

ANA CAROLINA TEZOTO FIGUEIRÔA

**Desenvolvimento de uma plataforma
de força otimizada para análise
biomecânica de usuários de muleta**

Trabalho de Conclusão de Curso apresentado
à Escola de Engenharia de São Carlos,
da Universidade de São Paulo

Curso de Engenharia Elétrica com ênfase
em Eletrônica

ORIENTADOR: Prof. Dr. Alberto Cliquet Júnior

São Carlos
2015

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

F475d Figueirôa, Ana Carolina Tezoto
 Desenvolvimento de uma plataforma de força
 otimizada para análise biomecânica de usuários de
 muleta / Ana Carolina Tezoto Figueirôa; orientador
 Alberto Cliquet Júnior. São Carlos, 2015.

Monografia (Graduação em Engenharia Elétrica com
 ênfase em Eletrônica) -- Escola de Engenharia de São
 Carlos da Universidade de São Paulo, 2015.

1. Plataforma de força. 2. PIC18F2550. 3. USB. 4.
LabVIEW. I. Título.

FOLHA DE APROVAÇÃO

Nome: Ana Carolina Tezoto Figueirôa

Título: "Desenvolvimento de uma plataforma de força otimizada para análise biomecânica de usuários de muleta"

Trabalho de Conclusão de Curso defendido e aprovado
em 25/11/2015,

com NOTA 5.0 (cinco, 310), pela Comissão Julgadora:

Prof. Titular Alberto Cliquet Júnior - (Orientador - SEL/EESC/USP)

Dr. Renato Varoto - (SEL/EESC/USP)

Mestre Renata Manzano Maria - (UNICAMP)

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Dr. José Carlos de Melo Vieira Júnior

Resumo

Algumas condições clínicas exigem a diminuição da carga aplicada sobre os membros inferiores. Assim, são utilizadas muletas, um apoio extra que ajuda o usuário a melhor distribuir esta carga. Esta redistribuição de cargas, porém, não deve ser tal a prejudicar outros membros, como os membros superiores.

Buscando a adequação na distribuição dessas cargas por meio do treinamento do usuário da muleta, propõe-se o desenvolvimento de uma plataforma de força que meça a carga aplicada do membro inferior saudável no solo (força de reação do solo) e envie ao usuário uma indicação quando esta carga é inadequada.

Com a utilização de células de carga para realizar a mensuração das cargas, a plataforma de força também contém um microcontrolador para interpretar o sinal obtido no transdutor e transmiti-lo, via USB, a um programa desenvolvido em LabVIEW. Para auxiliar o desenvolvimento do circuito junto ao microcontrolador e sua programação foi utilizado o programa de simulação Proteus.

Palavras-chave: Plataforma de força; PIC18F2550; USB; LabVIEW.

Abstract

There are certain medical conditions that requires the reduction of load applied on the lower limbs. Thus, crutches are used, an extra support that helps the user to better distribute this load. This redistribution, however, should not be such to affect other limbs, such as the upper limbs.

In order to adjust the distribution of these loads through the crutch user training, it is proposed the development of a force plate to measure the applied load of the healthy leg on the ground (ground reaction force) and send the user a statement when this charge is inadequate.

With the use of load cells to perform the measurement of the load, the force platform also contains a microcontroller to interpret the signal obtained in the transducer and transmit it via USB to a program developed in LabVIEW. To assist the development of the circuit to the microcontroller and its program, it was used the simulation program called Proteus.

Keywords: Force Platform; Load Cell; PIC18F2550; USB; LabVIEW.

SUMÁRIO

Resumo.....	5
Abstract	7
Capítulo 1 – Introdução.....	13
1.1 - Conceitos Básicos e Motivações	13
Capítulo 2 – Instrumentos Utilizados	15
2.1 - Extensômetro	15
2.2 - Célula de Carga.....	19
2.3 – Referência de Tensão - REF01	21
2.4 – Amplificadores – INA 129	22
2.5 – Microcontrolador – PIC18.....	25
2.6 – Interface USB.....	27
2.7 – LabVIEW	31
2.8 – Proteus – ISIS e ARES	33
Capítulo 3 - Realizações	34
Capítulo 4 – Resultados.....	43
Capítulo 5 – Conclusão.....	46
Referências Bibliográficas.....	47
Apêndice	49

LISTA DE FIGURAS

Figura 1: Curva típica da tensão x deformação (Andolfato, Camacho, & Brito, 2004)	16
Figura 2: Extensômetro metálico.....	17
Figura 3: Ponte de Wheatstone - um quarto. Reprodução: (Hoffmann)	18
Figura 4: Extensômetros na configuração de meia ponte. Reprodução: (Hoffmann).....	18
Figura 5: Extensômetros na configuração de ponte complete. Reprodução: (Hoffmann)....	19
Figura 6: Modelos industriais de célula de carga da fabricante Vishay Precision Group....	20
Figura 7: Plataforma de força com três células de carga e o circuito para cálculo.	20
Figura 8: Configuração de pinos do REF01 (Analog Devices, 2015).....	21
Figura 9: Esquemático LTSPice com a montagem do Ref01 com ajuste de saída	22
Figura 10: Amplificador Ina 129 (Texas Instruments, 2009).....	23
Figura 11: Esquemático em LTSpice com a montagem do circuito INA129	24
Figura 12: Diagrama com a configuração de pinos do PIC18F2550 (Microchip, 2009)	26
Figura 13: Padrões de conectores USB (USB International Forum, 2002)	29
Figura 14: Árvore de descrição para dispositivos USB. Reprodução: (Peacock, 2010).....	30
Figura 15: LabVIEW - Painel Frontal (Esquerda) e Diagrama de Blocos (Direita)	31
Figura 16: Tela do esquemático do circuito no programa Proteus	33
Figura 17: Circuito analógico	34
Figura 18: Saída Tensão x Resistência para o circuito em LTSpice da figura 16.....	35
Figura 19: Circuito PIC para simulação no software Proteus.....	37
Figura 20: Tela de interação com o usuário projetada no Labview	39
Figura 21: Diagrama de blocos representando o programa elaborado em Labview	40
Figura 22: Tela inicial do assistente de criação de driver do NI-Visa.....	41
Figura 23: Diagrama de blocos da lógica de programação do PIC18F2550	42
Figura 24: Linhas de comando apresentando os passos necessários para inserção de assinatura	43
Figura 25: Tela do usuário (LabVIEW) quando o peso inserido está correto.	44
Figura 26: Tela do usuário (LabVIEW) quando o peso inserido está incorreto	45

LISTA DE TABELAS

Tabela 1: Valores de Resistência a R_G para obter o ganho desejado (Texas Instruments, 2009)	24
Tabela 2: Características do PIC18F2550 da Microchip.....	26
Tabela 3: Padrão dos pinos USB.....	28
Tabela 5: Principais funções da paleta NI-VISA	32
Tabela 6: Comando simulação LTSpice	35

Capítulo 1 – Introdução

Atualmente, muletas são prescritas e tem uma variedade de aplicações na área de reabilitação da marcha; incluindo melhora do equilíbrio, compensação da musculatura fraca, redução da força sobre uma articulação lesionada e para a melhoria do padrão de movimento da marcha (P. T. Moran, 1995). Em casos de fratura ou pós cirúrgicos, uma recomendação para a fase aguda do crescimento do tecido ósseo é restringir cargas mecânicas na zona de fratura (Hershko, Tauber, & Carmeli, 2008). É comum a recomendação da colocação de peso gradativa conforme ocorre a recuperação. A justificativa para o avanço sustentação de peso é que as cargas podem estimular a atividade osteoblástica em certos padrões de fratura e reconstrução de fixação. (Hustedt et al, 2012).

Segundo Moran (1995) a recomendação geralmente feita ao paciente é que a carga máxima aplicada na muleta seja de no máximo 25% de seu peso. Baseada nesse valor, a orientação para a utilização da muleta é dada ao paciente normalmente por um fisioterapeuta. A técnica mais utilizada para isto envolve instruções verbais ou o uso de uma balança doméstica simples. Hershko, Tauber e Carmeli (2008) afirmam que os resultados de várias pesquisas mostram que os pacientes não conseguem seguir tais instruções sem ter um retorno imediato quanto a carga aplicada. Isto é um fato mais problemático do que aparenta: por exemplo, em pacientes com fratura, a aplicação excessiva de peso pode causar um atraso na recuperação e deve ser evitado, mas a colocação adequada de peso se faz necessária para manter a habilidade de caminhada do paciente e assim acelerar o processo de reabilitação. (Miyazaki & Iwakura, 1978).

1.1 - Conceitos Básicos e Motivações

A limitação de peso aplicado em uma perna é clinicamente prescrita quando se utiliza muletas ou bengalas no processo de cura de alguma fratura ou patologias localizadas abaixo do quadril, joelho e tornozelo, assim como procedimentos pós-cirúrgicos (Kaplan, 2007). Potencialmente, uma única carga ou carga repetitiva acima de um ponto de tolerância pode levar à deformação ou perda de alinhamento ou fixação de uma reconstrução cirúrgica. (Hustedt et al, 2012).

Porém, o usuário de muleta quando solicitado a apoiar apenas uma dada porcentagem de seu peso na perna lesionada, não consegue ter a percepção do quanto isto representa na realidade, podendo aplicar uma carga demasiada neste membro e acabar atrasando a

melhora e até piorando sua condição. Além disso, há o risco de, na tentativa de não ultrapassar este limite, se distribua erroneamente a carga entre o membro saudável e o braço que se apoia na muleta, podendo causar complicações a estes.

Estudos prévios mostram que o cumprimento com relação a instruções sobre o apoio de peso é melhor após um treino com um sistema de *biofeedback* quando comparado a um treino verbal ou com balanças simples (Hurkmans et al, 2003, Hershko et. al, 2008).

Desta forma, visa-se melhorar este quadro com um dispositivo que permita o treinamento do usuário da muleta a partir de uma plataforma de força que analise o peso da pessoa e a carga que esta está aplicando na perna saudável em conjunto com uma muleta instrumentada com a mesma função.

Propõe-se o desenvolvimento de uma plataforma de força de modo a auxiliar o treinamento de novos usuários de muleta, de forma independente do usuário. O sistema deve apresentar uma interface simples, para facilitar sua aplicação e utilização na área médica.

O equipamento consiste de um segmento eletrônico que realizará a medida das forças aplicadas conforme o peso do usuário e emitirá alertas simples quando houver o excesso de carga; e outro de software, onde os dados coletados serão analisados para promover o feedback, além de permitir conexões com outros dispositivos.

Com relação a instrumentação, são utilizadas células de carga compostas de extensômetros em ponte completa e amplificadores de sinais conforme a necessidade destes transdutores. Um microcontrolador, PIC 18F2550, com a função de controlar as interfaces com o usuário, calculando a carga máxima a ser aplicada no membro, assim como interagir com o software de análise, programa desenvolvido por meio do programa LabVIEW (National Instruments), via interface USB. Para a configuração da plataforma, há botões simples para a inserção do peso do usuário, com a exibição deste peso em uma dupla de display de sete segmentos. Além disso, tem-se LEDs e uma campainha para alertar o usuário da distribuição incorreta de forças.

Já o software de análise, feito em LabVIEW (National Instruments), também deve permitir ao usuário a inserção de dados, além de fornecer uma tela de análise mais detalhada do que o *feedback* obtido pelo microcontrolador.

Capítulo 2 – Instrumentos Utilizados

2.1 - Extensômetro

Quando uma força é aplicada em um corpo, este corpo sofre uma mudança em sua dimensão: é a chamada deformação do material (ou Strain). Esta deformação é representada pelo termo ε e pode ser de tensão (positivo) ou compressão (negativo). Seu cálculo é obtido pela variação na dimensão do objeto pela dimensão original, portanto não tem unidade, como pode ser observado (Roylance, 2001):

$$\varepsilon = \frac{\Delta L}{L}$$

Onde ε é a deformação do material, L a dimensão ou comprimento original do corpo e ΔL a variação na dimensão após ação externa.

Já a tensão (*Stress*) é o termo que indica o fator entre a carga aplicada a um certo material com sua capacidade de suportar esta carga: é referente a força por unidade de área (de uma face) em um corpo. A tensão mecânica aplicada perpendicularmente pode ser chamada de tensão normal e é dada pela seguinte formula (Roylance, 2001):

$$\sigma = \frac{F}{A} [Pa]$$

Onde σ é a tensão mecânica, F a força aplicada e A a área da face.

Ao aplicarmos uma força, incrementalmente, em uma barra feita de um dado material, medindo a tensão e a deformação, podemos obter o diagrama experimental da Tensão pela Deformação também chamada de curva de deformação do material, como apresentado na Figura 1.

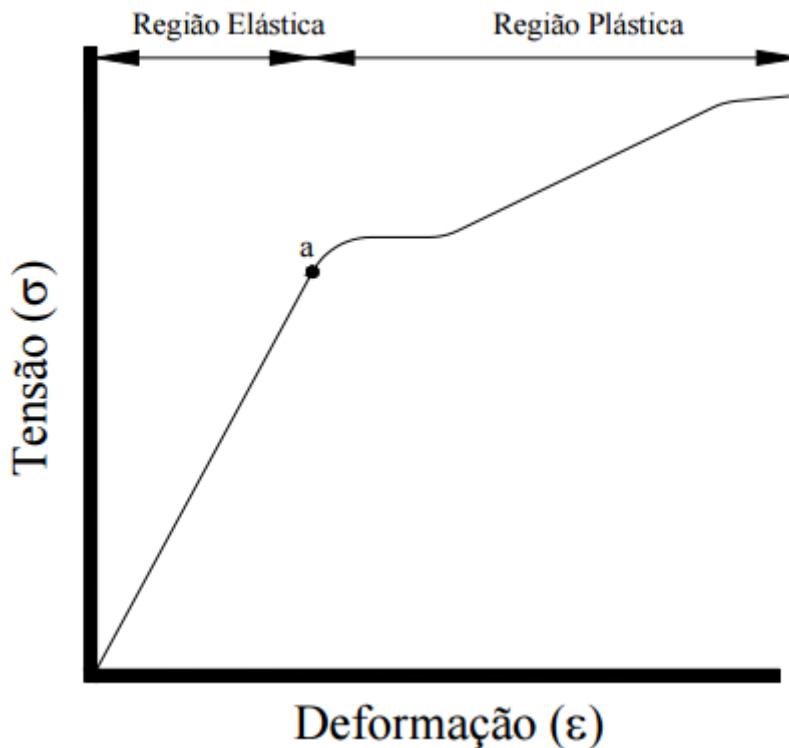


Figura 1: Curva típica da tensão x deformação (Andolfato, Camacho, & Brito, 2004)

É possível observar que até o ponto A (Figura 1), a curva permanece linear; esta região é, portanto, chamada de região linear. Esta região da curva de deformação de um material é onde seu comportamento é elástico, isto é, caso a força aplicada cesse, o material volta a sua forma original. (Agilent Technologies, 1999)

A partir do ponto A a deformação passa a ser permanente e o comportamento do material deixa de ser elástico, e sim plástico; é a faixa em que a curva deixa de ser linear. Para valores elevados de tensão, pode ocorrer a ruptura do material. (Agilent Technologies, 1999)

O Extensômetro se utiliza do princípio da deformação do material para medir a tensão aplicada: ele mede este deslocamento relativo do sólido variando sua resistência proporcionalmente. Há mais de um tipo de extensômetro existente, como o Extensômetro piezoresistivo, extensômetros de resistência de carbono, assim como o extensômetro metálico.

O extensômetro metálico consiste em um filamento ou uma película de metal, a qual é colocada junto ao material a ser deformado, em conformidade com o sentido da tensão aplicada. A mensuração é possível quando este filamento varia sua resistência conforme o material é submetido a esforços de tração ou compressão. Sua configuração é normalmente como mostrada na Figura 2.

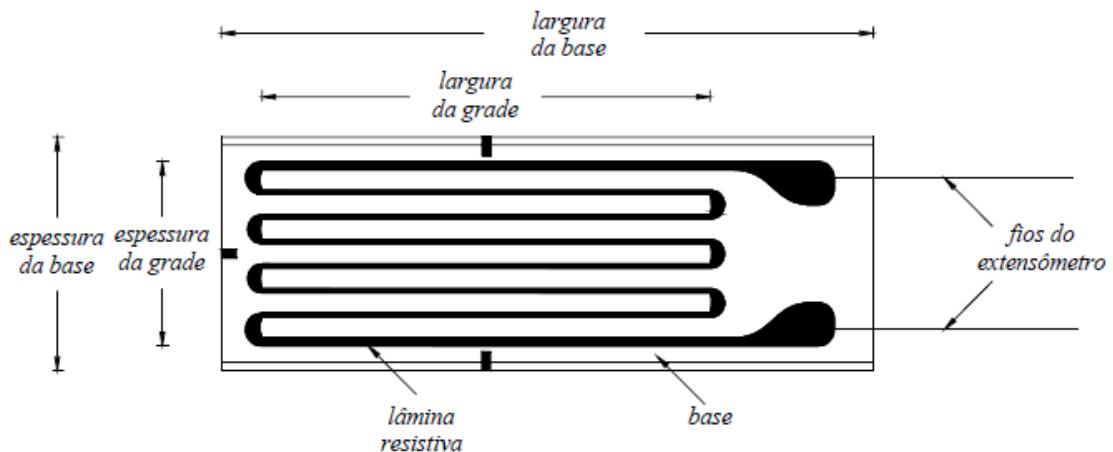


Figura 2: Extensômetro metálico.

Reprodução: (Andolfato, Camacho, & Brito, 2004)

Ao arranjar os filamentos desta forma, é possível reduzir as tensões de cisalhamento e Poisson, não desejadas na mensuração, permitindo ao transdutor medir apenas a tensão aplicada em um sentido, sendo as medidas para forças aplicadas na transversal não significativas.

Extensômetros são normalmente arranjados na forma de ponte de Wheatstone, podendo ser nas configurações de um quarto de ponte (um extensômetro; Figura 3), meia ponte (apenas dois extensômetros; Figura 4) ou ponte completa (quatro extensômetros; Figura 5). As configurações de meia ponte e ponte completa concedem uma sensibilidade maior em relação a configuração de um quarto de ponte (Automatic Control, Robot and Mechatronics Labs, 2007).

O funcionamento ocorre da seguinte forma: aplica-se uma tensão V_o nos terminais U_E , e mede-se a tensão subsequente nos terminais U_A . Caso a proporção entre as resistências seja unitária, a tensão de saída nos terminais U_A será 0V; pois a ponte está em

equilíbrio. Qualquer mudança no equilíbrio desta ponte irá resultar em valores de tensão diferentes na saída. Caso uma destas resistências seja um extensômetro, como na Figura 3, a aplicação de uma tensão no objeto a que o extensômetro está acoplado resultará na sua deformação, fazendo com que sua resistência varie, e a ponte saia do equilíbrio. Podemos então obter uma relação entre a tensão no terminal de saída da ponte com a carga aplicada ao objeto atrelado ao extensômetro.

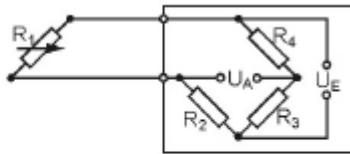


Figura 3: Ponte de Wheatstone - um quarto. Reprodução: (Hoffmann)

Um dos problemas na precisão de um extensômetro na configuração de um quarto de ponte é devido a variação de temperatura: ela causa uma variação na resistência e consequentemente na mensuração de carga aplicada. Uma forma de contornar este problema é colocando um segundo extensômetro na ponte (Figura 4) (Andolfato, Camacho, & Brito, 2004). Este segundo extensômetro deve ser cego: é colocado em uma posição transversal ao extensômetro primário, de forma que a carga aplicada a um objeto recaia sobre seu eixo insensitivo. Enquanto o primeiro extensômetro sofre os efeitos da carga, o segundo permanece alheio a este efeito; porém ocorrendo variações de temperatura, ambos os extensômetros sofrem estas variações na mesma proporção: de forma que devido a configuração da ponte, o efeito sofrido devido a temperatura por extensômetro compensa o efeito de outro, reduzindo os efeitos da temperatura no circuito.

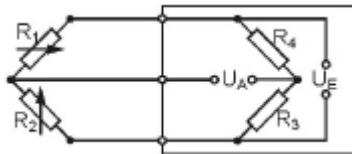


Figura 4: Extensômetros na configuração de meia ponte. Reprodução: (Hoffmann)

A utilização da configuração em ponte completa é a mais recomendada, pois esta forma tem o dobro da sensibilidade de um circuito em meia ponte e quatro vezes a de uma ponte com apenas um extensômetro. Não é apenas a melhor por ter uma maior sensibilidade em relação as outras configurações, mas porque é linear, enquanto as outras não o são (Automatic Control Lab of Mechanical Engineering Department, 2007).

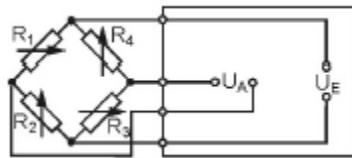


Figura 5: Extensômetros na configuração de ponte completa. Reprodução: (Hoffmann)

2.2 - Célula de Carga

Célula de carga é um transdutor para medição de força, torque ou pressão. Existem mais de uma categoria como as piezoeléctricas, hidráulicas, pneumáticas e a mais comum: com extensômetro. Enquanto o extensômetro transforma a deformação em sinal elétrico, a célula de carga é uma estrutura completa que inclui extensômetros posicionados de forma a medir a força ou pressão aplicada na estrutura corretamente.

Células de carga são utilizadas para a construção de aparelhos de medição de carga e forças. A Figura 6 apresenta dois modelos de um mesmo fabricante: o modelo 1024 e o modelo 120. O modelo 1024 é feito de alumínio e tem baixo custo, é utilizado principalmente na área de varejo, em balanças de bancada e contagem de preço (VPG Transducers, 2014). Já o modelo 120 é feito de aço inoxidável e utilizado principalmente em silos e balanças de caminhão (VPG Transducers, 2015).



Figura 6: Modelos industriais de célula de carga da fabricante Vishay Precision Group
(VPG Transducers, 2014)

Enquanto normalmente são escolhidas quatro células de carga como apoio da plataforma de força, neste caso foram escolhidas três células de cargas em uma configuração triangular, como apresenta a Figura 7. As células de carga utilizadas são de formato cilíndrico, assim como o modelo 120; porém com uma capacidade de medição de peso dentro da faixa desta aplicação.

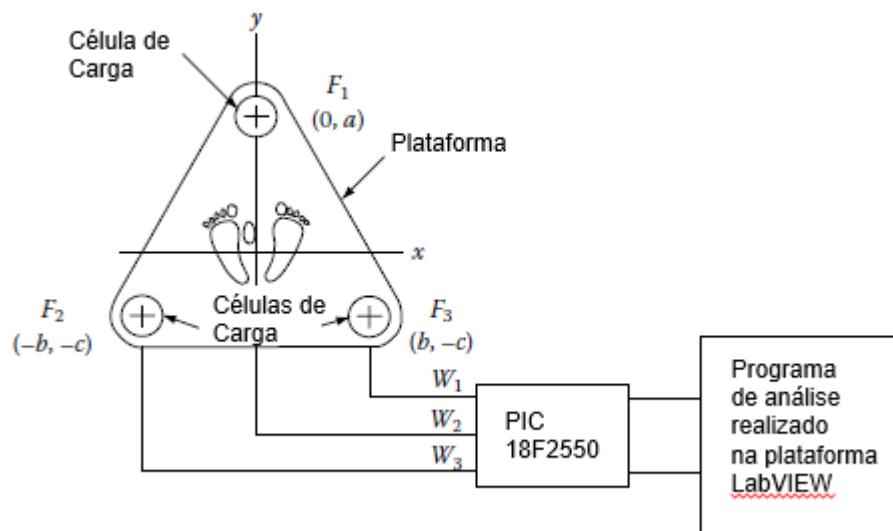


Figura 7: Plataforma de força com três células de carga e o circuito para cálculo.

2.3 – Referência de Tensão - REF01

O REF01 é um dispositivo de referência de tensão; ele fornece uma tensão estável de 10V com variações mínimas quando há mudanças de temperatura, na alimentação ou carga. (Analog Devices, 2015) Isto o torna ideal para utilização na alimentação da célula de carga; pois a ponte de extensômetros deve ter uma alimentação constante a fim de se ter uma medição mais precisa.

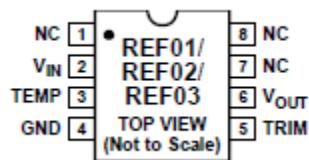


Figura 8: Configuração de pinos do REF01 (Analog Devices, 2015)

A Figura 8 apresenta o padrão de configuração de pinos do modelo considerado para o projeto. Os pinos de numeração 1,7 e 8 são de utilização dos desenvolvedores do circuito, não sendo pertinentes ao usuário. O pino de numeração 2 é onde inserimos a alimentação a ser estabilizada; e o pino 6, a saída com a tensão já estabilizada. O pino de número 3 serve para o controle de temperatura; enquanto que o pino numerado 5 serve para o ajuste fino da tensão de saída do circuito. O pino 4 deve ser aterrado.

A Figura 9 apresenta a montagem em LTSpice do circuito de referência de tensão a ser utilizado para alimentação das células de carga.

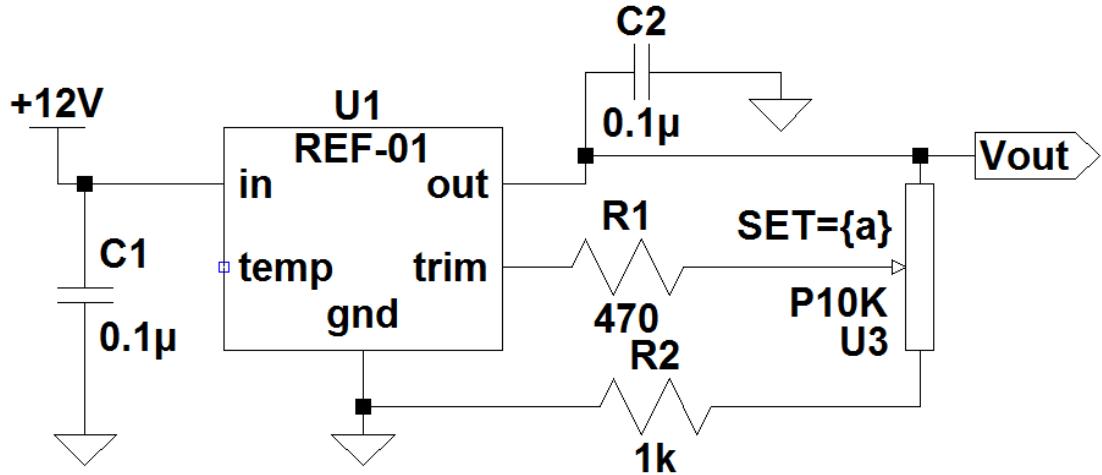


Figura 9: Esquemático LTSPice com a montagem do Ref01 com ajuste de saída

2.4 – Amplificadores – INA 129

INA129 é um amplificador de instrumentação de uso geral, com baixo consumo de energia e alta precisão. Entre as características listadas pelo fabricante, destacam-se:

- Baixa Tensão de Offset: $50\mu\text{V}$ (max)
- Baixa corrente de polarização na entrada: 5nA (max)
- Alto CMR: 120dB (min)
- Entradas protegidas até $\pm 40\text{ V}$
- Ampla faixa de alimentação: $\pm 2,25\text{ V}$ a $\pm 18\text{ V}$
- Baixa corrente quiescente: $700\mu\text{A}$

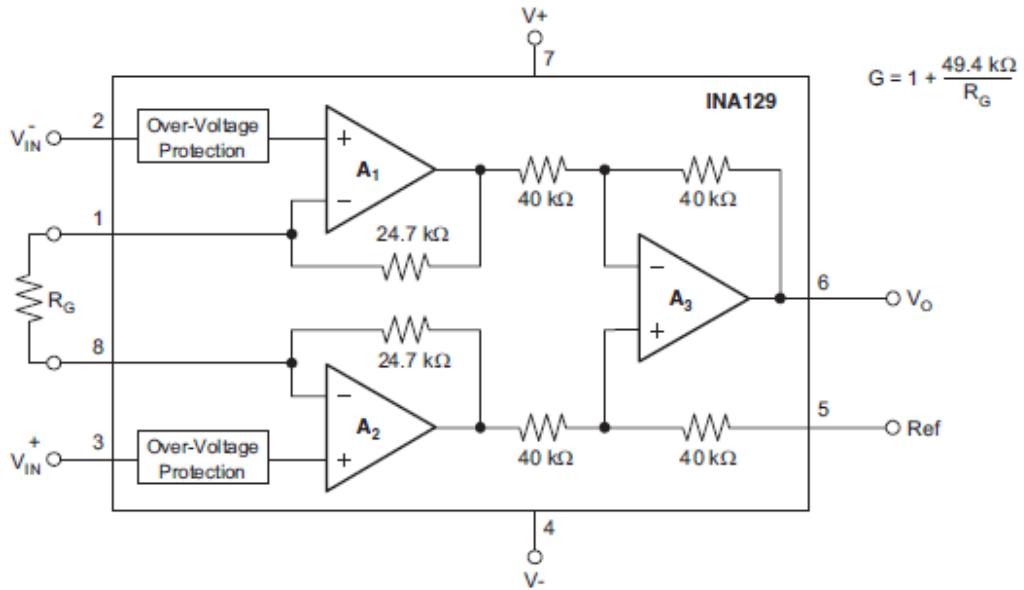


Figura 10: Amplificador Ina 129 (Texas Instruments, 2009)

Ele é composto por três amplificadores operacionais, como apresenta a Figura 10. A partir do valor da resistência externa R_G , é possível configurar os valores do ganho do amplificador: em uma faixa de amplificação de 1 a 10.000. Os terminais V_{in^-} (pino 2) e V_{in^+} (pino 3) são as entradas do amplificador. É aqui que devemos colocar os sinais de saída da célula de carga. Os terminais $V+$ (pino 7) e $V-$ (pino 4) são os terminais de alimentação do circuito integrado. Já os pinos 1 e 8 são os terminais aos quais devemos conectar a resistência R_G para a configuração do ganho do circuito. O terminal V_o (pino 6) é a saída do circuito, onde é obtido o sinal já amplificado. O terminal Ref (pino 5) é a referência do circuito, devendo estar conectada a um referencial nulo. (Texas Instruments, 2009)

A Tabela 1 apresenta o ganho desejado e os respectivos valores de R_G a serem selecionados para se obter a amplificação desejada. Já a Figura 11 apresenta a modelagem em LTSpice de como configurar o INA129 para a amplificação desejada pelo projeto.

Tabela 1: Valores de Resistência a R_G para obter o ganho desejado (Texas Instruments, 2009)

Ganho Desejado (V/V)	R_G (Ω)	Resistência (1%) mais próxima (Ω)
1	-	-
2	49.4K	49.9K
5	12.35K	12.4K
10	5489	5.49K
20	2600	2.61K
50	1008	1K
100	499	499
200	248	249
500	99	100
1000	49.5	49.9
2000	24.7	24.9
5000	9.88	9.76
10000	4.94	4.87

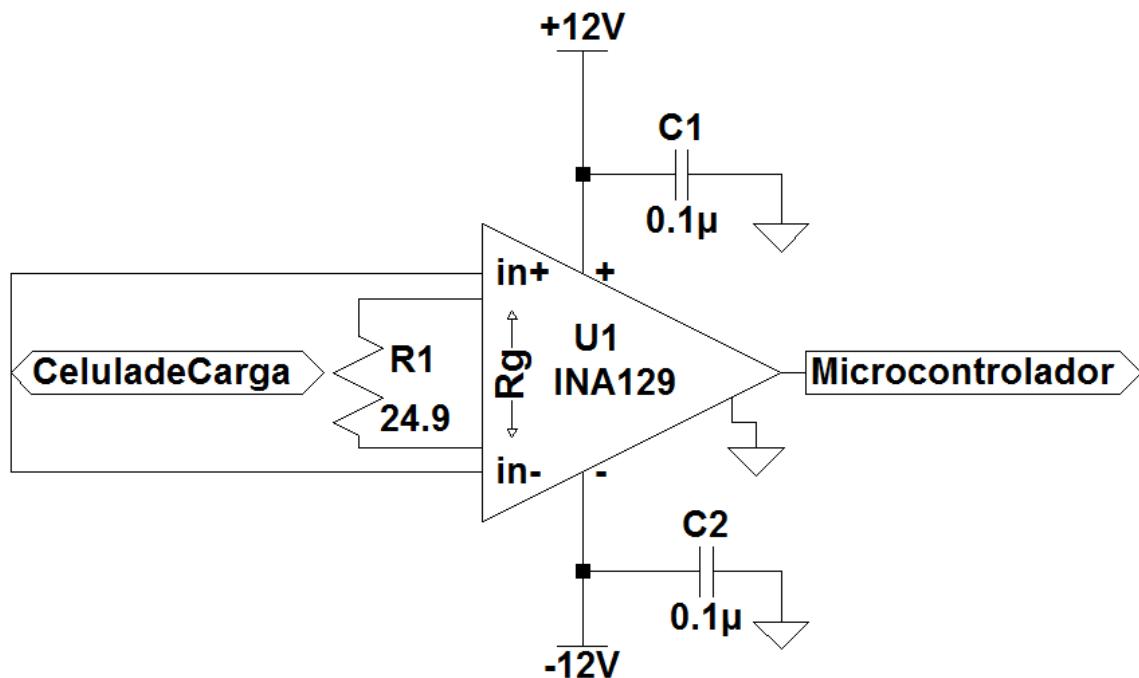


Figura 11: Esquemático em LTSpice com a montagem do circuito INA129

2.5 – Microcontrolador – PIC18

O termo microcomputador é usado para descrever um sistema que inclui no mínimo, um microprocessador, uma memória de programa, uma memória de dados e um dispositivo de entrada e saída (I/O) (Ibrahim, 2012); podendo incluir um ou mais componentes adicionais como temporizadores, contadores e conversores analógico-digitais (ADC).

Um microcontrolador é um tipo de microcomputador criado em um único circuito integrado e necessitando de uma quantidade reduzida de CI's de suporte. Seu princípio é que seja autossuficiente e barato. Não é utilizado com o intuito de processamento de dados, como vemos em computadores pessoais e avançados, e sim um núcleo de processamento para sistemas embarcados. Esta categoria de sistemas inclui pequenos aparelhos eletrônicos que utilizamos no dia a dia, como aparelhos domésticos, televisões e aparelhos portáteis em geral.

Existem diversos fabricantes e modelos de microcontroladores no mercado; uma delas é a Microchip que fabrica a família de processadores denominada PIC (*"Programmable Intelligent Computer"*). Atualmente a Microchip tem as famílias de PIC 8-bit: PIC10, PIC12, PIC16 e PIC18; a com 16-bit: PIC24, dsPIC30 e dsPIC33; e de 32-bit: PIC32. São classificados pela Microchip entre três grupos: de base, medianos e alta performance.

O PIC selecionado para o projeto, PIC 18F2550, da família 18F é considerado de alta performance. Sua configuração é apresentada na Figura 12.

Entre as características que levaram a escolha do PIC2550 é a possibilidade de comunicação via USB 2.0, em modo *full-speed*, que comporta uma velocidade de transmissão de 12Mbps, e torna a conexão fácil para usuários finais. Além disso, o PIC em questão contém o módulo de conversão analógico-digital, não exigindo a colocação de um módulo extra com esta finalidade no projeto; sendo também um dispositivo de preço acessível.

Tabela 2: Características do PIC18F2550 da Microchip

Memória de Programa	32KB
Velocidade Máxima do CPU	40MHz ou 12 MIPS
RAM	2048 Bytes
EEPROM	256 Bytes
Temporizadores	<ul style="list-style-type: none"> • 1 de 8-bit, • 3 de 16-bit
Conversor AD	10 canais, 10-bit
Comparadores	2
Periféricos (Comunicação Digital)	<ul style="list-style-type: none"> • UART • A/E/USART • SPI • I2C1-MSSP(SPI/I2C)
USB	1 canal com velocidade FS (<i>full-speed</i>) USB 2.0
Faixa de temperatura	De -40 °C a 85 °C
Faixa de Tensão	2V a 5.5 V
Pinos	28 pinos

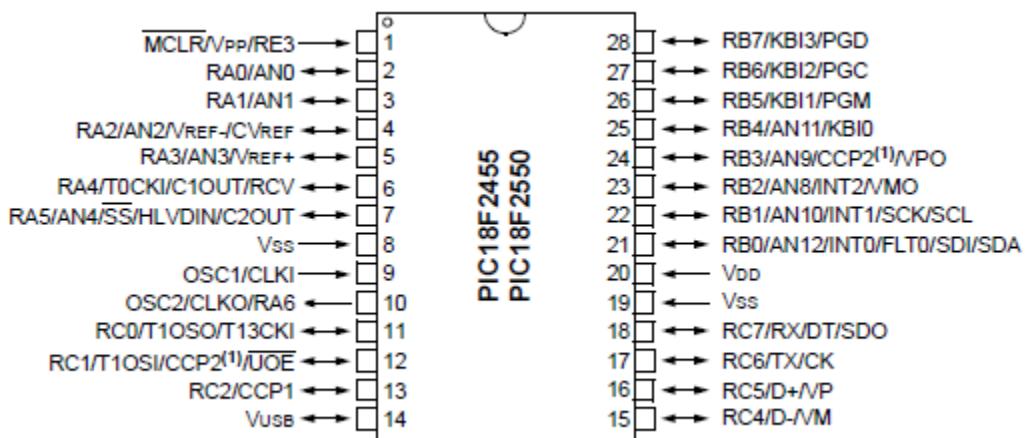


Figura 12: Diagrama com a configuração de pinos do PIC18F2550 (Microchip, 2009)

O conversor analógico-digital (ADC) converte o sinal analógico inserido na entrada do microcontrolador em valores digitais equivalentes. São especialmente úteis em aplicações envolvendo transdutores, pois estes normalmente produzem sinais analógicos. No caso do PIC 18F2550 o conversor é de 10-bit o que significa que teríamos 1024 valores possíveis, pois o número de valores possíveis com 10 bits é de 2^{10} . Selecionando-se os valores de referência para a conversão como:

- $V_{ref-} = 0V$, o valor mínimo, ao ser convertido teríamos 0000000000 (binário)
- $V_{ref+} = 5V$, o valor máximo, ao ser convertido, teríamos 1111111111 (binário)

A resolução deste conversor é obtida da seguinte forma:

$$Resolução_{ADC} = \frac{(V_{ref+} - V_{ref-})}{(2^{adc} - 1)} = 0.004887 \text{ V}$$

Este módulo (ADC) incorpora tempo de aquisição programável, permitindo que o canal seja selecionado e a conversão iniciada, sem esperar o período de amostragem, e assim reduzindo sobrecarga de código (Microchip, 2009).

Para a programação do circuito integrado PIC18F2550 foi utilizada a Linguagem C e o compilador PCWHD, da Custom Computer Services, Inc. (CCS). A linguagem C foi escolhida como alternativa a Linguagem Assembly devido ao fato da programação não necessitar da melhor performance do microcontrolador (o que exigiria linguagem Assembly), além do fato da linguagem C ter um desenvolvimento mais simples.

2.6 – Interface USB

O *Universal Serial Bus* (USB) atualmente é a interface mais comum em dispositivos eletrônicos em geral, pois permite a conexão de periféricos de banda média, como teclados, mouses, *tablets*, modems, telefones, CD-ROM drivers, impressoras, entre outros periféricos de baixa a média velocidade.

O USB inicialmente foi desenvolvido por um aglomerado de empresas que eventualmente criaram uma organização não lucrativa, a USB Implementers Forum Inc. (USB-IF), para padronização da interface. Participam deste fórum as empresas Hewlett-Packard, Intel Corporation, Microsoft Corporation, Renesas Eletronics e STMicroelectronics,

O USB é uma interface com quatro fios, numerados de 1 a 4, cujas cores de funções são padronizadas, conforme mostra a Tabela 3.

Tabela 3: Padrão dos pinos USB

Número do Pino	Cor do Cabo	Função
1	Vermelho	V_{BUS} (5 volts)
2	Branco	D-
3	Verde	D+
4	Preto	Terra

Há dois grupos de conectores padronizados: o tipo A e o tipo B. Os diversos tipos têm a função de evitar conexões perigosas, isto é, que possam danificar algum dispositivo, como por exemplo, encadeamentos com dispositivos não permitidos ou concatenação de cabos (Santos, 2009). O soquete de conector tipo A é normalmente encontrado nos *hosts*; enquanto que o soquete para o conector tipo B se encontra em dispositivos a serem conectados neste *host*. *Hosts* são os anfitriões dos dispositivos: eles iniciam a conexões enquanto os dispositivos apenas respondem ao chamado. (USB International Forum, 2002)

Além dos tipos A e B, existe padrão “mini”, que inclui mini-tipo A e mini-tipo B. Foram criados com o intuito de adaptar a tecnologia USB para o uso em dispositivos menores e portáteis, como os celulares, tablets, entre outros. Todos estes conectores estão retratados na Figura 13: Padrões de conectores USB

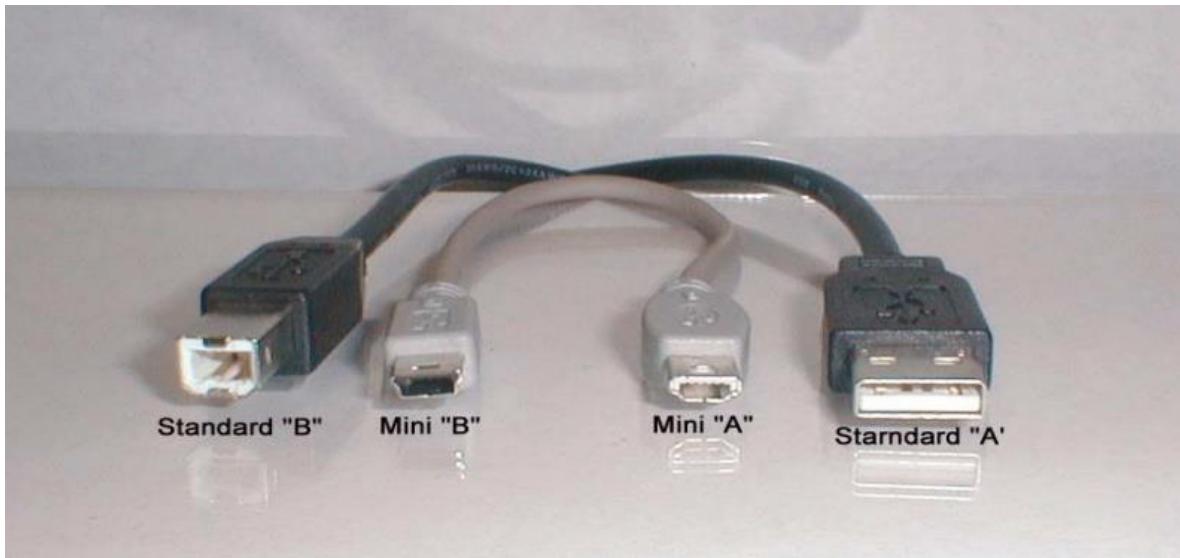


Figura 13: Padrões de conectores USB (USB International Forum, 2002)

A conexão USB contém uma fase chamada de Enumeração: ela é uma fase de identificação que ocorre quando o USB é conectado ao seu *Host*. Ela consiste em uma série de sinais e dados sendo trocados entre o dispositivo e seu *host*, que começam quando o *host* detecta a conexão, inclui a fase de descrição do dispositivo, a alocação dele a um endereço (o qual inicialmente está no endereço 0, dedicado a dispositivos que iniciam a conexão) e termina quando o *host* recolhe informações suficientes para carregar um *driver* adequado ao dispositivo. (Peacock, 2010)

Todo dispositivo USB contém uma hierarquia de descritores; estes servem para descrever as várias funções do dispositivo. Os descritores mais comuns são os de dispositivo, configuração, interface e endpoint.

Um dispositivo USB pode ter apenas um descriptor de dispositivo (primeira camada da Figura 14). Este descriptor inclui informações como qual versão o protocolo USB se refere, a identificação (ID) do produto e seu fornecedor para carregamento dos drivers corretos e o número de configurações que um dispositivo pode ter.

Já a descrição de configuração (segunda camada da Figura 14) especifica valores como a potência que esta configuração exige, se o dispositivo é autoalimentado ou não e o número de interfaces que ele tem neste caso. Apesar da possibilidade de se ter mais de

uma configuração, são raros os dispositivos que as contém; além disso, apenas uma configuração é permitida por vez. Quando o dispositivo é enumerado, há primeiro a leitura do descriptor de dispositivo, para que então o host defina qual configuração será ativada. (Peacock, 2010)

O descriptor de interface é a terceira camada (Figura 14): neste caso, não há limitações sobre quantas podem estar ativas concomitantemente. Ela se refere a funções daquele dispositivo, podendo ativar diferentes interfaces durante sua operação.

Por fim, os escritores de endpoint são utilizados para especificar o tipo de transferência, a direção, o intervalo de polling e o tamanho máximo do packet para cada endpoint.

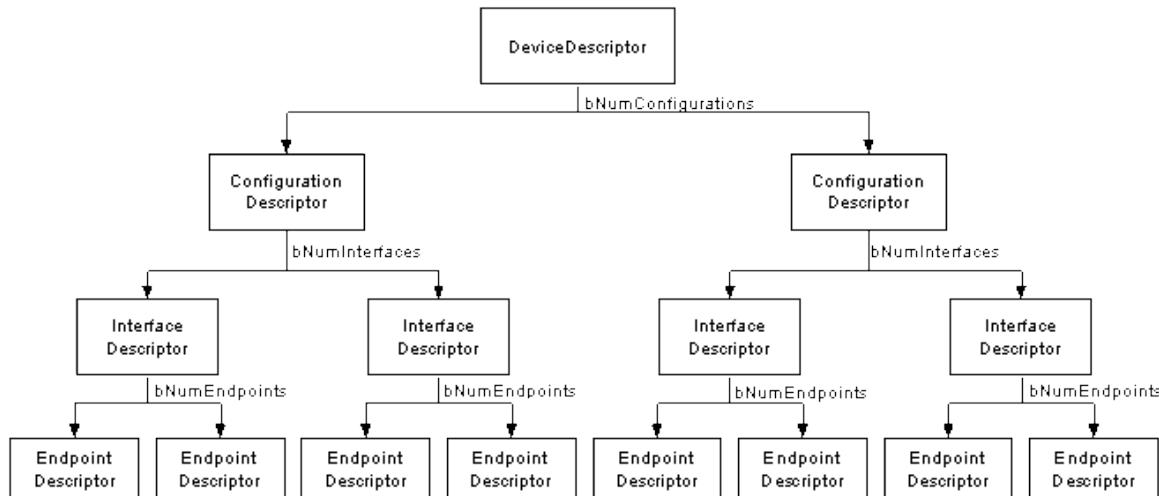


Figura 14: Árvore de descrição para dispositivos USB. Reprodução: (Peacock, 2010)

Há também classes de dispositivos USB que são descritos na **Erro! Fonte de referência não encontrada.**. Um dispositivo dentre estes que é de nosso interesse são os da classe de Dispositivo de Interface Humana (HID- *Human Interface Device*), categoria que inclui, por exemplo, mouses, teclados e controles de videogames. São dispositivos que geralmente ajudam os operadores a se comunicar de maneira fácil com os computadores (Santos, 2009).

2.7 – LabVIEW

LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*) da National Instruments é um programa baseado na utilização de Instrumentos virtuais (VI- *Virtual Instruments*). Fundamentado em uma linguagem chamada linguagem G (*Graphic* ou Gráfica), ele propõe que o usuário utilize versões virtuais (software) de instrumentos físicos (hardware). Esta instrumentação virtual permite que engenheiros e cientistas reduzam o tempo de desenvolvimento, e, portanto, aumentem a qualidade do projeto e reduza seus custos. (National Instruments)

O painel do LabVIEW consiste em dois planos: o Painel Frontal (*Front Panel*) e o Diagrama de Blocos (*Block Diagram*). O painel frontal é onde a interface com o usuário final é apresentada: é possível observar gráficos mostrando sinais tratados, utilizar botões para controlar o programa, entre outras funcionalidades. Já o diagrama de blocos é o código do programa: apresenta as funções, constantes, terminais correspondentes aos itens apresentados no painel frontal, entre outras funções; ele é, portanto, o núcleo do projeto. Ambos são apresentados na Figura 15.

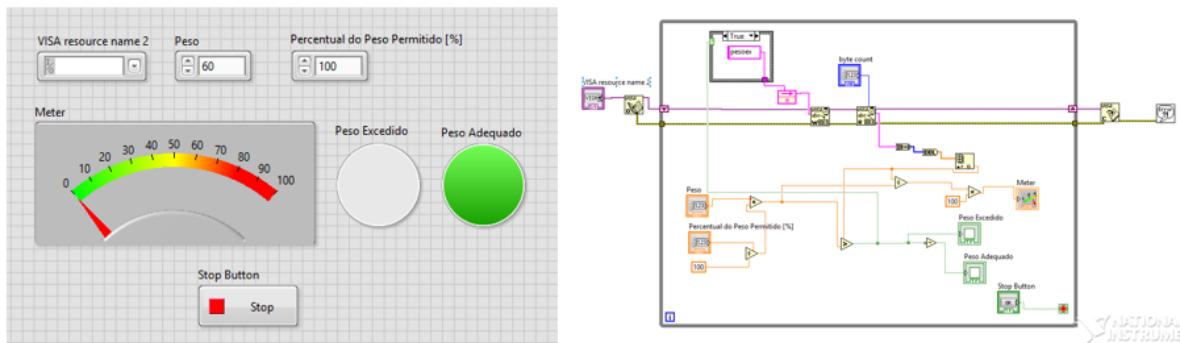


Figura 15: LabVIEW - Painel Frontal (Esquerda) e Diagrama de Blocos (Direita)

Uma das formas de se programar a interface USB com LabVIEW é utilizar um protocolo chamado *Virtual Instrument Software Architecture* (VISA). VISA é um padrão para configurar, programar e solucionar problemas de sistema de instrumentação incluindo

GPIB, VXI, PXI, Serial, Ethernet e interfaces USB. A versão para Labview é nomeada NI-VISA.

Um programa VISA típico seguiria os seguintes passos: (National Instruments, 2009)

1. Abrir uma sessão para um determinado recurso
2. Configurar o recurso escolhido (fixar Baude Rate, caracter de encerramento, etc.)
3. Realizar gravações de leituras no dispositivo
4. Fechar a sessão com o recurso
5. Corrigir erros que possam ter ocorrido

As principais funções presentes na paleta do NI_VISA são:

Tabela 4: Principais funções da paleta NI-VISA

Símbolo	Nome	Descrição
	VISA Open Function	Abre a sessão do dispositivo especificado pela VISA resource name e retorna um identificador de sessão a ser usada para chamar qualquer outra operação daquele dispositivo
	VISA Read Function	Lê o número específico de bytes de um dispositivo ou interface especificado pelo VISA resource name e retorna os dados em read buffer
	VISA Write Function	Escreve os dados do write buffer em um dispositivo ou interface especificada pela VISA resource name
	VISA Close Function	Encerra a sessão ou objeto de evento de um dispositivo especificado por VISA resource name

2.8 – Proteus – ISIS e ARES

Proteus é um software da empresa Labcenter Electronics, que permite a simulação de circuitos, montagem de placas de circuito impresso (PCB). O esquemático do circuito é montado na plataforma ISIS, que inclui modelos de Circuitos Integrados (CIs) de fabricantes como Microchip (PIC), Atmel, ARM, Freescale semiconductor, Texas Instruments, NXP, entre outros. Ele se utiliza de modelos Spice para simulação; mas sua grande vantagem é a possibilidade de simular circuitos como o PIC que são programáveis. Ou seja, pode-se colocar o firmware e o programa (extensão *.hex*) no circuito programável e simular o funcionamento do programa criado.

ISIS é o sistema dentro do Proteus que permite a realização de desenhos esquemáticos, como apresentado na Figura 16. Utilizando este mesmo circuito simulado no programa, pode-se utilizar o programa Ares para criar o layout da placa de circuito impresso para colocar os componentes indicados.

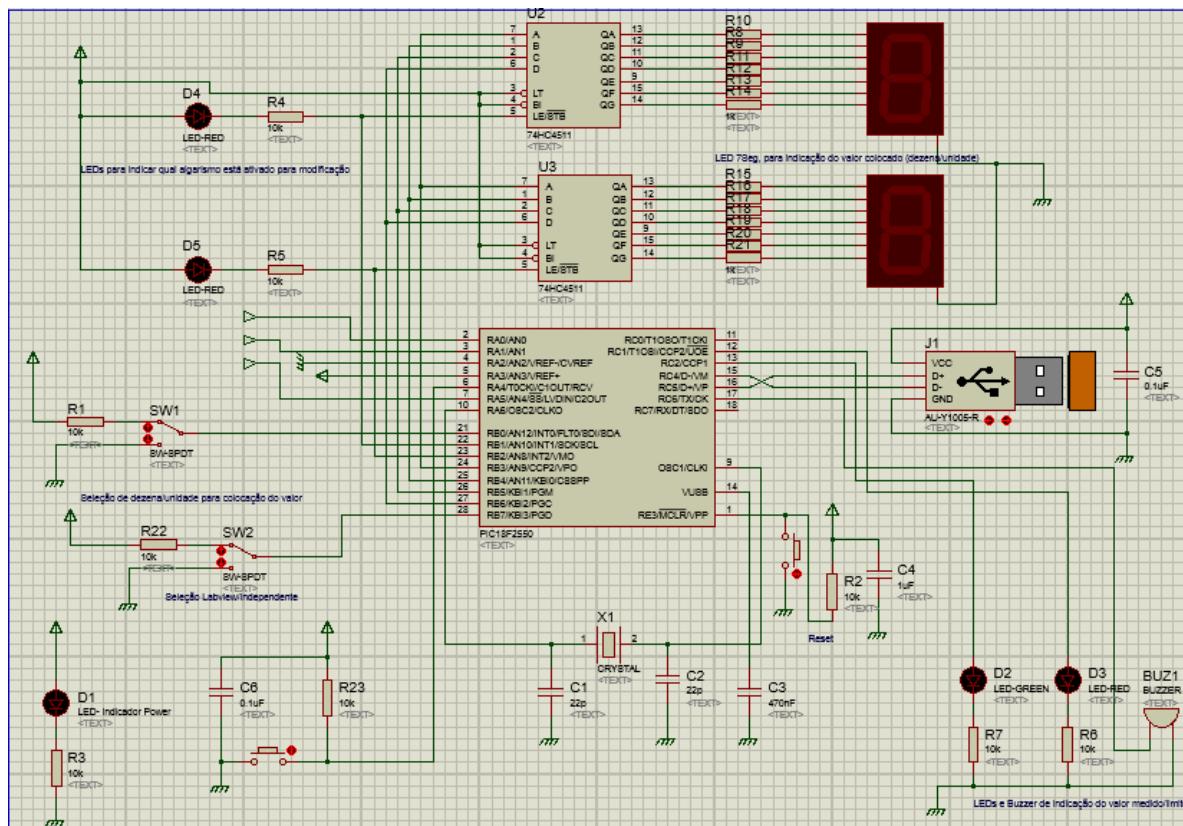


Figura 16: Tela do esquemático do circuito no programa Proteus

Capítulo 3 - Realizações

O projeto consiste em três células de carga, cada qual é composta de quatro extensômetros em ponte completa, sendo alimentadas pela referência REF01, fonte de tensão regular. O sinal de saída é então enviado ao amplificador INA129, configurado para uma ampliação em torno de 2000 V/V, utilizando um resistor de valor 28Ω . O circuito similar ao montado é apresentado na Figura 17; onde resistores fazem as partes da ponte de extensômetros que compõe a célula de carga.

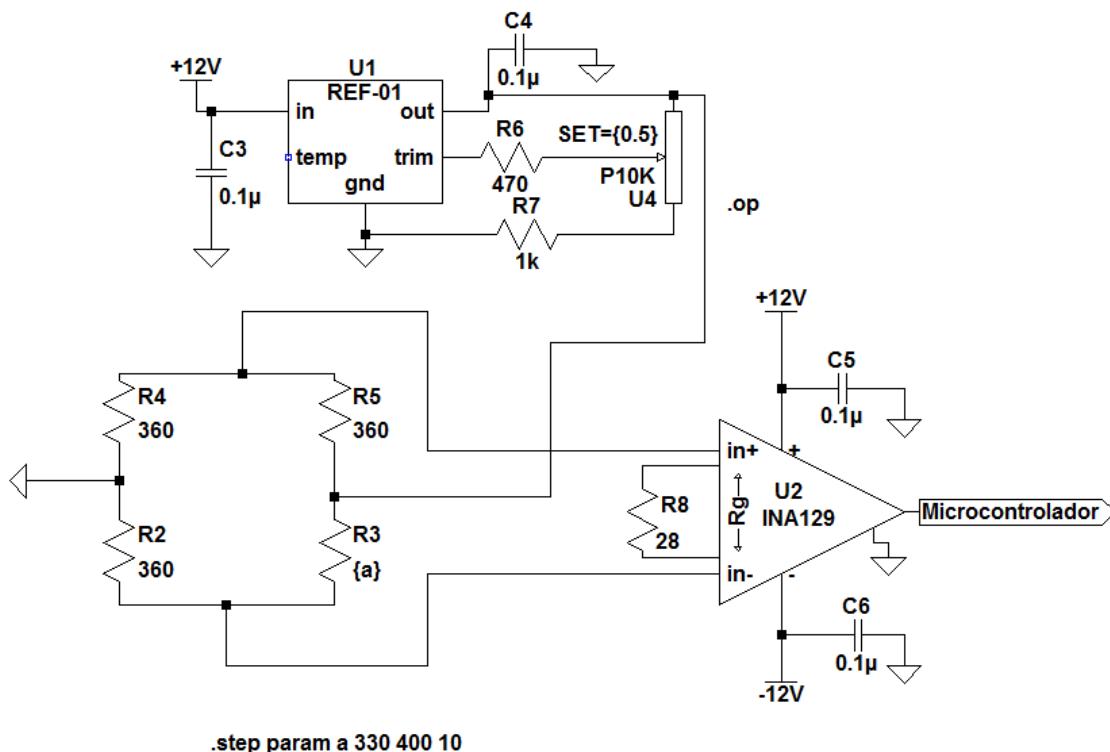


Figura 17: Circuito analógico

Para apresentar a saída de um circuito com um quarto de ponte, ou seja, apenas uma resistência é variável, foi efetuada a simulação de ponto DC, com o comando `.op`, no LTSpice. Variando o valor do resistor $R3(a)$ entre 330 a 400, podemos observar o seguinte efeito na saída do amplificador INA: mostrado na Figura 18. Para esta simulação foram utilizados os comandos:

Tabela 5: Comando simulação LTSpice

.step param a 330 400 10
.op

A simulação não reflete valores iguais ao que um extensômetro real apresentaria, e sim apenas valores ideais, porém, é possível observar que a variação na resistência da ponte reflete na tensão de saída do circuito: o ponto de tensão nula é onde a resistência do componente variável é igual ao dos outros braços da ponte.

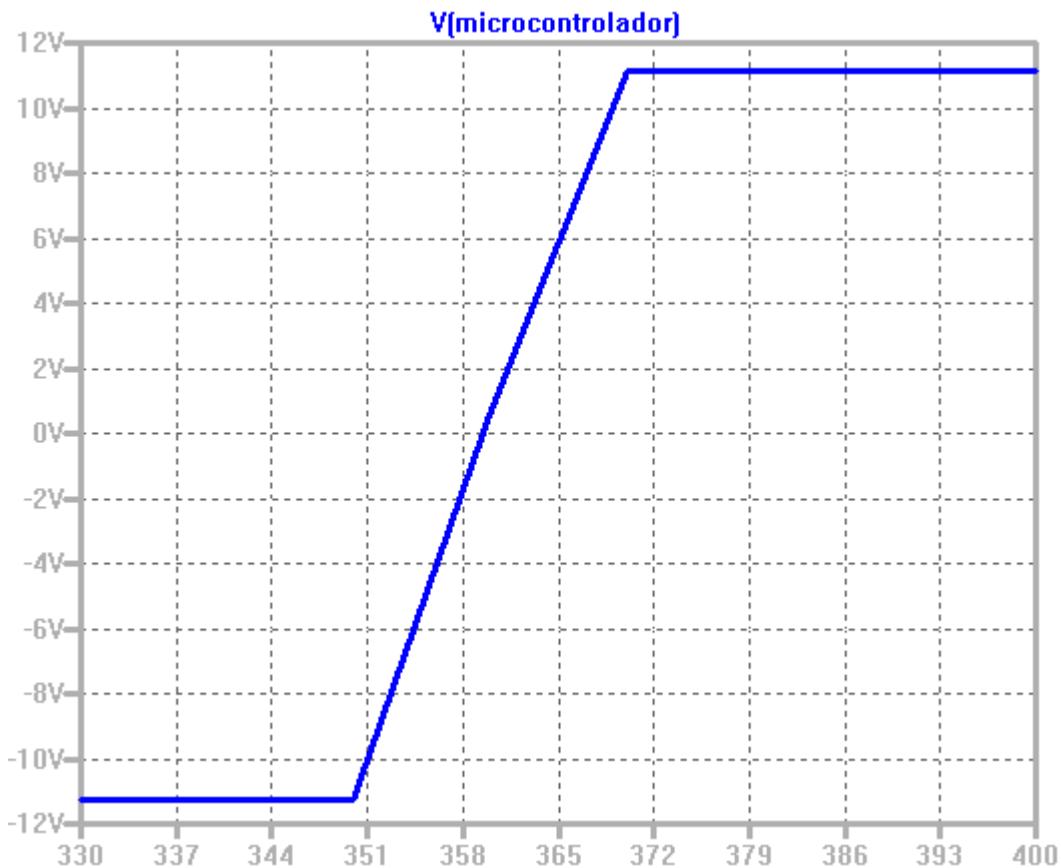


Figura 18: Saída Tensão x Resistência para o circuito em LTSpice da figura 16

Após obter o sinal de cada célula de carga, estes são enviados, então, para o circuito do PIC, como apresentado na Figura 19. O PIC recebe o sinal analógico das três células de carga da plataforma de força, estes sinais são inseridos nas entradas analógicas com

conversor analógico-digital; os valores de amplificação do INA129 foram escolhidos de forma que os sinais permanecessem dentro da faixa de tensão de 0V a 5V para permitir sua inserção no microcontrolador. Deve-se ter a curva de caracterização do transdutor para traduzir os valores de tensão em peso; esta curva é obtida com o auxílio de um dinamômetro, que deve medir a força aplicada aos transdutores e um osciloscópio, que deve medir a tensão de saída com precisão. A soma dos sinais dos três transdutores será o valor da força total aplicada a plataforma de força.

Implementou-se a opção no PIC, por meio de uma chave, de que forma sua utilização possa ser feita sem a necessidade de um computador (LabVIEW); um modo portátil, em que apenas o PIC, seu circuito auxiliar e os sensores seriam utilizados no treinamento do paciente.

Para este módulo independente de computador, o circuito digital do PIC contém a funcionalidade de se inserir o peso do usuário. A colocação do peso deve ser feita por meio de um botão *push-button*, que cada vez que é pressionado, soma-se uma unidade; para indicar se estamos somando na dezena ou na unidade do peso da pessoa, foram colocados dois displays de 7 segmentos –um representando a dezena e outro a unidade – e um LED para cada, indicando qual deles está ativo para inserção de valor. A seleção entre dezena e unidade deve ser feita por meio de uma chave. Há uma terceira chave presente no circuito: ela convém para selecionar se o usuário deseja inserir seu peso (modo configuração) ou utilizar o aparelho (modo operação).

Para auxiliar a utilização dos displays de sete segmentos, que apresentam o peso inserido, foi utilizado o componente 74HC4511. Este componente é *driver*, ou decodificador de código BCD para um display de sete segmentos. BCD ou “*Binary-Coded Decimal*” é um tipo de codificação alfa-numérica, que traduz valores decimais em um formato binário. Neste caso, há a economia de portas de saída, pois um display de sete segmentos exige a inserção de um valor para cada segmento, o que resultaria em sete portas de saída do PIC sendo reservadas para isto. Isto resultou na necessidade de apenas 6 pinos de saída do PIC para utilização do display de sete segmentos: quatro para a programação do valor apresentado e dois para a seleção de qual componente (entre dezena/unidade) está apto a ser programado.

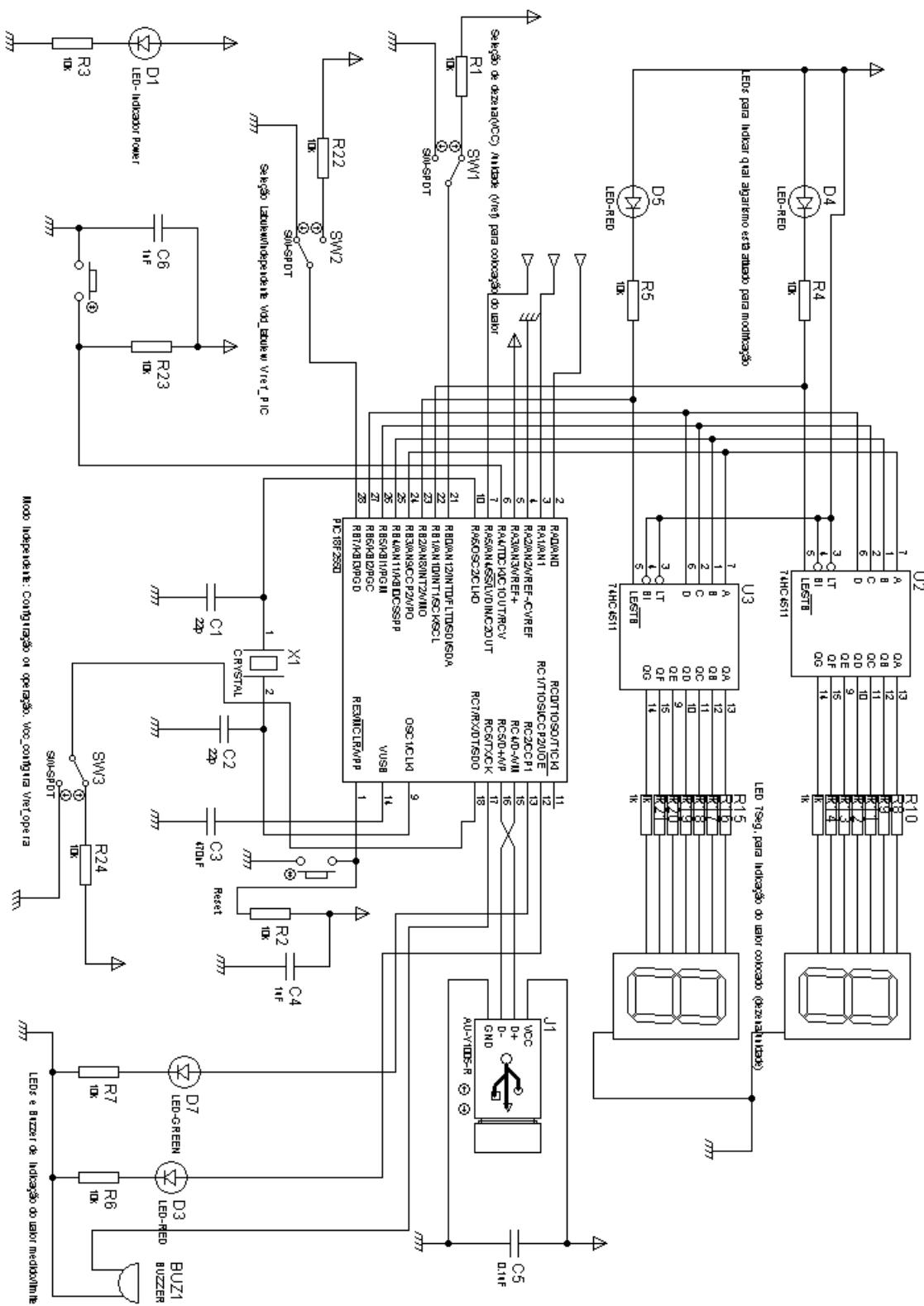


Figura 19: Circuito PIC para simulação no software Proteus

Um cristal de 20MHz foi colocado no circuito, pois para a utilização da porta USB em modo FS (*full-speed*) exige sua utilização para o funcionamento correto. Há também o LED de indicação de alimentação conectada, a saída USB e o módulo de *reset*.

Para o alerta do usuário do aparelho, foram colocados dois LEDs: um verde, indicando a inserção do peso correto, e um vermelho, que indica ao usuário quando ele saiu da faixa de força adequada. Além disso, temos a colocação de uma campainha (“*Buzzer*”) para também alertar o usuário sobre o excesso de peso. O código fonte para programação do PIC é apresentada no apêndice.

Para o modo completo, o PIC deve ser conectado, via porta USB, a um computador em que esteja sendo executado o programa desenvolvido em LabVIEW. Esta interface permite que o usuário selecione, via programa, o seu peso e a percentual que deve gerar alerta. O painel de usuário do LabVIEW apresenta LEDs verde e vermelho, assim como no circuito ligado ao PIC, gerando alertas em relação ao uso.

Foi conectado a um indicador a relação do peso atual com o desejado: ele apresenta ao usuário, em forma de um ponteiro, se o peso colocado está muito abaixo, próximo ou acima do valor desejado. A tela de apresentação para o usuário projetada em LabVIEW é apresentada na Figura 20. No modo de uso com programa foi adicionada a funcionalidade de edição do peso limite do paciente, pois indicações médicas de colocação de carga não são sempre as mesmas, dependendo do tipo de diagnóstico feito.

O diagrama de blocos referente a esta tela, com a programação e lógica por trás deste programa é apresentado na Figura 21.

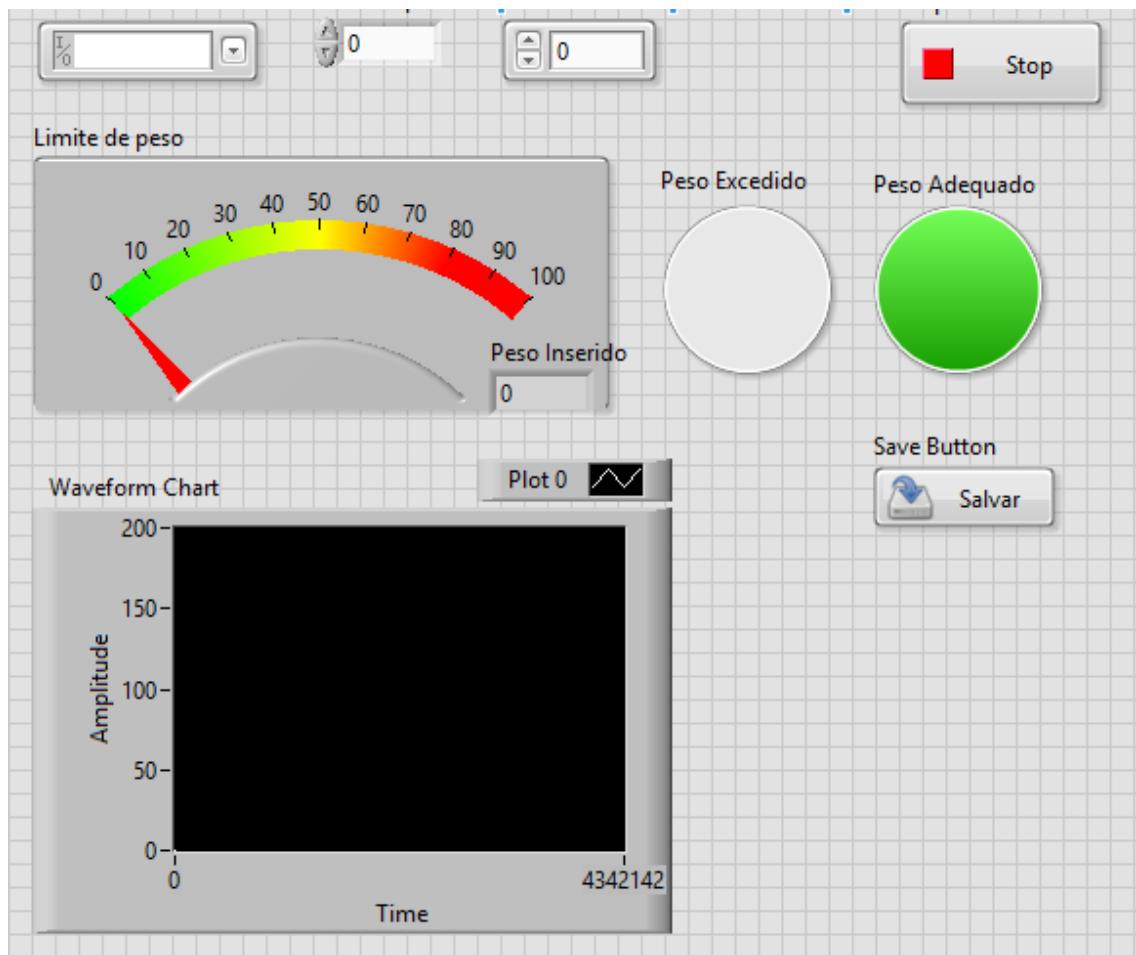


Figura 20: Tela de interação com o usuário projetada no Labview

O módulo de conexão USB do dispositivo com o computador exige firmwares e bibliotecas específicas na programação do PIC, além da criação de um driver junto ao assistente de criação de drivers da NI-VISA (NI-VISA Driver Wizard - Figura 22). Neste assistente, deve-se seguir os seguintes passos:

1. Seleciona USB quando pedir o “Hardware Bus”; clicar em “Next”;
2. Em “Device list” selecionar “Other...”; Clicar em “Next”;
3. Colocar os valores definidos de Vendor ID e Product ID nos respectivos campos e os nomes; clicar em “Next”;
4. Selecionar um prefixo para o arquivo e a pasta de destino; Clicar em “Next”
5. Instalar o .inf; se possível.

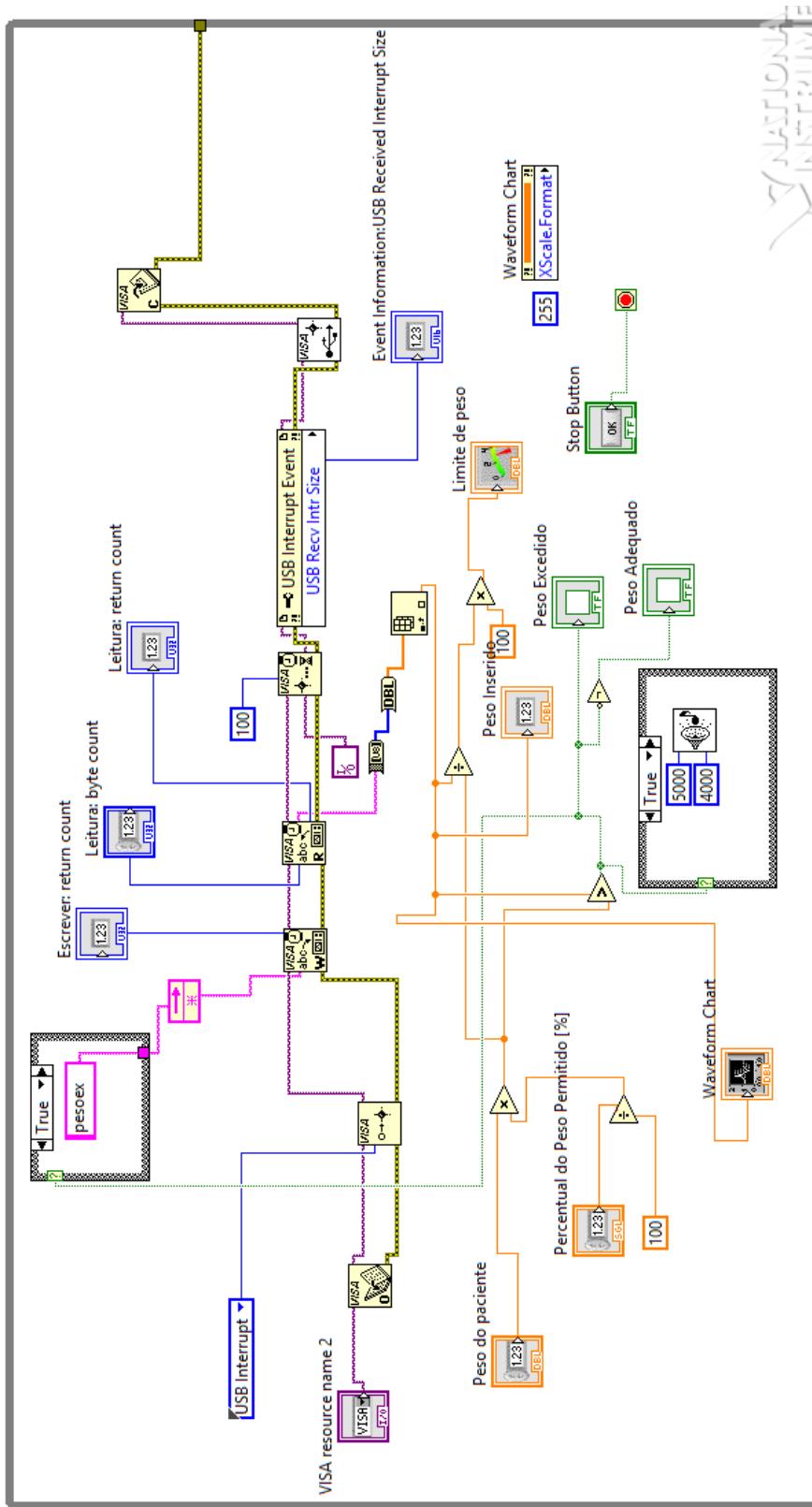


Figura 21: Diagrama de blocos representando o programa elaborado em Labview

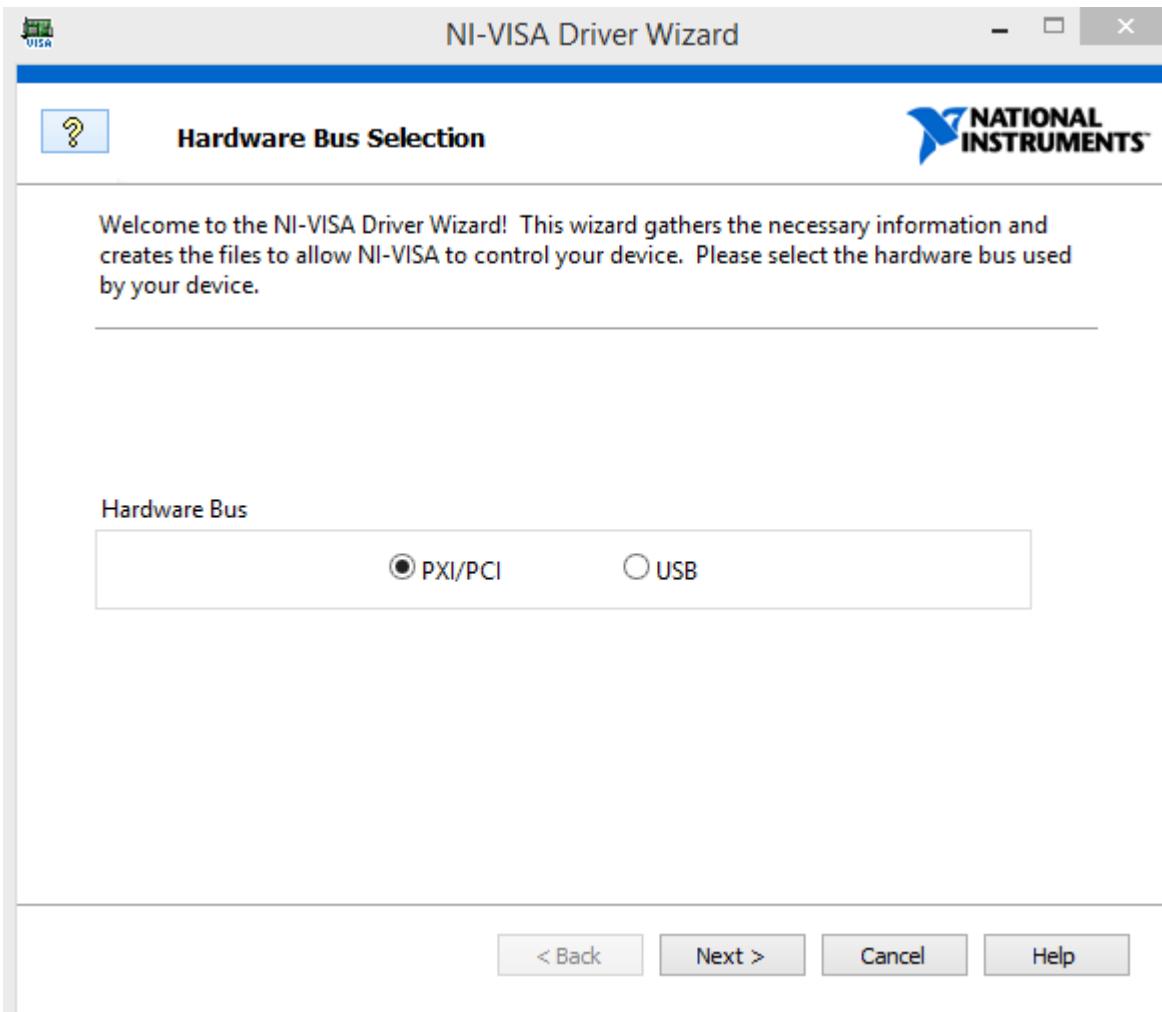


Figura 22: Tela inicial do assistente de criação de driver do NI-Visa

Para usuários do Windows 8 adiante, existe a exigência de uma assinatura digital para instalação do driver, dificultando seu uso por desenvolvedores e usuários sem intenção de uso lucrativo; pois estas assinaturas devem ser compradas. Existe, porém, uma forma de se contornar este detalhe utilizando o aplicativo Adobe Reader, que permite a criação de uma assinatura digital sem custos para colocação nos arquivos PDF. Pode-se exportar esta assinatura para permitir a instalação do driver.

O programa em C desenvolvido para o funcionamento do PIC foi adicionado no Apêndice. É possível ver a lógica de seu funcionamento no diagrama de blocos apresentado pelo Figura 23.

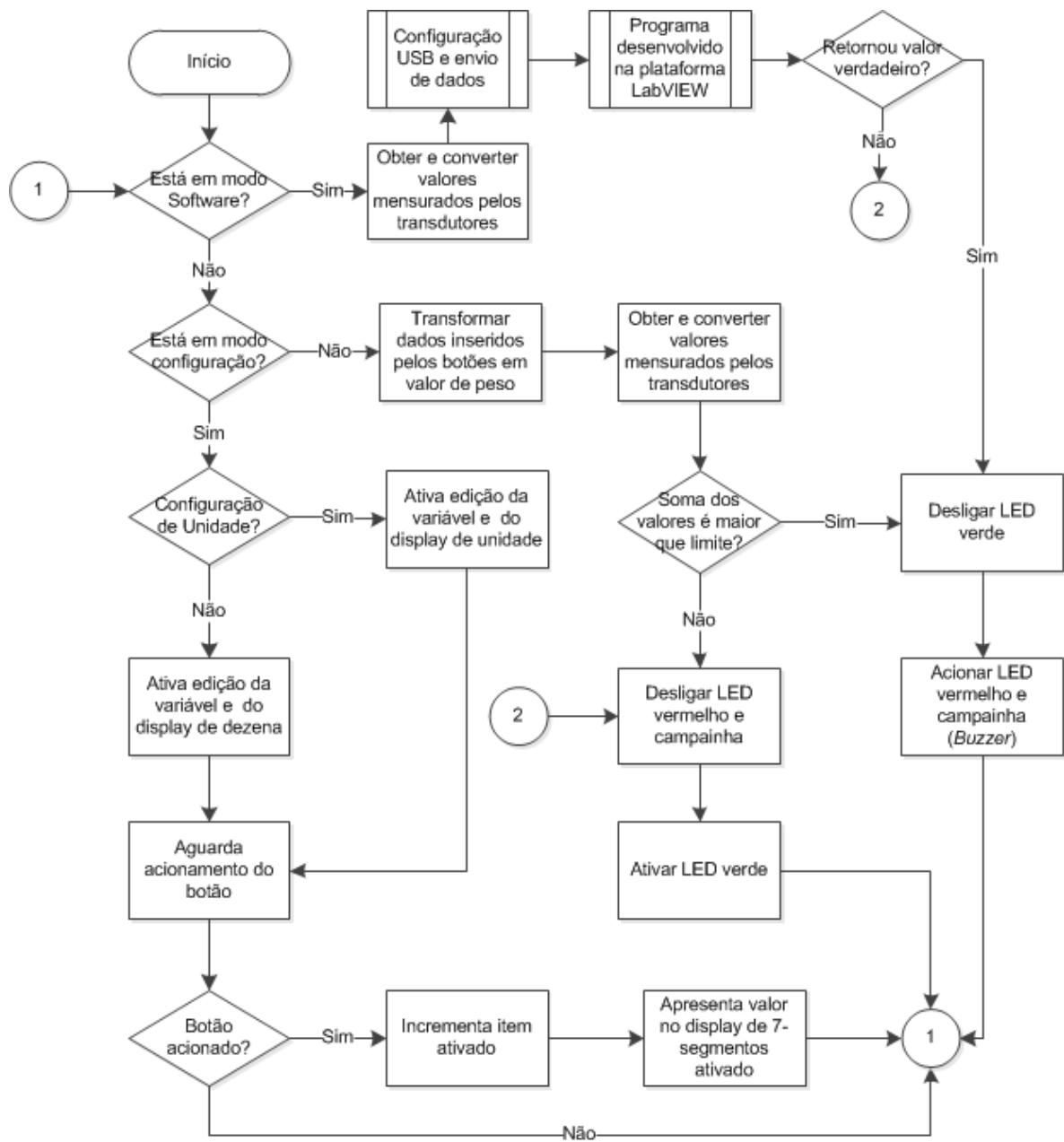


Figura 23: Diagrama de blocos da lógica de programação do PIC18F2550

Capítulo 4 – Resultados

Para o teste das células de carga foram utilizados um dinamômetro para medição de carga colocada no transdutor, além de osciloscópios. A colocação do REF01 permitiu que se tivesse uma tensão de alimentação estável; porém ainda não era possível observar um sinal na saída do circuito. A colocação do circuito INA129, configurada para a amplificação de 2000V/V, permitiu que osciloscópio apontasse o sinal do transdutor. Apesar disto, foi observar que os valores obtidos do transdutor não eram condizentes com o esperado teórico, não permitindo a obtenção de curva de comportamento nem uma possível calibração.

Com relação as simulações, o circuito montando junto ao PIC no aplicativo Proteus não apresentou problemas de simulação. Foi possível conectar o módulo USB ao computador com um circuito simples. Porém com a adição da programação completa do circuito, o módulo USB simulado deixou de conectar-se corretamente ao computador.

Para a conexão USB com o programa LabVIEW, devido a utilização do sistema operacional Windows 8, foi necessário a colocação de uma assinatura digital ao driver do módulo USB criado pelo NI-VISA Driver Wizard. Foi possível criar uma assinatura com o programa Adobe Reader e assinar o driver utilizando as linhas de comando (Figura 24) indicadas pela National Instruments.

```
C:\Users\AnaC>cd C:\Program Files (x86)\Windows Kits\8.1\bin\x86
C:\Program Files (x86)\Windows Kits\8.1\bin\x86>Inf2Cat /driver:C:\Users\AnaC\Documents\inffile /os:6_3_X86
-----
Signability test complete.

Errors:
None

Warnings:
None

Catalog generation complete.
C:\Users\AnaC\Documents\inffile\prefix2.cat

C:\Program Files (x86)\Windows Kits\8.1\bin\x86>cd C:\Program Files (x86)\Windows Kits\8.1\bin\x64
C:\Program Files (x86)\Windows Kits\8.1\bin\x86>signtool sign /f C:\Users\AnaC\OneDrive\certificadooo\AnaCarolina.pfx /p C:\Users\AnaC\Documents\inffile\prefix2.cat
Done Adding Additional Store
Successfully signed: C:\Users\AnaC\Documents\inffile\prefix2.cat

C:\Program Files (x86)\Windows Kits\8.1\bin\x86>mmc
C:\Program Files (x86)\Windows Kits\8.1\bin\x86>
```

Figura 24: Linhas de comando apresentando os passos necessários para inserção de assinatura

Já o programa de análise criado em LabVIEW, foram utilizados botões para colocação manual dos pesos para o teste da parte de cálculo do programa. É apresentado nas figuras seguintes o comportamento do programa conforme o peso inserido. A Figura 25, apresenta para um usuário de peso 100kg, e um limite de 30% do peso total, o funcionamento do programa quando este limite não foi ultrapassado; neste caso o peso colocado é de 27,55kg, menor que o limite de 30kg. O LED verde fica ligado, a campainha inserida não aciona. Já na Figura 26, é possível observar quando este valor ultrapassou o limite: o LED vermelho acende, enquanto o verde permanece apagado, a campainha colocada no código é acionada. Em ambas as figuras, é possível observar o gráfico que apresenta a variação no valor do peso inserido em função do tempo. Ambas as imagens foram retiradas em sequência, apresentando o mesmo histórico no gráfico.

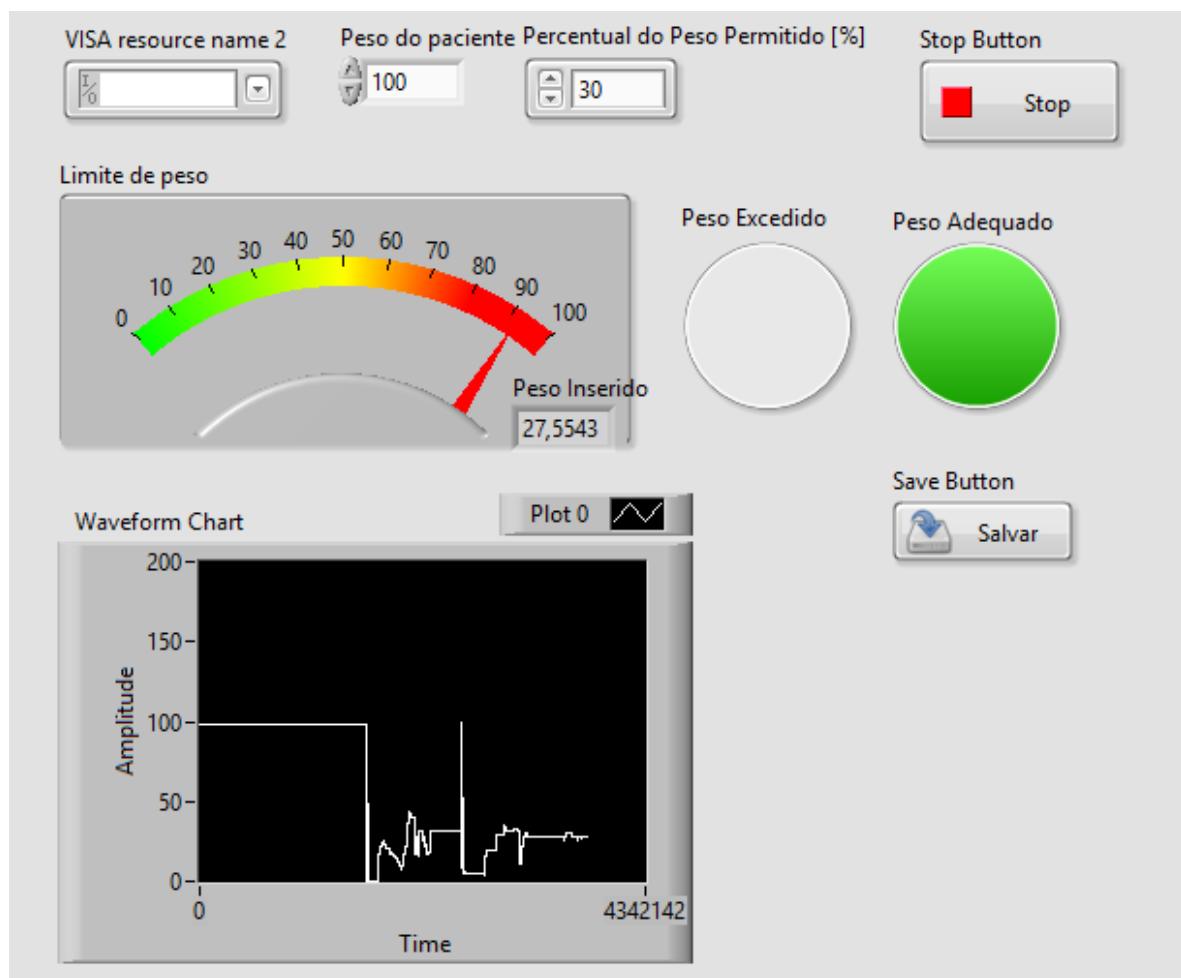


Figura 25: Tela do usuário (LabVIEW) quando o peso inserido está correto.

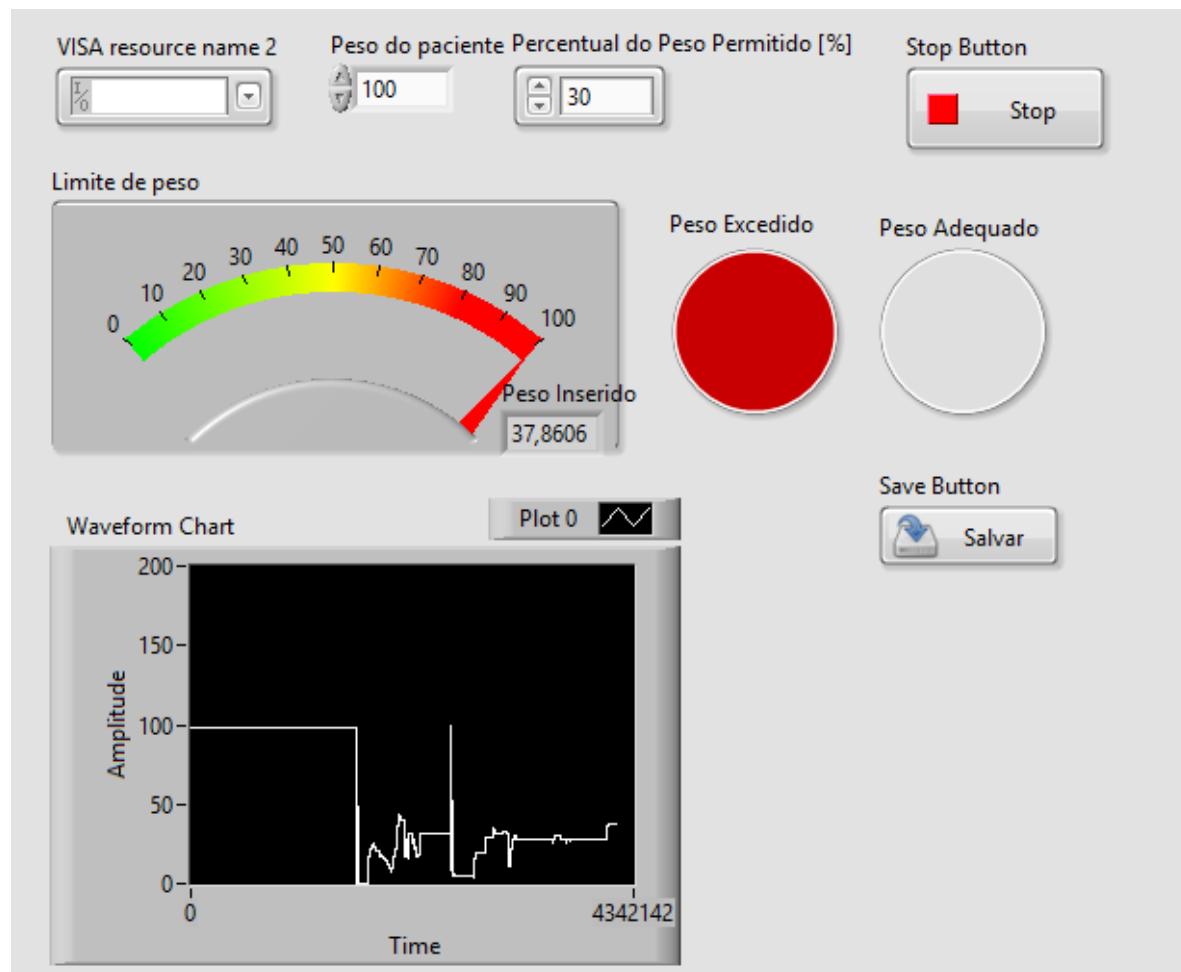


Figura 26: Tela do usuário (LabVIEW) quando o peso inserido está incorreto

Capítulo 5 – Conclusão

As simulações do projeto se mostraram favoráveis: Foi possível instalar o driver do dispositivo USB a partir de sua simulação com o software Proteus, e sua interação com o software LabVIEW. É possível simular os efeitos dos sinais de saída do circuito analógico (transdutores) colocando-se potenciômetros e variando-se seus valores dentro do software Proteus; apesar disto, a conexão entre os programas deixou de funcionar quando todas as funcionalidades necessárias foram adicionadas em ambos. Os testes em bancada com os sensores não foram bem-sucedidos até o momento.

Uma melhoria futura necessária ao projeto é colocação de extensômetros em funcionamento, o módulo PIC e sua conexão com os programas criados. Além disso, a interação com outros dispositivos, assim como a mensuração de centros de massa e gravidade, pode ser realizada com plataformas de forças; dispositivos deste tipo também podem ser usados para estudos e análises do “*gait*”, ou seja, a forma de andar, e postura de usuários. Alguns estudos reportaram que, com certas medidas em plataforma de forças, é possível prever o risco de quedas futuras em alguns pacientes (Condron & Hill, 2002). Ao prever estes riscos, pode-se agir de forma a amenizar estes riscos, o que pode implicar em, por exemplo, redução de fraturas e consequentemente a necessidade de treinamento de utilização de muletas.

Referências Bibliográficas

- ACROMAG INCORPORATED; (s.d.). *INTRODUCTION TO STRAIN & STRAIN MEASUREMENT*. Fonte: <https://www.acromag.com/sites/default/files/White%20Paper%20Introduction%20to%20Strain%20699A.pdf>
- Analog Devices. (2015). *Data Sheet REF01/REF02/REF03*.
- Andolfato, R. P., Camacho, J. S., & Brito, G. A. (2004). Extensometria Básica.
- Automatic Control, Robot and Mechatronics Labs.* (2007). Fonte: http://web.deu.edu.tr/mechatronics/TUR/strain_gauge.htm
- Buchanan, W. (2000). *Computer Busses*.
- Condron, J. E., & Hill, K. D. (2002). Reliability and validity of a Dual-Task Force Platform of Balance Performance: effect of age, balance impairment and cognitive task.
- Hershko, E., Tauber, C., & Carmeli, E. (2008). Biofeedback Versus Physiotherapy in. *Am J Orthop.*, pp. E92-E96.
- Hoffmann, K. (s.d.). *Applying the Wheatstone Bridge Circuit*.
- Hurkmans, H., Bussmann, J., Benda, E., Verhaar, J., & Stam, H. (2003). Techniques for measuring weight bearing during standing and walking.
- Hustedt, J. W., Blizzard, D. J., Baumgaertner, M. R., Leslie, M. P., & Grauer, J. N. (Novembro de 2012). Lower-extremity Weight-bearing Compliance Is Maintained Over Time After Biofeedback Training. *ORTHOPEDICS*.
- Ibrahim, D. (2012). *Advanced PIC Microcontroller Projects in C*. Newnes.
- Kaplan, Y. (2007). The use of a new biofeedback insole weight-bearing. *Journal of Sports Science and Medicine*.
- Microchip. (2009). *18F2550 Datasheet*.
- Miyazaki, S., & Iwakura, H. (1978). Limb-load alarm device for partial-weight-bearing walking exercise.
- National Instruments. (2009). *NI-VISA Overview*.
- P. T. Moran, G. F. (1995). A biofeedback cane system: instrumentation and subject application results," *Rehabilitation Engineering. IEEE Transactions on*, vol. 3, pp. 132-138.
- Peacock, C. (2010). Fonte: USB in a NutShell: <http://www.beyondlogic.org/>

- Roylance, D. (2001). STRESS-STRAIN CURVES.
- Santos, L. d. (2009). SISTEMA DE COMUNICAÇÃO USB COM MICROCONTROLADOR.
- Tagawa, T., Tamura, T., & Oberg, P. A. (2011). *Biomedical Sensors and Instruments*.
- Technologies, A. (1999). PRACTICAL STRAIN GAGE MEASUREMENTS.
- Texas Instruments. (2009). *INA129*.
- USB International Forum. (2002). Fonte: www.usb.org
- Varoto, R., Sato, A. M., Lins, C., & Cliquet, A. J. (2014). Can Simple Electronic Instrumentation Associated with Basic. *Proceedings of the International Conference on Biomedical Electronics and Devices*, pp. 259-264.
- VPG Transducers. (2014). Model 1042.
- VPG Transducers. (2015). Model 120 datasheet.
- Wikicommons. (s.d.). Fonte:
http://commons.wikimedia.org/wiki/File:Stress_Strain_Ductile_Material.pdf

Apêndice

Programa em linguagem C do PIC 18F2550

```
#include <18F2550.h>
#include <stdio.h>
#FUSES NOBROWNOUT      //Configuração fuses
#FUSES NOLVP          //Configuração fuses
#FUSES NOXINST         //Configuração fuses

#use delay(clock=48MHz,crystal=20MHz,USB_FULL) //clock para USB
#use FIXED_IO( B_outputs=PIN_B6,PIN_B5,PIN_B4,PIN_B3,PIN_B2,PIN_B1 ) //pinos de saída
#use FIXED_IO( C_outputs=PIN_C5,PIN_C2,PIN_C1 ) //pinos de saída
#define pushb  PIN_A4
#define dez_uni  PIN_B0
#define dez  PIN_B1
#define uni  PIN_B2
#define bcd_a  PIN_B3
#define bcd_b  PIN_B4
#define bcd_c  PIN_B5
#define bcd_d  PIN_B6
#define s_h  PIN_B7
#define gled  PIN_C1
#define rled  PIN_C2
#define buzz  PIN_C5
#define op_con PIN_C7
#define USB_CONFIG_BUS_POWER 500
#define USB_CONFIG_HID_TX_SIZE 63
#define USB_CONFIG_HID_RX_SIZE 63
#include <pic18_usb.h>
#include <usb_desc_hid.h>
#include <usb.c>
#define INT_USB
```

```

void buzzer(void);
int fun_conf(void);
int incrementa(void);
void segmento(void);
int leradc(void);
int ImprSeg (int valor);

int8 vsaida[63];
int8 ventrada[63];
int8 manual_ent[63];
int8 manual_sai[63];

void main()
{
    setup_adc_ports(AN0_TO_AN2); //configuração conversor AD inicial
    setup_adc(ADC_CLOCK_INTERNAL);
    //setup_adc(ADC_CLOCK_INTERNAL|ADC_TAD_MUL_0);
    enable_interrupts(INT_USB); //Habilita interrupção do USB
    enable_interrupts(GLOBAL);
    usb_init();

    if (S_H==1) //modo software, conexão com Labview. Controles via Labview
    {
        while(TRUE)
        {
            usb_init();
            usb_task();
            if (usb_enumerated())
            {

                vsaida[0]=leradc();
                usb_put_packet(1, vsaida, 63, USB_DTS_TOGGLE);

                if (usb_kbhit(1))

```

```

    {

        usb_get_packet(1, entrada, 1);
        if (entrada[0]==1)
        {
            output_high(rled); //LED vermelho e buzzer, peso incorreto!
            buzzer();
        }
        else
            output_high(gled); //LED verde, peso OK
        }
    }
    //TODO:codigo principal
}
} ELSE IF (S_H==0)//so PIC
{
    WHILE (TRUE)
    {
        if(op_con==1)//modo configuracao
        {
            int pes;
            int peso;
            do{
                pes=fun_conf();
            }while (op_con==1);
            } else if (op_con==0)
            {
                peso=adc();
                pes=pes*0.25;
                if (peso>=pes) //Peso ACIMA: liga LED vermelho. Ativa buzzer.
                {
                    buzzer();
                    output_high(rled);
                    buzzer();
                }
            }
        }
    }
}

```

```

        }

        else if(peso<pes) //peso OK: liga LED verde
        {
            output_high(gled);
        }
    }

}

}

}

}

func_conf() //função de configuração do peso
{
int pes;

if(pushb == 0) //Se o push-button é pressionado
{
    delay_ms(100); //delay para debounce
    if(pushb== 0) //verifica se foi realmente apertado o push button
    {
        pes=incrementa(); //função incrementa
    }
}

return pes;
}

void buzzer(void) //função para ativação beep do buzzer
{
    int8 i;
    for (i = 0 ; i < 170 ; i++)
    {
        output_high(buzz);
        delay_us(500);
        output_low(buzz);
        delay_us(400);
    }
}

```

```

int adc(void) // funcao de conversao analógico digital
{
    int8 peso;
    int8 aux4;
    int8 aux5;
    int8 aux6;
    set_adc_channel(PIN_A0);
    aux4=read_adc();
    set_adc_channel(PIN_A1);
    aux5=read_adc();
    set_adc_channel(PIN_A2);
    aux6=read_adc();
    delay_ms(400);
    peso=(aux4+aux5+aux6)*0.3;
    return peso;
}

int ImprSeg (int valor) //insere o valor de incremento do push button nos displays de 7seg
{
    int a=0b0000;
    int b=0b0000;
    int c=0b0000;
    int d=0b0000;
    int result;
    a=(0b0001&&valor);
    output_bit(bcd_a,a);
    b=(0b0010&&valor);
    output_bit(bcd_b,b);
    c=(0b0100&&valor);
    output_bit(bcd_c,c);
    d=(0b1000&&valor);
    output_bit(bcd_d,d);
    result=a&&b&&c&&d;
    return result;
}

```

```

}

int incrementa (void) //funcao para incrementar valores de peso, caso push button seja
pressionado
{
    int8 vdez;
    int8 vuni;
    int8 aux1;
    int8 aux2;
    int8 rest=0b00000000;
    int8 msb;
    int8 teste;
    teste=input(PIN_B0); //valor indicando se está modificando dezena ou centena - chave
conectada ao PIC.

    if(teste==1)          //testa se dezena
    {
        if(vdez<=0b1000)
        {
            ++vdez;
            output_low(PIN_B1);      //ativa o display dezena
            aux1=ImprSeg(vdez); //coloca dezena no display
            output_high(PIN_B1);     //desativa o display dezena
        }else if(vdez=0b1001)
        {
            vdez=0b0000;      //zera dezena
        }
    }else if (teste==0) //testa se unidade
    {
        if(vuni<=0b1000)      //incrementa se menor que 9
        {
            ++vuni;          //incrementa
            output_low(PIN_B2); //ativa o display unidade
            aux2=ImprSeg(vuni); //coloca unidade no display
            output_high(PIN_B2); //desativa o display unidade
        }else if(vuni=0b1001)
    }
}

```

```
{  
vuni=0b0000;      //zera unidade  
}  
}  
msb=vdez;  
swap(msb);  
rest=msb&&uni; //junta os valores  
return rest; //retorna valor inserido total  
}
```