



Artigo

# Desenvolvimento de um sistema supervisor para telemetria de uma bomba de fluidos utilizando java, arduino e express-pcb

José Rosenildo Bezerra da Silva<sup>[1]</sup>, Leonardo Augusto Casillo<sup>[2]</sup>

<sup>[1]</sup> Universidade Federal Rural do Semi-árido; j.rosenildo@bol.com.br

<sup>[2]</sup> Universidade Federal Rural do Semi-árido; casillo@ufersa.edu.br

Recebido: 25/06/2019;

Aceito: 08/08/2019;

Publicado: 07/10/2019.

**Resumo:** Com os avanços tecnológicos, grande parte dos sistemas fabris que foram mecanizados para atender à crescente demanda por bens e serviços está se tornando obsoleta. Para garantir a alta produtividade e maximizar os lucros, é necessário automatizar a manufatura aplicando sistemas computacionais. O trabalho proposto busca construir um sistema supervisor composto de hardware e software capaz de colher informações sobre três variáveis físicas de uma bomba de fluidos (pressão, temperatura e vazão) e apresentá-las através de uma interface gráfica ao usuário em tempo real, utilizando o Arduino, o Java e o ExpressPCB, que são ferramentas gratuitas de desenvolvimento. Efetuou-se então uma pesquisa experimental de natureza aplicada, onde foram desenvolvidos um esquema elétrico e um layout de circuito, necessários na construção de um protótipo de placa eletrônica baseada no Arduino UNO, utilizando a ferramenta DAC ExpressPCB, e um aplicativo para Windows com uma interface gráfica amigável para exibição dos dados coletados utilizando a linguagem de programação Java no ambiente de desenvolvimento integrado Netbeans. Como resultado, após a realização dos testes de funcionamento, identificou-se a interação entre todos os elementos do sistema e a efetiva amostragem dos dados colhidos, evidenciando que é sim viável o desenvolvimento de sistemas supervisórios para telemetria de equipamentos industriais como a bomba de fluidos utilizando ferramentas de licença livre.

**Palavras-chave:** Sistemas Supervisórios, Arduino, Java, ExpressPCB, Licença Livre

## 1. INTRODUÇÃO

O homem busca sempre simplificar e garantir a segurança na execução de tarefas do seu cotidiano. É sabido que desde o paleolítico produziam-se ferramentas para facilitar a sobrevivência e que posteriormente a modernização desses utensílios possibilitou a manufatura de objetos e prestação de serviços. A partir da revolução industrial na Europa no século XVIII, essas manufaturas foram gradativamente sendo mecanizadas pelo mundo, fazendo surgir as grandes e complexas estruturas fabris, iniciando a produção e comercialização em massa. Essas estruturas eram ambientes insalubres que exigiam uma grande quantidade de mão-de-obra, geravam muito desperdício de matéria prima, além de facilitar a ocorrência de falhas operacionais e acidentes de trabalho.

Nesse entremeio, a eletrônica e suas aplicações evoluíam timidamente, porém no século XIX, a manipulação de materiais semicondutores possibilitou a criação de dispositivos que aceleraram esse processo.

Em dezembro de 1947 os físicos William Shockley, John Bardeen e Walter Brattain apresentaram ao mundo o primeiro transfer resistor (transistor), demonstrando seu uso na amplificação de sinais nos laboratórios da empresa Bell Telephone [1]. A partir de então, circuitos eletrônicos passaram a ser desenvolvidos inicialmente para monitorar os equipamentos e posteriormente para controlar as suas funcionalidades em parte ou no todo criando o que conhecemos hoje por automação e controle industrial. Ficou visível a diminuição de erros e acidentes, além do aumento na produtividade, garantindo uma sensível redução nos custos de produção, o que gera melhorias na eficiência das estruturas fabris.

No estado do Rio Grande do Norte, dentre as atividades industriais mais fortes destaca-se a exploração de petróleo que faz uso de um grande maquinário eletromecânico. Nessa atividade existem processos produtivos complexos, que oferecem riscos ao meio ambiente e à saúde dos trabalhadores, que precisam ser monitorados constantemente e até operados remotamente para evitar falhas, perdas na produção ou acidentes. Para a extração do petróleo é necessário perfurar poços e efetuar manutenções periódicas nos mesmos, utilizando equipamentos denominados de sondas. As sondas são equipamentos eletromecânicos compostos por sistemas como sustentação de cargas, movimentação de cargas, geração de energia, rotação, circulação de fluidos, dentre outros que são automatizados e podem ser operados manualmente ou remotamente, a depender da complexidade da operação.

A utilização de sistemas supervisórios é vantajosa, pois possibilita o acompanhamento contínuo, a análise de desempenho e a identificação de falhas. Considerando que uma solução proprietária para qualquer sistema supervisório pode-se apresentar de forma onerosa, utilizando uma equipe multidisciplinar capacitada, vislumbra-se aqui a possibilidade da construção de um sistema, mantendo o foco na funcionalidade, aplicando ferramentas de código aberto e licença livre, com as quais entramos em contato durante o curso de graduação. Nesse sentido, a atenção se voltará para o sistema de circulação, que tem como principal componente a bomba de fluidos, geralmente é de operação manual e requer apenas monitoramento de parâmetros como vazão, pressão e temperatura. Portanto o objetivo será construir um sistema supervisório aplicável à telemetria da bomba de fluidos utilizando o Arduino, a linguagem de programação Java e o ExpressPCB como ferramentas de desenvolvimento.

## 2. REFERENCIAL TEÓRICO

### 2.1. Automação

Antes de expor aplicações específicas da automação é preciso apresentar seus conceitos básicos e um breve histórico. Automatizar significa dotar um equipamento ou processo produtivo de meios que permitam-lhe realizar determinadas tarefas sem a intervenção humana. De forma simplificada, automação é a substituição do trabalho humano ou animal por máquinas [2]. Ações humanas para reduzir ou eliminar esforços braçais na execução de tarefas tem origem na pré-história com a roda, os moinhos de vento, etc. Porém foi na revolução industrial que a automação ganhou força, passando a ser utilizada como importante ferramenta para as mais diversas atividades humanas. O Brasil, onde a atividade primária movimentava a economia quase que na sua totalidade, apenas na segunda metade do século XX viu a industrialização prosperar de maneira concreta.

Os investimentos e esforços contínuos desencadearam o surgimento e integração de novas tecnologias, dando origem à chamada era da informação. A mecanização e automatização tornaram-se ações obrigatórias na atividade industrial, pois possibilitaram reduções nas jornadas de trabalho, aumento na produtividade, maior segurança e um menor desperdício de matéria prima. Essa corrida pela modernização das plantas industriais fez com que diferentes empresas construíssem diversas formas distintas de comunicação e troca de dados entre equipamentos, gerando incompatibilidades, sendo necessária a criação de associações técnicas para padronizar as soluções. Para manter e conquistar novos mercados é necessário desenvolver soluções compatíveis com os padrões existentes no mercado industrial [3].

O advento dos computadores, além de alavancar o processo de automatização da indústria, por oferecer inúmeras possibilidades devido à grande capacidade de processar e armazenar informações viabilizou o uso da automação em outros setores da sociedade como no campo, no comércio, nas residências, etc. O importante é salientar que, sem o computador, não seria possível o nível de automação em que se vive atualmente [4].

### 2.2 Sistemas Supervisórios

O presente estudo tem como foco a automatização dos processos industriais, que a depender da atividade, por questões de comodidade ou segurança, necessitam de monitoramento à distância, requerendo a aplicação de sistemas supervisórios. Tais supervisórios denominados de SCADA (*Supervisory Control and Data Acquisition*) consistem em um conjunto de elementos físicos (*Hardware*) e lógicos (*Software*) com a finalidade de monitorar as instalações. As informações coletadas são manipuladas e apresentadas aos usuários para auxiliá-los nas tomadas de decisão. Os primeiros sistemas SCADA, basicamente telemétricos, permitiam informar periodicamente o estado corrente do processo industrial [5].

As tecnologias computacionais e da informação, aplicadas aos supervisórios, permitem o monitoramento de processos em ambientes complexos, geograficamente dispersos, além de possibilitar o tratamento lógico e

matemático de inúmeras variáveis ambientais simultaneamente. Com a telemetria, é possível supervisionar toda uma instalação fabril de qualquer lugar do planeta.

A parte física que compõe os supervisórios está dividida em sensores e atuadores, os módulos de aquisição de dados, uma estação central de processamento dos dados (CPD), e uma rede de comunicação. Os sensores são elementos que convertem os parâmetros das variáveis físicas em sinais elétricos analógicos ou digitais, e os enviam aos módulos de aquisição. Os atuadores são elementos utilizados para controlar o funcionamento dos equipamentos monitorados. Esses dispositivos são instalados junto aos equipamentos que compõem o processo industrial. Os módulos de aquisição de dados por sua vez, recebem e processam os sinais oriundos dos sensores, enviando-lhes ao sistema computacional central. Esses dispositivos se dividem em Controladores Lógicos Programáveis (CLP's) e Unidades Terminais Remotas (UTR's). A diferença entre esses dois tipos de módulos é que o primeiro apresenta flexibilidade na linguagem de programação e controle das entradas e saídas, enquanto que o segundo possui uma arquitetura mais distribuída entre a central de processamento e suas entradas e saídas, possibilitando uma maior precisão nas medições [5]. Os meios físicos de comunicação entre todos os dispositivos compõe a rede de comunicação, que deve seguir os padrões estabelecidos pelas agências regulamentadoras. A parte lógica do supervisório compreende o software desenvolvido para os sistemas operacionais e o firmware desenvolvido para os hardwares de aquisição de dados. Ela possibilita a comunicação entre os dispositivos, a coletar, o processamento e armazenamento os dados, além de apresentá-los ao usuário através de uma interface gráfica amigável.

Com a competitividade do mercado em todos os seguimentos, faz-se necessária a redução de custos e o aumento na produtividade, criando a demanda pela aplicação da automação nos processos. Existe uma grande evolução nas tecnologias computacionais para desenvolvimento de sistemas SCADA, que aumentam a confiabilidade, flexibilidade e conectividade das soluções, reduzindo o tempo de configuração e adaptação do sistema às necessidades de cada instalação [5].

Os sistemas supervisórios são geralmente soluções proprietárias desenvolvidas por empresas especializadas, o que eleva o seu custo e dificulta sua utilização em larga escala. Porém atualmente existem diversas ferramentas de desenvolvimento de código aberto como Arduino, Java e ExpressPCB, que vem sendo aprimoradas e despontam como alternativas para massificação de soluções em automação.

### *2.3 Aquisição e Tratamento de Dados*

Para compreender melhor o funcionamento de um sistema supervisório é necessário detalhar o caminho existente entre a informação apresentada ao usuário e a variável física que está sendo analisada. De forma generalista, começaremos definindo a aquisição de dados como o processo pelo qual a intensidade ou estado da variável física é convertido em sinais elétricos pelos sensores, levados aos módulos de aquisição de dados através de cabos elétricos simples ou de redes de comunicação industrial, onde são convertidos para a linguagem de máquinas e posteriormente transmitidos para o sistema computacional. O tratamento de dados é o processo no qual o sistema computacional manipula essas informações digitais, aplicando-lhes tratamentos lógicos, estatísticos, matemáticos, dentre outros, podendo armazená-los em uma base de dados e apresentá-los ao usuário através de uma interface gráfica [6].

Os tipos de sensores industriais mais utilizados são de posição, magnéticos, capacitivos, indutivos, óptico, ultrassônico, de pressão, de temperatura e de encoder. É considerado um tipo de transdutor e aplicados em larga escala nos mais diversos equipamentos que utilizamos no cotidiano, fornecendo como saída sinais elétricos que podem ser discretos ou analógicos. Os sinais discretos são pulsos elétricos (LIGADO/DESLIGADO) que tem origem em uma mudança de estado no sensor, a depender da presença ou ausência de um objeto alvo. Os sinais analógicos são contínuos, podendo representar uma variação de tensão ou corrente dentro de uma determinada faixa (RANGE), que será proporcional à variável física analisada. Descreveremos a seguir de forma superficial, o princípio de funcionamento de alguns sensores.

Os sensores de posição possuem contatos elétricos internos que podem ser normalmente abertos ou fechados, e um elemento de manobra que atuará em contato direto com o objeto alvo e provocará uma mudança de estado nos contatos internos, fornecendo um sinal discreto. Os sensores magnéticos possuem características similares aos de posição, porém seu mecanismo de comutação interno é composto por um material ferromagnético que será atuado através de um campo magnético externo preso ao objeto alvo, dispensando o contato direto. Os sensores indutivos são dotados de um circuito eletrônico que produz um campo magnético de alta frequência, detectando a presença de materiais ferromagnéticos a uma determinada distância, dispensando o contato direto com o objeto alvo. Os sensores capacitivos são constituídos por um circuito

eletrônico que produz um campo elétrico, e que a exemplo dos sensores indutivos, também detectará a proximidade de um objeto alvo, porém não se limitando a materiais ferromagnéticos, podendo ser aplicado para qualquer tipo de material.

Os sensores Ópticos também fornecem um sinal de saída discreto, funcionam através da emissão e recepção da luz infravermelha, que poderá ser interrompida ou refletida pelo objeto alvo. Os sensores ultrassônicos funcionam através da emissão, recepção e análise de ondas sonoras, que são refletidas ou interrompidas na presença do objeto alvo e podem fornecer sinais elétricos discretos ou analógicos. Com relação aos sensores de pressão e temperatura, ambos convertem a grandeza analisada em um sinal elétrico analógico. Por fim, os encoders são dispositivos dotados de circuitos eletrônicos que possibilitam a medição de velocidade de rotação ou posição de um objeto com alta precisão. Apesar de muitas literaturas não o considerarem como sensor, o encoder pode ser considerado um sensor de posicionamento que envolve deslocamento angular mecânico, reflexão óptica e conversão em sinais elétricos [3].

Os sensores possuem especificações técnicas de operação como tensão, temperatura, faixa de medição, grau de proteção, que dependem dos materiais e da engenharia utilizados no seu processo de fabricação. Como existe uma grande variedade de modelos no mercado, para escolha do sensor correto, garantindo funcionalidade e durabilidade, é preciso levar em consideração tanto a grandeza medida, quanto as características do local de instalação como espaço físico disponível, tensões de alimentação presentes, presença de partículas dispersas, umidade, temperatura, etc.

#### 2.4 Arduino

No ano de 2005 na Itália, os pesquisadores David Cuartielles, David Mellis, Gianluca Martino, Massino Banzi e Tom Igoe desenvolviam um projeto com o simples intuito de construir um hardware que possibilitasse a prototipagem de circuitos eletrônicos com um custo menor do que as ferramentas já disponíveis no mercado. Devido ao caráter flexível do hardware livre e a facilidade na utilização, logo essa plataforma tornou-se popular, vendendo milhares de exemplares pelo mundo. O Arduino apresenta-se sob forma de placas construídas com microcontroladores da família AVR (*Advanced Virtual RISC*) comercializadas pela Atmel Corporation, que fornecem uma série de entradas e saídas para conexão com outros dispositivos como sensores, atuadores, placas de expansão, etc. Tanto o hardware quanto o software são de fonte aberta, permitindo que indivíduos utilizem a ferramenta para qualquer propósito. A Figura 10 apresenta o Arduino MEGA 2560, uma das placas mais difundidas no mercado [7].

Podem-se usar as placas do Arduino para se montar equipamentos independentes, equipamentos que interajam com outros, como por exemplo, com computadores, ou ainda usá-las como base para desenvolver outros circuitos personalizados de acordo com a necessidade de aplicações específicas como em módulos de aquisição de dados para sistemas SCADA. Ao usar a plataforma como base, foca-se no microcontrolador Atmel, dessa forma parte dos esquemas elétricos das placas do Arduino são conservadas, e outros dispositivos são incrementados de acordo com a necessidade. Salienta-se que, apesar de a programação do microcontrolador poder ser realizada usando qualquer tipo de gravador de EPROM (*Erasable Programmable Read-Only Memory*), essa é uma tarefa atribuída e realizada com eficiência pelas placas do Arduino.

#### 2.5 Java

A linguagem de programação Java originou-se nos anos de 1990 em um projeto da Sun Microsystems que procurava viabilizar o desenvolvimento de aplicações para dispositivos eletrônicos de uso popular. É uma linguagem orientada a objetos, com sintaxe baseada em C++ e possui um vasto conjunto de bibliotecas de classes com recursos matemáticos, de rede, de segurança, entre outros. Os aplicativos desenvolvidos em Java são compilados em um código (*Bytecode*) que será lido e executado por uma máquina virtual, por isso a linguagem é dita como interpretada. Para se executar um aplicativo Java, obrigatoriamente deve-se possuir uma máquina virtual Java instalada em seu sistema operacional [8].

O Java atualmente pertence à empresa americana de tecnologia da informação Oracle Corporation, especializada no desenvolvimento e comercialização de software e hardware. Apesar de existir os kits de desenvolvimento (*JDKs premium*) pagos, a maior parte do Java está sob licença de software livre GPL (*General Public License*) e as funcionalidades da versão de fonte aberta continuam sendo aprimoradas, possibilitando aos desenvolvedores implementar soluções que fazem frente a outras linguagens muito populares na construção de softwares comerciais como o Visual Basic (VBA) [9].

Existem diversos IDEs para Java dos quais podemos destacar o NetBeans e o Eclipse por serem de fonte aberta e possibilitarem o desenvolvimento de aplicações desktop, web, mobile, etc. Existem também IDEs voltadas ao desenvolvimento de aplicações mobile como Android Studio e IntelliJ Idea com enfoque para aplicativos Android que também são muito utilizados. Fica então a critério do desenvolvedor a escolha do IDE que lhe proporcione uma maior familiaridade, produtividade ou mesmo um melhor custo benefício.

### 2.6 ExpressPCB

O ExpressPCB é uma ferramenta DAC composto de dois softwares para desenvolvimento de circuitos eletrônicos e elaboração de placas de circuito impresso. É uma plataforma gratuita para quem precisa desenvolver circuitos eletrônicos com precisão e profissionalismo. Não possui uma aparência amigável se comparado com outras ferramentas proprietárias de mesma finalidade, porém a disposição dos elementos e a introdução de um guia completo demonstra seu caráter didático, possibilitando que iniciantes consigam elaborar seus projetos sem muitas dificuldades.

A plataforma conta com uma comunidade on-line onde hobbystas, técnicos e engenheiros compartilham seus projetos, bibliotecas e experiências. Assim o ExpressPCB é o DAC gratuito mais utilizado para essa finalidade, pois reúne a qualidade presente nos softwares proprietários e a facilidade de utilização presente nas ferramentas pensadas para iniciantes [10].

## 3. MATERIAIS E MÉTODOS

A pesquisa realizada foi experimental, de natureza aplicada com objetivo explicativo e realizou-se no Laboratório de Automação, Arquitetura de Computadores e Sistemas Embarcados – LAACOSTE durante o período letivo de 2016.2 com duração de 14 (quatorze) semanas. Teve como objeto de estudo, protótipos de Hardware e Software desenvolvidos utilizando ferramentas gratuitas. O hardware foi desenvolvido usando o conjunto de ferramentas CAD do ExpressPCB, uma placa UNO da plataforma Arduino, uma placa de expansão para comunicação ethernet e alguns componentes eletrônicos.

Para a construção do hardware, inicialmente fez-se o levantamento de requisitos, de forma a permitir a escolha de sensores, do microcontrolador para o sistema embarcado, e demais dispositivos eletrônicos de forma otimizada. Nesse momento a plataforma Arduino junto com o módulo de expansão ethernet se mostra ideal por possibilitar os diversos testes em bancada de forma ágil, e quando integrada ao ExpressPCB, permite ajustes rápidos tanto no esquema elétrico quanto no layout da placa de circuito impresso. O software foi desenvolvido usando a linguagem de programação JAVA na IDE Netbeans e a comunicação entre software e hardware dar-se-á através de uma intranet usando o conjunto de protocolos TCP/IP.

A simulação de alterações nas variáveis analisadas, a visualização no software e o registro fotográfico serviram como instrumentos de coleta de dados, assim como a interação entre os elementos do sistema serviu como método de análise de desempenho e viabilidade, pois demonstrou o seu funcionamento de forma satisfatória. A Figura 1 mostra o fluxograma que descreve de maneira simplificada o funcionamento do sistema supervisorio.

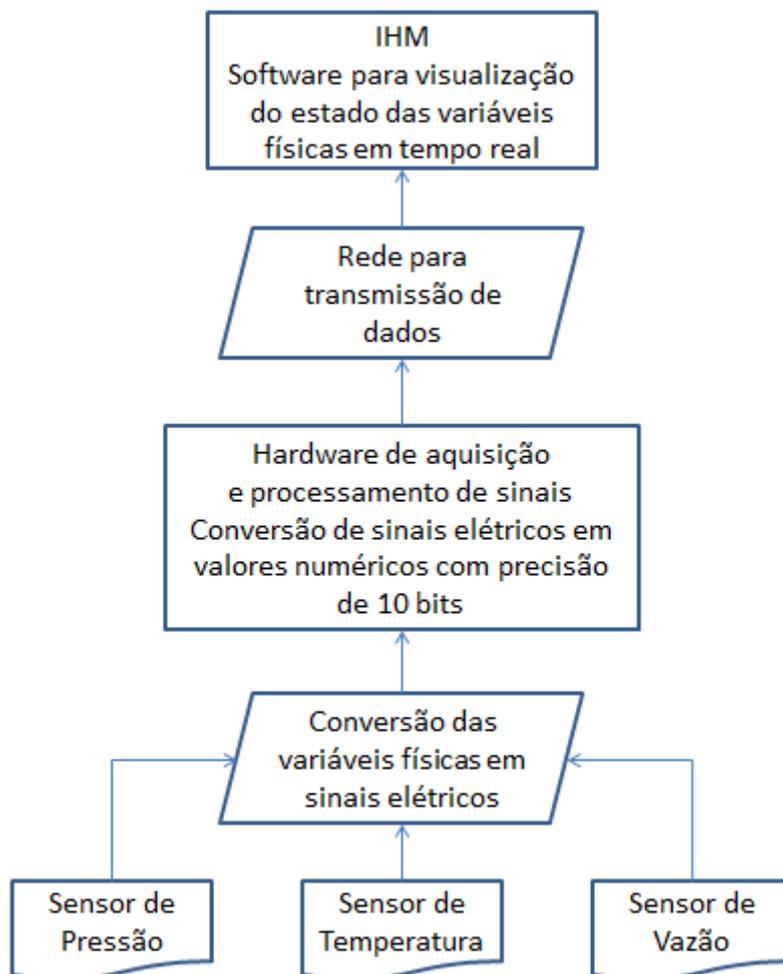


FIGURA 1. Fluxograma do funcionamento do sistema. (Autoria própria)

#### 4. RESULTADOS E DISCUSSÃO

Ao se propor a construção de um sistema, com o intuito de criar um produto que será comercializado, deve-se levar em consideração todas as etapas de um projeto, desde o levantamento de requisitos até a documentação e normatização para que se alcancem níveis de qualidade e durabilidade exigidos pelo mercado. Essa análise detalhada e estruturada do projeto demanda uma quantidade maior de tempo, que será recuperado devido a maior eficiência, confiabilidade e um menor número de correções nos produtos finais. No entanto, para o protótipo em questão onde o tempo está reduzido a algumas poucas semanas, visa-se alcançar as funcionalidades de um sistema supervisor, ficando sua otimização para trabalhos futuros.

##### 4.1 Construção do Hardware

Como parte do levantamento de requisitos do hardware, inicialmente fez-se a determinação das variáveis físicas críticas no funcionamento da bomba de fluidos. A bomba de fluidos é composta por uma parte mecânica que transforma a energia rotativa de um motor em um movimento alternativo que será aplicado aos pistões, e por uma parte hidráulica que transfere o movimento alternativo dos pistões para o fluido, gerando pressão e vazão. O equipamento posto em funcionamento cria uma pressão no sistema e coloca o fluido em movimento, o que faz surgir temperaturas elevadas em diversos pontos. Identificamos assim três variáveis físicas fundamentais - pressão, vazão e temperatura - que precisam ser monitoradas para garantir a eficiência do equipamento e preservar sua vida útil, além de proporcionar a segurança dos operadores e transeuntes.

Para monitorar a pressão, a vazão do fluido, e a temperatura no ponto mais crítico do equipamento, é preciso determinar os sensores a serem utilizados. A pressão e a temperatura são variáveis contínuas, o que requer sensores analógicos. No mercado existem diversos sensores analógicos dos mais variados fabricantes, que fornecem sinais elétricos de corrente (0 a 20mA ou 4 a 20mA) e de tensão (0 a 10V). Os dispositivos mais

utilizados, e que portanto serão escolhidos para essa finalidade, são os sensores com saída de corrente 4 a 20mA. Já a vazão, tanto pode ser determinada utilizando um sensor analógico, aplicando-o diretamente a um local por onde circule o fluido, quanto utilizando um sensor de proximidade, aplicando-o junto a um dos pistões e fazendo uma relação matemática entre a frequência do movimento dos pistões e a quantidade de fluido deslocado por estes. Como o monitoramento está levando em consideração apenas uma bomba de fluidos, os sensores discretos são mais utilizados para esse fim, pois garantem uma medição bem aproximada da vazão, com um grau de complexidade relativamente baixo e um custo inferior aos sensores analógicos de vazão.

Ao se determinar que serão monitoradas três variáveis físicas utilizando apenas três sensores, sabe-se que a quantidade de entradas de sinais necessárias no hardware de aquisição será pequena, portanto pode-se utilizar uma das placas do Arduino mais básicas como base para o protótipo. Escolhe-se a placa UNO, que é uma das mais utilizadas, em princípio por existir em abundância no mercado a um baixo custo, posteriormente por ser dotada de um microcontrolador com seis entradas analógicas e quatorze entradas digitais, das quais seis podem ser utilizadas para modulação por largura de pulso, o que atenderia à necessidade inicial e ainda possibilitaria o incremento de diversas outras funcionalidades ao protótipo.

Como as entradas analógicas do Arduino trabalham com uma faixa de tensão de 0 a 5V e os sensores analógicos escolhidos fornecem um sinal de corrente de 4 a 20mA, faz-se necessário o incremento de um circuito que possibilite a conversão de sinal. Dentre as possibilidades, existe um amplificador operacional de canal único (mono) desenvolvido pela empresa americana Texas Instruments, o RCV420. Além de efetuar a conversão, ele fornece segurança ao microcontrolador, pois possui proteções que limitam as tensões e correntes na sua saída. O ônus de utilizar esse componente é que ele, diferente do circuito do Arduino, requer uma fonte de alimentação simétrica, ou seja, com uma tensão positiva +V e outra tensão negativa -V, além da referencia comum de terra GND. Porém esse tipo de fonte é muito comum e de fácil confecção, utilizando componentes de baixo custo.

Como as entradas digitais do Arduino possuem uma tensão limite de 5V, e os sensores de proximidade disponíveis no mercado operam com tensões na faixa de 10V a 36V, faz-se necessário o incremento de um circuito que possibilite a atenuação desse sinal. Apesar de um circuito como a associação de resistores ser a solução mais simples e menos onerosa, para garantir a integridade do microcontrolador em caso de falhas, opta-se pela utilização de um acoplador óptico, que apresenta uma maior eficiência.

A placa Arduino UNO suporta alguns protocolos de comunicação como a interface Serial TTL (Transistor-Transistor Lógica), a comunicação SPI (*Serial Peripheral Interface*) e as comunicações por barramento One-Wire e I2C. Como os protocolos Serial, One-Wire e I2C exigem uma maior quantidade de dispositivos para serem integrados aos computadores modernos, além de possuir algumas limitações como volume de informações e distância entre dispositivos na instalação fabril, acabam perdendo espaço para a comunicação SPI, já que existem diversas placas de expansão no mercado para comunicação via cabo ou WiFi que usam o protocolo SPI, dentre as quais se destacam o Arduino WiFi, o módulo WiFi ESP8266, o Arduino Ethernet W5100 e o módulo ethernet ENC28J60. Como a maior parte das redes de chão de fábrica para sistemas supervisórios são cabeadas, opta-se pela comunicação SPI usando o módulo ethernet ENC28J60, apresentado na Figura 2, para o protótipo de hardware.

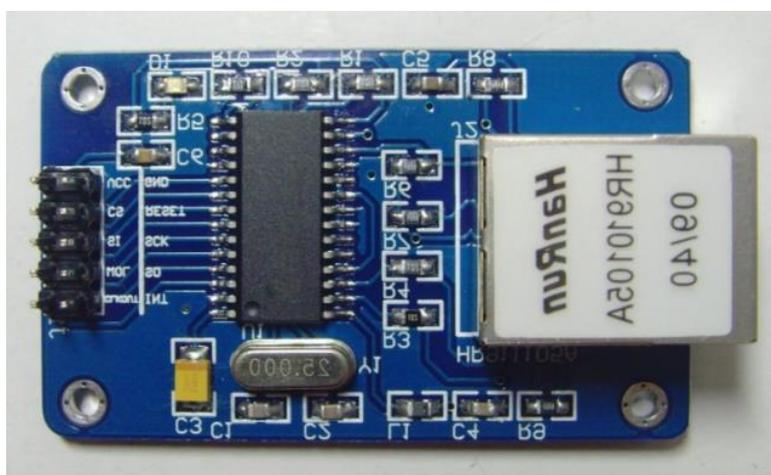


FIGURA 2. Foto da placa de expansão do módulo ethernet ENC28J60. (Autoria própria)

Após a determinação das variáveis físicas de interesse na bomba de fluidos, dos tipos de sensores e componentes adicionais para integra-los ao microcontrolador, da placa de expansão para comunicação e da análise do esquema elétrico da placa base do Arduino, pode-se iniciar o processo de desenvolvimento do esquema do circuito eletrônico que condensará todos esses elementos. De forma a simplificar sua alimentação, o circuito terá uma entrada para fonte simétrica (+12V, -12V e GND) e serão empregados circuitos integrados reguladores de tensão para controlar os níveis necessários para cada conjunto com alimentação semelhante. O esquema do circuito eletrônico completo está na Figura 3.

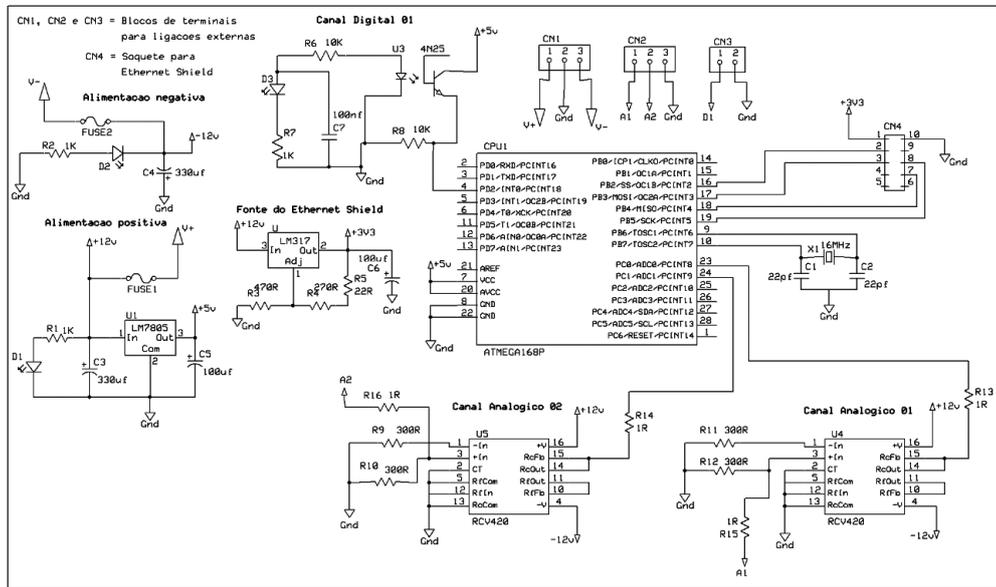


FIGURA 3. Esquema do circuito eletrônico. (Autoria própria)

Com o esquema finalizado, é possível iniciar o processo de desenvolvimento do layout da placa de circuito impresso usando o ExpressPCB. É preciso fazer algumas ressalvas quanto ao aspecto construtivo da placa eletrônica, já que produtos desenvolvidos para o mercado precisam obedecer a diversas normas técnicas regulamentadoras, e também devem-se considerar variáveis como os efeitos magnéticos e de interferências por sinais elétricos dinâmicos e estáticos. No entanto, foram levadas em consideração apenas as características dos componentes e dispositivos utilizados para se determinar níveis de tensão e corrente elétrica, disposição dos elementos e espessura das trilhas e ilhas na placa eletrônica. A Figura 4 apresenta o layout da placa do protótipo no ExpressPCB.

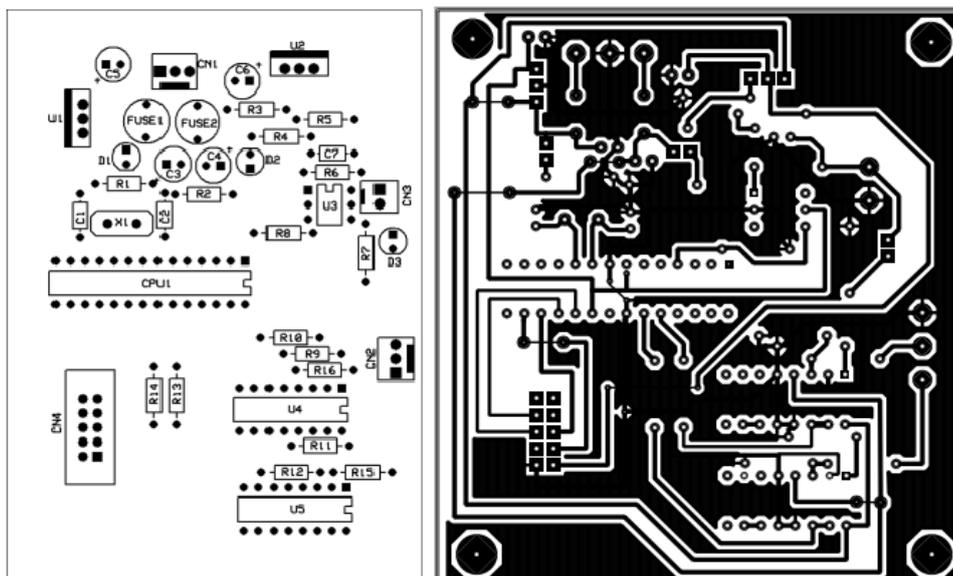


FIGURA 4. Layout da placa do circuito. (Autoria própria)

Após a confecção da placa, serão executadas a furação e montagem dos componentes eletrônicos para finalizar o processo de produção do protótipo. Por fim, dá-se o processo de desenvolvimento do programa do hardware (*firmware*), que para o protótipo em questão, utilizou-se o IDE do Arduino e a placa UNO, já que optou-se por não implementar uma interface de programação no protótipo do hardware. O protótipo finalizado e o código do firmware são apresentados nas Figuras 5 e 6.



FIGURA 5. Protótipo do *Hardware*. (Autoria própria)

```
#include <ETHER_28J60.h>
#include <avr/wdt.h>
static uint8_t mac[6] = {
  0x54, 0x55, 0x58, 0x10, 0x00,
static uint8_t ip[4] = {
  10, 0, 0, 254};
static uint16_t port = 80;
ETHER_28J60 ethernet;
const int sensor1 = A0, sensor;
float digital1;
unsigned long timeold;
void setup() {
  ethernet.setup(mac, ip, port);
  attachInterrupt(0, rpm_fun, R
  wdt_enable(WDTO_1S);
  digital1 = 0;
  timeold = 0;
}

void loop() {
  wdt_reset();
  if(timeold != 0 && (millis()-timeold) > 5000){
    timeold = 0;
    digital1 = 0;
  }
  if(ethernet.serviceRequest()) {
    ethernet.print(analogRead(sensor1)*2.93);
    ethernet.print("\n");
    ethernet.print(analogRead(sensor2)*0.1955);
    ethernet.print("\n");
    ethernet.print((int)digital1);
    ethernet.respond(); }
}

void rpm_fun() {
  if(timeold != 0) {
    digital1 = ((60000/(millis()-timeold))*0.815999);
  }
  timeold = millis();
}
```

FIGURA 6. Código do programa escrito em *Wiring*. (Autoria própria)

Quanto ao funcionamento do hardware, foram executadas medições de corrente elétrica nos canais analógicos e verificou-se uma relação linear com os valores mapeados e apresentado pelo conversor AD, como também observou-se uma relação direta entre a variação na velocidade de rotação do motor e os valores apresentados pelo hardware após a realização de uma requisição ao mesmo conectando-o a uma intranet e usando o browser de um computador. Isso demonstra o bom funcionamento tanto da leitura dos sensores quanto da resposta às requisições externas.

#### 4.2 Construção do Software

Dentro dos requisitos do software existem algumas funcionalidades necessárias para considera-lo parte de um sistema supervisorio, como por exemplo: a capacidade de interagir com os dispositivos de hardware no campo, como também a capacidade de coletar, tratar e apresentar as informações sobre as variáveis monitoradas de forma clara e precisa em tempo real. Grande parte dos sistemas supervisorios encontrados no mercado são soluções proprietárias desenvolvidas em VBA, utilizam os padrões de comunicação baseados no protocolo industrial COM (*Component Object Model*) desenvolvido pela Microsoft Corporation, visando dessa forma facilitar a utilização desses softwares nos sistemas operacionais Windows, por serem os mais populares.

A linguagem de programação Java possui um conjunto diverso de bibliotecas que torna possível a implementação dessas funcionalidades. Dentre os IDE's mais utilizados, o Netbeans destaca-se por facilitar a construção da interface gráfica, pois disponibiliza em um arranjo didático, vários elementos pré-definidos e prontos para utilização como campos de texto, botões, painéis, barras de menus, etc. Isso facilita o aspecto construtivo da interface e agiliza essa etapa do projeto, principalmente quando o projetista não é um programador de ofício.

A maioria dos softwares supervisorios disponíveis no mercado apresentam interfaces gráficas com visão de processo, usando animações para facilitar a compreensão por parte dos operadores. Esse tipo de funcionalidade exige a utilização de ferramentas gráficas mais sofisticadas que transcendem o propósito deste trabalho, ficando então determinado que a apresentação numérica da intensidade das variáveis é suficiente para fornecer de forma clara a informação sobre o estado das mesmas. Cria-se então uma classe JFrame e adicionam-se todos os demais elementos gráficos necessários para dar forma à interface. Afim de aproximar a aparência da interface gráfica com os dispositivos digitais de medição utilizados, adiciona-se uma imagem de fundo.

A função de coleta de dados do hardware é implementada criando-se um método que use um Socket Java pode ser visto na Figura 7, onde define-se a identificação do dispositivo de hardware na rede através do seu IP (*Internet Protocol*) e envia-se uma requisição ao mesmo, que quando respondida será interpretada e apresentada. Também foi adicionada uma rotina que verifica a eficiência na comunicação entre o software e o hardware, para em casos de falhas, retornar mensagens de erro que facilitem a identificação de problemas.

```
private void teste3() throws IOException {
    Socket s = new Socket();

    String host = "10.0.0.254";
    PrintWriter s_out = null;
    BufferedReader s_in = null;
    String vl[];

    try {
        s.connect(new InetSocketAddress(host, 80));
        //System.out.println("Conectado!");
        jLabel18.setForeground(new java.awt.Color(41, 150, 42));
        jLabel18.setText("On-Line: Registrando");
        jPanel11.setBackground(new java.awt.Color(215, 215, 215));

        //writer for socket
        s_out = new PrintWriter(s.getOutputStream(), true);
        //reader for socket
        s_in = new BufferedReader(new InputStreamReader(s.getInputStream()));
    } //Host not found
    catch (UnknownHostException e) {
        System.err.println("Sem resposta de : " + host);
    }
}
```

FIGURA 7. Trecho do código da aplicação usando a classe Scheduler. (Autoria própria)

Após implementar todo o código do aplicativo, o arquivo .JAR foi gerado e executado, sendo possível visualizar os valores das variáveis na interface gráfica como mostra a Figura 8, demonstrando a efetiva comunicação entre o software e o hardware. Os valores apresentavam a atualização cíclica determinada no código, exatamente em resposta às variações nos potenciômetros. É visível também uma pequena variação nos valores apresentados nas variáveis dos canais analógicos, em um percentual que pode ser significativo a depender dos critérios de avaliação das medições, como por exemplo, a precisão e a estabilidade. Como essas variações limitam-se aos canais analógicos, medições realizadas relacionam o valor da corrente de entrada de um dos amplificadores operacionais, o valor da tensão de saída do mesmo e o valor apresentado na interface gráfica. A Figura 9 apresenta uma série de amostras realizadas com intervalos de um segundo durante o período de um minuto, e a partir dele é possível identificar os circuitos integrados RCV420 como geradores do erro. As causas não foram identificadas, podendo advir da amplificação de ruídos, de interferências eletromagnéticas, etc.



FIGURA 8. Interface gráfica da aplicação em funcionamento. (Autoria própria)

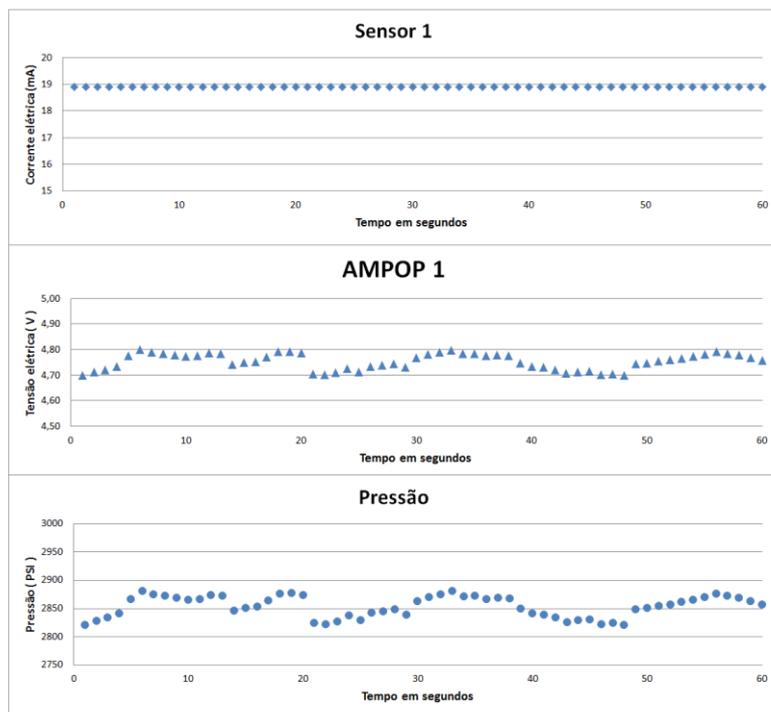


FIGURA 9. Relação entre corrente de saída do sensor, tensão de saída do conversor e valor da variável apresentado na interface gráfica. (Autoria própria)

Nesse capítulo, foram apresentadas todas as etapas do processo de construção do protótipo do sistema, com a determinação das variáveis a serem analisadas, a escolha de cada componente do circuito, a construção da placa eletrônica, a construção da aplicação em Java, bem como os testes que demonstraram o funcionamento efetivo do sistema. A seguir, se faram as considerações a respeito dos resultados do projeto.

## 5. CONCLUSÕES

O trabalho proposto evidenciou, após a realização dos testes e a verificação da efetiva funcionalidade do sistema, que é sim possível desenvolver um supervisor para telemetria de equipamentos como a bomba de fluidos utilizando ferramentas com licença livre, que são facilmente encontradas disponíveis para baixar em páginas da internet, juntamente com manuais e tutoriais que simplificam seu uso. O ExpressPCB mostrou-se eficiente tanto na elaboração do circuito eletrônico, quanto na construção do layout da placa, disponibilizando uma boa quantidade de componentes em sua biblioteca e também possibilitando a construção de novos componentes de forma descomplicada. O circuito baseado na placa UNO do Arduino, mesmo com a ausência de uma blindagem eletrostática, apresentou uma coleta de dados e uma resposta às requisições externas confiáveis, que são as características principais de um módulo de aquisição de dados adequado para aplicações em automação industrial.

O Java e sua vasta biblioteca de classes, não apenas tornou o processo de construção do software supervisor possível, mas também forneceu recursos para implementação de funcionalidades como a comunicação via rede com uma intranet ou internet, facilitando a utilização de uma tecnologia de comunicação já presente na maioria das plantas industriais, e também os recursos visuais que simplificam o processo construtivo para o programador e auxiliam na leitura e codificação das informações por parte dos usuários. A interação entre hardware e software deu-se de maneira bem sucedida, com uma coleta e apresentação cíclica dos dados, sem lentidão ou travamentos. Dessa forma, pode-se afirmar que o sistema supervisor construído é eficaz para a amostragem em tempo real das variáveis monitoradas.

Com relação às pequenas oscilações nos valores das variáveis dos canais analógicos apresentados na interface, os testes mostraram que se trata de variações no sinal elétrico fruto da conversão ocorrida no amplificador operacional RCV420, indicando a necessidade de aprimoramentos no circuito eletrônico, como o incremento de filtros e uma blindagem eletrostática. Já no tocante à interface gráfica, é sabido que o Java possui recursos para a melhoria funcional e estética da mesma, exigindo apenas um nível maior de conhecimento por parte do programador. Esses tipos de melhorias tanto de software quanto de hardware configuram uma otimização do sistema supervisor que pode ser alvo de estudo em trabalhos futuros.

Por fim, nota-se que a evolução da microeletrônica possibilita o desenvolvimento de sistemas de aquisição e tratamento de dados mais robustos, softwares com interfaces gráficas amigáveis que facilitam o controle do usuário sobre o processo, tudo isso atrelado a um custo de desenvolvimento menor a cada dia, incentivando as pesquisas e assegurando vantagens competitivas no mercado para empresas que enxergam os benefícios das novas tecnologias.

## REFERÊNCIAS

- [1] BOYLESTAD, N. Dispositivos Eletrônicos e Teoria de Circuitos. 8. ed. São Paulo: Prentice Hall, 2004.
- [2] RIBEIRO, M. A. Fundamentos de Automação. 1. ed. Salvador: Tek T. & C. Ltda, 2003.
- [3] CAPELLI, A. Automação industrial: controle do movimento e processos contínuos. 2. ed. São Paulo: Érica, 2012.
- [4] SILVEIRA, P. R. DA; SANTOS, W. E. DOS. Automação e Controle Discreto. 9. ed. São Paulo: Érica, 2013.
- [5] SILVA, A. P. G.; SALVADOR, M. O que são sistemas supervisórios? Disponível em: <[http://www.wectrus.com.br/artigos/sist\\_superv.pdf](http://www.wectrus.com.br/artigos/sist_superv.pdf)>. Acesso em: 15 nov. 2016.
- [6] CASILLO, D. S. S. Automação e Controle: Sistemas Supervisórios. Disponível em: <[http://www2.ufersa.edu.br/portal/view/uploads/setores/166/arquivos/Automacao%20e%20Controle%202010\\_2/Automa%C3%A7%C3%A3o%20e%20Controle%20-%20Supervis%C3%B3rio.pdf](http://www2.ufersa.edu.br/portal/view/uploads/setores/166/arquivos/Automacao%20e%20Controle%202010_2/Automa%C3%A7%C3%A3o%20e%20Controle%20-%20Supervis%C3%B3rio.pdf)>. Acesso em: 21 mar. 2017.
- [7] Arduino - Home. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 6 fev. 2017.
- [8] DEITEL, P.; DEITEL, H. M. Java Como Programar. 8. ed. São Paulo: Pearson, 2010.
- [9] HORSTMANN, C. S.; CORNELL, G. CORE JAVA - Fundamentos. 8. ed. São Paulo: Pearson, v.1, 2009.
- [10] ExpressPCB - Home. Disponível em: <<https://www.expresspcb.com/>>. Acesso em: 20 fev. 2017.