

universidade  
de aveiro

## Departamento de electrónica, telecomunicações e informática

Curso            8204 - Mestrado Integrado em Engenharia Electrónica e Telecomunicações  
Disciplina      41489 - Sistemas de Instrumentação Electrónica  
Ano Lectivo    2018/2019

### Terceira Fase

#### *Trabalho 4 - Controlo de Velocidade e Posição de um Elevador*

##### Autores:

76517      João André de Jesus Cruz  
76674      Henrique da Silva Bernardes Camello Martins

Turma      P4  
Grupo      G6

Data      1 de Julho de 2019  
Docente    Pedro Fonseca

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Descrição do Problema</b>	<b>2</b>
2.1	Objetivos do Projeto: . . . . .	2
<b>3</b>	<b>Equipamentos</b>	<b>3</b>
<b>4</b>	<b>Procedimento</b>	<b>3</b>
4.1	Arquitetura de Software . . . . .	3
4.1.1	Calculo da Posição . . . . .	3
4.1.2	Escolha do Andar . . . . .	4
4.1.3	Controlador PI . . . . .	4
4.1.4	External_isr_3(void) . . . . .	5
4.1.5	LCD . . . . .	5
4.2	Hardware . . . . .	6
4.2.1	Circuito de Acondicionamento dos Sensores de Hall . . . . .	6
<b>5</b>	<b>Testes, Resultados e Análise</b>	<b>9</b>
5.1	Testes da Contagem da Posição do Elevador . . . . .	9
5.2	Testes do Controlador PI . . . . .	10
5.3	Testes do LCD . . . . .	11
<b>6</b>	<b>Conclusão</b>	<b>12</b>
<b>7</b>	<b>Bibliografia</b>	<b>13</b>
<b>8</b>	<b>Anexos</b>	<b>15</b>

## 1 Introdução

Este projeto a ser realizado no contexto das aulas práticas de Sistemas de Instrumentação Eletrónica, tem como objetivo, o controlo da posição e velocidade de um elevador. Para isso, foi desenvolvido um sistema baseado num microcontrolador sendo este mesmo o Max32 Digilent.

Quanto ao funcionamento do elevador, resumidamente este está acoplado a um sistema cabo roldana, que está ligado a um motor e que tem o seu veio ligado a um codificador incremental bi-fásico.

A velocidade e posição do elevador, será controlada pelo microcontrolador. A escolha da posição será feita pelo utilizador. Quanto à velocidade, essa será controlada de acordo com um perfil calculado.

Nesta fase foi desenvolvido o resto do software que faltou da segunda fase nomeadamente o controlo do LCD, a posição, velocidade do encoder e o controlador PI. Este relatório referente à terceira fase apresenta o resto do software desenvolvido, resultados finais e conclusões.

## 2 Descrição do Problema

Revendo os objetivos do projeto o que se pretende é o controlo da posição e da velocidade de um elevador, este mesmo pode funcionar em duas direções diferentes (para baixo e cima), é usada uma ponte H que faz com que o motor varie o sentido de rotação conforme o sinal PWN recebido.

O utilizador, como já referido na Introdução irá interagir com o sistema, podendo escolher a posição. A velocidade e a posição irá ser apresentada num display LCD.

Na terceira fase, tem-se como objetivos o controlo da posição e velocidade do elevador e um controlador PI para o controlo do motor.

### 2.1 Objetivos do Projeto:

- Definição do ponto de chegada em coordenadas do encoder.
- Capacidade de programação dos parâmetros  $V_{max}$ ,  $V_{apr}$  e  $D_{apr}$  com configuração automática dos perfis
- Capacidade de alimentação de motor DC (5V / 1W).
- Encoder bi-fásico com saída TTL e 500 ipr
- Detecção de piso por sensor de proximidade (efeito de hall), com auto-calibração.
- Adicionalmente: capacidade de programação dos parâmetros do controlador.
- Medição da corrente consumida.

### 3 Equipamentos

- 4xDíodo BA157
- Potenciómetro 10k $\Omega$
- Condensador 100uF
- Díodo Zener 2V4
- Condensador 0.1uF
- Driver de Motor L293NE
- Condensador 100uF
- OpAmp TL084
- 3xResistência 1k $\Omega$
- 5xPushbuttons
- 16xResistência 10k $\Omega$
- LCD 1602a
- 8xResistência 330k $\Omega$
- Ficha DB9M
- Resistência 220 $\Omega$
- Resistência 560 $\Omega$
- Fios
- Fonte de Alimentação do laboratório
- Osciloscópio

### 4 Procedimento

#### 4.1 Arquitetura de Software

Nesta secção do relatório da fase três, são apresentados os últimos device drivers, o controlador PI mais fluxogramas que ajudam a compreender o seu funcionamento.

As funções suplementares desenvolvidas foram:

**external\_isr\_3(void)**

##### 4.1.1 Calculo da Posição

Sabendo que temos de controlar a posição do elevador, o cálculo das coordenadas do encoder é vital no bom funcionamento do controlador PI anteriormente exposto. No relatório da fase 2 teve-se um problema na leitura dos canais do encoder, em que aparecia uma oscilação de 50 Hz. Esta oscilação foi identificada como um problema de massas, que foi resolvido ligando as massas todas juntas.

Com a abordagem da fase anterior, para o calculo da posição, tinha-se uma contagem da posição com baixa repetibilidade, o que faz com a contagem seja inválida. Este facto foi verificado na prática com a ajuda do docente, ativando uma saída antes do printf e desativando logo a seguir à instrução. Recorrendo ao osciloscópio no canal 1 mediu-se o Canal A do encoder e no Canal B o pino da saída configurada. Por fim executando o código verificou-se uma perda de contagens elevada, pois este printf dentro da rotina de interrupção do Timer causa um atraso significativo e com efeito pouco previsível nas contagens. Para solucionar este problema, pode-se colocar o print das contagens dentro de um ciclo while na main. Executando de novo o software, observou-se, deslocando o elevador para cima e baixo, que o valor da posição, agora é válido. Alterando o sentido da rotação e fazendo várias repetições têm-se valores consistentes e com repetibilidade.

Quanto à estratégia da obtenção das contagens, mantém-se o mesmo procedimento da fase 2, apenas se fez um diagrama de fluxo adicional, para expor com a maior clareza o seu funcionamento, [Anexos - Diagrama de Fluxo Calculo da Posição](#).

#### 4.1.2 Escolha do Andar

Como já referido na fase 1, o utilizador irá escolher o andar desejado com recurso a botões. Para isso tem-se que ter a leitura do estado dos botões e a contagem da posição do elevador. A leitura do estado dos botões foi desenvolvida com sucesso na fase 2, o processo das contagens foi corrigido nesta ultima fase como explicado anteriormente.

Posto isto, sinteticamente o que esta rotina vai fazer é atribuir um valor de setpoint a cada andar, sendo esta variável do tipo volatile. Por exemplo, quando o utilizador prime o botão correspondente ao ultimo andar, atribui ao setpoint o valor da coordenada do encoder que coincide com o ultimo piso. Portanto como referido, no sub capitulo anterior, a contagem da posição é vital para o bom funcionamento do sistema.

A obtenção dos valores dos setpoints foi feita da seguinte maneira: fez-se um breakpoint na função de retorno do valor da contagem ( GetPos() ) e colocou-se o elevador a mover-se até ao piso desejado. No momento em que o elevador se encontra no piso, parou-se o movimento e passando o cursor do rato na variável da contagem tirou-se o valor da posição para cada andar. O software foi executado com a opção do MPLAB "Debug Main Project".

Concluindo têm-se quatro botões um para cada andar, cada um com o seu valor de setpoint dentro de uma rotina de interrupção de um timer com frequência de cerca de 10 Hz. Para maior clareza tem-se o diagrama de fluxo, [Anexos - Diagrama de Fluxo Escolha do Andar](#).

#### 4.1.3 Controlador PI

Para o controlo da posição e velocidade usou-se um controlador PI. Com base nos conceitos adquiridos nas UC de controlo construiu-se o controlador. Para calcular os parâmetros discretos deste controlador, colocou-se o elevador a subir e a descer. A posição foi retirada utilizando o modulo da UART e com um script simples no Matlab fez-se o calculo dos parâmetros do controlador. Desta maneira obtêm-se com precisão os valores para as constantes.

Após calculados os parâmetros implementou-se o controlador PI tal como se pretende. Para que o sistema funcione corretamente, deve ser executado de forma sequencial. Isto é, em primeiro lugar lê-se o valor do SetPoint e a Posição atual, e apenas de seguida se devem passar os parâmetros para o controlados, depois de este calcular o sinal de controlo adequado é que se deve atuar sobre os sinais PWM que tem o controlo sobre o motor. Não seguir esta sequencia poderá não trazer problemas no entanto a precisão do funcionamento do elevador será menor, uma vez que os sinais poderão estar a ser lidos já depois de, por exemplo, ter passado o SetPoint e assim a atuação irá ser feita fora do local que se pretende.

Numa primeira abordagem fez-se a execução do controlador PI, sem considerar a desaceleração do elevador na chegada ao piso destino. Tendo o valor dos setpoints e a função da posição a funcionar corretamente, obteve-se com sucesso o deslocamento do elevador para todos os pisos, sempre com precisão na chegada tanto para cima como para baixo.

Agora considerando a desaceleração do elevador na chegada ao piso destino, decidiu-se utilizar os sensores de Hall para efetuar a deteção do elevador na proximidade do piso. Com esta abordagem e com os device drivers da segunda fase da leitura da tensão dos sensores, pode-se com simplicidade saber a presença ou não do elevador no piso.

Quando a posição se aproxima do setpoint, os sensores de Hall começam a fazer o controlo da velocidade do sinal. Como já foi mencionado nos relatórios anteriores e como é feito mais à frente dos sensores de Hall, temos uma tensão média de 2.5V, quando dos afastamos deste a tensão aumenta. Com um circuito de acondicionamento e a leitura dos valores na ADC permite obter o valor do sinal entre 0 e 1023, a partir do qual conseguimos desacelerar o elevador assim que este se aproxima da posição do sensor de Hall.

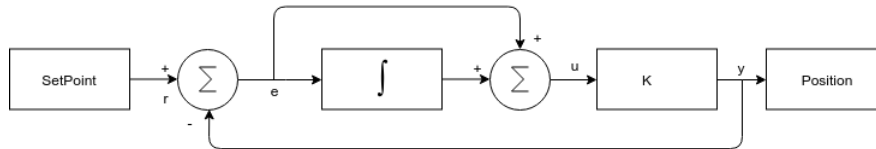


Figura 1: Diagrama de um controlador PI

#### 4.1.4 External\_isr\_3(void)

Esta rotina tem como função fazer a calibração do elevador, faz o reset aos parâmetros do controlador PI, da sua posição e também coloca o elevador no piso zero. Logo para isso, decidiu-se usar um pino de interrupção externa que tem ligado a si um botão.

A rotina da interrupção é executada quando se prime o botão de reset, a partir daí gera-se a PWM1 com um duty cycle 100 e a PWM2 com 0, desta maneira o elevador desce. O movimento mantém-se enquanto o bottom switch não for premido, no momento em que é premido o movimento cessa colocando o duty cycle das duas PWM a 0 e faz-se a reiniciação dos parâmetros do controlador PI. Para maior clareza tem-se o diagrama de fluxo, [Anexos - Diagrama de Fluxo Função da Calibração](#).

#### 4.1.5 LCD

Como já foi mencionado e descrito na fase anterior deste trabalho e é visível no diagrama de fluxo que se segue, para o correcto funcionamento do LCD é necessário, em primeiro lugar é necessário definir os portos do microcontrolador que se irão usar, de seguida é necessário fazer a correcta inicialização dos modos de operação, neste caso em específico deve ser executado sequencialmente os seguintes comandos, sempre que inicializado o display, Reset do display, Set do modo 4bits e 2 linhas, Desligar o cursor e por fim colocar no entry mode. Cada uma das formas de implementação pode ser consultada no datasheet do LCD [7]. Após a iniciação do LCD, é feito na função main o update da linha 1 do LCD, na linha 2 apenas é feito o update se for premido algum botão para mover o elevador, e nesse caso imprime qual é o próximo andar, ou no caso de ser premido o botão de calibração, que imprime uma mensagem a informar que o elevador está a ser calibrado. Este ciclo repete-se sempre que for premido um novo botão. [Anexos - Diagrama de Fluxo de funcionamento do LCD](#).

Um dos problemas que enfrentamos na segunda fase do projecto e também ao decorrer da fase 3, foi o facto de não ser possível utilizar o LCD ao mesmo tempo que a UART, ora, isso deveu-se ao facto de aquando da escolha dos pinos de dados para o LCD um deles ser o pino 0 do kit, que está sabém a ser utilizado pela UART, sendo que isso não permitia que se usassem os 2 pinos em simultâneo, pelo que para resolver esta situação apenas foi necessário atribuir novos pinos aos pinos de dados.

## 4.2 Hardware

Quanto a alterações de Hardware permaneceu tudo na mesma em relação às fases anteriores. Apenas propomos um circuito de acondicionamento de sinal dos sensores de Hall, para a maximização da resolução da ADC.

### 4.2.1 Circuito de Acondicionamento dos Sensores de Hall

Posto isto a partir do estudo do comportamento dos sensores de Hall na fase 2, tem-se resumidamente: tensão referencia 2.5(V) e tensão com a proximidade de um campo magnético 2.4(V). Portanto com uma tensão de 2.4(V) sabe-se que o elevador está no andar em questão.

Logo no caso de se ligar os sensores de Hall diretamente à ADC da PIC32 tem-se a seguinte resolução:

$$V_{ADC,1} = \frac{V_{Hall,1} * 2^{10}}{3.3} = \frac{2.5 * 2^{10}}{3.3} = 776$$

$$V_{ADC,2} = \frac{V_{Hall,2} * 2^{10}}{3.3} = \frac{2.4 * 2^{10}}{3.3} = 745$$

$$Res_{ADC} = \frac{2.5 - 2.4}{776 - 745} = 3(mV)$$

Portanto tem-se uma resolução de 3 mV com a ligação direta à ADC. Esta foi a implementação escolhida no projeto.

No caso de se querer maximizar a resolução dos sensores, tem de se projetar um circuito de acondicionamento de sinal. Este mesmo tem de ter uma tensão de offset que vai ser subtraída à tensão do sensor e por fim um ganho adequado para maximizar a resolução. Na figura seguinte tem-se a cadeia de acondicionamento de sinal por blocos, mais o circuito.

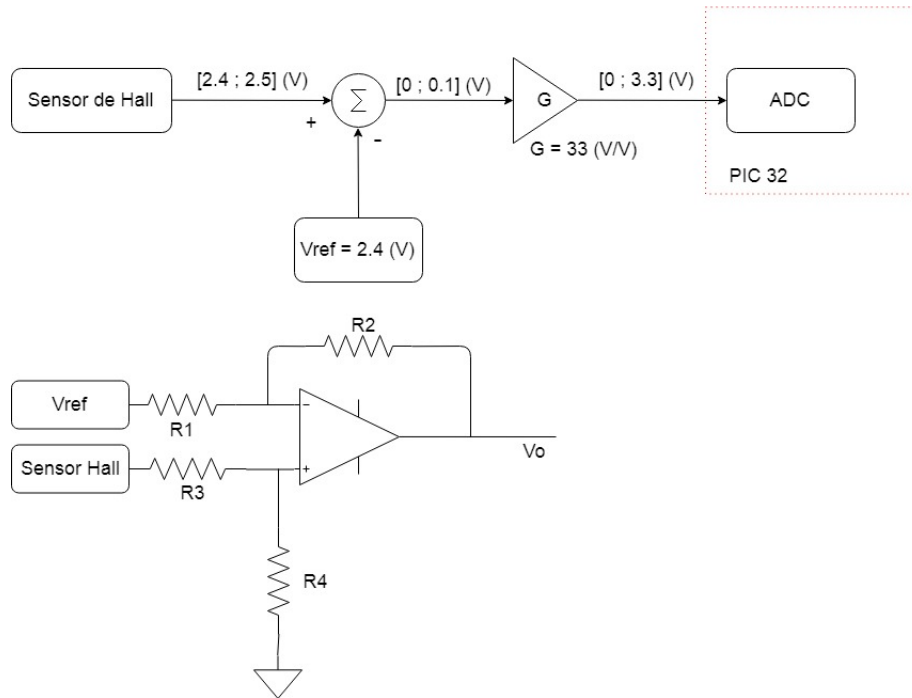


Figura 2: Acondicionamento do Sinal

Para se ter o ganho de 33 (V/V) e sabendo que o ganho do amplificador é:

$$A_v = \frac{R_2}{R_1} \Leftrightarrow R_2 = R_4 = 330k\Omega, R_1 = R_3 = 10k\Omega$$

Se

$$(R_2 = R_4, R_1 = R_3)$$

Para gerar a tensão de referencia pode-se usar o circuito seguinte:

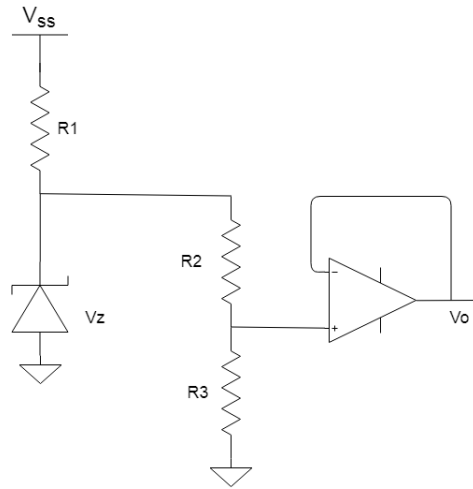


Figura 3: Circuito Gerador da Tensão Referencia

Em que  $V_z$  pode ser um díodo de zener de 3.3(V) e  $V_{ss}$  os 5(V) da PIC32. Tem-se então:

$$R_1 = \frac{V_{ss} - V_z}{I_z} \Leftrightarrow \frac{5 - 3.3}{15m} \Leftrightarrow R_1 = 113\Omega$$

$$V_0 = \frac{R_3}{R_3 + R_2} * V_z \Leftrightarrow 2.4 = \frac{R_3}{R_3 + R_2} * 3.3 \Leftrightarrow R_3 = 2.597 * R_2$$

Por fim têm-se os gráficos com as relações todas de tensões, resumindo toda a cadeia de acondicionamento de sinal mais o calculo da resolução.



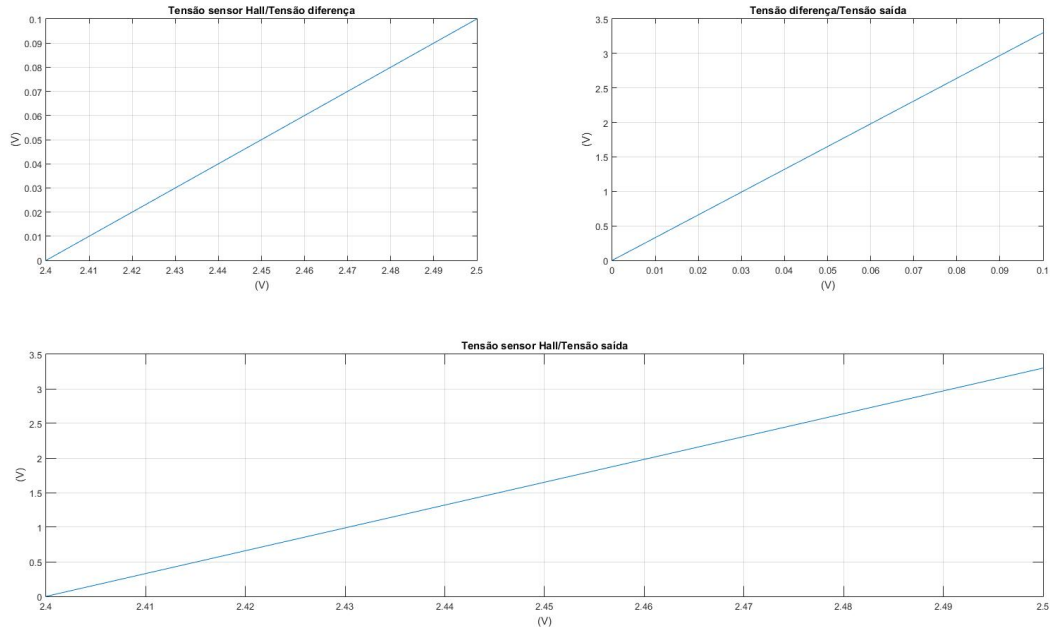


Figura 4: Relações entre as tensões

$$V_{ADC,1} = \frac{V_{Hall,1} * 2^{10}}{3.3} = \frac{3.3 * 2^{10}}{3.3} = 1023$$

$$V_{ADC,2} = \frac{V_{Hall,2} * 2^{10}}{3.3} = \frac{0 * 2^{10}}{3.3} = 0$$

$$Res_{ADC} = \frac{2.5 - 2.4}{1023} = 98(\mu V)$$

Concluindo desta maneira tem-se a maior resolução possível. Apesar da melhoria evidente na resolução da leitura, vem um acrescento ao custo do projeto. Decidiu-se que este custo adicional e hardware não seria necessário pois consegue-se produzir um resultado satisfatório com os sensores ligados diretamente à PIC32.

Apesar de tudo esta configuração pode ser útil, numa situação mais exigente, em que se quer a melhor deteção da aproximação do elevador ao piso.

## 5 Testes, Resultados e Análise

### 5.1 Testes da Contagem da Posição do Elevador

#### Objetivo:

Este teste tem como objetivo verificar, se a contagem da posição elevador é correta. E retirar o valor do setpoint de cada andar.

#### Descrição e Resultados Esperados:

Devido ao problema anteriormente referido da perda de contagens utilizando a função printf. Utilizou-se a opção do MPLAB "Debug Main Project" com um breakpoint na função GetPos() que retorna o valor do incremento que equivale a posição. Desta maneira colocou-se o elevador em movimento até à posição desejada e parava-se o movimento. De seguida passou-se o cursor na variável da contagem e consegue-se ver o valor da variável. Fez-se este procedimento para cada andar. É expectável conseguir-se retirar o valor correto da posição para cada andar. Correto quer dizer-se um valor com repetibilidade e sem perda de contagens.

#### Resultados:

Recorrendo à ferramenta "Debug Main Project" obtiveram-se os seguintes valores da posição para cada andar:

Andar	Posição
0	0
1	3090
2	5997
3	9063

#### Análise:

Os valores da posição foram obtidos repetindo o teste várias vezes e obtiveram-se valores sempre da ordem dos apresentados na tabela anterior. Portanto conclui-se que a contagem faz-se sem perdas e dando valores corretos consequentemente.

## 5.2 Testes do Controlador PI

### Objetivo:

Este teste tem como objetivo verificar, se o controlador PI implementado funciona corretamente, colocando o elevador no andar escolhido pelo utilizador.

### Descrição e Resultados Esperados:

Utilizando os valores de setpoint retirados na secção anterior de testes implementou-se o controlador PI. Fez-se a seleção de todos os andares, tanto a subir como a descer. É expectável que o elevador consiga deslocar-se para qualquer andar, sem perda de precisão no ponto de chegada.

### Resultados:

Após a execução do software com o controlador PI, testou-se todos os caminhos que o elevador podia tomar. De seguida tem-se uma tabela com os teste feitos:

Andar Inicial	Andar Final	Sentido	Sucesso
0	1	Subida	Sim
0	2	Subida	
0	3	Subida	
1	0	Descida	
1	2	Subida	
1	3	Subida	
2	0	Descida	
2	1	Descida	
2	3	Subida	
3	0	Descida	
3	2	Descida	
3	1	Descida	

### Análise:

Conclui-se que o controlador PI funciona corretamente, pois o elevador consegue deslocar-se para qualquer andar e com precisão na paragem no piso desejado.

### 5.3 Testes do LCD

#### Objetivo:

Este teste tem como objetivo verificar o bom funcionamento do LCD.

#### Descrição e Resultados Esperados:

Foi configurado o LCD para usar as duas linhas do display. Na primeira linha permanentemente estará impressa a string "Bem-Vindo", na segunda dependendo do botão premido apresentará as strings: "Andar 0", "Andar 1", "Andar 2", "Andar 3" e "Calibration". É esperado que apresente a string "Bem-Vindo" na primeira linha permanentemente, e depois na segunda linha as restantes strings anteriormente anunciadas dependendo do andar escolhido. A string "Calibration" é impressa se se premir o botão correspondente à função da calibração do elevador.

#### Resultados:



Figura 5: Display de Strings no LCD

Após a execução do software premiu-se cada botão obtendo os seguintes resultados:

#### Análise:

A partir das imagens anteriores concluiu-se que a impressão no LCD das strings foi feita com sucesso.

## 6 Conclusão

Quanto à terceira fase atingiu-se quase por completo os objetivos impostos pelo guião da disciplina, estando estes enumerados na secção [Descrição do Problema - Objetivos](#). Com sucesso foram alcançados: a definição do ponto de chegada em coordenadas do encoder, capacidade de alimentação do motor DC, deteção do piso por sensor de proximidade, implementação do controlador PI e desenvolvimento de software para o controlo de um LCD com duas linhas, sendo este último um objetivo extra adicionado pelo grupo. Em falta ficou a medição da corrente consumida, capacidade de programação dos parâmetros de velocidade e distancia de aproximação.

Balanceando todo o trabalho desenvolvido, concluímos que os objetivos principais foram feitos, sendo estes, o controlo da posição e velocidade, escolha do andar pelo utilizador, leitura dos sensores de Hall, controlador PI e LCD capaz de imprimir strings.

Dando por concluído o projeto, apesar de não terem sido realizadas todas as tarefas, alcançou-se os objetivos principais deste mesmo.

## 7 Bibliografia

- [1] Apontamentos de Sistemas de Instrumentação Eletrónica, fornecidos pelo docente.
- [2] Esquema do Elevador,  
[https://elearning.ua.pt/pluginfile.php/962673/mod\\_resource/content/1/praticas/trabalhos/t4/elevador.pdf](https://elearning.ua.pt/pluginfile.php/962673/mod_resource/content/1/praticas/trabalhos/t4/elevador.pdf)
- [3] OMRON ,Rotary Encoder E6B2 Datasheet,  
[https://elearning.ua.pt/pluginfile.php/962676/mod\\_resource/content/1/praticas/trabalhos/t4/omrom\\_e6b2.pdf](https://elearning.ua.pt/pluginfile.php/962676/mod_resource/content/1/praticas/trabalhos/t4/omrom_e6b2.pdf)
- [4] SGS-THOMSON MICROELECTRONICS, l293B,L293E, Datasheet,  
[https://elearning.ua.pt/pluginfile.php/962679/mod\\_resource/content/1/praticas/trabalhos/t4/l293.pdf](https://elearning.ua.pt/pluginfile.php/962679/mod_resource/content/1/praticas/trabalhos/t4/l293.pdf)
- [5] HONEYWELL,SS495A-S,Datasheet  
<https://sensing.honeywell.com/honeywell-sensing-sensors-linear-hall-effect-ics-ss490-series-datasheet-005843-2-en.pdf>
- [6] Manual de referência online da Max32 Digilent,  
[https://elearning.ua.pt/pluginfile.php/963675/mod\\_resource/content/1/chipkit-max32\\_rm.pdf](https://elearning.ua.pt/pluginfile.php/963675/mod_resource/content/1/chipkit-max32_rm.pdf)
- [7] Manual do LCD Utilizado - 1602A  
[http://www.eigenstudium.de/fileadmin/user\\_upload/Download/Datenblaetter/LCD-1602A.pdf](http://www.eigenstudium.de/fileadmin/user_upload/Download/Datenblaetter/LCD-1602A.pdf)



## 8 Anexos

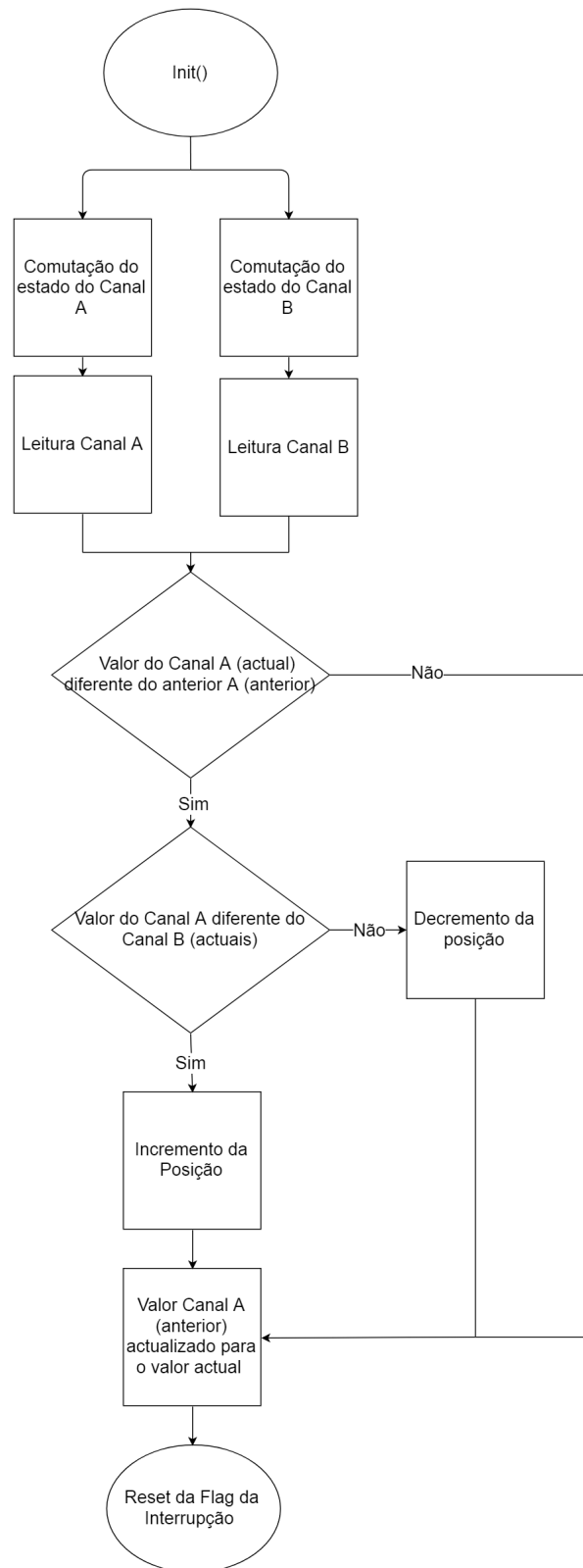


Figura 6: Diagrama de fluxo do Calculo da Posição



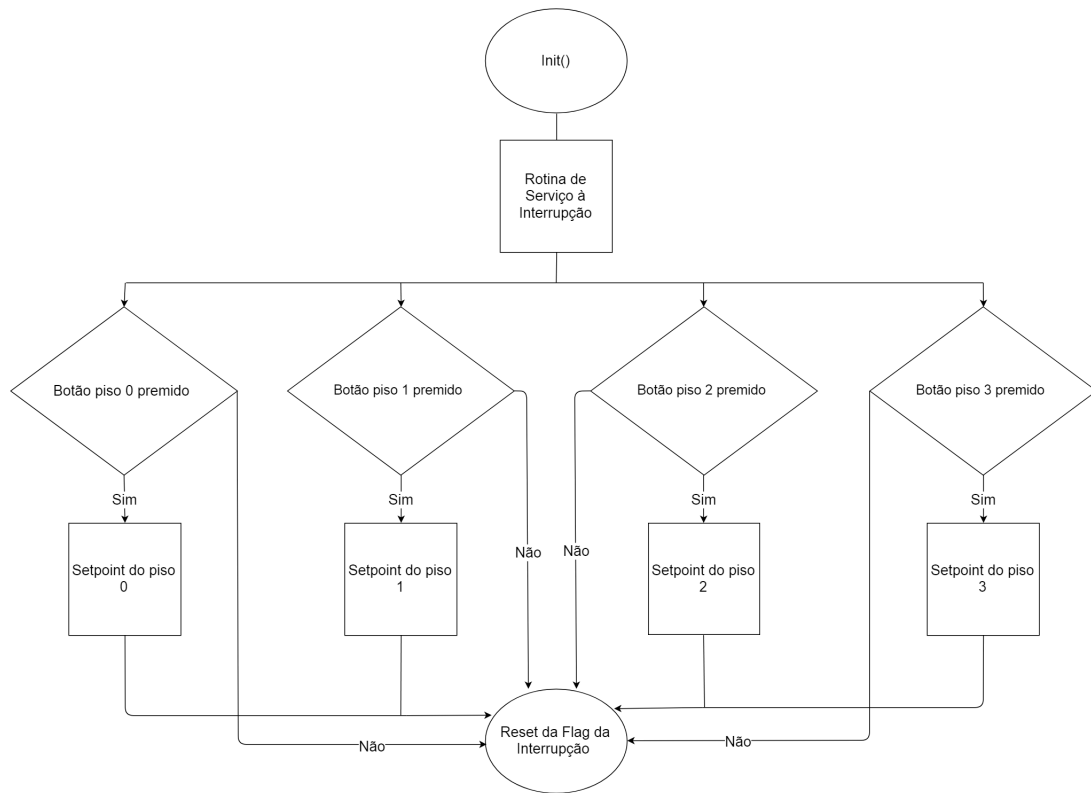


Figura 7: Diagrama de fluxo da Escolha do Andar

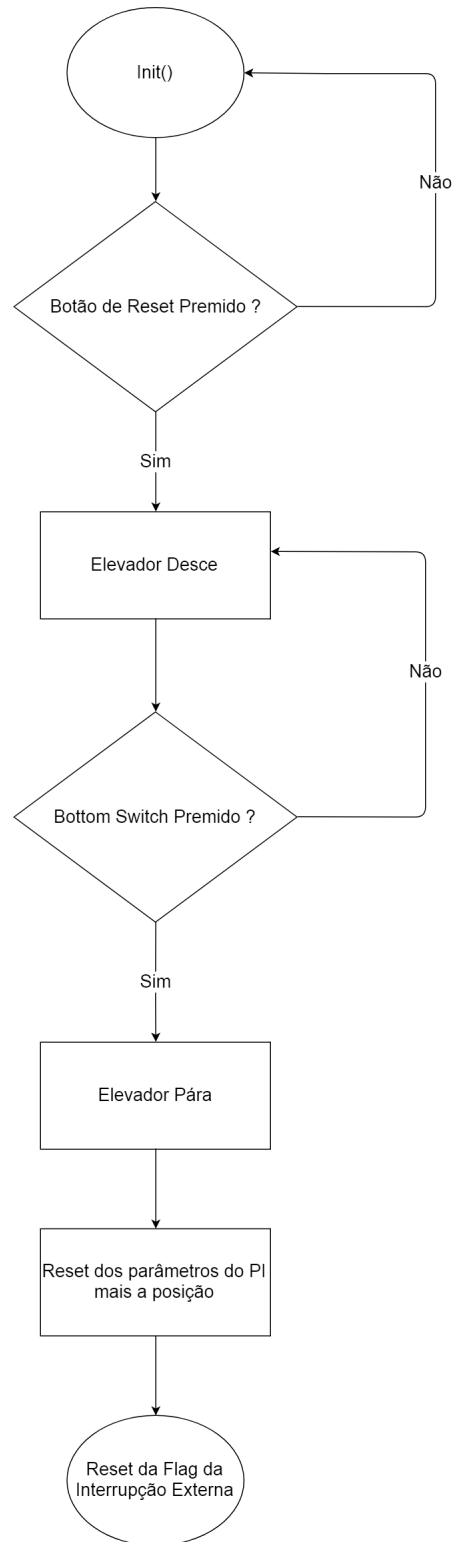


Figura 8: Diagrama de Fluxo da Função Calibração

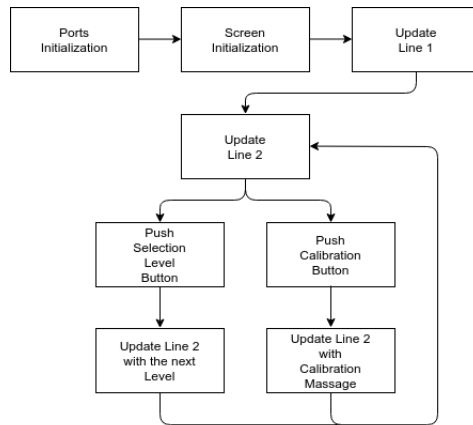


Figura 9: Diagrama de fluxo de funcionamento do LCD