

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO E CIENTÍFICO
DEPARTAMENTO DE ENGENHARIA ELETRÔNICA**

AYRTON LIMA DA ROSA
PEDRO HENRIQUE KAPPLER FORNARI
EQUIPO DE SORO ELETRÔNICO

FLORIANÓPOLIS, JULHO DE 2014.

SUMÁRIO

1- RESUMO.....	3
2-INTRODUÇÃO.....	3
2.1-DESCRIÇÃO DO PROBLEMA.....	3
2.2-MOTIVAÇÃO.....	3
3-SOLUÇÃO.....	4
3.1-SOLUÇÃO ARQUITETURAL.....	4
3.2-CIRCUITO INICIAL.....	5
3.3-METODOLOGIA DE TESTES.....	6
3.4-CIRCUITO FINAL.....	10
3.5-PROBLEMAS ADICIONAIS.....	11
4-RESULTADOS.....	11
4.1-VIABILIDADE ECONÔMICA.....	11
4.2-VIABILIDADE TÉCNICA.....	12
5-CONCLUSÃO.....	12
6-BIBLIOGRAFIA.....	13

1 - RESUMO

Neste relatório abordaremos o desenvolvimento de um equipo de soro eletrônico. O equipo de soro é um aparelho utilizado frequentemente em hospitais e atualmente manual. Traremos a motivação para criarmos este protótipo, a primeira ideia, os testes, a programação, as mudanças no circuito e os meios que utilizamos para obter o resultado final. Por fim apresentaremos o circuito final, o protótipo final e a verificação da viabilidade de produção do produto.

2-INTRODUÇÃO

O equipo de soro, apresentado na FIGURA.1, é utilizado frequentemente em todos os hospitais do mundo. Ele serve para controlar a vazão de soro para pacientes internados no hospital e com algum problema de saúde, como a desidratação ou a falta de vontade de urinar. Infortunadamente esses equipos ainda são manuais, o que traz alguns problemas para os hospitais.



FIGURA.1 – Equipo de soro.

2.1 – DESCRIÇÃO DO PROBLEMA

O maior problema deste equipo é a imprecisão, pois, por ser um aparelho manual, depende do erro humano, que quando se trata de saúde das pessoas pode ocasionar problemas futuros. A enfermeira pode contar as gotas errado, o aparelho pode se desregular com o tempo, até mesmo alguém não qualificado pode alterar o aparelho e acabar prejudicando a saúde do paciente.

Outro problema é o fato de que a enfermeira leva um bom tempo para regular o aparelho, pois é preciso contar a quantidade de gotas que estão caindo, antes de colocar no paciente e verificar constantemente qual é a situação do aparelho, regulando-o novamente se for necessário.

2.2 – MOTIVAÇÃO

Vistos estes problemas, decidimos automatizar este processo, de forma que as enfermeiras digitassem o número de gotas desejado e apertassem um botão que iniciasse o controle de vazão. Tendo uma margem de erro máxima de 2 gotas por minuto e uma auto regulação para manter o valor desejado durante o tratamento do paciente.

Por fim a enfermeira retiraria o equipo do paciente e apertaria um botão para desligar o aparelho. Com isso nós solucionaremos os problemas que o equipo manual traz e inventaremos uma nova tecnologia muito útil e com grandes chances de produção em larga escala.

3 – SOLUÇÃO

Para solucionar estes problemas pensamos em fazer um equipo eletrônico, totalmente automatizado, o qual, ao ser ativado, faria com que um pequeno motor apertasse a mangueira de soro, da mesma forma que o equipo manual faz. Porém o equipo manual é barato e descartável, logo, teríamos que criar algo reutilizável e móvel, para ter um preço acessível e dar mobilidade ao paciente, fazendo com que ele possa se locomover por exemplo até o banheiro ou andar pelo quarto durante sua recuperação. Além de que o aparelho deverá ser preciso, auto regulável com o tempo e fácil de se utilizar, para que as enfermeiras tenham confiança e possam ajustá-lo rapidamente.

3.1 – SOLUÇÃO ARQUITETURAL

Primeiramente precisávamos de algo isolado, que carregasse o soro por dentro do equipamento, possibilitando que o motor controlasse sua vazão, logo, fizemos a própria mangueira, já utilizada no soro manual, passar por dentro do equipamento.

Depois precisaríamos de algo que controlasse a vazão do soro, apertando a mangueira e que pudesse mudar a sua posição frequentemente. Para isso utilizamos um servo motor, pois dessa forma podemos controlar sua posição conforme o número de gotas escolhido pela enfermeira e se autorregular posteriormente.

Outra etapa é a interface com o usuário, ou seja, deveríamos dar um retorno às enfermeiras da quantidade de gotas que está programado para cair. Um display LCD 16X2 é o equipamento que escolhemos para deixar o usuário a par do que está acontecendo e este LCD deve ter uma regulagem de contraste, que será feito por um potenciômetro.

Agora que o usuário já terá como ver quantas gotas ele programará para cair por minuto, devemos ter algo que possibilite essa programação, de forma que ele escolha o controle de vazão do equipamento. Para isso, a melhor opção que encontramos foi utilizar dois botões, um de adição e outro de subtração, de forma que as enfermeiras pudessem apertar “x” vezes o botão de adição e “y” vezes o de subtração e a soma total (x-y) será apresentada no display e atualizada cada vez que o usuário pressionar um dos botões.

Outro ponto importante do projeto é ter o auto ajuste, que será feito a partir de um sensor. Este sensor contará, a cada 20 segundos, quantas gotas estão realmente caindo por minuto e comparando com quantas deveriam estar caindo, e se houver uma diferença maior do que 2 gotas o programa recalculará a posição do motor, de forma que volte a cair a quantidade correta. Este sensor foi pensado como um botão, que entraria em curto cada vez uma gota passasse pela ponta de dois fios, fazendo com que o curto mudasse o estado da leitura e dessa forma seria feita a contagem.

Também há a necessidade de ter uma plataforma que receba o programa e rode o mesmo para que o circuito funcione normalmente. O Arduino foi a plataforma que escolhemos, pois é uma plataforma fácil de programar, além de que se adaptava bem aos componentes do nosso circuito.

Por fim as enfermeiras devem ter um botão que ative ou desative o equipo, ou seja, movimente o motor e inicie o controle e que leve-o para a posição inicial quando o usuário desejar retirar o equipo do paciente. Utilizamos um botão maior, que indique diferença entre o resto nomeando-o “enter” e que fica ativado após a primeira pressão e desativa na segunda, ativando o equipamento quando apertado e desativando-o (quando solto).

A FIGURA.2 apresenta a solução arquitetural.

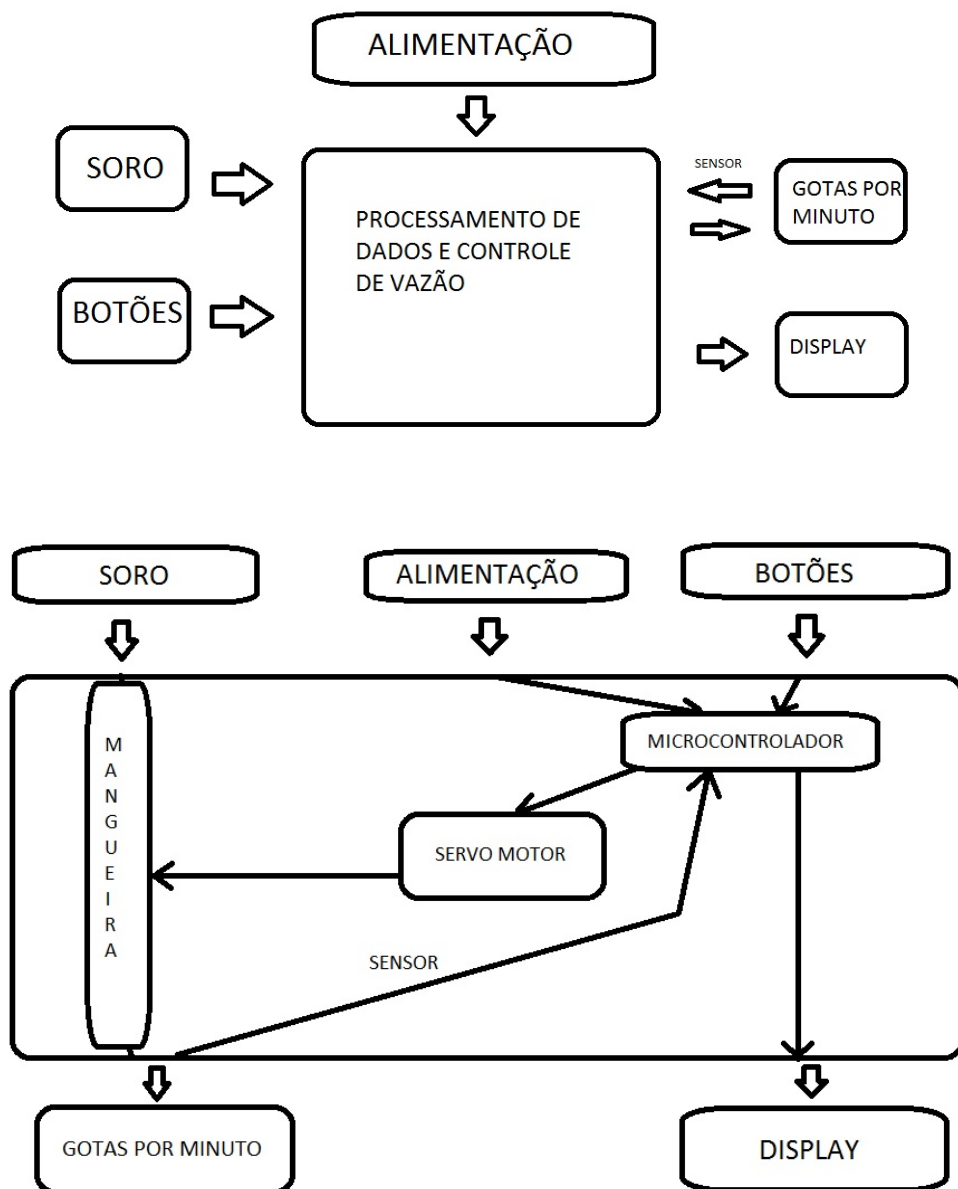


FIGURA.2 - Solução arquitetural.

3.2 – CIRCUITO INICIAL

Nosso circuito inicial, o qual não tinha sido testado, era composto pelos botões, juntamente de alguns resistores, fazendo um “pull down” e conectando eles a portas digitais do Arduino. O motor tinha seu pino de dados ligado a um pino digital do Arduino e os de alimentação aos terminais positivo e negativo do Arduino. O sensor, já que funcionaria como um botão, seria ligado a um resistor, fazendo outro “pull down”, conectado em uma porta digital do Arduino. O display de LCD foi conectado conforme a tabela do site “<http://arduino.cc/en/Tutorial/LiquidCrystal>” que indica as conexões que devem ser feitas nele e no Arduino, além de indicar o que é cada um dos 16 pinos.

A FIGURA.3 ilustra melhor como era nosso circuito inicial.

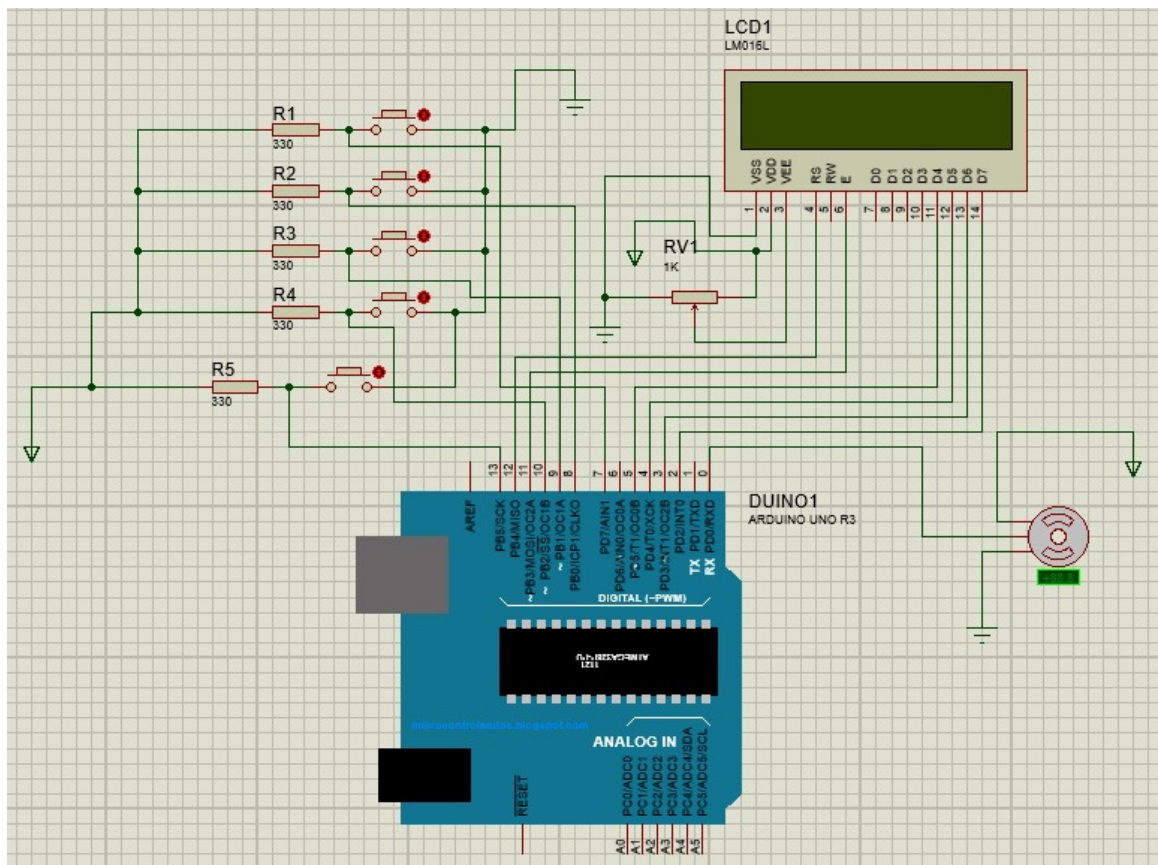


FIGURA.3 - Circuito inicial.

3.3 – METODOLOGIA DE TESTES

Para realizar os testes, separamos o circuito em 4 etapas, que são: a leitura dos botões, o servo motor, o sensor e o display de LCD. Dentro de cada etapa nós criamos um programa e realizamos testes até cada parte funcionar sozinha, e então sim unimos os programas em um programa final e testamos o circuito completo na protoboard. Por fim passamos o circuito para uma placa universal, soldando, deixando-o mais resistente e colocando-o em uma caixa. Estas etapas serão explicitadas abaixo.

- Leitura dos botões

Nesta etapa estudamos “pull down” e “pull up” pois dessa forma conseguiríamos ler o estado dos botões, que irá usar “pull up”. Esse circuito funciona basicamente da seguinte forma: um resistor é ligado ao positivo do Arduino, à entrada de um botão e a um pino digital ou analógico do Arduino, a saída do botão é ligada ao negativo do Arduino (terra), e o pino do Arduino é programado como um pino de leitura. Logo, quando o botão estiver pressionado, o pino do Arduino lerá 5V, já quando este for pressionado outra vez, o Arduino lerá 0V.

Assim saberemos qual é o estado dos botões e poderemos fazer uma lógica programacional para somar, subtrair e ativar o equipamento. Depois fizemos os resultados serem mostrados no serial do Arduino. A parte que tivemos maior dificuldade nesta etapa foi a de atualizar os dados na serial apenas quando um dos botões fosse pressionado.

Para programar essa etapa simplesmente declaramos os botões como pinos de entrada, iniciamos a serial do Arduino e utilizamos uma lógica que incrementava ou decrementava uma variável cada vez que um dos botões era pressionado. Além disso quando apertávamos o botão “enter” enviávamos uma mensagem à serial afirmando que o circuito estava ativado ou desativado.

Na FIGURA 4 ilustramos um exemplo de circuito pull up:

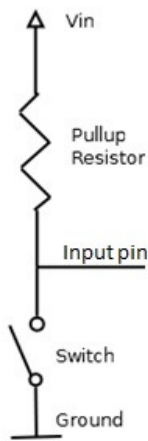


FIGURA.4 – Circuito pull up.

- Display de LCD

Nesta etapa aproveitamos o circuito e o programa da leitura dos botões e acrescentamos o LCD. Como antes os dados dos botões eram expostos na tela do computador, somente tivemos que estudar a programação e a forma de usar o display. Nos baseamos na biblioteca LiquidCrystal fornecida pelo site “<http://arduino.cc/en/Tutorial/LiquidCrystal>”. Nossa maior dificuldade aqui foi o fato de que o LCD não apagava o segundo dígito quando o usuário fazia uma subtração e a soma apresentada no display era maior do que 10, por exemplo, se tínhamos 10, ao pressionar o botão de subtração o display apresentava 90 ao invés de 9. Solucionamos isso apagando o campo de dados cada vez que atualizávamos os dados do display.

O circuito desta etapa e da etapa dos botões está apresentado na FIGURA.5

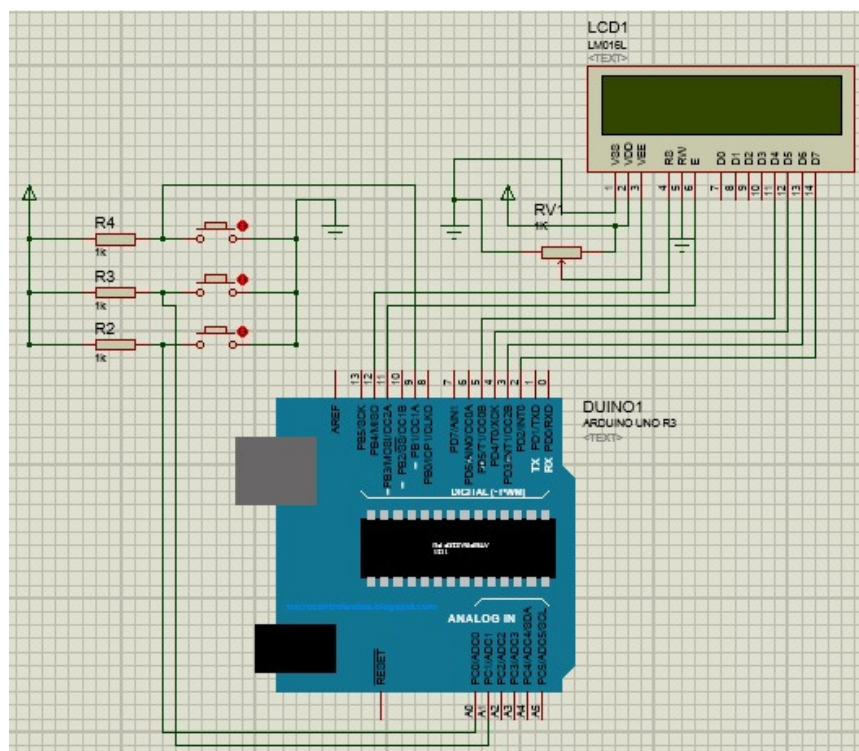


FIGURA.5 - circuito do LCD + botões.

- Sensor

Como fizemos o sensor da mesma forma de um botão a programação de leitura e o circuito foi o mesmo, porém tivemos um pequeno problema: o sensor não interpretava muito bem o 0V, logo

tentamos fazer seu circuito invertido: ao invés de um “pull up” utilizamos um “pull down” logo o sensor estaria sempre em 5V e quando a gota passasse por seus terminais a leitura da tensão seria menor do que 5V e próxima de 0V, o que deixou o sensor mais preciso. A maior dificuldade desta etapa foi fazer com que o resultado da contagem independesse do atraso gerado pelo resto do programa, logo utilizamos a função “every” da biblioteca “timer.h”, que encontramos no site: “<http://playground.arduino.cc/Code/Timer#Description>”, que realiza a função que verifica a contagem a cada 20 segundos em nosso programa. Com essa biblioteca conseguimos armazenar o valor da contagem a cada 20 segundos e, logo após isso, zerá-la.

Aqui precisamos declarar algumas variáveis, utilizar o serial para verificar a contagem e utilizar a lógica de soma cada vez que uma gota gerava contato entre os fios. Após isso criamos a função que armazenaria o valor da contagem e zeraria ela em seguida e que seria executada a cada 20 segundos.

Na FIGURA.6 ilustramos um exemplo de circuito pull down.

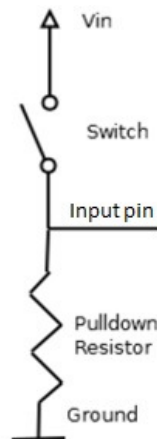


FIGURA.6 - Circuito pull down.

- Motor

O motor que escolhemos utilizar era inicialmente um Micro Servo Tower Pro SG90, pois ele seria pequeno o suficiente e seria bem fácil de controlar sua posição, porém logo nos primeiros testes verificamos que este motor não tinha torque suficiente para conseguir fechar completamente a mangueira e o fluxo de água.

Logo trocamos o motor, escolhemos o motor Servo Tower Pro SG-5010, que possuía um torque muito maior e com ele conseguimos obstruir a vazão completamente. Este motor é controlado por um pino digital do Arduino e seus outros terminais, os de alimentação, devem ser conectados à fonte. Estudamos sua programação pelo exemplo que se encontra no site “<http://blog.filipeflop.com/motores-e-servos/potenciometro-controlando-servo-motor.html>”, que é o site que vende o produto.

Para calibrar o motor, fizemos 10 testes, com 10 posições diferentes no motor, e no fim montamos um gráfico dos valores, com esse gráfico conseguimos funções que se aproximavam muito da curva, e para melhorar a aproximação separamos a curva em 4 partes, obtendo 4 fórmulas que foram usadas no programa. Nesta parte a maior dificuldade encontrada foi conseguir um meio de obstruir completamente a mangueira e fazer a calibragem do motor.

Utilizamos nesta parte a biblioteca “Servo.h”, pois com ela precisávamos somente iniciar o servo e escolher para que posição queríamos que ele se movesse. Abaixo encontrasse o gráfico utilizado para calibrar o motor, na FIGURA.7. O circuito seria apenas o motor ligado a um pino do Arduino e na fonte.

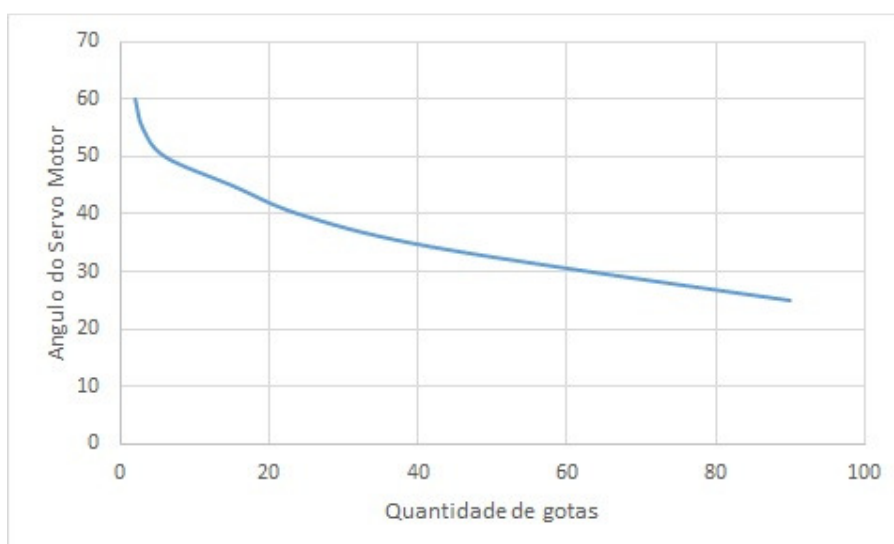


FIGURA.7 - Gráfico

- Junção do programa final, soldagem e montagem.

Nesta última etapa unimos os programas, somente tivemos diferença no programa do motor, já que tivemos que criar uma função “startmotor” e outra “finishmotor” que servem para controlar o motor apenas quando o “enter” é pressionado.

Para soldar, escolhemos uma placa universal, pois era mais simples e prática de realizar a solda e para a montagem, tivemos que alterar a colocação dos pinos dos botões do circuito inicial pois faltaram pinos digitais, logo utilizamos os pinos A0 e A1, que são pinos analógicos, para a leitura dos botões de soma e subtração.

Para a montagem, compramos uma caixa de plástico, medimos e cortamos o que era necessário para a entrada dos botões, da mangueira de soro e do LCD.

A maior dificuldade encontrada nesta parte foi soldar todo o circuito, para que ele juntamente com o Arduino, o motor e os botões coubessem dentro da caixa de plástico.

O programa baseou-se em declarar todas as variáveis das 4 partes, juntamente com as bibliotecas. Na setup inicializamos o servo e o LCD, assim como a contagem do sensor, no loop fizemos a lógica para somar e subtrair conforme os botões eram pressionados; a lógica da contagem e a ativação ou desativação do motor, utilizando as funções “startmotor” e “finishmotor”, as quais foram explicitadas depois. A função “startmotor” faz com que o motor utilize o valor atual da soma total de gotas escolhida pelo usuário como sua posição inicial, além de utilizar o valor da contagem dos últimos 20 segundos do sensor para mudar ou não a posição do motor com o tempo. Já a função “finishmotor” faz com que o motor volte para sua posição inicial. A última função do programa é a que salva o valor da contagem do sensor a cada 20 segundos para enviar para a função start motor.

Abaixo segue uma foto do protótipo (FIGURA.8).



FIGURA.8 - Foto do projeto final.

3.4 – CIRCUITO FINAL

Após os testes realizados tivemos algumas alterações no circuito, como a mudança do “pull down” para “pull up” no sensor e a colocação dos pinos utilizada nos botões de adição e subtração. Porém o circuito final ficou bem próximo do que imaginávamos.

Segue abaixo a FIGURA.9 que ilustra o circuito final.

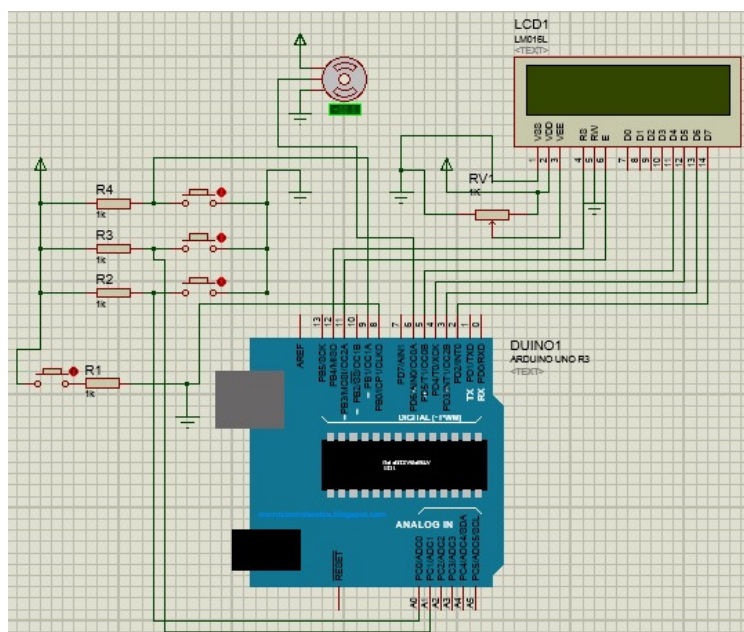


FIGURA.9 - Circuito final.

3.5 – PROBLEMAS ADICIONAIS

Na última semana, quando estávamos calibrando o motor novamente para a apresentação do segundo protótipo, nosso servo SG-5010 quebrou as engrenagens, provavelmente por ter feito força por muito tempo. Logo, tivemos que comprar outro motor, porém, para não repetir o que aconteceu, compramos um motor mais resistente e mais potente.

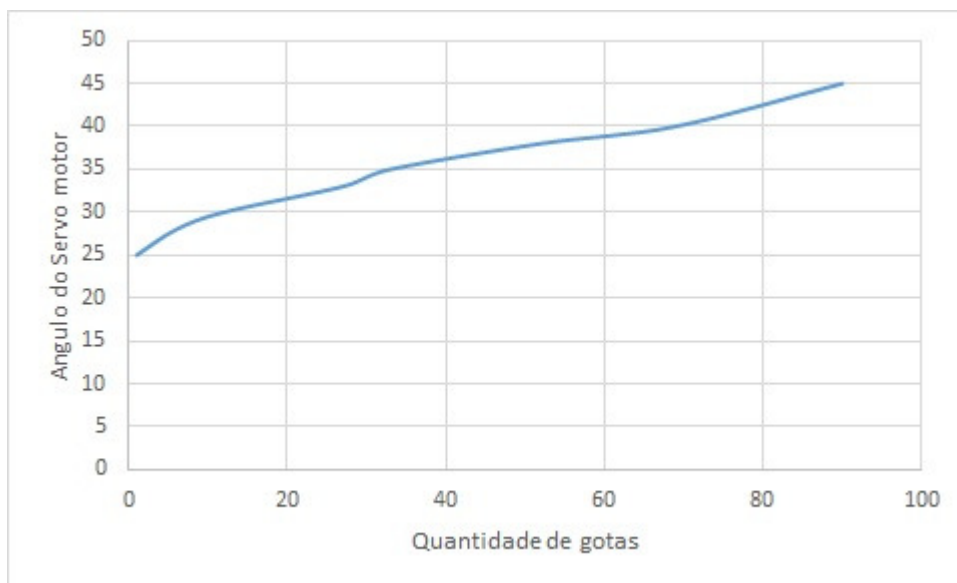
Iniciamos os testes com o novo motor, que a princípio parecia muito similar ao antigo, porém alimentando o motor com o arduino não conseguíamos fazer ele se movimentar corretamente. Nossa primeira impressão era a de que o arduino não conseguia fornecer a corrente necessária para movimentar o novo motor. Então tentamos alimentá-lo com pilhas, o que também não funcionou.

Logo levamos o motor ao Laboratório experimental da UFSC e ali descobrimos que o problema era a falta de corrente, porém a corrente que o motor estava puxando era maior do que 2 ampéres, o que poderia danificá-lo. Esse aumento de corrente ocorreu porque quando o motor era ativado e sua força era muito grande, ele tentava fechar a mangueira, porém mesmo com a mangueira totalmente fechada ele não parava de fazer força, o que aumentava muito sua necessidade de corrente.

Com isso, fomos obrigados a limitar a corrente, utilizando a fonte regulável de tensão DC da bancada do laboratório, porém com a corrente limitada, a tensão diminuía, de acordo com o aumento do ângulo do servo, o que acarretou na diminuição do torque e consequentemente o não funcionamento correto do motor.

Por isso, na apresentação final, não conseguimos a precisão correta do controle de vazão de gotas, mas fizemos testes experimentais mesmo assim, montando uma tabela de valores e um gráfico, que, apesar de não precisos, mostram o funcionamento eletrônico do projeto.

Segue abaixo o gráfico que obtivemos:



4 – RESULTADOS

A análise dos resultados pode ser separada em duas frentes de viabilidade: viabilidade econômica e viabilidade técnica. Satisfazendo esses dois pontos que já foram citados no começo do manuscrito, nós obteremos sucesso no projeto.

4.1 – VIABILIDADE ECONÔMICA

Como citamos no começo deste texto, o equipo manual é um equipamento muito barato e por isso descartável. Logo, como um motor não é barato, deveríamos fazer um equipamento onde pudesse

encaixar e retirar a mangueira, sendo assim reutilizável. Nós conseguimos fazer isso, existe um espaço entre o motor e a mangueira onde ela pode ser retirada e recolocada.

Outro ponto importante é o custo do protótipo: o motor custou R\$37,00; o display LCD custou R\$15,90; o Arduino custou R\$59,90; além disso gastamos em torno de R\$15,00 com botões, resistores, potenciômetro e placa. Então o custo aproximado total foi de R\$128,00, porém esses são preços para pessoa física, se comprássemos em grandes quantidades este preço iria diminuir muito, além de que o componente mais caro do nosso projeto, o Arduino, poderia ser trocado por um único ATMEGA, que custa em torno de R\$14,00 para pessoa física, ou seja comprando em quantidade poderíamos chegar até a menos de 60% do preço do protótipo. Isso para um hospital é um preço acessível se o produto for duradouro, ou seja, outro fator importante alcançado.

Com o problema da última semana nosso custo aumentou em R\$20,00, porém não traria grandes problemas. Esses são os principais pontos da viabilidade econômica do projeto, e nós alcançamos ambos, ou seja, o produto é viável economicamente.

4.2 – VIABILIDADE TÉCNICA

Para que nosso produto seja viável tecnicamente, ele deve, primeiramente, funcionar e ter um raio de erro máximo de 2 gotas por minuto. O protótipo funcionava até a troca do motor com uma precisão inicial em um raio de aproximadamente 5 gotas e o sensor dava conta dos ajustes mais finos, embora seja um tanto quanto lento. Porém após isso não conseguimos mais a precisão inicial.

Outro fator de relevância é a mobilidade do equipamento, ou seja, devemos permitir a locomoção do paciente com o equipo. Para isso fizemos com que o equipo se encaixasse em um suporte de ferro parecido com o utilizado nos hospitais, porém sem rodinhas. Com a possibilidade de encaixar o equipamento em um suporte com rodas, torna-se fácil a locomoção do paciente pelo quarto.

Um último ponto que devemos lembrar é o tamanho do equipamento, nossa caixa tem um tamanho de 54x85x123 mm. Um tamanho viável para ficar preso no suporte com rodas, logo é outro ponto que conseguimos viabilizar.

Sendo esses os três pontos mais importantes para a viabilidade técnica, podemos concluir que nosso projeto era viável tecnicamente antes da troca do motor, porém agora já não é mais.

5 – CONCLUSÃO

Podemos observar assim que os servos que utilizamos não são os ideais para este tipo de projeto, pois eles não tem uma precisão boa e também ou então não tem força suficiente para fechar a mangueira por muito tempo, porém o resto do projeto está adequado a nossa utilização.

Ainda assim, o projeto pode ser, se aprimorado, um projeto para venda aos hospitais, podendo ser utilizado frequentemente no futuro, trazendo menores gastos com enfermeiras e consequentemente maiores lucros aos hospitais, ou até o barateamento da internação. Também podemos dizer que os pacientes estarão mais seguros ao receber o soro, pois o equipamento é mais preciso.

Também podemos dizer que aprendemos muito sobre programação em Arduino principalmente a utilizar leituras de portas digitais e analógicas, como utilizar “pull up/down”, o funcionamento de servo motores, a programação de um display, além de aprender como reparar “bugs” muito comuns nesses tipos de aplicações.

Outro aprendizado de relevância foi a oportunidade de expor criatividade, como na parte do sensor, que criamos um sensor para o projeto, o meio de usar o motor para prender a mangueira. Além disso, ainda aprendemos a soldar e a montar o protótipo em uma caixa, o que conseguimos perceber que não é fácil.

6 – BIBLIOGRAFIA

- <http://arduino.cc/en/Tutorial/LiquidCrystal>
- <http://playground.arduino.cc/Code/Timer#Description>
- <http://blog.filipeflop.com/motores-e-servos/potenciometro-controlando-servo-motor.html>
- <http://arduino.cc>
- https://www.youtube.com/channel/UC5o8o_ruItnoPz6Nax6iDDA