

Sensor Inteligente para Medição de Cargas Mecânicas

Manuel António Rodrigues Vieira

Relatório Final da Dissertação/Trabalho de Projeto apresentado à:

**Escola Superior de Tecnologia e de Gestão
Instituto Politécnico de Bragança**

Para obtenção do grau de Mestre em:

Engenharia Industrial

Área de Especialização de Engenharia Eletrotécnica

Realizado sob a orientação de:

**Professor Doutor João Paulo Coelho
Professor Doutor João Castro Ribeiro**

Abril de 2016

“It does not matter how slowly you go as long as you do not stop”

(Confucius)

AGRADECIMENTOS

Para que o objetivo da presente tese tenha sido atingido, várias pessoas disponibilizaram o seu tempo, trabalho e dedicação, e apoiaram-me de forma incondicional. A elas, são dedicadas estas breves linhas.

Em primeiro lugar, quero expressar o meu enorme agradecimento ao Professor Doutor João Paulo Coelho e ao Professor Doutor João Castro Ribeiro, orientadores desta tese, pela disponibilidade constante, pelos conhecimentos que me transmitiram e acima de tudo, pelo incentivo.

Um agradecimento também ao técnico de laboratório Filipe Fernandes, pela ajuda e disponibilidade no fabrico das placas de circuito impresso e no fornecimento de material para este projeto.

Agradecimento especial à minha mãe, pelo seu apoio, compreensão e dedicação demonstrada ao longo deste trabalho.

Por fim, um magnânimo agradecimento ao meu pai (in memoriam), que sempre me motivou a dedicar empenho nos meus objetivos, e sem a ajuda dele, jamais poderia ser quem sou, e ter alcançado este grau académico.

A todos, muito obrigado.

RESUMO

A medição precisa da força é necessária para muitas aplicações, nomeadamente, para a determinação da resistência mecânica dos materiais, controlo de qualidade durante a produção, pesagem e segurança de pessoas. Dada a grande necessidade de medição de forças, têm-se desenvolvido, ao longo do tempo, várias técnicas e instrumentos para esse fim. Entre os vários instrumentos utilizados, destacam-se os sensores de força, também designadas por células de carga, pela sua simplicidade, precisão e versatilidade. O exemplo mais comum é baseado em extensómetros eléctricos do tipo resistivo, que aliados a uma estrutura formam uma célula de carga. Este tipo de sensores possui sensibilidades baixas e em repouso, presença de *offset* diferente de zero, o que torna complexo o seu condicionamento de sinal.

Este trabalho apresenta uma solução para o condicionamento e aquisição de dados para células de carga que, tanto quanto foi investigado, é inovador. Este dispositivo permite efetuar o condicionamento de sinal, digitalização e comunicação numa estrutura atómica. A ideia vai de encontro ao paradigma dos sensores inteligentes onde um único dispositivo eletrónico, associado a uma célula de carga, executa um conjunto de operações de processamento de sinal e transmissão de dados. Em particular permite a criação de uma rede *ad-hoc* utilizando o protocolo de comunicação IIC.

O sistema é destinado a ser introduzido numa plataforma de carga, desenvolvida na Escola Superior de Tecnologia e Gestão de Bragança, local destinado à sua implementação. Devido à sua estratégia de conceção para a leitura de forças em três eixos, contém quatro células de carga, com duas saídas cada, totalizando oito saídas. O *hardware* para condicionamento de sinal já existente é analógico, e necessita de uma placa de dimensões consideráveis por cada saída. Do ponto de vista funcional, apresenta vários problemas, nomeadamente o ajuste de ganho e *offset* ser feito manualmente, tornando-se essencial um circuito com melhor desempenho no que respeita a lidar com um *array* de sensores deste tipo.

Palavras-chave: Sensores inteligentes, motes, extensómetros, comunicação IIC, plataforma de forças.

ABSTRACT

The precise measurement of force is necessary for many applications, namely to determine material resistance, quality control during productions, weighing and personnel safety. Given the high necessity of force measurement, for many years, techniques and instruments have been developed to aid this quest. Among the many used instruments, force sensors, also known as load cells, have gained interest due to their simplicity, precision and versatility. The most common example is based on electrical strain gauges of resistive type, that joined to a structure, compose a load cell. This type of sensors has low sensibility, and in rest display offsets, which are different from zero, and makes signal conditioning quite complex.

This work presents a solution for conditioning and data acquisition for load cells that, as much as been investigated, is innovative. The developed device allows signal conditioning, digitalization and communication in an atomic structure. The idea meets the paradigm of smart sensors, where a single electronic device, coupled to a load cell, executes a set of signal processing and data transmission. Particularly, it allows a creation of an ad-hoc network, using the IIC protocol. The system is to be introduced in a loading platform, developed at the School of Technology and Management of Bragança, where it will be implemented.

Given that it is conceived to read the force in three axes, it has four load cells with two outputs each, totalizing eight outputs. The existing hardware for signal conditioning is analogue and needs a considerable sized board for each output. It displays some flaws in terms of functionality, namely the gain adjustment, and the fact that the offset is manual, being essential to develop a board with better performance in terms of dealing with an array of sensors of this type.

Keywords: Smart Sensors, motes, strain gauge, IIC bus, force platform.

ÍNDICE

LISTA DE ACRÓNIMOS	VIII
LISTA DE FIGURAS.....	IX
LISTA DE TABELAS.....	XII
1. INTRODUÇÃO:.....	1
1.1 DESCRIÇÃO DO PROBLEMA	1
1.2 REVISÃO DO ESTADO DA ARTE.....	2
1.3 OBJETIVOS DO TRABALHO	5
1.4 ESTRUTURA DO DOCUMENTO.....	6
2. MATERIAIS E MÉTODOS:.....	8
2.1 SENSORES PARA MEDIÇÃO DE FORÇA	8
2.2 CÉLULA DE CARGA E EXTENSÓMETROS	9
2.3 CONDICIONAMENTO DE SINAL.....	10
2.3.1 <i>Ponte de Wheatstone</i>	10
2.3.2 <i>Amplificação e Filtragem</i>	16
2.3.3 <i>Ajuste de zero</i>	17
2.4 PLATAFORMA DE FORÇAS	18
2.4.1 <i>Estrutura Mecânica</i>	18
2.4.2 <i>Geometria das Células de carga</i>	19
2.4.3 <i>Solução Atual de Condicionamento de Sinal</i>	20
3. SISTEMA INTEGRADO PARA MEDIÇÃO DE FORÇAS:	23
3.1 ARQUITETURA DO SENSOR INTELIGENTE.....	23
3.1.1 <i>Equilíbrio da ponte</i>	23
3.1.2 <i>Amplificador programável</i>	27
3.1.3 <i>Comunicação Master/Slave</i>	35
3.2 SOFTWARE DE CONTROLO E REGISTO	39
3.3 MONTAGEM DE UMA CÉLULA COM DOIS MOTES	49
3.4 ANÁLISE DE RESULTADOS.....	54
4. CONCLUSÃO E SUGESTÕES PARA TRABALHO FUTURO	66

REFERÊNCIAS	69
ANEXO 1.....	A
ANEXO 2.....	B
ANEXO 3.....	C

LISTA DE ACRÓNIMOS

ADC – Analog/Digital Converter
AmpOp - Amplificador Operacional
CAD - Computer-Aided Design
CAE - Computer-aided engineering
DC – Direct Current
EMI - Electromagnetic Interference
ESTiG – Escola Superior de Tecnologia e Gestão de Bragança
IIC - Inter-Integrated Circuit
ISP - In-System Programmer
LIFA – LabVIEW Interface for Arduino
LSB – Less Significant Byte
MSB – More Significant Byte
PCB – Printed Circuit Board
PWM - Pulse-Width Modulation
USB – Universal Serial Bus

LISTA DE FIGURAS

FIGURA 1-1. DIGITAL LOAD CELL CONVERTER. DIREITOS RESERVADOS À MANTRACOURT.	2
FIGURA 1-2. DIMENSÕES DO DSC. DIREITOS RESERVADOS À MANTRACOURT.	3
FIGURA 1-3. DCELL. DIREITOS RESERVADOS À MANTRACOURT.	4
FIGURA 1-4. DIAGRAMA DO DCELL. DIREITOS RESERVADOS À MANTRACOURT.	4
FIGURA 1-5. ESQUEMA GERAL DA REDE IIC.	6
FIGURA 2-1. EXTENSÓMETRO (ILUSTRAÇÃO).	9
FIGURA 2-2. CÉLULA DE CARGA RETANGULAR DE PONTE DE COMPLETA (ILUSTRAÇÃO).	10
FIGURA 2-3. PONTE DE WHEATSTONE. DIREITOS RESERVADOS À NATIONAL INSTRUMENTS.	11
FIGURA 2-4. PONTE DE WHEATSTONE (QUARTO).	13
FIGURA 2-5. PONTE DE WHEATSTONE (MEIA).	14
FIGURA 2-6. PONTE DE WHEATSTONE (COMPLETA).	15
FIGURA 2-7. AMPLIFICADOR DE INSTRUMENTAÇÃO. DIREITOS RESERVADOS À TEXAS INSTRUMENTS.	16
FIGURA 2-8. PONTE EM EQUILÍBRIO.	17
FIGURA 2-9. PLATAFORMA DE FORÇAS [21].	18
FIGURA 2-10. AVALIAÇÃO DA DEFORMAÇÃO [21].	19
FIGURA 2-11. DIAGRAMA DA CÉLULA DE CARGA [21].	19
FIGURA 3-1. CIRCUITO DE AJUSTE DE ZERO CONTROLADO POR PWM.	23
FIGURA 3-2. PINOS DA GAMA ATTINY. DIREITOS RESERVADOS À ATMEL CORPORATION.	24
FIGURA 3-3. PROGRAMADOR DO ATTINY85 COM ARDUINO ISP.	25
FIGURA 3-4. CÉLULA DE CARGA CILÍNDRICA.	26
FIGURA 3-5. AD8556. DIREITOS RESERVADOS À ANALOG DEVICES.	27
FIGURA 3-6. DIAGRAMA FUNCIONAL DO AD8556. DIREITOS RESERVADOS À ANALOG DEVICES.	28
FIGURA 3-7. DETECÇÃO DE ERROS. DIREITOS RESERVADOS À ANALOG DEVICES.	30

FIGURA 3-8. DIAGRAMA DE UM SINAL DIGITAL. DIREITOS RESERVADOS À ANALOG DEVICES.	31
FIGURA 3-9. DIAGRAMA DO CIRCUITO DESENVOLVIDO.	35
FIGURA 3-10. ESQUEMA DE COMUNICAÇÃO IIC.	35
FIGURA 3-11. VISTA DO ARDUINO UNO. DIREITOS RESERVADOS À ROBOMART.	36
FIGURA 3-12. PAINEL FRONTAL DO PROGRAMA.	39
FIGURA 3-13. BLOCOS DA LIFA PARA COMUNICAÇÃO COM ARDUINO.	41
FIGURA 3-14. BLOCOS DA LIFA PARA COMUNICAÇÃO IIC.	41
FIGURA 3-15. DIAGRAMA DE BLOCOS PARA ENVIO DE ORDEM.	42
FIGURA 3-16. DIAGRAMA DE BLOCOS DO CASE 0.	43
FIGURA 3-17. MENSAGEM DE ERRO DE COMUNICAÇÃO.	43
FIGURA 3-18. DIAGRAMA DE BLOCOS DO CASE 1.	44
FIGURA 3-19. DIAGRAMA DE BLOCOS DO CASE 2.	45
FIGURA 3-20. MENSAGEM DE AVISO DE EQUILÍBRIO.	45
FIGURA 3-21. MENSAGEM DE ERRO NA PONTE.	46
FIGURA 3-22. DIAGRAMA DE BLOCOS DO CASE 3.	46
FIGURA 3-23. DIAGRAMA DE BLOCOS PARA APRESENTAÇÃO DE VALORES E INTERPOLAÇÃO LINEAR.	47
FIGURA 3-24. FLUXOGRAMA DO SOFTWARE SLAVE.	48
FIGURA 3-25. ELEMENTO METÁLICO DA CÉLULA DE CARGA.	49
FIGURA 3-26. MARCAS DE REFERÊNCIA NA CÉLULA.	50
FIGURA 3-27. VISTA DA CÉLULA APÓS COLAGEM DOS EXTENSÓMETROS.	51
FIGURA 3-28. LAYOUT DA NOVA PLACA.	51
FIGURA 3-29. PLACAS DIGITAIS DE CONDICIONAMENTO DE SINAL.	52
FIGURA 3-30. COMPARAÇÃO ENTRE OS TAMANHOS DA CÉLULA E UMA DAS PLACAS.	53
FIGURA 3-31. UMA PLACA PRONTA A SER MONTADA NA CÉLULA.	53
FIGURA 3-32. HARDWARE MONTADO NA CÉLULA, FORMANDO O SMART SENSOR.	54
FIGURA 3-33. LIGAÇÃO DO HARDWARE FINAL.	54
FIGURA 3-34. EQUILÍBRIO DA PONTE COM GANHO DE X900.	55
FIGURA 3-35. CALIBRAÇÃO DA CÉLULA.	56
FIGURA 3-36. PESOS USADOS NA CALIBRAÇÃO DA CÉLULA.	57
FIGURA 3-37. PAINEL FRONTAL. TENSÃO MEDIDA PARA 24,615 N.	57

FIGURA 3-38. PAINEL FRONTAL. TENSÃO MEDIDA PARA 74,04 N.....	58
FIGURA 3-39. PAINEL FRONTAL. TENSÃO MEDIDA PARA 143,765 N.....	58
FIGURA 3-40. PAINEL FRONTAL. TENSÃO MEDIDA PARA 221,042 N.....	59
FIGURA 3-41. PAINEL FRONTAL. TENSÃO MEDIDA PARA 296,553 N.....	59
FIGURA 3-42. CURVA DE CALIBRAÇÃO.....	61
FIGURA 3-43. PAINEL FRONTAL. VALOR INTERPOLADO PARA 25,5 N.....	62
FIGURA 3-44. PAINEL FRONTAL. VALOR INTERPOLADO PARA 47,6N.....	63
FIGURA 3-45. PAINEL FRONTAL. VALOR INTERPOLADO PARA 99N.....	63
FIGURA 3-46. PAINEL FRONTAL. VALOR INTERPOLADO PARA 203,6N.....	64
FIGURA 3-47. PAINEL FRONTAL. VALOR INTERPOLADO PARA 302,5N.....	64

LISTA DE TABELAS

TABELA 1. TENSÕES DE DETEÇÃO DE ERROS. DIREITOS RESERVADOS À ANALOG DEVICES.	30
TABELA 2. CORRENTES DE DETEÇÃO DE ERROS. DIREITOS RESERVADOS À ANALOG DEVICES.	30
TABELA 3. DURAÇÃO DE PULSOS.....	31
TABELA 4. FORMATO DA PALAVRA DE 38 <i>BIT</i> . DIREITOS RESERVADOS À ANALOG DEVICES.	32
TABELA 5. VALORES DE CALIBRAÇÃO.	60

1. INTRODUÇÃO:

1.1 DESCRIÇÃO DO PROBLEMA

Extensómetros eléctricos do tipo resistivo são elementos que quando deformados sofrem uma variação na resistência proporcional à deformação do material onde estão agregados. São colocados em configuração de ponte de *wheatstone* numa estrutura que se deseja medir deformação, denominada de célula de carga. Diversos fatores tornam o condicionamento de sinal das células de carga uma tarefa complicada, devido à sensibilidade da saída ser relativamente baixa, a amplificação é o passo que estará em primeiro lugar para converter a variação de resistência numa tensão de saída, na ordem de mV ou V, podendo ser medida por voltímetros ou processada por micro controladores. O sinal amplificado é relacionado com a deformação do material devido à força aplicada.

Outros processos são fundamentais, como o controlo de *offset*, devido à saída apresentar valores inerentes ao desequilíbrio da célula, e também a filtragem, pois a relação sinal/ruído baixa neste tipo de sistemas. Estas dificuldades são ampliadas quando o número de células de carga aumenta e há a necessidade da medição simultânea de todas elas, como acontece na plataforma de forças a que se destina este projeto.

O circuito já existente apresenta diversos problemas. O equilíbrio da ponte de extensómetros (ajuste de zero) e ajuste de *offset* são efetuados manualmente. Devido ao facto da saída ser analógica, é preciso mais um componente no sistema, como uma placa de aquisição dados que adquira o sinal de cada saída e o converta para digital, para que os dados possam ser processados, e permita uma conversão de unidades de tensão para unidades de força/carga.

As dimensões do circuito e cablagem associada do antigo sistema também constituem um ponto a ser melhorado, visto não serem as mais apelativas.

1.2 REVISÃO DO ESTADO DA ARTE

Neste subcapítulo será feita uma revisão do estado da arte no que concerne aos equipamentos de condicionamento de células de carga existentes no mercado.

As células de carga são instrumentos indispensáveis no dia-a-dia, pois permitem medir várias grandezas, nomeadamente, pressões, forças, deformações, pesos, binários, entre outras. São utilizadas em várias indústrias; agricultura, engenharia civil, indústria alimentar, farmacêutica, medicina, automóvel, entre outras [1, 2].

Após pesquisa *online* em locais relevantes à temática, vários sistemas para células de carga foram encontrados, tanto como produtos existentes no mercado, patentes de células de carga e publicações científicas detalhando investigação relacionada com estas células. Em relação aos produtos existentes no mercado, uma empresa líder na comercialização de células de carga é a Mantracourt, tendo disponíveis duas soluções; o *Digital Load Cell Converter* (DSC) e o *DCell* [3]. Ainda que a finalidade de ambos seja para condicionar o sinal de *arrays* de células de carga, a sua forma de comunicação varia entre RS485, ASCII, *Modbus* ou CAN, não permitindo portanto comunicação através de IIC.



Figura 1-1. Digital Load Cell Converter. Direitos reservados à Mantracourt.

O DSC (Figura 1-1) é uma placa digital produzida pela Mantracourt, que permite o condicionamento de sinal para células de carga. Está disponível para os barramentos RS232, RS485 e CAN e possui as seguintes características:

- Resolução de 18 bit
- 500 Atualizações por minuto
- Autocalibração ou linearização
- Filtro dinâmico
- Compensação de temperatura
- Detecção de erros
- *Software* que permite 24 dispositivos ativos simultaneamente
- Calibração de shunt automática (ajuste de zero)

As dimensões estão patentes na Figura 1-2.

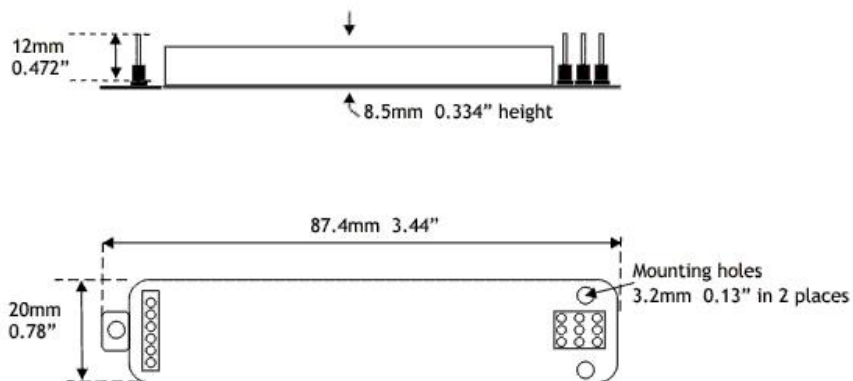


Figura 1-2. Dimensões do DSC. Direitos reservados à Mantracourt.

No caso de ser necessário montar a placa de condicionamento de sinal dentro da célula de carga de formato cilíndrico, a empresa fornece a DCell, que apresenta um diâmetro e espessura com cerca de 19 mm de diâmetro (Figura 1-3).

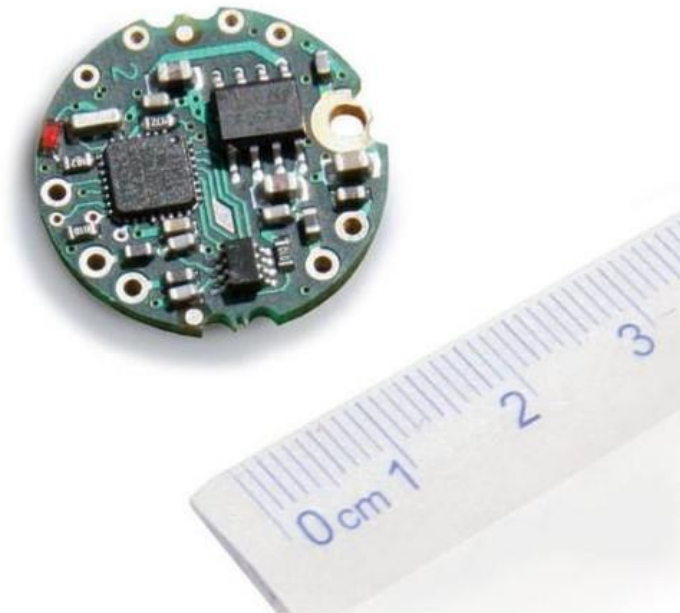


Figura 1-3. DCell. Direitos reservados à Mantracourt.

Na Figura 1-4, encontra-se um diagrama das dimensões do DCell. Este apresenta um formato circular compacto e diferenças físicas consoante o barramento, RS485 ou CAN.

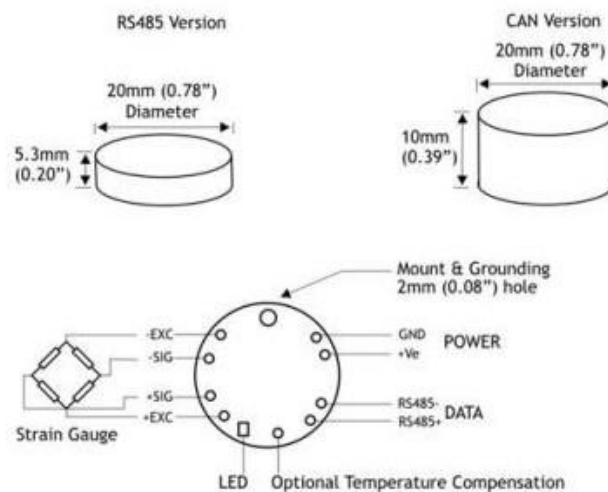


Figura 1-4. Diagrama do DCell. Direitos reservados à Mantracourt.

No que concerne a patentes ligadas a células de carga foram encontradas várias entradas, sendo que algumas apenas lidam com o desenho da célula de carga e outras incluem vários tipos de circuitos de condicionamento de sinal [4, 5, 6, 7, 8].

Finalmente, a nível de publicações científicas, foi demonstrada a utilização de uma célula de carga com comunicação por banda 2.4 GHz [9]. Em 1999, demonstraram a utilização de uma célula de carga no apoio à investigação de articulações músculo-esqueléticas humanas [10]. Em 2009 propuseram um sistema parecido ao proposto nesta dissertação [2], contudo a forma de comunicação entre a célula de carga e o dispositivo mestre dependia do barramento RS-232, além de outras limitações que serão ultrapassadas neste trabalho, nomeadamente por não permitir o controlo de ganho e *offset* automaticamente.

1.3 OBJETIVOS DO TRABALHO

Com os desafios apresentados na descrição do problema, pretende-se desenvolver um circuito compacto, que agregado a cada célula de carga, a converta num *smart sensor*. Dado que a plataforma de forças em uso permite medir forças em 3 eixos, é necessário recorrer a quatro células de carga, com duas pontes completas de extensómetros cada. Assim, este projeto consiste na implementação de um sistema digital por IIC do tipo mestre/escravos para substituir o antigo sistema analógico e, desta forma reduzir o tamanho da cablagem e do *hardware*, tendo um único dispositivo de controlo/leitura, para facilitar a adição/remoção de células de carga no sistema. A Figura 1-5 ilustra este conceito.

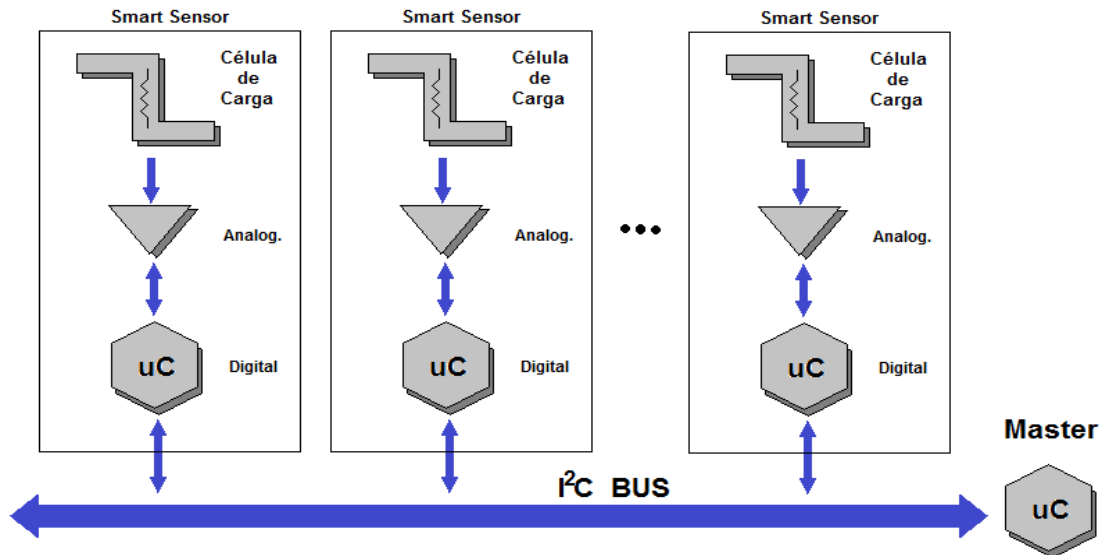


Figura 1-5. Esquema geral da rede IIC.

O novo sistema contará com uma nova placa eletrônica por cada ponte, facilmente adaptável na própria célula de carga, onde estarão presentes os componentes necessários para o condicionamento de sinal destes sensores inteligentes. Além disso, o sistema possuirá capacidade de variação do ganho e *offset* remotamente, e um processador local devidamente programado para fazer conversão analógica/digital, garantindo a comunicação de cada sensor escravo com um dispositivo mestre através do barramento IIC.

1.4 ESTRUTURA DO DOCUMENTO

Este documento, desenvolvido no âmbito da unidade curricular de Dissertação, do curso de Engenharia Industrial, para obtenção do grau de mestre no ramo de Engenharia Eletrotécnica, encontra-se dividido em 4 capítulos, e pretende descrever todo o trabalho desenvolvido neste projeto.

A introdução ao tema é realizada no capítulo 1. Sendo aqui efetuada a caracterização detalhada do problema a tratar e são explicitados os objetivos propostos para esta dissertação, assim como a revisão do estado da arte, onde se apresentam as soluções disponíveis no mercado.

Materiais e métodos é o conteúdo que preenche o capítulo 2 deste documento, onde é apresentada a teoria que está por trás das células de carga baseadas em extensômetros do tipo resistivo, e a plataforma de forças a que se destina este projeto.

O capítulo 3, Sistema Integrado para Medição de Forças, é dedicado à concepção do *hardware* e *software* desenvolvido no âmbito deste projeto é onde são efetuados testes ao novo sistema, bem como, a apresentação de resultados em função dos objetivos iniciais.

Finalmente, no capítulo 4, serão apresentadas as conclusões e propostas de melhorias a implementar futuramente.

2. MATERIAIS E MÉTODOS:

2.1 SENSORES PARA MEDIÇÃO DE FORÇA

O cientista inglês Robert Hooke, em 1678, desenvolveu um sistema de medição de forças, mostrando que a distância que uma mola se estendia era proporcional à quantidade de força nela aplicada. A sua teoria tornou-se conhecida como Lei de Hooke [11], e ao longo do tempo, as molas foram frequentemente utilizadas para medição de forças.

A força que atua sobre um objeto, provoca a sua deformação. Esta deformação pode ser medida de forma mecânica, ótica, acústica, pneumática ou elétrica. Esta última forma tem sido bastante diversificada através do uso de células de carga, que não são nada mais que transdutores de força para uma grandeza elétrica.

Um exemplo bastante comum de um sistema de medição de força eletrônico é uma balança digital, (industrial ou doméstica) que mede a quantidade de força exercida sobre ela a partir de uma célula de carga, mostrando a intensidade da força sob a forma de unidades de peso.

Uma célula de carga é um sensor que, a partir da deformação, converte um leque variado de informação como força, tensão mecânica e pressão, que são exercidos sobre ela num sinal elétrico [12]. Este sinal pode ser uma variação de tensão elétrica, corrente ou frequência dependendo do tipo de célula e circuito.

As células de carga piezoelétricas possuem uma propriedade denominada de piezoelectricidade, onde os elementos e os materiais que as constituem são capazes de gerar tensão elétrica por resposta a uma pressão mecânica [13]. Este tipo de plataforma tem a vantagem de ter uma velocidade de resposta às perturbações ocorridas muito favorável, por isso é utilizada mais frequentemente em avaliações de tarefas dinâmicas [14].

Devido ao elevado custo das células de carga com piezoelétricos, que são elementos caros e com um grau de eficiência elevado, é usual optar-se por uma alternativa onde são utilizados como sensores de medida os extensómetros elétricos do tipo resistivo. Estes extensómetros quando associados a uma estrutura deformável de

acordo com a lei de Hooke, formam células de carga utilizadas para medir grandezas físicas como força ou pressão.

2.2 CÉLULA DE CARGA E EXTENSÓMETROS

Os extensômetros são habitualmente finos e de forma retangular, consistindo numa serpentina de material (fio) condutor, composta por um fio metálico ou várias tiras metálicas que é depositada numa folha de material não condutor. Quando a serpentina é esticada (tração), fica mais fina e ocorre um aumento da resistência elétrica, já quando é comprimida (compressão), torna-se mais espessa e dá-se uma diminuição da resistência elétrica [15]. A sensibilidade dos extensômetros usados na célula deste trabalho é limitada a apenas um eixo, este tipo é ilustrado na Figura 2-1.

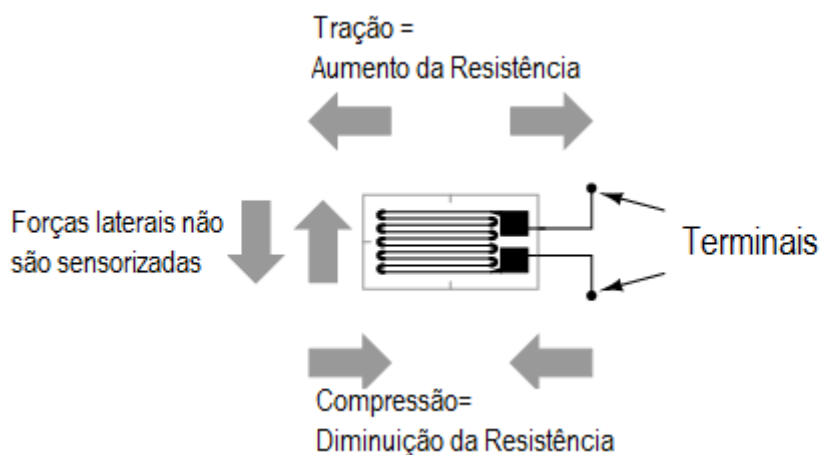


Figura 2-1. Extensômetro (ilustração).

Uma célula de carga do tipo extensométrica é composta por uma estrutura metálica devidamente estudada para a aplicação em causa, que vai sofrer deformação quando uma força é aplicada na mesma. Os elementos sensíveis são denominados extensômetros resistivos e que, estão colados em sítios destinados estrategicamente para o efeito, a fim de acompanhar a deformação física da estrutura.

Os extensômetros são ligados habitualmente em ponte de *Wheatstone*, com as configurações de um quarto de ponte, meia ponte, ou ponte completa [15]. Esta última configuração está patente na Figura 2-2.

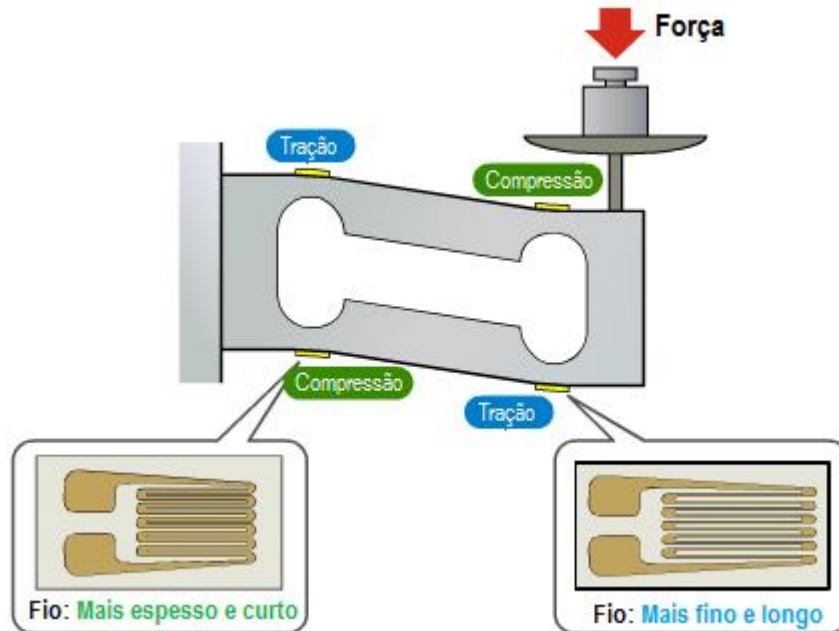


Figura 2-2. Célula de carga retangular de ponte de completa (ilustração).

2.3 CONDICIONAMENTO DE SINAL

2.3.1 PONTE DE WHEATSTONE

A montagem dos extensômetros é baseada na teoria da ponte de *Wheatstone*. As pontes de *Wheatstone* foram inventadas por Samuel Hunter Christie em 1833 e desenvolvidas por Charles Wheatstone em 1843 com o propósito de analisar solos e são basicamente circuitos formados por braços de resistências. A sua principal vantagem é fornecer medições de alta precisão [16]. A ponte é composta sempre por 4 resistências, mas os extensômetros podem ser utilizados nas três configurações distintas: ponte completa, meia ponte e um quarto de ponte, que relacionam-se com o número de

extensômetros utilizados, 4, 2 ou 1, respectivamente, e podem ser vistos como resistências ativas (variáveis) [15].

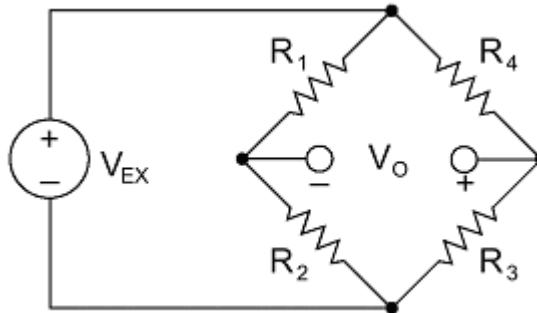


Figura 2-3. Ponte de Wheatstone. Direitos reservados à National Instruments.

A relação de proporcionalidade entre a variação da resistência do extensômetro e a tração é dada pelo fator de ganho do extensômetro – GF [15, 17].

$$\frac{dR}{R} = GF \times \frac{dL}{L},$$

Em que R é a resistência do condutor elétrico de tamanho L e dL corresponde à variação de comprimento provocada pela aplicação da força e conseqüente deformação da estrutura metálica, resultando na variação de resistência dR. A expressão anterior pode ainda tomar o seguinte aspecto:

$$\Delta R = GF \times \varepsilon,$$

Onde a variação da resistência está representada por ΔR e ε representa a extensão ou deformação, em notação inglesa é designada por *Strain*. Verifica-se então, através desta expressão, que é possível obter o valor da deformação, a partir da variação de resistência do extensômetro.

As pontes são alimentadas com uma fonte de tensão de excitação, e as pequenas variações de resistência que ocorrem, implicam uma tensão de saída proporcional, que pode ser de apenas alguns mV [15]. A Ponte de Wheatstone geral, ilustrada na Figura 2-

3, consiste em quatro braços resistivos com uma tensão de excitação, V_{EX} , que é aplicada através da ponte.

Pela lei de Kirchhoff e lei de Ohm tem-se que a tensão de saída da ponte, V_O , é igual a [15, 17]:

$$V_o = V_{R1} - V_{R4} = \left[\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right] \times V_{EX}$$

Desta equação, é aparente que quando $R_1/R_2 = R_4/R_3$, a tensão de saída V_o é zero e nestas condições, dizemos que a ponte está equilibrada. Isto é, não é necessário que $R_1 = R_4$ e $R_2 = R_3$ para atingir o equilíbrio, apenas que as razões das resistências de cada braço sejam iguais. Qualquer mudança de resistência em qualquer braço da ponte resulta numa tensão de saída diferente de zero [15].

Para o caso da ponte se encontrar desequilibrada, atendendo à Figura 2-3, existe a possibilidade da saída V_o se encontrar negativa, o que resulta de uma diminuição da resistência R_1 ou R_3 ou aumento da resistência R_2 ou R_4 . Por outro lado, a saída pode ser positiva, se houver uma diminuição da resistência R_4 ou R_2 ou aumento da R_1 ou R_3 . Claro que, o desequilíbrio pode ser causado por uma das situações ou uma combinação de várias.

Quarto de Ponte:

Quando um extensómetro é colocado na ponte, temos uma configuração em quarto de ponte (Figura 2-4), ou seja, uma das quatro resistências é ativa e podemos então medir uma força aplicada.

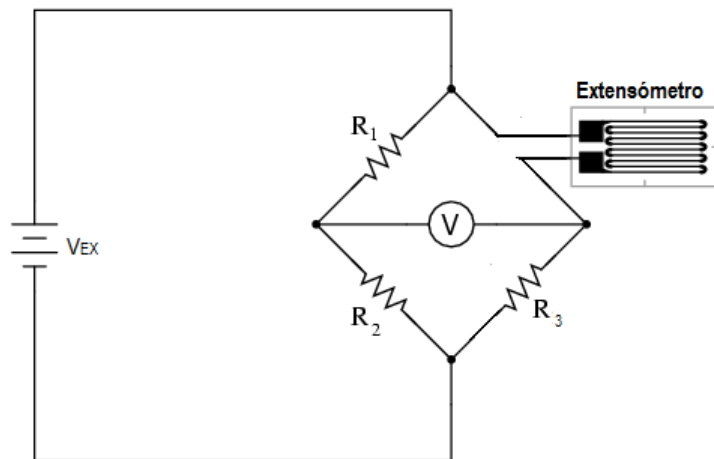


Figura 2-4. Ponte de Wheatstone (quarto).

Por convenção, quando a força é de tração, a saída é positiva e de compressão a saída é negativa, o que leva a apenas duas posições legais para o extensômetro, R_4 ou R_2 , neste caso vamos analisar o extensômetro colocado na posição de R_4 . Portanto, quando a tensão de saída da ponte V_o for positiva, será resultado de uma força de compressão que diminuiu a resistência do extensômetro R_4 , e quando tensão de saída da ponte V_o for negativa, será o oposto [17].

Uma variação da resistência devido a uma força aplicada é dada por:

$$\Delta R = R_g * GF * \varepsilon$$

Onde R_g é a resistência do extensômetro em repouso, GF o fator ganho do extensômetro e ε a deformação.

Se $R_1 = R_2$ e $R_3 = R_g$, substituindo R_4 por $R_g + \Delta R$ chega-se à seguinte expressão da sensibilidade da ponte:

$$V_o/V_{ex} = -GF * \varepsilon / 4 * [1 / (1 + GF * \varepsilon / 2)]$$

Note-se que devido ao termo $1/(1+GF*\varepsilon/2)$ existe não linearidade na resposta deste circuito.

Meia ponte

Analisando o circuito da figura, verifica-se que um dos braços da ponte está totalmente ativo, tornando assim metade da ponte ativa.

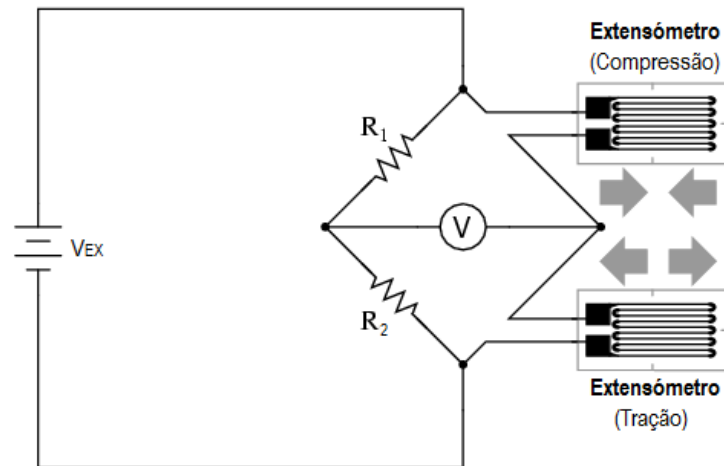


Figura 2-5. Ponte de Wheatstone (meia).

Os extensômetros são resistências, e portanto sofrem um aumento da temperatura com a passagem de corrente elétrica. Este fenômeno influencia o valor de resistência, o que implica em determinados circuitos a alteração da tensão de saída [18]. Nesta configuração o efeito da temperatura pode ser contornado caso os extensômetros sejam montados no mesmo braço, pois assim a variação da resistência devido à temperatura afetará ambos os extensômetros e o valor será anulado, ou seja, a razão da resistência não será alterada, portanto um extensômetro irá compensar o outro e anular a ação da temperatura na saída da ponte [15].

A sensibilidade da meia ponte é dada pela seguinte expressão:

$$V_o/V_{ex} = \frac{-GF * \epsilon}{2}$$

Ponte Completa

Com a configuração da Figura 2-6, todas as resistências são ativas. A ponte completa utiliza quatro extensômetros, em que um dos pares que formam os extensômetros opostos está combinado a trabalhar em compressão e o outro par em tração.

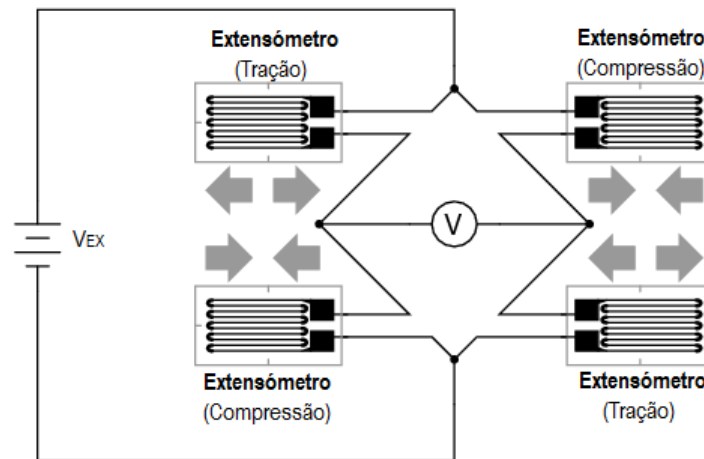


Figura 2-6. Ponte de Wheatstone (completa).

A ponte completa utiliza quatro extensômetros, em que um dos pares que formam os extensômetros opostos está combinado a trabalhar em compressão e o outro par em tração.

Esta configuração forma a ponte completa, o que garante duas vezes mais sensibilidade que a meia ponte e quatro vezes mais que o quarto de ponte sendo uma grande vantagem para a medição de pequenas deformações.

Outra grande vantagem de se usar os extensômetros em ponte de Wheatstone completa é a existência de compensação de temperatura tal como na configuração de meia ponte. Na ponte de Wheatstone (meia e completa) a existência de 2 resistências em cada braço reduz esse efeito, pois todas as resistências têm o mesmo valor e são do mesmo material, portanto sofrem o mesmo aumento de temperatura [15, 17].

A sensibilidade da ponte completa é dada pela seguinte expressão:

$$V_o/V_{ex} = - GF * \epsilon.$$

2.3.2 AMPLIFICAÇÃO E FILTRAGEM

Amplificação

Uma etapa fundamental do condicionamento de sinal para células de carga é a amplificação, devido à tensão de saída das pontes de extensômetros ser muito baixa. Geralmente, grande parte das pontes e outros transdutores de força têm gama de menos de 10 mV de saída por volt de tensão de excitação, logo, com 5 V de excitação, o sinal de saída é 50 mV [19]. Portanto, os circuitos condicionadores de sinal para células de carga normalmente incluem amplificadores de instrumentação (Figura 2-7) com ganho fixo ou variável para elevar o nível de tensão e assim aumentar a resolução da medição [19].

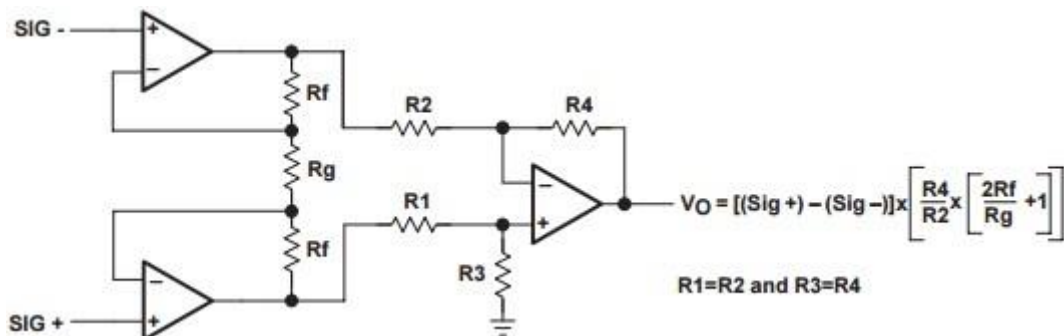


Figura 2-7. Amplificador de Instrumentação. Direitos reservados à Texas Instruments.

Filtragem

Devido à interferência maioritariamente eletromagnética (EMI), pela razão da localização destes sistemas de medição ser normalmente ambientes eletricamente ruidosos cujo ruído é proveniente de outros equipamentos, é essencial existir um processo de filtragem para eliminar ruídos que se possam acoplar ao sinal de saída das pontes extensométricas. Diversas tecnologias, às vezes somadas à inadequação das suas instalações, provocam a emissão de energia de alta frequência, o que perturba o sinal de equipamentos de instrumentação, alterando o valor medido pelos mesmos.

Filtros passa-baixo, quando usados nos circuitos de condicionamento, podem remover o ruído de alta frequência prevalente na maioria das configurações [20].

2.3.3 AJUSTE DE ZERO

Quando uma célula de carga não se encontra sob nenhuma força, é muito improvável que a saída seja exatamente 0V (Figura 2-8).

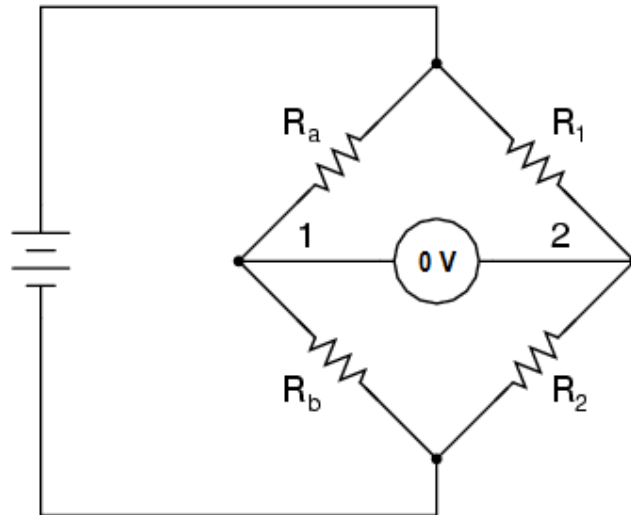


Figura 2-8. Ponte em equilíbrio.

Considerando variações iniciais na resistência entre os braços da ponte devido à estrutura da célula não ser perfeita à tolerância dos extensômetros e à resistência dos fios condutores, a ponte não estará em equilíbrio por si só, existirá uma certa tensão inicial (*offset*) diferente do ideal, que seria zero. A compensação de *offset* é habitualmente feita tanto por *hardware* ou *software*.

O ajuste de zero por *hardware*, consiste na utilização de um potenciômetro ou resistências fixas para equilibrar a ponte e configurar a saída inicial para zero volt, o que leva a se ter que realizar um novo procedimento físico de ajuste para todas as vezes que ocorrer um desequilíbrio diferente.

O ajuste por *software* consiste em subtrair o valor de desequilíbrio inicial em todas as medições. Este procedimento é feito de uma forma mais rápida e automática em relação ao ajuste por *hardware* mas perde pelo facto do *offset* físico da ponte nunca ser removido, levando a perda do alcance dinâmico da saída da célula de carga.

2.4 PLATAFORMA DE FORÇAS

2.4.1 ESTRUTURA MECÂNICA

Antes de avançar com a conceção do novo sistema, é necessário fazer uma análise à plataforma de forças (Figura 2-9) desenvolvida na Escola Superior de Tecnologia e Gestão (ESTiG) para entender o seu funcionamento, assim como estudar o respetivo circuito analógico de condicionamento de sinal.

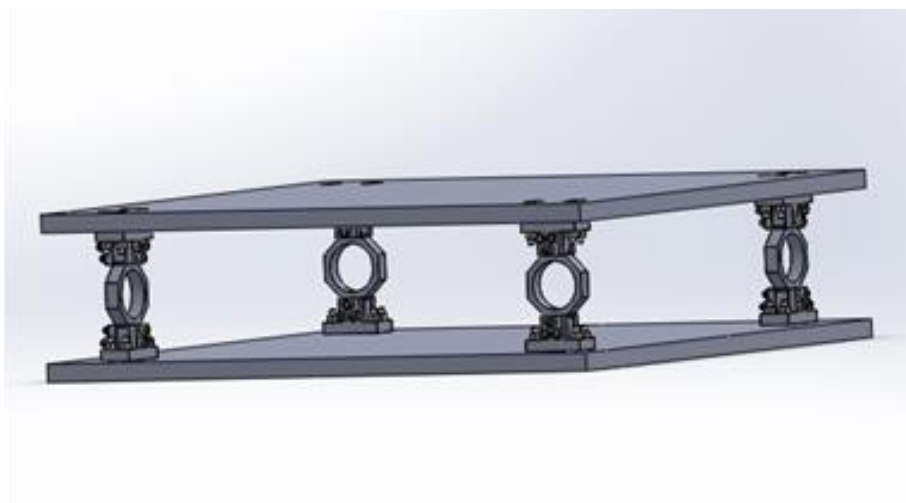


Figura 2-9. Plataforma de forças [21].

As células de carga utilizadas nesta plataforma foram desenvolvidas na ESTIG, têm formato hexagonal e são metálicas, mais concretamente em liga de alumínio (Al 6061). A colocação dos extensómetros obedece à configuração de ponte completa, e cada célula é capaz de medir forças em dois eixos, tendo portanto duas pontes completas de extensómetros de 120Ω , o que dá um total de oito extensómetros por célula.

Simulações realizadas anteriormente na célula de carga, com recurso ao programa de CAD/CAE *SolidWorks*, indicam as zonas de deformações e deslocamentos mais elevados quando são aplicadas as forças horizontais e verticais [21]. Os resultados obtidos nas simulações podem ser vistos na seguinte figura.

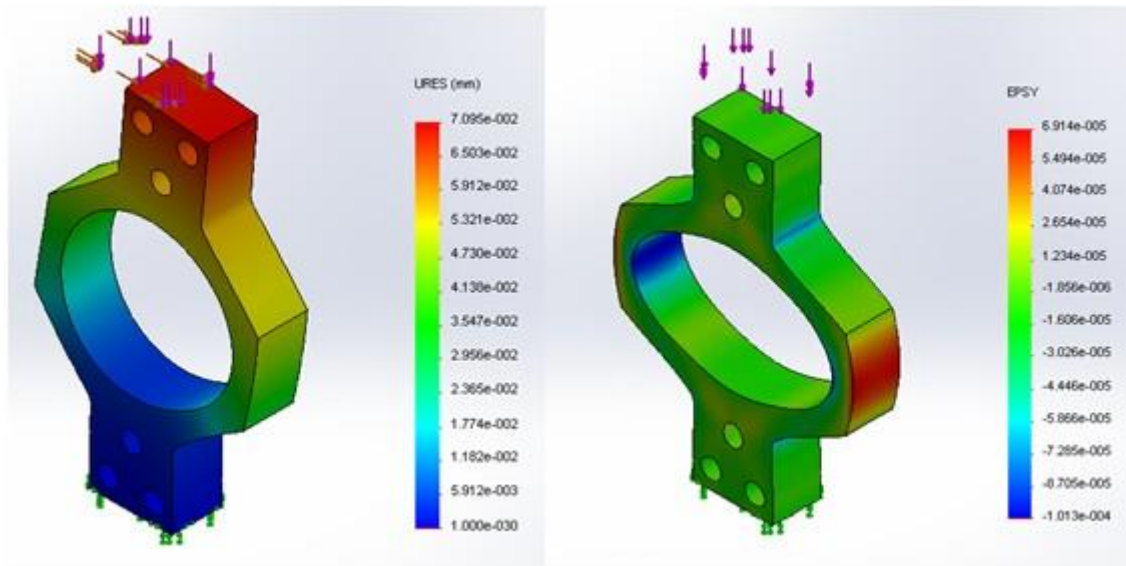


Figura 2-10. Avaliação da deformação [21].

2.4.2 GEOMETRIA DAS CÉLULAS DE CARGA

A célula foi desenhada com vista a sofrer deformação em diversas partes segundo uma determinada força, logo os extensómetros foram colocados estrategicamente, de acordo com a seguinte figura.

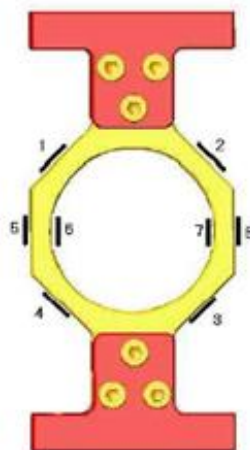


Figura 2-11. Diagrama da célula de carga [21].

Os extensómetros numerados de cinco a 8, pertencem à ponte que sensoriza a ação da componente de força vertical na célula de carga. Quando uma força vertical é aplicada, os extensómetros número 5 e 8 vão sofrer tração enquanto os 6 e 7 vão ser comprimidos, o que vai de encontro ao funcionamento da ponte completa de *wheatstone*. Por outro lado, os extensómetros numerados de 1 a 4, pertencem à ponte que é responsável por sensorizar as deformações na célula relativas à componente horizontal de força, e onde se pode realizar a mesma análise da ponte de extensómetros anterior. Temos portanto, uma célula de carga com dois sinais de saída.

2.4.3 SOLUÇÃO ATUAL DE CONDICIONAMENTO DE SINAL

O circuito de condicionamento de sinal desenvolvido anteriormente na ESTIG é puramente analógico, e faz o condicionamento de sinal para cada ponte de extensómetros, o que leva a serem necessárias oito pcs no total, visto existirem quatro células com duas pontes cada, o que totaliza oito pontes de extensómetros.

Esta pcb (Figura 2-13), tem a funcionalidade de filtragem do sinal, ajuste de zero e *offset* por meio de potenciômetros, e seleção de 3 níveis de ganho: x350, x500 e x1000. A saída é portanto analógica na gama de 0V a 5V, tendo que mais um equipamento complementar o sistema, como por exemplo uma placa de aquisição de dados, para que os valores em tensão possam ser convertidos para unidades de medida, e apresentados.

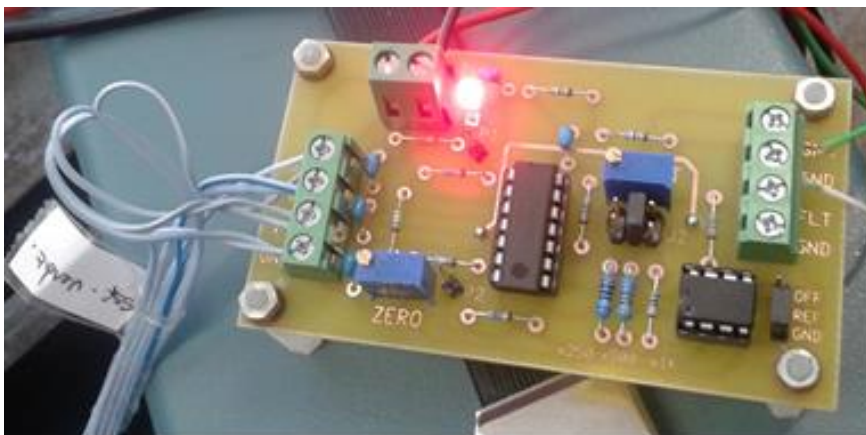


Figura 2-12. Placa analógica de condicionamento de sinal [21].

Atendendo ao diagrama da Figura 2-14, que representa a parte principal do circuito em análise, compreende-se que é constituído por vários componentes eletrónicos, nomeadamente amplificadores operacionais e resistências. São usados dois ampops LM324 em modo não inversor, com tensões nas entradas inversoras flutuantes entre si através de um potenciómetro, possibilitando o ajuste de zero. Um terceiro ampop com referência INA122P é usado, para amplificar o sinal com vista a tornar maior a sensibilidade da saída. Três condensadores fazem desacoplamento de ruído entre as saídas da célula de carga e as linhas de alimentação.

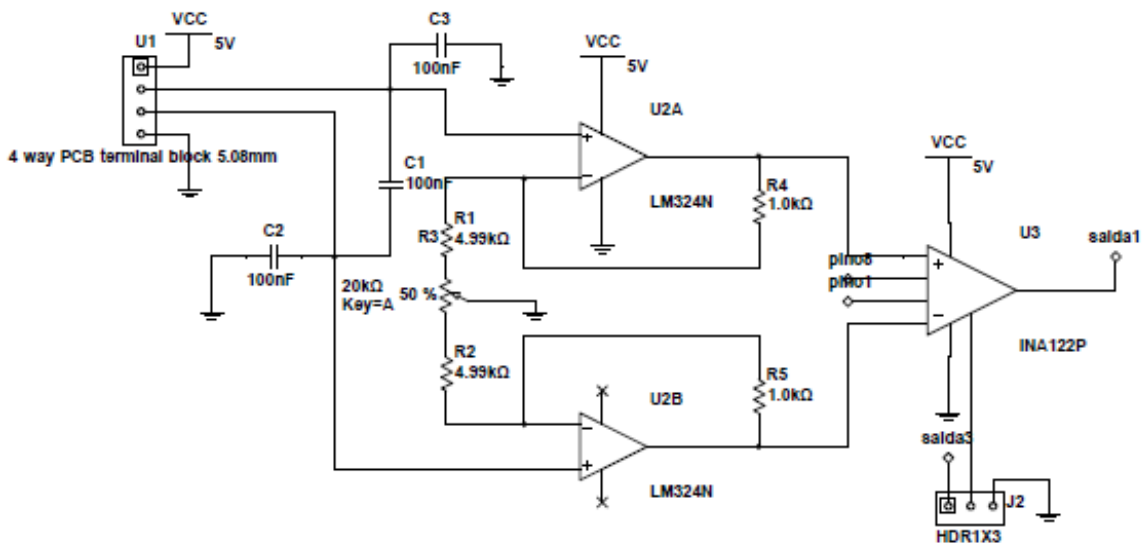


Figura 2-12. Esquema do circuito analógico de condicionamento de sinal [21].

Um circuito auxiliar na mesma pcb, faz o ajuste de *offset* da saída, permitindo colocar a mesma a uma tensão de 2.5V, para ser possível medir forças nos dois sentidos de cada eixo.

Para efeitos de teste deste circuito, foi fabricada uma placa na ESTIG, e montada uma ponte de *wheatstone* com resistências fixas com tolerancia alta, para simular os extensómetros a partir do próprio erro das mesmas. Com ganho de x500 foi possível perceber a elevada sensibilidade do circuito, visto que o próprio calor corporal por meio dos dedos em contacto com as resistências, fazia variar o seu valor de saída, na ordem das décimas de milivolt.

O circuito anteriormente desenvolvido apresenta vários pontos a ser melhorados, a começar pelo tamanho da própria PCB que mede $8,5 \times 5$ cm, o facto do ajuste de zero e *offset* da ponte ser manual, assim como o controlo de ganho estar limitado a três valores possíveis.

3. SISTEMA INTEGRADO PARA MEDIÇÃO DE FORÇAS:

3.1 ARQUITETURA DO SENSOR INTELIGENTE

3.1.1 EQUILÍBRIO DA PONTE

Depois de devidamente estudado o problema, e dados os objetivos impostos, começou o desenvolvimento do novo circuito que possuindo uma dimensão compacta, pudesse ser capaz de fazer o equilíbrio automático de cada ponte de extensómetros, amplificar o sinal e comunicar com um *master* através do barramento I2C, para cada ponte de extensómetros.

Numa fase inicial, optou-se por dar lugar ao desenvolvimento de um circuito para que fosse possível fazer o equilíbrio da ponte de *wheatstone* de uma forma automática, o que levou à ideia de controlar a parte responsável pelo ajuste de zero do antigo circuito analógico com pwm.

A ideia, que chegou a ser colocada na prática não mostrou ser muito viável, visto os componentes envolvidos não permitirem a placa assumir dimensões reduzidas, entre outros que se irão relatar em seguida. A figura 3-1 mostra a montagem do microcontrolador ATTiny85, aliado a um filtro passa-baixo que controlava por pwm as entradas inversoras dos LM324.

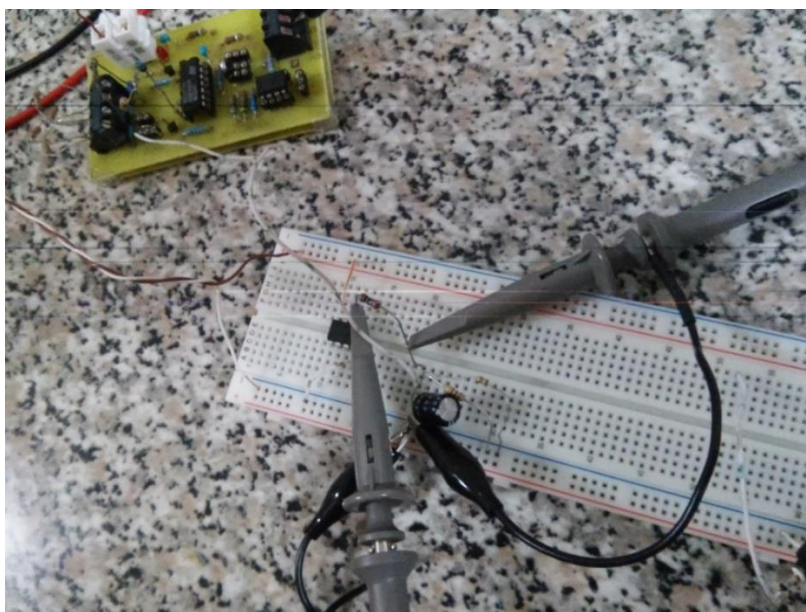


Figura 3-1. Circuito de ajuste de zero controlado por pwm.

O Attiny85 trata-se de um microcontrolador desenvolvido pela Atmel Corporation, cujo *layout* de pinos encontra-se na seguinte figura:

Pinout ATtiny25/45/85



Figura 3-2. Pinos da gama ATTiny. Direitos reservados à Atmel Corporation.

Algumas características de maior importância deste componente foram reunidas a partir da folha de dados do fabricante [22]:

- Tensão de funcionamento: 2.7 - 5.5V
- Microcontrolador 8-bit AVR® de alto desempenho
- Velocidade: 0 – 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
- Memória *flash* de 8KBytes
- EEPROM integrada de 512 Bytes
- Memórias não voláteis
- Vida: 10000 ciclos de reprogramação
- Consumo: 1 MHz, 1.8V: 300 μ A
- ADC de 10-bit
- USI – *Universal Serial Interface*
- Temporizador/contador de 8-bit com *prescaler*
- 2 Saídas de PWM (8-bit)

O sinal de pwm foi suavizado, a partir de um filtro implementado a partir de um condensador e uma resistência, cujo melhor comportamento foi obtido com os valores de 22uF e 1kΩ respectivamente. Um sistema em malha fechada foi criado, com vista em manter o sinal de pwm com um determinado nível de *duty-cycle* para que a saída do amplificador fosse nula. Procedeu-se à programação do microprocessador ATiny85 para tal, utilizando uma das entradas analógicas do mesmo, para leitura do sinal da saída do circuito analógico. A escolha deste processador teve como base as suas dimensões e os recursos existentes na ESTiG.

A linguagem de programação usada foi o C, e o programador (Figura 3-3) foi concebido tipo *shield*, em placa perfurada tendo como base o Arduino em modo programador (*firmware* ISP), utilizando portanto o Arduino IDE para compilar e programar o ATTiny85.

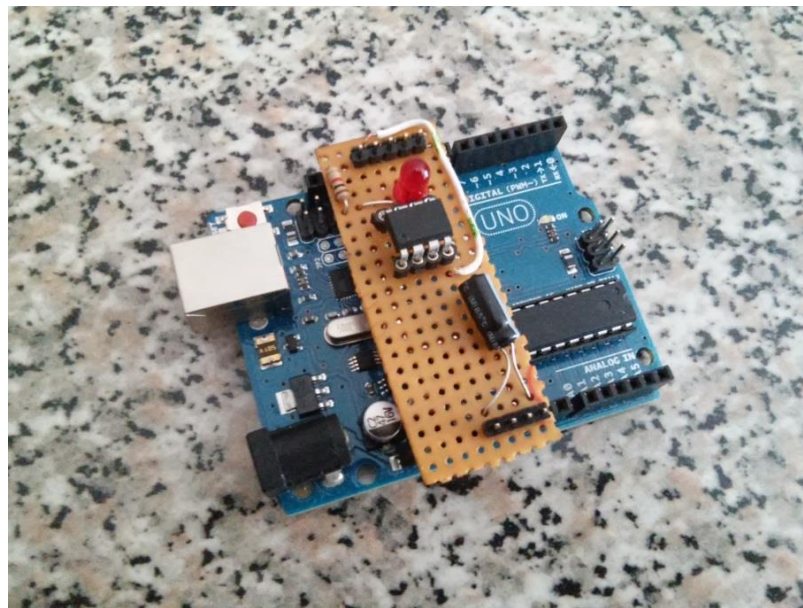


Figura 3-3. Programador do ATTiny85 com Arduino ISP.

Esta ideia foi baseada num projeto encontrado *online* [21], com o fim de encontrar o melhor método para programar o microprocessador. A plataforma IDE foi modificada com recurso ao *plugin* (disponibilizado na página *online* do projeto) dedicado à programação de processadores da gama ATTiny.

Devido ao sinal de controlo de equilíbrio por pwm assumir uma só polaridade, era impossível equilibrar a ponte em ambos os sentidos a partir deste método. Posto isto, a parte de ajuste de zero do circuito foi alterada para que se encontrasse em desequilíbrio na ausência de pwm, por meio de valores de resistências diferentes em cada uma das entradas inversoras dos amplificadores LM324, para que o ajuste de zero pudesse ser processado desta forma.

O resultado geral não foi bom, visto a gama em que era possível fazer o ajuste de zero ser muito baixa, devido ao facto do pwm gerado pelo ATTiny85 possuir resolução de apenas 8 *bit*, a gama de 5V teve que ser reduzida para apenas 1V, através de um divisor de tensão, para aumentar a precisão de controlo.

No decorrer das montagens e testes, foi usada inicialmente outra célula de carga (Figura 3-4), oriunda de outro projeto desenvolvido na ESTiG, que não a destinada a este trabalho visto ter sido montada posteriormente, o que ajudou melhor a perceber a capacidade adaptativa do novo sistema desenvolvido.

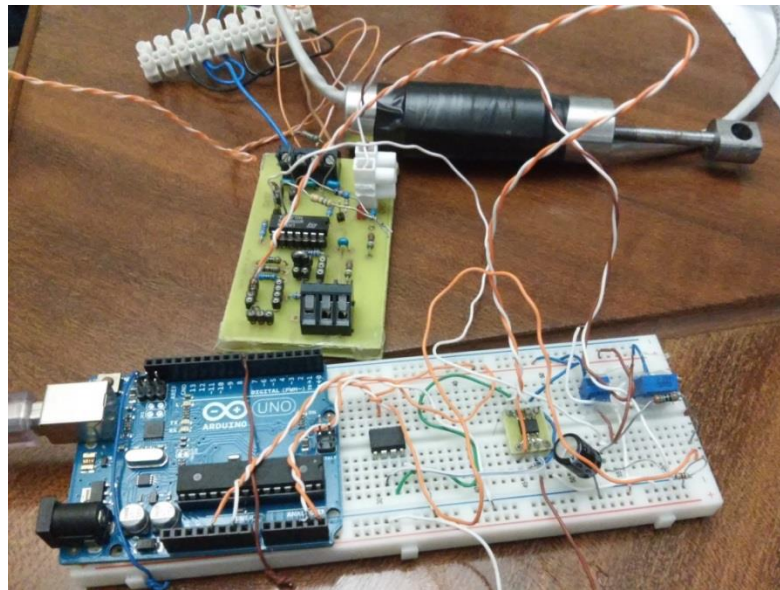


Figura 3-4. Célula de carga cilíndrica.

Mediante este problema, efetuou-se uma pesquisa acerca de componentes dedicados ao condicionamento de sinal, que tivessem *offset* programável digitalmente, do qual se destacou o amplificador AD8556 que apresenta outras características, descritas no próximo subcapítulo, interessantes para este projeto.

3.1.2 AMPLIFICADOR PROGRAMÁVEL

Segundo o *datasheet* deste componente, trata-se de um amplificador *zero-drift* (possível tradução: sem desvio), desenhado para converter as tensões de saída de pontes de extensómetros e outros sensores de precisão, numa gama de tensão bem definida e conta com ganho e *offset* digitalmente programáveis. Utiliza tecnologia patenteada da Analog Devices para criar um sinal preciso e flexível, num formato compacto. O encapsulamento do integrado é fornecido no formato SOIC_N de 8 pinos e LFCSP_VQ de 16 pinos e 4x4 mm. A configuração de pinos do encapsulamento SOIC_N deste componente pode ser visto na seguinte figura:



Figura 3-5. AD8556. Direitos reservados à Analog Devices.

O ganho pode ser programado numa gama entre 70 e 1280 através de uma interface de série, e pode ser simulado ou programado permanentemente através da tecnologia *polyfuse*, utilizando apenas um pino. O *offset* também é digitalmente programável, é determinado proporcionalmente à tensão de alimentação, e pode ser ajustado com uma resolução superior a 0.4% entre a tensão de alimentação (pinos VDD e VSS).

Este amplificador também inclui filtros EMI internos nos pinos VNEG, VPOS, FILT e VCLAMP o que permite conceber um *hardware* com imunidade a ruído. A função de filtragem passa-baixo é implementada através de um único condensador externo. Em adição à baixa tensão de *offset*, baixo *drift* de entrada, o AD8556 também inclui *pull-ups* e *pull-downs* que permitem deteção de circuito aberto e curto-circuito na entrada diferencial. Através de uma tensão de referência externa no pino VCLAMP é possível estabelecer um limite de saída por uma questão de segurança. Quando usado

com um ADC referenciado à mesma fonte de alimentação, o sistema torna-se imune às possíveis variações de alimentação. A temperatura especificada de funcionamento situa-se entre os -40°C e os $+140^{\circ}\text{C}$ e pode ser alimentado com tensões simples de 4.5V a 5.5V. O diagrama de funcionamento encontra-se patente na figura 3-6.

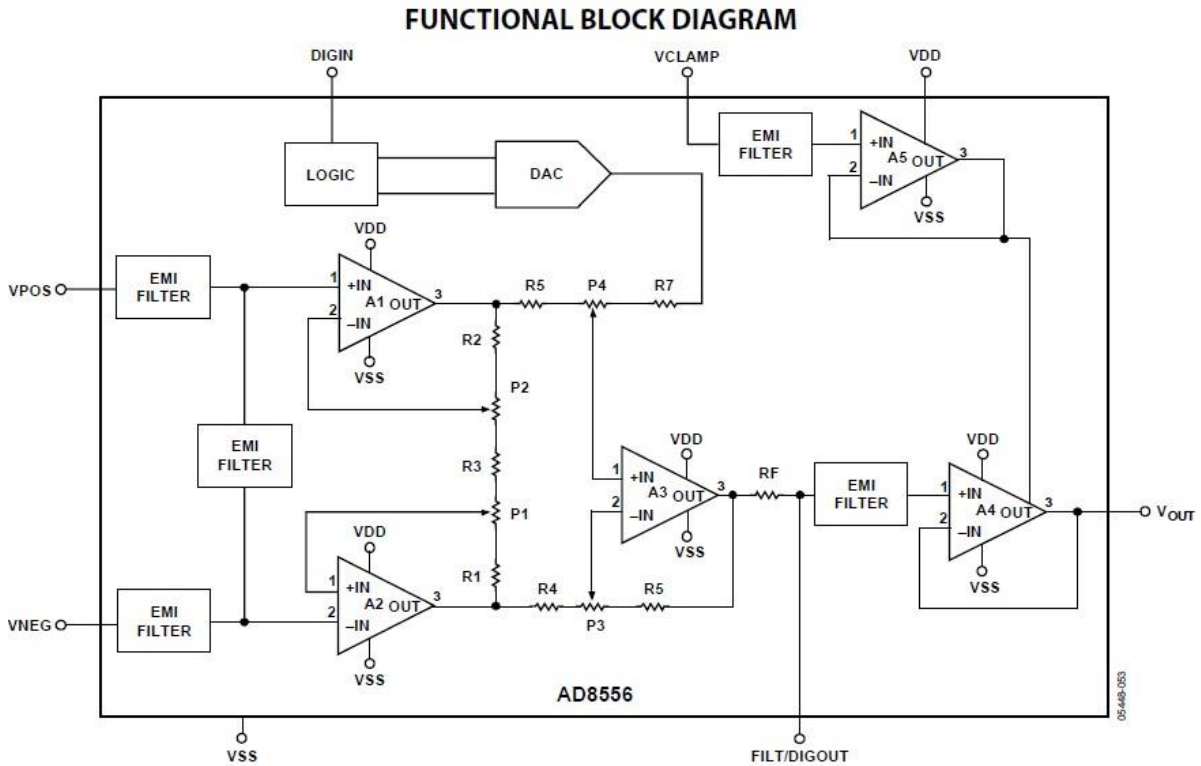


Figura 3-6. Diagrama funcional do AD8556. Direitos reservados à Analog Devices.

Os componentes A1, A2, R1, R2, R3, P1 e P2 fazem parte do estágio 1 de ganho (GAIN1) do amplificador diferencial onde P1 são P2 potenciômetros digitais que permitem GAIN1 variar entre 4.0 e 6.4 com 7-bit de resolução (ver Anexo 1) dando uma resolução de 0.37%. A equação do GAIN1 em função do código é traduzida pela seguinte expressão:

$$GAIN1 \approx 4 \times \left(\frac{6.4}{6}\right)^{\left(\frac{Code}{127}\right)}$$

Por outro lado, os componentes A3, R4, R5, R6, R7, P3, e P4 fazem parte do segundo estágio de ganho (GAIN2) do amplificador, onde P3 e P4 são os potenciômetros que controlam o GAIN2 entre 17.5 e 200, em 8 passos (ver Anexo 1). A1 e A2 e A3 são ampops com vista a minimizar erros de *offset* de entrada.

Apenas com um condensador ligado entre o pino FILT/DIGOUT e VSS ou VDD, é ativo o filtro passa-baixo do AD8556. O sinal filtrado, através do amplificador unitário A4, é convertido na saída de baixa impedância VOUT. A resistência RF é de 18 kΩ, permitindo um filtro passa-baixo de 880 Hz, quando ligado um condensador externo de 10 nF.

Um DAC de 8-bit é usado para gerar um *offset* variável para o amplificador. Este DAC usa como referência VSS e VDD, com o valor de 0 e 255 respetivamente para o processo de programação. A resolução é portanto equivalente a 0.39% da diferença entre VDD e VSS, o que para uma fonte de 5V dará 19.5mV. A saída do DAC, VDAC é dada por:

$$VDAC \approx \frac{Code + 0.5}{256} \times (VDD - VSS) + VSS$$

E a tensão de saída (VOUT) é dada por:

$$VOUT = GAIN (VPOS - VNEG) + VDAC,$$

Onde o ganho (GAIN) é o produto entre os ganhos de estágio 1 (GAIN1) e dois (GAIN2).

O AD8556 dispõe de deteção de falhas quando as entradas VPOS, VNEG, ou VCLAMP são curto-circuitadas a VDD ou VSS. A figura 3-7 mostra as regiões de tensão em VPOS, VNEG, e VCLAMP que disparam uma condição de erro. Quando o erro ocorre, a saída VOUT é ligada a VSS, apresentando assim uma tensão de saída nula em caso de *single-supply*.

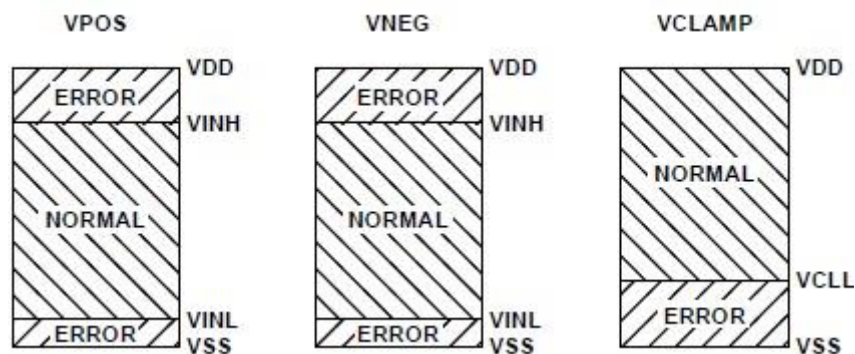


Figura 3-7. Detecção de erros. Direitos reservados à Analog Devices.

Onde os valores de tensão são dados na seguinte tabela:

Tabela 1. Tensões de detecção de erros. Direitos reservados à Analog Devices.

Voltage	Min (V)	Typ (V)	Max (V)	VOUT Condition
VINH	2.95	3.0	3.05	Short to VSS fault detection
VINL	1.95	2.0	2.05	Short to VSS fault detection
VCLL	1.05	1.1	1.15	Short to VSS fault detection

Por outro lado, se algumas das entradas referidas anteriormente forem deixadas em circuito aberto, esta condição também fará com que VO_{UT} seja ligada a VSS. As correntes que definem a detecção de erro de circuito aberto estão presentes na tabela que se segue:

Tabela 2. Correntes de detecção de erros. Direitos reservados à Analog Devices.

Mnemonic	Typical Current	Goal of Current
VPOS	49 nA pull-up	Pull VPOS above VINH
VNEG	49 nA pull-up	Pull VNEG above VINH
VCLAMP	0.2 μ A pull-down	Pull VCLAMP below VCLL

Analisando o *datasheet* novamente com fim a estudar o protocolo de comunicações, tira-se que existem dois modos de programação em que um é permanente, queimando fusíveis de *polysilicone* internamente neste circuito integrado, o qual não tem interesse para esta aplicação pois necessita-se de alterar os valores de ganho e *offset* inúmeras vezes, e o outro modo é em “simulação”.

Para minimizar o número de pinos e espaço do AD8556, o fabricante optou por uma interface de programação com apenas um fio, através do pino DIGIN, o qual possui histerese e é deixado com um sinal DC baixo quando não está a ser programado para minimizar a possibilidade de ser ativo inadvertidamente.

Um pulso rápido no pino DIGIN, com duração entre 50 ns e 10 us, carrega o valor digital 0 no registo, enquanto um pulso mais longo, de cerca de 50 us, carrega o valor digital 1. O tempo entre os pulsos deve ser pelo menos de 10 us. Para $VSS = 0V$, uma tensão entre VSS e $0.2 \times VDD$ é reconhecida como estado baixo enquanto uma tensão entre $0.8 \times VDD$ e VDD é reconhecida como estado alto. O diagrama da figura 3-8 ilustra a forma do sinal para o código digital 010011:

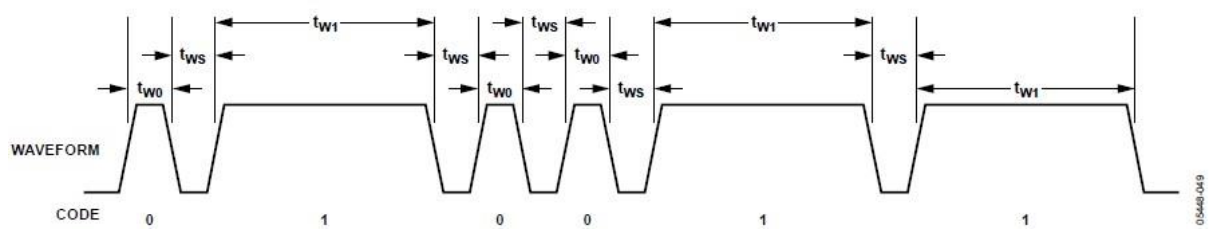


Figura 3-8. Diagrama de um sinal digital. Direitos reservados à Analog Devices.

A tabela complementa a figura anterior, legendando os tempos dos pulsos do sinal digital:

Tabela 3. Duração de pulsos.

Parâmetro	Descrição	Especificação
t_{w0}	Largura do pulso que carrega “0”	Entre 10 ns e 10 μ s
t_{w1}	Largura do pulso que carrega “1”	$\geq 50 \mu$ s
t_{w2}	Largura entre pulsos	$\geq 10 \mu$ s

Como se pode verificar na tabela 4, existem 6 campos na mensagem de 38 *bit* que será enviada ao AD8556 para a programação do mesmo. O campo 0 e 5 são respectivamente os pacotes de início e fim da mensagem com 12 *bit* cada, para garantir que a mensagem transferida é válida e portanto ativar a descodificação dos restantes campos.

Tabela 4. Formato da palavra de 38 *bit*. Direitos reservados à Analog Devices.

Table 10. 38-Bit Serial Word Format

Field No.	Bits	Description
0	0 to 11	12-Bit Start of Packet 1000 0000 0001
1	12 to 13	2-Bit Function 00: Change Sense Current 01: Simulate Parameter Value 10: Program Parameter Value 11: Read Parameter Value
2	14 to 15	2-Bit Parameter 00: Second Stage Gain Code 01: First Stage Gain Code 10: Output Offset Code 11: Other Functions
3	16 to 17	2-Bit Dummy 10
4	18 to 25	8-Bit Value Parameter 00 (Second Stage Gain Code): 3 LSBs Used Parameter 01 (First Stage Gain Code): 7 LSBs Used Parameter 10 (Output Offset Code): All 8 Bits Used Parameter 11 (Other Functions) Bit 0 (LSB): Master Fuse Bit 1: Fuse for Production Test at Analog Devices Bit 2: Parity Fuse
5	26 to 37	12-Bit End of Packet 0111 1111 1110

O campo 1, de 2 *bit*, é usado para definir se a programação do amplificador vai ser simulada ou permanente, e em seguida, o campo 2 também de 2 *bit* revela qual o parâmetro vai ser enviado, por exemplo, o ganho. O campo 3 divide a mensagem e assegura que nenhuma combinação de dados possa inadvertidamente ativar os campos de início e fim da mensagem. O penúltimo campo a ser enviado é um valor com 8 *bit* e é respetivo ao parâmetro do campo 2, ou seja, se este campo era o ganho, no campo 3

estará presente o valor do ganho. Os códigos binários referentes a cada campo estão presentes na tabela.

Em cada campo, o MSB tem que ser enviado primeiro, e o LSB depois. O registo possui *reset* ao iniciar para minimizar o risco de ser programado por lapso e ocorre quando VDD se encontra entre 0.7 V e 2.2 V.

Montagem e teste do AD8556

Com a chegada das amostras do circuito integrado AD8556 oferecidas pela Analog Devices, procedeu-se ao desenho e fabrico de uma pcb (Figura 3-10) cujo *layout* se encontra na Figura 3-9. A função foi converter os pinos em configuração *SOIC_N* para *trough hole*, a fim de se poder encaixar o circuito integrado numa *breadboard*, e efetuar testes a fim de verificar na prática se o componente correspondia ou não ao pretendido.

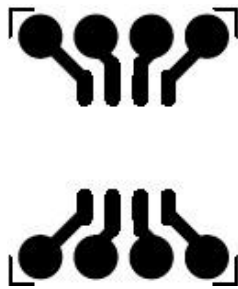


Figura 3-9. Conversor de pinos (*layout*).



Figura 3-10. Conversor de pinos (Pcb).

Existindo a necessidade de se utilizar um microprocessador a fim de programar o amplificador em modo de simulação, optou-se novamente pelo Attiny85. Fez-se a montagem do circuito para tal, e o processador foi programado em C, através do Arduino IDE, respeitando o protocolo de comunicações do AD8556 descrito anteriormente.

O amplificador mostrou-se bastante interessante para resolver alguns problemas deste projeto, vários valores de ganho e *offset* foram enviados para o amplificador com

êxito, e com a resolução de 0.39% de *offset* foi colocada a possibilidade de equilibrar a ponte através do ajuste do mesmo de forma automática.

Dada a existência de desequilíbrio inicial da ponte, apesar do *offset* do amplificador estar a 0V, a saída apresentava um valor de 0.2V, lido pelo multímetro. Trocando a polaridade à saída da ponte extensométrica, o valor de saída no amplificador apresentado era de 0V, existia portanto uma “zona morta” de 0.2V que não poderia ser lida visto o sistema estar a ser alimentado com *single supply*.

Para resolver este problema, e como cada ponte de extensómetros tem que apresentar leituras nos dois sentidos do eixo, inicialmente foi colocado um *offset* de 2.5V na saída, para poder existir excursão de valores entre 0V a 2.5V num dos sentidos, e entre 2.5V e 5V no outro sentido do eixo. Enviado *offset* de 2.5V e com a saída da ponte na polaridade inicial, o AD8556 apresentou cerca de 2.7V e não 2.5V, e com a ponte na polaridade inversa, apresentou cerca de 2.3V, o que facilmente se concluiu que o erro de desequilíbrio da ponte era de 0.2V à saída do amplificador.

Posto isto, o ATTiny85 foi programado para que seguidamente de colocar o valor de *offset* de 2.5V no AD8556, determinasse automaticamente o novo *offset* para que a saída do amplificador fosse 2.5V através das seguintes equações:

$$Tensão_{offset} = 5 - Tensão_{lida}, \text{ caso tensão lida superior a } 2.5V \text{ e,}$$

$$Tensão_{offset} = 2.5 + (2.5 - Tensão_{lida}), \text{ caso tensão lida inferior a } 2.5V.$$

Com alguns dos problemas já tratados, nomeadamente o equilíbrio automático da ponte, o ajuste de ganho automático, a amplificação e filtragem do sinal, partiu-se para a parte de comunicações entre *master/slave*. O esquema (Figura 3-9) mostra o circuito final para a PCB de cada ponte, que desempenhará a função de *slave* no barramento de comunicação I2C por parte do ATTiny85 visto este dispor de uma interface USI para o efeito.

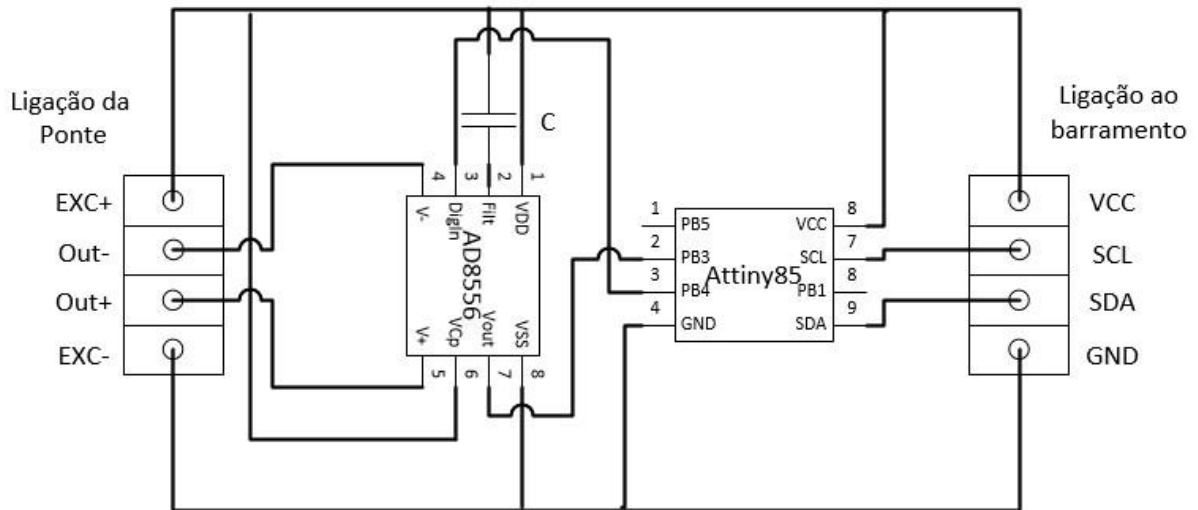


Figura 3-9. Diagrama do circuito desenvolvido.

3.1.3 COMUNICAÇÃO MASTER/SLAVE

O protocolo I2C ou IIC (*Inter IC*) é considerado popular e com provas mais que dadas no que toca à comunicação por dois fios, chamada de barramento IIC. Este protocolo desenvolvido inicialmente pela Philips, pode ser encontrado em larga escala, em várias aplicações que envolvem circuitos eletrônicos com capacidade de comunicação digital, e que funcionam em rede, com forte potencial de crescimento.

Este protocolo usa apenas duas linhas bidirecionais em coletor aberto, (*Serial Data - SDA*) e (*Serial Clock - SCL*) com resistências de *pull-up*, e as tensões tipicamente utilizadas são +5V ou +3.3V [24].

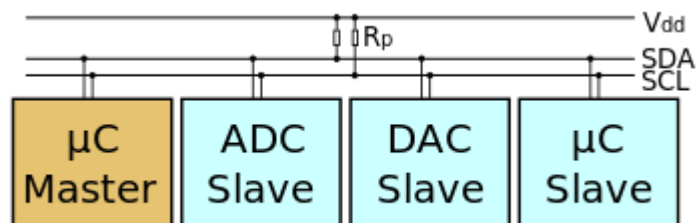


Figura 3-10. Esquema de comunicação IIC.

No âmbito deste projeto, pretende-se desenvolver uma rede IIC do tipo *master/slave*, onde células de carga serão convertidas em *smart sensors*, com a adição do novo circuito de condicionamento de sinal digital, que será o dispositivo escravo na rede IIC, e terá um endereço associado.

Com a necessidade de existência de um dispositivo mestre no sistema para lidar com os vários circuitos escravos das células de carga, foi escolhido o Arduino Uno para desempenhar tal tarefa, já anteriormente utilizado na programação do ATtiny85. O Arduino é uma plataforma *open-source* baseada numa simples placa com entradas e saídas tanto digitais como analógicas, entre elas, as linhas de comunicação IIC. Esta plataforma, possui o seu próprio ambiente de desenvolvimento, que implementa a linguagem C.

A placa do Arduino pode ser usada para desenvolver objetos interativos autónomos ou pode estar ligada a um computador. O Ambiente de desenvolvimento (IDE) *open-source* pode ser obtido gratuitamente (atualmente disponível para Mac OS X, Windows, e Linux). O Arduino Uno é uma versão do Arduino e possui uma placa eletrónica construída em torno do processador ATmega328P (Figura 3-11) [25].

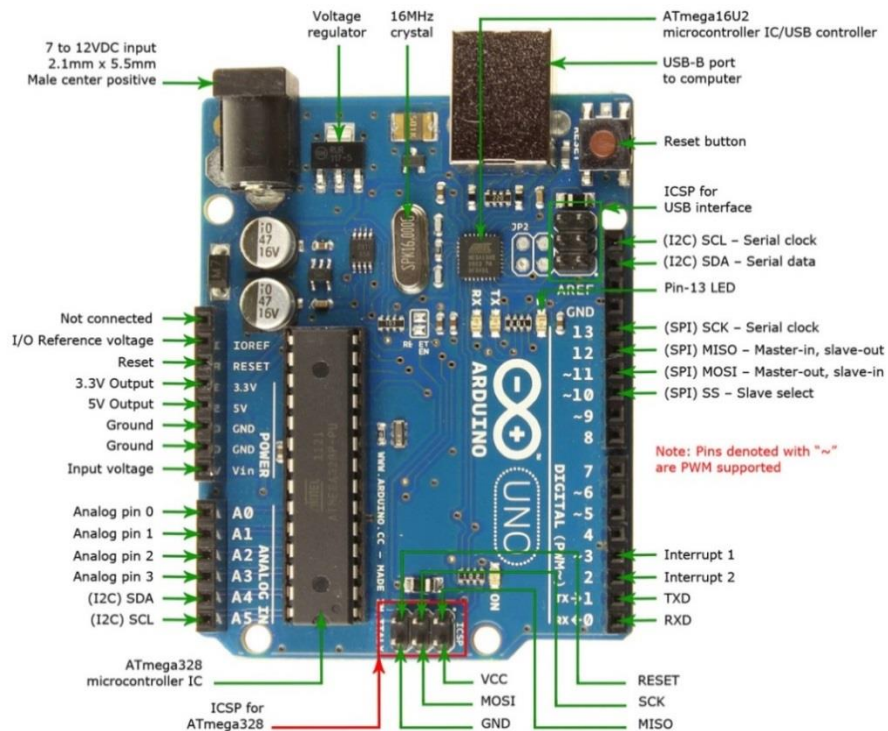


Figura 3-11. Vista do Arduino Uno. Direitos reservados à Robomart.

Algumas das características do Arduino Uno relevantes para este projeto são:

Voltagem operacional: 5V

Voltagem de alimentação (recomendada): 7-12V

Voltagem de alimentação (limites): 6-20V

Pinos I/O digitais: 14

Pinos de entrada analógica: 6 (2 dois quais podem ser usados para I2C)

Corrente contínua por pino I/O: 40 mA

Corrente contínua para o pino 3.3V: 50 mA

Memória *flash* 32 KB (2KB usados para o *bootloader*)

SRAM: 2 KB

EEPROM: 1 KB

Velocidade de *clock*: 16 MHz

Endereço I2C: *7-bit*

Este dispositivo pode ser alimentado por uma fonte externa ou pela própria porta USB, e possui uma saída de alimentação regulada de 5V, o que permite esta placa servir também como fonte de alimentação para as placas *slave*.

O ATTiny85 possui a função de processador local no circuito de condicionamento de cada ponte, sendo este responsável pela comunicação IIC da parte escrava. A biblioteca que permite a comunicação deste com o Arduino, chama-se *TinyWireS* e foi descarregada da internet no *site* dedicado ao Arduino [26].

Dada a existência de uma interface entre o programa LabView e o Arduino Uno, denominada LIFA, optou-se por realizar um programa em LabView que controlasse o Arduino Uno e este funcionasse como *master* remoto [27]. O LabVIEW (acrónimo de *Laboratory Virtual Instrument Engineering Workbench*) é uma linguagem de programação gráfica, desenvolvida pela National Instruments, também chamada de "G". Os principais campos de aplicação do LabVIEW são a realização de medições e a automação.

Os programas em LabVIEW são chamados de instrumentos virtuais ou, simplesmente, IVs. São compostos pelo painel frontal, que contém a interface de utilizador, e pelo diagrama de blocos, que contém o código gráfico do programa. O seu

desempenho é comparável à exibida pelas linguagens de programação de alto nível. Os VIs imitam instrumentos físicos, tanto na sua aparência e funcionamento.

Uma vantagem de usar o LabVIEW, é ser compatível com ferramentas de desenvolvimento semelhantes e podem trabalhar simultaneamente com programas de outra área de aplicação. Também pode ser usado em vários sistemas operativos, incluindo Windows e UNIX. Outra das vantagens mais importantes deste linguagem de programação é que ele permite fácil integração com vários tipos de *hardware*, como placas de aquisição de dados (DAQs), dispositivos que possuem comunicação por SPI ou I2C, entre outros [28].

É um programa virado para a instrumentação virtual, que tem numerosas ferramentas de apresentação em gráficos, botões, mostradores e controlos, que são muito esquemáticas e versáteis.

3.2 SOFTWARE DE CONTROLO E REGISTO

Desenvolvimento do programa em LabVIEW

Painel Frontal

O painel frontal do programa desenvolvido em LabVIEW para controlo do Arduino como *master*, tem como funcionalidade permitir a interface entre o utilizador e o código do programa (diagrama de blocos).

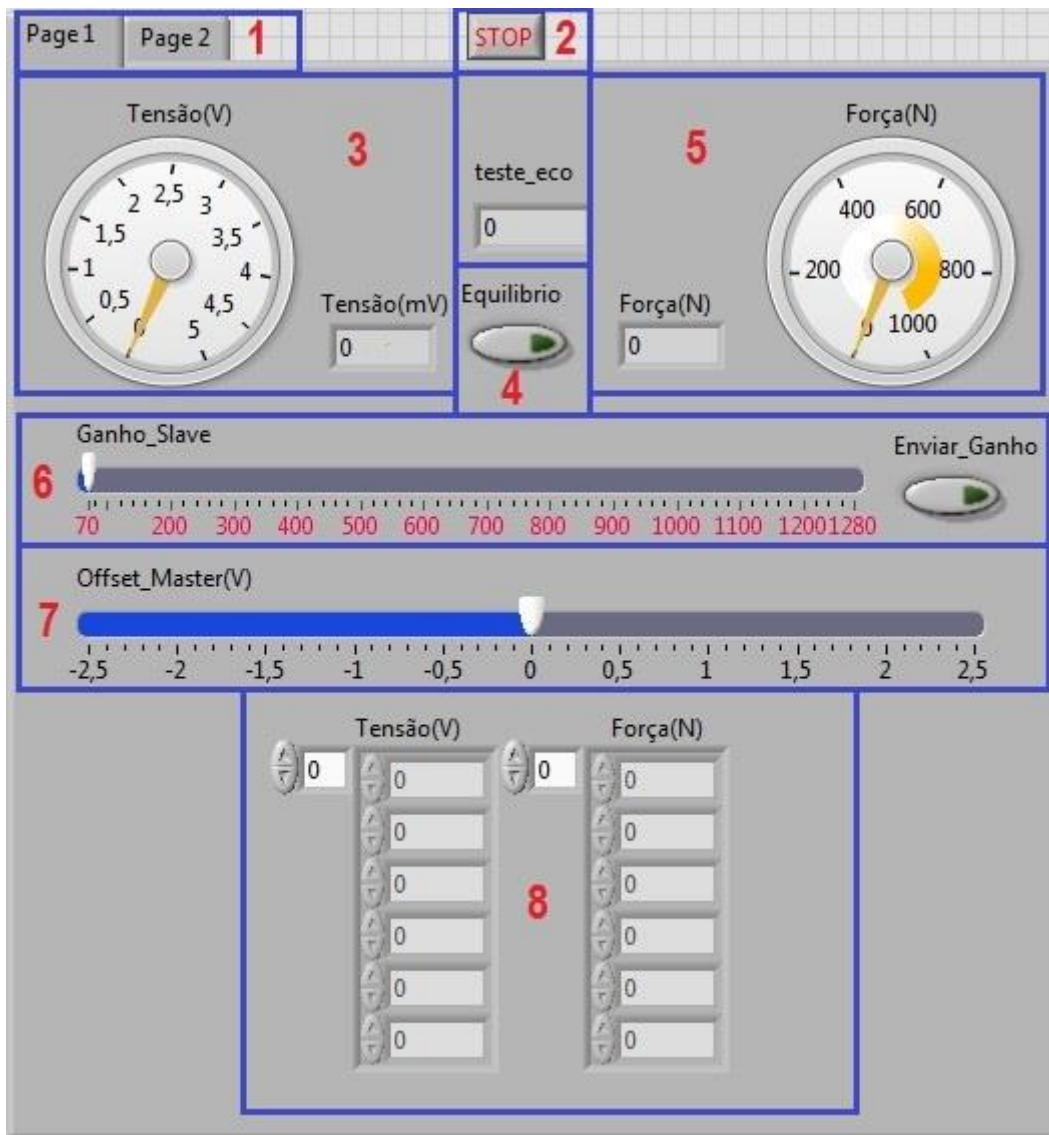


Figura 3-12. Painel Frontal do programa.

Com vista a explicar as suas funções, este painel, que encontra-se na Figura 3-12, foi dividido em várias áreas delimitadas a azul e respetivamente numeradas:

1. Seletor de página, respetiva ao escravo 1 ou 2
2. Botão “*STOP*” para paragem do programa
3. Valor da tensão medida, com mostrador analógico (V) e digital (mV)
4. Botão de equilíbrio da ponte (2.5V)
5. Valor da força em Newton, com mostrador do tipo analógico e digital
6. Seletor do valor de ganho e botão de envio do mesmo para o escravo
7. Nível de *Offset* calculado pelo master
8. Tabela destinada a introduzir 6 valores de calibração

Os mostradores analógicos são do tipo *knob* e os digitais *numeric*, e assumem valores do tipo *float* e *double* respetivamente para a tensão e força. No LabVIEW, os valores do tipo *float* que usam menor espaço de armazenamento (32 *bit*) são denominados *single-precision floating-point* (SGL). Os tipos de variáveis dos mostradores foram automaticamente definidos pelo LabVIEW, através da opção “*adapt to source*”.

A seleção do valor de “Ganho_Slave” e “Offset_Master” é feita por meio de botões *slide*, onde os valores numéricos são do tipo *unsigned integer* (16 *bit*). Os botões “Equilíbrio” e “Enviar_Ganho” são booleanos, assumindo o valor falso em repouso e verdadeiro quando pressionados.

Na parte inferior do painel frontal, encontram-se os *arrays* para introdução dos valores de calibração da célula, do tipo SGL, com 6 valores distintos. Este número pode variar facilmente, se o comprimento vertical dos *arrays* for reajustado na própria janela.

Todos estes itens encontram-se no diagrama de blocos do programa com o mesmo nome, e respetivas ligações às demais funções.

Diagrama de blocos

Devido às dimensões do diagrama de blocos, este será apresentado completo no Anexo 2. O seu funcionamento será explicado por partes neste subcapítulo, complementado com imagens devidamente comentadas.

Para que o programa de controlo estabeleça comunicação com o Arduino pela porta USB, a LIFA (que foi instalada no LabVIEW e no Arduino) possui dois blocos para este efeito apresentados na Figura 3-13. Necessita do bloco “*Init*” no início do programa, devidamente configurado para a versão do Arduino em uso, e do bloco ”*Close*” no fim.



Figura 3-13. Blocos da LIFA para comunicação com Arduino.

Como este programa tem o objetivo de verificar o pleno funcionamento de dois sensores inteligentes por IIC, a comunicação com os escravos com endereço 1 e 2 é possível, a partir da seleção das páginas 1 e 2 respetivamente, patentes no painel frontal. Cada página possui o mesmo diagrama de blocos, diferindo apenas no endereço IIC do escravo respetivo.

A LIFA possui três blocos dedicados à comunicação I2C para o Arduino, onde o “*I2C Init*” inicia uma nova comunicação. O envio e receção de 1 *byte* são feitos respetivamente pelo “*I2C Write*” e o “*I2C Read*”. Estes blocos podem ser vistos na Figura 3-14.



Figura 3-14. Blocos da LIFA para comunicação IIC.

Se não existir eco por parte do escravo, ou o mesmo não corresponder ao valor enviado, ocorre uma falha de comunicação entre o mesmo e o programa mestre, que pode ser derivada a uma conexão com problemas ou avaria do *hardware*. Nestas circunstâncias, o caso 0 da estrutura *case*, que se encontra na Figura 3-16, será ativo, e uma janela de erro será apresentada.

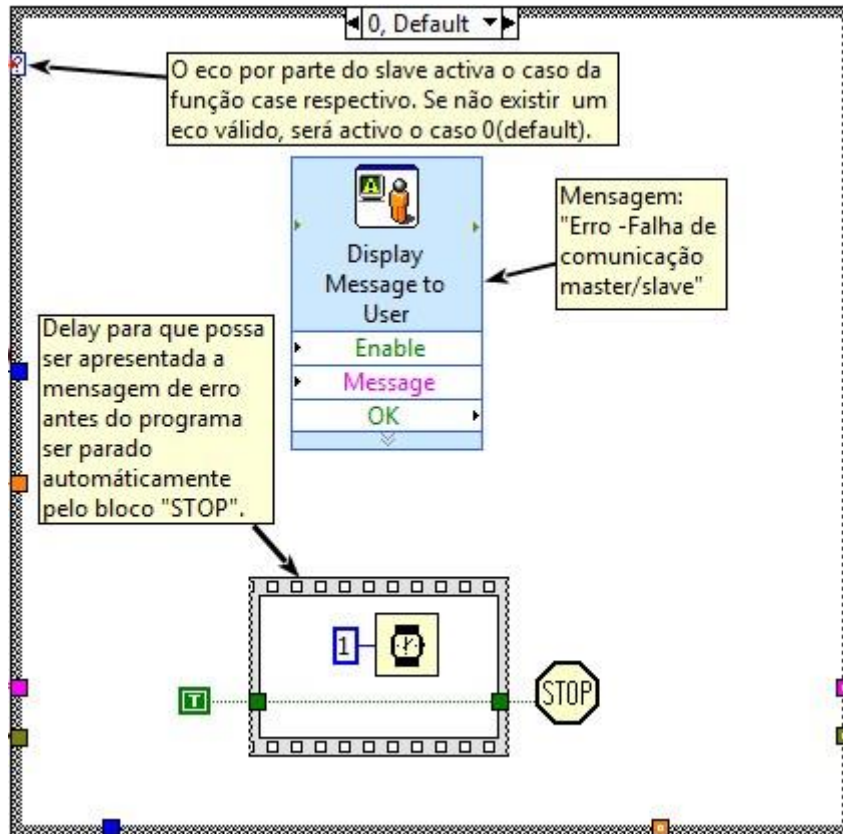


Figura 3-16. Diagrama de blocos do *case* 0.

Abaixo, na Figura 3-17, encontra-se a janela da mensagem de erro que aparece ao utilizador quando esta falha de comunicação é detetada.

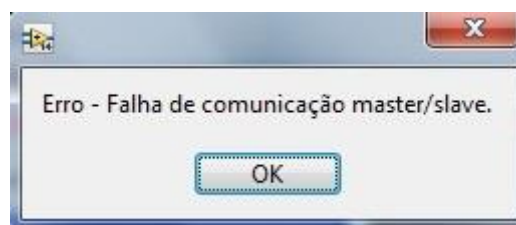


Figura 3-17. Mensagem de erro de comunicação.

Apesar do ganho do AD8556 ter resolução de 8 *bit*, o valor enviado pelo master é de 16 *bit*. Isto deve-se ao facto de estar compreendido entre os valores 70 e 1280, o que dá 1210 valores inteiros (superior a 1024). O ideal era este valor possuir 9 *bit*, mas o bloco de envio por IIC só suporta múltiplos de *byte*. A aproximação para 8 *bit* é feita por parte do processador escravo.

O valor proveniente do painel frontal é dividido em 2 *bytes*. Para que a comunicação se estabeleça sem falhas é feito um *delay* de 10 ms entre o MSB e o LSB e estes 2 *bytes* são enviados pelo bloco “I2C Write”.

O diagrama de blocos do caso 1, presente na Figura 3-18, diz respeito ao envio do valor de ganho ao processador local do sensor para que este programe o amplificador de instrumentação AD8556.

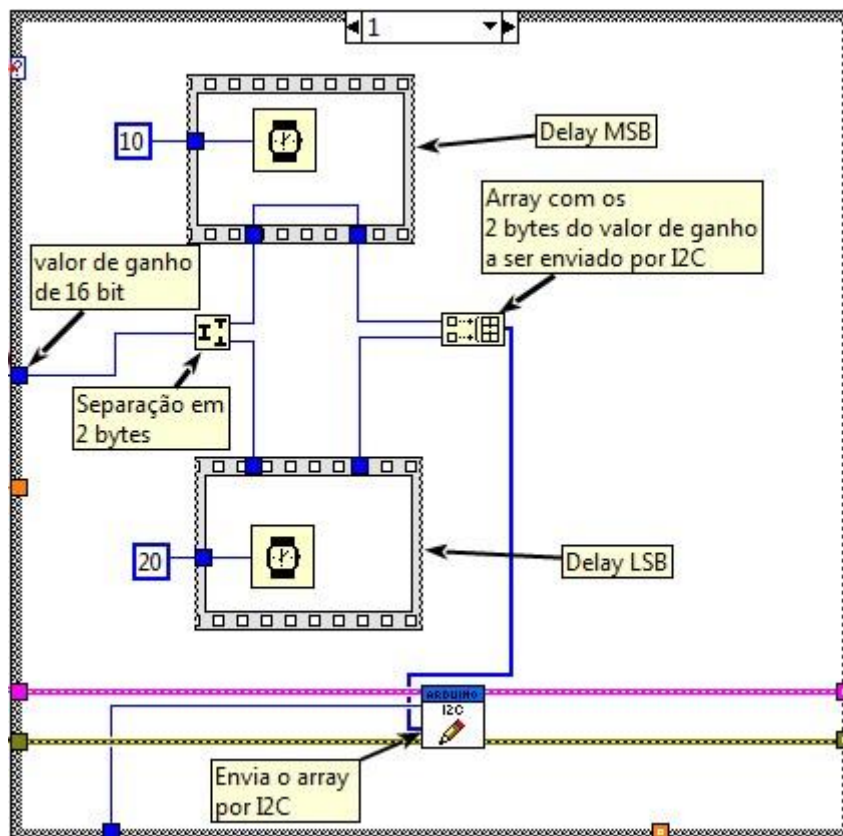


Figura 3-18. Diagrama de blocos do case 1.

A ordem para equilíbrio da ponte é dada pelo envio do valor “2” ao escravo. Após este retornar o eco, a estrutura case presente na Figura 3-19 é validada. Aqui é recebido um valor por I2C para verificar se o processo de equilíbrio da ponte extensométrica ocorreu ou não com sucesso.

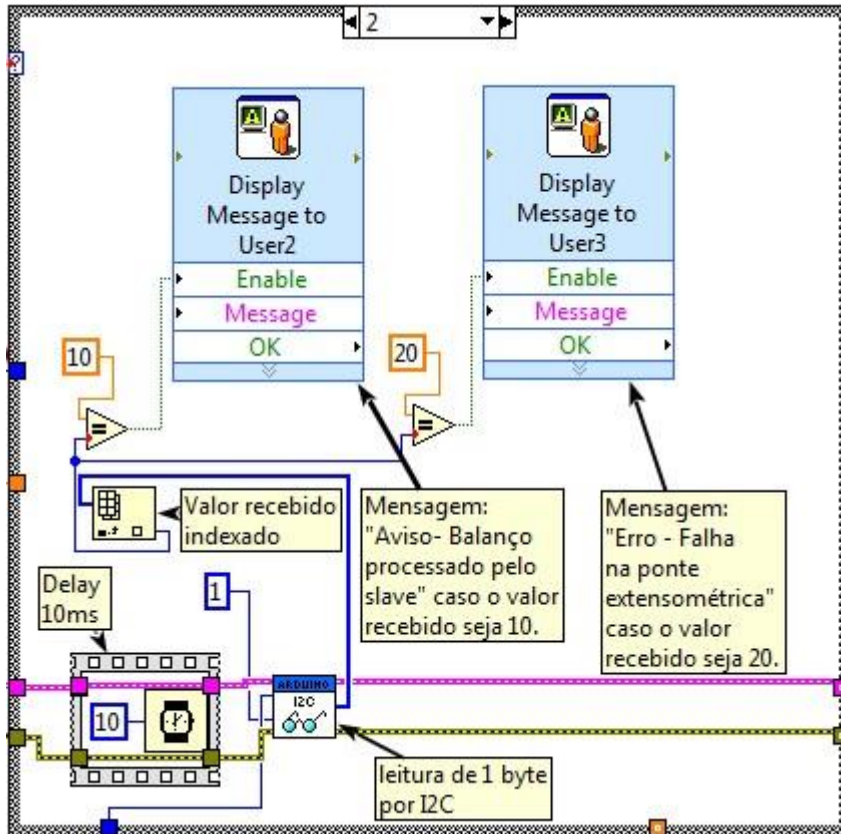


Figura 3-19. Diagrama de blocos do case 2.

O valor “10” é enviado pelo *slave* ao *software* de controlo caso o equilíbrio tenha sido possível, apresentando a mensagem da Figura 3-20.

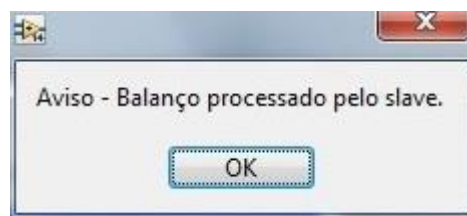


Figura 3-20. Mensagem de aviso de equilíbrio.

Caso exista algum problema que impeça o processador local de fazer o equilíbrio da ponte extensométrica, como por exemplo um fio desligado entre a mesma e a pcb, o valor “20” enviado pelo escravo, fará com que a janela, presente na Figura 3-21, seja apresentada:

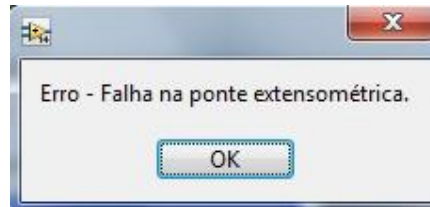


Figura 3-21. Mensagem de erro na ponte.

Através da ordem “3”, é requisitado ao escravo o valor de tensão, proporcional à força. Recebido o eco, é ativada em seguida a estrutura *case*, que se encontra na Figura 3-22. Este último caso, é aquele que o programa mestre constantemente ativa, se os botões “Enviar_ganho” e “Equilíbrio” não forem pressionados.

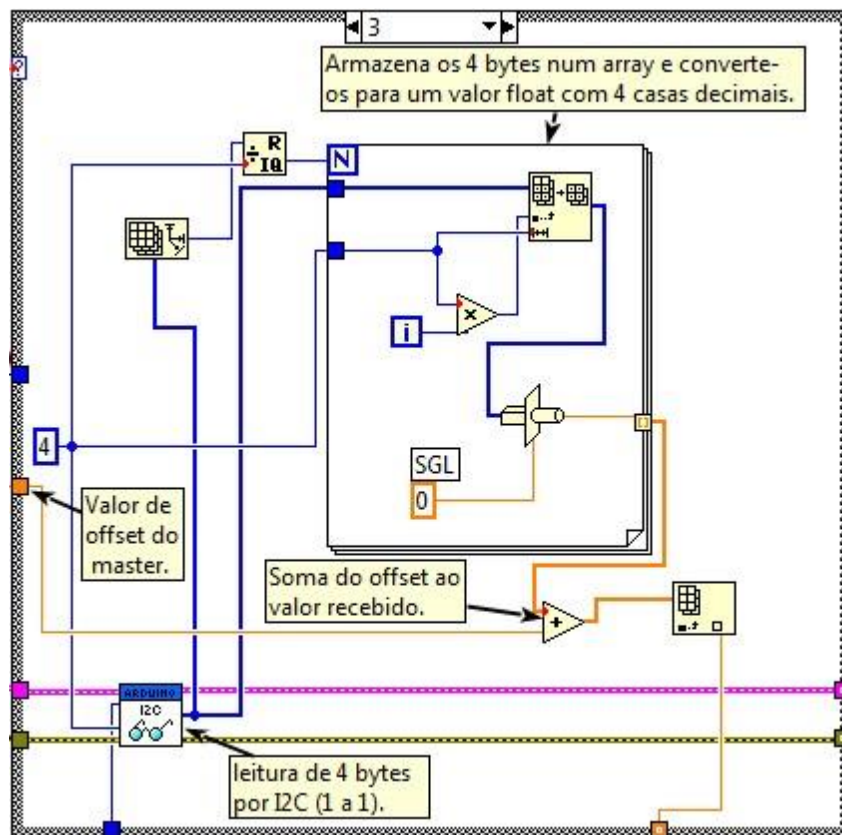


Figura 3-22. Diagrama de blocos do *case* 3.

Embora a resolução do ADC interno do ATtiny85 seja de 10 *bit*, a conversão para Volt resulta num valor com casas decimais. Como se pretende resolução na ordem dos mV, o microcontrolador local foi programado para transferir por IIC um valor de tensão do tipo *float* (32 *bit*) ao programa mestre, para que este usufrua do espaço ocupado pelo SGL, e conte com um valor de tensão com 3 casas decimais em toda a gama (0 a 5V). Posto isto, foi necessário alocar 4 *bytes* num *array*, após terem sido lidos por IIC. Um *offset* adicional é calculado aqui, a partir de uma função de soma, caso o utilizador deseje ter outro valor que não 2.5V no estado de repouso da célula.

Após o valor de carga ser adquirido e indexado num *array*, já fora da estrutura case, o código patente na Figura 3-23, converte o valor de tensão proveniente do *smart sensor*, numa unidade previamente definida, por exemplo Newton, graças à função “*Linear Fit*”.

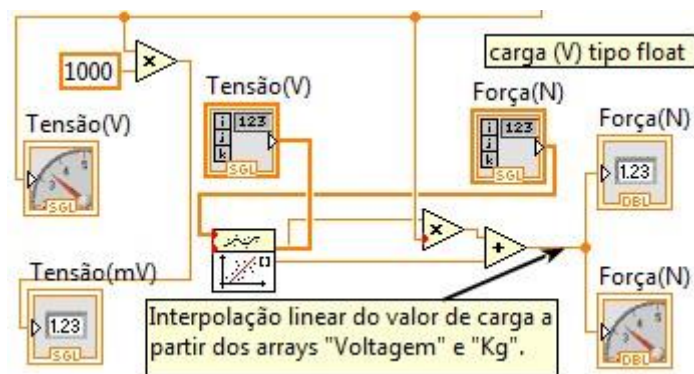


Figura 3-23. Diagrama de blocos para apresentação de valores e interpolação linear.

O programa é executado numa estrutura *while*, que será interrompida com o botão “STOP” através do painel frontal, ou caso ocorra o erro de comunicação *master/slave*. Um bloco *Wait* com 100 milissegundos de espera, dentro desta estrutura, limita a cadência de atualização do programa, a cerca de 10 vezes por segundo. É possível aumentar esta velocidade até ao dobro, porém para o efeito pretendido e com vista a poupar recursos do processador do computador onde se encontrar a correr este programa, este valor é aceitável.

Para que os processadores escravos ATTiny85 pudessem comunicar com o programa *master*, foi criado um novo programa em linguagem C, cuja versão final para este projeto se encontra no Anexo 3. O seu funcionamento é explicado resumidamente pelo fluxograma patente na Figura 3-24. A única diferença existente neste código em C para os dois escravos da célula é o endereço de cada escravo, que toma o valor 1 e 2 respetivamente para o escravo 1 e 2.

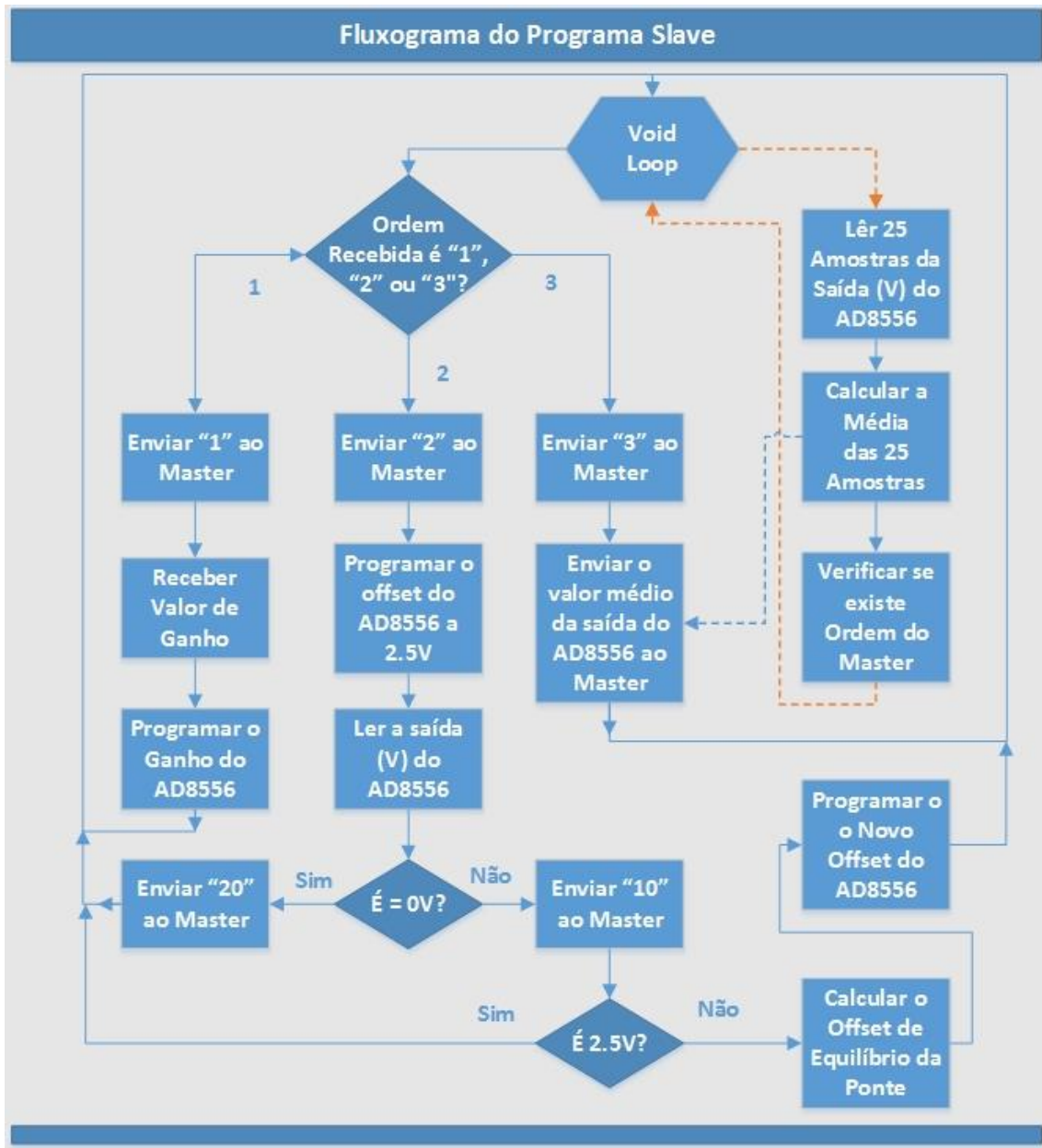


Figura 3-24. Fluxograma do *software slave*.

3.3 MONTAGEM DE UMA CÉLULA COM DOIS MOTES

Depois de se verificar o correto funcionamento do novo sistema de sensorização inteligente para uma ponte, foi montada uma célula de carga completa com as duas pontes de extensómetros, igual às usadas na plataforma referida no subcapítulo 2.4. O primeiro passo depois de adquirir a célula concebida na ESTiG, que pode ser visionada na Figura 3-25, foi lixar a mesma com lixa para metal de grão 150 seguida de grão 400, e por último grão 600 para a superfície tornar-se o mais perfeitamente plana para a colagem dos extensómetros.



Figura 3-25. Elemento metálico da célula de carga.

A colagem dos extensómetros é uma tarefa delicada e que portanto requer bastante cuidado. Cada extensómetro tem uma mira em cruz a 90 graus (entre outras referências) para facilitar a sua colagem, posto isto, guias com a mesma orientação foram desenhadas na superfície da célula destinada à colagem do extensómetro (Figura 3-26), depois de a célula ter sido limpa com acetona a fim de retirar gordura e impurezas presentes na mesma.



Figura 3-26. Marcas de referência na célula.

O extensômetro de 120Ω da RS Pro com referência 632-168 foi transferido, através de fita-cola, da sua respectiva embalagem para a célula, após ter sido aplicada uma gota de supercola 3 no centro da superfície. Depois de pressionar por alguns minutos o extensômetro a fim de promover a sua boa adesão à superfície da célula, foi removida a fita-cola em 45 graus com a superfície para que este processo não arrancasse o extensômetro.

Adicionalmente foi colada também uma tira de ligação, que facilita a ligação dos fios aos extensômetro. O processo foi repetido para os restantes sete extensômetros e o resultado pode ser visto na Figura 3-27:



Figura 3-27. Vista da célula após colagem dos extensómetros.

Com a célula de carga pronta a receber as pcbs, foi desenhado o *layout* das mesmas a partir do circuito desenvolvido anteriormente no subcapítulo 3.1, a fim de serem colocadas na própria célula e o sensor inteligente ter praticamente as suas dimensões limitadas à estrutura metálica. O *layout* desenhado no *software* Ultiboard, apresenta-se na Figura 3-28 e é composto pela parte de cima e de baixo respetivamente na figura da esquerda e da direita. Foi destinado ao fabrico das novas placas digitais de condicionamento de sinal das células de carga, que funcionarão como dispositivo escravo (condicionamento de sinal e comunicação IIC).

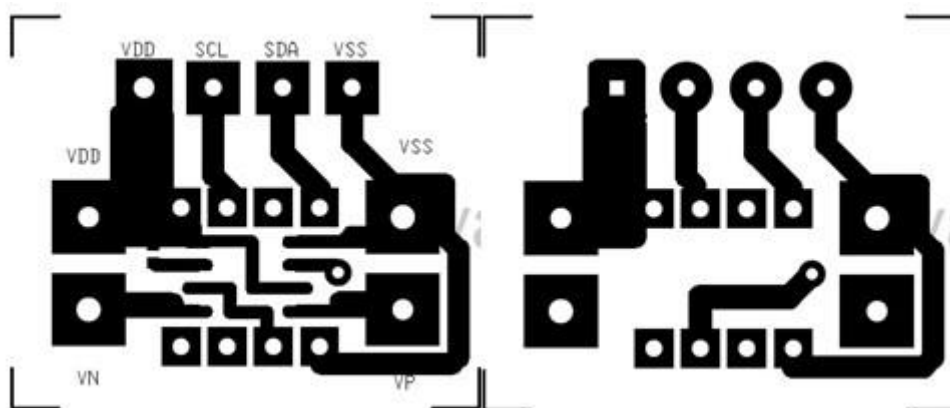


Figura 3-28. *Layout* da nova placa.

As pcbs da Figura 3-29 foram fabricadas na ESTIG, obedecendo ao presente *layout*. O resultado foi bastante bom em termos de área (2 x 2.5 cm) e robustez da placa, visto os componentes estarem localizados em apenas uma das faces, podendo a outra face ficar sujeita a contacto mecânico com a célula, apenas com o cuidado de ser colocado um isolante elétrico no meio, por exemplo um suporte ou cola.

Os dois circuitos integrados desta pcb ficam sobrepostos onde o *socket* do ATTiny85 serve de espaçador entre o mesmo e o AD8556, sendo esta a chave para a otimização de espaço. Nesta figura, encontra-se uma placa sem os componentes, e outra com os componentes já soldados. O conector verde permite que o circuito seja alimentado, e dispõe das linhas de I2C. Apesar de não ser muito favorável em termos de dimensões, era o que melhor se enquadrava dentro do disponível na ESTIG para esta aplicação.

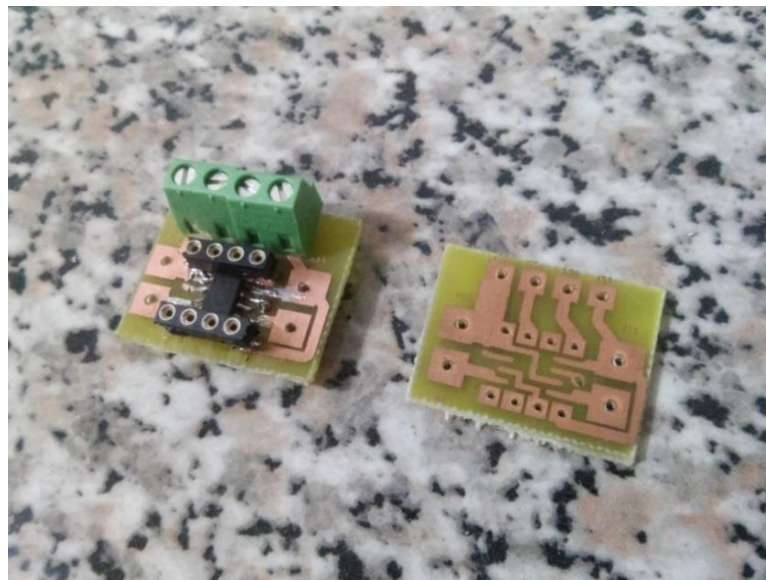


Figura 3-29. Placas digitais de condicionamento de sinal.

Para efeitos de reprogramação futura do microprocessador, este pode ser removido do seu *socket*. Outra ideia seria colocar uma versão SMD do ATTiny85, mas nesse caso seria preciso deixar um conector para a sua reprogramação, o qual iria ocupar espaço, e neste caso os dois integrados já não ficariam sobrepostos, sendo no geral uma pior solução. A Figura 3-30 mostra a relação de tamanho entre a célula de carga e uma das pcbs.

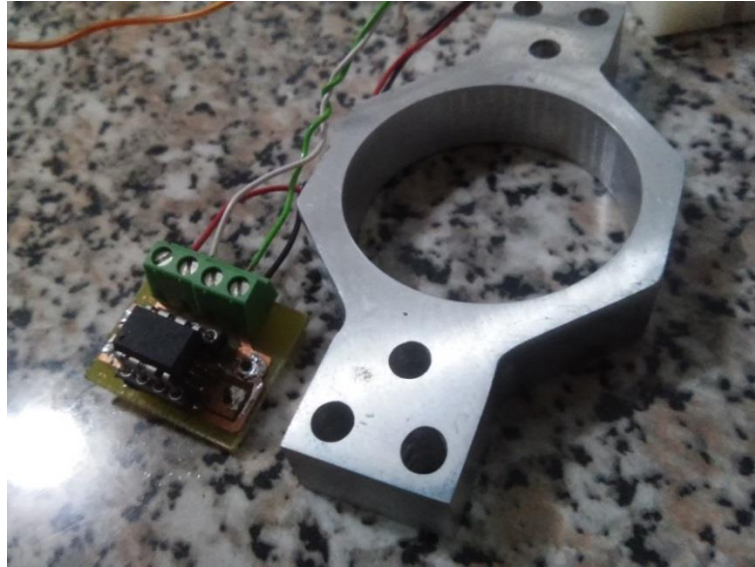


Figura 3-30. Comparação entre os tamanhos da célula e uma das placas.

Depois de comprovado o bom funcionamento das placas, e não existindo ainda um suporte na célula de carga onde pudessem ser alojadas, foram coladas com cola termo-fusível à célula, dentro da sua abertura circular central. As ligações elétricas foram feitas entre a placa (Figura 3-31) e os extensómetros por meio de fios de cobre soldados aos terminais, de forma a respeitar o circuito em ponte de *Wheatstone*.

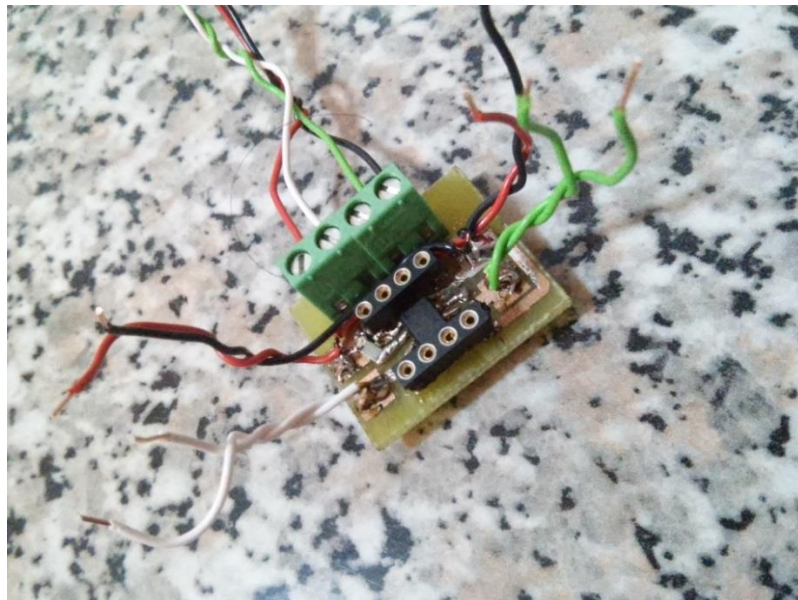


Figura 3-31. Uma placa pronta a ser montada na célula.

O resultado final pode ser visto na seguinte Figura 3-32, onde a célula se encontra com as duas pcbs montadas e já pode receber o nome de *smart sensor*.

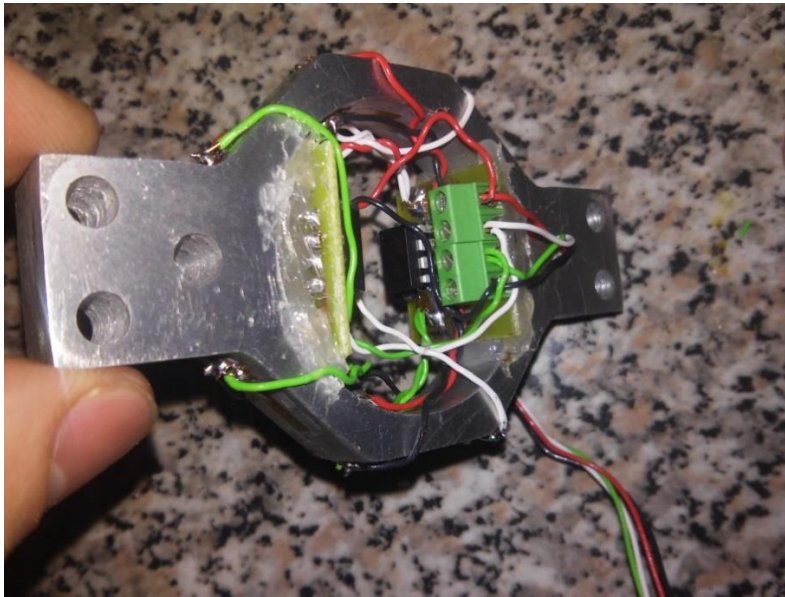


Figura 3-32. *Hardware* montado na célula, formando o *smart sensor*.

3.4 ANÁLISE DE RESULTADOS

Por último, estabeleceu-se a ligação por IIC do *smart sensor* ao Arduino, como ilustra a Figura 3-33, e foram verificados os resultados práticos.

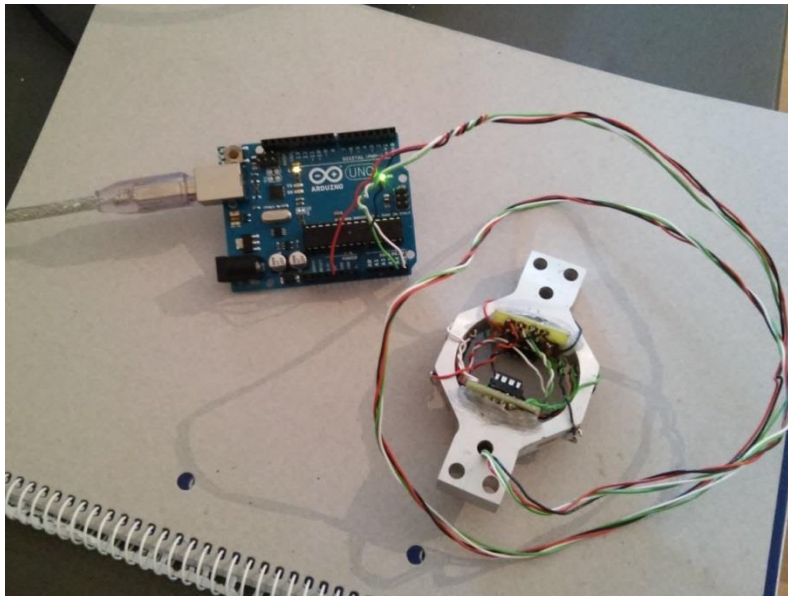


Figura 3-33. Ligação do *hardware* final.

A comunicação verificou-se sem problemas durante as horas de funcionamento em que o sistema foi testado. O visionamento das janelas de erro também ocorreu da forma esperada, mediante as condições que por defeito as ativam.

O primeiro teste que foi feito através do painel frontal, consistiu em equilibrar a ponte de extensómetros de cada *slave*, com diversos valores de ganho, uma vez que sempre que este é alterado, é necessário proceder a um novo ajuste de 2.5V. A Figura 3-34, mostra o valor de tensão no separador do escravo 1, após carregar no botão “Equilíbrio”, para o valor de ganho de x900.

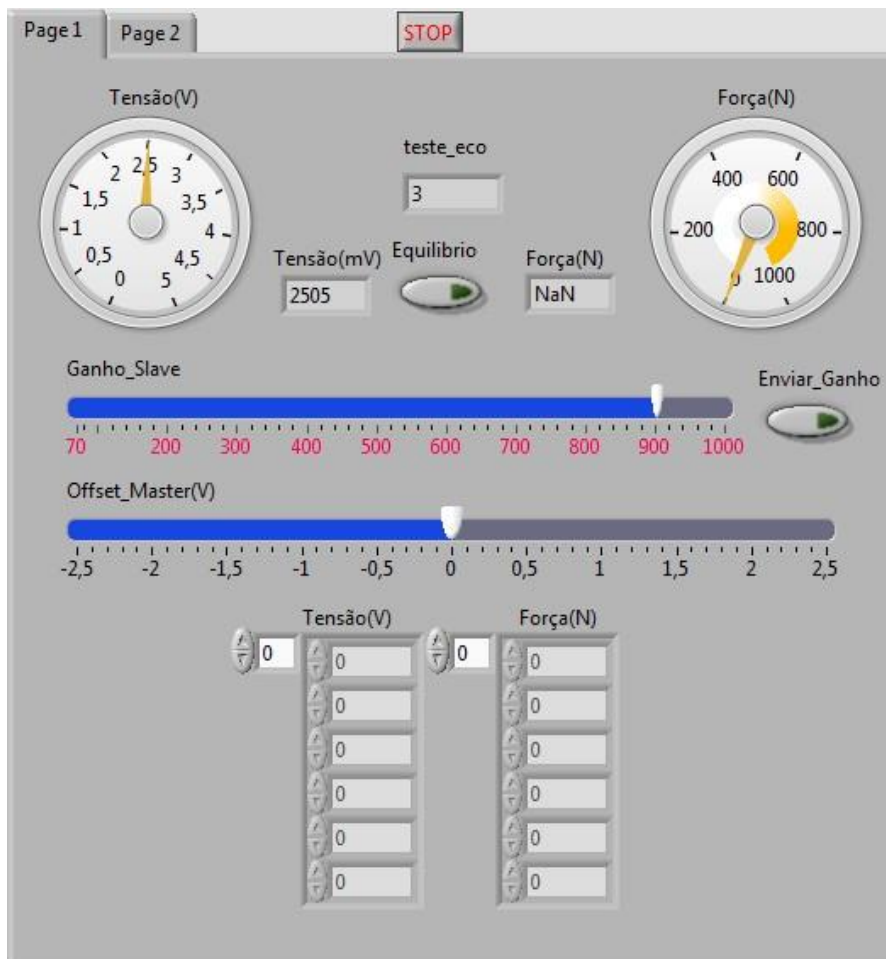


Figura 3-34. Equilíbrio da ponte com ganho de x900.

No seguimento dos testes, e dado que o ajuste de 2,5V foi processado com êxito, apenas apresentando um erro máximo de 5 mV para os diferentes níveis de ganho, foi realizada uma calibração da célula de carga dentro da sua gama de funcionamento (0N-300N), usando um equipamento de compressão (edómetro) da MaTest, referência S265. Este procedimento consistiu em colocar a célula sob uma força controlada segundo Y (Figura 3-35), onde se realizaram 6 ensaios com 6 valores de carga diferentes, totalizando 36 valores de tensão.



Figura 3-35. Calibração da célula.

A força de compressão foi imposta na célula por meio de pesos no prato do equipamento, considerando o fator de conversão (x10). Para um controlo minucioso da força, os pesos (Figura 3-36) foram previamente pesados numa balança de precisão da marca Precisa, modelo 4000C, a fim de se obter o peso real.



Figura 3-36. Pesos usados na calibração da célula.

O valor de ganho usado na calibração foi de $\times 900$, visto ser o que permitia ter maior gama de medida, sem saturar os indicadores de tensão do programa. As seguintes figuras mostram o *front panel* para os valores de tensão obtidos no primeiro ensaio, após ajuste de 2.5V (0 N).

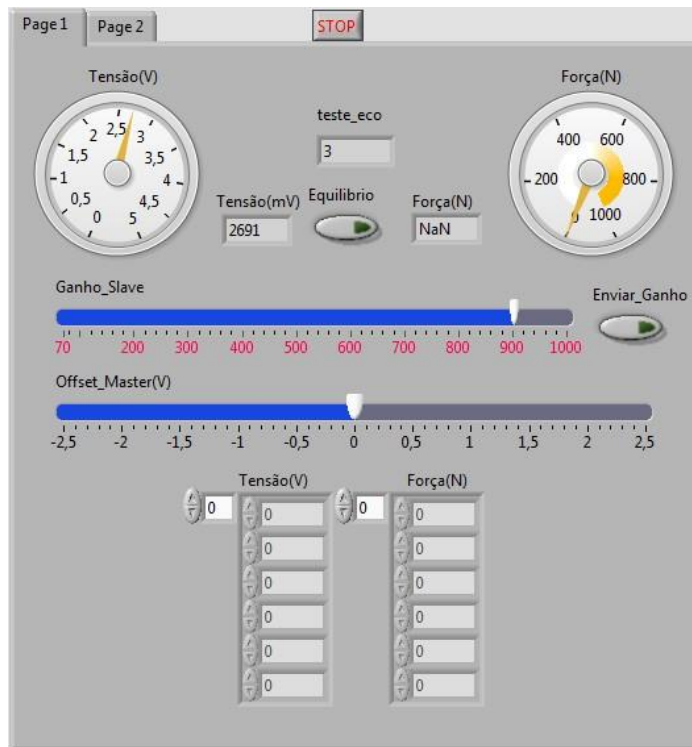


Figura 3-37. Painel frontal. Tensão medida para 24,615 N.

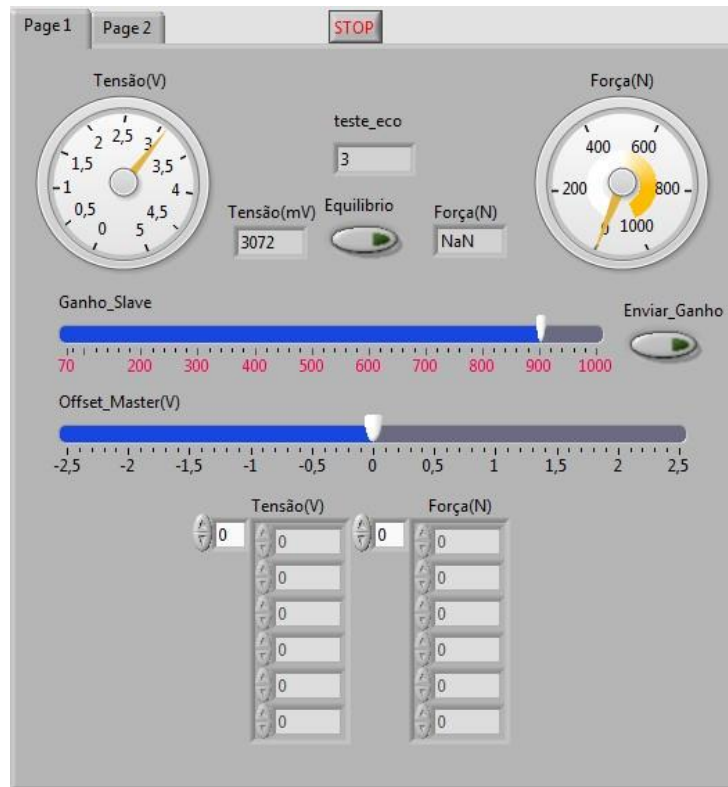


Figura 3-38. Painel frontal. Tensão medida para 74,04 N.

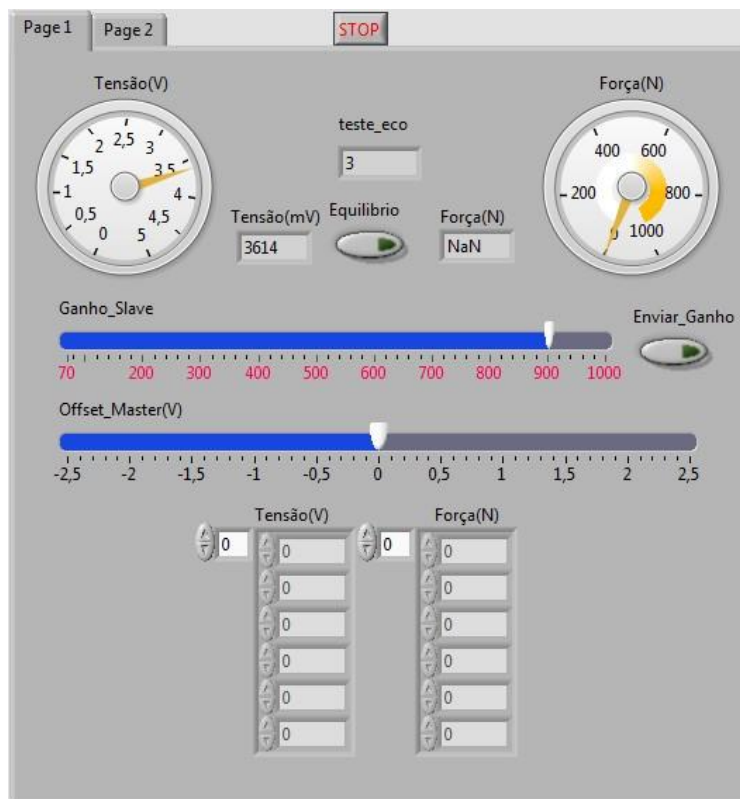


Figura 3-39. Painel frontal. Tensão medida para 143,765 N.

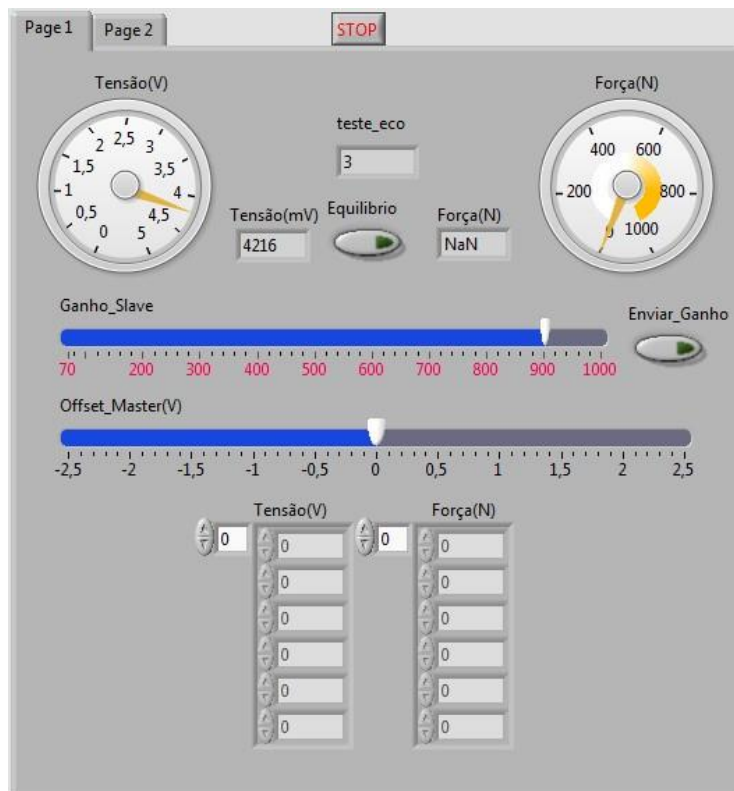


Figura 3-40. Painel frontal. Tensão medida para 221,042 N.

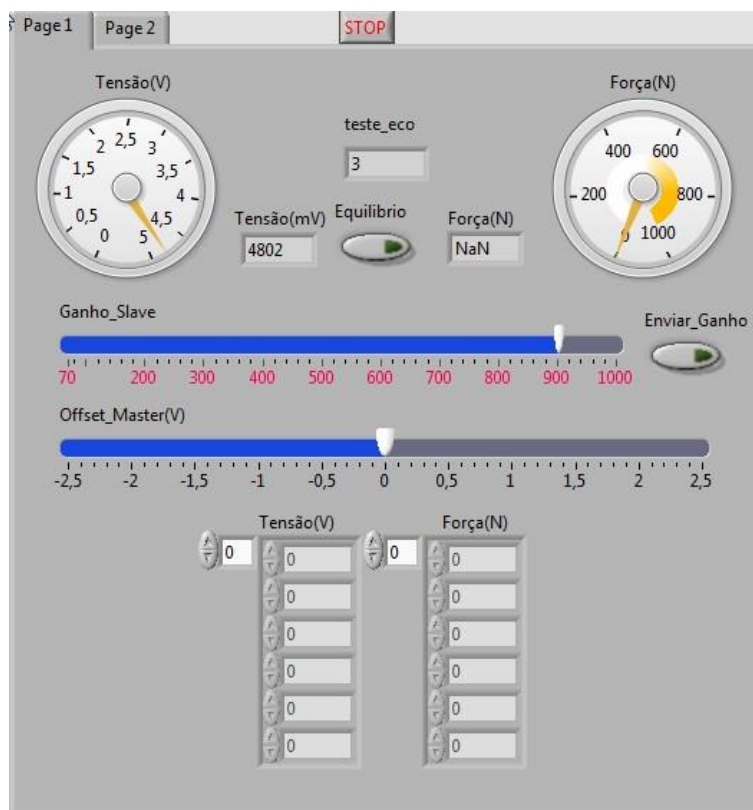


Figura 3-41. Painel frontal. Tensão medida para 296,553 N.

Em função da força exercida (em Newton), foram obtidos os 36 valores de tensão. Para analisar melhor os resultados, estes foram decrementados para 0, a partir da diferença destes e do valor inicial de cada ensaio (carga nula), de onde resultou a tabela 5. É de notar que a carga máxima de calibração não foi de 300 N efetivos, devido à inexistência de uma massa com esse peso, assim, optou-se por um valor de calibração próximo do valor máximo (296.55 N).

Tabela 5. Valores de calibração.

Força (N)	0	24,615	74,04	143,765	221,042	296,553
Ensaio 1 (mV)	2505	2691	3072	3614	4216	4802
Decremento 1 (mV)	0	186	567	1109	1711	2297
Ensaio 2 (mV)	2505	2691	3077	3614	4220	4807
Decremento 2 (mV)	0	186	572	1109	1715	2302
Ensaio 3 (mV)	2505	2696	3077	3614	4216	4802
Decremento 3 (mV)	0	191	572	1109	1711	2297
Ensaio 4 (mV)	2500	2691	3072	3614	4216	4802
Decremento 4 (mV)	0	191	572	1114	1716	2302
Ensaio 5 (mV)	2500	2691	3072	3614	4216	4802
Decremento 5 (mV)	0	191	572	1114	1716	2302
Ensaio 6 (mV)	2505	2696	3077	3614	4216	4802
Decremento 6 (mV)	0	191	572	1109	1711	2297

Recorrendo ao *software* Excel, através de regressão linear, obteve-se um gráfico (Figura 3-44) para entender se os 36 valores de calibração descreviam ou não uma reta, e o resultado foi excelente, obtendo-se um coeficiente de determinação (R²) unitário a que corresponde um excelente ajuste da curva aos valores medidos experimentalmente.

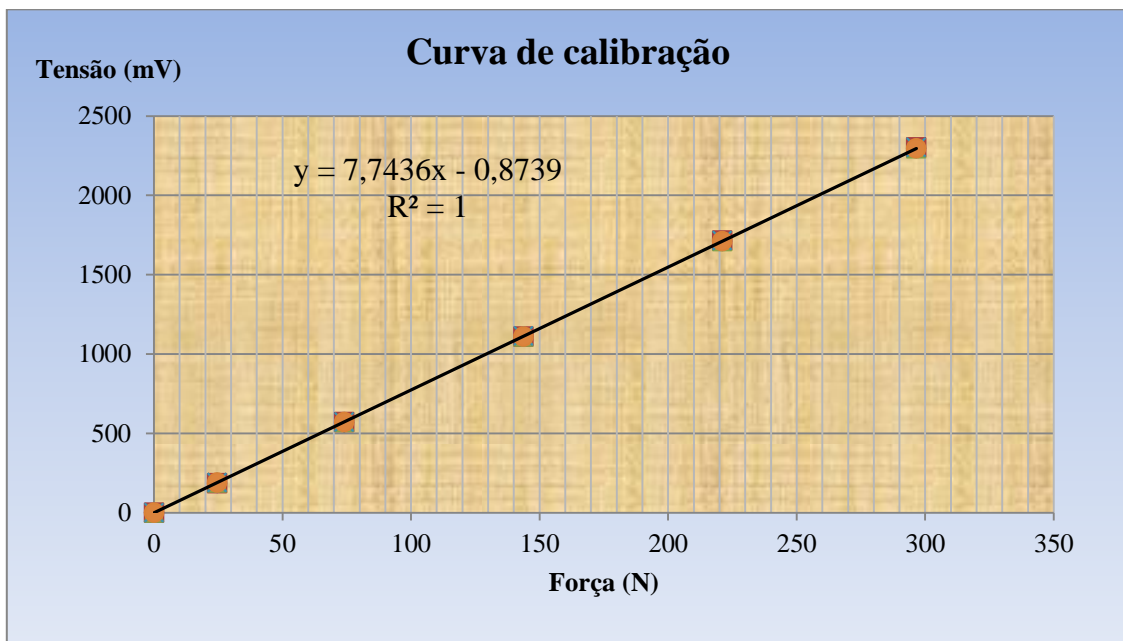


Figura 3-42. Curva de calibração.

Feita a média dos valores de tensão com vista a calibrar a célula de carga, estes foram introduzidos no painel frontal do programa, na tabela destinada a esse propósito, e vários resultados foram recolhidos. Recorrendo às várias massas existentes no laboratório, foi exercida uma força na célula com valores o mais afastados possível dos de calibração.

As seguintes figuras mostram o painel frontal do programa de controlo durante este processo, onde os indicadores de força entram em funcionamento, indicando o valor interpolado pelo programa mestre.

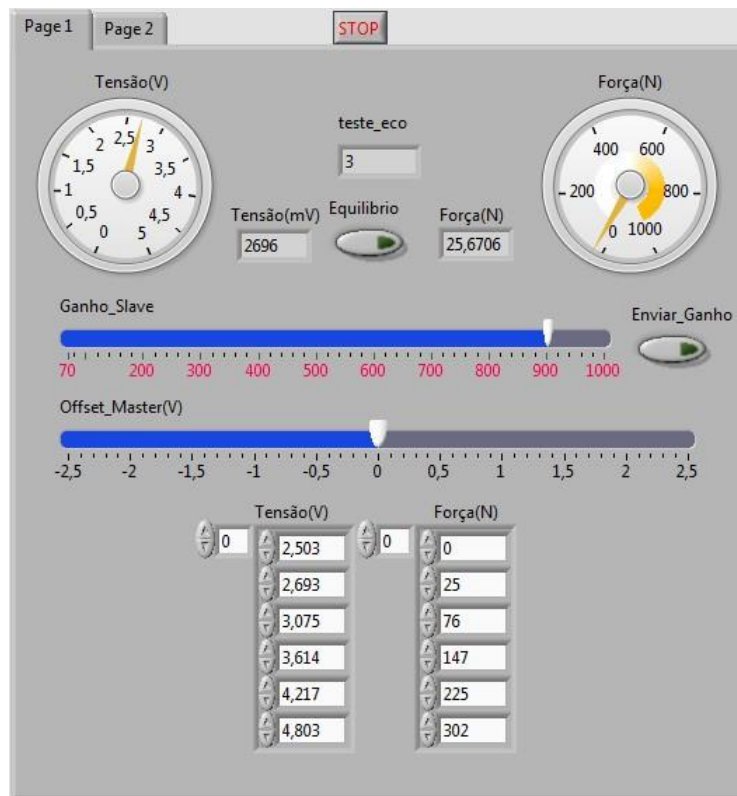


Figura 3-43. Painel Frontal. Valor interpolado para 25,5 N.

Nota: Aqui os valores inseridos no array “Força(N)” foram convertidos de Kgf para N multiplicando por 10 ao invés de 9.8, daí diferirem da tabela anterior, o que não interfere na análise dos valores interpolados visto que a força aplicada neste procedimento, presente na legenda das figuras, também sofreu o mesmo tipo de conversão.

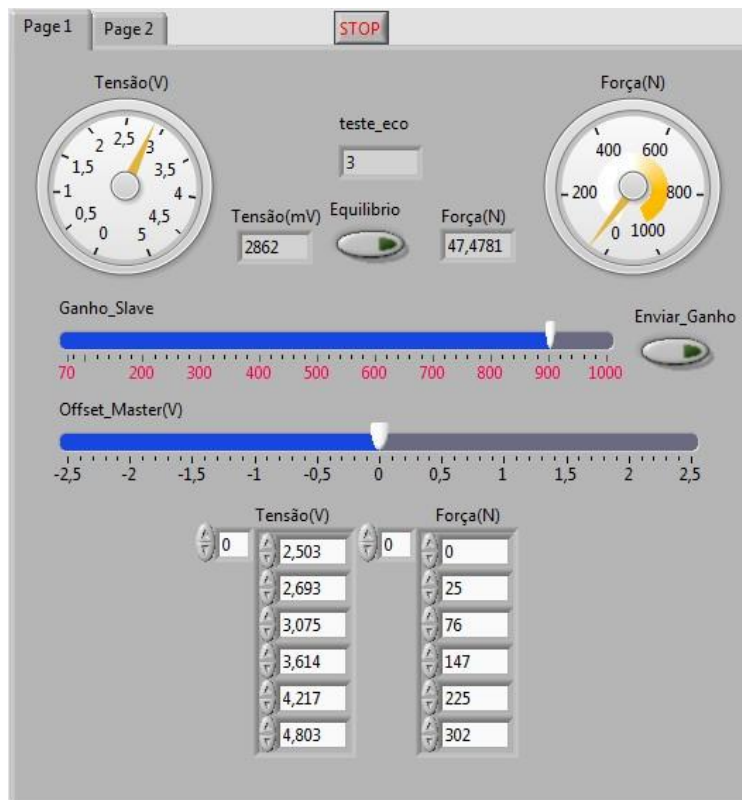


Figura 3-44. Painel Frontal. Valor interpolado para 47,6N.

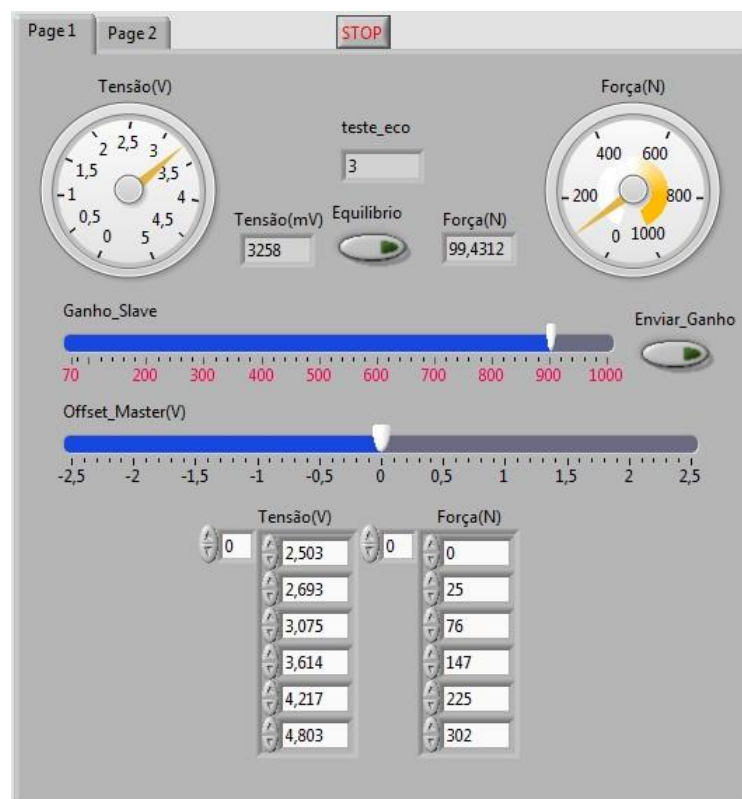


Figura 3-45. Painel Frontal. Valor interpolado para 99N.

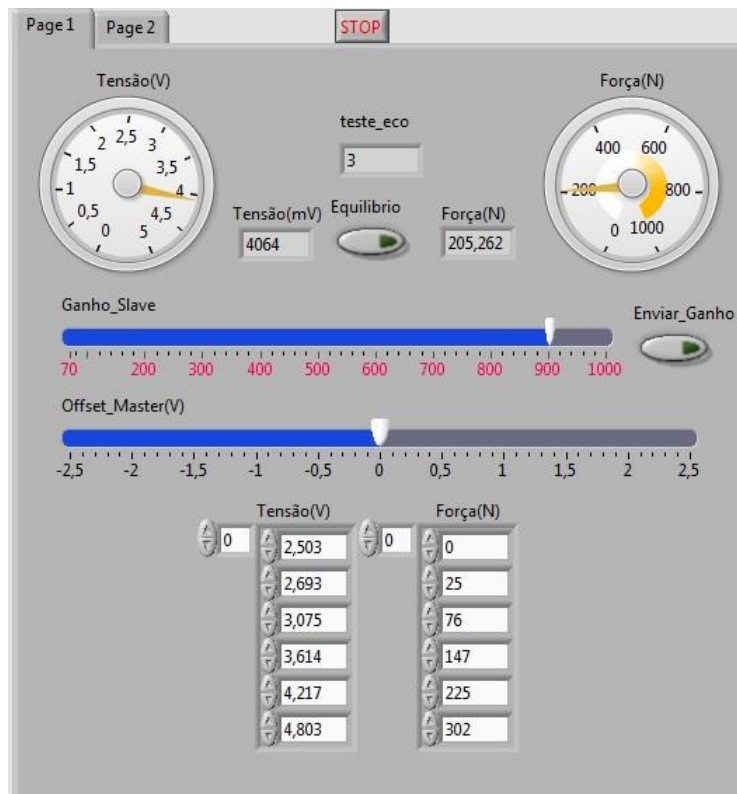


Figura 3-46. Painel Frontal. Valor interpolado para 203,6N.

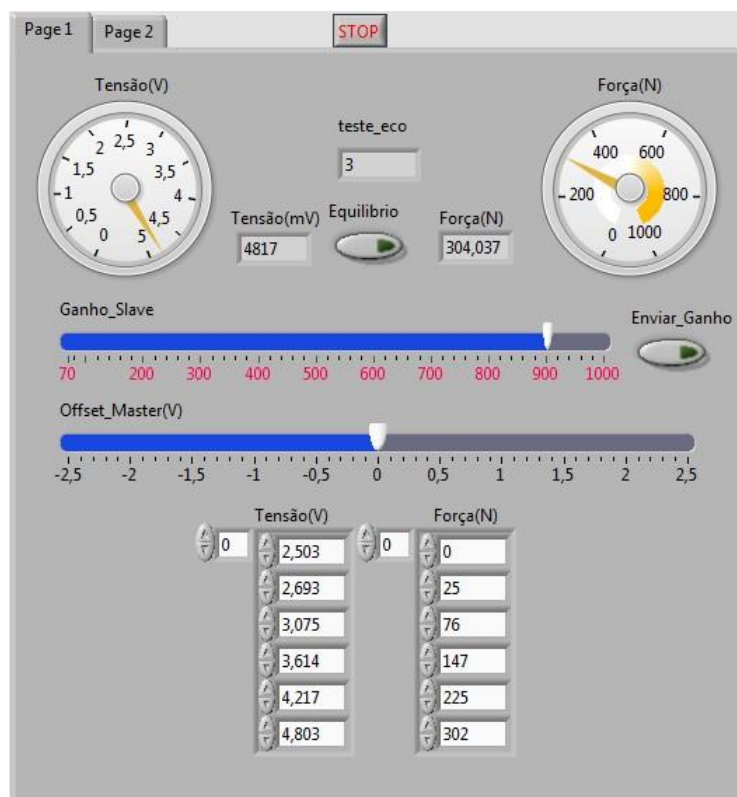


Figura 3-47. Painel Frontal. Valor interpolado para 302,5N.

O bom funcionamento do botão de “Offset_Master”, apesar de não ter uma função muito importante no programa, também foi verificado ao longo dos testes. A estabilidade geral deste sistema é bastante aceitável, assim como a sua precisão.

4. CONCLUSÃO E SUGESTÕES PARA TRABALHO FUTURO

O objetivo principal deste trabalho, consistiu na elaboração de um circuito eletrónico de reduzidas dimensões que fizesse o condicionamento de sinal para células de carga, tornando-as num *smart sensor*. Pretendeu-se, também, que o circuito eletrónico lidasse com um dispositivo de controlo e leitura por IIC, constituindo assim um sistema do tipo mestre/escravo. Estes objetivos foram atingidos e ultrapassados, na medida em que foi ainda desenvolvido um programa de controlo e registo. O *software* de desenvolvimento eleito foi LabVIEW, aliado ao Arduíno como suporte de *hardware* com a funcionalidade de mestre, de modo a gerir os vários dispositivos escravos.

A nível de objetivos específicos, este dispositivo permite efetuar o condicionamento de sinal, digitalização e transmissão de dados para células de carga. As maiores vantagens no que diz respeito ao trabalho desenvolvido, consiste na facilidade de ajuste de ganho e equilíbrio da ponte de extensómetros, através de *offset*, para as várias células de carga que constituem a plataforma de forças, assim como a recolha dos níveis de carga de uma forma centralizada. A visualização do valor de força numa determinada unidade é possível, através de interpolação linear. Além disso, de modo a evitar interferências de origem eletromagnética, o sistema é filtrado contra o ruído.

De modo a verificar a capacidade adaptativa do sistema, uma célula de carga com características diferentes (estrutura metálica da célula e resistência dos extensómetros) à utilizada por defeito ao longo do trabalho foi testada, e o sistema provou adaptar-se à mesma.

A dimensão da placa mostrou-se ser favorável a que ficasse no interior da célula, e comparando este sistema com os comercializados, referidos no estado da arte, este sistema é dotado de características equivalentes, e no que toca a velocidade de atualização e limite máximo de dispositivos, conseguiu supera-los. Graças à comunicação por IIC, é possível ter um endereço de *7-bit*, permitindo um número máximo de 128 *smart sensors*.

Apesar da resolução deste sensor inteligente ser apenas de 10 *bit*, este não pode ser diretamente comparado com a resolução dos dispositivos comercializados (18 *bit*). Estes possuem um ADC que mede diretamente a saída da ponte, sem um amplificador

de instrumentação, requerendo uma resolução muito superior. O valor de tensão medido pelo ADC interno do ATTiny85 garante uma resolução de 4,88 mV, em toda a gama de funcionamento (0 a 5000 mV).

O processo de calibração da célula permitiu verificar o comportamento linear da tensão medida por este sistema em função da carga aplicada, e também a sua estabilidade de medição, apresentando apenas uma oscilação de cerca de 5 mV, o que é bastante aceitável neste tipo de equipamento. Verificou-se também um excelente ajuste da curva de calibração às medições experimentais, tendo-se obtido um coeficiente de determinação (R^2) de 1.

Os valores de força interpolados pelo programa foram apresentados com sucesso, existindo apenas um pequeno desvio. Este sensor inteligente para medição de cargas mecânicas por IIC mostrou ser uma boa alternativa aos equipamentos existentes, devendo este ser estudado e melhorado futuramente. Algumas melhorias sugeridas são:

A nível de *hardware*:

- O conector das pcbs do *smart sensor* não é o mais indicado podendo este ser substituído por outro mais compacto e fácil de ligar.
- Um suporte deve ser concebido de modo a montar as pcbs na célula de uma forma mais prática e segura, com alguma flexibilidade dado a deformação que existe quando a célula é submetida a carga.
- Os cabos de ligação entre mestre/escravos deverão ser flexíveis e com algum blindagem contra EMI.

A nível de *software*:

- Atendendo ao funcionamento da plataforma de forças, uma importante melhoria seria a interface dispor dos valores de carga ao mesmo tempo e na mesma janela de trabalho, usando para isso, por exemplo, uma estrutura *for* que varresse os vários endereços dos escravos que possam coexistir no sistema.

- Algumas operações matemáticas poderão ser adicionadas ao programa em LabVIEW para efeitos de média das pontes de *wheatstone* que estarão a sensorizar forças no mesmo eixo.
- Os tipos de variáveis poderão ser melhor estudados com vista a otimizar o programa de controlo e o programa escravo.

REFERÊNCIAS

- [1] Chatterjee et al, Weighing System by Load Cell Response Rectification Method, International Journal of Measurement Technologies and Instrumentation Engineering, 2012.
- [2] Prabuwno e Akbar, PC Based Weight Scale System with Load Cell for Product Inspection, Faculty of Information Technology Budi Luhur University, 2009.
- [3] Mantracourt, 2016.
- [4] Dillon et al, Toledo Scale Corporation, 1989 (Patente US 4815547 A).
- [5] William, Crystal Gage Inc, 1989 (Patente US4838369 A).
- [6] Grimm e Loshbough, Mettler-Toledo Inc, 1997 (Patente US 5623128 A).
- [7] Kolaci, 1998 (Patente US5841077 A).
- [8] Frye e Huitt, 2005 (Patente US 6919516 B2).
- [9] Bae et al, Analysis of Digital Load Cell Using 2.4GHz Band's Zig-Bee, Pukyong National University, 2008.
- [10] Gilbertson et al, Improvement of Accuracy in a High-Capacity, Six Degree-of-freedom Load Cell, University of Pittsburgh, 1999.
- [11] Shusaku Horibe, Robert Hooke, Hooke's Law & the Watch Spring.
- [12] HP Ferreira, Force Measurement, NRCS Legal Metrology Division, 2009.
- [13] João Vitor, Piezoeletricidade, 2015.
- [14] Jayant Sirohi e Inderjit Chopra, Fundamental Understanding of Piezoelectric Strain Sensor, University of Maryland, 2000.
- [15] A. L. Window, Strain Gauge Technology, Measurements Group UK Ltd, Basingstroke, Hampshire, UK, 1992.
- [16] Earl D. Gates, Introduction to Basic Electricity and Electronics Technology, 2013.
- [17] National Instruments, 2016.
- [18] Eric Coates, Temperature Effects on Resistance, 2015.
- [19] Texas Instruments, Signal Conditioning Wheatstone Resistive Bridge Sensors, 1999.
- [20] Strain Gage Sensors, Department of Mechanical and Environmental Engineering University of California, Santa Barbara, 2003.

- [21] Patrícia Moreira e Vanessa Rocha, Conceção, desenvolvimento e construção de uma plataforma de força, Relatório final de projeto, Escola Superior de Tecnologia e Gestão, Bragança, 2014.
- [22] Atmel Corporation, 2016.
- [23] Programming an ATtiny w/ Arduino 1.6 (or 1.0), highlowtech. URL: <http://highlowtech.org/?p=1695>
- [24] Leens, An introduction to I2C and SPI protocols, 2009.
- [25] Plataforma Arduino, 2016.
- [26] Plataforma Arduino, 2016. URL: <http://playground.arduino.cc/Code/USIi2c>.
- [27] National Instruments, 2016. URL: <http://digital.ni.com/public.nsf/allkb/A20FBBD36820669086257886004D5F4D>.
- [28] National Instruments, 2016. URL: <http://www.ni.com/newsletter/51141/pt/>.

ANEXO 1

AD8556

GAIN VALUES

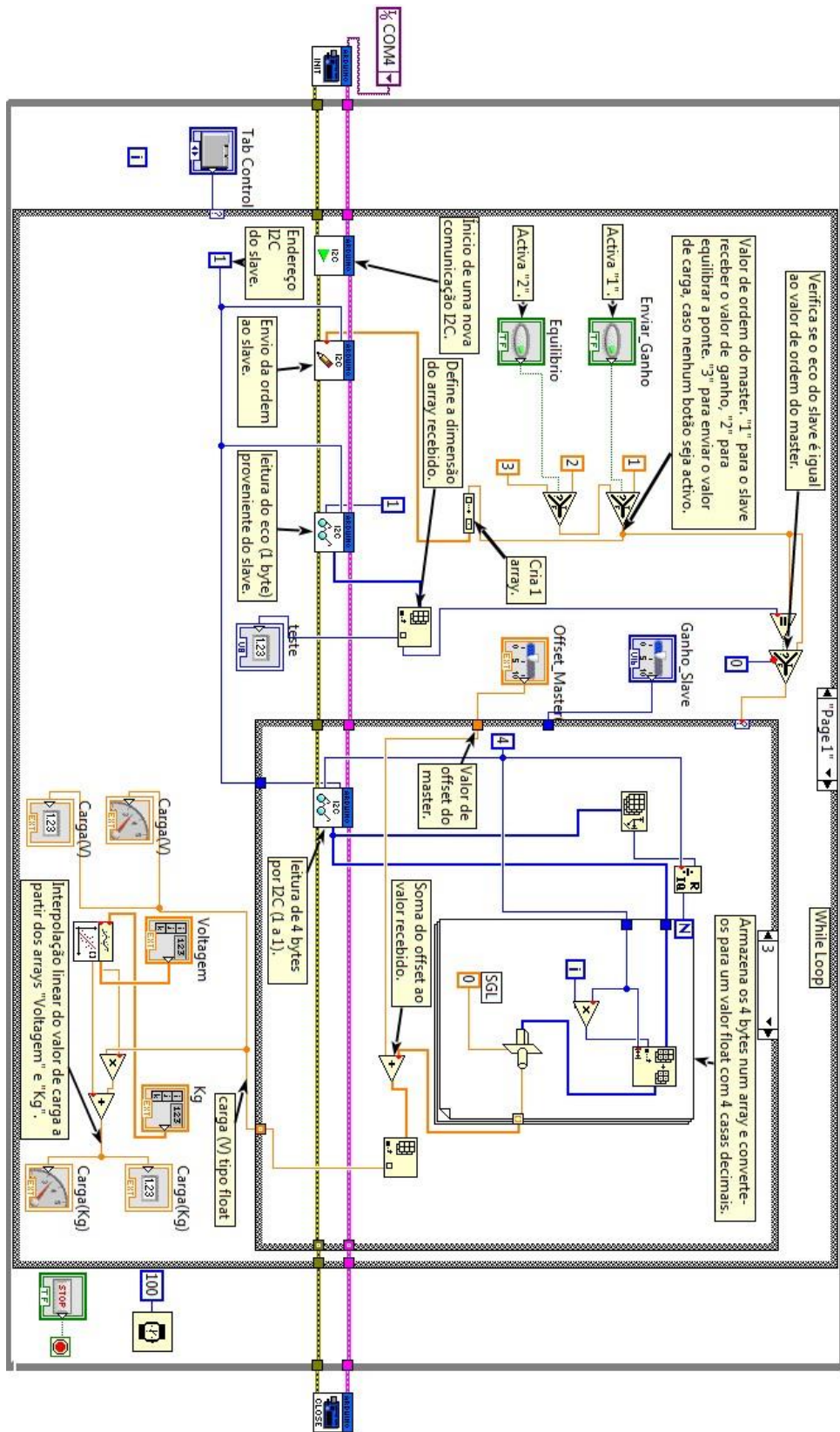
Table 5. First Stage Gain vs. First Stage Gain Code

First Stage Gain Code	First Stage Gain	First Stage Gain Code	First Stage Gain	First Stage Gain Code	First Stage Gain	First Stage Gain Code	First Stage Gain
0	4.000	32	4.503	64	5.069	96	5.706
1	4.015	33	4.520	65	5.088	97	5.727
2	4.030	34	4.536	66	5.107	98	5.749
3	4.045	35	4.553	67	5.126	99	5.770
4	4.060	36	4.570	68	5.145	100	5.791
5	4.075	37	4.587	69	5.164	101	5.813
6	4.090	38	4.604	70	5.183	102	5.834
7	4.105	39	4.621	71	5.202	103	5.856
8	4.120	40	4.638	72	5.221	104	5.878
9	4.135	41	4.655	73	5.241	105	5.900
10	4.151	42	4.673	74	5.260	106	5.921
11	4.166	43	4.690	75	5.280	107	5.943
12	4.182	44	4.707	76	5.299	108	5.965
13	4.197	45	4.725	77	5.319	109	5.988
14	4.213	46	4.742	78	5.339	110	6.010
15	4.228	47	4.760	79	5.358	111	6.032
16	4.244	48	4.778	80	5.378	112	6.054
17	4.260	49	4.795	81	5.398	113	6.077
18	4.276	50	4.813	82	5.418	114	6.099
19	4.291	51	4.831	83	5.438	115	6.122
20	4.307	52	4.849	84	5.458	116	6.145
21	4.323	53	4.867	85	5.479	117	6.167
22	4.339	54	4.885	86	5.499	118	6.190
23	4.355	55	4.903	87	5.519	119	6.213
24	4.372	56	4.921	88	5.540	120	6.236
25	4.388	57	4.939	89	5.560	121	6.259
26	4.404	58	4.958	90	5.581	122	6.283
27	4.420	59	4.976	91	5.602	123	6.306
28	4.437	60	4.995	92	5.622	124	6.329
29	4.453	61	5.013	93	5.643	125	6.353
30	4.470	62	5.032	94	5.664	126	6.376
31	4.486	63	5.050	95	5.685	127	6.400

Table 6. Second Stage Gain and Gain Ranges vs. Second Stage Gain Code

Second Stage Gain Code	Second Stage Gain	Minimum Combined Gain	Maximum Combined Gain
0	17.5	70	112
1	25	100	160
2	35	140	224
3	50	200	320
4	70	280	448
5	100	400	640
6	140	560	896
7	200	800	1280

ANEXO 2



ANEXO 3

```
#include <TinyWireS.h>

int outPin = 4; //saída para o AD8556
(digin)
int inPin = 3; //saída do AD8556 (valor
carga)

float voltage;
int ADValue;

float value_AD = 0;
int value_AD2 = 0;
float value_out = 0;
float erro = 0;

int value = 0;
int Param = 0;
#define Similar_COD 0b01
#define SSG_COD 0b00
#define FSG_COD 0b01
#define OFS_COD 0b10
#define Dummy_COD 0b00000010

const int numReadings = 25;
int readings[numReadings]; // the
readings from the analog input
int readIndex = 0; // the index of
the current reading
int total = 0; // the running
total
int average = 0;

void setup()
{
  pinMode(outPin, OUTPUT);
  pinMode(inPin, INPUT);

  TinyWireS.begin(1);

  for (int thisReading = 0; thisReading <
numReadings; thisReading++) {
    readings[thisReading] = 0;
  }
}

void loop()
{
  delay(1);

  total = total - readings[readIndex];
  readings[readIndex] =
analogRead(inPin);
  total = total + readings[readIndex];
  readIndex = readIndex + 1;

  // if we're at the end of the array...
  if (readIndex >= numReadings) {
    // ...wrap around to the beginning:
    readIndex = 0;
  }

  average = (total / numReadings);

  int a = 0; //1 - Ganho1;; 2 - equilibrio;;
3 - leitura

  if (TinyWireS.available())
  {
    a = TinyWireS.receive();
    delay(1);
    TinyWireS.send(a);

    switch (a)
    {
      case 1: //ganho
        value_AD2 = TinyWireS.receive();
        value_AD2 = value_AD2 << 8;
        value_AD = value_AD2 |
TinyWireS.receive();
        gain();
        break;

      case 2:
        equilibrio();
        break;

      case 3:
        leitura();
        break;
    }
  }
}
```

```

}
}
void leitura()
{
    value_AD = (average) * (5.0 / 1023.0);
    value_AD = value_AD + erro;

    byte* b = (byte*) &value_AD;

    TinyWireS.send(b[3]);
    TinyWireS.send(b[2]);
    TinyWireS.send(b[1]);
    TinyWireS.send(b[0]);
}

void equilibrio()
{
    //TinyWireS.send(10);
    value_AD = 2.5;
    offset();

    delay(5);

    ADValue = analogRead(inPin);
    voltage = ADValue * (5.0 / 1023.0);

    if (voltage > 2.5)
    {
        value_AD = 5 - voltage;
        offset();
    }

    if (voltage < 2.5)
    {
        value_AD = 2.5 + (2.5 - voltage);
        offset();
    }

    delay(10);

    if (voltage != 0)
    {
        TinyWireS.send(10);
    }
    else
    {
        TinyWireS.send(20);
    }
}
}
}
ADValue = analogRead(inPin);
voltage = ADValue * (5.0 / 1023.0);

erro = 2.5 - voltage;
}
void Seq() //realiza sequência da
"palavra" a enviar
{
    sendStart();
    sendParam(Simular_COD);
    sendParam(Param);
    sendParam(Dummy_COD);
    sendValue(value);
    sendStop();
}

void sendStart() //envia o campo 0
(1000 0000 0001)
{
    sendBit(1);
    for (int i = 0; i < 10; i++)
    {
        sendBit(0);
    }
    sendBit(1);
}

void sendParam(uint8_t value) //envia o
campo 1, 2 e 3 ("Simular_COD",
"Param" e "Dummy_COD")
{
    sendBit((value >> 1) & 1);
    sendBit(value & 1);
}

void sendValue(uint8_t value) //envia o
campo 4 ("value")
{
    uint8_t i = 8;
    do {
        i--;
        sendBit((value >> i) & 1);
    } while (i > 0);
}

void sendStop() //envia o campo 5
(0111 1111 1110)
{
    sendBit(0);
}

```

```

for (uint8_t i = 0; i < 10; i++)
    sendBit(1);
sendBit(0);
}

void sendBit(bool value) //bit 1 ou 0 ??
{
    if (value)
        send1();
    else
        send0();
}

void send0() //envia o sinal do nível 0
{
    digitalWrite(outPin, HIGH);
    digitalWrite(outPin, LOW);
    delayMicroseconds(10);
}

void send1() //envia o sinal do nível 1
{
    digitalWrite(outPin, HIGH);
    delayMicroseconds(65);
    digitalWrite(outPin, LOW);
    delayMicroseconds(10);
}

void gain()
{
    if (value_AD <= 70) {
        Param = FSG_COD;
        value = 0;
        for (int i = 0; i < 1; i++) {
            Seq();
        }
        Param = SSG_COD;
        value = 0;
        for (int i = 0; i < 1; i++) {
            Seq();
        }
    }
    else if (value_AD < 100) {
        Param = FSG_COD;
        value = (int)(((value_AD / 17.5) -
4.0) * 66.7);
        for (int i = 0; i < 1; i++) {
            Seq();
        }
        Param = SSG_COD;
        value = 0;
        for (int i = 0; i < 1; i++) {
            Seq();
        }
    }
    else if (value_AD < 140) {
        Param = FSG_COD;
        value = (int)(((value_AD / 25) - 4.0)
* 66.7);
        for (int i = 0; i < 1; i++) {
            Seq();
        }
        Param = SSG_COD;
        value = 1;
        for (int i = 0; i < 1; i++) {
            Seq();
        }
    }
    else if (value_AD < 200) {
        Param = FSG_COD;
        value = (int)(((value_AD / 35) - 4.0)
* 66.7);
        for (int i = 0; i < 1; i++) {
            Seq();
        }
        Param = SSG_COD;
        value = 2;
        for (int i = 0; i < 1; i++) {
            Seq();
        }
    }
    else if (value_AD < 280) {
        Param = FSG_COD;
        value = (int)(((value_AD / 50) - 4.0)
* 66.7);
        for (int i = 0; i < 1; i++) {
            Seq();
        }
        Param = SSG_COD;
        value = 3;
        for (int i = 0; i < 1; i++) {
            Seq();
        }
    }
    else if (value_AD < 400) {
        Param = FSG_COD;
        value = (int)(((value_AD / 70) - 4.0)
* 66.7);
        for (int i = 0; i < 1; i++) {
            Seq();
        }
    }
}

```

```

    }
    Param = SSG_COD;
    value = 4;
    for (int i = 0; i < 1; i++) {
        Seq();
    }
}
else if (value_AD < 560) {
    Param = FSG_COD;
    value = (int)(((value_AD / 100) - 4.0)
* 66.7);
    for (int i = 0; i < 1; i++) {
        Seq();
    }
    Param = SSG_COD;
    value = 5;
    for (int i = 0; i < 1; i++) {
        Seq();
    }
}
else if (value_AD < 800) {
    Param = FSG_COD;
    value = (int)(((value_AD / 140) - 4.0)
* 66.7);
    for (int i = 0; i < 1; i++) {
        Seq();
    }
    Param = SSG_COD;
    value = 6;
    for (int i = 0; i < 1; i++) {
        Seq();
    }
}
else if (value_AD < 1280) {
    Param = FSG_COD;

    value = (int)(((value_AD / 200) - 4.0)
* 66.7);
    for (int i = 0; i < 1; i++) {
        Seq();
    }
    Param = SSG_COD;
    value = 7;
    for (int i = 0; i < 1; i++) {
        Seq();
    }
}
else if (value_AD >= 1280) {
    Param = FSG_COD;
    value = 127;
    for (int i = 0; i < 1; i++) {
        Seq();
    }
    Param = SSG_COD;
    value = 7;
    for (int i = 0; i < 1; i++) {
        Seq();
    }
}
}

void offset()
{
    Param = OFS_COD;
    value = 256 * (value_AD) / 5 - 0.5;;
//função de transferência do offset
    for (int i = 0; i < 1; i++) {
        Seq();
    }
}
}

```