



**FACULDADE DE TECNOLOGIA SENAI CIMATEC
ESPECIALIZAÇÃO EM AUTOMAÇÃO, CONTROLE E ROBOTICA**

DAILLON CARDOSO PEREIRA

APLICAÇÃO DE TECNOLOGIAS OPEN-SOURCE EM AUTOMAÇÃO INDUSTRIAL

Salvador/BA
2015

DAILLON CARDOSO PEREIRA

APLICAÇÃO DE TECNOLOGIAS OPEN-SOURCE EM AUTOMAÇÃO INDUSTRIAL

Monografia apresentada à Faculdade de Tecnologia SENAI CIMATE como requisito parcial para a obtenção do Título de Especialista em Automação, Controle e Robótica.

Professor Orientador: Oberdan Rocha Pinheiro, Msc.

Salvador/BA
2015

Ficha catalográfica elaborada pela Biblioteca da Faculdade de Tecnologia SENAI CIMATEC

P436a Pereira, Daillon Cardoso

Aplicação de tecnologia Open-Source em automação industrial / Daillon Cardoso Pereira – Salvador, 2015.

46 f. : il. color.

Orientador: MSc. Oberdan Rocha Pinheiro.

Monografia (Especialização em Automação, Controle e Robótica) – Programa de Pós-Graduação, Faculdade de Tecnologia SENAI CIMATEC, Salvador, 2015.

Inclui referências.

1. Automação industrial. 2. Open-Source. 3. Software de supervisão.
I. Faculdade de Tecnologia SENAI CIMATEC. II. Pinheiro, Oberdan Rocha. III. Título.

CDD 629.8

DAILLON CARDOSO PEREIRA

**APLICAÇÃO DE TECNOLOGIAS OPEN-SOURCE EM AUTOMAÇÃO
INDUSTRIAL**

Projeto Final de Curso aprovado com nota 8,0 (oito) como requisito de Especialista em Automação, Controle e Robótica, tendo sido julgado pela Banca Examinadora formada pelos Professores:

Msc. Oberdan Rocha Pinheiro – Orientador
Faculdade de Tecnologia SENAI CIMATEC

Msc. Milton Bastos de Souza – Professor
Faculdade de Tecnologia SENAI CIMATEC

Salvador, 11 de dezembro de 2015.

*Dedico esse trabalho aos meus pais, às minhas irmãs,
à minha namorada e aos amigos por tudo que significam para mim.*

“A vida é uma peça de teatro que não permite ensaios. Por isso, cante, chore, dance, ria e viva intensamente, antes que a cortina se feche e a peça termine sem aplausos.”

Charles Chaplin

AGRADECIMENTOS

Agradeço a Deus, nosso poderoso pai, que sem ele nada é possível. Aos meus pais que dedicaram e empenharam em formar seus filhos, e que me deram todo o apoio possível. Ao meu pai, Miguel Rosa Pereira, por ser aquele pai presente, honesto, perseverante que sempre me deu toda a atenção e amor possível, dando-me a possibilidade de estudar e construir um futuro melhor.

Agradeço à minha mãe, Adinita Maria Cardoso Pereira, que sempre me acolheu e esteve presente durante toda a minha vida e a minha jornada até aqui, sempre me dando todo o amor possível.

Ao professor Oberdan Rocha Pinheiro, por ter me orientado. Aos tantos outros professores que participaram de minha formação.

À minha namorada Kivian Antunes Silva Queiroz, que esteve comigo durante todo esse tempo me apoiando.

RESUMO

Diante um cenário de alta competitividade, as empresas têm buscado novas soluções tecnológicas para reduzir custos. O desenvolvimento desta aplicação visa a atender a algumas características importantes como interoperabilidade, flexibilidade, baixa manutenção, redução de custos, segurança, controle, distribuição de informações e disponibilidade. Nesse contexto, este trabalho apresenta um software de supervisão industrial, o qual roda em um sistema operacional Linux UBUNTU, e utiliza tecnologia open-source para manipulação de variáveis em um CLP. A comunicação entre a aplicação cliente e o servidor é através do protocolo de transporte TCP/IP. A aplicação conta com um diferencial na modelagem da arquitetura em relação aos padrões atualmente utilizados na indústria, ou seja, uma mudança de foco na programação distribuída e orientada a objetos, para a programação distribuída e orientada a serviços. A codificação do processo cliente foi desenvolvida com a linguagem Java e ferramentas livres, objetivando simplificar ao máximo a codificação e implantação.

Palavras-chave: Automação; Open-source; Software;

ABSTRACT

Facing a high competitiveness scenario, companies have sought new technologies solutions to lower costs. The development of this application aims to aid some important characteristics such as interoperability, safety, monitoring, information distribution and availability. In this context, this project presents a industrial supervision software, which runs in Linux Ubuntu operational system and uses open-source technology to manipulate variables on a PLC. The communication between the client application and the server is through a TCP/IP transport protocol. The application has a differential in the architecture modeling compared to the actual patterns used in the industry, that is to say, a shift of focus in the distributed programming and object-oriented, to a distributed programming and service-oriented. The Client Process codification was developed in a Java language and free tools, aiming to simplify at the most the coding and implantation.

Key words: Automation; Open-Source; Software;

SUMÁRIO

CAPÍTULO I – INTRODUÇÃO	12
1.1 OBJETIVO	13
1.2 ORGANIZAÇÃO DA MONOGRAFIA	13
CAPÍTULO II - REVISÃO BIBLIOGRÁFICA	14
2.1 AUTOMAÇÃO INDUSTRIAL	14
2.1.1 CONTROLADORES LÓGICOS PROGRAMÁVEIS (CLP)	14
2.2 REDES INDUSTRIAIS	15
2.3 PADRÕES DE REDES INDUSTRIAIS	16
2.3.1 O MODELO DE REFERÊNCIA OSI	16
2.3.2 O MODELO DE REFERÊNCIA TCP/IP	17
2.3.3 ETHERNET	18
2.3.4 AS-I	19
2.3.4.1 CARACTERÍSTICAS	20
2.3.4.2 MODELO DE COMUNICAÇÃO E CABEAMENTO	20
2.3.4.3 PROTOCOLO DE COMUNICAÇÃO	21
2.3.5 PROFIBUS	23
2.3.5.1 CARACTERÍSTICAS E ARQUITETURA DA REDE PROFIBUS	24
2.3.6 DEVICENET	30
2.3.6.1 PROTOCOLO DEVICENET	31
2.3.6.2 A CAMADA LINK DE DADOS	31
2.3.6.3 MODOS DE COMUNICAÇÃO	32
2.3.6.4 MEIO FÍSICO E MEIO DE TRANSMISSÃO	32
2.3.6.5 TOPOLOGIA DE REDE	33
2.3.4 TECNOLOGIA OPEN-SOURCE	34
2.3.4.1 SISTEMA OPERACIONAL LINUX	35
CAPÍTULO III – PROJETO E IMPLEMENTAÇÃO	36
3.1 SOFTWARE DE DESENVOLVIMENTO: NETBEANS 8.0 E JAVA 7.0	36
3.2 SISTEMA OPERACIONAL – LINUX UBUNTU 14.04	36
3.3 SIEMENS SIMATIC S7-300	37
3.4 APLICAÇÃO	38
3.4.1 ARQUITETURA DO SISTEMA	38

3.4.2 CODIFICAÇÃO DO SISTEMA	39
CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES	42
REFERÊNCIAS	44

CAPÍTULO I – INTRODUÇÃO

Atualmente existe entre fornecedores e usuários de sistemas e equipamentos industriais um novo entendimento sobre a busca por arquiteturas autônomas que independam de fabricantes, que apresentem alto desempenho, segurança, robustez e modernidade. É preciso que estes produtos sejam desenvolvidos e possam atender às seguintes características: apresentar uma maior interoperabilidade entre plataformas de diferentes fabricantes; maior flexibilidade pra futuras manutenções e atualizações; atender às atuais exigências relacionadas a armazenamento, segurança, disponibilidade, controle e distribuição de informações.

A utilização de tecnologia open-source vem ao encontro dessas possibilidades, ou seja, um passo adiante para a disseminação de sistemas abertos em empresas de automação industrial. No ambiente industrial, os benefícios dos softwares open-source incluem, principalmente, a redução dos custos, visto que reduz-se os gastos com a aquisição de licenças de software. Além disso, como o código fonte é aberto, ele é testado por um tempo muito maior resultando em programas mais seguros e confiáveis.

Os desenvolvedores de softwares já estão se adaptando a essa nova tendência do mercado, embora ainda existam muitas barreiras para que esses tipos de softwares entrem no mercado de automação industrial. A redução de custos é hoje o maior desafio enfrentado pelos grandes empresas. É importante que as empresas procurem alternativas confiáveis, eficientes e mais baratas fazendo com que os custos reduzam, impactando diretamente no preço final do produto, ou seja, possibilitando manter uma maior competitividade.

É nesse contexto que se situa esse trabalho. Apresentando um sistema de supervisão industrial utilizando tecnologia open-source com baixo custo e alta confiabilidade.

1.1 OBJETIVO

Os principais objetivos dessa pesquisa são:

- Desenvolver um software utilizando tecnologia open-source para supervisão de um processo industrial.
- Estudar as diversas tecnologias de redes industriais existentes no mercado.
- Testar o software desenvolvido em uma aplicação industrial

1.2. ORGANIZAÇÃO DA MONOGRAFIA

A monografia é dividida em 4 capítulos. No capítulo 1 é apresentado a contextualização do trabalho, juntamente com a justificativa e problema, além dos objetivos gerais e específicos.

Já no Capítulo 2, é feita uma revisão bibliográfica sobre automação industrial, redes de automação utilizados na indústria e que servirão de aporte teórico para o desenvolvimento do projeto de pesquisa.

No capítulo 3, é apresentado o problema e a implementação, conterà também a descrição dos equipamentos utilizados e os testes executados na implementação.

Por fim, no capítulo 4, são apresentados os dados e as conclusões do trabalho, além de futuros trabalhos que poderão ser desenvolvidos a partir desse.

CAPÍTULO II – REVISÃO BIBLIOGRÁFICA

Esse capítulo da monografia é utilizado para uma revisar bibliograficamente sobre automação industrial, tecnologia open-source, redes industriais, de modo que sirva para aprofundar os conhecimentos, além de facilitar o entendimento dos termos e uma boa compreensão da monografia em geral.

2.1 AUTOMAÇÃO INDUSTRIAL

“A automação pode ser definida como a tecnologia por meio da qual um processo ou procedimento é alcançado sem a utilização humana. É realizado utilizando-se um programa de instruções combinado a um sistema de controle que executa as instruções.” (GROOVER, 2011, p.45)

Os sistemas automatizados devem ser capazes de executar tarefas por meio de decisões próprias, baseados em sinais dos sensores e equipamentos do processo a ser controlado e agir com a menor intervenção humana possível, ou seja, substituindo o trabalho do homem por uma máquina.

No início dos anos 1960, quando a automação industrial começou a ter resultados existia uma automação do tipo rígido, ou seja, a máquina era capaz de executar as tarefas na confecção de um tipo de peça utilizando a mínima atuação humana, porém se houvesse uma modificação no produto, a modificação da automação da máquina era muito difícil e com custos elevados, às vezes até impossível.

“Com o advento do circuito integrado (1960) e do microprocessador (1970), a quantidade de inteligência que pôde ser embutida em uma máquina a um custo razoável se tornou enorme” (RIBEIRO, 1999, p.45). Com esses adventos hoje é possível a integração de várias tecnologias, como a hidráulica e a pneumática, informática, o nome dado a essa integração é chamado FMS (sistema flexível de manufatura).

A automação industrial traz diversos benefícios, pois torna automáticas as tarefas que antes eram realizadas pelo homem, evitando o esforço muscular e mental, em troca de equipamentos eletromecânicos fazendo com haja um ganho em eficiência, segurança, redução de custos, além do aumento da produtividade.

2.1.1. CONTROLADORES LÓGICOS PROGRAMÁVEIS (CLP)

“Um controlador lógico programável (CLP) pode ser definido como baseado em um microcomputador que usa instruções armazenadas em uma memória programável para implementar lógica, sequenciamento, temporização contagem e funções aritméticas por meio de módulos de entrada/saída (E/S) digitais ou analógicas para controle de máquinas e processos.” (GROOVER, 2011, p.45)

Um CLP pode ser definido como um complexo sistema eletrônico para uso industrial capaz de gerir qualquer operação de controle industrial de maneira flexível. O CLP na verdade é um computador que realiza funções de controle em automação, é um equipamento eletrônico dotado de uma memória interna programável e outra não

programável, que contém as informações do programa a ser executado, interagindo com os dispositivos conectados nas entradas e saídas do CLP.

O CLP funciona através da análise dos sinais que chega à entrada (sinais analógicos ou digitais) a partir de elementos sensores e transdutores, através de uma lógica programada é emitido um sinal das bobinas de saída, que podem ser analógicos ou digitais, e assim permite o acionamento dos atuadores.

2.2 REDES INDUSTRIAIS

Para que os equipamentos de automação industrial pudessem se comunicar foi preciso que redes específicas fossem desenvolvidas. A comunicação rápida e confiável entre os equipamentos são fatores considerados bastante importantes e são indispensáveis na produtividade industrial.

A capacidade de comunicação entre dispositivos e o uso de mecanismos padronizados, abertos e transparentes são componentes indispensáveis do conceito de automação de hoje. A comunicação vem se expandindo rapidamente no sentido horizontal nos níveis inferiores (field level), assim como no nível vertical integrando todos os níveis hierárquicos. (PROFIBUS, 2000)

As características da aplicação e o custo são quem define os diferentes sistemas de comunicação existentes em redes industriais. Segundo (PROFIBUS, 2000) no nível de atuadores/sensores a rede AS-Interface é ideal, pois os sinais binários são transmitidos via um barramento extremamente simples e de baixo custo, juntamente com a energia possibilitando alimentar esses equipamentos.

Já para o nível de campo, segundo (PROFIBUS, 2000), a comunicação utilizando PROFIBUS-DP ou PROFIBUS-PA é bastante eficiente, transmitindo dados do processo de forma cíclica, enquanto alarmes, parâmetros e diagnósticos são transmitidos aciclicamente, ou seja, somente quando for necessário. Faz desse rede um eficiente sistema de comunicação em tempo real.

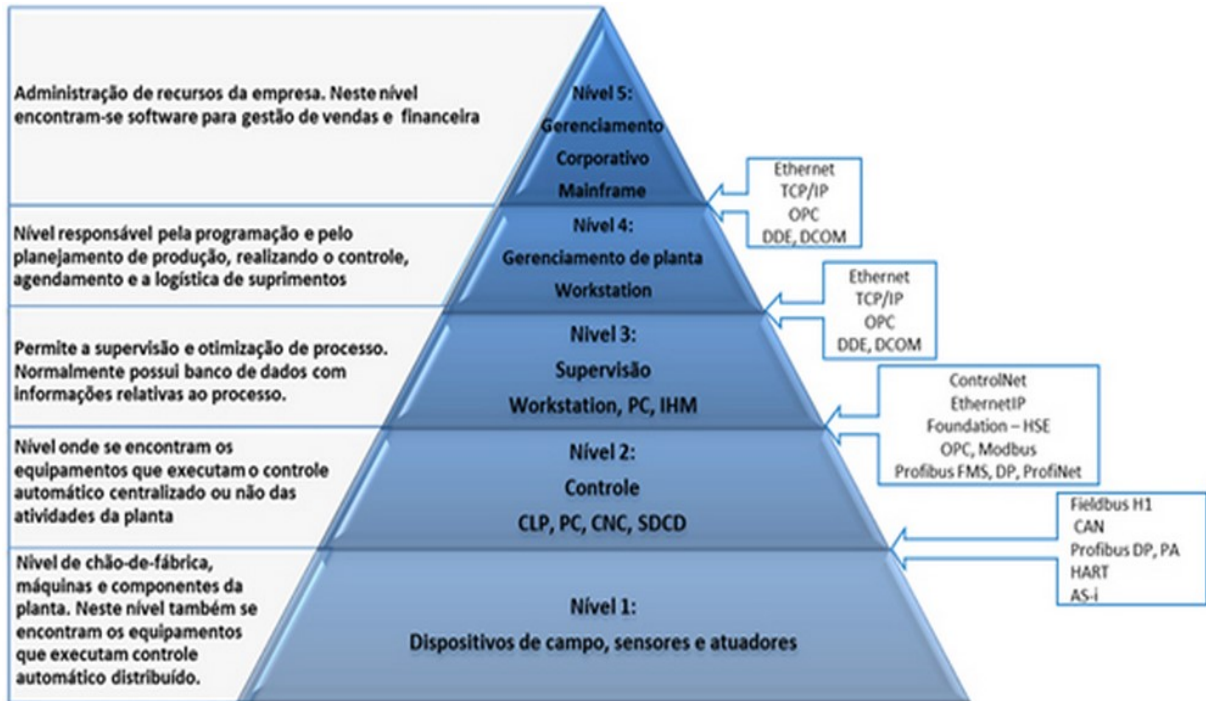


Figura 1: Pirâmide de automação/integração. (SMAR, 2012)

Observando a Figura 1, podemos associar cada nível da pirâmide a um específico protocolo de comunicação, como comentado acima, no nível 1 de atuadores e sensores temos a rede AS-I que apresenta características e funções ideais para tal nível.

A revolução da comunicação industrial na tecnologia da automação está revelando um enorme potencial na otimização de sistemas de processo e tem feito uma importante contribuição na direção da melhoria no uso de recursos. (PROFIBUS, 2000)

Os padrões na automação industrial foram criados para possibilitar que diferentes tipos de equipamentos de automação pudessem se comunicar de modo padronizado, pois os números de dispositivos tornaram-se bastante numerosos.

Os problemas com a integração ocorrem principalmente em relação à interligação de equipamentos de diferentes fabricantes. O uso de padrões internacionais de comunicação torna-se a solução mais adequada a ser adotada pelos fabricantes quanto à integração computacional e de equipamentos de tecnologias diferentes. (VILLAMAYOR, 1999, p.78)

2.3 PADRÕES DE REDES INDUSTRIAIS

2.3.1 O MODELO DE REFERÊNCIA OSI

O modelo OSI foi desenvolvido pela ISO (International Standards Organization) como iniciativa para a padronização internacional utilizada nas diversas camadas. Esse modelo trata da interconexão entre sistemas abertos, ou seja, para sistemas comunicarem com outros sistemas abertos. O modelo OSI, em si, não é

propriamente uma arquitetura de redes, pois não é especificado nele os serviços e protocolos pertencentes e utilizados em cada camada. Na verdade, ele simplesmente informa o que cada camada deve fazer.

Camada física – essa camada é responsável pela transmissão de bits brutos em cada canal de comunicação. Além disso ela deve garantir que as informações trafegadas serão recebidas pelo receptor de maneira exata, ou seja, conforme o enviado.

Camada de enlace de dados – essa camada transforma um canal de transmissão bruta em uma linha de transmissão livre de erros não detectados para a camada de rede. Essa camada faz com que o transmissor divida os dados em quadro de dados e transmita esses dados de forma sequencial. O receptor receberá a informação, caso confiável, de cada quadro e em seguida envia de volta um quadro confirmando o recebimento. Essa camada também impede que um transmissor mais rápido envie uma grande quantidade de dados maior do que o buffer do receptor pode suportar.

Camada de redes – essa camada controla a operação de sub-rede, ou seja, é ela quem determina o roteamento dos pacotes da origem até o destino final. Também é função dessa camada fazer o controle de congestionamento evitando que quando houver muitos pacotes na sub-rede eles dividam o mesmo caminho.

Camada de transporte – essa camada aceita os dados provenientes da camada superior a essa, se necessário divide em unidades menores, repassa os dados a camada de rede assegurando que todas as unidades chegarão corretamente ao seu destino. É essa camada também que permite que mudanças no hardware não interfira no funcionamento mantendo sempre a mesma eficiência.

Camada de sessão – essa camada é responsável por estabelecer sessões entre usuários de diferentes máquinas. Os serviços que pode ser oferecidos em cada sessão são: controle de diálogo, gerenciamento de símbolos e a sincronização.

Camada de apresentação – essa camada é responsável pela sintaxe e a semântica dos dados transmitidos. Para que haja comunicação entre equipamentos com diferentes representações de dados, as estruturas são definidas pela camada de maneira abstrata com um codificação padrão que será utilizada na conexão.

Camada de aplicação – essa camada é responsável por alguns dos protocolos necessários para os usuários como o HTTP (HyperText Transfer Protocol) que é a base para o Word Wide Web.

2.3.2 MODELO DE REFERÊNCIA TCP/IP

Segundo (TANENBAUM, 2003, p.18), o Departamento de Defesa dos Estados Unidos (DoD) patrocinava o desenvolvimento da ARPANET que era uma rede de computadores geograficamente distribuída. Centenas de universidades e repartições públicas eram conectadas pela ARPANET por meio de linhas telefônicas dedicadas.

A partir do momento em que as redes de rádio e satélites foram sendo desenvolvidas começou a surgir problemas com os protocolos existentes.

Partindo desses problemas foi desenvolvido o modelo de referência TCP/IP que desde o início foi desenvolvido para conectar várias redes de maneira uniforme. Nesse modelo as camadas foram divididas em inter-redes, transporte, aplicação e host/rede

Camada inter-redes – essa camada tem a função de integrar toda a arquitetura do modelo. Ela deve fazer com que os hosts injetem pacotes em qualquer rede garantindo que eles chegaram ao destino de forma independente. Um formato de pacote e um protocolo IP (internet protocol) é definido. Esses pacotes são entregues onde eles forem necessários. Além disso é feito o controle de congestionamento e roteamento. Sendo assim, essa camada inter-redes funciona de forma semelhante a camada de rede do modelo OSI.

Camada de transporte – essa camada fica acima da camada inter-redes. Segundo [6] essa camada permite que entidades pares dos hosts de origem e de destino estabeleçam uma conexão e mantenha uma conversação. Existe 2 protocolos que são definidos o TCP (Transmission Control Protocol) e o UDP (User datagram Protocol). O primeiro é um protocolo confiável e orientado a conexão, entregando sem erros um fluxo de dados em qualquer computador ligado a camada inter-redes. Além disso é função desse protocolo fazer o controle de fluxo, ou seja, impedir que um envio rápido de mensagens por parte do transmissor sobrecarregue o receptor. Já o Segundo, UDP, é um protocolo mais simples e não confiável, ou seja, não há garantia que os pacotes serão entregues ao destino. É um serviço sem conexão, pois ele não mantém um relacionamento longo entre cliente e servidor.

Camada de aplicação – no protocolo TCP/IP não implementa as camadas de sessão e apresentação. A camada acima da transporte é a de aplicação, nessa camada estão todos os protocolos de nível mais alto como TELNET, terminal virtual, FTP, transferência de arquivos, SMTP, protocolo de correios eletrônico. O protocolo TELNET permite que usuários de outros computadores faça acesso remoto, já o FTP possibilita que a transferência de dados entre computadores. Muitos outros protocolos foram adicionado posteriormente como o DNS (Domain Name Service), que tem a função de mapear os nomes dos hosts para seu endereço de rede, além do HTTP, que permite que páginas na World Wide Web sejam encontradas.

Camada host/rede – o modelo de TCP/IP não especifica as camadas abaixo da camada inter-redes, somente o fato de os hosts utilizarem algum protocolo que possibilite o envio de pacotes IP. Esse protocolo por não ser definido varia de host para host e de rede para rede.

2.3.3 ETHERNET

O protocolo Ethernet teve início nos laboratórios da Xerox em 1972. De início existia uma rede na qual todas as estações compartilhavam do mesmo meio de transmissão, um cabo coaxial, utilizando barramento e trafegando dados a uma taxa

de 2,94 Mbps. Dado a dificuldade de comercialização de equipamentos devido a falta de padronização, em 1980 a IEEE (Institute of Electrical and Electronic Engineers) ficou responsável por resolver o problema da padronização.

Basicamente três elementos compõem o padrão, ou seja, o meio físico, o controle de acesso ao meio e o quadro ethernet. É função também definir como os dados serão transmitidos pela rede. Além disso é função do padrão Ethernet agrupar os dados recebidos pelos protocolos de níveis mais altos e inseri-los dentro dos frames que serão transmitidos.

As três camadas da arquitetura Ethernet possuem as seguintes funções:

Controle do Link Lógico (LLC, IEEE 802.2): inclui informações do protocolo de alto nível que entregou o pacote a ser transmitido. Com isso, a máquina receptora tem como saber para qual protocolo de alto nível ela deve entregar os dados de um quadro que ela acabou de receber.

Controle de acesso ao meio (MAC, IEEE 802.3): Monta o quadro de dados a ser transmitido pela camada física, incluindo cabeçalhos próprios dessa camada aos dados recebidos da camada de Controle do Link Lógico.

Física: Transmite os quadros entregues pela camada de Controle de Acesso ao Meio usando o método CSMA/CD. Define como os dados são transmitidos através do cabeamento da rede e também o formato dos conectores usados na placa de rede. (PINHEIRO, 2015, p.2)

2.3.4 AS-I

Em meados dos anos 1990 foi formado um consórcio de onze empresas europeias ligadas a setores de automação com o objetivo de criar uma rede industrial para sensores e atuadores discretos. A padronização ocorreu de fato em 1998, denominado AS-interface ou simplesmente AS-I (Actuator Sensor Interface). O objetivo principal do projeto AS-I era que houvesse uma simplicidade no nível de sensores e atuadores e os mesmos benefícios dos níveis superiores da hierarquia de automação industrial. Então, a rede foi desenvolvida para que as conexões dos sensores e atuadores fossem mais rápidas e simples aos seus controladores e que fossem um complemento aos demais sistemas.

O protocolo AS-I propicia um mecanismo de conexão eletromecânica que foi projetado para operar através de dois cabos, trafegando simultaneamente dados e energia para alimentação dos dispositivos, gerando como resultado um baixo custo. A conexão entre os dispositivos na rede AS-I é compartilhada entre diversos dispositivos, enquanto que no cabeamento tradicional é necessário um cabo para cada dispositivo o que eleva o custo da instalação.

O desenvolvimento da rede partiu de uma série de requisitos para a elaboração de uma interface serial digital, tais requisitos foram:

- Interoperabilidade: sensores e atuadores de diferentes empresas poderão ser conectados a uma mesma interface serial digital.

- Facilidade de instalação e localização de falhas · Baixo custo de conexão por nó e sem restrições de topologia de rede
- Cabeamento realizado através de dois fios, de fácil uso e baixo custo, capazes de entregar tanto energia quanto dados.
- Confiabilidade operacional alta em ambientes industriais severos. · Baixo tempo de reação e comportamento determinístico.
- Volume pequeno dos dispositivos usados na conexão de circuitos. (ALMEIDA, 2015)

2.3.4.1 CARACTERÍSTICAS

O funcionamento da rede AS-I é do tipo mestre-escravo efetuando polling cíclico, ou seja, o mestre tem a função de controlar toda a rede realizando Polling cíclico nos dispositivos escravos da rede. O mestre AS-i realiza várias tarefas, como inicialização da rede, identificação dos escravos, diagnóstico dos escravos e de dados transferidos. Além disso, geralmente se comunica a um controlador (PLC ou PC) para receber a configuração de controle da rede AS-i, reportar erros, endereçar escravos substituídos, entre outras tarefas. O tempo máximo de ciclo para uma rede AS-i é de 5ms, mesmo estando a rede repleta, com 31 dispositivos.[citação profibus org]. Essas características permitem que sejam utilizadas botoeiras e acionadores, intertravamentos de maneira mais eficientes.

O mestre faz a requisição aos escravos que são dispositivos passivos da rede, podendo ser conectado até 8 dispositivos a ele, 4 sensores e 4 atuadores. Além disso pode-se encontrar escravos que trabalham com valores analógicos, mas para isso é necessário quatro ciclos de leitura/escrita.

Topologia: Estrutura em árvore.

Meio: Cabo de 2 condutores para dados e alimentação (24VDC nominal, 2 A tipicamente)

Comprimento do bus: 100m por cada mestre AS-Interface (300m com repetidores)

Numero de escravos: máximo de 31

Número de E/S: Até 4 sensores e 4 actuadores por escravo (máximo 248 E/S digitais)

Endereçamento: Cada escravo tem um endereço, definido pelo mestre ou equip. específico

Mensagens: Do mestre para o escravo com resposta imediata.

Formato mensagem: 4 bits por escravo e mensagem

Tempo de ciclo: Com 31 escravos: 5 ms. Com menos escravos o tempo diminui.

Deteção de erros: Mensagens incorrectas detectadas, gerando repetição da mensagem.

Módulo Interface: 4 portas configuráveis (entradas, saídas ou bidireccional) e 4 parâmetros

Figura 2: arquitetura AS-I. (ELETRIC, 2007, p.7)

2.3.4.2 MODELO DE COMUNICAÇÃO E CABEAMENTO

O cabeamento utilizado na rede AS-I é chamado de “Yellow Flat Cable”, esse cabo é composto por dois fios e transportam simultaneamente dados e alimentação para os equipamentos da rede e também tem como características ser um cabo sem blindagem e perfilado, que evita a inversão de polaridade.

A rede utiliza o princípio do cabo comum podendo ligar vários equipamentos. Cada módulo escravo contém um “chip” que permitem que sensores e atuadores se conectem a interface AS-I. A informação enviada pelo Master contém 4 bits em série e são transferidos para cada escravo em cada ciclo. Após o reconhecimento dos bits pelos escravos, outros 4 bits são enviados para o mestre a partir do escravo com entrada e saída.

Segundo (ALMEIDA, 2015) O do cabo sistema AS-I utiliza uma tecnologia de conexão rápida através de conectores que perfuram o isolante e estabelece o contato com os fios internos, esses conectores são do tipo vampiro.

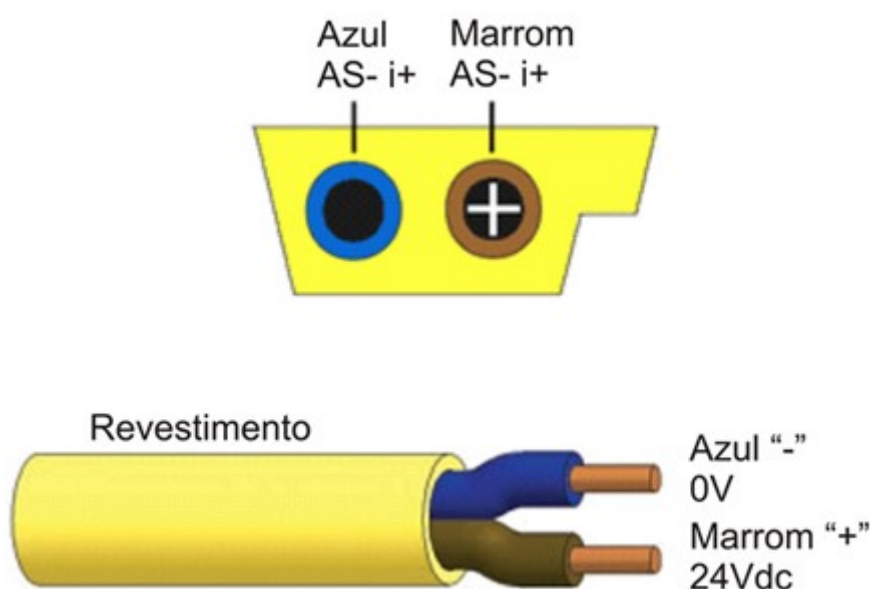


Figura 3: Padrão de cabeamento AS-i Fonte: (SMAR, 2015)

É possível observar na figura 3, o padrão do cabeamento AS-I que apresenta colocação externa na cor amarela, dois cabos especiais não blindados e perfilado, além de ser um cabo auto regenerativo, ou seja, o cabo retorna ao seu aspecto original caso seja feita alguma modificação.

2.3.4.3 PROTOCOLO DE COMUNICAÇÃO

O protocolo de rede AS-I utiliza um mestre por rede para fazer o controle da troca de dados que se inicia quando o mestre solicita sequencialmente de cada escravo, seguido de uma pausa e aguarda a resposta de cada escravo também seguida de outra pausa. Essa pausa por parte do mestre tem que ser mantida e checada pelos escravos após receber uma solicitação do mestre. Essa transmissão utiliza um formato fixo, ou seja, esse tipo de formato simplifica o processo de controle de transmissão permitindo assim que o mestre interogue todos os escravos e atualize suas entradas e saídas em um tempo inferior a 5ms.

É função de o mestre reconhecer os erros na transmissão, falha nos módulos escravos, além de verificar as tensões na rede e reportar ao controlador (CLP) caso ocorra alguma falha. A comunicação é mantida mesmo que durante a operação normal seja adicionado ou trocado algum escravo da rede.

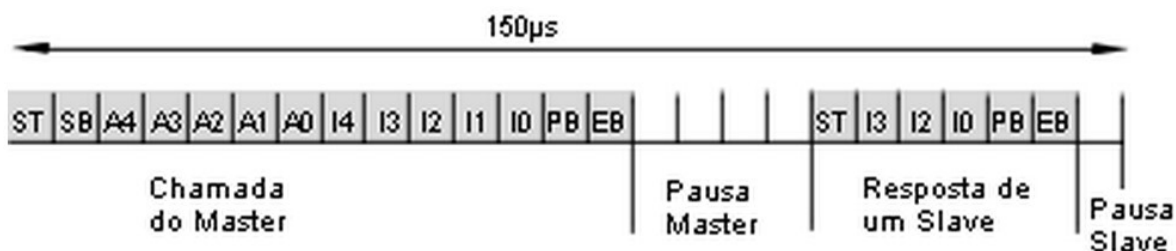


Figura 4: Formato e tipo de mensagem. (ALMEIDA, 2007)

Na figura acima é possível ver a estrutura dos frames enviados pelo mestre e pelo escravo que tem um tamanho diferente, já que o frame do mestre é composto por 14 bits e o frame do escravo contém apenas 7 bits.

Existem nove diferentes tipos de mensagens AS-I: duas para dados e parâmetros, duas para ajustes ou mudança de endereços de escravos e cinco para identificação de escravos ou para investigação de status. (ALMEIDA, 2007)

Segundo (PROFIBUS, 2009) As mensagens são:

Troca de dados (Data Exchange): Tem a função de transferir e solicitar dados de entrada de um escravo;

Escrita de parâmetro (Write parameter): Essa mensagem configura os parâmetros internos do escravo através da escrita.

Atribuição de endereço (Assign address): Essa mensagem configura um novo endereço para o módulo escravo.

Inicialização do escravo (Reset): Essa mensagem coloca o escravo em seu estado inicial.

Apagamento do endereço operacional (Delete operating address): Essa mensagem apaga o endereço de um escravo e é utilizado com o comando Assign address que configura um novo endereço.

Leitura de configuração de I/O (Read I/O configuration): Essa mensagem permite ao mestre fazer a leitura das configurações de entrada e saída de um dispositivo escravo.

Leitura de código de identificação (Read ID code): Essa mensagem serve para verificar o código do dispositivo que é único e especificado pelo fabricante, não podendo ser alterado.

Leitura de Status (Read Status): Essa mensagem faz a leitura do buffer de status do módulo escravo que contém dois flags, uma para informar que o dispositivo escravo está realizando uma rotina interna de armazenamento permanente do endereço, e o outro, para informar erro de leitura da memória não volátil que é gerado durante o processo de reset.

Leitura e apagamento de status (Read and Reset Status): Essa mensagem do mestre realiza a leitura e reset do buffer de status de um escravo.

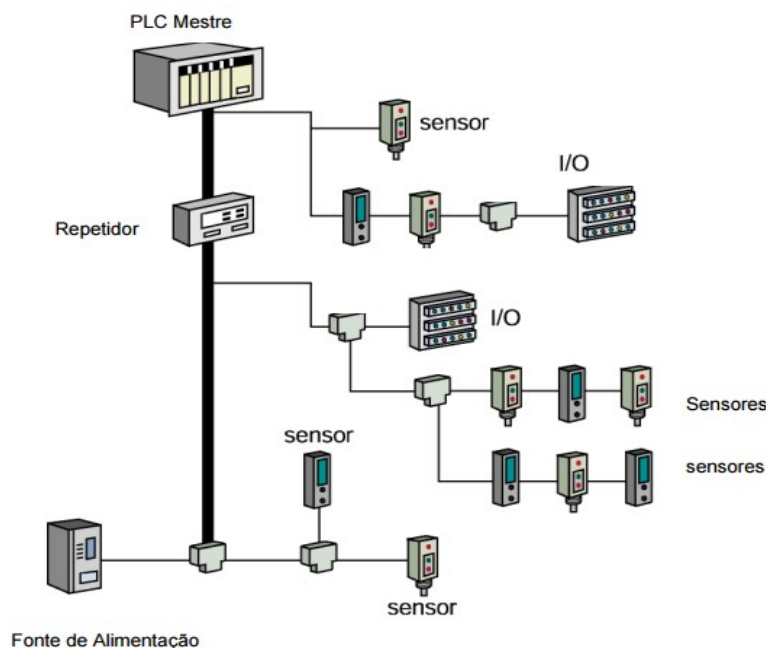


Figura 5: Exemplo de uma topologia de uma rede AS-Interface [9]

2.3.5 PROFIBUS

A história do PROFIBUS deu início devido a necessidade de padronização que não era aplicada as redes industriais, ou seja, até a época não era possível a compatibilidade entre equipamento de fabricantes diferentes. Foi ai que mais de 15 empresas e algumas instituições com apoio da Alemanha uniram forças para a criação de um padrão de comunicação aberto para sistemas de automação industrial.

Segundo (VILLAMAYOR, 1999, p. 45) A rede PROFIBUS apresenta algumas características próprias de redes chão de fábrica:

- Permite ser utilizada em grandes distancias;
- Apresenta mensagens curtas;
- Tempo Real / Determinístico
- Grande imunidade a ruídos eletromagnéticos;
- Permite a entrada e saídas de estações online;

O protocolo PROFIBUS é composto por 3 modos diferentes:

PROFIBUS DP: Foi desenvolvido para operar com uma alta velocidade e conexão de baixo custo, e é utilizado na comunicação entre sistemas de controle de automação e seus respectivos I/O's

distribuídos no nível de dispositivo. Pode ser usado para substituir a transmissão de sinal de 24V em sistemas de automação de manufatura assim como para a transmissão de sinais de 4 a 20 mA ou HART em sistemas de automação de processo (PROFIBUS, 1999)

PROFIBUS FMS: é o perfil de comunicação universal para tarefas de comunicação complexas, por exemplo, comunicação entre mestres. O FMS oferece muitas funções sofisticadas de comunicação entre dispositivos inteligentes, porém já foi descontinuado e atualmente vêm gradativamente sendo substituído por redes Profinet. (PROFIBUS 1999)

PROFIBUS PA: Essa é uma solução da PROFIBUS voltada para a automação de processos, além disso, essa solução permite que seja utilizado em ambientes onde existem riscos de explosão, pois ela tem a transmissão intrinsecamente segura. Também é utilizada em células de manufatura.

2.3.5.1 CARACTERÍSTICAS E ARQUITETURA DA REDE PROFIBUS

O Profibus especifica as características técnicas e funcionais de um sistema de comunicação industrial, através do qual dispositivos digitais podem se interconectar, desde do nível de campo até o nível de células. (PROFIBUS, 1999)

Segundo (PROFIBUS, 1999) o protocolo PROFIBUS opera em um sistema multi-mestre, sendo que o mestre é quem determina a comunicação de dados no barramento e os escravos são os dispositivos remotos como módulos de I/O's, válvulas, entre outros. As estações ativas, como são conhecidos os mestres, podem enviar mensagens desde que possuam o direito de acesso ao barramento, token, sem uma requisição externa. Já os escravos, estações passivas, somente podem enviar e reconhecer mensagens ao mestre quando forem solicitados.

A arquitetura da rede PROFIBUS é baseada no modelo OSI (Open System Interconnection) conforme o padrão internacional ISO 7498, porém com algumas camadas a menos como na figura 8. Apenas as camadas de enlace de dados, a camada física e a camada de implementação foram implementadas na arquitetura PROFIBUS.

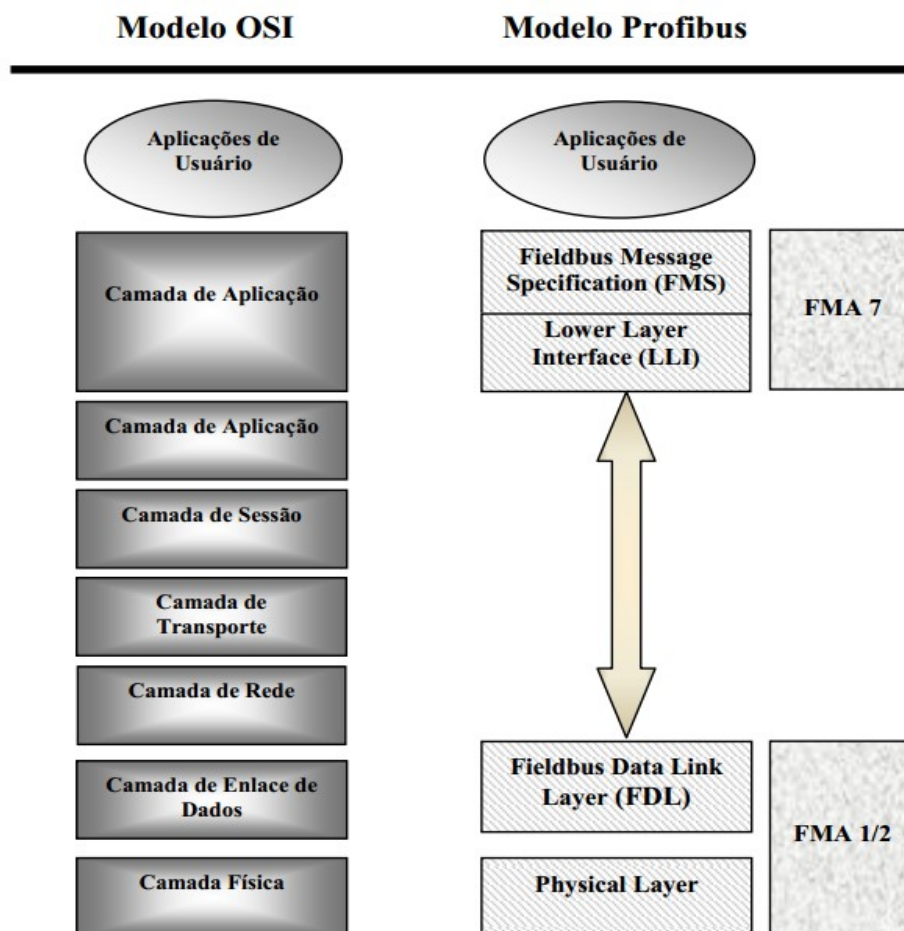


Figura 6: Modelo de comunicação reduzida FIELDBUS. (MORO, 2012, p.28)

Camada física – segundo (VILLAMAYOR, 1999, p.67) O chão de fábrica é afetada substancialmente pelo meio de transmissão e pela interface física do barramento. Os fatores como custo de instalação e integridade dos dados também é levado em consideração.

Os meios físicos de transmissão utilizados na rede PROFIBUS são:

RS-485 – é o meio mais comum utilizado, sendo aplicado com bastante intensidade em automação da manufatura, por apresentar alta taxa de transmissão aliada a uma instalação simples e custo reduzido. Para a maioria das aplicações um cabo par trançado de cobre blindado com um único par é suficiente. Segundo (PROFIBUS, 1999), a topologia permite adição e remoção de estações, sem afetar outras estações, simplificando expansões futuras que poderão ser implementadas com a rede em operação.

As velocidades de transmissão variam de 9,6 Kbit/s a 12 Mbit/s, a depender da distância e as características de cabeamento, tabela abaixo é possível ver a taxa de transmissão versus a distância máxima do barramento.

Taxa de transmissão [Kbits/s]	Comprimento máximo do barramento[m]
9,6 / 19,2 / 93,75	1200
187,5	600
500	200

Figura 7: Comparativo de taxa de transmissão e comprimento do barramento.
Fonte: (VILLAMAYOR, 1999, p.63)

IEC 61158-2 – a aplicação desse meio é voltada ao controle de processo. O PROFIBUS PA é a solução que atende os requisitos da automação de processo, onde se tem conexão de sistema de automação e sistemas de controle de processo com equipamentos de campo, tais como: transmissores de pressão, temperatura, conversores, posicionadores, etc. IEC 61158-2 – a aplicação desse meio é voltada ao controle de processo. (PROFIBUS, 2000)

Segundo (PROFIBUS, 2000) existem muitas vantagens na utilização desse meio de transmissão entre elas pode-se destacar a transmissão de informações confiáveis, tratamento de status das variáveis, sistema de segurança em caso de falha, alta resoluções nas medições, integrado com controle discreto em alta velocidade, aplicações em qualquer segmento, redução dos custos de instalação e manutenção, menor tempo de start-up, aumento significativo em funcionalidade e segurança. O PROFIBUS PA permite a medição e controle por uma linha a dois fios simples, além de alimentar os equipamentos de campo em áreas intrinsecamente seguras.

Para áreas classificadas como a indústria química e petroquímicas, o PROFIBUS PA foi desenvolvido junto a usuários da Industria de Controle e Professo (NAMUR). Para as áreas de segurança intrínsecas o PROFIBUS PA permite que os dispositivos de campos sejam energizados pelo próprio barramento. A taxa de transmissão definida em 31,25 Kbits/s funciona de forma síncrona conforme a norma IEC 6115-8. A norma determina que o meio físico seja um par de fios trançados.

Fibra ótica – Esse meio é utilizando quando se deseja aumentar o comprimento máximo com alta taxa de transmissão e para ambientes com grandes interferências magnéticas.

Camada enlace de dados – A segunda camada do modelo OSI fornece as funções de controle de acesso ao meio, integridade dos dados assim como executa o protocolo de transmissão e mensagem. (VILLAMAYOR, 1999, p.29).

A camada de enlace de dados é denominada FDL (Fieldbus Data Link). O Controle de acesso ao meio (MAC) define os procedimentos de permissão para que uma estação transmita dados, ou seja, o objetivo é assegurar que apenas uma estação transmita dados em um determinado momento.

Os requisitos básicos do MAC são:

- Durante a comunicação entre sistemas complexos de automação (mestres), deve ser assegurado que cada uma destas estações detém tempo suficiente para executar suas tarefas de comunicação dentro de um intervalo definido e preciso de tempo. (PROFIBUS, 2000)

- Por outro lado, a transmissão cíclica de dados em tempo real deverá ser implementada tão rápida e simples quanto possível para a comunicação entre um controlador programável complexo e seus próprios dispositivos de I/O's (escravos). (PROFIBUS, 2000)

- Portanto, o protocolo PROFIBUS de acesso ao barramento inclui o procedimento de passagem de Token, que é utilizado pelas estações ativas da rede (mestres) para comunicar-se uns com os outros, e o procedimento de mestre-escravo que é usado por estações ativas para se comunicarem com as estações passivas (escravos). (PROFIBUS, 2000)

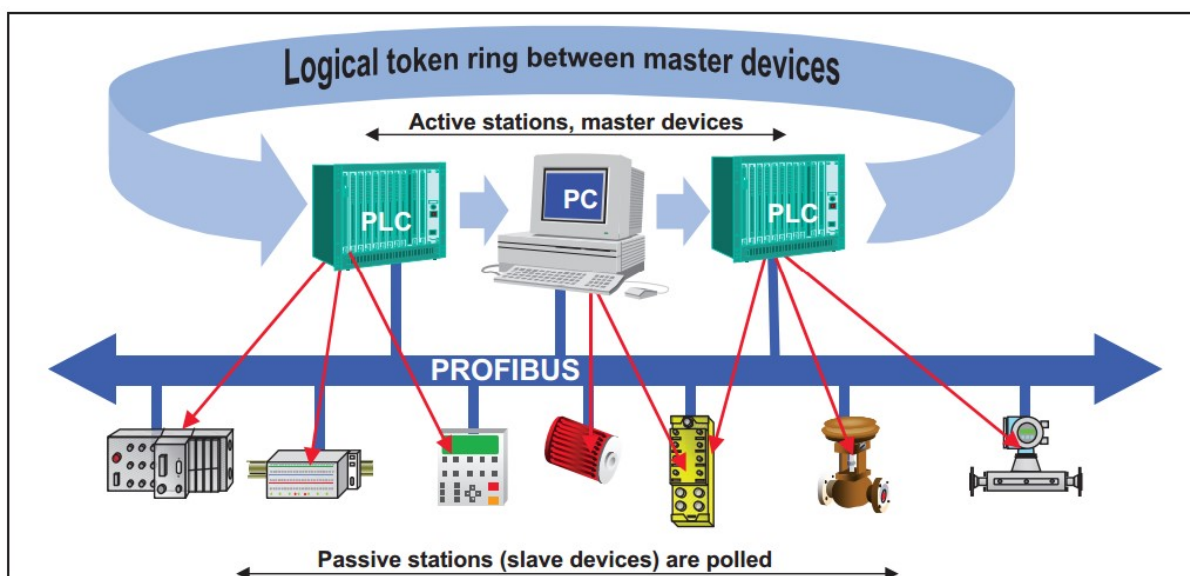


Figura 8: Configuração de uma rede PROFIBUS com mestres e escravos. (PROFIBUS, 2000)

Na figura acima mostra 3 estações ativas e 7 estações passivas, é possível observar na imagem que os 3 mestres forma um anel lógico de passagem de token, ou seja, a um determinado instante o PROFIBUS permite o acesso ao barramento para cada mestre. Durante o instante de tempo que um determinado mestre está ativo, ou seja, recebe a mensagem de Token, nesse tempo a estação mestre poderá comunicar-se com suas estações escravas e com todas as outras estações mestres.

Segundo (PROFIBUS, 2000), A mensagem de Token, nada mais é que a passagem de acesso de um mestre ao próximo mestre, devendo ser distribuída no anel lógico de Token para cada estação mestre em um intervalo de tempo máximo que é chamado de rotação de Token. Ou seja, a passagem de Token só é utilizada na comunicação entre as estações ativas.

O procedimento mestre-escravo permite ao mestre que no momento possui o Token acessar seus próprios escravos. O mestre pode enviar mensagens aos escravos ou ler mensagens dos escravos. Esse método de acesso permite as seguintes configurações de sistemas (PROFIBUS, 2000, p.15):

- Sistema puro mestre-escravo;
- Sistema puro mestre-mestre (com passagem de Token);
- Uma combinação dos dois; (PROFIBUS, 2000, p.15)

Segundo (VILLAMAYOR, 1999, p.56) A transmissão das mensagens na rede PROFIBUS opera em dois modos sem conexão, broadcast e multicast. Na comunicação em broadcast a estação ativa envia uma mensagem para todas as outras estações podendo ser mestre ou escravas sem confirmação de recebimento. Já a comunicação em multicast a estação ativa envia uma mensagem sem reconhecimento para um grupo de estações mestres e escravas.

Essa organização das estações em um anel de Token, baseia-se nos endereços de cada estação, nessa configuração o Token é passado de mestre para mestre de acordo com o endereço de cada estação em ordem crescente. Quando o sistema inicia, o MAC é responsável por essa lógica e é ele quem estabelece o anel de Token. Quando a rede está em operação, as estações defeituosas ou que estão fora de operação são removidas, assim como são adicionadas novas estações ao anel. Na configuração do Token é definido o tempo em que cada estação estará ativa.

Segundo (PROFIBUS, 2000, p.22), outras funções do Controle de Acesso ao Meio (MAC) são:

- Detecção no meio de transmissão ou no receptor;
- Detecção de erros de endereçamento;
- Detecção de erros na passagem do Token;

Uma função muito importante da camada 2 é que ela é responsável pela segurança dos dados que são trafegados, ou seja, é nessa camada do PROFIBUS que são formatados frames que asseguram a integridade dos dados.

O modo de operação da camada 2 é chamado “sem conexão”. A transmissão de dados pode ser ponto-a-ponto e multiponto (Broadcast e Multicast).

Comunicação Broadcast significa que uma estação ativa envia uma mensagem sem confirmação a todas outras estações (mestres e escravos). (PROFIBUS, 2000, p.14).

Comunicação Multicast significa que uma estação ativa envia uma mensagem sem confirmação a um grupo de estações pré-determinadas (mestres e escravos). (PROFIBUS, 2000, p.14).

Serviço	Função	DP	FMS	PA
SDA	Send Data with Acknowledge Envia dados com reconhecimento		•	
SRD	Send and Request Data with replay Envia e requisita dados com resposta	•	•	•
SDN	Send Data with No acknowledge Envia dados sem reconhecimento	•	•	•
CSRD	Cyclic Send and Request Data with replay Envia e requisita dados ciclicamente com resposta		•	

Figura 9: Serviços da camada de segurança de dados (Data Link Layer). (PROFIBUS, 2000, p.13)

Na figura acima é possível ver os diferentes serviços existentes para segurança dos dados e os modos PROFIBUS nos quais os serviços são implementados.

Camada de aplicação – A camada de aplicação do PROFIBUS é dividida em duas sub-camadas: a FMS (FieldBus Message Specification) e a LLI(Lower Layer Interface).

O perfil de comunicação FMS foi projetado para a comunicação no nível de células. Neste nível, controladores lógicos programáveis comunicam-se uns com os outros. Nesta área de aplicação, mais importante que um sistema com tempos de reação rápida é um sistema com diversidade grande de funções disponíveis. (PROFIBUS, 2000, p.16)

Através do modelo de comunicação PROFIBUS FMS é possível que aplicações distribuídas sejam unificadas em um processo comum utilizando do uso de relacionamentos de comunicação. O VFD (virtual field device) é a parte da aplicação que fica situada no dispositivo e pode ser acessada via comunicação.

Um dispositivo virtual é uma abstração de uma determinada classe de dispositivo real, tal como é apresentada ao usuário externo. O modelo VFD representa de forma única qual parte do processo aplicativo é visível e acessível pelo sistema de comunicação e define seu comportamento. (VILLAMAYOR, 1999, p.72)

O dicionário de objeto (OD) é o local onde são registrados todos os objetos de comunicação de um dispositivo FMS e esse dicionário contém diversas informações como:

- Descrição;
- Estrutura e tipo de dados;
- Associação entre os endereços internos e sua denominação no barramento;

Já os objetos de comunicação estáticos são registrados em outro dicionário, ou seja, o dicionário de objetos estáticos. Esses objetos uma vez configurados não podem ser modificados durante a operação. São cinco os tipos de objetos de comunicação:

- variáveis simples;
- matriz: série de simples variáveis do mesmo tipo;
- registro: série de variáveis simples de diferentes tipos;
- domínio;
- evento;

Também tem os objetos de comunicação dinâmica que são registrados no dicionário de objetos dinâmicos e que podem vir a ser modificados quando estão em operação.

Serviços FMS

Os serviços FMS são um subset dos serviços MMS (Manufacturing Message Specification), que foram otimizados para aplicações de barramentos e que foram estendidos por funções para a administração dos objetos de comunicação e gerenciamento de redes. (PROFIBUS, 2000, p.23)

Segundo (PROFIBUS, 2000, p.23), os serviços FMS são divididos da seguinte forma:

- **Gerenciamento do contexto** que tem a função de estabelecer ou encerrar conexões lógicas;
- **Acesso a variáveis** os quais tem a função de acessar variáveis, registros, matrizes ou listas de variáveis.
- **Gerenciamento de domínio** tem a função de transmitir grandes quantidades de memória;
- **Gerenciamento de chamada de programa** tem a função de fazer o controle dos programas;
- **Gerenciamento de eventos** tem a função de transmitir mensagens de alarmes;
- **VFD support** é utilizado para identificação e suporte.
- **Gerenciamento OD** que tem a função de acessar a leitura e escrita dos dicionário de objetivos;

O mapeamento das camadas 7 a 2 é gerenciada pela LLI que também realiza o controle de fluxo e monitoração da conexão.

Além dos serviços FMS, existe também o FMA7 (Fieldbus Management Layer 7) que faz o gerenciamento de rede. O FMA7 é opcional, permite uma configuração central, além de poder ser iniciada remota ou localmente.

2.3.6 DEVICENET

DeviceNet é uma rede digital, multi-drop para conexão entre sensores, atuadores e sistema de automação industrial em geral. Ela foi desenvolvida para ter máxima flexibilidade entre equipamentos de campo e interoperabilidade entre diferentes vendedores. (SMAR, 2015).

Segundo (CASSIOLATO, 2015), a rede DeviceNet foi desenvolvida pela Allen-Bradley em 1994 com base sobre o protocolo CAN (Controller Area Network) com especificação aberta e gerenciada pela DeviceNet Foundation. Em 1995, a tecnologia foi transferida para ODVA (Open DeviceNet Vendor Association) que é uma organização sem fins lucrativos que promove e divulga a rede.

De acordo com (SMAR, 2015) tem classificação no nível de rede chamada devicebus e apresenta como características principais a alta velocidade de comunicação, alto poder de diagnóstico, além de comunicação a nível de byte e comunicação com equipamentos discretos e analógicos.

A rede DeviceNet transporta informações do tipo cíclica para atuadores e sensores, ou seja, são dados que são trafegados de modo periódico entre o controlador e os dispositivos de campo. E também transporta informações do tipo acíclica, que estão indiretamente ligadas ao controle como configuração e diagnóstico. Esse tipo de informação é feita de forma eventual, ou seja, somente quando é realizada alguma configuração de equipamento e diagnóstico.

Uma rede DeviceNet pode conter até 64 dispositivos onde cada dispositivo ocupa um nó na rede, endereçados de 0 a 63. Qualquer um destes pode ser utilizado. Não há qualquer restrição, embora se

deve evitar o 63, pois esse costuma ser utilizado para fins de comissionamento. (SMAR, 2015)

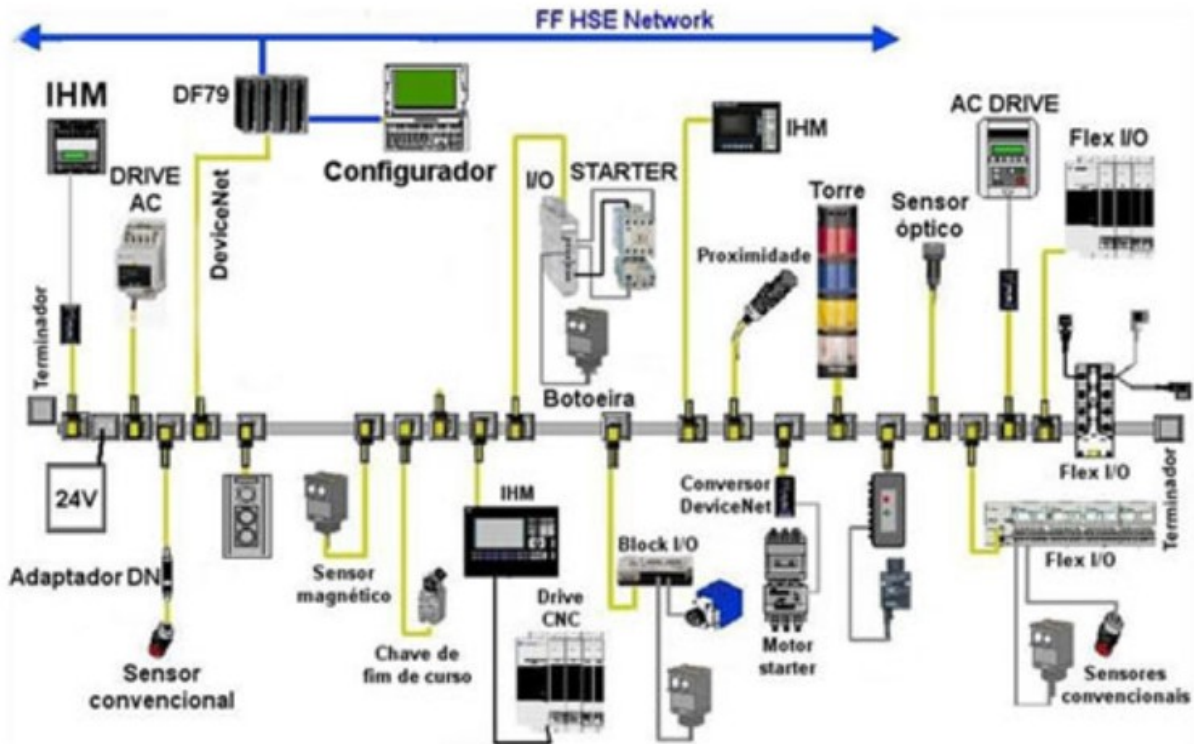


Figura 10: Exemplo de uma rede DeviceNet. (SMAR, 2015)

2.3.6.1 PROTOCOLO DEVICENET

DeviceNet é utilizada tecnologia aberta e padronizada de rede. Na camada de aplicação utiliza o CIP (Common Application Layer). Segundo [SMAR] A rede DeviceNet e ControlNet possuem estrutura comum de objetos, ou seja, independe do meio físico e da camada de enlace de dados. Essa padronização da camada de aplicação junto a interfaces de hardware e softwares abertos, torna uma plataforma de conexão universal em sistemas de automação, englobando desde o chão de fábrica até o nível da internet.

Os objetivos principais da CIP são:

- Transporte de dados de controle dos dispositivos de I/O's;
- Transporte de informações de configuração e diagnóstico do sistema sendo controlado. [SMAR]

Um nó DeviceNet é então modelado por um conjunto de objetos CIP, os quais encapsulam dados e serviços e determinam assim seu comportamento. (SMAR, 2015)

2.3.6.2 A CAMADA LINK DE DADOS

A rede DeviceNet é baseada no protocolo de comunicação CAN, ou seja, utiliza esse protocolo na camada de link de dados. Segundo (SMAR, 2015), o frame de dados DeviceNet utiliza o mínimo de largura de banda para transmissão das mensagens CIP. Na imagem abaixo é possível observar a estrutura do frame de dados DeviceNet.

2.3.6.3 MODOS DE COMUNICAÇÃO

Segundo (SMAR, 2015), O protocolo DeviceNet utiliza mensagens do tipo cycling I/O e explicit message, sendo utilizados para diferentes tipos de dados.

- **Cyclic I/O**: é um tipo de telegrama síncrono, utilizado em dados prioritários entre um produtor e um ou mais consumidores, sendo divididos em:

- **Polled**: nesse método, o mestre envia um telegrama para cada estação escrava de sua lista. Após o recebimento do telegrama, a estação escrava responde a solicitação do mestre. Essas solicitações são realizadas seguidamente até que todos os escravos respondam, após isso, inicia-se um novo ciclo de solicitações;

- **Bit-strobe**: nesse método, o mestre envia um telegrama de 8 bytes de dados na rede. Cada bit é desses 8 bytes é direcionado para uma unidade escrava, caso esteja endereçado, respondendo de acordo com a solicitação;

- **Change of state**: nesse método, a comunicação entre mestre e escravo se dá quando houver uma mudança nos valores monitorados/controlados, até um tempo limite. Após o estouro desse tempo, a comunicação ocorrerá mesmo sem que haja alguma alteração nos valores monitorados/controlados;

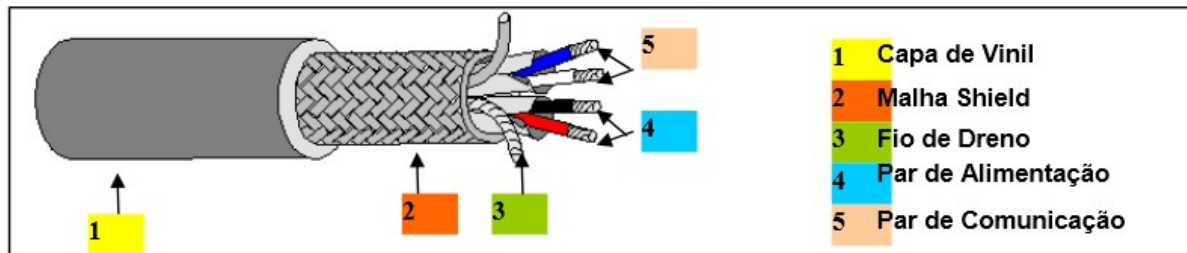
- **Cyclic**: nesse método, a comunicação ocorre em ciclos de tempo especificados, independente ou não de variações nos valores monitorados/controlados.

- **Explicit Message**: esse tipo de telegrama é utilizado em tarefas assíncronas e não prioritárias, ou seja, parametrização e configuração dos equipamentos.

2.3.6.4 MEIO FÍSICO E MEIO DE TRANSMISSÃO

Segundo (ALBUQUERQUE, 2015, p.2), dois pares de fios são utilizados na rede DeviceNet. Um par é destinado para a comunicação, sendo que um fio é de cor branca chamado de CAN high e outro de cor azul chamado de CAN LOW. O outro par de fios é utilizado para alimentação em corrente contínua dos equipamentos da rede, sendo um de cor vermelha (V+) e outro de cor preta (V-). Além disso, o cabeamento também conta com uma blindagem externa dos pares.

FIGURA 4 – Meio Físico DEVICENET



Fonte: [Redes de Comunicação de Dados, 2000]

Figura 11: Meio Físico DEVICENET. (PEREIRA, 2000, p.7)

Na figura acima mostra o cabo DeviceNet com seus fios e sua blindagem externa. Segundo (PEREIRA, 2000, p.7) uma técnica chamada de tensão diferencial para os níveis lógicos é utilizada para os sinais de comunicação. Essa técnica aliada com a blindagem minimiza as interferências eletromagnéticas fazendo com que a integridade da informação seja conservada.

2.3.6.5 TOPOLOGIA DE REDE

A rede DeviceNet utiliza diversas topologias para utilização e os componentes admissíveis. Na figura abaixo é possível ver as variedades de topologias que podem ser utilizadas.

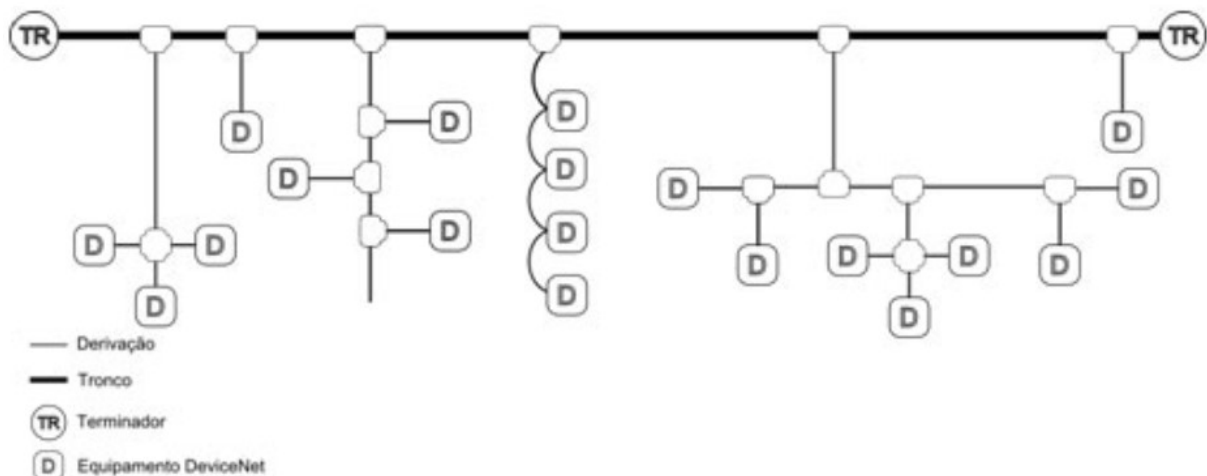


Figura 12: Topologia DeviceNet. (SMAR, 2015)

A taxa de transmissão da rede DeviceNet varia de 125 Kbits/s até 500 Kbits/s. Segundo (ALBUQUERQUE, 2000, p.3), para a grande maioria das aplicações, a taxa ideal é de 125Kbits/s, pois possui o melhor custo/benefício por permitir a maior quantidade de equipamentos devido ao maior comprimento possível do cabo. Um detalhe para a rede DeviceNet é que todos os equipamentos devem ser configurados para operarem com a mesma taxa de transmissão, caso contrário o funcionamento da rede pode ser comprometido.

Na tabela abaixo é feito um comparativo sobre a taxa de transmissão, comprimento máximo do cabo, o tipo do cabo e a função do cabo.

Tipo do Cabo	Função do Cabo	Taxa de Transmissão		
		125Kbits/s	250Kbits/s	500Kbits/s
Cabo Grosso	Tronco	500m	250m	100m
Cabo Fino	Tronco	100 m	100m	100m
Cabo Flat	Tronco	380m	200m	75m
Cabo Fino	Derivação	6m	6m	6m
Cabo Fino	S Derivações	156m	78m	39m

Figura 13: Comparativo tipos de cabeamento DeviceNet. (SMAR, 2015)

2.3 TECNOLOGIA OPEN-SOURCE

Segundo (CARVALHO apud RUSOVAN;LWAFORD; PARNAS, 2005, p.21), o software proprietário (SP) é protegido por *copyright* e licenças, além de ser caracterizado como um bem de propriedade intelectual. Uma outra característica importante do SP é que o seu código fonte não é disponibilizado ou acessível ao usuário. Já o software Open-Source, diferentemente do SP, o código fonte é aberto e distribuído, ou seja, permite que o usuário estude e faça modificações que atendam suas necessidades.

Há também que se diferenciar softwares Open-Source de software livre. O primeiro termo é empregado para softwares licenciados com licenças aprovadas pela Open Source Initiative e também necessitando estar em conformidade com alguns requisitos impostos pela The Open Definition. Já o segundo, tem que estar de acordo com The Free Software Definition que garante o software deve ter alguns requisitos de liberdade, ou seja, livre execução, estudo, redistribuição e melhoria.

Os princípios do SL (software livre)/ OS (open-source) fundamentam-se nas premissas básicas de liberdade de expressão, acesso a informação e do caráter eminentemente coletivo do conhecimento, que deve ser construído e disponibilizado democraticamente, e não privatizado. (GIANCARLO, 2005).

O Software Open-Source tem um conjunto de princípios que devem ser seguidos:

- 1) Distribuição livre, sem pagamento de royalties ou semelhante;
- 2) Código fonte deve sempre estar aberto;
- 3) Permitir modificações e trabalhos derivados;
- 4) Garantir integridade autoral do código fonte;

- 5) Não discriminar pessoas ou grupos;
- 6) Não discriminar áreas de conhecimento, setores e atividades;
- 7) Direitos de licença redistribuídos sem necessidade de licenças adicionais pelas partes;

Para a automação industrial o Software Open-Source além das características citadas acima, ele deve também possuir outras como:

- 1) Portabilidade – é importante que haja a possibilidade de executar o mesmo software em diferentes plataformas, ou seja, tanto em hardwares diferentes como em softwares.
- 2) Interoperabilidade – se no desenvolvimento do software for dada atenção a todos os problemas de portabilidade, então o software será capaz de ser executado em diferentes plataformas simultaneamente em uma mesma rede de trabalho, todos se comunicando e interagindo sem problemas.
- 3) Modularidade – essa característica permite que o software seja modificado com um impacto insignificante nos outros módulos.
- 4) Escalabilidade – essa habilidade permite que o software rode em centros de controles com tamanhos diferentes e que também podem apresentar funções diferentes.

Em resumo, um sistema aberto é aquele que deve ser interoperável e intercambiável, ou seja, os seus componentes podem ser adquiridos de um ou outro fornecedor, de acordo com um critério definido, seja qualidade, preço, atualização ou outro motivo. (WATANABE, 2006, p.58). Além disso, é importante que qualquer equipamento possa ser substituído por outro de um fabricante diferente sem que haja necessidade de fazer alterações no sistema. Para que isso ocorra é necessário que o fabricante siga rigorosamente o protocolo e que o sistema esteja documentado e aberto.

2.3.4.1 SISTEMA OPERACIONAL LINUX

O LINUX é um sistema operacional que tem a função de fazer a comunicação entre o hardware do equipamento onde será instalado e o software. Esse sistema é baseado no UNIX que foi feito para rodar em desktops residenciais possuindo como características semelhantes a robustez, segurança e flexibilidade. Utilizando um Software Open-Source, permite que o desenvolvimento do mesmo tenha maior liberdade.

O LINUX apresenta diversas tecnologias como o uso de multitarefa real, possibilidade de implementar TCP/IP, além de memória virtual e proteção entre processos. Aliando isso a possibilidade de utiliza-lo como servidor de aplicações e também como estação de trabalho, fazendo apenas algumas poucas alterações.

CAPÍTULO III – PROJETO E IMPLEMENTAÇÃO

Nesse capítulo de projeto e implementação será descrito o equipamento utilizado, o sistema operacional, a linguagem de programação, o software de desenvolvimento. O objetivo principal é criar um sistema de supervisão utilizando tecnologia open-source reduzindo os custos com softwares proprietários ao menor valor com a mesma qualidade e simplicidade.

3.1 SOFTWARE DE DESENVOLVIMENTO: NETBEANS 8.0 E JAVA 7.0

O software NetBeans 8.0 é uma ferramenta que fornece analisadores de código e editores prontos para programar com as tecnologias de programação Java. O editor conta com aprimoramentos constantes, muitas funcionalidades avançadas e uma extensa linha de ferramentas, modelos e exemplos.

A linguagem de programação, por sua vez, utilizada foi a Java na versão 7.0. O Java apresenta como uma de suas principais características a portabilidade, pois permite que seja utilizada em diferentes plataformas como Windows, Linux, Mac, Web, Celular sem problemas com compatibilidade. Desenvolvida pela Oracle Corporation e lançada em 1995, é uma linguagem de alto nível e orientada a objetos.

Uma outra vantagem dessa linguagem de programação é o gerenciamento automático de memória e um coletor de lixo, facilitando a programação apesar de exigir um pouco mais de processamento.

3.2 SISTEMA OPERACIONAL – LINUX UBUNTU 14.04

O sistema operacional Ubuntu é uma distribuição Linux que tem como foco principal a facilidade de uso por usuários comuns. É uma distribuição que apresenta um maior suporte, uma maior comunidade, além de ser a versão do Linux mais popular. O próprio manual do Ubuntu disponibilizado no menu “Sistema” e em seguida “sobre o Ubuntu” cita a seguinte definição sobre o sistema:

O Ubuntu é um sistema operacional de código totalmente aberto, construído em volta do kernel Linux. A comunidade do Ubuntu é construída em volta dos ideais descritos na Filosofia Ubuntu: que software deve ser disponibilizado gratuitamente, que ferramentas de software devam ser usáveis pelas pessoas em suas línguas locais e apesar de qualquer deficiência, e que as pessoas devem ter a liberdade de personalizar e alterar o software de qualquer maneira que os atenda. Por estas razões:

- Ubuntu será sempre gratuito, e não há cobrança adicional para a “edição empresarial”; disponibilizamos nosso melhor trabalho para todos nos mesmos termos de gratuidade.

- O Ubuntu inclui as melhores traduções e infraestrutura de acessibilidade que a comunidade de software livre tem a oferecer, fazendo o Ubuntu útil ao maior número de pessoas possível.
- Novas versões do Ubuntu são liberadas regularmente; uma nova versão é feita a cada seis meses. Você pode usar a versão estável ou de desenvolvimento. Cada versão é suportada por, no mínimo, 18 meses.
- O Ubuntu é totalmente comprometido com os princípios de desenvolvimento de software livre; nós encorajamos as pessoas a utilizarem, melhorarem e distribuírem software livre.

3.3 SIEMENS SIMATIC S7-300

O S7-300 é um controlador lógico programável modular utilizado em sistemas centralizados de pequeno a médio porte. Por apresentar uma arquitetura modular, esse controlador economiza espaço físico além de flexibilidade de configuração e expansão. As especificações são:

- Módulos de I/O (SM)
 - Digitais (24Vdc, 48-130 Vuc, 120/230 VAC, Relé etc)
 - Analógicos ($\pm 5V$, 0-10V, 0/4 - 20mA, Hert etc)
- Módulos de Comunicação (CP)
 - Profibus DP / FMS
 - Ethernet
 - AS-interface
 - Serial Ponto-a-Ponto
 - Modbus
- Módulos de Função (FM)
 - Contadores rápidos
 - Saídas de pulso rápida
 - Controle de posição
 - Controle de motor de passo
 - Controle em malha fechada (PID)

É possível a expansão através da utilização de até 32 módulos em uma configuração centralizada.

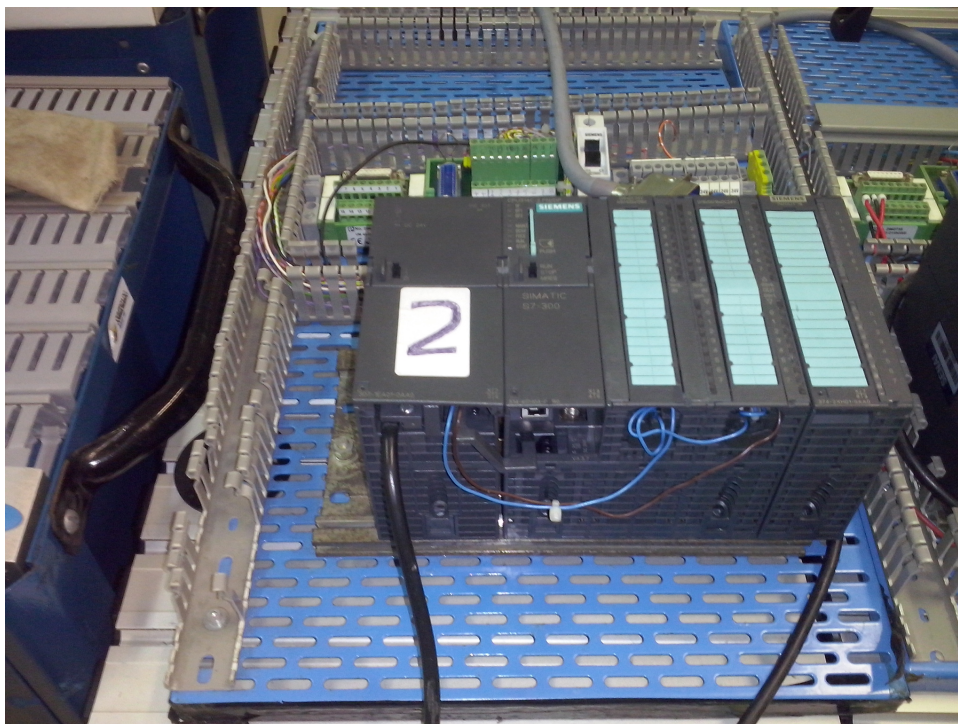


Figura 14: Siemens S7-300. (AUTOR, 2015)

3.4 APLICAÇÃO

Este projeto implementa uma abordagem da comunicação por ISO TCP nas redes de dados industriais utilizando socket java e tecnologias abertas, com o objetivo de possibilitar a gerência das informações de processos industriais. O resultado foi o desenvolvimento de uma aplicação usando socket java orientado a conexão, capaz de prover dados contidos em um CPL da Siemens S7-300. A comunicação entre a aplicação cliente e o servidor é através do protocolo de transporte TCP/IP e possibilita a visualização e manipulação on-line dos dados existentes no CLP.

3.3.1 ARQUITETURA DO SISTEMA

A modelagem da arquitetura do sistema proposto neste trabalho é o grande diferencial aos padrões utilizados na indústria atualmente, através da mudança de foco da programação distribuída, orientada a objetos, para a programação distribuída orientada a serviço. Saindo, assim, de padrões rígidos e onerosos para padrões abertos e de domínio público. A Figura 15 ilustra os componentes básicos da arquitetura.

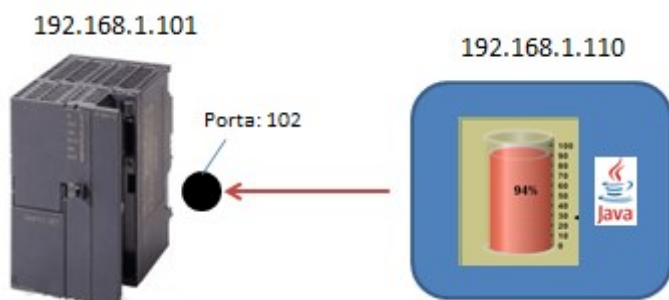


Figura 15 – Visão macro da arquitetura. (AUTOR, 2015)

O CLP host [192.168.1.101] trata-se do processo servidor, associado a ele existe um endereço IP e uma porta de serviço conhecidos para que seja possível a utilização do serviço pelos clientes, já o processo cliente, host [192.168.1.110] trata-se de uma aplicação java desktop, a troca de dados entre os processos ocorrem utilizando a primitiva de comunicação socket implementada no lado cliente em java.

3.4.2 CODIFICAÇÃO DO SISTEMA

A forma de codificação do processo cliente adotada para este trabalho foi utilizando a linguagem Java e ferramentas livres, tentando simplificar ao máximo a codificação e implantação. A figura 16 apresenta a implementação do método responsável pela criação do socket cliente.

```

DataIsoTCP(String host) {
    IP = host;
    //Nodave.Debug=Nodave.DEBUG_ALL;
    buf = new char[Nodave.OrderCodeSize];
    buf1 = new byte[Nodave.PartnerListSize];
    try {
        sock = new Socket(host, 102);
    } catch (IOException e) {
        System.out.println(e);
    }
}

```

Figura 16 – Método de criação do socket. (AUTOR, 2015)

A figura 16 apresenta a implementação do método responsável pelo estabelecimento do canal de comunicação entre a aplicação cliente e o CLP.

```

public static void StartConnection() {
    Connection = false;
    OutputStream oStream = null;
    InputStream iStream = null;
    slot = 2;
    if (sock != null) {
        try {
            oStream = sock.getOutputStream();
        } catch (IOException e) {
            System.out.println(e);
        }
        try {
            iStream = sock.getInputStream();
        } catch (IOException e) {
            System.out.println(e);
        }
        di = new PLCinterface(oStream, iStream, "IF1", 0, Nodave.PROTOCOL_ISOTCP);

        dc = new TCPConnection(di, 0, slot);
        int res = dc.connectPLC();
        if (0 == res) {
            Connection = true;
            System.out.println("Connection OK ");
        } else {
            System.out.println("No connection");
        }
    }
}
}

```

Figura 17 – Método de criação do canal de comunicação. (AUTOR, 2015)

A figura 17 apresenta a implementação dos métodos responsáveis pela leitura dos dados originadas no CLP.

```

public static void ReadData(int area, int db, int address, int bytes) {
    i = 0;
    if (area == 1) {
        dc.readBytes(Nodave.INPUTS, 0, address, bytes, null);
    } else if (area == 2) {
        dc.readBytes(Nodave.FLAGS, db, address, bytes, null);
    } else if (area == 3) {
        dc.readBytes(Nodave.OUTPUTS, 0, address, bytes, null);
    }
}

public static int ReadOutputs(int address, int bytes) {
    int res = dc.readBytes(Nodave.OUTPUTS, 0, address, bytes, null);
    if (res == 0) {
        b1 = (byte) dc.getBYTE();
        b2 = (byte) dc.getBYTE();
        b3 = (byte) dc.getBYTE();
    }
    return res;
}
}

```

Figura 18 – Métodos de leitura dos dados. (AUTOR, 2015)



Figura 19 – Tela da aplicação. (AUTOR, 2015)

Na figura 18 e 19 é possível observar a aplicação que faz uso da conexão com o CLP, caso a conexão seja estabelecida periodicamente a aplicação realiza a leitura dos dados e atualiza a interface da aplicação.

CONSIDERAÇÕES FINAIS E RECOMENDAÇÕES

Esse trabalho iniciou-se através de uma revisão bibliográfica das principais redes industriais utilizadas em automação como AS-I que é voltada para o nível mais baixo da pirâmide de automação, ou seja, a parte de interligação dos sensores e atuadores. Outra rede abordada foi a Profibus e suas diversas variações como a Profibus-DP, Profibus-PA e Profibus-FMS os quais tem um ampla gama de aplicações na área de processo, manufatura e automação predial.

Já a rede DeviceNet trata-se de uma rede digital, multi-drop com aplicação em automação industrial em geral, desenvolvida para uma maior flexibilidade entre os equipamentos de campo e interoperabilidade entre os diversos fabricantes. A Ethernet é baseado na norma IEEE 802.3, definindo os padrões de transmissão e interconexões entre redes locais.

Em seguida foi abordado sobre tecnologia open-source e alguns requisitos e características que os softwares utilizados em automação industriais devam apresentar como a portabilidade, importante para utilização em diferentes plataformas; a interoperabilidade, utilização em diversos sistemas simultaneamente e em uma mesma rede de trabalho interagindo sem problemas; modularidade, permite modificações em alguns módulos do software sem impactar nos outros módulos; e por fim a escalabilidade que permite que o software funcione em centros de controle com diferentes tamanhos. Ainda sobre open-source, foi feito um breve abordagem sobre o LINUX que é um sistema operacional.

Na parte de aplicação foi utilizado o LINUX em sua versão UBUNTO 14.04, o Netbeans 8.0 e a linguagem de programação Java 7.0 para desenvolvimento do aplicativo de supervisão. Na parte de hardware, foi utilizado um CLP Siemens Simatic S7-300.

A implementação foi baseado numa abordagem de comunicação ISO TCP nas redes de dados utilizando socket java e tecnologias open-source. A aplicação trouxe

a capacidade de prover dados contidos em um CLP da Siemens. Além disso, por utilizar o protocolo de transporte TCP/IP possibilitou a manipulação on-line dos dados existentes no CLP.

Outro fator importante da aplicação foi o fato de a modelagem da arquitetura utilizada trazer um diferencial muito grande em relação aos padrões atualmente utilizados na industrial através da mudança de foco da programação distribuída, orientada a objetos, para a distribuída orientada a serviços.

Os Objetivos dessa pesquisa foram alcançados satisfatoriamente. O estudo das diversas tecnologias de redes industriais existentes no mercado possibilitou uma ampliação dos conhecimentos através de suas diversas aplicações e características. Já o desenvolvimento de um software utilizando tecnologia open-source para supervisão de um processo industrial foi bastante produtivo, ou seja, a aplicação funcionou utilizando ferramentas gratuitas e com qualidades suficientes para ser utilizados em diversos processos industriais e com uma importante redução de custos.

A evolução natural desse trabalho é o desenvolvimento do software para uma aplicação maior, um sistema de supervisão mais amigável com o usuário com diversas funcionalidades que gerencie melhor o processo sempre utilizando tecnologia open-source que é uma tendência para os próximos sistemas.

Futuramente pode-se traçar um comparativo do sistema com atuais ferramentas existentes, sejam elas proprietárias ou open-source, destacando as principais vantagens de cada um para uma ou várias aplicações específicas, além das desvantagens, incluindo também um comparativo dos custos para aquisição dos softwares proprietários e o custo final de um sistema completo.

REFERÊNCIAS BIBLIOGRÁFICAS

ALBUQUERQUE, Ana Claudia Medeiros Lins de. A Tecnologia DeviceNet. 2000. 5 f. Curso de Engenharia da Computação e Automação, Universidade Federal do Rio Grande do Norte, Natal, 2000.

Almeida, M. B. Protocolo AS-I: Agregando Inteligência a Sensores e Atuadores. Disponível em: <http://www.profibus.org.br/news/abril2007/news.php?dentro=2>. Acesso em: 20 de set. 2015.

ALMEIDA, Marcelo Barros de. PROTOCOLO AS-I: AGREGANDO INTELIGÊNCIA A SENSORES E ATUADORES. Disponível em: <<http://www.profibus.org.br/news/abril2007/news.php?dentro=2>>. Acesso em: 01 set. 2015.

CARVALHO, Luciana Guimarães. Características da utilização de software de código aberto: Um estudo sobre o setor de tecnologia da informação. 2013. 63 f. Dissertação (Mestrado) - Curso de M Sistemas de Informação e Gestão do Conhecimento, Universidade Fumec - Faculdade de Ciências Empresariais, Belo Horizonte, 2013.

CASSIOLATO, César. Redes Industriais. Disponível em: <<http://www.smar.com/newsletter/marketing/index150.html>>. Acesso em: 20 set. 2015.

ELECTRIC, Schneidr. Documento Técnico - Redes de Comunicação Industrial. 2007. Disponível em: <http://www.schneiderelectric.pt/documents/product-services/training/doctecnico_redes.pdf>. Acesso em: 06 set. 2015.
em: 10 set. 2015

Giancarlo Nuti Stefanuto (Org.). O impacto do Software Livre e do Código Aberto na Industria de Software do Brasil.2005. Disponível em:

<<http://www.fumec.br/revistas/sigc/article/viewFile/1931/1227>>. Acesso em: 10 out. 2015.

GROOVER, M. P. Automação industrial e sistema de manufatura. 3 ed. São Paulo: Pearson, 2011. 374p.

MORO, Marcos Paulo. Análise e estimativa de desempenho de Redes Profibus. 2012. 99 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Computação, Universidade Federal do Rio Grande do Sul. Instituto de Informática., Porto Alegre, 2012.

PEREIRA, João Paulo. COMUNICAÇÃO DE DADOS EM REDES DE AMBIENTES INDUSTRIAIS – REDES PROFIBUS E DEVICENET. 2000. 13 f. Curso de Ciência da Computação, Universidade Presidente Antônio Carlos (unipac), Barbacena, 2000. Pinheiro, J. M. S. Curso de Tecnologia em Redes de Computadores. Disponível em: https://www.projotoderedes.com.br/aulas/ugb_redes_l/ugb_redes_l_material_de_apoio_02.pdf. Acesso em: 10 de set. 2015.

PINHEIRO, Jose Maurício S.. Curso de Tecnologia em Redes de Computadores. Disponível em: <http://www.projotoderedes.com.br/aulas/ugb_redes_l/ugb_redes_l_material_de_apoio_02.pdf>. Acesso em: 01 out. 2015.

PROFIBUS, Associação PROFIBUS Brasil - PROFIBUS - Descrição Técnica, São Paulo, Oubrutro 2000. Disponível em: <http://www.profibus.com.br>. Acesso em: 10 de set. 2015.

RIBEIRO, M. A. Automação industrial. 4 ed. Salvador: Tek Treinamento Consultoria LTDA, 1999. 340p.

SMAR - Redes industriais. OUT/12. Disponível em: <http://www.smar.com/brasil/artigo-tecnico/redes-industriais> - Acesso

SMAR, Equipamentos Industriais. DeviceNet. Disponível em:
<<http://www.smar.com/brasil/devicenet>>. Acesso em: 14 set. 2015.

TANENBAUM, A. S. – Redes de Computadores – 4ª Ed., Editora Campus (Elsevier), 2003.

VILLAMAYOR, Miriam Noemi Caceres. Estudo da utilização do Protocolo PROFIBUS para Aplicações com Grande Número de Estações Escravas. 1999. 155 f. Dissertação (Mestrado) - Curso de Engenharia de Minas, Metalurgia e Materiais, Universidade Federal do Rio Grande do Sul - Ufrs, Porto Alegre, 1999.

WATANABE, Edson Hiroshi. APLICAÇÃO DE SOFTWARE ABERTO EM REDES INDUSTRIAIS. 2006. 177 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Universidade Federal do Paraná - Ufpr, Curitiba, 2006.