

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**DANIEL ALVES FUSCO**

**ACIONAMENTO DE UMA CADEIRA DE RODAS ATRAVÉS  
DE UM ACELERÔMETRO BI-AXIAL COMO  
INCLINÔMETRO**

Porto Alegre

2010

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**ACIONAMENTO DE UMA CADEIRA DE RODAS ATRAVÉS  
DE UM ACELERÔMETRO BI-AXIAL COMO  
INCLINÔMETRO**

Projeto de Diplomação apresentado ao  
Departamento de Engenharia Elétrica da Universidade  
Federal do Rio Grande do Sul, como parte dos  
requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: Prof. Dr. Alexandre Balbinot

Porto Alegre

2010

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

DANIEL ALVES FUSCO

**ACIONAMENTO DE UMA CADEIRA DE RODAS ATRAVÉS  
DE UM ACELERÔMETRO BI-AXIAL COMO  
INCLINÔMETRO**

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de “Projeto de Diplomação”, do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: \_\_\_\_\_

Prof. Dr. Alexandre Balbinot, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Banca Examinadora:

Prof. Dr. Alexandre Balbinot, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Luiz Tiarajú dos Reis Loureiro, UFRGS

Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Yeddo Braga Blauth, UFRGS

Doutor pela Universidade Federal de Santa Catarina – Santa Catarina, Brasil

Porto Alegre, junho de 2010.

## **DEDICATÓRIA**

Dedico este trabalho aos meus pais, em especial pela dedicação e apoio em todos os momentos difíceis.

## **AGRADECIMENTOS**

A minha família principalmente meu pai e minha mãe pela ajuda até aqui.

A minha namorada Erica Bressan por todo o apoio e paciência.

Aos meus amigos que entraram junto comigo na UFRGS e aos que, mesmo não sendo “2005/2”, fizeram parte do dia a dia. Com certeza, sem eles teria sido muito mais difícil.

Ao Prof. Dr. Alexandre Balbinot pela orientação ao longo de todo o trabalho.

A empresa Datacom pela experiência profissional e pela disponibilidade de horário.

## RESUMO

Este trabalho descreve as atividades de desenvolvimento e os resultados obtidos no projeto e nos testes de uma cadeira de rodas controlada por um acelerômetro bi-axial disposto na parte superior da cabeça do usuário, tendo como objetivo principal facilitar a locomoção de pessoas portadoras de deficiência motora, que possuam movimento apenas da cabeça ou dificuldades nos movimentos do segmento mão-braço, apresentando um tipo de controle diferente da maioria das cadeiras motorizadas, que são comercializadas, que necessitam que o usuário movimente um *joystick* para controlar o movimento da cadeira de rodas. O sistema utiliza um conversor analógico-digital para medir a inclinação da cabeça do usuário captada pelo acelerômetro, permitindo ao sistema decidir qual deve ser o próximo movimento da cadeira através da comparação da inclinação medida com os valores limites obtidos na calibração do sistema. Após a decisão do movimento é gerado um sinal PWM para controle da velocidade dos motores localizados nos eixos das rodas, permitindo a movimentação em todas as direções. A calibração é individual sendo realizada ao inicializar, permitindo ao sistema se adaptar às condições de mobilidade de cada usuário. Foram realizados testes com voluntários e o sistema apresentou erro máximo de  $1,37^\circ$ , abaixo do erro máximo teórico que o sistema poderia obter. Os testes foram realizados com quatro voluntários testando o sistema desde a calibração até a movimentação das rodas, comparando os valores calibrados com os obtidos no momento do início do movimento.

**Palavras-chaves: Tecnologia Assistiva. Interface Homem-Máquina. Instrumentação Biomédica.**

## **ABSTRACT**

This document describes the development activities and results achieved in the design and testing process of a wheelchair controlled by bi-axial accelerometer placed at the top of the head of the user, having as main objective to facilitate the mobility of persons with disabilities who have movement only in the head or difficulties in the movements of the hand-arm segment, presenting a different kind of control of the most motorized wheelchairs marketed, which require that the user moves one joystick to control the movement of the wheelchair. The system uses a Digital-analog converter to measure the head movement acquired by the accelerometer, allowing to the system decide which should be the next movement of wheelchair, through the comparison between the head movement and the obtained values in the system calibration. After the system decides the direction of the movement the system generates the PWM sign to control the motor speed (localized in the wheel's axis), allowing the movement in all directions. The calibration is personal and performed during the startup of the system, allowing the system to adapt with movements of each user. It was performed tests with volunteers and the system presented maximum error of 1,37°, less than the maximum theoretical error of the system. The tests was performed with four volunteers, testing the system since the calibration (startup) until the wheel's movement, comparing the values acquired during the calibration and the values acquired during the beginning of the movement.

**Keywords: Assistive Technology. Human-Machine Interface. Biomedical Instrumentation.**

## SUMÁRIO

1 Introdução.....	14
2 Revisão bibliográfica .....	15
2.1 Tecnologia Assistiva .....	15
2.2 Cadeira de rodas - Um breve histórico.....	15
2.3 Introdução a Acelerometria .....	18
3 Metodologia Experimental .....	21
3.1 Aparato experimental proposto.....	21
3.2 Hardware .....	22
3.2.1 Circuito do Acelerômetro .....	22
3.2.2 Circuito do PWM .....	27
3.2.3 Circuito de Acionamento.....	30
3.3 Software.....	33
3.3.1 Software do Circuito do PWM .....	33
3.3.2 Software do Circuito do Acelerômetro .....	35
3.4 Mecânica.....	38
4 Resultados Alcançados .....	39
4.1 Circuito de Acionamento.....	39
4.1.1 Verificação do Acionamento dos Mosfets .....	39
4.2 Circuito do PWM .....	43
4.3 Circuito de Controle do Acelerômetro .....	44
4.4 Teste do sistema com voluntários .....	45
5 Conclusões.....	47
6 Trabalhos Futuros .....	48
Referências Bibliográficas .....	49
Anexos .....	50
Anexo A.1 – Software do circuito do PWM .....	50
Anexo A.2 – Software do circuito do acelerômetro .....	55
Anexo A.3 – Placa de circuito impresso do circuito de acionamento .....	62
Anexo A.4 – Placa do circuito impresso do circuito do PWM .....	63



Anexo A.5 – Placa do circuito impresso do circuito do acelerômetro .....	64
--	----

## LISTA DE FIGURAS

Figura 2.1 Gravura do Século IV AC .....	16
Figura 2.2 Cadeira de rodas do Rei Felipe II da Espanha .....	17
Figura 2.3 Cadeira de rodas Dobrável patenteada por H.C. Jennings .....	17
Figura 2.4 Acelerômetro paralelo ao solo .....	19
Figura 2.5 Acelerômetro perpendicular ao solo .....	20
Figura 2.6 Acelerômetro medindo uma parcela da aceleração gravitacional .....	20
Figura 3.1 Diagrama de blocos do sistema proposto .....	21
Figura 3.2 Resposta em tensão do acelerômetro .....	23
Figura 3.3 Esquemático da placa de Controle do Acelerômetro .....	24
Figura 3.4 Foto da placa do circuito de controle do acelerômetro .....	25
Figura 3.5 Foto da placa do circuito de controle do acelerômetro concluída .....	25
Figura 3.6 Foto do acelerômetro disposto no topo da cabeça do usuário .....	26
Figura 3.7 Foto das mensagens apresentadas durante o processo de calibração .....	26
Figura 3.8 Esquemático da placa do PWM .....	29
Figura 3.9 Diagrama de movimentos da cadeira .....	29
Figura 3.10 Foto da placa do circuito do PWM .....	30
Figura 3.11 Foto da placa do circuito do PWM concluída .....	30
Figura 3.12 Circuito de acionamento projetado .....	31
Figura 3.13 Foto da placa protótipo do circuito de acionamento .....	33
Figura 3.14 Fluxograma do software de controle .....	34
Figura 3.15 Diagrama de tempo .....	36
Figura 3.16 Validação da Calibração .....	37
Figura 3.17 Fluxograma do software de circuito de Controle do Acelerômetro .....	37
Figura 3.18 Fixação dos motores nas rodas .....	38
Figura 3.19 Fixação dos motores nas rodas .....	39
Figura 4.1 Simulação da forma de onda do gate do Mosfet-P .....	40
Figura 4.2 Medida da forma de onda do gate do Mosfet-P .....	41
Figura 4.3 Simulação da forma de onda do gate do Mosfet-N .....	41
Figura 4.4 Medida da forma de onda do gate do Mosfet-N .....	42

Figura 4.5 Sinal PWM do Circuito do PWM.....	43
Figura 4.6 Voluntário testando o sistema.....	46

## **LISTA DE TABELAS**

Tabela 3.1 Características do motor Imobras 14.001.12 – Motor DC de 4 pólos .....	38
Tabela 4.1 Verificação dos limites calibrados.....	42
Tabela 4.2 Verificação dos limites calibrados com voluntários.....	44

## **LISTA DE ABREVIATURAS**

DELET: Departamento de Engenharia Elétrica

UFRGS: Universidade Federal do Rio Grande do Sul

$V_{gs}$ : Diferença de Potencial entre *Gate* e *Source* em transistores Mosfet

$R_{on}$ : Resistência entre *Gate* e *Source* quando o *mosfets* está conduzindo

Clock: Relógio do sistema

MEMS: Micro Electro Mechanical System

# 1 INTRODUÇÃO

Este trabalho tem por objetivo desenvolver uma cadeira automatizada controlada por sinais oriundos de um acelerômetro capacitivo bi-axial. Este trabalho se justifica, pois explora um tipo de controle diferente do atual *joystick* para uma cadeira de rodas automatizada, permitindo, através deste, que mesmo usuários com nenhum ou limitado controle do segmento mão-braço, possam se locomover independentemente.

O controle é realizado através de um acelerômetro capacitivo bi-axial MEMS utilizado como inclinômetro disposto na parte superior da cabeça do usuário, assim através da medida da inclinação bi-dimensional da cabeça será realizado o movimento. A cadeira realiza os 4 movimentos básicos: girar no sentido horário, girar no sentido anti-horário, andar para frente e andar para trás, assim será capaz de executar diversos movimentos permitindo que o cadeirante se movimente livremente.

O acionamento dos motores é realizado via PWM para variar a velocidade permitindo um movimento suave e aceleração uniforme e, além disso, o chaveamento faz com que os *mosfets* do acionamento não trabalhem 100% do tempo diminuindo a potência dissipada. O motor DC funciona como filtro para a frequência alta do sinal PWM trabalhando apenas com a componente DC do sinal.

Este projeto, através do Decreto Federal 5.296/04, em seu art. 61[1], se encaixa na definição de Tecnologia Assistiva, pois aumenta a autonomia pessoal de indivíduos com deficiência ou mobilidade reduzida. Além disso, serve como base de estudo para outros meios de controle de cadeiras de rodas além do atual *joystick*.

## **2 REVISÃO BIBLIOGRÁFICA**

### **2.1 TECNOLOGIA ASSISTIVA**

Segundo Rita Bersch e José Carlos Tonolli [2], Tecnologia Assistiva é um termo utilizado pra identificar todo o arsenal de recursos e serviços que contribuem para proporcionar ou ampliar habilidades funcionais de pessoas com deficiência e consequentemente promover vida independente e inclusão. Para Bersch e Tonolli, esses recursos são todo ou qualquer equipamento desenvolvido sob-medida para aumentar as capacidades funcionais das pessoas com deficiência. Os serviços são aqueles métodos que auxiliam o deficiente a selecionar, comprar ou usar os recursos citados.

No Brasil, o Decreto Federal 5.296/04, em seu art. 61 [1] define Tecnologia Assistiva ou Ajudas Técnicas como sendo “produtos, instrumentos e equipamentos ou tecnologias adaptadas ou especialmente projetadas para melhorar a funcionalidade da pessoa portadora de deficiência ou com mobilidade reduzida, favorecendo autonomia pessoal total ou assistida”. Este decreto estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas portadores de deficiência ou com mobilidade reduzida, tais como, atendimento prioritário, acessibilidade em ambientes públicos como prédios e transporte público, além dos meios de comunicação como telefonia pública e internet.

### **2.2 CADEIRA DE RODAS – UM BREVE HISTÓRICO**

A ideia inicial nasceu na tentativa de facilitar a locomoção dos feridos e doentes de uma maneira mais eficiente que carregá-los nas costas. Segundo Otto Marques da Silva e

Ricardo José Del'Acqua [3], é impossível datar exatamente o momento em que o homem percebeu que era muito mais fácil usar rodas ao invés de arrastar sobre folhas ou troncos os doentes e feridos, mas os primeiros indícios de uma cadeira de rodas primitiva são do século IV AC. Como exemplo, Figura 2.1 mostra Hefesto, deus grego das artes manuais e da metalurgia, em uma cadeira movimentada por cisnes.



Figura 2.1 – Gravura do século IV AC.

Fonte – Ferreira, Ray [5].

O Disability Museum [4] possui exemplos das primeiras cadeiras de rodas criadas, como a do rei Felipe II da Espanha em 1595 que possui as seguintes características: repouso para os pés, mecanismo para inclinação podendo se tornar em um leito provisório, como pode ser visto na Figura 2.2.





Figura 2.2 – Cadeira de rodas do Rei Felipe II da Espanha.

Fonte – Silva, Otto Marquesda. Del'Acqua, Ricardo José [3].

Um grande avanço no processo evolutivo das cadeiras de rodas se deu quando, em 1933, Herbert A. Everest encomendou uma cadeira de rodas que pudesse ser transportada por um automóvel, assim o engenheiro H.C. Jennings patenteou juntamente com Everest a primeira cadeira de rodas dobrável [3], que pode ser vista na Figura 2.3.



Figura 2.3 – Cadeira de rodas dobrável patenteada por H.C. Jennings.

Fonte – Silva, Otto Marquesda. Del'Acqua, Ricardo José [3].

A partir de 1933 as cadeiras de rodas sofreram melhorias seguidamente, partindo da cadeira dobrável para a motorizada, controlada por um *joystick*, chegando, a partir do ano de 2004, ao desenvolvimento de projetos para controle de cadeiras de rodas por biosinais.

### 2.3 INTRODUÇÃO A ACELEROMETRIA

Na indústria automobilística os acelerômetros são utilizados no controle dos *air-bags*, no desenvolvimento de celulares, são utilizados para detectar a posição da tela, em informática são utilizados para proteger o HD dos computadores de possíveis impactos mecânicos, na indústria de entretenimento são utilizados para fabricação de controles de videogames, além, é claro, do grande campo da medicina que junto com as engenharias tentam facilitar a vida das pessoas portadoras de deficiência. Com acelerômetros pode-se caracterizar, por exemplo, movimentos, vibrações, acelerações, queda livre de objetos e inclinação. Na medicina, os acelerômetros são utilizados, por exemplo, na caracterização da contração muscular ou até a capacidade dos idosos de sentar-se ou levantar-se de uma cadeira.

Os acelerômetros podem medir acelerações em um, dois ou até três eixos, podendo tomar uma medida tridimensional da aceleração. Conforme a atividade que será monitorada deve-se procurar o acelerômetro adequado, especificando cuidadosamente, a faixa de aceleração de interesse, pois o acelerômetro utilizado em um computador para proteger o HD de um possível impacto mecânico é bem diferente do que deve ser escolhido para acionar *air-bags* em um automóvel. Assim como a aceleração máxima, a largura de banda também é uma característica básica e que depende totalmente do fenômeno que se deseja monitorar. Os acelerômetros não estão sujeitos apenas a acelerações dinâmicas, estão sujeitos, também, a aceleração estática nos acelerômetros capacitivos, por isso todas as medidas para esse tipo de acelerômetro possuem um offset proporcional a aceleração da gravidade no ponto onde está

montado o acelerômetro. Assim, é possível monitorar a inclinação de objetos, que é exatamente uma das características que serão medidas nesse trabalho, ou seja, como um sensor de inclinação (popularmente conhecido como inclinômetro).

Como exemplo, a Figura 2.4 apresenta o esboço de um acelerômetro genérico medindo aceleração. Neste exemplo, o acelerômetro está em paralelo com o solo, sendo assim, seu eixo Z mede a aceleração gravitacional.

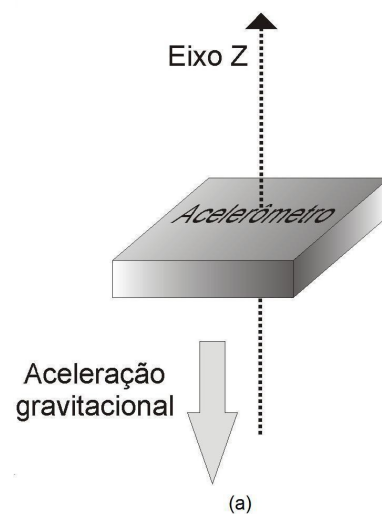


Figura 2.4 – Acelerômetro paralelo ao solo.

A Figura 2.5 apresenta o acelerômetro perpendicular ao solo, sendo assim, seu eixo Z não mede aceleração alguma, pois a única aceleração presente (aceleração da gravitacional) está totalmente perpendicular com o eixo de medida Z. Desta forma, o cosseno entre o eixo Z do acelerômetro e a aceleração da gravidade é igual a zero, pois os vetores formam um ângulo de  $90^\circ$  entre si.

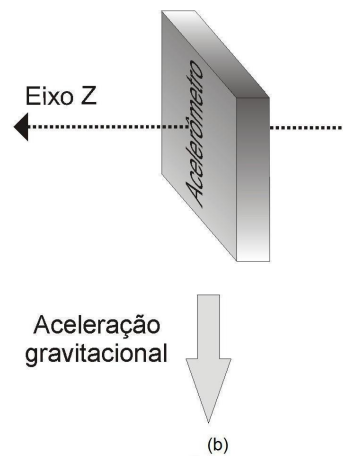


Figura 2.5 – Acelerômetro perpendicular ao solo.

Na Figura 2.6 tem-se o acelerômetro inclinado em relação ao solo medindo uma parcela da aceleração gravitacional proporcional ao cosseno do ângulo formado entre o eixo Z do acelerômetro e o eixo da aceleração gravitacional.

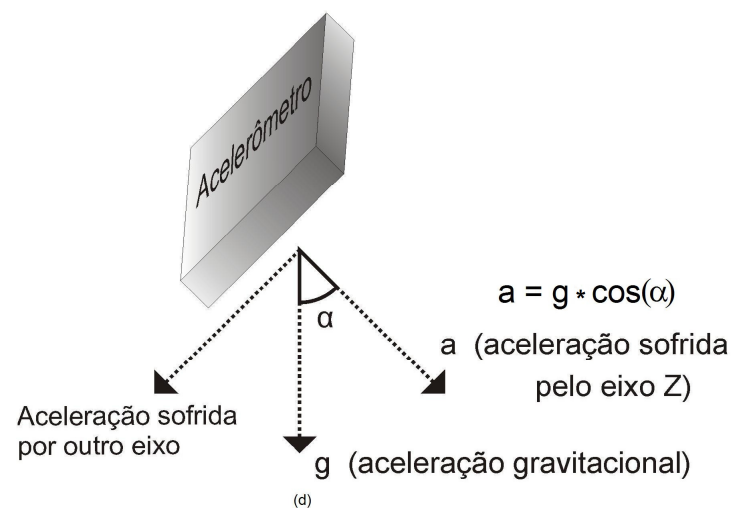


Figura 2.6 – Acelerômetro medindo uma parcela da aceleração gravitacional.

### 3 METODOLOGIA EXPERIMENTAL

#### 3.1 APARATO EXPERIMENTAL PROPOSTO

O diagrama de blocos proposto para o controle e acionamento da cadeira de rodas pode ser observado na Figura 3.1.

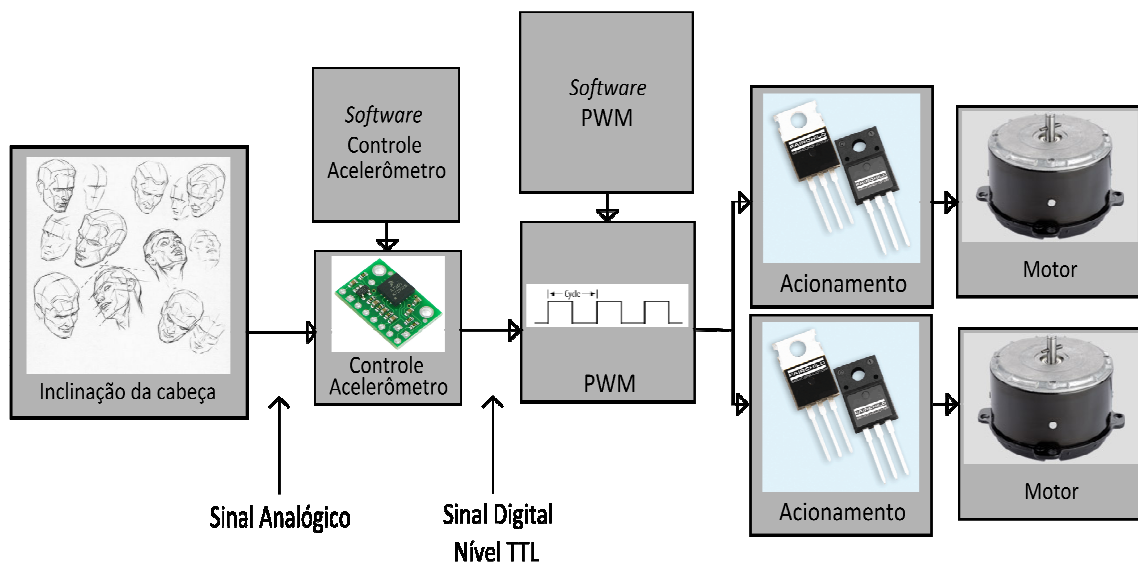


Figura 3.1 – Diagrama de blocos do sistema proposto.

O usuário deste sistema utiliza um boné onde está fixado o acelerômetro MMA7340L na parte superior da cabeça. O circuito do PWM deste acelerômetro capta os sinais da inclinação da cabeça do usuário caracterizando o sentido do movimento desejado. Seu sinal de saída é disponibilizado para o circuito do PWM através de sinais digitais de nível TTL. O circuito do PWM, usando um micro-controlador AT89S52, gera o sinal PWM para o acionamento, que através dos *mosfets* controla velocidade e direção da cadeira.

Foi proposto para este trabalho, que a parte de controle e acionamento fosse como uma “caixa preta”, facilitando o posterior uso da cadeira para outros sensores e não apenas o

acelerômetro, por isso foi separado o circuito do acelerômetro do controle. Ainda no sentido de facilitar mais o uso de outros sensores, os níveis de tensão e corrente usados na entrada do circuito do PWM são compatíveis com a maioria dos *buffers* atuais.

## **3.2 HARDWARE**

O hardware será descrito de acordo com os blocos apresentados na Figura 3.1.

### **3.2.1 Circuito do Acelerômetro**

O acelerômetro utilizado é o MMA7340L [8] tri-axial capacitivo, que possui três saídas em tensão, sendo uma para cada eixo, apresentando 1,65V para aceleração nula em sua saída. Apesar de ser utilizado um acelerômetro tri-axial apenas os eixos X e Y são utilizados nesse projeto. A sensibilidade típica é de 440mV/g para a escala de medida até 3g que foi configurada no acelerômetro. Usando novamente um microcontrolador AT89S52 e um conversor analógico-digital ADC0838 amostra-se a tensão de saída numa faixa de 0,5V até 3,3V nos dois eixos utilizados nesse projeto do acelerômetro para que seja possível fazer a decisão de qual deve ser o movimento da cadeira. Na Figura 3.2 apresentam-se as tensões de saída para as diferentes posições do acelerômetro, segundo seu fabricante.

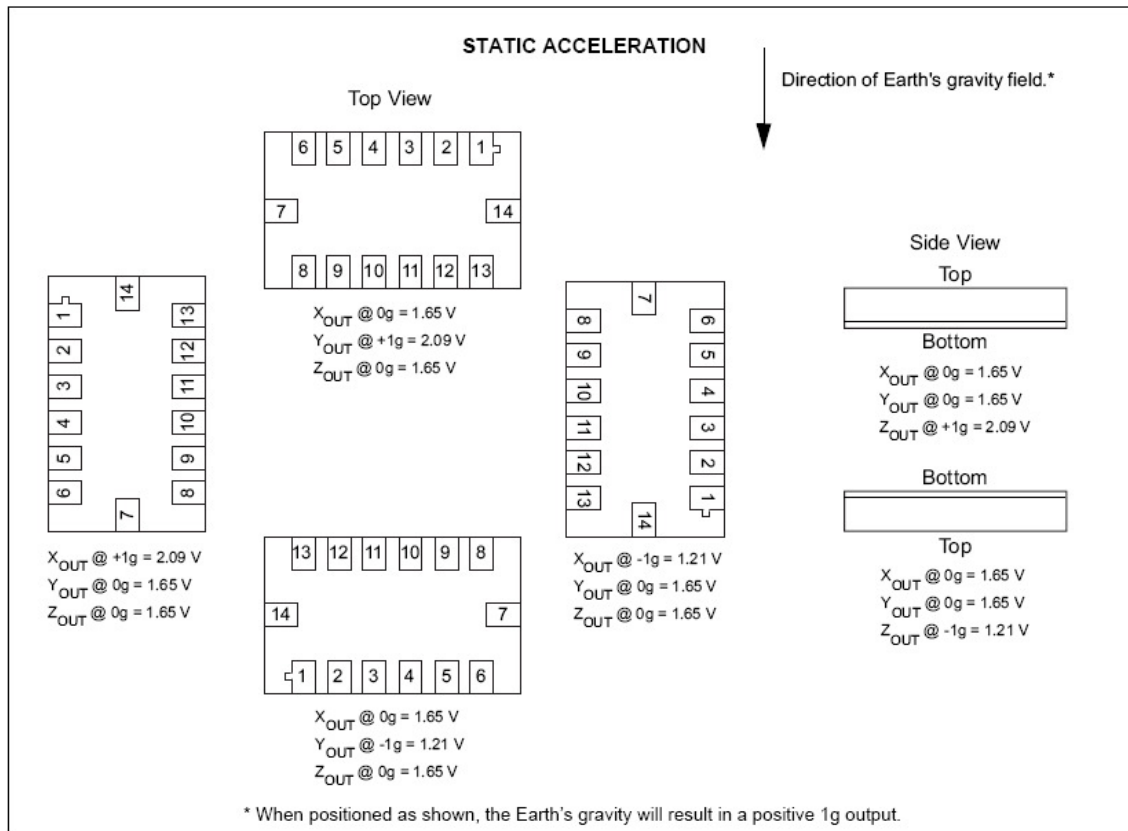


Figura 3.2 – Resposta em tensão do acelerômetro.

Fonte Freescale Semiconductor [8].

Conforme foi verificado, foi necessário filtrar as medidas do acelerômetro para que não ocorra um movimento indesejado. Para cada decisão são realizadas 10 medidas consecutivas e a média delas é utilizada para decidir o próximo movimento da cadeira. Na Figura 3.3 é apresentado o esquemático da placa do circuito do PWM do acelerômetro proposta.

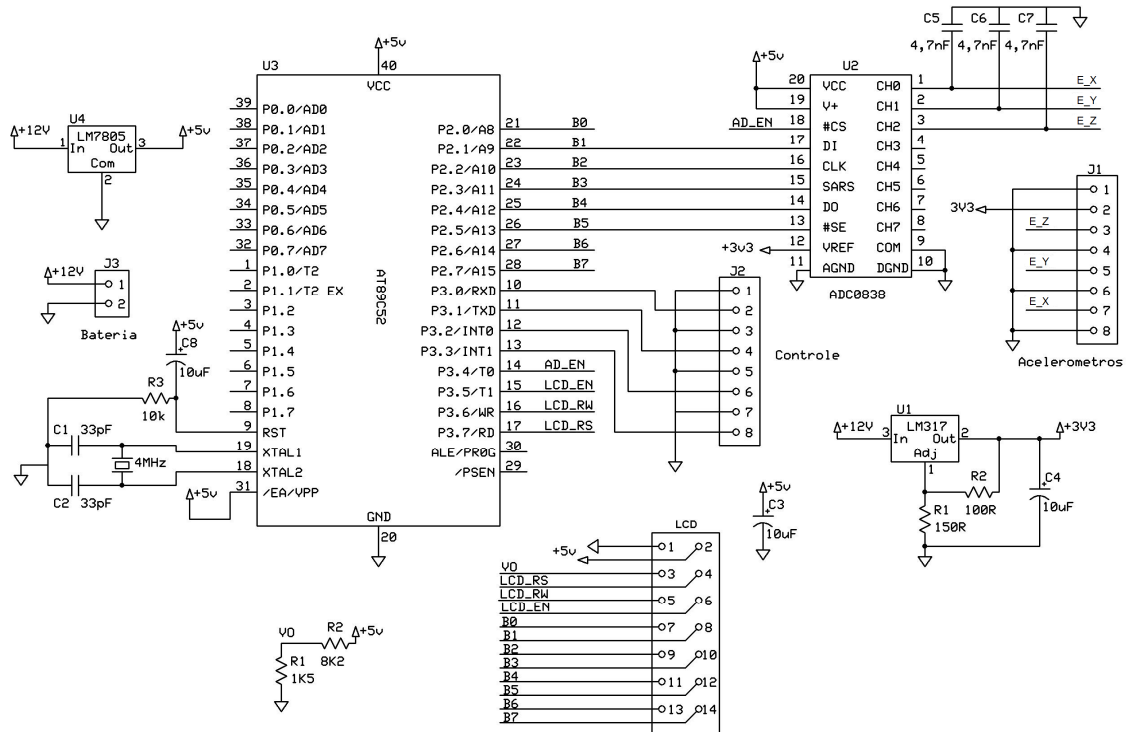


Figura 3.3 – Esquemático da placa de Controle do Acelerômetro.

O ADC0838 opera com frequência de *clock* máxima de 400kHz, por isso foi necessário criar uma função de atraso para dividir por 3 a frequência de trabalho do microcontrolador.

Uma característica importante é o erro máximo da medida da inclinação, usando como referência de tensão 3,3V no ADC0838 (8 bits) tem-se o erro máximo de 13mV logo em relação a inclinação o erro máximo é de  $1,7^\circ$  utilizando a equação  $V_{\text{erro}} = 0,44 * \text{sen}(\alpha_{\text{erro}})$ . A caracterização do movimento é individual para cada usuário, sendo calibrado ao inicializar o sistema, assim, a movimentação da cadeira é realizada conforme a possibilidade de movimento da cabeça do usuário. Os valores obtidos na calibração são utilizados como limites para a decisão sobre qual movimento deve ser realizado.

Um LCD de 2 linhas e 16 caracteres apresenta ao usuário as mensagens indicando quais movimentos são necessários para calibrar o sistema, guiando o usuário a realizar uma



calibração válida. Caso a calibração feita não seja válida o software ficará em *loop* até que seja realizada uma calibração aceitável. Qualquer valor medido na calibração que não possua valor absoluto maior que  $3,4^\circ$  em relação a posição de repouso será considerado inválido e uma nova calibração será necessária.

Na Figura 3.4 e 3.5 é apresentada a placa de controle do acelerômetro e no Anexo A.5 o seu *layout*. Na Figura 3.6 é apresentado o acelerômetro disposto na parte superior da cabeça do usuário fixado ao um chapéu.



Figura 3.4 – Foto da placa do circuito do PWM do acelerômetro.

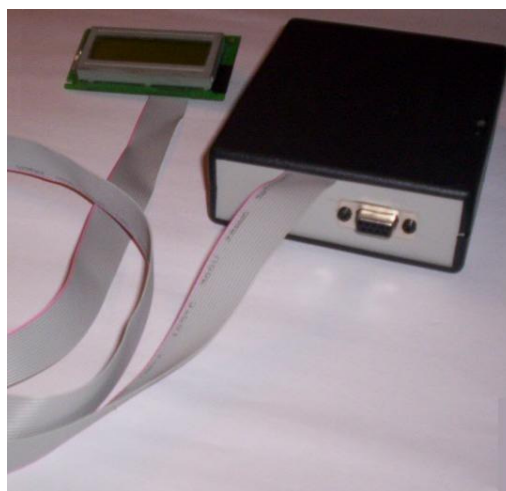


Figura 3.5 – Foto da placa do circuito do PWM do acelerômetro concluída.

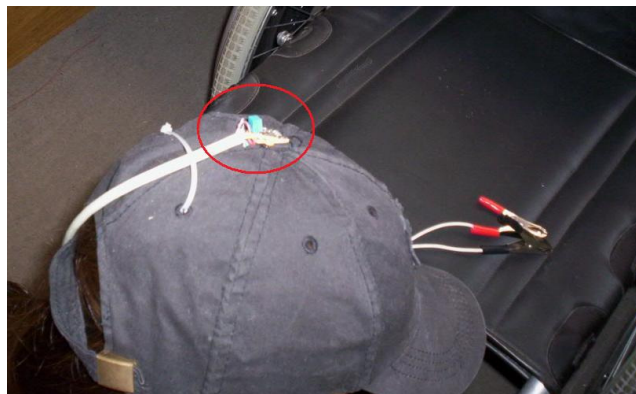


Figura 3.6 – Foto do acelerômetro disposto no topo da cabeça do usuário.

Na Figura 3.7 são apresentadas as mensagens que o display apresenta ao usuário para auxiliá-lo durante o processo de calibração. A obtenção do valor limite para cada movimento é realizado três segundos após a mensagem aparecer no display, sendo assim não é necessário o uso de botões o que seria conflitante com a idéia principal do projeto.



Figura 3.7 – Foto das mensagens apresentadas durante o processo de calibração.

### 3.2.2 Circuito do PWM

O circuito do PWM é a parte do projeto responsável por gerar o PWM para os motores de forma que a cadeira se movimente conforme o sinal oriundo do acelerômetro. A geração do PWM é feita pelo AT89S52 que é um microcontrolador de baixo custo, facilmente programável pela porta paralela do PC e que necessita de pouco *hardware* de apoio, pois possui EEPROM interna necessitando apenas de um cristal para referência de *clock* e um circuito RC para o *reset* do microcontrolador, pois esse precisa que, ao ligar, o sinal de reset fique em nível alto por pelo menos 2 ciclos de máquina.

O AT89S52 não possui capacidade para fornecer corrente diretamente aos optoacopladores por isso foi usado um buffer 74LS244 para prover a corrente necessária. O pino de habilitação das saídas do buffer foi conectado ao sinal P1.0 pois foi observado que o AT89S52 mantém em “1” lógico seus pinos durante o *power-up* o que colocaria em curto a alimentação, assim o software habilita o buffer apenas quando for realmente necessário deixando as saídas em alta impedância até o momento que o *software* passar a controlar o microcontrolador.

O controle possui 4 sinais de entrada sendo um para cada movimento da cadeira, possuindo 4 saídas para cada placa de acionamento tendo no total 8 saídas. Cada pino de entrada da placa do PWM possui diodos de proteção contra tensões superiores a 5V e tensões inferiores a 0V para evitar que níveis de tensão não aceitáveis cheguem ao microcontrolador.

Na Figura 3.8 temos o esquemático da placa do PWM proposta.

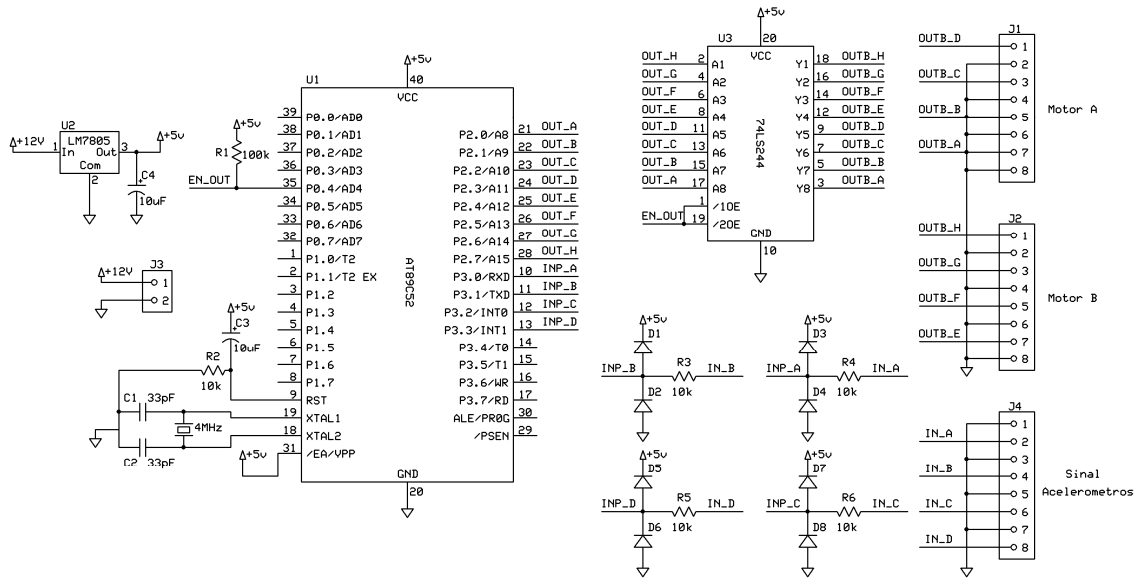


Figura 3.8 – Esquemático da placa do PWM.

Após receber o sinal do acelerômetro indicando para qual lado deve movimentar a cadeira, a placa do PWM inicia o movimento mantendo o PWM com ciclo ativo constante após um período de transição de 45  $\mu$ s. Se o movimento desejado for de andar para frente os dois motores são acionados com o mesmo ciclo ativo fazendo as duas rodas girarem igualmente, o mesmo é feito para movimentar-se para trás apenas invertendo a polaridade da tensão aplicada nos motores. O movimento para direita ou esquerda é realizado acionando o motor da roda oposta ao lado que se deseja andar e não acionando o do lado para onde se deseja andar, assim a cadeira gira em torno do seu eixo. Para um movimento contínuo o usuário necessita manter a cabeça inclinada durante todo o percurso, mas o ângulo de inclinação pode ser de apenas 3,4°. Na Figura 3.9 é apresentado como é realizada a movimentação da cadeira em função do comportamento dos dois motores.

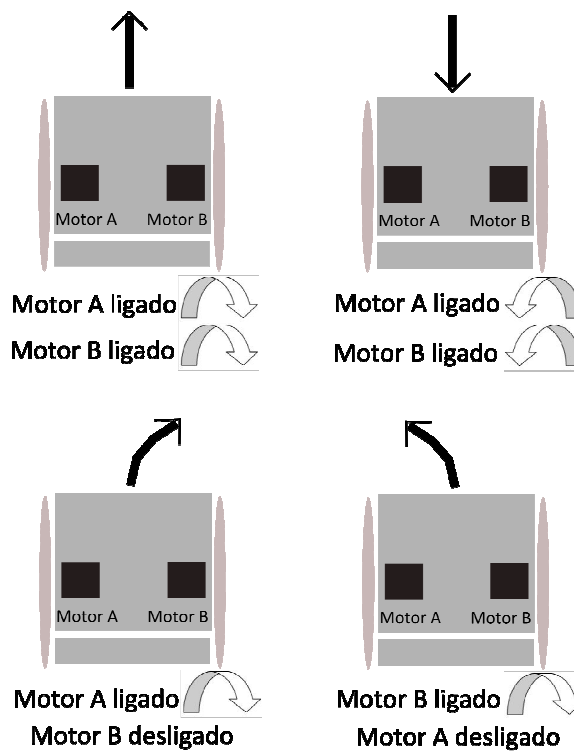


Figura 3.9 – Diagrama de movimentos da cadeira: a seta preta indica o sentido de movimento da cadeira e a branca indica o sentido de rotação dos motores.

A alimentação disponível para todo o circuito é 12VDC e por isso foi necessário regular a tensão de entrada para 5VDC (tensão de funcionamento do AT89S52 e do 74LS244) usando um LM7805 por ser de baixo custo e fornecer corrente suficiente para a aplicação. O LM7805 é um regulador de tensão, com tensão de saída fixa em 5V tendo como requisito que a tensão da entrada seja maior que 7V e menor que 35V, respeitando a potência dissipada sobre ele. Para o circuito do PWM a corrente máxima é de 60mA, assim temos 0,5W sendo dissipado sobre o LM7805. Como o circuito fica fechado dentro de uma caixa foi colocado dissipador no LM7805 e furos na mecânica para ventilação.

Na Figura 3.10 e 3.11 é apresentada a placa do circuito do PWM, que possui todos os conectores do tipo barra de pinos convertendo-se em conectores DB9 na caixa plástica e no Anexo A.4 é apresentado o *layout* do circuito.



Figura 3.10 – Foto da placa do circuito do PWM.



Figura 3.11 – Foto da placa do circuito do PWM concluída.

### 3.2.3 Circuito de Acionamento

O circuito de acionamento recebe o sinal PWM oriundo do controle e o utiliza para gerar os sinais de *gate* nos *mosfets*, acionando os motores de maneira que a velocidade cresça lentamente e não em um degrau. O sinal aplicado aos motores é um sinal de amplitude igual a  $\pm 12\text{VDC}$  e frequência  $4,7\text{kHz}$  com período ativo variável para controle da velocidade com o

motor agindo como filtro passa baixas do sistema, pois ele não consegue responder a estímulos de alta frequência utilizando, assim, somente o nível médio desse sinal.

Este circuito é baseado em opto-acopladores para isolar os sinais de controle que tem nível TTL, dos sinais que são gerados dentro do bloco, que possuem nível de tensão igual a 12VDC, protegendo o sistema de controle e o acelerômetro. O opto-acoplador, além da isolamento, provê o ajuste dos níveis de tensão para acionamento dos *mosfets*, pois estes necessitam de  $|V_{gs}| = 12V$ . No circuito de acionamento dos motores circula uma corrente nominal de 15A. Além disso, é possível através dos *mosfets* inverter a alimentação dos motores podendo, dessa maneira, controlar o sentido da rotação destes. Na Figura 3.12 é apresentado o esquemático do circuito de acionamento.

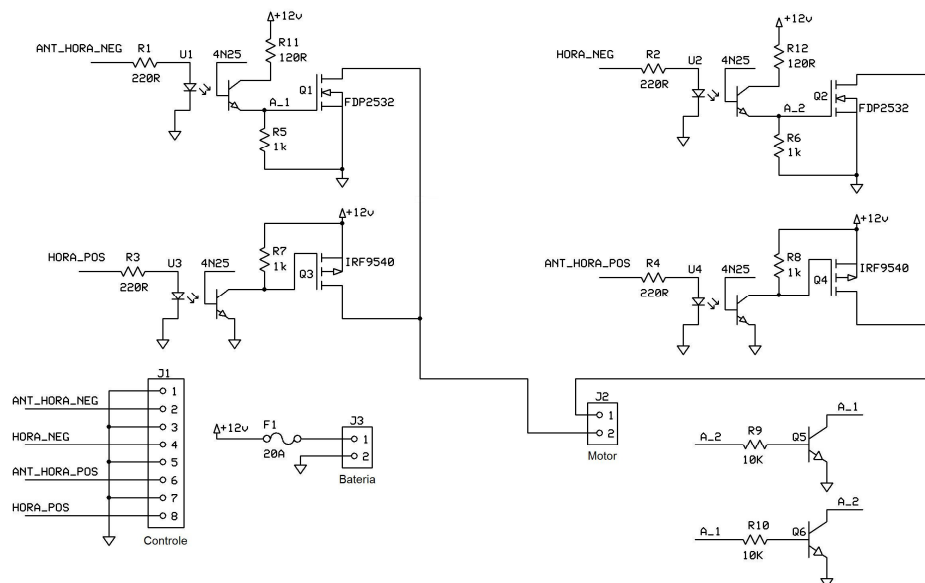


Figura 3.12 – Circuito de acionamento projetado.

A corrente de acionamento dos opto-acopladores é de 16mA com nível de tensão de acionamento TTL, ou seja, 5V, compatível com os buffers de baixo custo. O *mosfet* canal N utilizado é o FDP2532 que possui  $R_{on}$  muito baixo não ultrapassando 20m $\Omega$ , assim a

potência máxima dissipada, para o caso do mosfet ficar 100% do tempo acionado, é de 4,5W considerando a corrente igual a 15A. O *mosfet* canal P utilizado é o IRF9540 que possui  $R_{on}$  igual a 117m $\Omega$ , assim a potência dissipada sobre ele é de 26,5W considerando a corrente igual a 15A e o *mosfet* acionado 100% do tempo.

Para evitar que os *mosfets* esquentem muito foi montado neles um dissipador de calor, mas após alguns testes foi observado que, para aumentar a vida útil, seria melhor colocar dois *mosfets* canal P em paralelo, pois assim a corrente em cada um seria a metade da corrente do cálculo inicial (15A) fazendo a potência dissipada diminuir 4 vezes devido a dependência quadrática da potência em relação a corrente.

Na tentativa de ter uma placa de acionamento padrão de fácil reutilização foi utilizada uma barra de pinos como conector de entrada do sinal PWM, um conector para alimentação e um conector de porcelana com parafusos para conexão dos fios dos motores, facilmente desmontável e robusto o suficiente para suportar a corrente necessária para acionar os motores.

Para evitar que o *software* ou algum defeito de *hardware*, devido ao aquecimento, colocasse em curto a alimentação, foi projetado um circuito que não permite o acionamento do motor para trás e para frente ao mesmo tempo o que ocasionaria um curto na alimentação. Esse circuito pode ser visto na Figura 3.12, sendo formado pelos transistores Q5 e Q6 com seus resistores de polarização. Sempre que um dos *mosfets* canal N estiver conduzindo, o circuito de proteção inibe o *gate* do outro *mosfet* forçando que apenas um seja acionado de cada vez.

Essa parte do circuito não será montada em uma caixa plástica devido ao aquecimento. Na Figura 3.13 é apresentada a placa do circuito de acionamento e no Anexo A.3 o *layout* do circuito.





Figura 3.13 – Foto da placa protótipo do circuito de acionamento.

### 3.3 SOFTWARE

#### 3.3.1 Software do circuito do PWM

O software gera o sinal PWM conforme o sinal originado do acelerômetro. Foi configurada a porta P3 como porta de entrada usando apenas os 4 *bits* menos significativos para indicar os movimentos possíveis da cadeira, um “1” lógico nos 4 primeiros bits da porta P3 indica o movimento desejado fazendo o *software* iniciar a geração do sinal PWM para acionar os *mosfets* que ligam os motores. O *software* faz uma varredura de P3.0 até P3.3, respectivamente, buscando qual pino foi levado a “1” lógico, somente iniciando outro ciclo de varredura após o pino que estava em “1” voltar a “0”. O sinal PWM possui período fixo, variando apenas o tempo ativo. Isso é realizado usando um dos temporizadores disponíveis no AT89S52, fazendo a soma  $ton + toff = T$ , sendo *ton* o tempo ativo do sinal; *toff* o tempo inativo e *T* o período do sinal PWM que foi escolhido como sendo 1/4,7kHz. Utilizou-se a porta P2 como porta de saída, acionando os *mosfets* através do buffer. Cada motor necessita

de 4 *mosfets*, logo 4 sinais de acionamento, sendo dois sinais para cada sentido de rotação que se deseja acionar o motor.

Conforme foi mencionado, o circuito de proteção contra acionamento incorreto dos motores funciona sem intervenção do *software*, mesmo que ocorra algum erro ao inicializar o microcontrolador AT89S52 ou o *buffer* 74LS244 venha a falhar, o circuito de proteção impede o acionamento incorreto, desligando ambos. Ainda foi colocado um fusível para impedir correntes superiores a 20A pelo circuito.

Na Figura 3.14 está representado o fluxograma do *software* projetado para gerar o PWM e controlar os movimentos da cadeira.

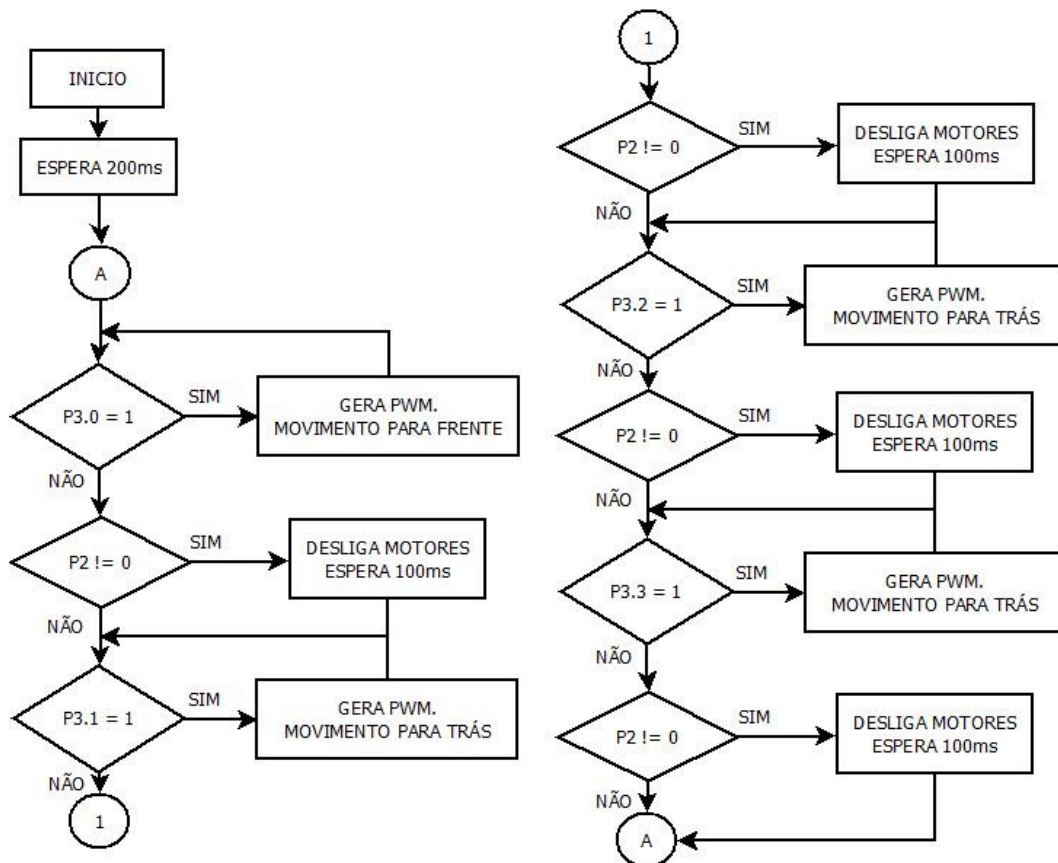


Figura 3.14 – Fluxograma do *software* de controle.

A inserção de um atraso de 100ms após o acionamento dos motores foi realizada para garantir que os *mosfets* do acionamento estarão totalmente desligados antes de realizar um novo movimento, isto se fez devido a capacitância do *gate* dos *mosfets* juntamente com o resistor de polarização formarem um circuito RC que diminui a resposta em frequência do circuito de *gate*, além é claro, do tempo de reação do motor ser relativamente grande. No Anexo A.1 é apresentado o código do *software* do circuito do PWM.

### 3.3.2 *Software* do circuito do Acelerômetro

O *software* do circuito do acelerômetro tem como objetivo digitalizar a saída analógica do acelerômetro para através de um processo de decisão determinar o movimento da cadeira. Cada decisão é tomada a partir da média de 10 medidas consecutivas do eixo “X” e “Y” para evitar movimentos espúrios caracterizando assim um filtro simples. O *software* compara ambas as medidas, do eixo X e Y, e testa qual é a medida que mais se afastou do valor médio obtido na calibração, assim toma a decisão de para qual lado deve movimentar a cadeira, caso as medidas não ultrapassem os limites da calibração o sistema não executa nenhum movimento. O ADC0838 é um conversor analógico-digital de comunicação serial possui oito canais com resolução de oito bits e tempo de conversão igual a oito pulsos de *clock* com frequência máxima de *clock* igual a 400kHz. O modo de acesso ao conversor é apresentado na Figura 3.15.

## Timing Diagrams (Continued)

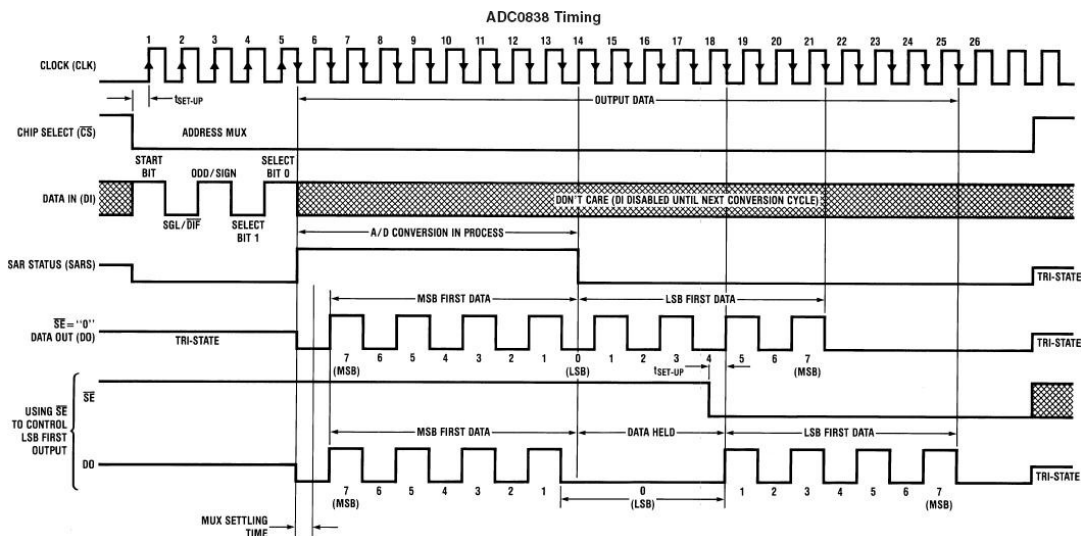


Figura 3.15 – Diagrama de tempo.

Fonte *National Semiconductor* [7].

A calibração é realizada ao inicializar o sistema. Um movimento de inclinação da cabeça para frente, para direita, para trás e para esquerda deve ser realizado ao inicializar conforme orientação das mensagens do LCD, assim os valores obtidos são usados como limite para iniciar o movimento. Essa característica do sistema permite flexibilizar seu uso para diversas pessoas.

O sistema possui uma validação da calibração, nem todos os valores são aceitos, como, por exemplo, não movimentar a cabeça, para qualquer lado, durante a calibração não é uma opção válida. Na Figura 3.16 é apresentado como é realizada a validação das medidas realizadas na calibração.

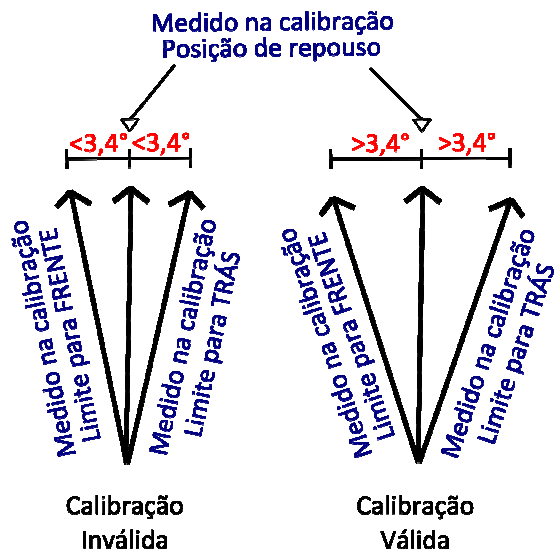


Figura 3.16 – Validação da calibração.

Caso a calibração seja validada pelo sistema, o programa avisa o usuário através do LCD e entra no *loop* principal, lendo continuamente a inclinação do acelerômetro buscando qual deve ser o próximo movimento. Na Figura 3.17 é apresentado o fluxograma do *software* do circuito de controle do acelerômetro.

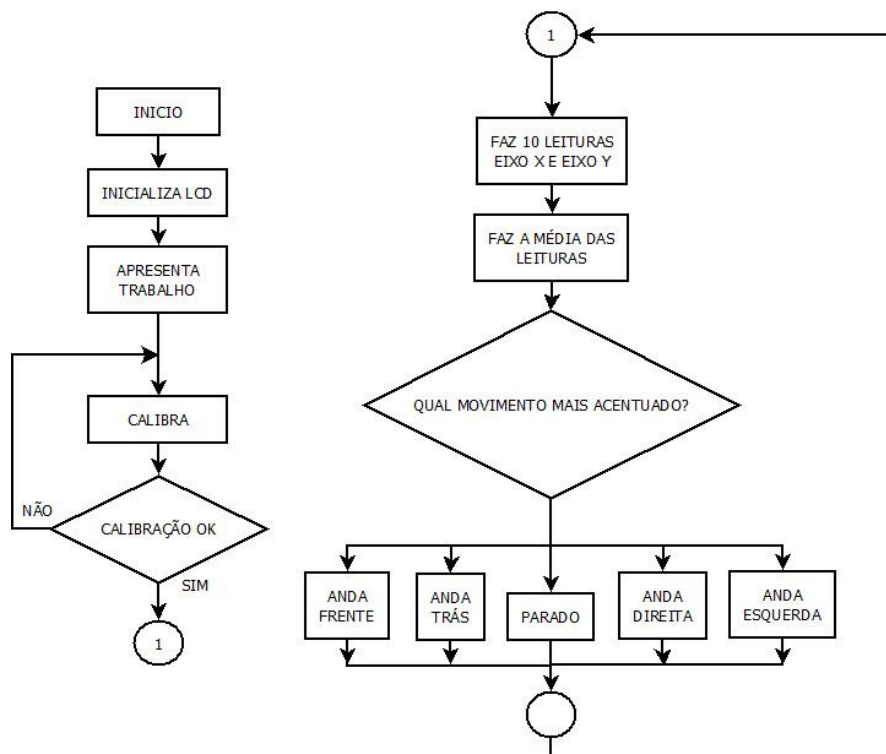


Figura 3.17 – Fluxograma do *software* do circuito de controle do acelerômetro.

No *loop* principal cada decisão tomada pelo circuito da placa de controle do acelerômetro é apresentada no display LCD. No Anexo A.2 é apresentado o código do *software* do circuito de controle do acelerômetro.

### 3.4 MECÂNICA

A cadeira utilizada é uma cadeira de rodas comum, não automatizada, que teve os eixos das rodas modificados, provendo suporte para o encaixe com o motor. Com a ajuda de um serralheiro foi possível montar uma estrutura para fixação do motor no eixo de cada roda, conforme apresenta a Figura 3.18 e Figura 3.19.

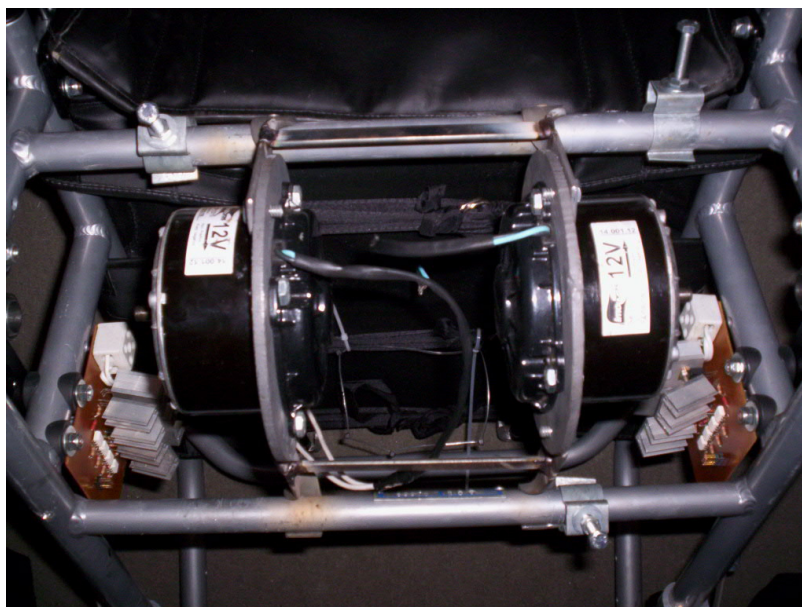


Figura 3.18 – Fixação dos motores nas rodas.

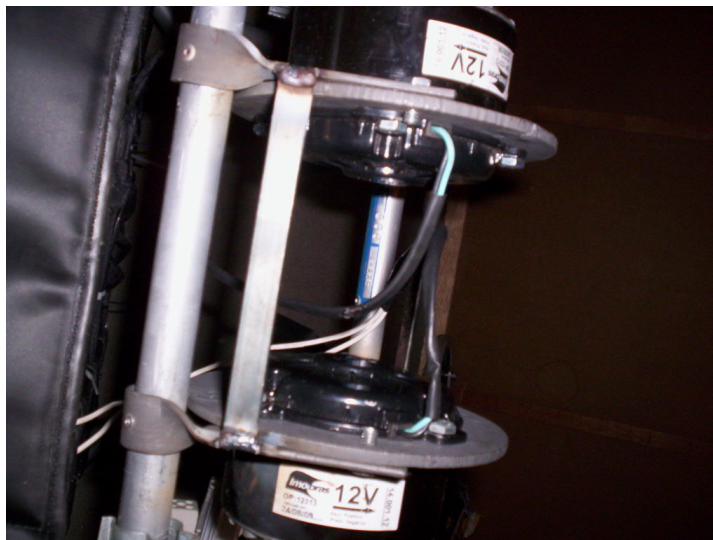


Figura 3.19 – Fixação dos motores nas rodas.

Os motores utilizados são da Imobras Industria de Motores Elétricos e possuem, conforme informações do fabricante, as características apresentadas na Tabela 3.1.

Tabela 3.1 – Características do motor Imobras 14.001.12 – Motor DC de 4 pólos.

Tensão	13 V
Corrente	15 A
Potência	195 W
Torque Nominal	0,4 N.m
RPM	2700/min
Massa	1,4 Kg
Sentido	L/R

Os motores foram acoplados diretamente nos eixos das rodas sem nenhum tipo de transmissão ou redução em função da indisponibilidade financeira. Assim os motores não tiveram força suficiente para mover a cadeira com qualquer peso, porém na continuidade deste projeto este detalhe será resolvido. O movimento das rodas só foi obtido levantando a cadeira para que as rodas girassem livremente sem nenhum tipo de peso para movimentar.

## 4 RESULTADOS ALCANÇADOS

### 4.1 CIRCUITO DE ACIONAMENTO

#### 4.1.1 Verificação do acionamento dos *mosfets*

Na Figura 4.1 é apresentada a simulação dos sinais de acionamento, sendo o canal laranja a onda triangular do gerador em 4,762kHz com período ativo de 145 $\mu$ s inserida na entrada do opto-acoplador e o canal azul o sinal de *gate* do *mosfet-P*, referente ao esquemático da Figura 3.12. A simulação foi realizada no *software* CircuitMaker6. Na Figura 4.2 é apresentada a medida do osciloscópio do gate do *mosfet-P* para comparação.

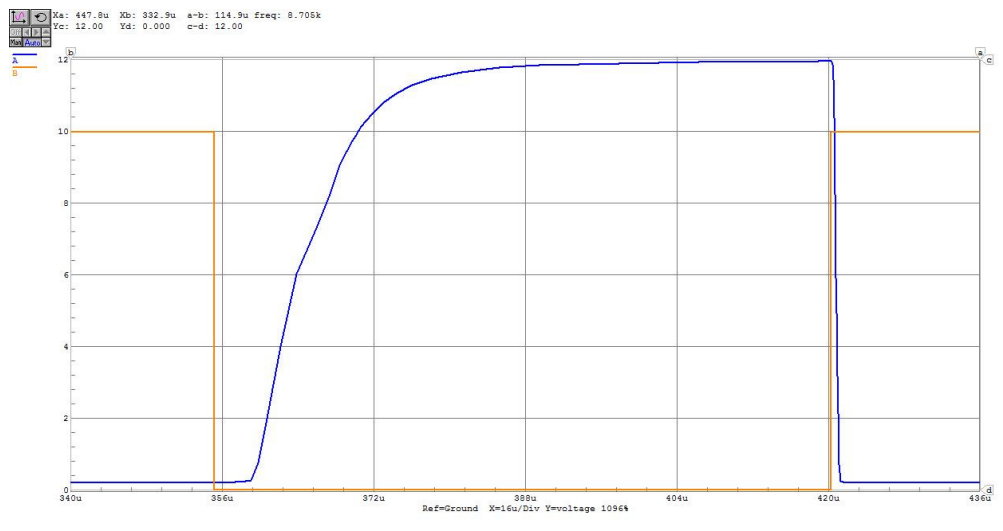


Figura 4.1 – Simulação da forma de onda do gate do *mosfet-P*.



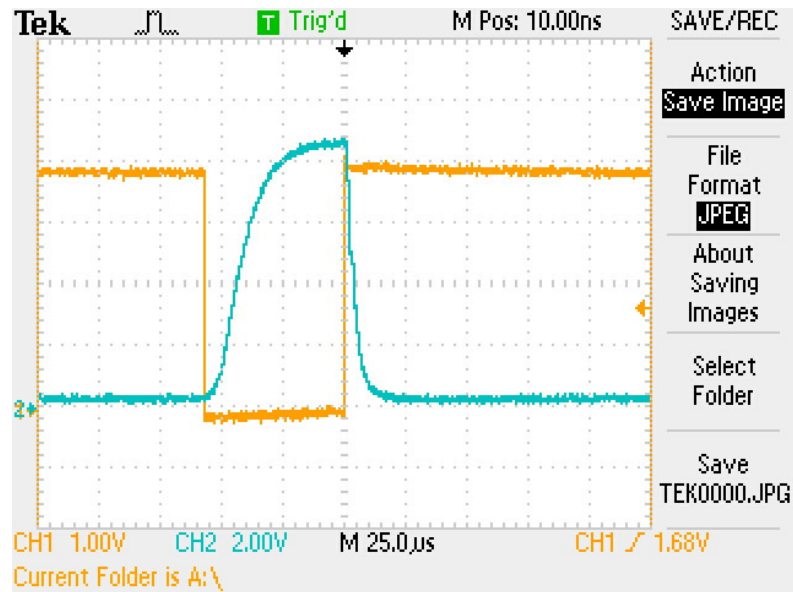


Figura 4.2 – Medida da forma de onda do gate do mosfet-P.

A mesma onda triangular de 4,762 kHz e com período ativo de  $145\mu\text{s}$  foi simulada e também aplicada no circuito protótipo para o mosfet-N. A Figura 4.3 e figura 4.4 apresenta a simulação do circuito e a medida com o osciloscópio respectivamente.

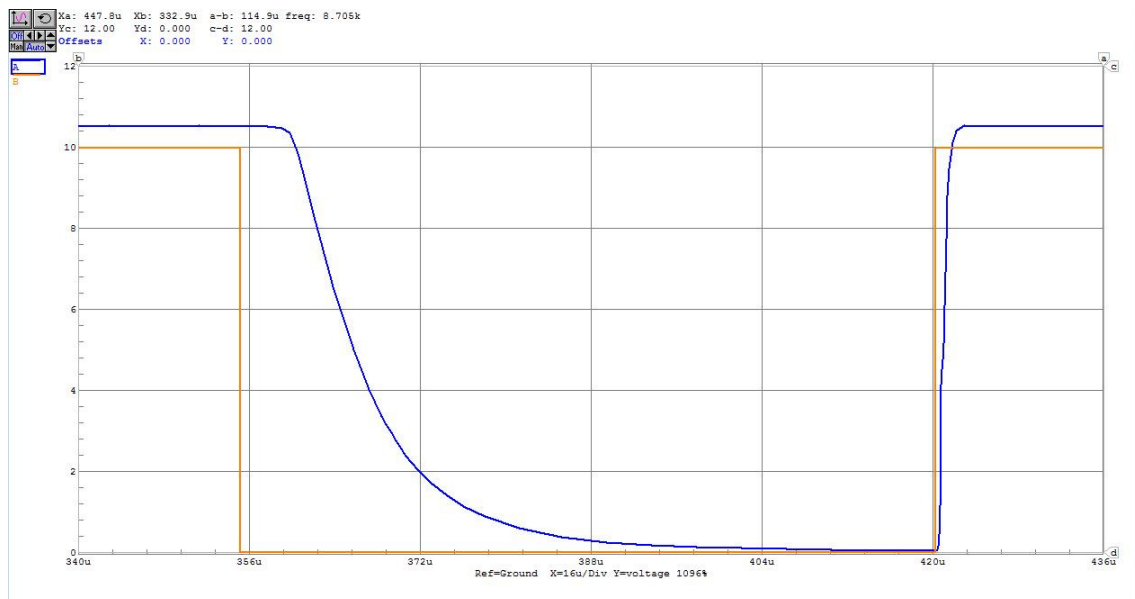


Figura 4.3 - Simulação da forma de onda do gate do mosfet-N.

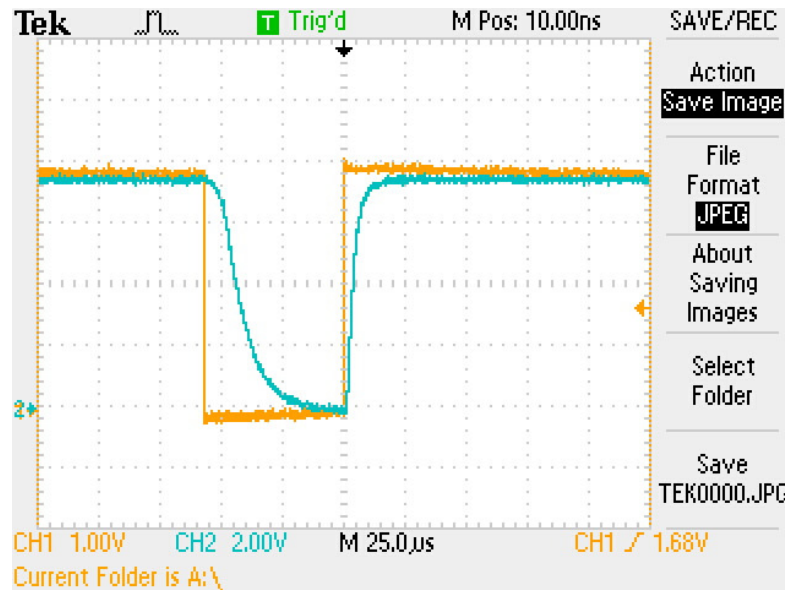


Figura 4.4 – Medida da forma de onda do gate do mosfet-N.

O período crítico do circuito de acionamento encontra-se durante a rampa de subida e descida dos sinais de *gate* dos *mosfets* canal-P e canal-N, pois é nesse momento que o *mosfet* possui resistência entre dreno e fonte maior que a resistência especificada no seu *datasheet* [9] e [10]. Nesse momento a dissipação de potência é maior que a calculada devido à alta resistência que o *mosfet* apresenta, por isso, a resistência equivalente de *gate* foi projetada para ser pequena o suficiente para que o circuito RC que é formado juntamente com o capacitor intrínseco do *mosfet* não deixe muito lenta a rampa de subida e descida do sinal.

Como pode ser visto na Figura 4.2 as rampas de subida e descida do *mosfet*-P no protótipo estão com  $15\mu\text{s}$  e  $8\mu\text{s}$  respectivamente. Analisando da mesma forma a Figura 4.4 as rampas de subida e descida do *mosfet*-N no protótipo estão com  $8\mu\text{s}$  e  $15\mu\text{s}$ .

## 4.2 CIRCUITO DO PWM

No circuito do PWM realizou-se um teste funcional. Configurou-se o circuito para gerar PWM para movimentar a cadeira para frente e também para trás, medindo assim todos os sinais de saída do circuito conforme pode ser observado na Figura 4.5.

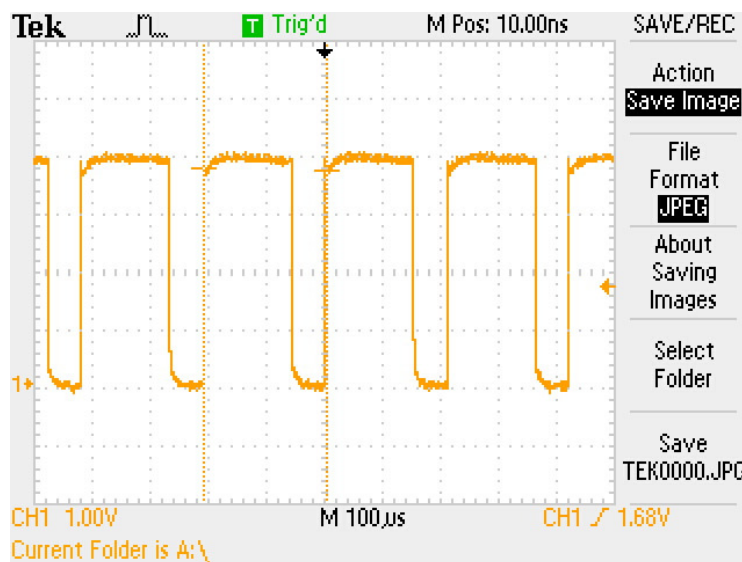


Figura 4.5 – Sinal PWM do circuito do PWM.

Com as medidas acima, pode-se validar o circuito também para movimentos para a esquerda e direita, visto que os sinais gerados para esses movimentos são iguais aos gerados para movimentar a cadeira para frente, a única diferença é que um dos motores não é acionado conforme o lado que se deseja movimentar. A geração do PWM está com aproximadamente 4762 Hz, como pode ser visto na Figura 4.3, tendo período de 210 $\mu$ s. O PWM gerado pelo circuito do PWM não ficou exatamente em 5000 Hz devido a atrasos gerados pelas funções de configuração do temporizador e devido ao micro-controlador estar operando com uma cristal de 4MHz. Com o uso do cristal de 4MHz o menor período de execução de uma tarefa é de 3 $\mu$ s, e como o período necessário para se ter 5000 Hz é de 200 $\mu$ s

poucas contagens com o temporizador são necessárias, assim as funções de configuração tornam-se expressivas, acentuando o efeito de atraso na contagem do período.

### 4.3 CIRCUITO DE CONTROLE DO ACELERÔMETRO

Para a verificação do circuito de controle do acelerômetro e também do sistema, é avaliada a saída em relação à entrada do sistema inteiro, para isso, realizou-se um teste funcional anotando os valores medidos com o multímetro durante a calibração e verificando se realmente após a calibração esses valores são utilizados como limites para caracterizar o movimento. A Tabela 4.2 apresenta os resultados observados que validam o sistema, pois nenhum erro ultrapassou  $1,7^\circ$ , conforme previsto para erro máximo do sistema. O valor calibrado foi medido diretamente nas saídas do acelerômetro e o início do movimento foi medido também nas saídas do acelerômetro no momento do início do movimento.

Tabela 4.2 – Verificação dos limites calibrados.

Direção	Valor calibrado(V)	Valor que iniciou o movimento(V)	Erro em graus
FRENTE	1,850	1,852	0,3
DIREITA	1,293	1,298	1,1
TRÁS	2,220	2,227	1,34
ESQUERDA	1,705	1,710	0,65

O erro em graus do sistema representa a diferença entre o ângulo que foi calibrado na inicialização e o ângulo em que iniciou o movimento das rodas. Este teste permite validar o sistema, pois foi observada a saída em relação a uma entrada imposta. Nesse teste foi também verificado que o movimento é realizado apenas quando o usuário inclina a cabeça e não ao girar para os lados. Olhar ao redor é uma necessidade do usuário que não será limitada pelo uso do sistema de controle através do inclinômetro.

#### 4.4 TESTE DO SISTEMA COM VOLUNTÁRIOS

Para verificação do sistema foi realizado um teste funcional com quatro voluntários. Avaliando-se os valores medidos na calibração em relação aos valores que iniciaram o movimento pode-se verificar a taxa de acerto e de erro do sistema. Foi pedido aos voluntários para que realizassem a calibração do sistema e usassem a cadeira normalmente. Assim foram anotados os valores de calibração diretamente nas saídas do acelerômetro e foram comparados com os valores medidos também nas saídas do acelerômetro no momento que foi iniciado o movimento desejado pelo voluntário.

A Tabela 4.3 apresenta os resultados obtidos durante os testes.

Tabela 4.3 – Verificação dos limites calibrados com voluntários.

Voluntário	Direção	Valor calibrado(V)	Valor que iniciou o movimento(V)	Erro em graus
A	FRENTE	2,010	2,009	0,13
	DIREITA	1,398	1,402	0,63
	TRÁS	2,205	2,209	0,73
	ESQUERDA	1,710	1,714	0,53
B	FRENTE	2,103	2,098	0,73
	DIREITA	1,478	1,480	0,28
	TRÁS	2,305	2,309	1,37
	ESQUERDA	1,729	1,733	0,53
C	FRENTE	2,059	2,050	1,25
	DIREITA	1,270	1,275	1,27
	TRÁS	2,276	2,279	0,76
	ESQUERDA	1,695	1,697	0,26
D	FRENTE	2,090	2,088	0,29
	DIREITA	1,374	1,377	0,50
	TRÁS	2,307	2,310	1,05
	ESQUERDA	1,705	1,710	0,66

Verificando a Tabela 4.3 pode-se verificar que o erro máximo encontrado foi de 1,37°, abaixo do erro máximo teórico devido ao erro do Conversor Analógico Digital. Desta forma,

pode-se validar o sistema visto que experimentalmente o erro máximo encontrado foi menor que o máximo teórico. Além disso, foi verificado durante o teste com os voluntários que o erro não é percebido pelo usuário provando que ele é pequeno o suficiente. Na Figura 4.6 é apresentado um voluntário utilizando a cadeira e realizando o teste de calibração.



Figura 4.6 – Voluntário testando o sistema

## 5 CONCLUSÕES

O protótipo projetado e implementado mostrou-se capaz de realizar o que foi proposto - controlar o movimento de uma cadeira de rodas através da medida da inclinação da cabeça do usuário. Os resultados obtidos no item 4.3 mostram que o controle apresentou erro inferior ao erro máximo teórico, assim a movimentação e calibração da cadeira é realizada com um erro máximo teórico de  $1,7^\circ$  e erro máximo experimental baseados nos testes do item 4.4 de  $1,37^\circ$ , o que possibilita garantir, com uma boa precisão, que os valores calibrados serão obedecidos pelo sistema. Cabe observar que a análise do erro experimental necessita uma avaliação mais rigorosa com equipamentos mais precisos, porém os resultados apresentados foram compatíveis com os objetivos deste trabalho.

Para pessoas com baixa mobilidade inclusive da cabeça a cadeira possibilita calibrar-se com pequenas inclinações, a partir de  $3,4^\circ$  de inclinação em relação à posição de repouso, garantindo que a inclinação não será mal interpretada, pois no mínimo inclinou-se o dobro do erro máximo.

O acionamento dos motores através do sinal PWM possibilita controle da velocidade, apesar disso, para os testes dos protótipos a velocidade foi mantida praticamente constante, a fim de se observar o funcionamento global do sistema e validar a ideia de controle através do inclinômetro.

Os motores com redução apresentaram ótima movimentação para poucos quilos de carga na cadeira, mas ao realizar testes com maior carga um eixo quebrou, devido ao material do eixo do motor não ser apropriado.

Mais testes devem ser feitos em relação à temperatura de trabalho dos *mosfets* em um ensaio com os motores acionados com carga sobre a cadeira, pois devido ao baixo torque dos motores montados a cadeira não conseguiu realizar movimentos com nenhum peso.

## 6 TRABALHOS FUTUROS

Foram observadas algumas melhorias que poderiam ser implementadas. O circuito de acionamento se fosse realimentado garantiria a mesma rotação para ambas as rodas quando o movimento fosse para frente ou para trás, possibilitando um movimento em linha reta, além disso, poderia corrigir ondulações no piso onde está a cadeira, como rampas ou descidas que poderiam aumentar a velocidade ultrapassando o limite de segurança. O software poderia agregar o controle de velocidade conforme a inclinação da cabeça do usuário, assim o cadeirante poderia controlar a velocidade do movimento desejado. O software ainda poderia ser complementado adicionando um estágio de parada da cadeira além de um sistema de segurança que desligasse o sistema em caso de travamento.

O circuito do acelerômetro poderia ser totalmente construído com tecnologia SMT, podendo ficar quase imperceptível na cabeça do usuário. Outra possível melhoria para o circuito do acelerômetro seria usar um acelerômetro tri-axial e em *software* possibilitar durante a calibração que o usuário escolha o local onde quer usar o acelerômetro, como por exemplo, topo da cabeça, parte posterior ou até integrado como um aparelho auditivo.

Algumas outras implementações poderiam ser realizadas, como andar em diagonal através do controle de velocidade das rodas, adicionar uma EEPROM para salvar os valores da calibração possibilitando assim cadastrar usuários da cadeira automatizada, verificador de carga na bateria e também controle de temperatura dos *mosfets* de acionamento para evitar possíveis problemas.

Projetar um circuito para carregar a bateria automaticamente, possibilitando ao usuário carregar a bateria apenas ligando a cadeira na rede elétrica. Assim, ao chegar em casa, a cadeira poderia ser carregada sem desmontar nenhuma parte do circuito.



## REFERENCIAS BIBLIOGRÁFICAS

- [1] Casa Civil, Presidência da República. Decreto N° 5.296 de 2 de dezembro de 2004 - [http://www.planalto.gov.br/ccivil/\\_ato2004-2006/2004/decreto/d5296.htm](http://www.planalto.gov.br/ccivil/_ato2004-2006/2004/decreto/d5296.htm)
- [2] Bersch, Rita. Tonolli, José Carlos. Tecnologia Assistiva - <http://www.assistiva.com.br/>, disponível em 20 de junho de 2010.
- [3] Silva, Otto Marques da. Del'Acqua, Ricardo José. Cadeiras de Rodas e sua evolução histórica - <http://www.crfaster.com.br/Cadeira%20Rodas.htm> disponível em 20 de junho de 2010.
- [4] Disability History Museum - <http://www.disabilitymuseum.org/> disponível em 20 de junho de 2010.
- [5] Ferreira, Ray. Mitologia e diversidade funcional: O artesão do Olimpo (final) - [http://diversidadefuncional.blogspot.com/2009\\_03\\_01\\_archive.html](http://diversidadefuncional.blogspot.com/2009_03_01_archive.html) disponível em 21 de junho de 2010.
- [6] Grabianowski, Ed. Como funciona a interface cérebro-computador - <http://informatica.hsw.uol.com.br/interface-cerebro-computador.htm> disponível em 21 de junho de 2010.
- [7] National Semiconductor em “*ADC0831/ADC0832/ADC0838 8-bit serial I/O A/D Converter with Multiplexer Options*”.
- [8] Freescale Semiconductor em “*±3g, ±11g Three Axis Low-g Micromachined Accelerometer*” – Technical Data –Rev. 02, 08/2007
- [9] International Rectifier em “*IRF9540N HEXFET® Power MOSFET*” – Technical Data
- [10] Fairchild Semiconductor em “*FDB2532/FDP2532/FDI2532 N-Channel PowerTrench Mosfet 150V, 79A, 16mΩ*” – Technical Data – August 2002.

# ANEXOS

## ANEXO A.1 – SOFTWARE DO CIRCUITO DO PWM

A linguagem de programação escolhida foi C para AT89S52 usando o ambiente de programação Keil uVision3. Abaixo o código do software do circuito do PWM.

```
/*-----*/
Arquivo: PWM.C
Autor; Daniel Fusco
Revisão: 01
Descrição da Versão: Movimento para os lados, para frente e para trás. Sem movimento na diagonal.
Description revision 01: Colocado tempo entre movimentos para proteger mosfets
Description revision 01: Versão inicial.
-----*/

#include <REG52.H>          //special function register declarations //
                           //for the intended 8051 derivative //
#include <stdio.h>         //prototype declarations for I/O functions //

/*-----ENTRADA-----//
/*-P3.0 = RXD - FRENTE----//
/*-P3.1 = TXD - TRÁS-----//
/*-P3.2 = INT0 - DIREITA---//
/*-P3.3 = INT1 - ESQUERDA--//
/*-----//

/*-----SAÍDA-----//
/*-P2.0 E P2.2 - FRENTE 1º MOTOR -//
/*- P2.1 E P2.3 - TRÁS 1º MOTOR -//
/*-P2.4 E P2.6 - FRENTE 2º MOTOR -//
/*- P2.5 E P2.7 - TRÁS 2º MOTOR -//
/*-----//
/*-- Considerando o motor 1º a direita de quem olha a cadeira por trás --*/

void main (void)
{
int i, ton, toff;
P0 = 0x0FF;
P2 = 0x00;
TMOD = 0x11;
P3 = 0x0FF;
for(i=0;i<=5;i++)          // ESPERA ACELEROMETRO INICIALIZAR //
{
TH1 = 0x000;
TL1 = 0x000;
TR1 = 1;                   //- Inicia contagem - //
while (TF1 != 1) {}       //- Espera contagem - //
TR1 = 0;                   //- Desliga contagem - //
TF1 = 0;                   //- Desliga flag do overflow - //
}
P0 = 0x00;
while(1)
{
P2 = 0x00;                //desliga motores
//toff = 241;            // lento para ver nos leds
toff = 19;                // rápido pra funcionar a 5khz
```

```

// Norte //
while ((P3 & 0x01) == 0x01)
{
    if (toff == 19)
    {
        P2 = 0x55;
        for(i=0;i<=2;i++)
        {
            TH1 = 0x000;
            TL1 = 0x000;
            TR1 = 1;           // Inicia contagem - //
            while (TF1 != 1) {} // Espera contagem - //
            TR1 = 0;          // Desliga contagem - //
            TF1 = 0;          // Desliga flag do overflow - //
        }
    }
    //for (i=0;i<=15;i++)
    //{
    ton = 256 - toff;
    //TH1 = ton;           //Valor alto da contagem - // Lento para ver nos leds //
    TH1 = 0x0FF;         //rápido pra funcionar a 5khz //
    //TL1 = 0x0FF;        //Valor baixo da contagem - //Lento para ver nos leds //
    TL1 = ton;           // Valor baixo da contagem - ///- Rápido pra funcionar
                        a 5khz //
    P2 = 0x00;           // Ativei motores para frente - //
    TR1 = 1;             // Inicia contagem - //
    while (TF1 != 1) {} // Espera contagem - //
    TR1 = 0;             // Desliga contagem - //
    TF1 = 0;             // Desliga flag do overflow - //
    // TH1 = toff;        // valor alto da contagem para ver nos leds- //
    TH1 = 0x0FF;         // Rápido pra funcionar a 5khz //
    TL1 = 256 - (20-toff); //valor alto da contagem rápido pra funcionar a 5khz//
    //TL1 = 0x0FF;        //Valor baixo da contagem - // Lento para ver nos leds //
    P2 = 0x55;           // Desligo motores - //
    TR1 = 1;             // Inicia contagem - //
    while (TF1 != 1) {} // Espera contagem - //
    TR1 = 0;             // Desliga contagem - //
    TF1 = 0;             // Desliga flag do overflow - //
    //}
    if (toff != 4) toff = toff - 1; else toff = 4;
}

if (P2 != 0x00)
{
    P2 = 0x00;           //desliga motores
    // Tempo entre acionamentos para evitar curto entre os mosfets
    TH1 = 0x000;
    TL1 = 0x0FF;
    TR1 = 1;             // Inicia contagem - //
    while (TF1 != 1) {} // Espera contagem - //
    TR1 = 0;             // Desliga contagem - //
    TF1 = 0;             // Desliga flag do overflow - //
}

//toff = 230;           // Lento para ver nos leds //
toff = 19;              // Rápido pra funcionar a 5khz //

//Sul //
while ((P3 & 0x02) == 0x02)
{
    if (toff == 19)
    {
        P2 = 0x0AA;
        for(i=0;i<=2;i++)
        {
            TH1 = 0x000;

```

```

        TL1 = 0x000;
        TR1 = 1;          //- Inicia contagem - //
        while (TF1 != 1) { //- Espera contagem - //
            TR1 = 0;      //- Desliga contagem - //
            TF1 = 0;      //- Desliga flag do overflow - //
        }
    }
    //for (i=0;i<=15;i++)
    //{
    ton = 256 - toff;
    //TH1 = ton;          //Valor alto da contagem - // Lento para ver nos leds //
    TH1 = 0x0FF;         //- rápido pra funcionar a 5khz //
    //TL1 = 0x0FF;       //Valor baixo da contagem - // Lento para ver nos leds //
    TL1 = ton;           //Valor baixo da contagem// Rápido pra funcionar a
                        5khz //
    P2 = 0x00;          //- Ativei motores para trás - //
    TR1 = 1;            //- Inicia contagem - //
    while (TF1 != 1) { //- Espera contagem - //
        TR1 = 0;        //- Desliga contagem - //
        TF1 = 0;        //- Desliga flag do overflow - //
        //TH1 = toff;    //- valor alto da contagem para ver nos leds- //
        TH1 = 0x0FF;    //- rápido pra funcionar a 5khz //
        TL1 = 256 - (20-toff); //valor alto da contagem rápido pra funcionar a 5khz //
        //TL1 = 0x0FF;   //Valor baixo da contagem - // Lento para ver nos leds //
        P2 = 0x0AA;     //- Desligo motores - //
        TR1 = 1;        //- Inicia contagem - //
        while (TF1 != 1) { //- Espera contagem - //
            TR1 = 0;    //- Desliga contagem - //
            TF1 = 0;    //- Desliga flag do overflow - //
        }
        if (toff != 4) toff = toff - 1; else toff = 4;
    }

if (P2 != 0x00)
{
    P2 = 0x00;          //desliga motores
    // Tempo entre acionamentos para evitar curto entre os mosfets
    TH1 = 0x000;
    TL1 = 0x0FF;
    TR1 = 1;           //- Inicia contagem - //
    while (TF1 != 1) { //- Espera contagem - //
        TR1 = 0;       //- Desliga contagem - //
        TF1 = 0;       //- Desliga flag do overflow - //
    }
}

//toff = 230;         // Lento para ver nos leds //
toff = 19;           //- Rápido pra funcionar a 5khz //

//Leste //
while ((P3 & 0x04) == 0x04)
{
    if (toff == 21)
    {
        P2 = 0x05;
        for(i=0;i<=2;i++)
        {
            TH1 = 0x000;
            TL1 = 0x000;
            TR1 = 1;          //- Inicia contagem - //
            while (TF1 != 1) { //- Espera contagem - //
                TR1 = 0;      //- Desliga contagem - //
                TF1 = 0;      //- Desliga flag do overflow - //
            }
        }
    }
    //for (i=0;i<=15;i++)
    //{
    ton = 256 - toff;

```

```

//TH1 = ton; //Valor alto da contagem - //Lento para ver nos leds //
TH1 = 0x0FF; // Rápido pra funcionar a 5khz //
//TL1 = 0x0FF; //Valor baixo da contagem - // Lento para ver nos leds //
TL1 = ton; //Valor baixo da contagem - Rápido pra funcionar a 5khz
//
P2 = 0x00; // Ativei motores para leste - //
TR1 = 1; // Inicia contagem - //
while (TF1 != 1) { // Espera contagem - //
TR1 = 0; // Desliga contagem - //
TF1 = 0; // Desliga flag do overflow - //
//TH1 = toff; // valor alto da contagem para ver nos leds- //
TH1 = 0x0FF; // Rápido pra funcionar a 5khz //
TL1 = 256 - (20-toff); //valor alto da contagem rápido pra funcionar a 5khz//
//TL1 = 0x0FF; //Valor baixo da contagem - //Lento para ver nos leds //
P2 = 0x05; // Desligo motores - //
TR1 = 1; // Inicia contagem - //
while (TF1 != 1) { // Espera contagem - //
TR1 = 0; // Desliga contagem - //
TF1 = 0; // Desliga flag do overflow - //
}

//}
if (toff != 4) toff = toff - 1; else toff = 4;
}

if (P2 != 0x00)
{
P2 = 0x00; //desliga motores
// Tempo entre acionamentos para evitar curto entre os mosfets
TH1 = 0x000;
TL1 = 0x0FF;
TR1 = 1; // Inicia contagem - //
while (TF1 != 1) { // Espera contagem - //
TR1 = 0; // Desliga contagem - //
TF1 = 0; // Desliga flag do overflow - //
}
//toff = 230; // Lento para ver nos leds //
toff = 19; // Rápido pra funcionar a 5khz //

//Oeste //
while ((P3 & 0x08) == 0x08)
{
if (toff == 19)
{
P2 = 0x50;
for(i=0;i<=2;i++)
{
TH1 = 0x000;
TL1 = 0x000;
TR1 = 1; // Inicia contagem - //
while (TF1 != 1) { // Espera contagem - //
TR1 = 0; // Desliga contagem - //
TF1 = 0; // Desliga flag do overflow - //
}
}
}
//for (i=0;i<=15;i++)
//{
ton = 256 - toff;
//TH1 = ton; // Valor alto da contagem - /// Lento para ver nos leds //
TH1 = 0x0FF; // Rápido pra funcionar a 5khz //
//TL1 = 0x0FF; //Valor baixo da contagem - //Lento para ver nos leds //
TL1 = ton; // Valor baixo da contagem - ///- Rápido pra funcionar
a 5khz //
P2 = 0x00; // Ativei motores para oeste - //
TR1 = 1; // Inicia contagem - //
while (TF1 != 1) { // Espera contagem - //
TR1 = 0; // Desliga contagem - //
TF1 = 0; // Desliga flag do overflow - //
}
}
}

```

```

//TH1 = toff;           //- valor alto da contagem para ver nos leds- //
TH1 = 0x0FF;          //- Rápido pra funcionar a 5khz //
TL1 = 256 - (20-toff); //valor alto da contagem rápido pra funcionar a
                      //5khz - //

//TL1 = 0x0FF;        //- Valor baixo da contagem - /// Lento para
                      //ver nos leds //

P2 = 0x50;            //- Desligo motores - //
TR1 = 1;              //- Inicia contagem - //
while (TF1 != 1) {}   //- Espera contagem - //
TR1 = 0;              //- Desliga contagem - //
TF1 = 0;              //- Desliga flag do overflow - //
//}
if (toff != 4) toff = toff - 1; else toff = 4;
}
if (P2 != 0x00)
{
    P2 = 0x00;          //-desliga motores
    // Tempo entre acionamentos para evitar curto entre os mosfets
    TH1 = 0x000;
    TL1 = 0x0FF;
    TR1 = 1;            //- Inicia contagem - //
    while (TF1 != 1) {} //- Espera contagem - //
    TR1 = 0;            //- Desliga contagem - //
    TF1 = 0;            //- Desliga flag do overflow - //
}
}
}

```

## ANEXO A.2 – SOFTWARE DO CIRCUITO DO ACELERÔMETRO

A linguagem de programação escolhida foi C para AT89S52 usando o ambiente de programação Keil uVision3. Abaixo o código do software do circuito de controle do acelerômetro.

```

/*-----
Arquivo: PWM.C
Autor; Daniel Fusco
Revisão: 02
Descrição da Versão: Leitura do acelerômetro eixo X e Y para decidir movimento da cadeira.
Description revision 01: São feitas várias leituras para fazer a média servindo como um filtro simples para movimentos espúrios.
Description revision 01: Versão inicial.
Description revision 02: Adicionado display LCD, adicionado processo de calibração e mensagens no display
-----*/

#include <REG52.H>          /*special function register declarations */
                          /*for the intended 8051 derivative */

#include <stdio.h>         /*prototype declarations for I/O functions */
#include <string.h>       /*prototype declarations for I/O functions */

sbit ad_en = P3^4;
sbit lcd_en = P3^5;
sbit lcd_rw = P3^6;
sbit lcd_rs = P3^7;

/*-----SINAIS AD-----*/
/*--P2.0 = B0 --*/
/*--P2.1 = DI e B1 --*/
/*--P2.2 = CLK e B2 --*/
/*--P2.3 = SARS e B3 --*/
/*--P2.4 = DO e B4 --*/
/*--P2.5 = #SE e B5 --*/
/*--P2.6 = B6 --*/
/*--P2.7 = B7 --*/
/*-----*/

/*-----SAÍDA-----*/
/*-P3.0 - FRENTE -*/
/*-P3.1 - TRAS -*/
/*-P3.2 - DIREITA -*/
/*- P3.3 - ESQUERDA -*/
/*- P3.4 - AD_EN -*/
/*- P3.5 - LCD_EN -*/
/*- P3.6 - LCD_RW -*/
/*- P3.7 - LCD_RS -*/
/*-----*/

void read_ad(unsigned char a);          /* DECLARANDO FUNÇÕES */
void delay();                          /* DECLARANDO FUNÇÕES */
void delay_1s();                       /* DECLARANDO FUNÇÕES */
void write_lcd_primeira_linha(unsigned char *str); /* DECLARANDO FUNÇÕES */
void write_lcd_segunda_linha(unsigned char *str); /* DECLARANDO FUNÇÕES */
void write_lcd_char(unsigned char letra); /* DECLARANDO FUNÇÕES */
void write_lcd_comand(unsigned char val); /* DECLARANDO FUNÇÕES */
void delay_lcd();                      /* DECLARANDO FUNÇÕES */

void main()
{
    unsigned char V_X, V_Y, SENTIDO_X, SENTIDO_Y,i,V_Xa, V_Ya, FRENTE, DIREITA, ESQUERDA, TRAS;
    float V_XAA, V_YAA, MEDIA_FRENTE_TRAS, MEDIA_DIREITA_ESQUERDA;

```

```

P3 = 0xD0;

/* INICIALIZA DISPLAY */
write_lcd_comand(0x38);
write_lcd_comand(0x38);
write_lcd_comand(0x06);
write_lcd_comand(0x0C);
write_lcd_comand(0x01);
/* APRESENTA TRABALHO */
write_lcd_primeira_linha(" PROJETO DE ");
write_lcd_segunda_linha(" DIPLOMACAO ");
delay_1s();
delay_1s();
write_lcd_primeira_linha(" DANIEL ");
write_lcd_segunda_linha(" FUSCO ");
delay_1s();
delay_1s();
/* CALIBRANDO O SISTEMA */
CALIBRA_NOVAMENTE:
write_lcd_primeira_linha(" PROCESSO ");
write_lcd_segunda_linha(" CALIBRACAO ");
delay_1s();
delay_1s();
delay_1s();
delay_1s();
write_lcd_primeira_linha("INCLINE A CABECA");
write_lcd_segunda_linha(" PARA FRENTE ");
delay_1s();
delay_1s();
delay_1s();
read_ad(0);
FRENTE = ACC;
/* write_lcd_primeira_linha("INCLINE A CABECA"); */
write_lcd_segunda_linha(" PARA DIREITA ");
delay_1s();
delay_1s();
delay_1s();
read_ad(1);
DIREITA = ACC;
/* write_lcd_primeira_linha("INCLINE A CABECA"); */
write_lcd_segunda_linha(" PARA TRAS ");
delay_1s();
delay_1s();
delay_1s();
read_ad(0);
TRAS = ACC;
/* write_lcd_primeira_linha("INCLINE A CABECA"); */
write_lcd_segunda_linha(" PARA ESQUERDA ");
delay_1s();
delay_1s();
delay_1s();
read_ad(1);
ESQUERDA = ACC;
write_lcd_primeira_linha("FIQUE NA POSICAO");
write_lcd_segunda_linha(" DE REPOUSO ");
delay_1s();
delay_1s();
delay_1s();
read_ad(0);
MEDIA_FRENTE_TRAS = ACC;
read_ad(1);
MEDIA_DIREITA_ESQUERDA = ACC;
delay_1s();
delay_1s();
delay_1s();
/* VALIDANDO CALIBRAÇÃO */

```



```

if(MEDIA_FRENTE_TRAS <= (FRENTE + 0x02) || MEDIA_DIREITA_ESQUERDA <= (DIREITA + 0x02) ||
MEDIA_FRENTE_TRAS >= (TRAS - 0x02) || MEDIA_DIREITA_ESQUERDA >= (ESQUERDA - 0x02))
{
    write_lcd_primeira_linha(" CALIBRACAO ");
    write_lcd_segunda_linha(" INVALIDA ");
    delay_1s();
    delay_1s();
    write_lcd_primeira_linha(" NECESSARIO ");
    write_lcd_segunda_linha(" RECALIBRAR ");
    delay_1s();
    delay_1s();
    goto CALIBRA_NOVAMENTE;
}
else
{
    write_lcd_primeira_linha(" CALIBRACAO ");
    write_lcd_segunda_linha(" FINALIZADA ");
    delay_1s();
    delay_1s();
}
/* LOOP PRINCIPAL */
while(1)
{
    V_XAA = V_YAA = 0x0000;

    for(i=0;i<=9;i++)
    {
        read_ad(0); /* MEDE EIXO X */
        V_X = ACC;
        read_ad(1); /* MEDE EIXO Y */
        V_Y = ACC;
        V_XAA = V_X + V_XAA;
        V_YAA = V_Y + V_YAA;
        /* delay_1s(); */
    }
    /* FILTRO SIMPLES */
    V_X = V_XAA/10; /* FAZ A MÉDIA DAS MEDIDAS DO EIXO X */
    V_Y = V_YAA/10; /* FAZ A MÉDIA DAS MEDIDAS DO EIXO Y */

    /*P1 = V_X; DEBUG
    P3 = V_Y;*/

    V_Ya = V_Xa = 0x00; /* IGUALANDO OS VALORES PARA O CASO DE NÃO ESTAREM
IGUAIS QUANDO NÃO EXISTIU MOVIMENTO ACENTUADO */

    if (V_X < FRENTE)/* FAZ A DIFERENÇA ABSOLUTA EM RELAÇÃO AO 0G PARA O EIXO X */
    {
        V_Xa = MEDIA_FRENTE_TRAS - V_X;
        SENTIDO_X = 1;
    }
    else if (V_X > TRAS)
    {
        V_Xa = V_X - MEDIA_FRENTE_TRAS;
        SENTIDO_X = 0;
    }

    if (V_Y < DIREITA)/* FAZ A DIFERENÇA ABSOLUTA EM RELAÇÃO AO 0G PARA O EIXO Y */
    {
        V_Ya = MEDIA_DIREITA_ESQUERDA - V_Y;
        SENTIDO_Y = 1;
    }
    else if (V_Y > ESQUERDA)
    {
        V_Ya = V_Y - MEDIA_DIREITA_ESQUERDA;
        SENTIDO_Y = 0;
    }
}

```

```

/* TESTA QUAL EIXO TEVE A MAIOR VARIAÇÃO E PARA QUAL SENTIDO FOI O
MOVIMENTO EM RELAÇÃO AO EIXO */
/* P3 = 0x00; */
if ((V_Xa > V_Ya) && (SENTIDO_X == 1))
{
    P3 = 0xD1;        /* FRENTE */
    write_lcd_primeira_linha(" MOVIMENTO: ");
    write_lcd_segunda_linha(" PARA FRENTE ");
}
else if ((V_Xa > V_Ya) && (SENTIDO_X == 0))
{
    P3 = 0xD2;        /* TRAS */
    write_lcd_primeira_linha(" MOVIMENTO: ");
    write_lcd_segunda_linha(" PARA TRAS ");
}
else if ((V_Ya > V_Xa) && (SENTIDO_Y == 1))
{
    P3 = 0xD4;        /* DIREITA */
    write_lcd_primeira_linha(" MOVIMENTO: ");
    write_lcd_segunda_linha(" PARA DIREITA ");
}
else if ((V_Ya > V_Xa) && (SENTIDO_Y == 0))
{
    P3 = 0xD8;        /* ESQUERDA */
    write_lcd_primeira_linha(" MOVIMENTO: ");
    write_lcd_segunda_linha(" PARA ESQUERDA ");
}
else
{
    P3 = 0xD0;        /* CASO NENHUM DELES PÁRA O MOVIMENTO */
    write_lcd_primeira_linha(" MOVIMENTO: ");
    write_lcd_segunda_linha(" NENHUM ");
}

/*for(i=0;i<=2;i++)
{
    delay_1s();
}*/

}

}

void delay_1s()
{
    unsigned char i;
    for (i=0;i<=4;i++)
    {
        TMOD = 0x11;
        TH1 = 0x00;
        TL1 = 0x00;
        TR1 = 1;          /*- Inicia contagem - */
        while (TF1 != 1) {} /*- Espera contagem - */
        TR1 = 0;         /*- Desliga contagem - */
        TF1 = 0;         /*- Desliga flag do overflow - */
    }
}

void delay()
{
    TMOD = 0x11;
    TH1 = 0x0FF;
    TL1 = 0x0FD;
    TR1 = 1;          /*- Inicia contagem - */
    while (TF1 != 1) {} /*- Espera contagem - */
    TR1 = 0;         /*- Desliga contagem - */
}

```

```

    TF1 = 0;          /*- Desliga flag do overflow - */
}

void delay_lcd()
{
    TMOD = 0x11;
    TH1 = 0x0FF;
    TL1 = 0x00;
    TR1 = 1;          /*- Inicia contagem - */
    while (TF1 != 1) {} /*- Espera contagem - */
    TR1 = 0;          /*- Desliga contagem - */
    TF1 = 0;          /*- Desliga flag do overflow - */
}

void read_ad(unsigned char a)
{
    unsigned char ABB = 0x00;
    unsigned char i;

    ad_en = 0;
    P2 = 0x01B;
    delay();

    P2 = 0x01A;      /* #CS=0 E DI=1 - INICIALIZA COMUNICAÇÃO */
    delay();
    P2 = 0x01E;      /* #CS=0 E DI=1 - BORDA DE SUBIDA */
    delay();
    P2 = 0x01A;      /* #CS=0 E DI=1 - BORDA DE DESCIDA */
    delay();

    P2 = 0x1A;       /* #CS=0 E DI=1 - INICIALIZA COMUNICAÇÃO */
    delay();
    P2 = 0x1E;       /* #CS=0 E DI=1 - BORDA DE SUBIDA */
    delay();
    P2 = 0x1A;       /* #CS=0 E DI=1 - BORDA DE DESCIDA */
    delay();

    if (a == 0)
    {
        P2 = 0x018;   /* #CS=0 E DI=1 - INICIALIZA COMUNICAÇÃO */
        delay();
        P2 = 0x01C;   /* #CS=0 E DI=1 - BORDA DE SUBIDA */
        delay();
        P2 = 0x018;   /* #CS=0 E DI=1 - BORDA DE DESCIDA */
        delay();
    }

    if (a == 1)
    {
        P2 = 0x1A;     /* #CS=0 E DI=1 - INICIALIZA COMUNICAÇÃO */
        delay();
        P2 = 0x1E;     /* #CS=0 E DI=1 - BORDA DE SUBIDA */
        delay();
        P2 = 0x1A;     /* #CS=0 E DI=1 - BORDA DE DESCIDA */
        delay();
    }

    P2 = 0x18;        /* #CS=0 E DI=1 - INICIALIZA COMUNICAÇÃO */
    delay();
    P2 = 0x1C;        /* #CS=0 E DI=1 - BORDA DE SUBIDA */
    delay();
    P2 = 0x18;        /* #CS=0 E DI=1 - BORDA DE DESCIDA */
    delay();

    P2 = 0x18;        /* #CS=0 E DI=1 - INICIALIZA COMUNICAÇÃO */
    delay();
}

```

```

P2 = 0x1C;      /* #CS=0 E DI=1 - BORDA DE SUBIDA */
delay();
P2 = 0x18;      /* #CS=0 E DI=1 - BORDA DE DESCIDA */
delay();

for(i=0;i<=7;i++) /* ESPERA CONVERSÃO FICAR PRONTA */
{
    P2 = 0x01C;   /* CLK = 1 */
    delay();
    P2 = 0x018;   /* CLK = 0 */
    delay();
}

P2 = 0x01C;     /* CLK = 1 */
delay();

ABB = ((P2 & 0x10)/0x10) | ABB; /* LE BIT 0 */
P2 = 0x018;     /* CLK = 0 */
delay();
P2 = 0x01C;     /* CLK = 1 */
delay();

ABB = ((P2 & 0x10)/0x08) | ABB; /* LE BIT 1 */
P2 = 0x018;     /* CLK = 0 */
delay();
P2 = 0x01C;     /* CLK = 1 */
delay();

ABB = ((P2 & 0x10)/0x04) | ABB; /* LE BIT 2 */
P2 = 0x018;     /* CLK = 0 */
delay();
P2 = 0x01C;     /* CLK = 1 */
delay();

ABB = ((P2 & 0x10)/0x02) | ABB; /* LE BIT 3 */
P2 = 0x018;     /* CLK = 0 */
delay();
P2 = 0x01C;     /* CLK = 1 */
delay();

ABB = ((P2 & 0x10)/0x01) | ABB; /* LE BIT 4 */
P2 = 0x018;     /* CLK = 0 */
delay();
P2 = 0x01C;     /* CLK = 1 */
delay();

ABB = ((P2 & 0x10)*0x02) | ABB; /* LE BIT 5 */
P2 = 0x018;     /* CLK = 0 */
delay();
P2 = 0x01C;     /* CLK = 1 */
delay();

ABB = ((P2 & 0x10)*0x04) | ABB; /* LE BIT 6 */
P2 = 0x018;     /* CLK = 0 */
delay();
P2 = 0x01C;     /* CLK = 1 */
delay();

ABB = ((P2 & 0x10)*0x08) | ABB; /* LE BIT 7 */
P2 = 0x018;     /* CLK = 0 */
delay();
P2 = 0x01C;     /* CLK = 1 */
delay();

P2 = 0x01B;
ad_en = 1;

```

```
    delay();

    ACC = ABB;
}

void write_lcd_primeira_linha(unsigned char *str)
{
    write_lcd_comand(0x80);
    while (*str != '\0')
    {
        write_lcd_char(*str);
        ++str;
    }
}

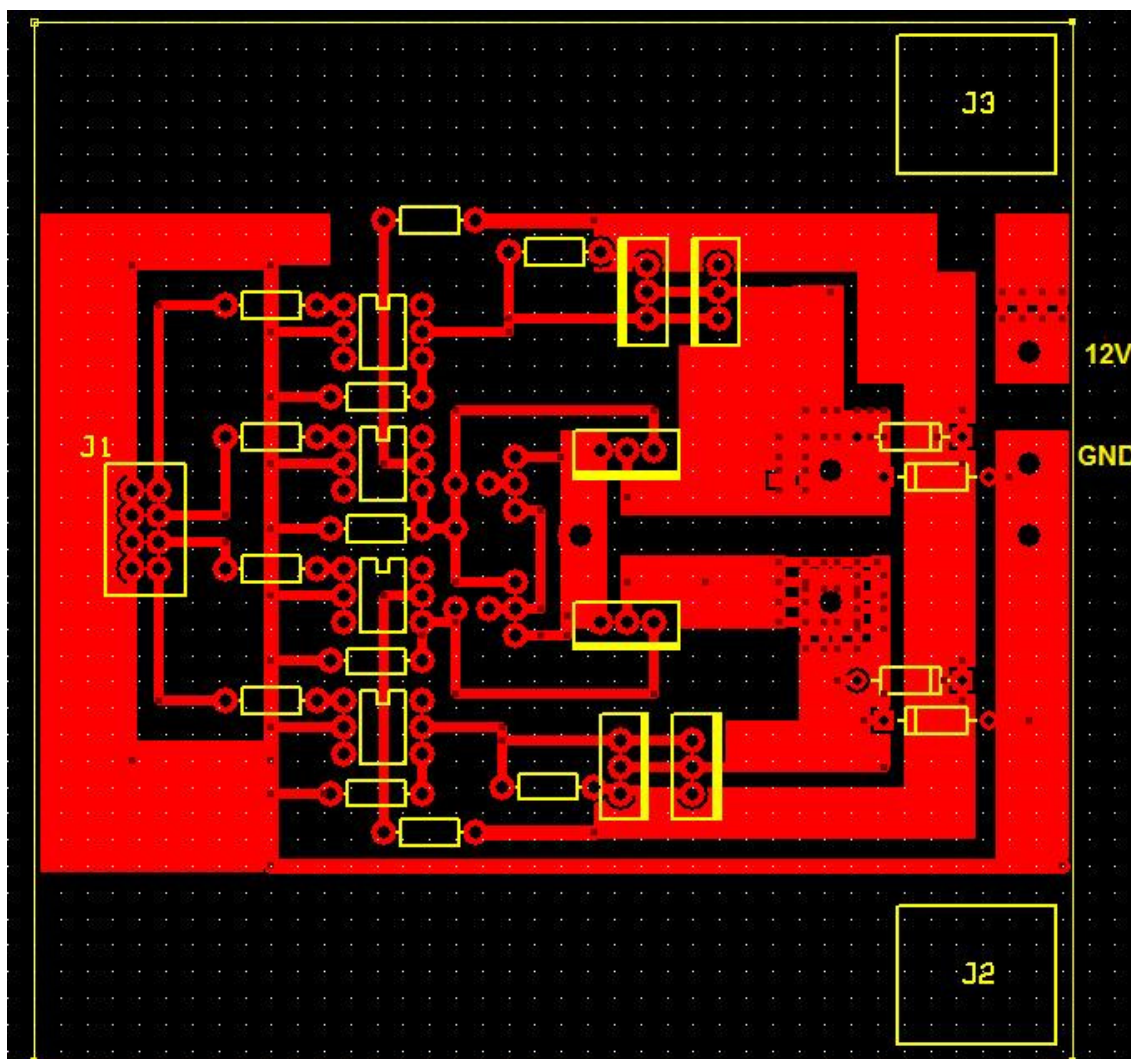
void write_lcd_segunda_linha(unsigned char *str)
{
    write_lcd_comand(0x0C0); /* Cursor no inicio da segunda linha */
    while (*str != '\0')
    {
        write_lcd_char(*str);
        ++str;
    }
}

void write_lcd_char(unsigned char letra)
{
    lcd_rw = 0;
    lcd_rs = 1;
    lcd_en = 1;
    delay_lcd();
    P2 = letra;
    delay_lcd();
    lcd_en = 0;
}

void write_lcd_comand(unsigned char val)
{
    lcd_rw = 0;
    lcd_rs = 0;
    lcd_en = 1;
    delay_lcd();
    P2 = val;
    delay_lcd();
    lcd_en = 0;
}
```

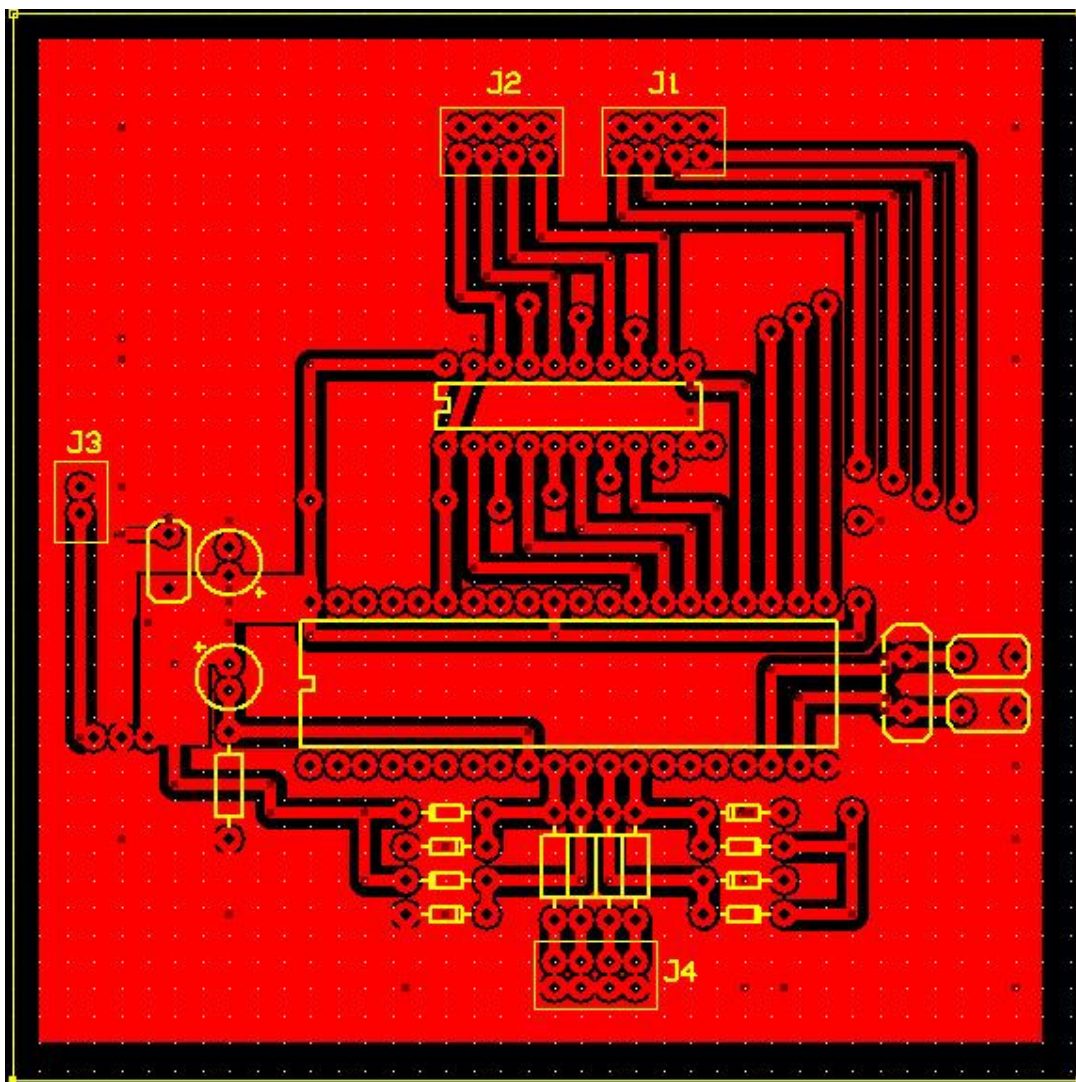
### ANEXO A.3 – PLACA DE CIRCUITO IMPRESSO DO CIRCUITO DE ACIONAMENTO

Abaixo é apresentado o *layout* do circuito de acionamento.



**ANEXO A.4 – PLACA DE CIRCUITO IMPRESSO DO CIRCUITO DO PWM**

Abaixo é apresentado o *layout* do circuito do PWM.



## ANEXO A.5 – PLACA DE CIRCUITO IMPRESSO DO CIRCUITO DO ACELERÔMETRO

Abaixo é apresentado o *layout* do circuito do acelerômetro.

