

**CENTRO FEDERAL DE EDUCAÇÃO
TECNOLÓGICA DE MINAS GERAIS**

WAGNER LUIZ MONTANDON

**SISTEMA DE CONTROLE EM REDE MODBUS DAS PLANTAS DE NÍVEL E VAZÃO
DO LABORATÓRIO DE CONTROLE E AUTOMAÇÃO DO CEFET-MG BASEADO NA
PLATAFORMA ARDUINO**

**MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO
INDUSTRIAL**

Araxá, 2013

WAGNER LUIZ MONTANDON

**SISTEMA DE CONTROLE EM REDE MODBUS DAS PLANTAS DE NÍVEL E VAZÃO
DO LABORATÓRIO DE CONTROLE E AUTOMAÇÃO DO CEFET-MG BASEADO NA
PLATAFORMA ARDUINO**

**Monografia apresentada ao Curso de Engenharia
de Automação Industrial, como parte do requisito
para a obtenção do grau de Engenheiro de
Automação Industrial.**

Orientador: Professor Dr. Henrique José Avelar

**Araxá
CEFET-MG
DEZEMBRO/2013**

AGRADECIMENTOS

Agradeço aos meus pais e irmão pelo apoio que recebi durante toda minha vida, afinal, eles são a base do que sou hoje. Agradeço também minha noiva Sara por seu apoio e compreensão, pois juntos abrimos mãos de muitas coisas em razão de meus estudos. Agradeço a todos os professores e colegas de turma que se dedicaram e se empenharam em compartilhar seus conhecimentos.

“O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo. Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas admiráveis.”

José de Alencar

RESUMO

As redes industriais são de suma importância por se tratarem de um meio de comunicação rápido e de grande confiabilidade no que diz respeito a sistemas automatizados nas indústrias. Existem vários meios de comunicação e protocolos utilizados em diferentes ambientes de trabalho, seja em plantas onde as salas de controle encontram-se muito longe dos equipamentos que elas controlam; ambientes que possuem equipamentos que geram campos elétricos e causam interferência na transmissão dos dados que trafegam nestas redes entre outros. O Modbus RTU é um protocolo muito utilizado pelas indústrias, sendo mais comum em redes RS-485, que possibilita a comunicação com até 32 dispositivos escravos e um mestre, além de possibilitar a comunicação entre eles a uma distância de até 1200 metros (KRON, 2001), de forma simultânea junto ao mestre, no qual é possível configurar um sistema SCADA (Sistema de Controle de Supervisório e Aquisição de Dados), constituído de sensores, atuadores e CLPs. Em razão da sua relativa simplicidade, é de grande valia implantar este sistema no laboratório de Controle e Automação do CEFET-MG, utilizando dispositivos de baixo custo adaptando-os às plantas de controle de nível e vazão do laboratório, podendo assim oferecer aos alunos um sistema flexível em que possam realizar desde a criação de telas de supervisório, configurações dos *drivers* de comunicação com os dispositivos e até implementar algum tipo de controle, como PID, de forma a aprimorem seus conhecimentos no que diz respeito às redes industriais e sistemas de controle.

Palavras-chave: redes industriais, Modbus, SCADA.

ABSTRACT

Industrial networks are extremely important because they are a means of communication fast and highly reliable with regard to industrial automated systems. There are several means of communication and protocols used in different working environments, where the plants control rooms are further away in terms of equipment, environments that have equipment that generate electric fields and cause interference in the transmission of data traveling on these networks among other factors. The Modbus RTU protocol is widely used by industries, which are the most common RS-485 networks allow communication with up to 32 slave devices and only one master, and can communicate with each other at a distance of up to 1200 meters (KRON, 2001), simultaneously with the master, in which you can configure a SCADA system (Supervisory Control and Data Acquisition), consisting of sensors, actuators, and PLCs. Because of its relative simplicity, is valuable, deploy this system in laboratory control systems CEFET - MG, using low-cost devices, thus being able to offer students a flexible system that can realize from creating screens supervisory, the driver settings for communicating with devices and to implement some kind of control, such as PID, so it enhances your knowledge regarding the industrial networks and control systems.

Keyword: Industrial networks, Modbus, SCADA.

SUMÁRIO

1 - INTRODUÇÃO	12
2 - REDES INDUSTRIAIS DE AUTOMAÇÃO	16
2.1 - Meio Físico de Comunicação em uma Rede Industrial	18
2.1.1 - Par Trançado sem Blindagem (UTP).....	18
2.1.2 - Par Trançado com Blindagem (STP).....	19
2.2 - Topologia Física.....	19
2.2.1 - Ponto a Ponto	19
2.2.2 - Barramento	19
2.2.3 - Anel	19
2.2.4 - Estrela	20
2.3 - A Norma RS-485.....	20
2.4 - Tecnologia de Comunicação.....	20
2.5 - Comunicação <i>Half-Duplex</i>	21
2.6 - Protocolos de Comunicação	22
2.6.1 - Modbus	22
2.6 - Sistemas de Controle de Supervisório e Aquisição de Dados	25
2.6.1 - Elipse Scada	26
2.7 - Plantas de Controle de Nível e Vazão	27
2.7.1 - Planta de Controle de Nível	28
2.7.1 - Planta de Controle de Vazão	32
3 - DESENVOLVIMENTO DA TELA DE SUPERVISÓRIO	37
3.1 - Criação das TAGS e Configurações de Endereços de Comunicação	40
3.2 - Criando Links entre as TAGs, Gráficos e Botões das Telas de Supervisório.....	44
4 - CONFIGURAÇÃO DO ARDUINO COM O SISTEMA	45
4.1 - Arduino.....	45
4.2 - Código-Fonte do Protocolo Modbus.....	47
5 - MONTAGEM E RESULTADOS	48

5.1 - Conversor USB/RS-485 e <i>Transceiver</i> TTL/RS-485	48
5.2 - Ligações do Conversor USB/RS-485 com o Barramento Principal.....	50
5.3 - Resultados	55
6 - CONCLUSÃO	56

REFERÊNCIA

ANEXOS

LISTA DE FIGURAS

Figura 1 - Evolução das Tecnologias de Automação	18
Figura 2 - Esquema de Ligação <i>Half-Duplex</i> de uma Rede RS-485	21
Figura 3 - Comprimento do Cabo pela Velocidade de Transmissão	23
Figura 4 - Estrutura das Mensagens	24
Figura 5 - Interface Gráfica do Elipse SCADA.....	26
Figura 6 - Planta de Controle de Nível	29
Figura 7 - Visão Sinóptica da Planta de Controle de Nível.....	29
Figura 8 - Unidade Eletrônica de Controle do CRL	31
Figura 9 - Planta de Controle de Vazão	33
Figura 10 - Visão Sinóptica da Planta de Controle de Vazão.....	34
Figura 11 - Unidade Eletrônica de Controle do CRF	36
Figura 12 - Tela Inserindo Gráfico de Barras	37
Figura 13 - Tela Inserindo Botão de Set-Point	38
Figura 14 - Tela Inserindo o Gráfico de Valores das Variáveis e Set-Point.....	39
Figura 15 - Tela de Criação de TAGS	40
Figura 16 - Tela de Criação da TAG e Configuração de Comunicação	41
Figura 17 - Tela de Criação das TAGs e Configurações	42
Figura 18 - Tela de Relação de TAG ao Gráfico de Tendência.....	44
Figura 19 - Arduino UNO.....	46
Figura 20 - Plataforma de Desenvolvimento do Código-Fonte.....	47
Figura 21 - Shield RS-485.....	49
Figura 22 - Conversor USB/RS-485	50
Figura 23 - Cabo USB / Mini USB	50
Figura 24 - Esquema de Ligação da Rede	51
Figura 25 - Entrada A0 do Arduino.....	52
Figura 26 - Bornes de Saída de Sinal da Variável Controlada	52
Figura 27 - Saída PWM no pino 9 do Arduino	53
Figura 28 - Bornes de Entrada do Sinal Regulador.....	54
Figura 29 - Tela de Supervisório do Controle de Vazão.....	54
Figura 30 - Tela de Supervisório do Controle de Vazão.....	55

LISTA DE TABELAS

Tabela 1 - Configurações de Endereços e Tempo de Varredura.....	43
Tabela 2 - Características do Arduino UNO.....	46

LISTA DE SIGLAS E NOMECLATURAS

ASCII - American Standard Code for Information Interchange (Código Padrão Norte-Americano para Intercâmbio de Informações)

AD / DA - Analógico Digital / Digital Analógico

CEFET-MG - Centro Federal de Educação Tecnológica de Minas Gerais

CLPs - Controladores Lógicos Programáveis

CRC - Cycling Redundancy Check (Verificação de Redundância Cíclica)

CRF - Controlador Regulador de Vazão

CRL - Controlador Regulador de Nível

DC - Direct Current (Corrente Contínua)

EEPROM - Erasable Programmable Read-Only Memory (Memória Programável Apagável Somente de Leitura)

IHM - Interface Homem Máquina

I/O - Input/Output

I/P - Corrente/Pressão

LED - Light Emitting Diode (Diodo Emissor de Luz)

MRRP - Módulo de Registro e Controle Pneumático

MSRR - Controlador Pneumático Opcional

PC - Personal Computer (Computador Pessoal)

PH - Potencial Hidrognônico

P/I - Pressão/Corrente

PID - Proporcional Integral Derivativo

PWM - Pulse Width Modulation (Modulação por Largura de Pulso)

PSI - (Libra por Polegada Quadrada)

RS-485 - Recommended Standard (Padrão Recomendado)

RTU - Remote Terminal Unit (Unidade Terminal Remota)

SCADA - Supervisory Control And Data Acquisition (Sistema de Supervisão e Aquisição de Dados)

SRAM - Static Random Access Memory (Memória Estática de Acesso Aleatório)

STP - Shielded Twisted-Pair (Par Trançado Blindado)

USB - Universal Serial Bus (Barramento Universal Serial)

UTP - Unshielded Twisted Pair (Par Trançado sem Blindagem)

1 - INTRODUÇÃO

Em razão das exigências do mercado em obter produtos com alta qualidade, de pronta entrega e em grandes quantidades, se faz necessária a utilização de sistemas automatizados que possuam alta confiabilidade dentro de processos industriais e manufatura, o que garante um produto final dentro dos padrões exigidos e em tempo hábil. Para melhor compreensão citamos dois exemplos: o controle de nível de um tanque que deve manter-se em uma determinada altura para que assim não se perca o produto através do transbordamento do tanque ou até mesmo para que seja utilizada a sua capacidade a fim de se ter um índice máximo de produção. No caso da vazão de um fluido, ela deve ser mantida dentro de uma determinada faixa para que o produto final não fuja de suas características padrão, seja sua densidade ou PH por exemplo. Em razão desses fatores, se faz necessário o uso de tecnologia que atenda às indústrias de forma a garantir a confiabilidade dos sistemas e, conseqüentemente, uma produção dentro do prazo, e a qualidade dentro dos padrões exigidos.

Dentre as várias áreas, no que diz respeito a estas tecnologias, uma que se destaca no ramo industrial é a que diz respeito à comunicação entre os equipamentos automatizados e seus controles, ou seja, os componentes que irão trocar informações para manter uma determinada planta funcionando de forma sincronizada e produtiva.

Existem vários meios de comunicação entre equipamentos no ambiente industrial, sendo eles sem fio, através de antenas, ou conectados fisicamente, com a utilização de cabos de par trançado, coaxial ou mesmo fibra ótica, os quais utilizam inúmeros protocolos de comunicação e, dentre eles, uns possuem vantagens sobre os outros dependendo de cada caso e ambiente onde será aplicado.

Um protocolo de comunicação muito utilizado nas indústrias é o Modbus. Trata-se de um protocolo que trabalha em redes físicas RS-232, RS-422 e RS-485, sendo este o mais popular pelo fato de ser possível a comunicação multiponto, ou seja, tem a capacidade de comunicação entre vários equipamentos em um único barramento. Além disso, esta possui uma capacidade de transmissão através de cabos com até 1200 metros de comprimento entre equipamentos e salas de supervisão (KRON, 2001).

Juntamente com as redes de comunicação e componentes de um sistema de automação, utiliza-se um software onde são desenvolvidas as telas dos sistemas supervisório. Através do monitor dos computadores é possível visualizar todos os valores e *status* dos equipamentos localizados no campo, tais como a temperatura de uma caldeira, a vazão de um fluído ou a densidade de um material em um determinado processo. Além de indicar os valores dessas variáveis, nela também pode-se definir, através de botões virtuais na tela, por exemplo, um valor de temperatura no qual desejamos que a água permaneça em um tanque que se encontra no campo. Os equipamentos que executam os comandos para tais ações são denominados controladores que, por fim, enviam comandos ao atuador, localizado no campo, que mantém o controle de alguma variável do processo, seja ela vazão, nível, temperatura entre outras. No mercado, existe uma infinidade de marcas, tipos e modelos destes equipamentos, que as indústrias adquirem de acordo com suas necessidades.

No laboratório do CEFET-MG existe uma planta de controle de nível e outra de vazão, as quais utilizam comunicação com protocolo RS-232, ou seja, um computador com sistema supervisório específico para cada planta, o que, em uma situação de ambiente industrial, torna-se extremamente inviável.

Neste trabalho, será desenvolvido um sistema de controle que constituirá das plantas de controle de nível e vazão, as quais farão a comunicação com um sistema supervisório do tipo SCADA através de uma rede com protocolo Modbus em um barramento RS-485. Será utilizado um controlador na planta de nível e outro na planta de vazão de forma que essas atendam às condições de estabilidade. Um *notebook* conectado à rede supervisionará o controle das plantas.

O sistema supervisório utilizado será desenvolvido no Eclipse Scada, que ficará encarregado da supervisão e aquisição de dados. O aplicativo terá uma tela que irá gerar um gráfico referente à resposta do sistema de cada uma das plantas, além de possuir *displays* que indicarão os valores das variáveis manipuladas e controladas, assim como botões virtuais de incremento e decremento dos valores de *Set-Point*, seja ele de vazão ou nível.

No meio acadêmico, mais precisamente nas aulas de laboratório de Sistema de Controle, esse tipo de sistema é importante para que os alunos conheçam bem um dos

meios de comunicação entre equipamento de controle, os protocolos e componentes que são utilizados nas indústrias onde existe automação. Embora essa questão do aprendizado e conhecimento dos alunos seja importante, muitas das vezes isso é barrado por questões de recursos, principalmente financeiros, impedindo a aquisição dos componentes utilizados nesses sistemas por esses se constituírem de *hardwares* e *softwares* de elevado custo financeiro.

Ao contrário, em uma indústria, por este ser um meio que deve seguir padrões de instalação, projetos, além de ser um ambiente que exige componentes robustos, faz-se necessária a utilização de equipamentos devidamente testados e aprovados, o que acaba tornando-os bastante caros. No caso de um ambiente acadêmico, é aceitável a utilização de equipamentos e componentes de funcionamento similar, porém inferiores aos de nível industrial, e a grande vantagem disso é o custo que, no caso, é bem menor, em relação aos componentes de nível industrial.

Como desenvolver um sistema de controle, em uma planta de nível e outra de vazão, utilizando um Sistema de Controle de Supervisório e Aquisição de Dados com comunicação RS-485, protocolo Modbus e um controlador de baixo custo?

No mercado e, principalmente, no comércio eletrônico, existe uma variedade de componentes que podem ser utilizados em um sistema de controle de pequeno porte e que realizam tarefas similares aos componentes de padrão industrial. Existem softwares de desenvolvimento de telas de supervisório, onde fabricantes disponibilizam gratuitamente versões de demonstrações. Além deste *software* temos o *hardware*, de baixo custo, que no caso é o Arduino. Trata-se de uma placa de circuito que possui um microcontrolador com entradas e saídas, tanto discretas quanto analógicas. Contudo, ele pode muito bem comportar-se como um controlador, uma vez que tenha um código armazenado em sua memória, baseado em um algoritmo desenvolvido de forma a executar as funções de acordo com as necessidades de algum processo na planta que se deseja controlar.

O uso do Arduino viabiliza a construção de uma pequena rede de automação utilizando as plantas didáticas, onde poderemos monitorar não apenas uma planta, mas várias plantas simultaneamente através de um único computador, com uma tela de supervisório para cada planta. Com essa implantação, os estudos de uma rede Modbus

serão satisfatórios, garantindo assim o aprendizado do aluno, pois ele poderá também programar não somente um código baseado em um algoritmo de controle PID como utilizado nesse trabalho, mas também outros algoritmos de controle, além de ter a flexibilidade de implementar telas de supervisão, e configurar *drivers* de comunicação do sistema com o controlador.

Desse modo, o objetivo geral deste trabalho é montar uma pequena rede industrial com comunicação RS-485 sobre protocolo Modbus RTU composta pelas plantas de controle de nível e vazão do laboratório de Controle e Automação do CEFET-MG.

Para alcançar este resultado, desenvolveremos uma tela de supervisão em um *notebook*, o qual terá como finalidade indicar os valores lidos pelos instrumentos de campo e determinar os valores do *Set-Point* de funcionamento da variável controlada. Também montaremos um circuito eletrônico composto pelo Arduino que será o controlador do sistema e fará a comunicação com o computador.

Como procedimentos metodológicos, em princípio foram realizadas pesquisas teóricas através de artigos, livros e pesquisas na internet, focando o meio de comunicação RS-485 com protocolo Modbus RTU, buscando suas principais vantagens e configurações necessárias para sua utilização em uma rede industrial. Posteriormente foram levantados os componentes necessários para montagem de uma rede industrial baseado nesse protocolo, sendo eles, *hardwares* e *softwares*. Adquiridos esses componentes, partimos para a montagem de toda parte física do sistema, assim como a programação e desenvolvimento do supervisão e, por fim, efetuamos os testes necessários buscando um funcionamento satisfatório de todo sistema.

Nosso trabalho se apresenta em seis capítulos. No primeiro é descrita uma introdução e objetivos a serem atingidos. No segundo capítulo apresentamos os pressupostos teóricos necessários ao desenvolvimento da pesquisa, a saber: apresentação de uma introdução sobre as redes industriais, sua topologia em sistemas de automação e meios físicos na qual trafegam os dados. Além disso, abordamos sobre o protocolo Modbus, sendo ele muito utilizado na comunicação entre os equipamentos localizados no campo e salas de comando das fábricas. Descrevemos também a respeito de sistemas SCADA, que se trata de um sistema de controle de supervisão e aquisição de dados de uma fábrica, responsável por monitorar os valores das variáveis

de campo, e executar os comandos para execução de alguma ação nos equipamentos também localizados no campo. Descrevemos sobre as plantas de controle de nível e vazão utilizada no laboratório de sistemas de controle do CEFET-MG, as quais foram utilizadas como parte do trabalho proposto nesta monografia. No terceiro capítulo focamos no desenvolvimento de uma tela de supervisorio utilizando o Elipse Scada e efetuamos as configurações para que possa haver a comunicação do *notebook* com os controladores do sistema. No quarto capítulo, foi apresentada uma breve descrição sobre o Arduino. Foram mostradas as ligações elétricas do sistema e configurações a serem realizadas no Arduino, sendo este o responsável por controlar o sistema juntamente com o sistema supervisorio e os dispositivos de atuação e sensoriamento. No quinto capítulo foram descritos os resultados e melhorias realizadas durante a montagem do sistema e ajustes realizados para o devido funcionamento. E, por fim, no sexto capítulo, foi apresentada a conclusão referente ao trabalho e montagem realizada do sistema proposto.

2 - REDES INDUSTRIAIS DE AUTOMAÇÃO

Nos dias de hoje é inimaginável, em ambientes industriais, termos sistemas de comunicação e controle como os que tínhamos há 70 anos. O controle antigamente baseava-se apenas em sistemas pneumáticos e mecânicos, não havia componentes eletrônicos. Eram utilizadas linhas de transmissões pneumáticas, constituídas de tubos de cobre ou vinil de ¼" (diâmetro externo). Em casos especiais (atmosferas oxidantes), usavam-se tubos de aço inox. A distância prática para transmissão do campo para o painel era de aproximadamente 150 m, (SMAR, 98). Conforme os instrumentos de medição sofressem alguma variação no processo, alterava-se a pressão do ar e assim, equipamentos mecânicos localizados em uma sala, indicavam valores referentes aos lidos pelos instrumentos de campo de acordo com essa variação de pressão. Isso permitia que os controladores ficassem reunidos em um mesmo local, ou seja, na sala de controle do processo. Embora esse sistema fosse revolucionário, tinha um grande empecilho, a vias de transmissão eram constituídas de inúmeras tubulações que em questão de espaço físico tornava-se bastante complicado em uma implantação (SMAR,

1998). Com o passar do tempo, por volta de 1947, foi desenvolvido o transistor, que ajudou a impulsionar o desenvolvimento da automação. Tratava-se de um componente eletrônico capaz de controlar a passagem da corrente elétrica em determinados sistemas, e desta forma foram desenvolvidos inúmeros equipamentos eletrônicos que vieram ajudar na evolução das máquinas. Em 1960, empresas automobilísticas, produziam em massa, com rapidez e qualidade, mas não ofereciam muitas opções para os clientes, já que a linha de produção não era flexível (Goeking, 2010). A partir da década de 80, a eletrônica digital começou a tomar conta das questões que se tratavam de alta tecnologia. Em meados da década de 1980, vários componentes foram integrados em um mesmo *chip*, o que resultou em circuitos integrados e assim surgiram os microcontroladores, componentes constituídos de milhares de transistores com alto poder de processamento de informações, (MIKROELEKTRONIKA, 2003), eles alavancaram na difusão dos sistemas embarcados, (NETO, 2010). Na década de 90, o mundo começou a presenciar enormes avanços na área tecnológica, em que os circuitos eletrônicos passaram a proporcionar maior eficiência, maiores velocidades de processamento, mais funcionalidades, consumo menor, espaços físicos menores e ainda com redução de custos. Ao mesmo tempo em que impulsionou o desenvolvimento de computadores, interfaces e periféricos mais poderosos, com alta capacidade de processamento e memória, deu também vazão à alta escala de produção com custos reduzidos, o que foi uma vantagem de forma geral, aumentaram também a oferta de microcontroladores para toda a indústria (Cassiolato, 2012).

Atualmente as redes de automação industrial estão cada vez mais eficientes, uma vez que são constituídas por componentes que as caracterizam como uma comunicação rápida, precisa e confiável.

Além da comunicação analógica, de 4 a 20 mA, ainda utilizada por muitas indústrias, temos também as digitais, também chamadas redes de campo, as quais possuem um grande número de protocolos, e dentre os mais comuns são: Modbus, Profibus, Fieldbus, Controlnet, Devicenet e AS-I.

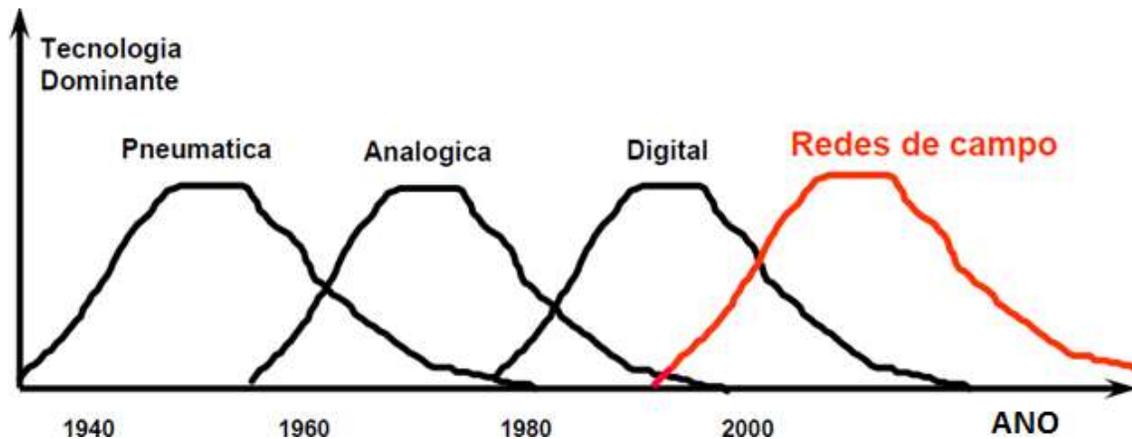


Figura 1 - Evolução das Tecnologias de Automação

Fonte: <http://www.dca.ufrn.br>

É importante diferenciarmos rede de controle e rede de campo. A função da rede de controle é interligar os sistemas de controle de supervisão e aquisição de dados aos CLPs e remotas de aquisição de dados. A função da rede de campo é garantir a conectividade entre os diversos dispositivos atuantes diretamente no “chão de fábrica”, sejam eles dispositivos de aquisição de dados, atuadores ou CLPs (CASSIOLATO, 2012).

2.1 - Meio Físico de Comunicação em uma Rede Industrial

O meio físico de transmissão está relacionado ao cabeamento utilizado para a interconexão dos dispositivos. Existem muitos tipos de meios físicos de transmissão, alguns exemplos são: par trançado, cabo coaxial e fibra ótica (MORAES; CASTRUCCI, 2007).

2.1.1 - Par Trançado sem Blindagem (UTP)

Existem vários tipos de cabos UTP, desde aplicações para telefonia até cabeamento para alta velocidade, como por exemplo, *Ethernet*. São cabos constituídos de quatro pares de fios. Cada par de fios é enrolado entre si, o que ajuda a eliminar

interferência do par adjacente e de outros dispositivos elétricos (MORAES; CASTRUCCI, 2007).

2.1.2- Par Trançado com Blindagem (STP)

O cabo STP, se comparado ao UTP, são aplicados em ambiente mais susceptíveis a interferências elétricas, como por exemplo, em subestações de usinas e lugares com interferência de frequência de rádio, ao passo que o UTP não seria o ideal em ambiente como estes descritos.

2.2- Topologia Física

A topologia física da rede está relacionada com a disposição construtiva na qual os dispositivos estão conectados às redes de barramentos distribuídas ou a sistemas distribuídos (MORAES; CASTRUCCI, 2007).

2.2.1 - Ponto a Ponto

Essa topologia tem como característica a comunicação entre dois processadores, conectados diretamente, ou seja, não podem usar outros nós nos roteadores. Essa topologia é pouco utilizada, pois não possibilita a adição de novos equipamentos (MORAES; CASTRUCCI, 2007).

2.2.2 - Barramento

Nesse tipo de topologia, o meio físico de comunicação é compartilhado entre todos os processadores, sendo que o controle pode ser centralizado ou distribuído. É largamente utilizado, pois possui alto poder de expansão, e um nó único com falha não prejudica os demais (MORAES; CASTRUCCI, 2007).

2.2.3- Anel

Trata-se de uma arquitetura ponto a ponto em que cada processador é conectado a outro, fechando-se o último segmento ao primeiro. O sinal circula no anel até chegar ao ponto de destino. É mais confiável em relação à topologia ponto a ponto, porém,

possui grande limitação quanto à sua expansão, pois devido ao aumento da quantidade de equipamentos haverá maior retardo na transmissão (MORAES; CASTRUCCI, 2007).

2.2.4 - Estrela

Essa topologia utiliza um só nó central para a comunicação entre as máquinas. Nós em falhas não afetam os outros, com exceção do nó central, que provoca falha em toda a rede. Por esse motivo, são utilizados processadores em redundância para garantir a confiabilidade do sistema (MORAES; CASTRUCCI, 2007).

2.3 - A Norma RS-485

A norma RS-485 define esquema de transmissão de dados balanceados que oferecem soluções robustas para transmitir dados em longas distâncias em ambientes ruidosos. Essa não define qual o protocolo a ser utilizado para a comunicação dos dados e é adotada como especificação da camada física de diversos protocolos, como por exemplo Modbus, Profibus e muitos outros (KRON, 2001). Este meio de transmissão possibilita taxas de transmissão de até 12 Mbits/s (MORAES; CASTRUCCI, 2007).

2.4 - Tecnologia de Comunicação

Uma das formas de gerenciamento entre os pontos de comunicação da rede no tocante a comunicação de dados é o Mestre-Escravo podendo ser Monomestre ou Multimestre (MORAES; CASTRUCCI, 2007). O Mestre-Escravo, o escravo é um periférico, geralmente dispositivos inteligentes de entradas e saídas, *drivers* e válvulas que recebem uma informação do processo e/ou utilizam informações de saída do mestre para atuar na planta.

Monomestre: é onde existe somente um mestre no barramento. Os escravos são descentralizadamente acoplados no barramento através do meio de transmissão de dados.

Multimestre: os dados de entrada podem ser lidos por todos os mestres, porém somente um mestre pode controlar um dado escravo (MORAES; CASTRUCCI, 2007).

2.5- Comunicação *Half-Duplex*

Esta é a forma de comunicação mais popular de utilização da RS-485 (KRON, 2001). Caracteriza-se pela utilização de um único par de fios para transmissão e recepção de dados. Múltiplos dispositivos são ligados na forma de um barramento, conforme ilustra a figura 2. Para a ligação dos barramentos de comunicação entre os dispositivos da rede deve-se utilizar cabo tipo par trançado, tendo o cuidado de interconectar os terminais 'comum' de todos os dispositivos da rede. A adoção de um condutor adicional para interligação do 'comum' de todos os dispositivos da rede é altamente recomendada. A não observância desta recomendação pode resultar na queima dos circuitos de comunicação de um ou mais equipamentos da rede (KRON, 2001).

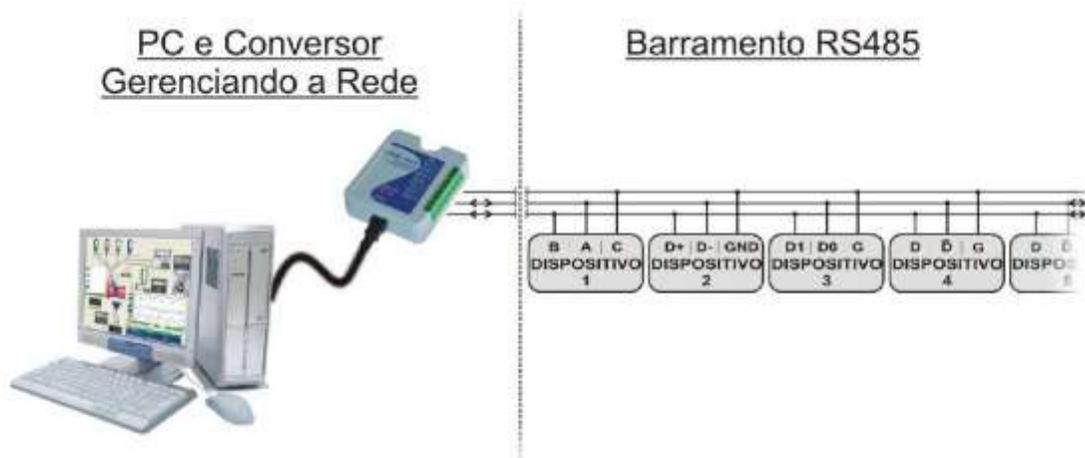


Figura 2 - Esquema de Ligação *Half-Duplex* de uma Rede RS-485

Fonte: <http://www.kronweb.com.br>

2.6 - Protocolos de Comunicação

Protocolos de comunicação caracterizam os elementos de maior importância nas redes de automação industrial, tanto que elas passam a ser denominadas pelos protocolos utilizados (MORAES; CASTRUCCI, 2007). Basicamente, um protocolo é um conjunto de regras sobre o modo como se dará a comunicação entre as partes envolvidas. A violação do protocolo dificultará a comunicação e em alguns casos poderá impossibilitá-la (TANEMBAUM, 1997). Dentre os protocolos utilizados em redes industriais os mais comuns são: HART, Profibus, Fieldbus, Controlnet, Devicenet e o Modbus, no qual focaremos no nosso trabalho.

2.6.1 - Modbus

Desenvolvido e publicado pela *Modicon Industrial Automation Systems*, em 1980 o protocolo Modbus se tornou padrão na indústria (SCHNEIDER, 2007). É um dos protocolos mais antigos utilizados em rede de controladores lógicos programáveis para aquisição de sinais de instrumentos e comandos de atuadores utilizando uma porta serial. Atualmente parte do grupo Schneider Electric, a MODICON definiu normas e especificações que determina o Modbus como domínio público e, em razão disso, milhares de equipamentos existentes são uma opção barata com este tipo de protocolo na área de automação industrial. O protocolo Modbus trabalha sobre as normas RS-232, RS-422 ou RS-485, e utiliza a tecnologia de comunicação mestre-escravo, sendo que somente um mestre e no máximo 32 escravos podem ser conectados à rede. A comunicação se inicia sempre pelo mestre e os escravos não se comunicam entre si. As mensagens transmitidas pelo mestre podem ser de duas formas:

- a) Mensagem tipo *unicast*: o mestre envia uma requisição para um escravo definido e este retorna uma mensagem-resposta ao mestre. Portanto, nesse modo são enviadas duas mensagens: uma requisição e uma resposta;
- b) Mensagem tipo *broadcast*: o mestre envia a requisição para todos os escravos, e não é enviada nenhuma respostas para o mestre (MORAIS; CASTRUCCI, 2007).

O meio físico mais utilizado com este protocolo é o cabo de par trançado. A comunicação acontece através de um único par de fios, onde cada dispositivo transmite e recebe dados. A norma RS-485 especifica um comprimento máximo de 1200 metros

para os cabos de comunicação (KRON, 2001). Como regra geral, quanto mais longo o cabo menor será a velocidade de comunicação.

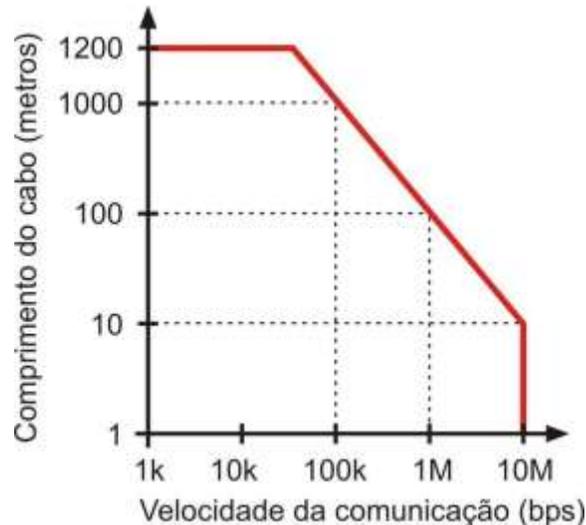


Figura 3 - Comprimento do Cabo pela Velocidade de Transmissão

Fonte: <http://www.kronweb.com.br>

O protocolo Modbus define também quatro tipos de blocos de dados referentes aos registradores onde são armazenadas informações. Cada um desses registradores possui um endereço e função específica:

- a) *Read Coil* - seu endereço vai de 000001 até 065535. Essa função é utilizada para escrever um bit na saída de um controlador, ou seja, ligar ou desligar uma bobina.
- b) *Discrete Input* - seu endereço vai de 100001 até 165535. Essa função é utilizada para ler um bit de entrada em um controlador, ou seja, uma botoeira quando ela é ou não acionada.
- c) *Input Register* - seu endereço vai de 30000 até 365535. Essa função é utilizada para escrever valores de 16 bits de entrada em um controlador, ou seja, a leitura de uma variável analógica, como o sinal de corrente elétrica.
- d) *Holding Register* - seu endereço vai de 400001 até 465535. Essa função é utilizada como propósito geral, podendo ser de escrita ou leitura em um controlador, seja de 1 ou 16 bits.

O protocolo Modbus define dois modos de transmissão de dados:

a) ASCII: caracteres codificados em 7 bits + 1 bit de paridade.

b) RTU: no modo RTU, cada palavra transmitida possui 1 *start* bit, oito bits de dados, 2 *stop* bits, sem paridade (WEG, 2007).

Na estrutura das mensagens no modo RTU toda comunicação se inicia com o mestre fazendo uma solicitação a um escravo, e este responde ao mestre o que foi solicitado. Em ambos os telegramas (pergunta e resposta), a estrutura utilizada é a mesma: Endereço, Código da Função, Dados e *Checksum*. Apenas o conteúdo dos dados possui tamanho variável (WEG, 2007).

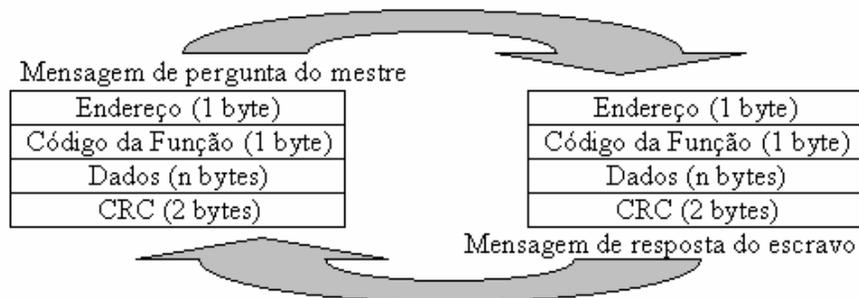


Figura 4 - Estrutura das Mensagens

Fonte: Manual da WEG, 2007

Conforme figura 4, a estrutura apresenta as seguintes funções:

a) Endereço: o mestre envia um byte com o endereço do escravo que ele deseja comunicação. Ao enviar a resposta, o escravo inicia o pacote com o seu próprio endereço, assim o mestre sabe qual escravo lhe enviou a resposta. É possível também o mestre enviar uma mensagem a todos os escravos, basta destiná-la ao endereço 0, mas assim nenhum escravo responde, eles apenas recebem a mensagem.

b) Código da Função: este campo contém um byte, no qual o mestre especifica o tipo de serviço ou função solicitada ao escravo, ou seja, escrita ou leitura. De acordo com o

protocolo, cada função é utilizada para acessar um tipo específico de dado. Os dados estão disponibilizados com tipo *holding* (*words*) ou tipo *coil/input discrete* (bits) (WEG, 2007).

c) Campo de Dados: este campo tem seu tamanho variável, o qual depende da função que se utiliza e dos valores transmitidos. A descrição da função está junto com esse campo (WEG, 2007).

d) CRC: é a última parte do pacote, responsável por checar se existem erros na transmissão (WEG, 2007).

2.6 - Sistemas de Controle de Supervisório e Aquisição de Dados

A sigla SCADA vem do inglês *Supervisory Control And Data Acquisition*, o que significa Controle Supervisório e Aquisição de Dados. Sistemas SCADA servem como interface entre o operador e processos dos mais variados tipos, como máquinas industriais, controladores automáticos e sensores (SCADABR, 2010). Através destes sistemas é possível desenvolver desde aplicativos simples, com indicação de sensores e automação, painéis de controle de indústrias de mineração, automotiva, alimentícia, até sistemas de controle de distribuição de energia. Um sistema SCADA típico oferece *drivers* de comunicação com equipamentos, sistemas de registro de dados contínuos e uma interface gráfica ao usuário do sistema, também chamadas de IHM. Nas IHMs são mostrados elementos gráficos, botões, ícones e displays, os quais indicam as condições e variáveis do ambiente industrial, ou seja, o comportamento dos equipamentos dos processos da fábrica. Dentre as funções mais utilizadas estão:

- a) Geração de gráficos e relatórios com histórico do processo;
- b) Detecção de alarmes e registros de eventos
- c) Controle de processo através de envio de comandos de *Set-Point* e acionamentos de equipamentos.

No mercado existem vários fabricantes, dentre os quais se destacam: IFIX, INDUSOFT, TATSOFT, INTOUCH e o ELIPSE SCADA, sendo este o utilizado em nosso trabalho

em razão de possuir uma versão demonstração que atenda às necessidades aqui propostas.

2.6.1 - Elipse Scada

O Elipse Scada é desenvolvido pela Elipse Software e conta com mais de 450 *drivers* de comunicação compatíveis com os equipamentos de inúmeros fabricantes. Os operadores podem monitorar e controlar com precisão os processos do chão de fábrica, bem como máquinas e recursos. Dados em tempo real são apresentados de forma gráfica, permitindo tratar as informações de diversas maneiras como o armazenamento histórico, a geração de relatórios e a conexão remota, entre outras possibilidades. Análises precisas e respostas rápidas resultam em menos perdas e altos níveis de qualidade (ELIPSE SOFTWARE, 2013).

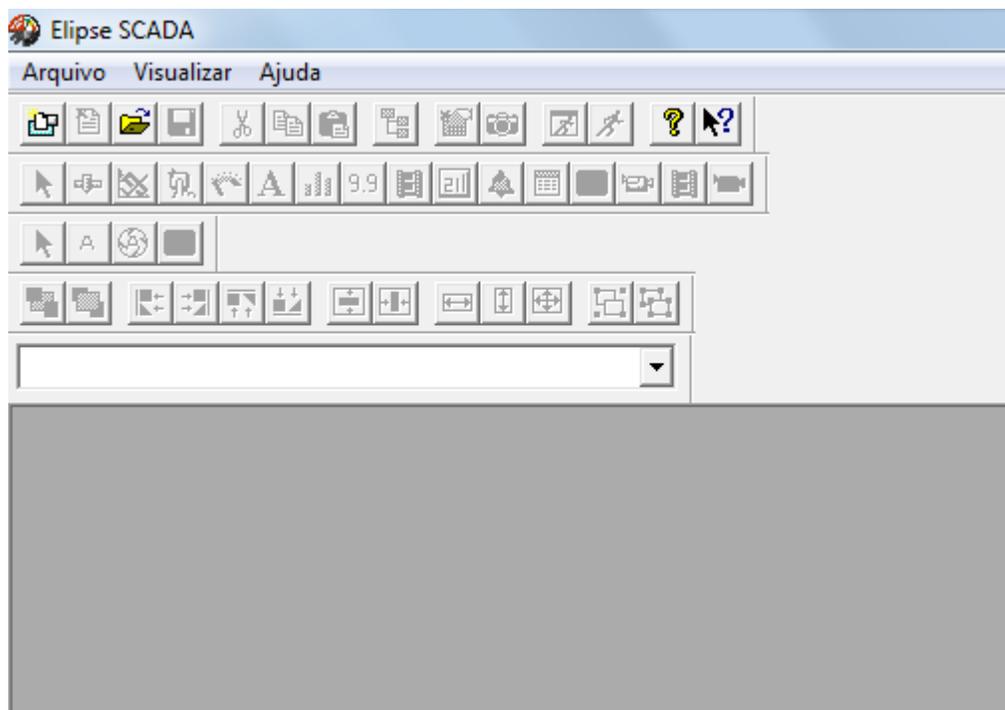


Figura 5 - Interface Gráfica do Elipse SCADA

Fonte: Autor

No *site* do fabricante (www.elipse.com.br) é disponibilizado para *download* uma versão demonstração, a qual utilizamos. Essa versão possui todos os recursos existentes no módulo configurador, com as seguintes diferenças:

- a) Não permite criar aplicações com mais de 20 TAGS (Tags, Ram, CLP, Matriz, Expressão, Demo, DDE, Bloco CLP, Crono).
- b) Permite até cinco (5) conexões simultâneas do Elipse Web.
- c) Permite a execução de uma aplicação e comunicação com equipamentos de aquisição de dados por até duas horas.

Mesmo com essas limitações, o nosso trabalho pode fluir normalmente, uma vez que não iremos utilizar recursos além dos oferecidos pela versão demonstração, já que o número de TAGs a serem utilizadas não passou de vinte.

2.7- Plantas de Controle de Nível e Vazão

Em uma indústria existem diferentes processos os quais possuem diferentes tipos de variáveis, que devem ser controladas a fim de se obter um determinado produto pelo qual todo o processo foi projetado. As variáveis mais encontradas são nível, vazão, pressão, densidade, peso, velocidade e temperatura. Um exemplo seria a preparação da polpa de um concentrado de minério. Levamos em consideração que o processo é contínuo, então a quantidade de concentrado de minério seco a ser adicionado ao tanque agitador de polpa deve ser controlada, ou seja, a velocidade da correia que transporta o material ao tanque deve ser contínua ao mesmo tempo em que a água adicionada à mistura deve permanecer em vazão constante, também controlada. Desta forma obtêm-se a receita dentro dos padrões pré-estabelecidos. O CEFET-MG possui em seu laboratório de controle e automação, plantas de controle de nível e vazão que são utilizadas nas aulas de sistema de controle e instrumentação, com o intuito de simular condições às quais os alunos encontrarão em ambientes industriais.

2.7.1 - Planta de Controle de Nível

A planta de controle de nível, ou CRL, foi desenvolvida para o estudo das técnicas de controle de nível em malha aberta ou fechada. Ela ilustra uma situação real e frequentemente encontrada nas indústrias. Certa quantidade de líquido (por exemplo, água) é contida em um tanque. Em razão de alguma necessidade se faz encher o tanque com líquido e mantê-lo em quantidade suficiente. A ideia é realizar um controle capaz de manter constante o nível do líquido no tanque, de acordo com o *Set-Point* determinado. Desta forma, o nível do líquido no tanque mantém-se constante, independente da vazão de saída do tanque (o volume de água “gasta”). A unidade de controle desta planta trabalha o fluxo de entrada de água no tanque com a abertura e fechamento de uma válvula de controle pneumática do tipo diafragma. Um motor elétrico aciona uma bomba que succiona a água do reservatório e a envia para o tanque através de uma tubulação. De acordo com o *Set-Point* determinado, a válvula que se encontra nesta tubulação realiza o controle do nível, permitindo a passagem de mais ou menos água para o tanque, mantendo o nível dentro do valor desejado. A planta de controle de nível e seus sinóticos são apresentados na figura 6.

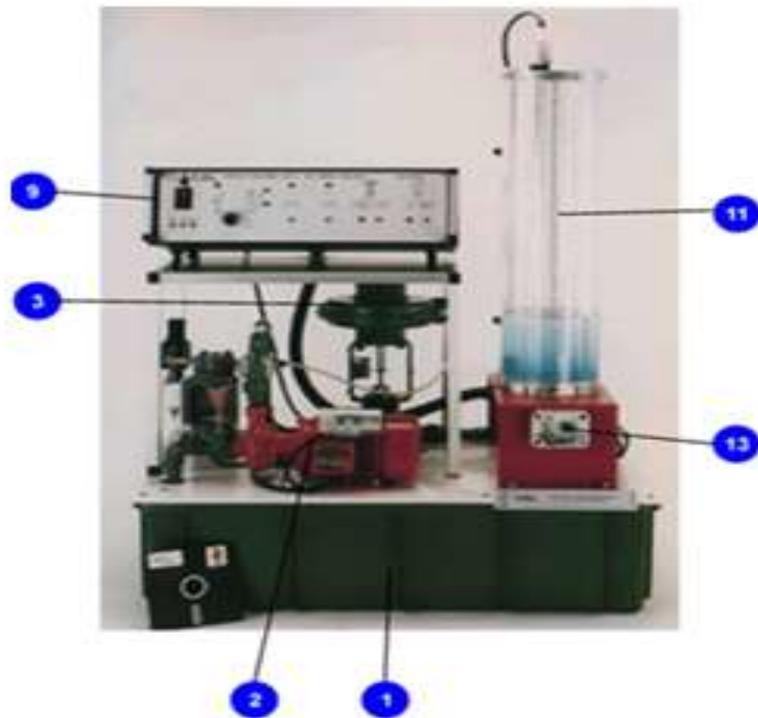


Figura 6 - Planta de Controle de Nível

Fonte: Manual do CRL, página 5

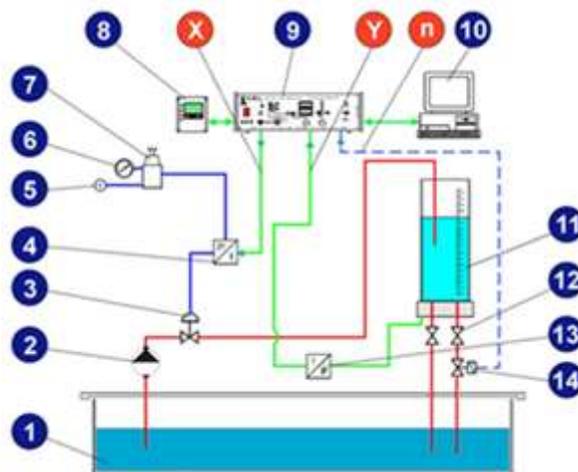


Figura 7 - Visão Sinóptica da Planta de Controle de Nível

Fonte: Manual do CRL, página 5

Conforme as figuras 6 e 7, o equipamento possui os seguintes componentes:

- 1-Tanque de coleta de líquido;
- 2-Bomba centrífuga;
- 3-Válvula pneumática proporcional;
- 4-Transdutor de I/P;
- 5-Entrada de ar comprimido;
- 6-Medidor de pressão de ar comprimido;
- 7-Dispositivo de ajuste manual da pressão de ar comprimido;
- 8-Controlador eletrônico opcional MiniReg ;
- 9-Equipamento elétrico
- 10-Computador pessoal
- 11-Tanque graduado transparente;
- 12-Válvulas manuais de descarga;
- 13-Transdutor de P/I;
- 14-Válvula de solenóide para introduzir a interferência;
- X-Sinal ativo;
- Y-Sinal variável controlado.
- n-Sinal de interferência

O líquido contido no tanque de descarga (1) é transferido para o tanque graduado (11) por meio da bomba centrífuga (2) sob o controle da válvula pneumática proporcional (3). O sinal de regulação (X) que vem da unidade (9) é transmitido à válvula (3) pelo transdutor de I/P (4). A pressão aplicada ao fundo do tanque (11), que é proporcional ao nível do líquido contido no tanque, é detectada pelo transdutor de P/I (13) e é transmitida, como a variável controlada (Y), para a unidade de condicionamento (9). As válvulas de descarga (12) tornam possível a abertura manual para liberação do líquido no tanque (1). A válvula de solenóide (14) torna possível controlar um fluxo de interferência. A unidade (9) pode ser conectada a um controlador eletrônico externo (8) ou a um PC (10).

A planta também possui a unidade e controle elétrico, conforme mostra a figura 9, a qual é responsável pelas seguintes funções:

- a) Fornecimento de energia a todos os componentes elétricos da unidade;
- b) Condicionamento dos sinais elétricos para a placa de conversão AD/DA para adaptar as entradas/saídas da placa para as entradas/saídas dos transdutores e acionadores da unidade;
- c) Provisão de vários interruptores e seletores;
- d) Provisão de vários pontos de teste para verificar os sinais elétricos mais importantes por meio de um osciloscópio;
- e) Provisão de um display digital indicador de nível.

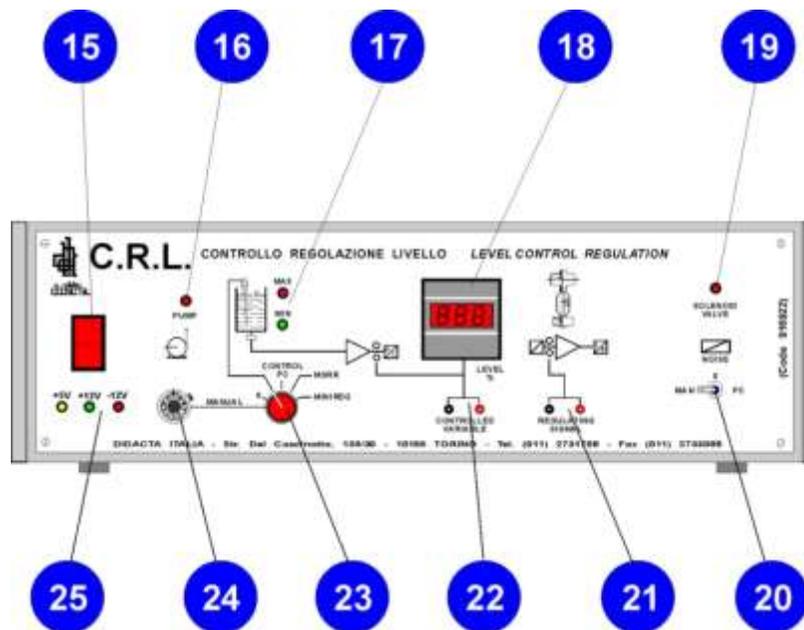


Figura 8 - Unidade Eletrônica de Controle do CRL

Fonte: Manual do CRL, página 8

Conforme a figura 8, o equipamento possui os seguintes componentes:

- 15-Comutador Principal
- 16-LED indicador de operação da bomba;
- 17-LED indicador de nível mínimo/máximo;
- 18-Visor de % de nível de líquido;

19-LED indicador para válvula de solenóide usada para introduzir a interferência;

20-Seletor de interferência de 3 posições:

0 - a válvula de solenóide é desativada e sua operação é desabilitada;

MAN (OPERADOR) - a válvula de solenóide é ativada;

PC - a operação da válvula de solenóide é controlada pelo PC;

21-Sinal de regulação, entrada analógica;

22-Variável controlada, saída analógica;

23-Seletor de Modo (*CONTROL*) de 6 posições:

- Comando manual direto da válvula proporcional (MANUAL);
- Unidade desligada (0);
- Controle por meio de sondas resistivas;
- Controle por meio de PC (PC);
- Controle por meio de MiniReg, controlador eletrônico opcional (MINIREG);
- Controle por meio de MRRP, controlador pneumático opcional (MSRR);

24-Potenciômetro multi-voltas para ajuste manual de válvula proporcional;

25-LEDS indicadores de suprimento de energia.

2.7.1 - Planta de Controle de Vazão

A planta de controle de vazão, ou CRF, foi projetada para o estudo de técnicas de controle em malha aberta ou fechada. A unidade permite a passagem de certo fluxo de água por um circuito; para esta finalidade, a água é retirada de um tanque coletor com a ajuda de uma bomba centrífuga e é recirculada por tubulações, onde seu fluxo é controlado por uma válvula pneumática proporcional. A medição do fluxo é feita por uma pequena turbina e, dependendo do valor obtido *versus* o valor desejado, pode ser possível trabalhar na abertura da válvula proporcional. Após a válvula proporcional e antes que o circuito retorne a água para o tanque coletor, é possível trabalhar em um tubo de ramificação para simular variações súbitas nas características do sistema. Em particular, a abertura de um canal auxiliar torna possível mudar a relação existente entre a posição da válvula proporcional e o fluxo que surge de tal posição. Assim, a

introdução do canal auxiliar fornece meios para estudar o comportamento do sistema na presença de uma “perturbação” que altere suas características. O propósito da ação de controle é manter o fluxo de água em circulação constante e no valor desejado (Ponto de Ajuste), até mesmo na presença de possíveis perturbações externas (ruído). Neste sistema de controle, a quantidade controlada é o fluxo e a ação de controle é feita por meio de um sinal que alcança a válvula proporcional e controla sua abertura. A planta de controle de vazão é apresentada na figura 9 e seu sinóptico é apresentado na Fig. 10.

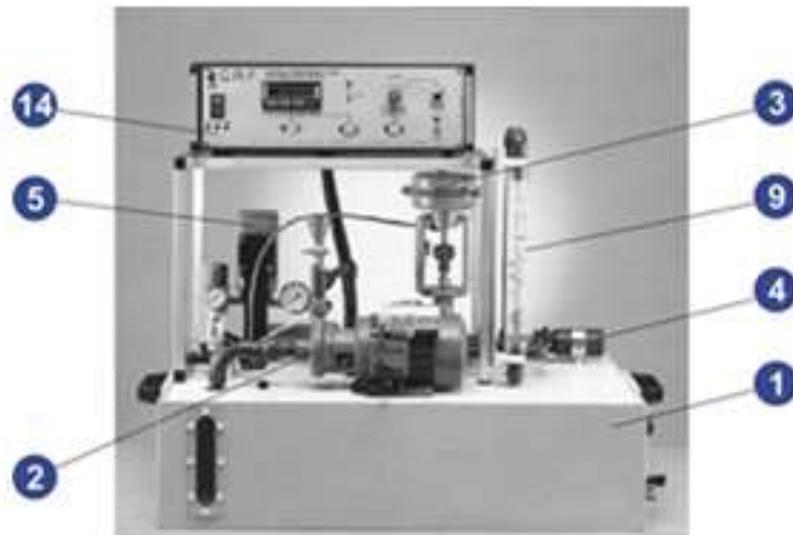


Figura 9 - Planta de Controle de Vazão

Fonte: Manual da CRF, página 4

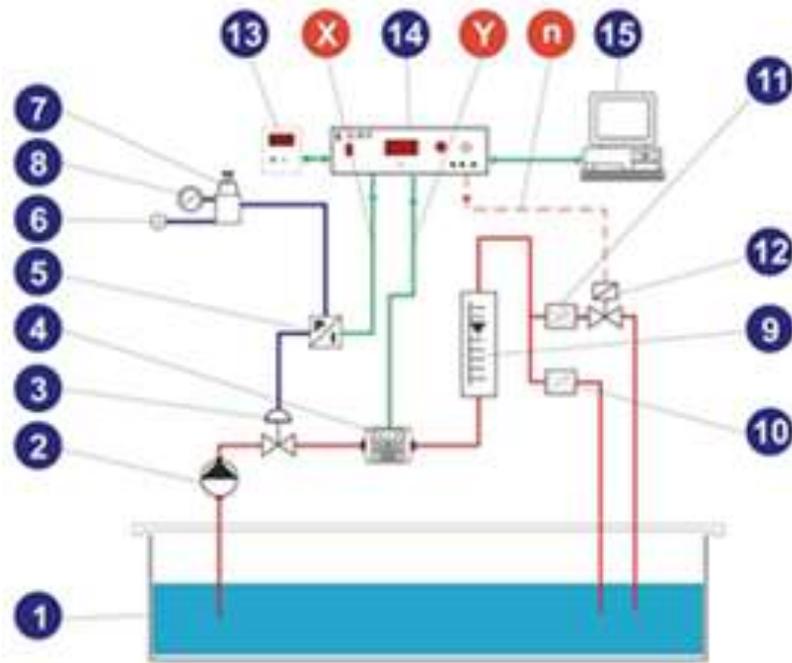


Figura 10 - Visão Sinóptica da Planta de Controle de Vazão

Fonte: Manual da CRF, página 4

Conforme as figuras 9 e 10, o equipamento possui os seguintes componentes:

- 1-Tanque coletor de água;
- 2-Bomba centrífuga;
- 3-Válvula proporcional pneumática;
- 4-Transdutor de fluxo da turbina;
- 5-Convertor de I/P (Corrente / Pressão) para o controle da válvula proporcional;
- 6-Entrada de ar comprimido;
- 7-Ajustador manual da pressão de ar comprimido;
- 8-Medidor manual da pressão de ar comprimido;
- 9-Medidor de fluxo;
- 10-Válvula manual de controle de fluxo de perturbação;
- 11-Válvula manual de controle de fluxo de perturbação;
- 12-Válvula solenóide para introduzir um fluxo de perturbação;
- 13-Controlador eletrônico opcional MiniReg;

- 14-Equipamento elétrico;
- 15-Computador pessoal;
- X-Sinal atuador;
- Y-Sinal variável controlado;
- n-Sinal de ruído.

A água contida no tanque (1) flui pelo circuito por meio da bomba (2). A válvula proporcional (3) se ajusta a um sinal de controle adequado (X), o sistema de controle determina um determinado fluxo pelo circuito. Uma válvula (10) manual torna possível alterar características do sistema para variar o efeito produzido no fluxo. A válvula solenóide (12) torna possível conectar o circuito a um canal auxiliar para recuperar certa quantidade do fluxo de água no tanque coletor. Esta característica pode ser usada para modificar a configuração geral do sistema e examinar a operação do controlador como uma função de tais alterações. O canal auxiliar também é equipado com uma válvula manual (11), que é semelhante à válvula (10) e torna possível efetuar pequenas variações de fluxo, causado distúrbios no sistema. Imediatamente após a válvula proporcional, existe uma pequena turbina (4) que mede o fluxo de água. A turbina gera um sinal elétrico de onda quadrada, cuja frequência é proporcional ao fluxo. Esta onda quadrada é convertida em um valor de tensão por um conversor especial de F/V (frequência / tensão) contido no equipamento elétrico, sendo que o valor da tensão representando a variável controlada (Y) é transmitido ao controlador.

A planta também possui a unidade e controle elétrico, conforme mostra a figura 11, a qual é responsável pelas seguintes funções:

- a) Alimentação de todos os componentes elétricos da unidade;
- b) Condicionamento dos sinais elétricos para e da placa de conversão AD/DA para adaptar as entradas da placa, saídas para as entradas, saídas dos transdutores e atuadores da unidade;
- c) Fornecimento de vários interruptores e seletores;
- d) Fornecimento de vários pontos de teste para verificar os sinais elétricos mais importantes por meio de um osciloscópio;
- e) Fornecimento de display indicador de fluxo digital.

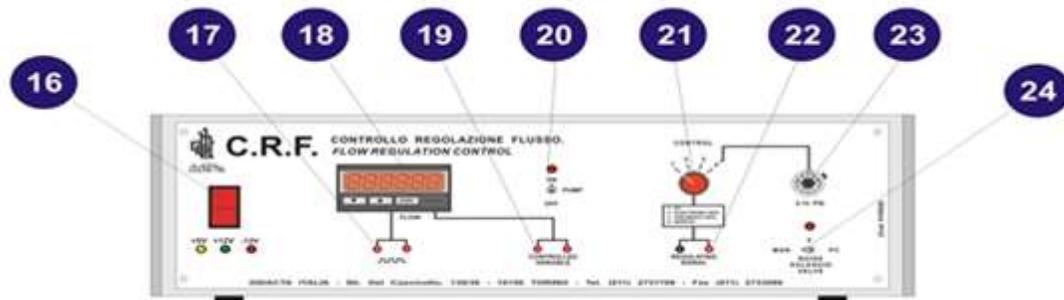


Figura 11 - Unidade Eletrônica de Controle do CRF

Fonte: Manual do CRF, página 6

16-O interruptor liga-desliga da unidade, com LEDs indicadores para a tensão da fonte de alimentação;

17-Pontos de teste para o sinal de onda quadrada gerado pela turbina;

18-Unidade de vídeo, mostrando o fluxo de água da taxa medida (em l/h);

19-Saída auxiliar do sinal que corresponde à variável controlada, Y (0 - 5V proporcional ao fluxo medido 0 - 100 l/h);

20-Interruptor liga-desliga com LED indicador de operação;

21-Seletor com modo operacional de quatro posições (*CONTROL*):

1-Controle pelo PC.

2-Controle por meio de regulador eletrônico externo (MiniReg).

3-Controle por meio de regulador pneumático (MiniReg).

4-Comando manual direto da válvula proporcional.

22-Entrada auxiliar de sinal do controle X (0 - 5V);

23-*Trimmer* para o comando manual direto da abertura da válvula proporcional (3 - 15 psi);

24-Seletor de ruído de 3 posições:

0 - a válvula solenóide é desativada e sua operação é interrompida.

MAN - a válvula solenóide é ativada.

PC - a operação da válvula solenóide é controlada pelo PC.

3 - DESENVOLVIMENTO DA TELA DE SUPERVISÓRIO

Nesta etapa serão apresentadas as construções e configurações das telas de supervisão dos controles das plantas de nível e vazão. Utilizamos o manual do usuário do *software*, disponível no *site* do fabricante.

Primeiro passo – Ao abrirmos o Elipse Scada, acessamos a aba: arquivo / criar / nova aplicação. A tela que se abre é a nossa área de trabalho.

Segundo passo – inserimos os *Bar Graph* através do botão indicado pela seta, como mostra a figura 12, os quais são responsáveis por indicar os valores das variáveis controladas, manipuladas e o *Set-Point*.

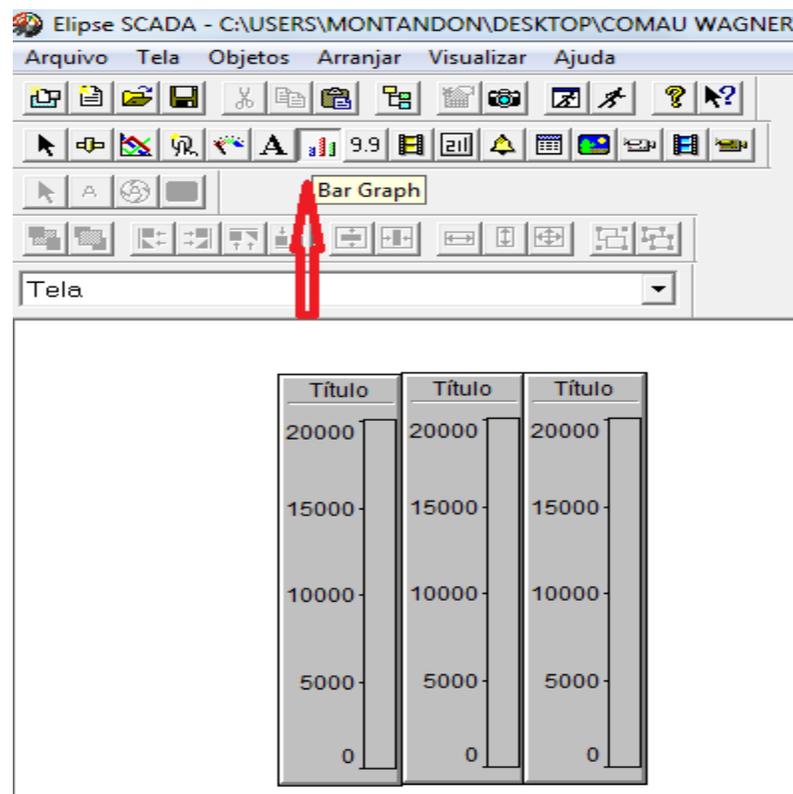


Figura 12 - Tela Inserindo Gráfico de Barras

Fonte: Autor

Terceiro passo – em *Set-Point*, clicamos no botão indicado pela seta, como mostra a figura 13, assim abre-se um retângulo cujo tamanho e formato se estende conforme movimentação do mouse.

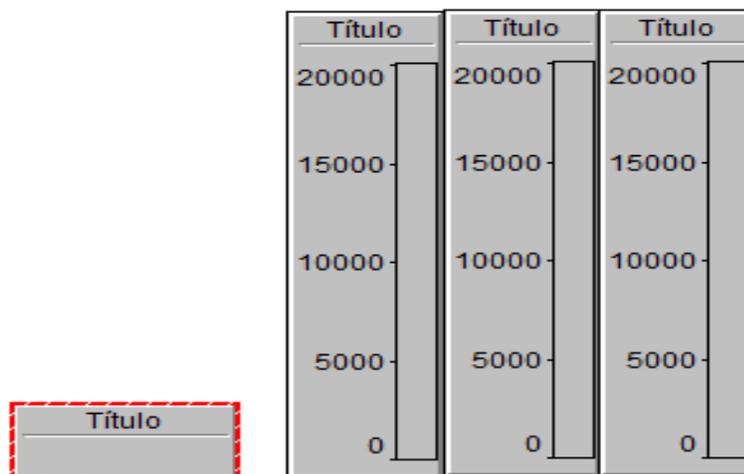
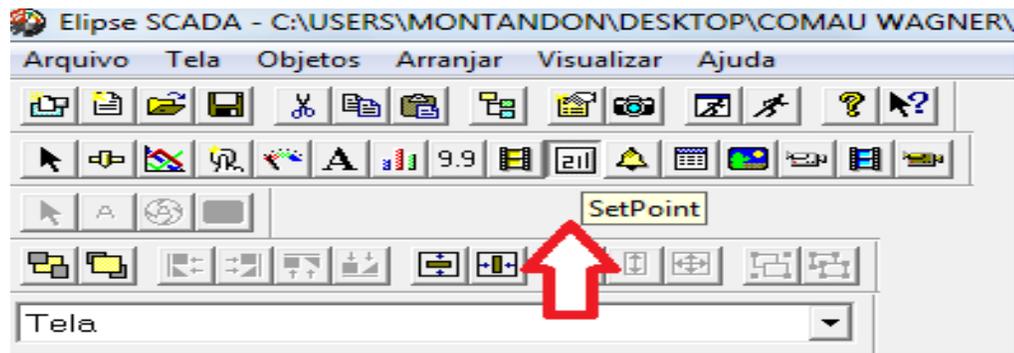


Figura 13 - Tela Inserindo Botão de Set-Point

Fonte: Autor

Quarto passo – Inserimos o *Trend Graph* através do botão indicado pela seta na figura 14. A tela de gráficos criada foi o responsável por mostrar os valores das variáveis controladas, manipuladas e o valor de *Set-Point* do sistema, em função do tempo.

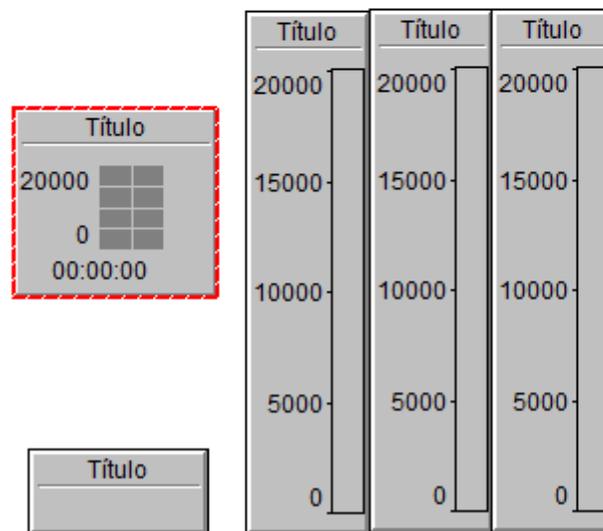
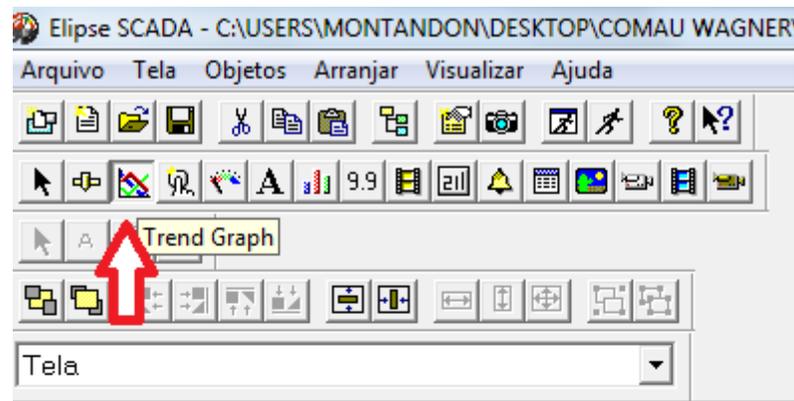


Figura 14 - Tela Inserindo o Gráfico de Valores das Variáveis e Set-Point

Fonte: Autor

Estes foram os quatro passos básicos para criação das telas nas quais utilizamos para monitorar as variáveis e setar os valores desejados para que as variáveis do sistema se estabilizem. Criados os botões e gráficos de indicação, criaremos os TAGs ligando-os às variáveis do sistema, as quais, posteriormente, ligamos a cada um dos ícones criados anteriormente.

3.1 - Criação das TAGS e Configurações de Endereços de Comunicação

Nesta etapa é mostrado os passos para a criação dos TAGs, os quais foram ligados aos gráficos de indicação e *Set-Point* de valores. Estes TAGs receberam um endereço corresponde a cada uma das variáveis do controlador, cujas configurações serão mostradas no próximo capítulo. Para a criação fizemos o seguinte:

Clicamos no ícone *organizer* na barra de ferramentas, no *menu* superior da tela do programa, conforme mostra a figura 15, ou então com o comando ALT + O entramos na tela de onde tem a raiz das configurações da aplicação que criamos.

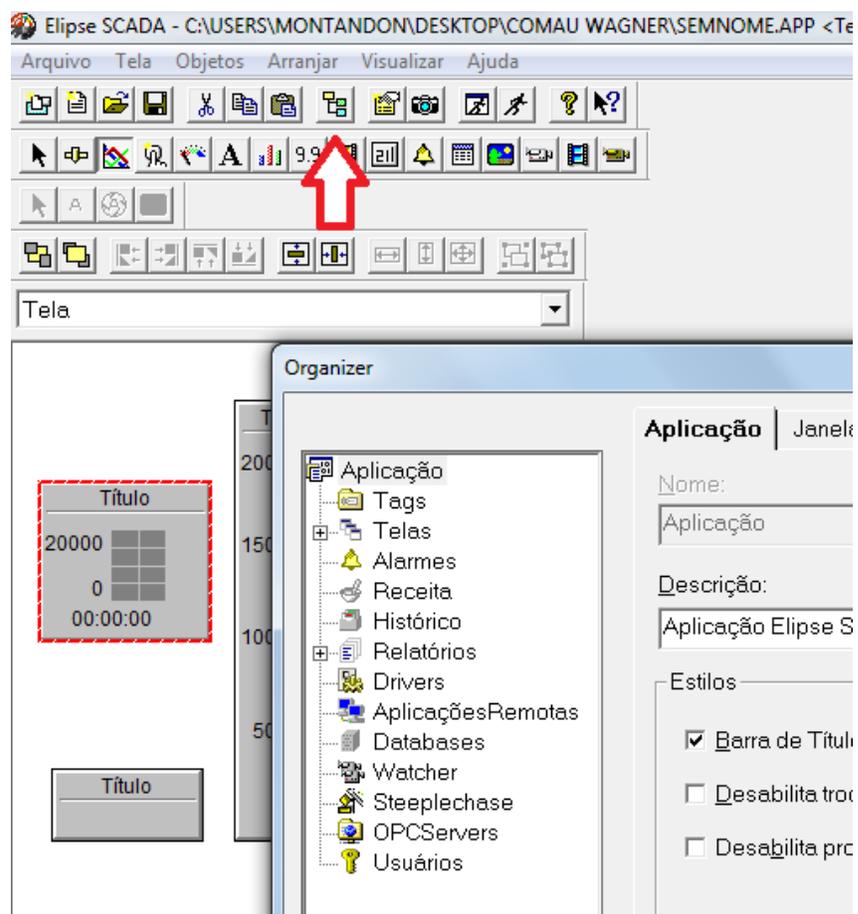


Figura 15 - Tela de Criação de TAGS

Fonte: Autor

Em *Tags*, clicamos no botão: Novo Tag / Tag PLC. Nomeamos de *SET_POINT* e com um duplo *click* sobre o ícone criado abre-se uma janela como mostra no canto direito da figura 16.

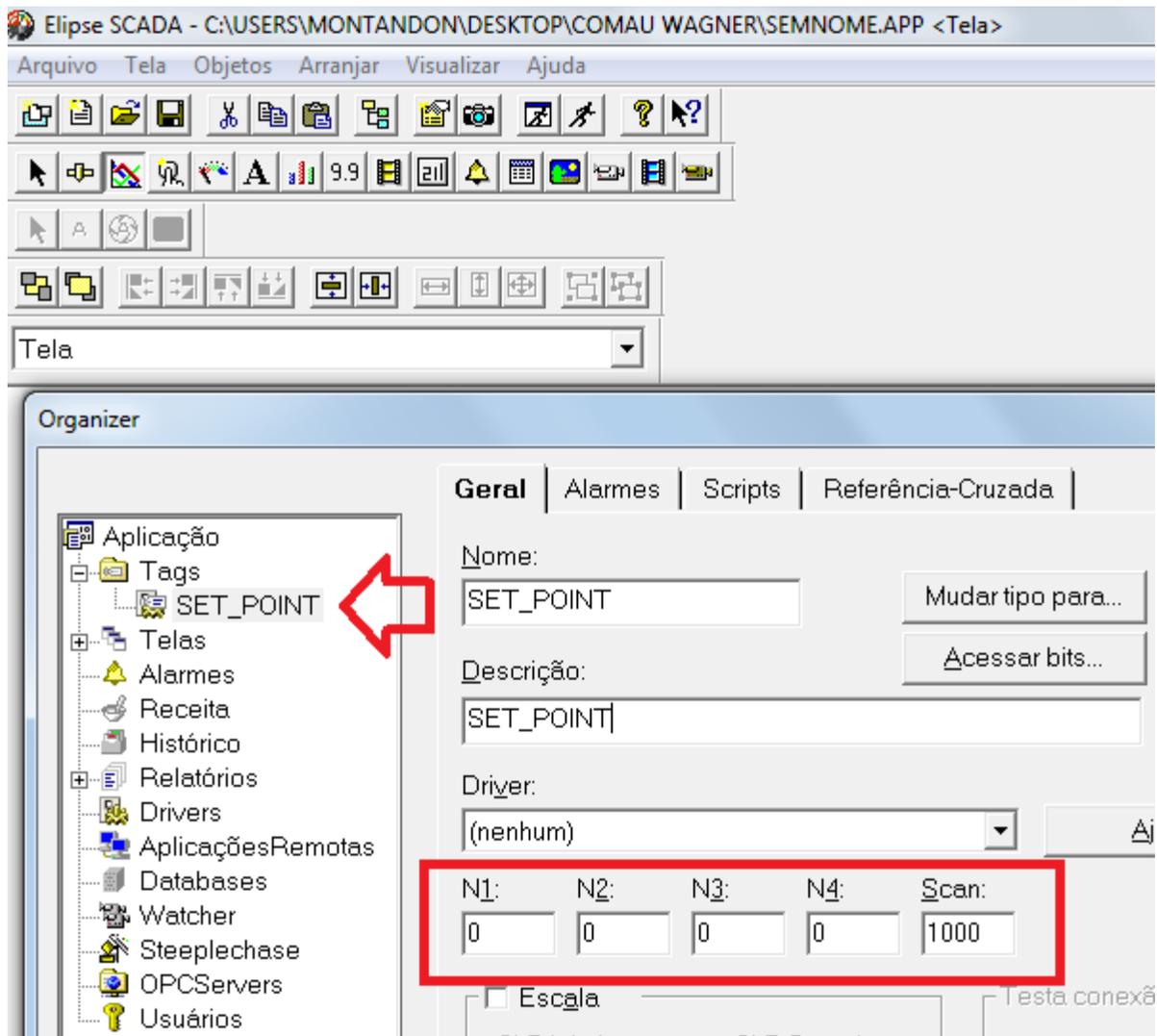


Figura 16 - Tela de Criação da TAG e Configuração de Comunicação

Fonte: Autor

Conforme o retângulo em destaque mostrado na figura 16, estão os campos onde inserimos os valores referentes aos seguintes itens:

N1 - Endereço do controlador escravo.

N2 - Função de *Holding Register* do Modbus.

N3 - Não usado.

N4 - Endereço referente ao *Holding Register*, sendo ele o responsável por armazenar o valor das variáveis de leitura e escrita do sistema.

Scan - É o tempo que o *driver* leva para atualizar os dados lidos do controlador.

Em seguida, criamos as seguintes variáveis, conforme mostra o retângulo no canto superior esquerdo da figura 17.

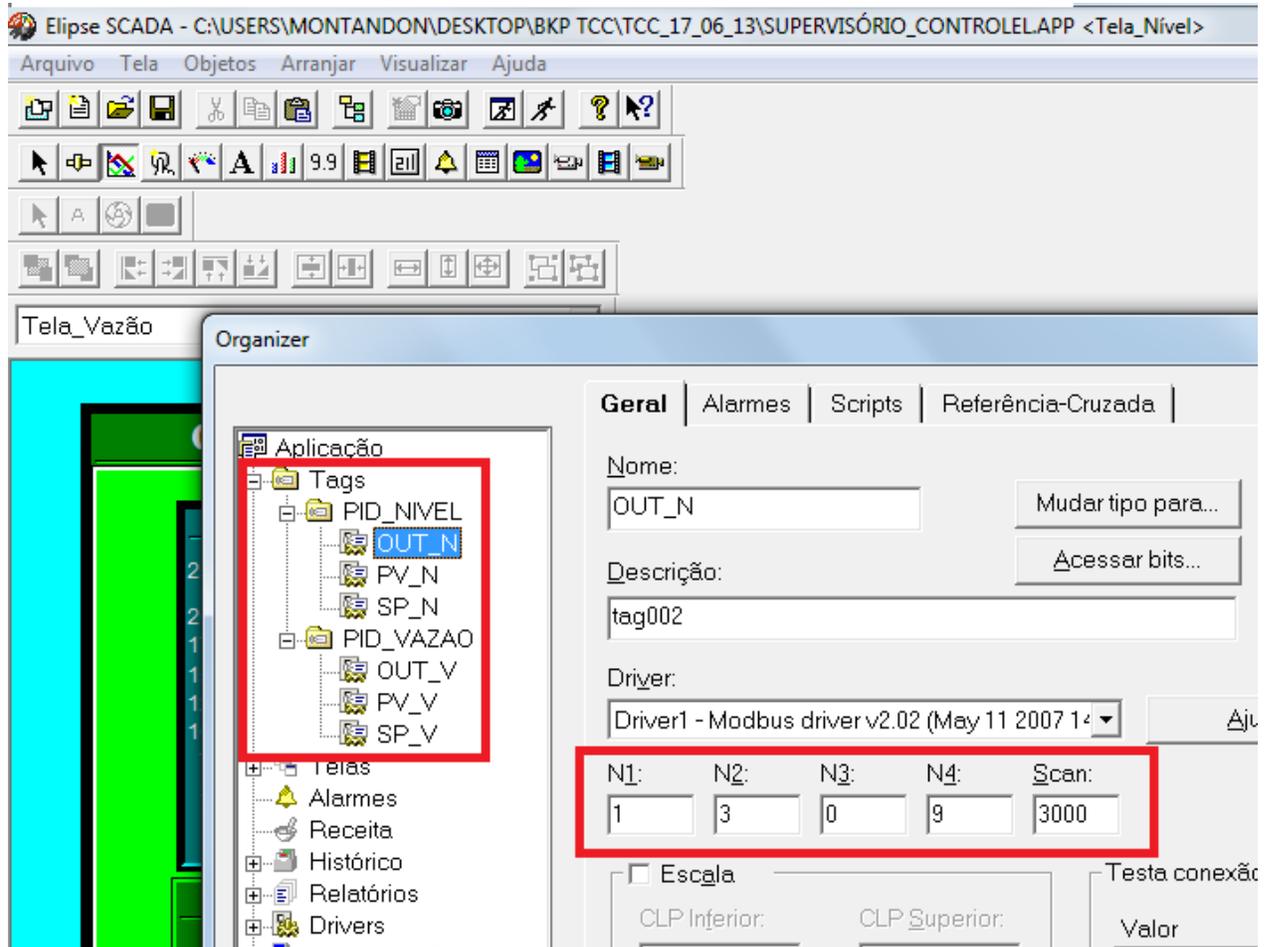


Figura 17 - Tela de Criação das TAGs e Configurações

Fonte: Autor

Conforme visto, criamos a pasta PID_NÍVEL, na qual estão as TAGs referente à planta de controle de nível. Criamos as seguintes TAGs: OUT_N, PV_N e SP_N. A pasta PID_VAZÃO diz respeito às TAGs das variáveis da planta de controle de vazão.

Criamos as seguintes TAGS: OUT_V, PV_V e SP_V.

Em seguida, configuramos os campos mostrados dentro do retângulo no canto direito da figura 17, para cada uma das TAGs criadas, conforme a tabela 1.

Tabela 1 - Configurações de Endereços e Tempo de Varredura

	N1	N2	N3	N4	SCAN
OUT_N	1	4	0	9	500
PV_N	1	4	0	19	500
SP_N	1	4	0	18	500
OUT_V	2	4	0	9	500
PV_V	2	4	0	19	500
SP_V	2	4	0	18	500

Fonte: Autor

Conforme visto, a única variação entre cada uma das TAGs é o campo N1, que é o endereço do componente escravo, no nosso caso, o controlador. Utilizamos um controlador para cada planta. O campo N2 refere-se ao tipo de registrador que utilizamos no sistema, que no caso é o *Holding Register* . O campo N3 não tem utilização. O campo N4 diz respeito ao endereço de registrador, onde utilizamos apenas os dois últimos números, uma vez que o registrador *Holding Register* utiliza endereços que vão de 400001 até 465535, assim variamos apenas os dois primeiros números. O campo *scan* foi configurado para que as atualizações de leituras de dados de cada uma das TAGs ocorressem a cada 500 milissegundos.

3.2 - Criando Links entre as TAGs, Gráficos e Botões das Telas de Supervisório

Depois de configurados os endereços das variáveis que utilizamos, relacionamos os links das TAGs criadas e os gráficos e botões que criamos no primeiro passo. Este procedimento definiu a indicação da variação das variáveis do sistema.

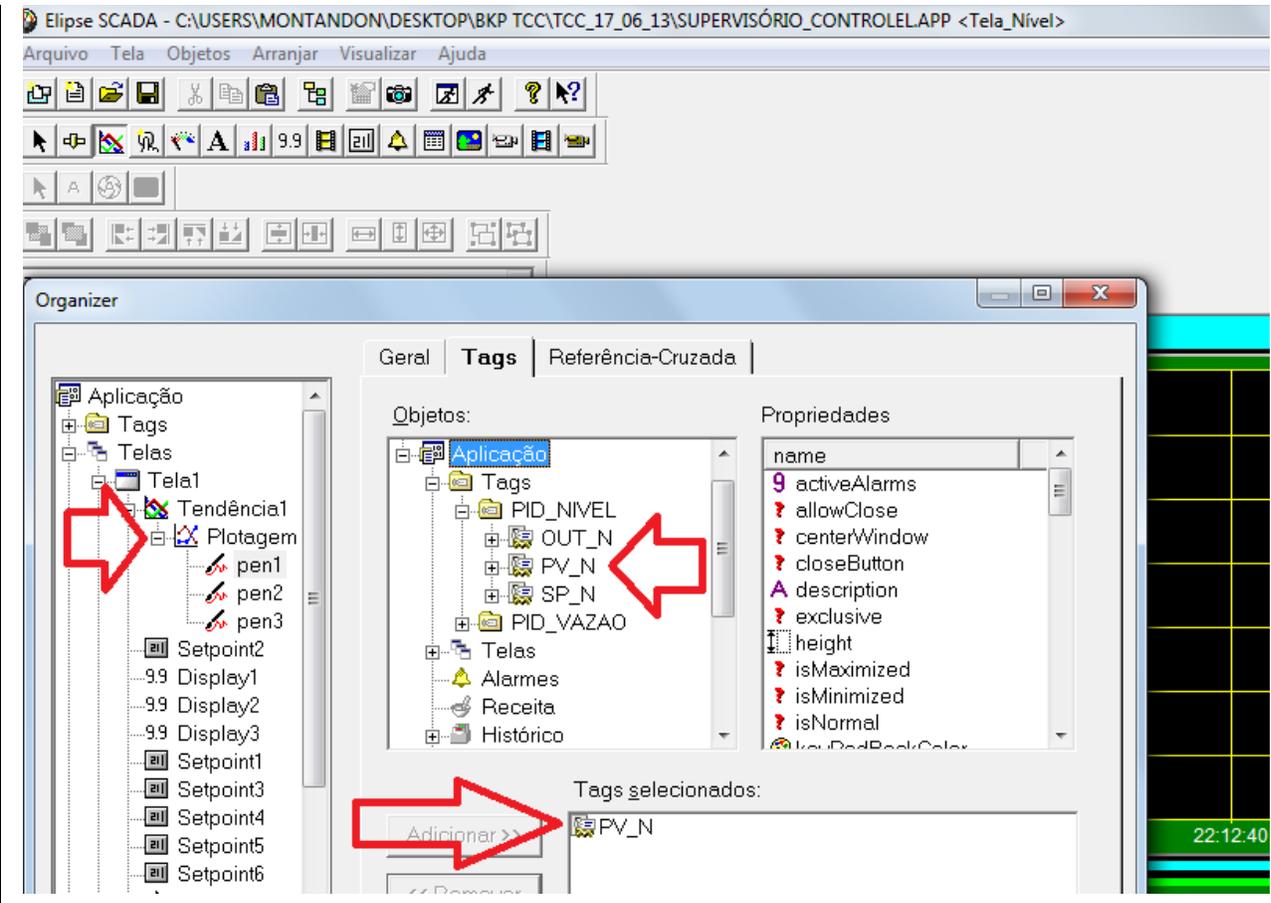


Figura 18 - Tela de Relação de TAG ao Gráfico de Tendência

Fonte: Autor

Como visto na figura 18, foi feita uma relação entre o TAG PV_N à uma pena ou linha, a qual gera uma linha contínua no gráfico, ou seja, a medida que a variável PV_N for se alterando é mostrado no gráfico uma linha com variação na amplitude em função do

tempo. Este procedimento de associação de TAGs foi análogo a todas outras TAGs criadas.

4 - CONFIGURAÇÃO DO ARDUINO COM O SISTEMA

Em sistemas automatizados que utilizam o protocolo Modbus, utiliza-se o CLP (Controlador Lógico Programável) que processa todas as informações e retorna uma resposta de acordo com as configurações realizadas no sistema. No caso desse trabalho, utilizamos o Arduino, que trata-se de um pequena placa de circuito constituída de entradas e saídas analógicas e digitais, e um microcontrolador, que processa os dados de entrada e saída da placa. Por se tratar de um laboratório acadêmico, é admissível o uso de placas como esta do Arduino.

4.1 - Arduino

O Arduino é um microcontrolador de placa única e um conjunto de *software* para programá-lo. Uma plataforma de desenvolvimento *open source* baseada em software e *hardware* fáceis de manusear. Utiliza em sua programação uma linguagem (baseada em processamento), que é muito parecida com a linguagem C. Sua plataforma é livre para ser criada e modificada em qualquer lugar, mas com a ideia de que todos disponibilizem seus conhecimentos e inovações, mantendo assim o desenvolvimento de novas criações (NISHI, 2012). No nosso trabalho estamos utilizando o modelo UNO, mostrado na figura 19.



Figura 19 - Arduino UNO

Fonte: www.arduino.cc

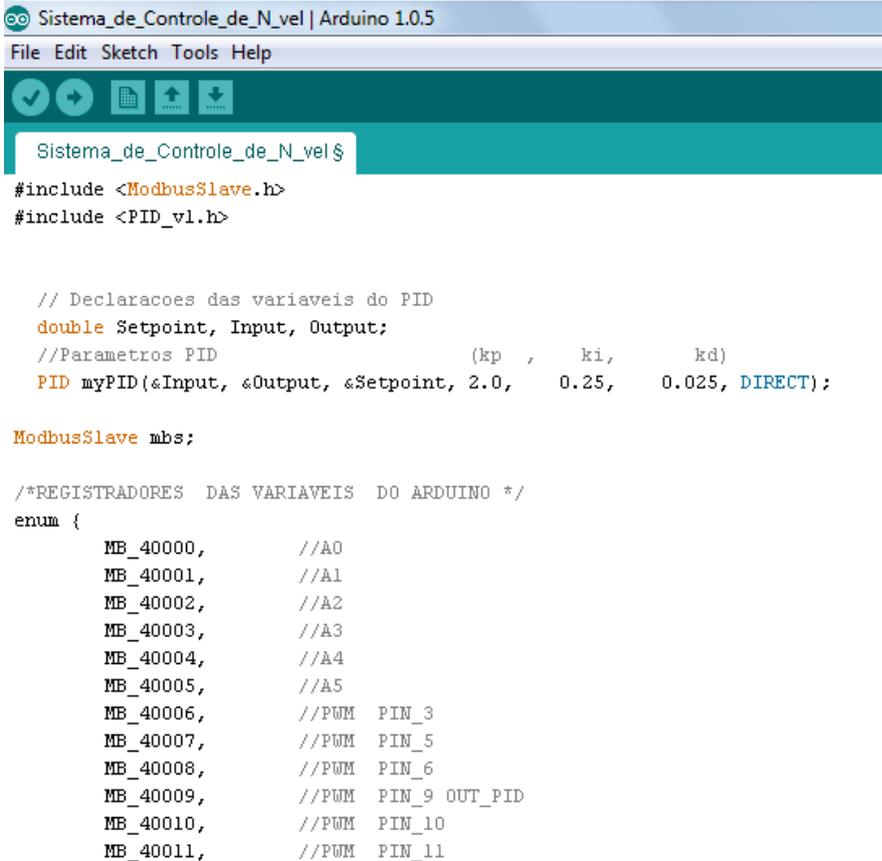
O Arduino UNO possui 14 entradas / saídas digitais (das quais 6 podem ser usadas como saídas PWM), 6 entradas analógicas, um ressonador cerâmico de 16 MHz, uma conexão USB, um conector de alimentação, e um botão de reset. Na tabela 2 são mostradas as suas características.

Tabela 2 - Características do Arduino UNO

Microcontrolador	ATmega328
Tensão de funcionamento	5V
Tensão de entrada(recomendado)	7-12V
Tensão de entrada(limites)	6-20V
Digital pinos de I / O	14 dos quais 6 fornecem uma saída <i>PWM</i>
Entradas Analógicas Pinos	6
<i>DC Current I / O Pin</i>	40 mA
Corrente <i>DC</i> para <i>Pin 3.3 V</i>	50 mA
Memória <i>Flash</i>	32 KB (ATmega328), dos quais 0.5 KB utilizado pelo <i>bootloader</i>
<i>SRAM</i>	2 KB (ATmega328)
<i>EEPROM</i>	1 KB (ATmega328)
Velocidade do relógio	16 MHz

Fonte: Adaptado de www.arduino.cc

O fabricante do Arduino disponibiliza gratuitamente, para *download* em seu *site*, a plataforma para o desenvolvimento do código-fonte, código este que será compilado e gravado na memória do Arduino. Na figura 20 é mostrado o ambiente de desenvolvimento e compilação do código-fonte.



```

Sistema_de_Controlde_de_N_vel | Arduino 1.0.5
File Edit Sketch Tools Help

Sistema_de_Controlde_de_N_vel $
#include <ModbusSlave.h>
#include <PID_v1.h>

// Declaracoes das variaveis do PID
double Setpoint, Input, Output;
//Parametros PID (kp , ki, kd)
PID myPID(&Input, &Output, &Setpoint, 2.0, 0.25, 0.025, DIRECT);

ModbusSlave mbs;

/*REGISTRADORES DAS VARIAVEIS DO ARDUINO */
enum {
    MB_40000, //A0
    MB_40001, //A1
    MB_40002, //A2
    MB_40003, //A3
    MB_40004, //A4
    MB_40005, //A5
    MB_40006, //PWM PIN_3
    MB_40007, //PWM PIN_5
    MB_40008, //PWM PIN_6
    MB_40009, //PWM PIN_9 OUT_PID
    MB_40010, //PWM PIN_10
    MB_40011, //PWM PIN_11
}

```

Figura 20 - Plataforma de Desenvolvimento do Código-Fonte

Fonte: Autor

4.2 - Código-Fonte do Protocolo Modbus

O Arduino é o dispositivo responsável por receber comandos do supervisor e enviar comandos aos atuadores do sistema, no caso um drive de controle PWM. Ele também recebe os valores das variáveis lidas nas plantas, ou seja, valores que varia de

0 a 5 volts, proporcionais à variação do nível do tanque ou à vazão do fluido na tubulação. Em razão destes fatores, o Arduino é o responsável por efetuar o controle do sistema. O código-fonte utilizado foi desenvolvido por Juan Pablo Zometa, estudante PhD do Instituto de Engenharia de Automação e assistente de pesquisa na Universidade de Magdeburga, Alemanha. Ele publicou este código-fonte em 2010, disponibilizando para livre uso acadêmico, proibindo-o para uso comercial. Esse código realiza as funções impostas pelo protocolo Modbus, porém ele foi desenvolvido baseado apenas na utilização de *Holding Registers* , os quais são registradores utilizados tanto para armazenar valores de escrita quanto leitura. Juntamente com o código foi utilizada uma biblioteca PID (Proporcional, Integral e Derivativo), a qual foi incorporada no código-fonte. A biblioteca PID tem como função fazer com que o Arduino mantenha o controle do sistema estável, ou seja, manter os valores de *Set-Point* , tanto do nível do tanque quanto a vazão do fluido na tubulação, próximos ou iguais aos valores determinados, assim como uma resposta rápida e sem oscilações de valores. A biblioteca PID é disponibilizada pelo fabricante do Arduino, também encontrada para *download* no *site* .

5 - MONTAGEM E RESULTADOS

A montagem foi realizada no laboratório de sistemas de controle do CEFET-MG, campus Araxá. Para realizar a comunicação do Arduino com o *notebook* foi necessária a utilização de um conversor USB/RS-485 e um *shield* RS-485. A utilização destes componentes se deu em razão da utilização de um *notebook* , o qual possui portas de comunicação USB.

5.1 - Conversor USB/RS-485 e *Transceiver* TTL/RS-485

Os conversores e *transceivers* são muito comuns em redes industriais, uma vez que nesta área existem inúmeros meios físicos e protocolos de comunicação. Os conversores vieram para atender as indústrias de forma que uma vez adquirido um equipamento com um determinado protocolo ou meio de comunicação, não ficasse escravo do fabricante deste equipamento, tendo assim maior flexibilidade quanto à

utilização de outros meios físicos e protocolos. No caso de nosso trabalho utilizamos um *transceiver* TTL/RS-485, mostrado na figura 21 e um conversor USB/RS-485, mostrado na figura 22.

a) *Shield* RS-485 - O *shield* contém um chip *transceiver* RS-485 que faz o trabalho de conversão dos dados de TTL para RS-485 e vice versa. Os LEDs indicam o fluxo do tráfego de dados na rede; os bornes são utilizados para ligar os cabos do barramento da rede; os *jumpers* são utilizados para selecionar uso de resistores de terminação e bias da rede, selecionar o uso de uma porta de comunicação serial física do ATmega do Arduino ou usar uma porta *soft-serial*. Para a comunicação pela *hard-serial* (HS) são utilizados os pinos digitais 0 e 1 do Arduino. Para a comunicação usando *soft-serial* (SS) são usados os pinos digitais 2 e 3 do Arduino. O pino digital 6 é usado para comutação entre os modos de transmissão e recepção do *transceiver* do *shield*.



Figura 21 - Shield RS-485

Fonte: www.labdegaragem.org

b) O conversor USB/RS-485 usa o chip FT232RL para criar um dispositivo serial em qualquer microcomputador (porta COM) que pode ser acessada pelo usuário ou programas. O conversor possui um chip *transceiver* RS-485, assim como os *jumpers*, resistores, LEDs e bornes presentes no *shield* RS-485.



Figura 22 - Conversor USB/RS-485

Fonte: www.labdegaragem.org

5.2 - Ligações do Conversor USB/RS-485 com o Barramento Principal

Para a ligação entre o conversor USB/RS-485 e o *notebook* utilizamos um cabo USB / mini USB conforme mostra a figura 23.



Figura 23 - Cabo USB / Mini USB

Fonte: Autor

A ligação entre o conversor USB/RS-485 e o *shield* RS-485 foi feita com um cabo do tipo par trançado sem blindagem (UTP). A topologia de rede utilizada em nosso trabalho foi a de barramento, na qual o cabo UTP que foi conectado entre o conversor

USB/RS-485 e o *shield* RS-485, sendo ele o barramento principal. O esquema de ligação pode ser representado conforme figura 24.

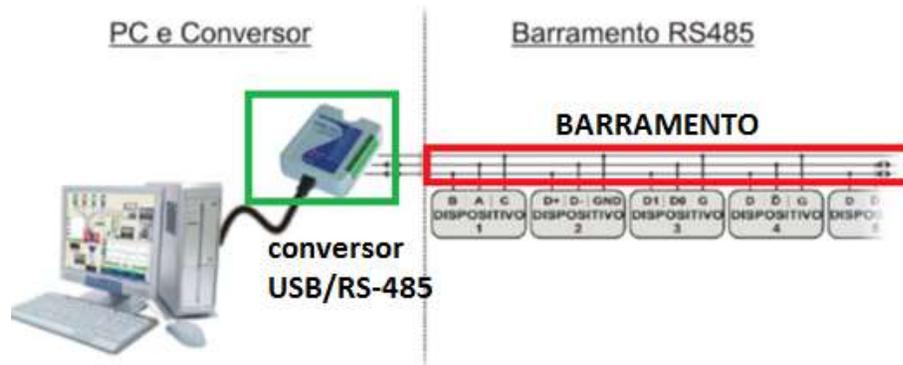


Figura 24 - Esquema de Ligação da Rede

Fonte: <http://www.kronweb.com.br>

Dentro do retângulo em destaque à direita temos o barramento principal, o qual pode chegar até 1200 metros, valor máximo estipulado quando utilizamos o protocolo Modbus. Ligado ao barramento, temos os dispositivos, que no caso foram os *shields* RS-485 conectados aos Arduinos. Cada um destes dispositivos foi utilizado como controlador de cada uma das plantas, a de nível e a de vazão. Dentro do retângulo em destaque à esquerda temos a representação do conversor USB/RS-485, o qual foi interligado ao *notebook* com um cabo USB / mini USB de comprimento não superior a 1,8 metros. A seguir mostra as ligações realizadas entre o *notebook*, conversor USB/RS-485 e os dispositivos controladores com os seus *shields*. Conforme as figuras a seguir, realizamos as seguintes ligações:

a) O pino A0 do Arduino, mostrado na figura 25, indicado pela seta, foi ligado ao pino da unidade eletrônica de controle, que diz respeito aos valores da variável controlada, o qual fornece um valor analógico que varia de 0 a 5V, proporcional à variação do nível ou vazão mostrado na figura 26, indicado pela seta.

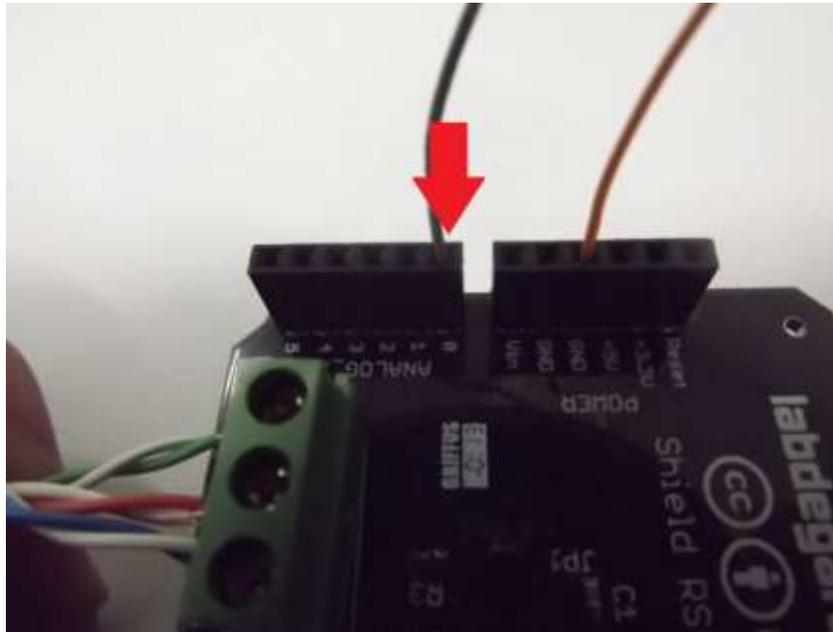


Figura 25 - Entrada A0 do Arduino

Fonte: Autor

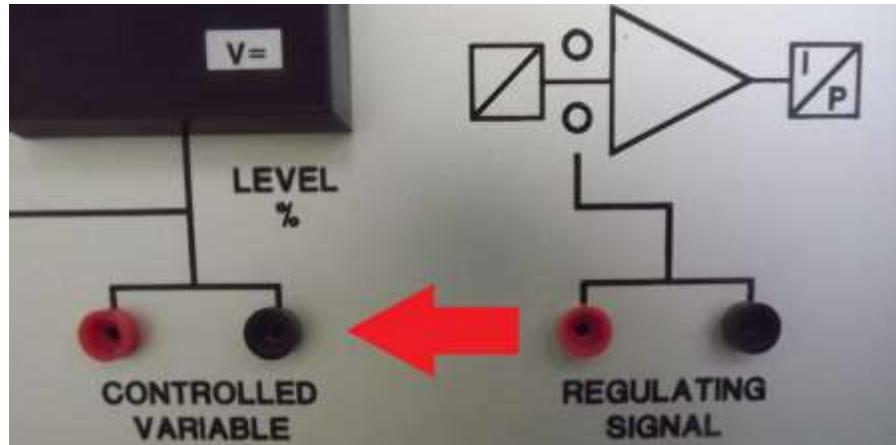


Figura 26 - Bornes de Saída de Sinal da Variável Controlada

Fonte: Autor

b) O pino 9 do Arduino, que fornece um valor de 0 a 5V através de PWM, é mostrado na figura 27, indicado pela seta, o qual foi ligado à entrada de sinal analógico, também da unidade eletrônica de controle, mostrada na figura 28, indicado pela seta. O Arduino fornece

um valor de tensão que varia de 0 a 5V, proporcional às necessidade do sistema para que a variável controlada se estabilize.

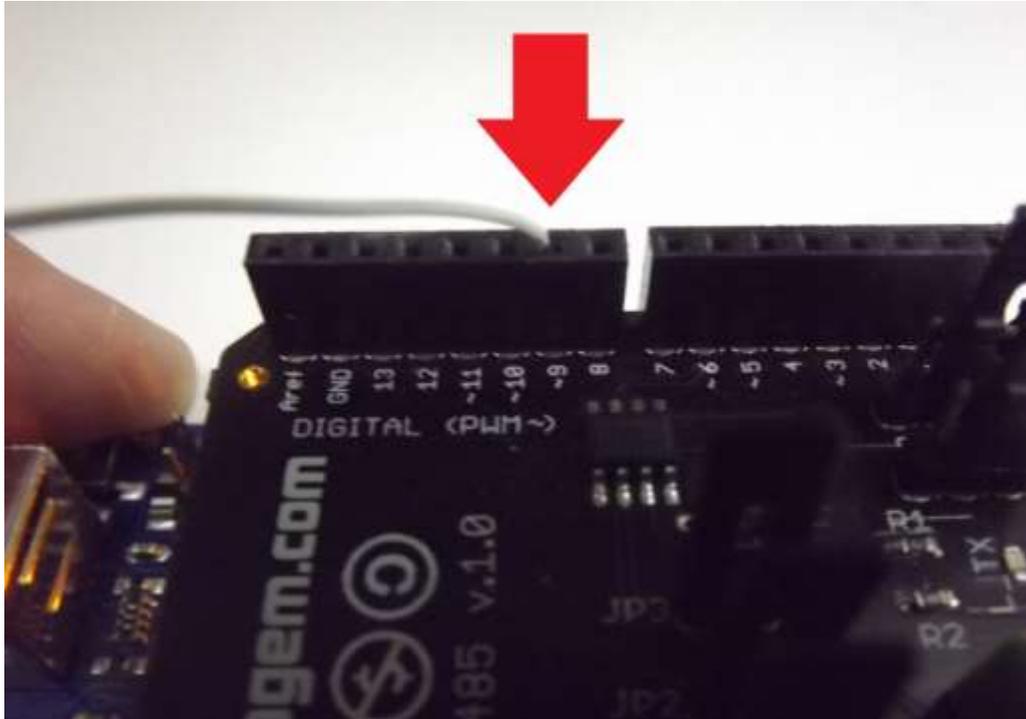


Figura 27 - Saída PWM no pino 9 do Arduino

Fonte: Autor

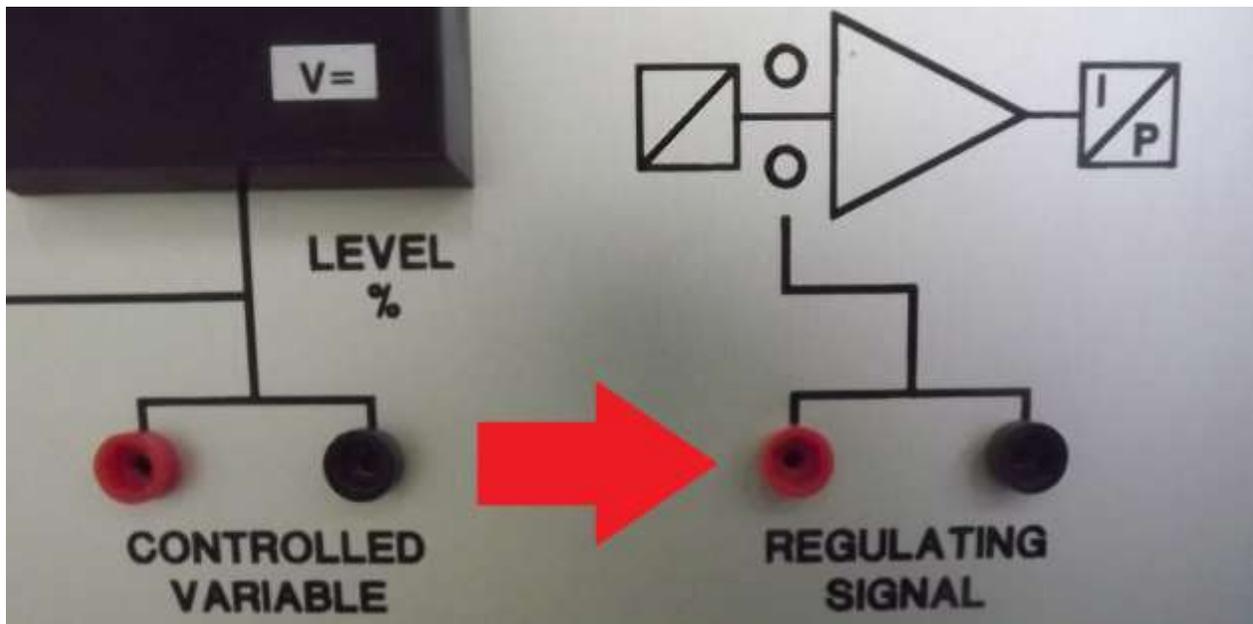


Figura 28 - Bornes de Entrada do Sinal Regulador

Fonte: Autor

Essas ligações foram realizadas nas plantas de controle de vazão e nível. A figura 29 e 30 mostram as telas de supervisão das plantas de controle de nível e vazão.

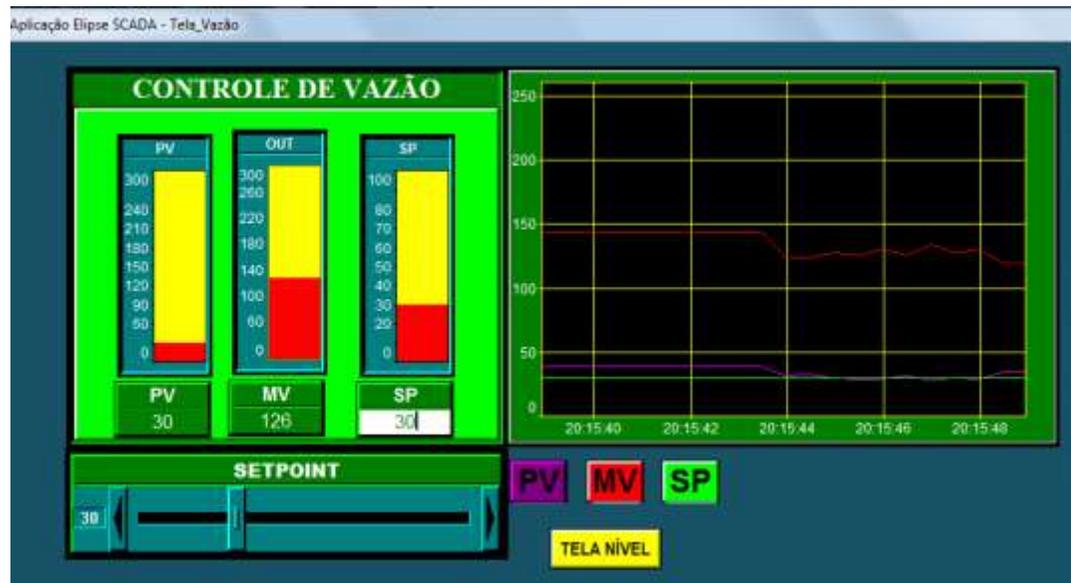


Figura 29 - Tela de Supervisão do Controle de Vazão

Fonte: Autor



Figura 30 - Tela de Supervisório do Controle de Vazão

Fonte: Auto

5.3 - Resultados

Depois de ligado todo o sistema, obtivemos êxito no que diz respeito à comunicação do supervisório instalado no *notebook*. As duas plantas de controle transmitiram os valores lidos por seus sensores, tanto de nível quanto de vazão, os quais foram visualizados na tela de supervisório do *notebook*. Utilizamos valores de 2 para o ganho proporcional, de 0.25 para ganho integral. Estes valores são padrões da biblioteca PID do Arduino. Inserimos um valor de *Set-Point* de 30% em ambas as plantas e aguardamos até que se estabilizassem. Posteriormente, ajustamos o valor do *Set-Point* em 50% em cada uma das plantas. Na planta de controle de vazão o sistema se estabilizou rapidamente. Já na planta de controle de nível tivemos uma relativa demora no que diz respeito ao tempo para que o sistema de estabilizasse. Não foi necessária a utilização de métodos de sintonia PID, pois encontramos valores de ganhos satisfatórios sem nenhuma dificuldade, baseado apenas na tentativa e erro. Tais valores de ganho utilizados em cada uma das plantas são mostrados no código-fonte, em documentos anexos, assim como as linhas de programação de controle PID e comunicação de cada uma das plantas.

6 - CONCLUSÃO

Neste trabalho, atingimos todos os objetivos propostos. A comunicação entre os componentes controladores, tanto da planta de controle de nível quanto de controle de vazão e o computador supervisor, ocorreu sem nenhum problema de atraso de transmissão ou recepção de dados, erros de leitura e escrita ou travamento do aplicativo supervisor. A planta de controle de vazão respondeu de forma satisfatória, não apresentando nenhum tipo de instabilidade. No controle da planta de nível foram utilizados ganhos proporcional e integral diferentes dos utilizados na planta de controle de vazão, uma vez que são sistemas com comportamentos diferentes. Esses valores de ganho foram ligeiramente maiores que os utilizados na planta de vazão, com exceção do ganho derivativo que se manteve, assim obtemos uma resposta do sistema satisfatória o qual se estabilizava rapidamente. Tais valores foram encontrados sem nenhuma dificuldade baseando apenas na tentativa e erro de forma descartamos a necessidade da utilização de métodos de sintonia PID para encontrar os ganhos. Vimos também que estes sistemas podem ser adaptados às plantas de controle de temperatura e pressão, também disponíveis no laboratório do CEFET-MG, uma vez que os dispositivos reguladores são similares aos utilizados nas plantas de controle de nível e vazão, necessitando apenas da alteração nos valores dos ganhos K_p , K_i e K_d no código-fonte do Arduino, pois cada planta possui características diferentes. Por fim, notamos que o sistema oferece flexibilidade de forma que o usuário possa desenvolver um algoritmo baseado em algum outro tipo controle, necessitando apenas atribuir às variáveis utilizadas aos endereços de *Holding Registers* disponibilizados no código-fonte, além da necessidade de configurar as TAGs na estação de trabalho.

REFERÊNCIAS

- GUEDES, L.A. **Classificação das redes para automação industrial**. 2005. Disponível em: < http://www.dca.ufrn.br/~affonso/DCA0447/aulas/rai_cap3_part1.pdf>
- MORAES, C. C. de; CASTRUCCI, P. L. **Engenharia de Automação Industrial**. 2ª ed. LTC, 2007.
- TANENBAUM, A. S. **Redes de Computadores**. 3ª. ed. Editora Campus, 1997.
- GUTIERREZ, V.M .R e PAN, K. S. S. **Complexo Eletrônico: Automação do Controle Industrial**, 2008. Disponível em: <http://www.bndes.gov.br/SiteBNDES/export/sites/default/bndes_pt/Galerias/Arquivos/conhecimento/bnset/set2807.pdf>
- GOEKING, W. **Da Máquina a Vapor aos Softwares de Automação**, 2010. Disponível em:<<http://www.osetoreletrico.com.br/web/component/content/article/57-artigos-e-materias/343-xxxx.html>>
- NETO, L. **Sistemas Embarcados – Uma Breve Introdução**, 2010. Disponível em: <<http://saladatecnologica.wordpress.com/2010/01/06/sistemas-embarcados-uma-breve-introducao>>
- <<http://www.smar.com/brasil/artigostecnicos/artigo.asp?id=48>>
- <www.kronweb.com.br/download2.php?id=353>
- < <http://pt.scribd.com/doc/123911111/Capitulo-9-Telemetria-Transmissao-Em-Instrumentacao> >
- < <http://brahms.di.uminho.pt/discip/lecom/li2-0405/material/PicBook-PT.pdf> >
- SCHNEIDER ELECTRIC. **Redes de Comunicação Industrial**, Documento Técnico nº2, Setembro 2007. Disponível em: <http://www.schneiderelectric.pt/documents/product-services/training/doctecnico_redes.pdf>
- WEG. **Manual da comunicação Modbus-RTU**. 2007. Disponível em: <<http://www.gigawattsistemas.com.br/file/1-15618inteligente.pdf>>
- SCADABR **Manual do Software**. 2010. Disponível em: <<http://www.scadabr.com.br/?q=downloads>>

- ELIPSE SCADA **LIMITAÇÕES DO MODO DEMO**. Disponível em:
<<http://kb.elipse.com.br/enus/questions/958/KB30097%3A+Limita%C3%A7%C3%B5es+do+modo+demo+no+Elipse+SCADA.>>

- NISHI, Y. W. **Controle a Distância de um Barco com Arduino+Bluetooth**, 2012. Disponível em: <<http://conferencias.utfpr.edu.br/ocs/index.php/sicite/2012/paper/viewFile/988/437>>

- ZOMETA J. P. **CÓDIGO-FONTE MODBUS RTU**. Disponível em:
<<https://sites.google.com/site/jpmzometa/arduino-mbrt/examples>>

- **Manual do CRF**. Disponível nos computadores do Laboratório de Sistema de Controle do CEFET-MG - CAMPUS ARAXÁ.

- **Manual do CRL**. Disponível nos computadores do Laboratório de Sistema de Controle do CEFET-MG - CAMPUS ARAXÁ.

ANEXOS

//Código-Fonte do da Planta de Controle de Vazão

```

#include <ModbusSlave.h>
#include <PID_v1.h>

// double constante=6; /* Constante que converge o sinal do potenciômetro
// para um sinal de 0-255 níveis para a entrada Input*/

// Declarações das variáveis do PID
double Setpoint, Input, Output;
//Parâmetros PID (kp , ki, kd)
PID myPID(&Input, &Output, &Setpoint, 2.0, 0.25, 0.025, DIRECT);

/*
 * History:
 * I had modified original library Arduino Modbus Slave:
 * http://sites.google.com/site/jpmzometa/arduino-mbrt/arduino-modbus-slave
 * adding:
 * - Function 0x01 - Read Coils
 * - Function 0x02 - Read Input Status
 * - Function 0x04 - Read Input Registers
 * - Function 0x05 - Write Single Coil
 * - New methods for setup arrays of holding registers, input registers, coils and discrete inputs.
 * - Arduino 0.22 and 1.0 support
 */

/* First step MBS: create an instance */

ModbusSlave mbs;

/*REGISTRADORES DAS VARIÁVEIS DO ARDUINO */

enum {
    MB_40000, //A0
    MB_40001, //A1
    MB_40002, //A2
    MB_40003, //A3
    MB_40004, //A4
    MB_40005, //A5
    MB_40006, //PWM PIN_3
    MB_40007, //PWM PIN_5
    MB_40008, //PWM PIN_6
    MB_40009, //PWM PIN_9 OUT_PID
    MB_40010, //PWM PIN_10

```

```

MB_40011, //PWM PIN_11
MB_40012, //Pino 2 do arduino 1= liga Led 0 desliga led.
MB_40013, //Pino 4 do arduino 1= liga Led 0 desliga led.
MB_40014, //Pino 7 do arduino 1= liga Led 0 desliga led.
MB_40015, //Pino 8 do arduino 1= liga Led 0 desliga led.
MB_40016, //Pino 12 do arduino 1= liga Led 0 desliga led.
MB_40017, //Pino 13 do arduino 1= liga Led 0 desliga led.

//MEMORIAS_PID
MB_40018, //SP
MB_40019, //IN
MB_40020,

MB_REGS // total number of holding registers on slave */
};

int regs[MB_REGS];

unsigned long wdog = 0; // watchdog */
unsigned long tprev = 0; // previous time*/

void setup()
{
/* the Modbus slave configuration parameters */

const unsigned char SLAVE = 1; // slaveId */
const long BAUD = 115200; // baud rate */
const char PARITY = 'n'; // n=none; e=even; o=odd */
const char TXENPIN = 6; // output driver enable pin */

myPID.SetMode(AUTOMATIC);

/* Second step MBS: configure */
mbs.configure(SLAVE, BAUD, PARITY, TXENPIN);
mbs.setup_regs(regs, MB_REGS);
}

```

```

void loop()
{
  /* Third and last step MBS: update in loop*/
  if(mbs.update())
    wdog = millis();
  if ((millis() - wdog) > 3000);

  // ***** associação dos registradores para entradas..... físicas *****

  regs[MB_40000] = analogRead(A0); //A0
  regs[MB_40001] = analogRead(A1); //A1
  regs[MB_40002] = analogRead(A2); //A2
  regs[MB_40003] = analogRead(A3); //A3
  regs[MB_40004] = analogRead(A4); //A4
  regs[MB_40005] = analogRead(A5); //A5

  analogWrite(3,regs[MB_40006]); //PWM PIN_3
  analogWrite(5,regs[MB_40007]); //PWM PIN_5
  analogWrite(6,regs[MB_40008]); //PWM PIN_6
  analogWrite(9,regs[MB_40009]); //PWM PIN_9
  analogWrite(10,regs[MB_40010]); //PWM PIN_10
  analogWrite(11,regs[MB_40011]); //PWM PIN_11

  digitalWrite(2, regs[MB_40012]); //Pino 2 do arduino
  digitalWrite(4, regs[MB_40013]); //Pino 4 do arduino
  digitalWrite(7, regs[MB_40014]); //Pino 7 do arduino
  digitalWrite(8, regs[MB_40015]); //Pino 8 do arduino
  digitalWrite(12, regs[MB_40016]); //Pino 12 do arduino
  digitalWrite(13, regs[MB_40017]); //Pino 13 do arduino

  Input= map(analogRead(0),0,1023,0,100); //CONVERTE O VALOR DO SINAL DE
  ENTRADA ANALOGICA

  // if (Input>=999){ //LIMITA A ENTRADA EM 999
  // Input=999;
  // }
  Setpoint = regs[MB_40018] ; //REGISTRADOR DE SETAMENTO DE SETPOINT.
  // if (Setpoint>=999){
  // Setpoint=999;
  // }

```

```

myPID.Compute(); // Calcula Saída do PID

regs[MB_40009]=Output; //REGISTRADOR DE LEITURA DA SAIDA.
regs[MB_40019]=Input; //REGISTRADOR DE LEITURA DA ENTRADA.

}

```

```

//Código-Fonte do da Planta de Controle de Nível

#include <ModbusSlave.h>
#include <PID_v1.h>

// double constante=6; /* Constante que converge o sinal do potenciômetro
// para um sinal de 0-255 níveis para a entrada Input*/

// Declarações das variáveis do PID
double Setpoint, Input, Output;
//Parâmetros PID (kp , ki, kd)
PID myPID(&Input, &Output, &Setpoint, 2.5, 0.5, 0.025, DIRECT);

/*
* History:
* I had modified original library Arduino Modbus Slave:
* http://sites.google.com/site/jpmzometa/arduino-mbrt/arduino-modbus-slave
* adding:
* - Function 0x01 - Read Coils
* - Function 0x02 - Read Input Status
* - Function 0x04 - Read Input Registers
* - Function 0x05 - Write Single Coil
* - New methods for setup arrays of holding registers, input registers, coils and discrete inputs.
* - Arduino 0.22 and 1.0 support
*/

/* First step MBS: create an instance */

ModbusSlave mbs;

/*REGISTRADORES DAS VARIÁVEIS DO ARDUINO */

enum {
    MB_40000, //A0
    MB_40001, //A1

```

```

MB_40002, //A2
MB_40003, //A3
MB_40004, //A4
MB_40005, //A5
MB_40006, //PWM PIN_3
MB_40007, //PWM PIN_5
MB_40008, //PWM PIN_6
MB_40009, //PWM PIN_9 OUT_PID
MB_40010, //PWM PIN_10
MB_40011, //PWM PIN_11
MB_40012, //Pino 2 do arduino 1= liga Led 0 desliga led.
MB_40013, //Pino 4 do arduino 1= liga Led 0 desliga led.
MB_40014, //Pino 7 do arduino 1= liga Led 0 desliga led.
MB_40015, //Pino 8 do arduino 1= liga Led 0 desliga led.
MB_40016, //Pino 12 do arduino 1= liga Led 0 desliga led.
MB_40017, //Pino 13 do arduino 1= liga Led 0 desliga led.

//MEMORIAS_PID
MB_40018, //SP
MB_40019, //IN
MB_40020,

MB_REGS          /* total number of holding registers on slave */
};

int regs[MB_REGS];

unsigned long wdog = 0;    /* watchdog */
unsigned long tprev = 0;  /* previous time*/

void setup()
{
/* the Modbus slave configuration parameters */

const unsigned char SLAVE = 2;    /* slaveId */
const long BAUD = 115200;        /* baud rate */
const char PARITY = 'n';        /* n=none; e=even; o=odd */
const char TXENPIN = 6;        /* output driver enable pin */

myPID.SetMode(AUTOMATIC);

```

```

/* Second step MBS: configure */
mbs.configure(SLAVE, BAUD, PARITY, TXENPIN);
mbs.setup_regs(regs, MB_REGS);

}

void loop()
{
  /* Third and las step MBS: update in loop*/
  if(mbs.update())
    wdog = millis();
  if ((millis() - wdog) > 3000);

// ***** associação dos registradores para entradas..... fisicas *****

regs[MB_40000] = analogRead(A0); //A0
regs[MB_40001] = analogRead(A1); //A1
regs[MB_40002] = analogRead(A2); //A2
regs[MB_40003] = analogRead(A3); //A3
regs[MB_40004] = analogRead(A4); //A4
regs[MB_40005] = analogRead(A5); //A5

analogWrite(3,regs[MB_40006]); //PWM PIN_3
analogWrite(5,regs[MB_40007]); //PWM PIN_5
analogWrite(6,regs[MB_40008]); //PWM PIN_6
analogWrite(9,regs[MB_40009]); //PWM PIN_9
analogWrite(10,regs[MB_40010]); //PWM PIN_10
analogWrite(11,regs[MB_40011]); //PWM PIN_11

digitalWrite(2, regs[MB_40012]); //Pino 2 do arduino
digitalWrite(4, regs[MB_40013]); //Pino 4 do arduino
digitalWrite(7, regs[MB_40014]); //Pino 7 do arduino
digitalWrite(8, regs[MB_40015]); //Pino 8 do arduino
digitalWrite(12, regs[MB_40016]); //Pino 12 do arduino
digitalWrite(13, regs[MB_40017]); //Pino 13 do arduino

Input= map(analogRead(0),0,1023,0,100); //CONVERTE O VALOR DO SINAL DE
ENTRADA ANALOGICA

```

```
// if (Input>=999){ //LIMITA A ENTRADA EM 999
// Input=999;
// }
Setpoint = regs[MB_40018] ; //REGISTRADOR DE SETAMENTO DE SETPOINT.
// if (Setpoint>=999){
// Setpoint=999;
// }

myPID.Compute(); // Calcula Saída do PID

regs[MB_40009]=Output; //REGISTRADOR DE LEITURA DA SAIDA.
regs[MB_40019]=Input; //REGISTRADOR DE LEITURA DA ENTRADA.

}
```

|