

3 de Dezembro de 2015

SISTEMA MESTRE-ESCRAVO DE UMA MÃO MECÂNICA ATUADA POR LIGA DE EFEITO TÉRMICO DE MEMÓRIA DE FORMATO

Gabriel Barsi Haberfeld

SISTEMA MESTRE-ESCRAVO DE UMA MÃO MECÂNICA ATUADA POR LIGA DE EFEITO TÉRMICO DE MEMÓRIA DE FORMATO

Aluno: Gabriel Barsi Habermfeld

Orientador: Marco A. Meggiolaro
Coorientador: Alexandre O. G. Barbosa

Trabalho apresentado com requisito parcial à conclusão do curso de Engenharia de Controle e Automação na Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil



Agradecimentos

Gostaria de agradecer a todos que tornaram este trabalho possível em todas as formas.

À minha mãe, que me possibilitou ser tudo que sou hoje como pesquisador e pessoa.

Ao corpo da PUC-Rio pela excelência em ensino e estrutura.

Ao professor Marco A. Meggiolaro pela excelente orientação como professor e amigo.

À professora Ana M. B. Pavani pelo excelente ensino e coordenação.

Ao meu co-orientador e chefe Alexandre Ormiga, pelo aprendizado que não acharia em nenhum outro lugar e toda a ajuda para tornar este e outros projetos possíveis.

À empresa INSFOR por fornecer experiência e recursos para completar meu curso e trabalho.

Aos membros e ex-membros da equipe RioBotz por todo o apoio e companhia.

A todos que conheci durante meus anos na PUC-Rio, por mudar a minha vida e me possibilitar mudar as deles.

Aos meus amigos, indispensáveis e sempre presentes quando precisei. Em especial, João Carlos Virgolino Soares por toda a ajuda em todos os sentidos, Igor Girsas pelo melhor espírito que conheço, Rodrigo Bianchi pelas importantes lições sobre o que a vida tem a oferecer, Eduardo Cota por compartilhar e apoiar a jornada, Daniel Freitas pela direta porém eficaz forma de me motivar e Julio Guedes por toda a compreensão em um período atarefado.

A todos que não mencionei por culpa da minha falha memória, mas que não me abandonaram nos momentos difíceis que passei, e que apesar de meus defeitos foram pessoas excelentes comigo.

"The most exciting phrase to hear in science, the one that heralds new discoveries, is not 'Eureka!' but 'That's funny...' "

- Isaac Asimov

Resumo

O mundo moderno traz diversos novos desafios para engenheiros, e hoje a robótica é um dos campos mais requisitados neste aspecto. Robôs são capazes de realizar infinitas tarefas diferentes e são extremamente úteis, principalmente em termos de segurança e assistência aos operadores humanos. Recentemente no desastre da planta nuclear de Fukushima robôs foram utilizados para inspecionar os danos causados pelo devastador terremoto que assolou o Japão naquele ano, uma vez que a radiação era extrema e mortal para seres humanos. Todos os dias próteses robóticas são implantadas em humanos que perderam membros periféricos. Tendo em mente estas necessidades crescentes este trabalho apresenta um novo modelo de sistema para operação remota (inspeção e manipulação de ambientes perigosos aos seres humanos) ou reabilitação. O sistema proposto é composto de uma mão mecânica atuada por fios de Nitinol e controladas por meio de realimentação a partir de um operador. Os modelos mecânico e eletrônico são propostos, construídos e testados e sua eficácia analisada por meio de ensaios diretos. O modelo prova-se flexível e funcional, observações sobre uso real são avaliadas e apresentadas.

Palavras-chave: Nitinol; SMA; prótese; mestre-escravo; mão

Abstract

A master-slave prosthetic hand system actuated by a heat shape memory alloy

The modern world brings multiple new challenges to engineers and, today, robotics is one of the most active fields in this aspect. Robots are capable of performing an infinite amount of tasks and are extremely useful especially when considering safety and assistance to human operators. Recently this need became obvious at Fukushima, Japan, where a devastating earthquake caused fatal damages to the nuclear power plant, requiring the use of robots that could withstand levels of radiation humans simply can't. Every day robotic prosthetic limbs are implanted on humans. Having these growing needs in mind this paper presents a novel system for remote operation or rehabilitation. The proposed system is composed of a mechanical hand actuated by Nitinol strands e controled by a remotely operated feedback system. The mechanical models are proposed, built and tested and their efficiency evaluated. The model proves flexible and functional, real-world use observations are made and presented.

Key-words: Nitinol; SMA; prosthetics; master-slave; hand

Lista de Figuras

1.1 Mão mecânica <i>bebionic 3 myoelectric</i> [18].	2
1.2 Mão mecânica de TUAT/Karlsruhe [19].	2
2.1 Estrutura óssea da mão humana [29].	4
2.2 Estrutura de tendões no dedo humano, setas indicam caminho das forças de extensão.	5
2.3 Estados de um PAM; relaxado, parcialmente contraído e completamente contraído, da esquerda para a direita.	6
2.4 Posições e sinais de controle referentes de um servomotor RC.	7
2.5 Motor de passo	7
2.6 Solenoide	8
2.7 Fases de um material [6].	9
2.8 Curva de calor do Nitinol [33, 34, 35].	10
2.9 Curvas de transformação do Flexinol [35]. Histerese aparente	11
2.10Evolução da saída de um filtro passa-baixas quando aplicado um PWM com $D = 75\%$	12
2.11Transdutor de movimentos angulares.	13
2.12Transdutor de movimentos lineares, também conhecido como <i>fader</i>	13
2.13Transdutor de esticamento, da empresa <i>adafruit</i>	14
2.14Transdutor de flexão, da empresa <i>Spectra Symbol</i> [36].	14
2.15Esquemático de funcionamento do transdutor de flexão[36].	16
2.16Multímetro <i>Fluke 87V Industrial Multimeter</i> configurado para medição de resistência em Ohms.	16
2.17Transdutor de flexão ligado ao multímetro e fixo á mesa.	17
2.18Transferidor de aço Starrett 19 garantindo angulação desejada.	17
2.19Dados e métodos de interpolação. Claramente o sensor exhibe comportamento não linear.	18
2.20Resultado com nível de confiança para interpolação exponencial de segunda ordem.	18
3.1 Diagrama de <i>pinout</i> do PIC18F4431 no pacote TQFP-44 [38].	21
3.2 Todas as saídas nos diferentes estados vistos na tabela 5 [38].	23
3.3 Aplicação típica do LM2575 de saída fixa (não ajustável) [41].	24
3.4 Divisor de tensão padrão.	25
3.5 Curva de sensibilidade ao variar-se R_1	25
3.6 Aplicação típica do LM3940 [42].	26
3.7 Curva de carga do gate do MOSFET Toshiba TPH4R008NH com $V_{DS} = V_{GS} = 5V$	27
3.8 Circuito de ativação de um MOSFET de canal N.	27
3.9 Diagrama funcional do MAX490 [40].	28
3.10Multiplexação do sinal PWM proveniente das portas CCPX.	28
3.11Janela principal do software ISIS.	29
3.12Janela principal do software ARES.	30
3.13Janela principal do software SolidWorks.	30
4.1 Render do modelo completo.	32
4.2 Vista isométrica 3D da palma.	33
4.3 Vistas e cotas principais em [mm] da palma, detalhes indicados.	33
4.4 Vista isométrica 3D da falange proximal.	34
4.5 Vistas e cotas principais em [mm] da falange proximal, detalhes indicados.	34
4.6 Vista isométrica 3D da falange média.	35
4.7 Vistas e cotas principais em [mm] da falange média, detalhes indicados.	35
4.8 Vista isométrica 3D da falange distal.	36
4.9 Vistas e cotas principais em [mm] da falange distal, detalhes indicados.	36
4.10Impressora 3D BFB 3D touch.	37
4.11Janela principal do software Axon 2.	38
4.12Configurações utilizadas na impressão das peças.	39
4.13Arquivo de impressão.	39
4.14Esquemático do estágio de potência para atuar uma carga diferencial RL	40
4.15Associação de portas no PIC18F4431 QFP.	40
4.16Circuito de aquisição analógica empregando um filtro RC dentro de um divisor de tensão.	40
4.17Modelo tridimensional da placa de controle.	43

4.18	Detalhe do arquivo Gerber com a nova tolerância para a camada <i>solder resist</i> .	44
4.19	Resultado da produção da placa de controle.	45
4.20	Resultado da produção da placa de aquisição.	46
4.21	Resultado da produção da placa de controle, sendo manuseada.	46
4.22	Resultado final da placa de controle, exibida ligada e funcionando.	47
4.23	Placa de controle, material isolante térmico e elétrico protege a parte lógica.	48
4.24	Detalhe das soldas dos pinos do PIC18F4431, o espaço entre a metade de cada pino (<i>pitch</i>) é de apenas 0.8mm.	48
5.1	Alocação das memórias por função do programa.	51
6.1	Tela do Osciloscópio exibindo a curva de carga do gate.	53
6.2	Dados importados e filtrados.	54
6.3	Análise PSD com o método de Welch.	54
6.4	Comparativo entre a previsão teórica e as medições.	55
6.5	Mesa de testes inicial controlando apenas uma junta, o sensor fixo á mesa fornece o <i>setpoint</i> .	56
6.6	Dedo completamente estendido.	56
6.7	Dedo completamente contraído. Esta configuração foi obtida artificialmente uma vez que o Flexinol não forneceu contração suficiente, conforme discutido.	57
9.1	Esquemático completo da placa de aquisição.	68
9.2	Layout completo da placa de controle, <i>output</i> do ARES.	69
9.3	Layout da camada superior da placa que aquisição.	70
9.4	Layout da camada inferior da placa que aquisição.	70
9.5	Painel completo de produção das placas, output visual dos arquivos Gerber RS-274X através do programa Gerbv.	77

Lista de Tabelas

1	Características elétricas e forças máximas do Flexinol fornecidas pelo fabricante [34]	10
2	Melhoramento relativo na velocidade relaxamento de diversos métodos [35].	11
3	Amostras de medições com o Multímetro	15
4	Comparação da família PIC18FXX31 [38].	19
5	Quatro configurações dos bytes OVDCOND e OVDCONS [38].	23

Sumário

1	Introdução	1
1.1	Mãos humanoides	1
1.2	Métodos de atuação	1
1.3	Nitinol como atuador	2
1.4	Corpo do trabalho	3
2	Visão geral	4
2.1	Anatomia da mão humana	4
2.1.1	Estrutura óssea	4
2.1.2	Músculos e tendões	4
2.1.3	Tradução para sistema robótico	5
2.2	Atuadores Tradicionais	5
2.2.1	Pneumática	5
2.2.2	Hidráulica	6
2.2.3	Servo Motores	6
2.2.4	Outros tipos comuns	7
2.3	Ligas de memória térmica de formato como atuadores	7
2.3.1	Nitinol (NiTi)	8
2.3.2	Flexinol	8
2.4	Sensores	12
2.4.1	Transdutores Resistivos de Deslocamento (potenciômetros)	12
2.4.2	Transdutor de Esticamento	13
2.4.3	Transdutor de Flexão	13
3	Métodos e Ferramentas	19
3.1	Microcontrolador PIC18F4431	19
3.1.1	Portas I/O	20
3.1.2	Interrupções	20
3.1.3	Timers e contadores	22
3.1.4	CCPX PWM	22
3.1.5	Power PWM - PWMX	22
3.1.6	Enhanced Universal Synchronous and Asynchronous Serial Receiver and Transmitter - EUSART	23
3.1.7	Conversor analógico-digital (ADC)	24
3.2	Texas Instruments LM2575SX-5.0	24
3.3	Texas Instruments LM3940	24
3.4	MOSFET Toshiba TPH4R008NH	26
3.5	Maxim MAX490CSA+ RS-485 Transceiver	27
3.6	Texas Instruments SN74LVC2G08DCTR	28
3.7	Ferramentas de CAD	28
3.7.1	PROTEUS Suite	28
3.7.2	SolidWorks	29
3.8	Compilador CCS PCWHD IDE	29
4	Implementações	31
4.1	O sistema mestre-escravo	31
4.2	Desenvolvimento do modelo mecânico	31
4.2.1	Modelo anatômico	31
4.2.2	Design CAD em 3D	31
4.3	Fabricação do modelo mecânico	36
4.3.1	Formatação do modelo para impressão	37
4.4	Desenvolvimento do circuito esquemático das placas	38
4.4.1	Placa de controle	38
4.4.2	Placa de aquisição	41
4.5	Desenvolvimento do layout das placas	41
4.6	Desenvolvimento do layout da placa de controle	41
4.7	Desenvolvimento do layout da placa de aquisição	42

4.8	Fabricação das placas impressas	43
4.8.1	Pré-produção	43
4.8.2	Geração dos arquivos Gerber/Excellon	43
4.8.3	Produção	44
4.8.4	Montagem	44
4.9	Considerações sobre as implementações	49
5	Software	50
5.1	Configuração da MCU	50
5.1.1	Configuração primária	50
5.2	Rotina de controle	51
5.2.1	Estratégia de controle	51
5.2.2	Controle via PWMX	51
5.2.3	Controle via CCPX	52
5.3	Rotina de comunicação	52
5.4	Código da placa de aquisição	52
6	Resultados	53
6.1	Performance do MOSFET	53
6.2	Performance geral das placas	53
6.3	Performance da mão robótica	55
7	Conclusão	58
8	Referências	59
9	Apêndices	62
9.1	Código em MATLAB para análise do sensor	62
9.2	Código em MATLAB para otimização da resistência divisora	63
9.3	Código em MATLAB para frequência máxima de chaveamento	64
9.4	Esquemático completo da placa de controle	64
9.5	Esquemático completo da placa de aquisição	68
9.6	Layout completo da placa de controle	69
9.7	Layout completo da placa de aquisição	70
9.8	Código da Placa de Controle	71
9.9	Código da Placa de Aquisição	75
9.10	Painel de produção completo das placas	77
9.11	Código em MATLAB para aquisição e comparação da performance do MOSFET	78

1 Introdução

Recentemente robôs estão cada vez mais presentes em vários aspectos da indústria e de nossas vidas. Robôs geralmente envolvem interação com o mundo físico por meio de um efetuator e sente o ambiente por meio de sensores para tomar decisões [1]. Na indústria essa interação é dada, normalmente, por de alguma ferramenta especializada para aumentar a eficiência geral do sistema.

Robôs que necessitam substituir um humano, no entanto, requerem um efetuator mais próximo ao modelo de uma mão humana, já que o mundo foi desenhado por e para seres humanos. Exemplos de situações assim podem ir desde a área médica (como próteses e robôs assistentes) até operação de plantas ou outros sistemas com uma interface voltada para humanos (como operação de válvulas, abrir portas, e assim por diante). Em situações assim um robô que consiga imitar completamente um humano é desejado, porém este objetivo ainda está distante e, no momento, é utópico.

Este é um problema tão marcante que os principais centros tecnológicos no mundo tem alguma linha de pesquisa voltada para este assunto. A DARPA (*Defense Advanced Research Projects Agency*) por exemplo realiza uma competição internacional completamente voltada para achar a plataforma que melhor realiza uma tarefa que um humano conseguiria. Centros como MIT, Carnegie-Mellon University e outros possuem linhas de pesquisa especializadas para lidar com estes assuntos.

Nossas mãos são as ferramentas que permitem nossa versatilidade e, portanto, um dos pontos principais para atingir estas metas. Por possuírem sensores e atuadores é natural definir uma mão robótica como um robô em si.

Neste capítulo será feita um breve revisão de trabalhos anteriores e fatores importantes envolvidos neste trabalho. Ao final o corpo do trabalho é apresentado.

1.1 Mãos humanoides

Tentativas de copiar uma mão humana são antigas. Existem duas direções principais de pesquisa e foco [2]. Uma é a mão humanoide voltada para próteses [2, 3, 4, 5] enquanto que a outra vertente foca em robôs humanoides [7, 8, 9, 10, 11, 12, 13]. Próteses tendem a valorizar uma integração maior com o ser humano, buscando estética, anatomia, custo baixo, facilidade de operação e baixo peso. Aplicações em robôs humanoides [14, 15, 16, 17] por sua vez priorizam precisão, rapidez e robustez.

Um exemplo do nível de sofisticação atingido pode ser visto no modelo *bebionic 3 myoelectric* [18] que pode ser vista na figura 1.1. Por ser um modelo comercial nenhum artigo foi publicado. Este exemplo demonstra o uso da tecnologia voltada para próteses, empregando as mais modernas tecnologias disponíveis.

1.2 Métodos de atuação

A robótica em geral possui uma ampla gama de tipos de atuação. Exemplos clássicos são atuadores elétricos, pneumáticos e hidráulicos, podendo existir exemplos mais exóticos e específicos para certas situações. Essa conjuntura não é diferente em uma mão robótica e, portanto, as pesquisas em como atuar estes sistemas é extremamente diversificada e não existe um modelo padrão [2] para o design destes sistemas.

A estratégia comum é copiar a cinemática da mão humana e restringir eixos de movimento para impor simplicidade. Um caminho desta estratégia são as mãos sub-atuadas, onde apenas um atuador é capaz de contrair mais de uma junta de forma manipular um objeto ou o ambiente. Este tipo de atuação pode ser visto em [3, 5, 9] e, embora restrinja certos movimentos facilita muito a operação e projeto dos sistemas. Um exemplo deste tipo de sistema é visto em [19] e a figura 1.2 exibe seu mecanismo. Podemos ver que apenas uma força **J** atua em todas as juntas.



Figura 1.1: Mão mecânica *bebionic 3 myoelectric* [18].

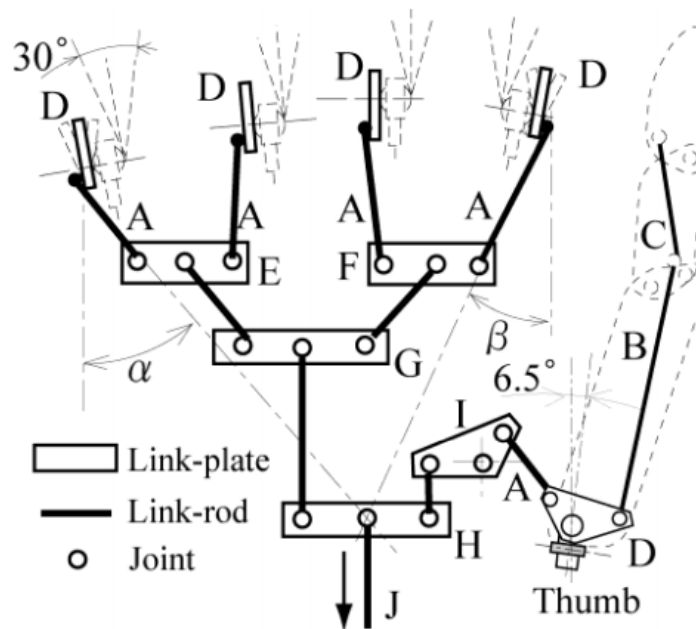


Figura 1.2: Mão mecânica de TUAT/Karlsruhe [19]

Normalmente são utilizados atuadores que realizam uma força de tração em uma haste ou cabo com o intuito de copiar o sistema biológico onde músculos puxam tendões. Estes tipos de atuação serão mais explorados no capítulo 2. No escopo deste trabalho será analisado um atuador mais exótico, porém mais próximo ao análogo humano.

1.3 Nitinol como atuador

Dentro do grupo de atuadores mais exóticos residem os músculos artificiais. Estes normalmente são da forma de músculos pneumáticos porém uma família ainda mais desconhecida é das ligas de memória térmica de formato. Músculos artificiais desta categoria se parecem com fios de aço e contraem em comprimento quando aquecidos além de uma determinada temperatura. Diversos compostos apresentam este comportamento e, dentre eles, o Nitinol (NiTi) é o mais amplamente utilizado pelo relativo baixo custo e facilidade de utilização. Nitinol é extremamente semelhante aos músculos biológicos em função. Nossos músculos são compostos de milhares de filamentos individuais e, analisados um a um, funcionam contraindo em comprimento de forma a gerar forças. Um filamento de Nitinol funciona da mesma forma e podemos utilizar os mesmos princípios anatômicos do corpo humano para extrair função.

Nitinol é um candidato muito desejável a primeira vista: possui custo relativamente baixo, simples, consegue exercer forças consideráveis, pequeno, leve e comercialmente disponível. É uma surpresa, então, ver que não existem dispositivos que conseguiram com sucesso utilizar Nitinol a longo prazo e de forma robusta. O motivo disto são as dificuldades que surgem em relação a resistência contra fadiga e controle preciso quando o seu parâmetro varia livremente mudanças ambientes.

Estudos que lidam com estes problemas são abundantes, porém apenas apresentam problemas ou não lidam com certos aspectos. Em [20] os autores apresentam um modelo funcional, mas não tratam os problemas de controle a longo prazo. Em [21] os autores lidam com baixas temperaturas com sucesso. Em [2] o autor apresenta [22, 23, 24, 25, 26, 27, 28] como exemplos onde, também, problemas foram descobertos. As ocorrências mais comuns são complicações decorrentes de histerese excessiva, alta dissipação de energia, baixa resistência a fadiga e baixa deformação.

1.4 Corpo do trabalho

Para uso em uma mão robótica é necessário obter controle sobre a deformação do Nitinol assim como conseguir repetição e resultados previsíveis por meio de um modelo mecânico robusto. Este trabalho mira atingir estes objetivos tomando uma metodologia simples e flexível, de forma a construir um sistema base para a pesquisa de um modelo refinado e avançado da solução proposta.

O capítulo 1 apresentou o problema. O capítulo 2 apresenta uma visão geral dos processos e mecanismos envolvidos, introduz e compara Nitinol frente a outros atuadores e explica a escolha de peças centrais ao trabalho como sensores e processadores. No capítulo 3 são desenvolvidas as ideias e decisões principais, dispositivos eletrônicos descritos e softwares de desenvolvimento apresentados. O esquemático da placa, implementação em hardware e desenvolvimento e fabricação do modelo mecânico é vista no capítulo 4, o software é desenvolvido no capítulo 5 juntamente com a lei de controle, resultados são apresentados no capítulo 6 e a conclusão é feita no capítulo 7.

2 Visão geral

Este capítulo fornecerá a base teórica e considerações necessárias para a evolução do trabalho. A anatomia da mão humana é apresentada para posteriormente ser utilizada como modelo para o sistema mecânico. Em seguida Atuadores são apresentados e comparados, onde observações são feitas. O final do capítulo foca em tipos de sensores que podem ser utilizados e sistemas eletrônicos necessários para a integração do projeto.

2.1 Anatomia da mão humana

A mão humana é composta de quatro dedos e um polegar e apresenta uma enorme complexidade de movimentos. Polegares opostos do tipo polpa-a-polpa são únicos aos seres humanos e é o que nos proporciona maior destreza e liberdade no manuseio e interação com o ambiente

2.1.1 Estrutura óssea

A estrutura óssea da mão humana passou por milhares de anos de evolução para chegar a uma disposição ótima de seus componentes. É natural, portanto, utilizar esta estrutura como base para qualquer sistema que procura imitar a mão humana. A figura 2.1 exibe a estrutura óssea da mão humana. As estruturas Carpais e Metacarpais compõem o pulso e a palma respectivamente. As três Falanges compõem os elos em cada dedo, polegar possui somente dois elos.

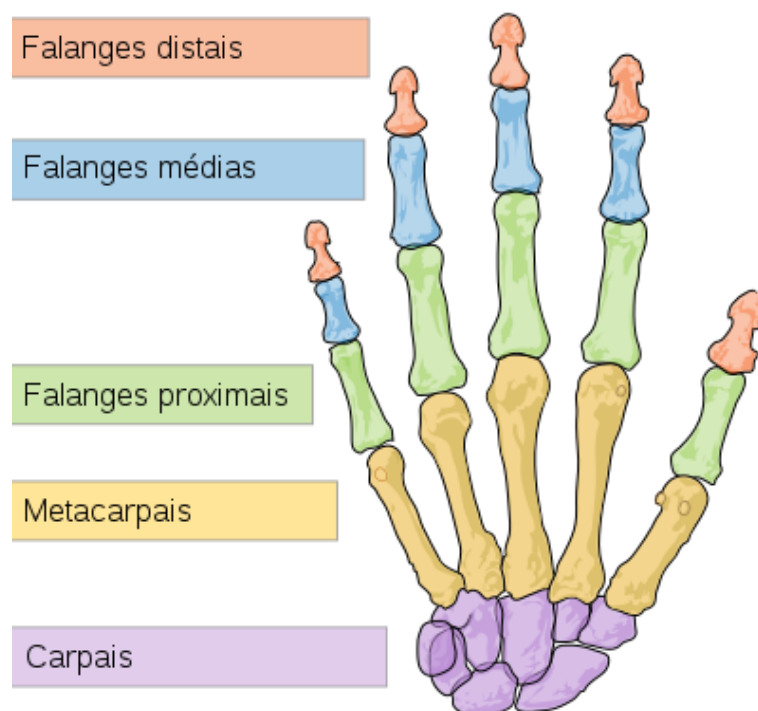


Figura 2.1: Estrutura óssea da mão humana [29]

Entre cada elo existe uma junta rotativa de pelo menos um grau de liberdade, já que algumas conseguem mais do que um eixo de rotação.

2.1.2 Músculos e tendões

A atuação das mãos humanas é feita por meio de músculos e tendões. A mão necessita exercer forças muito maiores do que as capazes pelo volume muscular que cabe na mesma,

por este motivo os músculos principais são colocados no antebraço e as forças que eles geram são transmitidas para os dedos por meio de tendões. A figura 2.2 exibe a estrutura de um dedo humano. Cada dedo possui dois tendões de flexão e dois de extensão, que atuam em três juntas.

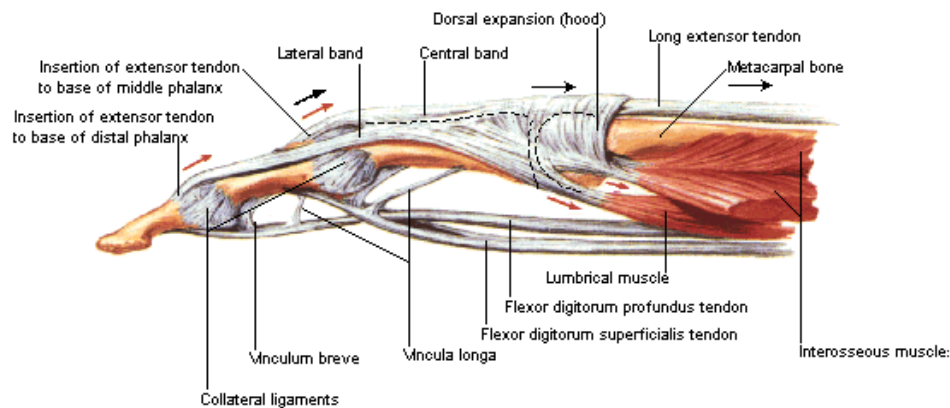


Figura 2.2: Estrutura de tendões no dedo humano, setas indicam caminho das forças de extensão.

A chave para atuar uma mão humana encontra-se na eficiência e precisão com a qual é possível operar cada junta, e utilizando a base biológica pode-se fazer um sistema que mimica a disposição de músculos e tendões para aproveitar a geometria ótima.

2.1.3 Tradução para sistema robótico

Estabelecida a complexidade da mão humana é preciso compreender como traduzir estas complicações da forma mais simples possível de forma que atenda as necessidades do projeto. De acordo com [30] a mão humana possui 27 graus de liberdade (DoF, do inglês *Degrees of Freedom*), sendo 21 nos dedos (5 no polegar e 4 nos outros) e 6 na palma. É comum ignorar os movimentos da palma em sistemas robóticos e utilizar uma palma fixa no lugar. Os dedos robóticos normalmente perdem um eixo de rotação na primeira junta e ficam com três graus de liberdade num plano, ficando similar a um modelo de braço robótico RRR (Rotativo-Rotativo-Rotativo) 2D. Esta construção nos permite controle de posição, suficiente para manipulação básica do ambiente.

2.2 Atuadores Tradicionais

Esta seção revisa atuadores normalmente utilizados em aplicações robóticas.

2.2.1 Pneumática

Sistemas pneumáticos reside seu princípio de atuação em movimentar gás pressurizado de forma a criar movimento, normalmente enchendo-se a câmara de um cilindro com ar. Velocidades de $1m/s$ são comuns e sistemas de alta velocidade conseguem atingir até $15m/s$ [2]. Dentre os diversos tipos de cilindros existentes na pneumática moderna, os mais comuns são os de simples-ação ou dupla-ação. Nos de simples ação um mecanismo mecânico (normalmente uma mola) retorna o pistão para sua posição inicial, enquanto que nos de dupla-ação o ar pode ser direcionado a qualquer uma das duas pontas do cilindro de forma a causar movimento na direção desejada. Embora muito utilizados na indústria, sistemas pneumáticos precisam ser acompanhados de um reservatório, filtro, compressor e válvulas, tornando o sistema pouco viável para aplicações móveis ou pequenas tal como uma mão robótica.

Dentro da pneumática existem um grupo especial de atuadores chamados Músculos Pneumáticos Artificiais (normalmente chamados de PAMs, do inglês *Pneumatic Artificial Muscles*). Pesquisas com estes músculos já foram feitas na PUC-Rio [31] com muito sucesso. A figura 2.3 exemplifica um PAM e seus estados.



Figura 2.3: Estados de um PAM; relaxado, parcialmente contraído e completamente contraído, da esquerda para a direita.

Músculos Pneumáticos Artificiais evoluíram muito na última década, porém o controle preciso ainda é não-trivial e continua exigindo um sistema pouco portátil.

2.2.2 Hidráulica

A hidráulica é uma forma de atuação muito próxima da pneumática, trocando gases por fluídos. Um sistema desta natureza consiste, em sua forma mais simples, de uma bomba, um reservatório, válvulas e atuadores. A bomba gera pressão forçando líquido (tipicamente óleo) do reservatório para o sistema hidráulico, sendo capaz de exercer forças maiores das que as de um sistema pneumático. Historicamente estes sistemas são muito utilizados na indústria, porém, devido ao seu tamanho e dificuldade de controle preciso, não é um bom candidato a controlar sistemas robóticos tão pequenos quanto uma mão.

2.2.3 Servo Motores

Motores existem em diversas formas e todos podem ser utilizados em aplicações robóticas por possuírem boa eficiência e tamanho flexível, além de serem um dos métodos mais fáceis de se atuar um sistema. Dentro da família dos motores, um Servomotor é um mecanismo que, através de um sistema realimentado, apresenta movimento proporcional a um comando, normalmente controlando posição velocidade angular.

Um exemplo de servomotor são os servomotores RC, utilizados em sistemas rádio-controlados. Estes servomotores recebem um sinal modulado por largura de pulso (PWM, do inglês *Pulse Width Modulation*) e o traduz para uma posição angular equivalente. O sinal possui período de $20ms$ e sua razão cíclica tipicamente varia entre $\frac{1ms}{20ms}$ e $\frac{2ms}{20ms}$ [32], conforme a figura 2.4.

Este tipo de atuador é muito popular devido ao seu fácil controle, porém por serem comerciais pode não existir algum com tamanho e torque suficiente para certas aplicações. Por este motivo é muito comum em sistemas modernos a implementação de um servomotor por meio de um motor comum e um sensor de posição ou velocidade como um encoder ou resolver, deixando a cargo do desenvolvedor implementar o controle.

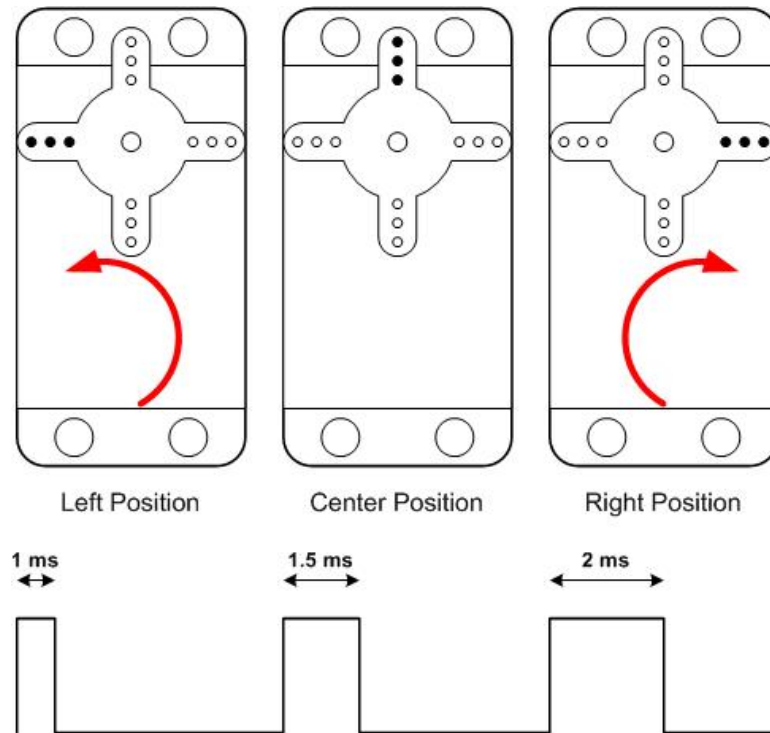


Figura 2.4: Posições e sinais de controle referentes de um servomotor RC.

2.2.4 Outros tipos comuns

Os atuadores apresentados acima representam a maioria das implementações em robótica, porém atuadores como motores de passo (ilustrado na figura 2.5) e solenoides (ilustrado na figura 2.6) são também encontrados em aplicações específicas que extraem as qualidades destes tipos.

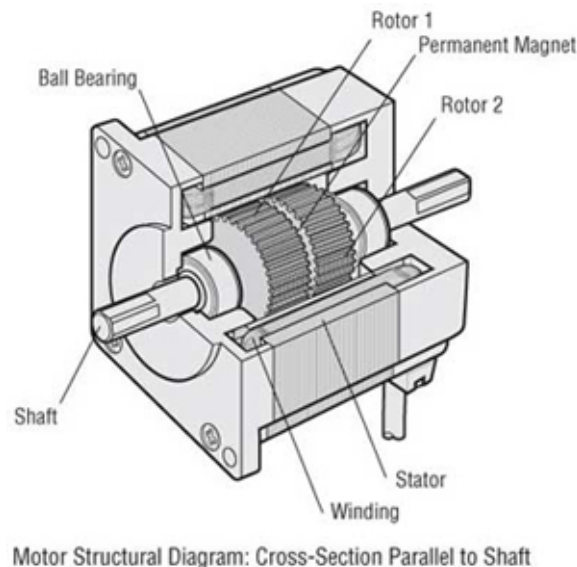


Figura 2.5: Motor de passo

2.3 Ligas de memória térmica de formato como atuadores

Nesta seção o conceito de memória térmica de formato é apresentado e seu princípio de funcionamento traduzido para aplicações como atuador. Estas ligas e outros materiais inte-

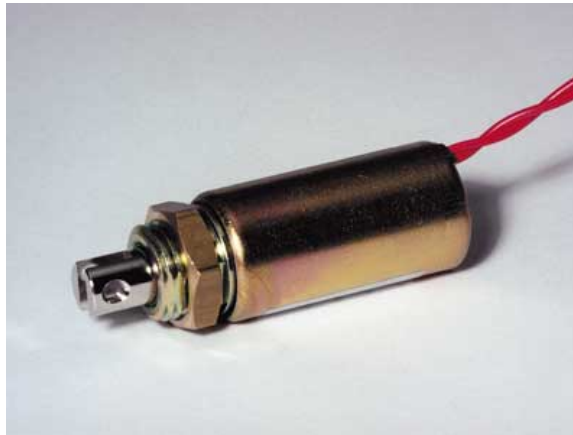


Figura 2.6: Solenoide

ligentes estão se sobressaindo nos últimos anos porém ainda não são amplamente utilizados por, normalmente, apresentarem pelo menos uma grande desvantagem. Nesta seção estas características são apresentadas. O foco será dado exclusivamente ao Nitinol, liga de memória de formato mais comum e a tratada neste trabalho, embora os conceitos são completamente generalizáveis a outras ligas.

2.3.1 Nitinol (NiTi)

Ligas de memória de formato (SMA do inglês *Shape Memory Alloys*) são um grupo de ligas que apresentam o efeito de memória de formato (SME) [25]. O efeito de memória de formato é uma transformação entre duas fases dependente da temperatura da liga. Estas duas fases são a martensita e austenita, ambas ilustradas na figura 2.7. A martensita, também conhecida como fase fria do material, ocorre normalmente a temperatura ambiente. Contrário ao pensamento intuitivo esta não é a fase original do material, e sim sua fase deformada. Isto é devido a estrutura cristalina do material que, neste caso, encontra-se esticada (figura 2.7c). Transferindo-se calor para o material a transformação de martensita para austenita se inicia na temperatura A_{start} . Aquecimento subsequente leva ao alinhamento da estrutura cristalina levando o material para a fase de austenita (figura 2.7a) na temperatura final A_{finish} . Quando o resfriamento é iniciado a transformação reversa começa na temperatura M_{start} e termina na temperatura M_{finish} , estando neste momento no estado geminado em relação a fase austenita (figura 2.7b). Este estado é chamado assim pois apenas resfriamento não retorna o material a sua forma martensita, tensão é necessária para deformar o material de volta a sua forma original (figura 2.7c). As curvas de deformação podem ser vistas na figura 2.8, onde também vemos o primeiro indício da forte histerese presente nestas ligas.

2.3.2 Flexinol

Flexinol é o nome comercial de um fio de NiTi produzido pela empresa Dynalloy. Estes fios são popularmente referenciados como músculos artificiais pela sua semelhança anatômica as fibras musculares reais, assim como sua semelhança com os PAMs.

A tabela 1 relaciona os diâmetros disponíveis e suas características eletro-mecânicas.

Flexinol está disponível em duas versões, uma com $A_{start} = 70^{\circ}\text{C}$ e outra com $A_{start} = 90^{\circ}\text{C}$. A primeira tem a clara vantagem de precisar de um aumento menor de temperatura para começar a exercer força, porém sua temperatura relativa ao ambiente não só deixa o sistema mais suscetível a flutuações de temperatura, como também diminui a taxa de resfriamento abaixo da temperatura M_{start} (por um fator de 1.3 a 3.0 de acordo com [35]), acarretando em uma resposta mais lenta.

De acordo com o fabricante [35] uma contração do comprimento total é esperada em aproximadamente 4.5%. A figura 2.9 exhibe as curvas de transformação para os dois tipos de Flexinol

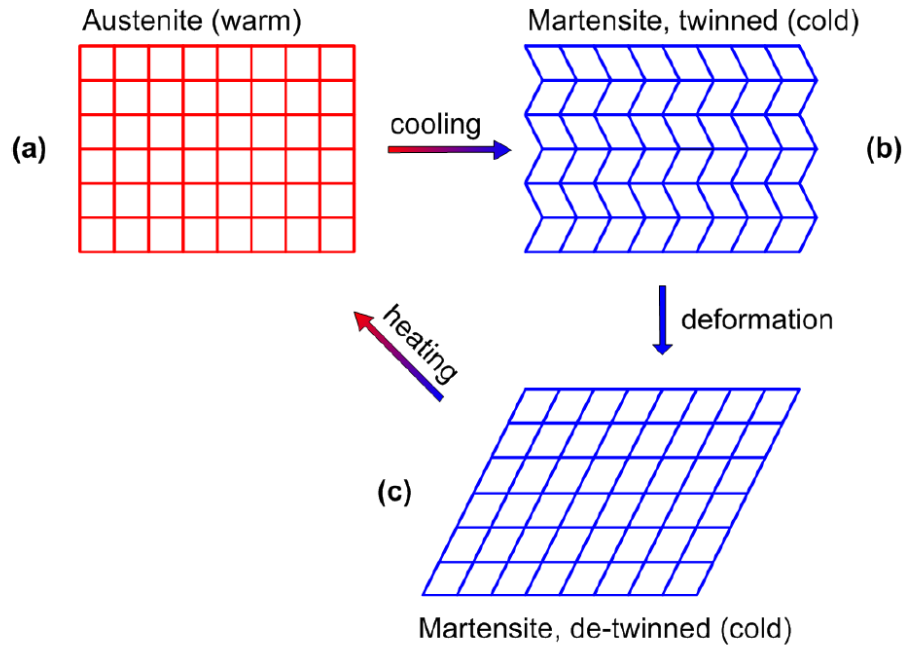


Figura 2.7: Fases de um material [6]

disponíveis. Vemos que a transformação para cada tipo começa antes da temperatura especificada, porém com intensidade baixa. Vemos também as temperaturas de saturação (A_{finish}) para cada tipo. A figura 2.9 exibe também dois problemas discutidos em trabalhos anteriores [2, 4, 6, 20, 21, 22, 23, 24, 25, 26, 27, 28], sendo estes a forte histerese a alta variação de comprimento em uma curta faixa de temperatura. A óbvia consequência decorrente é a dificuldade de controle, pois para controle preciso do comprimento de um fio de Flexinol seria necessário conhecer o ciclo na qual se encontra e modelagem e sensoriamento de alta performance para monitorar ou estimar a temperatura.

As temperaturas envolvidas permitem o uso do Flexinol como atuador em condições normais. Condições de temperatura ambiente muito alta ou muito baixa dificultam ou impedem completamente o resfriamento ou aquecimento do fio de Flexinol. Como a única variável envolvida na força exercida pelo Flexinol é a temperatura, qualquer método para levar o fio de A_{start} para A_{finish} pode ser utilizado, porém o mais comum é por meio da resistência característica do material e do efeito Joule, passando-se uma corrente pelo fio. A tabela 1 contém os dados necessários para este método. A coluna 1 lista os diâmetros disponíveis, como a resistência depende diretamente da seção reta do fio, a coluna 2 que contém as resistências de cada tamanho varia diretamente, também. O diâmetro também especifica a força máxima, exibida na coluna 3. A coluna 5 lista a corrente elétrica necessária para atingir A_{finish} em 1 segundo, aumentando ou diminuindo esta corrente pode-se variar o tempo de resposta, fator explorado mais adiante. A relação que governa o efeito é a da potência dissipada no fio, que neste caso é modelado como um resistor de resistência $R_{Flexinol}$, e a potência em Watts pode ser calculada com a equação fundamental (1):

$$P_{Flexinol} = R_{Flexinol} \cdot I_{Flexinol}^2 \quad (1)$$

A equação (1) pode ser reescrita utilizando-se a Lei de Ohm ($V = R \cdot I$):

$$P_{Flexinol} = V_S \cdot I_{Flexinol} \quad (2)$$

Estas equações permitem calcular todos os parâmetros necessários para atingir 1s de contração. Um estudo de caso rápido com o fio de diâmetro de 0.25mm e 1m de comprimento demonstra o processo:

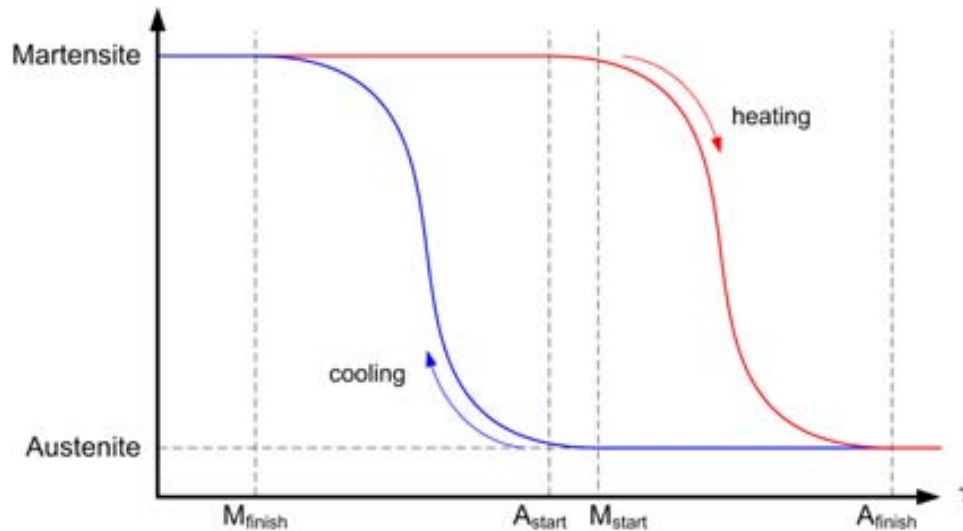


Figura 2.8: Curva de calor do Nitinol [33, 34, 35].

Diâmetro [mm]	Resistência [Ω/m]	Força [N]	Força de Retorno [N]	Corrente (1s de contração) [mA]	Tempo de resfriamento ($70^{\circ}C$) [s]	Tempo de resfriamento ($90^{\circ}C$) [s]
0.025	1425	0.089	0.036	45	0.18	0.15
0.038	890	0.20	0.08	55	0.24	0.20
0.050	500	0.36	0.14	85	0.4	0.3
0.076	232	0.80	0.32	150	0.8	0.7
0.10	126	1.43	0.57	200	1.1	0.9
0.13	75	2.23	0.89	320	1.6	1.4
0.15	55	3.21	1.28	410	2.0	1.7
0.20	29	5.70	2.28	660	3.2	2.7
0.25	18.5	8.91	3.56	1050	5.4	4.5
0.31	12.2	12.80	5.12	1500	8.1	6.8
0.38	8.3	20.04	8.02	2250	10.5	8.8
0.51	4.3	35.60	14.24	4000	16.8	14.0

Tabela 1: Características elétricas e forças máximas do Flexinol fornecidas pelo fabricante [34]

$$R_{Flexinol} = 18.5 \cdot 1 = 18.5\Omega$$

$$V_S = 18.5 \cdot 1050 \cdot 10^{-3} = 19.425V$$

$$P_{Flexinol} = V_S \cdot I_{Flexinol} = 19.425 \cdot 1050 \cdot 10^{-3} = 20.39W$$

Então para atingirmos contração completa em $1s$ é necessária a aplicação de $19.425V$ e o processo dissipará $20.39W$ de energia. Como notado em [35] podemos transferir muito mais energia ao sistema caso uma contração mais rápida seja necessária, porém uma corrente que faria o fio de Flexinol contrair em, por exemplo, $1ms$ irá aquecer o fio muito além de A_{finish} caso seja mantida por mais tempo que o necessário, podendo danificar o atuador.

A performance do Flexinol também depende diretamente da velocidade com a qual o mesmo esfriará. O primeiro passo para se considerar, de acordo com [35] é não sobre-aquecer o fio, ou seja, desligar o controle assim que o alongamento necessário seja atingido. O próximo é o meio e tipo de fio utilizado. A tabela 2 relaciona alguns métodos e suas eficácias relativas a utilizar o fio de menor temperatura ($A_{start} = 70^{\circ}C$) resfriado em ar estacionário na temperatura ambiente.

Materiais resistivos, tais como o Flexinol, aumentam sua resistência quando aquecidos. Enquanto que isto é verdade para a maioria dos materiais o mesmo não acontece com o Flexinol pelo motivo de que a forma do material muda em função da sua temperatura. A fórmula para a resistência de um material é $R = \frac{\rho l}{A}$, onde ρ é a resistividade característica do material, l é

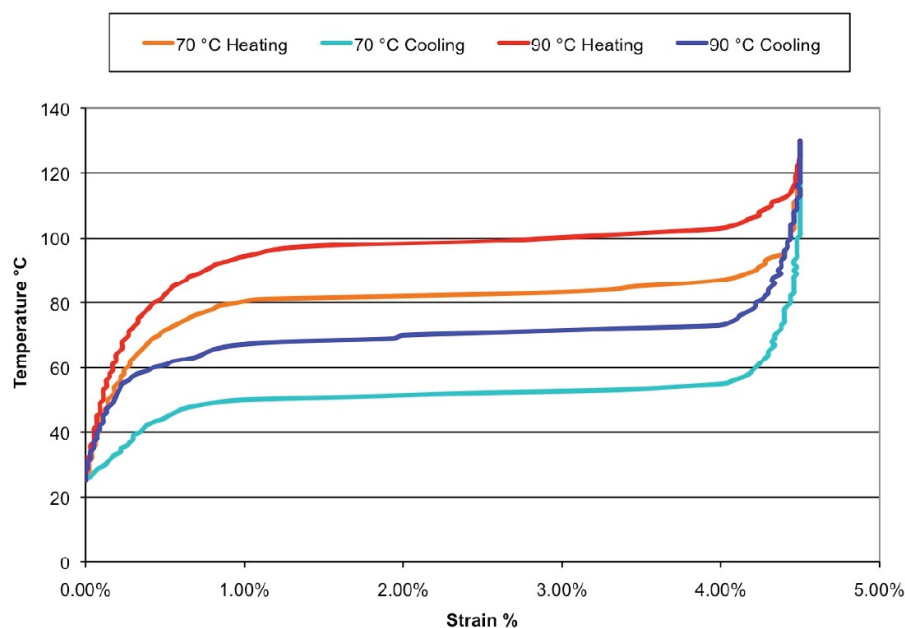


Figura 2.9: Curvas de transformação do Flexinol [35]. Histerese aparente

Método	Melhoramento na velocidade
Aumentar a tensão (mecânica)	1.2:1
Usar o fio de maior temperatura	2:1
Usar materiais sólidos como poço de calor	2:1
Convecção forçada	4:1
Pasta condutora térmica	10:1
Imersão em óleo	25:1
Água com Glicol	100:1

Tabela 2: Melhoramento relativo na velocidade relaxamento de diversos métodos [35].

o comprimento do material e A é a área da seção reta do material. Da fórmula fica claro que a resistência do material muda com a mudança de l e A , que no caso do Flexinol faz com que a resistência diminua com o aumento da temperatura, já que o volume do fio pode ser considerado é constante. Mesmo que ρ aumente com a temperatura, este aumento é insignificante frente as variações de l e A .

Analisando as colunas 6 e 7 da tabela 1 fica claro que o aumento do diâmetro leva a um aumento do tempo de resfriamento. Isto pode ser facilmente visto sabendo que o tempo de resfriamento por convecção depende da razão entre a área de superfície e o volume do fio. Como a área varia linearmente com o diâmetro ($A_s = \pi \cdot D \cdot l$) e o volume quadraticamente ($V = \pi \cdot (D/2)^2 \cdot l$) fica claro que o aumento do diâmetro acarretará em um aumento direto e não linear do tempo de resfriamento. Para mitigar este efeito a técnica mais comum para atingir grandes forças com tempos de resfriamento mais baixos é a paralelização de fios de Flexinol. Este método é discutido em [27], a desvantagem é o maior espaço necessário e a complexidade adicional de montagem.

O método mais comum de se regular a corrente desejada, proporcional a resistência da carga e a tensão aplicada, é utilizando o PWM (*Pulse Width Modulation*), que é uma onda quadrada de razão cíclica D variável, onde $D = t_{on}/t_{sinal}$. Neste tipo de sinal a tensão efetiva é o produto da razão cíclica com a tensão aplicada, que em processos suficientemente lentos para a frequência do PWM aplicado, resulta em uma regulação da tensão efetiva e, consequentemente, da corrente fluindo através da carga. No caso do Flexinol isso nos leva a regular diretamente a temperatura, já que por ser um processo lento funciona, essencialmente, como um filtro passa-baixas quando a tensão aplicada é rapidamente chaveada. A figura 2.10 exemplifica a saída de um filtro passa-baixas. Neste caso a curva em laranja representando a saída é proporcional a

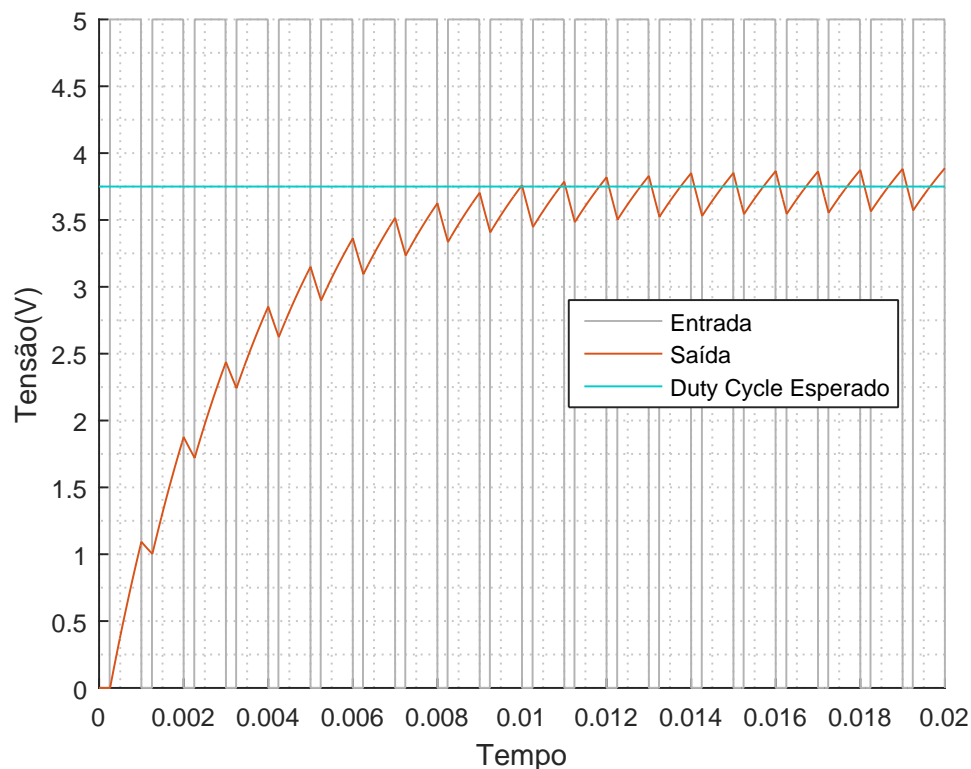


Figura 2.10: Evolução da saída de um filtro passa-baixas quando aplicado um PWM com $D = 75\%$

temperatura no fio de Flexinol.

2.4 Sensores

Dentro do escopo deste projeto apenas necessitamos medir deslocamentos, e não forças. Por este motivo apenas serão apresentados tipos de sensores que podem ser utilizados nesta aplicação. Para trabalhos futuros ou com necessidades diferentes, qualquer tipo de sensor que relacione o sinal de controle com algum efeito do Flexinol pode ser utilizado com alterações na lei de controle.

2.4.1 Transdutores Resistivos de Deslocamento (potenciômetros)

Estes transdutores, normalmente chamados de potenciômetros, relacionam um deslocamento angular com uma variação no valor de sua resistência, normalmente sendo traduzida para uma variação de tensão a ser medida por um conversor analógico-digital (ADC). A figura 2.11 exibe um transdutor deste tipo para movimentos angulares, enquanto que a figura 2.12 exibe um para movimentos lineares.

Embora o princípio de operação seja muito simples, estes dispositivos apresentam características desvantajosas para um sistema que requer controle fino como uma mão robótica, especialmente quando o atuador requer um controle trivial, como é o caso. Primeiramente por se tratar de uma resistência variável existe um forte atrito na haste de contato, parâmetro indesejável em qualquer tipo de controle. Segundo é a dificuldade de implementação em alguns casos, não só devido a um curso finito (um potenciômetro tipicamente não funciona em várias revoluções) como também quanto a necessidade de calibrar e modelar o acoplamento mecânico corretamente. Além destes problemas intrínsecos o tamanho e peso também os tornam indesejáveis em dispositivos muito pequenos, especialmente em uma mão mecânica onde estes provavelmente seriam instalados nas juntas dos dedos, onde o espaço é muito limitado.



Figura 2.11: Transdutor de movimentos angulares.

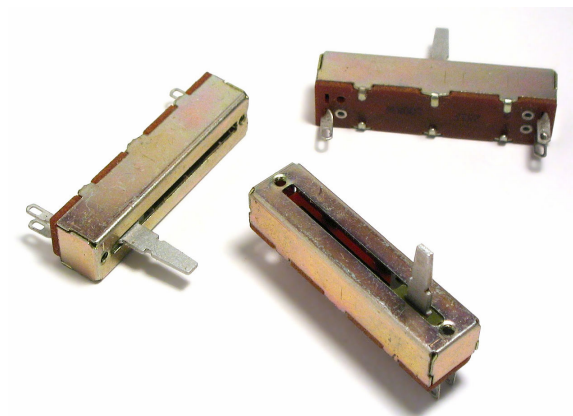


Figura 2.12: Transdutor de movimentos lineares, também conhecido como *fader*.

2.4.2 Transdutor de Esticamento

Estes transdutores utilizam o mesmo fenômeno que varia a resistência do Flexinol quando o mesmo estica ou contrai, porém utilizando um composto elástico e condutor. A figura 2.13 exibe um exemplo deste tipo de sensor. Embora este sensor apresente a grande vantagem de não necessitar de muito espaço, após rápidas observações foi decidido que não seria ótimo para a aplicação neste trabalho. Uma vez que é muito ruidoso, possui tempo de resposta lento especialmente durante a descontração, e o efeito elástico introduz forças externas indesejáveis para o controle.

2.4.3 Transdutor de Flexão

Este sensor tem seu princípio de funcionamento muito próximo a de uma *Strain Gauge*. Uma disposição em zigue-zague de um filamento condutor montado sobre um material flexível faz com que mudanças na angulação do dispositivo cause mudanças no l e A , mudando a resistência que neste caso aumenta com a flexão. Este sensor pode ser visto na figura 2.14 e um esquemático de seu funcionamento está disponível em [36] e exibido na figura 2.15.



Figura 2.13: Transdutor de esticamento, da empresa *adafruit*.

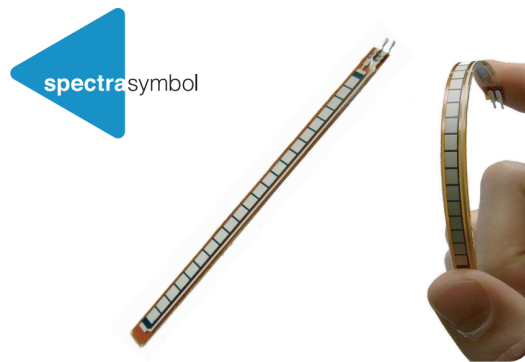


Figura 2.14: Transdutor de flexão, da empresa *Spectra Symbol*[36].

A tabela 3 foi construída tomando-se três medidas a cada 10° , começando em 0° (completamente esticado). O instrumento utilizado foi o multímetro *Fluke 87V Industrial Multimeter*. A empresa *Fluke* é altamente reconhecida no meio, portanto as medições são confiáveis. As configurações e aparência do instrumento podem ser vistas na figura 2.16. O transdutor foi ligado ao multímetro e fixo á mesa (figura 2.17) e em seguida foi utilizado um transferidor preciso de aço para forçar o transdutor a ficar na angulação desejada (figura 2.18).

O fabricante relata que o sensor é linear, porém por observação inicial da tabela foi percebido uma possível não linearidade. Para investigar a possibilidade de não linearidade foi criado um código em MATLAB (apêndice 9.1) que importa os dados da tabela e aproxima para diversos métodos de *fitting*. Após processamento de dados foi gerada o gráfico exibido na figura 2.19. O *Output* do código informou que a melhor interpolação foi a de exponencial de segunda ordem, vista na figura 2.20. A função de transferência final, relacionando a resistência $R_S[\Omega]$ com o ângulo $\theta_S[^\circ]$ é do sensor no intervalo de 0° a 110° com grau de confiança de 95% é:

$$R_S(\theta_S) = a \cdot e^{b \cdot \theta_S} + c \cdot e^{d \cdot \theta_S} \quad (3)$$

onde os coeficientes a , b , c e d são:

$$a = -2.999 \times 10^{-5}$$

$$b = 0.175$$

$$c = 2.429 \times 10^4$$

$$d = 0.0115$$

Medição	Ângulo [°]	Resistência [Ω]
1	0	25510
2	0	25310
3	0	25480
4	10	27100
5	10	27460
6	10	27200
7	20	29130
8	20	29900
9	20	29560
10	30	34330
11	30	33430
12	30	34180
13	40	39130
14	40	38710
15	40	37980
16	50	44160
17	50	43270
18	50	43190
19	60	47060
20	60	47740
21	60	47840
22	70	56300
23	70	54300
24	70	53500
25	80	61200
26	80	61800
27	80	61800
28	90	66600
29	90	67200
30	90	69400
31	100	74900
32	100	76400
33	100	75600
34	110	80900
35	110	79800
36	110	76900

Tabela 3: Amostras de medições com o Multímetro

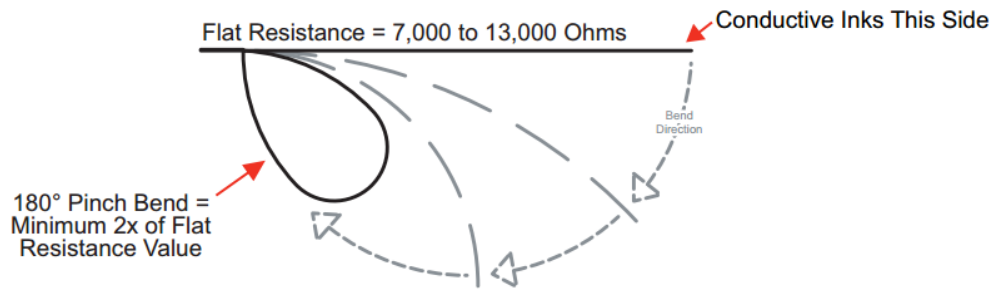


Figura 2.15: Esquemático de funcionamento do transdutor de flexão[36].



Figura 2.16: Multímetro Fluke 87V Industrial Multimeter configurado para medição de resistência em Ohms.

Feitas estas observações este foi o transdutor escolhido para esta aplicação. A não linearidade foi tratada e o transdutor apresenta diversas características vantajosas para implementação em uma mão robótica humanoide.

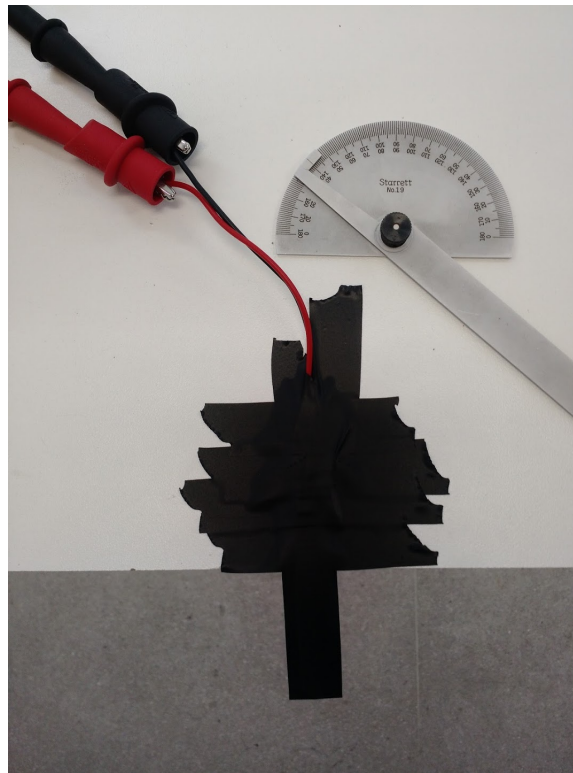


Figura 2.17: Transdutor de flexão ligado ao multímetro e fixo á mesa.

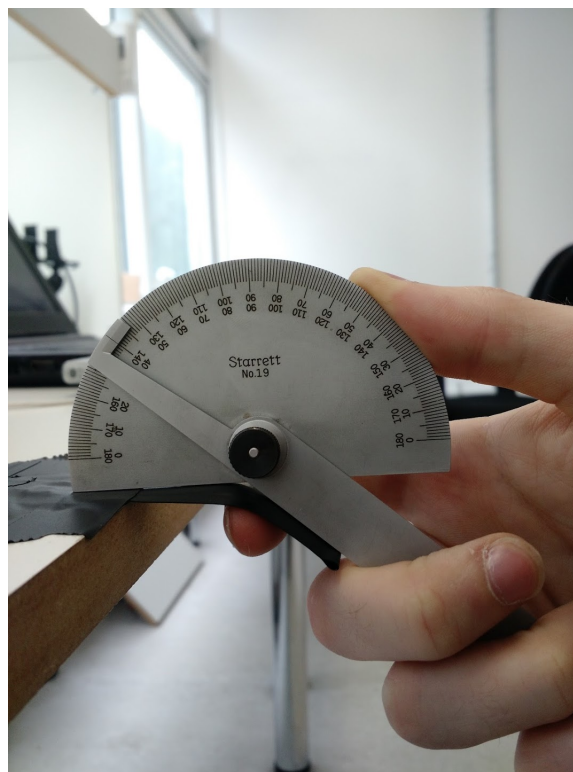


Figura 2.18: Transferidor de aço Starrett 19 garantindo angulação desejada.

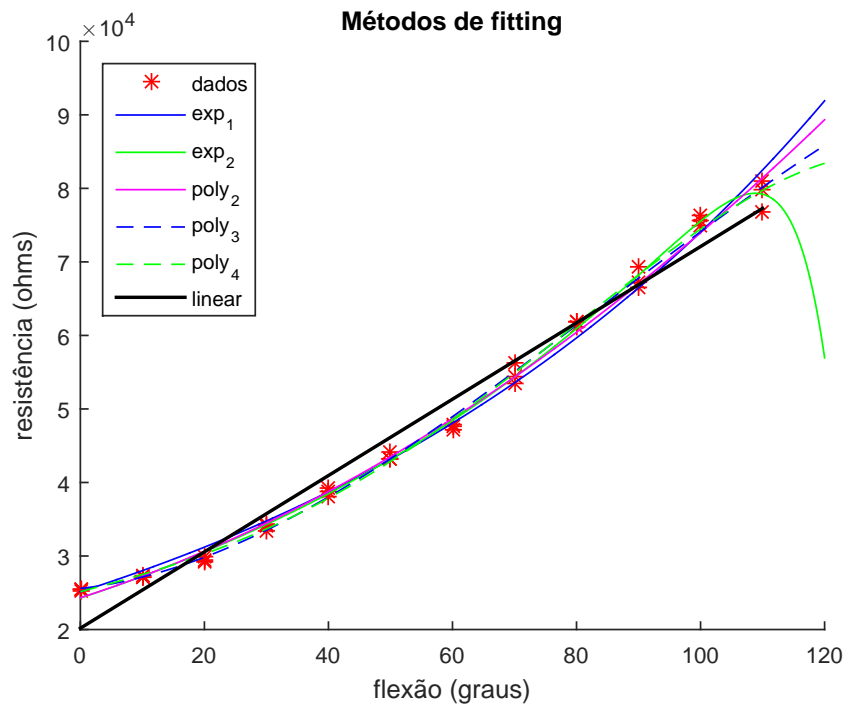


Figura 2.19: Dados e métodos de interpolação. Claramente o sensor exibe comportamento não linear.

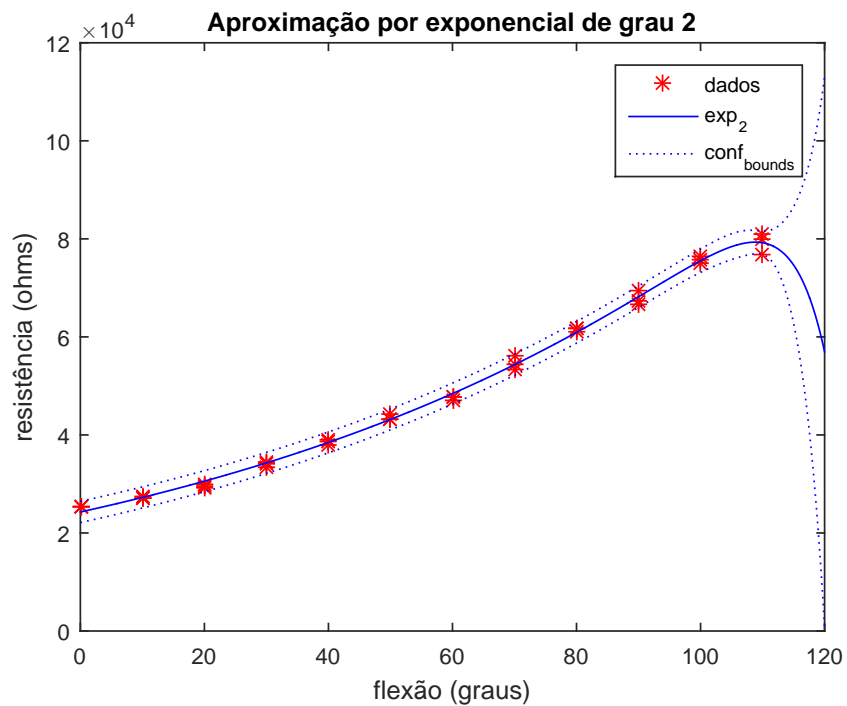


Figura 2.20: Resultado com nível de confiança para interpolação exponencial de segunda ordem.

3 Métodos e Ferramentas

Neste capítulo serão descritos métodos e ferramentas utilizadas na implementação dos conceitos apresentados no capítulo anterior. Primeiramente são apresentados o microprocessador PIC18F4431, reguladores de tensão LM2575-5.0 e LM3940, o MOSFET TPH4R008NH, conversor RS-232 para RS-485 MAX490CSA+ e porta lógica AND SN74LVC2G08DCTR. Depois são apresentadas as ferramentas de desenvolvimento CAD (PROTEUS e SolidWorks) e o compilador (CCS PCWHD).

3.1 Microcontrolador PIC18F4431

Uma mão robótica necessita tanto ler dados de seu estado como sinais de controle para atingir uma configuração desejada. Além disso, no caso de um sistema mestre-escravo, necessita-se a capacidade de comunicação com o sistema gêmeo. Estas necessidades podem ser todas atendidas utilizando-se uma MCU (*Master Control Unit*) capaz. Para este projeto a MCU escolhida foi o microcontrolador PIC18F4431 [38] da empresa Microchip.

A Microchip possui sua família de microcontroladores PIC muito difundida no mercado, e em diversas versões. A família PIC18 é a vertente de 8bits da Microchip, e a letra F indica os voltados para controle de diversos tipos, normalmente motores. Como motores são controlados por PWM assim como o Flexinol, é natural a escolha da família PIC18F para esta aplicação. Dentro desta existem diversos modelos, e após filtrar pelo site do fabricante foi escolhido o PIC18F4431, possuindo todas as características mínimas e, ainda, muitas funções úteis para as tarefas deste projeto.

De todas as características desta MCU, algumas são destacadas por sua importância:

- Até três timers operando em 8 ou 16bits.
- 4 canais complementares (total de 8 canais) de PWM de 14bits de resolução.
- 9 canais para o Conversor A/D, operando em até 10bits e até 200Ksps (*Kilo samples per second*).
- Alta velocidade: frequência de oscilação máxima de 40MHz, resultando em até 10Mips (*Mega instructions per second*).
- Alta corrente de sink/source, até 25mA.
- Três interrupções externas
- Dois módulos CCP (Capturar/Comparar/PWM), com resolução de PWM de até 10bits.
- Suporte nativo a RS-232 e RS-485 (a serem explicados) via módulo EUSART (*Enhanced Universal Synchronous Asynchronous Receiver Transmitter*).

A tabela 4 foi extraída de [38] e compara os dispositivos da família PIC18FXX31.

Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	CCP	SSP		EUSART	Quadrature Encoder	14-bit PWM (ch)	Timers 8/16-bit
	Flash (bytes)	# Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)				SPI	Slave I ² C™				
PIC18F2331	8192	4096	768	256	24	5	2	Y	Y	Y	Y	6	1/3
PIC18F2431	16384	8192	768	256	24	5	2	Y	Y	Y	Y	6	1/3
PIC18F4331	8192	4096	768	256	36	9	2	Y	Y	Y	Y	8	1/3
PIC18F4431	16384	8192	768	256	36	9	2	Y	Y	Y	Y	8	1/3

Tabela 4: Comparação da família PIC18FXX31 [38].

Como o projeto é um protótipo o encapsulamento escolhido para este componente foi o TQFP-44 (pacote TQFP de 44 pinos) e seu diagrama de *pinout* pode ser visto na figura 3.1. Este PIC neste encapsulamento contém 4 registradores de 8bits de uso geral, e um de 4bits, além de tensão de comparação de conversão analógica externamente customizável por meio dos pinos AV_{SS} e AV_{DD}. Este encapsulamento também apresenta quatro pinos não conectados (NC). Das

notas fornecidas por [38] na figura 3.1 a mais importante é a 4, informando que a saída PWM4 (quinto sinal de PWM) está presente também em um pino alternativo.

3.1.1 Portas I/O

Como dito anteriormente este microprocessador possui diversas portas com usos e funções diferentes. Os pinos das portas A, B, C e D todos podem servir como entrada ou saída lógica, com exceção da porta E3, que somente funciona como entrada fixa durante a execução do programa. Abaixo as funções importantes e utilizadas neste projeto são explicadas:

RX Receptor serial do módulo USART (*Universal Synchronous Asynchronous Receiver Transmitter*).

TX Transmissor serial do módulo USART.

PWMX Saída X de PWM. As saídas são agrupadas dois a dois a partir da saída 0, podendo operar somente uma das duas, ambas iguais, ou sinais complementares.

CCPX Porta X do módulo CCP (Capturar/Comparar/PWM). Estas portas podem ser configuradas para capturar ou comparar um sinal, mas neste projeto serão utilizadas na sua terceira função, que gera cada um sinal independente de PWM.

INTX Interrupção externa X. Causa interrupção da pilha de execução para uma função específica programável. Uso desta função será explicado e explorado no capítulo 4.

ANX Entrada analógica X. Neste dispositivo existem 9 canais analógicos multiplexados ao módulo do ADC. Cada porta pode ser ligada a um sinal analógico diferente de forma a ser convertido quando necessário.

AV_{DD} Tensão positiva de comparação analógica. Esta é a tensão que implicará no valor máximo do conversor quando comparada a uma tensão analógica a ser convertida. Exemplo, se $AV_{DD} = 3.3V$ então a leitura do ADC de um sinal analógico de $3.3V$ se configurado em 8bits será **0xFF**, ou 255.

AV_{SS} Tensão negativa de comparação analógica. Esta é a tensão que implicará no valor mínimo do conversor quando comparada a uma tensão analógica a ser convertida. Exemplo, se $AV_{SS} = 0V$ então a leitura do ADC de um sinal analógico de $0V$ se configurado em 8bits será **0x00**, ou 0.

PGC/PGD Portas de clock e dados de programação, respectivamente. São conectadas ao programador para upload da *firmware* do dispositivo, de forma a carregar a pilha de execução (ou programa) desejado. Estas portas são utilizadas para programação do tipo ICSP (*In-circuit serial programming*), que significa programar o dispositivo no sistema em que está instalado, no caso uma PCB (*Printed Circuit Board*, Placa de Circuito Impresso).

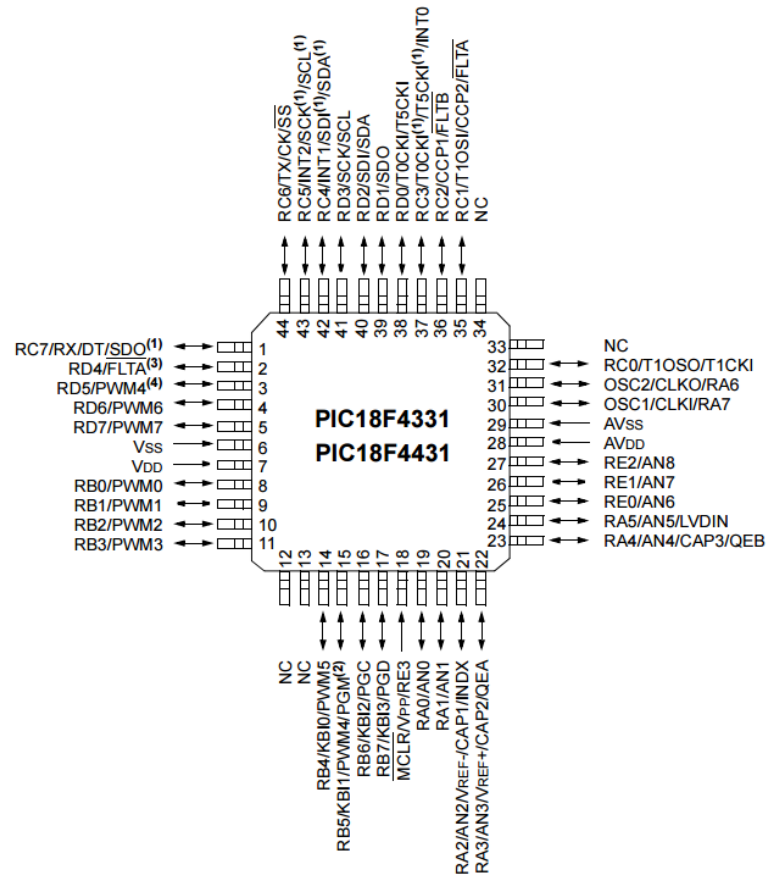
OSC1/2 Entradas do oscilador externo. Para velocidades $> 4MHz$ é necessário um oscilador externo, a ser conectado nestas duas portas.

3.1.2 Interrupções

Como dito anteriormente esta MCU apresenta capacidade de interrupções. Interrupções são eventos que removem a necessidade de constantemente aquisitar os estados do sistema, ao informar que certa mudança de estado precisa ser tratada imediatamente. Estas mudanças podem ser provenientes de hardware quanto de software, sendo dois respectivos exemplos apertar um botão e passar um determinado período. Interrupções param a execução do programa e um comando de *jump* aponta a execução para outra pilha de execução, que é a função a ser executada quando a interrupção ocorre. Após o tratamento da interrupção o programa resume de onde parou.

No *Datasheet* do microcontrolador [38] todas as informações sobre interrupções e seus tipos encontram-se no capítulo 9, embora todas as configurações ali informadas podem ser operadas em um nível mais alto de programação no ambiente do compilador.

44-Pin TQFP



- Note 1:** RC3 is the alternate pin for T0CKI/T5CKI; RC4 is the alternate pin for SDI/SDA; RC5 is the alternate pin for SCK/SCL.
- 2:** Low-voltage programming must be enabled.
- 3:** RD4 is the alternate pin for FLTA.
- 4:** RD5 is the alternate pin for PWM4.

Figura 3.1: Diagrama de *pinout* do PIC18F4431 no pacote TQFP-44 [38].

3.1.3 Timers e contadores

Neste microcontrolador estão disponíveis três timers: **TIMER0**, **TIMER1** e **TIMER2**. O **TIMER5** também é um timer, porém é voltado para propósitos não tratados neste projeto. Os três timers principais operam em 8 ou 16bits, dependendo da configuração ou restrição do timer.

Timers funcionam dividindo o clock global por um valor fixo e ajustável, podendo dividir o clock por $2^n, n = 0...8$, este valor é conhecido como *Prescaler*, ou *PRSCl*. Após esta divisão o timer incrementa (ou conta, por isso pode ser considerado um contador) até o valor máximo possível ou até um valor ajustado pelo programador, gera uma interrupção caso assim esteja configurado, e recomeça a contagem. O clock natural é dividido pelo PIC por 4 antes de virar o clock global, portanto o tempo em segundos até ocorrer um *overflow* (o timer chegou em seu valor máximo e retornou a zero) depende da frequência de oscilação (F_{OSC}), do *Prescaler* (*PRSCl*) e do número de bits (b) máximo e é:

$$\frac{4}{F_{OSC}} \cdot PRSCl \cdot 2^b \quad (4)$$

Para utilizarmos a equação (4) podemos ou contar a quantidade de vezes que foi gerada uma interrupção por *overflow* via contador externo, ou pré-carregar o contador com um valor conhecido, de forma que o tempo até o *overflow* seja menor porém mais ajustável. Podemos exemplificar gerando uma interrupção a cada exatos $1ms$ a $20MHz$ e com $PRSCl = 8$ operando em 16bits. Começamos achando o tempo de *overflow*:

$$t_{ovf} = \frac{4}{20 \cdot 10^6} \cdot 2 \cdot 2^{16} = 26.21ms$$

Queremos achar qual a pré-carga necessária para que o *overflow* ocorra a cada $1ms$, portanto devemos resolver a seguinte equação para a , onde a é a pré-carga:

$$\frac{4}{20 \cdot 10^6} \cdot 2 \cdot (2^{16} - a) = 1ms$$

Resolvendo a equação achamos $a = 63036$, como o valor é inteiro a interrupção ocorrerá a cada exatos $1ms$. Esta manipulação é vantajosa se, por exemplo, queremos regular precisamente o período de controle.

3.1.4 CCPX PWM

Das fontes de PWM disponíveis, a proveniente do módulo CCP é a mais fácil de configurar. Sua frequência é igual à frequência de *overflow* do **TIMER2**, e sua razão cíclica é ajustável e proporcional ao inverso da contagem máxima que o **TIMER2** pode atingir considerando a pré-carga. Se, por exemplo, **TIMER2** está configurado para contar até 255 (8bits), então uma razão cíclica de 50% equivale a dizer ao módulo CCP para trocar de estado quando chegar a 127 ($127/255 \approx 0.5$). A configuração destes sinais PWM será feita por software no capítulo 5.

3.1.5 Power PWM - PWMX

Estas saídas PWM possuem uma disposição mais complexa. Começando da saída PWM0 e agrupadas duas a duas, cada par (ou grupo) pode operar com as duas ligadas com o mesmo sinal, as duas desligadas, ou uma com o sinal complementar da outra. Isto é problemático se queremos atuar em cada saída separadamente. Felizmente a arquitetura do PIC18F permite este tipo de controle seletivo, por meio dos bytes de configuração **OVDCOND** e **OVDCONS**. A configuração destes bytes a suas explicações serão vistas no capítulo 5.

Quando configurados corretamente **OVDCOND** e **OVDCONS** habilitam ao programador o controle independente de dos canais **PWMX**, conforme a tabela de configuração 5 e o referente resultado na figura 3.2. Estas configurações serão explicitadas adiante, porém para

State	OVDCOND (POVD)	OVDCONS (POUT)
1	11000011b	00000000b
2	11110000b	00000000b
3	00111100b	00000000b
4	00001111b	00000000b

Tabela 5: Quatro configurações dos bytes **OVDCOND** e **OVDCONS** [38].

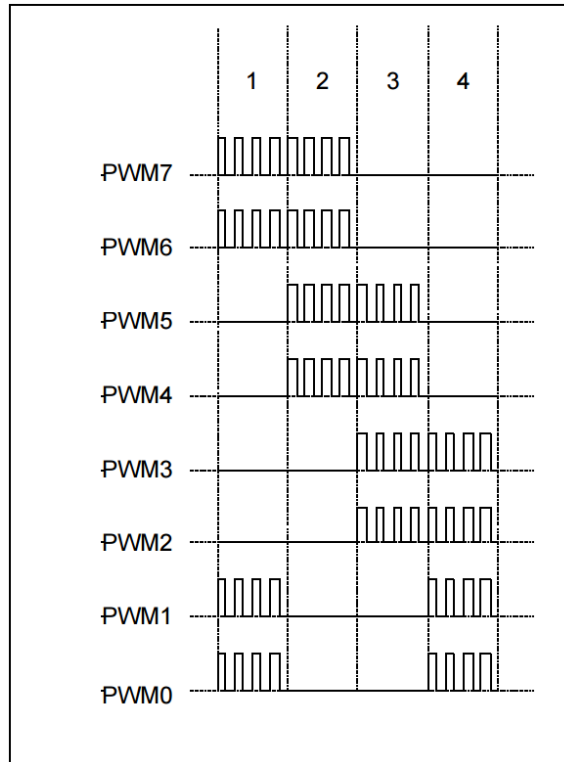


Figura 3.2: Todas as saídas nos diferentes estados vistos na tabela 5 [38].

entendimento das próximas seções e sub-seções é necessária a informação de que o controle independente é possível.

3.1.6 Enhanced Universal Synchronous and Asynchronous Serial Receiver and Transmitter - EUSART

Para comunicação o PIC18F4431 apresenta um USART, neste caso chamado de EUSART pelas melhorias que a Microchip implementou. Em sua forma mais básica, e a utilizada no software de controle, o módulo funciona de forma assíncrona e envia e recebe bits por meio dos pinos **TX** e **RX**, respectivamente. Nesta configuração o módulo USART funciona em modo duplex completo (consegue enviar e receber dados ao mesmo tempo) e suporta número ajustável de bits, *baud rate* (*bits/s*) e bits de parada, assim como paridade selecionável.

Dentre as funções disponíveis no PIC18F4431 uma é a geração de interrupções sob chegada de dados no módulo USART, o que é muito útil quando se quer tratar comandos assim que chegarem otimizando, desta forma, o tempo de resposta do sistema. Outra função importante é a emulação de outros canais seriais em pinos não ligados diretamente ao módulo USART permitindo comunicação independente com mais de um alvo, evitando desta forma a criação de uma rede de comunicação.

3.1.7 Conversor analógico-digital (ADC)

Como especificado anteriormente esta MCU apresenta até 9 canais configuráveis de conversão A/D, operando em até 10bits e até 200Ksps (*Kilo samples per second*, até $5\mu s$ por aquisição). Outras funções deste módulo incluem gatilho de aquisição (aquisitar quando algum evento programado acontecer, tal como a mudança de estado de um pino). Em alguns compiladores é necessário acesso aos 9 registradores do módulo para operá-lo, porém aqui isto não será necessário uma vez que o compilador utilizado implementa todas as operações nativamente.

3.2 Texas Instruments LM2575SX-5.0

Como visto anteriormente, os fios de Flexinol necessitam de tensões maiores do que máxima aceita pela MCU e outros componentes lógicos da placa. Para alimentar a placa corretamente existem duas soluções: utilizar duas fontes com tensões diferentes e um terra comum, ou regular uma tensão maior para 5V. Embora a segunda aumente o tamanho da placa e gere um sistema ligeiramente mais custoso, os benefícios em praticidade, especialmente a longo prazo, mais que compensam. Existem dois caminhos para se regular uma tensão: reguladores lineares e chaveados. Como o nome sugere reguladores lineares variam sua resistência interna de forma manter uma saída constante. Embora sejam a forma mais simples de se regular uma tensão, são muito ineficientes. Suponha que queiramos reduzir uma tensão de 12V para 5V e o sistema irá puxar apenas 500mA, neste caso a potência dissipada será:

$$P_{reg} = V_{drop} \cdot I_{reg} = (12 - 5) \cdot 0.5 = 3.5W$$

Embora 3.5W possa parecer pouco, irá gerar calor excessivo e desnecessário, além de deixar todo o sistema com uma margem de segurança menor. Nestes casos um regulador chaveado (*Switching Regulator*) é recomendado. Estes reguladores utilizam o princípio da indutância e um sinal de PWM internamente regulado para gerar um sinal DC fixo, basicamente filtrando um PWM de alta tensão com um passa-baixas inteligente, melhorando enormemente a eficiência do sistema. O regulador mais famoso e popular destes é a família LM2575, fornecendo até 1A e oferecendo proteções internas contra sobre-aquecimento e sobre-corrente. Um esquemático simples de aplicação típica encontra-se na figura 3.3. Os capacitores, indutor e diodo presentes são necessários para estabilização da linha de alimentação assim como para filtrar a saída e fornecer uma fonte limpa de ruídos. Este regulador pode ser encontrado em diversas versões de saída não ajustável, claramente a versão utilizada será a que possui 5V de saída, o LM2575SX-5.0.

Typical Application

(Fixed Output Voltage Versions)

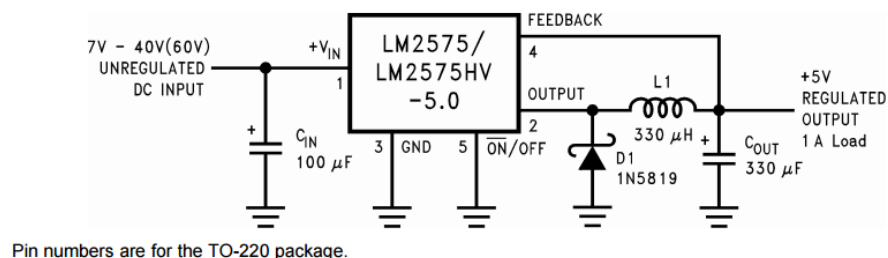


Figura 3.3: Aplicação típica do LM2575 de saída fixa (não ajustável) [41].

3.3 Texas Instruments LM3940

O método padrão para se medir um sensor de resistência variável, tal como o utilizado, é por meio de um divisor de tensão (figura 3.4) cuja a função de transferência é (5), onde duas resistências em série criam uma tensão intermediária a ser medida pelo ADC. Na figura 3.4 R_1

equivale ao sensor e R_2 ao resistor de divisão (ou resistência divisora) e V_{out} é o sinal analógico a ser lido pela MCU. Este sensor, por apresentar resistências relativamente altas (até $81K\Omega$ de acordo com a tabela 3) a sensibilidade tende a baixa. Para verificar esta hipótese um simples código foi desenvolvido em MATLAB (apêndice 9.2) para encontrara a resistência ótima para a divisão de tensão, a sensibilidade máxima, resolução do DAC e tensões máxima e mínima possíveis para a configuração.

$$V_{out} = V_{in} \frac{R_2}{R_1 + R_2} \quad (5)$$

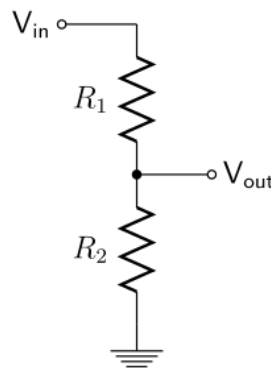


Figura 3.4: Divisor de tensão padrão.

O código resultou em uma resistência divisora ótima $R_{div} = 45K\Omega$, uma tensão máxima de $3.21V$, uma tensão mínima de $1.78V$ e uma resolução de $19.53mV/bit$, além da figura 3.5 exibindo a curva de sensibilidade, claramente indicando um máximo local.

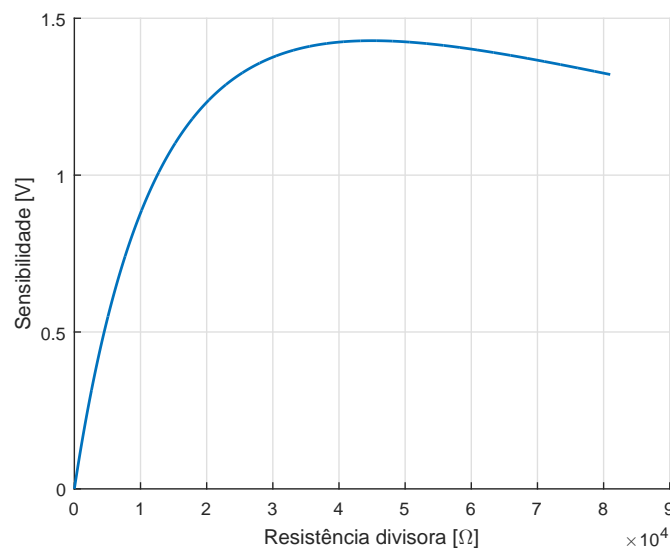


Figura 3.5: Curva de sensibilidade ao variar-se R_1 .

Esta análise fez possível a descoberta de um método simples para aumentar a resolução máxima: como o sensor não passará de $3.3V$, podemos utilizar um regulador para fornecer uma fonte estável de referência para o ADC, efetivamente aumentando a resolução para $12.89mV/bit$. O motivo da escolha desta tensão é não só pela proximidade do máximo teórico, como também por ser uma tensão padrão reguladores de $5V$ para $3.3V$ estão facilmente disponíveis no mercado. Dentre os existentes, foi escolhido o Texas Instruments LM3940 [42], pela simplicidade oferecida. Embora seja um regulador linear, as correntes puxadas pelo ADC e a queda de tensão são consideravelmente baixas, não acarretando em nenhum dos problemas decorrentes do uso destes reguladores. A utilização de um divisor de tensão para esta aplicação é perigosa pois a porta AV_{DD} pode puxar corrente o suficiente para gerar uma tensão indesejável no divisor

de tensão, efetivamente afetando negativamente a medição. A figura 3.6 exibe o esquemático típico de implementação do LM3940.

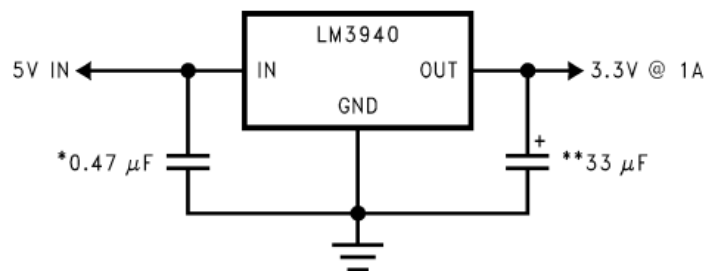


Figura 3.6: Aplicação típica do LM3940 [42].

3.4 MOSFET Toshiba TPH4R008NH

Como queremos chavear o Flexinol a uma tensão superior a 5V e com correntes muito superiores a 25mA será necessário o uso de um MOSFET (*metal-oxide-semiconductor field-effect transistor*). O princípio básico de funcionamento de um MOSFET é que, ao aplicar-se uma diferença de potencial positiva (no caso de um MOSFET de canal N) entre o *gate* e a fonte (*source*) do MOSFET, denominada de V_{GS} , o dispositivo permite a passagem de corrente em etapas diferentes, sendo elas a de corte, triodo e de saturação. Em aplicações de alta tensão e corrente é necessária uma análise extensiva sobre todos os parâmetros da carga para se escolher um MOSFET que funcione eficientemente. Neste trabalho, no entanto, as correntes são relativamente baixas para um MOSFET, portanto podemos utilizar um dispositivo menos potente, mais rápido e mais fácil de se operar.

Quando procura-se um dispositivo assim, queremos minimizar a carga de *gate* Q_{SW} e a resistência dreno-fonte ligada $R_{DS(ON)}$. Minimizar Q_{SW} traz o benefício de atingir a saturação mais rápido enquanto diminuindo a corrente necessária para esta velocidade, o que será vantajoso pois iremos alimentar o *gate* deste MOSFET diretamente da MCU, que consegue fornecer somente 25mA. Já uma baixa $R_{DS(ON)}$ diminui a potência dissipada (e consequentemente o calor gerado). Após uma pesquisa dos MOSFETS disponíveis localmente na PUC-Rio foi decidido o uso do Toshiba TPH4R008NH, que apresenta $R_{DS(ON)} = 3.3m\Omega$ e $Q_{SW} = 18nC$.

Exemplificando com uma carga típica de 1A podemos calcular a potência dissipada em regime permanente neste MOSFET, que neste caso será praticamente desprezível:

$$P_{FET} = R_{DS(ON)} \cdot I^2 = 3.3mW$$

Como não queremos que a corrente saindo do PIC seja maior do que 25mA e considerando uma margem de segurança, um resistor de *gate* de 220Ω garantirá esta regulagem. O circuito de chaveamento do *gate* torna-se, então, um simples circuito RC. Para achar a frequência máxima de chaveamento foi feito um pequeno código em MATLAB (apêndice 9.3) que resultou em um período de subida (*rise time*) de 5.8µs. Como temos um ciclo de carga e um de descarga em cada período do PWM, para achar a frequência máxima multiplicamos o período de assentamento por dois e invertemos, conseguindo uma frequência máxima de 86.20KHz. Para esta aplicação tal frequência de chaveamento é desnecessariamente alta, porém é importante garantir que o sistema estará numa faixa segura de operação. A figura 3.7 exibe a curva de carga do *gate*. Devido à necessidade de uma tensão positiva no *gate*, o circuito básico de ativação do MOSFET é dado pela figura 3.8, onde a trilha PWM liga diretamente na MCU.

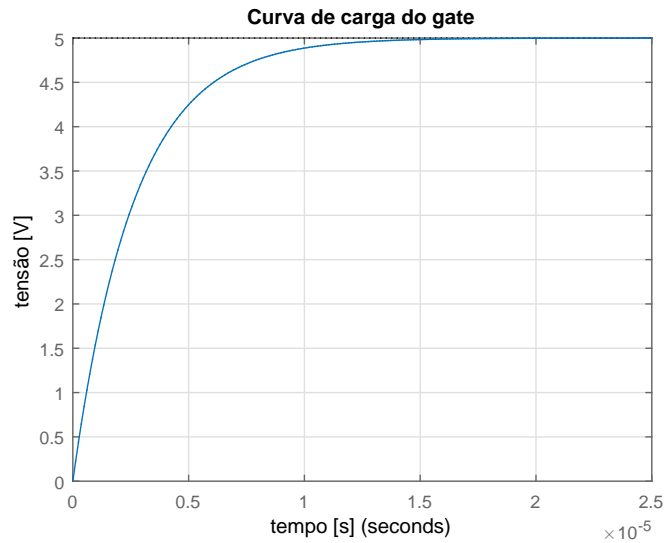


Figura 3.7: Curva de carga do gate do MOSFET Toshiba TPH4R008NH com $V_{DS} = V_{GS} = 5V$.

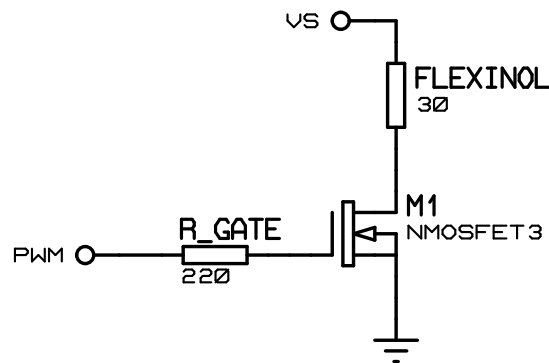


Figura 3.8: Circuito de ativação de um MOSFET de canal N.

3.5 Maxim MAX490CSA+ RS-485 Transceiver

A comunicação serial a ser utilizada neste trabalho será assíncrona e *full-duplex*. A saída do módulo serial do PIC18F4431 é do tipo RS-232 de baixa tensão (nível lógico), onde existe uma via de transmissão, uma de recepção e um terra de referência. Este protocolo de transmissão é simples e útil para pequenas distâncias (até 5 metros), porém para sistemas remotos é essencial poder operar a longas distâncias. O protocolo que atende este requisito é o RS-485, também conhecido como TIA-485-A.

O protocolo RS-485 utiliza duas vias de transmissão e duas de recepção, e o sinal de dados é o diferencial entre as tensões de cada condutor. A figura 3.9 exibe o diagrama funcional do MAX490, que opera sob o protocolo RS-485. O sinal de dados é a diferença de potencial entre os condutores Y e Z ou A e B, dependendo se o sinal é de transmissão ou recebimento. O resistor R_t é chamado de resistor de terminação e sua função é dissipar ecos eletromagnéticos na linha, melhorando o alcance e velocidade máxima de transmissão. O protocolo RS-485 quando bem implementado pode chegar a distâncias de até 1200 metros, possibilitando o operador a ficar a uma distância segura do robô.

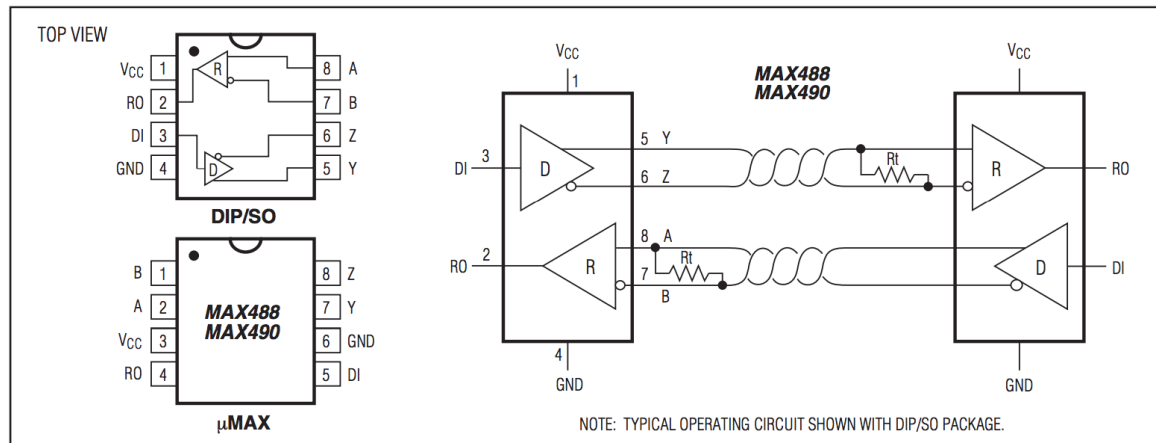


Figura 3.9: Diagrama funcional do MAX490 [40].

3.6 Texas Instruments SN74LVC2G08DCTR

O sistema mecânico idealizado (a ser apresentado no capítulo 4) necessitará de 12 canais de PWM. Das portas **PWMX** pode-se ter um total de 8 canais, e das portas CCPX mais 2, totalizando 10 canais possíveis. Para resolver os dois canais faltantes foi feita uma multiplexação das portas CCPX por meio de um seletor simples baseado em portas lógicas do tipo AND colocando-se em uma das entradas da porta o PWM e na outra entrada o sinal seletor, conforme a figura 3.10.

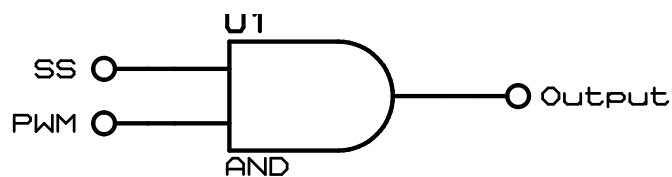


Figura 3.10: Multiplexação do sinal PWM proveniente das portas CCPX.

Como a maioria das portas lógicas servem apenas para processar sinais de baixíssima corrente, foi necessária a utilização de um dispositivo capaz de fornecer ao menos $25mA$ para se equiparar ao PIC18F4431. Após pesquisa dos componentes disponíveis foi decidido utilizar o SN74LVC2G08DCTR pela velocidade, capacidade de fornecer até $50mA$ contínuos e o fato do encapsulamento vir com duas portas, diminuindo o gasto de espaço na placa já que serão necessárias quatro portas no total.

3.7 Ferramentas de CAD

Nesta seção serão apresentadas as duas ferramentas utilizadas para tarefas de desenho e fabricação, PROTEUS e SolidWorks.

3.7.1 PROTEUS Suite

O software PROTEUS da empresa *Labcenter Electronics* é composto de duas partes, uma é o ISIS que é a de captura de esquemáticos, o outro é o ARES que é a de construção de layout de PCBs.

O ISIS é onde o projetista monta esquematicamente as ligações e dispositivos da placa. A figura 3.11 exibe o aspecto do programa, assim como uma folha do projeto deste trabalho aberta mostrando diversos componentes e ligações. Neste programa o projetista tem a liberdade de não se preocupar com um sistema físico ainda, podendo focar em deixar a lógica correta e simular os programas da MCU para confirmar dentro do possível que tudo irá funcionar.

Após captura do esquemático, o projetista passa para o projeto físico da placa, por meio do programa ARES, cuja janela principal com o projeto deste trabalho está exibida 3.12. No ARES

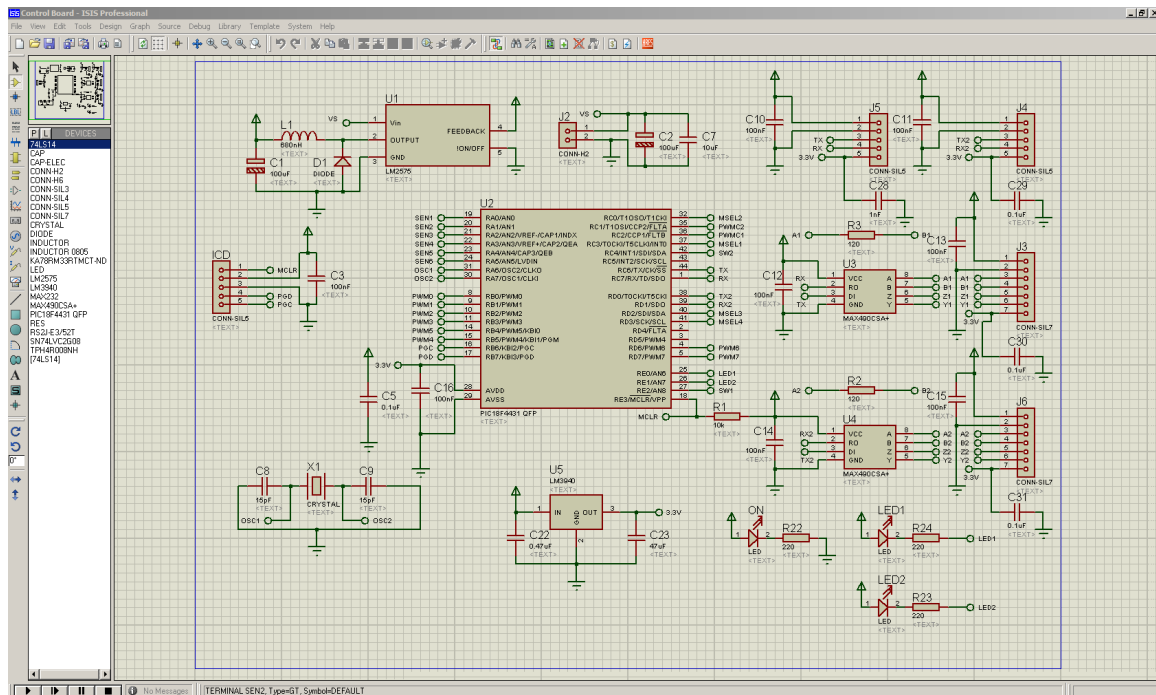


Figura 3.11: Janela principal do software ISIS.

é onde se encontra a maior parte do trabalho pois é em layout que a maioria dos erros podem ocorrer. Aqui é necessário o uso de todos os encapsulamentos certos, comprimentos de trilha apropriados dentre outros, que serão mais discutidos no capítulo 4. No ARES é onde, também, são gerados os arquivos CAM (*Computer Aided Machining*) para produção da placa.

3.7.2 SolidWorks

Desenvolvido pela Dassault Systèmes, SolidWorks é um software de CAD voltado para modelagem de sólidos, embora também possua fortes ferramentas de CAE (*Computer Aided Engineering*). Este software foi utilizado para modelar e preparar a produção do modelo mecânico da mão robótica. A figura 3.13 exibe a área de trabalho do software onde o modelo final da mão robótica (a ser apresentado no capítulo 4) está aberto.

3.8 Compilador CCS PCWHD IDE

O terceiro e último software de desenvolvimento utilizado foi o compilador, que neste caso foi o PCWHD versão 4.057, da CCS. O CCS é, em muitas formas, o compilador mais avançado do mercado. Possui diversas funções não presentes em outros compiladores além de um banco considerável de MCUs. O código será explicado no capítulo 5.

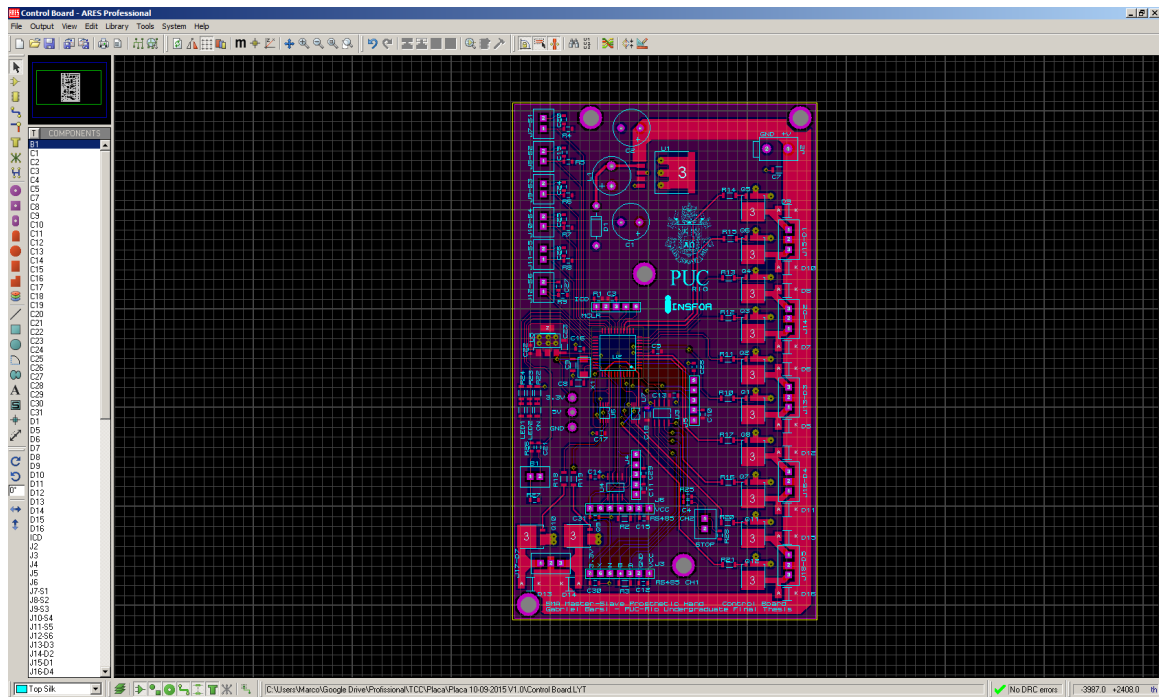


Figura 3.12: Janela principal do software ARES.

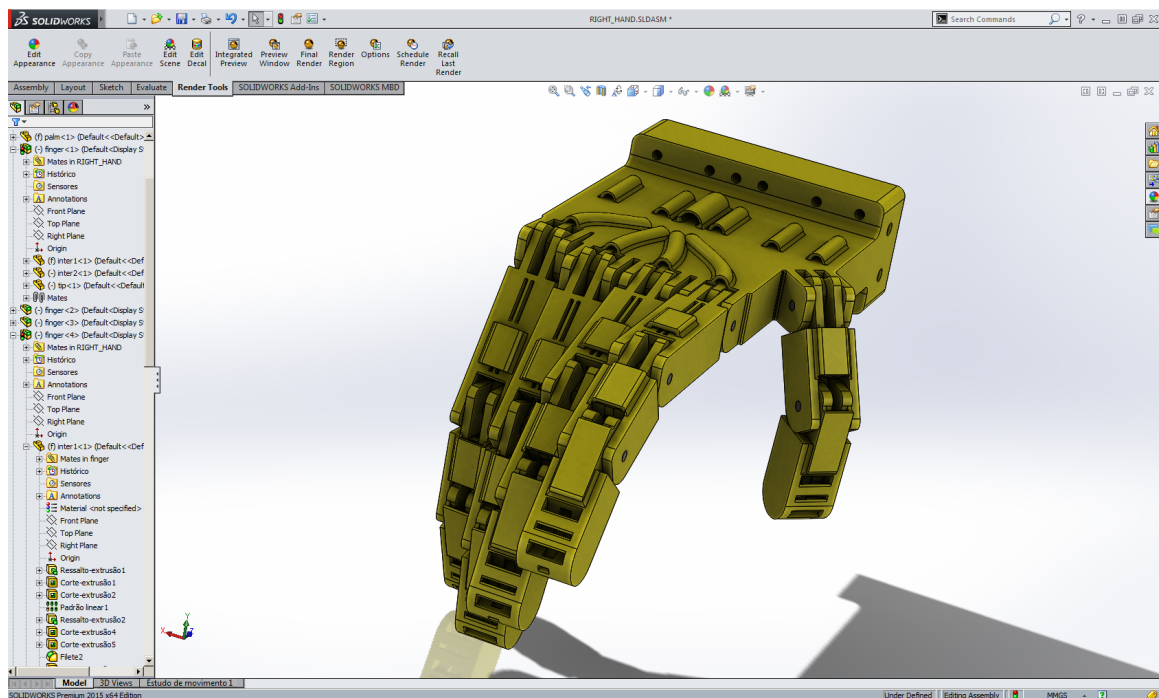


Figura 3.13: Janela principal do software SolidWorks.

4 Implementações

4.1 O sistema mestre-escravo

Como visto, este sistema é idealizado para dois usos: como prótese ou dispositivo de reabilitação e como sistema de manipulação remota. No primeiro caso todo o processamento de aquisição de sinais é local nenhuma comunicação é necessária. No segundo é necessária a comunicação entre o sistema mestre e o sistema escravo e, para isso, é necessária uma eletrônica embarcada para enviar os comandos.

Com isto em mente foram desenvolvidos dois sistemas eletrônicos. Uma placa de controle, contendo estágio de potência, reguladores, sensoramento e comunicação ficará junta ao robô, ou seja, será o sistema escravo. A outra placa, a mestre, ficará junto ao operador e somente será responsável por aquirir os comandos e envia-los ao sistema escravo.

4.2 Desenvolvimento do modelo mecânico

Desenvolvido no SolidWorks o design foi inspirado na anatomia humana, mas também considerando que o dedo será atuado por Flexinol, tentando deixar o sistema o menos complexo possível.

4.2.1 Modelo anatômico

O modelo apresentado é parcialmente disposto como a mão humana. Mecanicamente consiste de uma palma, três falanges por dedo sendo apenas duas para o polegar. Entre cada falange e entra a falange proximal e a palma existe uma junta rotativa livre. Vários graus de liberdade menos essenciais foram eliminados, resultando em um sistema com 10 DoF (dois por dedo) onde cada dedo é análogo a um braço robótico 2D RRR sub-atuado por dois atuadores. Mais graus de liberdade podem ser adicionados em designs futuros com pouca adaptação.

Conforme foi apresentado na seção 2.1 a anatomia humana utiliza músculos para gerar forças e tendões para transmiti-las. O roteamento de tendões na mão humana é feito de forma altamente sofisticada, de forma que um único tendão causa movimento tanto na falange média quanto na falange distal (figura 2.1) e outro tendão movimentar a falange proximal, tornando a mão humana muito prática para agarrar e manipular objetos. Este mecanismo é complexo, e se não controlado direito pode tornar o modelo inutilizável. Para contornar estas dificuldades a geometria foi feita de forma a primeiro rotacionar a falange média e, quando esta encontrar um bloqueio (seja um objeto ou fim de curso) a falange distal começa a rotacionar. As medidas foram tomadas a partir da mão de um humano masculino de 23 anos.

4.2.2 Design CAD em 3D

O design dos elementos da mão mecânica foi feito levando em conta todas as considerações feitas além de permanecer dentro das restrições do processo de fabricação aditivo (neste caso impressão 3D em ABS amarelo) de acordo com as regras em [43]. A figura 4.1 exibe o modelo completo da mão robótica. Cada parte será explicada e especificada separadamente.

Palma

A palma foi projetada para rotear os fios de Flexinol assim como fornecer a base para os dedos. Furos rosqueados fixam os fios de Flexinol e fornecem um mecanismo para ajustar a tensão individual de cada fio. Sua vista isométrica pode ser vista na figura 4.2.

O mecanismo de pivô foi feito de forma a restringir rotação além do ponto de extensão total e rotação fora do plano de trabalho de cada dedo. Os fios de Flexinol são roteados por cima para extensão e por baixo para flexão. Para simplificar o modelo todas as falanges proximais são atuadas pelo mesmo canal de PWM, portanto seus atuadores são unificados em paralelo

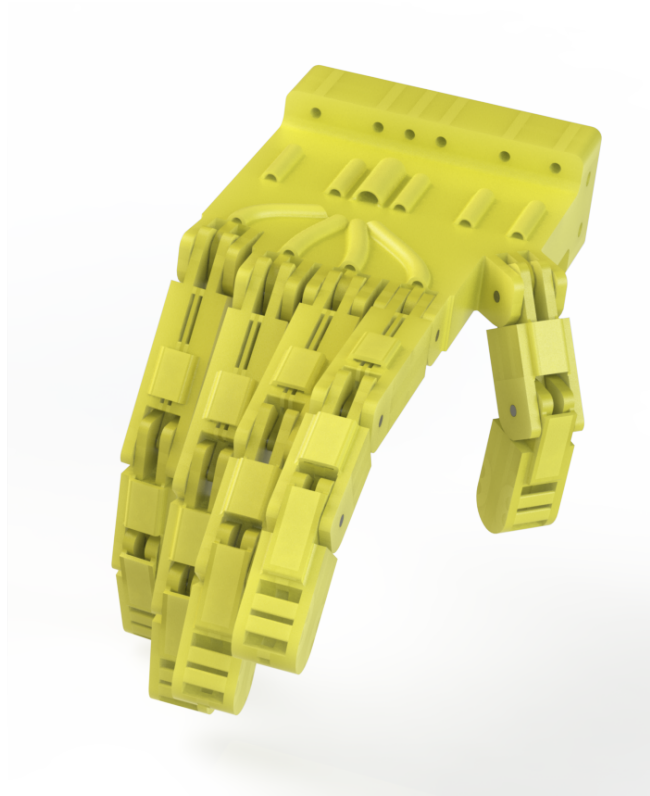


Figura 4.1: Render do modelo completo.

conforme visto na sub-seção 2.3.2. As vistas principais com cotas básicas e detalhamento de pontos importantes são exibidas na figura 4.3.

Falange Proximal

A falange proximal é o primeiro segmento com uma junta em cada extremidade. Por este segmento passarão quatro fios de Flexinol, um par flexão/extensão para o próprio segmento e outro par para a falange média e falange distal. Cada junta é feita, da mesma forma como na palma, de forma restringir sobre-extensão e rotação lateral. Um furo de $3mm$ permite a passagem de um pequeno eixo ou parafuso, a ser fixado externamente por um agente selante. A vista isométrica do modelo 3D pode ser vista na figura 4.4. As vistas principais com cotas básicas e detalhamento de pontos importantes são exibidas na figura 4.3. Nota-se, também, um canal largo na parte superior do segmento para fixação do sensor de flexão. Nota-se que as juntas foram projetadas de forma a permitir qualquer sequência ou número de segmentos, uma vez que o encaixe é o mesmo dado que a orientação foi respeitada. Isto é importante pois torna o design muito mais flexível.

Falange Média

A falange média foi modelada de forma muito similar a proximal, com a diferença de que agora são apenas dois canais de passagem para os fios de Flexinol e não existe ponto de ancoragem, já que os fios serão ancorados na falange distal. A vista isométrica do modelo 3D e as vistas principais com cotas básicas e detalhamento podem ser vistas nas figuras 4.6 e 4.7, respectivamente.

Falange Distal

A falange distal é o último segmento do dedo e, portanto, difere dos demais. A junta é exatamente igual as outras, porém na outra extremidade foi feito um acabamento redondo remetendo ao formato humano. De forma parecida com a falange média, aqui existem apenas

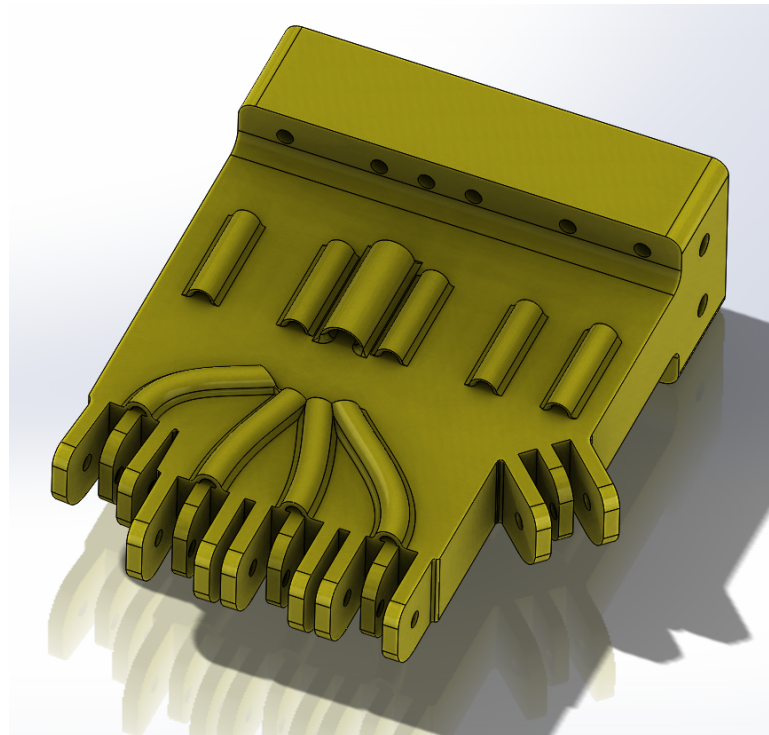


Figura 4.2: Vista isométrica 3D da palma.

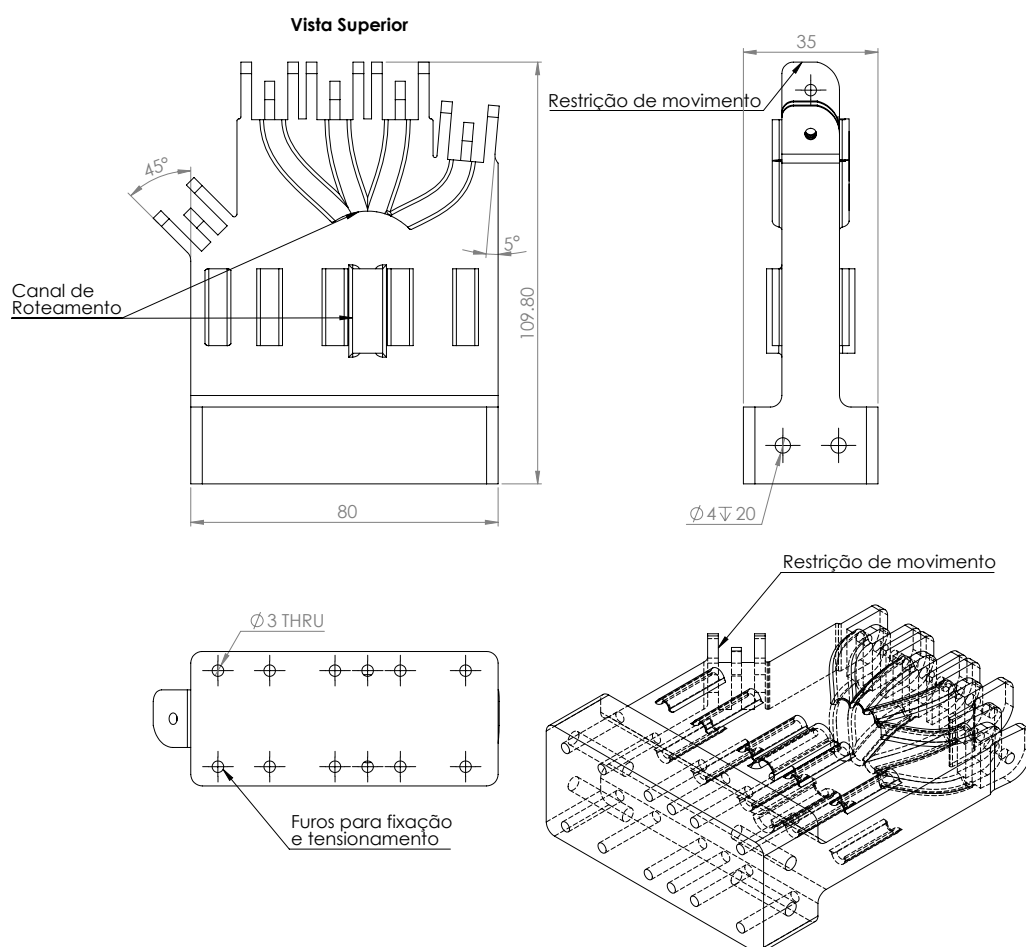


Figura 4.3: Vistas e cotas principais em [mm] da palma, detalhes indicados.

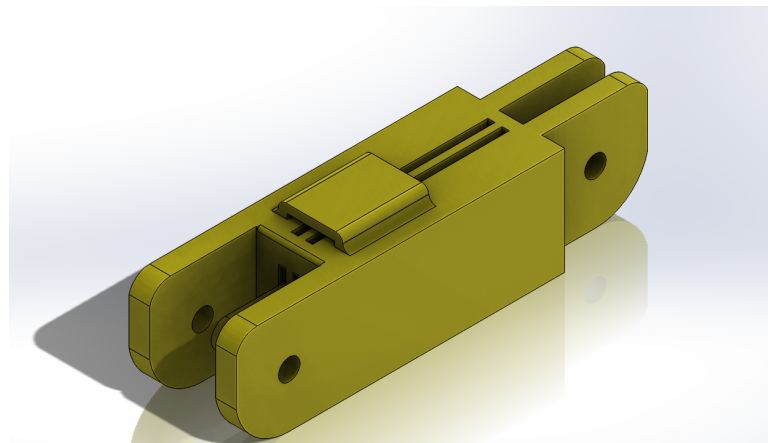


Figura 4.4: Vista isométrica 3D da falange proximal.

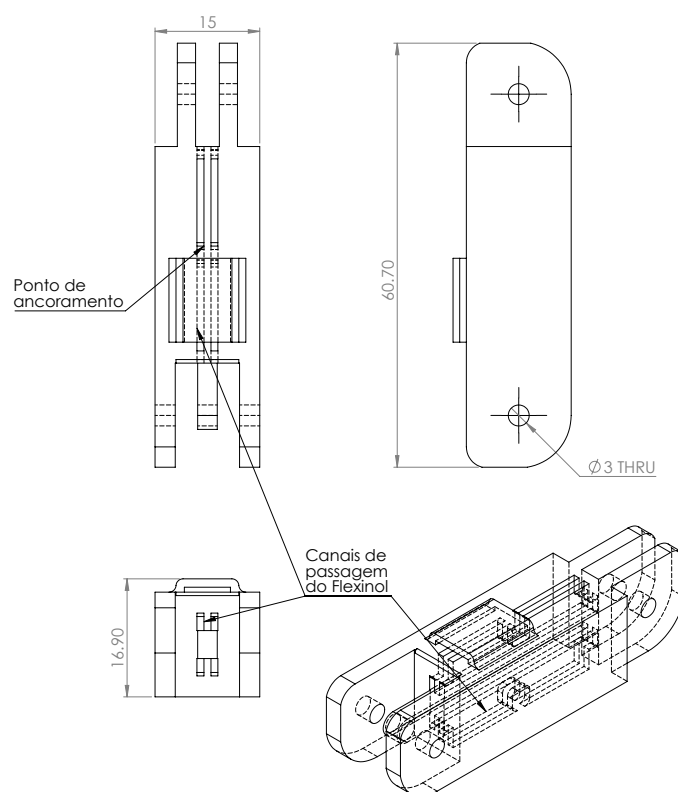


Figura 4.5: Vistas e cotas principais em [mm] da falange proximal, detalhes indicados.

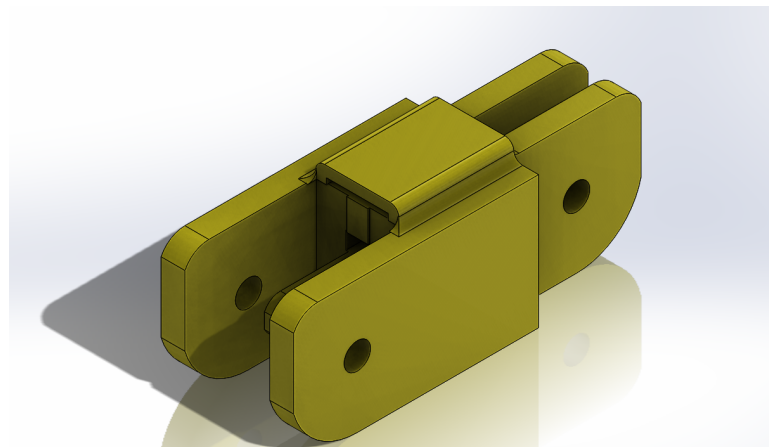


Figura 4.6: Vista isométrica 3D da falange média.

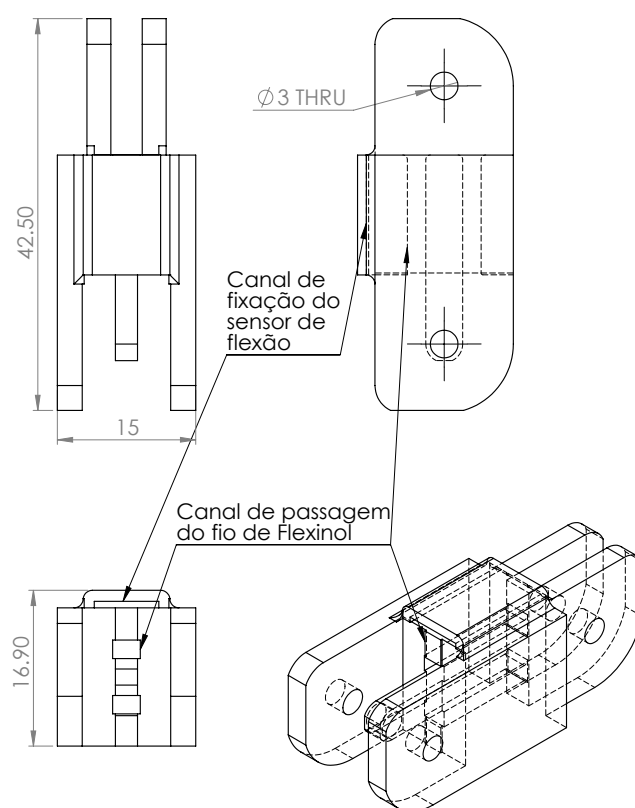


Figura 4.7: Vistas e cotas principais em [mm] da falange média, detalhes indicados.

dois canais de passagem para os fios de Flexinol, porém neste segmento foram feitos pontos de ancoragem onde pode-se amarrar os fios de Flexinol, para fixá-los. Neste segmento, também, o canal do sensor de flexão termina. A vista isométrica do modelo 3D e as vistas principais com cotas básicas e detalhamento podem ser vistas nas figuras 4.8 e 4.9, respectivamente.

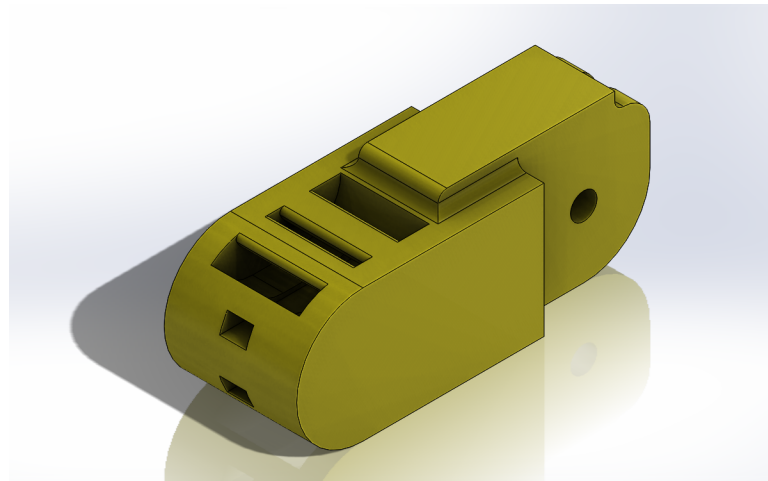


Figura 4.8: Vista isométrica 3D da falange distal.

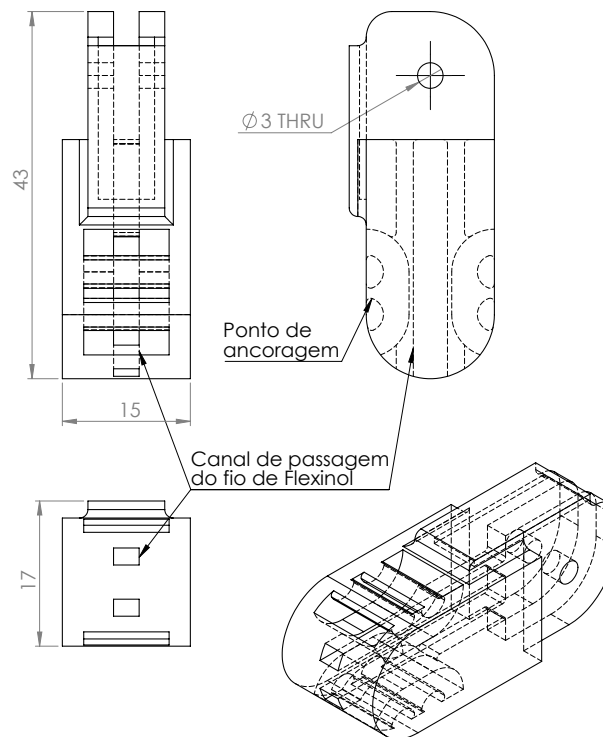


Figura 4.9: Vistas e cotas principais em [mm] da falange distal, detalhes indicados.

4.3 Fabricação do modelo mecânico

Tratando-se de um protótipo o método de fabricação empregado foi o de impressão 3D em ABS (material plástico). A impressora utilizada foi a BFB 3D Touch da empresa Bits From Bytes, vista na figura 4.10. A impressora possui dois extrusores, resolução de até 0.125mm e capacidade de imprimir tanto em ABS quanto em PLA. O uso desta impressora foi cedido pelo departamento de design da PUC-Rio.

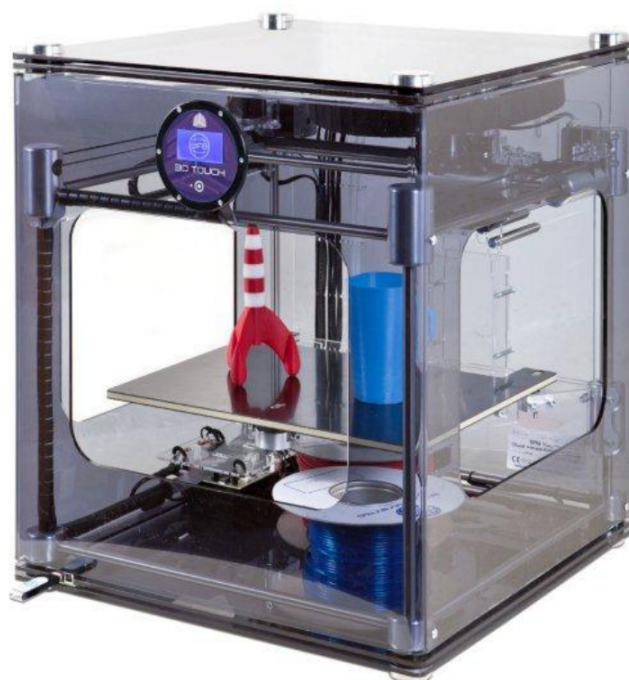


Figura 4.10: Impressora 3D BFB 3D touch.

Impressão 3D embora revolucionário requer muitos cuidados para que o usuário obtenha um resultado desejável. Para folgas e ajustes, além de boas práticas, o documento utilizado como referência foi [43], além dos cuidados tomados durante o projeto das partes.

4.3.1 Formatação do modelo para impressão

Os modelos projetados no SolidWorks foram exportados para o formato padrão STL (*STereoLithography*) que representa apenas as superfícies do modelo na forma de triângulos. O arquivo STL foi então aberto no programa Axon 2, da própria Bits From Bytes. A janela de trabalho do Axon 2 pode ser vista na figura 4.11. A superfície cinza é a mesa de trabalho, e a peça está representada em azul.

Após importar o modelo o projetista deve escolher a orientação correta para impressão ótima além de parâmetros como suporte, *rafting* e preenchimento. Suporte, como o nome sugere, deve ser escolhido de forma suportar partes do modelo que possuem algum tipo de vão livre, ou peças muito altas. *Rafting* é o processo de construir uma "cama" antes de iniciar a impressão, isto deve ser feito para isolar a superfície da peça de imperfeições que ocorrem no contato com a mesa de trabalho. Preenchimento é a porcentagem de material que irá preencher a peça impressa. Normalmente não se preenche 100% da peça, uma vez que acarretará em gasto excessivo de material e um tempo de impressão elevado. As configurações escolhidas, após tentativas falhas e opiniões de profissionais experientes foi decidido utilizar as configurações exibidas na figura 4.12. Foi escolhido fazer o *raft* e suportes, assim como preenchimento de 50% na forma de colmeia.

Após configuração é necessário o processamento das formas para gerar o arquivo CAM a ser utilizado pela impressora, o arquivo resultante já inclui o *raft* e suportes, e o projetista pode mover o *slider* de altura para ter uma prévia da impressão. A figura 4.13 exibe o arquivo resultante assim como os artifícios da impressão 3D.

Após confirmação de que tudo está correto o arquivo é importado para a máquina onde o processo automatizado é iniciado.

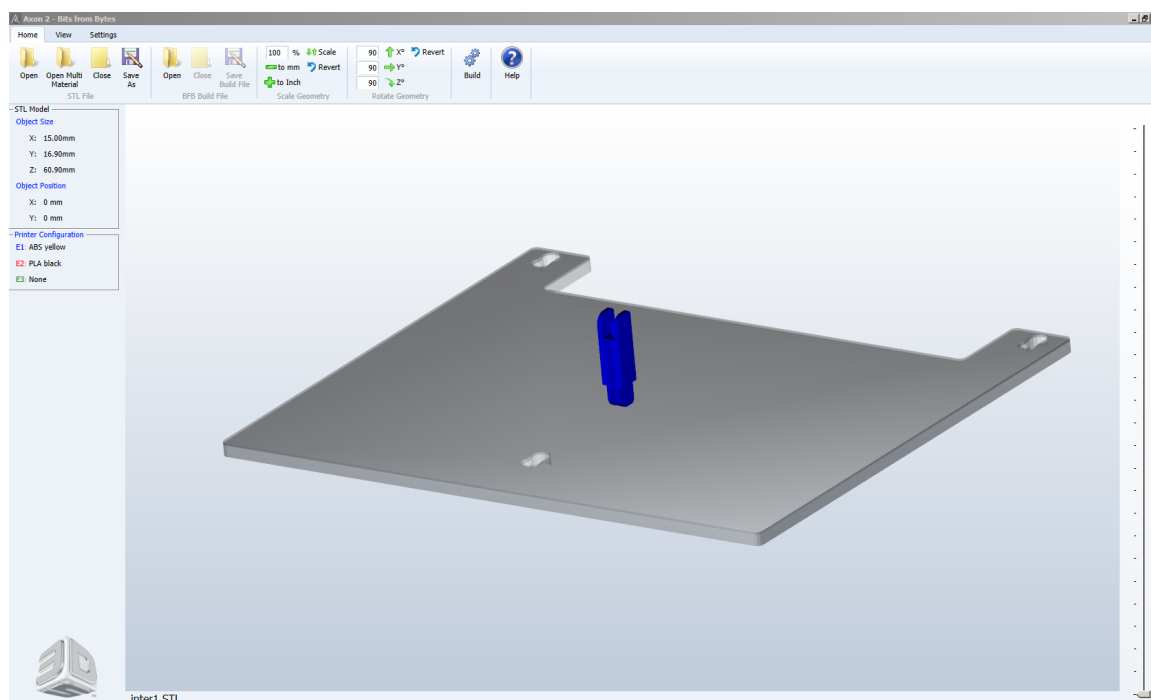


Figura 4.11: Janela principal do software Axon 2.

4.4 Desenvolvimento do circuito esquemático das placas

Esta seção focará no projeto das placas eletrônicas, mais especificamente o esquemático das mesmas no software ISIS apresentado anteriormente.

4.4.1 Placa de controle

A placa de controle é a principal do sistema, embora seja, no caso da utilização remota, o sistema escravo. O esquemático completo pode ser visto no apêndice 9.4.

Como a carga é de caminho único (o sentido da corrente não influencia a atuação) nenhuma configuração além da vista anteriormente na figura 3.8 será necessária. Como temos um atuador para extensão e um para contração é útil coloca-los no mesmo conector e no mesmo canal de PWM, uma vez que somente um de cada vez será acionado. A figura 4.14 exibe a configuração escolhida. Os diodos são de circulação, necessários uma vez que embora a maior impedância da carga seja resistiva, por ser um fio longo e sinuoso pode existir uma componente indutiva que não deve ser ignorada. O conector liga a fonte (V_S) ao dreno de cada MOSFET. Esta configuração permite o uso de apenas três vias por par de atuador onde seriam necessários 4 numa configuração tradicional.

O microcontrolador foi conectado conforme a figura 4.15. Os sensores foram conectados nas portas **ANO-5**, canais de PWM nas portas **PWM0-7** e **CCP1-2**. Comunicação serial foi colocada tanto nas portas padrão **RX** e **TX** como nas portas **D0** e **D1**, sendo estas últimas as emuladas por software. As vias de programação também foram devidamente conectadas. As demais portas foram escolhidas de forma a otimizar o layout, uma vez que são simples portas lógicas sem nenhum requisito especial (LEDs, chaves e seletor de multiplexação).

Como visto anteriormente teremos seis leituras de posição portanto o mesmo número de canais AD foram implementados. O esquemático de conexão dos sensores pode ser visto na figura 4.16. Além da resistência divisora foi colocado um capacitor para gerar um filtro RC padrão com o intuito de eliminar ruídos. A única peculiaridade desta disposição é que, como a resistência do sensor varia, a frequência de corte varia junto, realizando um filtro passa-baixas com frequência de corte variável $964Hz \leq f_c \leq 2893Hz$.

Os reguladores de tensão e circuitos de comunicação serial foram conectados de acordo com o *datasheet*, conforme as figuras 3.3, 3.6 e 3.9. Após colocar os componentes princi-

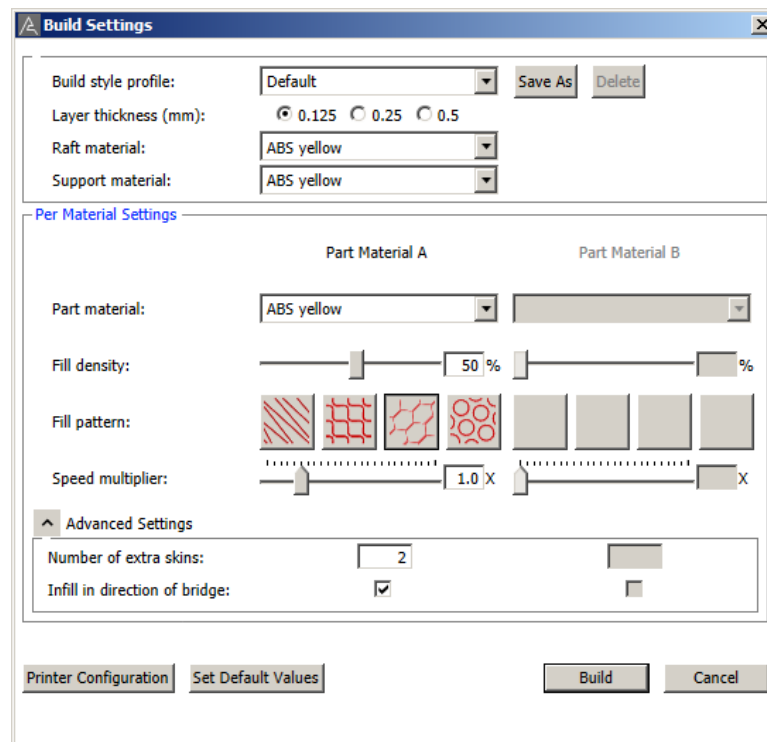


Figura 4.12: Configurações utilizadas na impressão das peças.

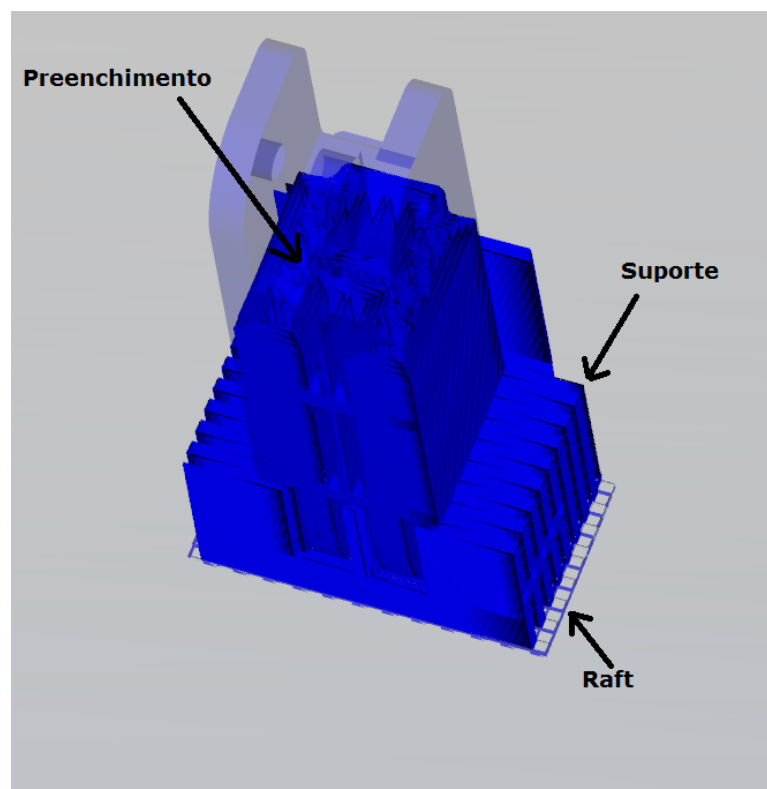


Figura 4.13: Arquivo de impressão.

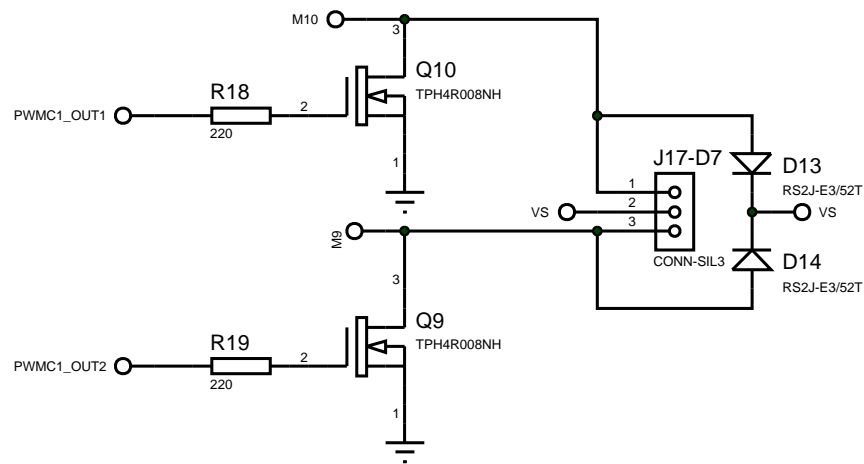


Figura 4.14: Esquemático do estágio de potência para atuar uma carga diferencial RL .

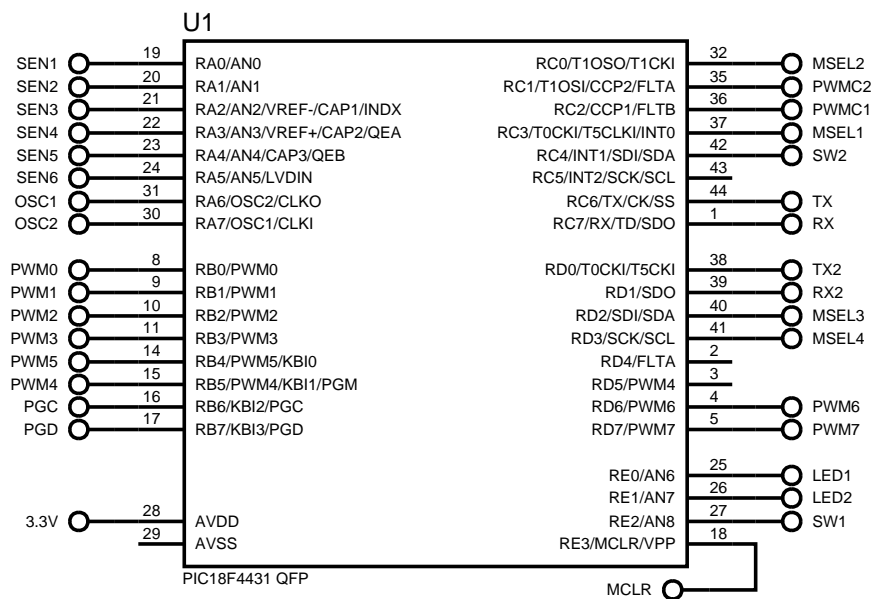


Figura 4.15: Associação de portas no PIC18F4431 QFP.

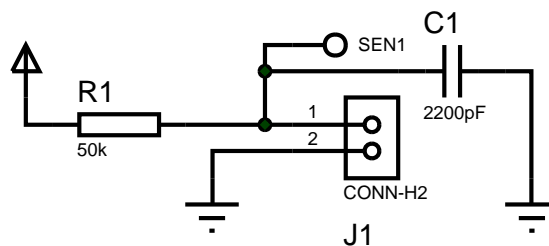


Figura 4.16: Circuito de aquisição analógica empregando um filtro RC dentro de um divisor de tensão.

país foram feitos os conectores de comunicação, potência e foram posicionados capacitores de desacoplamento de forma estratégica perto de cada fonte de energia e circuito integrado. Estes capacitores são prática padrão para evitar danos por picos de tensão ou ESD (*Electrostatic discharge*). Estes componentes podem ser vistos no apêndice 9.4.

Os conectores de comunicação, além das vias de comunicação do RS-485 e do terra de referência, contêm uma linha de 5V e uma de 3.3V. O motivo desta decisão é simples: a placa mestra necessita destas tensões tanto para alimentação quanto para fonte externa de tensão de comparação do ADC para ter uma leitura consistente dos sensores nas duas partes.

4.4.2 Placa de aquisição

A placa de aquisição é muito mais simples do que a de controle. Ela somente precisa ler seis canais analógicos e os enviar via serial para a placa de controle. Desta forma foi feito o esquemático seguindo exatamente as diretrizes da placa de controle, apenas mudando algumas portas digitais simples. O esquemático completo pode ser visto no apêndice 9.5.

4.5 Desenvolvimento do layout das placas

Esta seção focará no desenvolvimento do layout das duas placas no software de CAD ARES apresentado anteriormente.

4.6 Desenvolvimento do layout da placa de controle

A placa de controle teve seu desenvolvimento inicial com algumas considerações importantes que definiram o resto do processo, além das regras padrão como não realizar ângulos retos no roteamento de trilhas, manter ao menos 5th (5/1000in unidade padrão de desenho de placas, e aqui referenciada como 1th = 1/1000in) de tolerância, caminhos de retorno para o terra sempre o mais curtos possível e realizar o mínimo de passagens de planos (trocar, por exemplo, do plano superior para o inferior) o possível durante o roteamento de trilhas. As considerações específicas foram:

- A placa será de duas camadas e utilizaria quase que unicamente dispositivos SMD (*Surface Mount Devices*, dispositivos montados na superfície da placa).
- O encapsulamento de resistores e capacitores será ou 0805 ou 1206 métrico, uma vez que a placa será soldada a mão.
- O plano superior (*top copper*) será das tensões positivas, ou seja, 5V do regulador e V_S da fonte.
- O plano inferior será completamente dominado pelo plano terra (abreviado aqui para GND, do inglês *ground*).
- Tratando-se de um protótipo todos os dispositivos serão montados na parte superior da placa, de forma a facilitar medições, estudos e acesso.
- Toda e qualquer forma de se evitar erros fatais que inutilizariam a placa serão implementados, tais como vias de acesso e locais de *jumping* (ligar uma trilha a outra).
- A placa não terá nenhuma restrição de tamanho, porém nenhum espaço será desperdiçado sem propósito.

Feitas estas considerações se deu início o processo de desenho da placa. O resultado final pode ser visto no apêndice 9.6. No layout anexado somente a camada superior está representada, uma vez que a inferior é um plano sólido de cobre para o GND, salvo poucas trilhas de comunicação. Durante todo o processo os capacitores de desacoplamento foram colocados sempre junto com o dispositivo ou conector devido.

Inicialmente foram feitos todos os encapsulamentos inexistentes na biblioteca padrão do ARES. Os principais foram o do LM2575 e do MOSFET. Em seguida foram colocados os itens mais sensíveis, primeiramente a MCU (U2), o conector de programação (ICD) e o oscilador (X1)

foram posicionados. O motivo desta ordem são os sinais de alta frequência que operam nas trilhas que os ligam, é importante manter trilhas de alta potência e/ou alta frequência o mais curtas o possível para evitar efeitos *RLC* parasitas que ocorrem com a indutância, capacitância e resistência natural de qualquer material sendo, neste caso, o cobre.

Colocados os dispositivos críticos e sensíveis, foi feita a linha de potência. Observando-se o apêndice 9.6 pode-se notar que existem dois planos principais. A maioria da placa está inundada com o plano de $5V$, uma vez que a maioria dos dispositivos utilizam esta tensão como alimentação, e necessitam que a mesma fique o mais estabilizada o possível. Próximo as arestas lateral direita e inferior podemos ver um segundo plano, que se estende também até a metade da aresta superior. Este plano é o de alta tensão (V_S) que alimenta o regulador LM2575 e todos os conectores para os fios de Flexinol.

A próxima fase foi o posicionamento dos MOSFETS e diodos de circulação. O layout desta placa deve ser intuitivo o suficiente para ser didático, portanto os MOSFETS foram colocados em ordem e seus *gates* foram devidamente ligados. A *source* do MOSFET foi ligada ao GND por meio de vias de grande porte, enquanto que o dreno foi ligado ao conector por meio de trilhas largas e curtas, deixando o MOSFET muito próximo do conector. Nota-se também a proximidade dos diodos de circulação (D5-16) do conector dos fios de Flexinol e do MOSFET. O transiente que estes diodos devem suprimir são do pior tipo para uma eletrônica: alta tensão e alta frequência, por isso a posição destes diodos seja, possivelmente, o posicionamento mais crítico de todo o projeto, principalmente considerando a flexibilidade que esta placa oferece e o fato de estar superdimensionada abrindo portas para cargas maiores em outros projetos em que possa vir a ser utilizada. Nesta etapa foi respeitada a regra de pelo menos $10th/A$ para a largura da trilha sendo que a mesma foi dimensionada para $6A$, muito acima do que realmente será pedido de cada canal.

A próxima etapa foi posicionar o regulador LM3940. Este regulador deve ficar próximo ao PIC18F4431 para evitar ao máximo flutuações na linha de $3.3V$, já que qualquer flutuação afetará negativamente todas as medições. Este regulador foi colocado próximo da MCU e um plano regional de GND foi gerado para garantir boa dissipação térmica caso surgisse a necessidade.

Em seguida foi posicionada a parte lógica. as portas AND (U6 e U7) e o MAX490 (U3) referente a via principal de comunicação foram colocados primeiro e mais próximos da MCU. Em seguida o MAX490 secundário (U4) foi posicionado. Juntamente aos MAX490 foram construídos os conectores de comunicação, foi feito o principal (RS-485) e também um secundário (RS-232) com o intuito de ter uma saída caso o protocolo mais sofisticado não funcionasse.

A última etapa consistiu em posicionar os componentes menos importantes. Dois botões com *anti-bounce* (sendo um deles uma interrupção para emergências) foram colocados em locais vazios assim como o conjunto de três LEDs (dois de estado e um indicando se a placa está ou não ligada). Além destes foram colocados pontos de acesso direto ao GND, $3.3V$ e $5V$ caso fosse necessário fonte externa para algum e furos de fixação da placa com medidas inteiras.

A serigrafia foi feita após o fechamento do layout e consistiu em apenas fornecer informações práticas ao usuário, assim como dados sobre a placa e instituições envolvidas.

4.7 Desenvolvimento do layout da placa de aquisição

A placa de aquisição teve um processo mais simples, sendo a única dificuldade a restrição de espaço. Como esta placa deverá ser instalada em uma espécie de luva ou similar é importante manter o tamanho o menor possível. De forma semelhante à placa de controle o plano superior é inteiro de $5V$ enquanto que o inferior é GND. Nesta placa não foi necessário o uso de um cristal devido a sua simplicidade. O primeiro componente a ser colocado foi o conector de programação e em seguida o MAX490.

Após a colocação dos componentes principais foi-se colocando o resto da melhor forma a fim de manter a área da placa a menor possível. O resultado do processo pode ser visto no apêndice 9.7.

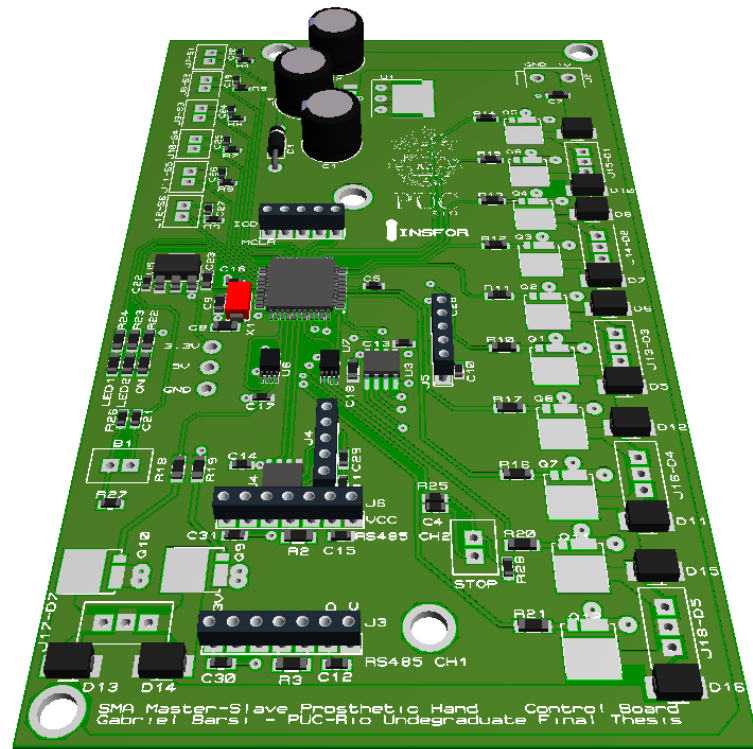


Figura 4.17: Modelo tridimensional da placa de controle.

4.8 Fabricação das placas impressas

Esta seção explicará o processo de fabricação desde a pré-produção até a montagem final das placas.

4.8.1 Pré-produção

A pré-fabricação é feita ainda no ARES. a primeira etapa é realizar uma verificação visual do 3D da placa. O ARES automaticamente gera um modelo tridimensional para verificar se não existem componentes em contato ou pontos inacessíveis. A visualização em 3D também facilita enxergar erros menos visíveis no ambiente normal de projeto. A figura 4.17 exibe o modelo.

Após a verificação visual e re-checagem de todas as medidas e encapsulamentos é feita a checagem de pré-produção. Esta rotina tem como intuito verificar se não existem erros nas malhas, interferência de trilhas, ou qualquer coisa que interfira na fabricação das placas. Ambas as placas projetadas passaram nestes testes sem problemas, o que é de se esperar se o projetista não interferiu nas regras padrão de design.

4.8.2 Geração dos arquivos Gerber/Excellon

Nesta etapa o ARES prepara o arquivo CAD a ser enviado para a fábrica. Como placas de circuito impresso são normalmente feitas em grandes quantidades e em painéis contendo múltiplas placas (esta prática reduz drasticamente os custos de fabricação) é vantajoso criar um layout de fabricação que utiliza o máximo possível de uma região quadrada que contenha as placas a serem fabricadas. Seguindo esta estratégia foi montado o painel visto no apêndice 9.10. Para cada placa de controle produzida serão produzidas também duas mestras. Embora desnecessárias as placas mestras extras virão de material que seria descartado, não havendo desperdício desta forma.

Quando gerando os arquivos de produção, é importante atentar-se aos limites e tolerâncias. Neste caso a distância entre as bordas do painel e as placas, assim como a distância entre as

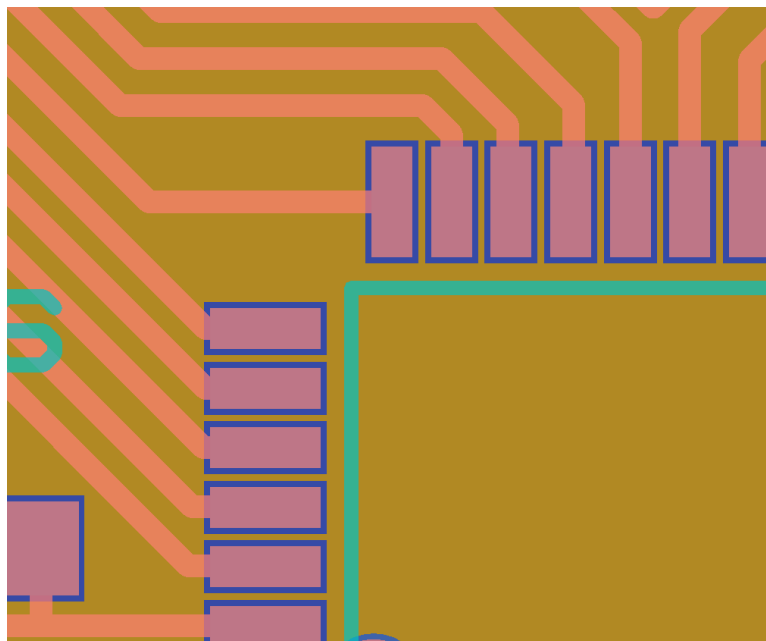


Figura 4.18: Detalhe do arquivo Gerber com a nova tolerância para a camada *solder resist*.

placas foi configurada como pelo menos 100th, o que é uma medida excessiva porém garante a qualidade do produto final. O segundo fator a se considerar é a tolerância da camada *solder resist*, a camada responsável por impedir que o estanho da solda faça conexão com o cobre abaixo. O ARES exporta os arquivos Gerber com a tolerância padrão para a *solder resist* de 10th o que, no caso de existirem componentes com *pitch* (espaçamento entre os pinos físicos do componente) muito pequeno, pode ocasionar a inexistência da camada nas proximidades dos *pads* (região de solda dos pinos de componentes SMD) destes dispositivos. Como é o caso com o PIC18F4431 e os SN74LVC2G08DCTR a tolerância foi reduzida para 5th, o que está bem dentro dos padrões da indústria de prototipagem de placas. A figura 4.18 exibe o detalhe do vértice superior esquerdo do PIC18F4431 com a nova tolerância. Pode-se identificar a clara separação dos *pads*, com a tolerância padrão veria-se uma faixa azul continua percorrendo os pinos.

4.8.3 Produção

A produção das placas inicialmente seria feita no Brasil, porém o alto preço das empresas disponíveis desencorajou esta iniciativa e o serviço foi prestado por uma empresa chinesa, a ShenZhen2U. A quantidade mínima de painéis é 5, portanto foram feitas 5 placas de controle e 10 de aquisição. As placas foram feitas na espessura padrão de painéis de silício de 1.6mm assim como as trilhas de cobre cuja espessura padrão é 1oz (1 onça por polegada quadrada). A única outra opção escolhida foi o acabamento HASL (*Hot Air Solder Leveling*), que aplica uma fina camada de solda para isolar o cobre da atmosfera, não só nivelando todos os *pads* como também evitando corrosão do cobre. O custo final de 5 painéis, excluindo frete, foi de US\$46,50.

O resultado final da produção pode ser visto nas figuras 4.19, 4.20 e 4.21.

4.8.4 Montagem

A montagem das placas consistiu em soldar os componentes manualmente utilizando uma estação de solda aterrada, e fluxo de solda para garantir bom espalhamento do estanho no *pad*. Outros cuidados incluíram boa uniformidade térmica e banho de álcool isopropílico ao final da montagem. As figuras 4.22, 4.23 e 4.24 exibem o resultado final da montagem.

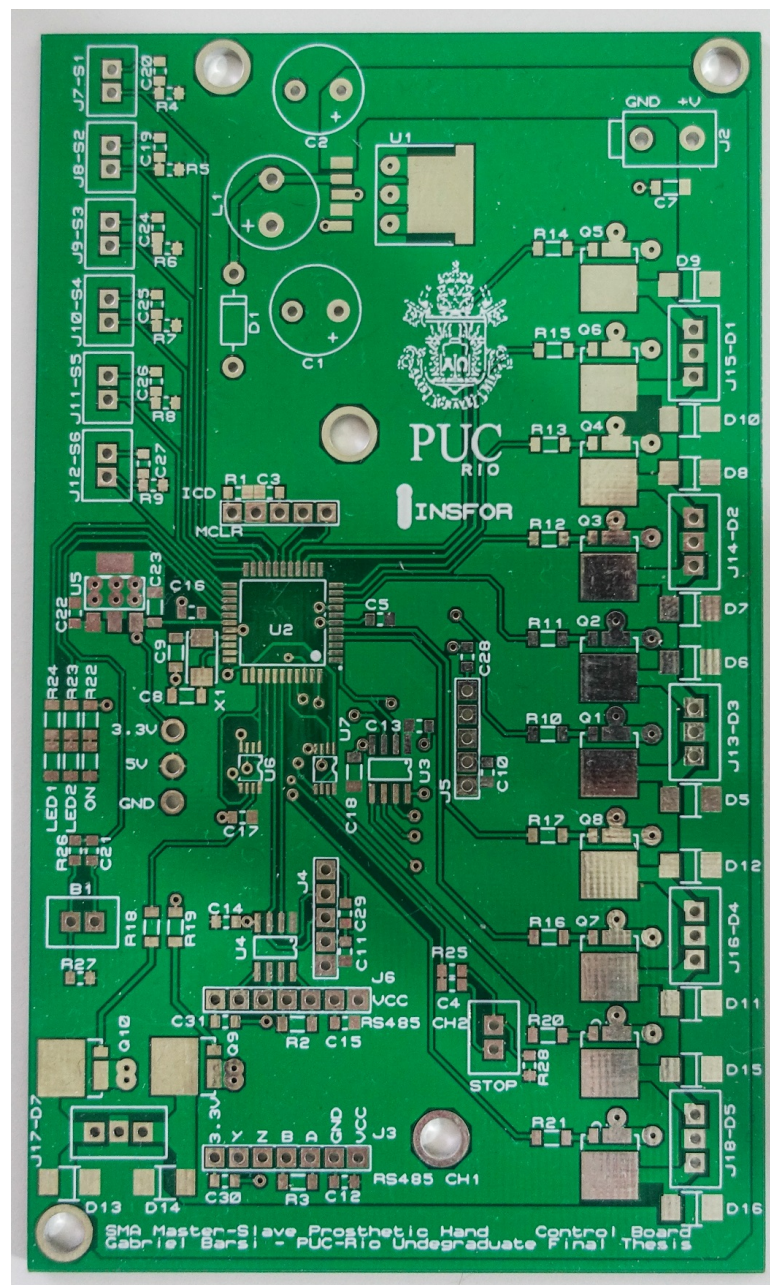


Figura 4.19: Resultado da produção da placa de controle.

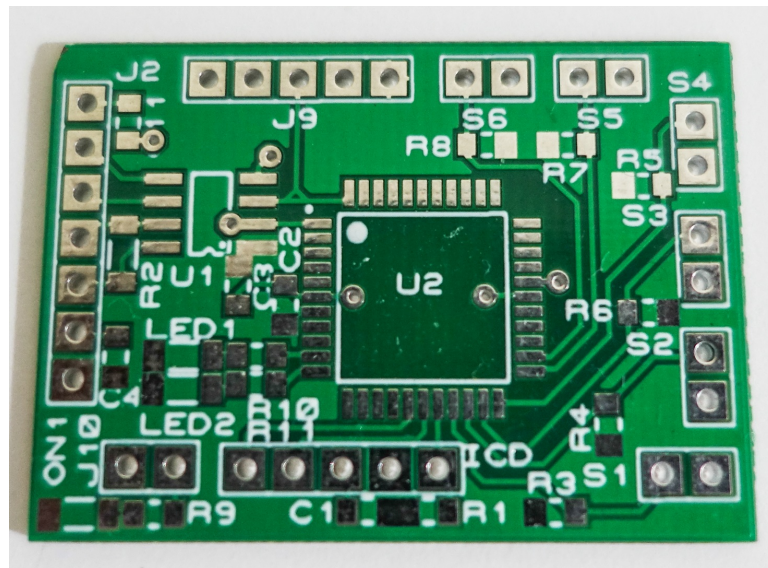


Figura 4.20: Resultado da produção da placa de aquisição.

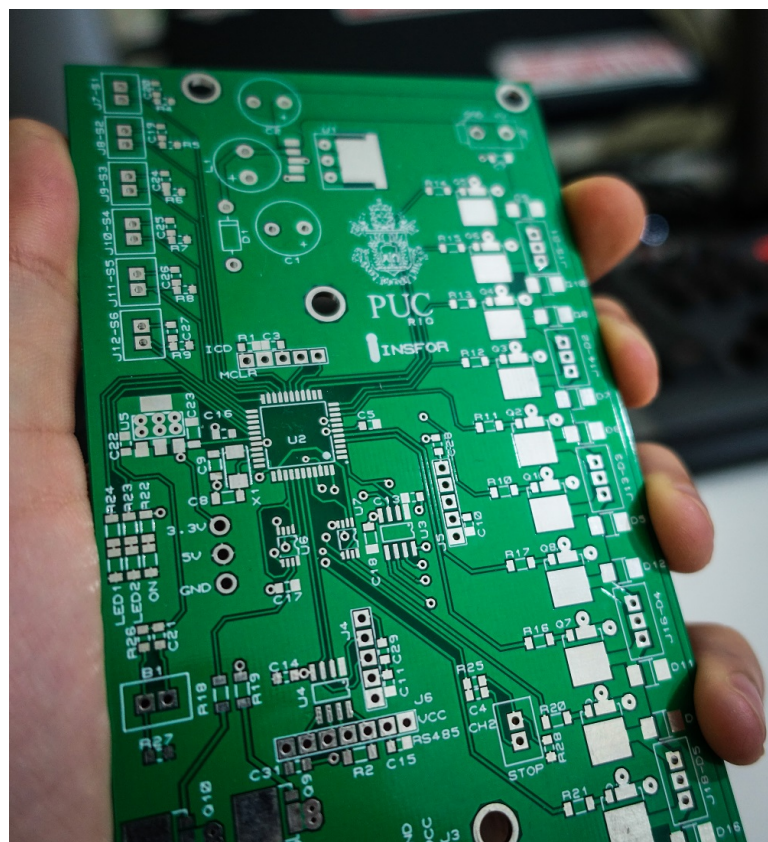


Figura 4.21: Resultado da produção da placa de controle, sendo manuseada.



Figura 4.22: Resultado final da placa de controle, exibida ligada e funcionando.



Figura 4.23: Placa de controle, material isolante térmico e elétrico protege a parte lógica.

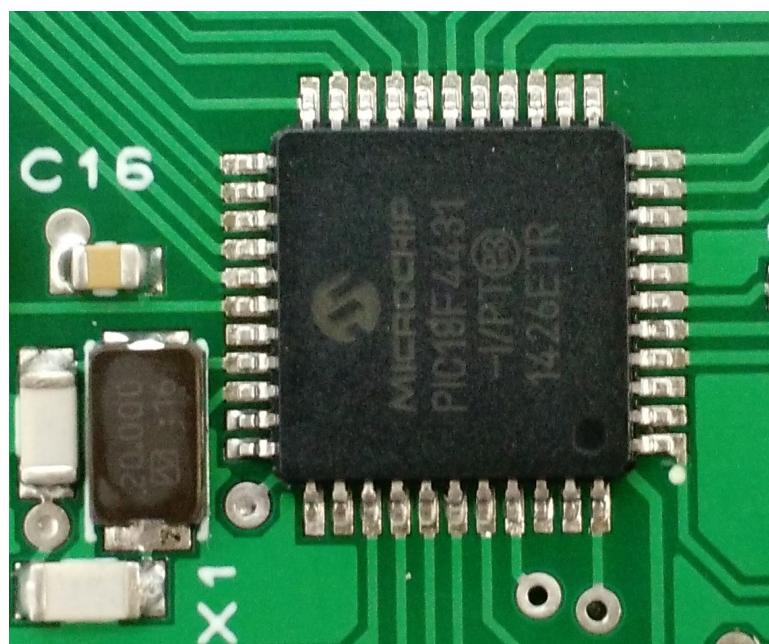


Figura 4.24: Detalhe das soldas dos pinos do PIC18F4431, o espaço entre a metade de cada pino (*pitch*) é de apenas 0.8mm.

4.9 Considerações sobre as implementações

Os processos descritos neste capítulo apresentaram cada um uma dificuldade única. A modelagem e fabricação do sistema mecânico foram exercícios de bom senso e desenvolvimento intuitivo. A montagem das placas foi um desafio técnico que requer habilidade, destreza e cuidado. O desenvolvimento e produção das placas foi, no entanto, uma tarefa extremamente difícil e que, por não estar no escopo de nenhuma matéria oferecida na graduação da PUC-Rio, exigiu amplo conhecimento prévio e experiência, uma vez que as tecnologias e dispositivos implementados nestas placas são tecnologia recente e de ponta.

Mesmo com estas dificuldades todos os componentes no trabalho funcionaram perfeitamente em testes iniciais, e a placa conseguiu lidar com cargas excedendo 15A, muito além da necessidade desta aplicação porém prova a robustez e flexibilidade do sistema eletrônico desenvolvido.

5 Software

O software foi puramente desenvolvido em C++ dentro do ambiente de desenvolvimento da CCS, o PCWHD v4.057. Os códigos foram completamente desenvolvidos para esta aplicação e encontram-se nos apêndices 9.8 e 9.9. Abaixo serão explicados trechos específicos.

5.1 Configuração da MCU

A configuração da MCU consiste em configurar o compilador para que o mesmo possa, em seguida, configurar a MCU. Abaixo está o início da configuração:

```
//Setup device and addresses
#include <18F4431.h> //set the device
#define adc=8 //use 8-bit ADC

#define HS //high speed oscillator
#define delay(clock=20000000) //20MHz oscillator
#define rs232(baud=9600, xmit=PIN_C6,rcv=PIN_C7, STREAM=COM_A) //setup serial port A
#define rs232(baud=9600, xmit=PIN_D0,rcv=PIN_D1, STREAM=COM_B, DISABLE_INTS) //setup serial port B
```

Este trecho se inicia dizendo ao compilador qual MCU está sendo implementada. Em seguida começamos a configurar a MCU configurando o ADC como 8bits, acionando o fuso para oscilador de alta velocidade ($\geq 4MHz$) e configurando as portas seriais com *baudrate* de 9600bits/s, pinos de envio e recebimento e identificadores. Adicionalmente foi desabilitada a interrupção da segunda porta serial, uma vez que esta se comunica com um PC e qualquer dado que não chegar do mestre não deve ser tratado.

Os bytes de controle **OVDCOND** e **OVDCONS** devem ser configurados manualmente de acordo com o fabricante [38]. Abaixo associamos os endereços fornecidos ao nome para que possam ser manipulados posteriormente.

```
#byte OVDCOND = 0xF6B //assign OVD registers addresses
#byte OVDCONS = 0xF6A
```

5.1.1 Configuração primária

Esta etapa encontra-se na função principal. Embora esta sendo apresentado fora de ordem, esta é a forma lógica com a qual o processador percorre o código. Aqui serão feitas diversas configurações secundárias para que a MCU gere as saídas esperadas.

```
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_64); //setup TIMER0, PRSCLR = 64
setup_timer_1(T1_INTERNAL|T1_DIV_BY_8); //setup TIMET1, PRSCLR = 8
setup_timer_2(T2_DIV_BY_16, 255,1); //setup TIMER2, PRSCLR = 16, ovf=8bits
enable_interrupts(GLOBAL); //enable interrupts
enable_interrupts(INT_RDA); //enable serial receive interrupts
ext_int_edge( H_TO_L ); //enable external interrupts on falling edge

setup_adc_ports(ALL_ANALOG|VSS_VDD); //setup ADC with external reference
setup_adc(ADC_CLOCK_INTERNAL); //natural ADC clock
set_adc_channel(0); //initial channel
```

O bloco acima é a configuração inicial na função principal. Primeiramente os timers são configurados da mesma forma como visto na subseção 3.1.3, em seguida as interrupções são ativadas. A segunda parte liga e configura o ADC para utilizar a referência de 3.3V. A próxima fase é configurar todos os PWM. Estas configurações foram feitas de acordo com a documentação do compilador para atingir a frequência alvo de 1250Hz, que é um valor seguro porém rápido o suficiente. Estas configurações podem ser vistas abaixo:

```
setup_power_pwm_pins(PWM_BOTH_ON,PWM_BOTH_ON,PWM_BOTH_ON,PWM_BOTH_ON); //setup PMWX, all ON
setup_power_pwm(PWM_FREE_RUN, 1, 0, POWER_PWM_PERIOD, 0, 1,0); //setup modes

setup_ccp1(CCP_PWM); //CCPX as PWM, f_pwm = 1250Hz
setup_ccp2(CCP_PWM);
```

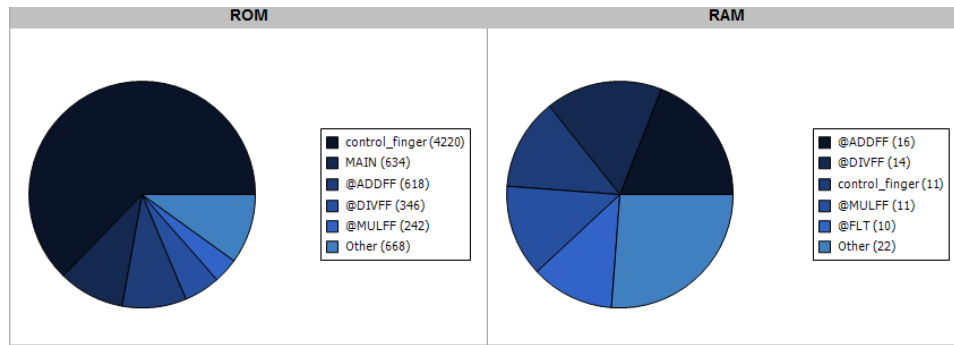



Figura 5.1: Alocação das memórias por função do programa.

Como **TIMER2** está configurado para que o *overflow* ocorra em 255 bits, este será o valor equivalente a uma razão cíclica de 100% nos PWM provenientes das portas CCPX. O controle da razão cíclica das portas PWMX é uma fração de seu período, definido no código, portando uma variável do tipo flutuante ditará onde no contador o PWM mudará de estado.

5.2 Rotina de controle

A rotina de controle foi implementada na forma de uma única função, que recebe como parâmetro qual junta deseja-se controlar. Seu protótipo encontra-se abaixo:

```
void control_finger(int idx); //control one joint
```

Dentro desta função existem diversas sub-rotinas. Embora seja a função que ocupe o maior espaço de programa (ROM) não é a que ocupa o maior espaço de processamento (RAM) uma vez que existem inúmeras operações ocorrendo para, por exemplo, gerar os PWMs e o envio de dados. As estatísticas de ocupação de memória podem ser vistas na figura 5.1.

5.2.1 Estratégia de controle

O controle escolhido foi o PI. Embora simples os atuadores possuem resposta lenta e muito atrito nas juntas devido ao processo de fabricação, eliminando a necessidade de um algoritmo mais complexo ou da parte derivativa do PID.

```
integrali[0] = integrali[0] + error*dt/1000; //calculate integral contribution
if (integrali[0]>127) integrali[0] = 127; //anti wind-up
outc = error*kpi + integrali[0]*kii;
if (outc>127) outc = 127; //limit the output to avoid damage
set_pwm1_duty(outc);
```

No bloco acima o erro integral é calculado e em seguida a saída é calculada com os ganhos K_i e K_p . E erro é calculado como a diferença entre o *setpoint* e o estado de cada junta. É feito um *anti wind-up* e limitação de *output* e em seguida o *output* é colocado no registrador correto. O processo é semelhante para as outras juntas, com pequenas variações.

5.2.2 Controle via PWMX

Para controlar as saídas PWMX é feita uma manipulação dos registradores **OVDCOND** e **OVDCONS**:

```
if(-trsh<=error && error<=trsh){ //threshold, no output
    BIT_CLEAR(OVDCOND,0);
    BIT_CLEAR(OVDCOND,1);
}
else if (error>0){
    BIT_SET(OVDCOND, 0); //positive direction
    BIT_CLEAR(OVDCOND, 1);
}
```

```

        out = (float)error/(float)max; //calculate normalized error
    }
    else if (error<0){
        BIT_SET(OVDCOND, 1); //negative direction
        BIT_CLEAR(OVDCOND, 0);
        out = abs((float)error/(float)max); //calculate normalized absolute error
    }

```

Primeiro verifica-se se o erro não está dentro da tolerância, se for o caso os *outputs* são desligados. Em seguida checa-se o sinal do erro e habilita-se apenas a saída correspondente. Também é calculado um erro normalizado para que a razão cíclica do PWM possa ser corretamente configurada.

5.2.3 Controle via CCPX

O controle via CCPX é muito semelhante ao anterior, com a exceção de que não é necessário o cálculo de um erro normalizado e o controle de direção é feito pelos bits de multiplexação das portas AND.

```

    if(-trsh<=error && error<=trsh){ //threshold, no output
        output_low(MSEL1);
        output_low(MSEL2);
    }
    else if (error>0){
        output_high(MSEL1); //positive direction
        output_low(MSEL2);
    }
    else if (error<0){
        output_high(MSEL2); //negative direction
        output_low(MSEL1);
    }

```

5.3 Rotina de comunicação

A rotina de comunicação é baseada na interrupção de chegada de dados por serial:

```

#int_rda //data received from master
void isr(){

    char data;

    data = fgetc(COM_A);
    if (data>0 && read_counter == 0) return; //wait until identifier is found

    sp[read_counter]=data; //continuously read data and allocate
    read_counter++;

    if (read_counter == 5){ //all setpoints were read
        read_counter = 0; //reset counter
        received = 1; //control may start/resume
    }
    return;
}

```

Ao chegar algum dado este dado é lido e, caso a condição de início não seja atendida (dado nulo e não foi iniciada a leitura dos *setpoints*) nada é feito. Uma vez que a condição de início ocorre são lidos todos os dados e armazenados no array correto. Após a primeira leitura um *flag* de início é ativado para que o controle possa iniciar.

5.4 Código da placa de aquisição

O código da placa de aquisição é consideravelmente menos complexo que o da de controle. Sua única função é esperar para que a placa de controle esteja pronta e começar a enviar os dados lidos. O código completo encontra-se no apêndice 9.9.

6 Resultados

Nesta seção resultados quantitativos e qualitativos são apresentados.

6.1 Performance do MOSFET

O maior indicativo da qualidade do projeto eletrônico é a performance do estágio de potência. Para verificar se os objetivos foram cumpridos o modelo desenvolvido na sub-seção 3.4 a curva de carga (e, por consequência, de saturação) foi medida com um osciloscópio, onde o resultado visual pode ser verificado na figura 6.1.

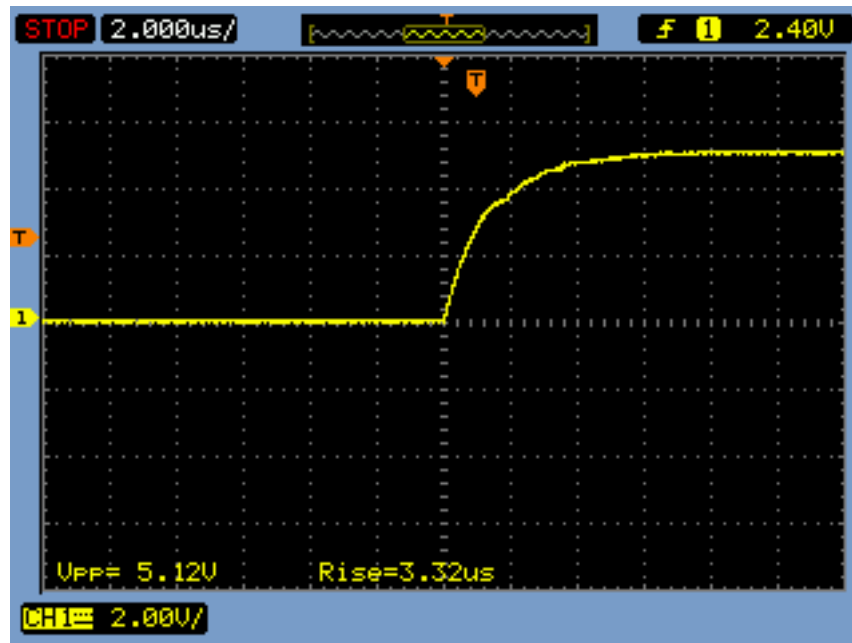


Figura 6.1: Tela do Osciloscópio exibindo a curva de carga do gate.

O sinal foi importado via CSV para o MATLAB, onde o código apresentado no apêndice 9.3 filtrou as mudanças de nível lógico do ADC do osciloscópio por meio de um filtro de média móvel de grau 10 em dupla passagem eliminou o atraso temporal (o sinal é convoluído nos dois sentidos, embora computacionalmente caro e impossível de ocorrer em tempo real, elimina atrasos temporais para análises finas). O resultado comparativo do filtro e a análise PSD (*Power Spectral Density*) comprovam a melhoria e eliminação do ruído e podem ser vistos nas figuras 6.2 e 6.3 respectivamente.

O comparativo final entre a simulação e o encontrado pode ser visto na figura 6.4 abaixo. A diferença se deve ao fato de que a capacitância de *gate-drain* varia com o aumento da tensão de gate, esta variação no entanto não foi significativa e, se o modelo mais complexo for utilizado, irá ser compensada [44].

O estágio de potência funcionou corretamente e, sendo a parte mais crítica do projeto, provou a robustez da placa como um todo.

6.2 Performance geral das placas

Esta avaliação foi puramente qualitativa, uma vez que a verificação do funcionamento de todos os sistemas pode ser feita completamente por observação. Verificou-se que todos os parâmetros funcionaram conforme esperado e dentro da performance necessária. A comunicação serial pôde, com segurança e resistência a ruídos, operar em *full duplex* corretamente. Todos os canais A/D operaram corretamente e dentro do intervalo de medição externo ($[0, 3.3]V$) e apresentaram tempo de resposta e repetição ótima. A única ressalva observada no quesito

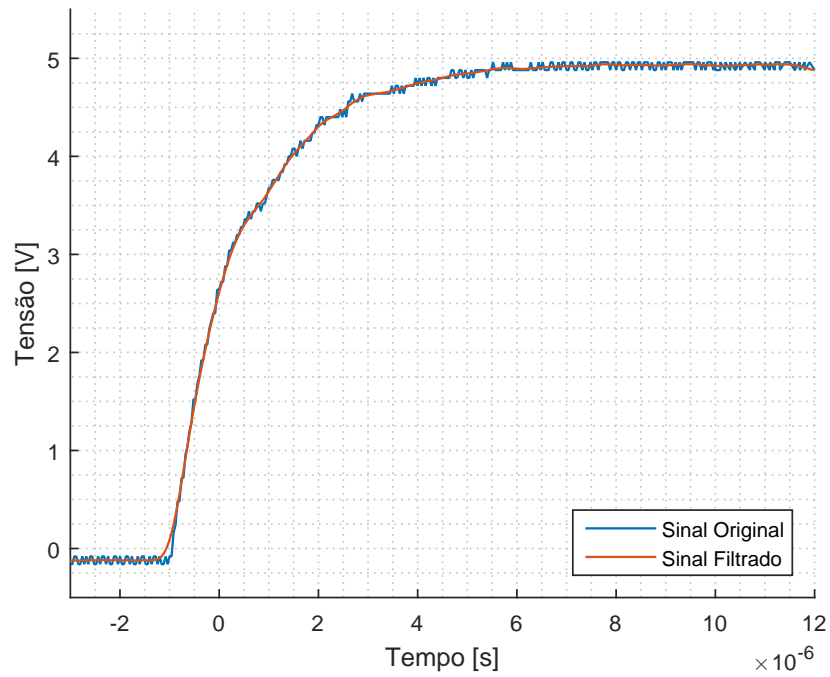


Figura 6.2: Dados importados e filtrados.

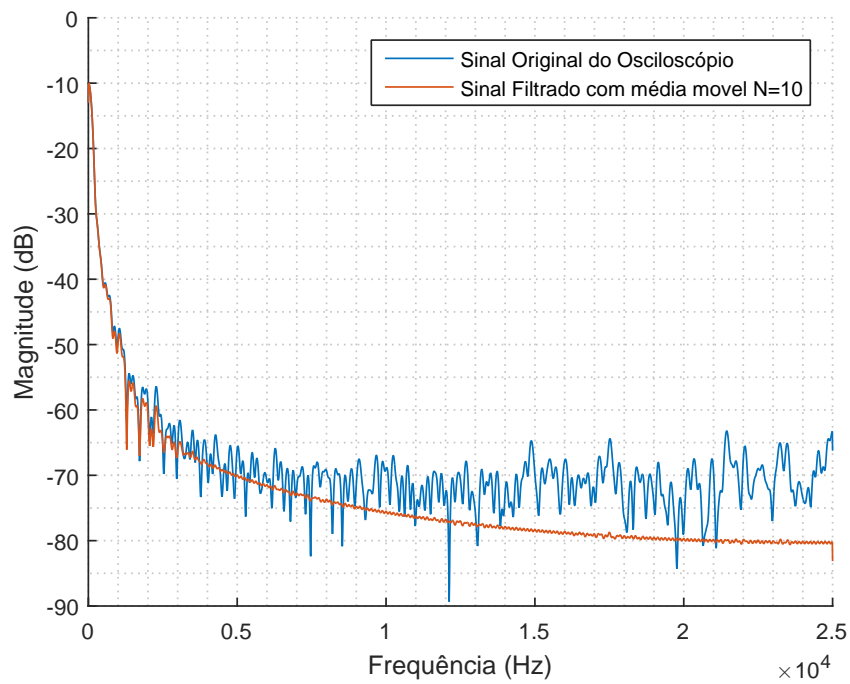


Figura 6.3: Análise PSD com o método de Welch.

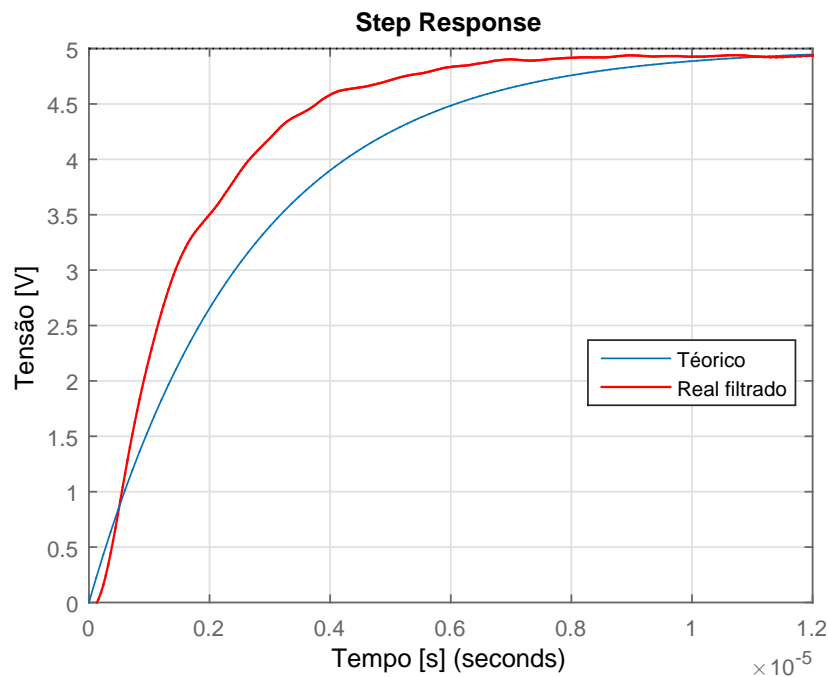


Figura 6.4: Comparativo entre a previsão teórica e as medições.

de aquisição de dados foi uma degeneração (inconsistência com valores anteriormente lidos) do sensor flexível após diversos ciclos de flexão e ocasional flexão no sentido contrário. Tal degeneração foi temporariamente solucionada com o uso de sensores novos para o andamento do projeto.

A única inconsistência elétrica observada foi uma interferência causada no conversor AD pela operação dos LEDs. Este problema foi contornado desligando-se os LEDs porém uma análise aprofundada é necessária para correção em versões futuras.

6.3 Performance da mão robótica

Novamente esta avaliação foi qualitativa. O maior problema encontrado no processo de realização da mão robótica foi a imprecisão e mau acabamento fornecido pela impressora 3D disponível. Esta situação longe do ótimo gerou atrito excessivo e não modelável para compensação (devido à maleabilidade do plástico e construção não preenchida o atrito nas juntas é não-linear com a rotação). Além disso constatou-se um problema previsto porém inevitável com o material disponível: a temperatura máxima que os fios de Flexinol atingiam superaram a temperatura de fusão do ABS, ocasionando no derretimento local. Este problema foi contornado utilizando-se fitas de Kapton (material isolante térmico) nas regiões de contato entre o ABS e os atuadores.

Outro fator que dificultou a performance das juntas foi a falta de espaço para que os fios de Flexinol tivessem comprimento o suficiente para contrair a distância necessária para atuar completamente as juntas. Este problema foi previsto porém, como uma solução ótima não foi encontrada, reserva-se para trabalhos futuros um mecanismo que aumente a amplificação de movimento dos atuadores empregados.

Feitas estas observações a mão, e o sistema como um todo, operou conforme esperado. Como o controlador é apenas PI, os atritos embora não ideais foram compensados, e a performance foi avaliada somente no volume de trabalho disponível. O tempo de assentamento do sistema dado um degrau como entrada foi medido manualmente e foram observados na ordem de 0.5 a 1s, dependendo da amplitude da entrada. Foi constatado que os tempos de retorno foram superiores em média aos de flexão, a principal hipótese para este fenômeno é a de que

é necessário o resfriamento do atuador de flexão antes que a extensão possa ter início. O que normalmente não acontece quando flexionando já que o sistema encontra-se em repouso.

Devido a geometria do modelo mecânico mesmo utilizando atuadores de 0.25mm (os maiores disponíveis para este projeto) cada dedo conseguiu controle com até apenas aproximadamente 3N de força na extremidade atuando contra o atuador. Isto era esperado devido a amplificação de movimento acarretada pela junta rotativa.

As figuras 6.5, 6.6 e 6.7 exibem o estágio inicial dos testes, assim como exemplos de configurações das juntas.

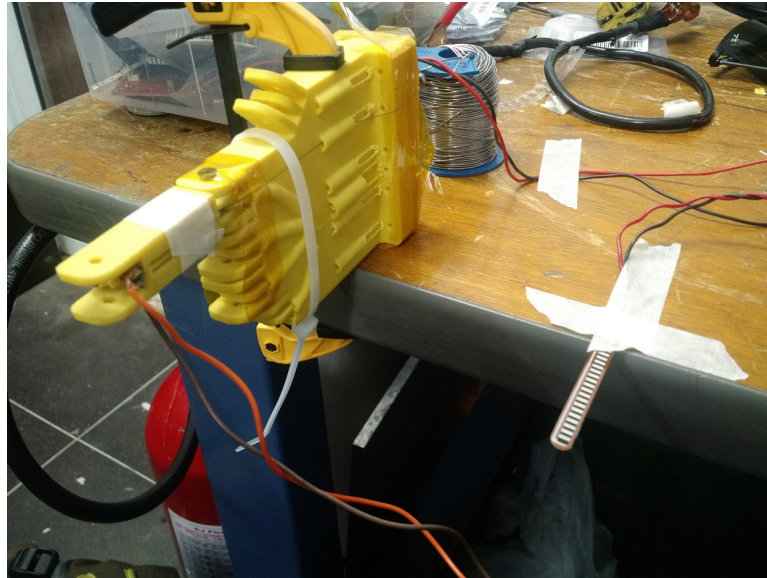


Figura 6.5: Mesa de testes inicial controlando apenas uma junta, o sensor fixo à mesa fornece o *setpoint*.



Figura 6.6: Dedo completamente estendido.

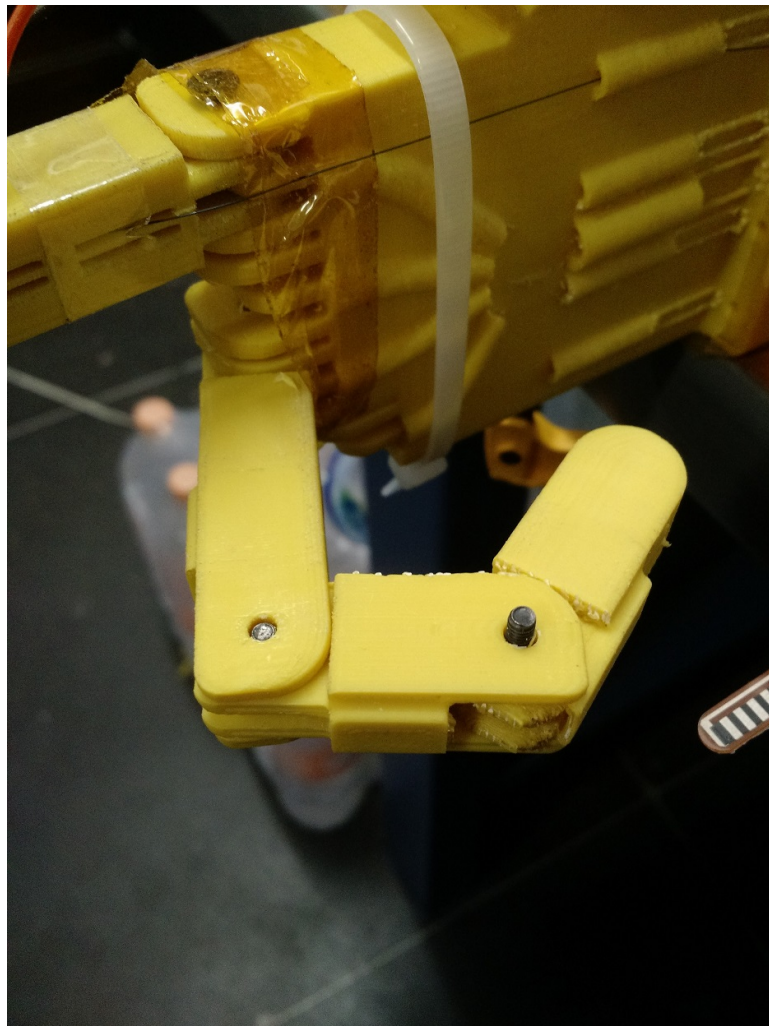


Figura 6.7: Dedo completamente contraído. Esta configuração foi obtida artificialmente uma vez que o Flexinol não forneceu contração suficiente, conforme discutido.

7 Conclusão

Devido a pouca pesquisa com ligas de efeito térmico de memória como atuadores o objetivo deste trabalho era iniciar desenvolvimento neste campo criando um sistema capaz de operar de forma robusta os atuadores propostos.

O sistema proposto inclui ampla capacidade de expansão por meio de protocolos modernos de comunicação assim como um estágio de potência capaz e confiável. O modelo mecânico inicial foi capaz de representar a possibilidade de uso do Flexinol como atuador, porém mais pesquisa e desenvolvimento são necessárias para amplificar a força e volume de trabalho. Foi constatada a necessidade de outro tipo de sensoriamento para uso a longo prazo.

Conforme visto em trabalhos anteriores mais pesquisa é necessária para viabilizar o uso do Flexinol como atuador em sistemas robóticos finos com segurança, robustez e repetibilidade. Este trabalho, no entanto, alcançou a performance esperada dos atuadores e desenvolveu conceitos e ferramentas necessárias para trabalhos futuros.

Além das recomendações já feitas, foi observado um grande potencial em implementar-se controle adaptativo nestes sistemas. Como os atuadores dependem da temperatura o sistema é completamente variante no tempo e não-linear, e o uso de controle adaptativo para modelar estes parâmetros desconhecidos é potencialmente poderoso vista a susceptibilidade do controle PI a estas mesmas variações (foi observado que a introdução súbita de uma corrente de ar perturbou o controle PI por um período excessivamente longo). Algoritmos como MRAC (*Model Reference Adaptive Control*) e \mathcal{L}_1 já obtiveram sucesso em controlar atuadores SMA em outras aplicações [45, 46] porém pesquisas nesta área são poucas e não foram aplicadas em sistemas robóticos. Recomenda-se para trabalhos futuros a modelagem do sistema com parâmetros desconhecidos e a implementação destes algoritmos (ou outros de performance similar) buscando performance estável em transientes dos parâmetros.

8 Referências

- [1] M. A. Meggiolaro. "Slides de aula de Introdução à Robótica", Rio de Janeiro, Brasil. 2015.
- [2] O. Saether. "Flexinol as Actuator for a Humanoid Finger - Possibilities and Challenges", Oslo, Noruega, Novembro 2008.
- [3] K.B. Fite, T.J. Withrow, Xiangrong Shen, K.W. Wait, J.E. Mitchell, and M. Goldfarb. "A gas-actuated anthropomorphic prosthesis for transhumeral amputees". *Robotics, IEEE Transactions on*, 24(1):159–169, Fevereiro 2008.
- [4] Kathryn J. De Laurentis and Constantinos Mavroidis. "Mechanical design of a shape memory alloy actuated prosthetic hand". *IOS Press, Technology and Health Care*, 10(2):91–106, 2002.
- [5] B. Massa, S. Roccella, M.C. Carrozza, and P. Dario. "Design and development of an underactuated prosthetic hand". In *Robotics and Automation*, 2002. Proceedings. ICRA '02. IEEE International Conference on, volume 4, pages 3374–3379 vol.4, 2002.
- [6] C. Mavroidis. "Development of advanced actuators using shape memory alloys and electrorheological fluids". *Research in Nondestructive Evaluation*, 14(1):1 - 32, Janeiro 2002.
- [7] Kenji Kaneko, Kensuke Harada, and Fumio Kanehiro. "Development of multi-fingered hand for life-size humanoid robots". In *ICRA*, pages 913–920. IEEE, 2007.
- [8] Kiyoshi Hoshino and Ichiro Kawabuchi. "Mechanism of humanoid robot arm with 7 dofs having pneumatic actuators". *IEICE Trans Fundamentals*, E89-A(11):3290–3297, 2006.
- [9] Naoki Fukaya, Shigeki Toyama, Tamim Asfour, and Rüdiger Dillmann. "Design of the tuat-/karlsruhe humanoid hand", Maio 2000.
- [10] Shadow Robot Company. "Design of a dextrous hand for advanced clawar applications" [www.shadowrobot.com/downloads/dextrous hand final.pdf](http://www.shadowrobot.com/downloads/dextrous%20hand%20final.pdf), Outubro 2015.
- [11] A. Kargov, T. Asfour, C. Pylatiuk, R. Oberle, H. Klosek, S. Schulz, K. Regenstein, G. Bretthauer, and R. Dillmann. "Development of an anthropomorphic hand for a mobile assistive robot". *Rehabilitation Robotics*, 2005. ICORR 2005. *9th International Conference* 182–186, Junho e Julho 2005.
- [12] H. Kawasaki, T. Komatsu, and K. Uchiyama. "Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand ii". *Mechatronics, IEEE/ASME Transactions on*, 7(3):296–303, Setembro 2002.
- [13] Bertrand Tondou, Serge Ippolito, Jérémie Guiochet, and A. Daidie. "A seven-degrees-of-freedom robot arm driven by pneumatic artificial muscles for humanoid robots". *I. J. Robotic Res*, 24(4):257–274, 2005.
- [14] J. Ramos, A. Wang, W. Ubellacker, J. Mayo, and S. Kim, "A Balance Feedback Interface for Whole-Body Teleoperation of a Humanoid Robot and Implementation in the Hermes System", in *Humanoid Robots*, 2015 *15th IEEE-RAS International Conference on*. IEEE, 2015.
- [15] Boston Dynamics. "Atlas - The Agile Anthropomorphic Robot". www.bostondynamics.com 2015
- [16] Asfour, T.; Regenstein, K.; Azad, P.; Schroder, J.; Bierbaum, A.; Vahrenkamp, N. Dillmann, R. "ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control". *Humanoid Robots*, 2006 *6th IEEE-RAS International Conference on*
- [17] Diftler, M.A.; Mehling, J.S.; Abdallah, M.E.; Radford, N.A.; Bridgwater, L.B.; Sanders, A.M.; Askew, R.S.; Linn, D.M.; Yamokoski, J.D.; Permenter, F.A.; Hargrave, B.K.; Piatt, R.; Savely, R.T.; Ambrose, R.O. "Robonaut 2 - The first humanoid robot in space" *Robotics and Automation (ICRA)*, 2011 *IEEE International Conference on*.
- [18] Steeper USA. "bebionic Medium / Large Technical Manual". http://bebionic.com/downloads/product_literature 2015.
- [19] Fukaya, N. ; Asfour, T. ; Dillmann, R. ; Toyama, S. "Development of a five-finger dexterous hand without feedback control: The TUAT/Karlsruhe humanoid hand" *Intelligent Robots and Systems (IROS)*, 2013 *IEEE/RSJ International Conference on*

- [20] T. Dutta and T. Chau. *A feasibility study of flexinol as the primary actuator in a prosthetic hand. In IEEE CCECE 2003, Canadian Conference on Electrical and Computer Engineering*, volume 3, p. 1449–1452 vol.3, 2003.
- [21] Reband, K.; Dees, M.; Conrad, J.M., "Robot muscles in a subzero temperature environment," in *IEEE SoutheastCon 2010 (SoutheastCon), Proceedings of the*, vol., no., pp.73-76, 18-21 March 2010
- [22] F. Gori, D. Carnevale, A. Doro Altan, S. Nicosia, and Pennestrí. "A new hysteretic behavior in electrical resistivity of flexinol shape memory alloys versus temperature". *International Journal of Thermophysics*, 27(3):866–879, May 2006.
- [23] H.H. Selden, B.; Kyu-Jin Cho; Asada. "Segmented binary control of shape memory alloy actuator systems using the peltier effect". *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, 5:4931–4936 Vol.5, 26 April-1 May 2004.
- [24] M G Faulkner T B Wolfe (née Bunton) and J Wolfaardt. "Development of a shape memory alloy actuator for a robotic eye prosthesis". *Smart Materials and Structures*, 14(4):759–768, 2005.
- [25] W M Ostachowicz A J Zak, M P Cartmell and M Wiercigroch. "One-dimensional shape memory alloy models for use with reinforced composite structures". *Smart Materials and Structures*, 12(3): 338–346, 2003.
- [26] Pavel L. Potapov and Edson P. Da Silva. "Time response of shape memory alloy actuators". *Journal of Intelligent Material Systems and Structures*, 11(2):125–134, 2000.
- [27] Michael J. Mosley and Constantinos Mavroidis. "Design and dynamics of a shape memory alloy wire bundle actuator", 1999.
- [28] V. Grant, D.; Hayward. "Design of shape memory alloy actuator with high strain and variable structure control". *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, 3:2305–2312 vol.3, 1995.
- [29] Falange <https://pt.wikipedia.org/wiki/Falange> 2015
- [30] G. ElKoura, K. Singh, "Handrix: Animating the Human Hand ", *Eurographics/SIGGRAPH Symposium on Computer Animation (2003)*
- [31] Ramos, João Luiz Almeida de Souza; Meggiolaro, Marco Antônio. "Controle de Torque de um Exoesqueleto Atuado por Músculos Pneumáticos Artificiais Utilizando Sinais Eletromiográficos". *Rio de Janeiro, 2013. 149p. Dissertação de Mestrado - Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro*.
- [32] Meggiolaro, Marco Antonio. "Riobotz Comboto Tutorial" *Rio de Janeiro, March 2009, version 2.0*.
- [33] Dynalloy, Inc. "Flexinol Actuator Wire Heat Curves". 2015
- [34] Dynalloy, Inc. "Flexinol Actuator Wire Technical and Design Data".
- [35] Dynalloy, Inc. "Technical Characteristics of Flexinol".
- [36] Spectra Symbol "FS Flex Sensor Manufacturer Datasheet".
- [37] Toshiba. "TPH4R008NH Manufacturer Data sheet".
- [38] Microchip. "PIC18F2331/2431/4331/4431 Manufacturer Data sheet".
- [39] Texas Instruments. "SN74LVC2G08DCTR Manufacturer Data sheet".
- [40] Maxim Integrated. "MAX487 - MAX491 Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers Manufacturer Data sheet".
- [41] Texas Instruments. "LM1575/LM2575/LM2575HV 1A Step-Down Voltage Regulator Manufacturer Data sheet".
- [42] Texas Instruments. "LM3940 1A Low-Dropout Regulator for 5V to 3.3V Conversion Manufacturer Data sheet".
- [43] EOS. "Basic Design Rules for Additive Manufacturing (AM)"2015.

- [44] Vishay Siliconix. "Power MOSFET Basics: Understanding Gate Charge and Using it to Assess Switching Performance" Application Note 608.
- [45] S. Kannan, C. Giraud-Audine, E. Patoor. "Control of shape memory alloy (SMA) actuator using series-parallel model reference adaptive control (MRAC)". *SMASIS 2009, 2nd ASME Conference on Smart Materials, Adaptive Structures and Intelligent Systems, Proceedings, Vol. 1: Active Materials, Mechanics and Behavior. Modeling, Simulation and Control, Ox-nard, CA, US, Sep 21-23, 2009*.
- [46] M. Bongani, G. D. Bucknera. " \mathcal{L}_1 adaptive control of a shape memory alloy actuated flexible beam" *Systems Science & Control Engineering*, 3:1, 460-471.

9 Apêndices

9.1 Código em MATLAB para análise do sensor

```

%% Initialize
%
clc
clear all
close all
%% Acquire data
%
M = csvread('senread.csv'); %read from csv

rows = size(M,1); %read number of rows

deg = M(1:rows,1); %associate data
res = M(1:rows,2);

%% Fit to Methods
%

[modele1, ~] = fit( deg, res, 'exp1' ); %first-order exponential
[modele2, ~] = fit( deg, res, 'exp2' ); %second order exponential
[modelp2, ~] = fit( deg, res, 'poly2' ); %second order polynomial
[modelp3, ~] = fit( deg, res, 'poly3' ); %third order polynomial
[modelp4, ~] = fit( deg, res, 'poly4' ); %fourth order polynomial
P = polyfit(deg,res,1); %linear
modell = P(1)*deg+P(2);
%% Show results
%

modele1
modele2
modelp2
modelp3
modelp4

%% Plot all Methods
%

figure
hold on
plot(deg, res, 'r*' );
plot(modele1, 'b');
plot(modele2, 'g');
plot(modelp2, 'm');
plot(modelp3, 'b--');
plot(modelp4, 'g--');
plot(deg,modell, 'black', 'LineWidth', 1.5);
hold off
title('Métodos de fitting');
xlabel('flexão (graus)');
ylabel('resistência (ohms)');
legend('dados', 'exp_1', 'exp_2', 'poly_2', 'poly_3', 'poly_4', 'linear', ...
      'Location', 'NorthWest');

%% Plot best
%

```

```

figure
plot(deg, res, 'r*' );
hold on
plot(modele2, 'b', 'predobs' );
hold off
xlabel('flexão (graus)');
ylabel('resistência (ohms)');
title('Aproximação por exponencial de grau 2');
legend('dados', 'exp_2', 'conf_{bounds}');

```

9.2 Código em MATLAB para otimização da resistência divisora

```

%% Otimização de valor de resistência de divisão
%

clc
clear all
close all
format long
%% Parâmetros e variáveis
%

bits = 8;
vs = 5;
max_r = 81*10^3;
min_r = 25*10^3;
i = 0;
max_s = 0;

%% Função a ser otimizada
%

r = 0:max_r;
v_grad = vs*((max_r./(r+max_r)) - (min_r./(r+min_r)));

%% Plotar resultado
%
figure
hold on
grid on
plot(r, v_grad, 'LineWidth', 1.5);
xlabel('Resistência divisora [\Omega]');
ylabel('Sensibilidade [V]');
hold off

%% Busca Linear
%

for i=1:1:(max_r-1)
    if(v_grad(i)>=max_s)
        max_s = v_grad(i);
    else
        break;
    end
end

max_s

```

```

r_opt = (r(i-1))
max_v = vs*(max_r./(r_opt+max_r))
min_v = vs*(min_r./(r_opt+min_r))
resmin = 5/2^bits*10^3
resmax = 3.3/2^bits*10^3

```

9.3 Código em MATLAB para frequência máxima de chaveamento

```

%% Switching frequency calculator
%

clc
clear all
close all

%% Declare parameters and TF
%

s = tf('s');
R = 220;          %%gate resistor
C = 12*10^-9;    %%gate-drain charge
H = 5/(1+s*R*C); %%RC TF

%% Plot the Step Response
%

figure
hold on
step(H)
grid on
title('Curva de carga do gate')
xlabel('tempo [s]')
ylabel('tensão [V]')

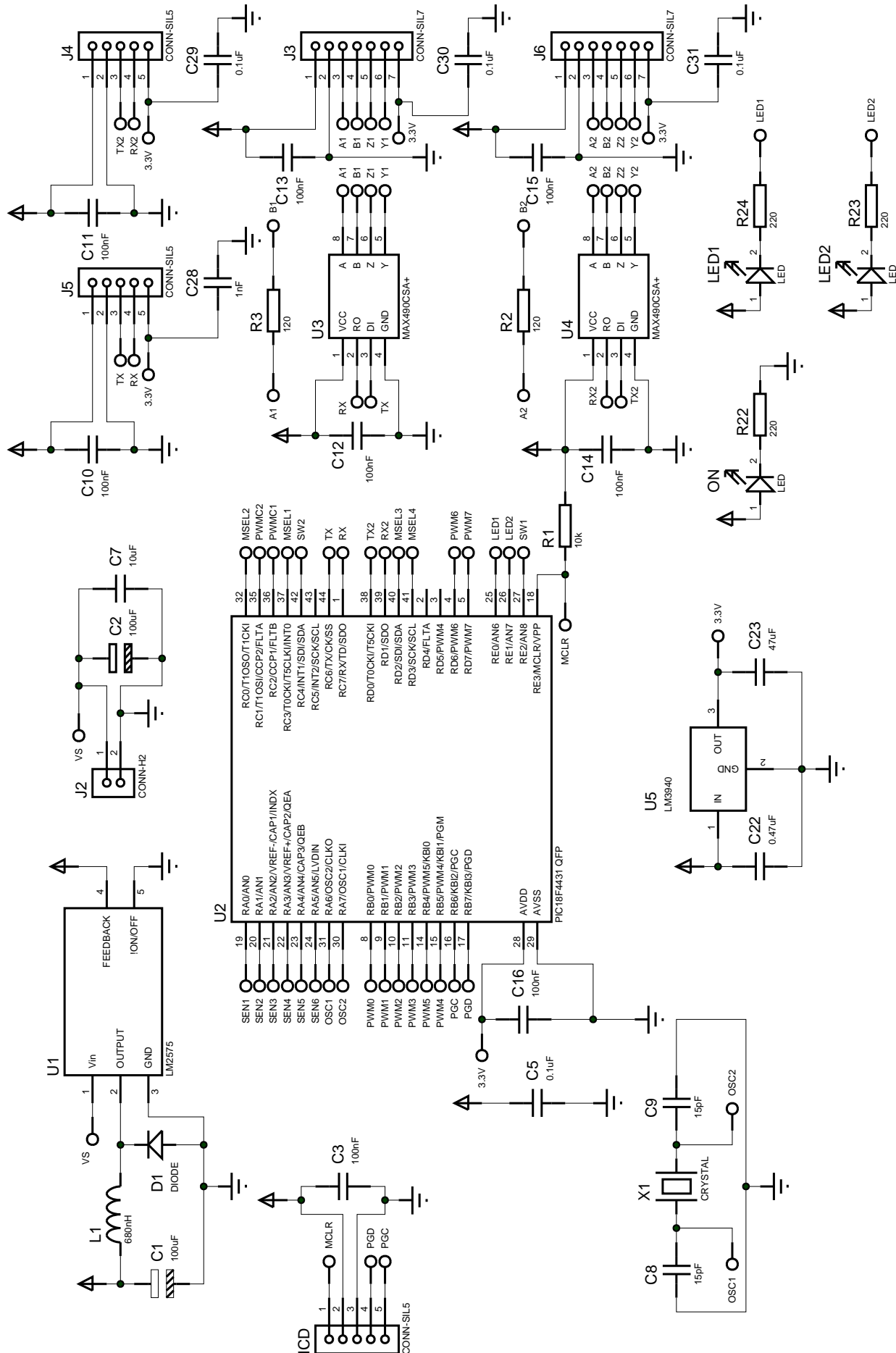
%% Get Data from system
%

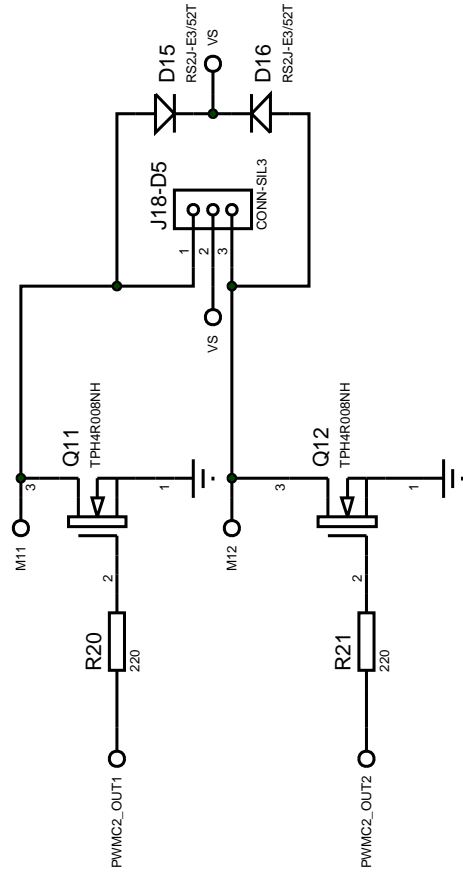
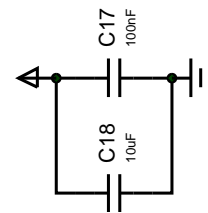
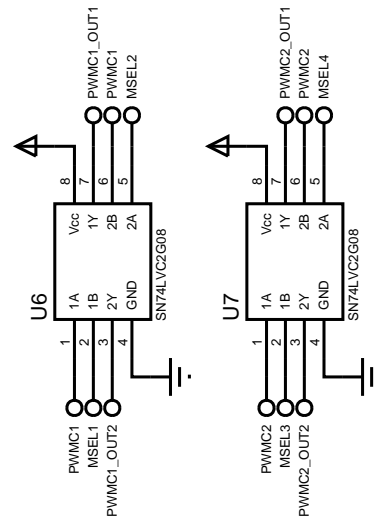
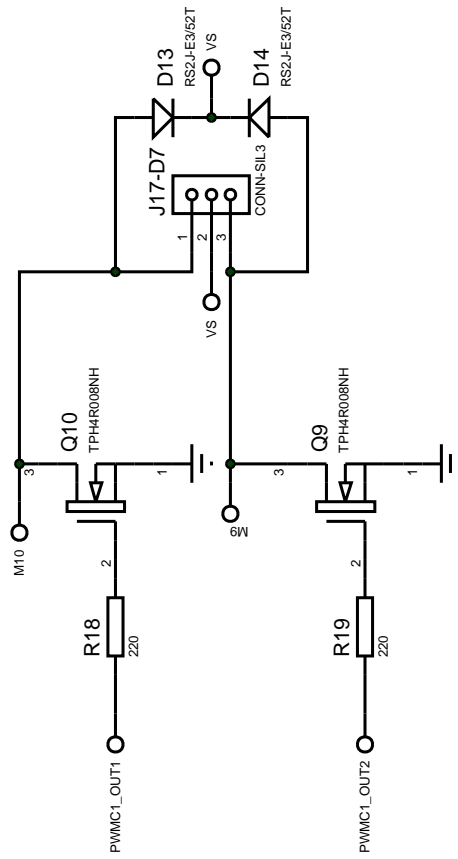
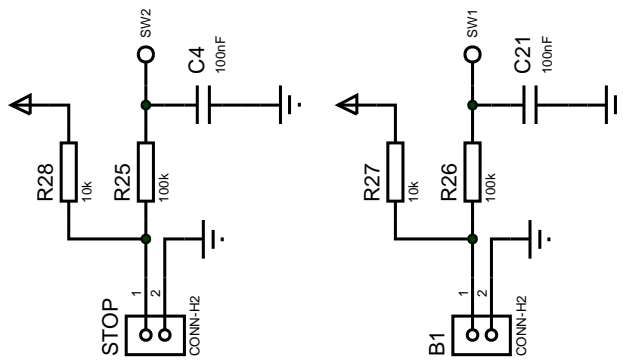
S = stepinfo(H)

max_f = 1/(2*S.RiseTime)

```

9.4 Esquemático completo da placa de controle





9.5 Esquemático completo da placa de aquisição

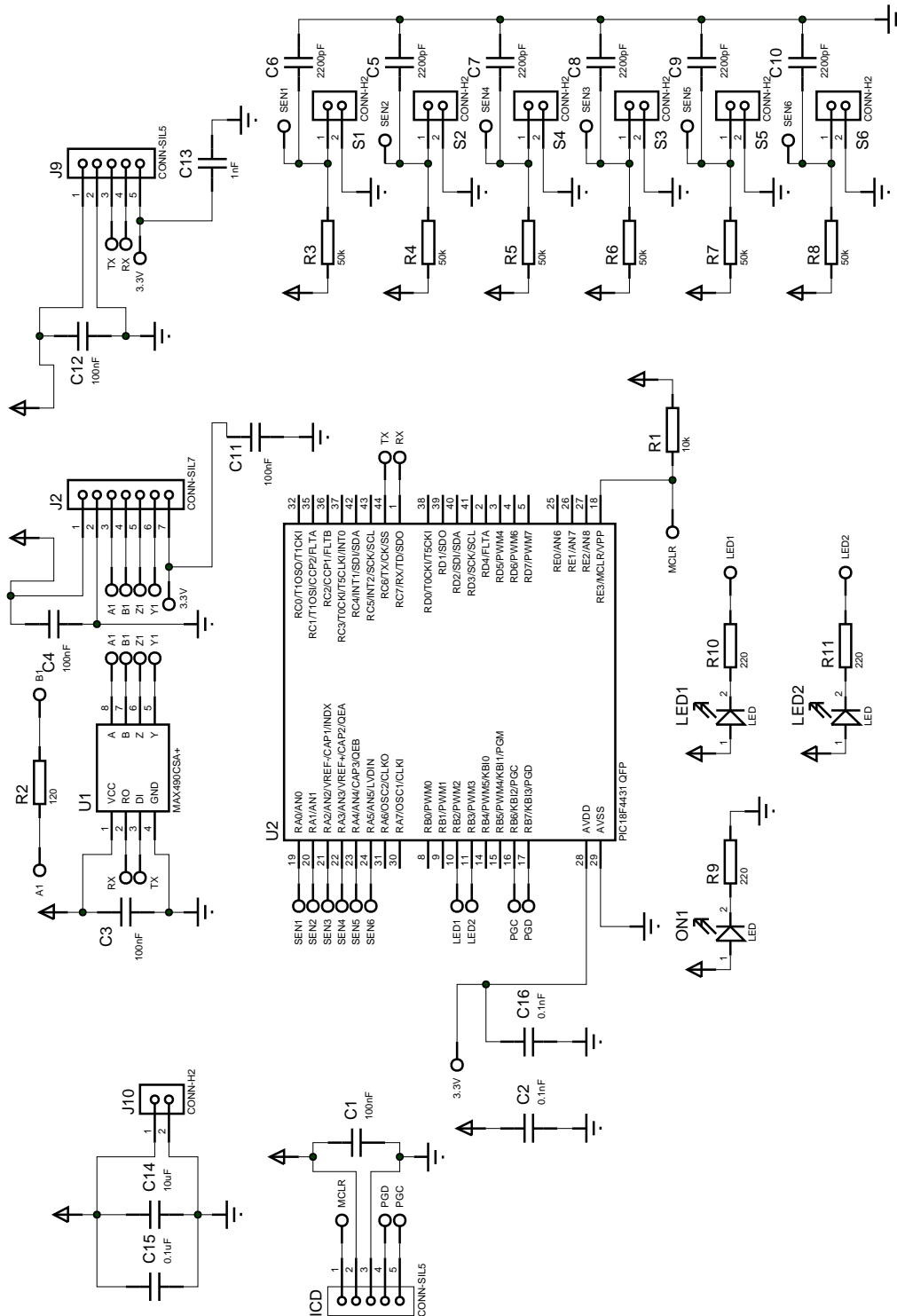


Figura 9.1: Esquemático completo da placa de aquisição.

9.6 Layout completo da placa de controle

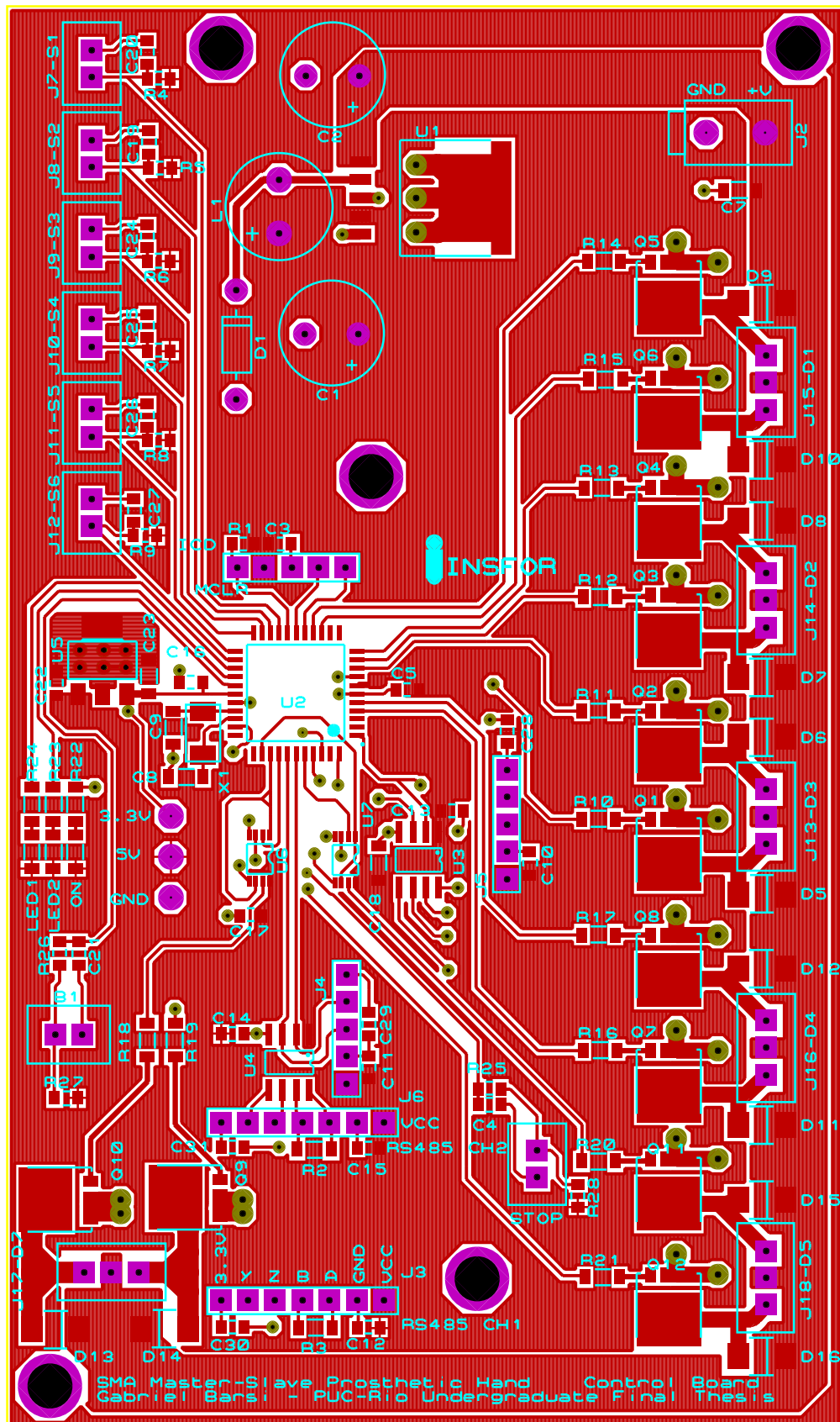


Figura 9.2: Layout completo da placa de controle, *output* do ARES.

9.7 Layout completo da placa de aquisição

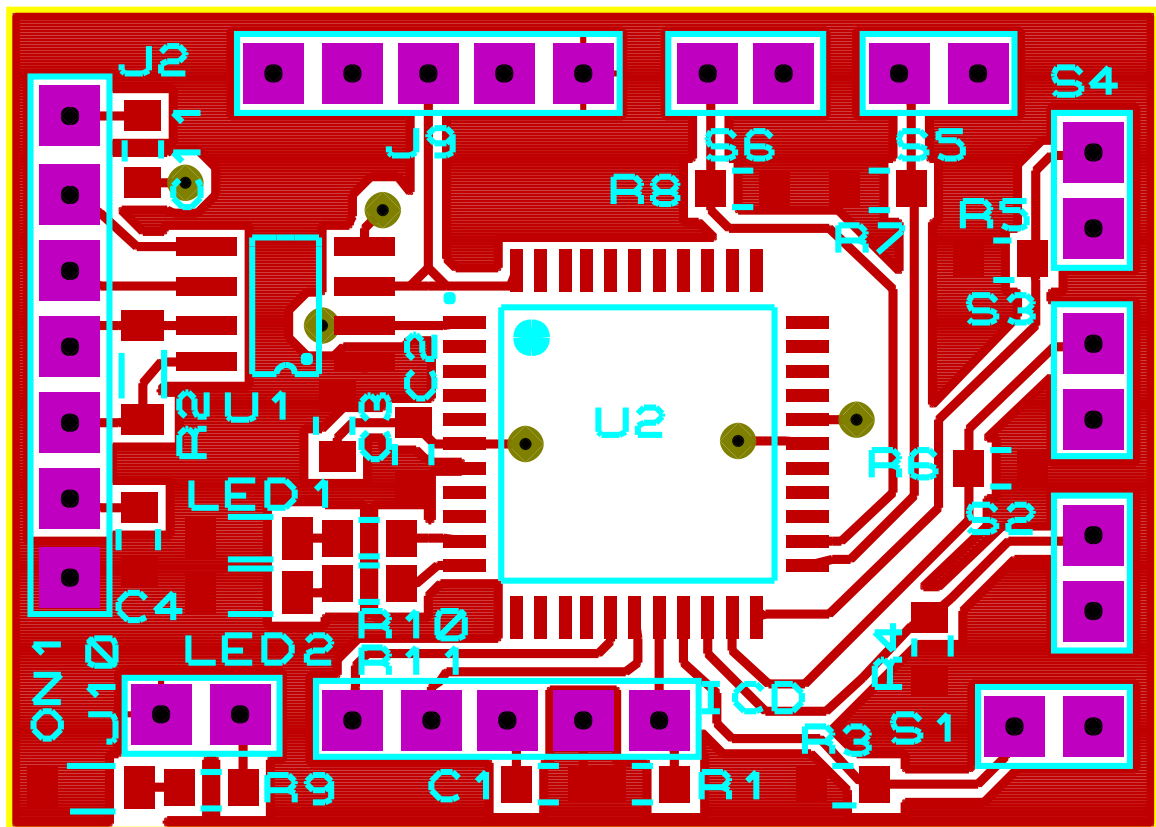


Figura 9.3: Layout da camada superior da placa que aquisição.

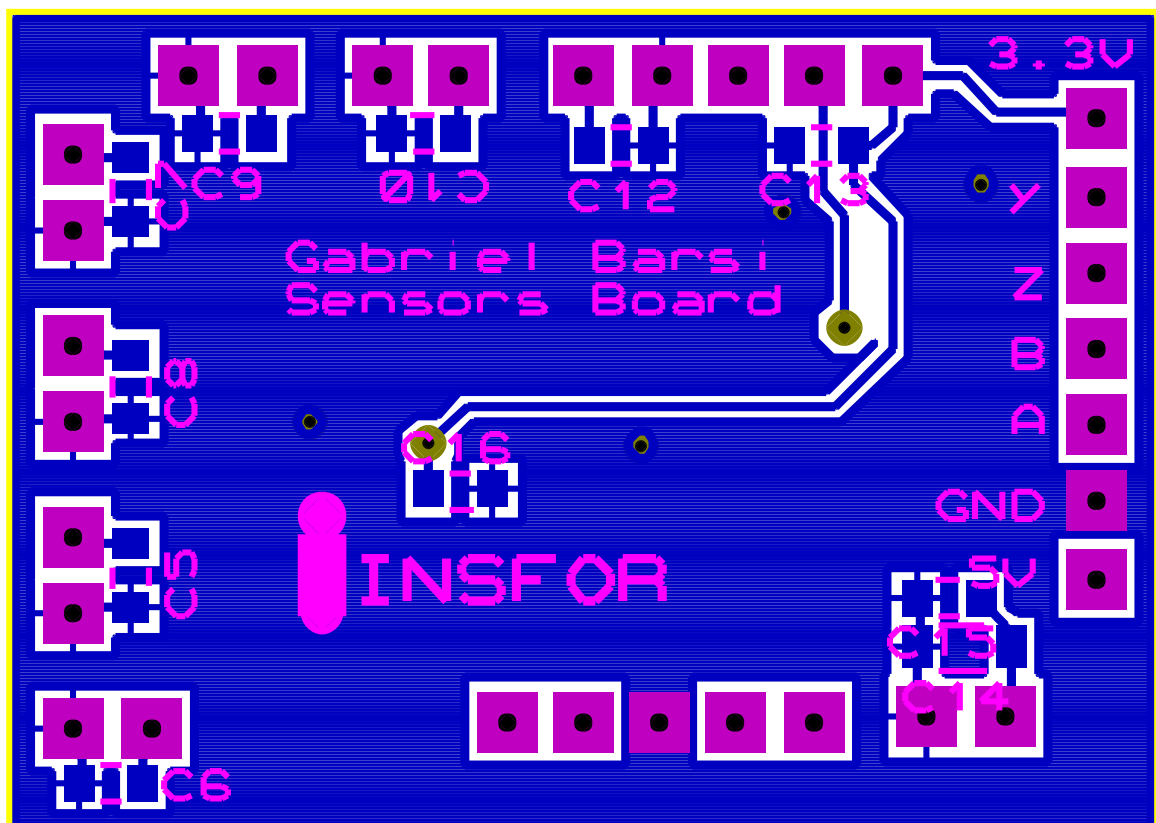


Figura 9.4: Layout da camada inferior da placa que aquisição.

9.8 Código da Placa de Controle

```

/*
Gabriel Barsi Haberfeld - Undergraduate Final Thesis
Control and Automation Engineering 2015.1
Control Board Code

CCS PCWHD r.4.057
*/

//Setup device and addresses
#include <18F4431.h> //set the device
#define adc=8 //use 8-bit ADC

#fuses HS //high speed oscillator
#use delay(clock=20000000) //20MHz oscillator
#use rs232(baud=9600, xmit=PIN_C6,rcv=PIN_C7, STREAM=COM_A) //setup serial port A
#use rs232(baud=9600, xmit=PIN_D0,rcv=PIN_D1, STREAM=COM_B, DISABLE_INTS) //setup serial port B

#define POWER_PWM_PERIOD 3999 //PWMX period counter, f_pwm = 1250Hz

#define LED1 PIN_E0 //assign pin names
#define LED2 PIN_E1
#define MSEL1 PIN_C3
#define MSEL2 PIN_C0
#define MSEL3 PIN_D2
#define MSEL4 PIN_D3

#byte OVDCOND = 0xF6B //assign OVD registers addresses
#byte OVDCONS = 0xF6A

//Initialize global variables
unsigned int r = 0, max=255,min=150, received = 0, trim = 3; //GP variables
float out;
signed int error;
unsigned int sp[6] = {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF}, bsl_t=0,bsl_r=0, outc; //setpoints and
baselines
int trsh = 1; //threshold
float kpf = 50, kif = 100, integralf[4] = {0,0,0,0}; //PI float gains and integral sums
unsigned int kpi = 10, kii = 20, integrali[2] = {0,0}; //PI int gains and integral sums
int dt = 10; //aprox. desired control period in ms
int read_counter = 0;

#int_rda //data received from master
void isr(){

char data;

data = fgetc(COM_A);
if (data>0 && !read_counter) return; //wait until identifier is found

sp[read_counter]=data; //continuously read data and allocate
read_counter++;

if (read_counter == 5){ //all setpoints were read
read_counter = 0; //reset counter
received = 1; //control may start/resume
}
return;
}

void control_finger(int idx){ //control one joint

//little finger = PWM0/1, idx 0
//ring finger = PWM2/3, idx 1
//middle finger = PWM4/5, idx 2
//index finger = PWM6/7, idx 3
//thumb = CCP1, idx 4
//Prox. Phalanges = CCP2, idx 5;

if (!received) idx = -1; //no data sent yet, keep actuators off

```



```

switch (idx){

    case 0: set_adc_channel(0); //little finger
            delay_us(100); //wait for adc settle time
            r = read_adc();
            error = sp[idx] - r - trim; //calculate error with std correction
            if(-trsh<=error && error<=trsh){ //threshold, no output
                BIT_CLEAR(OVDCOND,0);
                BIT_CLEAR(OVDCOND,1);
            }
            else if (error>0){
                BIT_SET(OVDCOND, 0); //positive direction
                BIT_CLEAR(OVDCOND, 1);
                out = (float)error/(float)max; //calculate normalized error
            }
            else if (error<0){
                BIT_SET(OVDCOND, 1); //negative direction
                BIT_CLEAR(OVDCOND, 0);
                out = abs((float)error/(float)max); //calculate normalized absolute error
            }

            integralf[idx] = integralf[idx] + (float)error/(float)max*dt/1000; //calculate float
            integral contribution
            if (integralf[idx]>0.5) integralf[idx] = 0.5; //anti wind-up
            out = out*kpf + integralf[idx]*kif;
            if (out >0.5) out = 0.5; //limit the output to avoid actuator damage
            set_power_pwm0_duty((int16)(((POWER_PWM_PERIOD+1) *4) * out)); //output correct DC
            fputc(0x00,COM_B); //output state to PC
            fputc(r,COM_B);
            break;

    case 1: set_adc_channel(1); //ring finger
            delay_us(100); //wait for adc settle time
            r = read_adc();
            error = sp[idx] - r - trim; //calculate error with std correction
            if(-trsh<=error && error<=trsh){ //threshold, no output
                BIT_CLEAR(OVDCOND,2);
                BIT_CLEAR(OVDCOND,3);
            }
            else if (error>0){
                BIT_SET(OVDCOND, 2); //positive direction
                BIT_CLEAR(OVDCOND, 3);
                out = (float)error/(float)max; //calculate normalized error
            }
            else if (error<0){
                BIT_SET(OVDCOND, 3); //negative direction
                BIT_CLEAR(OVDCOND, 2);
                out = abs((float)error/(float)max); //calculate normalized absolute error
            }

            integralf[idx] = integralf[idx] + (float)error/(float)max*dt/1000; //calculate float
            integral contribution
            if (integralf[idx]>0.5) integralf[idx] = 0.5; //anti wind-up
            out = out*kpf + integralf[idx]*kif;
            if (out >0.5) out = 0.5; //limit the output to avoid actuator damage
            set_power_pwm2_duty((int16)(((POWER_PWM_PERIOD+1) *4) * out)); //output correct DC
            fputc(0x01,COM_B);
            fputc(r,COM_B);
            break;

    case 2: set_adc_channel(2); //middle finger
            delay_us(100); //wait for adc settle time
            r = read_adc();
            error = sp[idx] - r - trim; //calculate error with std correction
            if(-trsh<=error && error<=trsh){ //threshold, no output
                BIT_CLEAR(OVDCOND,4);
                BIT_CLEAR(OVDCOND,5);
            }
            else if (error>0){
                BIT_SET(OVDCOND, 4); //positive direction
                BIT_CLEAR(OVDCOND, 5);
                out = (float)error/(float)max; //calculate normalized error
            }
            else if (error<0){
                BIT_SET(OVDCOND, 5); //negative direction

```

```

        BIT_CLEAR(OVDCOND, 4);
        out = abs((float)error/(float)max); //calculate normalized absolute error
    }

    integralf[idx] = integralf[idx] + (float)error/(float)max*dt/1000; //calculate float
    integral contribution
    if (integralf[idx]>0.5) integralf[idx] = 0.5; //anti wind-up
    out = out*kpf + integralf[idx]*kif;
    if (out >0.5) out = 0.5; //limit the output to avoid actuator damage
    set_power_pwm4_duty((int16)(((POWER_PWM_PERIOD+1) *4) * out)); //output correct DC
    fputc(0x02,COM_B);
    fputc(r,COM_B);
    break;

case 3:set_adc_channel(3); //index finger
    delay_us(100); //wait for adc settle time
    r = read_adc();
    error = sp[idx] - r - trim; //calculate error with std correction
    if(-trsh<=error && error<=trsh){ //threshold, no output
        BIT_CLEAR(OVDCOND,6);
        BIT_CLEAR(OVDCOND,7);
    }
    else if (error>0){
        BIT_SET(OVDCOND, 6); //positive direction
        BIT_CLEAR(OVDCOND, 7);
        out = (float)error/(float)max; //calculate normalized error
    }
    else if (error<0){
        BIT_SET(OVDCOND, 7); //negative direction
        BIT_CLEAR(OVDCOND, 6);
        out = abs((float)error/(float)max); //calculate normalized absolute error
    }

    integralf[idx] = integralf[idx] + (float)error/(float)max*dt/1000; //calculate float
    integral contribution
    if (integralf[idx]>0.5) integralf[idx] = 0.5; //anti wind-up
    out = out*kpf + integralf[idx]*kif;
    if (out >0.5) out = 0.5; //limit the output to avoid actuator damage
    set_power_pwm6_duty((int16)(((POWER_PWM_PERIOD+1) *4) * out)); //output correct DC
    fputc(0x03,COM_B);
    fputc(r,COM_B);
    break;

case 4:set_adc_channel(4); //thumb
    delay_us(100); //wait for adc settle time
    r = read_adc();
    error = sp[idx] - r - trim; //calculate error with std correction
    if(-trsh<=error && error<=trsh){ //threshold, no output
        output_low(MSEL1);
        output_low(MSEL2);
    }
    else if (error>0){
        output_high(MSEL1); //positive direction
        output_low(MSEL2);
    }
    else if (error<0){
        output_high(MSEL2); //negative direction
        output_low(MSEL1);
    }

    integrali[0] = integrali[0] + error*dt/1000; //calculate integral contribution
    if (integrali[0]>127) integrali[0] = 127; //anti wind-up
    outc = error*kpi + integrali[0]*kii;
    if (outc>127) outc = 127; //limit the output to avoid damage
    set_pwm1_duty(outc);
    fputc(0x04,COM_B);
    fputc(r,COM_B);
    break;

case 5:set_adc_channel(5); //prox. phalanges
    delay_us(100); //wait for adc settle time
    r = read_adc();
    error = sp[idx] - r - trim; //calculate error with std correction
    if(-trsh<=error && error<=trsh){ //threshold, no output
        output_low(MSEL1);

```

```

        output_low(MSEL2);
    }
    else if (error>0){
        output_high(MSEL1); //positive direction
        output_low(MSEL2);
    }
    else if (error<0){
        output_high(MSEL2); //negative direction
        output_low(MSEL1);
    }
    integrali[1] = integrali[1] + error*dt/1000; //calculate integral contribution
    outc = abs(error)*kpi + integrali[1]*kii;
    if (integrali[0]>127) integrali[0] = 127; //anti wind-up
    //if (outc>127) outc = 127; //not needed since actuators are paralelized
    set_pwm2_duty(outc);
    fputc(0x05,COM_B);
    fputc(r,COM_B);
    break;

default: delay_ms(5000); //conditions not met, wait a period and try again
    return;

return;
}

}

//Main stack
void main ()
{
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_64); //setup TIMERO, PRSCLR = 64
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_8); //setup TIMET1, PRSCLR = 8
    setup_timer_2(T2_DIV_BY_16, 255, 1); //setup TIMER2, PRSCLR = 16, ovf=8bits
    enable_interrupts(GLOBAL); //enable interrupts
    enable_interrupts(INT_RDA); //enable serial receive interrupts
    ext_int_edge( H_TO_L ); //enable external interrupts on falling edge

    setup_adc_ports(ALL_ANALOG|VSS_VDD); //setup ADC with external reference
    setup_adc(ADC_CLOCK_INTERNAL); //natural ADC clock
    set_adc_channel(0); //initial channel

    setup_power_pwm_pins(PWM_BOTH_ON,PWM_BOTH_ON,PWM_BOTH_ON,PWM_BOTH_ON); //setup PMWX, all ON
    setup_power_pwm(PWM_FREE_RUN, 1, 0, POWER_PWM_PERIOD, 0, 1,0); //setup modes

    setup_ccp1(CCP_PWM); //CCPX as PWM, f_pwm = 1250Hz
    setup_ccp2(CCP_PWM);

    output_low(MSEL1); //initialize at no output for safety
    output_low(MSEL2);
    output_low(MSEL3);
    output_low(MSEL4);

    set_pwm1_duty (0x00); //initialize at no output for safety, 100% DC = 0xFF
    set_pwm2_duty (0x00);

    OVDCONS = 0x00; //all PWMX overridden
    OVDCOND = 0x00; //all PWMX OFF

    set_adc_channel(0); //get max baseline reading from sensors, must eval each one
    bsl_r = read_adc(); //since read_adc(); doesnt allow a variable as parameter
    if (bsl_r>bsl_t) bsl_r=bsl_t;
    set_adc_channel(1);
    bsl_r = read_adc();
    if (bsl_r>bsl_t) bsl_r=bsl_t;
    set_adc_channel(2);
    bsl_r = read_adc();
    if (bsl_r>bsl_t) bsl_r=bsl_t;
    set_adc_channel(3);
    bsl_r = read_adc();

```

```

if (bsl_r>bsl_t) bsl_r=bsl_t;
set_adc_channel(4);
bsl_r = read_adc();
if (bsl_r>bsl_t) bsl_r=bsl_t;
set_adc_channel(5);
bsl_r = read_adc();
if (bsl_r>bsl_t) bsl_r=bsl_t;
set_adc_channel(6);
bsl_r = read_adc();
if (bsl_r>bsl_t) bsl_r=bsl_t;

delay_ms(1000);      //wait for master
fputc(0x01,COM_A);    //tell master that control is ready

while (true)
{
delay_ms(dt);        //wait control period

control_finger(0);    //control all fingers
control_finger(1);
control_finger(2);
control_finger(3);
control_finger(4);
control_finger(5);

fputc(0x06,COM_B);    //output all setpoints to PC
fputc(sp[0],COM_B);
fputc(sp[1],COM_B);
fputc(sp[2],COM_B);
fputc(sp[3],COM_B);
fputc(sp[4],COM_B);
fputc(sp[5],COM_B);

}

}

```

9.9 Código da Placa de Aquisição

```

/*
Gabriel Barsi Haberfeld - Undergraduate Final Thesis
Control and Automation Engineering 2015.1
Sensors Board Code

CCS PCWHD r.4.057
*/

//Setup device and addresses
#include <18F4431.h>      //set the device
#define adc=8            //use 8-bit ADC

#define delay(clock=4000000) //4MHz oscillator
#define rs232(baud=9600, xmit=PIN_C6,rcv=PIN_C7, STREAM=COM_A) //setup serial port A

int received = 0, r;

#define int_rda //data received from control
void isr(){

char data;

data = fgetc(COM_A);

if (data == 0x01) received = 1; //control is ready

return;
}

//Main stack
void main ()
{

```

```
enable_interrupts(GLOBAL);           //enable interrupts
enable_interrupts(INT_RDA);          //enable serial receive interrupts

setup_adc_ports(ALL_ANALOG|VSS_VDD); //setup ADC with external reference
setup_adc(ADC_CLOCK_INTERNAL);       //natural ADC clock
set_adc_channel(0);                   //initial channel

while (!received){                   //wait until system is ready
}

while (true)
{
    delay_ms(10);                     //wait 10ms before next batch
    fputc(0x00, COM_A);               //tell control that the batch is starting
    delay_us(100);                    //wait for ADC settling time
    set_adc_channel(0);               //set channel
    r = read_adc();                   //read value
    fputc(r, COM_A);                  //send to control
    delay_us(100);                    //repeat for subsequent channels
    set_adc_channel(1);
    r = read_adc();
    fputc(r, COM_A);
    delay_us(100);
    set_adc_channel(2);
    r = read_adc();
    fputc(r, COM_A);
    delay_us(100);
    set_adc_channel(3);
    r = read_adc();
    fputc(r, COM_A);
    delay_us(100);
    set_adc_channel(4);
    r = read_adc();
    fputc(r, COM_A);
    delay_us(100);
    set_adc_channel(5);
    r = read_adc();
    fputc(r, COM_A);
}

}
```

9.10 Pannel de produção completo das placas

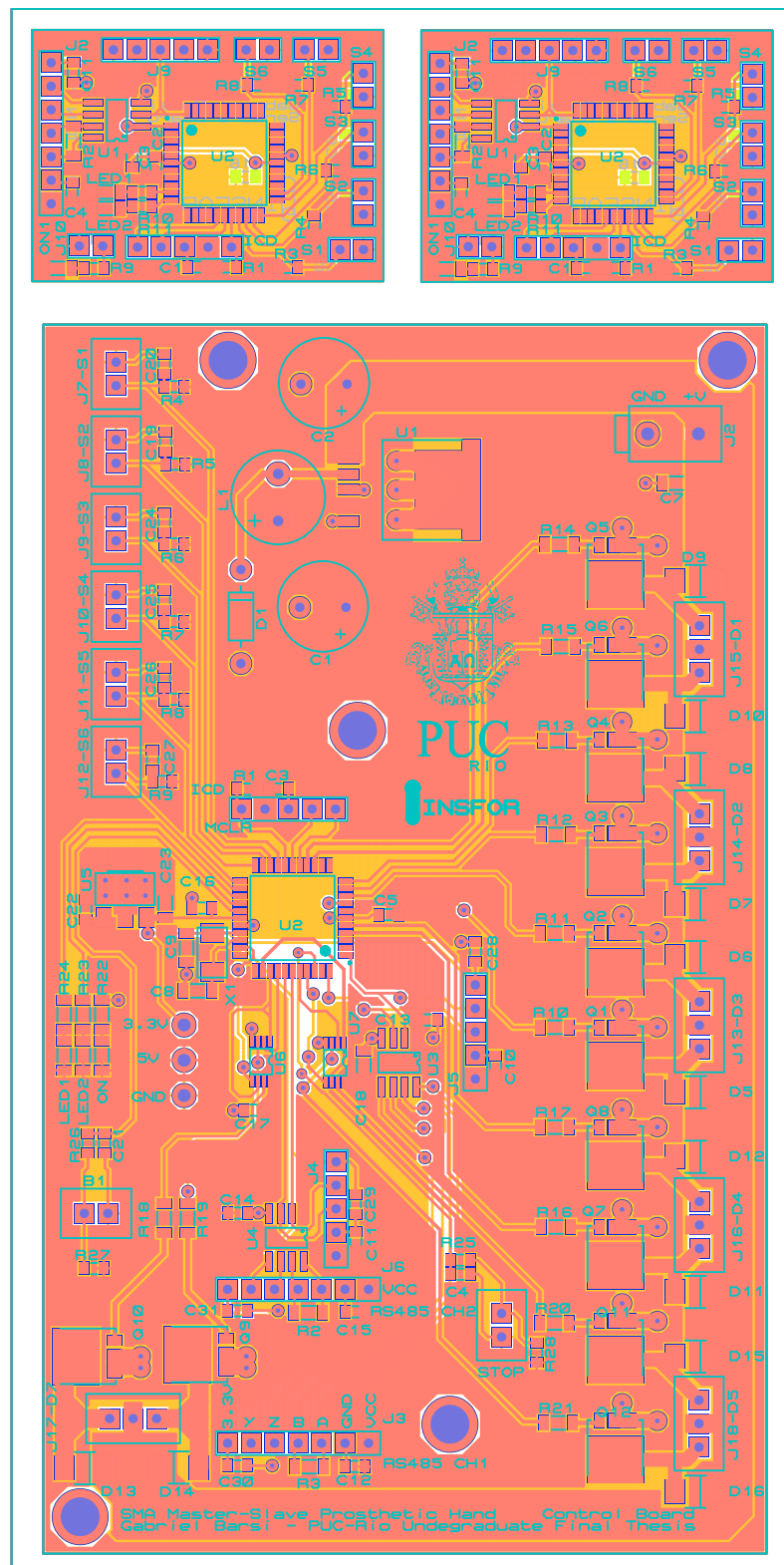


Figura 9.5: Pannel completo de produção das placas, output visual dos arquivos Gerber RS-274X através do programa Gerbv.

9.11 Código em MATLAB para aquisição e comparação da performance do MOSFET

```

%% Initialize
%
clc
clear all
close all
%% Acquire data
%
M = csvread('wvfc_gate.csv'); %read from csv

rows = size(M,1); %read number of rows

time = M(1:rows,1); %associate data
volt = M(1:rows,2);

%% Plot from oscilloscope
%

figure
hold on
plot(time,volt, 'LineWidth',1,'DisplayName','Sinal Original');
grid minor
axis([-3*10^-6 12*10^-6 -0.5 5.5])
N = 10;
d(1:N)=1/N;
voltf=filtfilt(d,1,volt);
plot(time,voltf, 'LineWidth',1,'DisplayName','Sinal Filtrado');
h = legend('-DynamicLegend');
h.Location = 'SouthEast';
xlabel('Tempo [s]')
ylabel('Tensão [V]')
hold off

%% FFT Analysis
%

[pxx,f] = pwelch(volt,400,[],5000,50000);
figure
hold on
plot(f,pow2db(pxx))
xlabel('Frequência (Hz)')
ylabel('Magnitude (dB)')
[pxx,f] = pwelch(voltf,400,[],5000,50000);
plot(f,pow2db(pxx))
legend('Sinal Original do Osciloscópio','Sinal Filtrado com média movel N=10');
grid minor
hold off

%% Declare parameters and TF
%

s = tf('s');
R = 220;          %%gate resistor
C = 12*10^-9;    %%gate-drain charge
H = 5/(1+s*R*C); %%RC TF

```

```
%% Plot comparison
%

figure
hold on
step(H,12*10^-6)
plot(time+.12*10^-5,voltf,'red','LineWidth',1,'DisplayName','Sinal Filtrado');
h = legend('Téorico','Real filtrado');
h.Location = 'best';
xlabel('Tempo [s]')
ylabel('Tensão [V]')
grid minor
hold off
```