

FACULDADE DE TECNOLOGIA DE SÃO PAULO - FATEC-SP

Danrlei de Andrade Silva
Diego Fernando Bazela
Guilherme Fernandes Pereira

BALANÇA

Sistema automatizado de coleta e pesagem de metais para reciclagem.

São Paulo – SP
2020

Danrlei de Andrade Silva
Diego Fernando Bazela
Guilherme Fernandes Pereira

BALANÇA

Sistema automatizado de coleta e pesagem de metais para reciclagem.

Trabalho de Conclusão de Curso
apresentado à Faculdade de
Tecnologia de São Paulo para
obtenção do Título de Tecnólogo em
Eletrônica Industrial.

Orientador: Prof. Me. Carlos Takeo Akamine

São Paulo – SP
2020

Danrlei de Andrade Silva
Diego Fernando Bazela
Guilherme Fernandes Pereira

BALANÇA

Sistema automatizado de coleta e pesagem de metais para reciclagem.

Trabalho submetido como exigência parcial para a obtenção do Grau de
Tecnólogo em Eletrônica Industrial.

Parecer do Professor Orientador

Conceito/Nota Final: _____

Atesto o conteúdo contido na mídia entregue pelos alunos e assinada por mim
para avaliação do TCC.

Estou ciente de que se os alunos não tiverem entregado a mídia conforme regras
do roteiro eles estarão reprovados na disciplina mesmo que esteja aprovado por
mim.

Orientador: Prof. Me. Carlos Takeo
Akamine

SÃO PAULO, _____ de _____ de 2020.

Assinatura do Orientador

AGRADECIMENTOS

Agradecemos primeiramente a Deus que nos concedeu a oportunidade de cursar uma universidade pública conceituada nos dando força e discernimento para nossa chegada até aqui, as nossas famílias, em especial as nossas mães, por nos dar a oportunidade de buscar uma boa formação acadêmica, pelo amor e pela paciência conosco ao longo dos anos para que obtivéssemos nossa graduação.

Agradecemos a todos os amigos que contribuíram com o grupo, aos mais próximos Victor Cassamassimo, Pedro Henrique, Jonatas Florêncio, Joílson Briana, Denimar e Bruno Maia, por trocarem conhecimentos em momentos fáceis e difíceis sempre serão lembrados.

Agradecemos nosso orientador Prof. Me. Carlos Takeo Akamine pelo apoio, paciência e broncas as quais nos fizeram pensar e melhorar, contribuindo para nosso trabalho.

Agradecemos uns aos outros, pela confiança para o cumprimento dos prazos, pela paciência pelos desabafos, pelos momentos difíceis e desanimadores sempre levantando e encorajando aquele que estava para baixo não deixando ninguém desistir e ficar para trás.

RESUMO

O projeto desenvolvido foi o protótipo de uma balança para uso residencial com finalidade de pesar o material metálico reciclável, armazenar essa medida para gerar crédito em conta corrente (desconto no condomínio se optar) após a retirada do material reciclado. A balança no primeiro momento vai apresentar características simplificadas somente com a célula de carga e o programa para realizar a inteligência. Para atender nosso problema essa balança deverá possuir uma interface IHM (interface homem máquina) para os usuários entrarem com seu LOGIN e realizar seus depósitos do seu metal, o sistema armazenará os depósitos acolhidos. Ao chegar à capacidade máxima do coletor o sistema deverá emitir um sinal para proceder com a retirada do montante pela empresa recicladora que irá efetuar o pagamento, que será dividido conforme depósitos realizados pelos usuários.

Palavras-Chaves: reciclagem, célula de carga, retorno financeiro.

ABSTRACT

The project developed was the prototype of a scale for residential use with the purpose of weighing the recyclable metallic material, storing this measure to generate credit in current account after the withdrawal of the recycled material. The scale at first will present simplified features only with the load cell and the program to perform intelligence. To meet our problem this scale should have an HMI interface for users to enter with your LOGIN and make their deposits of your metal, the system will store the deposits received. When the maximum capacity of the collector is reached, the system will emit a signal to proceed with the withdrawal of the amount by the recycling company, which will make the payment, which will be divided according to the deposits made by the users.

Key words: recycling, load cell, financial return.

LISTA DE FIGURAS

Figura 1: Microcontrolador PIC16F877A.	13
Figura 2: Diagrama de blocos do microcontrolador	14
Figura 3: Disposição dos pinos.	15
Figura 4: PL2303 USB/TTL módulo adaptador conversor.	15
Figura 5: Diagrama de blocos PL2303.	16
Figura 6: Disposição dos pinos.	17
Figura 7: Display LCD 16x2.	17
Figura 8: Diagrama de blocos.	18
Figura 9: Módulo HX711	19
Figura 10: Estrutura do módulo HX711.	19
Figura 11: Pinagem do HX.	21
Figura 12: Célula de carga até 20Kg.	21
Figura 13: HX711 junto com a ponte de Wheatstone.	22
Figura 14: Tela de apresentação do MiKroC	23
Figura 15: PicKit 2 Programmer	24
Figura 16: Tela de apresentação do Proteus.....	24
Figura 17: Microsoft Visual Studio 2019 (MVS 2019).	25
Figura 18: Ambiente de desenvolvimento MVS 2019.	25
Figura 19: Camada de conexão com o Banco de Dados.	26
Figura 20: SQL Server Management Studio v18.2.	27
Figura 21: Ambiente de desenvolvimento SSMS v18.2.	27
Figura 22: Barra de Ferramentas SSMS v18.2.....	28
Figura 23: Comando select para visualizar informações da tabela Usuários. ...	29
Figura 24: Placa em desenvolvimento.	30
Figura 25: Esquema elétrico do circuito.	31
Figura 26: Placa desenvolvida de suporte do módulo HX.	31
Figura 27: Circuito elétrico final no software Proteus.....	32
Figura 28: Fase de teste inicial e calibração da balança.	33
Figura 29: Parte mecânica da balança.	34
Figura 30: Balança comercial de 10 Kg.....	34
Figura 31: Caracterização da balança com o peso do multímetro.	35
Figura 32: Gráfico de operação do sistema.....	36

Figura 33: Conexão do conversor com PIC.....	37
Figura 34: Tela de Login do Aplicativo.	38
Figura 35: Tela de transição da aplicação.....	38
Figura 36:Tela do administrador.....	39
Figura 37: Tela do cliente.....	39
Figura 38: Hardware e Software integrados.	40
Figura 39: Medida com o barbante.....	41
Figura 40: Medida com uma caixa de som.....	41
Figura 41: Medida com uma panela de pressão.....	42
Figura 42: Fluxograma das etapas do projeto	43
Figura 43: Mapa Canvas.....	44
Figura 44: Bonificação de alguns dos itens triciclo.	45
Figura 45: Retorna Machine.....	45
Figura 46: Sensor indutivo.	47

LISTA DE QUADROS

Quadro 1: Atribuição dos pinos LCD.	18
Quadro 2: Especificações elétricas do HX711.....	20
Quadro 3: Especificações da célula de carga.	22
Quadro 4: Lista dos materiais utilizados no projeto.	40
Quadro 5: Levantamento de erro do sistema.	42

SUMÁRIO

AGRADECIMENTOS	4
RESUMO	5
Palavras-Chaves:	5
ABSTRACT.....	6
Key words:	6
LISTA DE FIGURAS.....	7
LISTA DE QUADROS	8
1. INTRODUÇÃO	10
2. OBJETIVO	11
2.1 OBJETIVO PRINCIPAL.....	11

2.2 OBJETIVOS ESPECIFICOS.....	11
3. JUSTIFICATIVA	12
4. FUNDAMENTAÇÃO TEÓRICA	12
4.1 DISPOSITIVOS	12
4.2 PIC16F877A.....	13
4.3 PL 2303	15
4.4 DISPLAY LCD	17
4.5 HX 711.....	19
4.6 CÉLULA DE CARGA.....	21
5. SOFTWARES	23
5.1 MikroC	23
5.2 PicKit 2 v2.61.....	23
5.3 Proteus	24
5.4 Microsoft Visual Studio 2019 (MVS 2019)	25
5.5 Microsoft SQL Server Management Studio v18.2 (SSMS v18.2)	26
6. DESENVOLVIMENTO DO SISTEMA E TESTES PRELIMINARES.....	29
7. RESULTADOS	40
8. FLUXOGRAMA DAS ETAPAS DO PROJETO.....	43
9. OPERAÇÃO DO SISTEMA.....	43
10. MODELO DE NEGÓCIO.....	43
11. EQUIPAMENTOS SEMELHANTES	44
12. CONCLUSÕES FINAIS.....	46
13. TRABALHOS FUTUROS	47
14.REFERÊNCIAS	48
15. ANEXOS.....	49
15.1 Código Principal do PIC:	49
15.2 Código de Caracterização da célula de carga:.....	53

1. INTRODUÇÃO

Reciclagem é um assunto que gera preocupações em nível mundial. Para além de uma forte sensibilização à população, a aposta dos mercados mundiais tem se voltado nos últimos tempos para os grandes investimentos em alta tecnologia em métodos e processos automatizados que permitem uma grande recuperação dos materiais, com menos recursos utilizando-se de tecnologia de ponta. A reciclagem ajuda a amenizar um dos maiores problemas da atualidade o descarte do lixo, reciclar significa transformar objetos, materiais usados em novos produtos para o consumo. Esta necessidade foi despertada pelos seres humanos, a partir do momento em que verificaram os benefícios que este procedimento traz para o planeta. A partir da década de 1980, a produção de embalagens e produtos descartáveis aumentou significativamente, assim como a produção de lixo.

Muitos governos e ONGs estão cobrando de empresas posturas responsáveis, o crescimento econômico deve estar aliado à preservação do meio ambiente. Atividades como campanhas de coleta seletiva de lixo e reciclagem de alumínio e papel, já são comuns em várias partes do mundo. No processo de reciclagem, que além de preservar o meio ambiente também gera riquezas, os materiais mais reciclados são o vidro, o alumínio, o papel e o plástico.

Esta reciclagem contribui para a diminuição significativa da poluição do solo, da água e do ar. Muitas indústrias estão reciclando materiais como uma forma de reduzir os custos de produção. Outro benefício da reciclagem é a quantidade de empregos que ela tem gerado nas grandes cidades. Muitos desempregados estão buscando trabalho neste setor e conseguindo renda para manterem suas famílias. Cooperativas de catadores de papel e alumínio já é realidade nos centros urbanos do Brasil.

Dentre as muitas razões do porquê reciclar as mais importantes são economia e prevenção. Seu processo ajuda a diminuir o desperdício de energia, água, a contaminação do solo e lençóis freáticos, além de ajudar a proteger recursos minerais importantes para manutenção do ecossistema e a vida humana. Não é apenas o ambiente que se beneficia com a reciclagem, nós também, já que nele

está nossa casa, trabalho e opções de lazer. Quando reciclamos, aquilo que antes era lixo pode ser transformado em novos produtos, conservando, assim, a matéria-prima e protegendo o ambiente para nossa vida presente e futura.

Um quilo de alumínio corresponde a aproximadamente 74 latas do material. O alumínio líquido (700 °C) demora até duas horas e meia para atingir o estado sólido. O ciclo médio de vida de uma lata de alumínio é de 30 dias, desde sua produção até seu retorno já reciclada. O processo de reciclagem de alumínio libera apenas 5% das emissões de gás de efeito estufa. A cada quilo de alumínio reciclado, cinco quilos de bauxita (minério de onde se produz o alumínio) são poupados. Para se reciclar uma tonelada de alumínio, gasta-se somente 5% da energia que seria necessária para se produzir a mesma quantidade de alumínio primário, ou seja, a reciclagem do alumínio proporciona uma economia de 95% de energia elétrica. Para se ter uma ideia, a reciclagem de uma única latinha de alumínio economiza suficiente energia para manter um aparelho de TV ligado durante três horas [2].

2. OBJETIVO

2.1 OBJETIVO PRINCIPAL

Este projeto tem como objetivo desenvolver uma balança que traga benefícios ao meio ambiente e retorno financeiro para os usuários.

2.2 OBJETIVOS ESPECIFICOS

Nossos objetivos específicos são:

- Desenvolver um programa em linguagem C para atender a necessidade da balança;
- Montar um protótipo de balança;
- Criar uma interface de comunicação.

3. JUSTIFICATIVA

Logo o desenvolvimento deste trabalho, justifica a construção de uma balança, automatizada e direcionada a captação de metais recicláveis que proporcione para o usuário satisfação de contribuir com o meio ambiente e que também traga um retorno financeiro.

4. FUNDAMENTAÇÃO TEÓRICA

Para fundamentar esta escolha falaremos sobre a empregabilidade e os conceitos teóricos necessários para a construção de uma balança automatizada. A balança é um instrumento antigo criado por volta do ano 5000 a.C. [3], os egípcios inventaram a balança para o pesarem ouro, pois, a necessidade de pesar o ouro que foi considerado o mais valioso dos metais, as balanças egípcias estão representadas em inúmeros murais e papiros. Mais tarde no tempo surgiu a chamada “balança romana”. Esta balança tinha os dois braços de pesagem com comprimentos desiguais e o objeto que se pretendia pesar era sempre colocado no braço mais curto. A balança estava pendurada num gancho existente num ponto fixo e no braço mais longo deslizava um peso (ou pilão) que se fazia correr em ambas as direções até se encontrar o ponto de equilíbrio. Nesse braço mais longo existiam marcações que indicavam o peso do objeto. Este tipo de balança teve um enorme sucesso em Roma e acabou por se espalhar pelo mundo, chegando até aos nossos dias, podendo-se ainda encontrar algumas em uso por vendedores de artigos alimentares em feiras e mercados.

4.1 DISPOSITIVOS

Os dispositivos que fazem parte da construção da balança e do circuito eletrônico necessário e suas aplicações para a realização desse trabalho são: PIC16F877A, PL2303, DISPLAY LCD, HX711 e a CÉLULA DE CARGA.

4.2 PIC16F877A

Para efetuar esse projeto vamos utilizar o microcontrolador PIC16F877A (Figura 1), fabricado pela empresa Microchip, eles possuem várias famílias de microcontroladores, as diferenças entre essas famílias, são periféricos, espaço de memória, encapsulamentos e pinos de entrada e saída. Ele atende todas as necessidades do projeto, pois possui pinos de entradas/saídas suficientes, comunicação serial que é usada no projeto e espaço de memória suficiente para armazenar e executar o programa em linguagem C.



Figura 1: Microcontrolador PIC16F877A.

Ele possui estas especificações:

- 40 pinos, incluindo analógicos e 33 Inputs/Outputs digitais;
- 15 tipos de interrupções disponíveis;
- Clock de 4 a 20MHz;
- Pilha com oito níveis de profundidade;
- Memória RAM com 368 bytes;
- Conversores A/D;
- PWM;
- Modo UART (comunicação serial);
- 8kBytes de memória ROM.

Na Figura 2 está expresso o diagrama de blocos deste PIC, ele deve ser levado em consideração na hora de fazer a programação, para respeitar seu funcionamento e de seus periféricos.

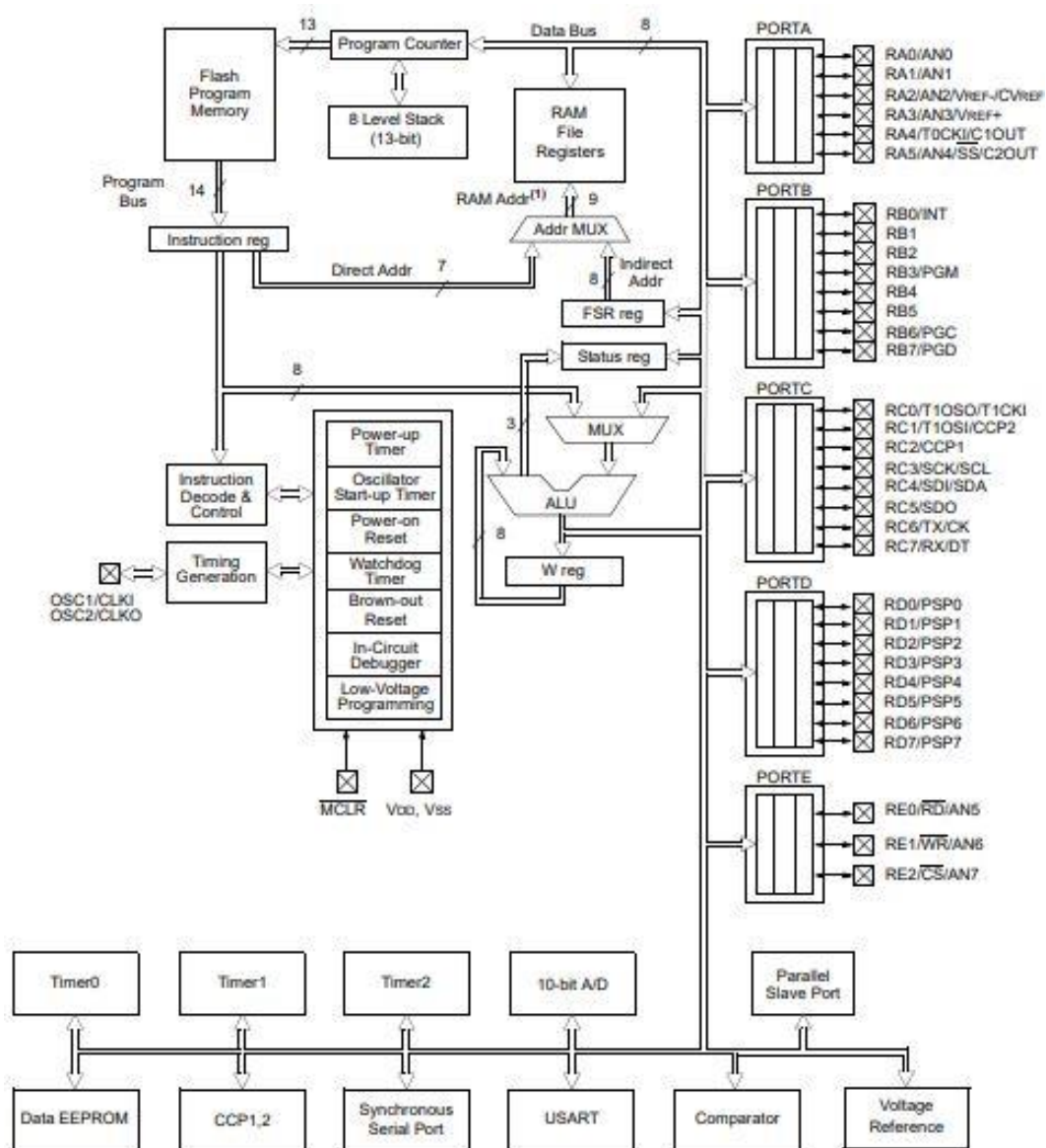


Figura 2: Diagrama de blocos do microcontrolador

Esse microcontrolador é disponível nos encapsulamentos PDIP (Plastic Dual In-line Package), PLCC (Plastic leaded chip carrier) , TQFP (Thin Quad Flat Packages), QFN (Quad Flat No leads). Cada encapsulamento é adequado

para uma montagem de placa diferente, neste caso será utilizado o PDIP como mostra a Figura 3.

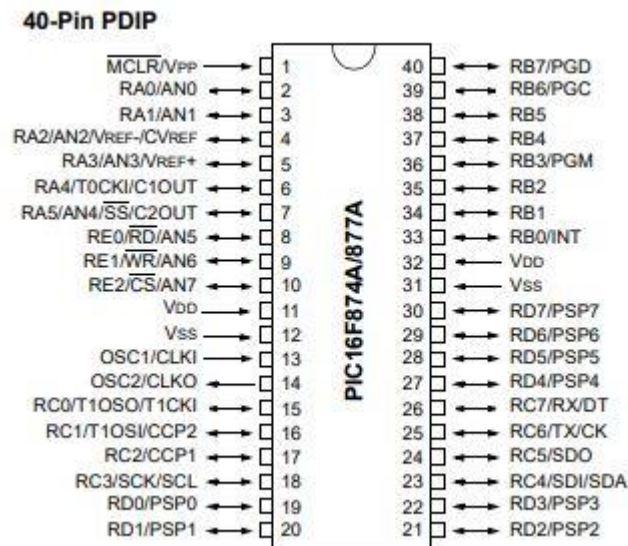


Figura 3: Disposição dos pinos.

4.3 PL 2303

É um conversor USB-Serial TTL PL2303 ele é uma placa com o circuito integrado PL2303HX, com a placa pronta fica fácil a aplicação. Esse circuito integrado faz a conversão de sinais USB para RS232, tornando mais prático a comunicação entre computadores e microcontroladores. Possui tensões de 3.3V e 5V. O PL2303 já é comercializado em uma placa pronta e é vendida como DRIVER PL2303, como mostra a Figura 4.



Figura 4: PL2303 USB/TTL módulo adaptador conversor.

A Figura 5 mostra o diagrama de blocos do PL2303.

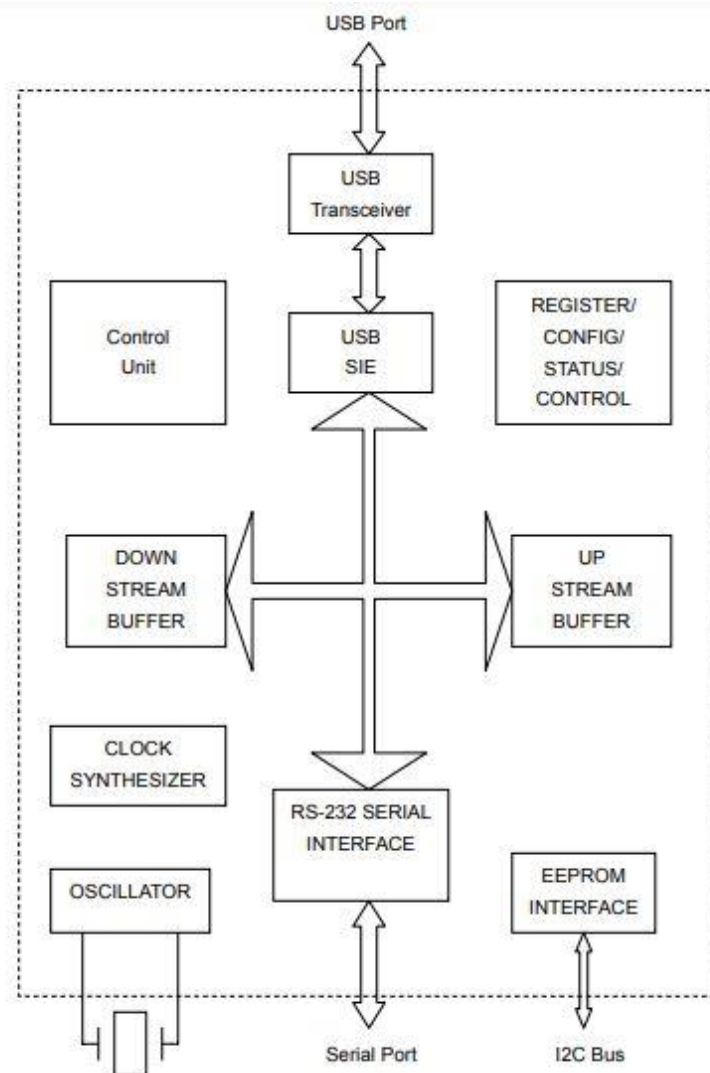


Figura 5: Diagrama de blocos PL2303.

Este circuito integrado é disponível no encapsulamento SSOP (Shrink small-outline package). A Figura 6 mostra a disposição dos pinos do circuito integrado, bem como os nomes dos pinos deste componente.

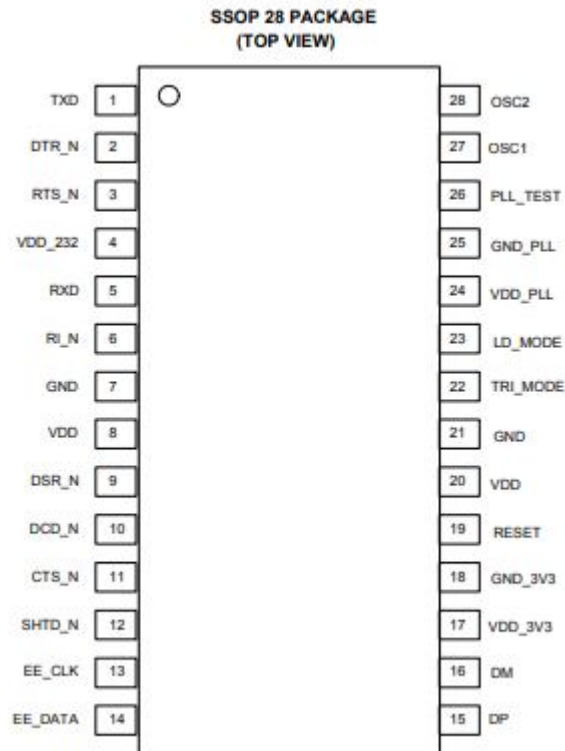


Figura 6: Disposição dos pinos.

4.4 DISPLAY LCD

Para fazer a interface da balança e mostrar o peso que está sendo medido, teremos o aplicativo para armazenar esse dado e mostrar para o usuário, mas no protótipo foi utilizado um display LCD como o da Figura 7, para mostrar através de seus caracteres este valor medido.

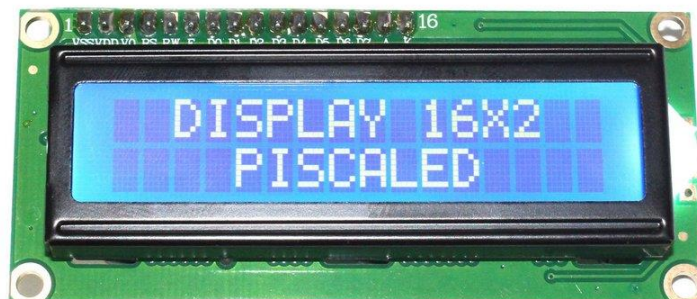


Figura 7: Display LCD 16x2.

O display utilizado é o 16x2 (1602A). Possui 16 pinos, fundo azul com os caracteres em branco. O Quadro 1 mostra a atribuição dos pinos, essas atribuições são levadas em consideração na inicialização do LCD.

No.	Symbol	Level	Function
1	Vss	--	0V
2	Vdd	--	+5V
3	V0	--	for LCD
4	RS	H/L	Register Select: H:Data Input L:Instruction Input
5	R/W	H/L	H--Read L--Write
6	E	H,H-L	Enable Signal
7	DB0	H/L	Data bus used in 8 bit transfer
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	Data bus for both 4 and 8 bit transfer
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	
15	BLA	--	BLACKLIGHT +5V
16	BLK	--	BLACKLIGHT 0V-

Quadro 1: Atribuição dos pinos LCD.

Para a aplicação desse projeto esse display atende bem a necessidade, porém no mercado existe display com maiores capacidades, se utilizado aumentaria bem o custo do projeto. Na placa do display LCD há circuitos integrados responsáveis pelo processamento dos dados que será enviado até o próprio. A Figura 8 mostra o diagrama de blocos de utilização do display, a configuração de inicialização e escrita são controladas pela lógica desenvolvida no programa MiKroC.

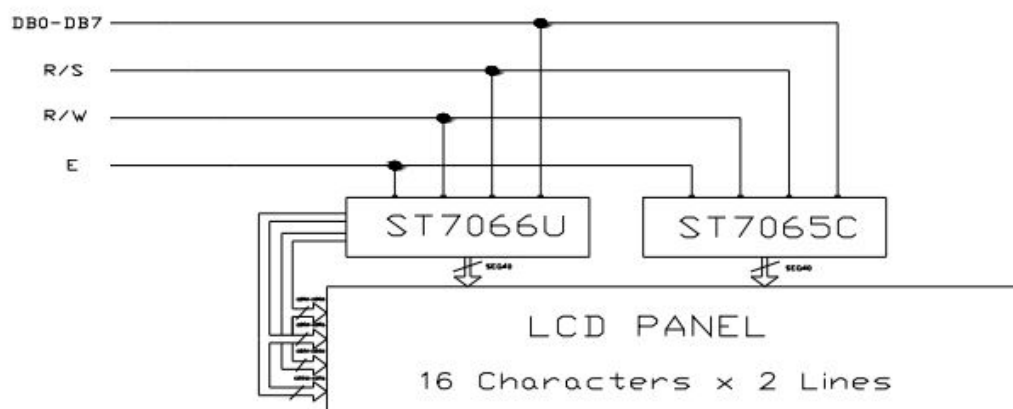


Figura 8: Diagrama de blocos.

4.5 HX 711

O HX711 é um circuito integrado que converte o sinal analógico para digital, e quando ele absorve o sinal já amplifica-o, é bem utilizado em circuitos como balança e é comercializado em uma placa pronta com esse circuito integrado, como mostra a Figura 9.

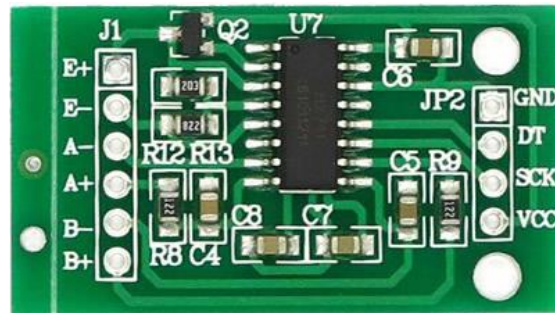


Figura 9: Módulo HX711

O ganho na saída pode ser de 32, 64 e 128, essa conversão analógica/digital consiste em captar o sinal analógico nos pinos : E+, E-, A-, A+, B- e B+, com esse sinal captado é feita a conversão do sinal para uma informação de 24bits, que será tratado pelo pic através de sua lógica de programação, que pega essa informação dos pinos DT e SCK. A Figura 10 mostra a estrutura do circuito integrado.

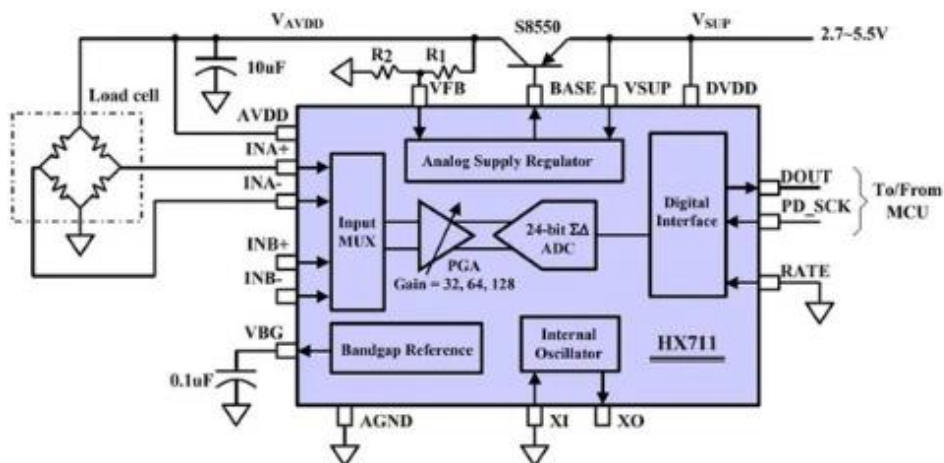


Figura 10: Estrutura do módulo HX711.

O Quadro 2 mostra as especificações técnicas do módulo HX711, como corrente de alimentação, variação de ganhos e alguns outros aspectos. Essas informações são necessárias para o desenvolvimento do circuito elétrico, para não causar danos ao módulo, como queima das saídas que controlam o sinal.

Parameter	Notes	MIN	TYP	MAX	UNIT
Full scale differential input range	$V(\text{inp}) - V(\text{inn})$	$\pm 0.5(\text{AVDD}/\text{GAIN})$			V
Common mode input		AGND+1.2		AVDD-1.3	V
Output data rate	Internal Oscillator, RATE = 0		10		Hz
	Internal Oscillator, RATE = DVDD		80		
	Crystal or external clock, RATE = 0		$f_{\text{clk}}/1,105,920$		
	Crystal or external clock, RATE = DVDD		$f_{\text{clk}}/138,240$		
Output data coding	2's complement	800000		7FFFFF	HEX
Output settling time ⁽¹⁾	RATE = 0		400		ms
	RATE = DVDD		50		
Input offset drift	Gain = 128		0.2		mV
	Gain = 64		0.4		
Input noise	Gain = 128, RATE = 0		50		nV(rms)
	Gain = 128, RATE = DVDD		90		
Temperature drift	Input offset (Gain = 128)		± 6		nV/°C
	Gain (Gain = 128)		± 5		ppm/°C
Input common mode rejection	Gain = 128, RATE = 0		100		dB
Power supply rejection	Gain = 128, RATE = 0		100		dB
Reference bypass (V _{BG})			1.25		V
Crystal or external clock frequency		1	11.0592	20	MHz
Power supply voltage	DVDD	2.6		5.5	V
	AVDD, VSUP	2.6		5.5	
Analog supply current (including regulator)	Normal		1400		μA
	Power down		0.3		
Digital supply current	Normal		100		μA
	Power down		0.2		

(1) Settling time refers to the time from power up, reset, input channel change and gain change to valid stable output data.

Quadro 2: Especificações elétricas do HX711.

A Figura 11 mostra a descrição dos pinos e encapsulamento do C.I. HX711, esses pinos já vêm conectados as entradas e saídas do módulo utilizado, os pinos que têm maior relevância para o projeto são: PD_SCK e DOUT. Eles estão conectados as saídas do módulo que vão para o microcontrolador.

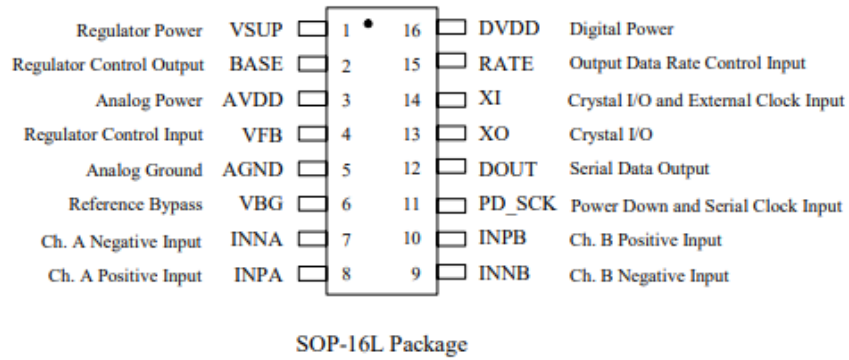


Figura 11: Pinagem do HX.

4.6 CÉLULA DE CARGA

O hardware foi desenvolvido através do princípio de extensometria (strain gauge), onde é formado uma ponte de Wheatestone, onde temos resistores fixos, que são completados por células que quando ocorre uma deformação mecânica altera-se a sua resistência fazendo-se com que a tensão de saída obtida nos terminais de saída seja captada, a variação é pequena, devido a isso precisa ser amplificada, a Figura 12 mostra a célula de carga utilizada neste projeto.

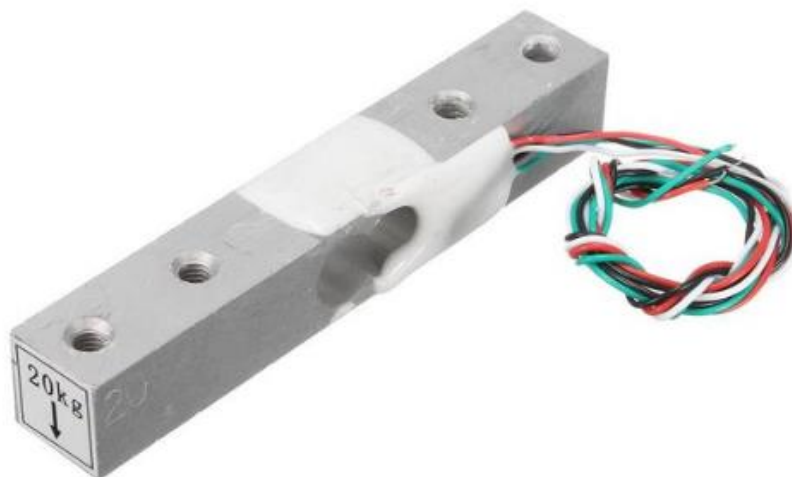


Figura 12: Célula de carga até 20Kg.

Tem diversos tipos de células de carga, a que será utilizada já é uma ponte de Wheatestone completa, tem tipos de células de carga em que é necessário completar com resistores, pois são apenas metade da ponte. Foi utilizado essa célula devido a fácil aplicabilidade, como mostra a Figura 13 onde os cabos vermelhos (+) e preto (-) são a alimentação, e o verde e branco são os sinais obtidos pela variação de tensão que será processado pelo HX711 e consequentemente enviado para o microcontrolador.

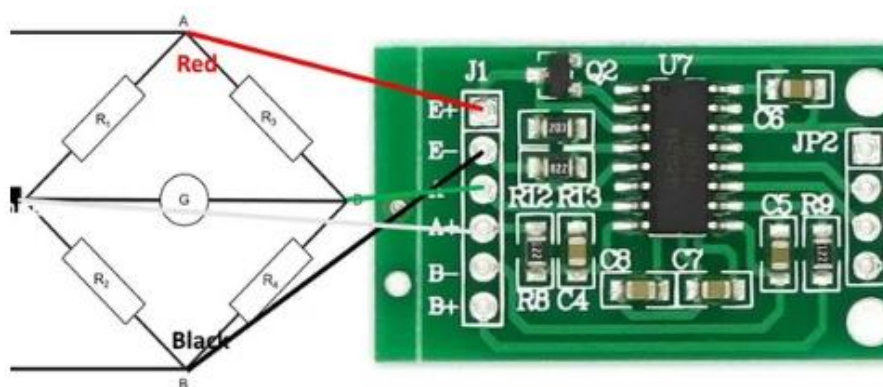


Figura 13: HX711 junto com a ponte de Wheatstone.

O Quadro 3 mostra as especificações da célula de carga (dados fornecidos pelo fornecedor).

Carga nominal	0 a 20 kg
Potência nominal de saída	1.0mv/v $\pm$ 0.15mv/v
Zero saída	$\pm$ 0,1mv/v
Creep	0,03% f.s./30 Min
Extremidade de entrada	Vermelho + (energia), Preto - (energia)
Extremidade de saída	Verde + (sinal), Branco - (sinal)
Tensão de funcionamento recomendada	3 ~ 12 VDC
Tensão máxima de operação	15 VDC
Impedância de entrada	410 +/-30 ohm
Impedância de saída	350 +/-3 ohm
Isolamento	> 2000 mega ohm/50 VDC
Faixa de compensação de temperatura	-10C ~ 40C
Faixa de temperatura de operação	-20C ~ 60C
Classe de proteção	IP65
Material	<u>liga</u> de alumínio

Quadro 3: Especificações da célula de carga.

5. SOFTWARES

5.1 MikroC

A Figura 14 mostra o software MiKroC, ele foi utilizado para realizar a programação do código C a ser carregada no PIC, é um software que compila a programação para o formato hexadecimal, que é o necessário para a gravação do microcontrolador. Nesta programação estão contidas as rotinas de conversão analógica/digital, diminuição de erro da medida, envio da medida para a IHM e escrita do valor de medida em gramas no LCD 16X2.

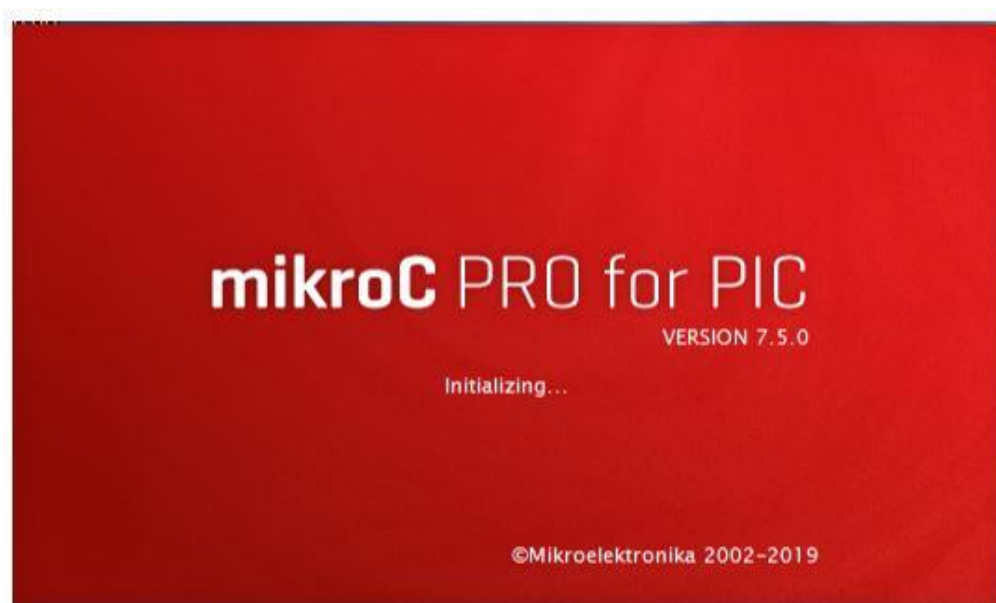


Figura 14: Tela de apresentação do MiKroC

5.2 PicKit 2 v2.61

A Figura 15 mostra o software PicKit 2 que é utilizado para gravação do código hexadecimal no microcontrolador, este código é gerado pelo MiKroC após a compilação do código fonte do projeto. A versão 2 do PicKit já está descontinuada, mas atende as necessidades do projeto. Este programa estabelece conexão com o PIC e grava o código hexadecimal na memória de programa dele.

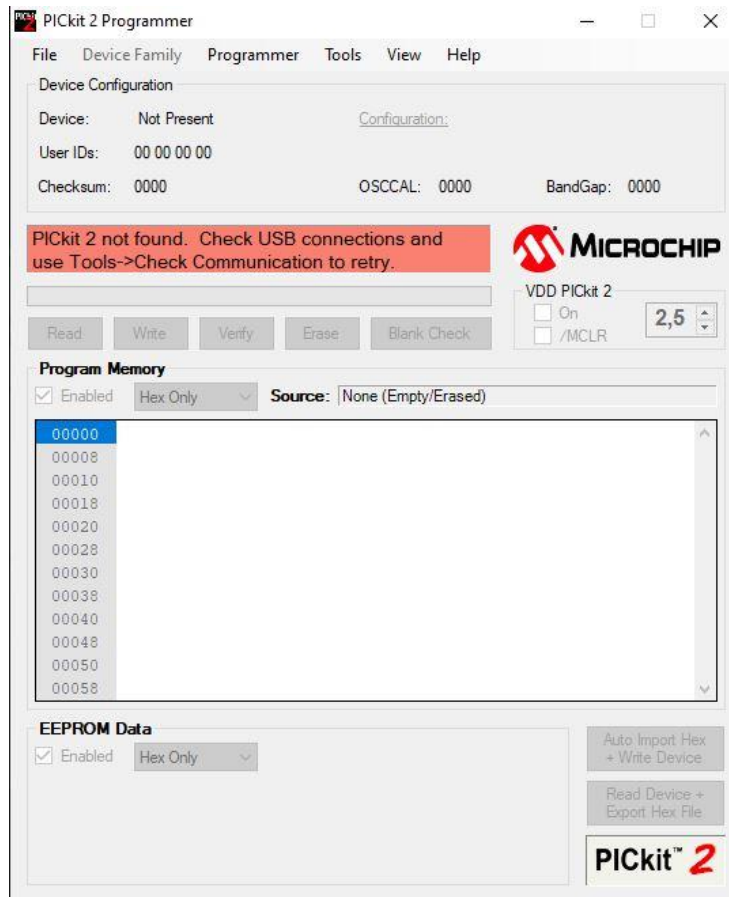


Figura 15: PicKit 2 Programmer

5.3 Proteus

A Figura 16 mostra o software Proteus, versão 8, ele foi utilizado para a elaboração do circuito elétrico para simular o sistema concebido antes dos testes em protoboard.

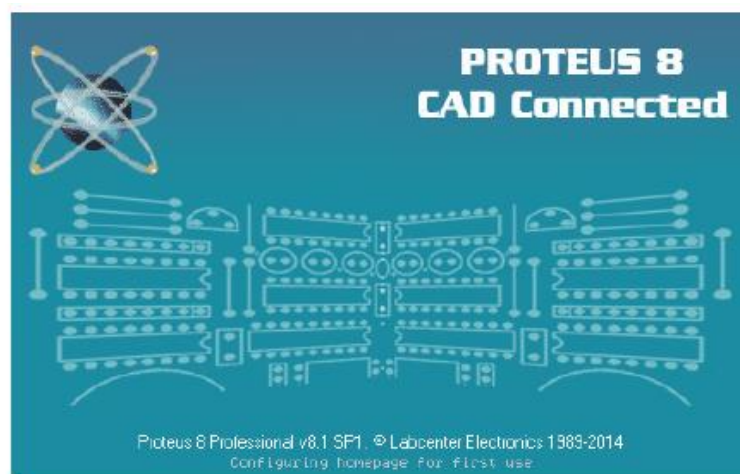


Figura 16: Tela de apresentação do Proteus.

5.4 Microsoft Visual Studio 2019 (MVS 2019)

A Figura 17 mostra a tela de início do software, esta plataforma foi utilizada para o desenvolvimento da IHM (Interface Homem-Máquina), a linguagem de programação utilizada foi a CSharp (C#), ela é uma linguagem de programação, orientada a objetos (POO), ela possui uma interface gráfica onde é possível arrastar os componentes a serem utilizados no projeto.



Figura 17: Microsoft Visual Studio 2019 (MVS 2019).

Estes componentes em grande maioria são visuais como mostra a Figura 18, eles fazem o trabalho de mostrar ao usuário informações, ações que ele pode tomar entre outras tarefas que o aplicativo possa vir a fazer.

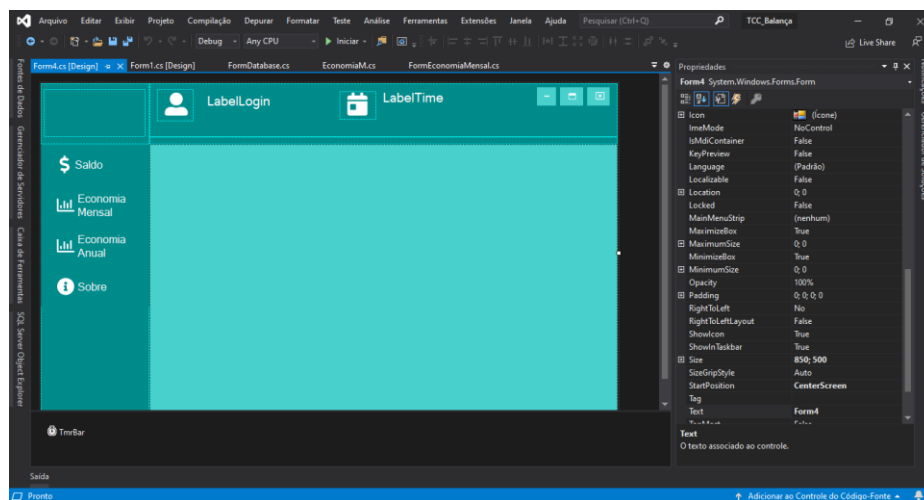


Figura 18: Ambiente de desenvolvimento MVS 2019.

Este tipo de projeto pode ter algumas configurações de organização, neste caso a utilizada foi a de modelo de 3 camadas, a primeira camada é chamada de camada apresentação, ela é responsável por dialogar com o usuário através de botões, textos entre outras ações. A segunda camada trata os dados que o usuário gerou, nesta parte já sendo em linguagem C# e a terceira camada é chamada de DAL, ela normalmente é usada para gerir conexões externas que o programa possa ter, como bancos de dados, como mostra a Figura 19.

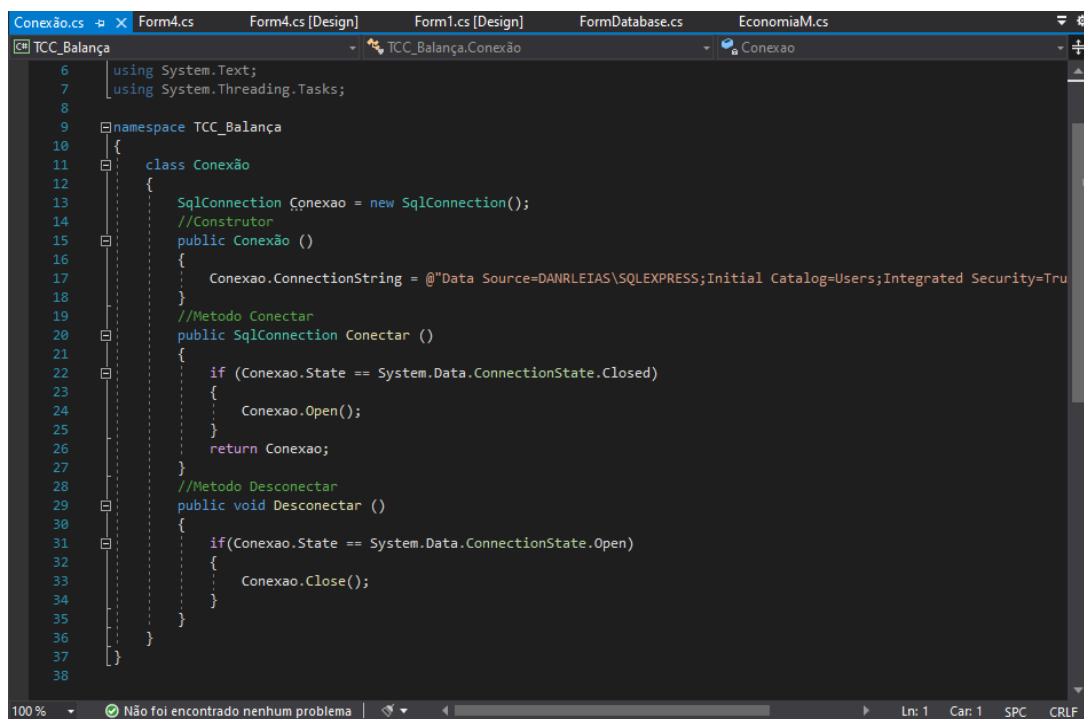


Figura 19: Camada de conexão com o Banco de Dados.

5.5 Microsoft SQL Server Management Studio v18.2 (SSMS v18.2)

A Figura 20 mostra uma das ferramentas que é utilizada para gerir o banco de dados, neste estão presentes as informações de usuários, medições da balança, logs de funcionamento do banco de dados e conexão serial. Existem várias opções de programas que constroem e manipulam banco de dados, a escolha desta foi exclusivamente por conta de ser uma ferramenta Microsoft, pois o Visual Studio também é do mesmo fornecedor.

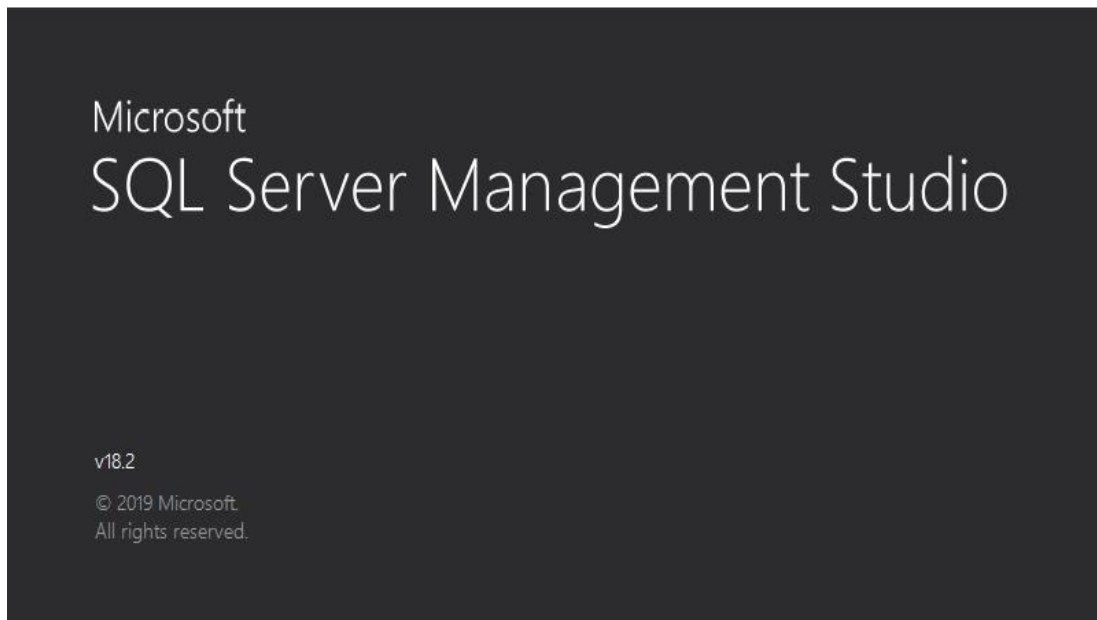


Figura 20: SQL Server Management Studio v18.2.

Ao entrar e colocar as credenciais do seu usuário, é aberta a tela gráfica da ferramenta de gerenciamento de banco de dados, como mostra a Figura 21, pode-se perceber a organização dos bancos de dados existentes no seu computador.

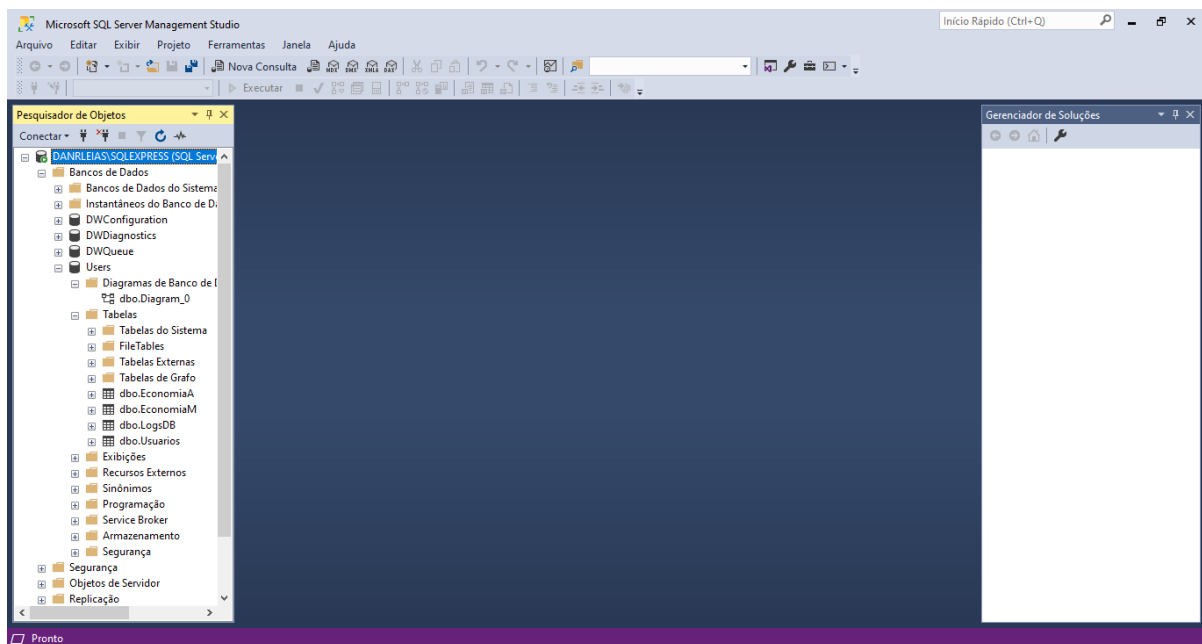


Figura 21: Ambiente de desenvolvimento SSMS v18.2.

No canto esquerdo estão listados todos os bancos de dados e outros tipos de armazenamentos desta ferramenta, como mostra a Figura 22, dentro do banco de dados Users, estão abertas as tabelas onde estão contidas as informações necessárias para o projeto funcionar corretamente. Estas tabelas podem ser acessadas através das procedures criadas, procedures é uma programação que fica guardada dentro do banco de dados, podendo ela receber nenhum parâmetro, um ou vários parâmetros. Seu acionamento pode ocorrer via programação quando necessário.

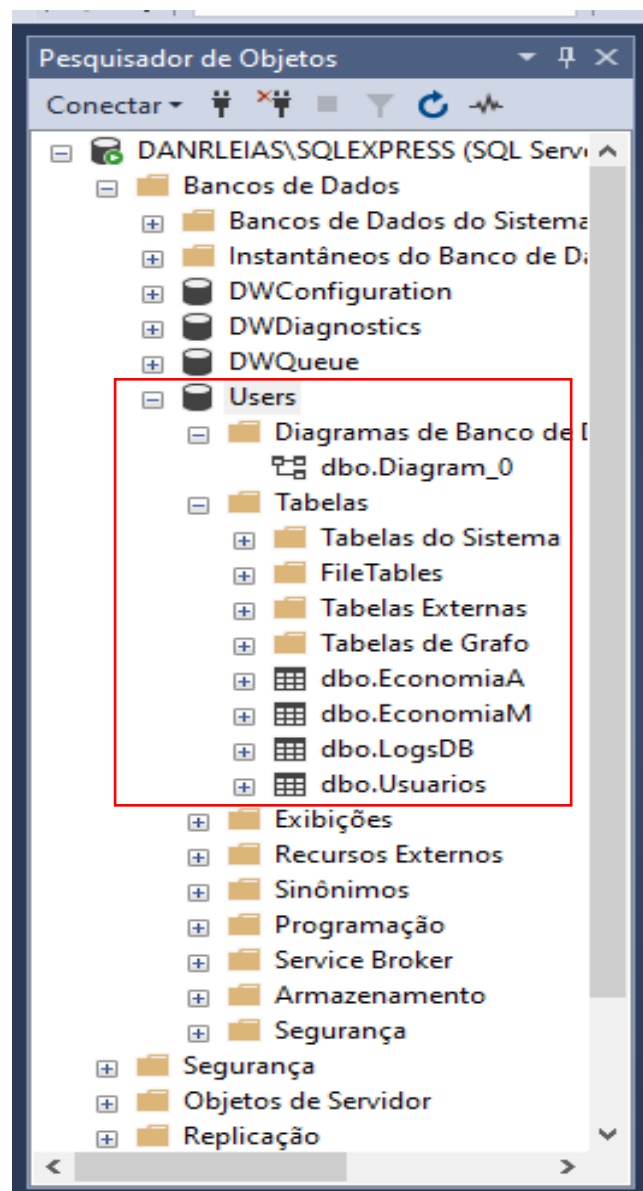


Figura 22: Barra de Ferramentas SSMS v18.2.

Para efetuar leituras dos dados destas tabelas, é necessário usar a linguagem SQL, como mostra a Figura 23, sua sintaxe permite várias ações como criar e deletar tabelas, inserir, alterar, excluir e visualizar dados das tabelas existentes, ainda é possível se utilizar de operadores lógicos para aumentar as funcionalidades desta ferramenta.

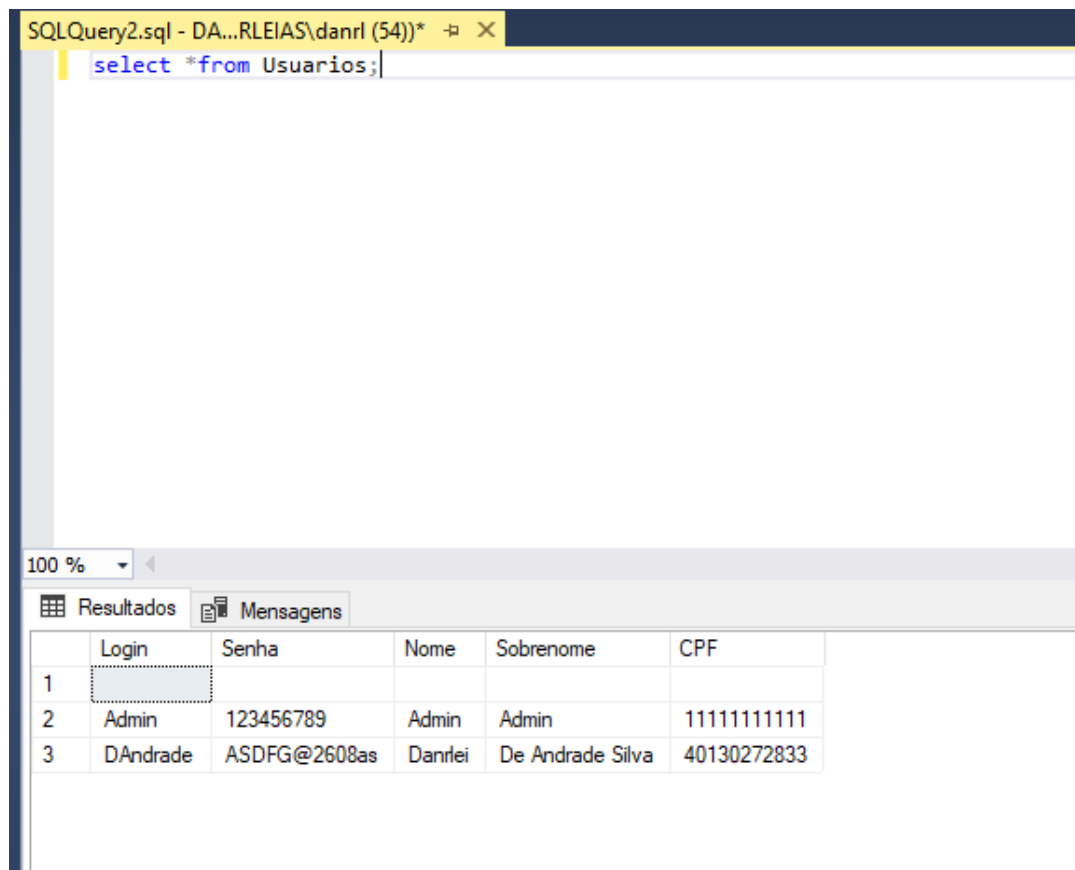


Figura 23: Comando select para visualizar informações da tabela Usuários.

6. DESENVOLVIMENTO DO SISTEMA E TESTES PRELIMINARES

O sinal obtido nos terminais da célula será absorvido pelo módulo HX711, que é um amplificador de sinal junto com um conversor A/D, podendo ser aplicado junto a um microcontrolador com maior facilidade, esse sinal obtido é processado pelo microcontrolador PIC16F877A. A Figura 24 mostra a placa de testes do microcontrolador, está foi desenvolvida durante o quinto semestre de aula, ela possui vários componentes que agregam nos projetos que são utilizados esses tipos de microcontroladores. Após o microcontrolador processar o sinal, é possível ser visualizado através de uma interface gráfica.

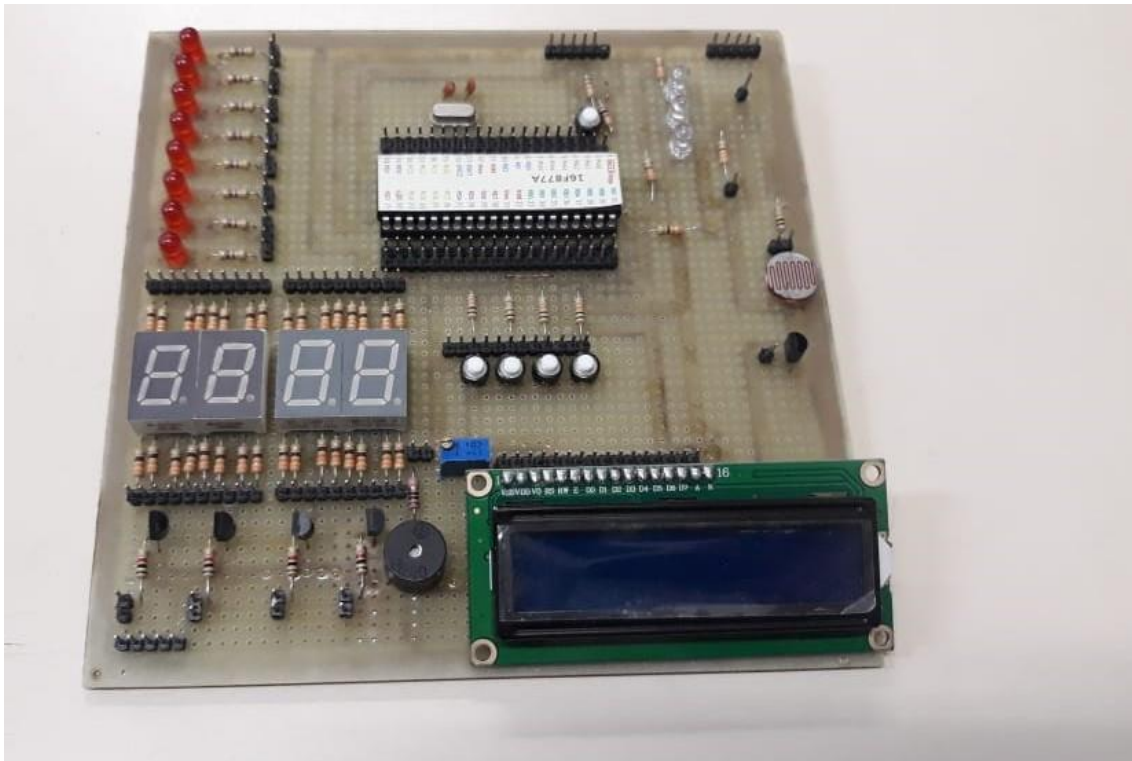


Figura 24: Placa em desenvolvimento.

No momento em que o circuito foi simulado, foram utilizados resistores para simular a célula de carga como está mostrando a Figura 25, pois no Protheus fica mais análogo, até pelo fato de o circuito se basear na diferença de uma resistência através de uma força atribuída. A marcação em vermelho mostra os pinos que controlam a passagem do sinal para o microcontrolador, o pino DOUT manda o valor em bits conforme o pino DSCK está nível alto ou não.

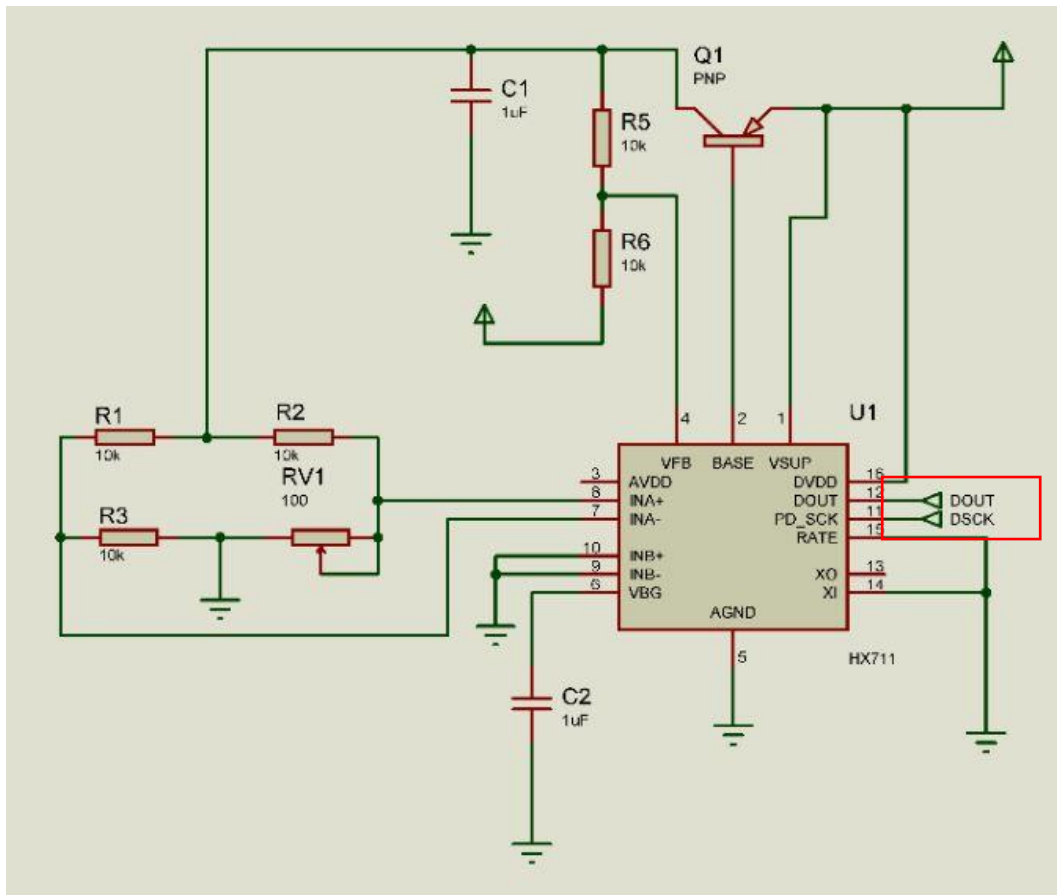


Figura 25: Esquema elétrico do circuito.

O módulo HX711 já existe no mercado pronto, a Figura 26 mostra a placa que foi desenvolvida para facilitar as conexões, não ter necessidade de ficar soldando e dessoldando toda vez que precisasse efetuar algum teste.

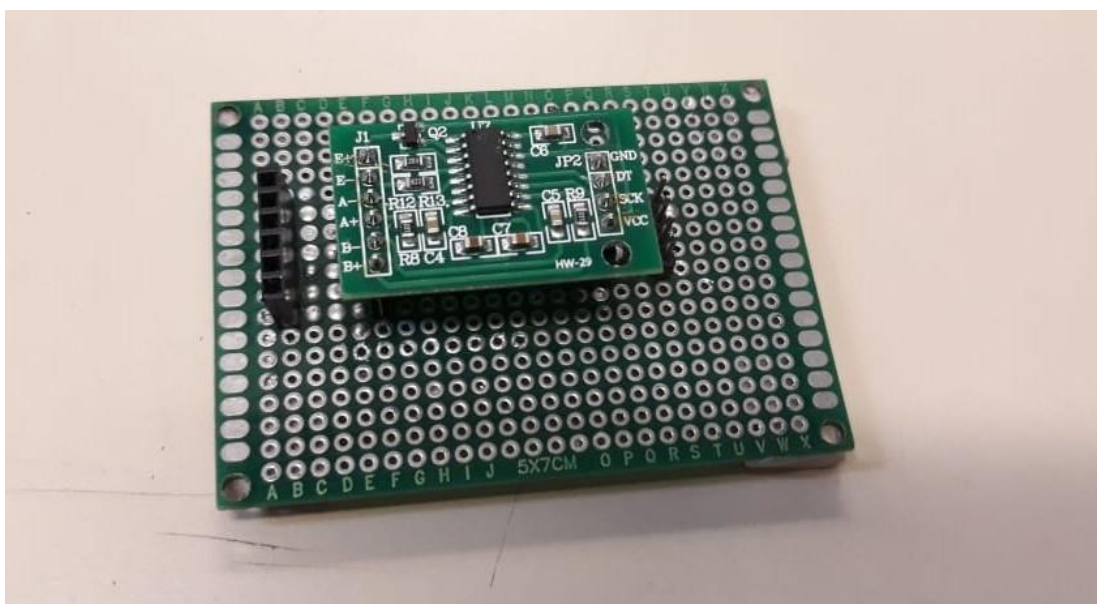


Figura 26: Placa desenvolvida de suporte do módulo HX.

A Figura 27 mostra o circuito do módulo HX711, PIC16F877a e LCD, os três fazem parte do circuito final do projeto. Este esquema elétrico faz a leitura da célula da carga, converte para um valor decimal, transforma esse decimal para um valor de peso em gramas e mostra-o no LCD.

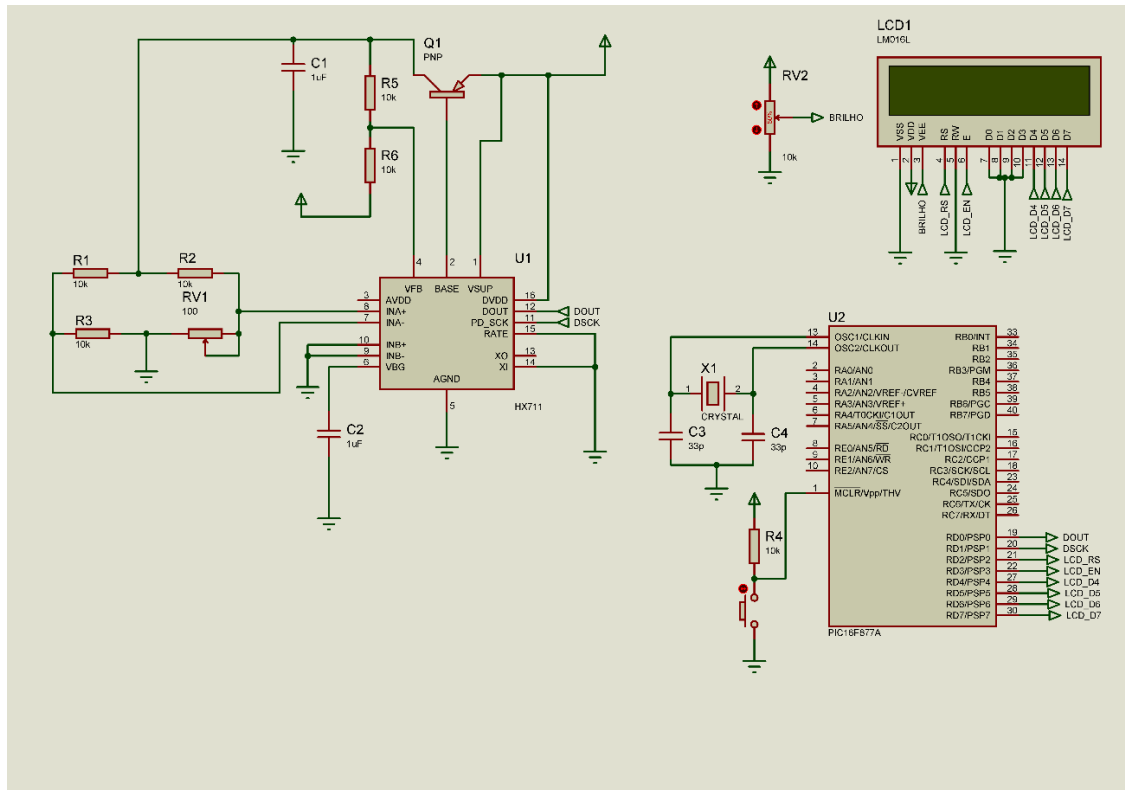


Figura 27: Circuito elétrico final no software Proteus.

Para o desenvolvimento do programa é necessário achar um valor de conversão, é utilizado uma massa de referência que no caso foi utilizado uma moeda. A moeda tem x gramas, foi mensurado o sinal digital em dois momentos, a balança com a massa e a balança sem a massa, através desse método foi possível calibrar a balança. O primeiro contato com o sistema foi o método demonstrado pela Figura 28, onde foi fixado a célula de carga com pedaços de fita crepe e adicionado uma moeda de 1 real sobre a célula de carga, esta moeda tem aproximadamente 7,9 gramas.

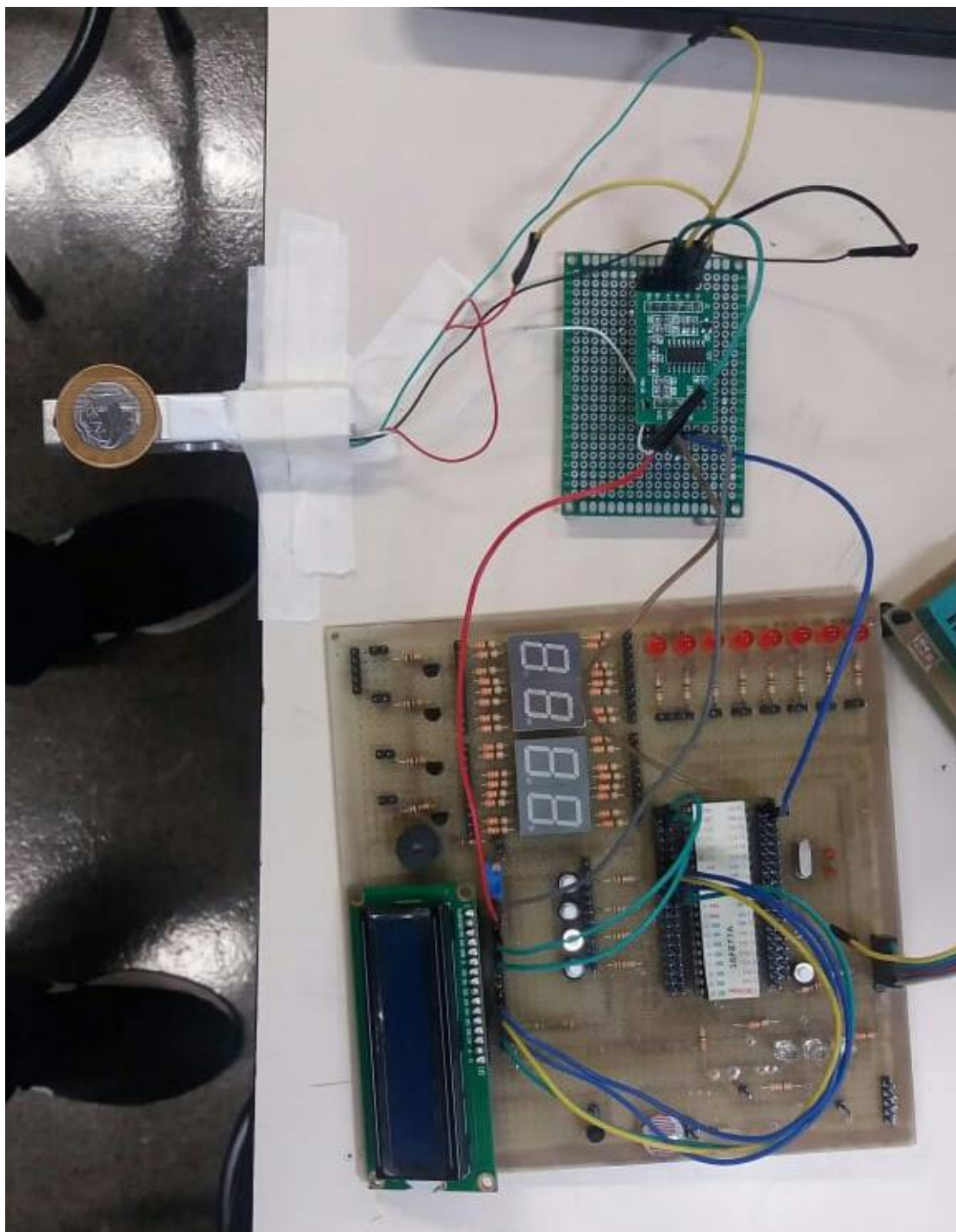


Figura 28: Fase de teste inicial e calibração da balança.

Após a evolução dos testes foi desenvolvido o nosso protótipo como mostra a Figura 29, durante os testes iniciais as medidas variavam muito, por esse motivo foi desenvolvido este protótipo que está limitado a massas de até 20Kg, onde a parte inferior célula de carga foi fixada por parafusos em uma superfície de madeira, e a parte superior foi fixada com um parafuso a uma superfície de acrílico.



Figura 29: Parte mecânica da balança.

Após os testes com este protótipo, a balança começou a ser caracterizada, foram utilizados vários objetos com pesos diferentes. As massas foram medidas em uma balança, como a da Figura 30, ela possui o mesmo tipo de célula de carga utilizado neste projeto.



Figura 30: Balança comercial de 10 Kg

Descobrimos o peso em gramas destes objetos, foi efetuado uma nova pesagem com o protótipo desenvolvido. Um desses objetos é um multímetro que pesa 208g, como mostra a Figura 31. No LCD que está presente no canto inferior da figura, é mostrado o valor decimal correspondente ao peso do multímetro.

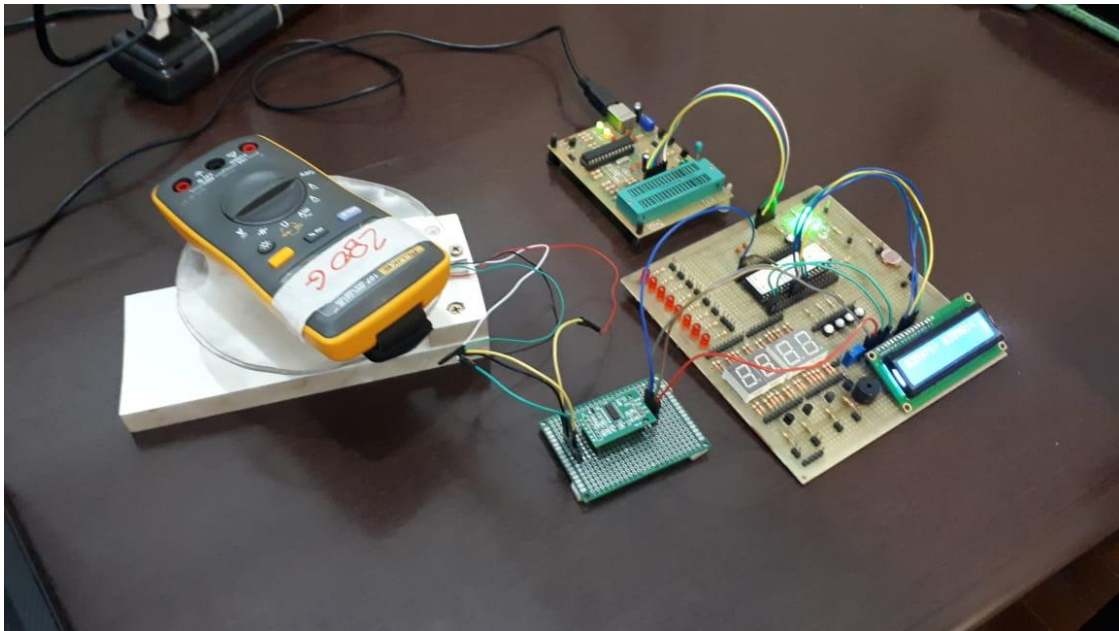


Figura 31: Caracterização da balança com o peso do multímetro.

Com o protótipo medindo o peso como um número decimal, foi efetuada uma lógica para descobrir a reta de operação da Figura 32, essa lógica consiste em utilizar uma balança comercial com o mesmo tipo de célula de carga para pesar várias massas, estas massas eram postas no protótipo e era armazenado o valor decimal dessa medida. Com esses 2 valores: peso em gramas e valor decimal, foi utilizado o Excel para traçar essa reta de operação da célula de carga. Sendo os valores de x em decimal e valores de y em gramas. Com essa reta descoberta, foi possível efetuar a medida dos objetos e mostrar em grandeza de gramas e não mais em números decimais.



Figura 32: Gráfico de operação do sistema.

Com esta curva de funcionamento, foi possível implementar a equação de reta no programa do microcontrolador e efetuar a medida de peso em gramas. Esse valor já convertido é utilizado em dois momentos, na escrita do LCD e escrita na porta serial do PIC. A porta serial utiliza dos pinos 25 e 26, sendo TX e RX respectivamente, no pino TX ele envia e no RX ele recebe um caractere ou uma sequência de caracteres em nível de tensão TTL (5V/0V), sendo 0 volts para o bit 0 e 5 volts para o bit 1. Neste ponto entra em ação o PL2303, ele converte este nível de tensão para o padrão USB (-12V/+12V), sendo -12 volts para o bit 1 e +12 volts para o bit 0, o conversor também pode fazer o caminho inverso indo do padrão USB para o TTL. A Figura 33 mostra de forma simplificada a conexão do conversor PL2303 com o microcontrolador, no canto esquerdo inferior está o conector USB que manda e recebe seus dados através dos pinos D+ e D- conectados com o conversor, o conversor transforma o padrão para UART e envia e recebe dados através dos pinos TXD e RXD.

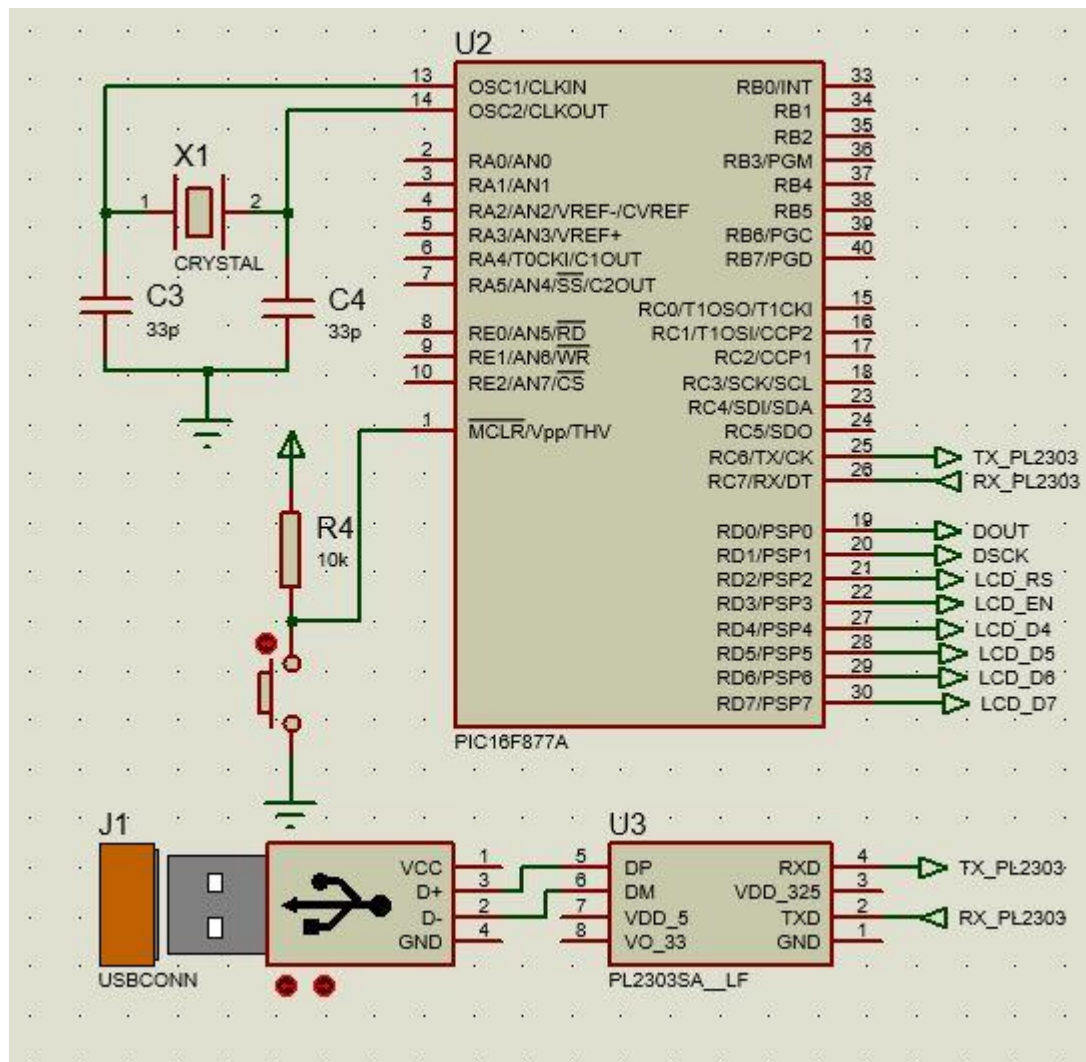


Figura 33: Conexão do conversor com PIC.

A partir deste momento o aplicativo desenvolvido em linguagem C#, através do Visual Studio, recebe e pode enviar dados para o hardware desenvolvido, este aplicativo tem 4 telas principais, a primeira a tela login onde são necessários digitar usuário e senha do sistema, a segunda onde é identificado o tipo do usuário, sendo eles administrador ou cliente do sistema, a terceira é a tela do administrador com suas opções e quarta a tela do usuário também com suas próprias opções.

A Figura 34 mostra a tela de login do sistema, nela o usuário deverá entrar com o usuário no campo Login e senha no campo Senha, ao clicar no botão Entrar é efetuada uma busca no banco de dados, caso o usuário e senha estejam corretos, o sistema avança para a próxima tela, caso contrário aparece uma

mensagem de erro avisando que as credenciais não estão válidas. E ao clicar no botão Sair a aplicação é instantaneamente fechada.

A interface de login do aplicativo, intitulada "Recycle AI", apresenta um fundo verde-escuro. No topo, há um ícone de reciclagem branco e o texto "Recycle AI" em branco. Abaixo, os campos "Login:" e "Senha:" são exibidos em branco. O campo de login contém o texto "Admin" e o campo de senha contém "*****". Na base da interface, há dois botões: "Entrar" com uma seta verde e "Sair" com uma seta verde.

Figura 34: Tela de Login do Aplicativo.

Com as credenciais corretas o sistema avança para a próxima tela (Figura 35), ela recebe o usuário da tela de login e verifica qual o tipo de usuário entrou no sistema, a aplicação possui 2 tipos usuários: administrador e cliente. Enquanto o sistema identifica o tipo de acesso, a barra de progresso é carregada para informar ao usuário que já está perto de entrar para sua tela de usuário. Esse tempo de espera é de 1,5 segundos.



Figura 35: Tela de transição da aplicação.

Com essa verificação efetuada e identificando que o usuário é o administrador do sistema, é aberta tela da Figura 36. Nela o responsável pelo sistema pode adicionar usuários, consultar e excluir usuários existentes, checar conexões com o banco de dados e comunicação serial e ler sobre o desenvolvimento do projeto clicando em Info.

Recycle AI

Usuário: Admin 12/11/2020 00:01

Cadastrar

Consultar

Excluir

Database

Conexão

Info

Login:

Senha:

Nome:

Sobrenome:

CPF:

Criar Usuario

Figura 36:Tela do administrador.

Caso o usuário identificado for de um cliente, é aberta a tela da Figura 37. Nela o cliente pode consultar o saldo do mês na opção Saldo, a economia dos meses do ano corrente na opção Economia Mensal, a economia dos meses de anos anteriores na opção Economia Anual e ler sobre o desenvolvimento do projeto clicando em Sobre.



Figura 37: Tela do cliente.

No Quadro 4 são listados os componentes utilizados no projeto e seus respectivos preços, os softwares utilizados são gratuitos, logo não entram no custo do projeto.

Item	valor R\$:
Pic 16F877A	R\$ 25,00
HX711+célula de carga	R\$ 37,00
LCD 16X2	R\$ 20,00
DRIVER USB PL2303	R\$ 12,00
PLACA 15X10MM	R\$ 13,00
TOTAL GASTO:	R\$ 107,00

Quadro 4: Lista dos materiais utilizados no projeto.

7. RESULTADOS

Com as partes desenvolvidas isoladamente, foi integrado o hardware com o software desenvolvidos no item 6, a Figura 38 está integração foi efetuado alguns testes com 3 pesos diferentes.

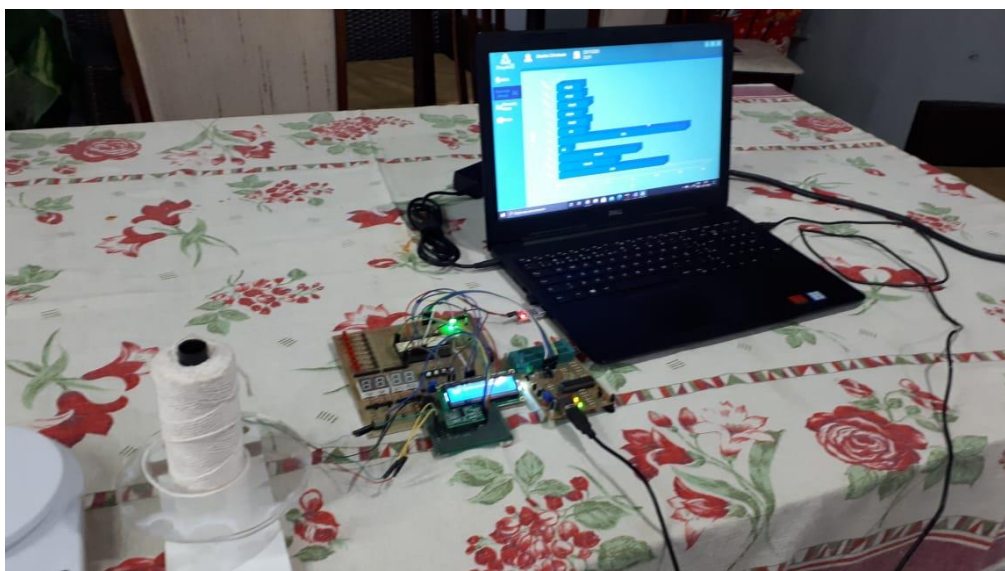


Figura 38: Hardware e Software integrados.

Com o sistema integrado, foram efetuadas três medidas com três objetos distintos, sendo eles: um rolo de barbante, uma caixa de som e uma panela de pressão, a Figura 39 mostra a medida efetuada com o barbante, tanto na balança que serviu de comparação, como no sistema desenvolvido.



Figura 39: Medida com o barbante.

A Figura 40 mostra a medida efetuada com a caixa de som, tanto na balança que serviu de comparação, como no sistema desenvolvido.



Figura 40: Medida com uma caixa de som.

A Figura 41 mostra a medida efetuada com a panela de pressão, tanto na balança que serviu de comparação, como no sistema desenvolvido.

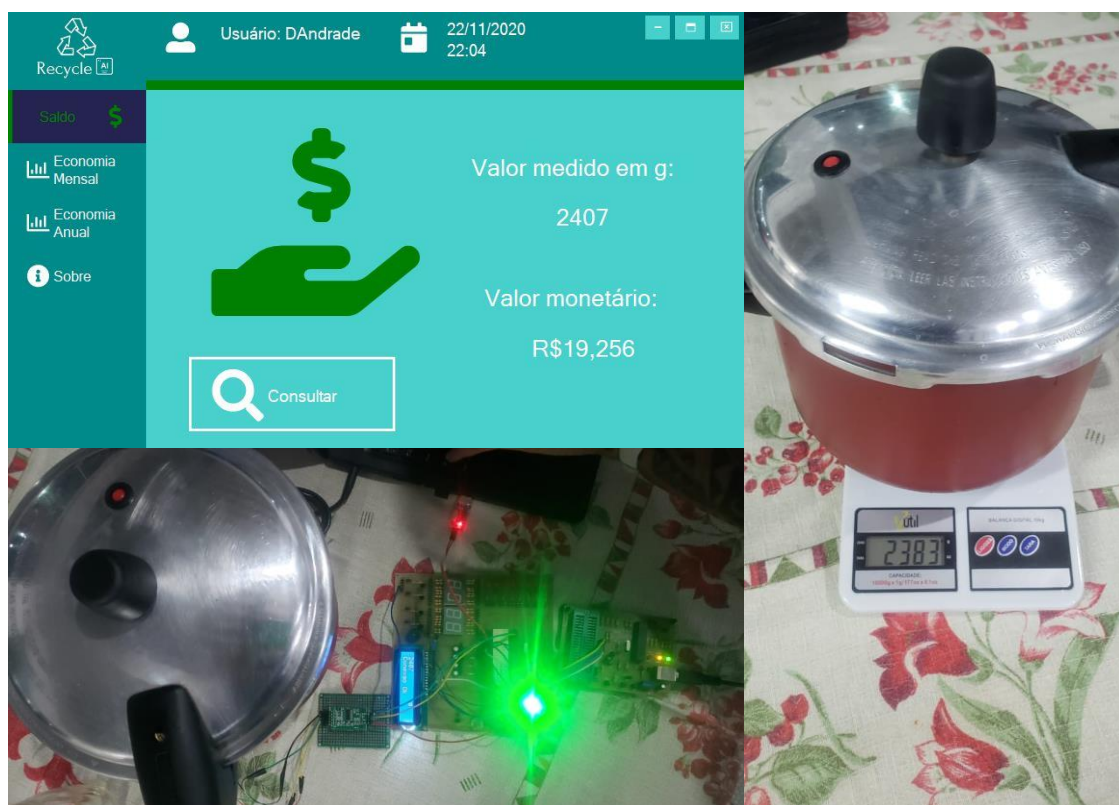


Figura 41: Medida com uma panela de pressão

Com estas medidas foi gerado o Quadro 5, nele estão expostos valores medidos com a balança de comparação, sistema desenvolvido e variação na mediada.

Objeto	Peso da Balança	Peso no Sistema	Variação em (%)
Rolo de Barbante	96 g	97 g	1,04
Caixa de som	776 g	784 g	1,03
Panela de pressão	2387 g	2407 g	0,83

Quadro 5: Levantamento de erro do sistema.

Com isto o erro de medida do sistema é de aproximadamente 1%, que é ocasionado pelo sensor utilizado.

8. FLUXOGRAMA DAS ETAPAS DO PROJETO

A Figura 42 demonstra as etapas do projeto e a ordem seguida para o desenvolvimento do projeto.



Figura 42: Fluxograma das etapas do projeto

9. OPERAÇÃO DO SISTEMA

O sistema foi projetado para operar de 0 até 20Kg, em um espaço curto. A premissa é a mesma, podendo se expandir pois é necessário adicionar mais células de cargas, e aumentar a área da superfície e a base do projeto. Existem células de carga com maiores precisões, porém são necessárias células de carga com o range maior, até a própria precisão pode ser alterada, porém o custo é maior, quanto maior a precisão da balança maior é o custo do projeto.

10. MODELO DE NEGÓCIO

Como toque final do projeto, foi idealizado um modelo de negócio, o modelo utilizado é o mapa canvas (Figura 43). Nele estão expressos todos os aspectos que devem se levar em consideração, para o protótipo se aproximar de um produto rentável. São 9 aspectos sendo eles: parcerias chaves, atividades chaves, recursos chaves, proposta de valor, relacionamento, segmentos de cliente, canais, estrutura de custos e fluxo de receitas.



Figura 43: Mapa Canvas.

11. EQUIPAMENTOS SEMELHANTES

Hoje em dia tem algumas empresas que trabalham com esse sistema que estamos desenvolvendo, uma delas é a “Retorna Machine”. Essa empresa tem máquinas em diversos condomínios e trabalham com bonificação em troca de resíduos sólidos. A máquina recebe diversos resíduos com código de barras como mostra a Figura 44, e dessa maneira conseguem ver o material que foi depositado e converter em pontos para ser utilizado com parceiros. Mas esse projeto trabalha com código de barras, o nosso trabalha mensurando o peso do alumínio que será depositado no recipiente. No sistema deles cada material tem uma pontuação [3].

VIDRO	AÇO	ALUMÍNIO
Embalagens de bebida e de alimento até 2,5L	Embalagens de bebida e de alimento até 2,5L	Embalagens de bebida até 2,5L *Com exceção dos lácteos
		
5 PONTOS	10 PONTOS	15 PONTOS

Figura 44: Bonificação de alguns dos itens triciclo.

Esse sistema está implementado em diversos locais e conta com bastante parceiros. Esse sistema conta com um aplicativo como ilustra a Figura 45, com a mesma ideia do que desenvolvemos.



Figura 45: Retorna Machine.

12. CONCLUSÕES FINAIS

O objetivo geral deste projeto foi o desenvolvimento de uma balança para validar o hardware, que está sendo desenvolvido. A balança irá medir a massa do alumínio, e irá avisar o usuário o valor da bonificação obtida por ele. E esse teste se concluiu com êxito, porém agora é adicionar sensores, e o armazenamento e processamento de dados, para aumentar a confiabilidade do sistema. Deixamos o hardware pronto, que pode ser adaptado para operar com maiores precisões e até mesmo um range maior de massa.

13. TRABALHOS FUTUROS

A ideia do projeto futuramente é transformar em um plano de negócio e para isso precisamos melhorar alguns pontos que já estão sendo discutidos. O primeiro é adicionar um sensor para validar o metal, se é alumínio mesmo por exemplo, para evitar possível fraude, através de um sensor indutivo (Figura 46) iremos validar o material que será adicionado na balança. E embutido com o sensor um sistema mecânico, como por exemplo uma janela, que se abrirá quando o usuário for se conectar e conforme o alumínio será adicionado o sensor irá validar o material que está sendo depositado. Precisaremos desenvolver um aplicativo mobile, focado nos sistemas operacionais Android e Ios.



Figura 46: Sensor indutivo.

Futuramente pretendemos abranger uma gama de equipamentos igual ao Retorna Machine, mas para isso será necessário parceiros, pois com o modelo atual que é com alumínio fica mais rentável a bonificação do usuário devido ao alumínio ter um valor de mercado melhor que o papelão e vidro, matérias que futuramente podem ser coletadas.

14.REFERÊNCIAS

- [1] HISTÓRIA da Balança. *In: História da Balança*. Contagem/MG: RF Balanças e Automação, 23 fev. 2015. Disponível em: <http://www1.rfbalancas.com.br/?p=2021>. Acesso em: 8 set. 2020.
- [2] RECICLAR Alumínio. *In: Reciclar Alumínio*. [S. l.]: Portal São Francisco, 16 mar. 2016. Disponível em: <https://www.portalsaofrancisco.com.br/meio-ambiente/reciclar-aluminio>. Acesso em: 11 set. 2020.
- [3] PROGRAMA Triciclo de Fidelidade Retorna Machine. [S. l.]: Triciclo SOLUÇÕES SUSTENTAVÉIS, 2015. Disponível em: <https://www.retornamachine.com.br/>. Acesso em: 15 set. 2020.
- [4] AUTOCORE ROBÓTICA. **Conversor Usb-Serial Ttl PL2303**. [S. l.]: AUTOCORE ROBÓTICA, 2020. Disponível em: <https://www.autocorerobotica.com.br/produto/adaptador-usb-serial-ttl-pl2303.html#:~:text=O%20Conversor%20Usb%2DSerial%20Ttl%20PL2303%20%C3%A9%20uma%20placa%20de,5V%20o%20que%20lhe%20permite>. Acesso em: 25 out. 2020.
- [5] (PROLIFIC TECHNOLOGY INC. **PL-2303 Edition USB to Serial Bridge Controller Product Datasheet**. 1.6. ed. rev. e atual. Nan Kang, Taipei 115, Taiwan, R.O.C.: [s. n.], 2005. 18 p. v. 1. Disponível em: <https://www.estudioelectronica.com/wp-content/uploads/2018/10/PL2303.pdf>. Acesso em: 17 set. 2019.
- [6] SHENZHEN EONE ELECTRONICS CO. **Specification for LCD Module 1602A-1**. 1.2. ed. [S. l.: s. n.], 20-. 12 p. v. 1. Disponível em: <https://www.openhacks.com/uploadsproductos/eone-1602a1.pdf>. Acesso em: 17 set. 2019.
- [7] MICROCHIP TECHNOLOGY. **PIC16F87XA**: 28/40/44-Pin Enhanced Flash Microcontrollers. [S. l.: s. n.], entre 2001 e 2013. 234 p. Disponível em: <http://ww1.microchip.com/downloads/en/DeviceDoc/39582C.pdf>. Acesso em: 2 out. 2019.

15. ANEXOS

15.1 Código Principal do PIC:

```
1: ////////////////////////////////// BLOCO 1 //////////////////////////////////
2: /*
3:     Projeto: Trabalho de Conclusão de Curso
4:     Autores: Danrlei, Diego e Guilherme
5:     Versão: 34 04/11/2020
6:
7:     Descrição:
8:
9:     Este projeto tem como objetivo medir a grandeza de gramas, converter
10: esse valor analógico para digital, escrever no LCD este valor em gramas e
11: enviar via comunicação serial esse valor.
12: Depois esse valor será tratado no aplicativo desenvolvido pelo Visual
13: Studio, que foi desenvolvido em C#.
14:
15:     Material Utilizado:
16:
17:     PIC: PIC16F877A / 8MHZ
18:     IDE: mikroC PRO for PIC v.7.5.0
19:     GRAVADOR : PIC Kit 2
20:
21:     Periféricos: Comunicação Serial.
22:
23:     Conexões externas: LCD ( 16x2 backlight azul ) ;
24:                         Cabos (Tipo Arduino para ligar as conexões );
25:
26:                         Pinos de entrada e saída;
27:                         Conversor USB/TTL PI2303.
28:
29:     Diretivas de conexões:
```

```

30:    LCD_RS      pino RD2 --> RS
31:    LCD_EN      pino RD3 --> EN
32:    LCD_D7      pino RD7 --> D7
33:    LCD_D6      pino RD6 --> D6
34:    LCD_D5      pino RD5 --> D5
35:    LCD_D4      pino RD4 --> D4
36:    TX          pino RC6 --> TX
37:    RX          pino RC7 --> RX
38:    ADD0        pino RD0 --> Pino de controle da leitura AD do HX711
39:    ADSK        pino RD1 --> Pino de envio o valor convertido em 24 bits
40: */
41: ////////////////////////////////////////////////// FIM DO BLOCO 1 //////////////////////////////////////
42:
43: ////////////////////////////////////////////////// BLOCO 2 //////////////////////////////////////
44: sbit LCD_RS_Direction at TRISD2_bit; // Seleciona os pinos que serão
45: sbit LCD_EN_Direction at TRISD3_bit; // utilizados pela rotina do
46: sbit LCD_D7_Direction at TRISD7_bit; // LCD como I/O.
47: sbit LCD_D6_Direction at TRISD6_bit;
48: sbit LCD_D5_Direction at TRISD5_bit;
49: sbit LCD_D4_Direction at TRISD4_bit;
50:
51: sbit LCD_RS at RD2_bit;           // Direciona os lables para os bits
52: sbit LCD_EN at RD3_bit;           //
53: sbit LCD_D7 at RD7_bit;           // Obs: Caso não sejam feitas essas diretivas
54: sbit LCD_D6 at RD6_bit;           // as funções do LCD, não funcionam, bem
55: sbit LCD_D5 at RD5_bit;           // como não adicionar a biblioteca do LCD e
56: sbit LCD_D4 at RD4_bit;           // conversões.
57:
58: unsigned char i;                  // variável de movimentação
59: unsigned long Count, Count_Med=0, Count_Peso = 0, Diferenca;
60: char Controle=0;                  // Variável de Controle de escrita no LCD
61: char peso[12];                    // vetor string do peso
62: char receive = '0';               // Variável char que recebe caractere da
63:                                   // Serial
64: ////////////////////////////////////////////////// FIM DO BLOCO 2 //////////////////////////////////////
65:
66: ////////////////////////////////////////////////// BLOCO 3 //////////////////////////////////////

```

```

67: unsigned long leitura_peso()    // Função que efetua leitura do peso
68: {                                //
69:     PORTD.RD0=1;                // ADD0=1
70:     PORTD.RD1=0;                // ADSK=0
71:     Count=0;                    // variável de contagem zerada
72:     while(PORTD.RD0);           // AD em progresso
73:     for(i=0;i<24;i++)           // 24 bits
74:     {
75:         PORTD.RD1=1;            // ADSK=1
76:         Count=Count<<1;         // Desloca o bit 1 para a esquerda
77:         PORTD.RD1=0;            // ADSK=0
78:         if(PORTD.RD0) Count++;  // incrementa 1 caso ADD0==1
79:     }
80:     PORTD.RD1=1;                // ADSK=1
81:     Count=Count^0x800000;       // Fazendo lógica XOR com o valor decimal
82:     PORTD.RD1=0;                // ADSK=0
83: }
84: void escrita_peso()             // Função que calcula e escreve no LCD
85: {
86:     Diferenca = 8454337 - Count; // Calculando a diferença de valor decimal
87:     Count_Peso = (0.008*Diferenca) + 24.0649 - 24; // Implantando curva
88:                                     // do projeto
89:     if(Count_Peso < 2 || Count_Peso > 20000) // Limitando o range de medida
90:     {                                // e identificando o zero
91:         Lcd_Out(1,13,"g");          // Escrevendo na col. 13 da linha 1
92:         Lcd_Out(1,1," 0 ");         // Escrevendo o valor 0
93:     }
94:     else
95:     {
96:         LongtoStr(Count_Peso,peso); // conversão de Long para Char
97:         ltrim(peso);                 // Tirando zeros a esquerda
98:         Controle++;                  // Incrementando variável de
99:         Delay_ms(500);               // Espera de 500ms
100:        LCD_Out(1,1," ");             // Limpando a linha do LCD
101:        Lcd_Out(1,1,peso);            // LCD mostra o peso medido
102:        Lcd_Out(1,13,"g");           // Escrevendo a letra g
103:    }
104: }
105: void Escrita_Leitura_Uart()        // Escrita do peso via serial

```

```

106: {
107:   if(UART1_Data_Ready()==1)      // Se receber algum Char pela serial
108:   {
109:     receive = UART1_Read();      // Atribuindo valor à variável
110:     if(receive == '1')           // Se igual a 1 execute os comandos
111:     {
112:       Lcd_Out(2,1,"Conexao Ok"); // Escrevendo que a conexão aconteceu
113:       Uart1_Write(peso[0]);      // Escrevendo a pos. 0 do vetor peso
114:       Uart1_Write(peso[1]);      // Escrevendo a pos. 1 do vetor peso
115:       Uart1_Write(peso[2]);      // Escrevendo a pos. 2 do vetor peso
116:       Uart1_Write(peso[3]);      // Escrevendo a pos. 3 do vetor peso
117:       Uart1_Write(peso[4]);      // Escrevendo a pos. 4 do vetor peso
118:     }
119:   }
120: }
121: ////////////////////////////////////////////////// FIM DO BLOCO 3 //////////////////////////////////////
122:
123: ////////////////////////////////////////////////// BLOCO 4 //////////////////////////////////////
124: void main()                      // Função principal
125: {
126:   TRISD=1;                      // Pino RC0 entrada (ADD0) e Pino RC1 saída (ADSK)
127:   PORTD=0;                      // Pinos em nível baixo
128:   Lcd_Init();                   // Inicializa do LCD
129:   Lcd_Cmd(_Lcd_Clear);         // Limpando LCD
130:   Lcd_Cmd(_Lcd_Cursor_Off);    // Desligando cursor do LCD
131:   Uart1_Init(9600);            // Iniciando comunicação serial com Baudrate
132:                                // de 9600
133:   while(1)                     // Loop infinito
134:   {
135:     leitura_peso();             // chama a função de leitura do peso
136:     escrita_peso();             // Chama a função de escrita do peso no LCD
137:     Escrita_Leitura_Uart();     // Escrevendo Valor do peso na COM Serial
138:   }
139: }
140: ////////////////////////////////////////////////// FIM DO BLOCO 4 //////////////////////////////////////

```

15.2 Código de Caracterização da célula de carga:

```
1: ////////////////////////////////// BLOCO 1 //////////////////////////////////
2: /*
3:     Projeto: Caracterização da célula de carga
4:     Autores: Danrlei, Diego e Guilherme
5:     Versão: 21 04/11/2020
6:
7:     Descrição:
8:
9:     Este código tem como finalidade efetuar medidas de peso em valor
10: decimal, onde estes valores serão utilizados para traçar a curva de operação
11: célula de carga.
12:
13:     Material Utilizado:
14:     PIC: PIC16F877A / 8MHZ
15:     IDE: mikroC PRO for PICv.7.5.0
16:     GRAVADOR : PIC Kit 2
17:
18:     Conexões externas: LCD ( 16x2 backlight azul ) ;
19:     Cabos ( Tipo Arduino para ligar as conexões );
20:     Pinos de entrada e saída;
21:
22:     Diretivas de conexões:
23:
24:     LCD_RS pino RD2 --> RS
25:     LCD_EN pino RD3 --> EN
26:     LCD_D7 pino RD7 --> D7
27:     LCD_D6 pino RD6 --> D6
28:     LCD_D5 pino RD5 --> D5
29:     LCD_D4 pino RD4 --> D4
30:     ADD0 pino RD0 --> Pino de controle da leitura AD do HX711
31:     ADSK pino RD1 --> Pino de envio o valor convertido em 24 bits
32: */
```

```

33: ////////////////////////////////// FIM DO BLOCO 1 //////////////////////////////////
34:
35: ////////////////////////////////// BLOCO 2 //////////////////////////////////
36: sbit LCD_RS_Direction at TRISD2_bit; // Seleciona os pinos que serão
37: sbit LCD_EN_Direction at TRISD3_bit; // utilizados pela rotina do LCD como
38: sbit LCD_D7_Direction at TRISD7_bit; // I/O.
39: sbit LCD_D6_Direction at TRISD6_bit; //
40: sbit LCD_D5_Direction at TRISD5_bit; //
41: sbit LCD_D4_Direction at TRISD4_bit; //
42:
43: sbit LCD_RS at RD2_bit;           // Direciona os lables para os bits
44: sbit LCD_EN at RD3_bit;           //
45: sbit LCD_D7 at RD7_bit;           // Obs: Caso não sejam feitas essas diretivas
46: sbit LCD_D6 at RD6_bit;           // as funções do LCD, não funcionam, bem
47: sbit LCD_D5 at RD5_bit;           // como não adicionar a biblioteca do LCD
48: sbit LCD_D4 at RD4_bit;           // e conversões.
49:
50: unsigned char i;                  // variável de movimentação
51: unsigned long Count, Count_Med=0, Count_Med2=0; // variável do peso
52: char Controle=0, Controle2=0;    // variáveis de controle das médias
53: char peso[12];                   // string do peso
54: char peso_MED [12];              // string de média do peso
55: char peso_MED2[12];              // string de média da média do peso
56: ////////////////////////////////// FIM DO BLOCO 2 //////////////////////////////////
57:
58: ////////////////////////////////// BLOCO 3 //////////////////////////////////
59: unsigned long leitura_peso()      // Função que efetua leitura do peso
60: {
61:     while(Controle2!=10)          // Controle 2 diferente de 10 exec. o bloco
62:     {
63:         PORTD.RD0=1;              // ADD0=1
64:         PORTD.RD1=0;              // ADSK=0
65:         Count=0;                  // variável de contagem zerada
66:         while(PORTD.RD0);          //AD em progresso
67:         for(i=0;i<24;i++)          // 24 bits
68:         {
69:             PORTD.RD1=1;           // ADSK=1
70:             Count=Count<<1;        // Desloca o bit 1 para a esquerda

```

```

71:   PORTD.RD1=0;           // ADSK=0
72:   if(PORTD.RD0) Count++; // incrementa variável
73: }
74:   Controle++;           // Incrementa variável da primeira média
75:   PORTD.RD1=1;           // ADSK=1
76:   Count=Count^0x800000; // Fazendo lógica XOR com o valor decimal
77:   PORTD.RD1=0;           // ADSK=0
78:   LongtoStr(Count,peso); // conversão do valor decimal em g
79:   ltrim(peso);           // Tirando zeros a esquerda
80:   Delay_ms(250);         // Espera de 250 ms
81:   Count_Med = Count_Med + Count; // Somador
82:   Lcd_Out(1,1,peso);      // LCD mostra o peso
83:   if(Controle==50)        // Se Controle igual a 50 executa
84:   {
85:     Controle2++;          // Incrementa variável da segunda média
86:     LongtoSTR(Count_Med/50, peso_MED); // Conversão Long/String
87:     ltrim(peso_MED);      // Tirando zeros a esquerda
88:     Lcd_Out(1,9,peso_MED); // Escrevendo na coluna 9 da linha 1
89:
90:     Count_Med2 = Count_Med2 + Count_Med; // Somador
91:
92:     if(controle2==10)     // Se Controle2 igual a 10 executa o bloco
93:     {
94:       LongtoSTR(Count_Med2/500, peso_MED2); // Conversão Long/String
95:       ltrim(peso_MED2);   // Tirando zeros a esquerda
96:       LCD_Out(2,1,peso_MED2); // Escrevendo na coluna 1 da linha 2
97:     }
98:     Count_Med=0;          // Zerando variável de controle
99:     Controle =0;          // Zerando variável de controle
100:   }
101: }
102: }
103: // FIM DO BLOCO 3 //
104:
105: // BLOCO 4 //
106: void main()
107: {
108:   TRISD=1; // Pino RC0 entrada (ADD0) e Pino RC1 saída (ADSK)
109:   PORTD=0; // Pinos em nível baixo

```

```
110:  Lcd_Init();                // Inicializa LCD
111:  Lcd_Cmd(_Lcd_Clear);        // Limpa LCD
112:  Lcd_Cmd(_Lcd_Cursor_Off);    // Desliga cursor
113:
114:  while(1) // Loop infinito
115:  {
116:      leitura_peso();           // chama a função de leitura do peso
117:  }
118: }
119: ////////////////////////////////// FIM DO BLOCO 4 //////////////////////////////////
```