



Universidade Federal de Ouro Preto - UFOP
Escola de Minas
Colegiado do curso de Engenharia de Controle e
Automação - CECAU



Diogo Luis de Moraes

Desenvolvimento de um Sistema de Controle e Medição de Vazão para Bombas Peristálticas

Ouro Preto, 2016

Diogo Luis de Moraes

Desenvolvimento de um Sistema de Controle e Medição de Vazão para Bombas Peristálticas

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientadora: Adrielle C. Santana

Coorientador: Diógenes Viegas

Ouro Preto, 2016

M827d Moraes, Diogo Luis.
Desenvolvimento de um sistema de controle e medição de vazão para bombas peristálticas [manuscrito] / Diogo Luis Moraes. - 2016.

71f.: il.: color; grafs; tabs; mapas; Códigos de Programação.

Orientador: Prof. MSc. Adrielle de Carvalho Santana.
Coorientador: Eng. Diógenes Viegas Mendes Ferreira.

Monografia (Graduação). Universidade Federal de Ouro Preto. Escola de Minas. Departamento de Engenharia de Controle e Automação e Técnicas Fundamentais.

1. Bombas de vácuo - Bombas peristálticas. 2. Vazão - Sistema de medição. 3. Medidores de fluxo. I. Santana, Adrielle de Carvalho. II. Ferreira, Diógenes Viegas Mendes. III. Universidade Federal de Ouro Preto. IV. Título.

CDU: 624

Catalogação: ficha@sisbin.ufop.br

Monografia defendida e aprovada, em 04 de agosto de 2016, pela comissão avaliadora constituída pelos professores:



Prof. M. Sc. Adrielle de Carvalho Santana Silva - Orientadora



Prof. Dr. Paulo Marcos de Barros Monteiro – Professor Convidado



Prof. Dr. Agnaldo José da Rocha Reis – Professor Convidado

AGRADECIMENTOS

Primeiramente gostaria de agradecer a Deus e a São Judas Tadeu por sempre iluminar o meu caminho.

Aos meus pais, Robson Luis e Agna dos Santos, pelo sacrifício para que eu pudesse chegar onde eu estou e dedicação em todos os momentos de minha vida. A minha irmã, Barbara Moraes por todo apoio e compreensão.

Aos meus familiares que sempre estiveram na torcida pelo meu sucesso e vivendo todas as minhas alegrias e tristezas.

A minha namorada, Jéssica Armendane, por toda paciência, carinho e companheirismo.

Aos meus orientadores, Adrielle de Carvalho e Diógenes Viegas, por todas as nossas conversas, por dedicarem e acreditar em meu projeto. A todos professores, em especial ao Paulo Monteiro e Agnaldo José, por toda a ajuda desde que eu ingressei na UFOP, agradeço pelos conselhos e oportunidades.

Aos meus amigos de Itabirito, Ouro Preto e Ouro Branco que sempre me apoiaram e me deram suporte quando precisei.

Agradeço também pela eterna família Lunáticos em Ouro Branco, onde eu pude amadurecer, respeitar e ganhar verdadeiros irmãos.

A Outotec Tecnologia e Fundação Gorceix, em especial aos integrantes da área de teste, Luan e Eron, e a todos companheiros do CTIII, pelo apoio e ensinamentos, tanto pessoal quanto profissionalmente.

“Nas grandes batalhas da vida, o primeiro passo para a vitória é o desejo de vencer”
(Mahatma Gandhi)

RESUMO

A vazão é uma variável importante em diversos processos industriais, como: alimentício, mineração e químico. Em muitos casos, o controle de vazão é um fator decisivo para uma boa produção. Desta maneira, aplicar controles automatizados de vazão é benéfico, pois reduz erros de controles realizados manualmente. Com esse princípio, nessa monografia é desenvolvido um controle de vazão para bombas peristálticas, utilizando uma plataforma Arduino e um sensor de vazão. Para o sensor, foram calibrados e testados duas opções: um medidor por placa de orifício e um sensor por efeito Hall HS05. O sistema funciona de forma que o usuário introduz o valor de vazão em um supervisório e o sensor de vazão realiza as leituras em tempo real dos valores e envia para o supervisório. Os objetivos desse trabalho foram estudar e desenvolver métodos para medição de vazão, sendo controlados e monitorados por uma interface homem – máquina. Foram realizados estudos com dois tipos de medidores de vazão: medidor por placa de orifício e medidor de fluxo por efeito hall. A vazão é determinada pelo usuário em um supervisório e ocorre a comunicação simultânea entre a bomba peristáltica, *shield* e Arduino. O estudo de medição de vazão para este projeto resultou na utilização do medidor de fluxo HS05 no controle de vazão para bombas peristálticas uma vez que o medidor por placa de orifício não apresentou leituras confiáveis para esse tipo de bomba. Por fim, conclui-se que o medidor utilizado foi capaz de realizar leituras precisas, resposta rápida, comprovando a justificativa e hipóteses deste projeto.

Palavras Chaves: Vazão, Bombas Peristáltica, Medidores de vazão, Arduino, Calibração.

ABSTRACT

The flow rate is an important variable in many industrial processes, such as food, mining and chemical industries. The flow control is a decisive factor for a good production. Apply automated controls flow is beneficial because it reduces errors controls performed manually. With this principle in this paper is developed a flow control for peristaltic pumps, using an Arduino platform and a flow sensor. For the sensor has been calibrated and tested two options: a meter of orifice plate and a Hall effect sensor HS05. The system works when the user enters the flow value in a supervisory and flow sensor takes readings in real-time values and sends it to the master. The objectives of this work were to study and develop methods for measuring flow, being controlled and monitored by a man - machine interface. Studies have been conducted with two types of flow meters: meter for orifice plate flow meter and hall effect. The flow is determined by the user in a supervisory and occurs simultaneous communication between the peristaltic pump, shield and Arduino. The flow measurement study for this project resulted in the use of HS05 flow meter in the flow control for peristaltic pumps since the meter by orifice plate did not provide reliable readings for this type of pump. Concludes that the used meter is able to perform accurate readings, fast response, proving the rationale and assumptions of this project.

Key – words: Flow, peristaltic pumps, flow meters, Arduino, Calibration.

LISTA DE FIGURAS

Figura 2.1 Placas de Orifício	8
Figura 2.2 Sensor HS05.....	10
Figura 2.3 Sensor de Pressão por membrana	12
Figura 2.4 Sensor de Pressão por Capacitância	12
Figura 2.5 Sensor de Pressão piezoelétrico	13
Figura 2.6 Figura ilustrativa da técnica de bombeio por pressurização.....	15
Figura 2.7 Bomba de seringa	16
Figura 2.8 Bomba Peristáltica MasterFlex.....	17
Figura 2.9 Vista interna do cabeçote de uma bomba peristáltica	17
Figura 2.10 - A Vista Posterior da Bomba. (A) Datasheet Interface. (B) interruptor seletor de Modalidade (LOCAL/REMOTE). (C) Alimentação. (D) Caixa de fusíveis.	18
Figura 2.11 Vista frontal da bomba. (A) Orifícios de montagem para caçote. (B) Molde de Asa para movimentação (C) Indicação e variação da rotação. (D) Regulador de Velocidade. (E) Indicador de Força. (F) Interruptor de Alimentação/ Sentido de Rotação	19
Figura 2.12 Conector DB9. (1) Entrada de controle de velocidade (2-10V). (2)Entrada de controle de velocidade (4-20mA). (3) Entrada de referência (Comum). (4) Controle Remoto/Local (+). (5) Controle Remoto/Local (-). (6) Controle Start/Stop (+). (7) Controle Start/Stop. (8)(9) Terra	19
Figura 2.13 Montagem de controle por tensão.....	20
Figura 2.14 Montagem de controle por corrente	20
Figura 2.15 Fluxograma do Processo.....	21
Figura 2.16 Ilustração do ambiente de Trabalho C#.	22
Figura 2.17 Descrição da placa Arduino Uno	23
Figura 2.18 IDE do Arduino	24
Figura 2.19 Conversão analógico/digital de um sinal senoidal.....	27
Figura 2.20 Filtro passa baixas	28
Figura 2.21 Amplificador Inversor com realimentação negativa.....	29
Figura 2.22 Amplificador Não Inversor	30
Figura 3.1 Sinal PWM do Arduino	31
Figura 3.2 Sinal convertido pelo Filtro passa-baixa.....	32

Figura 3.3 Circuito dos pinos - LM358.....	32
Figura 3.4 Montagem do circuito amplificador.....	33
Figura 3.5 Calibração da bomba	34
Figura 3.6 Esquema de montagem para calibração.....	34
Figura 3.7 Curva de calibração	35
Figura 3.8 Sensor de Pressão MPX5700	36
Figura 3.9 Protótipo do medidor de vazão	36
Figura 3.10 Curva de Resposta - Sensor de vazão.....	38
Figura 3.11 Curva de Erros - Sensor de Vazão.....	39
Figura 3.12. Circuito da placa no Proteus	40
Figura 3.13 Desenho da Shield	41
Figura 3.14. Processo de soldagem da placa	42
Figura 3.15. <i>Shield</i> com sensores conectados.....	42
Figura 3.16. Tela inicial do Supervisório	43
Figura 3.17. Tela de Configuração Serial.....	44
Figura 3.1 Oscilação de resposta com baixa vazão	44
Figura 3.2 Oscilação de resposta com máxima vazão	45
Figura 4.1. Vista posterior da bomba acoplada com a <i>shield</i> . (1) Conector Fêmea DB9. (2) Conector Fêmea DB9. (3) Sensor de Fluxo. (4) Bomba Peristáltica. (5) Arduino. (6) <i>Shield</i>	47
Figura 4.2. Montagem da bomba acoplada com <i>shield</i> e sensor	47
Figura 4.3. Medição com vazão com 1000 ml/min	48
Figura 4.4. Medição de vazão com 600 ml/min	48
Figura 4.5. Medição com vazão de 400 ml/min	49

SUMÁRIO

1. INTRODUÇÃO	1
1.1 OBJETIVO.....	2
1.1.1 Objetivo Geral	2
1.1.2 Objetivos Específicos	2
1.2 JUSTIFICATIVA DO TRABALHO.....	2
1.3 METODOLOGIA PROPOSTA.....	2
1.4 ESTRUTURA DO TRABALHO.....	3
2. REVISÃO DE LITERATURA	5
2.1 VAZÃO.....	5
2.1.1 Calor Específico	5
2.1.2 Viscosidade	5
2.1.3 Número de Reynolds	6
2.2 MEDIDORES DE VAZÃO.....	6
2.3 MEDIÇÃO DE VAZÃO.....	7
2.3.1 Placa de Orifício	7
2.3.2 Medidor de Vazão com sensor de Efeito Hall – HS05.....	10
2.4 PRESSÃO.....	11
2.4.1 Sensores de Pressão	11
2.4.2 Exemplos de Sensores de Pressão	11
2.5 BOMBAS DE FLUXO CONSTANTE.....	14
2.5.1 Frasco Pressurizado	14
2.5.2 Bomba de Seringa ou de Pistão	15
2.5.3 Bomba Peristáltica	16
2.6 FLUXOGRAMA DE FUNCIONAMENTO.....	20
2.7 SUPERVISÓRIO.....	21
2.7.1 Visual Studio	21
2.7.2 Linguagem C#	21
2.8 ARDUINO.....	22
2.8.1 Sinais Digitais e Analógicos	25
2.9 CONVERSÃO DE SINAL DIGITAL PWM PARA ANALÓGICO.....	26
2.10 FILTRO PASSA-BAIXAS.....	27

2.11	AMPLIFICADORES OPERACIONAIS.....	28
3.	DESENVOLVIMENTO	31
3.1	CONVERSÃO DO SINAL	31
3.2.1	FILTRO PASSA - BAIXA	31
3.2	CALIBRAÇÃO DA BOMBA PELO CONTROLADOR DE VAZÃO.....	33
3.3	DESENVOLVIMENTO DO MEDIDOR DE VAZÃO POR PLACA DE ORIFÍCIO	35
3.4	CALIBRAÇÃO MEDIDOR DE VAZÃO POR EFEITO HALL	37
3.5	PROJETO E MONTAGEM DA PLACA.....	40
3.6	SUPERVISÓRIO	42
4.	RESULTADOS E DISCUSSÃO	44
4.1	SENSOR DE VAZÃO COM PLACA DE ORIFÍCIO	44
4.2	SENSOR DE FLUXO POR EFEITO HALL.....	46
5.	CONCLUSÕES.....	50
5.1	TRABALHOS FUTUROS	51
	REFERÊNCIAS BIBLIOGRÁFICAS.....	52
	APÊNDICE A	54
	APÊNDICE B	56

1. INTRODUÇÃO

Cada vez mais é comum ouvir sobre a precisão de medidas, relacionadas a diversos setores, como farmacêutico, químico, alimentício, minerações e bebidas. A precisão nas medidas é um fator importante para atender as necessidades da qualidade

A automação industrial juntamente com a Instrumentação está crescendo muito rapidamente devido ao aprimoramento das tecnologias, à necessidade de trabalho e a adaptação dos funcionários. Com isso, tende-se a melhorar os resultados, a qualidade, a eficiência, reduzir o tempo de processo e oferecer flexibilidade.

A automação de bombas para o controle de vazão é de grande importância em um processo, pois é preciso ter medidas precisas, calibradas e com rápido tempo de resposta, para beneficiar o processo industrial. A medição de vazão de fluidos está presente no dia a dia de muitas pessoas. Por exemplo, o hidrômetro de uma residência, o marcador de uma bomba de combustível nos veículos, etc.

Na História, grandes nomes marcaram suas contribuições. Em 1502 Leonardo da Vinci observou que a quantidade de água por unidade de tempo que escoava em um rio era a mesma em qualquer parte, independente da largura, profundidade, inclinação e outros. Mas o desenvolvimento de dispositivos práticos só foi possível com o surgimento da era industrial e o trabalho de pesquisadores como Bernoulli, Pitot e outros (CASSIOLATA e ALVES, 2006).

Os sistemas automatizados são frequentemente encontrados nos ramos industriais, desde simples a complexos controles automatizados. Esses são baseados em uma interação entre *software* e *hardware*. Podemos citar softwares como os ambientes de desenvolvimento de programação tais como RSTLogix 5000 e a IDE Arduino e softwares para a criação de Interfaces

Homem-Máquina, tal como o Visual Basic C#. Para os hardwares pode-se citar os microcontroladores, controladores e circuitos integrados (CIs) diversos.

Com base no disposto anteriormente foi elaborado um projeto, desenvolvendo um trabalho teórico e prático com aplicação real juntamente com o estágio na empresa Outotec Tecnologia Brazil, no laboratório de testes na Fundação Gorceix, Ouro Preto – MG; buscando otimizar o funcionamento das bombas peristálticas nos testes de equipamentos de mineração.

1.1 Objetivo

1.1.1 Objetivo Geral

Desenvolver um sistema de controle e medição de vazão para bombas peristálticas com uma interface homem-máquina integrando *hardware e software*.

1.1.2 Objetivos Específicos

Possibilitar o controle de uma bomba por meio de uma interface homem-máquina. Aliar os estudos das disciplinas cursadas durante a graduação, como Mecânica dos Fluidos, Sistemas de Medição, Eletrônica, Instrumentação, Interfaceamento de Sistemas e construção e calibração do sensor de vazão.

1.2 Justificativa do Trabalho

A grande motivação para o desenvolvimento deste projeto está relacionado à necessidade de um sistema de controle e medição da vazão de uma bomba peristáltica no estágio realizado pelo aluno; sendo um projeto que irá reduzir tempo, custo e aumentar a precisão e qualidade na utilização destas bombas.

1.3 Metodologia Proposta

É proposta uma melhoria nos testes de espessamento, no Laboratório da Fundação Gorceix, em específico controle automatizado das bombas

peristálticas via comunicação serial com um microcontrolador. Englobando as etapas da construção da interface e do controle com a parte física, além de atestar o funcionamento do mesmo. Têm-se, portanto um projeto fundamentado nas seguintes etapas:

- Estudo sobre funcionamento/comunicação das Bombas;
- Estudo sobre os equipamentos que auxiliam o processo;
- Seleção de componentes (Resistores, Amplificadores, Capacitores);
- Montagem da *Shield*;
- Calibração da Bomba;
- Calibração do Sensor;
- Desenvolvimento de interface de comunicação para acionamento e monitoramento dos equipamentos;
- Implementação;
- Realização dos testes em bancada;
- Discussão e análise dos resultados.

1.4 Estrutura do Trabalho

Este trabalho foi dividido em cinco partes: introdução, base teórica, desenvolvimento, resultados e conclusões.

Em sua primeira parte, tem-se uma introdução ao tema, juntamente com os objetivos e justificativa.

Na segunda parte é realizado uma revisão teórica dos componentes, equipamentos, instrumentos que são necessários para o entendimento e desenvolvimento do projeto.

A terceira parte é o desenvolvimento, especificando as calibrações dos sensores, calibração da bomba, desenvolvimento da *shield*, comunicação entre o microcontrolador, bomba peristáltica e Interface.

Na quarta parte são feitos testes e então analisados os resultados do projeto.

Na quinta parte, são explicitadas as conclusões sobre o projeto e discutimos possíveis projetos futuros que podem ser desenvolvidos utilizando, de certa maneira, este como base.

2. REVISÃO DE LITERATURA

Na maioria das operações realizadas nos processos industriais é muito importante efetuar a medição e o controle de fluxo de líquidos, gases e até sólidos granulados, não só para fins contábeis, como também para a verificação do rendimento do processo

2.1 Vazão

A vazão é a terceira unidade de grandeza mais medidas nos processos e possui diversas aplicações desde as simples medições de vazão de água em tratamentos de residências até nas grandes indústrias, com fluidos diferentes de água, como gases, fluidos corrosivos e químicos. Deve-se levar em conta alguns fatores como a precisão da medição, tipo de fluido, condições de temperatura e do ambiente e espaço físico disponível. (CASSIOLATA e ALVES, 2006)

Alguns conceitos relativos a fluidos devem ser entendidos e estabelecidos, pois são fundamentais na vazão, conforme será visto a seguir.

2.1.1 Calor Específico

Define-se calor específico como o quociente da quantidade infinitesimal de calor fornecido a uma unidade de massa de uma substância pela variação infinitesimal de temperatura resultante deste aquecimento. Na prática, temos: A quantidade de calor necessária para mudar a temperatura de 1 grama de uma substância em 1°C. (MUNSON e col,2004)

2.1.2 Viscosidade

É definida como sendo a resistência ao escoamento de um fluido em um duto qualquer. Esta resistência provocará uma perda de carga adicional que deverá ser considerada na medição de vazão. (MUNSON e col,2004)

2.1.3 Número de Reynolds

Número adimensional utilizado para determinar se o escoamento se processa em regime laminar ou turbulento. Sua determinação é importante como parâmetro modificador do coeficiente de descarga. (MUNSON e col, 2004)

2.2 Medidores de Vazão

Medidores de fluxo ou vazão são equipamentos para se obter a medida de um fluxo volumétrico ou gravimétrico. Essas medidas podem ser realizadas, podem ser realizadas de diversas maneiras com princípios físicos variados.

Atualmente, estes medidores têm uma grande importância em um processo, pois são decisivos para garantir que específicos fluxos são fornecidos a uma taxa adequada durante o processo, resultando em uma maior confiabilidade no produto final. Com as mais variadas aplicações, existem diversos tipos de medidores de vazão, que, de forma sucinta, podem ser classificados em dois grandes grupos funcionais (CASSIOLATA e ALVES, 2006):

- Medição da quantidade de fluido que passa pela pelo medidor em um determinado espaço de tempo, por meio da medição direta de volume.
- Medição da vazão, ou seja, a taxa na qual o fluido escoo pelo medidor, por meio da inferência da interação deste com o fluido.

Existem alguns critérios importantes que se levam em conta na hora da escolha dos medidores, tais como: analisar a vazão operacional, as características do fluido, características da operação e instalação, custo e exatidão.

Segundo Delmeé (2003),

“Os medidores são constituídos de partes distintas com funções bem específicas que são elementos primário e secundário. O Elemento Primário, é aquela que está em contato com o fluido. Esse contato resulta em alguma forma de interação que pode ser, por exemplo, o deslocamento do Elemento Primário ou a aceleração ou desaceleração do fluido. A segunda parte é o Elemento Secundário, que transforma a interação medida entre o

elemento primário e o fluido em um sinal que pode ser convertido em volume, massa ou vazão. Além disso, o elemento secundário normalmente indica e/ou armazena esses valores. ”

2.3 Medição de Vazão

Cita-se a seguir os principais métodos de se realizar a medição de vazão de fluidos e os principais sensores relacionados a cada método:

- Medição por Pressão Diferencial: Placa de Orifício, Tubo Venturi, Tubo Pitot, Bocal;
- Medição por Área: Variável Rotâmetro;
- Medição através de Velocidade: Turbina;
- Medição através de Força: Placa de Impacto;
- Medição por Tensão Induzida: Medidor Magnético;
- Medição em Canais Abertos: Calha Parshall Vertedores;

2.3.1 Placa de Orifício

A primeira aplicação da medição de vazão com placa de orifício ou outro gerador de pressão diferencial se perde na Antiguidade. Antes da era cristã, os romanos já usavam a placa de orifício para a medição de vazão de água de consumo. É considerado o mais simples e mais usado método, consistindo em uma placa perfurada a qual é instalada perpendicularmente ao eixo da tubulação. É essencial que as bordas do orifício estejam sempre perfeitas, porque, se ficarem, imprecisas ou corroídas pelo fluido, a precisão da medição será comprometida. Estas placas podem ser feitas com aço inox, monel ou latão, de acordo com as especificações do fluido. (FERREIRA, 2007).

As placas podem ser de diferentes tipos a depender da forma e localização do orifício.

- Orifício concêntrico: Este tipo de placa é utilizado para líquidos, gases e vapor que não contenham sólidos em suspensão.
- Orifício Excêntrico: Utilizada quando se tem fluido com sólidos em suspensão, os quais possam ser retidos e acumulados na base da placa, sendo o orifício posicionado na parte de baixo do tubo.

- Orifício segmental: Esta placa tem a abertura para passagem de fluido, disposta em forma de segmento de círculo. É destinada para uso em fluídos laminados e com alta porcentagem de sólidos em suspensão

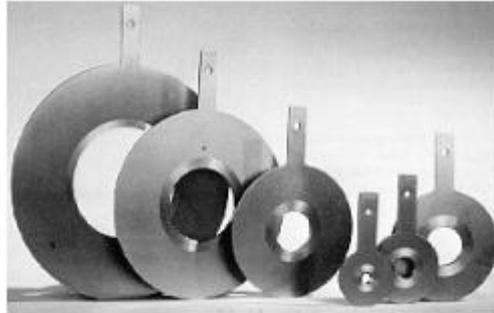


Figura 2.1 Placas de Orifício

Fonte: RIBEIRO, 2002.

À medida que o fluido se aproxima da placa há um ligeiro aumento na pressão, e depois há uma súbita queda após a passagem pelo orifício. A pressão continua a cair até atingir um ponto de pressão mínima chamado “vena contracta”. É quando a pressão começa a subir novamente até atingir a pressão máxima após a placa. Essa pressão será sempre menor que a pressão antes da placa. (FERREIRA, 2007).

Quando passa pelo orifício, a redução da pressão é resultado do aumento em sua velocidade passando pela área reduzida. O fluido escoando através da área da tubulação tende a querer passar pelo orifício que possui área menor. Dessa forma o fluido é “acelerado”, o que causa a queda de pressão. (FERREIRA, 2007).

Essa relação pode ser representada de forma simplificada pela equação 2.1:

$$Q = V * A \quad (2.1)$$

onde,

Q= vazão (m³/s)

V= Velocidade (m²/s)

A = Área (m²)

A equação para determinar vazão com placa de orifício é derivada da equação de Bernoulli, dada em 2.2.

$$\frac{P_1}{\rho g} + \frac{1}{2g} v_1^2 = \frac{P_3}{\rho g} + \frac{1}{2g} v_3^2 + \Sigma f \quad (2.2)$$

onde,

P= Pressões em cada tomada (Pa);

V1= Velocidade dos fluidos antes da placa (m³/s);

V3= Velocidade dos fluidos depois da placa (m³/s);

g = aceleração da gravidade (m/s²);

ρ = densidade do fluido;

f = representa as perdas totais, (atrito, turbulência);

Esta equação pode ser simplificada e reescrita determinando a equação geral da vazão para placa de orifício, da seguinte forma:

$$F_1 = A_1 V_1 = C_{meter} Y A_3 \sqrt{\frac{2(P_1 - P_3)}{\rho(1 - A_3^2 / A_1^2)}} \quad (2.3)$$

O coeficiente do medidor ou coeficiente de escape “Cmeter”, determinado experimentalmente, se encarrega de incluir na equação o comportamento do medidor real, tais como perdas por atrito e geometria, a relação entre as áreas das seções transversais da tubulação e do orifício (também conhecida como β) e as características de viscosidade do fluido (número de Reynolds). O fator Y está relacionado à compressibilidade do fluido, sendo 1 para fluidos incompressíveis. Quando o processo está em regime, todos os parâmetros do medidor e do fluido são então conhecidos. Logo, basta medir a pressão diferencial entre as tomadas. (MARTINS, 1998).

2.3.2 Medidor de Vazão com sensor de Efeito Hall – HS05

O sensor de fluxo de água é um dispositivo para realizar medições de vazão, para diversas aplicações. O sensor HS05 é um sensor de alta precisão que trabalha com uma turbina. No seu interior existe um tipo de hélice que gira com a força da água. Junto a esta hélice temos fixado um ímã, e paralelamente ao ímã tem-se um sensor de efeito Hall, esse sensor detecta cada volta dada pelo ímã e sua estrutura é constituída de nylon. (NEVES, 2015).



Figura 2.2 Sensor HS05

Um Sensor de Efeito Hall é um transdutor que quando é aplicado a um campo magnético, responde com uma variação em sua tensão de saída. Efeito Hall é o que ocorre quando uma corrente num condutor tem sua trajetória desviada pela ação de um campo magnético. O desvio faz com que a Tensão de Hall seja gerada e comunicada para um circuito externo, e é o que sensores Hall fazem. A tensão de Hall pode ser medida por um circuito externo ou ainda ser utilizada para efeitos de sensoriamento, pois ela é proporcional à intensidade do campo que cria. (CASSIOLATO, 2006).

2.4 Pressão

Um elemento fundamental no medidor de placa de orifício é o medidor de pressão diferencial, com o princípio de funcionamento de simultaneamente ler a diferença de pressão nas tomadas de alta (antes da placa de orifício) e baixa (depois da placa de orifício) e ainda calcular a vazão correspondente.

2.4.1 Sensores de Pressão

Diversos processos industriais têm a necessidade de realizar a medição de pressão como aqueles relacionados com a distribuição de fluidos (líquidos e gases). Sensores eletrônicos de pressão são de grande importância atualmente, uma vez que eles podem fornecer sinais elétricos diretamente para circuitos de controle ou de monitoramento. O conhecimento do modo como esses sensores funcionam e do seu uso, portanto, são de grande importância. (BRAGA, 2007).

A unidade de pressão no Sistema Internacional de Unidades (SI) é o Pascal, definido como 1 newton por metro quadrado (N/m^2). No entanto, existem outras unidades que podem ser usadas em alguns casos, como libras por polegada quadrada (PSI), atmosferas (atm), bar, polegadas de mercúrio (in Hg) e milímetros de mercúrio (mm Hg).

A medida de pressão pode ser tanto estática como dinâmica. Quando não existe movimento do fluido cuja pressão está sendo medida, no interior de um reservatório, por exemplo, a medida da pressão é estática. Quando o fluido está em movimento, escapando de um reservatório, por exemplo, a medida é dinâmica. (BRAGA, 2007).

2.4.2 Exemplos de Sensores de Pressão

Os três principais tipos de sensores que são usados nas aplicações práticas na data em que este trabalho foi produzido, são comentados a seguir.

2.4.2.1 Membrana

Os mais comuns de todos os sensores de pressão são os de membrana com configuração em ponte de Wheatstone. Quando uma pressão é aplicada ao diafragma, onde está acoplado um sensor piezoresistivo onde acontece a variação e é medido por um circuito externo.

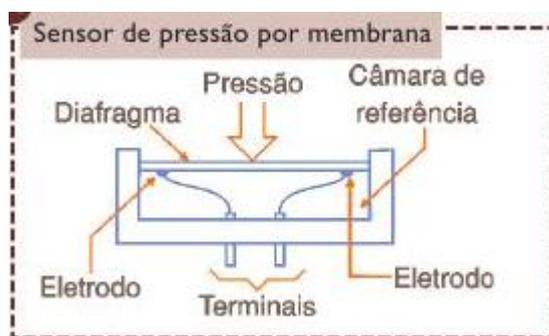


Figura 2.3 Sensor de Pressão por membrana

Fonte: BRAGA, 2007

2.4.2.2 Capacitância

Este sensor se baseia na alteração da capacitância entre um diafragma que se deforma, e uma placa fixa no seu interior.

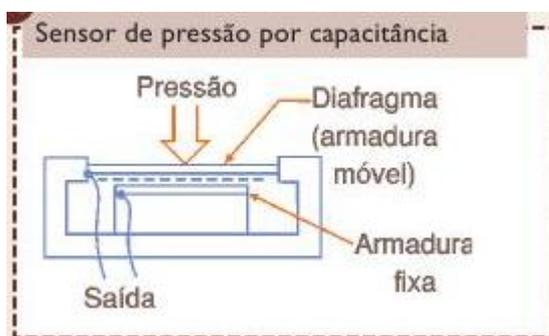


Figura 2.4 Sensor de Pressão por Capacitância

Fonte: BRAGA, 2007

Com o aumento da pressão aplicada ao diafragma, a armadura móvel que ele representa se aproxima da armadura fixa e, com isso, a capacitância do sensor aumenta.

Apesar de serem sensíveis e terem uma resposta linear estável, esses sensores também são sensíveis à alta temperatura e exigem circuitos mais complexos para processar o sinal que fornecem, ou seja, a capacitância.

2.4.2.3 Piezoelétricos

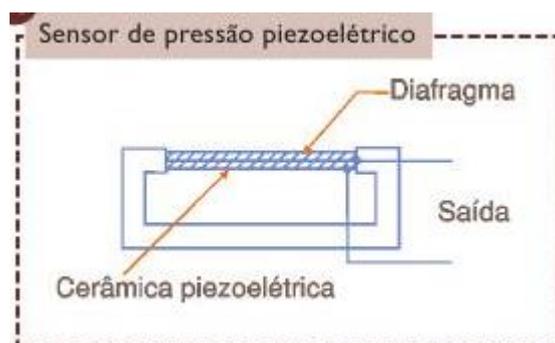


Figura 2.5 Sensor de Pressão piezoelétrico

Fonte: BRAGA, 2007

Este sensor tem o funcionamento através de contato de uma agulha com um cristal piezoelétrico. Então, quando se muda a pressão da agulha sobre o cristal, ele se deforma, gerando uma tensão elétrica que se manifesta na sua saída. Essa tensão é proporcional à pressão da agulha sobre o elemento sensor.

2.5 Bombas de fluxo constante

Segundo Macintyre (2012),

“Bombas são máquinas geratrizes cuja finalidade é realizar o deslocamento de um líquido por escoamento. Sendo uma máquina geratriz, ela transforma o trabalho mecânico que recebe para seu funcionamento em energia, que é comunicada ao líquido sob as formas de energia de pressão e cinética”.

Uma bomba de fluxo constante, independente da rotação da bomba, força uma quantidade fixa ou invariável de fluido através do orifício de saída durante cada revolução da bomba. Bombas de fluxo constante são, algumas vezes, chamadas bombas de volume constante ou de fluxo fixo. Elas entregam uma quantidade fixa de fluido por revolução, independente da pressão solicitada.

Desde que a bomba de fluxo constante proporcione uma quantidade fixa de fluido durante cada revolução da bomba, a quantidade de fluido entregue por minuto dependerá da rotação da bomba. Quando uma bomba de fluxo constante é usada em um sistema hidráulico, na qual a pressão deve ser mantida em um valor constante, um regulador de pressão é necessário.

Para sistemas de análise em fluxo existem basicamente três tipos de sistemas para o bombeamento de líquidos que oferecem fluxo constante (sem pulsos): Frasco pressurizado, Bomba de seringa e Bomba peristáltica.

2.5.1 Frasco Pressurizado

O princípio de bombeamento com frasco pressurizado consiste na injeção de um “gás inerte” em um frasco contendo o líquido que se deseja bombear, forçando a saída do líquido pelo aumento da pressão interna.

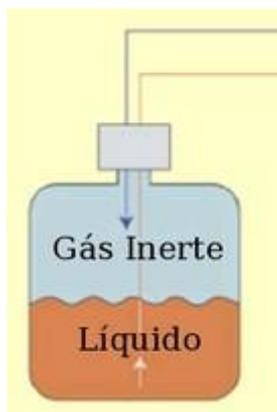


Figura 2.6 Figura ilustrativa da técnica de bombeio por pressurização.

Fonte: (MARKOS, 2013)

2.5.2 Bomba de Seringa ou de Pistão

Bombas com mecanismo de infusão com seringa ou bombas de seringa utilizam uma seringa como reservatório do líquido e garantem precisão elevada e fluxo contínuo para pequenos volumes (menores que 100ml). Por infundirem soluções com precisão elevada e baixo fluxo, são particularmente apropriadas para aplicações pediátricas e terapia intensiva, onde volumes pequenos de medicamentos com concentração elevada devem ser infundidos por um longo período de tempo.

Neste mecanismo a rotação do motor de passo é transmitida a um fuso (rosca sem fim) que movimenta o êmbolo da seringa. Normalmente uma mola ou um mecanismo a gás é utilizado para empurrar o êmbolo com força constante, criando uma pressão de infusão constante.



Figura 2.7 Bomba de seringa

Fonte: (MARKOS, 2013)

2.5.3 Bomba Peristáltica

A bomba peristáltica é um projeto inspirado na “*peristalsis*”, o qual é um movimento natural do sistema digestivo, responsável pelo deslocamento do alimento dentro do intestino. É um princípio extremamente simples: um mangote posicionado no curso do rotor é alternadamente comprimido e aliviado por sapatas nas extremidades pela rotação do mesmo, o que gera uma sucção negativa e conseqüente transferência de fluido dentro desse mangote.

Tal princípio de bombeamento confere grande força de sucção, vencendo resistências e eliminando o risco de “*slip*” de produto. Assim, as bombas peristálticas apresentam funcionamento superior na transferência de fluidos viscosos, com alto teor de fibras e sólidos, e abrasivos, sendo largamente utilizada em diversos setores da indústria. Como a quantidade de fluido bombeado por rotação da bomba é constante, também funcionam como excelentes bombas dosadoras.

Outro ponto importante é a ausência de contaminação, pois o projeto dessas bombas implica que apenas o material da mangueira ou mangote entra em contato com o fluido bombeado. Desse modo, difundiu-se o uso de bomba peristáltica de mangueira em aplicações de laboratório.

A bomba MasterFlex possui dois tipos de acionamentos para variação da vazão, controle remoto e controle local. O local é a utilização do potenciômetro instalado na parte frontal para a variação da bomba. O remoto é com a utilização de um circuito independente, ou seja, na parte traseira da bomba existe uma conexão onde recebe valores de tensão, variando entre 2 e 10 V, e corrente, variando entre 4 e 20 mA. Como mostrado nas figuras 2.13 e 2.14.



Figura 2.8 Bomba Peristáltica MasterFlex



Figura 2.9 Vista interna do cabeçote de uma bomba peristáltica

Fonte: MARKOS, 2013

Uma bomba peristáltica é formada pelos seguintes componentes ilustrados nas figuras 2.8 a 2.13.

1. Cabeçote para guiar o tubo flexível;
2. Tubo flexível;
3. Roletes montados no braço porta-rolete, que pressionam o tubo flexível contra o cabeçote;
4. Eixo central da bomba, que gira o braço porta-rolete;
5. Conexão para o tubo flexível (entrada ou saída);
6. Conexão para o tubo flexível (entrada ou saída);

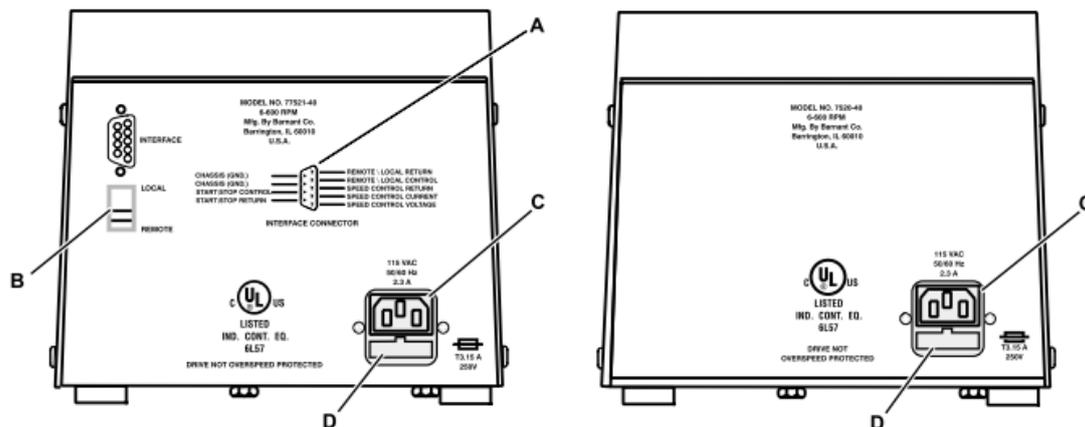


Figura 2.10 - A Vista Posterior da Bomba. (A) Datasheet Interface. (B) interruptor seletor de Modalidade (LOCAL/REMOTE). (C) Alimentação. (D) Caixa de fusíveis.

Fonte: MASTERFLEX, 2000

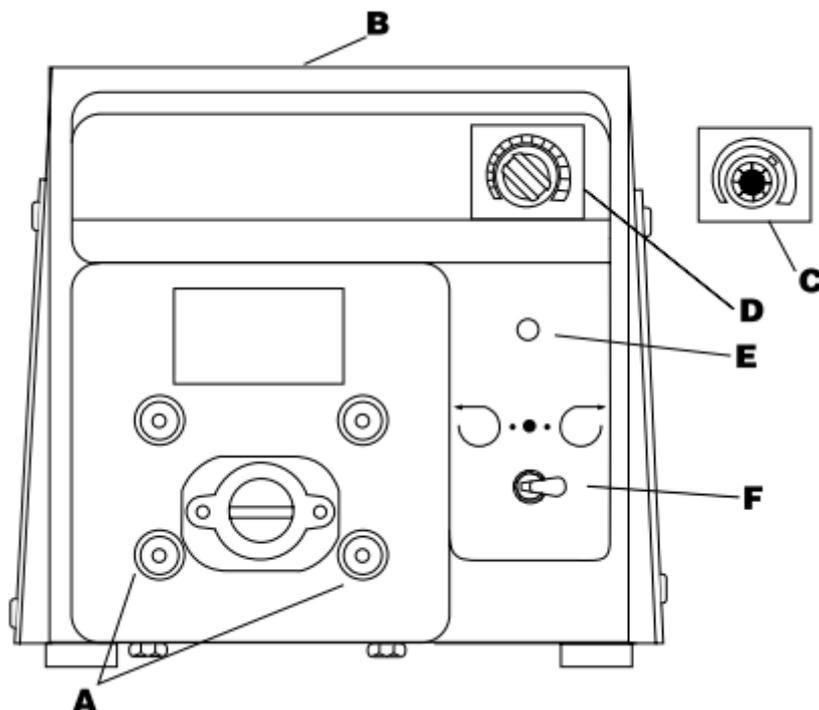


Figura 2.11 Vista frontal da bomba. (A) Orifícios de montagem para caçote. (B) Molde de Asa para movimentação (C) Indicação e variação da rotação. (D) Regulador de Velocidade. (E) Indicador de Força. (F) Interruptor de Alimentação/ Sentido de Rotação

Fonte: MASTERFLEX, 2000

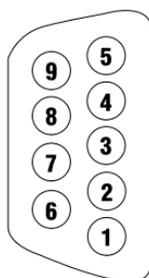


Figura 2.12 Conector DB9. (1) Entrada de controle de velocidade (2-10V). (2)Entrada de controle de velocidade (4-20mA). (3) Entrada de referência (Comum). (4) Controle Remoto/Local (+). (5) Controle Remoto/Local (-). (6) Controle Start/Stop (+). (7) Controle Start/Stop. (8)(9) Terra

Fonte: MASTERFLEX, 2000

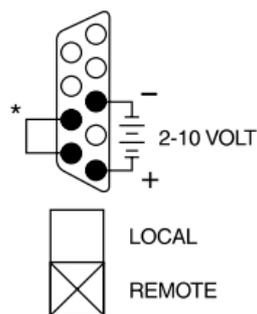


Figura 2.13 Montagem de controle por tensão

Fonte: MASTERFLEX, 2000

EXAMPLE 2

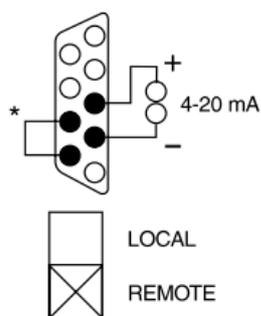


Figura 2.14 Montagem de controle por corrente

Fonte: MASTERFLEX, 2000

2.6 Fluxograma de funcionamento

O controle da bomba desse projeto se baseia em diferentes etapas, conforme serão citados nos itens subsequentes. O primeiro passo é o usuário comunicar com a porta serial e entrar com o valor de vazão, no sistema supervisor, que será enviado via comunicação serial para o microcontrolador. Este recebe o sinal e dá uma saída conforme sua equação de calibração. Este valor sairá de acordo com a largura de pulsos PWM e em seguida para o filtro passa – baixa para converter de sinal PWM para analógico. Após isso, o sinal de saída do filtro passa – baixa é ligado no amplificador operacional, de acordo com a figura 3.1. Sendo assim, o sinal de saída amplificado é ligado na bomba. Com o funcionamento paralelo, o sensor de vazão irá mandar para o Arduino os

valores lidos de vazão. Neste circuito, o Arduino comunica com o supervisor onde é mostrado o valor real de vazão para o usuário.

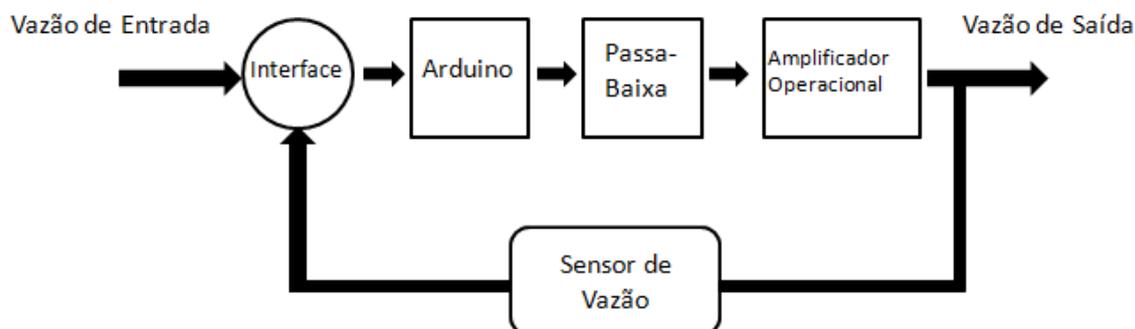


Figura 2.15 Fluxograma do Processo

2.7 Supervisório

2.7.1 Visual Studio

O Visual Studio é um ambiente de desenvolvimento integrado e sofisticado para criação de aplicativos impressionantes para Windows, Android e iOS, aplicativos Web modernos e serviços de nuvem. Elaborado para desenvolvedores individuais construírem aplicações com multicamadas de alto desempenho. Possui ferramentas e serviços para projetos de qualquer tamanho ou complexidade e também operações de desenvolvimento com implantações automatizadas e monitoramento contínuo (VISUAL STUDIO, 2015).

2.7.2 Linguagem C#

C# (CSharp) é uma linguagem de programação orientada a objetos desenvolvida no programa Microsoft Visual Studio, que faz parte da sua plataforma .NET. A companhia baseou C# na linguagem C++ e Java. (VISUAL STUDIO, 2015). A figura 2.15 ilustra a IDE do C#.

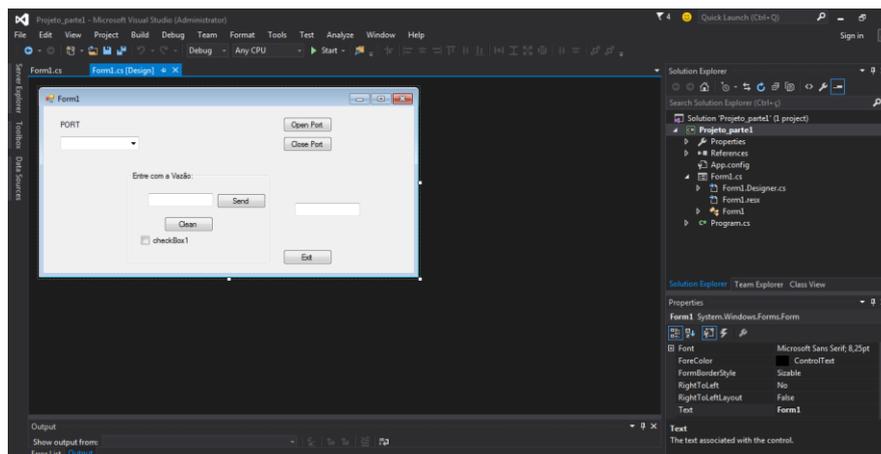


Figura 2.16 Ilustração do ambiente de Trabalho C#.

2.8 Arduino

O Arduino UNO é um microcontrolador que constitui uma plataforma de prototipagem eletrônica baseada em *hardware* e *software*. Pela sua vasta adaptação e aplicações pode ser executado em *Linux*, *Windows* e *Macintosh*.

Outra de suas características é ser *open source*, ou seja, possui código fonte aberto e a placa pode ser adaptada e modificada de acordo com a aplicação.

A placa, ilustrada na figura 2.15 é composta por um microcontrolador e cristal oscilador 16 MHz, possuindo uma entrada USB, uma entrada para uma fonte de alimentação externa, pinagem para controle de sinais de *inputs* e *outputs*, pinagem de terra e +Vcc, assim como *inputs* analógicos.

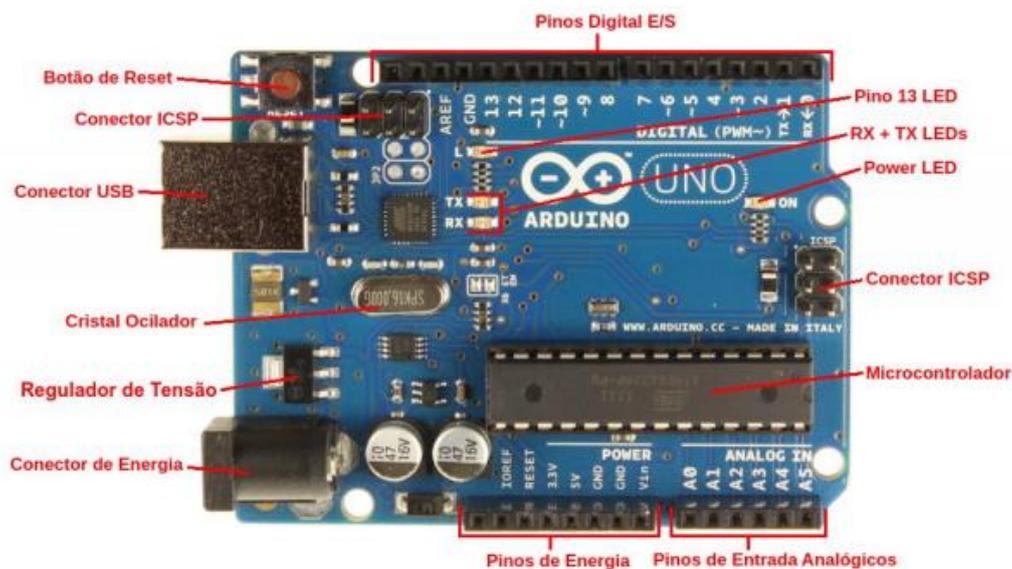


Figura 2.17 Descrição da placa Arduino Uno

Fonte: OLIVEIRA, 2013

O ambiente de desenvolvimento (IDE) de programas do arduino, ilustrado na figura 2.16 é uma IDE que trabalha com criação de *sketches*. A linguagem de programação utilizada é *Wiring* e o código escrito é traduzido para a linguagem C. O ciclo de utilização se dá da seguinte forma:

1. Conexão da placa a uma porta USB do computador;
2. Desenvolvimento de um *sketch* com comandos para a placa;
3. Upload do *sketch* para a placa, utilizando a comunicação USB.
4. Aguardar a reinicialização, após ocorrerá à execução do *sketch* criado.

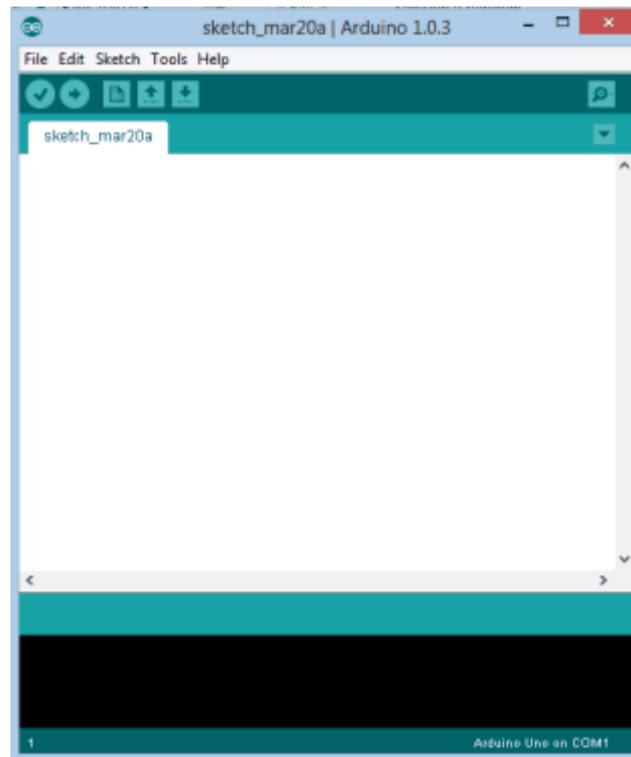


Figura 2.18 IDE do Arduino

Na *Toolbar* há uma guia, ou um conjunto de guias, com o nome do *sketch*. Ao lado direito há um botão que habilita o serial monitor. No topo há uma barra de menus, com os itens File, Edit, Sketch, Tools e Help. Os botões na *Toolbar* fornecem acesso rápido às funções mais utilizadas dentro desses menus.

- Abaixo são identificados os ícones de atalho da IDE:
- *Verify*
 - Verifica se existe erro no código digitado.
- *Upload*
 - Compila o código e grava na placa Arduino se corretamente conectada;
- *New*
 - Cria um novo *sketch* em branco.
- *Open*
 - Abre um *sketch*, presente no sketchbook.
- *Save*

- Salva o *sketch* ativo
- *Seria monitor*
 - Abre o monitor serial.

2.8.1 Sinais Digitais e Analógicos

Um fenômeno natural, por exemplo, eletricidade ou som, são analógicos, ou seja, ao observar estes processos físicos têm-se um sinal analógico. Uma das propriedades importantes destes sinais analógicos é que eles são contínuos no tempo. Para cada ponto no tempo, você pode medir a força do sinal, e, em princípio, você pode registrar até mesmo a variação menor do sinal (SCHMIDT, 2011).

Mas apesar do mundo ser analógico, também prevalece a era digital. Quando os primeiros computadores foram construídos há algumas décadas atrás, as pessoas rapidamente perceberam que é muito mais fácil trabalhar com informações do mundo real quando ele é representado como números e não como um sinal analógico, como tensão ou volume. Para exemplo, é muito mais fácil de manipular sons usando um computador quando as ondas sonoras são armazenadas como uma sequência de números (SCHMIDT, 2011).

O Arduino executa automaticamente a conversão de analógico para digital e vice-versa. (SCHMIDT, 2011).

O Arduino UNO possui 14 pinos que podem ser usados como entrada ou saída digitais. Estes pinos operam em 5 V, onde cada pino pode fornecer ou receber uma corrente máxima de 40 mA. Cada pino possui resistor de pull-up interno que pode ser habilitado por software. Alguns desses pinos possuem funções especiais: (SOUZA, 2013):

- PWM: 3,5,6,9,10 e 11 podem ser usados como saídas PWM de 8 bits por meio da função *analogWrite()*;
- Comunicação serial: 0 e 1 podem ser utilizados para comunicação serial. Deve-se observar que estes pinos são ligados ao microcontrolador responsável pela comunicação USB com o PC;
- Interrupção externa: 2 e 3. Estes pinos podem ser configurados para gerar uma interrupção externa, por meio da função *attachInterrupt()*.

- Para interface com o mundo analógico, a placa Arduino UNO possui 6 entradas, onde cada uma tem a resolução de 10 bits. Por padrão a referência do conversor AD está ligada internamente a 5V, ou seja, quando a entrada estiver com 5V o valor da conversão analógica digital será 1023.

2.9 Conversão de Sinal Digital PWM para Analógico

A conversão de números digitais para valores analógicos ou vice-versa é realizada por Conversores digital/analógico (D/A) e analógico/digital (A/D). Em sua maioria, estes conversores são circuitos assíncronos e que se utilizam de circuitos analógicos para realizar a conversão. Suas aplicações são amplas e seu uso é muito comum quando sistemas digitais e analógicos se misturam para formar um único circuito, por exemplo, em áreas onde a medição, monitorização ou controle de grandezas analógicas são realizadas por intermédio de sistemas digitais. Nos conversores, as grandezas analógicas, normalmente na forma de tensão, limitadas em amplitude e frequência, tem suas amplitudes codificadas em números binários (UFRJ, 2013).

Um obstáculo com esta conversão são os infinitos valores analógicos para discriminar em um número finito de bits, ocorrendo pequena diferença entre o valor real e convertido, ou seja, erro. Este valor exato de tensão e aquele quantificado pelo número digital pode ser considerado como ruído.

A conversão de um sinal analógico em digital pode ser vista na Figura 2.17. Nela estão o sinal real (verde), sinal convertido (vermelho) e o erro (azul).

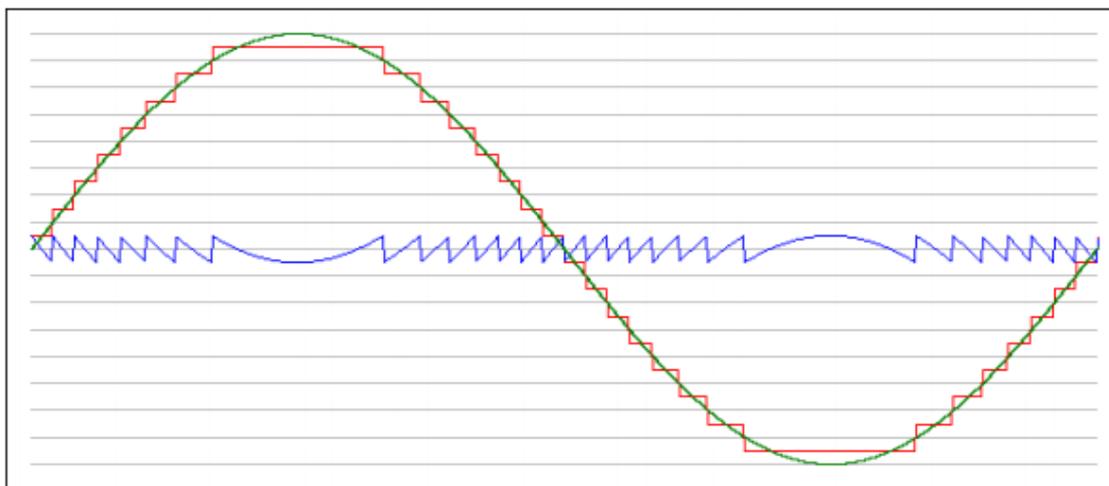


Figura 2.19 Conversão analógico/digital de um sinal senoidal.

Fonte: UFRJ, 2013.

2.10 Filtro Passa-Baixas

Filtro passa-baixas é o nome comum dado a um circuito Eletrônico que permite a passagem de baixas frequências sem dificuldades e atenua (ou reduz) a amplitude das frequências maiores que a frequência de corte. A quantidade de atenuação para cada frequência varia de filtro para filtro.

Este filtro passa-baixas é um tipo muito comum de conversor D/A que baseia na transformação de largura de pulso em tensão ou de frequência em tensão. A partir de um único bit digital e juntamente com o filtro passa-baixa é realizado a conversão. Estes conversores trabalham como um integrador com perdas, ou seja, um circuito para cálculo de valor médio (UFRJ, 2013).

A conversão de sinal PWM (*Pulse Width Modulation*) com estes conversores são o mais comum, onde a largura de um pulso, em um sinal de frequência fixa, altera proporcionalmente o valor médio da tensão analógica de saída. O circuito de um filtro passa-baixas é ilustrado na figura 2.18.

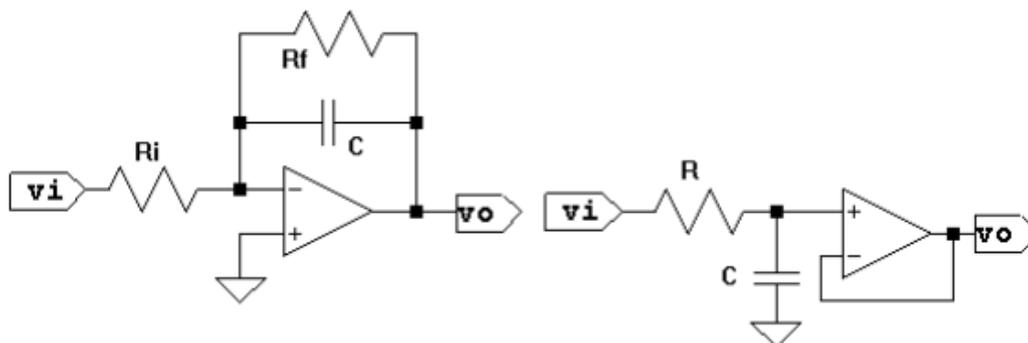


Figura 2.20 Filtro passa baixas

Fonte: UFRJ, 2013.

2.11 Amplificadores Operacionais

Os amplificadores operacionais (AO) são dispositivos extremamente versáteis com alto ganho de acoplamento direto que são normalmente usados em conexões de realimentação para controle de suas características. Estes amplificadores evoluíram de computação analógica, onde eram utilizados para executar operações matemáticas, soma e integração. Então, pelo desempenho e vantagens econômicas, estes AO possuem uma imensa gama de aplicações, sendo úteis em inúmeras aplicações em instrumentação, sistemas de controle, sistemas de regulação de tensão e corrente, processamento de sinais. Também são utilizados em amplificação, controle e geração de formas de onda senoidais (ROBERGE, 1975).

As propriedades de um circuito amplificador operacional ideal são:

- Ganho de tensão diferencial infinito;
- Ganho de tensão de modo comum igual a zero;
- Tensão de saída nula para tensão de entrada igual a zero;
- Impedância de entrada infinita;
- Impedância de saída igual a zero;
- Faixa de passagem infinita;
- Deslocamento de fase igual a zero;
- Deriva nula da tensão de saída para variações de temperatura;

2.11.1 Modos de Operação

Componentes de um AO precisam de uma alimentação em corrente contínua para sua polarização. Sendo assim estes amplificadores trabalham dos seguintes modos: (CEFET, 2015)

- Sem realimentação: O ganho do AO é definido pelo próprio fabricante, este modo é também denominado operação em malha aberta. Este tipo de operação é muito útil quando se utiliza circuitos comparadores
- Com realimentação positiva: Este tipo de operação é denominado operação em malha fechada. Apresenta como inconveniente o fato de não se conseguir controle sobre a saída do circuito
- Com realimentação negativa: O mais utilizado modo de operação em circuitos com AO, sendo também um modo de operação em malha fechada. Esta realimentação é utilizada em diversas configurações obtidas com AO.

2.11.2 Amplificador Inversor

Esta configuração o sinal a ser amplificado é aplicado à entrada inversora do AmpOp, enquanto a entrada não inversora é conectada ao GND do circuito. Esta conexão com o GND fornece uma referência forte, e na verdade todo sinal aplicado à entrada não inversora será uma referência forte para as análises.

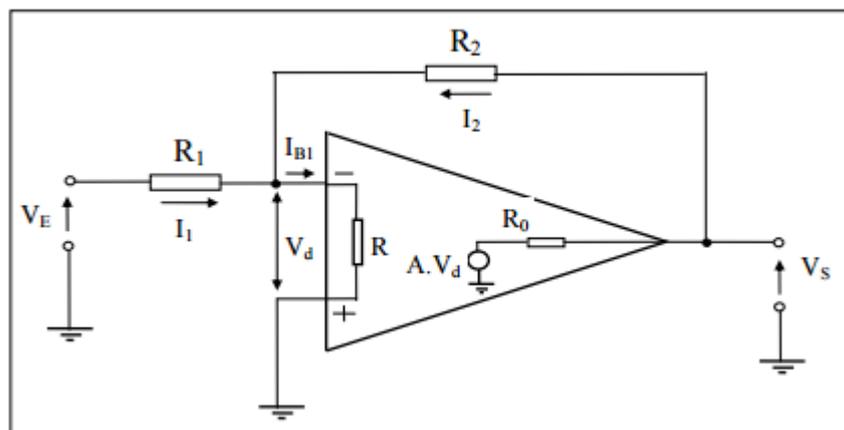


Figura 2.21 Amplificador Inversor com realimentação negativa

Fonte: CEFET, 2015

2.11.3 Amplificador Não-Inversor

O amplificador não-inversor não apresenta defasamento entre os sinais de entrada e saída. Esta configuração o sinal a ser amplificado é aplicado diretamente na entrada não inversora.

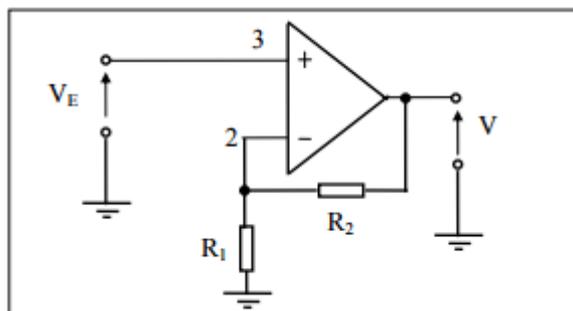


Figura 2.22 Amplificador Não Inversor

Fonte: CEFET, 2015

3. DESENVOLVIMENTO

3.1 Conversão do sinal

3.2.1 Filtro Passa - Baixa

Foi realizado o teste do circuito de conversão do sinal PWM do Arduino para sinal analógico. Esse circuito consiste em um filtro passa baixa, conforme o item 2.10. O circuito contém um capacitor, amplificador operacional e resistor. Os valores dos componentes foram calculados a partir da equação 3.1 conforme abaixo:

$$C = \frac{1}{2\pi R F_c} \quad (3.1)$$

Onde o valor da frequência (F_c) fornecida pelo Arduino é igual a 490 Hz, fixando o valor da resistência para 10 k Ω , foi obtido o valor do capacitor de aproximadamente a 32,5 *10⁻⁶. A conversão do sinal foi demonstrada no osciloscópio, como mostrado na figura 3.2 o sinal PWM e na figura 3.3 o sinal convertido.

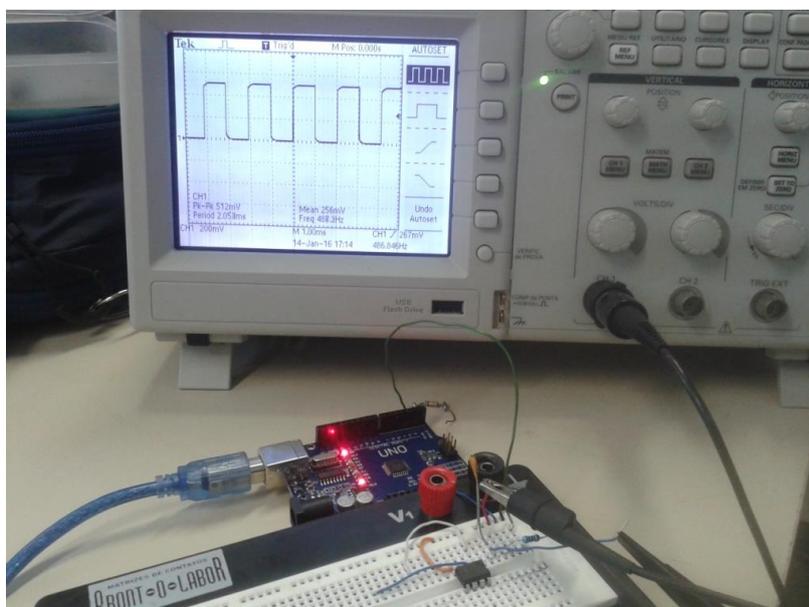


Figura 3.1 Sinal PWM do Arduino

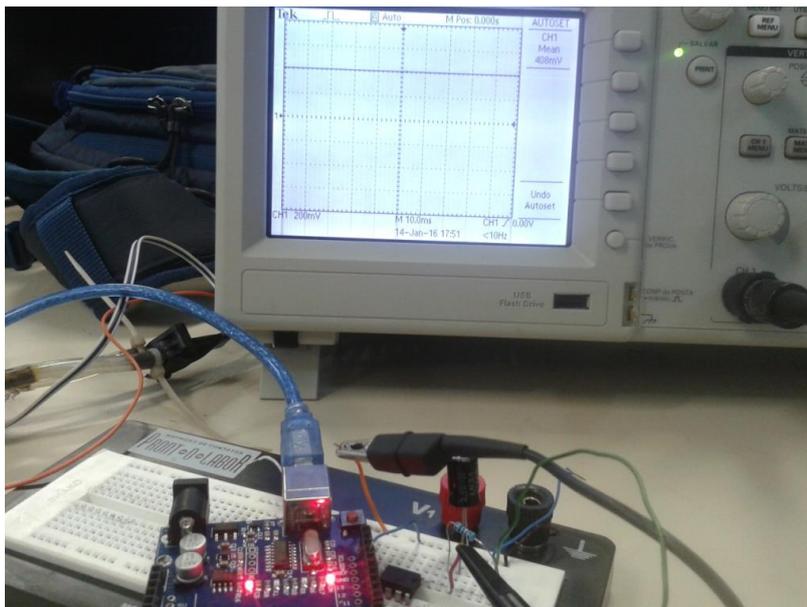


Figura 3.2 Sinal convertido pelo Filtro passa-baixa

3.2.2 Amplificador Operacional

Conforme citado no item 2.5.3, a bomba tem controle de tensão variando entre 2 V até 10 V. Como a tensão de saída do Arduino é de 5V, foi feito um tratamento para amplificar este sinal.

O amplificador operacional (AO) utilizado foi o LM358 (Motorola), montado na configuração não-inversora, conforme explicado no Item 2.11.3. Este CI precisa de uma alimentação externa entre 6 e 12 V. Nesse trabalho foi utilizada uma fonte de 12V.

Este AO tem suas configurações de pinos, conforme mostra a figura 3.4.

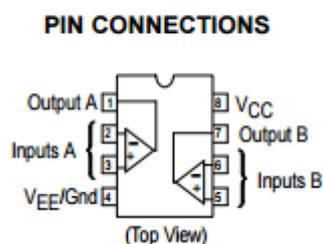


Figura 3.3 Circuito dos pinos - LM358

FONTE: DATASHEET MOTOROLA

O valor do ganho do amplificador foi calculado a partir da equação 3.2 e 3.3.

$$Ganho = \frac{V_0}{V_1} \quad (3.2)$$

$$Ganho = 1 + \frac{R_2}{R_1} \quad (3.3)$$

Onde R1 e R2 são as resistências, Vo é a tensão de saída igual a 10 V e V1 é a tensão do Arduino, ou seja, 5 V.

Portanto, foi aplicado as fórmulas e calculados os valores das resistências. Como elas precisam ter o mesmo valor para se ter um ganho igual a 2 no amplificador, foram utilizadas duas resistências de 10 kΩ.

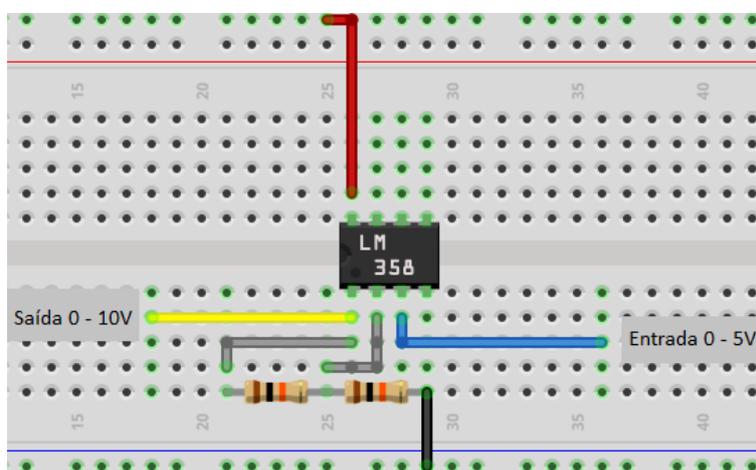


Figura 3.4 Montagem do circuito amplificador

3.2 Calibração da Bomba pelo Controlador de vazão

Com o circuito de tratamento de sinal devidamente montado, foi realizada calibração para o controlador de vazão. Foram utilizadas provetas graduadas, cronômetro e o código de programação anexado ao Apêndice A. Foram medidos os valores de vazão de acordo com a variação da tensão que era introduzida na serial do Arduino. Por meio da interface com o monitor serial do Arduino, valores diferentes para a tensão eram digitados e obtidos por meio de uma saída PWM em um pino específico. Cada valor de tensão corresponde a uma largura de pulso específico na modulação PWM. Essa tensão da saída PWM era enviada ao circuito de tratamento e a saída desse circuito, por sua vez, alimentava a bomba conectada à proveta. A cada variação de tensão era

medido o valor da vazão para 1 minuto e dessa forma, se conseguiu a vazão em ml/min. A figura 3.6 ilustra a montagem feita para essa calibração.



Figura 3.5 Calibração da bomba



Figura 3.6 Esquema de montagem para calibração

Os dados da calibração estão disponíveis na Tabela 1, onde foi possível obter a equação e o gráfico de calibração da vazão da bomba, conforme a figura 3.8.

Tabela 3.1. Dados de calibração da Bomba

Largura de Pulso	Vazão (ml/min)
60	95
76,5	220
102	495
127,5	775
153	1120
178,5	1460

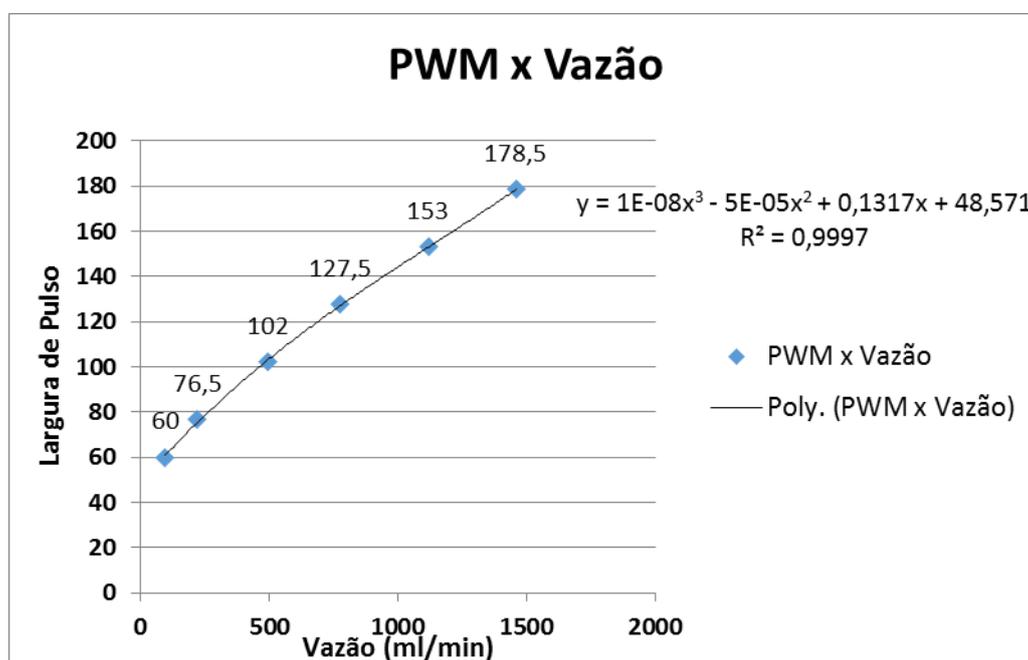


Figura 3.7 Curva de calibração

3.3 Desenvolvimento do medidor de vazão por placa de orifício

Para se obter a vazão utilizando uma placa de orifício, é necessário obter a medida da diferença de pressão antes e depois da placa na tubulação. O componente utilizado para essa medição foi o transdutor de pressão, fabricado pela empresa Motorola. Estes sensores são os transdutores piezoresistivos. De acordo com fabricante, este modelo trabalha com temperaturas entre 0° e 85 °C e possui dois pontos de entrada de pressão, como ilustrado na figura 3.9,

podendo ser utilizado para medir pressão relativa ou pressão diferencial. Neste projeto, foi utilizado como pressão diferencial em uma escala de 0 a 700 KPa.

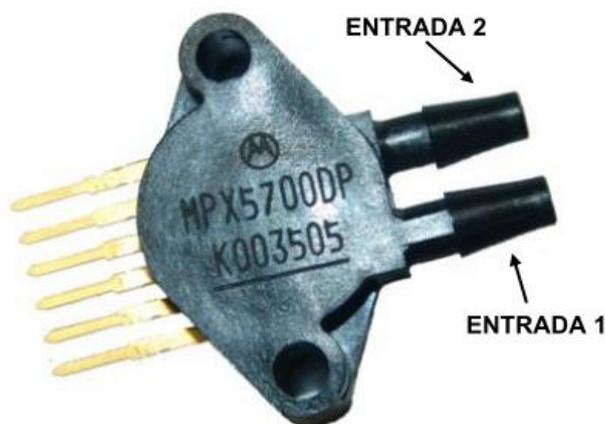


Figura 3.8 Sensor de Pressão MPX5700

FONTE: SANTANA, 2010.

O desenvolvimento da parte estrutural do medidor foi construído a partir da Norma Internacional ISO 5167. Para o desenvolvimento da placa de orifício foi estabelecido primeiramente o diâmetro. As mangueiras utilizadas pela bomba são mangueiras *Viton* para bomba peristáltica com diâmetro externo igual a 9 mm e diâmetro interno igual a 7,5 mm. Então, o medidor de vazão foi desenvolvido a partir destes valores com seus diâmetros internos e externos iguais ao das mangueiras. Como é mostrado na figura 3.10.



Figura 3.9 Protótipo do medidor de vazão

Em relação as tomadas de pressão, as mesmas foram estabelecidas pelas normas do Munson (2004). De acordo com os valores do diâmetro externo de 9

mm, as tomadas foram instaladas com um valor de D (diâmetro) e D/2 a partir do ponto da placa de orifício.

A parte dos tubos foi desenvolvido no torno mecânico no setor de mecânica da Fundação Gorceix em Ouro Preto – MG. O protótipo foi construído de nylon e as tomadas de aço inox, para evitar a oxidação. A placa de orifício foi construída com nylon do tipo concêntrica, como é citada no item 2.3.1.

3.4 Calibração Medidor de vazão por efeito hall

A calibração do sensor de vazão HS05 foi realizada com provetas de 1000 ml, 500, ml, 250 ml e um cronômetro. O início do procedimento para a calibração foi de acordo com a largura de pulso do sinal PWM. Foi enviado pela porta serial os valores de largura de pulso variando entre 0 e 255, correspondendo entre 0 a 10 V, e simultaneamente era recebido do sensor HS05 pela serial os valores de pulsos para cada minuto, sendo medido com as provetas o volume. Foram relacionados os valores de vazão, pulsos e largura de pulso, de acordo com a tabela 2.

Tabela 3.2 Dados da Calibração do Sensor

PWM	Pulso	Vazão
60.64	1	97
75.23	4	233
102.72	11	500
125.26	16	770
147.4	23	1080
165.39	28	1320

Com a calibração foram realizadas 6 medições de vazões diferentes com 5 repetições em cada. De posse desses dados, como mostrado na Tabela 3 e por meio do software Excel, plotou-se um gráfico relacionando esses dados, e traçou-se a curva de resposta, mostrada na figura 3.11. A partir do gráfico, obteve-se a equação de calibração do sistema, que foi incluída na programação.

Tabela 3.3. Dados de repetições de vazão

Pulsos/Segundos	ml/ min					Média
1	97	95	100	96	98	97,10
5	233	235	240	230	234	234,30
10	500	505	493	505	502	500,90
17	770	780	770	780	780	776,00
25	1080	1085	1085	1080	1083	1082,50
32	1320	1320	1325	1330	1333	1325,50

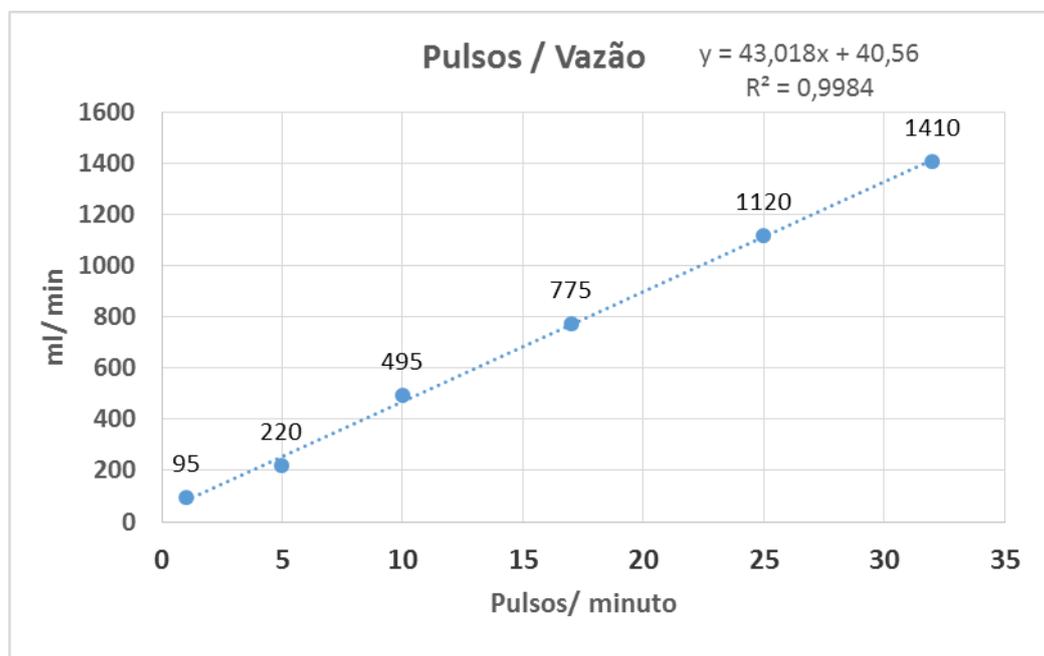


Figura 3.10 Curva de Resposta - Sensor de vazão

Com a equação de calibração, as medidas padrões mostradas na tabela 1 foram mantidas, e então foram realizadas mais 5 medições para cada padrão. Com este número de medições e por meio a tabela de coeficiente de "t" de *student*, obtém-se o valor da estatística t a 2,132 para um nível de confiança de 90% bicaudal e 5 graus de liberdade. Calculou - se o desvio padrão (u),

variância e média das 5 medidas pelos comandos de funções no Excel, conforme mostrado na tabela 4.

Por meio da fórmula $Re = t * u$, calculou - se a repetitividade (ou repetibilidade) e plotou - se o gráfico da curva de erros, mostrada na figura 3.12 que representa a distribuição dos erros sistemáticos e aleatórios ao longo da faixa de medição. A repetitividade é variação das medidas obtidas em cada ensaio, com o mesmo equipamento e método.

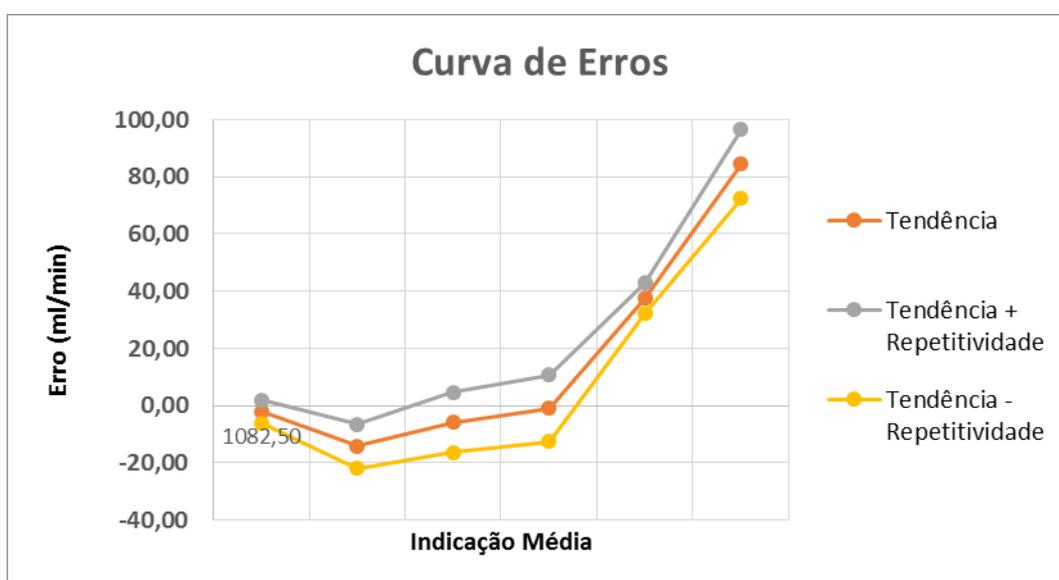


Figura 3.11 Curva de Erros - Sensor de Vazão

Tabela 3.4. Dados Estatísticos

Padrão	Média	Tendência	Correção	Variância	Desvio Padrão	Repetitividade	T + R	T - R
95	97,10	-2,10	2,1	3,55	1,884	4,017	1,92	-6,12
220	234,30	-14,30	14,3	13,45	3,667	7,819	-6,48	-22,12
495	500,90	-5,90	5,9	24,3	4,930	10,510	4,61	-16,41
775	776,00	-1,00	1	30	5,477	11,677	10,68	-12,68
1120	1082,50	37,50	-37,5	6,25	2,500	5,330	42,83	32,17
1410	1325,50	84,50	-84,5	32,5	5,701	12,154	96,65	72,35

3.5 Projeto e montagem da placa

O primeiro passo para a confecção da placa de circuito foi o desenvolvimento do projeto no *software* de desenho e simulação Proteus. Foi desenhado no *workspace* do software o desenho da placa, adicionando ao workspace todos os componentes, resistores, capacitor, CI's, conectores para tensão, sinal PWM e terra. Assim, realizando as ligações de acordo com a montagem feita na *protoboard*. O circuito no *workspace* pode ser visto na figura 3.13.

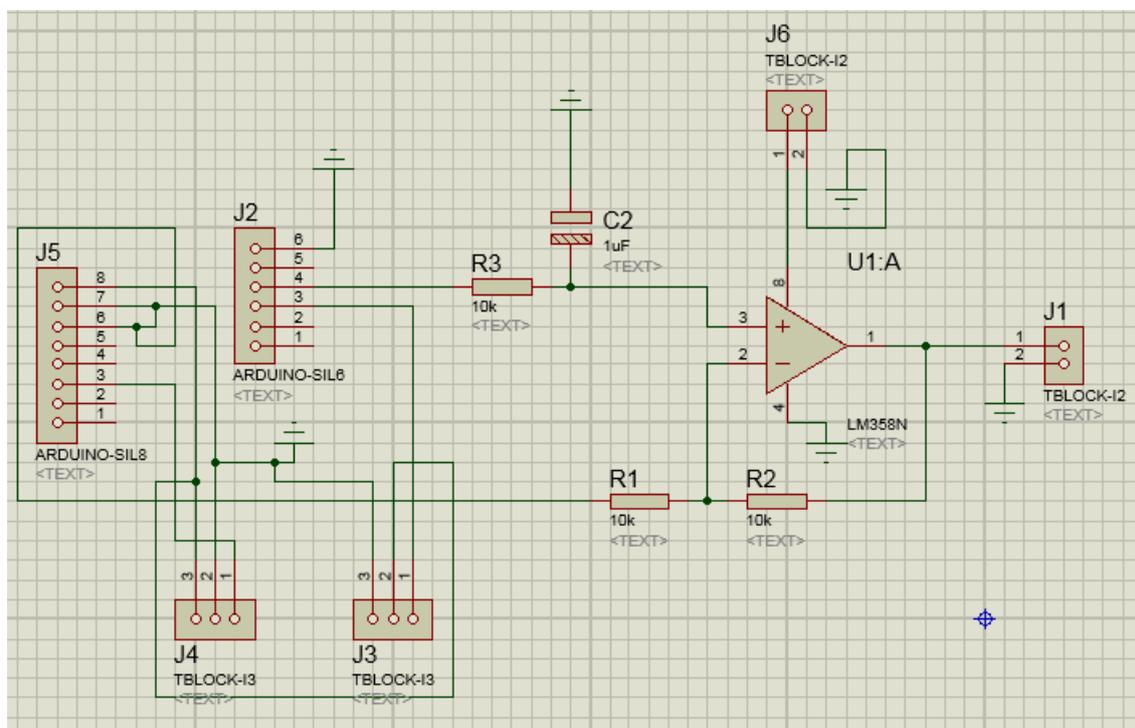


Figura 3.12. Circuito da placa no Proteus

Com o desenho montado corretamente, a próxima etapa foi transferir o circuito do workspace para o *Ares Proteus*, para dar início no desenho da placa. Com a exportação, os componentes foram adequados e foi preciso ajustar os pinos de acordo com as medidas do Arduino Uno. Assim, foi utilizando a função do software *autorouter*, onde foi criado as trilhas e foi preciso adequar para retirar o cruzamento entre trilhas e elevadas inclinações.

O software Proteus gerou um arquivo em pdf do layout da placa para ser impresso com uma impressora a *laser* em um papel fotográfico. A figura 3.14 mostra como o *layout* foi impresso.

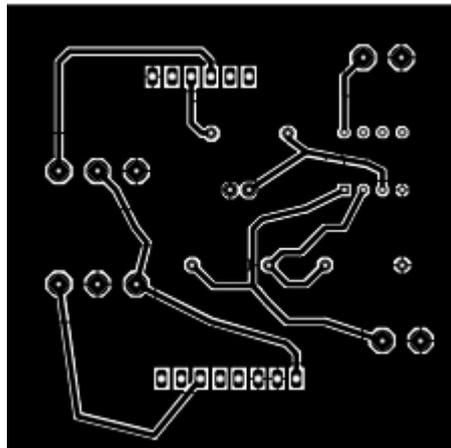


Figura 3.13 Desenho da Shield

O próximo passo foi a limpeza e ajuste da placa fenolite, de acordo com o tamanho da *shield*. Com uma lã de aço foi realizada a limpeza da placa para ter uma melhor aderência da tinta. Com a placa limpa, foi colocado na prensa térmica o *layout* fotográfico junto com a placa. A prensa térmica foi ajustada para 165°C com um tempo de prensagem de 3 minutos.

Após a prensagem, a placa foi colocada em um recipiente com água morna para retirar o papel. Então, a parte impressa foi selada na placa.

A próxima etapa foi a corrosão, a placa ficou em uma solução ácida entre 15 e 20 minutos até ficar totalmente corroída. Depois disso, foi levada para lavagem em água corrente para retirar o ácido.

Com a utilização da furadeira de bancado do laboratório, foi realizado os furos com uma broca de 0.8 mm.

Com a placa corroída e furada, foi realizado os testes de continuidade entre as trilhas.

Assim, as próximas etapas foram inserir os componentes na *shield* e processo de soldagem. Ao inserir os componentes foi observado os componentes que possuem anodo e catodo e o *datasheet* para fazer a ligação da maneira correta. Foi seguido o modelo de projeto do *Proteus* para auxiliar na montagem.

Com a placa fixada em um mandril, de acordo com a figura 3.15, foi realizado a soldagem em um processo demorado e cauteloso, foi fixado cada filamento do

componente na parte cobreada da ilha para ocorrer a condução de corrente elétrica.



Figura 3.14. Processo de soldagem da placa

Com a ligação dos sensores de vazão, do conector DB9 e a fonte de tensão, foi finalizado a montagem da placa, de acordo a com a figura 3.16.

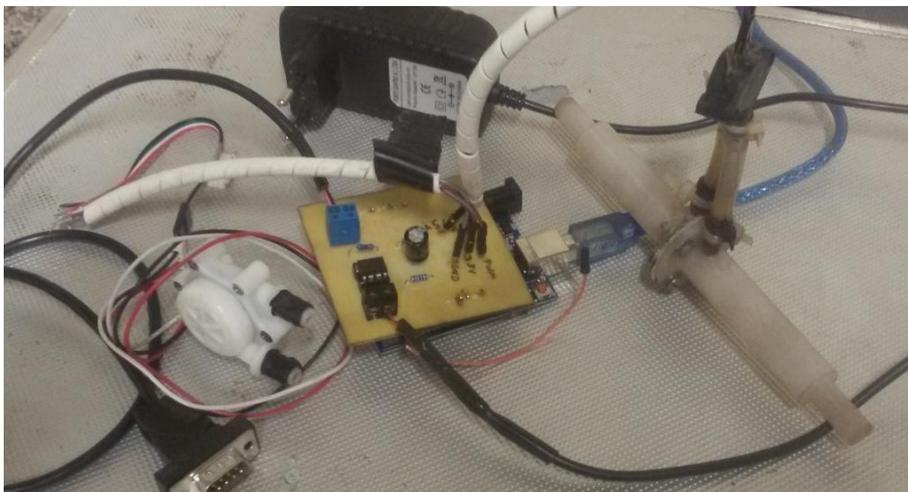


Figura 3.15. Shield com sensores conectados

3.6 Supervisor

O sistema supervisor foi desenvolvido por meio do Visual Studio 2012, utilizando a linguagem C#, onde o código de programação pode ser visto no Apêndice B. Este supervisor contém a tela principal, conforme mostrado na figura 3.17.

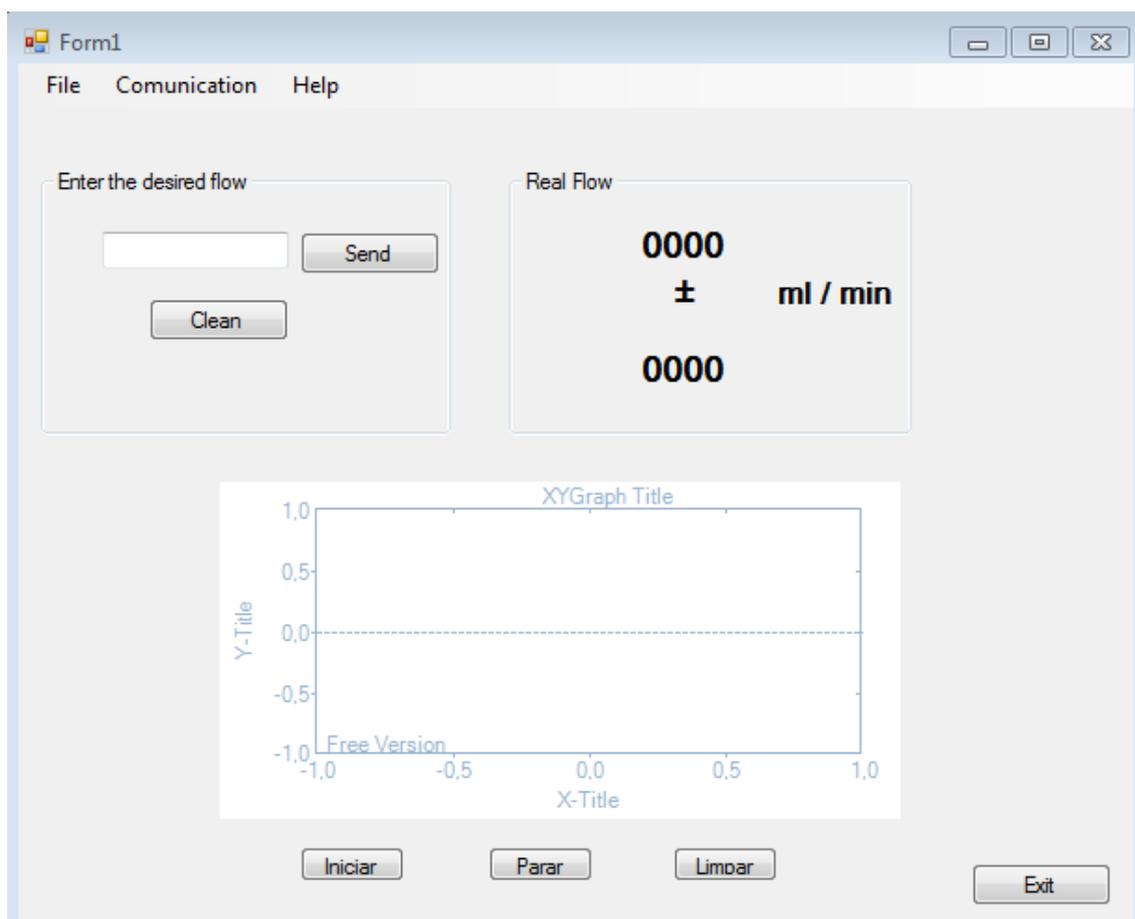


Figura 3.16. Tela inicial do Supervisório

Na tela principal o usuário entra com o valor desejado da vazão e visualiza o valor real de vazão retornado do sensor de vazão, com a correção e a repetitividade obtidos no item 3.5. De acordo com o valor informado pelo usuário, uma interpolação linear é feita com os valores da curva de erros a fim de se identificar a correção e repetitividade a serem. Assim, pode se verificar se o valor de entrada está de acordo com o de saída. Nesta mesma tela, existem as opções de comunicação e Arquivos. Na aba comunicação o usuário configura os valores da comunicação serial, ou seja, configurar porta, velocidade, paridade, bits de dados e bits de parada, conforme mostrado na figura 3.18.

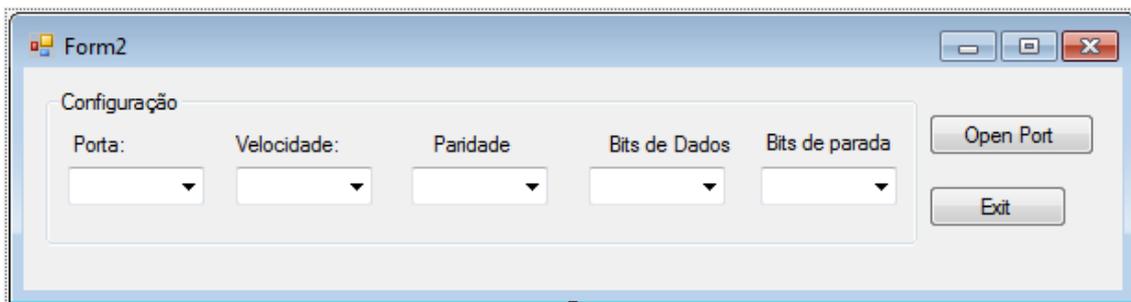


Figura 3.17. Tela de Configuração Serial

4. RESULTADOS E DISCUSSÃO

4.1 Sensor de Vazão com placa de orifício

Os primeiros testes foram realizados com o sensor de vazão por placa de orifício, onde não foi possível obter valores reais de medições. Um dos problemas, foi a bomba peristáltica, pois é uma bomba de pulso constante. Dessa forma, a passagem da água pelo sensor não se dava de forma linear, ou seja, o sinal lido pela serial do Arduino apresentava picos, fazendo com que houvesse variações, impossibilitando a obtenção da curva de calibração. Esse comportamento está ilustrado nas figuras 4.1 e 4.2.

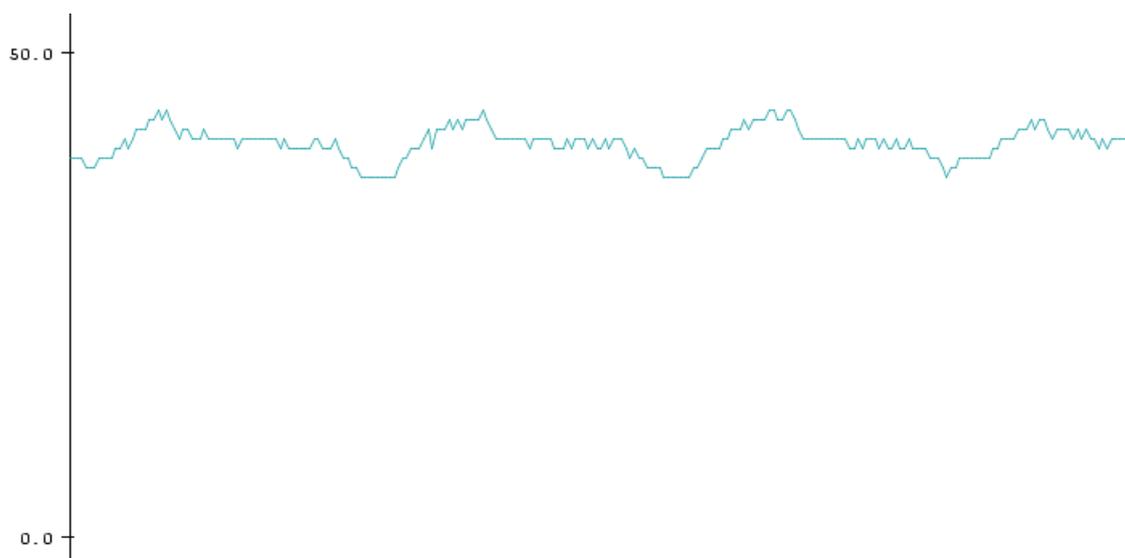


Figura 3.1 Oscilação de resposta com baixa vazão

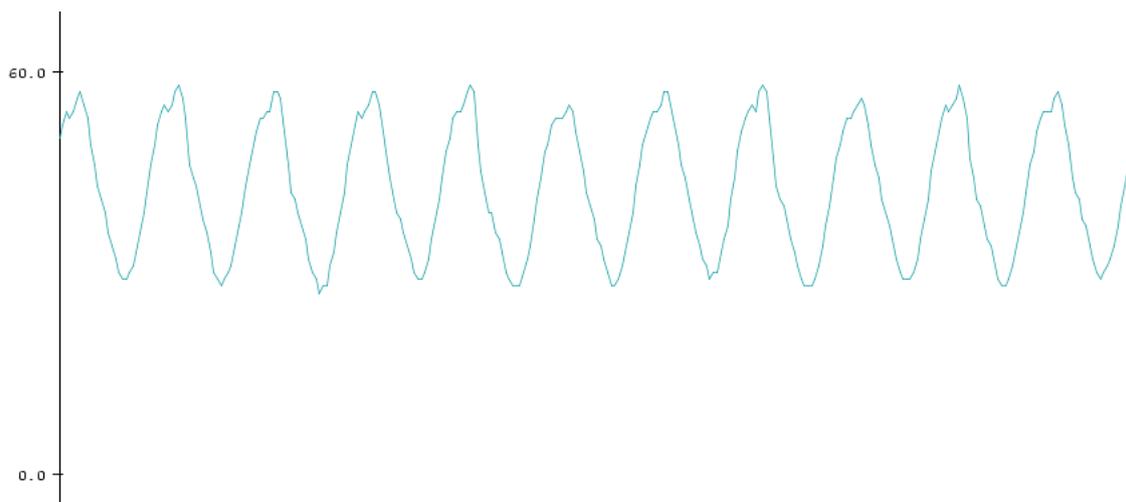


Figura 3.2 Oscilação de resposta com máxima vazão

O sensor da Motorola, modelo MPX5700, também não obteve bons resultados, pois é um sensor para alta pressão na faixa de até 700 KPa. Portanto, o sensor trabalhou em um *range* pequeno, conforme mostrado nas tabelas 5 e 6.

Tabela 4.1 - Dados de leitura analógica com vazão máxima.

Alta Vazão		
44	57	33
48	53	30
50	48	28
52	45	29
55	42	29
55	41	30
55	38	33
55	37	35
58	35	37

Tabela 4.2 - Dados de leitura analógica com baixa vazão.

Baixa Vazão		
42	38	38
41	38	38
41	38	39
40	38	39
40	37	40
40	38	40
40	37	40
39	38	41
38	38	41

Uma tentativa de alternativa foi adicionar entre a bomba e o medidor, um recipiente que tinha o objetivo de tornar o fluxo constante, evitando as variações de pico no sensor. Nesse caso, também não foi possível obter bons resultados, pois, os valores reais não eram condizentes com os valores medidos. Também houveram problemas com a água no recipiente, onde não é desejável deixar material de um teste para o outro.

Outro problema na aplicação do medidor por placa de orifício, foi sua escala, uma vez que, as mangueiras são de diâmetros pequenos, o medidor foi desenvolvido de acordo com essas medidas. Com tais dimensões, tornou-se difícil o trabalho com o torno e com a parte de furação e montagem da placa.

O sensor por placa de orifício quando instalado ao sistema de abastecimento de água da cidade, teve bom funcionamento. Pois a vazão da torneira é contínua e com maior pressão que fornecida pela bomba peristáltica.

4.2 Sensor de Fluxo por efeito Hall

Para se realizar a medição de vazão desejada, trocou-se a o medidor por placa de orifício pelo sensor de vazão HS05. O sensor tem um *range* entre 150 ml/min e 2000 ml/min, dentro da faixa de vazão que a bomba fornece, então foi possível ter valores mais precisos.

Outro ponto importante, é o fato que, suas conexões de entrada e saída de fluxo tem seu diâmetro de 8 mm, ou seja, com o mesmo valor de diâmetro das mangueiras *Viton*, que são utilizadas nos testes.

Com o sensor acoplado a *shield*, Arduino e a bomba, como visto nas figuras 4.3 e figura 4.4.



Figura 4.1. Vista posterior da bomba acoplada com a *shield*. (1) Conector Fêmea DB9. (2) Conector Fêmea DB9. (3) Sensor de Fluxo. (4) Bomba Peristáltica. (5) Arduino. (6) *Shield*



Figura 4.2. Montagem da bomba acoplada com *shield* e sensor

Com o sistema devidamente montado e calibrado, foi obtido na tabela 7 os resultados de medição, relacionando o valor de entrada, leitura do sensor de vazão e o valor de saída. As figuras 4.5, 4.6 e 4.7 ilustram as medições feitas e o sistema de supervisão. Foram realizados vários testes de validação para cada valor de entrada.

Tabela 3.3- Dados do ensaio de vazão

Valor de Entrada	Leitura do Sensor	Medido na proveta
250	242	240
400	375	380
600	551	550
1000	770	780

De acordo com a tabela é possível observar que quanto maior os valores de vazão, maior é o erro entre os valores inseridos pelo usuário e os valores de leitura. Isso se deve ao fato de ser valores perto do limite da bomba. A bomba geralmente é operada com vazões entre 100ml/min á 800 ml/min.



Figura 4.3. Medição com vazão com 1000 ml/min

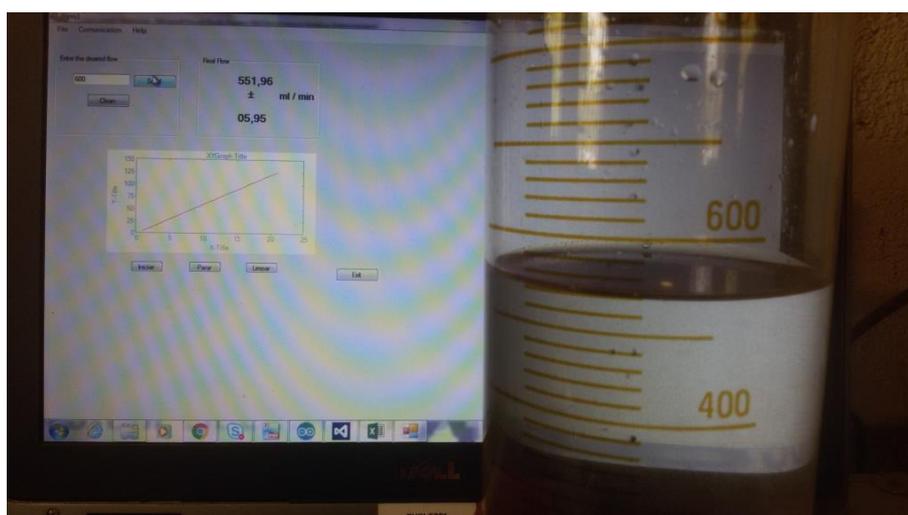


Figura 4.4. Medição de vazão com 600 ml/min

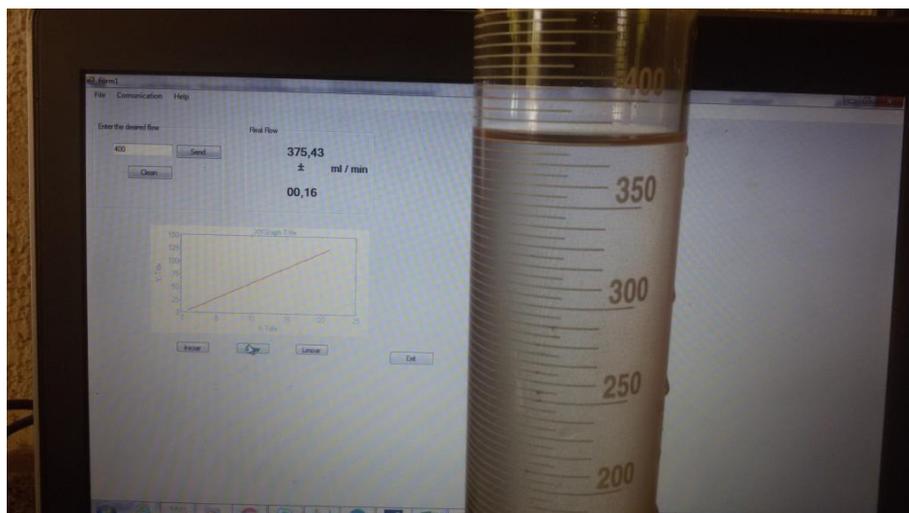


Figura 4.5. Medição com vazão de 400 ml/min

5. CONCLUSÕES

Com base no desenvolvimento e nos resultados obtidos, para as condições que se realizaram o projeto apresentado neste trabalho, conclui – se com as seguintes afirmações:

- O sensor de fluxo HS05 teve resultados satisfatórios, dando confiabilidade nas medições e com fácil adaptação e programação. O sensor operou em diversas vazões diferentes, mostrando bom desempenho em uma faixa de 150 ml/min e 1200 ml/min.
- Os testes de bancada de sedimentação tiveram seu tempo otimizado, pois não será mais utilizado o ajuste de vazão manualmente.
- Houve influência na posição e altura das mangueiras conectadas ao sensor, indicando que há influência gravitacional.
- O protótipo de medidor de vazão por placa de orifício precisa ser mais estudado, pois seu sensor de pressão está superdimensionado para as pressões trabalhadas neste projeto. Além disso, em testes posteriores com um fluxo contínuo e pressões maiores (diretamente de uma mangueira ligada ao sistema de abastecimento da cidade) o sensor mostrou medições mais próximas das esperadas indicando que tanto o superdimensionamento do sensor de pressão como a ausência de um fluxo contínuo da bomba peristáltica podem ser fatores que influenciaram no bom funcionamento deste sistema de medição.

5.1 Trabalhos Futuros

- Estudar outros fatores que interferem em medição de vazão, como força gravitacional, densidade do líquido e viscosidade.
- Dar continuidade ao estudo do medidor de vazão com placa de orifício utilizando outros sensores ou outras configurações de placas de orifícios.

REFERÊNCIAS BIBLIOGRÁFICAS

- BRAGA, Newton C. *Sensores de pressão, como usar? Mecatrônica Fácil*, Ano:6, Número:36, set /2007
- CASSIOLATA, César; ALVES, Evaristo. *Mecatrônica Atual - Ano 5 - Edição 26 - Fev. /Mar/06*
- CASSIOLATO, César. *Sensor Hall – A tecnologia dos Posicionadores Inteligentes de última geração*, 2006. *Smar Equipamentos Industriais*. Sertãozinho – São Paulo.
- CEFET, *Apostila de Circuitos Eletrônicos do CEFETES*. 32 páginas.
- DALLA, Leonardo P. *Sensor de fluxo de água, Equipe Usinainfo*. Disponível em <www.usinainfo.com.br>. Acesso em 22 de março de 2016.
- DELMEÉ, Gerrard. J. *Manual de medição de vazão; 3ª Edição – 2003*; Editora Edgard Blücher Ltda.
- FERREIRA, Cláudio C.S, *Avaliação macro e micro geométrico de dispositivos utilizados na indústria do petróleo e gás*. 2007. 66. Tese (Graduação em Engenharia Química) – Universidade Federal do Rio Grande do Norte, Rio Grande do Norte.
- *Infusion Pump Inspection Frequencies. Health devices*, vol. 27, nos. 4-5, pp. 148- 148-150, 1998.
- MACINTYRE, Joseph A. M. *Bombas e Instalações de Bombeamento*. 2ª ed. Editora LTC, 2012. 782 paginas
- MARKOS, *Água – Um olhar integrado, Automação livre para laboratórios de Águas*. 2013.
- MARTINS N, *Manual de medição de vazão através de placas de orifício, bocais e venturis*. 1ª Edição – 1998; Editora Interciência.
- MASTERFLEX, *Operating Manual: Pump Drives*, Edição 4, 2000.
- MUNSON, Bruce R; YOUNG, Donald F; OKIISHI Theodore H. *Uma Introdução Concisa à Mecânica dos Fluidos*. São Paulo, Editora Edgar Blücher Ltda, 2004,

756 páginas.

- NEVES, Saulo M. Tutorial para criar um sistema de aquisição de dados remoto, alimentado por células fotovoltaicas, 2015. Departamento de Engenharia de Controle e Automação. Universidade Federal de Ouro Preto. Ouro Preto – Minas Gerais.
- OLIVEIRA, André L.F, A utilização do Microcontrolador Arduino para Controle de computador de mesa a distância. 2013, 80. Tese (Curso de pós-graduação *lato sensu* em engenharia de sistemas) – Escola Superior Aberta do Brasil – ESAB, Vila Velha – ES.
- RIBEIRO, Marco A., Instrumentação – 9ª Edição, 2002.
- ROBERGE, James K. *Operation Amplifiers: Theory and Practice*. Cambridge, Massachusetts. Editora John Wiley & Sons, Inc. 1975. 678 páginas.
- SANTANA, Áureo O., Desenvolvimento de um sensor para avaliar a pressão exercida ao solo decorrente do tráfego de máquinas agrícolas, 2010. Escola Superior de Agrícola. Universidade de São Paulo. Piracicaba – São Paulo.
- SCHMIDT, Maik, Arduino A Quick-Start Guide. Dallas, Texas Editora Potomac Indexing, 2011, 300 páginas.
- SOUZA, Fábio. Embarcados, 29 nov, 2013. Disponível em: <http://www.embarcados.com.br/arduino-uno/>. Acesso em 30 de novembro de 2015.
- TELES, V. M. *Extreme programming*: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade. São Paulo: Novatec Editora, 2006.
- UFRJ, Instrumentação e Técnicas de Medida – Conversores AD e DA. 2013, 38 páginas.
- VISUAL STUDIO, Disponível em: < <https://www.visualstudio.com/>> Acesso em 30 de novembro de 2015.

Apêndice A

```
#include "math.h "

int pwm = 11;
float d = 0;
int contaPulso; //Variável para a quantidade de pulsos
int i = 0; //Variável para contagem
int soma = 0; //Variável para armazenar o total de pulsos
int teste = 5;

void setup() {
    // Inicializa a comunicação serial com uma taxa de 9600 bauds.
    Serial.begin(9600);
    pinMode(2, INPUT);
    attachInterrupt(0, incpulso, RISING); //Configura o pino 2(Interrupção
0) para trabalhar como interrupção
}

void loop() {
    contaPulso = 0; //Zera a variável para contar os giros por segundos
    sei(); //Habilita interrupção
    delay (1000); //Aguarda 1 segundo
    cli(); //Desabilita interrupção
    i++; // Incrementa i
    soma = soma + contaPulso; // Soma dos pulsos mensurados
    Serial.println(contaPulso); //Imprime na serial o valor de pulsos
}

void incpulso ()
{
```

```
    contaPulso++; //Incrementa a variável de contagem dos pulsos
}
void serialEvent() {
    String buffer = "";
    while (Serial.available()){
        char x = Serial.read(); // Lê byte do buffer serial;
        buffer = buffer+x;
        float c = buffer.toFloat();

        d = 1*pow(10,-8)*pow(c,3) - 5*pow(10,-5)*pow(c,2)+ 0.1317*c +
48.571;

        analogWrite(pwm, d);
        delay(1000);
    }
}
```

Apêndice B

```

namespace Projeto_parte1
{
    public partial class Form1 : Form
    {
        //Instantiating the Class
        SerialPort SerialCom = new SerialPort();

        string Buffer_Receive = string.Empty;

        //Criating the metod (fuction) Fdelegate that will
        receive a parameter type string 'a'

        public delegate void Fdelegate(string a);

        float eixo_x = 0;
        float eixo_y = 0;
        int aux;
        double vazao;
        int aux2, aux3;
        double te, co;
        public Form1()
        {
            InitializeComponent();
            SerialCom.DataReceived += new
SerialDataReceivedEventHandler(SerialCom_DataReceived);
        }

        void SerialCom_DataReceived(object sender,
SerialDataReceivedEventArgs e)
        {
            Buffer_Receive = SerialCom.ReadLine();

            this.BeginInvoke(new Fdelegate(Serial_Receive), new
object[] { Buffer_Receive });
        }

        public void Serial_Receive(string a)
        {
            string txt_rec = string.Empty;
            int aux;

            txt_rec += a;
            txt_rec = string.Empty;
        }
    }
}

```

```

try {
    aux = Int32.Parse(a);
    vazao = 43.018 * aux + 40.56;
    aux2 = Convert.ToInt32(vazao);
    aux3 = aux2/60;

    if (vazao>=95 && vazao <220) {
        co = ((vazao - 220)*(-2.1+14.3)/(220-95)) -
14.3;
        te = ((1.92+ 6.48)*(vazao - 220)/(220-95))
- 6.48;
    }

    else if (vazao < 95){
        co=0;
        te=0;
        vazao = 0;
    }

    else if (vazao >= 220 && vazao < 495)
    {
        co = ((vazao - 220) * (-5.9 + 14.3) / (495 -
220)) - 14.3;
        te = ((4.61 + 6.48) * (vazao - 220) / (495 -
220)) - 6.48;
    }

    else if (vazao >= 495 && vazao < 775)
    {
        co = ((vazao - 495) * (-1 + 5.9) / (775 -
495)) - 5.9;
        te = ((10.68 - 4.61) * (vazao - 495) / (775
- 495)) + 4.61;
    }

    else if (vazao >= 775 && vazao < 1120)
    {
        co = ((vazao - 775) * (37.5 + 1) / (1120 -
775)) - 1;
        te = ((42.83 - 10.68) * (vazao - 775) /
(1120 - 775)) + 10.68;
    }

    if (vazao >= 1120)
    {

```

```

        co = ((vazao - 1120) * (84.5 - 37.5) / (1410
- 1120)) + 37.5;
        te = ((96.65 - 42.83) * (vazao - 1120) /
(1410 - 1120)) + 42.83;
    }

    vazao = vazao + co;
    label1.Text = String.Format("{0:0,0.00}", vazao);
    label2.Text = String.Format("{0:0,0.00}", te);
}

catch
{
}

}

#region
/* FORM1 Loading
private void Form1_Load(object sender, EventArgs e)
{
    int i = 0;

    #region Ports Load
    foreach (string str in SerialPort.GetPortNames())
    {
        comboBox1.Items.Add(str);

        if (str == "COM1") comboBox1.SelectedIndex = i;

        i++;
    }
    #endregion
}

private void button4_Click(object sender, EventArgs e)
{
    if (SerialCom.IsOpen == true) SerialCom.Close();

    SerialCom.PortName = comboBox1.Text;
    try
    {

        SerialCom.Open();

```

```

        button4.Enabled = false;
        button2.Enabled = true;
        button3.Enabled = true;
        button5.Enabled = true;
    }

    catch
    {
    }
}*/

private void button5_Click(object sender, EventArgs e)
{
    SerialCom.Close();
}

private void button2_Click(object sender, EventArgs e)
{
    if (MessageBox.Show("Do you want to quit?",
        "Verification", MessageBoxButtons.YesNo, MessageBoxIcon.Question,
        MessageBoxDefaultButton.Button2) == DialogResult.Yes)
        this.Close();
}

private void button1_Click(object sender, EventArgs e)
{
    if (SerialCom.IsOpen)
    {
        SerialCom.Write(textBox1.Text + "\r");
    }
}

private void button3_Click(object sender, EventArgs e)
{
    textBox1.Text = "";
}
#endregion

private void aboutToolStripMenuItem_Click(object sender,
EventArgs e)
{
}

```

```

private void serialSetupToolStripMenuItem_Click(object
sender, EventArgs e)
{
    Form2 nForm = new Form2();
    nForm.ShowDialog();
    if (!Classe_Global.Cancela)
    {
        try
        {
            if (SerialCom.IsOpen) SerialCom.Close();

            SerialCom.PortName = Classe_Global.Porta;
            SerialCom.BaudRate =
int.Parse(Classe_Global.BRate);
            SerialCom.StopBits =
(StopBits)Enum.Parse(typeof(StopBits), Classe_Global.SBits);
            SerialCom.DataBits =
int.Parse(Classe_Global.DBits);
            SerialCom.Parity =
(Parity)Enum.Parse(typeof(Parity), Classe_Global.Paridade);

            SerialCom.Open();
        }
        catch
        {
            MessageBox.Show("Failure with the
communication ");
        }
    }
}

private void exitToolStripMenuItem_Click(object sender,
EventArgs e)
{
    if (MessageBox.Show("Do you want quit?",
"Verication", MessageBoxButtons.YesNo, MessageBoxIcon.Question,
MessageBoxDefaultButton.Button2) == DialogResult.Yes)
this.Close();
}

private void textBox1_TextChanged(object sender,
EventArgs e)
{
}

private void Form1_Load(object sender, EventArgs e)
{
}

```

```
        xyGraph2.AddGraph("Gráfico Vazão x Tempo",
DashStyle.Solid, Color.Red, 1, false);
    }

    private void textBox2_TextChanged(object sender,
EventArgs e)
    {
    }

    private void timer1_Tick(object sender, EventArgs e)
    {
        eixo_y += aux3;
        eixo_x++;
        if (eixo_x >= 200)
        {
            timer1.Enabled = false;
        }
        xyGraph2.AddValue(0, eixo_x, eixo_y);
        xyGraph2.DrawAll();
    }

    private void button4_Click(object sender, EventArgs e)
    {
        timer1.Enabled = true;
    }

    private void button5_Click_1(object sender, EventArgs e)
    {
        timer1.Enabled = false;
    }

    private void xyGraph2_Load(object sender, EventArgs e)
    {
    }

    private void button6_Click(object sender, EventArgs e)
    {
        }
    }
}
```