



UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Guilherme Pires de Campos Linardi

DESENVOLVIMENTO DE UM SISTEMA SUPERVISÓRIO PARA O LABORATÓRIO DE INSTRUMENTAÇÃO INDUSTRIAL

Manaus
2015

Guilherme Pires de Campos Linardi

DESENVOLVIMENTO DE UM SISTEMA SUPERVISÓRIO PARA O LABORATÓRIO DE INSTRUMENTAÇÃO INDUSTRIAL

Trabalho de Conclusão de Curso submetida à Coordenação do curso de Engenharia de Controle e Automação da Universidade do Estado do Amazonas como parte dos requisitos necessários para a obtenção do grau de Engenheiro. em conformidade com as normas ABNT.

Orientador Me. Moisés Pereira Bastos
Coorientador: Dr. Luís Delfín Rojas Purón

Manaus
2015

Guilherme Pires de Campos Linardi

**DESENVOLVIMENTO DE UM SISTEMA
SUPERVISÓRIO PARA O LABORATÓRIO DE
INSTRUMENTAÇÃO INDUSTRIAL**

Trabalho de Conclusão de Curso submetida à Coordenação do curso de Engenharia de Controle e Automação da Universidade do Estado do Amazonas como parte dos requisitos necessários para a obtenção do grau de Engenheiro, em conformidade com as normas ABNT.

Aprovado em de de 2015.

BANCA EXAMINADORA

Me. Moisés Pereira Bastos
Orientador

Dr. Walter Andrés Vermehren
Valenzuela
Convidado 1

Dr. Israel Francisco Benitez Pina
Convidado 2

Me. Almir Kimura Junior
Convidado 3

Manaus
2015

AGRADECIMENTOS

Em primeiro lugar, agradeço a meus pais Norberto e Maria Helena, que além de todo o apoio financeiro, me deram exemplos de vida e incentivo para não desistir. Agradeço também aos meus irmãos e familiares pelo apoio e pela construção dos meus propósitos, princípios e valores. Agradeço minha namorada Aimée por me apoiar e dar todo incentivo para lutar pelos meus sonhos. Outro agradecimento aos meus amigos Luiz e Nilteomar que me ajudaram na elaboração deste trabalho. Agradeço também a todos os amigos que fiz durante a minha faculdade por todos os momentos que passamos juntos e espero que nossa amizade se prolongue por décadas. Agradeço ao professor Me. Moisés Bastos por aceitar esse desafio ao meu lado, por sua dedicada orientação, competência e amizade criada durante o processo. Agradeço também aos professores Dr. Walter Andrés Vermehren Valenzuela, Dr. Israel Francisco Benitez Pina, Me. Almir Kimura Junior pelas considerações feitas com objetivo de engrandecer meu trabalho. Por fim, mas não menos importante, agradeço a Universidade do Estado do Amazonas, por prover a estrutura e oportunidade para minha formação, à coordenação de Engenharia de Controle e Automação e todos os seus professores, pela disseminação do conhecimento, e em especial ao professor Dr. Daniel Guzmán del Río.

RESUMO

A grande quantidade de processos automatizados existentes nos mais diversos meios industriais faz com que a utilização dos sistemas supervisórios, sistemas em que é permitido monitorar um processo em tempo real e obter seus parâmetros de processo, seja motivada. Atualmente a competitividade exige produtividade. Mas com os sistemas de controle e supervisão manual, o tempo perdido na coleta de dados e a baixa confiabilidade, devido a imprecisão desses dados, vai contra essa necessidade de aumento de produção. Com isso, o presente trabalho desenvolveu um sistema supervisório SCADA (*Supervisory Control & Data Acquisition Systems*) com objetivo de monitorar um processo industrial simulado em bancada. Para o desenvolvimento do mesmo, foi utilizado controladores lógico programáveis de pequeno e médio porte e também interface humano-máquinas de pequeno e médio porte. Com isso, foi desenvolvido o sistema supervisório em três níveis hierárquicos, disponível no laboratório, foi estabelecida a comunicação entre CLP e IHM, a comunicação entre os CLP's e a rede industrial TCP/IP foi efetiva na comunicação. Com isso, foi realizado o monitoramento e controle da variável de processo temperatura.

Palavras-chaves: Sistema supervisório; CLP; redes industriais de comunicação; IHM.

ABSTRACT

The large amount of existing automated processes in various industrial environments makes use of supervisory systems, systems in which it is allowed to monitor a real-time process and get their process parameters, is motivated. Currently competitiveness requires productivity. But with the control and manual control systems, the time lost in data collection and low reliability due to inaccuracy of such data, goes against this need for increased production. Thus, this study developed a supervisory system SCADA (Supervisory Control & Data Acquisition Systems) in order to monitor an industrial process simulated bench. For its development, it used programmable logic controllers of small and medium size as well as small and medium human-machine interface. Thus, it was developed the supervisory system in three hierarquics levels, available in the lab, was stablished the communication between PLC and HMI and PLC with PLC, and the communication industrial network TCP/IP was effective in the communication. Thus, the monitoring and control of temperature process variable was possible.

Key-words: Supervisory System; PLC;Industrial Communication Networks; HMI.

LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura Básica do CLP	14
Figura 2 – CLP Micro 820	16
Figura 3 – CLP CompactLogix L24ER	17
Figura 4 – PanelView™Plus 600	18
Figura 5 – PanelView™Component 600	18
Figura 6 – Estrutura de um Sistema Supervisório	19
Figura 7 – Arquitetura com a estação de monitoração centralizada	20
Figura 8 – Arquitetura com a estação de monitoração distribuída	21
Figure 9 – Camadas de uma rede industrial	23
Figura 10 –Arquitetura de Integração	26
Figura 11 –Arquitetura do SISAL	27
Figura 12 –Exemplo de Esquema Básico	28
Figura 13 –Arquitetura do Sistema Proposto.	30
Figure 14 –Ambiente de trabalho do Connected Component Workbench	31
Figure 15 –Tela da IHM - Parâmetro Temperatura	32
Figure 16 – Relatório gerado pelo gráfico do PVC 600	32
Figura 17 –Alarme acionado no sistema	33
Figure 18 –Editor de <i>Tags</i>	34
Figure 19 – <i>Tags</i> criadas para a IHM da Estação Operador	35
Figura 20 –Configurando IP e Máscara da Sub-Rede	36
Figura 21 –Reconhecimento da conexão entre PC e PLC	36
Figura 22 –Programação da Estação Operador	37
Figura 23 –Protocolo de Comunicação Selecionado	38
Figura 24 – <i>Tags</i> da IHM endereçadas	39
Figura 25 –Sensor de Temperatura LM35	40
Figura 26 –Circuito teste do Sensor de Temperatura	40
Figura 27 –Primeira programação para leitura do sensor LM35	41
Figura 28 –Segunda programação para leitura do sensor LM35	41
Figura 29 –Monitor Serial com sensor de temperatura	42
Figura 30 –Variáveis Locais para Sensor de Temperatura	42
Figura 31 –Programação do Bloco de Função do Sensor	43
Figure 32 –Faixa de Trabalho do Motor de Passo	43
Figura 33 –Esquema de Ligação	44
Figura 34 –Sequência de Fases	44
Figura 35 –Conexão do driver a4988 e do motor de passo	45

Figura 36	–Primeira Programação para motor de passo	46
Figura 37	–Variáveis Locais para Atuador	46
Figura 38	–Segunda Programação para motor de passo	47
Figura 39	–Diagrama do temporizador TP	47
Figura 40	–Diagrama do temporizador TOF	48
Figura 41	–PVP 600 com IP estático	49
Figura 42	–Drivers de comunicação via OPC	49
Figura 43	–Dispositivos do driver Ethernet PVP 600	50
Figura 44	–Tela Inicial do programa desenvolvido no PVP 600	50
Figura 45	–Tela de Temperatura com Atuador Desativado	51
Figura 46	–Tela de Temperatura com Atuador Ativado	51
Figura 47	–Tela de Configuração	52
Figura 48	–Menu Tag	52
Figura 49	–Tags Criadas para PVP 600	53
Figura 50	–Exemplo de Tag selecionada na aba de conexão	53
Figura 51	–Programação para tag do atuador	54
Figura 52	–Pirâmide da Automação	54
Figura 53	–OPC com CompactLogix L24ER e seu atalho	55
Figura 54	–Tag de mensagem de leitura	56
Figura 55	–Tag de mensagem de escrita	57
Figura 56	–Primeira programação para teste da comunicação entre CLP's	57
Figura 57	–Segunda programação para teste da comunicação entre CLP's	58
Figura 58	–Programação para transformar as casas decimais em inteiro	58
Figura 59	–Programação no CompactLogix L24ER para adicionar o decimal do inteiro da Temperatura	59

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Problemática	10
1.2	Objetivo geral	11
1.3	Objetivos específicos	11
1.4	Justificativa	11
1.5	Motivação	12
1.6	Metodologia	12
1.7	Organização	12
2	REFERENCIAL TEÓRICO	14
2.1	Controlador lógico programável	14
2.1.1	Micro 820	15
2.1.2	CompactLogix L24ER	16
2.2	Interface humano-máquina	17
2.3	Sistemas supervisórios	19
2.3.1	Sistema supervisório de controle e aquisição de dados (SCADA)	21
2.4	Redes industriais	23
3	TRABALHOS RELACIONADOS	25
3.1	Estudo da integração entre sistemas SCADA, MES e ERP em empresas de manufatura discreta que utilizam processos de usinagem	25
3.2	SISAL - Um sistema supervisório para elevação artificial de petróleo	26
3.3	Sistemas supervisórios	27
4	METODOLOGIA	30
4.1	Arquitetura do sistema proposto	30
4.2	Estação do operador	31
4.2.1	Interface humano-máquina PanelView™ Component 600	31
4.2.2	Controlador lógico programável Micro820	35
4.2.3	Comunicação entre CLP e IHM	37
4.2.4	Processo industrial simulado	39
4.2.4.1	Sensor de Temperatura LM35	39
4.2.4.2	Motor de passo AK17 e <i>driver</i> a4988	43
4.3	Estação central de monitoramento	48
4.3.1	Interface humano-máquina PanelView™ Plus 600	48
4.3.2	Controlador lógico programável CompactLogix L24ER	54

4.4	Comunicação entre as estações	55
5	CONCLUSÃO	60
	Referências	61

1 INTRODUÇÃO

Os sistemas supervisórios surgiram da necessidade de facilitar o controle dos processos produtivos. Num cenário o qual toda coleta de dados e todos os controles de parâmetros de um processo eram feitos de maneira manual e lenta, muitas vezes sem ser possível obter confiabilidade, gastava-se muito tempo e mão de obra que não resultavam, ao fim, em grandes resultados de produtividade.

A fim de suprir a necessidade e melhorar o cenário, os sistemas supervisórios surgiram oferecendo à seus usuários rápida e confiável coleta de dados e informações, armazenagem e fácil acesso à grande quantidade de dados e o controle das variáveis do processo, muitas vezes também corrigindo imediatamente seus erros. O sistema faz isso tudo de maneira automática e em tempo real, permitindo que não exista mais tempo morto de seus usuários e aumentando a confiabilidade de dados e de controle do processo.

Basicamente, isso se torna possível por meio da implementação de controladores lógicos programáveis (CLP's), e de redes industriais que trocam informações entre si, por meio de protocolos de comunicação.

Como Clarke, Reynders e Wright (2004) diz em seu livro, os sistemas supervisórios são o coração da indústria moderna, que vão desde plantas de mineração e água até instalações de serviços elétricos para, por exemplo, usinas de petróleo e gás.

Na manufatura moderna e nos processos industriais é necessário conectar equipamentos e sistemas separados por longas distâncias. Essa distância varia de alguns metros para até milhares de quilômetros. A telemetria é utilizada para enviar comandos e programas e receber informações de monitoramento desses locais remotos. Ou seja, um sistema supervisório basicamente é uma combinação de telemetria e aquisição de dados (BAILEY; WRIGHT, 2003).

Com o advento dos computadores e outros aparelhos eletrônicos, os fabricantes de lógica de transmissão, até então utilizada para controle de produção e sistemas de plantas, incorporaram a eletrônica digital em seus equipamentos (BAILEY; WRIGHT, 2003).

1.1 Problemática

Atualmente, a competitividade exige produtividade. Mas, com os sistemas de controle e supervisão manual, o tempo perdido na coleta de dados e a baixa confiabilidade, devido a imprecisão desses dados, vai contra essa necessidade de aumento de produção. Algumas das desvantagens da utilização do sistema manual são:

- Falta da precisão de monitoramento das variáveis do processo devido a coleta manual;
- Necessidade de mão de obra especializada e alto tempo morto;
- Baixo desempenho de produção devido a demora de resposta da mão de obra;

Todas essas desvantagens podem ser resolvidas com a implementação de um sistema supervisório.

1.2 Objetivo geral

Desenvolver um sistema supervisório visando monitorar e controlar um processo industrial simulado no laboratório de instrumentação industrial. O sistema vai utilizar uma interface humano-máquina (IHM) e uma rede industrial de comunicação visando interligar os dois controladores lógicos programáveis (CLPs) do processo, configurando um sistema de controle descentralizado.

1.3 Objetivos específicos

- Realizar levantamento bibliográfico e estudo sobre as áreas envolvidas no projeto.
- Implementar a comunicação entre CLP's e IHM.
- Desenvolver a comunicação entre as estações multiníveis.
- Elaborar o sistema supervisório.
- Testar o sistema.

1.4 Justificativa

Num processo industrial pode existir um alto número de parâmetros a serem controlados, a fim de estabilizar o processo em si. Esse controle nem sempre pode ser obtido manualmente, o que faz com que determinada variável crítica possa sair do limite sem que seja detectada, impactando diretamente na qualidade do produto fabricado, de maneira qualitativa e quantitativa.

A implementação de um sistema supervisório tem como função a captura e armazenamento dos parâmetros de processo em um banco de dados. Após isso, os mesmos são comparados com um padrão de condição ideal pré-estabelecido pelo usuário, e então, por meio dos CLP's, o sistema faz as alterações necessárias de maneira automática para retornar os parâmetros do processo à condição ideal.

Esse controle autônomo oferece como principal benefício o aumento da produtividade além de gerar maior confiabilidade e que sejam reduzidos custos referentes a mão de obra necessária. Em termos econômicos, o uso do sistema permite que o processo tenha uma relação custo-benefício muito mais atrativa do que o controle manual.

1.5 Motivação

Durante aula de informática industrial ao ter acesso a artigo sobre sistemas supervisórios foi possível adquirir conhecimento crítico para perceber que em um ambiente de trabalho industrial se perde muitas informações importantes ao longo dos processos por consequência da má organização e de falta de supervisão adequada. A partir dessa concepção foi possível entender a importância que o assunto possui perante a formação profissional de um engenheiro de controle de automação.

A automatização de processos é parte fundamental do curso de Engenharia de Controle e Automação e para este projeto serão utilizados os conhecimentos que foram adquiridos nas disciplinas de Instrumentação Industrial, Modelagem e Simulação de Sistemas Discretos, Redes de Computadores I e Informática Industrial.

1.6 Metodologia

Para o desenvolvimento desse trabalho são empregadas as seguintes atividades:

- Estudo de CLP, IHM, sistemas supervisórios e redes industriais de comunicação;
- Definição do processo que será utilizado: estudo do processo mais viável para implementação do sistema supervisório;
- Pesquisa de custos para aquisição dos CLPs e IHMs: a pesquisa será baseada no levantamento dos CLP's disponíveis no mercado;
- Estudo da comunicação entre o CLP e o IHM: verificar qual a melhor comunicação entre os CLP e o IHM do fabricante;
- Implementação da comunicação entre o CLP e IHM,
- Estudo e definição do tipo de rede industrial de comunicação a ser utilizada: levando em consideração o CLP selecionado e a comunicação estabelecida, será selecionada a rede industrial utilizada;
- Desenvolvimento da Interface humano máquina, levando em consideração os aspectos de confiabilidade, usabilidade e segurança;
- Implementação do sistema supervisório.

1.7 Organização

O Capítulo 1 apresenta uma visão geral do trabalho, compreendendo o resumo do trabalho, uma introdução ao tema em geral, além de apresentar os objetivos, a justificativa e a motivação do trabalho, em seguida, apresenta a metodologia utilizada.

Os Capítulos 2 e 3 são destinados à teoria envolvida pelo tema do trabalho. o Capítulo 2 é voltado para a descrição da teoria sobre controladores lógico programáveis, interface humano-máquina, redes industriais e sistemas supervisórios.

No Capítulo 3 apresenta-se um resumo sobre alguns trabalhos relacionados com o tema abordado.

O Capítulo 4 é relacionado ao desenvolvimento do projeto e resultados obtidos, envolvendo o projeto do sistema supervisório, a comunicação, por meio da rede industrial determinada, entre os CLPs e as IHMs, e a parte de comunicação do sensor e do atuador com o sistema. Por fim, no Capítulo 5 é feita a conclusão do trabalho.

2 REFERENCIAL TEÓRICO

2.1 Controlador lógico programável

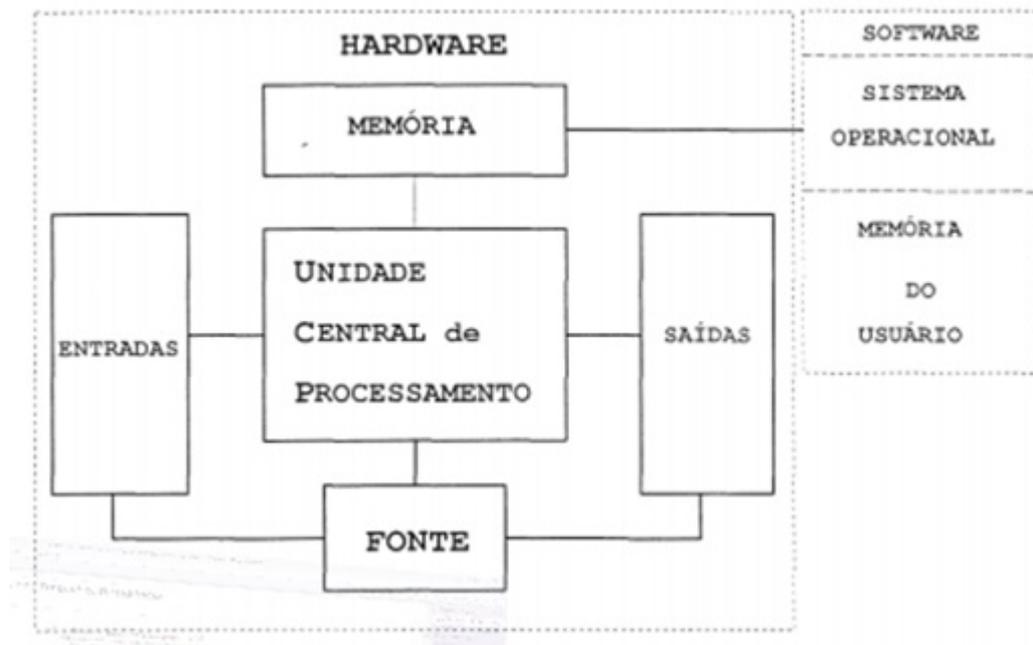
A definição de controlador lógico programável (CLP) é bastante ampla. Pode-se dizer que CLP é um computador que atende a necessidade de operação para ambientes industriais. Ele possui um software para uma determinada atividade de controle e automação e um sistema operacional de tempo real e com memória com uma configuração fixa (KOPELVSKI, 2010).

A norma NEMA (National Electrical Manufacturers Association), ICS3-1978, parte ICS3-304, define um controlador programável como:

Aparelho eletrônico digital que utiliza uma memória programável para armazenamento interno de instruções para implementação de funções específicas, tais como lógica, sequenciamento, temporização, contagem e aritmética para controlar por meio de módulos de entradas e saídas, vários tipos de máquinas ou processos. Um computador digital que é utilizado para desempenhar as funções de um controlador programável é considerado dentro deste escopo. Estão excluídas as chaves tambores e outros tipos de sequenciadores mecânicos.

A estrutura básica de um CLP pode ser exemplificada de acordo com o esquema representado na figura 1.

Figura 1 – Arquitetura Básica do CLP



Fonte: (KOPELVSKI, 2010)

A estrutura de um CLP é composta por unidades responsáveis por diferentes atividades:

- Unidade Central de Processamento (UCP): onde recebe, decodifica, reserva e processa informação. Também executa o programa de controle armazenado na memória do CLP. Ela usa instruções especiais codificadas para executar suas funções (PRUDENTE; ROCHA; II, 2011);
- Fonte de alimentação: responsável pelo fornecimento de energia elétrica ao CLP. Fornece todos os níveis de tensão exigidos para a operações internas do CLP. Como os CLPs são modulares, algumas vezes é necessário pensar em uma segunda fonte para suportar o aumento de periféricos (KOPELVSKI, 2010);
- Entradas e saídas: são como o CLP se comunica com o processo que vai ser controlado. As entradas recebem os sinais analógicos e os transformam em sinais digitais para que sejam processados pela UCP. Finalizado o processamento, os dados digitais enviados pela UCP são convertidos novamente em sinais analógicos pelas saídas (GEVAERD; SILVA; BONACORSO, 2011).

2.1.1 Micro 820

No presente projeto o CLP que será utilizado é o sistema de controle Micro820, apresentado na figura 2. Esse sistema foi projetado visando o controle independente em máquinas de pequeno porte e para automação remota dependente de comunicação flexível e recurso de E/S (AUTOMATION, 2014d).

As características gerais desse CLP são: dois canais de entrada analógica não isoladas de 0 a 10 volts e dois canais de saída analógica, incorporados, para controle de velocidade de um inversor, comunicações incorporadas via uma porta serial não isolada (para comunicações RS-232 e RS-485) e porta Ethernet, slot microSDTM incorporado para transferência de programas, registro de dados e gestão de recepção e suporte incorporado para até quatro entradas de termistor de temperatura de 10k (AUTOMATION, 2014d). Neste trabalho foi utilizado uma entrada analógica, uma saída digital e o protocolo de comunicação TCP/IP.

Figura 2 – CLP Micro 820



Fonte: (AUTOMATION, 2014d)

2.1.2 CompactLogix L24ER

Além do Micro 820, também iremos usar outro tipo de CLP: o CompactLogix L24ER, como pode-se ver na figura 3. Esses controladores fornecem soluções estáveis de controle capazes de atender uma grande variedade de aplicações. Suas características são: duas portas Ethernet para comunicação, uma porta USB, suporta módulos de expansão, controle de módulos I/O local e distribuído, usa, para memória não volátil, a segurança digital 1784-SD1 ou 1784-SD2 e tem uma solução de armazenamento interna de energia, eliminando a necessidade de bateria (AUTOMATION, 2014a). Nesse trabalho, foi utilizado o protocolo de comunicação Ethernet/IP, mas nenhuma entrada ou saída porque todas as variáveis utilizadas foram entradas via IHM.

Figura 3 – CLP CompactLogix L24ER



Fonte: (AUTOMATION, 2014a)

2.2 Interface humano-máquina

Em todo sistema supervisório, se faz a necessidade de uma Interface Humano Máquina (IHM), que é o canal de comunicação por meio do qual o usuário pode interagir com a máquina ou o processo. Em outras palavras, a IHM é um hardware industrial normalmente composto por uma tela de LCD e por um conjunto de teclas que possibilita a navegação e inserção dos dados para um software proprietário da programação. Além disso, pode-se utilizar a IHM para várias situações, como a visualização dos dados do processo, das anomalias do sistema, por meio de alarmes, entre outras funções (CORDEIRO; OLIVEIRA; CHANQUINI, 2009).

Dentre os vários tipos de IHM existentes serão utilizados dois tipos no presente trabalho, o PanelView Plus 600 (PVP 600) e o PanelView Component 600 (PVC 600), ambos desenvolvidos pela fabricante Allen-Bradley.

A diferença básica entre eles é que os PVP 600 são bons para aplicações que requerem habilidade para monitorar, controlar e mostrar informação de maneira dinâmica. Isso permite com que o usuário entenda o status do processo para então tomar decisões melhores. Ele também reduz o tempo de sistema parado e, conseqüentemente, aumenta a produtividade (AUTOMATION, 2014c). A comunicação dessa IHM é via OPC. A interface pode ser visualizada na figura 4.

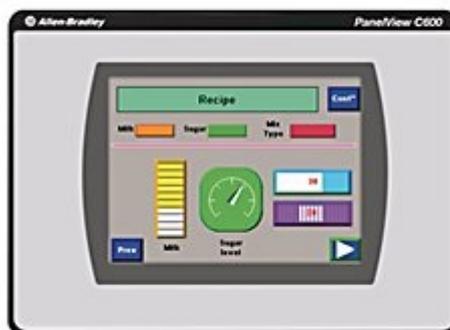
Figure 4 – PanelView™Plus 600



Fonte: (AUTOMATION, 2014c)

Já os PVC 600 podem ser utilizados para os usuários implementarem um software de programação incorporado e grampos de montagem integrados para ajudar a melhorar a produtividade e manutenção. Esse componente tem como melhor integração os CLP's Micro 820 (AUTOMATION, 2014b). A comunicação dessa IHM é por cabo ethernet ou pela saída serial. Neste trabalho, foi utilizado via ethernet. A interface pode ser visualizada na figura 5.

Figura 5 – PanelView™Component 600



Fonte: (AUTOMATION, 2014b)

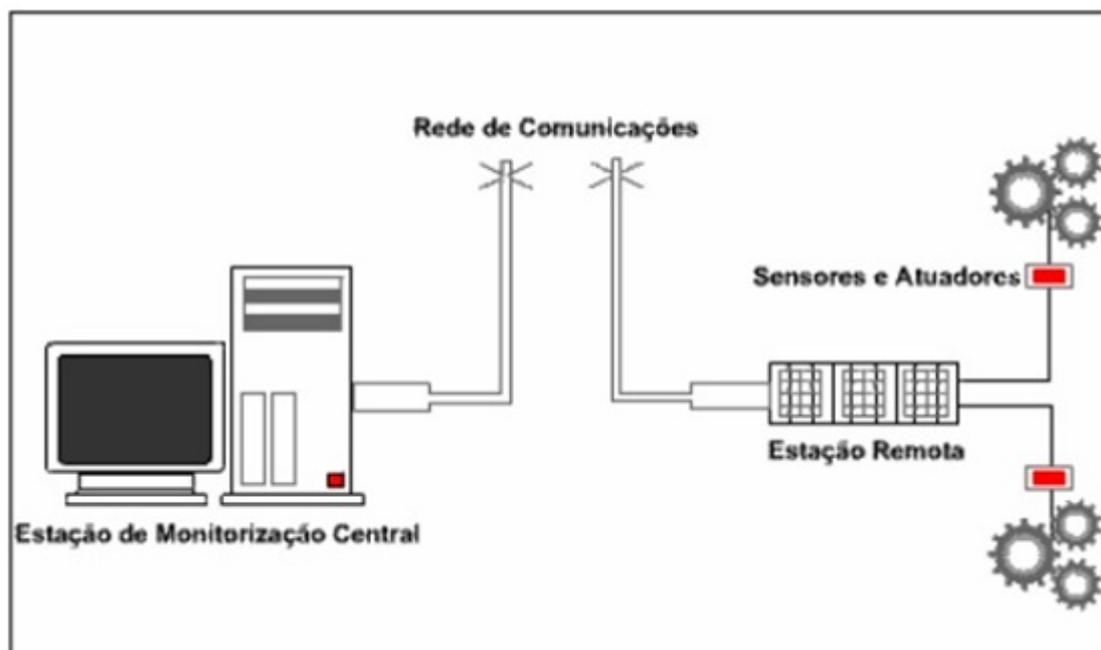
2.3 Sistemas supervisórios

Os sistemas supervisórios vieram da necessidade de rápida construção de modelos bem definidos dos sistemas de automação industrial complexos. Basicamente, um sistema supervisório destina-se à capturar e armazenar em um banco de dados informações sobre um processo de produção (JURIZATO; PEREIRA, 2003). Elas vêm por meio de sensores específicos para cada circunstância.

A utilização de sistemas supervisório oferece diversas vantagens, sendo uma das principais a análise de tendência, onde por meio das informações armazenada num banco de dados se torna possível para o usuário tomar ações para maximizar a produção, reduzindo consequentemente falhas, paradas de máquina entre outras perdas (BAILEY; WRIGHT, 2003).

Para termos um sistema supervisório, precisamos simplificadamente de sensores, atuadores, rede de comunicação, estações remotas e de monitoração central (aquisição e controle e sistema computacional, respectivamente), como podemos ver na figura 6. (SILVA; SALVADOR, 2005).

Figura 6 – Estrutura de um Sistema Supervisório



Fonte: (SILVA; SALVADOR, 2005).

Cada componente possui uma função específica:

- Sensores: são responsáveis para converterem os dados do processo, como a vazão de água num cano, e transformá-los em sinais analógicos e digitais, por meio das estações remotas (JURIZATO; PEREIRA, 2003).
- Atuadores: são os responsáveis para atuar sobre o sistema, como ligando ou desligando um equipamento específico (SILVA; SALVADOR, 2005).

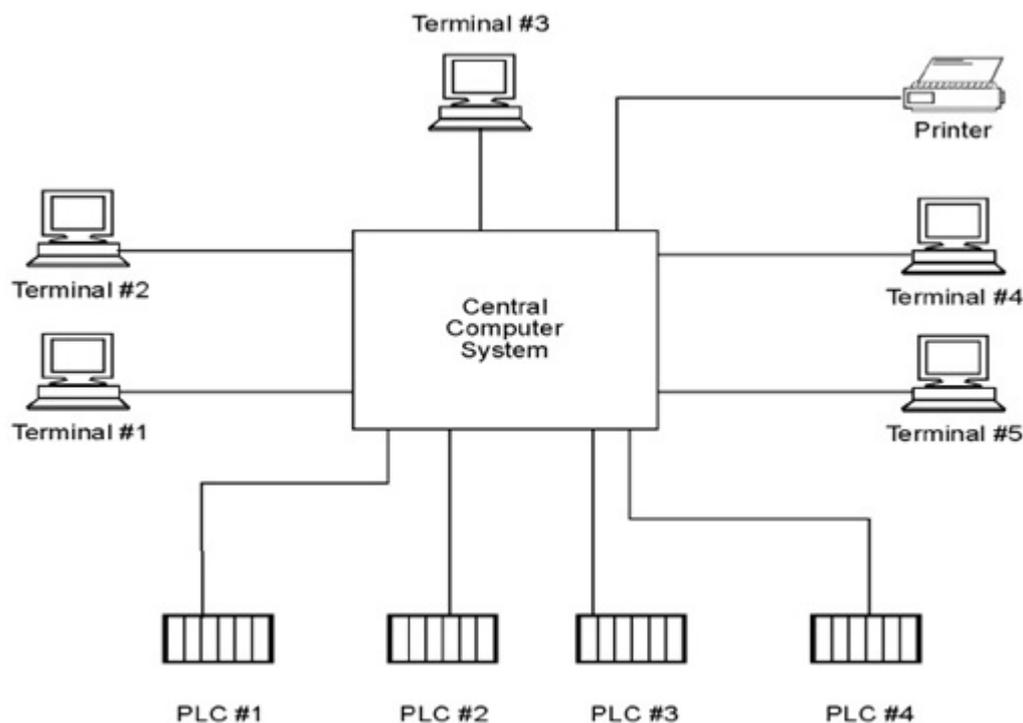
- Rede de comunicação: é a plataforma em que as informações vinda do CLP ou da unidade terminal remota (UTR) vão para o sistema supervisor. Elas podem ser implementadas via cabos Ethernet, rádio, modems, dependendo apenas dos requisitos do sistema a distância que o mesmo irá cobrir (SILVA; SALVADOR, 2005).

- Estações remotas, CLPs e UTRs: são responsáveis para a aquisição de dados, processo de controle, com a leitura de valores atuais dos dispositivos que a ele estão conectados e seu respectivo controle (SILVA; SALVADOR, 2005).

- Estações de monitoração central: unidade principal de um sistema supervisor, são responsáveis para obter as informações geradas pelas estações remotas e atuar em conformidade com os eventos detectados. Elas podem ser centralizadas ou distribuídas (JURIZATO; PEREIRA, 2003).

Quando a estação de monitoração é centralizada, todo o monitoramento e banco de dados da planta está guardado somente nesse computador ou mainframe. A principal vantagem desse tipo é a segurança da informação, uma vez que, por ser uma única estação, ela fornece alto grau de segurança e controle de cópias de segurança e recuperação. Porém, suas desvantagens são: custo inicial bem elevado para um sistema pequeno, o sistema tem que ter redundância, necessidade de mão de obra de manutenção especializada (BAILEY; WRIGHT, 2003). Podemos ver na figura 7 um exemplo de arquitetura de um processo centralizado.

Figura 7 – Arquitetura com a estação de monitoração centralizada

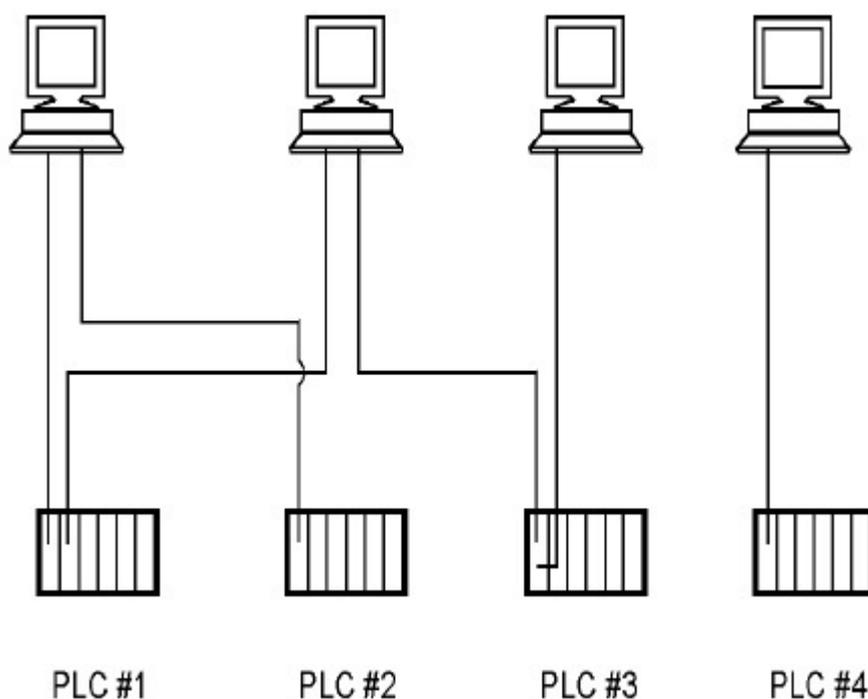


Fonte: (BAILEY; WRIGHT, 2003)

No caso de a monitoração ser distribuída, o sistema é compartilhado em vários compu-

tadores. As desvantagens que tínhamos no sistema centralizado, nesse tipo passa a ser as vantagens, como o baixo custo inicial e a não necessidade de redundância no sistema todo (BAILEY; WRIGHT, 2003). Suas desvantagens são: a comunicação entre diferentes computadores pode não ser simples, ocasionando problema de configuração, o processamento de dados e o banco de dados deve ser duplicado por meio de todos os computadores do sistema e não há uma abordagem sistemática para obter os dados dos dispositivos da planta (BOARETTO; KOVALESKI; SCANDELARI, 2004). A figura 8 demonstra um exemplo de sistema distribuído.

Figura 8 – Arquitetura com a estação de monitoração distribuída



Fonte: (BAILEY; WRIGHT, 2003).

2.3.1 Sistema supervisorio de controle e aquisição de dados (SCADA)

Um SCADA é um sistema formado por um número de UTRs que coletam os dados do processo e transmitem para a estação principal por meio de um sistema de comunicação. A estação principal exibe os dados adquiridos e também permite que o usuário execute tarefas de controle remoto (BAILEY; WRIGHT, 2003). O seu diferencial é a operação a distância ou remotamente.

Esses sistemas normalmente dividem suas principais tarefas em módulos que vão poder permitir maior ou menor robustez e flexibilidade, dependendo de qual seja a solução desejada (KALAPATAPU, 2004). De maneira geral, podemos dividir esses módulos em núcleo de processamento, comunicação com CLPs/UTRs, gerenciamento de alarmes, históricos e banco de dados, lógicas de programação interna (scripts) ou controle, interface

gráfica, relatórios, comunicação com outras estações SCADA, comunicação com sistemas externos/corporativos, outros (SILVA; SALVADOR, 2005). Neste trabalho foram utilizados os históricos, lógicas de programação internas, interface gráficas e relatórios.

O SCADA caracteriza-se por fazer a aquisição de dados, tornar os dados disponíveis visualmente, processar eventos e ativar alarme e ser tolerante a falhas. Essas características garantem a execução das atividades de supervisão, operação e controle, também conhecidas como as três atividades básicas, supervisão, operação e controle, de um sistema SCADA (SOUZA, 2005).

A principal função desses sistemas é a troca de dados com, basicamente:

- Comunicação com os CLPs e UTRs;
- Comunicação com outros sistemas SCADA;
- Comunicação com outros sistemas.

O primeiro tipo de comunicação, realizado por meio de um protocolo comum, pode ser tanto de acesso restrito ou de domínio público, e geralmente pode ocorrer por mestre-escravo (SILVA; SALVADOR, 2005).

O mestre-escravo é a filosofia mais simples de ser usada e pode ser pela topologia do tipo ponto-a-ponto. Nela, o mestre está em controle total do sistema de comunicação e faz pedidos, de maneira regular, por dados e para transferir dados, para todos os escravos. O escravo apenas responde a solicitação do mestre. Se, num período de tempo determinado, o escravo não responder, o mestre re-envia até atingir um número máximo de tentativas pré-estabelecido. Caso o escravo ainda não consiga responder, o mestre o considera como incomunicável e passa para o próximo nó escravo (BAILEY; WRIGHT, 2003).

A comunicação com outros sistemas SCADA se dá por meio de um protocolo desenvolvido pelo próprio fabricante do sistema ou por um protocolo conhecido via rede Ethernet TCP/IP (SILVA; SALVADOR, 2005).

Já a comunicação com outros sistemas, como por exemplo outros coletores e/ou fornecedores de dados, pode ser por meio da implementação de módulos específicos nos bancos de dados ou outras tecnologias como o XML e o OPC (SOUZA, 2005).

O XML é uma recomendação para gerar linguagens de marcação para necessidades especiais. Ele é capaz de descrever diversos tipos de dados e seu objetivo principal é a facilidade de compartilhamento de informações por meio da internet (MELLO, 2003). Já a tecnologia OPC é uma interface padronizada de comunicação que foi criada na tentativa de minimizar os problemas relacionados à inconsistência dos "drivers" de equipamentos industriais de diferentes fabricantes.

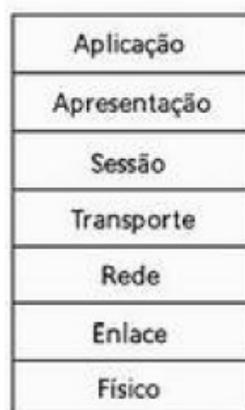
No presente trabalho, foi utilizado o protocolo TCP/IP para a comunicação entre o Micro820 e a PVC 600 e entre o Micro820 e o CompactLogix L24ER. Já a tecnologia utilizada para a comunicação entre o CompactLogix L24ER e a PVP 600 foi a OPC.

2.4 Redes industriais

As redes industriais vieram com o intuito de compartilhamento de recursos e seu objetivo é tornar todos os programas, equipamentos e principalmente dados ao alcance de todos os usuários da rede (TANENBAUM, 2003). Essas redes utilizam protocolos de comunicação digital padronizado, permitindo a comunicação entre vários equipamentos independente dos fabricantes.

A arquitetura das redes industriais é por meio de camadas. Uma arquitetura desse tipo permite discutir e analisar uma parcela específica de um sistema grande e complexo. Sua vantagem é que ela provê modularidade, permitindo com que se torne mais fácil modificar a implementação da camada. Isso é possível desde que a camada alterada forneça os mesmos dados para a camada superior e necessite das mesmas informações da camada inferior (KUROSE; ROSS, 2006). Podemos ver na figura 9 as camadas de uma rede industrial, no modelo OSI (Open System Interconnection).

Figure 9 – Camadas de uma rede industrial



Fonte: (KUROSE; ROSS, 2006).

- Camada física: define as interfaces mecânica, elétrica e de sincronização para a rede (TANENBAUM, 2003). Sua tarefa é de movimentar os bits individuais que estão dentro de um nó para o seguinte. Os protocolos dessa camada depende tanto da camada de enlace quanto do próprio meio de transmissão (KUROSE; ROSS, 2006);

- Camada de enlace de dados: executa várias funções específicas: fornece uma interface bem definida à camada de rede, lida com os erros de transmissão e regula o fluxo de dados para evitar que receptores lentos não sejam atropelados por transmissores rápidos (TANENBAUM, 2003). Esses serviços dependem do protocolo específico que é empregado no enlace (KUROSE; ROSS, 2006);

- Camada de rede: responsável pela movimentação de pacotes (conhecidos como datagramas) de um equipamento para outro. Ela tem dois componentes principais: o protocolo que define os campos no datagrama e o modo como os roteadores e os sistemas finais agem nesses campos (KUROSE; ROSS, 2006);

- Camada de transporte: é o núcleo de toda a hierarquia de protocolos. Promove uma transferência de dados confiável e econômica entre a origem e o destino, independente das redes físicas em uso (TANENBAUM, 2003);
- Camada de sessão: responsável pela sincronização e pela delimitação da troca de dados. Isso inclui os meios de construção um esquema de pontos de verificação e recuperação (KUROSE; ROSS, 2006);
- Camada de apresentação: responsável em prover serviços que as aplicações de comunicação entendam o significado dos dados trocados. A compreensão de dados e a sua codificação fazem parte dos serviços dessa camada (KUROSE; ROSS, 2006);
- Camada de aplicação: onde as aplicações de rede e seus protocolos residem. Um protocolo dessa camada é distribuído em diversos sistemas finais, sendo que sua aplicação nesses sistemas utiliza o protocolo para trocar os datagramas com a aplicação em outro sistema final (KUROSE; ROSS, 2006).

3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados alguns trabalhos desenvolvidos, encontrados na literatura científica, relacionados à metodologia proposta nesta monografia.

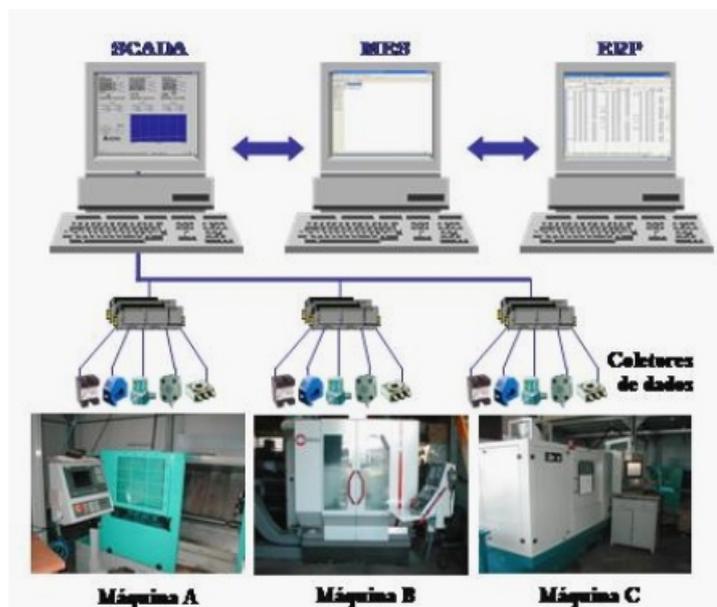
3.1 Estudo da integração entre sistemas SCADA, MES e ERP em empresas de manufatura discreta que utilizam processos de usinagem

O trabalho de Mardegan, Martins e OLIVEIRA (2003) tem como objetivo a análise da integração de sistemas supervisórios SCADA, Sistema de Execução da Manufatura (MÊS) e Planejamento de Recursos Empresariais (ERP) com o intuito de ter uma aquisição de dados mais confiável entre chão de fábrica e o restante da empresa.

A problemática do trabalho corresponde ao fato das informações imprecisas de manufatura discreta, ou seja, no monitoramento da produção baseado no apontamento manual das informações. Além disso, outro fator citado é a demora na atualização de status da operação. Porém, em sistemas mais críticos do negócio, não se pode perder tempo e as tomadas de decisões devem ser feitas em questões de segundos ou, no máximo, poucos minutos. Isso se deve ao fato de que, cada minuto parado, o processo gera um prejuízo elevado para a empresa.

Paralelo a necessidade de monitoramento em tempo real dos processos, vem surgindo a necessidade da utilização de sistemas de gestão para integração entre os processos envolvidos. Para isso, têm-se utilizado o sistema ERP. Tendo um sistema de controle e aquisição de dados (SCADA) no chão de fábrica e um sistema de planejamento de recursos (ERP), veio a necessidade de se estabelecer um nível intermediário entre eles. Com isso, segundo Corrêa, Giansi e Caon (2001), os sistemas MES destinam-se a aumentar a dinâmica dos sistemas de planejamento da produção. A figura 10 ilustra a arquitetura de integração.

Figura 10 – Arquitetura de Integração



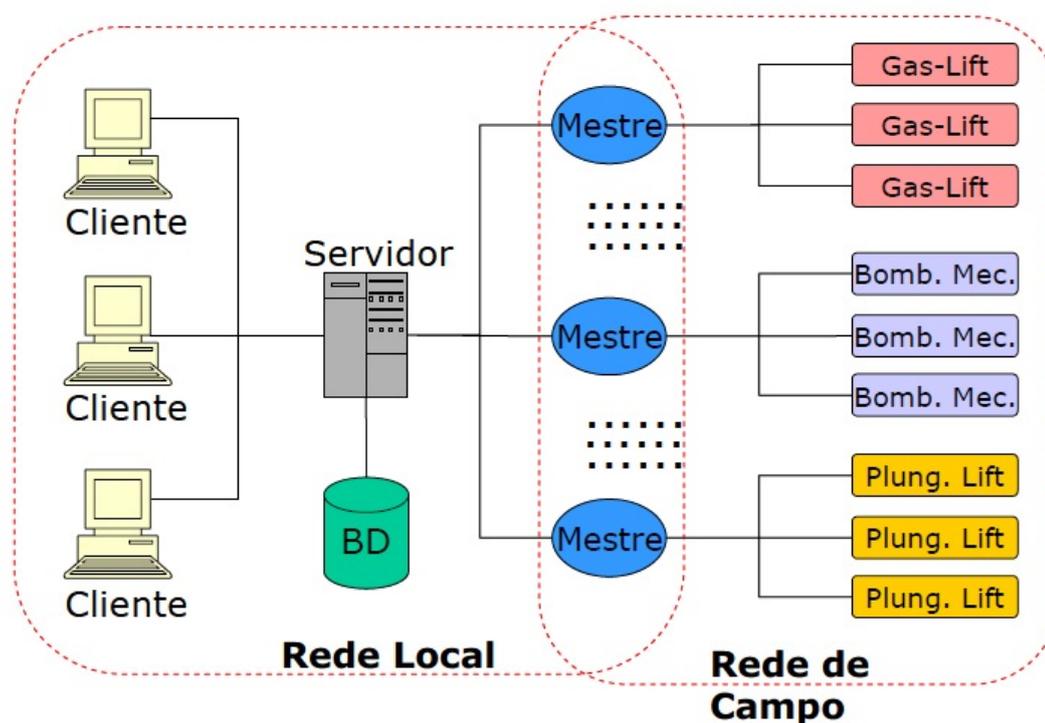
Fonte: (MARDEGAN; MARTINS; OLIVEIRA, 2003).

Segundo os autores, a proposta da integração dos três sistemas vem da necessidade cada vez mais nítida das empresas integrarem todos os processos de negócios e os maiores responsáveis são os sistemas de planejamento de recursos. Além disso, outra intenção era mostrar que os sistemas ERP não conseguem contemplar com detalhes os eventos que ocorrem no chão de fábrica.

3.2 SISAL - Um sistema supervisório para elevação artificial de petróleo

O trabalho de Souza et al. (2006) tem como objetivo apresentar um sistema supervisório que, independentemente da técnica de elevação artificial sendo utilizada e dos controladores e dispositivos de conexão com os poços, é capaz de coletar dados dos processos e entregá-los ao sistema de gerenciamento. Este sistema é composto de três aplicações: cliente, servidor e mestre. a figura 11 ilustra a arquitetura do mesmo.

Figura 11 – Arquitetura do SISAL



Fonte: (SOUZA et al., 2006).

O cliente é responsável por transmitir, por meio de uma interface gráfica amigável, as informações dos poços para os usuários. O servidor deve rotear as requisições dos clientes para um dos mestres de comunicação (conectado ao poço cuja informação se está solicitando). O mestre é dedicado ao controlador por traduzir um pedido de informações de alto nível dos clientes para um acesso a determinada região de memória do controlador. Essa arquitetura faz com que seja possível monitorar poços de diferentes métodos de elevação utilizando um único software de supervisão.

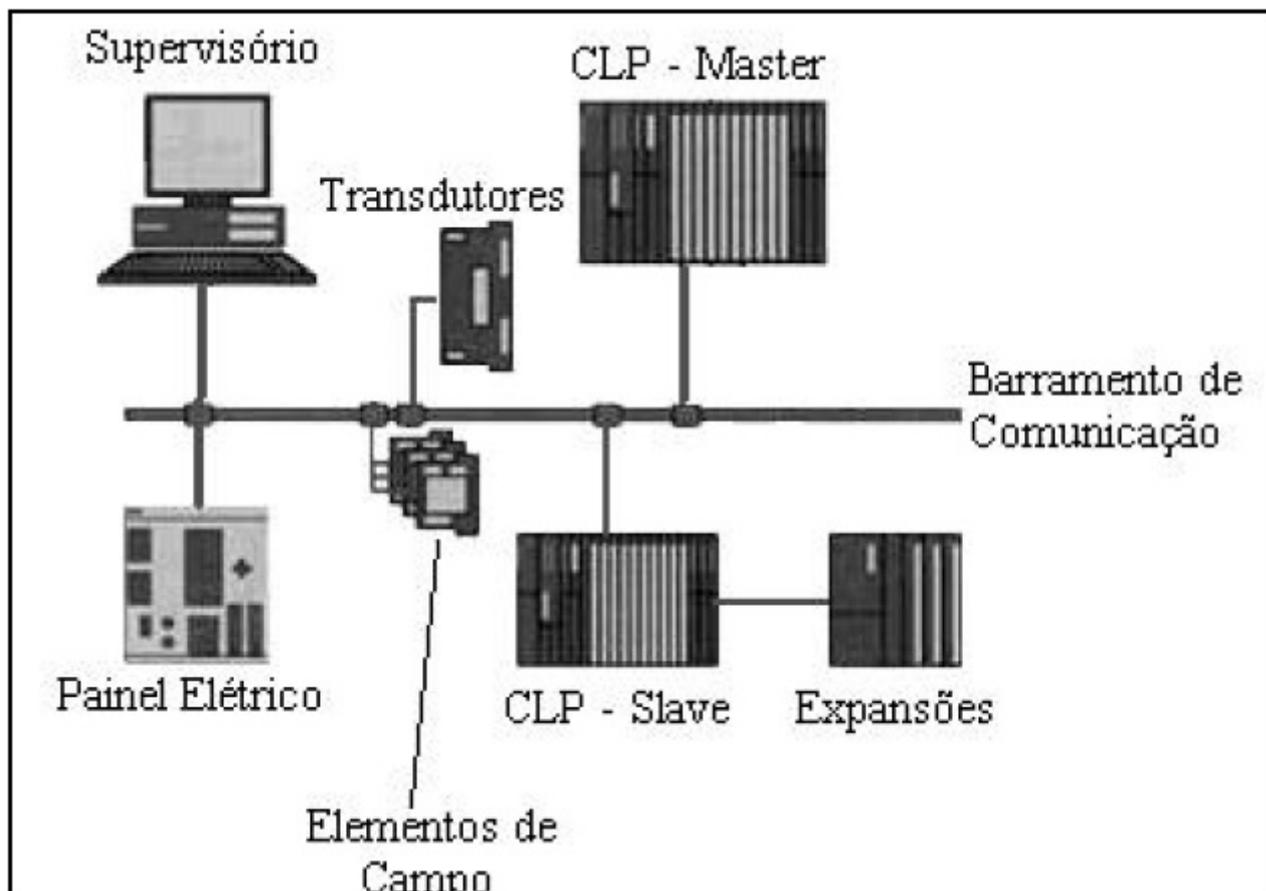
No sistema SISAL somente a aplicação mestre deve conhecer os detalhes de comunicação com as estações escravas. Logo, o acesso aos dados pelo cliente será feito de forma transparente e independente do hardware utilizado na rede de campo. Esse sistema possui algumas vantagens como vários usuários simultâneos, acesso remoto, vários mestres em um único servidor e vários clientes.

3.3 Sistemas supervisórios

No artigo apresentado por Jurizato e Pereira (2003), tem-se como objetivo apresentar características, como telas gráficas e esquema básico do sistema, e ambientes de alguns dos mais utilizados sistemas supervisórios do mercado, implementados, principalmente, em plantas industriais.

As aplicações de controle de processos podem ser encontradas em todas as empresas em que existe a necessidade de um controle rígido de processos envolvidos. Os Sistemas Supervisórios podem ser vistos como sistemas que supervisionam ou monitoram processos executados em uma planta industrial, por meio da visualização de parâmetros da planta que está sendo automatizada. A figura 12 ilustra um exemplo de esquema básico de um sistema supervisório.

Figura 12 – Exemplo de Esquema Básico



Fonte: (JURIZATO; PEREIRA, 2003).

As características técnicas de um sistema supervisório, baseado nos softwares Elipse Windows e WinCC (Windows Control Center), dois dos mais empregados no Brasil, atendem a praticamente todas as necessidades de implantação e dispõem de uma biblioteca ampla de objetos. Algumas dessas características são:

- Arquitetura cliente-servidor;
- Comunicação por meio da OPC;
- Programação orientada a Objeto;
- Segurança por meio de redundância de arquivos;
- Funções de HMI/SCADA avançadas.

A construção dos sistemas podem ser diretamente aplicados na planta industrial, a partir de sua modelagem, têm sua aplicação imediata. O emprego desses sistemas tornou-se

possível graças à programação orientada a objetos (OOP) e ao desenvolvimento de CLPs com custo reduzido e arquiteturas flexíveis.

A integração do TCP/IP nos hardwares dos CLPs estimula e abre as possibilidades de uso de um sistema supervisório e a monitoração via Intranet e Internet por meio da construção das telas gráficas já é possível e realidade em muitas plantas industriais.

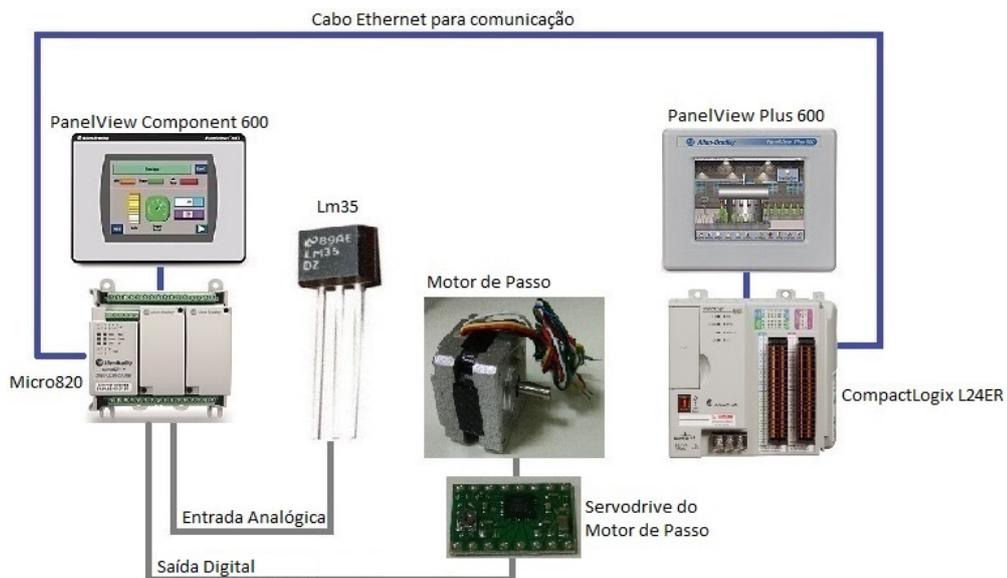
4 METODOLOGIA

Para o trabalho em questão foi desenvolvido um sistema supervisorio com o objetivo de monitorar e controlar um processo industrial simulado em bancada no laboratório de instrumentação industrial.

Para o desenvolvimento do sistema supervisorio, foi utilizado os CLPs Micro820 e CompactLogix L24ER, as IHMs PVC C600 e PVP 600, ambos da fabricante Allen Bradley®, e a rede industrial TCP/IP. Para a programação do CLP e da IHM, foram utilizado os *softwares* do fabricante, o Connected Components Workbook®, Studio 5000® e o FactoryTalk View Studio®.

4.1 Arquitetura do sistema proposto

Figura 13 – Arquitetura do Sistema Proposto.



Fonte: Autor

Como pode-se visualizar na figura 13, o sistema é dividido em dois subsistemas: a estação do operador e a estação central de monitoramento. A estação operador representa o chão de fábrica em que o usuário apenas visualiza, por meio da IHM, o parâmetro do processo industrial: qual o limite de operação, o valor atual e se o atuador está acionado ou não. Neste caso, o parâmetro é a temperatura. Já a estação central de monitoramento é a estação num nível hierárquico acima em que o usuário controla o limite de operação dos parâmetros e é nela que fica o sistema supervisorio em si.

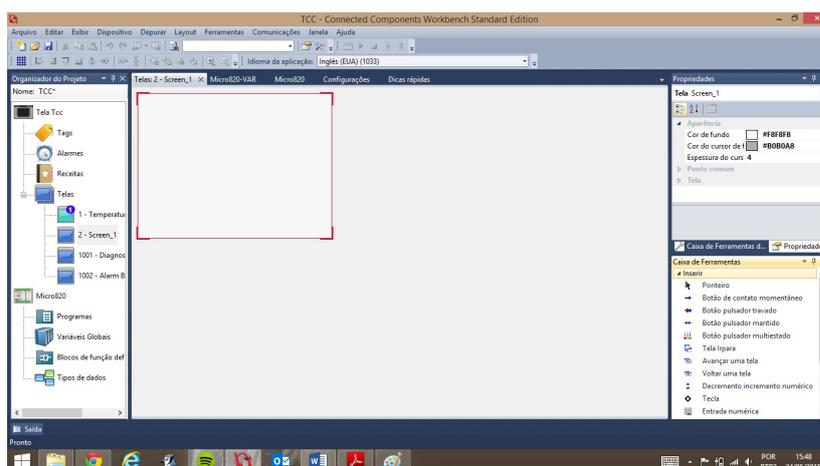
4.2 Estação do operador

Nesta estação estão localizados um controlador lógico programável micro820, uma interface humano-máquina PVC 600, sensor de temperatura LM35 e o conjunto atuador constituído por um motor de passo e um *driver* para acionamento do mesmo.

4.2.1 Interface humano-máquina PanelView™ Component 600

Para o desenvolvimento da interface humano-máquina, foi utilizado a interface humano-máquina (IHM) PanelView™ c600, da Allen-Bradley®. O mesmo possui um próprio ambiente para a elaboração de suas telas, acessando por meio do seu IP no browser, ou pode utilizar o *software* Connected Component Workbench® (CCW) para a elaboração das mesmas. A tela foi desenvolvida por meio do *software* CCW uma vez que o mesmo também faz a parte de programação do CLP e a interação entre ambos é facilitada. A figura 14 ilustra seu ambiente de trabalho.

Figure 14 – Ambiente de trabalho do Connected Component Workbench

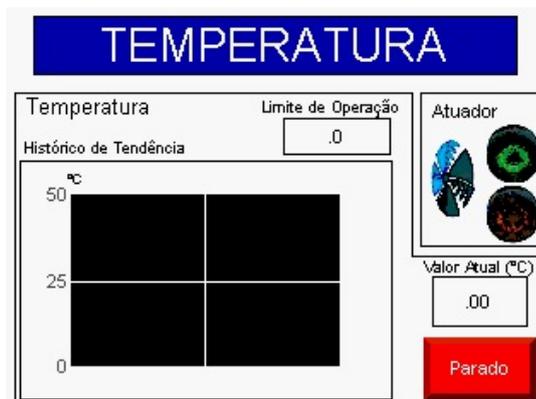


Fonte: Autor

Para ser estabelecida a comunicação com a IHM, foi definido um IP estático e uma máscara de sub-rede para a mesma no próprio terminal. Nas suas configurações, no menu de comunicação, foi desativado o DHCP e estabelecido o seu IP. O IP estático selecionado foi 192.168.1.21.

Considerando o parâmetro a ser controlado indiretamente (temperatura) e o atuador para atingir os resultados (motor de passo), foi desenvolvida uma tela onde pode-se acompanhar o comportamento da temperatura em um determinado instante, o limite de operação estabelecido pelo sistema central de monitoramento e o acionamento do atuador. Essa tela é ilustrada na figura 15.

Figure 15 – Tela da IHM - Parâmetro Temperatura



Fonte: Autor

O gráfico de histórico de tendência gera para saída Universal Serial Bus (USB) um arquivo em excel de extensão .csv com os dados de valor da variável temperatura, data e hora que foi obtido o dado, no intervalo de 1 segundo. Com isso, o usuário consegue fazer uma análise melhor da variável sendo controlada por sua estação. A figura 16 ilustra como os dados ficam armazenados no arquivo em excel.

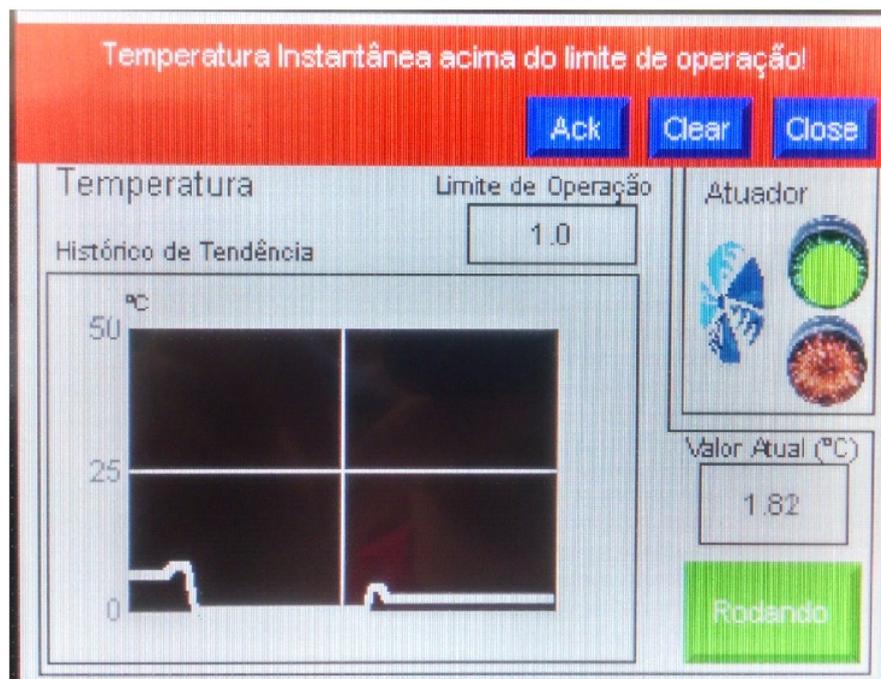
Figure 16 – Relatório gerado pelo gráfico do PVC 600

	A
2	10:32:37 AM,5/28/2015,11.230469
3	10:32:38 AM,5/28/2015,9.765625
4	10:32:39 AM,5/28/2015,11.230469
5	10:32:40 AM,5/28/2015,11.71875
6	10:32:41 AM,5/28/2015,11.230469
7	10:32:42 AM,5/28/2015,10.742188
8	10:32:43 AM,5/28/2015,11.230469
9	10:32:44 AM,5/28/2015,10.742188
10	10:32:45 AM,5/28/2015,10.742188

Fonte: Autor

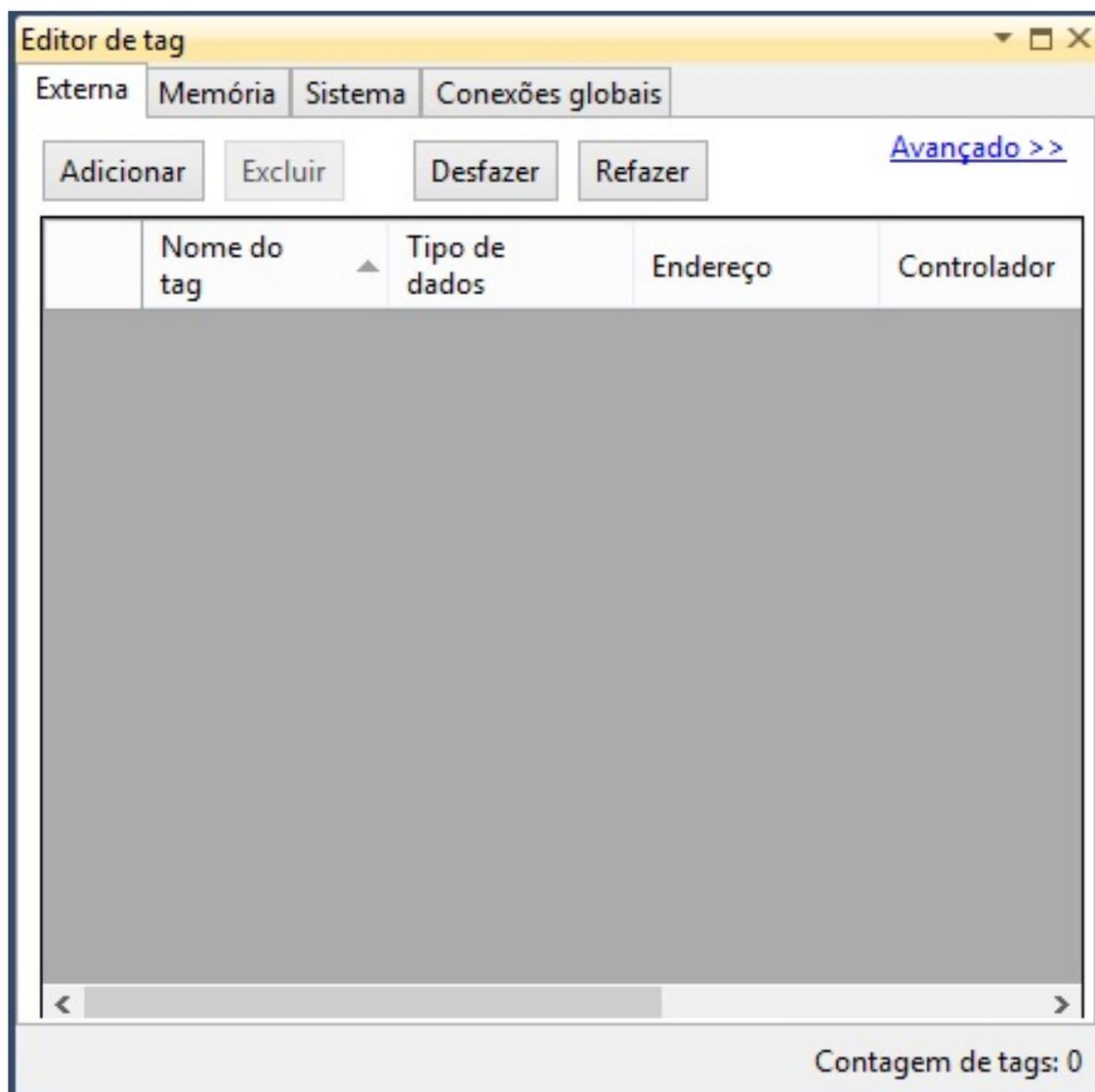
Devido a limitação de hardware, não foi desenvolvida uma tela com as informações dos alarmes ocorridos. O alarme estabelecido está relacionado à *tag* atuador, citada mais abaixo. O mesmo indica que a temperatura real está maior que o limite de operação. A figura ilustra o alarme acionado.

Figura 17 – Alarme acionado no sistema



Fonte: Autor

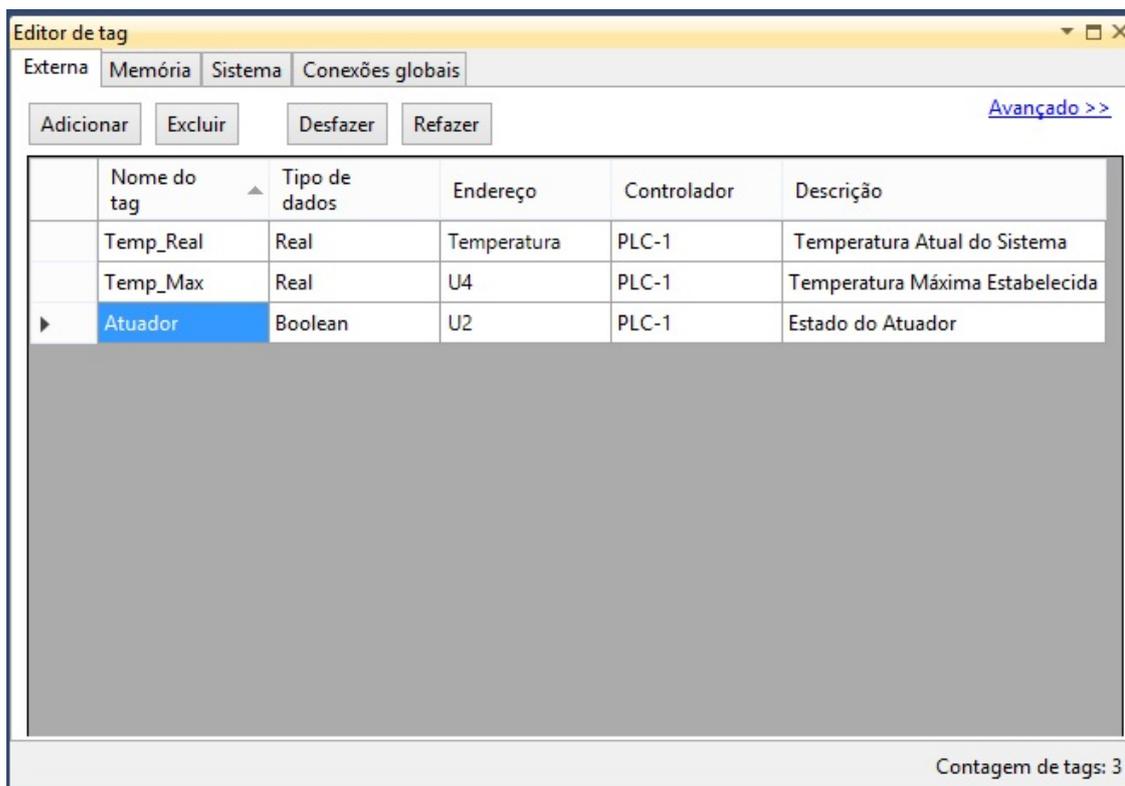
Desenvolvida a tela, foram criadas as *tags* de leitura de temperatura máxima, temperatura real e do Atuador. Essas *tags* são apenas de leitura, já que essa tela não recebe entrada de dados. Para isso, foi aberto o editor de *tags*, onde pode ser adicionado *tags* externas. Esse editor é ilustrado na figura 18.

Figure 18 – Editor de *Tags*

Fonte: (ROCKWELL AUTOMATION, 2015)

As *tags* criadas para essa tela foram a de temperatura real, para o histórico de tendência e para o valor atual, temperatura máxima, para o limite de operações, e atuador, para estabelecer quando o atuador está ligado ou desligado. As *tags* criadas são ilustradas na figura 19.

Figure 19 – Tags criadas para a IHM da Estação Operador



	Nome do tag	Tipo de dados	Endereço	Controlador	Descrição
	Temp_Real	Real	Temperatura	PLC-1	Temperatura Atual do Sistema
	Temp_Max	Real	U4	PLC-1	Temperatura Máxima Estabelecida
▶	Atuador	Boolean	U2	PLC-1	Estado do Atuador

Fonte: Autor

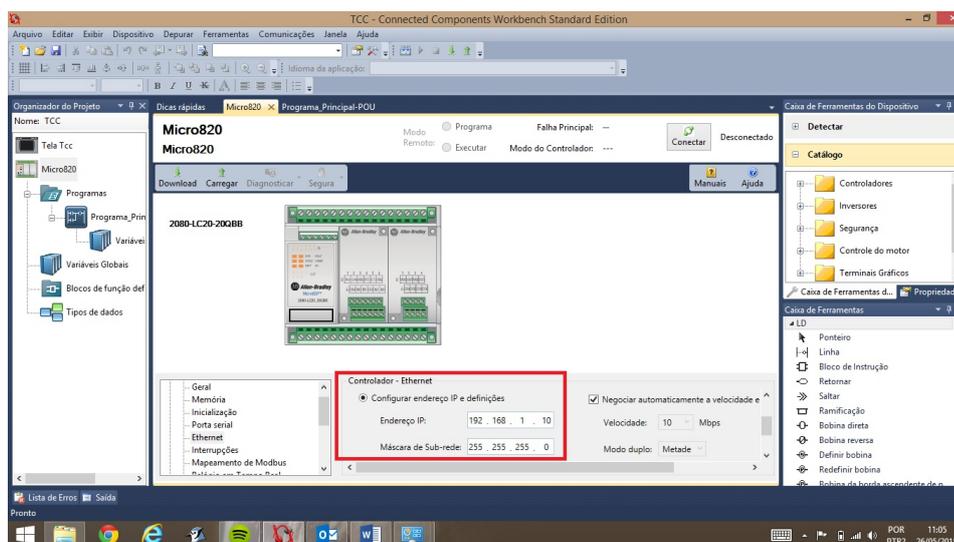
Os endereços das *tags* definidos e o controlador são mostrados no presente trabalho na parte de comunicação entre o CLP e o IHM nesse mesmo capítulo. Os tipos de dados definidos foram do tipo real, para as *tags* de temperatura, e booleana para a *tag* do atuador.

4.2.2 Controlador lógico programável Micro820

Realizada a tela da IHM e com o sensor e atuador funcionando, o próximo passo foi o desenvolvimento da programação do CLP micro820. Foi utilizado o mesmo *software* do desenvolvimento da tela da IHM (CCW).

Para estabelecer a comunicação com o CLP, primeiro foi definido um IP estático e a máscara da sub-rede para a porta ethernet do computador. O IP escolhido foi o 192.168.1.10 e a máscara 255.255.255.0. Realizado isso, no *software* CCW, foi estabelecido na configuração do CLP seu IP estático, como é visualizado na figura 20.

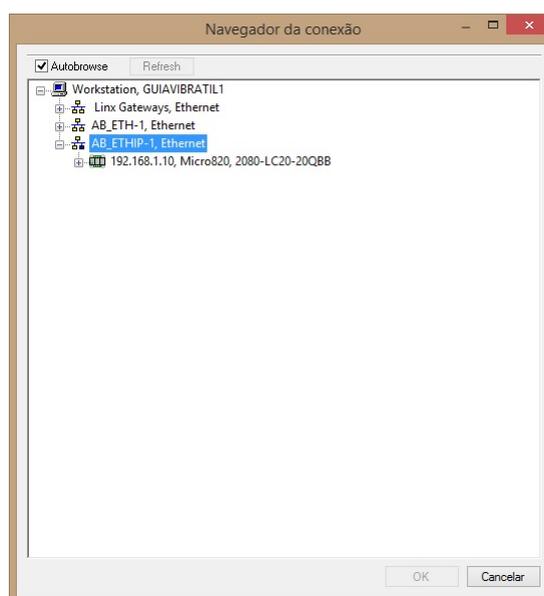
Figura 20 – Configurando IP e Máscara da Sub-Rede



Fonte: Autor

Estabelecido isso, deve-se então criar e configurar o *driver* de comunicação. Essa criação e configuração pode ser desenvolvida no próprio *software* CCW. Na barra superior, no menu de comunicação tem o item “configurar *driver*”. Foi adicionado o “Ethernet/IP *driver*”, uma vez que iremos fazer a comunicação via ethernet, utilizando o IP. Tendo esse *driver*, pode-se estabelecer a comunicação com o CLP. A figura 21 ilustra o computador reconhecendo o CLP com o IP configurado.

Figura 21 – Reconhecimento da conexão entre PC e PLC

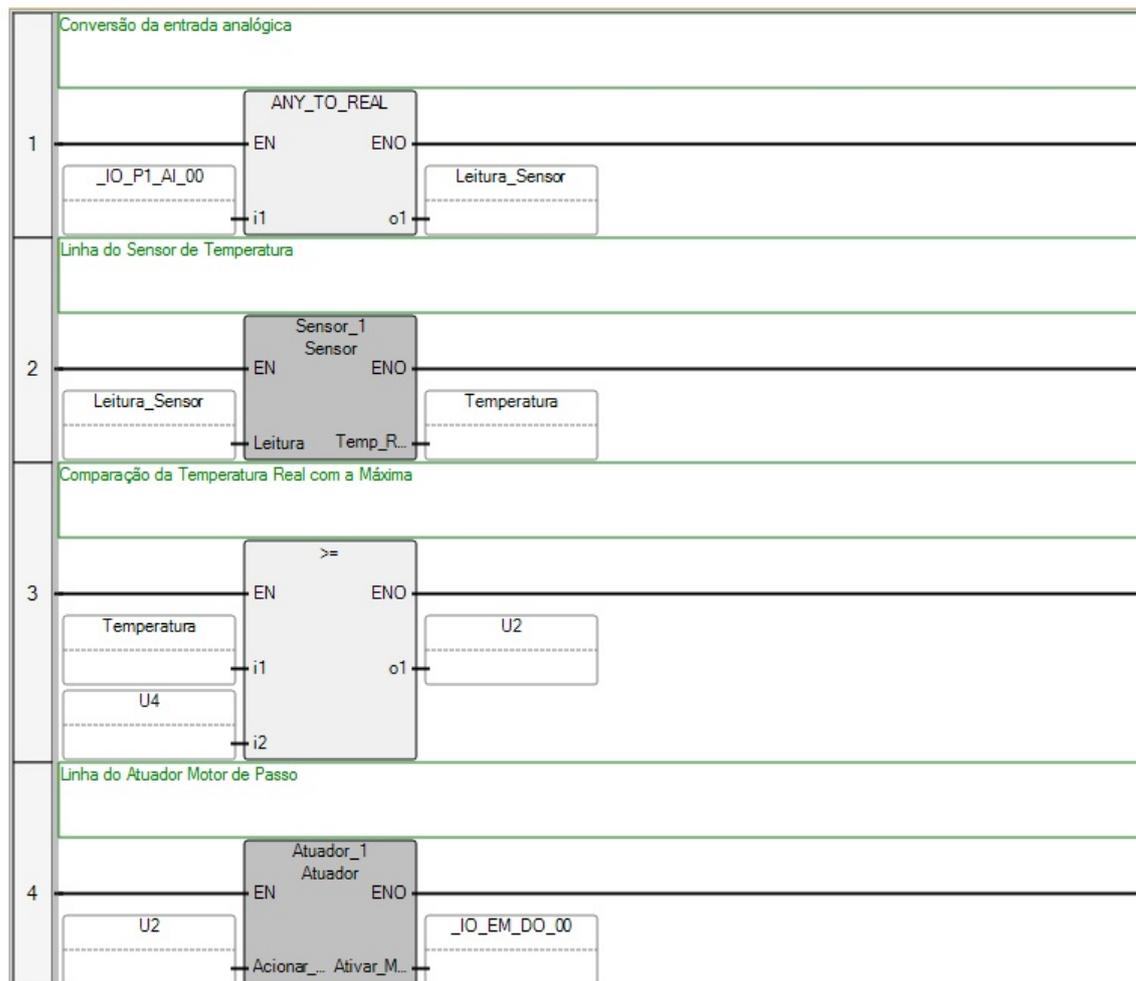


Fonte: Autor

Com a conexão estabelecida, pode-se então fazer a programação do CLP. Essa programação deve conter a recepção do sinal proveniente do sensor de temperatura, a interpretação desse sinal, a comparação com o limite de temperatura e o acionamento do atuador para

retornar a temperatura para o limite de operação. Utilizando os blocos já mostrados do sensor de temperatura e do atuador, a programação realizada é ilustrada na figura 22.

Figura 22 – Programação da Estação Operador



Fonte: Autor

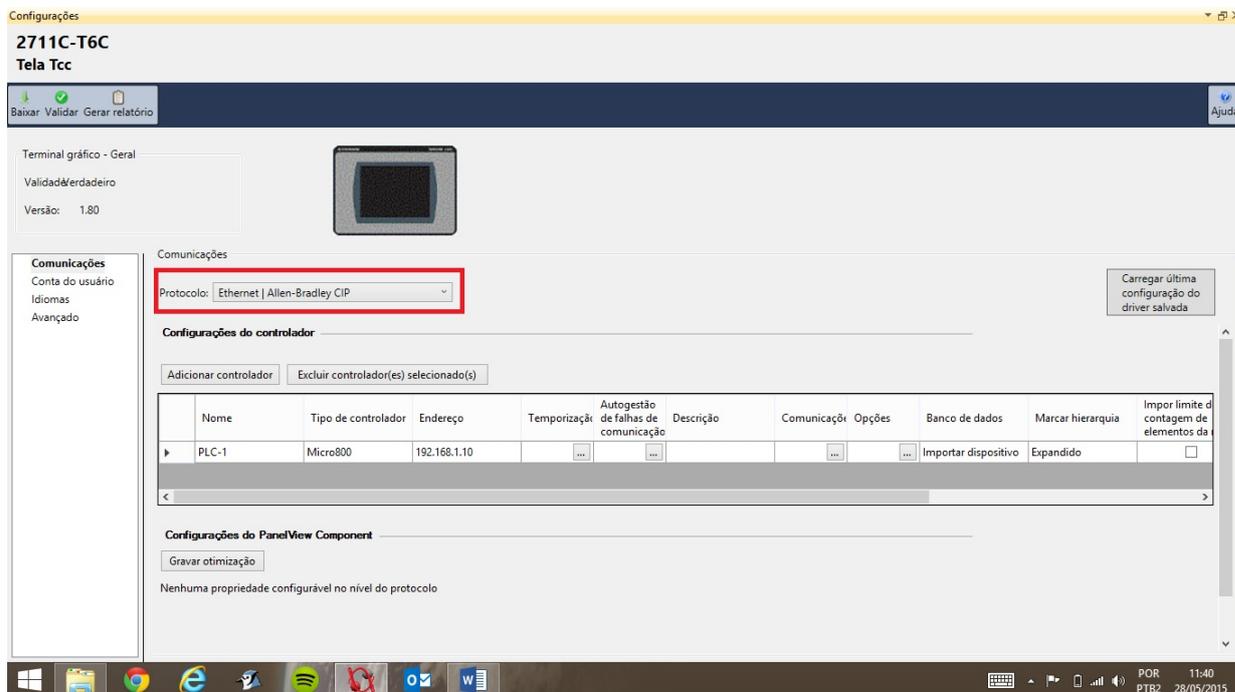
Além desses dois blocos, foram utilizados também um bloco de conversão (na primeira linha), para converter a entrada analógica da unidade “UINT” (Unsigned 16 bit integer) para real, e um bloco de comparação, para verificar se a temperatura atual é maior ou igual a temperatura máxima (representada pela variável U4). Essa temperatura máxima é definida na estação central de monitoramento. Caso a temperatura real seja maior ou igual a máxima, ele aciona a variável U2 que é o acionamento do bloco do motor de passo.

4.2.3 Comunicação entre CLP e IHM

Depois que foram estabelecidos os IP estáticos da IHM e do CLP, pode ser estabelecida a comunicação entre ambos por meio da configuração do CLP, utilizando o *software* CCW. O primeiro passo é estabelecer o protocolo de comunicação: serial ou ethernet. No caso do projeto, foi selecionado o ethernet. Definido o protocolo, foi estabelecido então o tipo da

comunicação que o CLP micro820 realiza, sendo denominado o Allen-Bradley® CIP. A figura 23 ilustra o protocolo selecionado no ambiente de comunicação.

Figura 23 – Protocolo de Comunicação Selecionado



Fonte: Autor

Realizado isso, foi inserido os dados de *configuração* do CLP para a IHM. Para isso, foi colocado no campo de endereço do CLP o IP estático do mesmo. Com isso, quando o programa é executado, se houver alguma *tag* dependente do CLP, a IHM automaticamente tenta conectar com o mesmo via o protocolo estabelecido (no caso, ethernet).

Após estabelecida a conexão entre o CLP e a IHM, foram endereçadas as *tags* apropriadamente. Para estabelecer um endereço para as mesmas, tem-se que garantir que as variáveis as quais serão referenciadas estejam na parte de variáveis globais do CLP. As *tags* da IHM não se comunicam com variáveis locais de blocos de função ou do próprio programa principal.

Considerando que as *tags* utilizadas na IHM são referentes ao acionamento do motor, a temperatura real e o limite de operação, estabelecido pela estação central de monitoramento, os parâmetros devem ser o valor de saída do bloco de função do atuador (`_IO_EM_DO_00`), o valor da saída do bloco do sensor de temperatura (*tag* de nome "Temperatura") e a variável U4, respectivamente. Definidos os parâmetros, pode-se então endereçar as *tags*, conforme a figura 24.

Figura 24 – Tags da IHM endereçadas

	Nome do tag	Tipo de dados	Endereço	Controlador	Descrição
	Temp_Real	Real	Temperatura	PLC-1	Temperatura Atual do Sistema
	Temp_Max	Real	U4	PLC-1	Temperatura Máxima Estabelecida
▶	Atuador	Boolean	U2	PLC-1	Estado do Atuador

Fonte: Autor

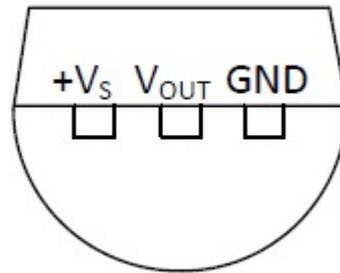
4.2.4 Processo industrial simulado

Apenas para efeitos de teste do sistema, foram utilizado um sensor de temperatura LM35 e um motor de passo AK17 funcionando como atuador, visando que o foco do projeto não é desenvolver o processo a ser monitorado.

4.2.4.1 Sensor de Temperatura LM35

Para o desenvolvimento do processo industrial simulado, foi utilizado o sensor de temperatura lm35 para aferir a temperatura ambiente. A mesma foi utilizada como o parâmetro do processo a ser controlado. Para isso, foi realizado o estudo do sensor por meio da sua documentação (*datasheet*). O mesmo trabalha na faixa de temperatura de -55 até 150 °C, com um erro de 0,25 °C para mais ou para menos. Esse sensor capta a temperatura e sua saída é uma tensão, em que, a cada 10 milivolts na sua saída representa 1 °C. A figura 25 ilustra o sensor e seus pinos.

Figura 25 – Sensor de Temperatura LM35

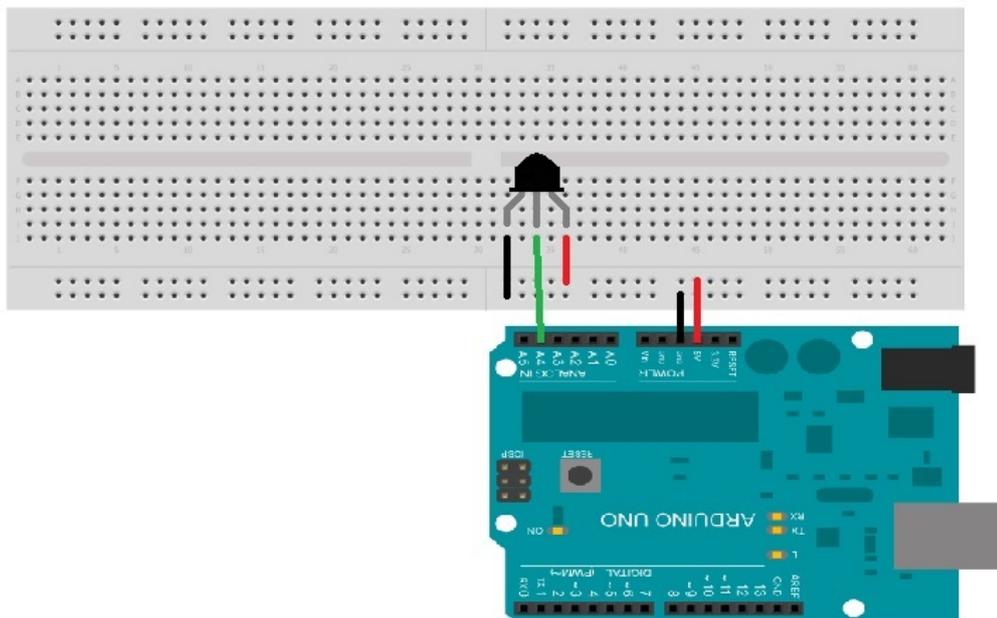


Fonte: (TEXAS INSTRUMENTS, 2013)

O pino +V_s é a tensão de entrada do sensor. Ele pode ser alimentado entre -0.2 volts até 35 volts. O pino V_{out} é o pino de saída em que a tensão de saída varia entre -1 e 6 volts. O pino GND é o terra. Todos esses dados foram adquiridos no seu datasheet. O mesmo pode ser encontrado no site do fabricante.

Para efeito de teste de uso, foi desenvolvido um circuito em *protoboard*, com uma plataforma Arduino Uno. A figura 26 ilustra como foi montado o circuito.

Figura 26 – Circuito teste do Sensor de Temperatura



Fonte: Autor

Como se trata de uma variável analógica, o pino do meio do sensor foi ligado na porta analógica de número 4 (A4). Tanto a alimentação quanto o aterramento utilizados foram os disponíveis no próprio arduino (fios vermelhos e pretos, respectivamente). A tensão de alimentação utilizada foi de 5 volts. Utilizando o compilador do arduino, foi desenvolvido o trecho de código representado na figura 27 com objetivo de adquirir o dado lido pela entrada analógica do arduino Uno.

Figura 27 – Primeira programação para leitura do sensor LM35

```
int reading;
int tempPino=4;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  reading=analogRead(tempPino);
  Serial.print("Valor lido pelo Sensor de Temperatura = ");
  Serial.println(tempPino);
  delay(500);
}
```

Fonte: Autor

Porém o valor obtido não representa nem a temperatura nem a tensão. Para obter-se a temperatura por meio da leitura, teve-se que fazer com que uma transformação. Baseando-se nas informações do sensor, devemos multiplicar o valor lido pela tensão aplicada no sensor (em milivolts), dividir por dez, uma vez que a cada 10 milivolts temos 1 °C, e pegar esse resultado e dividir por 1024 para obtermos o percentual da faixa. Com isso, a programação ficou conforme a figura 28.

Figura 28 – Segunda programação para leitura do sensor LM35

```
float tempC;
int reading;
int tempPino=4;

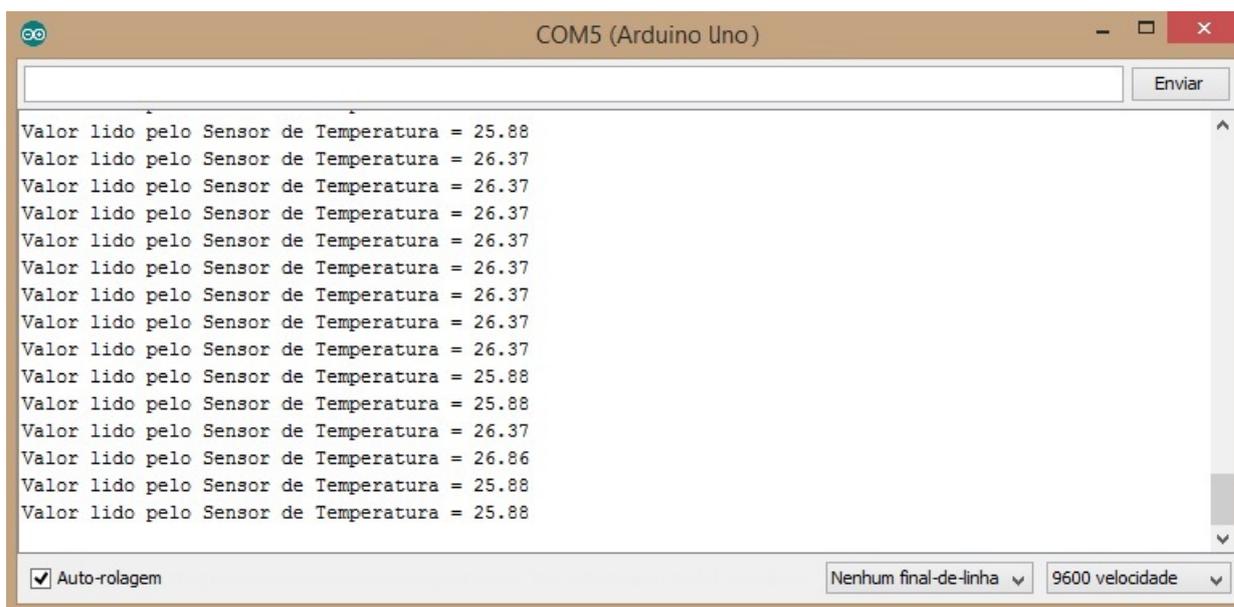
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  reading=analogRead(tempPino);
  tempC=(5.0*reading*100)/1024;
  Serial.print("Valor lido pelo Sensor de Temperatura = ");
  Serial.println(tempC);
  delay(500);
}
```

Fonte: Autor

Dessa forma, foi possível receber no monitor serial do compilador do arduino dados em forma de temperatura em graus celsius, como pode-se ver na figura 29.

Figura 29 – Monitor Serial com sensor de temperatura



Fonte: Autor

Finalizado o teste com o arduino, o próximo passo foi desenvolver um bloco de função em linguagem ladder para este sensor ser utilizado em um CLP. Para isso, foram criadas variáveis globais e variáveis locais. Por se tratar de um bloco de função, deve-se ter as entradas e saídas nas variáveis locais. Considerando a fórmula de conversão utilizada nas programações, foram criadas as constantes de valor 500 e também 1024, uma variável temporária U1, a variável de entrada Leitura e a variável de saída Temp_Real. A figura 30 ilustra as variáveis locais criadas para o bloco de função.

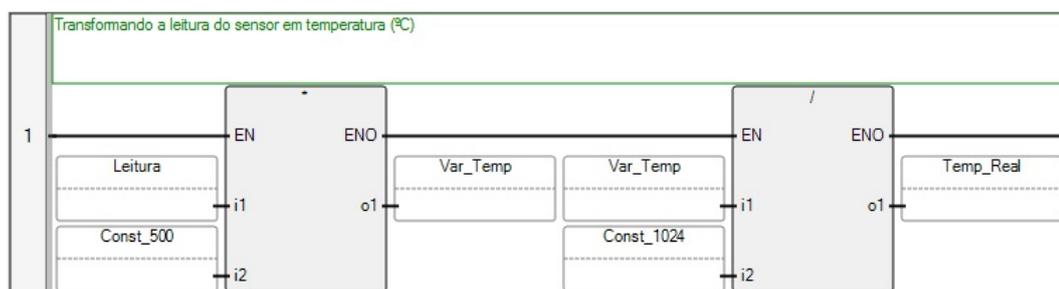
Figura 30 – Variáveis Locais para Sensor de Temperatura

	Nome	Tipo de dado	Direção	Atributo	Aliás	Dimensão	Valor inicial
	Leitura	REAL	VarInput	Ler			
	Temp_Real	REAL	VarOutput	Gravar			
	U1	REAL	Var	Ler/gravar			
	Const_500	REAL	Var	Ler/gravar			500.0
	Const_1024	REAL	Var	Ler/gravar			1024.0
*							

Fonte: Autor

Esse bloco tem como função fazer a transformação da tensão lida para temperatura em graus celsius. Ele foi programado no *software* Connected Component Workbench®, do fabricante do CLP utilizado, Allen-Bradley®, e sua programação é ilustrada na figura 31.

Figura 31 – Programação do Bloco de Função do Sensor



Fonte: Autor

Tal bloco de função contém dois blocos aritméticos. O primeiro é de multiplicação, onde a entrada i1 “Leitura” é a variável Real que indica o valor lido no sensor e a entrada i2 “Const_500” é uma constante de valor 500 (5000 mV dividido por 10 mV/°C), tendo como saída o1 “Var_Temp” que é uma variável temporária com o valor da multiplicação das duas entradas. O segundo é de divisão, tendo como entrada i1 “Var_Temp” resultado do bloco anterior e entrada i2 “Const_1024”, uma constante de valor 1024, e tendo como saída o1 “Temp_Real” que é o valor da temperatura real.

4.2.4.2 Motor de passo AK17 e driver a4988

Para o desenvolvimento do processo industrial simulado, foi utilizado o motor de passo acoplado a uma ventoinha como atuador para reduzir a temperatura. Ele é bipolar e sua faixa de trabalho é ilustrado na figura 32.

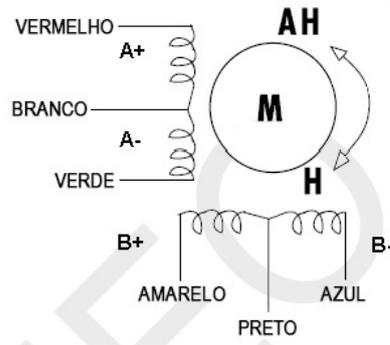
Figure 32 – Faixa de Trabalho do Motor de Passo

NEMA	Conexão	Holding torque (kgf.cm)	Corrente (mA/fase)	Tensão (V/fase)	Resistência (Ω/fase)	Indutância (mH/fase)
17	Bipolar – Série	1.10	70	16.8	140	148
	Unipolar	0.77	100	12	70	37

Fonte: (NEOYAMA, 2015)

Esse motor tem um ângulo de passo de 1,8°, ou seja, 200 passos para fechar uma volta completa. Ele possui 6 fios (vermelho, branco, verde, amarelo, preto e azul), sendo os fios branco e preto representando o terra e os demais fios como atuadores de bobina. As figuras 33 e 34 ilustram o esquema de ligação e a sequência de fases, respectivamente.

Figura 33 – Esquema de Ligação



Fonte: (NEOYAMA, 2015)

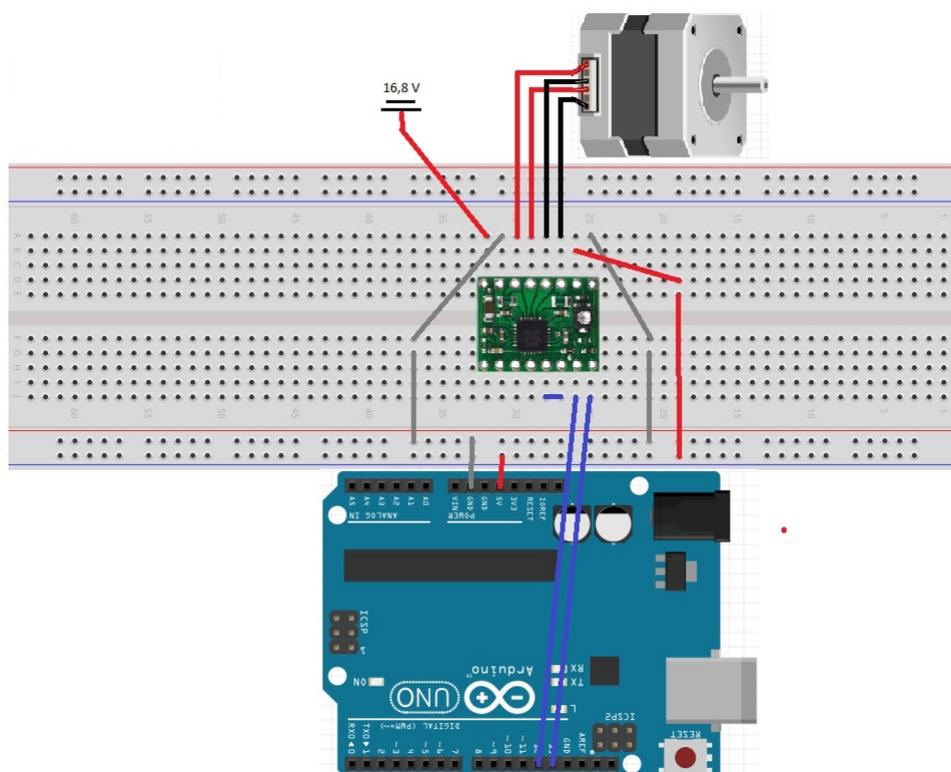
Figura 34 – Sequência de Fases

	Passo	A+	B-	A-	B+	
Sentido: horário	0	+	+			Sentido: anti-horário
	1		+	+		
	2			+	+	
	3	+			+	

Fonte: (NEOYAMA, 2015)

Foi utilizado o drive de motor de passo a4988. Esse drive precisa ser alimentado com uma tensão de 5 volts e possui um pino para a alimentação do motor de passo, além das saídas das bobinas. Ele possui também os pinos de direção e habilitação do motor. Essa conexão é ilustrada na figura 35.

Figura 35 – Conexão do driver a4988 e do motor de passo



Fonte: Autor

Apenas para efeito de teste de funcionamento, foi utilizada a plataforma arduino Uno com uma programação simples, mandando um pulso ao pino “Step” para manter o motor girando numa frequência de 1 kHz. Além disso, a cada 7 voltas completas (1400 passos), o motor vai esperar meio segundo e começar a girar no sentido oposto. A programação utilizada para esse teste é ilustrada na figura 36.

Figura 36 – Primeira Programação para motor de passo

```

int dirPino = 5;
int stepperPino = 4;
void setup() {
  pinMode(dirPino, OUTPUT);
  pinMode(stepperPino, OUTPUT);
}
void step(boolean dir, int steps) {
  digitalWrite(dirPino, dir);
  delay(10);
  for (int i = 0; i < steps; i++) {
    digitalWrite(stepperPino, HIGH);
    delayMicroseconds(500);
    digitalWrite(stepperPino, LOW);
    delayMicroseconds(500);
  }
}
void loop() {
  step(true, 1400); //girar no sentido anti-horário
  delay(500);
  step(false, 1400); //girar no sentido horário
  delay(500);
}

```

Fonte: Autor

Realizado o teste e obtido resultado conforme determinado na programação, foi desenvolvido um bloco de função em linguagem ladder para o motor de passo. Para isso, as variáveis de entrada, Acionar_Bloco, e saída, Ativar_Motor, foram criadas. Além delas, foram criadas duas variáveis, U3 e U5, para fazer o chaveamento de pulso na saída digital. A figura 37 ilustra as variáveis locais criadas para o bloco de função.

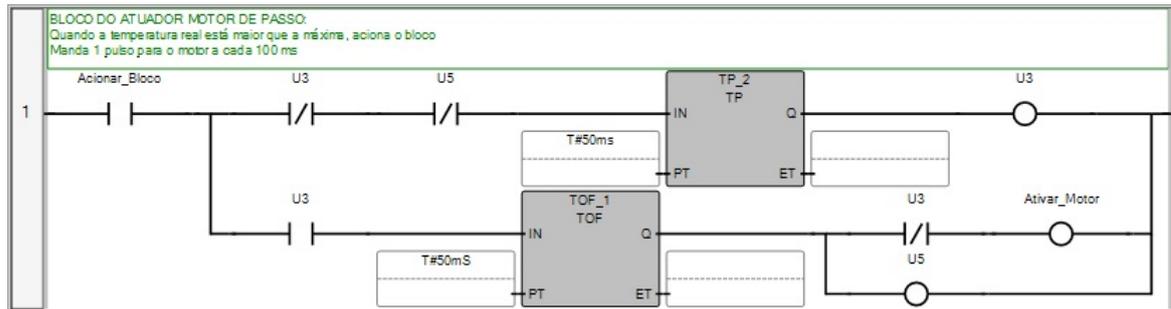
Figura 37 – Variáveis Locais para Atuador

Atuador-VAR				
	Nome	Tipo de dado	Direção	Atributo
	Acionar_Bloco	BOOL	VarInput	Ler
	Ativar_Motor	BOOL	VarOutput	Gravar
	U3	BOOL	Var	Ler/gravar
	U5	BOOL	Var	Ler/gravar
*				

Fonte: Autor

Para acionar o bloco, é necessário comparar a temperatura real com a temperatura máxima estabelecida pelo usuário. Ele foi programado no *software* Connected Component Workbench® e sua programação é ilustrada na figura 38.

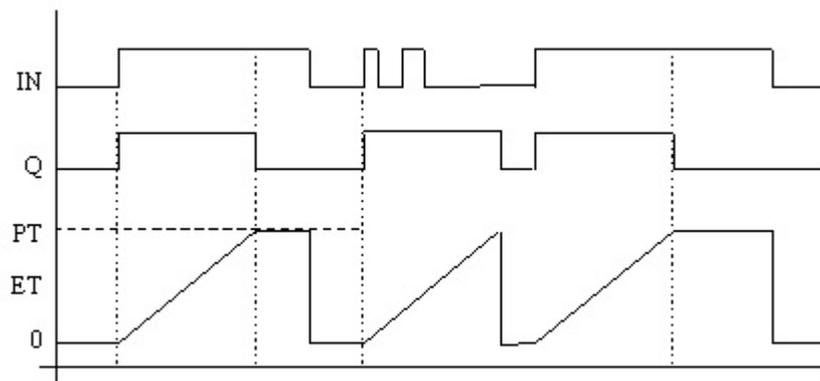
Figura 38 – Segunda Programação para motor de passo



Fonte: Autor

Como o motor de passo deve ficar recebendo pulsos numa frequência e não continuamente, foi realizado um jogo com os blocos temporizadores TP (*Pulse Timer*) e TOF (*Timer Off Delay*). O primeiro bloco só precisa de um pulso na entrada e ele manda um sinal lógico “1” na saída até chegar no tempo estabelecido. O pino “IN” é sua entrada, o “Q” sua saída, “PT” é o tempo estabelecido e o “ET” é o tempo estimado. Seu diagrama é ilustrado na figura 39.

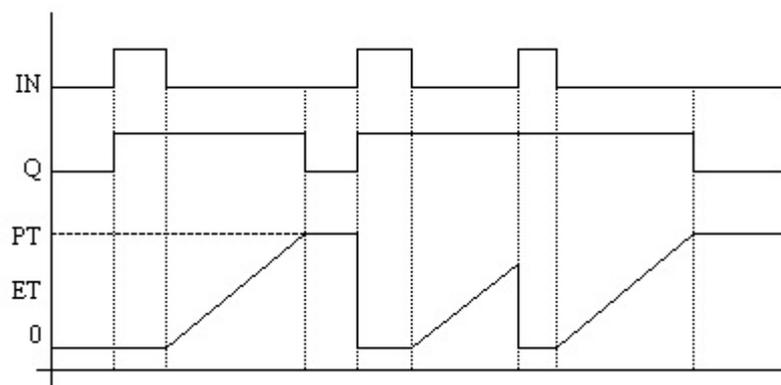
Figura 39 – Diagrama do temporizador TP



Fonte: (ROCKWELL AUTOMATION, 2015)

O segundo temporizador, TOF, recebe uma alimentação na entrada e, assim que essa alimentação cessa, ele libera um sinal digital “1” na saída pelo tempo estipulado no pino “Q”. Seu diagrama é ilustrado na figura 40 .

Figura 40 – Diagrama do temporizador TOF



Fonte: (ROCKWELL AUTOMATION, 2015)

Com a utilização desses dois blocos, pode-se ser realizado a variação entre saídas “1” e “0” para mandar pulsos até o motor de passo. Isso foi necessário porque o CLP utilizado não possui um bloco para ficar alternando a saída entre “1” e “0” quando a entrada do mesmo está acionada. A frequência estabelecida para o envio de pulso é de 100 Hz. Com isso, a cada 10 milisegundos, o motor de passo dá um passo de $1,8^\circ$.

Como a fonte do CLP micro820 é de 24 volts, foi necessário a utilização de um circuito integrado (CI), LM7805, regulador de tensão de 24 volts para 5 volts. Com isso, na saída do CI a tensão é reduzida para 5 volts. Assim, o *driver* do motor de passo pode-se ser conectado a saída digital do CLP, acionando o motor.

4.3 Estação central de monitoramento

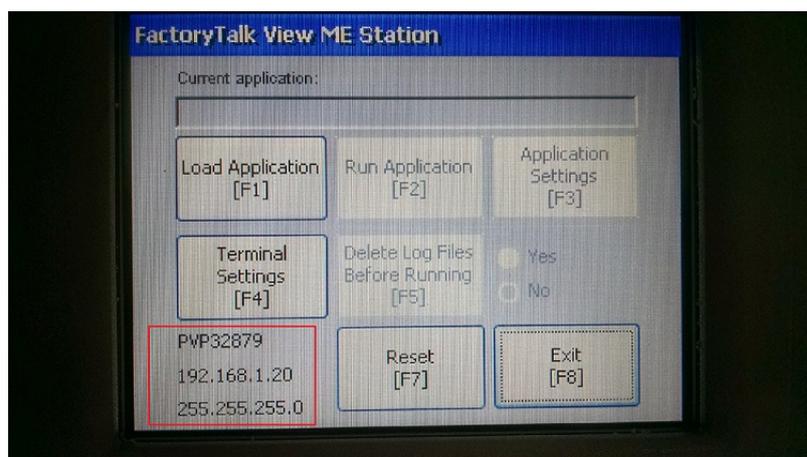
Nesta estação estão localizados um controlador lógico programável CompactLogix L24ER e uma interface humano-máquina PVP 600. Nessa estação que é definido o limite de operação do sistema.

4.3.1 Interface humano-máquina PanelView™ Plus 600

Para o desenvolvimento da interface humano-máquina, foi utilizado a interface humano-máquina (IHM) PanelView™ Plus 600, da Allen-Bradley®. Por se tratar da estação central de monitoramento, sua interface permite a configuração do limite de operação. Para o desenvolvimento da mesma, foi utilizado o *software* compatível à IHM, FactoryTalk View Studio®, o qual já dispõe de OPC próprio, pelo qual é estabelecida a comunicação.

O primeiro passo foi estabelecer a comunicação entre computador e a IHM. Para isso, na própria IHM foi estabelecido o valor de IP estático e máscara de sub-rede com o objetivo de evitar problemas, ou até queda de comunicação devido a IP duplicados. O IP estático estabelecido foi 192.168.1.20 e a máscara de sub-rede estabelecida foi 255.255.255.0. A figura ilustra a IHM com o IP estático e a máscara de sub-rede configurada.

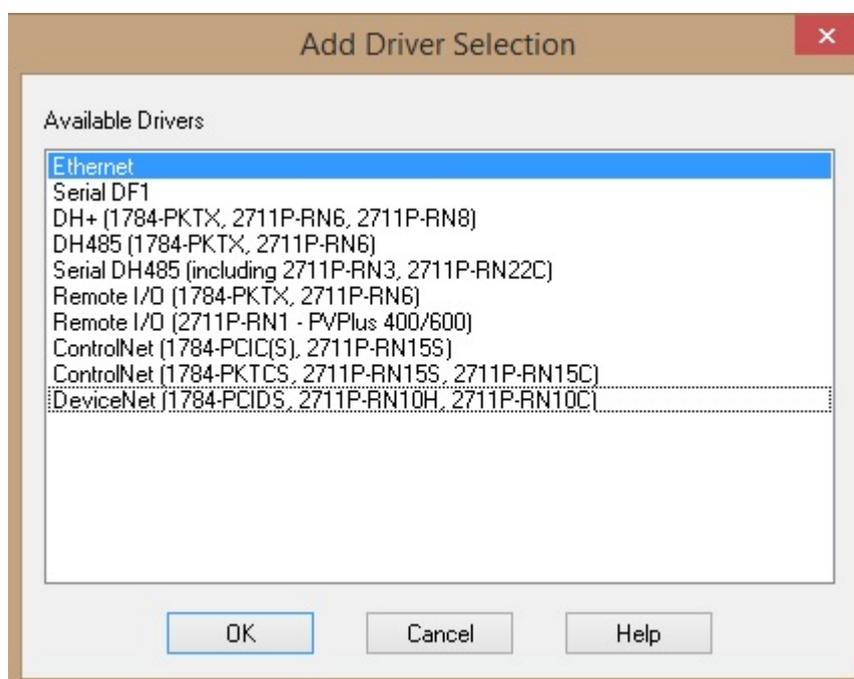
Figura 41 – PVP 600 com IP estático



Fonte: Autor

Para ser realizada a comunicação com a IHM por meio do *software* próprio, foi adicionado um novo *driver* ao OPC. O *driver* define qual o protocolo de comunicação que será utilizado. No caso do trabalho, foi adicionado um *driver* ethernet, com o nome da interface utilizada, para estabelecer a comunicação entre computador e IHM. A figura 42 ilustra os *drivers* disponíveis para seleção e o *driver* de ethernet selecionado.

Figura 42 – Drivers de comunicação via OPC

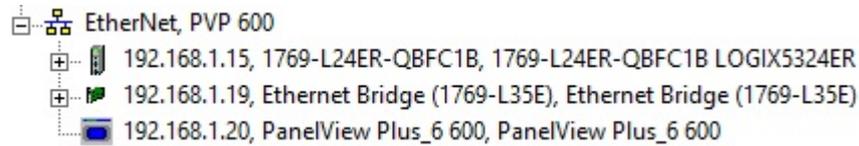


Fonte: Autor

Feito isso, foram adicionados dois dispositivos ao *driver*: a IHM PanelView™ Plus 600 e uma ponte ethernet, com IP estático de 192.168.1.19. Essa segunda foi adicionada para permitir a transmissão de dados via ethernet com a IHM, como *download* e *upload* de

programas. Como pode-se ver na figura 43, têm-se três dispositivos. O terceiro dispositivo refere-se ao CLP CompactLogix L24ER e é explicado na subsecção 4.3.2.

Figura 43 – Dispositivos do driver Ethernet PVP 600



Fonte: Autor

Com isso, pôde-se estabelecer a comunicação entre computador e a IHM e consequente transferência de dados entre os mesmos. Então foi desenvolvida a interface gráfica por meio do *software*. Essa interface possui três telas: inicial, de temperatura e de configurações.

A tela inicial possui dois botões, temperatura e configurações, que vão para suas respectivas telas. Além disso, ela contém algumas informações, como a universidade em que foi desenvolvido, e o curso. Percebe-se na tela inicial, da figura 44, um botão de configurações, identificado por engrenagens mecânicas, que permite o acesso às configurações da IHM. Tal acesso permite mudança de IP, inutilização da IHM, entre outras características necessárias para o funcionamento adequado do sistema. Essa tela é ilustrada na figura 44.

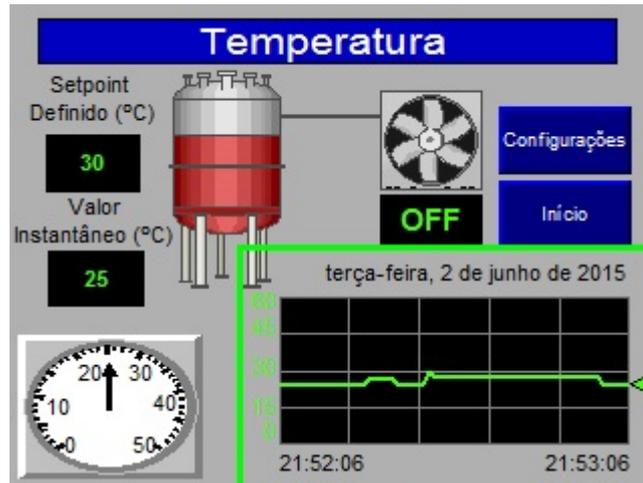
Figura 44 – Tela Inicial do programa desenvolvido no PVP 600



Fonte: Autor

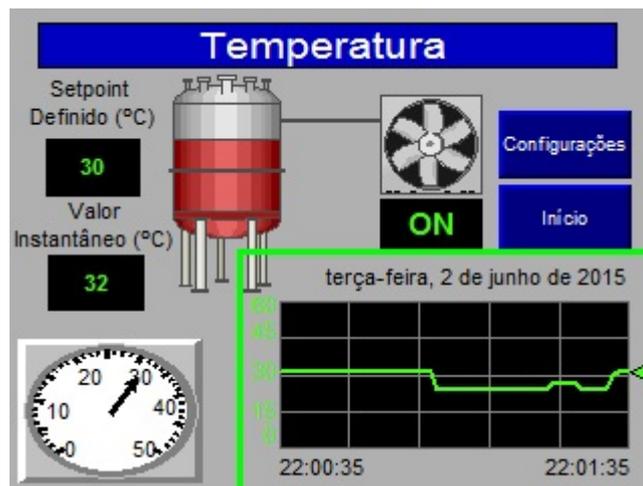
Na tela de temperatura é monitorada a variável. Nela têm-se os valores de setpoint definido pelo usuário, valor instantâneo e se o atuador está ou não ativado. As figuras 45e 46 ilustram dois casos: um quando o valor instantâneo está menor que o setpoint, ilustrando que o atuador está desativado, e outro quando o valor instantâneo está maior que o setpoint, ilustrando o atuador ativado.

Figura 45 – Tela de Temperatura com Atuador Desativado



Fonte: Autor

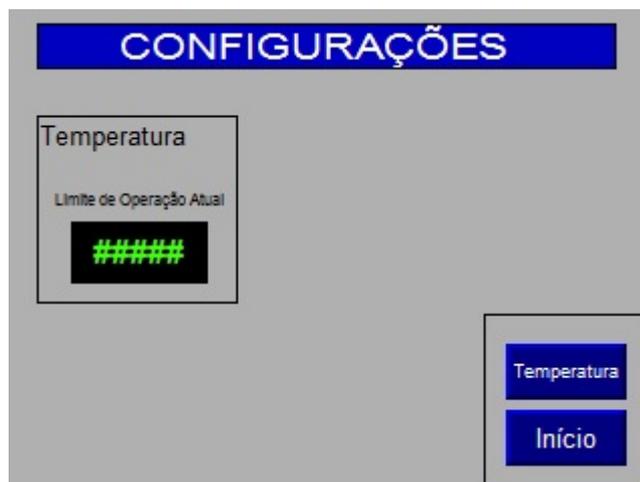
Figura 46 – Tela de Temperatura com Atuador Ativado



Fonte: Autor

A tela de configurações permite que o usuário da estação central de monitoramento possa estabelecer o limite de operação para cada variável do processo. Como na simulação foi utilizada a temperatura, somente o limite de operação da mesma está disponível, conforme a figura 47.

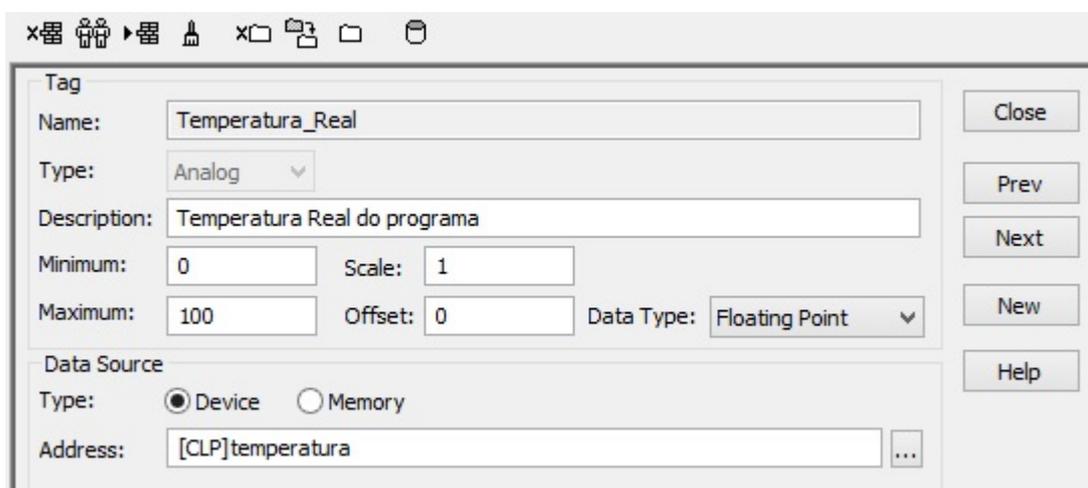
Figura 47 – Tela de Configuração



Fonte: Autor

Com as telas prontas, foram criadas as *tags* para o limite de operação e para o valor de temperatura instantâneo. Para isso, no menu “*Tags*”, foi criada uma pasta com o nome de “TagsPVP600”. A figura 48 ilustra o menu de tags e onde criar a pasta.

Figura 48 – Menu Tag



Fonte: Autor

Com a pasta criada, foram criadas duas tags do tipo analógico: uma chamada de setpoint, que representa o limite de operação, e uma chamada de “tempreal”, que representa o valor da temperatura instantânea. Conforme a figura 49, pode-se notar que ambas as tags possuem “TagsPVP600\” antes de seu nome. Isso se deve ao fato de ser a pasta onde estas tags estão alocadas.

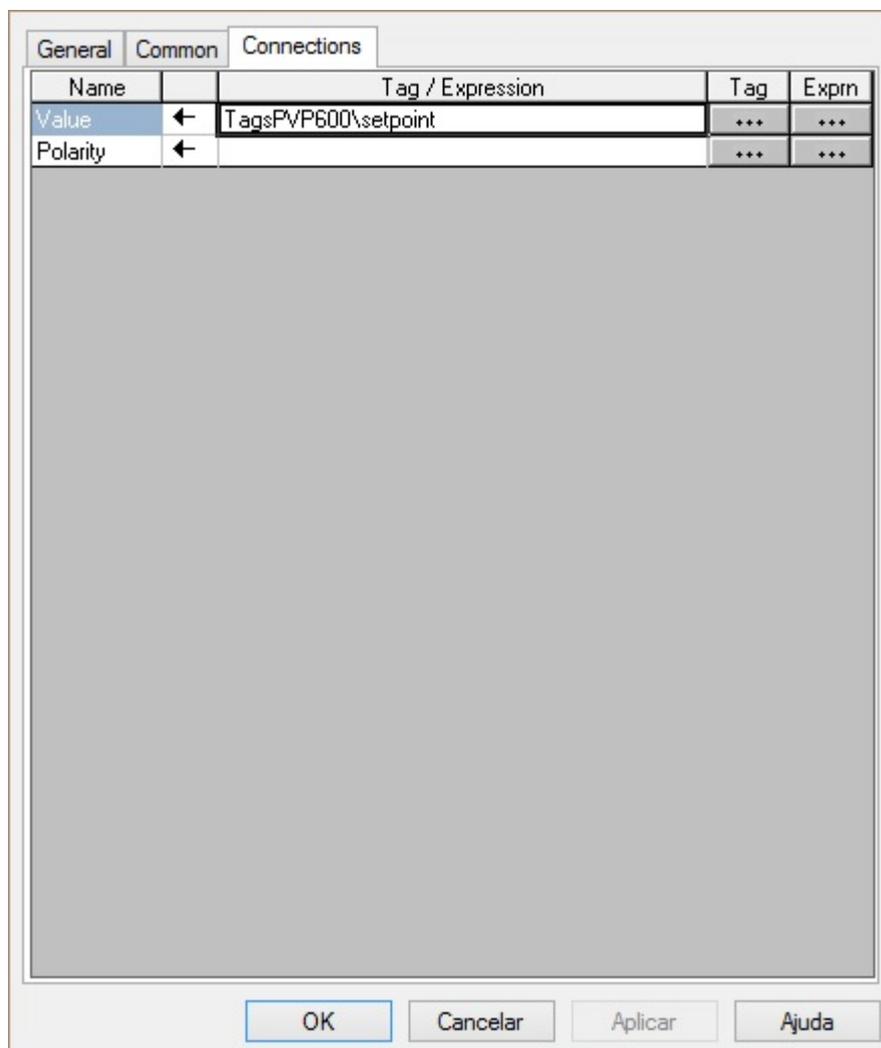
Figura 49 – Tags Criadas para PVP 600

	Tag Name	Type	Description
1	TagsPVP600\setpoint	Analog	Limite de Operação
2	TagsPVP600\tempreal	Analog	Valor de Temperatura Instantânea
3			

Fonte: Autor

Com as tags estabelecidas, foi estabelecida a conexão com os displays e gráficos. Para isso, selecionado o display/gráfico, em propriedades, na aba de conexão, basta selecionar a tag específica. A figura 50 ilustra um exemplo de tag selecionada na aba de conexão.

Figura 50 – Exemplo de Tag selecionada na aba de conexão

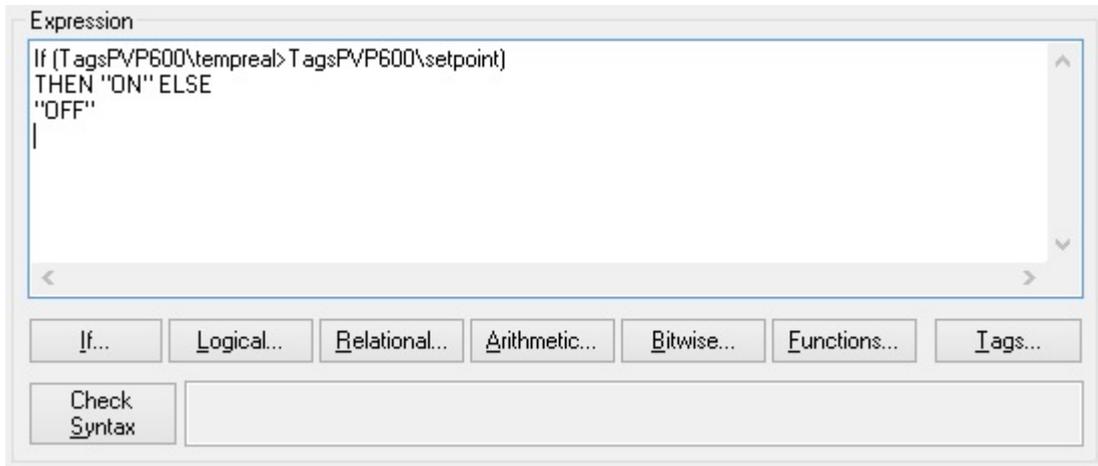


Fonte: Autor

Como pode-se notar, não foi criada a tag para o atuador. Isso se deve porque o atuador funciona por meio de uma comparação entre as duas tags criadas. Para isso, foi desenvolvida uma programação comparando as duas tags e, quando maior, a saída ilustra

“ON”. Caso contrário, sua saída ilustra “OFF”. Essa comparação é de maneira simultânea entre o CLP e a IHM. A figura 51 ilustra essa programação desenvolvida na IHM.

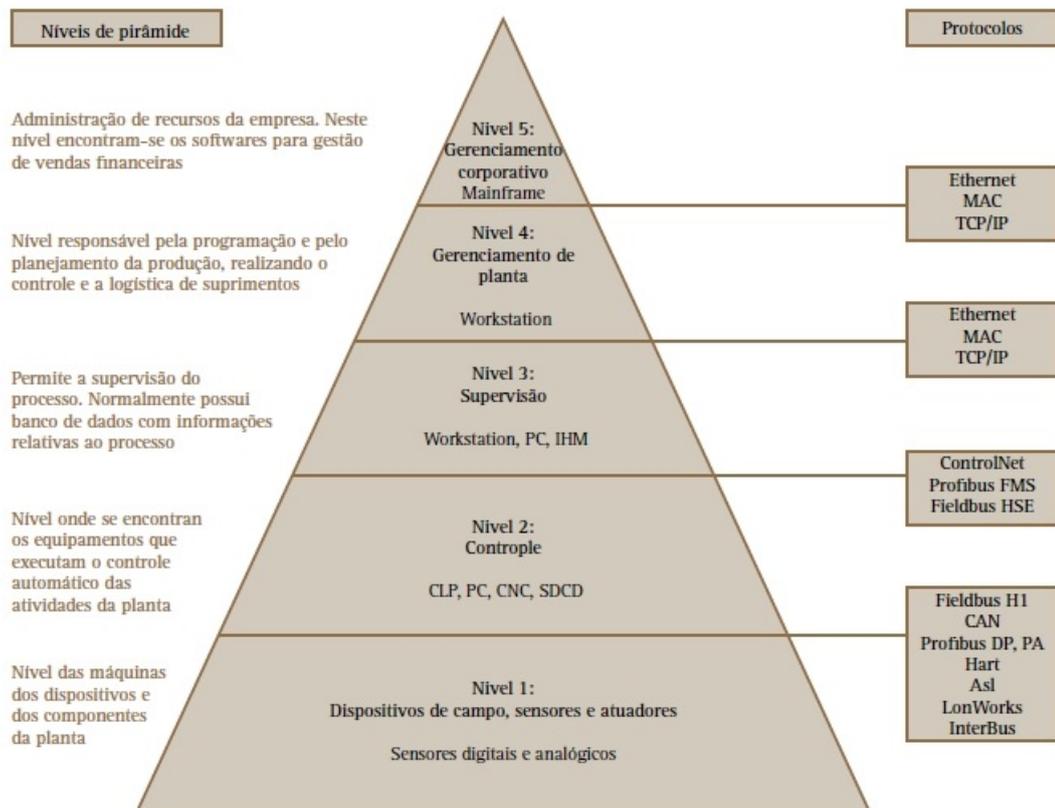
Figura 51 – Programação para tag do atuador



Fonte: Autor

4.3.2 Controlador lógico programável CompactLogix L24ER

Figura 52 – Pirâmide da Automação

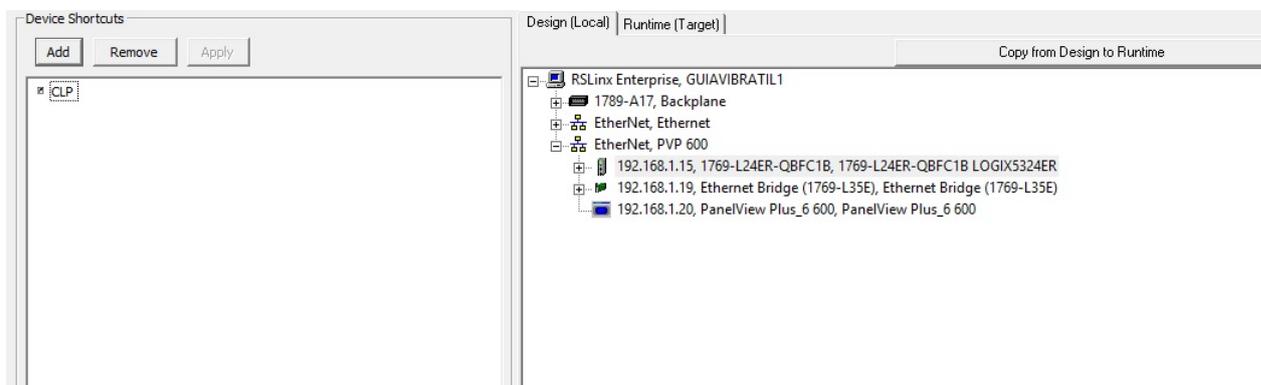


Fonte: Autor

A estação central de monitoramento, conforme a pirâmide da automação ilustrada na figura 52, atua no terceiro nível de hierarquia, realizando a supervisão do sistema, por meio da IHM e do CLP. Por estabelecer a comunicação com a IHM, o CompactLogix L24ER é responsável por armazenar o dado da variável de limite de operação, estabelecida pelo usuário por meio da IHM PanelView™ Plus 600, e enviá-lo ao CLP micro820. Com isso, o CompactLogix foi utilizado como uma ponte de comunicação entre a IHM PVP 600 e o CLP micro 820. É necessário a utilização do mesmo porque a IHM PanelView™ Plus 600 não estabelece comunicação com o CLP Micro820.

Para estabelecer a comunicação entre computador e o CLP CompactLogix, foi utilizado o *software* da Allen-Bradley®, o Studio 5000®. A comunicação desse CLP é realizada da mesma maneira que é estabelecida com a IHM PVP 600. A única diferença é que para estabelecer a comunicação, na parte do RSLinx Enterprise, deve-se criar um atalho do tipo “processor” e aplicá-lo ao CLP. Foi criado um atalho nomeado de “CLP”. Isso se deve porque, com esse atalho, consegue se ter acesso às tags do CLP. Também foi definido o IP estático 192.168.1.15 para o CLP. A figura 53 ilustra o CompactLogix L24ER no *driver* Ethernet da IHM, com seu IP estático, e também ilustra o seu atalho aplicado ao mesmo.

Figura 53 – OPC com CompactLogix L24ER e seu atalho



Fonte: Autor

4.4 Comunicação entre as estações

Para estabelecer a comunicação entre os CLP's, foram encontradas as seguintes dificuldades:

- O Micro820 trabalha no protocolo de comunicação TCP/IP enquanto o CompactLogix L24ER trabalha no protocolo de comunicação Ethernet/IP;
- O *software* CCW não reconhece o CompactLogix e o *software* Studio 5000® não reconhece o Micro820;

- A comunicação ponto-a-ponto entre os dois CLP's na porta analógica não é possível porque a saída analógica do CLP Micro820 é do tipo UINT (*Unsigned Integer*) e a saída analógica do CLP CompactLogix L24ER é do tipo inteiro.

Não há artigos no banco de pesquisa da Allen-Bradley® que contemplem a comunicação entre esses dois modelos. A proposta da engenharia da Allen-Bradley®, baseada no artigo X, que estabelece a comunicação entre um Micro850 e um CompactLogix L24ER, foi a comunicação por mensagem. A mesma se dá pela leitura e escrita realizada pelo CompactLogix L24 ER. Para isso, duas tags do tipo “*message*”, uma para leitura da variável e outra para escrever na variável, foram criadas. As figuras 54 e 55 demonstram as tags de leitura e escrita, respectivamente.

Figura 54 – Tag de mensagem de leitura

The image shows a software window titled "Message Configuration - MSG_Read". It has three tabs: "Configuration", "Communication", and "Tag". The "Tag" tab is selected. The main area contains the following fields and controls:

- Message Type:** A dropdown menu set to "CIP Data Table Read".
- Source Element:** A text input field containing "temperatura".
- Number Of Elements:** A spinner control set to "1".
- Destination Element:** A dropdown menu set to "temperatura".
- New Tag...:** A button to the right of the Destination Element dropdown.

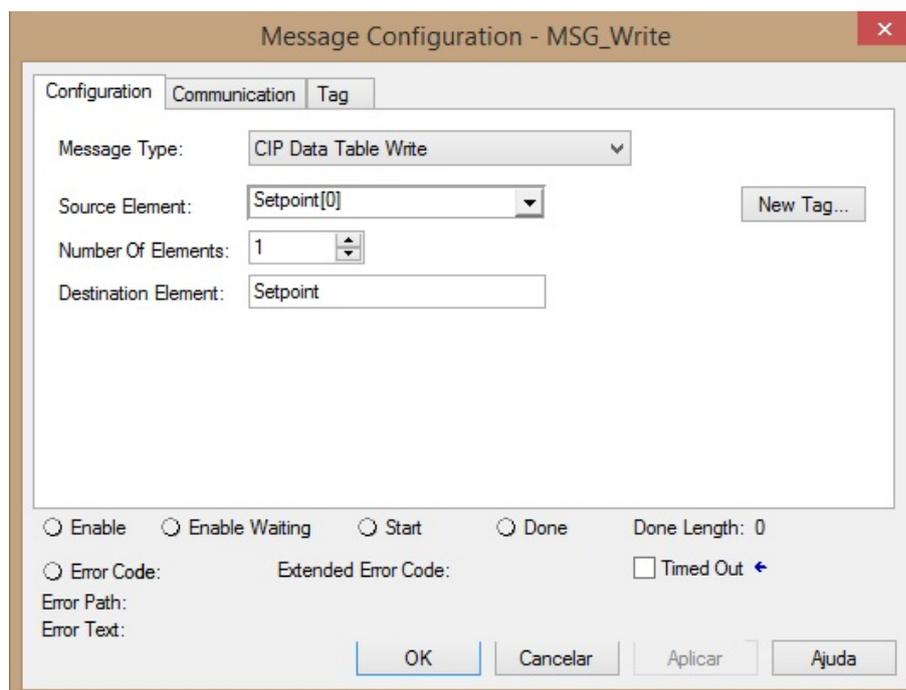
Below the main area, there are several options and error handling settings:

- Radio buttons for: Enable, Enable Waiting, Start, Done. Next to "Done" is "Done Length: 0".
- Radio buttons for: Error Code: and Extended Error Code:.
- Text input fields for "Error Path:" and "Error Text:".
- A checkbox for "Timed Out" which is checked.

At the bottom of the dialog are four buttons: "OK", "Cancelar", "Aplicar", and "Ajuda".

Fonte: Autor

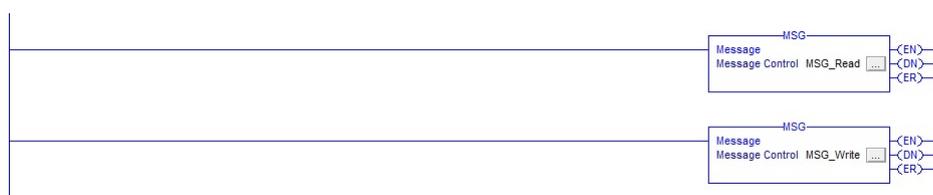
Figura 55 – Tag de mensagem de escrita



Fonte: Autor

Com as tags elaboradas, foi desenvolvida uma programação simples para testar a comunicação. O bloco de mensagem possui três pinos de saída: o pino “EN”, que indica quando a saída do bloco está habilitada, “DN” que indica quando a mensagem foi executada da maneira correta, e “ER” que indica quando houve erro na mensagem. Essa programação é demonstrada na figura 56 .

Figura 56 – Primeira programação para teste da comunicação entre CLP's

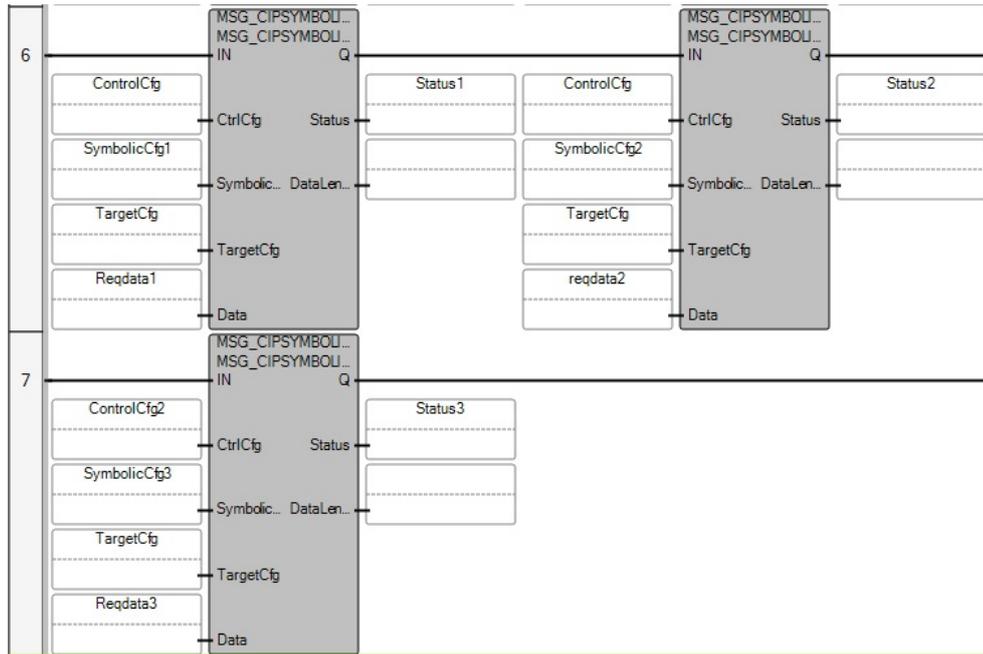


Fonte: Autor

Como é notado na figura 56, a saída “ER” está indicando que possui algum erro. Isso se deve ao fato de o CompactLogix ter conseguido enviar uma solicitação de leitura ou escrita, porém, devido ao Micro820 não reconhecer o protocolo de comunicação Ethernet/IP, a tag não foi lida ou escrita.

Foi então realizado um teste fazendo a leitura e a escrita pelo Micro820. Para isso, foi desenvolvida uma programação com os blocos *CIPSymbolic* no *software* CCW. Essa programação é demonstrada na figura 57.

Figura 57 – Segunda programação para teste da comunicação entre CLP's

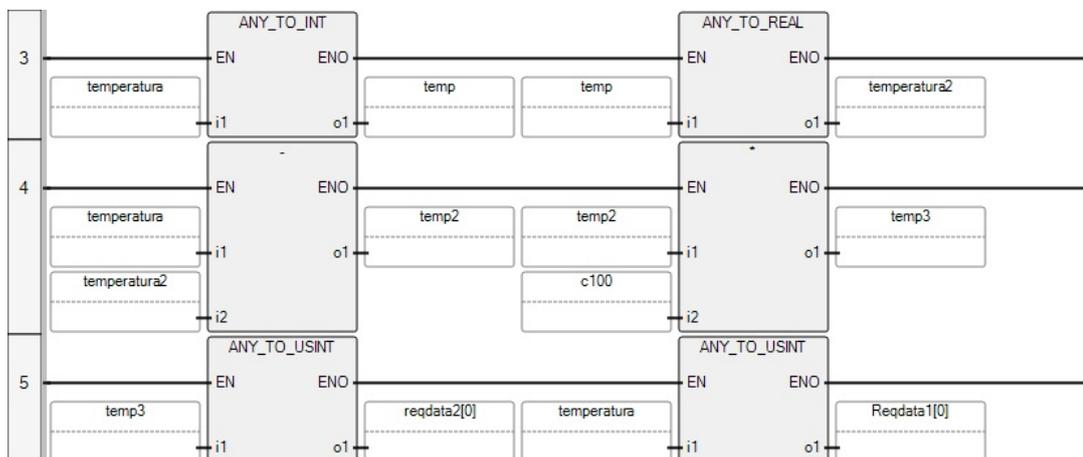


Fonte: Autor

Quando a programação da figura 57 foi baixada pelo Micro820 e colocada para executar, a mensagem de leitura conseguiu retornar o valor da tag que se encontrava no CompactLogix L24ER e a mensagem de escrita conseguiu escrever um valor numa tag do CompactLogix. Com a comunicação estabelecida, a temperatura real deverá ser escrita no CompactLogix L24ER e o limite de operação deve ser lido e armazenado no Micro820.

Porém a mensagem manda uma variável do tipo USINT (*Unsigned Short Integer*) e, com isso, se perde informação das casas decimais da temperatura atual. Como solução, foi retirada a casa decimal da temperatura e armazenada numa variável, transformada essa variável em inteiro, multiplicando por 100 para então enviar para o CompactLogix L24ER. A figura 58 ilustra o programa desenvolvido para essa solução.

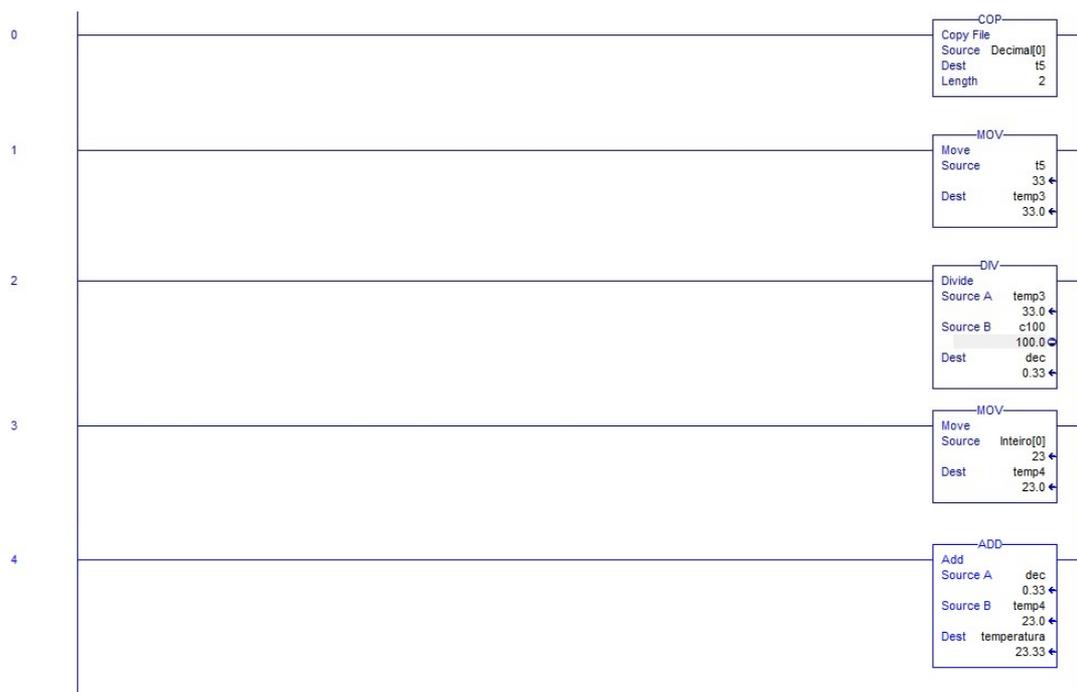
Figura 58 – Programação para transformar as casas decimais em inteiro



Fonte: Autor

As variáveis “reqdata2[0]” e “reqdata1[0]” representam as tags da casa decimal e inteira da temperatura, respectivamente. Com isso, foram enviadas para o CompactLogix L24ER e nele é feito o trabalho inverso: dividindo a decimal por 100 e então somar com a variável inteira, obtendo a temperatura real. A programação desenvolvida no CLP é ilustrada na figura 59.

Figura 59 – Programação no CompactLogix L24ER para adicionar o decimal do inteiro da Temperatura



Fonte: Autor

5 CONCLUSÃO

Foi desenvolvido um sistema supervisório em três níveis hierárquicos. No nível referente aos sensores e atuadores foram utilizados o sensor de temperatura LM35 e o motor de passo AK17 para simulação de um processo industrial. No próximo nível, responsável pelo controle, utilizou-se o CLP micro820 e a IHM PVC 600, sendo a mesma responsável apenas por uma indicação local. No nível de supervisão, foram utilizados o CLP CompactLogix L24ER e a IHM PVP 600, representando o sistema supervisório de todo o processo, podendo ser estendido para monitoramento e controle agregando um número maior de CLPs, a nível de processo local, em desenvolvimentos futuros.

Visando manter um nível de supervisão confiável e com capacidade de escalabilidade da rede, foi utilizado o CLP de médio porte CompactLogix L24ER que possui um desempenho superior aos CLPs micro820, utilizados no nível de controle deste trabalho.

Com relação ao sistema supervisório, foi estabelecida uma comunicação via protocolo TCP/IP com as interfaces humano-máquina, possibilitando a troca de dados manipulados e dispostos nas variáveis apresentadas na tela principal da estação do operador e salvos num dispositivo, via USB da IHM, permitindo ao usuário do sistema realizar consultas ao histórico da variável envolvida no sistema.

Com relação à rede industrial de comunicação, é possível afirmar que o protocolo TCP/IP foi efetivo tendo em vista que o mesmo conseguiu estabelecer comunicação com os diferentes tipos de dispositivos, utilizando o tipo de comunicação CIP da Allen-Bradley® para os dispositivos de baixo porte e o OPC para os dispositivos de médio porte, sem perda de informações do processo.

Como sugestões de trabalhos futuros tem-se a implementação de novas variáveis para serem controladas para estabelecer a comunicação da estação central de monitoramento com mais de uma bancada, fazendo uma topologia do tipo estrela do sistema supervisório. Outra sugestão de trabalho futuro é a utilização de um outro tipo de rede industrial de comunicação, como a profibus por exemplo, para ter uma simulação mais próxima da realidade de um ambiente industrial. Um trabalho futuro será o desenvolvimento de uma apostila para ensinar os alunos a fazer a comunicação entre os CLP's.

REFERÊNCIAS

- AUTOMATION, R. *CompactLogix Controllers Specifications*. 2014. Acessado em: 30/09/2014. Disponível em: <http://literature.rockwellautomation.com/idc/groups/literature/documents/td/1769-td005_en-p.pdf>. 16, 17
- AUTOMATION, R. *PanelView Component*. 2014. Acessado em: 26/09/2014. Disponível em: <http://literature.rockwellautomation.com/idc/groups/literature/documents/pp/2711c-pp001_pt-p.pdf>. 18
- AUTOMATION, R. *PanelView Plus 6 400 and 600 Terminals*. 2014. Acessado em: 03/10/2014. Disponível em: <<http://ab.rockwellautomation.com/pt/Graphic-Terminals/2711P-PanelView-Plus-600>>. 17, 18
- AUTOMATION, R. *Sistemas de controlador logico programavel Micro820*. 2014. Acessado em: 26/09/2014. Disponível em: <<http://ab.rockwellautomation.com/pt/Programmable-Controllers/Micro820>>. 15, 16
- BAILEY, D.; WRIGHT, E. *Practical SCADA for industry*. [S.l.]: Newnes, 2003. 10, 19, 20, 21, 22
- BOARETTO, N.; KOVALESKI, J. L.; SCANDELARI, L. Coleta de dados e monitoramento de chão de fábrica na manufatura discreta-integração com as ferramentas de gestão. In: *SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO*. [S.l.: s.n.], 2004. v. 11, p. 1-8. 21
- CLARKE, G. R.; REYNDERS, D.; WRIGHT, E. *Practical modern SCADA protocols: DNP3, 60870.5 and related systems*. [S.l.]: Newnes, 2004. 10
- CORDEIRO, R.; OLIVEIRA, M.; CHANQUINI, T. Utilização de conceitos de interface homem-máquina para adaptação da disciplina de requisitos do rup. *São Paulo*, 2009. 17
- CORRÊA, H. L.; GIANESI, I. G.; CAON, M. Planejamento, programação e controle da produção. *São Paulo: Atlas*, v. 1, 2001. 25
- GEVAERD, B. d. M.; SILVA, F. C.; BONACORSO, N. G. Desenvolvimento de um controlador lógico programável modular dedicado ao controle e ao acionamento de equipamentos automatizados. *Caderno de Publicações Acadêmicas*, v. 2, n. 1, p. 71, 2011. 15
- JURIZATO, L. A.; PEREIRA, P. S. R. Sistemas supervisórios. *Nova Odessa, Network Technologies*, v. 1, p. 2, 2003. 19, 20, 27, 28
- KALAPATAPU, R. Scada protocols and communication trends. *ISA EXPO*, Citeseer, 2004. 21
- KOPELVSKI, M. M. *Teoria de CLP*. 2010. Acessado em: 26/09/2014. Disponível em: <www.cefetsp.br/edu/maycon/arqs/ap_clp_rev00.pdf>. 14, 15
- KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a internet*. São Paulo: Person, 2006. 23, 24

- MARDEGAN, R.; MARTINS, V.; OLIVEIRA, J. d. Estudo da integração entre sistemas scada, mes e erp em empresas de manufatura discreta que utilizam processos de usinagem. *ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO*, v. 2, n. 200, p. 1–8, 2003. 25, 26
- MELLO, R. dos S. Gerenciamento de dados xml. *EIN-Escola de Informática Norte da SBC*, p. 15–34, 2003. 22
- PRUDENTE, L. N.; ROCHA, H. M. da; II, E. J. L. Implementação em sistema scada de controle supervisório baseado em modelos em redes de petri de alto nível. 2011. 15
- SILVA, A. P. G. da; SALVADOR, M. O que são sistemas supervisórios? *Este artigo descreve o que é e quais as funcionalidades de um sistema supervisório*, p. 1–5, 2005. 19, 20, 22
- SOUZA, R. B. et al. Sisal, um sistema supervisório para elevação artificial de petróleo. In: *Rio Oil & Gas Expo and Conference*. [S.l.: s.n.], 2006. p. 11–14. 26, 27
- SOUZA, R. B. de. *Uma arquitetura para sistemas supervisórios industriais e sua aplicação em processos de elevação artificial de petróleo*. Tese (Doutorado) — UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE, 2005. 22
- TANENBAUM, A. S. *Computer networks 4th edition?* ed: *Prentice Hall*, 2003. 23, 24