



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Mateus Coutinho Sarmiento

Relatório de Estágio Supervisionado

Campina Grande, Paraíba

Agosto de 2017

Mateus Coutinho Sarmiento

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.

Orientador: George Acioli Júnior

Campina Grande, Paraíba

Agosto de 2017

Mateus Coutinho Sarmiento

Relatório de Estágio Supervisionado/ Mateus Coutinho Sarmiento. – Campina Grande, Paraíba, Agosto de 2017-

33 p. : il. ; 30 cm.

Orientador: George Acioli Júnior

Relatório de Estágio Supervisionado – Universidade Federal de Campina Grande – UFCG, Agosto de 2017.

Mateus Coutinho Sarmento

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.

Aprovado em ____ / ____ / ____

Professor Avaliador

Universidade Federal de Campina Grande
Avaliador

George Acioli Júnior

Universidade Federal de Campina Grande
Orientador

Campina Grande, Paraíba
Agosto de 2017

Aos meus pais, minha eterna gratidão.

Agradecimentos

Aos meus pais, André e Luciane, por serem meu porto seguro e sempre estarem me apoiando nas minhas decisões.

Aos meus irmãos, Andréa e Davi, pelo carinho e companheirismo.

Aos meus familiares e amigos, por se fazerem presentes apesar da distância.

Aos amigos e colegas de curso, companheiros de algumas noites em claro, de vários finais de semana de estudo e de incontáveis dias de aulas e provas, os quais espero que possam continuar presentes em minha vida.

Ao pessoal do LIEC, pelas boas horas de companhia, bom humor, ajuda mútua e companheirismo.

A todos que de alguma forma, direta ou indiretamente, contribuíram para a minha formação.

Enfim, agradeço ao corpo docente, à direção e à administração do DEE por seu trabalho na formação dos alunos de Engenharia Elétrica.

“I was originally supposed to become an engineer but the thought of having to expend my creative energy on things that make practical everyday life even more refined, with a loathsome capital gain as the goal, was unbearable to me.”

(Albert Einstein)

Resumo

Este relatório apresenta o trabalho realizado pelo aluno Mateus Coutinho Sarmento 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), durante 16 de Maio de 2017 a 10 de agosto de 2017 sob orientação do Professor George Acioli Júnior e supervisão do Professor Rafael Bezerra Correia Lima. O foco do estágio foi na manutenção e aprimoramento da sala de automação do Laboratório de Instrumentação Eletrônica e Controle.

Palavras-chave: Estágio Supervisionado; CLP; Comunicação OPC; Planta Didática; Interface Homem Máquina.

Lista de ilustrações

Figura 1 – Faixada do prédio do LIEC	13
Figura 2 – Sala de Automação do LIEC	14
Figura 3 – Esquema de comunicação SepServer II	16
Figura 4 – Ícone do modulo SS2S	17
Figura 5 – Tela principal do SS2M	17
Figura 6 – Criando driver ethernet	18
Figura 7 – Parâmetros do canal ethernet	18
Figura 8 – Configurando o atalho do SS2S	19
Figura 9 – Configurando o SOS	19
Figura 10 – GUI do OPC ToolBox	20
Figura 11 – Adicionando uma variável a lista de itens	21
Figura 12 – Peltier G2-1	22
Figura 13 – Peltier G1-1	22
Figura 14 – Teste de dinâmica realizado no simulink	23
Figura 15 – Teste de dinâmica do Peltier G2-1	23
Figura 16 – Teste de dinâmica do Peltier G2-2	24
Figura 17 – Tela inicial do VMware	25
Figura 18 – Plataforma didática	27
Figura 19 – Numeração das chaves e LEDs	28
Figura 20 – Processo Peltier G2-3	30

Lista de tabelas

Tabela 1 – Entradas digitais	29
Tabela 2 – Saídas digitais	29
Tabela 3 – Entradas e saídas analógicas	29

Lista de abreviaturas e siglas

PC	Computador Pessoal
OPC	Open Platform Communications
SNR	Signal-noise ratio
UFCG	Universidade Federal de Campina Grande
LIEC	Laboratório de Instrumentação Eletrônica e Controle
DEE	Departamento de Engenharia Elétrica
CLP	Controlador Logico Programável
IHM	Interface Homem Máquina
GUI	Graphical User Interface

Sumário

1	INTRODUÇÃO	12
2	LOCAL DO ESTÁGIO SUPERVISIONADO	13
2.1	Sala de Automação	13
3	ATIVIDADES REALIZADAS	15
3.1	Configuração de Computadores e CLPs da HI Tecnologia	15
3.1.1	Instalação do servidor OPC	15
3.1.1.1	ScpServer II	15
3.1.1.2	SS2S	16
3.1.1.3	SS2M	17
3.1.1.4	SOS	19
3.1.2	OPC ToolBox do MatLab	20
3.1.2.1	GUI do OPC ToolBox	20
3.2	Instalação dos Processos Peltiers	21
3.3	Instalação de Máquinas Virtuais	24
3.4	Desenvolvimento de Guias Experimentais	25
3.4.1	Guia 0 - Guia Rockwell	25
3.4.2	Guia 1 - Linguagem Ladder	26
3.4.3	Guia 2 - Linguagem Ladder 2	26
3.5	Desenvolvimento de uma Plataforma Didática	26
3.5.1	CLP Compact Logix	27
3.5.2	Entradas e Saídas Digitais	28
3.5.3	Entradas e Saídas Analógicas	29
4	CONCLUSÃO	31
	REFERÊNCIAS	32
	ANEXO A – GUIAS EXPERIMENTAIS	33

1 Introdução

Neste relatório são apresentadas as atividades desenvolvidas no Estágio Supervisionado realizado por Mateus Coutinho Sarmiento no Laboratório de Instrumentação Eletrônica e Controle (LIEC), pertencente ao Departamento de Engenharia Elétrica (DEE) da Universidade Federal de Campina Grande (UFCG), com carga horaria de 210 horas.

Dentre as atividades realizadas durante o estágio estão inclusas a manutenção das máquinas, a instalação de softwares, a instalação e configuração da comunicação OPC, o desenvolvimento de guias de laboratório para a disciplina de Sistemas de Automação Industrial e o desenvolvimento de uma plataforma didática composta por um CLP, uma IHM e um Peltier.

Este relatório é dividido em quatro capítulos. No primeiro é apresentada uma introdução. O segundo capítulo apresenta o local aonde foi realizado o estagio. No terceiro capítulo são descritas as atividades desenvolvidas. Por fim, no quarto capítulo é feito uma análise geral do estágio e dos resultados alcançados.

2 Local do Estágio Supervisionado

O Laboratório de Instrumentação Eletrônica e Controle é um laboratório do Departamento de Engenharia Elétrica cujos integrantes trabalham nas áreas de Controle, Automação e Eletrônica. Os projetos de pesquisa contam com a participação dos alunos do curso e do corpo docente. O LIEC está situado no Setor C da Universidade Federal de Campina Grande em Campina Grande - Paraíba. Possui uma área de aproximadamente 900 m^2 com oito Laboratórios de desenvolvimento, duas salas de apoio técnico, uma moderna sala de apresentação de trabalhos, salas para Pós-graduação e professores (LIEC... , 2017). A fachada do prédio é ilustrada na Figura 1.



Figura 1 – Faixada do prédio do LIEC
Fonte: Adaptado de (LIEC... , 2017)

2.1 Sala de Automação

A sala de automação do LIEC é a sala de numero 201 e fica localizada no segundo andar do prédio. Nela se encontra nove computadores, nove CLPs e seis plantas didáticas, dentre as quais quatro são processos peltier e dois são sopradores de calor.

Nesta sala são desenvolvidas atividades de ensino e pesquisa por alunos de graduação e pós-graduação. Durante a graduação, os alunos utilizam este laboratório na disciplina de Sistemas de Automação Industrial para as aplicações práticas utilizando CLP.

As atividades do estágio foram focadas na manutenção e aprimoramento desta sala. A Figura 2 ilustra a sala de automação.



Figura 2 – Sala de Automação do LIEC
Fonte: Próprio autor

3 Atividades Realizadas

Neste capítulo serão apresentadas as principais atividades realizadas pelo estagiário no LIEC. Todas as atividades foram acompanhadas por professores ou técnicos do laboratório.

3.1 Configuração de Computadores e CLPs da HI Tecnologia

Como primeira atividade foi designado a instalação dos softwares básicos nos computadores. Dentre eles pode-se citar:

- MatLab
- SPDSW
- Servidor OPC HI Tecnologia

Foi necessário também configurar os IPs dos CLPs corretamente para que se adequassem a faixa estabelecida para a sala de automação e para que cada computador se comunique especificamente com o CLP ao seu lado. Para realizar essa configuração foi utilizado o software SPDSW.

Com a instalação do servidor OPC foi também necessário fazer a sua configuração para cada máquina com relação ao seu CLP.

3.1.1 Instalação do servidor OPC

Para a configuração do servidor OPC é necessário os seguintes softwares: SS2S, SS2M e SOS, fornecidos pela HI Tecnologia, presentes no pacote ScpServer II. As instruções para a instalação do servidor OPC foram obtidas através do manual disponibilizado pela própria HI Tecnologia em (HI. . . , 2017).

3.1.1.1 ScpServer II

O servidor de comunicação ScpServer II permite que múltiplos clientes possam se conectar com múltiplos equipamentos da HI tecnologia. É possível fazer a conexão através do canal serial ou via cabo ethernet. Neste estudo o segundo método foi escolhido.

A comunicação ocorre da seguinte maneira, os equipamentos se comunicam com o SS2S, servidor de comunicação para o protocolo SCP-HI, que por sua vez ira se comunicar

com o SOS, módulo servidor OPC, sendo este o servidor em que todos os clientes OPC irão se conectar. A Figura 3 ilustra este esquema de comunicação.

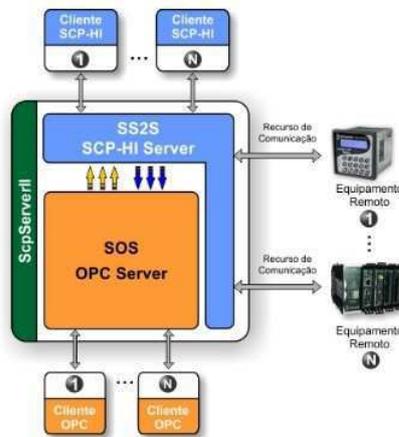


Figura 3 – Esquema de comunicação ScpServer II

Fonte: Adaptado de (HI. . . , 2017)

3.1.1.2 SS2S

O módulo SS2S é um servidor de comunicação para o protocolo SCP-HI. Ele pode ser executado como uma aplicação iconizada ou como um serviço do Windows, durante este estudo a primeira opção foi utilizada. É neste módulo onde ficará salva as configurações e os endereços dos equipamentos a serem utilizados. Porém, devido ao fato de que esse modulo roda apenas como uma aplicação iconizada será necessário fazer o uso de um outro modulo para configurá-lo, este será o papel do modulo SS2M.

Ao executar o modulo SS2S um ícone ira aparecer na barra de tarefas do Windows, este ícone contém diversas informações sobre o estado do servidor que podem ser melhor explicadas pela Figura 4.

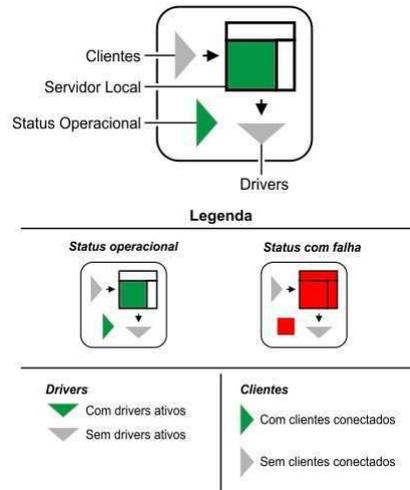


Figura 4 – Ícone do módulo SS2S
Fonte: Adaptado de (HI..., 2017)

Aonde os drivers são responsáveis por implementar os mecanismos de entrada e saída dos equipamentos e também gerenciar as tags. Os clientes são qualquer aplicação que se utilizada do ScpServerII. Enquanto que o servidor local fica responsável por aceitar conexões de clientes e fazer a ligação entre os mesmos e os drivers disponíveis.

3.1.1.3 SS2M

O SS2M é um aplicativo utilizado para criar o arquivo de configuração do servidor SS2S, bem como monitorar o funcionamento do servidor SS2S. Ao se abrir este aplicativo será possível ver a sua tela principal, Figura 5.

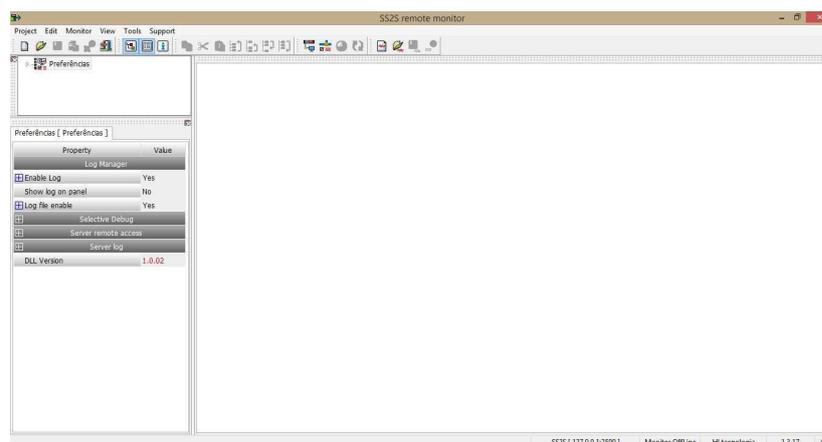


Figura 5 – Tela principal do SS2M
Fonte: Próprio autor

É necessário primeiro criar um novo projeto, clicando em **Projects** -> **New**. Após se escolher um nome, devemos agora adicionar um driver a este projeto.

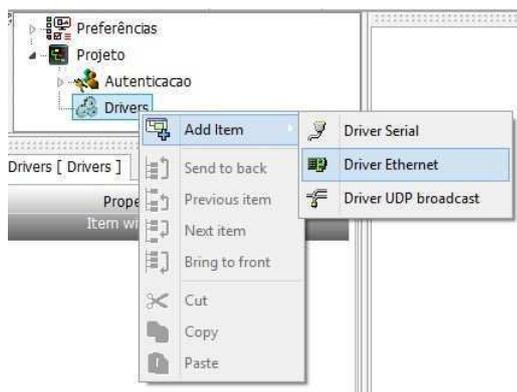


Figura 6 – Criando driver ethernet
Fonte: Próprio autor

O driver utilizado no laboratório é o driver ethernet, visto que já existe uma conexão deste tipo sendo utilizada pelo CLP.

Com o driver criado, por *default* seu nome será Driver_ 1000, devemos agora configurar o canal. É no canal que estarão salvas as configurações de endereço do equipamento que vamos utilizar, na Figura 7 está todos os parâmetros que devemos configurar.

Property	Value
Channel name	:hernet_001
Target Port	2016
Device IP	192.168.0.2
Target Gateway IP	192.168.0.1
Target Subnet mask	255.255.255
Equipments Mapping	
Initial ID	255
Final ID	255
Enable Remap ID	No

Figura 7 – Parâmetros do canal ethernet
Fonte: Próprio autor

Os parâmetros Target Port, Device IP, Target Gateway IP e Target Subnet mask são obtidos pelas configurações do CLP, que podem ser visualizadas através do software SPDSW. Os parâmetros Initial ID e Final ID se referem ao mapeamento dos equipamentos, como no nosso caso existe apenas um não é necessário alterar. É importante notar que este é o endereço que iremos utilizar para acessar este canal.

É necessário agora informar ao SS2S qual o arquivo contém as configurações que desejamos para o nosso servidor, isto é feito via link.

Para passarmos um parâmetro via link é necessário alterar o atalho do SS2S, criado na área de trabalho no momento da instalação. No campo "Destino" devemos adicionar o seguinte parâmetro “-o projects\NOME DO PROJETO.s2p”

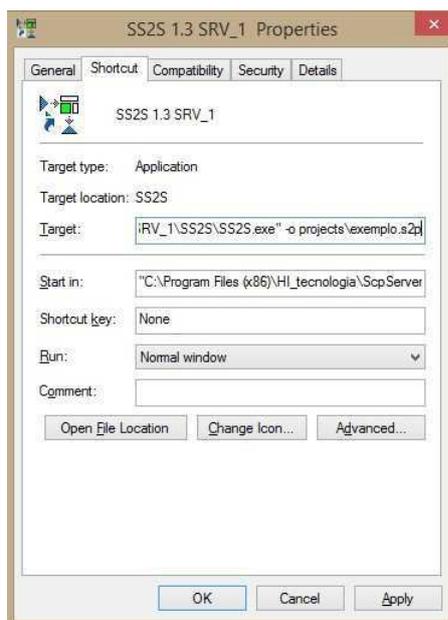


Figura 8 – Configurando o atalho do SS2S
Fonte: Próprio autor

Isto indica ao SS2S que ele deve executar as configurações salvas no arquivo exemplo.s2p que está na pasta projects presente no diretório de instalação do próprio SS2S.

3.1.1.4 SOS

O SOS é uma aplicação que cria uma interface servidora compatível com o protocolo OPC DA 2.0. Ele se comunica com o SS2S para acessar os equipamentos, assim permitindo que os clientes OPCs tenham acesso aos equipamentos conectados ao SS2S.

Para configurar o SOS é preciso clicar com o botão direito em seu ícone e selecionar Open Interface. A janela da Figura 9 irá ser aberta.

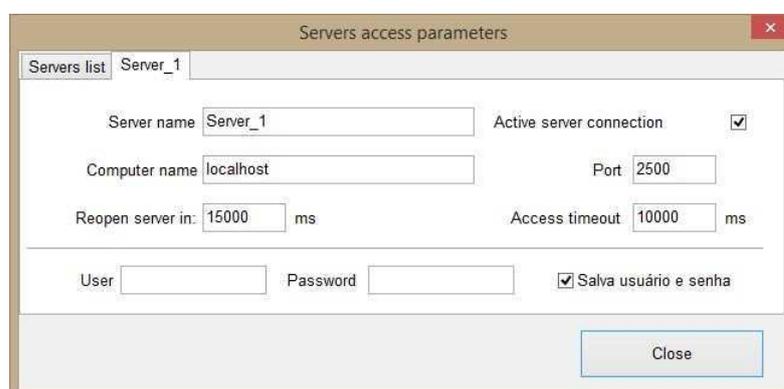


Figura 9 – Configurando o SOS
Fonte: Próprio autor

Na Figura 9, Server name é o nome do servidor a ser acessado, Computer name é o nome do computador na rede ou endereço de IP e Port é a porta de acesso ao servidor a ser conectado.

Como o servidor está na própria máquina aonde o SS2S está sendo executado não é necessário alterar nenhum parâmetro.

3.1.2 OPC ToolBox do MatLab

Com o servidor OPC e o Matlab instalados é possível utilizar o OPC ToolBox para monitorar as variáveis do CLP. O manual para este ToolBox se encontra em (OPC..., 2017).

3.1.2.1 GUI do OPC ToolBox

A GUI pode ser acessada por meio do comando `opctool` que irá abrir a tela na Figura 10.

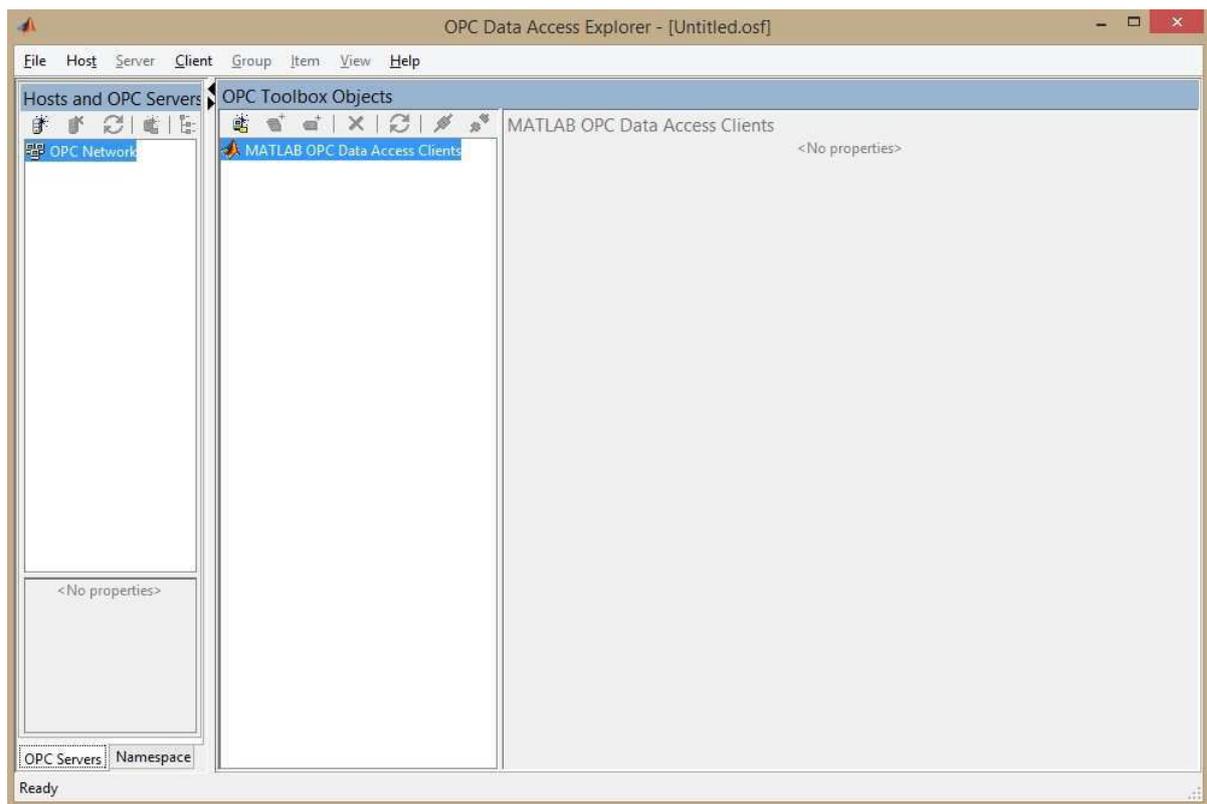


Figura 10 – GUI do OPC ToolBox
Fonte: Próprio autor

Nesta tela é possível se conectar ao servidor OPC e acessar diretamente as variáveis desejadas.

Após conectado ao servidor caso se deseje escrever ou ler uma certa variável é necessário primeiro adicioná-la a nossa lista de itens. Isto é feito por um comando que possui o formato `Server_ N.Driver_ XXXX.DEVYYY.VARIAVEL`.

A Figura 11 ilustra um exemplo caso se deseje acessar a variável R0 que está no equipamento de endereço 255, no driver de ID 1000 e servidor 1

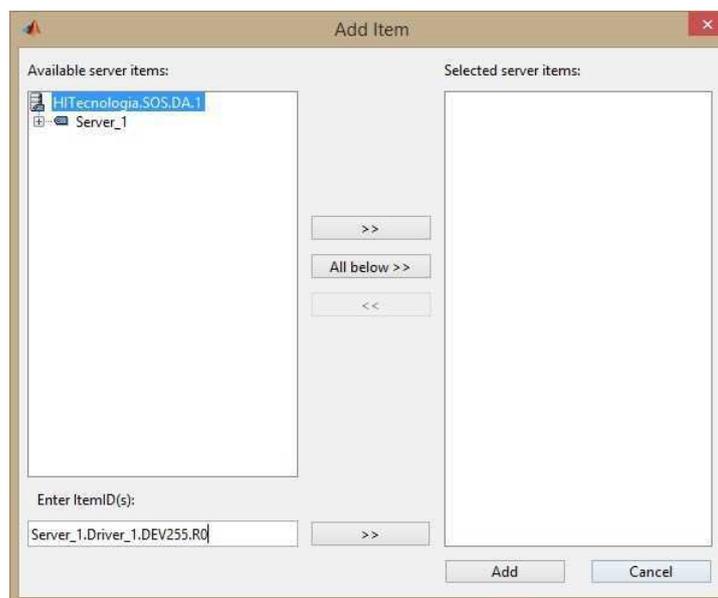


Figura 11 – Adicionando uma variável a lista de itens
Fonte: Próprio autor

Após adicionada a variável será possível ler e escrever seus valores.

3.2 Instalação dos Processos Peltiers

Para esta atividade foi pedido que se instalasse um processo Peltier para cada uma das bancadas. Estes processos iriam posteriormente ser utilizados por alunos da disciplina Sistemas de Automação Industrial durante a parte prática da disciplina. Sendo assim foram instalados cinco Peltiers, dois sendo de versões mais antigas e três novos. Os Peltier das versões antigas foram nomeados como Peltier G1-1 e Peliter G1-2. Já os das versões mais novas foram nomeados Peltier G2-1, G2-2 e G2-3. Os Peltiers das versões mais novas são diferenciados entre si pelo modo como foram instaladas suas massas metálicas, que é o meio pelo qual ocorre a transferência de calor. As Figuras 12 e 13 ilustram dois dos modelos instalados. Para realizar a instalação foi necessário primeiro estudar como funciona suas ligações, para que então fosse possível ligar estes processos nos CLPS.



Figura 12 – Peltier G2-1
Fonte: Próprio autor

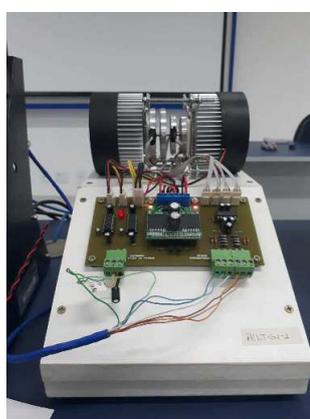


Figura 13 – Peltier G1-1
Fonte: Próprio autor

Foi então criado um código na linguagem Ladder para que fosse possível realizar a leitura e escrita nas saídas e entradas do processo. Feito isto, foi então possível realizar testes para observar a dinâmica destes processos. O teste foi realizado através do simulink, fazendo uso da comunicação OPC. A Figura 14 ilustra o programa do simulink utilizado para realizar o teste. As Figuras 15 e 16 ilustram alguns dos testes realizados.

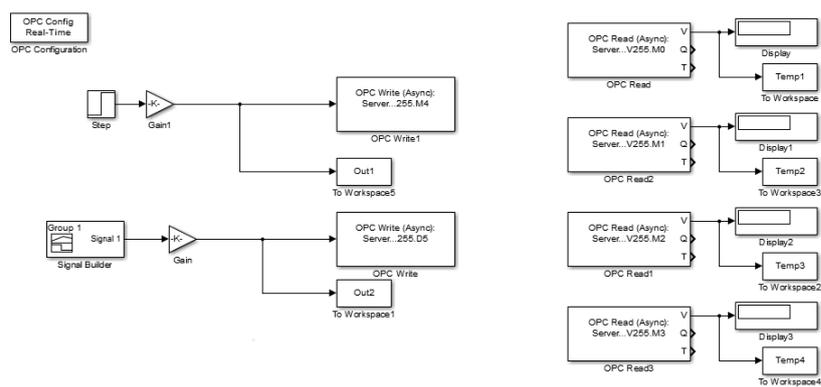


Figura 14 – Teste de dinâmica realizado no simulink
 Fonte: Próprio autor

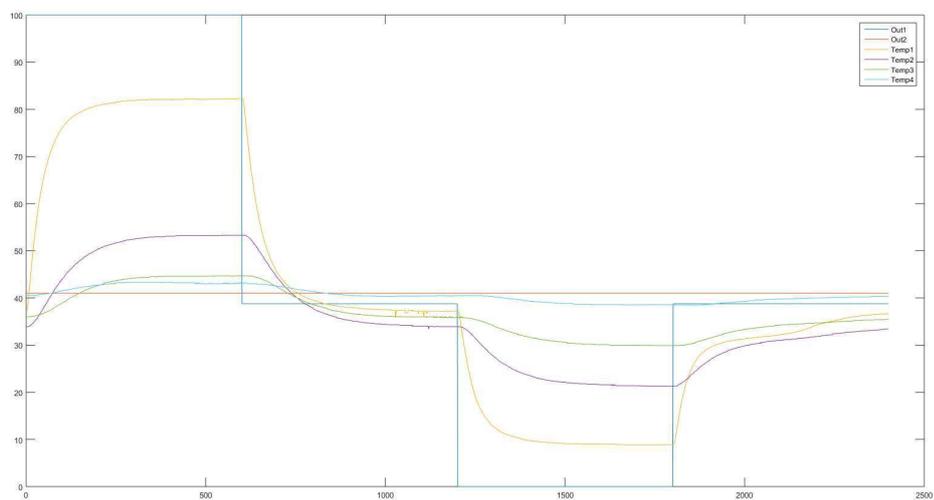


Figura 15 – Teste de dinâmica do Peltier G2-1
 Fonte: Próprio autor

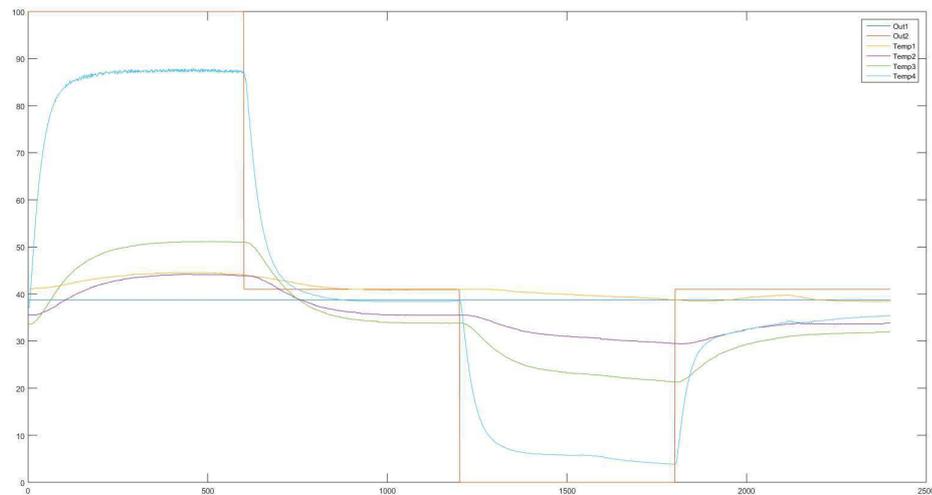


Figura 16 – Teste de dinâmica do Peltier G2-2
Fonte: Próprio autor

3.3 Instalação de Máquinas Virtuais

O segundo tipo de CLP utilizado na sala é o CLP da CompactLogix 1769 da Rockwell, os softwares necessários para a configuração e programação destes CLPs estão instalados em uma máquina virtual, que por sua vez, esta salva em um HD externo. Então, para que seja possível fazer uso de tais softwares foi necessário primeiro instalar a máquina virtual nos computadores.

O software escolhido para rodar a máquina virtual foi o *VMware Workstation 12 Player*.

O procedimento para a instalação da máquina virtual foi o seguinte:

1. Copiar os arquivos do HD externo para o computador;
2. Instalar o software VMware;
3. Abrir os arquivos da máquina virtual no VMware;
4. Clicar em *Play Virtual Machine*

Estas máquinas virtuais foram instaladas apenas nas máquinas mais novas, devido a capacidade de processamento limitada das máquinas antigas.

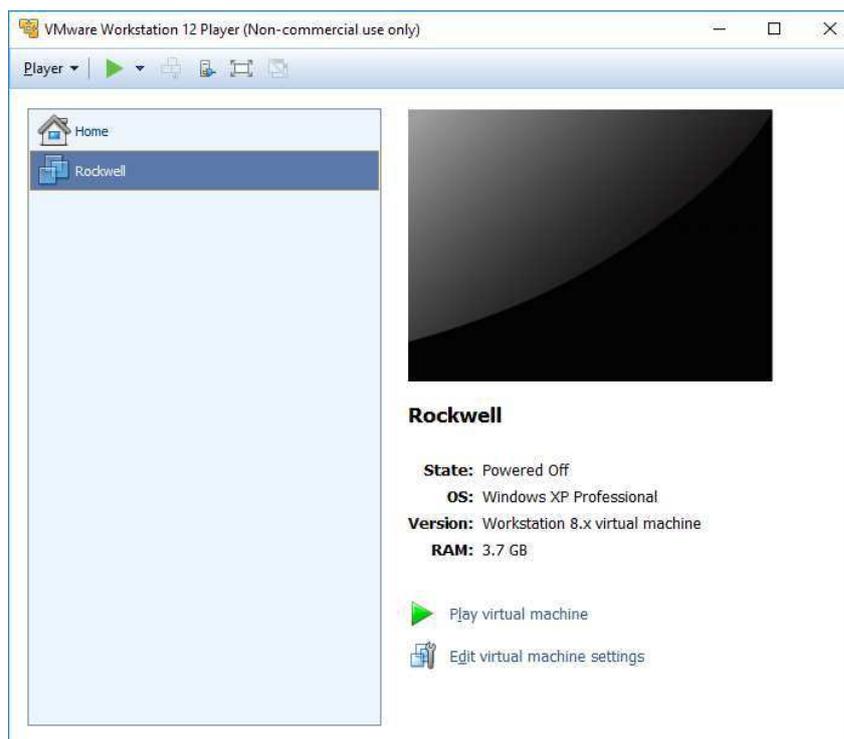


Figura 17 – Tela inicial do VMware
Fonte: Próprio autor

3.4 Desenvolvimento de Guias Experimentais

Com o objetivo de ampliar o conhecimento dos alunos da disciplina de Sistemas de Automação Industrial com relação aos CLPs disponíveis no mercado foi pedido para que se desenvolvesse guias experimentais para introduzir o CLP da Rockwell na disciplina. Para isto foi realizado um estudo sobre o CLP a partir dos manuais (AUTOMATION, 2013) e (AUTOMATION, 2015). Foi então desenvolvido três guias, sendo eles:

- Guia 0 - Guia Rockwell
- Guia 1 - Linguagem Ladder
- Guia 2 - Linguagem Ladder 2

Os objetivos de cada guia foram os seguintes:

3.4.1 Guia 0 - Guia Rockwell

Este guia tem como objetivo familiarizar o aluno com a plataforma didática disponível no laboratório de automação do LIEC. Incluindo como instalar a máquina virtual, como criar um projeto no software RSLogix5000 e como funciona a plataforma didática.

3.4.2 Guia 1 - Linguagem Ladder

Este guia tem como o objetivo familiarizar o aluno com a programação em Ladder utilizada pelo software RSLogix 5000 para o CLP Compact Logix do fabricante Allen-Bradley. Durante o guia serão apresentados blocos básicos, o aluno deverá estudar o funcionamento destes blocos para que durante a aula execute as atividades apresentadas ao final do guia.

Neste guia foram abordados os seguintes tópicos:

- Como Programar em Ladder
- Instruções Básicas
- Blocos Bit
- Blocos Temporizadores e Contadores
- Blocos de Comparação
- Blocos de Operações Matemáticas
- Blocos Move/Logical

3.4.3 Guia 2 - Linguagem Ladder 2

Este guia tem como o objetivo de aprofundar o conhecimento do aluno com a programação em Ladder utilizada pelo software RSLogix 5000 para o CLP Compact Logix do fabricante Allen-Bradley. Durante o guia o aluno será confrontado com diferentes problemas reais e deverá escrever uma lógica Ladder que solucione tais problemas.

3.5 Desenvolvimento de uma Plataforma Didática

Como última atividade do estágio foi pedido para auxiliar o aluno de Doutorado Moisés Tavares da Silva no desenvolvimento de uma plataforma didática que seria utilizada posteriormente pelos alunos para interagir com o CLP da Rockwell.

A plataforma didática desenvolvida, Figura 18, ficou composta pelos seguintes dispositivos:

- CLP Compact Logix
- IHM PanelView Plus 400
- Conjunto de chaves e leds

- Processo Peltier



Figura 18 – Plataforma didática
Fonte: Próprio autor

Uma das atividades relacionadas ao desenvolvimento da plataforma foi realizar a ligação dos fios para as entradas e saídas digitais e analógicas. A seguir serão explicados os diferentes módulos presentes no CLP e a maneira como as ligações foram realizadas.

3.5.1 CLP Compact Logix

O Compact Logix é um CLP modular, o mesmo possui quatro módulos instalados, são eles:

- 1769-IQ16 DC Input
- 1769-OW8 AC/DC Output
- 1769-IF4 Analog Input
- 1769-OF2 Analog Output

Aonde o modulo 1769-IQ16 DC Input é um modulo que disponibiliza 16 entradas digitais e possui uma alimentação de 24V DC. Já o modulo 1769-OW8 AC/DC Output é um modulo que disponibiliza 8 saídas digitais, normalmente aberta, através de reles e possui uma alimentação de 24V DC.

O primeiro modulo analógico instalado é o 1769-IF4 Analog Input 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

Por fim, temos o modulo 1769-OF2 Analog Output 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

Ambos os módulos analógicos também são alimentados por 24V DC.

3.5.2 Entradas e Saídas Digitais

Os módulos de entrada e saída digitais foram conectados as chaves e leds, respectivamente. Aonde, na caixa, as quatro chaves inferiores são chaves normalmente abertas e as quatro chaves superiores são chaves normalmente fechadas. As chaves e LEDs são numerados como mostra a Figura 19.



Figura 19 – Numeração das chaves e LEDs
Fonte: Próprio autor

A tabela 1 mostra o endereço de cada chave.

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

Os endereços dos LEDs são apresentados na tabela 2.

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

3.5.3 Entradas e Saídas Analógicas

As entradas e saídas analógicas foram ligadas a um processo Peltier, ilustrado na Figura 20, de acordo com a tabela 3.

Tabela 3 – Entradas e saídas analógicas

Entradas	Endereço
U1	Local:4:O.Ch0Data
U2	Local:4:O.Ch1Data
Saídas	Endereço
T1	Local:3:I.Ch0Data
T2	Local:3:I.Ch1Data
T3	Local:3:I.Ch2Data
T4	Local:3:I.Ch3Data

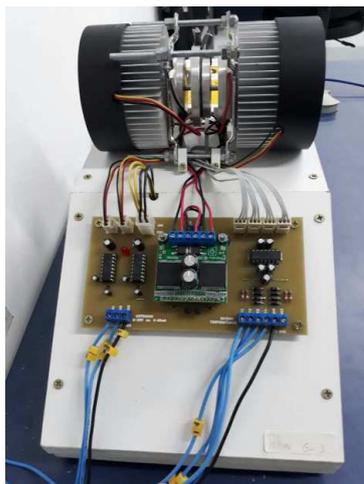


Figura 20 – Processo Peltier G2-3
Fonte: Próprio autor

4 Conclusão

Por meio das atividades realizadas durante o estágio supervisionado foi possível aprofundar os conhecimentos a respeito de controladores lógicos programáveis através das dificuldades enfrentadas no momento de realizar sua configuração. As atividades realizadas durante o estágio supervisionado permitiram o aluno aplicar os conhecimentos adquiridos nas disciplinas de Controle Digital e Sistemas de Automação Industrial.

Com os conhecimentos adquiridos foi possível produzir guias experimentais que explicam o funcionamento básico do CLP, bem como da linguagem Ladder. Estes guias estarão disponíveis para os alunos futuros da disciplina de Sistemas de Automação Industrial. Ainda como resultado do estágio, foi desenvolvida uma plataforma didática, que também ficara a disposição dos alunos do laboratório para atividades futuras. Sendo assim, a partir dessas considerações, pode-se concluir que o estágio atingiu o objetivo proposto.

Referências

AUTOMATION, R. *1769 CompactLogix Controllers User Manual*. [S.l.], 2013. Citado na página 25.

AUTOMATION, R. *Logix5000 Controllers Common Procedures*. [S.l.], 2015. Citado na página 25.

HI Tecnologia. 2017. <http://doc.hitecnologia.com.br/H7SCPSEVERII_MAN_ROOT/00000H7SCPSEVERII_MAN>. Acessado em: 04-08-2017. Citado 3 vezes nas páginas 15, 16 e 17.

LIEC Laboratorio. 2017. <<https://www.liec.ufeg.edu.br>>. Acessado em: 04-08-2017. Citado na página 13.

OPC ToolBox. 2017. <<https://www.mathworks.com/products/opc.html>>. Acessado em: 04-08-2017. Citado na página 20.

ANEXO A – Guias Experimentais

Universidade Federal da Paraíba
Centro de Ciências e Tecnologia
Departamento de Engenharia Elétrica
Laboratório de Controle Digital

Guia Rockwell

16 de Agosto de 2017

1 Objetivo

Este guia tem como objetivo familiarizar o aluno com a plataforma didática disponível no laboratório de automação do LIEC. Incluindo como instalar a maquina virtual, como criar um projeto no software RSLogix5000 e como funciona a plataforma didática.

2 Instalação da Maquina Virtual

Os softwares necessários para configurar o CLP da Rockwell estão instalados em uma maquina virtual, que por sua vez, esta salva em um HD externo. Então, para que seja possível fazer uso de tais softwares é necessário primeiro instalar a maquina virtual.

O software escolhido para rodar a maquina virtual foi o *VMware Workstation 12 Player*.

O procedimento para a instalação da maquina virtual é o seguinte:

1. Copiar os arquivos do HD externo para o computador;
2. Instalar o software VMware;
3. Abrir os arquivos da maquina virtual no VMware;
4. Clicar em *Play Virtual Machine*

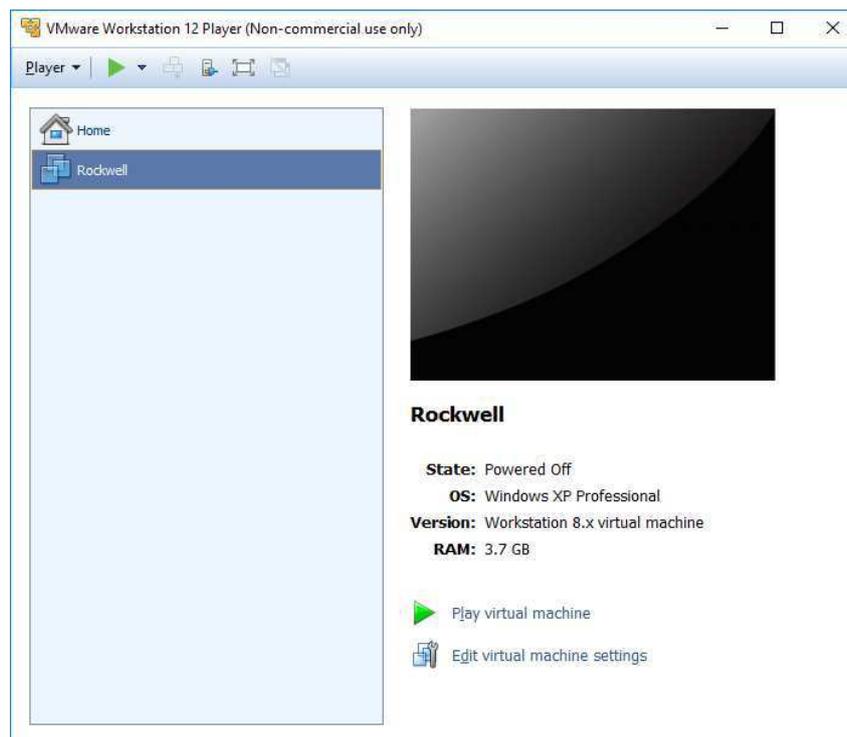


Figura 1: Tela inicial do VMware

3 Criando um Projeto

3.1 Como criar um projeto

A criação de um projeto deve ser o primeiro passo ao se começar a trabalhar com o controlador CompactLogix. Para a criação de um projeto sera utilizado o software RSLogix 5000 instalado na maquina virtual.

Ao abrir o RSLogix 5000, crie um novo projeto e a tela exibida na Figura 2 sera exibida.

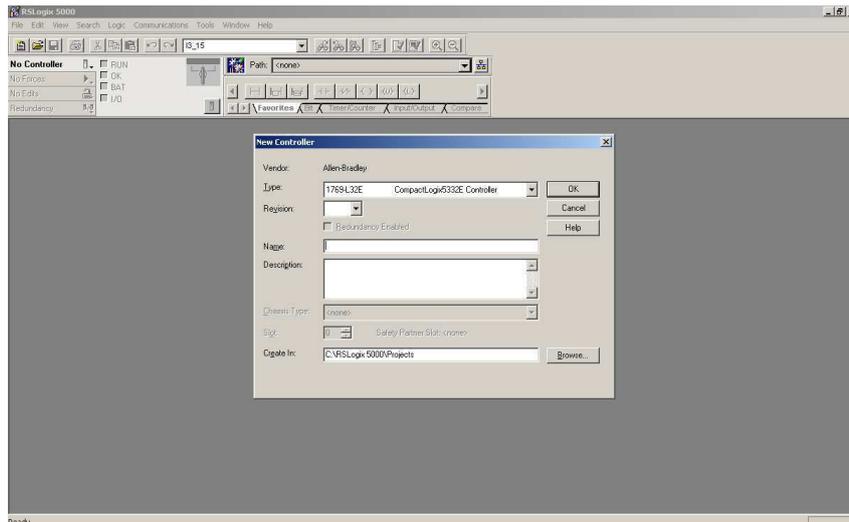


Figura 2: Novo projeto no RSLogix 5000

Deve-se então escolher o tipo do controlador de acordo com o que sera utilizado, no LIEC se utiliza o 1769-L32E, revisão 15.

Por fim, é necessário definir um nome para o controlador (este será também o nome do projeto), uma descrição e o local aonde sera salvo o projeto.

3.2 Como adicionar módulos E/S

Com o projeto criado é necessário agora adicionar os módulos E/S. A arquitetura básica de um controlador CompactLogix do LIEC possui os seguintes módulos:

- 1769-IQ16 DC Input
- 1769-OW8 AC/DC Output
- 1769-IF4 Analog Input
- 1769-OF2 Analog Output

Os módulos devem ser adicionados nos respectivos slots em que estão montados fisicamente.

Para adicionar um modulo ao seu projeto, deve-se clicar com o botão direito em CompactBus Local e depois em New Module como ilustra a Figura 3.

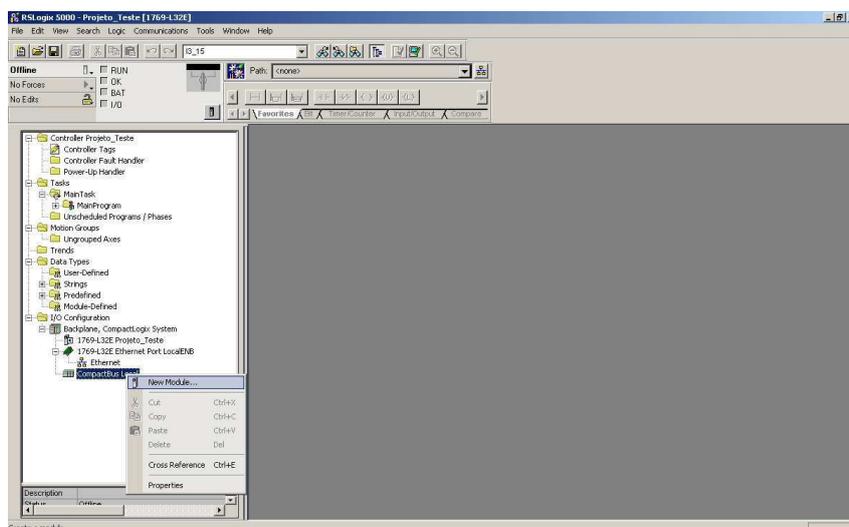


Figura 3: Adicionando um novo modulo

Uma nova janela para selecionar o modulo devera aparecer, basta então procurar e selecionar o modulo desejado. Logo apos, deve-se escolher um nome para o modulo e o seu respectivo slot, como ilustra a Figura 4.

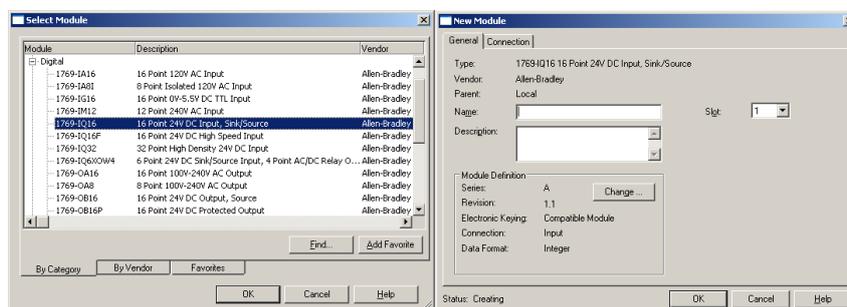


Figura 4: Adicionando um novo modulo

Este processo devera ser repetido para todos os outros módulos que estejam instalados. Lembre-se de salvar este projeto para as atividades futuras, de modo a evitar a necessidade de criar um novo projeto toda vez que for utilizar o CLP.

3.3 Lendo dados dos módulos E/S

As informações de E/S estão dispostas em tags. Cada tag possui uma estrutura de dados, esta estrutura vai depender do respectivo modulo a qual a tag pertence.

O nome da tag é baseado na localização do modulo E/S no sistema. Um endereço E/S segue o seguinte formato: Localização:Slot:Tipo:Membro.SubMembro.Bit.

Ao se adicionar um modulo novo o software automaticamente cria as respectivas tags para os dados disponíveis para aquele modulo. Estas tags estão disponíveis em **Controller Tags**, como ilustra a Figura 5.

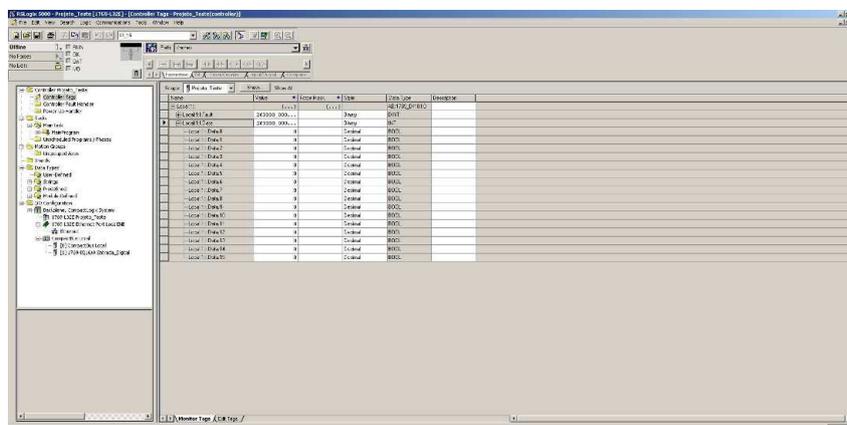


Figura 5: Tags de um modulo adicionado

3.4 Criando um alias para uma tag

De modo a aumentar a abstração do programa muitas vezes se deseja atribuir nomes que representem a função de cada variável. Para fazer isto, pode-se definir um alias para uma das tags de um modulo de E/S.

Para definir um alias deve-se ir na aba **Edit Tags**, escolher um nome e depois indicar qual o endereço da tag para a qual este alias se refere. A Figura 6 ilustra este procedimento.

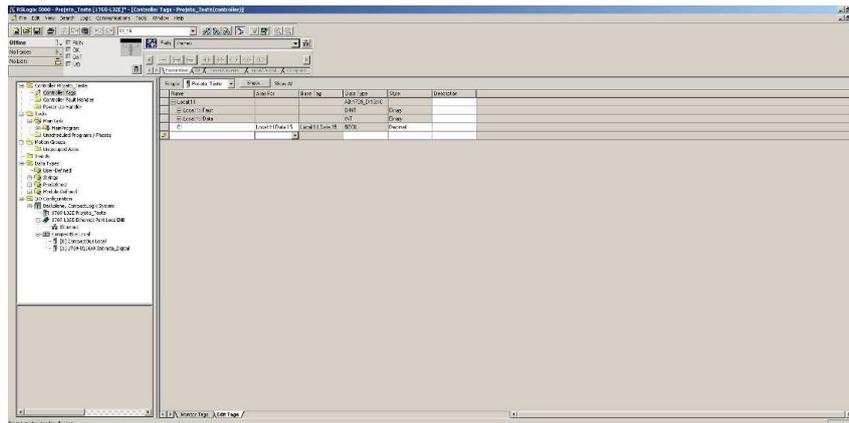


Figura 6: Definindo um alias

4 Linguagem Ladder

4.1 Rotina principal

Quando você cria um novo projeto o software automaticamente cria uma rotina em ladder. Esta rotina esta localizada na pasta MainProgram, como ilustra a Figura 7.

Para inserir uma instrução na linha, clique na linha para seleciona-la e então clique no ícone correspondente da instrução desejada. Se a instrução que você deseja não esta visível, como por exemplo um timer, clique na aba de grupos apropriada.

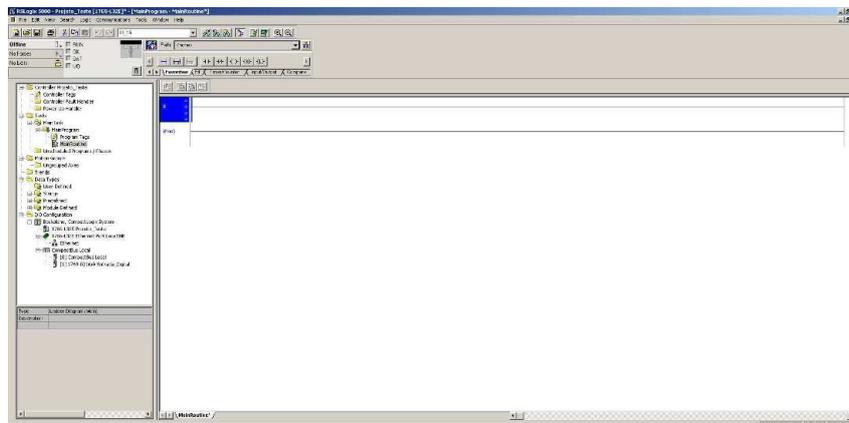


Figura 7: Rotina Principal

5 Fazendo Download/Upload do Projeto

Para estabelecer uma conexão com o CLP deve-se abrir a janela Who Active, depois selecionar a aba AB_ETH-X, Ethernet em que o seu CLP esteja localizado. Por fim, deve-se selecionar o IP do CLP e então clicar em Go Online. A Figura 8 ilustra este procedimento.

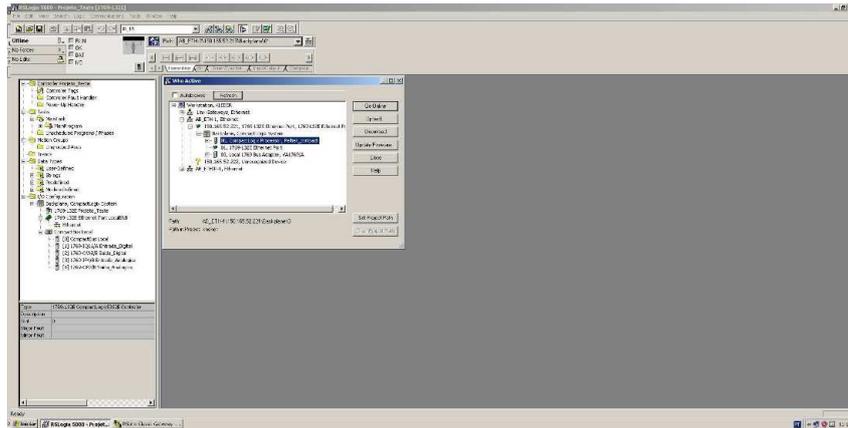


Figura 8: Janela Who Active

Ao ficar online com o CLP, você terá duas opções fazer Upload do programa que esta no CLP para sua maquina ou fazer Download do programa que esta atualmente no RSLogix5000 para o CLP.

6 Plataforma Didática

Para facilitar o uso do CLP foi desenvolvido pelo LIEC uma plataforma didática, Figura 9, que é composta pelos seguintes dispositivos:

- CLP Compact Logix
- IHM PanelView Plus 400
- Conjunto de chaves e leds
- Processo Peltier



Figura 9: Plataforma didática

7 Entradas e Saídas Digitais

Os módulos de entrada e saída digitais foram conectados as chaves e leds, respectivamente. Aonde, na caixa, as quatro chaves inferiores são chaves normalmente abertas e as quatro chaves superiores são chaves normalmente fechadas. As chaves e LEDs são numerados como mostra a Figura 10.

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



Figura 10: Numeração das chaves e LEDs

A tabela 1 mostra o endereço de cada chave.

Os endereços dos LEDs são apresentados na tabela 2.

8 Entradas e Saídas Analógicas

As entradas e saídas analógicas foram ligadas a um processo Peltier, ilustrado na Figura 11, de acordo com a tabela 3.

Tabela 3: Entradas e saídas analógicas

Entradas	Endereço
U1	Local:4:O.Ch0Data
U2	Local:4:O.Ch1Data
Saídas	Endereço
T1	Local:3:I.Ch0Data
T2	Local:3:I.Ch1Data
T3	Local:3:I.Ch2Data
T4	Local:3:I.Ch3Data

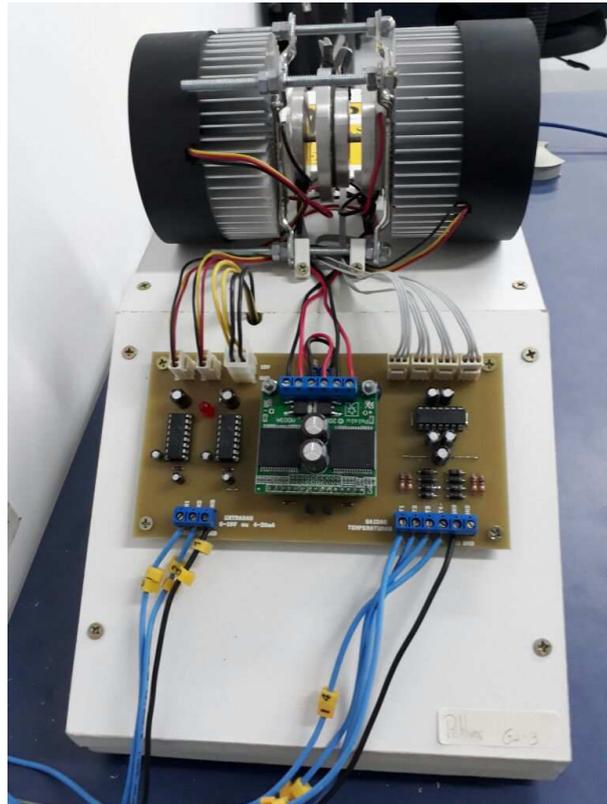


Figura 11: Processo Peltier

O CLP CompactLogix utiliza uma palavra de 16 bits, utilizando a representação por complemento de dois, para os dados analógicos. Sendo assim, tanto a saída quanto a entrada possuem uma resolução de -32767 a 32767.

Universidade Federal da Paraíba
Centro de Ciências e Tecnologia
Departamento de Engenharia Elétrica
Sistemas de Automação Industrial

Guia 1 - Linguagem Ladder

16 de Agosto de 2017

1 Objetivos

Este guia tem como o objetivo familiarizar o aluno com a programação em Ladder utilizada pelo software RSLogix 5000 para o CLP Compact Logix do fabricante Allen-Bradley. Durante o guia serão apresentados blocos básicos, o aluno deverá estudar o funcionamento destes blocos para que durante a aula execute as atividades apresentadas ao final do guia.

2 Programando em Ladder

A programação em ladder é realizada em linhas, aonde é possível colocar instruções em cada linha. Existem dois tipos básicos de instruções, são elas:

Instruções de entrada: Uma instrução que compara e avalia condições específicas da sua máquina ou processo.

Instruções de saída: Uma instrução que executa uma ação, como: ligar ou desligar um aparelho, copiar um dado ou calcular um valor.

É possível ainda adicionar instruções em paralelo com a utilização de *branches*, como ilustra a Figura 1.

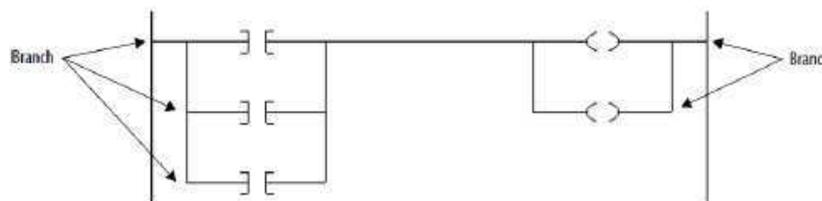


Figura 1: Exemplos de *branches*

O controlador avalia as instruções ladder baseando-se nas condições de linha precedendo a instrução. Como ilustra a Figura 2.

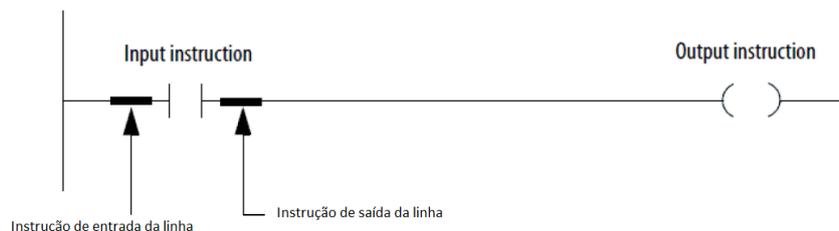


Figura 2: Condições de entrada e saída

Caso a condição de entrada de uma instrução de entrada seja verdadeira, o controlador avalia a instrução e seta a condição de saída para corresponder ao resultado da avaliação.

- Se o resultado da avaliação for verdadeiro, a condição de saída se torna verdadeira.
- Se o resultado da avaliação for falso, a condição de saída se torna falsa.

Uma instrução de saída não muda a sua condição de saída.

- Se a condição de entrada de uma instrução de saída for verdadeira, sua condição de saída também será verdadeira.
- Se a condição de entrada de uma instrução de saída for falsa, sua condição de saída também será falsa.

3 Instruções Básicas

Esta são instruções que se encontram na barra de ferramentas que estarão sempre visíveis, como pode ser visto na Figura 3, são elas:

- Rung
- Branch
- Branch Level

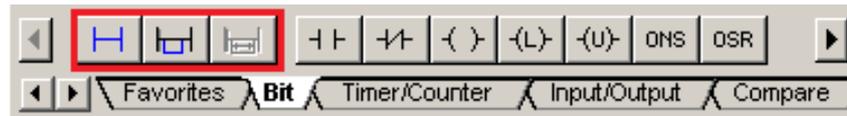


Figura 3: Instruções básicas

A instrução Rung serve para adicionar uma nova linha. Já as instruções Branch e Branch Level são utilizadas para a adição de *Branches* nas linhas.

4 Blocos Bit

Nesta seção de instruções se encontram as instruções de Bit da linguagem Ladder. Dentre as quais, temos:

- Examine ON
- Examine OFF
- Output Energize

A instrução Examine ON corresponde a chave normalmente aberta, enquanto que a instrução Examine OFF corresponde a chave normalmente fechada, estas são instruções de condição de linha. Temos ainda a instrução Output Energize que corresponde a um relé, aonde ao ser ativada irá energizar o bit correspondente. Estas instruções podem ser vistas na Figura 3 ao lado das instruções básicas.

5 Blocos Temporizadores e Contadores

Estes blocos estão localizados na aba *Timer/Counter* da barra de tarefas.

Dentre estes blocos estão incluídos os blocos:

- Timer On Delay
- Timer Off Delay
- Count Up
- Count Down
- Reset

Tabela 1: Tabela para o bloco TON		
Mnemônico	Tipo de Dado	Descrição
.EN	BOOL	Indica que o bloco esta ativo
.TT	BOOL	Indica que uma operação de tempo esta em processo
.DN	BOOL	Indica que o temporizador foi atingiu sua condição. $ACC \geq PRE$
.PRE	DINT	Valor predeterminado que especifica quanto tempo o ACC deve alcançar antes de setar
.ACC	DINT	Valor acumulado desde que o bloco TON/TOF foi ativado

5.1 Timer On/Off Delay (TON/TOF)

O bloco TON é um temporizador não-retentor que acumula tempo quando a sua condição de linha é verdadeira. Já o bloco TOF é também um temporizador não-retentor, porem este acumula tempo quando a sua condição de linha é falsa. A Figura 4 ilustra estes blocos.

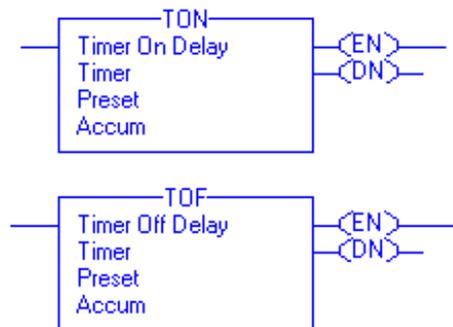


Figura 4: Blocos TON e TOF

Neste bloco é preciso definir três campos, são eles: TIMER, PRESET e ACCUM.

O campo TIMER deve ser preenchido com uma tag, esta tag sera uma estrutura de tempo. Já os campos PRESET e ACCUM são preenchido com valores inteiros. A estrutura de tempo TIMER contem diferentes variáveis que estão definidas na Tabela 1. Note que a estrutura de tempo de um bloco TOFF é um pouco diferente de uma estrutura de tempo para um bloco TON, deve-se então consultar o help.

5.2 Count Up/Count Down (CTU/CTD)

Como o nome sugere o bloco CTU é um contador que conta de forma crescente. Enquanto que o bloco CTD é um bloco que conta de forma decrescente. O bloco CTU conta a partir das bordas de subida do bit .CU e quando atingir o seu valor predeterminado o bit .DN vai de zero para um. De maneira similar o bloco CTD usa as bordas de subida do bit .CD para contar de forma decrescente e quando atingir seu valor predeterminado o bit .DN vai de um para zero. A Figura 5 ilustra ambos os blocos.

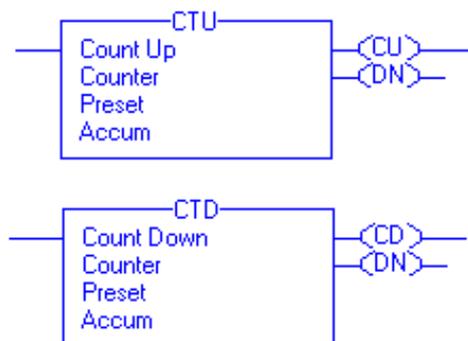


Figura 5: Blocos CTU e CTD

Tabela 2: Tabela para o bloco CTU

Mnemônico	Tipo de Dado	Descrição
.CU	BOOL	Bit de ativação do CTU, indica que a instrução CTU esta ativa
.DN	BOOL	Bit de termino, indica que $.ACC \geq .PRE$
.OV	BOOL	Bit de overflow indica que o contador excedeu o limite superior de 2,147,483,647. O con
.UN	BOOL	Bit de underflow, indica que o contador excedeu o limite inferior de -2,147,483,647. O c
.PRE	DINT	O valor predeterminado que especifica qual valor acumulado deve ser alcançado para qu
.ACC	DINT	Valor acumulado, indica o numero de transições que a instrução já contou.

Tabela 3: Tabela para o bloco CMP

Operador	Descrição
=	Igual
<	Menor que
<=	Menor ou igual que
>	Maior que
>=	Maior ou igual que
<>	Diferente de

Os campos para estes blocos são similares aos campos dos blocos TON e TOF. Sendo assim, é necessário que o usuário defina uma tag para conter a estrutura de tempo. As variáveis contidas na estrutura de tempo para o bloco CTU estão especificadas na Tabela 2. As variáveis para a estrutura de tempo do bloco CTD são similares, porem o aluno pode consultar o help para obter maiores informações.

O valor acumulado continua a incrementar/decrementar, mesmo apos o bit .DN ser setado. Para reiniciar o valor acumulado deve-se utilizar a instrução RES que referencia a estrutura de tempo do contador ou escrever 0 no valor acumulado.

6 Blocos de Comparação

Estes blocos estão localizados na aba *Compare* da barra de tarefas.

Dentre estes blocos estão incluídos os blocos:

- Compare
- Equal
- Not Equal
- Less Than
- Greater Than
- Less Than or Equal
- Greater Than or Equal

6.1 Compare (CMP)

O instrução CMP executa uma comparação na operação aritmética que o usuário especifica na expressão. É possível utilizar este bloco para executar logicas de comparação mais complexas. A equação sera lida da esquerda para direita e faz uso de diversos operadores. A Tabela 3 mostra alguns operadores básicos que podem ser utilizados. Para uma lista completa dos operadores a ser usados deve-se acessar o help deste bloco.

6.2 Blocos Equal (EQU), Not Equal (NEQ), Less Than (LES), Greater Than (GRT), Less Than or Equal (LEQ) e Greater Than or Equal (GEQ)

Estes são blocos de comparação simples, aonde o programa compara a fonte A com a fonte B de acordo com a condição de cada bloco. Estas fontes podem ser tanto tags quanto valores imediatos. Valores imediatos são valores que podem ser atribuídos ao próprio campo do bloco de forma imediata sem a necessidade de tags.

A Figura 6 ilustra um exemplo do bloco GRT. Aonde caso *value 1* seja maior que *value 2*, *light 1* recebe 1. E caso *value 1* seja menor ou igual que *value 2*, *light 1* recebe 0.

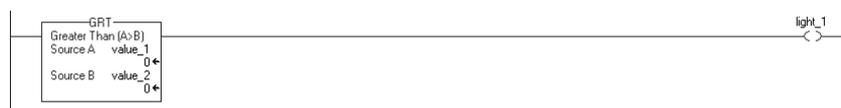


Figura 6: Exemplo do bloco GRT

7 Operações Matemáticas

O RSLogix5000 disponibiliza diversos blocos para a realização de operações matemáticas. São eles:

- Compute (CPT)
- Add (ADD)
- Subtract (SUB)
- Multiply (MUL)
- Divide (DIV)
- Modulo (MOD)
- Square Root (SQR)
- Negate (NEG)
- Absolute Value (ABS)

De modo similar aos blocos de comparação, os blocos de operações matemáticas possuem diversos blocos para executar operações simples, como adição, subtração, multiplicação e outros, e um bloco que possibilita a execução de diversas operações por meio de uma expressão, sendo este o bloco CPT.

7.1 Bloco Compute (CPT)

O bloco CPT executa a operação aritmética definida pelo usuário na expressão. Quando ativado, o bloco CPT calcula o resultado da expressão e coloca o resultado no campo de destino.

A execução de uma instrução CPT é um pouco mais lenta e utiliza mais memória do que a execução de outras instruções matemáticas. Porém, possui a vantagem de que o usuário pode especificar expressões mais complexas em uma única instrução. As operações são executadas da esquerda para a direita, exceto quando existe a presença de parenteses.

A Figura 7 ilustra um exemplo do bloco CPT. Quando ativado, o bloco CPT avalia o *value 1* multiplicado por 5 e divide este valor pelo resultado do *value 2* dividido por 7 e coloca o resultado final em *result 1*.

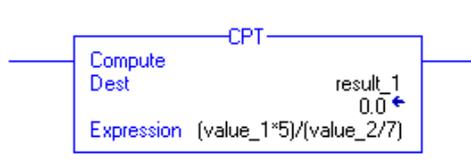


Figura 7: Bloco CPT

7.2 Blocos Add (ADD), Subtract (SUB), Multiply (MUL), Divide (DIV), Modulo (MOD), Square Root (SQR), Negate (NEG) e Absolute Value (ABS)

Estes são blocos de operações matemáticas simples, aonde a função de cada bloco é intuitiva de acordo com seu nome. O programa executa uma operação entre fonte A e a fonte B de acordo com a operação específica de cada bloco e então salva o resultado no campo de destino. Alguns destes blocos necessitam apenas de uma fonte, são eles: MOD, SQR, NEG e ABS. Estas fontes podem ser tanto tags quanto valores imediatos. Valores imediatos são valores que podem ser atribuídos ao próprio campo do bloco de forma imediata sem a necessidade de tags.

8 Blocos Move/Logical

O bloco de movimento (MOV) é o bloco que possui como finalidade a movimentação de variáveis, este bloco é bastante utilizado na leitura de entradas analógica ou na escrita de saídas. Também estão inclusos os blocos de operações lógicas, dentre eles temos os blocos:

- AND
- OR
- XOR
- NOT

8.1 Bloco Move (MOV)

A instrução MOV copia a Fonte para o Destino, mantendo a Fonte inalterada. A Fonte pode ser tanto uma tag quanto um valor imediato, já o Destino devera ser uma tag. A Figura 8 ilustra o bloco MOV.

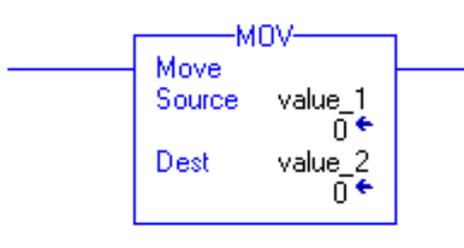


Figura 8: Bloco MOV

8.2 Blocos Bitwise And (AND), Bitwise Or (OR), Bitwise Exclusive Or (XOR) e Bitwise Not (NOT)

A instrução AND executa uma operação AND bit a bit entre os bits da Fonte A e da Fonte B, o resultado é então salvo no campo de Destino. De forma similar a instrução OR executa uma operação OR bit a bit, a instrução XOR executa uma operação XOR bit a bit e a instrução NOT executa uma operação NOT bit a bit, na operação NOT é necessário apenas um campo de Fonte. Os campos de Fonte podem uma tag ou um valor imediato, enquanto que o campo de Destino devera ser uma tag.

9 Atividades Praticas

Exercício 1

Utilize o bloco CTU para simular um contador de 0 a 10. Utilize a chave 1 para gerar os pulsos. O que acontece com o bit .CU quando se pressiona a chave 1? O que acontece com o bit .DN quando o contador atinge o valor 10?

Exercício 2

Escreva um programa que ative as saídas digitais do CLP. A primeira devera ser ativada apos 5 segundos do acionamento da chave 1, a segunda devera ser ativada com 15 segundos, a terceira com 30 segundos e a quarta com 60 segundos. Utilize apenas um bloco TON.

Exercício 3

Utilize os blocos de operação matemática para criar uma logica que converta uma medida em polegadas para centímetros. Depois faça uma segunda logica que converta uma temperatura em Fahrenheit para Celcius.

Exercício 4

A entrada analógica do CLP possui um intervalo de 0 a 4095. Escreva uma logica Ladder que converta esta entrada para um intervalo de 0 a 100. Para testar a logica faça uso de valores imediatos.

Exercício 5

Execute uma operação de mascaramento com a seguinte informação de 16 bits XXXX000110010001, aonde os bits X podem ser 1 ou 0. Na saída devera então aparecer 0000000110010001. O mascaramento é a operação de AND bit a bit entre a informação desejada e a mascara. A mascara é um numero com a mesma quantidade de bits que a informação, aonde os bits correspondentes as entradas que se deseja manter são forçados a 1 e forçados a 0 os bits que devem ser mascarados, no nosso caso são os quatro bits mais significativos.

Universidade Federal da Paraíba
Centro de Ciências e Tecnologia
Departamento de Engenharia Elétrica
Sistemas de Automação Industrial

Guia 2 - Linguagem Ladder

16 de Agosto de 2017

1 Objetivos

Este guia tem como o objetivo de aprofundar o conhecimento do aluno com a programação em Ladder utilizada pelo software RSLogix 5000 para o CLP Compact Logix do fabricante Allen-Bradley. Durante o guia o aluno sera confrontado com diferentes problemas reais e devera escrever uma logica Ladder que solucione tais problemas.

2 Controle de um Semáforo

O aluno devera elaborar uma logica Ladder que faça o controle de um semáforo de acordo com as seguintes condições:

- O semáforo devera iniciar no sinal vermelho
- O sinal vermelho e verde deverão permanecer ligados por 5s
- O sinal amarelo devera permanecer ligado por 2s
- Apenas uma das saídas deverão estar ativas a qualquer momento
- Devera existir um botão para desligar completamente o semáforo e um para ligar

O aluno é aconselhado a fazer uso do bloco TON para o controle do semáforo.

3 Peltier

Para o processo Peltier instalado na plataforma didática, Figura 1, deseja-se que o aluno escreva um programa em Ladder que seja capaz de ler e escrever nas suas saídas. Atendo-se as seguintes exigências:

- A temperatura lida devera estar em graus Celsius
- A saída deve estar em porcentagem (0-100)

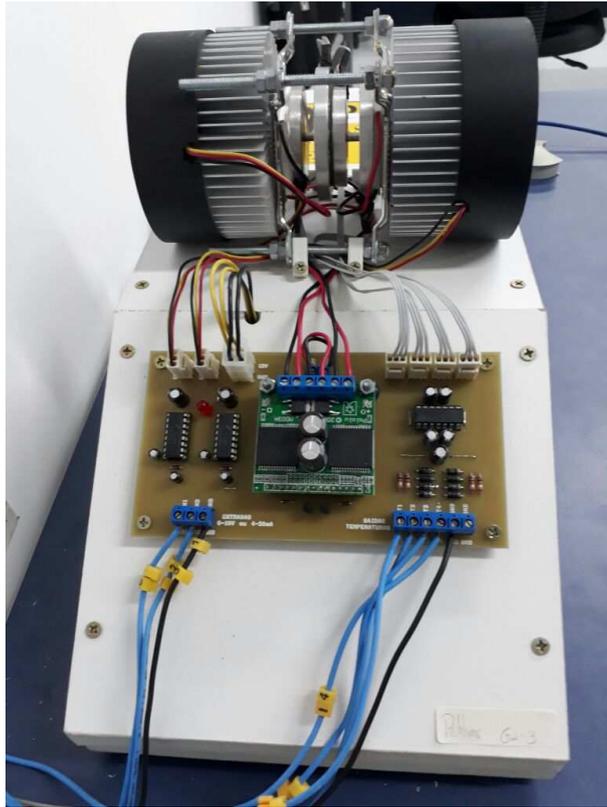


Figura 1: Processo Peltier

Para normalizar a entrada analógica lida deve-se utilizar a seguinte expressão:

$$V_n = V_l * \frac{(U_{emax} - U_{emin})}{V_{max}} + U_{emin} \quad (1)$$

Aonde, V_n é o valor normalizado, V_l é o valor lido, U_{emax} é o valor máximo da escala de temperatura (100), U_{emin} é o valor mínimo da escala de temperatura (0) e V_{max} é o valor máximo da escala em que a variável lida esta (32767). Utilize da mesma equação para normalizar o valor da saída analógica, que possui os mesmos valores de máximo e mínimo.

Dica: utilize o bloco de movimento para ler ou escrever nas saídas analógicas.