

CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA
FATEC SANTO ANDRÉ
Tecnologia em Eletrônica Automotiva

DOUGLAS CAVALCANTE DE SOUZA
NATANAEL GONÇALVES DE SOUZA

OSCILOSCÓPIO DIDÁTICO VIA LABVIEW

Santo André – São Paulo

2017

CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA
FATEC SANTO ANDRÉ
Tecnologia em Eletrônica Automotiva

DOUGLAS CAVALCANTE DE SOUZA
NATANAEL GONÇALVES DE SOUZA

OSCIOSCÓPIO DIDÁTICO VIA LABVIEW

Monografia apresentada ao Curso de Tecnologia em Eletrônica Automotiva da FATEC Santo André, como requisito parcial para conclusão do curso em Tecnologia em Eletrônica Automotiva.

Orientador: Prof. Dr. Edson Caoru Kitani

Santo André – São Paulo

2017

FICHA CATALOGRÁFICA

S729o

Souza, Douglas Cavalcante de
Osciloscópio Didático via LabView / Douglas Cavalcante de
Souza, Natanael Gonçalves de Souza. - Santo André, 2017.
– 96f: il.

Trabalho de Conclusão de Curso – FATEC Santo André.
Curso de Tecnologia em Eletrônica Automotiva, 2017.

Orientador: Prof. Dr. Edson Caoru Kitani

1. Eletrônica. 2. Osciloscópio. 3. Software LabView. 4. Siste-
mas eletrônicos. 5. Usuário. 6. Interface. 7. Microcontrolador
PIC. I. Souza, Natanael Gonçalves de. II. Osciloscópio auto-
motivo via LabView.

621.389

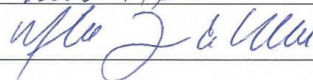
LISTA DE PRESENÇA

Santo André, 22 de Dezembro de 2017

LISTA DE PRESENÇA REFERENTE À APRESENTAÇÃO DO
TRABALHO DE CONCLUSÃO DE CURSO COM O TEMA:
“OSCILOSCÓPIO DIDÁTICO VIA LABVIEW
SEMESTRE DESTA U.E.

BANCAPRESIDENTE:
PROF. EDSON CAORU KITANIMEMBROS:
PROF. PAULO TETSUO HOASHI

PROF. MURILO ZANINI DE CARVALHO

**ALUNOS:**

DOUGLAS CAVALCANTE DE SOUZA



NATANAEL GONÇALVES DE SOUZA



AGRADECIMENTOS

Primeiramente a Deus por ter nos dado força, paciência e coragem para que mantivéssemos nossa determinação durante a execução deste trabalho.

Ao nosso orientador, Edson, pela paciência em nos ajudar a preencher as lacunas necessárias para poder dar prosseguimento ao projeto.

A *INDUSPRIME*, por disponibilizar espaço e equipamentos para teste.

Aos professores em geral, coordenação, colaboradores da Fatec Santo André, nossos amigos e familiares por estarem sempre conosco, nos apoiando em todos os momentos.

E a todos aqueles que direta ou indiretamente fizeram parte da nossa formação, o nosso muito obrigado.

RESUMO

A cada dia mais o uso do osciloscópio para a análise de sistemas eletrônicos tem se tornado indispensável devido a resposta que esse instrumento pode trazer ao operador, dispor dessa tecnologia para realizar medições e ter uma confiabilidade maior da análise requer um preço, e esse preço nem todos podem pagar, pensando nisso o projeto a seguir tratará do desenvolvimento de um osciloscópio de baixo custo para uso didático, baseado no *software LabVIEW*. Será utilizado um *hardware* principal em conjunto com outro *hardware* que desenvolveremos ao longo do trabalho. O *hardware* principal é usado nas aulas da FATEC e possui o microcontrolador PIC18F4550 da *microchip* e porta *USB* para comunicar com o computador. O *hardware* adicional condicionará o sinal para que o mesmo possa ser lido pelo microcontrolador e posteriormente o código fonte em C através do *software CCS C Compiler* para a conversão do sinal analógico para digital que será transferido via *USB* e será mostrado na tela do computador através do *software LabVIEW*, em que será desenvolvida a interface para o usuário.

Palavras chaves: *USB*, *LabVIEW*, PIC18F4550, Osciloscópio Didático.

ABSTRACT

Increasingly, the use of the oscilloscope for an analysis of electronic systems has become indispensable due to a response that this instrument can bring to the operator, to have this technology to perform measurements and to have greater reliability of the analysis requires a price, and this price not everyone can afford, thinking about it the following project will be a development of a low cost oscilloscope for didactic use, it was not the LabVIEW software. It will use Fatec's microcontroller hardware in conjunction with other developed hardwares. The main hardware is used in FATEC classes and has microchip PIC18F4550 microcontroller and UBS port to communicate with the computer. The additional hardware contains an order for the same microcontroller software as well as the C source code through the CCS C Compiler software for converting the analog to digital signal that is transferred via USB and will be displayed on the computer screen through the software LabVIEW, in which an interface is developed for the user.

Key words: *USB, LabVIEW, PIC 18F4550, Didactic Oscilloscope.*

LISTA DE ILUSTRAÇÕES

Figura 1 - Vista em corte de um tubo de raios catódicos.	16
Figura 2 - Tubo de raios catódicos	17
Figura 3 – Diagrama de Blocos de um Osciloscópio Analógico	18
Figura 4 – Onda dente de serra aplicada no sistema de deflexão horizontal	19
Figura 5 - Sinais sincronizado e não sincronizado	20
Figura 6 – Osciloscópio Analógico Tektronix 2225.....	20
Figura 7 - Diagrama de blocos de um osciloscópio digital	21
Figura 8 - Diagrama de blocos de um osciloscópio digital com conversores A/D e D/A	22
Figura 9 - Osciloscópio Digital Tektronix TBS1154	23
Figura 10 - Diagrama de Pinagem do PIC 18F4550	24
Figura 11 - Camadas de <i>software</i> e <i>hardware</i> que constituem o funcionamento da conexão do <i>USB</i> com um computador.....	28
Figura 12 - Etapas de eventos que ocorrem durante a conexão de um periférico a porta <i>USB</i>	29
Figura 13 - Tipos de Conectores <i>USB</i>	31
Figura 14 – Sequência para carregamento do <i>firmware</i> no Pic.....	33
Figura 15 – Configuração seguidor de tensão.....	34
Figura 16 - Pinagem de um amplificador Operacional	35
Figura 17 - Curva característica de transferência do Amplificador Operacional.....	36
Figura 18 - Circuito seguidor de tensão	36
Figura 19 – Teste do circuito seguidor de tensão.....	37
Figura 20 – Pinagem do CI LM358.....	37
Figura 21 – Saída do seguidor de tensão ao aplicar 2 VDC em sua entrada.....	39
Figura 22 – Saída do seguidor de tensão ao aplicar 5 VDC em sua entrada.....	40
Figura 23 - Sinal de Saída com Ganho Unitário e sinal de entrada 0.5V	40
Figura 24 - Sinal de Saída com ganho quatro	41
Figura 25 - Sinal de Saída com ganho dez	41
Figura 26 - <i>Front Panel</i> e <i>Block Diagram</i> de uma aplicação com <i>Labview</i>	47
Figura 27 - Diagrama de Blocos da comunicação da Aplicação com <i>Hardware</i>	48
Figura 28 - Leitura do Sinal com Ganho Unitário Apresentado na Tela do <i>Datalogger</i>	49

Figura 29 - Leitura do Sinal com Ganho Unitário Efetuada Pelo Multímetro	49
Figura 30 - Leitura do Sinal com Ganho Quatro Apresentado na Tela do <i>Datalogger</i>	50
Figura 31 - Leitura do Sinal com Ganho Quatro Efetuada Pelo Multímetro	50
Figura 32 - Teste da Leitura do Sinal Alternado Efetuado pelo <i>Datalogger</i>	51
Figura 33 - Sinal Alternado com Ganho 2	52
Figura 34 - Trecho da programação em <i>LabVIEW</i> que faz a atenuação	53
Figura 35 - Trecho da Programação em <i>LabVIEW</i> responsável por apresentar o sinal no gráfico.....	54
Figura 36 - Interface Gráfica para operação dos sinais.....	55
Figura 37 - Case Responsável pelo ganho do sinal	56
Figura 38 - Leitura e comparação da onda quadrada a 0.193Hz	57
Figura 39 – Leitura da Onda quadrada a 10.71Hz no <i>Datalogger</i>	58
Figura 40 - Leitura onda quadrada a 10.71Hz no Osciloscópio	59
Figura 41 – Sinal de 5Hz capturado pelo Osciloscópio Digital Via <i>LabVIEW</i>	60
Figura 42 – Sinal de 60Hz capturado pelo Osciloscópio Digital Via <i>LabVIEW</i>	60
Figura 43 – Interface do Osciloscópio didático no <i>LabVIEW</i>	61
Figura 44 – Leitura sinal a 60Hz com o osciloscópio Didático.....	62
Figura 45 - Comparativo das funções do osciloscópio digital com o osciloscópio didático	63

LISTA DE TABELAS

Tabela 1 – Sinal de saída teórico do seguidor de tensão.....	38
Tabela 2 – Sinal de saída Real do seguidor de tensão	38
Tabela 3 – Tabela <i>ASCII</i>	43

LISTA DE SIGLAS E ABREVIações

ADC	<i>Analog to Digital Converter</i> (Conversor Analógico-Digital)
CCS	<i>Custom Computer Service</i>
CDC	<i>Communications Device Class</i> (Classe de Comunicação do Dispositivo)
CRT	<i>Cathode Ray Tube</i> (Tubo de Raios Catódicos)
DLL	<i>Dynamic Link Library</i>
DVD	<i>Digital Versatile Disc</i> (Disco Digital Versátil)
GND	<i>Ground</i> [terra (eletricidade)]
HID	<i>Human Interface Device</i> (Dispositivo de Interface Humana)
Hz	Unidade de Frequência derivada do SI (Hertz)
IBM	<i>International Business Machines</i>
I/O	<i>In/Out</i> (entrada ou saída de um módulo eletrônico)
KB	<i>Kilobytes</i> (Quilobytes)
Kbps	<i>Kilo bits per second</i> (Kilobits por segundo)
LabVIEW	<i>Laboratory Virtual Instruments Engineering Workbench</i>
LED	<i>Light Emitting Diode</i> (Diodo Emissor de Luz)
LSB	<i>Least Significant Bit</i>
Mbps	<i>Mega bits per second</i> (Megabits por Segundo)
MCLR	<i>Master Clear</i> (Reset do PIC)
MHz	Unidade de frequência derivada do SI (Mega-hertz)
MSB	<i>Most Significant Bit</i>
MSD	<i>Mass Storage Device</i> (Dispositivo de Armazenamento em Massa)
NRZI	<i>Non Return to Zero Inverted</i>
PC	<i>Personal Computer</i>
PIC	<i>Programmable Integrated Circuit</i> (Circuito Integrado Programável)
PnP	<i>Plug and Play</i> (Ligar e Usar)
STR	Sistema de Tempo Real
USB	<i>Universal Serial Bus</i>
V	Unidade de tensão derivada do SI (volt)
VI	<i>Virtual Instrument</i> (Instrumento Virtual)

SUMÁRIO

1. INTRODUÇÃO	14
1.2 Motivação	15
1.3 Objetivo.....	15
2. FUNDAMENTAÇÃO TEÓRICA.....	16
2.1 Histórico do Osciloscópio.....	16
2.2.1 Princípio de Funcionamento do Osciloscópio Analógico	17
2.2.2 Princípio de Funcionamento do Osciloscópio Digital	20
2.2.3 Métodos de Amostragem nos osciloscópios digitais.....	22
2.3 Microcontrolador PIC18F4550	24
2.4 COMPILADOR CCS	25
2.5 PORTA USB	26
3. PROJETO	32
3.1 <i>Hardware</i> Principal	32
3.2 Desenvolvimento do <i>Hardware</i> Adicional	33
3.3 Desenvolvimento do Código Fonte em Linguagem C.....	42
3.4 Desenvolvimento da Interface no <i>LabVIEW</i>	46
4. RESULTADOS	57
5. CONSIDERAÇÕES FINAIS	64
5.1 Conclusão.....	64
5.2 Proposta de melhoria.....	65
6. REREFÊNCIAS BIBLIOGRÁFICAS	66
7 ANEXOS	69
7.1 Esquemático do <i>Hardware</i> Principal.....	77
7.2 Layout do <i>Hardware</i> Principal.....	78
7.3 Lista de Componentes do <i>Hardware</i> Principal.....	79
7.4 Esquemático do <i>Hardware</i> Adicional	80
7.5 <i>Layout</i> do <i>Hardware</i> Adicional	81

7.6 Lista de Componentes do <i>Hardware</i> Adicional	82
7.7 Código Fonte em C	83
7.8 Construção do software em LabVIEW	96

1. INTRODUÇÃO

O osciloscópio é um equipamento muito importante no mundo da eletrônica, pois através desse equipamento é possível identificar o comportamento de um sistema, e esse comportamento é analisado através de sinais que são captados e mostrados em sua tela. Dessa forma o operador pode verificar os sinais que estão sendo apresentados e se estes estão conforme o que se espera.

O osciloscópio é um equipamento de alto custo e muitos estudantes e profissionais, em especial os técnicos de manutenção eletrônica, têm a necessidade de dispor de um equipamento deste porte, mas devido ao custo elevado, adquirir esse produto acaba sendo uma dificuldade. Pensando nisso, o presente trabalho discutirá o desenvolvimento de um osciloscópio didático de baixo custo utilizando o *software LabVIEW*. O trabalho aqui apresentado foi realizado partindo do projeto realizado pelos autores Bruno e Roberto (SAMPAIO e JUNCKER, 2014), que tem como título “plataforma didática para aquisição de dados via interface *USB* e *LabVIEW* com *Bootoader* integrado”.

O trabalho realizado por esses autores teve algumas limitações quanto aos sinais captados, pois sinais em frequências superiores a 10 Hz produziam muitas distorções ao ser visualizado no *Front Panel* do *LabVIEW*. E uma das propostas futuras sugeridas era a melhoria dessa aquisição para sinais em frequência maiores do que 10Hz. Esse é um dos propósitos do trabalho que será realizado, o desenvolvimento de osciloscópio didático que consiga ler sinais de até 1kHz. Não é de hoje que vem a ideia de criar um osciloscópio de baixo custo. Pesquisando na *internet* podem ser encontradas diversas maneiras de desenvolver um osciloscópio.

1.2 Motivação

A motivação para a realização desse trabalho vem da dificuldade financeira que muitos profissionais e estudantes possuem de adquirir um osciloscópio. Devido a complexidade tecnológica presente nesse equipamento, o seu valor acaba sendo alto, e poucas são as pessoas que podem estar adquirindo o produto. Levando em conta essa problemática surgiu a motivação de desenvolver um osciloscópio didático de baixo custo para que estudantes e profissionais possam ter disponível uma plataforma para poder realizar seus experimentos. Vale ressaltar aqui que a ideia de projetar um osciloscópio baseado no *software LabVIEW* vem do trabalho proposto pelos autores Roberto e Bruno (SAMPAIO e JUNCKER, 2014) em seu trabalho de conclusão de curso, que tem como título “plataforma didática para aquisição de dados via interface *USB* e *LabVIEW* com *Bootloader* integrado”. Partindo da ideia proposta pelos autores tivemos o interesse em trabalhar nesse projeto tendo em vista a nossa motivação de fornecer aos estudantes e profissionais uma ferramenta para fins didáticos.

1.3 Objetivo

Conforme as motivações discutidas na subseção 1.2, entende-se que o objetivo principal do presente trabalho foi o desenvolvimento de um osciloscópio didático, que foi confeccionado a partir da ferramenta computacional *LabVIEW*, onde foi realizado interações através da porta de comunicação *USB* (*Universal Serial Bus*) entre o computador e a plataforma, que condiciona os sinais para que possa ser realizada a leitura dos sinais.

2. FUNDAMENTAÇÃO TEÓRICA

Esta seção traz conceitos que foram importantes ao longo do desenvolvimento do projeto. O conjunto de informações inseridas nesta seção traz ao leitor, uma base para que possa compreender os passos que foram seguidos durante o trabalho.

2.1 Histórico do Osciloscópio

Em 1897 o físico alemão **Karl Ferdinand Braun** construiu o tubo de raios catódicos (TRC) em inglês (CRT) *cathode ray tube*, que consistia em um dispositivo que produzia imagem quando uma quantidade de elétrons incidia sobre a tela recoberta por fósforo. O feixe de elétrons que atingia a tela poderia ser deflexionado quando submetido a um campo elétrico ou magnético, o que possibilitava a projeção de diversas imagens na tela. O tubo de raios catódicos foi uma invenção de grande valia para a eletrônica, uma vez que os monitores dos computadores, televisões e osciloscópios foram produzidos partindo de seu princípio de funcionamento. A Figura 1 abaixo apresenta uma ideia da construção de um tubo de raios catódicos:

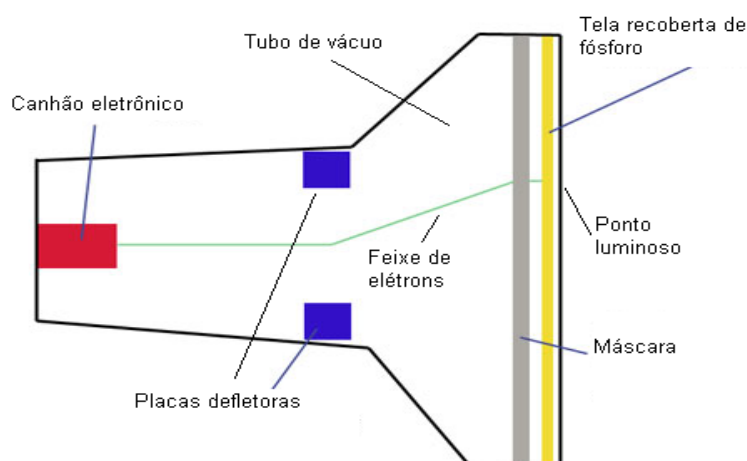


Figura 1 - Vista em corte de um tubo de raios catódicos (Disponível em <<http://www.newtoncbraga.com.br/index.php/almanaque/865-tubo-de-raios-catodicos.html>>).

Observando a Figura 1 acima e ligando o conceito com a Figura 2, nota-se uma representação mais real do osciloscópio. Vale ressaltar que essa representação se

baseia em um modelo de osciloscópio analógico, cujo princípio se baseia no tubo de raios catódicos.

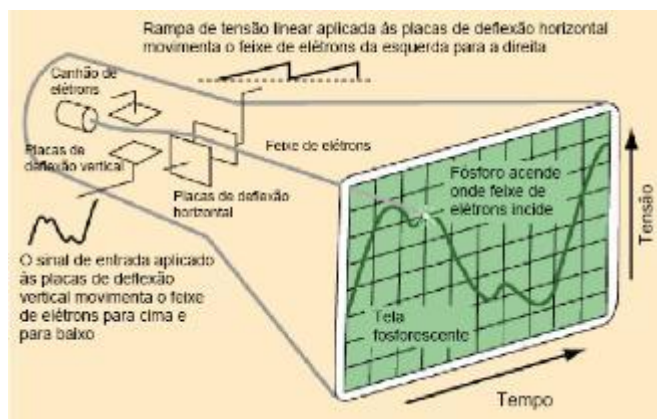


Figura 2 - Tubo de raios catódicos (Disponível em <<http://www.ebah.com.br/content/ABAAAV7EAL/medicoes-com-osciloscopio>>).

2.2.1 Princípio de Funcionamento do Osciloscópio Analógico

Partindo da subseção 2.1 onde foi discutido um breve histórico do osciloscópio, pode-se agora tratar do princípio de funcionamento do mesmo. Antes de tratar propriamente do princípio de funcionamento do osciloscópio, vale informar que o presente trabalho traz diversos termos técnicos. Pensando nisso foi acrescentado um anexo que contém uma espécie de minidicionário, caso aja alguma dúvida referente a algum termo técnico, basta conferir o capítulo de anexos. Nesse anexo estará contido os termos técnicos com as devidas explicações, que foram extraídos da obra, Eletrônica Total nº156 (2013).

Nos osciloscópios analógicos o sinal é produzido através de feixes de elétrons que incidem sob a tela fosforescente. O sinal de tensão adquirido passa primeiramente pelo “sistema vertical” do osciloscópio. Esse sistema possui duas placas defletoras na horizontal, mas que permite o desvio do feixe de elétrons na vertical. No “sistema vertical”, dependendo de como a escala é ajustada, o sinal terá a amplitude reduzida ou aumentada através de circuitos atenuadores e amplificadores. Em seguida o sinal passa pelo “sistema horizontal” do osciloscópio, que permitirá a

visualização do sinal no domínio do tempo, isso é conseguido através de duas placas defletoras localizadas na vertical, mas que permite o desvio do feixe de elétrons na horizontal (ALVES, 1998). A Figura 3 mostra um diagrama de blocos simples onde se transmite a ideia de como o sinal é desenhado na tela de um osciloscópio analógico:

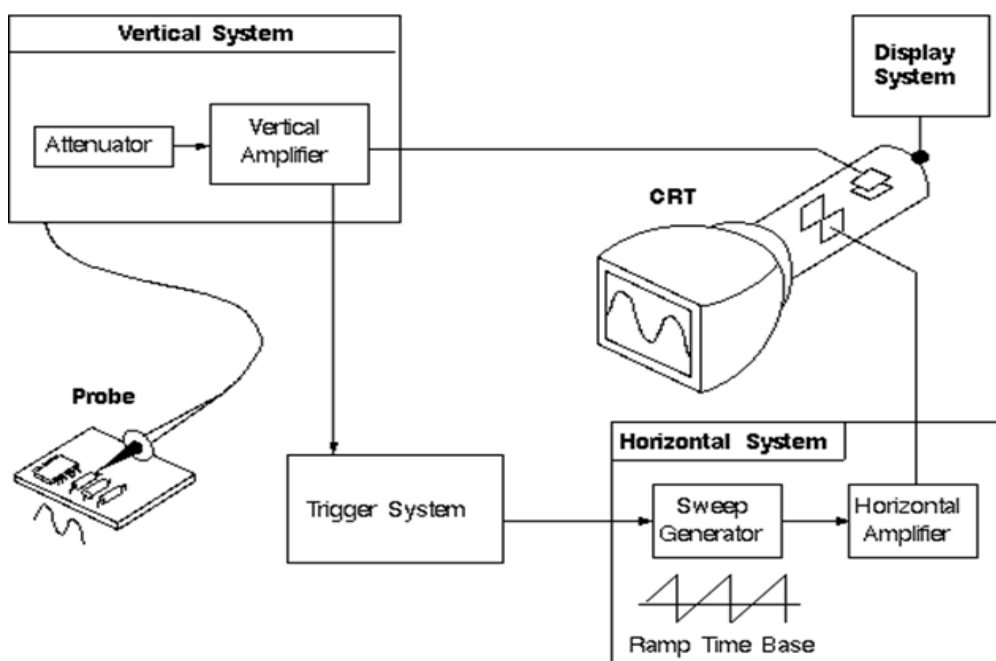


Figura 3 – Diagrama de Blocos de um Osciloscópio Analógico (Disponível em < <http://forums.anandtech.com/showthread.php?t=2213348>>).

Para se ter o deslocamento do feixe de elétrons na horizontal ou seja, da esquerda para a direita para se ter uma visualização do sinal no domínio do tempo, é necessário aplicar uma tensão em forma de rampa (Figura 4), de maneira que quando o sinal aplicado for o mais baixo, significará que o feixe de elétrons estará no extremo esquerdo da tela do osciloscópio, enquanto que quando a tensão atingir seu valor máximo, significará que o feixe de elétrons estará no extremo direito da tela do osciloscópio.

Dessa forma o feixe de elétrons se deslocará da esquerda para a direita conforme o valor de tensão que estará sendo aplicado. O tempo de varredura do sinal dependerá da inclinação do sinal de tensão em forma de rampa que está sendo aplicado. O tempo de varredura significa o tempo que o feixe de elétrons leva para ir de um extremo a outro (ALVES, 1998).

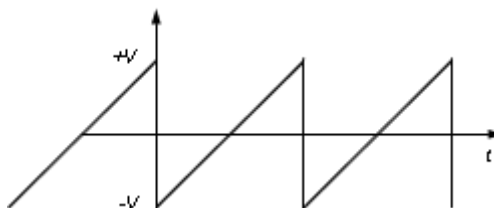


Figura 4 – Onda dente de serra aplicada no sistema de deflexão horizontal (base de tempo)
(Disponível em <<http://www.ifsc.usp.br/~strontium/Teaching/Material2010-2%20FFI0106%20Lab-Fisicall/09-InstrumentosdeMedidasEletricas-II.pdf>>).

A capacidade temporal de fosforescência que a tela revestida de fósforo possui é baixa, ou seja o sinal formado pelos feixes de elétrons ao tocar a tela revestida de fósforo aparecerá em um curto período de tempo, de forma que não será possível enxergar a forma de onda no osciloscópio. Logo, se pode concluir que não será possível realizar as medições se for realizada apenas uma varredura. Portanto, o processo de varredura deverá ser repetido diversas vezes de forma que o osciloscópio mantenha a imagem estabilizada na tela. Para isso o sinal que está sendo medido deve ser periódico e, além disso, o feixe de elétrons deverá percorrer o mesmo caminho toda vez que for iniciado um novo processo de varredura (ALVES, 1998).

Para que se possa ter uma imagem estabilizada do sinal medido é preciso que exista um sincronismo entre a base de tempo que determina o sinal em forma de rampa e o sinal que se deseja medir. Para que o feixe de elétrons faça sempre o mesmo percurso é preciso ter um sistema que controla o momento exato em que será gerado o sinal em forma de rampa, e esse sistema é conhecido como *trigger* (ALVES,

1998). O *trigger* permite que o sinal se mantenha estabilizado, o que facilita a visualização do sinal. Pode-se ver na figura 5 um sinal sincronizado e um não sincronizado:

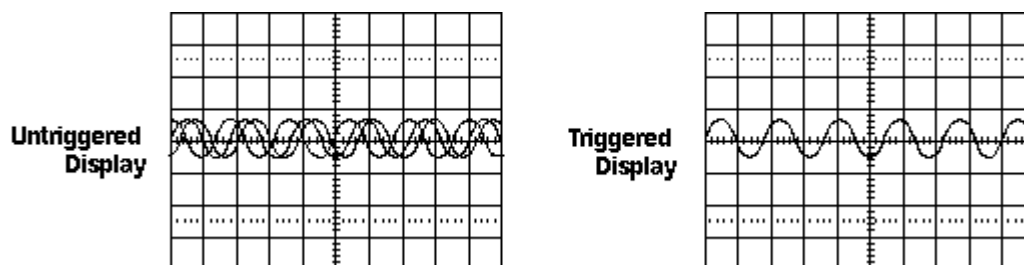


Figura 5 - Sinais sincronizado e não sincronizado (Disponível em < <http://www.ceset.unicamp.br/~leobravo/TT%20305/O%20Osciloscopio.pdf>>).

Pode-se ver na Figura 6 abaixo um osciloscópio analógico.

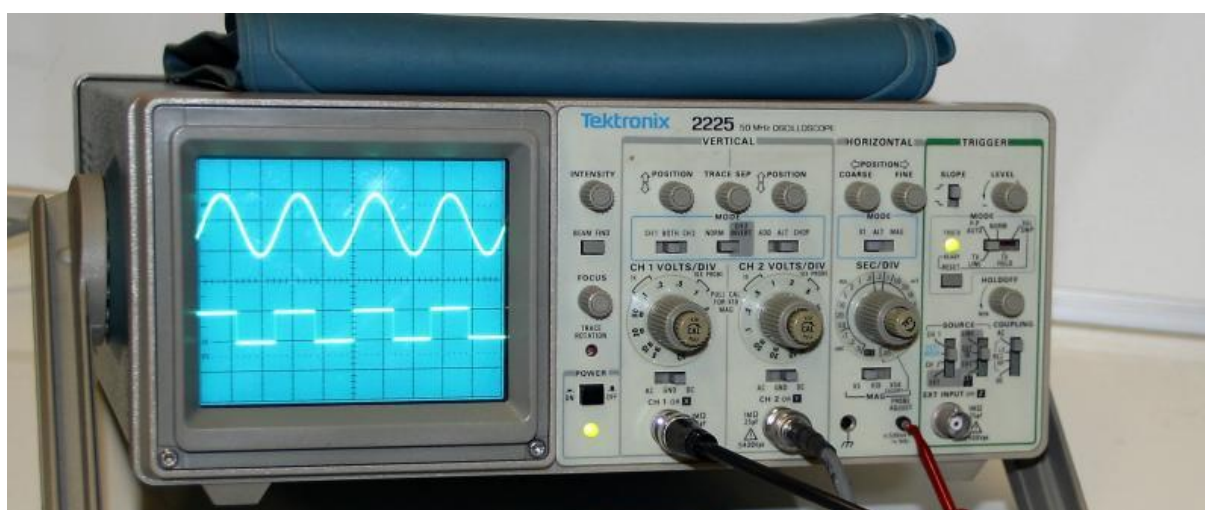


Figura 6 – Osciloscópio Analógico Tektronix 2225 (Disponível em < <http://www.woutersenwouters.be/sold-products/3259/tektronix-2225-50mhz-oscilloscope>>).

2.2.2 Princípio de Funcionamento do Osciloscópio Digital

O osciloscópio digital possui algumas características que o diferenciam do osciloscópio analógico, e essas características podem ser identificadas na Figura 7, onde se tem um diagrama de blocos simplificado, representando os principais sistemas que fazem parte construtiva do osciloscópio digital:

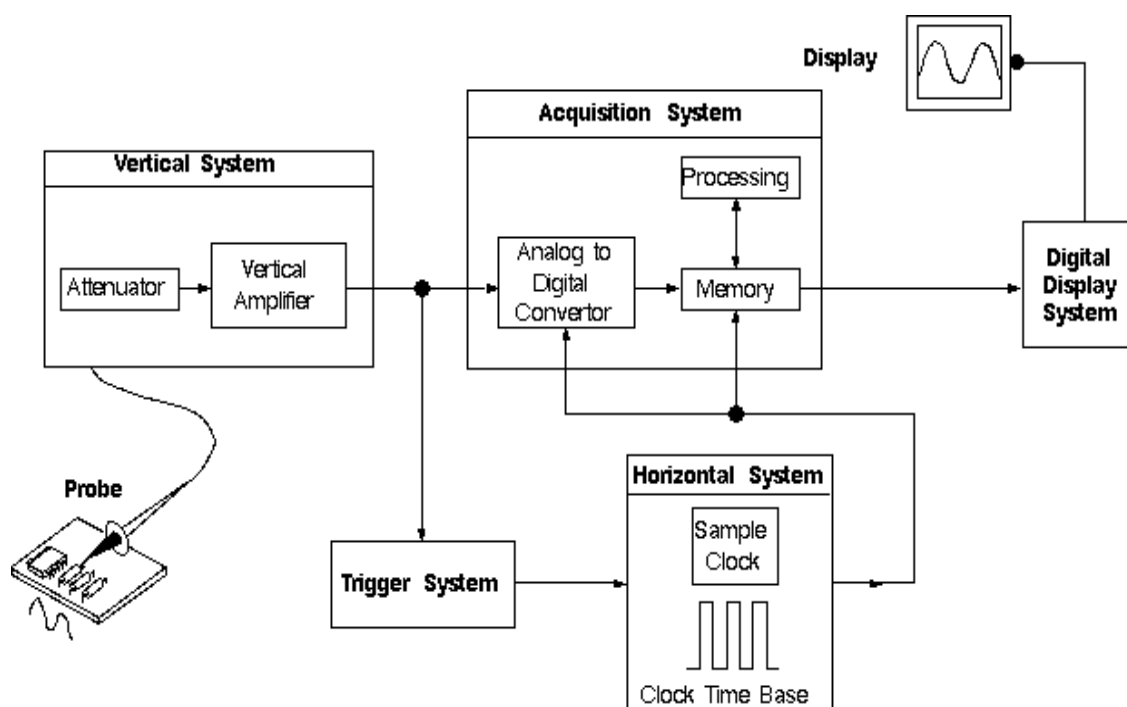


Figura 7 - Diagrama de blocos de um osciloscópio digital (Disponível em < <http://forums.anandtech.com/showthread.php?t=2213348>>).

Se observarmos o diagrama de blocos do osciloscópio digital veremos que, assim como no analógico, haverá um sistema responsável por efetuar o deslocamento do sinal na vertical. O que muda é a forma como o osciloscópio digital faz isso, pois agora não haverá as placas defletoras, mas sim, um circuito responsável por efetuar o ajuste da amplitude do sinal.

O osciloscópio digital também possui um sistema responsável por fazer a aquisição do sinal, de forma que o sistema “recolherá” amostras do sinal que está sendo medido e converterá o valor analógico de cada amostra em um valor digital.

O sistema horizontal possui um dispositivo chamado “*sample clock*”. Ele será responsável por determinar a frequência com que o conversor A/D realizará a aquisição das amostras e a conversão das mesmas. Após ter concluído a conversão, as amostras são armazenadas na memória para posteriormente serem mostradas na tela.

Cada amostra possuirá um conjunto de bits que o representará, e cada amostra será um ponto, que no final quando todos esses pontos forem juntados, formará o sinal que está sendo medido. É como se fosse um quebra cabeça, onde as amostras representam as peças, e o sinal formado a partir de todas essas amostras, representa o quebra cabeça montado por completo, ou seja, representa o resultado final. O conjunto de amostras é chamado de registro, e o osciloscópio digital, assim como no analógico, terá de ter um sistema de sincronismo, no caso do osciloscópio digital o sistema responsável pelo sincronismo determinará o início e o fim do registro, de forma que será definido o número de amostras necessárias e, conseqüentemente se terá um registro, que será armazenado na memória para posteriormente ser apresentado na tela.

2.2.3 Métodos de Amostragem nos osciloscópios digitais

Os osciloscópios digitais como foi discutido na subseção 2.3, realizam a aquisição do sinal que está sendo medido, realiza a conversão, grava na memória, e posteriormente converte para um sinal analógico para que possa ser mostrado no visor, isso pode ser visto na Figura 8. Na Figura 8 pode-se identificar que existe um bloco a mais que não continha no diagrama anterior (Figura 7), que era a conversão digital para analógico:

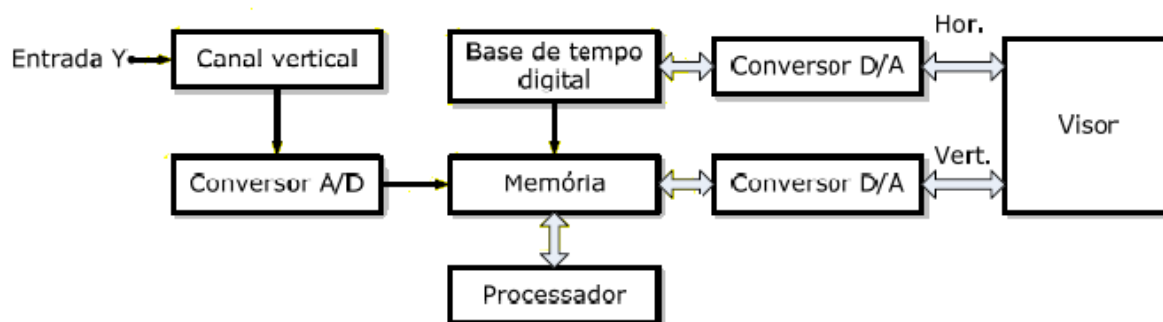


Figura 8 - Diagrama de blocos de um osciloscópio digital com conversores A/D e D/A (Adaptado e disponível em <<http://nebm.ist.utl.pt/repositorio/ficheiros/372>>).

Após o armazenamento do registro na memória, apresentado na tela LCD (estranho isso, pois se vai na tela LCD não é convertido, é sempre digital) para que o operador do equipamento possa enxergar o sinal. Com isso pode-se entrar no método de amostragem, que nada mais é que a forma com que o osciloscópio faz a aquisição das amostras.

Para sinais em frequências baixas o osciloscópio não encontra nenhum problema quanto a captura das amostras, de forma que geralmente um osciloscópio irá apresentar um sinal com uma boa qualidade, caso o sinal medido esteja a uma frequência considerada baixa. Porém existe um problema quando o sinal medido está a uma frequência alta, pois dependendo da capacidade do osciloscópio em questão, pode ser que o mesmo não consiga capturar a quantidade de amostras necessárias,

de forma que a precisão e resolução desse sinal ficarão comprometidas, devido a isso se têm os seguintes métodos de amostragem:

2.2.4 Amostragem em tempo-real (*real-time sampling*)

Nesse tipo de amostragem, o osciloscópio coleta uma determinada quantidade de amostras em um único ciclo de aquisição e, posteriormente, através da interpolação o sinal será construído. A interpolação é usada para que se possa estimar a forma de onda baseado em alguns pontos. Quanto maior o grau do polinômio mais preciso será o sinal, em contrapartida o poder computacional do osciloscópio deverá ser maior devido a complexidade dos cálculos (ALVES, 1998).

2.3 Amostragem em tempo-equivalente (*equivalent-time sampling*)

Nesse tipo de amostragem o sinal é formado aos poucos, ou seja, a cada ciclo de aquisição uma parcela do sinal é armazenada, e depois de um determinado número de ciclos de aquisição o sinal poderá ser construído e apresentado na tela para o usuário. Com toda a conceituação a respeito dos osciloscópios digitais discutido nos itens anteriores, pode-se observar na Figura 9 um osciloscópio digital de 4 canais e 100 MHz da Tektronix.

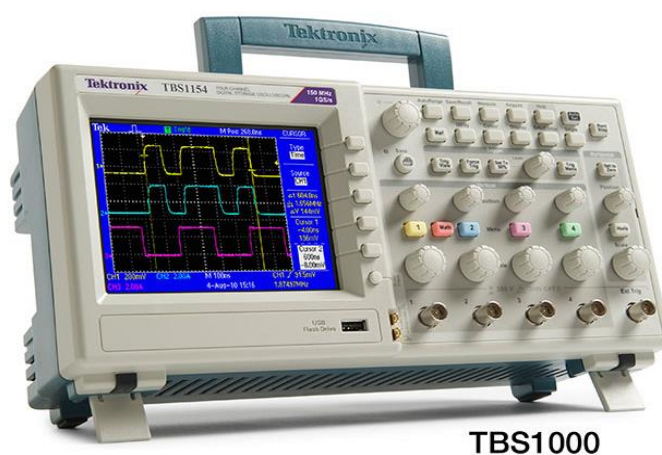


Figura 9 - Osciloscópio Digital Tektronix TBS1154 [Extraído de (TEKTRONIX® datasheet, 2017)].

2.3 Microcontrolador PIC18F4550

O microcontrolador PIC 18F4550 da *Microchip* foi o escolhido devido a vantagem que ele traz quando se trata de comunicação, pois o mesmo possibilita realizar uma comunicação com o computador via *USB*, além de ser muito usado nas disciplinas na Fatec Santo André.

Pode-se visualizar na Figura 10 as pinagens do microcontrolador PIC 18F4550:

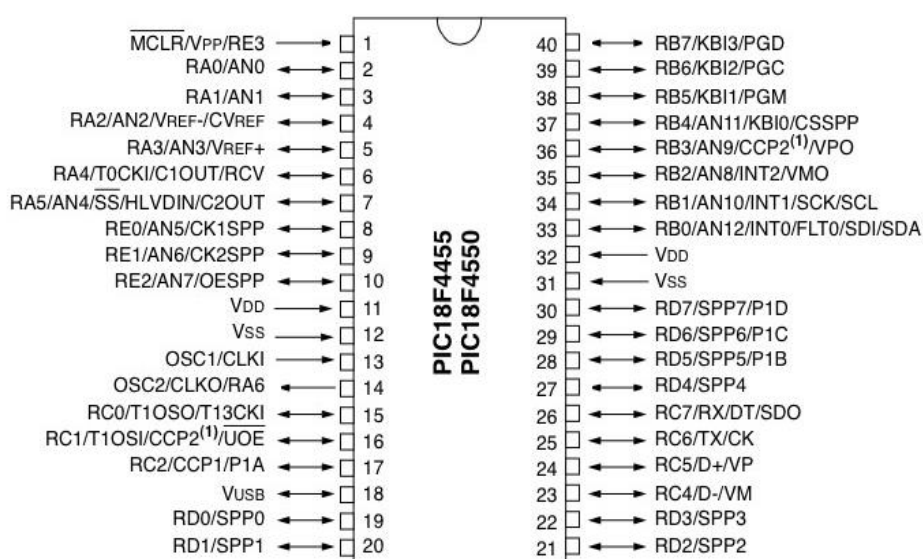


Figura 10 - Diagrama de Pinagem do PIC 18F4550 [Extraído de (MICROCHIP® datasheet, 2009)].

Características do PIC18F4550:

- Memória *FLASH* para armazenamento de programa: 32 Kbytes;
- Memória *SRAM* para armazenamento de dados: 2 kbytes;
- Memória *EEPROM* de dados: 256 bytes;
- Portas configuráveis como entradas ou saídas digitais: 35;
- Portas configuráveis como canais de entradas analógicas: 13;
- Módulo *CCP* (*Capture / Compare / PWM*)
 - *Capture* – Permite a medição do tempo de duração de um evento;

- Comparação – compara valores entre os registros do TMR1 e CCPR;
 - *PWM – Pulse Width Modulation* – gera sinais de frequência e variáveis.
- Porta paralela de 8 bits (SPP – *Streaming Parallel Port*);
 - Temporizadores de 8 e 16-bits: 4;
 - *Watchdog Timer*;
 - Frequência de operação de até 48 MHz;
 - Múltiplas fontes de interrupção (20);
 - Dois comparadores;
 - Periféricos avançados de comunicação. Porta de comunicação serial, Porta de comunicação USB 2.0;
 - Arquitetura *Harvard*, tecnologia *RISC* com um conjunto de 75 instruções;
 - Pilha de 31 níveis;

Pode ser encontradas mais informações no *datasheet* do PIC18F4550.

2.4 COMPILADOR CCS

Aqui nesta seção se encontram alguns conceitos referentes aos compiladores, em especial o compilador *PCWH* da *CCS*. Antes mesmo de começar a mostrar as características deste compilador, entenda o que é um compilador.

Um compilador é um programa de sistema que traduz um programa descrito em uma linguagem de alto nível para um programa equivalente em código de máquina para um processador. Em geral, um compilador não produz diretamente o código de máquina, mas sim um programa em linguagem simbólica (*assembly*) semanticamente equivalente ao programa em linguagem de alto nível. O programa em linguagem simbólica é então traduzido para o programa em linguagem de máquina através de montadores. (RICARTE,2003).

Após a breve explicação dos conceitos sobre compiladores, agora se pode trazer as características do compilador CCS C, que será utilizado para o desenvolvimento do código fonte responsável pela conversão A/D e comunicação com o computador através da porta *USB*.

O compilador *PCWH* da *CCS* é constituído de uma IDE gráfica que pode ser executado em qualquer plataforma *Windows*. Este ambiente permitirá uma programação dos PICs em linguagem C. Existe a possibilidade de se fazer uma integração com o ambiente *MPLAB* da própria microchip. Como vantagem do uso desse compilador pode-se citar a grande eficiência do código gerado, compatibilidade com o padrão ANSI e ISO salvo algumas exceções e a sua grande diversidade de funções e bibliotecas desenvolvidas em linguagem C (ANTONIO,2006,p.16).

Segundo Pereira (2003,p.33), as principais características do compilador da *CCS* são:

- Compatibilidade com a padronização ANSI E ISO (algumas características do compilador não fazem parte da normatização ANSI devido ao fato de serem específicas para a arquitetura PIC);
- Grande eficiência no código gerado;
- Grande diversidade de funções e bibliotecas da linguagem C (padrão ANSI), tais como: entrada/saída serial, manipulação de *strings* e caracteres, funções matemáticas C, etc.
- Grande portabilidade de código entre os diversos microcontroladores PIC e inclusive com o código escrito para outros microcontroladores ou sistemas. Isto significa que é muito fácil adaptar um programa escrito em C para outro dispositivo ou sistema (seja ele um outro PIC, outro tipo de microcontrolador, ou mesmo um programa para PC).

2.5 PORTA USB

A comunicação *USB* surgiu após muitas empresas notarem a necessidade de uma porta serial de comunicação padronizada que fosse mais compacta e veloz, de forma a resolver a dificuldade do usuário em ter que lidar com diversas conexões de periféricos do computador (ALECRIM, 2017).

Deu-se início a porta *USB (Universal Serial Bus)* padronização que com o passar do tempo ganhou novas versões com melhorias e aumento na velocidade de transmissão.

- **USB 0.7:** novembro de 1994;
- **USB 0.8:** dezembro de 1994;
- **USB 0.9:** abril de 1995;
- **USB 0.99:** agosto de 1995;
- **USB 1.0:** janeiro de 1996;
- **USB 1.1:** setembro de 1998;
- **USB 2.0:** abril de 2000;
- **USB 3.0:** novembro de 2008;
- **USB 3.1:** agosto de 2013.

O *USB* trouxe muitas vantagens. Com a padronização de conexão *USB* se tornou possível trabalhar com diversos dispositivos diferentes sem a necessidade de um conector específico para cada um deles, é compatível com diversas plataformas e sistemas operacionais, não é necessário fazer configurações devido o sistema fazê-lo de forma automática, fornece alimentação, permitindo que dispositivos sejam carregados ou entrem em funcionamento sem a necessidade de uma fonte externa além de fornecer uma alimentação que permite carregar *smartphones* e *tablets* (ALECRIM, 2017).

Visto as vantagens da *USB*, veja o funcionamento da porta *USB* quando conectado ao computador, a partir da explicação da Figura 11, que conta com o diagrama de blocos das camadas de *software* e *hardware* que constituem o funcionamento da conexão do *USB* com um computador.

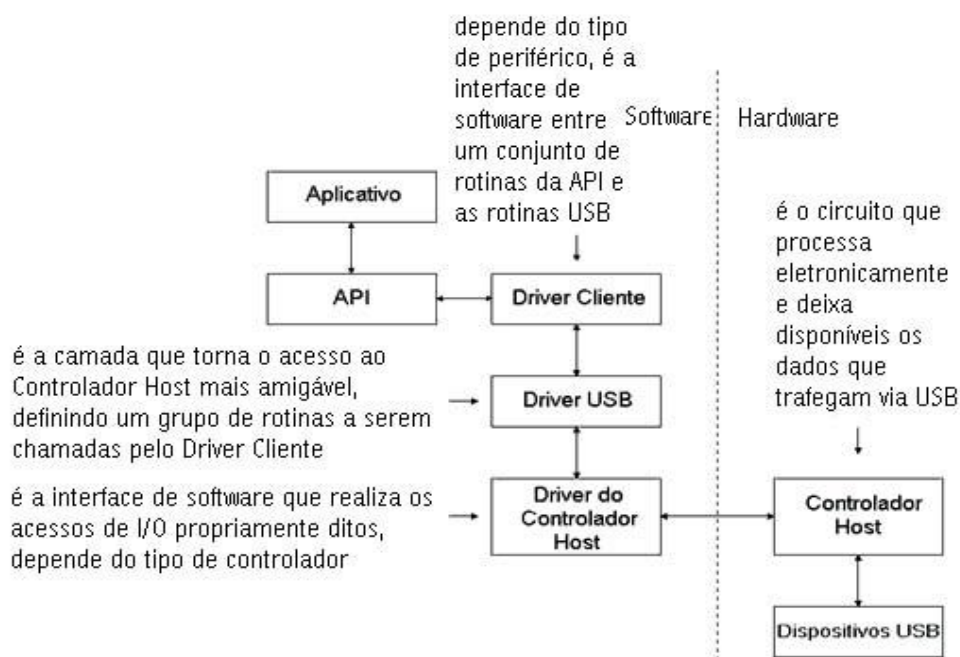


Figura 11 - Camadas de software e hardware que constituem o funcionamento da conexão do USB com um computador (Disponível em < https://www.gta.ufrj.br/grad/04_2/usb/).

O acesso à periféricos comuns funciona da mesma forma que o acesso à um periférico *USB*. Os dois se chamam funções da API. Então ocorre a comunicação da API com o driver do periférico conectado na porta *USB*. O driver que por sua vez fica armazenado no sistema operacional ou vem junto com o CD de instalação do dispositivo *USB* recebe as informações da API, traduz, e envia para *USB*, depois disso, vem a camada do Driver *USB*, torna o sistema operacional compatível com a *USB*.

A camada de software chamada Driver do Controlador Host., realiza os acessos de I/O mediante à recepção das chamadas vindas do Driver *USB*. Essas chamadas são organizadas por este driver, para serem acessadas pelo Controlador Host.

O Controlador Host por sua vez, é um circuito eletrônico que interpreta as rotinas enviadas pelo Driver do Controlador Host e ao mesmo tempo conecta todos os periféricos *USB*.

A seguir veja a Sequência de eventos que mostram como é conectado um periférico ao computador mesmo ele estando ligado. Pode-se ver na Figura 12 abaixo. Quando um periférico é conectado na porta *USB*, o Controlador Host detecta esta conexão e avisa ao Driver do Controlador Host que avisa ao Driver *USB*. O Driver

USB faz com que o Driver Cliente seja iniciado tornando possível a utilização do periférico recém-conectado (MEDINA, 2004).

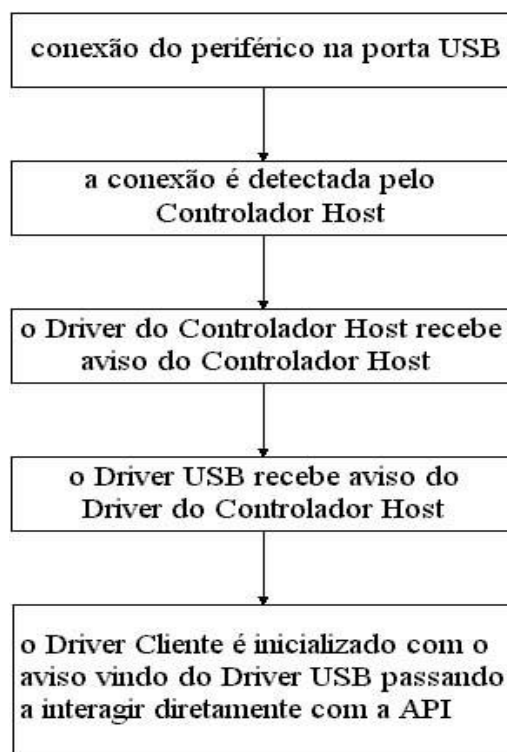


Figura 12 - Etapas de eventos que ocorrem durante a conexão de um periférico a porta *USB* (Disponível em < https://www.gta.ufrj.br/grad/04_2/usb/ >).

Para prover energia elétrica a determinados dispositivos os cabos *USB* contam no mínimo quatro fios internos: VBus (VCC), D+, D- e GND. O primeiro é o responsável pela alimentação elétrica. O segundo e o terceiro são utilizados na transmissão de dados. O quarto, atua no controle elétrico, servindo como "fio terra".

A comunicação entre dispositivos conectados via *USB* é feita por meio de um protocolo. Nele, o computador ou o equipamento que recebe as conexões, emite um sinal para encontrar os dispositivos conectados e estabelece um endereço para cada um deles. Depois que estabelecida a comunicação, o host recebe a informação sobre o tipo de conexão que o dispositivo conectado utiliza. Onde há quatro possibilidades (ALECRIM, 2017):

Bulk: utilizado por dispositivos que lidam com grandes volumes de dados, como impressoras e scanners. O Bulk possui um recurso para detectar erros e garantir a transferência sem perda de dados.

Control: utilizado para transmissão de parâmetros de controle e configuração do dispositivo;

Interrupt: utilizado para dispositivos que transferem poucos dados, como mouses e teclados.;

Isochronous: Utilizados em transmissões contínuas, onde os dados são transferidos a todo o momento, este não possui recurso de detecção de erro para não gerar atraso na transmissão.

Abaixo se podem verificar através da Figura 13 os tipos de conectores *USB* que se encontra no mercado.



Figura 13 - Tipos de Conectores *USB* (Disponível em < <http://medidasuteis.blogspot.com.br/2015/08/tipos-de-conectores-de-audio-e-video.html>>).

3. PROJETO

Esta seção irá trazer como foi o desenvolvimento do *hardware* adicional, código fonte em C, a *interface* no *LabVIEW* e os detalhes de cada um destes objetivos em subseções distintas.

3.1 *Hardware* Principal

O *hardware* principal empregado no projeto é o usado nas disciplinas da Fatec para auxiliar os alunos a desenvolverem habilidades em programação de microcontroladores. O diferencial desse *hardware* é a comodidade que traz ao usuário no processo de carregamento do código fonte para o microcontrolador, isso pelo fato de o microcontrolador utilizado ser o PIC 18F4550 que possibilita a comunicação via porta *USB 2.0*. O problema que se tinha, era a necessidade de possuir um gravador para carregar o firmware no microcontrolador, de forma que o usuário teria que investir em um gravador além de ter que investir no *hardware* a qual o microcontrolador foi empregado. Outro problema para usuário é o tempo de gravação, pois é necessário identificar os pinos corretos do microcontrolador para fazer a gravação.

Por isso a escolha do PIC 18F4550, entre os PIC 16F877A e o PIC 18f4552 que também foram utilizados durante o curso, pois existe a possibilidade de implementar o *Bootloader*, permitindo a comunicação via porta *USB 2.0* de forma que o usuário ganhará muito tempo para carregar um *firmware* no PIC, além de trazer segurança, pois não tem como ligar errado o cabo *USB* evitando a queima do PIC. Na Figura 14 abaixo se pode verificar melhor o processo para carregar um programa no PIC através do *hardware* principal com o *Bootloader*. Primeiramente o usuário deverá compilar o seu programa gerando o arquivo *.hex*. Depois de realizada a compilação, a placa deverá ser resetada, aguardando a conexão, uma vez estabelecida a conexão o usuário deverá carregar o arquivo e fazer o *upload*.



Figura 14 – Sequência para carregamento do *firmware* no Pic (Fonte: Os autores).

3.2 Desenvolvimento do *Hardware* Adicional

O *hardware* adicional condicionará o sinal para que este possa ser lido pelo microcontrolador. Inicialmente o sinal passará por um atenuador na entrada, ao todo são seis valores de atenuações, são elas:

$$1/2, 1/8, 1/16, 1/50, 1/75, 1/100.$$

O sinal passará por um seguidor de tensão e posteriormente por um “deslocador de zero”, que não passa de um circuito subtrator com resistências iguais que fará o papel de subtrair um sinal do outro a fim de elevar a referência de zero para 2V. Isso foi feito pelo motivo da entrada analógica do microcontrolador realizar a leitura de sinais na faixa de 0 a 5V. Uma vez passado pelo “deslocador de zero”, o sinal poderá obter um ganho, que no caso estão em quatro faixas, ganho de 1, 2, 4 e 10.

Os ganhos serão definidos pelo operador do osciloscópio através do *LabVIEW* por meio de uma comunicação serial com o microcontrolador através da porta *USB*, o microcontrolador acionará um pino do CI HCF4066 que fechará um contato com um resistor específico, fazendo com que o amplificador de ganho não inversor seja configurado para o ganho desejado.

Foram realizados diversos testes em cada bloco do circuito a fim de comprovar o perfeito funcionamento do mesmo. Foi testado primeiramente o bloco composto pelo seguidor de tensão e subtrator. O teste foi realizado com sinal contínuo e alternado senoidal.

A fórmula que indica o resultado na saída do subtrator é:

$$U_{out} = \frac{R2}{R1} \times (U2 - U1), \text{ onde:}$$

U_{out} = Tensão de saída em Volts.

$U1$ = Tensão de Entrada (Inversora) em Volts.

$U2$ = tensão de entrada (não Inversora) em Volts.

$R1$ e $R2$ são as resistências em ohms.

Antes de entrar nos testes propriamente ditos, vale ressaltar que trabalhando com componentes reais nem toda a teoria por trás dos amplificadores, por exemplo, pode funcionar exatamente como esperado. Para ser explicada a teoria de um componente qualquer sempre é mostrado, primeiramente, o modelo ideal ou seja, nesse modelo não existe nenhum tipo de perda, tudo funciona exatamente como se espera, mas na vida real não é bem assim. Então vem o modelo real do componente que mostra as pequenas perdas que o componente pode ter e como fazer para minimizá-las. O exemplo a seguir é simples, mas deixará claro ao leitor o comportamento de um amplificador operacional. Abaixo um amplificador configurado como seguidor de Tensão.

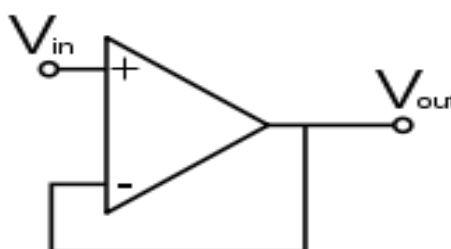


Figura 15 – Configuração seguidor de tensão (Disponível em < <http://athoselectronics.com/amplificador-operacional/>>).

Configuração mais simples que se pode realizar com um amplificador operacional, o Sinal de Saída V_{out} é igual ao Sinal de entrada V_{in} da porta não inversora, devido o amplificador possuir alta impedância de entrada não ocorre a circulação de corrente na entrada do Amplificador, possibilitando na saída V_{out} um sinal sem perdas. Veja a pinagem básica de um amplificador operacional.

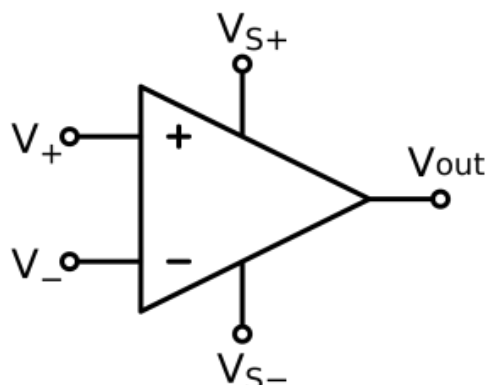


Figura 16 - Pinagem de um amplificador Operacional (Disponível em < <http://athoselectronics.com/amplificador-operacional/>>).

Como se pode ver, o amplificador é composto de duas entradas (V_+ e V_-), uma saída V_{out} e dois pinos para a alimentação V_{S+} e V_{S-} . O amplificador trabalhará na região de tensão de alimentação, ou seja, se a alimentação é de +5V e -5V não terá como ele trabalhar em uma região maior do que essa. Para trabalhar com sinais que possuam uma variação maior de tensão, bastará aumentar a tensão aplicada aos pinos de alimentação do amplificador, isso se no *datasheet* do componente estiver informando que essa tensão pode ser maior.

Existe um gráfico do amplificador operacional que indica a sua faixa dinâmica, ou seja, a região a qual ele está limitado a trabalhar, o gráfico mostra a região linear, e as regiões de saturação. A região de saturação é atingida quando uma tensão igual ou maior que a alimentação é aplicada na entrada do amplificador, isso no modelo ideal, pois no modelo real, devido o componente possuir perdas, a saturação acontece antes de atingir a tensão de alimentação, se um amplificador está sendo alimentado com +5V e -5V, quando for aplicado um sinal de +4V e -4V ou até menos, o amplificador atingirá a região de saturação, ou seja, parte do sinal é perdido. Abaixo existe um gráfico de um amplificador operacional.

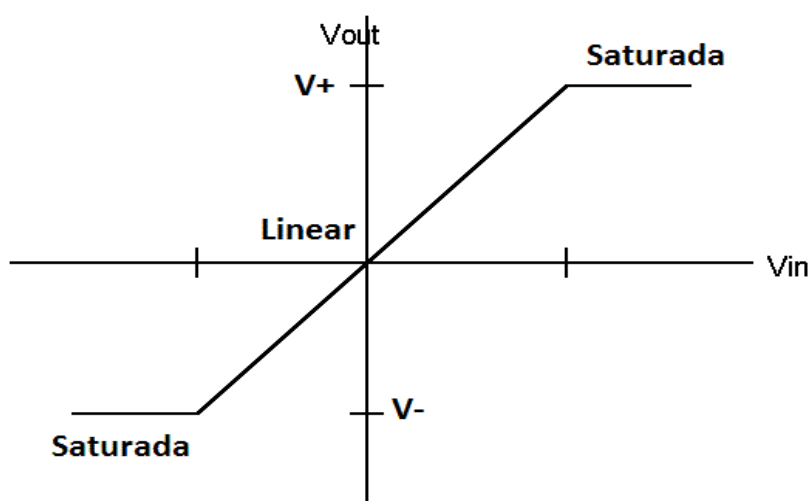


Figura 17 - Curva característica de transferência do Amplificador Operacional (Disponível em <<http://athoselectronics.com/amplificador-operacional/>>).

Antes de montar o circuito do TCC foi montado um circuito seguidor de tensão no *protoboard*, como pode ser visto na Figura 18 abaixo. Isso foi feito para poder mostrar ao leitor que o componente real tem perdas, e a limitação do componente quanto a essas perdas, e o que se pode fazer para minimizá-las.

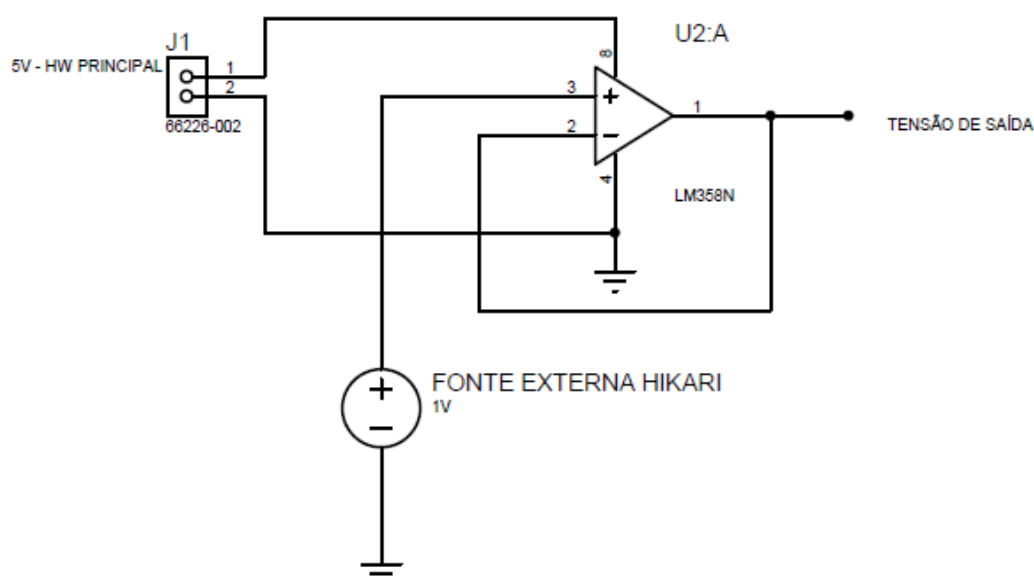


Figura 18 - Circuito seguidor de tensão (Fonte: Os autores)

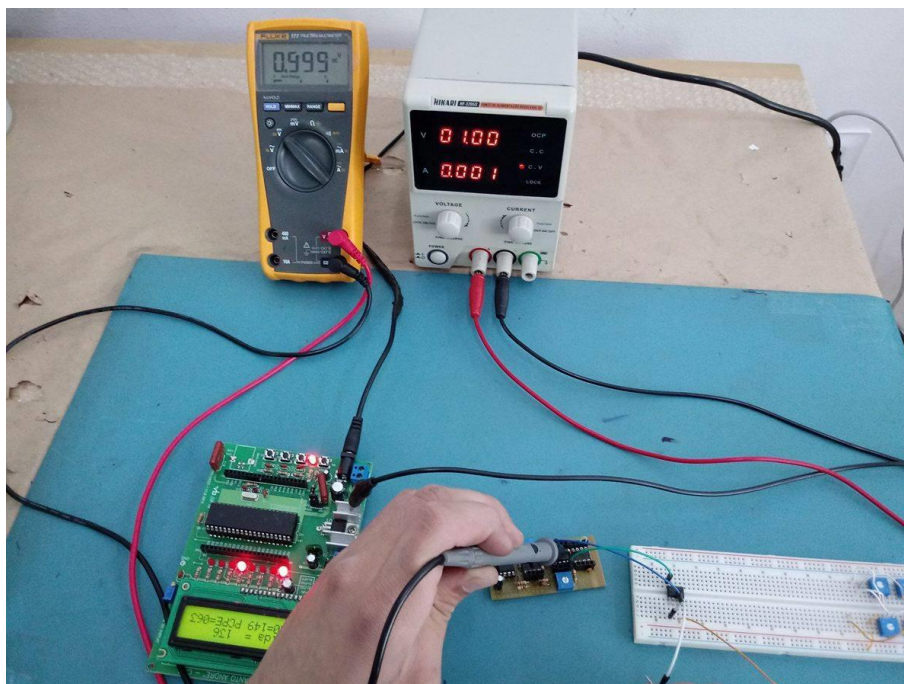


Figura 19 – Teste do circuito seguidor de tensão (Fonte: Os autores).

A Figura 20 indica a pinagem do CI LM358.

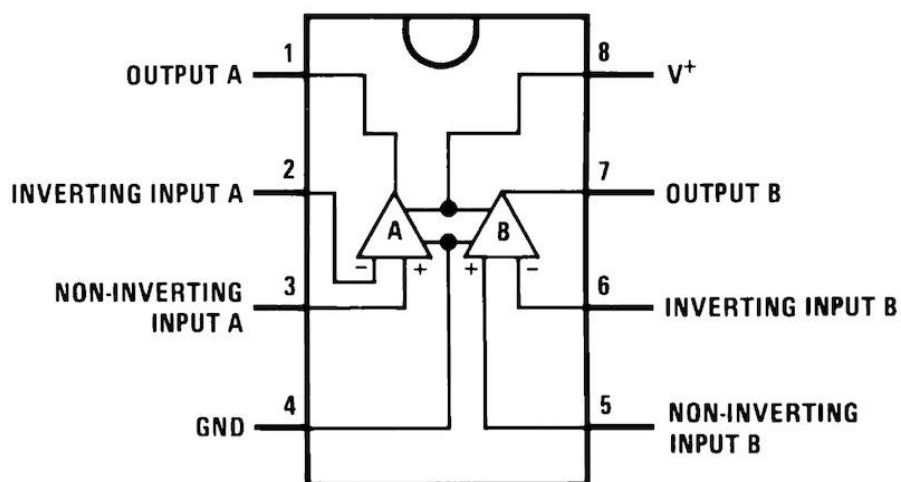


Figura 20 – Pinagem do CI LM358 [Extraído de (Texas Instruments, 2017)].

Como se pode ver na figura 18, o CI LM358 está sendo alimentado pelo *hardware* principal, que está fornecendo 5 VDC. A fonte como se pode ver está ajustada para 1 VDC e está conectada ao pino 3 do amplificador, e o pino 2 está ligado

com o pino 1, configurando o amplificador como seguidor de tensão, o multímetro está indicando o valor de tensão na saída, onde obteve-se um ganho unitário, ou seja, a tensão de saída é praticamente igual a tensão de entrada.

Abaixo, a Tabela 1 mostra valores teóricos de tensão, onde é feita uma comparação entre o sinal de entrada e o sinal que pode ser esperado na saída.

Entrada (Pin 3)	Saída (Pin1)
1 VDC	1 VDC
2 VDC	2 VDC
3 VDC	3 VDC
4 VDC	4 VDC
4.30 VDC	4.30 VDC
4.50 VDC	4.50 VDC
4.60 VDC	4.60 VDC
4.70 VDC	4.70 VDC
5 VDC	5 VDC
6 VDC	5 VDC
7 VDC	5 VDC

Tabela 1 – Sinal de saída teórico do seguidor de tensão (Fonte: Os autores).

Abaixo se mostra uma tabela real com os sinais contínuos de entrada e saída, a fim de identificar a faixa dinâmica do componente, provando ao leitor a limitação do mesmo.

Entrada (Pin 3)	Saída (Pin 1)
1 VDC	0.999 VDC
2 VDC	1.999 VDC
3 VDC	3.003 VDC
4 VDC	3.997 VDC
4.30 VDC	4.306 VDC
4.50 VDC	4.502 VDC
4.60 VDC	4.596 VDC
4.70 VDC	4.664 VDC
5 VDC	4.859 VDC
6 VDC	4.895 VDC
7 VDC	4.887 VDC

Tabela 2 – Sinal de saída Real do seguidor de tensão (Fonte: Os autores).

Como se pode ver, o amplificador começou a ter uma perda quando a tensão de entrada foi de 4.70 VDC, o que prova que o amplificador operacional como qualquer outro componente real, possui perdas. Outro ponto interessante a se observar é a tensão que foi obtida na saída quando a tensão de entrada foi de 5 VDC, o que mostra que não se pode atingir a tensão máxima estabelecida pela alimentação do componente. Caso fosse necessário obter um sinal de 5 VDC na saída, seria necessário um aumento na tensão de alimentação do mesmo.

Abaixo, as figuras comprovam os resultados do teste com sinal contínuo.

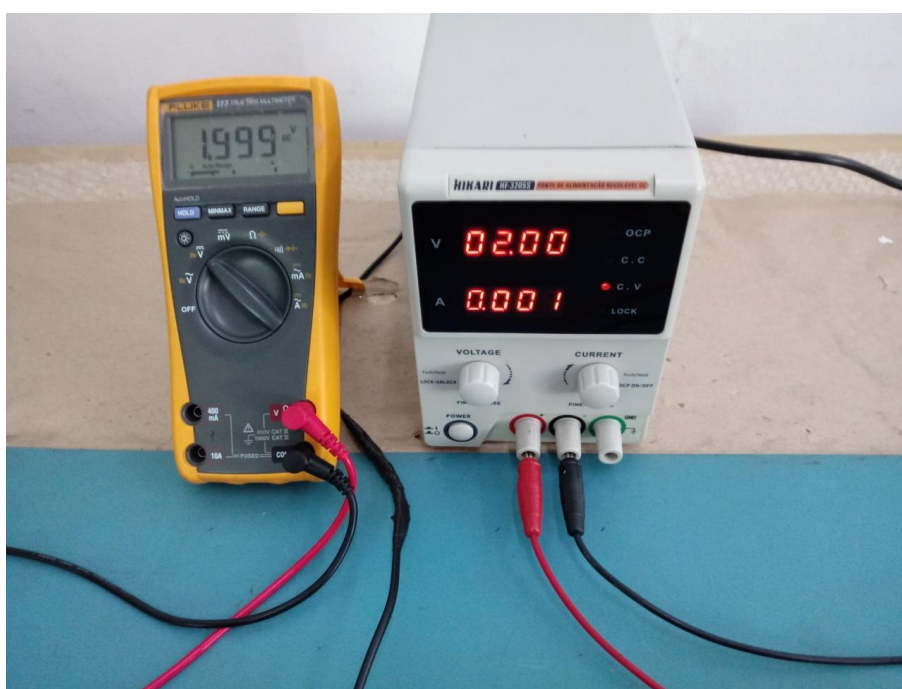


Figura 21 – Saída do seguidor de tensão ao aplicar 2 VDC em sua entrada (Fonte: Os autores).



Figura 22 – Saída do seguidor de tensão ao aplicar 5 VDC em sua entrada (Fonte: Os autores).

Visto o experimento realizado quanto à limitação do amplificador operacional, o próximo texto é sobre o teste do *hardware* adicional com os sinais contínuos e alternados. Podem-se verificar nas Figuras seguintes os resultados obtidos durante o teste do ganho do sinal de saída.

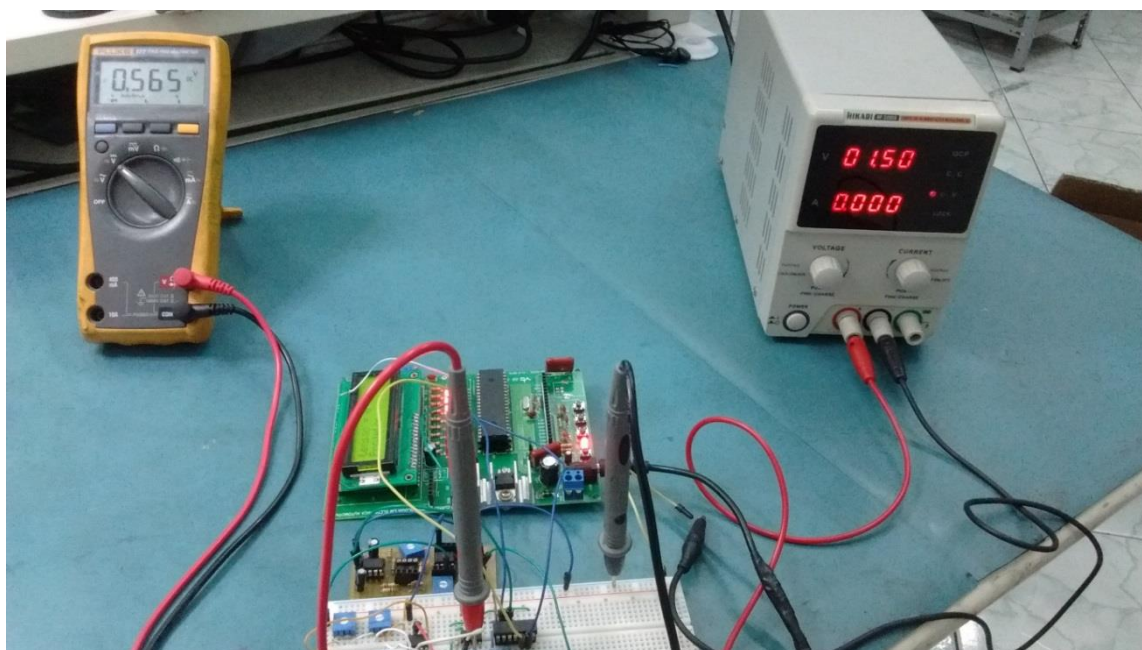


Figura 23 - Sinal de Saída com Ganho Unitário e sinal de entrada 0.5V (Fonte: Os autores).

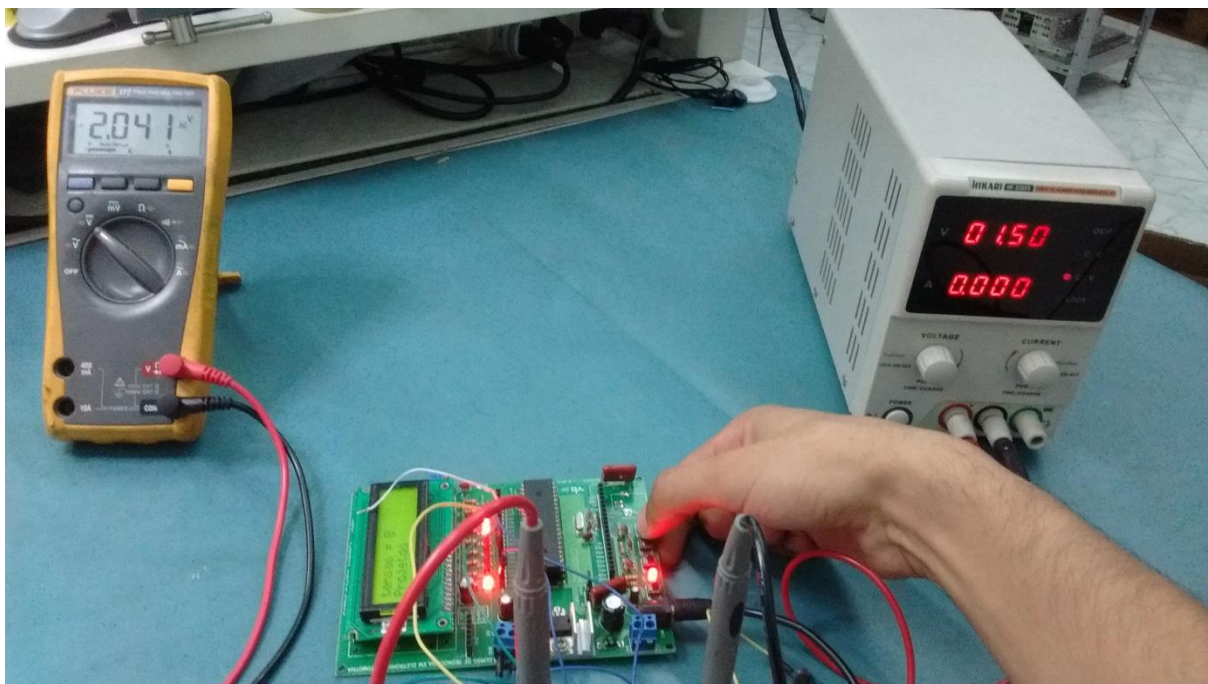


Figura 24 - Sinal de Saída com ganho quatro (Fonte: Os autores).

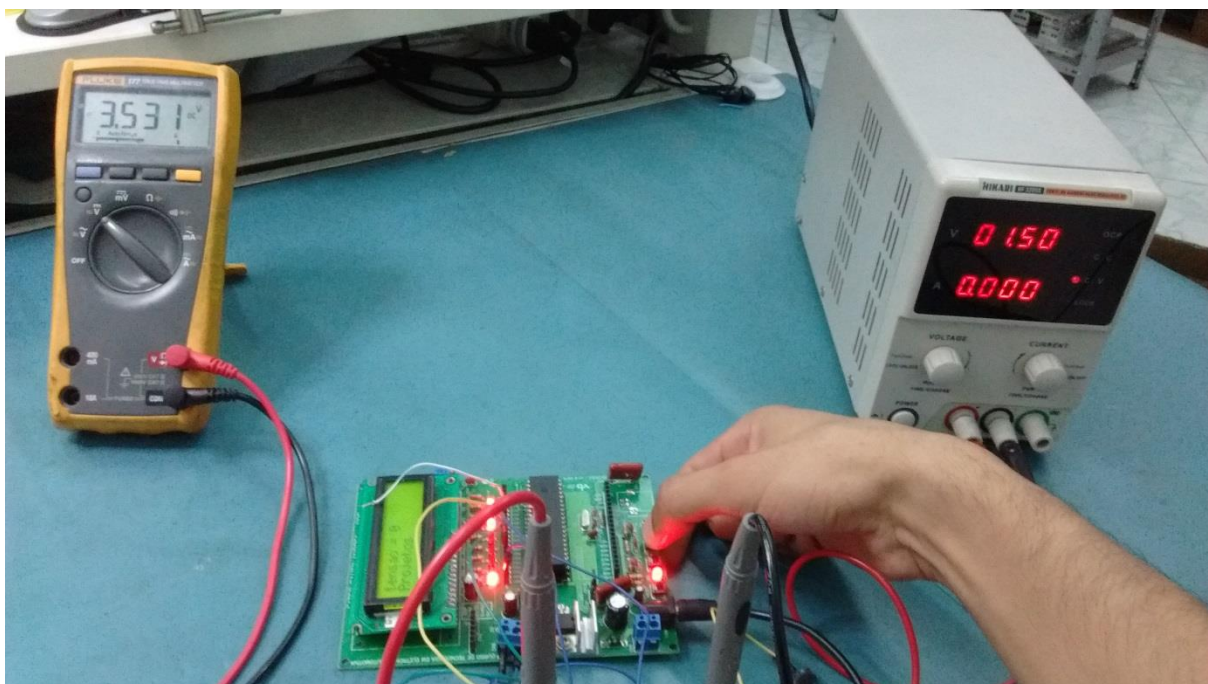


Figura 25 - Sinal de Saída com ganho dez (Fonte: Os autores).

Pode-se verificar que o circuito terá pequenas variações no ganho do sinal, mas está muito próximo do desejado. Se voltar ao início da seção onde foi apresentada a equação que indica a tensão na saída do circuito subtrator e aplicar a tensão da fonte, sabendo que as resistências são iguais, se obtém a seguinte equação:

$$U_{out} = (2V - U1), \text{ Onde:}$$

$U1$ é a tensão de entrada, que poderá ser alternada ou contínua.

No teste em questão o sinal de entrada foi um sinal contínuo, que vinha da fonte, como se pode ver na figura 23. Pode-se verificar através do multímetro que a tensão está próxima do esperado, o ponto de medição escolhido foi exatamente o ponto que irá para o microcontrolador, como se pode ver, o teste do circuito foi realizado manualmente, onde cada botão ao ser pressionado fecha um contato do CI HFC4066 que resulta em um ganho desejado. Quando não apertar botão nenhum se tem o ganho unitário, e foi exatamente o que se obteve na saída.

O *hardware* adicional está funcionando normalmente, o próximo passo será realizar a integração entre o *hardware* adicional e o *hardware* principal, desenvolvendo o código fonte em C e a interface no *LabVIEW*. Os próximos passos serão tratados nas próximas seções.

3.3 Desenvolvimento do Código Fonte em Linguagem C

Essa seção visa informar ao leitor apenas a premissa para a construção e modificação do código fonte. Para maiores detalhes consultar o código completo em anexos.

Para que o sinal seja lido e mostrado em boa qualidade e precisão precisa-se de um tempo de leitura pequeno e contínuo, para isso foi configurado o timer0 do PIC18F4550 para que haja um estouro a cada 1 ms.

Para a obtenção do sincronismo entre a leitura do Sinal desejado e o envio desses dados pela *USB*, um vetor foi preenchido com amostras de leitura, este vetor ao estar completamente preenchido, transfere suas amostras a outro vetor que

posteriormente transferirá suas amostras para o *LabVIEW* através do *USB*. Onde só o faz, se o primeiro vetor finalizar a transferência das amostras para ele e que o PIC tenha terminado de enviar os dados pela *USB*.

Isto permite que os dados sejam coletados e enviados de forma sincronizada evitando que haja subscrição de dados ao tentar ler e enviar executando uma das operações de forma mais rápida que a outra, mas, também faz com que se percam sinais, pois quando os novos dados não foram passados para o envio pela *USB* por ela ainda estar enviando o anterior, esses novos dados são perdidos e substituídos por dados ainda mais novos.

Os dados enviados pelo *LabVIEW* através do *USB* são em *ASCII*, portanto, precisa-se convertê-los para que se possa realizar as operações necessárias.

Para melhor entendimento consulte a Tabela 3 abaixo.

Dec	Hex	Oct	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr	Dec	Hex	Oct	HTML	Chr
0	0	000	NULL	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	Start of Header	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	Start of Text	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	End of Text	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	End of Transmission	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	Enquiry	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	Acknowledgment	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	Bell	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	Backspace	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	Horizontal Tab	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	Line feed	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	Vertical Tab	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	Form feed	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	Carriage return	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	Shift Out	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	Shift In	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	Data Link Escape	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	Device Control 1	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	Device Control 2	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	Device Control 3	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	Device Control 4	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	Negative Ack.	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	Synchronous idle	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	End of Trans. Block	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	Cancel	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	End of Medium	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	Substitute	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	Escape	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	File Separator	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	Group Separator	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	Record Separator	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	Unit Separator	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		Del

Tabela 3 – Tabela *ASCII* (Disponível em <<http://www.codcad.com/lesson/7>>).

Se por exemplo, o número 0x32 for enviado, primeiramente será enviado o mais significativo, número 3 (conforme a Tabela 3 corresponde a 0x33), valor real

que será enviado pelo *LabVIEW*. Aqui já se pode notar que para cada um dígito que ele deseja enviar, é necessário um Byte, portanto para enviar 0x32 utilizam-se dois bytes, sendo um byte para o número 3 e um byte para o número 2.

Para converter o valor 0x33 em 3 novamente, o subtraímos por 0x30, desta forma temos $0x33 - 0x30 = 0x03$. Em seguida é enviado o número 2, 0x32 em ASCII, para convertê-lo realizamos a mesma operação, resultando em 0x02.

Agora temos dois Bytes, 0x03 e 0x02, para que volte a ter 1 Byte que represente esses dois valores, precisa-se deslocar o 0x03 4 bits para esquerda ficando 0x30 e em seguida fazer uma operação lógica “OU” para que se junte os dois valores, ficando: $0x30 | 0x02 = 0x32$.

Como o programa no *LabVIEW* enviará números ou letras em hexadecimal, nem sempre a subtração será 0x30. Para letras se efetua a mesma lógica. Mas em vez de 0x30, se subtrai 0x37.

Se por exemplo, seja enviado 0x0B, primeiramente será enviado o 0 (*ASCII*) como anteriormente, e em seguida o B (*ASCII*), correspondente a 0x42 em hexadecimal e 66 em decimal. 0x37 é o mesmo que 55 em decimal, logo se tem $66 - 55 = 11$ que é o mesmo que B.

Para descobrir se precisa fazer a subtração de 0x30 ou 0x37 verifica-se se o valor que chegará é menor que 40, pois os números de 0 á 9 variam de 0x30 a 0x39 e as letras de A a F variam de 0x41 à 0x46, sendo assim, se o dado for verdadeiro será um número, se for falso será uma letra.

O trecho do código abaixo fica responsável por identificar o ganho desejável pelo usuário, que o seleciona através da interface no *LabVIEW* e configura o PIC18F4550 de forma que ele efetue a instrução desejada. Ao receber o valor do ganho, que fica armazenado na variável C, o valor é identificado e a partir dele configurado as saídas do PIC18f4550 de forma que ele envie a tensão de 5V apenas no pino do CI CD4066 que fecha o contado correspondente ao ganho selecionado pelo usuário e os demais pinos recebem 0V para que permaneçam abertos. O usuário terá apenas 4 opções de ganho, sendo elas: 1, 2, 4 e 10.

```
switch (c)
{

    case 1 : output_low(pin_b5);
            output_low(pin_b6);
            output_low(pin_b7);
            output_high(pin_b4);

    break;

    case 2 : output_low(pin_b4);
            output_low(pin_b6);
            output_low(pin_b7);
            output_high(pin_b5);

    break;

    case 4 : output_low(pin_b4);
            output_low(pin_b5);
            output_low(pin_b7);
            output_high(pin_b6);

    break;

    case 10 : output_low(pin_b4);
            output_low(pin_b5);
```

```
        output_low(pin_b6);  
        output_high(pin_b7);  
  
    break;  
  
    default: output_low(pin_b5);  
            output_low(pin_b6);  
            output_low(pin_b7);  
            output_high(pin_b4);  
  
}
```

3.4 Desenvolvimento da Interface no *LabVIEW*

No decorrer do trabalho, verificou-se a necessidade da aquisição de uma versão do *software LabVIEW*, por se tratar de um *software* de alto custo, optou-se pela versão estudante que é gratuita e liberada por seis meses.

O lado esquerdo e direito da Figura 26 representa o *front panel* e o *block diagram* do *LabVIEW* respectivamente.

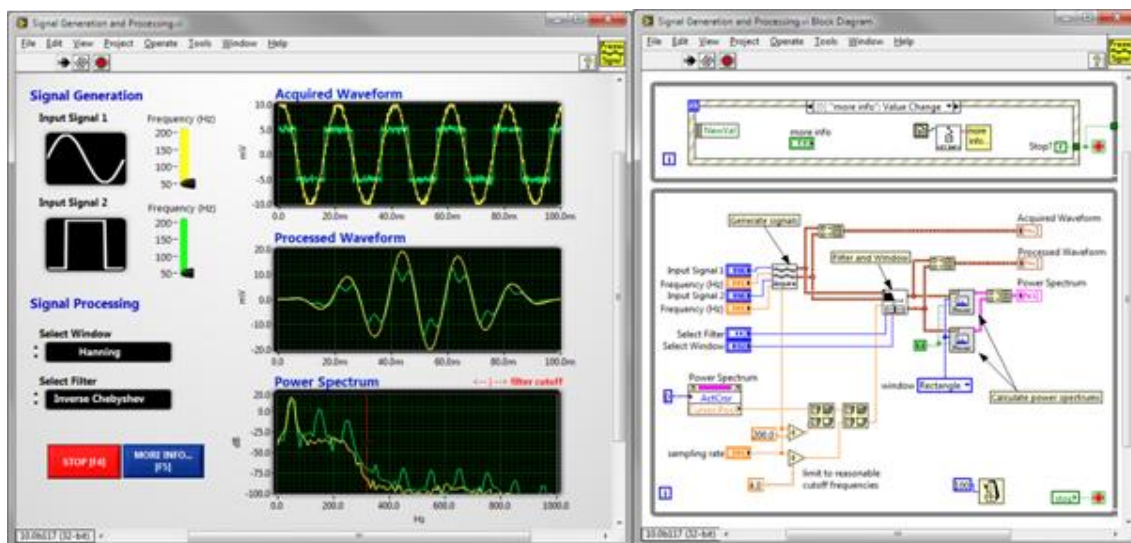


Figura 26 - *Front Panel* e *Block Diagram* de uma aplicação com *Labview* (Fonte: Disponível em: < <http://www.ni.com/newsletter/51141/pt/>>).

Observa-se que o *LabVIEW* traz uma tela para o programador (*Block Diagram*), onde desenvolve-se toda a aplicação necessária, e a tela do usuário (*Front Panel*), onde tem-se indicadores, controles, gráficos entre outros. Além disso o *LabVIEW* traz ao programador a possibilidade de customização da interface no *Front Panel*.

A interface usada no projeto partiu de uma interface que era usada na aula de ferramentas computacionais, onde se comunicava o *hardware* principal com o *LabVIEW*. Para o projeto foi realizada a modificação de uma interface já existente.

Um dos primeiros passos antes de realizar a comunicação com o *hardware* foi a instalação do *VISA* (*Virtual Instrument Software Architecture*).

VISA é um *software* da *National Struments* que configura, programa e soluciona problemas de sistemas de instrumentação que inclua a comunicação *USB* por exemplo, além de outros tipos de comunicação. Uma vez instalado o *VISA*, a comunicação entre o *LabVIEW* e o *hardware* principal pode ser estabelecida. Na Figura 27, pode-se verificar o diagrama que representa o “papel” do *VISA*.

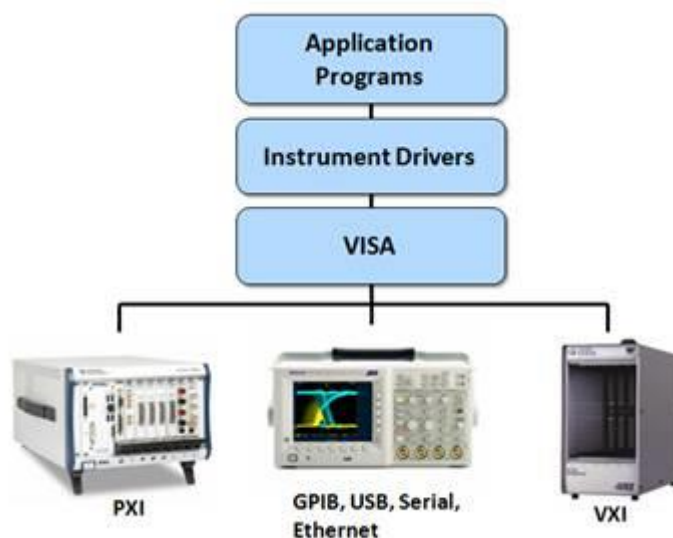


Figura 27 - Diagrama de Blocos da comunicação da Aplicação com *Hardware* [Extraído de (National Instruments, 2012)].

Após realizar a instalação dos *softwares* necessários e o desenvolvimento do código fonte em C, o próximo passo foi realizar o teste de comunicação do *hardware* com o computador e visualizar o sinal que o microcontrolador está lendo. O teste a seguir foi realizado com um sinal contínuo de 1.8 VDC que quando passa pelo *hardware* é deslocado, como visto na seção anterior, a tensão na saída do *hardware* adicional possuirá um ganho de 1, 2, 4 e 10.

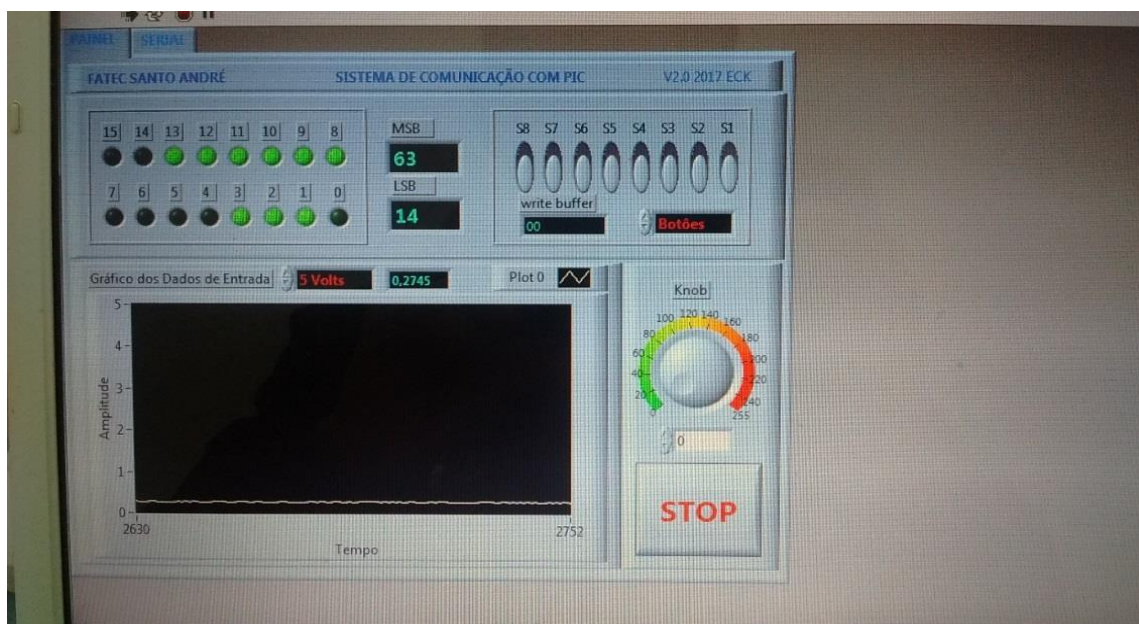


Figura 28 - Leitura do Sinal com Ganho Unitário Apresentado na Tela do *Datalogger* (Fonte: Os autores).

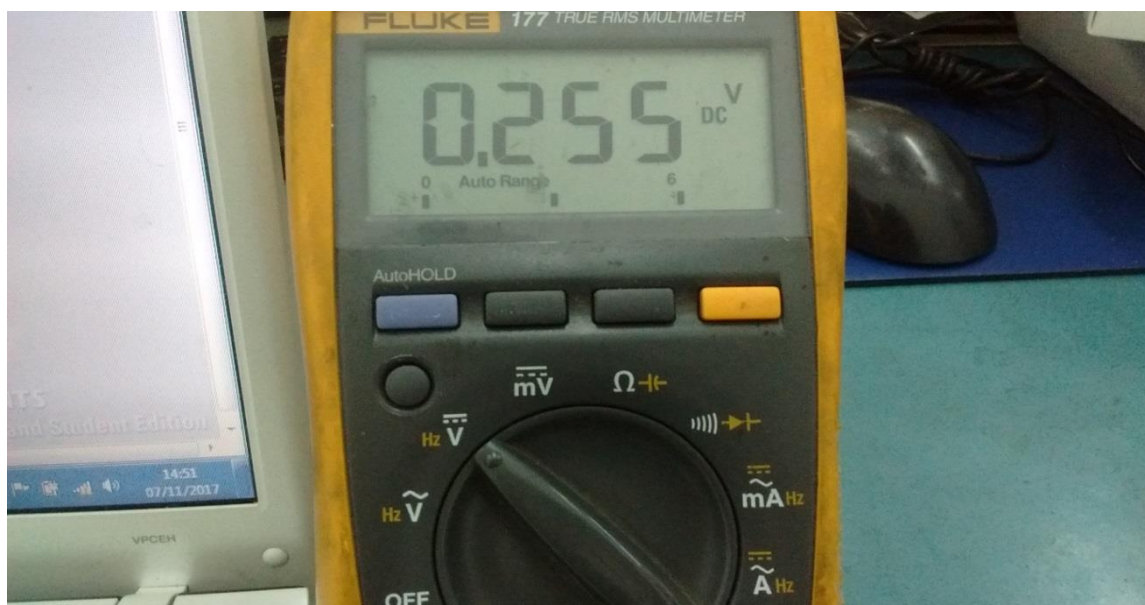


Figura 29 - Leitura do Sinal com Ganho Unitário Efetuada Pelo Multímetro (Fonte: Os autores).

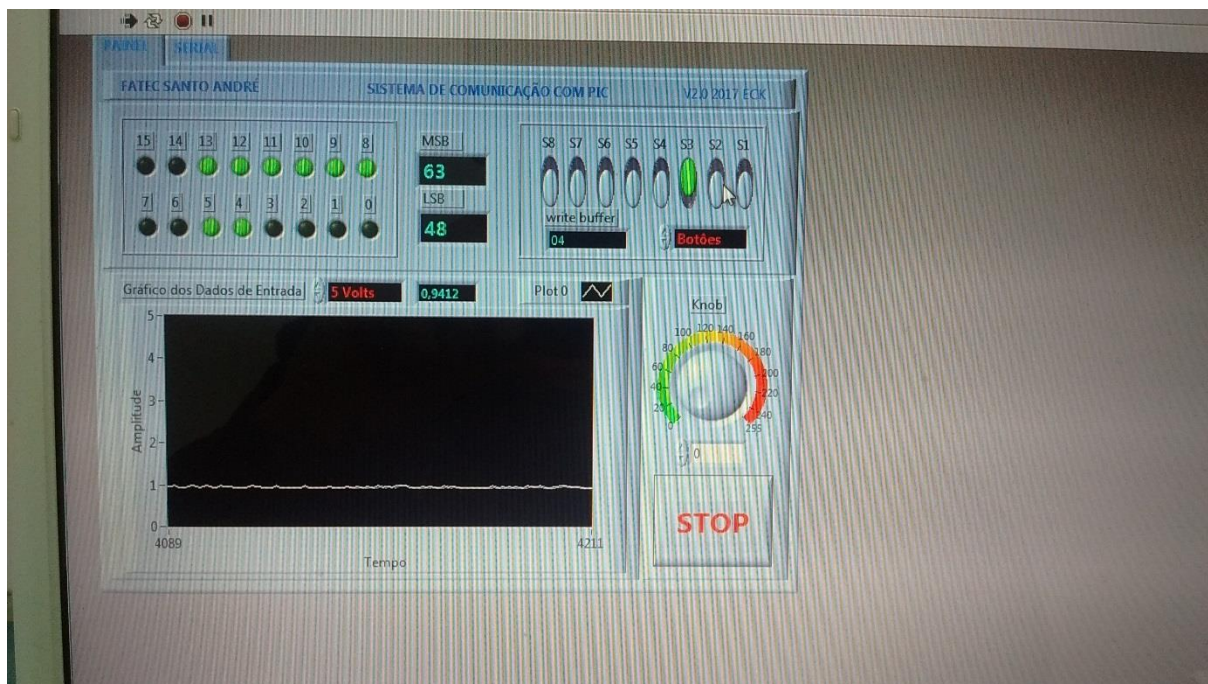


Figura 30 - Leitura do Sinal com Ganho Quatro Apresentado na Tela do *Datalogger* (Fonte: Os autores).



Figura 31- Leitura do Sinal com Ganho Quatro Efetuada Pelo Multímetro (Fonte: Os autores).

Como se pode ver, a leitura apresentada no gráfico do *Datalogger* está bem próxima do valor lido pelo multímetro. Outra coisa que também pode ser observada é

o ganho que o sinal obteve quando o botão B3 do *Datalogger* foi acionado, o que mostra que a comunicação entre o *Hardware* e o *LabVIEW* está funcionando normalmente. O teste a seguir foi realizado com sinais alternados, onde foi utilizado um Regulador de Tensão Alternada (*Variac*) como se pode verificar na Figura 32.

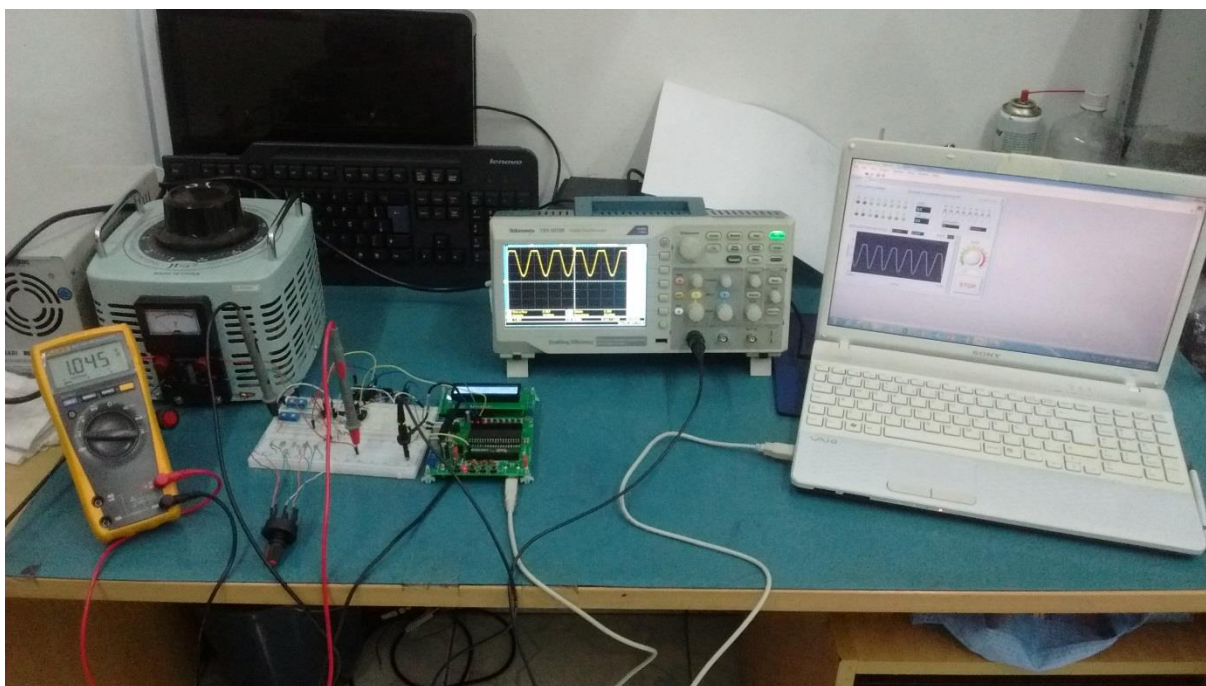


Figura 32 - Teste da Leitura do Sinal Alternado Efetuado pelo *Datalogger* (Fonte: Os autores).

O regulador de tensão alternada foi regulado para uma tensão de 1.045 VAC, como se pode verificar no multímetro. O *datalogger* apresentou o sinal idêntico ao sinal lido pelo osciloscópio.

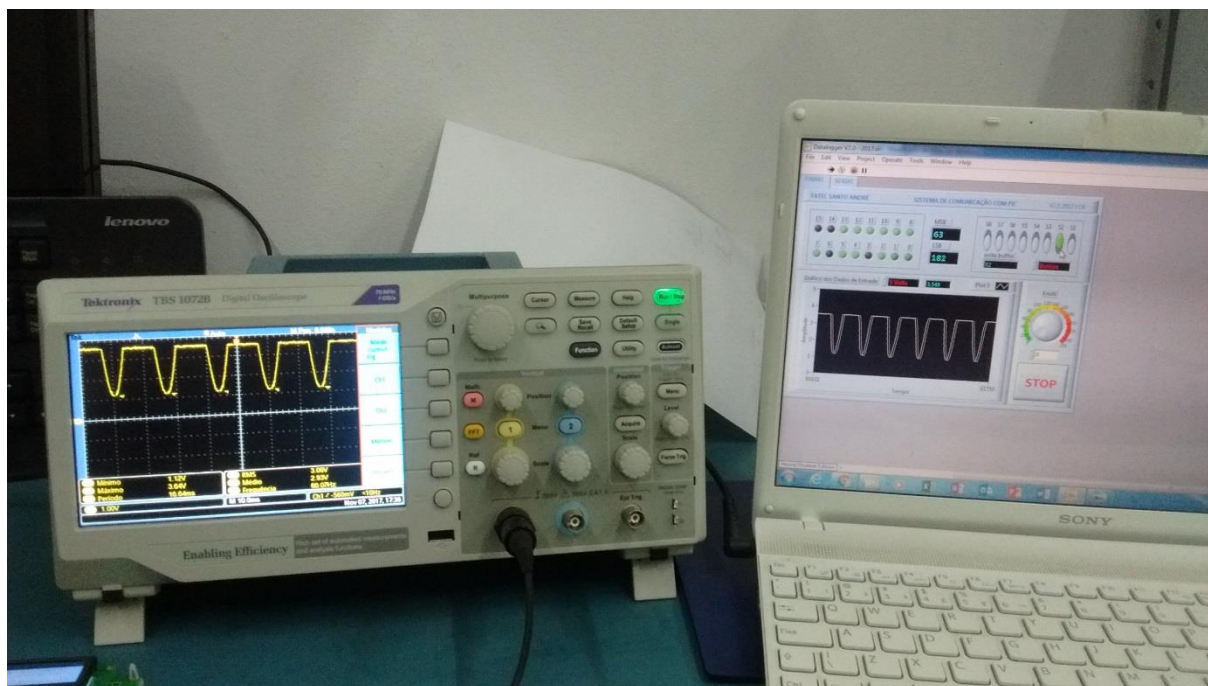


Figura 33 - Sinal Alternado com Ganho 2 (Fonte: Os autores).

Analisando o sinal do osciloscópio em relação ao sinal do *Datalogger* pode chegar a conclusão que a leitura está sendo efetuada corretamente. Vale ressaltar que o *Datalogger* não identifica as características do sinal lido como Tensão de Pico a Pico, Tensão RMS, Frequência entre outros. Esses dados serão inseridos durante a adaptação do *Datalogger* para o Osciloscópio Didático.

A seguir será discutido os trechos da programação em *LabVIEW*, para maior entendimento do funcionamento do osciloscópio didático.

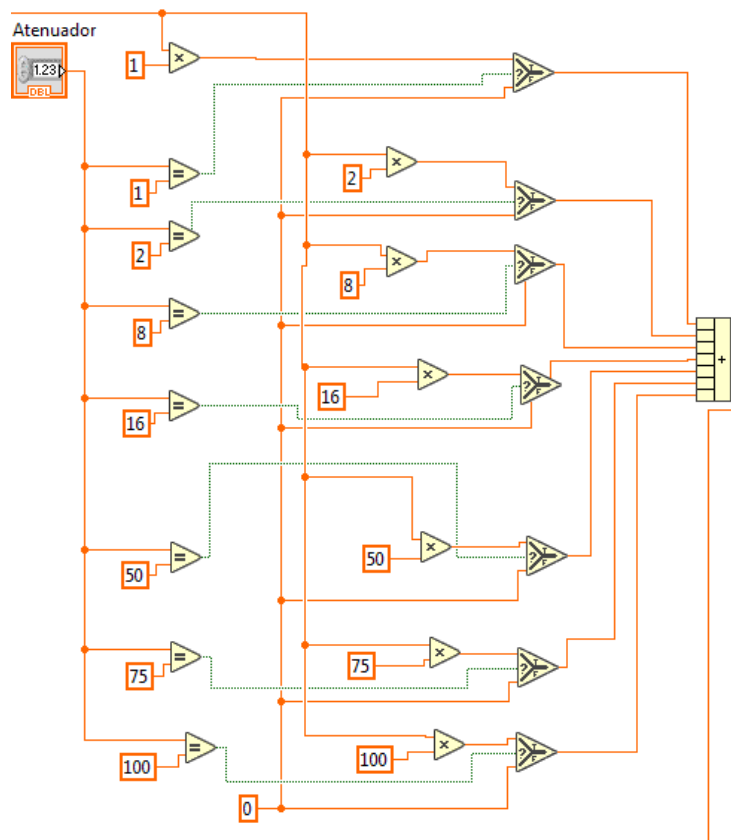


Figura 34 - Trecho da programação em *LabVIEW* que faz a atenuação (Fonte: Os autores).

A Figura 34 como pode ser visto, traz a construção do trecho do *software* responsável pela “compensação” da atenuação do sinal, ou seja, quando é selecionada a atenuação através da chave seletora no *hardware* adicional, o *LabVIEW* não realiza a compensação da atenuação de forma automática afim de indicar o sinal com os valores adequados, é necessário que o usuário indique de forma manual através do controle “Atenuador” em quantas vezes o sinal foi atenuado, para que o sinal indicado no gráfico seja apresentado com os valores corretos.

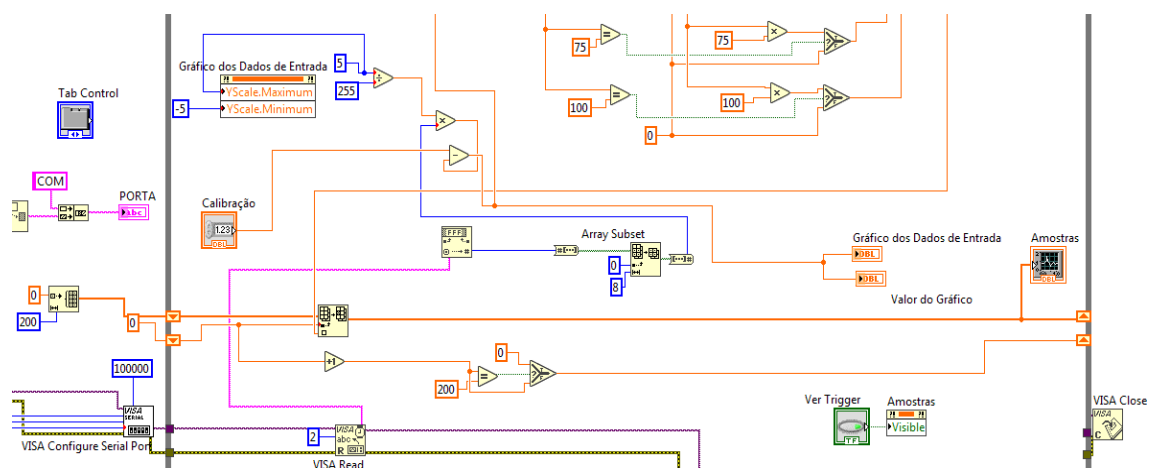


Figura 35 - Trecho da Programação em *LabVIEW* responsável por apresentar o sinal no gráfico (Fonte: Os Autores).

A Figura 35 acima se trata do trecho do *software* em *LabVIEW* responsável por armazenar e apresentar os sinais adequadamente no gráfico. Foram acrescentados dois gráficos, um apresentará o sinal sendo lido a cada iteração e o outro apresentará os valores que estão armazenados no vetor de duzentas posições que foi criado. A finalidade do gráfico que indica as duzentas posições do vetor, é de apresentar o sinal estabilizado na tela, fazendo a função *trigger*. O usuário pode optar em visualizar ou não o gráfico que indica o sinal estabilizado.

Observou-se a necessidade de implementar um *case structure* com dois cases, sendo um para realizar a leitura e apresentação do sinal no gráfico e o outro para aplicar um ganho ao sinal obtido, porém durante o teste verificou-se que quando parte do trecho do código da Figura 35 foi acrescentado dentro do case, o sinal no gráfico começou a correr mais lentamente, por esse motivo optou-se em manter o trecho responsável pela leitura fora do *struct case*, para deixar o *LabVIEW* executando o programa com uma velocidade maior.

Através desta alternativa observou-se que a leitura e apresentação estaria ocorrendo a todo o momento a partir do instante em que o *LabVIEW* fosse inicializado, quando o usuário desejasse que o sinal obtivesse um dos ganhos, seria necessário que o programa fosse pausado, pois uma vez que o programa executasse o case responsável pelo ganho ao passo que realizava a leitura, trazia problemas pelo motivo de estar tentando fazer a escrita ao mesmo tempo em que é desejado efetuar a leitura dos sinais, então em determinado momento o programa apresentava uma falha de

comunicação devido ao estouro do tempo em estabelecer a comunicação com o *hardware* principal.



Figura 36 - Interface Gráfica para operação dos sinais (Fonte: Os autores).

Observa-se através da Figura 36, o controle (*Knob*) no *Front Panel*, responsável por atribuir um ganho desejado ao sinal que está sendo lido. A figura 37 indica o trecho do código responsável pelo ganho do sinal, que está limitado a quatro ganhos e é executado quando o case está 1, configurado como Ganho.

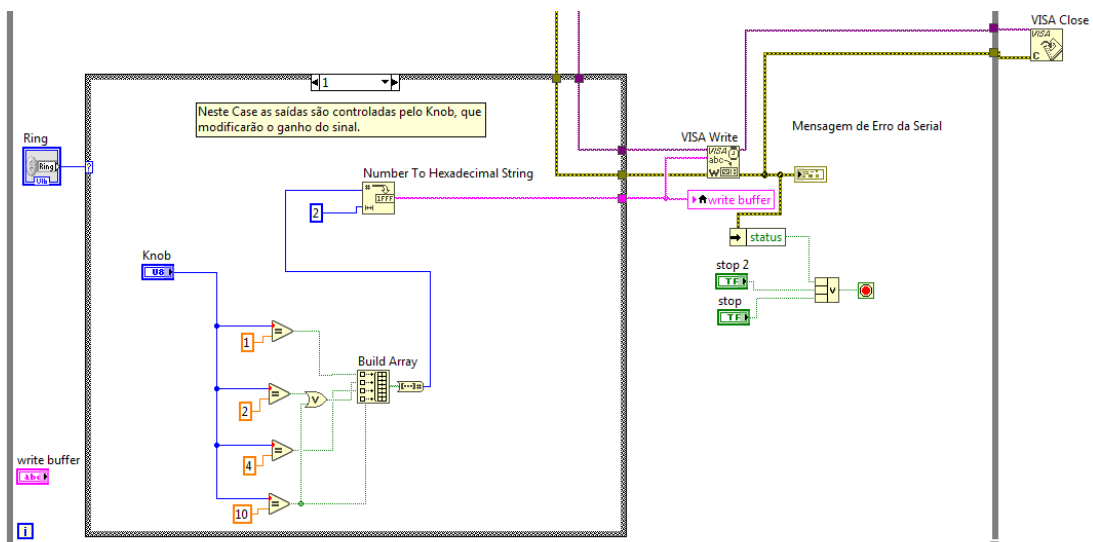


Figura 37 - Case Responsável pelo ganho do sinal (Fonte: Os autores).

Para mais detalhes da construção do software, verifique em anexos.

4. RESULTADOS

O microcontrolador está limitado a ler sinais entre 0V e 5V, portanto foi necessário construir um condicionador de sinais para deslocar o sinal somente para o campo positivo. Dessa forma o *hardware* limitou-se a fazer leituras de sinais na faixa de -2V a +2V, para leituras de sinais com amplitudes mais altas, no caso de -50V a +50V que é o máximo que o hardware está limitado a ler, montamos atenuadores.

Houve problemas quanto a comunicação com o *LabVIEW* pois para sinais com frequências maiores que 1 Hz o *LabVIEW* não apresentava corretamente o sinal que estava sendo inserido. Para sinais contínuos e constantes o *LabVIEW* apresentava o sinal corretamente.

Comparando os sinais do Osciloscópio com o sinal apresentado no *LabVIEW*. O sinal do canal 1 do osciloscópio está conectado a saída do gerador de função, e o canal 2 estão conectados ao Pino RB1 do PIC18F4550, que foi configurado para piscar o Led na frequência do sinal de onda quadrada que está sendo lido pelo canal 0 do PIC. Na Figura 38 podemos verificar o comportamento do sinal.

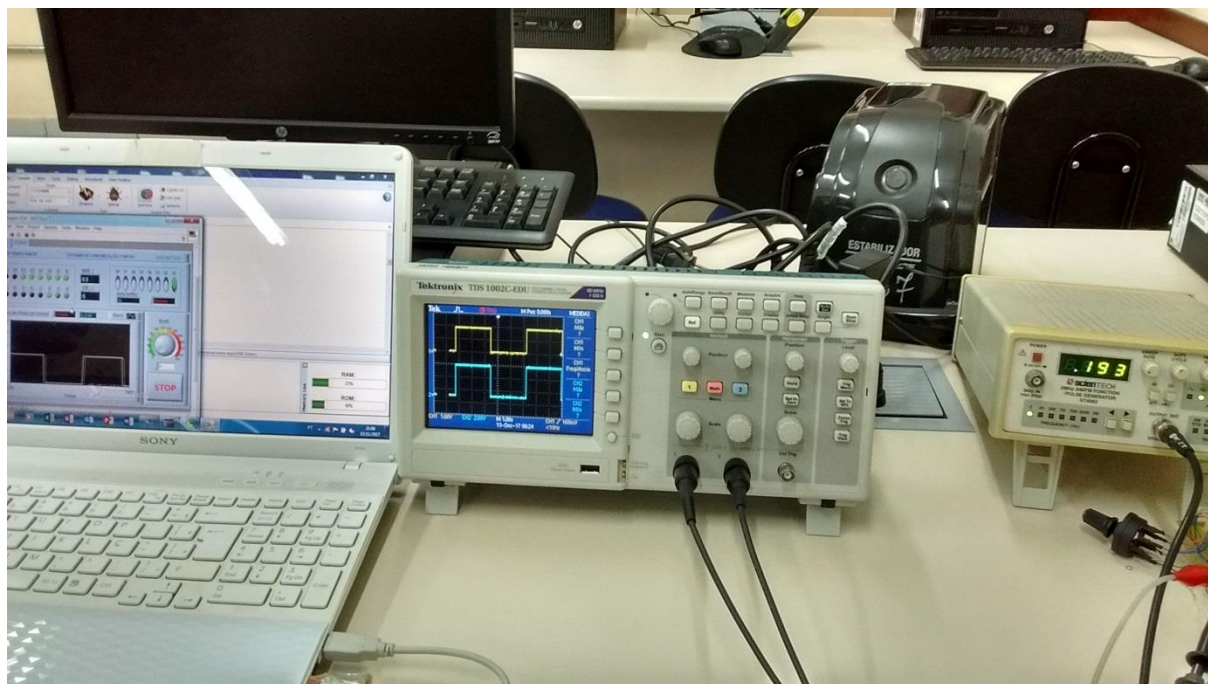


Figura 38 - Leitura e comparação da onda quadrada a 0.193Hz (Fonte: Os autores).

O sinal está sendo apresentado corretamente no *LabVIEW* em baixa frequência, nesse caso 0.193 Hz. Porém, estamos querendo ultrapassar a barreira enfrentada pelos autores do projeto da “*Plataforma Didática Para Aquisição de Dados Via Interface USB e LabVIEW com Bootloader Integrado*”, que estava limitado a 10Hz. Verificando a Figura 39 e na Figura 40 abaixo, podemos ver que o sinal a 10.71Hz já não é apresentado corretamente no *Datalogger*.

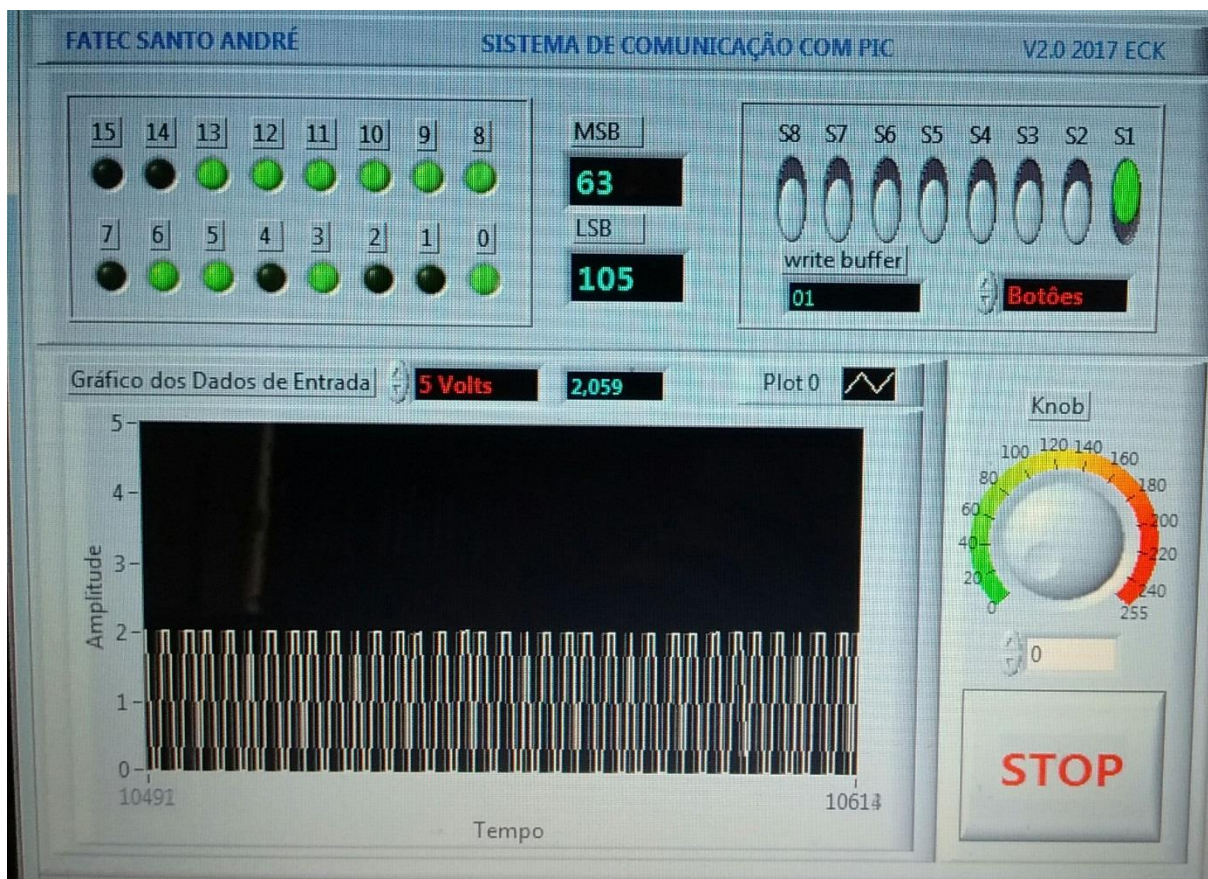


Figura 39 – Leitura da Onda quadrada a 10.71Hz no *Datalogger* (Fonte: Os autores).

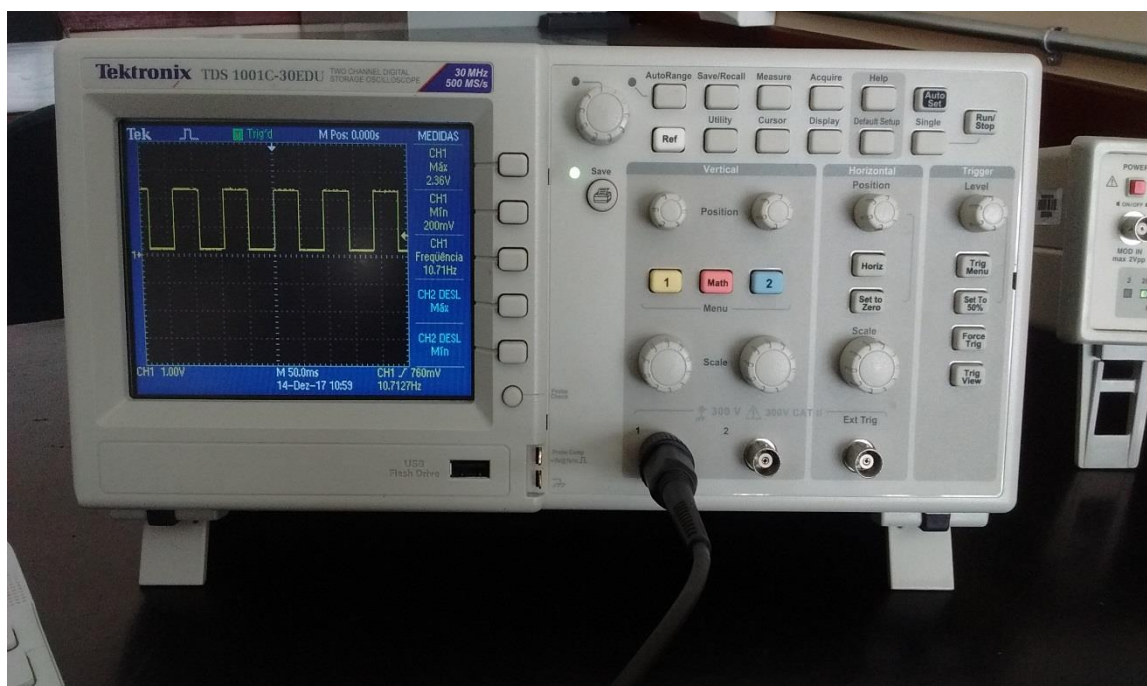


Figura 40 - Leitura onda quadrada a 10.71Hz no Osciloscópio (Fonte: Os autores).

Verificando o comportamento do sinal, dá para ver claramente que o sinal lido pelo *datalogger* nada se parece com a onda quadrada lida pelo osciloscópio Digital. A medida tomada para resolver esse problema foi criar um Vetor que armazenasse 200 amostras, sendo cada amostra coletada a 1ms. Após o Vetor ter se carregado totalmente, as amostras eram enviadas pela *USB* limitando a frequências na faixa de 5 a 60Hz, como podemos observar na Figura 41 e 42.

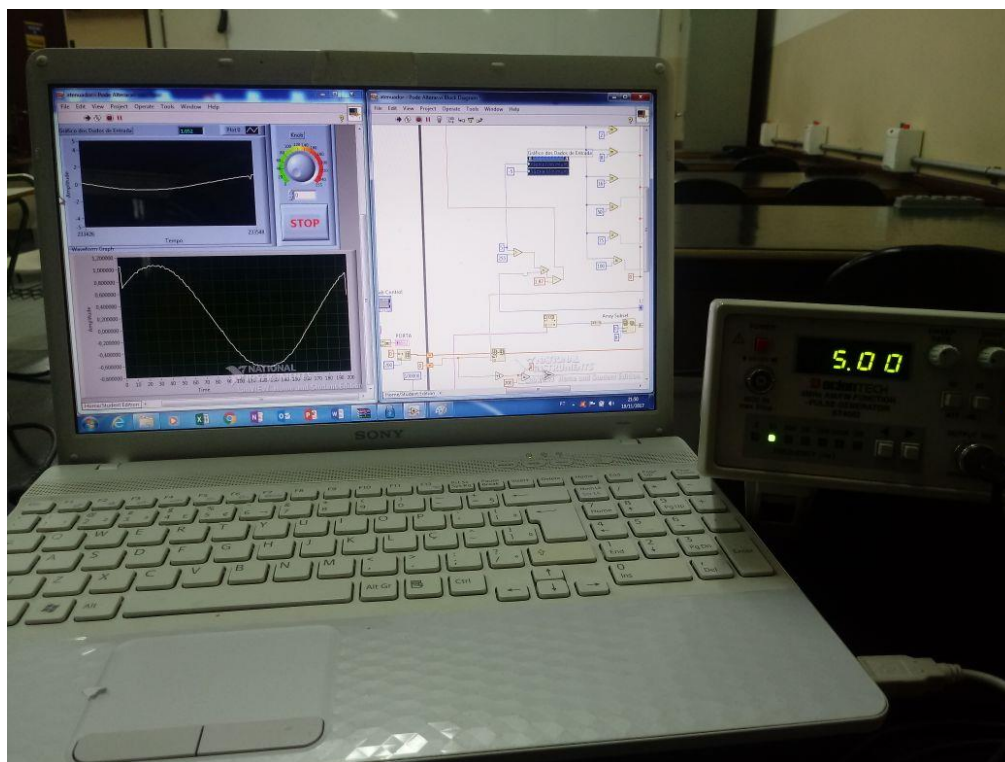


Figura 41 – Sinal de 5Hz capturado pelo Osciloscópio Digital Via LabVIEW (Fonte: Os autores).

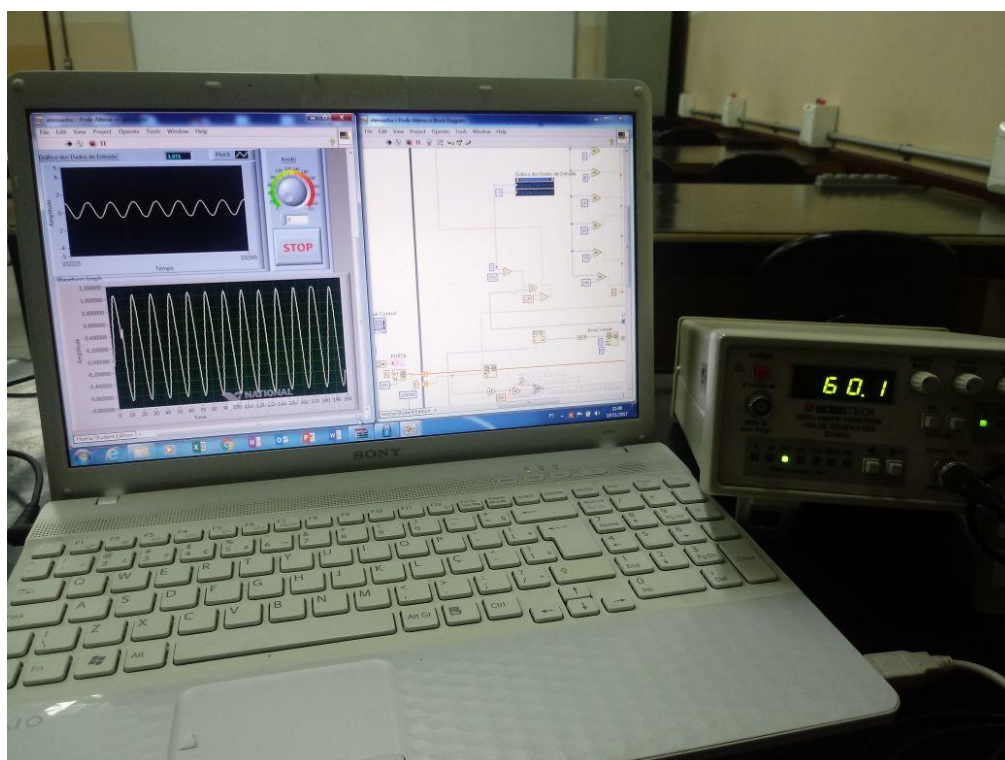


Figura 42 – Sinal de 60Hz capturado pelo Osciloscópio Digital Via LabVIEW (Fonte: Os autores).

Visto que a captura de sinais está limitada a frequências entre 5Hz e 60Hz, a essência do osciloscópio foi mantida, ou seja, ele possui atenuador e amplificador de sinais, *trigger* e conseguimos construir um *hardware* que condiciona o sinal para que possa ser lido pelo microcontrolador sem que o danifique. Abaixo na Figura 43, podemos verificar como ficou o osciloscópio didático.

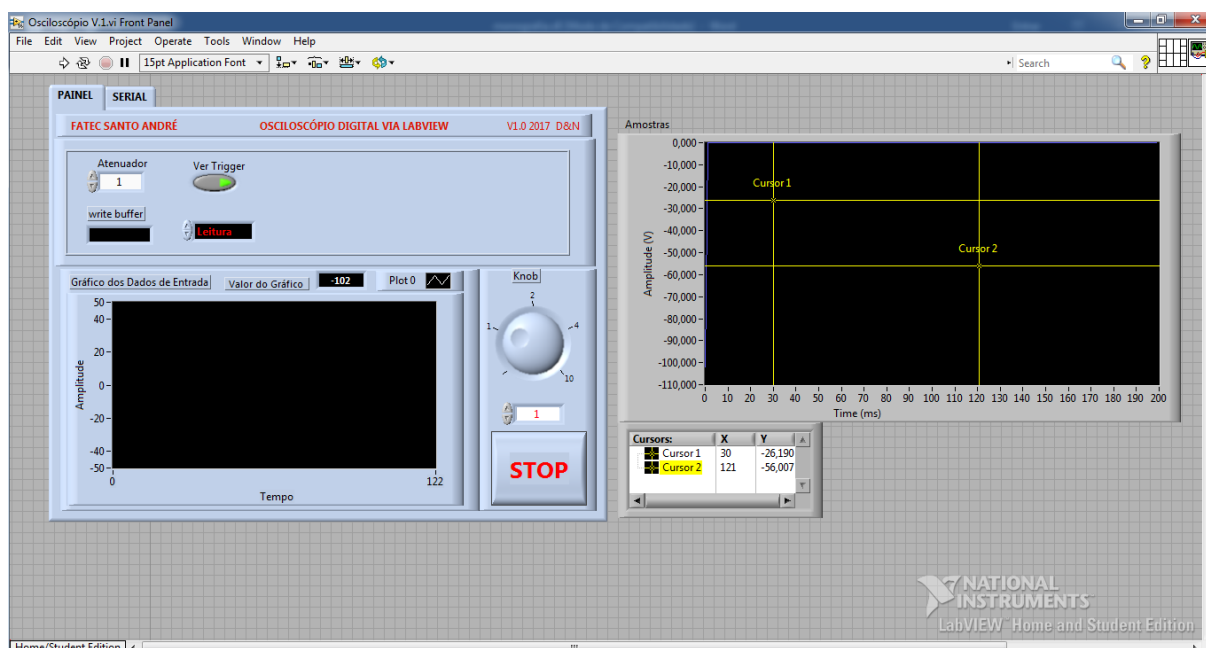


Figura 43 – Interface do Osciloscópio didático no *LabVIEW* (Fonte: Os autores).

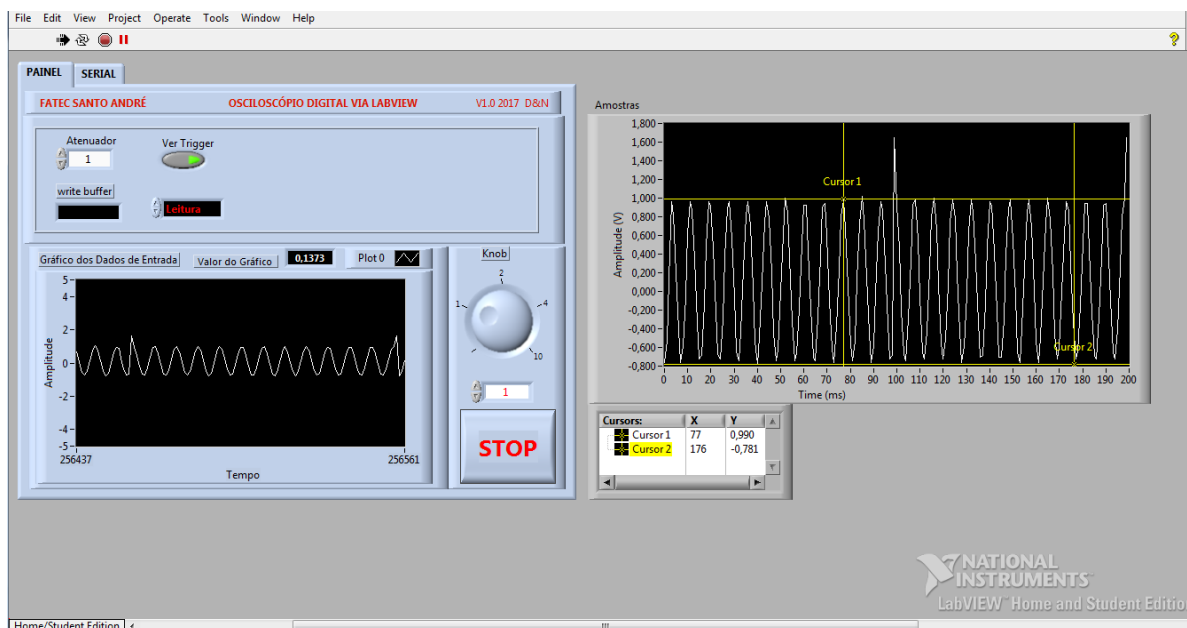


Figura 44 – Leitura sinal a 60Hz com o osciloscópio Didático (Fonte: Os autores).

O projeto apresentado obteve muitas limitações, sendo elas:

- Quando desejar utilizar o osciloscópio didático para ler sinais em torno de 100Vpp, é necessário virar a chave do atenuador para a atenuação de 100, para não danificar os CI's do *hardware* adicional e conseqüentemente o microcontrolador.
- Toda vez que necessário mudar o ganho do sinal, é necessário parar o laço *while* do *LabVIEW*, para posteriormente habilitar a leitura do sinal com o ganho desejado.
- O ajuste fino do timer 0, afim de se obter um estouro de exatamente 1ms.
- *Trigger* fixo em 1V.
- Implementação de cursores para poder efetuar as operações que mostrem o valor de Máximo, Mínimo, RMS entre outros.

A Figura a seguir é um esquemático que mostra a semelhança entre os principais sistemas que fazem parte do Osciloscópio digital e o desenvolvido no TCC.

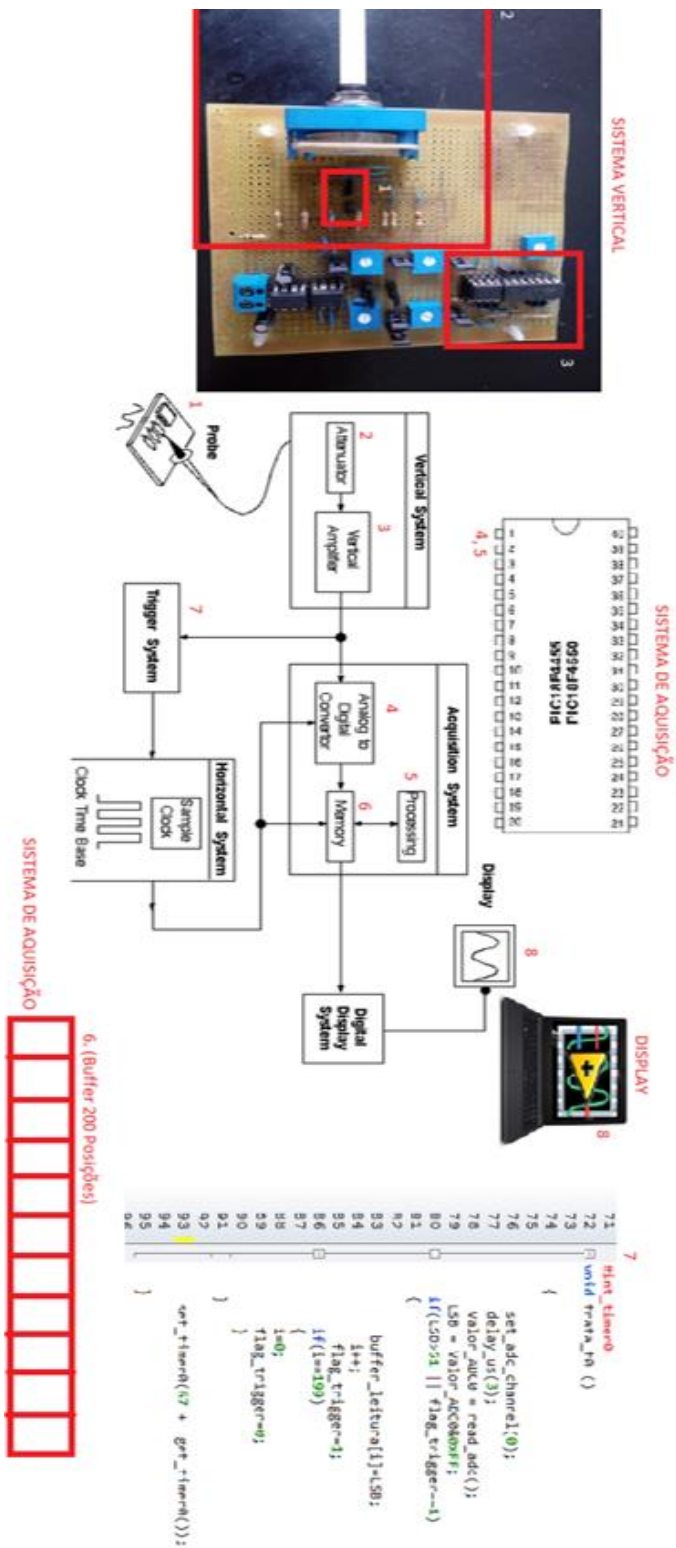


Figura 45 - Comparativo das funções do osciloscópio digital com o osciloscópio didático (Fonte: Os autores).

5. CONSIDERAÇÕES FINAIS

Nessa seção discutiremos brevemente as propostas de melhoria do projeto, bem como nossas conclusões referentes ao mesmo.

5.1 Conclusão

Baseado nos resultados obtidos, pode-se concluir que a proposta inicial de desenvolvimento de um osciloscópio didático via *LabVIEW* foi atendida. O projeto possibilita enxergar a importância dos osciloscópios e a entender o seu princípio de funcionamento, e mais ainda a importância do *software LabVIEW*, que é amplamente empregada em aplicações que deseja-se desenvolver instrumentos virtuais. Outra questão que deve ser ressaltada refere-se ao custo do projeto, que foi direcionado para a licença do *LabVIEW*, onde a versão *LabVIEW Base* possui o valor de R\$349,00/ano (2018), o *hardware* principal e adicional, ficando em torno de R\$125,00 (2017), e a licença do CCS onde a versão *Base PCWH* custa R\$1610,00 (2018). Com isso, apesar das limitações do projeto, o custo para desenvolvê-lo foi relativamente baixo, pois o custo de um Osciloscópio, tomando como exemplo o Tektronics TBS1072B tem o custo de R\$2.848,89 (2018).

A hesitação que ocorre diante de produzir algo que ainda não foi realizado ou que tenha, mas com *hardwares* melhores do que os que se possuem permitiu entender que, mais importante que isso, com um conceito muito bem definido como, princípio de funcionamento do que se deseja desenvolver, no caso o osciloscópio didático, somados aos aprendizados e habilidades conquistadas no curso, permite que se alcance grandes resultados, este trabalho, que contou com o um microcontrolador PIC e suas respectivas limitações, permitiu desenvolver um osciloscópio capaz de ler sinais de 100V pico a pico em uma frequência de 5 a 60 hertz.

5.2 Proposta de melhoria

Visto as limitações que o projeto resultou, apresentamos as seguintes sugestões para otimiza-lo:

- Estudo mais profundo quanto ao desenvolvimento da linguagem C, a fim de se obter um *software* que faça a aquisição, processamento e envio dos dados de forma mais rápida e precisa.
- Adicionar conversores A/D a parte, em conjunto com o microcontrolador, a fim de melhorar o processamento do microcontrolador.
- Estudo mais aprofundado da Ferramenta *LabVIEW*, a fim de se obter uma interface mais próxima de um osciloscópio real.
- Construir o *hardware* adicional com componentes SMD de forma que a placa do *hardware* adicional possa ser encaixada na do hardware principal
- Usar componentes mais precisos, como resistores de precisão.
- Aumentar a frequência máxima de leitura.

6. REREFÊNCIAS BIBLIOGRÁFICAS

(2013) Noteplace Notebooks. Disponível em: <<http://www.noteplace.com.br/artigo/entendendo-a-porta-serial-e-o-usb>>. Acessado em 19 de Junho de 2017.

(2014) Instituto Newton C Braga. Disponível em: <<http://www.newtoncbraga.com.br/index.php/almanaque/865-tubo-de-raios-catodicos.html>>. Acessado em 11 de Maio de 2016.

ALECRIM, Emerson (2017), **Tecnologia USB (Universal Serial Bus)**. Disponível em: <<https://www.infowester.com/usb.php>>. Acessado em 22 de Junho de 2017.

ALVES, Mário (1998), **ABC Do osciloscópio**. Disponível em: <<http://www.ceset.unicamp.br/~leobravo/TT%20305/O%20Osciloscopio.pdf>>Acessado em 11 de Maio de 2016.

ANTÔNIO, Marco, **Programação de Microcontroladores Pic usando Linguagem C**. Disponível em: <<http://www.pictronics.com.br/downloads/apostilas/Apostila-Pic-C.pdf>>. Acessado em 24 de Junho de 2017.

BASTOS, Alex, **Pic18f4550**. Disponível em: <<http://www.decom.ufop.br/alex/arquivos/bcc425/slides/PIC18f4550.pdf>>. Acessado em 22 de Junho de 2017.

Biography.com Editor, **Ferdinand Braun Biography**. Disponível em: <<http://www.biography.com/people/ferdinand-braun-9224860>>. Acessado em 11 de Maio de 2016.

BRAIN Marshall, **O tubo de raio catódico**. Disponível em: <<http://tecnologia.hsw.uol.com.br/televisao3.htm>>. Acessado em 12 de Maio de 2016.

Eletrônica total - Nº 156 (2013), **Osciloscópio: pequeno dicionário de termos técnicos em inglês**. Disponível em: <<http://user.das.ufsc.br/~elisanm/FuncoesScopeDigital.pdf>>. Acessado em 12 de Maio de 2016.

FAZANO Carlos, **Conjunto de tópicos relacionados com a evolução em caráter mundial da Eletrônica nos 100 anos do século XX**. Disponível em: <<http://www.fazano.pro.br/port82.html>>. Acessado em 11 de Maio de 2016.

KAREN, Cristiane (2009), **Medições com osciloscópio**. Disponível em: <<http://www.ebah.com.br/content/ABAAA7EAL/medicoes-com-oscioscopio>>. Acessado em 12 de Maio de 2016.

MARTIN, Raúl (2008), **Acetatos de Instrumentação e Aquisição de Sinais do ano 2007/2008**. Disponível em: <<http://nebm.ist.utl.pt/repositorio/ficheiros/372>>. Acessado em 12 de Maio de 2016.

MEDINA, Gabriel B. B (2004), **USB 1.1, USB 2.0 e Wireless USB**, Trabalho, Universidade Federal do Rio de Janeiro, Rio de Janeiro.

PEREIRA, Fábio. **Microcontroladores PIC: Programação em C**. São Paulo: Érica, 2003. 360 p.

RICARTE, Ivan Luiz Marques, **material das disciplinas EA876 (Introdução a Software de Sistema) e EA877 (Mini e microcomputadores: software) da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas (FEEC/UNICAMP)**. Disponível em:

<<http://www.dca.fee.unicamp.br/cursos/EA876/apostila/HTML/node2.html>>.

Acessado em 24 de Junho de 2017.

SAMPAIO e JUNCKER (2014), **Plataforma didática para aquisição de dados via interface USB e LabVIEW com Bootloader integrado**, Monografia, Faculdade de Tecnologia de Santo André, Santo André.

SOUZA, Kleber, JUNIOR Paulo (2006), **TCC Osciloscópios**. Disponível em:

<http://www.joinville.ifsc.edu.br/~mtaques/Medidas_Eletricas/osciloscopio%20digital.pdf>. Acessado em 11 de Maio de 2016.

7 ANEXOS

7.1 Lista de Termos técnicos e suas definições

AC: *Alternating Current*, que em português é abreviado por CA (corrente alternada), significa um sinal que varia de intensidade de forma repetitiva com o tempo.

AC Coupling: (*Acoplamento AC*). Trata-se de uma forma de acoplamento de sinal que bloqueia a componente DC (contínua de um sinal) de modo que só sua parte alternada (que varia) seja observada. Uma tensão de *ripple*, por exemplo, pode ser melhor observada com este recurso.

Accuracy: (Precisão). Refere-se à maneira como se obtém a indicação do valor de um sinal, o mais próximo possível do real.

ADC: *Analog-to-Digital Converter* (Conversor Analógico para Digital). Trata-se de circuito ou componente que converte valores analógicos de uma grandeza em sinais digitais binários.

Acquisition: (Aquisição). É o processo de amostragem dos sinais de uma entrada para a sua digitalização.

Aliasing: (Falsificando). Trata-se da apresentação falsa de um sinal na tela de um osciloscópio digital pela velocidade de amostragem insuficiente, quando se trata de altas frequências. A velocidade de amostragem é baixa demais para visualizar o sinal de entrada.

Alternate Mode: (Modo Alternado). Trata-se de uma modalidade de operação em que o osciloscópio logo que termina de traçar o sinal de um canal, traça o sinal do outro.

Attenuation: (Atenuação). O grau de amplitude em que o sinal é reduzido quando passa por um circuito. Por exemplo, se uma ponta de prova tem uma atenuação x100, isso significa que ela reduz ou atenua a amplitude de um sinal de entrada (tensão, por exemplo) em 100 vezes.

Automatic Trigger Mode: (Modo de disparo automático). Trata-se de um modo de disparo que faz com que o osciloscópio comute automaticamente, mesmo quando não existem pontos do sinal que possam ser usados para esta finalidade (sincronismo).

Auto-set: (Autofixação). Trata-se de uma função em que o osciloscópio encontra automaticamente uma forma de imagem estável, de tamanho apropriado que

possibilite sua melhor observação. O *auto-set* ajusta automaticamente o número de volts por divisão e o tempo por divisão.

Average: (Média). Trata-se de uma modalidade de operação em que o osciloscópio faz diversas amostragens de um sinal, processando-o e eliminando o ruído aparente e depois apresenta um sinal que é a média de todas as aquisições ou amostragens.

Bandwidth: (Faixa passante). Trata-se da especificação do ganho em função da faixa de frequências em que o osciloscópio pode operar. As frequências superior e inferior desta faixa são aquelas em que corresponde a 3dB. Muitos osciloscópios possuem como frequência inferior desta faixa de sinais DC, ou seja, 0.

Channel: (Canal). Trata-se da entrada ou seleção de sinais que pode variar entre 1 e 4, conforme o equipamento.

CRT: (TRC). É o *Cathode Ray Tube*, ou Tubo de Raios Catódicos, em português. É o elemento principal dos osciloscópios tradicionais e consiste na “válvula” que, através de feixe de elétrons, projeta a imagem dos sinais a serem observados numa tela de fósforo.

Chop Mode: (Modo Comutado). Trata-se de um modo de apresentação do sinal em que pequenas porções de diversos sinais são projetadas simultaneamente de modo que estes sinais possam ser observados ao mesmo tempo.

Circuit Loading: (Carga do circuito). É o efeito que a ligação do osciloscópio em um circuito tem sobre ele, “carregando-o” e com isto alterando eventualmente o próprio sinal que será observado.

Compensation: (Compensação). Trata-se de um ajuste que deve ser feito nas pontas de prova x10 para compensar os efeitos da capacitância apresentada pelos circuitos de entrada do osciloscópio.

Coupling: (Acoplamento). É o modo segundo o qual os circuitos são interligados. O acoplamento é direto quando a sua ligação é feita por um condutor, mas pode ser feito através de componentes como capacitores, transformadores, etc, caso em que temos acoplamentos capacitivos, indutivo, etc.

DC (Direct Current): Ou corrente contínua (CC), em português. Um sinal que não varia de intensidade com o tempo.

DC Coupling: (Acoplamento DC). Trata-se de um modo de acoplamento para os sinais de entrada de um osciloscópio que permite observar os sinais alternados (AC) e a componente contínua (DC).

Delay by Time: (Atraso pelo tempo). Trata-se de um modo de disparo (*trigge-ring*) em que o ponto de disparo da varredura ou aquisição é atrasado em relação a um sinal de disparo de referência constante do sinal a ser observado.

Delay Measurement: (Medida de atraso). É a medida do tempo de atraso entre os pontos de cruzamento por zero e dois sinais.

Delay Time: (Medida de atraso). Tempo que ocorre entre um evento e a aquisição (ou observação) deste evento.

Differentiate: (Diferencie). É a capacidade que certos osciloscópios possuem de apresentar na tela a forma de onda de um sinal que corresponde à derivada da função correspondente ao sinal original. Matematicamente, esta derivada corresponde à apresentação das variações do sinal ou “*slew rate*”.

Digitizing: (Digitalizando). É o processo segundo o qual um sinal analógico é convertido em sinais digitais, representando a amplitude instantânea do sinal a cada amostragem.

Division: (Divisão). Marcação da graticula do osciloscópio, que permite a realização de medidas de amplitude e tempo.

Earth Ground: (Terra). É a conexão que permite desviar para a terra correntes elevadas.

Edge Trigger: (disparo pela borda). Neste modo de operação o disparo ocorre quando o osciloscópio detecta a passagem do sinal por um nível de tensão determinado numa dada direção (denominado *trigger slope*).

Envelope: (envolvente). Nesta modalidade o disparo do osciloscópio detecta os pontos de máximo e mínimo de um sinal.

Equivalent time sampling: (tempo de amostragem equivalente). Nos osciloscópios digitais a imagem é obtida por amostragens sucessivas do sinal onde os valores instantâneos são convertidos para a forma digital. No modo *random* equivalente time (tempo equivalente aleatório) o osciloscópio amostra continuamente o sinal, independente do sincronismo.

Fall time: (tempo de descida). É uma medida que é feita num sinal e que é equivalente ao tempo que ocorre entre o instante em que o sinal tem 90% de sua amplitude máxima e apenas 10% da mesma.

Fft (fast fourier transform): A transformada de Fourier Rápida é uma função que possibilita ao osciloscópio transformar uma forma de onda de um display (tomada como amplitude versus tempo) nas amplitudes das diversas frequências em que a forma de onda pode ser decomposta. Com esse procedimento é possível medir a composição harmônica de uma forma de onda, a sua distorção, analisar filtros e identificar fontes de ruído em circuitos digitais.

Focus: (foco). É o controle do osciloscópio que ajusta o tamanho do ponto da imagem gerado pelo feixe de elétrons, e assim o foco da imagem.

Frequency: (frequência). Número de vezes em que um ciclo do sinal se repete em um segundo. Medida de *Hertz*.

Gated measurements: (medidas gatilhadas). Recurso de um osciloscópio que possibilita escolher numa forma de onda as partes que devem ser observadas.

Glitch: trata-se de um erro intermitente de um circuito

Gpib (general purpose interface bus): o barramento de interface de uso geral é um protocolo que possibilita a conexão de diversos instrumentos em uma rede sob controle de um computador também conhecido como barramento IEEE 488, permite a transferência de dados de oito linhas de controle e cinco linhas adicionais.

Graticule: (graticula). Grade de tela do osciloscópio que determina os eixos horizontal e vertical.

Hardy copy: (cópia impressa) recurso que possibilita a impressão ou cópia em papel da forma de onda observada no osciloscópio.

Hf reject: (rejeição de alta frequência). Este recurso permite a remoção das componentes de alta frequência do sinal de disparo. Desta forma, somente os componentes de baixa frequência passam para o sistema de disparo para controlar a aquisição de dados, este recurso atenua normalmente sinais acima de 3khz.

Hodolf trigger: esta especificação corresponde ao tempo que ocorre entre o sinal de disparo e o instante em que ele pode aceitar um novo sinal para isso. Este tempo determina estabilidade da imagem.

Instavu acquisition mode: tratasse de um modo de aquisição de dados muito rápido que possibilita a amostragem de 400.000 formas de onda por segundo.

Integrate:(integrar) tratasse do recurso encontrado em alguns osciloscópios que permite a visualização da forma de onda correspondente a integral da função do sinal principal. A integral do sinal possibilita a visualização da energia e potencia envolvida neste sinal.

Interpolation: (interpolação). Tratasse de um modo de processamento de sinal em que o osciloscópio estima as linhas que unem os pontos de amostragem gerados de forma a projetar a forma de onda original como uma linha continua. Os osciloscópios trabalham com tipos de interpolação: linear e $\text{sen}(x)/x$.

Lf rejection: (rejeição de baixa frequência). Nesta modalidade de operação o osciloscópio remove as componentes de baixa frequência de um sinal que esta sendo observado evitando que elas atuem sobre o gatilhamento. Os osciloscópios comuns rejeitam frequências abaixo de 80 khz, quando nesta modalidade de operação.

Logic state trigger: (disparo por estado logico). Nesta modalidade de disparo ele é feito quando é detectada certa combinação de níveis lógicos na entrada.

Math: (matemática). Capacidade que os osciloscópios mais avançados possuem de manipular matematicamente um sinal. As funções básicas disponíveis neste caso são de soma, subtração, divisão, multiplicação e inversão.

Mean:(médio). Medida da média dos valores de amplitude de um sinal em um ciclo completo.

Megasamples per second: (MS/s. Milhões de amostragens por Segundo). É a quantidade de amostragens que um osciloscópio faz para projetar o sinal no *display*.

Negative duty cycle: (ciclo ativo negativo). Medida de tempo que corresponde a relação entre a largura de pulso negativo em relação ao período completo de um sinal. Este valor é expresso como uma porcentagem.

Negative width: (largura negativa). Medida de tempo que corresponde ao tempo entre dois pontos de determinada amplitude de um sinal. Normalmente este tempo corresponde a 50% da amplitude.

Noise reject: (rejeição de ruído). A rejeição de ruído é usada para reduzir a sensibilidade do circuito de disparo.

Normal trigger mode: (modo de disparo norma). Nesta modalidade o disparo (ou gatilhamento) é feito quando o osciloscópio detecta o sinal a ser observado com um pulso usado para esta finalidade.

Noise: (ruído). Tensão indesejável no sinal que vai ser observado.

Peak: (pico). É o valor máximo que uma tensão (vp) ou corrente (ip) atinge em um ciclo de um sinal analisado.

Peak to peak: (pico a pico). É a diferença de valores (amplitude) entre os pontos de máximo e de mínimo e um ciclo do sinal a ser analisado.

Peak detection: (detecção de pico). É um modo de aquisição de dados do osciloscópio digitais que possibilita a visualização dos pontos de máximo e mínimo de um sinal.

Phase: (fase). Medida da diferença entre os pontos de mesma amplitude relativamente entre dois sinais de mesma frequência, ou de frequências harmônicas medidas em graus.

Positive duty cycle: (ciclo ativo positivo). Relação entre o tempo de duração da parte (pulso) positiva de um sinal e o tempo total (período) do mesmo, expressa na forma de uma porcentagem.

Post trigger: (disparo posterior). Parte de uma forma de onda depois do disparo que contém informações.

Pre trigger. (disparo anterior). Parte de uma forma de onda antes do disparo que contém informações.

Probe: (ponta de prova). Dispositivo usado para ligar o osciloscópio ao circuito em teste.

Probe compensation: (compensação da ponta de prova). Ajusta usado para melhorar a resposta de baixa frequência de uma ponta de prova.

Pulse trigger. (disparo de pulso). Forma de gatilhamento de um osciloscópio que ocorre quando ele detecta um pulso no sinal.

Quantizing: (quantização). Processo usado para converter uma grandeza analógica (tensão ou corrente) em um valor digital.

Real time sampling: (amostragem em tempo real). Modalidade em que o osciloscópio é rápido o suficiente para produzir uma imagem em apenas um ciclo de amostragem.

Record length: (tempo de gravação). Número de pontos de uma forma de onda usado para gerar um sinal.

Reference memory: (memória de referência). Nos osciloscópios digitais a memória de referência é usada para armazenar as formas de onda que podem ser observadas posteriormente para análise.

Rise time: (tempo de subida). Tempo que decorre entre o instante que o sinal tem 10% de sua amplitude máxima e 90% de sua amplitude máxima.

Rms: (*root mean square*) valor médio quadrático de uma forma de onda.

Run model: nesta modalidade de operação, o sinal é apresentado na tela antes mesmo de se completar um ciclo de amostragem. Este modo de operação é interessante quando se observa sinais de frequências muito baixas.

Runt trigger. o termo “*runt*” se refere de maneira popular ao filhote menor de uma ninhada de animais. Tecnicamente, para os osciloscópios trata-se de um pulso muito pequeno para ser detectado em amostragens seguidas.

Sample: (amostra). É o valor do sinal no processo de amostragem convertido para forma original.

Sample interval: (intervalo de amostra). É o intervalo entre duas amostragens sucessivas.

Sampling: (amostragem). É o processo segundo o qual o osciloscópio mede valores instantâneos da forma de onda a ser projetada, convertendo-os para forma digital.

Screen: (tela). É a superfície na qual as imagens das formas de onda ou sinais são projetadas.

Setup/hold: (fixar e manusear). Trata-se de uma modalidade de disparo ou gatilhamento em que o osciloscópio dispara quando a fonte de dados muda de estado a partir de um valor fixado, ou tempo com base em um *clock*.

Sinal processing: (processamento de sinal). Para o caso dos osciloscópios, refere-se as unções que possibilitam trabalhar as formas de onda capturadas como, por exemplo, para medidas de sinal, FFTs, integração, diferenciação, etc.

Single shop: (pulso único). Trata-se de um sinal que acontece apenas uma vez ou transiente.

Single sweep: (varredura única). Nesta modalidade o sinal é projetado por uma única varredura.

Single rate: (taxa de crescimento). É a velocidade com que a amplitude de um sinal cresce e decresce.

Slop: (inclinação). A direção para onde tende um ponto da forma de onda.

Sweep: (varredura). Movimento do feixe de elétrons na tela de um osciloscópio no sentido de gerar a imagem correspondente a forma de onda.

Sweep speed: (velocidade de varredura). Velocidade com que se movimenta na tela o feixe de elétrons para gerar a imagem de forma de onda ser observada.

Time base: (base de tempo). Circuito do osciloscópio que controla a velocidade de varredura horizontal.

Time base accuracy: (precisão de base de tempo). Valor que permite determinar a precisão com que períodos e frequências são medidos com um osciloscópio.

Time blase range: (faixa de base de tempo). Faixa de valores entre o mínimo e o máximo alcançados pela base de tempo.

Timeout trigger: (disparo fora do tempo). Modo de operação em que o disparo ou gatilhamento ocorre quando o osciloscópio não encontra um pulso de disparo no sinal, depois de um tempo previsto.

Trace: (traço). Linha traçada pelo feixe de elétrons ao formar a figura na tela do osciloscópio.

Transiente: (transiente). Sinal normalmente de curta duração, que ocorre apenas uma vez. Também “denominado single shot”

Trigger: (gatilho). Circuito que inicia a varredura horizontal de um osciloscópio e que determina o ponto de início da forma de onda projetada.

Trigger holdoff: controle que existe no osciloscópio e que faz com que o circuito deixe de procurar um nível de disparos após um certo tempo especificado, de pois que determina uma forma de onda.

Trigger level (nível de disparo). É a tensão em que o circuito de disparo ou gatilhamento entra em ação.

Vertical bars cursor: (cursores de barra vertical). São duas barras verticais que são posicionadas na tela para que se possa medir na tela o intervalo de tempo entre os pontos de uma forma de onda. Os osciloscópios que possuem esse recurso apresentam de forma numérica o tempo entre as posições da barra na forma de onda.

Vertical accuracy: (precisão vertical). É a precisão com que se pode medir a amplitude de um sinal observado. **Vertical resolution:** (resolução vertical).

Vertical sensitivity: (sensibilidade vertical). É dada pelo número de bits usados para a amostragem de um sinal. Para uma digitalização de 8 bits temos 255 níveis possíveis de resolução.

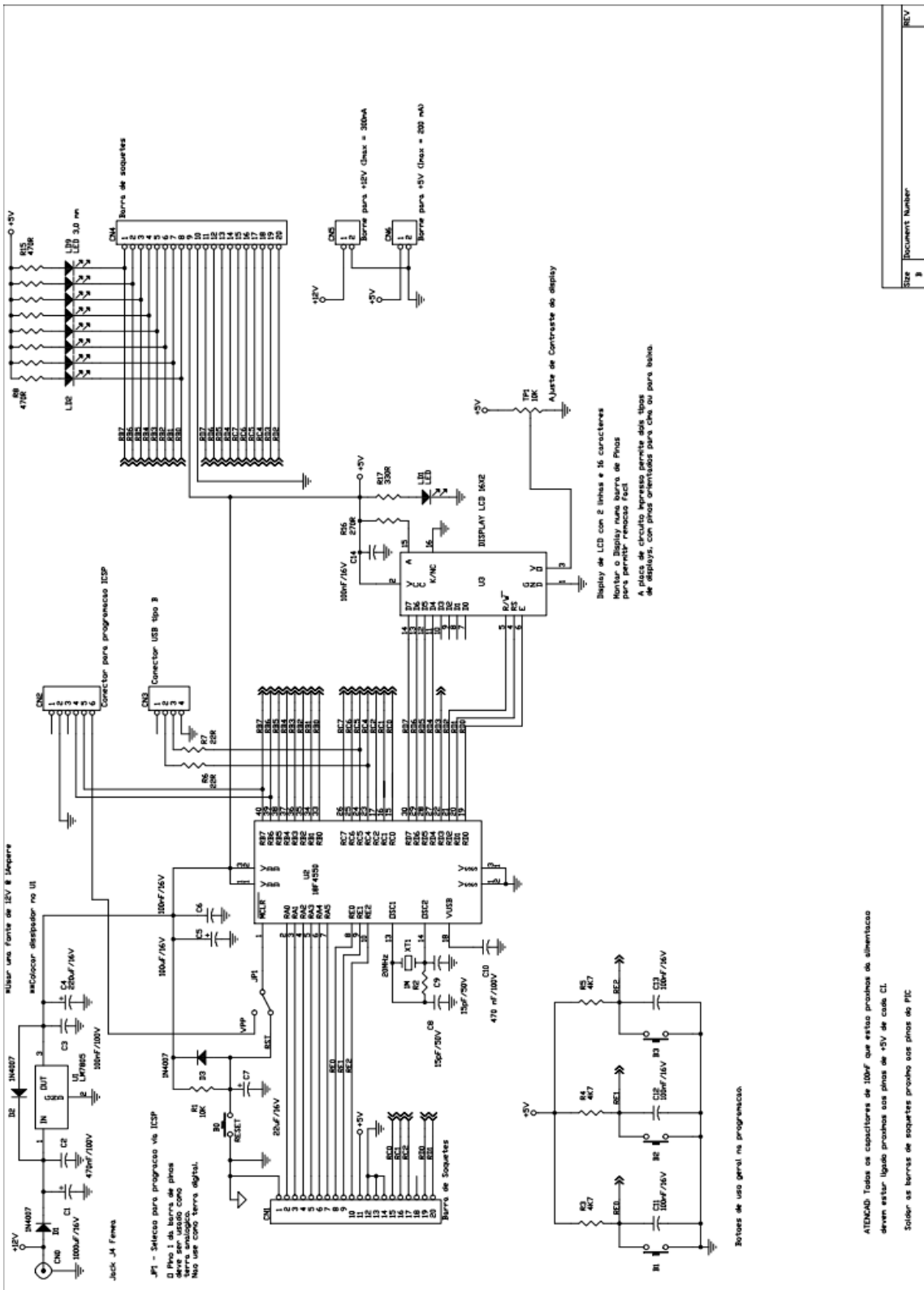
Waveform: (forma de onda). É a representação gráfica de um sinal.

Xy format: (formato xy). É um modo de apresentação de formas de onda que compara as tensões de duas formas de onda, ponto por ponto. Esta modalidade é interessante para se estudar a relação de fase, ponto a ponto, entre duas formas de onda.

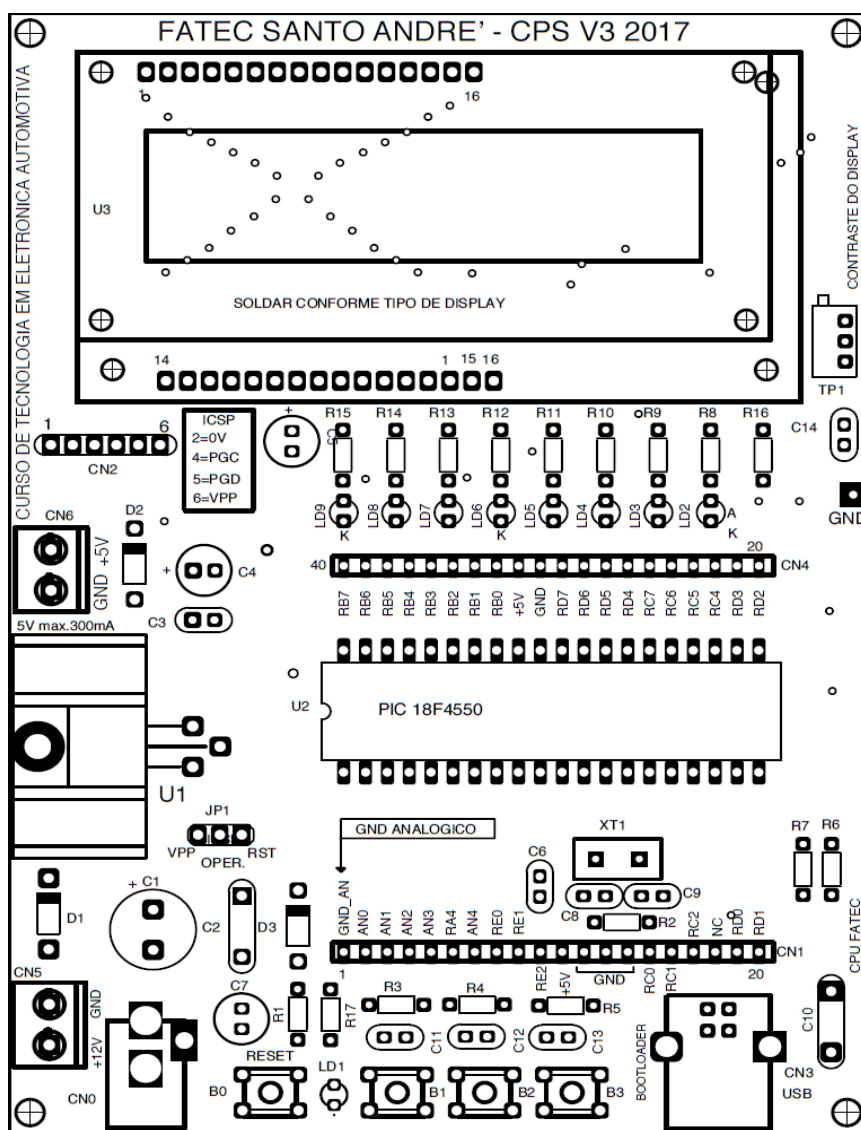
Yt format: (formato yt). É o formato convencional usado pelos osciloscópios em que temos um eixo vertical (Y) que representa a amplitude da grandeza analisada, e um eixo horizontal de tempo (T) que corresponde a varredura.

z-axis: +(eixo z). é uma entrada de um sinal que permite controlar a intensidade ou (brilho) do feixe de elétrons que incide na tala.

7.1 Esquemático do Hardware Principal

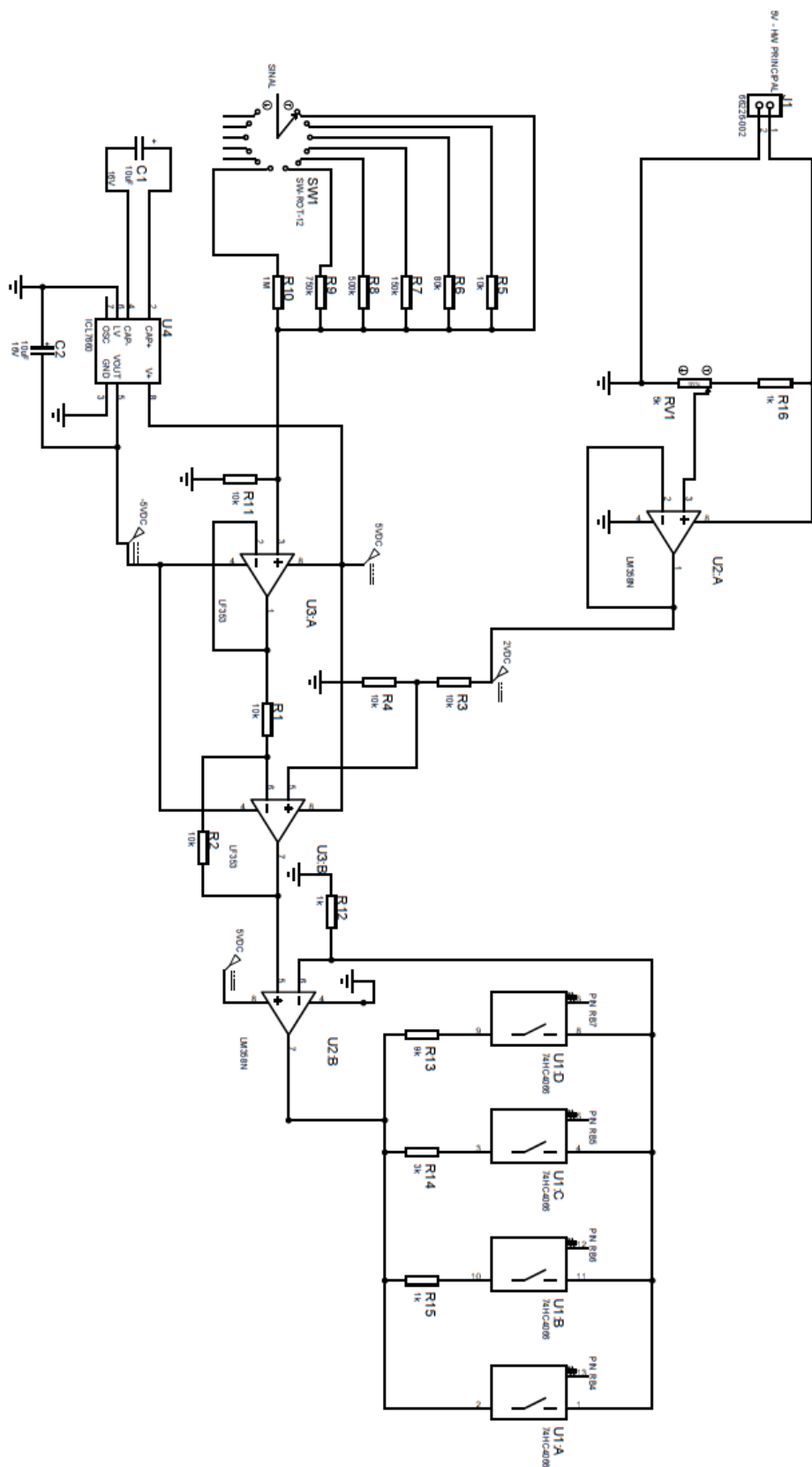


7.2 Layout do Hardware Principal

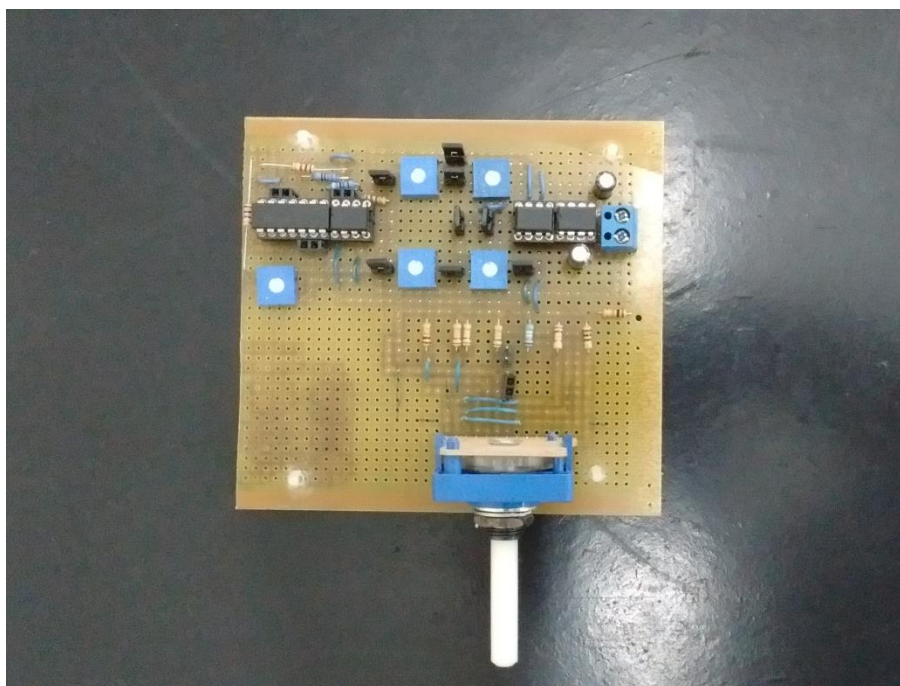


7.3 Lista de Componentes do *Hardware* Principal

Quantidade	Label	Componente	Especificação	Fabricante	Custo	Fornecedor	Subtotal
1	U1	LM 7805	Regulador de 5V	Vários	R\$ 1,26	www.baudaeletronica.com.br	R\$ 1,26
1	U2	18F4550	Microcontrolador	Microchip	R\$ 37,81	www.soldafria.com.br	R\$ 37,81
1	U3	Display LCD 16X2	LCD Azul	Vários	R\$ 19,71	www.baudaeletronica.com.br	R\$ 19,71
1	D1	1N4007	Diodo de Retificação	Vários	R\$ 0,07	www.soldafria.com.br	R\$ 0,07
1	D2	1N4007	Diodo de Retificação	Vários	R\$ 0,07	www.soldafria.com.br	R\$ 0,07
1	D3	1N4007	Diodo de Retificação	Vários	R\$ 0,07	www.soldafria.com.br	R\$ 0,07
9	LD1-LD9	LED 3,0 mm	Led	Vários	R\$ 0,36	www.baudaeletronica.com.br	R\$ 3,24
1	C1	1000 µF/16V	Capacitor Eletrolítico Radial		R\$ 0,45	www.baudaeletronica.com.br	R\$ 0,45
1	C2	470 nF/100V	Capacitor de Poliéster metalizado		R\$ 0,26	www.soldafria.com.br	R\$ 0,26
1	C3	100 nF/100V	Capacitor de Poliéster metalizado		R\$ 0,26	www.soldafria.com.br	R\$ 0,26
1	C4	220 µF/16V	Capacitor Eletrolítico Radial		R\$ 0,27	www.baudaeletronica.com.br	R\$ 0,27
1	C5	100 µF/16V	Capacitor Eletrolítico Radial		R\$ 0,27	www.baudaeletronica.com.br	R\$ 0,27
1	C6	100 nF/16V	Capacitor de Cerâmica		R\$ 0,05	www.baudaeletronica.com.br	R\$ 0,05
1	C7	22 µF/16V	Capacitor Eletrolítico Radial		R\$ 0,05	www.baudaeletronica.com.br	R\$ 0,05
1	C8	15 pF/50V	Capacitor de Cerâmica		R\$ 0,05	www.baudaeletronica.com.br	R\$ 0,05
1	C9	15 pF/50V	Capacitor de Cerâmica		R\$ 0,05	www.baudaeletronica.com.br	R\$ 0,05
1	C10	470 nF/100V	Capacitor de Poliéster metalizado		R\$ 0,26	www.soldafria.com.br	R\$ 0,26
1	C11	100 nF/16V	Capacitor de Cerâmica		R\$ 0,05	www.baudaeletronica.com.br	R\$ 0,05
1	C12	100 nF/16V	Capacitor de Cerâmica		R\$ 0,05	www.baudaeletronica.com.br	R\$ 0,05
1	C13	100 nF/16V	Capacitor de Cerâmica		R\$ 0,05	www.baudaeletronica.com.br	R\$ 0,05
1	C14	100 nF/16V	Capacitor de Cerâmica		R\$ 0,05	www.baudaeletronica.com.br	R\$ 0,05
1	R1	10K 1/4W	Resistor de Carbono		R\$ 0,09	www.baudaeletronica.com.br	R\$ 0,09
1	R2	1M 1/4W	Resistor de Carbono		R\$ 0,09	www.baudaeletronica.com.br	R\$ 0,09
1	R3	4K7 1/4W	Resistor de Carbono		R\$ 0,09	www.baudaeletronica.com.br	R\$ 0,09
1	R4	4K7 1/4W	Resistor de Carbono		R\$ 0,09	www.baudaeletronica.com.br	R\$ 0,09
1	R5	4K7 1/4W	Resistor de Carbono		R\$ 0,09	www.baudaeletronica.com.br	R\$ 0,09
1	R6	22R 1/4W	Resistor de Carbono		R\$ 0,09	www.baudaeletronica.com.br	R\$ 0,09
1	R7	22R 1/4W	Resistor de Carbono		R\$ 0,09	www.baudaeletronica.com.br	R\$ 0,09
8	R8 - R15	470R 1/4W	Resistor de Carbono		R\$ 0,09	www.baudaeletronica.com.br	R\$ 0,72
1	R16	270R 1/4W	Resistor de Carbono		R\$ 0,09	www.baudaeletronica.com.br	R\$ 0,09
1	R17	330R 1/4W	Resistor de Carbono		R\$ 0,09	www.baudaeletronica.com.br	R\$ 0,09
1	TP1	10K	Trimpot	Bourns	R\$ 1,35	www.baudaeletronica.com.br	R\$ 1,35
1	CN0	Conector J4 DC-005	Conector para placa		R\$ 0,46	www.soldafria.com.br	R\$ 0,46
1	U1a	Dissipador 183001/15	Dissipador de Alumínio		R\$ 0,97	www.soldafria.com.br	R\$ 0,97
1	CN1	Barra de Soquetes 180	Barra com 40 pinos		R\$ 0,17	www.soldafria.com.br	R\$ 0,17
1	CN2	Barra de Soquetes 180	Barra com 40 pinos		R\$ 0,17	www.soldafria.com.br	R\$ 0,17
1	CN3	Conector USB Fêmea tipo B			R\$ 1,03	www.soldafria.com.br	R\$ 1,03
1	CN4	Barra de Soquetes 180 graus	Barra com 40 pinos		R\$ 0,17	www.soldafria.com.br	R\$ 0,17
1	U3a	Barra de Soquetes 180 graus	Barra com 40 pinos		R\$ 0,17	www.soldafria.com.br	R\$ 0,17
1	U3b	Barra de Pinos 180 graus			R\$ 0,44	www.soldafria.com.br	R\$ 0,44
1	CN5	Borne de 2 polos KF-301 2T			R\$ 0,68	www.baudaeletronica.com.br	R\$ 0,68
1	CN6	Borne de 2 polos KF-301 2T			R\$ 0,68	www.baudaeletronica.com.br	R\$ 0,68
2	U2a	Soquete DIP 40 pinos	Soquete para CI Pinos Torneados		R\$ 4,05	www.baudaeletronica.com.br	R\$ 8,10
1	JP1	Jumper MKBL com Aba					
1	XT1	Cristal de 20,0 Mhz					
4		Botões TACT	Comprar conforme o tamanho do layout				
						Total	R\$ 80,32

7.4 Esquemático do *Hardware* Adicional

7.5 *Layout do Hardware Adicional*



7.6 Lista de Componentes do *Hardware* Adicional

Quantidade	Label	Componente	Especificação	Fabricante	Custo	Fornecedor	Subtotal
3		1K 1/4W	Resistor de Carbono		R\$ 0,05	www.soldafria.com.br	R\$ 0,15
1		3K 1/4W	Resistor de Carbono		R\$ 0,05	www.soldafria.com.br	R\$ 0,05
2		9K1 1/4W	Resistor de Carbono		R\$ 0,05	www.soldafria.com.br	R\$ 0,10
2		10K 1/4W	Resistor de Carbono		R\$ 0,05	www.soldafria.com.br	R\$ 0,10
3		150K 1/4W	Resistor de Carbono		R\$ 0,05	www.soldafria.com.br	R\$ 0,15
1		500K 1W	Resistor de Carbono		R\$ 0,30	www.soldafria.com.br	R\$ 0,30
1		750K 1/4W	Resistor de Carbono		R\$ 0,05	www.soldafria.com.br	R\$ 0,05
1		1M 1/4W	Resistor de Carbono		R\$ 0,05	www.soldafria.com.br	R\$ 0,05
4		5K	Trimpot		R\$ 1,28	www.soldafria.com.br	R\$ 5,12
1		10K	Trimpot		R\$ 1,21	www.soldafria.com.br	R\$ 1,21
1		10uF 25V	Capacitor Eletrolítico		R\$ 0,11	www.soldafria.com.br	R\$ 0,11
1		LF353	Amplificador Operacional		R\$ 1,34	www.soldafria.com.br	R\$ 1,34
1		LM358N	Amplificador Operacional		R\$ 0,43	www.soldafria.com.br	R\$ 0,43
1		CD4066	Circuito Integrado		R\$ 0,94	www.baudaeletronica.com.br	R\$ 0,94
1		ICL 7660	Gerador de Tensão Negativa		R\$ 5,39	www.tiggercomp.com.br	R\$ 5,39
1		1 Polo 12 Posições	Chave Seletora		R\$ 15,00	www.mercadolivre.com.br	R\$ 15,00
4		Espaçador 3.0mm			R\$ 0,30	www.eletronicacastro.com.br	R\$ 1,20
8		Mini Jumper MCBL 003-A	Femêa		R\$ 0,16	http://www.eletronicacastro.com.br	R\$ 1,28
1		Barra de Pinos Macho	1x40 Vias		R\$ 0,45	www.soldafria.com.br	R\$ 0,45
1		Soquete De Barra MCI 180	1x40 Vias		R\$ 0,91	www.soldafria.com.br	R\$ 0,91
1		Placa Padrão Ilhada	10cmx15cm		R\$ 10,52	www.soldafria.com.br	R\$ 10,52
						Total	R\$ 44,85

7.7 Código Fonte em C

```
/* Modificação do LabVIEWProf. Edson - 2017 para Osciloscópio Didático Natanael -  
Douglas - 2017*/
```

```
#include <18F4550.h>
```

```
//*****
```

```
#device ADC=8
```

```
#fuses HSPLL,NOWDT,NOPROTECT,NOLVP,NODE-  
BUG,NOCPD,PUT,NOWRT,BROWNOUT,USBDIV,PLL4,CPUDIV1,VREGEN,NOP-  
BADEN
```

```
#use delay(clock=48000000)
```

```
#use fast_io(a)
```

```
#use fast_io(b)
```

```
#use fast_io(c)
```

```
#use fast_io(d)
```

```
#use fast_io(e)
```

```
/* Configuração da Porta D para Acessar o LCD */
```



```
#include <usb_cdc.h>

#include<lcd.c>

//*****

// Define as interrupções

//*****

#ORG 0x1000, 0x3000

#int_timer0

void trata_t0 ()

{

    set_adc_channel(0);

    delay_us(3);

    Valor_ADC0 = read_adc();

    LSB = Valor_ADC0&0xFF;

    if(LSB>51 || flag_trigger==1)

    {

        buffer_leitura[i]=LSB;

        i++;

    }

}
```

```
flag_trigger=1;

if(i==199)
{
    i=0;
    flag_trigger=0;
}
}

set_timer0(67 + get_timer0());
}
```

// Converte um número de 8 bits para ASCII e guarda num vetor

```
void byte_ascii(unsigned int A, unsigned int x,y)
{
    int i = 2;
    int j = 0;
    unsigned int ASCII[3];

    ASCII[0] = 0x30;
    ASCII[1] = 0x30;
```

```
ASCII[2] = 0x30;
```

```
while (A>=10)
```

```
{
```

```
    ASCII[i] = A%10;
```

```
    A = A/10;
```

```
    ASCII[i] = ASCII[i]+0x30;
```

```
    --i;
```

```
}
```

```
ASCII[i] = A+0x30;
```

```
lcd_gotoxy(x,y);
```

```
while(j<3)
```

```
{
```

```
    lcd_putc(ASCII[j]);
```

```
    ++j;
```

```
}
```

```
}
```

```
// Converte um número de 16 bits para ASCII e guarda num vetor
```

```
void word_ascii(unsigned long int A, unsigned int x,y)
```

```
{
```

```
    int i = 4;
```

```
int j = 0;

unsigned int ASCII[5];

ASCII[0] = 0x30;

ASCII[1] = 0x30;

ASCII[2] = 0x30;

ASCII[3] = 0x30;

ASCII[4] = 0x30;

while (A>=10)
{
    ASCII[i] = A%10;

    A = A/10;

    ASCII[i] = ASCII[i]+0x30;

    --i;
}

ASCII[i] = A+0x30;

lcd_gotoxy(x,y);

while(j<5)
{
    lcd_putc(ASCII[j]);

    ++j;
}
```



```
}
```

```
/******  
******/
```

```
// PROGRAMA PRINCIPAL
```

```
/******  
******/
```

```
void main(void)
```

```
{
```

```
    char b, c = 0;
```

```
/******  
******/
```

```
// Habilita interrupção de tempo no INT0
```

```
/******  
******/
```

```
    setup_timer_0 (RTCC_INTERNAL | RTCC_DIV_64 | RTCC_8_bit);
```

```
    set_timer0(67);
```

```
    enable_interrupts (global | int_timer0);
```

```
// Funções Preparatórias para ler o canal ADC
```

```
    setup_adc_ports(AN0_TO_AN4);
```

```
setup_adc(ADC_CLOCK_INTERNAL);

// Funções para acessar USB

usb_cdc_init();

usb_init();

// Prepara Pinos de I/O

set_tris_b(0x00);

set_tris_c(0b00111111);

set_tris_e(0xFF);

output_b(0xFF);

output_c(0xE0);

// Inicializa Display e faz Saudação

lcd_init();

lcd_gotoxy(1,1);

lcd_putc("Osciloscopio");

lcd_gotoxy(1,2);

lcd_putc("Didatico");

delay_ms(3000);
```

```
    lcd_putc("\fAN0=000 ");

// Laço Principal

    delay_ms(1000);

    while(1)
    {

        {

            if(i==0 && usb_ok==1)

            {

                for(k=0;k<200;k++)

                {

                    vetor_usb[k]=buffer_leitura[k];

                }

                usb_ok=0;

            }

        }

        //-----

        // Rotina para enviar e receber dados do Datalogger PIC USB

        //-----
```

```
// Este trecho envia o conteúdo da variável Valor_ADC0 para a USB. Lembrar que  
devem ser enviados
```

```
// sempre dois bytes.
```

```
byte_ascii(Valor_ADC0,5,1);
```

```
if(usb_ok==0)
```

```
{
```

```
for(j=0;j<200;j++)
```

```
{
```

```
printf(usb_cdc_putc,"%2X", vetor_usb[j]);
```

```
delay_ms(2);
```

```
}
```

```
usb_ok=1;
```

```
}
```

```
//-----
```

```
// Este trecho recebe dois bytes do Datalogger.
```

```
//-----
```

```
if(usb_cdc_kbhit())
```

```
{
```

```
c = (usb_cdc_getc());
```

```
if(c < 0x40)
{
    c = c - 0x30;
}
else
{
    c = c - 0x37;
}
c = c << 4;
}

if(usb_cdc_kbhit())
{
    b = (usb_cdc_getc());
    if(b < 0x40)
    {
        b = b - 0x30;
    }
    else
    {
        b = b - 0x37;
    }
}
```

// Este trecho envia o byte lido da USB e coloca na porta B.

```
}  
  
c = b | c;  
  
switch (c)  
{  
  
    case 1 : output_low(pin_b5);  
             output_low(pin_b6);  
             output_low(pin_b7);  
             output_high(pin_b4);  
  
    break;  
  
    case 2 : output_low(pin_b4);  
             output_low(pin_b6);  
             output_low(pin_b7);  
             output_high(pin_b5);  
  
    break;  
  
    case 4 : output_low(pin_b4);  
             output_low(pin_b5);  
             output_low(pin_b7);
```

```
        output_high(pin_b6);  
break;  
  
case 10 : output_low(pin_b4);  
        output_low(pin_b5);  
        output_low(pin_b6);  
        output_high(pin_b7);  
  
break;  
  
default: output_low(pin_b5);  
        output_low(pin_b6);  
        output_low(pin_b7);  
        output_high(pin_b4);  
  
}  
  
}  
  
}  
  
}
```

7.8 Construção do software em LabVIEW

