

Universidade Federal  
do Rio de Janeiro

---

Escola Politécnica

PROJETO DE UM SISTEMA SUPERVISÓRIO PARA UMA PLANTA  
MECATRÔNICA DE ESTOCAGEM DE PEÇAS

Tiago Cardoso França

Projeto de Graduação apresentado ao Corpo Docente do Departamento de Engenharia Elétrica da Escola Politécnica da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro Eletricista.

Orientador: Marcos Vicente de Brito Moreira

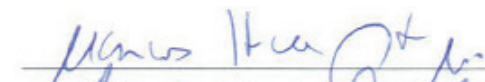
Rio de Janeiro  
Março de 2015

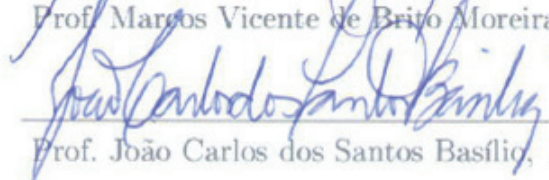
PROJETO DE UM SISTEMA SUPERVISÓRIO PARA UMA PLANTA  
MECATRÔNICA DE ESTOCAGEM DE PEÇAS

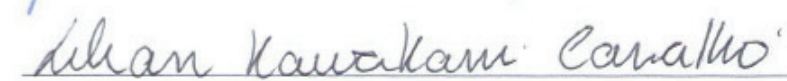
Tiago Cardoso França

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE  
DO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA ESCOLA  
POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO  
COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO  
GRAU DE ENGENHEIRO ELETRICISTA.

Examinado por:

  
Prof. Marcos Vicente de Brito Moreira, D.Sc. (Orientador)

  
Prof. João Carlos dos Santos Basílio, Ph.D.

  
Prof. Lilian Kawakami Carvalho, D.Sc.

RIO DE JANEIRO, RJ – BRASIL  
MARÇO DE 2015

França, Tiago Cardoso

Projeto de um Sistema Supervisório para uma Planta Mecatrônica de Estocagem de Peças / Tiago Cardoso França. – Rio de Janeiro: UFRJ/Escola Politécnica, 2015. XV, 182 p.: il.; 29, 7cm.

Orientador: Marcos Vicente de Brito Moreira

Projeto de Graduação – UFRJ/Escola Politécnica/ Departamento de Engenharia Elétrica, 2015.

Referências Bibliográficas: p. 98 – 98.

1. Interface. 2. Homem-Máquina. 3. IHM. 4. CLP. 5. HMI. 6. PLC. I. Moreira, Marcos Vicente de Brito. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Departamento de Engenharia Elétrica. III. Projeto de um Sistema Supervisório para uma Planta Mecatrônica de Estocagem de Peças.

*“Agente firme, saiba quem você  
é e siga seu próprio caminho,  
que tudo ficará bem.”  
Mraz, Jason.*

# Agradecimentos

Ao professor Marcos Moreira pela exímia orientação durante a execução do projeto final e pelas excelentes aulas que me motivaram a escolher esta área.

Aos todos os meus familiares, em especial aos meus pais, por sempre me instruírem a fazer meu melhor em tudo e para todos, e ao meu irmão, que sempre tiramos dúvidas um do outro, o que me fez crescer meus conhecimentos em engenharia elétrica.

A todos os meus colegas de turma que tanto me ajudaram a realizar os trabalhos da faculdade e a estudar para as provas.

A todos meus amigos da vida, que me trouxeram muitos momentos felizes enquanto relaxava entre os estudos e que também me trouxeram momentos felizes durante os estudos.

A minha namorada, que, mesmo de longe, me apoiou de todas as maneiras possíveis para a realização deste projeto, me insentivando e me forçando a parar de procrastinar.

A todos, este projeto foi realizado por mim, mas foi feito por vocês.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Eletricista

## PROJETO DE UM SISTEMA SUPERVISÓRIO PARA UMA PLANTA MECATRÔNICA DE ESTOCAGEM DE PEÇAS

Tiago Cardoso França

Março/2015

Orientador: Marcos Vicente de Brito Moreira

Departamento: Engenharia Elétrica

Apresenta-se, neste projeto de conclusão de curso, o desenvolvimento de um tutorial de um sistema supervisório, utilizando uma interface homem-máquina para uma planta mecatrônica de estocagem de peças. O sistema de estocagem de peças é utilizado para o armazenamento de blocos de forma ordenada em uma prateleira, a retirada das peças da prateleira de forma arbitrária e a realocação das peças na prateleira também de forma ordenada preenchendo as lacunas. O sistema é acionado com o auxílio de uma esteira, um braço giratório e um elevador móvel. O sistema supervisório tem como intuito observar, de forma remota, os estados da planta mecatrônica e acionar o sistema.

Abstract of Graduation Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Electrical Engineer

## SUPERVISORY SYSTEM PROJECT FOR A MECHATRONIC STATION OF BLOCK STORAGE

Tiago Cardoso França

March/2015

Advisor: Marcos Vicente de Brito Moreira

Department: Electrical Engineering

In this work it is presented a tutorial of a supervisory system, using a human-machine interface, for a mechatronic storage plant. This system is used for the storage of cubes in a shelf, the removal of the cubes from the shelf, arbitrarily, and the replacement of the cubes in the shelf filling the gaps. The system is composed of a conveyor, a robotic arm and a lift. The supervisory system aims to remotely observe the states of the mechatronic plant and control the system.

# Sumário

|                                                                            |            |
|----------------------------------------------------------------------------|------------|
| <b>Lista de Figuras</b>                                                    | <b>xii</b> |
| <b>Lista de Tabelas</b>                                                    | <b>xv</b>  |
| <b>1 Introdução</b>                                                        | <b>1</b>   |
| 1.1 Principais dispositivos de um sistema automatizado . . . . .           | 1          |
| 1.2 Redes industriais: PROFINET . . . . .                                  | 2          |
| 1.3 Motivação . . . . .                                                    | 2          |
| 1.4 Objetivos . . . . .                                                    | 3          |
| <b>2 Fundamento Teórico da Lógica Ladder</b>                               | <b>4</b>   |
| 2.1 Funcionamento básico do diagrama Ladder . . . . .                      | 4          |
| 2.2 Representação dos elementos básicos de um diagrama Ladder . . . . .    | 5          |
| 2.2.1 Contatos . . . . .                                                   | 6          |
| 2.2.2 Bobinas . . . . .                                                    | 7          |
| 2.2.3 Temporizadores . . . . .                                             | 8          |
| 2.2.4 Contadores de alta velocidade . . . . .                              | 9          |
| 2.2.5 Instruções de comparação . . . . .                                   | 10         |
| 2.3 Utilização do diagrama Ladder em dispositivos como CLP e IHM . . . . . | 11         |
| <b>3 Projeto Automatizado da Planta Mecatrônica</b>                        | <b>13</b>  |
| 3.1 Descrição da planta mecatrônica e de seus componentes . . . . .        | 13         |
| 3.2 Descrição do sistema automatizado . . . . .                            | 18         |
| <b>4 Instalação dos Dispositivos de Controle e Supervisão</b>              | <b>22</b>  |
| 4.1 Conexão da IHM, do CLP e do computador . . . . .                       | 22         |
| 4.2 Criação de um projeto . . . . .                                        | 24         |
| 4.3 Selecionando o CLP . . . . .                                           | 27         |
| 4.4 Selecionando a IHM . . . . .                                           | 29         |
| 4.5 Configuração do número de IP . . . . .                                 | 29         |

|          |                                                                                          |           |
|----------|------------------------------------------------------------------------------------------|-----------|
| <b>5</b> | <b>Projeto do Sistema Supervisório da Planta Mecatrônica</b>                             | <b>33</b> |
| 5.1      | Esboço do sistema supervisório . . . . .                                                 | 33        |
| 5.2      | Elementos de um projeto . . . . .                                                        | 35        |
| 5.3      | Definição de TAGs . . . . .                                                              | 35        |
| 5.3.1    | TAG do CLP . . . . .                                                                     | 36        |
| 5.3.2    | TAG da IHM . . . . .                                                                     | 38        |
| 5.4      | Telas . . . . .                                                                          | 43        |
| 5.4.1    | Criação de telas . . . . .                                                               | 43        |
| 5.4.2    | Propriedades, animações e eventos de uma tela . . . . .                                  | 44        |
| 5.5      | Passo 1: Adicionar um novo dispositivo . . . . .                                         | 45        |
| 5.6      | Passo 2: Criação de template . . . . .                                                   | 50        |
| 5.6.1    | Figuras . . . . .                                                                        | 51        |
| 5.6.2    | Texto . . . . .                                                                          | 52        |
| 5.6.3    | Botão . . . . .                                                                          | 53        |
| 5.6.4    | Interruptor . . . . .                                                                    | 54        |
| 5.7      | Passo 3: Edição de telas . . . . .                                                       | 56        |
| 5.7.1    | “Tela Inicial” . . . . .                                                                 | 56        |
| 5.7.2    | “Visão geral” . . . . .                                                                  | 58        |
| 5.7.3    | “Tela esteira” . . . . .                                                                 | 60        |
| 5.7.4    | “Tela braço” . . . . .                                                                   | 61        |
| 5.7.5    | “Tela prateleira” . . . . .                                                              | 63        |
| 5.8      | Passo 4: Criação das funcionalidades dos elementos . . . . .                             | 64        |
| 5.8.1    | Botões funcionais . . . . .                                                              | 65        |
| 5.8.2    | Tela da esteira . . . . .                                                                | 65        |
| 5.8.3    | Tela do braço . . . . .                                                                  | 66        |
| 5.8.4    | Tela da prateleira . . . . .                                                             | 67        |
| 5.8.5    | Tela de visão geral . . . . .                                                            | 68        |
| 5.9      | Passo 5: Implementação do projeto . . . . .                                              | 69        |
| <b>6</b> | <b>Implementação do Módulo Manual</b>                                                    | <b>71</b> |
| 6.1      | Descrição do projeto manual . . . . .                                                    | 71        |
| 6.1.1    | Comando de acionar pistão . . . . .                                                      | 72        |
| 6.1.2    | Comandos de ligar esteira para a esquerda e de ligar esteira<br>para a direita . . . . . | 72        |
| 6.1.3    | Comando de calibrar o braço . . . . .                                                    | 73        |
| 6.1.4    | Comando de girar o braço . . . . .                                                       | 73        |
| 6.1.5    | Comando de o braço pegar peça da esteira . . . . .                                       | 73        |
| 6.1.6    | Comando de o braço entregar peça para o elevador . . . . .                               | 74        |
| 6.1.7    | Comando de o braço pegar peça do elevador . . . . .                                      | 74        |

|          |                                                                 |           |
|----------|-----------------------------------------------------------------|-----------|
| 6.1.8    | Comando de o elevador retirar peça da prateleira . . . . .      | 74        |
| 6.1.9    | Retorno do modo manual para o modo automático . . . . .         | 74        |
| 6.2      | Passo 1: Edição de telas e template . . . . .                   | 75        |
| 6.2.1    | Template 2 . . . . .                                            | 75        |
| 6.2.2    | Tela da esteira . . . . .                                       | 77        |
| 6.2.3    | Tela do braço . . . . .                                         | 77        |
| 6.3      | Passo 2: Funcionalidade dos botões . . . . .                    | 78        |
| 6.3.1    | Lógica da ação do pistão . . . . .                              | 79        |
| 6.3.2    | Lógica da ação de ligar a esteira para a esquerda . . . . .     | 80        |
| 6.3.3    | Lógica da ação de ligar a esteira para a direita . . . . .      | 80        |
| 6.3.4    | Lógica da ação de calibragem do braço . . . . .                 | 80        |
| 6.3.5    | Lógica da ação de girar o braço . . . . .                       | 80        |
| 6.3.6    | Lógica da ação de o braço pegar ou entregar peça à esteira . .  | 81        |
| 6.3.7    | Lógica da ação de o braço entregar peça para o elevador . . .   | 81        |
| 6.3.8    | Lógica da ação de o braço pegar a peça do elevador . . . . .    | 82        |
| 6.3.9    | Lógica da ação de o elevador retirar peça da prateleira . . . . | 82        |
| 6.3.10   | Lógica de retorno do modo manual para o modo automático .       | 83        |
| 6.4      | Passo 3: Criação dos alarmes . . . . .                          | 83        |
| 6.4.1    | Alarmes . . . . .                                               | 83        |
| 6.4.2    | Criação das TAGs dos alarmes e lógicas de acionamento . . .     | 86        |
| 6.4.3    | Criação dos alarmes . . . . .                                   | 87        |
| 6.4.4    | Avisos de alarme . . . . .                                      | 88        |
| <b>7</b> | <b>Funções e Objetos Não Utilizados</b>                         | <b>90</b> |
| 7.1      | Objetos inanimados . . . . .                                    | 90        |
| 7.1.1    | Linhas . . . . .                                                | 90        |
| 7.1.2    | Círculos, elipses e retângulos . . . . .                        | 91        |
| 7.2      | Objetos animados . . . . .                                      | 91        |
| 7.2.1    | Barra . . . . .                                                 | 91        |
| 7.2.2    | Gráfico . . . . .                                               | 92        |
| 7.3      | Listas . . . . .                                                | 93        |
| 7.3.1    | Lista de usuários . . . . .                                     | 93        |
| 7.3.2    | Lista de receitas . . . . .                                     | 93        |
| 7.4      | Receitas . . . . .                                              | 94        |
| 7.4.1    | Ingredientes . . . . .                                          | 94        |
| 7.4.2    | Instruções . . . . .                                            | 94        |
| 7.5      | Agenda de tarefas . . . . .                                     | 95        |
| <b>8</b> | <b>Conclusão</b>                                                | <b>96</b> |
| 8.1      | Sugestões de trabalhos futuros . . . . .                        | 97        |

|                                                          |            |
|----------------------------------------------------------|------------|
| <b>Referências Bibliográficas</b>                        | <b>98</b>  |
| <b>Índice Remissivo</b>                                  | <b>99</b>  |
| <b>A Lógica Ladder Utilizada no Sistema Automatizado</b> | <b>100</b> |
| A.1 Módulo de Inicialização . . . . .                    | 100        |
| A.2 Módulo dos Eventos . . . . .                         | 103        |
| A.3 Módulo das Condições . . . . .                       | 106        |
| A.4 Módulo da Dinâmica . . . . .                         | 130        |
| A.5 Módulo das Ações . . . . .                           | 164        |

# Lista de Figuras

|      |                                                                               |    |
|------|-------------------------------------------------------------------------------|----|
| 2.1  | Exemplo de estrutura Ladder com contatos e bobinas. . . . .                   | 5  |
| 2.2  | Representação do contato normalmente aberto. . . . .                          | 6  |
| 2.3  | Representação do contato normalmente fechado. . . . .                         | 6  |
| 2.4  | Representação do contato de borda de subida. . . . .                          | 7  |
| 2.5  | Representação do contato de borda de descida. . . . .                         | 7  |
| 2.6  | Representação da bobina simples. . . . .                                      | 7  |
| 2.7  | Representação da bobina SET. . . . .                                          | 8  |
| 2.8  | Representação da bobina RESET. . . . .                                        | 8  |
| 2.9  | Representação do bloco temporizador de pulso. [3] . . . . .                   | 9  |
| 2.10 | Sinais de entrada e saída do bloco temporizador de pulso. [3] . . . . .       | 9  |
| 2.11 | Representação do bloco temporizador TimerOn Delay. [3] . . . . .              | 9  |
| 2.12 | Sinais de entrada e saída do bloco temporizador TimerOn Delay. [3] . . . . .  | 10 |
| 2.13 | Representação do bloco temporizador TimerOff Delay. [3] . . . . .             | 10 |
| 2.14 | Sinais de entrada e saída do bloco temporizador TimerOff Delay. [3] . . . . . | 11 |
| 2.15 | Representação do contador de alta velocidade. . . . .                         | 12 |
| 2.16 | Representação geral do bloco comparador. . . . .                              | 12 |
| 3.1  | Planta mecatrônica Cube Assembly. [5] . . . . .                               | 14 |
| 3.2  | Módulos da planta mecatrônica Cube Assembly. . . . .                          | 14 |
| 3.3  | Módulo 1 - Reconhecimento de peça. [5] . . . . .                              | 15 |
| 3.4  | Sensores de reconhecimento de peça. [5] . . . . .                             | 16 |
| 3.5  | Módulo 2 - Processamento e transporte de peça. [5] . . . . .                  | 16 |
| 3.6  | Módulo 3 - Estocagem de peça. [5] . . . . .                                   | 17 |
| 3.7  | Dispositivos de controle e proteção da planta mecatrônica. [5] . . . . .      | 18 |
| 3.8  | Painel de acionamento da planta mecatrônica. [5] . . . . .                    | 19 |
| 3.9  | Numeração das posições da prateleira. . . . .                                 | 19 |
| 4.1  | Equipamentos SIEMENS do Laboratório de Controle e Automação da UFRJ. . . . .  | 23 |
| 4.2  | Esquema de conexões entre os dispositivos e o computador. . . . .             | 23 |
| 4.3  | Elementos do Portal view. . . . .                                             | 24 |

|      |                                                                        |    |
|------|------------------------------------------------------------------------|----|
| 4.4  | Elementos do Project view. . . . .                                     | 25 |
| 4.5  | Tela de criação de um novo projeto. . . . .                            | 26 |
| 4.6  | Tela “First Step”. Configuração de dispositivo. . . . .                | 26 |
| 4.7  | Como adicionar um dispositivo a partir do “Project tree”. . . . .      | 27 |
| 4.8  | Como adicionar um dispositivo a partir do “Overview”. . . . .          | 28 |
| 4.9  | Seleção do CLP. . . . .                                                | 28 |
| 4.10 | Seleção do IHM. . . . .                                                | 29 |
| 4.11 | Propriedades de conexão local. . . . .                                 | 30 |
| 4.12 | Propriedades de protocolo TCP/IP. . . . .                              | 30 |
| 4.13 | Número de IP do CLP. . . . .                                           | 31 |
| 4.14 | Número de IP da IHM. . . . .                                           | 31 |
| 5.1  | Criando uma nova TAG do CLP. . . . .                                   | 37 |
| 5.2  | Determinando o tamanho da TAG - Booleana. . . . .                      | 38 |
| 5.3  | Determinando o tamanho da TAG - Inteira. . . . .                       | 38 |
| 5.4  | Determinando o endereço da TAG - Booleana. . . . .                     | 39 |
| 5.5  | Determinando o tipo da TAG - Booleana. . . . .                         | 39 |
| 5.6  | Determinando o endereço e o tipo da TAG - Inteira. . . . .             | 40 |
| 5.7  | Determinando o endereço e o tipo da TAG - Inteira. . . . .             | 40 |
| 5.8  | Criando uma nova TAG da IHM. . . . .                                   | 41 |
| 5.9  | Determinando o tamanho da TAG - Interna. . . . .                       | 41 |
| 5.10 | Determinando a TAG do CLP ou TAG de referência - Externa. . . .        | 42 |
| 5.11 | Identificação da TAG externa. . . . .                                  | 42 |
| 5.12 | Adicionando tela ao projeto da IHM. . . . .                            | 43 |
| 5.13 | Determinando evento de mudança de tela. . . . .                        | 44 |
| 5.14 | Configuração da IHM – Conexão com o CLP. . . . .                       | 45 |
| 5.15 | Configuração da IHM – Template 1. . . . .                              | 46 |
| 5.16 | Configuração da IHM – Avisos de alarme. . . . .                        | 47 |
| 5.17 | Configuração da IHM – Escolha da árvore de telas. . . . .              | 48 |
| 5.18 | Configuração da IHM – Árvore de telas escolhida para o projeto. . .    | 49 |
| 5.19 | Configuração da IHM – Escolha das telas do sistema. . . . .            | 50 |
| 5.20 | Configuração da IHM – Escolha dos botões do sistema. . . . .           | 51 |
| 5.21 | Configuração da IHM – Botões escolhidos para o projeto supervisorio. . | 52 |
| 5.22 | Configuração da IHM – Template 2 concluído. . . . .                    | 55 |
| 5.23 | Edição de telas – Tela inicial. . . . .                                | 59 |
| 5.24 | Edição de telas – Tela de visão geral. . . . .                         | 60 |
| 5.25 | Edição de telas – Tela da esteira. . . . .                             | 61 |
| 5.26 | Edição de telas – Tela do braço. . . . .                               | 63 |
| 5.27 | Edição de telas – Tela da prateleira. . . . .                          | 64 |

|      |                                                                          |    |
|------|--------------------------------------------------------------------------|----|
| 5.28 | Seleção e início de transferência do projeto para os dispositivos. . . . | 69 |
| 5.29 | Mensagem de parada do CLP para a transferência. . . . .                  | 69 |
| 6.1  | Módulo manual – Template 2 concluído. . . . .                            | 76 |
| 6.2  | Módulo manual – Tela da esteira. . . . .                                 | 77 |
| 6.3  | Módulo manual – Tela do braço. . . . .                                   | 78 |

# Lista de Tabelas

|     |                                                                 |    |
|-----|-----------------------------------------------------------------|----|
| 5.1 | Estrutura de endereços mnemônicos de sistemas de automação SIE- |    |
|     | MENS . . . . .                                                  | 36 |
| 5.2 | Tipo de variáveis de sistemas de automação SIEMENS . . . . .    | 37 |
| 5.3 | Tipo de variáveis aceitas pelas TAGs multiplex . . . . .        | 42 |
| 6.1 | Tipo de variáveis aceitas pelos alarmes . . . . .               | 84 |

# Capítulo 1

## Introdução

Os sistemas automatizados estão sendo cada vez mais usados por indústrias e equipamentos de alta tecnologia, como aviões e navios. Esses sistemas são utilizados para integrar, de forma funcional, os maquinários utilizados em um equipamento de alta tecnologia, reduzir a quantidade de mão de obra necessária em um processo industrial ou para supervisionar um sistema de forma a notificar o operador de possíveis falhas ou, até mesmo eliminar o problema, caso venha a ocorrer.

Em uma indústria de grande porte, se utiliza cada vez menos mão de obra para confeccionar o produto. Utiliza-se maquinários de grande porte que constituem um sistema integrado e automatizado, que executa as tarefas com necessidade de pouca interferência humana. A interferência humana pode ser necessária, mas não precisa ser presencial, já que traz risco para o andamento do sistema e, principalmente, para a integridade do operador. Uma forma de reduzir os riscos que uma interferência humana pode trazer é utilizar um sistema supervisorio para que o operador utilize-se de um controle remoto para acionar ou executar tarefas do sistema automatizado.

### 1.1 Principais dispositivos de um sistema automatizado

Os sistemas automatizados são constituídos de pequenas tarefas executadas em sequência, seguindo uma lógica pré-programada. Todas as tarefas e a ordem em que são executadas são comandadas por um computador lógico programável (CLP). O CLP é responsável pelo controle do sistema automatizado, recebendo sinais de sensores e dispositivos de acionamento, processando esses sinais de acordo com a lógica do sistema, e enviando sinais de saída para executar tarefas.

Mesmo com as tarefas sendo definidas pelo CLP, normalmente é necessário um operador para acionar alguns comandos do sistema ou para supervisionar o andamento dos processos. Visando a segurança do operador e uma maior capacidade de

acompanhamento dos processos, é necessário utilizar um sistema supervisório. O sistema supervisório é constituído de sensores e atuadores conectados a um painel de controle, que será manuseado pelo operador do sistema, também chamado de interface homem-máquina (IHM). Esse painel pode conter botoeiras, interruptores, lâmpadas sinalizadoras e até mesmo painéis luminosos com telas sensíveis ao toque.

## 1.2 Redes industriais: PROFINET

A comunicação entre os dispositivos de um sistema automatizado é feita por meio de uma rede industrial. Como descrito em [1], as redes industriais contribuem de forma significativa para os sistemas industriais, porque possuem grande confiabilidade e modularidade, facilidade de compreensão e, em grande parte dos casos, redução de custos. Outra vantagem é a possibilidade e facilidade de conectar diversos dispositivos sem perda na funcionalidade e na operacionalidade do sistema, permitindo trafegar informações pela rede sem perda por sobreposição.

As redes industriais seguem determinados protocolos visando garantir a interoperabilidade e interconexão entre os fabricantes. Esses protocolos consistem em padrões a serem seguidos que são determinados e divididos em sete camadas, que vão desde a camada física dos equipamentos e cabeamentos e da codificação dos dados até a camada de aplicação, onde os dados têm sua finalidade e é permitida a interação com o usuário.

No Laboratório de Controle e Automação (LCA) da Universidade Federal do Rio de Janeiro (UFRJ), o protocolo de rede utilizado é o PROFINET Component Based Automation (CBA) Transmission Control Protocol and Internet Protocol (TCP/IP), que trabalha com cabos de rede Ethernet.

A rede Ethernet permite facilitar o transporte de dados entre o chão de fábrica e os dispositivos mestres, porque não necessita de um tratamento e uma codificação dos dados. Esse rápido transporte de dados garante uma comunicação entre os componentes da rede em tempo real.

## 1.3 Motivação

Com a grande utilização de sistemas supervisórios nas indústrias, a UFRJ disponibiliza, para os alunos, equipamentos atuais e com alta tecnologia, sendo alguns deles no LCA, para enriquecer o aprendizado e dar instruções práticas de utilização de dispositivos para a implementação de sistemas de automação.

O desenvolvimento de um sistema supervisório em uma IHM pode não ser intuitivo, caso seja o primeiro contato de um aluno com o equipamento. Assim, com o intuito de melhor apresentar a IHM e suas funções, neste trabalho é apresentado um

tutorial listando, de forma prática, as funcionalidades deste dispositivo para auxiliar e dar base para futuros projetos em sistemas supervisórios.

## 1.4 Objetivos

Este projeto final tem o intuito de ensinar, por meio de um tutorial, como criar um projeto supervisório com base em um projeto de automação criado em um CLP. Este trabalho está organizado da seguinte forma: no capítulo 2, é apresentada uma introdução teórica sobre Ladder e suas funcionalidades; no capítulo 3, é apresentada uma descrição do sistema de estocagem de peças a ser utilizado como base para a criação do sistema supervisório; no capítulo 4, é apresentada uma explicação inicial de como conectar os equipamentos; no capítulo 5, é dado um exemplo de criação de um sistema supervisório baseado em um sistema de estocagem de peças em formato de tutorial; no capítulo 6, é apresentado um exemplo de extensão do projeto supervisório para uma atuação manual; no capítulo 7, são apresentadas as funções não utilizadas nos exemplos; e, no capítulo 8, é apresentada a conclusão.

# Capítulo 2

## Fundamento Teórico da Lógica Ladder

Neste capítulo serão apresentados os fundamentos básicos sobre a estrutura de funcionamento e sobre os elementos do diagrama Ladder [2]. Será apresentada, também, a utilização do diagrama Ladder em CLP e IHM.

### 2.1 Funcionamento básico do diagrama Ladder

O diagrama Ladder é uma estrutura computacional ou física que desencadeia ações ou sinais de resposta precedidos de certas condições. Seu funcionamento básico consiste em transmitir um sinal em uma rede energizada passando por contatos e energizando bobinas, que são capazes de emitir respostas. A abertura e o fechamento dos contatos determinam as condições de energização das bobinas, enquanto as bobinas determinam a abertura e o fechamento de um contato ou uma ação externa.

A estrutura do diagrama Ladder segue um padrão que assemelha-se a uma escada. Ele é formado por uma barra vertical à esquerda e pequenas barras horizontais saindo da barra vertical, que, por sua vez, se encontram e se dividem para terminar em bobinas. Nessas barras horizontais ficam localizados os contatos para determinar a condição de acionamento das bobinas.

Na figura 2.1, é possível observar um exemplo de estrutura Ladder em que os pares de linhas verticais são representações de contatos e as estruturas abauladas são representações de bobinas.

A lógica em um diagrama Ladder consiste em energizar a barra vertical para que sinais sejam enviados pelas barras horizontais e sigam a lógica de condição determinada pelos contatos, que só permitem transmissão de sinais quando fechados, e energizar as bobinas. O diagrama Ladder segue uma ordem de varredura de cima para baixo e da esquerda para a direita nas barras horizontais e apenas considera

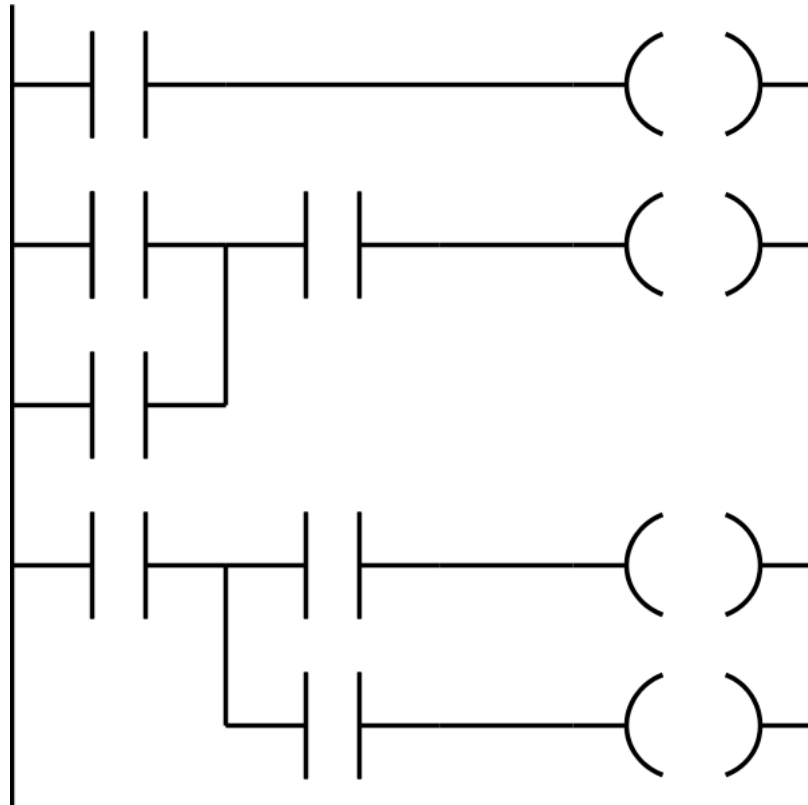


Figura 2.1: Exemplo de estrutura Ladder com contatos e bobinas.

o estado de cada contato no instante de sua varredura. A varredura completa de todas as barras horizontais configura um ciclo de varredura. Com esta sequência, é possível determinar ações baseadas em um ou mais eventos.

Os contatos são acionados por eventos ou pelas próprias bobinas. O acontecimento do evento é informado ao diagrama Ladder por meio de sensores, botões ou interruptores. As bobinas, quando energizadas, podem acionar contatos ou determinar ações, enviando sinais lógicos para dispositivos externos que determinam ligar ou desligar este dispositivo. Além dos contatos e bobinas, existem também outros componentes importantes que serão utilizados no projeto.

## 2.2 Representação dos elementos básicos de um diagrama Ladder

Dentre os contatos e bobinas, existem diversas configurações e representações básicas para cada um, além de diversos componentes bastante utilizados.

### 2.2.1 Contatos

Os contatos são componentes que possuem dois estados (aberto e fechado) e podem estar energizados ou não, o que faz variar o estado do contato. Os contatos só transmitem informação quando encontram-se fechados. Cada contato pode ser representado por um bit de controle que determina se o contato encontra-se energizado (bit de controle igual a 1) ou se o contato não encontra-se energizado (bit de controle igual a 0).

São quatro os principais modelos de contatos: contato normalmente aberto (NA); contato normalmente fechado (NF); contato de borda de subida (tipo P); e contato de borda de descida (tipo N).

O contato normalmente aberto segue a lógica de que, se o bit de controle está com o valor zero, o contato encontra-se aberto, e, se o bit de controle está com o valor igual a um, o contato é fechado e permite a transmissão de informação. A representação do contato normalmente aberto pode ser observada na figura 2.2.



Figura 2.2: Representação do contato normalmente aberto.

O contato normalmente fechado segue a lógica inversa do contato NA. Se o bit de controle está com o valor zero, o contato encontra-se fechado e permite transmissão de informação, mas se o bit de controle está com o valor igual a um, o contato encontra-se aberto. A representação do contato normalmente fechado encontra-se na figura 2.3.



Figura 2.3: Representação do contato normalmente fechado.

O contato de borda de subida leva em conta a alteração do valor do bit de controle de zero para um. O contato tipo P será considerado fechado apenas no ciclo de varredura em que o bit de controle sofrer a alteração de valor de zero para um. Nos demais ciclos, o contato permanece aberto. O contato de borda de subida está representado na figura 2.4.

O contato de borda de descida considera a alteração de valor do bit de controle de um para zero, diferente do contato de borda de subida. Assim, o contato tipo N será considerado fechado apenas no ciclo de varredura em que o bit de controle sofrer a alteração de valor de um para zero, permanecendo aberto nos demais ciclos. Sua representação encontra-se na figura 2.5.



Figura 2.4: Representação do contato de borda de subida.



Figura 2.5: Representação do contato de borda de descida.

Após descrever todos os tipos de contato, serão descritos os tipos de bobina a serem utilizados.

## 2.2.2 Bobinas

As bobinas, iguais aos contatos, também possuem dois estados (energizada e não energizada). Quando energizada, a bobina determina um comando que será dado a um bit de referência dependendo do tipo de bobina. As ações dos atuadores são comandadas pelo bit de referência, e não pela bobina em si. O bit de referência também pode alterar o estado de um contato, como dito anteriormente. Para tanto, basta que o bit de referência seja o mesmo bit que controla o contato.

São três os principais modelos de bobina: bobina simples; bobina SET; e bobina RESET.

A bobina simples altera o valor do bit de referência de acordo com seu estado. Quando a bobina simples encontra-se não energizada, o bit de referência é zero, e quando encontra-se energizada, o bit de referência é um. A bobina simples está representada na figura 2.6.



Figura 2.6: Representação da bobina simples.

A bobina SET determina que o bit de referência assumo o valor um, sempre que a bobina for energizada, e permaneça igual a um mesmo que a bobina seja desenergizada. Esta determinação serve como uma mudança inercial, ou seja, o valor do bit de referência não retornará a zero enquanto outra bobina não determinar que seu valor seja alterado. A representação da bobina SET encontra-se na figura 2.7.

A bobina RESET também determina o valor do bit de referência de forma inercial, mas, diferente da bobina SET, o bit de referência assume o valor zero quando a bobina é energizada. A bobina RESET está representada na figura 2.8.

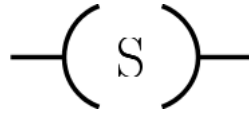


Figura 2.7: Representação da bobina SET.



Figura 2.8: Representação da bobina RESET.

Para auxiliar na definição da lógica do diagrama Ladder, foram criados outros componentes. Estes componentes seguem lógicas bem peculiares e podem ter a mesma funcionalidade de um contato, ou seja, condicionar a transmissão de informação para as bobinas; podem ter a mesma funcionalidade das bobinas, ou seja, determinar ações; e podem ter ambas as funcionalidades, determinando uma ação dependendo de uma condição específica.

Existem muitos componentes que podem ser utilizados, mas apenas três tipos serão apresentados neste projeto: temporizador TimerOn Delay (TON); Contador de alta velocidade (High Speed Counter – HSC); e blocos comparadores.

### 2.2.3 Temporizadores

Os temporizadores são utilizados para controlar ou determinar uma condição dependendo do tempo. Os temporizadores mais utilizados são o temporizador de pulso (TP); o temporizador TimerOn Delay (TON); e o temporizador TimerOff Delay (TOF).

Todos os temporizadores possuem duas entradas (IN e PT) e duas saídas (Q e ET). A entrada IN é utilizada como sinal de referência para a contagem do tempo; a entrada PT determina o tempo de referência para o bloco; a saída Q emite o sinal lógico de saída do bloco; e a saída ET permite armazenar em uma variável o tempo transcorrido na contagem, seguindo uma lógica para cada tipo de contador.

O temporizador de pulso, figura 2.9, interpreta a borda de subida do sinal de referência como o início de uma contagem. Após o início da contagem, o bloco emite um sinal lógico de saída com a mesma duração do tempo de referência. O tempo transcorrido é armazenado enquanto há um sinal de referência na entrada do bloco ou enquanto o bloco estiver emitindo um sinal lógico de saída. Os gráficos referentes aos sinais de entrada e saída podem ser observados na figura 2.10.

O bloco temporizador TON, figura 2.11, serve para atrasar a transmissão da informação, entre o sinal de referência e o sinal lógico de saída, por um tempo definido pelo tempo de referência. O tempo transcorrido acompanha o sinal de

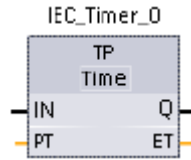


Figura 2.9: Representação do bloco temporizador de pulso. [3]

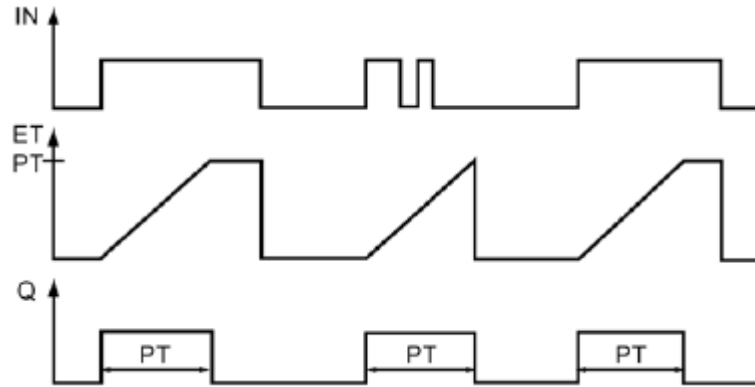


Figura 2.10: Sinais de entrada e saída do bloco temporizador de pulso. [3]

referência e é limitado pelo tempo de referência. Os gráficos dos sinais de entrada e saída do bloco se encontram na figura 2.12.

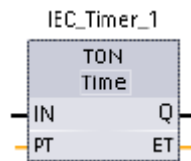


Figura 2.11: Representação do bloco temporizador TimerOn Delay. [3]

Por fim, o temporizador TOF, figura 2.13, emite um sinal lógico de saída que segue o sinal de referência e se prolonga pelo tempo de referência quando cessa o sinal de referência da entrada. A contagem do tempo transcorrido inicia junto com a contagem do tempo de referência, e se mantém armazenado até a próxima borda de subida do sinal de referência. Os gráficos referentes aos sinais de entrada e saída podem ser observados na figura 2.14.

## 2.2.4 Contadores de alta velocidade

O contador de alta velocidade (High Speed Counter – HSC), figura 2.15, é utilizado para contar pulsos que possuem frequências maiores que a frequência de um ciclo de varredura, enquanto os contadores comuns contam apenas um pulso por ciclo de varredura.

Para que o HSC funcione corretamente, é necessário que a entrada EN receba um sinal positivo constantemente. O HSC é acionado para a contagem quando

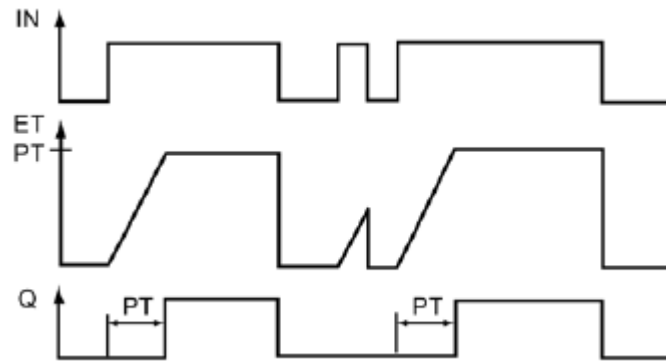


Figura 2.12: Sinais de entrada e saída do bloco temporizador TimerOn Delay. [3]

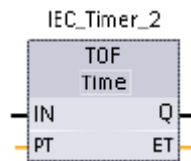


Figura 2.13: Representação do bloco temporizador TimerOff Delay. [3]

a entrada SW\_GATE recebe o valor um, podendo ser definida uma lógica com contatos e outros componentes para iniciar a contagem. Em SET\_DO, é feita a contagem de pulsos, que deve ser vinculada ao “encoder A”. O valor da contagem será expresso na variável COUNTVAL, acessível a outros componentes como valores de referência.

## 2.2.5 Instruções de comparação

Os blocos comparadores funcionam de forma similar aos contatos, a diferença é que este bloco não é controlado por um bit de controle, mas sim um número de controle e um número de referência. O funcionamento dos blocos comparadores consiste em fechar o contato e permitir a transmissão de informação quando o número de controle, comparado com o número de referência, atende à restrição determinada.

Existem seis tipos de restrição que atendem à lógica de que o número de controle deve ser comparado ao número de referência: controle maior que a referência ( $>$ ); controle maior ou igual à referência ( $>=$ ); controle igual à referência ( $=$ ); controle menor ou igual à referência ( $<=$ ); controle menor que a referência ( $<$ ); e controle diferente da referência ( $<>$ ).

Outra consideração a ser feita é que os tipos de variáveis do número de controle e número de referência devem ser iguais. Uma melhor descrição quanto aos tipos de variáveis será realizada no capítulo 5.

A figura 2.16 mostra a representação do bloco comparador.

Após descrever todos os componentes a serem utilizados no projeto supervisorio,

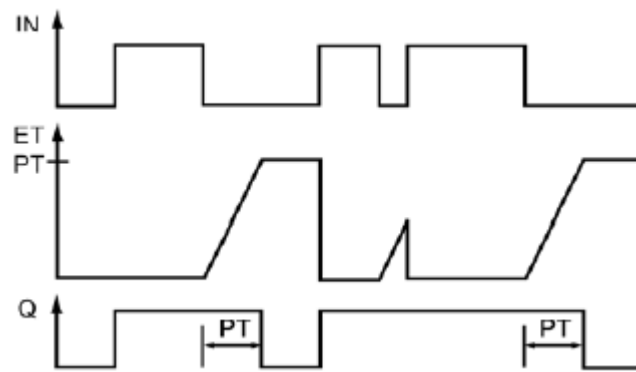


Figura 2.14: Sinais de entrada e saída do bloco temporizador TimerOff Delay. [3]

será descrita como utilizar o diagrama Ladder em dispositivos como CLP e IHM.

## 2.3 Utilização do diagrama Ladder em dispositivos como CLP e IHM

O CLP, como dito anteriormente, é utilizado para comandar um sistema automatizado seguindo a lógica programada e inserida no CLP pelo desenvolvedor. Essa lógica, no caso do CLP SIMATIC S7 da SIEMENS, deve ser feita através de um computador e transferida para o CLP. A SIEMENS possui um software próprio, para programação dessas lógicas, chamado Totally Integrated Automation Portal (TIA Portal), que aceita algumas linguagens para programação, dentre elas o diagrama Ladder.

A lógica de programação, em um CLP SIEMENS, é toda baseada em TAGs. As TAGs são variáveis do sistema que são utilizadas para representar os bits e números, de referência e controle, que devem ser declaradas na lista de TAGs. Cada TAG deve ser alocada em uma posição da memória do CLP. As TAGs serão melhor descritas no capítulo 5.

A IHM não permite programação em Ladder, mas permite fazer referências às TAGs do CLP, utilizando os valores obtidos pelo diagrama Ladder do sistema automatizado. É possível utilizar o diagrama Ladder para desenvolver uma funcionalidade exclusiva da IHM, mas é necessário que este diagrama seja implementado em um CLP.

No próximo capítulo, será descrito o sistema automatizado que será utilizado como base para o projeto supervisor.

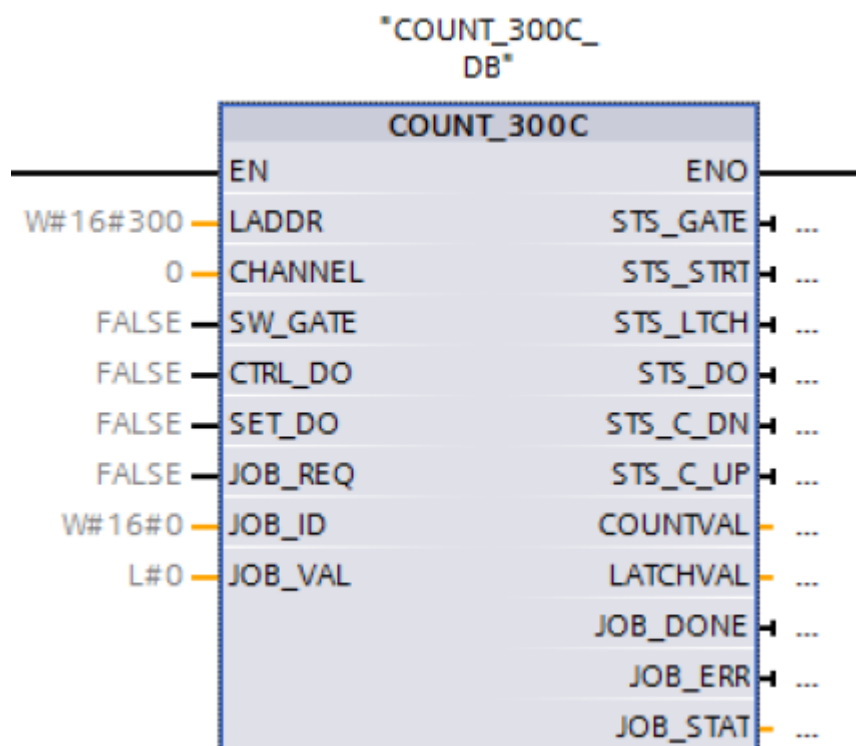


Figura 2.15: Representação do contador de alta velocidade.

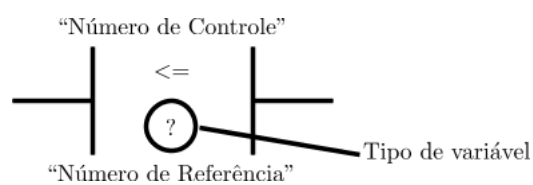


Figura 2.16: Representação geral do bloco comparador.

## Capítulo 3

# Projeto Automatizado da Planta Mecatrônica

Neste capítulo será descrito o sistema de estocagem de peças, apresentado em [4], da planta mecatrônica encontrada no Laboratório de Controle e Automação da Universidade Federal do Rio de Janeiro. O projeto foi implementado utilizando o software da Siemens, TIA Portal, e programado em linguagem Ladder. Esse projeto será utilizado como base para o tutorial, a ser desenvolvido utilizando a IHM, que ensinará, de forma prática e genérica, como criar um sistema supervisor.

Primeiramente serão descritos a planta mecatrônica e seus componentes.

### 3.1 Descrição da planta mecatrônica e de seus componentes

A planta mecatrônica utilizada é a montadora de cubos, fabricada por Christiani-Technical Institute for Vocation Training. Uma imagem representativa da planta mecatrônica é encontrada figura 3.1.

Como pode ser observado na figura 3.2, a planta possui três módulos que interagem entre si e representam funções que simulam um processo industrial, como a escolha da matéria prima, o processamento dessa mesma matéria e a estocagem. Na planta mecatrônica, a matéria prima é representada por cubos de metal ou de plástico, e com variação entre as cores preto e branco para os cubos de plástico. Cada módulo terá suas funções e seus componentes explicados separadamente a seguir.

O módulo 1, figura 3.3, é responsável pela escolha dos cubos. Esse módulo é constituído de um regulador de pressão de ar (1), para acionar os pistões presentes em toda a planta mecatrônica; duas torres de armazenamento de peças (2), cada uma com um pistão para empurrar as peças para a esteira; uma esteira (3), com um sensor capacitivo de fim de curso; e quatro sensores de reconhecimento de peça com

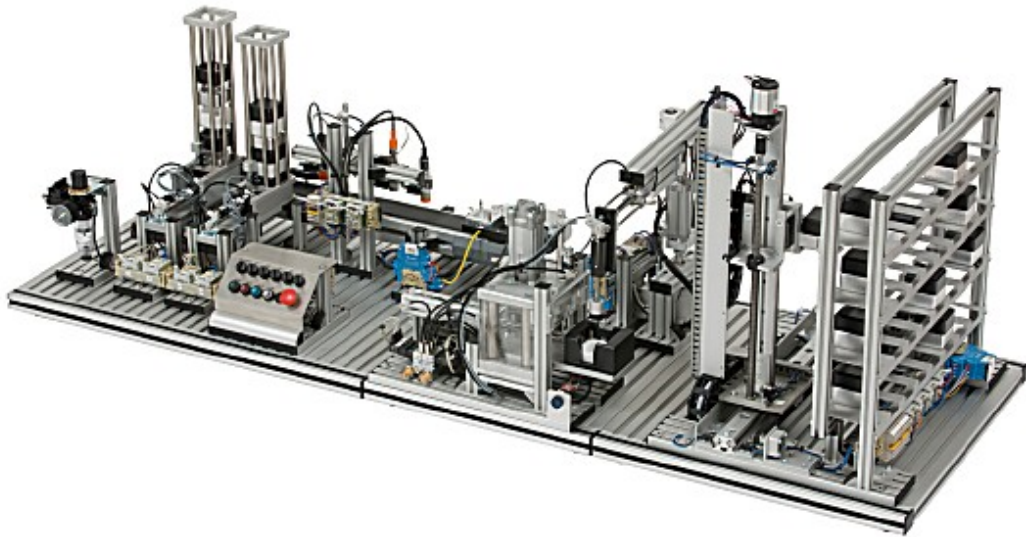


Figura 3.1: Planta mecatrônica Cube Assembly. [5]

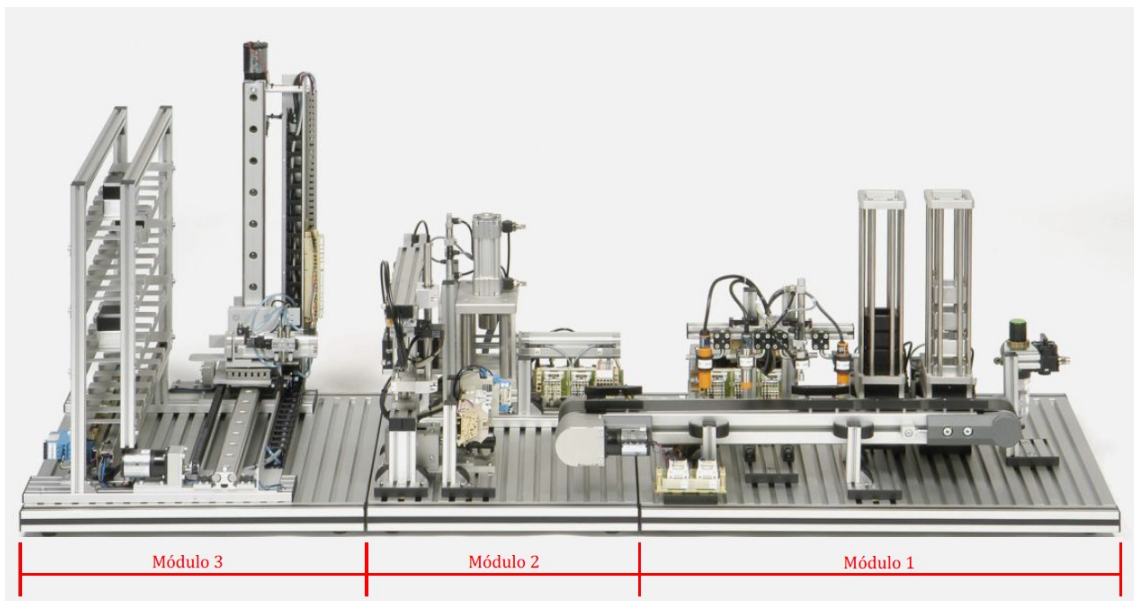


Figura 3.2: Módulos da planta mecatrônica Cube Assembly.

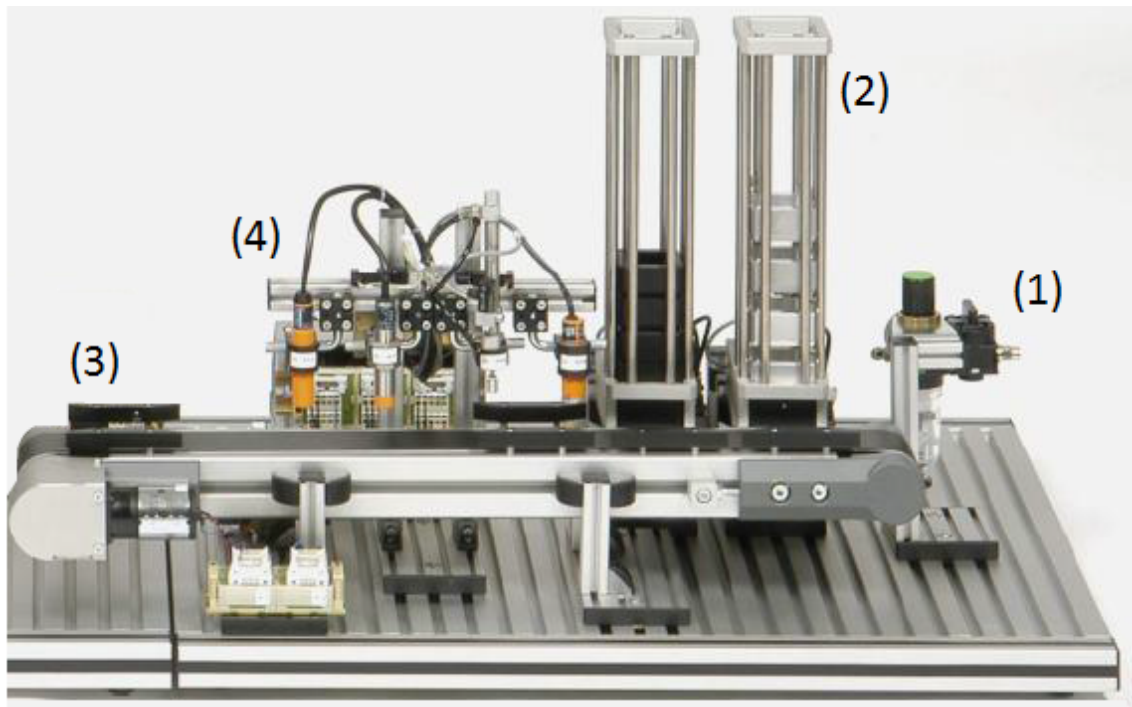


Figura 3.3: Módulo 1 - Reconhecimento de peça. [5]

funções variadas (4).

Cada sensor de reconhecimento de peça possui sua função. O sensor capacitivo é utilizado para identificar a passagem de alguma peça, dando início à seleção de peças; o sensor de posição identifica se a peça encontra-se voltada para cima ou para baixo; o sensor óptico é capaz de identificar a capacidade de reflexão da peça, ou seja, se a peça for branca ou metálica, a capacidade de reflexão é alta e o sensor óptico reconhece, e se a peça é da cor preta, o sensor óptico não é capaz de perceber a presença da peça; e o sensor indutivo, que identifica o material da peça, se a peça é plástica ou metálica. Os sensores encontram-se discriminados na figura 3.4 e apenas o sensor capacitivo será utilizado no sistema supervisor.

O módulo 2, figura 3.5, proporciona a interação entre os módulos 1 e 3, com o auxílio do braço giratório (1), e processa os cubos, unindo dois cubos com o auxílio da prensa (2). O braço giratório possui pistões de acionamento pneumático, para levantar e estender o braço, um sistema de sucção a vácuo para carregar as peças e um contador de giro de alta velocidade. A prensa também utiliza pistões para acionar seus comandos de recolher as peças para o interior da prensa e unir as peças. A prensa não será utilizada o projeto de estocagem de peças.

Por fim, o módulo 3, figura 3.6, é utilizado para estocar as peças na prateleira de alumínio, com o auxílio do elevador. O elevador se locomove na horizontal, através de um trilho e tracionado por uma polia ligada a um motor elétrico, e na vertical, através de uma rosca sem-fim tracionada por outro motor elétrico. Para controlar

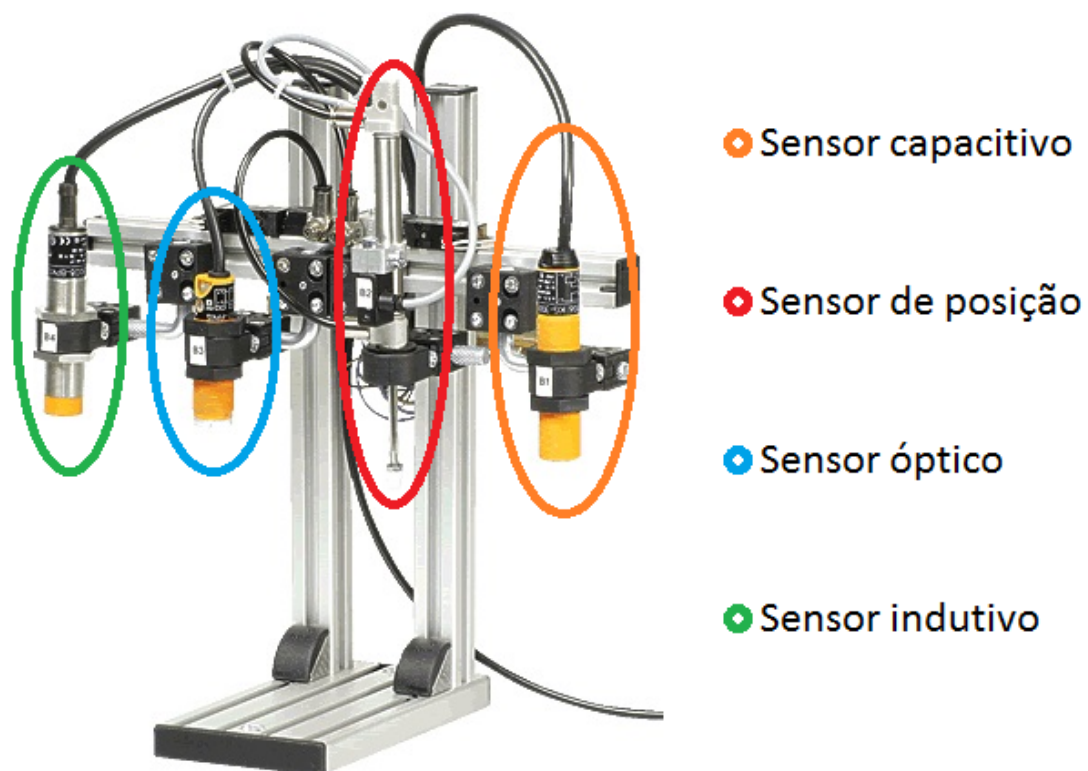


Figura 3.4: Sensores de reconhecimento de peça. [5]

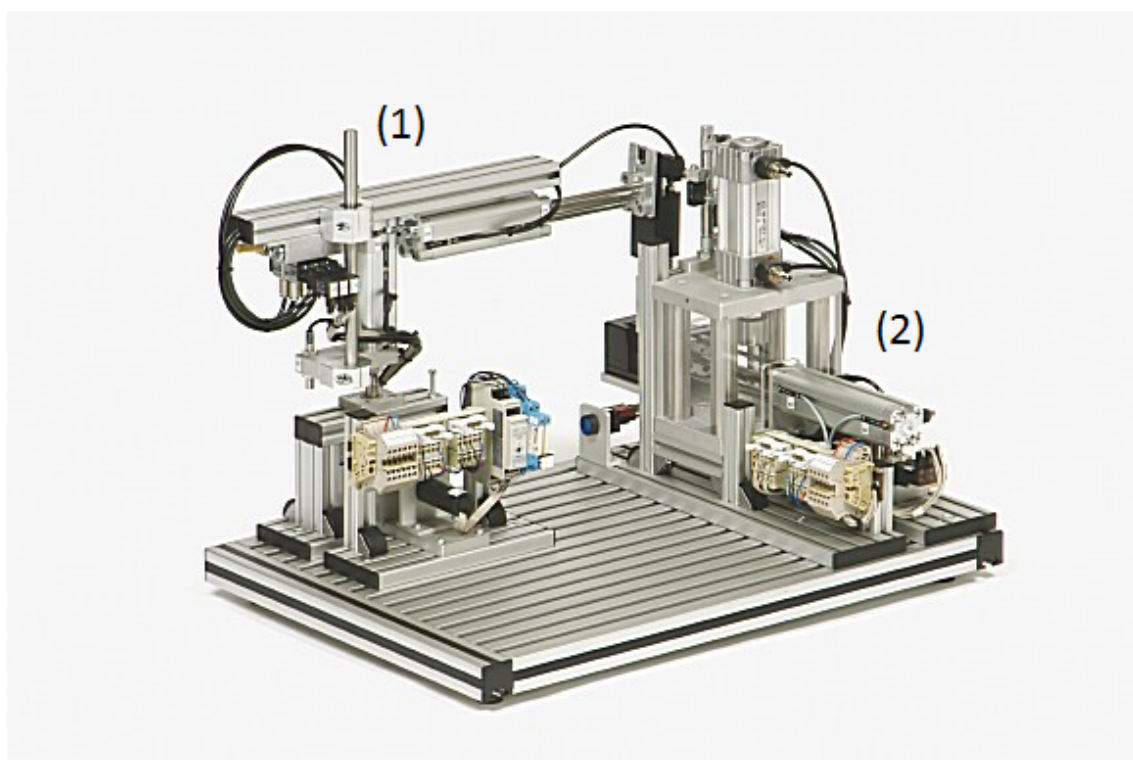


Figura 3.5: Módulo 2 - Processamento e transporte de peça. [5]

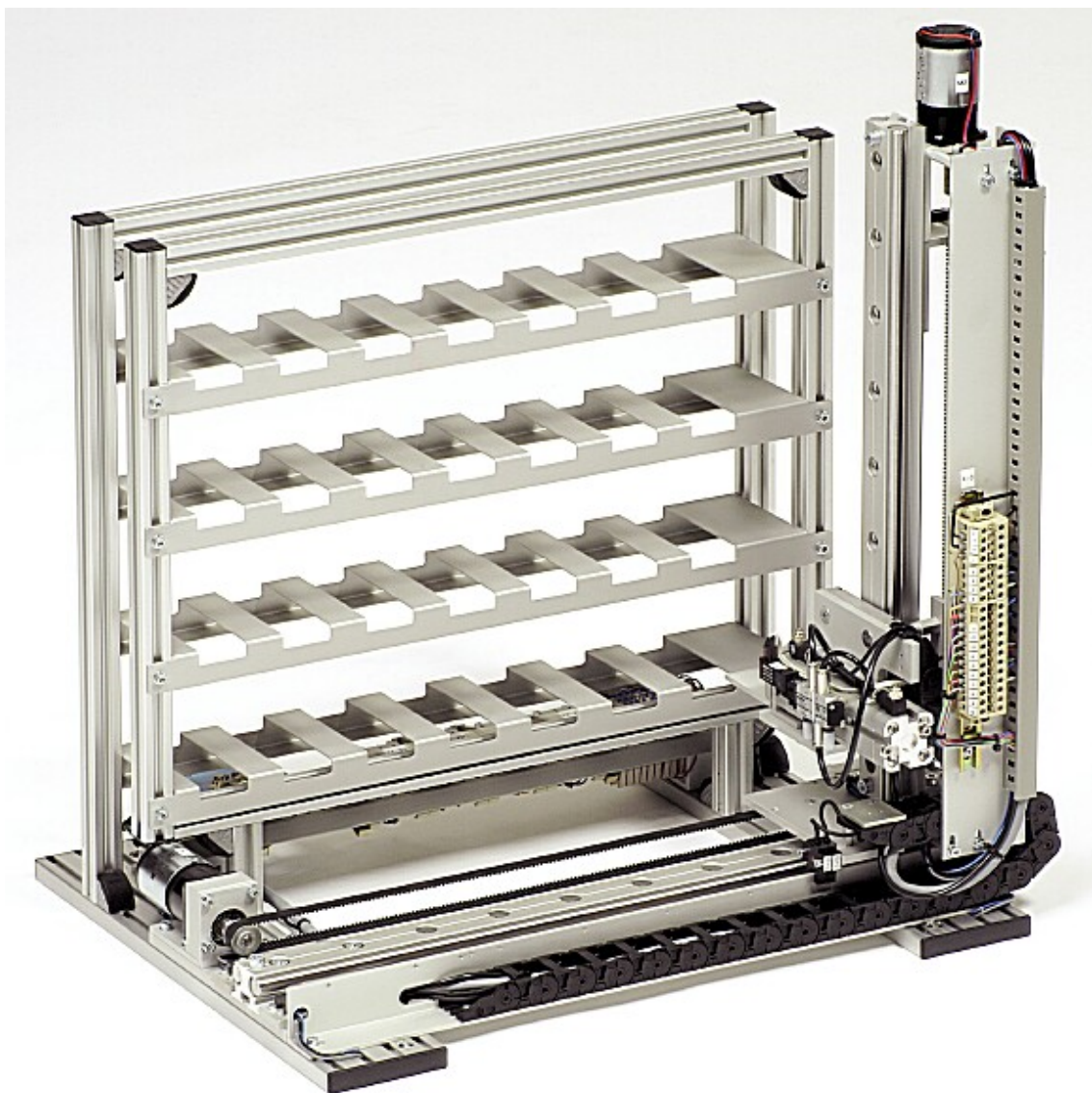


Figura 3.6: Módulo 3 - Estocagem de peça. [5]

seus movimentos, são utilizadas chaves de fim de curso, para que o elevador não ultrapasse os limites do trilho e da rosca sem-fim, e sensores ópticos que reconhecem pequenos orifícios referentes às vinte e oito posições da prateleira. O elevador conta também com um pistão para posicionar e retirar as peças no estoque.

Todos os módulos são controlados, de forma centralizada, pelo CLP S7-300 da SIEMENS e compartilham o mesmo sistema de proteção para seus dispositivos. O CLP é auxiliado por mais duas expansões para aumentar o número de entradas digitais, de 16 para 48, e o número de saídas digitais, de 16 para 32. Três disjuntores são responsáveis pela proteção do CLP e de suas expansões e um relé atua na proteção dos motores e dos demais componentes elétricos encontrados na planta. Os equipamentos de proteção encontram-se abaixo da planta e sua representação pode ser observada na figura 3.7.

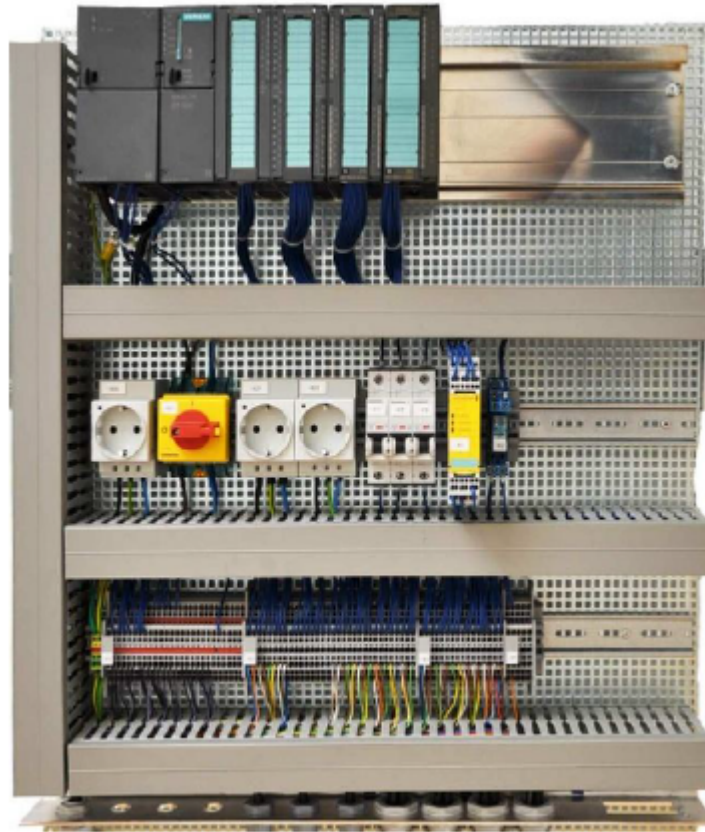


Figura 3.7: Dispositivos de controle e proteção da planta mecatrônica. [5]

Para acionar e comandar a planta, é utilizado um painel de controle, figura 3.8, com quatro botoeiras (Start, Stop, Quit e S6), cinco chaves de duas posições (S1 – S5), uma chave de três posições (Hand/Auto) e um botão de emergência.

Após conhecer todos os componentes da planta mecatrônica, será descrito como é o funcionamento do sistema automatizado.

## 3.2 Descrição do sistema automatizado

O sistema automatizado, desenvolvido para a planta mecatrônica, consiste no transporte e no controle de estoque de peças na prateleira de alumínio. As peças são organizadas na prateleira de forma ordenada da direita para a esquerda e de baixo para cima seguindo a numeração presente na figura 3.9. A peça a ser retirada pode ser escolhida utilizando-se o painel da planta mecatrônica.

Ao iniciar o sistema automatizado, apenas o botão Start está aceso e apenas ele permite alguma ação. Acionando-o, o sistema é calibrado ligando a esteira para a esquerda para dispensar possíveis peças nela contida; retraindo o pistão da torre de peças; calibrando o braço robótico; e retornando o elevador para a posição inicial. A calibração do braço robótico consiste em girar o braço no sentido horário, até acionar o sensor indutivo presente no braço, e retornar o giro do braço, no sentido



Figura 3.8: Painel de acionamento da planta mecatrônica. [5]



Figura 3.9: Numeração das posições da prateleira.

anti-horário, por um número determinado de pulsos, contado pelo contador de alta velocidade, até parar na direção da esteira.

Após a calibragem do sistema, o botão Start fica novamente habilitado e, ao acioná-lo, o transporte de peças se inicia. O transporte de peças não para enquanto tiver pelo menos duas peças na torre de peças, enquanto a prateleira não estiver cheia ou enquanto o botão Stop não for acionado. O botão Stop fica habilitado apenas quando o transporte de peças está ativo. Ao pressioná-lo, o sistema só para quando concluir o transporte de todas as peças já lançadas ao sistema (peças que estejam na esteira, no braço ou no elevador).

O transporte de peças consiste no pistão esticar e liberar a peça para a esteira; acionar a esteira para a direita até que a peça chegue ao sensor de presença final; descer o braço e acionar a sucção para pegar a peça e subir novamente; girar o braço no sentido anti-horário em  $180^\circ$  e parar na posição adequada para aguardar o elevador posicionar-se; posicionar o elevador abaixo do braço para receber a peça; baixar o braço e desativar a sucção, para soltar a peça no elevador, e levantar o braço novamente; retornar o elevador para a posição inicial (mover para a direita); rodar o braço no sentido horário até acionar o sensor indutivo presente no braço e retornar o giro do braço, no sentido anti-horário, até parar na direção da esteira novamente; e posicionar a peça no primeiro espaço vazio da prateleira utilizando o elevador.

Durante o transporte de peça, apenas o botão Stop fica habilitado. Quando acionado e após o final de todos os transportes de peças já iniciados, os botões Start e Quit são habilitados. O botão Start iniciará outra cadeia de transporte de peças e o botão Quit iniciará a retirada de peça.

A retirada de peça depende da escolha do usuário. Para que a retirada seja devidamente iniciada, é necessário, quando o sistema encontrar-se em repouso (após o acionamento do botão Stop), acionar o botão Quit, escolher a peça e acionar o botão Quit novamente para confirmar a escolha de peça. Caso o usuário escolha uma posição da prateleira que não possui peça, o botão Quit pisca três vezes e o sistema retorna para o estado de repouso. A escolha da peça a ser retirada é feita através das chaves de duas posições encontradas no painel utilizando a conversão do número da posição da prateleira de decimal para binário, sendo S1 o bit mais significativo e S5 o bit menos significativo.

Após a escolha correta de uma peça, o processo de retirada se inicia. Esse processo consiste na retirada da peça escolhida, pelo elevador, e o retorno do elevador para a posição inicial, movendo-o para a direita; no giro do braço em  $180^\circ$  no sentido anti-horário para aguardar o elevador; no posicionamento do elevador abaixo do braço; na descida do elevador e início da sucção para a coleta da peça e subida do braço em seguida; no retorno do elevador para a posição inicial; na rotação do

braço no sentido horário até o acionamento do sensor indutivo presente no braço e o retorno do giro do braço, no sentido anti-horário, até parar na direção da esteira novamente; em descer do braço e desativar a sucção para colocar a peça na esteira; e no ligamento da esteira para a esquerda para a conclusão da retirada da peça. Ao final da retirada, o braço calibra novamente para correção de possíveis erros de contagem do contador de alta velocidade.

Ambos os processos, de transporte e retirada de peça, podem ser repetidos enquanto a prateleira não estiver totalmente cheia ou totalmente vazia, dependendo da ação, e enquanto tiver pelo menos duas peças na torre, para o caso do transporte de peça.

Conhecendo o comportamento desejado do sistema automatizado a ser supervisionado e analisando a lógica Ladder presente no apêndice A, é possível dar início ao projeto supervisorio a ser implantado na IHM. A lógica Ladder do sistema automatizado foi obtida a partir da técnica presente no artigo [6] e, a partir dela, é possível obter a rede de Petri do projeto, caso necessário. Deve-se, também, levar em consideração que o projeto do sistema automatizado não deve ser alterado, quando da implementação do sistema supervisorio, mas receber adendos para não gerar erros previamente inexistentes.

## Capítulo 4

# Instalação dos Dispositivos de Controle e Supervisão

Neste capítulo, antes de iniciar o projeto supervisorio, será ensinado, com base na explicação presente no projeto [7], como conectar o painel IHM (SIMATIC HMI Basic Panel) e o controlador lógico programável (PLC SIMATIC S7), com o auxílio de um Switch, e como configurar o número de IP de cada equipamento. Os equipamentos utilizados podem ser visualizados na figura 4.1.

Será apresentado também o software TIA Portal da SIEMENS e como criar um projeto novo utilizando um CPL e painel IHM. Após criar o projeto, será necessário configurar o número de IP para que não haja conflito na comunicação dos dispositivos.

### 4.1 Conexão da IHM, do CLP e do computador

Todos esses equipamentos são do fabricante Siemens e necessitam de apenas um programa para suas implementações, o Totally Integrated Automation Portal. Esse programa deverá estar instalado em um computador e todos os equipamentos deverão ser conectados através de um Switch, como mostra a figura 4.2. O switch é utilizado para permitir a comunicação entre três ou mais equipamentos.

Para suas conexões, utiliza-se cabeamento ethernet industrial, rede PROFINET. Basta conectar os três equipamentos (IHM, CLP e computador) ao switch. Desta forma é possível configurar a IHM e o CLP pelo computador e também permitir a interação entre a IHM e o CLP durante o funcionamento do sistema.

Após conectar todos os equipamentos, é necessário saber como criar um projeto no TIA Portal.

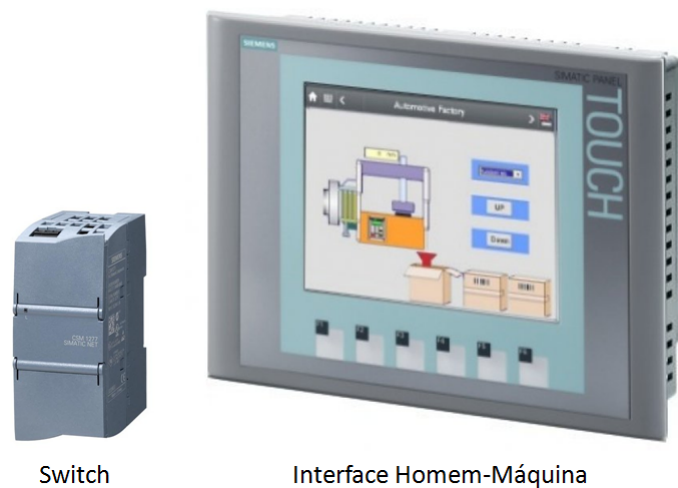


Figura 4.1: Equipamentos SIEMENS do Laboratório de Controle e Automação da UFRJ.



Figura 4.2: Esquema de conexões entre os dispositivos e o computador.

## 4.2 Criação de um projeto

O TIA Portal possui duas formas principais de visualização do projeto, o “Portal view” e o “Project view”. Para situar melhor, as figuras 4.3 e 4.4 mostram os elementos de cada forma de visualização.

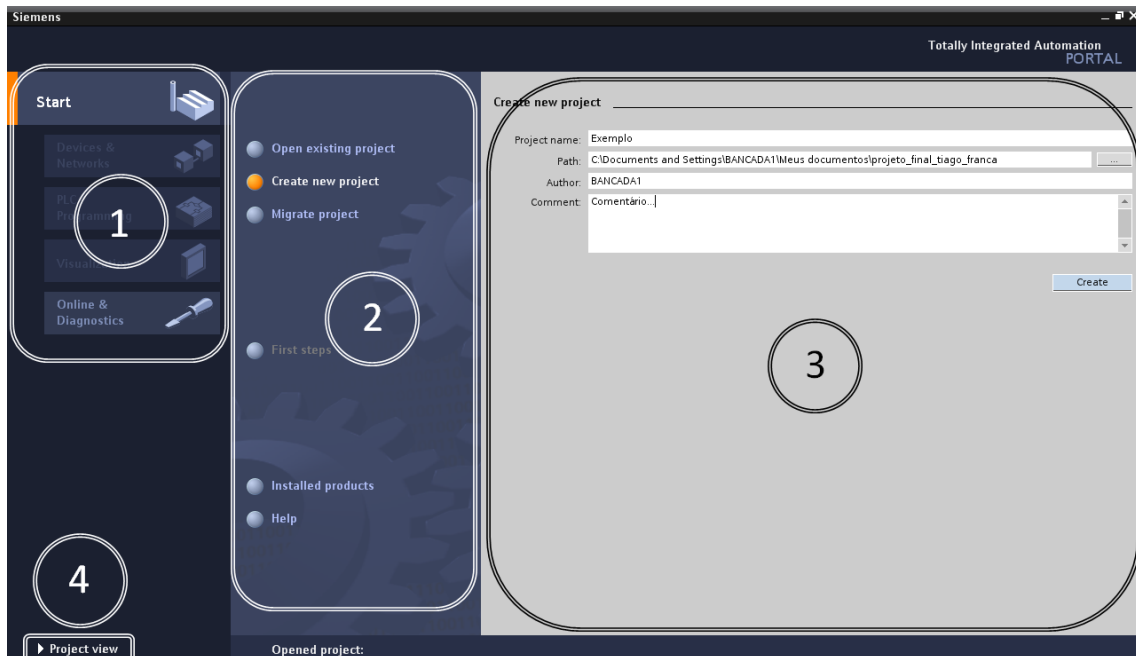


Figura 4.3: Elementos do Portal view.

Elementos do “Portal view”

1. Portal para diferentes tarefas
2. Tarefas para o portal selecionado
3. Painel de seleção para a ação selecionada
4. Ir para o “Project view”

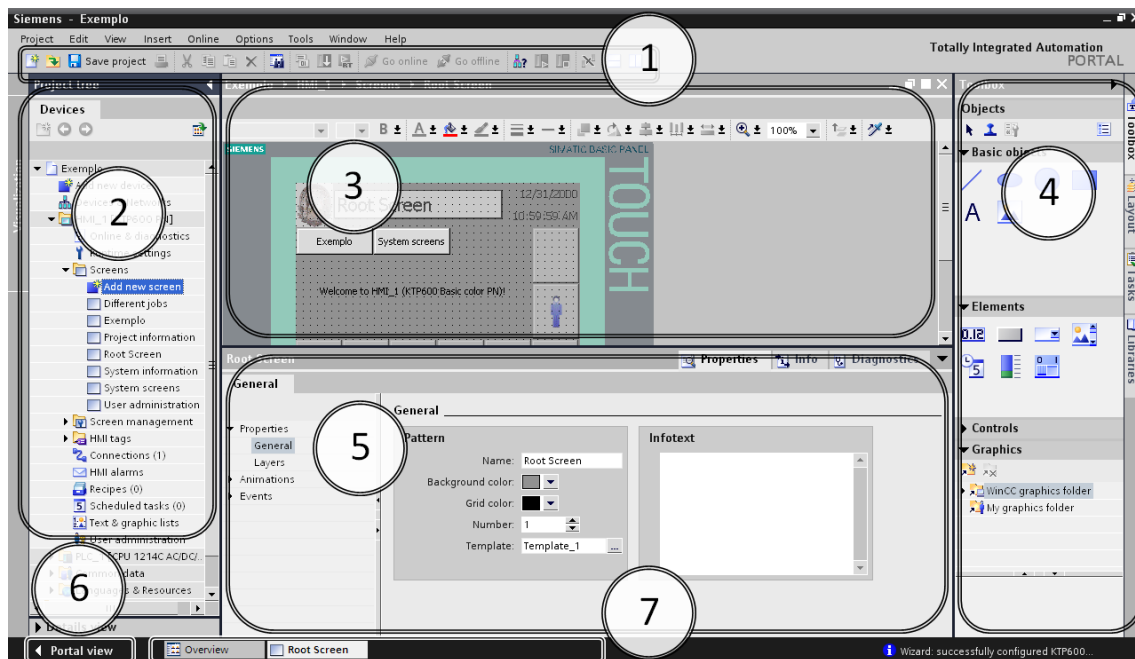


Figura 4.4: Elementos do Project view.

#### Elementos do “Project view”

1. Menus e barra de ferramentas
2. Explorador do projeto
3. Área de trabalho
4. Lista de tarefas
5. Janela de inspeção
6. Ir para o “Portal view”
7. Abas do editor

Para criar um projeto novo no programa TIA Portal, na aba lateral “Start”, clique em “Create new Project”. Selecione o nome do projeto e o diretório em que será salvo e prossiga apertando em “Create”, como mostra a figura 4.5.

O próximo passo é adicionar os dispositivos. Para isso, clique em “Configure a device” e, em seguida, “Add new device”, como mostrado na figura 4.6.

A forma mais recomendada de se adicionar um dispositivo é a partir da tela “Portal view”, figura 4.3. Para retornar ao “Portal view”, utilize o botão indicado pelo número 6 na figura 4.4, contudo, também é possível adicionar um dispositivo a partir da tela do projeto.

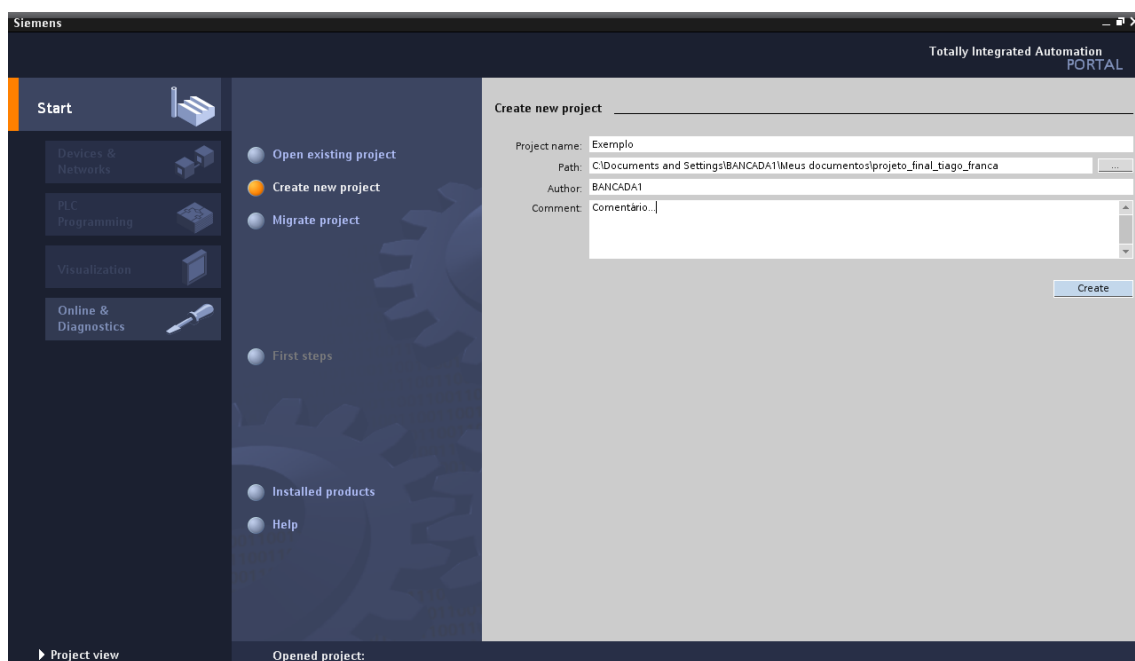


Figura 4.5: Tela de criação de um novo projeto.

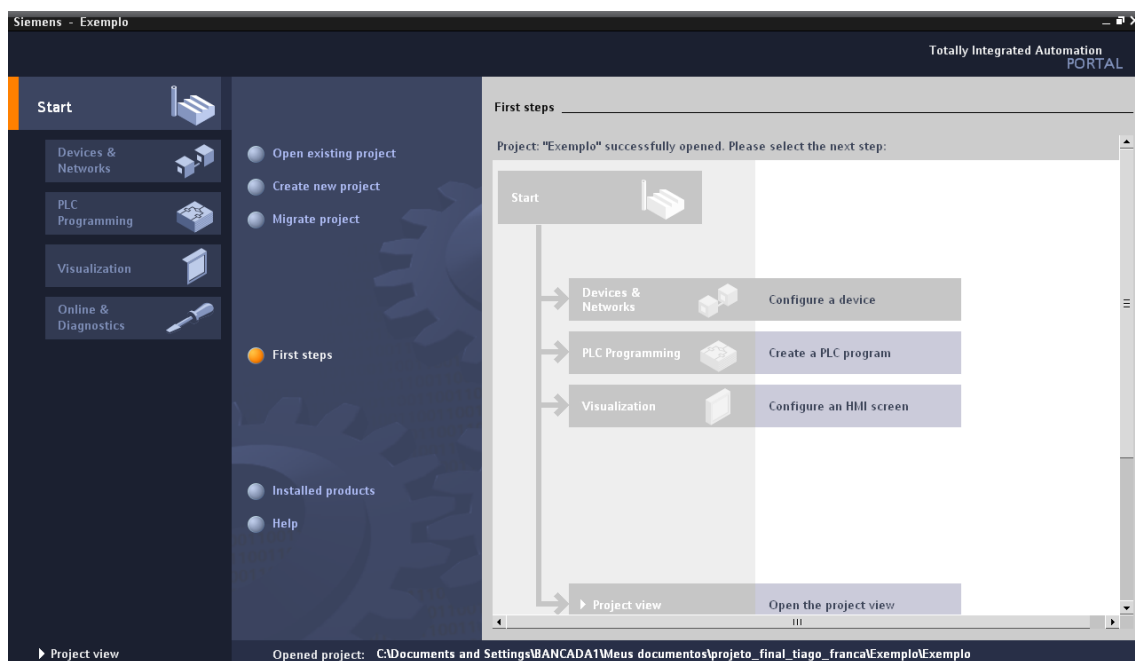


Figura 4.6: Tela “First Step”. Configuração de dispositivo.

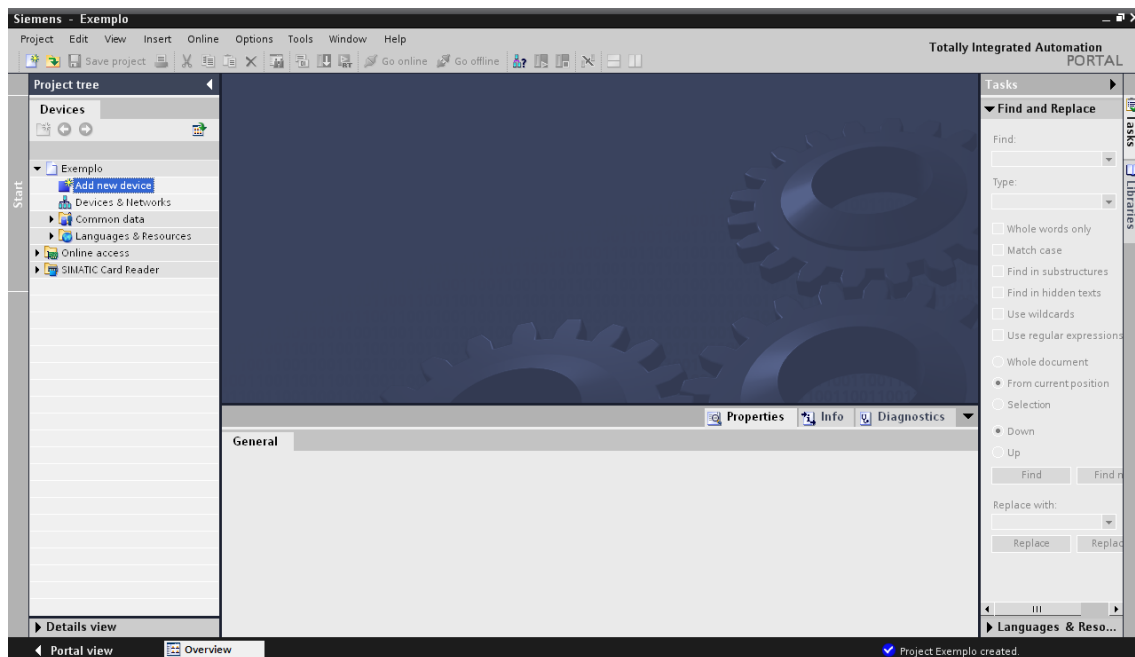


Figura 4.7: Como adicionar um dispositivo a partir do “Project tree”.

Na tela do projeto, na aba “Project tree”, da esquerda, dê clique duplo em “Add new device”, para adicionar um dispositivo, como mostra a figura 4.7. Também é possível adicionar um dispositivo a partir da aba “Overview” dando clique duplo em “Add new device”, figura 4.8. A aba “Overview” pode ser acessada a partir da barra inferior indicada pelo número 7 na figura 4.4.

Após adicionar cada dispositivo, é necessário alterar suas configurações e selecionar o modelo correto.

### 4.3 Selecionando o CLP

Na tela “Add new device”, selecione o CLP que será utilizado. Clique primeiramente em SIMATIC PLC.

Considerando que será utilizado o CLP encontrado no Laboratório de Controle e Automação da Universidade Federal do Rio de Janeiro, que encontra-se conectado à planta mecatrônica, selecione o modelo SIMATIC S7-1200 CPU 1214C AC/DC/Rly (6ES7 214-1BE30-0XB0), de acordo com a figura 4.9. Ao finalizar, aperte “Add”.

Após concluir a seleção do CLP, será necessário adicionar a IHM, selecionar o modelo correto e alterar as configurações internas.

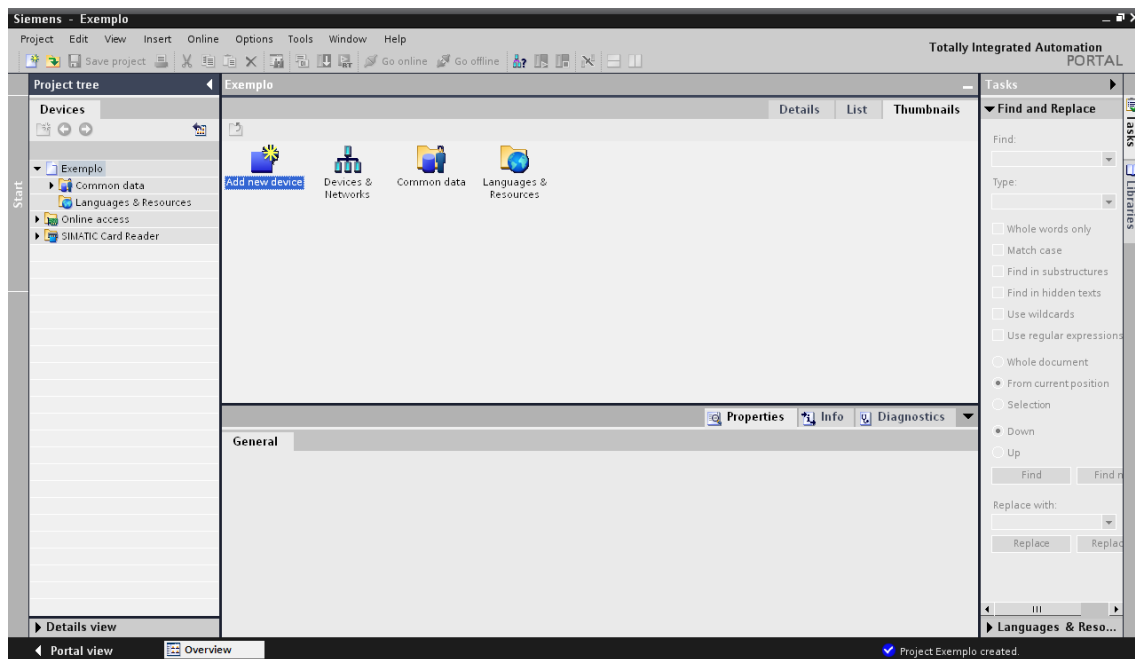


Figura 4.8: Como adicionar um dispositivo a partir do “Overview”.

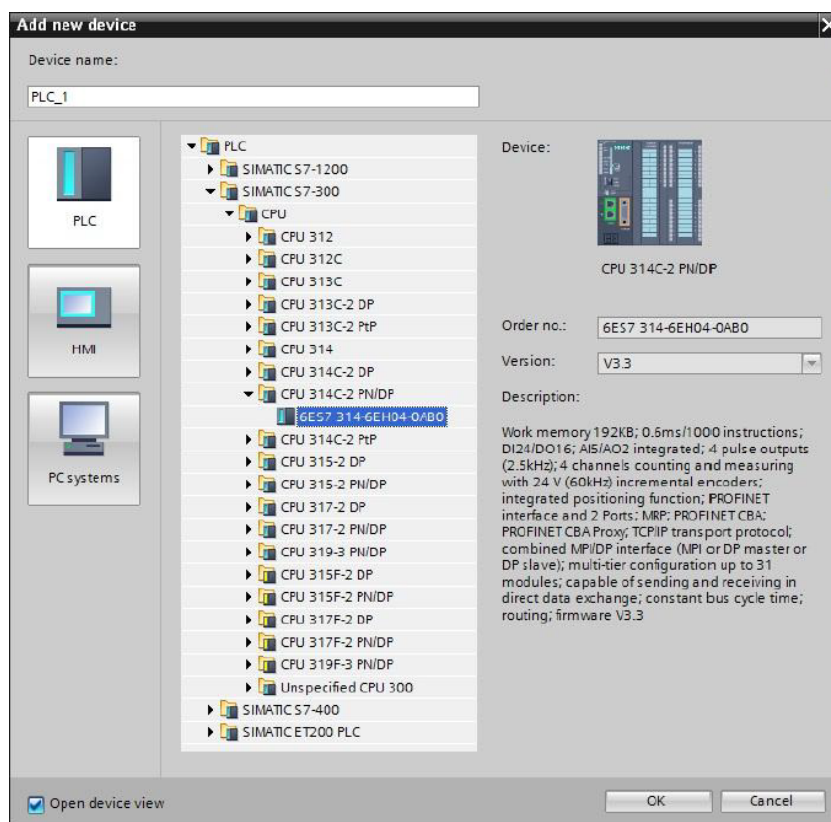


Figura 4.9: Seleção do CLP.

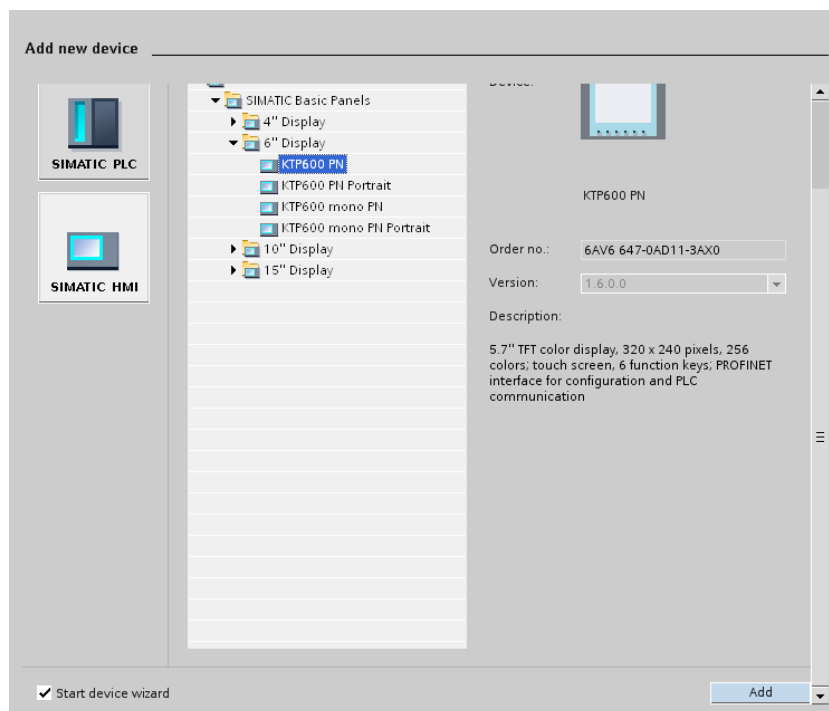


Figura 4.10: Seleção do IHM.

## 4.4 Selecionando a IHM

Para selecionar a IHM, na tela “Add new device”, selecione a IHM que será utilizada. Clique primeiramente em SIMATIC HMI.

Considerando que será utilizada uma IHM encontrada no Laboratório de Controle e Automação da Universidade Federal do Rio de Janeiro, selecione o modelo SIMATIC Basic Panels 6” Display KTP600 PN, de acordo com a figura 4.10. Ao finalizar, aperte “Add”.

A IHM necessita de uma configuração inicial que será abordada no passo 1 do capítulo 5. Primeiramente, o foco será explicar os passos necessários para configurar o número de IP dos dispositivos.

## 4.5 Configuração do número de IP

Para que haja comunicação sem conflito de informação, faz-se necessária a designação de um número de IP para cada equipamento conectado em rede.

Como padrão serão utilizadas as numerações 192.168.0.X. É necessário escolher um número para cada equipamento e substituir o “X”. Os números devem ser diferentes para que não haja conflito de IP.

Outro número que deve ser determinado é a máscara de sub-rede. Como padrão, utilize o número 255.255.255.0.

Para verificar ou alterar o número de IP do computador, vá em “Propriedades de

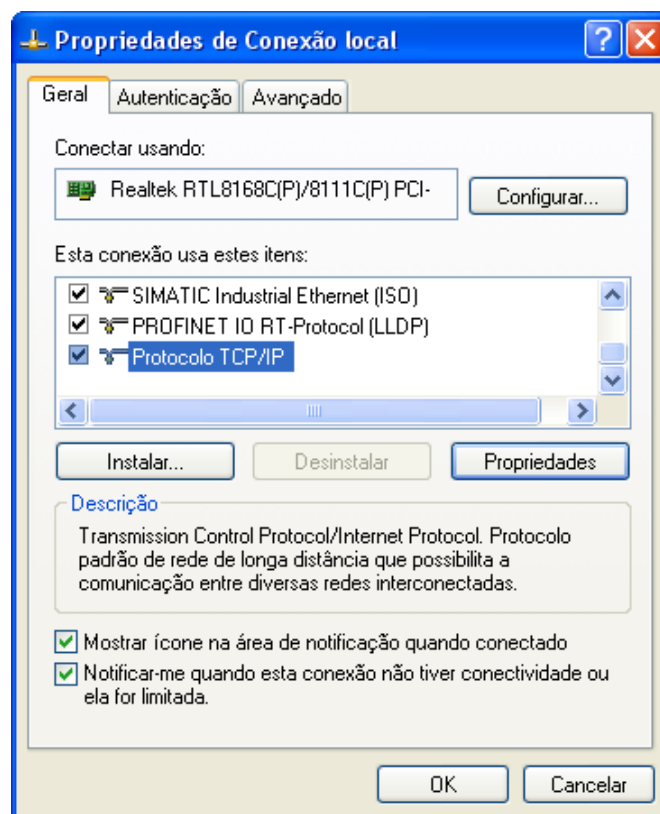


Figura 4.11: Propriedades de conexão local.

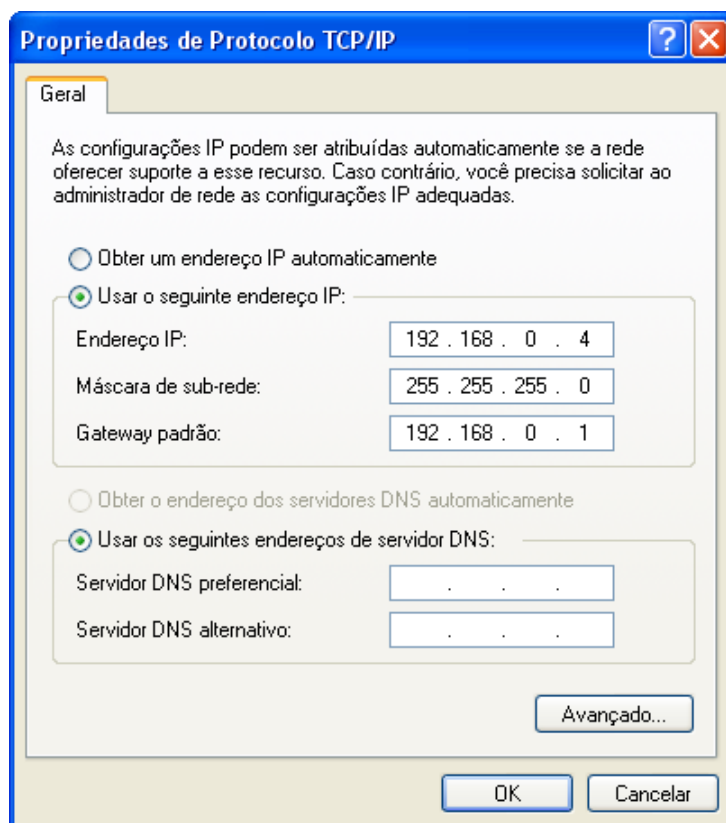


Figura 4.12: Propriedades de protocolo TCP/IP.

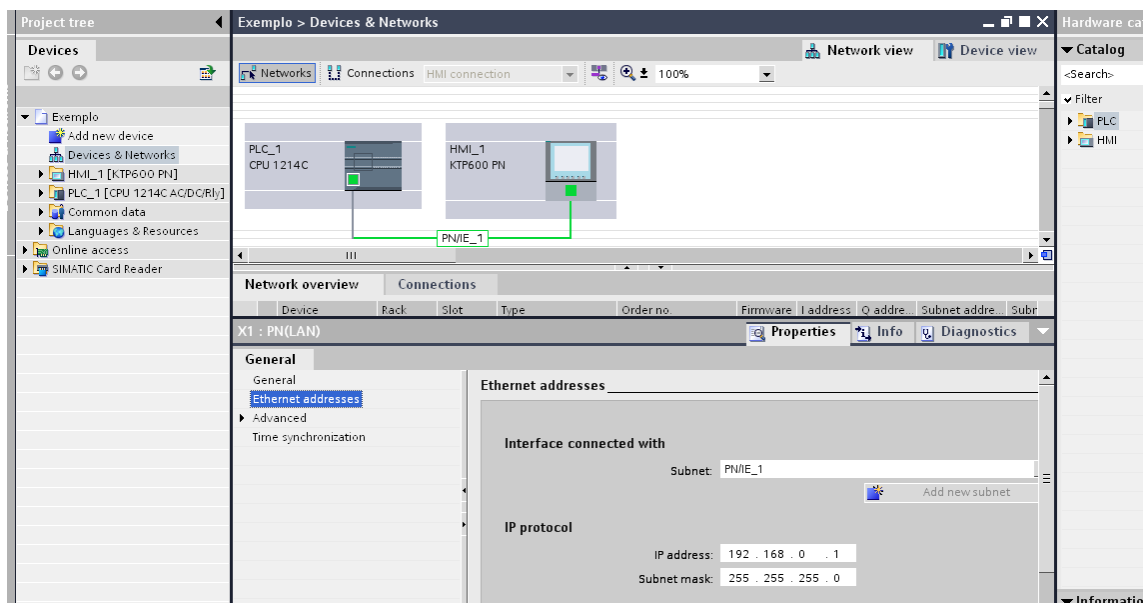


Figura 4.13: Número de IP do CLP.

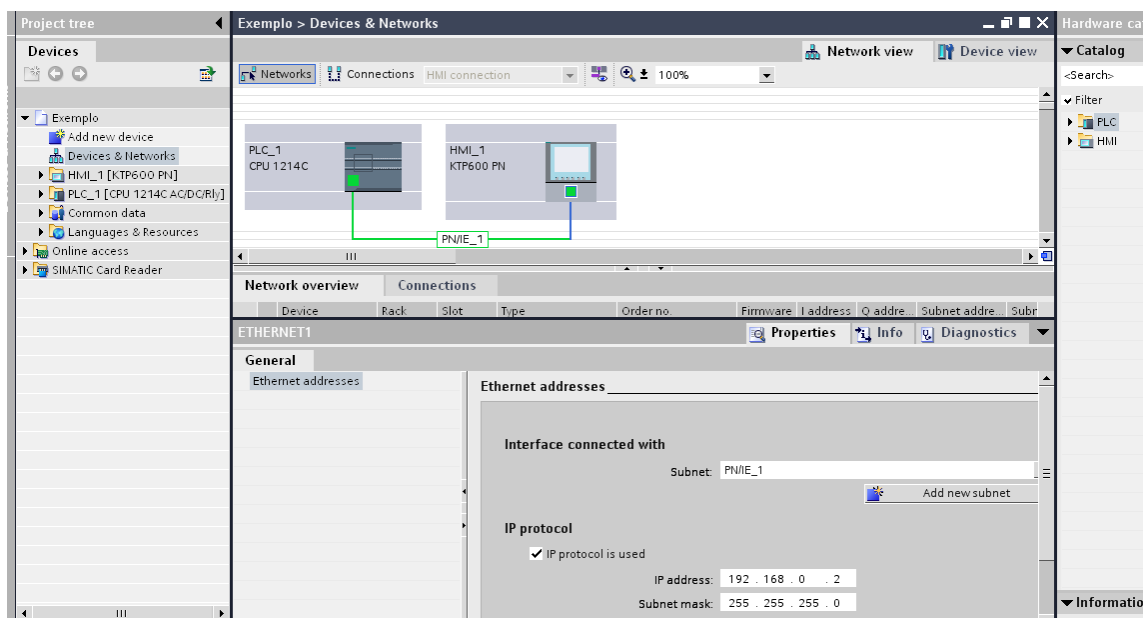


Figura 4.14: Número de IP da IHM.

Conexão local”, selecione a opção “Protocolo TCP/IP” e clique em “Propriedades”, como mostra a figura 4.11. Em “Propriedades”, é possível verificar o número de IP no campo “Endereço IP”, como mostra a figura 4.12.

Para verificar ou alterar o número de IP do CLP ou da IHM, vá em “Devices & Networks” e selecione a entrada do dispositivo que desejar. Vá em “Properties”, “General” e “Ethernet address”, como mostram a figuras 4.13 e 4.14.

Após conectar e configurar todos os dispositivos necessários, criar o projeto e configurar os dispositivos a serem utilizados, será dado início ao projeto supervisor no software TIA Portal da Siemens.

## Capítulo 5

# Projeto do Sistema Supervisório da Planta Mecatrônica

Neste capítulo será apresentado um tutorial sobre a criação de um projeto de um sistema supervisório utilizando uma IHM SIEMENS para supervisionar os eventos de um sistema de automação existente. Será utilizado como base o sistema automatizado descrito no Capítulo 3.

Este tutorial será desenvolvido passo a passo e, a cada passo que utilize uma função nova, antes de demonstrar como esta função foi utilizada no projeto supervisório, a função será descrita de forma genérica.

Primeiramente serão descritos alguns elementos básicos de um projeto supervisório de uma IHM, que consistirá em uma breve explicação sobre TAGs e telas e uma introdução aos demais elementos a serem utilizados.

Ainda antes de começar um projeto, é necessário visualizar e fazer um esboço de como se pretende representar o sistema automatizado e qual será a forma de interação do usuário com o painel IHM.

### 5.1 Esboço do sistema supervisório

Com o intuito de observar os estados de todos os atuadores, sensores e componentes do sistema de estocagem de peças, serão criadas cinco telas na IHM. Das cinco telas, uma será de apresentação, contendo também as funções internas da IHM, uma para visão geral do sistema, onde os módulos serão representados com poucos detalhes, e três telas para representar, de forma mais detalhada, cada módulo da planta mecatrônica.

A tela de apresentação conterá o logo da universidade, um campo com o nome da tela, data e hora, um botão para iniciar o sistema supervisório e cinco botões do sistema, para observação dos alarmes, para retornar à tela inicial, para trocar de

usuário, para acessar a tela do sistema interno da IHM e para desligar o sistema. O botão para iniciar o sistema será customizado e terá legendas para facilitar o entendimento do usuário. Os demais componentes serão criados e configurados no momento da seleção da IHM.

Todas as demais telas devem conter botões funcionais para o acionamento da planta mecatrônica. Portanto, os botões “Start”, “Stop” e “Quit”, além do interruptor de alternância entre o painel IHM e o painel da planta mecatrônica, devem ser adicionados em um template para os botões funcionais apareçam nas telas desejadas. Tais botões devem seguir a lógica de habilitação e desabilitação do sistema de estocagem de peças e devem ficar claros para o usuário quanto ao seu estado, sem que os botões desapareçam da tela.

A tela de visão geral, além dos botões funcionais, conterá simplificações das representações dos estados de cada módulo. Cada simplificação de representação servirá de botão para alternar entre as telas de seus respectivos módulos. A representação simplificada do módulo da esteira informará apenas se a esteira encontra-se ligada; a representação simplificada do módulo do braço informará ao usuário os estados dos pistões contidos no braço, ou seja, se o braço encontra-se levantado ou abaixado, e estendido ou retraído; e a representação do módulo da prateleira informará quais posições da prateleira possuem peças.

A tela referente ao módulo da esteira representará o estado do pistão da torre de peças, ou seja, demonstra o instante em que a peça é colocada na esteira, representará também se a esteira encontra-se ligada e a posição estimada da peça na esteira ou se não há peça na esteira.

A tela do módulo do braço representará, além dos pistões contidos no braço, o sistema de sucção e a posição do braço durante o giro. Haverá uma representação diferenciada para a calibragem do braço e para as demais posições do giro.

Por fim, a tela do módulo da prateleira informará quais posições da prateleira estão ocupadas por peças e conterá botões para seleção de peça para a retirada. O botão “Quit”, constante em todas as telas de representação do sistema, ao ser pressionado, deve redirecionar a IHM para a tela do módulo da prateleira, para que o usuário escolha a peça a ser retirada.

O esboço criado deverá ser seguido durante todo o projeto do sistema supervisorio para garantir a completa observabilidade do sistema de estocagem de peça. Após concluir o esboço, é possível iniciar a criação do sistema supervisorio, precedida de uma explicação dos elementos básicos de um projeto em IHM.

## 5.2 Elementos de um projeto

Os elementos de um projeto podem ser variáveis do sistema, denominadas TAGs, telas de representação de estado e objetos, que podem ser utilizados para melhor representar o projeto físico na IHM. Cada elemento de um projeto possui suas propriedades, animações e eventos.

As propriedades determinam as configurações necessárias de cada elemento utilizado; as animações determinam que movimentos e quando esses movimentos serão executados pelo elemento, dependendo de uma TAG; e os eventos determinam quais ações serão executadas quando ocorre uma alteração de estado do elemento, como, por exemplo, o acionamento de um botão que executa a ação de trocar de tela.

Os objetos podem ser encontrados na lista de tarefas na aba “Toolbox”, e esta aba divide-se em quatro partes: “Basic objects”, “Elements” e “Controls”.

Em “Basic objects”, é possível encontrar linhas, formas geométricas, textos e figuras; já em “Elements”, é possível encontrar botões, elementos de entrada e saída de dados, elementos de data e hora, barras e interruptores; e em “Controls”, é possível encontrar avisos de alarme, gráficos, listas de usuários e listas de receita.

## 5.3 Definição de TAGs

TAGs são variáveis do sistema que podem ser utilizadas como variáveis internas ou variáveis externas, pelo ponto de vista da IHM. As internas serão utilizadas apenas pela IHM e as externas devem ser definidas como TAGs já existentes no CLP.

As internas devem ser utilizadas para auxiliar na definição da apresentação das telas para o operador, ou seja, são TAGs que controlam elementos dentro da própria IHM. Já as externas devem ser utilizadas para receber dados do CLP ou enviar dados para o CLP. Essas TAGs serão utilizadas para demonstrar para o operador o estado em que o sistema se encontra, pela tela da IHM, enviar alertas e alarmes para o operador ou para ativar transições, alterando o estado do sistema.

As TAGs do CLP podem ser TAGs do tipo entrada (“I”), do tipo saída (“Q”) ou do tipo memória (“M”). Todos os tipos de TAG permitem o acesso e o envio de informações à IHM, mas é aconselhável utilizar TAGs do tipo memória para garantir que não ocupará endereços importantes do CLP.

Cada TAG pode ter tamanho diferente e pode ocupar um lugar específico na memória do CLP. Essas estruturas são chamadas de estruturas de endereços mnemônicos e estão descritas na tabela 5.1. Os valores e dados encontrados na tabela foram obtidos em [2] e [7].

Tabela 5.1: Estrutura de endereços mnemônicos de sistemas de automação SIE-MENS

| Mnemônicos | Definição             | Tipo de Variável                  | Áreas de Endereço          |
|------------|-----------------------|-----------------------------------|----------------------------|
| I          | Input bit             | BOOL                              | 0.0..3.7 e<br>124.0..125.7 |
| Q          | Output bit            | BOOL                              | 4.0..5.7 e<br>124.0..125.7 |
| M          | Memory bit            | BOOL                              | 0.0..8191.7                |
| MB         | Memory byte           | BYTE, CHAR, SINT,<br>USINT        | 0..8191                    |
| MW         | Memory word           | WORD, INT, UINT                   | 0..8190                    |
| MD         | Memory<br>double word | DWORD, DINT, UDINT,<br>REAL, TIME | 0..8188                    |

Além do tipo da TAG, é importante saber o tamanho que uma TAG ocupa na memória do CLP. Deve-se ter atenção no tamanho das TAGs durante a sua criação, porque, **se uma TAG sobrepuser à outra, seus valores serão alterados sem o conhecimento do operador.**

Cada TAG deve ser pensada e definida de forma correta, porque, além do problema de sobreposição de TAGs, alguns eventos da IHM e algumas ferramentas do CLP só podem ser executados por alguns tamanhos de TAG. Os tamanhos encontram-se definidos na tabela 5.2.

Mesmo sabendo o que é uma TAG, é necessário saber as especificidades da criação de uma TAG do CLP e uma TAG da IHM.

### 5.3.1 TAG do CLP

Para criar uma TAG no CLP, é necessário, na aba “Project tree”, selecionar a opção “PLC\_1/PLC tags/PLC tags (n)” e usar o clique duplo para determinar o nome da TAG, como mostra a figura 5.1.

Para exemplificar, considere a criação de duas TAGs, uma booleana e uma inteira. A booleana com o nome “TagPLC” e a inteira com o nome “TagInteira”. Para determinar o tamanho de uma TAG, é necessário alterar o campo “Data type”, como mostram as figuras 5.2 e 5.3.

Ao criar a TAG booleana, seu endereço é sugerido automaticamente pelo programa, sendo necessário determinar se será uma TAG de entrada, saída ou memória, como mostram as figuras 5.4 e 5.5.

Tabela 5.2: Tipo de variáveis de sistemas de automação SIEMENS

| Tipo de Variável | Comprimento (bits) | Formato Padrão         | Faixa de Valores                                                    |
|------------------|--------------------|------------------------|---------------------------------------------------------------------|
| BOOL             | 1                  | Boolean                | TRUE/False                                                          |
| BYTE             | 8                  | Hexadecimal number     | 16#0 a 16#FF                                                        |
| WORD             | 16                 | Hexadecimal number     | 16#0 a 16#FFFF                                                      |
| DWORD            | 32                 | Hexadecimal number     | 16#0000_0000 a 16#FFFF_FFFF                                         |
| SINT             | 8                  | Signed integers        | -128 a 127                                                          |
| USINT            | 8                  | Unsigned integers      | 0 a 255                                                             |
| INT              | 16                 | Signed integers        | -32768 a 32767                                                      |
| UINT             | 16                 | Unsigned integers      | 0 a 65535                                                           |
| DINT             | 32                 | Signed integers        | -2147483648 a 2147483647                                            |
| UDINT            | 32                 | Unsigned integers      | 0 a 4294967295                                                      |
| REAL             | 32                 | Floating-point numbers | -3.402823e+38 a -1.175495e-38<br>±0<br>1.175495e-38 a +3.402823e+38 |
| TIME             | 32                 | Time period with sign  | T#-24d20h31m23s648ms a<br>T#+24d20h31m23s647ms                      |
| CHAR             | 8                  | ASCII characters       | ASCII character set                                                 |

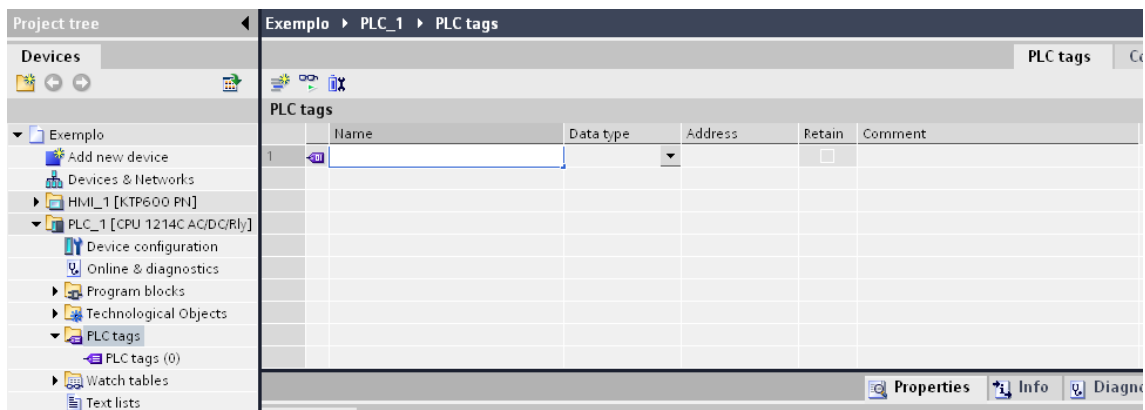


Figura 5.1: Criando uma nova TAG do CLP.

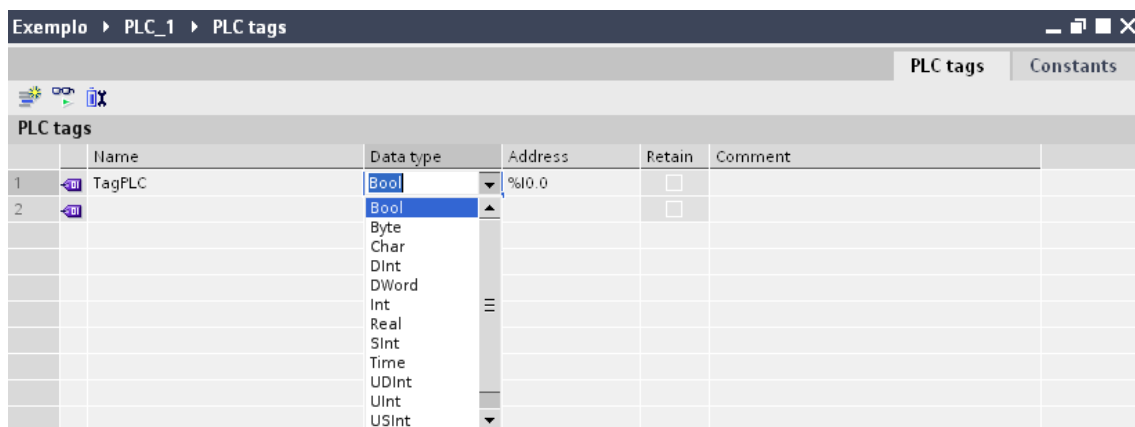


Figura 5.2: Determinando o tamanho da TAG - Booleana.

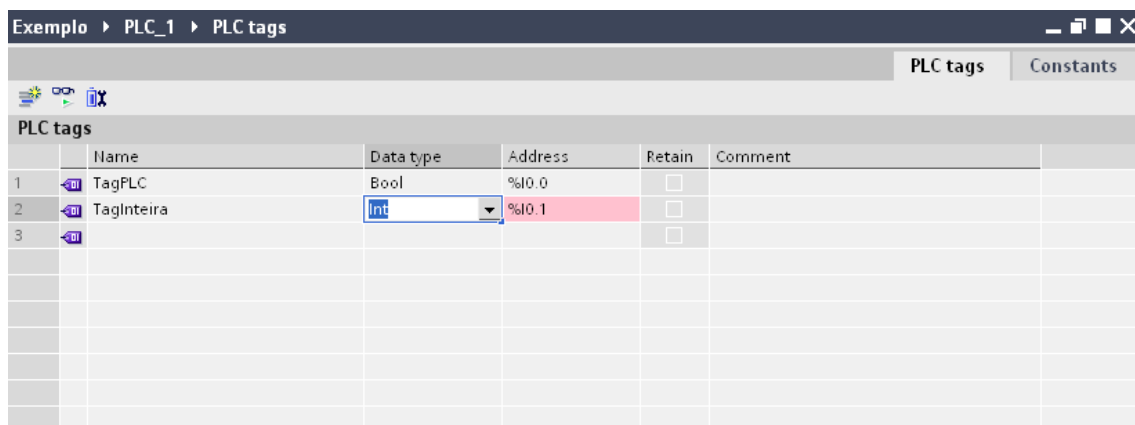


Figura 5.3: Determinando o tamanho da TAG - Inteira.

Ao criar a TAG inteira, o TIA Portal seguiu o mesmo padrão da TAG booleana, copiando o padrão da TAG anterior. Por esse motivo, o endereçamento ficou incorreto e marcado em vermelho.

Para corrigir o endereçamento, abra a janela de alteração do endereço da TAG e confirme a alteração, como mostram as figuras 5.6 e 5.7, levando em consideração que a sobreposição de endereçamentos pode levar ao mau funcionamento do sistema. Durante a correção do endereçamento, é sugerido determinar o tipo da TAG.

### 5.3.2 TAG da IHM

Para criar uma TAG na IHM, é necessário, na aba “Project tree”, selecionar a opção “HMI\_1/HMI tags/HMI tags (n)” e usar o clique duplo para determinar o nome da TAG, como mostra a figura 5.8.

Para exemplificar, considere a criação de duas TAGs, uma interna e uma externa, sendo a interna com o nome “Tag Interna” e a externa com o nome “Tag Externa”.

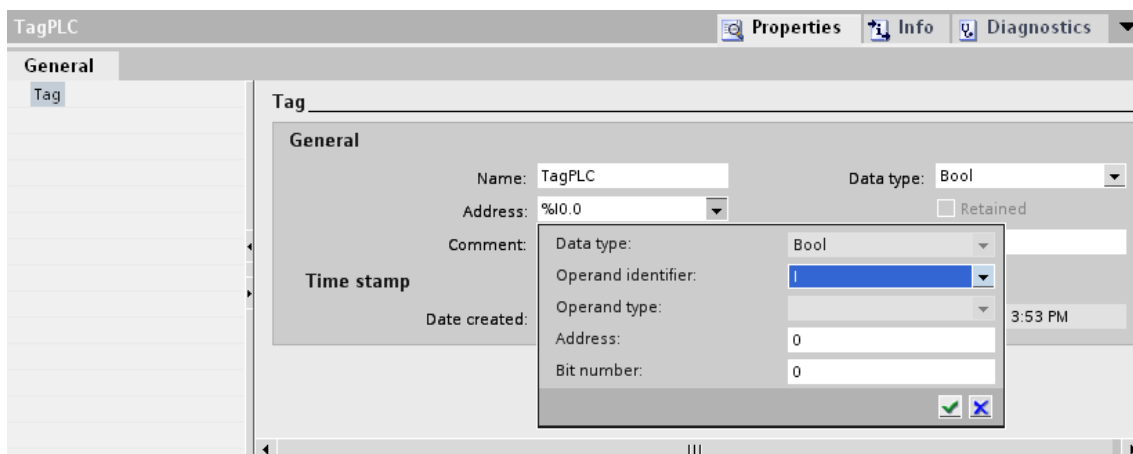


Figura 5.4: Determinando o endereço da TAG - Booleana.

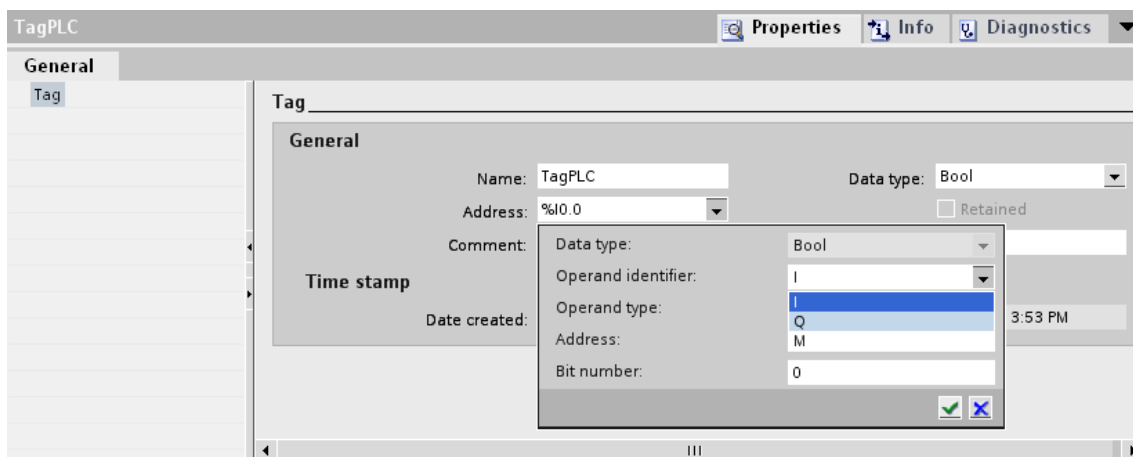


Figura 5.5: Determinando o tipo da TAG - Booleana.

Em uma TAG interna, é necessário determinar o tamanho da TAG, alterando o campo “Data type”, como mostra a figura 5.9.

A TAG externa terá seu tamanho determinado pelo tamanho da TAG do CLP. No campo, “PLC tag” determine qual será a TAG do CLP que servirá de referência para a TAG externa, como mostra a figura 5.10. Note que o campo “Connection” foi alterado automaticamente para “HMI\_connection”, como mostra a figura 5.11.

As propriedades das TAGs dividem-se em 6 campos: “General”, “Limits”, “Linear scaling”, “Settings”, “Comment” e “Multiplexing”.

O campo “General” representa as mesmas opções de nome, conexão, tipo e endereço da lista de TAGs encontradas nas figuras 5.8, 5.9, 5.10 e 5.11; em “Linear scaling”, é possível determinar o valor da TAG da IHM em relação à TAG do CLP. Caso esta opção não seja selecionada, então o ganho será unitário; “Settings” permite determinar o valor inicial da TAG e, mesmo com o valor inicial definido, se a TAG for externa ela poderá ter seu valor alterado pelo CLP; e, em “Comment”, é possível

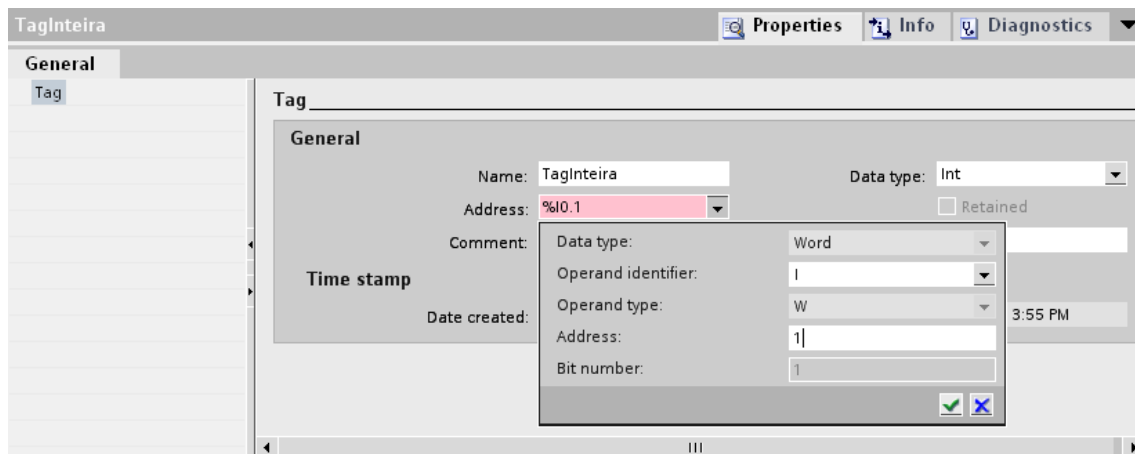


Figura 5.6: Determinando o endereço e o tipo da TAG - Inteira.

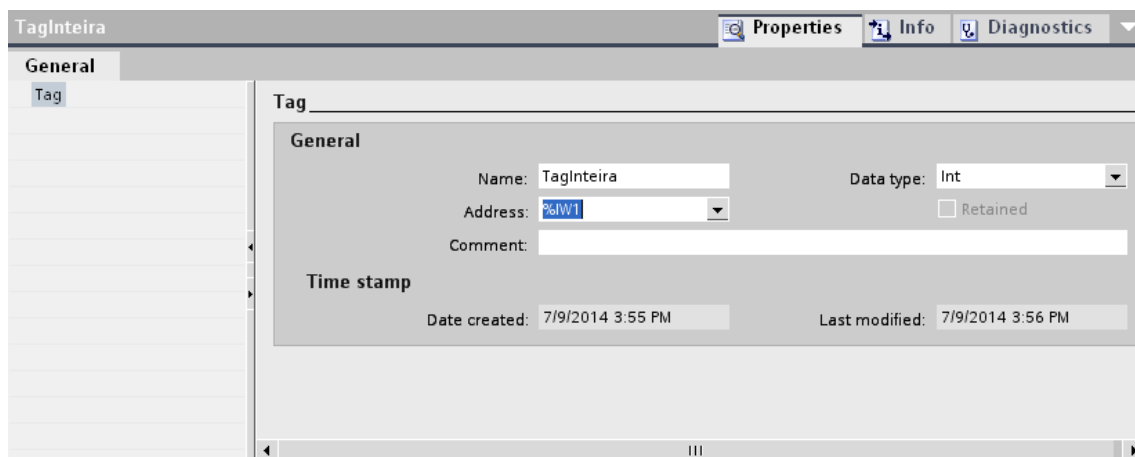


Figura 5.7: Determinando o endereço e o tipo da TAG - Inteira.

descrever com palavras, dentro do software, as características e funcionalidades da TAG.

“Limits” permite determinar um limite máximo e um limite mínimo. Esses limites são restrições extremas e não é possível ultrapassá-los com comandos internos à IHM. Ao tentar utilizar uma função da IHM para alterar o valor da TAG e ultrapassar esses limites, o sistema acusa erro e ignora o comando. Contudo, caso a TAG seja alterada por uma função do CLP e o limite seja ultrapassado, a IHM não impede a alteração de valor. Nesses casos, é aconselhável que se crie alarmes para corrigir o problema ou ações dependentes de eventos da própria TAG. A diferença entre essas duas formas de ultrapassar o limite é que, se a correção for feita a partir de um evento da TAG, ela só poderá ser escolhida para ser executada quando o limite for de fato ultrapassado, ou seja, quando a alteração da TAG partir de um comando do CLP, mas não será executada caso a IHM não permita a alteração de seu valor para um valor acima do limite. Para o alarme, ações podem ser executadas quando o

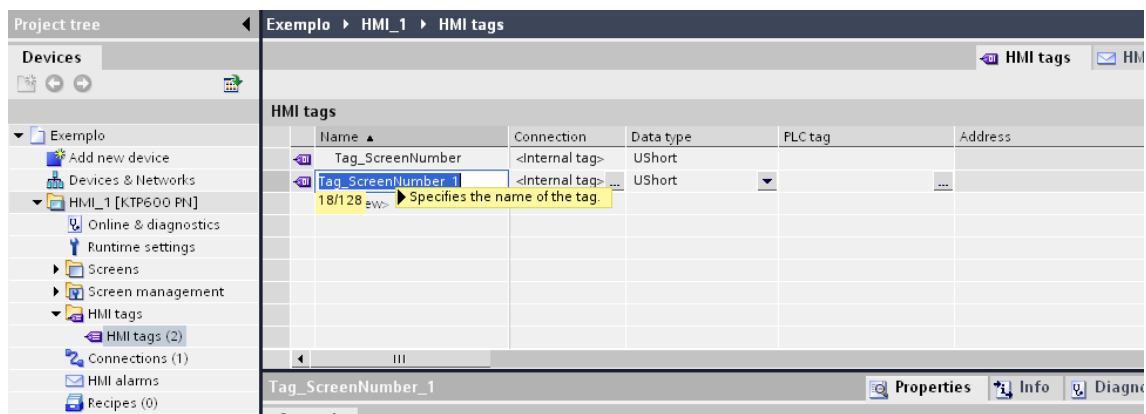


Figura 5.8: Criando uma nova TAG da IHM.

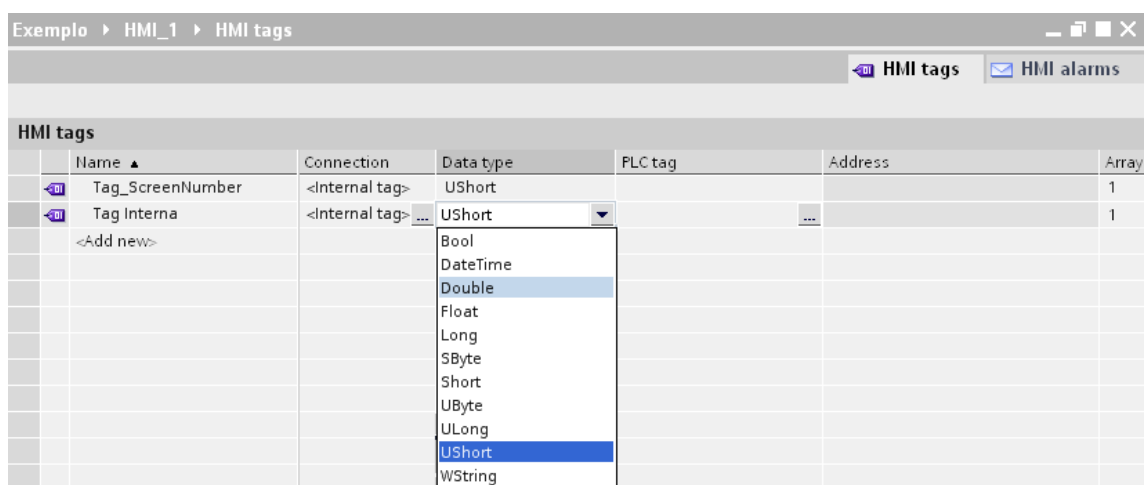


Figura 5.9: Determinando o tamanho da TAG - Interna.

limite for ultrapassado, quando o limite deixar de ser ultrapassado, quando o usuário tomar conhecimento do alarme ou se o alarme persistir.

“Multiplexing” permite determinar que a TAG poderá assumir diversos valores predeterminados, dependendo de uma TAG de referência, podendo fazer analogia a um vetor utilizado em programação. A TAG de referência determinará o valor do “Index” e a TAG multiplex assumirá o valor da TAG de saída armazenada no endereçamento indicado. As TAGs de saída devem ser previamente listadas no campo “Multiplexing” e referenciadas por um valor “Index”.

As TAGs multiplex devem ser TAGs internas e terão seus tamanhos determinados pelas TAGs de saída. As TAGs de saída e de referência devem ter tamanhos específicos para o reconhecimento de seus valores. Os possíveis tamanhos das TAGs estão descritos na tabela 5.3.

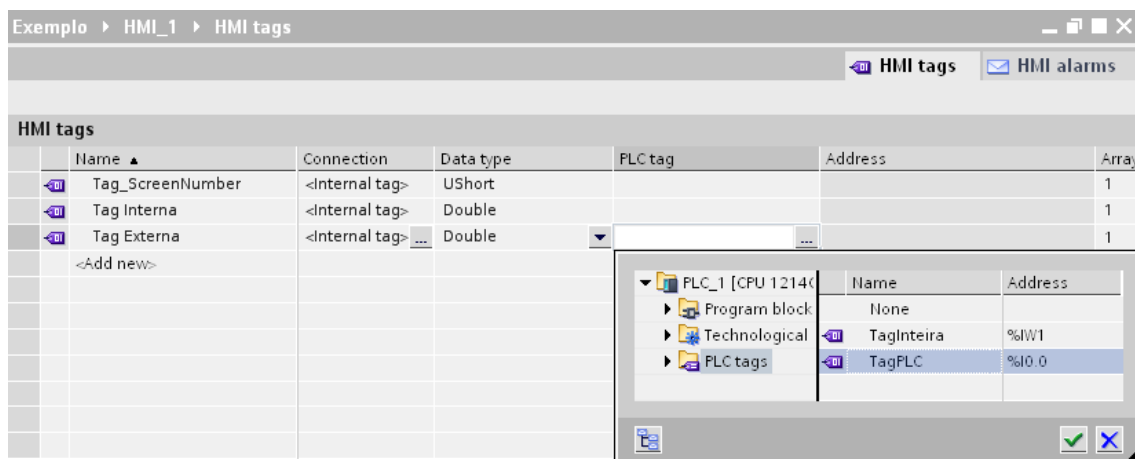


Figura 5.10: Determinando a TAG do CLP ou TAG de referência - Externa.

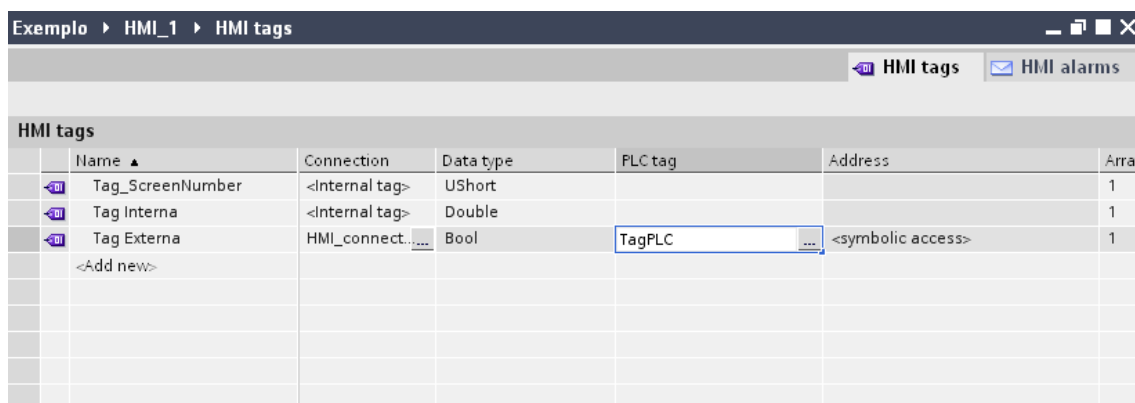


Figura 5.11: Identificação da TAG externa.

Tabela 5.3: Tipo de variáveis aceitas pelas TAGs multiplex

| TAG multiplex e TAG de saída |        | TAG de referência |        |
|------------------------------|--------|-------------------|--------|
| Bool                         | SByte  | Bool              | SByte  |
| Double                       | Short  | Double            | Short  |
| Float                        | ULong  | Float             | ULong  |
| Long                         | UShort | Long              | UShort |
| WString                      |        |                   |        |

Em relação aos eventos, as TAGs aceitam o evento “Value change”, que executa uma ação quando o valor da TAG é alterado, e os eventos “High High limit violated” e “Low Low limit violated”, que executam as ações quando os limites são ultrapassados. Lembrando que esses limites só são ultrapassados quando o valor é alterado pelo CLP. As TAGs multiplex não aceitam eventos.

Além das TAGs, outro componente importante de um projeto são as telas.

## 5.4 Telas

Em um sistema automatizado com IHM, as telas podem representar estados, pequenas partes ou uma visão geral do sistema. Um bom sistema começa com uma boa definição da árvore de telas.

Ao definir a árvore de telas na configuração inicial da IHM, os botões de mudança de tela são criados automaticamente. As telas não precisam necessariamente ser definidas neste instante, podendo criar e editar a árvore de telas após o início do projeto.

### 5.4.1 Criação de telas

Para adicionar uma tela nova, na aba da esquerda (“Project Tree”), abra as opções “HMI/Screens” e dê clique duplo em “Add screen”, como mostra a figura 5.12.

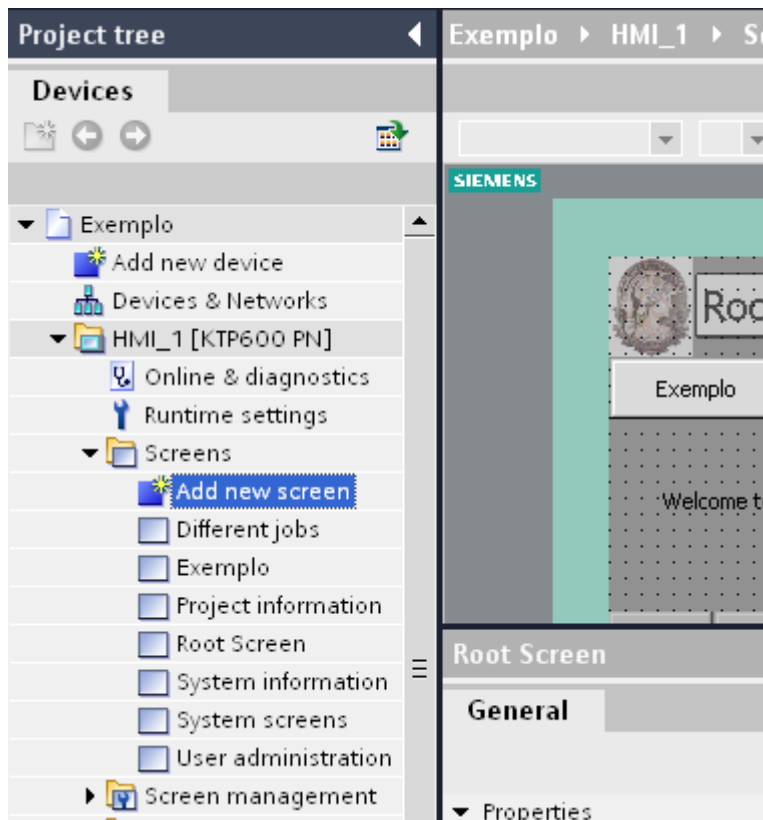


Figura 5.12: Adicionando tela ao projeto da IHM.

Ao criar uma tela, a tela não entra automaticamente na árvore de telas e, para que seja colocado na posição certa dentro da árvore de telas, é necessário que se coloque um botão para cada tela que deriva desta e um botão dentro de cada tela que leva a esta tela. Copie um botão de mudança de tela já pronto ou selecione outro botão na caixa de ferramentas (“Toolbox”) na aba da direita. Com o botão criado, ajuste ou crie um evento para a mudança de tela.

Para criar ou alterar um evento, selecione o botão desejado e abra as propriedades na aba abaixo da tela. Abra os eventos e crie/altere o evento “Click” (podendo ser qualquer evento, dependendo do instante em que o evento deve ser executado) definindo como ação “ActivateScreen” e escolha a tela desejada, como mostra a figura 5.13.

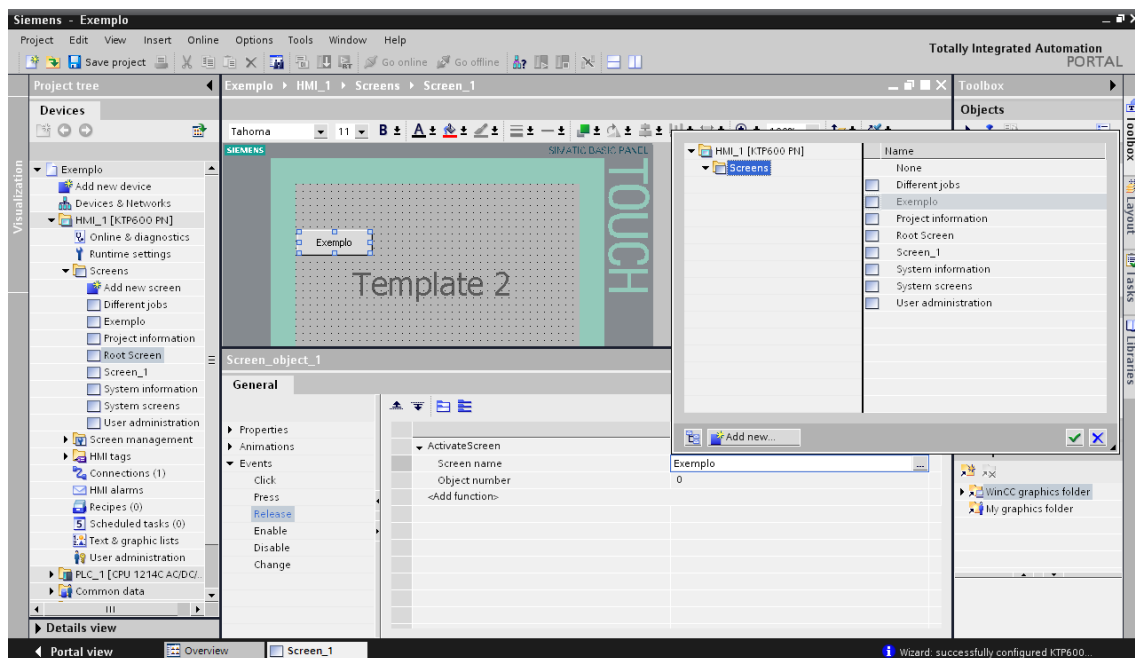


Figura 5.13: Determinando evento de mudança de tela.

Após criar a tela, é necessário conhecer e configurar suas propriedades.

## 5.4.2 Propriedades, animações e eventos de uma tela

As propriedades de uma tela são divididas em apenas dois campos: “General” e “Layers”. Os eventos também dividem-se em dois: “Loaded” e “Cleared”. Em relação às animações, as telas só permitem um tipo: “Visibility”, que determina quando a tela será visível, dependendo de uma TAG.

Nos campos de propriedade, constam os campos “General”, que permite editar o nome da tela, a cor do fundo, a cor da grade (pontos que aparecem na tela, apenas no TIA Portal, para auxiliar o posicionamento de elementos), o número, qual template ela seguirá e a descrição interna dela (que também só aparece no TIA Portal); e “Layers”, que permite determinar quais camadas irão aparecer na tela ao mesmo tempo, sendo que cada elemento permite a escolha de sua camada.

Os eventos da tela, como dito anteriormente, dividem-se em dois tipos, “Loaded” e “Cleared”. O evento “Loaded” ocorre quando a tela é carregada, ou seja, quando ocorre a transição de outra tela para a tela em questão. O evento “Cleared” ocorre

quando a tela é descarregada, ou seja, quando ocorre a transição da tela em questão para outra tela.

As telas não são suficientes para representar o sistema automatizado sem que hajam outros elementos distribuídos nela.

## 5.5 Passo 1: Adicionar um novo dispositivo

Para iniciar o projeto, é necessário adicionar o painel IHM como um novo dispositivo ao projeto do sistema automatizado, como ensinado no Capítulo 4. Ao adicionar a IHM, a janela “HMI Device Wizard” irá iniciar para que sejam feitos os ajustes iniciais para auxiliar no projeto.

Existem seis ajustes que podem ser feitos na janela “Wizard”: “PLC Connections” (Conexões do CLP); “Screen Layout” (Composição da tela); “Alarms” (Alarmes); “Screens” (Telas); “System screens” (Telas do sistema); e “Buttons” (Botões).

No ajuste das conexões do CLP, você irá selecionar qual CLP se conectará com a IHM. Clique em “Browse...”, escolha o “PLC\_1”, figura 5.14 e aperte “Next»”.

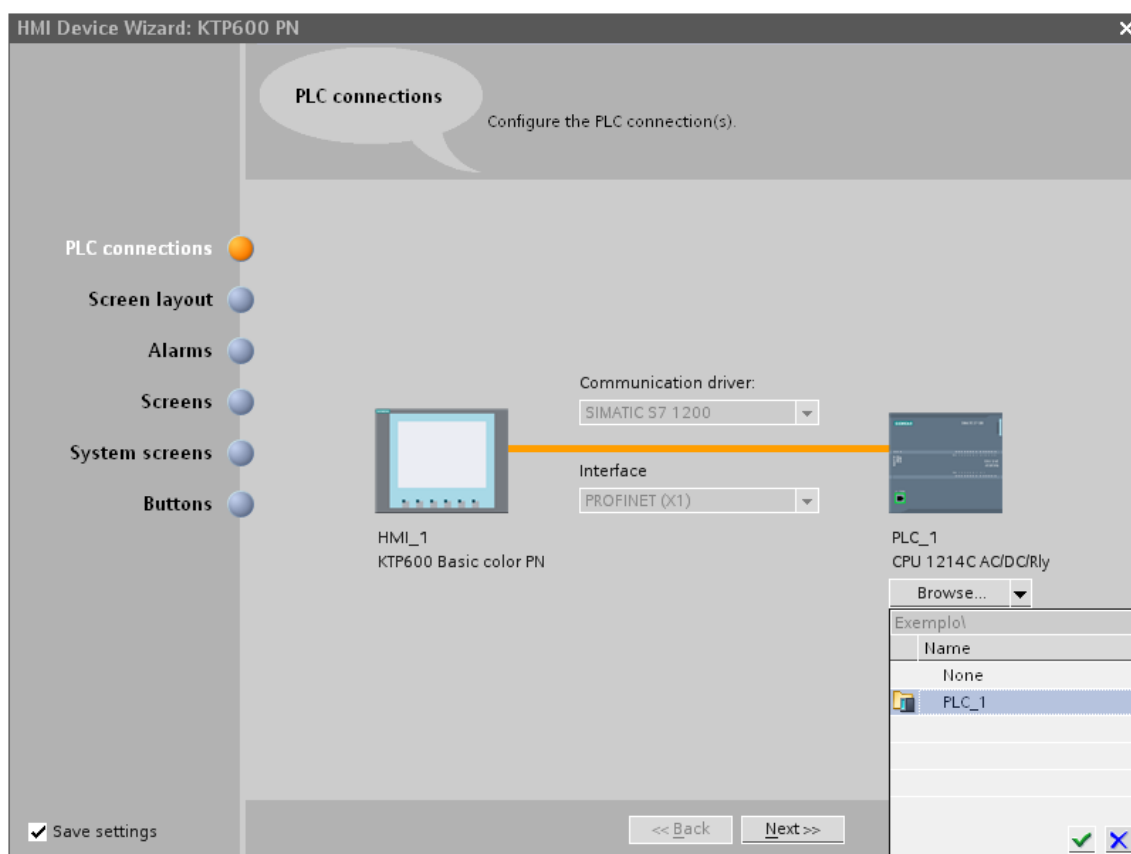


Figura 5.14: Configuração da IHM – Conexão com o CLP.

Para o ajuste da composição da tela inicial, será escolhido o template 1. Contudo, outros templates podem ser criados posteriormente. É possível escolher a cor do

plano de fundo e escolher se terá cabeçalho (Data/Hora e Logomarca), sendo que a logomarca pode ser importada do computador. Após selecionar tudo, clique em “Next»”.

Para o projeto do sistema supervisorio, foram escolhidos a cor cinza para o fundo de tela e um cabeçalho com logomarca e data/hora. A logomarca foi importada como o símbolo da UFRJ, a Minerva, como mostra a figura 5.15.

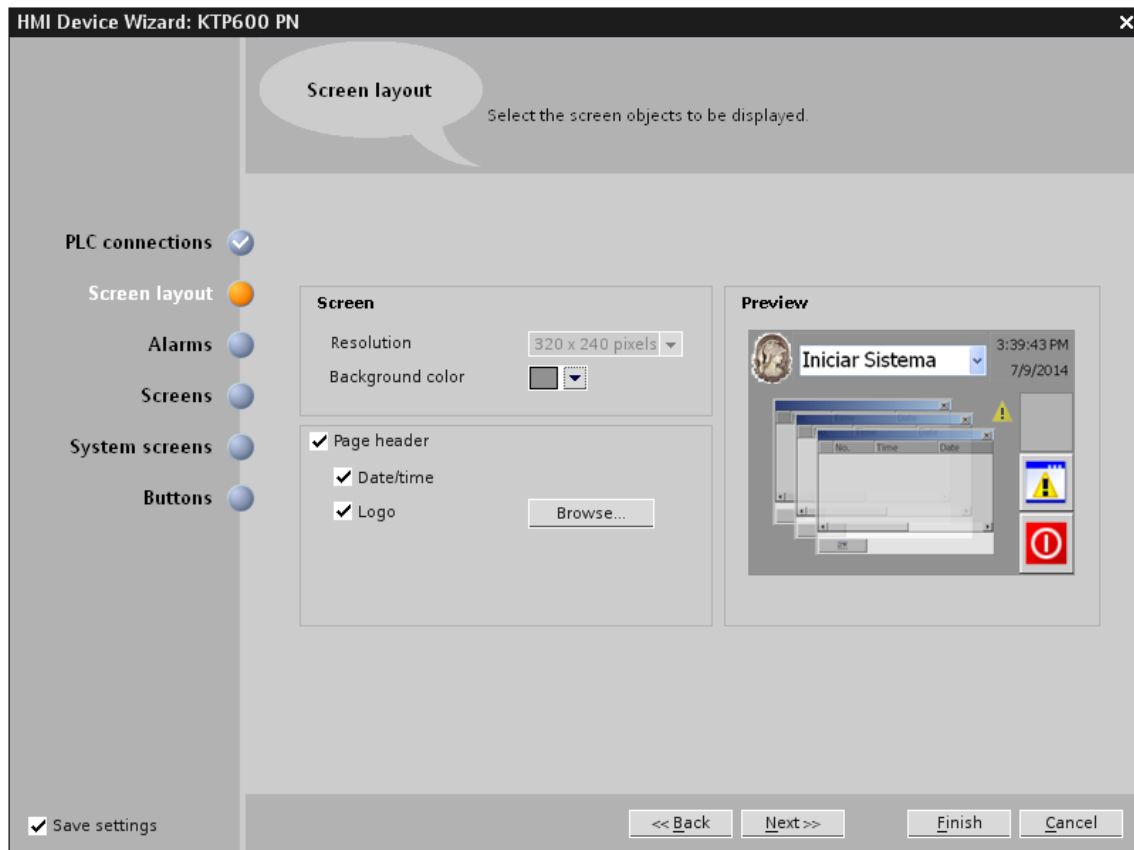


Figura 5.15: Configuração da IHM – Template 1.

No ajuste de alarmes é possível escolher os tipos de alarmes que terão telas do sistema para alertar o operador, “Unacknowledged alarms”, “Active alarms” e “Active system events”, como mostra a figura 5.16.

“Unacknowledged alarms” são alarmes que ainda não foram reconhecidos pelo usuário; “Active alarms” são os alarmes que se encontram acionados; e “Active system events” são as ações do sistema, como ligar, desligar e ativar o sistema operante na IHM. Após selecionar, clique em “Next»”. Todas as opções de alarme foram selecionadas para o projeto supervisorio.

Na opção “Screens”, é possível determinar como será a árvore de telas (figura 5.17), mas, como tudo nesta configuração inicial, é possível alterar posteriormente. Para apagar uma tela desta árvore, aperte “Delete” e para criar uma nova tela, clique em “+”. Para alterar o nome de qualquer tela, clique no nome da tela em questão.

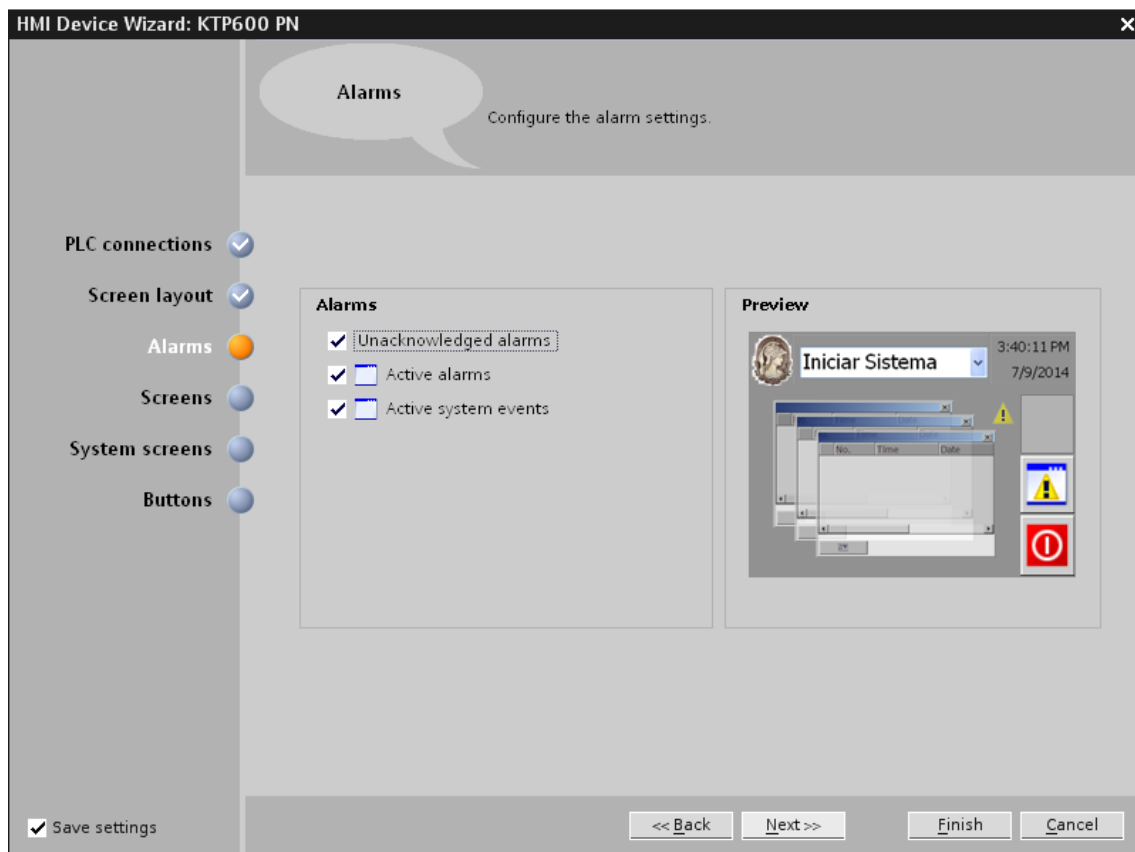


Figura 5.16: Configuração da IHM – Avisos de alarme.

Ao finalizar, clique em “Next»”.

Neste projeto, serão criadas cinco telas. Uma tela inicial, apenas para apresentação; uma tela de visão geral, para que todos os componentes do sistema sejam visualizados unicamente; uma tela da esteira, para que as ações e condições da esteira e da torre de peças possa ser representada com mais detalhes; uma tela do braço, para que os movimentos e posições do braço sejam representadas com mais detalhes; e uma tela da prateleira, para que as peças possam ser representadas após serem posicionadas na esteira e para que o usuário possa escolher qual peça deseja retirar da prateleira.

Com um esboço das telas, é possível criar uma árvore de telas que permita o usuário transitar entre as telas de forma intuitiva. A árvore de telas criada para o projeto supervisorio pode ser visualizada na figura 5.18.

No ajuste de telas do sistema, é possível determinar quais telas do sistema terão no projeto, como mostra a figura 5.19.

A tela “Root Screen” é a mesma que aparece na árvore de telas. É a tela inicial do sistema e onde irá aparecer o botão para a tela “System screens”.

“System screens” é onde ficam os botões para a escolha de qual tela do sistema se deseja visualizar.

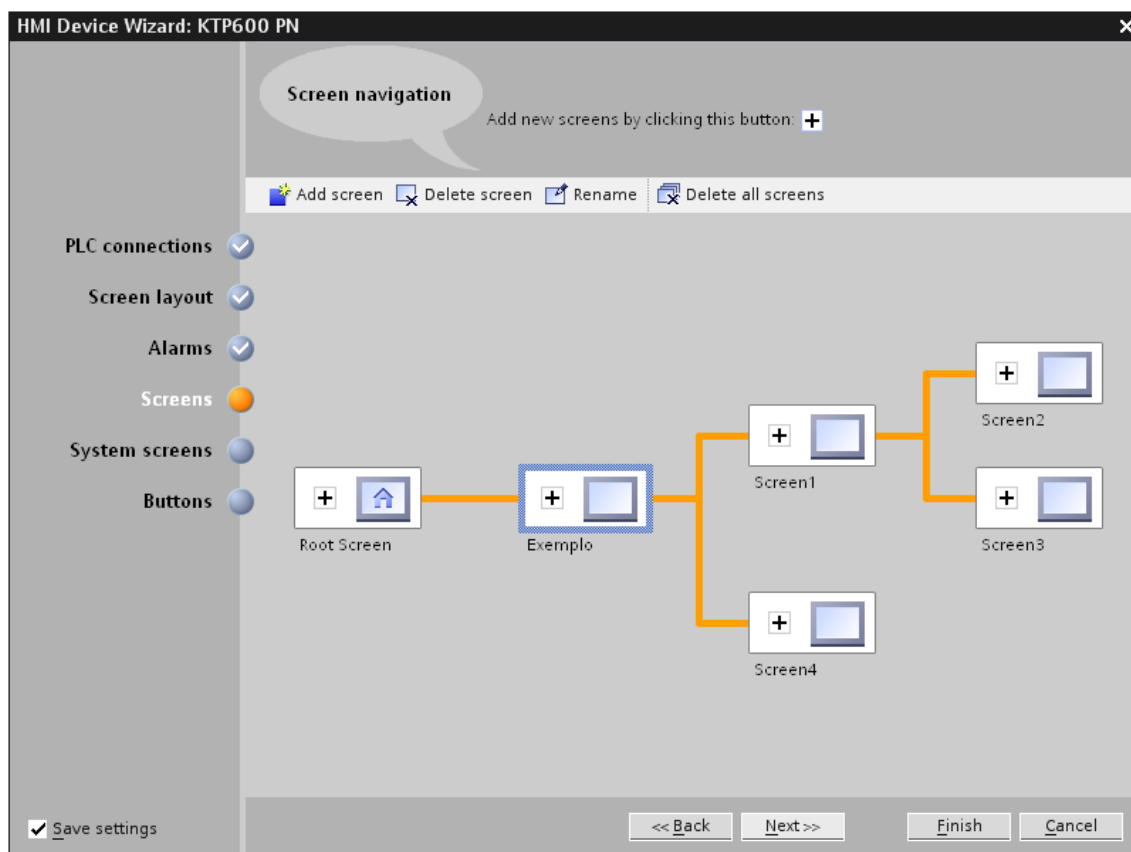


Figura 5.17: Configuração da IHM – Escolha da árvore de telas.

Em “Project information” é possível encontrar informações do projeto, como nome, data de criação, autor e descrição.

A tela “Different jobs”, a única tela que possui opções sobre quais funções existirão na tela, conta com cinco botões, caso todas as opções sejam selecionadas. Caso selecione a opção “Operating modes”, os botões “Online”, “Offline” e “Transfer” aparecerão na tela. Esses botões servem para alterar o modo de operação para online ou offline, ou transferir um projeto ou uma atualização do projeto do computador para a IHM. Caso selecione a opção “Language switching”, o botão “Toggle language” estará disponível na tela e serve para alterar a língua da IHM. A opção “Stop Runtime” torna um botão, com o mesmo nome, disponível. Este botão serve para parar os processos internos da IHM.

“User administration” contém uma lista de todos os usuários e seus respectivos grupos.

Por fim, a tela “System information” contém informações mais detalhadas sobre a versão e utilização da IHM. As informações existentes na tela são a versão da IHM, o tipo de conexão e qual CLP encontra-se conectado.

Ao selecionar as telas desejadas, clique em “Next»”. Contudo, nenhuma tela do sistema será necessária para o projeto supervisório.

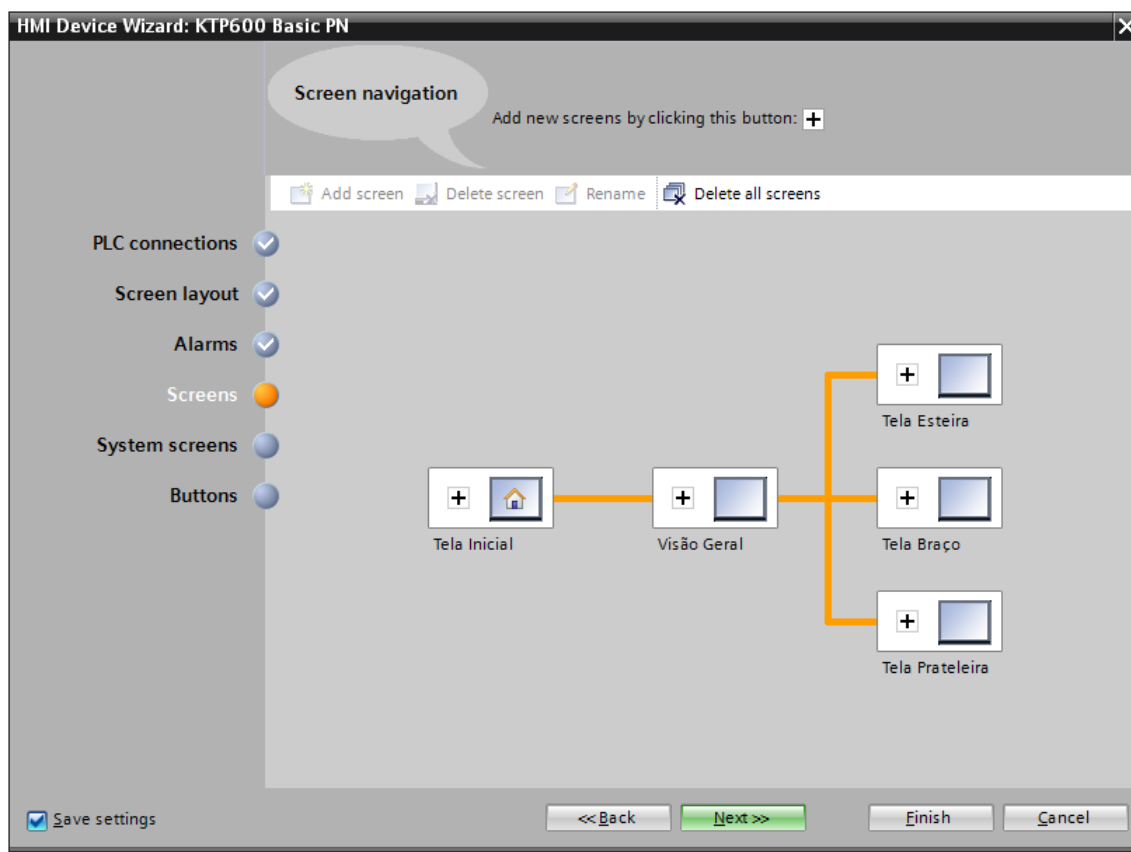


Figura 5.18: Configuração da IHM – Árvore de telas escolhida para o projeto.

Na opção “Buttons”, é possível escolher quais botões do sistema serão adicionados ao template 1 e a posição deles. Eles podem ser posicionados à direita, à esquerda ou abaixo, como mostra a figura 5.20.

Existem cinco possíveis botões para selecionar. O único botão que é selecionado na opção “Alarms” é o botão “Unacknowledged alarms”, que só aparece se esta caixa for marcada. Este botão mostra a lista de alarmes que ainda não foram identificados pelo usuário.

Existem também os botões “Start screen”, que leva automaticamente para a tela inicial; “Log in”, que permite que um usuário entre com uma conta pré-cadastrada, sendo que algumas contas podem ter funcionalidades bloqueadas; “Language”, que permite a escolha de idioma da IHM; e “Exit”, que desliga o sistema operante na IHM. Após selecionar os botões, clique em “Finish”.

Todos os botões foram escolhidos para fazerem parte do template 1 e serão posicionados à direita. O template 1 será utilizada apenas na tela inicial. Será criado outro template para ser utilizado nas demais telas. A escolha de botões para o projeto supervisorio pode ser observada na figura 5.21.

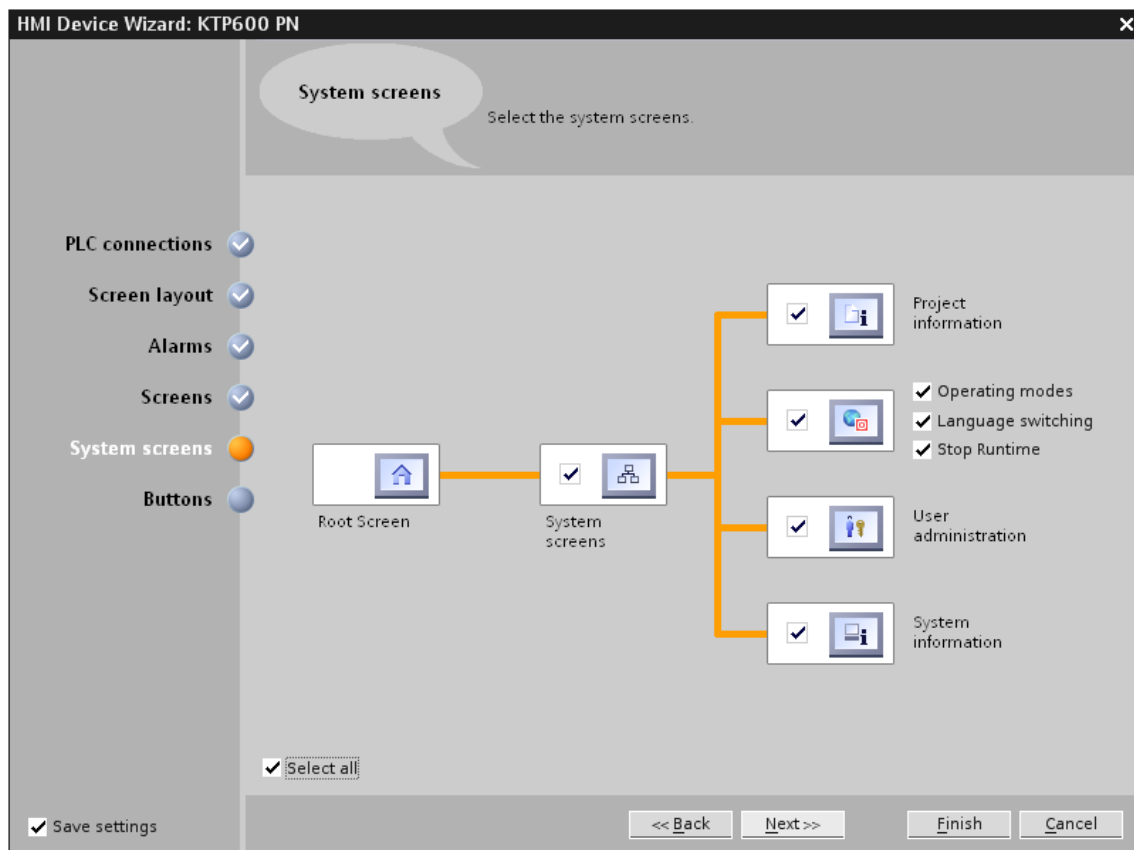


Figura 5.19: Configuração da IHM – Escolha das telas do sistema.

## 5.6 Passo 2: Criação de template

No passo 1, foi criado o template 1, que será usada na tela inicial e, para as demais telas, será criado outra template. Um template não interfere no outro, mas é possível alterar o template global para que algum objeto apareça em todas as telas, independente do template escolhido.

O template global contém todos os avisos de alarme e permite gerenciar e editar estes avisos, ou adicionar novos avisos de alarme. O template global fica localizada na opção “Global Screen”, em “Project tree”.

Para adicionar um novo template, na barra da esquerda, em “HMI\_1/Screen management/Templates” selecione a opção “Add new template”. Nas propriedades do template, é possível escolher o nome, a cor de fundo, a cor da grade de auxílio (esta grade não aparecerá na IHM) e quais camadas aparecerão neste template.

As camadas seguem um conceito que serve para todas as telas. É possível determinar em qual camada o objeto estará localizado. As camadas com maior numeração aparecerão acima das camadas com menor numeração e é possível desabilitar uma camada inteira nas propriedades da tela ou do template.

Para construir o template da forma desejada, é necessário utilizar objetos como

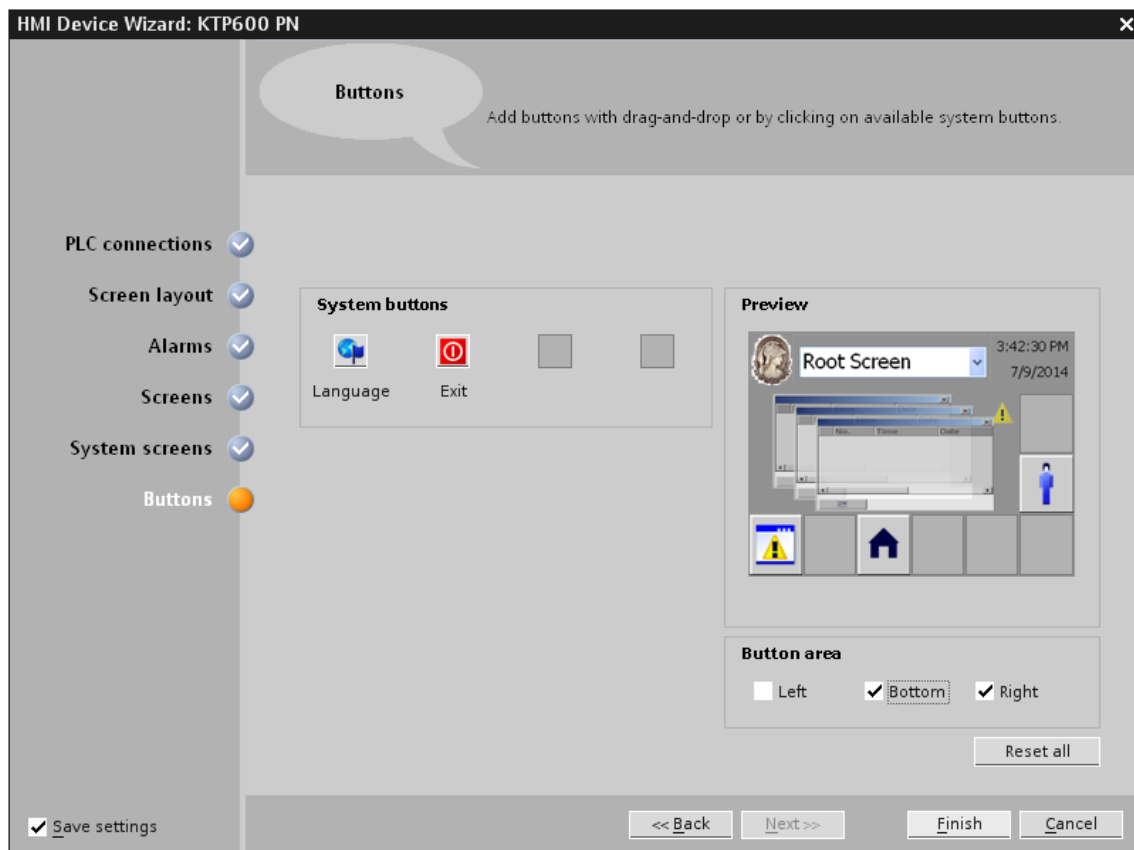


Figura 5.20: Configuração da IHM – Escolha dos botões do sistema.

figuras, botões e outros. Esses objetos serão descritos neste trabalho quando usados pela primeira vez.

### 5.6.1 Figuras

As figuras auxiliam na representação de um sistema físico por poderem ser figuras importadas do computador, ou seja, permite que imagens mais elaboradas sejam produzidas em outro software. As figuras fazem parte do grupo de objetos inanimados e, portanto, não possuem eventos, como pressionar ou ligar.

As figuras podem ser editadas a partir de suas propriedades, e elas dividem-se em “General”, “Appearance”, “Layout” e “Miscellaneous”.

Em “General”, é possível selecionar qual imagem será usada, podendo escolher imagem importada do computador ou imagem do banco de imagens do TIA Portal; “Appearance” permite determinar a cor de fundo e as definições da borda; as propriedades de “Layout” são utilizadas para determinar a posição e o tamanho do elemento e como a imagem se posicionará em relação à caixa do elemento; por fim, em “Miscellaneous”, é possível editar o nome e escolher em qual camada ela ficará em uma tela.

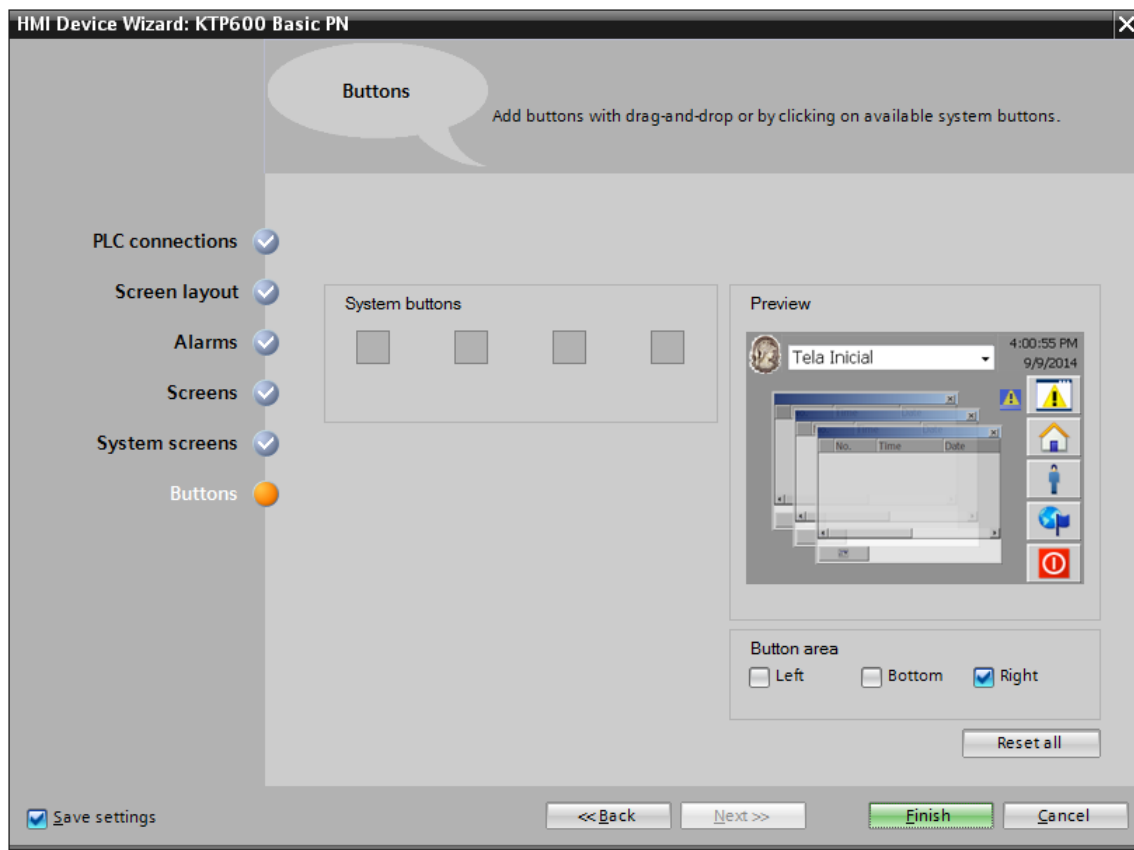


Figura 5.21: Configuração da IHM – Botões escolhidos para o projeto supervisório.

Para o template 2, foram usadas três figuras iguais para representar quando os botões “Start”, “Stop” e “Quit” estarão desabilitados. Foi escolhida uma imagem de um botão quadrado cinza no banco de imagens do TIA Portal, replicada três vezes e posicionada no lugar desejado. Essas imagens foram alocadas na camada zero para que os outros elementos sobreponham-na quando estiverem habilitados.

A seguir, é necessário nomear as figuras, utilizando textos.

### 5.6.2 Texto

Os textos são como figuras, mas podem conter apenas caracteres e uma envoltória retangular. Eles são facilmente editados sem a necessidade de outro software, diferentemente das figuras, são utilizados para descrever elementos ou estados na tela e fazem parte do grupo de objetos inanimados e, portanto, não possuem eventos.

Os textos podem ser editados a partir de suas propriedades, e elas dividem-se em “General”, “Appearance”, “Layout”, “Text format”, “Flashing” e “Miscellaneous”.

O campo “General” possibilita editar o que estará escrito no texto, a fonte das letras e determinar como o formato da caixa de texto se comportará em relação ao tamanho do texto; em “Appearance”, é possível editar a cor do fundo, se o fundo é

sólido ou transparente, a cor do texto, a largura, o estilo e a cor da borda e se a borda terá efeito 3D; “Layout”, como nas figuras, permite determinar a posição e o tamanho da caixa de texto ou se a caixa se adaptará ao tamanho do texto, além de determinar as margens que o texto deve seguir em relação à caixa de texto; em “Text format”, é possível determinar a fonte do texto e o alinhamento em relação à caixa de texto; “Flashing” determina se o texto ficará piscando ou não; e “Miscellaneous” permite a escolha do nome e em qual camada da tela o texto será alocado.

Os textos foram utilizados para distinguir cada botão funcional (“Start”, “Stop” e “Quit”). Para isso, foram criados três textos, cada um com o nome de um botão, e posicionados acima da imagem previamente posicionada. Além de distinguir os botões, os textos foram utilizados para determinar a representação de cada posição do interruptor que determinará se os comandos serão dados pela IHM ou pelo painel de botões do CLP.

Com as imagens de botão e os textos para descrever os botões e o interruptor, é necessário adicionar cada botão e interruptor em sua devida posição.

### 5.6.3 Botão

Botões são objetos interativos que são capazes de executar eventos quando acionados pelo usuário. Esses eventos podem configurar uma ação na própria IHM, como trocar de tela, ou dar alguma resposta ao sistema controlado pelo CLP.

As propriedades de um botão dividem-se em sete campos: “General”, “Appearance”, “Design”, “Layout”, “Text format”, “Miscellaneous” e “Security”.

No campo “General”, é possível determinar um atalho para o botão e se o botão será um texto, uma figura ou invisível, podendo escolher qual será o texto ou a figura que aparecerá quando o botão encontra-se ativo ou desativado; “Appearance” permite determinar a cor de fundo e a cor do texto; “Design” serve apenas para determinar se o botão terá efeito 3D e qual será a cor de foco; seguindo para o campo “Layout”, é possível determinar a posição e o tamanho do botão e como o conteúdo se ajustará de acordo com o elemento; em “Text format”, é possível determinar a fonte do texto e o alinhamento em relação ao botão; no campo “Miscellaneous”, é possível determinar o nome do botão, a camada em que ele se encontrará na tela e descrever com palavras, dentro do software, as características e funcionalidades do botão; e, por fim, em “Security”, é possível determinar quem, dentro do cadastro de usuários, tem autorização para utilizar este elemento.

Os botões permitem seis tipos diferentes de evento, que dividem-se basicamente em dois grupos. Um grupo depende do toque do usuário e outro grupo depende do estado do botão.

Para o grupo que depende do toque do usuário, existem os eventos “Click”, que

executa uma ação quando o usuário pressiona e solta o botão; “Press”, que executa uma ação quando o usuário pressiona o botão; e “Release”, que executa uma ação quando o usuário solta o botão.

Já no grupo que depende do estado do botão, existem os eventos “Enable”, que executa uma ação quando o botão é ativado; “Disable”, que executa uma ação quando o botão é desativado; e “Change”, que executa uma ação quando o botão troca de um estado para o outro.

Para substituir os botões “Start”, “Stop” e “Quit” do painel da planta mecatrônica, foram utilizados botões na IHM que tenham, basicamente, a mesma finalidade. Por ora, os botões apenas serão posicionados (exatamente acima das imagens utilizadas para representar o botão desativado) sem receberem nenhuma determinação de evento, pois o foco atual é planejar e montar a estrutura das telas.

Além da posição dos botões, é necessário determinar a aparência de cada um. Foram escolhidos botões quadrados (semelhantes às imagens dos botões desativados) com uma cor específica para o botão, seguindo a cor do painel da planta mecatrônica. Foi definida uma imagem para cada estado do botão, pressionado e não pressionado. Para finalizar, é importante lembrar de alterar a camada do botão para a camada um, pois só assim ele sobreporá as imagens previamente posicionadas quando encontrarse habilitado.

Após adicionar os botões, será adicionado o interruptor de transição entre o painel IHM e o painel da planta mecatrônica.

#### 5.6.4 Interruptor

O interruptor também é interativo e é utilizado para os mesmos fins do botão, mas com a diferença de que ele permanece pressionado até que o operador aperte novamente.

As propriedades do interruptor são basicamente iguais às do botão, diferindo apenas em “General” e por ter uma oitava opção, “Limits”.

Diferentemente do campo “General” nas propriedades do botão, no interruptor é necessário, também, determinar uma TAG de referência para que a informação de que o interruptor foi acionado seja armazenada e utilizada. Em “Limits”, é possível determinar uma cor para que o elemento assuma caso a TAG de referência ultrapasse certo valor.

Os eventos do interruptor dividem-se em “Change”, “Switch ON”, “Switch OFF”, “Enable” e “Disable”. O evento “Switch ON” executa uma ação quando o interruptor passa de desligado para ligado e o evento “Switch OFF” executa uma ação quando o interruptor passa de ligado para desligado; já o evento “Change” executa uma ação quando o interruptor alterna de um estado para o outro; por fim, o evento “Enable”

executa uma ação enquanto o interruptor encontra-se ligado e “Disable” executa uma ação enquanto o interruptor encontra-se desligado.

Para que o interruptor permaneça ativado enquanto o operador não o pressione novamente, é necessário que se defina ações para os eventos “Switch ON” e “Switch OFF”. Essas ações devem ser ambas “Set TAG” e deve-se escolher a TAG de referência do interruptor. Para “Switch ON”, determine o valor da TAG como 1 (um) e, para “Switch OFF”, determine o valor da TAG como 0 (zero).

O interruptor foi utilizado para que o operador possa alternar entre comandar a planta mecatrônica a partir do painel IHM ou a partir do painel da planta mecatrônica.

Após concluir a edição da aparência do template 2, altere o template de cada tela, com exceção da tela inicial, para o template 2 em “General/Template”. O template 2 está representado na figura 5.22.

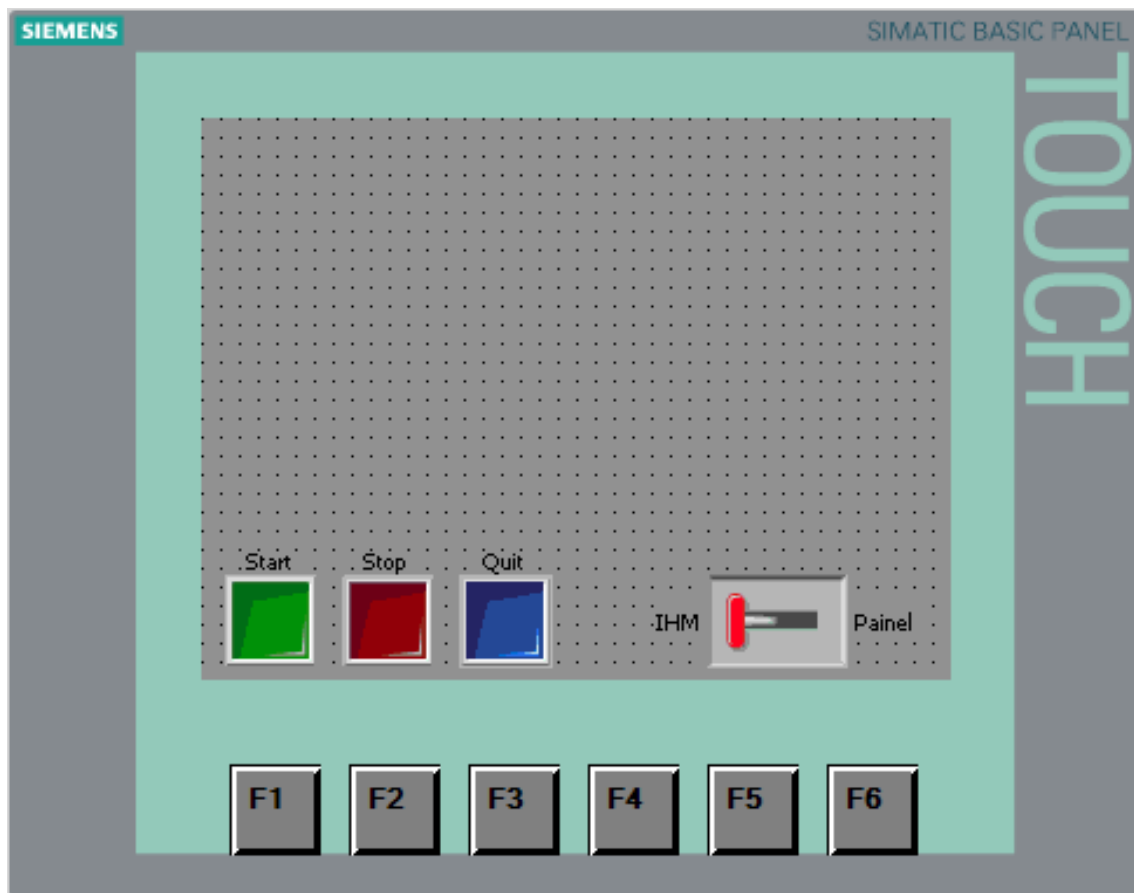


Figura 5.22: Configuração da IHM – Template 2 concluído.

O próximo passo a ser seguido é editar as telas do projeto.

## 5.7 Passo 3: Edição de telas

No passo 3, será descrito como as telas do projeto supervisorio foram planejadas e editadas na IHM. Como dito anteriormente, o projeto supervisorio terá cinco telas e cada tela terá um objetivo específico.

### 5.7.1 “Tela Inicial”

Esta será a tela inicial, de apresentação, do projeto, onde serão dadas as boas vindas e constarão algumas informações importantes, como data, hora, nome da tela e logomarca da universidade. Constará, também, o botão para iniciar o projeto, mas não será possível utilizar os botões funcionais, como “Start”, “Stop” e “Quit”, a partir dela.

Para inserir a logomarca da universidade, o nome da tela, a data e a hora, garanta que, nas propriedades gerais da tela, no campo “Template” esteja selecionado o template 1, caso tenha seguido à risca este tutorial. Caso contrário, se algum destes elementos não encontrar-se na tela, ele deverá ser inserido manualmente no template ou na própria tela.

Para a logomarca, insira uma figura importada do computador com a logomarca da universidade e posicione-a no canto superior esquerdo da tela (seção 5.6.1).

Para inserir o nome da tela, é necessário criar uma lista de dados, que será explicada a seguir.

#### 5.7.1.1 Listas de dados

As listas de dados servem para definir um texto ou uma figura dependendo de sua indexação. Cada elemento da lista recebe um valor de indexação e é chamado quando a TAG de referência da lista assume o determinado valor.

Uma lista de dados consiste em um nome, uma gama de valores que a lista permite, e uma descrição. Dentro desta lista, seus elementos são definidos por valores e o texto ou a figura referente a cada valor. Dentre estes elementos, é possível determinar um elemento padrão, que define o valor inicial da lista.

Para criar o projeto nos padrões descritos, é necessário criar uma lista de dados de texto com os nomes de cada tela para cada indexação.

Além da lista, é necessário adicionar um elemento que acesse os dados desta lista.

#### 5.7.1.2 Elemento de entrada e saída de dados

Cada elemento de entrada e saída de dados deve ser referenciado a uma lista de dados e a uma TAG de referência, podendo ser de texto ou figura. A TAG de referência percorre a lista de dados para que o elemento seja apresentado ao operador, tendo

a possibilidade do usuário selecionar um elemento da lista para alterar o valor da TAG de referência.

As propriedades de um elemento de entrada e saída de dados variam, dependendo se é um elemento de texto ou figura. Em geral, elas dividem-se em seis campos: “General”, “Appearance”, “Layout”, “Limits”, “Miscellaneous” e “Security”. Quando o elemento é composto por um texto, é possível editar o campo “Text format”.

Em “General”, é possível escolher a TAG de referência, a lista de dados que fará parte do conteúdo e se o elemento será apenas de entrada, apenas de saída ou de entrada e saída; já em “Appearance”, é possível editar as cores e a borda do elemento; “Layout” permite determinar a posição e o tamanho do elemento na tela; “Limits” permite determinar a cor do campo de dados caso a TAG de referência ultrapasse o limite superior ou o limite inferior; em “Miscellaneous”, é possível determinar o nome, a camada que ele ficará em uma tela e descrever com palavras, dentro do software, as características e funcionalidades do elemento; por fim, em “Security”, é possível determinar quem tem autorização para utilizar este elemento.

Para os elementos de texto, é possível editar a fonte em que o texto aparecerá para o operador, no campo “Text Format”.

Os eventos dos elementos de entrada e saída de dados dividem-se em três: o evento “Enable”, que é acionado quando o campo é ativado para uso; o evento “Disable”, que aciona quando o campo é desativado para uso; e o evento “Change”, que aciona quando o elemento tem sua saída alterada.

Para o projeto supervisorio, será utilizado um elemento de entrada e saída de texto no template 1. Esse elemento de entrada e saída de texto deve ser relacionado à lista de texto criada na seção 5.7.1.1. Além disso, deverá ser criada uma TAG de referência para este elemento e será necessário criar eventos do tipo “Loaded” para cada tela e determinar a ação “Set tag” para alterar o valor da TAG de referência para o valor de indexação da lista de texto referente à tela. Desta forma, cada tela irá indicar um valor de indexação e acessará a lista de dados para que o nome da tela apareça no elemento.

A seguir é necessário adicionar a data e a hora do no canto superior direito da tela.

### **5.7.1.3 Data e hora**

O elemento data e hora serve para mostrar na tela a data e a hora do computador ou o valor de uma variável de tempo. Este elemento aceita também uma entrada do tipo tempo pelo usuário, caso tenha sido escolhida uma TAG de referência.

As propriedades do elemento data e hora são divididas em seis campos: “General”, “Appearance”, “Layout”, “Text format”, “Miscellaneous” e “Security”.

“General” permite determinar se o elemento utilizará a hora do computador ou

de uma TAG de referência, se irá mostrar a hora ou a data e se será um elemento de entrada, saída ou ambos; já no campo “Appearance”, é possível editar a cor do fundo, se o fundo é sólido ou transparente, a cor do texto, o estilo do texto e a cor da borda; em “Layout”, é possível determinar a posição do elemento, o tamanho do elemento e as margens que o texto deve seguir em relação ao elemento; “Text format” permite determinar a fonte do texto e o alinhamento em relação à borda do elemento; “Miscellaneous” permite determinar o nome do elemento, a camada em que ele se encontrará na tela e descrever com palavras, dentro do software, as características e funcionalidades do elemento; e, em “Security”, é possível determinar quem tem autorização para utilizar este elemento.

Finalizando a edição do template 1, foi utilizado um elemento de data e hora posicionado no canto superior direito apenas para mostrar a data e a hora do computador.

Após garantir que o template 1 está corretamente configurado, retorna-se para a edição da tela inicial do projeto supervisorio. Foram utilizados apenas três elementos, duas caixas de texto e um botão.

O botão foi posicionado aproximadamente no centro da tela com o evento “Release” que aciona a ação “ActivateScreen” para alternar para a tela “Visão geral”. Uma caixa de texto, com uma borda grossa, foi posicionada acima do botão com o texto “Bem Vindo!” e a outra caixa de texto, sem borda, foi posicionada abaixo do botão com o texto “Pressione a peça para iniciar.”. A tela inicial concluída encontra-se na figura 5.23

A próxima tela a ser criada é a tela de visão geral.

### 5.7.2 “Visão geral”

Na tela “Visão geral”, serão apresentadas pequenas representações das três principais partes do sistema físico, a esteira, o braço e a prateleira. Cada uma destas partes será representada por uma imagem que servirá de botão para uma representação mais detalhada, além de anunciar se esta parte do sistema está sendo utilizada, para o caso da esteira, os quatro estados do braço, sem considerar o sistema de sucção, ou onde estão posicionadas as peças, para o caso da prateleira.

Além destas representações, a tela de visão geral contará com os botões funcionais posicionados na parte inferior da tela e um botão de retornar para a tela anterior, como as outras telas de representação do sistema. Como mencionado anteriormente, os botões funcionais são o botão “Start”, que inicia o transporte de peça, o botão “Stop”, que pausa o transporte de peça, o botão “Quit”, que inicia a retirada de peça, e o interruptor “IHM/Painel”, que determina qual painel, o painel IHM ou o painel da planta mecatrônica, comandará o sistema físico.



Figura 5.23: Edição de telas – Tela inicial.

Para garantir que os botões funcionais sejam alocados na tela de visão geral, nas propriedades gerais da tela, no campo “Template”, selecione o template 2, criado na seção 5.6.

A inserção dos elementos na tela de visão geral será feita por camadas, indo da menor para a maior. Para percorrer as camadas, vá às propriedades da tela e selecione a camada ativa desejada.

Começando pela camada zero, foram inseridas as imagens referentes à torre, ao pistão, à esteira ligada, ao braço levantado e esticado e à prateleira com peças. Estas imagens ficarão fixas e as demais imagens irão aparecer dependendo do estado do sistema físico.

Seguindo para a camada um, foram posicionadas as imagens da esteira ligada, do braço abaixado e esticado e as imagens cinza que cobrirão as peças da prateleira. Estas imagens cinza irão desaparecer quando uma peça for posicionada na prateleira.

Na camada dois, foram alocadas as imagens que representam o braço retraído, tanto levantado quanto abaixado.

Por fim, na camada três, foram posicionados os botões de transferência de tela. Utilize os mesmos botões que foram criados com a inicialização do projeto, altere

seus tamanhos e posições para cobrirem as imagens e altere a aparência dos botões para transparentes sem borda 3D. A tela concluída pode ser visualizada na imagem 5.24.

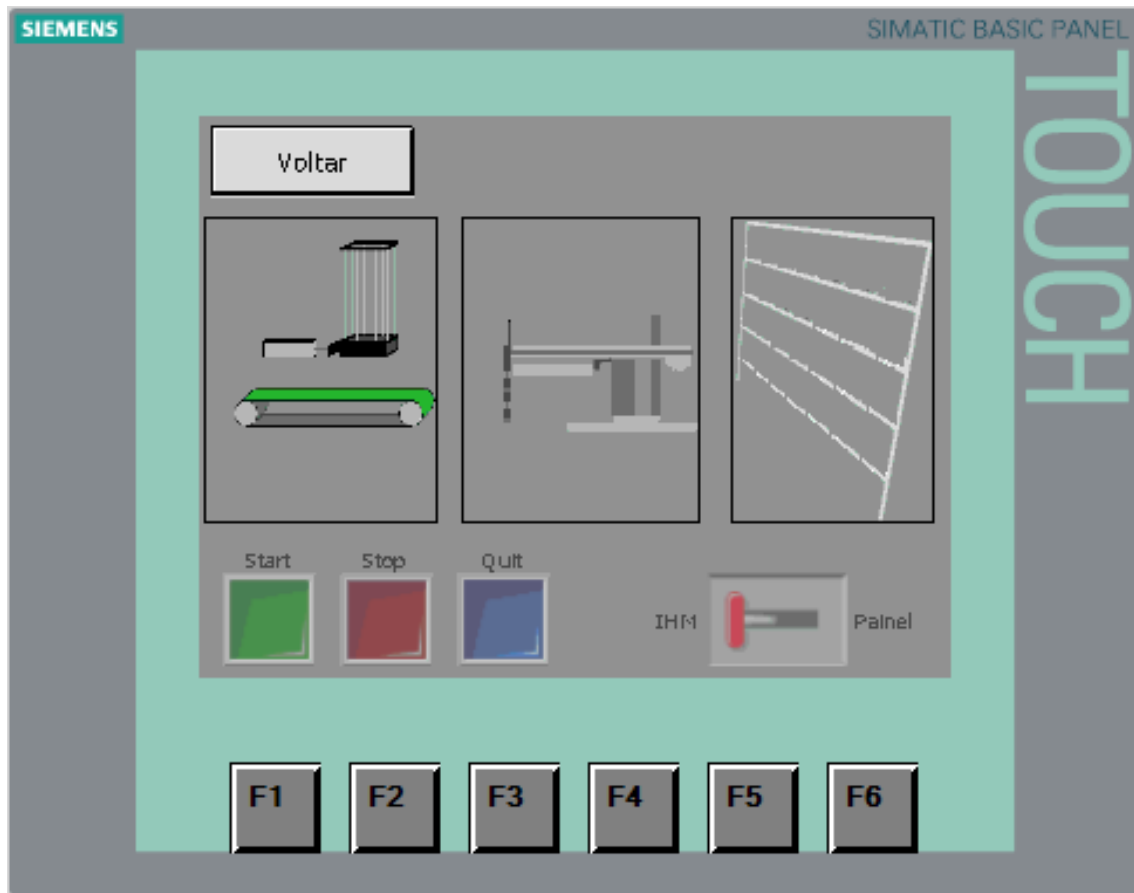


Figura 5.24: Edição de telas – Tela de visão geral.

A seguir, será descrito como foi criada a tela da esteira, que representa de forma mais detalhada os estados da esteira e do pistão.

### 5.7.3 “Tela esteira”

A tela da esteira, como dito previamente, servirá como uma representação mais detalhada dos eventos que ocorrem na esteira e no pistão. Nela, será facilmente reconhecido quando o pistão é ativado para empurrar a peça, se a esteira encontra-se ligada e onde a peça se encontra, caso esteja na esteira.

A alocação das imagens na tela da esteira será aproximadamente a mesma que a feita para a tela de visão geral para a parte da esteira e do pistão. As únicas diferenças são que terá um botão para retornar à tela de visão geral e serão incluídas as peças em cima da esteira.

Por tanto, seguindo a mesma ordem utilizada na tela de visão geral e utilizando

as imagens em uma escala maior, para melhor preencher a tela, pode-se completar grande parte da edição da tela da esteira. Para concluir, posicione o botão de retornar à tela de visão geral no canto superior esquerdo e adicione as peças em cima da esteira, na segunda camada. Ao concluir, a tela deverá ficar semelhante à imagem 5.25.

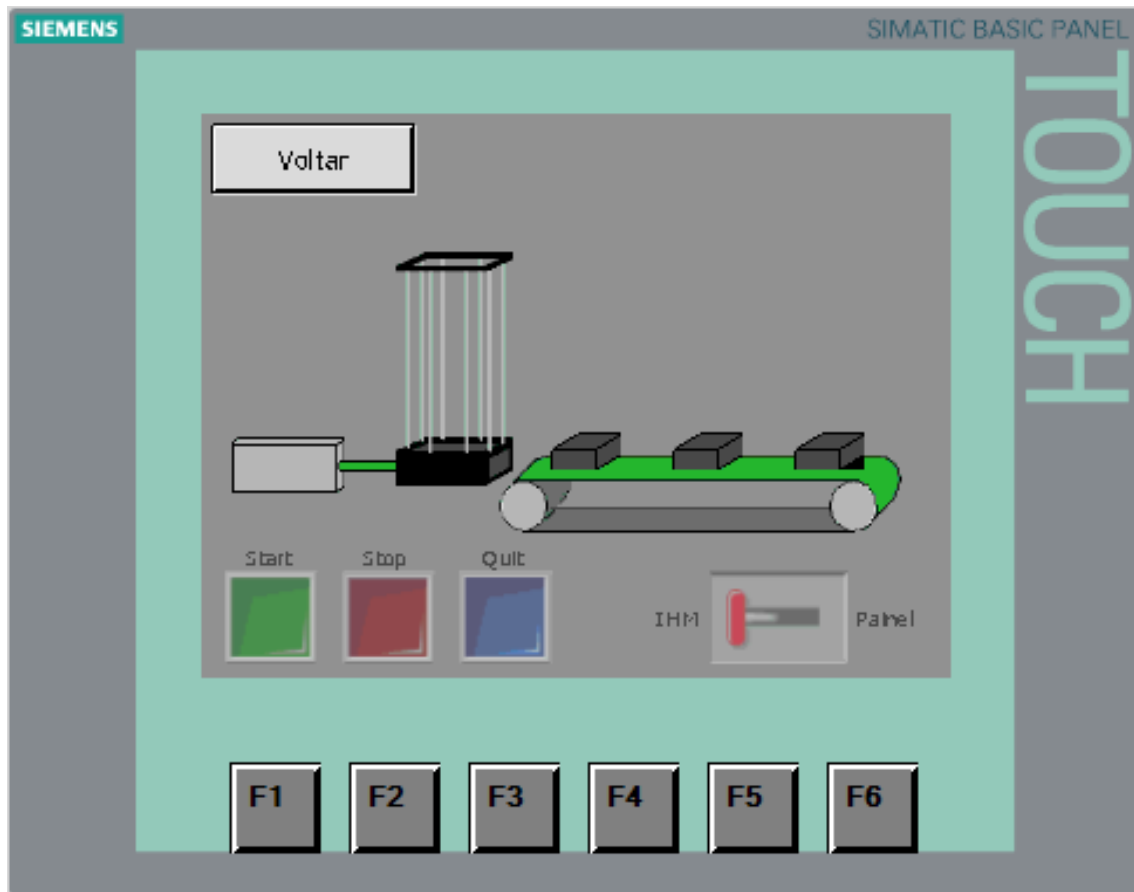


Figura 5.25: Edição de telas – Tela da esteira.

A próxima tela a ser editada é a tela do braço, que representará de forma mais detalhada os movimentos do braço giratório.

#### 5.7.4 “Tela braço”

A tela do braço, exatamente como a tela da esteira, servirá como uma representação mais detalhada dos estados de uma parte do sistema, mas, neste caso, será da parte referente ao braço giratório que transporta as peças da esteira para a prateleira ou o contrário. Nesta tela, é possível observar qual a posição do braço quando estiver girando, se o braço está levantado ou abaixado, se está esticado ou retraído e se o braço está utilizando o sistema de sucção para carregar uma peça.

O posicionamento das imagens também será feito por camadas e será mais completo que a representação feita na tela de visão geral.

Para representar a posição que o braço se encontra quando estiver girando, serão utilizadas imagens formando “três quartos de pizza”, onde o quadrante que será retirado da “pizza” será o primeiro quadrante. O segundo quadrante representará a calibragem do braço, ou seja, o momento em que o braço roda a partir da esteira até o sensor indutivo e retorna para a posição inicial. Esta representação terá três estados: braço calibrando e braço não calibrando. Os terceiro e quarto quadrantes representarão o transporte de peça entre a esteira e o elevador. A representação do transporte terá seis estados: braço parado, braço calibrando e quatro posições correspondentes ao instante da rotação.

A representação do giro do braço foi feita com imagens no formato de “pizza” para os três quadrantes utilizados. Quando o braço encontra-se parado, as imagens ficam cinza, e, para que as demais imagens as sobreponham, estas devem ser colocadas na camada zero. Na camada um, serão posicionadas a imagem do quadrante dois do braço calibrando e as imagens dos quadrantes três e quatro do braço movendo, que serão representadas por imagens na cor verde. Por fim, na camada dois, serão utilizadas as imagens que determinam a posição do braço durante o transporte de peça. A posição do braço será representada por uma fatia na cor vermelha.

A outra representação desta tela terá as mesmas imagens que a representação do braço na tela de visão geral, com a diferença de ter movimento com oito estados diferentes para esta representação. Os estados são uma combinação das possibilidades de evento, descritas anteriormente: se o braço está levantado ou abaixado, se está esticado ou retraído e se o braço está utilizando o sistema de sucção para carregar uma peça.

Os estados do braço serão representados por dez imagens que serão alocadas em camadas diferentes. Na camada zero, será colocada a imagem do braço levantado e esticado sem o sistema de sucção estar atuando, que será o estado base, e os outros estados serão representados por imagens sobrepostas a esta. Para a camada um, será colocada a imagem que representa o braço abaixado e estendido. Serão posicionadas duas imagens na camada dois, que representarão o braço retraído. Uma imagem para sobrepor a representação do braço levantado e outra imagem para sobrepor a representação do braço abaixado. Por fim, na camada três, serão posicionadas quatro imagens que representarão o sistema de sucção do braço. As imagens irão variar dependendo dos quatro estados do braço.

Após alocar todas as imagens em suas camadas, insira um botão de retorno para a tela de visão geral para concluir a edição da tela do braço. A tela concluída pode ser verificada na imagem 5.26.

Para finalizar a edição das telas, será editada a tela da prateleira.

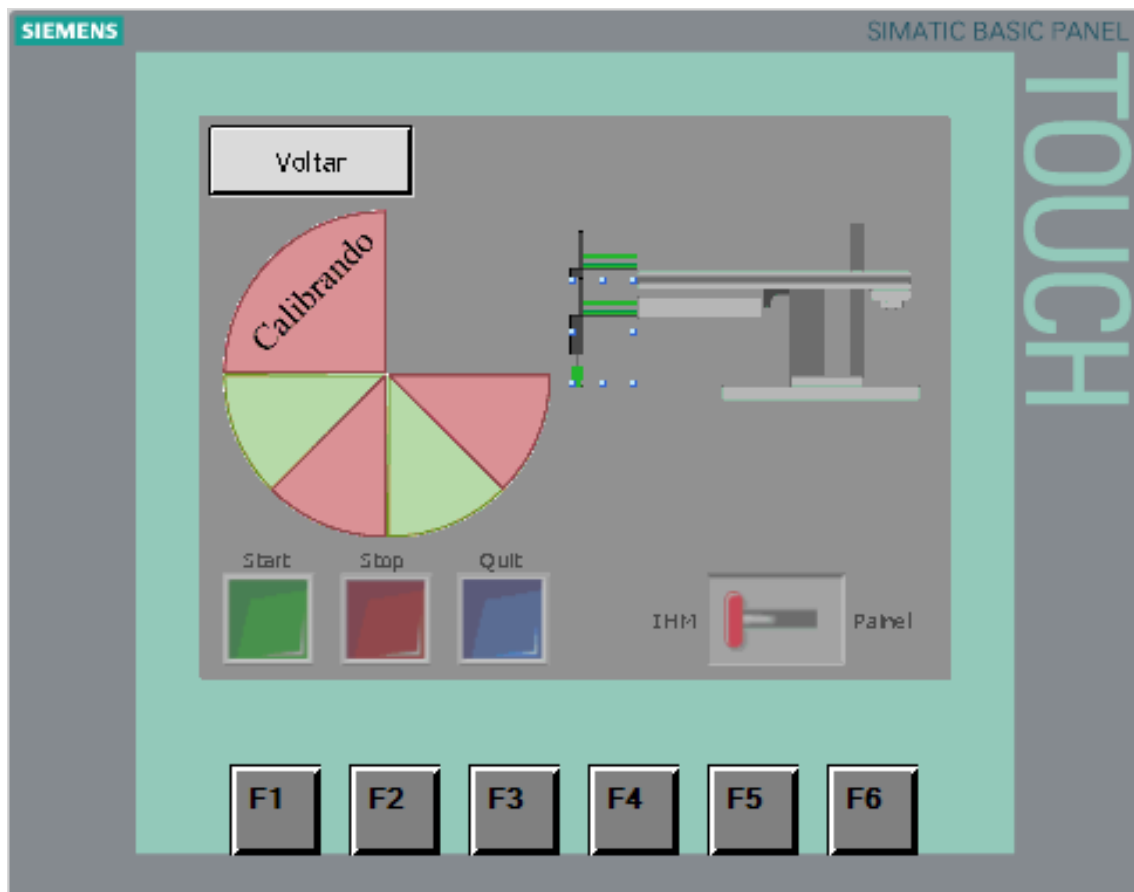


Figura 5.26: Edição de telas – Tela do braço.

### 5.7.5 “Tela prateleira”

A tela da prateleira será utilizada para ver em quais posições da prateleira encontram-se peças e para escolher qual peça retirar caso o usuário pressione o botão “Quit”.

Esta tela terá imagens e botões também distribuídos em camadas para que as peças apareçam à frente da prateleira e os botões sobreponham as peças. Caso as peças ficassem à frente dos botões, eles não seriam pressionáveis.

Na camada zero, será colocada a imagem da prateleira vazia para servir de base para as demais imagens e botões. Já na camada um, vinte e oito imagens de botões serão posicionados em cada espaço da prateleira para que todos os espaços sejam devidamente representados. Por fim, a camada dois dará lugar aos vinte e oito botões que representam cada posição da prateleira e devem preencher por completo cada espaço. Os botões devem ser transparentes para não esconderem as peças.

Esta tela será concluída ao posicionar o botão de retorno para a tela de visão geral no canto superior esquerdo. A tela completa pode ser visualizada na imagem 5.27.

Com isso, conclui-se o passo três, de edição de telas, faltando apenas adicionar

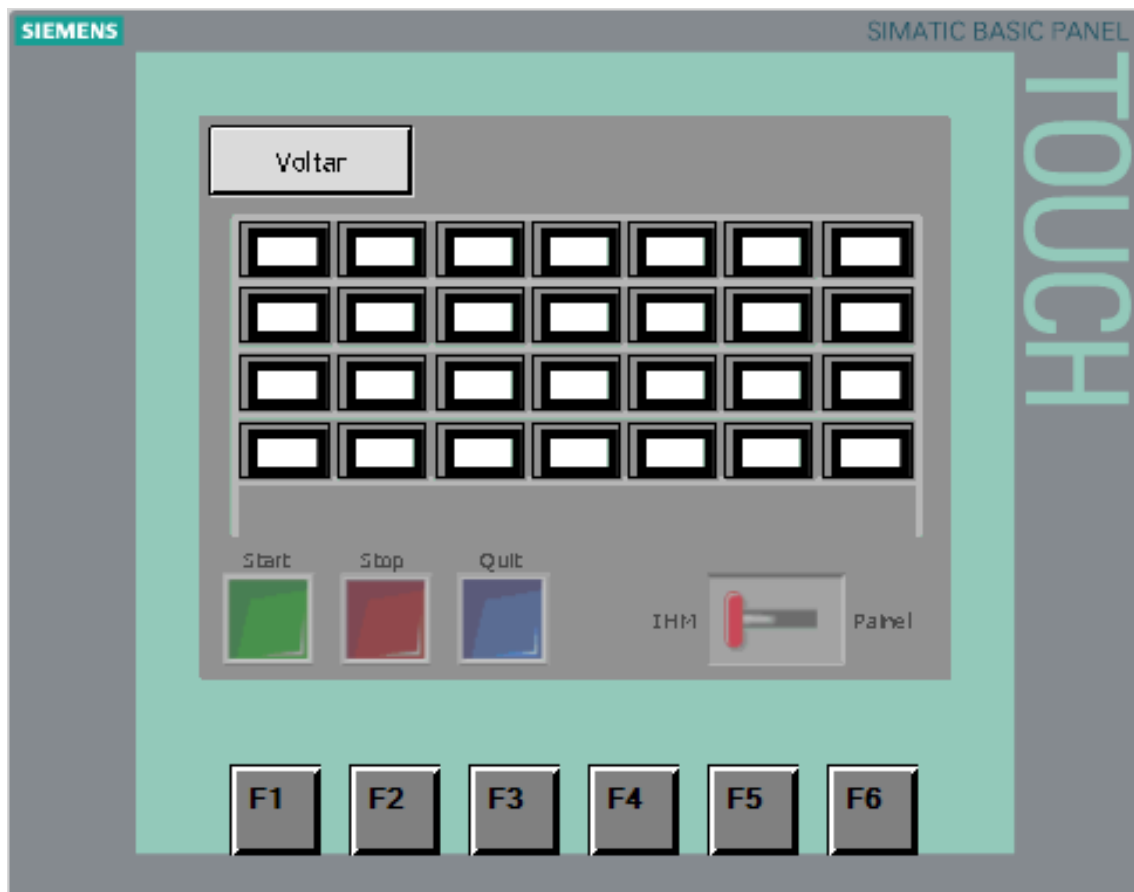


Figura 5.27: Edição de telas – Tela da prateleira.

as funcionalidades de cada elemento adicionado às telas.

## 5.8 Passo 4: Criação das funcionalidades dos elementos

O passo quatro é o último passo para concluir o projeto supervisório baseado em um projeto de automação previamente criado. Este passo consiste em vincular as representações criadas nas telas da IHM aos estados da planta física e programar as funcionalidades dos elementos.

Os vínculos entre as representações e os estados do sistema físico serão dados pelo CLP, mais precisamente pelas TAGs do CLP. Para isso, é aconselhável criar uma tabela de TAGs no CLP exclusiva para a IHM para evitar possíveis enganos e não confundir as TAGs. Vá em “PLC\_1/PLC tags/Add new tag table” na aba “Project Tree”, à esquerda.

Os vínculos realizados serão descritos separadamente por tela e para o template 2.

### 5.8.1 Botões funcionais

Os botões funcionais encontram-se no template 2 e existem em todas as telas que representam o sistema. Eles devem iniciar o transporte de peça, pausar o transporte de peça, iniciar a retirada de peça ou selecionar qual painel que irá controlar o sistema.

Cada botão ou interruptor deve ter uma TAG associada criada na tabela de TAGs do CLP. Após criar as quatro TAGs, é necessário vincular o elemento à TAG correspondente, alterando, nas propriedades gerais, a TAG do elemento para a TAG criada no CLP. Esta TAG será replicada automaticamente para a lista de TAGs da IHM.

Na lista de TAGs da IHM, é possível e aconselhável alterar o tempo de leitura de cada TAG para o menor possível, para que evite atrasos na utilização das funcionalidades da IHM e na representação dos eventos do sistema.

Apenas criar a TAG não é o suficiente para o botão funcionar. É necessário inseri-lo na lógica Ladder do CLP. As lógicas Ladder utilizadas não serão descritas neste projeto, serão apenas sugeridas superficialmente como possíveis resoluções.

Para o caso dos botões “Start”, “Stop” e “Quit”, insira contatos normalmente abertos em paralelo com o contato de reconhecimento dos botões físicos no módulo de eventos. Já para o interruptor, foram utilizados contatos normalmente abertos antes dos contatos de reconhecimento dos botões físicos e contatos normalmente fechados antes dos contatos criados para os botões. Além de condicionar o reconhecimento dos botões, fez-se necessário condicionar o LED de iluminação de cada botão do painel físico e o aparecimento dos botões no painel IHM apenas quando puderem ser pressionados.

Por fim, após configurar corretamente a lógica Ladder, é necessário criar um vínculo entre o aparecimento dos botões no painel IHM e a TAG criada para representar este aparecimento. Para isso, crie uma animação do tipo “Visibility” vinculada à TAG do CLP.

A primeira tela a ter suas funcionalidades e seus vínculos configurados será a “Tela esteira”.

### 5.8.2 Tela da esteira

Na tela da esteira, será necessário ativar a representação do pistão e do movimento da peça na esteira. Estas representações serão baseadas nos sensores presentes na esteira e no pistão.

Para o pistão, crie uma TAG representativa que acione de forma complementar ao sensor de presença do pistão retraído e vincule uma animação do tipo “Visibility” com a imagem do pistão. Lembrando que é aconselhável reduzir o tempo de leitura

das TAGs vinculadas a elementos da IHM.

A posição da peça na esteira pode ser determinada utilizando os sensores de presença existentes no pistão e ao longo da esteira. Ao todo, são três sensores que podem ser utilizados: sensor de presença do pistão esticado, sensor de presença do meio da esteira e sensor de presença final. Criando uma lógica Ladder simples, é possível determinar TAGs de saída para serem vinculadas a três posições básicas da esteira (as três posições nas quais as peças foram posicionadas durante a edição da tela).

A lógica Ladder também deve ser pensada para o momento da retirada da peça, onde o sensor de presença do pistão esticado não poderá mais ser utilizado. Para contornar, é possível utilizar as transições do sistema para determinar o fim do trajeto da peça pela esteira.

A próxima tela a ter seus vínculos e funcionalidades configurados é a “Tela braço”.

### 5.8.3 Tela do braço

A representação do braço será dividida em duas partes, uma que determinar a posição do braço durante o giro e outra que determina a condição do braço com relação a seus pistões internos e seu sistema de sucção. A representação do giro será dada por imagens semelhantes a uma pizza e a representação da condição do braço será dada por uma réplica dotada de movimento, como mencionado na seção 5.7.4. É aconselhável que todas as TAGs criadas para a representação do braço tenham seus tempos de varredura reduzidos na IHM para que a representação ocorra em tempo real.

Começando pela representação do giro, será utilizada uma TAG inteira que determinará quais imagens aparecerão em cada instante dependendo do valor adotado pelo contador de alta velocidade do CLP. Dois pontos são importantes para auxiliar na hora de determinar a dependência da TAG inteira ao valor do contador de alta velocidade: os valores para o giro no sentido horário são negativos, sendo necessário calcular o módulo do valor antes determinar o valor da TAG inteira; e o valor da TAG, algumas vezes, é mantido após a finalização de um giro, atrapalhando a contagem seguinte e causando falha na representação. É aconselhável que o valor seja zerado nestas situações.

Com a determinação do valor da TAG inteira vinculado ao valor do contador de alta velocidade, associe o aparecimento das imagens com uma faixa de valores da TAG inteira. Para isso, crie uma animação do tipo “Visibility” vinculada à respectiva faixa de valores para cada imagem da representação.

As únicas imagens que não terão vínculo de visibilidade com a TAG inteira serão as imagens que determinam que o braço está parado e as imagens que determinam se

o braço está girando sem especificar a posição, que são utilizadas para complementar o desenho. As imagens que determinam que o braço está parado não terão nenhum tipo de animação. Já as imagens que determinam se o braço encontra-se girando terão um vínculo com uma nova TAG booleana que deverá ser criada e adicionada à lógica Ladder para ativá-la quando o braço estiver girando.

A representação da condição do braço será vinculada a quatro TAGs, duas inteiras e duas booleanas. Uma TAG booleana será vinculada ao estado vertical do braço e outra TAG booleana será vinculada ao estado horizontal do braço, sendo utilizada apenas de forma auxiliar. As TAGs inteiras serão vinculadas ao sistema de sucção e ao estado horizontal do braço. A TAG inteira vinculada ao sistema de sucção do braço assumirá o valor zero para os instantes em que o sistema de sucção encontrar-se desligado e assumirá valores pré-determinados para os quatro possíveis estados do braço. Os valores pré-determinados serão os mesmos valores utilizados para a TAG inteira vinculada ao estado horizontal do braço.

Os valores da TAG inteira vinculada ao estado horizontal do braço podem ser determinados com a utilização de TAGs auxiliares vinculadas às ações do braço. Para criar tais vínculos, utilize bobinas SET e RESET em paralelo com as ações do braço e em seguida crie uma lógica Ladder para determinar o valor da TAG, baseado nas condições criadas. Para a representação do sistema de sucção, utilize um contato normalmente aberto em série com uma bobina, que iguale os valores das duas TAGs inteiras, e um contato normalmente fechado em série com uma bobina que determine o valor zero para a TAG inteira do sistema de sucção.

Ao concluir todas as lógicas Ladder, é necessário vincular as imagens aos valores das TAGs utilizando uma animação do tipo “Visibility” para cada imagem e determinar seu respectivo valor com relação à TAG.

Após a edição da tela do braço, será editada a tela da prateleira, deixando por último a tela de visão geral, por utilizar basicamente as mesmas lógicas das telas com representações mais detalhadas.

#### **5.8.4 Tela da prateleira**

A tela da prateleira não será utilizada apenas para representar quantas e quais peças foram posicionadas na prateleira. Será utilizada, também, para a escolha da peça que o usuário deseja retirar.

A adaptação da função de retirada de peça demanda um estudo prévio da lógica Ladder do sistema, apêndice A. O sistema automatizado utilizado neste projeto possui dois lugares para cada posição da prateleira, que determinam se há peça ou não nesta posição (como exemplo, o lugar “P10+” determina se existe uma peça na posição 9). Para a representação da ocupação da prateleira, vincule o aparecimento

da imagem da peça à TAG desse lugar, com a utilização de uma animação do tipo “Visibility”.

Como a função de retirada de peça da IHM será acionada de forma diferente do painel do CLP, viu-se necessário alterar as transições entre os lugares da lógica Ladder quando o controle estiver sendo executado pela IHM, para que a retirada de peça seja acionada apenas com um toque na peça desejada. É necessário garantir que as transições retornem ao estado normal ao alternar quem acionará os comandos durante a escolha de peças.

As transições do controle da IHM devem bloquear a escolha de peças pelos interruptores do painel do CLP e desabilitar a necessidade de pressionar o botão “Quit” para executar a retirada de peça. Utilizando o lugar “P33” do sistema automatizado como referência, é possível habilitar os botões de escolha de retirada de peça, da tela da prateleira, dependendo também da presença de peça em cada posição.

A habilitação do botão pode ser definida utilizando uma TAG e criando a lógica descrita acima. Essas TAGs serão as referências para uma animação do tipo “Visibility” para cada botão.

Além das animações, os botões devem conter eventos que ativem os lugares com os nomes “P##\*” para que iniciem a retirada da peça selecionada. Lembrando também que é necessário ativar algumas transições que podem ter sido desativadas e que são importantes para o início da retirada de peça. Estas transições devem ser ativadas também através de eventos utilizando a ação “Set tag”.

Para finalizar a edição de telas, será editada a tela “Visão geral”.

### 5.8.5 Tela de visão geral

A tela de visão geral foi deixada por último por utilizar simplificações dos vínculos criados para as demais telas.

Os vínculos da esteira ligada e o vínculo da representação da condição do braço sem considerar o sistema de sucção devem ser copiados para as representações presentes na tela de visão geral.

O único vínculo que difere é o criado para a visualização das peças que estão presentes na prateleira. Como, na tela de visão geral, as imagens que encontram-se nas camadas superiores são as imagens responsáveis por esconder as peças, os vínculos devem também ser uma animação do tipo “Visibility” para as mesmas TAGs utilizadas para cada peça em cada posição da prateleira, mas considerando que as imagens devem ser invisíveis quando as imagens da tela da prateleira encontrarem-se visíveis. Desta forma, as peças da prateleira irão sumir quando não houver peça na posição correspondente.

A criação dos vínculos de todas as telas conclui o projeto supervisorio criado no

software TIA Portal da SIEMENS, faltando apenas implementar o projeto.

## 5.9 Passo 5: Implementação do projeto

Após criar e salvar o projeto, é necessário transferir o projeto para a IHM e para o CLP, testar os possíveis erros de programação e corrigi-los. Ao comandar a transferência, o próprio programa compila o projeto e configura os dispositivos.

Para comandar a transferência, selecione ambos os dispositivos e clique em “Download to device”, como mostra a figura 5.28

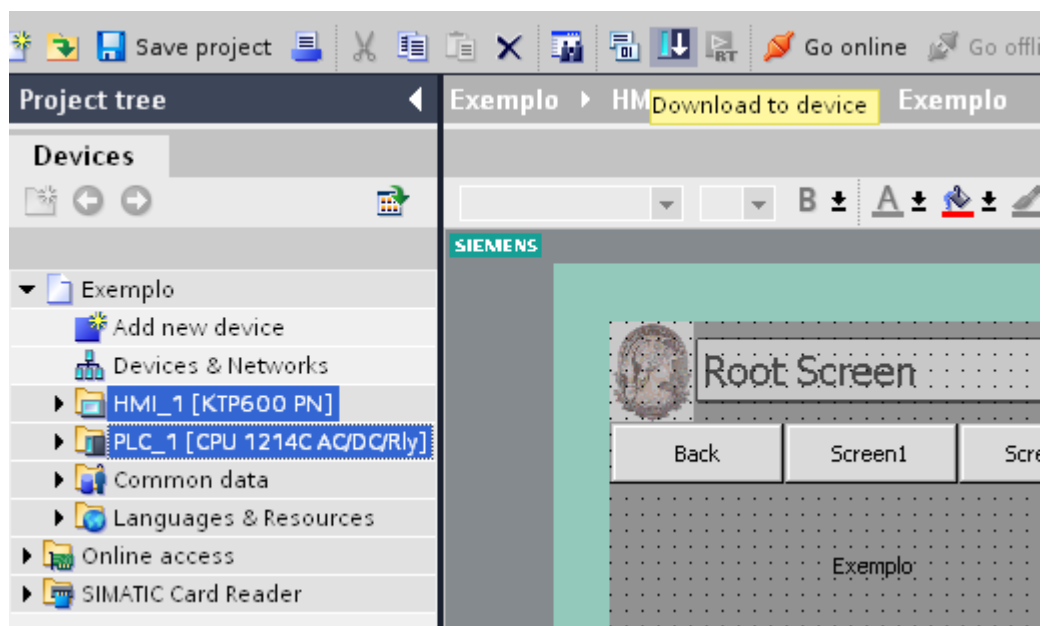


Figura 5.28: Seleção e início de transferência do projeto para os dispositivos.

É necessário que o CLP esteja parado enquanto a transferência é realizada. Portanto, uma mensagem aparecerá caso o CLP não esteja parado. Confirme a para do CLP, como mostra a figura 5.29

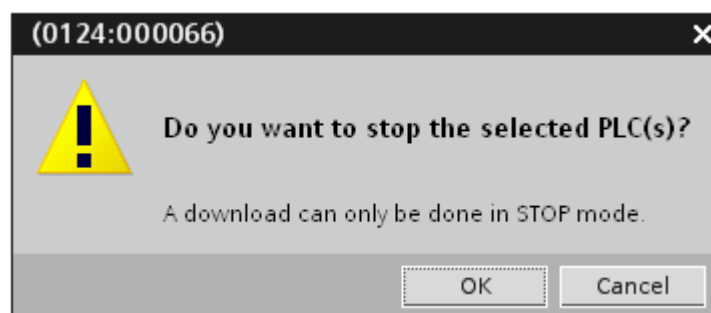


Figura 5.29: Mensagem de parada do CLP para a transferência.

Após configurar as conexões, caso necessário, uma tela de confirmação aparecerá. É necessário que selecione a opção “Overwrite all” para que o sistema que encontra-

se na IHM seja sobre-escrito. Caso não tenha nenhum sistema, a IHM se encontrará em uma tela com três botões: “Transfer”, “Start” e “Control Panel”. Pressione o botão “Transfer” para permitir a transferência do projeto.

Após a transferência, selecione a opção “Start all” e pressione o “Finish” para concluir. Teste o sistema por completo, corrija os possíveis erros e repita o processo até sua conclusão. Com o projeto supervisorio concluído, será adicionado o módulo manual.

# Capítulo 6

## Implementação do Módulo Manual

Neste capítulo será feita uma alteração no projeto para que o projeto supervisor conte com uma função de controle manual da IHM sem perder as funções de supervisão dos eventos do sistema automático. As alterações serão realizadas baseadas no projeto do Capítulo 5.

Seguindo o mesmo padrão do Capítulo 5, este tutorial será desenvolvido passo a passo e, a cada passo que utilize uma função que ainda não foi explicada anteriormente, antes de demonstrar como esta função foi aplicada ao projeto supervisor, será descrita a função de forma genérica.

Antes de iniciar a alteração do projeto, serão descritas as funções planejadas para o módulo manual.

### 6.1 Descrição do projeto manual

Para realizar as alterações no projeto, é necessário planejar suas alterações previamente para que a edição seja feita de forma sistêmica. Por isso, será descrito, inicialmente, como as alterações foram planejadas para esta função de controle manual.

Primeiramente, alternar entre as funções de controle manual e automática durante um processo ou um estado do sistema pode ser prejudicial aos componentes do sistema, visto que qualquer erro durante o projeto pode causar danos às peças se os controles forem alternados no momento errado. Para contornar estas situações foi estipulado que só será possível alternar do controle automático para o controle manual quando o sistema encontrar-se pausado, ou seja, encontrar-se na função “Stop”. Da mesma forma, para alternar do controle manual para o controle automático é necessário que o sistema retorne para o estado determinado pela função “Stop” antes de alternar o controle. Será necessário retirar do sistema as peças que não encontrem-se nem na torre de peças e nem na prateleira, despejando-as pela esteira, e recalibrar o sistema.

O controle manual permitirá apenas comandos dados pela IHM, já que o painel da planta mecatrônica não possui botões suficientes. Os comandos serão dados por botões posicionados nas telas dos sistema e irão comandar um ciclo fechado de ações, para restringir as possibilidades do usuário e minimizar as falhas prejudiciais ao sistema. Será possível acionar apenas um comando de cada vez. Na tentativa de acionar um segundo comando ao mesmo tempo, o sistema acionará um alarme com um aviso.

Serão criados nove comandos para o sistema, sendo eles: acionar pistão; ligar esteira para a esquerda; ligar esteira para a direita; calibrar o braço; girar o braço; braço pega peça da esteira; braço entrega peça para o elevador; braço pega peça do elevador; e elevador retira peça da prateleira. Além dos comandos, será necessário criar uma lógica para retornar ao modo automático do sistema.

### **6.1.1 Comando de acionar pistão**

O comando de acionar pistão servirá para retirar uma peça da torre de peças e colocar na esteira.

Ao acionar o comando, o pistão da torre de peças irá empurrar uma peça da torre para a esteira e retornará ao estado retraído. Este comando será restringido se a torre de peças possuir menos que duas peças ou se existir uma peça na esteira posicionada à frente do pistão. Nesses casos, será acionado um alarme para avisar ao usuário sobre a falha.

### **6.1.2 Comandos de ligar esteira para a esquerda e de ligar esteira para a direita**

Os comandos de ligar a esteira para a esquerda ou para a direita irão acionar a esteira por um determinado tempo, de forma que descarte as peças que nela se encontrem ou leve a peça até o final da esteira, onde estará posicionada para interagir com o braço.

No momento em que o usuário aciona o comando de ligar a esteira para a esquerda, a esteira é ligada para a esquerda por 12,5 segundos, tempo suficiente para descartar as peças que encontram-se na esteira. E no momento em que o usuário aciona o comando de ligar a esteira para a direita, a esteira só irá movimentar-se caso não haja peça no final da esteira e só irá parar quando a peça chegar à posição desejada ou transcorrerem 12,5 segundos desde o acionamento. Caso a esteira seja acionada para a direita e haja uma peça no final da esteira, será acionado um alarme que avisará ao usuário sobre a impossibilidade.

### **6.1.3 Comando de calibrar o braço**

Acionar o comando de calibrar o braço fará com que o braço calibre sua posição em relação ao seu eixo giratório. O braço será acionado no sentido horário e retornará, no sentido anti-horário, para a direção da esteira.

O comando, quando acionado, gira o braço no sentido horário até disparar o sensor indutivo e retorna no sentido anti-horário, por 808 pulsos do contador de alta velocidade, parando na posição correta para interagir com a esteira.

### **6.1.4 Comando de girar o braço**

“Girar o braço” significa alternar sua posição entre a posição em que interage com a esteira e um giro de  $180^\circ$  em seu eixo a partir desta posição, ou seja, a posição em que interage com o elevador.

O acionamento deste comando depende do estado em que se encontra o braço. Se o braço encontrar-se voltado para a esteira, ao acionar o comando, o braço irá girar no sentido anti-horário e parar a  $180^\circ$  da posição anterior. Caso o braço encontre-se voltado para a prateleira, ao acionar o comando, o braço irá girar no sentido horário, como se estivesse calibrando, até disparar o sensor indutivo, e retornará para a posição de interação com a esteira.

### **6.1.5 Comando de o braço pegar peça da esteira**

O comando de o braço pegar peça da esteira é utilizado tanto para retirar a peça da esteira, com o braço, quanto para devolver a peça, que se encontra no braço, para a esteira. A ação será determinada pela localização da peça, se ela está sendo carregada pelo sistema de sucção do braço ou se está localizada ao final da esteira.

Considerando a peça localizada ao final da esteira como sendo um estado inicial, ao acionar o comando do braço pegar a peça da esteira, o braço irá estender e levantar, para posicionar-se acima da peça, descer e ativar o sistema de sucção, para pegar a peça, e levantar e retraindo, finalizando o ciclo do comando. Quando o braço está carregando a peça e o final da esteira está vazio, ao acionar o comando, o braço devolve a peça à esteira, estendendo-se, descendo até a esteira, desativando o sistema de sucção e subindo enquanto retrai. Este comando só é possível quando o braço encontra-se na direção da esteira e quando tem uma, e apenas uma, peça no sistema de sucção do braço ou ao final da esteira. Caso contrário, não será possível acionar o comando e ativará um alarme alertando ao usuário quanto ao erro.

### **6.1.6 Comando de o braço entregar peça para o elevador**

O braço entregar a peça para o elevador é um ciclo de ação em que o braço, estando virado para a esteira e com peça em seu sistema de sucção, entrega peça para o elevador vazio e retorna calibrando, parando quando estiver voltado para a esteira. Após o retorno do braço, o elevador posiciona a peça no primeiro espaço vazio da prateleira e retorna ao ponto inicial.

A ação só pode ser executada caso o braço esteja voltado para a prateleira, se houver peça no sistema de sucção do braço e se o elevador encontrar-se vazio. Caso alguma dessas condições não seja atendida, a ação não poderá ser executada e soará um alarme alertando ao usuário o motivo do erro.

### **6.1.7 Comando de o braço pegar peça do elevador**

A ação “Braço pega peça do elevador” é mais simples que a ação “Braço entrega peça para o elevador”, já que não é necessário que o elevador mexa com a prateleira e nem que o braço calibre ao final da ação. Com o braço virado para a prateleira, o elevador será posicionado abaixo do braço e o braço coletará a peça, terminando o ciclo da ação quando o elevador retornar ao estado inicial.

O comando apenas poderá ser acionado caso o braço encontre-se virado para a prateleira, se não houver peça no sistema de sucção do braço e se houver peça no elevador. Caso uma destas condições não seja atendida, a ação não será executada e um alarme alertará ao usuário o motivo do erro.

### **6.1.8 Comando de o elevador retirar peça da prateleira**

Por fim, o comando do elevador retirar peça da prateleira consistirá em o elevador retirar da prateleira a peça escolhida pelo usuário a partir da IHM. A qualquer momento o usuário deve ter a possibilidade de retirar a peça desejada da prateleira, considerando apenas que deve haver peça na posição desejada, que não há outra ação sendo executada no momento e que o braço não encontra-se virado para a prateleira.

Como se trata de uma ação mais complexa, os botões de acionamento da ação, que serão os mesmo botões utilizados para o módulo automático do sistema, não estarão disponíveis para o usuário caso as condições acima não tenham sido satisfeitas. Desta forma, não há necessidade do uso de alarmes para esta ação.

### **6.1.9 Retorno do modo manual para o modo automático**

Como dito anteriormente, o modo manual só pode ser iniciado caso o modo automático se encontre parado. Com isso, para retornar ao modo automático, é necessário

que o sistema retorne para o mesmo estado.

Existem quatro possibilidades que influenciam o estado em que o sistema pode se encontrar no momento do retorno do modo manual para o modo automático: haver peça na esteira; haver peça no sistema de sucção do braço; haver peça no elevador; e o braço encontrar-se voltado para a prateleira.

Considerando isso, deve-se verificar primeiro se há peça na esteira e retirá-las. Em seguida, deve-se verificar se há peça no sistema de sucção do braço. Caso haja, será necessário calibrar o braço, para garantir que o mesmo encontra-se na direção da esteira, e entregar a peça para a esteira retirá-la do sistema. Deve-se verificar, também, se há peça no elevador. Em caso afirmativo e após a execução da correção dos testes anteriores, o braço deve pegar a peça do elevador e entregar para a esteira para que a esteira possa retirar a peça do sistema. Por fim, é necessário calibrar o braço em todas as ocasiões, para garantir que o braço encontra-se na posição correta em relação a esteira e sem nenhum ângulo de defasagem.

Conhecendo todas as ações necessárias para a criação do módulo manual do sistema, é possível iniciar o passo a passo da edição e das alterações necessárias no projeto.

## **6.2 Passo 1: Edição de telas e template**

No passo 1, serão demonstradas quais alterações devem ser feitas nas telas do sistema e template 2 para que o projeto manual seja implementado da forma desejada.

### **6.2.1 Template 2**

A adição do módulo manual ao sistema faz com que o interruptor IHM-Painel deva ser desabilitado quando o sistema encontrar-se no modo manual. Anteriormente, o interruptor IHM-Painel estava sempre habilitado, então o interruptor ficava sempre aparente e possível de ser alternado. No modo manual, será necessário que o interruptor suma, para que o usuário não possa utilizar o painel da planta mecatrônica para o acionamento manual da planta, e fique claro para o usuário que o interruptor encontra-se desabilitado, mas que existe. Portanto, deverá ser posicionada uma imagem do interruptor desabilitado na camada zero do mesmo tamanho e na mesma posição que o interruptor IHM-Painel, e o interruptor IHM-Painel deverá ser transferido para a camada um.

Para alternar entre os módulos do sistema, manual e automático, deve-se adicionar um interruptor da mesma forma que foi adicionado o interruptor IHM-Painel. Na camada zero, devem ser adicionados os textos “Manual” e “Automático”, para determinar qual estado do interruptor representa cada módulo, e a imagem do interrup-

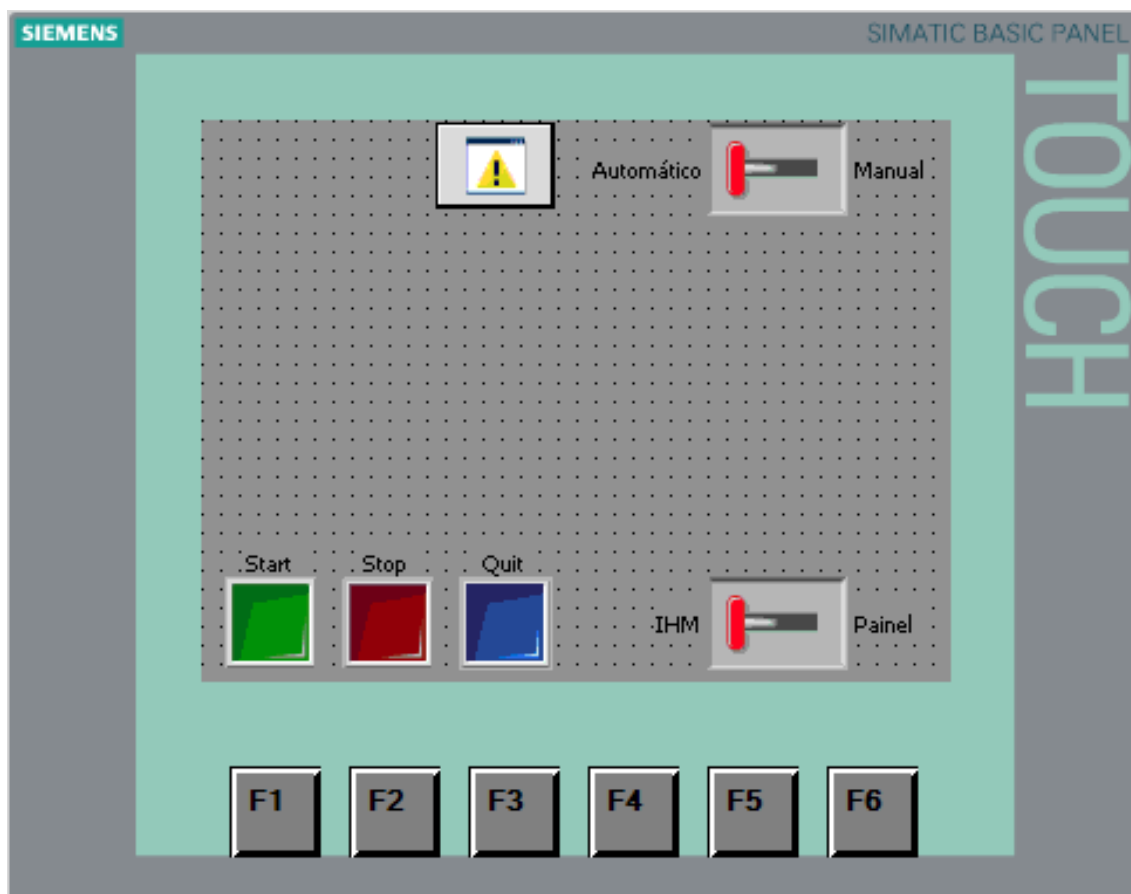


Figura 6.1: Módulo manual – Template 2 concluído.

tor desativado. Na camada um, deve-se adicionar o interruptor Manual-Automático, sendo do mesmo tamanho e ocupando a mesma posição que a imagem de interruptor desativado.

Por fim, deve-se adicionar um botão que indique ao usuário que um alarme foi acionado e que, ao clicar, mostre a descrição dos alarmes acionados. Este botão pode ser copiado da tela inicial, onde já se encontra o botão “Unacknowledged alarms”, para facilitar a configuração da funcionalidade do botão. Caso este tutorial não tenha sido seguido à risca e este botão não encontre-se na tela inicial, na seção 6.4.4, será descrito como configurar a funcionalidade do botão, juntamente com a sua função de visibilidade. A aparência final do template 2 encontra-se na figura 6.1.

Após editar os botões funcionais, no template 2, é necessário editar as telas, começando pela tela da esteira.

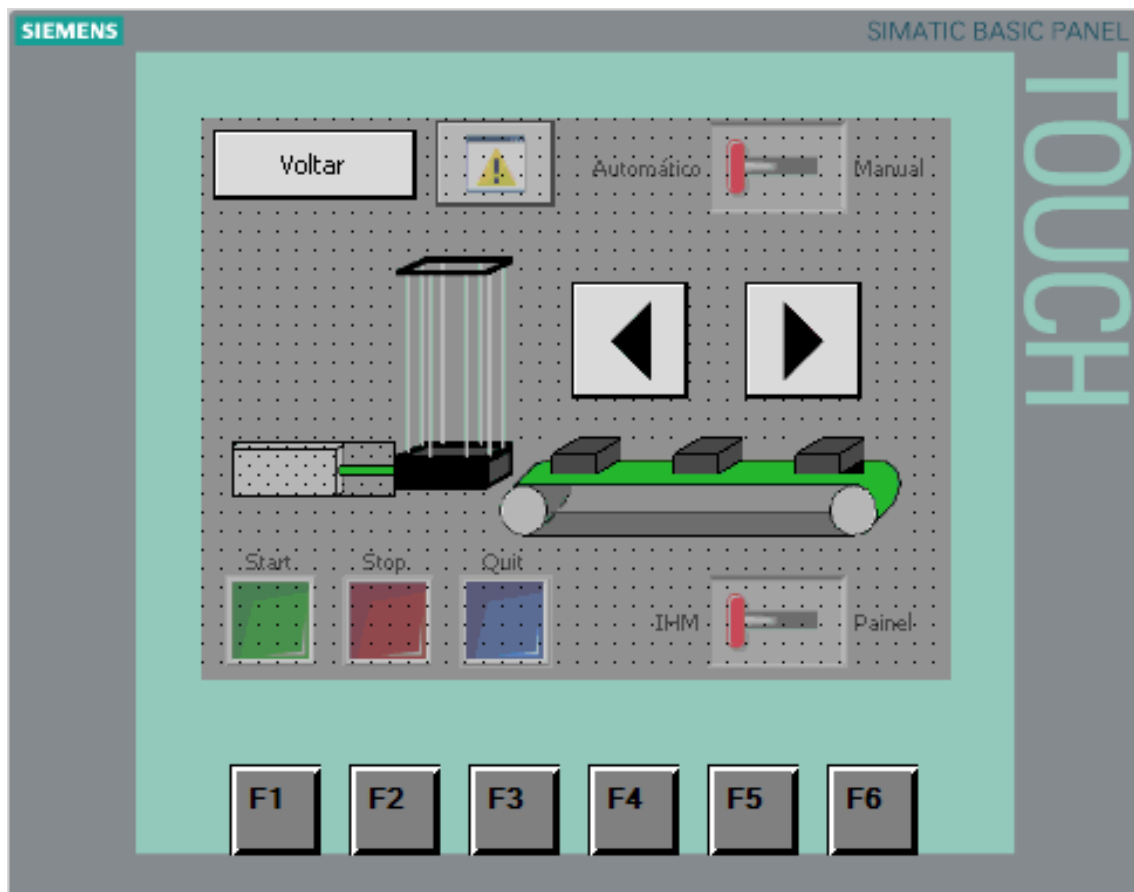


Figura 6.2: Módulo manual – Tela da esteira.

### 6.2.2 Tela da esteira

As ações comandadas pela tela da esteira serão as ações de acionar pistão e ligar esteira para o sentido desejado. Para cada ação, será adicionado um botão. O botão de acionamento do pistão será posicionado sobrepondo a representação do pistão na camada dois e será invisível, para que não atrapalhe a representação previamente criada. Os botões que ligam a esteira serão posicionados acima da esteira, na camada zero, como setas indicando o sentido. A tela da esteira finalizada pode ser observada na figura 6.2.

Para finalizar a alteração das telas, será editada a tela do braço.

### 6.2.3 Tela do braço

A tela do braço é a última tela que necessitará de alteração para a adição do módulo manual. Nela serão acionadas as ações de calibrar o braço, de girar o braço, de pegar ou entregar peça à esteira e de pegar ou entregar peça ao elevador.

Será adicionado um botão para calibrar o braço, na camada três, sobrepondo o quadrante referente à calibração do braço. Outro botão será adicionado na camada

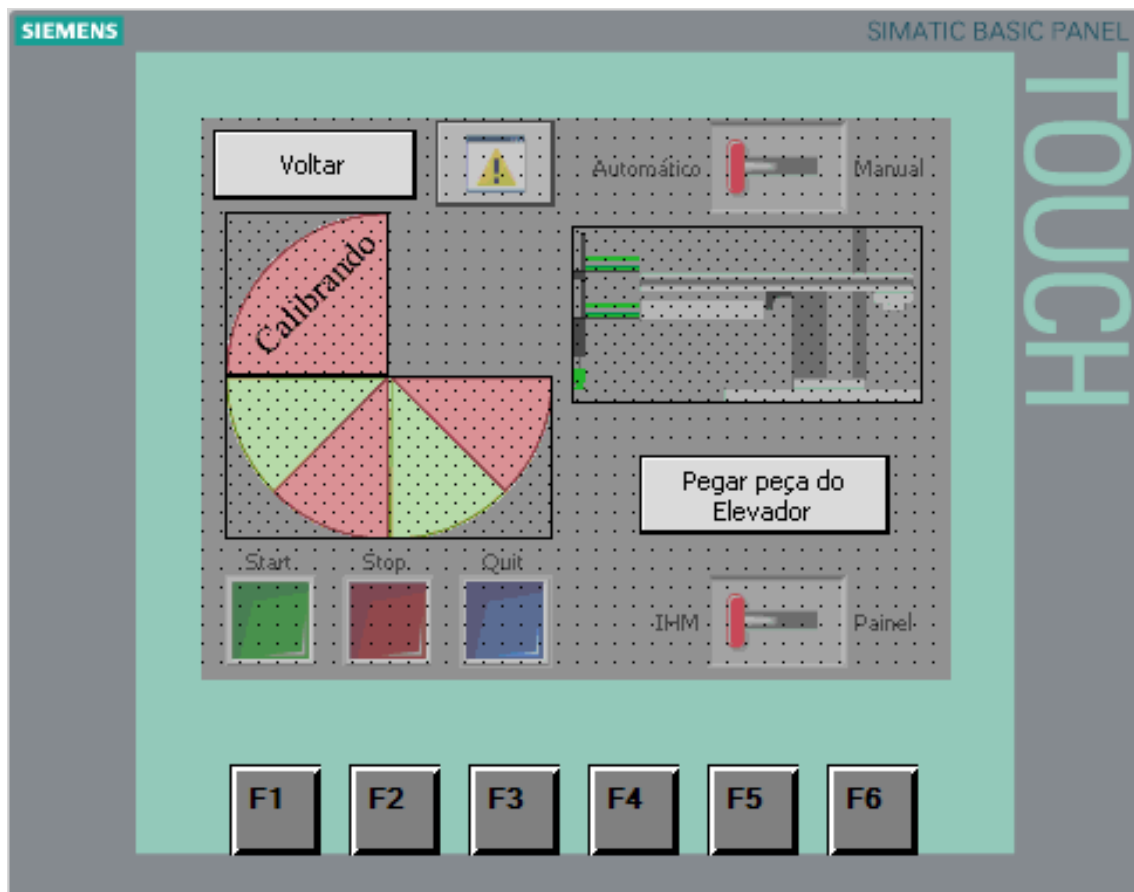


Figura 6.3: Módulo manual – Tela do braço.

três, sobrepondo o resto da pizza que representa o giro do braço, para executar a ação de girar o braço. Para executar a ação de pegar ou entregar peça à esteira, será adicionado um botão sobrepondo o braço, na camada quatro. Por fim, na camada zero, deve-se adicionar dois botões abaixo do braço para que um execute a ação de pegar a peça do elevador e outro execute a ação de entregar a peça ao elevador, com os nomes “Pegar peça do elevador” e “Entregar peça ao elevador”, respectivamente.

A tela do braço finalizada encontra-se representada na figura 6.3.

Concluindo a alteração de todas as telas necessárias para a adição do módulo manual, serão adicionadas as funcionalidades dos novos botões.

### 6.3 Passo 2: Funcionalidade dos botões

No passo 2, será descrita e sugerida uma forma de criar as lógicas ladder de todos os botões, para que executem as ações necessárias e estejam habilitados para o usuário, e a lógica de iniciar o modo manual e de retornar ao modo automático.

Para iniciar o modo manual, é necessário que o sistema esteja sendo comandado pela IHM e que esteja no estado de pausa (Stop). Para garantir esta situação, o inter-

ruptor Manual-Automático só estará disponível para o usuário nos estados corretos. Esta lógica deve ser criada com o auxílio de uma TAG “PermiteManual”, que será vinculada à animação de visibilidade do interruptor. A TAG “PermiteManual” será habilitada por uma bobina condicionada pelos contatos vinculados às TAGs “P1” e “P8b” em paralelo, para considerar apenas o estado de pausa do sistema, e em série com o contato IHM-Painel, para que esteja habilitada apenas com o comando sendo dado pela IHM.

O interruptor Manual-Automático será vinculado a uma nova TAG chamada “Manual”. Esta TAG será utilizada para outras lógicas, mas será também vinculada à animação de visibilidade do interruptor IHM-Painel. Os botões funcionais do módulo automático já possuem uma lógica para a animação de visibilidade de cada botão, e devem ser alteradas para incluir um contato NF com a TAG “Manual”, para que só apareçam para o usuário quando o sistema encontrar-se no modo automático. Os botões de ação do módulo manual terão sua animação de visibilidade vinculada diretamente à TAG “Manual”.

Outra TAG importante a ser criada é a TAG “Hab\_Acao”, que será utilizada para permitir o início de uma nova ação. Todas as lógicas de ação devem depender desta TAG para iniciar e devem desabilitar a TAG logo no início da ação, habilitando-a novamente apenas quando concluir todo o ciclo da ação. O interruptor Manual-Automático deve estar invisível para o usuário quando uma ação estiver sendo executada, para que não seja possível retornar ao modo manual durante o ciclo de uma ação. Portanto, deve-se condicionar a TAG “PermiteManual” a um contato NA da TAG “Hab\_Acao”.

Lembrando que, para o melhor funcionamento e segurança do sistema, é aconselhável reduzir o tempo de varredura de todas as TAGs criadas. A seguir serão descritas as lógicas das ações do módulo manual.

### **6.3.1 Lógica da ação do pistão**

A ação do pistão só pode ser executada quando não houver peça posicionada à frente do pistão, por tanto, é necessário criar uma TAG para registrar a existência de uma peça à frente do pistão ao final de cada ciclo. O ciclo da ação deve estender o pistão, até que o sensor de pistão estendido seja acionado, e retrain o pistão, até que o sensor de pistão retraído seja acionado, fechando o ciclo.

O início desta ação deve estar vinculado ao botão que sobrepõe o pistão, posicionado na tela da esteira.

### **6.3.2 Lógica da ação de ligar a esteira para a esquerda**

O ciclo da ação de ligar a esteira para a esquerda será composta apenas pelo ligamento da esteira para esquerda e o desligamento da esteira após 12,5 segundos. Como esta ação retira as peças da esteira, a TAG que registra a existência de uma peça à frente do pistão deve ser alterada para zero, utilizando uma bobina Reset.

O botão que aciona esta ação é o botão que encontra-se acima da esteira, na tela da esteira, referente ao movimento para a esquerda.

### **6.3.3 Lógica da ação de ligar a esteira para a direita**

O ciclo da ação de ligar a esteira para a direita é semelhante ao ciclo da ação de ligar a esteira para a esquerda, porém, se alguma peça ativar o sensor de fim de curso da esteira antes da contagem dos 12,5 segundos, o ciclo também é concluído. Como esta ação tem por finalidade transportar a peça para o final da esteira, ela apenas pode ser inicializada caso não haja outra peça no final da esteira. Da mesma forma que a ação de ligar a esteira para a esquerda, esta ação deve alterar a TAG que registra a existência de uma peça à frente do pistão para zero, utilizando uma bobina Reset.

O botão que aciona esta ação também encontra-se acima da esteira, na tela da esteira, mas, neste caso, será o botão referente ao movimento para a direita.

### **6.3.4 Lógica da ação de calibragem do braço**

Por segurança e para o bom andamento do sistema, a ação de calibragem do braço pode ser executada a qualquer instante, contanto que não haja outra ação sendo executada ao mesmo tempo. O ciclo desta ação consiste em girar o braço no sentido horário até o sensor indutivo do braço ser acionado. Neste instante, o braço retorna, girando no sentido anti-horário por 808 pulsações do contador de alta velocidade, caso não haja peça no sistema de sucção do braço, e 804 pulsações do contador de alta velocidade, caso haja peça no sistema de sucção do braço.

Esta ação será iniciada ao pressionar o botão que sobrepõe a representação do braço calibrando, encontrado na tela do braço.

### **6.3.5 Lógica da ação de girar o braço**

O ciclo da ação de girar o braço contém dois estados estáticos, braço virado para a esteira e braço virado para a prateleira, e ambos devem habilitar a TAG “Hab\_Acao” para que outras ações possam ser executadas. Enquanto o braço estiver virado para a esteira, ao executar a ação, o braço deve girar no sentido anti-horário por 1620 pulsações do contador de alta velocidade e parar, chegando ao estado em que o braço

encontra-se virado para a prateleira. Neste momento, ao executar a ação, o braço deve girar da mesma forma que a ação de calibragem do braço. Uma TAG deve ser criada para registrar em qual estado o ciclo se encontra.

Esta ação se iniciará com o comando do usuário dado ao botão que sobrepõe a representação de giro do braço, existente na tela do braço.

### **6.3.6 Lógica da ação de o braço pegar ou entregar peça à esteira**

O ciclo da ação de o braço pegar ou entregar peça à esteira também possui dois estados estáticos, braço sem peça e braço com peça, e, portanto, ambos devem habilitar a TAG “Hab\_Acao” para permitir que outras ações sejam executadas. Em ambos os estados, esta ação apenas será executada se o braço estiver virado para a esteira, ou seja, depende do estado do ciclo da ação de girar o braço.

Quando o braço estiver sem peça em seu sistema de sucção, ao executar a ação, o braço deve levantar, caso esteja abaixado, estender até acionar o sensor de braço estendido, ligar o sistema de sucção, abaixar e aguardar 1 segundo para garantir que a peça seja sugada, subir novamente e retrain. Nesse estado, a ação só pode ser executada se houver peça ao final da esteira.

No estado em que há peça no sistema de sucção do braço, ao executar a ação novamente o braço deve estender até acionar o sensor de braço estendido, abaixar e desligar o sistema de sucção, liberando a peça na esteira. Em seguida, o braço levanta e retrai novamente, fechando o ciclo. Nesse estado, a ação só pode ser executada se não houver peça ao final da esteira.

O início desta ação dependerá do acionamento do botão sobreposto ao braço, na tela do braço.

### **6.3.7 Lógica da ação de o braço entregar peça para o elevador**

A ação de o braço entregar peça para o elevador depende do estado do ciclo da ação de girar o braço e da presença de peça no sistema de sucção do braço. Esta ação apenas será executada caso o braço esteja virado para a prateleira, caso haja peça no sistema de sucção do braço e caso não haja peça no elevador. Faz-se necessário criar uma TAG para registrar se há peça no elevador.

Para facilitar a criação da lógica, conhecendo bem a lógica Ladder do sistema automático, acione a TAG referente ao lugar “P10”. Ao fazer isso, é possível garantir a segurança do sistema, já que o sistema seguirá os mesmo passos do módulo automático e não apresentará conflitos entre o braço e o elevador.

Após a execução da ação, é necessário informar ao sistema que o braço encontra-se voltado para a esteira e que não há peça no sistema de sucção do braço.

A execução desta ação estará vinculada ao botão “Entregar peça ao elevador”. Este botão estará visível ao usuário apenas se não houver peça no elevador.

### **6.3.8 Lógica da ação de o braço pegar a peça do elevador**

A ação de o braço pegar a peça do elevador também depende do estado do ciclo da ação de girar o braço e da presença de peça no sistema de sucção do braço, além de depender da TAG que registra se há peça no elevador. A ação só será iniciada caso o braço esteja virado para a prateleira, caso não haja peça no sistema de sucção e caso haja peça no elevador.

O ciclo desta ação consistirá em o elevador posicionar-se abaixo do braço, o braço estender até acionar o sensor de braço estendido, ligar o sistema de sucção, o braço descer por 1 segundo, para garantir que o braço pegou a peça, o braço levantar e retrain, e o elevador retornar à posição inicial.

Após a conclusão do ciclo da ação, é necessário alterar, no sistema, o estado do ciclo de o braço pegar ou entregar peça à esteira para considerar que há peça no sistema de sucção, e alterar a TAG que registra se há peça no elevador.

Esta ação será executada com o acionamento do botão “Pegar peça do elevador”, localizado na tela do braço. Este botão estará visível ao usuário apenas se houver peça no elevador.

### **6.3.9 Lógica da ação de o elevador retirar peça da prateleira**

Para facilitar o projeto, serão utilizados os mesmo botões de escolha de peça do módulo automático. Apenas serão alteradas as lógicas de aparecimento dos botões e suas funções no sistema.

Caso a ação não possa ser executada, os botões não estarão disponíveis ao usuário, para reduzir a chance de dano à planta mecatrônica. Os botões só estarão disponíveis caso haja peça na posição desejada, caso não haja outra ação sendo executada no momento, caso o braço esteja apontando para a esteira e caso não haja peça no elevador.

Para facilitar a criação do ciclo, conheça bem a lógica ladder do sistema e acione a TAG referente ao lugar “P33” e a TAG referente ao lugar “P###\*”, considerando que “##” é a numeração da posição desejada. Caso apenas estas TAGs fossem acionadas, o sistema iria continuar até retirar a peça pela esteira. Portanto, para parar o sistema no instante desejado, utilize bobinas Reset vinculadas às TAGs “P46” e “P47” durante toda a execução do ciclo para que elas sejam desativadas sem conseguirem dar continuidade no sistema.

Ao final do ciclo, haverá peça no elevador, e, portanto, será necessário registrar esta informação na TAG anteriormente criada.

### **6.3.10 Lógica de retorno do modo manual para o modo automático**

O retorno do modo manual para o modo automático deve garantir que o sistema encontra-se parado, da mesma forma que estava antes de iniciar o modo manual. Para isso, é necessário criar uma lógica com um ciclo de ações para retirar as peças que não encontrem-se na prateleira ou na torre do pistão. Para facilitar, serão utilizadas as ações criadas para o módulo manual. A lógica deve realizar três testes e executar as ações necessárias para retirar cada peça.

Para iniciar, deve ser testado se há peça na esteira, ou seja, se há peça à frente do pistão ou se há peça ao final da esteira. Caso haja, deve-se iniciar a ação de ligar a esteira para a esquerda.

Em seguida, deve-se testar se há peça no sistema de sucção do braço. Em caso afirmativo, deve-se iniciar sequencialmente as ações de calibrar o braço, de o braço pegar ou entregar peça à esteira e de ligar a esteira para a esquerda.

O último teste a ser realizado é se há peça no elevador. Caso haja, deve-se iniciar sequencialmente as ações de calibrar o braço, de girar o braço, de o braço pegar a peça do elevador, de calibrar o braço novamente, de o braço pegar ou entregar peça à esteira e de ligar a esteira para a esquerda.

Para finalizar, mesmo que todos os testes tenham se mostrado falsos, é necessário executar a ação de calibrar o braço para evitar erros no modo automático.

Com as lógicas de todas as ações do módulo manual criadas, será iniciado o passo 3, para a criação de alarmes.

## **6.4 Passo 3: Criação dos alarmes**

No passo 3, serão criados os alarmes e as suas respectivas lógicas de acionamento. Antes de iniciar a aplicação dos alarmes no projeto, será descrito o que é um alarme e quais são os tipos de alarme.

### **6.4.1 Alarmes**

Os alarmes desempenham um papel muito importante em um projeto automatizado. Um sistema de alarmes bem planejado garante a segurança dos operadores, a integridade do maquinário e a qualidade do produto ou serviço. Ele pode servir para alertar o operador, para tomar uma providência ou para ambos, simultaneamente.

Alarmes podem ser discretos ou analógicos e cada tipo aceita certos tamanhos de TAG como referência, podendo ser uma TAG da IHM ou do CLP. Os tamanhos de TAG aceitos por cada tipo encontram-se na tabela 6.1.

Tabela 6.1: Tipo de variáveis aceitas pelos alarmes

| Alarme Discreto |       | Alarme Analógico |       |
|-----------------|-------|------------------|-------|
| IHM             | CLP   | IHM              | CLP   |
| Short           | Dint  | Double           | Byte  |
| Ushort          | Dword | Float            | Char  |
|                 | Int   | Long             | Dint  |
|                 | Time  | Sbyte            | Dword |
|                 | UDInt | Short            | Int   |
|                 | UInt  | Ubyte            | Real  |
|                 | Word  | Ulong            | Sint  |
|                 |       | Ushort           | Time  |
|                 |       |                  | UDInt |
|                 |       |                  | UInt  |
|                 |       |                  | USInt |
|                 |       |                  | Word  |

A seguir, serão descritos os alarmes discretos e analógicos e suas funcionalidades.

#### 6.4.1.1 Alarmes discretos

Os alarmes discretos reconhecem apenas se um único bit de uma TAG está ativo ou não, isto é, se é 1 ou 0. A partir deste reconhecimento, os eventos se ativam.

As propriedades de um alarme discreto são divididos em “General”, “Trigger”, “Infotext” e “Acknowledgement”.

Em “General”, é possível determinar o texto do alarme, o número de identificação do alarme, a classe do alarme, a qual grupo de alarme ele pertence e o nome do alarme; “Trigger” permite determinar qual bit ativa o alarme e a qual TAG este bit pertence; por fim, as propriedades de “Infotext” são utilizadas para descrever o alarme.

A função “Acknowledgement” não funciona em um “Basic Panel”, que é o caso da IHM encontrada no LCA da UFRJ. Apenas para conhecimento, esta função permite que o próprio sistema tome conhecimento do alarme de forma automática. Essa funcionalidade pode ser usada no caso de um diagnosticador de falhas corrigir o problema automaticamente.

#### 6.4.1.2 Alarmes analógicos

Os alarmes analógicos reconhecem se a TAG é maior ou menor que um determinado valor. O valor deve ser do mesmo tipo da TAG. Valores inteiros para TAGs inteiras,

valores reais para TAGs reais e valores de tempo para TAGs de tempo. A partir deste reconhecimento, os eventos se ativam.

As propriedades de um alarme analógico dividem-se em três campos: “General”, “Trigger” e “Infotext”.

O campo “General” possibilita determinar o texto do alarme, o número de identificação do alarme, a classe do alarme, a qual grupo de alarme ele pertence e o nome do alarme; em “Trigger”, é possível determinar a TAG que ativa o alarme, o atraso entre a alteração do valor da TAG e a ativação do alarme, o valor limite, se o alarme será acionado quando o valor da TAG for maior ou menor que o valor limite, se terá uma tolerância, quantitativa ou percentual e quando esta tolerância será considerada; e “Infotext” permite descrever o alarme.

Em relação aos eventos, os alarmes aceitam o evento “Incoming”, que executa a ação quando o alarme é acionado, o evento “Outgoing”, que executa a ação quando o alarme é desativado, o evento “Acknowledge”, que executa a ação quando o operador informa para o sistema que já tem conhecimento de que o alarme foi acionado, e o evento “Loop-in alarm”, que executa a ação se o operador ordenar, pressionando o botão “Loop-in-alarm” a partir da mensagem de alarme, quando o alarme for acionado.

Os alarmes são classificados em classes baseado na importância de cada um. As propriedades e funcionalidades de cada classe será explicado a seguir.

#### **6.4.1.3 Classes de alarme**

As classes de alarme servem para determinar a importância do alarme. Três classes são predefinidas pelo sistema, “Errors”, “Warnings” e “System”, e não podem ser apagadas.

- A classe “Errors” necessita de “Acknowledge”, ou seja, necessita que o operador reconheça que o alarme foi acionado pressionando o botão de “Acknowledge” na mensagem de alarme. Como configuração padrão, ao utilizar um gerenciador de alarmes, os alarmes da classe “Errors” são identificados com um ponto de exclamação (“!”).
- A classe “Warnings” não permite “Acknowledgement” e, como configuração padrão, os alarmes da classe “Warnings” não são identificados por nenhum símbolo.
- A classe “System” não permite “Acknowledgement” e, como configuração padrão, os alarmes da classe “System” são identificados com um cifrão (“\$”).

As propriedades de uma classe dividem-se em quatro campos: “General”, “Acknowledgement”, “Status” e “Colors”.

No campo “General”, é possível determinar o nome da classe e o símbolo de identificação; “Acknowledgement” permite determinar se a classe necessitará do reconhecimento do operador ao ocorrer um alarme; “Status” serve para determinar o código para identificação da situação do alarme que pode ser ativado, desativado ou já ter enviado um aviso para os dispositivos; e, por fim, em “Colors”, é possível determinar qual será a cor apresentada na lista de alarmes para cada possível situação.

Para organizar os alarmes, são utilizados grupos de alarme, e suas propriedades e funcionalidades serão descritas a seguir.

#### **6.4.1.4 Grupos de alarme**

Os grupos de alarme servem apenas como forma de organização. É aconselhável que, para um sistema grande, sejam criados grupos de alarmes e que os alarmes sejam alocados por setores.

A única possível edição do grupo de alarmes é o nome e encontra-se em suas propriedades gerais.

Após conhecer os tipos de alarme, as classes de alarme e os grupos de alarme, é possível iniciar a criação dos alarmes para o módulo manual.

#### **6.4.2 Criação das TAGs dos alarmes e lógicas de acionamento**

Para iniciar a criação dos alarmes, é necessário criar as TAGs dos alarmes. Os alarmes serão criados pensando em explicar para o usuário as situações em que as ações do módulo manual não são executadas.

Será necessário criar doze TAGs booleanas no CLP para receber todos os tipos de alarme, que são acarretados pelas situações enumeradas abaixo:

1. Nenhuma ação estar habilitada por haver outra ação acontecendo no instante.
2. Não ser possível ativar o pistão por haver peça à frente do pistão.
3. Não poder ligar a esteira para a direita por haver peça ao final da esteira.
4. Ao tentar pegar ou entregar peça para a esteira, o braço estiver voltado para a prateleira.
5. Ao tentar pegar peça da esteira, não houver peça ao final da esteira.
6. Ao tentar entregar peça à esteira, já houver peça ao final da esteira.
7. Ao tentar entregar peça ao elevador, o braço estiver voltado para a esteira.

8. Ao tentar entregar peça ao elevador, não houver peça no sistema de sucção do braço.
9. Ao tentar entregar peça ao elevador, já houver peça no elevador.
10. Ao tentar pegar a peça do elevador, o braço estiver voltado para a esteira.
11. Ao tentar pegar a peça do elevador, já houver peça no sistema de sucção do braço.
12. Ao tentar pegar peça do elevador, não houver peça no elevador.

Essas TAGs devem ser criadas no CLP utilizando posições sequenciais na memória e ocupando apenas dois bytes no total. Para que a IHM possa reconhecer essas TAGs e utilizá-las em um alarme, é necessário criar uma TAG na IHM sem referenciar nenhuma TAG em específico. Essa TAG deve ser do tipo Word, deve ser alterada para uma TAG externa com “HMI\_connections” vinculada ao CLP e deve ocupar os dois bytes utilizados pelas TAGs que representam as situações dos alarmes. Lembrando que é importante reduzir o tempo de varredura da TAG criada para o melhor funcionamento do sistema.

As lógicas de acionamento dos alarmes devem ser criadas utilizando bobinas do tipo Set para acionar o alarme e utilizando os contatos necessários, para cada lógica, de forma a obedecer a descrição das situações enumeradas.

Em seguida, serão criados os alarmes e seus textos, com as descrições que aparecerão para o usuário.

### **6.4.3 Criação dos alarmes**

Como descrito na seção 6.4.1, os alarmes discretos são utilizados para condições booleanas de alarme, como os alarmes que serão utilizados neste projeto, e os alarmes analógicos são utilizados para condições numéricas de alarme.

Deve-se criar alarmes discretos da classe “Error”, referenciar a TAG do tipo Word criada na IHM e determinar o bit de disparo como a posição da memória referente ao respectivo alarme. O nome do alarme será o título da descrição do alarme apresentada ao usuário, portanto deve-se escrever de forma clara os nomes, para que fique elucidativo ao usuário qual alarme foi acionado.

Com o objetivo de o alarme manter-se acionado enquanto o usuário não reconhecer o alarme, a lógica foi criada utilizando bobinas do tipo Set. Para que os alarmes reconhecidos pelo usuário não fiquem aparentes, deve-se criar um evento do tipo acknowledged, para cada alarme, que altere o valor da TAG, que inicia o alarme, para zero.

Por fim, vá nas propriedades de cada alarme e edite seu “Infotext” para melhor descrever o motivo do alarme para o usuário.

Na edição template 2, para a adição do módulo manual ao projeto, foi sugerido que copiasse o botão que apresenta para o usuário os alarmes acionados, localizado no template 1. Caso deseje criar um botão novo, realize todos os ajustes de aparência desejados no novo botão e crie um evento do tipo “Click” para abrir o aviso de alarme dos alarmes do tipo unacknowledged. Se o projeto não possui este aviso de alarme, será necessário criar um novo com características específicas.

#### 6.4.4 Avisos de alarme

Os avisos de alarme servem para alertar o operador que ocorreu um determinado tipo de alarme. Nas telas de aviso de alarme, ficam listados todos os alarmes que ocorreram, com as informações escolhidas durante o projeto, e ficam disponíveis os botões de “Loop-in-alarm”, “Acknowledge” e “Infotext”.

As propriedades dos avisos de alarme dividem-se em oito campos: “General”, “Appearance”, “Layout”, “Display”, “Text format”, “Columns”, “Miscellaneous” e “Security”.

No campo “General”, é possível escolher quais tipos de alarme serão indicados por este elemento e a situação do alarme; “Appearance” permite determinar a cor do texto e a cor do aviso; “Layout” serve para determinar a posição e tamanho do campo ou se o campo se adequará ao tamanho do conteúdo, quantas linhas serão usadas por alarme e quantos alarmes aparecerão por aviso; seguindo para o campo “Display”, é permitido escolher se será possível rolar verticalmente a lista de alarmes, se terá uma barra de rolagem na lateral, se terá um botão de informações, se terá um botão para reconhecer os alarmes do sistema e se terá um botão para acionar o evento de loop-in do alarme; e em “Text format”, é possível determinar a fonte dos textos no aviso de alarmes.

“Columns” permite escolher quais dados serão apresentados para cada alarme, se o tempo será apresentado em milissegundos e se a ordem será crescente ou decrescente. As opções de dados que podem ser apresentados por alarme são o número do alarme, o momento do alarme, o status do alarme, o nome do alarme, a data em que ocorreu o alarme, a classe do alarme, o grupo do alarme, se o alarme é diagnosticável e a qual dispositivo ele pertence

No campo “Miscellaneous”, é possível determinar o nome do aviso de alarmes e a camada em que ele se encontrará na tela; e, por fim, a função “Security” do aviso de alarme não permite edições.

Em relação aos eventos, os avisos de alarme aceitam o evento “Enable”, que é acionado quando o aviso aparece para o usuário, e o evento “Disable”, que aciona

quando o aviso some.

O aviso de alarme criado pelo usuário deve exibir alarmes do tipo “Error” e “Unacknowledged alarms” apenas. É necessário escolher quais informações estarão disponíveis na lista de alarmes, no campo “Columns”, e permitir os botões “Infotext” e “Acknowledge” no campo “Display”.

Isso conclui o módulo manual, com todas as suas ações e seus alarmes de aviso ao usuário, e sua adição ao projeto do sistema de automação da planta mecatrônica. No capítulo final, serão apresentadas as funções que não foram utilizadas no projeto.

# Capítulo 7

## Funções e Objetos Não Utilizados

Durante a criação do projeto, conforme uma função ou um objeto era necessário, sua aplicação foi precedida de uma explicação. Este capítulo será reservado para explicar as funções e os objetos não utilizados no projeto, começando pelos objetos inanimados, os objeto menos complexos.

### 7.1 Objetos inanimados

Os objetos inanimados são definidos pelos objetos que não sofrem alterações em sua aparência, nem com a interação com o usuário e nem com variáveis do sistema, portanto, não possuem eventos e são utilizadas unicamente para editar a apresentação da tela. Os objetos inanimados não utilizados foram: linhas, círculos, elipses e retângulos.

#### 7.1.1 Linhas

As linhas podem ser utilizadas como linhas para desenhar uma figura, como linhas de contorno ou como setas para salientar alguma representação na tela.

As propriedades de uma linha dividem-se em três campos: “Appearance”, “Layout” e “Miscellaneous”.

O campo “Appearance” possibilita editar a grossura da linha, o estilo (contínua ou tracejada), a cor da linha, a cor do fundo (caso a linha seja tracejada) ou se este fundo será transparente, se as pontas serão setas e se os cantos serão arredondados ou quadrados; em “Layout”, é possível editar a posição e o tamanho do retângulo circunscrito à linha e a posição dos pontos iniciais e finais da linha; e “Miscellaneous” permite editar o nome da linha e em qual camada ela ficará em uma tela.

### 7.1.2 Círculos, elipses e retângulos

Os objetos como círculo, elipse e retângulo também são utilizados para criar a representação do sistema por meio da IHM. Eles possuem as mesmas propriedades e animações, por isso serão explicados em conjunto.

As propriedades destes objetos dividem-se em três campos: “Appearance”, “Layout” e “Miscellaneous”.

No campo “Appearance”, é possível editar a cor do objeto, se seu interior será sólido ou transparente, a grossura da linha de contorno, a continuidade da linha de contorno e a cor da linha de contorno; “Miscellaneous” permite editar o nome da forma geométrica e em qual camada ele ficará em uma tela.

Em “Layout”, é possível determinar a posição e o tamanho da forma geométrica escolhida. A maneira de determinar o tamanho varia com a forma geométrica. Para a elipse, é possível determinar os diâmetros ou os raios horizontal e vertical. Para o retângulo, é possível determinar o tamanho de cada lado e adoçar as arestas. Para o círculo, basta determinar o raio.

Os próximos objetos não utilizados a serem descritos são os objetos que possuem animações para interações com o sistema.

## 7.2 Objetos animados

Os objetos animados são definidos pelos objetos que sofrem alterações em sua aparência dependendo de variáveis do sistema, portanto, são utilizadas para apresentar ao usuário o valor de uma determinada Tag. Os objetos animados não utilizados foram: barras e gráficos.

### 7.2.1 Barra

A barra é utilizada para representar o nível de algum sistema físico ou o valor de alguma variável contínua e é capaz de alertar quando um limite predeterminado é ultrapassado. Caso algum limite extremo seja definido, ao ser ultrapassado, a IHM irá disparar um alarme.

As propriedades da barra dividem-se em nove campos: “General”, “Appearance”, “Design”, “Scales”, “Label”, “Layout”, “Text format”, “Limits” e “Miscellaneous”.

Em “General”, é possível determinar o máximo e mínimo da barra e a TAG de referência; já em “Appearance”, é possível determinar a cor que representa o conteúdo do recipiente, a cor do que não está preenchido pelo conteúdo, se este gradiente de cores será aplicado em toda a barra ou não, a cor do texto, a cor do fundo e se os limites serão representados por linhas ou setas; “Design” permite determinar o estilo da borda e se terá efeito 3D; e “Scale” permite determinar se terá ou não uma escala

na lateral da barra, em quantas partes cada intervalo será dividido, a cada quantas marcações terá indicação de valor e de quanto será o espaço entre cada marcação grande.

Em “Label”, é possível determinar quantos algarismos serão utilizados à esquerda da vírgula e à direita da vírgula, se terá marcação de números e se usará escala logarítmica; no campo “Layout”, é possível determinar a posição e o tamanho da barra, se a escala será colocada à direita e embaixo ou à esquerda e em cima e o alinhamento da barra; “Text format” permite determinar a fonte do texto; “Limits” permite determinar a cor do campo de dados caso a TAG de referência ultrapasse o limite superior ou o limite inferior; por fim, em “Miscellaneous”, é possível determinar o nome da barra e a camada em que ela se encontrará na tela.

### 7.2.2 Gráfico

Os gráficos também são utilizados para representar o nível de algum sistema físico ou o valor de alguma variável contínua, mas com a diferença de que são capazes de armazenar os últimos valores registrados pela variável, podendo ser utilizados para avaliar sua tendência.

As propriedades do gráfico dividem-se também em nove campos: “Trend”, “Appearance”, “Layout”, “Text format”, “Table”, “Time axis”, “Left value axis”, “Right value axis” e “Miscellaneous”.

“Trend” permite criar uma nova curva para o gráfico, escolhendo o nome da curva, o estilo da linha, quantos valores aparecerão no gráfico, a TAG de referência, o lado do eixo e as cores da curva dependendo dos limites; já no campo “Appearance”, é possível determinar a cor dos eixos, a cor de fundo, o sentido do gráfico, se mostrará a escala do gráfico e qual será sua cor; em “Layout”, é possível determinar a posição e o tamanho do gráfico; e em “Text format”, é possível determinar a fonte dos textos.

Em “Table”, é possível determinar se haverá tabela ou não, se haverá grade na tabela, qual será a cor da linha e a cor de fundo, quantas linhas serão visíveis e as cores do cabeçalho; no campo “Time axis”, é possível determinar se haverá eixo de tempo, eixo definido por pontos ou eixo com valores final e inicial definidos, qual o intervalo de tempo no eixo, se haverá marcações, qual será o espaçamento entre elas e a cada quantas marcações terá valor; já nos campos “Left value axis” e “Right value axis”, é possível determinar qual eixo aparecerá no gráfico, o valor inicial e final do eixo ou se o eixo será gerado automaticamente, se haverá uma linha para algum valor de referência, se haverá marcações, a cada quantas marcações haverá valores e qual a quantidade de algarismos desses valores; e “Miscellaneous” permite determinar o nome do gráfico e a camada em que ele se encontrará na tela.

Em relação aos eventos, os gráficos aceitam o evento “Enable”, que é acionado

quando o campo é ativado para uso, e o evento “Disable”, que aciona quando o campo é desativado para uso.

Além dos objetos, outros elementos que compõem a gama de possibilidades disponível pelo TIA Portal são as listas.

## 7.3 Listas

As listas são utilizadas como interface de navegação para um banco de dados, criado ainda durante o projeto. As listas não utilizadas foram: as listas de usuários e as listas de receitas.

### 7.3.1 Lista de usuários

A lista de usuários serve como controle de quais usuários cadastrados existem e de qual grupo eles fazem parte.

As propriedades da lista de usuários dividem-se em cinco campos: “General”, “Appearance”, “Layout”, “Miscellaneous” e “Security”.

Em “General”, é possível determinar o número de linhas que terá a lista, a cor de fundo e a cor e a fonte do texto; em “Appearance”, é possível determinar se terá uma barra de rolagem; “Layout” permite determinar a posição e o tamanho do campo ou se ele se adaptará ao conteúdo; “Miscellaneous” permite determinar o nome da lista de usuários e a camada em que ela se encontrará na tela; e “Security” permite determinar quem tem autorização para utilizar este elemento.

Em relação aos eventos, a lista de usuários aceita o evento “Enable”, que é acionado quando a lista aparece para o usuário, e o evento “Disable”, que aciona quando a lista some.

### 7.3.2 Lista de receitas

A lista de receitas serve para consultar as anotações ou receitas previamente feitas pelo usuário que contenham instruções de comando.

As propriedades da lista de receitas dividem-se em oito campos: “General”, “Appearance”, “Layout”, “Text format”, “Buttons”, “Simple view”, “Security” e “Miscellaneous”.

“General” permite determinar a receita de referência e em qual TAG será gravada a receita; já no campo “Appearance”, é possível determinar as cores do campo e as cores de um objeto selecionado; em “Layout”, é possível determinar a posição e o tamanho da lista e quantas receitas serão visíveis por vez; “Text format” permite determinar qual será a fonte do texto; “Buttons” permite determinar quais botões aparecerão na lista, sendo eles os botões de informações, de “Salvar como...”, de

escrita e leitura no CLP, de adicionar receita, de apagar receita, de salvar e de renomear receita, além dos botões de menu e de voltar; em “Simple view”, é possível determinar a posição das receitas, o comprimento do campo e o número de linhas para cada receita, se aparecerá o número da receita, se o tamanho da lista se adaptará ao conteúdo e se terá barra de rolagem; em “Security”, é possível determinar quem tem autorização para utilizar este elemento; por fim, em “Miscellaneous”, é possível determinar o nome do aviso de alarmes e a camada em que ele se encontrará na tela.

Para que haja uma lista de receitas, é necessário criar as receitas e seus ingredientes.

## 7.4 Receitas

As receitas servem para anotar o passo a passo de como produzir certo produto usando o sistema. Elas devem ser predefinidas ou adicionadas pelo operador e servem para consulta ou para coordenar o sistema de forma que produza o bem.

As propriedades de uma receita dividem-se em três campos: “General”, “Loading” e “Infotext”.

Em “General”, é possível editar o nome da receita, o nome que aparece para o operador, a versão da receita em forma de data e horário e o número da receita; “Loading” permite sincronizar a receita; por fim, em “Infotext”, é possível descrever a receita.

As receitas são compostas por ingredientes e instruções para utilização dos ingredientes.

### 7.4.1 Ingredientes

Cada receita possui seus ingredientes e cada ingrediente deve ter uma TAG de referência para que a receita seja seguida.

As propriedades de um ingrediente dividem-se em três campos: “General”, “Values” e “Infotext”.

O campo “General” possibilita editar o nome do ingrediente e o nome que aparece para o operador; em “Value”, é possível determinar a TAG de referência e seu valor padrão; e em “Infotext”, é possível descrever o ingrediente.

### 7.4.2 Instruções

É necessário, também, determinar as instruções de uma receita. As instruções servem para determinar a quantidade que cada ingrediente será usado para a receita ou para uma variação da mesma receita.

As propriedades das instruções dividem-se em apenas dois campos: “General” e “Comment”.

Em “General”, é possível editar o nome da instrução, o nome que aparece para o operador e o número da instrução; e em “Comment”, é registrada uma anotação.

Além dos elementos já citados durante o presente capítulo, o último elemento não utilizado no projeto foi a agenda de tarefas.

## 7.5 Agenda de tarefas

A agenda de tarefas permite que uma série de ações seja executada na ocorrência de um determinado evento. Os eventos capazes de executar uma tarefa agendada são: “Disabled”, “Runtime stop”, “Screen change”, “Alarm buffer overflow”, “User change”, “When dialog is opened” e “When dialog is closed”. As especificações dos eventos encontram-se listadas abaixo.

- Ao selecionar “Disabled”, esta tarefa fica desabilitada;
- O evento “Runtime stop” ocorre quando o sistema da IHM para;
- O evento “Screen change” ocorre quando troca-se de uma tela para outra;
- O evento “Alarm buffer overflow” ocorre quando o registro de alarmes chega ao limite;
- O evento “User change” ocorre quando troca-se de um usuário para outro; e
- Os eventos “When dialog is opened” e “When dialog is closed” ocorrem quando a janela de mensagem abre ou fecha, respectivamente.

Após concluir o projeto supervisorio, adicionar o módulo manual e listar todas as funções não utilizadas, conclui-se o projeto.

# Capítulo 8

## Conclusão

Este trabalho teve como objetivo a criação de um tutorial de como criar um projeto de sistema supervisório para um sistema automatizado de estocagem de peças em uma prateleira, realizado na planta mecatrônica Cube Assembly localizada no Laboratório de Controle e Automação da Universidade Federal do Rio de Janeiro. Foi utilizado, como painel de interface homem-máquina, o modelo KTP600 da SIEMENS e, como computador lógico programável, o modelo S7-1200, também da SIEMENS.

Foram introduzidos o funcionamento da lógica Ladder para controle, os componentes utilizados durante o projeto, como bobinas e contatos, e como aplicar a lógica Ladder em um CLP e uma IHM. Foi descrito o projeto [4], com suas funcionalidades e formas de utilização, juntamente com todos os dispositivos da planta mecatrônica Cube Assembly.

Após a introdução do conceito e a descrição do projeto, foi descrito como criar um novo projeto supervisório em uma IHM KTP600 com o auxílio do software Totally Integrated Automation Portal, desde a instalação dos dispositivos até às suas configurações dentro do software.

O projeto do sistema supervisório teve como objetivo inicial a modelagem do sistema de estoque de peças, para que os estados de todos os atuadores e sensores fossem transcritos de forma clara para o usuário através do painel IHM e permitisse que todos os comandos fossem acionados de forma remota.

Para incrementar o projeto do sistema supervisório, foi adicionado um módulo manual para acionar os atuadores separadamente, mas restritos a nove ciclos lógicos fechados. Os ciclos possuíam intertravamentos para evitar falhas no sistema, ao mesmo tempo que todos os intertravamentos acionavam um alarme para informar ao usuário o motivo da impossibilidade de iniciar o ciclo.

Todos os passos da criação do projeto do sistema supervisório foram descritos neste projeto, dando a possibilidade de reproduzi-lo, considerando um certo conhecimento em sistemas a eventos discretos, e pode ser utilizado como base para o

aprendizado inicial das funcionalidades da IHM KTP600.

## **8.1 Sugestões de trabalhos futuros**

O sistema supervisorio permite o acionamento e controle de sistemas industriais de forma remota. Com isso, podem ser sugeridos como trabalhos futuros a criação de sistemas supervisorios mais complexos, com a utilização das funções de receita, do sistema de hierarquia de usuários e da agenda de tarefas, as quais não foram utilizadas no presente trabalho, e a transcrição do tutorial apresentado neste trabalho para uma IHM de um modelo mais moderno, mais robusto e com maior número de funcionalidades.

Espera-se que o presente trabalho auxilie na criação de novos projetos e desenvolvimento de novas tecnologias utilizando um painel de interface homem-máquina sensível ao toque em conjunto com um computador lógico programável.

# Referências Bibliográficas

- [1] LUGLI, A. B., SANTOS, M. M. D. *Redes Industriais para Automação Industrial: AS-I, PROFIBUS e PROFINET*. 1 ed. São Paulo, Editora Érica Ltda., 2010.
- [2] LANCELLOTE JUNIOR, F. P. *Automação de uma Planta Mecatrônica Modelada por uma Rede de Petri Interpretada para Controle*. Projeto de graduação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, 2014.
- [3] *SIMATIC S7-1200 Easy Book*. Manual A5E02486774-AG, SIEMENS AG, 2015.
- [4] SILVEIRA, W. R. *Projeto de um Sistema de Armazenamento Automatizado*. Projeto de graduação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, 2015.
- [5] “christiani.eu”. 2015. Disponível em: <<http://www.christiani.eu/AutomationTechnology.html>>. Acessado dia 03 de Março de 2015.
- [6] MOREIRA, M. V., BASÍLIO, J. C. “Bridging the Gap Between Design and Implementation of Discrete-Event Controllers”. In: *IEEE Transactions on Automation Science and Engineering*, v. 11, IEEE, pp. 48–65, 2014.
- [7] BOTELHO, D. S. *Projeto de um Sistema de Automação de uma Célula de Manufatura Utilizando CLP SIEMENS S7-1200*. Projeto de graduação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, 2011.

# Índice Remissivo

Alarms, 83

- Alarm classes, 85

- Alarm groups, 86

- Alarm view, 88

- Analog alarms, 84

- Discrete alarms, 84

Bar, 91

Button, 53

Circle, 91

Date/time field, 58

Ellipse, 91

Graphic view, 50

I/O field, 57

Line, 90

Recipes, 94

- Data records, 94

- Elements, 94

- Recipe view, 93

Rectangle, 91

Scheduled tasks, 95

Screens, 43

Switch, 55

TAG, 35

Text & graphic lists, 57

Text field, 52

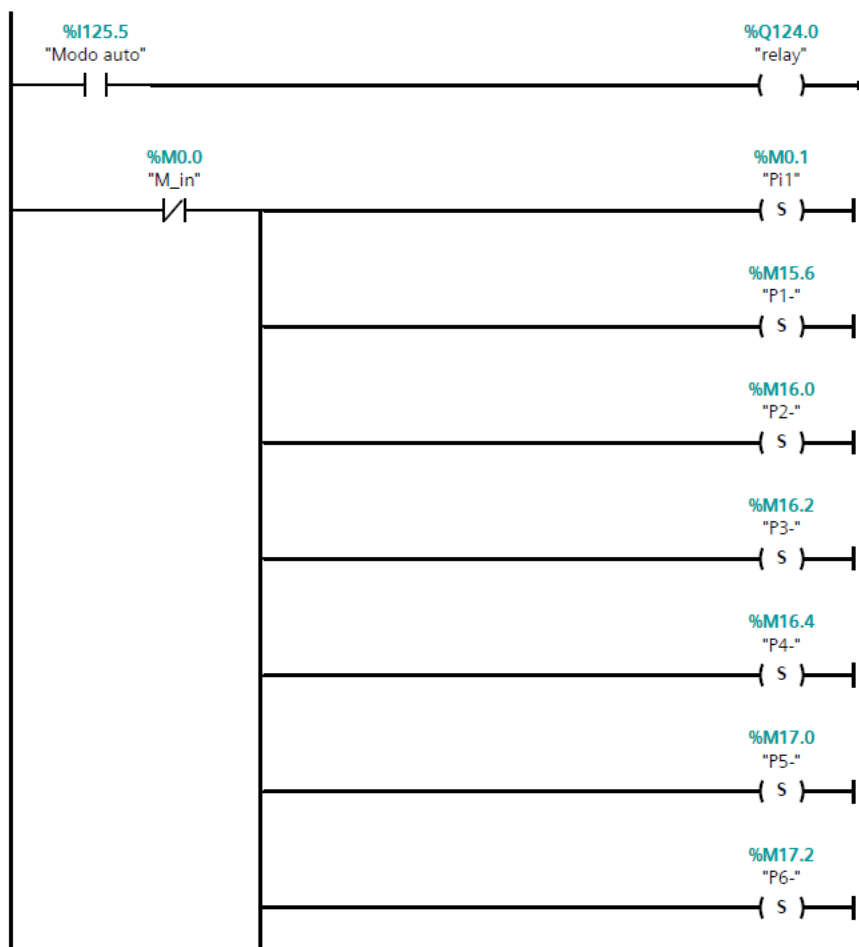
Trend view, 92

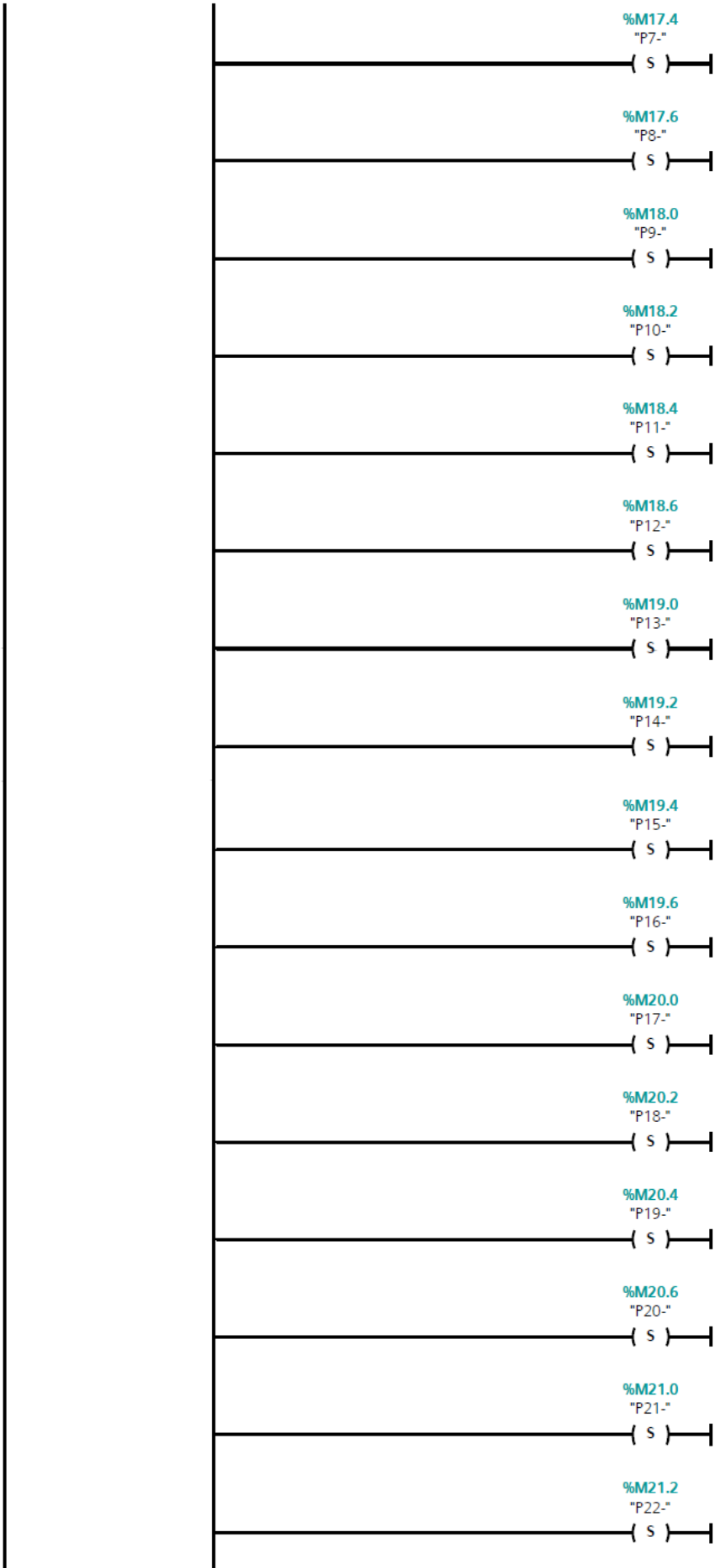
User view, 93

# Apêndice A

## Lógica Ladder Utilizada no Sistema Automatizado

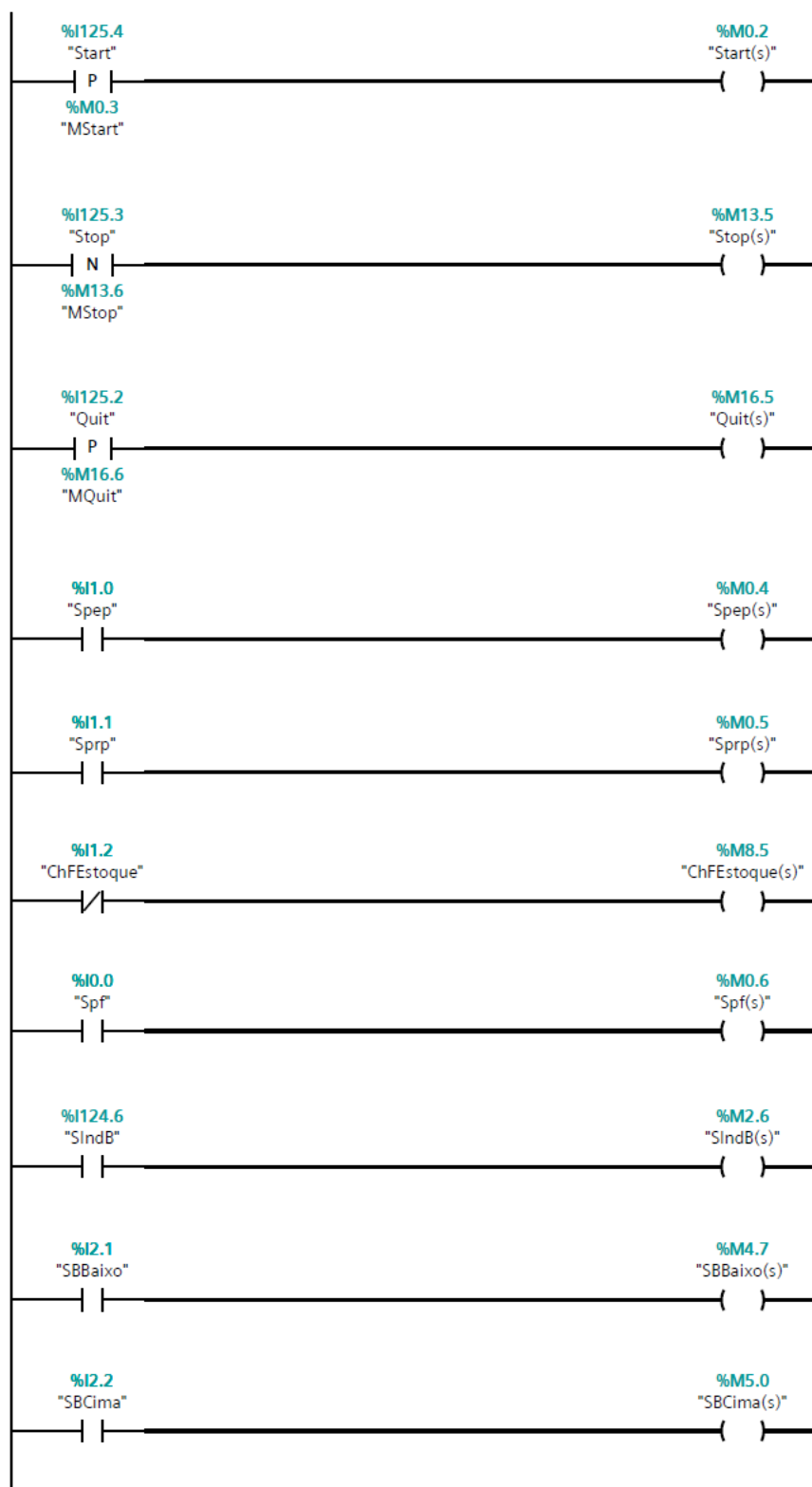
### A.1 Módulo de Inicialização

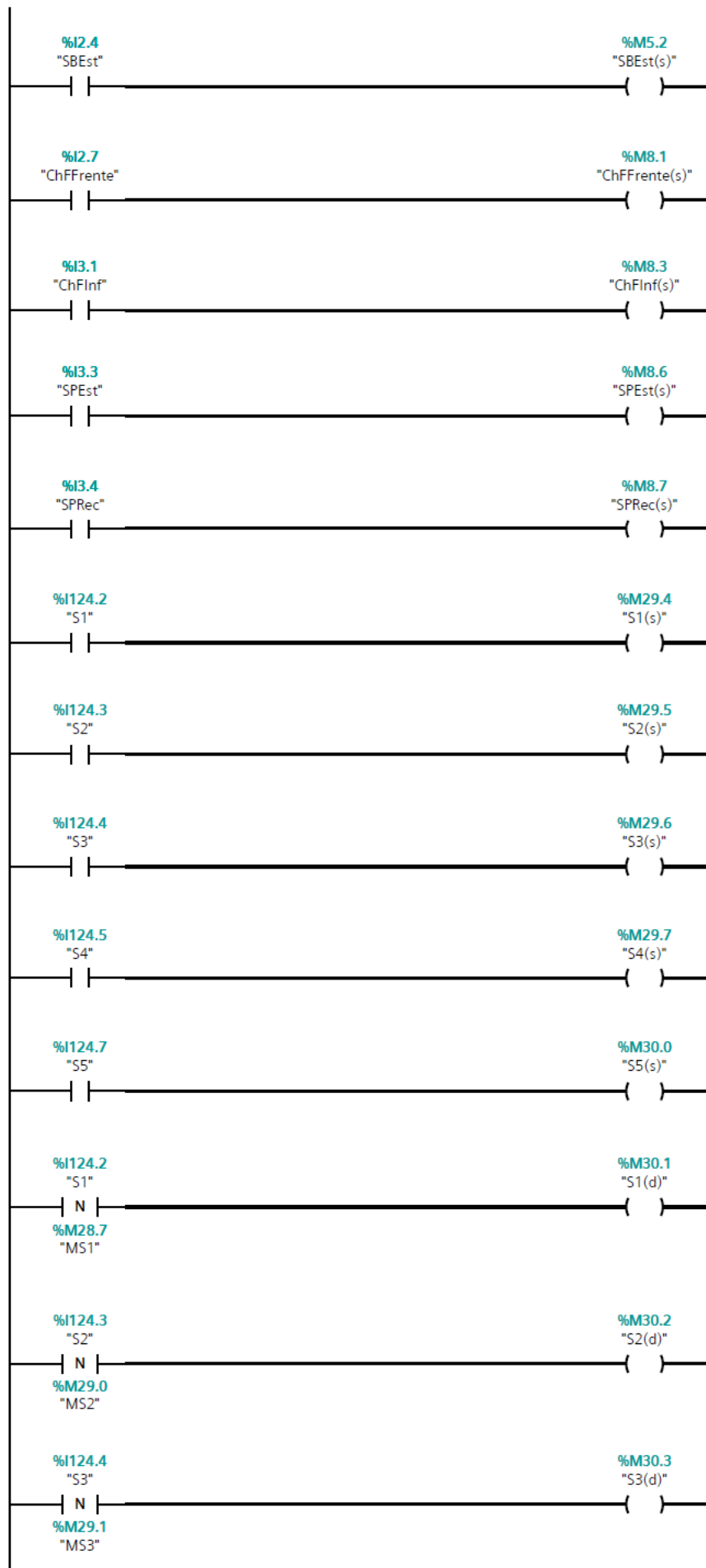






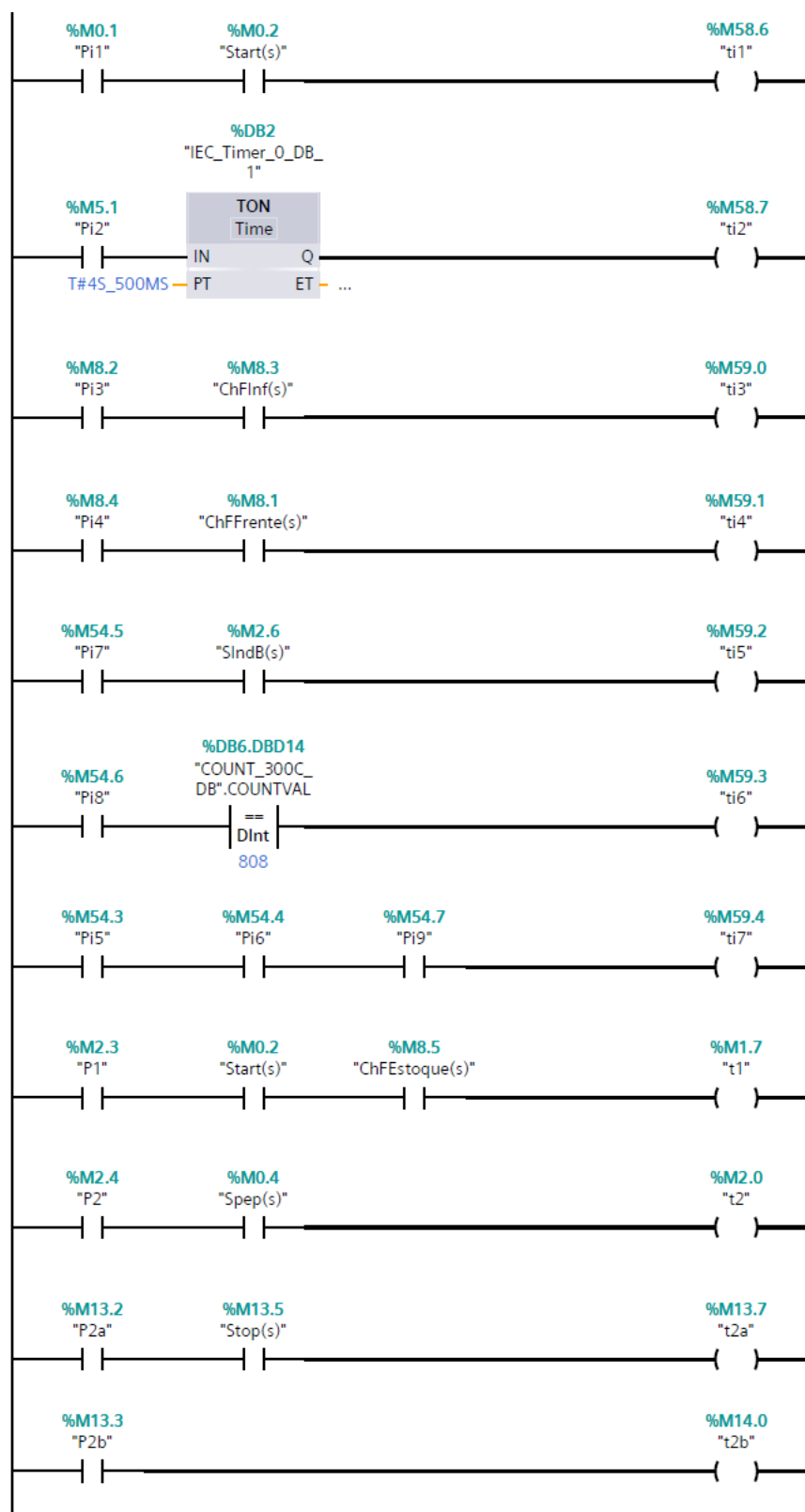
## A.2 Módulo dos Eventos

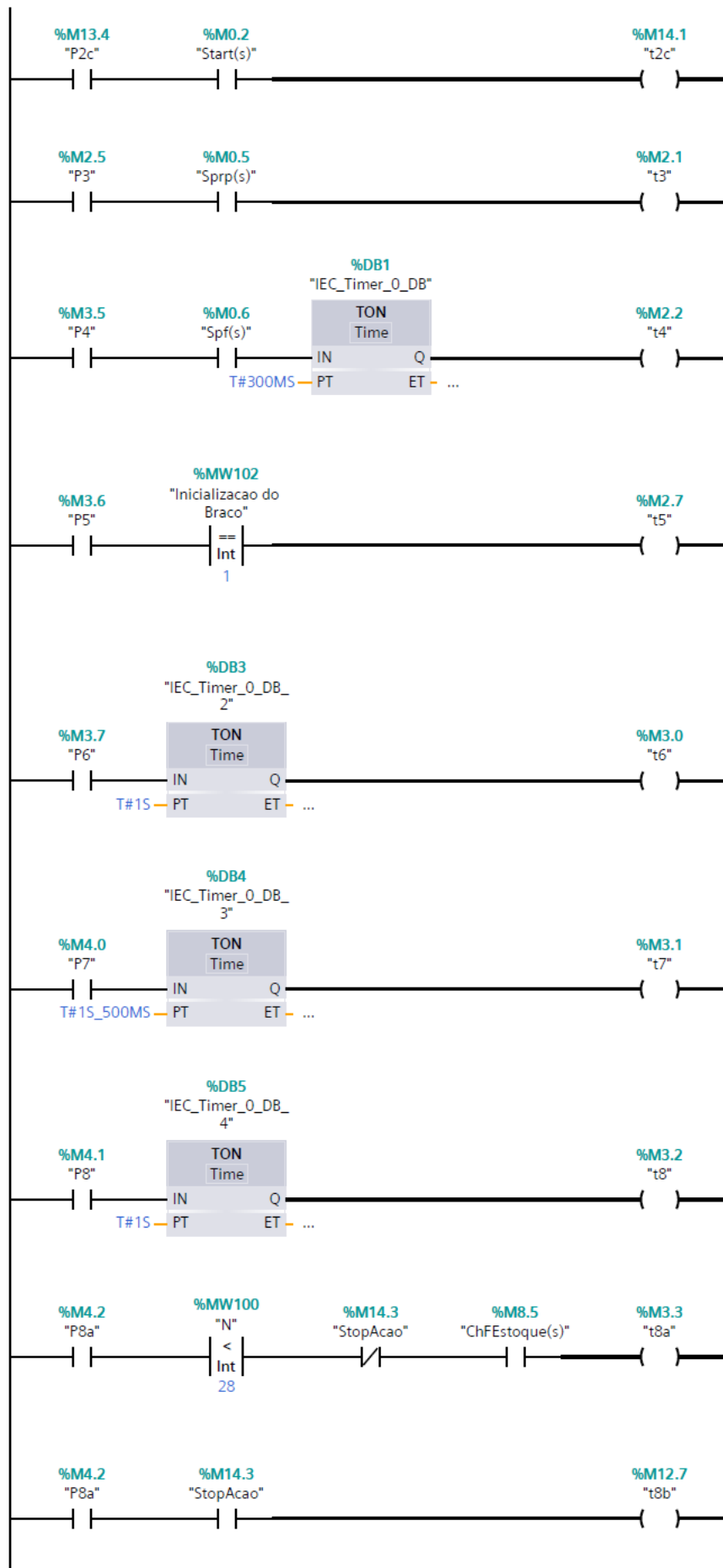


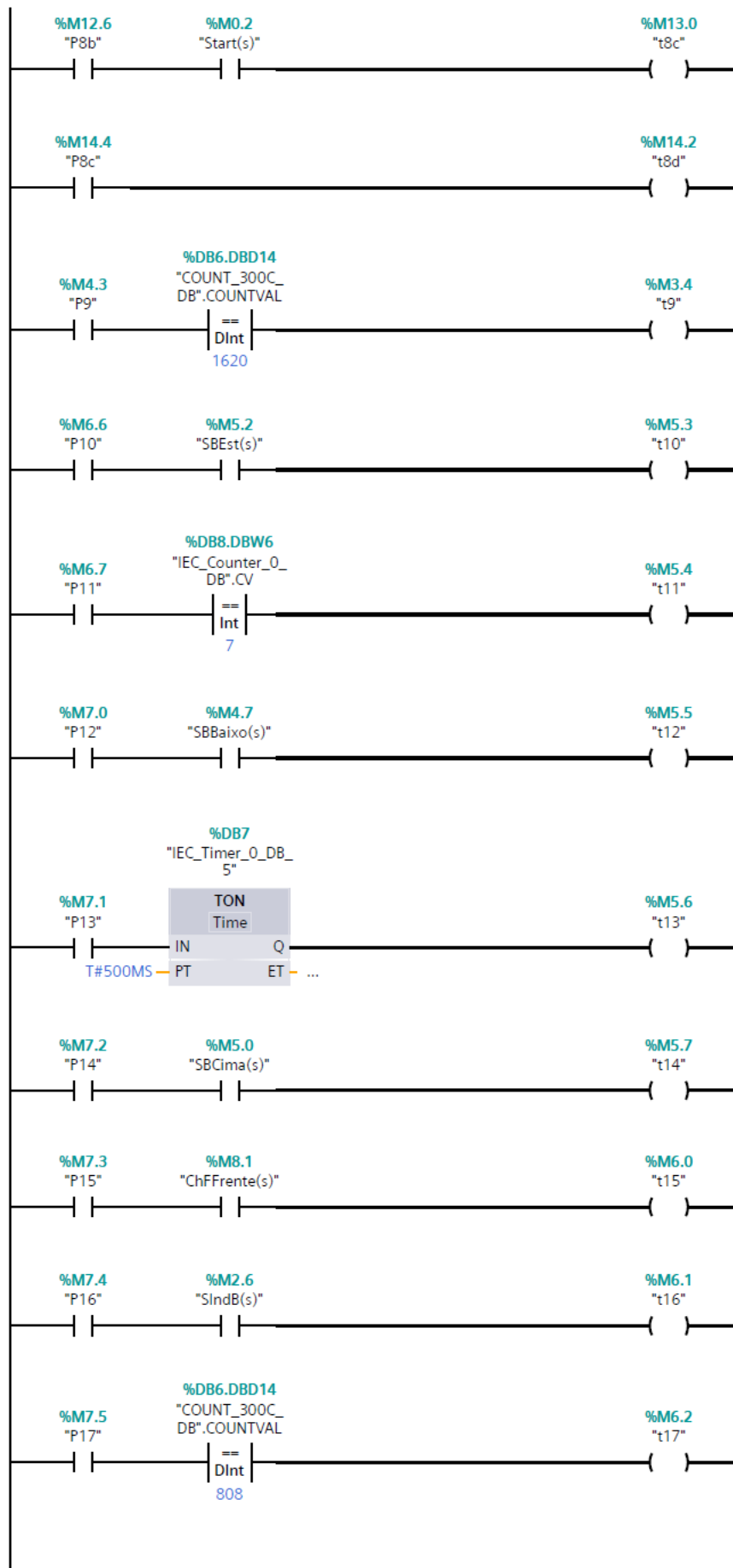


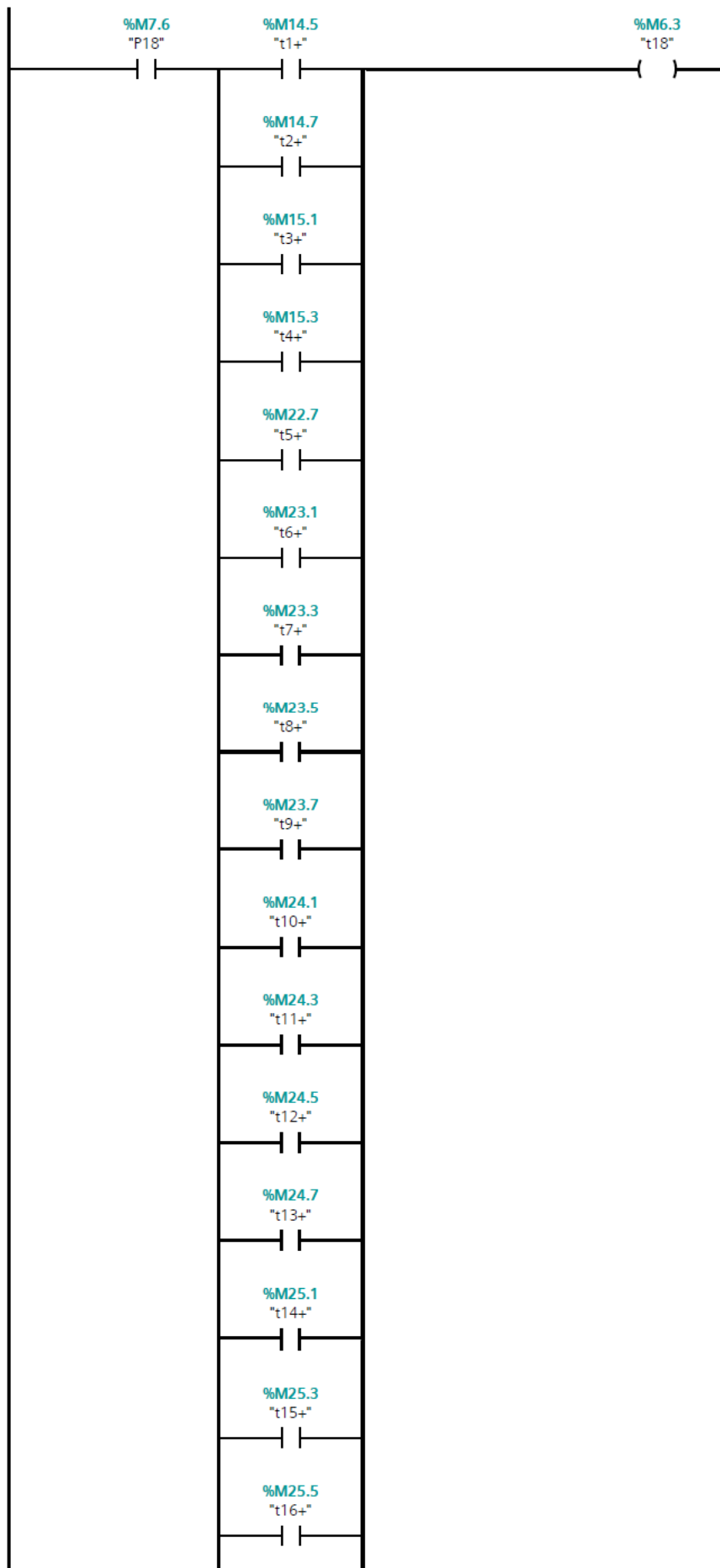


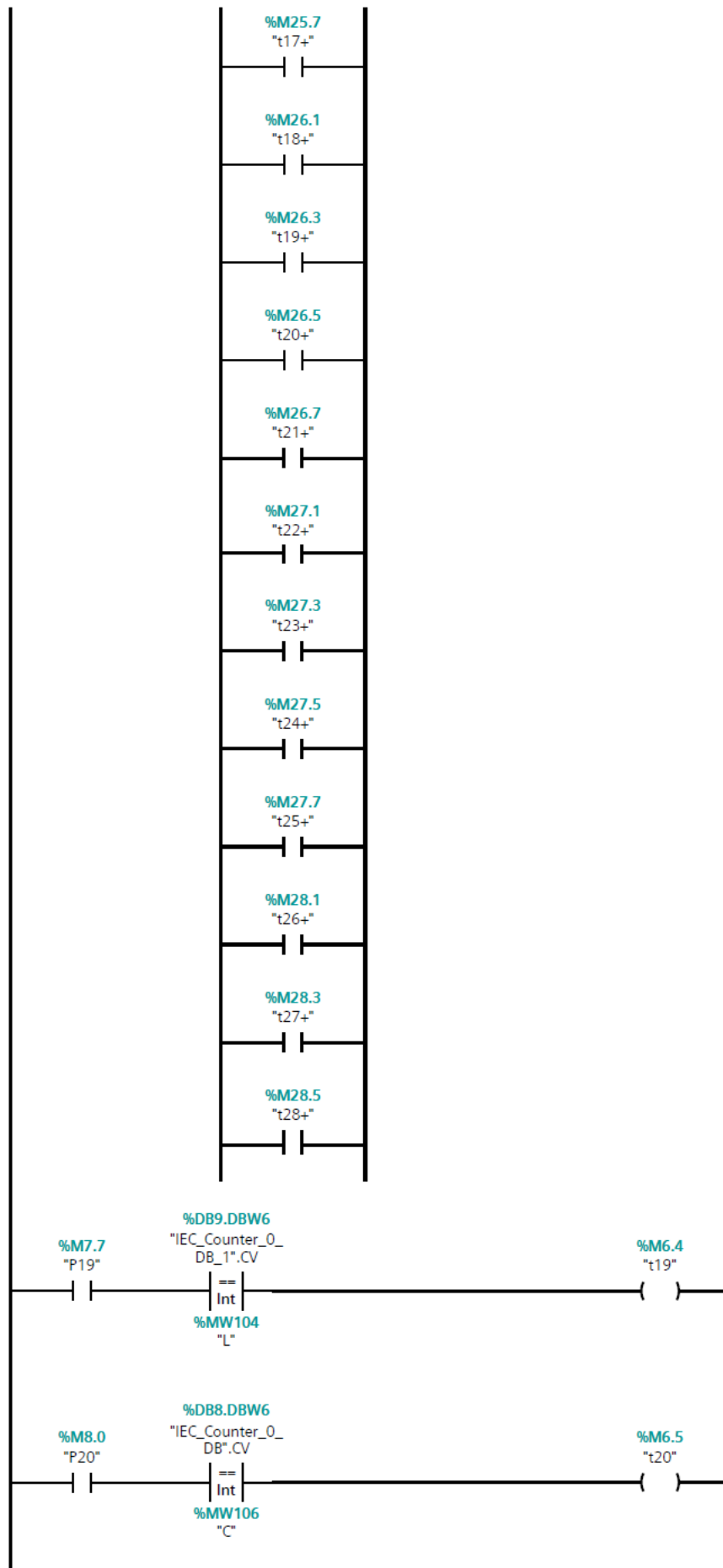
## A.3 Módulo das Condições

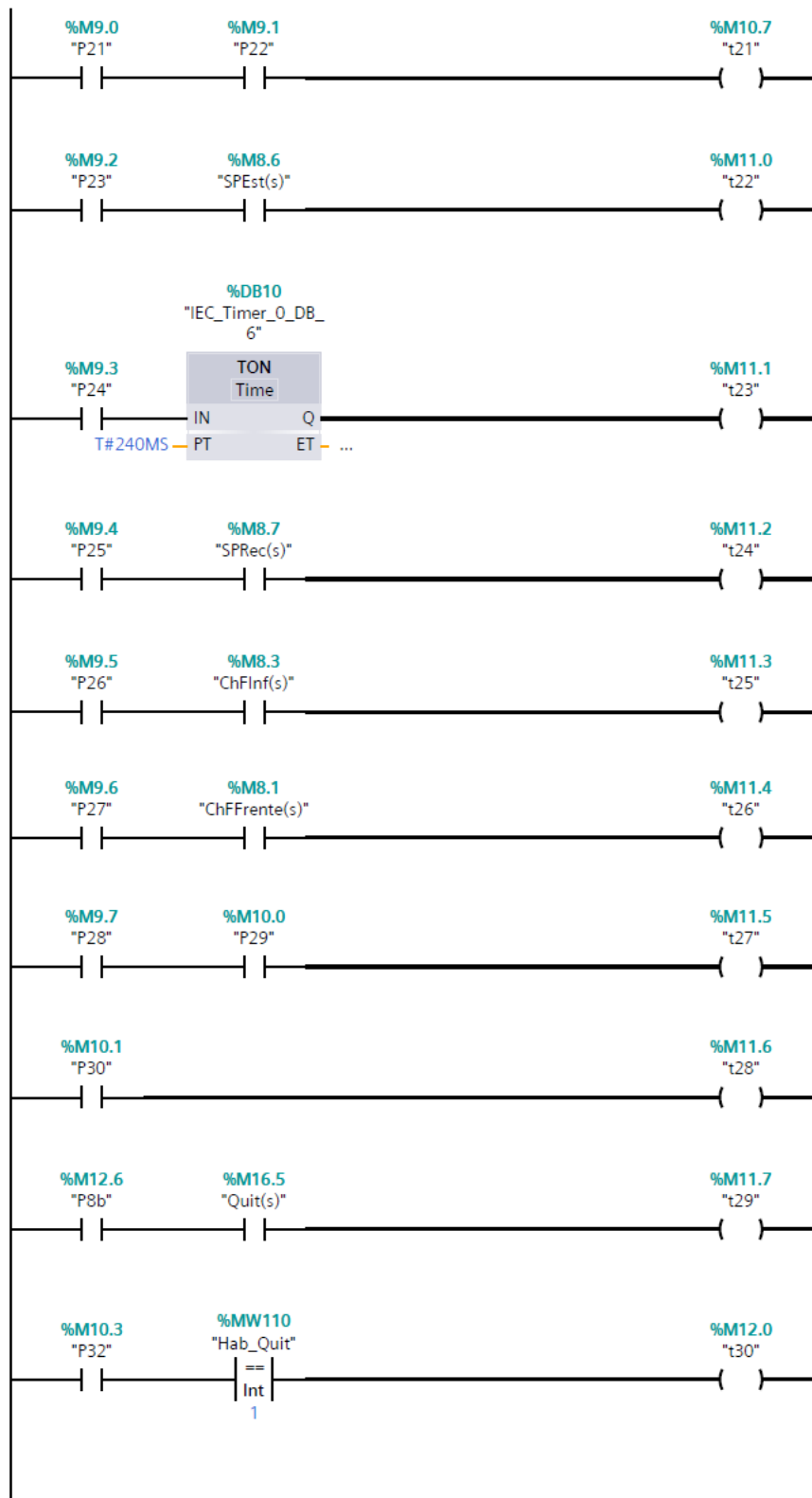


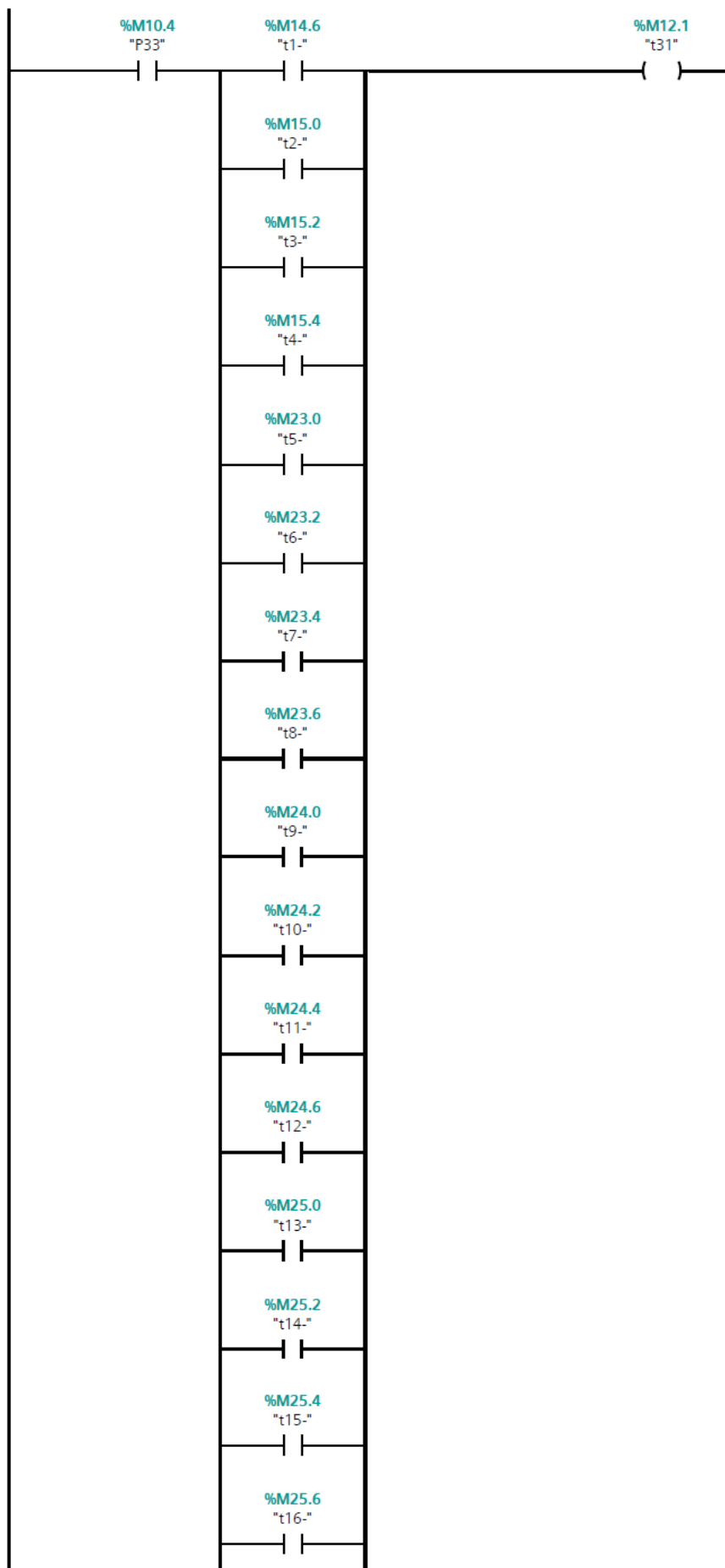


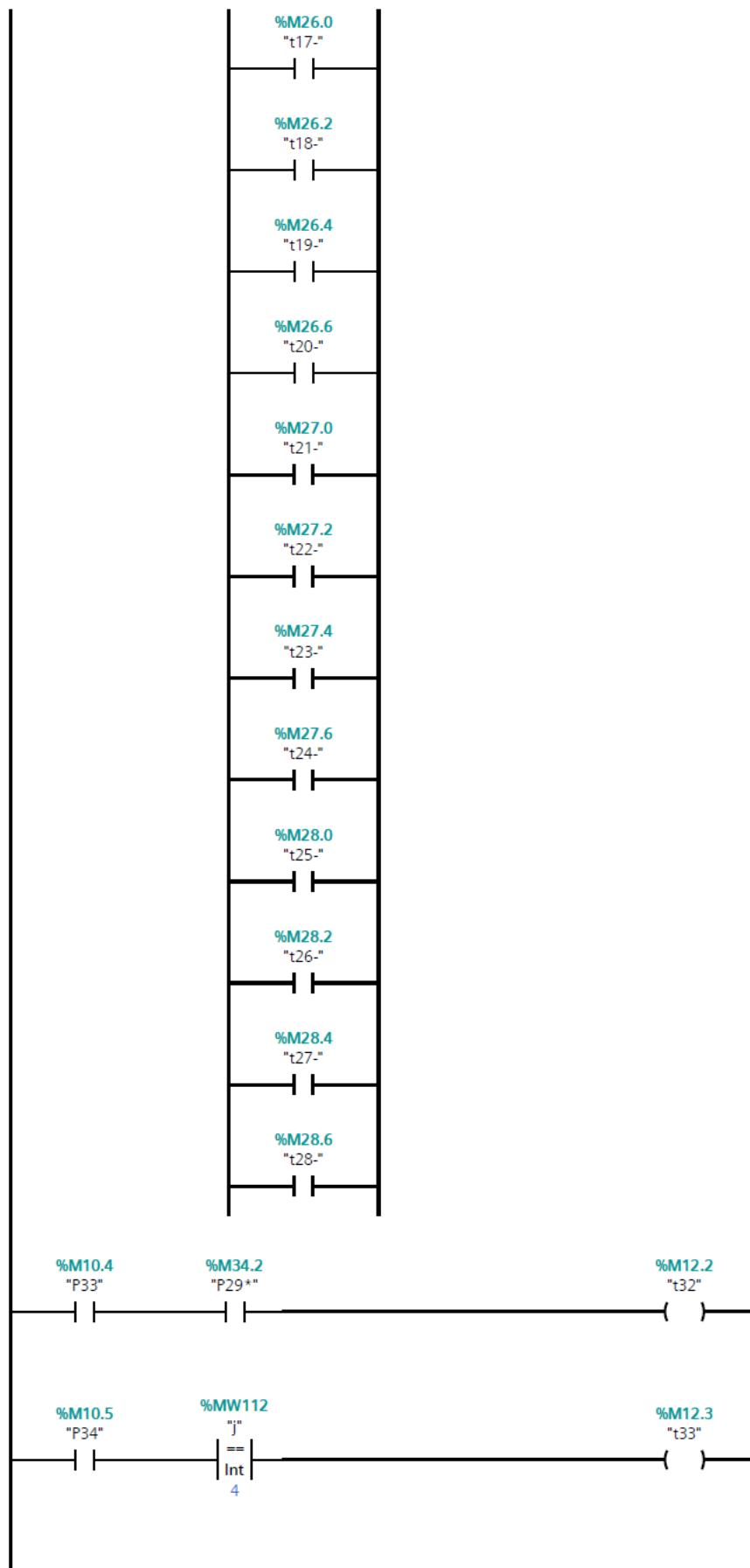


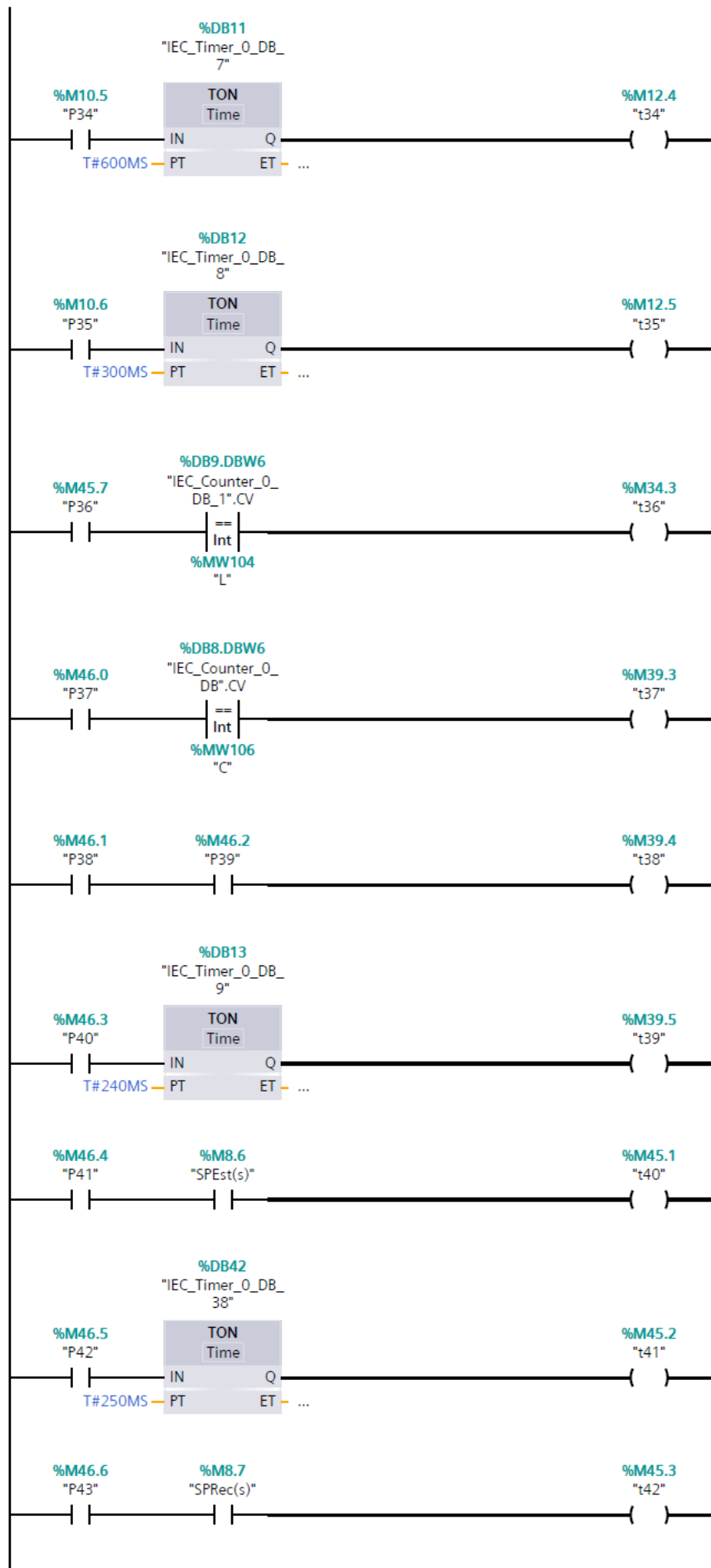


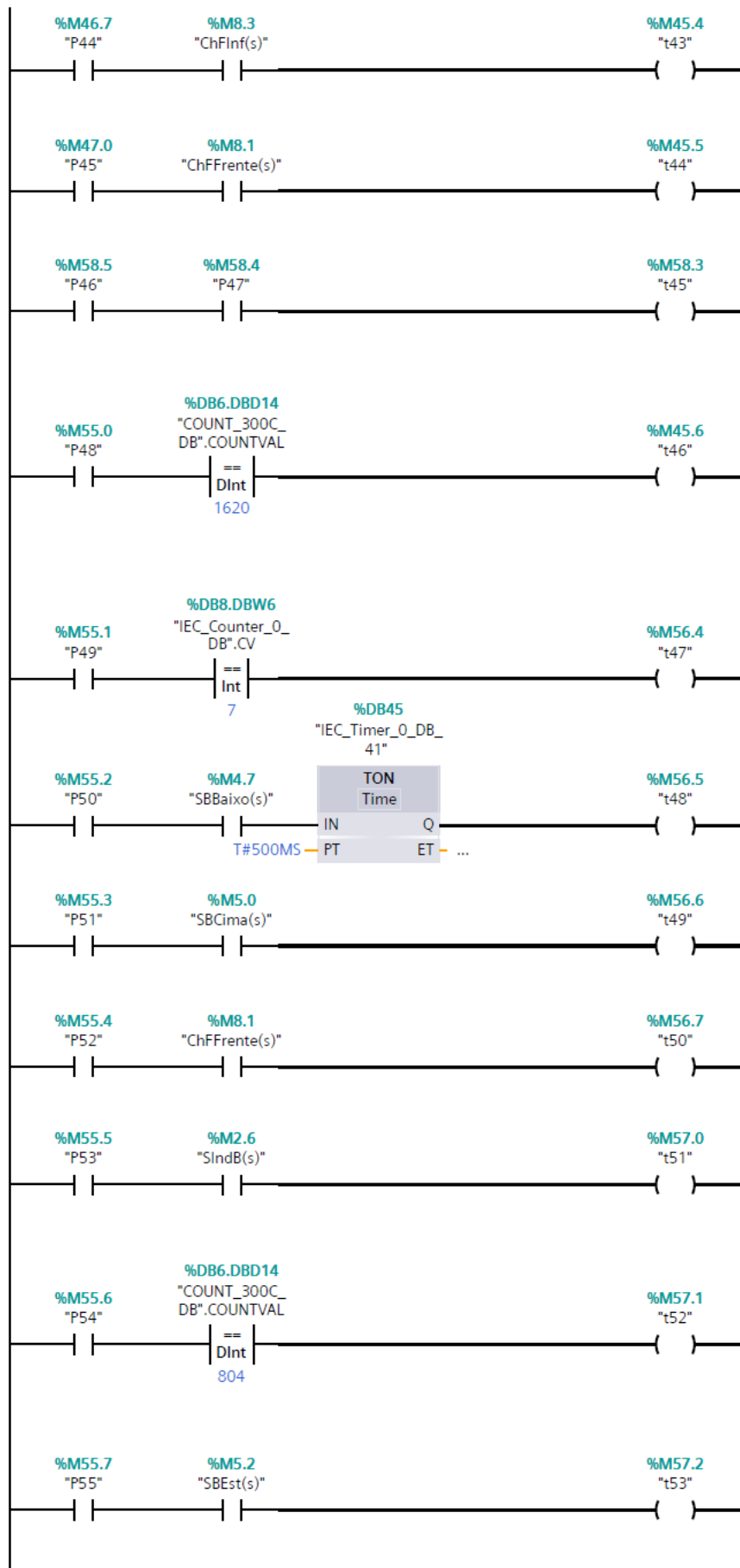


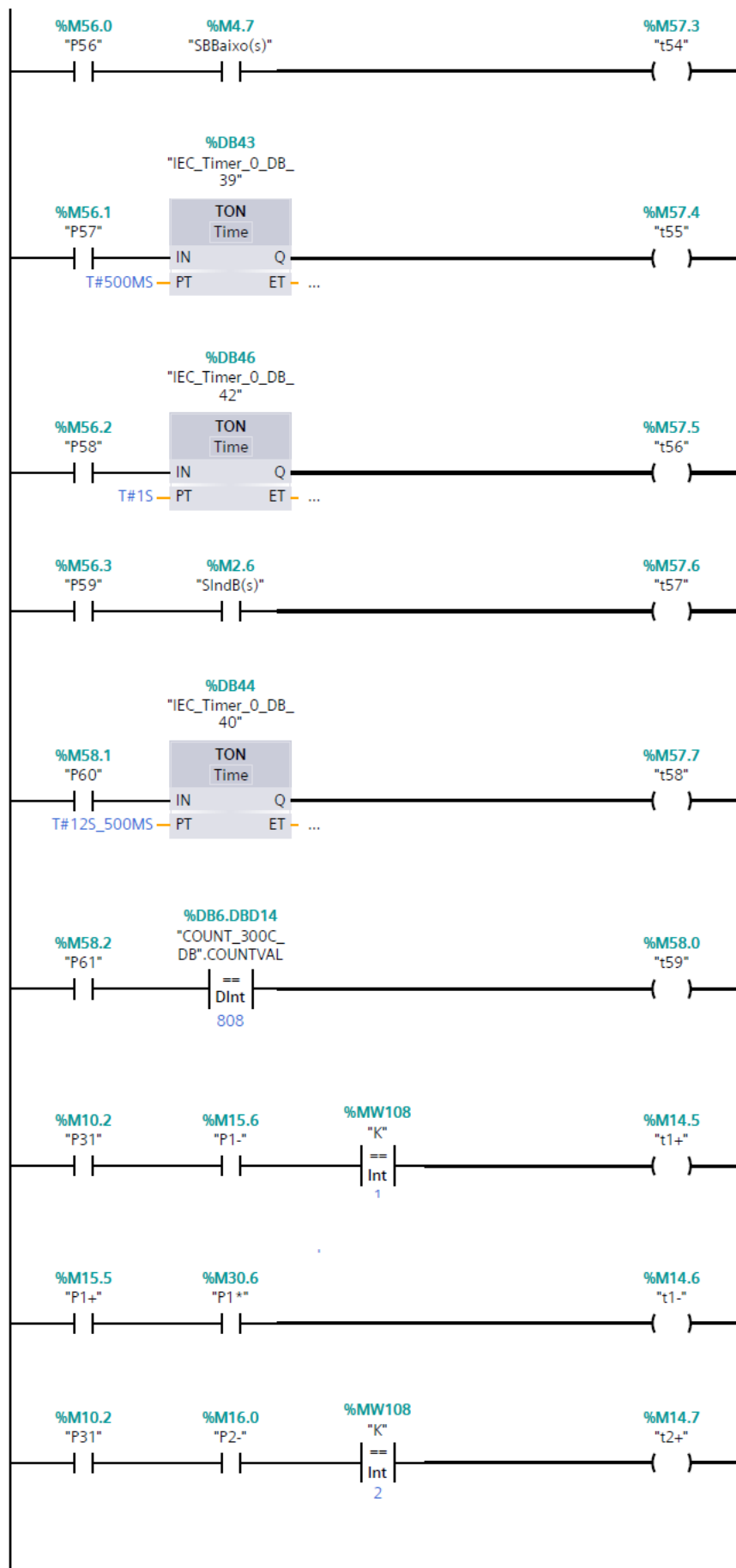


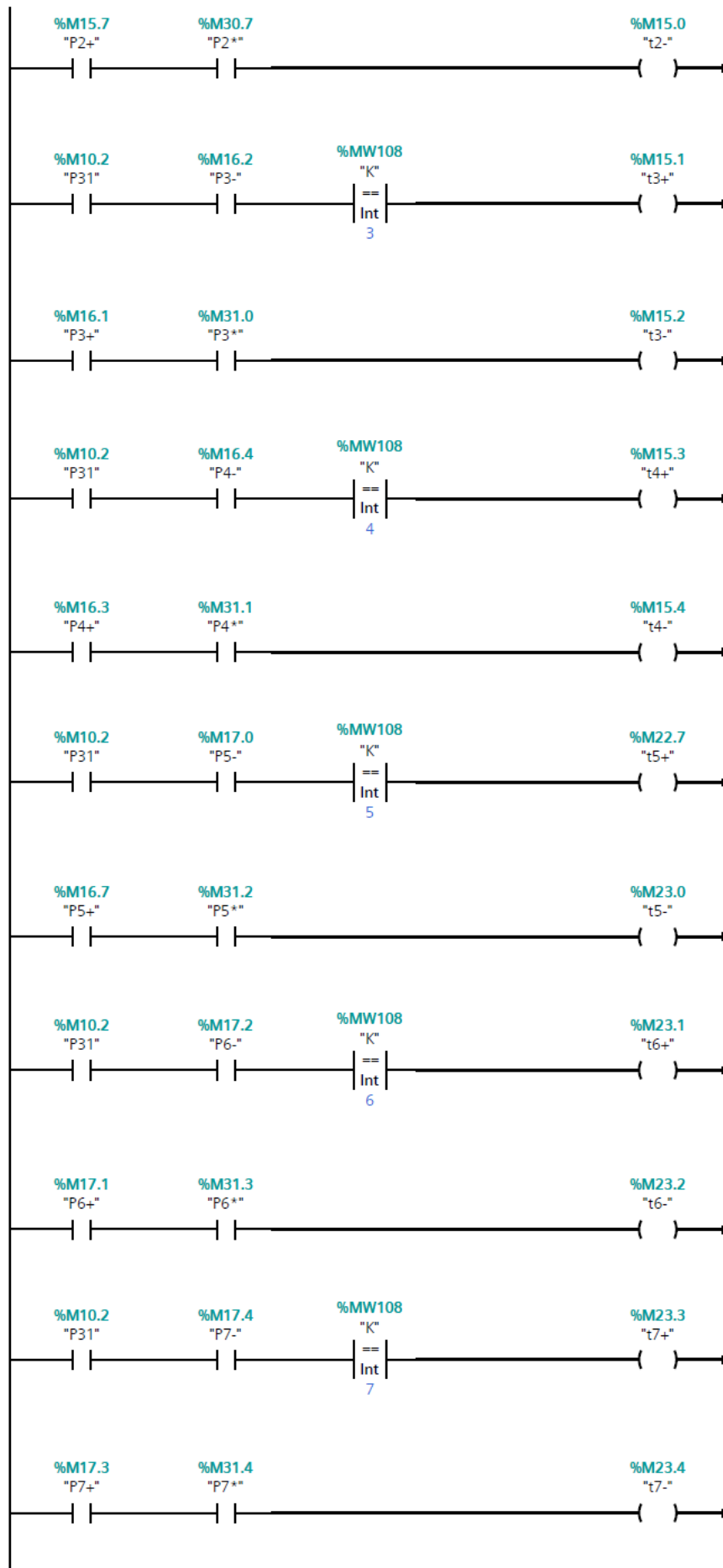


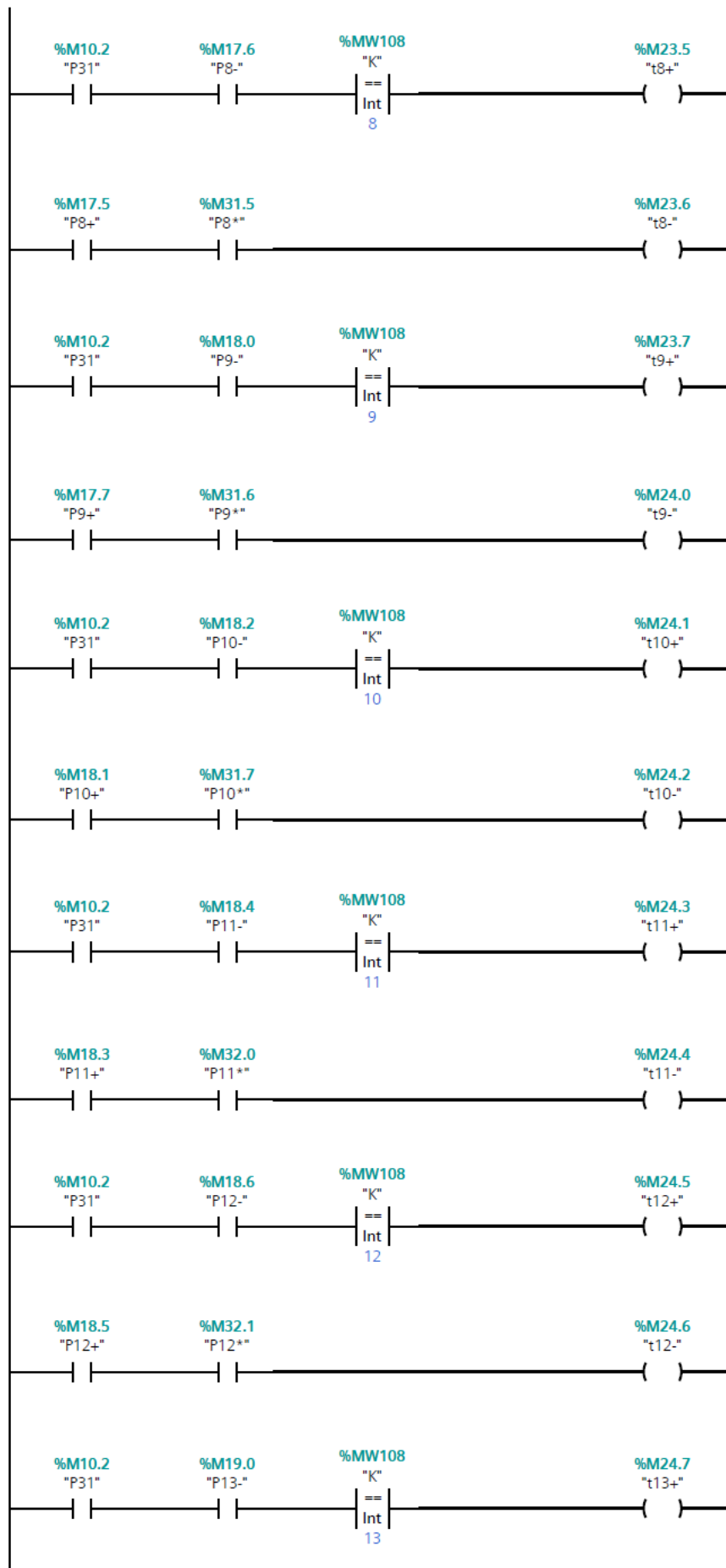


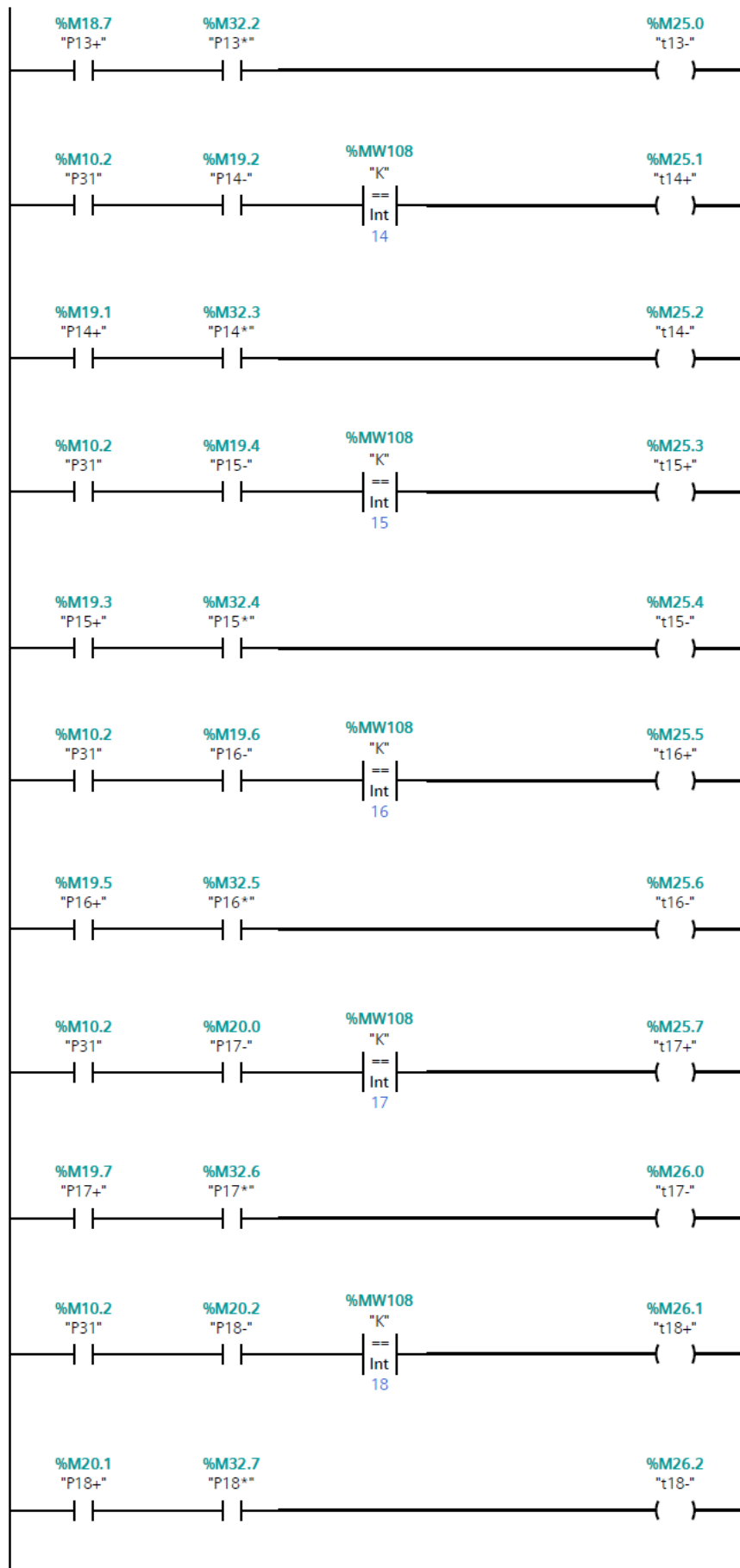


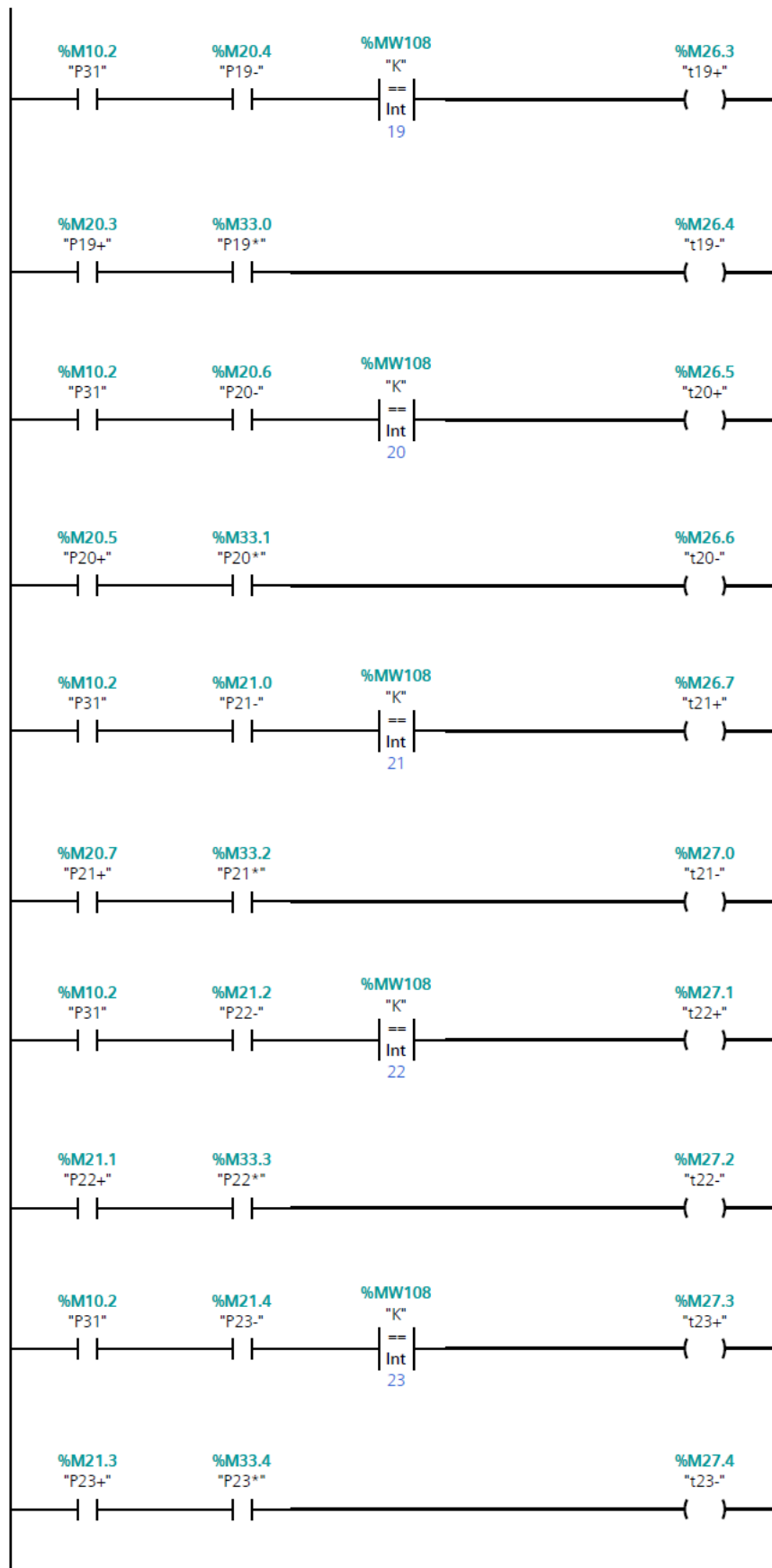


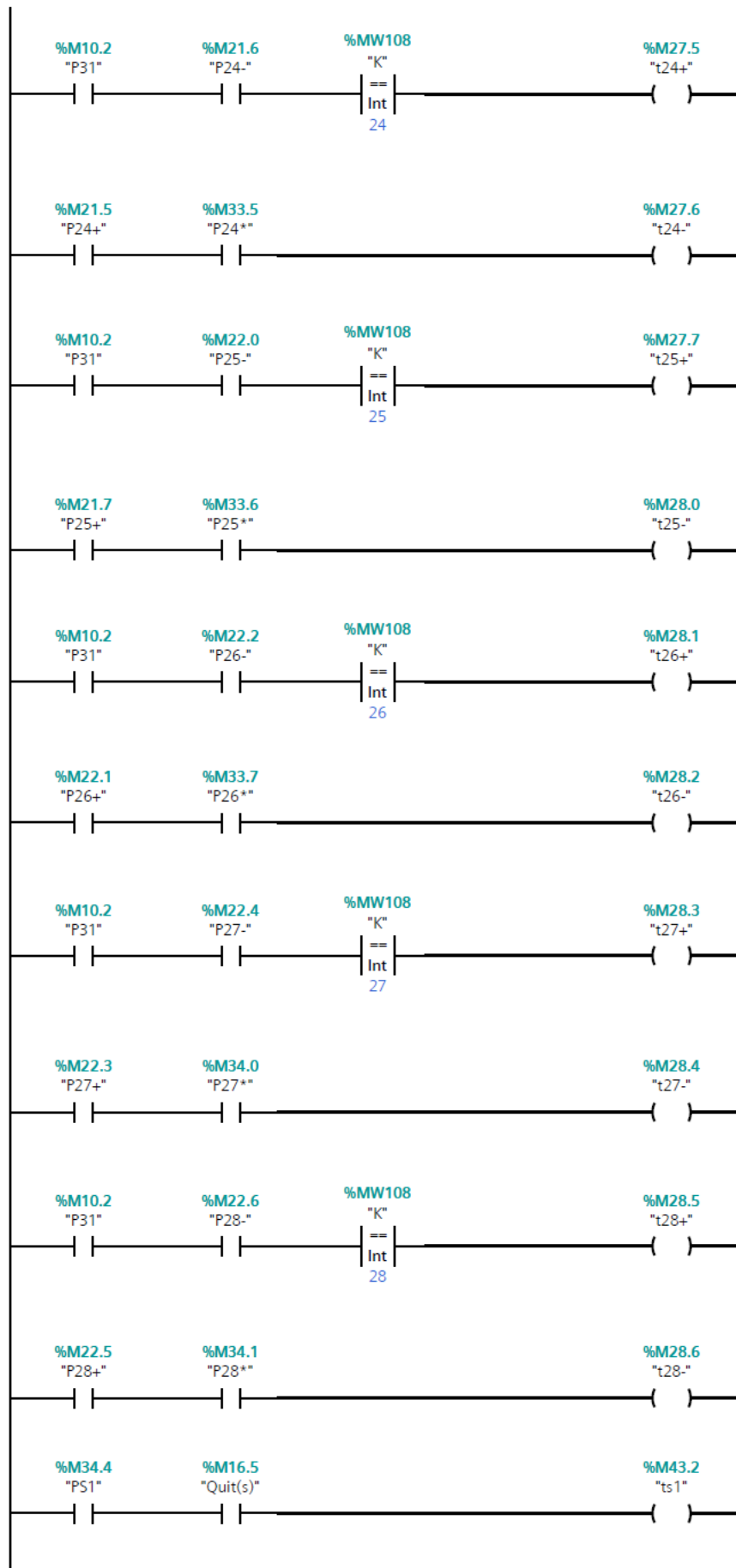


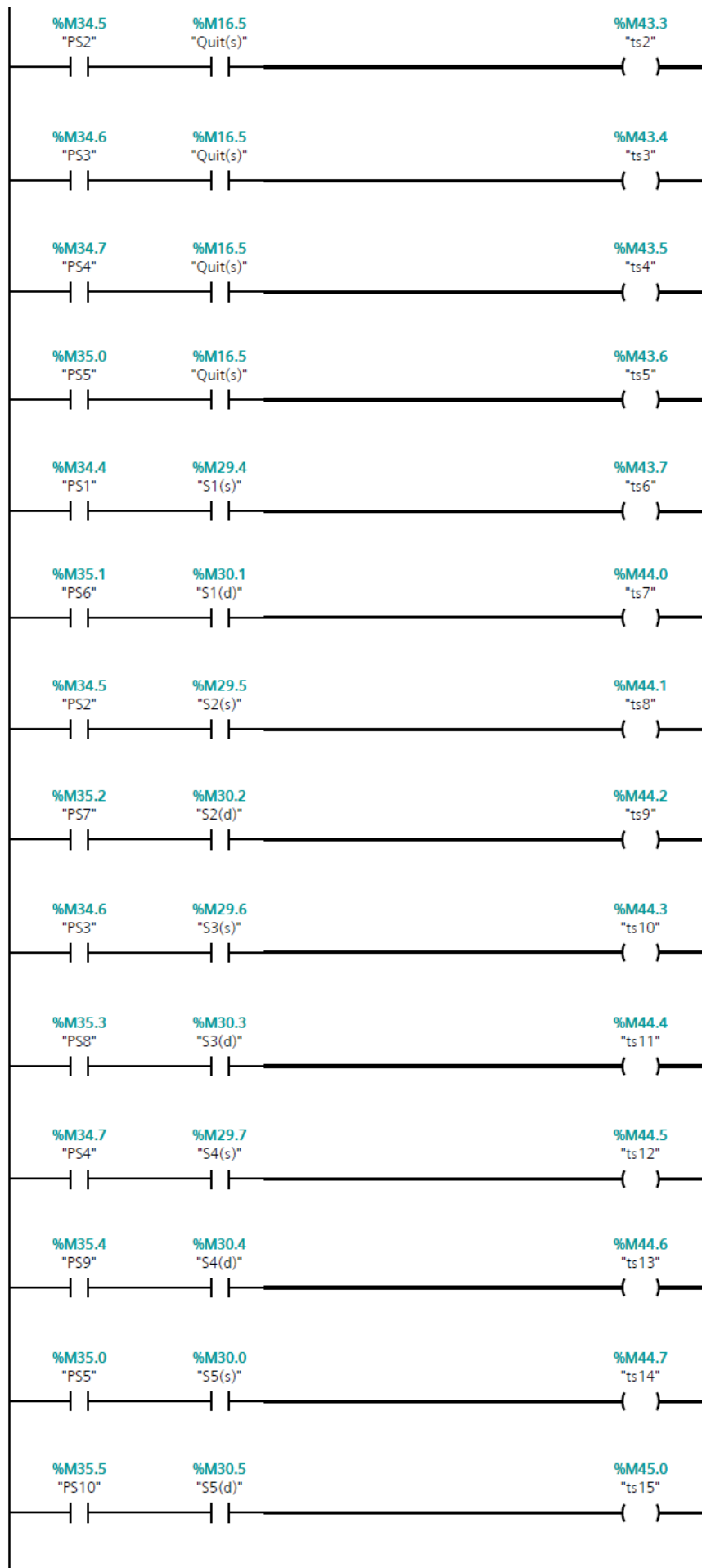


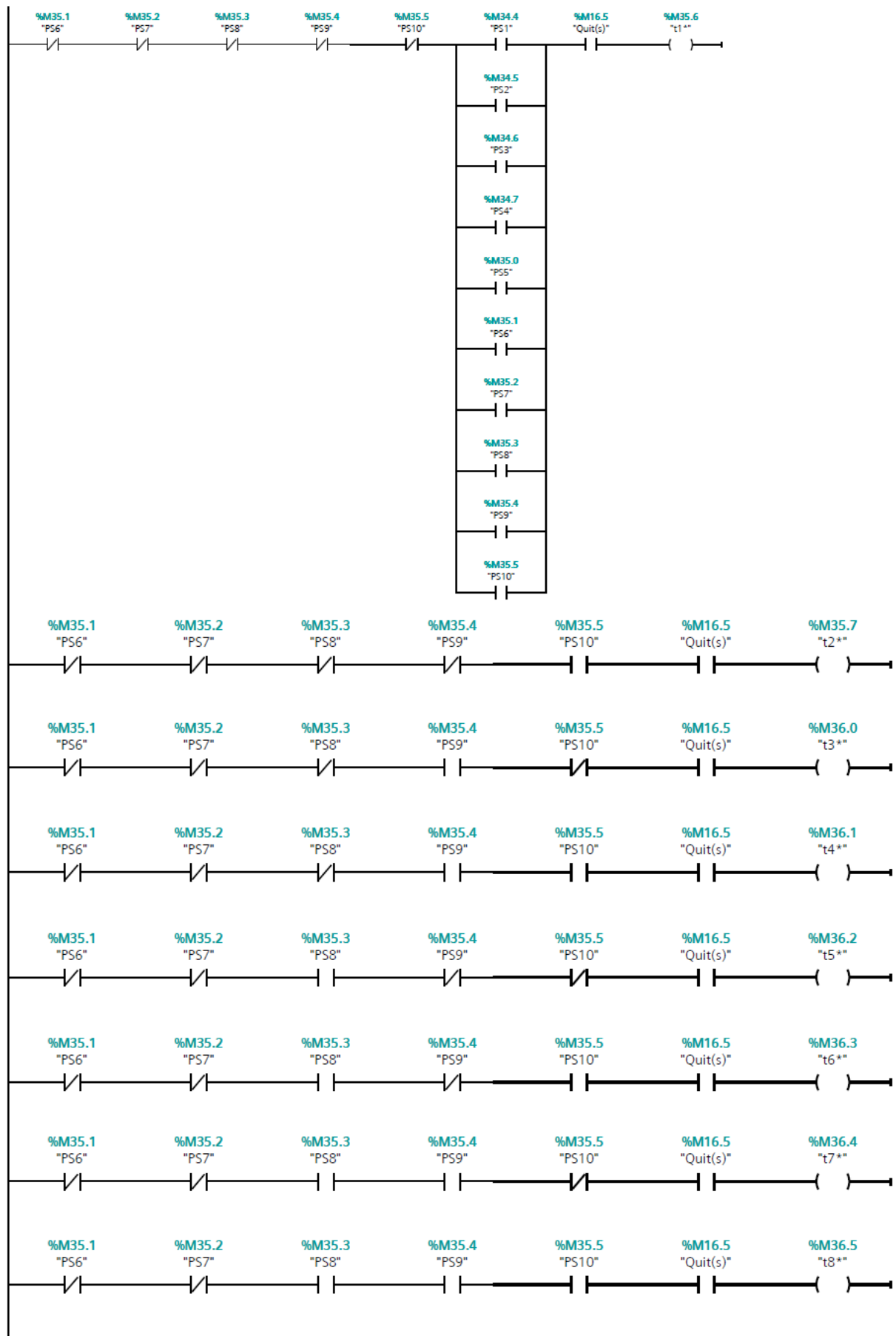


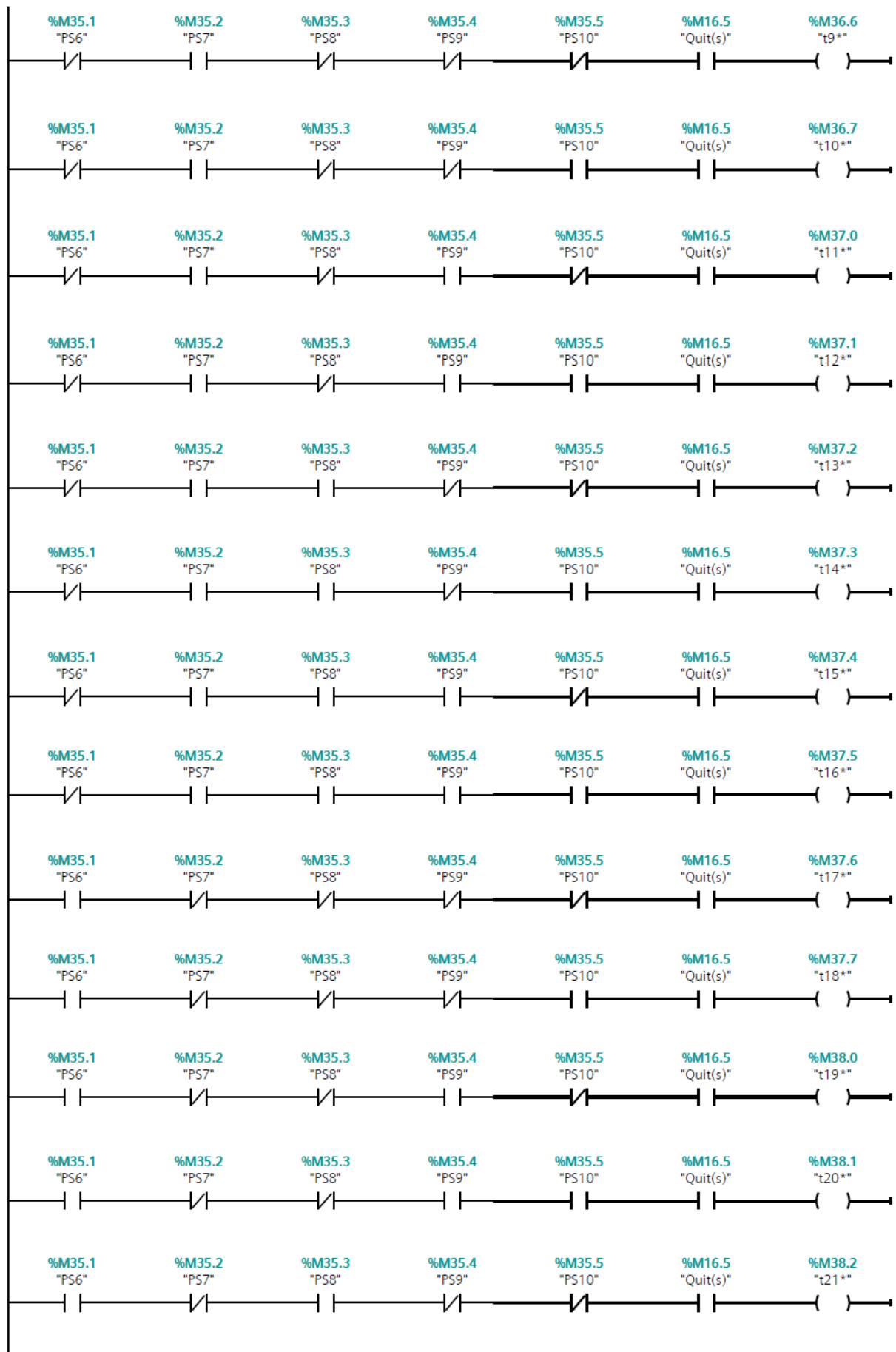


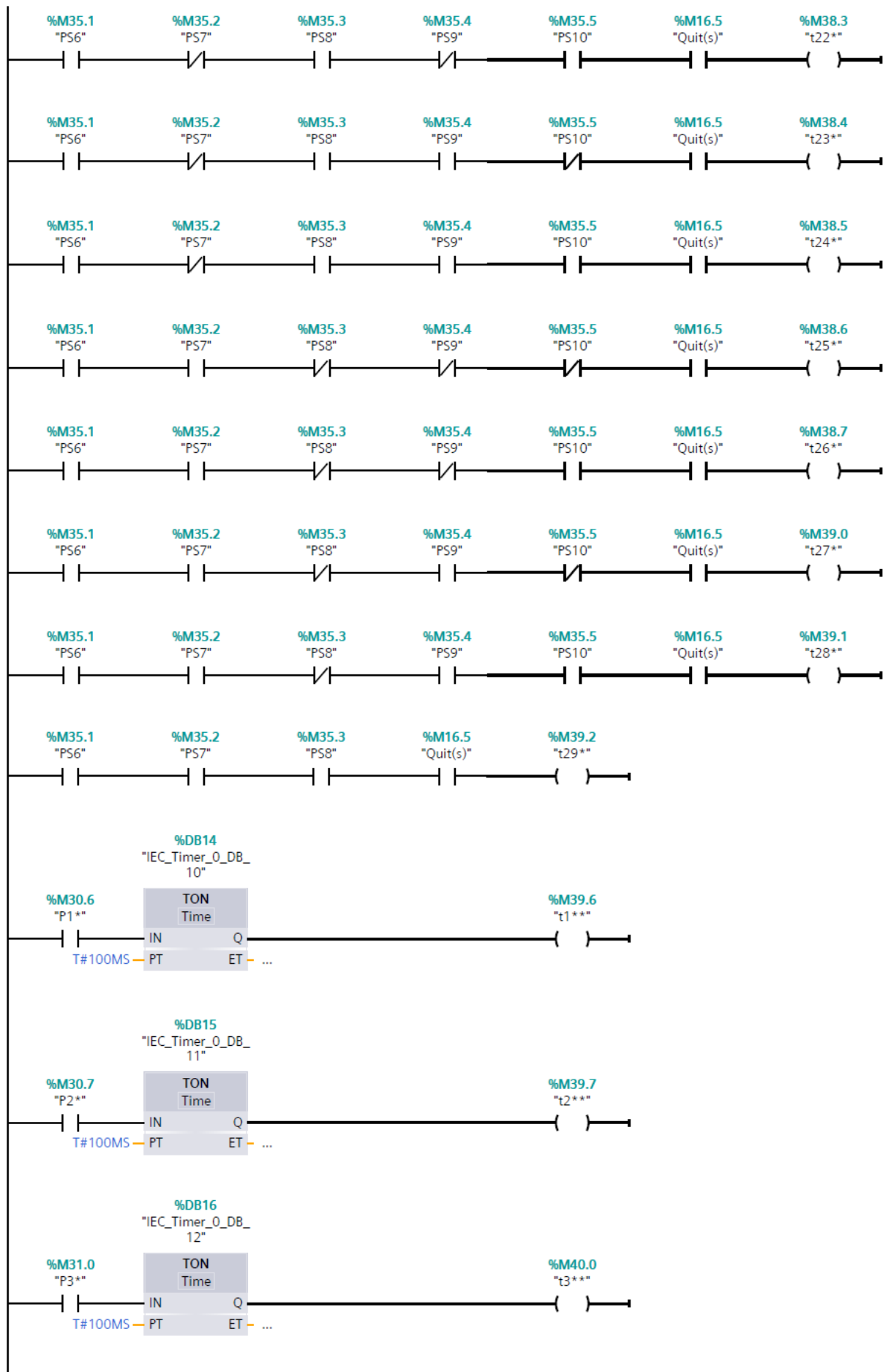


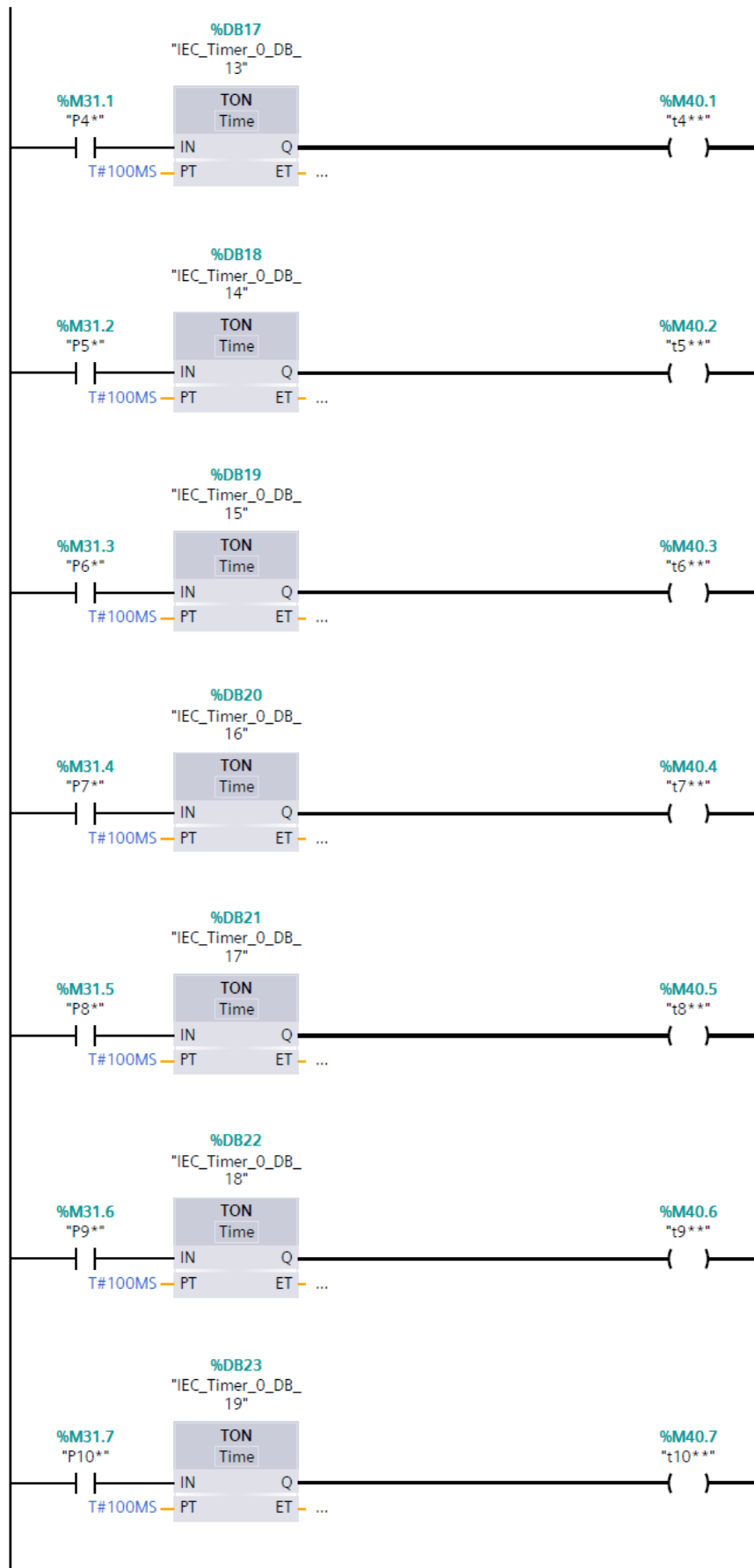


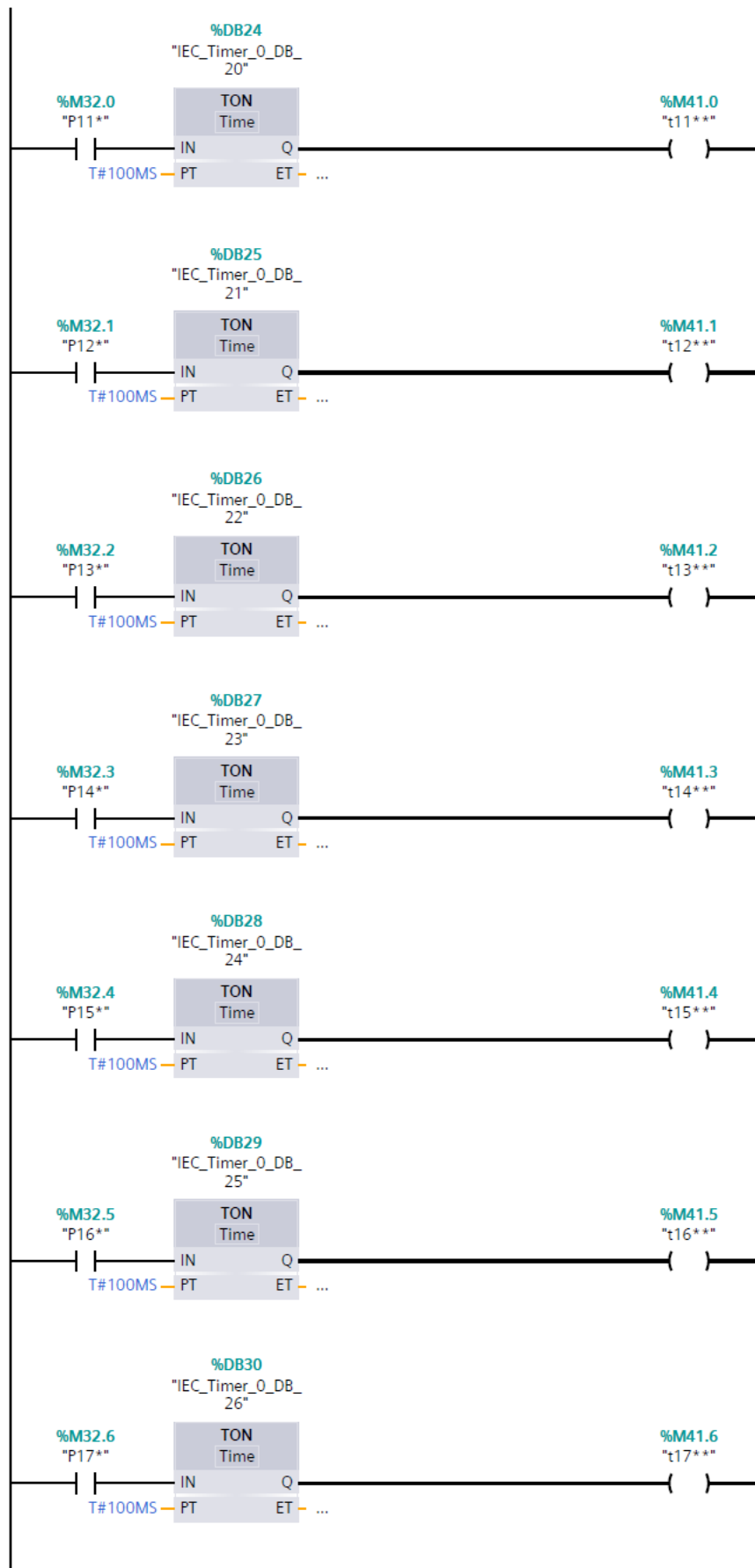


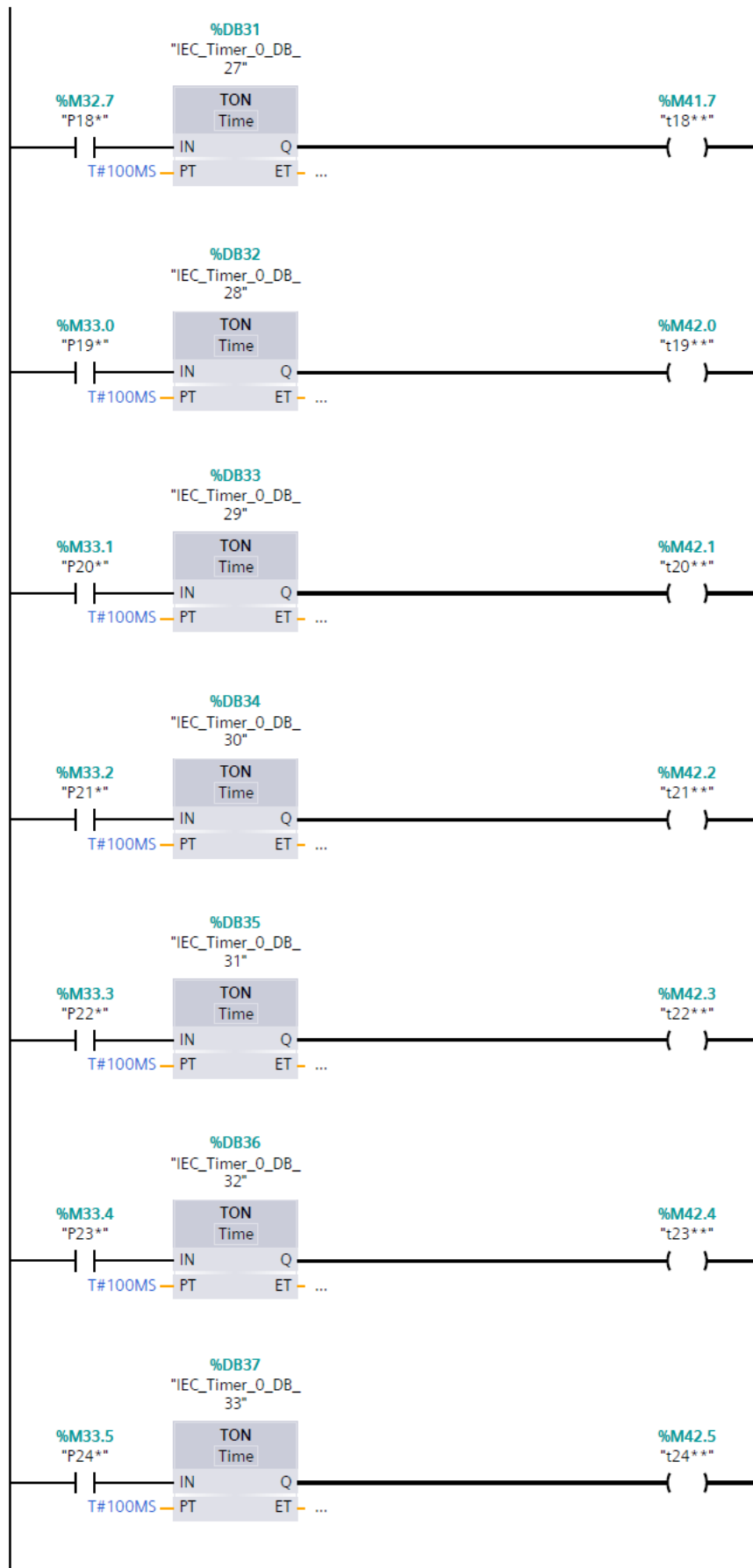


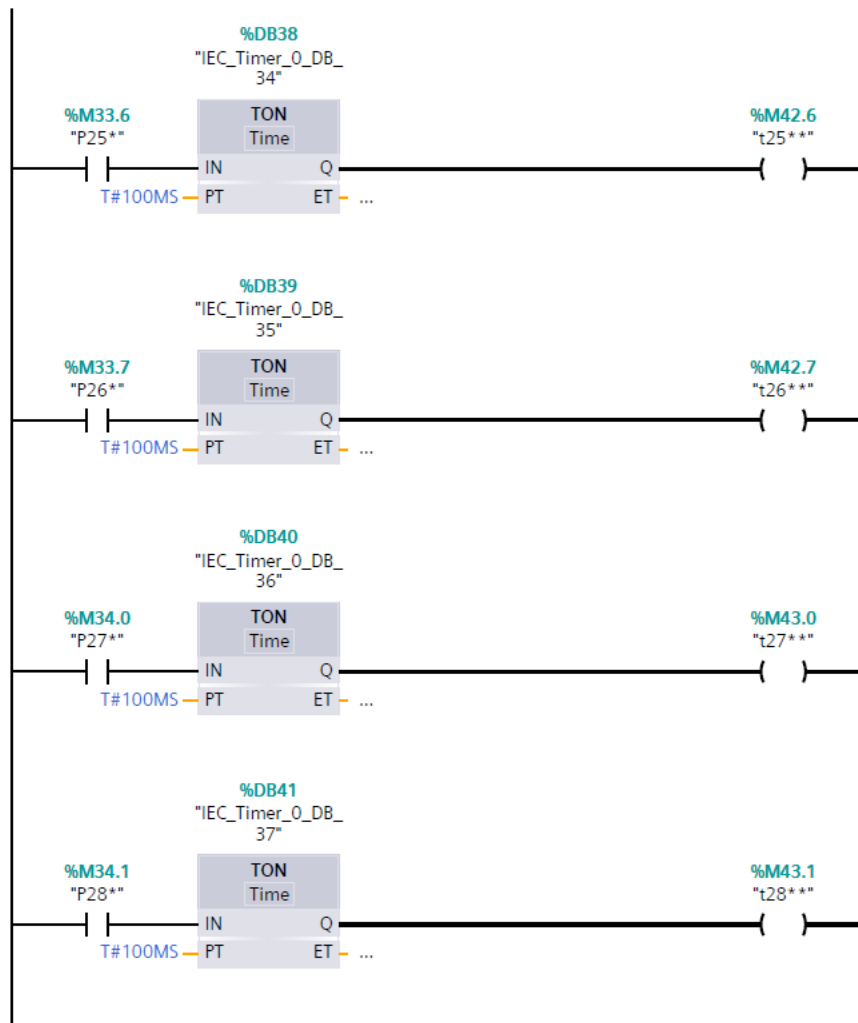




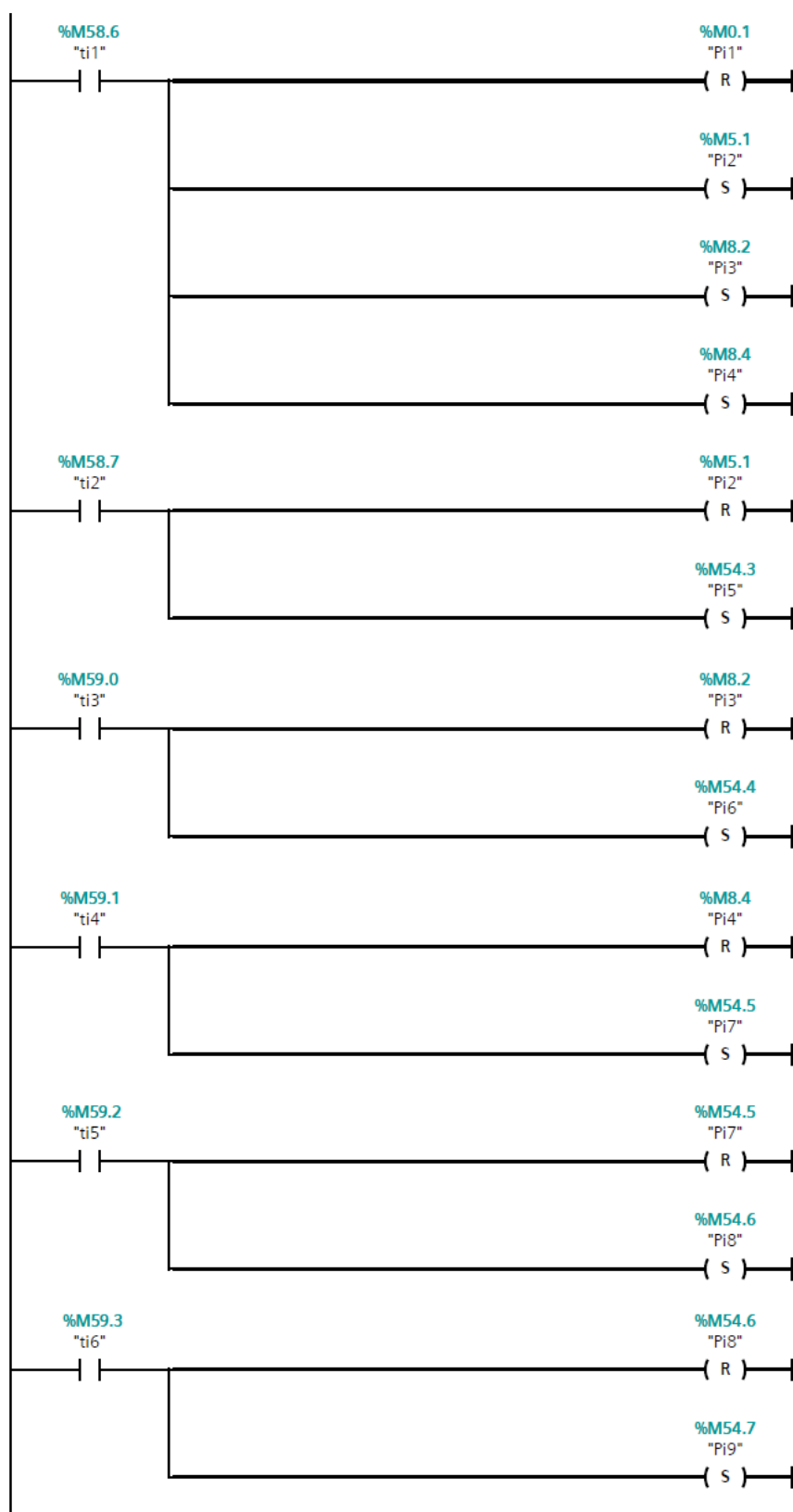


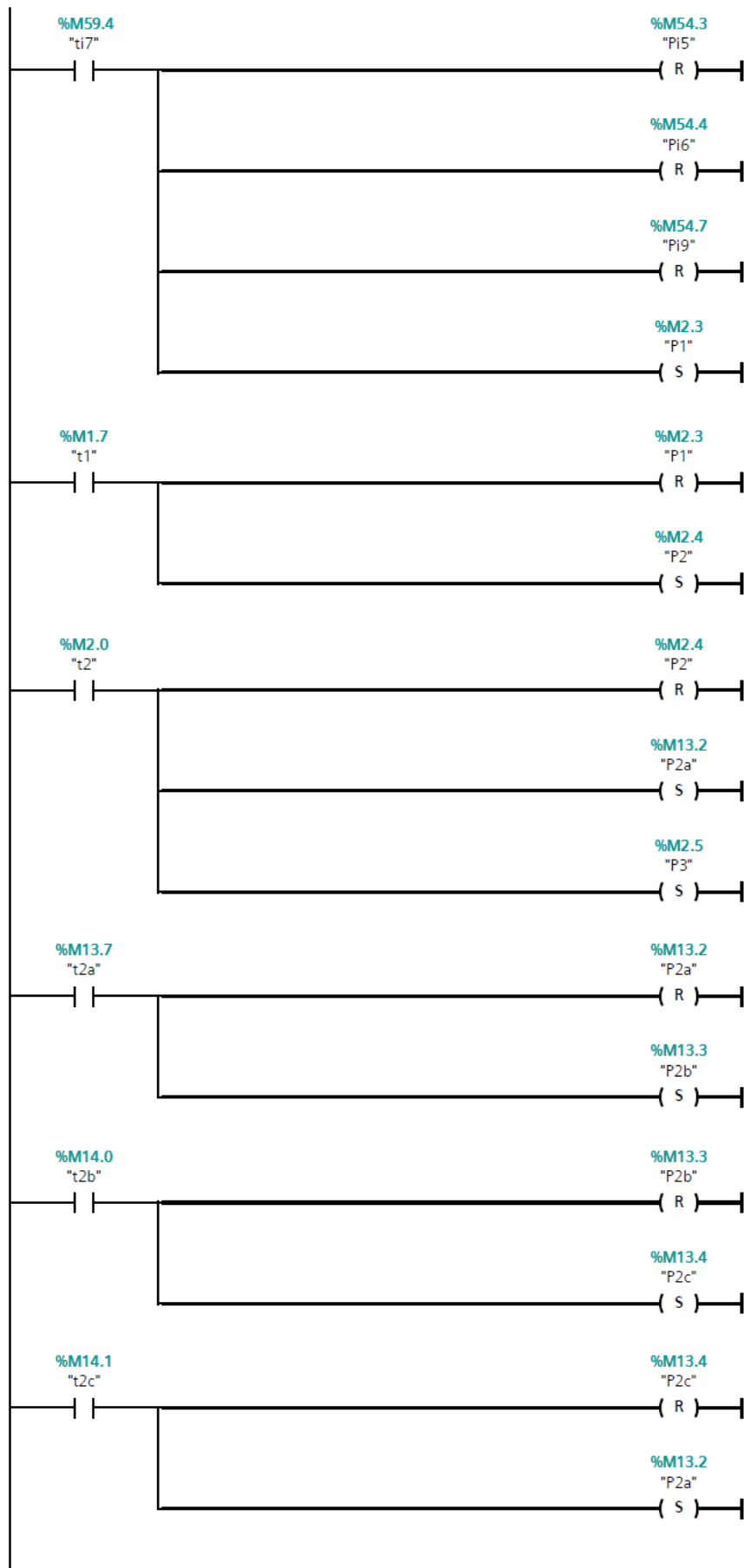


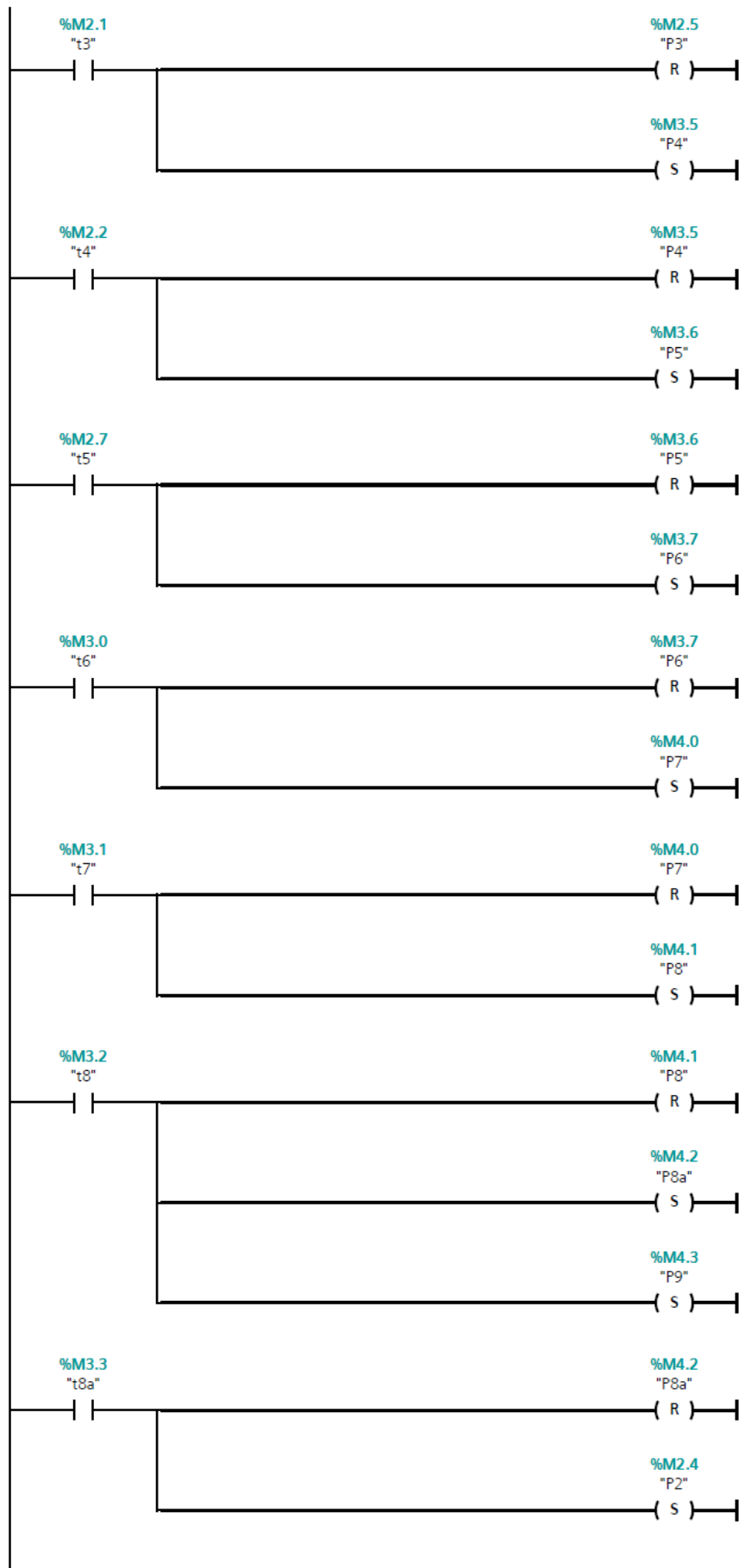


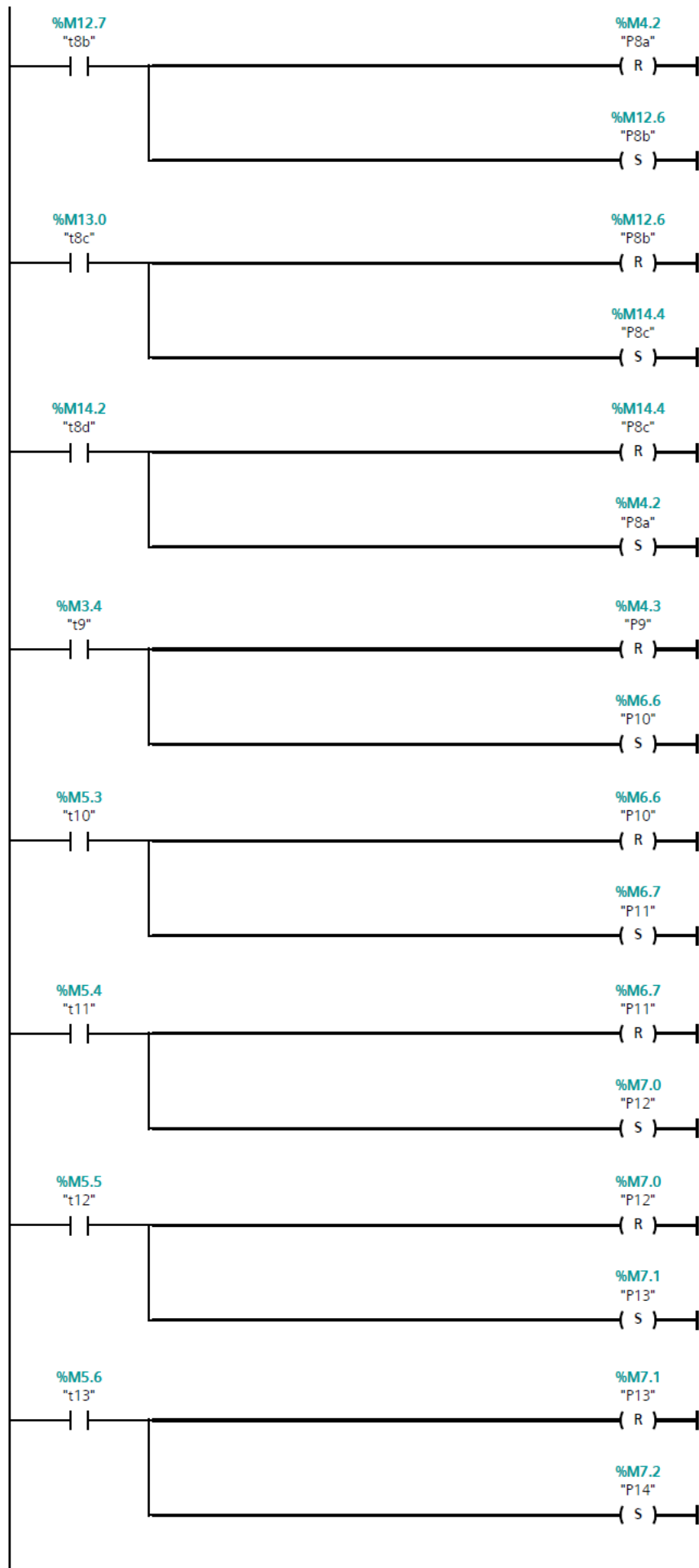


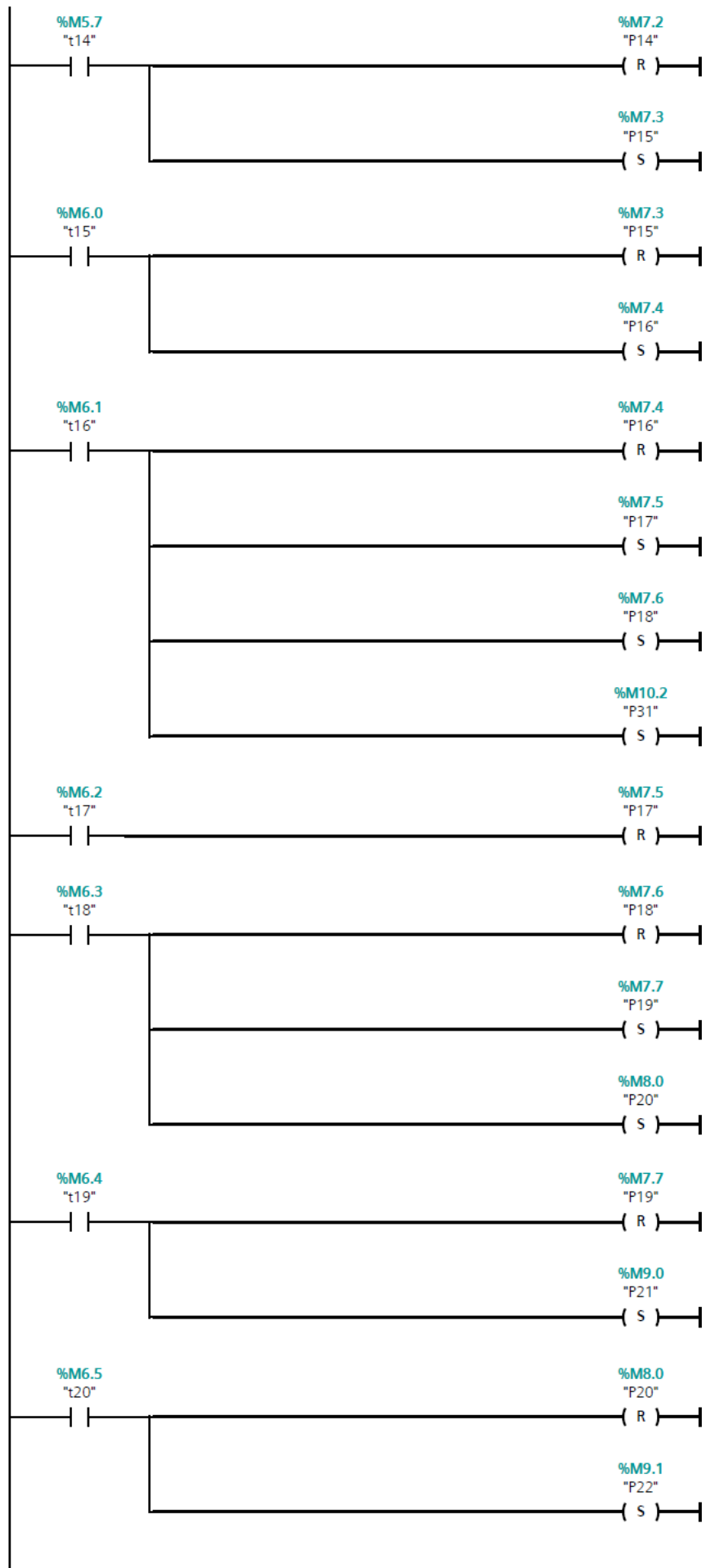
## A.4 Módulo da Dinâmica

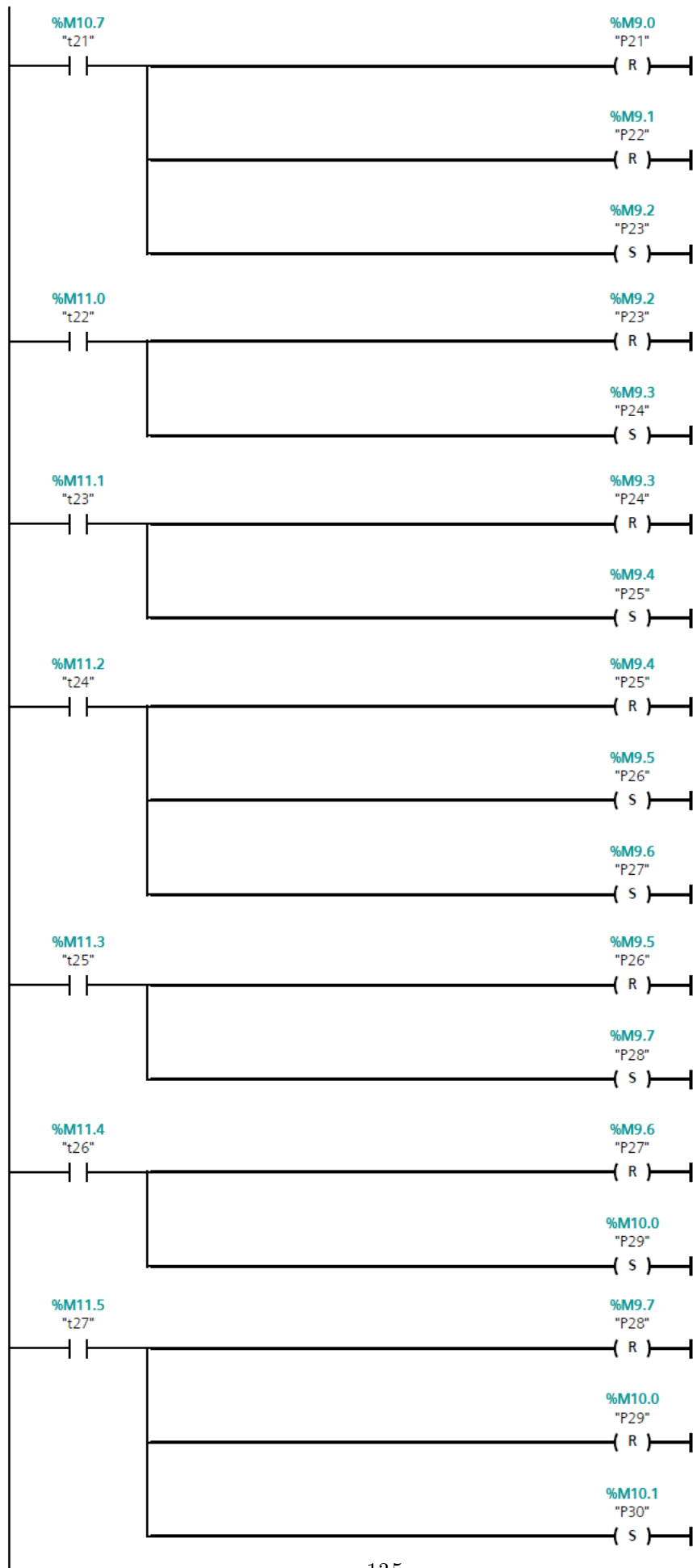


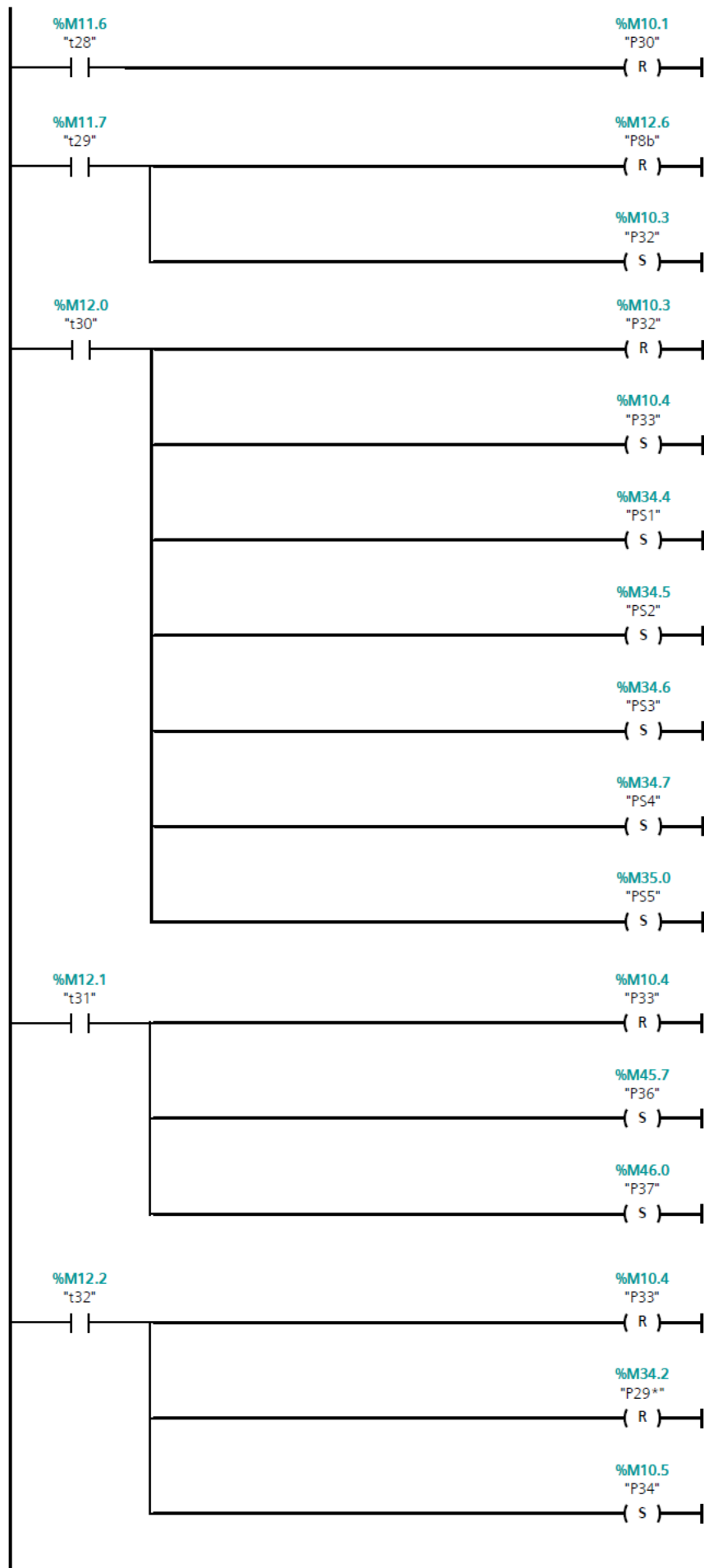


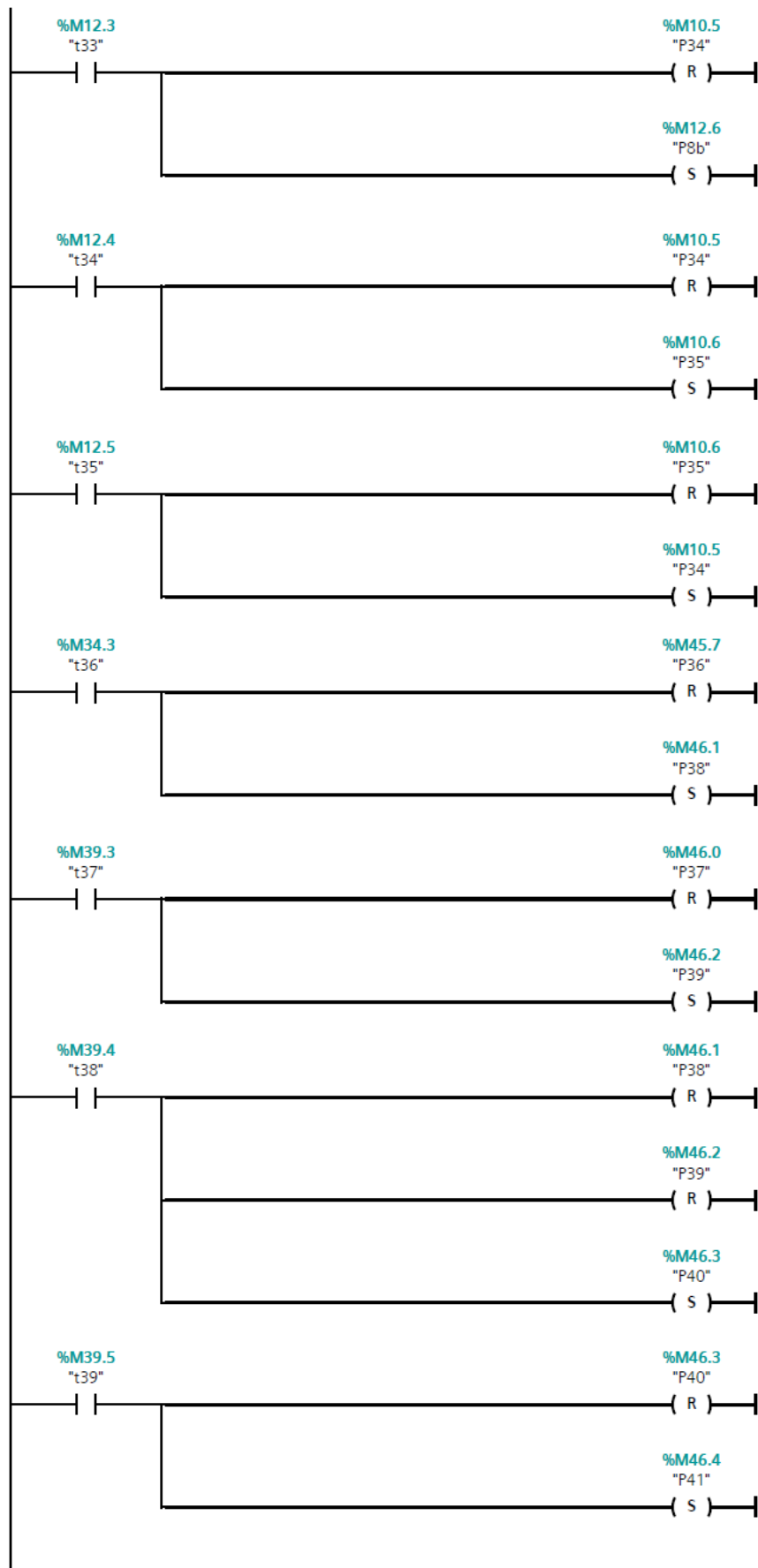


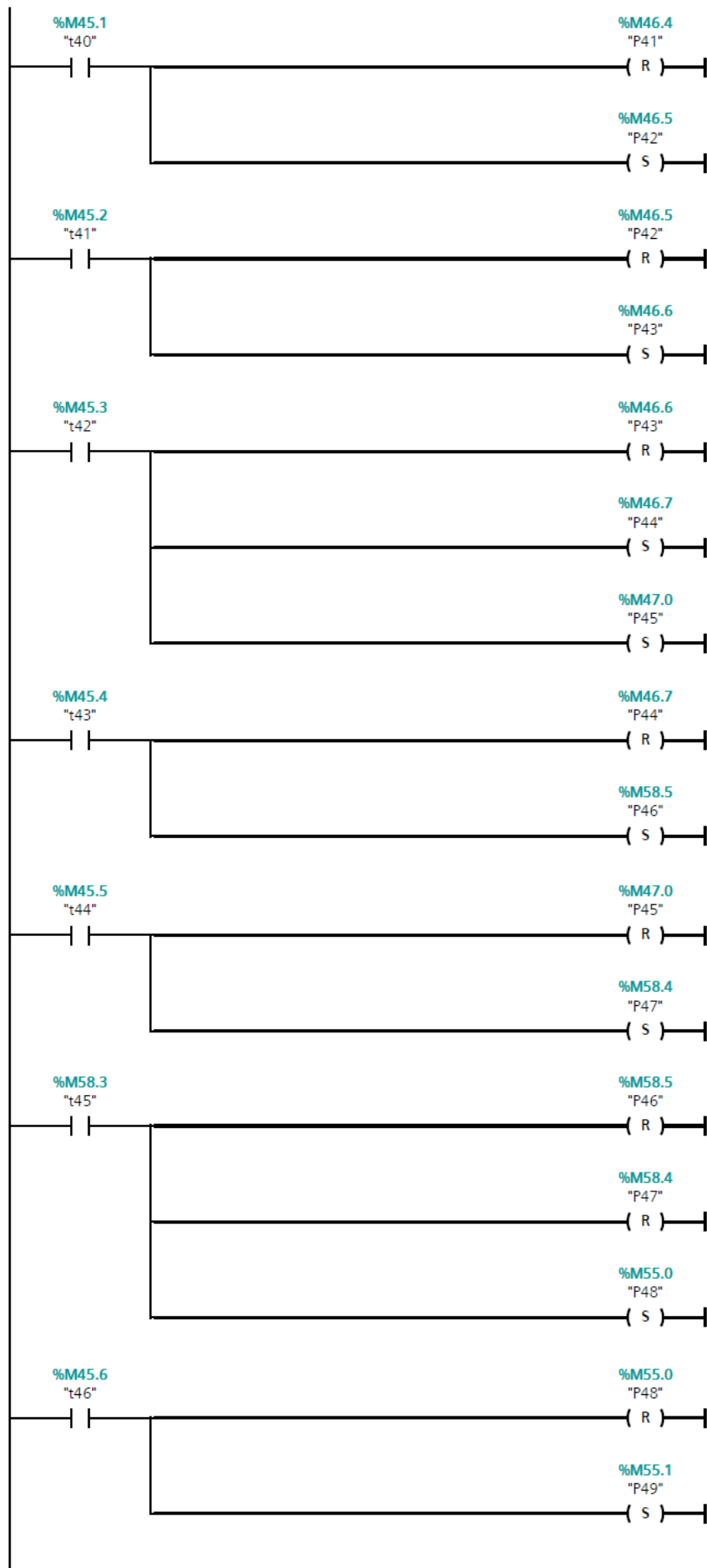


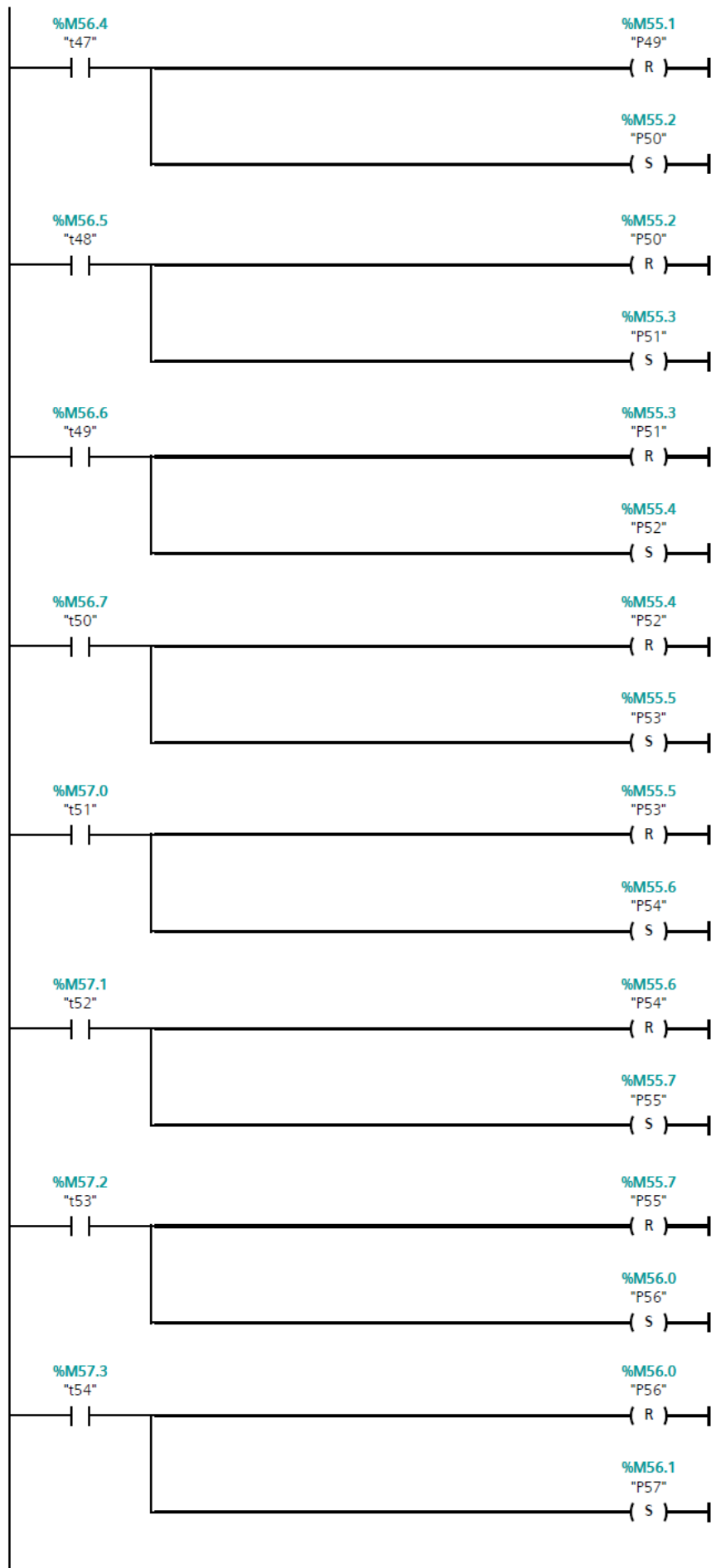


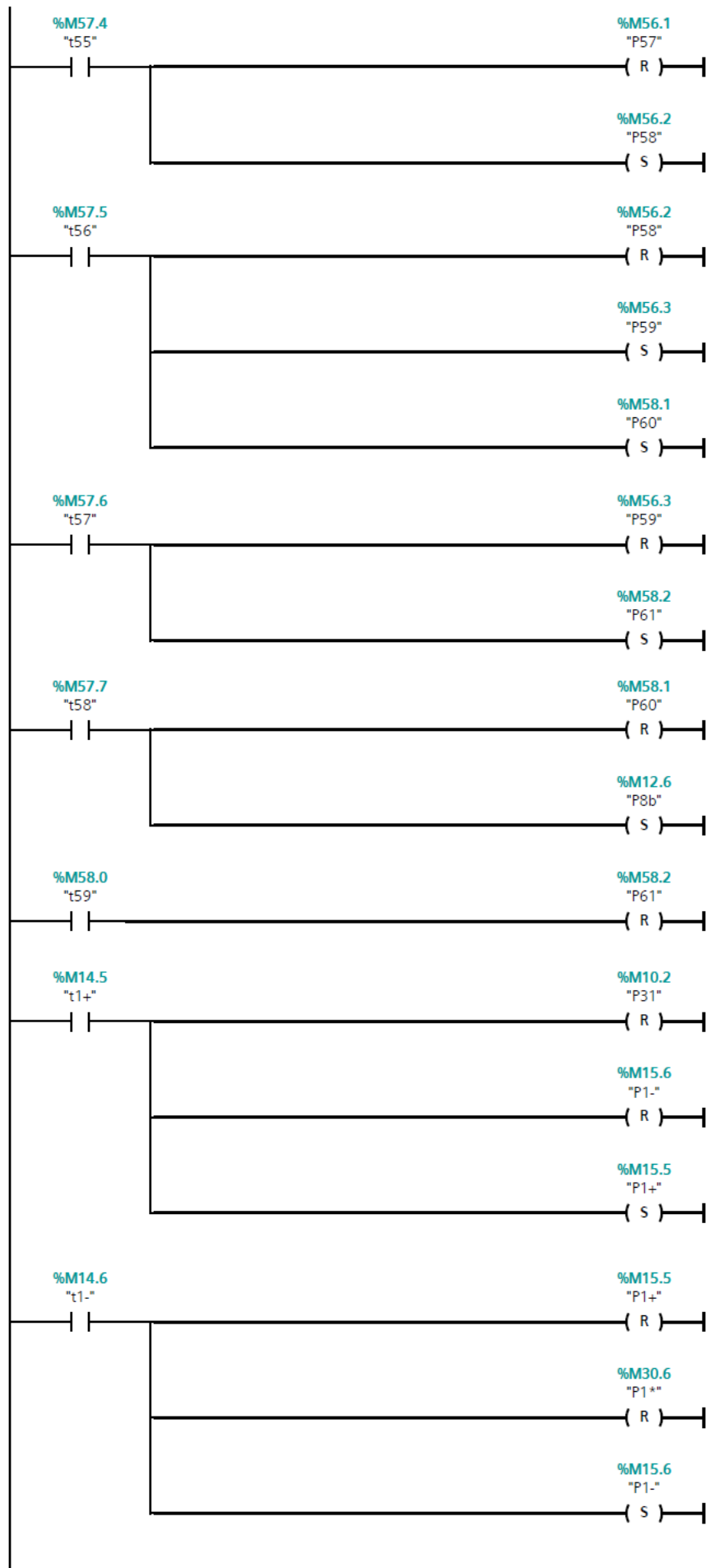


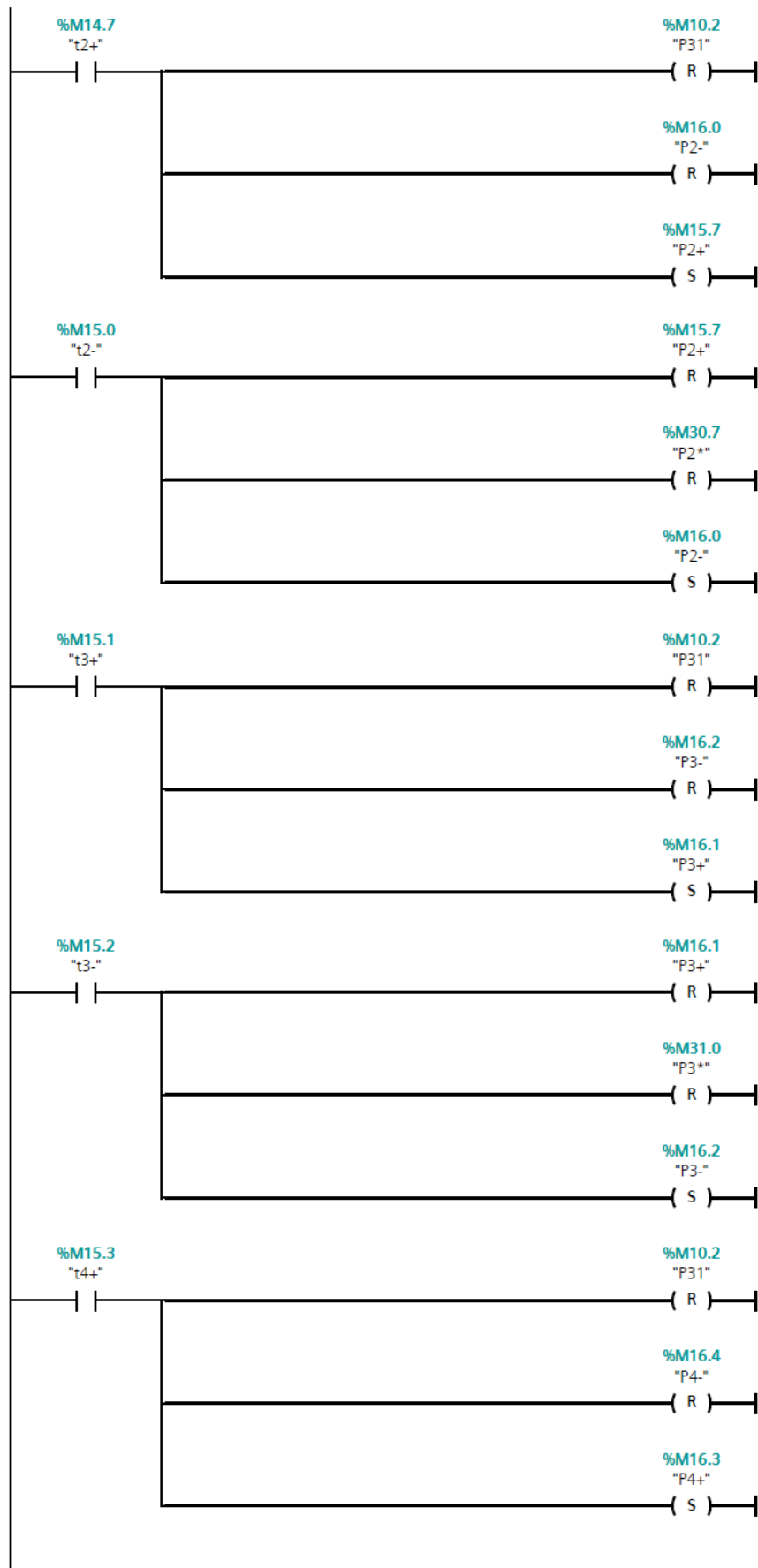


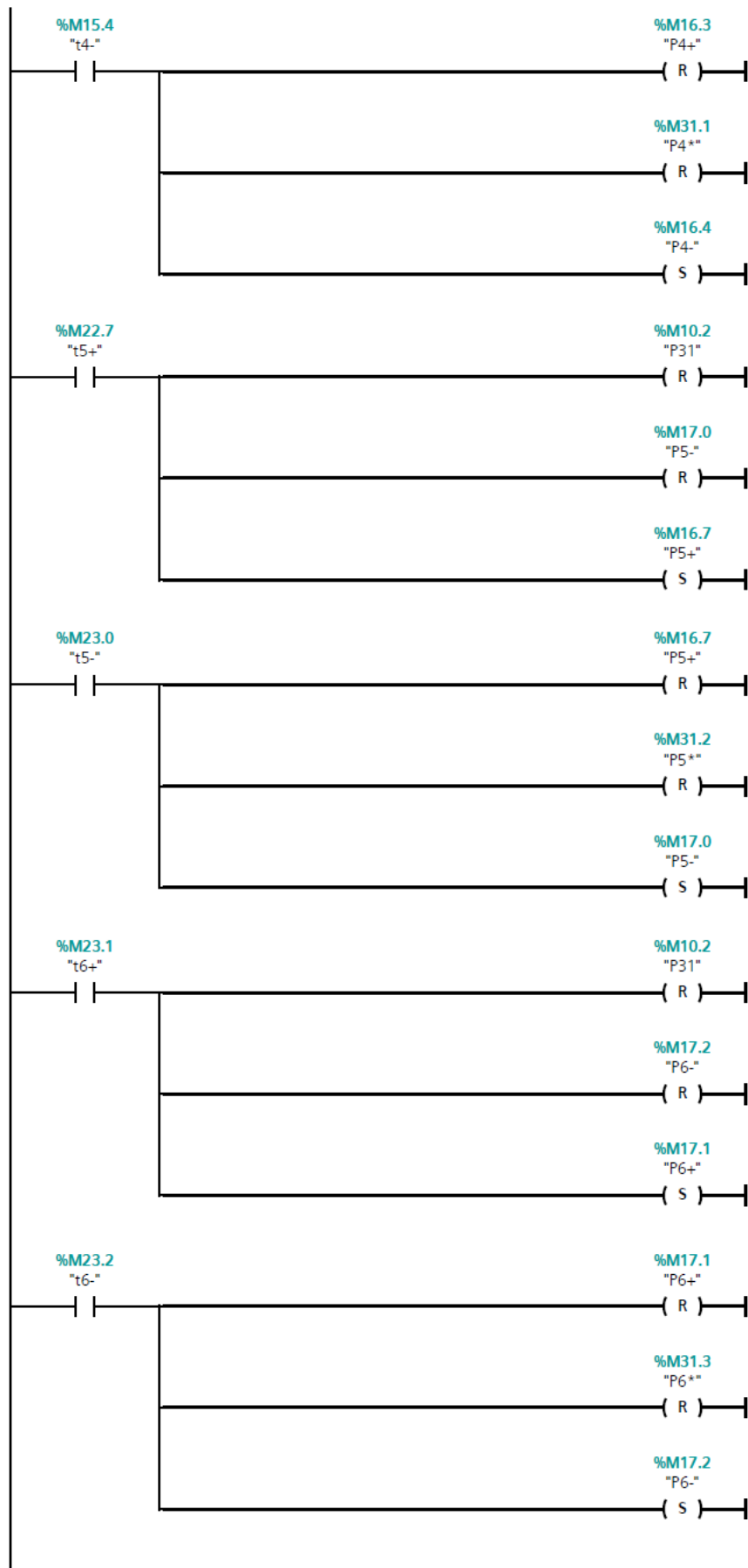


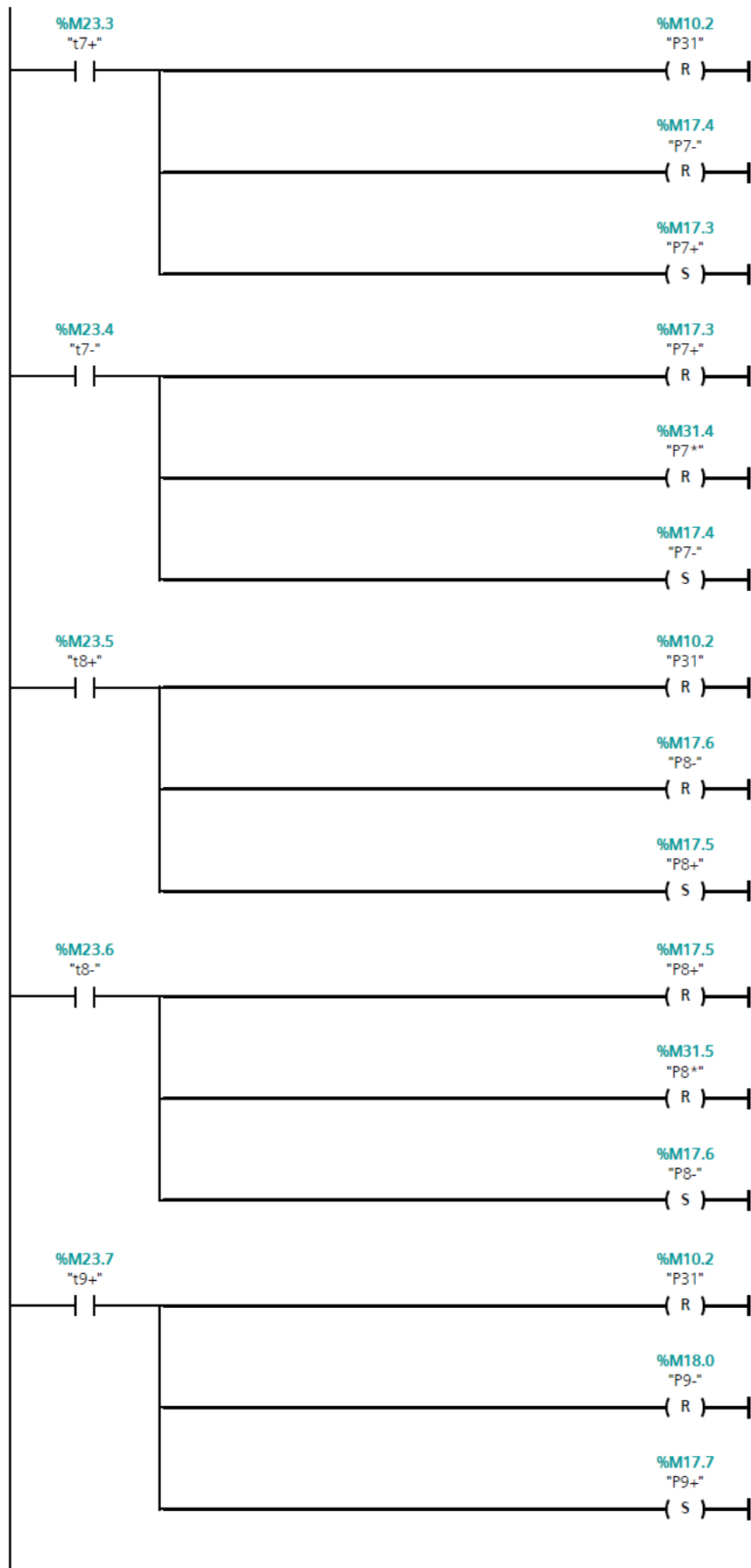


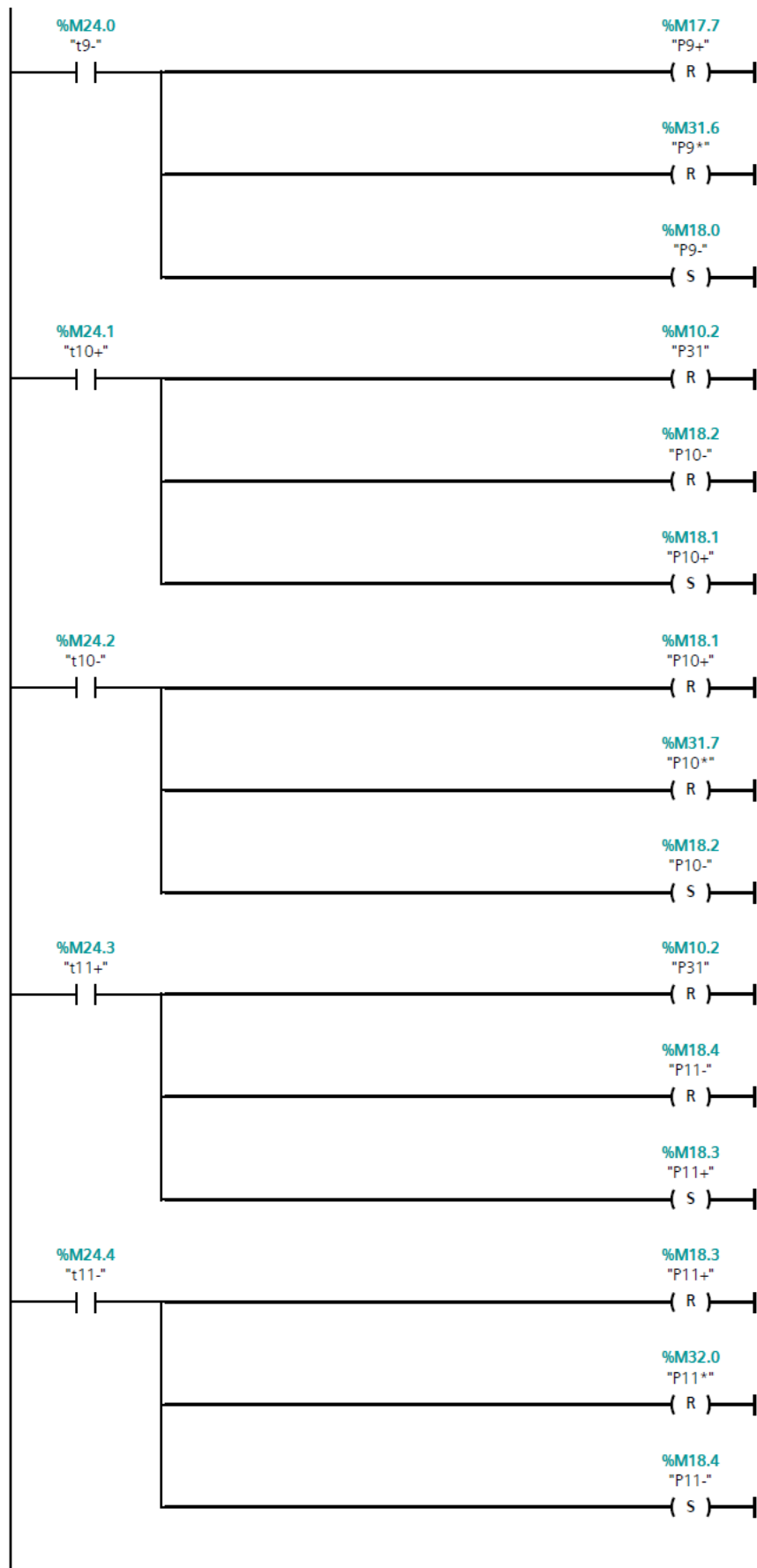


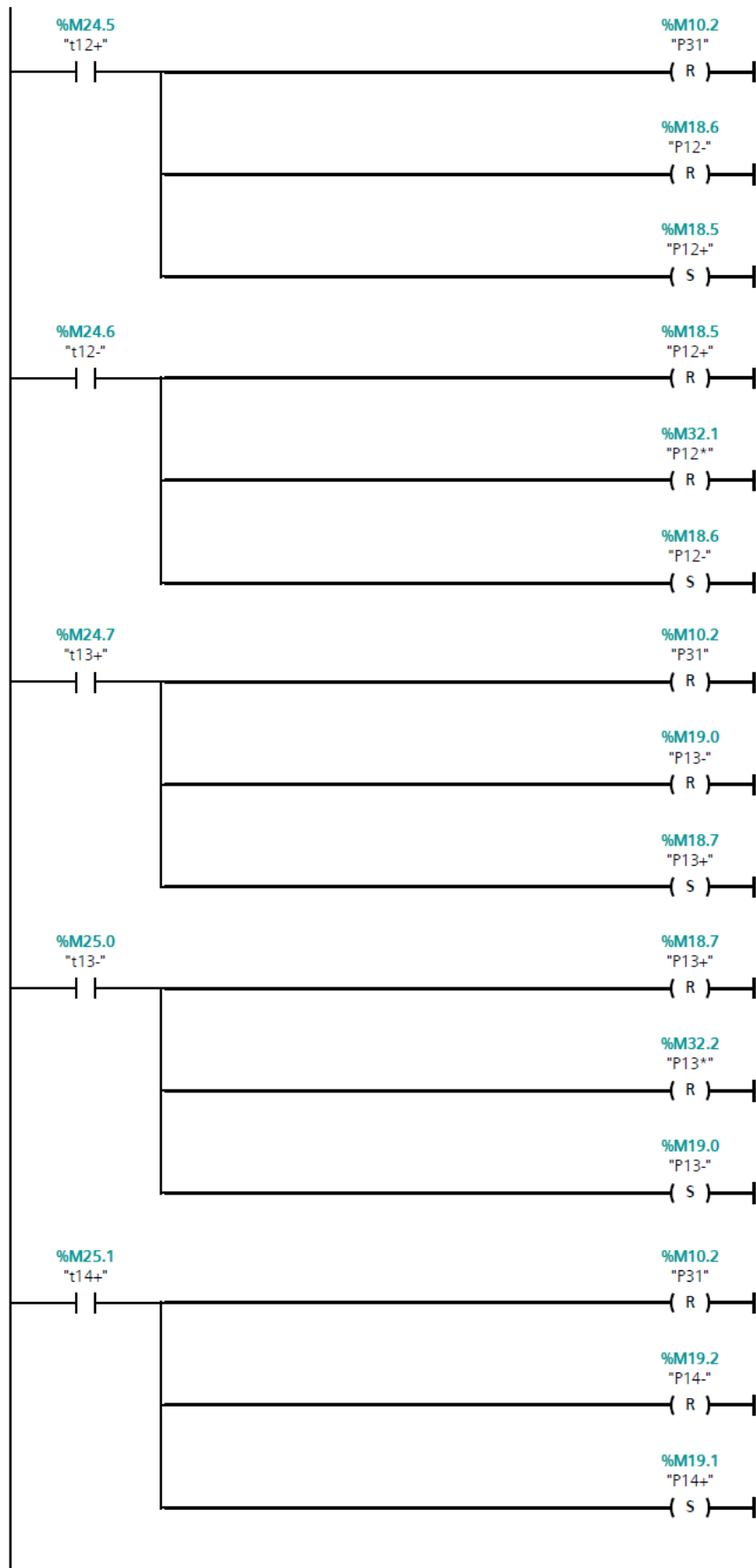


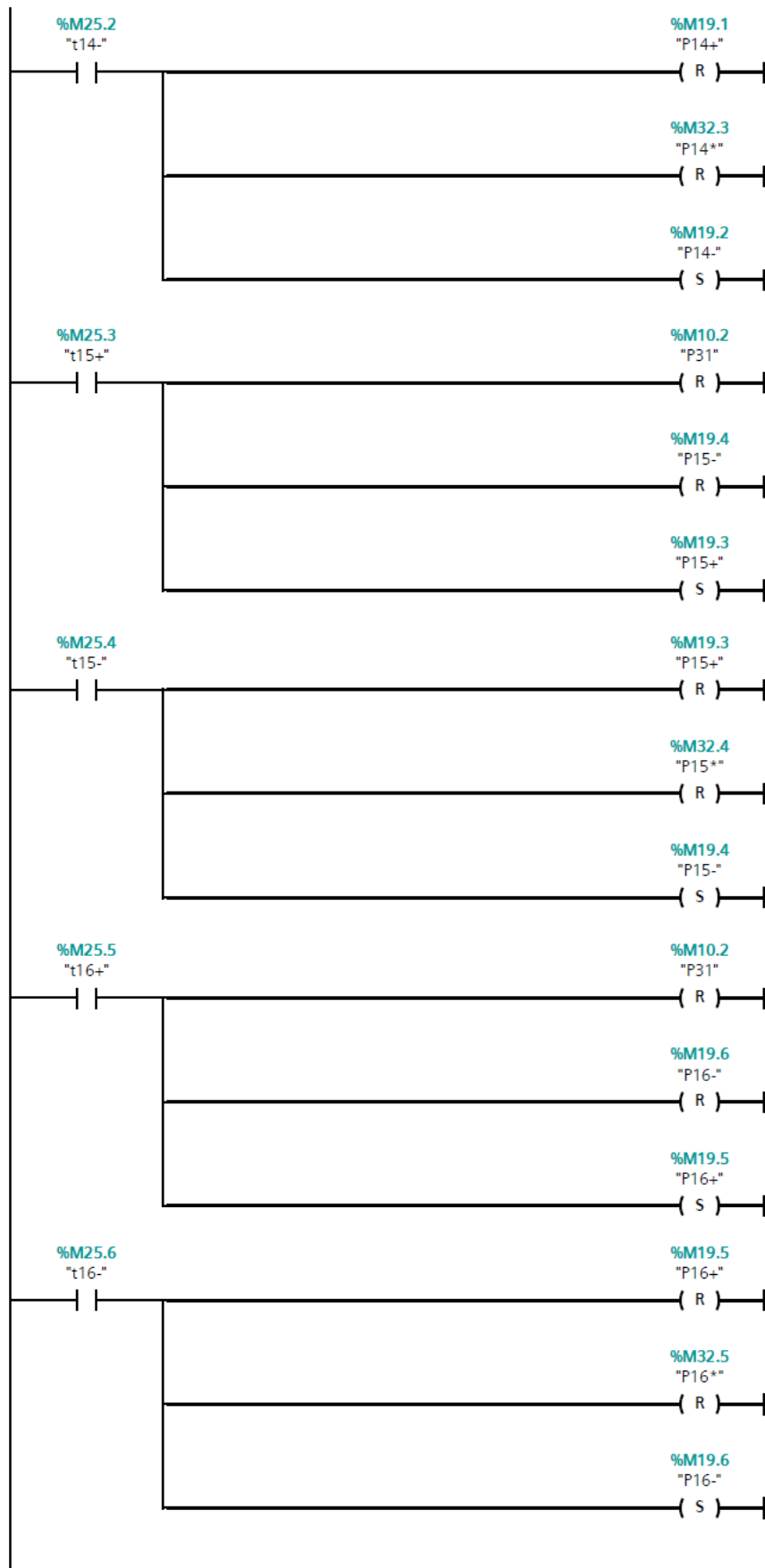


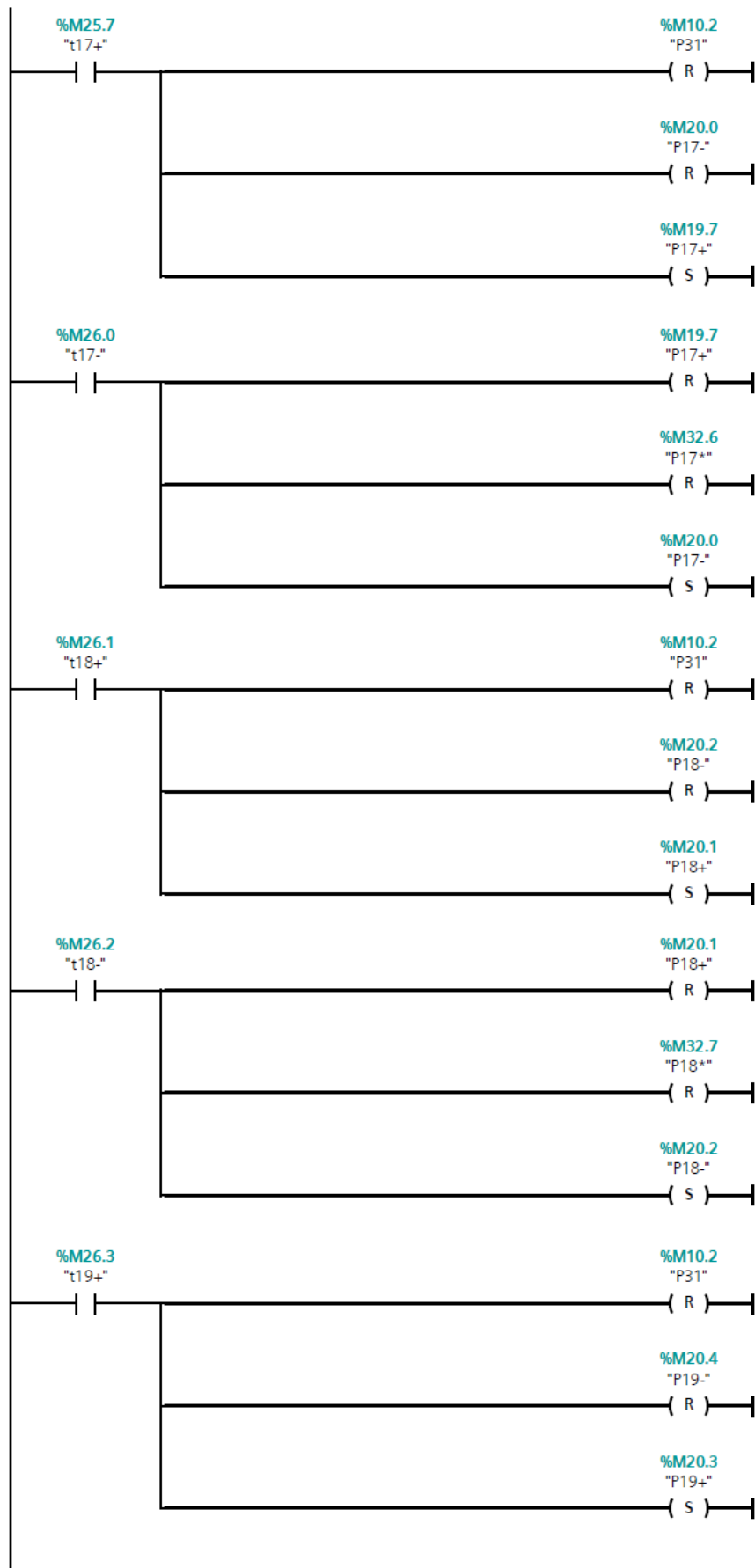


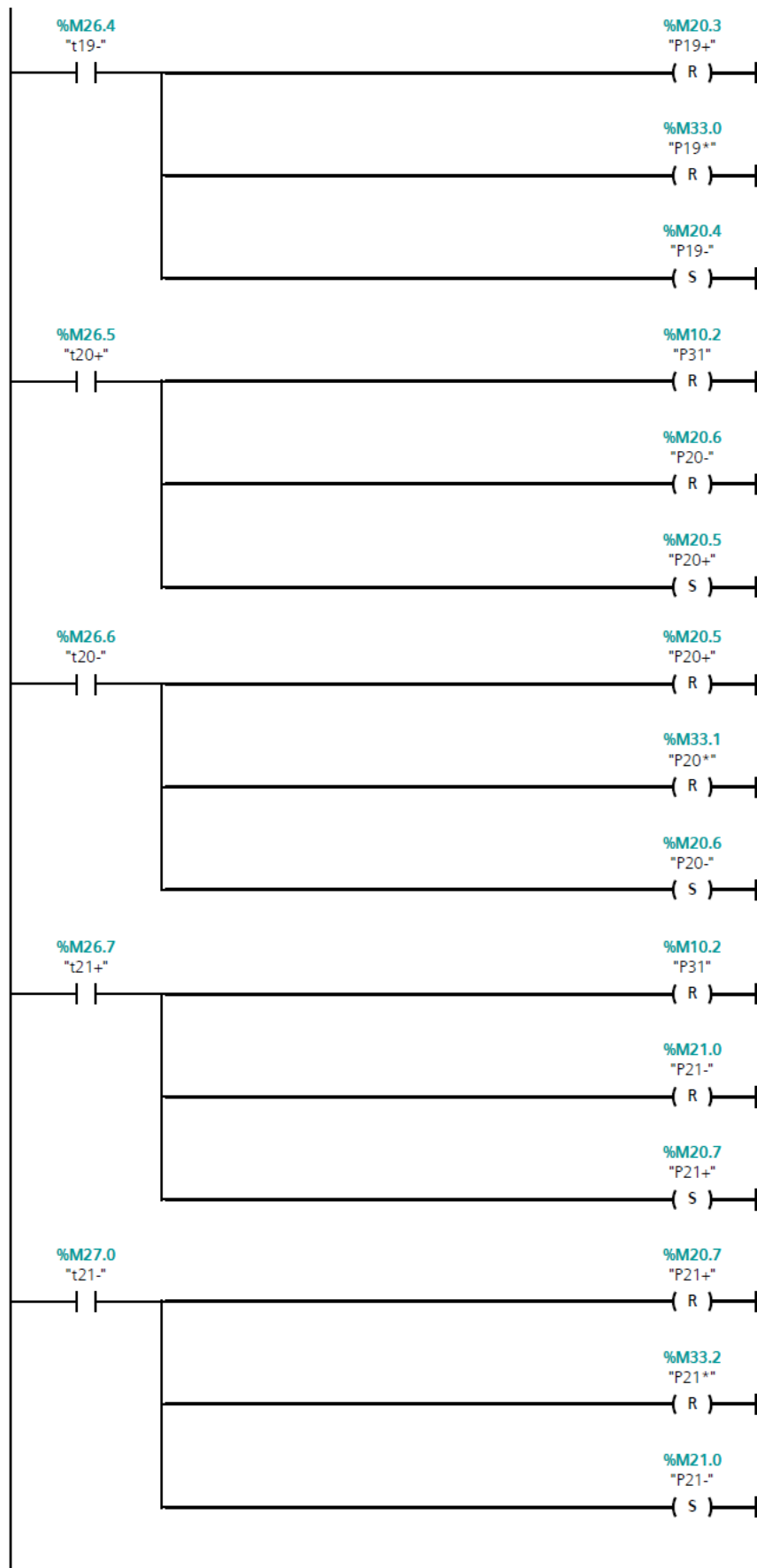


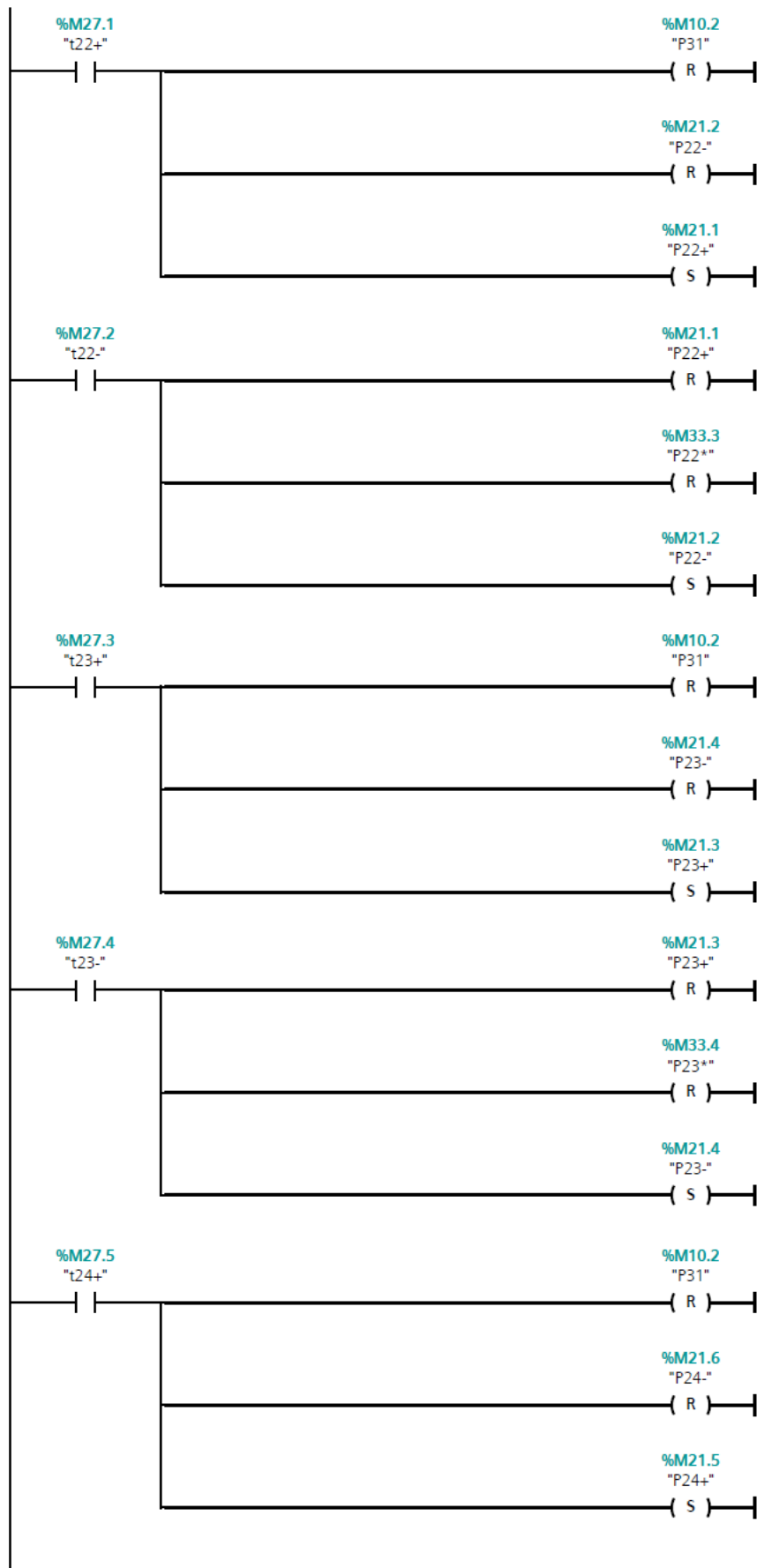


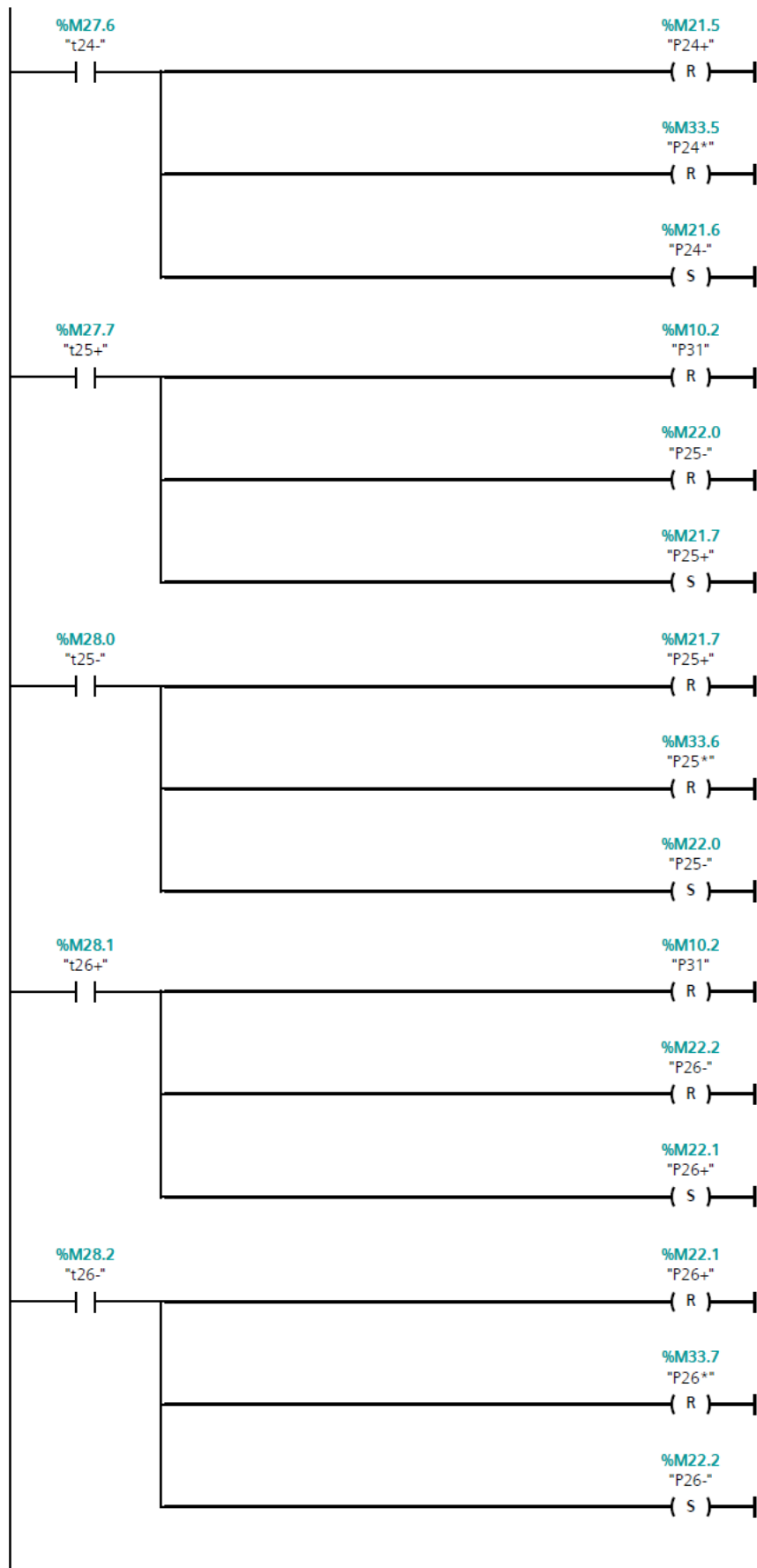


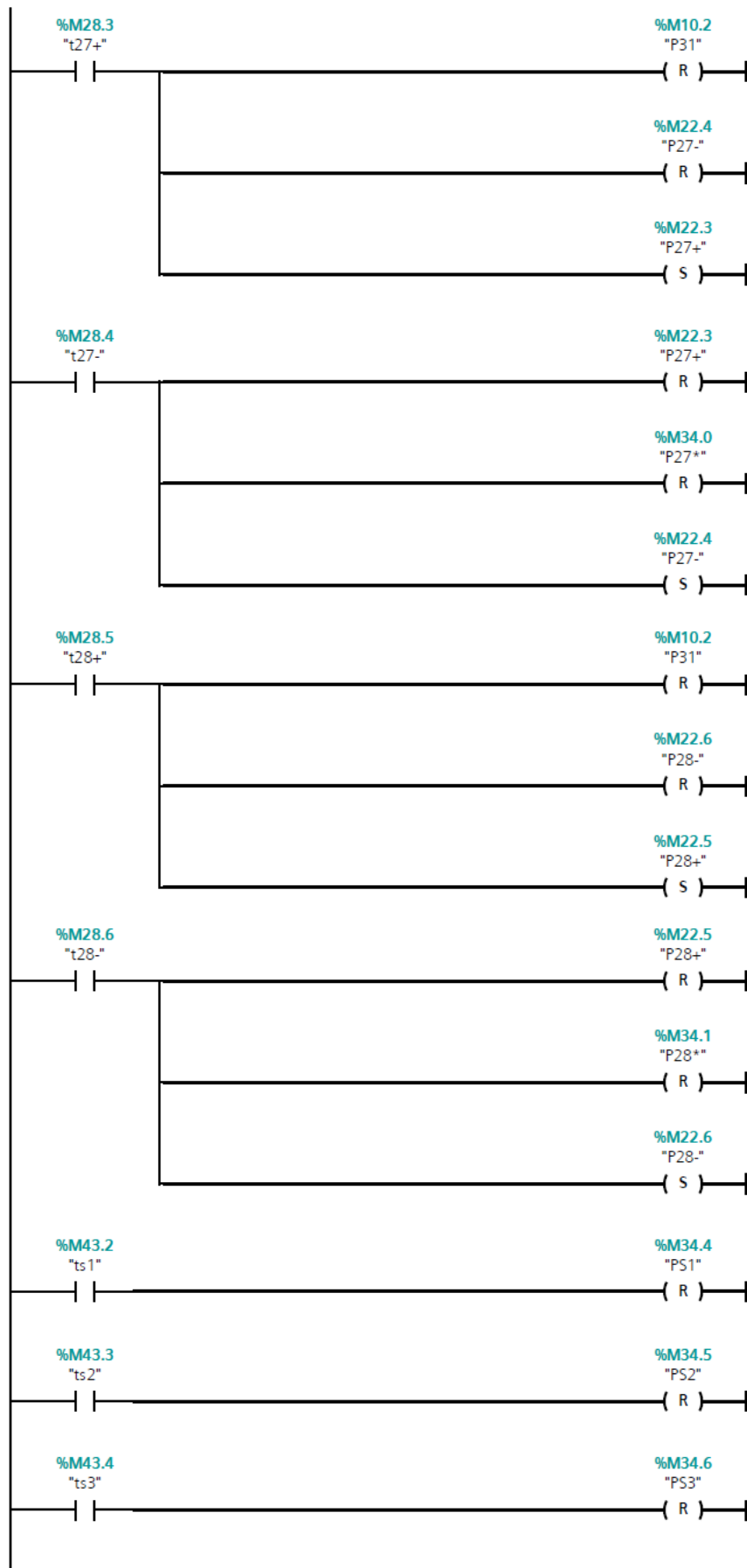


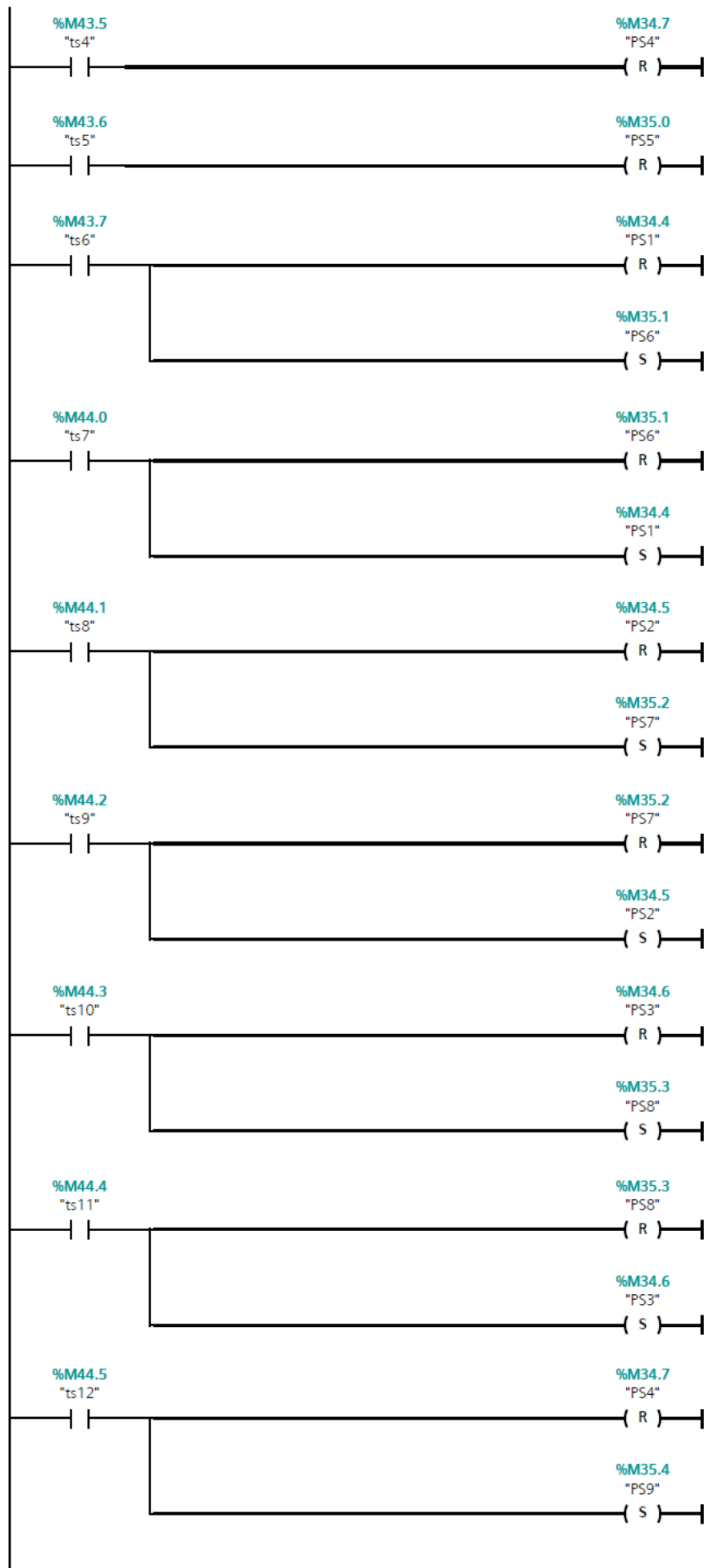


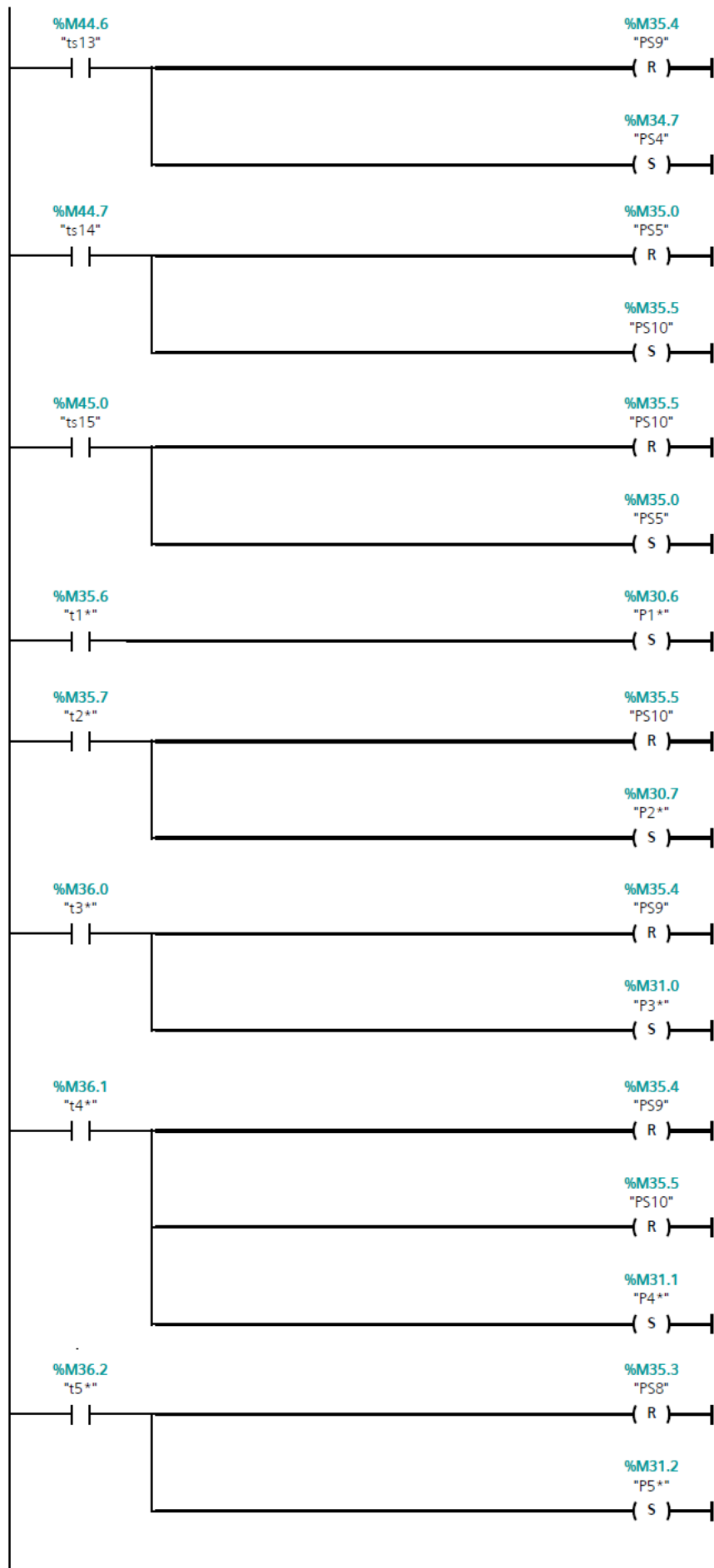


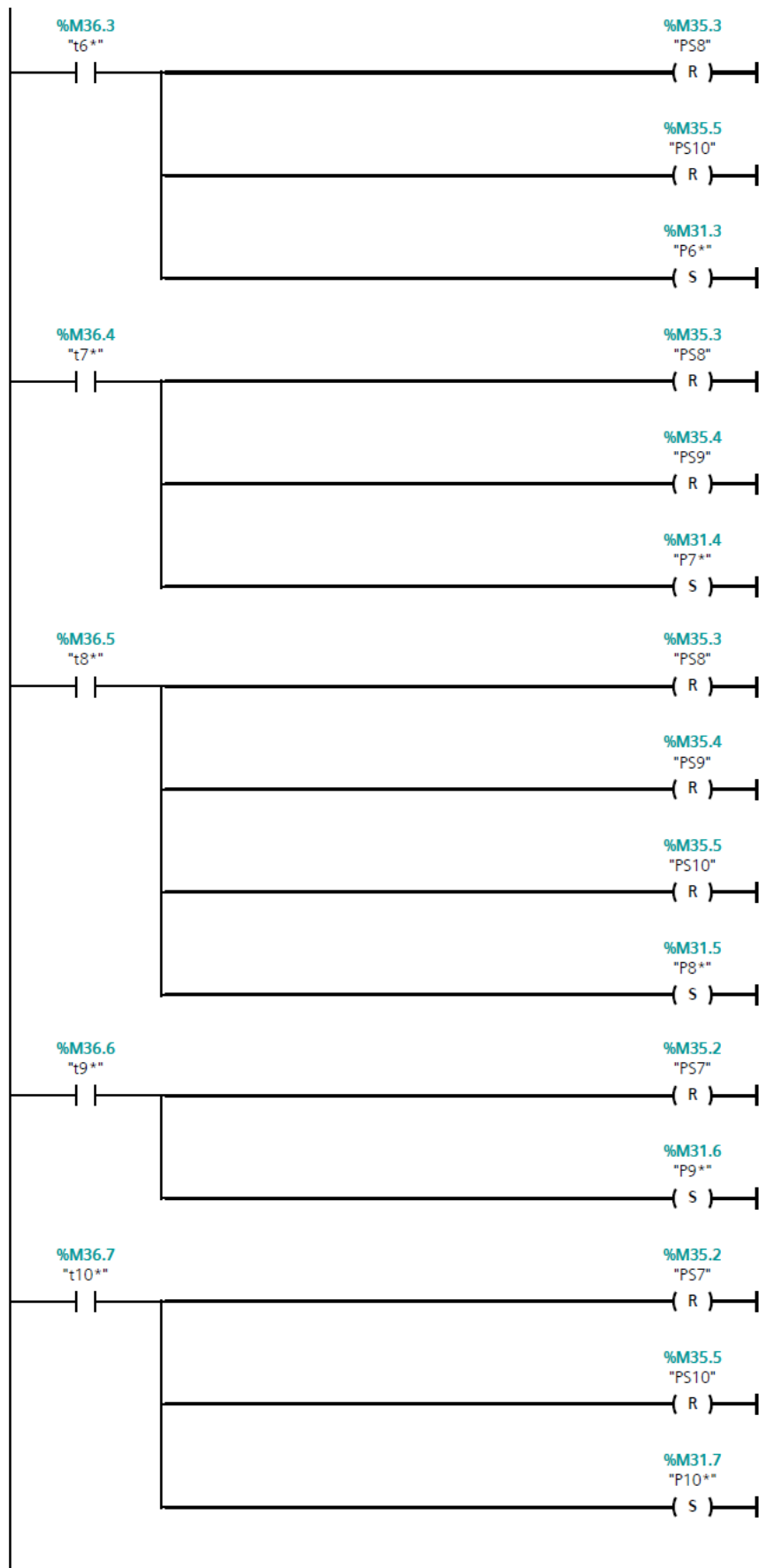


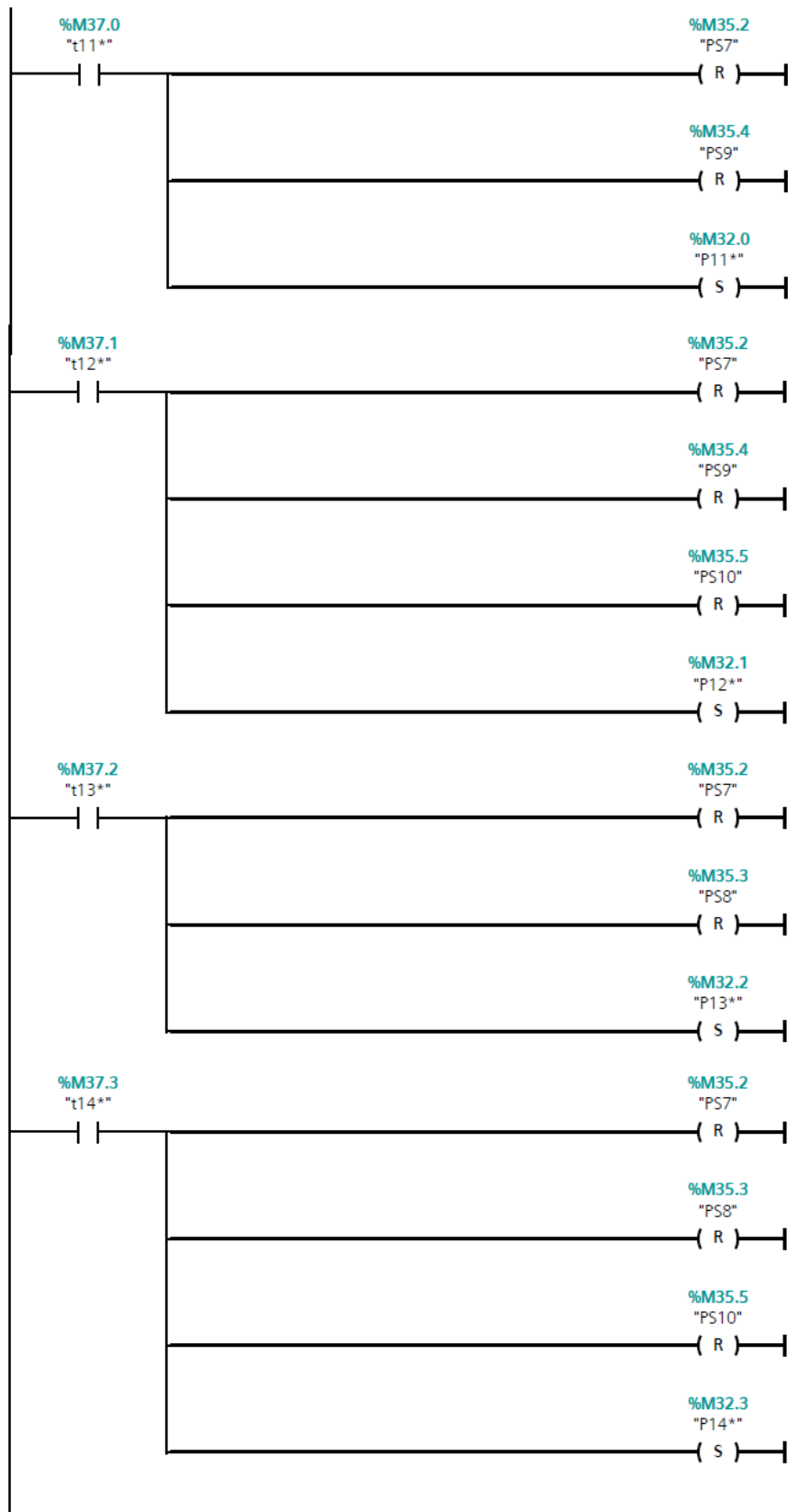


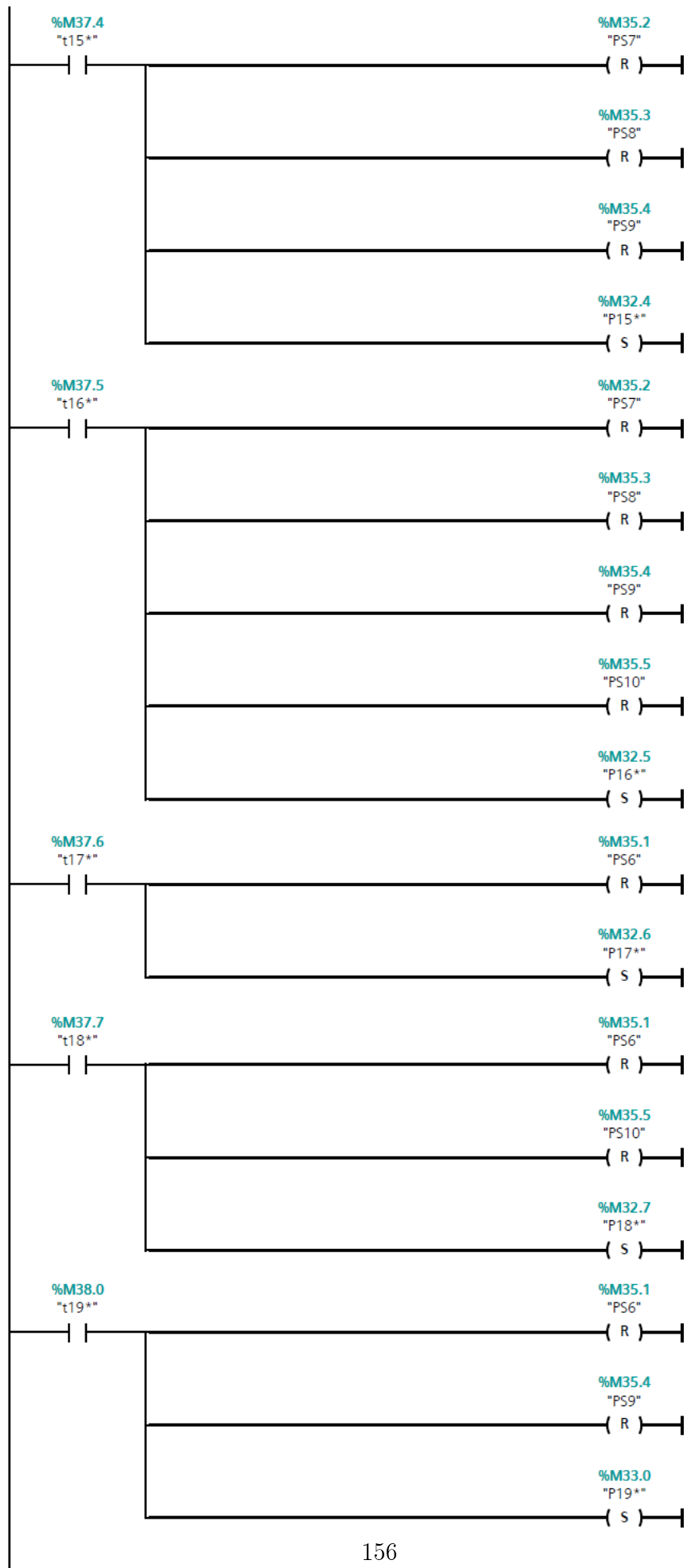


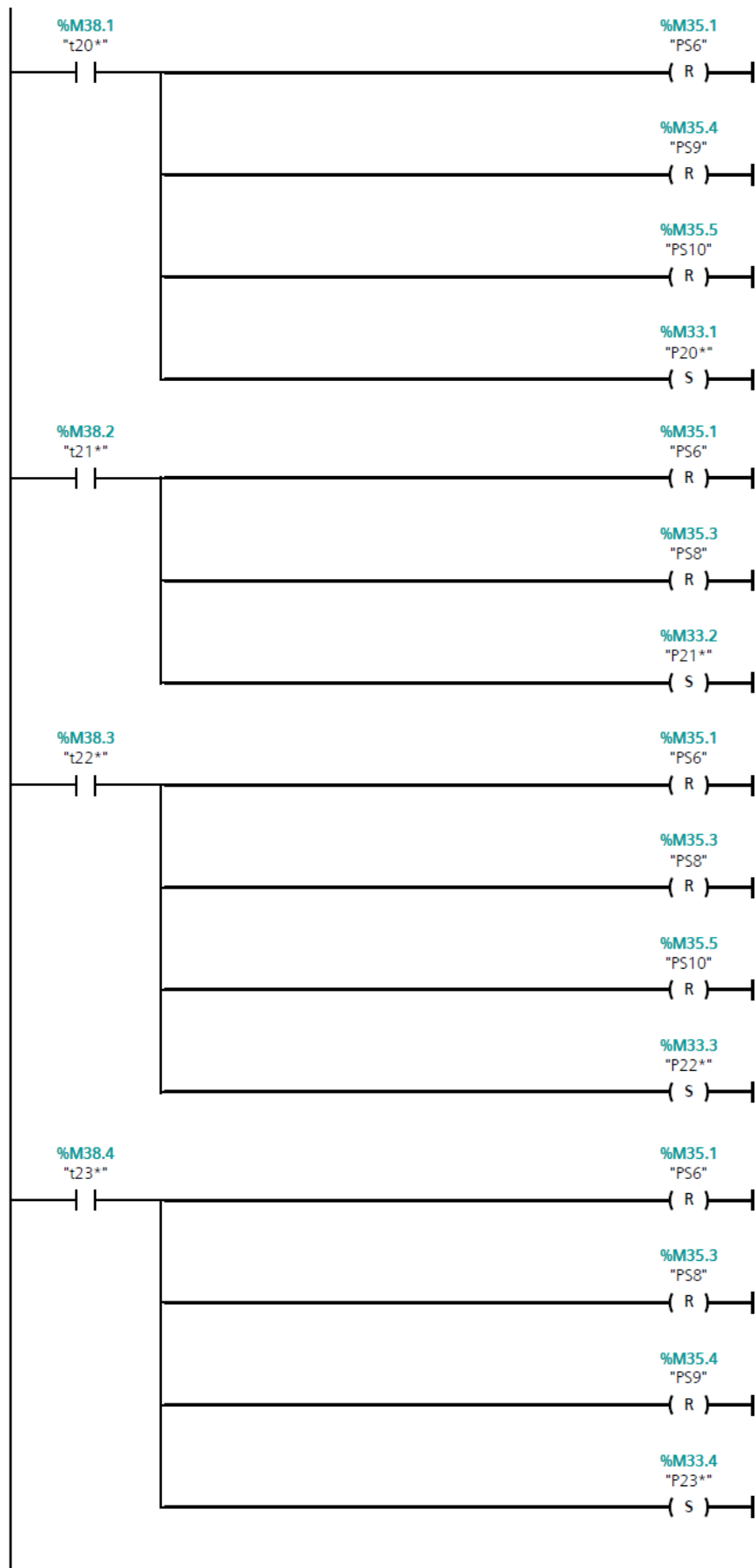


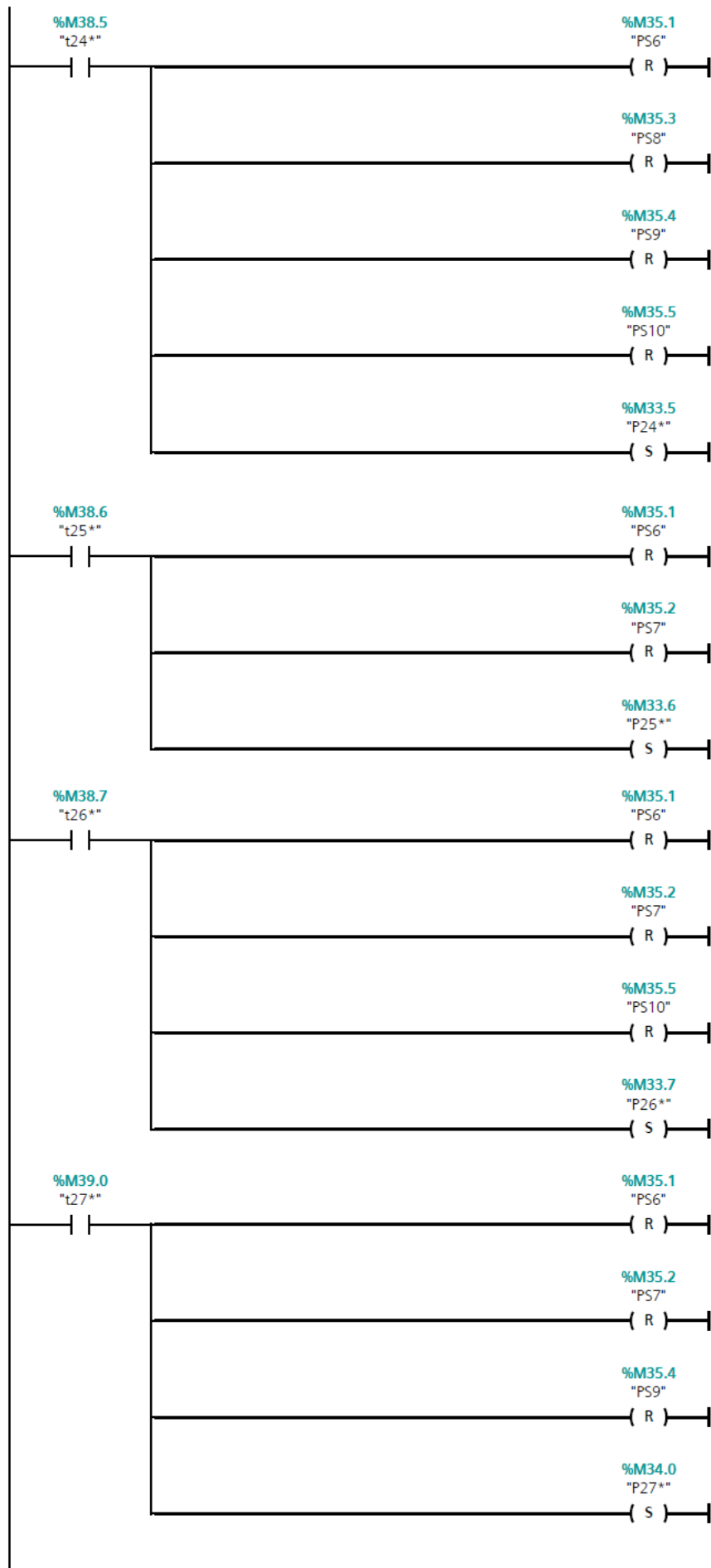


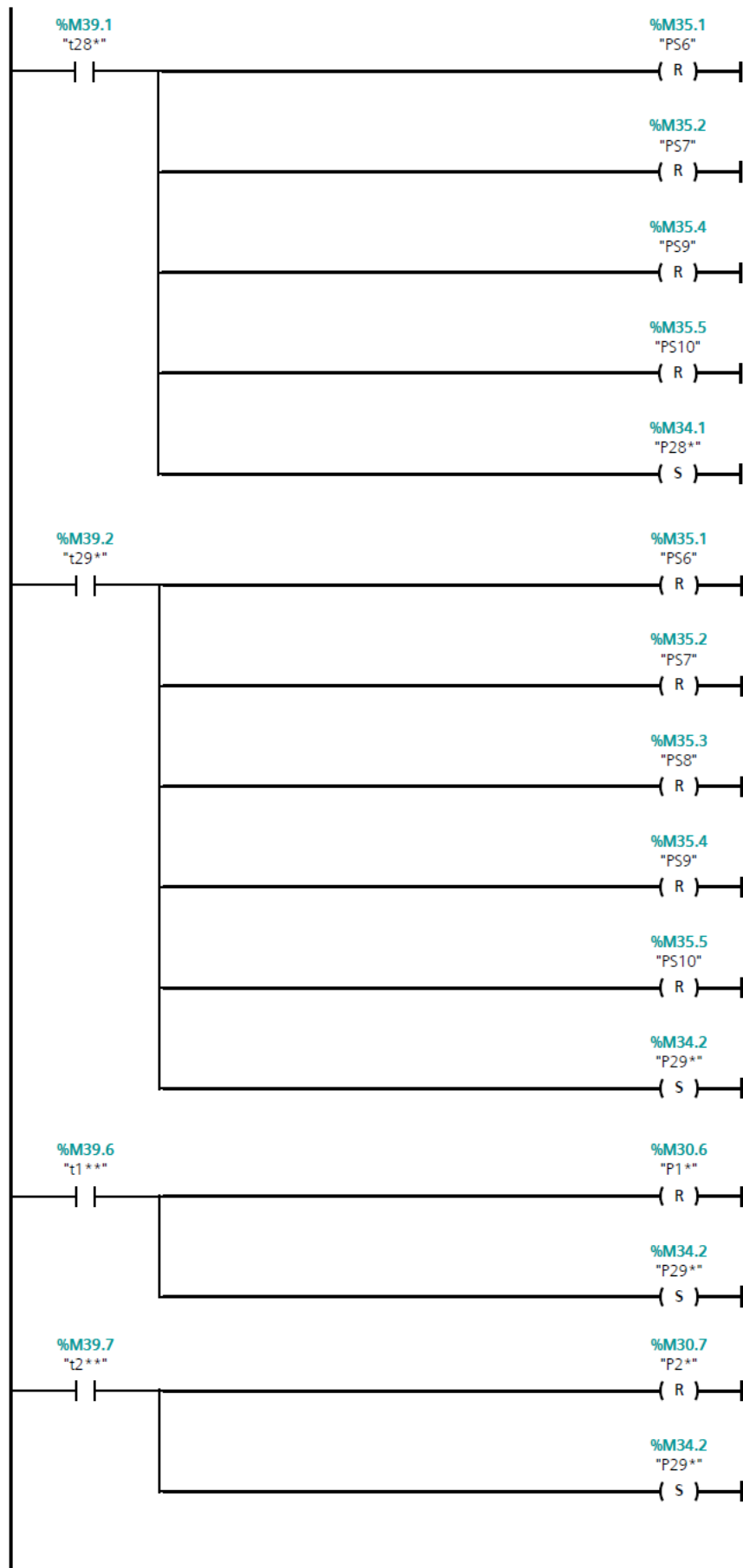


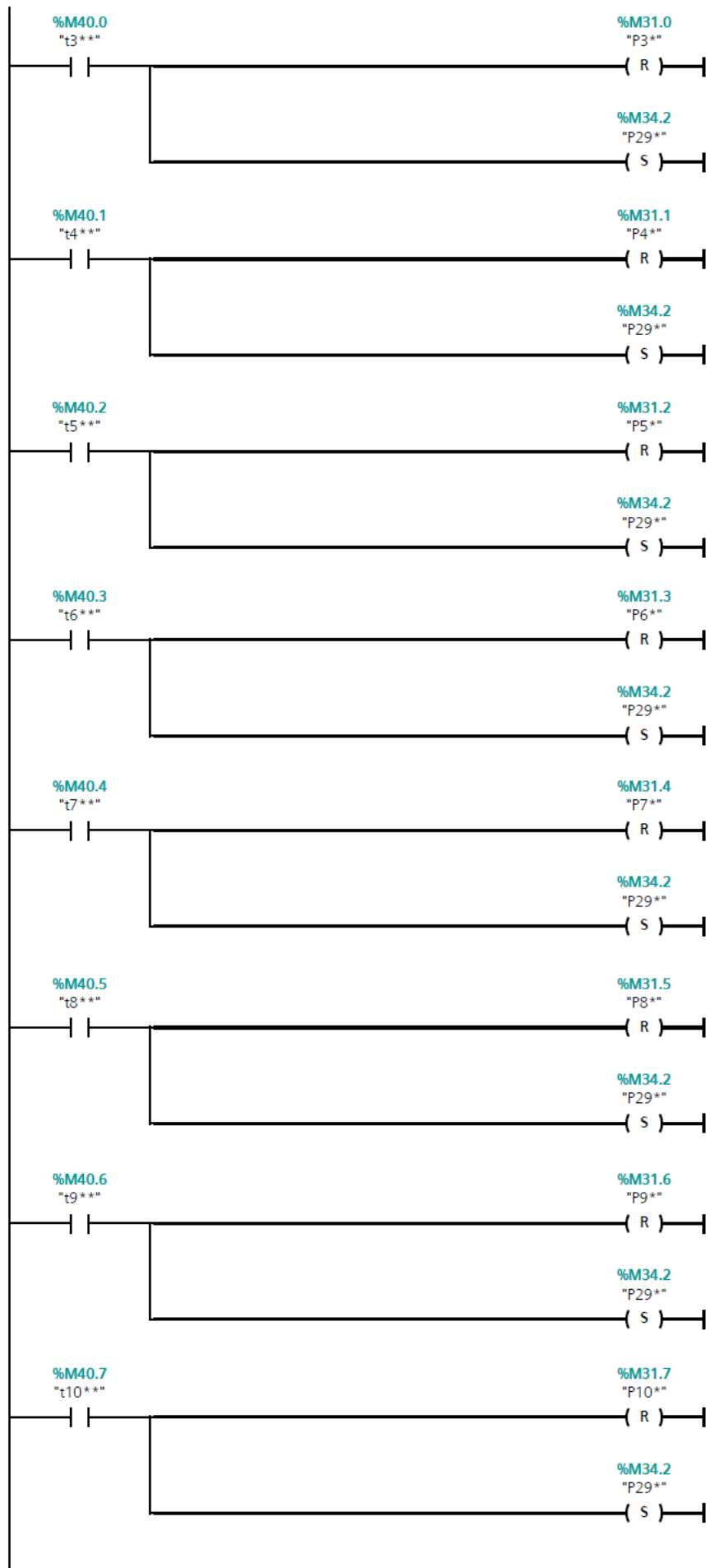


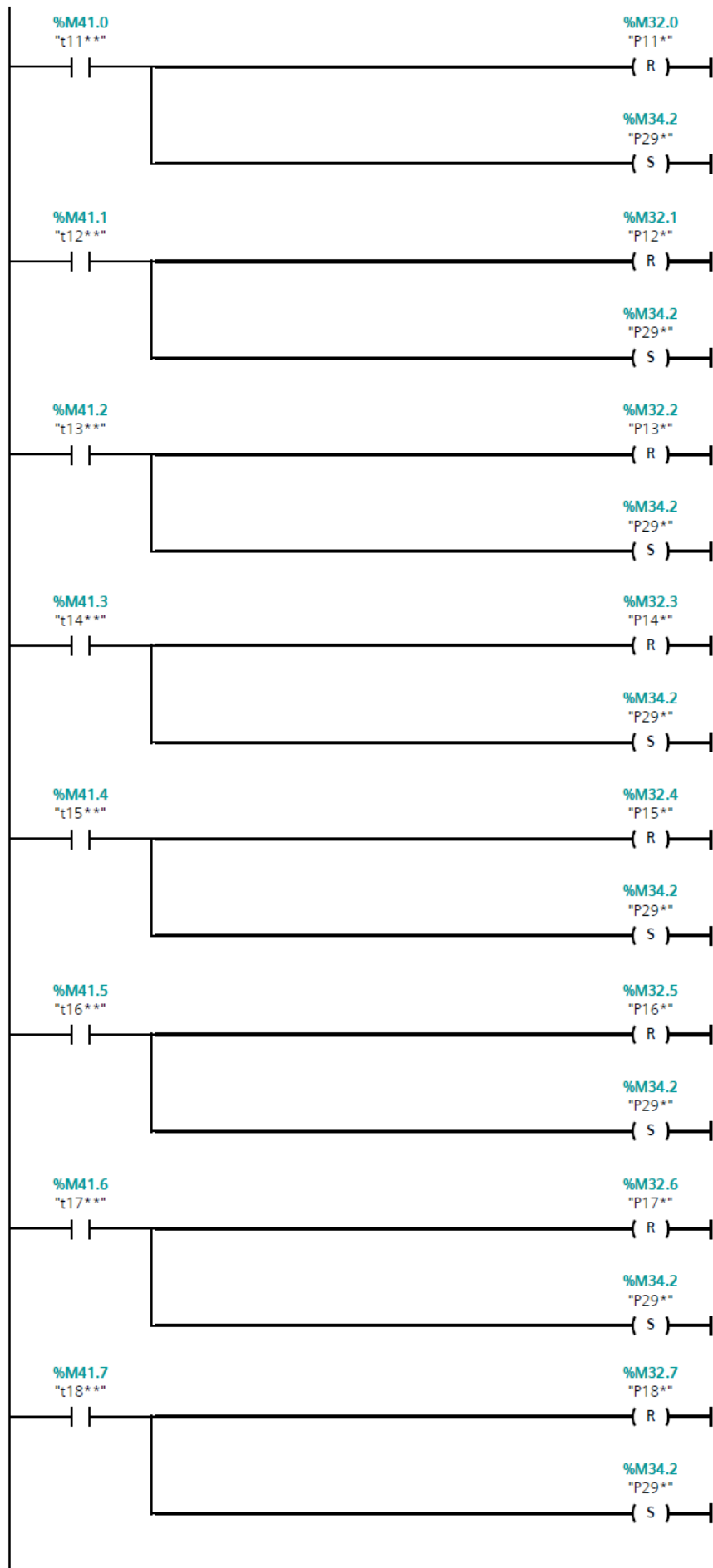


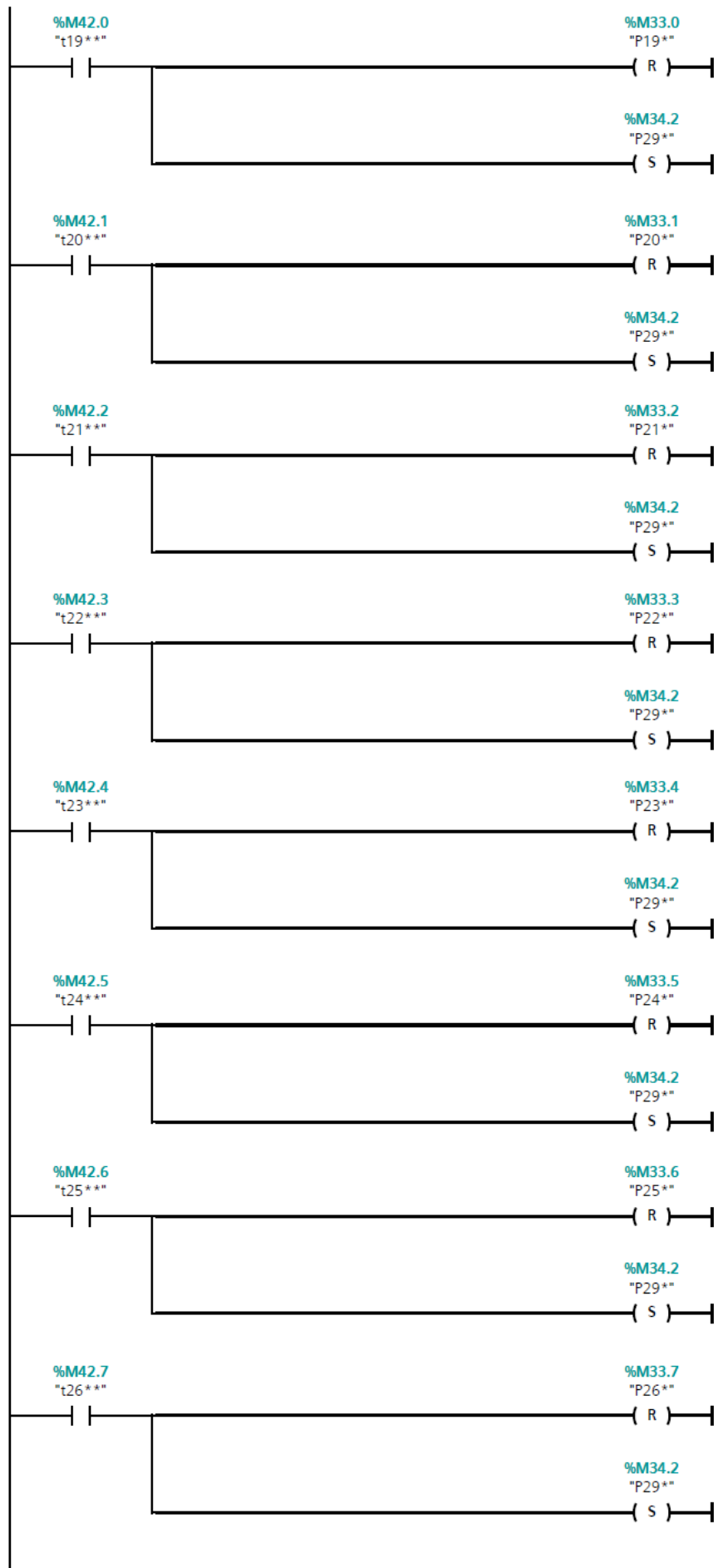


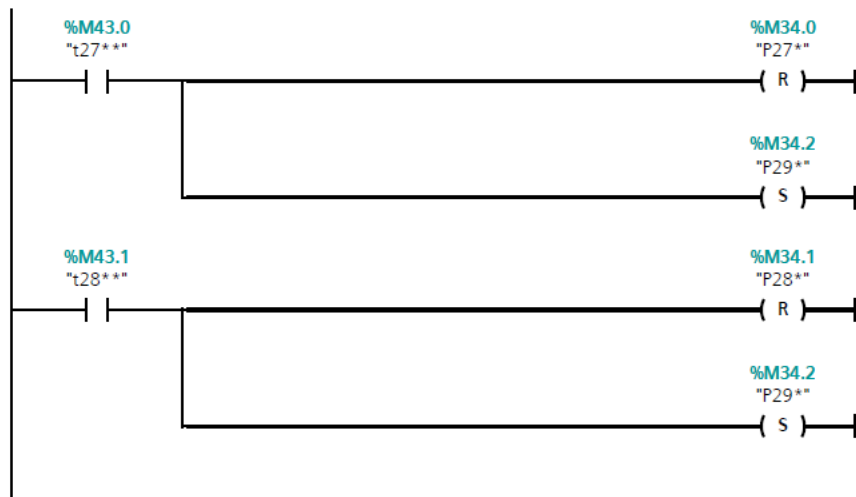












## A.5 Módulo das Ações

