

TRABALHO DE GRADUAÇÃO

**DESENVOLVIMENTO DE UM SISTEMA SUPERVISÓRIO
PARA EXPERIMENTOS DE CONTROLE E
IMPLEMENTAÇÃO DE UM PENDUBOT**

Eduardo de Moraes Vieira Silva

Yuji Wilker Nakahati

Brasília, julho de 2007

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

**DESENVOLVIMENTO DE UM SISTEMA SUPERVISÓRIO
PARA EXPERIMENTOS DE CONTROLE E
IMPLEMENTAÇÃO DE UM PENDUBOT**

Eduardo de Moraes Vieira Silva

Yuji Wilker Nakahati

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Geovany Araújo Borges, ENE/UnB
Orientador

Profa. Flavia Maria Guerra de Sousa Aranha
Oliveira, ENE/UnB

Prof. Ricardo Zelenovsky, ENE/UnB

Dedicatórias

À minha mãe, Janis, por ser a maior referência em minha vida e por ter me orientado quando eu mais precisava. Aos meus exemplares tios, White e Marília, que possibilitaram o meus estudos em Brasília. Ao meu amigo Eduardo e ao professor Geovany pelo esforço neste trabalho.

Yuji Wilker Nakahati

Dedico este trabalho aos meus pais, Luis Eugênio e Rosangela, e a minha irmã Simone. Aos meus familiares e amigos, além do Yuji e dos professores Geovany e Edson. .

Eduardo de Moraes Vieira Silva

Agradecimentos

*Agradeço, principalmente, a Deus por tudo que conquistei até aqui.
Aos meus pais que propiciaram esse momento.
À minha irmã que me aturou todo esse tempo.
Aos familiares que tanto me incentivaram.
Aos professores Edson e Geovany pela confiança no meu trabalho.
Aos amigos de Brasília por esses alegres anos de convivência.
Aos amigos do curso que vivenciaram essa fase junto comigo.
À MecaJun pelas oportunidades oferecidas.
Aos amigos do LARA que tanto contribuíram para esse trabalho.
Aos amigos de São Sebastião pelas férias e viagens revitalizantes.
Ao amigo Yuji pelo companheirismo e trabalho durante vários projetos juntos.*

Eduardo de Moraes Vieira Silva

*À minha grande família, Janis, Ysamu, Júnior, Jean, Vinícius, Vitória, Ana Kelly, White, Marília, Leonardo, Gabriel, Túlio, Jorge e Maurício, que me apoiam em tudo que faço.
À minha grande namorada e amiga, Ana Kelly, que acompanha e incentiva, incondicionalmente, em todos os momentos.
Ao professor Geovany por ser um excelente professor, orientar este trabalho e jamais economizar esforços para atender as minhas dúvidas.
Ao professor Edson pelo apoio nos trâmites da faculdade e por ser um exemplar profissional.
Ao meu amigo Eduardo pela dedicação no trabalho e compreensão nos meus momentos de ausência em todos os projetos que realizamos.
Aos amigos da faculdade que me acompanham nesta longa jornada.
Aos funcionários do SG-11 e SG-09, Cícero, Silmar, Adaiu, Arthur, Marcos e Carlos.*

Yuji Wilker Nakahati

RESUMO

O presente trabalho visa desenvolver uma ferramenta para automatizar e padronizar a supervisão de experimentos de controle. Neste sentido, foi projetada uma placa multifuncional que permite adquirir sinais, processá-los e gerar saídas para controlar diversos experimentos. Além disso, tal placa é capaz de enviar dados pré-selecionados via conexão USB. Para análise dos dados e interface com o usuário utilizou-se o software Matlab[®], criando assim uma janela amigável para que o usuário possa preocupar-se apenas com o seu experimento. Uma vez que o Matlab[®] não possui *driver* para comunicação USB (*Universal Serial Bus*), criou-se um programa servidor para gerenciar a troca de dados entre a placa e o Matlab[®]. Esse servidor foi desenvolvido para ser transparente ao protocolo de comunicação e poder ser utilizado por outros aplicativos. O pendubot foi construído a fim de validar o sistema supervisorio e também permitir estudos de técnicas de controle mais avançadas.

ABSTRACT

This work aims at the development of a supervisory system for control experiments. A multifunctional board is designed to acquire signals, process data them and compute outputs for many types of control experiments. This board is also capable to send pre selected data to a computer using USB communication. Matlab[®] software was used to analyse and present the data in a friendly interface. It is developed a program to control data exchange between Matlab and the control module since Matlab doesn't provide a driver for USB communication. This program is a server for interfacing with any other program in user space using shared memory. The system is validated on a control experiment designed only for this purpose: the pendubot. The Pendubot allows studies on more advanced control techniques.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	OBJETIVOS DO PROJETO	1
1.3	SISTEMA PROPOSTO	2
1.4	APRESENTAÇÃO DO MANUSCRITO	2
2	FUNDAMENTOS BÁSICOS	3
2.1	PLACAS MULTIFUNCIONAIS	3
2.2	USB	4
2.3	FERRAMENTAS DE ANÁLISE GRÁFICA PARA EXPERIMENTOS DE CONTROLE	5
2.4	AMBIENTE DE DESENVOLVIMENTO ARM	5
2.5	ARQUITETURAS DE PÊNULOS	6
2.6	SENSORES DE MEDIÇÃO ANGULAR	7
2.7	MODELO E ACIONAMENTO DE MOTOR CC	10
3	DESENVOLVIMENTO	12
3.1	ARQUITETURA GERAL	12
3.2	MÓDULO DE CONTROLE	12
3.2.1	ROTINA DE CONTROLE	15
3.2.2	CONVERSOR A/D	15
3.2.3	CONVERSOR D/A	18
3.2.4	CODIFICADOR DE MEDIÇÃO ANGULAR	20
3.2.5	E/S	24
3.2.6	MODULAÇÃO POR LARGURA DE PULSO	24
3.2.7	BIBLIOTECA DE FUNÇÕES	25
3.2.8	USB E JTAG	26
3.3	COMUNICAÇÃO ENTRE INTERFACE GRÁFICA E O MÓDULO DE CONTROLE	26
3.4	INTERFACE MATLAB [®]	28
3.5	PENDUBOT	30
3.5.1	MOTOR	31
3.5.2	MÓDULO DE POTÊNCIA	31
3.5.3	CODIFICADOR ABSOLUTO E INCREMENTAL	34
3.5.4	BASE ESTRUTURAL	34
3.6	SISTEMA SUPERVISÓRIO	35

4 CONCLUSÕES	39
REFERÊNCIAS BIBLIOGRÁFICAS	41
ANEXOS	42
I DIAGRAMAS ESQUEMÁTICOS	43
II DESENHOS MECÂNICOS	47
III DESCRIÇÃO DO CONTEÚDO DO CD	56

LISTA DE FIGURAS

1.1	Sistema proposto.....	2
2.1	Placa de aquisição USB-1208FS	3
2.2	Placa de desenvolvimento OP_KMAVR103.....	3
2.3	Placa de desenvolvimento Q8.....	4
2.4	Ambiente de desenvolvimento WinARM	5
2.5	Pêndulo invertido.....	6
2.6	Pêndulo duplo	6
2.7	Acrobot [1]	7
2.8	Pêndulo de Furuta [2]	7
2.9	Pendubot [3]	7
2.10	Potenciômetro convencional.....	7
2.11	Disco de codificação de Gray de 3-bits	8
2.12	Disco do codificador óptico incremental	8
2.13	Diagrama de fase	9
2.14	Codificador magnético incremental.....	9
2.15	codificador magnético absoluto.....	9
2.16	Circuito equivalente do motor [4].....	11
2.17	Tensão sobre o motor no acionamento chaveado.....	11
3.1	Diagrama da arquitetura geral do sistema supervisorio.....	12
3.2	Foto da arquitetura desenvolvida	13
3.3	Módulo de controle.....	14
3.4	Placa do microcontrolador	14
3.5	Diagrama do módulo de controle	14
3.6	Arquitetura da rotina de controle.	16
3.7	Amplificador Instrumental INA126	16
3.8	Circuito de um canal ADC	17
3.9	Sinal captado pelo conversor A/D.....	18
3.10	Circuito de um canal do conversor D/A	19
3.11	Sinal gerado utilizando o D/A.....	19
3.12	Onda atualizado a cada 1ms.....	19
3.13	Circuito de um canal do codificador	20
3.14	Gráfico de incrementos x tempo.	21
3.15	Gráfico da velocidade incremental x tempo.....	22

3.16	Diagrama de medição de ciclo de trabalho	22
3.17	Gráfico de ciclo de trabalho x Tempo	23
3.18	Gráfico de velocidade de ciclo de trabalho x Tempo	23
3.19	Circuito de um canal de entrada digital.	24
3.20	Circuito de um canal de saída digital.	24
3.21	Sinais S0 e S1	25
3.22	Sinais S2 e S3	25
3.23	<i>Fall-time</i> dos sinais S2 e S3	25
3.24	Implementação de controle	26
3.25	Seleção de variáveis	26
3.26	Diagrama de comunicação	27
3.27	Janela do sistema	29
3.28	Parte de comunicação	29
3.29	Botões de controle	30
3.30	Parte de visualização rápida	30
3.31	Diagrama do pendubot	31
3.32	Motor GM-14902S020	31
3.33	Módulo de potência	32
3.34	Módulo de potência aberto	32
3.35	Diagrama blocos do sistema	32
3.36	Resposta do sistema em malha aberta	33
3.37	LGR do sistema	33
3.38	Resposta do sistema em malha fechada	33
3.39	Gráfico Tensão referência <i>versus</i> Corrente	34
3.40	Ângulos θ_1 e θ_2 do pendubot	35
3.41	Codificador óptico incremental E5S	35
3.42	Codificador magnético absoluto MA2-P	35
3.43	Desenho da base	36
3.44	Foto da base	36
3.45	Link1	36
3.46	Link2	36
3.47	Acoplamentos do link 1 e link 2	37
3.48	Dados do sensor incremental	38
3.49	Dados do sensor absoluto	38
3.50	Gráfico de posição angulo com torque constante	38
I.1	Circuito dos conversores A/D e D/A	44
I.2	Circuito dos periféricos IO, MLP, Codificadores e a fonte de alimentação	45
I.3	Circuito da placa comercial SAM7-H64	46
II.1	Base do experimento	48
II.2	Suporte de aço para o motor e barras	49
II.3	Motor e caixa de redução	50
II.4	Extensor do eixo do motor	51

II.5	Primeira Barra: Link1	52
II.6	Eixo de conexão entre o Link1 e Link2	53
II.7	Suporte do codificador absoluto	54
II.8	Segunda Barra: Link2	55

LISTA DE TABELAS

2.1	Comparação de ambientes de desenvolvimento ARM	6
2.2	Tabela de decodificação de Gray de 3bits	8
2.3	Variáveis do motor	10
3.1	Mensagens de comunicação	28
3.2	Valores do motor	31
II.1	Desenhos contidos no anexo	47

LISTA DE SÍMBOLOS

Subscritos

<i>in</i>	entrada
<i>out</i>	saída
<i>Ref</i>	Referência

Siglas

A/D	Analógico para digital
D/A	Digital para analógico
MLP	Modulação por Largura de Pulso (<i>Pulse Width Modulation</i>)
E/S	Entrada/Saída
USB	Universal Serial Bus
JTAG	Joint Test Action Group
SPI	Serial Port Interface
CRC	Cyclic Redundancy Check
PCI	Peripheral Component Interconnect
LGR	Lugar Geométrico das Raízes
PIT	Periodic Interval Timer
LARA	Laboratório de Robótica e Automação
BNC	British Naval Connector

Capítulo 1

Introdução

1.1 Contextualização

A Engenharia de Controle e Automação é uma área muito abrangente, uma vez que é a sinergia entre três grandes campos: a Engenharia Elétrica, a Engenharia Mecânica e a Ciência da Computação, além da influência de outros campos como Administração e a Engenharia de Produção. Por ser um curso relativamente novo, vários laboratórios e experimentos ainda estão sendo implementados em diversas universidades. Este trabalho surge no intuito da criação de uma plataforma para supervisão de experimentos de controle.

Atualmente, diversos dispositivos de aquisição e processamento de sinais são encontrados comercialmente. Mesmo com diversas configurações disponíveis no mercado, escolheu-se criar um módulo próprio para obter as características desejadas e, principalmente, para dominar o conhecimento dessa tecnologia. Para externalização dos dados desse módulo de controle, optou-se por um meio de comunicação rápido e que fosse a nova tendência dessa área - o protocolo USB.

Uma característica importante para os sistemas supervisórios é prover ferramentas para análise e visualização dos dados. Ferramentas gráficas como Matlab®/Simulink® e Labview® são amplamente utilizadas em laboratórios para simulação de experimentos. Procurou-se aliar o poder dessas ferramentas gráficas com o módulo de controle para prover um sistema capaz de atender às necessidades dos usuários.

Por fim, escolheu-se o pendubot [3] como um experimento inicial para testes do sistema. A implementação desse experimento propicia o contato com diversas áreas tais como: eletrônica de potência, medição de grandezas e técnicas de construção mecânica. Além disso, esse experimento é usado para ensino nos cursos de controle e automação, pesquisa em sistemas não lineares, robótica[3].

1.2 Objetivos do projeto

Este trabalho tem como objetivo desenvolver um sistema supervisório para experimentos de controle a fim de que o usuário possa realizar seu experimento de forma simples e transparente.

Tal sistema, deve conter circuitos de aquisição e geração de sinais, rotina de controle executada em tempo real, protocolo que permita a identificação de erros e uma interface gráfica para visualização e análise das variáveis do experimento. Outro objetivo deste trabalho é a construção do aparato experimental de um pendubot. Este trabalho é uma continuação de [5] que elaborou uma simulação, modelagem e o controle deste tipo de processo.

1.3 Sistema Proposto

Este trabalho propõe um sistema supervisorio composto por módulo de controle, programa servidor, interface gráfica, módulo de potência e o pendubot conforme a Figura 1.1.

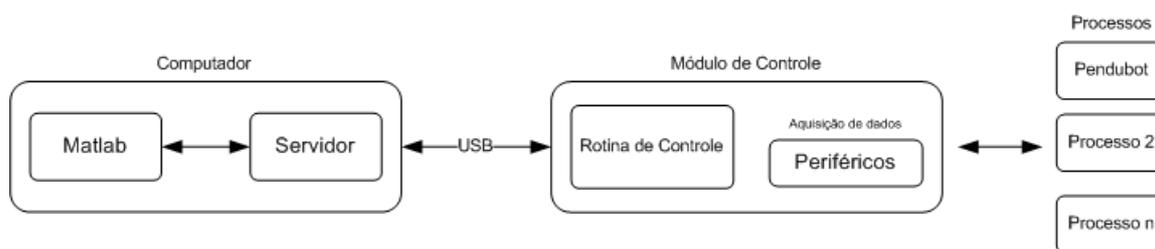


Figura 1.1: Sistema proposto

O módulo de controle é responsável pela aquisição das grandezas do experimento, neste caso o pendubot, além da execução da rotina de controle, geração de sinais e transmissão dos dados ao computador. A análise e visualização dos dados é auxiliada pelo software Matlab[®] sob um interface gráfica proposta. O padrão USB foi escolhido como canal de comunicação entre a interface e o módulo de controle. A troca de dados entre eles é feita por um programa servidor transparente ao protocolo de comunicação. É proposto um pendubot para validação do sistema e estudo de técnicas de controle.

1.4 Apresentação do manuscrito

Este documento está organizado da seguinte forma. No Capítulo 2, são demonstrados alguns fundamentos básicos para a compreensão deste trabalho. No Capítulo 3, é descrito o processo de desenvolvimento das diversas partes que compõem esse sistema com seus respectivos resultados preliminares. Ainda nesse capítulo, é demonstrada uma aplicação real desse sistema utilizando como experimento o pendubot assim como os resultados obtidos, seguido das conclusões no Capítulo 4. Os anexos contêm material complementar.

Capítulo 2

Fundamentos Básicos

2.1 Placas multifuncionais

Para realização de experimentos de controle, necessita-se de um sistema para aquisição dos dados. Tipicamente, os sistemas que realizam a coleta de dados são as placas de aquisição. Essas diferenciam-se uma das outras pela quantidade e variedade de entradas que elas suportam, acrescidas do barramento para comunicação externa. Atualmente, placas com barramentos USB (*Universal Serial Bus*), serial, ethernet etc são facilmente encontradas, porém a maioria ainda possui o barramento PCI (Peripheral Component Interconnect). Na Figura 2.1, tem-se a placa de aquisição USB-1208FS¹, baseada no barramento USB.



Figura 2.1: Placa de aquisição USB-1208FS

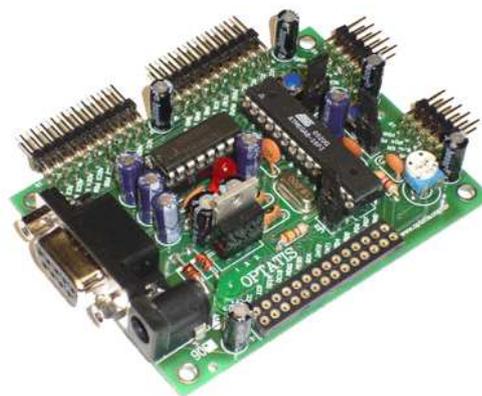


Figura 2.2: Placa de desenvolvimento OP_KMAVR103

As placas multifuncionais são outra forma para a aquisição dos dados, podendo ser acrescidas de outras funcionalidades como programação de seu microcontrolador. Dessa forma, o usuário passa a ter um controle maior sobre sua placa, além das vantagens do microcontrolador como entradas e saídas analógicas e digitais, conversores digital-analógico e analógico-digital, temporizadores e barramento para comunicação externa como serial, USB e outros. Vale ressaltar que cada placa ou *kit* possui suas características de acordo com seu fabricante e modelo. Na Figura 2.2, tem-se

¹<http://www.measurementcomputing.com>

a placa multifuncional OP_KMAVR103² externa ao computador com barramento serial. Já na Figura 2.3, tem-se a placa Q8³ que possui dois temporizadores de 32 bits, 32 canais digitais de E/S entre outras funcionalidades. Sendo seu barramento de comunicação PCI e essa placa é interna ao computador.



Figura 2.3: Placa de desenvolvimento Q8

2.2 USB

O padrão USB surgiu da necessidade de se integrar o computador ao telefone [6]. Atualmente, o padrão USB é muito utilizado nos mais variados tipos de dispositivos, suportando vários tipos de aplicações. Dentre das suas diversas características, destacam-se as seguintes: baixo custo, barramento diferencial, proteção CRC (*Cyclic Redundancy Check*), cabos padronizados, reconhecimento automático via *drivers*, suporte a variados tamanhos de pacotes de dados etc. Tal barramento está dividido em três principais intervalos:

- *Low-Speed* : 10 - 100kb/s;
- *Full-Speed*: 500kb/s - 10Mb/s;
- *High-Speed*: 25 - 400Mb/s.

No protocolo especificado, existem quatro tipos de transferência de dados[6]:

- *Control Transfer*: voltado para transferência de dados de configuração, status etc;
- *Isochronous Transfer*: voltado para comunicações com taxa de transferência constante;
- *Interrupt Transfer*: voltado para comunicações esporádicas;
- *Bulk Transfer*: voltado para comunicações de grande quantidade de dados.

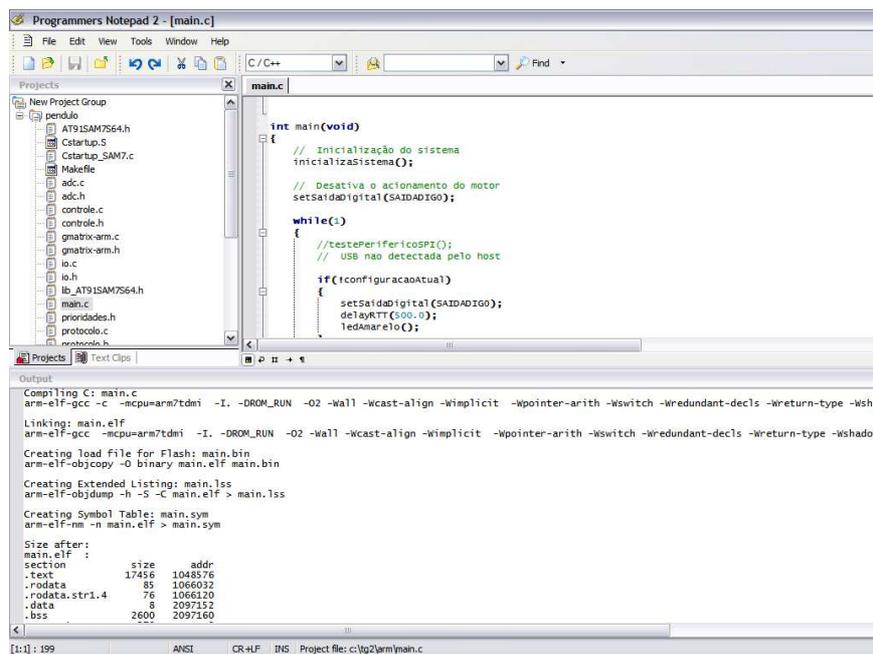
²<http://www.optatis.com.br>

³<http://www.quanser.com>

2.3 Ferramentas de análise gráfica para experimentos de controle

Ferramentas de análise gráfica são úteis, pois facilitam a análise e visualização dos dados. Duas ferramentas amplamente conhecidas no meio científico são o Matlab® e o LabView®. Uma terceira ferramenta que está aumentando sua comunidade de usuários é o Scilab. A família de produtos Matlab® disponibiliza uma linguagem de programação de alto nível, um ambiente técnico computacional interativo, funções para desenvolvimento de algoritmo, cálculos numéricos etc. O Matlab® também possui um conjunto significativo de *toolboxes* que auxiliam os experimentos de controle, tais como: *Control System Design and Analysis*, *Simulink* e outros. O Labview® é também uma ferramenta para aquisição de dados. Ele possui um ambiente de desenvolvimento gráfico e suporta uma vasta gama de placas de aquisição, além de permitir o desenvolvimento de uma interface gráfica para visualização dos dados e múltiplas opções de comunicação. Já o Scilab é um pacote de desenvolvimento científico para computação numérica principalmente de aplicações científicas e de engenharia. A principal diferença do Scilab é ser um projeto de código aberto. O Scilab também possui uma linguagem de programação de alto nível, além de aceitar códigos de outras linguagens como C, C++ e Fortran.

2.4 Ambiente de desenvolvimento ARM



```
int main(void)
{
    // Inicialização do sistema
    inicializaSistema();

    // Desativa o acionamento do motor
    setSaídaDigital(SAIDADIG0);

    while(1)
    {
        //testePerifericoSPI();
        // USB não detectada pelo host

        if(!configuracaoAtual)
        {
            setSaídaDigital(SAIDADIG0);
            delayRTT(500.0);
            ledanalelo();
        }
    }
}
```

Output

```
Compiling C: main.c
arm-elf-gcc -c -mcpu=arm7tdmi -I. -DROM_RUN -O2 -Wall -Wcast-align -Wimplicit -Wpointer-arith -Wswitch -Wredundant-decls -Wreturn-type -Wshadow

Linking: main.elf
arm-elf-gcc -mcpu=arm7tdmi -I. -DROM_RUN -O2 -Wall -Wcast-align -Wimplicit -Wpointer-arith -Wswitch -Wredundant-decls -Wreturn-type -Wshadow

Creating load file for Flash: main.bin
arm-elf-objcopy -O binary main.elf main.bin

Creating Extended Listing: main.lss
arm-elf-objdump -h -S -C main.elf > main.lss

Creating Symbol Table: main.sym
arm-elf-nm -n main.elf > main.sym

Size after:
main.elf :
section      size      addr
.text      17456    1048576
.rodata      85    1066032
.rodata.str1.4  76    1066120
.data         8    2097152
.bss        2600    2097160
```

Figura 2.4: Ambiente de desenvolvimento WinARM

Existem diversos ambientes de desenvolvimento para a família de microcontroladores ARM. Tais ambientes combinam ferramentas que auxiliam o processo de criação de programas, pacote convencional, como por exemplo: editor de texto, compilador, ligador, depurador etc. A Tabela 2.1, contém as principais características dos ambientes mais utilizados.

Dentre esses, o WinARM, Figura 2.4, destaca-se pelo fato de ser um *software* livre, ter uma

Tabela 2.1: Comparação de ambientes de desenvolvimento ARM

<i>Software</i>	Ferramentas Adicionais	Licença
<i>IAR Embedded Workbench</i>	RTOS	Proprietário
<i>CrossWork for ARM</i>	RTOS	Proprietário
<i>ARM Development Tools</i>	Multi-threaded	Proprietário
<i>RealView MDK-ARM</i>	RTOS, USB, GUI, Matlab®Interface	Proprietário
<i>WinARM</i>	<i>GNU ARM Toolchain etc</i>	Livre
<i>YAGARTO</i>	<i>GNU ARM Toolchain etc</i>	Livre

grande gama de exemplos para variados tipos de microcontroladores, conter uma interface simples e funcional e possuir uma lista de discussão ativa⁴.

2.5 Arquiteturas de Pêndulos

Dentre os vários experimentos de controle já propostos, tem-se uma classe específica que é a de pêndulos. O principal objetivo dessa classe é manter uma barra na posição vertical. Um dos mais conhecidos é o pêndulo invertido, Figura 2.5, em que se deve manter a barra na posição vertical apenas com o movimento linear do carrinho nos trilhos. No pêndulo duplo, a barra no segundo estágio (*second stage*) tem movimento livre na junta presa à barra no primeiro estágio (*first stage*), Figura 2.6.

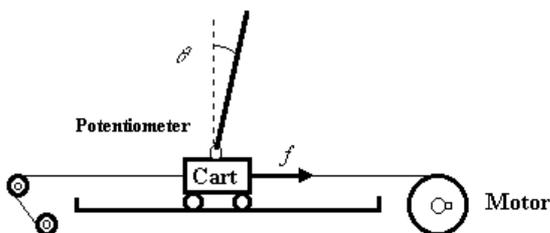


Figura 2.5: Pêndulo invertido

(fonte:<http://www.aptronix.com/fuzzynet/applnote/twostage.htm>)

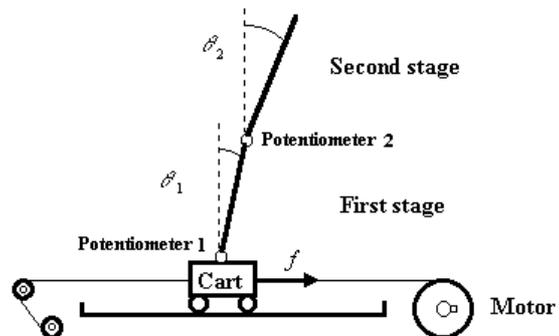


Figura 2.6: Pêndulo duplo

(fonte:<http://www.aptronix.com/fuzzynet/applnote/twostage.htm>)

Outro tipo de pêndulo é o Acrobot [1], mostrado na Figura 2.7, em que a barra interior (link 1) tem uma extremidade presa a um rolamento (ombro) e na junção (cotovelo) com a barra exterior (link 2) coloca-se o atuador. Baseado no Acrobot, surgiu o pêndulo de Furuta [2], Figura 2.8, no qual as barras são colocadas de forma ortogonal entre si. Procura-se atingir o objetivo nesse sistema apenas com a rotação da barra horizontal. Baseado em ambos, tem-se o Pendubot [3], Figura 2.9, que possui sua mecânica parecida com a do Acrobot, porém o atuador fica no ombro

⁴<http://en.mikrocontroller.net/>

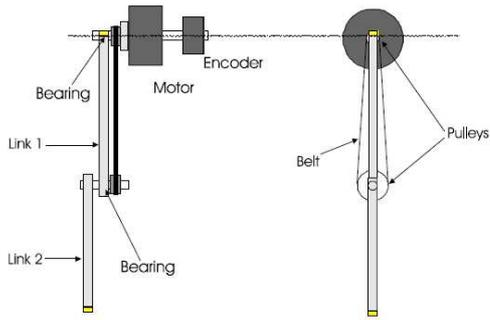


Figura 2.7: Acrobot [1]

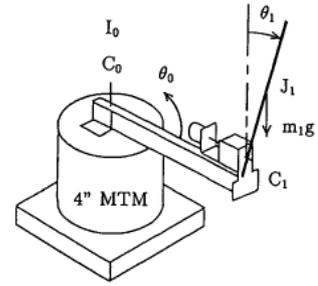


Figura 2.8: Pêndulo de Furuta [2]

e não no cotovelo como no pêndulo de Furuta.

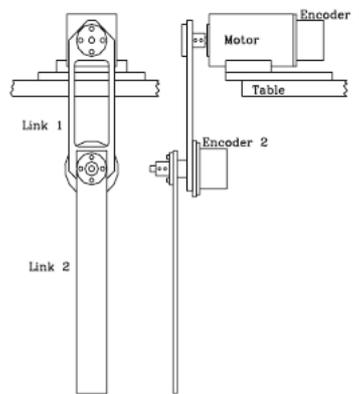


Figura 2.9: Pendubot [3]

2.6 Sensores de medição angular

Existem diversos tipos de sensores para medição de ângulos de rotação. Neste parágrafo, serão descritos os sensores resistivos, ópticos e magnéticos. O sensor resistivo é, possivelmente, a forma mais simples e barata para se medir uma determinada posição angular. O seu funcionamento baseia-se apenas em um resistor variável (potenciômetro) cujo esquema de funcionamento encontra-se na Figura 2.10, construído de maneira que a resistência em seus terminais é proporcional à posição angular θ do seu eixo.

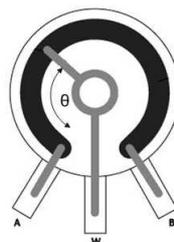


Figura 2.10: Potenciômetro convencional

Para tal sensor, tem-se que:

$$\theta_{A \rightarrow W} = 2\pi \frac{R_{A \rightarrow W}}{R_{A \rightarrow B}} \quad (2.1)$$

em que $\theta_{A \rightarrow W}$, $R_{A \rightarrow W}$, $R_{A \rightarrow B}$, é o ângulo entre os terminais A e W, resistência entre os terminais A e W e resistência entre os terminais A e B da Figura 2.10.

O sensor angular óptico, mais conhecido como *encoder* ou codificador óptico é largamente conhecido devido a sua flexibilidade de configuração, compatibilidade com sistemas digitais e por ser largamente utilizado no meio industrial. O seu funcionamento baseia-se em um par óptico (emissor e receptor) e um disco de codificação, os quais combinados podem fornecer a posição absoluta ou relativa. A posição absoluta é gerada utilizando um disco de codificação binária que geralmente utiliza o código de Gray. Na Figura 2.11, tem-se um exemplo de um código de Gray de 3 bits, os quais informam sua posição angular absoluta de acordo com a Tabela 2.2.

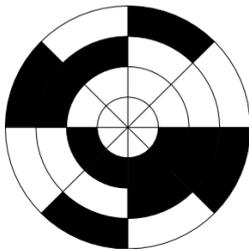


Figura 2.11: Disco de codificação de Gray de 3-bits

(http://en.wikipedia.org/wiki/Image:Encoder_disc\%283-Bit\binary\%29.svg)

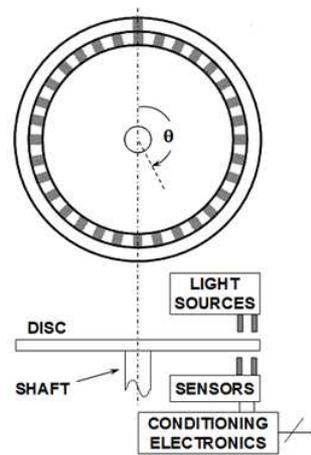


Figura 2.12: Disco do codificador óptico incremental

(http://www.analog.com/Analog_Root/static/miniTutorials/images/MT029_Figure1.gif)

Tabela 2.2: Tabela de decodificação de Gray de 3bits

Bit0	Bit1	Bit2	Ângulo
0	0	0	0 - 45 ⁰
0	0	1	45 - 90 ⁰
0	1	1	90 - 135 ⁰
0	1	0	135 - 180 ⁰
1	1	0	180 - 225 ⁰
1	1	1	225 - 270 ⁰
1	0	1	270 - 315 ⁰
1	0	0	315 - 360 ⁰

O codificador óptico incremental mede posição angular relativa que é obtida por meio de um disco simples que contém n divisões, como mostrado na Figura 2.12. Em casos mais completos,

pode ser acompanhado por uma marcação especial - *index* - e por dois pares ópticos (Canal 1 e Canal 2). Por meio desses pares e das n divisões, é possível contar a quantidade de pulsos captados relativo à posição inicial do codificador. A marcação especial é responsável por garantir essa posição. Os dois pares (canais) são fisicamente posicionados de maneira que eles possuam uma defasagem de 90° , assim é possível detectar o sentido de rotação analisando o sinal da defasagem de acordo com a Figura 2.13.

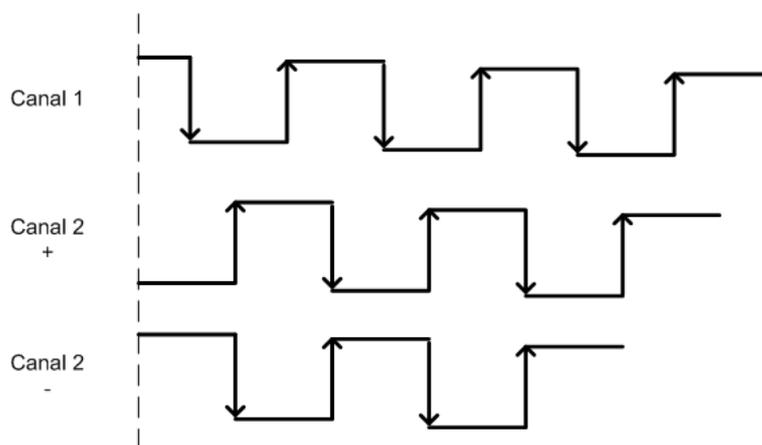


Figura 2.13: Diagrama de fase

O codificador magnético também pode ser usado para medidas incrementais e absolutas. Uma das formas para medidas incrementais é colocar um sensor de efeito Hall próximo a um disco dentado, em que os dentes são constituídos de material ferromagnético, Figura 2.14. A cada dente que passa em frente ao sensor um pulso é gerado. Para medidas absolutas, outros formatos são utilizados. Um desses formatos é colocar um ímã permanente sobre uma superfície e nessa superfície, um anel constituído de pequenos sensores de efeito Hall são alocados, Figura 2.15. Combinando as informações de cada sensor é possível detectar a distribuição do fluxo magnético.

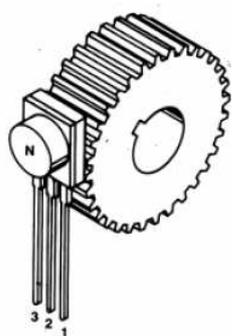


Figura 2.14: Codificador magnético incremental.

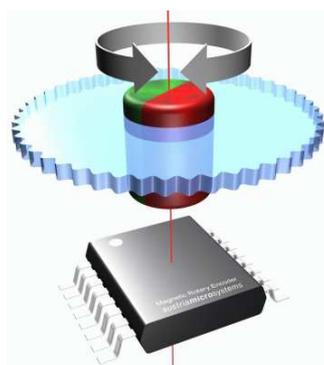


Figura 2.15: codificador magnético absoluto.
([http://www.eba.com.hk/upload/Magnetic\%20Rotary\%20Encoder\(press\).jpg](http://www.eba.com.hk/upload/Magnetic\%20Rotary\%20Encoder(press).jpg))

Devido às imperfeições dos codificadores, é aconselhável aplicar técnicas de condicionamento de sinais nas variáveis medidas. Dentre as diversas técnicas existentes, vale ressaltar para este trabalho três técnicas brevemente descritas abaixo.

- Saturação: garante que medidas empíricas ou fisicamente irreais sejam desconsideradas;
- Banda morta: garante a banda morta de acordo com a resolução do codificador e dessa forma permite que ele responda de forma esperada;
- Filtro passa baixa: minimiza ruídos indesejados com frequência acima da frequência de corte do filtro.

2.7 Modelo e acionamento de motor CC

O motor de corrente contínua é modelado segundo as equações abaixo [4]:

$$T_{em} = k_T i_a \quad (2.2)$$

$$E_a = k_E \omega_m \quad (2.3)$$

$$V_t = E_a + R_a I_a \quad (2.4)$$

$$\omega_m = \frac{1}{K_E} \left(V_t - \frac{R_a}{k_T} T_{em} \right) \quad (2.5)$$

Na Tabela 2.3, tem-se o significado de cada variável e, na Figura 2.16, tem-se a representação do circuito equivalente do motor.

Tabela 2.3: Variáveis do motor

T_{em}	Torque mecânico
k_T	Constante de torque do motor
i_a	Corrente de armadura do motor
k_e	Constante elétrica do motor
ω_m	Velocidade angular do eixo do motor
R_a	Resistência equivalente do motor
L_a	Indutância equivalente do motor
V_t	Tensão aplicada na armadura do motor

Pela equação 2.2, verifica-se que aplicando uma corrente na armadura do motor um torque proporcional é aplicado no seu eixo e pela equação 2.4, aplicando-se uma tensão nos terminais da armadura verifica-se o surgimento da corrente de armadura, ou seja, o motor pode ser acionado por corrente ou por tensão. O acionamento pode ser linear ou chaveado. O acionamento linear, geralmente, é usado para sistemas de baixa potência, pois parte da potência que deveria ser entregue ao motor é transformada em calor devido aos transistores estarem operando em sua região linear [4]. No segundo tipo de acionamento, a tensão é alternada como uma chave liga/desliga. Considerando t_H o instante de tempo em que a chave fica acionada e t_L o tempo em que a chave fica desacionada, tem-se que o valor da tensão média aplicada no motor V_M é

$$V_M = \frac{V_H t_H + V_L t_L}{t_H + t_L}, \quad (2.6)$$

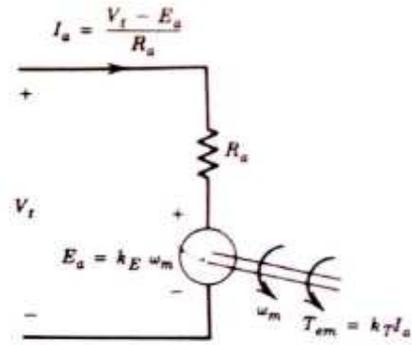


Figura 2.16: Circuito equivalente do motor [4]

em que V_H e V_L são as tensões aplicadas ao motor nos instantes t_H e t_L , respectivamente conforme mostra a Figura 2.17.

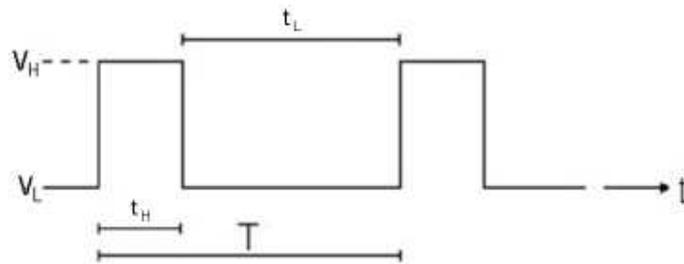


Figura 2.17: Tensão sobre o motor no acionamento chaveado.

Capítulo 3

Desenvolvimento

3.1 Arquitetura geral

O sistema supervisorio para experimentos de controle é composto pelo módulo de controle e uma interface gráfica que foi desenvolvida no ambiente Matlab®. Juntamente com o usuário, essas ferramentas tornam-se poderosas e capazes de controlar e estudar diversos tipos de experimentos, tais como: Pendubot, Ball on Beam (Bola na barra), Helicopter Pitch Control (Controle do ângulo de cabeceio do helicóptero), Motor Control (controle de motor) etc. Na Figura 3.1, tem-se o diagrama da arquitetura geral do sistema e na Figura 3.2, tem-se a foto da arquitetura desenvolvida.

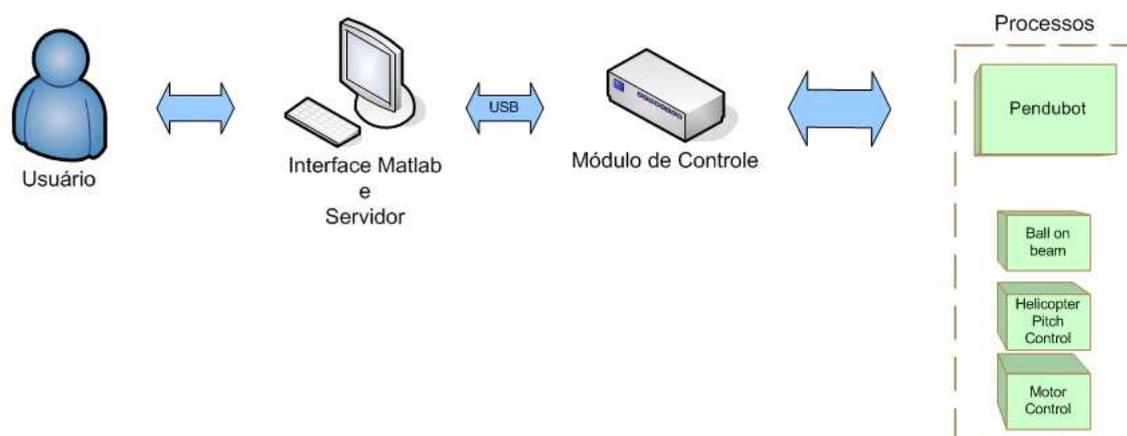


Figura 3.1: Diagrama da arquitetura geral do sistema supervisorio

3.2 Módulo de Controle

O módulo de controle, Figura 3.3, foi projetado no intuito de ser uma plataforma capaz de proporcionar um ambiente real para implementar diversas técnicas de controle em variados tipos



Figura 3.2: Foto da arquitetura desenvolvida

de experimentos. Para tanto, os seguintes requisitos foram especificados:

- Periférico de captura sinais: capturar sinais de sensores analógicos e digitais;
- Periférico de geração de sinais: gerar sinais para acionadores analógicos e digitais;
- Comunicação com o computador: trocar os dados do experimento em tempo real e programar o microcontrolador;
- Rotina de controle em tempo real: garantir o período constante da rotina de controle;
- Biblioteca de funções: biblioteca simplificada que permita acesso aos diversos periféricos do módulo de controle.

A fim de alcançar tais requisitos, foi escolhida a placa comercial SAM7-H64 como parte principal desse módulo de controle. Essa placa, Figura 3.4, foi desenvolvida baseada no microcontrolador AT91SAM7S64. Ela contém alguns circuitos preliminares responsáveis, principalmente, pela alimentação, clock e externalização dos pinos do microcontrolador. AT91SAM7S64 é um microcontrolador baseado no processador ARM7TDMI-S RISC de 32-bit de alto desempenho e contém 64Kbytes de memória flash, 16Kbytes de memória SRAM, periféricos diversos, conectividade USB 2.0 etc. Para captura de sinais, foi projetado o periférico de conversão A/D (Analogico para digital), codificadores e entradas digitais. Para a geração de sinais, foi projetado o periférico de conversão D/A (Digital para analógico), MLP (Modulação por largura de pulso) e saídas digitais. De posse de todos esses periféricos e a placa SAM7-H64, foi proposta arquitetura da Figura 3.5 para o módulo de controle.

Para atender aos requisitos de comunicação foram utilizados dois barramentos. Para a troca de dados entre o computador e o módulo de controle, foi utilizado o barramento USB que permite alta taxa de transferência e para a programação do microcontrolador, foi utilizado o barramento



Figura 3.3: Módulo de controle



Figura 3.4: Placa do microcontrolador

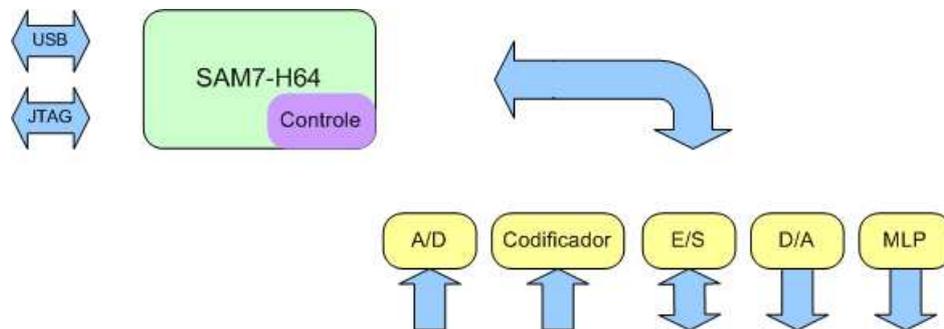


Figura 3.5: Diagrama do módulo de controle

JTAG que permite tanto a programação quanto o *debug* do microcontrolador. Para criação da rotina de controle em tempo real, foi utilizado o periférico PIT (Periodic Interval Timer) do microcontrolador AT91SAM7S64 que permite que ela seja executada com frequência ajustável. A biblioteca de funções é um conjunto de algoritmos escritos na linguagem C que permite o acesso aos diversos periféricos do microcontrolador. Além desses requisitos, foi implementado dois estados de funcionamento: *standby* - experimento parado e diversos comandos podem ser executados, *running* - experimento em execução e somente comando de finalizar pode ser executado. Dessa forma, o módulo de controle torna-se uma ferramenta poderosa no ensino em laboratórios de controle. Segue abaixo suas principais características:

- Microcontrolador: AT91SAM7S64 com arquitetura de alto desempenho ARM7TDMI-S RISC;
- Interfaces de comunicação: USB para troca de dados entre o computador e o módulo de controle e JTAG para a programação e *debug* do microcontrolador;

- Rotina de controle: rotina dedicada executada a frequência ajustável;
- Canais de entradas/saídas digitais: 6 canais de entrada e 6 canais de saída digitais;
- Canais A/D e D/A: 4 canais A/D de 10bits e 2 canais D/A de 12 bits;
- Canais de codificadores: dois canais de codificadores configuráveis para trabalhar em modo incremental ou absoluto;
- Canais MLP: 4 canais MLP com frequência ajustável;
- Modos de funcionamento: *standby* e *running*;
- *Leds* indicativos: 3 *leds* indicando sistema ligado, USB em atividade e rotina de controle ativada.

3.2.1 Rotina de controle

Essa rotina foi construída voltada para controlar sistemas relativamente lentos, uma vez que o módulo de controle possui uma série de limitações, tais como: taxa de atualização do periférico Conversor D/A, custo de processamento de sinais, custo de envio dos dados em tempo real etc. Dessa forma, ela foi programada para ser executada com uma periodicidade entre 1ms e 300ms. Nela, é possível adquirir sinais, processá-los, excitar o experimento ou somente estudá-lo, assim como selecionar os dados que devem ser enviados para a interface Matlab®. Na Figura 3.6, é demonstrado o fluxograma dessa rotina.

3.2.2 Conversor A/D

O periférico Conversor A/D foi projetado a fim de seja capaz de capturar sinais analógicos. Para que ele funcione de forma satisfatória dentro da rotina de controle, alguns requisitos mínimos são necessários, tais como: ler sinais entre -5V a 5V com considerável resolução, baixa tensão de *offset* e ser imune à ruídos de alta frequência. Para tanto, utilizou-se o conversor A/D de 10 bits do microcontrolador de que opera entre 0V a 3.3V. A fim de corrigir essa incompatibilidade, utilizou-se o amplificador de instrumentação INA126 desenvolvido para aplicações que exigem alta precisão, Figura 3.7.

De acordo com o manual do amplificador INA126¹, tem-se que

$$V_{out} = (V_{in}^+ - V_{in}^-)G + V_{Ref} \quad (3.1)$$

$$G = 5 + \frac{80k}{R_G} \quad (3.2)$$

e utilizando V_{in}^+ como a variável da função V_{out} e de acordo com o requisito de intervalo de leitura (-5V a 5V), tem-se que

¹Micropower Instrumentation Amplifier Single and Dual Versions - Texas Instruments Inc.

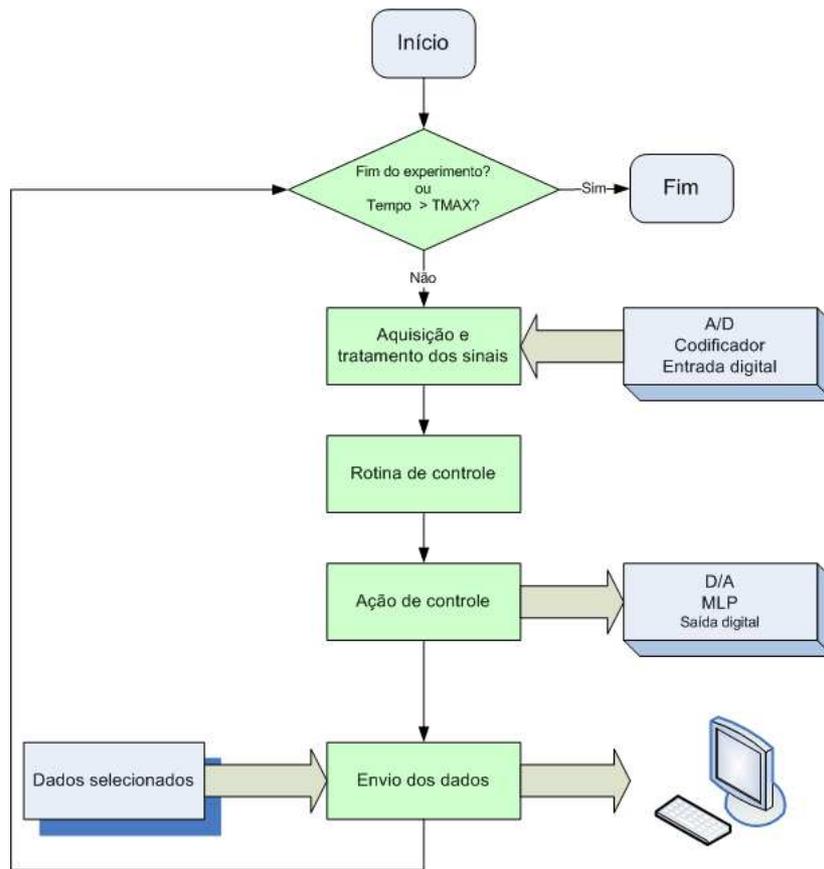


Figura 3.6: Arquitetura da rotina de controle.

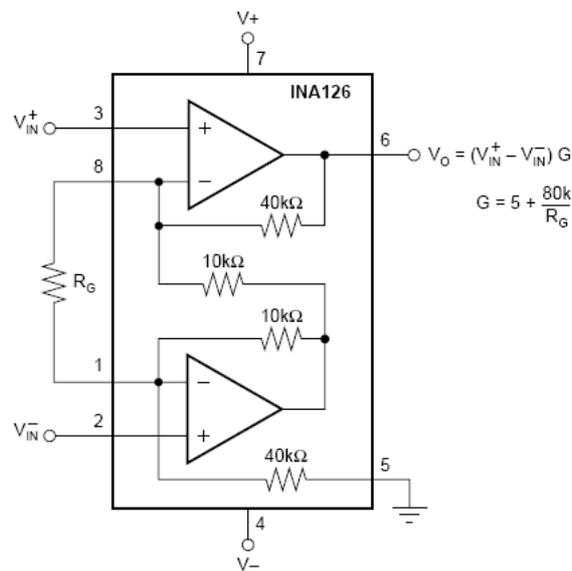


Figura 3.7: Amplificador Instrumental INA126

$$V_{out}(+5) = 3.3 \quad (3.3)$$

$$V_{out}(-5) = 0 \quad (3.4)$$

em que V_{out} , V_{in}^+ , V_{in}^- , G e V_{Ref} é a tensão de saída, tensão de entrada não inversora, tensão de entrada inversora, ganho e tensão de referência do amplificador de instrumentação INA126 (pino cinco da Figura 3.7), respectivamente. Fazendo

$$V_{in}^- = 0 \quad (3.5)$$

$$V_{out}(x) = ax + b \quad (3.6)$$

utilizando 3.3 e 3.4, tem-se

$$G = \frac{3.3}{10} \quad (3.7)$$

$$V_{Ref} = \frac{3.3}{2} \quad (3.8)$$

mas como $G \geq 5$, então adicionou-se, com um resistor variável, uma atenuação de $\frac{3.3}{50}$ antes de V_{in}^+ utilizando um divisor de tensão antecedido por *buffers* isoladores construídos utilizando o amplificador operacional *rail-to-rail* MCP604 desenvolvido para aplicações de conversão analógica digital². Também foi utilizado um filtro capacitivo a fim filtrar ruídos de alta frequência. Vide a Figura 3.8 para os detalhes do circuito.

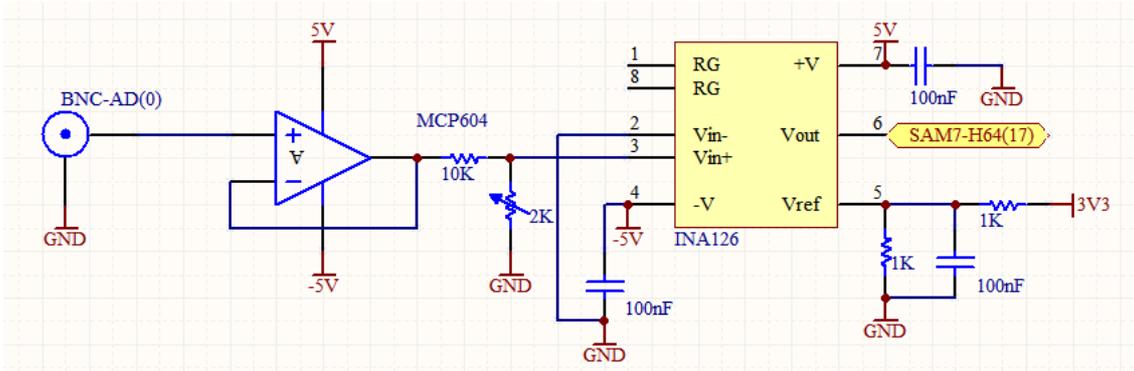


Figura 3.8: Circuito de um canal ADC

Dessa forma, foi possível corrigir, teoricamente, a incompatibilidade de intervalo de leitura. A resolução deste periférico é de aproximadamente $\frac{5-(-5)}{2^{10}} = 9,7656mV$. A tensão de *offset* deste periférico, $T_{offset-A/D}$, pode ser estimada somando-se a tensão de *offset* do amplificador MCP604, $T_{offset-MCP604} = \pm 0,7mV^3$, mais a tensão de *offset* do amplificador INA126, $T_{offset-INA126} = \pm 0,250mV^4$ e mais o tensão de *offset* de V_{Ref} devido aos erros de 5% nos resistores, $T_{offset-Vref} = \pm 866.25mV$. Logo, $T_{offset-A/D} = \pm 867,2mV$. Para minizar $T_{offset-A/D}$, aconselha-se utilizar resistores de precisão ou um resistor variável para o ajuste correto de V_{Ref} . Vale a pena ressaltar que a rotina de controle é executada a uma frequência máxima de 1KHz, logo requisitos de largura de banda são facilmente alcançados.

Para testar esse módulo, um gerador de onda foi acoplado a uma das entradas e foi aplicado uma onda senoidal $5.04sen(wt)$ com $w = 2\pi$. A Figura 3.9, foi obtida utilizando a rotina de

²MCP601/2/3/4 - 2.7V to 5.5V Single-Supply CMOS Op Amps - Microchip Technology Inc.

³MCP601/2/3/4 - 2.7V to 5.5V Single-Supply CMOS Op Amps - Microchip Technology Inc.

⁴Micropower Instrumentation Amplifier Single and Dual Versions - Texas Instruments Inc.

controle que estava sendo executada a cada $10ms$. Pelo gráfico, nota-se que houve saturações em torno de $4.355V$ e $-4.9V$. Isso ocorreu devido ao amplificador instrumental INA126 não possuir a característica *rail-to-rail* o que pode ser corrigido facilmente se ele for alimentado com $\pm 12V$. Observam-se os efeitos da amostragem mesmo com a frequência de amostragem 100 vezes maior que a frequência do sinal.

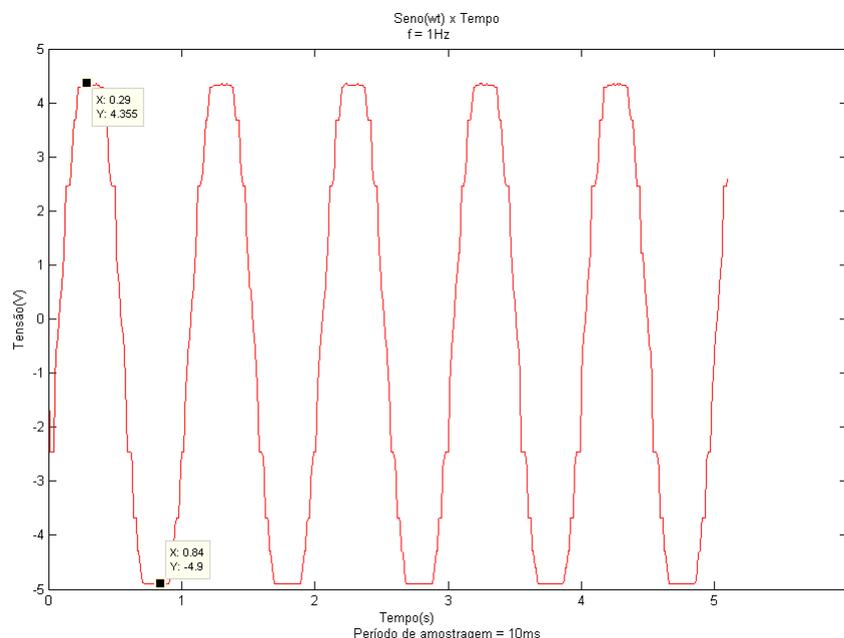


Figura 3.9: Sinal captado pelo conversor A/D

3.2.3 Conversor D/A

O periférico Conversor D/A foi projetado a fim de que possa gerar sinais analógicos. Para que ele opere satisfatoriamente, alguns requisitos são necessários, tais como: gerar sinais entre $-5V$ a $5V$, considerável resolução e baixa tensão de *offset*. Para tanto e por meio do periférico SPI (Serial Port Interface) do microcontrolador, utilizou-se o circuito integrado MCP4922 de 12 bits de alta precisão desenvolvido para aplicações industriais⁵, uma vez que o microcontrolador não possui um conversor D/A. Porém, tal conversor gera sinais entre $0V$ a $5V$, logo foi necessário utilizar o amplificador INA126 da Figura 3.7 para corrigir essa incompatibilidade intervalo de tensões.

De acordo com o requisito de intervalo de tensão ($-5V$ a $5V$) e utilizando a equação 3.6, tem-se que

$$V_{out}(+5) = 3.3 \quad (3.9)$$

$$V_{out}(-5) = 0 \quad (3.10)$$

$$G = 2 \quad (3.11)$$

$$V_{Ref} = -5 \quad (3.12)$$

⁵MCP4921/4922 - 12-Bit DAC with SPI Interface - Microchip Technology Inc.

mas como $G \geq 5$ (restrição do amplificador INA126), então adicionou-se uma atenuação de $\frac{1}{11}$ e ajustou-se $G = 22$. Assim, foi possível construir o circuito da Figura 3.10.

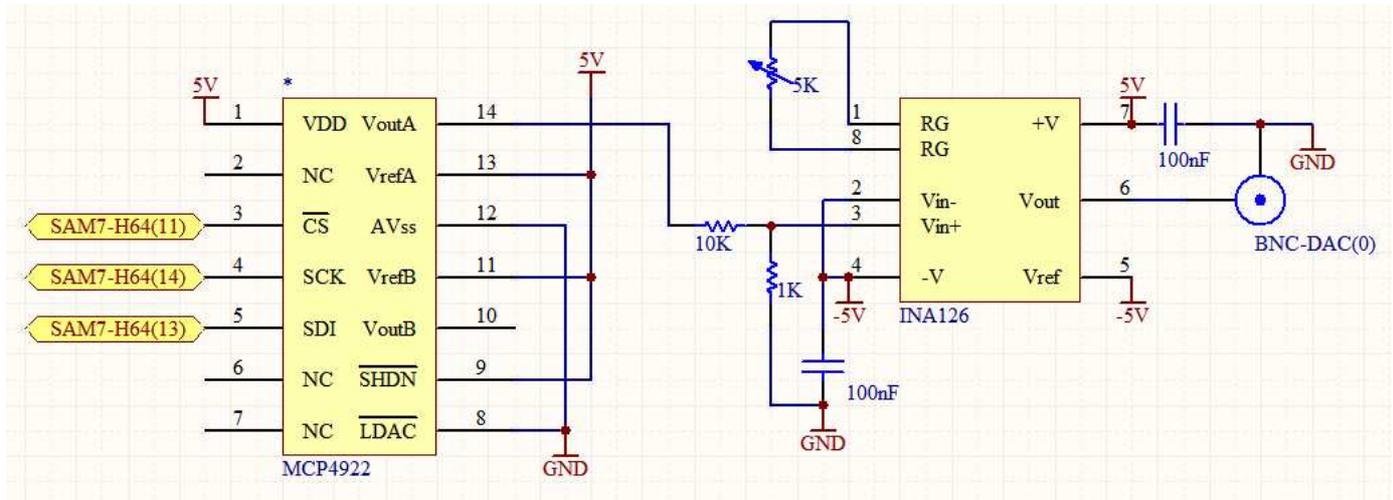


Figura 3.10: Circuito de um canal do conversor D/A

Utilizando esse amplificador, foi possível corrigir, teoricamente, a incompatibilidade de intervalo de geração de sinais. A resolução deste periférico é de aproximadamente $\frac{5-(-5)}{2^{12}} = 2,4414mV$. A tensão de *offset* deste periférico, $T_{offset-D/A}$ pode ser estimada somando-se a tensão de *offset* do conversor D/A MCP4922, $T_{offset-MCP4922} = \pm 1mV^6$, mais a tensão de *offset* do amplificador INA126, $T_{offset-INA126} = \pm 0,250mV$. Logo, $T_{offset-D/A} = \pm 1,25mV$. Vale a pena ressaltar que a rotina de controle é executada a uma frequência máxima de 1KHz, logo requisitos de taxa de atualização de saída do periférico conversor D/A podem ser alcançados. Para testar o periférico, tentou-se gerar uma onda senoidal $5sen(\omega t)$ com $\omega = 2\pi 100$, Figura 3.11, utilizando a rotina de controle que estava sendo executada a cada 1ms, Figura 3.12.

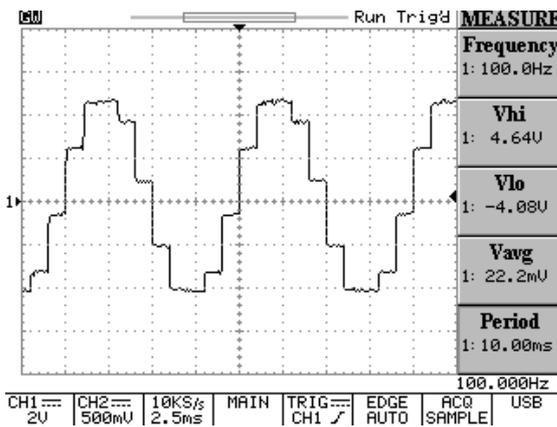


Figura 3.11: Sinal gerado utilizando o D/A.

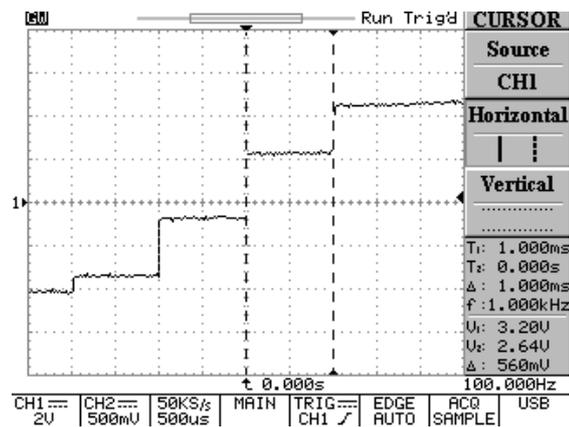


Figura 3.12: Onda atualizado a cada 1ms

Verificou-se que a taxa de atualização do valor da onda gerada foi satisfatória, uma vez que a rotina de controle foi projetada para ser executada com um período mínimo de 1ms. Porém, em razão da saturação do amplificador INA126, ocorreram limitações no sinal gerado de $V_{hi} = 4.64V$ e

⁶MCP4921/4922 - 12-Bit DAC with SPI Interface - Microchip Technology Inc.

$V_{lo} = -4.08V$ em que V_{hi} e V_{lo} é a tensão máxima e mínima alcançadas pelo periférico conversor D/A, respectivamente. Isso pode ser corrigido facilmente se o amplificador INA126 for alimentado com $\pm 12V$.

3.2.4 Codificador de medição angular

O periférico Codificador de medição angular foi projetado, basicamente, com a finalidade de captar sinais padrões de codificadores incrementais e o ciclo de trabalho de um sinal modulado por largura de pulso. Dessa forma, foi possível medir tanto o número de incrementos ou ciclo de trabalho quanto a variação dessas medidas com o tempo. Utilizou-se o circuito integrado 74LS74, flip-flop do tipo D, no intuito de detectar o sentido de rotação do disco do codificador incremental. Essa escolha deve-se, principalmente, à alta frequência de *clock* (20KHz)⁷. Também foram utilizadas as portas inversoras *schmitt-trigger* 74LS14 com histerese de 0,8V⁸ para conferir maior proteção e imunidade a ruídos. Na Figura 3.13, mostra-se o circuito de um dos canais desse periférico.

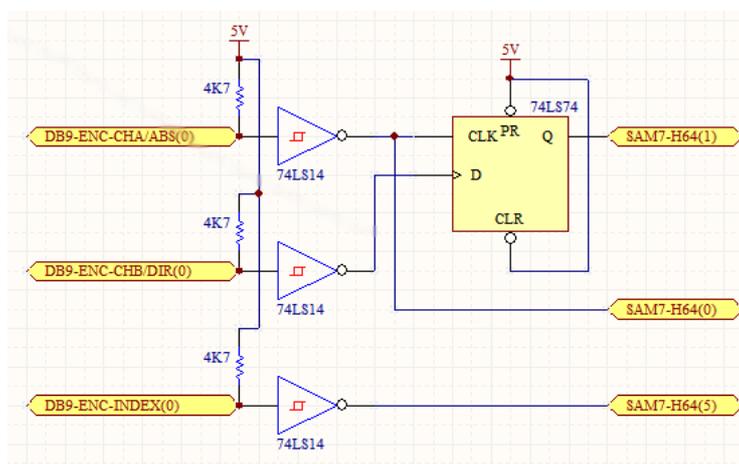


Figura 3.13: Circuito de um canal do codificador

Para a medição do número de incrementos, utilizou-se interrupção na borda de descida. Uma variável inteira de 32bits é incrementada ou decrementada de acordo com o sentido de rotação do disco. Experimentalmente, notou-se que utilizar ambas as bordas da interrupção não é uma boa política, pois devido às imperfeições na construção mecânica dos codificadores, a medição da variação dessa grandeza em relação ao tempo torna-se potencialmente ruidosa. Na Figura 3.14, tem-se um exemplo da medida de posição angular do codificador incremental utilizado no pendubot. Essa medida foi amostrada dentro da rotina de controle que está sendo executada a cada 10ms. É interessante observar que a primeira passagem pelo *index* ocorre na posição angular em torno de 200 incrementos. As demais ocorrem nos valores ± 900 incrementos, valores máximos desse codificador incremental. Experimentalmente, também foi observado que ocorreu um escorregamento na medição de incrementos devido à banda morta entre as fases do codificador.

⁷74LS74/DM54LS74A/DM74LS74A Dual Positive-Edge-Triggered D Flip-Flops with Preset, Clear and Complementary Outputs - National Semiconductor Corp.

⁸74LS14/DM74LS14 Hex Inverters with Schmitt Trigger Inputs - National Semiconductor Corp.

Tal problema foi solucionado utilizando uma máquina de estados alimentada pelos dois canais do codificador garantindo assim seu correto incremento.

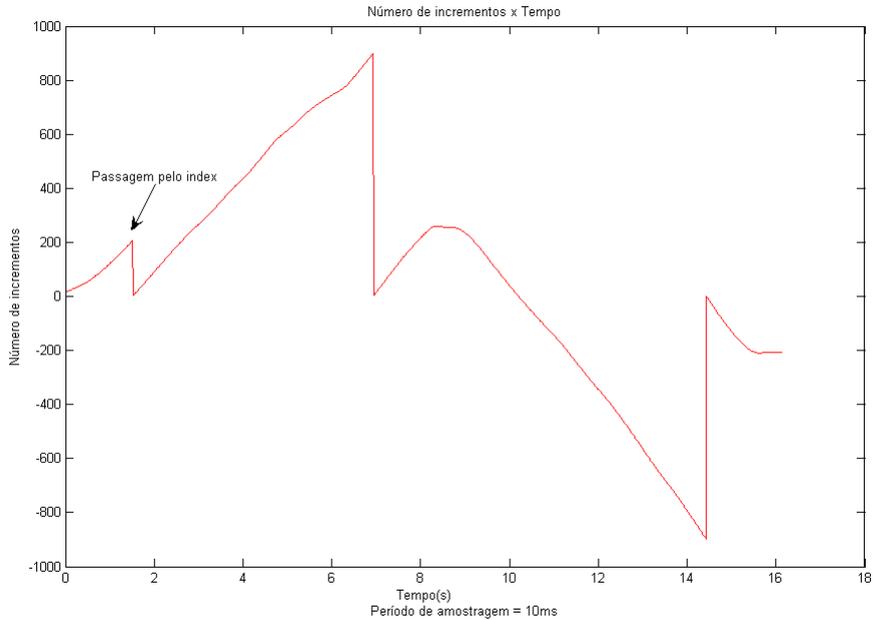


Figura 3.14: Gráfico de incrementos x tempo.

Uma vez que a granularidade do tempo é muito maior que o número de pulsos da maioria dos atuais codificadores, bastou calcular a frequência incremental do codificador de acordo com a equação 3.13 para obter a variação do número de incrementos com relação ao tempo.

$$F_{inc} = \frac{1}{\Delta Tempo} \quad (3.13)$$

em que $\Delta Tempo$ é o tempo decorrido entre cada incremento do codificador. Na Figura 3.15, tem-se a medida de frequência incremental obtida do gráfico da Figura 3.14. Essa medida foi amostrada na mesma rotina de controle descrita anteriormente. Os ruídos periódicos da Figura 3.15 ocorrem em virtude de uma imperfeição mecânica das engrenagens do motor do pendubot.

Para medição do ciclo de trabalho, foram utilizados dois registradores do periférico TC. São eles:

- TC_RB: salva o tempo decorrido desde a última borda de descida;
- TC_RA: salva o tempo decorrido entre a borda de descida e a borda de subida;

Na Figura 3.16, tem-se um diagrama ilustrativo:

Dessa forma, para obter o ciclo de trabalho - CT - basta:

$$CT = \frac{V_{TC_RA}}{V_{TC_RB}} \quad (3.14)$$

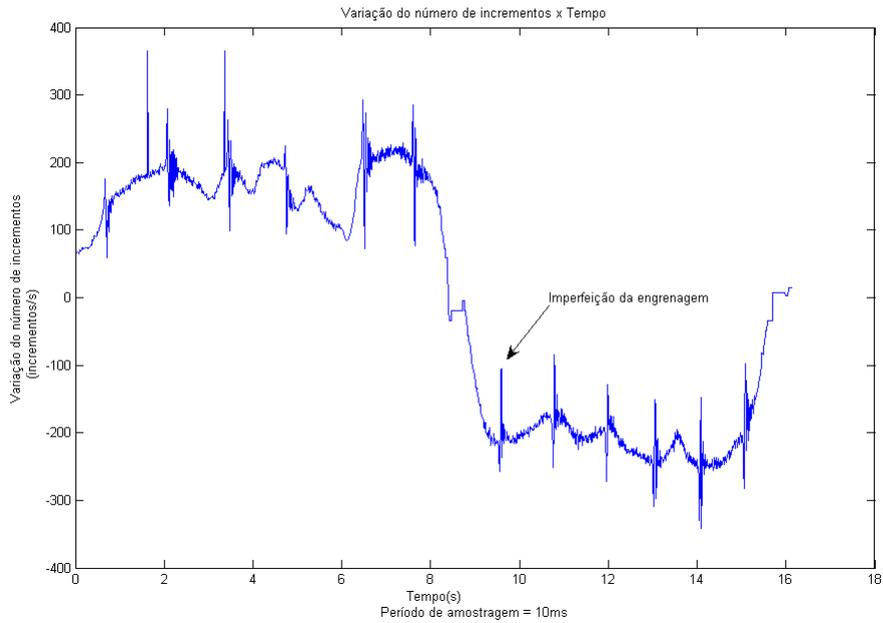


Figura 3.15: Gráfico da velocidade incremental x tempo.

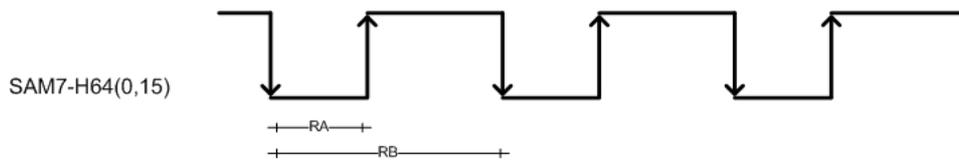


Figura 3.16: Diagrama de medição de ciclo de trabalho

em que V_{TC_RA} e V_{TC_RB} é o valor contido no registrador TC_RA e TC_RB, respectivamente. Vale a pena observar que o sinal SAM7-H64(0,15) está invertido devido à porta isoladora. Na Figura 3.17, é mostrado a medição de ciclo de trabalho do codificador absoluto magnético utilizado no pendubot. Essa medida foi amostrada na mesma rotina de controle previamente descrita. Devido a característica intrínseca do codificador, observa-se um comportamento não linear quando o ciclo de trabalho passa de 0 para 1. Tal comportamento deve ser considerado caso a aplicação utilize essa região de medida.

Para medir a variação do ciclo de trabalho com relação ao tempo - V_{CT} - utilizou-se:

$$V_{CT} = \frac{\Delta CT}{\Delta Tempo} \quad (3.15)$$

em que ΔCT é a variação do ciclo de trabalho e $\Delta Tempo$ é o tempo decorrido entre dois valores consecutivos de ciclo de trabalho. Na Figura 3.18, tem-se o gráfico da variação do ciclo de trabalho pelo tempo obtido do mesmo experimento da Figura 3.17. Em razão da não linearidade anteriormente citada, ocorre um comportamento anômalo no gráfico de velocidade. Para aplicações em que se utiliza essa região não linear, tal comportamento deve ser corrigido, visto que tamanha anomalia pode, por exemplo, ocasionar forte instabilidade em um sistema a ser controlado. Essa correção foi realizada utilizando a equação 3.16 que ajusta os quadrantes dos ângulos entre

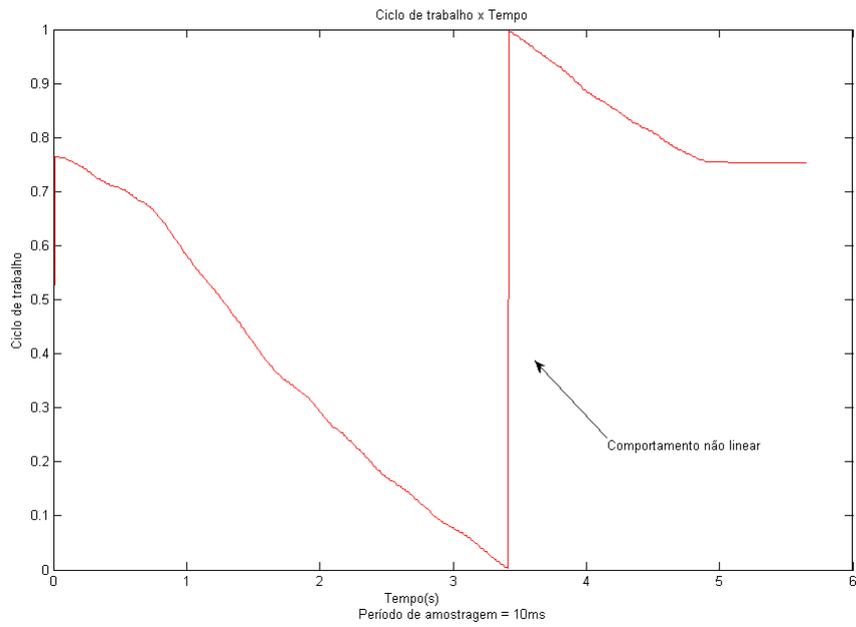


Figura 3.17: Gráfico de ciclo de trabalho x Tempo

$-\pi$ a π .

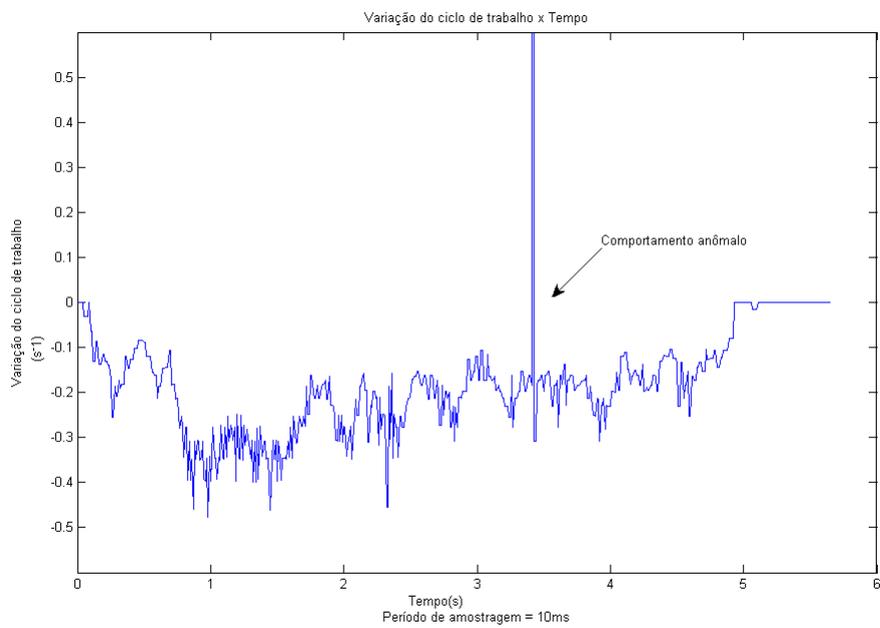


Figura 3.18: Gráfico de velocidade de ciclo de trabalho x Tempo

$$V_{CT} = \frac{\text{atan2}(\sin(\Delta CT), \cos(\Delta CT))}{\Delta \text{Tempo}} \quad (3.16)$$

3.2.5 E/S

O periférico E/S (Entrada/Saída) foi projetado para atender às diversas situações em que necessitassem de acoplamentos digitais. Neste periférico existem 6 canais de entrada e 6 canais de saída que foram implementados da mesma maneira que se encontram no microcontrolador a menos das portas inversoras *schmitt-trigger* 74LS14 e resistores de *pull-up* que conferem maior estabilidade, proteção e imunidade aos ruídos. Nas figuras 3.19 e 3.20, têm-se os circuitos de cada canal.

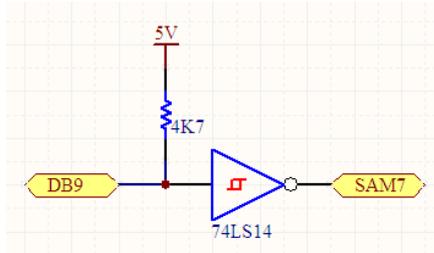


Figura 3.19: Circuito de um canal de entrada digital.

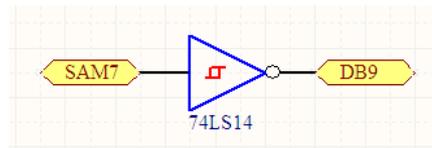


Figura 3.20: Circuito de um canal de saída digital.

3.2.6 Modulação por largura de pulso

Modulação por largura de pulso (MLP) é uma forma de acionamento digital amplamente utilizado em variados tipos de sistema. Para tanto, construiu-se o periférico MLP com 4 canais de saída configuráveis com frequências entre 1KHz a 30KHz. Eletronicamente, o circuito foi projetado de acordo com a Figura 3.20. Para fins de testes, foram geradas quatro sinais modulados por largura de pulso:

- S0: MLP com frequência de 1KHz com ciclo de trabalho de 10%, Figura 3.21;
- S1: MLP com frequência de 1KHz com ciclo de trabalho de 90%, Figura 3.21;
- S2: MLP com frequência de 30KHz com ciclo de trabalho de 10%, Figura 3.22;
- S3: MLP com frequência de 30KHz com ciclo de trabalho de 90%, Figura 3.22;

Em ambos os sinais da Figura 3.21, nota-se que a frequência está de acordo com o esperado. Na Figura 3.22, nota-se uma diferença na frequência medida pelo osciloscópio devido à escala escolhida. Para obter a medida correta, observe logo abaixo de V_{lo} da Figura 3.22. O ciclo de trabalho da Figura 3.21 e da Figura 3.22 contém diferenças nas medidas esperadas que também ocorrem devido à escolha de escala. Outro fato importante é o tempo de descida dos quatro sinais que está em torno de $8\mu s$. Dependendo do sistema no qual o periférico MLP estiver acoplado, essa limitação pode ser considerável. Observe a Figura 3.23 para obter o tempo de descida dos sinais S2 e S3. Vale ressaltar que os sinais do periférico MLP são de baixa potência. Caso o MLP seja acoplado diretamente a uma carga de baixa impedância, o valor máximo (V_{hi}) e mínimo (V_{lo}) devem ser considerados.

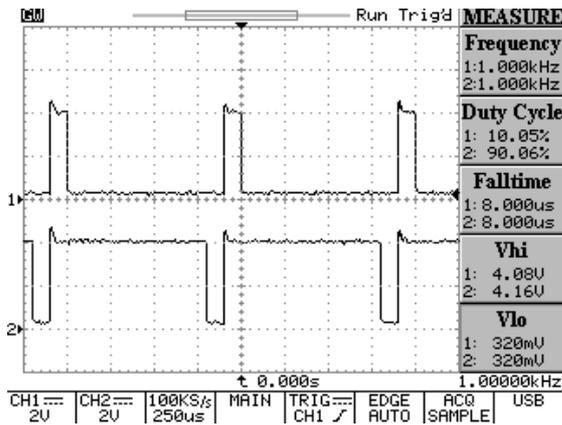


Figura 3.21: Sinais S0 e S1

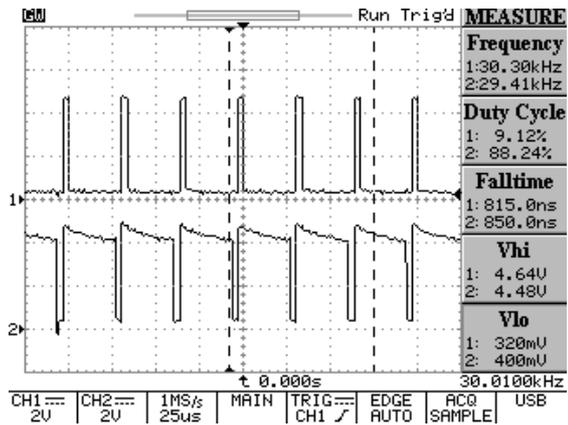


Figura 3.22: Sinais S2 e S3

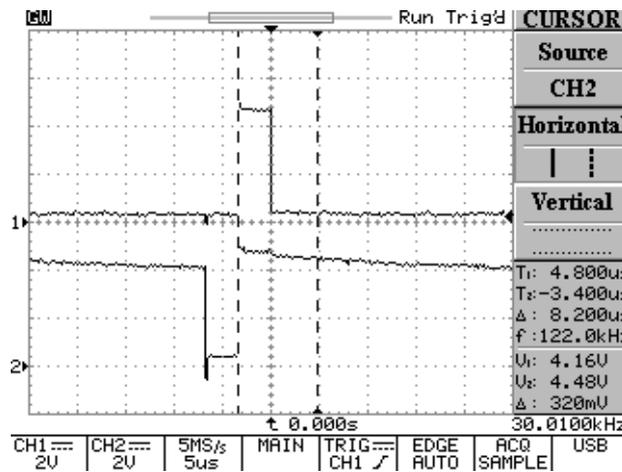


Figura 3.23: *Fall-time* dos sinais S2 e S3

3.2.7 Biblioteca de funções

A biblioteca de funções desenvolvida para o módulo de controle é constituída de diversos arquivos construídos modularmente. Cada periférico do módulo possui um arquivo fonte (.C) e outro arquivo cabeçalho (.H) e cada arquivo fonte possui funções de inicialização assim como funções de entrada e saída dos periféricos. Exemplificando, têm-se os arquivos adc.c e adc.h, responsáveis pelo funcionamento do periférico Conversor A/D, que possui as seguintes funções:

- inicializaADC(): inicializa o periférico A/D;
- converterADC(): faz a conversão dos valores analógicos em digital.

Para implementação da rotina de controle, elaboraram-se dois arquivos, controle.c e controle.h, nos quais é possível implementar as técnicas de controle, selecionar as variáveis que serão enviadas à interface gráfica, selecionar o período de controle etc. Observe a Figura 3.24 e Figura 3.25 que detalhada um exemplo de controle para o pendubot e a seleção de variáveis que serão enviadas para a interface gráfica, respectivamente.

```

// #### INTERRUPTO DE CONTROLE #### //
__ramfunc void controle(void)
{
    setSaidaDigital(SAIDADIGI);
    nControle += PERIODOCONTROLE/1000.0;
    if(nControle > CONTROLETIMEOUT) flagStartExperimento = 0;

    // #### CAPTURA DOS DADOS #### //
    // Offset capturado experimentalmente
    teta1 = (((double)contadorPulsos/1800.0)*DOISPI - 3.2568) - PISOBDOIS;
    teta2 = (dutyCyclen0 - 1.61089 + PISOBDOIS) - PISOBDOIS;

    omega1 = (velocidadeIncremental1/2048.0)*DOISPI;
    omega2 = velocidadeDutyCycl1e0;

    // #### FILTROS DE MEDICAO #### //
    teta1Fil = butterworth(teta1);
    teta2Fil = butterworth(teta2);
    omega1Fil = butterworth(omega1);
    omega2Fil = butterworth(omega2);

    // #### ROTINA DE CONTROLE #### //
    u = -(Kot0*teta1 + Kot1*omega1 + Kot2*teta2 + Kot3*omega2);
    acionaPlanta(u);

    // #### ENVIIO DOS DADOS #### //
    enviarVarComunicacao();

    // Reconhecimento o tratamento da interrupcao
    AT91C_BASE_PITC->PITC_PIVR;
    AT91C_BASE_PITC->PITC_PISR;
    clearSaidaDigital(SAIDADIGI);
}

```

Figura 3.24: Implementação de controle

```

// Período da rotina de controle em ms
#define PERIODOCONTROLE 10.0

// Tempo de execucao maxima da rotina de controle em s
#define CONTROLETIMEOUT (2*60)

// Definicao de variaveis de controle
#define teta1 varComunicacao[0]
#define NOMEVARA "teta1"
#define HABILITAENVIIOA

#define omega1 varComunicacao[1]
#define NOMEVARB "omega1"
#define HABILITAENVIIOB

#define teta2 varComunicacao[2]
#define NOMEVARC "teta2"
#define HABILITAENVIIOC

#define omega2 varComunicacao[3]
#define NOMEVARD "omega2"
#define HABILITAENVIIOD

#define u varComunicacao[4]
#define NOMEVARE "u"
//#define HABILITAENVIIOE

#define Kot0 varComunicacao[5]
#define NOMEVARF "Kot0"
//#define HABILITAENVIIOF

```

Figura 3.25: Seleção de variáveis

Para que o arquivo controle.c tenha acesso às variáveis dos periféricos, declaram-se essas variáveis como *volatile* dentro dos arquivos dos periféricos, tais como: adc.c, pwm.c, udp.c etc, e então declaram-se essas variáveis como *extern volatile* dentro do arquivo. Outro arquivo que vale ressaltar é o main.c que é responsável pela inicialização de todos os periféricos do módulo de controle. A biblioteca completa desenvolvida assim como um *template* de aplicação se encontram nos anexos.

3.2.8 USB e JTAG

A interface USB é responsável pela comunicação em tempo real entre o módulo de controle e a interface Matlab[®]. Eletricamente, a USB foi implementada assim como está concebida na placa SAM7-H64. Já em nível de software e no intuito de minimizar a chance de perda de dados, incluíram-se buffers cíclicos de 1KByte na entrada e na saída de dados.

A interface JTAG foi implementada utilizando um cabo do tipo *Macraigor Wiggler*⁹. Tal interface tem a função de realizar a gravação do microcontrolador assim como a depuração do sistema via *software*.

3.3 Comunicação entre Interface gráfica e o módulo de controle

A comunicação realizada entre o módulo de controle e a interface gráfica está separada de duas formas: comunicação externa que utiliza um servidor para realizar a troca de informações entre o computador e o módulo, comunicação interna que é responsável pela comunicação entre os aplicativos dentro do computador. O diagrama de comunicação do sistema encontra-se ilustrado na Figura 3.26.

⁹<http://www.macraigor.com/wiggler.htm>

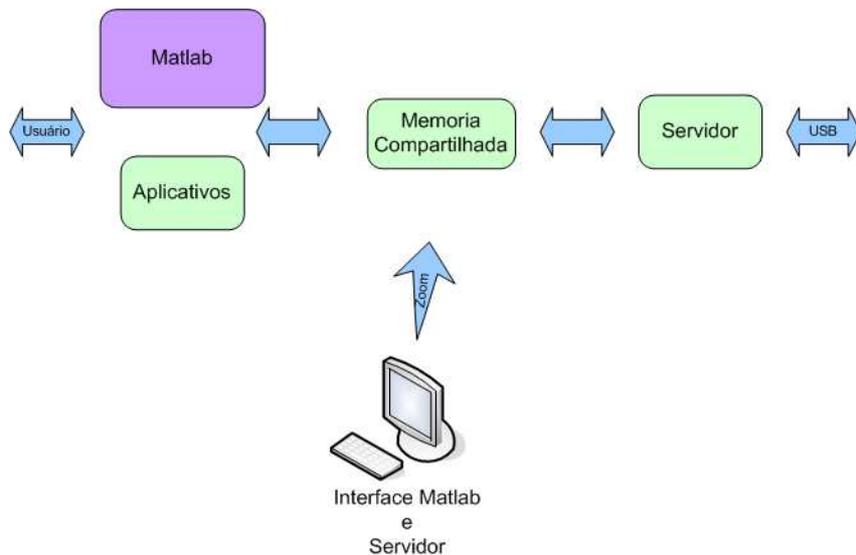


Figura 3.26: Diagrama de comunicação

Para a comunicação externa, escolheu-se o barramento USB. O motivo dessa escolha foi a alta taxa de transferência de dados e sua disponibilidade no microcontrolador escolhido. Para a troca de informações entre o servidor e módulo de controle, modificou-se um *driver* disponibilizado pelo fabricante do microcontrolador. Esse *driver* foi desenvolvido somente para o sistema operacional Windows.

A comunicação entre processos (aplicativos e servidor) ocorre utilizando memória compartilhada. Uma região da memória do computador é inicializada por um aplicativo e, de acordo com sua configuração, o sistema operacional disponibiliza o acesso a essa memória para outros aplicativos. Criaram-se três regiões de memória: uma permite o servidor disponibilizar os dados vindos do módulo de controle, outra permite que os aplicativos, neste caso o Matlab[®], enviem os dados para o módulo de controle e uma terceira região permite a troca de estado entre o servidor e os aplicativos. O algoritmo de comunicação interna é elaborado no intuito de evitar conflitos de acesso à memória.

O papel principal do servidor é colocar na memória compartilhada os dados proveniente do módulo de controle e transferir os dados depositados pelos aplicativos para esse módulo. Inicializado o sistema, o servidor mantém-se em um laço até que uma mensagem para desligá-lo seja colocada na região de estados. Dentro desse laço, o servidor verifica se existe mensagens para transmitir ao módulo de controle. Caso exista tal mensagem, o servidor a transmite. Logo em seguida, ele verifica se existem dados provenientes desse módulo de controle. Em caso afirmativo, o servidor coloca esses dados na região de memória compartilhada correspondente. Dessa forma, o servidor torna-se transparente na implementação do protocolo de comunicação.

Para obter um protocolo mais robusto, cada comando enviado ao módulo de controle gera resposta positiva ou negativa e, se necessário, um valor de retorno. Para aumentar o desempenho do sistema, quando o módulo de controle está em modo *running*, somente é possível enviar o comando de finalização. Definiu-se o formato das mensagens como:

#comando, varivel, valor;

Caso a mensagem não necessite de alguns dos campos, não é necessário enviar tais campos e nem o seu separador de mensagens anterior (,). Na Tabela 3.1, têm-se os comandos implementados e suas possíveis respostas.

Tabela 3.1: Mensagens de comunicação

Função	Mensagem	Resposta 1	Resposta 2
Iniciar o experimento	#j;	#o;	#e;
Finalizar o experimento	#k;	#o;	#e;
Pedido das variáveis de comunicação	#t;	#o;	#e;
Set Valor 6.7980 para variável A	#s,A,6.7980;	#o;	#e;
Get Valor da variável B	#g,B;	#r,B,X.XXXX;#o;	#e;

3.4 Interface Matlab[®]

Uma vez que Matlab[®] é uma excelente ferramenta para projetos de controle, decidiu-se utilizá-lo como interface gráfica para o sistema supervisorio desenvolvido. Como foi dito na seção anterior, os aplicativos depositam seus dados em um região de memória compartilhada, porém o Matlab[®] não possui funções que fazem acesso direto a esse tipo de memória. Dessa forma, utilizou-se uma classe específica de funções (C-mex) que permite execução de códigos na linguagem C e C++ dentro do ambiente Matlab[®]. As funções criadas para esse tipo de comunicação são:

- servidorUSB: Inicializa a memória compartilhada para troca de estado;
- escreverEstado: Altera o estado do sistema;
- lerEstado: Lê o estado do sistema;
- lerUSB: Lê os novos dados disponíveis na memória;
- escreverUSB: Escreve os dados a serem transmitidos para memória.

De posse dos dados, criaram-se funções para interpretá-los. Uma dessas funções - responderUSB - é responsável pela geração e interpretação dos comandos quando o módulo de controle está em modo *standby*, outra função - maqUSB - é a máquina de estados que interpreta os comandos vindos do módulo de controle quando ele está em modo *running* e a última função - maqTabela - é responsável pela troca dos nomes das variáveis pré definidas dentro do módulo de controle. Como pode ser visto, o sistema é composto por várias funções que, para seu funcionamento correto, devem ser executadas repetidamente. Para tanto, criou-se uma janela amigável, Figura 3.27, de maneira que o usuário se preocupasse, exclusivamente, com o experimento em estudo. A janela

é composta pelos botões de controle, parte de comunicação, visualização rápida dos dados e uma parte informativa.



Figura 3.27: Janela do sistema



Figura 3.28: Parte de comunicação

Na parte de comunicação, Figura 3.28, têm-se as opções de comando, variável e valor, que são suficientes para ajustar e obter o valor das variáveis do experimento. Para usuários mais experientes, tem-se a linha de comando na qual eles podem enviar outros comandos ao módulo de controle. Vale ressaltar que, no momento que o sistema é ligado, as variáveis escolhidas pelo usuário são carregadas na janela automaticamente. Logo abaixo da seção de comunicação, tem-se uma parte informativa que mostra o que está sendo processado pelo programa ou uma resposta às suas mensagens de comunicação, Figura 3.28. Também na Figura 3.28, pode-se visualizar uma

mensagem informativa. Na parte inicial, tem-se o comando completo que foi enviado ao módulo de controle. Na segunda parte da informação, tem-se a resposta do sistema(OK).



Figura 3.29: Botões de controle

Os botões de controle, Figura 3.29, foram criados para facilitar a utilização da janela. Vale ressaltar que ao acionar o botão ligar, visualizado na Figura 3.27, imediatamente o programa chama o servidor. Assim que o sistema é ligado, uma mensagem informativa é passada ao usuário e esse botão torna-se o botão de desligar. Os botões iniciar e finalizar só podem ser usados separadamente, por isso, quando um está habilitado, o outro fica desabilitado. Por último, tem-se a parte de visualização rápida dos dados, Figura 3.30, em que o usuário pode visualizar os dados que foram enviados durante o experimento. Para ser uma ferramenta funcional, basta selecionar as variáveis no campo ao lado do gráfico. Caso necessite de uma análise mais profunda, o usuário pode salvar os dados e carregá-los no próprio Matlab[®].

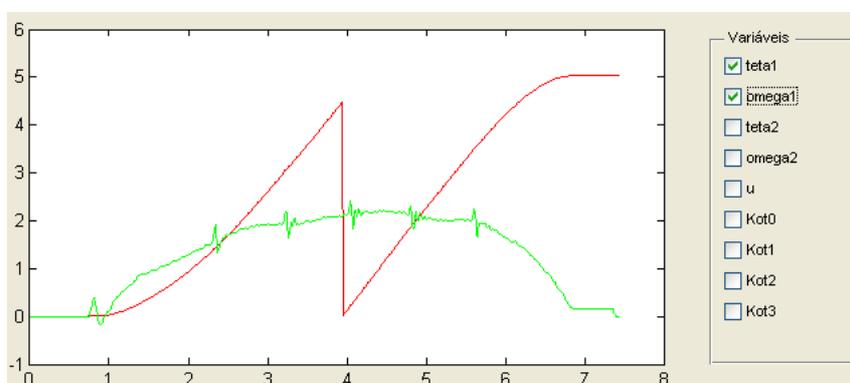


Figura 3.30: Parte de visualização rápida

3.5 Pendubot

Sabendo que não existe uma única configuração que permita implementar o sistema do pendubot, foi selecionada uma configuração capaz de atender aos requisitos físicos do modelo, ser economicamente viável e que fosse passível de ser construída na oficina mecânica da UnB. Utilizando as simulações em [5], foi possível obter os principais requisitos do sistema. Na Figura 3.31, tem-se o diagrama de funcionamento do pendubot.

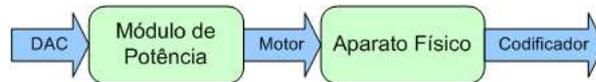


Figura 3.31: Diagrama do pendubot

3.5.1 Motor

Após serem realizadas diversas simulações, duas características principais foram identificadas: torque de 5 N.m utilizando um fator de segurança 1.5 e tensão de alimentação de 24V. Foi utilizado também um fator de segurança 2 para o valor do torque devido ao fato de ser um projeto pioneiro. A limitação de 24V deve-se ao fato da dificuldade da elaboração de uma fonte linear de alta tensão. Em posse dessas características, encontrou-se o motor GM-14902S020 com tensão de alimentação de 24V e torque de $3.5N.m^{10}$ que mais se aproximou das especificações requeridas, Figura 3.32.



Figura 3.32: Motor GM-14902S020

Tabela 3.2: Valores do motor

Característica	Valor	Unidade
T_{em}^{11}	3.5	Nm
k_T	5.51E-2	Nm/A
i_a^{12}	13.9	A
k_e	5.51E-2	V/(rad/s)
ω_m^{13}	60	RPM
R_a	1.73	Ohms
L_a	2.54	mH
V_t^{14}	24	V

3.5.2 Módulo de potência

Os sinais oriundos do módulo de controle são insuficientes para o acionamento do motor, portanto faz-se necessário uma etapa de potência. Essa etapa é executada na módulo de potência. Na Figura 3.33 e 3.34 , tem-se fotos do Módulo de Potência.

Vale enfatizar que vários projetos de acionadores lineares foram testados. Mesmo sabendo das dificuldades, tentou-se implementar uma fonte de corrente analógica para correntes na ordem de

¹⁰GM14902S020 Lo-Cog DC Gearmotor - Pittman Express



Figura 3.33: Módulo de potência



Figura 3.34: Módulo de potência aberto

10A e também uma fonte de tensão regulável de 0 a 30V que suportassem correntes de 8A, mas todos sem sucesso. A grande dificuldade encontrada foi a dissipação de potência dos componentes. Devido ao curto prazo, decidiu-se utilizar o Servo-amplificador PWM[7] funcionando no modo fonte de corrente controlada por tensão. Uma modificação feita é a adição de um regulador de 12V entre a alimentação da parte de potência e o sinal de alimentação do circuito integrado IR2104, pois a alimentação máxima desse circuito é de 20V e a alimentação da parte de potência é de 34V.

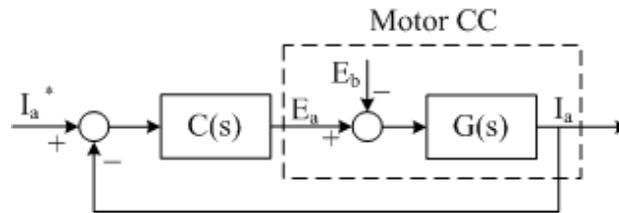


Figura 3.35: Diagrama blocos do sistema

A escolha dos parâmetros P1, P2, P3, G1 e G2 do controlador PI interno da placa foi feita da seguinte maneira. Utilizando os parâmetros fornecido pelo fabricante do motor, encontrou-se a função de transferência

$$G(s) = \frac{1}{0.00254s + 1.73} = \frac{I_a(s)}{V_a(s)} \quad (3.17)$$

A planta $G(s)$ é do tipo zero. Como atuação no motor é feita por E_a , Figura 3.35, pode-se considerar E_b como uma entrada interferente. Utilizando um controlador PI, garante-se erro nulo em regime permanente para uma entrada degrau. A função de transferência do controlador analógico é:

$$C(s) = K_p + \frac{K_i}{s} \quad (3.18)$$

Utilizando a ferramenta *Control and Estimation Tools Manager* do Matlab[®], fez-se a alocação do pólo e do zero nas posições desejadas. O LGR (Lugar Geométrico das Raízes) do sistema encontra-se na Figura 3.37. Na Figura 3.36 e 3.38, tem-se a resposta a uma entrada degrau do sistema em malha aberta e fechada, respectivamente.

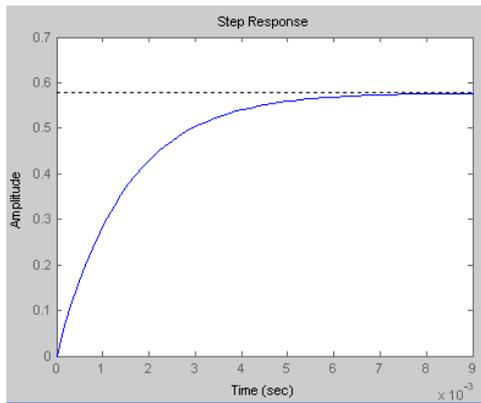


Figura 3.36: Resposta do sistema em malha aberta

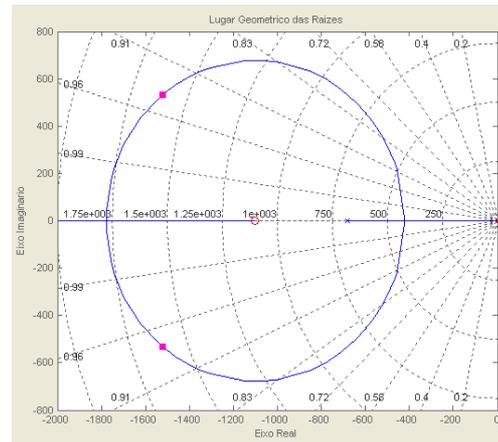


Figura 3.37: LGR do sistema

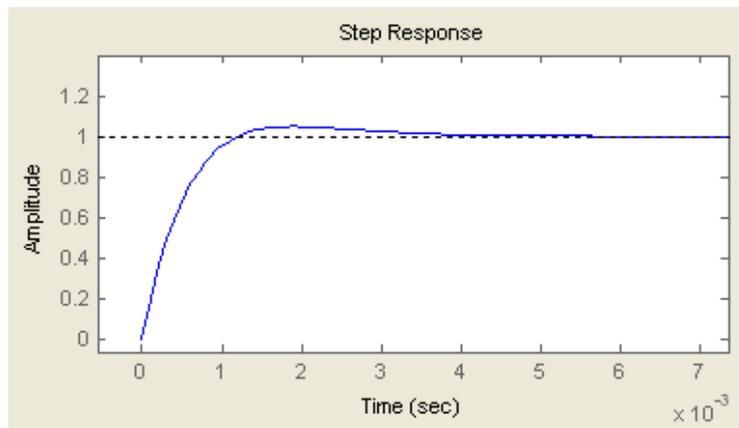


Figura 3.38: Resposta do sistema em malha fechada

de posse dos valores do controlador fornecido pelo Matlab[®], encontrou-se

$$K_p = 5 \quad K_i = 1000$$

de posse dos valores acima, e empregando-se as fórmulas de [7], encontraram-se os valores para

$$\begin{aligned} P1 &= 3,8K\Omega & P2 &= 0 & P3 &= 10K\Omega \\ P4 &= 47nF & G1 &= 2,22K\Omega & G2 &= 2,2K\Omega \end{aligned}$$

Outra vantagem do Servo-amplificador PWM é que ele possui uma saída de tensão proporcional à corrente. Para o funcionamento desse Servo-amplificador PWM, basta adicionar uma tensão de referência e aplicar uma tensão inferior a 3.5V no sinal de Habilitação. A relação tensão de referência versus corrente aplicada no motor está presente no gráfico 3.39.

É utilizado conector do tipo BNC (British Naval Connector) para os terminais de entrada e para o terminal de saída de tensão proporcional a corrente. O conector usado para o motor é do tipo parafusado. Por medidas de segurança, se ambos os conectores de entrada do módulo de potência estiverem desconectados o motor não entrará em funcionamento. Os cabos do motor também podem ser desconectados do módulo de potência. Um botão foi adicionado ao módulo

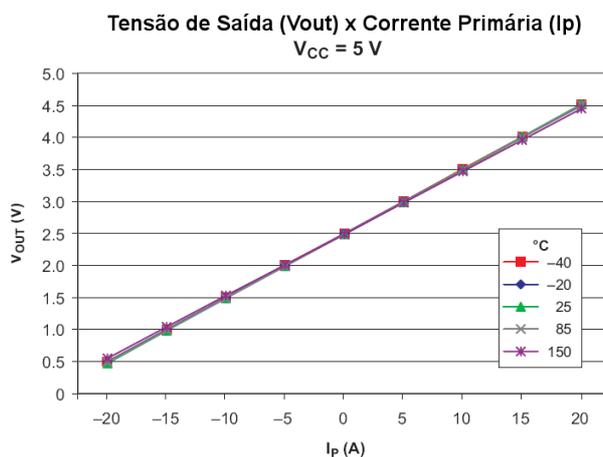


Figura 3.39: Gráfico Tensão referência *versus* Corrente

para ligar e desligar o sistema. Uma luz indicativa também foi acrescentada para indicar a ativação do Servo-amplificador PWM.

3.5.3 Codificador Absoluto e Incremental

Para que fosse possível medir os ângulos de rotação de acordo com a Figura 3.40, foram utilizados dois codificadores. Devido, principalmente, a uma restrição mecânica, foi utilizado um codificador incremental para medir o ângulo de rotação do eixo do motor (θ_1). Para medir o ângulo de rotação de uma barra em relação à outra (θ_2) foi utilizado um codificador absoluto. A escolha desses codificadores baseou-se, substancialmente, em requisitos de projeto e disponibilidade. Dessa forma, foi escolhido o codificador incremental óptico E5S-900-394-IHG para medir θ_1 . Tal codificador possui uma resolução de até 0.1° , canal de *index*, montagem flexível e tamanho reduzido¹⁵. Para medir θ_2 , utilizou-se o codificador absoluto magnético MA2-P que possui uma resolução de até 0.35° (10 bits de precisão) e saída MLP de 1KHz em que a posição do eixo do codificador é proporcional ao ciclo de trabalho¹⁶.

Na Figura 3.41 e 3.42, encontram-se os codificadores instalados.

3.5.4 Base Estrutural

A base foi concebida de maneira que apresentasse duas principais características: estabilidade às forças paralelas ao solo e facilidade de construção. Para tanto, foi utilizado chapas de aço carbono 1020 de 3mm de espessura soldadas no formato L e aparafusada em uma plataforma permitindo que a inércia da base seja muito maior que a dinâmica restante do sistema. Dessa forma, foi possível minimizar a significância de suas vibrações no momento que o pêndulo estiver em movimento. Na Figura 3.43, tem-se o desenho da base estrutural assim com uma foto de sua implementação, Figura 3.44.

Utilizando a configuração acima, foi possível obter as características das barras que compõem

¹⁵E5-Optical Kit Encoder - US Digital

¹⁶MA2-Miniature Absolute Magnetic Shaft Encoder - US Digital

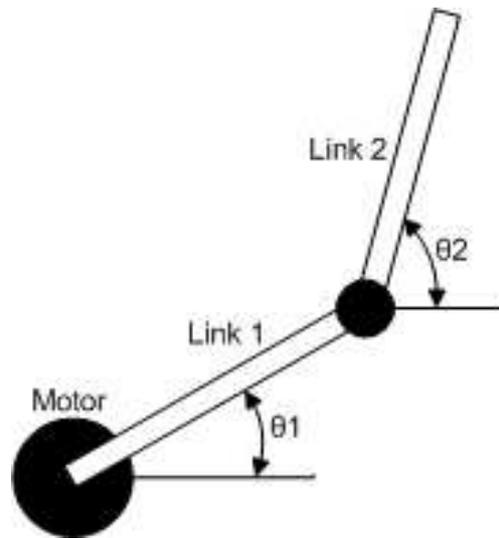


Figura 3.40: Ângulos θ_1 e θ_2 do pendubot



Figura 3.41: Codificador óptico incremental E5S



Figura 3.42: Codificador magnético absoluto MA2-P

o sistema, são elas: massa reduzida, dimensões apropriadas que garantissem apenas um grau de liberdade e projeto da segunda barra permitindo o acoplamento de uma massa móvel. Para tanto, foram escolhidas barras de alumínio 6063 duro de 10mm de espessura com furos passante a fim de reduzir sua massa. Depois de terem sido usinadas, foi possível obter suas massas finais utilizando uma balança digital. Essas massas foram de aproximadamente 162.32g para o link1 e 71.27g para o link 2. Na Figura 3.45, têm-se o desenhos das barras implementas.

Dada a configuração de motor, barras e sensores, os acoplamentos foram elaborados de maneira que pudessem unir as partes da estrutura e garantir o seu correto funcionamento. Na Figura 3.47, tem-se o desenho dos acoplamentos.

3.6 Sistema supervisório

Para validação do sistema supervisório, alguns experimentos foram realizados. Um desses experimentos foi escolhido para mostrar o funcionamento do sistema. Primeiramente, programou-

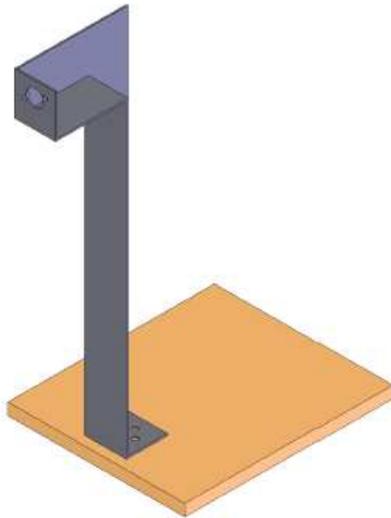


Figura 3.43: Desenho da base



Figura 3.44: Foto da base



Figura 3.45: Link1



Figura 3.46: Link2

se o módulo de controle para a cada $10ms$ coletar os dados dos sensores, transformá-los em medidas de posição e velocidade, aplicar um filtro digital de Butterworth a essas medidas e por fim enviá-los ao computador. Após essa programação, conectou-se o cabo de comunicação USB e iniciou-se a interface gráfica. Acionou-se o botão ligar e o servidor inicializou a comunicação com o módulo de controle. As variáveis definidas pelo usuário para comunicação em modo *standby* e em modo *running* foram identificadas pela interface e tornaram-se disponíveis ao usuário.

Em modo *standby*, o primeiro teste foi o envio de uma instrução pela linha de comando. Já o segundo teste foi utilizar o comando *set* das opções da janela escolhendo uma variável e configurando seu novo valor. O último teste foi utilizar o comando *get* escolhendo outra variável e verificando seu valor de retorno. Ambos os testes retornaram os valores esperados, o comando OK para as duas primeiras instruções e o valor de retorno seguido do comando OK para a terceira instrução.

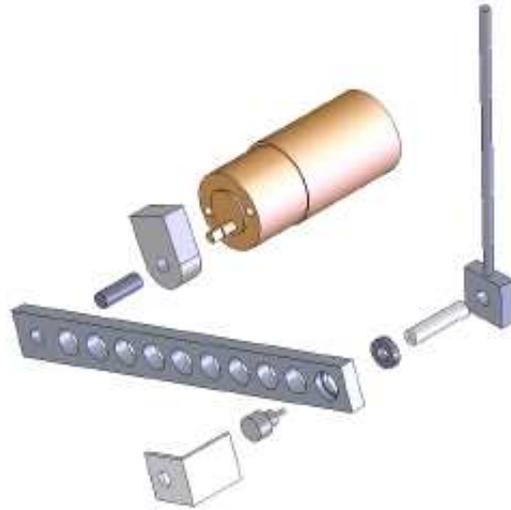


Figura 3.47: Acoplamentos do link 1 e link 2

Para inicializar esse experimento, acionou-se o botão iniciar e movimentou-se manualmente os links 1 e 2. O acionamento manual foi realizado em virtude da limitação física do cabo do codificador do link2 que pode girar no máximo 2 voltas no mesmo sentido. Após clicar no botão finalizar, o experimento foi finalizado e os botões de controle para visualização e armazenamento dos dados foram liberados. Os gráficos resultantes das variáveis de posição e velocidade angular do sensor incremental e do sensor absoluto são encontrados na Figura 3.48 e 3.49. Os valores identificados pela interface foram corretamente salvos no arquivo `matrix_saida.mat` e puderam ser comprovados pelo *log* do servidor. Clicou-se no botão desligar e a comunicação com o módulo de controle foi encerrada, o servidor foi desligado e as regiões de memória compartilhada foram liberadas.

Outro teste que vale ressaltar consistiu no acionamento do motor. Programou-se o módulo de controle para fazer um acionamento com torque constante e suficiente para o pêndulo girar livremente, sem o cabo do codificador do link2. Pelo gráfico da Figura 3.50, verifica-se o comportamento da posição angular do link 1 de acordo com o esperado.

Dentre os outros experimentos realizados, alguns problemas foram detectados, tais como: limite de cinco variáveis para serem transmitidas em modo *running*, quebra de um dente da engrenagem e limite físico do cabo do codificador do link2.



Figura 3.48: Dados do sensor incremental



Figura 3.49: Dados do sensor absoluto

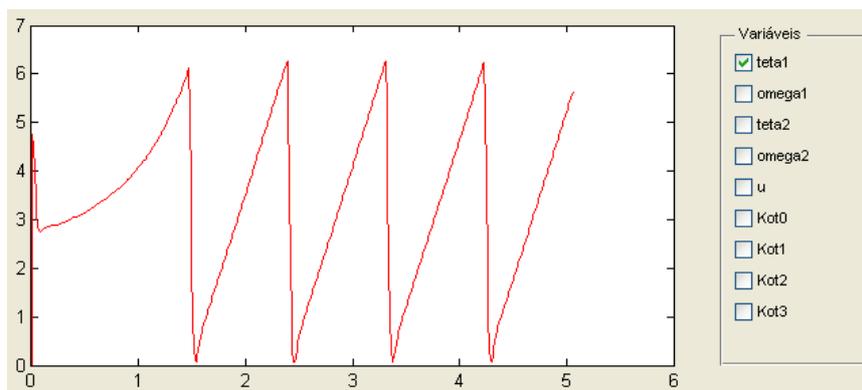


Figura 3.50: Gráfico de posição angulo com torque constante

Capítulo 4

Conclusões

Por meio da análise de experimentos de controle, procurou-se construir uma plataforma que fosse capaz de atender à demanda dos projetos em experimentos de controle projetando um sistema de baixo custo financeiro e de considerável qualidade.

O módulo de controle foi projetado para ser uma plataforma capaz de adquirir sinais, processá-los, fornecer saídas e enviar dados ao computador. O microcontrolador AT91SAM7S64 foi capaz de suprir as principais necessidades do projeto - velocidade de processamento, comunicação e capacidade de armazenamento. Já os diversos periféricos apresentaram algumas limitações como, por exemplo, intervalo de leitura dos conversores A/D e D/A. Essas limitações devem ser consideradas no projeto de controle a ser utilizado. Recomenda-se utilizar filtros apropriados nos sinais adquiridos dos codificadores. Para futuro trabalhos, aconselha-se utilizar amplificadores instrumentais *rail-to-rail*, filtros passa-baixa nos sinais de modulação por largura de pulso, reguladores de tensão de maior precisão e, principalmente, utilizar uma placa de circuito impresso.

No intuito de facilitar o experimento de controle, escolheu-se o Matlab[®] como interface. Uma janela mais confortável ao usuário foi criada procurando deixar o sistema mais simples e transparente possível. Para comunicação entre o módulo de controle e o computador, escolheu-se o padrão USB em virtude de suas vantagens em relação aos outros meios disponíveis. Para tanto, teve-se que adaptar um *driver* e criar um programa servidor para a troca de dados entre ambos. Na janela gráfica, adicionaram-se medidas de segurança para que o usuário não tentasse pular etapas do experimento podendo causar danos ao sistema supervisor. Mensagens informativas também foram acrescentadas para facilitar sua utilização. Para trabalhos futuros, sugerem-se que os gráficos da variáveis previamente selecionadas sejam mostrados em tempo real.

O pendubot foi implementado baseado em simulações previamente realizadas que facilitaram a escolha de alguns componentes na construção física do experimento. Apesar de uma considerável limitação de materiais e processos mecânicos, tal estrutura se mostra como uma plataforma interessante para o estudo de variadas técnicas de controle. Após testar o sistema físico, notou-se que a base da estrutura apresenta instabilidade quando grandes esforços são exigidos. O cabo de transmissão dos dados do codificador absoluto MA2 deve ser evitado uma vez que a rotação contínua do motor é capaz de rompê-lo facilmente. Para trabalhos futuros, propõe-se uma base estrutural mais estável aumentando a massa da base e adicionando vigas de fixação ao suporte de

ação assim como trocar o cabo do codificador absoluto por rádio transmissores.

O módulo de potência utilizando o Servo-amplificador PWM com algumas modificações específicas para o pendubot obteve resultados satisfatórios. Uma melhoria que pode ser realizada é o aumento do intervalo de entrada para a tensão de referência que, atualmente, é de 1,5V a 3,5V, para -5V a 5V.

O sistema supervisorio mostrou ser uma plataforma útil para realização de experimentos de controle. Porém, algumas modificações podem ser realizadas, tais como: aumento do número de variáveis para comunicação em modo *running*, utilizar outras ferramentas gráficas para análise e aquisição dos dados como, por exemplo, LabView[®] e tornar o sistema compatível com *toolboxes*(Simulink[®] , RTW[®]).

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] SPONG, M. Swing up control of the acrobot. *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, v. 3, p. 2356–2361, 1994.
- [2] FURUTA K.; YAMAKITA, M. K. S. Swing up control of inverted pendulum. *Industrial Electronics, Control and Instrumentation, 1991. Proceedings. IECON '91., 1991 International Conference on*, v. 3, p. 2193–2198, 1991.
- [3] SPONG, M.; BLOCK, D. The pendubot: a mechatronic system for control research and education. *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, v. 1, p. 555–556, 1995.
- [4] BECKMAN, E. D.; SANTOS, G. F. *Desenvolvimento de controladores de juntas para prótese robótica de perna*. Julho 2006. Trabalho de graduação do curso de Engenharia Mecatrônica - Departamento de Engenharia Elétrica - UnB.
- [5] GONÇALVES, C. W. P. *Simulação e controle de um pêndulo duplo invertido*. Dezembro 2006. Trabalho de graduação do curso de Engenharia Mecatrônica - Departamento de Engenharia Elétrica - UnB.
- [6] UNIVERSAL Serial Bus Specification. [S.l.], 2000.
- [7] SCANDAROLI, G. G. *Controle adaptativo de juntas para uma prótese robótica de perna*. Julho 2007. Trabalho de graduação do curso de Engenharia Mecatrônica - Departamento de Engenharia Elétrica - UnB.

ANEXOS

I. DIAGRAMAS ESQUEMÁTICOS

Este anexo contém os circuitos completos do módulo de controle. Na Figura I.1, tem-se o circuito dos conversores D/A e A/D. Na Figura I.2, tem-se o circuito dos periféricos E/S, MLP, Codificadores de medição angular e fonte que alimenta a placa. Já na Figura I.3, tem-se o circuito da placa comercial SAM7-H64.

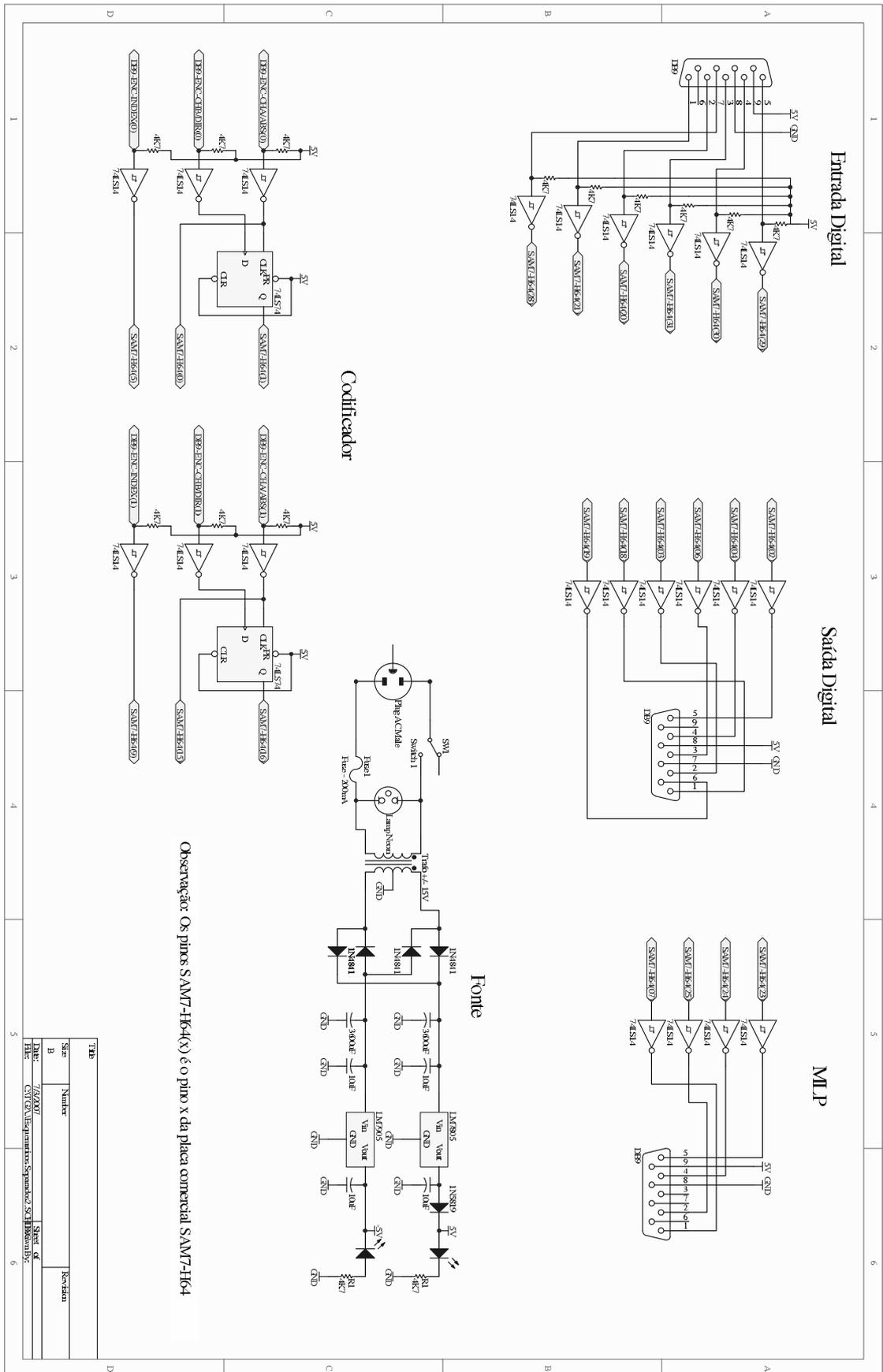


Figura I.2: Circuito dos periféricos IO, MLP, Codificadores e a fonte de alimentação

II. DESENHOS MECÂNICOS

Este anexo contém os desenhos mecânicos do pendubot. Na Tabela II.1, contém os desenhos contidos no anexo. Vale a pena observar que, exceto na Figura II.3, todas as medidas estão em milímetros.

Tabela II.1: Desenhos contidos no anexo

Descrição	Figura
Base do experimento	II.1
Suporte de aço para o motor e barras	II.2
Motor e caixa de redução	II.3
Extensor do eixo do motor	II.4
Primeira Barra: Link1	II.5
Eixo de conexão entre o Link1 e Link2	II.6
Suporte do codificador absoluto	II.7
Segunda Barra: Link2	II.8

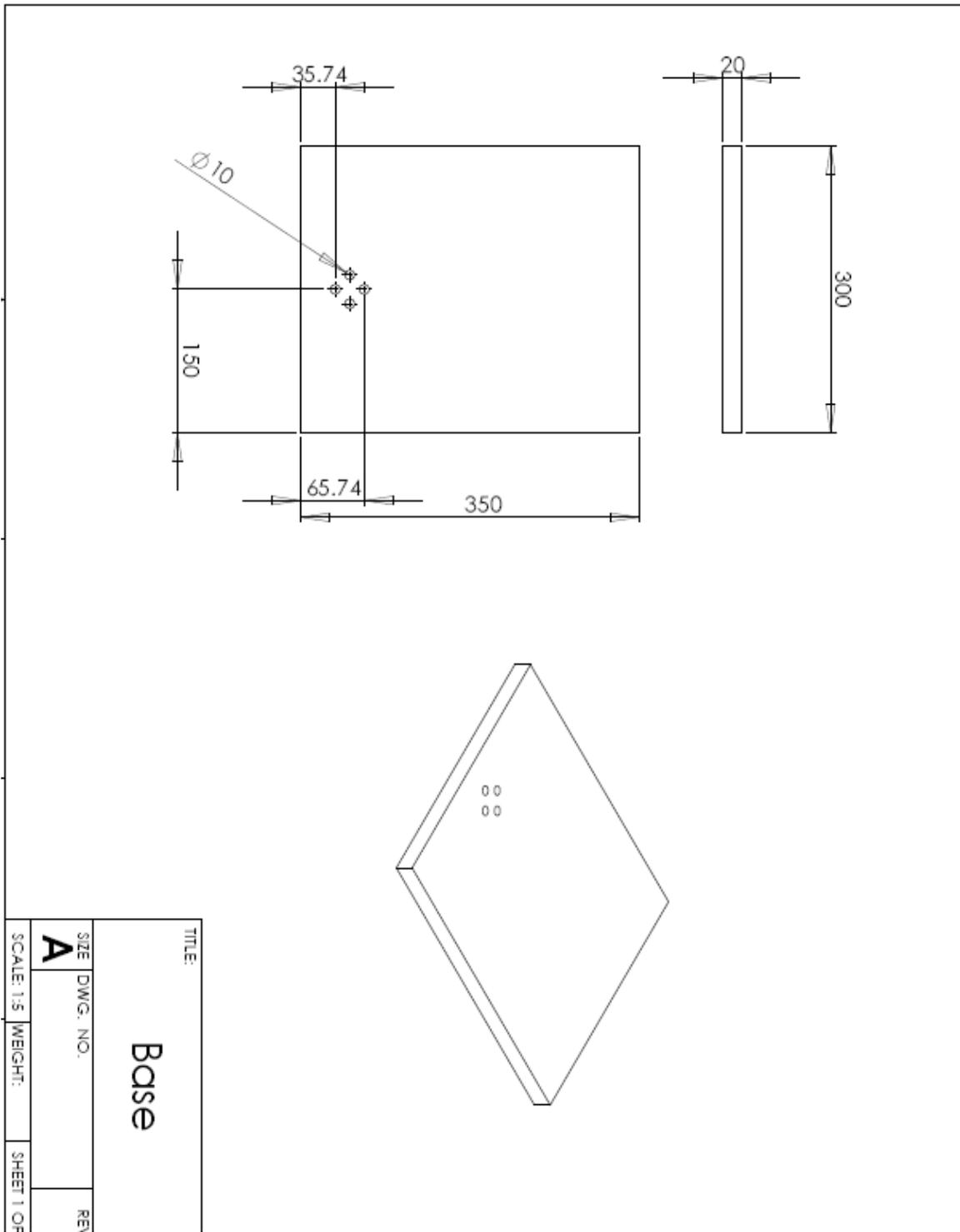


Figura II.1: Base do experimento

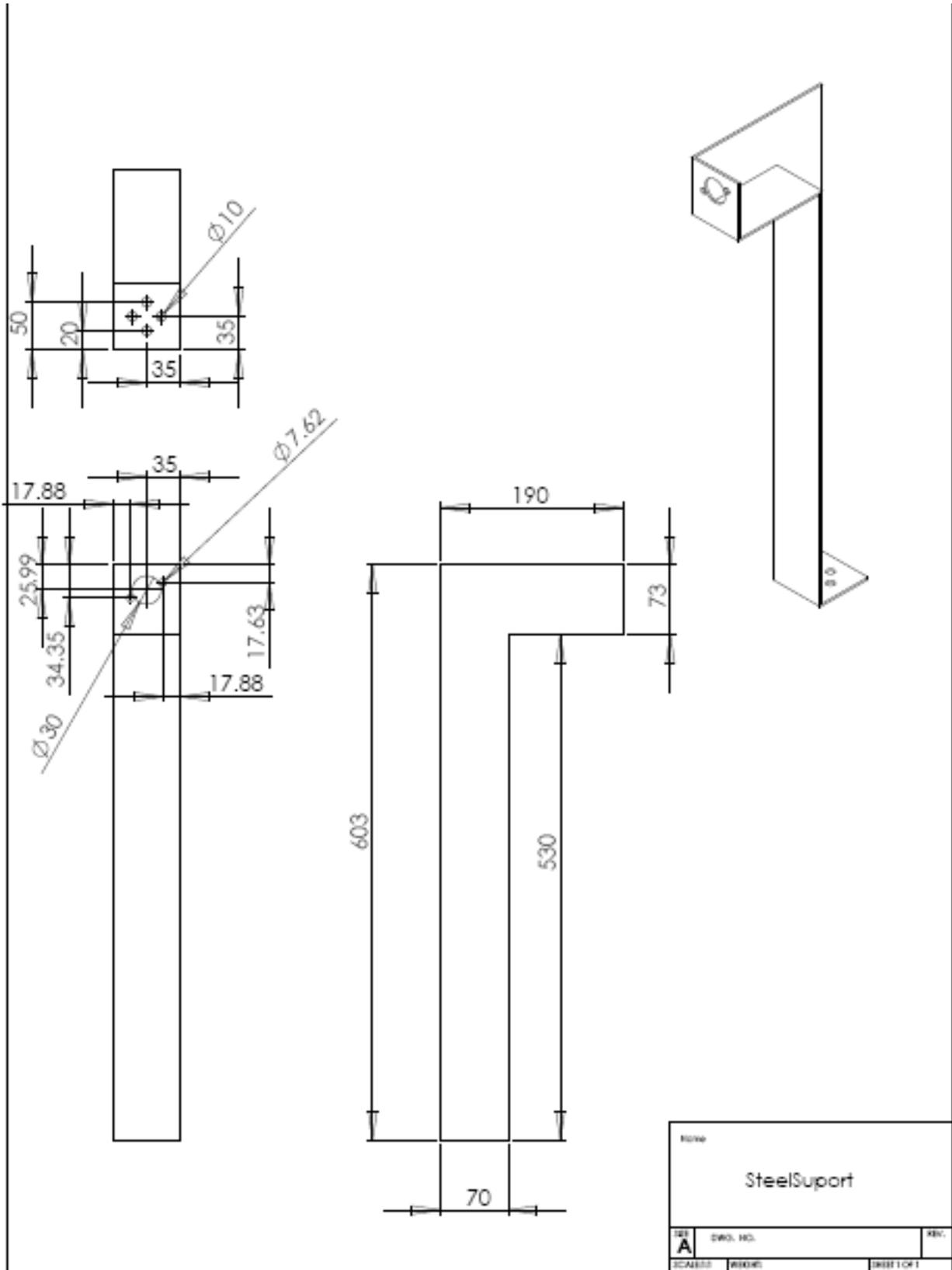


Figura II.2: Suporte de aço para o motor e barras

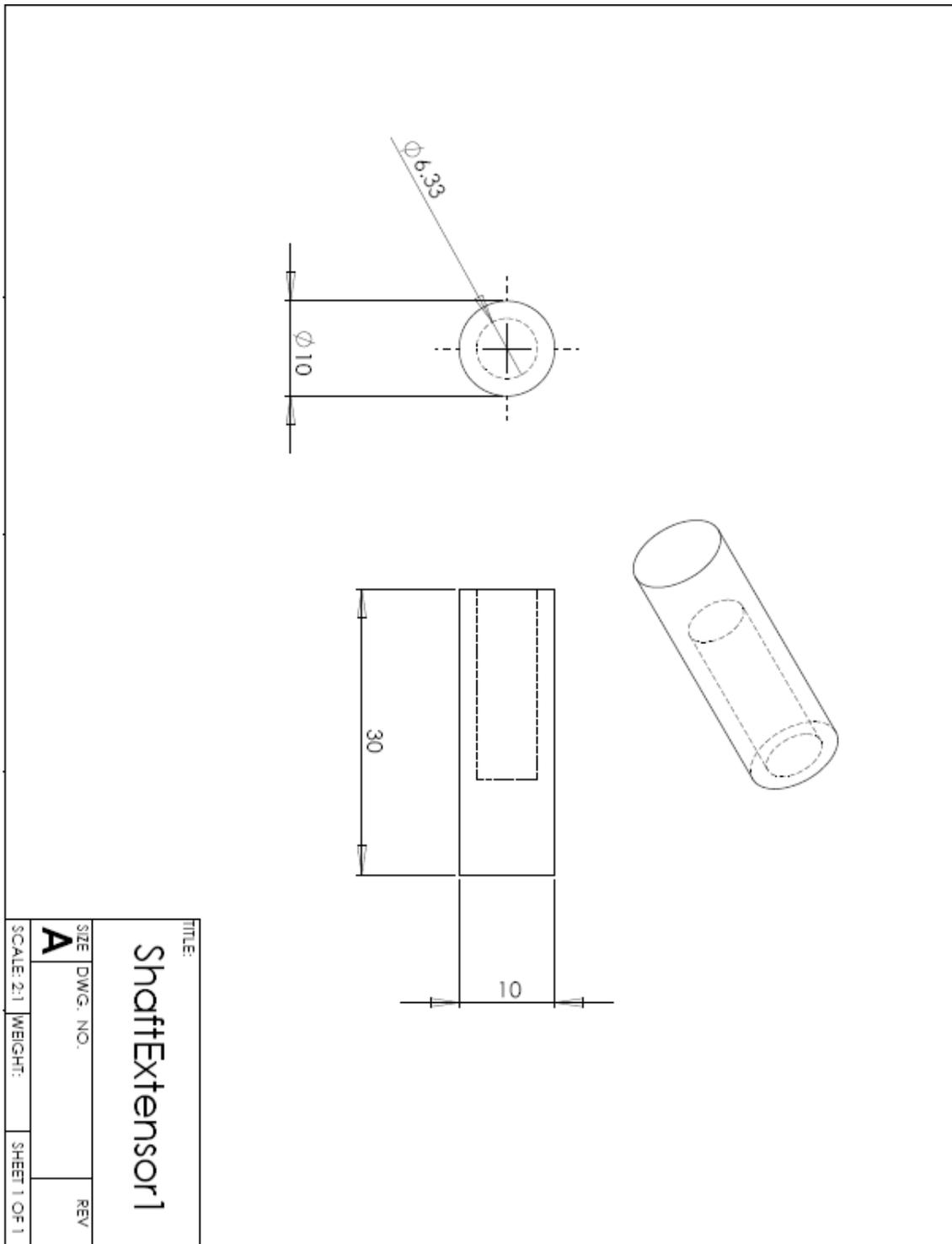


Figura II.4: Extensor do eixo do motor

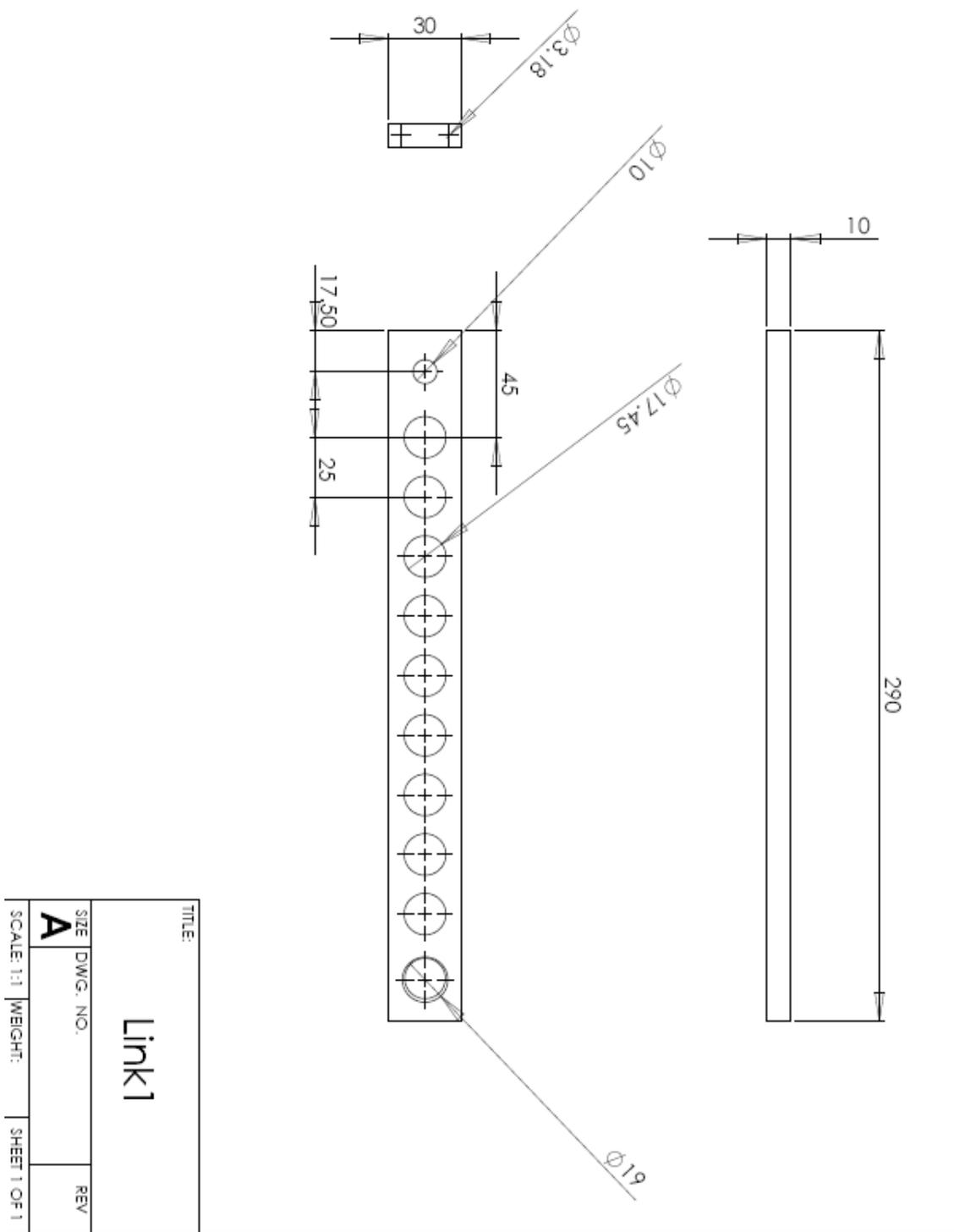


Figura II.5: Primeira Barra: Link1

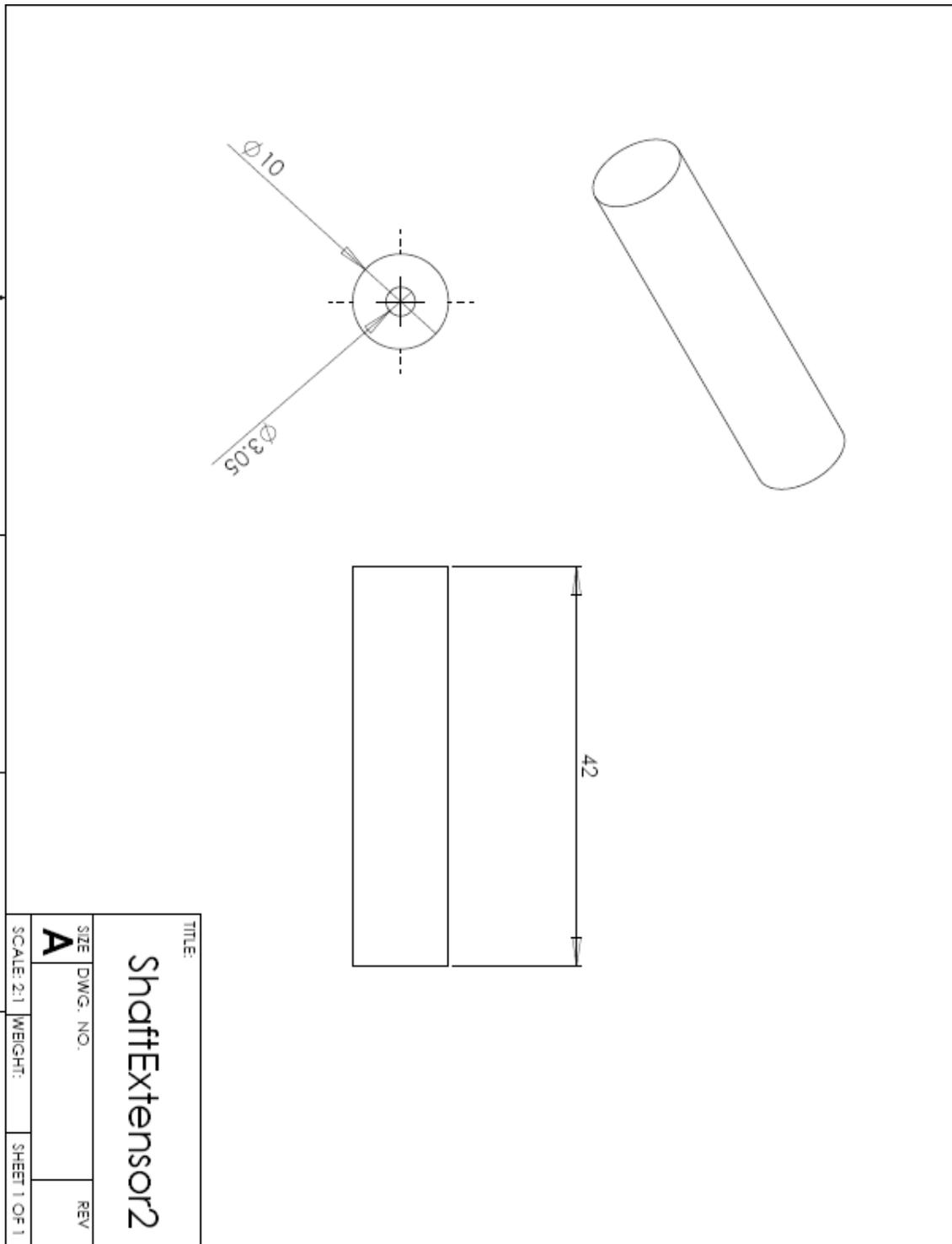


Figura II.6: Eixo de conexão entre o Link1 e Link2

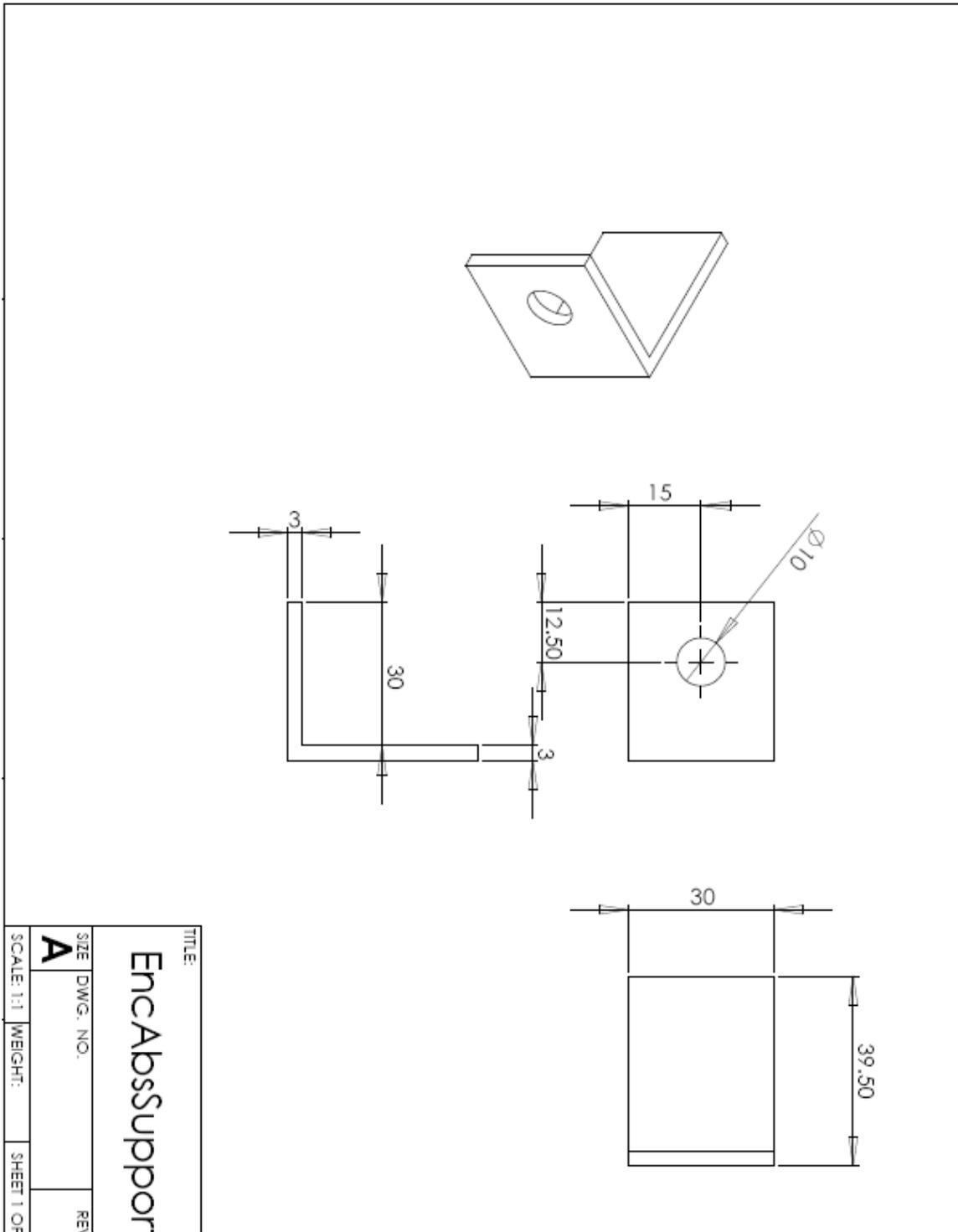


Figura II.7: Suporte do codificador absoluto

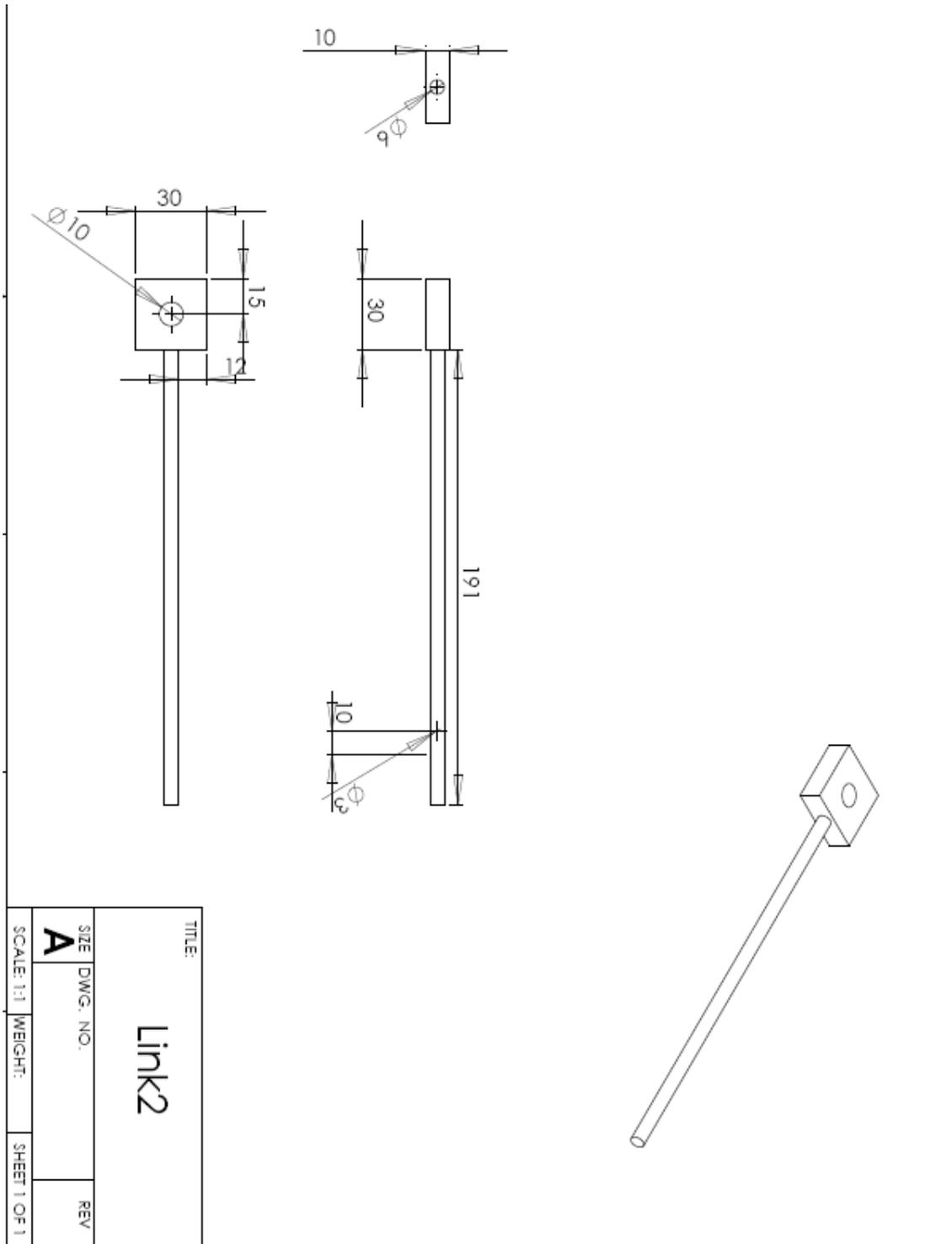


Figura II.8: Segunda Barra: Link2

III. DESCRIÇÃO DO CONTEÚDO DO CD

O CD em anexo contém os códigos de programas, o relatório de projeto final, esquemáticos, desenhos e apresentação utilizada neste projeto. Ele está dividido em seis pasta:

- Códigos do ARM
 - adc.c : configuração e funções do ADC;
 - adc.h : cabeçalho da função adc.c;
 - controle.c : configuração e funções de controle;
 - controle.h : cabeçalho da função controle.c;
 - io.c : configuração e funções de IO;
 - io.h : cabeçalho da função io.c;
 - main.c : inicialização e execução do sistema;
 - pwm.c : configuração e funções do MLP;
 - pwm.h : cabeçalho da função pwm.c;
 - rtt.c : configuração e funções de tempo global do sistema;
 - rtt.h : cabeçalho da função rtt.c;
 - spi.c : configuração e funções do conversor D/A;
 - spi.h : cabeçalho da função spi.c;
 - tc.c : configuração e funções dos codificadores;
 - tc.h : cabeçalho da função tc.c;
 - udp.c : configuração e funções para comunicação USB;
 - udp.h : cabeçalho da função udp.c;
 - prioridade.h : arquivo com prioridades das interrupções do sistema;
 - AT91SAM7S64.h : cabeçalho dos registradores do ARM;
 - Cstartup_SAM7.c : função de inicialização do ARM em C;
 - Cstartup_SAM7.S : função de inicialização do ARM em assembly;
 - sam7.cfg : arquivo de configuração do openocd;
 - sam7.script : script para programação do ARM;
 - arm.pnproj : arquivo de projeto para o WINARM;
 - AT91SAM7S64-ROM.ld : arquivo para criação da imagem da ROM;
 - AT91SAM7S64-RAM.ld : arquivo para criação da imagem da RAM;
 - Makefile : arquivo de configuração para compilação

- Códigos do Matlab®
 - maqUSB.m : interpreta os dados em modo *running*;
 - answerUSB.m: interpreta os dados em modo *standby*;
 - maqTabela.m : inicializa as variáveis de comunicação;
 - LaraControlSystem.m : construção da janela gráfica e interpretação das ações da janela;
 - LaraControlSystem.fig: arquivo do Matlab® para construção da janela;
 - Lara_logo.JPG: logo do LARA que é mostrada na inicialização da janela;
 - readUSB.cpp: código para leitura da memória compartilhada;
 - readUSB.mexw32: arquivo mex gerado a partir de readUSB.cpp;
 - writeUSB.c: código para escrita na memória compartilhada;
 - writeUSB.mexw32: arquivo mex gerado a partir de writeUSB.c;
 - serverUSB.c: código para inicializar memória compartilhada da região status;
 - serverUSB.mexw32: arquivo mex gerado a partir de serverUSB.c;
 - lerStatus.c: código para leitura da região status da memória compartilhada;
 - lerStatus.mexw32: arquivo mex gerado a partir de lerStatus.c
 - escreverStatus.c: código para escrita na região status da memória compartilhada;
 - escreverStatus.mexw32: arquivo mex gerado a partir de escreverStatus.c;
 - server.exe: arquivo executável do servidor que deseja-se utilizar;
- Códigos do Servidor
 - server.cpp: código do servidor;
 - demais arquivos: bibliotecas necessárias para compilação do código do servidor;
- Esquemáticos
- Desenhos 3D do pendubot
- Fotos e Vídeos demonstrativos