

Mateus de Oliveira Queiroga

# **Relatório de Estágio Supervisionado**

Campina Grande

Outubro de 2017

Mateus de Oliveira Queiroga

## **Relatório de Estágio Supervisionado**

Relatório de Estágio Supervisionado submetido à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.

Trabalho aprovado. Campina Grande, 9 de outubro de 2017:

---

Professor George Acioli Júnior  
Orientador

---

Avaliador

Campina Grande  
Outubro de 2017

Mateus de Oliveira Queiroga

## **Relatório de Estágio Supervisionado**

Relatório de Estágio Supervisionado submetido à Coordenação do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Engenharia Elétrica.

Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Engenharia Elétrica

Orientador: Professor George Acioli Júnior

Campina Grande  
Outubro de 2017

# Agradecimentos

Primeiramente agradeço ao Grande Arquiteto do Universo por toda a força e coragem para vencer tantos obstáculos que surgiram neste período de graduação em Engenharia Elétrica.

Para a realização deste sonho de ser um Engenheiro Eletricista contei com o apoio de inúmeras pessoas e instituições, mas gostaria particularmente de destacar e agradecer:

- A minha família em geral, em especial a minha mãe Marineide de Oliveira Queiroga e meu pai Francisco de Assis Queiroga.
- A Universidade Federal de Campina Grande, e em especialmente a Unidade Acadêmica de Engenharia Elétrica, por toda a excelente estrutura de ensino e apoio aos estudantes, em especial a Sra. Adail Silva Paz.
- Aos meus inúmeros queridos amigos e colegas que compartilharam varias horas de aula, muitas provas e muitas horas sem dormir por causa dos estudos.
- E por fim, um agradecimento mais que especial a minha amada esposa Larissa Marie e ao meu filho Miguel Queiroga pelo apoio, compreensão e motivação.

# Resumo

Este relatório apresenta o trabalho realizado pelo aluno Mateus de Oliveira Queiroga em seu estágio supervisionado no Laboratório de Instrumentação Eletrônica e Controle (LIEC), pertencente ao Departamento de Engenharia Elétrica (DEE) da Universidade Federal de Campina Grande (UFCG), no intervalo de 05 de junho de 2017 a 25 de agosto de 2017, sob orientação do Professor George Acioli Júnior e supervisão do Professor Péricles Rezende Barros. O foco do estágio foi na elaboração de um manual de utilização da IHM PanelView Plus da Allen-Bradley e na manutenção e aprimoramento da sala de automação do Laboratório de Instrumentação Eletrônica e Controle.

**Palavras-chaves:** Estágio Supervisionado. CLP. IHM. FactoryTalk View.

# Lista de ilustrações

Figura 1 – Faixada do LIEC. . . . .	11
Figura 2 – Sala de Automação. . . . .	12
Figura 3 – Plataforma Didática utilizada. . . . .	13
Figura 4 – Compact Logix 1769. . . . .	14
Figura 5 – PanelView Plus 400. . . . .	15
Figura 6 – PanelView Plus 400. . . . .	17
Figura 7 – Interface RSLogix5000. . . . .	17
Figura 8 – Interface RSLogix5000. . . . .	18
Figura 9 – Tag Browser mostrando as Tags do CLP. . . . .	19
Figura 10 –Objetos. . . . .	19
Figura 11 –Tela exemplo. . . . .	20
Figura 12 –Transfer Utility. . . . .	21

# Lista de tabelas

Tabela 1 – Entradas digitais. . . . .	16
Tabela 2 – Saídas digitais. . . . .	16

# Lista de abreviaturas e siglas

LIEC	Laboratório de Instrumentação Eletrônica e Controle
CLP	Controlador Lógico Programável
VDC	Voltage Direct Current
IHM	Interface Homem Máquina
CA	Corrente Alternada
CC	Corrente Contínua
NA	Normalmente Aberta
NF	Normalmente Fechada

# Lista de símbolos

$\Delta$  Letra grega Delta

# Sumário

<b>1</b>	<b>Introdução</b>	<b>10</b>
<b>2</b>	<b>Local do Estágio</b>	<b>11</b>
2.1	Sala de Automação	11
<b>3</b>	<b>Atividades Desenvolvidas</b>	<b>13</b>
3.1	Equipamentos utilizados	13
3.1.1	CLP Compact Logix	13
3.1.2	Entradas e saídas digitais	15
3.1.3	PanelView Plus 400	15
3.2	Softwares utilizados	16
3.2.1	RSLogix5000	16
3.2.2	FactoryTalk View Studio	18
3.3	Tutorial de configuração do PanelView Plus	18
	<b>Conclusão</b>	<b>22</b>
	<b>Referências</b>	<b>23</b>

# 1 Introdução

Neste relatório são apresentadas as atividades desenvolvidas no Estágio Supervisionado realizado por Mateus de Oliveira Queiroga no Laboratório de Instrumentação Eletrônica e Controle (LIEC), pertencente ao Departamento de Engenharia Elétrica (DEE) da Universidade Federal de Campina Grande (UFCG), com carga horária de 180 horas.

Dentre as atividades realizadas durante o estágio estão inclusas a manutenção e aprimoramento do laboratório de automação, estudo e pesquisa sobre as ferramentas a serem utilizadas e elaboração de um manual de utilização para IHM PanelView Plus 400.

No segundo capítulo deste relatório, é apresentado o local onde foram realizadas as atividades do estágio.

No terceiro capítulo estão descritas as atividades desenvolvidas, com uma breve explicação sobre cada equipamento e software utilizado no estágio.

O quarto e último capítulo é feita uma conclusão geral do estágio realizado.

## 2 Local do Estágio

O Laboratório de Instrumentação Eletrônica e Controle, LIEC, é um Laboratório do Departamento de Engenharia Elétrica cujos integrantes participam de projetos nas áreas de automação, controle e instrumentação.

Dessa forma, o embasamento teórico e prático necessário para o futuro engenheiro eletricista é desenvolvido e solidificado. Modernas instalações e um grupo de pessoas altamente qualificado fazem do LIEC um ambiente bastante agradável para se trabalhar e pesquisar.

Figura 1 – Faixada do LIEC.



Fonte: (LIEC, )

### 2.1 Sala de Automação

A Sala de automação, localizada no segundo andar do prédio, possui 9 computadores, 9 CLP's e 6 plantas didáticas, dentre as quais quatro são processos peltier e dois são sopradores de calor.

Esta sala é utilizada por alunos da graduação e pós-graduação para desenvolver atividades de ensino e pesquisa. Esta sala também é utilizada como apoio para as atividades de laboratório da disciplina de Sistemas de Automação Industrial para as aplicações praticas utilizando CLP.

As atividades do estágio foram focadas na manutenção e aprimoramento desta sala.

Figura 2 – Sala de Automação.



Fonte: O autor.

## 3 Atividades Desenvolvidas

Este capítulo irá apresentar as principais atividades desenvolvidas pelo estagiário no laboratório. Inicialmente explanando sobre os equipamentos que foram utilizados pelo estagiário e em seguida sobre suas atividades práticas.

### 3.1 Equipamentos utilizados

Os equipamentos utilizados no estágio foram o CLP Compact Logix 1769 e a IHM PanelView Plus 400. Eles encontram-se na sala de automação do LIEC na plataforma didática vista na figura 3.

Figura 3 – Plataforma Didática utilizada.

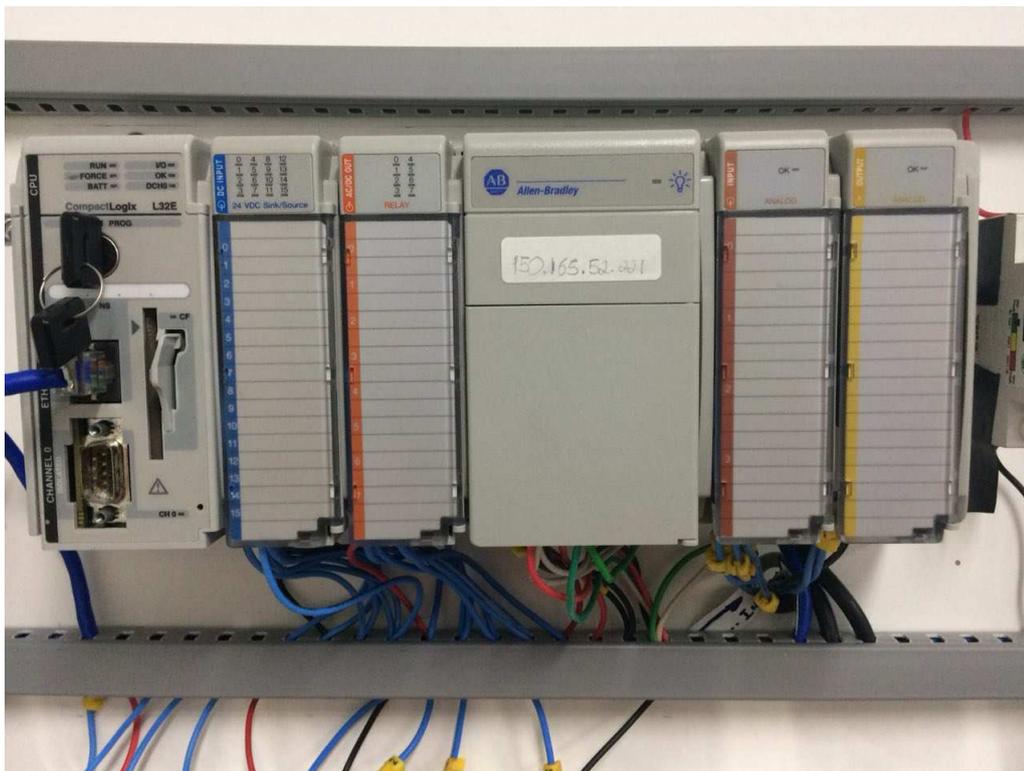


Fonte: O autor.

#### 3.1.1 CLP Compact Logix

O Controlador Lógico Programável, ou CLP, é um equipamento projetado para comandar e monitorar máquinas ou processos industriais. Neste estágio foi utilizado o CLP da Rockwell Automation, modelo Compact Logix 1769, como pode ser visto na figura 4, que possui 4 módulos integrados.

Figura 4 – Compact Logix 1769.



Fonte: O autor.

O módulo 1769-IQ16 DC Input disponibiliza 16 entradas digitais e possui alimentação 24 VDC.

O módulo 1769-OW8 AC/DC Output disponibiliza 8 saídas digitais, normalmente abertas através de relés e possui alimentação de 24 VDC.

O módulo 1769-IF4 Analog Input com alimentação 24 VDC, disponibiliza 4 entradas diferenciais analógicas isoladas, que podem ser de corrente ou tensão, nos seguintes intervalos:

- $\pm 10V$
- 0-10V
- 0-5V
- 1-5V
- 0-20mA
- 4-20mA
- 1769-OF2 Analog Output

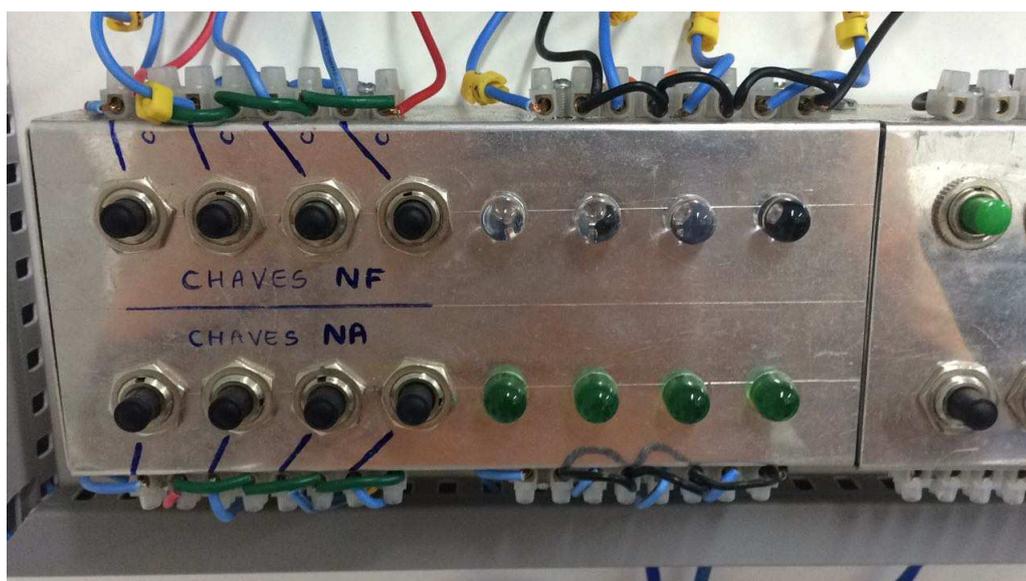
Por fim, temos o módulo 1769-OF2 Analog Output com alimentação 24 VDC, que possui 2 saídas analógicas de tensão/corrente, cujo os intervalos podem ser os seguintes:

- $\pm 10V$
- 0-10V
- 0-5V
- 1-5V
- 0-20mA
- 4-20mA

### 3.1.2 Entradas e saídas digitais

Os módulos de entrada e saída digitais que estão conectados as chaves e leds, respectivamente, podem ser vistos na figura 5. As quatro chaves inferiores são chaves normalmente abertas e as quatro chaves superiores são chaves normalmente fechadas.

Figura 5 – PanelView Plus 400.



Fonte: O autor.

As chaves e os LEDs estão conectados às entradas e saídas do CLP conforme as tabelas 1 e 2, respectivamente.

### 3.1.3 PanelView Plus 400

O PanelView Plus 400 mostrado na figura 6, que foi a IHM utilizada neste estágio, oferece ao usuário unidades de configurações base, módulos de comunicação, visor gráfico em tom cinza, suporte de entrada para teclado ou mouse podendo ser alimentado por uma fonte de alimentação CA (85 a 264 V) ou CC (18 a 30 V).

Como unidades de configurações base, esta IHM conta com uma entrada Serial (RS-232), interface de Ethernet e comunicações modulares. Os módulos de comunicação

Tabela 1 – Entradas digitais.

Chave	Tipo	Endereço
1	NA	Local:1:I.Data.0
2	NA	Local:1:I.Data.1
3	NA	Local:1:I.Data.2
4	NA	Local:1:I.Data.3
5	NF	Local:1:I.Data.4
6	NF	Local:1:I.Data.5
7	NF	Local:1:I.Data.6
8	NF	Local:1:I.Data.7

Tabela 2 – Saídas digitais.

LED	Endereço
1	Local:2:O.Data.0
2	Local:2:O.Data.1
3	Local:2:O.Data.2
4	Local:2:O.Data.3
5	Local:2:O.Data.4
6	Local:2:O.Data.5
7	Local:2:O.Data.6
8	Local:2:O.Data.7

oferecem recursos complementares de unidades de configuração base com uma interface de comunicação modular.

A IHM também conta com uma porta USB para conexão de mouse, teclado, impressora, scanner de código de barras ou outros dispositivos.

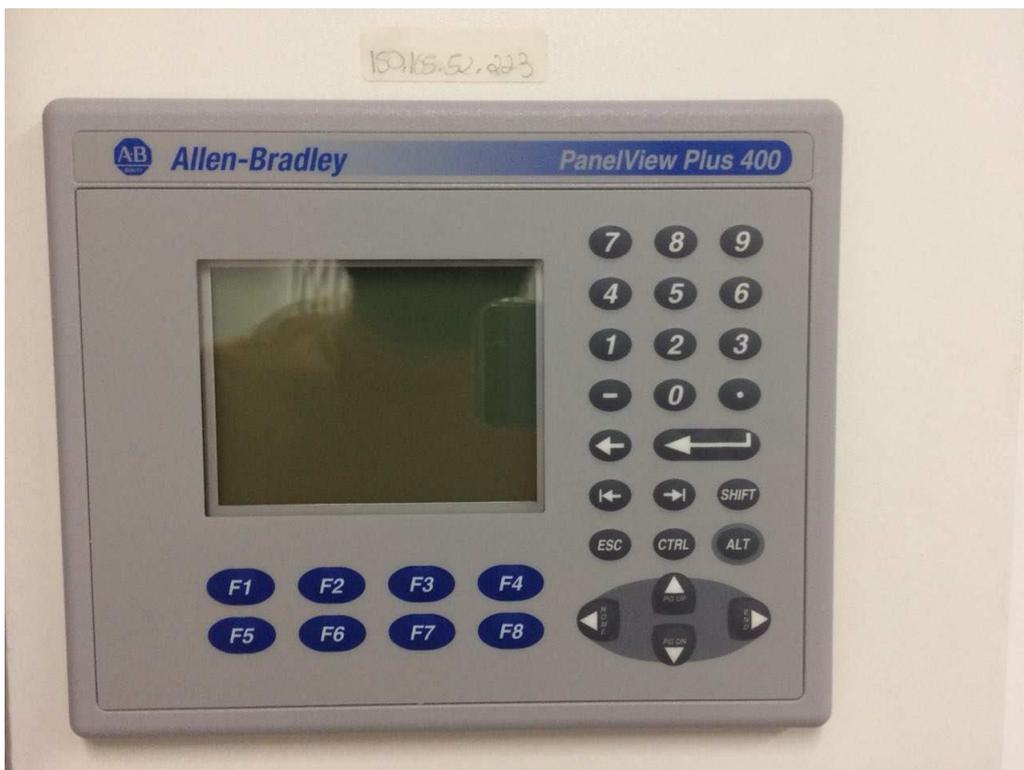
## 3.2 Softwares utilizados

Durante o estágio, foi necessário desenvolver habilidades no uso de alguns softwares para realização das atividades. Inicialmente o RSLogix5000 que é o software responsável pela programação do CLP Compact Logix 1769 e em seguida o Factory Talk View Studio que é o software responsável pela programação da IHM PanewViewl Plus.

### 3.2.1 RSLogix5000

Para desenvolver as aplicações para o CLP foi necessário um estudo do software RSLogix 5000 juntamente com os conhecimentos em linguagem de programação Ladder aprendidos na disciplina Sistemas de Automação Industrial para a programação do mesmo. Para o aprendizado desta ferramenta o estagiário dedicou suas primeiras horas de trabalho ao estudo do manual da mesma, ([ROCKWELL, b](#)).

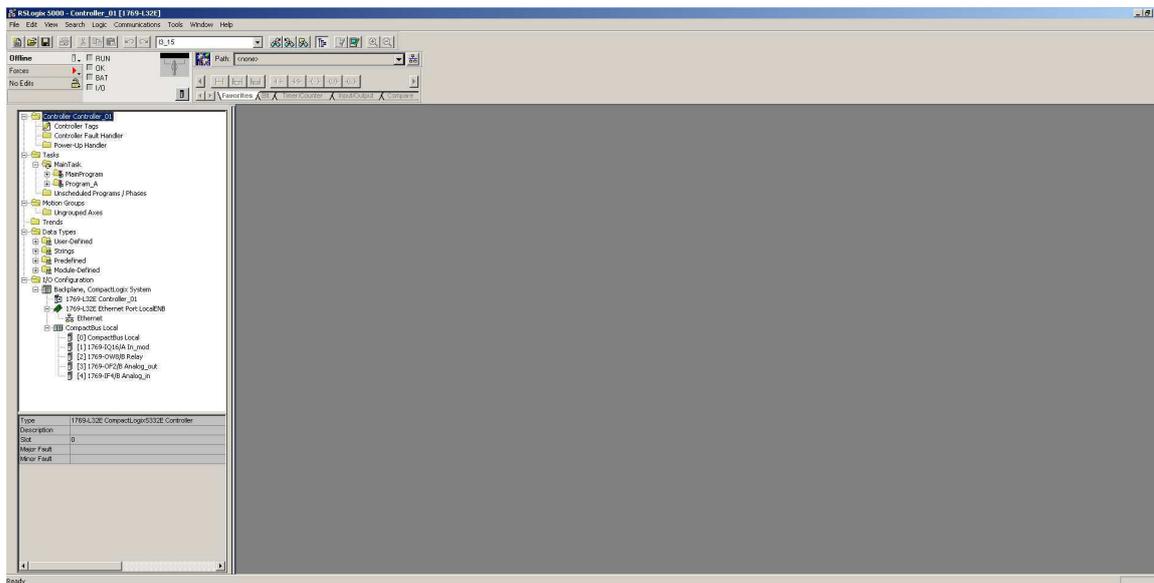
Figura 6 – PanelView Plus 400.



Fonte: O autor.

A figura 7 mostra a interface do software. O RSLogix 5000 V15.02.00 não possui compatibilidade com o sistema operacional utilizado nos computadores do laboratório, portanto para utilizá-lo se faz necessário o uso de uma máquina virtual com Windows XP. Foi utilizado o software VMware Workstation 12 Player para a utilização do RSLogix5000.

Figura 7 – Interface RSLogix5000.

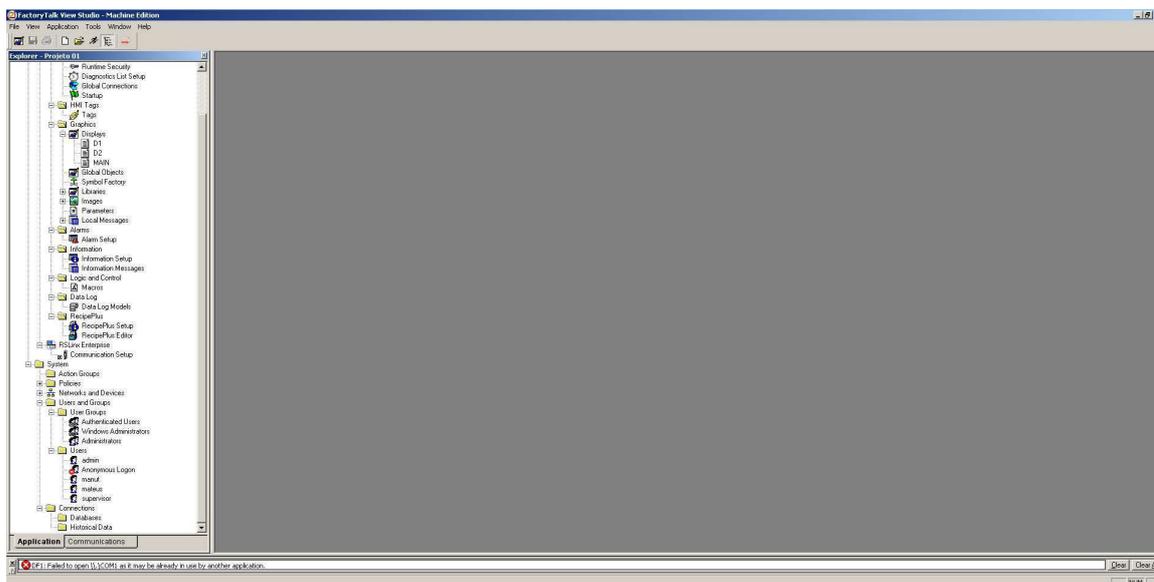


Fonte: O autor.

### 3.2.2 FactoryTalk View Studio

Para desenvolver as aplicações para a IHM foi necessário um estudo sobre o software FactoryTalk View Studio através de (ROCKWELL, a). A figura 8 mostra a interface do FactoryTalk View Studio, que foi o software foco do estágio. O FactoryTalk View Studio conta com uma diversidade enorme de recursos e ferramentas para programação dos vários modelos de IHM PanelView Plus. Algumas ferramentas não foram utilizadas pois o PanelView Plus 400 não suporta certos recursos.

Figura 8 – Interface RSLogix5000.



Fonte: O autor.

## 3.3 Tutorial de configuração do PanelView Plus

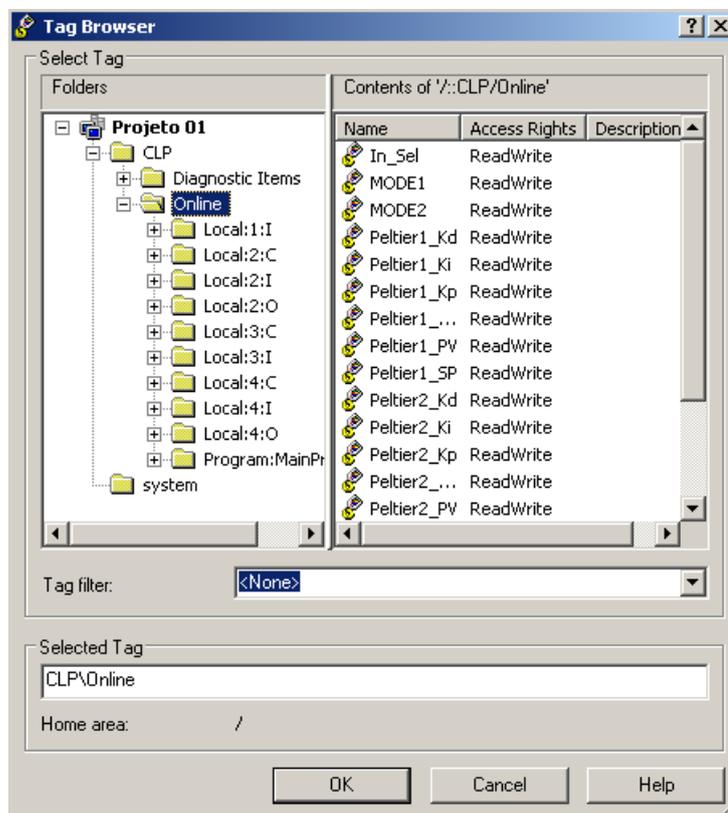
A principal atividade do estagiário foi o desenvolvimento de um manual de utilização do software FactoryTalk View Studio, que se encontra anexado a este relatório, incluindo as configurações do PanelView Plus 400. No título do tutorial não consta o modelo 400, pois o mesmo pode ser utilizado para todos os modelos PanelViewPlus.

O primeiro capítulo do tutorial ensina sobre a criação de um projeto no software, bem como todas as configurações iniciais do mesmo. Partindo em seguida para as configurações de comunicação do software com a IHM e com o CLP. Com as configurações de comunicação com o CLP realizadas, o software terá acesso às Tags do CLP, podendo monitorá-las e até mesmo alterá-las, tornando possível acionar determinados equipamentos conectados ao CLP através da IHM.

Após as configurações iniciais, o segundo capítulo do tutorial ensina sobre a criação e configuração das Tags e telas do projeto. A figura 9 mostra o Tag Browser do FactoryTalk

exibindo as tags do software que está no CLP, podendo associa-las às tags do software a ser programado na IHM.

Figura 9 – Tag Browser mostrando as Tags do CLP.



Fonte: O autor.

Quanto às configurações das Telas do projeto, o tutorial explica cada ferramenta que pode ser utilizada na aplicação, e como realizar a configuração de cada um dos objetos. A figura 10 mostra a barra de objetos que podem ser utilizados na aplicação.

Figura 10 – Objetos.



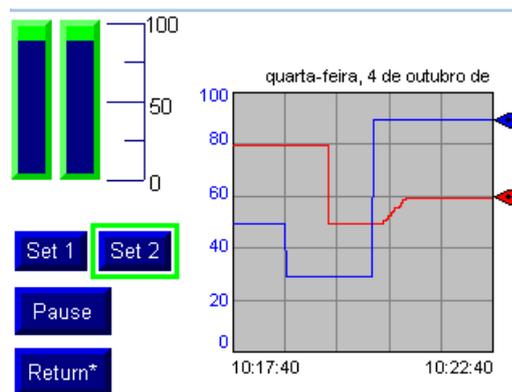
Fonte: O autor.

A figura 11 mostra um exemplo de uma aplicação criada para o entendimento de alguns objetos. Nela pode-se observar os seguintes objetos:

- Numeric Input Enable - Os botões Set1 e Set2, que servem como entrada de um número a ser salvo em duas Tags analógicas, Analog e Analog2 criadas na memória da IHM.
- Bar Graph - Uma barra que mostra o nível atual de uma variável. Nota-se que a tela possui duas barras, referentes às Tags Analog e Analog2.

- Scale - Para se ter uma noção de escala das Bar Graphs.
- Trend Object - Gráfico que monitora o estado de uma variável ao longo do tempo, como pode ser visto na figura 11, onde está sendo mostrado as variações das duas Tags analógicas nos últimos 5 minutos.

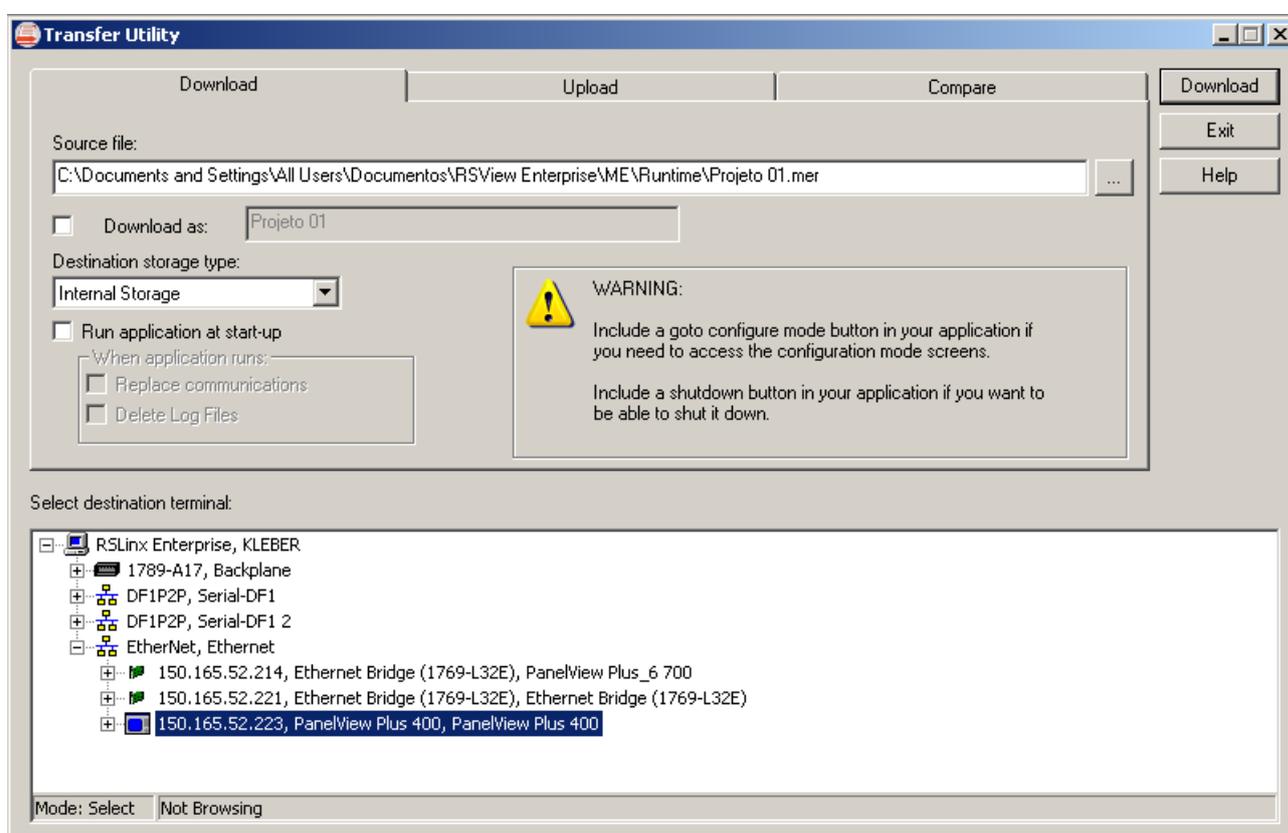
Figura 11 – Tela exemplo.



Fonte: O autor.

O terceiro capítulo do tutorial, ensina sobre algumas configurações básicas a serem realizadas na IHM, para que a comunicação e a aplicação a ser implementada funcionem corretamente. O quarto e último capítulo ensina a usar o Transfer Utility, que pode ser visto na figura 12. Este é utilizado para o envio das aplicações para a memória interna da IHM, bem como para a transferência de arquivos da memória da IHM para o computador.

Figura 12 – Transfer Utility.



Fonte: O autor.

# Conclusão

Como o estágio foi realizado com o foco na elaboração de um material didático, foi possível ter adquirido um excelente conhecimento sobre os equipamentos e os softwares que foram utilizados. Além disto, o uso do CLP e da IHM acrescentou muito ao conhecimento que pode ser utilizado no mercado de trabalho. Utilizar equipamentos e tecnologias que já são ou serão prática na indústria conseguiu enriquecer bastante o estágio.

Devido ao tempo disponível para o estágio e o cronograma previamente estabelecido, não foi viável realizar mais atividades, mas a contribuição documentada deste estágio para o laboratório permitirá que futuros alunos aprendam com menos dificuldade a programar um sistema supervisorio em uma IHM utilizando o FactoryTalk View Studio.

O manual elaborado ficou bastante simples e seu arquivo editável se encontra salvo no laboratório para possíveis atualizações, correções e melhoramentos futuros.

# Referências

LIEC. *Laboratório de Instrumentação Eletrônica e Controle*. Disponível em: [<http://liec.ufcg.edu.br/>](http://liec.ufcg.edu.br/). Citado na página 11.

ROCKWELL, A. I. *FactoryTalk View Machine Edition*. Disponível em: [<http://www.rockwellautomation.com/>](http://www.rockwellautomation.com/). Citado na página 18.

ROCKWELL, A. I. *Início Rápido dos Controladores Logix5000*. Disponível em: [<http://www.rockwellautomation.com/>](http://www.rockwellautomation.com/). Citado na página 16.

# **ANEXO** - Manual de utilização do PanelView Plus

Sistemas de Automação Industrial

9 de outubro de 2017

# 1 Criando e configurando o projeto

## 1.1 Criando o projeto

Abra o FactoryTalk View Studio (Iniciar > Programas > Rockwell Software > FactoryTalk View > FactoryTalk View Studio).

Na aba **New**, crie um novo projeto, como mostrado na Figura 1.

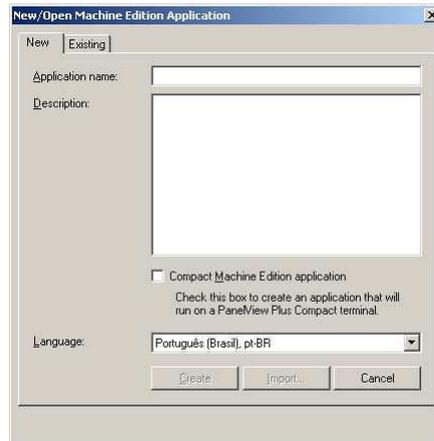


Figura 1: Nova aplicação.

Observe que a opção **Compact Machine Edition application** não deverá ser selecionada pois não se aplica ao HMI que utilizaremos no laboratório.

Clique em **Create** e o programa irá criar uma estrutura básica para seu projeto, como podemos observar na figura 2.

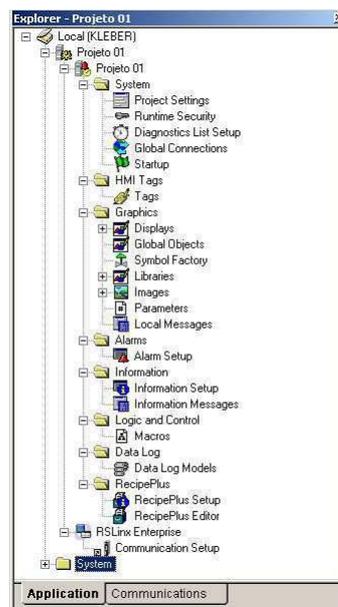


Figura 2: Estrutura básica do projeto.

## 1.2 Configurando a comunicação da HMI

Clique em **RSLinx Enterprise > Communication Setup**, selecione **Create a new configuration** e depois em **Concluir**

Clique com o botão direito em **Ethernet** e selecione **Add Device**. Ver figura 3.

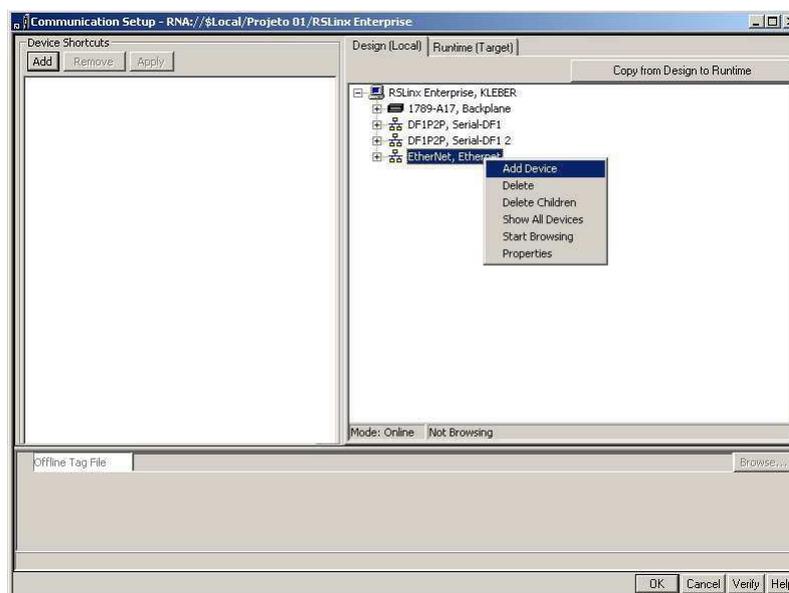


Figura 3: Configurando a comunicação.

Em **EtherNetIP Devices** selecione o HMI a ser utilizado (**PanelView Plus 400**) e pressione **OK**.

Escolha um nome para o HMI e coloque o endereço de IP do mesmo. Ao apertar OK seu HMI deverá aparecer e o Factory Talk já consegue se comunicar com ele.

OBS: Para saber o endereço IP do seu HMI, consulte o item 3.1 deste manual.

## 1.3 Configurando a comunicação com o CLP

Clique com o botão direito em **Ethernet** e selecione **Add Device**. Ver figura 3.

Em **EtherNetIP Devices** selecione o CLP a ser utilizado e pressione OK.

## 1.4 Configurações de execução da aplicação

Abra o Project Settings (**System > Project Settings**). Aqui você irá configurar a resolução da tela para o modelo do seu PanelView. Em **Project window size** selecione **PVPlus 400/600 (320x240)**.

Abra o Startup (**System > Startup**). Desmarque as opções **Alarms** e **Information messages**, marque a opção **Initial Graphics** escolha a tela inicial da aplicação. (**MAIN**).

## 2 Criando e configurando as tags e telas do projeto

### 2.1 Tags

Clique em Tags (**HMI Tags > Tags**). Aqui você poderá criar tags para sua aplicação. Digite o nome da Tag e uma descrição para identificação da mesma.

Escolha o tipo (**Análogica, digital ou string**).

No campo **Data source**, você irá direcionar o endereço da tag que está sendo criada, podendo ser diretamente na memória do PanelView ou fazendo referência a alguma tag de algum dispositivo (Por exemplo, o CLP). Abra o Tag Browser (clcando em ...). O CLP que foi configurado em 1.3 deverá aparecer no browser, com na figura 4.

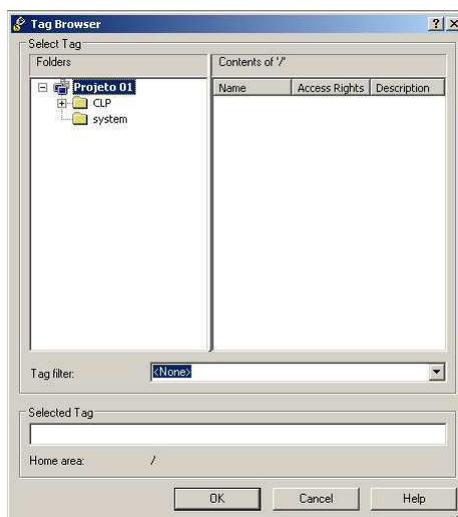


Figura 4: Tag Browser.

### 2.2 Telas

Em **Graphics > Displays** você irá configurar as telas para sua aplicação.

A tela principal (**MAIN**) já se encontra criada. Para criar outras, clique com o botão direito em **Displays** e depois em **New**. Uma nova tela irá aparecer para edição, esta deverá ter a resolução que configuramos no item 1.4. Ao final da edição, salve a mesma e dê um nome a ela.

Para o desenvolvimento das telas de sua aplicação, iremos agora explorar as ferramentas de desenho e entender as funcionalidades de alguns objetos.

#### 2.2.1 Objects

**Select**  - realiza a seleção de um ou mais itens na tela.



Figura 5: Objects

**Rotate**  - realiza a rotação do objeto selecionado (apenas um objeto, para realizar a rotação de 2 ou mais objetos é necessário agrupá-los, ver item 2.2.2). Selecione o objeto, clique em Rotate depois com o mouse escolha a posição desejada para o item.

**Text**  - Selecione com o mouse a área do texto na tela, logo após deverá aparecer uma janela de propriedades. No campo **Text**, digite o texto desejado. Também pode-se escolher a fonte, cor, tamanho, alinhamento e outras propriedades. Em **Insert Variable...** pode-se incluir variáveis no texto, para isso selecione o tipo da variável (Numeric ou String), clique em ... e selecione a variável desejada.

**Image**  - Selecione a área onde deseja colocar a imagem, clique em **Add from file**, escolha a imagem e clique em **Ok**. Ajuste o tamanho da imagem na tela.

**Panel**  - Desenha um painel na tela.

Os itens  são bastante intuitivos, cada um referente a uma forma geométrica que pode-se desenhar na tela.

Abaixo serão apresentados os botões (Push Buttons). As configurações dos botões se dá pela janela de propriedades dos mesmos.

Para todos os botões, algumas configurações são comuns entre eles: (**Botão direito > Properties**)

Na aba **General**, temos as configurações de aparência, como largura, formato e cor das bordas, formato do botão.

Na aba **States**, temos as configurações dos estados do botão. Aqui, podemos inserir/editar o texto do botão, incluir valor de variáveis, cor do fundo, modificando as configurações para cada estado diferente do botão. Em **Value** escolhemos um valor para cada estado do botão.

Na aba **Common**, podemos editar o tamanho e a posição do botão e dar um nome a ele, para melhor organização dos objetos.

Na aba **Connections**, no campo **Value** podemos editar o valor que o botão irá tomar como estado inicial. Caso não haja nenhum valor, o botão irá para o estado de Error. Aqui podemos também inserir uma Tag para ser modificada conforme o valor do botão.

**Momentary Push Button**  - Este botão quando mantido pressionado modifica seu estado, e quando não pressionado, mantém seu estado inicial. Na aba **General** podemos modificar o botão para ser normalmente fechado, normalmente aberto ou um valor determinado.

**Maintained Push Button**  - Este botão alterna os estados da seguinte forma: Quando pressionado no estado 0, seu estado muda para 1 na borda de subida; quando pressionado no estado 1, seu estado muda para 0 na borda de descida. Para melhor entendimento veja a figura 6.

**Latched Push Button**  - Este botão trava em um valor e só é resetado externamente. Na aba **Connections**, no campo Handshake deve ser colocada a Tag responsável

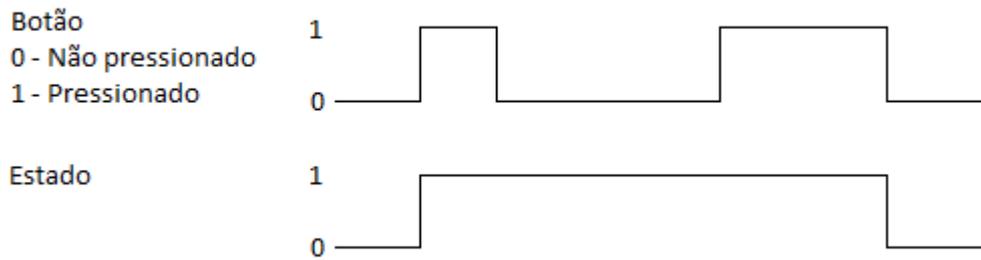


Figura 6: Mudança de estado do Maintained Push Button

pelo reset do botão. Na aba **General**, em **Button settings** podemos configurar se o reset será feito na borda de subida da Tag ou quando o valor da Tag é diferente de zero. Neste ultimo caso, se a Tag responsável pelo reset estiver diferente de zero, o Latched Button fica travado no estado 0.

**Multistate Push Button**  - Este botão pode possuir quantos estados forem necessários para ele, possuindo as mesmas configurações comuns aos outros botões para cada estado. Na aba **General** pode-se escolher o numero de estados, ou na aba **States** inserir ou deletar os estados.

**Interlocked Push Button**  - Este botão de intertravamento ao ser pressionado mudando seu estado de 0 para 1, ele segura o valor em 1 ou 0 (Valor definido na aba **General** no item **Button Value**). Caso este valor esteja salvo em uma Tag, ela ficará travada naquele valor (definido em Button Value) e só retornará ao valor anterior caso algum outro botão ou objeto altere esta Tag.

**Ramp Button**  - Este botão tem como função incrementar ou decrementar variáveis. Na aba **Connections** insira em **Value** a Tag a ser modificada pelo botão. Na aba **General** podemos escolher se o botão irá incrementar (**Ramp Up**) ou decrementar (**Ramp Down**) a Tag, bem como inserir os limites (**Up/Down limit**) e o valor do incremento (**Ramp value**). Para utilizar uma outra tag como limite ou valor do incremento, podemos inserir a Tag na aba **Connections** e selecionar na aba **General** que irá utiliza-las, como na figura 7.



Figura 7: Configurações do **Ramp Button** na aba **General**

**Numeric Display**  - Este objeto mostra o valor de alguma Tag. Na aba **Connections** podemos incluir a Tag e a polaridade para exibição, podendo ser positiva (0) ou negativa (1).

**Numeric Input Enable**  - Este objeto funciona como botão para entrada de valores. Ao clicar no objeto pode-se inserir um valor para salva-lo em uma Tag. Na aba **Numeric** podemos editar as configurações de entrada do valor. Em Numeric pop-up, pode-se escolher entre Keypad (Aparece para digitar o valor clicando na tela), Scratchpad (digita-se o valor por um teclado conectado na HMI ou pelo teclado da mesma) ou None.

**Numeric Input Cursor Point**  - Este botão é bastante similar ao Numeric Input Enable, mas na aba **Numeric** podemos adicionar uma Tag e/ou texto que irá aparecer

na hora de digitar. Um exemplo pode ser visto na figura 8. (Esta funcionalidade não é compatível com algumas versões)

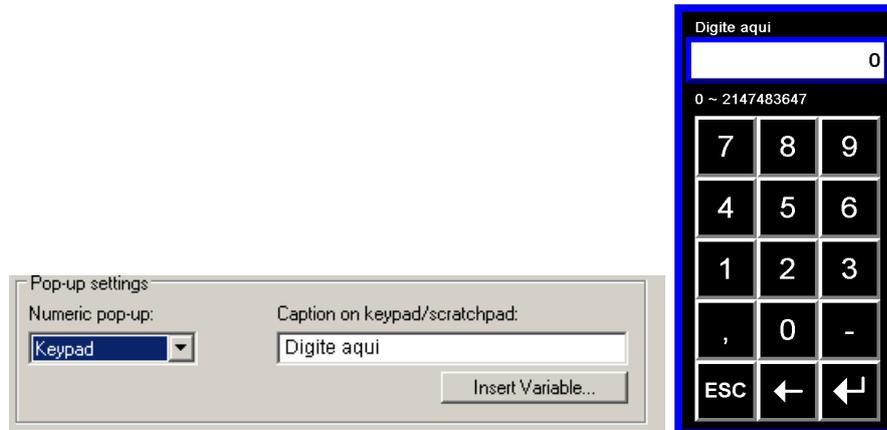


Figura 8: Utilizando o Numeric Input Cursor Point

**String Display**  - Este objeto funciona como Display para tags do tipo String.

**String Input Enable**  - Este objeto funciona da mesma forma que o Numeric Input Enable, porém para strings.

**Goto Display Button**  - Este objeto direciona para outro display (Ver como criar um novo display no item 2.2). Na aba **General** pode-se configurar o Display destino.

**Return to Display Button**  - Botão para retornar ao display anterior.

**Close Display Button**  - Fecha o display.

**Display List Selector**  - Este objeto faz uma lista de displays para seleção. Na aba **States** pode-se adicionar ou remover os estados, cada qual referente a um display. Em Caption dê um nome para aparecer na lista, podendo utilizar o nome do próprio display.

**Multistate Indicator**  - Este objeto indica o estado de uma Tag, podendo indicar vários estados para os diferentes valores a que Tag assumir.

**Symbol**  - Aqui pode-se adicionar um símbolo, que poderá mudar a cor conforme a mudança de estado de uma Tag.

**List Indicator**  - Indica o estado de uma tag, em forma de lista. Exibe a lista de com todos os estados selecionando o estado em que a Tag se encontra atualmente.

**Bar Graph**  - Este objeto funciona como um gráfico para mostrar o nível de uma Tag. Seus valores máximo e mínimo podem ser configurados na aba **General**. Ver figura 9.

**Gauge**  - Este objeto é um indicador para o valor de uma Tag, podendo ser editado seus valores máximo e mínimo na aba **Display**.

**Scale**  - Este objeto é uma escala. Podendo editar apenas quantidade de divisões e subdivisões. Utilize o objeto Text para indicar os valores de cada divisão, como pode ser visto na figura 9.

**Trend**  - Este objeto tem como função monitorar graficamente os valores de uma ou mais tags ao longo do tempo. Na aba **Pens** pode-se configurar as 'canetas' que irão riscar os valores das tags no gráfico. Na aba **Connections** pode-se associar uma tag a cada

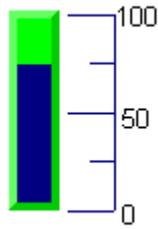


Figura 9: Exemplo mostrando o **Bar Graph** com o **Scale**

caneta. Em **X-Axis** pode-se configurar o tempo em que será mostrado no gráfico. Em **Y-Axis** pode-se configurar os valores do eixo Y, podendo em Custom colocar um valor máximo e mínimo pre-definido ou controlado por alguma tag. Para controlar o mínimo e máximo através de uma ou duas Tags, elas devem ser colocadas na aba **Connections**. Em **Display** pode-se editar as configurações de exibição do gráfico.

**Pause Trend Button** - Este objeto pausa o monitoramento.

**Next Pen Button** - O valor do eixo Y do gráfico é referente a uma Tag, para mudar a tag de referencia, utiliza-se este botão.

**Control List Selector** - ????????

**Piloted Control List Selector** - ????????

**Local Message Display** - Este objeto mostra uma mensagem de acordo com o valor assumido por uma Tag. Na aba **Connections** adicione a tag. Na aba **General** escolha o arquivo de mensagem. Para adicionar um arquivo de mensagem, vá no explorer da sua aplicação, em **Graphics > Local Messages**, clique com o botão direito, depois New. Agora configure a mensagem referente a cada valor da tag como podemos ver na figura 10

**Login Button** - Botão para efetuar login de um usuário.

**Logout Button** - Botão de logout.

**Password Button** - Botão para alterar o password do usuário logado.

**Macro Button** - Este botão executa uma macro. Configure as macros em **Logic and Control > Macros**.

**Display Print Button** - Captura imagem da tela e salva **ONDE??**

**Shutdown Button** - Sai da aplicação para o menu da HMI.

**Time and Date Display** - Exibe a data e a hora.

**Active X Control** -

**RecipePlus Button** -

**RecipePlus Selector** -

**RecipePlus Table** -

**Language Switch Button** -

**Symbol Factory** -

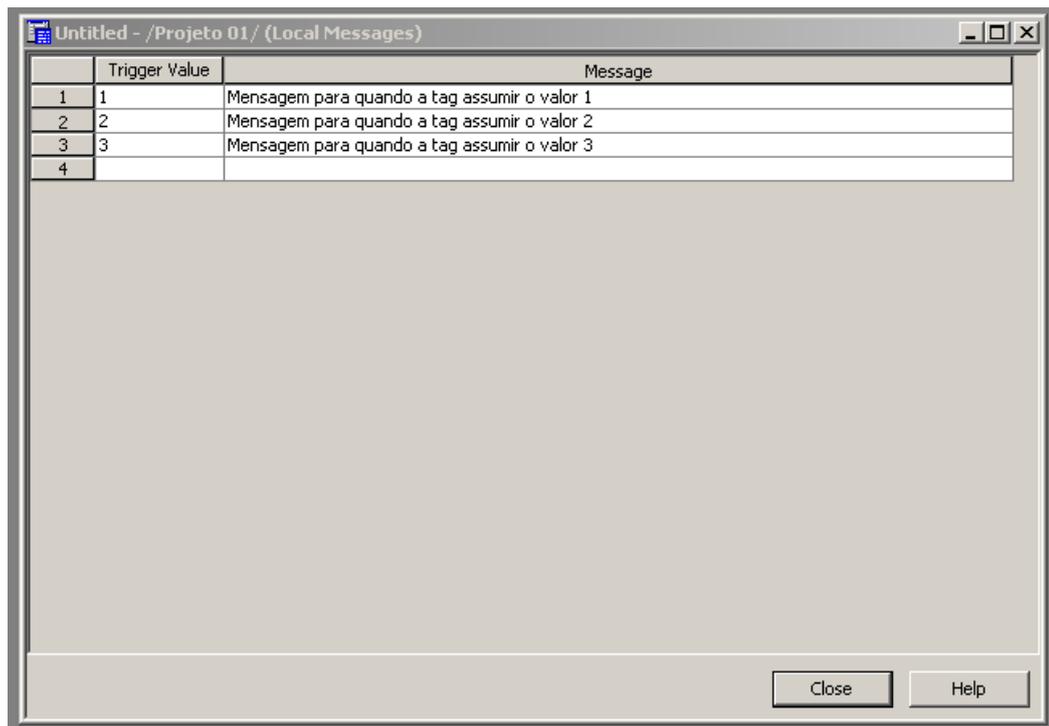


Figura 10: Configurando Mensagens

### 2.2.2 Graphics



Figura 11: Alinhamento dos objetos

Aqui podemos copiar, colar, duplicar, deixar os objeto igualmente espaçados na vertical ou horizontal (ver figura 12), espelhar, agrupar ou desagrupar, colocar um objeto na frente ou atrás de outro, dar zoom, desfazer e refazer uma ação, mostrar painel de propriedades e object explorer.

Com o painel de propriedades aberto, podemos rapidamente alterar as configurações dos objetos.

### 2.2.3 Alignment

Aqui podemos alinhar os objetos para melhor organização dos mesmos nas telas. Selecione dois ou mais objetos para serem alinhados e depois vá em alignment e selecione como se deseja alinha-los.

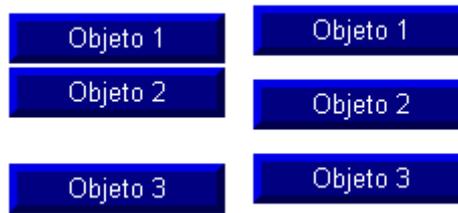


Figura 12: Espaçamento vertical dos objetos



Figura 13: Alinhamento dos objetos

#### 2.2.4 Test application

Durante o desenvolvimento das telas, elas podem ser testadas através da ferramenta Test application . Basta clicar neste botão que será aberta a aplicação a ser testada no próprio computador.

#### 2.2.5 Animações

Para cada objeto, é possível atribuir animações que acontecem de acordo com determinado evento. Por exemplo, pode-se atribuir a um botão para ele só ser mostrado dependendo do valor de uma Tag, ou para que um painel contendo alguma informação mude de cor.

Para configurar as animações, clique com o botão direito sobre o objeto, vá em **Animations** e escolha a configuração que deseja realizar. Nem todas as configurações de animação estão disponíveis para todos os objetos. Quando já existir um configuração de animação, ela ficará selecionada como pode-se observar na figura 15.

Ao selecionar a configuração a ser realizada, aparecerá uma tela, como na figura 16, onde se poderá inserir uma Tag e as configurações para os valores da Tag. Por exemplo na configuração da figura 16, o objeto irá aparecer nas cores preto e vermelho para o valor 0 da tag Dig, e nas cores preto e verde para o valor 1.

Como pode ser observado, nas outras abas pode-se realizar outras configurações de ani-



Figura 14: Alinhamento dos objetos

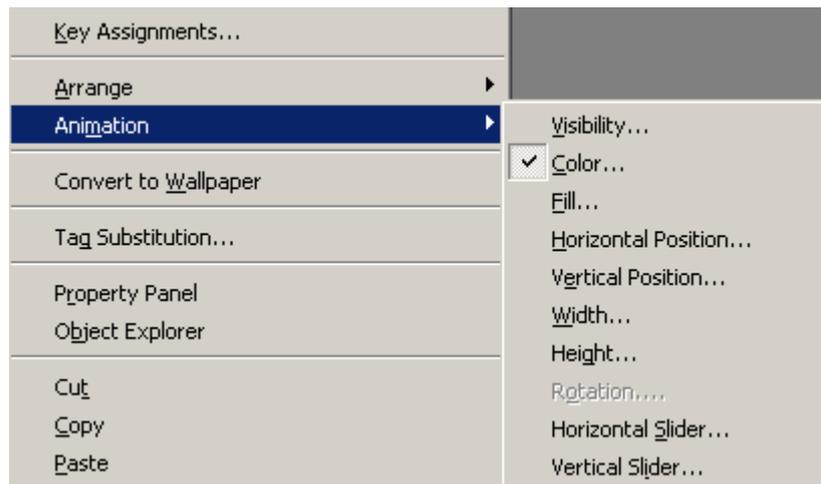


Figura 15: Configurações de animação.

mação como posição horizontal e vertical, visibilidade, deslizamento vertical e horizontal, dentre outras.

### 3 Configurações na HMI - PanelView Plus 400

Aqui veremos como realizar algumas configurações básicas no PanelView Plus.

Na primeira tela temos duas opções iniciais: **Load Application** e **Terminal Settings**. Selecione Terminal Settings para acessar as configurações.

#### 3.1 Configurações de rede

Em **Terminal Settings > Networks and Communications > Network Connections** podemos editar as configurações de rede da HMI.

**Device Name** - Nome do aparelho.

**Network Adaptors** - Aqui podemos alterar as configurações do IP (selecioneando **IP Address**, para utilizar um IP fixo desmarque a opção Use DHCP, caso esta esteja marcada.). Em **Name Servers** podemos também inserir os servidores DNS e WINS.

#### 3.2 Gerenciando arquivos

Em **Terminal Settings > File Management** podemos apagar ou copiar os arquivos da memória do PanelView.

#### 3.3 Verificando versão do arquivo

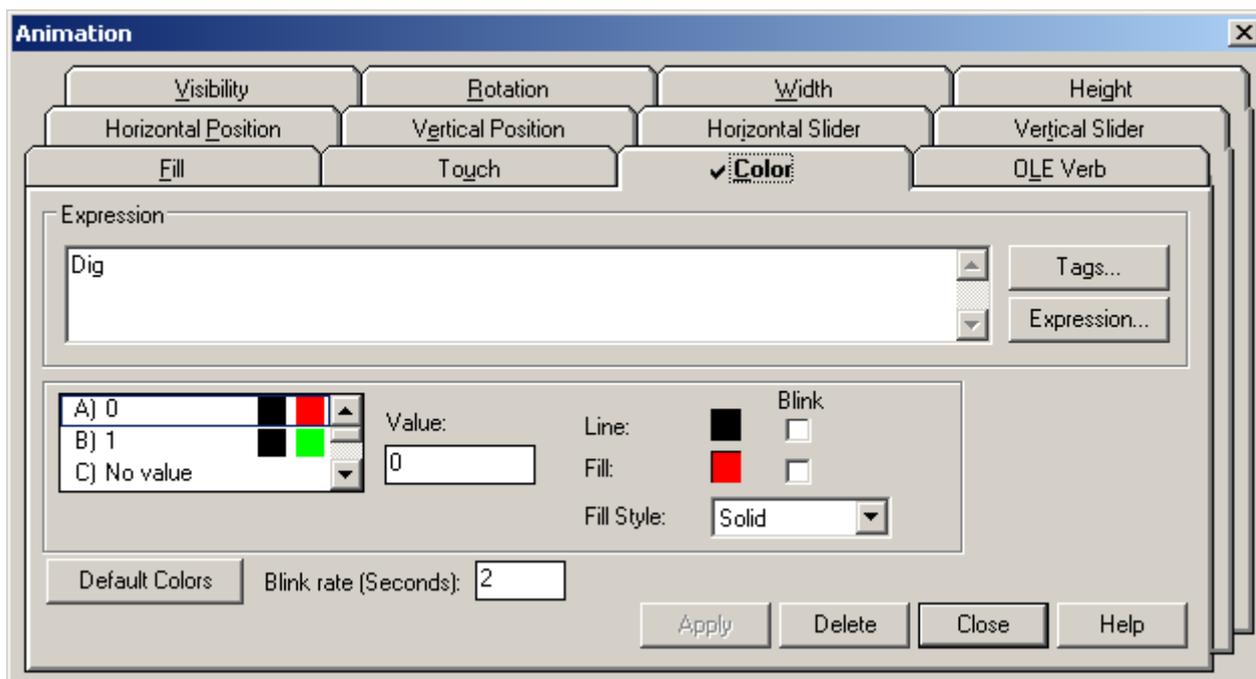


Figura 16: Configurando animações.

Em **Terminal Settings > System Information > About RSView ME Station** verifique a versão do Firmware.

## 4 Enviando a aplicação para a HMI

Primeiramente antes de enviar a aplicação, é necessário criar a aplicação no formato \*.mer. Para isso, vá em **Application > Create Runtime Application...** e salve o arquivo com a mesma versão da sua HMI (para saber a versão veja o item 3.3).

Abra o Transfer Utility  (**Tools > Transfer Utility**), na aba **Download**, em **Source file**, e selecione o arquivo \*.mer. Caso queira modificar o nome quando transferir para o PanelView, selecione **Download as** e coloque o nome, caso não esteja selecionado será transferido com o mesmo nome. Selecione o local destino (O armazenamento interno do PanelView ou algum dispositivo externo). Caso deseje que a aplicação abra sempre que ligar o PanelView, selecione a opção **Run application at start-up**, neste caso é recomendado que a aplicação tenha um botão shutdown.

Selecione o dispositivo ao qual deseja enviar a aplicação, como na Figura 17, e clique em **Download**.

Para transferir um arquivo da HMI para o computador, vá na aba **Upload**, selecione a pasta destino, escolha o arquivo e clique em **Upload**.

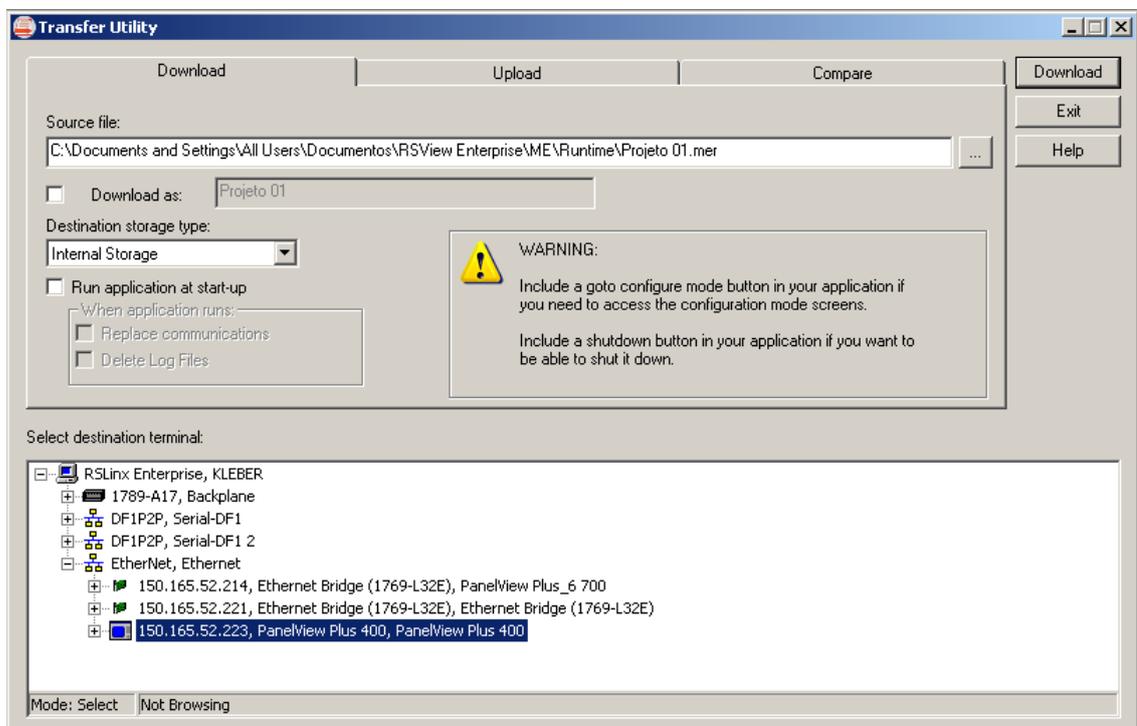


Figura 17: Transfer Utility