"Control".Group1_Start — L_start

DB10.DBX0.2
Group 1 Stop Comand
"Control".Group1_Stop — L_stop

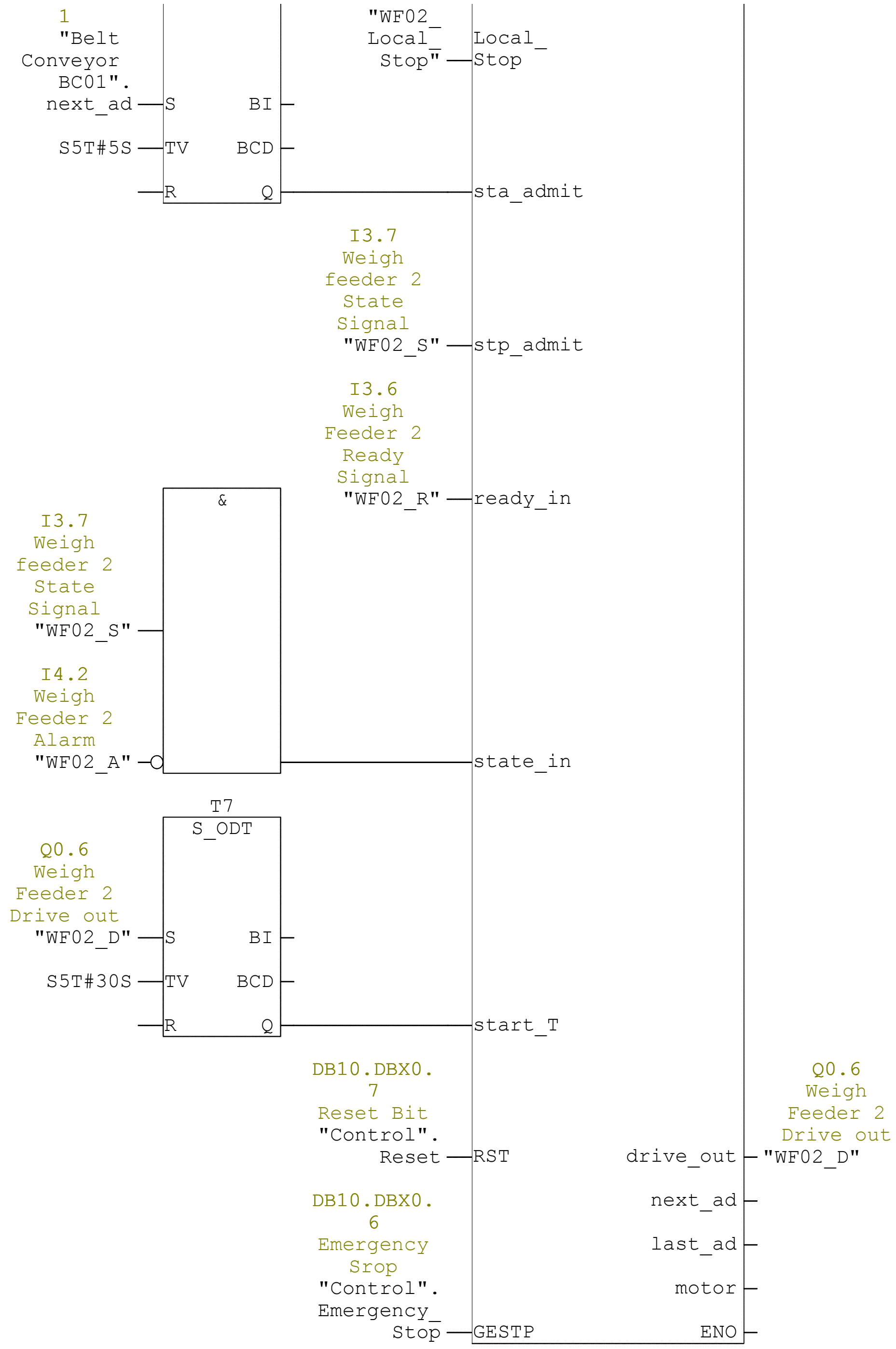
DB10.DBX3.2
Weigh Feeder 2 Remote/Local
"Control".WF02_Local — Remote_Local

I4.0
Weigh Feeder 2 Local Start
"WF02_Local_Start" — Local_Start

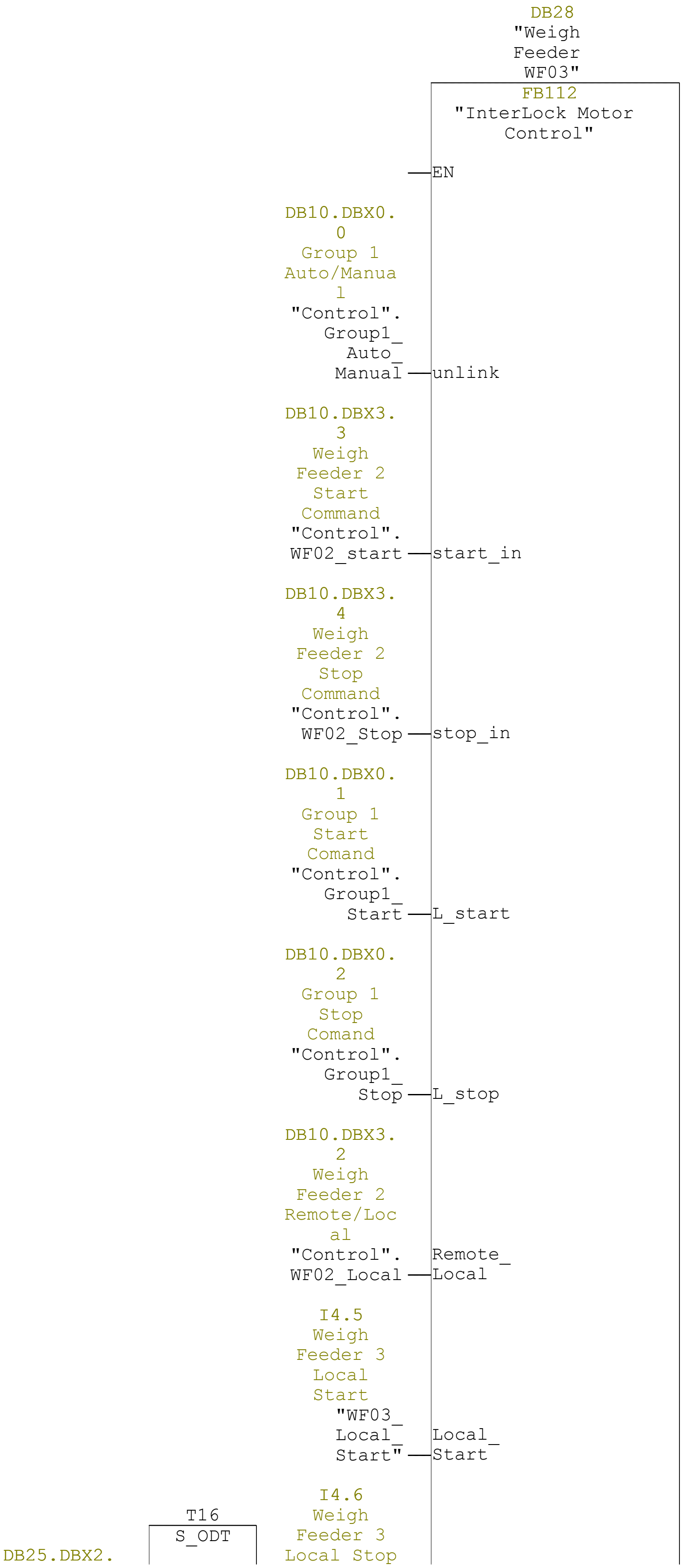
I4.1
Weigh Feeder 2 Local Stop

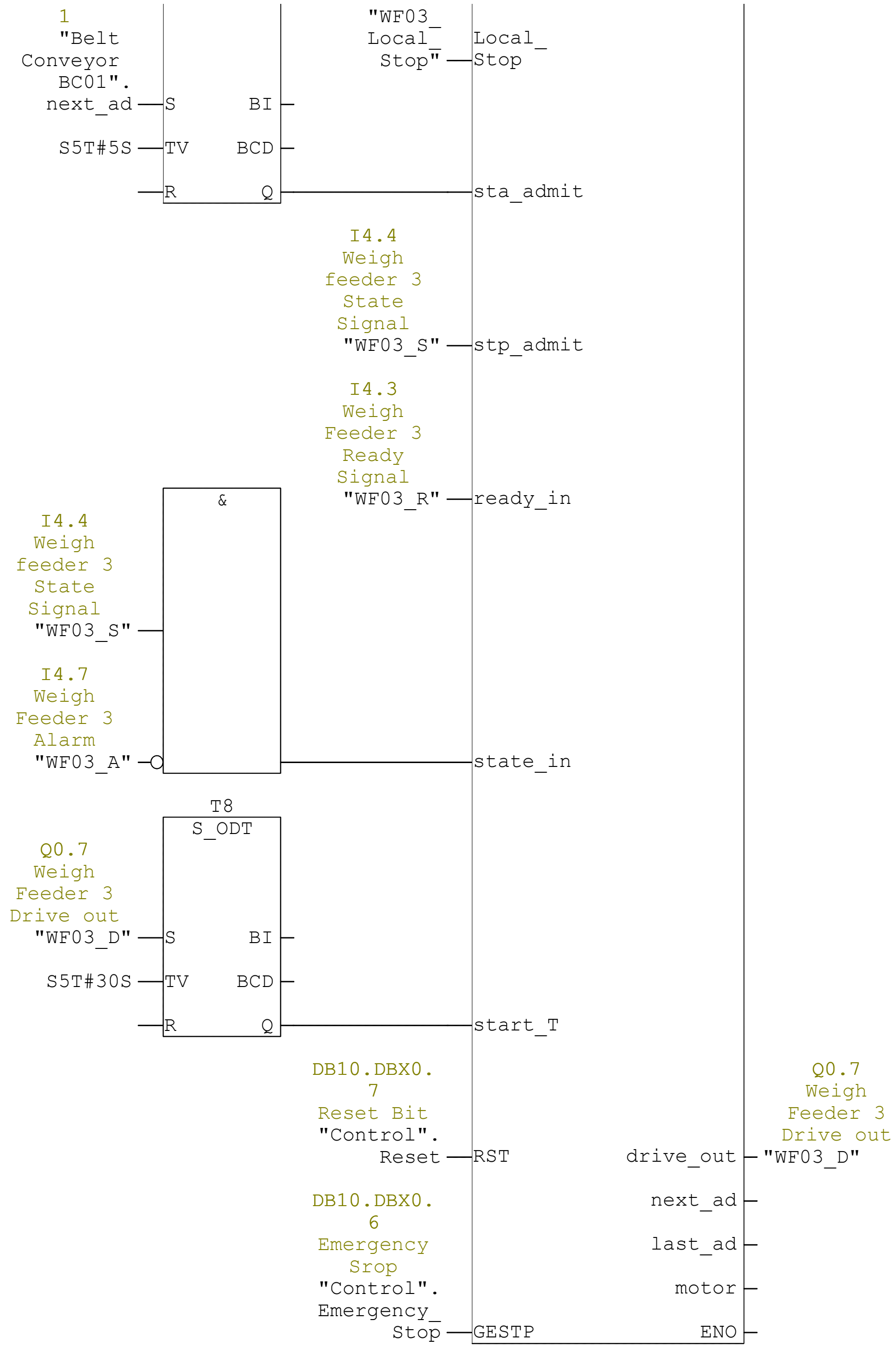
T15
S_ODT

DB25.DBX2.

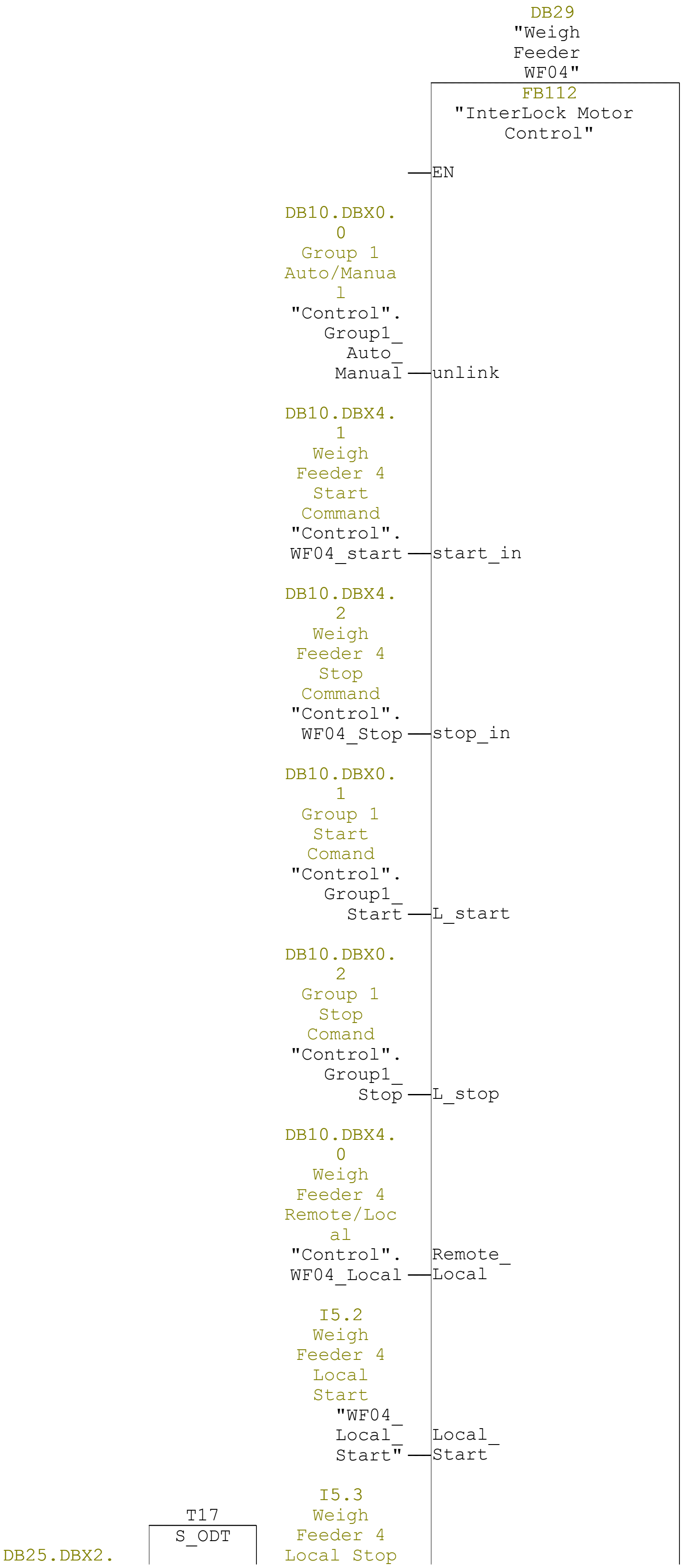

1
"Belt
Conveyor
BC01".
next_ad
S
BI
S5T#5S
TV
BCD
R
Q
"WF02_
Local_
Stop"
Local_
Stop
sta_admit
I3.7
Weigh
feeder 2
State
Signal
"WF02_S"
stp_admit
I3.6
Weigh
Feeder 2
Ready
Signal
"WF02_R"
ready_in
&
I3.7
Weigh
feeder 2
State
Signal
"WF02_S"
I4.2
Weigh
Feeder 2
Alarm
"WF02_A"
state_in
T7
S_ODT
Q0.6
Weigh
Feeder 2
Drive out
"WF02_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q0.6
Weigh
Feeder 2
Drive out
"WF02_D"
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
next_ad
last_ad
motor
ENO

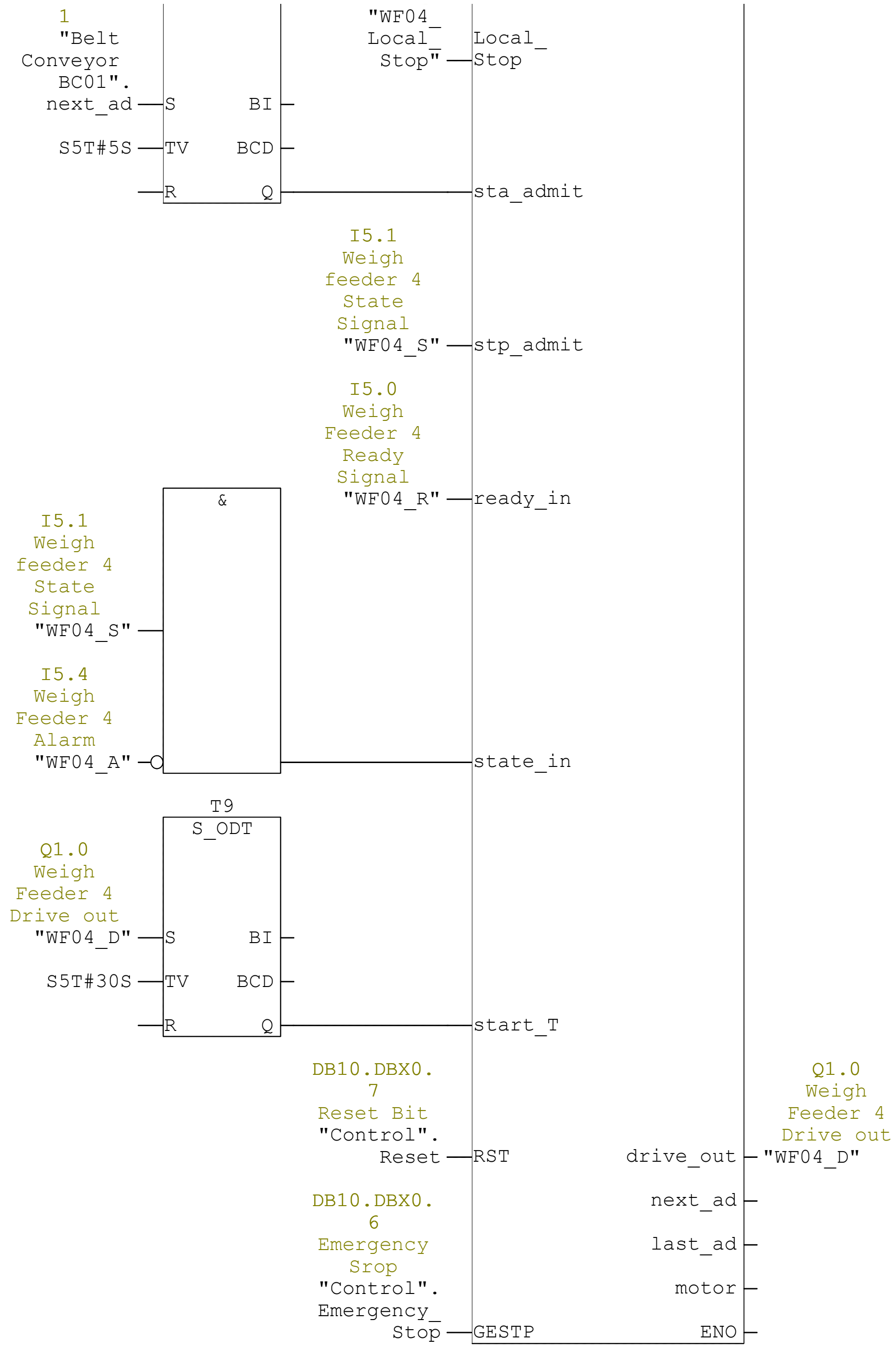
Network: 8


DB28
"Weigh
Feeder
WF03"
FB112
"InterLock Motor
Control"
EN
DB10.DBX0.
0
Group 1
Auto/Manua
l
"Control".
Group1_
Auto_
Manual
unlink
DB10.DBX3.
3
Weigh
Feeder 2
Start
Command
"Control".
WF02_start
start_in
DB10.DBX3.
4
Weigh
Feeder 2
Stop
Command
"Control".
WF02_Stop
stop_in
DB10.DBX0.
1
Group 1
Start
Comand
"Control".
Group1_
Start
L_start
DB10.DBX0.
2
Group 1
Stop
Comand
"Control".
Group1_
Stop
L_stop
DB10.DBX3.
2
Weigh
Feeder 2
Remote/Loc
al
"Control".
WF02_Local
Remote_
Local
I4.5
Weigh
Feeder 3
Local
Start
"WF03_
Local_
Start"
Local_
Start
I4.6
Weigh
Feeder 3
Local Stop
T16
S_ODT
DB25.DBX2.

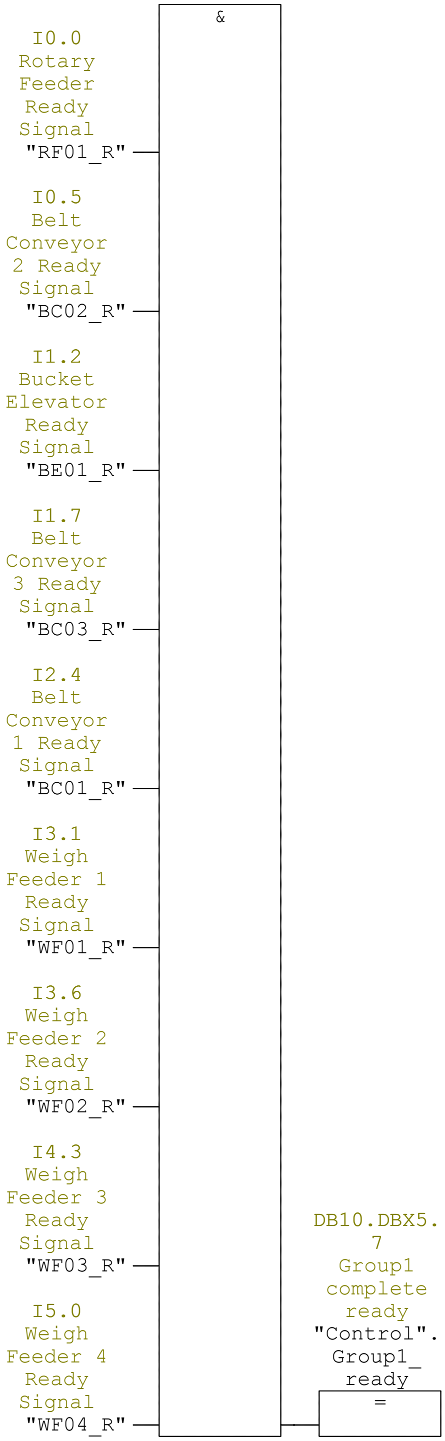

1
"Belt
Conveyor
BC01".
next_ad
S
BI
S5T#5S
TV
BCD
R
Q
"WF03_
Local_
Stop"
Local_
Stop
sta_admit
I4.4
Weigh
feeder 3
State
Signal
"WF03_S"
stp_admit
I4.3
Weigh
Feeder 3
Ready
Signal
"WF03_R"
ready_in
&
I4.4
Weigh
feeder 3
State
Signal
"WF03_S"
I4.7
Weigh
Feeder 3
Alarm
"WF03_A"
state_in
T8
S_ODT
Q0.7
Weigh
Feeder 3
Drive out
"WF03_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q0.7
Weigh
Feeder 3
Drive out
"WF03_D"
next_ad
last_ad
motor
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
ENO

Network: 9

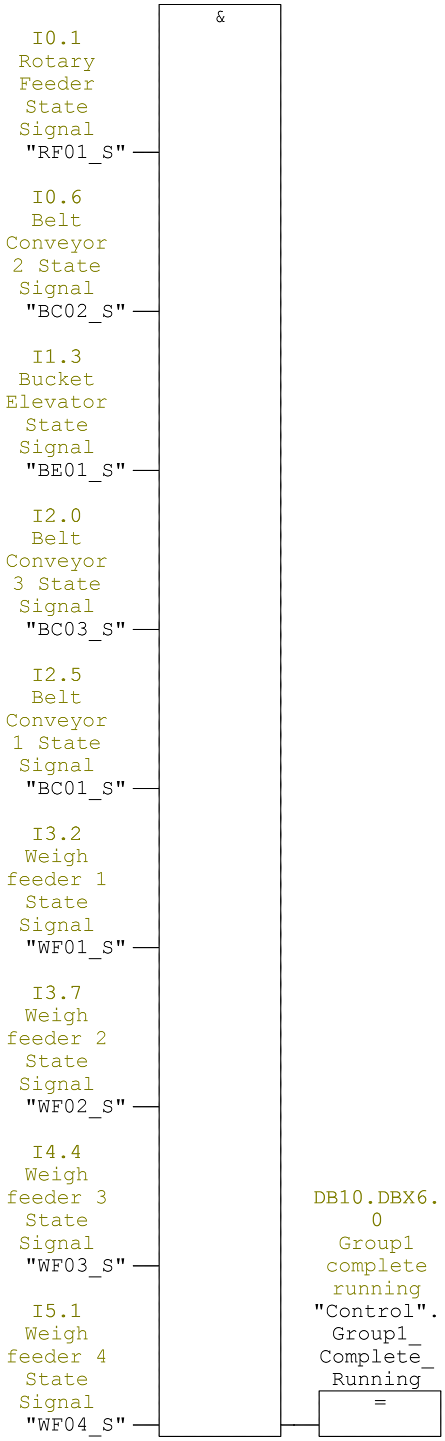

DB29
"Weigh Feeder WF04"
FB112
"InterLock Motor Control"
EN
DB10.DBX0.0
Group 1 Auto/Manual
"Control". Group1_ Auto_ Manual
unlink
DB10.DBX4.1
Weigh Feeder 4 Start Command
"Control". WF04_start
start_in
DB10.DBX4.2
Weigh Feeder 4 Stop Command
"Control". WF04_Stop
stop_in
DB10.DBX0.1
Group 1 Start Comand
"Control". Group1_ Start
L_start
DB10.DBX0.2
Group 1 Stop Comand
"Control". Group1_ Stop
L_stop
DB10.DBX4.0
Weigh Feeder 4 Remote/Local
"Control". WF04_Local
Remote_ Local
I5.2
Weigh Feeder 4 Local Start
"WF04_ Local_ Start"
Local_ Start
I5.3
Weigh Feeder 4 Local Stop
T17
S_ODT
DB25.DBX2.


1
"Belt
Conveyor
BC01".
next_ad
S
BI
S5T#5S
TV
BCD
R
Q
"WF04_
Local_
Stop"
Local_
Stop
sta_admit
I5.1
Weigh
feeder 4
State
Signal
"WF04_S"
stp_admit
I5.0
Weigh
Feeder 4
Ready
Signal
"WF04_R"
ready_in
&
I5.1
Weigh
feeder 4
State
Signal
"WF04_S"
I5.4
Weigh
Feeder 4
Alarm
"WF04_A"
state_in
T9
S_ODT
Q1.0
Weigh
Feeder 4
Drive out
"WF04_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q1.0
Weigh
Feeder 4
Drive out
"WF04_D"
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
next_ad
last_ad
motor
ENO

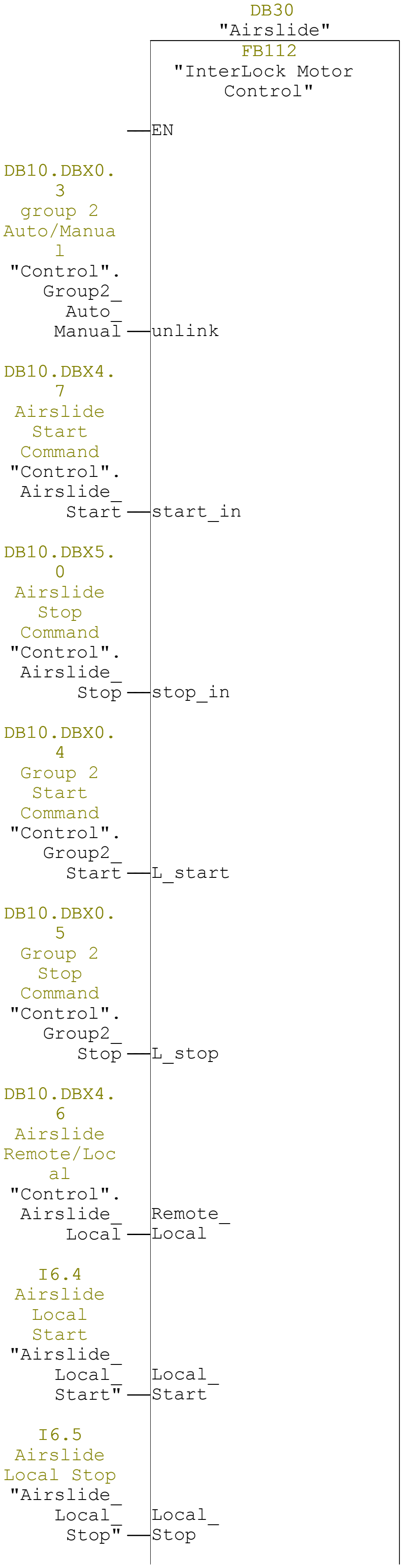
Network: 10 Group1 complete ready


&
I0.0
Rotary Feeder Ready Signal
"RF01_R"
I0.5
Belt Conveyor 2 Ready Signal
"BC02_R"
I1.2
Bucket Elevator Ready Signal
"BE01_R"
I1.7
Belt Conveyor 3 Ready Signal
"BC03_R"
I2.4
Belt Conveyor 1 Ready Signal
"BC01_R"
I3.1
Weigh Feeder 1 Ready Signal
"WF01_R"
I3.6
Weigh Feeder 2 Ready Signal
"WF02_R"
I4.3
Weigh Feeder 3 Ready Signal
"WF03_R"
I5.0
Weigh Feeder 4 Ready Signal
"WF04_R"
DB10.DBX5.7
Group1 complete ready
"Control".Group1_ready
=

Network: 11 Group1 complete running


&
I0.1
Rotary Feeder State Signal
"RF01_S"
I0.6
Belt Conveyor 2 State Signal
"BC02_S"
I1.3
Bucket Elevator State Signal
"BE01_S"
I2.0
Belt Conveyor 3 State Signal
"BC03_S"
I2.5
Belt Conveyor 1 State Signal
"BC01_S"
I3.2
Weigh feeder 1 State Signal
"WF01_S"
I3.7
Weigh feeder 2 State Signal
"WF02_S"
I4.4
Weigh feeder 3 State Signal
"WF03_S"
I5.1
Weigh feeder 4 State Signal
"WF04_S"
DB10.DBX6.0
Group1 complete running
"Control".Group1_Complete_Running
=

Network: 12



DB30
"Airslide"
FB112
"InterLock Motor Control"

EN

DB10.DBX0.3
group 2 Auto/Manual
"Control".Group2_Auto_Manual — unlink

DB10.DBX4.7
Airslide Start Command
"Control".Airslide_Start — start_in

DB10.DBX5.0
Airslide Stop Command
"Control".Airslide_Stop — stop_in

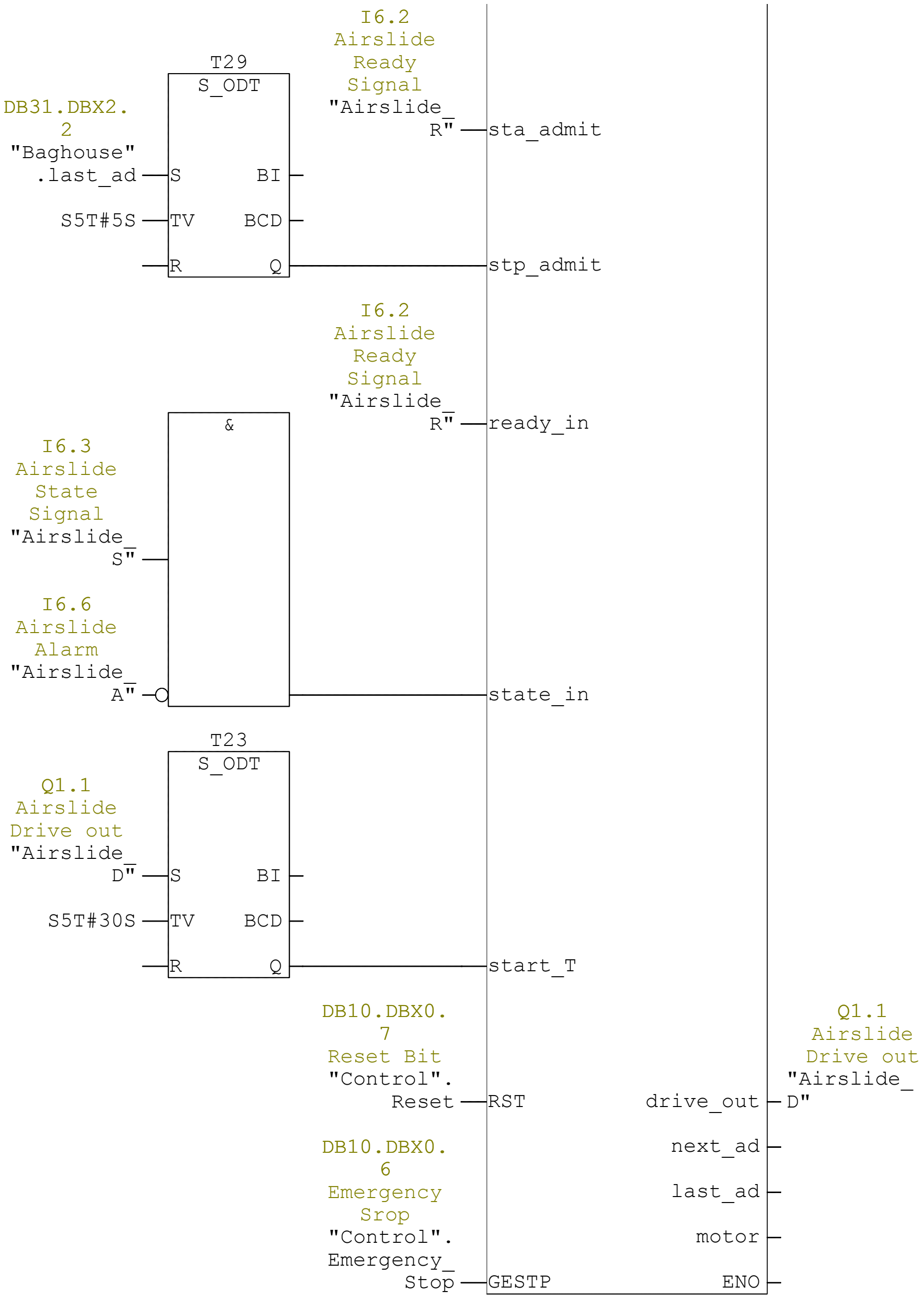
DB10.DBX0.4
Group 2 Start Command
"Control".Group2_Start — L_start

DB10.DBX0.5
Group 2 Stop Command
"Control".Group2_Stop — L_stop

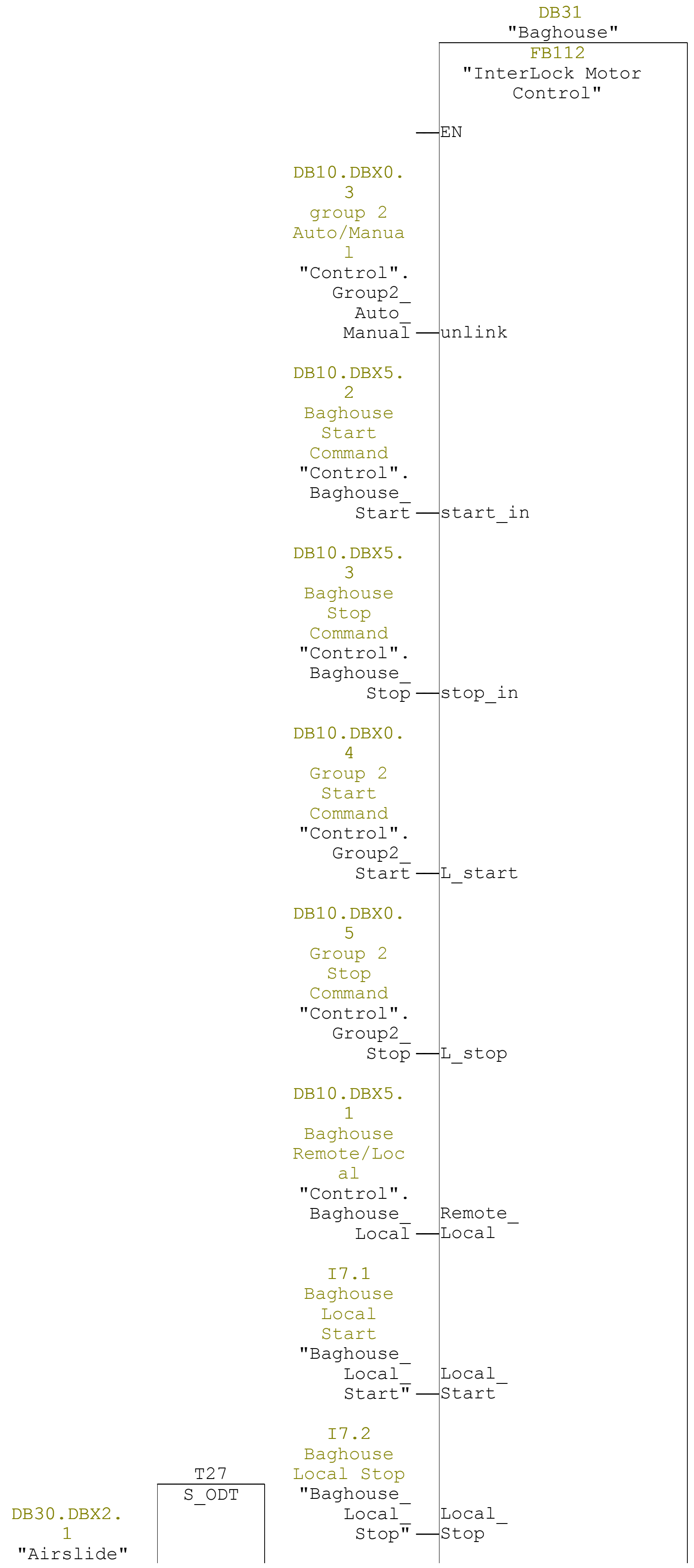
DB10.DBX4.6
Airslide Remote/Local
"Control".Airslide_Local — Remote_Local

I6.4
Airslide Local Start
"Airslide_Local_Start" — Local_Start

I6.5
Airslide Local Stop
"Airslide_Local_Stop" — Local_Stop


I6.2
Airslide
Ready
Signal
"Airslide_
R"
sta_admit
T29
S_ODT
DB31.DBX2.
2
"Baghouse"
.last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I6.2
Airslide
Ready
Signal
"Airslide_
R"
ready_in
&
I6.3
Airslide
State
Signal
"Airslide_
S"
I6.6
Airslide
Alarm
"Airslide_
A"
state_in
T23
S_ODT
Q1.1
Airslide
Drive out
"Airslide_
D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q1.1
Airslide
Drive out
"Airslide_
D"
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
next_ad
last_ad
motor
ENO

Network: 13



DB31
"Baghouse"
FB112
"InterLock Motor Control"

EN

DB10.DBX0.3
group 2 Auto/Manual
"Control".Group2_Auto_Manual — unlink

DB10.DBX5.2
Baghouse Start Command
"Control".Baghouse_Start — start_in

DB10.DBX5.3
Baghouse Stop Command
"Control".Baghouse_Stop — stop_in

DB10.DBX0.4
Group 2 Start Command
"Control".Group2_Start — L_start

DB10.DBX0.5
Group 2 Stop Command
"Control".Group2_Stop — L_stop

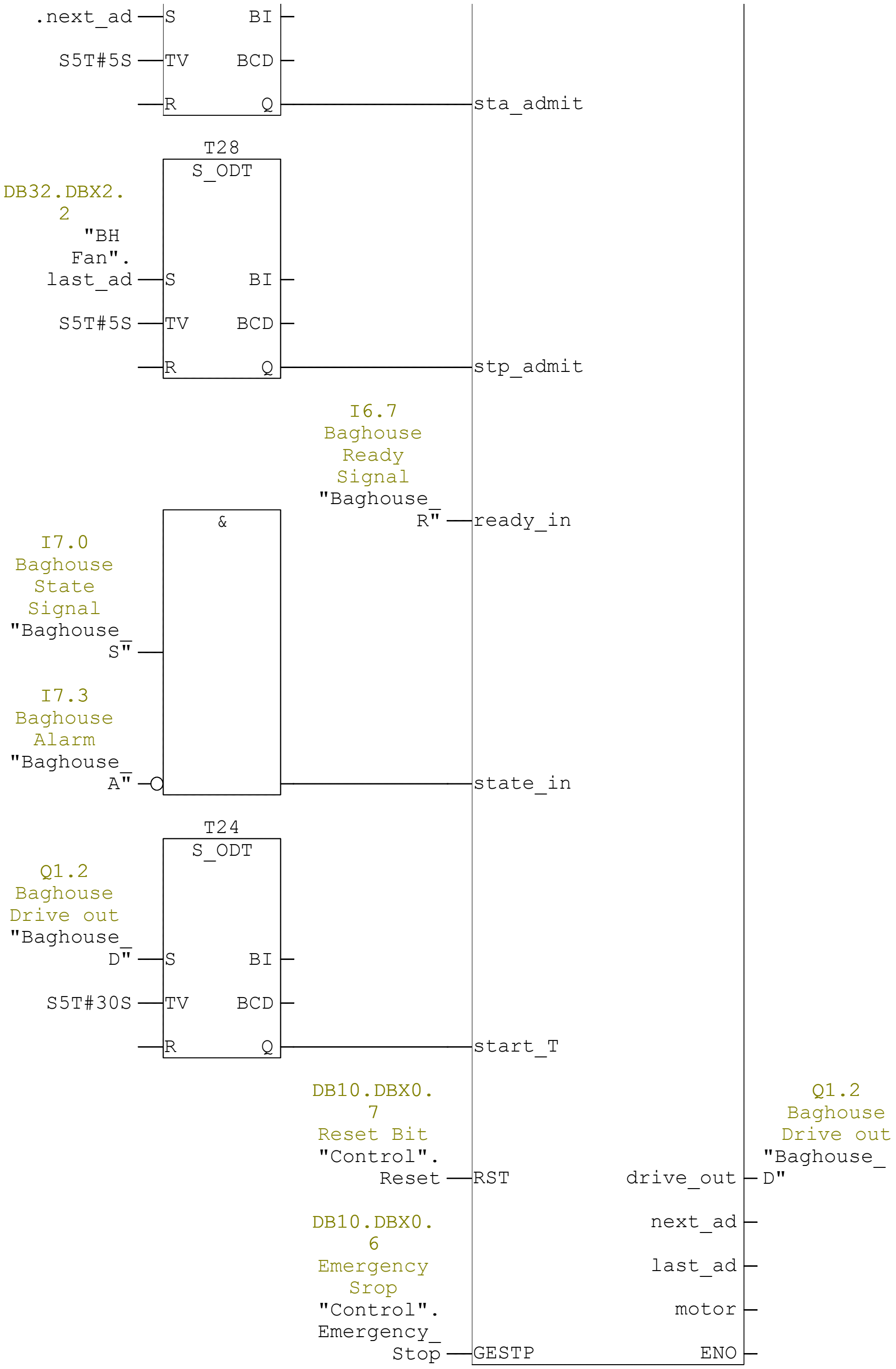
DB10.DBX5.1
Baghouse Remote/Local
"Control".Baghouse_Local — Remote_Local

I7.1
Baghouse Local Start
"Baghouse_Local_Start" — Local_Start

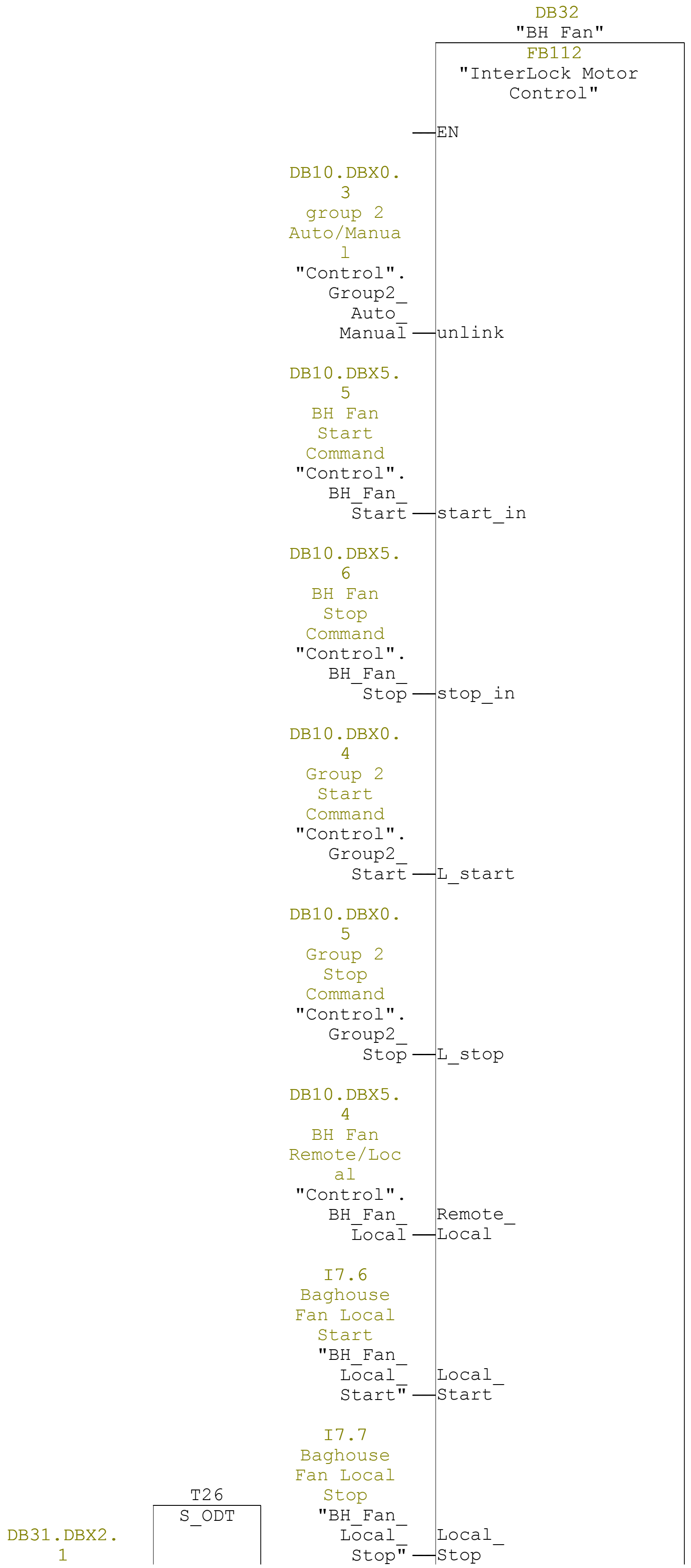
I7.2
Baghouse Local Stop
"Baghouse_Local_Stop" — Local_Stop

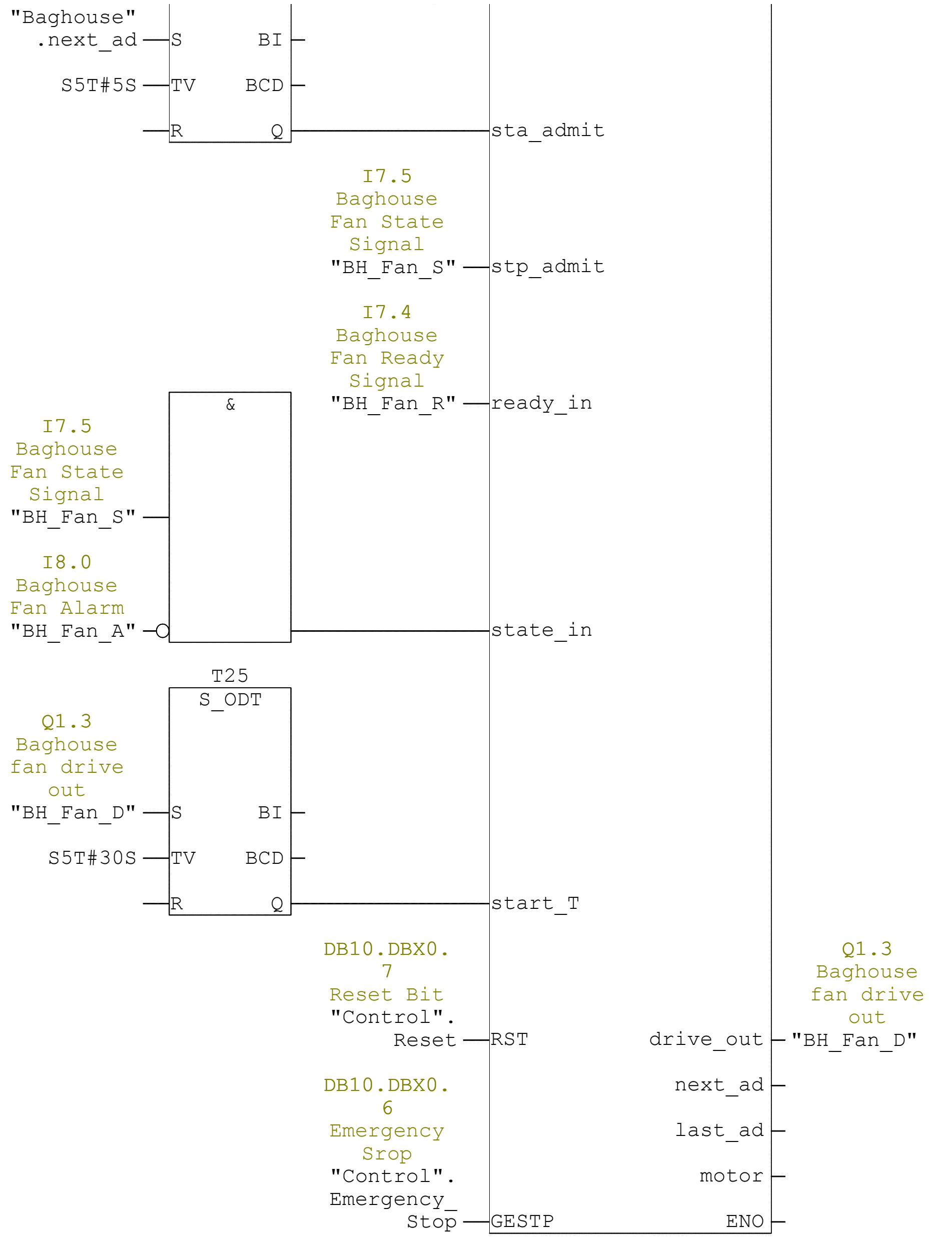
T27
S_ODT

DB30.DBX2.1
"Airslide"

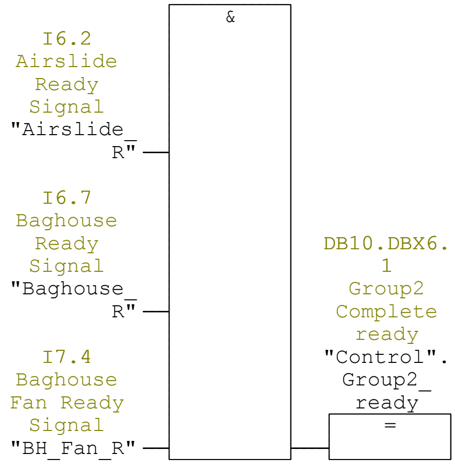

.next_ad
S
BI
S5T#5S
TV
BCD
R
Q
sta_admit
T28
S_ODT
DB32.DBX2.
2
"BH
Fan".
last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I6.7
Baghouse
Ready
Signal
"Baghouse_
R"
ready_in
&
I7.0
Baghouse
State
Signal
"Baghouse_
S"
I7.3
Baghouse
Alarm
"Baghouse_
A"
state_in
T24
S_ODT
Q1.2
Baghouse
Drive out
"Baghouse_
D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q1.2
Baghouse
Drive out
"Baghouse_
D"
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
next_ad
last_ad
motor
ENO

Network: 14

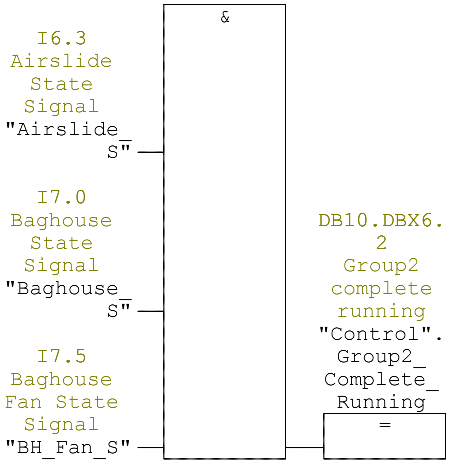

DB32
"BH Fan"
FB112
"InterLock Motor Control"
EN
DB10.DBX0.3
group 2 Auto/Manual
"Control".Group2_Auto_Manual
unlink
DB10.DBX5.5
BH Fan Start Command
"Control".BH_Fan_Start
start_in
DB10.DBX5.6
BH Fan Stop Command
"Control".BH_Fan_Stop
stop_in
DB10.DBX0.4
Group 2 Start Command
"Control".Group2_Start
L_start
DB10.DBX0.5
Group 2 Stop Command
"Control".Group2_Stop
L_stop
DB10.DBX5.4
BH Fan Remote/Local
"Control".BH_Fan_Local
Remote_Local
I7.6
Baghouse Fan Local Start
"BH_Fan_Local_Start"
Local_Start
I7.7
Baghouse Fan Local Stop
"BH_Fan_Local_Stop"
Local_Stop
T26
S_ODT
DB31.DBX2.1



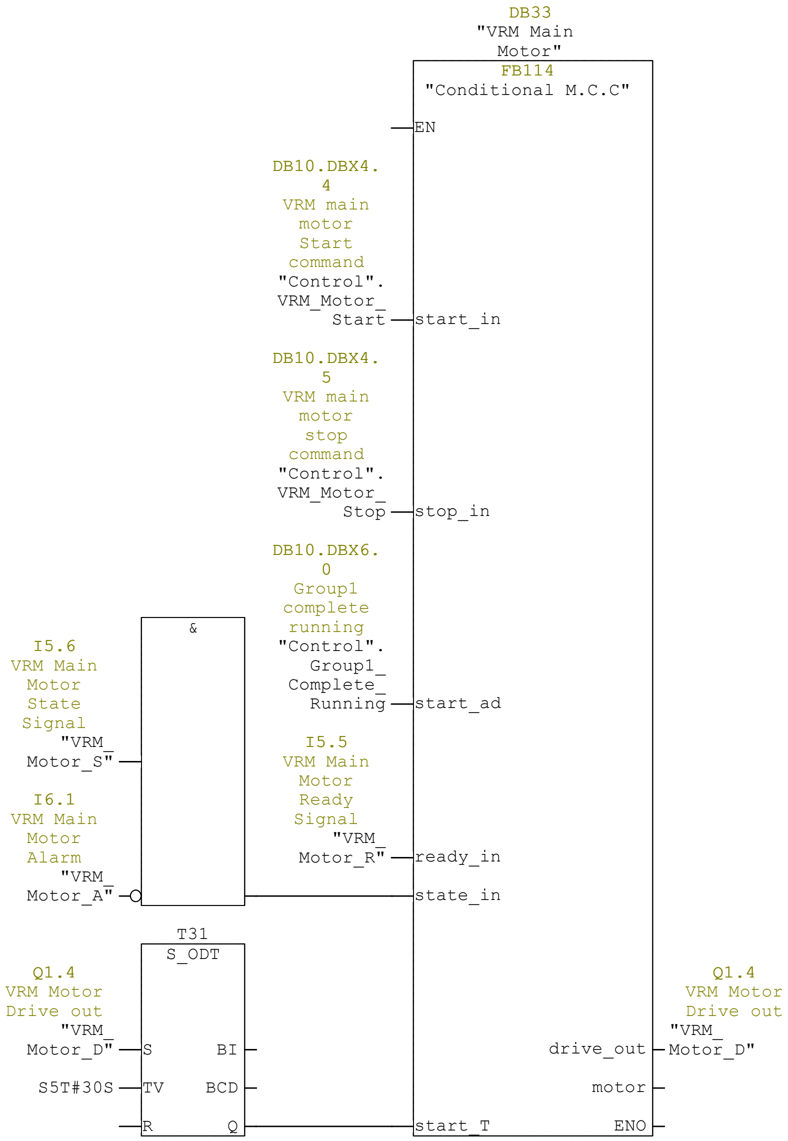
Network: 15 Group1 Complete ready



Network: 16 Group1 complete running



Network: 17



FC3 - <offline>

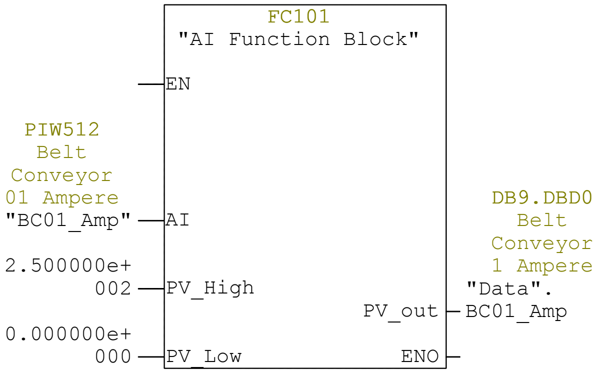
"Analog Parameters"

Name:	**Family:**	
Author:	**Version:** 0.1	
	Block version: 2	
Time stamp Code:	04/24/2022 03:39:14 PM	
Interface:	04/09/2022 01:53:10 PM	
Lengths (block/logic/data):	01484 01338 00018	

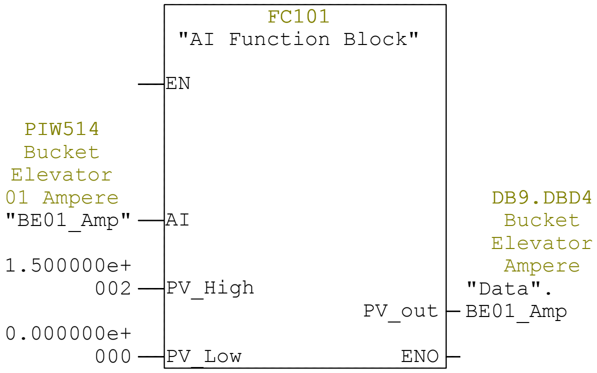
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC3

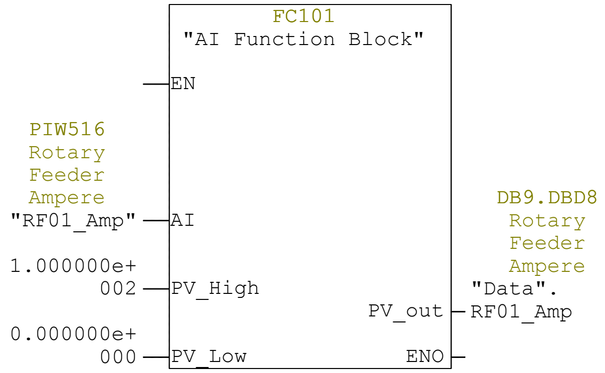
Network: 1



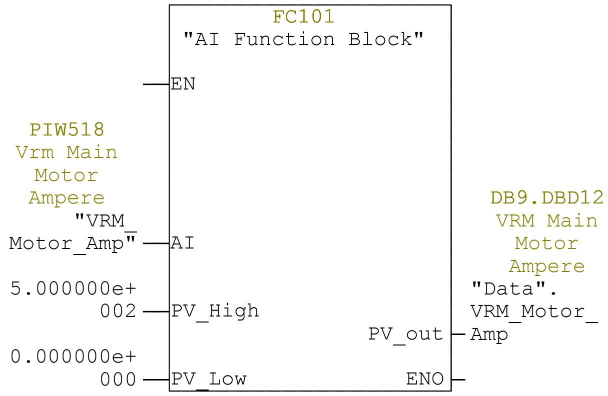
Network: 2



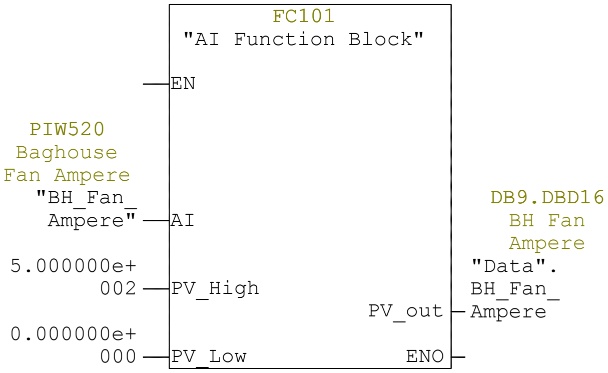
Network: 3



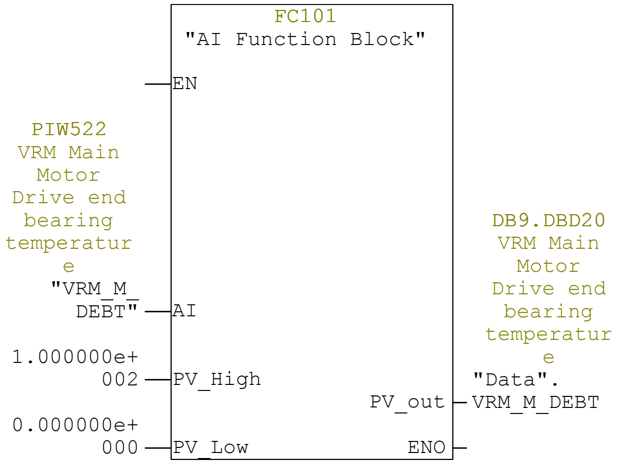
Network: 4



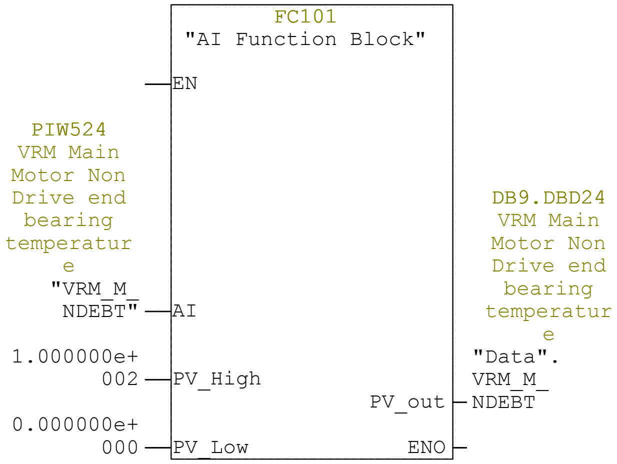
Network: 5



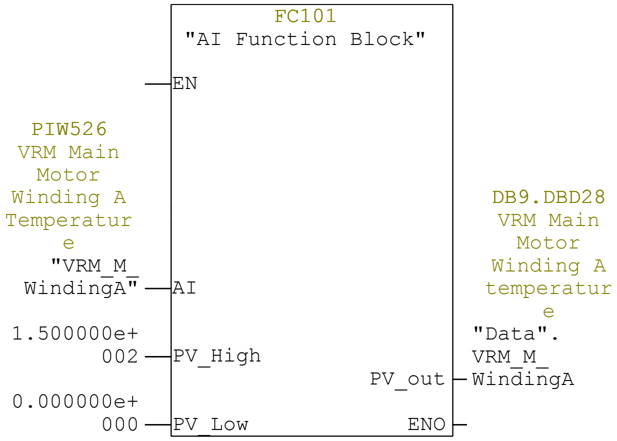
Network: 6



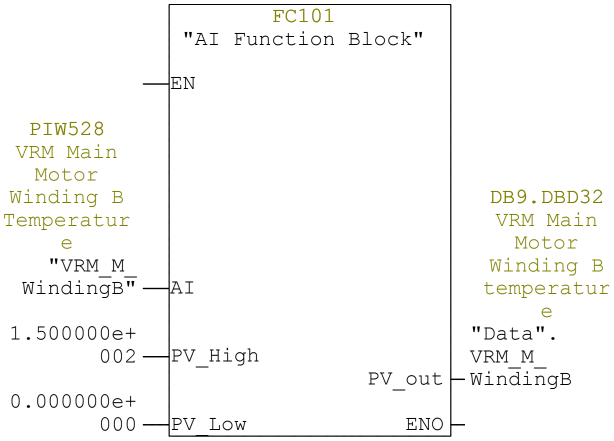
Network: 7



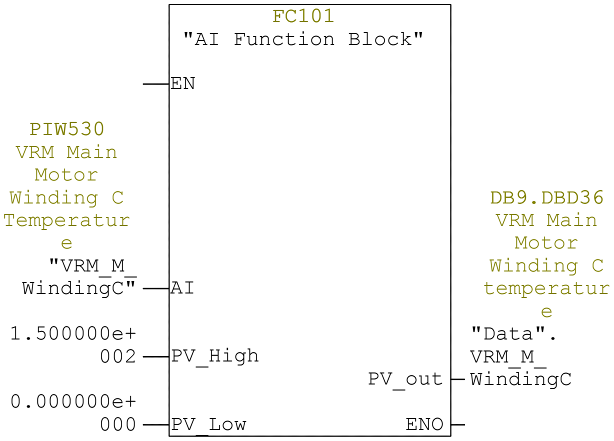
Network: 8



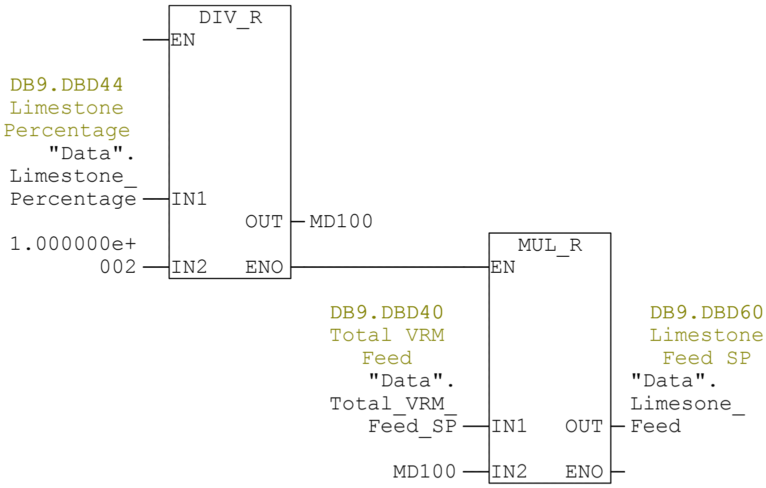
Network: 9



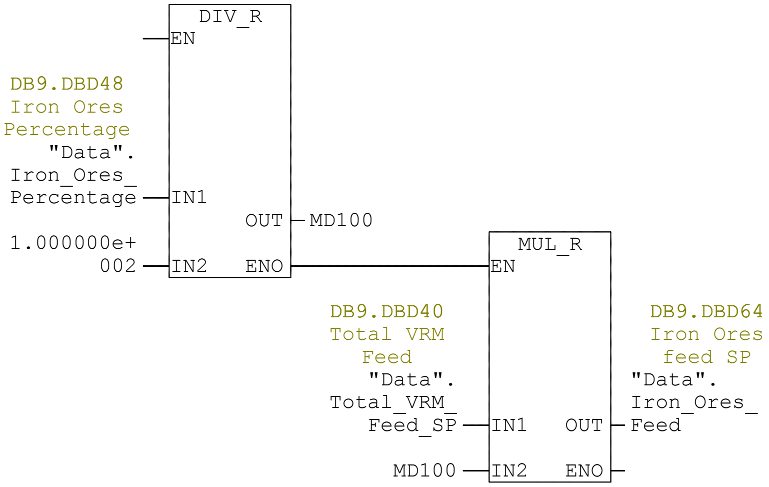
Network: 10



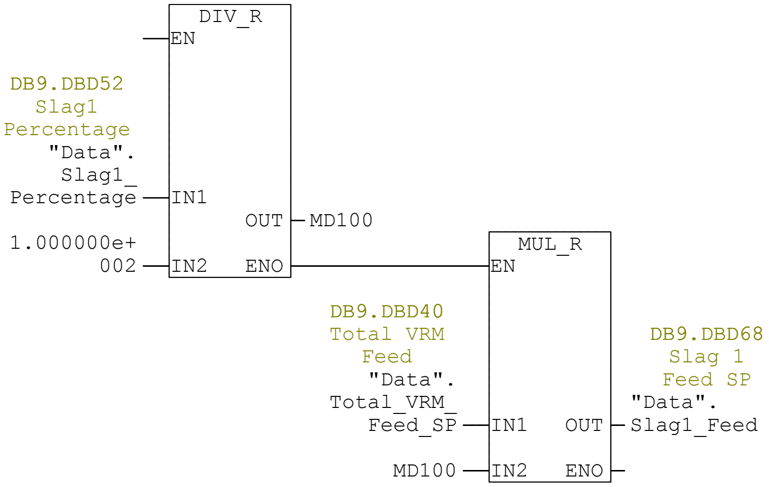
Network: 11



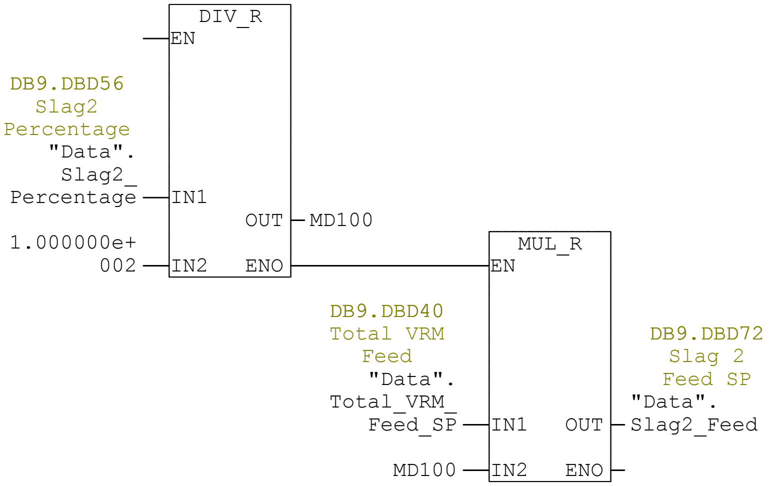
Network: 12



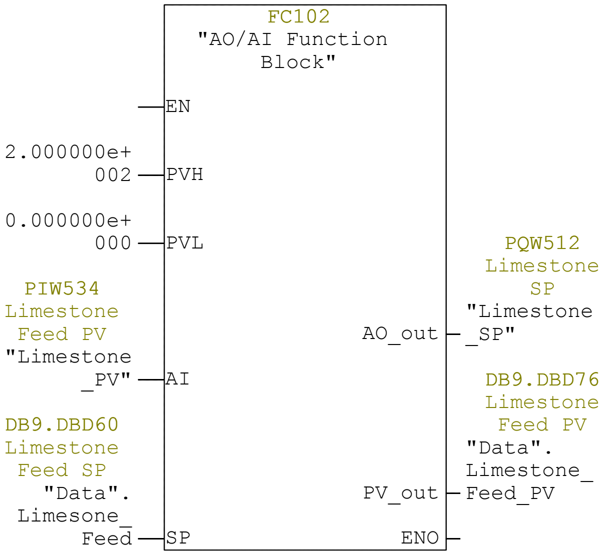
Network: 13



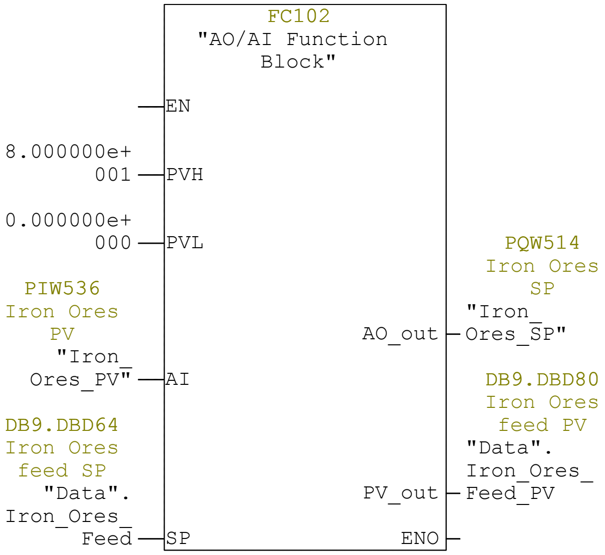
Network: 14



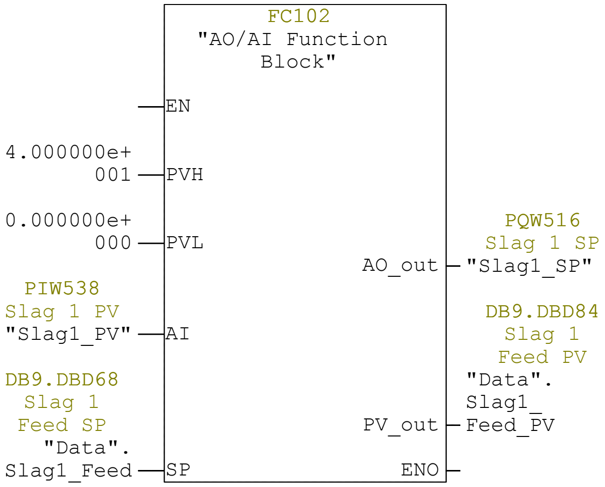
Network: 15



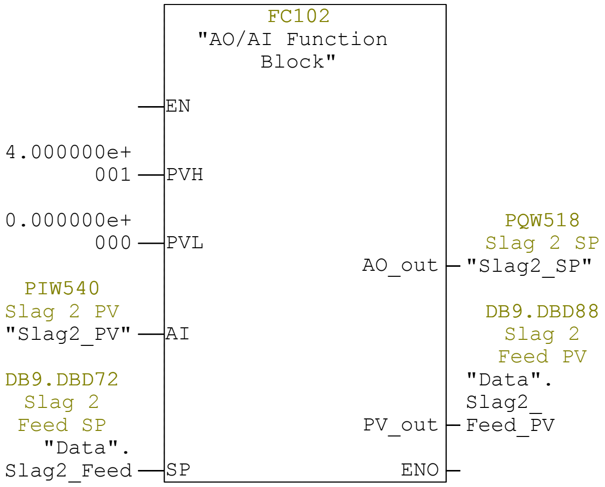
Network: 16



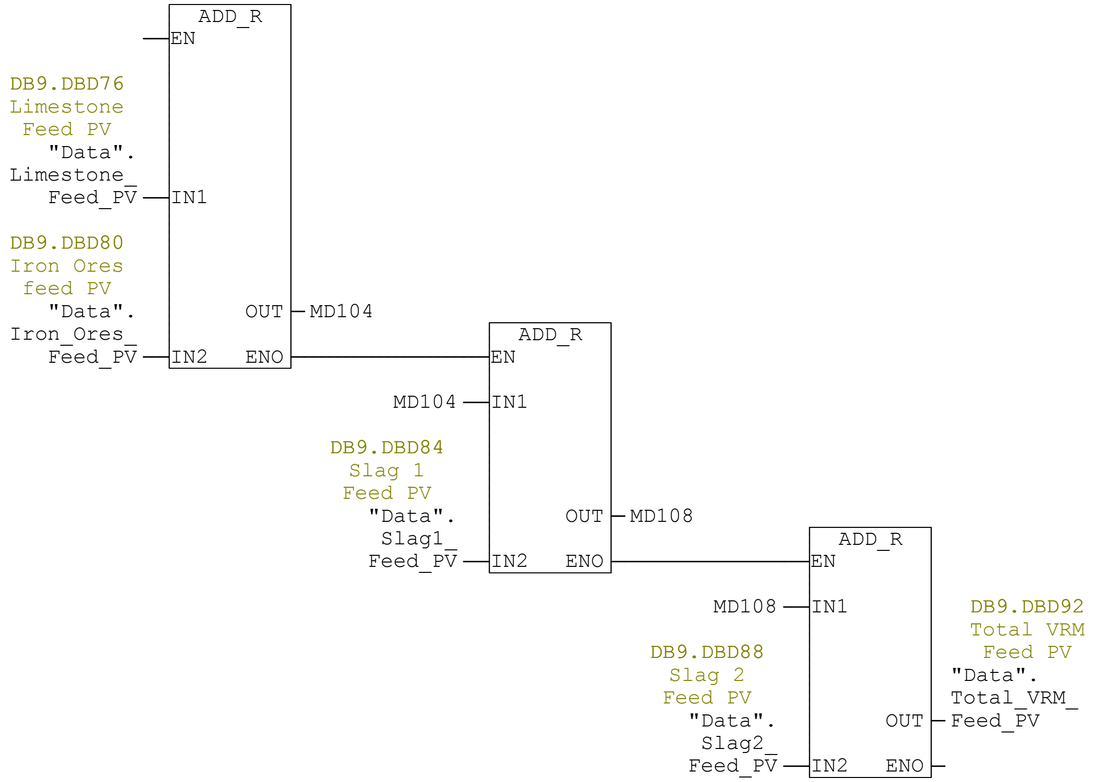
Network: 17



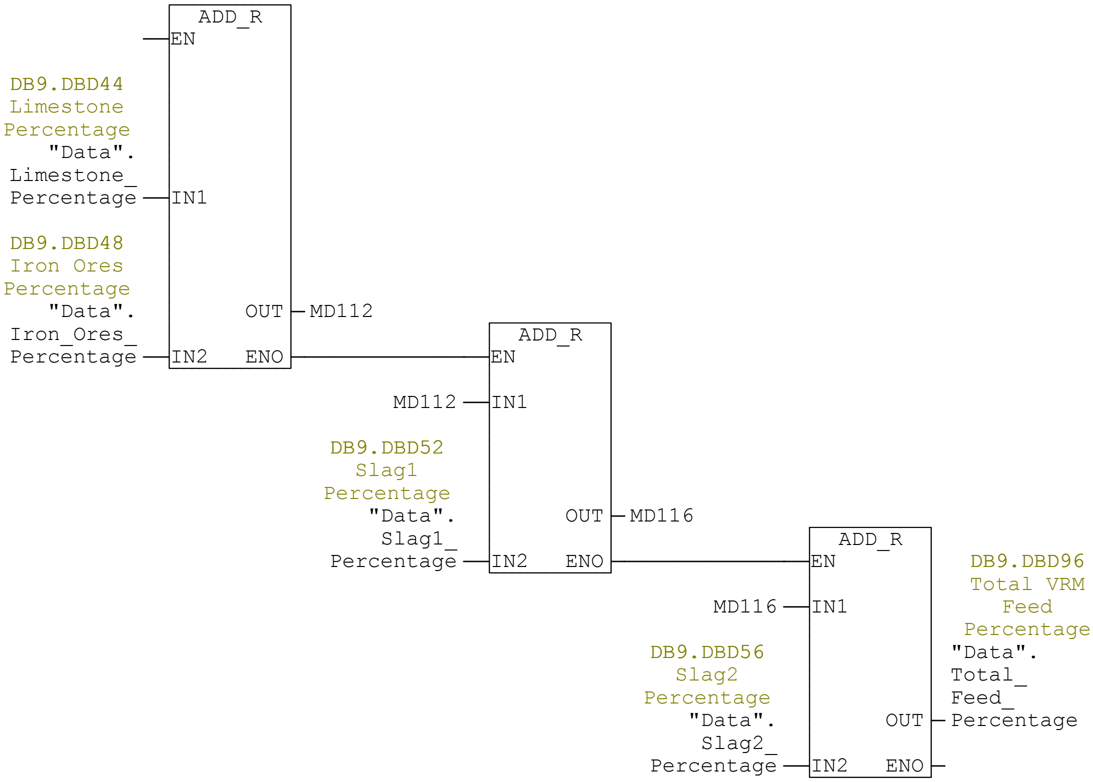
Network: 18



Network: 19



Network: 20


ADD_R
EN
DB9.DBD44 Limestone Percentage
"Data". Limestone_ Percentage
IN1
DB9.DBD48 Iron Ores Percentage
"Data". Iron_Ores_ Percentage
IN2
OUT
MD112
ENO
ADD_R
EN
MD112
IN1
DB9.DBD52 Slag1 Percentage
"Data". Slag1_ Percentage
IN2
OUT
MD116
ENO
ADD_R
EN
MD116
IN1
DB9.DBD56 Slag2 Percentage
"Data". Slag2_ Percentage
IN2
OUT
DB9.DBD96 Total VRM Feed Percentage
"Data". Total_ Feed_ Percentage
ENO

FC4 - <offline>

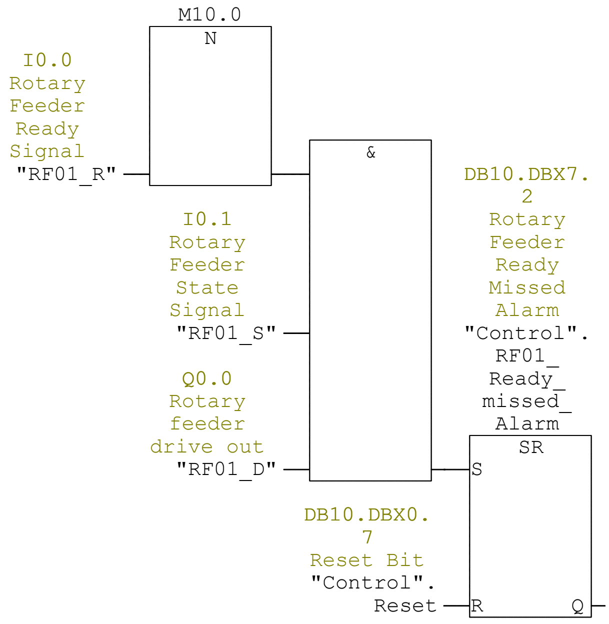
"Alarm"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/02/2022 05:03:39 PM
Interface: 04/30/2022 03:51:12 PM
Lengths (block/logic/data): 01030 00886 00000

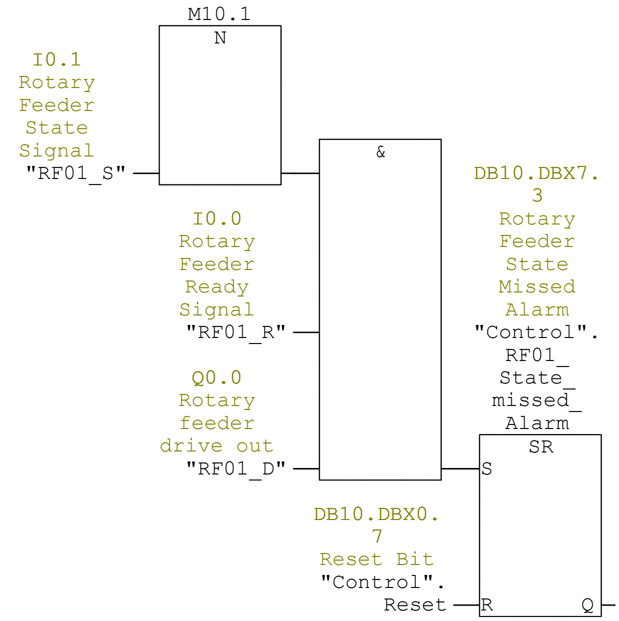
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC4

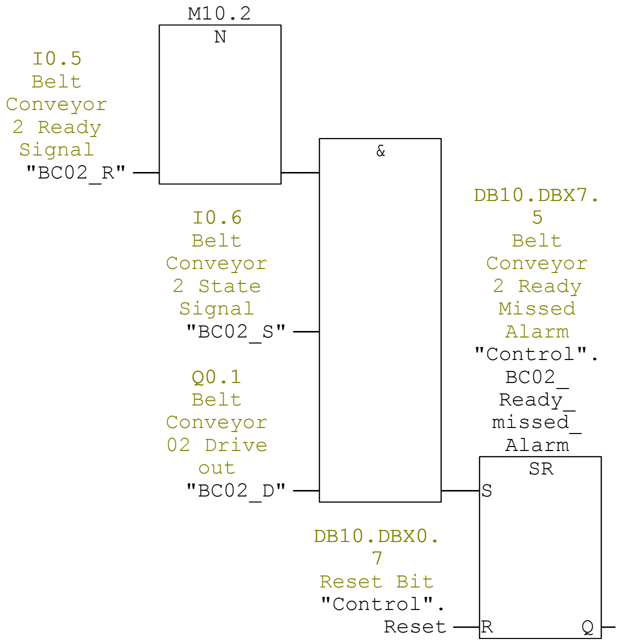
Network: 1 Apron Conveyor Ready Missed



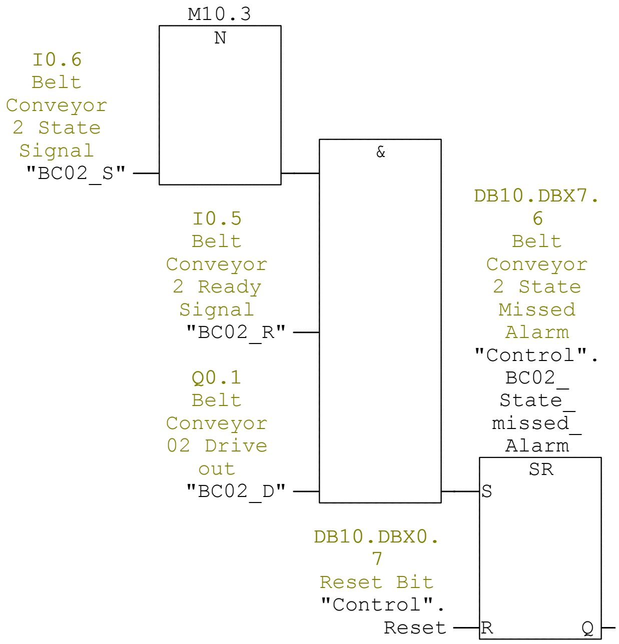
Network: 2 Apron Conveyor State Missed



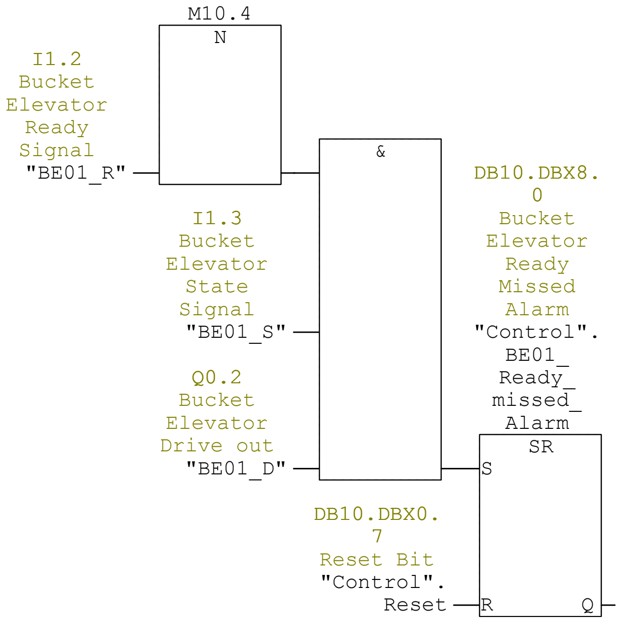
Network: 3 Main Motor Ready Missed Alarm



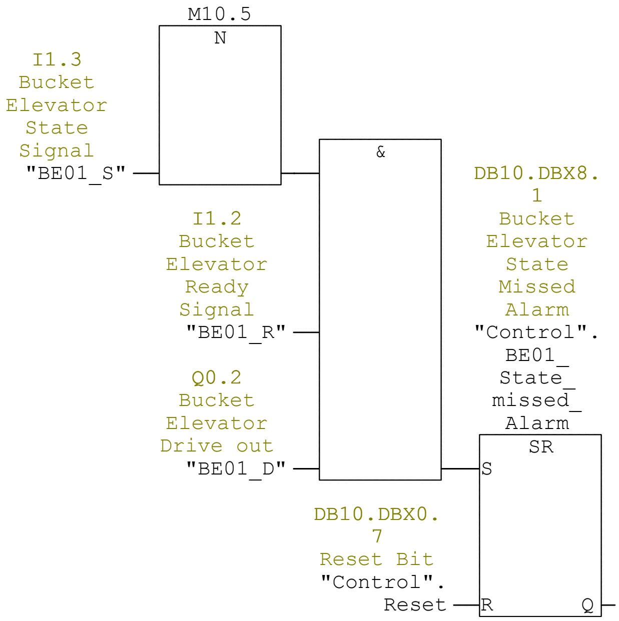
Network: 4 Main Motor State Missed Alarm



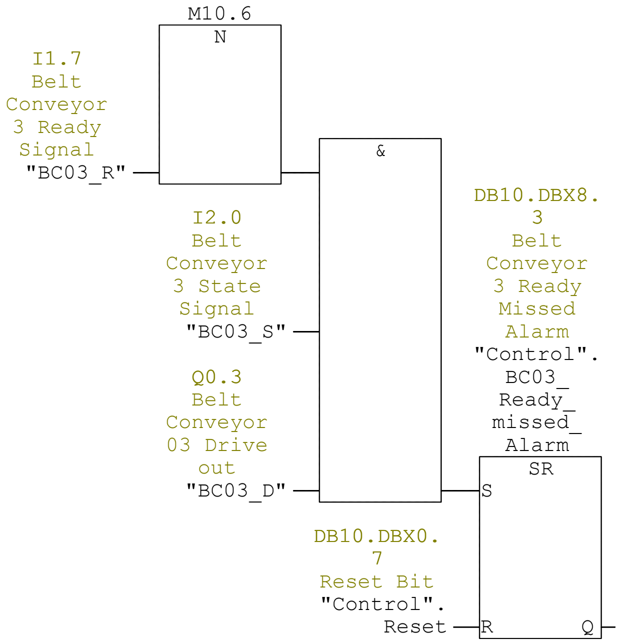
Network: 5 Crusher Belt1 Ready missed alarm



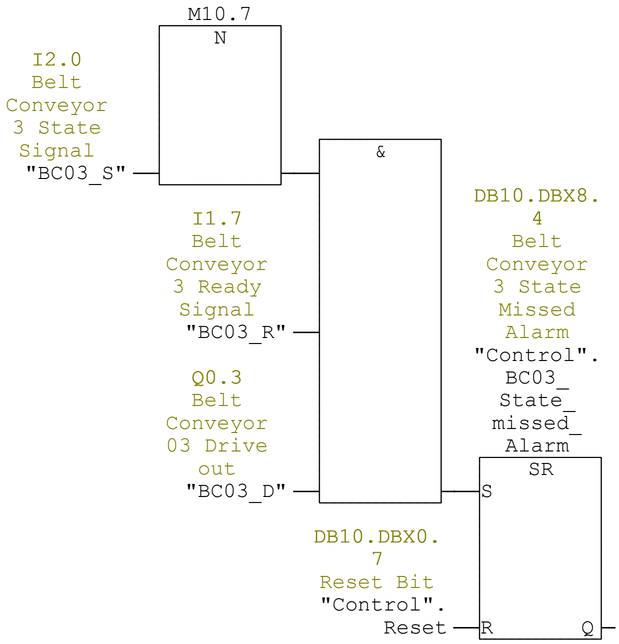
Network: 6 Crusher Belt1 State missed alarm



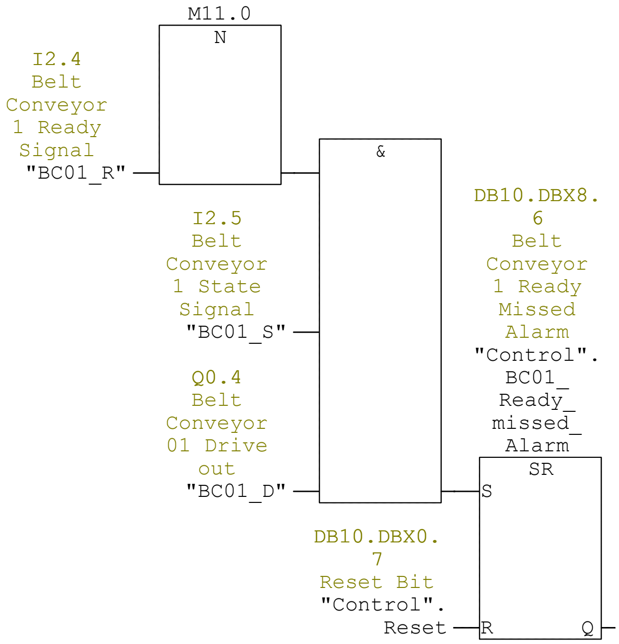
Network: 7 Crusher Belt2 Ready missed alarm



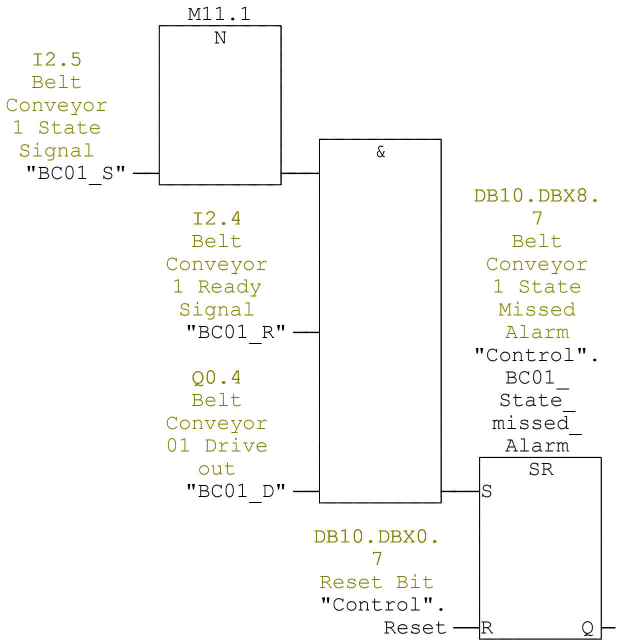
Network: 8 Crusher Belt2 State missed alarm


M10.7
N
I2.0
Belt Conveyor 3 State Signal
"BC03_S"
&
I1.7
Belt Conveyor 3 Ready Signal
"BC03_R"
Q0.3
Belt Conveyor 03 Drive out
"BC03_D"
DB10.DBX8.4
Belt Conveyor 3 State Missed Alarm
"Control".BC03_State_missed_Alarm
SR
S
DB10.DBX0.7
Reset Bit
"Control".Reset
R
Q

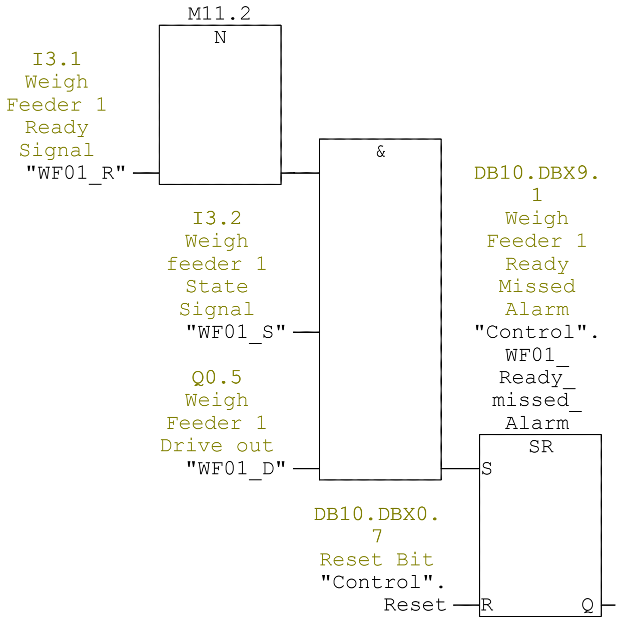
Network: 9 Crusher Belt3 Ready missed alarm


M11.0
N
I2.4
Belt Conveyor 1 Ready Signal
"BC01_R"
&
I2.5
Belt Conveyor 1 State Signal
"BC01_S"
Q0.4
Belt Conveyor 01 Drive out
"BC01_D"
DB10.DBX8.6
Belt Conveyor 1 Ready Missed Alarm
"Control".BC01_Ready_missed_Alarm
SR
S
DB10.DBX0.7
Reset Bit
"Control".Reset
R
Q

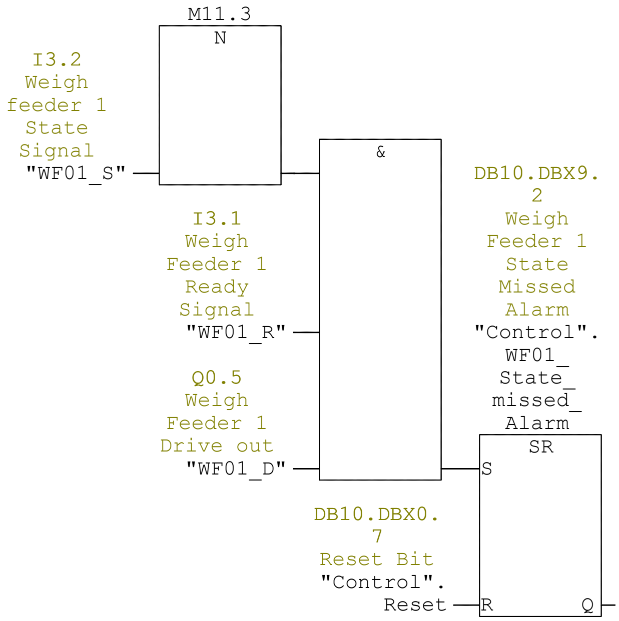
Network: 10 Crusher Belt3 State missed alarm


M11.1
N
I2.5
Belt Conveyor 1 State Signal
"BC01_S"
&
I2.4
Belt Conveyor 1 Ready Signal
"BC01_R"
Q0.4
Belt Conveyor 01 Drive out
"BC01_D"
DB10.DBX8.7
Belt Conveyor 1 State Missed Alarm
"Control".BC01_State_missed_Alarm
SR
S
DB10.DBX0.7
Reset Bit
"Control".Reset
R
Q

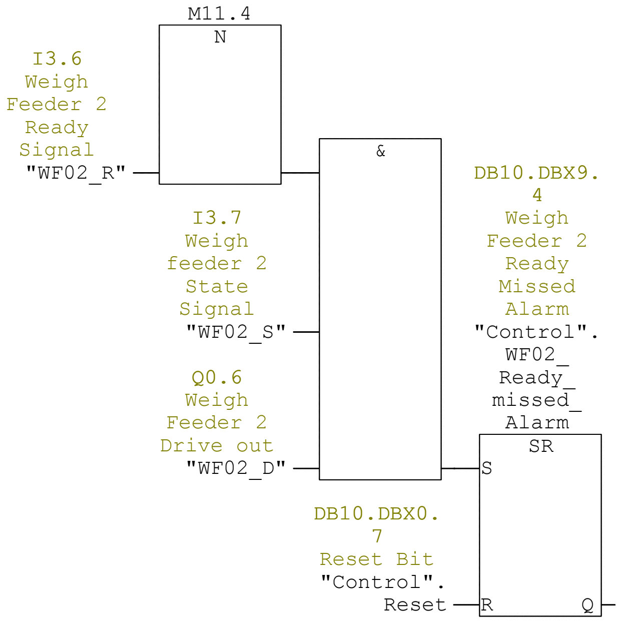
Network: 11 Stacker Belt Ready Missed Alarm


M11.2
N
I3.1
Weigh Feeder 1 Ready Signal
"WF01_R"
&
I3.2
Weigh feeder 1 State Signal
"WF01_S"
Q0.5
Weigh Feeder 1 Drive out
"WF01_D"
DB10.DBX9.1
Weigh Feeder 1 Ready Missed Alarm
"Control".WF01_Ready_missed_Alarm
SR
S
DB10.DBX0.7
Reset Bit
"Control".Reset
R
Q

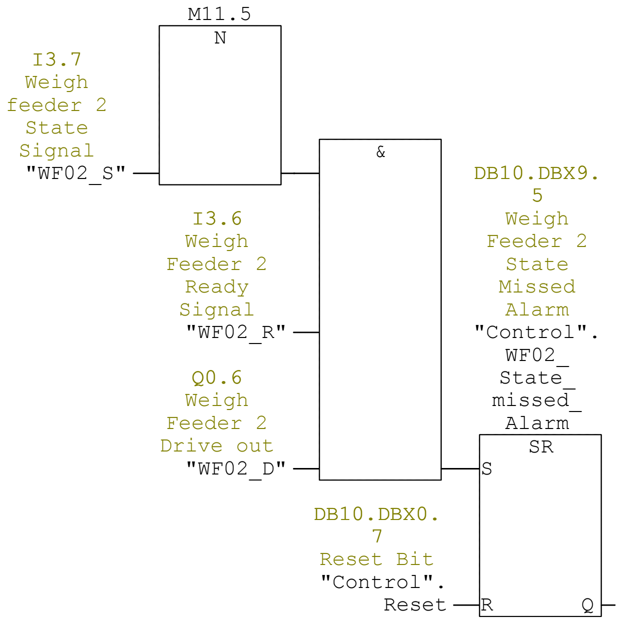
Network: 12 Stacker Belt State Missed Alarm



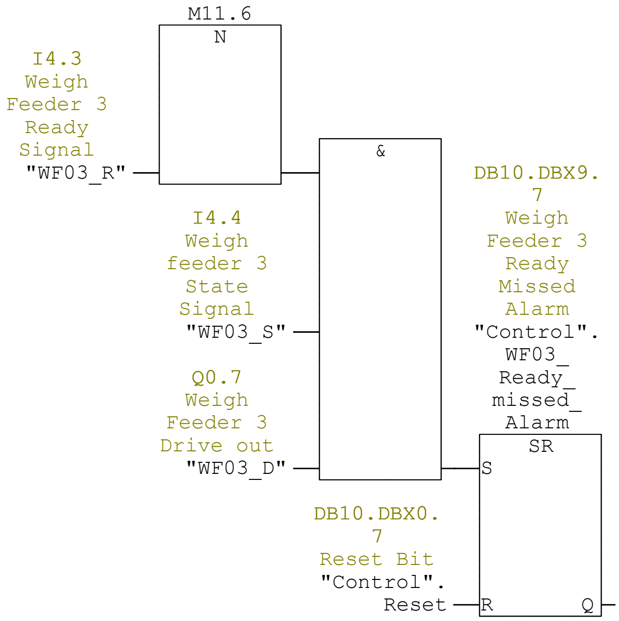
Network: 13 Hopper Feeding Belt Ready Missed Alarm



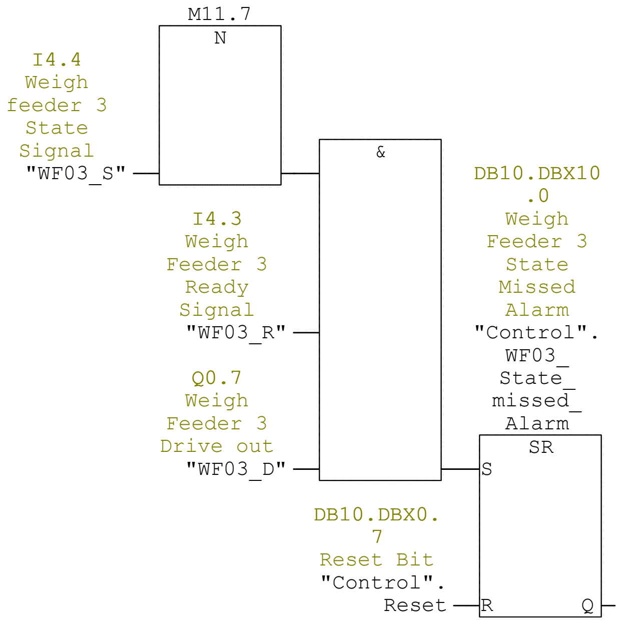
Network: 14 Hopper Feeding Belt State Missed Alarm



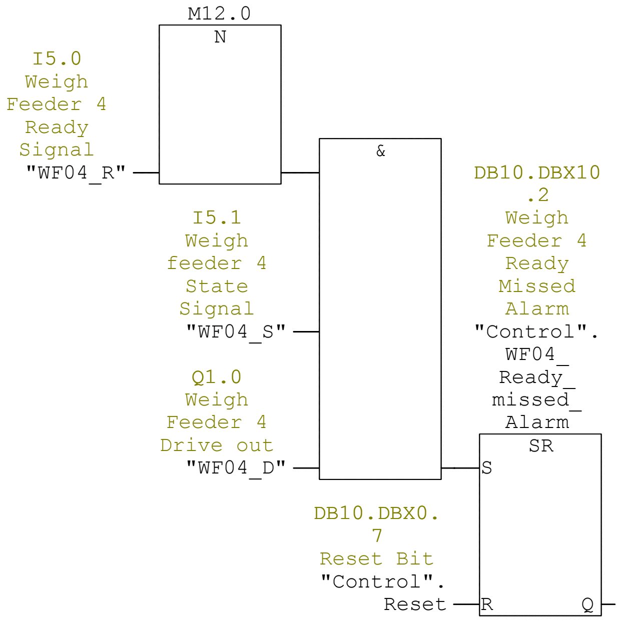
Network: 15 Weigh Feeder 3 Ready Missed Alarm



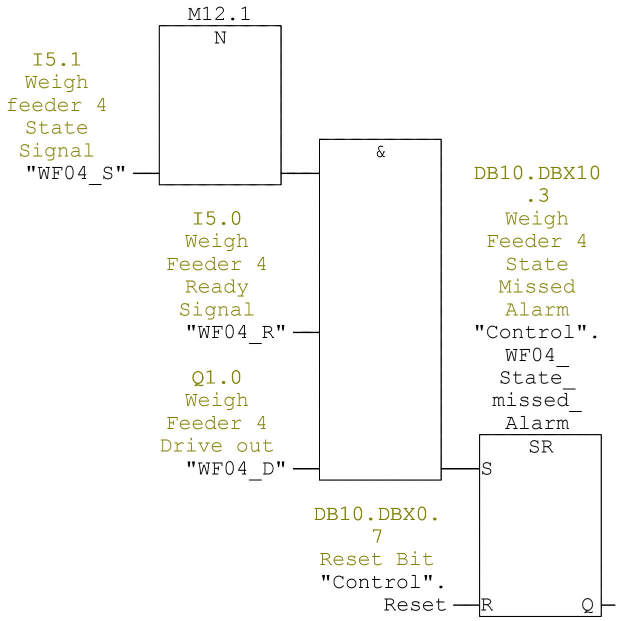
Network: 16 Weigh Feeder 3 State Missed Alarm



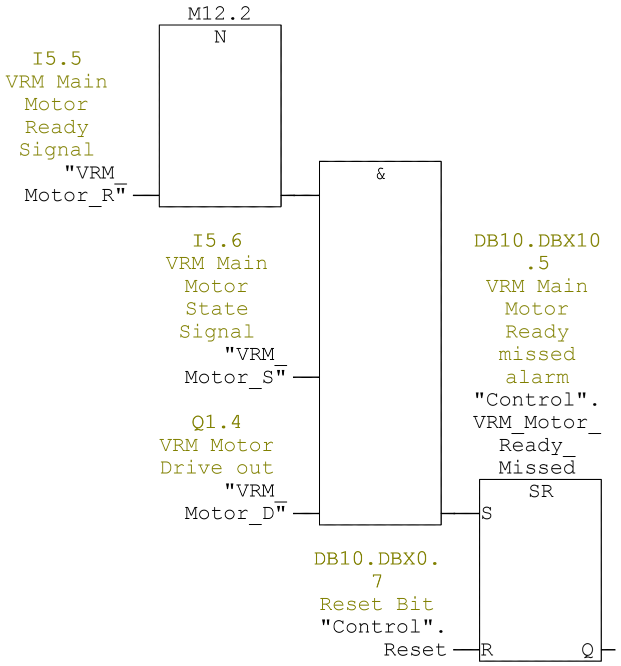
Network: 17 Weigh Feeder 4 Ready Missed Alarm



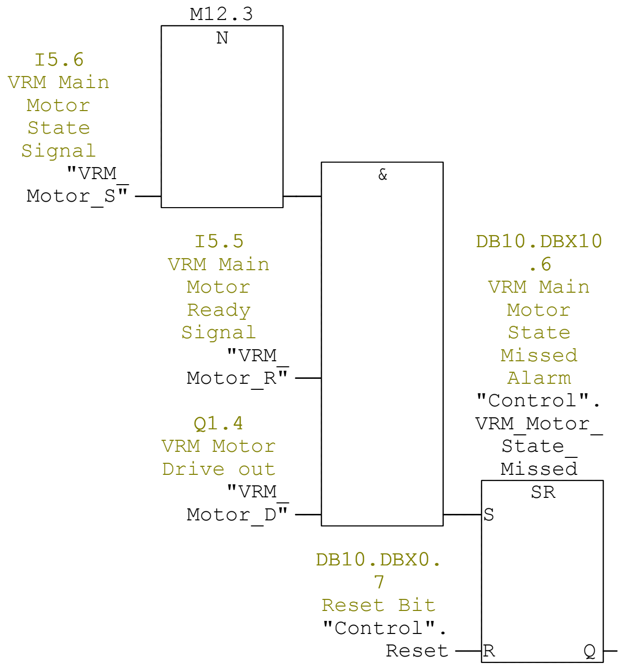
Network: 18 Weigh Feeder 4 State Missed Alarm



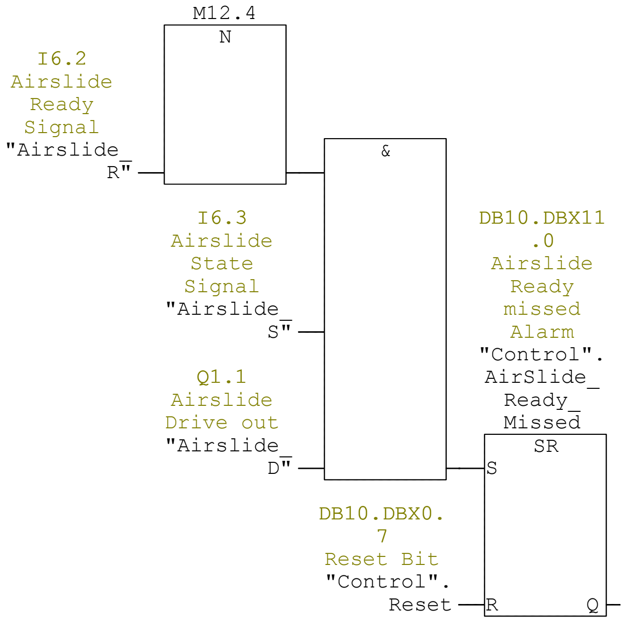
Network: 19 VRM Main Motor Ready missed alarm



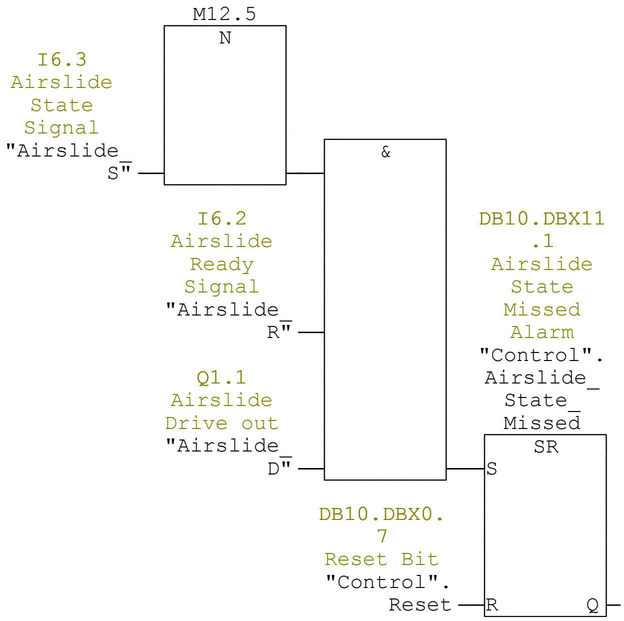
Network: 20 VRM Main Motor State Missed Alarm



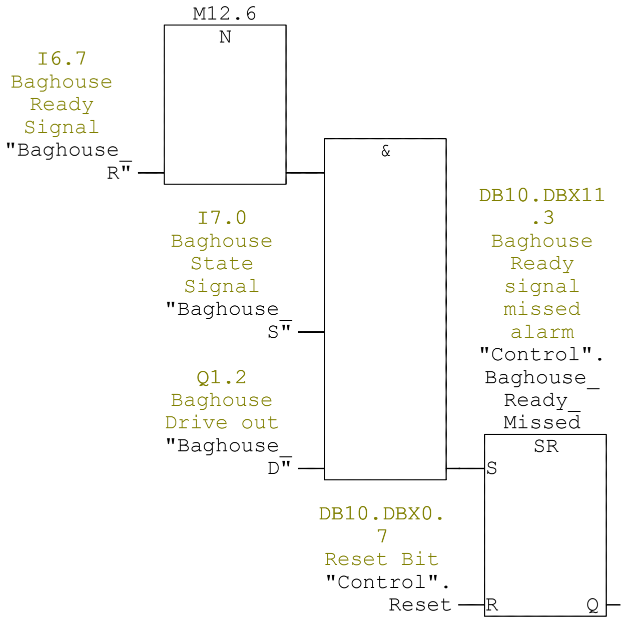
Network: 21 Airslide Ready missed Alarm



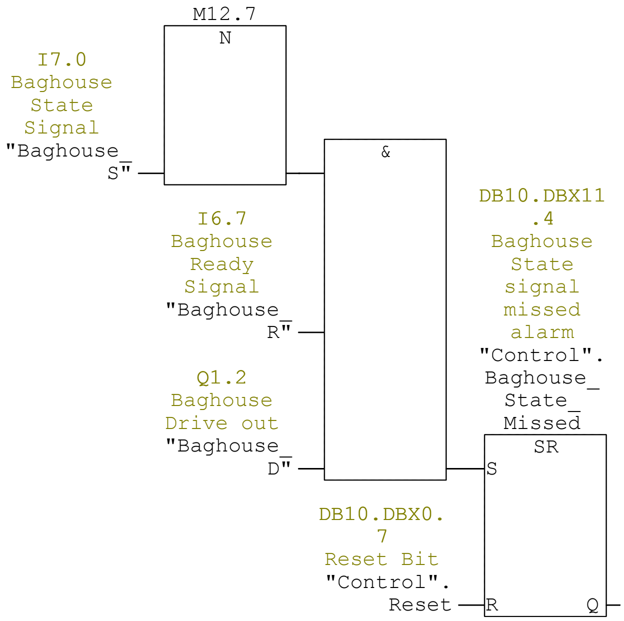
Network: 22 Airslide State Missed Alarm



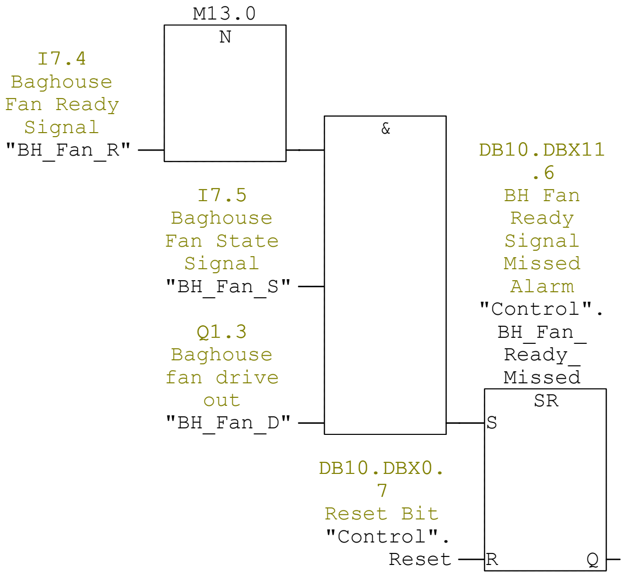
Network: 23 Baghouse Ready signal missed alarm



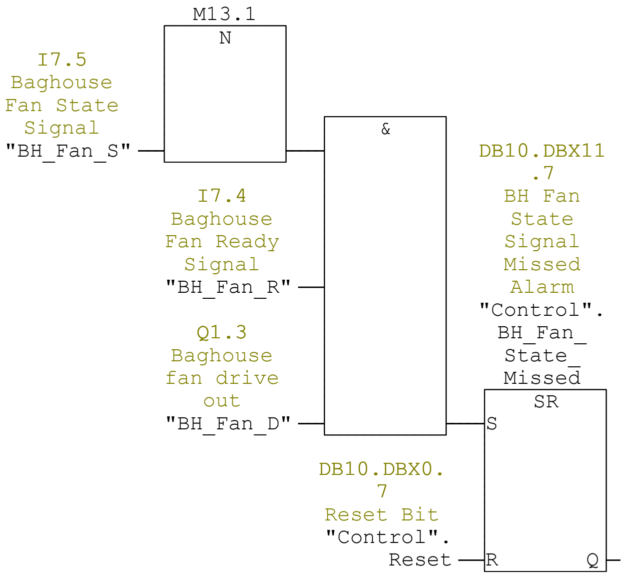
Network: 24 Baghouse State signal missed alarm



Network: 25 BH Fan Ready Signal Missed Alarm



Network: 26 BH Fan State Signal Missed Alarm



Network: 27 Apron Electrical Cabinet Alarm

```
      A     "RF01_A"                        I0.4              -- Rotary Feeder Alarm
      =     "Control".RF01_Electrical_Fault  DB10.DBX7.4       -- Rotary Feeder Electrical Cabinet Fault
      A     "BC02_A"                        I1.1              -- Belt Conveyor 2 Alarm
      =     "Control".BC02_Electrical_Fault  DB10.DBX7.7       -- Belt Conveyor 2 Electrical Cabinet Fault
      A     "BE01_A"                        I1.6              -- Bucket Elevator Alarm
      =     "Control".BE01_Electrical_Fault  DB10.DBX8.2       -- Bucket Elevator Electrical Cabinet Fault
      A     "BC03_A"                        I2.3              -- Belt Conveyor 3 Alarm
      =     "Control".BC03_Electrical_Fault  DB10.DBX8.5       -- Belt Conveyor 3 Electrical Cabinet Fault
      A     "BC01_A"                        I3.0              -- Belt Conveyor 1 Alarm
      =     "Control".BC01_Electrical_Fault  DB10.DBX9.0       -- Belt Conveyor 1 Electrical Cabinet Fault
      A     "WF01_A"                        I3.5              -- Weigh Feeder 1 Alarm
      =     "Control".WF01_Electrical_Fault  DB10.DBX9.3       -- Weigh Feeder 1 Electrical Cabinet Fault
      A     "WF02_A"                        I4.2              -- Weigh Feeder 2 Alarm
      =     "Control".WF02_Electrical_Fault  DB10.DBX9.6       -- Weigh Feeder 2 Electrical Cabinet Fault
      A     "WF03_A"                        I4.7              -- Weigh Feeder 3 Alarm
      =     "Control".WF03_Electrical_Fault  DB10.DBX10.1      -- Weigh Feeder 3 Electrical Cabinet Fault
      A     "WF04_A"                        I5.4              -- Weigh Feeder 4 Alarm
      =     "Control".WF04_Electrical_Fault  DB10.DBX10.4      -- Weigh Feeder 4 Electrical Cabinet Fault
      A     "VRM_Motor_A"                   I6.1              -- VRM Main Motor Alarm
      =     "Control".VRM_Motor_Elec_Fault  DB10.DBX10.7      -- VRM Main Motor Electrical Fault
      A     "Airslide_A"                    I6.6              -- Airslide Alarm
      =     "Control".Airslide_Electrical_FAul DB10.DBX11.2    -- Airslide Electrical FAult
      A     "Baghouse_A"                    I7.3              -- Baghouse Alarm
      =     "Control".Baghouse_Electrical_Faul DB10.DBX11.5    -- Baghouse Electrical FAult
      A     "BH_Fan_A"                      I8.0              -- Baghouse Fan Alarm
      =     "Control".BH_Fan_Electrical_Fault  DB10.DBX12.0    -- BH Fan Electrical Fault
```

FC1 - <offline>

"Data Sharing"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 06/12/2022 06:12:35 PM
Interface: 01/28/2022 03:26:22 PM
Lengths (block/logic/data): 00108 00014 00000

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC1

Network: 1 VRM BH Fan running State

```
      A     "Receive From FCS02".BH_Fan_Running_Cmd  DB4.DBX0.1          -- BH Fan running Cmd
      =     "Control".VRM_BH_Fan_Running_State       DB10.DBX13.7        -- VRM BH Fan running State
```

Network: 2

FC2 - <offline>

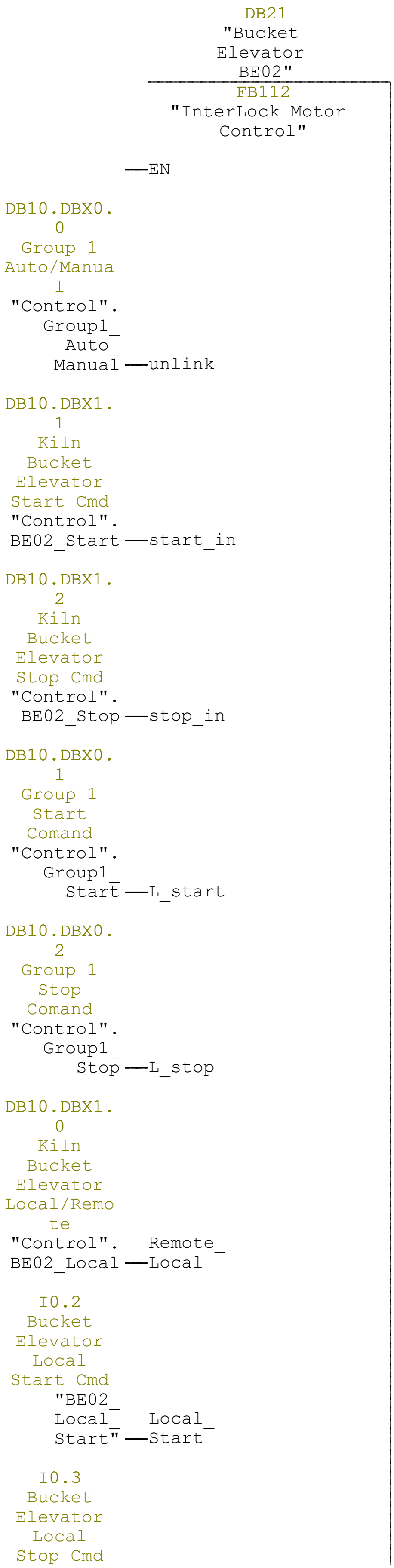
"Motors"

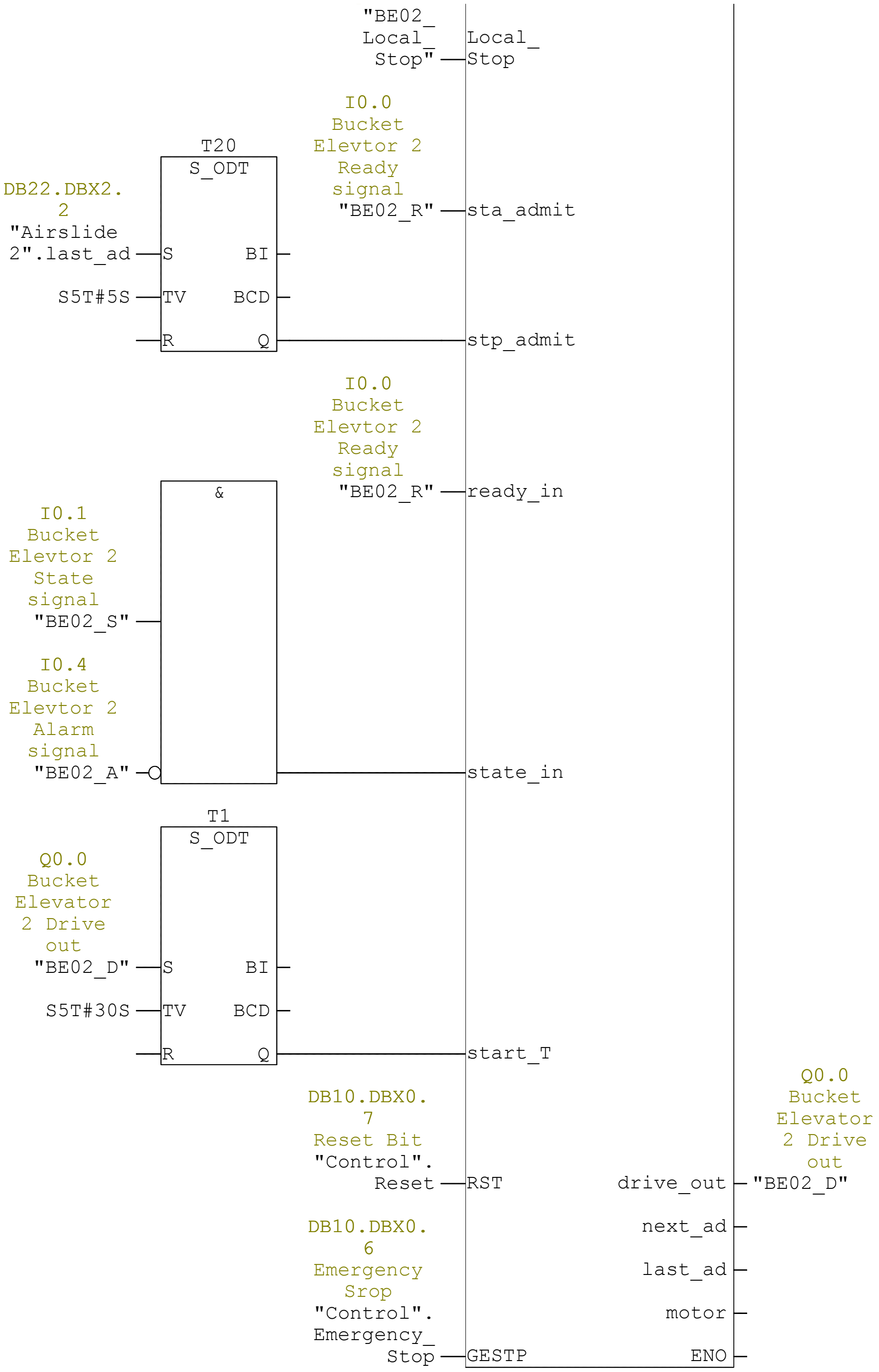
Name:		**Family:**	
Author:		**Version:**	0.1
		Block version:	2
Time stamp Code:	06/10/2022 04:50:39 PM		
Interface:	04/11/2022 03:12:13 AM		
Lengths (block/logic/data):	05534 05400 00008		

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

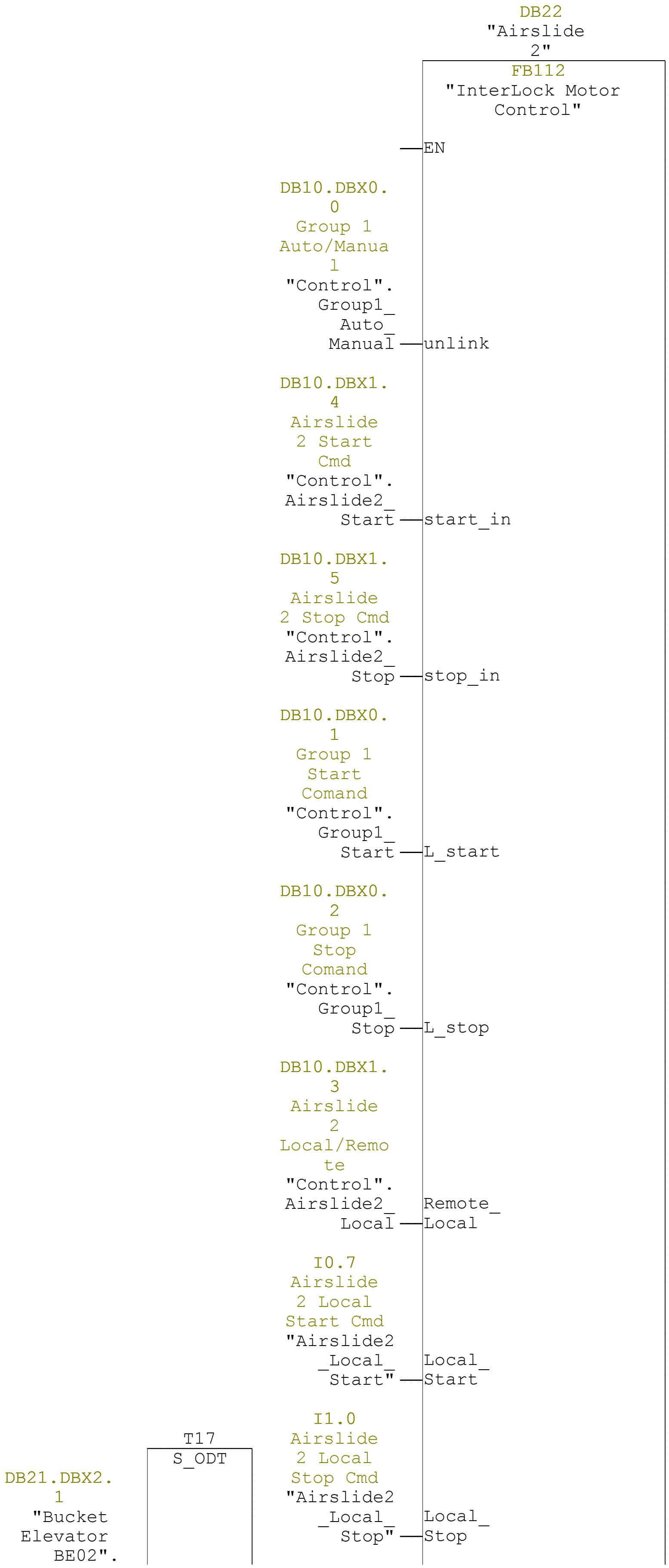
Block: FC2

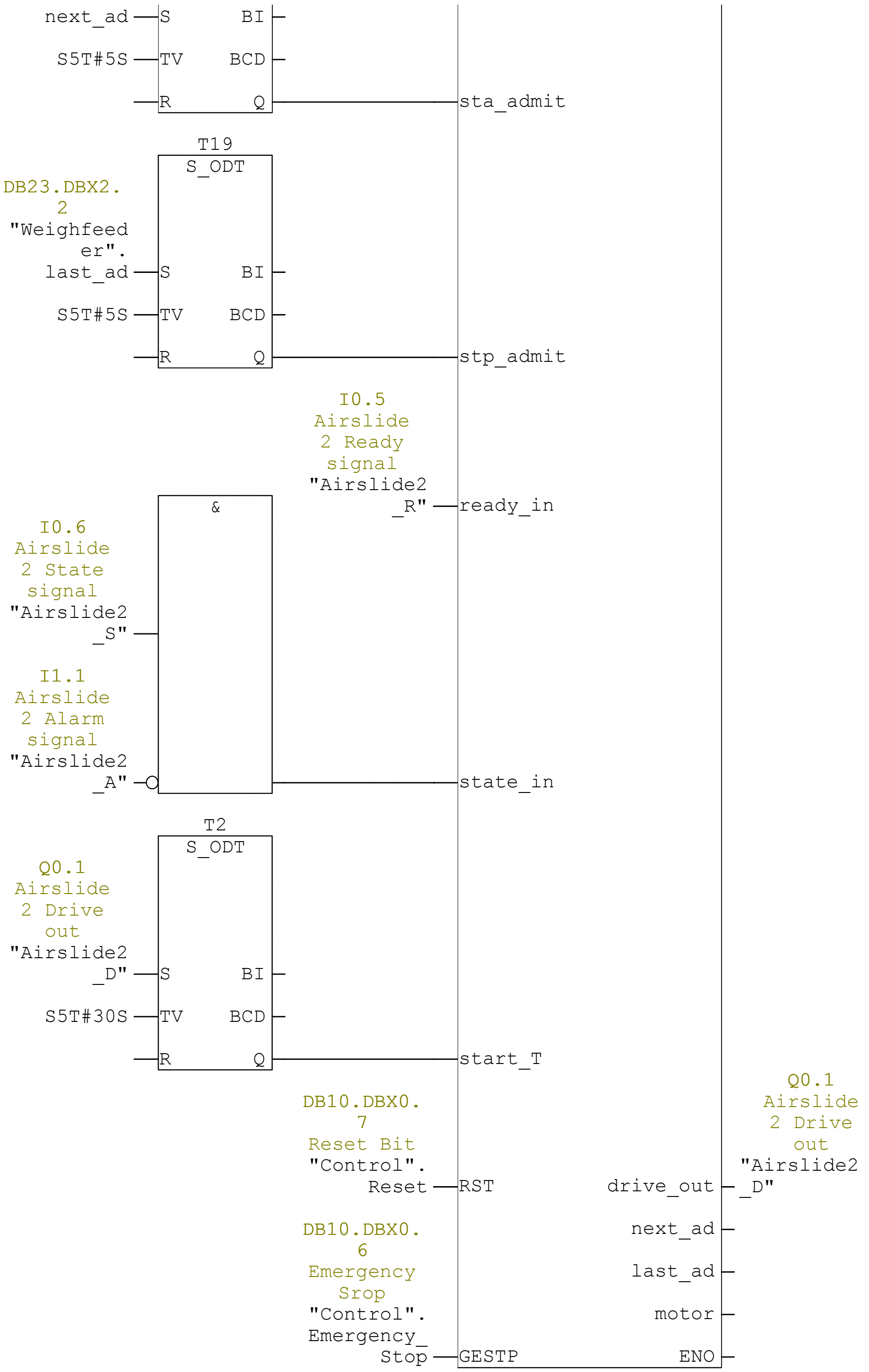
Network: 1


DB21
"Bucket Elevator BE02"
FB112
"InterLock Motor Control"
EN
DB10.DBX0.0
Group 1 Auto/Manual
"Control".Group1_Auto_Manual
unlink
DB10.DBX1.1
Kiln Bucket Elevator Start Cmd
"Control".BE02_Start
start_in
DB10.DBX1.2
Kiln Bucket Elevator Stop Cmd
"Control".BE02_Stop
stop_in
DB10.DBX0.1
Group 1 Start Comand
"Control".Group1_Start
L_start
DB10.DBX0.2
Group 1 Stop Comand
"Control".Group1_Stop
L_stop
DB10.DBX1.0
Kiln Bucket Elevator Local/Remote
"Control".BE02_Local
Remote_Local
I0.2
Bucket Elevator Local Start Cmd
"BE02_Local_Start"
Local_Start
I0.3
Bucket Elevator Local Stop Cmd


"BE02_
Local_
Stop"
Local_
Stop
I0.0
Bucket
Elevtor 2
Ready
signal
"BE02_R"
sta_admit
T20
S_ODT
DB22.DBX2.
2
"Airslide
2".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I0.0
Bucket
Elevtor 2
Ready
signal
"BE02_R"
ready_in
&
I0.1
Bucket
Elevtor 2
State
signal
"BE02_S"
I0.4
Bucket
Elevtor 2
Alarm
signal
"BE02_A"
state_in
T1
S_ODT
Q0.0
Bucket
Elevator
2 Drive
out
"BE02_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q0.0
Bucket
Elevator
2 Drive
out
"BE02_D"
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
next_ad
last_ad
motor
ENO

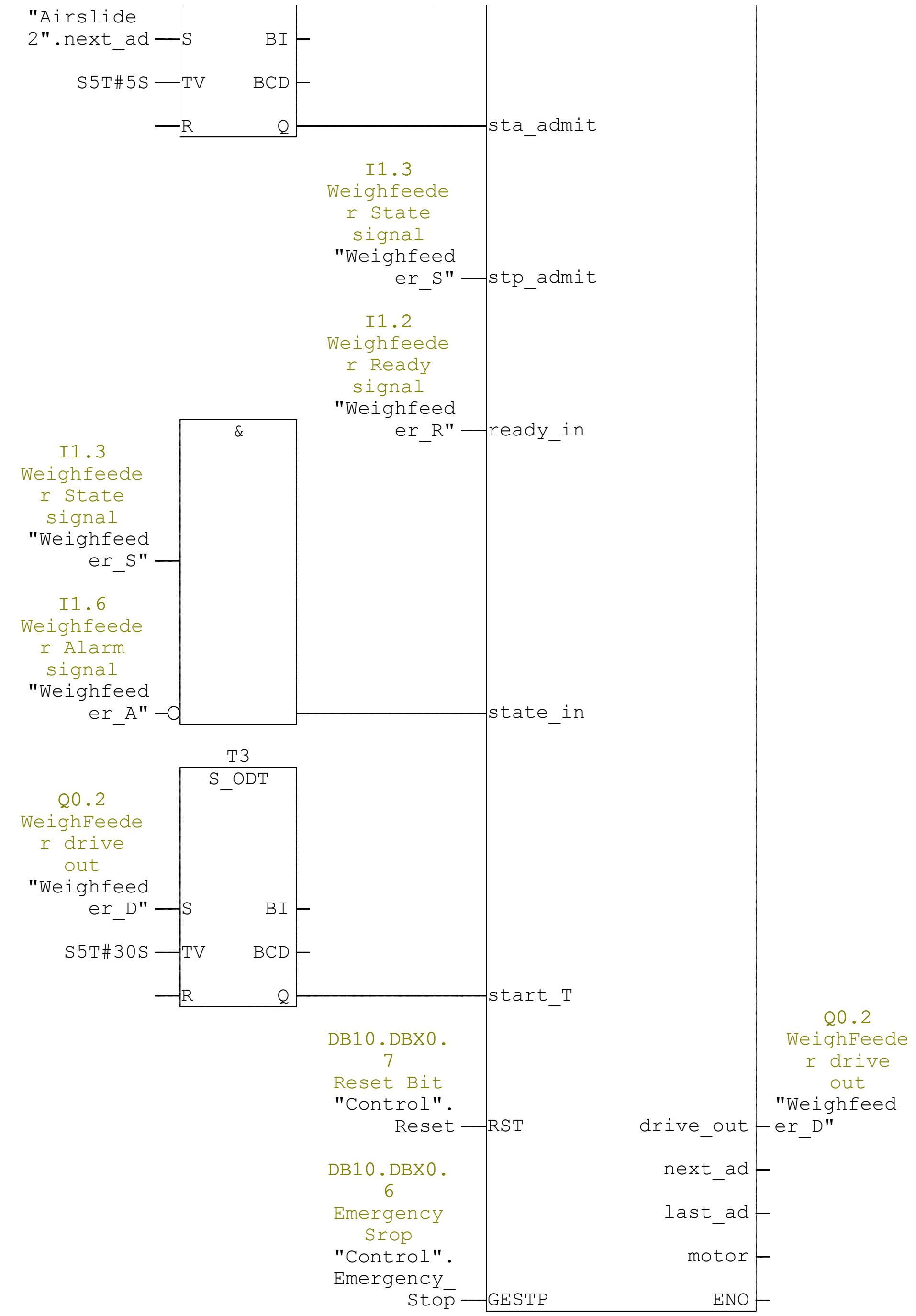
Network: 2


DB22
"Airslide 2"
FB112
"InterLock Motor Control"
EN
DB10.DBX0.0
Group 1 Auto/Manual
"Control".Group1_Auto_Manual
unlink
DB10.DBX1.4
Airslide 2 Start Cmd
"Control".Airslide2_Start
start_in
DB10.DBX1.5
Airslide 2 Stop Cmd
"Control".Airslide2_Stop
stop_in
DB10.DBX0.1
Group 1 Start Comand
"Control".Group1_Start
L_start
DB10.DBX0.2
Group 1 Stop Comand
"Control".Group1_Stop
L_stop
DB10.DBX1.3
Airslide 2 Local/Remote
"Control".Airslide2_Local
Remote_Local
I0.7
Airslide 2 Local Start Cmd
"Airslide2_Local_Start"
Local_Start
I1.0
Airslide 2 Local Stop Cmd
"Airslide2_Local_Stop"
Local_Stop
T17
S_ODT
DB21.DBX2.1
"Bucket Elevator BE02".

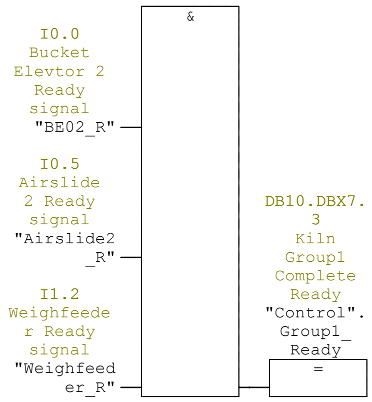

next_ad
S
BI
S5T#5S
TV
BCD
R
Q
sta_admit
T19
S_ODT
DB23.DBX2.
2
"Weighfeed
er".
last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I0.5
Airslide
2 Ready
signal
"Airslide2
_R"
ready_in
&
I0.6
Airslide
2 State
signal
"Airslide2
_S"
I1.1
Airslide
2 Alarm
signal
"Airslide2
_A"
state_in
T2
S_ODT
Q0.1
Airslide
2 Drive
out
"Airslide2
_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q0.1
Airslide
2 Drive
out
"Airslide2
_D"
next_ad
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
last_ad
motor
ENO

Network: 3

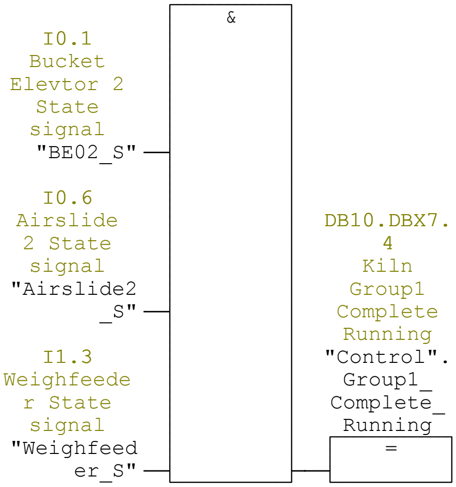
```
                                                 DB23
                                              "Weighfeed
                                                 er"
                                        +----------------------+
                                        |        FB112         |
                                        |  "InterLock Motor    |
                                        |       Control"       |
                                        |                      |
                                      --|EN                    |
                   DB10.DBX0.           |                      |
                       0                |                      |
                    Group 1             |                      |
                   Auto/Manua           |                      |
                       l                |                      |
                   "Control".           |                      |
                      Group1_           |                      |
                        Auto_           |                      |
                       Manual---------- |unlink                |
                   DB10.DBX1.           |                      |
                       7                |                      |
                     Weigh              |                      |
                     Feeder             |                      |
                   Start Cmd            |                      |
                   "Control".           |                      |
                   WeighFeede           |                      |
                      r_Start---------- |start_in              |
                   DB10.DBX2.           |                      |
                       0                |                      |
                     Weigh              |                      |
                     Feeder             |                      |
                    Stop Cmd            |                      |
                   "Control".           |                      |
                   WeighFeede           |                      |
                       r_Stop---------- |stop_in               |
                   DB10.DBX0.           |                      |
                       1                |                      |
                    Group 1             |                      |
                     Start              |                      |
                    Comand              |                      |
                   "Control".           |                      |
                      Group1_           |                      |
                        Start---------- |L_start               |
                   DB10.DBX0.           |                      |
                       2                |                      |
                    Group 1             |                      |
                     Stop               |                      |
                    Comand              |                      |
                   "Control".           |                      |
                      Group1_           |                      |
                         Stop---------- |L_stop                |
                   DB10.DBX1.           |                      |
                       6                |                      |
                     Weigh              |                      |
                     Feeder             |                      |
                   Local/Remo           |                      |
                       te               |                      |
                   "Control".           |                      |
                   WeighFeede           |Remote_               |
                      r_Local---------- |Local                 |
                      I1.4              |                      |
                   Weighfeede           |                      |
                    r Local             |                      |
                   Start Cmd            |                      |
                   "Weighfeed           |                      |
                    er_Local_           |Local_                |
                       Start"---------- |Start                 |
                      I1.5              |                      |
                   Weighfeede           |                      |
                    r Local             |                      |
            T18     Stop Cmd            |                      |
         +-------+                      |                      |
         | S_ODT |                      |                      |
DB22.DBX2.|      | "Weighfeed           |                      |
    1    |       |  er_Local_           |Local_                |
         |       |       Stop"--------- |Stop                  |
```



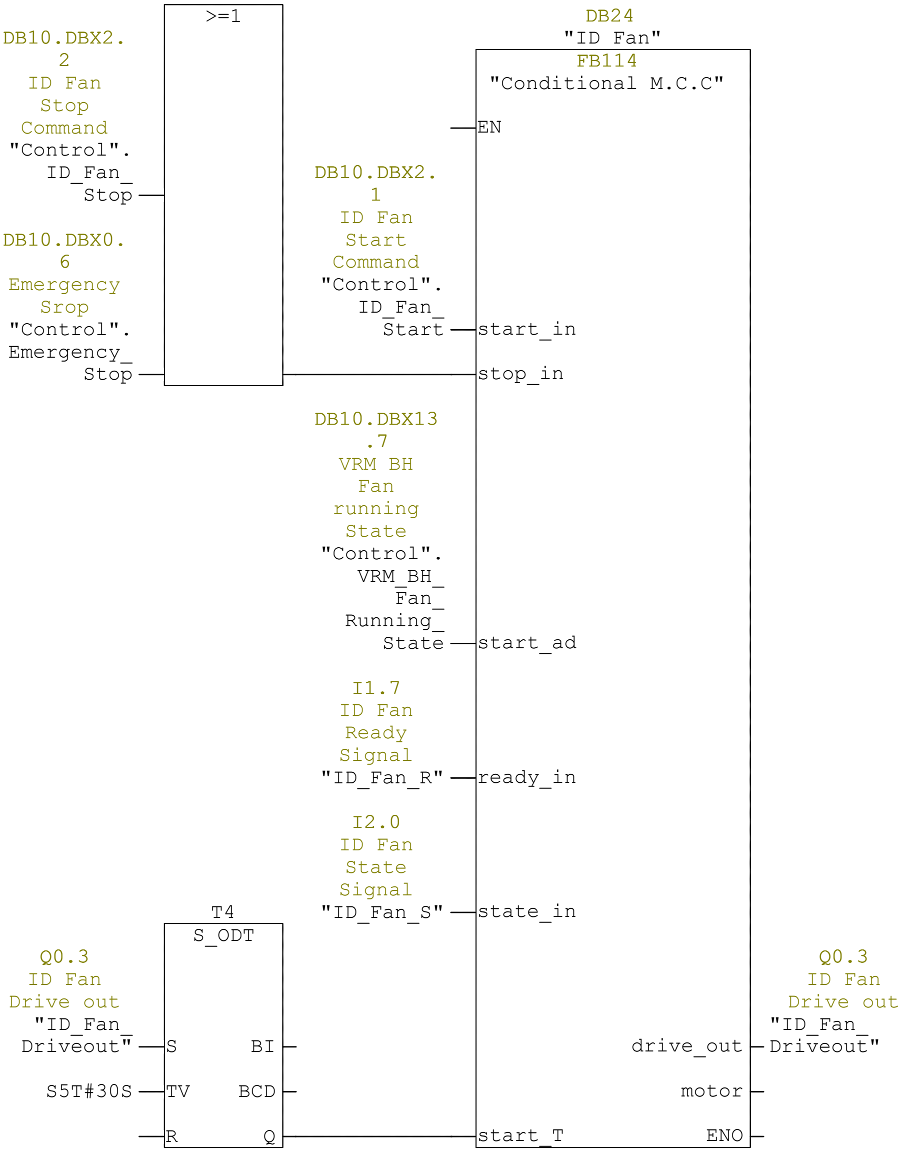
Network: 4 Kiln Group1 Complete Ready



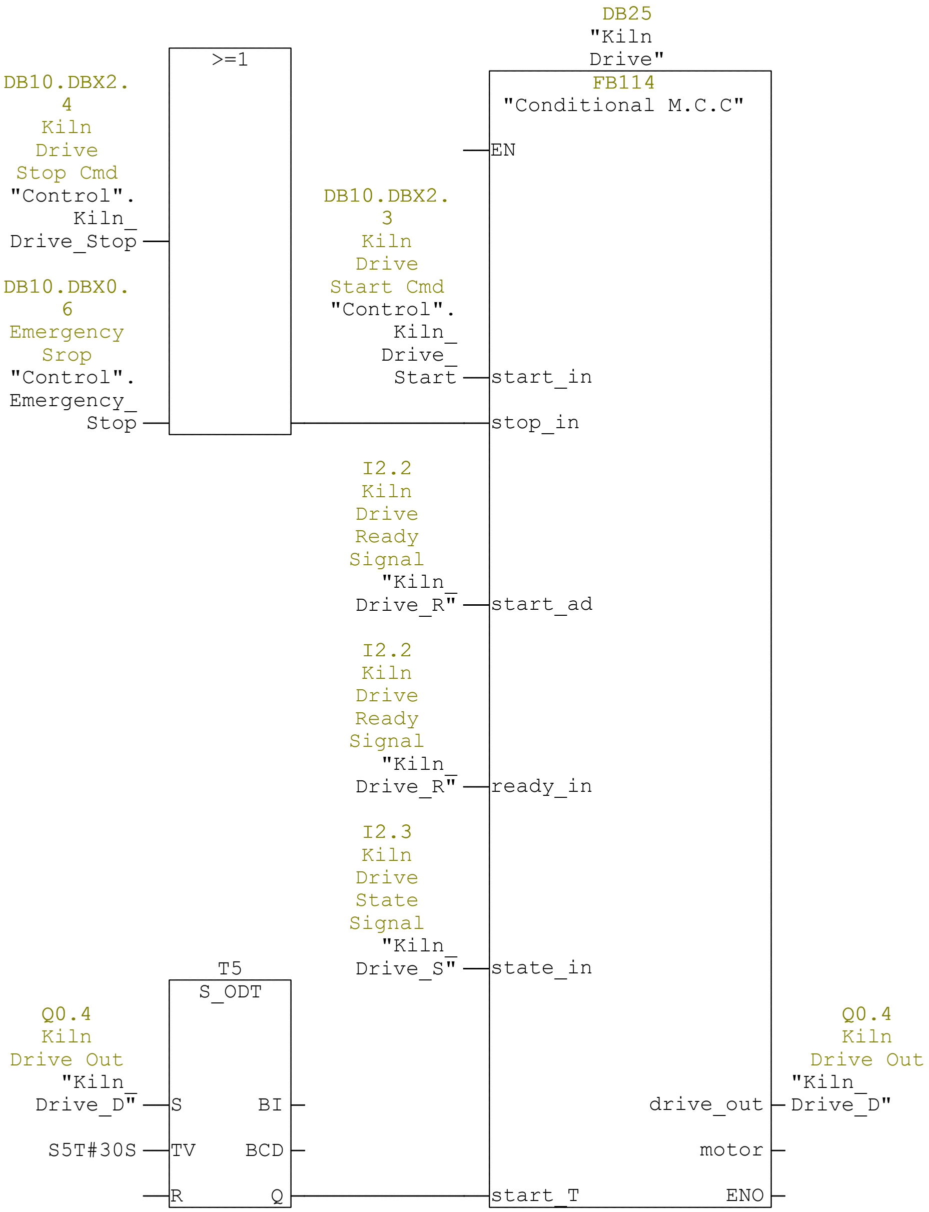
Network: 5 Kiln Group1 Complete Running



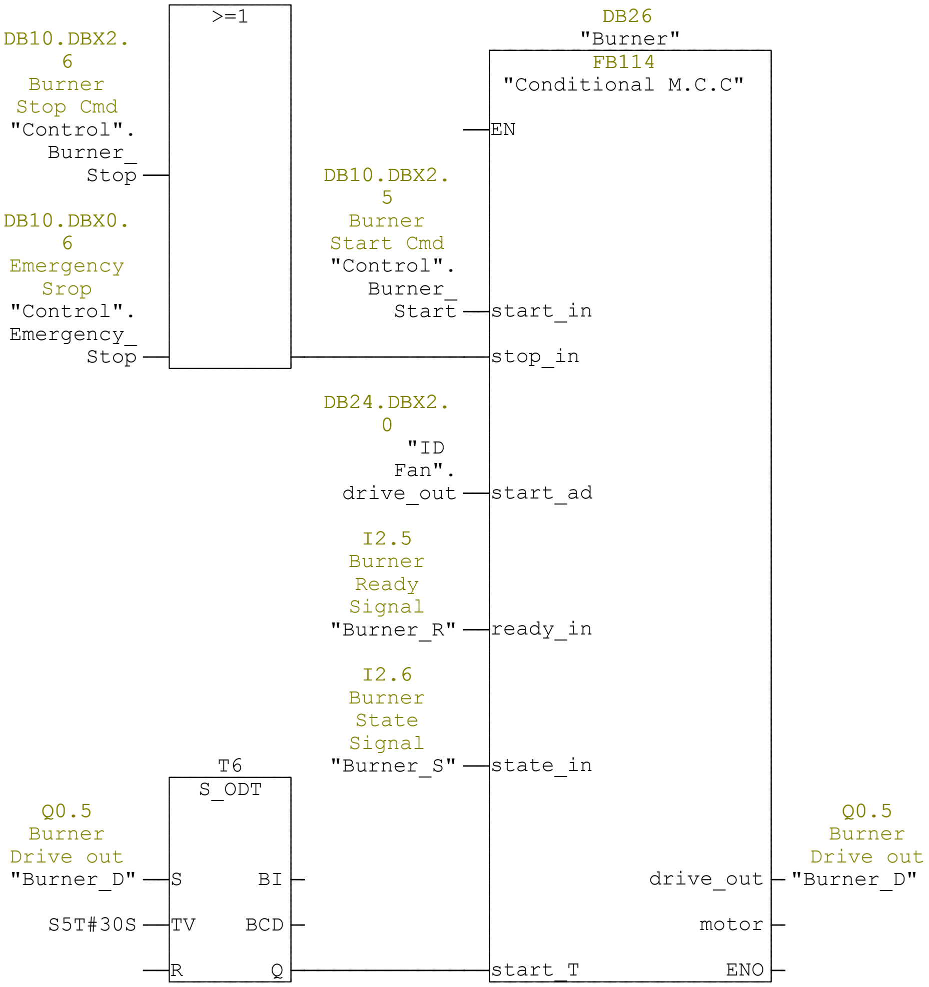
Network: 6



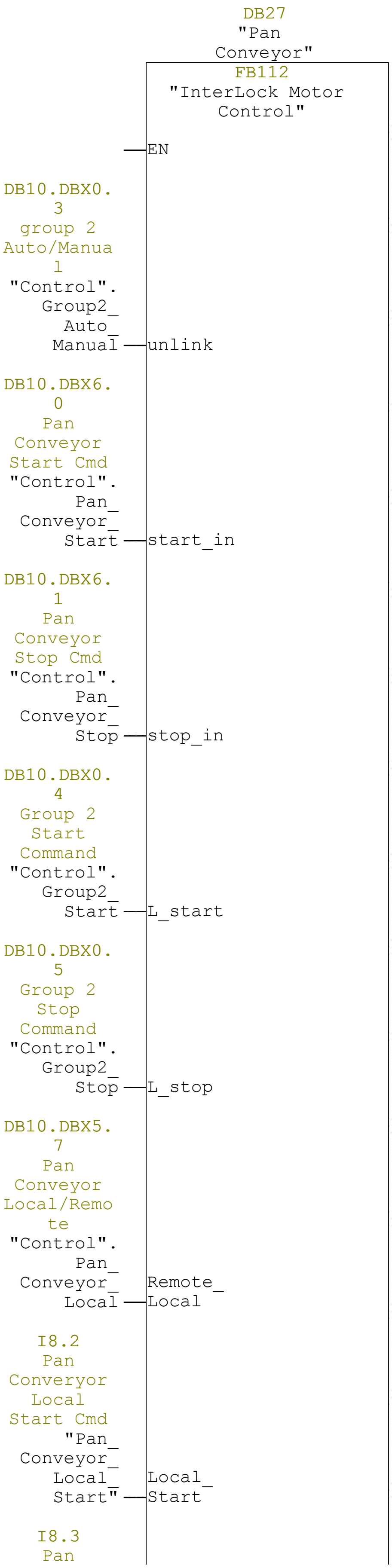
Network: 7


DB25
"Kiln
Drive"
FB114
"Conditional M.C.C"
EN
>=1
DB10.DBX2.
4
Kiln
Drive
Stop Cmd
"Control".
Kiln_
Drive_Stop
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
DB10.DBX2.
3
Kiln
Drive
Start Cmd
"Control".
Kiln_
Drive_
Start
start_in
stop_in
I2.2
Kiln
Drive
Ready
Signal
"Kiln_
Drive_R"
start_ad
I2.2
Kiln
Drive
Ready
Signal
"Kiln_
Drive_R"
ready_in
I2.3
Kiln
Drive
State
Signal
"Kiln_
Drive_S"
state_in
T5
S_ODT
Q0.4
Kiln
Drive Out
"Kiln_
Drive_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
drive_out
motor
ENO
Q0.4
Kiln
Drive Out
"Kiln_
Drive_D"

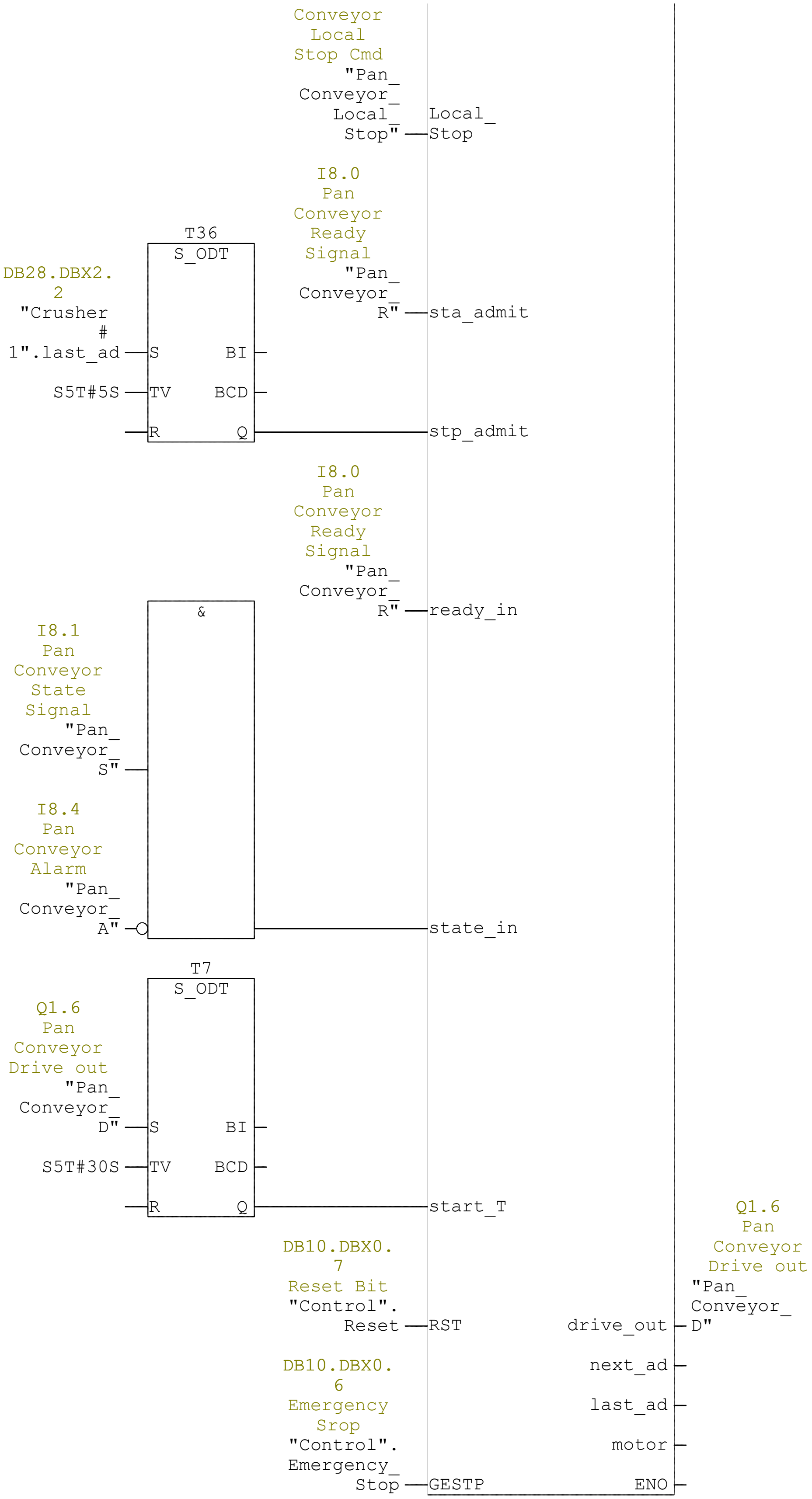
Network: 8


>=1
DB10.DBX2.6
Burner Stop Cmd
"Control".Burner_Stop
DB10.DBX0.6
Emergency Srop
"Control".Emergency_Stop
DB26
"Burner"
FB114
"Conditional M.C.C"
EN
DB10.DBX2.5
Burner Start Cmd
"Control".Burner_Start
start_in
stop_in
DB24.DBX2.0
"ID Fan".drive_out
start_ad
I2.5
Burner Ready Signal
"Burner_R"
ready_in
I2.6
Burner State Signal
"Burner_S"
state_in
T6
S_ODT
Q0.5
Burner Drive out
"Burner_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
drive_out
motor
ENO
Q0.5
Burner Drive out
"Burner_D"

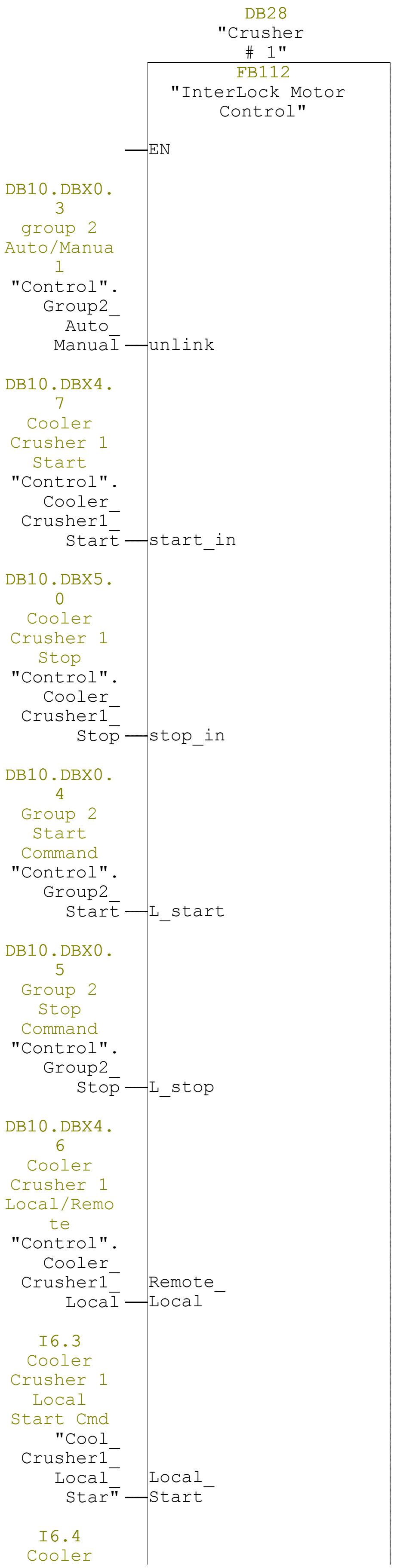
Network: 9



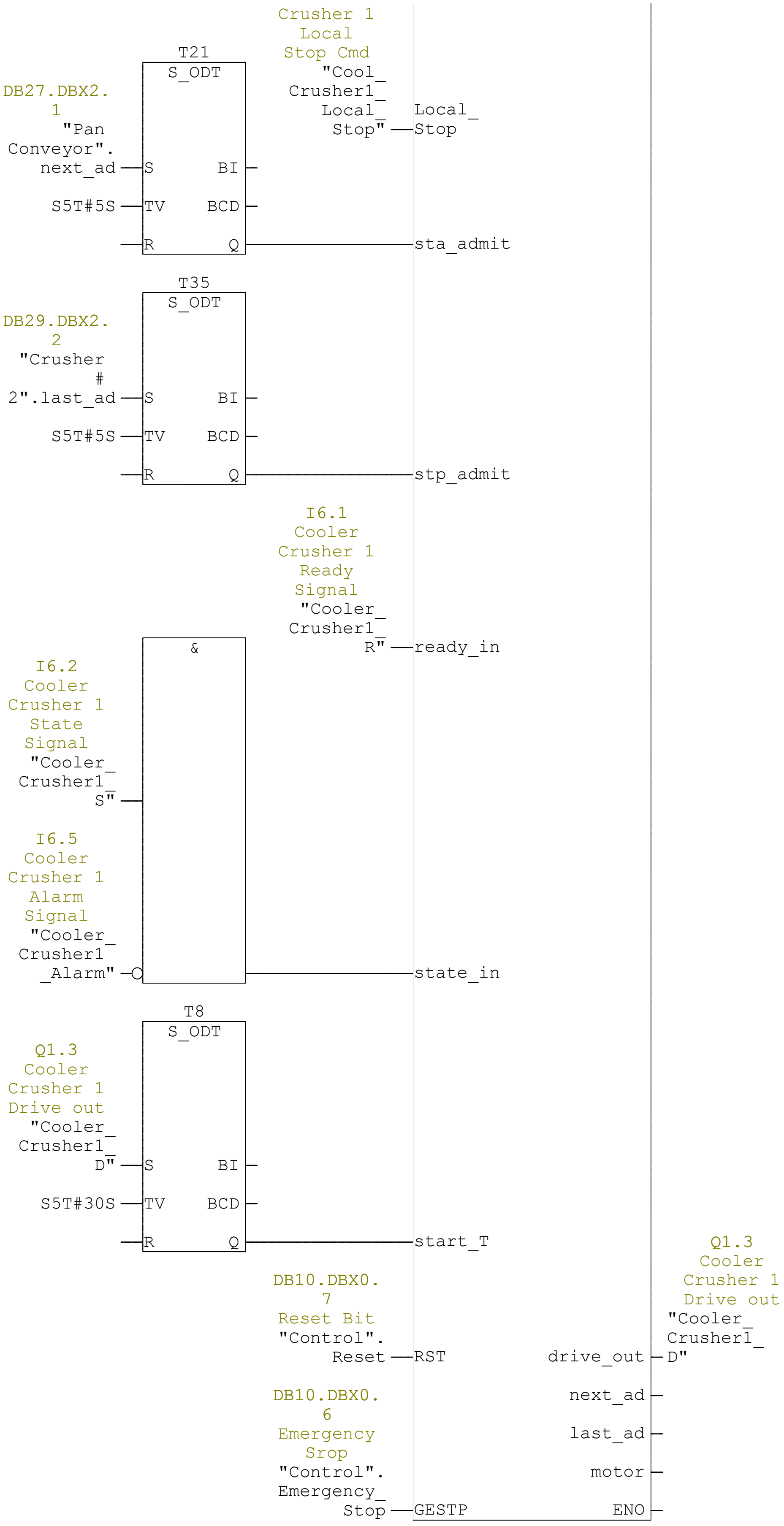
```
                        DB27
                        "Pan
                      Conveyor"
                   +-----FB112------+
                   |"InterLock Motor|
                   |    Control"    |
                 --|EN              |
DB10.DBX0.3        |                |
group 2            |                |
Auto/Manual        |                |
"Control".Group2_Auto_Manual --|unlink
DB10.DBX6.0        |                |
Pan Conveyor Start Cmd              |
"Control".Pan_Conveyor_Start --|start_in
DB10.DBX6.1        |                |
Pan Conveyor Stop Cmd               |
"Control".Pan_Conveyor_Stop --|stop_in
DB10.DBX0.4        |                |
Group 2 Start Command               |
"Control".Group2_Start --|L_start
DB10.DBX0.5        |                |
Group 2 Stop Command                |
"Control".Group2_Stop --|L_stop
DB10.DBX5.7        |                |
Pan Conveyor Local/Remote           |
"Control".Pan_Conveyor_Local --|Remote_Local
I8.2               |                |
Pan Converyor Local Start Cmd       |
"Pan_Conveyor_Local_Start" --|Local_Start
I8.3               |                |
Pan                |                |
```


Conveyor Local Stop Cmd
"Pan_ Conveyor_ Local_ Stop"
Local_ Stop
I8.0 Pan Conveyor Ready Signal
"Pan_ Conveyor_ R"
sta_admit
T36
S_ODT
DB28.DBX2. 2
"Crusher # 1".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I8.0 Pan Conveyor Ready Signal
"Pan_ Conveyor_ R"
ready_in
&
I8.1 Pan Conveyor State Signal
"Pan_ Conveyor_ S"
I8.4 Pan Conveyor Alarm
"Pan_ Conveyor_ A"
state_in
T7
S_ODT
Q1.6 Pan Conveyor Drive out
"Pan_ Conveyor_ D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
Q1.6 Pan Conveyor Drive out
"Pan_ Conveyor_ D"
DB10.DBX0. 7
Reset Bit
"Control". Reset
RST
drive_out
next_ad
DB10.DBX0. 6
Emergency Srop
last_ad
"Control". Emergency_ Stop
motor
GESTP
ENO

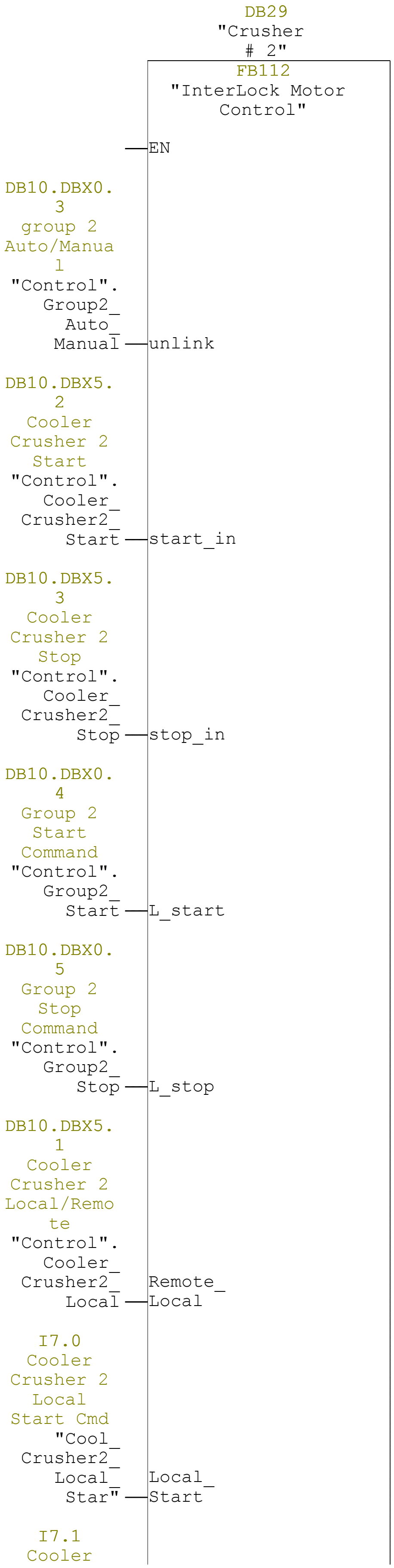
Network: 10



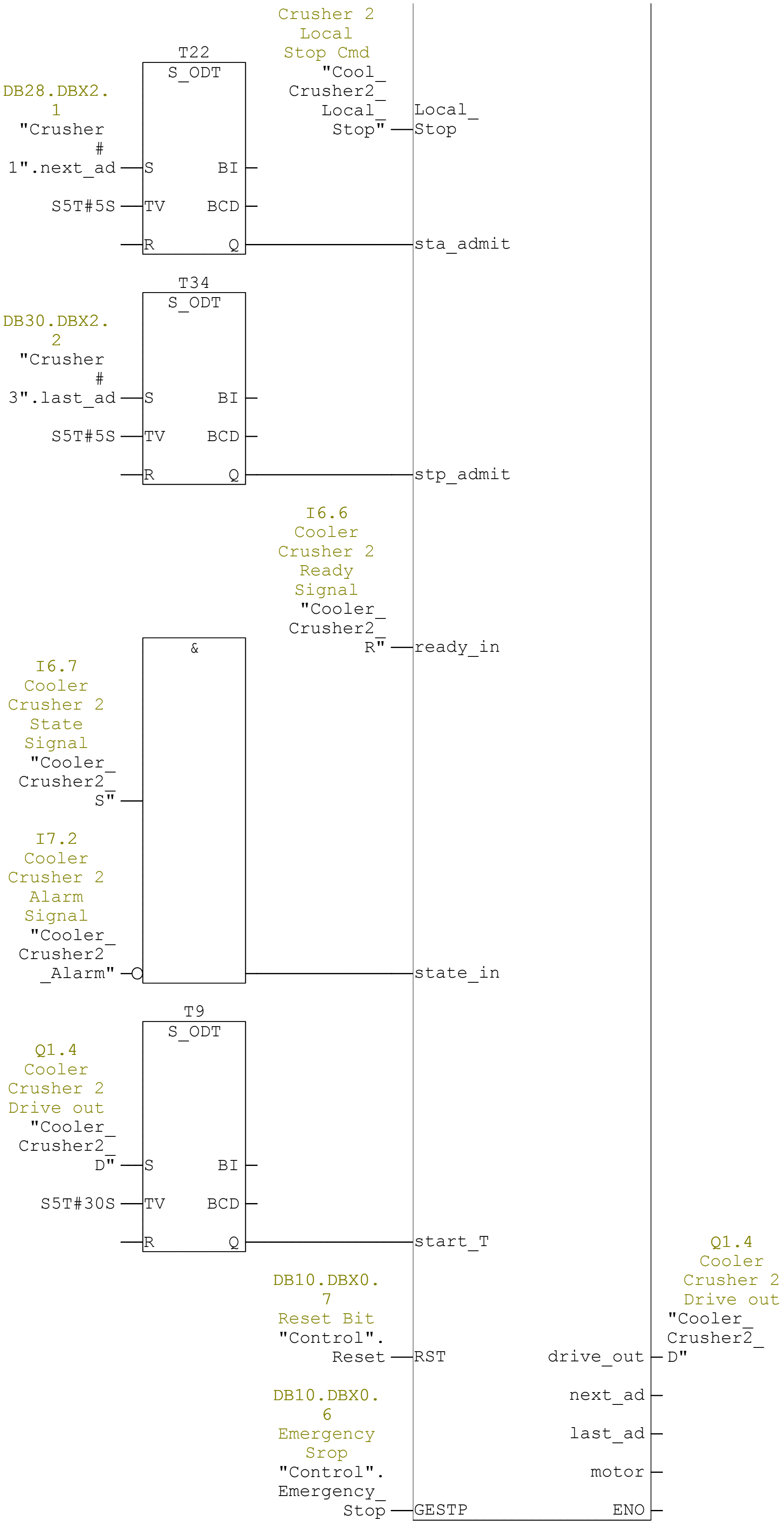
```
                     DB28
                  "Crusher
                    # 1"
                ┌───FB112──────┐
                │"InterLock Motor│
                │   Control"    │
                │               │
              ──┤EN             │
                │               │
DB10.DBX0.      │               │
    3           │               │
  group 2       │               │
Auto/Manua      │               │
    l           │               │
"Control".      │               │
   Group2_      │               │
     Auto_      │               │
    Manual ─────┤unlink         │
                │               │
DB10.DBX4.      │               │
    7           │               │
  Cooler        │               │
Crusher 1       │               │
  Start         │               │
"Control".      │               │
   Cooler_      │               │
 Crusher1_      │               │
     Start ─────┤start_in       │
                │               │
DB10.DBX5.      │               │
    0           │               │
  Cooler        │               │
Crusher 1       │               │
   Stop         │               │
"Control".      │               │
   Cooler_      │               │
 Crusher1_      │               │
      Stop ─────┤stop_in        │
                │               │
DB10.DBX0.      │               │
    4           │               │
  Group 2       │               │
   Start        │               │
  Command       │               │
"Control".      │               │
   Group2_      │               │
     Start ─────┤L_start        │
                │               │
DB10.DBX0.      │               │
    5           │               │
  Group 2       │               │
   Stop         │               │
  Command       │               │
"Control".      │               │
   Group2_      │               │
      Stop ─────┤L_stop         │
                │               │
DB10.DBX4.      │               │
    6           │               │
  Cooler        │               │
Crusher 1       │               │
Local/Remo      │               │
    te          │               │
"Control".      │               │
   Cooler_      │               │
 Crusher1_      │Remote_        │
     Local ─────┤Local          │
                │               │
   I6.3         │               │
  Cooler        │               │
Crusher 1       │               │
   Local        │               │
 Start Cmd      │               │
    "Cool_      │               │
 Crusher1_      │               │
    Local_      │Local_         │
     Star" ─────┤Start          │
                │               │
   I6.4         │               │
  Cooler        │               │
```


Crusher 1
Local
Stop Cmd
"Cool_
Crusher1_
Local_
Stop"
Local_
Stop
T21
S_ODT
DB27.DBX2.
1
"Pan
Conveyor".
next_ad
S
BI
S5T#5S
TV
BCD
R
Q
sta_admit
T35
S_ODT
DB29.DBX2.
2
"Crusher
#
2".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I6.1
Cooler
Crusher 1
Ready
Signal
"Cooler_
Crusher1_
R"
ready_in
&
I6.2
Cooler
Crusher 1
State
Signal
"Cooler_
Crusher1_
S"
I6.5
Cooler
Crusher 1
Alarm
Signal
"Cooler_
Crusher1
_Alarm"
state_in
T8
S_ODT
Q1.3
Cooler
Crusher 1
Drive out
"Cooler_
Crusher1_
D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
Q1.3
Cooler
Crusher 1
Drive out
"Cooler_
Crusher1_
D"
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
next_ad
last_ad
motor
GESTP
ENO

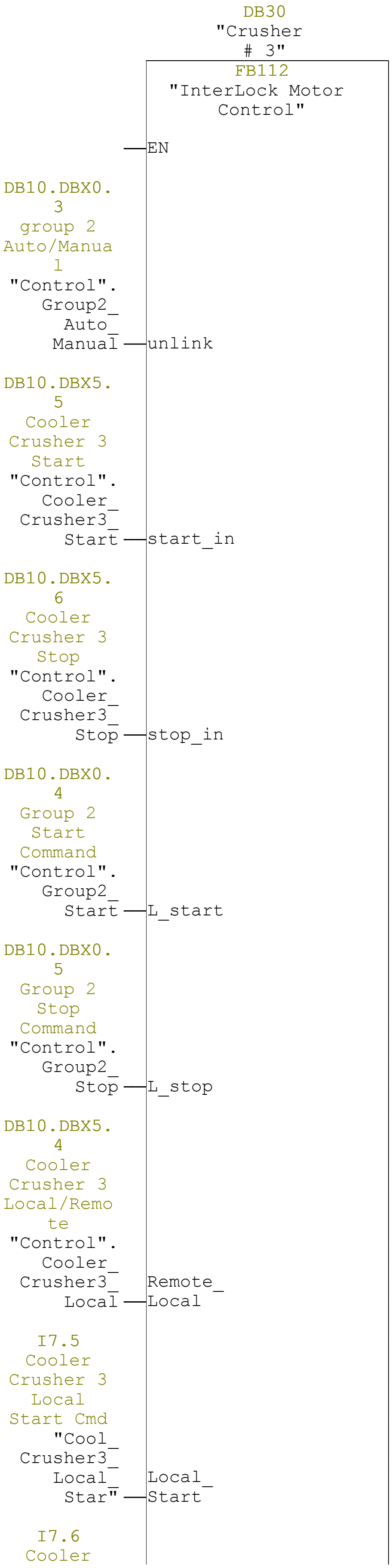
Network: 11



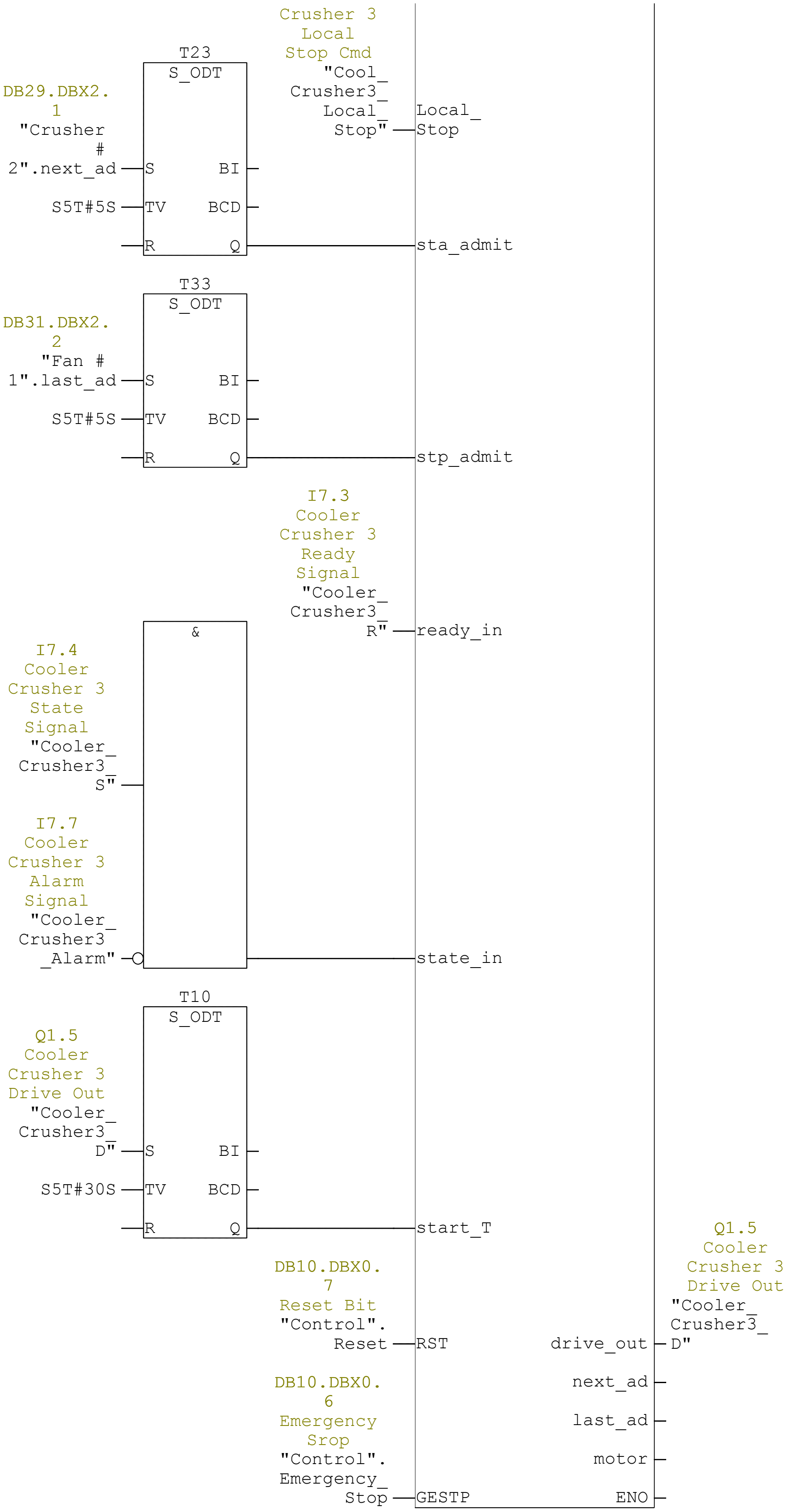
```
                        DB29
                      "Crusher
                        # 2"
                    ____________
                   |   FB112    |
                   |"InterLock Motor
                   |  Control"  |
                 --|EN          |
DB10.DBX0.         |            |
   3               |            |
 group 2           |            |
Auto/Manua         |            |
   l               |            |
"Control".         |            |
   Group2_         |            |
     Auto_         |            |
    Manual --------|unlink      |
DB10.DBX5.         |            |
   2               |            |
 Cooler            |            |
Crusher 2          |            |
  Start            |            |
"Control".         |            |
   Cooler_         |            |
 Crusher2_         |            |
     Start --------|start_in    |
DB10.DBX5.         |            |
   3               |            |
 Cooler            |            |
Crusher 2          |            |
  Stop             |            |
"Control".         |            |
   Cooler_         |            |
 Crusher2_         |            |
      Stop --------|stop_in     |
DB10.DBX0.         |            |
   4               |            |
 Group 2           |            |
  Start            |            |
 Command           |            |
"Control".         |            |
   Group2_         |            |
     Start --------|L_start     |
DB10.DBX0.         |            |
   5               |            |
 Group 2           |            |
  Stop             |            |
 Command           |            |
"Control".         |            |
   Group2_         |            |
      Stop --------|L_stop      |
DB10.DBX5.         |            |
   1               |            |
 Cooler            |            |
Crusher 2          |            |
Local/Remo         |            |
   te              |            |
"Control".         |            |
   Cooler_         |            |
 Crusher2_         |Remote_     |
     Local --------|Local       |
  I7.0             |            |
 Cooler            |            |
Crusher 2          |            |
  Local            |            |
Start Cmd          |            |
    "Cool_         |            |
 Crusher2_         |            |
    Local_         |Local_      |
     Star" --------|Start       |
  I7.1             |            |
 Cooler            |            |
```


T22
S_ODT
DB28.DBX2.1
"Crusher #1".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
Crusher 2 Local Stop Cmd
"Cool_Crusher2_Local_Stop"
Local_Stop
sta_admit
T34
S_ODT
DB30.DBX2.2
"Crusher #3".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I16.6
Cooler Crusher 2 Ready Signal
"Cooler_Crusher2_R"
ready_in
&
I16.7
Cooler Crusher 2 State Signal
"Cooler_Crusher2_S"
I7.2
Cooler Crusher 2 Alarm Signal
"Cooler_Crusher2_Alarm"
state_in
T9
S_ODT
Q1.4
Cooler Crusher 2 Drive out
"Cooler_Crusher2_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.7
Reset Bit
"Control".Reset
RST
drive_out
Q1.4
Cooler Crusher 2 Drive out
"Cooler_Crusher2_D"
next_ad
DB10.DBX0.6
Emergency Srop
"Control".Emergency_Stop
last_ad
motor
GESTP
ENO

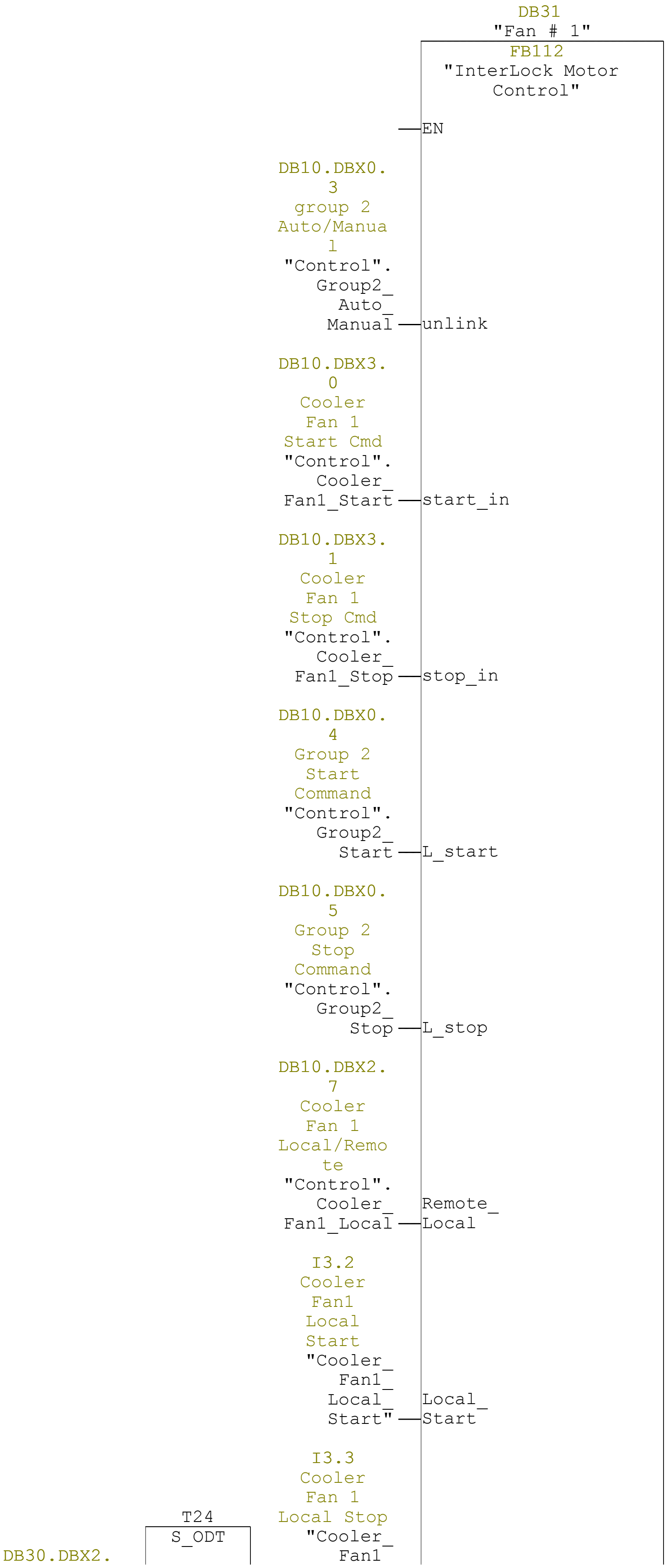
Network: 12



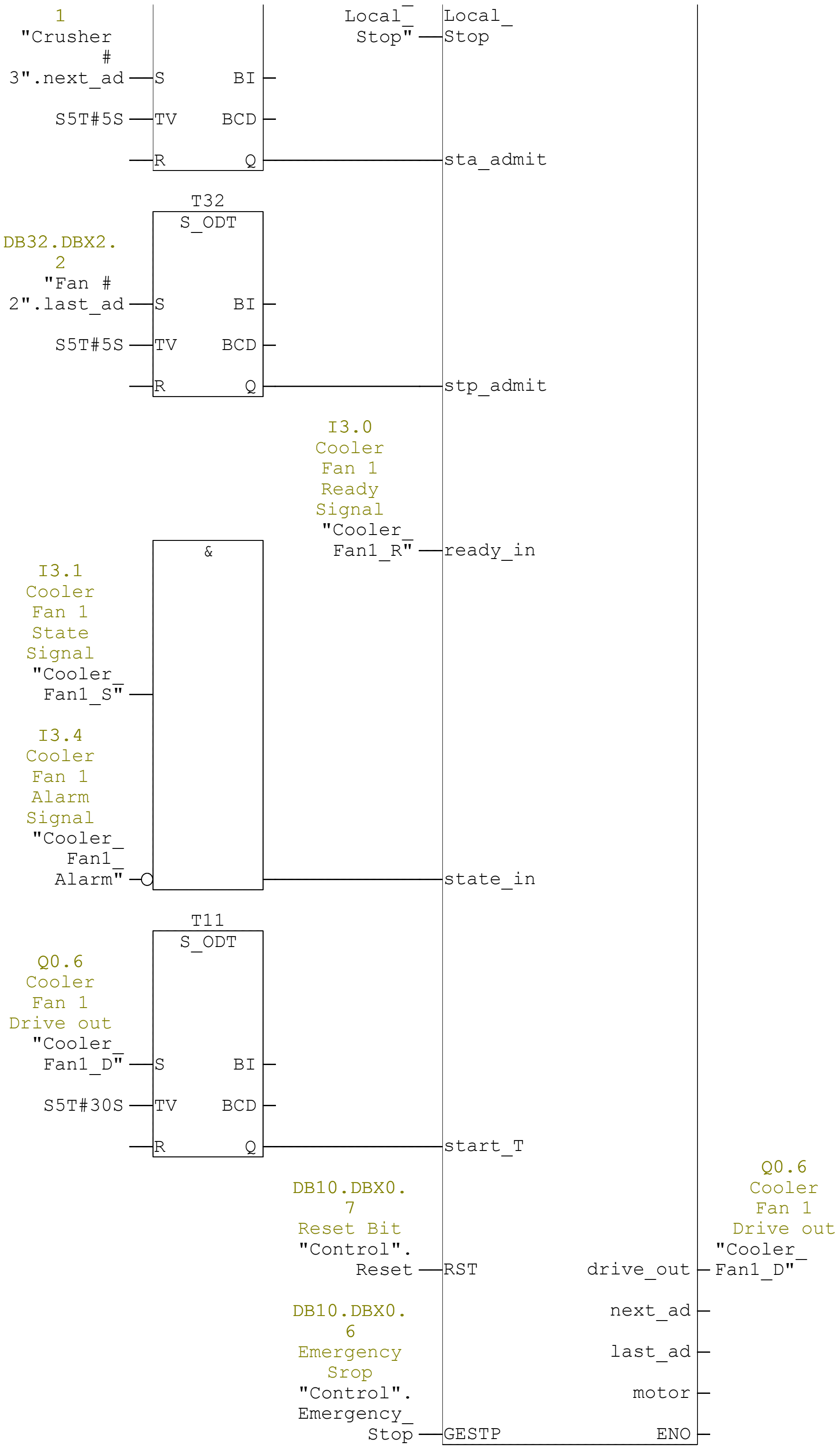
```
                     DB30
                   "Crusher
                     # 3"
              ┌──────────────────┐
              │      FB112       │
              │"InterLock Motor  │
              │    Control"      │
              │                  │
            ──┤EN                │
              │                  │
DB10.DBX0.    │                  │
    3         │                  │
  group 2     │                  │
Auto/Manua    │                  │
    l         │                  │
"Control".    │                  │
   Group2_    │                  │
     Auto_    │                  │
    Manual ───┤unlink            │
              │                  │
DB10.DBX5.    │                  │
    5         │                  │
  Cooler      │                  │
Crusher 3     │                  │
  Start       │                  │
"Control".    │                  │
   Cooler_    │                  │
 Crusher3_    │                  │
     Start ───┤start_in          │
              │                  │
DB10.DBX5.    │                  │
    6         │                  │
  Cooler      │                  │
Crusher 3     │                  │
   Stop       │                  │
"Control".    │                  │
   Cooler_    │                  │
 Crusher3_    │                  │
      Stop ───┤stop_in           │
              │                  │
DB10.DBX0.    │                  │
    4         │                  │
  Group 2     │                  │
   Start      │                  │
  Command     │                  │
"Control".    │                  │
   Group2_    │                  │
     Start ───┤L_start           │
              │                  │
DB10.DBX0.    │                  │
    5         │                  │
  Group 2     │                  │
   Stop       │                  │
  Command     │                  │
"Control".    │                  │
   Group2_    │                  │
      Stop ───┤L_stop            │
              │                  │
DB10.DBX5.    │                  │
    4         │                  │
  Cooler      │                  │
Crusher 3     │                  │
Local/Remo    │                  │
    te        │                  │
"Control".    │                  │
   Cooler_    │                  │
 Crusher3_    │Remote_           │
     Local ───┤Local             │
              │                  │
   I7.5       │                  │
  Cooler      │                  │
Crusher 3     │                  │
  Local       │                  │
Start Cmd     │                  │
    "Cool_    │                  │
 Crusher3_    │                  │
    Local_    │Local_            │
     Star" ───┤Start             │
              │                  │
   I7.6       │                  │
  Cooler      │                  │
```


Crusher 3 Local Stop Cmd
"Cool_Crusher3_Local_Stop"
Local_Stop
T23
S_ODT
DB29.DBX2.1
"Crusher #2".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
sta_admit
T33
S_ODT
DB31.DBX2.2
"Fan #1".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I7.3 Cooler Crusher 3 Ready Signal
"Cooler_Crusher3_R"
ready_in
&
I7.4 Cooler Crusher 3 State Signal
"Cooler_Crusher3_S"
I7.7 Cooler Crusher 3 Alarm Signal
"Cooler_Crusher3_Alarm"
state_in
T10
S_ODT
Q1.5 Cooler Crusher 3 Drive Out
"Cooler_Crusher3_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.7 Reset Bit
"Control".Reset
RST
drive_out
Q1.5 Cooler Crusher 3 Drive Out
"Cooler_Crusher3_D"
DB10.DBX0.6 Emergency Srop
"Control".Emergency_Stop
next_ad
last_ad
motor
GESTP
ENO

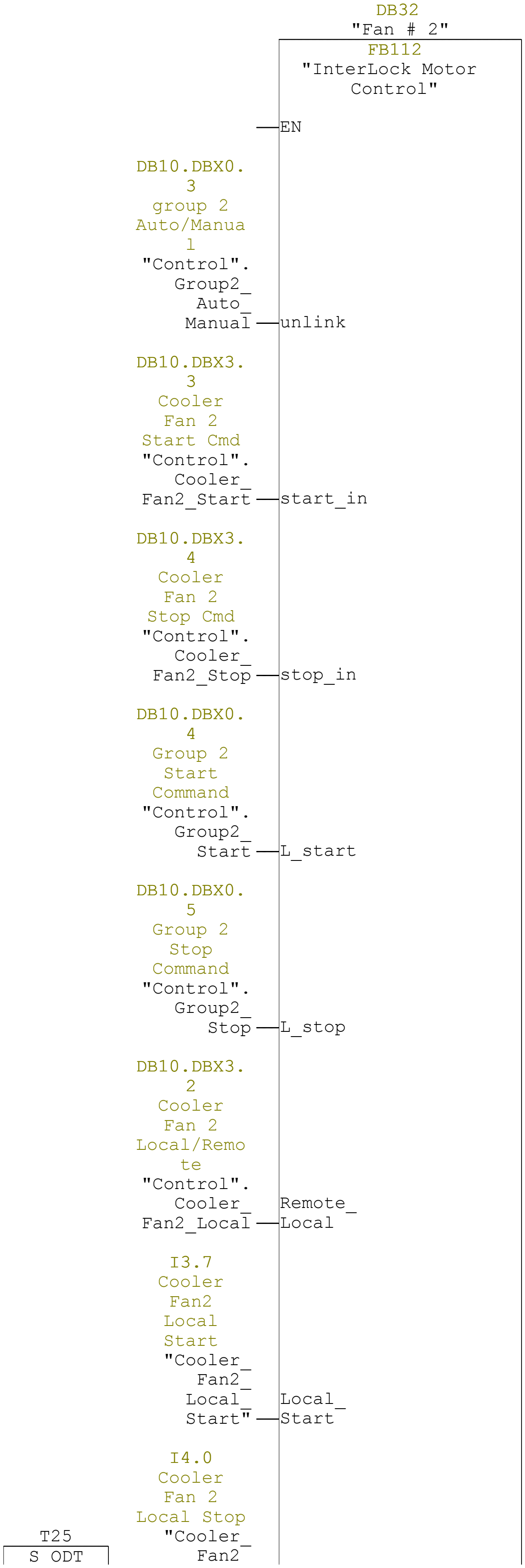
Network: 13



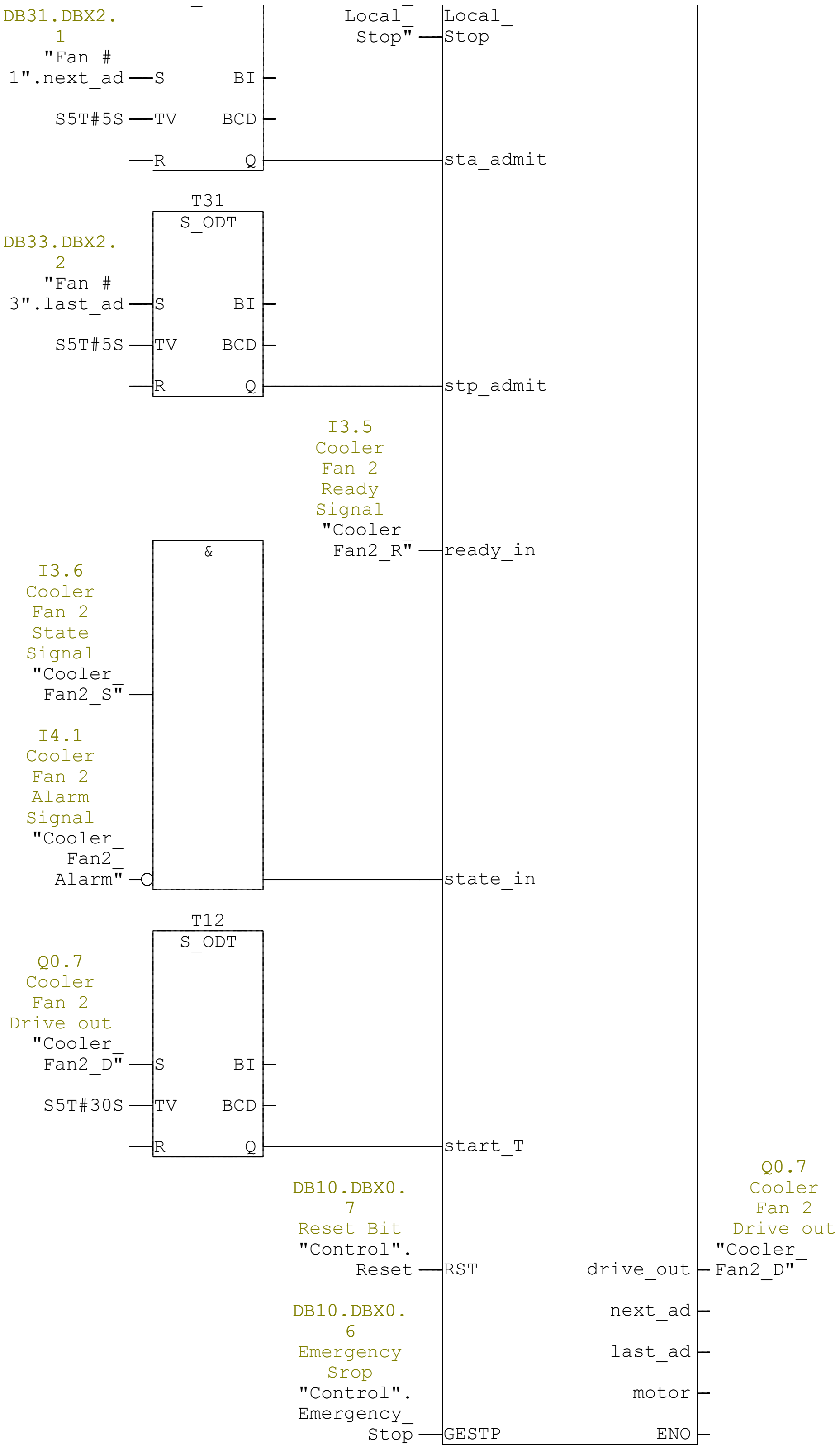
```
                                     DB31
                                   "Fan # 1"
                                     FB112
                              "InterLock Motor
                                   Control"

                           --EN

     DB10.DBX0.
         3
      group 2
    Auto/Manua
         l
    "Control".
       Group2_
         Auto_
       Manual --unlink

     DB10.DBX3.
         0
       Cooler
       Fan 1
     Start Cmd
    "Control".
       Cooler_
    Fan1_Start --start_in

     DB10.DBX3.
         1
       Cooler
       Fan 1
      Stop Cmd
    "Control".
       Cooler_
     Fan1_Stop --stop_in

     DB10.DBX0.
         4
      Group 2
       Start
      Command
    "Control".
       Group2_
         Start --L_start

     DB10.DBX0.
         5
      Group 2
        Stop
      Command
    "Control".
       Group2_
          Stop --L_stop

     DB10.DBX2.
         7
       Cooler
       Fan 1
    Local/Remo
         te
    "Control".
       Cooler_   Remote_
    Fan1_Local --Local

       I3.2
      Cooler
       Fan1
       Local
       Start
     "Cooler_
         Fan1_
        Local_   Local_
        Start" --Start

       I3.3
      Cooler
       Fan 1
          T24  Local Stop
         S_ODT  "Cooler_
DB30.DBX2.         Fan1
```


1
"Crusher
#
3".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
Local_
Stop"
Local_
Stop
sta_admit
T32
S_ODT
DB32.DBX2.
2
"Fan #
2".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I3.0
Cooler
Fan 1
Ready
Signal
"Cooler_
Fan1_R"
ready_in
&
I3.1
Cooler
Fan 1
State
Signal
"Cooler_
Fan1_S"
I3.4
Cooler
Fan 1
Alarm
Signal
"Cooler_
Fan1_
Alarm"
state_in
T11
S_ODT
Q0.6
Cooler
Fan 1
Drive out
"Cooler_
Fan1_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q0.6
Cooler
Fan 1
Drive out
"Cooler_
Fan1_D"
next_ad
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
last_ad
motor
ENO

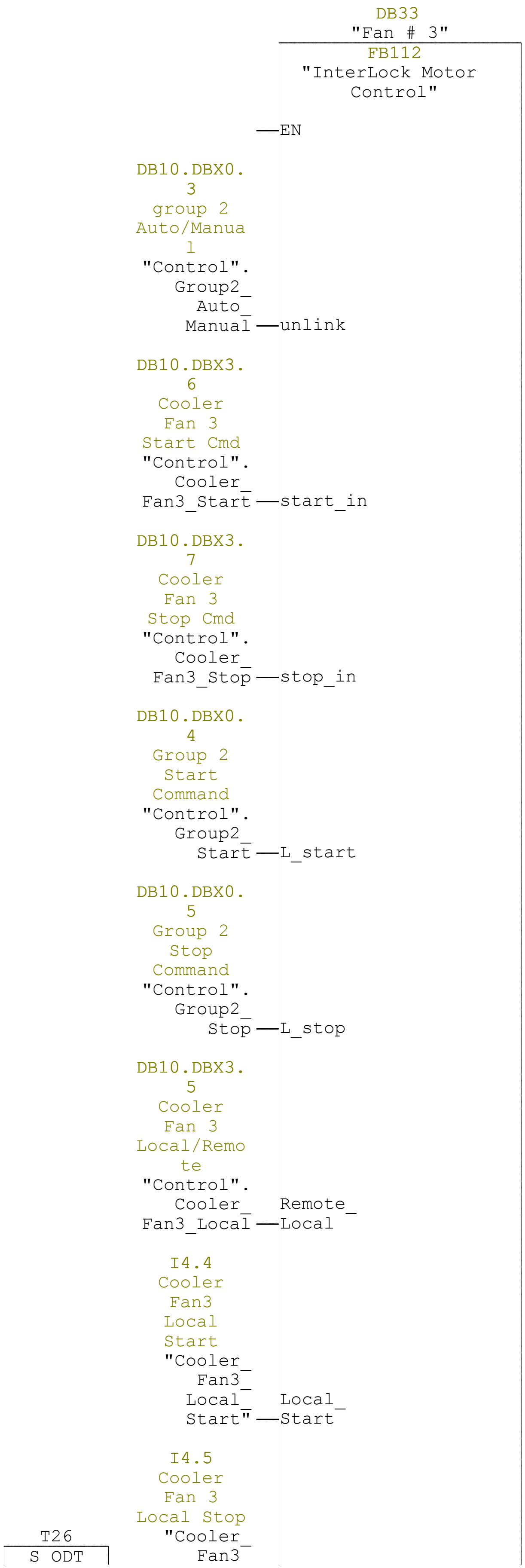
Network: 14

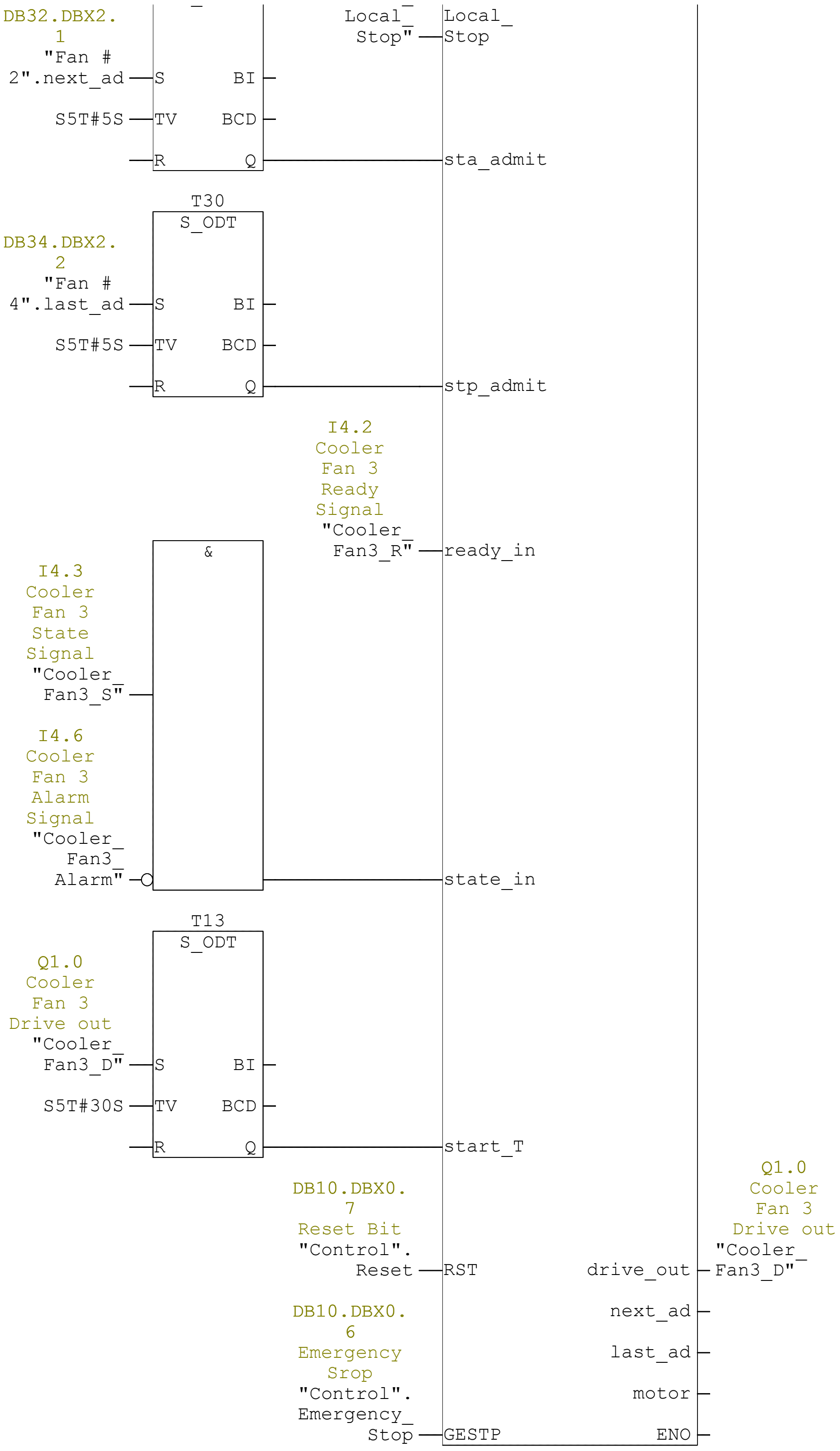


```
                        DB32
                     "Fan # 2"
                       FB112
                 "InterLock Motor
                     Control"

               --EN

DB10.DBX0.
    3
 group 2
Auto/Manua
    l
"Control".
   Group2_
      Auto_
    Manual --unlink

DB10.DBX3.
    3
  Cooler
   Fan 2
 Start Cmd
"Control".
   Cooler_
Fan2_Start --start_in

DB10.DBX3.
    4
  Cooler
   Fan 2
 Stop Cmd
"Control".
   Cooler_
 Fan2_Stop --stop_in

DB10.DBX0.
    4
 Group 2
  Start
 Command
"Control".
   Group2_
     Start --L_start

DB10.DBX0.
    5
 Group 2
   Stop
 Command
"Control".
   Group2_
      Stop --L_stop

DB10.DBX3.
    2
  Cooler
   Fan 2
Local/Remo
    te
"Control".
   Cooler_     Remote_
Fan2_Local --Local

   I3.7
  Cooler
   Fan2
  Local
  Start
  "Cooler_
     Fan2_
    Local_     Local_
    Start" --Start

   I4.0
  Cooler
   Fan 2
Local Stop
  T25          "Cooler_
 S_ODT            Fan2
```


DB31.DBX2.
1
"Fan #
1".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
Local_
Stop"
Local_
Stop
sta_admit
T31
S_ODT
DB33.DBX2.
2
"Fan #
3".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I3.5
Cooler
Fan 2
Ready
Signal
"Cooler_
Fan2_R"
ready_in
&
I3.6
Cooler
Fan 2
State
Signal
"Cooler_
Fan2_S"
I4.1
Cooler
Fan 2
Alarm
Signal
"Cooler_
Fan2_
Alarm"
state_in
T12
S_ODT
Q0.7
Cooler
Fan 2
Drive out
"Cooler_
Fan2_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q0.7
Cooler
Fan 2
Drive out
"Cooler_
Fan2_D"
next_ad
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
last_ad
motor
ENO

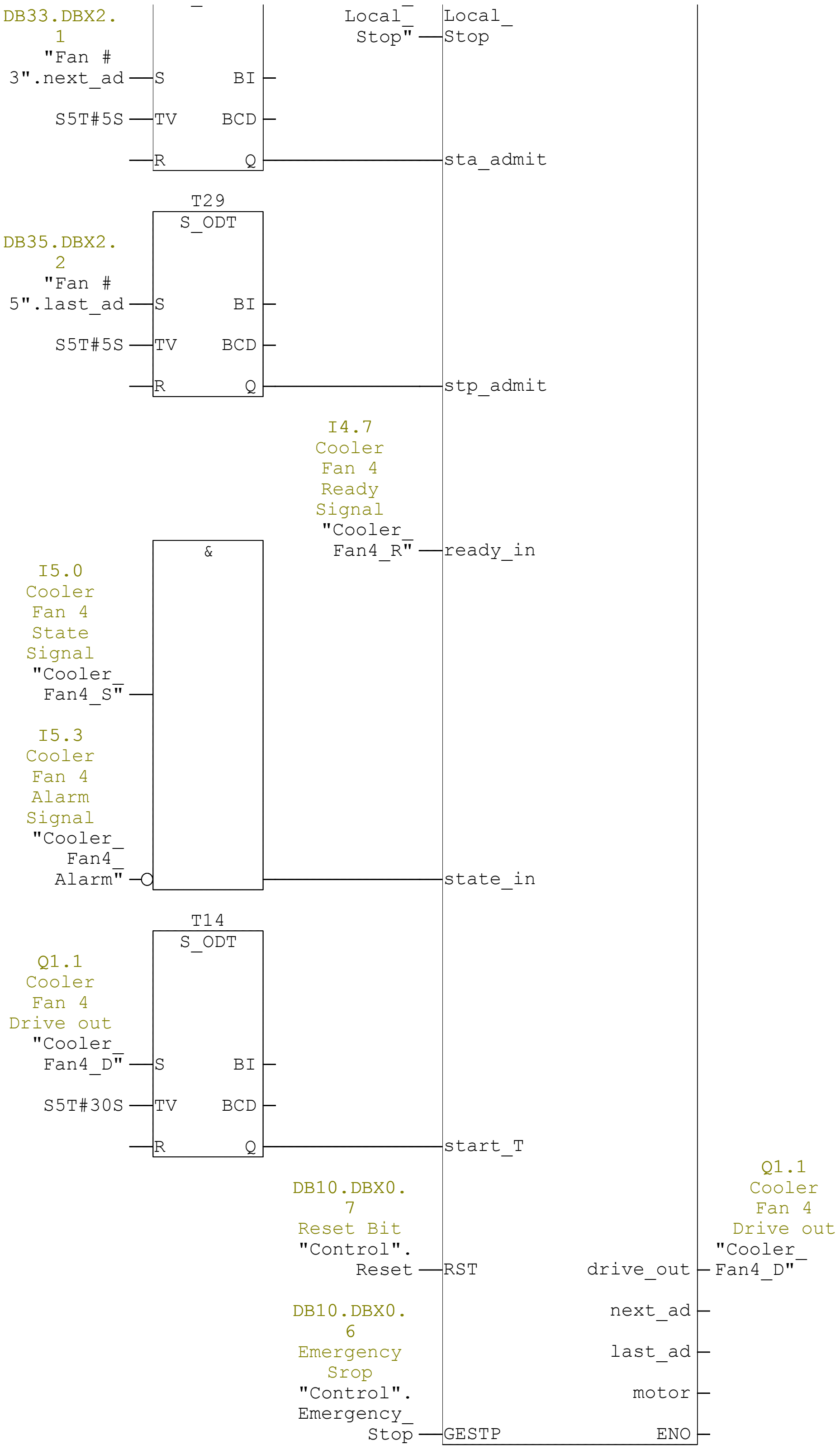
Network: 15


DB33
"Fan # 3"
FB112
"InterLock Motor Control"
EN
DB10.DBX0.3
group 2 Auto/Manual
"Control".Group2_Auto_Manual
unlink
DB10.DBX3.6
Cooler Fan 3 Start Cmd
"Control".Cooler_Fan3_Start
start_in
DB10.DBX3.7
Cooler Fan 3 Stop Cmd
"Control".Cooler_Fan3_Stop
stop_in
DB10.DBX0.4
Group 2 Start Command
"Control".Group2_Start
L_start
DB10.DBX0.5
Group 2 Stop Command
"Control".Group2_Stop
L_stop
DB10.DBX3.5
Cooler Fan 3 Local/Remote
"Control".Cooler_Fan3_Local
Remote_Local
I4.4
Cooler Fan3 Local Start
"Cooler_Fan3_Local_Start"
Local_Start
I4.5
Cooler Fan 3 Local Stop
"Cooler_Fan3
T26
S ODT

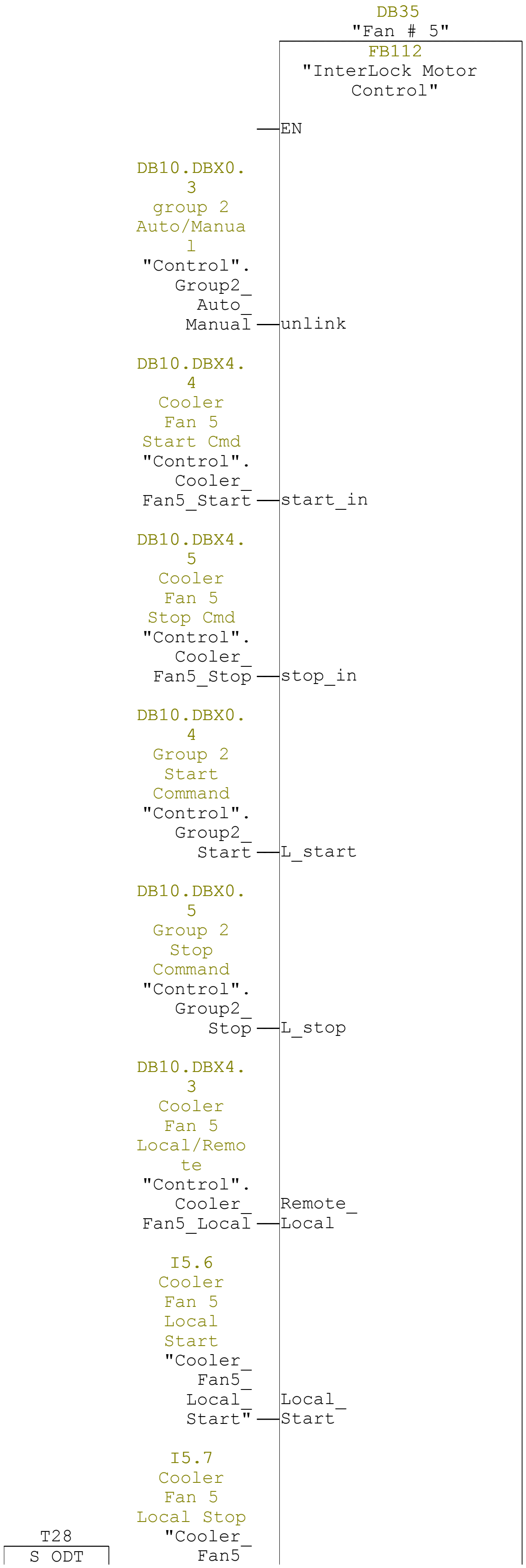

DB32.DBX2.1
"Fan # 2".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
Local_Stop"
Local_Stop
sta_admit
T30
S_ODT
DB34.DBX2.2
"Fan # 4".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I4.2
Cooler Fan 3 Ready Signal
"Cooler_Fan3_R"
ready_in
&
I4.3
Cooler Fan 3 State Signal
"Cooler_Fan3_S"
I4.6
Cooler Fan 3 Alarm Signal
"Cooler_Fan3_Alarm"
state_in
T13
S_ODT
Q1.0
Cooler Fan 3 Drive out
"Cooler_Fan3_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.7
Reset Bit
"Control".Reset
RST
drive_out
Q1.0
Cooler Fan 3 Drive out
"Cooler_Fan3_D"
DB10.DBX0.6
Emergency Srop
"Control".Emergency_Stop
GESTP
next_ad
last_ad
motor
ENO

Network: 16

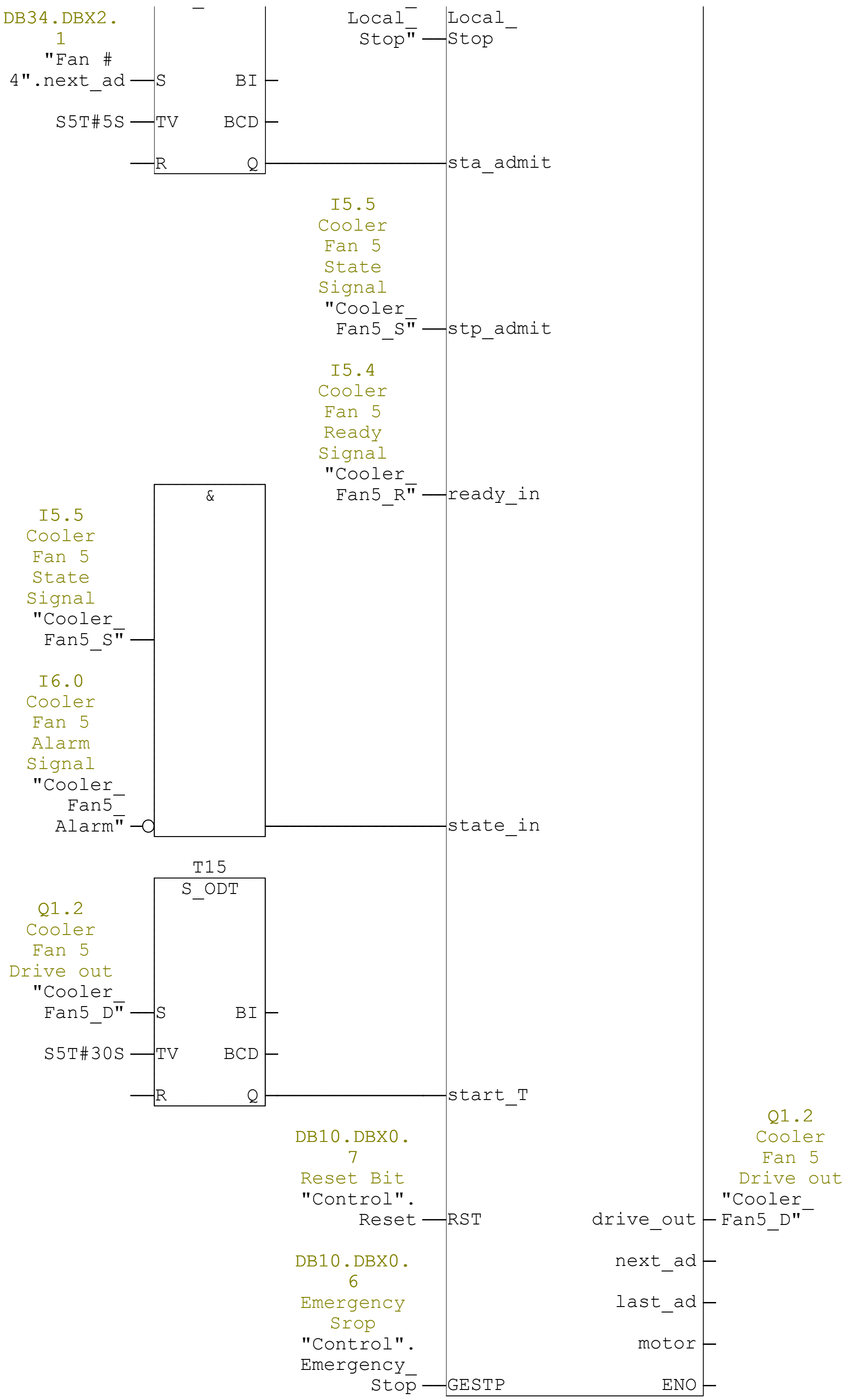
```
                                        DB34
                                     "Fan # 4"
                                ┌──────────────────────┐
                                │       FB112          │
                                │  "InterLock Motor    │
                                │      Control"        │
                                │                      │
                              ──┤EN                    │
                                │                      │
         DB10.DBX0.             │                      │
             3                  │                      │
          group 2               │                      │
         Auto/Manua             │                      │
             l                  │                      │
         "Control".             │                      │
             Group2_            │                      │
               Auto_            │                      │
              Manual──┤unlink                │
                                │                      │
         DB10.DBX4.             │                      │
             1                  │                      │
           Cooler               │                      │
            Fan 4               │                      │
          Start Cmd             │                      │
         "Control".             │                      │
             Cooler_            │                      │
         Fan4_Start ──┤start_in              │
                                │                      │
         DB10.DBX4.             │                      │
             2                  │                      │
           Cooler               │                      │
            Fan 4               │                      │
          Stop Cmd              │                      │
         "Control".             │                      │
             Cooler_            │                      │
          Fan4_Stop ──┤stop_in               │
                                │                      │
         DB10.DBX0.             │                      │
             4                  │                      │
          Group 2               │                      │
           Start                │                      │
          Command               │                      │
         "Control".             │                      │
             Group2_            │                      │
               Start ──┤L_start               │
                                │                      │
         DB10.DBX0.             │                      │
             5                  │                      │
          Group 2               │                      │
           Stop                 │                      │
          Command               │                      │
         "Control".             │                      │
             Group2_            │                      │
                Stop ──┤L_stop                │
                                │                      │
         DB10.DBX4.             │                      │
             0                  │                      │
           Cooler               │                      │
            Fan 4               │                      │
         Local/Remo             │                      │
             te                 │                      │
         "Control".             │                      │
             Cooler_            │Remote_               │
         Fan4_Local ──┤Local                 │
                                │                      │
            I5.1                │                      │
           Cooler               │                      │
            Fan4                │                      │
            Local               │                      │
            Start               │                      │
            "Cooler_            │                      │
               Fan4_            │                      │
              Local_            │Local_                │
              Start" ──┤Start                 │
                                │                      │
            I5.2                │                      │
           Cooler               │                      │
            Fan 4               │                      │
         Local Stop             │                      │
    T27     "Cooler_            │                      │
 ┌ S ODT ┐     Fan4             │                      │
```


DB33.DBX2.
1
"Fan #
3".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
Local_
Stop"
Local_
Stop
sta_admit
T29
S_ODT
DB35.DBX2.
2
"Fan #
5".last_ad
S
BI
S5T#5S
TV
BCD
R
Q
stp_admit
I4.7
Cooler
Fan 4
Ready
Signal
"Cooler_
Fan4_R"
ready_in
&
I5.0
Cooler
Fan 4
State
Signal
"Cooler_
Fan4_S"
I5.3
Cooler
Fan 4
Alarm
Signal
"Cooler_
Fan4_
Alarm"
state_in
T14
S_ODT
Q1.1
Cooler
Fan 4
Drive out
"Cooler_
Fan4_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q1.1
Cooler
Fan 4
Drive out
"Cooler_
Fan4_D"
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
next_ad
last_ad
motor
ENO

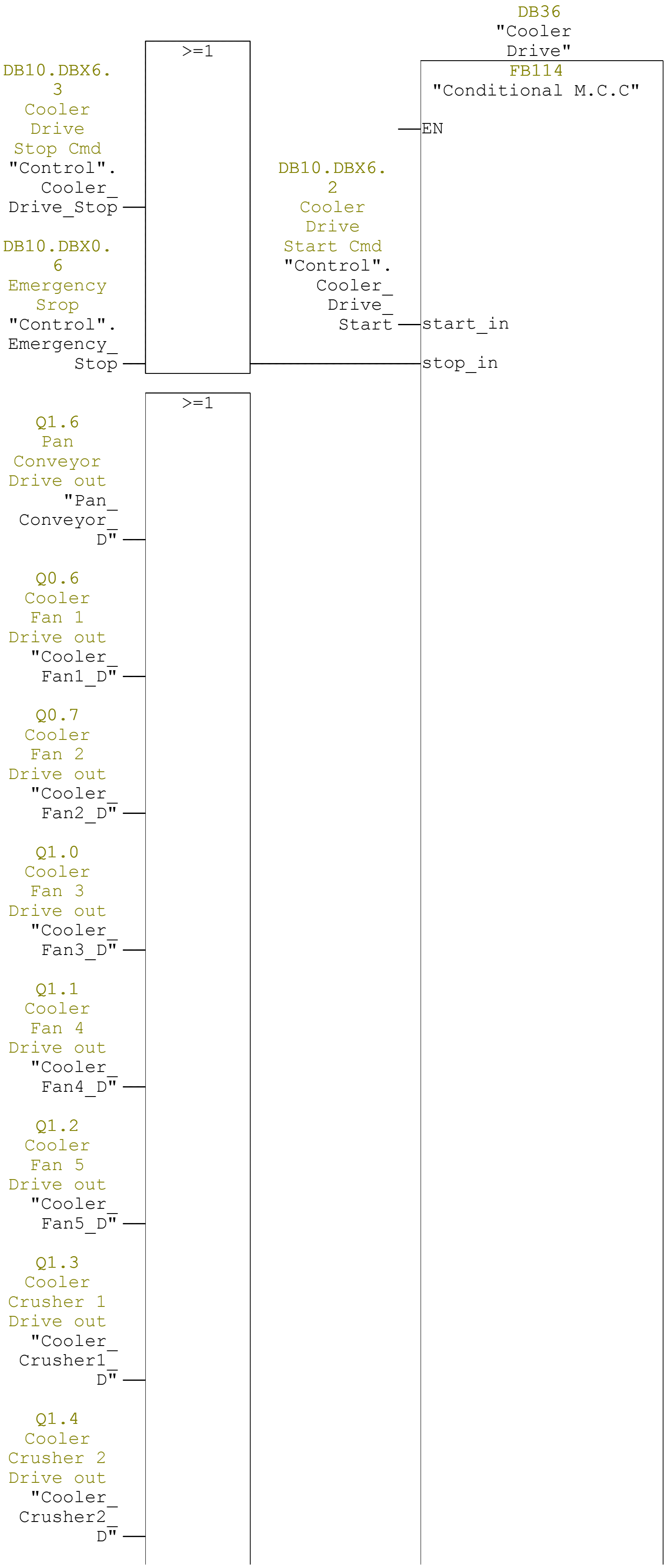
Network: 17

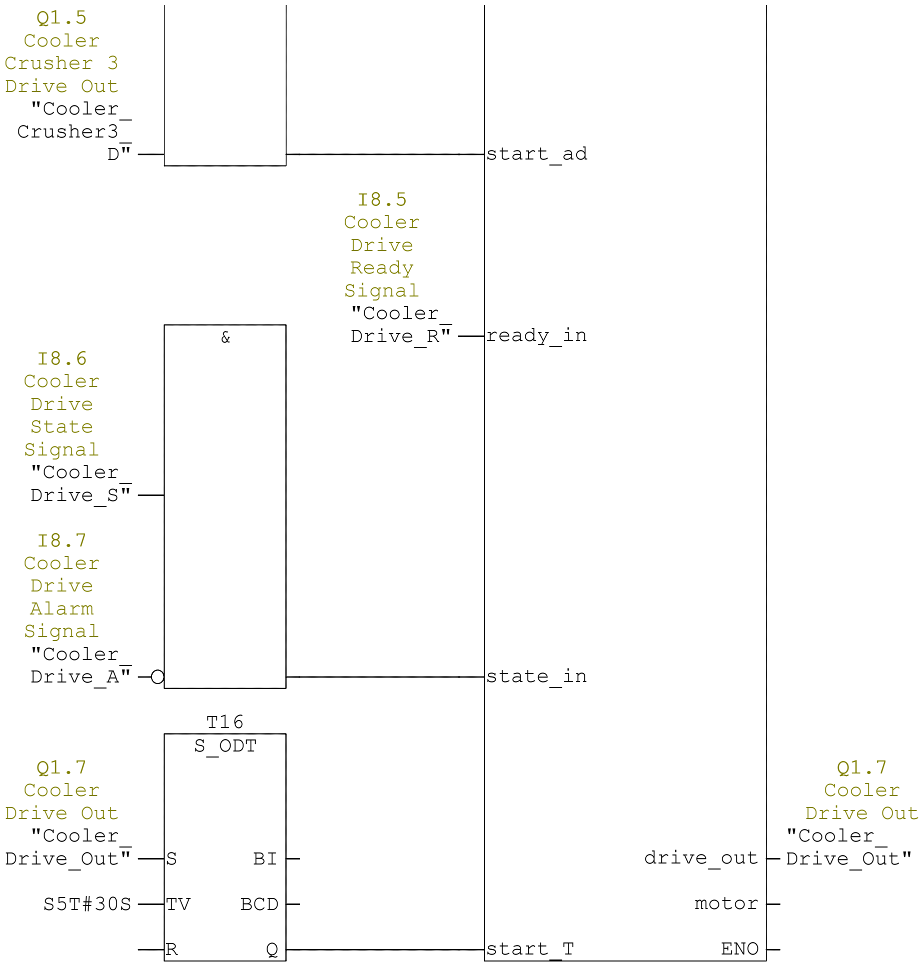


```
                                    DB35
                                 "Fan # 5"
                              FB112
                           "InterLock Motor
                               Control"

                        ──EN

      DB10.DBX0.
          3
       group 2
      Auto/Manua
          l
      "Control".
         Group2_
           Auto_
          Manual──unlink

      DB10.DBX4.
          4
        Cooler
        Fan 5
      Start Cmd
      "Control".
         Cooler_
      Fan5_Start──start_in

      DB10.DBX4.
          5
        Cooler
        Fan 5
       Stop Cmd
      "Control".
         Cooler_
       Fan5_Stop──stop_in

      DB10.DBX0.
          4
       Group 2
        Start
       Command
      "Control".
         Group2_
           Start──L_start

      DB10.DBX0.
          5
       Group 2
         Stop
       Command
      "Control".
         Group2_
            Stop──L_stop

      DB10.DBX4.
          3
        Cooler
        Fan 5
      Local/Remo
          te
      "Control".
         Cooler_   Remote_
      Fan5_Local──Local

         I5.6
        Cooler
        Fan 5
        Local
        Start
       "Cooler_
           Fan5_
          Local_   Local_
          Start"──Start

         I5.7
        Cooler
        Fan 5
      Local Stop
   T28 "Cooler_
 S ODT     Fan5
```

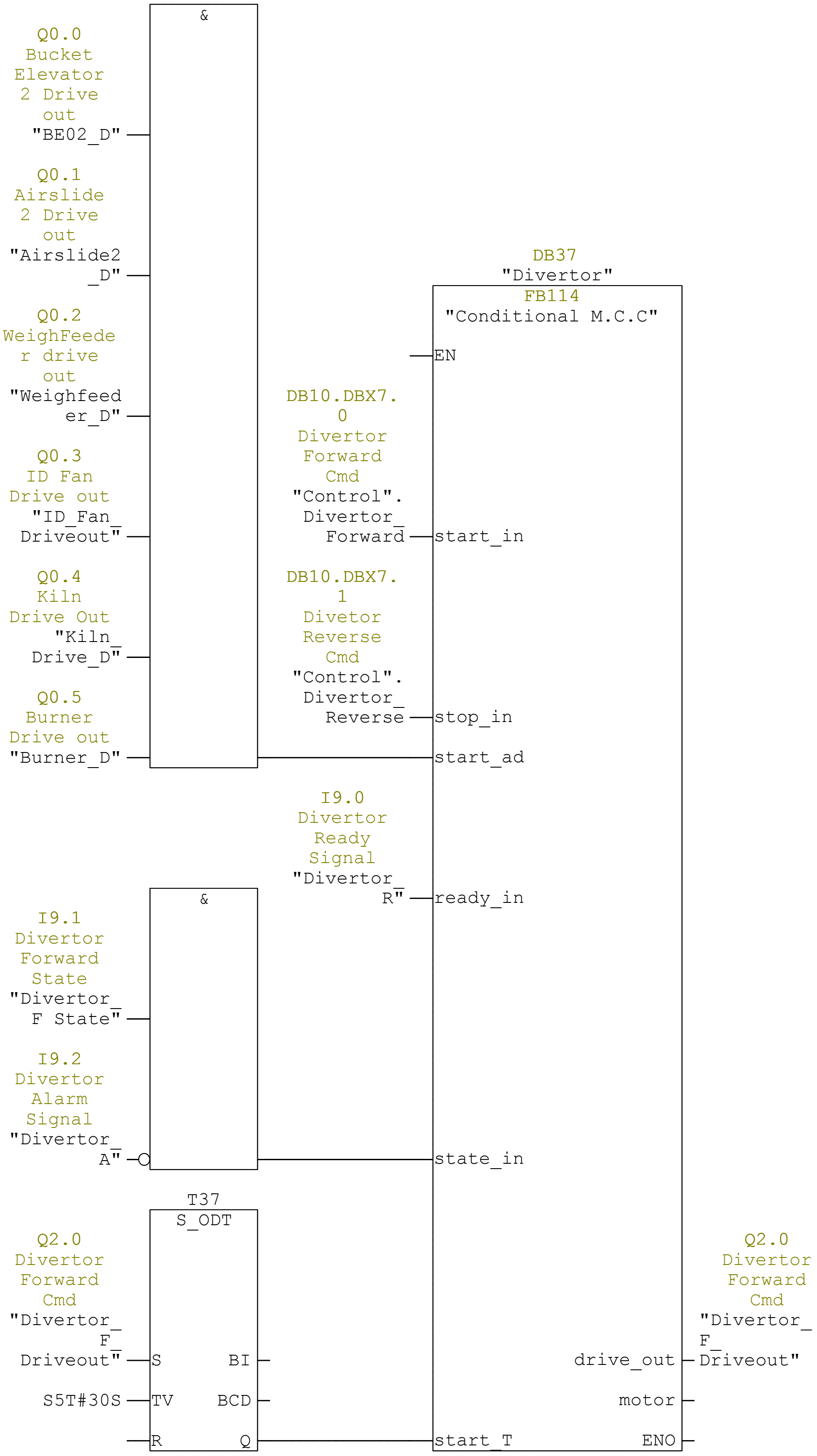

DB34.DBX2.
1
"Fan #
4".next_ad
S
BI
S5T#5S
TV
BCD
R
Q
Local_
Stop"
Local_
Stop
sta_admit
I5.5
Cooler
Fan 5
State
Signal
"Cooler_
Fan5_S"
stp_admit
I5.4
Cooler
Fan 5
Ready
Signal
"Cooler_
Fan5_R"
ready_in
&
I5.5
Cooler
Fan 5
State
Signal
"Cooler_
Fan5_S"
I6.0
Cooler
Fan 5
Alarm
Signal
"Cooler_
Fan5_
Alarm"
state_in
T15
S_ODT
Q1.2
Cooler
Fan 5
Drive out
"Cooler_
Fan5_D"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
DB10.DBX0.
7
Reset Bit
"Control".
Reset
RST
drive_out
Q1.2
Cooler
Fan 5
Drive out
"Cooler_
Fan5_D"
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
GESTP
next_ad
last_ad
motor
ENO

Network: 18

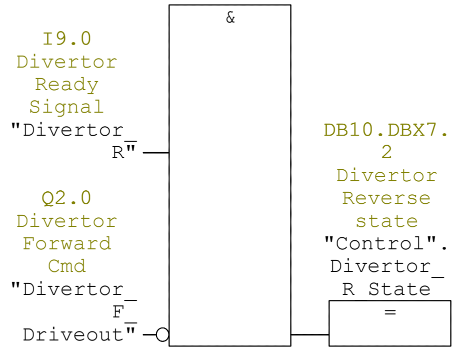

DB36
"Cooler
Drive"
FB114
"Conditional M.C.C"
EN
>=1
DB10.DBX6.
3
Cooler
Drive
Stop Cmd
"Control".
Cooler_
Drive_Stop
DB10.DBX0.
6
Emergency
Srop
"Control".
Emergency_
Stop
DB10.DBX6.
2
Cooler
Drive
Start Cmd
"Control".
Cooler_
Drive_
Start
start_in
stop_in
>=1
Q1.6
Pan
Conveyor
Drive out
"Pan_
Conveyor_
D"
Q0.6
Cooler
Fan 1
Drive out
"Cooler_
Fan1_D"
Q0.7
Cooler
Fan 2
Drive out
"Cooler_
Fan2_D"
Q1.0
Cooler
Fan 3
Drive out
"Cooler_
Fan3_D"
Q1.1
Cooler
Fan 4
Drive out
"Cooler_
Fan4_D"
Q1.2
Cooler
Fan 5
Drive out
"Cooler_
Fan5_D"
Q1.3
Cooler
Crusher 1
Drive out
"Cooler_
Crusher1_
D"
Q1.4
Cooler
Crusher 2
Drive out
"Cooler_
Crusher2_
D"


Q1.5
Cooler
Crusher 3
Drive Out
"Cooler_
Crusher3_
D"
start_ad
I8.5
Cooler
Drive
Ready
Signal
"Cooler_
Drive_R"
ready_in
&
I8.6
Cooler
Drive
State
Signal
"Cooler_
Drive_S"
I8.7
Cooler
Drive
Alarm
Signal
"Cooler_
Drive_A"
state_in
T16
S_ODT
Q1.7
Cooler
Drive Out
"Cooler_
Drive_Out"
S
BI
TV
BCD
R
Q
S5T#30S
start_T
drive_out
motor
ENO
Q1.7
Cooler
Drive Out
"Cooler_
Drive_Out"

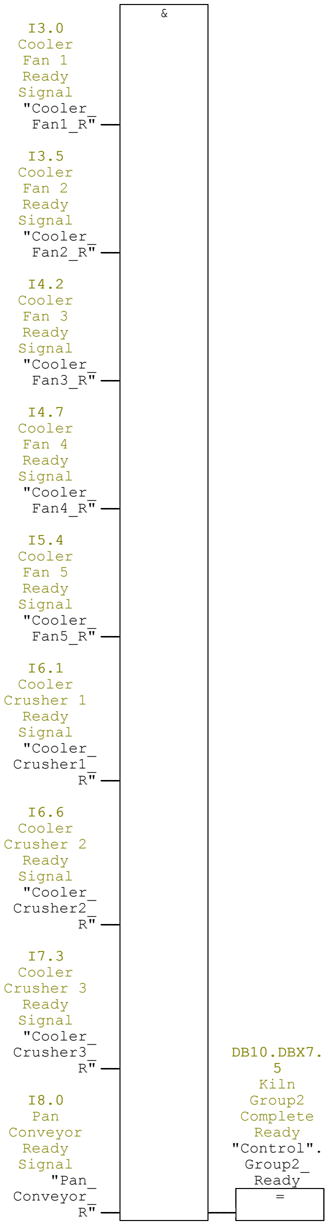
Network: 19


&
Q0.0
Bucket
Elevator
2 Drive
out
"BE02_D"
Q0.1
Airslide
2 Drive
out
"Airslide2
_D"
Q0.2
WeighFeede
r drive
out
"Weighfeed
er_D"
Q0.3
ID Fan
Drive out
"ID_Fan_
Driveout"
Q0.4
Kiln
Drive Out
"Kiln_
Drive_D"
Q0.5
Burner
Drive out
"Burner_D"
DB37
"Divertor"
FB114
"Conditional M.C.C"
EN
DB10.DBX7.
0
Divertor
Forward
Cmd
"Control".
Divertor_
Forward
start_in
DB10.DBX7.
1
Divetor
Reverse
Cmd
"Control".
Divertor_
Reverse
stop_in
start_ad
I9.0
Divertor
Ready
Signal
"Divertor_
R"
ready_in
&
I9.1
Divertor
Forward
State
"Divertor_
F State"
I9.2
Divertor
Alarm
Signal
"Divertor_
A"
state_in
T37
S_ODT
Q2.0
Divertor
Forward
Cmd
"Divertor_
F_
Driveout"
S
BI
S5T#30S
TV
BCD
R
Q
start_T
drive_out
motor
ENO
Q2.0
Divertor
Forward
Cmd
"Divertor_
F_
Driveout"

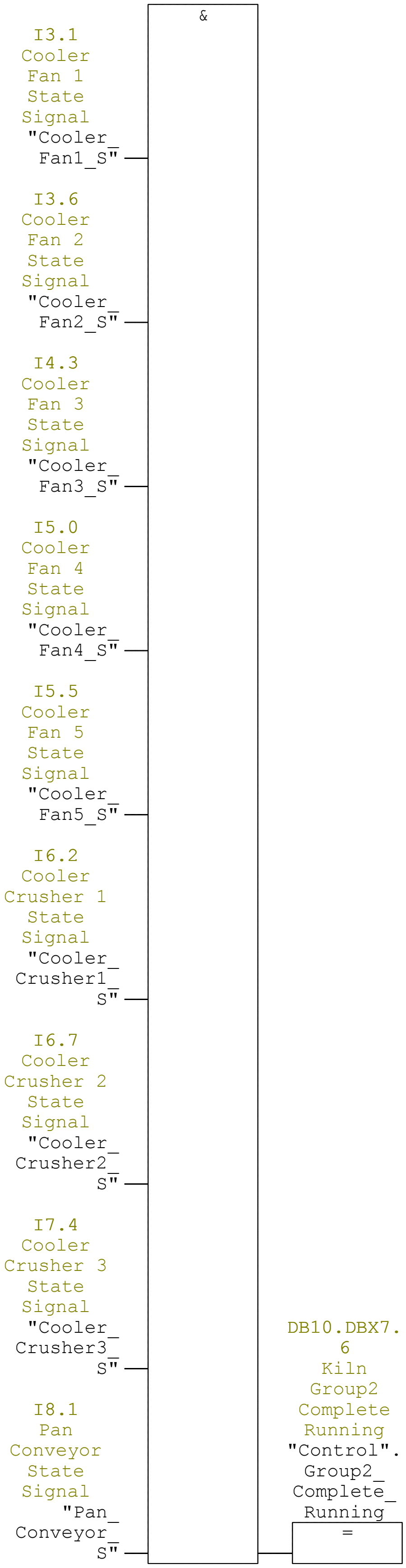
Network: 20 Divertor Forward Reverse state



Network: 21



Network: 22 Kiln Group2 Complete Running


I3.1 Cooler Fan 1 State Signal "Cooler_Fan1_S"
I3.6 Cooler Fan 2 State Signal "Cooler_Fan2_S"
I4.3 Cooler Fan 3 State Signal "Cooler_Fan3_S"
I5.0 Cooler Fan 4 State Signal "Cooler_Fan4_S"
I5.5 Cooler Fan 5 State Signal "Cooler_Fan5_S"
I6.2 Cooler Crusher 1 State Signal "Cooler_Crusher1_S"
I6.7 Cooler Crusher 2 State Signal "Cooler_Crusher2_S"
I7.4 Cooler Crusher 3 State Signal "Cooler_Crusher3_S"
I8.1 Pan Conveyor State Signal "Pan_Conveyor_S"
&
DB10.DBX7.6 Kiln Group2 Complete Running "Control".Group2_Complete_Running
=

FC3 - <offline>

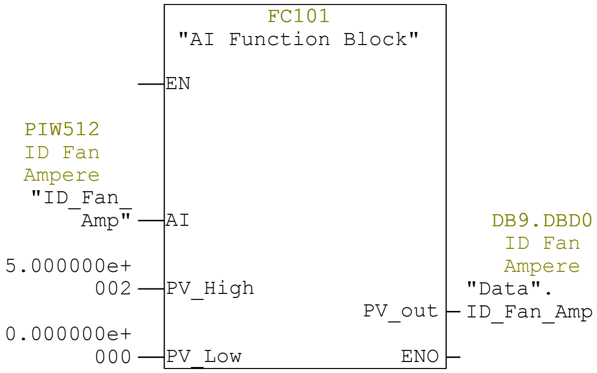
"Analog Parameters"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 04/24/2022 01:30:43 PM
Interface: 04/11/2022 03:12:30 AM
Lengths (block/logic/data): 02700 02532 00018

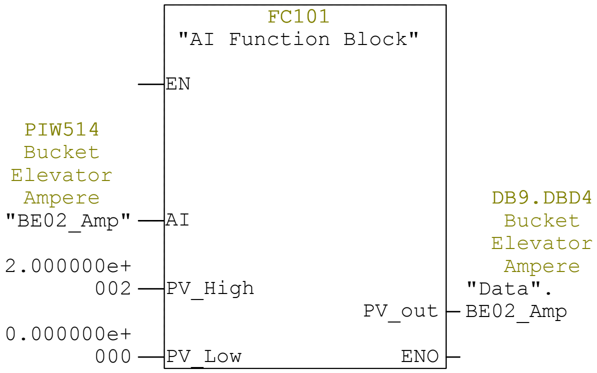
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC3

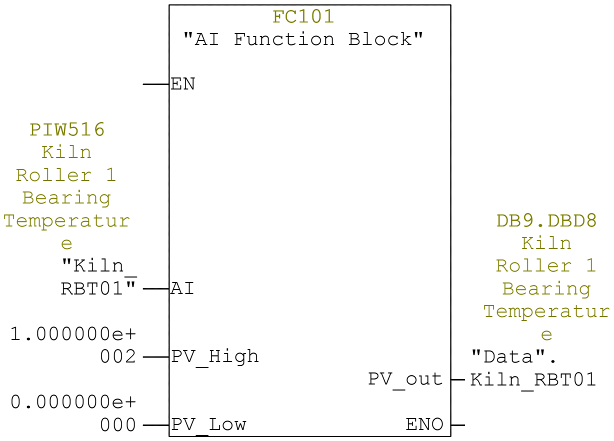
Network: 1



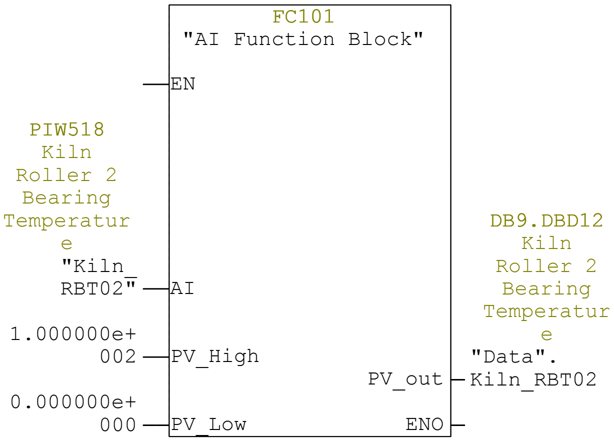
Network: 2



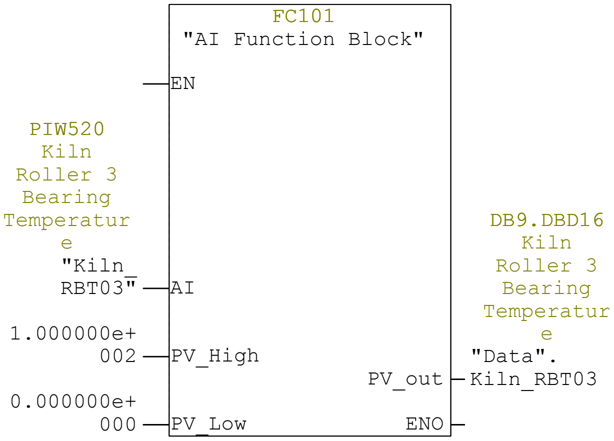
Network: 3



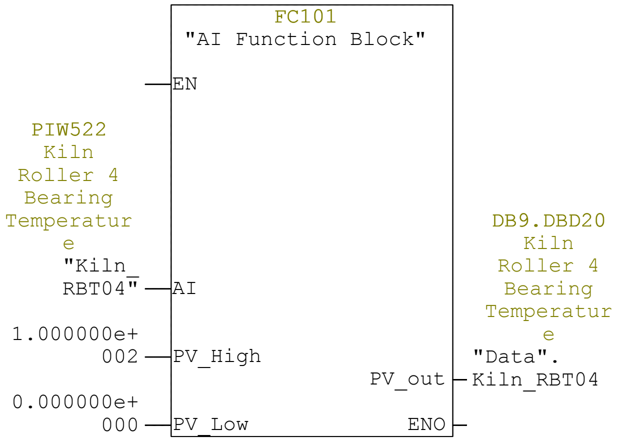
Network: 4



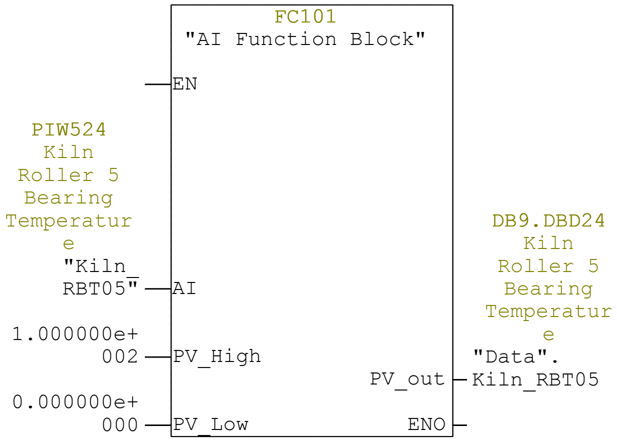
Network: 5



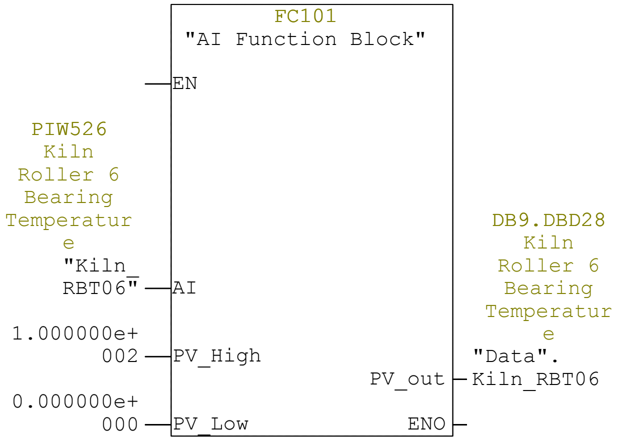
Network: 6



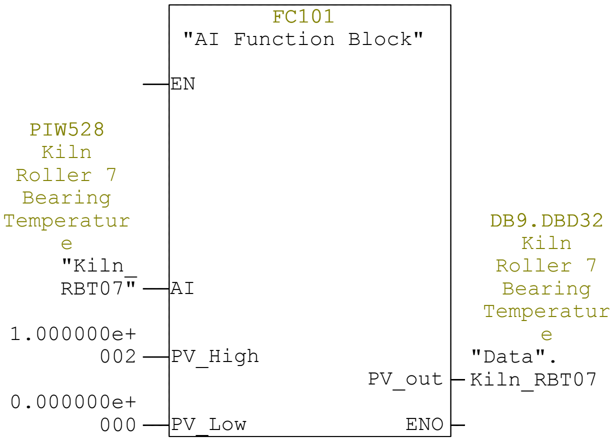
Network: 7



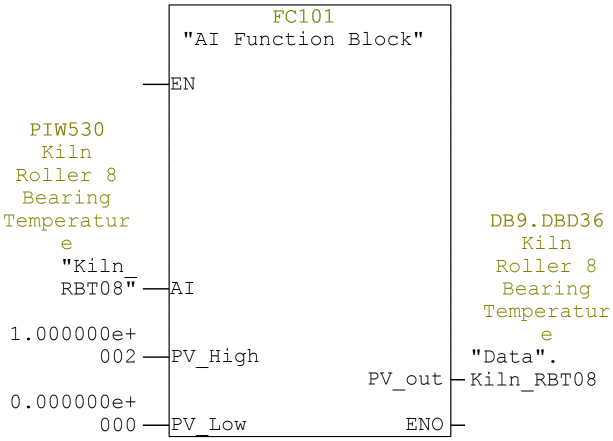
Network: 8



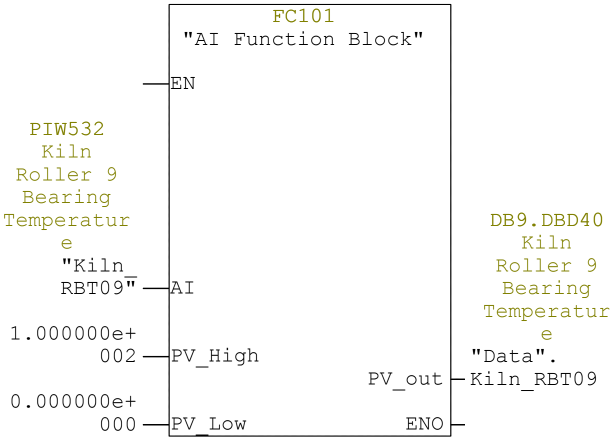
Network: 9



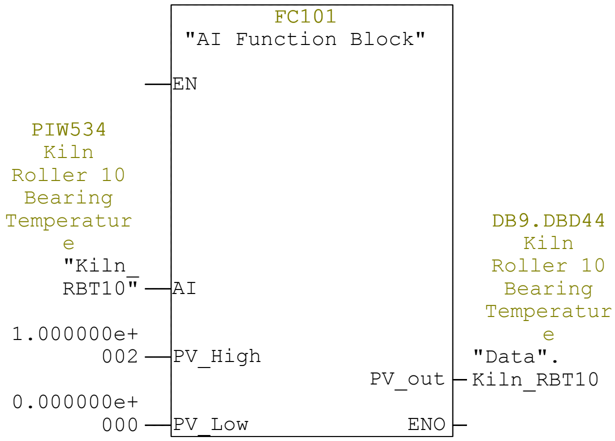
Network: 10



Network: 11

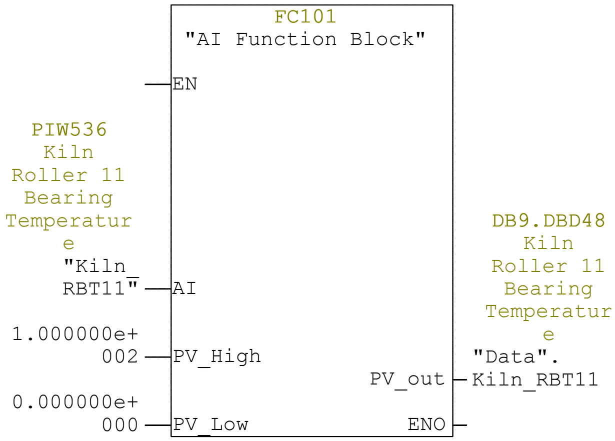


Network: 12



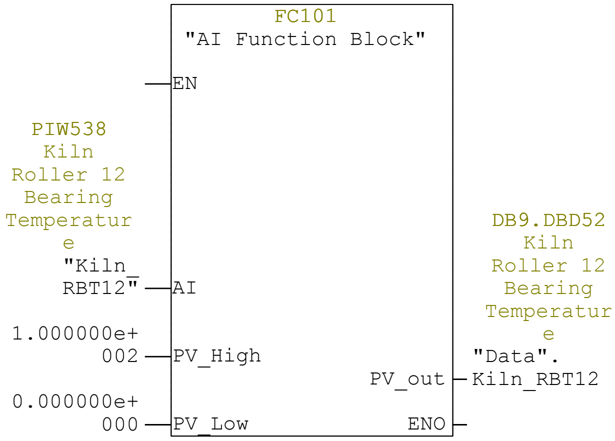
FC101
"AI Function Block"
EN
PIW534
Kiln Roller 10 Bearing Temperature
"Kiln_RBT10" — AI
1.000000e+002 — PV_High
0.000000e+000 — PV_Low
PV_out — DB9.DBD44
Kiln Roller 10 Bearing Temperature
"Data".Kiln_RBT10
ENO

Network: 13



FC101
"AI Function Block"
EN
PIW536
Kiln Roller 11 Bearing Temperature
"Kiln_RBT11" — AI
1.000000e+002 — PV_High
0.000000e+000 — PV_Low
PV_out — DB9.DBD48
Kiln Roller 11 Bearing Temperature
"Data".Kiln_RBT11
ENO

Network: 14



FC101
"AI Function Block"
EN
PIW538
Kiln Roller 12 Bearing Temperature
"Kiln_RBT12" — AI
1.000000e+002 — PV_High
0.000000e+000 — PV_Low
PV_out — DB9.DBD52
Kiln Roller 12 Bearing Temperature
"Data".Kiln_RBT12
ENO

Network: 15

```
                     FC101
               "AI Function Block"
             --|EN
  PIW540       |
  Cooler       |
   Fan 1       |
  Ampere       |                         DB9.DBD56
   "Cooler_    |                           Cooler
 Fan1_Amp" --|AI                            Fan 1
               |                           Ampere
1.500000e+     |                        "Data".
       002 --|PV_High                   Cooler_
               |              PV_out |-- Fan1_Amp
0.000000e+     |
       000 --|PV_Low             ENO |--
```

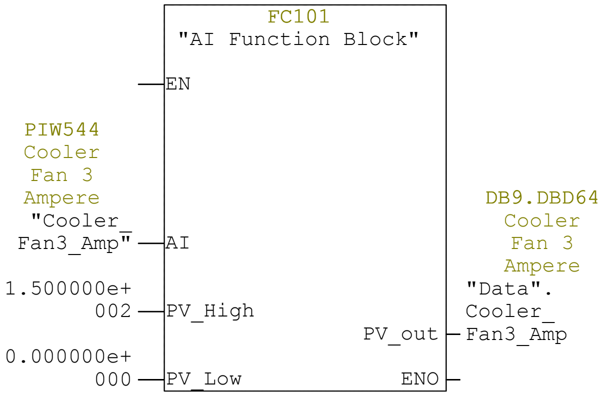
Network: 16

```
                     FC101
               "AI Function Block"
             --|EN
  PIW542       |
  Cooler       |
   Fan 2       |
  Ampere       |                         DB9.DBD60
   "Cooler_    |                           Cooler
 Fan2_Amp" --|AI                            Fan 2
               |                           Ampere
1.500000e+     |                        "Data".
       002 --|PV_High                   Cooler_
               |              PV_out |-- Fan2_Amp
0.000000e+     |
       000 --|PV_Low             ENO |--
```

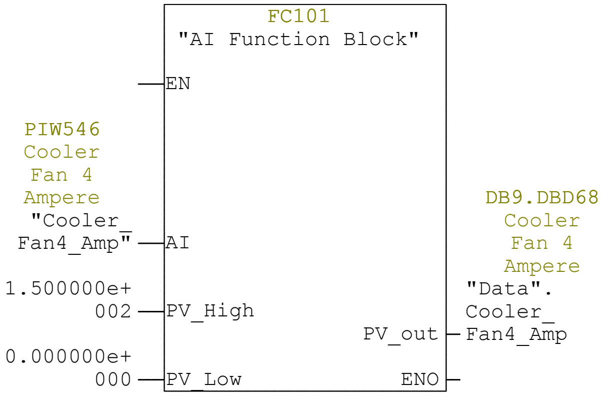
Network: 17

```
                     FC101
               "AI Function Block"
             --|EN
  PIW544       |
  Cooler       |
   Fan 3       |
  Ampere       |                         DB9.DBD64
   "Cooler_    |                           Cooler
 Fan3_Amp" --|AI                            Fan 3
               |                           Ampere
1.500000e+     |                        "Data".
       002 --|PV_High                   Cooler_
               |              PV_out |-- Fan3_Amp
0.000000e+     |
       000 --|PV_Low             ENO |--
```

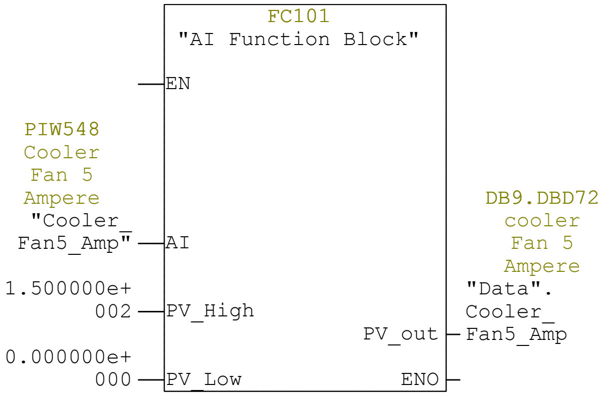
Network: 18



Network: 19



Network: 20



Network: 21

```
                    FC101
               "AI Function Block"
              +----------------------+
          ----|EN                    |
  PIW550      |                      |
  Cooler      |                      |
 Crusher 1    |                      |
  Ampere      |                      |   DB9.DBD76
  "Cooler_    |                      |     Cooler
 Crusher1_    |                      |   Crusher 1
      Amp" ---|AI                    |     Ampere
              |                      |  "Data".
3.000000e+    |                      |  Cooler_
       001 ---|PV_High               |  Crusher1_
              |                PV_out|--Amp
0.000000e+    |                      |
       000 ---|PV_Low             ENO|--
              +----------------------+
```

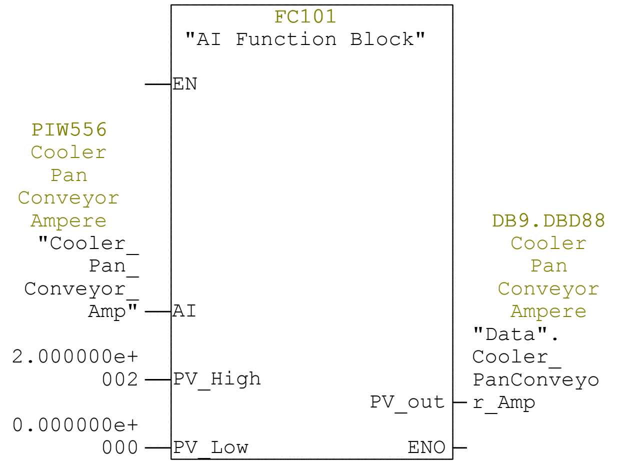
Network: 22

```
                    FC101
               "AI Function Block"
              +----------------------+
          ----|EN                    |
  PIW552      |                      |
  Cooler      |                      |
 Crusher 2    |                      |
  Ampere      |                      |   DB9.DBD80
  "Cooler_    |                      |     Cooler
 Crusher2_    |                      |   Crusher 2
      Amp" ---|AI                    |     Ampere
              |                      |  "Data".
3.000000e+    |                      |  Cooler_
       001 ---|PV_High               |  Crusher2_
              |                PV_out|--Amp
0.000000e+    |                      |
       000 ---|PV_Low             ENO|--
              +----------------------+
```

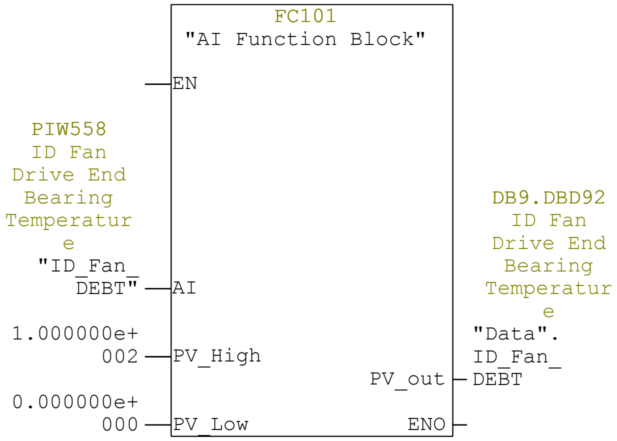
Network: 23

```
                    FC101
               "AI Function Block"
              +----------------------+
          ----|EN                    |
  PIW554      |                      |
  Cooler      |                      |
 Crusher 3    |                      |
  Ampere      |                      |   DB9.DBD84
  "Cooler_    |                      |     Cooler
 Crusher3_    |                      |   Crusher 3
      Amp" ---|AI                    |     Ampere
              |                      |  "Data".
3.000000e+    |                      |  Cooler_
       001 ---|PV_High               |  Crusher3_
              |                PV_out|--Amp
0.000000e+    |                      |
       000 ---|PV_Low             ENO|--
              +----------------------+
```

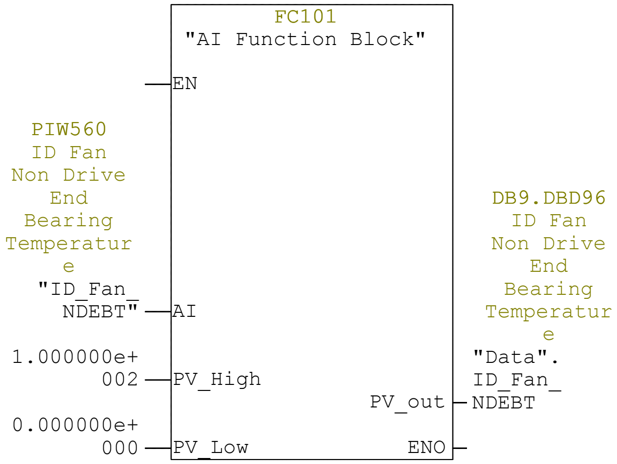
Network: 24



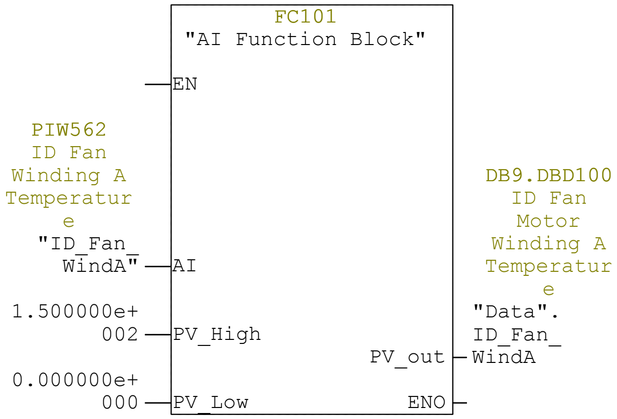
Network: 25



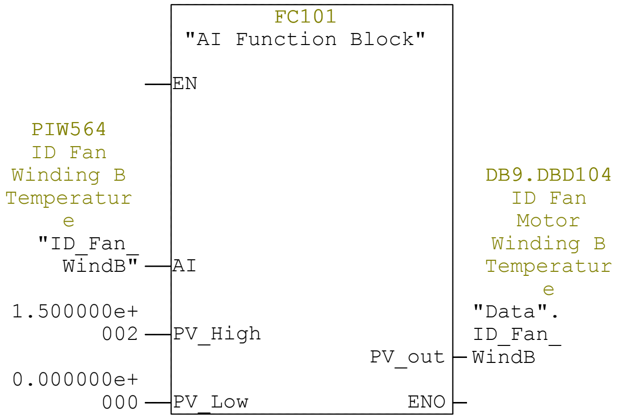
Network: 26



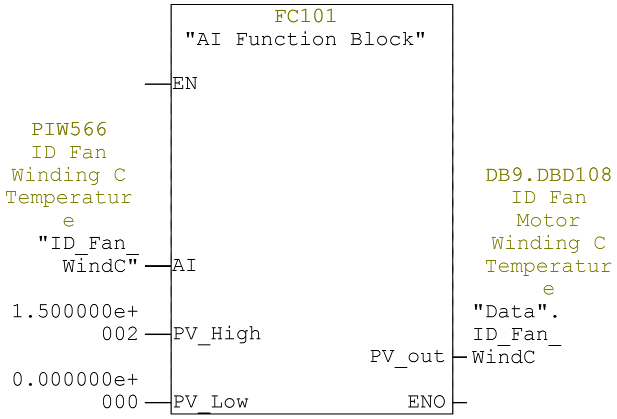
Network: 27



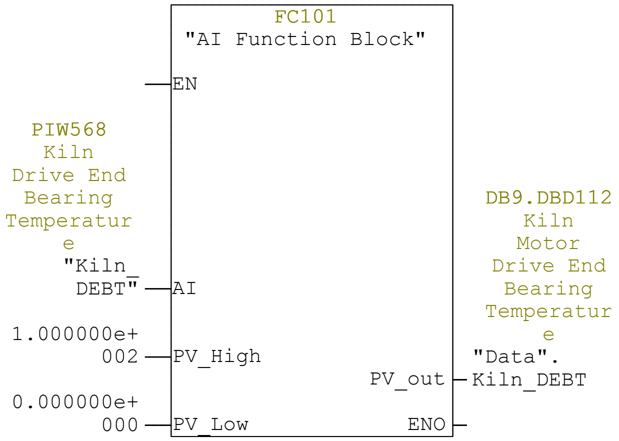
Network: 28



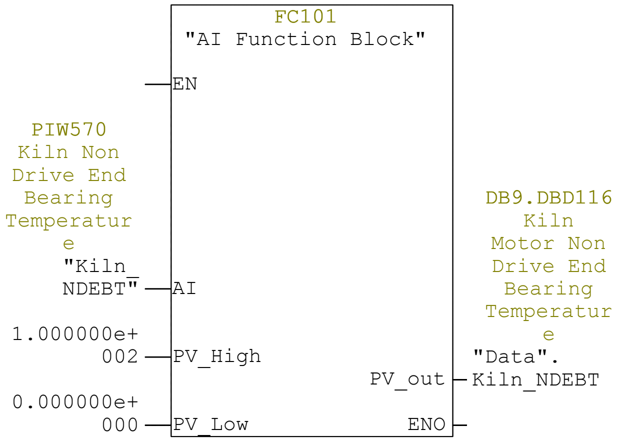
Network: 29



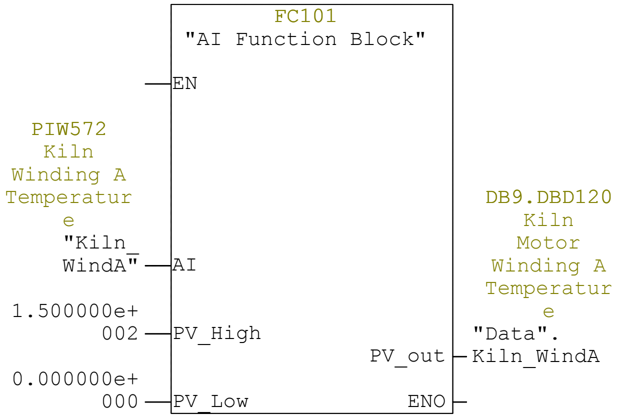
Network: 30



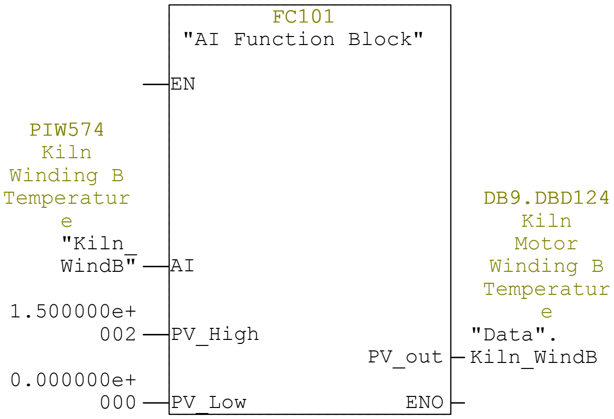
Network: 31



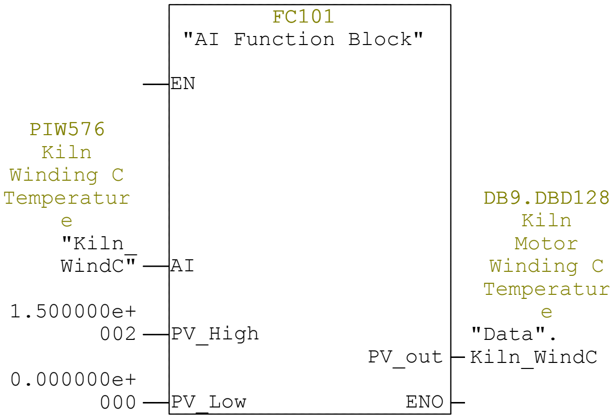
Network: 32



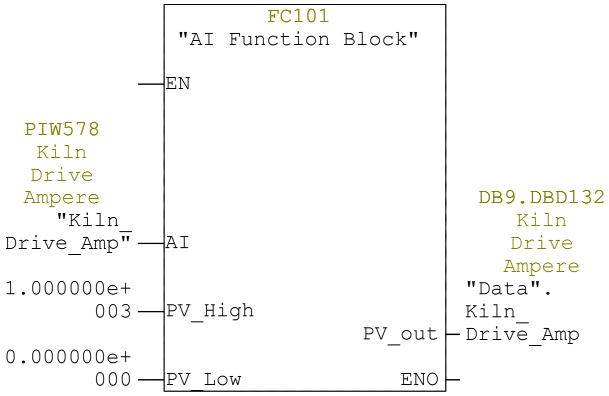
Network: 33



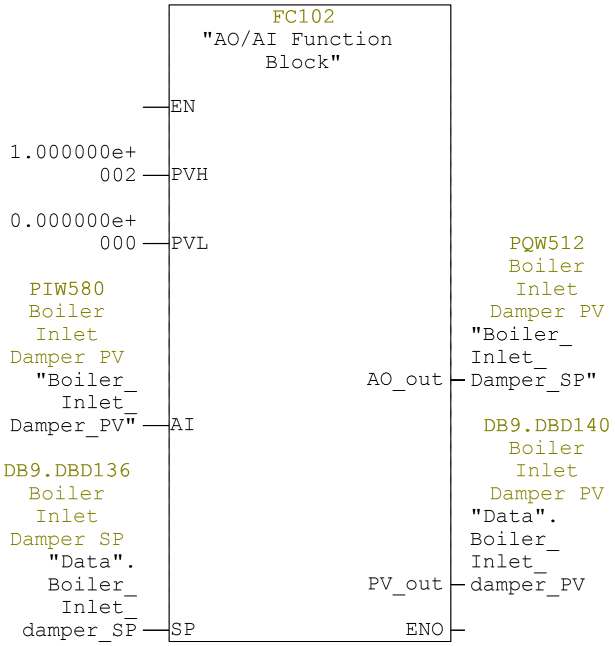
Network: 34



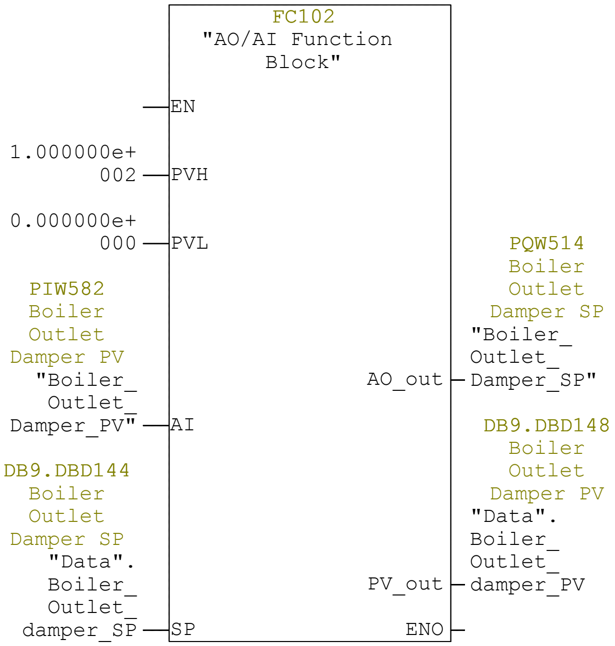
Network: 35



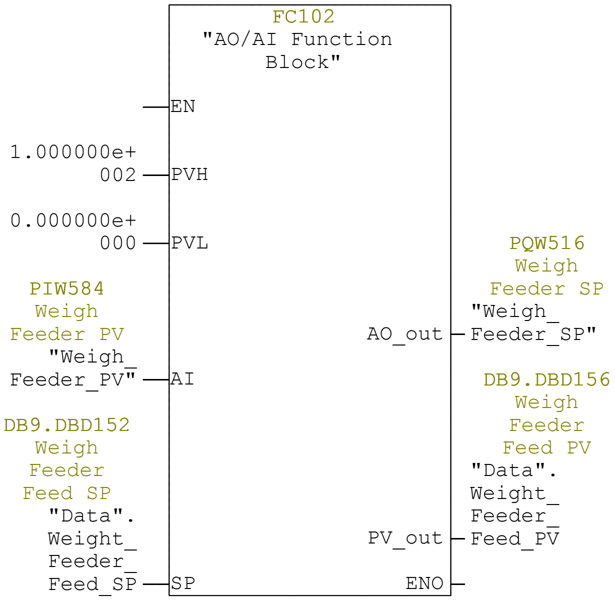
Network: 36



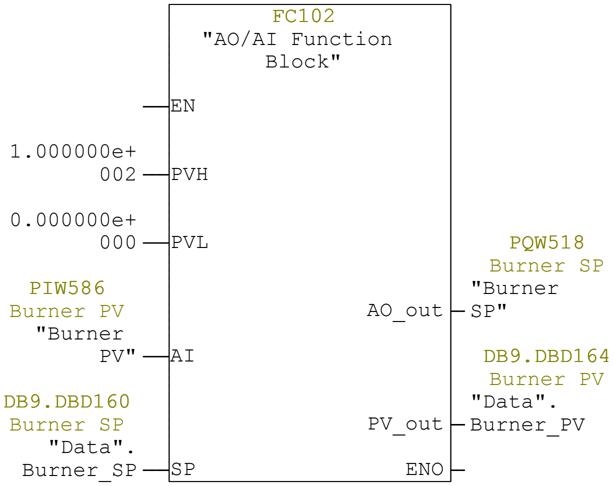
Network: 37



Network: 38



Network: 39



FC4 - <offline>

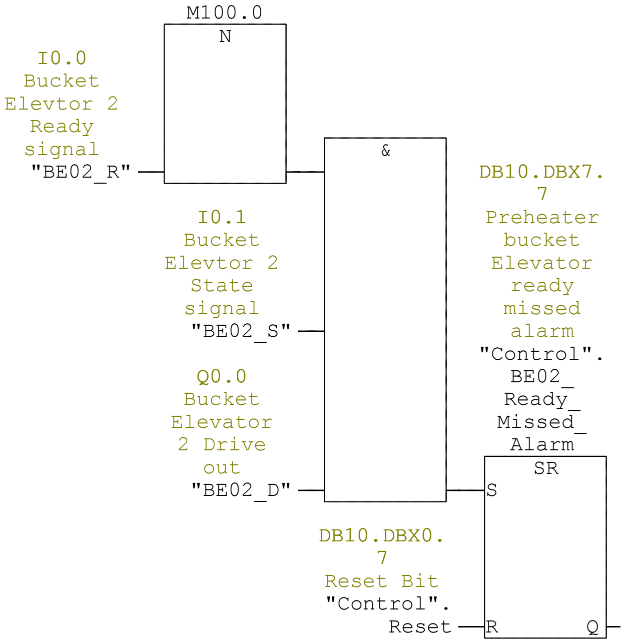
"Alarm"

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 05/06/2022 12:49:11 PM
Interface: 05/04/2022 10:07:58 AM
Lengths (block/logic/data): 01246 01090 00000

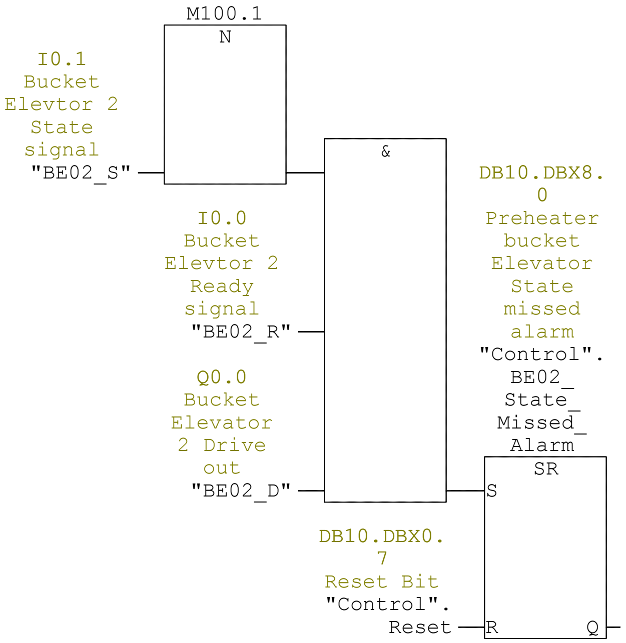
Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC4

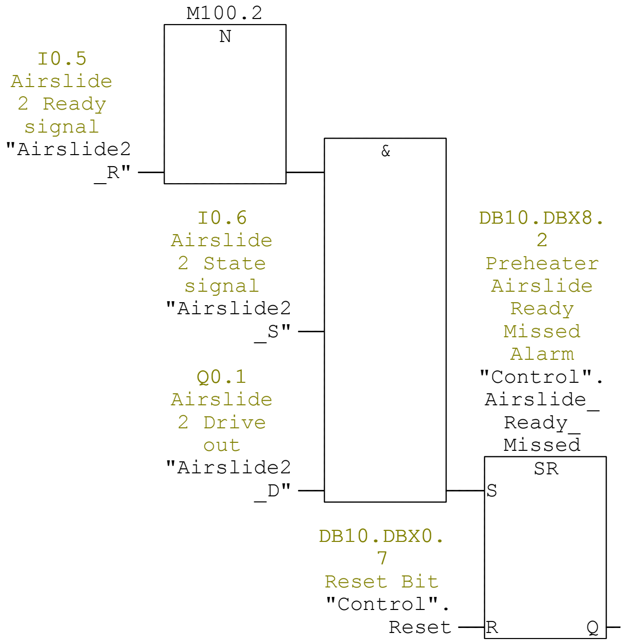
Network: 1 Preheater bucket Elevator ready missed alarm



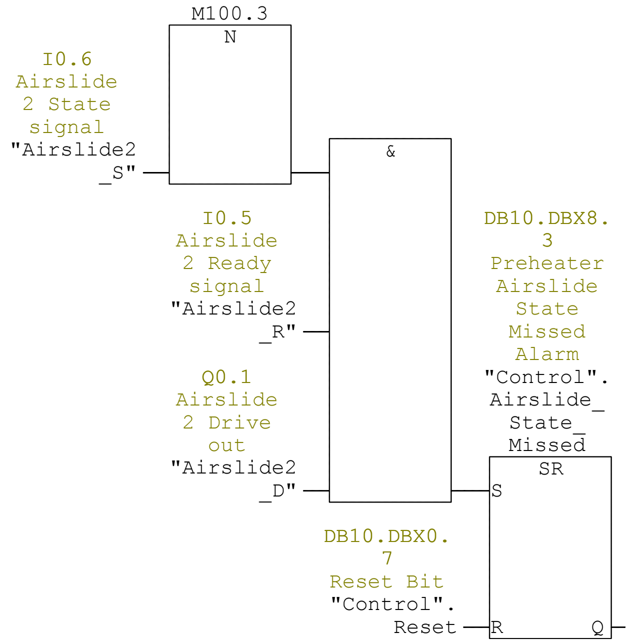
Network: 2 Preheater bucket Elevator State missed alarm



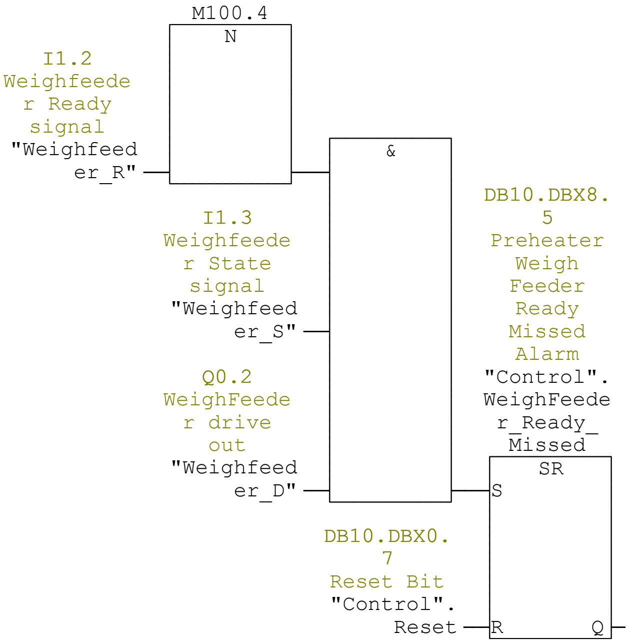
Network: 3



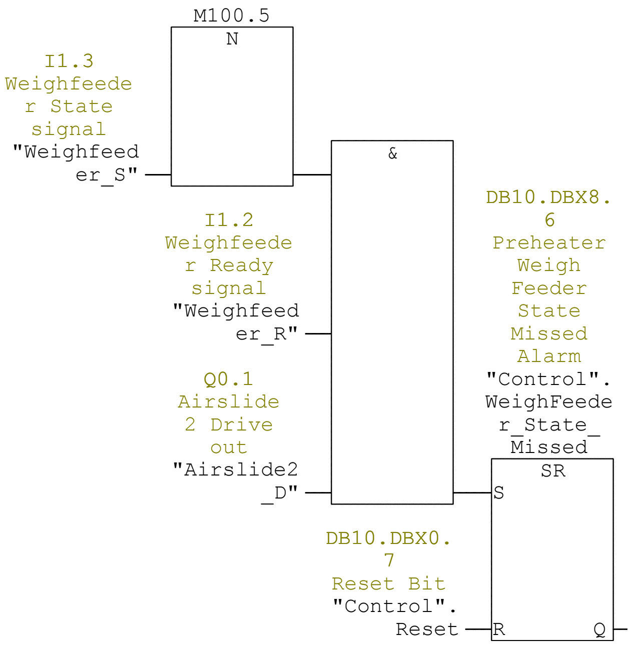
Network: 4



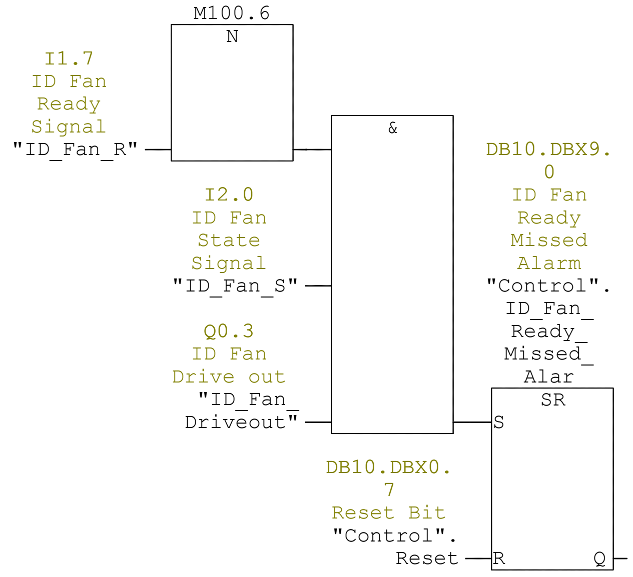
Network: 5



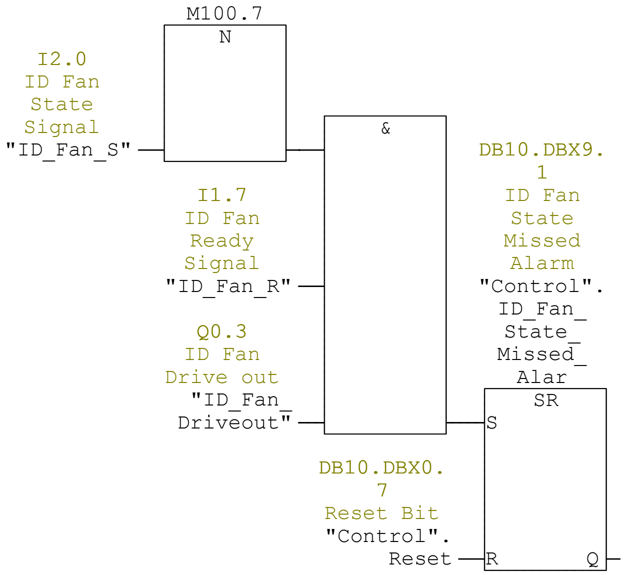
Network: 6



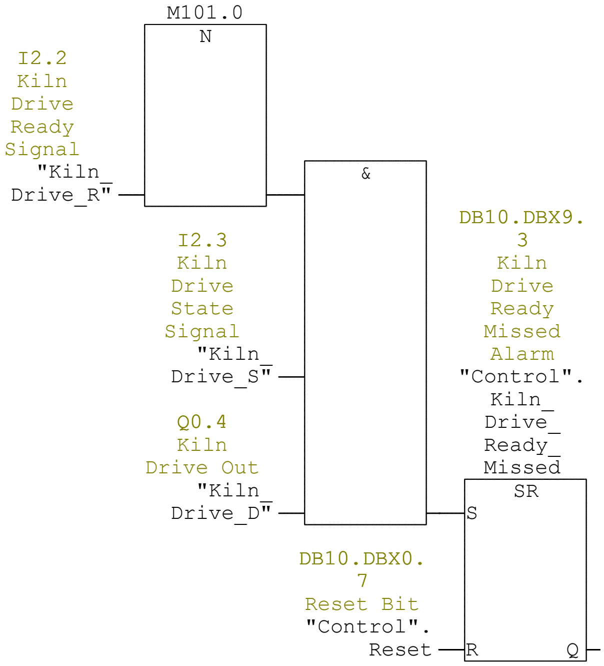
Network: 7



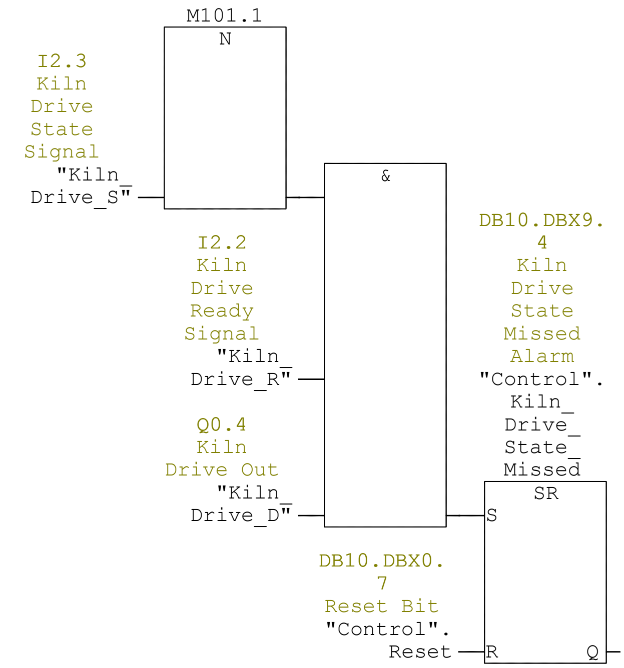
Network: 8



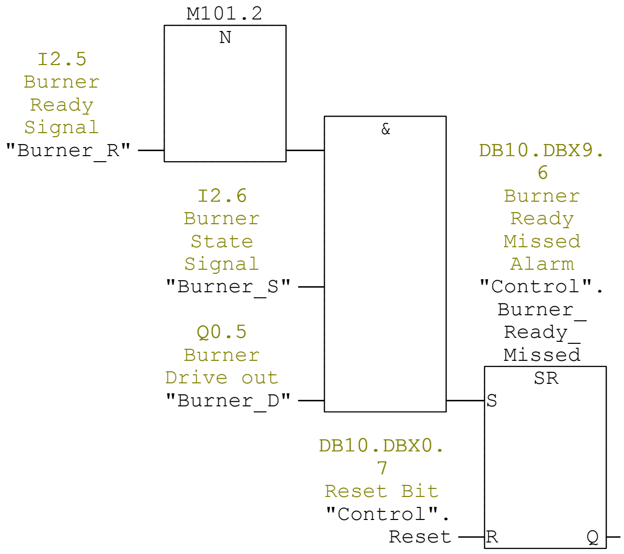
Network: 9



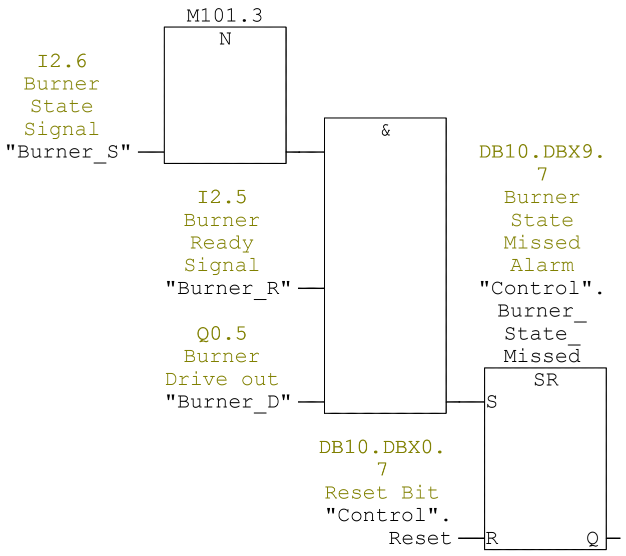
Network: 10



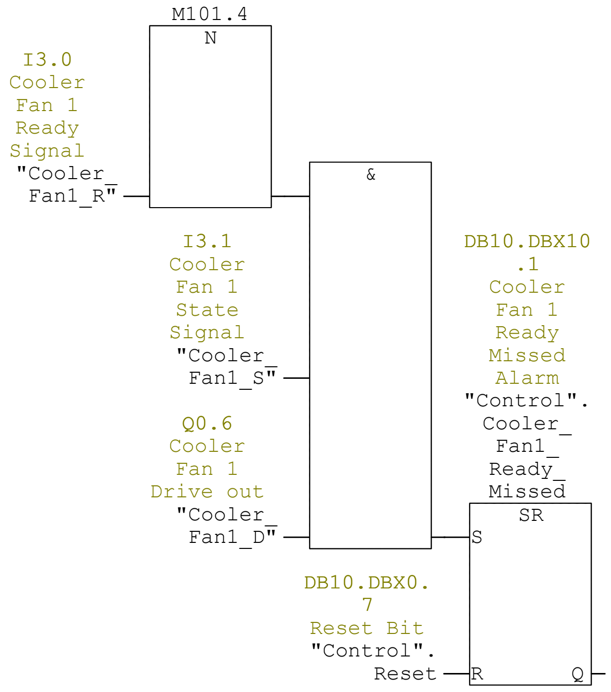
Network: 11



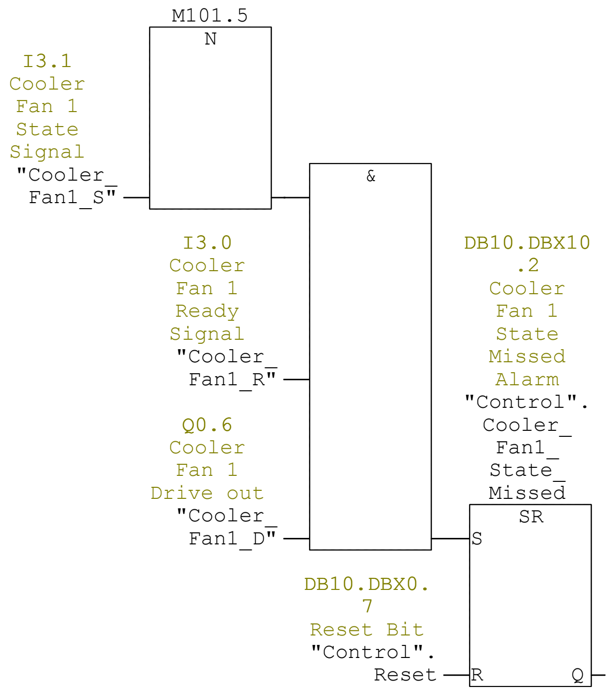
Network: 12 Burner State Missed Alarm



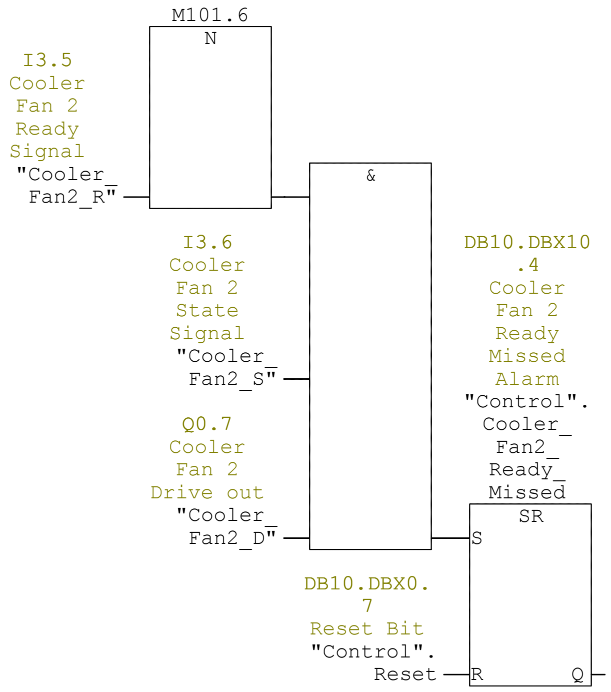
Network: 13



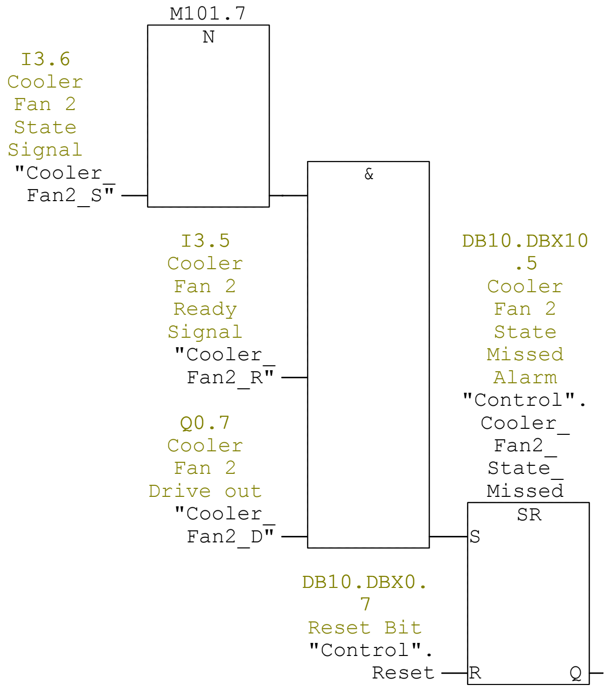
Network: 14



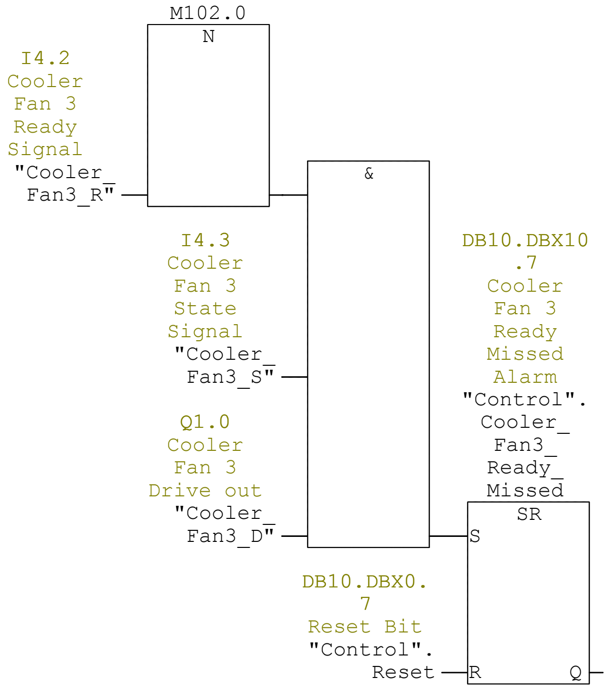
Network: 15



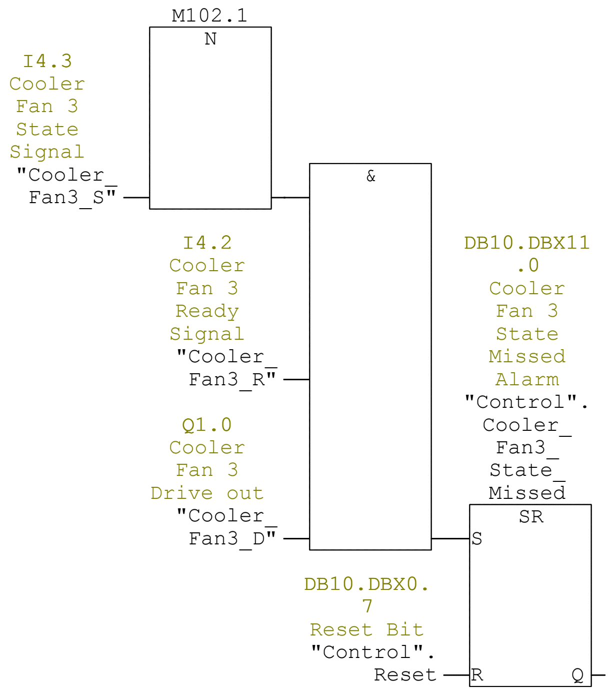
Network: 16



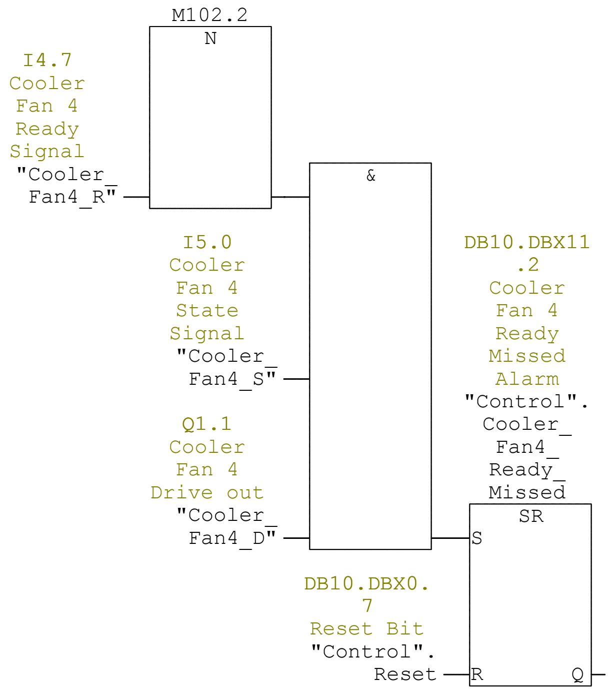
Network: 17



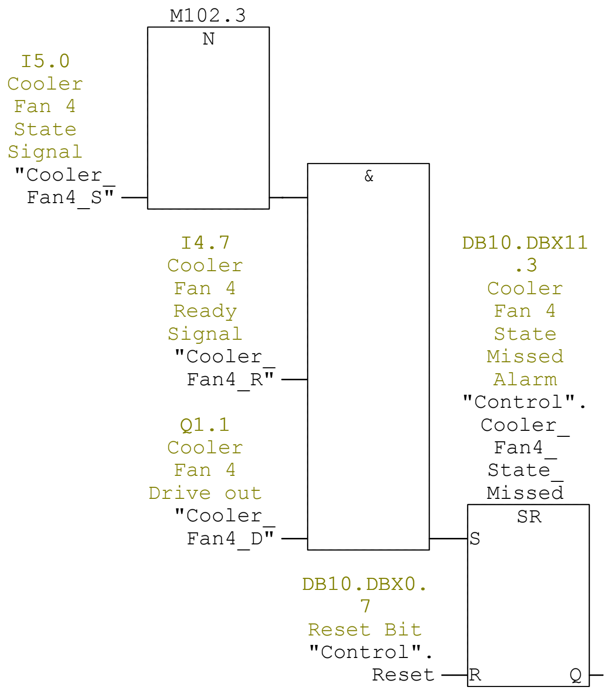
Network: 18



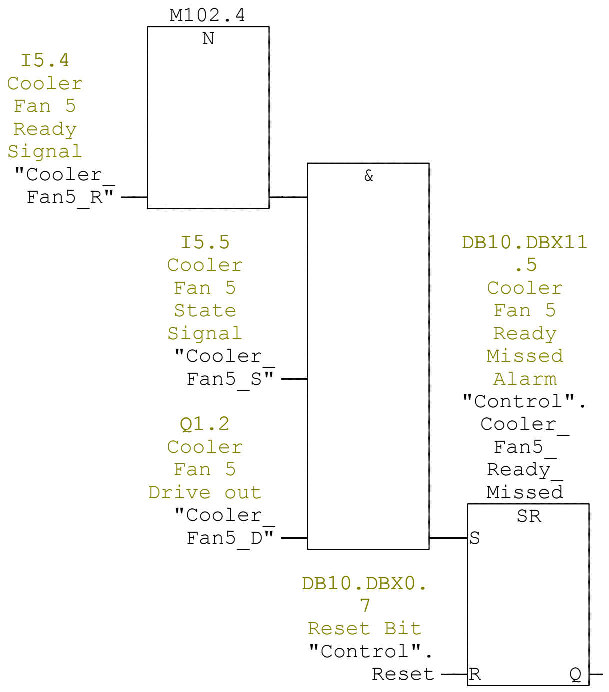
Network: 19



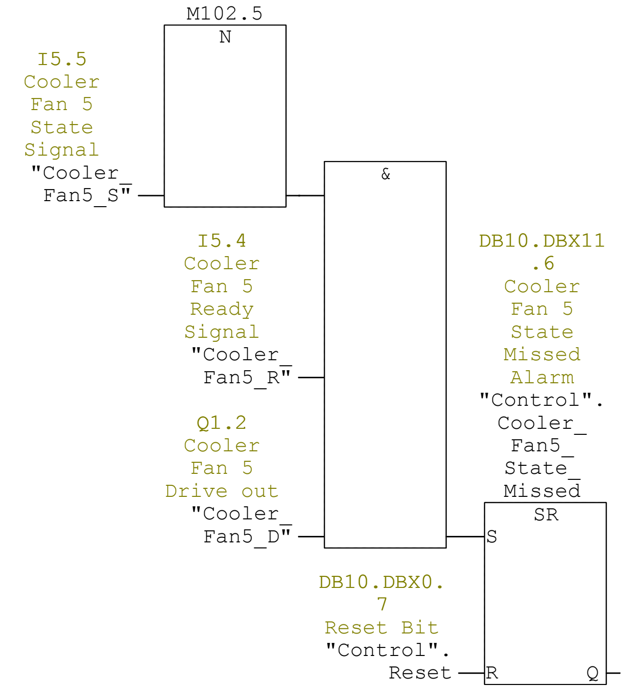
Network: 20



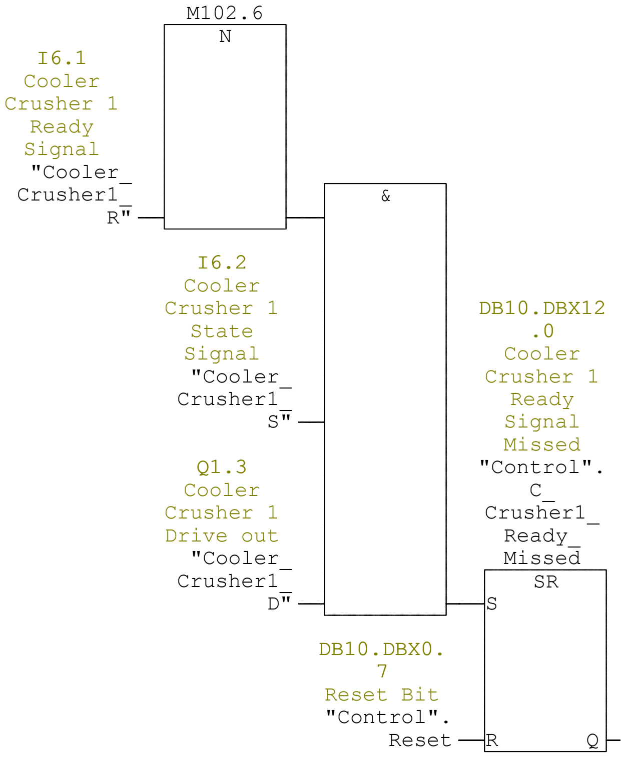
Network: 21



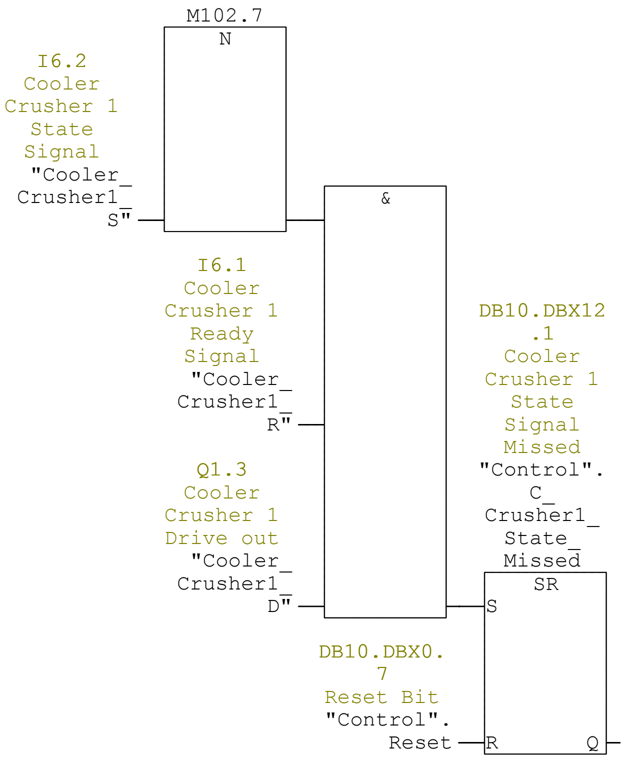
Network: 22



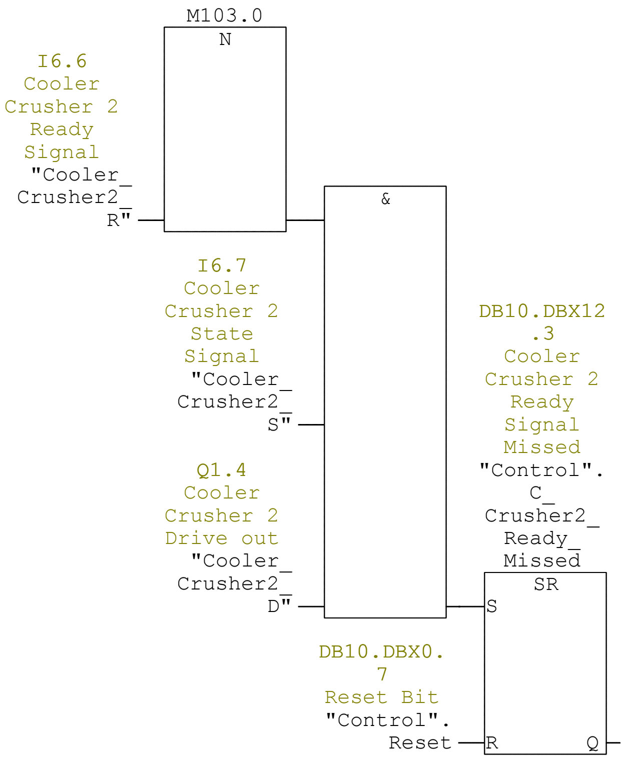
Network: 23 Cooler Crusher 1 Ready Signal Missed



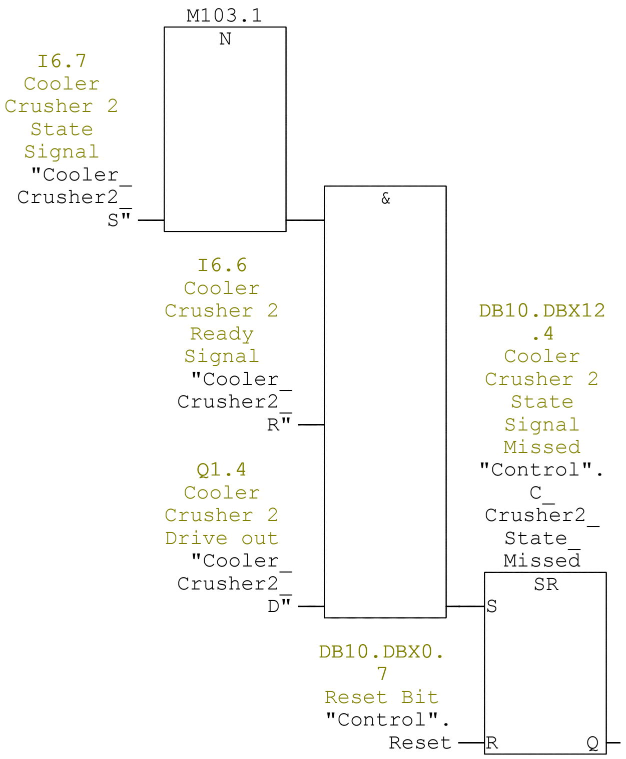
Network: 24 Cooler Crusher 1 State Signal Missed


M102.7
N
I6.2
Cooler Crusher 1 State Signal
"Cooler_Crusher1_S"
&
I6.1
Cooler Crusher 1 Ready Signal
"Cooler_Crusher1_R"
Q1.3
Cooler Crusher 1 Drive out
"Cooler_Crusher1_D"
DB10.DBX12.1
Cooler Crusher 1 State Signal Missed
"Control".C_Crusher1_State_Missed
SR
S
DB10.DBX0.7
Reset Bit
"Control".Reset
R
Q

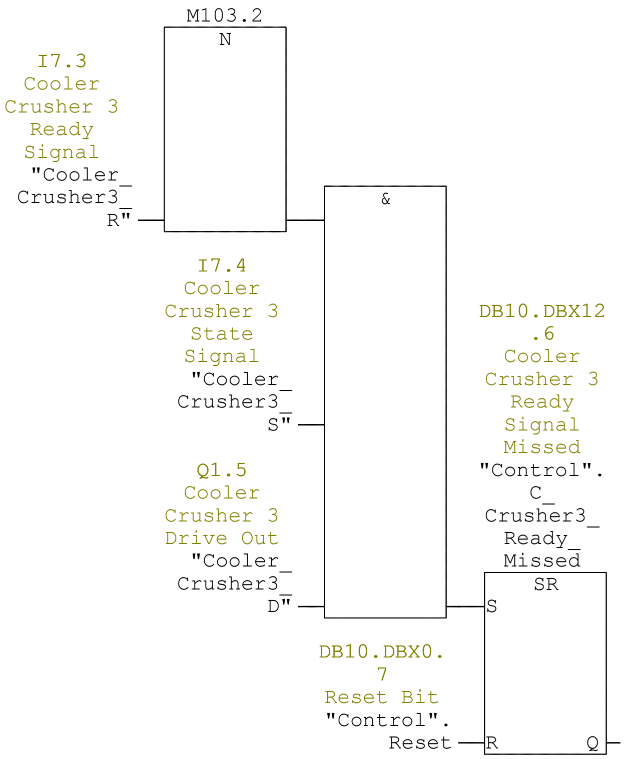
Network: 25


M103.0
N
I6.6
Cooler Crusher 2 Ready Signal
"Cooler_Crusher2_R"
&
I6.7
Cooler Crusher 2 State Signal
"Cooler_Crusher2_S"
Q1.4
Cooler Crusher 2 Drive out
"Cooler_Crusher2_D"
DB10.DBX12.3
Cooler Crusher 2 Ready Signal Missed
"Control".C_Crusher2_Ready_Missed
SR
S
DB10.DBX0.7
Reset Bit
"Control".Reset
R
Q

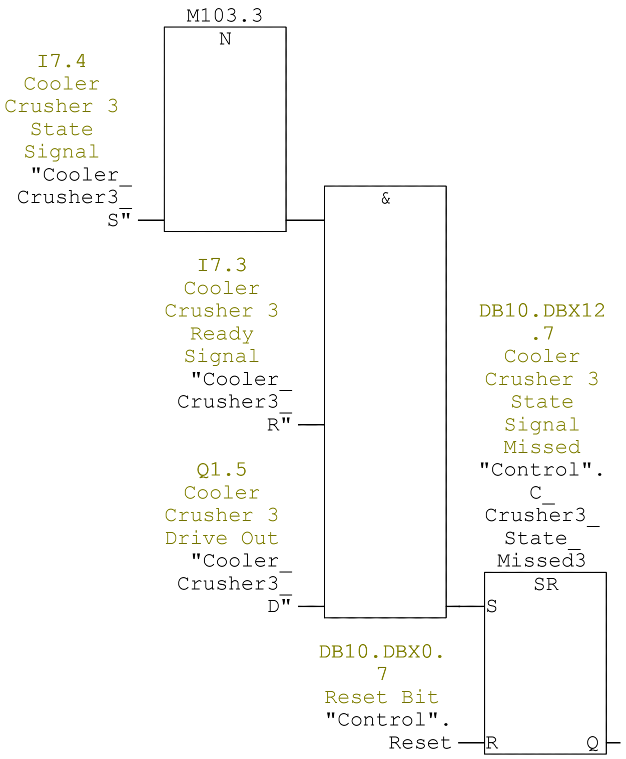
Network: 26



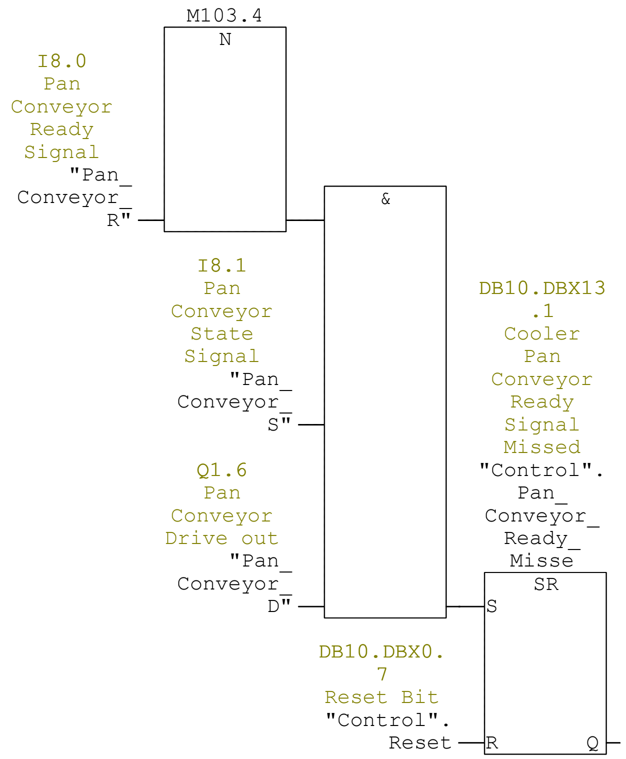
Network: 27



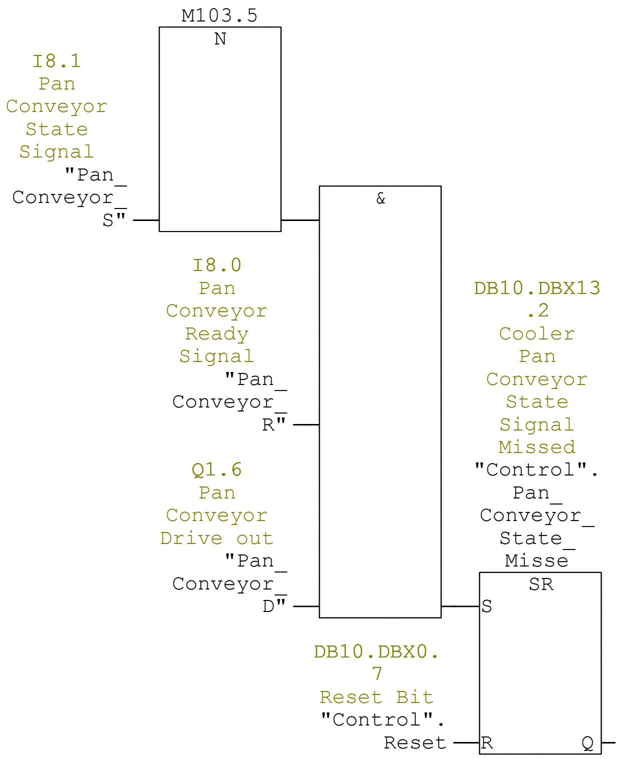
Network: 28 Cooler Crusher 3 State Signal Missed



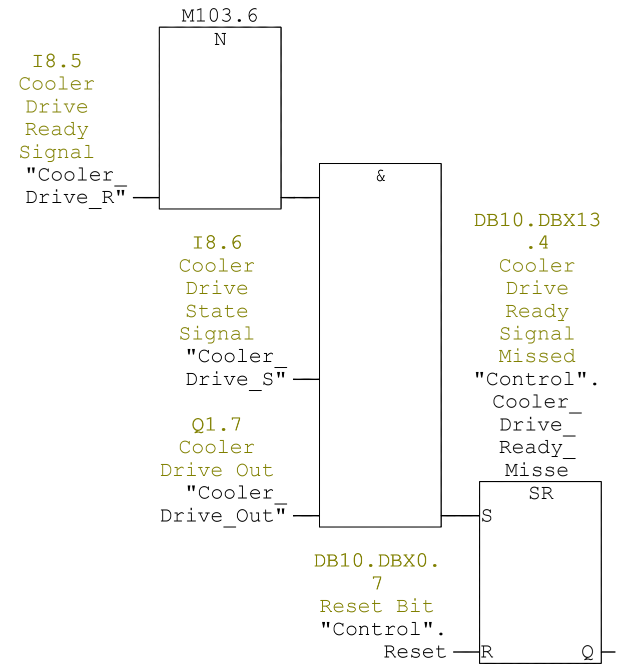
Network: 29



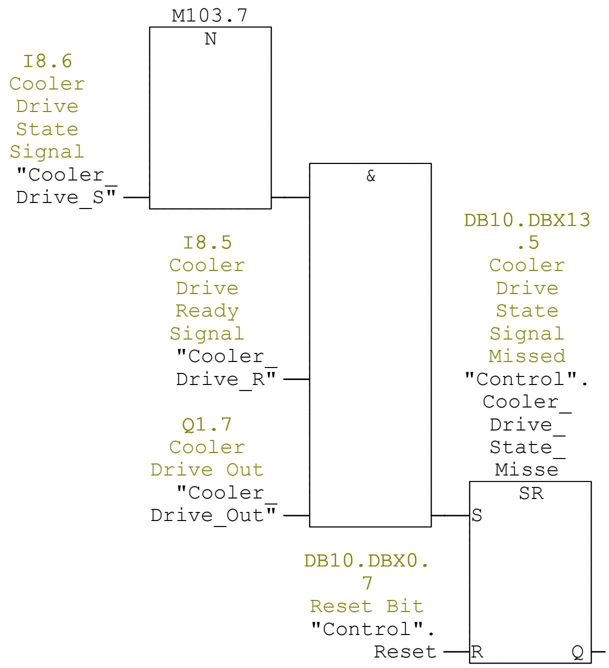
Network: 30



Network: 31



Network: 32



Network: 33 Preheater Bucket Elevator Electrical Cabinet Fault

```
      A     "BE02_A"                        I0.4              -- Bucket Elevtor 2 Alarm signal
      =     "Control".BE02_Electrical_Fault  DB10.DBX8.1       -- Preheater Bucket Elevator Electrical Cabinet Fault
      A     "Airslide2_A"                   I1.1              -- Airslide 2 Alarm signal
      =     "Control".Airslide_Electrical_Faul DB10.DBX8.4     -- Preheater Airslide Electrical Cabinet Fault
      A     "Weighfeeder_A"                 I1.6              -- Weighfeeder Alarm signal
      =     "Control".WeighFeeder_Electrical_F DB10.DBX8.7     -- Preheater Weigh Feeder Electrical Cabinet Fault
      A     "ID_Fan_A"                      I2.1              -- Id Fan Alarm Signal
      =     "Control".ID_Fan_Electrical_Fault DB10.DBX9.2      -- ID Fan Electrical Cabinet Fault
      A     "Kiln_Drive_A"                  I2.4              -- Kiln Drive Alarm Signal
      =     "Control".Kiln_Drive_Electrical  DB10.DBX9.5       -- Kiln Drive Electrical Cabinet Fault
      A     "Burner_A"                      I2.7              -- Burner Alarm Signal
      =     "Control".Burner_Electrical_Fault DB10.DBX10.0     -- Burner Electrical Cabinet Fault
      A     "Cooler_Fan1_Alarm"             I3.4              -- Cooler Fan 1 Alarm Signal
      =     "Control".Cooler_Fan1_Electrical_F DB10.DBX10.3    -- Cooler Fan 1 Electrical Cabinet Fault
      A     "Cooler_Fan2_Alarm"             I4.1              -- Cooler Fan 2 Alarm Signal
      =     "Control".Cooler_Fan2_Electrical_F DB10.DBX10.6    -- Cooler Fan 2 Electrical Cabinet Fault
      A     "Cooler_Fan3_Alarm"             I4.6              -- Cooler Fan 3 Alarm Signal
      =     "Control".Cooler_Fan3_Electrical_F DB10.DBX11.1    -- Cooler Fan 3 Electrical Cabinet Fault
      A     "Cooler_Fan4_Alarm"             I5.3              -- Cooler Fan 4 Alarm Signal
      =     "Control".Cooler_Fan4_Electrical_F DB10.DBX11.4    -- Cooler Fan 4 Electrical Cabinet Fault
      A     "Cooler_Fan5_Alarm"             I6.0              -- Cooler Fan 5 Alarm Signal
      =     "Control".Cooler_Fan5_Electrical_F DB10.DBX11.7    -- Cooler Fan 5 Electrical Cabinet Fault
      A     "Cooler_Crusher1 _Alarm"        I6.5              -- Cooler Crusher 1 Alarm Signal
      =     "Control".C_Crusher1_Electrical_F DB10.DBX12.2     -- Cooler Crusher 1 Electrical Cabinet Fault
      A     "Cooler_Crusher2 _Alarm"        I7.2              -- Cooler Crusher 2 Alarm Signal
      =     "Control".C_Crusher2_Electrical_F DB10.DBX12.5     -- Cooler Crusher 2 Electrical Cabinet Fault
      A     "Cooler_Crusher3 _Alarm"        I7.7              -- Cooler Crusher 3 Alarm Signal
      =     "Control".C_Crusher3_Electrical_F DB10.DBX13.0     -- Cooler Crusher 3 Electrical Cabinet Fault
      A     "Pan_Conveyor_A"                I8.4              -- Pan Conveyor Alarm
      =     "Control".Pan_Conveyor_Electrical DB10.DBX13.3     -- Cooler Pan COnveyor Electrical Cabine Fault
      A     "Cooler_Drive_A"                I8.7              -- Cooler Drive Alarm Signal
      =     "Control".Cooler_Drive_Electrical DB10.DBX13.6     -- Cooler Drive Electrical Cabinet Fault
```