



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

Segunda 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 7 de Maio de 2019
Docente Pedro Fonseca

Índice

1	Introdução	2
2	Objetivos	2
3	Especificações do Projeto	2
4	Descrição do Hardware	3
4.1	Driver do Motor	3
4.2	Interface com o utilizador	4
4.3	Display LCD	4
4.4	Botões	4
4.5	Encoder	5
4.6	Sensor de Efeito de Hall	6
5	Arquitetura de Software	9
5.1	ConfigBits	9
5.2	PWM	9
5.3	LCD	10
5.4	ADC	11
5.5	Encoder	11
5.6	PushButtons	12
5.7	UART	13
6	Testes e Resultados	14
6.1	Testes da geração dos sinais PWM	14
6.2	Testes dos sinais obtidos na ponte H e movimento do elevador	15
6.3	Testes da leitura da tensão dos sensores de efeito de Hall e deteção de piso	19
6.4	Testes da leitura dos botões e bottom switch	21
6.5	Testes da leitura dos canais do Encoder	23
7	Memória Descritiva	27
8	Conclusão	28
9	Bibliografia	28

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, irá se desenvolver-se 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 e pelo PC. A escolha da posição será feita pelo utilizador, com recurso a uma consola. Quanto à velocidade, essa será controlada de acordo com um perfil calculado, a partir de parâmetros enviados por um PC.

Este relatório referente à segunda fase apresenta as especificações dos procedimentos de validação e várias tabelas com os resultados dos testes efetuados. É importante referir que no fim da segunda fase todos os componentes estão montados e o software base está desenvolvido.

2 Objetivos

Sabendo que o objetivo do projeto é o controlo e a 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 segunda fase, tem-se como objetivos a apresentação das especificações dos procedimentos de validação, várias tabelas com os resultados dos testes efetuados, todos os componentes devidamente montados e o software base desenvolvido.

3 Especificações 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.

4 Descrição do Hardware

4.1 Driver do Motor

Para o driver do motor utilizou-se o integrado L293NE. Após a análise do datasheet do componente e da investigação do funcionamento de uma Ponte H chegámos às seguintes conclusões.

O L293 é controlado por um sinal PWM, que terá de ter uma tensão mínima de alimentação e lógica de cerca de 4.5 V. Posto isto, recorreremos às saídas open drain do micro controlador que permitem a geração de sinais até 5 V, o que cumpre com os mínimos de tensão anteriormente apresentados. A alimentação do componente, é feita através da fonte de alimentação da bancada da sala de aula.

Quanto ao funcionamento da ponte H este é um circuito que contém normalmente quatro transístores bipolares FETs que funcionam como switches. Cada um é controlado individualmente. Com Q1 e Q4 ativos há passagem de corrente pelo motor fazendo com que rode num sentido. Com Q3 e Q2 ativos o motor roda no sentido contrário. Os quatro transístores nunca podem estar os quatro ativos pois pode-se danificar o circuito. No caso do L293 já possui um mecanismo de proteção para este tipo de situação.

O controle do L293 é feito com um sinal PWM que corresponde a um sinal periódico retangular com frequência constante em que o seu tempo ON (sinal a 1) pode ser variável. Para isso varia-se o duty cycle que equivale à relação do período do sinal e o período a ON (sinal a 1). Logicamente com a variação do duty cycle do sinal PWM consegue-se controlar, o tempo em que os transistores estão ativos, o que equivale a controlar a velocidade do motor. A expressão do duty cycle do sinal PWM é dado pela seguinte fórmula:

$$DutyCycle = \frac{t_{ON}}{T} * 100\%$$

Já as conexões entre o motor e o driver são efetuadas através de uma ficha DBF9, que está no elevador, foi então necessário um cabo com uma ficha num dos lados e headers na outra, para fazer a ligação necessária. Tal como é visível no seguinte esquema:

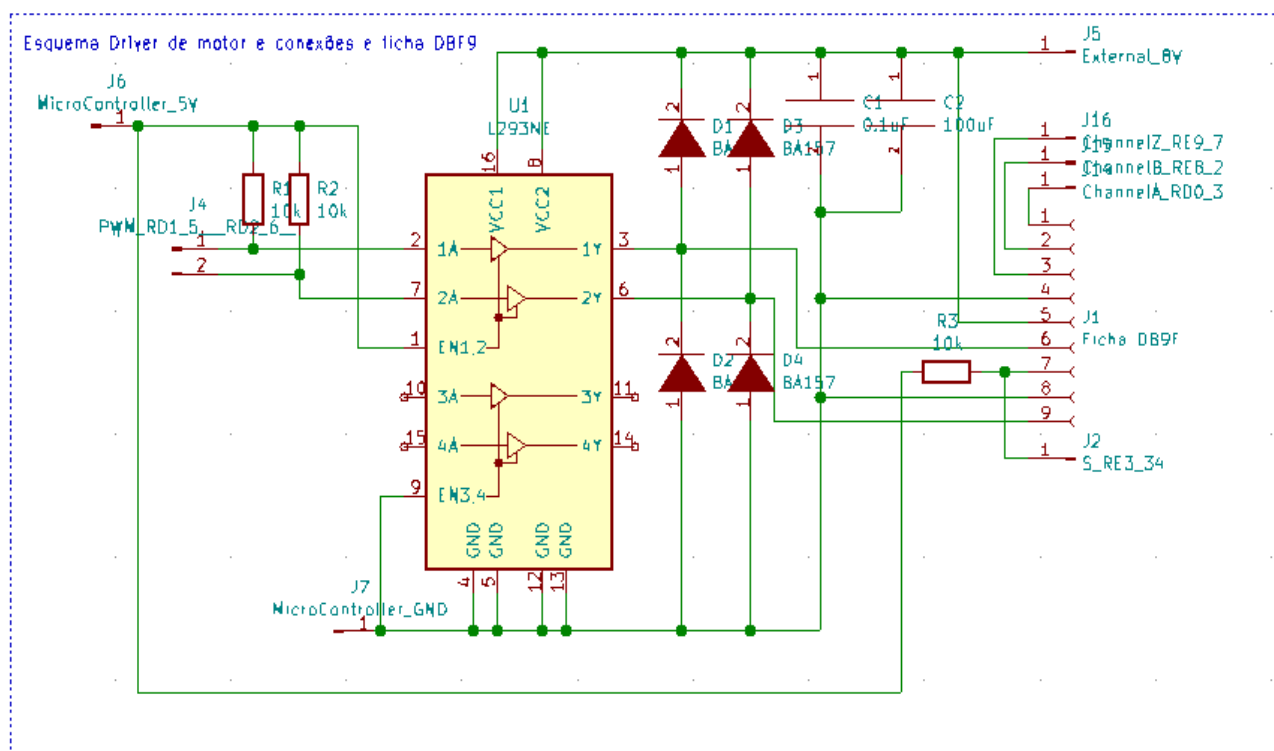


Figure 1: Esquema Elétrico do driver de motor

Os pinos utilizados nos quais obtemos um sinal PWM como saída, correspondem aos pinos 5 (RD1) e 6 (RD2) do Kit, o ground e a alimentação, que contém uma resistência de pull-up para cada uma das saídas de PWM, para que o open drain funcione.

4.2 Interface com o utilizador

4.3 Display LCD

Para a apresentação da posição e da velocidade do elevador escolheu-se usar um display LCD.

O display LCD tem como pinos: o display de contraste (V0), pinos de alimentação (5V e GND), 8 pinos de dados (D0-D7), um pino do modo de leitura ou escrita (R/W), um pino de enable para autorizar a escrita nos registos e por fim um pino da escolha do registo(RS) que controla onde se escreve na memória do LCD ou enviar comandos.

O display LCD pode ser controlado por dois modos de 4 bits e de 8 bits. O de 4 bits requer o uso de sete pinos do microcontrolador e o de 8 bits o uso de 11 pinos.

Na figura seguinte apresenta-se o esquema elétrico com as ligações do display LCD.

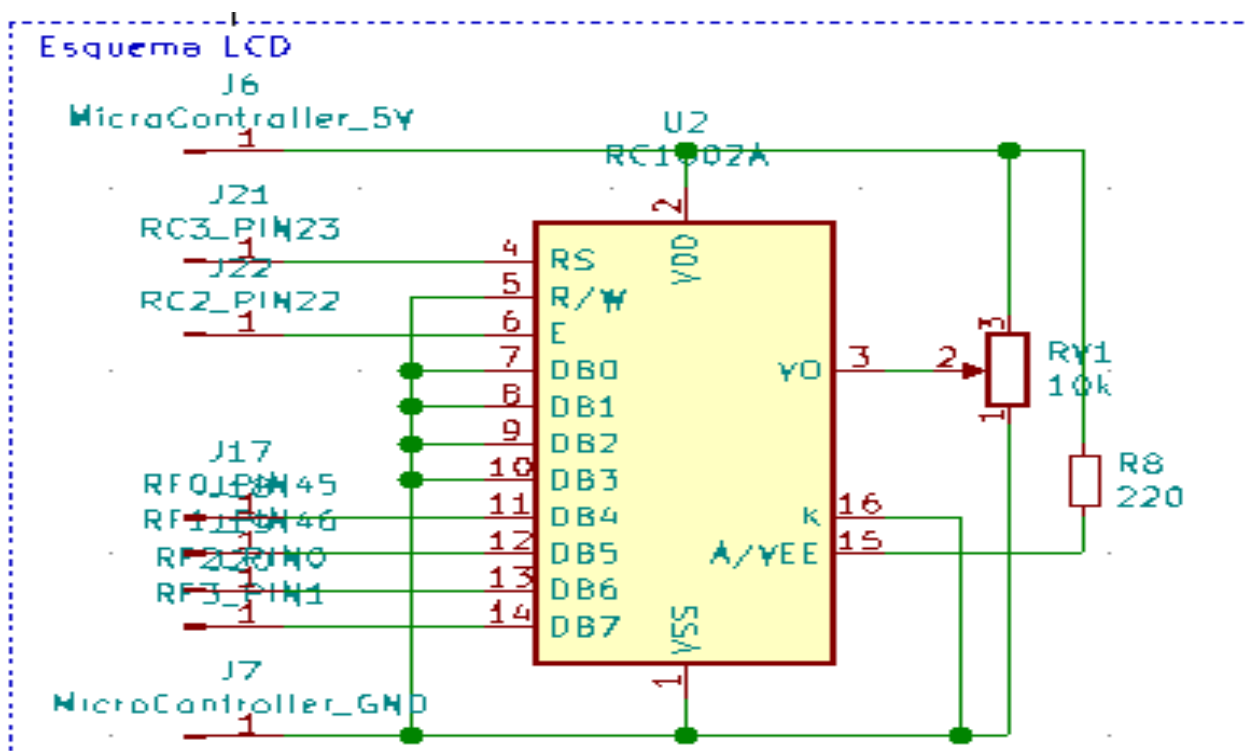


Figure 2: Esquema Elétrico do LCD

Para ligar o LCD à PIC utilizamos o modo de 4 bits e os pinos RS e E, não é necessário ligar o pino R/W uma vez que apenas se vão efectuar operações de escrita, assim este pino só precisa de ser ligado à massa. Os 4 pinos de dados (DB4-DB7) foram ligados aos pinos 45 (RF0), 46 (RF1), 0 (RF2) e 25 (RF3), já o pino enable ao 22 (RC2) e o pino RS ao 23 (RC3).

4.4 Botões

Para o controlo da posição do elevador utilizaram-se quatro botões, um para cada piso.

Na figura seguinte apresenta-se um esquema sintético de um push button, quando o botão é acionado cria-se uma ligação entre os pinos de cada lado (BC e AD) onde neste caso ocorre passagem de corrente.

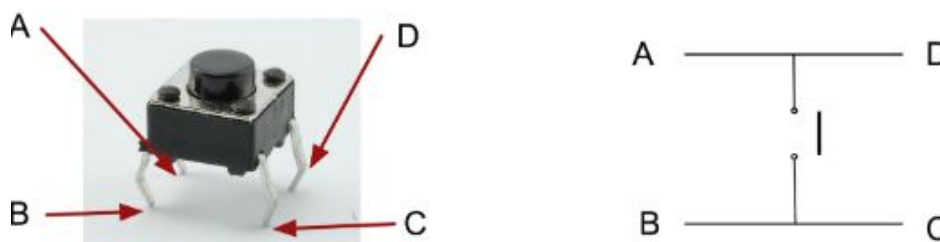


Figure 3: Esquema Elétrico de um push button

Quando à montagem dos botões recorreu-se a uma tensão de alimentação de 5 V (5 V do porto do microcontrolador), conectou-se uma resistência de pull up de cerca de 10k Ω em série com o botão (uma em cada botão) e por fim liga-se um pino do microcontrolador entre a resistência e o botão para assim fazer-se a sua leitura. Posto isto, a resistência liga-se ao pino B do botão e o pino D liga-se à massa. Na figura seguinte apresenta-se as ligações feitas dos quatro botões.

Pretende-se que o botão apresente apenas dois estados High(5 V) e Low (0 V) para isso é que se usou a resistência pull up com o objetivo que o valor não varie. Quando o botão não está premido, o circuito está em aberto fazendo com que a queda de tensão na resistência seja nula, logo o valor de tensão lido pelo microcontrolador é de 5 V equivalente ao estado High, mais tarde negado pelo software. No caso de estar premido, o circuito encontra-se fechado logo ocorre passagem de corrente no botão e por consequência a queda de tensão na resistência é de cerca de 5 V. Portanto o valor de tensão lido pelo microcontrolador é de 0 V equivalente ao estado Low, sendo que mais tarde é negado pelo software. Assim o microcontrolador consegue fazer uma leitura correta do estado do botão em causa e concluir se o botão está premido ou não.

Tal como já está descrito acima a forma como os push buttons funcionam, mostra-se na figura seguinte como se conectam ao microcontrolador.

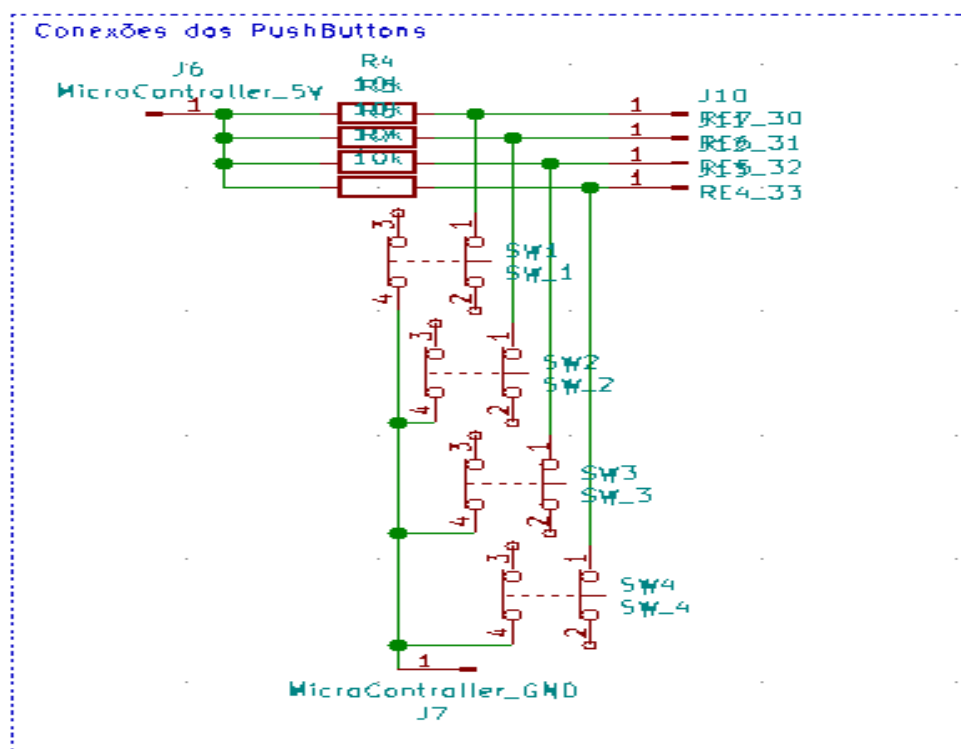


Figure 4: Esquema Elétrico dos PushButtons

Como é visível no esquema eléctrico os quatro botões encontram-se ligados aos pinos 30 (RE7), 31 (RE6), 32 (RE5) e 33 (RE4), enquanto o botão do piso zero se encontra ligado ao pino 34(RE3).

4.5 Encoder

Neste projeto usa-se um encoder incremental que produz 500 impulsos por revolução e tem três canais A,B e Z. O modelo usado neste projeto é o E6B2-CWZ6C da OMRON.

O encoder tem dois sentidos de rotação, sentido horário ou anti-horário. Analisando o sentido horário, a fase A irá estar adiantada 90 $^\circ$ em relação à fase B. Comparando diretamente a fase A e B e observando atentamente o nível das duas fases, chega-se à conclusão que neste caso (sentido horário) o nível da fase A é diferente do da fase B. Logo quando há uma mudança do nível da fase A o nível da fase B contraria o nível do da A. Este fenómeno é evidenciado na figura seguinte.

No sentido anti-horário a fase A encontra-se atrasada 90 $^\circ$ em relação à fase B. Quanto ao nível das fases, neste caso, encontram-se sempre iguais. Este fenómeno é evidenciado na figura seguinte. Portanto através da leitura e análise das fases do encoder, consegue-se afirmar qual o sentido de rotação deste mesmo, recorrendo a um algoritmo simples de controlo por software. Este baseia-se na observação do nível das fases, quando o nível do canal B é diferente do A incrementa-se a posição pois o encoder roda no sentido horário. Com os níveis dos dois canais diferentes a posição decrementa-se pois roda no sentido anti-horário.

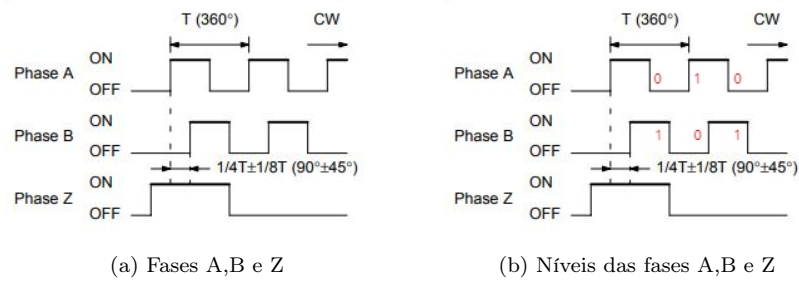


Figure 5: Sentido horário

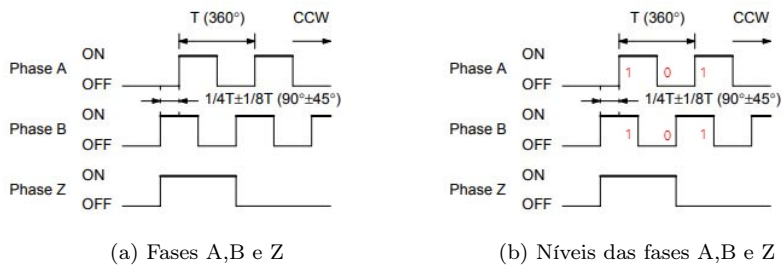


Figure 6: Sentido anti-horário

Quanto à implementação, sabe-se que a porta DB9F tem os três canais e que cada um é um sinal TTL de 5V. Logo pode-se ligar diretamente as fases ao microcontrolador, pois os pinos deste mesmo são tolerantes a 5V. A leitura dos níveis dos três canais é feito com interrupções externas que detetam mudanças de nível, sabendo que apenas alguns dos pinos são tolerantes a interrupções deste género, seleccionaram-se as interrupções 0, 1 e 2, que correspondem aos pinos 3(RD0), 2(RE8) e 7(RE9) para as fases A, B e Z, respetivamente. O método implementado está mais à frente aprofundado na secção [Arquitetura de Software](#).

Quanto à alimentação do encoder, de acordo com o seu datasheet, é cerca de 5 V. Aplica-se uma tensão à fonte de alimentação do elevador, que por sua vez alimenta o elevador mais o encoder. Este regulador é o tipo 7805 que pelo seu datasheet necessita no mínimo 7V de tensão de alimentação para funcionar corretamente. E ainda existe um díodo que apresenta uma queda de tensão, logo considerando todos estes fatores mais o conselho por parte do docente, a alimentação irá ser de 8V.

μA7800 SERIES
POSITIVE-VOLTAGE REGULATORS

SLVS056J – MAY 1976 – REVISED MAY 2003

recommended operating conditions

		MIN	MAX	UNIT
V_I Input voltage	μA7805C	7	25	V
	μA7808C	10.5	25	
	μA7810C	12.5	28	
	μA7812C	14.5	30	
	μA7815C	17.5	30	
	μA7824C	27	38	

Figure 7: Extrato do Datasheet do LM7085

4.6 Sensor de Efeito de Hall

O efeito de Hall por definição é: o aparecimento de uma diferença de potencial em um condutor elétrico, transversal ao fluxo de corrente e um campo magnético perpendicular à corrente.

Os Sensores de efeito de Hall são ativados por um campo magnético exterior, logo o sinal de saída é em função da densidade do campo magnético na proximidade do sensor. O campo magnético tem dois aspetos a densidade de fluxo magnético (B) e a sua polaridade (Norte ou Sul). No caso do sensor ser linear e for analógico, a sua tensão de saída é dada pela equação seguinte:

$$V_{th} = R_{th} * \left(\frac{I}{t} * B \right)$$

Em que V_{th} é a tensão de Hall, R_{th} é o coeficiente de Hall, I é a corrente que atravessa o sensor, t é a espessura do sensor em (mm) e B é o fluxo magnético em Tesla. A resposta típica de um sensor de Hall a um campo magnético é dado pelo seguinte gráfico. Conclui-se que a tensão de saída do sensor aumenta com o aumento do campo magnético, até que comece a saturar pelos limites impostos pela sua alimentação. Portanto vê-se que a tensão de saída é proporcional ao campo magnético.

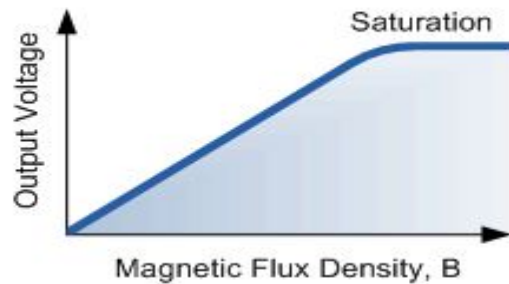


Figure 8: Gráfico V_0/B num sensor de efeito de Hall

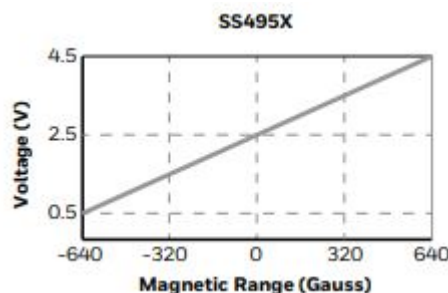
Quanto ao projeto em si, cada andar possui um sensor, obtendo no total quatro sensores de efeito de Hall de modelo SS495A da Honeywell. Após uma leitura e análise do datasheet do componente, conclui-se que para uma densidade do campo magnético de 0 B apresenta uma tensão de saída referencia de 2.5 V (V_{dc}). Com um campo magnético aplicado (de -670 B a 670 B tipicamente) obtêm-se variações típicas de $0.2 * V_{dc}$ a $V_s - (0.2 * V_{dc})$, também se sabe que a tensão de alimentação tem de ser no mínimo 4.5 V. Portanto sabendo que os pinos do microcontrolador conseguem produzir uma tensão de 5 V, os sensores são alimentados com ela e obtém-se uma variação entre 0.5 V e 4.5 V à saída de cada um. Este facto pode-se observar no gráfico seguinte do datasheet do componente.

Operating Temperature Range	-40 °C to 150 °C [-40 °F to 302 ° F]
Storage Temperature	-55 °C to 165 °C [-67 °F to 329 ° F]
Output Voltage	0.2 Vdc to ($V_s - 0.2$ Vdc) typ., 0.4 Vdc to ($V_s - 0.4$ Vdc) min.

(a) Gama da tensão de saída

Magnetic Range (typ.)	-67 mT to 67 mT [-670 G to 670 G]
Magnetic Range (min.)	-60 mT to 60 mT [-600 G to 600 G]

(b) Gama da densidade do campo magnético



(c) Gráfico V_0/B

Figure 9: Extratos do Datasheet do SS495A

A montagem dos sensores de Hall já se encontra feita no elevador logo, apenas é necessário fazer a leitura de

cada sensor recorrendo ao microcontrolador e fazer a sua devia alimentação. Para se lerem os valores de tensão obtidos nos sensores é necessário utilizar uma ADC, em que a mesma seja uma porta analógica, neste caso em concreto usou-se as portas analógicas AN0, AN1, AN2 e AN3 que correspondem aos pinos 54(RB0) a 57(RB3). Mais à frente na secção dos testes, conclui-se que o sensor na presença de um campo magnético tem uma tensão de saída de cerca de 2.4V, com um tensão referencia de 2.5V.

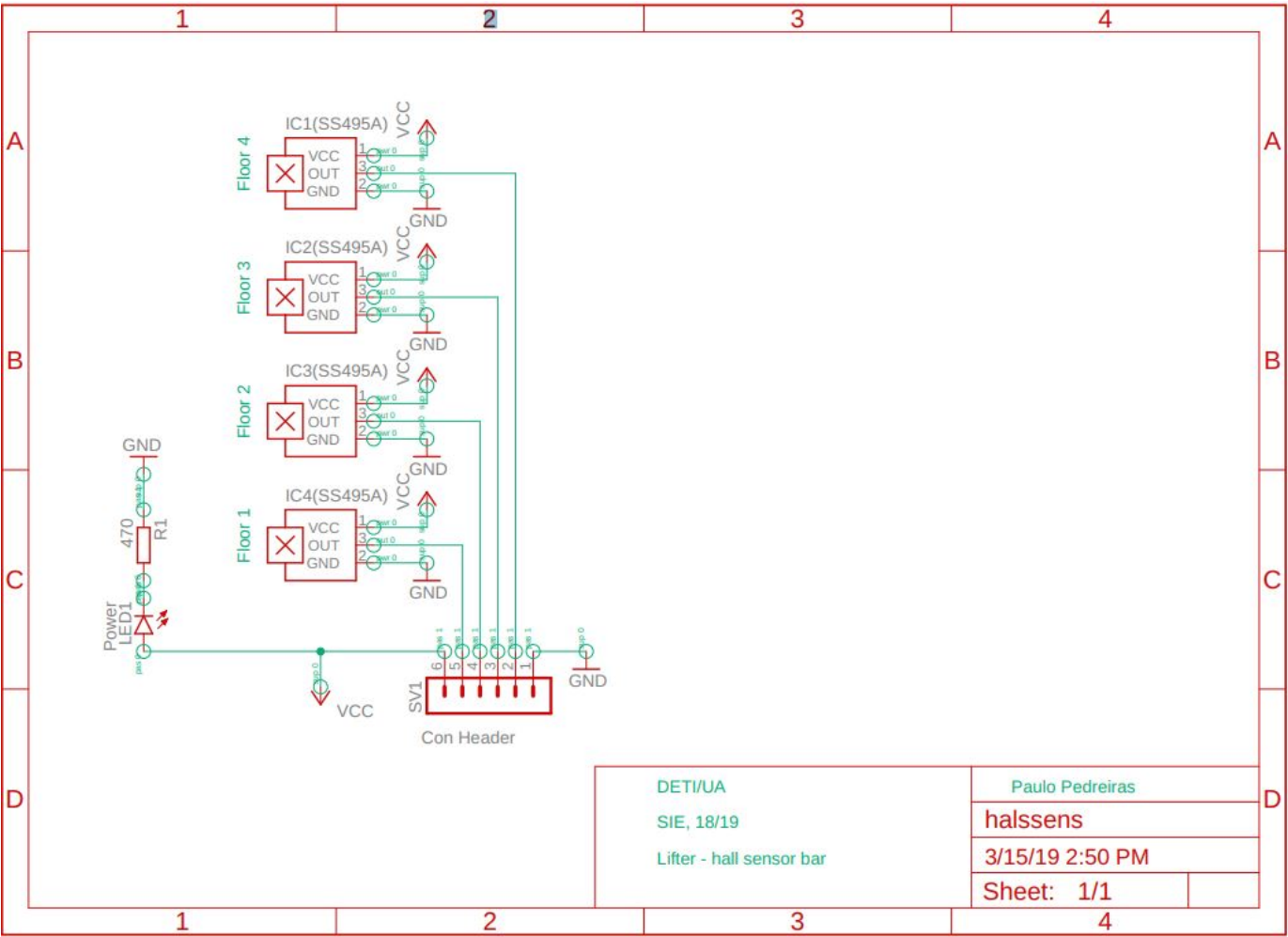


Figure 10: Esquema da montagem dos sensores de efeito de Hall

5 Arquitetura de Software

Nesta secção do relatório da fase dois, são apresentados os device drivers e fluxogramas que ajudam a compreender o seu funcionamento.

Para a melhor organização, foi criado um ficheiro para cada módulo desenvolvido. Logo tem-se que :

PWM

LCD

ADC

Encoder

PushButtons

UART

É importante frisar que poderão ocorrer alterações dos device drivers na ultima fase do projeto mas na altura serão devidamente evidenciadas.

5.1 ConfigBits

Este módulo já nos é fornecido, tem como função a configuração do CLOCK e PBCLK, com os valores 80MHz e 40MHz, respetivamente.

5.2 PWM

Este módulo, junta todas as funções necessárias, para gerar dois sinais PWM com duty cycle variável e frequência de 20kHz. A figura seguinte resume o funcionamento do módulo desenvolvido.

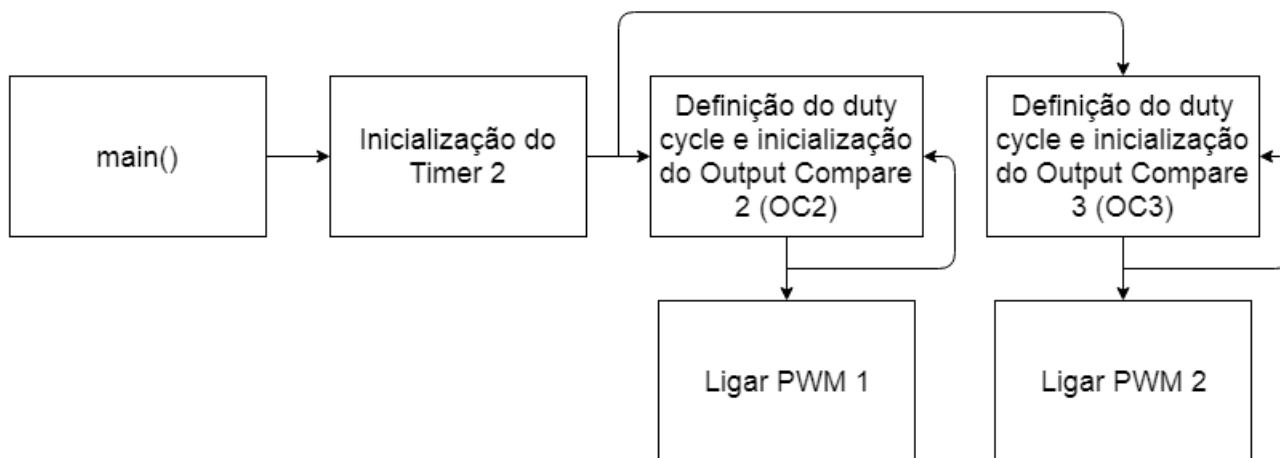


Figure 11: Diagrama de Fluxo da geração de duas PWM

A configuração deste módulo em específico foi facilitado pelo código já desenvolvido no projeto 0. Para a geração dos dois sinais PWM foi usado o Output Compare (OC2 e OC3) e para ambos o Timer 2 com uma frequência de cerca de 20kHz.

As funções desenvolvidas foram as seguintes:

SetTimer(void)

SetPWN1(dutyCycle)

SetPWN2(dutyCycle)

StartTimer(void)

A função **SetTimer** inicia os portos e configura o Timer 2 com uma frequência de 20 kHz. Para esse efeito usa-se a expressão enuncia a seguir para se obter o valor do PR2. Sendo o $f_{in} = 40MHz$ e o valor pretendido para a frequência $f_{out} = 20k$ o valor obtido para o PR2 é de 2000, com um valor de prescaler de 1.

$$f_{out} = \frac{f_{in}}{PRx + 1}$$

A função **SetPWM1** e **SetPWM2**, são equivalentes sendo que, ambas têm como parâmetro de entrada o valor de duty cycle desejado pelo utilizador e cada uma produz um sinal PWM. A primeira recorre ao Output Compare 2 e a segunda ao Output Compare 3, ambas como já referido anteriormente usam o Timer 2. A partir do registo OC2RS e OC3RS respetivamente, é possível alterar os valores dos duty cycles dos sinais PWM's gerados, de acordo com a seguinte formula:

$$OCxRs = \frac{(PR2 + 1) * \delta}{100}$$

O sinal PWM proveniente do **SetPWM1** é obtido no pino 5 e o do **SetPWM2** no pino 6 da. Para além disto foram configuradas as saídas em open drain para que se consigam gerar sinais PWM com tensão máxima de 5V. Pois a ponte H necessita de tensões mínimas de 4.5 V, desta maneira evita-se o uso de hardware adicional, o que é uma vantagem. Apenas se usa um pull up resistor por cada saída.

Por fim a função **StartTimer** inicia a geração dos dois sinais PWM, configurando o registo T2CONbits.TON a 1.

5.3 LCD

Este módulo é responsável pelo display do da velocidade e o nível onde o elevador se encontra. A figura seguinte resume o funcionamento do módulo desenvolvido.

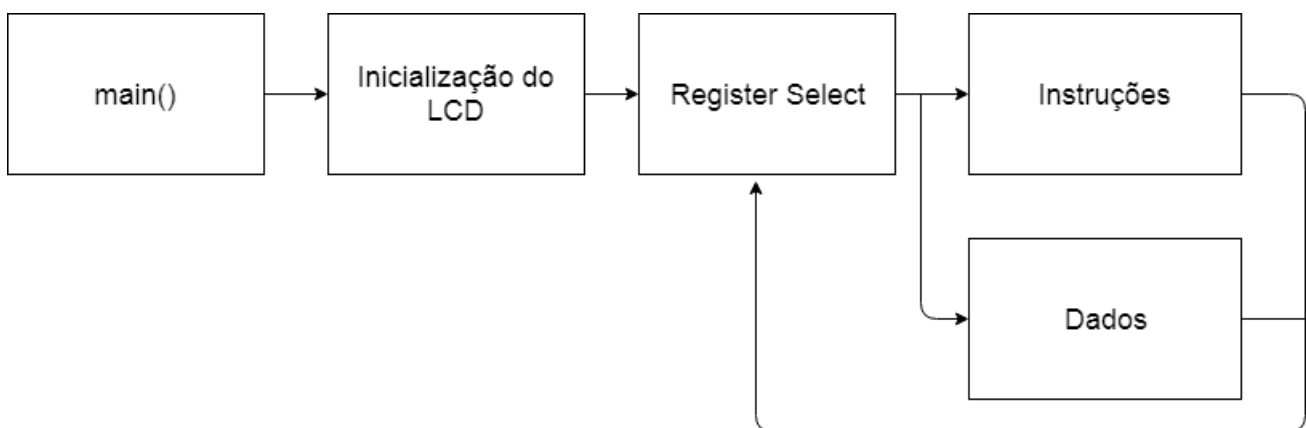


Figure 12: Diagrama de Fluxo da impressão de informação no LCD

Para a configuração e o display, o envio de comandos e caracteres, foram criadas as seguintes funções:

InitialiseDisplay(void)

ResetDisplay(void)

SendCommandToLCD(uint8_t uiCommand)

LCDSendString(char * str)

LCDSendChar(char character)

delay(uint32_t period)

A função mais importante deste módulo é a **SendCommandToLCD**, esta função é responsável por enviar comandos com as tarefas que o display deve executar, como Clear, Return Home, Mode Set, Display ON/OFF, cursor Shift, etc. Estes comandos são responsáveis por muitos das tarefas que o display executa.

Depois da função anterior a função mais importante é **InitialiseDisplay**, esta função executa alguns dos comandos pela função anterior, que são responsáveis pelo iniciar do display, como o modo de funcionamento, 4/8bits ou 1/2 linhas, desligar o cursor, iniciar o Entry Mode seguindo-se pelo inicial do display. Entre cada um

dos comandos é necessário um delay para que o comando seja executado, para isso existe também uma simples função que cria um **delay**, como parâmetro de entrada o valor de delay que se pretende em mS.

A função **ResetDisplay** executa por algumas vezes o Return Home.

Por fim as funções **LCDSendString** recebe um ponteiro de caracteres e envia um de cada vez para ser enviado para a função **LCDSendChar**, esta última envia os caracteres para o LCD.

5.4 ADC

Este módulo, junta todas as funções necessárias, para realizar a leitura da tensão dos quatro sensores de Hall. A figura seguinte resume o funcionamento do módulo desenvolvido.

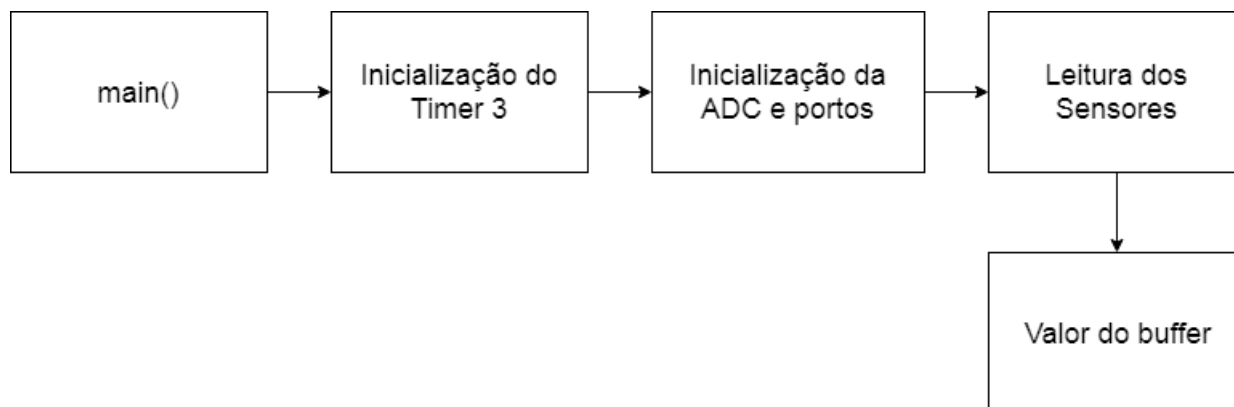


Figure 13: Diagrama de Fluxo da leitura dos sensores de Hall

As funções desenvolvidas foram as seguintes:

initADC(void)

initADCTimer(void)

readADCx(void)

buffer(void)

A primeira função deste módulo é **initADC**, esta função é responsável pelas configurações da ADC e a iniciação dos portos analógicos desta. São configurados os portos AN0 a AN3 como entradas.

A função **initADCTimer** serve para que seja feita uma leitura ao buffer com uma determinada frequência de amostragem sendo esta de cerca 400Hz. Para isso usou-se o Timer 3 com valores de prescaler e PR3, de 2 e 50000.

As funções **readADCx**, são 4 uma para cada sensor, cada uma selecciona o porto analógico devido a ler e fazem a decisão da comparência do elevador no andar em questão.

A função **buffer** é usada apenas para devolver o valor do buffer lido da ADC.

5.5 Encoder

Este módulo é responsável pela leitura dos três canais do encoder e pela decisão do sentido do movimento de elevador (para cima ou para baixo). A figura seguinte resume o funcionamento do módulo desenvolvido.

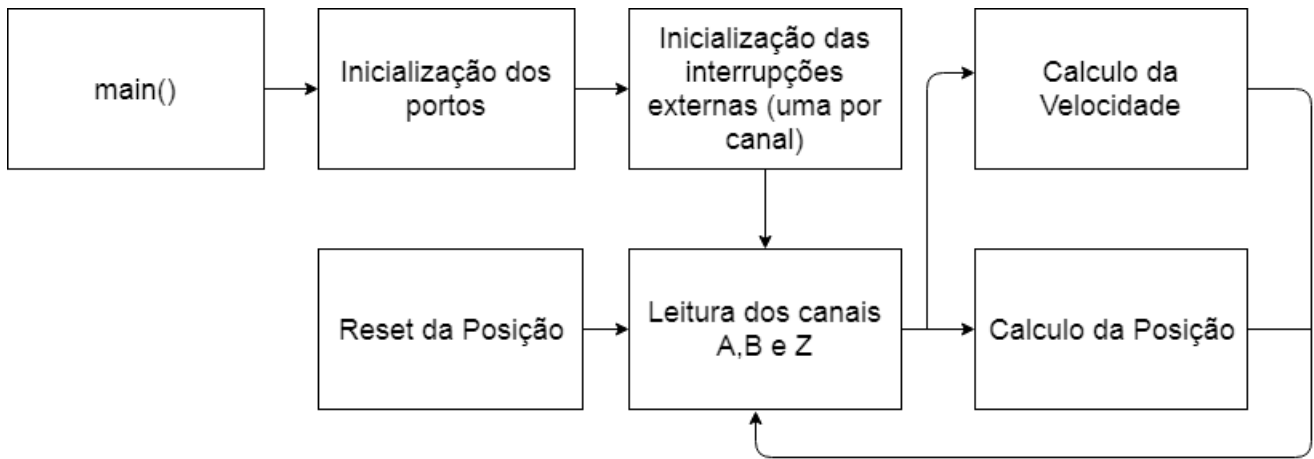


Figure 14: Diagrama de Fluxo da leitura dos canais A,B e Z, mais calculo de posição e velocidade

As funções implementadas neste módulo são as seguintes:

SetEncoderPorts(void)

SetExternalIntx(void)

GetChannelx(void)

GetPos(void)

ResetPos(void)

Init_Timerx_Interrupt()

A função **SetEncoderPorts**, serve para definir os portos de entrada responsáveis pela leitura do estado dos canais do encoder (Canal A,B e Z)

A **SetExternalIntx** inicia as interrupções externas para cada um dos canais A,B e Z do encoder.

A **GetChannelx** é responsável pela leitura do estado dos canais.

A função **GetPos** e **ResetPos** são responsáveis por devolver a posição do encoder e colocar o mesmo valor a zero.

Já a última função **Init_Timerx_Interrupt**, inicia dois timers por interrupções, cada um deles responsável pelo update da posição e o outro por calcular a velocidade. Tem-se o Timer 4 com uma frequência de 50Hz sendo este responsável pela obtenção da posição do elevador e o Timer 1 com uma frequência de 400Hz para o calculo da velocidade. O calculo da posição é feito por um método muito usado por software. Este algoritmo compara o estado dos canais A e B, se forem diferentes incrementa-se o contador da posição e forem iguais decrementa-se. O da velocidade faz-se a amostragem com uma frequência de 400Hz e dentro desse espaço de tempo vê-se a variação da posição do elevador. No fim divide-se a variação de posição pelo periodo de amostragem.

Sendo o objetivo deste módulo a leitura dos três canais do encoder, as funções **SetExternalIntx** estão configuradas para serem ativas numa raising edge do porto e no momento em que corre a interrupção lê-se o estado do canal e inverte-se o seu funcionamento para failing edge. Isto é possível graças ao registo INTCONbits.INTxEP. Logo com este principio de funcionamento o microcontrolador consegue ler com sucesso os três canais do encoder.

5.6 PushButtons

Este módulo é responsável pela leitura do estado dos quatro botões e do bottom switch instalado no elevador. A figura seguinte resume o funcionamento do módulo desenvolvido.

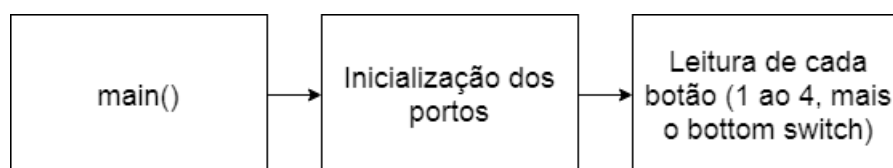


Figure 15: Diagrama de Fluxo da leitura dos botões e do bottom switch

Para isso criaram-se as seguintes funções:

InitButtons(void)

PushButtonx(void)

BottomSwitch(void)

A função **InitButtons** é responsável pela configuração das portas RE3 a RE7 como entradas.

Já as funções **PushButtonx** são as responsáveis por retornar o estado dos botões, 1 a 4 e do bottom switch. Estes mesmos estão ligados às portas RE3 a RE7.

5.7 UART

Este módulo surge da necessidade do sistema comunicar com o utilizador. A figura seguinte resume o funcionamento do módulo desenvolvido.

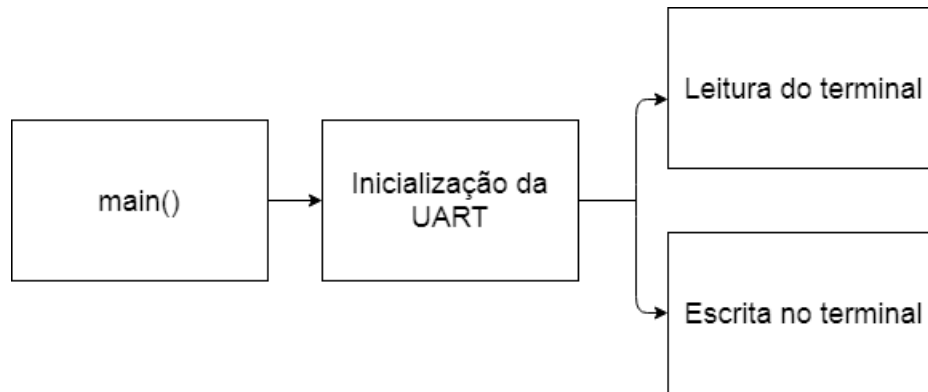


Figure 16: Diagrama de Fluxo da utilização da UART

Para tal foram usadas as funções seguintes:

InitUART(void)

PutChar(char r)

PutString(char *str)

GetChar(void)

A primeira função, **InitUART**, é responsável pela configuração da UART. Para isso é necessário definir a baudrate por forma a calcular o Baud Rate Register, por fim é necessário apenas iniciar a transmissão e a recepção. Para calcular o valor do registo mencionado utiliza-se a seguinte expressão:

$$UxBRG = \frac{PBCLK + 8 * baudrate}{1616 * baudrate} - 1$$

As restantes funções **PutChar**, **PutString** e **GetChar**, estas são responsáveis pelo envio e receção de caracteres na uart. A primeira pela escrita do carácter como argumento, a segunda pela escrita de uma string no formato de array de caracteres e a terceira para a leitura de caracteres do terminal.

6 Testes e Resultados

6.1 Testes da geração dos sinais PWM

Objetivo:

Este teste tem como objetivo verificar a geração de dois sinais PWM, com tensão pico a pico de 5V, para um dado valor de duty cycle com uma frequência de 20 KHz.

Descrição e Resultados Esperados:

Tendo em conta que a ponte H, necessita de uma tensão mínima de 4.5 V, as saídas da Max32 Digilent têm de estar configuradas em open drain, pois com esta configuração consegue-se gerar sinais PWM de 5 V. Apenas se tem de adicionar um pull up resistor ao pino de saída.

Utilizando as funções **StartTimer**, **SetPWM1** e **SetPWM2**, gerou-se na PWM1 um sinal com duty cycle de 50 % e na PWN2 com 30 %.

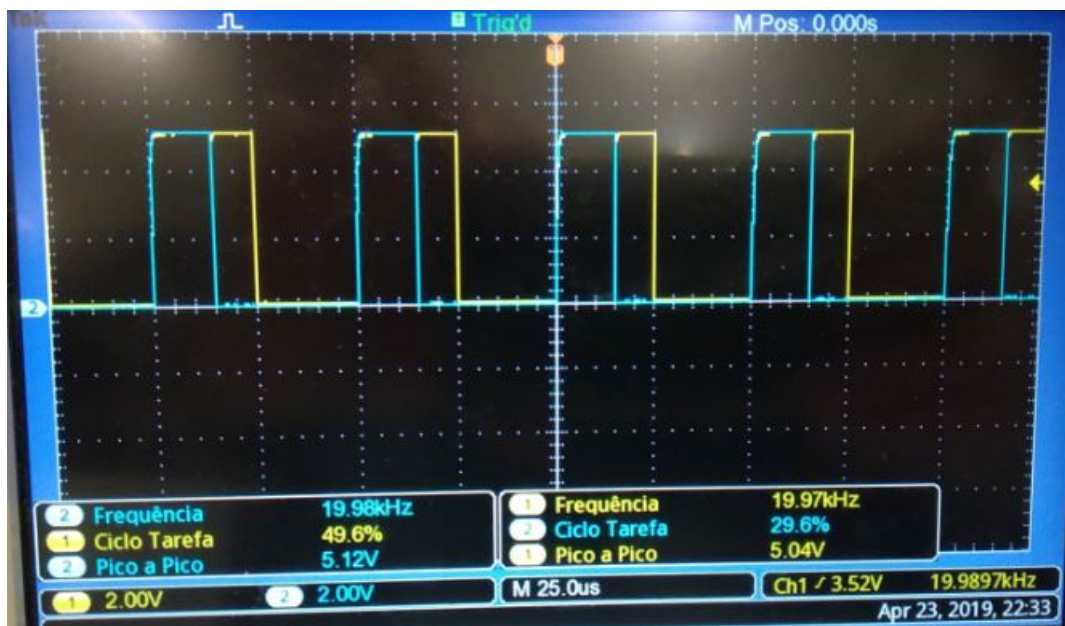
É expectável observar dois sinais PWM com os duty cycles referidos anteriormente e com uma tensão pico a pico de cerca de 5V.

Resultados:

Recorrendo ao osciloscópio da bancada mediram-se os dois sinais PWM gerados, tendo obtido as três figuras seguintes. Na primeira têm-se o sinal PWM1 com duty cycle de 50 %, na segunda a PWM2 com duty cycle de 30 e por fim os dois sinais juntos.



Figure 17: PWN 1 com $\delta = 50\%$

Figure 18: PWN 2 com $\delta = 30\%$ Figure 19: PWM 1 com $\delta = 50\%$ e PWM 2 com $\delta = 30\%$ **Análise:**

Os valores de duty cycle obtidos vão de encontro com o esperado. Quanto à frequência de ambas é cerca de 20kHz e tensão máxima de 5 V como esperado. Portanto conclui-se que os resultados obtidos vão de encontro ao esperado e estão corretos. Sendo que este módulo é capaz de produzir dois sinais PWM à escolha, com qualquer valor de duty cycle e frequência de 20kHz.

6.2 Testes dos sinais obtidos na ponte H e movimento do elevador**Objetivo:**

Verificação do bom funcionamento da ponte H e observação do movimento do elevador.

Descrição e Resultados Esperados:

Com vista em testar a ponte H, foi montada de acordo com o já explicado na secção [Arquitetura de Software - Driver do Motor](#).

Utilizando as funções **StartTimer**, **SetPWM1** e **SetPWM2** geraram-se dois sinais PWM para várias combinações de duty cycles. Os pares de valores foram 70% e 0% , 0% e 70%, 100% e 0%, 0% e 100% e por fim 50% e 50%.

Quanto aos resultados expectáveis, no primeiro par de valores um sinal PWM com visto anteriormente em [Testes da geração dos sinais PWM - Resultados](#) para o PWM 1 e no PWM2 uma tensão de cerca de 0V. No segundo par de valores o inverso da situação anterior, no terceiro na PWM1 uma tensão constante de 5V e na PWM2 uma tensão de cerca de 0V, no quarto o inverso dos resultados na terceira e por ultimo dois sinais PWM exatamente iguais.

Quanto ao movimento do elevador se o duty cycle da PWM1 é maior que a PWM2 o elevador desce, inversamente se o duty cycle da PWM2 for maior que o da PWM1 o elevador sobe. No caso em que os sinais são iguais o elevador prevê-se não possuir qualquer movimento. Também se prevê que quanto maior o dutycyle da função mais rápido será o deslocamento do elevador, pois maior é o tempo em que o sinal se encontra a ON.

Resultados:

Utilizando as funções já enunciadas na secção [Testes e Resultados - Testes de geração dos sinais PWM](#) e usando vários valores de duty cycle para cada um dos sinais PWM, estes dois sinais foram conectados às entradas na Ponte H. Logo utilizando o osciloscópio da bancada, as saídas da ponte H foram as seguintes:



Figure 20: PWM 1 com $\delta = 70\%$ e PWM 2 com $\delta = 0\%$



Figure 21: PWM 1 com $\delta = 0\%$ e PWM 2 com $\delta = 70\%$

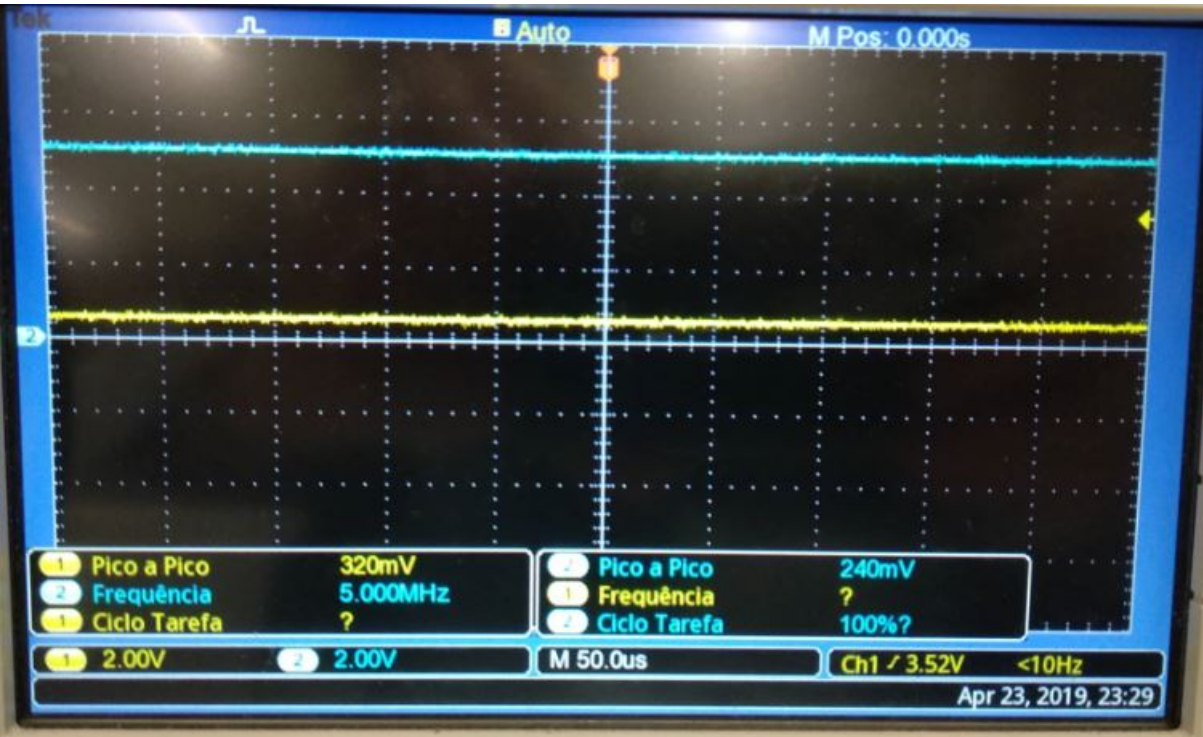


Figure 22: PWM 1 com $\delta = 100\%$ e PWM 2 com $\delta = 0\%$



Figure 23: PWM 1 com $\delta = 0\%$ e PWM 2 com $\delta = 100\%$



Figure 24: PWM 1 com $\delta = 50\%$ e PWM 2 com $\delta = 50\%$

Análise:

Quanto à forma dos sinais os resultados obtidos estão de acordo com as expectativas. Com um duty cycle de 100% obtém-se uma tensão constante de 5V, com 0% um tensão constante de cerca de 0V. De seguida, analisando o movimento do elevador, com as duas ondas com duty cycle de 50% o motor como previsto não

possui movimento pois os dois sinais iram-se anular e o encoder terá sinal resultante nulo. Também se observou que quanto maior o duty cycle da onda, maior será a velocidade do encoder. Quanto ao sentido do seu movimento, se o duty cycle da PWM1 é maior que o da segunda, o elevador desce. Inversamente quando o duty cycle da PWM2 é maior que o da um, o elevador sobe.

Portanto quanto ao movimento do elevador os resultados obtidos vão de encontro com o esperado e estão corretos.

6.3 Testes da leitura da tensão dos sensores de efeito de Hall e deteção de piso

Objetivo:

Verificação do bom funcionamento dos sensores de efeito de Hall, da sua leitura por parte da ADC do microcontrolador ([Arquitetura de Software - ADC](#)), o uso do módulo da UART ([Arquitetura de Software - UART](#)) e finalmente da correta deteção de cada andar pelos sensores de efeito de Hall.

Descrição e Resultados Esperados:

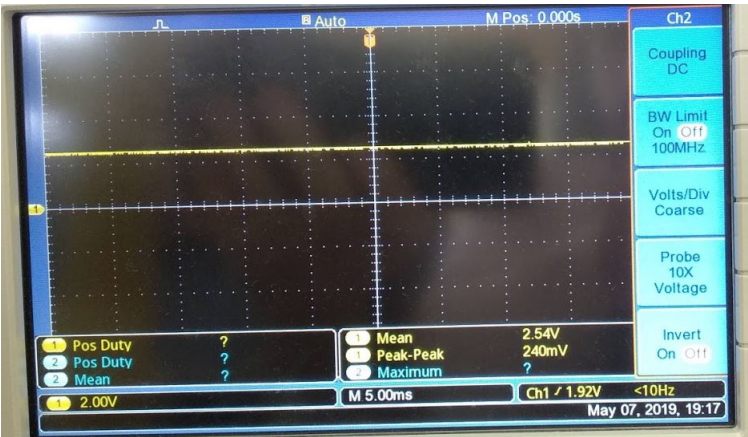
Para testar os sensores de efeito de Hall, foi feita a montagem partindo do esquema elétrico da secção [Descrição do Hardware - Sensor de Efeito de Hall](#). O pino 6 ligou-se a 5V, o 1 à massa e os pinos 5 a 2 foram ligados ao microcontrolador nos portos AN0 a AN3.

Utilizando as funções `initADCTimer`, `initADC`, `buffer` e `InitUART` fez-se o display das leituras no terminal do valor do buffer da ADC. Numa primeira aproximação variou-se a posição do elevador à mão e viu-se a resposta do valor do buffer consoante a proximidade do íman do elevador com os sensores de efeito de Hall. Após algumas tentativas conseguiu-se encontrar uma gama para a qual o valor do buffer é consistente com a proximidade do elevador ao sensor de efeito de Hall. Também se sabe que a tensão de saída referencia dos sensores é 2.5V e com a proximidade do íman instalado a sua tensão baixa para cerca de 2.4 Logo a partir daí já se consegue determinar o andar em que o elevador se encontra graças à leitura da tensão produzida pelos sensores de Hall. Utilizando o módulo da UART fez-se o display do andar em causa, cada vez que o microcontrolador deteta um valor do buffer na gama escolhida.

É expectável que o microcontrolador, consiga fazer, a leitura correta dos sensores, a deteção certa do andar em que o elevador se encontra e que o módulo da UART funcione corretamente.

Resultados:

Utilizando as funções já enunciadas na secção anterior, o Putty para a leitura da UART e fazendo variar a posição do elevador à mão. Obteve-se os seguintes resultados no terminal do computador:



(a) Sem campo magnético



(b) Com campo magnético

Figure 25: Tensão produzida pelo Sensor de Efeito de Hall

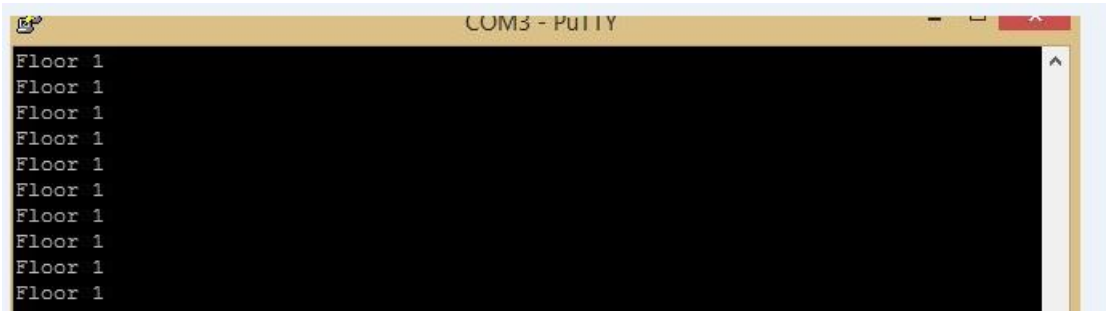


Figure 26: Detecção do Andar 1

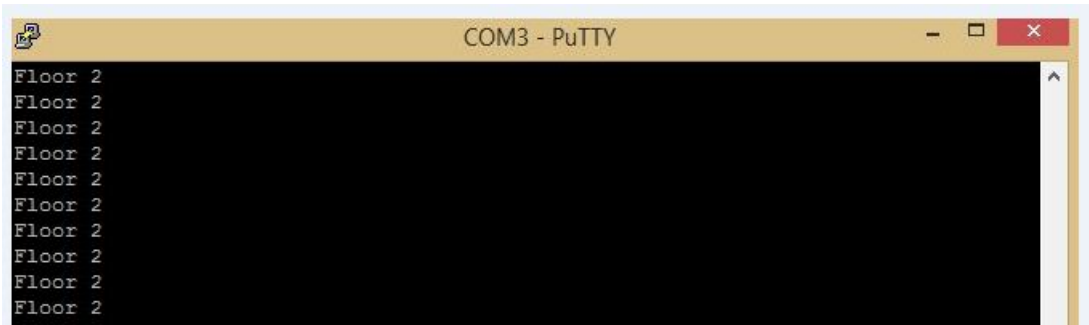


Figure 27: Detecção do Andar 2

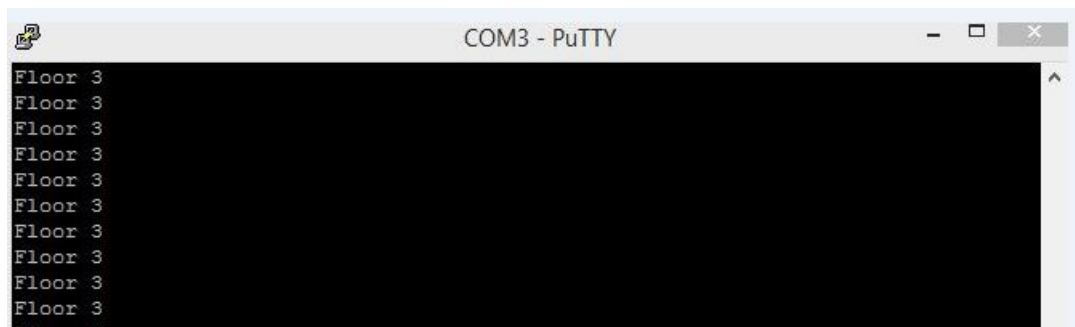


Figure 28: Detecção do Andar 3

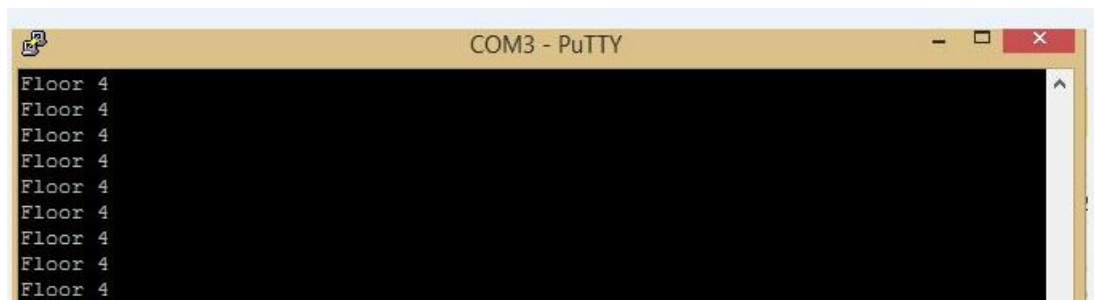


Figure 29: Detecção do Andar 4

Análise:

O microcontrolador consegue localizar com alguma exatidão e precisão o andar em que o elevador se encontra. Na próxima fase iremos aumentar ainda mais a precisão da deteção do piso do elevador.

Portanto os resultados obtidos foram de encontro ao esperado e estão corretos.

6.4 Testes da leitura dos botões e bottom switch

Objetivo:

Verificação da correta leitura do estado dos quatro botões e do bottom switch instalado no elevador.

Descrição e Resultados Esperados:

Para testar os botões, fez-se a montagem partindo do esquema elétrico da secção [Descrição do Hardware - Botões](#).

Utilizando as funções **InitButtons**, **PushButton0**, **PushButton1**, **PushButton2**, **PushButton3** e **BottomSwitch** fez-se a leitura do estado do porto e consequentemente o display do estado de cada botão quando premido pelo utilizador.

É expectável que o microcontrolador, consiga fazer a leitura do estado do porto e o display do estado pelo módulo da UART.

Resultados:

Utilizando as funções já enunciadas na secção anterior e o Putty para a leitura da UART. Obteve-se os seguintes resultados no terminal do computador:



Figure 30: Leitura do Bottom Switch

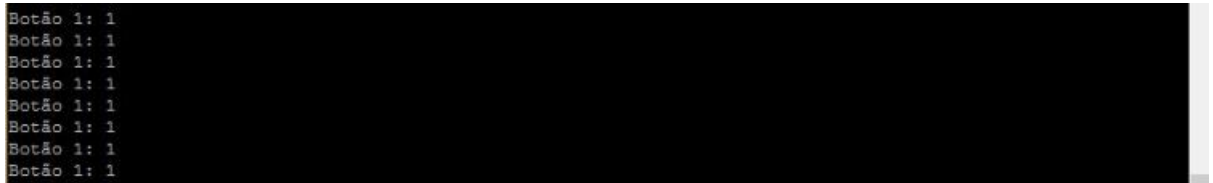


Figure 31: Leitura do Botão 1

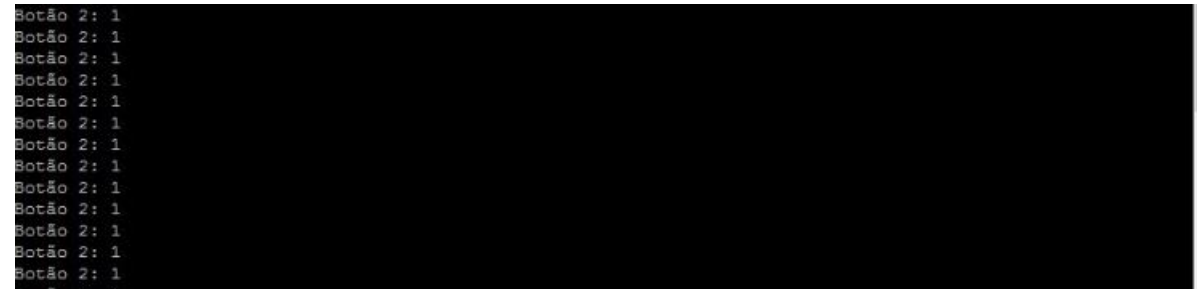


Figure 32: Leitura do Botão 2

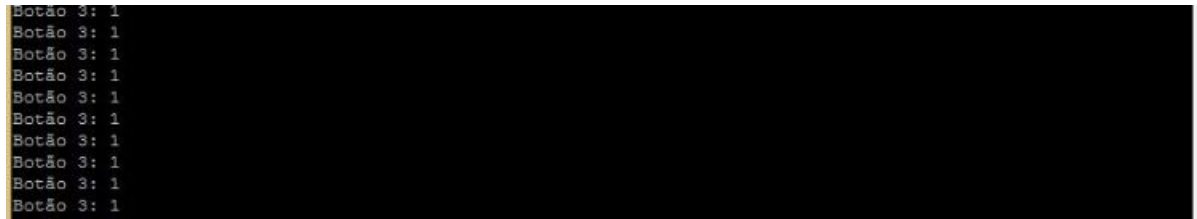


Figure 33: Leitura do Botão 3

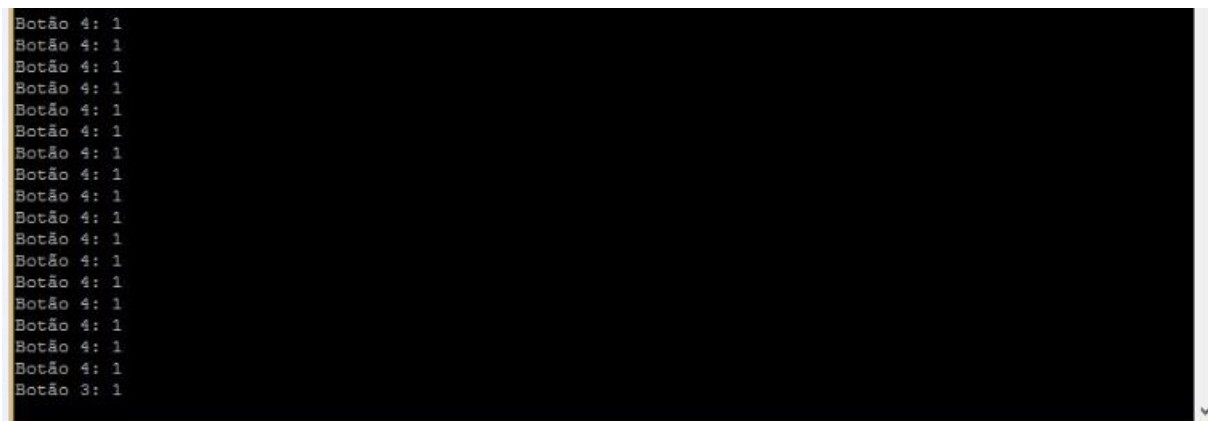


Figure 34: Leitura do Botão 4

Análise:

O microcontrolador consegue com sucesso detetar o estado de cada botão. O que permite na próxima fase interligar o estado do botão com o andar escolhido pelo utilizador.

Concluindo os resultados obtidos foram de encontro ao esperado e estão corretos.

6.5 Testes da leitura dos canais do Encoder

Objetivo:

Verificação da correta leitura do estado dos três canais do Encoder pelo microcontrolador.

Descrição e Resultados Esperados:

Para testar os botões, fez-se a montagem exposta na secção [Descrição do Hardware - Encoder](#), ligando cada canal do encoder a um porto do microcontrolador com interrupções externas.

Utilizando as funções **SetEncoderPorts**, **SetExternalInt0** e **SetExternalInt1**, **SetExternalInt2** fez-se a leitura dos três canais e por fim o display do estado de cada um.

É expectável que o microcontrolador, consiga fazer a leitura correta do estado de cada canal e o display deste mesmo. A interrupção do canal Z apenas vai ser executada quando o elevador deixar de ter movimento, pois nesse instante o estado do canal Z muda. Também se espera que no sentido descendente o sinal do Canal B esteja adiantado em relação ao A, no sentido contrario o sinal do Canal B estará atrasado. Quanto ao Canal Z deve estar a HIGH enquanto o elevador possuir movimento, independentemente do sentido de rotação.

Resultados:

Utilizando as funções já enunciadas na secção anterior, o Putty para a leitura da UART e o osciloscópio da bancada para medir os sinais. Obteve-se os seguintes resultados:

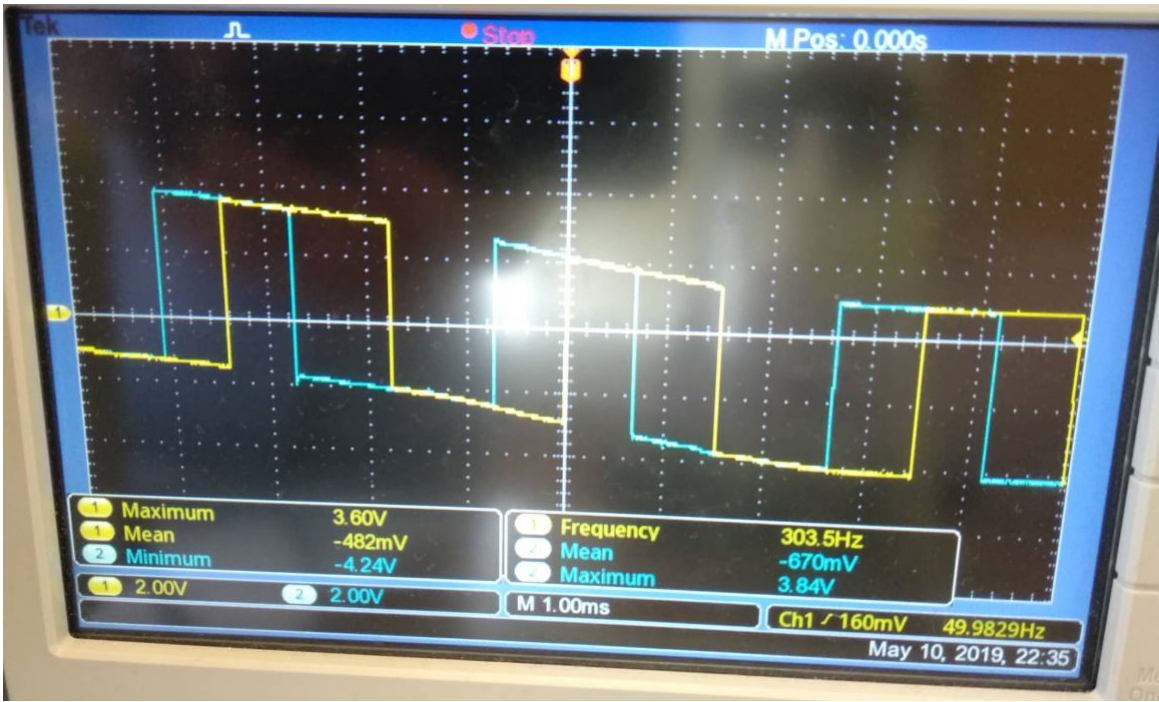


Figure 35: Canal A(Ch1) e B(Ch2), sentido descendente

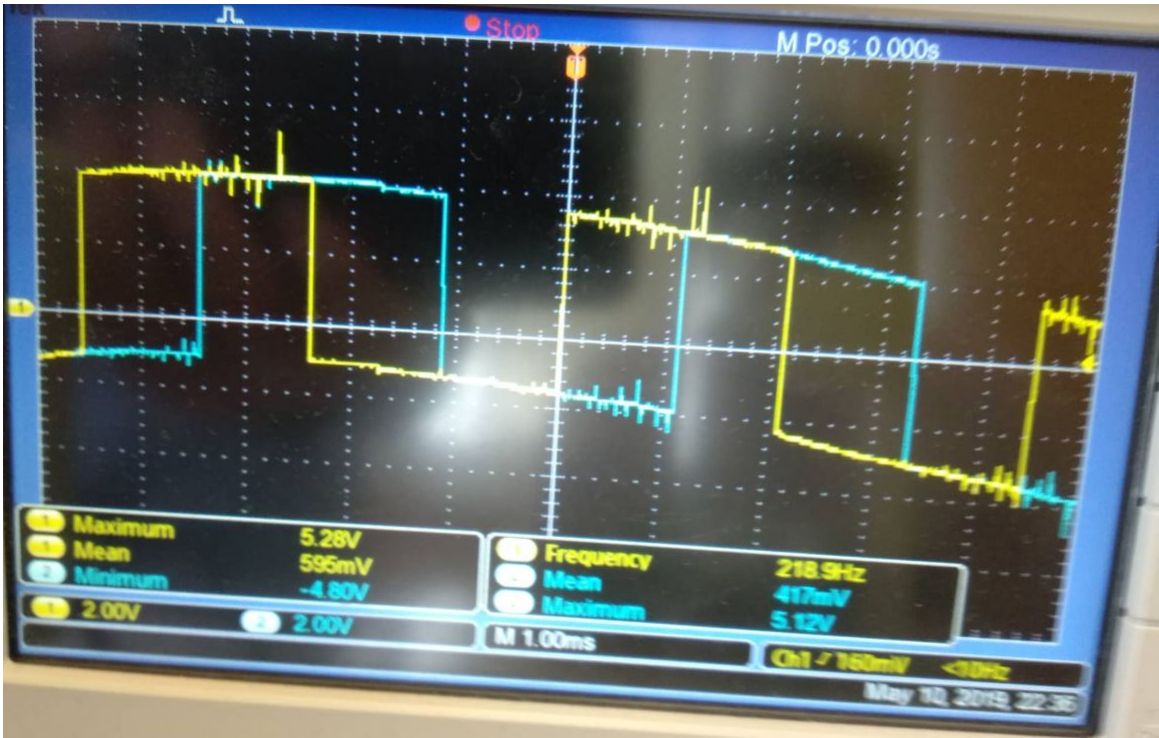


Figure 36: Canal A(Ch1) e B(Ch2), sentido ascendente

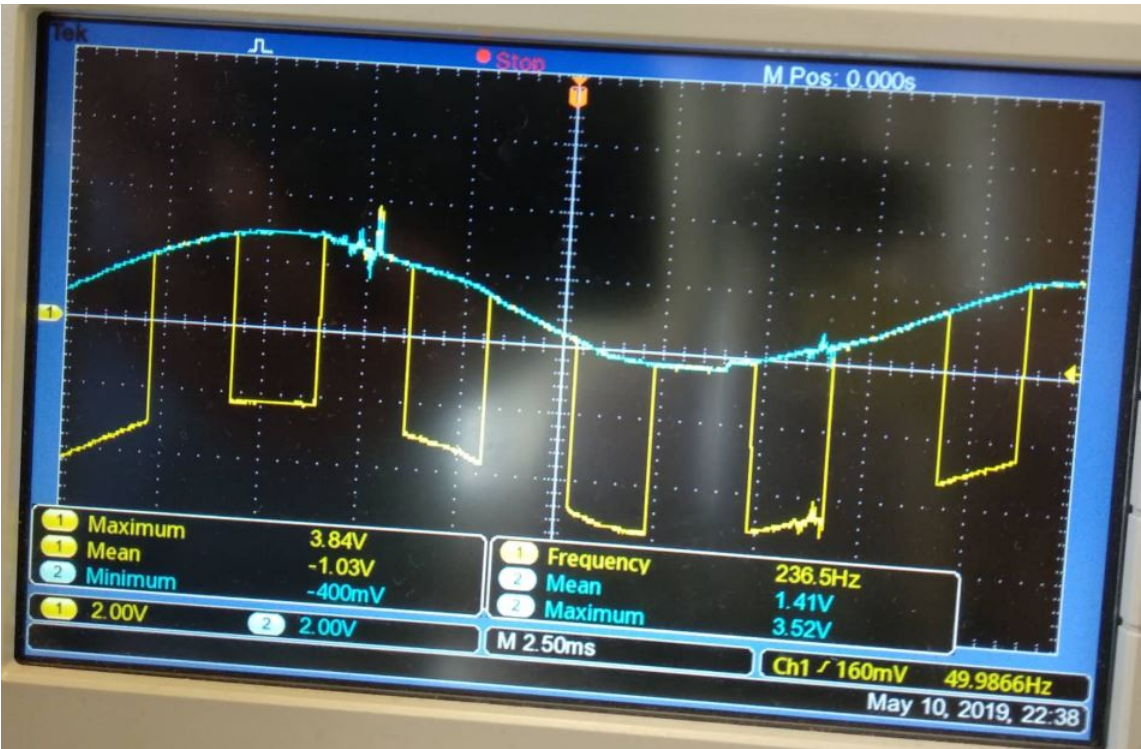


Figure 37: Canal A(Ch1) e Z(Ch2), sentido descendente

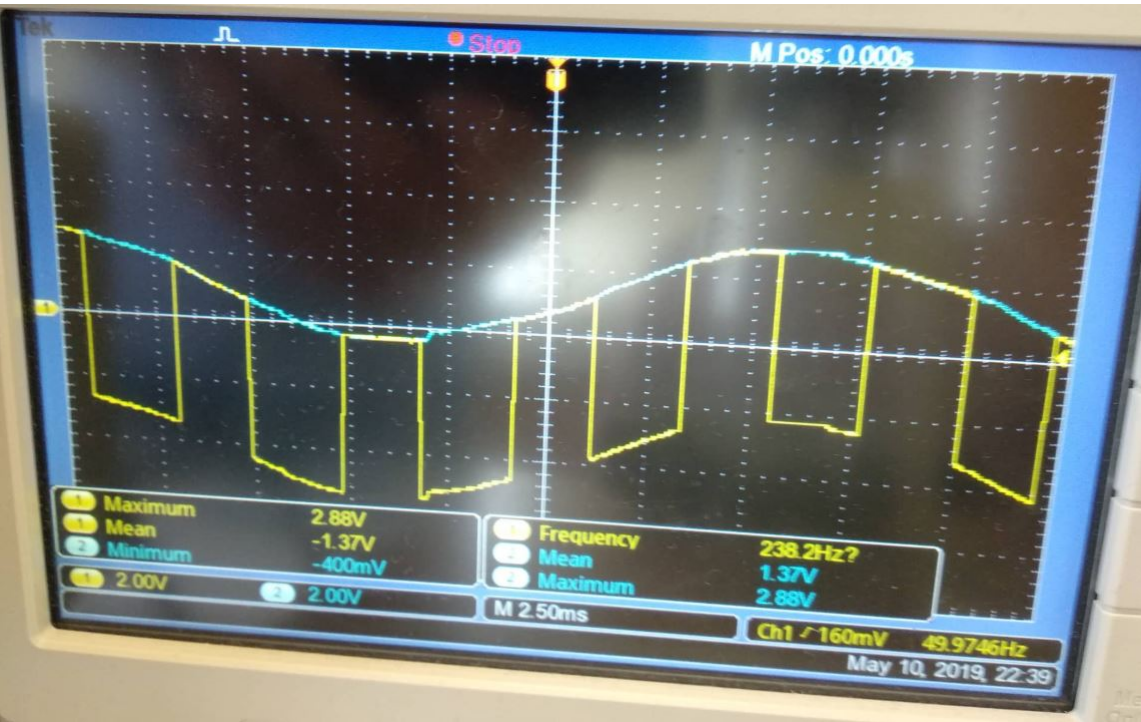


Figure 38: Canal A(Ch1) e Z(Ch2), sentido ascendente

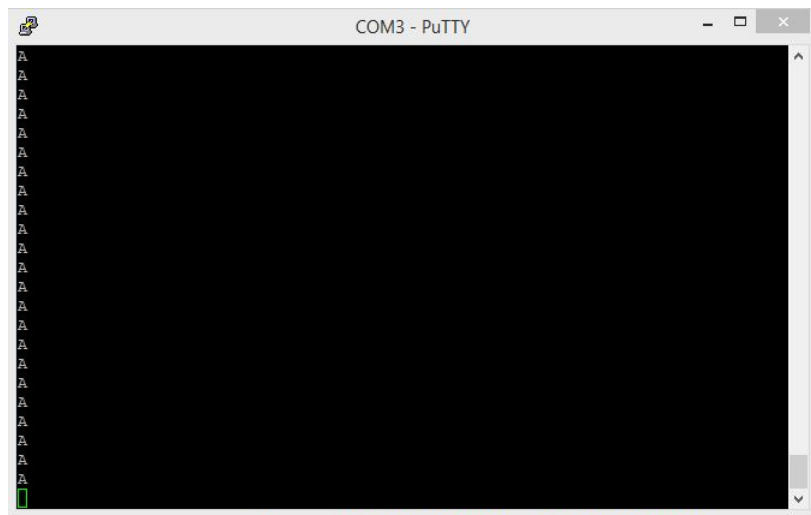


Figure 39: Detecção da interrupção externa do canal A

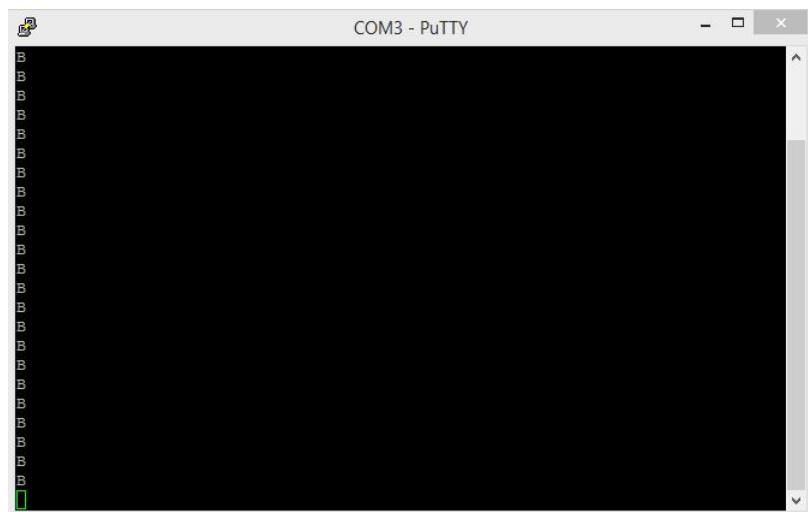


Figure 40: Detecção da interrupção externa do canal B



Figure 41: Detecção da interrupção externa do canal Z

Análise:

Quanto aos sinais lidos pelo osciloscópio vão de encontro ao esperado pelo que existe uma oscilação de 50Hz, que pensamos ser da rede que alimenta a fonte de tensão. O microcontrolador consegue com sucesso detetar

o estado de cada canal do Encoder, pelo que executa as interrupções do Canal A e B e apenas executa a do Canal Z quando se pára o movimento do elevador. A leitura dos canais, permite na próxima fase determinar o sentido de rotação do encoder e por consequência a posição e velocidade do elevador.

Concluindo os resultados obtidos foram de encontro ao esperado, mas a oscilação da fonte de tensão vai ter que ser eliminada para podermos fazer uma leitura o mais correta do estado.

7 Memória Descritiva

O objetivo principal deste projeto é a realização de um sistema que controle a posição e velocidade de um elevador. Para tal a primeira meta que definimos foi analisar o hardware já dado pelo prototipo do elevador e como controlar o encoder. Concluiu-se que o encoder funciona com sinais PWM e recorrendo a uma ponte H consegue-se controlar o sentido do movimento do encoder. Quanto à velocidade, quanto maior o duty cycle do sinal PWM, maior será a velocidade.

Portanto já sabendo como controlar a velocidade e posição do encoder, tem de se escolher um integrado para controlar o encoder. Para isso usando a sugestão do docente usou-se o L293, que é, um componente muito comum no controlo de motores. Analisando o datasheet do componente, evidenciou-se que para funcionar é preciso um sinal TTL com tensão mínima de 4.5 V e é capaz de fornecer uma corrente de 1 A nas suas saídas. Surge a questão de como gerar um sinal TTL de tensão mínima de 4.5V, para isso recorrendo mais uma vez à sugestão do docente, configurando as saídas do microcontrolador em open drain e com um pull up resistor consegue-se produzir sinais com uma tensão de 5V. Desta maneira consegue-se aplicar estes sinais diretamente do microcontrolador para o integrado sem recurso a hardware adicional. Posto isto foi desenvolvido o módulo da **PWM** que cria dois sinais PWM com a opção de escolha do duty cycle de cada uma. Para impor uma frequência aos sinais foi usado o Timer2 para ambas. De seguida fez-se uma montagem simples para o teste dos sinais de acordo com a secção [Descrição do Hardware - Driver do Motor](#). Para boa prática foram adicionados quatro diodos responsáveis por minimizar o efeito dos picos de tensão devidos ao efeito indutor do motor. Foi escolhido o modelo BA157 para os diodos, pois são indicados para elevadas velocidades de comutação. Após os testes bem sucedidos, referidos na secção [Testes e Resultados - Testes da ponte H e movimento do elevador](#) deu-se por concluído o desenvolvimento deste módulo. Foi observado o aumento da velocidade do elevador com o aumento do duty cycle e a variação do sentido do movimento do elevador com os sinais produzidos pelo microcontrolador.

Partiu-se então, para a leitura do estado dos quatro botões e o do bottom switch. A utilidade dos quatro botões será para que o utilizador escolha o nível que deseja e o bottom switch para indicar a permanência do elevador no nível 1. Este último poderá ser útil, no futuro, quando for desenvolvida a rotina da calibração do elevador. Sendo que, alerta o sistema que o elevador se encontra na posição zero dando fim à calibração. Quanto ao software de leitura, foi desenvolvido o módulo **PushButtons**, que de uma maneira simples lê o estado do porto a que cada interruptor está ligado. Este lê o estado HIGH quando os interruptores estão premidos e LOW no caso contrário. Para o display do estado destas entradas é preciso a comunicação via terminal. Logo paralelamente foi desenvolvido o módulo **UART**, que resolve essa necessidade.

Depois virou-se a nossa atenção para os sensores de Hall. Tem-se um por cada andar, num total quatro. Após uma leitura e investigação do funcionamento deste tipo específico de sensores, concluiu-se que a sua tensão varia com a proximidade a um campo magnético, neste caso dado por um íman colado à parte inferior da caixa do elevador. Portanto o microcontrolador vai ter que ler esta variação de tensão, para isso desenvolveu-se o módulo **ADC** que permite a leitura de quatro portas analógicas ao mesmo tempo e com a ajuda do Timer 3 impõe-se uma frequência de amostragem de 400Hz nas leituras. Ligando a alimentação aos sensores e a saída de cada um ao microcontrolador, conseguiu-se ler o valor do buffer de cada entrada e a partir daí, chegar a uma gama de variação para a qual o íman se encontra na proximidade do sensor. Por outras palavras que o elevador se situa no andar específico. Por fim referindo os testes na secção [Testes e Resultados - Testes sensores de Hall](#), o microcontrolador consegue decidir o andar em que se situa o elevador com alguma exatidão e precisão. Como já referido anteriormente, a precisão da leitura vai ser ainda melhorada, na ultima fase do projeto, para obtermos o melhor resultado possível.

Quanto ao funcionamento do encoder, teve-se de pensar numa estratégia para se determinar o sentido de rotação do motor e mais tarde a posição. Para isso após alguma investigação do funcionamento de encoders incrementais, pensou-se em ter uma interrupção externa por cada canal do encoder (A,B e Z). Cada interrupção é iniciada com a deteção de uma raising edge, logo nesse momento o microcontrolador lê o estado do canal e inverte o funcionamento da interrupção para falling edge. Portanto consegue-se ler o estado de cada canal, invertendo o funcionamento da interrupção entre raising edge e falling edge. Fez-se uma simples montagem para ler os canais e deparámo-nos rapidamente com um problema. o canal B e Z possuíam muito ruído, o que faz com que a deteção de mudança de nível seja incerta e não correta pelo microcontrolador. Mais tarde descobriu-se que o setup da nossa sala se encontrava danificado pois não permitia fazer uma leitura sem ruído dos canais. Logo usando o setup da outra sala conseguimos obter leituras corretas dos canais A,B e Z, apesar de estes

estarem com uma oscilação vinda da fonte de alimentação. Quanto ao módulo desenvolvido para o encoder as interrupções externas funcionam corretamente, apenas se tem na próxima fase de eliminar a oscilação de 50Hz dos canais B e Z. A partir daí já temos as rotinas escritas para o cálculo da posição e velocidade do elevador. A rotina para o cálculo da velocidade implementa um timer com uma frequência de amostragem e mede-se a diferença da posição entre esse intervalo de amostragem. A partir daí calcula-se a velocidade sabendo a variação da posição e de tempo. O algoritmo da posição compara o estado do canal A e B, se estes forem diferentes incrementa a posição se forem iguais decrementa a posição.

Por fim o último módulo a desenvolver é o responsável pelo display do andar em que o elevador se encontra e a sua velocidade. O módulo do LCD ainda não se encontra totalmente desenvolvido e funcional, devido à sua complexidade imprevista.

8 Conclusão

Quanto à segunda fase atingimos quase por completo os objetivos impostos pelo guião da disciplina, sendo estes a apresentação dos testes feitos, desenvolvimento do software base na sua totalidade e montagem total do hardware. O módulo que ficou por acabar foi o do display LCD, que acaba por ter menos prioridade do que os outros módulos responsáveis pelo funcionamento pleno do sistema do elevador.

No futuro podem haver alterações no que foi apresentado nesta segunda fase. Pois na fase seguinte, procede-se ao desenvolvimento por completo do sistema de controlo do elevador. As possíveis alterações serão apresentadas, em versões atualizadas no relatório da terceira fase.

9 Bibliografia

Apontamentos de Sistemas de Instrumentação Eletrónica, fornecidos pelo docente.

Esquema do Elevador,

https://elearning.ua.pt/pluginfile.php/962673/mod_resource/content/1/praticas/trabalhos/t4/elevador.pdf

OMRON ,Rotary Encoder E6B2 Datasheet,

https://elearning.ua.pt/pluginfile.php/962676/mod_resource/content/1/praticas/trabalhos/t4/omrom_e6b2.pdf

SGS-THOMSON MICROELECTRONICS, l293B,L293E, Datasheet,

https://elearning.ua.pt/pluginfile.php/962679/mod_resource/content/1/praticas/trabalhos/t4/l293.pdf

HONEYWELL,SS495A-S,Datasheet

<https://sensing.honeywell.com/honeywell-sensing-sensors-linear-hall-effect-ics-ss490-series-\datasheet-005843-2-en.pdf>

Manual de referência online da Max32 Digilent,

https://elearning.ua.pt/pluginfile.php/963675/mod_resource/content/1/chipkit-max32_rm.pdf