**2. Desenvolvimento**

A tecnologia eletrônica envolvida para desenvolver o sistema telemédico de ultrassom movido por braço mecânico se divide em duas grandes áreas, estas que estão diretamente ligadas às duas regiões existentes no sistema (Principal e Escravo). Além destas duas grandes áreas existe a comunicação entre elas, onde podemos observar a interação entre elas no fluxograma abaixo, figura 1.

**Figura 1:** Fluxograma geral do funcionamento do sistema eletrônico

**2.7 Sistema principal - Módulo estação de controle**

O controle dos movimentos do equipamento será feito pelo médico a partir da estação de controle, esta que faz parte do sistema principal. O médico terá em mãos um aparelho que simula um transdutor de ultrassonografia. A carcaça do aparelho terá o mesmo formato de um transdutor real, para maior comodidade do operador, e será impressa em uma impressora 3D. Dentro desta carcaça será inserido um Arduino juntamente com o módulo sensor inercial, escolhido para a captação dos movimentos do operador.

A escolha do sensor inercial é devido a proposta do produto ser o mais simples possível para utilização do médico. Pressupondo que o médico especialista na qual realizará o exame possui experiência na realização de exames de ultrassom, esse sistema de controle facilita a utilização do produto pois é semelhante a um exame local, com o médico realizando o exame com o paciente presente. Ocasionando uma experiência necessária para utilização sendo apenas um breve tutorial de como o sistema funciona e todas as suas funcionalidades.

Para fora do dispositivo sairá apenas um cabo USB para a comunicação do Arduino como computador da estação de controle. Os dados enviados do Arduino para o computador são do tipo serial, assim, o programa no computador deve ler os dados seriais vindos da porta USB a qual o Arduino está conectado.

O sensor inercial envia ao Arduino 7 sinais:

* 3 eixos de acelerômetro (movimento de translação nos eixos x,y e z);
* 3 eixos de giroscópio (movimento de rotação nos eixos x, y e z);
* 1 sinal de temperatura do ambiente.

Sendo assim, é possível enviar estes pacotes de dados para os motores na estação de exame onde o paciente se encontra, fazendo com que a combinação dos 6 sinais de movimentação controle os motores do braço mecânico. É necessário realizar um processamento de sinais para enviar as informações corretas para cada motor existente nos eixos do braço mecânico.

**Especificações:**

Para o módulo sensor inercial temos como principais critérios de escolha o módulo ter giroscópio, acelerômetro, não ter compasso (não existe necessidade) e ser compatível com a plataforma Arduino. Com base nestes critérios selecionamos os seguintes componentes:

**1) MPU-6050**

Tensão de Operação: 2,375 – 3,46V

Faixa de Giroscópio: +/- 250, 500, 1000 e 2000 graus/segundo

Faixa de Aceleração: +/- 2g, +/- 4g, +/- 8g, +/- 16g

Dimensões: 16 mm x 21 mm

Preço Médio: R$ 20,00



Figura 2: Módulo MPU-6050 (Sensor Inercial)

**2) GY-80**

Tensão de Operação: Mínima=3V , Máxima=5V

Faixa de Giroscópio: +/- 250, 500 e 2000 graus/segundo

Faixa de Aceleração: +/- 2g, +/- 4g, +/- 8g, +/- 16g

Dimensões: 25.8 mm x 16.8 mm

Preço Médio: R$ 100,00

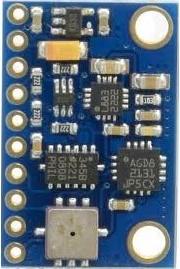


Figura 3: Módulo GY-80 (Sensor inercial);

Analisando as opções disponíveis e suas especificações técnicas dos sensores acelerômetro e giroscópio podemos concluir que ambos os módulos operam com critérios idênticos ou muito próximos. Em seguida temos as três últimas especificações que são dimensão, preço e tensão de operação, como nestas especificações o MPU-6050 se sai melhor por possuir menor dimensão, menor preço e menor tensão de operação sendo assim o escolhido como módulo sensor inercial.

Para escolha do Arduino temos como principais critérios de escolha possuir uma entrada USB/mUSB para saída dos dados seriais, ter dimensões mínimas e possuir pinos para conexão do módulo sensor inercial escolhido, conforme a figura 4. Com base nestes critérios escolhemos o Arduino Nano como melhor escolha visto que outros modelos ou tem USB integrado mas são muito grandes, como Arduino Uno e Mega, ou tem dimensões mínimas mas não existe conexão USB para comunicação serial com o computador. Algumas especificações do Arduino Nano:

* Microcontrolador: ATmega328
* Tensão de Operação: Mínima=3.3V , Máxima=20V
* Dimensões: 18.5 mm x 43.2 mm

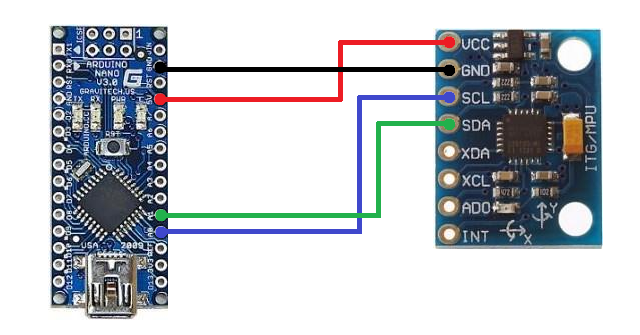


Figura 4: Conexão entre Arduino Nano e Módulo MPU 6050

Por fim temos a caracterização da carcaça que envolve o Arduino com o módulo sensor inercial. A carcaça é composta de duas partes iguais, que serão parafusadas uma contra a outra para comportar componentes, que serão inseridos na cavidade retangular. O cabo USB que alimenta o arduino e transmite os dados passará pelo orifício de 5 mm de diâmetro da parte superior. O seu formato baseado no transdutor convexo que será utilizado no braço mecânico. A peça será impressa em 3D com plástico ABS. O desenho e as dimensões em si podem ser visualizados, nas figuras 6 e 7 (modelo desenhado no software CATIA V5R19).

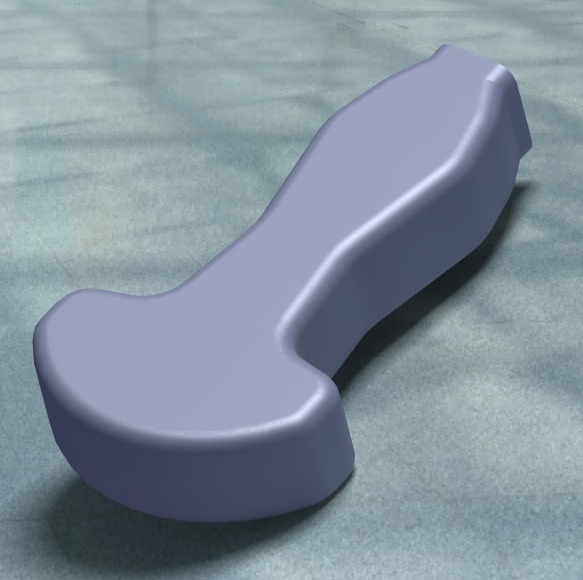
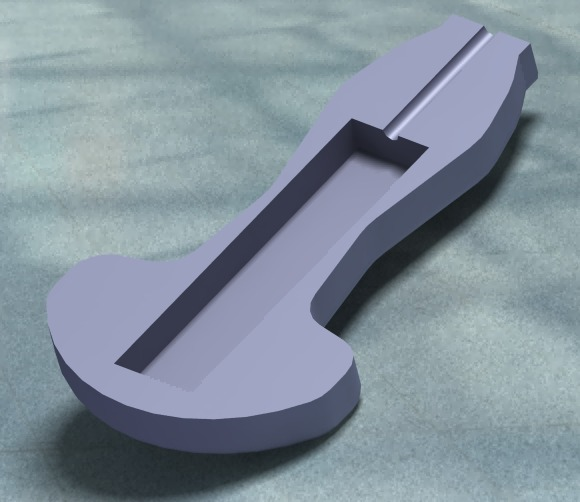


Figura 6: Modelo CAD da carcaça

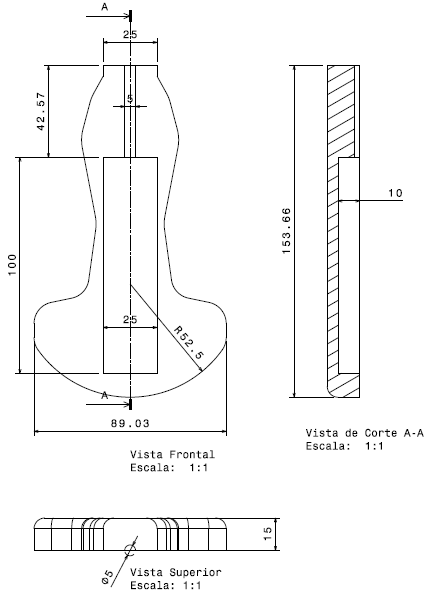


Figura 7: Medidas principais da carcaça

**2.8 Sistema Escravo**

O sistema escravo é composto de dois subsistemas eletrônicos distintos, conforme visto no fluxograma na figura 1. A seguir temos a descrição de cada um deles conceitualmente.

**2.8.1 Subsistema de controle do braço mecânico**

Para a movimentação do braço mecânico primeiramente analisamos os dois melhores tipos de motores que podem ser utilizados para esta função, os servomotores e os motores de passo.

Os motores de passo possuem uma grande quantidade de polos, geralmente entre 50 e 100, enquanto servo possuem usualmente 4 a 12 polos. O alto número de pólos permite que o motor de passo se mova de forma precisa e mais fácil entre cada pólo fazendo com que não necessite de um feedback de posição de um sensor externo. Devido ao seu design o motor de passo possui um torque constante de *holding* (segurar a posição) sem a necessidade de alimentação. A variedade de modelos, preço menor e disponibilidade no mercado são uma grande vantagem comparado com outros tipos de motores.

Para todas as suas vantagens, os motores de passo têm algumas limitações que podem causar importantes problemas de implementação e operação, dependendo da sua aplicação. Os motores de passo perdem uma quantidade significativa de torque quando eles se aproximam da velocidade máxima. A perda de 80% do torque nominal em 90% da velocidade máxima é típica. Os motores de passo também não são tão bons quanto os servos na aceleração de uma carga, além de problemas com vibração e ressonância.

Para aplicações em que alta velocidade e alto torque são necessários, servomotores se destacam. Servos além de possuírem velocidades bem maiores que os motores de passo, eles conseguem manter aproximadamente 90% do torque mesmo em alto velocidade. Além disso servos motores não vibram ou sofrem de problemas de ressonância. Os servomotores são capazes de fornecer mais potência do que os motores de passo, mas exigem circuitos de acionamento muito mais complexos e feedback de posicionamento para um posicionamento preciso. Os servomotores também são muito mais caros do que os motores de passo e muitas vezes são mais difíceis de encontrar.

Portanto baseado na aplicações necessárias do projeto e analisando as vantagens e desvantagens de cada tipo de motor, o motor de passo foi o escolhido.

Para a compra do motores e seus driver foi escolhido os produtos do grupo Grupo Neoyama que tem uma diversidade de modelos de motores de passo, drivers e fontes chaveadas no Brasil. Em específico os motores de passo da família Nema 23 e o driver D2282 devido a compatibilidade em relação ao torque máximo de holding e movimentação do braço.

Na imagem abaixo pode ser visto os diferentes modelos de motores de passos produzidos pela Neoyama e que cobrem os requisitos de cada parte do projeto que podem ser usados na estrutura do projeto, assim como a especificações de cada parte.

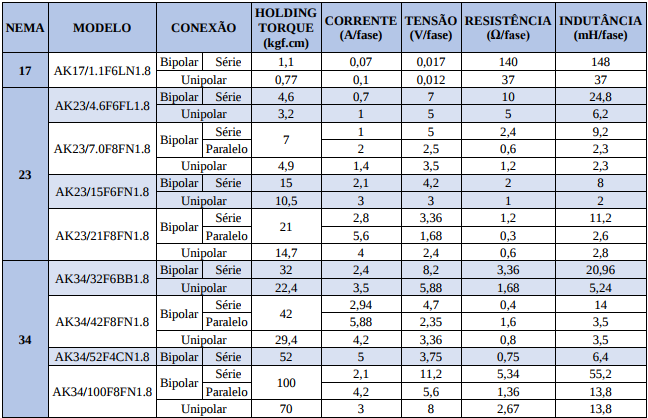


Figura 8: Especificações de motores de passo Nema.

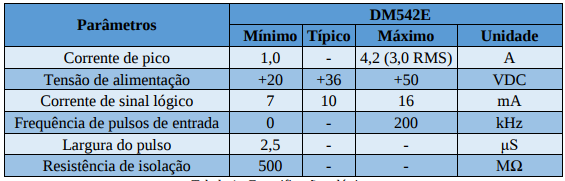


Figura 9: Driver DM542E

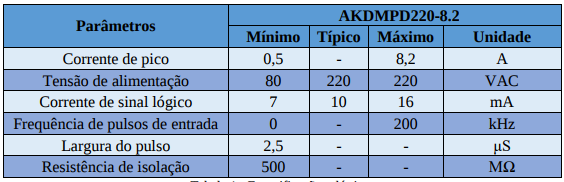


Figura 10: Driver D2282 (driver escolhido)

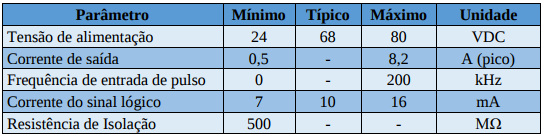


Figura 11: Driver DMA860E

A conexão entre o motor de passo Nema 23 e o driver é conforme o esquemático da figura 12.

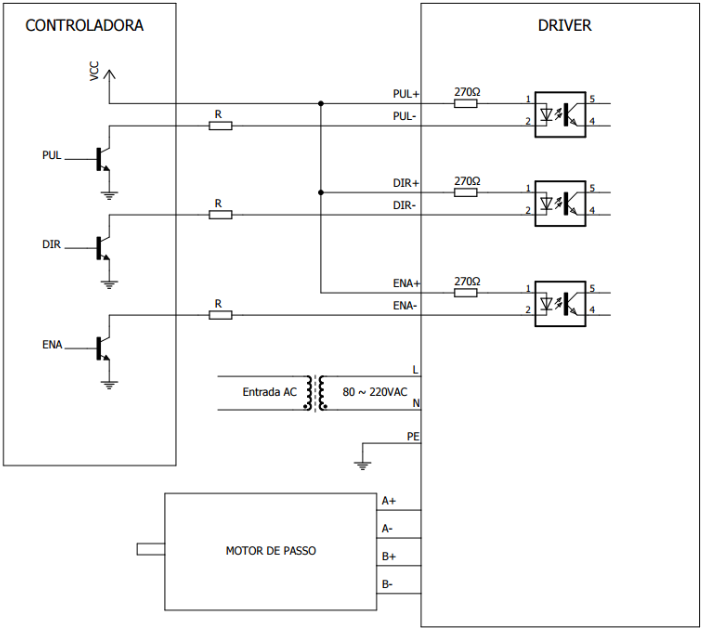


Figura 12: Esquemático da conexão entre driver, motor de passo e controlador [9]

Um sinal é a representação de uma variável, física ou não, dependente que varia em função de uma variável dependente ou independente, quando um ou mais sinais são processados ou modificados por uma ou mais entidades estas são consideradas sistemas.[referencia 1]

Para o controle dos motores será utilizada o Arduino Due devido a necessidade de utilizar um chip ARM para processar os sinais advindos do sistema de controle para os comandos para cada um dos motores de passo, além da necessidade de uma porta USB que envia os sinais seriais do computador para o Arduino e de uma quantidade maior de pinos de conexão para os vários motores existentes. Para programação do Arduino será utilizada principalmente a biblioteca stepper.h, nativa ao software IDE do arduino.

A biblioteca stepper.h permite controlar motores de passo unipolares e bipolares . Um driver apropriado deve ser utilizado a velocidade pode ser controlada. Também é possível girar o motor umas quantidade de passos desejada. Na figura 14 temos um esboço básico demonstrando como seria o controle de um motor de passo simples.

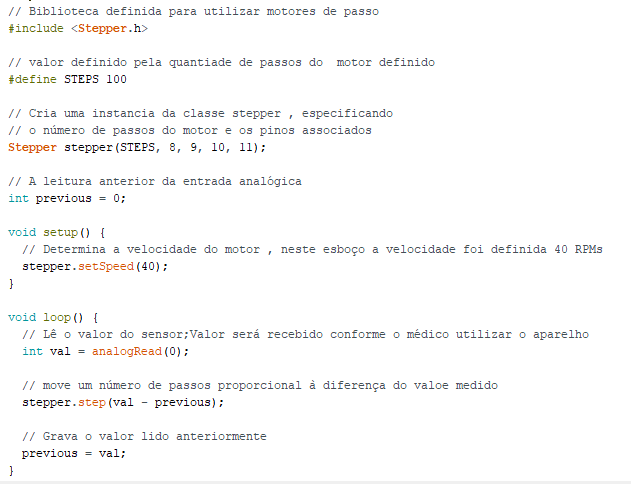


Figura 14: Esboço do código para controle de um motor de passo.

Um sistema pode ser classificado como causal ou não-causal, linear ou não-linear e variante ou não no tempo, dentre outras denominações. O sistema aqui abordado processa sinais(coordenadas) compara com as coordenadas do transdutor, modifica a posição do transdutor com o auxílio de motores, aproveitando a característica das juntas que funcionam como rótulas permitindo a movimentação de forma independente entre os elementos lineares, ou elos [apostila de robótica].

Da forma como foi concebido o sistema é causal pois depende apenas da posição informada pelo sistema de comunicação e da posição atual do transdutor, necessitando assim armazenar a posição recebida e subtrair da posição do transdutor para poder determinar o quanto deve se deslocar para que o transdutor permaneça na posição desejada. Desta forma:

y(n) = x(n)-x(n-1)

Onde:

• x(n) é a posição informada.

• x(n-1) é a posição do transdutor.

• y(n) é a quantidade de espaço em que o transdutor deve ser movimentado para permanecer nas coordenadas informadas.

Para movimentação em cada eixo será criado um procedimento. O algoritmo é representado, de forma simples, através da figura 13:

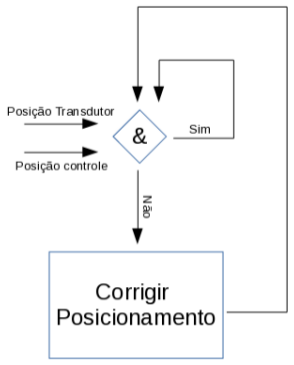


Figura 13: Fluxograma do algoritmo para movimentação dos eixos

Há um problema com relação a orientação, é possível que quando o sistema seja iniciado, os eixos estejam trocados ocasionando no seguinte cenário: O médico move o equipamento para a esquerda de forma horizontal e o braço “responde “ movimentando o transdutor para a direita, isso pode ocorrer devido ao posicionamento dos sensores não coincidirem exigindo uma modificação nos rótulos dos eixos e possível utilização de técnicas de transformações lineares.

**2.8.2 Sistema de ultrassom**

Com base nos requisitos exigidos para este projeto o sistema de ultrassom escolhido deve ter como especificação mínima uma saída de vídeo, VGA ou S-VIDEO, ser portátil e ser compatível com transdutores lineares, para exame vascular, muscular ou de pequenas áreas, e transdutores convexos, para exame na região abdominal. Essas características mínimas são com base na limitação para exames na região abdominal, do torax e a necessidade de transmissão das imagens da tela do ultrassom para um computador com conexão de internet.

Com base nestas características temos a comparação de alguns modelos existentes no mercado na tabela 2.

Tabela 2: (Será feito mais tarde)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

Analisando os dados fornecidos pela tabela 2 podemos concluímos que a melhor opção, levando em consideração os requisitos e o preço, sendo o Sonoscape A6, figura 15. Este sistema suporta vários tipos de transdutores, incluindo um linear e dois convexos, para realização dos mais variados tipos de exame. O transdutor linear utilizado será o Linear Array L745, 128 elementos e frequência doppler entre 5-12 MHz, e o transdutor convexo utilizado será o Convex Array C351, 128 elementos e frequência doppler entre 3-8Mhz, sendo este último modelo indicado para uso em adultos, conforme escopo.



Figura 15: Sistema de ultrassonografia Sonoscape A6;

Para transmissão dos dados das imagens do aparelho de ultrassom para o computador, utilizando a porta VGA, devemos utilizar um dispositivo de captura de vídeo. Este aparelho tem função de receber as imagens advindas da porta VGA e converter em sinais de imagem serial (USB) transmitidas para o computador na codificação MPEG4/H.264. Tanto a codificação MPEG4 quando a H.264 são suportadas pelos sistemas operacionais Windows quanto Ubuntu do computador que recebe os sinais de imagem. Normalmente estes dispositivos são muito utilizados para outras aplicações, principalmente no entretenimento, existindo assim vários modelos e muitas marcas para escolha.

Para este projeto escolhemos o USB 3.0 Video Capture Device da empresa StarTech, as principais características deste produto é suportar imagens até a qualidade 1080p 60fps e entrada de vídeo VGA. A saída para este aparelho é uma porta USB 3.0 que é conectada a um computador compatível com a tecnologia USB 3.0, conforme especificação do fabricante não funciona em aparelhos com USB 2.0 ou inferior.

**2.9. Transmissão dos dados**

A forma de comunicação Arduino/PC escolhida foi a USART (Universal Synchronous Asynchronous Receiver Transmitter). Essa forma de comunicação caracteriza-se principalmente por ser Full-Duplex (Registradores de recepção e transmissão independentes) e pode trabalhar de forma síncrona ou assíncrona, além de ser também uma forma bastante flexível de comunicação serial.

É uma abordagem simples de comunicação que utiliza de 3 fios (Rx, Tx e GND) que pode ser facilmente configurada. Rx e Tx são os fios que irão nos pinos Rx e Tx do microcontrolador para fazer a recepção (Rx) e transmissão (Tx) dos dados. Nos Arduinos com porta USB/mUSB integrados temos a opção de transmitir os mesmos dados deste 3 fios via USB. O Arduino já possui bibliotecas nativas para a utilização deste protocolo e no PC basta configurar a porta serial para a recepção e envio de dados, como a porta USB.

Utilizando um oscilador de 16MHz no microcontrolador pode-se alcançar uma taxa de transmissão de até 2Mbps, podendo chegar a 2.5Mbps caso seja utilizado um oscilador de 20MHz.[5]

O frame enviado por este protocolo é composto de 1 bit de início, de 5 a 9 bits de dados, 1 ou nenhum bit de paridade e 1 ou 2 bits de parada. No final da transmissão de um frame pode-se transmitir outro frame ou pode-se ficar em estado de iddle (nível lógico alto). Caso trabalhe em modo síncrono os bits de início e parada não serão necessários

**Referências:**

[1]

[2]

[3]

[4]

[5]

[6]

[7]

[8]