



Universidade Federal de Campina Grande

Centro de Engenharia Elétrica e Informática

Curso de Graduação em Engenharia Elétrica

ANDRÉA DE ANDRADE MENDES CABRAL

PROJETO DE SUPERVISÓRIO PARA ESTEIRA TRANSPORTADORA

Campina Grande, Paraíba
Outubro de 2016

ANDRÉA DE ANDRADE MENDES CABRAL

PROJETO DE SUPERVISÓRIO PARA ESTEIRA TRANSPORTADORA

*Trabalho de Conclusão de Curso submetido à
Unidade Acadêmica de Engenharia Elétrica da
Universidade Federal de Campina Grande como
parte dos requisitos necessários para a obtenção do
grau de Bacharel em Ciências no Domínio da
Engenharia Elétrica.*

Área de Concentração: Controle e Automação

Orientador:

Professor George Acioli Júnior, D. Sc.

Campina Grande, Paraíba
Outubro de 2016

ANDRÉA DE ANDRADE MENDES CABRAL

PROJETO DE SUPERVISÓRIO PARA ESTEIRA TRANSPORTADORA

Trabalho de Conclusão de Curso submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Controle e Automação

Aprovado em ____ / ____ / _____

Universidade Federal de Campina Grande
Avaliador

Professor George Acioli Júnior, D. Sc.
Universidade Federal de Campina Grande
Orientador, UFCG

*Dedico este trabalho aos meus pais, à minha esposa,
aos meus irmãos e aos meus sobrinhos.*

AGRADECIMENTOS

Agradeço a toda minha família, em especial aos meus pais, Edivânia e José Ribeiro, e meus irmãos, Leandro e Júnior, pelo apoio incondicional. E também aos meus sobrinhos, Elias Henrique e João Lucas, por sempre alegrar o meu dia.

Aos professores do Departamento de Engenharia Elétrica, por todo conhecimento repassado, em especial aos professores George Acioli Júnior, por ter aceitado me orientar durante este trabalho e Péricles Rezedo Barros, pelo acompanhamento durante este projeto.

A todos os colegas de laboratório que me auxiliaram sempre que precisei.

À minha esposa e companheira de todas as horas, Carol, por seu amor e dedicação, e a todos meus amigos que me acompanharam e me deram força para enfrentar essa jornada.

A todos vocês, meu muito obrigado.

RESUMO

Este projeto de conclusão de curso consiste no desenvolvimento de uma IHM para um sistema de monitoramento de uma esteira transportadora de modelo reduzido para aplicações acadêmicas. O sistema em questão é composto por uma esteira conectada a um CLP (ZAP 91X da HI Tecnologia), no qual estão conectados alguns sensores e motor que geram ou recebem as variáveis de interesse. O CLP foi conectado à rede via cabo Ethernet e fez-se a comunicação com o servidor no computador localizado na mesma sala e conectado à mesma rede. A IHM foi desenvolvida para o sistema de supervisão, utilizando o software InTouch Wonderware, possibilitando o monitoramento de temperatura, execução de rotinas de ação e calibração do motor DC. A comunicação entre o sistema supervisório e o CLP do sistema foi realizada por meio do protocolo de comunicação OPC, padrão para aplicações industriais. Por meio dele construiu-se um servidor OPC, HT1 Power Tool, e um cliente OPC, InTouch, para leitura e escrita de variáveis, denominadas tagnames. Além disso, foi necessário utilizar o OPCLink para criar um canal de comunicação entre o servidor e o cliente OPC.

Palavras-chave: CLP, Ladder, HT1 Power Tool, SPDSW, OPCLink, Protocolo OPC, IHM, InTouch.

ABSTRACT

This project involves the development of a graphic interface, useful for a specific automation system, a size reduced belt conveyor for academic learning. First of all, was loaded in Programmable Logic controller a program written in Ladder using the SPDSW software. Secondly, was developed for the monitoring system (SCADA), Wonderware InTouch using software. The interface enables the monitoring of the variables of interest and calibration of the DC motor. The communication between the supervision system and PLC system was performed by the OPC communication protocol standard for industrial applications. Through it built up an OPC server, HT1 Power Tool, and an OPC client, InTouch, for reading and writing of variables, called tagnames. Furthermore, it was necessary to use OPCLink to create a communication channel between the server and the OPC client.

Keywords: PLC, Ladder, HT1 Power Tool, SPDSW, OPCLink, OPC Protocol, HMI, InTouch.

LISTA DE FIGURAS

Figura 1 - Esteira Transportadora.	16
Figura 2 - Motor DC. Fonte [1]	17
Figura 3 - Encoder. Fonte: [2]	17
Figura 4 - Sensor óptico de barreira.	18
Figura 5 - Módulo de aquecimento.	19
Figura 6 - Módulo de Resfriamento.	19
Figura 7: Modos de operação do InTouch.....	20
Figura 8: Objetos de tela básicos. Fonte[3]	21
Figura 9: Objetos de Tela Especiais – Wizard.	22
Figura 10: Tipos de tagnames. Fonte [3].....	22
Figura 11: Animação de objetos. Fonte [3]	23
Figura 12: Campos de alarmes. Fonte [3].....	25
Figura 13: Modelo de tarefa para aplicação Calibração do Motor.	28
Figura 14: Modelo de tarefa para aplicação Monitoramento e Execução de Rotinas.	29
Figura 15: Diagrama de caso de uso – Calibração de Motor.....	30
Figura 16: Diagrama de Caso de Uso da aplicação Monitoramento e execução de rotinas.....	31
Figura 17: Calibração de motor – Diagrama de Sequência.	33
Figura 18: Execução de Rotina de Ação 1 com calibração de motor – Diagrama de Sequência.	34
Figura 19: Execução de Rotina de Ação 2 – Diagrama de Sequência.	35
Figura 20: Janela inicial da aplicação Calibração do Motor.	37
Figura 21: Janela Sobre Calibração.	37
Figura 22: Tela Calibração Sentido Horário.....	38
Figura 23: Configuração dos parâmetros do gráfico de tendência velocidade – Sentido Horário.....	39
Figura 24: Configuração dos parâmetros do gráfico de tendência S0Motor – Sentido Horário.	39
Figura 25: Window Script da janela Calibração Sentido Horário – On Show.....	40
Figura 26: Window Script da janela Calibração Sentido Horário – While Showing.....	41
Figura 27: Action Script do botão Iniciar - Calibração Sentido Horário.....	42

Figura 28: Tela Calibração Sentido Anti-horário.....	43
Figura 29: Configuração dos parâmetros do gráfico de tendência velocidade – Sentido Anti-horário.....	43
Figura 30: Configuração dos parâmetros do gráfico de tendência SOMotor – Sentido Anti-horário.....	44
Figura 31: Window Script da janela Calibração Sentido Anti-horário – On Show.	44
Figura 32: Window Script da janela Calibração Sentido Horário – While Showing.....	45
Figura 33: Action Script do botão Iniciar - Calibração Sentido Horário.....	46
Figura 34: Tela Login – Inativo.....	47
Figura 35: Tela Login – Digitando usuário e senha.	48
Figura 36: Tela Login – Ativo.....	48
Figura 37: Janela Inicial.	49
Figura 38: Janela Valores Padrões de Velocidade.....	50
Figura 39: Janela Limite de Variáveis.....	50
Figura 40: Câmara de Aquecimento.....	51
Figura 41: Configuração dos parâmetros do gráfico de Temperatura (°C).....	52
Figura 42: Configuração dos parâmetros do gráfico de Duty Cycle (%).	52
Figura 43: Window Script Câmara de Aquecimento – While Showing.....	53
Figura 44: Câmara de Resfriamento.....	54
Figura 45: Configuração dos parâmetros do gráfico de temperatura de resfriamento.	54
Figura 46: Diagrama de estados da rotina de ação 1.	55
Figura 47: Rotina de Ação 1 – Window Maker.....	56
Figura 48: Deslocamento do bloco 1 relacionado à tagname deslocamento.....	57
Figura 49: Visibilidade do bloco 1 relacionada à tagname state.	57
Figura 50: Rotina de Ação 1 - Estado inicial (state=1, SBO1=1).	59
Figura 51: Rotina de ação 1 – Estado 2 (state=0, state1=1, SBO2=1).	59
Figura 52: Rotina de ação 1 – Estado 3 (cont1=0, state1=1).....	60
Figura 53: Rotina de Ação 1 – Estado 4 (state1=0, state2=1).	61
Figura 54: Rotina de Ação 1 – Estado 5 (state3=1, state2=0).	61
Figura 55: Rotina de Ação 1 – Estado 6 (state3=1).....	62
Figura 56: Rotina de Ação 1 – Estado 7 (state3=1).....	62
Figura 57: Diagrama de estados da rotina de ação 1.	63
Figura 58: Rotina de Ação 2 – Window Maker.....	64
Figura 59: Deslocamento do bloco 1 relacionada à tagname deslocamento.	64

Figura 60: Rotina de Ação 2 - Estado inicial (state=1, SBO4=1).	65
Figura 61: Rotina de Ação 2 - Estado 1 (state=1, SBO4=1).	66
Figura 62: Rotina de Ação 2 - Estado 2 (state=0, state1=1, SBO3=1).	66
Figura 63: Rotina de ação 2 – Estado 4 (state1=0, state2=1, SBO2=1).	67
Figura 64: Rotina de ação 2 – Estado 5 (state2=0, state3=1, SBO1=1).	68
Figura 65: Rotina de ação 2 – Estado 6 (state3=1).	68
Figura 66: Rotina de ação 2 – Estado 7 (state3=1, SBO4=1).	69
Figura 67- Janela inicial do SPDSW.	73
Figura 68- Janela inicial do SPDSW.	74
Figura 69- Janela do SPDSW para localizar controladores em rede.	75
Figura 70- Janela do SPDSW para identificar equipamentos em rede.	75
Figura 71- Janela que apresenta equipamentos em rede.	76
Figura 72- Aba Ethernet da janela Configuração de comunicação de equipamento.	76
Figura 73- Janela para Configuração do Driver Comunicação - Drivers.	77
Figura 74- Janela para Configuração do Driver Comunicação – Ethernet.	77
Figura 75 – Janela inicial do Programa HT1 Power Tool.	78
Figura 76 – Janela do HT1 para identificar os equipamentos na rede.	79
Figura 77 – Janela do programa HT1 para configuração do canal.	79
Figura 78 - Janela do programa HT1 para configuração dos parâmetros equipamento.	80
Figura 79 - Janela do programa HT1 para configuração dos parâmetros do bloco de dados. ...	81
Figura 80 – Janela <i>Statistics</i> do programa HT1 – Visualização de qualidade de dados.	82
Figura 81 – Janela inicial do OPCLink.	82
Figura 82 – Janela do OPCLink para criação de tópicos.	83
Figura 83 - Janela do OPCLink para configuração de tópicos.	83
Figura 84 - Janela do OPCLink para definição de tópicos – novo tópico.	84
Figura 85 – Janela inicial do InTouch – Lista de aplicações.	84
Figura 86 – Janela inicial do InTouch Maker – criação de nova tela.	85
Figura 87 - Janela do INTOUCH MAKER para ajuste de parâmetros de nova tela.	85
Figura 88: Janela para configuração do Access.	86
Figura 88- Janela <i>Tagname Dictionary</i>	87

LISTA DE TABELAS

Tabela 1: Tipos de script.	24
Tabela 2: Parâmetros do deslocamento e visibilidade da rotina de ação1.....	58
Tabela 3: Parâmetros do deslocamento e visibilidade da rotina de ação 2.....	65
Tabela 4: Associação de Tags do CLP para aplicação do Intouch – Calibração de Motor....	69
Tabela 5: Associação das tags do CLP com as tagnames do InTouch.	69
Tabela 6: Tagnames de memória da aplicação para calibração do motor.	70
Tabela 7: Tagnames de memória da aplicação para calibração do motor.	71

SUMÁRIO

1	INTRODUÇÃO.....	14
2	ESTEIRA TRANSPORTADORA.....	14
	2.1 MotorDC	16
	2.2 Encoder.....	17
	2.3 Sensores Ópticos de Barreira	17
	2.4 Módulo de Aquecimento	18
	2.5 Módulo de Resfriamento	19
3	TUTORIAL INTOUCH.....	19
4	DESENVOLVIMENTO DA IHM.....	25
	4.1 Levantamento de Requisitos e Especificações da IHM	26
	4.1.1 Objetivos de Usabilidade	27
	4.1.2 Levantamento do Perfil do Usuário	27
	4.1.3 Análise da Tarefa.....	28
	4.2 Diagramas UML.....	30
	4.2.1 Diagramas de Casos de Uso.....	30
	4.2.2 Diagramas de Sequência	32
	4.3 Desenvolvimento de telas	36
	4.3.1 Calibração da Velocidade do Motor	36
	4.3.2 Aplicação para Monitoramento e Execução de Rotinas	46
	4.4 Dicionário de Tags	69
	4.4.1 Tags I/O	69
	4.4.2 Memory Tags	70
5	COMUNICAÇÃO OPC.....	71
	5.1 Software SPDSW	73
	5.2 Servidor OPC	78
	5.3 OPCLink.....	82
	5.4 Cliente OPC.....	83
6	CONCLUSÃO.....	86

LISTA DE ABREVIACOES

CLP – Controlador Lgico Programvel.

IHM – Interface Homem-Mquina.

MCIE – Mtodo de Construo de Interfaces Ergonmicas

SCADA – Supervisory Control And Data Acquisition.

UML – Unified Modeling Language.

1 INTRODUÇÃO

Este projeto de conclusão de curso consiste no desenvolvimento da IHM para um sistema de uma esteira transportadora de tamanho reduzido. A instrumentação feita no objeto de estudo permite acesso às variáveis de entrada e saída tais como: temperatura, velocidade de rotação e detector de presença.

A IHM foi desenvolvida para o sistema de supervisão (SCADA), utilizando o InTouch Wonderware. A IHM possibilita o ajuste dos parâmetros de calibração de velocidade do motor DC e monitoramento e execução de rotinas de ação, para estimular o ensino do conceito de sistema supervisório e a importância da interface gráfica nas aplicações de automação. A representação dos objetos de tela se deu de forma animada com a finalidade de proporcionar a visualização de como de fato deve estar acontecendo o processo.

Este projeto foi desenvolvido, pela aluna Andréa de Andrade Mendes Cabral no Laboratório de Instrumentação Eletrônica e Controle (LIEC), situado na Universidade Federal de Campina Grande (UFCG), sob a orientação do professor George Acioli Júnior e supervisão do professor Péricles Rezende Barros e foi dividido nas seguintes etapas:

Etapa 1: Revisão bibliográfica;

Etapa 2: Desenvolvimento da especificação;

Etapa 3: Desenvolvimento da IHM;

Etapa 4: Testes e Redação da Documentação do Projeto;

Etapa 5: Redação do Relatório do projeto do Trabalho de Conclusão de Curso (TCC)

2 ESTEIRA TRANSPORTADORA

O objeto de estudo foi uma esteira transportadora de tamanho reduzido construída pelo aluno Ezequiel com finalidade de desenvolver seu Trabalho de Conclusão de Curso. [1]

A estrutura física da esteira transportadora foi reaproveitada de uma esteira da marca CALOI modelo FITNESS.

Na figura 1, identifica-se os principais elementos que compõem a esteira, que são: 1 - motor DC; 2 - encoder 3, 4, 5 e 6 - sensores ópticos de barreira; 7 - câmara de resfriamento; 8 - câmara de aquecimento e 9 - fonte de alimentação.

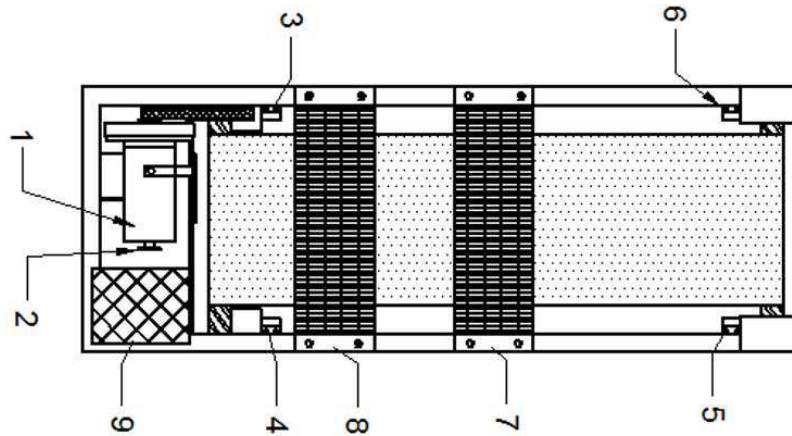


Figura 1 - Esteira Transportadora.
Fonte: [1]

As entradas e saídas da esteira estão conectadas a um CLP e podem ser controladas e/ou medidas.

2.1 MOTOR DC

O modelo do motor (Figura 2) é o original da esteira CALOI e possui as seguintes especificações:

- Modelo: MOTOR PMDC;
- Tensão: 180 V DC (podendo operar com tensões inferiores);
- Potência: 1HP;
- Fabricante: Universal Motors (Indústria Brasileira)



Figura 2 - Motor DC.
Fonte [1]

No programa em Ladder, o motor DC foi acionado para mover-se no sentido horário ou anti-horário, com diferentes velocidades, seguindo os valores da saída analógica S0000, com *tag* MIV-S-301, entre 0 e 4095, o qual fornece um valor de tensão (0 - 10V).

2.2 ENCODER

A esteira possui um encoder incremental, Figura 3, para determinar a velocidade do Motor CC. O modelo do encoder corresponde ao "ww - 12 encoder" produzido pela empresa NU-BOTICS.

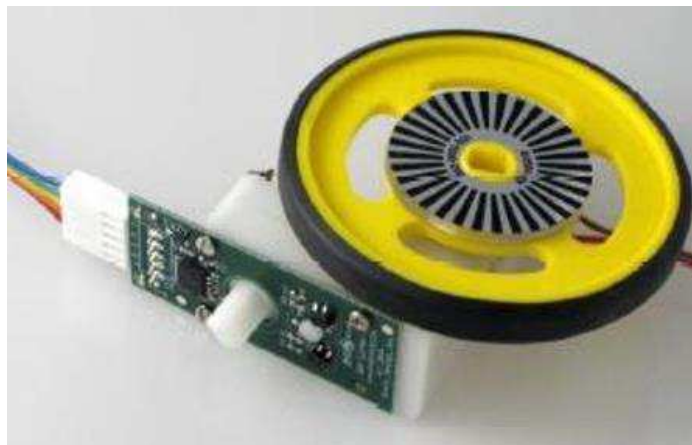


Figura 3 - Encoder.
Fonte: [2]

2.3 SENSORES ÓPTICOS DE BARREIRA

No sensor óptico de barreira que a esteira possui, Figura 4, foi utilizado como elemento emissor um LED alto brilho azul e, como elemento receptor, um foto transistor TIL81 com circuito dedicado.



Figura 4 - Sensor óptico de barreira.
Fonte: [1]

Os sensores de barreira foram conectados às entradas digitais I0008 a I0011 do CLP.

2.4 MÓDULO DE AQUECIMENTO

Como pode-se ver na Figura 5, o módulo de aquecimento possui duas resistências de 1800 W e dois *coolers*, responsáveis pela variação térmica no interior da câmara, e um sensor de temperatura LM35, para realizar a medição da temperatura. Além disso, existe um sensor de barreira, o par emissor e receptor.

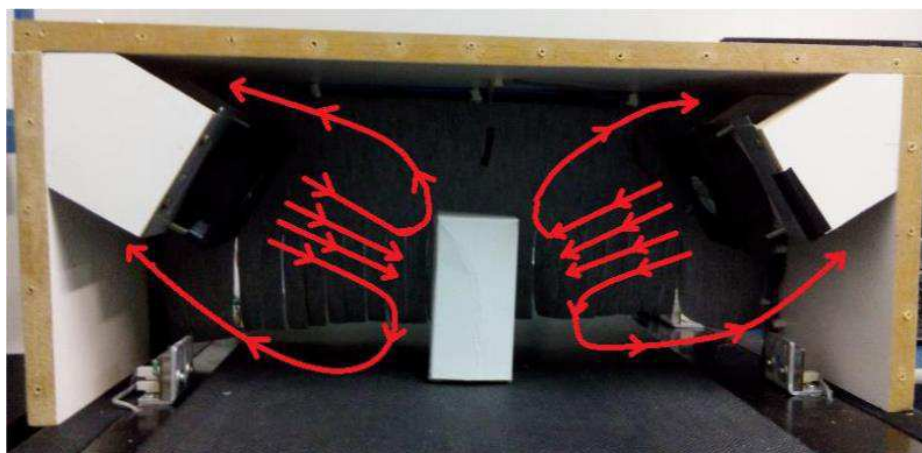


Figura 5 - Módulo de aquecimento.
Fonte: [1]

Pode-se identificar no código carregado no CLP, Figura 11, que a *tag* TT20 refere-se à leitura de tensão do sensor de temperatura enviada pela entrada digital E0006. Esse valor varia entre 0 mV e +10 mV e necessita de conversão (1) para unidade de engenharia (°C).

$$T_{°C} = \frac{10 * T_{lido}}{409,5} \quad (1)$$

2.5 MÓDULO DE RESFRIAMENTO

O módulo de resfriamento, Figura 6, é composto por um sensor de temperatura LM35 e dois *coolers*. A leitura da temperatura do sensor e o acionamento dos *coolers* são realizados utilizando uma entrada analógica e uma saída digital, respectivamente.



Figura 6 - Módulo de Resfriamento.
Fonte [1]

A leitura da temperatura do módulo de resfriamento é análoga à do módulo de aquecimento, inclusive a conversão de unidade (1).

O sensor LM35 está conectado ao CLP pela entrada analógica E0007, *tag* TT10. Após a conversão o valor é armazenado na variável TF CELSIUS.

Os *coolers* são acionados simultaneamente e sem controle da velocidade de rotação, pela saída digital O0010, identificada pela *tag* VENTFRIO.

3 TUTORIAL INTOUCH

Neste capítulo serão apresentadas as ferramentas de software utilizadas no projeto da IHM. Fundamentando algumas das principais funcionalidade do InTouch, tais como: a criação das telas, criação de gráficos de tendência e alarmes, uso de animações e script utilizados no tratamento dos dados que serão apresentados ao operador. [3]

Modos de Operação

O InTouch possui dois modos de operação (Figura7): Modo de desenvolvimento (window maker) e Modo de execução (Window Viewer).

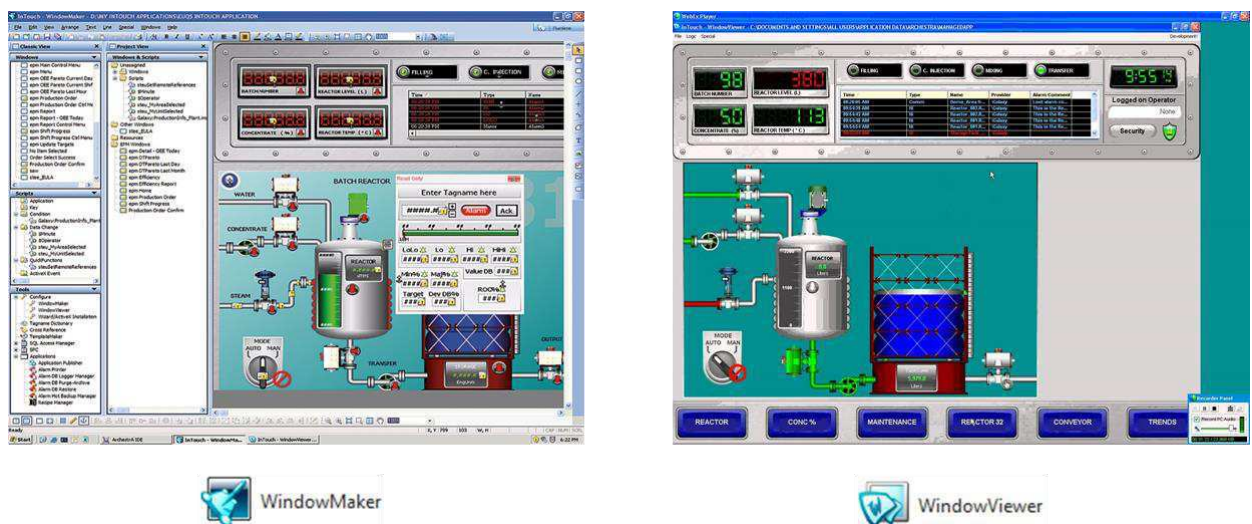


Figura 7: Modos de operação do InTouch.

O Modo de desenvolvimento é destinado à criação das aplicações e janelas, e edição das animações, ou seja, é a área para construção da IHM. Enquanto o Modo de execução apenas executa as telas produzidas no Window Maker. [4]

Objetos de tela básicos

Os objetos de tela básicos servem para criar desenhos primitivos, textos, gráficos de tendência e botões. É possível realizar diversas formas de animação, mas esses são os elementos mais simples do programa. Na Figura 8 tem-se uma lista com os objetos de tela e seus ícones.













Objeto	Descrição
	Retângulo
	Retângulo com bordas arredondadas
	Elipse
	Linha
	Linha vertical ou horizontal
	Seqüência de linhas
	Polígono
	Texto
	Importação de bitmap
	Gráfico de Tendência Real
	Gráfico de Tendência Histórica
	Botão

Figura 8: Objetos de tela básicos.
Fonte[3]

Wizard: objetos de tela especiais

Quando deseja-se trabalhar com elementos mais complexos e desenhos mais semelhantes aos das plantas, ambientes de operação, pode-se utilizar os objetos de tela do Wizard. No Wizard encontram-se os objetos de tela especiais que, dependendo do elemento, também podem ser animados. Na Figura 9, pode-se ver a janela do Wizard.

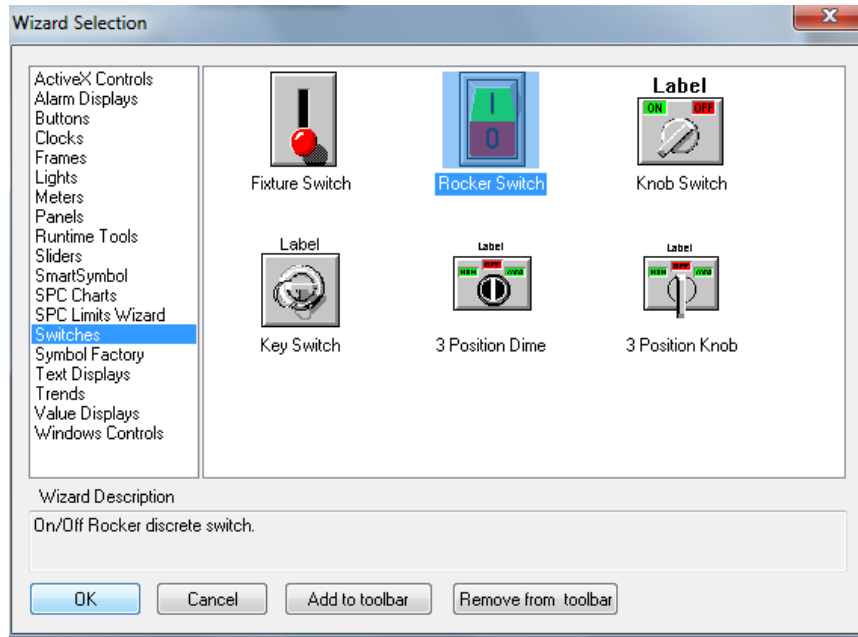


Figura 9: Objetos de Tela Especiais – Wizard.

Tagnames

As variáveis do InTouch são denominadas tagnames e podem ser numéricas ou alfanuméricas, de memória ou I/O. Alguns tipos principais são descritos na figura 10.

TIPO	DESCRIÇÃO	EXEMPLO DE APLICAÇÃO
Discrete	Variável que possui apenas dois níveis, 0 ou 1, ativada ou não ativada, ligado ou não ligada.	Bombas, válvulas on/off, lâmpadas, alarmes, etc.
Integer	Variáveis inteiras, ou seja, números inteiros (conjunto Z)	Indicações inteiras, saídas inteiras, etc.
Real	Variável real, ou seja, conjunto R.	Indicações reais, saídas reais, etc.
Message	Variável alfanumérica acumula números e/ou letras.	Informações que podem ser números e/ou letras.
Hist Trend	Variável do gráfico de tendência histórica. Cada gráfico necessita de uma. O gráfico de tendência histórica é o equivalente ao registrador, com as vantagens da carta não "embolar", a tinta não "borrar" ou acabar, não ocupar espaço no painel, além de poder ser criado diversas vezes.	Gráfico de tendência histórica e wizard.
I/O	Variável usada para trocar informações com os equipamentos de aquisição de dados, usando drivers de I/O fornecidos pelo fabricante do supervisor de acordo com o tipo do equipamento utilizado.	PLC, Controladores digitais, etc

Figura 10: Tipos de tagnames.

Fonte [3]

Quando cria-se uma tag, é necessário informar o tipo da variável e, caso ela seja I/O, o seu Access Name e o Item. Existe um dicionário de tags, Tagname Dictionary, onde pode-se criar e editar tags.

Animação de objetos

As tagnames fazem o elo entre o processo e a animação. É apresentada na Figura 11 a janela destinada a animação de objetos, nesse caso é um objeto do tipo retângulo.

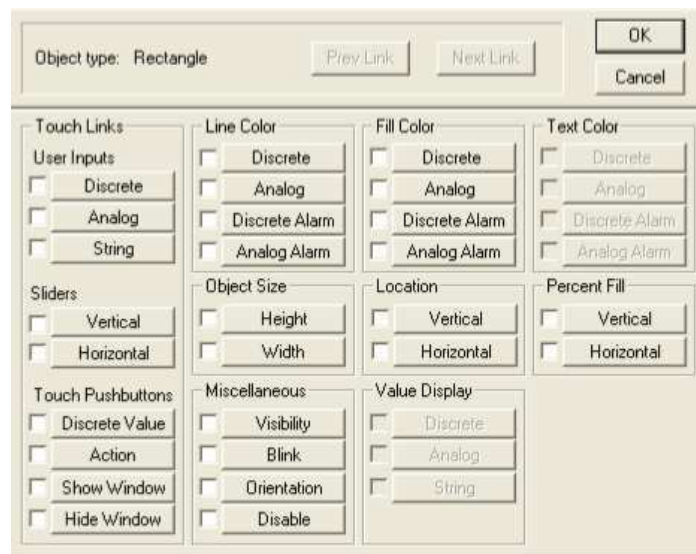


Figura 11: Animação de objetos.
Fonte [3]

Pode-se listar essas funcionalidades com o objetivo de especificar seu uso.

- ❖ Touch Links – Interação com o usuário através de:
 - Entradas (User Input)
 - Sliders
 - Ações por botões (Touch pushbottoms)
- ❖ Line Color – Coloração de linha.
- ❖ Object Size – Tamanho do objeto.
- ❖ Miscellaneous – Animações de diversas: visibilidade, pisca, rotação e indisponibilidade do objeto.
- ❖ Fill Color – Coloração do obejto.
- ❖ Location – Posicionamento do objeto.
- ❖ Value Display – Valor a ser visualizado.

- ❖ Text Color – Coloração do texto.
- ❖ Percent Fill – Percentagem de preenchimento do objeto

Script

Scripts são módulos em linguagem, onde se pode definir linhas de código como na linguagem Basic. Todos os Scripts do Intouch são orientados a eventos e eles podem ser resultantes de mudança do dado, condicional, pelo clique do mouse ou timer, por exemplo. Na Tabela 1, descreveu-se os tipos de script.

Tabela 1: Tipos de script.

Tipo	Descrição
Application	Ligado a aplicação inteira
Window	Ligado a uma janela específica
Key	Ligado a uma tecla específica
Condition	Ligado a uma tag discreta ou expressão
Data Change	Ligado a uma tagname e/ou tagname.field APENAS
Action Push button	Associado a um objeto que se liga a um Touch
Link	Ligado a um botão

Alarmes

Alarmes são usados para sinalizar algum tipo de problema que está ocorrendo na planta, para então tomar as ações apropriadas. Por exemplo, para um tanque pode-se atribuir um alarme relacionado ao nível. Quando o nível estiver muito baixo, de acordo com a especificação do problema, ativa o campo LoLo do alarme. Na mesma forma, atribui-se os níveis baixo, alto e extremamente alto aos campos Low, High e HiHi, respectivamente. Os campos de alarme foram descrito na Figura 12.

campo	descrição
LoLo	Alarme Baixo Crítico. Define um intervalo de valores (menor igual) onde o Tag é considerado em um estado de Alarme Baixo Crítico. É usado quando o valor do Tag está abaixo de um mínimo, ou seja, extremamente baixo..
Low	Alarme Baixo. Define um intervalo de valores (menor igual) onde o Tag é considerado em estado de alarme baixo. É usado quando o valor do Tag está abaixo do normal.
High	Alarme Alto. Define um intervalo de valores (maior igual) onde o Tag é considerado em estado de Alarme Alto. É usado quando o valor do Tag está mais alto do que o normal.
HiHi	Alarme Alto Crítico. Define um intervalo de valores (maior igual) onde o Tag é considerado em estado de Alarme Alto Crítico. É usado quando o valor do Tag é está acima de um máximo, ou seja, extremamente alto.

Figura 12: Campos de alarmes.
Fonte [3]

Quando trabalha-se com mais de uma tag com alarme, deve-se ordená-las por prioridade. A prioridade varia entre 1 e 999, sendo 999 a maior prioridade.

Janelas

Dependendo do processo, é comum precisar de mais de uma tela e, para garantir a fluidez, é necessário ter uma forma de mudar de janela durante o modo de execução. Para isso, pode-se adicionar botões que levem de uma janela para outra.

Também há casos nos quais deseje-se que apareça uma determinada janela em situações de alarme, então, basta relacionar alarmes a janelas.

Pode-se configura a janela para que seja do tipo Popup, entrão, ao executar a aplicação, abre essa janela específica primeiro.

4 DESENVOLVIMENTO DA IHM

O supervisório é um software destinado a promover a interface homem/máquina (IHM), proporciona uma supervisão plena de seu processo por meio de telas previamente configuradas. As telas que representam o processo, onde estas podem ser animadas em função das informações recebidas pelo CLP.

Quando se tratando de SCADA (Supervisory Control and Data Acquisition), existem algumas definições necessárias, tais como:

- Monitoração: exibir os dados básicos em tempo real.
- Supervisão: possibilitar alterações e solicitações de processo.
- Alarmes: reconhecimento de eventos excepcionais e relatá-los.
- Controle: capacidade de ajuste de valores do processo.

As principais tarefas de um sistema supervisório são leitura e escrita de variáveis e gerenciamento de dados. A leitura e escrita de variáveis está relacionada à retirada ou inserção de informações do processo através da conexão do computador com o clp, controlador do processo. Enquanto o gerenciamento de dados, se refere a como os dados serão tratados e exibidos para o usuário, em tempo real de execução, por meio de telas, relatórios, gráficos de tendência, etc. [5]

Este capítulo apresenta a concepção e desenvolvimento de duas aplicações de interface gráfica, Interface Homem-Máquina (IHM), que integrará o software de um sistema supervisório empregado no monitoramento de variáveis de leitura e/ou escrita, uma denominada calibração da velocidade do motor e a outra, aplicação de monitoramento e execução de rotinas de ação. O método adotado neste processo é descrito a seguir.

4.1 LEVANTAMENTO DE REQUISITOS E ESPECIFICAÇÕES DA IHM

Escolher um bom método de organização para o desenvolvimento da forma de interação homem-máquina é de extrema importância para a usabilidade do sistema, pois, mesmo que o projeto seja bem elaborado do ponto de vista da codificação, se o usuário não souber utilizá-la de forma eficiente, torna-se incompleto. Portanto, optou-se por utilizar um método aprendido durante o curso da disciplina de Informática Industrial, da ênfase de controle, para a especificação do projeto.

O Método para Concepção de Interfaces Ergonômicas, ou MCIE, é um método que apoia a concepção e o desenvolvimento de interfaces bem estruturadas que atendem as necessidades do usuário e do cliente. O método MCIE em diversas etapas, escolheu-se três delas para resumir a documentação: Objetivos de usabilidade, levantamento do perfil de usuário e análise da tarefa. [6]

4.1.1 OBJETIVOS DE USABILIDADE

O método se inicia pela análise de requisitos, onde o desenvolvedor deve analisar o contexto no qual a IHM será inserida, as características dos usuários que utilizaram regularmente a interface, e após ter conhecimento dessas especificações o desenvolvedor pode desenvolver o modelo da tarefa para o projeto, especificando detalhadamente cada tarefa que será realizada pela IHM.

Primeiramente, a interface deve ser intuitiva de forma que seja dispensável treinamento dos usuários. Sendo intuitiva, é de fácil memorização. Além de intuitiva, a interface será composta por animações, visto que objetivo é proporcionar ao usuário a ideia de movimento e monitoramento em tempo real. É simples e pouco profundo, sendo assim, é de rápida interação. O sistema também deve ser rápido, visto que as grandezas devem ser monitoradas em tempo real.

4.1.2 LEVANTAMENTO DO PERFIL DO USUÁRIO

Com a finalidade de traçar as características do usuário, fez-se o preenchimento de um formulário do ponto de vista dos usuário, aluno e professor. Os formulários constam em anexo, Anexo A.

Diz respeito as características do usuário que são escolhidas pelo projetista, de acordo com a relevância para o projeto. Neste caso, foram analisadas as característica: idade, sexo, motivações, frequência de utilização, experiência com o sistema, conhecimento da tarefa realizada pela IHM, estilo cognitivo, habilidades motoras e de percepção.[6]

No método MCIE, o levantamento do perfil do usuário pode ser apoiado por formulários, por observação, ou por questionário. Neste projeto foram identificadas as características a seguir:

Usuário: Aluno

- Idade: a partir de 20 anos;
- Sexo: sem predominância;
- Objetivo: monitorar variáveis de interesse e executa rotinas de ação.
- Frequência de utilização: Baixa;

- Aprendizado: por acesso à documentação e orientação de professores ou monitores;
- Estilo cognitivo: Médio;
- Nível de percepção visual (Média).

Usuário: Professor

- Idade: a partir de 25 anos;
- Sexo: sem predominância;
- Objetivo: Ensinar os alunos o conceito de sistemas supervisórios de forma prática.
Frequência de utilização: Baixa;
- Aprendizado: treinamento informal;
- Estilo cognitivo: Persistência (Elevada);
- Nível de percepção visual (Alta).

4.1.3 ANÁLISE DA TAREFA

De acordo com o MCIE [6], a análise da tarefa envolve a compreensão e representação das ações e sequências de ações necessárias para realização da tarefa.

Os modelos das tarefas propostos para este projeto fornece a descrição detalhada das tarefas, subtarefas e métodos envolvidos na utilização do sistema supervisório. Estes modelos estão representado nas figuras 13 e 14.

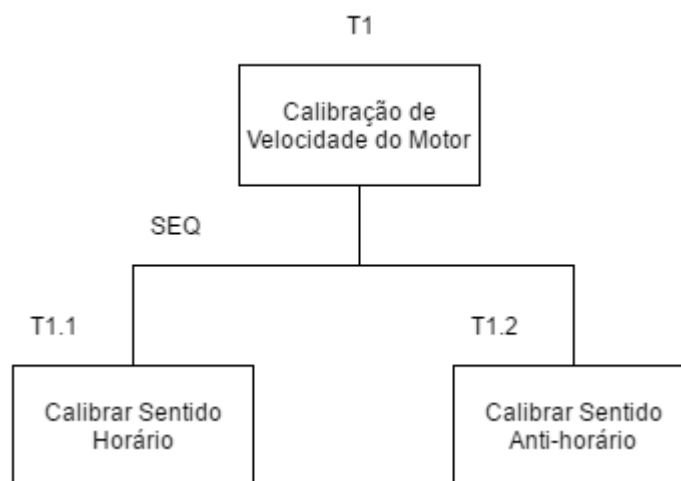


Figura 13: Modelo de tarefa para aplicação Calibração do Motor.

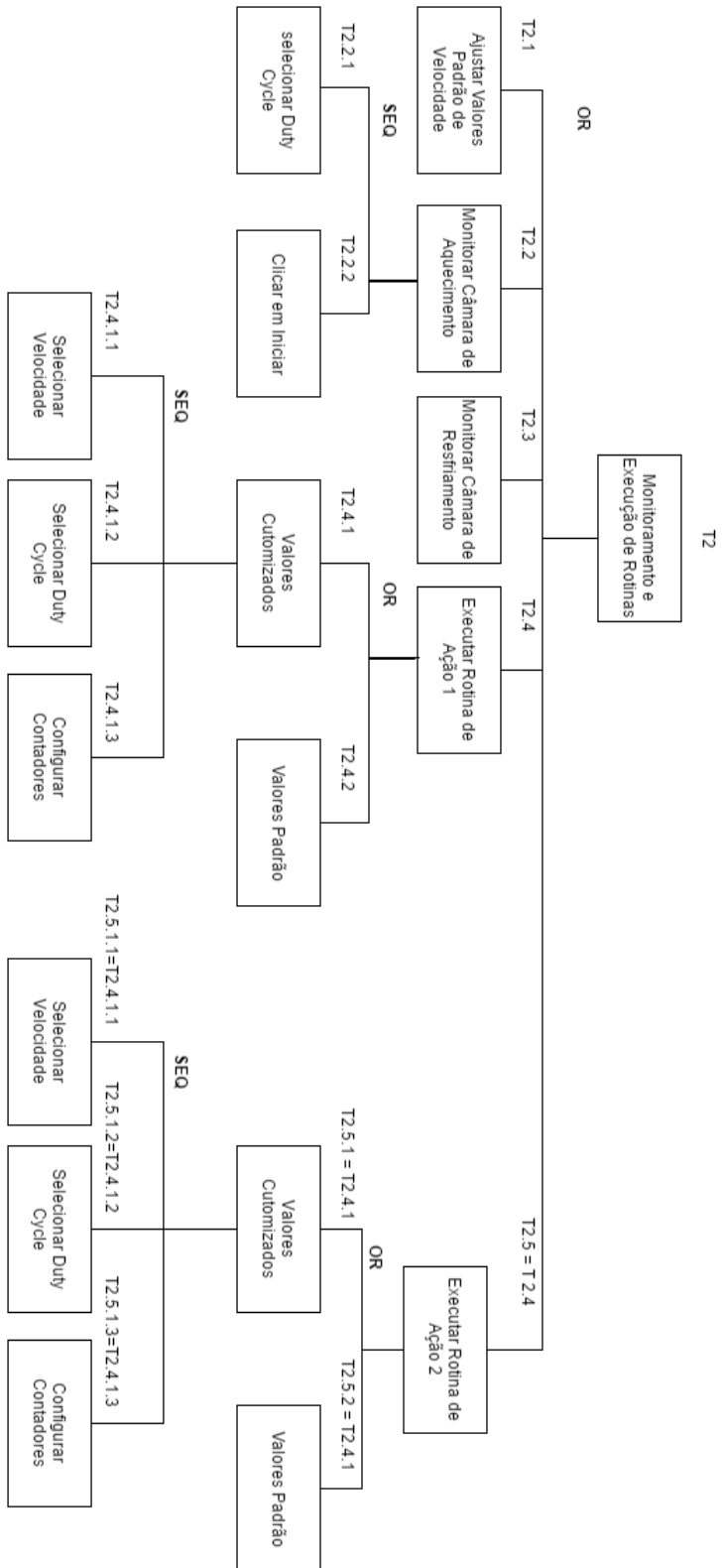


Figura 14: Modelo de tarefa para aplicação Monitoramento e Execução de Rotinas.

4.2 DIAGRAMAS UML

O UML (Unified Modeling Language) [7] é uma linguagem padrão para elaboração de modelos em projetos de software, e que possui diagramas que podem ser combinados com a finalidade de obter todas as visões e aspectos do sistema. Os principais objetivos dos diagramas UML são:

- Visualizar o sistema como ele é ou como deseja que ele seja.
- Especificar a estrutura ou o comportamento de um sistema.
- Proporcionar um guia para construção do sistema.
- Documentar o sistema.

4.2.1 DIAGRAMAS DE CASOS DE USO

O diagrama de caso de uso é um diagrama dinâmico que modela o comportamento do sistema percebido por atores externos. Caracteriza um objetivo final a ser alcançado pelo ator externo ao utilizar o sistema. Nos diagramas das figuras 15 e 16, pode-se observar que qualquer usuário pode acessar as telas.

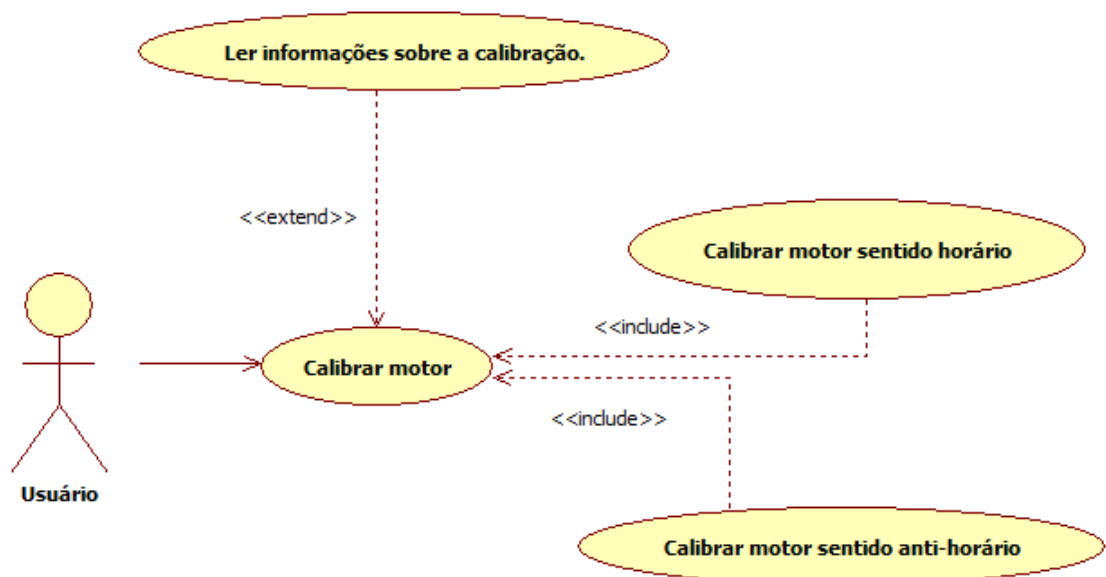


Figura 15: Diagrama de caso de uso – Calibração de Motor.

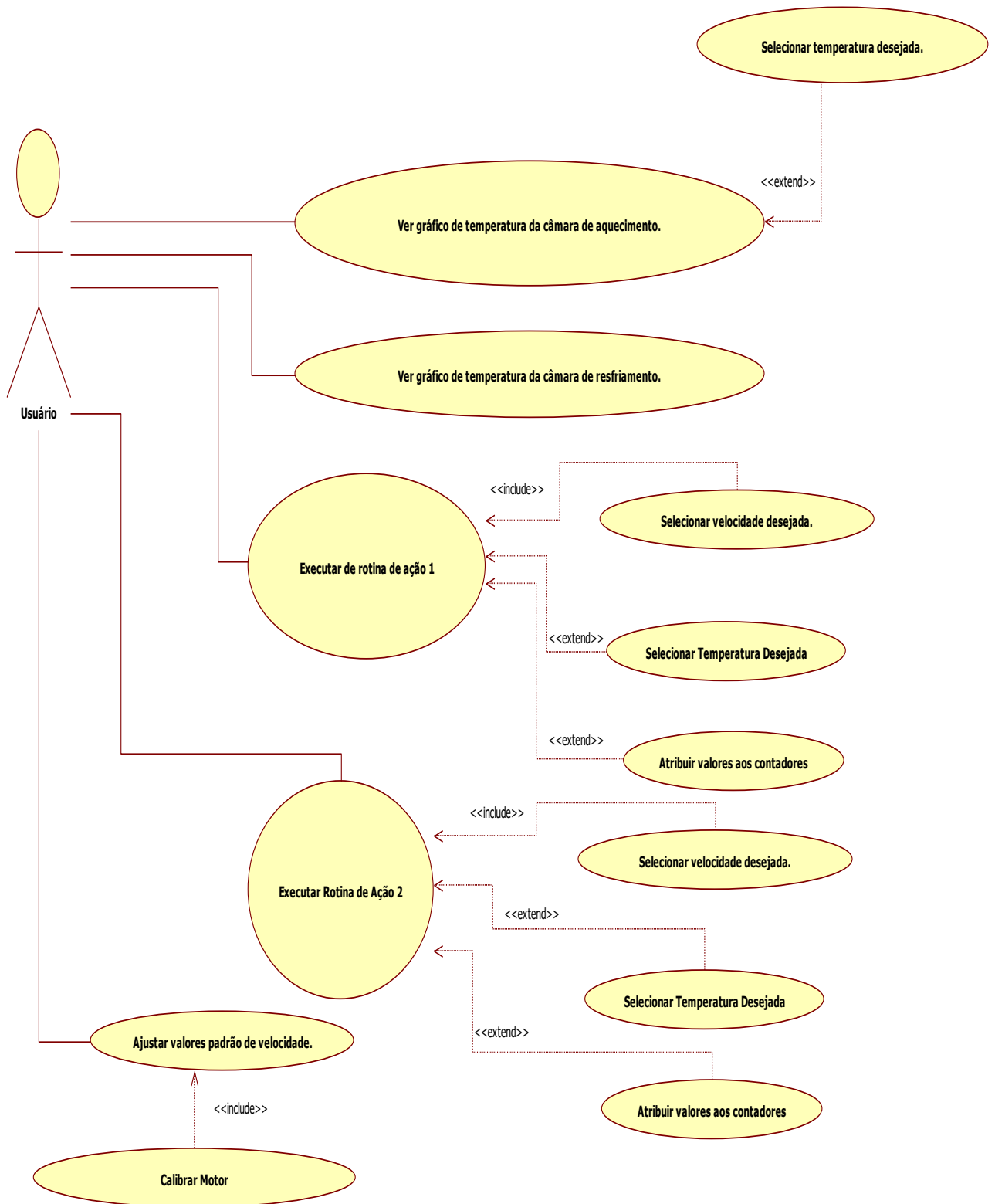


Figura 16: Diagrama de Caso de Uso da aplicação Monitoramento e execução de rotinas.

4.2.2 DIAGRAMAS DE SEQUÊNCIA

Os diagramas de sequência permitem a modelagem de processos através da troca de mensagens, CALL e RETURN, entre os objetos do sistema. Cada objeto é uma linha vertical, e as mensagens são setas que partem do objeto que invoca um outro objeto.

Nos diagramas de sequência nas ilustrados nas figuras seguintes são apresentados dois cenários de excessão, calibração do motor (Figura 17) e Execução da Rotina de Ação 1 com ajuste de parâmetros de calibração de velocidade (Figura 18), e um cenário de rotina, Execução de Rotina de Ação 2 (Figura 19).

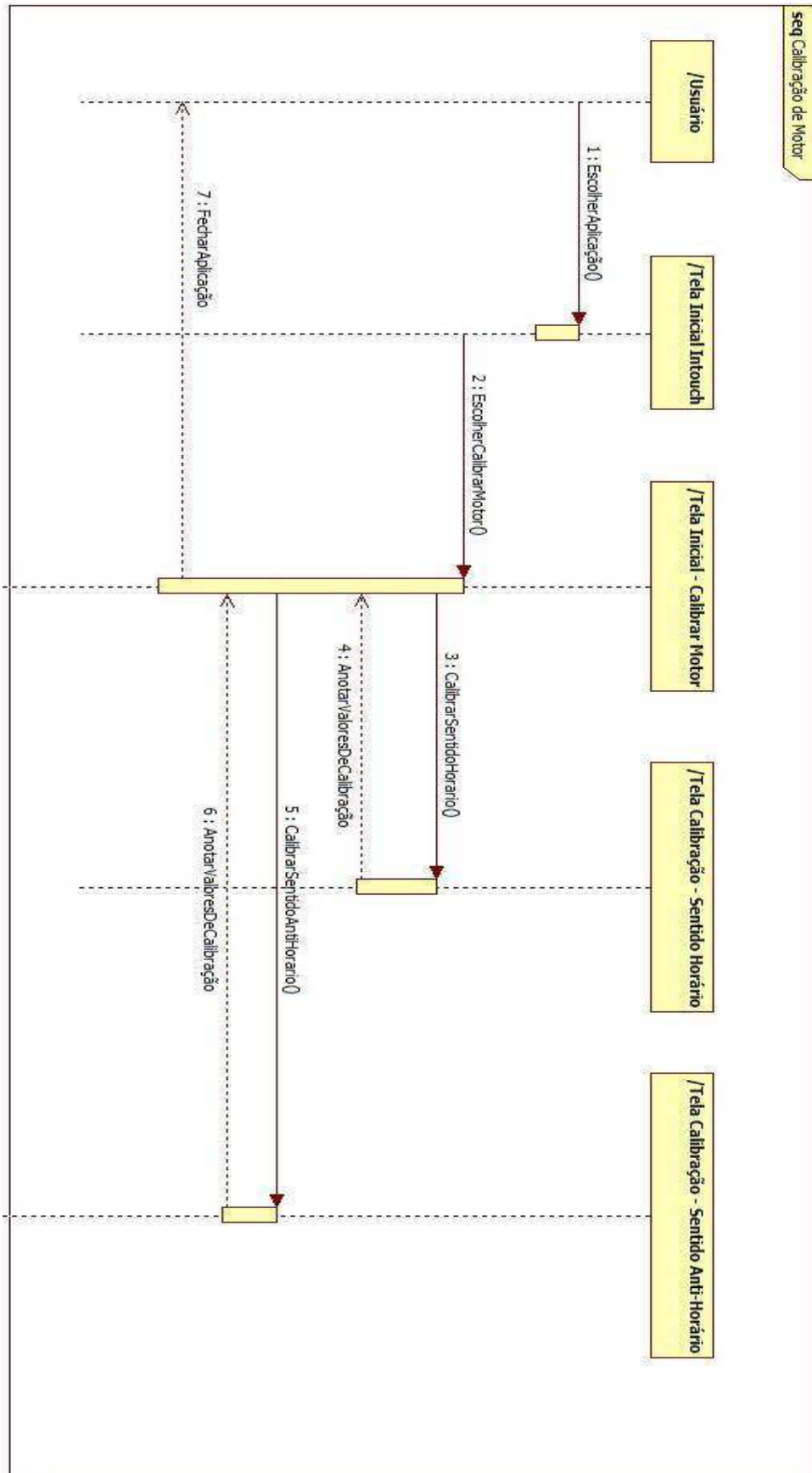


Figura 17: Calibração de motor – Diagrama de Sequência.

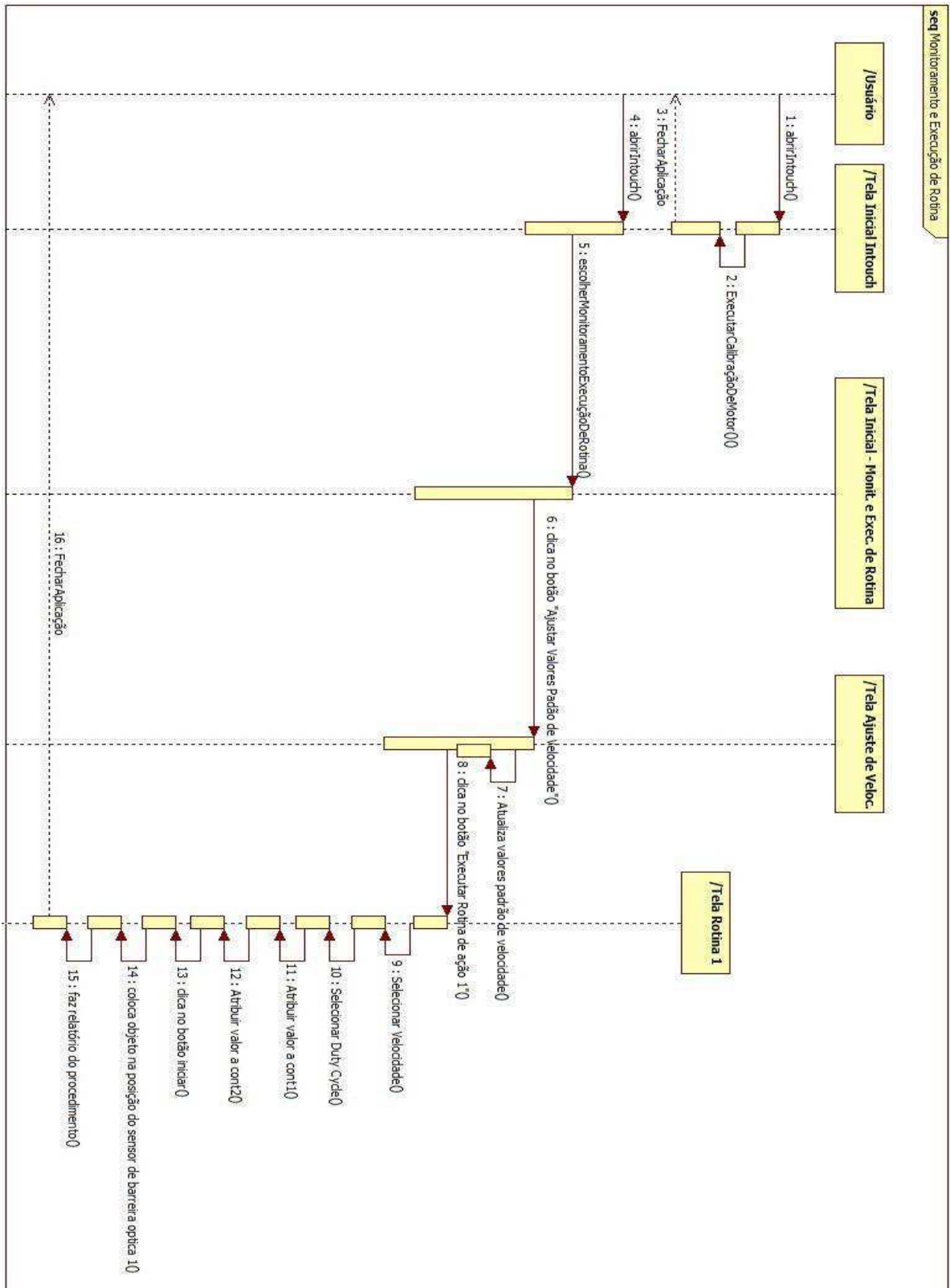


Figura 18: Execução de Rotina de Ação 1 com calibração de motor – Diagrama de Sequência.

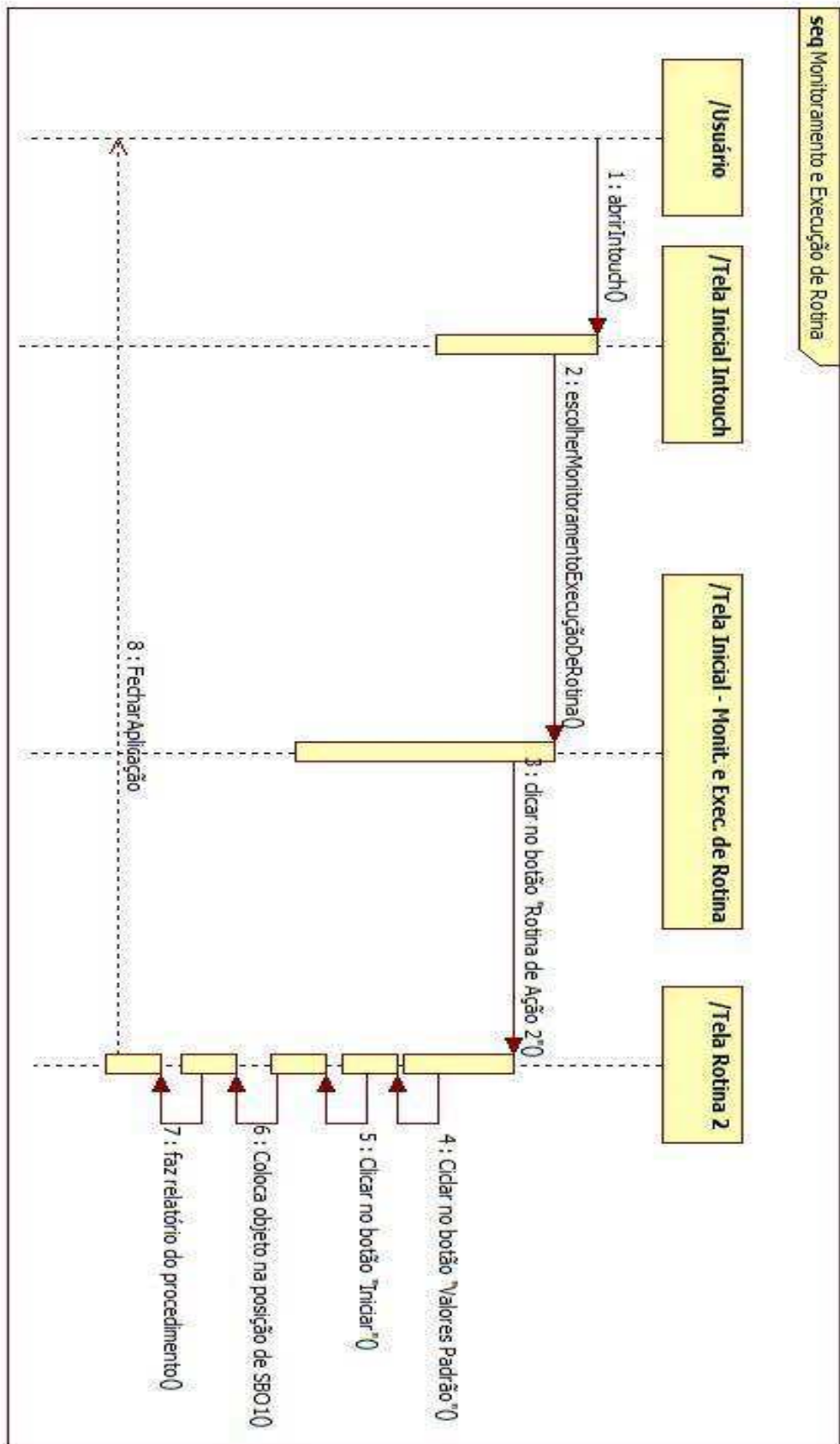


Figura 19: Execução de Rotina de Ação 2 – Diagrama de Sequência.

4.3 DESENVOLVIMENTO DE TELAS

As telas foram divididas em duas aplicações: uma para a operação de excessão e outra para operação de rotina. A aplicação de excessão é destinada à calibração do motor. Enquanto que a aplicação de rotina é para monitoramento da temperatura das câmaras de aquecimento e resfriamento e localização e para execução de rotinas de ação.

4.3.1 CALIBRAÇÃO DA VELOCIDADE DO MOTOR

A aplicação Calibração da Velocidade do Motor é uma operação de excessão destinada à execução de uma subrotina para gerar parâmetros de calibração. Esses parâmetros são valores da entrada digital do motor em função da velocidade de rotação da lona da esteira. Devido as dimensões da esteira, foi conveniente utilizar a unidade cm/s para a velocidade.

As velocidades máximas no sentido horário e anti-horário são 22 cm/s e 20 cm/s, respectivamente. Mas, após diversos testes, observou-se que a faixa de velocidade ideal para a aplicação era entre 1 cm/s e 5 cm/s, como será explicado melhor no tópico 4.3.2. Logo, para otimizar o tempo de execução, dividiu-se em faixas variando entre 950 e 2250, para o sentido horário e 2550 e 2850, para o sentido anti-horário de forma a proporcionar apenas velocidades dentro do range desejado.

Além disso, desenvolveu-se uma tela destinada à informar ao usuário quando e para que efetuar a calibração do motor.

Portanto, esta aplicação é dividida em:

- Janela Inicial: Calibração do Motor;
- Sobre Calibração;
- Calibração do Motor Sentido Horário;
- Calibração do Motor Sentido Anti-horário.

Janela Inicial

A janela inicial é destinada a observação dos valoves de calibração nos sentidos horário e anti-horário e ao acesso de informações sobre a calibração.

Nela existem três botões O botão “Sobrecalibração”, “calibrar sentido horário” e “Calibrar sentido anti-horário”, que encaminham às janelas com informações sobre como e

quando deve ser feita a calibração, destinada a calibração no sentido horário e para executar a calibração no sentido anti-horário, respectivamente, e um painel para exibição de valores obtidos na calibração. Inicialmente esses valores encontram-se nulos e recebem valores de variáveis auxiliares que são manipuladas em subrotinas, como pode-se ver na Figura 20.

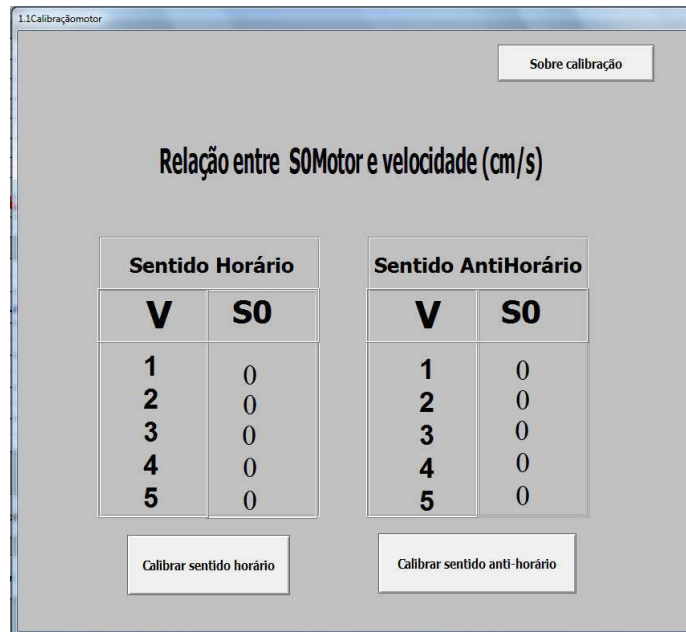


Figura 20: Janela inicial da aplicação Calibração do Motor.

Sobre Calibração

A janela “Sobre Calibração” tem o objetivo de explicar o que é a calibração e informar ao usuário quando deverá ser feita a calibração, Figura 21.

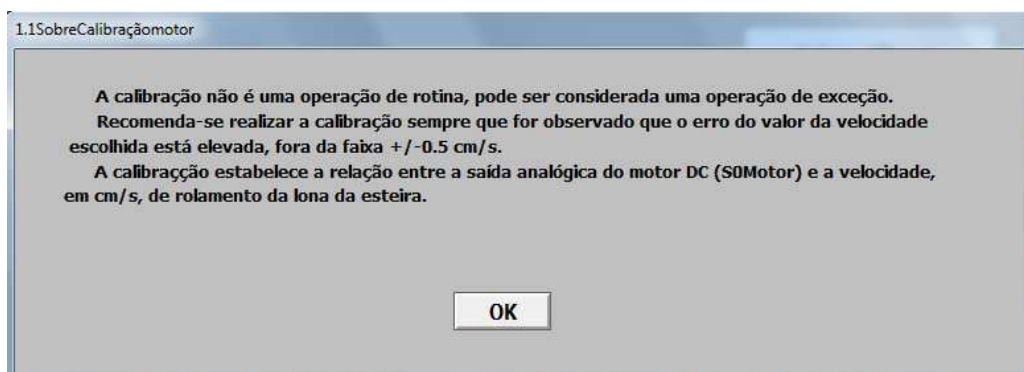


Figura 21: Janela Sobre Calibração.

Para retornar à janela inicial, basta clicar no botão “OK”.

Calibração Sentido Horário

A tela “Calibração Sentido Horário” é composta por dois gráficos de tendência real, S0Motor e Velocidade (cm/s), dois botões, “Iniciar calibração” e “Voltar”, e cinco displays para exibir os parâmetros de calibração obtidos no experimento. Olhar Figura 22.

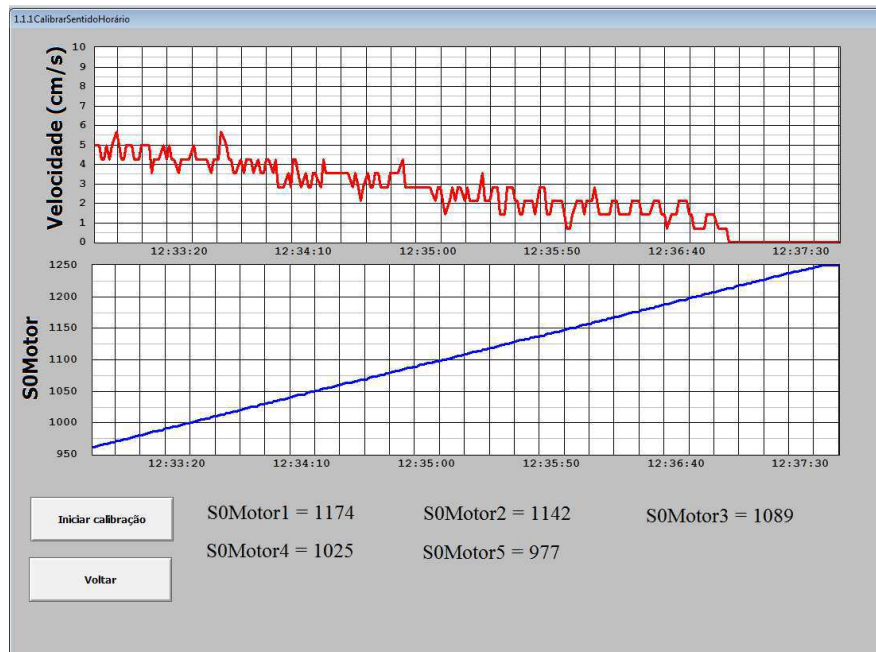


Figura 22: Tela Calibração Sentido Horário.

O gráfico de tendência real “velocidade” exibe os valores da velocidade em cm/s. Esses valores estão associados, por meio do “window script” a valores de S0Motor e variam com o tempo.

Configurou-se o gráfico de tendência real para apresentar valores da tagname “velocidade” variando entre 0 e 10, pegando uma amostra a cada segundo, cobrindo 300 segundos, como mostra a Figura 23.

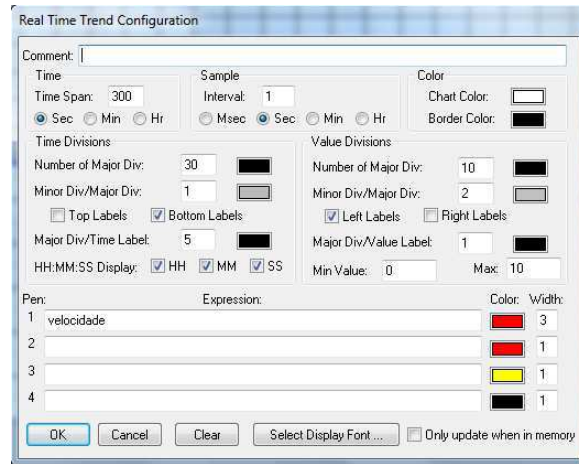


Figura 23: Configuração dos parâmetros do gráfico de tendência velocidade – Sentido Horário.

Enquanto isso, o gráfico de tendência real “S0Motor” exibe os valores da tagname “S0Motor” sendo amostrado a cada segundo e com Span de 300 segundos, Figura 24.

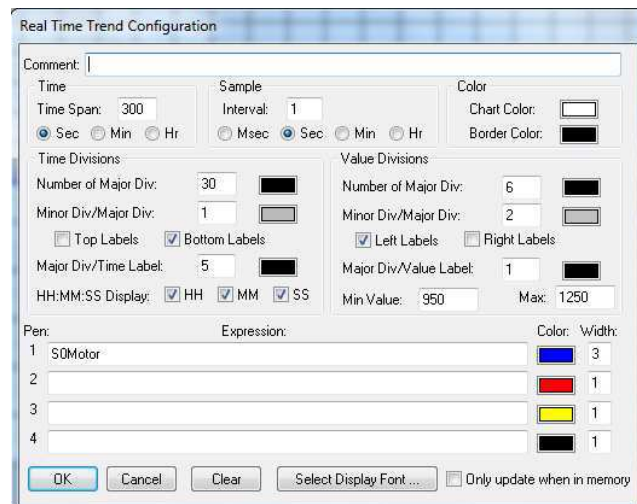


Figura 24: Configuração dos parâmetros do gráfico de tendência S0Motor – Sentido Horário.

As janelas podem ser programadas por meio de scripts. Existem três modos para o “Window Script” que são “On Show”, “While Showing” e “While Hide”.

A “Condition Type” “On Show” é executada ao abrir a janela e, para a janela “Calibrar Sentido Horário”, é responsável pela inicialização das variáveis. Ver Figura 25.

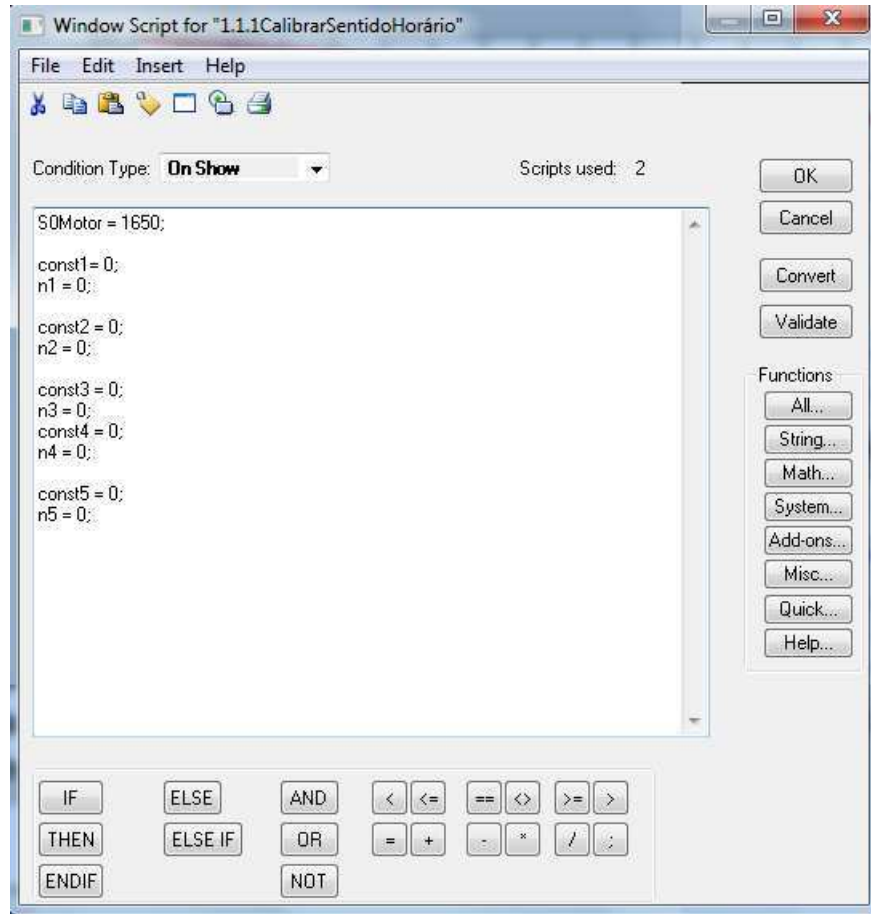


Figura 25: Window Script da janela Calibração Sentido Horário – On Show.

A “Condition Type” “While Showing” é executada enquanto a janela “Calibrar Sentido Horário” estiver sendo executada. Esse script é executado a cada segundo e é responsável por fazer uma varredura em valores de S0Motor entre 950 e 2250, como mostra a Figura 26.

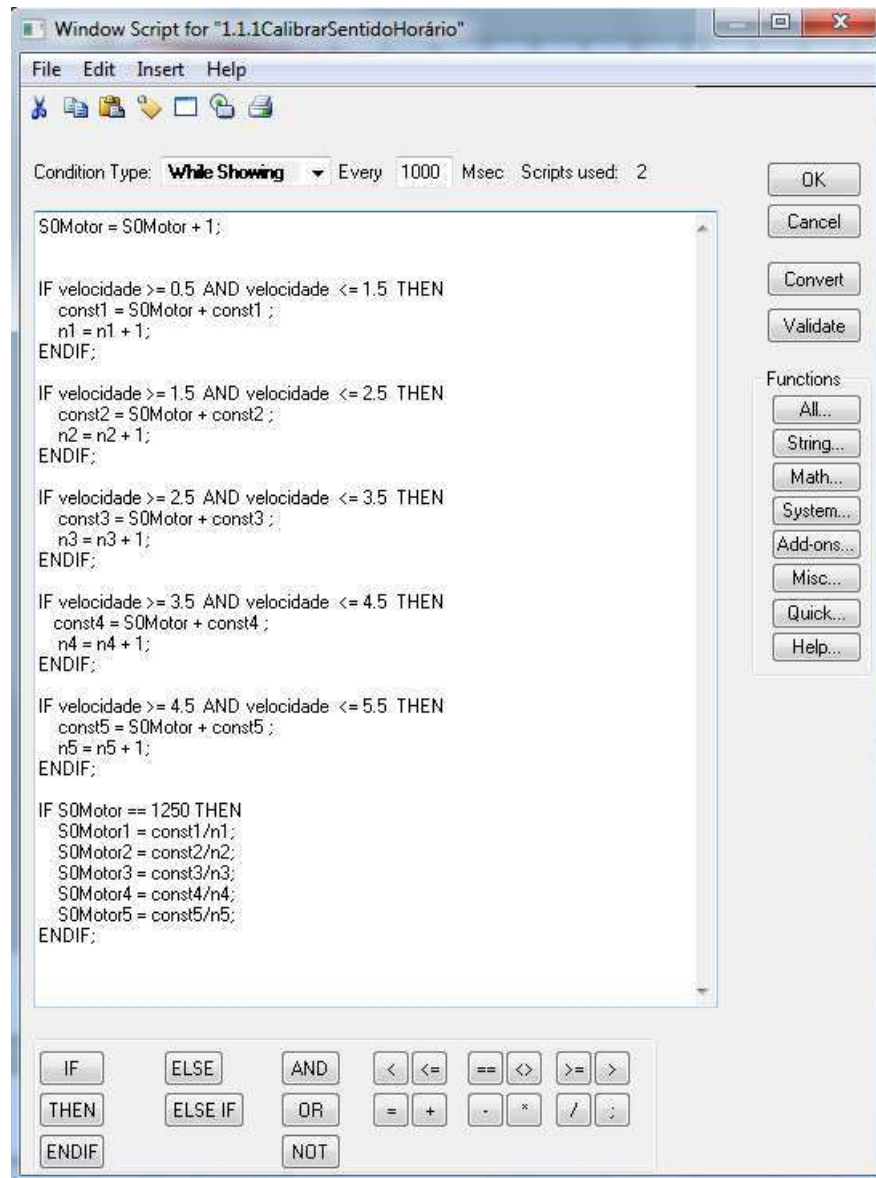


Figura 26: Window Script da janela Calibração Sentido Horário – While Showing.

Como mencionado no início deste tópico, a faixa de velocidade ideal está entre 1 cm/s e 5 cm/s. Então, discretizou-se essa faixa para os valores inteiros, sendo que, para cada valor inteiro, tinha tolerância de $\pm 0,5$ cm/s.

As constantes const1, const2, const3, const4 e const5 acumulam os valores de SOMotor para a faixa especificada, enquanto n1, n2, n3, n4 e n5 funcionam como um contador para o número de elementos registrados em suas respectivas faixas de valores.

Ao final da execução dessa rotina, ou seja, quando SOMotor fosse igual 1250, atribuiu-se às variáveis de calibração SOMotor1, SOMotor2, SOMotor3, SOMotor4 e SOMotor5 os valores das constantes acumuladoras divididas pelo número de elementos registrados nas faixas correspondentes.

O botão “Iniciar Calibração” está associado a uma subrotina que dá início, ou reinicia, à execução da calibração no sentido horário. Essa subrotina atribui os valores iniciais das variáveis, valor nulo às variáveis auxiliares e o valor inicial de 950 à S0Motor. Como pode-se ver na Figura 27.

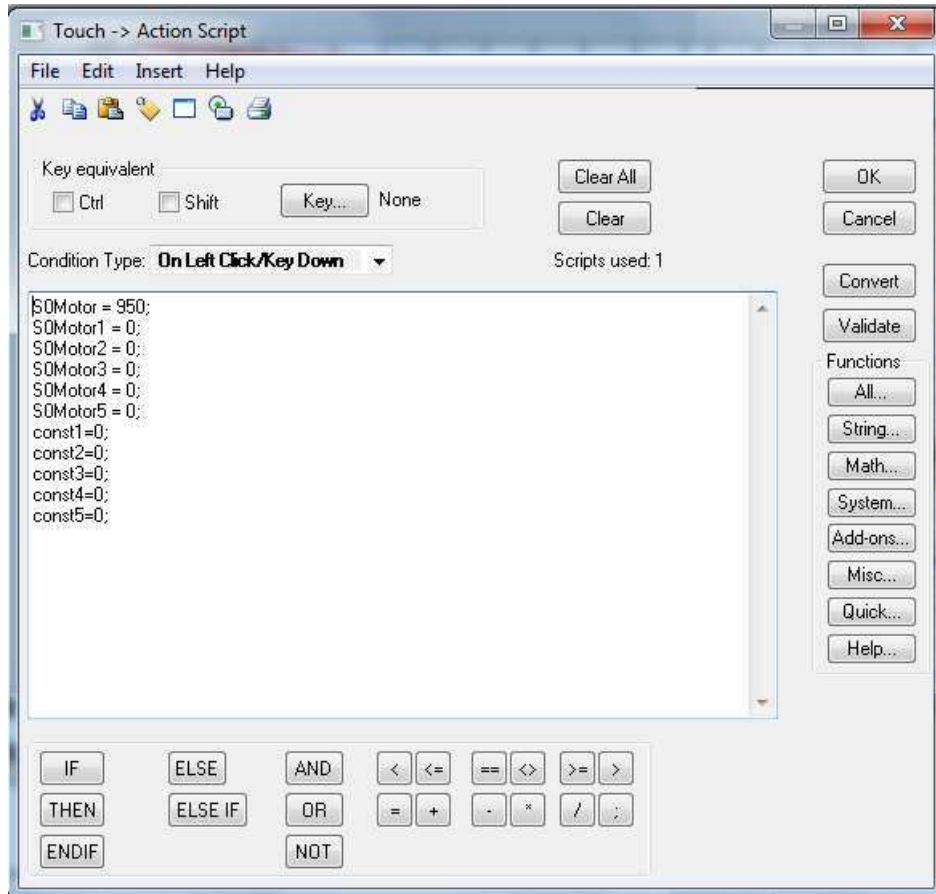


Figura 27: Action Script do botão Iniciar - Calibração Sentido Horário.

Calibração Sentido Anti-horário

A tela “Calibração Sentido Anti-horário” é composta por dois gráficos de tendência real, S0Motor e Velocidade (cm/s), dois botões, “Iniciar calibração” e “Voltar”, e cinco displays para exibir os parâmetros de calibração obtidos no experimento, como mostra a Figura 28.

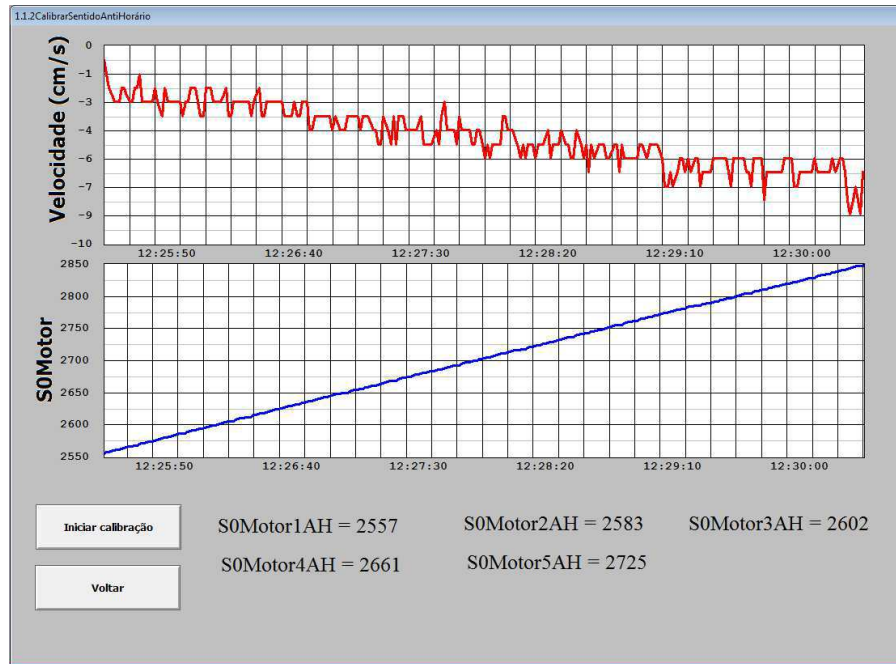


Figura 28: Tela Calibração Sentido Anti-horário.

O gráfico de tendência real “velocidade” exibe os valores da velocidade em cm/s. Esses valores estão associados, por meio do “window script” a valores de S0Motor e variam com o tempo.

Configurou-se o gráfico de tendência real para apresentar valores da tagname “velocidade” variando entre -10 e 0, pegando uma amostra a cada segundo, cobrindo 300 segundos, como mostra a Figura 29.

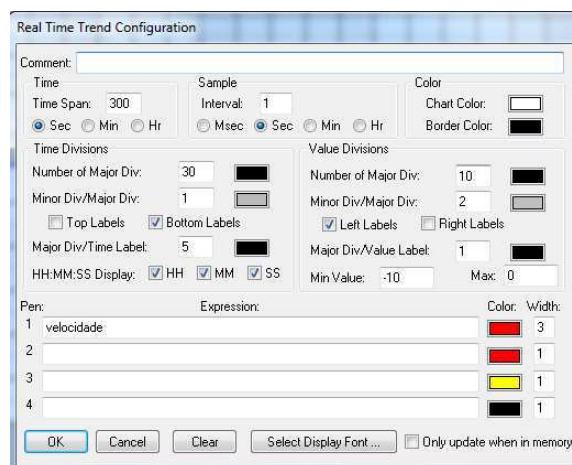


Figura 29: Configuração dos parâmetros do gráfico de tendência velocidade – Sentido Anti-horário.

Enquanto isso, o gráfico de tendência real “S0Motor” exibe os valores da tagname “S0Motor” sendo amostrado a cada segundo e com Span de 300 segundos, Figura 30.

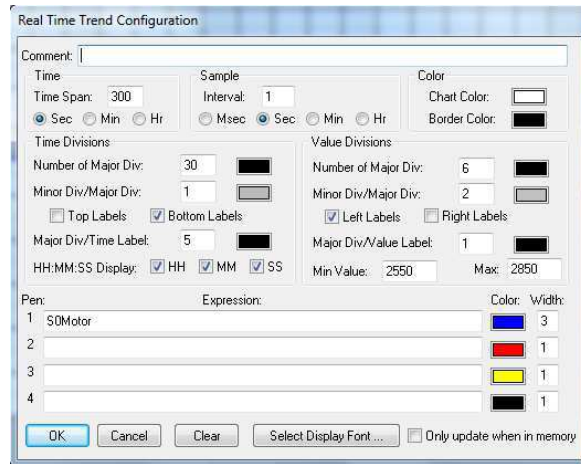


Figura 30: Configuração dos parâmetros do gráfico de tendência S0Motor – Sentido Anti-horário.

A “Condition Type” “On Show”, Figura 31, é executada ao abrir a janela e, para a janela “Calibrar Sentido Anti-horário”, é responsável pela inicialização das variáveis.

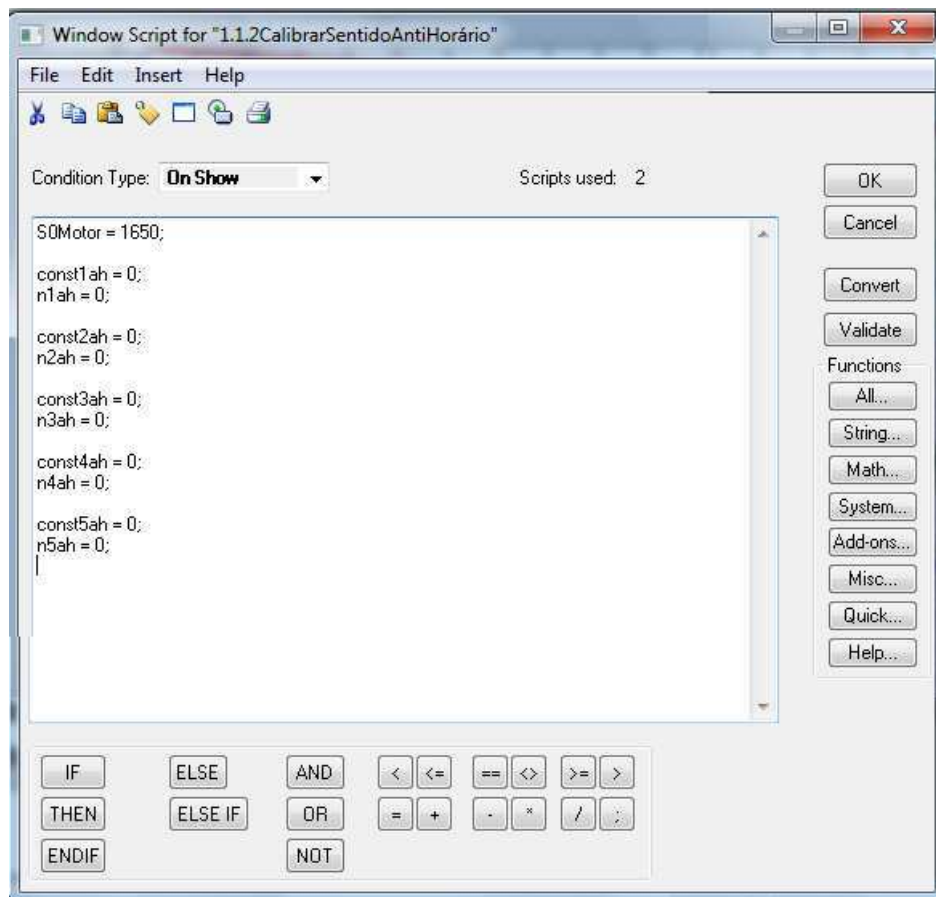


Figura 31: Window Script da janela Calibração Sentido Anti-horário – On Show.

A “Condition Type” “While Showing” é executada enquanto a janela “Calibrar Sentido Anti-horário” estiver sendo executada. Esse script é executado a cada segundo e é

responsável por fazer uma varredura em valores de S0Motor entre 2250 e 2850, como mostra a Figura 32.

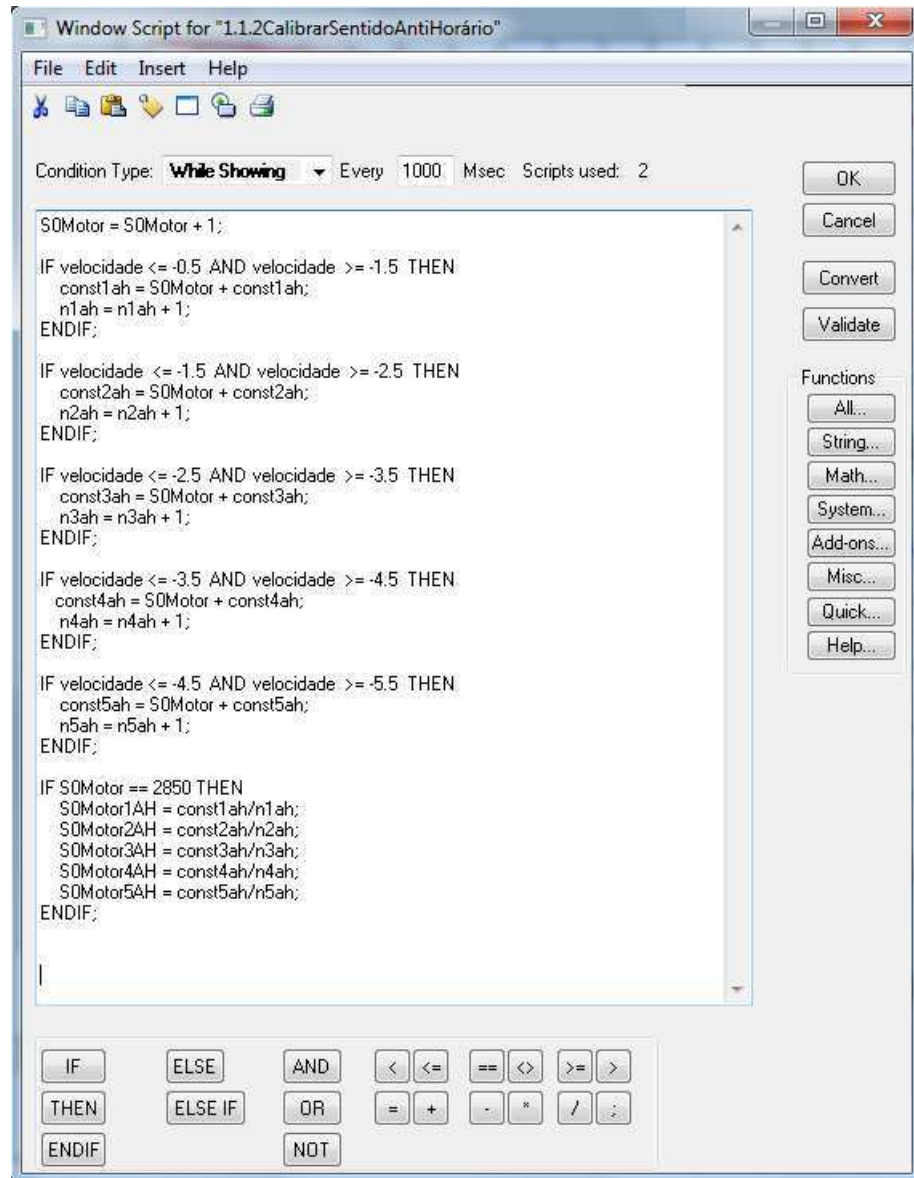


Figura 32: Window Script da janela Calibração Sentido Horário – While Showing.

Como mencionado no início deste tópico, a faixa de velocidade ideal está entre -1 cm/s e -5 cm/s. Então, discretizou-se essa faixa para os valores inteiros, sendo que, para cada valor inteiro, tinha tolerância de $\pm 0,5$ cm/s.

As constantes const1ah, const2ah, const3ah, const4ah e const5ah acumulam os valores de S0Motor para a faixa especificada, enquanto n1AH, n2AH, n3AH, n4AH e n5AH funcionam como um contador para o número de elementos registrados em suas respectivas faixas de valores.

Ao final da execução dessa rotina, ou seja, quando S0Motor fosse igual 1250, atribuiu-se às variáveis de calibração S0Motor1AH, S0Motor2AH, S0Motor3AH, S0Motor4AH e S0Motor5AH os valores das constantes acumuladoras divididas pelo número de elementos registrados nas faixas correspondentes.

O botão “Iniciar Calibração” está associado a uma subrotina que dá início, ou reinicia, à execução da calibração no sentido horário. Essa subrotina atribui os valores iniciais das variáveis, valor nulo às variáveis auxiliares e o valor inicial de 2250 à S0Motor. Como pode-se ver na Figura 33.

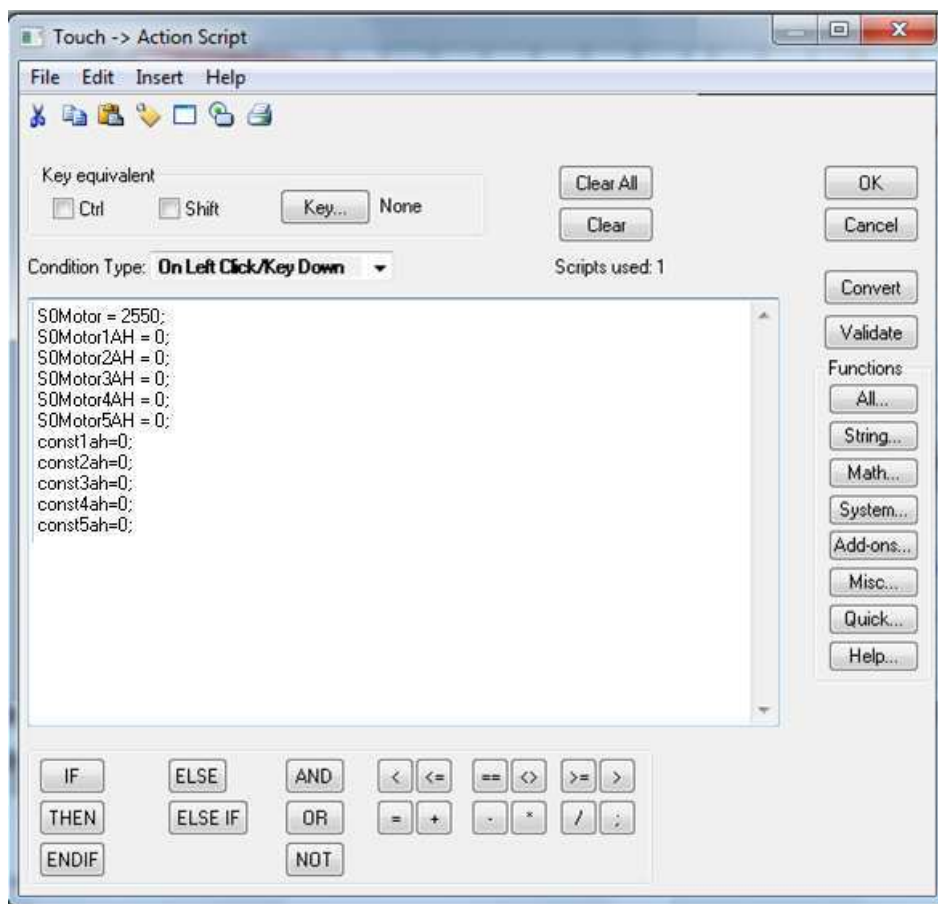


Figura 33: Action Script do botão Iniciar - Calibração Sentido Horário.

4.3.2 APLICAÇÃO PARA MONITORAMENTO E EXECUÇÃO DE ROTINAS

A aplicação “Monitoramento e Execução de Rotinas” é uma operação de rotina, ou seja, é mais frequentemente executada. Tanto faz o monitoramento das variáveis de interesse, temperatura, presença, velocidade do motor, quanto a execução de rotinas de ação, tudo de forma animada.

Para inicializar essa rotina, não é necessário que executar a aplicação de calibração. Visto que nela já estão configurados valores padrões.

Segue a lista de telas que compõem a aplicação:

- Tela de Login;
- Janela Inicial;
- Ajustar Valores Padrões de Velocidade;
- Sobre Limites de Velocidade;
- Monitorar Câmara de Aquecimento;
- Monitorar Câmara de Resfriamento;
- Executar Rotina de Ação 1;
- Executar Rotina de Ação 2.

Login

Inicialmente, criou-se um usuário para apenas quando “logado” o usuário poderá utilizar as opções disponibilizadas nessa aplicação. Inicialmente, como pode-se ver na Figura 34, o botão destinado a abrir a aplicação e todos os botões da janela inicial da aplicação, encontram-se inativos.

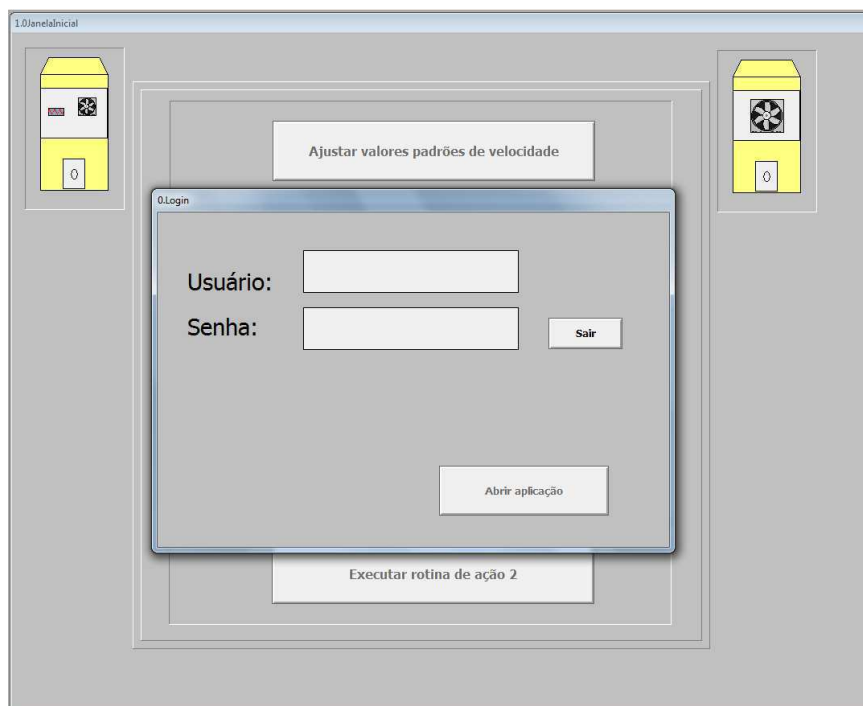


Figura 34: Tela Login – Inativo.

Criou-se uma tagname denominada `botao_login`, do tipo discreta, que recebe o valor 1, verdadeiro, quando o usuário (tagname `user`) for a string “aluno” e a senha (tagname `senha`) for a string “1234”. O botão “Abrir aplicação” é habilitado quando a tagname “`botao_login`” recebe o valor 1.

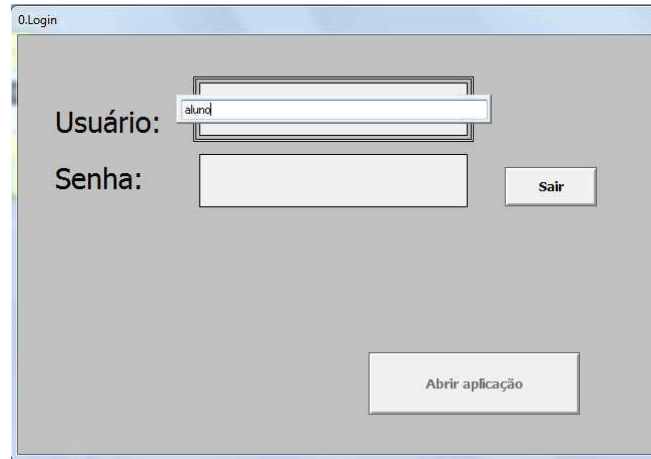


Figura 35: Tela Login – Digitando usuário e senha.

Após digitar a string “aluno” no campo usuário e clicar na tecla “Enter” e a string “1234” no campo senha e também clicar no botão “Enter”, o botão “Abrir aplicação” e os botões da janela inicial da aplicação encontram-se liberados, Figura 36.

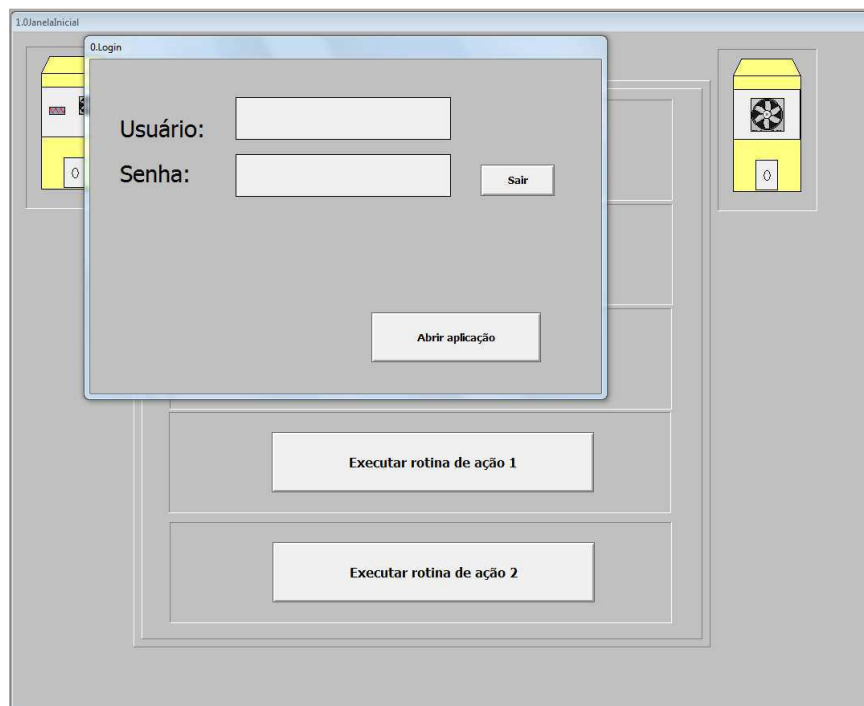


Figura 36: Tela Login – Ativo.

Tela Principal

A Tela Principal é composta por cinco botões cada um deles encaminha o usuário para a respectiva funcionalidade, Figura 37.

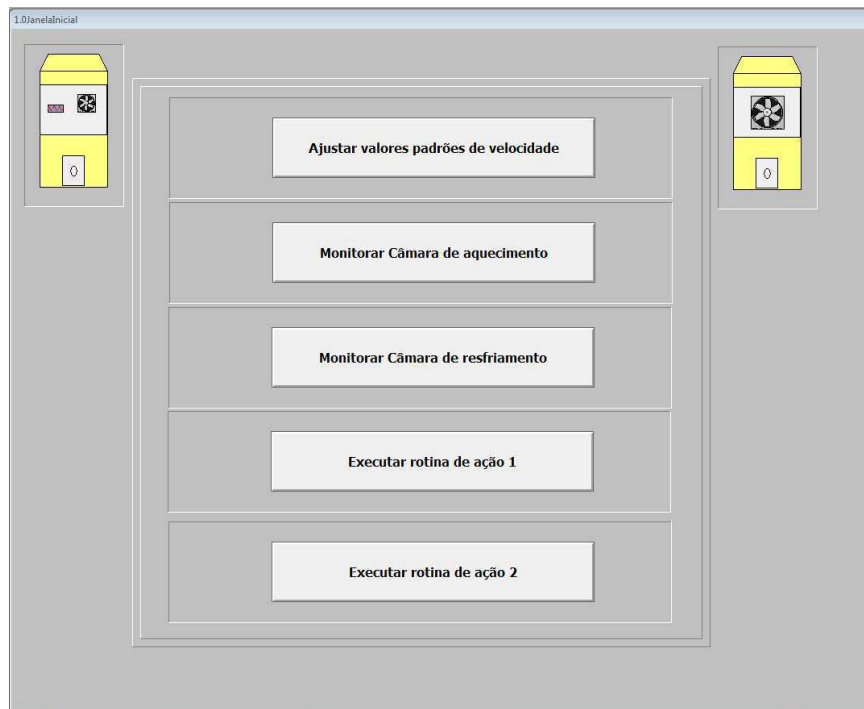


Figura 37: Janela Inicial.

Valores Padrões de Velocidade

A janela “Valores Padrões de Velocidade” é útil para atribuir valores de obtidos em uma nova calibração. Apesar da aplicação “Calibração de Motor” obter os valores para as velocidades 1 cm/s a 5 cm/s, não utilizou-se a velocidade 1 cm/s por ser demasiadamente lenta.

O botão “Valores padrão” atribui às variáveis valores de um experimento realizado anteriormente, o botão “Voltar” encaminha o usuário para a janela inicial, também há botões que encaminham o usuário para as outras funções da aplicação, evitando que ele precise retornar à tela inicial. Além disso, existe o botão “Sobre limites de operação das variáveis” que mostra uma janela que explica para o usuário qual a faixa de velocidade utilizada e o porquê, Figura 38.

1.1.AjustaSetpoints

Sobre limites de operação das variáveis.

Velocidade (cm/s)	Sentido	
	Horário	Anti-horário
2	<input type="text"/>	<input type="text"/>
3	<input type="text"/>	<input type="text"/>
4	<input type="text"/>	<input type="text"/>
5	<input type="text"/>	<input type="text"/>

Valores padrão

Rotina de ação 1 Rotina de ação 2

Voltar Câmara de resfriamento Câmara de aquecimento

Figura 38: Janela Valores Padrões de Velocidade.

Limites de Velocidade

A velocidade teve de ser reduzida a essa faixa pois, caso fosse mais veloz, o sensor de barreira não conseguia identificar o objeto, e ele acabava caindo. Por outro lado, quando era menor que 2 cm/s, a esteira movimentava-se muito lentamente.

1.1.1LimitesdeVariáveis

Limites de velocidade

A velocidade de operação para a aplicação apresentada deve ser entre 2cm/s e 5 cm/s. Visto que, para valores maiores ou menores, não teria sentido dado o contexto. Portanto, apesar de ser possível desenvolver velocidades maiores, optou-se por limitá-lo.

OK

Figura 39: Janela Limite de Variáveis.

Câmara de Aquecimento

A tela “Câmara de Aquecimento” é composta por dois gráficos: Temperatura (°C) e Duty Cycle (%), ambos em função do tempo. O sensor de barreira da câmara de aquecimento foi associado a uma lâmpada por meio da tagname “SBO2”. O botão “Iniciar” atribui o valor setado para o PWM (tagname SPPWMResistencia) à variável de entrada e saída de tagname

PWMResistencia. O botão “Voltar” retorna à tela inicial e existem botões que encaminham o usuário para as outras funções da aplicação, evitando que ele precise retornar à tela inicial. Olhar Figura 40.

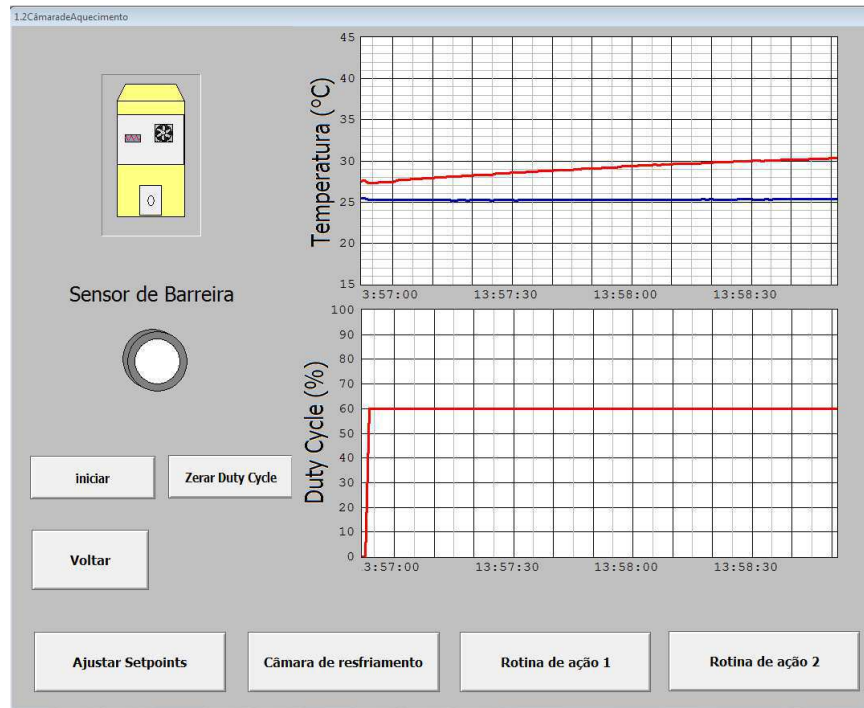


Figura 40: Câmara de Aquecimento.

O primeiro gráfico exibe a temperatura na câmara de aquecimento (tagname TempQ) e a temperatura da câmara de resfriamento (tagname TempF), que funciona como referência quando o cooler da câmara de resfriamento encontra-se desligado.

Com o objetivo de apresentar os valores de leitura de forma mais clara, limitou-se a faixa de temperatura entre 15 °C e 45 °C, valores de operação. O tempo atribuído foi 120 segundos, como mostra a Figura 41.

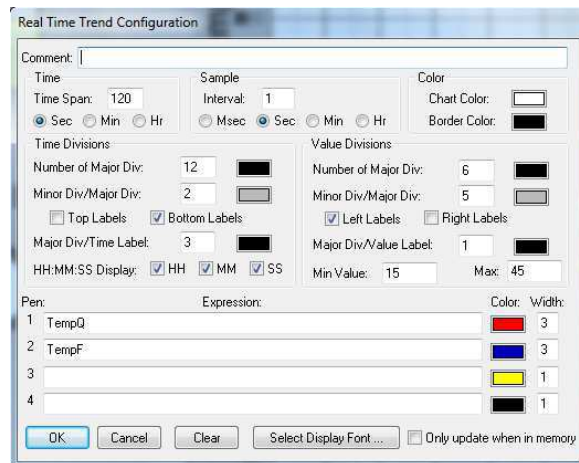


Figura 41: Configuração dos parâmetros do gráfico de Temperatura (°C).

O segundo gráfico mostra o duty cycle do PWM (Tagname PWMResistencia) relacionada à resistência localizada no interior da câmara de aquecimento em função do tempo. Os valores do duty cycle variam entre 0 e 100% e, como o objetivo era fazer uma comparação entre o ciclo de trabalho e a temperatura da câmara de aquecimento, o tempo de exibição desse gráfico também foi configurado para 120 segundos, como na Figura 42.

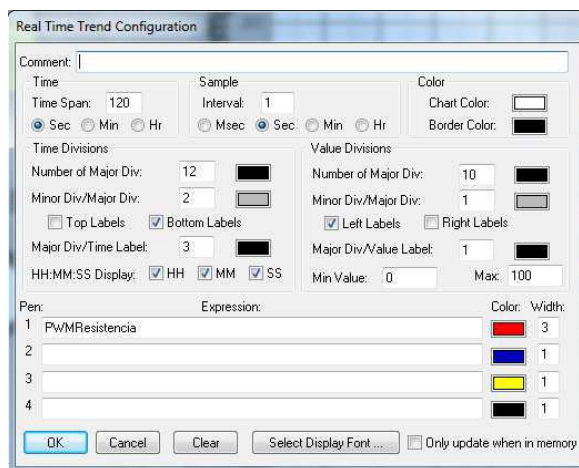


Figura 42: Configuração dos parâmetros do gráfico de Duty Cycle (%).

A figura mostra um experimento realizado para determinação da faixa de temperatura em função do Duty Cycle. No caso específico desta figura, o Duty cycle foi fixado em 60% e ao final na primeira etapa do experimento, após 120 segundos, obteve-se o valor de aproximadamente 30 °C.

O modelo do experimento foi baseado no funcionamento da aplicação, sabe-se que dificilmente a aplicação será executada por menos de dois minutos (120 segundos), portanto, este valor foi considerado o valor T_0 , temperatura inicial.

A segunda etapa do experimento consiste na observação da estabilização da temperatura, T, quando o sistema entra em regime. Dados os experimentos realizados, ocorreu em no máximo dez minutos.

Os experimentos foram realizados para os valores múltiplos de dez menores que 100. Os que apresentaram melhores resultados foram 40%, 60%, 80% e 100%. Esses valores foram escolhidos por apresentar acréscimo considerável a temperatura e pode-se observar um espaço entre eles. Por exemplo, o duty de 50% por vezes se confundia com os de 40% e 60%.

Após a realização desses experimentos e a delimitação do setpoint do duty cycle (temperatura), desenvolveu-se um script para atribuir o valores do setpoint (tagname SPPWMResistencia) à variável I/O correspondente ao PWM (tagname PWMResistencia), como mostra a Figura 43.

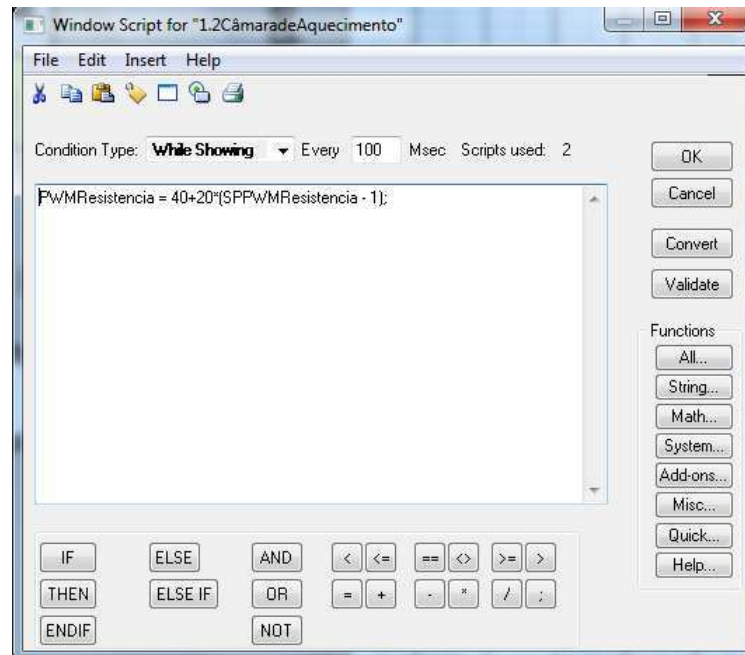


Figura 43: Window Script Câmara de Aquecimento – While Showing.

Câmara de Resfriamento

A tela “Câmara de Resfriamento”, Figura 44, é composta apenas por um gráficos: Temperatura (°C) em função do tempo. O sensor de barreira da câmara de aquecimento foi associado a uma lâmpada por meio da tagname “SBO3”. O botão “Voltar” retorna à tela inicial e existem botões que encaminham o usuário para as outras funções da aplicação, evitando que ele precise retornar à tela inicial.

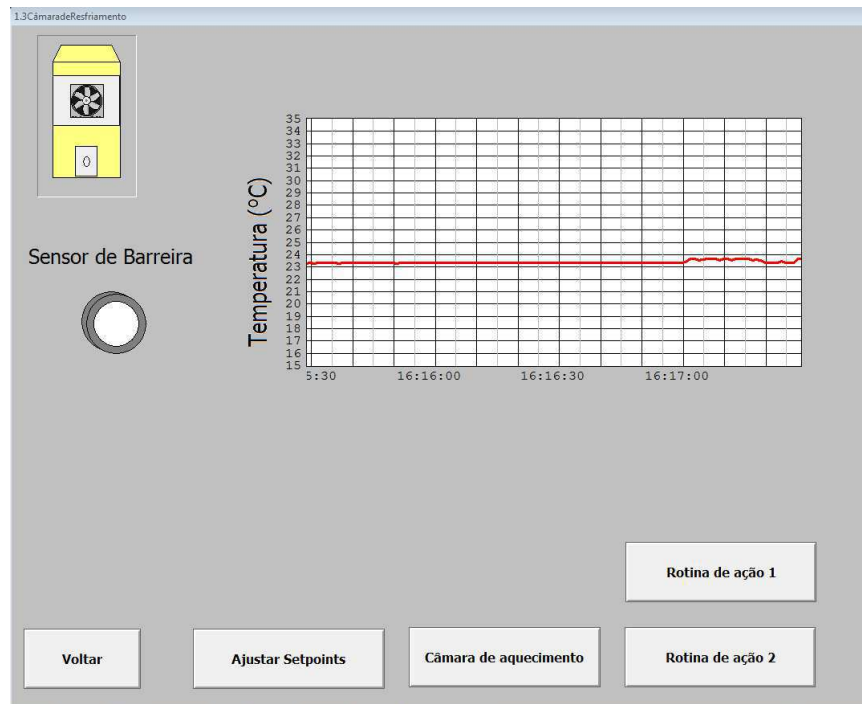


Figura 44: Câmara de Resfriamento.

De forma análoga ao gráfico de temperatura da câmara de aquecimento, limitou-se a faixa de temperatura (Tagname TempF) entre 15 °C e 45 °C, valores de operação e o tempo de exibição, 120 segundos, Figura 45.

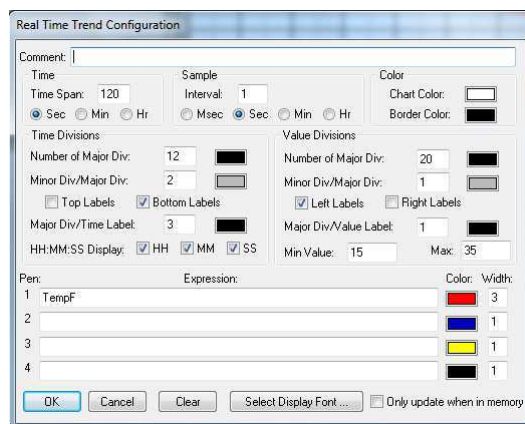


Figura 45: Configuração dos parâmetros do gráfico de temperatura de resfriamento.

Rotina de Ação 1

A tela “Rotina de Ação 1” tem o objetivo de executar a rotina que, ao localizar um objeto no sensor de barreira 1, mova-o até chegar à câmara de aquecimento, onde encontra-se o sensor de barreira 2. Ao chegar neste ponto, inicia-se uma contagem regressiva em segundos atribuída pelo usuário. Quando o contador é zerado, o objeto volta a mover-se até

atingir a câmara de resfriamento e ser detectado pelo sensor de barreira 3. Neste ponto é iniciado o segundo contador, que pode ser igual ou diferente do primeiro. No momento que este segundo contador é zerado o objeto volta a mover-se no sentido horário até atingir o sensor de barreira 4 e retornar ao ponto de partida.

Para melhorar a documentação da interface, fez-se o diagrama de estados da rotina de ação 1, Figura 46.

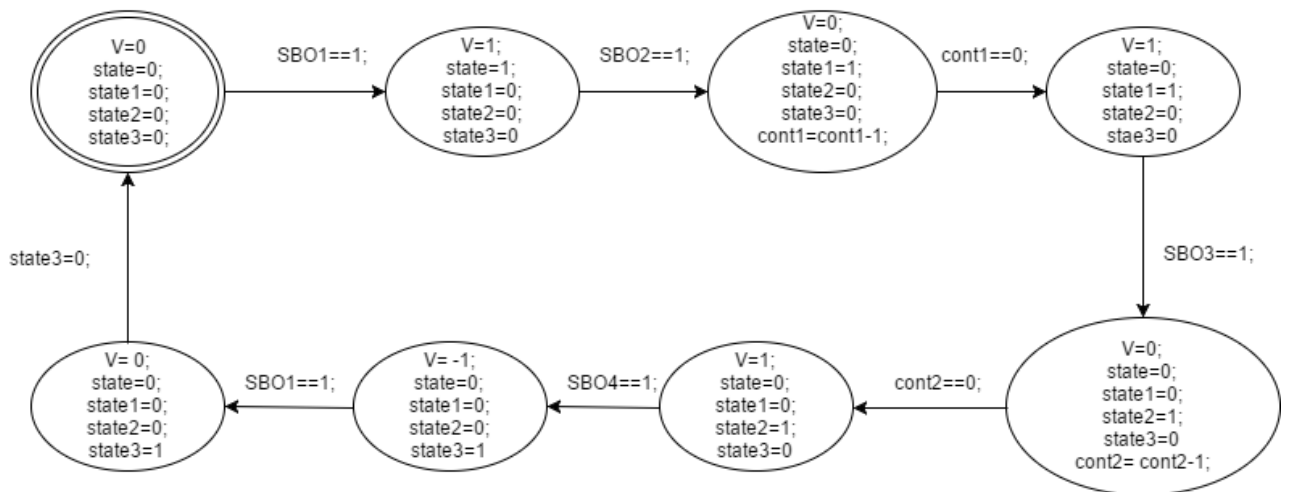


Figura 46: Diagrama de estados da rotina de ação 1.

Antes de iniciar a execução da rotina 1 é necessário selecionar os setpoints de duty cycle (temperatura) e velocidade e inicializar os contadores cont1 e cont2, das câmaras de aquecimento e resfriamento, respectivamente. Também foi proposto que o usuário tenha acesso às variáveis I/O do sistema durante a execução da rotina. Logo, também funciona como monitoramento. Ver Figura 47.

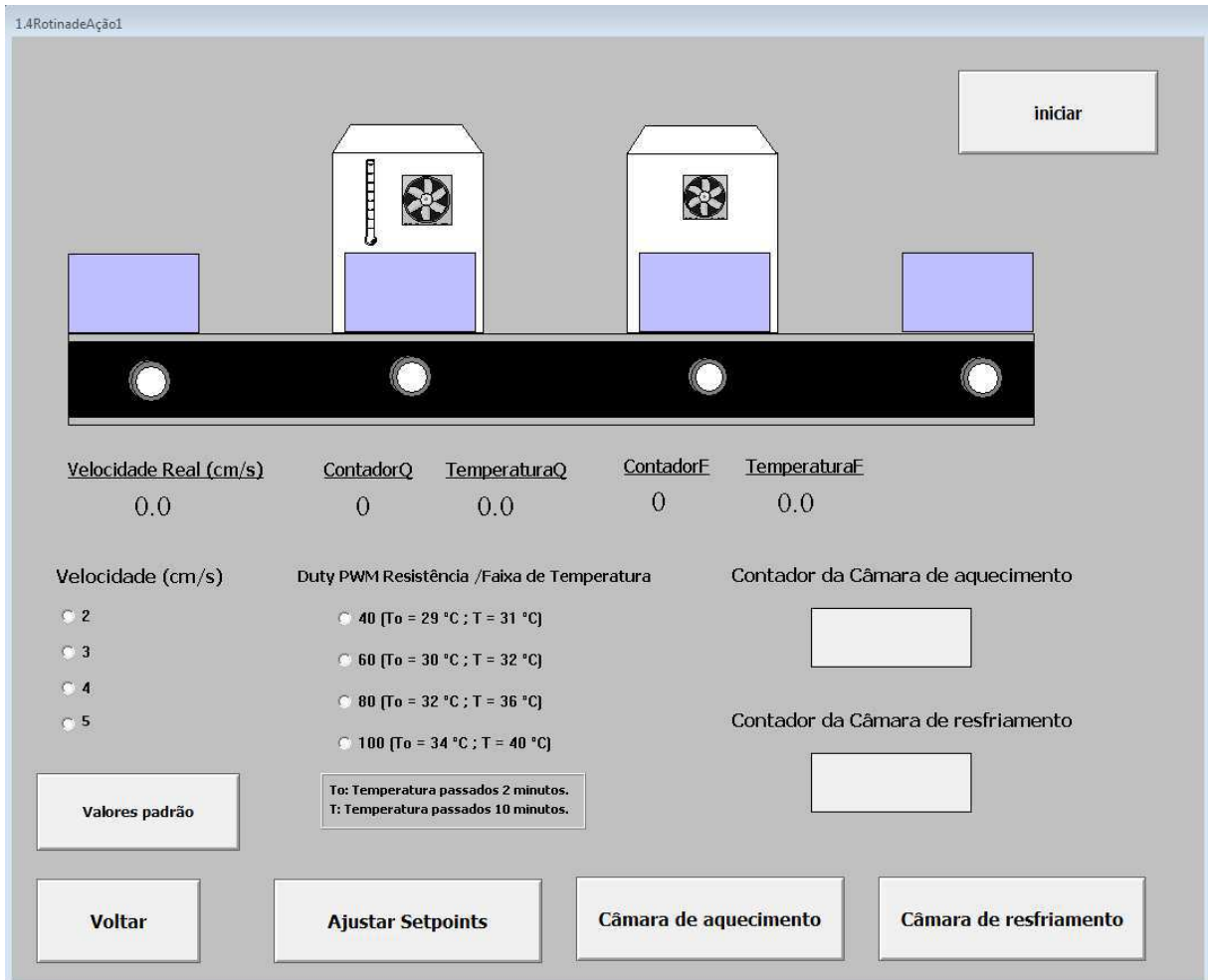


Figura 47: Rotina de Ação 1 – Window Maker

Foi necessário encontrar uma relação entre a dimensão da esteira real e os pixels da imagem. A relação escolhida foi de 1:9, sendo o tamanho real da esteira de 102 cm, equivalente a 918 pixels.

Desenvolver a ideia de animação requer uma construção de imagens. Para isso, foi necessário utilizar quatro blocos idênticos e ajustar o deslocamento, Figura 48, e quando devem ser exibidos e escondidos, Figura 49.

Properties	Value	Horizontal Movement
At Left End:	0	To Left: 0
At Right End:	312	To Right: 269

Figura 48: Deslocamento do bloco 1 relacionado à tagname deslocamento.

Figura 49: Visibilidade do bloco 1 relacionada à tagname state.

A função para determinar o “At Right/Left End” e “To Right/Left” são determinados da seguinte forma. O “At Right/Left End” equivale ao tamanho em pixels do objeto e o “To Right/Left” equivale à distância entre o ponto inicial e o ponto final.

Primeiramente foi realizado o cálculo para o bloco com maior deslocamento, nesse caso, o bloco 4. O tamanho total da esteira é de 918 pixels, como mencionado anteriormente. Logo, para calcular o “To Left” basta fazer a diferença entre o tamanho da esteira e o tamanho do bloco em pixels.

$$To\ Left = 918 - 125 = 793$$

Para calcular o “To Right/Left” faz-se o mesmo cálculo, posição final subtraído da posição inicial. Enquanto que para manter a mesma velocidade, deve-se seguir a mesma proporção para calcular o “At Right/Left End”.

$$At\ (Right/Left)\ End = \frac{918}{793} To(Right/Left)$$

Enumerando os blocos da esquerda para a direita e realizando os cálculos mencionados, tem-se a Tabela 2 que resume como estão associados os deslocamentos e a visibilidade de cada um deles.

Tabela 2: Parâmetros do deslocamento e visibilidade da rotina de ação1.

Bloco	At Right End	To Right	At Left End	To Left	Visibilidade
1	312	269	0	0	state
2	303	262	0	0	state1
3	271	234	0	0	state2
4	0	0	918	793	state3

O Window Script encontra-se em anexo, mas seguem figuras de que mostram momentos diferentes do processo.

A figura mostra o momento em que o objeto é detectado pelo sensor de barreira (SBO1), mas ainda não iniciou o deslocamento. Logo após, ele passa a mover-se até chegar à câmara de aquecimento.

O deslocamento é uma função que relaciona a velocidade real e a proporção entre a dimensão real da esteira e o tamanho em pixel e o contador. Visto que a rotina é executada a cada segundo, sempre que o objeto deve começar a mover-se, inicia-se o contador e quando o objeto está parado, o contador é zerado. Sempre que a velocidade for diferente de zero e o deslocamento for no sentido horário, é dado pela seguinte expressão:

$$\text{deslocamento [pixels]} = 9[\text{pixels/cm}] * \text{velocidade [cm/s]} * \text{contador[s]};$$

Além disso, como a velocidade no sentido anti-horário é negativa, sempre que o objeto estiver se movendo no sentido anti-horário a expressão será da seguinte forma:

$$\text{deslocamento [pixels]} = -9[\text{pixels/cm}] * \text{velocidade [cm/s]} * \text{contador[s]};$$

O bloco que é mostrado na figura é o bloco 1, cuja visibilidade depende da tagname state, ele passa a ser exibido no momento em que o SBO1==1 e state3=0 e, conseqüentemente, state=1, Figura 50.

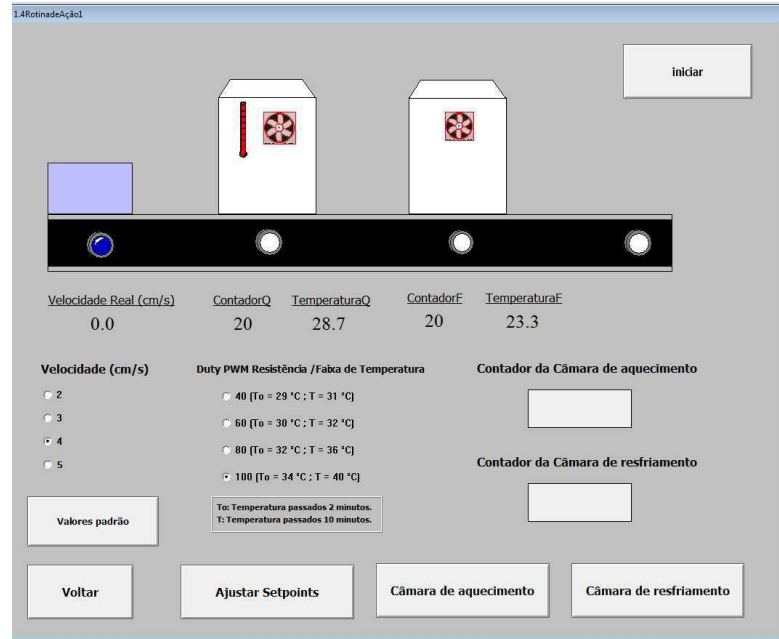


Figura 50: Rotina de Ação 1 - Estado inicial (state=1, SBO1=1).

Nesse segundo momento, ao chegar à câmara de aquecimento, o objeto é detectado pelo sensor de barreira (SBO2 = 1, state1=1 e state=0) no interior da câmara de aquecimento, o bloco 1, desaparece e o bloco 2 passa a ser exibido. Então, inicia-se a contagem regressiva do primeiro contador (cont1), Figura 51.

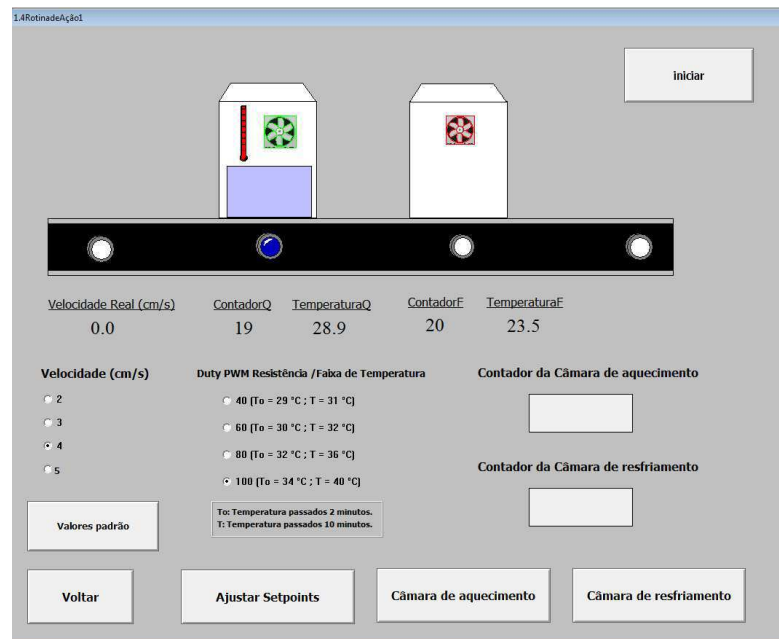


Figura 51: Rotina de ação 1 – Estado 2 (state=0, state1=1, SBO2=1).

Quando o SBO2 é ativado, o cooler liga e o PWM da resistência (PWMResistencia) recebe o valor que é função do duty cycle setado (PWMResistencia = 40 +

($SPPWMResistencia - 1$)*20). Por exemplo, no caso da figura o valor setado foi 4, quarta opção, logo, o duty do PWM estará em 100%.

Logo após o término do contador, o objeto retorna a mover-se no sentido horário, como é apresentado na figura. O bloco que aparece na figura 52 ainda é o bloco 2.

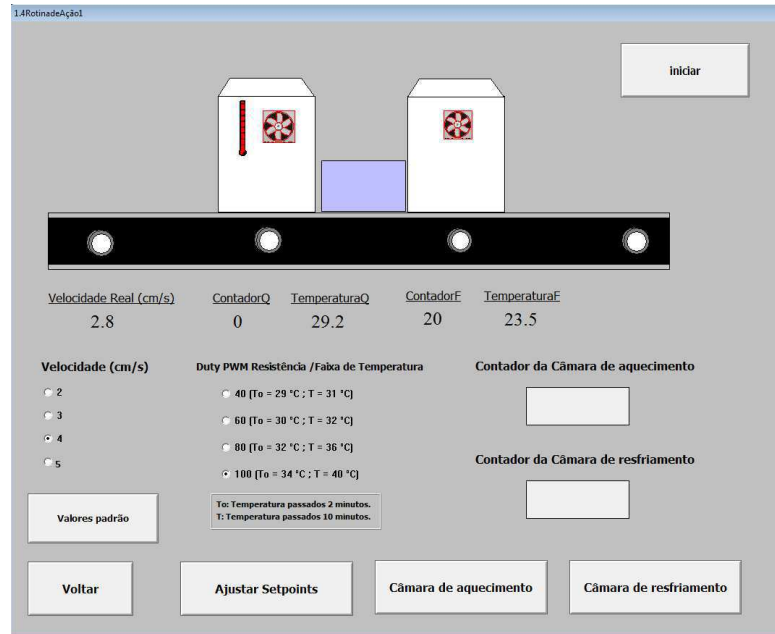


Figura 52: Rotina de ação 1 – Estado 3 (cont1=0, state1=1).

Ao ser detectado pelo sensor de barreira SBO3, o bloco 2 é escondido e o bloco 3 começa a ser exibido. Nesse momento, a esteira pára e o cooler e o contador são acionados, Figura 53.

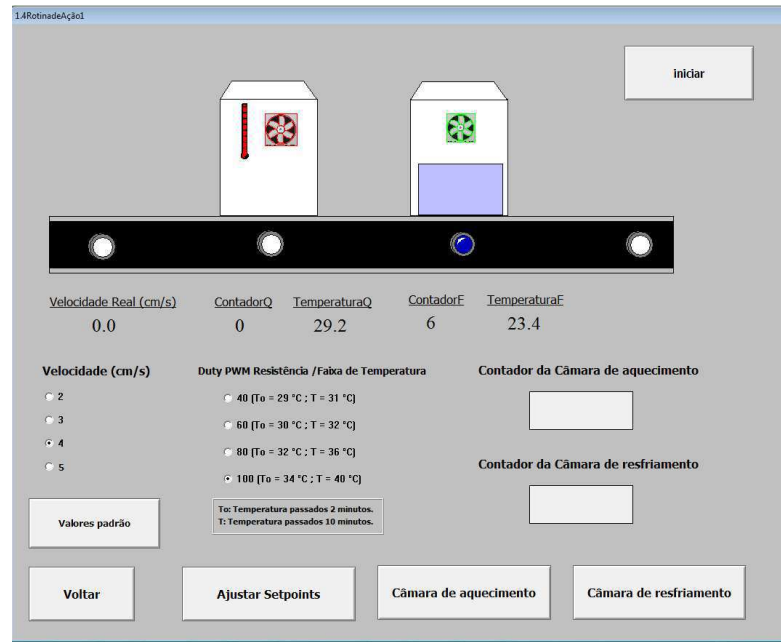


Figura 53: Rotina de Ação 1 – Estado 4 (state1=0, state2=1).

Após o contador ser zerado o bloco 3 passa a mover-se no sentido horário até atingir o sensor optico de barreira SBO4, quando a expressão que determina a visibilidade do bloco 3 é zerada, state2=0, e a expressão de visibilidade do bloco 4 é ativada, state3=1, Figura 54.

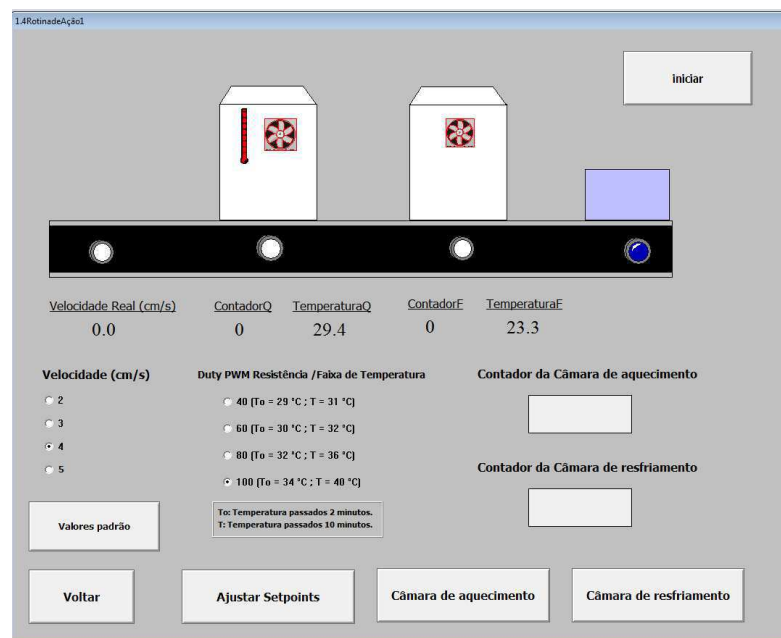


Figura 54: Rotina de Ação 1 – Estado 5 (state3=1, state2=0).

Quando state3=1 e SBO4=1, a esteira inverte o sentido e o bloco 4 passa a mover-se no sentido anti-horário, Figura 55.

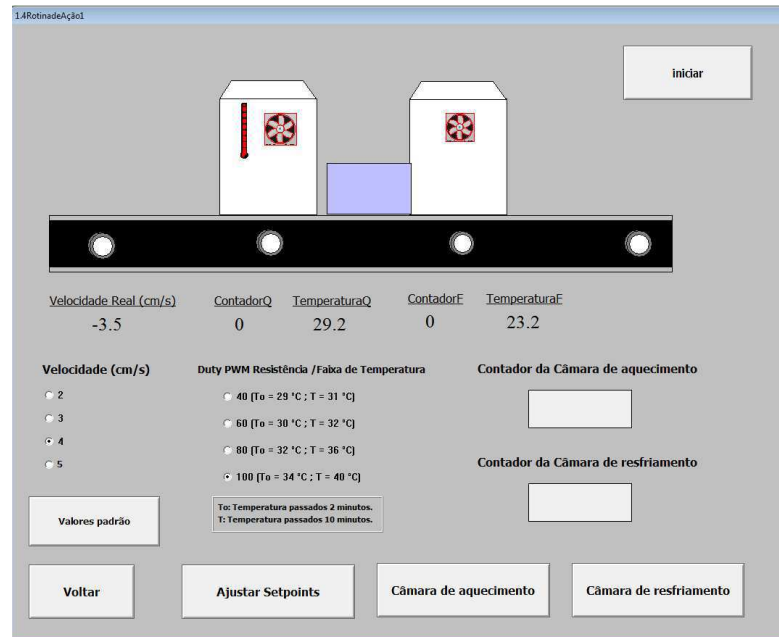


Figura 55: Rotina de Ação 1 – Estado 6 (state3=1)

Até atingir o sensor de barreira SBO1 e permanecer parado. Caso deseje-se reiniciar o processo, basta clicar no botão “Iniciar” que as variáveis serão zeradas, incluindo state3, e a rotina será executada novamente, Figura 56.

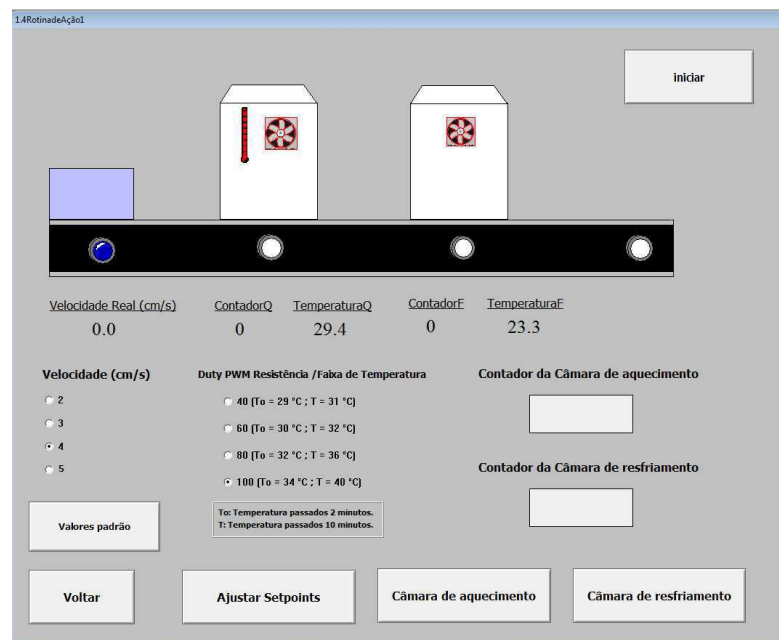


Figura 56: Rotina de Ação 1 – Estado 7 (state3=1)

Rotina de Ação 2

A tela “Rotina de Ação 2” tem o objetivo de executar a rotina que, ao localizar um objeto no sensor de barreira 4 (SBO4), mova-o até chegar à câmara de resfriamento, onde encontra-se o sensor de barreira 3 (SBO3). Ao chegar neste ponto, inicia-se uma contagem regressiva (cont2) em segundos atribuída pelo usuário. Quando o contador é zerado (cont2=0), o objeto volta a mover-se até atingir a câmara de aquecimento e ser detectado pelo sensor de barreira 2 (SBO2). Neste ponto é iniciado o segundo contador (cont1), que pode ser igual ou diferente do primeiro. No momento que este segundo contador (cont1=0) é zerado o objeto volta a mover-se no sentido anti-horário até atingir o sensor de barreira 1 (SBO1) e retornar ao ponto de partida.

Para melhorar a documentação da interface, fez-se o diagrama de estados da rotina de ação 2, Figura 57.

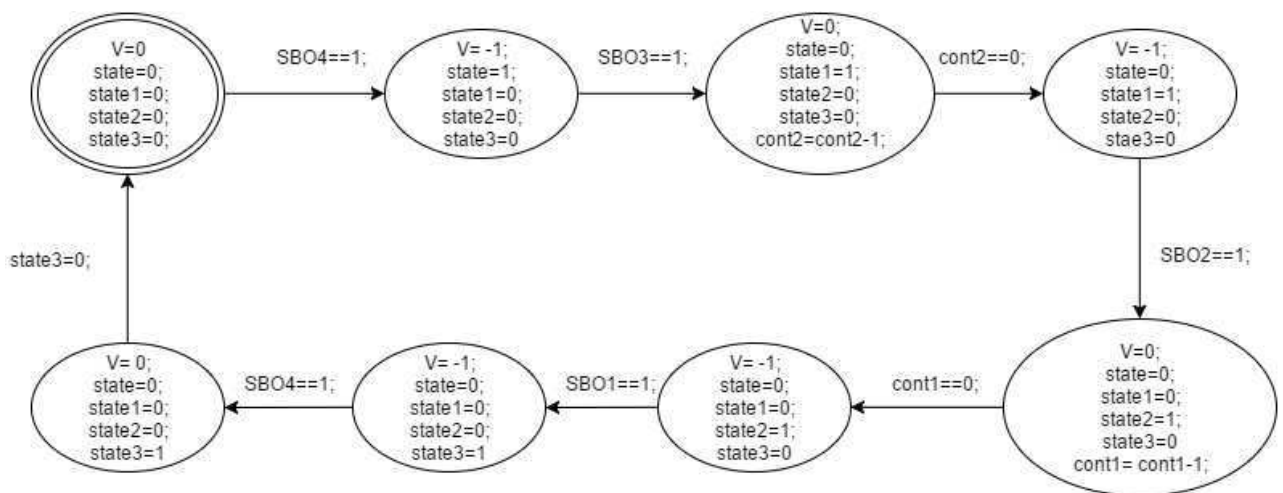


Figura 57: Diagrama de estados da rotina de ação 1.

Antes de iniciar a execução da rotina 2 é necessário selecionar os setpoints de duty cycle (temperatura) e velocidade e inicializar os contadores cont1 e cont2, das câmaras de aquecimento e resfriamento, respectivamente. Também foi proposto que o usuário tenha acesso às variáveis I/O do sistema durante a execução da rotina. Logo, também funciona como monitoramento. Olhar Figura 58.

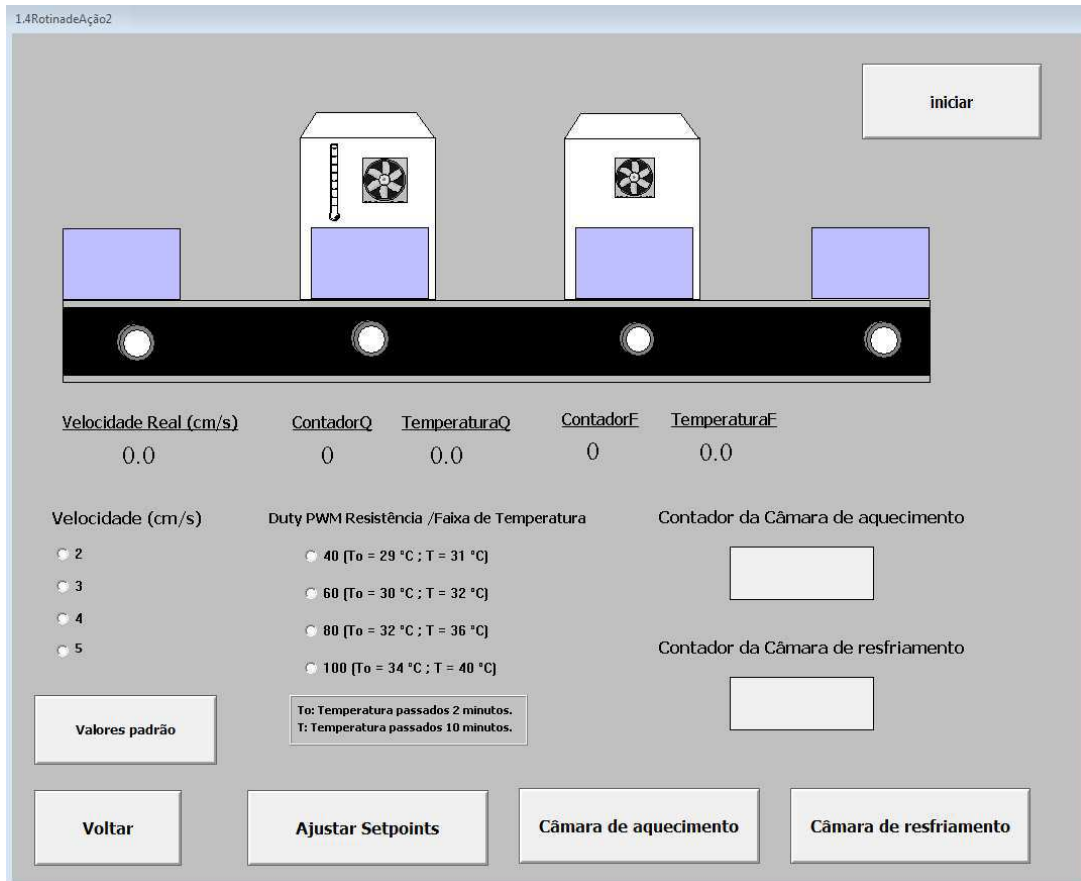


Figura 58: Rotina de Ação 2 – Window Maker.

Foi utilizada a mesma proporção da rotina de ação 1 entre a dimensão da esteira real e os pixels da imagem, 1:9, e a mesma ideia de animação com quatro blocos. A diferença se deu no sentido do movimento, inicializado em 1 (sentido=1), correspondente ao sentido anti-horário. Além disso, os blocos foram enumerados da direita para a esquerda.

Também houve uma pequena mudança na expressão do deslocamento, como a esteira e todos os elementos dela foram invertidos, o que antes era para direita ficou para esquerda e o que era esquerda, passou a ser direita, Figura 59.

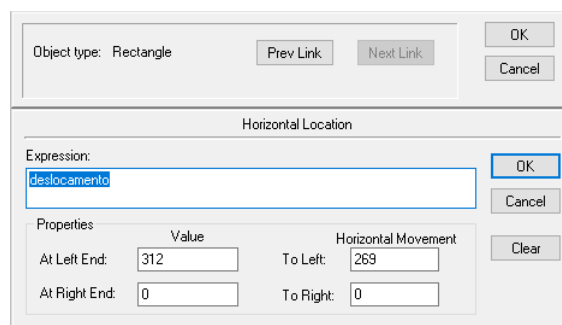


Figura 59: Deslocamento do bloco 1 relacionada à tagname deslocamento.

Enumerando os blocos da direita para a esquerda e realizando os cálculos mencionados na rotina de ação 1, tem-se a tabela 3 resume como estão associados os deslocamentos e a visibilidade da rotina de ação 2.

Tabela 3: Parâmetros do deslocamento e visibilidade da rotina de ação 2.

Bloco	At Right End	To Right	At Left End	To Left	Visibilidade
1	0	0	312	269	state
2	0	0	303	262	state1
3	0	0	271	234	state2
4	918	793	0	0	state3

O Window Script encontra-se em anexo, mas seguem figuras que mostram momentos diferentes do processo.

A figura 60 mostra o momento em que o objeto é detectado pelo sensor de barreira (SBO4), mas ainda não iniciou o deslocamento. O bloco 1, cuja visibilidade depende da tagname state, passa a ser exibido no momento em que o SBO4=1 e state3=0 e, conseqüentemente, state=1.

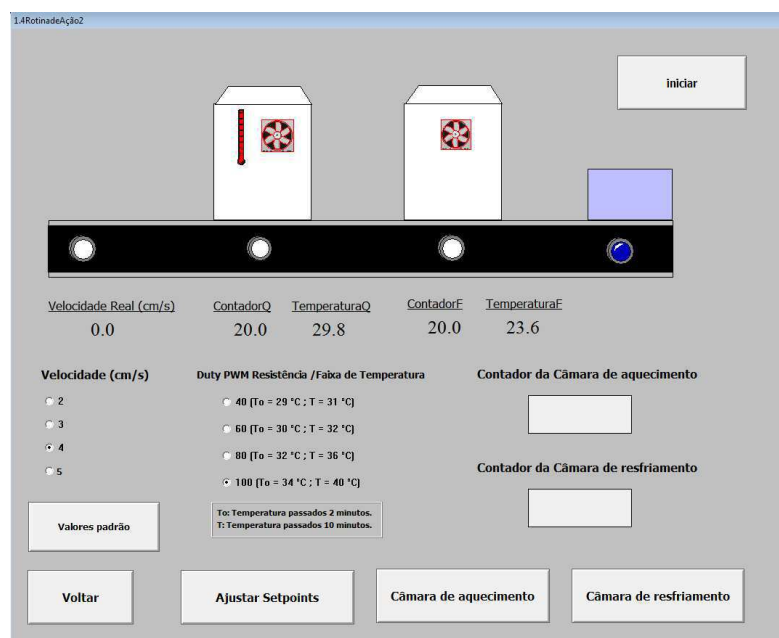


Figura 60: Rotina de Ação 2 - Estado inicial (state=1, SBO4=1).

Logo após, ele passa a mover-se no sentido anti-horário até chegar à câmara de resfriamento e ser detectado pelo sensor de barreira SBO3, Figura 61.

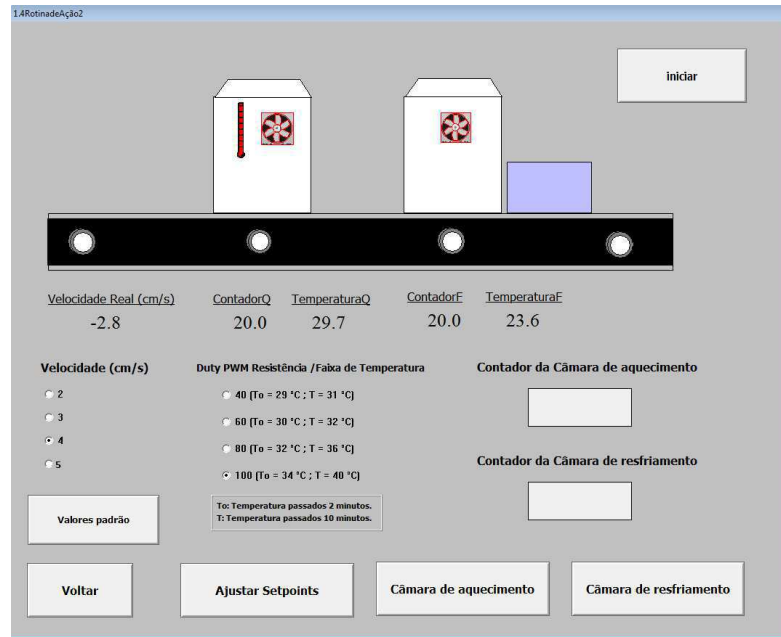


Figura 61: Rotina de Ação 2 - Estado 1 (state=1, SBO4=1).

No momento em que o bloco 1 chega à câmara de resfriamento, é detectado pelo sensor de barreira (SBO3 = 1, state1=1 e state=0). Então, o bloco 1 desaparece e o bloco 2 passa a ser exibido. Então, inicia-se a contagem regressiva do contador da câmara de resfriamento (cont2) e o cooler liga, figura. Quando o contador zera, o bloco volta a mover-se, Figura 62.

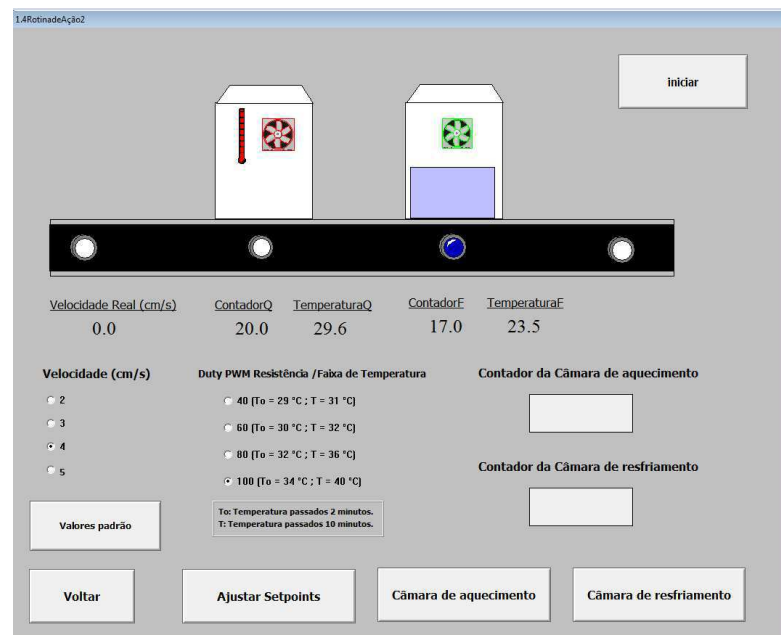


Figura 62: Rotina de Ação 2 - Estado 2 (state=0, state1=1, SBO3=1).

Quando o bloco 2 chega à câmara de aquecimento, é detectado pelo sensor de barreira (SBO2 = 1, state1=0 e state2=1). Então, o bloco 2 desaparece e o bloco 3 passa a ser exibido. Então, inicia-se a contagem regressiva do contador da câmara da aquecimento (cont1) e o PWM e o cooler ligam, figura 63.

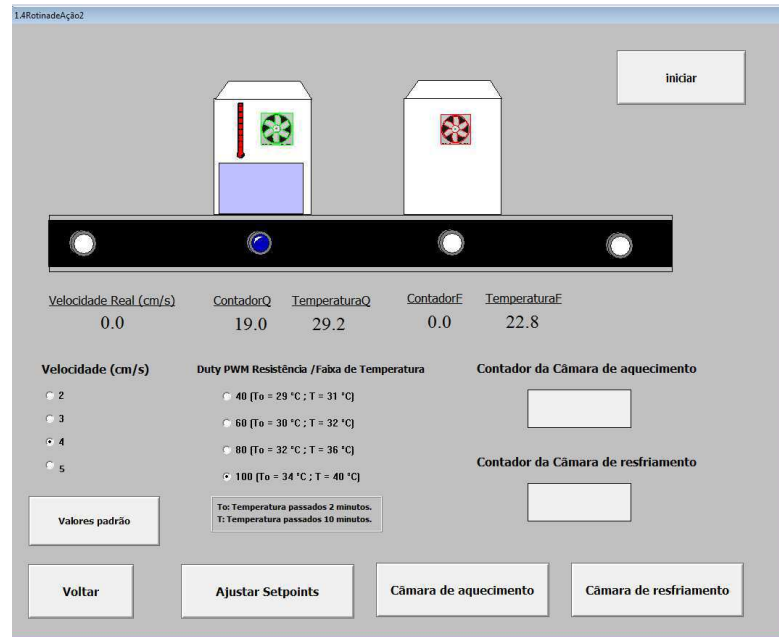


Figura 63: Rotina de ação 2 – Estado 4 (state1=0, state2=1, SBO2=1).

Quando cont1 zera, o bloco volta a mover-se no sentido anti-horário até atingir o sensor de barreira (SBO1). Nesse momento, o bloco 3 desaparece é mostrado o bloco 4, Figura 64.

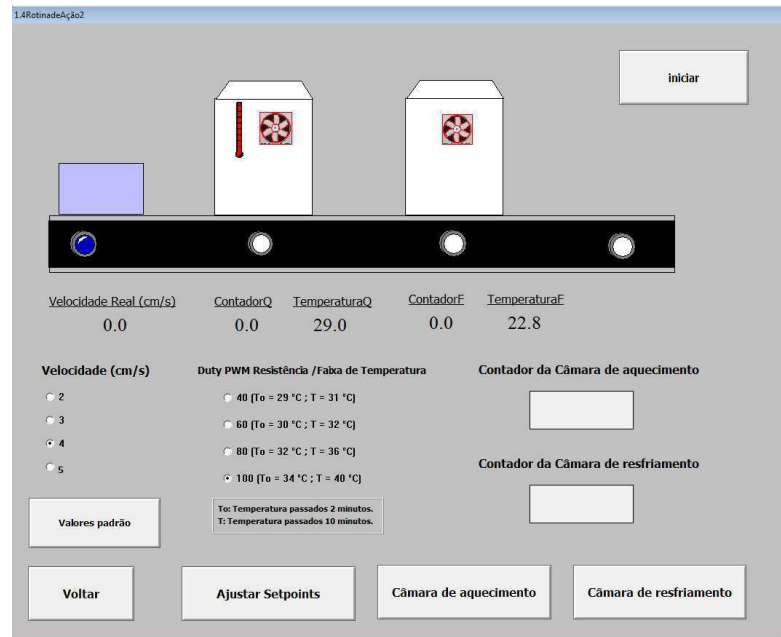


Figura 64: Rotina de ação 2 – Estado 5 (state2=0, state3=1, SBO1=1).

Então, quando state3=1 e SBO1=1, a esteira inverte o sentido e o bloco 4 passa a mover-se no sentido horário, Figura 65.

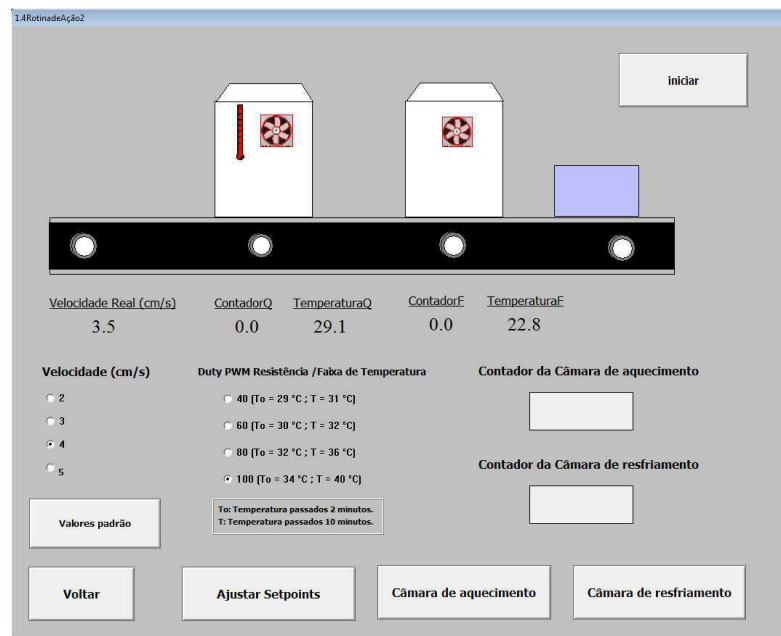


Figura 65: Rotina de ação 2 – Estado 6 (state3=1).

Até atingir o sensor de barreira SBO4 e permanecer parado. Caso deseje-se reiniciar o processo, basta clicar no botão “Iniciar” que as variáveis serão zeradas, incluindo state3, e a rotina será executada novamente, Figura 66.

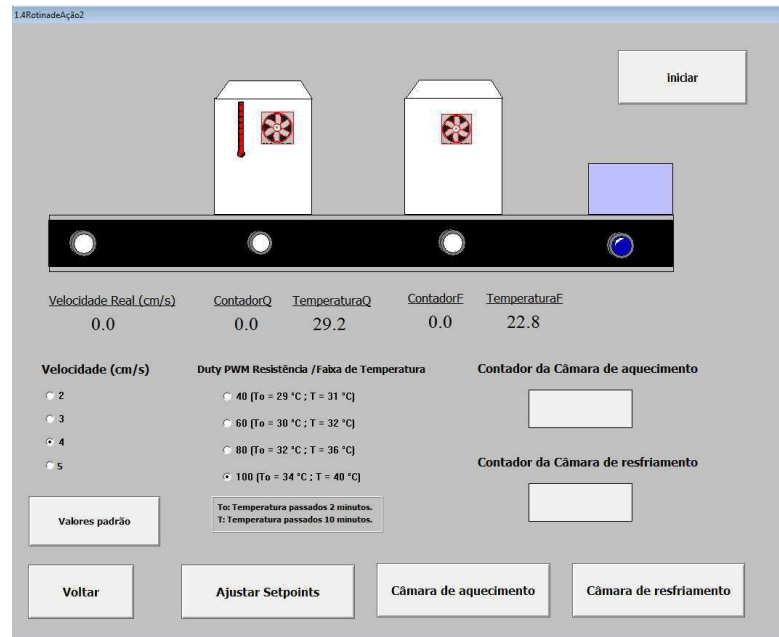


Figura 66: Rotina de ação 2 – Estado 7 (state3=1, SBO4=1).

4.4 DICIONÁRIO DE TAGS

4.4.1 TAGS I/O

Calibração de Velocidade de Motor

São utilizadas as variáveis D0 (Tagname S0Motor) e D6 (Tagname velocidade), como mostra a Tabela 4.

Tabela 4: Associação de Tags do CLP para aplicação do InTouch – Calibração de Motor.

Tags CLP		Tagnames
Endereço	Tag	
D0	S0MOTOR	S0Motor
D6	VELOCIDADE	velocidade

Monitoramento e Execução de Rotinas de Ação

No sistema de monitoramento e execução de rotinas de execução são utilizadas todas as tags I/O, como mostra na Tabela 5.

Tabela 5: Associação das tags do CLP com as tagnames do InTouch.

Tags CLP		Tagnames
Endereço	Tag	
D0	S0MOTOR	S0Motor
D6	VELOCIDADE	velocidade
D12	TQCELSIUS	TempQ
D15	TFCELSIUS	TempF
D16	SEN0	SBO1
D17	SEN1	SBO2
D18	SEN2	SBO3
D19	SEN3	SBO4
D22	DUTY	PWMResistencia

4.4.2 MEMORY TAGS

Calibração de Velocidade de Motor

Na Tabela 6 encontram-se todas as tagnames de memória da aplicação “Calibração de Velocidade do Motor”.

Tabela 6: Tagnames de memória da aplicação para calibração do motor.

Tagname	Tipo
n1	Inteira
n1ah	Inteira
n2	Inteira
n2ah	Inteira
n3	Inteira
n3ah	Inteira
n4	Inteira
n4ah	Inteira
n5	Inteira
n5ah	Inteira
S0Motor1	Real

S0Motor1AH	Real
S0Motor2	Real
S0Motor2AH	Real
S0Motor3	Real
S0Motor3AH	Real
S0Motor4	Real
S0Motor4AH	Real
S0Motor5	Real
S0Motor5AH	Real

Monitoramento e Execução de Rotinas de Ação

Na Tabela 7 encontram-se todas as tagnames de memória da aplicação “Monitoramento e Execução de Rotinas de Ação”.

Tabela 7: Tagnames de memória da aplicação para calibração do motor.

Tagname	Tipo
botao_login	Discreta
contador	Inteira
cont1	Inteira
cont2	Inteira
deslocamento	Inteira
PWMResistencia	Real
S0Motor1	Real
S0Motor1AH	Real
S0Motor2	Real
S0Motor2AH	Real
S0Motor3	Real
S0Motor3AH	Real
S0Motor4	Real
S0Motor4AH	Real
senha	String
sentido	Discreta
SPPWMResistencia	Inteira

SPvelocidade	Inteira
state	Discreta
state1	Discreta
state2	Discreta
state3	Discreta
user	String

5 COMUNICAÇÃO OPC

Os CLPs mais atuais trabalham com padrões de protocolo de comunicação para facilitar a interface com equipamentos de outros fabricantes, e também com Sistemas Supervisórios e Redes Internas de comunicação como o protocolo OPC (Object Linked and embedding for Process Control – o qual permite realizar transferências de objetos entre diferentes aplicações) cujo desenvolvimento foi encabeçado pela Microsoft, e tende a se tornar o padrão de relacionamento entre os diversos módulos de software ou hardware de diferentes fabricantes.

O elo de comunicação entre a IHM e o CLP (ou outro equipamento de controle monitorado) normalmente se dá por meio de um protocolo de comunicação específico que reproduz na IHM as variáveis do processo através de Tags – o protocolo OPC. Assim, uma Tag representa, em última análise, uma variável na IHM que pode ser do tipo discreto, numérico ou alfanumérico. Devido à bidirecionalidade do sentido de comunicação entre CLP e IHM, uma Tag pode tanto monitorar o status do controlador, como também enviar valores (ou setpoints – valores predefinidos) a ele. [8]

Para entender as aplicações desenvolvidas no InTouch com a finalidade de realizar leitura e escrita de variáveis, é necessário que se saiba primeiramente como ocorre a comunicação desde o equipamento até o software responsável pela criação da Interface Homem-Máquina (IHM), Figura 67.

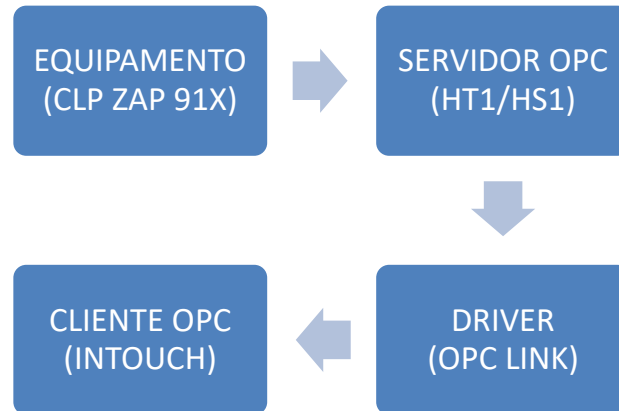


Figura 67- Janela inicial do SPDSW.

O protocolo OPC (OLE Process Control) consiste na especificação de um conjunto de padrões de interface que surgiu da colaboração entre empresas do mercado de Automação Industrial e a Microsoft.

O equipamento, neste caso o CLP ZAP 91X da HI Tecnologia, é um hardware com entradas e saídas, com memória programável capaz de executar diversas operações. Pode-se acoplar às suas variáveis de entrada e saída diversos dispositivos, tais como sensores e motores, para que sejam realizadas leituras e escritas.

O servidor OPC recebe os valores lidos das variáveis de entrada do equipamento e envia para o cliente. Além disso, recebe valores escritos pelo cliente e escreve nas variáveis de saída do equipamento.

O Driver, como o próprio nome sugere, é apenas um condutor que cria um caminho entre o servidor e o cliente OPC.

O cliente OPC é o responsável por receber valores das variáveis lidas pelas variáveis de entrada do equipamento e pode atribuir valores a variáveis de saída.

5.1 SOFTWARE SPDSW

O software utilizado para carregar os programas no CLP foi o SPDSW versão 3.3.03, Figura 68, disponibilizado gratuitamente pela HI Tecnologia.



Figura 68- Janela inicial do SPDSW.

Caso deseje-se realizar a instalação do software, basta seguir as instruções do material de referência disponibilizado pelo fabricante. No caso, não foi necessário instalar o software, pois a máquina utilizada já estava configurada, inclusive a comunicação e a conexão das entradas e saídas.

Com ele é possível carregar aplicações no CLP e monitorar os estados das variáveis.

Existem duas formas para conectar o CLP ZAP 91X da HI Tecnologia ao computador: serial ou ethernet. Ambas as formas serão explicadas.

Uma outra opção é carregar uma aplicação no CLP via ethernet. Primeiramente, deve-se identificar os dispositivos ativos na rede, para isso, como mostra a Figura 69, basta clicar em Ctrl+F11 ou seguir a sequência:

Ferramentas>>Comunicação>>Equipamento G3/GII-DUO



Figura 69- Janela do SPDSW para localizar controladores em rede.

Após isso, irá abrir a janela da Figura 70, contendo a lista de controladores na rede e deve-se verifica-se o IP do controlador que deseja-se conectar.

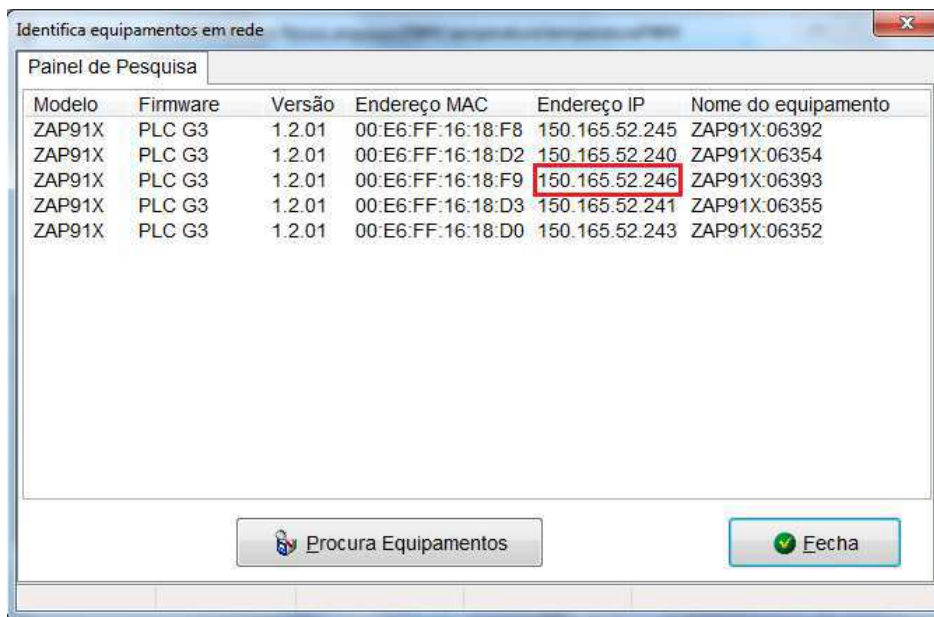


Figura 70- Janela do SPDSW para identificar equipamentos em rede.

O segundo passo é clicar com o botão direito em cima do equipamento escolhido e, em seguida, em Visualiza/Edita base de comunicação, Figura 71, para configurar os parâmetros de comunicação.

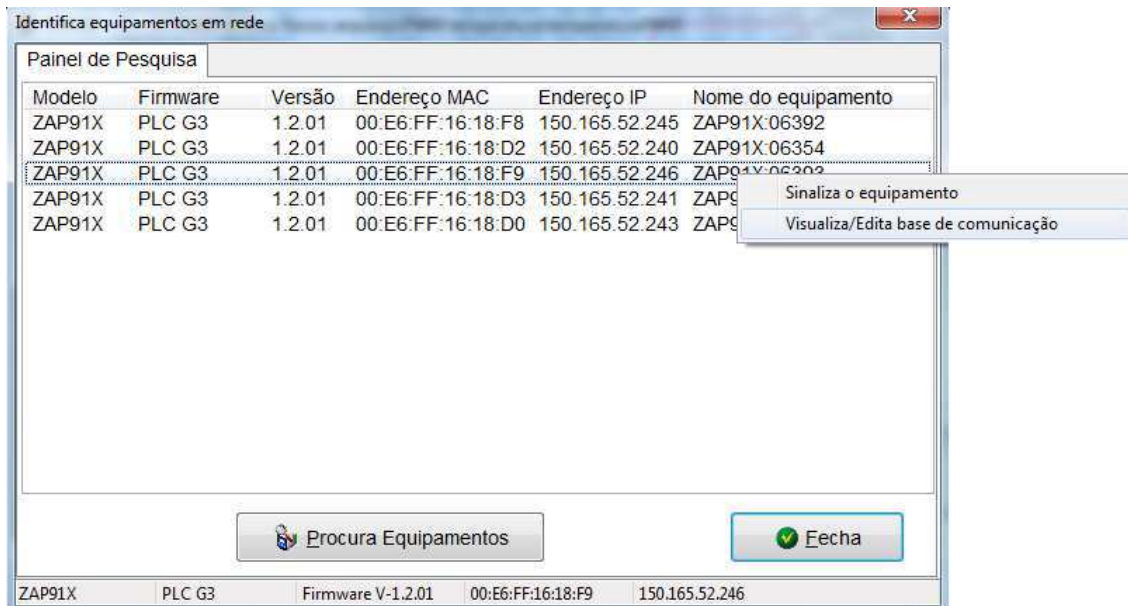


Figura 71-Janela que apresenta equipamentos em rede.

Na janela “Configuração de comunicação do equipamento”, deve-se ajustar os parâmetros de acordo com a Figura 72.

Deve-se prestar atenção principalmente ao IP do gateway, Endereço IP e a porta.

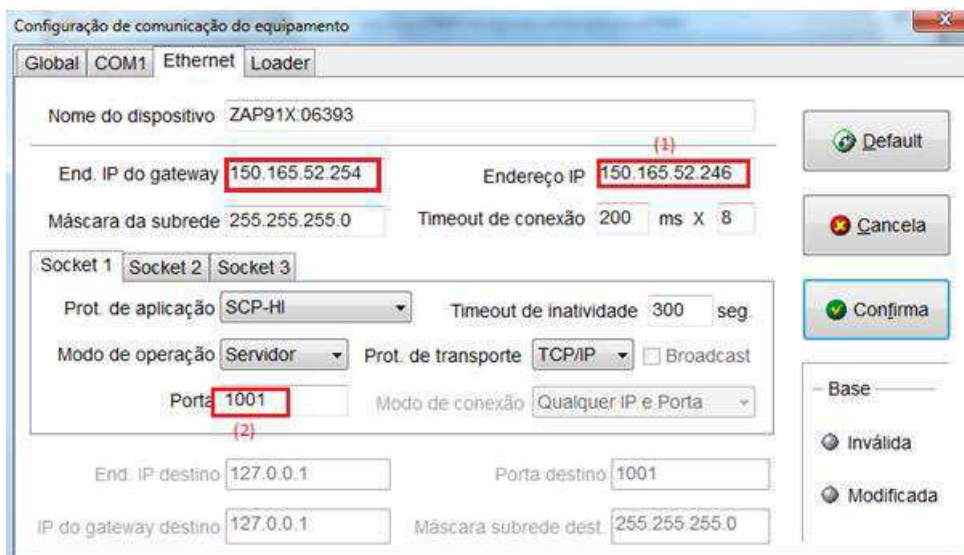


Figura 72- Aba Ethernet da janela Configuração de comunicação de equipamento.

Após configurado, é só clicar em “confirma” na janela “Configuração de comunicação de equipamento” e “fechar” a janela “Identifica equipamentos em rede”.

Em seguida, clique em sequência em:

Ferramentas>>Comunicação>>Configurar>>Computador–Setup de comunicação

ou clique em Ctrl + F8. Vai abrir a janela das figuras 73 e 74.

Na aba Drivers selecione a opção “Canal Ethernet TCP/IP ou UDP” e ajuste os parâmetros de acordo com a Figura 74.

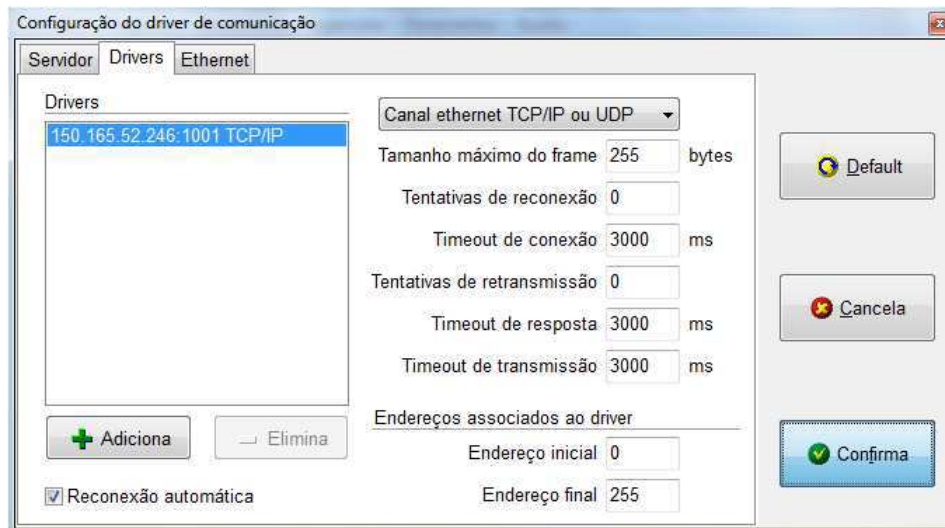


Figura 73- Janela para Configuração do Driver Comunicação - Drivers.

Na aba Ethernet ajuste o IP do equipamento, IP do Gateway e porta destinada à comunicação, Figura 74.

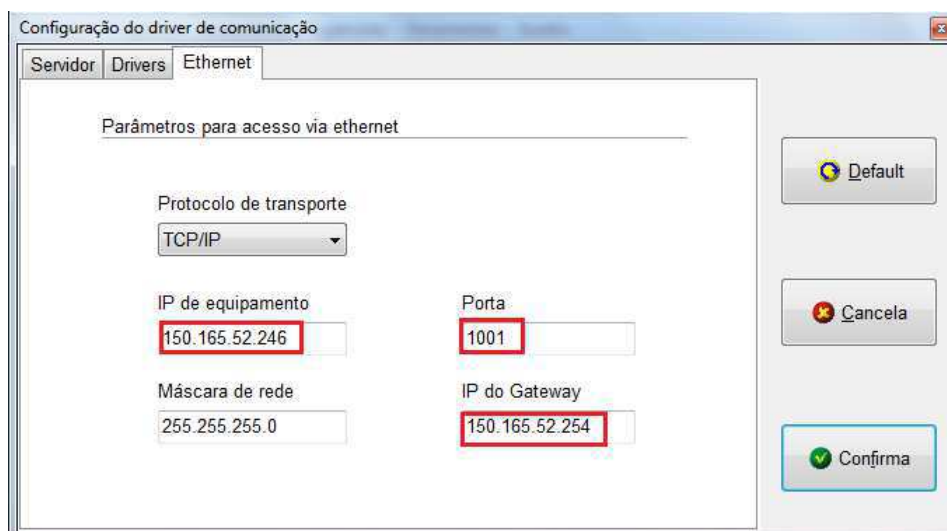


Figura 74- Janela para Configuração do Driver Comunicação – Ethernet.

Finalmente, de volta à aba Drivers, selecione o canal configurado, como na Figura 73,então, clique em “Confirma”, ainda na janela de configuração do driver de comunicação.

Por último, para concluir a comunicação, resta clicar em conectar na janela principal.

5.2 SERVIDOR OPC

Para criar um servidor OPC utilizou-se um software disponibilizado pelo fabricante, o HT1 Power Tool para receber e enviar dados via comunicação serial. Para esta versão, os recursos disponibilizados são para escrita e leitura de variáveis do tipo R, M ou D.

Os procedimentos necessários para instalação foram efetuados seguindo as recomendações da Nota de Aplicação ENA00033 da HI Tecnologia [9]. Os passos relatados a seguir referem-se apenas à utilização do software.

Passo 1: Carregar a aplicação desejada no CLP utilizando a programa SPDSW.

Passo 2: Fechar o programa SPDSW ou outro que esteja se comunicando com o CLP pela serial.

Passo 3: Abrir o programa HT1 Power Tool e clicar em *Connect*, Figura 75.

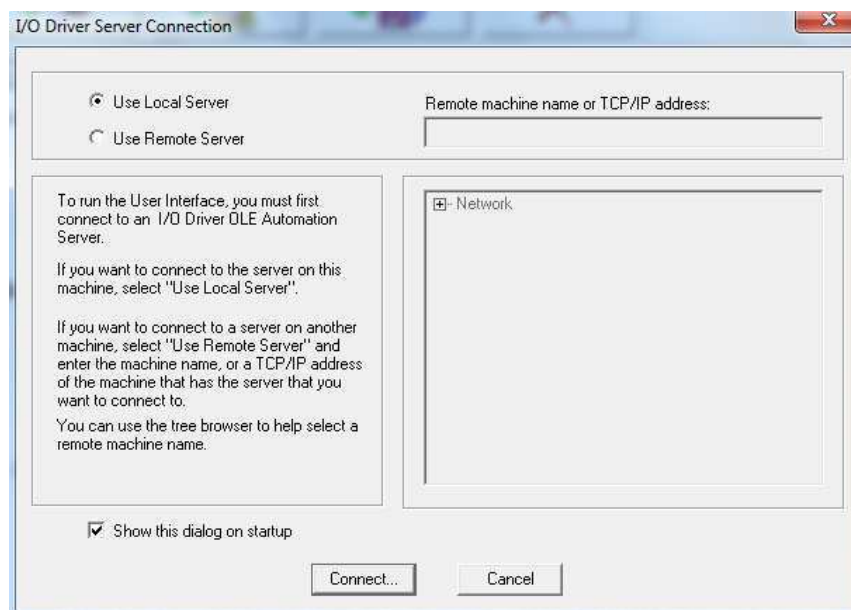


Figura 75 – Janela inicial do Programa HT1 Power Tool.

Passo 4: Identificar equipamentos na rede e criar novo canal, Figura 76.

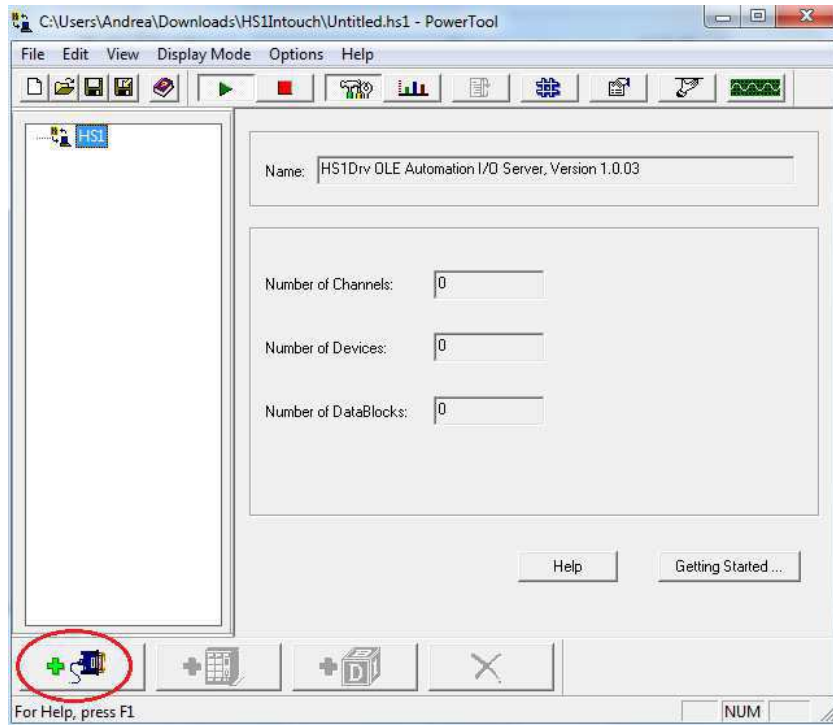


Figura 76 –Janela do HT1 para identificar os equipamentos na rede.

Passo 5: (1) Habilitar o canal e (2) criar novo Device, Figura 77.

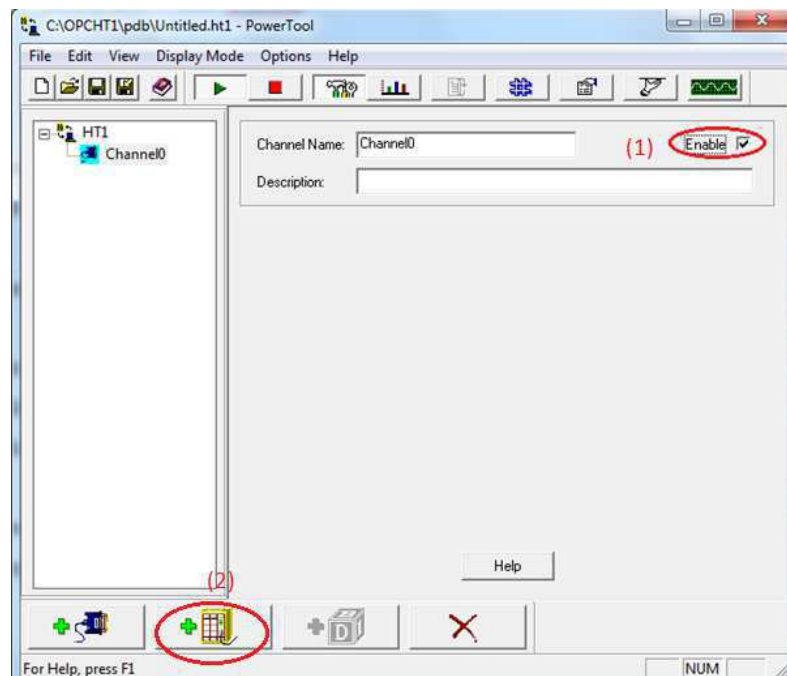


Figura 77 – Janela do programa HT1 para configuração do canal.

Passo 6: De acordo com a Figura 78, deve-se: (1) Habilitar o equipamento e ajustar parâmetros. (2) Lembre-se de colocar o Primary IP address exatamente igual ao IP do

equipamento. (3) Criar um ou mais *DataBlocks*, existente na aplicação carregada no CLP, e habilitá-los. Os *DataBlocks* são os endereços das variáveis de interesse.

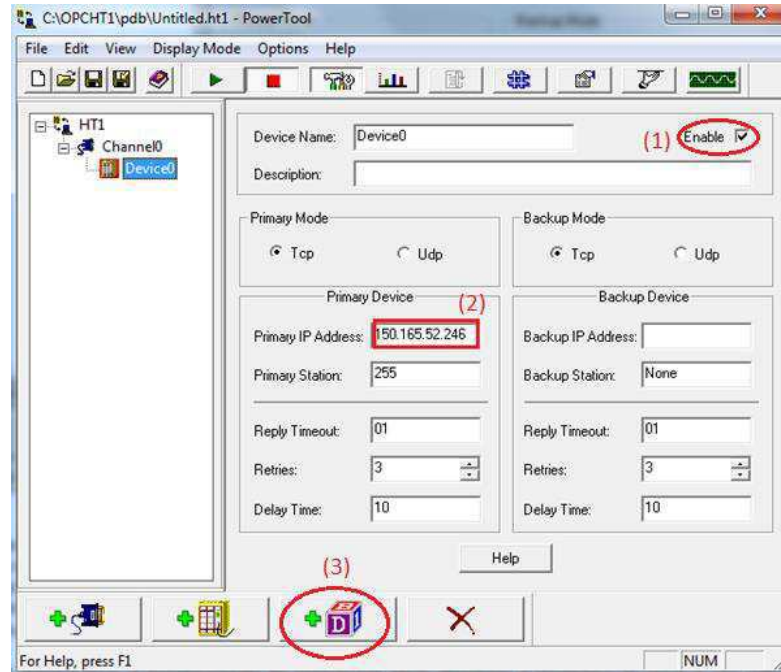


Figura 78 - Janela do programa HT1 para configuração dos parâmetros equipamento.

Passo 7: De acordo com a Figura 79: (1) Habilite o bloco de dados e ajuste seus parâmetros. (2) Lembre-se que o endereço do bloco de dados só poderá ser habilitado, caso esteja sendo utilizado na aplicação programada no CLP.

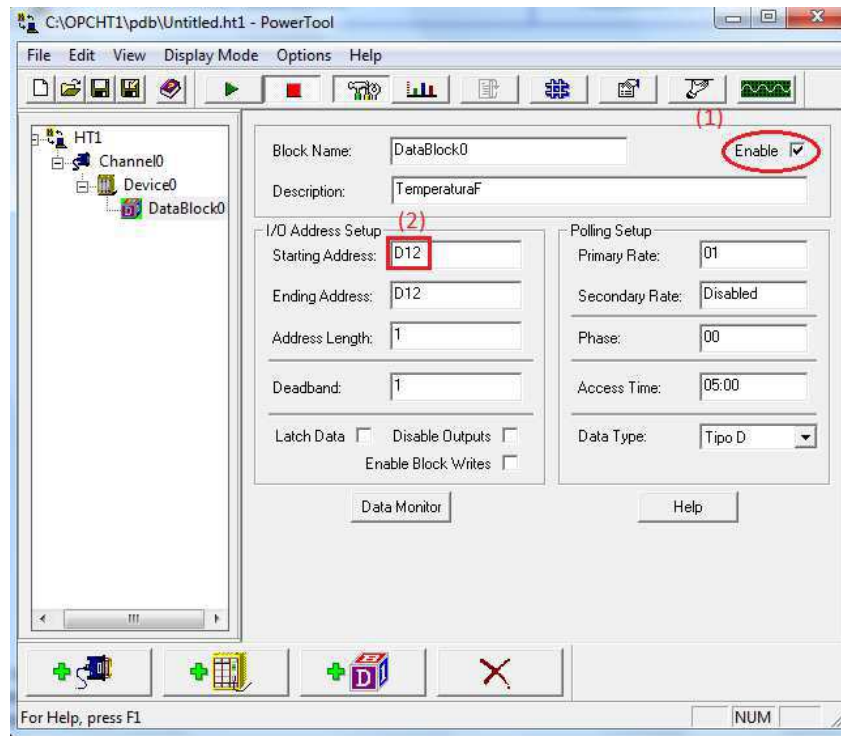


Figura 79 - Janela do programa HT1 para configuração dos parâmetros do bloco de dados.

Passo 8: Finalmente, basta clicar no botão *PLAY* e verificar se foi realizada a comunicação clicando no botão *Statistics*, Figura 80. Caso a qualidade dos dados esteja boa, a comunicação até esse ponto foi feita de forma correta. Caso contrário, refaça os pontos anteriores.

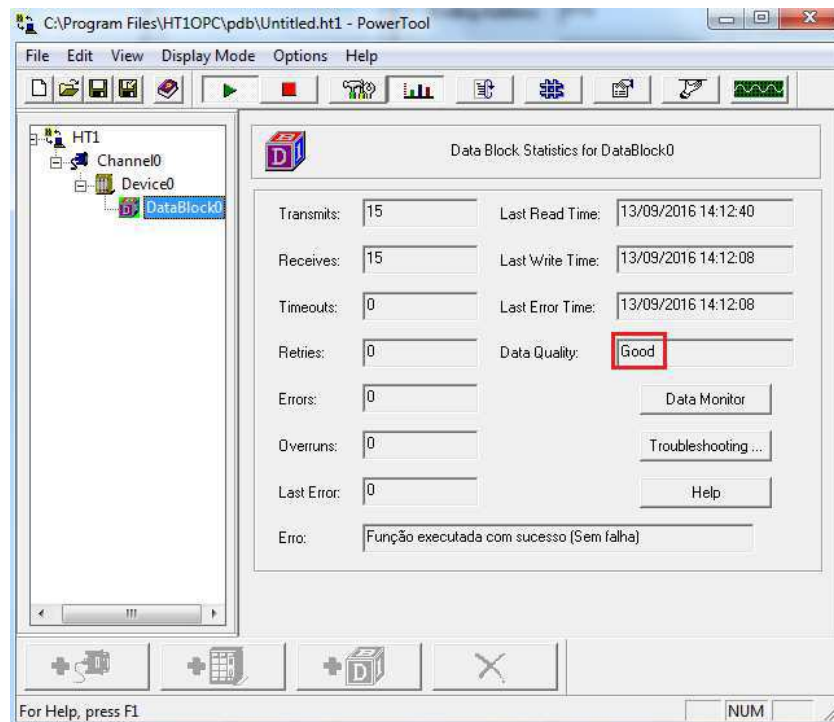


Figura 80 – Janela *Statistics* do programa HT1 – Visualização de qualidade de dados.

Passo 9: Salve e minimize a janela do Programa HT1 Power Tool. Abra o OPCLink para continuar o procedimento.

5.3 OPCLINK

Para servir como interface para comunicação entre o servidor e o InTouch, utilizou-se o OPCLink [10]. Para esta versão, os recursos disponibilizados são para escrita e leitura de variáveis do tipo R, M ou D. Nesta etapa, deve-se manter o HT1/HS1 aberto.

Passo 1: Criação e definição de novo tópico. Como na Figura 81, siga a sequência para abrir a janela de definição de tópicos.

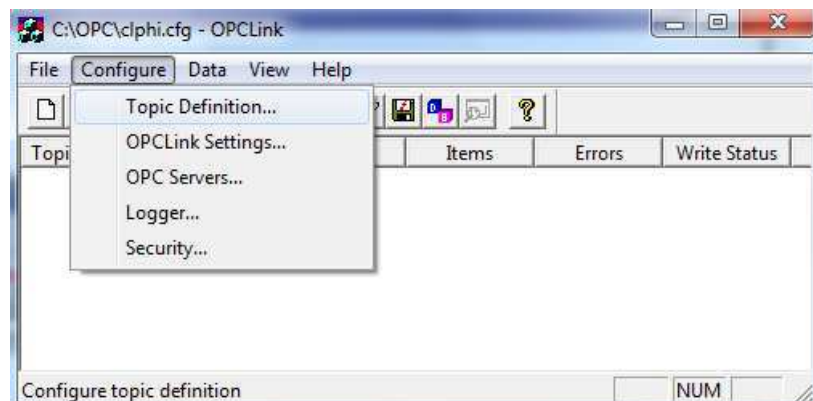


Figura 81 – Janela inicial do OPCLink.

Para criar novo tópico clique em *New*, Figura 82.

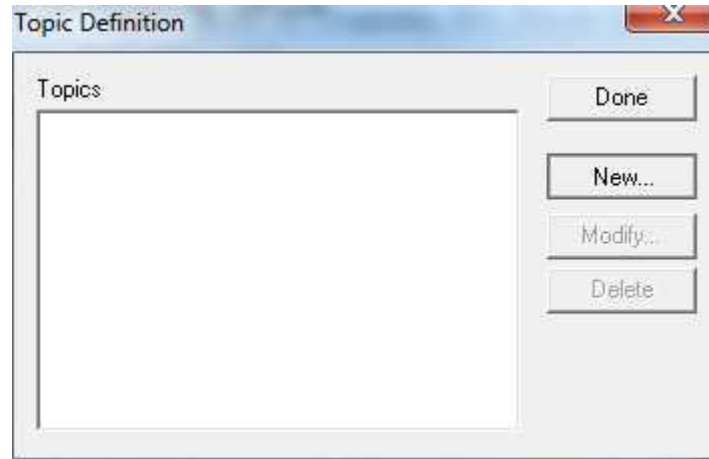


Figura 82 – Janela do OPCLink para criação de tópicos.

Em seguida, deve-se identificar o servidor. Neste caso é o *Intellution.HT1OPC*, que é o servidor do HT1 power Tool. Preencha o OPC Path de acordo com a variável de interesse. Neste caso, a variável escolhida foi D12, como na Figura 83.

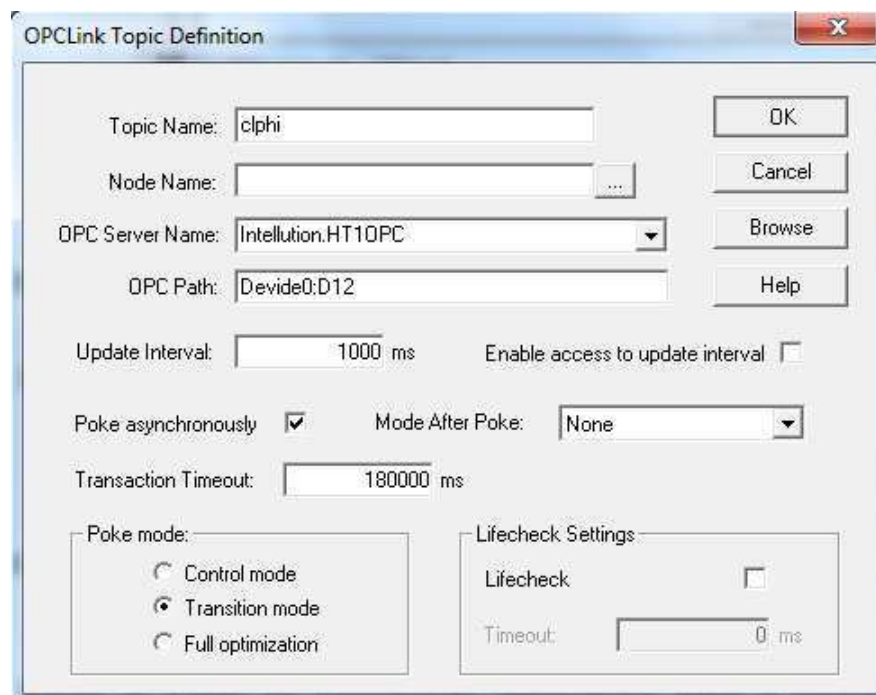


Figura 83 - Janela do OPCLink para configuração de tópicos.

Depois, clique em *Done*, Figura 84, e mantenha o OPCLink aberto.

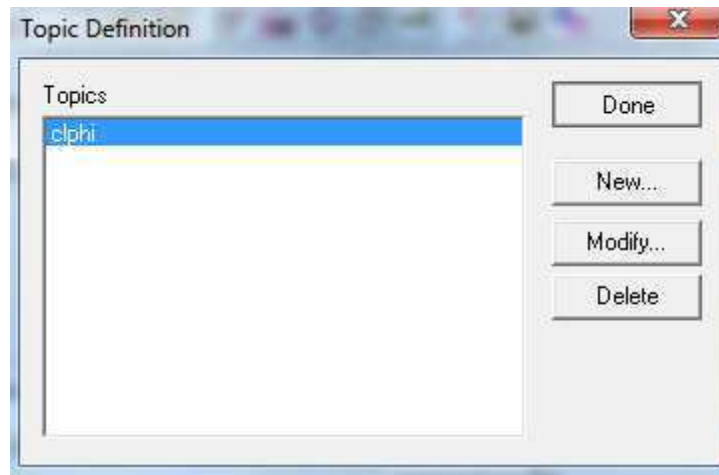


Figura 84 - Janela do OPCLink para definição de tópicos – novo tópico.

5.4 CLIENTE OPC

Passo 1: Abra o InTouch e crie uma nova aplicação.

Passo 2: Abrir a aplicação no InTouch Maker. Para isso, clique duas vezes na aplicação exemplo ou clique no botão direito e selecione a opção “abrir com o Intouch Maker”, Figura 85.

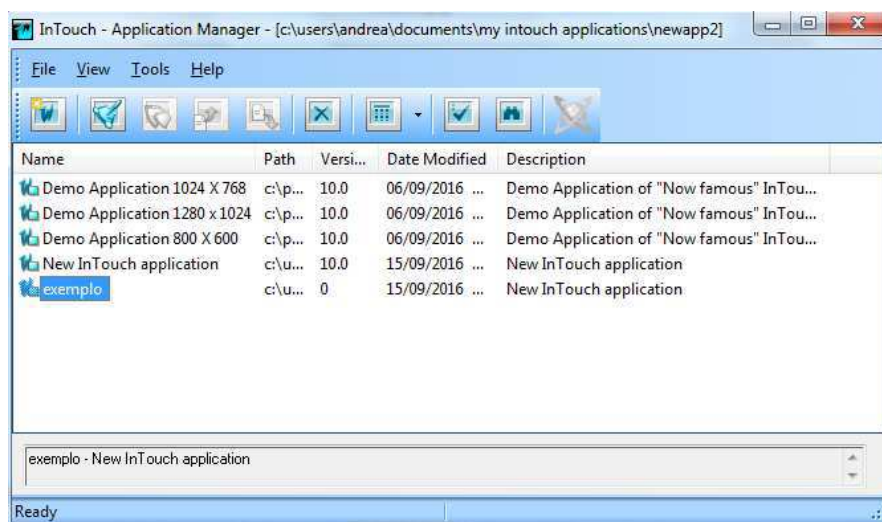


Figura 85 – Janela inicial do InTouch – Lista de aplicações.

Passo 3: Crie nova tela no InTouch Maker, Figura 86.

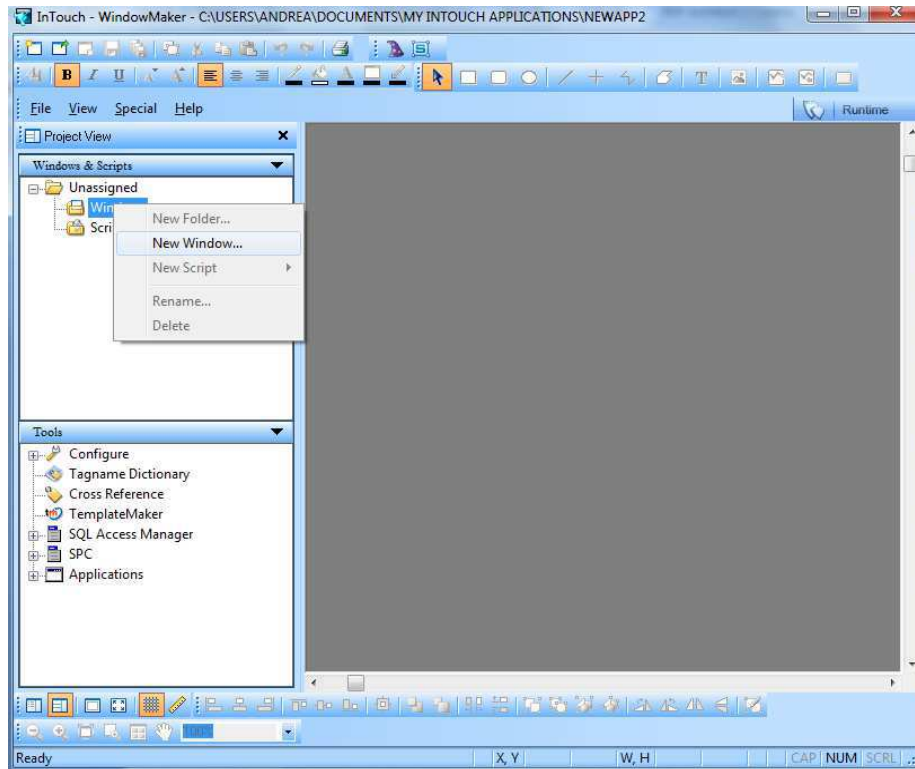


Figura 86 – Janela inicial do InTouch Maker – criação de nova tela.

Passo 4: Atribuir um nome à tela, neste caso o nome escolhido foi “tela 1”, Figura 87, e ajustar os parâmetros. Não é necessário preocupar-se com a largura e altura inicial da tela, pois pode ser alterado manualmente, expandindo ou diminuindo utilizando o cursor do mouse.

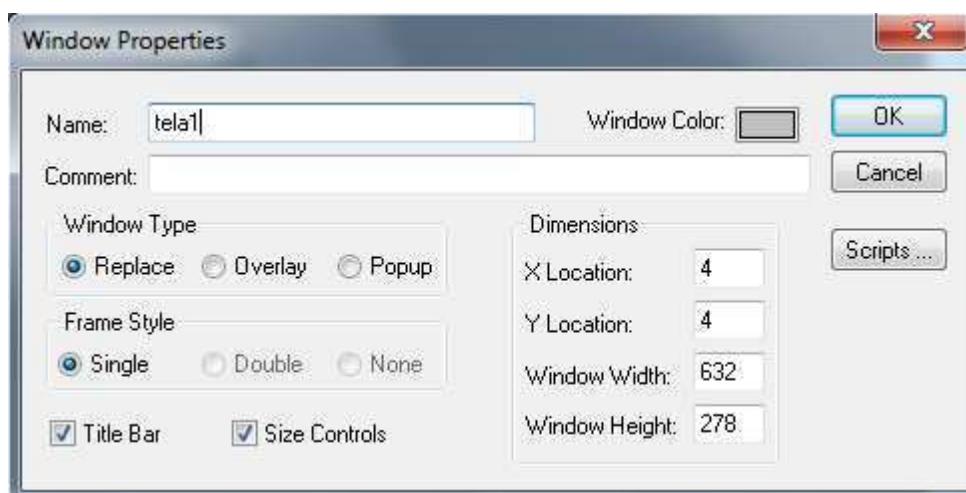


Figura 87 - Janela do INTOUCH MAKER para ajuste de parâmetros de nova tela.

Passo 5: Configurar por onde deverão ser acessados os valores atribuídos às *Tags*. É importante mencionar que o Topic Name deve ser o mesmo utilizado no OPCLink. Deve-se configurar os parâmetros de acordo com o apresentado na Figura 88.

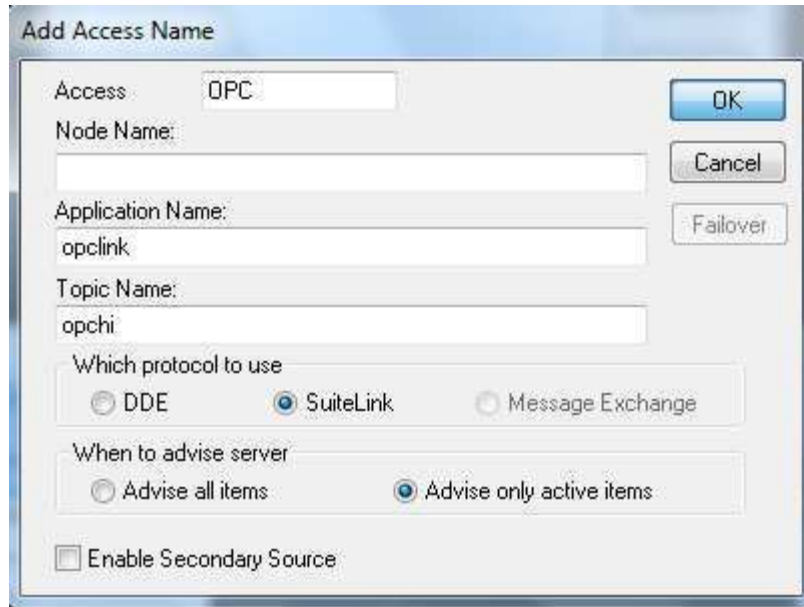


Figura 88: Janela para configuração do Access.

A comunicação é feita utilizando um protocolo TCP/IP. O SuiteLink é projetado especificamente para satisfazer as necessidades industriais, como integridade de dados, alta velocidade de processamento e diagnósticos mais fáceis.

Passo 6: Criação de uma nova Tag.

Abra a janela *Tagname Dictionary*, Figura 89. Siga a sequência:

Special>>*Tagname Dictionary*

Quando abrir a janela, clique em *Type* e selecione o tipo da tag, como na Figura 89. Em seguida, selecione o *Access Name OPC*, já configurado anteriormente.

Figura 89-Janela *Tagname Dictionary*.

Passo 7: Adicione um *display* para exibir o valor da Tag.

6 CONCLUSÃO

A principal contribuição desse trabalho foi a criação de duas interfaces gráficas para promover a interação do usuário com o sistema. A primeira interface trata-se da aplicação para um cenário de excessão, a calibração do motor. Essa interface, após realizar os experimentos, retorna ao usuário valores de parâmetros que ajustam a velocidade de rotação da sonda da esteira. A segunda interface desenvolvida é aplicável para o cenário de rotina, usado com mais frequência, no qual executa-se rotinas de ação e o usuário pode acompanhar o processo por meio de animação visual.

Foi fundamental o estudo do conceito de interface gráfica, bem como as ferramentas necessárias para seu desenvolvimento e configuração. No desenvolvimento das interfaces produzidas por esse trabalho, utilizou-se o método MCIE e diagramas UML para garantir a documentação necessária para facilitar o entendimento do usuário e o reuso do software. Além disso, necessitou-se entender o funcionamento do protocolo de comunicação OPC e configurar todas suas etapas para as aplicações.

REFERÊNCIAS

- [1] BATISTA, E. S. Instrumentação e Controle de Esteira Transportadora Utilizando CLP. Universidade Federal de Campina Grande (Trabalho de Conclusão de Curso). Campina Grande. 2015.
- [2] Nubotics WheelWatcher - WW-12. Disponível em: http://nubotics.com/products/ww12/ww12_manual.pdf . Acesso em 06/06/2016.
- [3] Neto, J. T. C. Controladores Lógicos Programáveis. Universidade Federal do Rio Grande do Norte (Apostila do curso de Eng. De Computação e Automação). Natal. 2011.
- [4] CARNEIRO. S. A. Supervisórios: Intouch. CEFETES-SERRA, ES (Apostila). 2007.
- [5] Wonderware FactorySuite: InTouch User's Guide. Version A, 2002.
- [6] M. F. Q. Vieira, “Notas de aula da disciplina Informática Industrial e Método MCIE,” Disponível em: https://sites.google.com/a/dee.ufcg.edu.br/fatima/cursosministrados/informaticaindustrial/downloads_infind. Acesso em 10/09/2016.
- [7] Practical UML: A Hands-On Introduction for Developers.
- [8] M. FONSECA, “Comunicação OPC – Uma abordagem prática. VI Process Automation Seminar,,” 2002
- [9] ENA0003300- HS1-OPC. Disponível em: <http://www.hitecnologia.com.br/software/softwaresoletos/hs1-opc> Acesso em 07/06/2016.
- [10] Wonderware OPCLink: InTouch User's Guide. Version A, 2001.

ANEXOS

A. Aplicação Calibração de Velocidade de Motor

SENTIDO HORÁRIO

Modo: On Show

```
S0Motor = 1650;
Const1 = 0;
n1 = 0;
const2 = 0;
n2 = 0;
const3 = 0;
n3 = 0;
const4 = 0;
n4 = 0;
const5 = 0;
n5 = 0;
```

Modo: While Showing

```
S0Motor = S0Motor + 1;
```

```
IF velocidade >= 0.5 AND velocidade <= 1.5 THEN
  const1 = S0Motor + const1 ;
  n1 = n1 + 1;
ENDIF;
```

```
IF velocidade >= 1.5 AND velocidade <= 2.5 THEN
  const2 = S0Motor + const2 ;
  n2 = n2 + 1;
ENDIF;
```

```
IF velocidade >= 2.5 AND velocidade <= 3.5 THEN
  const3 = S0Motor + const3 ;
  n3 = n3 + 1;
ENDIF;
```

```
IF velocidade >= 3.5 AND velocidade <= 4.5 THEN
  const4 = S0Motor + const4 ;
  n4 = n4 + 1;
ENDIF;
```

```
IF velocidade >= 4.5 AND velocidade <= 5.5 THEN
  const5 = S0Motor + const5 ;
  n5 = n5 + 1;
ENDIF;
```

```
IF S0Motor == 1250 THEN
  S0Motor1 = const1/n1;
  S0Motor2 = const2/n2;
  S0Motor3 = const3/n3;
```

```

S0Motor4 = const4/n4;
S0Motor5 = const5/n5;
ENDIF;

```

Botão: Iniciar calibração

```

S0Motor = 950;
S0Motor1 = 0;
S0Motor2 = 0;
S0Motor3 = 0;
S0Motor4 = 0;
S0Motor5 = 0;
const1=0;
const2=0;
const3=0;
const4=0;
const5=0;

```

SENTIDO ANTI-HORÁRIO

Modo: On Show

```
S0Motor = 1650;
```

```

const1ah = 0;
n1ah = 0;
const2ah = 0;
n2ah = 0;
const3ah = 0;
n3ah = 0;
const4ah = 0;
n4ah = 0;
const5ah = 0;
n5ah = 0;

```

Modo: While Showing

```
S0Motor = S0Motor + 1;
```

```

IF velocidade <= -0.5 AND velocidade >= -1.5 THEN
  const1ah = S0Motor + const1ah;
  n1ah = n1ah + 1;
ENDIF;

```

```

IF velocidade <= -1.5 AND velocidade >= -2.5 THEN
  const2ah = S0Motor + const2ah;
  n2ah = n2ah + 1;
ENDIF;

```

```

IF velocidade <= -2.5 AND velocidade >= -3.5 THEN
  const3ah = S0Motor + const3ah;
  n3ah = n3ah + 1;
ENDIF;

```

```

IF velocidade <= -3.5 AND velocidade >= -4.5 THEN
  const4ah = S0Motor + const4ah;
  n4ah = n4ah + 1;

```

```

ENDIF;

IF velocidade <= -4.5 AND velocidade >= -5.5 THEN
    const5ah = S0Motor + const5ah;
    n5ah = n5ah + 1;
ENDIF;

IF S0Motor == 2850 THEN
    S0Motor1AH = const1ah/n1ah;
    S0Motor2AH = const2ah/n2ah;
    S0Motor3AH = const3ah/n3ah;
    S0Motor4AH = const4ah/n4ah;
    S0Motor5AH = const5ah/n5ah;
ENDIF;

```

Botão: Iniciar calibração

```

S0Motor = 2550;
S0Motor1AH = 0;
S0Motor2AH = 0;
S0Motor3AH = 0;
S0Motor4AH = 0;
S0Motor5AH = 0;
const1ah=0;
const2ah=0;
const3ah=0;
const4ah=0;
const5ah=0;

```

B. Aplicação de Monitoramento e Execução de Rotina

LOGIN

Modo: On Show

```

user = “ ”;
senha= “ ”;
botao_login= “ ”;

```

Modo: While Showing (executada a cada 100 ms)

```

IF user = “aluno” AND senha = “1234” THEN
    botao_login ==1;
ENDIF;

```

EXECUÇÃO DE ROTINA DE AÇÃO 1

Modo: On Show

```

contador = 0;
sentido = 0;
deslocamento = 0;
state = 0;
state1=0;
state2=0;
state3=0;
S0Motor=1650;

```

PWMResistencia = 0;

Modo: While Showing (executada a cada 1 s)

IF sentido==0 THEN

IF SPvelocidade == 1 THEN
SOMotor = SOMotor1;
ENDIF;

IF SPvelocidade == 2 THEN
SOMotor = SOMotor2;
ENDIF;

IF SPvelocidade == 3 THEN
SOMotor = SOMotor3;
ENDIF;

IF SPvelocidade == 4 THEN
SOMotor = SOMotor4;
ENDIF;

IF SBO1==1 THEN
state = 1;
ENDIF;

IF state==1 AND SBO2==0 THEN
contador = contador + 1;
deslocamento = 9*contador*velocidade;
ENDIF;

IF SBO2 == 1 AND cont1 > 0 THEN
SOMotor=1650;
contador = 0;
deslocamento = 0;
cont1 = cont1 - 1;
PWMResistencia = 10*SPPWMResistencia;
state1=1;
state=0;
ENDIF;

IF cont1==0 AND SBO2==1 THEN
state1=1;
state=0;
ENDIF;

IF cont1==0 AND state1==1 THEN
contador = contador + 1;
deslocamento = 9*contador*velocidade;
ENDIF;

IF SBO3 == 1 AND cont2>0 THEN
SOMotor=1650;
contador = 0;
deslocamento = 0;
state1=0;
state2 = 1;
cont2 = cont2 - 1;
ENDIF;

```

IF cont2 == 0 AND SBO3==1 THEN
    state1=0;
    state2=1;
ENDIF;

IF cont2==0 AND state2==1 THEN
    contador = contador +1;
    deslocamento = 9*contador*velocidade;
ENDIF;

IF SBO4 ==1 THEN
    sentido=1;
    SOMotor=1650;
    contador = 0;
    deslocamento = 0;
    state2=0;
    state3=1;
ENDIF;

ELSE

IF SPvelocidade == 1 THEN
    SOMotor = SOMotor1AH;
ENDIF;

IF SPvelocidade == 2 THEN
    SOMotor = SOMotor2AH;
ENDIF;

IF SPvelocidade == 4 THEN
    SOMotor = SOMotor3AH;
ENDIF;

IF SPvelocidade == 3 THEN
    SOMotor = SOMotor4AH;
ENDIF;

IF SBO1 == 0 THEN
    contador = contador +1;
    deslocamento = - 9*contador*velocidade;
ELSE
    SOMotor=1650;
    contador = 0;
    deslocamento = 0;
    state3=0;
    state=1;
ENDIF;

ENDIF;

```

EXECUÇÃO DE ROTINA DE AÇÃO 2

Modo: On Show

```

contador = 0;
sentido = 1;
deslocamento = 0;
state = 0;
state1=0;
state2=0;

```

```
state3=0;
SOMotor=1650;
PWMResistencia = 0;
```

Modo: While Showing (executada a cada 1 s)

```
IF sentido==1 THEN

    IF SPvelocidade == 1 THEN
        SOMotor = SOMotor1AH;
    ENDIF;

    IF SPvelocidade == 2 THEN
        SOMotor = SOMotor2AH;
    ENDIF;

    IF SPvelocidade == 4 THEN
        SOMotor = SOMotor3AH;
    ENDIF;

    IF SPvelocidade == 3 THEN
        SOMotor = SOMotor4AH;
    ENDIF;

    IF SBO4==1 THEN
        state = 1;
    ENDIF;
    IF state==1 AND SBO3==0 THEN
        contador = contador +1;
        deslocamento = - 9*contador*velocidade;
    ENDIF;

    IF SBO3 == 1 AND cont2 > 0 THEN
        SOMotor=1650;
        contador = 0;
        deslocamento = 0;
        cont2 = cont2 - 1;
        PWMResistencia = 10*SPPWMResistencia;
        state1=1;
        state=0;
    ENDIF;

    IF cont2==0 AND SBO3==1 THEN
        state1=1;
        state=0;
    ENDIF;

    IF cont2==0 AND state1==1 THEN
        contador = contador + 1;
        deslocamento = - 9*contador*velocidade;
    ENDIF;

    IF SBO2 ==1 AND cont1>0 THEN
        SOMotor=1650;
        contador = 0;
        deslocamento = 0;
        state1=0;
        state2 =1;
        cont1 = cont1 - 1;
    ENDIF;
```

```

IF cont1 == 0 AND SBO2==1 THEN
    state1=0;
    state2=1;
ENDIF;

IF cont1==0 AND state2==1 THEN
    contador = contador +1;
    deslocamento = - 9*contador*velocidade;
ENDIF;

IF SBO1 ==1 THEN
    sentido=0;
    SOMotor=1650;
    contador = 0;
    deslocamento = 0;
    state2=0;
    state3=1;
ENDIF;

ELSE

    IF SPvelocidade == 1 THEN
        SOMotor = SOMotor1;
    ENDIF;

    IF SPvelocidade == 2 THEN
        SOMotor = SOMotor2;
    ENDIF;

    IF SPvelocidade == 3 THEN
        SOMotor = SOMotor3;
    ENDIF;

    IF SPvelocidade == 4 THEN
        SOMotor = SOMotor4;
    ENDIF;
    IF SBO4 == 0 THEN
        contador = contador +1;
        deslocamento = 9*contador*velocidade;
    ELSE
        SOMotor=1650;
        contador = 0;
        deslocamento = 0;
        state3=0;
        state=1;
    ENDIF;
ENDIF;

```

C. Formulário para o levantamento do Perfil do Usuário

Características do usuário, escolhidas pelo projetista, de acordo com a relevância, para o projeto. Levantamento baseado em:

Fatos Opinião do usuário Dados medidos ou observados

Parte I – Características Gerais

Faixa etária: a partir dos 20 anos;
 Sexo: Masculino ou feminino;
 Habilidades específicas necessárias para executar a tarefa:
 Níveis de percepção: percepção visual;
 Grau de instrução: cursando engenharia elétrica;
 Função desempenhada: **aluno**;
 Tarefa realizada na Função: monitora a temperatura das câmaras de aquecimento e resfriamento e executa rotinas de ação.
 Frequência de execução das Tarefas na função: mensalmente;
 Objetivos da empresa: ensinar os alunos o conceito de sistemas supervisórios de forma prática.
 Motivações do usuário: Aprender;

Parte II - CONHECIMENTO CONCEITUAL necessário à execução das tarefas:

Conhecimento Semântico

Nível de experiência¹

Função (aluno): experiente;
 Método (através do supervisório): inexperiente;
 Tarefa (monitora a temperatura das câmaras de aquecimento e resfriamento e executa rotinas de ação): inexperiente;
 Computadores (uso geral): experiente;
 Ferramentas utilizadas na execução das tarefas (ou similares): experiente;

Conhecimento Sintático

Nível de experiência

Uso de dispositivos especiais de interação (ex. TST): computador;
 Uso de terminologia específica (do processo): especialista;

Parte III - ESTILO COGNITIVO:

Aprendizado: treinamento informal;
 Capacidade de solucionar problemas: com ajuda;
 Capacidade de reter o aprendizado (alta, média, baixa): média;

Personalidade:

Nível de curiosidade (baixo, médio, elevado): elevado;
 Nível de persistência (baixo, médio, elevado): médio;
 Nível de inovação (baixo, médio, elevado): elevado;
 Estilo de tomada de decisão (Impulsivo, Reflexivo): reflexivo;

Formulário para o levantamento do Perfil do Usuário

© 2011, Laboratório de Interfaces Homem-Máquina (LIHM), UFCG



¹ nível de experiência: inexperiente, experiente, especialista, ...

Características do usuário, escolhidas pelo projetista, de acordo com a relevância, para o projeto. Levantamento baseado em:

Fatos Opinião do usuário Dados medidos ou observados

Parte I – Características Gerais

Faixa etária: a partir dos 25 anos;

Sexo: Masculino ou feminino;

Habilidades específicas necessárias para executar a tarefa. :

Níveis de percepção: boa acuidade visual;

Habilidades motoras: rapidez de resposta;

Grau de instrução: Superior;

Função desempenhada: **Professor**;

Tarefa realizada na Função:

1- Monitora a temperatura das câmaras de aquecimento e resfriamento e executa rotinas de ação.

2- Auxilia alunos a utilizar o supervisor.

Frequência de execução das Tarefas na função: mensal;

Objetivos da empresa: ensinar os alunos o conceito de sistemas supervisórios de forma prática.

Motivações do usuário: Ensinar;

Parte II - CONHECIMENTO CONCEITUAL necessário à execução das tarefas:

<u>Conhecimento Semântico</u>	Nível	de
experiência ²		

Função (professor): experiente;

Método (através do supervisor): experiente;

Tarefa (monitoramento): experiente;

Computadores (uso geral): experiente;

Ferramentas utilizadas na execução das tarefas (ou similares): experiente;

<u>Conhecimento Sintático</u>	Nível de experiência

Uso de dispositivos especiais de interação (ex. TST): computador;

Uso de terminologia específica (do processo): experiente;

Parte III - ESTILO COGNITIVO:

Aprendizado: não se aplica;

Capacidade de solucionar problemas (sozinho, com ajuda): sozinho;

Capacidade de reter o aprendizado (alta, média, baixa): média;

Personalidade:

Nível de curiosidade (baixo, médio, elevado): elevado;

Nível de persistência (baixo, médio, elevado): médio;

Nível de inovação (baixo, médio, elevado): baixo;

² nível de experiência: inexperiente, experiente, especialista, ...