



Universidade Federal de Uberlândia
Faculdade de Engenharia Elétrica

ALINE GONÇALVES FERNANDES

**SUPERVISÃO E CONTROLE DE TEMPERATURA EM TÚNEIS DE
RESFRIAMENTO INDUSTRIAIS DE PEQUENO PORTE UTILIZANDO
TECNOLOGIAS DE AUTOMAÇÃO OPEN SOURCE**

Uberlândia
2017

ALINE GONÇALVES FERNANDES

**SUPERVISÃO E CONTROLE DE TEMPERATURA EM TÚNEIS DE
RESFRIAMENTO INDUSTRIAIS DE PEQUENO PORTE UTILIZANDO
TECNOLOGIAS DE AUTOMAÇÃO OPEN SOURCE**

Trabalho apresentado como requisito de avaliação
na disciplina Trabalho de Conclusão de Curso do
Curso de Engenharia Elétrica da Universidade
Federal de Uberlândia.

Orientador: Márcio José da Cunha

Uberlândia
2017

ALINE GONÇALVES FERNANDES

Supervisão e Controle de Temperatura em Túneis de Resfriamento Industriais de Pequeno Porte Utilizando Tecnologias de Automação Open Source

Trabalho apresentado como requisito de avaliação na disciplina Trabalho de Conclusão de Curso do Curso de Engenharia Elétrica da Universidade Federal de Uberlândia.

Banca de Avaliação:

Prof. Dr. Márcio José da Cunha
Orientador

Prof. Me. Renato Santos Carrijo
Membro

Prof. Dr. Josué Silva de Moraes
Membro

Uberlândia (MG), 18 dezembro de 2017

Dedico este trabalho à minha família, por todo o apoio e compreensão, e por ter me permitido chegar até aqui.

AGRADECIMENTOS

Agradeço a Deus, pela minha vida e por me guiar todos os dias.

Agradeço às pessoas que mais me incentivaram a seguir meus próprios sonhos, que me apoiaram em todos os momentos, que seguraram nas minhas mãos e foram meu suporte, meu porto seguro quando eu pensava que não conseguiria mais, meus pais Vitalino, que não está mais entre nós, e à minha mãe Suely.

Agradeço aos meus progenitores Cirene e Allan (in memoriam) que me deram a possibilidade de vir ao mundo, por estarem próximos e sempre me protegendo em suas orações.

Agradeço a minha pequena filha Luiza, que tem sido a minha maior inspiração para prosseguir nesta jornada. Minha fonte de responsabilidade, paz, tranquilidade, amor, família.

Agradeço aos meus irmãos que foram meus exemplos de vida, minha influência direta para andar corretamente nos caminhos da vida.

Ao meu orientador Prof. Márcio José da Cunha por ter se disponibilizado a me auxiliar tão prontamente, pelos conselhos profissionais que me encorajaram a não oferecer nada menos do que o meu melhor.

Aos demais professores da Faculdade de Engenharia Elétrica, por terem contribuído na minha maturidade como profissional e por toda a assistência prestada.

Aos meus colegas de turma pela convivência, pela troca de aprendizado e experiências.

Agradeço também a todos os outros que contribuíram direta ou indiretamente na minha formação e na realização deste trabalho.

RESUMO

Este trabalho de conclusão de curso apresenta uma solução em supervisão e controle em ambientes industriais de pequeno porte, através do controle da temperatura interna de um túnel de resfriamento a partir da variação da velocidade do rotor de um ventilador. O desenvolvimento do projeto foi possível através da comunicação Modbus TCP/IP, entre o sistema supervisório ScadaBR e a planta industrial, processada pela plataforma de desenvolvimento Beaglebone Black.

Palavras-chave: Automação Industrial; Sistema Supervisório; Modbus TCP/IP.

ABSTRACT

This end of course project presents a solution in supervision and control in small industrial environments, through the control of the internal temperature of a cooling tunnel from the variation of the rotor speed of a fan. The development of the project was proposed through Modbus TCP/IP communication, between the ScadaBR supervisory system and an industrial plant, processed by the Beaglebone Black development platform.

Keywords: Industrial automation; Supervisory system; Modbus TCP / IP.

LISTA DE ILUSTRAÇÕES

Figura 1 - Pirâmide Hierárquica da Automação Industrial.	5
Figura 2 - O processo de tecelagem usando um tear jacquard.	7
Figura 3 - Elementos básicos do modelo de referência OSI.	8
Figura 4 - Representação das camadas propostas no modelo OSI.	9
Figura 5 - Rede de comunicação com a topologia em barra.	11
Figura 6 - Rede de comunicação com a topologia em anel.	11
Figura 7 - Rede de comunicação com a topologia em estrela.	12
Figura 8 - Rede de Comunicação com a topologia em grafo.	12
Figura 9 - Rede de comunicação com a topologia em árvore.	13
Figura 10 - Pilha de comunicação Modbus	14
Figura 11 - Arquitetura de rede utilizando Modbus -	15
Figura 12 - Estrutura do frame Modbus.	17
Figura 13 - Exemplo de cabo para comunicação RS-232.	17
Figura 14 - Exemplo de uma ligação RS-485 entre dispositivos.	17
Figura 15 - Frame Modbus com o cabeçalho MBAP	18
Figura 16 – Requisição de ligação.	19
Figura 17 – Requisição de ligação com IP e Porta disponíveis.	19
Figura 18 - Diagrama simplificado de um sistema de controle automático.	20
Figura 19 - Exemplos de sensores discretos.	21
Figura 20 - Exemplos de sensores analógicos.	21
Figura 21 - Exemplos de atuadores.	22
Figura 22 - Sistema de controle por realimentação.	23
Figura 23 - Sistema de controle a malha aberta.	23
Figura 24 - CLP Siemens SIMATIC S7 – 1200.	24
Figura 25 - Componentes do CLP.	25
Figura 26 - Exemplo de interface de um sistema SCADA –	26
Figura 27 - Interfacimento do sistema de resfriamento.	26
Figura 29 - Interface do ScadaBR para controle de planta industrial.	29
<i>Figura 30 - Beaglebone Black.</i>	29
Figura 31 - Detalhamento dos Pinos da BeagleBone Black.	30
Figura 32 - Configuração do PuTTY para conexão com a Beaglebone Black.	31
Figura 33 - Sensor de Temperatura LM35.	32
Figura 34 - L298 Módulo Driver de Motor.	33
Figura 35 - Entradas e saídas do módulo L298.	Erro! Indicador não definido.
Figura 36 – Micro ventilador DC 12 V.	35

Figura 37 – Componentes para a montagem da esteira. a) Chassi. b) Sistema de engrenagens para acionamento. c) Mini Motor DC.	36
Figura 38 - Configuração das propriedades do Data source Modbus IP.	45
Figura 39 - Data points.	46
Figura 41 - Tela Inicial do Sistema Supervisório.	46
Figura 42 - Visão Geral do Sistema Supervisório.....	47
Figura 43 - Alarmes configurados para o Sistema Supervisório.....	48
Figura 44 - Gráficos configurados para o Sistema Supervisório.....	49

LISTA DE TABELAS

Tabela 1 – Exemplos dos Códigos de Função utilizados na comunicação Modbus.	16
Tabela 2 - Faixa de temperaturas monitoradas.....	27
Tabela 3 – Características da Beaglebone Black.	31
Tabela 4 - Ligação entre o LM 35 e a Beaglebone Black.	33
Tabela 5 - Características construtivas do módulo L298.....	34
Tabela 6 - Descrição dos pinos do módulo L298.....	34
Tabela 7 - Ligação entre o módulo L298 e a Beaglebone Black.....	35
Tabela 8 - Ligação entre o cooler e o módulo L298	35
Tabela 9 - Ligação entre o mini motor e o módulo L298.....	37
Tabela 10 - Ligação entre o módulo L298 e a Beaglebone Black.....	37
Tabela 11 - Correspondência dos pinos e seus endereços de registro na Beaglebone	38
Tabela 12 - Faixa de temperaturas monitoradas.	48

LISTA DE ABREVIATURAS E SIGLAS

CLP – Controlador Lógico Programável

CN – Comando Numérico

CNC – Comando Numérico Computadorizado

EIA – Associação da Indústria Eletrônica do inglês *Electronic Industry Association*

HTML – Linguagem de Marcação de Hipertexto, do inglês *HyperText Markup Language*

IHM – Interface Homem-Máquina

IP – Internet Protocol

ISO – Organização Internacional de Padrões, do inglês *International Standards Organization*

OSI – Modelo de Referência para Interconexão de Sistemas Abertos, do inglês *Reference Model for Open Systems Interconnection*

SCADA – Sistemas de Supervisão e Aquisição de Dados, do inglês *Supervisory Control and Data Acquisition*

SDCD – Sistema Digital de Controle Distribuído

SGBD – Sistema de Gerenciamento de Banco de Dados

TCP - Protocolo de Controle de Transmissão, do inglês *Transmission Control Protocol*

SUMÁRIO

1. INTRODUÇÃO	1
1.1. Justificativa	2
1.2. Objetivos.....	3
1.3. Estrutura do Projeto.....	3
2. REVISÃO BIBLIOGRÁFICA.....	4
2.1. A Automação Industrial.....	4
2.2. Redes de Comunicação	8
2.2.1. Topologia das Redes	10
2.2.2. Protocolos de comunicação industrial.....	13
2.2.2.1. Modbus.....	14
2.2.3. Sockets de Comunicação.....	18
2.3. Elementos de um Sistema de Controle Automático.....	19
2.3.1. Sensores e Atuadores.....	20
2.3.2. Sistema de Controle.....	22
2.3.3. Dispositivos para Controle Automático	23
2.3.3.1. Controlador Lógico Programável (CLP).....	24
2.4. Sistemas Supervisórios	25
3. DESENVOLVIMENTO	26
3.1. ScadaBR	28
3.2. Beaglebone Black.....	29
3.3. Libmodbus	32
3.4. Sensor de Temperatura LM35.....	32
3.5. Módulo L298.....	33
3.6. Micro Ventilador DC.....	35
3.7. Esteira	36
3.8. Configuração da conexão cliente/servidor Modbus	37
4. RESULTADOS.....	46
5. CONSIDERAÇÕES FINAIS	49
6. REFERÊNCIAS.....	51

1. INTRODUÇÃO

A evolução das tecnologias no campo da eletrônica e computação, trouxeram consigo o avanço nos processos de parametrização da informação para gestão melhorada dos recursos industriais. Os procedimentos necessários para a manufatura e fabricação em grande escala estão, cada vez mais, sendo monitorados a fim de alcançar as metas de redução de custo, maximização dos lucros e uso sustentável dos meios de produção, (ROSÁRIO, 2015).

Os processos industriais caracterizam a transformação de matéria-prima em um produto final através da utilização de máquinas e controle de fabricação. O uso de controladores e computadores aplicados ao controle automático são destinados para que a qualidade, otimização de tempo, custo e outras variáveis pudessem ser melhoradas. As plantas industriais, que podem ser conceituadas como sendo a estrutura da produção, ou onde ocorrem os processos industriais, são compostas por todos os elementos físicos utilizados no sistema. A instrumentação da planta é realizada para definir todos os equipamentos necessários juntamente com a conexão entre eles, (ROSÁRIO, 2015).

A aquisição de dados do sistema acontece através das máquinas, componentes e dispositivos da planta, como sensores, atuadores e dispositivos de campo. A informação recebida é comunicada aos dispositivos de controle automático, onde são realizadas operações lógicas e matemáticas a fim de devolver instruções para o sistema e fornecer dados para tomada de decisões de acordo com a estratégia programada, (ROSÁRIO, 2015).

Os controladores lógicos programáveis, ou CLPs, são caracterizados como computadores industriais para aquisição de dados, controle de máquinas e processos em diferentes aplicações. Foram desenvolvidos para trabalhar em condições industriais extremas e em ambientes agressivos com alto índice de poeira, ruído e temperaturas elevadas. Possuem a característica de serem flexíveis, permitindo a conexão com módulos I/O, comunicação e interface com outros dispositivos, (MORAES E CASTRUCCI, 2001).

A informação lida precisa ser transmitida aos vários níveis do processo produtivo e tal transmissão deve ser estabelecida através de meio físico ou eletromagnético obedecendo as regras de envio de mensagens, ou protocolos de comunicação industrial. Os equipamentos de uma planta podem ser de diferentes fabricantes, que utilizam tecnologias diversas na construção de cada instrumento. Para estabelecer uma interligação entre todos os dispositivos, é necessário que estes utilizem os protocolos de comunicação industrial, (MORAES E CASTRUCCI, 2001).

Além de estabelecer a comunicação entre os dispositivos, se faz necessária a existência de uma interface que realize o cruzamento das informações para a geração de gráficos e relatórios e assim permitir que se tenha uma visão geral da produção e também o controle da mesma, (MORAES E CASTRUCCI, 2001).

O presente projeto apresenta uma solução em supervisão e controle em ambientes industriais de pequeno porte. Apresentando, inicialmente, o gerenciamento de duas variáveis de processo, ou seja, a temperatura interna de um túnel de resfriamento com esteiras e a velocidade do rotor do ventilador que causa a alteração da temperatura. O resfriamento é destinado a evitar a sobrecarga de calor dos objetos posicionados sobre uma esteira que se encontra abaixo do ventilador.

1.1. Justificativa

O ambiente industrial é composto por equipamentos de diversos fabricantes, e alguns deles ainda possuem métodos de comunicação proprietários, dificultando e onerando o processo de fabricação, visto que exige mão de obra especificamente qualificada para realizar a instalação e manutenção dos dispositivos da planta.

A sociedade irá se beneficiar da solução proposta por este trabalho, pois nele é mostrado a possibilidade de realizar a comunicação Modbus TCP/IP, através de dispositivos open source, como o sistema supervisório ScadaBR e a plataforma de desenvolvimento Beaglebone Black. E terá base para realizar, de modo facilitado e livre, o gerenciamento de variáveis de processo, tais como a leitura da temperatura

de um ambiente e o controle do rotor de um ventilado, auxiliando assim, a evolução das técnicas de produção industrial.

A relevância para a comunidade científica diz respeito a disponibilização de um material que mostra como a comunicação entre os dispositivos foi realizada, permitindo assim, que o conhecimento seja expandido, e as soluções sejam melhoradas.

1.2. Objetivos

A fim de aplicar os conhecimentos adquiridos ao longo da graduação, este projeto se propõe a oferecer uma solução *open source* de gerenciamento das informações de uma planta industrial, ou seja, sua estrutura é composta por softwares, hardwares e protocolos de comunicação livres de cobrança de licença.

O projeto proposto teve como objetivo o estudo dos processos industriais, a aquisição de conhecimento nas áreas de metrologia e instrumentação dos equipamentos de campo, redes de comunicação e protocolos industriais.

A busca pela compreensão de uma planta industrial e os meios necessários para desenvolver um processo de fabricação automático, juntamente com a escolha dos equipamentos, sensores, controladores, atuadores.

O desenvolvimento do projeto foi composto pela realização de testes, desenvolvimento de códigos e scripts para comunicação entre o mestre e o escravo da rede. O entendimento do sistema supervisor ScadaBR e a criação de uma interface IHM que possibilitou a leitura do sensor de temperatura e a atuação no rotor do ventilador.

1.3. Estrutura do Projeto

A seguir serão abordados a motivação necessária para a realização do estudo em questão e os desejáveis objetivos a serem alcançados. Posteriormente o

capítulo 2 menciona a história da automação industrial e os fatos que a levaram a ganhar tanta importância no cenário capitalista contemporâneo.

O capítulo 3 especifica as redes de comunicação industrial e suas características. O capítulo 4 descreve os elementos de um sistema de controle automático, e dá ênfase aos que serão utilizados no desenvolvimento deste projeto. O capítulo 5 trata sobre os sistemas supervisórios e em especial do ScadaBR.

O sexto capítulo dá início ao desenvolvimento do sistema, mostra como foi realizada a comunicação entre os diferentes dispositivos e a criação das telas do sistema supervisório. Posteriormente, no sétimo capítulo, será mostrada a conclusão a que este projeto chegou e apontará as possíveis ações a serem realizadas futuramente.

2. REVISÃO BIBLIOGRÁFICA

Os itens a seguir dizem respeito à revisão bibliográfica dos elementos utilizados neste trabalho. Primeiramente será conceituado a automação industrial, juntamente com os fatos históricos que marcaram surgimento e evolução das técnicas de produção. Posteriormente serão referenciados os componentes que integram um sistema de controle automático. E, mais tarde, será abordada a rede industrial como o elemento responsável por interligar todos os equipamentos do processo.

2.1. A Automação Industrial

A automação industrial realiza a implementação de equipamentos, redes de comunicação e softwares, incluindo sistemas supervisórios e interfaces homem-máquina a um determinado processo industrial. E possui a finalidade de aumentar a supervisão da atividade industrial, analisar e solucionar rapidamente problemas que porventura venham a ocorrer e maximizar os lucros com a redução do consumo de energia e uso consciente das matérias primas.

É importante também para estabelecer melhores condições de segurança dos equipamentos, dos operários e das informações trafegadas, e ainda reduzir o esforço humano sobre o processo.

As funções de um sistema de automação podem ser categorizadas em vários níveis hierárquicos diferentes, como pode ser visto na Figura 1.



Figura 1 - Pirâmide Hierárquica da Automação Industrial.

Fonte: Moraes e Castrucci, 2001 (Adaptado).

Na base encontram-se os controladores distribuídos, dispositivos de campo, sensores, máquinas e componentes da planta industrial. No segundo nível estão os equipamentos que executam o controle automático e centralizado do sistema (CLP, CNC, PC).

O terceiro nível é relacionado com a supervisão dos processos juntamente com a interface em um banco de dados com todas as informações relativas ao processo. No nível quatro encontram-se as ações de planejamento da produção a fim de auxiliar no controle dos processos e na logística da matéria-prima.

O quinto e último nível é responsável por gerenciar as informações e relacioná-las com os recursos da empresa e assim auxiliar na tomada de decisões importantes para a empresa.

A seguir serão abordados conceitos importantes para o entendimento da importância e necessidade da automação industrial na atualidade, a partir de seu surgimento, sua configuração e padronização, seus elementos básicos de controle e sensoriamento.

Segundo Groover (2010, p.57), a automação industrial possui três grandes marcos históricos no contexto de seu surgimento, os quais foram a invenção e desenvolvimento de três elementos básicos que constituem um sistema automatizado.

O primeiro marco diz respeito ao surgimento da energia, inicialmente gerada por dispositivos mecânicos que foram criados e aperfeiçoados ao longo dos anos, desde a invenção da roda (3200 a.C.) até a construção de máquinas a vapor (1765). E posteriormente transmitida para alimentar diferentes sistemas, tais como teares (1733) e locomotivas (1803). Portanto o primeiro grande marco do surgimento da automação foi a geração e transmissão da energia.

O segundo tem relação com o advento do controle de uma máquina, ou sistema, através da retroalimentação, ou seja, correção do comportamento de um equipamento, a partir de dados que ele mesmo fornece. James Watt e Matthew Boulton desenvolveram um dos primeiros dispositivos com essa tecnologia, o chamado controlador centrífugo (1785).

O terceiro, e não menos importante marco histórico foi a invenção da primeira máquina programável, o tear de Jacquard, em torno de 1800. O tear tinha o objetivo de produzir tecidos de maneira automática, substituindo as operações manuais e monótonas dos tecelões.

Na Figura 2 abaixo é possível visualizar, da esquerda para a direita: a) cartões perfurados montados no tear, b) tear montado com ganchos e cordas neste caso 4800 fios, com quatro repetições visíveis do padrão ao longo da largura, c) tecendo o tecido, d) a seda final, e) um tecido com diferentes fios de trama coloridos em uma corrida de tear com uma cadeia de cartões.

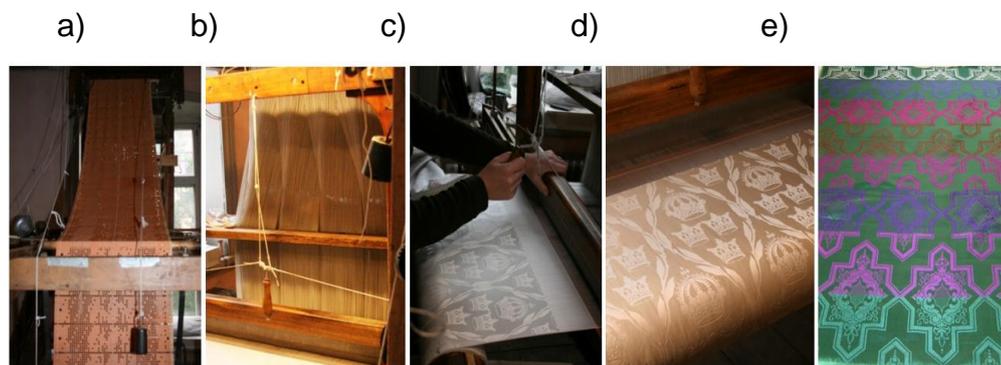


Figura 2 - O processo de tecelagem usando um tear jacquard.

Fonte: FERNAEUS, JONSSON, THOLANDER, 2012.

A evolução dos elementos que constituíram os marcos da automação industrial aconteceu graças ao avanço das pesquisas das áreas de eletrônica (como as válvulas, os transistores, discos rígidos, memórias) e informática (linguagens de programação, sistemas operacionais e inteligência artificial).

Segundo Rosário (2005, p.30), a indústria no Brasil teve seu período de ascensão no início do século XX, impulsionada pelos grandes investimentos de empresários e subsídios governamentais. Na primeira metade do século, houve a instalação da Companhia Siderúrgica Nacional (1947) e foi instituída uma das mais promissoras empresas estatais, a PETROBRAS (1953). Porém, após o período de Ditadura Militar, o país sofreu uma forte recessão juntamente com uma alta inflação prejudicando o desempenho da indústria nacional.

Após a década de 90, o Brasil conseguiu uma maior estabilidade econômica e aos poucos pôde presenciar um retorno da modernização industrial. Nos dias atuais, o parque industrial do país, se encontra desatualizado em relação aos países desenvolvidos, e enfrenta problemas de concorrência internacional, sendo forçado a modernizar seus equipamentos e máquinas a fim de conseguir produzir com qualidade, eficiência e altas margens de lucros.

Moraes e Castrucci (2001, p.358) apontam a criação dos equipamentos pneumáticos (1940) como possibilidade para a redução do tempo de acionamento das válvulas, uma vez que o controle seria feito de forma automatizada. A partir de

então, os equipamentos de supervisão e monitoramento não necessariamente deveriam estar geograficamente próximos aos atuadores e sensores, eles poderiam estar confinados em uma central de controle, ou distribuídos pela fábrica.

A seguir será mostrada a rede de comunicação que compõe um ambiente industrial, quais suas características e configurações necessárias para realizar a interligação entre os equipamentos, capaz de monitorar, gerenciar, medir e configurar o funcionamento dos sensores e atuadores dentro de uma indústria.

2.2. Redes de Comunicação

A automação industrial é subordinada à existência de uma rede onde todos os equipamentos possam estar interligados a fim de estabelecer uma comunicação e trocar informações. E segundo Moraes e Castrucci (2001, p.95), é importante a existência de normas técnicas aceitas pelos dispositivos, a fim de padronizar a comunicação entre eles.

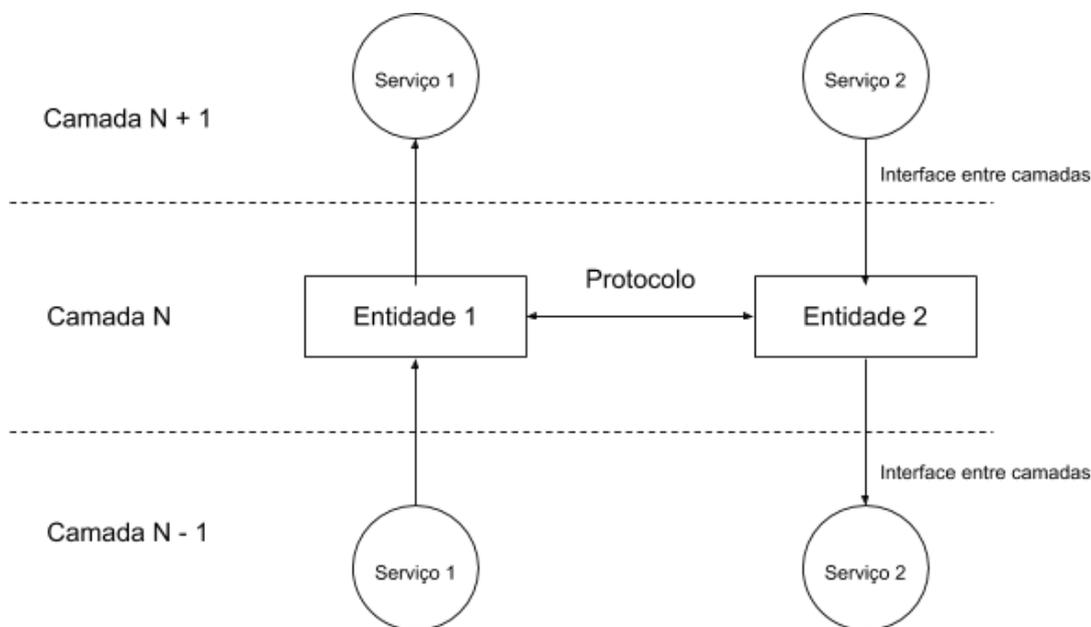


Figura 3 - Elementos básicos do modelo de referência OSI.

Fonte: Kurose e Ross, 2010 (Adaptado).

A Figura 3 anterior ilustra alguns conceitos de redes de comunicação, camadas, entidades e protocolos propostos pela Organização Internacional de

Normalização, ISO (*International Standards Organization*). Ela foi responsável por criar o modelo de referência para a conexão universal de equipamentos digitais, ou do inglês, OSI (*Open Systems Interconnection Reference Model*).

As camadas de rede foram propostas para classificar, de forma hierárquica, as ações que ocorrem durante a comunicação. As entidades são os dispositivos que irão provocar as ações, ou seja, são os softwares, os aplicativos executados em cada equipamento. Os protocolos, por sua vez, são as normas que padronizam e garantem que a comunicação ocorra, fazendo com que os dispositivos “conversem na mesma língua”. O modelo OSI propôs sete níveis hierárquicos, ou camadas, as quais estão ilustradas na Figura 4 abaixo:

7	Aplicação
6	Apresentação
5	Sessão
4	Transporte
3	Rede
2	Enlace
1	Física

Figura 4 - Representação das camadas propostas no modelo OSI.

Fonte: Kurose e Ross, 2010 (Adaptado).

A camada mais inferior, chamada de camada Física, define as especificações do meio de transmissão em que a informação deverá trafegar. As especificações devem seguir uma padronização específica para as conexões entre equipamentos em ambientes industriais. Os principais padrões utilizados são o EIA RS232 Standard e o Padrão 20 mA.

A primeira, sendo utilizada para transmissões mais lentas, com poucos equipamentos interligados. A segunda, envia todos os sinais por corrente elétrica de 4 a 20 mA, e é mais vantajosa por sua imunidade a ruídos eletromagnéticos.

A segunda camada, Enlace, encontra e corrige possíveis erros da camada física e controla o fluxo de dados e estabelece um protocolo de comunicação entre as entidades interligadas. A camada de Rede é responsável por realizar o roteamento, ou seja, define a rota em que a informação irá trafegar de uma entidade a outra, se necessário, fragmenta a informação no destinatário antes de enviá-la, e a remonta no receptor.

A camada de Transporte é responsável pela transferência de dados entre dois equipamentos, realiza o controle de fluxo e detecta erros de comunicação entre as duas entidades comunicantes. A camada Sessão, possibilita a comunicação entre dois dispositivos, define como será feita a transmissão da informação, selecionando os dados a serem transmitidos.

A camada de Apresentação faz a tradução dos dados recebidos pela camada superior, em um formato padronizado conhecido pelo protocolo utilizado. Finalmente, a camada de Aplicação diz respeito às aplicações, ou softwares executados nos equipamentos, ou seja, todo o processo específico do sistema executado, tanto na máquina que envia dados, como também na que recebe a informação.

2.2.1. Topologia das Redes

Segundo Moraes e Castrucci (2001, p.101), a forma com que os equipamentos de uma rede estão interligados determina sua flexibilidade, a velocidade com que a informação é trafegada e a segurança a qual os dados estão submetidos. Caso o projeto tenha a necessidade de possuir um meio partilhado, é possível realizar duas configurações principais, em forma de barramento e na forma de anel.

A rede em barra, mostrada na Figura 5, é caracterizada por permitir que apenas uma máquina faça solicitações por vez, ou seja, a cada instante, apenas uma pode ser o mestre (*master*).

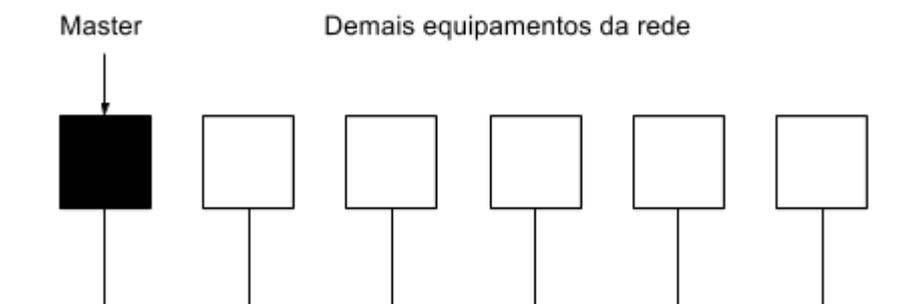


Figura 5 - Rede de comunicação com a topologia em barra.

Fonte: Ross, 2008 (Adaptado).

De acordo com a Figura 6 abaixo, é possível entender a rede em anel como sendo uma topologia em que a informação a ser transmitida deve passar pelos demais dispositivos conectados.

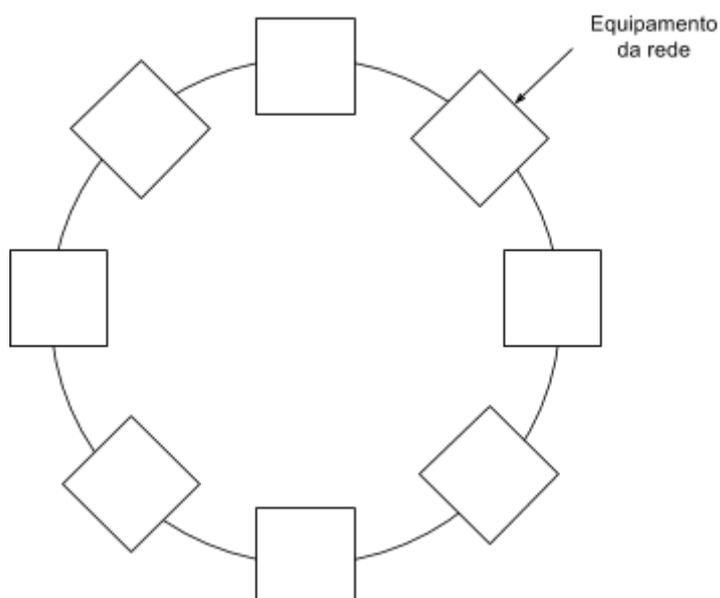


Figura 6 - Rede de comunicação com a topologia em anel.

Fonte: Ross, 2008 (Adaptado).

Caso a configuração do projeto demande uma rede cuja conexão deve ser ponto a ponto, ou seja, a comunicação deve ocorrer sempre em pares de máquinas, é possível utilizar as topologias em estrela, grafo ou árvore (e suas variações).

A topologia em estrela, como mostrada na Figura 7, permite que os equipamentos se comuniquem entre si, através de um ponto central, ou uma máquina que exerce o papel de supervisão do sistema.

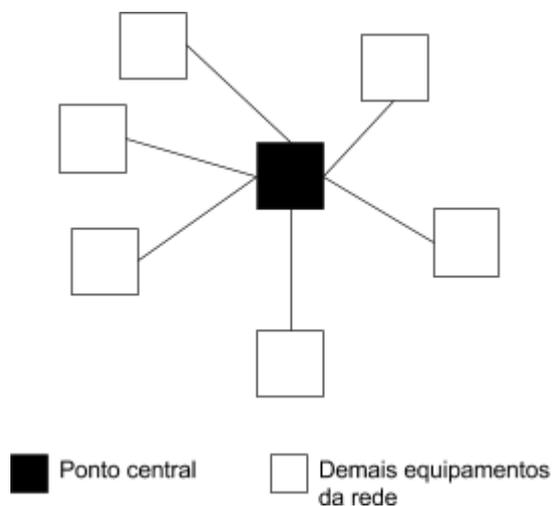


Figura 7 - Rede de comunicação com a topologia em estrela.

Fonte: Ross, 2008 (Adaptado).

A rede em grafo permite que todos os dispositivos estejam conectados diretamente entre si.

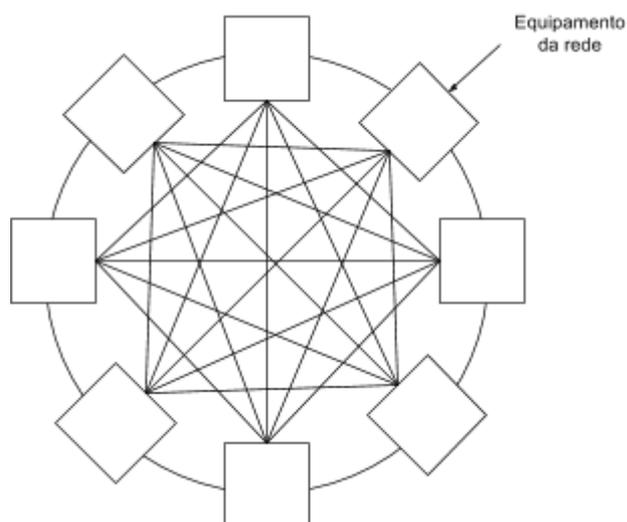


Figura 8 - Rede de Comunicação com a topologia em grafo.

Fonte: Ross, 2008 (Adaptado).

A topologia em árvore é utilizada em sistemas onde existe uma hierarquia de computadores para controle, sincronismo e registro dos processos monitorados.

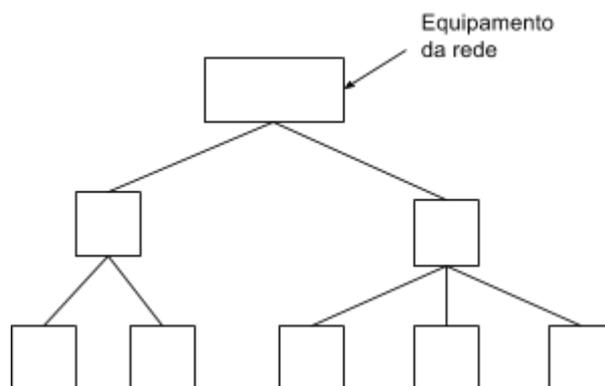


Figura 9 - Rede de comunicação com a topologia em árvore.

Fonte: Ross, 2008 (Adaptado).

2.2.2. Protocolos de comunicação industrial

A rede de comunicação industrial garante que equipamentos trabalhem em modo de cooperação a fim de aumentar a produtividade e supervisionar um determinado processo. Os elementos de um sistema de controle automático, que serão vistos na próxima sessão deste projeto, precisam estar interconectados para enviar e receber dados de forma segura e eficiente. Segundo Bertelli (2015, p.22), as redes de comunicação industrial podem ser divididas em três grupos, o Sensorbus, o Devicebus e o Fieldbus.

As redes de sensores ou Sensorbus atuam na conexão de sensores, atuadores e controladores. Já a rede de dispositivo ou Devicebus oferece serviços de comunicação para dispositivos que conseguem realizar múltiplas funções e comunicar informações sobre diagnósticos e funções a realizar e já realizadas. Os exemplos mais comuns de redes Devicebus são: Profibus-DP, DeviceNet, Interbus-S, Modbus e Foundation Fieldbus HSE.

A rede de instrumentação ou Fieldbus atua com a finalidade de receber e exportar dados de sensores e atuadores para dispositivos de campo, controladores

programáveis e sistemas supervisórios. São exemplos de redes Fieldbus: Foundation Fieldbus H1, HART, WorldFIP, Profibus-PA.

O que afirma a eficácia da comunicação são os padrões de comunicação, ou protocolos de comunicação industrial, que são caracterizados por serem um conjunto de regras em que cada um dos dispositivos comunicantes deve seguir. Os equipamentos industriais podem utilizar os seguintes protocolos mais comuns do setor industrial: Modbus, HART, CAN, Profibus, Foundation Fieldbus, ASI, OPC.

2.2.2.1. Modbus

O Modbus pode ser definido como um protocolo de comunicação industrial da camada de aplicação, a sétima camada do modelo OSI, e é caracterizado por providenciar a comunicação cliente/servidor entre dois dispositivos interligados através de um meio físico. Foi criado em 1979 pela Modicon e a partir de 2004 se tornou livre de taxas de licenciamento pela Schneider Electric, atual controladora da Modicon (SILVEIRA, 2017).

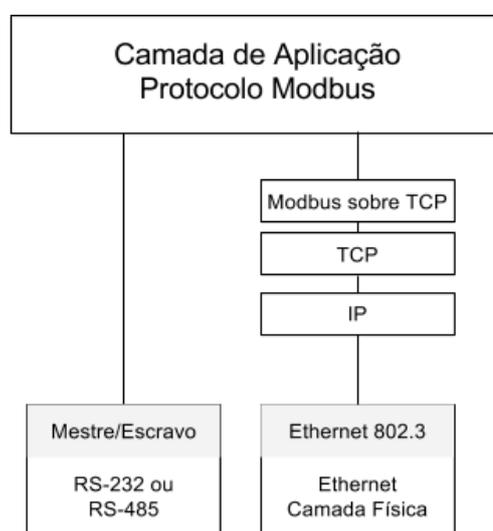


Figura 10 - Pilha de comunicação Modbus

Fonte: MODBUS, 2015 (Adaptado).

O Modbus pode ser utilizado em diferentes padrões de meio físico, sendo mais comumente a utilização dos meios: RS-232, RS-485 e Ethernet TCP/IP (Modbus TCP), conforme indica a Figura 10 acima. A velocidade de comunicação, o

comprimento máximo da rede e a quantidade de dispositivos a serem conectados na rede, são características inerentes ao meio físico e interferem diretamente na eficácia da aplicação do protocolo Modbus.

Segundo a especificação oficial do protocolo de aplicação Modbus, (MODBUS, 2005) qualquer dispositivo industrial pode usar o protocolo para iniciar uma comunicação remota, seja ele um PLC, ou um painel de controle, ou um dispositivo de E/S, entre outros. A Figura 11 a seguir mostra um exemplo de arquitetura de rede utilizando Modbus.

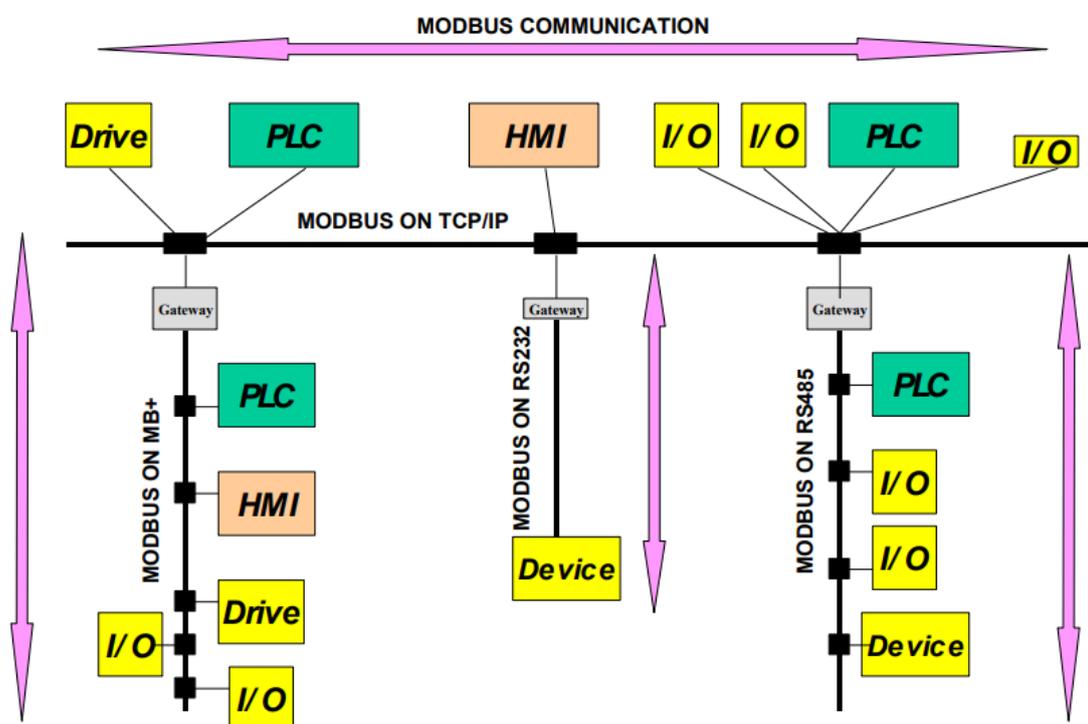


Figura 11 - Arquitetura de rede utilizando Modbus -

Fonte: MODBUS, 2005.

A comunicação entre o mestre e escravo em uma rede Modbus obedecem a um padrão de mensagens baseadas em PDU, do inglês *Protocol Data Unit* ou Unidade de Dados, que é caracterizado como um bloco de dados transmitido entre dois dispositivos em uma mesma conexão. Durante a comunicação, um PDU pode ser de três tipos diferentes, o PDU de requisição, o PDU de resposta e o PDU de resposta com exceção.

O PDU de requisição possui o código e os dados da função solicitada. O PDU de resposta possui o código da função e os dados da resposta. O PDU de resposta com exceção possui o código da função, e o código que especifica a exceção. A Tabela 1 abaixo, mostra os códigos de função utilizados na troca de mensagens entre dispositivos em uma rede Modbus.

Tabela 1 – Exemplos dos Códigos de Função utilizados na comunicação Modbus.

Fonte: MODBUS, 215 (Adaptado).

Códigos de Função	Descrição
1	Read Coils
2	Read Discrete Inputs
3	Read Holding Register
4	Read Input Register
5	Write Single Coil
6	Write Single Register
7	Read Exception status
8	Diagnostic
11	Get Com Event counter
12	Get Com Event Log
15	Write Multiple Coils
16	Write Multiple Register
17	Report Server ID
20	Read File record
21	Write File record
22	Mask Write Register
23	Read/Write Multiple Registers
24	Read FIFO queue
43	Encapsulated Interface Transport

O pacote de dados a ser enviado durante a comunicação pode receber alguns campos adicionais, dependendo do barramento utilizado. O conjunto completo, incluindo a PDU e os demais campos adicionais de cabeçalho, são chamados ADU (Application Data Unit), conforme indica a Figura 12 abaixo.

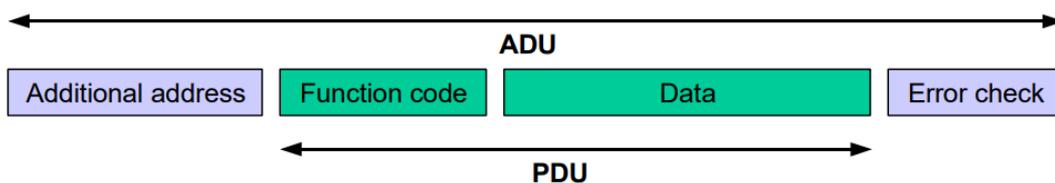


Figura 12 - Estrutura do frame Modbus

Fonte: MODBUS, 2005.

Segundo Freitas (2014), o padrão de meio físico RS-232 (Recommendad Stantart-232) ou EIA-232 (Eletronic Industries Alliance-232) é utilizado quando há a necessidade de estabelecer uma comunicação direta entre dois dispositivos, pois só admite a interface ponto a ponto entre um par de equipamentos, ou seja, um mestre e um escravo. A velocidade de comunicação é próxima de 115Kbps e a distância máxima permitida entre os equipamentos é de 30m.



Figura 13 - Exemplo de cabo para comunicação RS-232.

Fonte: Instructables, 2017.

O padrão RS-485 ou EIA-485 admite uma velocidade de comunicação bem maior que o RS-232, que pode ir de 12Mbps até 50Mbps, a distância máxima entre equipamentos pode ser de até 1200m e podem ser conectados até 32 dispositivos no barramento da rede.

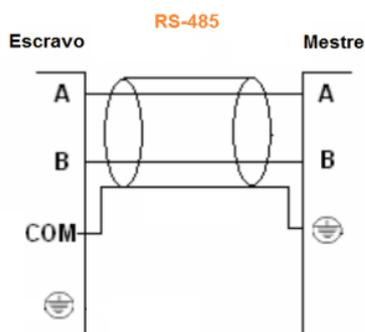


Figura 14 - Exemplo de uma ligação RS-485 entre dispositivos.

Fonte: FREITAS, 2017.

O Modbus TCP é uma vertente do protocolo Modbus baseado no protocolo TCP/IP (Transmission Control Protocol / Internet Protocol). A velocidade de comunicação pode variar de 100Mbps até 10Gbps e a distância entre equipamentos pode variar de 100m até 200m. O mestre Modbus TCP estabelece uma comunicação TCP com o escravo através da porta 502, com a finalidade de enviar requisições e receber as respostas correspondentes.

Ele utiliza a configuração do TCP para comunicação e adiciona um cabeçalho específico, chamado MBAP Header. O cabeçalho possui 7 bytes, e é composto pelos seguintes campos:

- Transaction identifier que identifica a resposta para a transação (2 bytes).
- Protocol identifier indica se está utilizando o Modbus (2 bytes).
- Length determina a contagem de todos os próximos bytes (2 bytes).
- Unit identifier estabelece o escravo remoto em uma rede Modbus RTU (1 byte).

A Figura 15 a seguir mostra a estrutura geral de um frame Modbus com o cabeçalho MBAP.

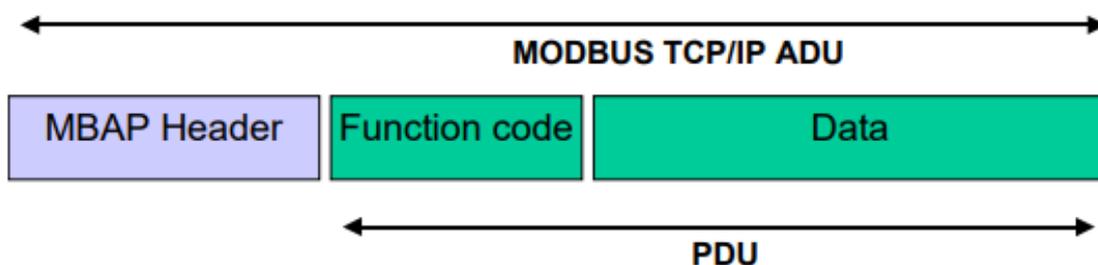


Figura 15 - Frame Modbus com o cabeçalho MBAP

Fonte: MODBUS, 2005.

2.2.3. Sockets de Comunicação

A troca de mensagens entre processos de uma máquina ou aplicação servidor e cliente é realizada por meio de um mecanismo de serviços de transporte chamado Socket. O Socket de comunicação é o responsável por solicitar uma

ligação do cliente para o servidor a fim de iniciar a conexão, conforme indica a Figura 16 a seguir.

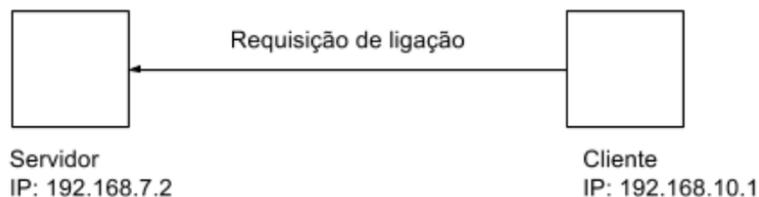


Figura 16–Requisição de ligação.

Fonte: PPLWARE, 2017 (Adaptado).

Os sockets podem ser categorizados em relação à confiabilidade do serviço de entrega dos pacotes transmitidos, sendo, portanto, TCP ou UDP. O socket TCP, que garante a entrega dos dados enviados, pode ser implementado em uma aplicação cliente, por exemplo, que irá solicitar uma ligação ao servidor através do endereço IP do mesmo e a porta disponibilizada por ele (ver Figura 17).

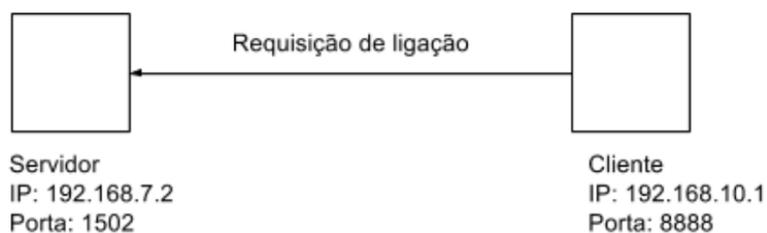


Figura 17–Requisição de ligação com IP e Porta disponíveis.

Fonte: PPLWARE, 2017 (Adaptado).

O servidor deve aceitar a ligação caso não encontre restrições na rede, e assim estabelecer um canal de comunicação com o cliente, respondendo às requisições solicitadas. (PINTO, 2012)

2.3. Elementos de um Sistema de Controle Automático

As redes de comunicação são de extrema importância no ambiente industrial, pois fazem a interligação entre os equipamentos de campo, sensores, atuadores e os sistemas de supervisão, garantindo o desempenho estabelecido para o processo de automação.

Com a finalidade de compreender o comportamento típico de uma planta industrial, serão apresentados a seguir os dispositivos responsáveis pelas principais variáveis monitoradas em um processo, os quais são: controlador, atuador, processo e sensor. A Figura 18 abaixo mostra a relação existente entre eles através de um diagrama simplificado de um sistema de controle automático.

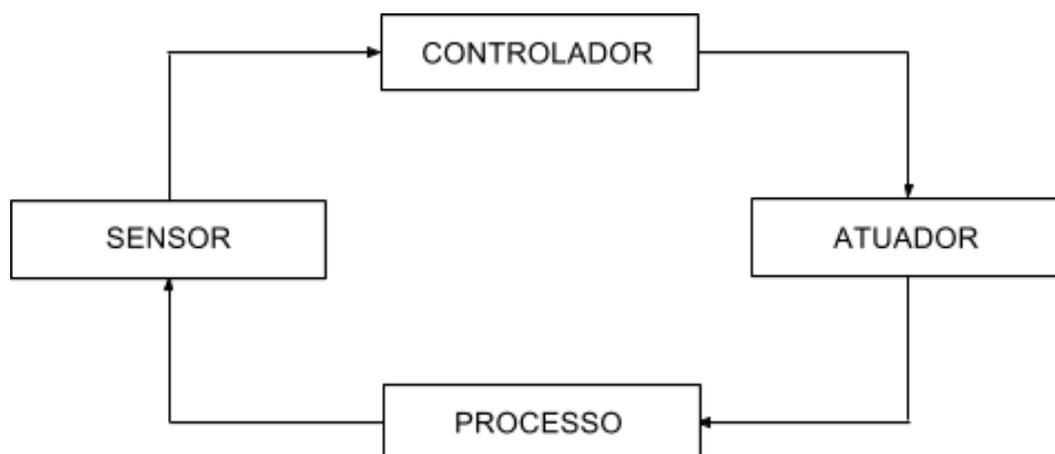


Figura 18 - Diagrama simplificado de um sistema de controle automático.

Fonte: Groover, 2010 (Adaptado).

2.3.1. Sensores e Atuadores

Segundo LAMB (2015, p.79), os sensores são os responsáveis por inserir dados de formatos diversos no sistema. Existem duas categorias distintas para os sensores, os discretos, que indicam o status de ligado ou desligado, e os analógicos, que informam uma variação contínua da leitura de determinada variável, como a temperatura por exemplo.

Os sensores discretos podem ser do tipo chave, relé ou botão que possuem contatos mecânicos em suas partes internas e podem ser sensíveis ao toque sensitivo ou capacitivo. Como exemplo, é possível citar os sensores indutivos, fotoelétricos, capacitivos, ultrassônicos, magnéticos e RFID.



Figura 19 - Exemplos de sensores discretos

Fonte: TERGOLINA, 2015.

Os sensores analógicos, ou transdutores, indicam uma leitura proporcional a uma referência, e para isso existem configurações necessárias, chamadas de *offsets*, para compensar a leitura e colocá-las no referencial desejado, assim como corrigir ocasionais erros do processo. Os sensores de nível, vazão, temperatura e pressão são exemplos de sensores analógicos.



Figura 20 - Exemplos de sensores analógicos.

Fonte: TERGOLINA, 2015.

Os atuadores são dispositivos que agem diretamente nas variáveis de saída do sistema, alterando suas propriedades a fim de seguir o que foi proposto pelas variáveis de entrada. Eles servem para movimentar ferramentas, controlar a posição de uma peça de trabalho, converter energia elétrica em energia de movimento, assim como os motores, cilindros pneumáticos e válvulas solenóides.



Figura 21 - Exemplos de atuadores.

Fonte: BONFIM, 2012.

2.3.2. Sistema de Controle

O sistema de controle de modo geral é um sistema que recebe dados externos, os processa e a partir dos resultados dos seus cálculos diferenciais e algoritmos envia uma saída para algum dispositivo atuador específico. No setor industrial, o sistema a ser controlado recebe o nome de planta, ou planta industrial, os dados externos a serem lidos, ou a entrada do sistema é chamada de variável de controle e à saída denomina-se de variável de controle ou variável de processo (PV).

Segundo Groover (2010 p.63), o sistema de controle tem duas configurações básicas e distintas, podendo se apresentar como sistema de controle a malha fechada (realimentação) e também como sistema de controle a malha aberta.

O sistema de controle por realimentação pode ser visto na Figura 22 e é caracterizado por confrontar o valor de saída com a variável de entrada, e caso a

saída não esteja dentro dos padrões especificados, é aplicada uma medida corretiva a fim de que o sistema seja estável.

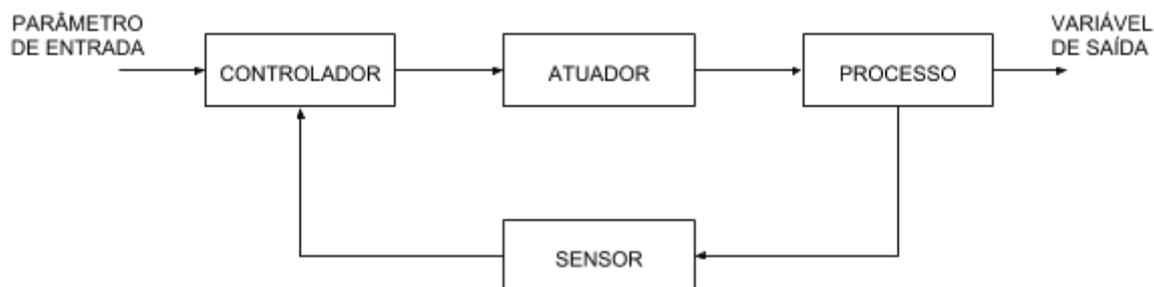


Figura 22 - Sistema de controle por realimentação.

Fonte: Groover, 2010 (Adaptado).

O sistema de controle a malha aberta trabalha sem comparar a variável de saída com a de entrada, utiliza apenas um molde matemático para coordenar a ação do atuador. Abaixo, na Figura 23, é possível observar o diagrama que ilustra o sistema a malha aberta.

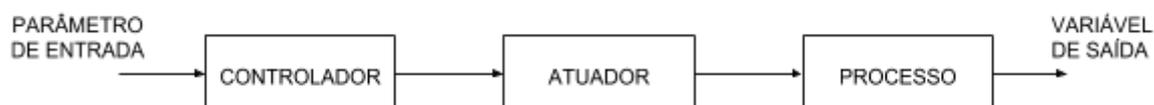


Figura 23 - Sistema de controle a malha aberta.

Fonte: Groover, 2010 (Adaptado).

2.3.3. Dispositivos para Controle Automático

Segundo Bertelli (2015, p.16), os controladores utilizados nos sistemas de automação industrial podem possuir os mais variados tipos de processadores, como ARM Cortex A, i386 ou FPGA e sua utilização depende das necessidades do projeto. Algumas das características dos controladores a serem consideradas no desenvolvimento do sistema são a velocidade de processamento, a quantidade de dispositivos possíveis a serem controlados, a segurança da informação, o custo de aplicação e manutenção.

Os principais controladores para a área da automação são: o Comando Numérico Computadorizado (CNC), o Controlador Lógico Programável (CLP) e os Computadores *single-board*, como a Beaglebone Black por exemplo.

2.3.3.1. Controlador Lógico Programável (CLP)

Segundo GROOVER (2015, P.213) O controlador lógico programável (CLP) é caracterizado por possuir um microcontrolador programável modularizado para gerenciar suas entradas e saídas a fim de realizar o controle automático em sistemas de produção. A Figura 24 a seguir mostra um CLP comercial da marca SIEMENS.



Figura 24 - CLP Siemens SIMATIC S7 – 1200.

Fonte: Siemens, 2017.

Os componentes básicos de um CLP são: o processador, a unidade de memória, a fonte de energia, os módulos de entrada e saída, e seu dispositivo de programação.

O processador garante a execução de diferentes funções lógicas e de sequenciamento a partir da manipulação dos módulos de entrada e saída. A unidade de memória armazena as instruções lógicas e sequenciais e também os arquivos de dados destes programas.

A fonte de alimentação conectada ao CLP converte 115 V ac em $\pm 5V$ cc. Os módulos de entrada e saída permitem receber e enviar informações personalizadas

para cada tipo de aplicação. A seguir é mostrado um diagrama básico da estrutura de um CLP.

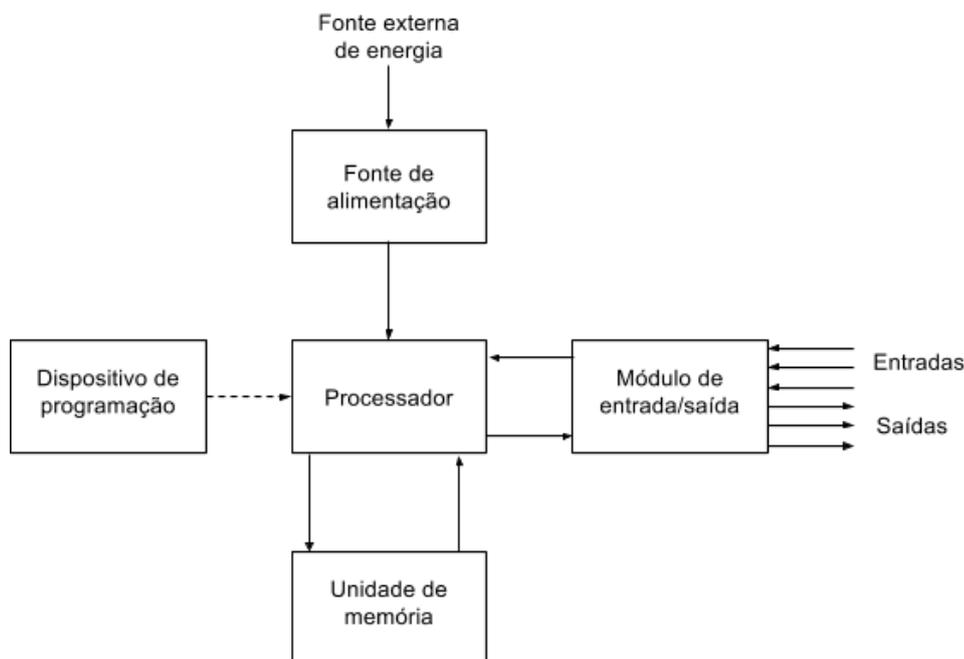


Figura 25 - Componentes do CLP.

Fonte: Siemens, 2017 (Adaptado).

2.4. Sistemas Supervisórios

Os sistemas SCADA (*Supervisory Control and Data Application*) ou Sistemas de Controle Supervisório e Aquisição de Dados podem ser descritos como sistemas que monitoram variáveis do ambiente industrial e controlam equipamentos/atuadores com a finalidade de oferecer dados tanto para o operador, com alarmes e registros de eventos, como para a gerência da empresa, através de gráficos e relatórios.

Um Sistema SCADA é implementado através de uma interface homem-máquina (IHM), que utiliza tecnologias de computação e comunicação a fim de ter visão gráfica geral do processo industrial. A Figura 26 abaixo ilustra a interface de um sistema SCADA.



Figura 26 - Exemplo de interface de um sistema SCADA –

Fonte: SCADA, 2017.

3. DESENVOLVIMENTO

O projeto proposto tem como intenção agregar todos os conhecimentos citados anteriormente, a fim de criar uma solução de resfriamento industrial, através de um sistema de supervisão conectado a uma planta industrial, conforme indica o diagrama de blocos da Figura 27.

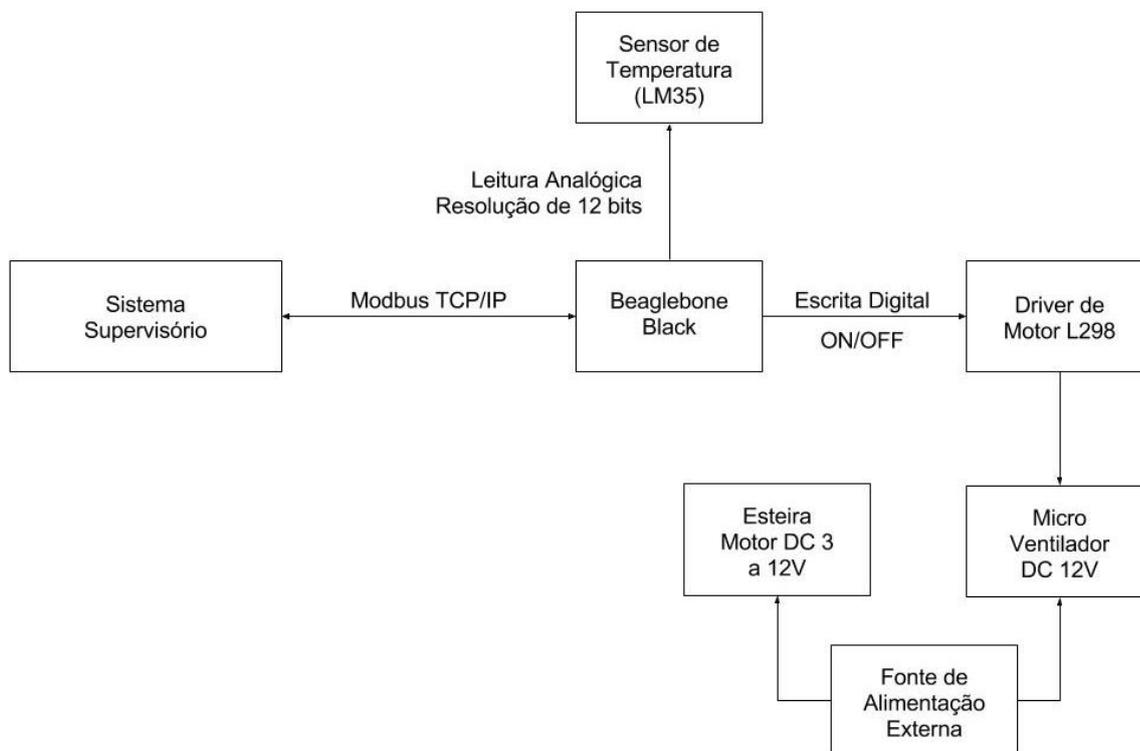


Figura 27 - Interfacemento do sistema de resfriamento.

O processo é responsável por diminuir a sobrecarga de calor de objetos transportados por uma esteira, através do controle de temperatura no tubo de

resfriamento posicionado acima da esteira. O controle é efetivado por meio do acionamento do rotor de um ventilador quando a temperatura atinge um valor determinado.

O ScadaBR é o responsável por realizar a supervisão do sistema, requisitar da planta a leitura do sensor de temperatura, e acionar o ventilador quando esta temperatura atinge um nível determinado. A planta é controlada pela single-board Beaglebone Black, onde estão ligados diretamente o sensor e o driver para o controle dos motores da esteira e do ventilador.

A comunicação entre o sistema supervisor e a planta foi realizada através do protocolo industrial Modbus TCP/IP. E de acordo com a configuração do protocolo, o ScadaBR se comportou como mestre da conexão e a Beaglebone Black como o escravo, respondendo às requisições do mestre sempre quando solicitado.

O projeto de resfriamento deverá ser capaz de manter a temperatura de 25°C no tubo de refrigeração, a Tabela 2 a seguir indica as temperaturas a serem monitoradas pelo sistema de supervisão. Ao atingir a temperatura máxima, ou $T_{High\ Critica}$, o rotor do ventilador é acionado, e quando atingir a temperatura $T_{Low\ Critica}$, o ventilador é desligado.

Tabela 2 - Faixa de temperaturas monitoradas.

Indicação	Temperatura
$T_{High\ Critica}$	Maiores que 28°C
T_{High}	Entre 25°C e 28°C
$T_{Operação}$	25°C
T_{Low}	Menores que 25°C

Os itens seguintes mostram o desenvolvimento do projeto, identificam quais foram os componentes utilizados, e expõem como foi realizada a interconexão entre eles.

3.1. ScadaBR

O ScadaBR é um tipo de Sistema SCADA, porém, ao contrário dos principais sistemas de supervisão existentes no mercado, é *open-source*, ou seja, sua licença é gratuita e seu código é aberto, podendo ser reescrito e adaptado para diferentes fins. (ScadaBR, 2010).

O ScadaBR, como um ambiente baseado em Java, utiliza o Tomcat como Servidor de Aplicação e pode ser acessado através de uma página web comum. O Servidor de Aplicação é um ambiente de execução responsável por auxiliar os Servidores Web a processar alguma página Web que possui um conteúdo, ou uma extensão específica. Ele processa a informação do lado do servidor, fornecendo ao servidor Web uma resposta regular, para que este possa exibir como uma html padrão ao usuário.

Assim como um Sistema SCADA, o ScadaBR precisa de um banco de dados para armazenar as informações dos usuários e histórico da planta industrial. Os SGBDs, ou Sistemas de Gestão de Banco de Dados são os responsáveis por realizar tanto o armazenamento das informações, como também garantir a integridade e a segurança dos dados trafegados. Ele pode ser configurado para executar duas opções de SGBD, o MySQL ou o Derby, sendo que este último é embarcado ao ScadaBR e é de fácil manipulação. Já o MySQL é um SGBD externo, de complexidade mais elevada, sendo necessário configurá-lo antes de instalar o ScadaBR.

O ScadaBR é utilizado para criar uma IHM de acordo com a necessidade do projeto, sendo que o projetista poderá criar diferentes telas, botões, alarmes a partir de uma interface Web. Para que a IHM seja criada, editada e monitorada, é necessário realizar a instalação do software no computador e para acessá-la é preciso que a mesma esteja em configurada em uma rede industrial. A Figura 28 abaixo ilustra a interface de um sistema criado através do ScadaBR.

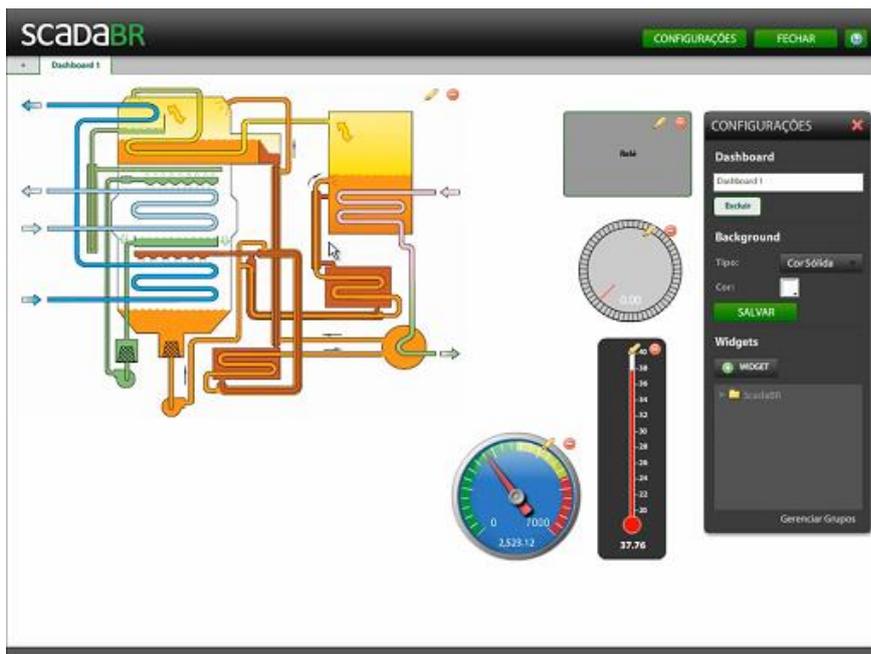


Figura 28 - Interface do ScadaBR para controle de planta industrial.

Fonte: ScadaBR, 2010.

3.2. Beaglebone Black

O controlador baseado em um computador *single-board* é a descrição mais simplificada da plataforma de desenvolvimento de baixo custo, Beaglebone Black (Figura 29). Ela é composta pelo microprocessador Am335x, com base no processador ARM Cortex-A8 da Texas Instruments (COLEY, 2013).

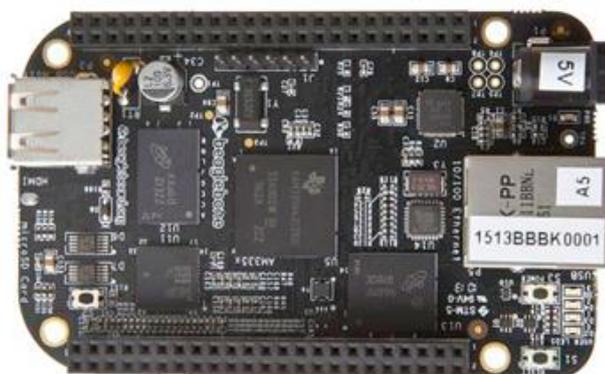


Figura 29 - Beaglebone Black.

Fonte: COLEY, 2013.

O processador possui opções de interface industrial de tempo real usados no modo mestre e escravo: EtherCAT®, PROFINET, EtherNet/IP™, PROFIBUS, POWERLINK, SERCOS III. Ele também suporta os sistemas operacionais de alto nível como o Android, Angstrom, Ubuntu, Debian, ArchLinux, Gentoo, Sabayon, Buildroot, Erlang, Fedora e Cloud9 IDE com Node.js w/ BoneScript library.

A Beaglebone Black foi desenvolvida para atender demandas de automação industrial de pequeno a médio porte, desde os dispositivos de entrada e saída (I/O Devices: Sensores, Drivers de Motores e Gateways) até as IHMs Industriais e CLPs. E pode ser programada em diversas linguagens de programação, como por exemplo: C, C++, Java (Qt Jambi), Python (PyQt, PySide, PythonQt), Ruby (QtRuby).

O diagrama abaixo mostra o detalhamento dos pinos de entrada e saída da Beaglebone Black.

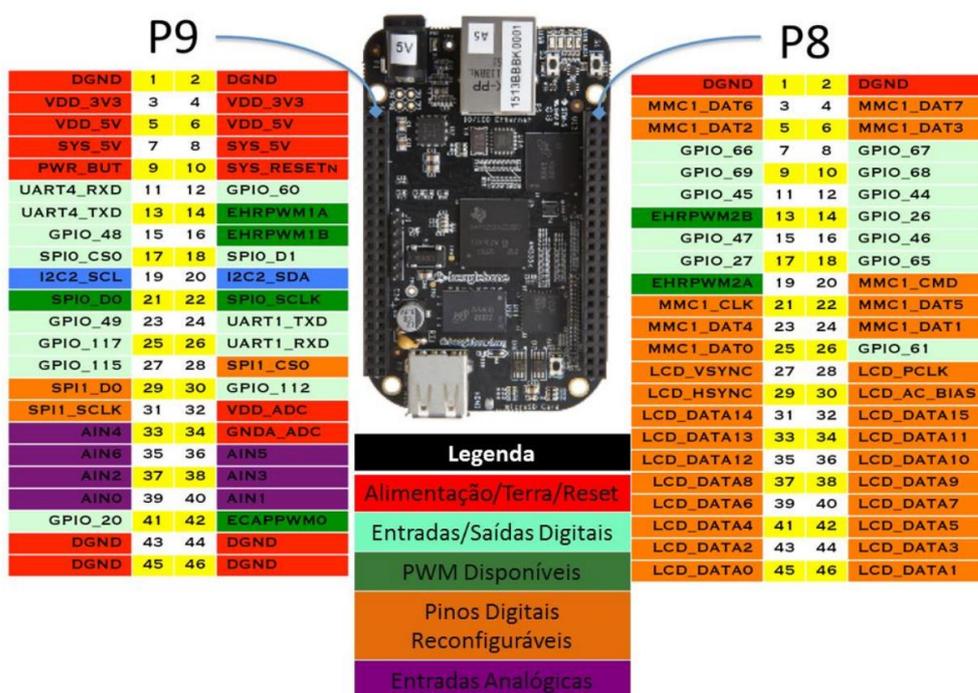


Figura 30 - Detalhamento dos Pinos da BeagleBone Black.

Fonte: COLEY, 2013.

A Tabela 3 a seguir, mostra as principais características da plataforma:

Tabela 3 – Características da Beaglebone Black.

Fonte: COLEY, 2013 (Adaptado).

Característica	Descrição
Processador	Siatra AM3359AZCZ100 1GHz, 2000 MIPS
Engine Gráfica	SGX530 3D, 20M Polígonos/s
Memória SDRAM	512MB, DDR3L, 800MHz Onboard Flash 4GB, MMC embarcado de 8 bits
Fontes	Mini USB, USB ou entrada DC (5V)
Ethernet	10/100 RJ45
Pinos	VCC 5V e 3.3V, 19 GPIO (entrada/saída digital, 3.3V), 7 entradas analógicas (1.8V), 5 PWM, 4 UART, 4 timers, entradas digitais configuráveis, etc
Conector	SD/MMC Micro SD, 3.3V

O desenvolvimento deste projeto aconteceu através da comunicação do computador com a Beaglebone Black. O computador em questão utiliza o Windows 10 como sistema operacional e se conecta à single-board através do PuTTY (software de emulação de terminal de código livre), que por sua vez utiliza o protocolo SSH (Secure SHell), o qual permite o acesso virtual a um servidor através de seu endereço IP. A Figura 31 abaixo mostra a configuração a ser realizada no PuTTY para se conectar ao terminal da Beaglebone.

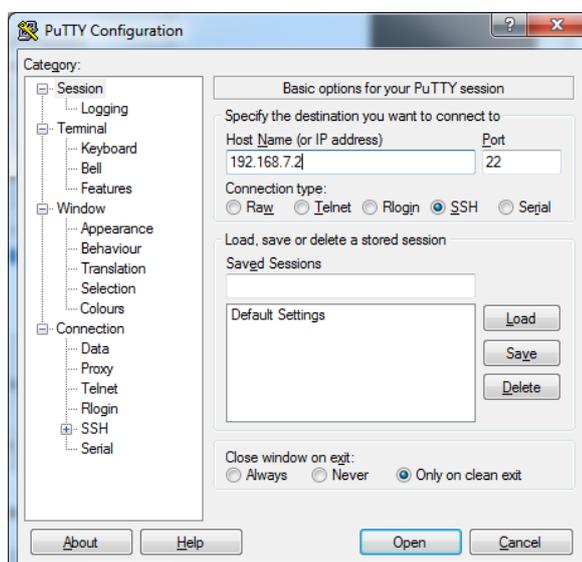


Figura 31 - Configuração do PuTTY para conexão com a Beaglebone Black.

A Tabela 4 a seguir mostra a ligação dos pinos do sensor aos respectivos pinos da Beaglebone Black.

Tabela 4 - Ligação entre o LM 35 e a Beaglebone Black.

LM35		Beaglebone Black	
Pino	Descrição	Pino	Descrição
Vcc	Tensão de alimentação	3V3	Saída de 3.3 V
Vs	Sinal de saída	AIN0	Canal ADC
Vout	Ground	DGND	Ground

3.5. Módulo L298

Os motores DC inclusos no projeto demandam uma quantidade de corrente superior à que as portas da Beaglebone Black conseguem fornecer. Portanto, para realizar o controle dos motores é necessário utilizar um driver que amplifique a corrente de saída da *single-board*. Com a finalidade de resolver este problema, foi utilizado o circuito integrado L298, que utiliza transistores a fim de amplificar a corrente e alterar o sentido de rotação do motor.

O projeto proposto utilizou o módulo L298 (Figura 33), que pode ser caracterizado como um circuito pré-configurado, com resistores, capacitores e entradas que facilitam a montagem e utilização dos motores.

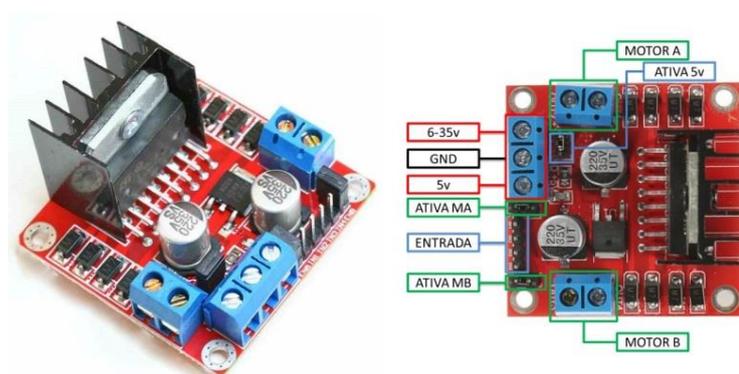


Figura 33 - L298 Módulo Driver de Motor. a) Visão geral. b) Entradas e saídas do módulo.

Fonte: Instructables, 2017.

O módulo em questão possui as características construtivas listadas na Tabela 5.

Tabela 5 - Características construtivas do módulo L298

Fonte: Instructables, 2017 (Adaptado).

Características	Valores de referência
Circuito Integrado	L298
Tensão para os motores	5 a 35V
Corrente máxima para os motores	2A
Potência máxima	25W
Tensão Lógica	3 a 5V
Corrente Lógica	0 a 36mA

A Tabela 6 indica a descrição de cada um dos pinos do módulo.

Tabela 6 - Descrição dos pinos do módulo L298.

Fonte: Instructables, 2017 (Adaptado).

Pino	Descrição
6-35 V	Pino para alimentação do módulo, com tensão de 6 a 35V.
Ativa 5 V	Quando ativo, a placa utilizará o regulador de tensão integrado para fornecer 5 V (na porta 5 V) quando a porta 6-35V estiver sendo alimentada por uma tensão entre 6 e 35V.
5 V	Em casos de não haver fonte de alimentação com mais de 6V podemos alimentar a placa com 5V por esta porta.
Ativa MA	Quando ativo, aciona o motor A com velocidade máxima. Para controlar a velocidade do motor A basta remover o jumper e alimentar o pino com uma tensão entre 0 e 5v, onde 0V é a velocidade mínima (parado) e 5V a velocidade máxima.
Ativa MB	Quando jumpeado aciona o motor B com velocidade máxima. Para controlar a velocidade do motor A basta remover o jumper e alimentar o pino com uma tensão entre 0 e 5 V, onde 0V é a velocidade mínima (parado) e 5V a velocidade máxima.
IN1 e IN2	São utilizados para controlar o sentido do motor A.
IN3 e IN4	São utilizados para controlar o sentido do motor B

A Tabela 7 mostra a ligação dos pinos do módulo aos respectivos pinos da Beaglebone Black.

Tabela 7 - Ligação entre o módulo L298 e a Beaglebone Black.

Módulo L298		Beaglebone Black	
Pino	Descrição	Pino	Descrição
IN1	Ativa o Motor	GPIO68	P8_10
IN2	Ativa o Motor	GPIO69	P8_9

3.6. Micro Ventilador DC

A fim de simular o comportamento do ventilador industrial, foi utilizado um micro ventilador com alimentação de 12V e 0,5 A. A sua ligação à planta acontece através do módulo L298 e é alimentado por uma fonte de tensão externa. A Figura 34 a seguir mostra o modelo utilizado para este projeto.



Figura 34—Micro ventilador DC 12V.

Fonte: Fonte: Instructables, 2017.

O micro ventilador foi conectado a um módulo L298 com alimentação externa de 12V, a Tabela 8 a seguir mostra como aconteceu a ligação.

Tabela 8 - Ligação entre o cooler e o módulo L298

Micro Ventilador		Módulo L298	
Cor	Descrição	Pino	Descrição
Preto	Input A	OUT21	MOTOR A
Vermelho	Input B	OUT22	MOTOR A

3.7. Esteira

A fim de realizar a simulação da esteira industrial, por onde são transportados os objetos que necessitam de resfriamento, foi utilizado um mini motor DC conectado a um sistema de engrenagem que proporciona a redução necessária para realizar o acionamento da esteira montada sobre o chassi ilustrado na Figura 35 a).

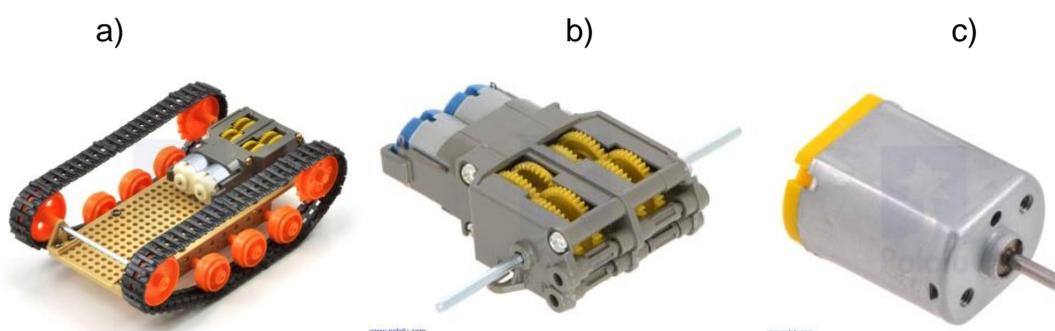


Figura 35 – Componentes para a montagem da esteira. a) Chassi. b) Sistema de engrenagens para acionamento. c) Mini Motor DC.

Fonte: Instructables, 2017.

As características construtivas do motor DC estão descritas na Tabela 9 a seguir.

Tabela 9 - Características construtivas do motor DC

Fonte: Instructables, 2017 (Adaptado).

Características	Valores de referência
Tensão de alimentação	3 a 5 V
Corrente de pico	1.1 A
Corrente de operação	100mA
Rotação nominal	10500 rpm
Torque nominal	0.3 oz.in

A esteira também foi conectada à Beaglebone Black através de um módulo L298, porém, como o motor possui uma faixa de alimentação de 3 a 5V, e corrente de operação de 100Ma, foi possível utilizar a alimentação da própria placa. A Tabela 10 mostra a ligação do mini motor aos respectivos pinos do módulo L298.

Tabela 10 - Ligação entre o mini motor e o módulo L298

Mini Motor		Módulo L298	
Cor	Descrição	Pino	Descrição
Laranja	Input A	OUT21	MOTOR A
Amarelo	Input B	OUT22	MOTOR A

A Tabela 11 a seguir mostra a ligação dos pinos do módulo aos respectivos pinos da Beaglebone Black.

Tabela 11 - Ligação entre o módulo L298 e a Beaglebone Black.

Módulo L298		Beaglebone Black	
Pino	Descrição	Pino	Descrição
IN1	Ativa o Motor	GPIO66	P8_7
IN2	Ativa o Motor	GPIO67	P8_8

A esteira ainda possui um recurso adicional de ser acionada através de um botão externo, do tipo ON/OFF e a partir da leitura deste botão, a Beaglebone ativa o motor e envia as informações ao sistema supervisório ScadaBR.

3.8. Configuração da conexão cliente/servidor Modbus

O projeto proposto possui o modelo cliente/servidor implementado via Socket TCP e utiliza o TCP/IP como protocolo de comunicação. A configuração da Beaglebone Black como servidor foi baseada na biblioteca Libmodbus, e a partir de então foi desenvolvido um programa em C que aguarda o pedido de conexão, via socket, do cliente e responde as requisições do mesmo.

O arquivo denominado main.c foi criado a fim de aplicar a lógica de comunicação entre a Beaglebone e o ScadaBR, realizar a leitura do sensor e o acionamento dos motores. O trecho de código a seguir exibe uma visão geral das bibliotecas incluídas, definições, variáveis e métodos criados.

```

#include "modbus/modbus.h"
#include <unistd.h>
#include <stdio.h>

//BIBLIOTECAS PARA O ACIONAMENTO DOS MOTORES ESTEIRA E COOLER
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>

#define SYSFS_ADC_DIR "/sys/bus/iio/devices/iio:device0"
#define MAX_BUF 64

#define IN1_ESTEIRA "/sys/class/gpio/gpio66/value"
#define IN2_ESTEIRA "/sys/class/gpio/gpio67/value"
#define EN_ESTEIRA "/sys/class/gpio/gpio45/value"

#define IN1_VENT "/sys/class/gpio/gpio68/value"
#define IN2_VENT "/sys/class/gpio/gpio69/value"

```

A biblioteca Libmodbus foi utilizada no programa desenvolvido através da inclusão da biblioteca modbus.h que possibilita a comunicação Modbus entre os dois dispositivos. Os pinos analógicos e digitais tiveram seus endereços salvos em constantes, conforme mostra a Tabela 12 abaixo.

Tabela 12 - Correspondência dos pinos e seus endereços de registro na Beaglebone

Pino	Nome da constante	Endereço do pino
AIN0	SYSFS_ADC_DIR	"/sys/bus/iio/devices/iio:device0"
GPIO_45	EN_ESTEIRA	"/sys/class/gpio/gpio45/value"
GPIO_66	IN1_ESTEIRA	"/sys/class/gpio/gpio66/value"
GPIO_67	IN2_ESTEIRA	"/sys/class/gpio/gpio67/value"
GPIO_68	IN1_VENT	"/sys/class/gpio/gpio68/value"
GPIO_69	IN2_VENT	"/sys/class/gpio/gpio69/value"

O programa criado possui dois métodos principais, onde acontece a configuração da conexão Modbus, a leitura da temperatura e o processamento

necessário para o acionamento dos motores, tanto da esteira, como também do ventilador.

```

/*****
PROGRAMA PRINCIPAL
*****/

int main(int argc, char* argv[])
{...}

void set_ventilador(float temp, bool statusAtual):bool
{...}

void intHandler(int sig)
{...}
void *modbusServerImproved(void *arg)
{...}

```

O `intHandler()` estabelece algumas configurações e inicializa variáveis utilizadas ao longo do código. O `modbusServerImproved()` define algumas ações importantes para a execução do programa, a primeira diz respeito a criação de uma *thread* para executar este método em paralelo com a função `main()`, a segunda é a configuração básica da estrutura da mensagem modbus, e mais especificamente, é onde os endereços das variáveis Modbus são configurados.

```

mb_mapping = modbus_mapping_new_start_address(
0, 2050, // coil
    0, 100, // inputs
    0, 10700, // holding
    0, 100); //input bis

```

A função `main()` é a responsável por executar a *thread* criada na `intHandler()`, e também por configurar as variáveis da leitura do sensor e acionamento dos motores. Ela possui um loop `while(keepRunning)` que é executado a cada 1000ms. A abaixo mostra a manipulação necessária para a leitura da temperatura.

```

/*////////////////////////////////////
      manipulacao para leitura da temperatura
      //////////////////////////////////////*/
snprintf(buf, sizeof(buf), SYSFS_ADC_DIR "/in_voltage%d_raw", ain);

//Abre a conexão com o pino AIN0 e define que ele é somente leitura
fd = open(buf, O_RDONLY);

//Realiza a leitura do canal adc
read(fd, channel, 4);

//Atribui a variavel ADvalue o valor lido no adc
ADvalue = atoi(channel);

/*
  Realiza a conversão do sinal analógico lido, para o sinal de tensão.

  Como a resolução do canal adc é de 12 bits
  e a tensão de referencia do adc é 1.8V

  tensão      | ADC
  1.8V        | 4096
  tensaoConvertida | ADvalue

*/
tensaoConvertida = ADvalue * (1.80f/4096.0f);

/*
  Realiza a conversão da tensão referente a leitura do sinal em graus Celsius

  Quando submetido a temperatura de 25 °C, fornece tensão de 250mV na saída

  tensão      | °C
  250mV       | 25
  tensaoConvertida | grausCelsius

*/
grausCelsius = (25.0f/0.25f)*tensaoConvertida;

//Finaliza a conexão com o canal ADC
close(fd);

//Envia o valor de temperatura para o registrador de entrada do SCADABR
mb_mapping->tab_registers[0] = grausCelsius;

```

A partir da leitura do canal analógico, foi feita a conversão de unidades, a fim de se obter o valor correspondente em graus Celsius. O cálculo da tensão relativa à

leitura analógica foi realizado levando em consideração a resolução de 12bits do canal adc e o seu valor de tensão de referência (1.8V). Também foi considerado a variação da leitura do sensor de acordo com a tensão lida, a qual varia de 10mV/°C.

A manipulação para acionamento do ventilador aconteceu de acordo com o trecho de código reproduzido no trecho de código abaixo, onde foi possível determinar o momento em que o rotor do ventilador é acionado, de acordo com a variação da leitura de temperatura. Os valores configurados são passados ao ScadaBR através da atribuição do novo status do motor ao registrador de entrada.

```

/*////////////////////////////////////
      manipulacao para acionamento do ventilador
////////////////////////////////////
*/

      //abre a conexão com os pinos GPIO68 e GPIO69
in1_ven = open(IN1_VENT, O_WRONLY);
in2_ven = open(IN2_VENT, O_WRONLY);

//realiza a verificação manual da temperatura lida no sensor
//caso satisfaça as relações da função set_ventilador,
//liga ou desliga o rotor do ventilador
if(set_ventilador(grausCelsius)){

      //liga o ventilador
write(in1_ven, "0", 1);
write(in2_ven, "1", 1);

//Envia o status do acionamento para o input register do SCADABR
mb_mapping->tab_input_bits[0] = true;
}

      //desliga o ventilador
write(in1_ven, "0", 1);
write(in2_ven, "0", 1);

//Envia o status do acionamento para o input register do SCADABR
mb_mapping->tab_input_bits[0] = false;
}

      //encerra a conexão com os pinos 68 e 69
close(in1_ven);
close(in2_ven);

```

O trecho abaixo mostra o método que processa a informação do sensor de temperatura lida e retorna um valor booleano a fim de ligar ou desligar o ventilador. A esteira é acionada através da leitura de um botão ligado ao pino GPIO45 da Beaglebone Black, e de acordo com tal leitura, os pinos GPIO66 e GPIO67 são configurados a fim de ligar ou desligar o motor. Posteriormente o status do motor é passado ao ScadaBR através do registrador de entrada.

```
void set_ventilador(float temp, bool statusAtual):bool{
bool liga;

/*
  Esta função realiza o processamento do controle
  do acionamento do ventilador, de acordo com a leitura
  da temperatura.

  STATUS ATUAL | TEMPERATURA | AÇÃO
ON            | <25      | desliga
ON            | >25      | mantem
OFF           | >28      | liga
outros casos | ----     | mantem

*/
if(statusAtual && temp <= 25) liga = false;
else if(statusAtual && temp > 25) liga = statusAtual;
else if(!statusAtual && temp >= 28) liga = true;
else liga = statusAtual;

return liga;
}
```

O trecho de código abaixo se refere a manipulação para acionamento da esteira. A esteira é ligada no momento em que o operador pressiona o botão ON/OFF, que está ligado aos pinos GPIO45. A Beaglebone realiza a leitura do pino, e caso o nível seja alto, a esteira é ligada, e desligada caso contrário.

```

/*//////////////////////////////////////
manipulacao para acionamento da esteira
//////////////////////////////////////*/

//abre a conexão com os pinos GPIO66, GPIO67 e GPIO45
in1_est = open(IN1_ESTEIRA, O_WRONLY);
in2_est = open(IN2_ESTEIRA, O_WRONLY);
en_est = open(EN_ESTEIRA, O_RDONLY);

//caso o botão conectado ao pino 45 for acionado, liga o motor da esteira
if(read(en_est, channel, 2)){

    write(in1_est, "0", 1);
    write(in2_est, "1", 1);

//Envia o status do acionamento para o input register do SCADABR
mb_mapping->tab_input_bits[1] = true;
}else{
    write(in1_est, "0", 1);
    write(in2_est, "0", 1);

    mb_mapping->tab_input_bits[1] = true;
}

//encerra a conexão com os pinos 66, 67 e 45
close(in1_est);
close(in2_est);
close(en_est);

```

Após a criação do código supracitado, foi necessário realizar a sua compilação a fim de que a Beaglebone consiga interpretá-lo e para isso, foi utilizado o recurso Makefile, que gera automaticamente o arquivo compilado, main.o. As configurações básicas do Makefile para este projeto estão mostradas no trecho abaixo.

```
CC=arm-linux-gnueabi-gcc
CFLAGS=-c -std=c99
LDFLAGS=-lmodbus -pthread

SOURCES=main.c
OBJECTS=$(SOURCES:.c=.o)
EXECUTABLE=modbus_slave

all: $(SOURCES) $(EXECUTABLE)

$(EXECUTABLE): $(OBJECTS)
    $(CC) $(OBJECTS) $(LDFLAGS) -o $@

.c.o:
    $(CC) $(CFLAGS) $< -o $@

clean:
    rm -f *.o
    rm -f $(EXECUTABLE)

run:
    scp $(EXECUTABLE) bbb6:~/
    ssh bbb6 './modbus_slave'
```

A configuração realizada para que o ScadaBR se comportasse como o cliente do sistema, foi através da inserção de um Datasource do tipo Modbus IP, com as propriedades mostradas na Figura 36 abaixo.

Propriedades do modbus IP

Nome:

Export ID (XID):

Período de atualização: ▼

Quantificação:

Timeout (ms):

Tentativas:

Apenas quantidades contínuas:

Criar pontos de monitor de escravo:

Máxima contagem de leitura de bits:

Máxima contagem de leitura de registradores:

Máxima contagem de escrita de registradores:

Tipo de transporte: ▼

Host:

Porta:

Encapsulado:

Níveis de alarme de eventos

Exceção de data source: ▼ 📌

Exceção de leitura de data point: ▼ 📌

Exceção de escrita em data point: ▼ 📌

Figura 36 - Configuração das propriedades do Data Source Modbus IP.

Os itens que merecem atenção nesta configuração, são o período de atualização e o timeout, alterados para 500ms. A definição do TCP como protocolo de comunicação (Tipo de Transporte), o endereço IP da Beaglebone Black na rede (192.168.7.2) e a sua respectiva porta para a comunicação, configurada no programa desenvolvido na Beaglebone (1502).

Após configurar o Data Source para a conexão, foi necessário configurar os Data Points, ou seja, os pontos da planta industrial a serem monitorados pelo ScadaBR. A Figura 37 abaixo indica os três pontos criados, sendo que a Esteira e o Ventilador correspondem ao status do acionamento dos motores, e o LM35 indica a temperatura lida pelo sensor.

Data points					
Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)
ESTEIRA	Binário		1	Status de entrada	0
LM35	Numérico		1	Registrador holding	0
VENTILADOR	Binário		2	Status de entrada	1

Figura 37 - Data Points.

4. RESULTADOS

As telas para o monitoramento foram desenvolvidas no ScadaBR e podem ser vistas nas figuras a seguir. Primeiramente é apresentada a tela inicial do sistema supervisório, conforme indica a Figura 38.

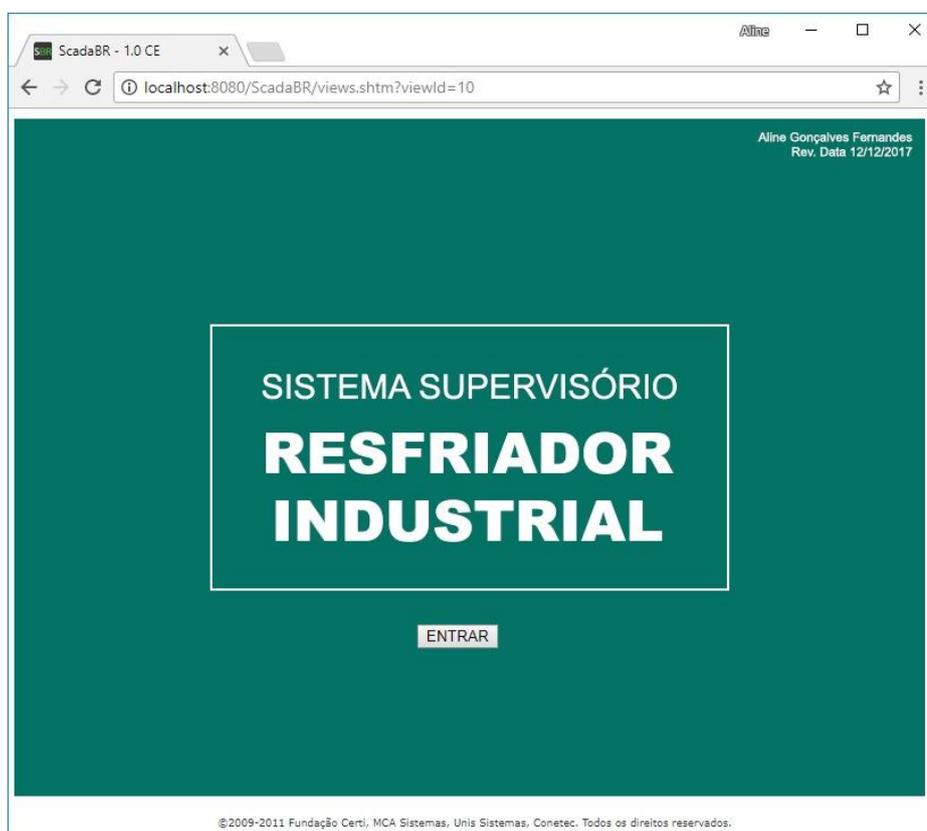


Figura 38 - Tela Inicial do Sistema Supervisório.

A visão geral do sistema pode ser vista através da Figura 39, que indica o diagrama da esteira, sensor de temperatura e o rotor do ventilador. O ventilador possui uma indicação luminosa, que está atrelada ao *Data Point* VENTILADOR,

indicando que o mesmo encontra-se em modo de operação (vermelho) ou em repouso (verde).

O motor da esteira também possui uma indicação luminosa indicando seu status de funcionamento. O sensor de temperatura está vinculado a um componente de Data Point simples, indicando o valor de temperatura lido.

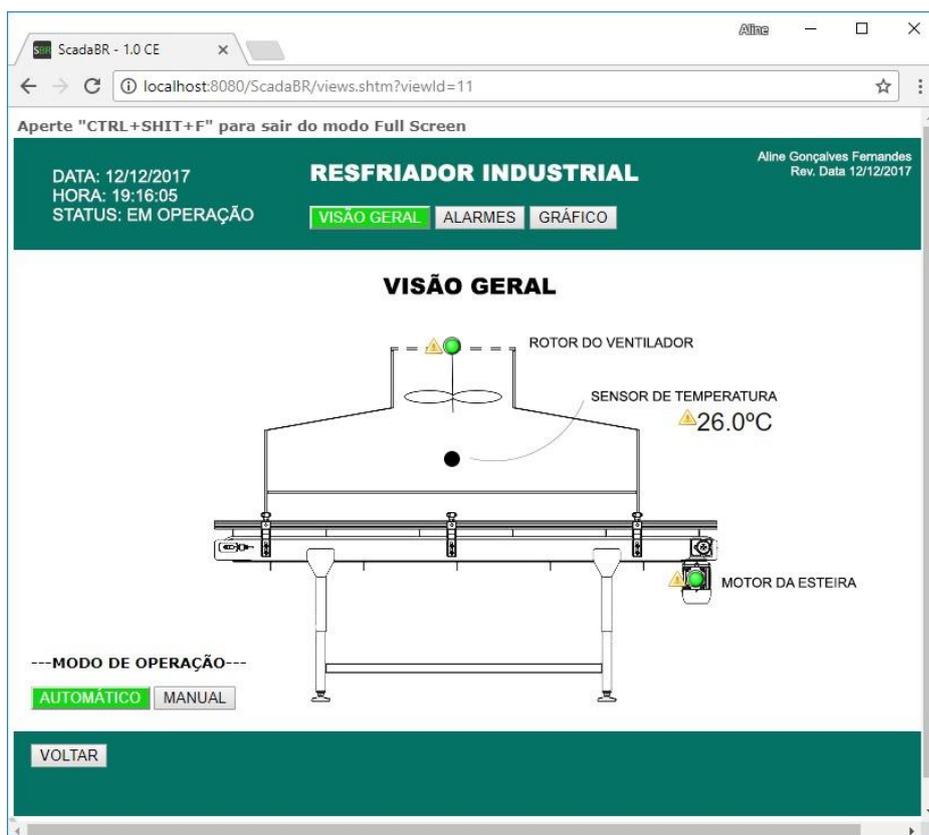


Figura 39 - Visão Geral do Sistema Supervisório.

Os alarmes definidos para o projeto podem ser vistos na Figura 40 e contemplam o monitoramento da operação dos dois motores em conjunto com a leitura da temperatura.

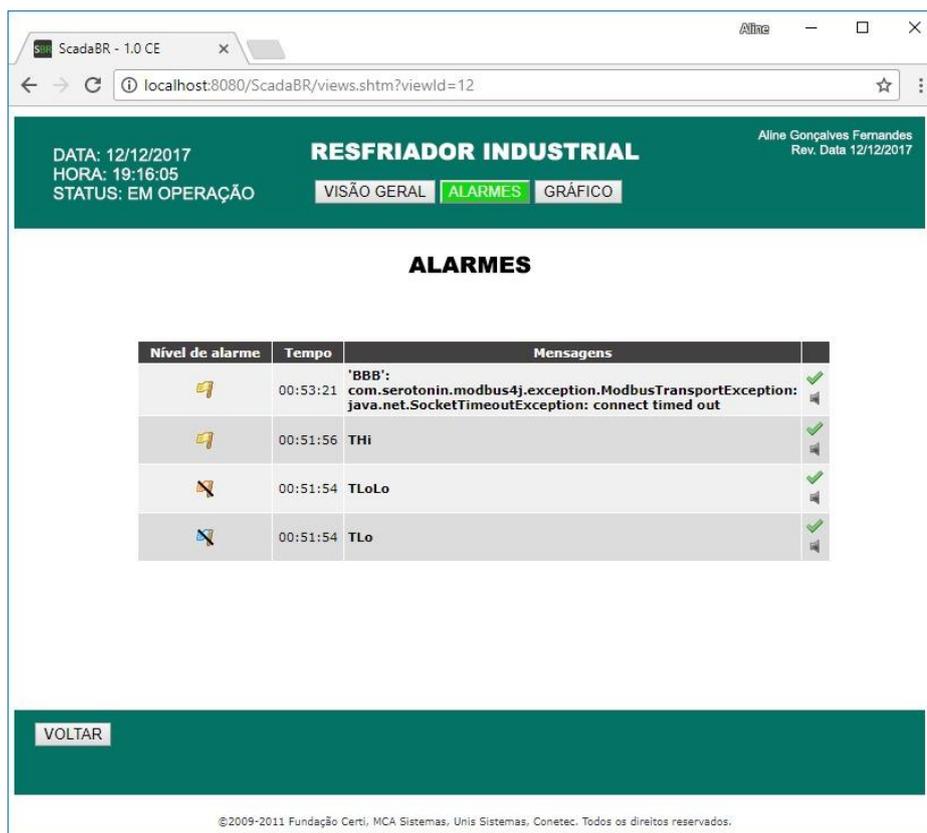


Figura 40 - Alarmes configurados para o Sistema Supervisório.

O alarme para os motores possui nível de informação e é gerado na mudança de estado (ON/OFF). O alarme para a temperatura pode ser a nível de informação, crítico ou urgente, dependendo da faixa em que se encontra a temperatura. A Tabela 13 a seguir, mostra a faixa das temperaturas que foram monitoradas neste projeto.

Tabela 13 - Faixa de temperaturas monitoradas.

Indicação	Temperatura	Nível de alarme
Hih	Maiores que 28°C	Urgente
Hi	Entre 25°C e 28°C	Crítico
Operação	25°C	Informação
Lo	Menores que 25°C	Informação

A tela a seguir, indicada pela Figura 41, indica a visualização gráfica do sistema, informando o momento em que os motores foram ligados, e a influência da ventilação na redução da temperatura dos objetos que trafegam pela esteira.

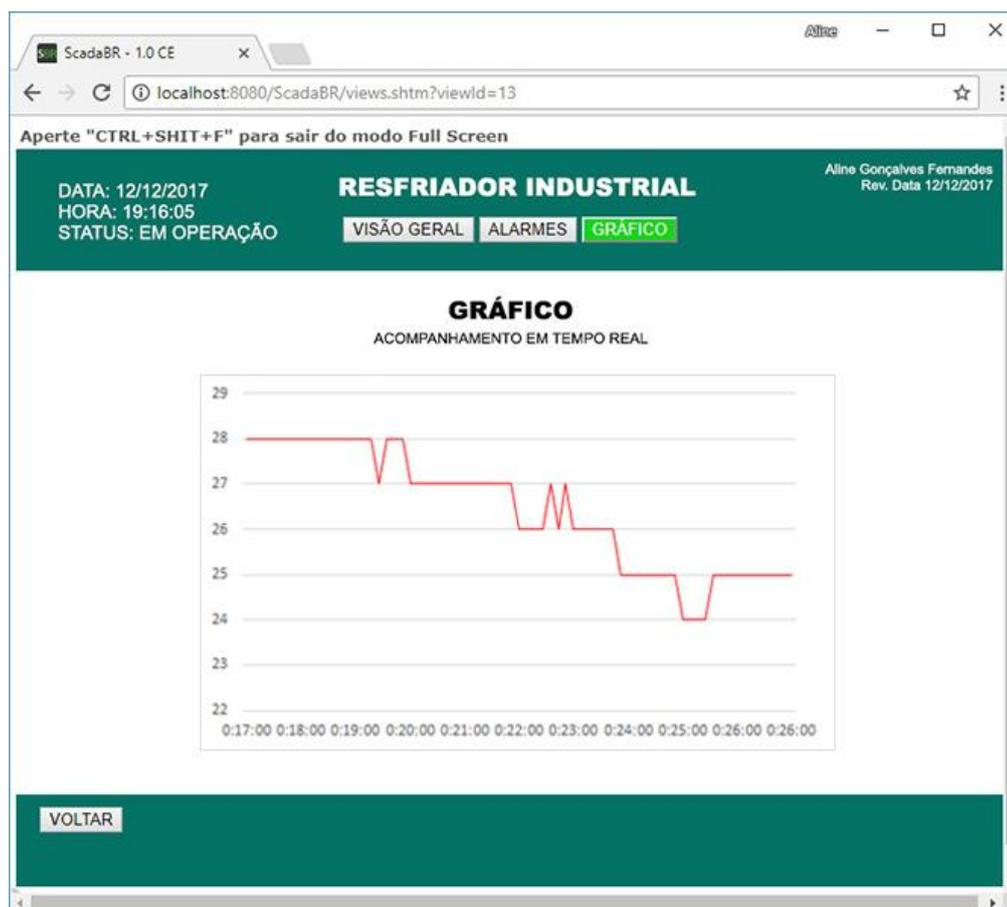


Figura 41 - Gráficos configurados para o Sistema Supervisório.

A partir da configuração e habilitação dos Data Points é iniciada a aquisição de dados e controle do processo em tempo real pela Beaglebone, em paralelo acontece o estabelecimento da conexão via socket entre os equipamentos e aplicações, a transferência dos pacotes via TCP formatados no padrão Modbus.

5. CONSIDERAÇÕES FINAIS

O desenvolvimento do projeto de um sistema de monitoramento industrial proposto mostrou-se proveitoso por agregar várias áreas do conhecimento exploradas ao longo da graduação em engenharia elétrica, como eletrônica de potência, sistemas de controle, sistemas supervisórios, sistemas distribuídos, redes industriais, protocolos de comunicação, bem como conhecimento da instrumentação da planta a ser trabalhada. Foi observado grande avanço no conhecimento dos sistemas open-sources embarcados para processamento da informação.

O desempenho da BeagleBone Black se mostrou satisfatório para os testes didáticos realizados, uma vez que atende aos requisitos de quantidade de entradas e saídas digitais configuráveis, resolução de 12bits no canal ADC, comunicação TCP e possível programação em C do sistema de controle.

A utilização do ScadaBR foi conveniente para estudar a planta proposta e realizar os testes de maneira mais intuitiva, configurando os data sources e data points a fim de realizar a comunicação TCP com a Beaglebone. Porém, em âmbito geral, a ferramenta é indicada para aplicações de pequeno porte, voltada para o desenvolvimento científico e didático, não sendo ideal para sistemas reais, complexos e de grande porte. Este fato é notável, visto que o ScadaBR, como sendo software open-source, não possui uma infraestrutura oficial para implementação e manutenção contínua.

A Libmodbus foi indispensável para a implementação do projeto, visto que abrange toda a conexão via socket entre o ScadaBR e a Beaglebone Black. Sua implementação e instalação, embora não seja tão intuitiva, está documentada e possui grande valor para futuros projetos envolvendo comunicação Modbus.

O projeto proposto, juntamente com seu desenvolvimento e documentação será útil para desenvolver aplicações futuras, com mais funcionalidades e maior possibilidade de controle. Como sugestão de trabalhos futuros é possível citar a instrumentação de uma planta industrial de pequeno porte a fim de realizar o controle manual e automático das variáveis de processo; a realização de pesquisa para aprimoramento do sistema como um todo, em especial o controle de temperatura através do uso de ventiladores mais eficientes.

O trabalho teve seu objetivo alcançado, visto possibilitou a utilização das diversas áreas de conhecimento adquiridas na engenharia. Também disponibilizou materiais, testes e comprovações para a sociedade e comunidade científica a fim de compartilhar e expandir o conhecimento.

6. REFERÊNCIAS

ALL DATA SHEET. Disponível em: < <http://www.alldatasheet.com/> >. Acesso em: 11 dez. 2017.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - ABNT. **NBR 6023:** informação e documentação / referências / elaboração. Rio de Janeiro, 2002. Disponível em: <<http://www.abnt.org.br/>>. Acesso em: 11 dez. 2017.

AUTOMAÇÃO INDUSTRIAL. **O que são Redes Industriais?** 2012. Disponível em: <<https://www.automacaoindustrial.info/o-que-sao-redes-industriais-parte-1/>>. Acesso em: 05 out. 2017.

BONFIM, Bruno. **Conhecimento é Sabedoria: Atuadores Rotativos e Lineares.** 2012. Disponível em: < <http://brunobonfimlopes.blogspot.com.br/2012/03/atuadores-rotativos-e-lineares.html> >. Acesso em: 05 out. 2017.

CCM - COMMENT CA MARCHE. **O que é Protocolo TCP/IP. 2017.** Disponível em: <<http://br.ccm.net/contents/285-o-que-e-o-protocolo-tcp-ip>>. Acesso em: 05 out. 2017.

COLEY, Gerald; DAY, Robert P. J. **BeagleBone Black System Reference Manual.** 2013. Disponível em: < https://cdn-shop.adafruit.com/datasheets/BBB_SRM.pdf >. Acesso em: 05 out. 2017.

FERNAEUS, Ylva; JONSSON, Martin; THOLANDER, Jakob. **Revisiting the jacquard loom: threads of history and current patterns in HCI, Proceedings of**

the **SIGCHI Conference on Human Factors in Computing Systems**, 2012, Austin, Texas, USA [doi>10.1145/2207676.2208280]

FREITAS, Carlos Márcio. **Embarcados: Rede de comunicação em RS-485**. 2017. Disponível em: <<https://www.embarcados.com.br/redes-de-comunicacao-em-rs-485/>>. Acesso em: 11 dez. 2017.

GROOVER, Mikell P. **Automação Industrial e Sistemas de Manufatura**: 3. ed. São Paulo: Editora Pearson, 2010.

INSTRUCTABLES. 2017. Disponível em: <<http://www.instructables.com/>>. Acesso em: 11 dez. 2017.

KUROSE, James; ROSS, Keith. **Redes de Computadores e a Internet. Uma abordagem Top-Down**: 5. ed. São Paulo: Editora Pearson, 2010.

LAMB, Frank. **Automação Industrial na Prática**. 1. ed. Porto Alegre. Editora AMGH, 2015.

Libmodbus. 2017. Disponível em: <<http://libmodbus.org/>>. Acesso em: 30 nov. 2017.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Fundamentos de Metodologia Científica**. 6. ed. São Paulo. Editora Atlas, 2007.

MODBUS. **Modbus Application Protocol Specification V1.1b3**. 2005. Disponível em: <http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf>. Acesso em: 08 out. 2017.

MODBUS. **Modbus Messaging On Tcp/Ip Implementation Guide V1.0b**. 2005. Disponível em: <
http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf
>. Acesso em: 08 out. 2017.

MORAES, Cícero Couto de; CASTRUCCI, Plínio de Lauro. **Engenharia de Automação Industrial**. 1. ed. Rio de Janeiro. Editora LTC, 2001.

PISKE, Rafael Bender. **Estágio em T.I. - Estúdio de Projeção de sites de Internet**. 1. ed. São Paulo. Editora Braúna, 2016.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de gerenciamento de banco de dados**. 3. ed. Porto Alegre. Editora AMGH, 2011.

PPLWARE. **Sabe o que são sockets de comunicação?** 2012. Disponível em: <
<https://pplware.sapo.pt/tutoriais/networking/redes-sabe-o-que-sao-sockets-de-comunicacao-parte-i/>>. Acesso em: 11 dez. 2017.

ROSÁRIO, J. **Princípios de Mecatrônica**: 1. ed. São Paulo: Editora Pearson, 2005.

ROSS, Júlio. **Redes de Computadores**: 1. ed. Rio de Janeiro: Editora Antenna, 2008.

SCADABR. **Manual do Software**. 2010. Disponível em: <
<http://www.scadabr.com.br/index.php/2017/06/06/2939/>>. Acesso em: 08 out. 2017.

Siemens. **s7-1200.** Disponível em:
<<http://w3.siemens.com.br/automation/br/pt/automacao-e-controle/automacao-industrial/simatic-plc/s7-cm/s7-1200/pages/default.aspx>>. Acesso em 20 nov. 2017.

SMAR. **Redes Industriais.** 2011. Disponível em:
<<http://www.smar.com/newsletter/marketing/index150.html>>. Acesso em: 05 out. 2017.

TERGOLINA, Rogerio Lampert. **Sensores Industriais: material para fins didáticos.** PUCRS. Escola Politécnica. Rio Grande do Sul: não publicado, 2015.