



**Universidade Federal de Campina Grande**

**Centro de Engenharia Elétrica e Informática**

Curso de Graduação em Engenharia Elétrica

MARCIO HERSON DE AGUIAR

**RELATÓRIO DE ATIVIDADES DESENVOLVIDAS NA  
EMPRESA DATASONIC NO PROJETO DE DESENVOLVIMENTO  
DE UM KIT EDUCACIONAL PARA ENSINO/APRENDIZAGEM  
DE CIRCUITOS ELÉTRICOS**

Campina Grande, Paraíba  
Agosto de 2015

MARCIO HERSON DE AGUIAR

RELATÓRIO DE ATIVIDADES DESENVOLVIDAS NA  
EMPRESA DATASONIC NO PROJETO DE DESENVOLVIMENTO  
DE UM KIT EDUCACIONAL PARA ENSINO/APRENDIZAGEM  
DE CIRCUITOS ELÉTRICOS

*Relatório de Estágio Supervisionado submetido  
à Unidade Acadêmica de Engenharia Elétrica  
da Universidade Federal de Campina Grande  
como parte dos requisitos necessários para a  
obtenção do grau de Bacharel em Ciências no  
Domínio da Engenharia Elétrica.*

Área de Concentração: Processamento de Dados

Orientador:

Professor Gutemberg Gonçalves dos Santos Júnior, D. Sc.

Campina Grande, Paraíba  
Agosto de 2015

MARCIO HERSON DE AGUIAR

RELATÓRIO DE ATIVIDADES DESENVOLVIDAS NA  
EMPRESA DATASONIC NO PROJETO DE DESENVOLVIMENTO  
DE UM KIT EDUCACIONAL PARA ENSINO/APRENDIZAGEM  
DE CIRCUITOS ELÉTRICOS

*Relatório de Estágio Supervisionado submetido  
à Unidade Acadêmica de Engenharia Elétrica  
da Universidade Federal de Campina Grande  
como parte dos requisitos necessários para a  
obtenção do grau de Bacharel em Ciências no  
Domínio da Engenharia Elétrica.*

Área de Concentração: Processamento de Dados

Aprovado em        /        /

---

**Professor Avaliador**

Universidade Federal de Campina Grande

Avaliador

---

**Professor Gutemberg Gonçalves dos Santos Júnior, D. Sc.**

Universidade Federal de Campina Grande

Orientador

## AGRADECIMENTOS

Agradeço a Deus por ter proporcionado o alimento espiritual que me deu forças para enfrentar os diversos obstáculos ao longo do período de graduação, como a distância e a saudade de casa.

Aos meus pais, Luzia e Neto, por ter me dado uma educação preparatória para enfrentar, sempre com respeito ao próximo, as adversidades da vida adulta.

Ao Prof. Maurício e ao proprietário da DATASONIC, Marcos Procópio, pela oportunidade do estágio nesta empresa.

Ao Prof. Gutemberg, pelas orientações no desenvolvimento das atividades durante o período de estágio.

À equipe de desenvolvimento do projeto do Kit Educacional, especialmente ao colega Kelson Santana, pela ajuda e suporte na realização das atividades.

Queria agradecer à UFCG e a todos os mestres que participaram da minha formação profissional durante esses anos de graduação.

E a todos os meus familiares, colegas e demais amigos que fizeram parte dessa trajetória e acreditaram no meu sucesso.

*"Mesmo o conhecimento precisa de algo mais para aprender, algo que seja aperitivo  
para a curiosidade."*

(Baltasar Gracián)

## RESUMO

A utilização de ferramentas que ofereçam a experiência prática de laboratório é de suma importância no aprendizado de circuitos elétricos. Entretanto, a maioria destas ferramentas apresenta um alto custo, o que impossibilita a utilização delas em grande parte dos centros de educação técnica. Visando atender a esta necessidade, a DATASONIC propôs o desenvolvimento de um Kit Educacional de baixo custo para ensino e aprendizagem de circuitos elétricos. Desta forma, este relatório de estágio tem por objetivo descrever as atividades realizadas no projeto do sistema de aquisição de dados para ser utilizado no Kit. Essencialmente, o processo de aquisição e análise de dados dos experimentos consiste de três etapas: conversão analógica para digital (A/D) das grandezas elétricas dos experimentos, transmissão dos dados para um computador e processamento e recuperação dos dados em uma interface gráfica. As atividades foram propostas com objetivo de desenvolver um protótipo que realizasse estas funções e que apresentasse um baixo custo.

**Palavras-chave:** Kit Educacional, Circuitos Elétricos, Baixo Custo, Sistema de Aquisição de Dados, Conversão A/D, Transmissão de Dados, Interface Gráfica.

# SUMÁRIO

1	Introdução.....	1
1.1	Kits Educacionais.....	1
1.2	Objetivos.....	2
1.3	Especificações do Projeto.....	3
1.4	Organização do Relatório.....	3
2	Atividades Desenvolvidas.....	4
3	Especificação do Microcontrolador.....	5
4	Sistema de Aquisição de Dados.....	6
4.1	Definição de Aquisição de Dados.....	6
4.2	Cabos de Ligação.....	7
4.3	Circuito de Condicionamento de Sinal.....	8
4.3.1	Projeto do Circuito de Condicionamento.....	9
4.3.2	Simulação do Circuito de Condicionamento.....	11
4.4	Hardware de Aquisição de Dados.....	12
4.5	Sistema de Conversão A/D.....	13
4.5.1	Conversão A/D por Aproximações Sucessivas.....	14
4.5.2	Diagrama de Blocos do Módulo Conversor A/D.....	15
4.5.3	Registradores do Módulo A/D.....	16
4.5.4	Modo de Operação do Conversor A/D.....	17
4.6	Sistema de Comunicação.....	20
4.6.1	Comunicação Serial Assíncrona.....	20
4.6.2	Formato das Mensagens de Dados.....	21
4.6.3	Velocidade de Transmissão de Dados.....	21
4.6.4	Interface UART.....	22
4.6.5	Módulo UART do Microcontrolador.....	23
4.6.6	Registradores do Módulo UART.....	24
4.6.7	Conversor USB/RS-232.....	26
4.7	Programação do Microcontrolador.....	27
5	Projeto da Placa de Circuito Impresso e Montagem.....	30
6	Interface Gráfica.....	32
6.1	Gráfico.....	33
6.2	Painel para Configurar Comunicação.....	33
6.3	Algoritmo para Reconstruir os Sinais.....	33
6.4	Painéis de Controle da Escala de Tensão.....	35
6.5	Painel de Controle da Escala de Tempo.....	35
6.6	Painel de Controle do <i>Trigger</i> .....	35
7	Testes e Resultados.....	36

7.1	Teste do Circuito de Condicionamento.....	36
7.2	Verificação da Reconstrução do Sinal .....	37
7.3	Teste de Conservação da Diferença de Fase .....	39
7.4	Teste da Função Trigger da Interface.....	40
7.5	Teste das Funções Ganho e Posição.....	42
8	Conclusão .....	43
	Bibliografia.....	44
	Apêndice I – Esquema Elétrico do <i>Hardware</i> .....	45
	Apêndice II – Lista de Componentes do <i>Hardware</i> .....	46
	Apêndice III – Biblioteca de Funções em C utilizadas na Programação do PIC24 .....	47
	Apêndice IV – Código Fonte da Programação do Microcontrolador .....	49

# 1 INTRODUÇÃO

O ensino/aprendizagem de circuitos elétricos requer, além do embasamento teórico, uma análise experimental dos sistemas relacionados com a ementa deste curso para uma melhor consolidação dos conhecimentos.

Equipamentos como gerador de funções, osciloscópio e ferramentas de cálculo numérico como o MATLAB, proporcionam boas experiências para o estudante de circuitos elétricos. Porém, estes equipamentos e ferramentas apresentam custo elevado e, muitas vezes, são complexos para quem está apenas iniciando em tal estudo. Logo, é notável a necessidade de um Kit Educacional de baixo custo que facilite o ensino/aprendizagem e proporcione boas experiências a quem se aventura nessa área do conhecimento.

Visando atender a necessidade exposta anteriormente, o desenvolvimento de um Kit Educacional de baixo custo para o ensino/aprendizagem de circuitos elétricos foi proposto pela empresa DATASONIC. O projeto apresenta duas vertentes, sendo uma relacionada com o *hardware* que permite a interação dos alunos com as grandezas físicas estudadas e a outra, com o *software* (sistema) que permite ao professor a aplicação de questionamentos para facilitar o ensino.

Sendo assim, neste relatório são descritas as atividades desempenhadas durante o estágio supervisionado realizado no projeto de desenvolvimento do *hardware* para empresa DATASONIC, no período de Abril/2015 a Junho/2015.

## 1.1 KITS EDUCACIONAIS

São plataformas de educação desenvolvidas para criar experiências de aprendizagem aos estudantes de forma mais eficiente, potencializando o talento e as ideias inovadoras.

A concepção destas ferramentas de ensino é dada com o intuito de realizar o treinamento prático dos alunos de uma disciplina ou área de conhecimento a qual estejam direcionadas, de modo a realizar a ligação entre teoria, experimentação e, em alguns casos, simulação.

Na área de ciências e engenharia, há um grande número destes kits projetados tanto para usos específicos como para serem utilizados de maneira adaptativa, atendendo a uma ampla quantidade de áreas de estudos.

Neste segundo contexto, entram os Kits Educacionais como o NI ELVIS (*Educational Laboratory Virtual Instrumentation Suite*) da *National Instruments* (Figura 1.1a) e o AReS (*All-in One Real time Instruments & Educational System*) da empresa Coreana *Chungpa* (Figura 1.1b). Diversos módulos de experimentos podem ser conectados a base do Kit Educacional. Na Figura 1.2 temos três placas de experimentos do AReS.



Figura 1.1. Plataformas Educacionais para Engenharia.

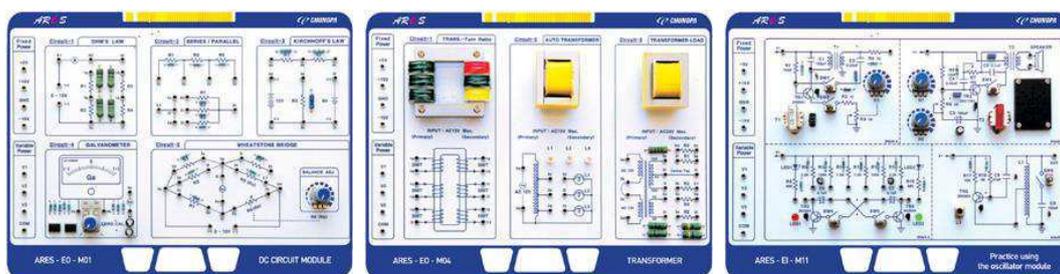


Figura 1.2. Módulos de Experimento do AReS.

## 1.2 OBJETIVOS

As atividades foram realizadas com objetivo de desenvolver um sistema de aquisição de dados e comunicação com *Personal Computer* (PC), juntamente com uma interface gráfica. Com estes dois sistemas (*hardware* e *software*) será possível adquirir e visualizar as grandezas elétricas dos experimentos.

### 1.3 ESPECIFICAÇÕES DO PROJETO

Como as atividades foram direcionadas a desenvolver um protótipo de parte de um produto que vai ser comercializado, especificações para o *hardware* foram definidas com base na utilização pretendida do Kit Educacional. As especificações foram as seguintes:

- Comunicação USB com o computador;
- Dois canais de entrada analógica;
- Faixa de tensão dos sinais: -10 a +10 V;
- Faixa de frequência dos sinais: 1 a 15 kHz.

### 1.4 ORGANIZAÇÃO DO RELATÓRIO

No desenvolvimento do relatório é feito, inicialmente, uma relação das atividades desenvolvidas. Na sequência, elas são descritas começando pela determinação do microcontrolador, seguida do projeto do sistema de aquisição de dados, onde é mostrado o desenvolvimento de cada subcircuito, a implementação da interface gráfica e, por fim, o desenho do *layout* da placa de circuito impresso.

Os subcircuitos são abordados na seguinte sequência: condicionador de sinal, conversor analógico/digital e interface de comunicação serial. Como as atividades constam fundamentalmente de atividades de projeto, o embasamento teórico e a simulação dos sistemas são realizados na medida em que é apresentado o projeto, procurando fundamentá-lo. O relatório é finalizado com a apresentação de testes e o encaminhamento das conclusões.

## 2 ATIVIDADES DESENVOLVIDAS

As atividades desenvolvidas no estágio foram propostas de forma a realizar o projeto e a implementação de um sistema que adquirisse dados e que pudesse se comunicar com o computador.

Entretanto, para testar o *hardware* que seria desenvolvido, havia a necessidade de uma interface no PC. Logo, foi proposto também o desenvolvimento de uma interface que configurasse a comunicação e permitisse a visualização dos sinais adquiridos.

Dessa forma, foram realizadas basicamente as seguintes atividades gerais:

- Escolha do microcontrolador;
- Projeto e implementação do sistema de aquisição de dados;
- Desenvolvimento da interface gráfica;
- Projeto da placa de circuito impresso e montagem.

Para realizar as atividades foram utilizados os seguintes recursos do laboratório de eletrônica (Figura 2.1) da empresa DATASONIC: fonte de tensão ajustável, osciloscópio, gerador de sinais, multímetro digital, *protoboard*, programador de PIC, estação de solda e os softwares OrCAD, Proteus e CCS Compiler.



Figura 2.1. Laboratório de eletrônica da DATASONIC.

### 3 ESPECIFICAÇÃO DO MICROCONTROLADOR

A ideia inicial do projeto do Kit Educacional foi de ter um sistema que fizesse aquisição de dados, realizasse o processamento destes, e então os enviasse para um PC onde seriam visualizados em uma interface gráfica. Para realizar estas funções era necessário embarcar uma "inteligência" ao sistema. Esta, por sua vez, poderia ser obtida por meio de um microcontrolador.

Considerando as funções descritas acima, necessitava-se de um microcontrolador que além de possuir conversor analógico/digital (A/D) e comunicação serial, tivesse uma boa taxa de amostragem para a conversão dos sinais, considerando a faixa de frequência dos sinais de entrada de 1 a 15 kHz.

Como os microcontroladores da linha PIC mais comuns (PIC16, PIC18) não possuem uma taxa de conversão suficiente para realizar amostragem de sinais na faixa de frequência requerida, foi escolhido o microcontrolador PIC24FJ16GA002 da linha PIC24 com arquitetura de 16 bits, 28 pinos e taxa de amostragem de 500 ksp/s (kilo sample-per-second). Além dos critérios citados acima, foram consideradas também as seguintes características para a escolha deste microcontrolador:

- Existe em encapsulamento DIP (facilitação das montagens);
- Permite simulação no *software* PROTEUS-ISIS;
- Permite desenvolvimento de *firmware* no CCS Compiler;
- Possui documentação acessível;
- Gravador/programador facilmente disponível no mercado.

O aspecto físico do microcontrolador e o equipamento utilizado no estágio para programá-lo, são mostrados nas Figuras 3.1a e 3.1b, respectivamente.

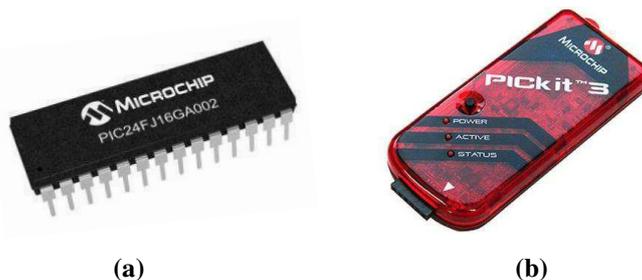


Figura 3.1. a) Microcontrolador PIC24 b) Gravador PICKit 3.

## 4 SISTEMA DE AQUISIÇÃO DE DADOS

No decorrer deste capítulo é mostrado o projeto do sistema de aquisição de dados que compõe o *hardware* do Kit. Primeiramente é apresentada a definição de aquisição de dados, depois considerações sobre os cabos de ligação, em seguida, o projeto do condicionador de sinal, a configuração do conversor A/D e a implementação da comunicação serial. Nos Apêndices I e II, tem-se, respectivamente, o esquema elétrico do *hardware* e a lista dos componentes utilizados.

### 4.1 DEFINIÇÃO DE AQUISIÇÃO DE DADOS

Aquisição de dados é o processo pelo o qual um fenômeno físico do mundo real é transformado em um sinal elétrico que é medido e convertido num formato digital para processamento, análise e armazenamento por um computador.

Numa definição mais estendida, um sistema de aquisição de dados (DAQ - *Data Acquisition*) não somente adquirem dados, mas também atua no sistema, ou seja, inclui aspectos de controle no sistema total (PARK e MACKAY, 2003).

Os elementos básicos de um sistema de aquisição de dados, como mostrado no diagrama funcional da Figura 4.1, são os seguintes:

- Sensores e transdutores;
- Cabos de ligação;
- Condicionadores de sinal;
- *Hardware* de aquisição de dados;
- PC (sistema operacional)
- *Software* de aquisição de dados.

Para a implementação do *hardware* básico do kit educacional, nem todos esses elementos foram utilizados, pois ele consiste inicialmente de uma placa que recebe sinais já na forma elétrica. Portanto, não foi preciso utilizar sensores e transdutores.

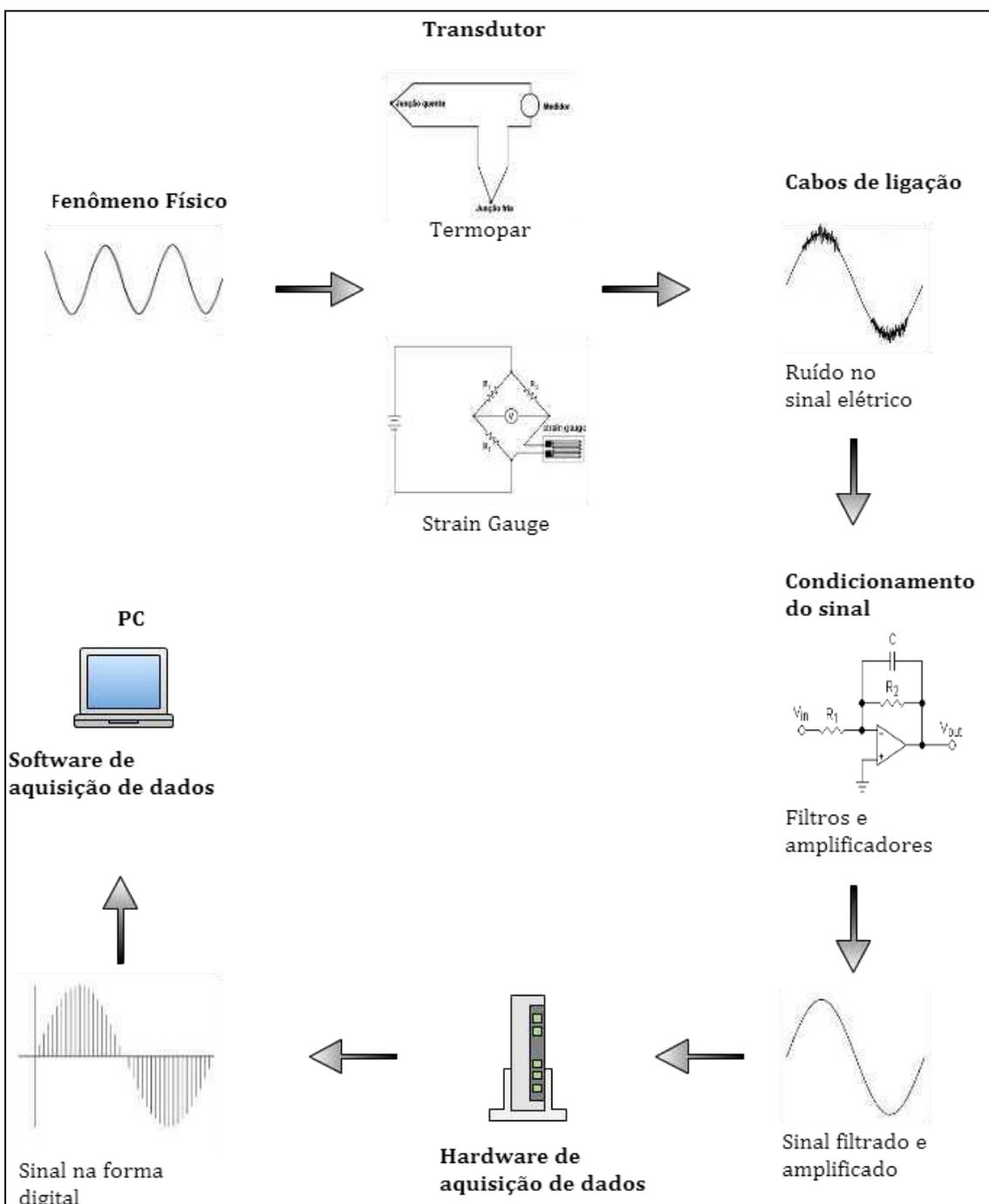


Figura 4.1. Diagrama funcional de um sistema de aquisição de dados.

## 4.2 CABOS DE LIGAÇÃO

Cabos de ligação representam a conexão física dos transdutores e sensores com *hardware* de condicionamento de sinal e/ou o *hardware* de aquisição de dados. Quando estes dispositivos estão localizados distante do PC, os cabos de ligação providenciam

uma ligação física entre eles e o PC. Se essa ligação física é uma interface de comunicação RS-232 ou RS-485, este componente é frequentemente referido como cabos de comunicação.

Dentre todos os elementos, estes são os mais susceptíveis a efeitos de ruídos externos, especialmente em ambientes industriais. Este componente passivo de um sistema de aquisição de dados deve ser visto como um item de extrema importância e necessita-se tomar cuidados com ele, caso contrário, tornará o sistema sem acurácia, precisão e exatidão.

No Kit desenvolvido é utilizado somente um cabo USB para interligá-lo ao PC. Este cabo deve ter um filtro supressor de interferência eletromagnética para diminuir os efeitos das fontes de interferência conduzida. Um modelo de cabo USB com supressor pode ser visto na Figura 4.2.



Figura 4.2. Cabo USB com supressor de EMI.

### 4.3 CIRCUITO DE CONDICIONAMENTO DE SINAL

Sinais elétricos gerados por transdutores frequentemente precisam ser convertidos em uma forma aceitável para o DAQ, particularmente conversores A/D que converte os sinais de dados para uma forma digital requerida. Condicionamento de sinal é um termo geralmente utilizado para descrever o pré-processamento requerido para converter os sinais elétricos recebidos de transdutores e sensores em sinais que o DAQ possa aceitar. As principais tarefas realizadas no condicionamento do sinal são:

- Amplificação;
- Filtragem;
- Linearização;

- Isolação;
- Excitação.

Como no projeto os sinais de entrada não têm um padrão específico, pois dependem do tipo de experimento realizado com o Kit, somente a tarefa de amplificação listada acima é utilizada. Na verdade, o que se deve fazer é uma atenuação dos sinais de entrada, já que a faixa de tensão especificada é de -10 a +10 V e a aquisição dos sinais é feita pelo conversor A/D do microcontrolador que apresenta uma faixa permitida de tensão de entrada compreendida entre 0 e 3,3 V. Portanto, houve a necessidade de utilizar um circuito de condicionamento de sinal para manter o sinal dentro desta faixa e proteger o microcontrolador.

#### 4.3.1 PROJETO DO CIRCUITO DE CONDICIONAMENTO

Nas Figuras 4.3a e 4.3b, são ilustrados, respectivamente, um exemplo de um sinal com os limites de tensão suportados pelo sistema e a forma como ele deve ser apresentado na entrada do microcontrolador.



Figura 4.3. Valores críticos de tensão de entrada: a) no sistema b) no conversor A/D.

Tendo em vista os dados mencionados e realizando a devida comparação de escalas entre as duas figuras anteriores, percebe-se que a conversão do sinal de entrada do sistema para outro sinal que mantenha o seu comportamento dinâmico e que possa ser aplicado ao microcontrolador é feita por um circuito que implemente a seguinte função polinomial de 1º ordem:

$$f(x) = 0,165x + 1,65. \quad (4.1)$$

O circuito utilizado para realizar esta conversão foi feito por meio de amplificadores operacionais configurados como inversores e somadores. Na Figura 4.4 tem-se o esquema completo do circuito de condicionamento do Kit, que consiste no *hardware* que implementa a equação (4.1) mais um circuito de proteção. Para efeito de compensação, a função inversa é realizada no *software* de aquisição de dados de forma digital a fim de restaurar os valores originais.

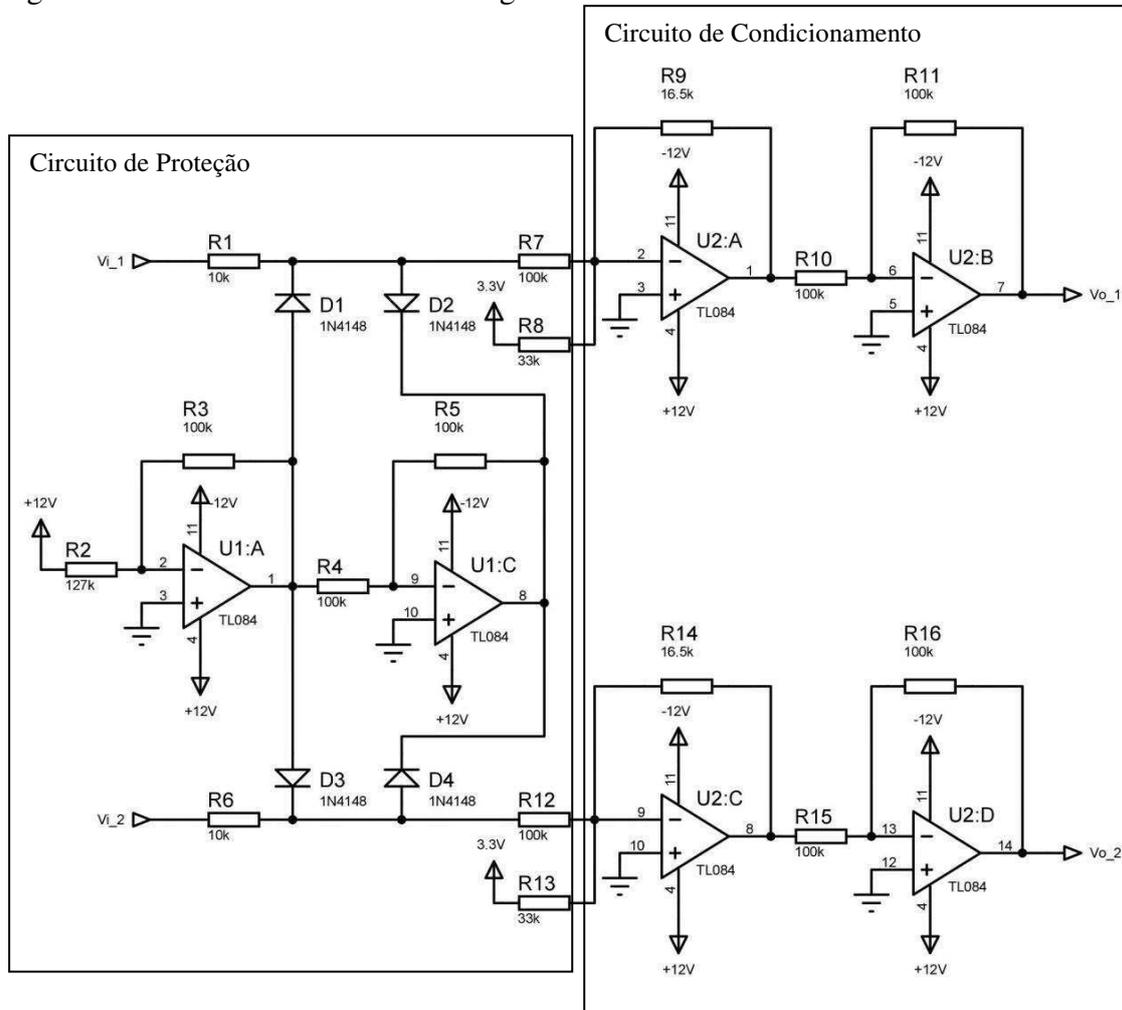


Figura 4.4. Esquema completo do circuito de condicionamento de sinal.

Nota-se que se o sinal de entrada apresentar amplitude maior do que 10 V a saída do circuito de condicionamento apresentará tensões fora da faixa permitida pelo microcontrolador. Uma vez que o uso do equipamento será didático, não é difícil prever que alunos desprevenidos irão aplicar tensões maiores do que as devidas.

Como mencionado, os limites de máximo e mínimo são de respectivamente 10 V e -10 V, portanto o circuito de proteção realiza a função de limitar a entrada nos valores supracitados. Para realizar essa proteção, foram utilizados circuitos limitadores de tensão ou ceifadores de tensão que ceifam segmentos de sinal aplicado em sua entrada,

sem distorcer o restante da forma de onda no tempo. Logo, desejamos um circuito que corte o sinal de entrada nas partes em que ele é maior do que 10 V e menor que -10V. Nas Figuras 4.5a e 4.5b, são ilustrados a topologia deste circuito e o seu comportamento para uma entrada fora dos limites de tensão.

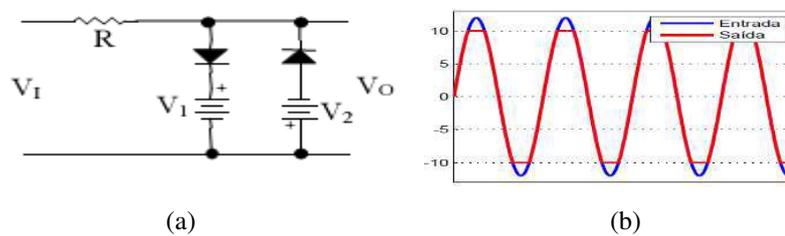


Figura 4.5. Circuito ceifador de tensão: a) topologia b) comportamento.

#### 4.3.2 SIMULAÇÃO DO CIRCUITO DE CONDICIONAMENTO

Antes de realizar a montagem dos circuitos reais, foi feito primeiramente uma simulação para verificar o funcionamento e os valores esperados. Foi utilizado no estágio o *software* de simulação e projeto de circuitos eletrônicos OrCAD para as simulações dos circuitos analógicos, devido ao fato dele apresentar uma modelagem de componentes comerciais bem próxima do real.

A seguir é apresentado o resultado de três simulações diferentes, onde foram utilizadas amplitudes diferentes no sinal de entrada e observado a saída. Na primeira simulação o sinal de entrada apresenta uma amplitude de 4 V. Conseqüentemente, dado que o sinal se localiza na faixa de tensão permitida, espera-se que o circuito de proteção não atue e que o sinal de saída (verde) esteja dentro da faixa de 0 a 3,3 V com um nível médio de 1,65 V. Na Figura 4.6, podemos observar estes resultados. O sinal na saída do circuito de proteção (vermelho) é igual ao da entrada (roxo), ou seja, eles estão sobrepostos no gráfico.

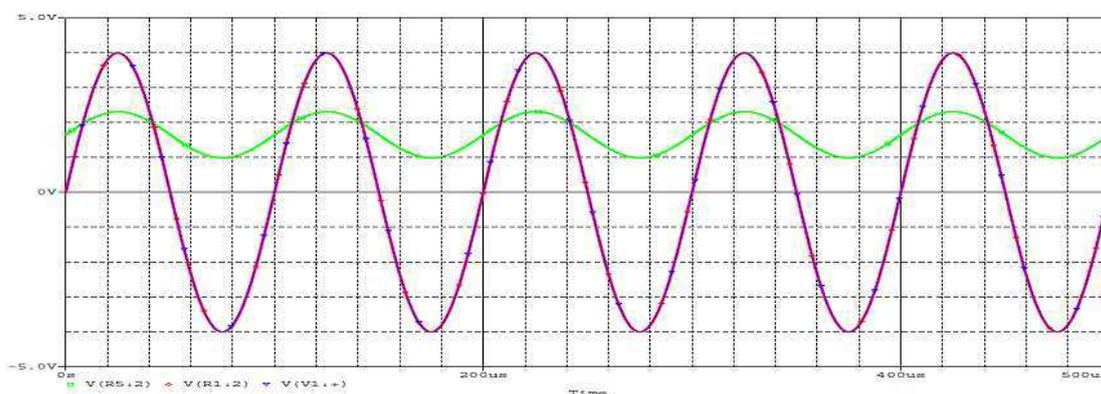


Figura 4.6. Simulação do circuito de condicionamento para entrada com 4 V de amplitude.

Na segunda simulação foi utilizado um sinal de entrada com amplitude de 10 V que é o valor crítico permitido. Portanto, espera-se que o sinal de saída esteja dentro da faixa de 0 a 3,3 V, com um nível médio de 1,65 V, com seu valor mínimo próximo de 0 e o máximo próximo de 3,3 V. Podemos ver o resultado na figura abaixo.

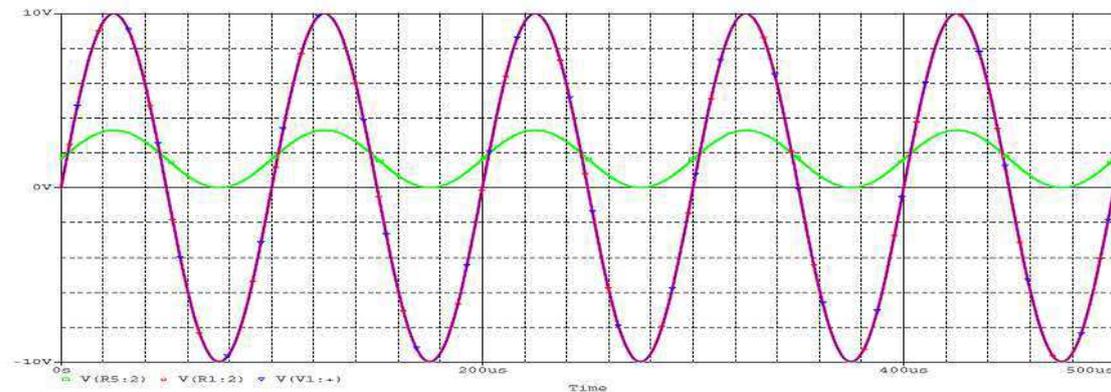


Figura 4.7. Simulação do circuito de condicionamento para entrada com 10 V de amplitude.

Na última simulação foi utilizado um sinal bem acima do permitido com uma amplitude de 15 V. Espera-se que o circuito de proteção atue neste caso. E é o que observamos na Figura 4.8, pois o sinal na saída do circuito de proteção está ceifado. Notamos isto também na saída final do circuito de condicionamento que apresenta seus valores de mínimo e máximo respectivamente em 0 e 3,3 V, de forma a proteger o microcontrolador.

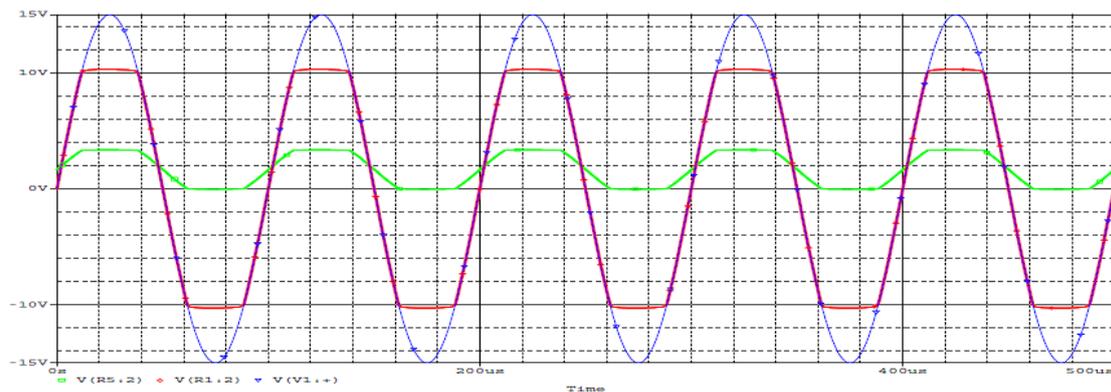


Figura 4.8. Simulação do circuito de condicionamento para entrada com 15 V de amplitude.

## 4.4 HARDWARE DE AQUISIÇÃO DE DADOS

O *hardware* DAQ atua como a interface entre o computador e sinais do mundo exterior. Ele funciona basicamente como um dispositivo que digitaliza sinais analógicos de entrada de forma que um computador possa interpretá-los.

Pode ser definido como um sistema de aquisição de dados e controle, o componente que permita qualquer dos seguintes pontos (PARK e MACKAY, 2003):

- A entrada de sinais de dados analógicos medidos de um sistema, processando-o e convertendo-o para o formato digital utilizando ADCs (*Analog-to-Digital Converter*). Os dados são depois transferidos para um computador para ser mostrado, armazenado e analisado;
- A entrada de sinais digitais, que contem informações de um sistema ou processo;
- O processamento de sinais digitais do computador, sendo em seguida convertidos para o formato analógico utilizando DACs (*Digital-to-Analog Converter*). O sinal analógico é utilizado para controlar um sistema ou um processo;
- A saída de sinais digitais de controle.

Neste protótipo inicial do Kit, há somente a entrada de dados analógicos, sem saída de sinais de controle, fazendo que ele se enquadre nas funções do primeiro item mostrado na definição acima.

O PIC24 apresenta um módulo ADC e uma interface UART (*Universal Asynchronous Receiver/Transmitter*) para realizar a conversão e a comunicação, respectivamente. Nas Seções 4.5 e 4.6, é mostrada a implementação destes dois subsistemas com o microcontrolador.

## 4.5 SISTEMA DE CONVERSÃO A/D

O ADC do microcontrolador apresenta principalmente as seguintes características:

- Conversão por aproximações sucessivas;
- Velocidade de conversão de até 500 ksps;
- Resolução de 10 bits;
- Até 13 pinos de entrada analógica;
- Entradas de referência externa;

- Modo de varredura automática dos canais;
- *Buffer* de 16 posições para guardar os resultados das conversões.

#### 4.5.1 CONVERSÃO A/D POR APROXIMAÇÕES SUCESSIVAS

O módulo A/D do microcontrolador é baseado no método de conversão A/D por aproximações sucessivas. O diagrama de blocos do funcionamento deste método é mostrado na Figura 4.9. Ele consiste de três blocos principais: registrador de aproximações sucessivas (SAR), conversor digital/analógico (DAC) e comparador.

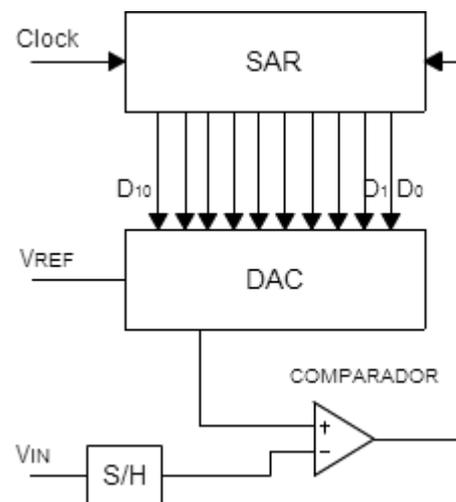


Figura 4.9. Diagrama de blocos do método de conversão A/D por aproximações sucessivas.

O funcionamento é dado da seguinte forma: o registrador SAR é um contador que é incrementado por uma entrada de *clock*, seu valor binário é disponibilizado por linhas de saída que estão interligadas à entrada de um DAC. Na medida em que ocorre a contagem, o DAC converte o valor digital do registrador para um valor analógico que é disponibilizado na entrada não inversora do comparador, que já dispõe na sua entrada inversora a amostra do sinal analógico que se deseja converter. O registrador inicia a contagem de zero e vai incrementando até que tenha um valor imediatamente maior do que a amostra do sinal. Neste momento, o comparador emite um sinal de *flag* para o registrador, informando que terminou a conversão. O valor de parada da contagem representa a amostra do sinal de entrada num formato digital.

O tempo de conversão depende do nível de tensão da amostra, pois para um valor alto é necessário mais ciclos de *clock*. O número máximo de *clocks* depende da resolução do conversor. No caso do conversor do PIC24 que apresenta uma resolução

de 10 bits, significa que o conversor pode necessitar de uma quantidade de *clocks* de até 1024 para realizar uma conversão, segundo (4.2).

$$n^{\circ} \text{ máx de clocks} = 2^{n^{\circ} \text{ de bits}}. \quad (4.2)$$

Este valor também pode ser utilizado para calcular a resolução do conversor A/D em *Volts*. Utilizando uma tensão de referência ( $V_{REF}$ ) de 3,3 V para o conversor do PIC, ele apresentará a seguinte resolução em Volts.

$$Res_{Volts} = \frac{V_{REF}}{2^{n^{\circ} \text{ de bits}}}. \quad (4.3)$$

$$Res_{Volts} = \frac{3,3}{1024} = 3,22 \text{ mV}. \quad (4.4)$$

#### 4.5.2 DIAGRAMA DE BLOCOS DO MÓDULO CONVERSOR A/D

Na Figura 4.10, tem-se o diagrama de blocos do conversor A/D do PIC24. Pode-se ver claramente o bloco comparador, DAC e SAR da conversão por aproximações sucessivas, o *buffer* de 16 posições e as 13 entradas disponíveis. Como existe somente um conversor A/D, as entradas são multiplexadas pelos blocos MUX A e MUX B para serem utilizadas pelo mesmo conversor. Logo, percebe-se que quanto maior o número de entradas utilizadas, menor será a frequência de amostragem dos sinais, pois enquanto uma entrada é utilizada outra fica na espera.

Existe também o bloco amplificador S/H (*sample and hold*) que realiza a função de segurar o valor da amostra enquanto é realizada a conversão. O bloco "*Control Logic*" faz a decodificação dos controles e as configurações setadas nos registradores do microcontrolador.



- AD1PCFG(L) e AD1PCFGH: Registradores de configuração das portas A/D;
- AD1CSSL e AD1CSSH: Registradores de seleção das entradas para varredura.

Os registradores AD1CON1, AD1CON2 e AD1CON3 controlam a operação geral do módulo A/D. Isto inclui habilitar o módulo, configurar o clock de conversão e fontes de tensão de referência, selecionar os *triggers* de amostragem e conversão, e o controle manual das sequências de amostragem/conversão.

O registrador AD1CHS seleciona os canais de entrada para serem conectados ao amplificador S/H. O registrador AD1PCFG configura os pinos de I/O como entrada analógica ou I/O digital. O registrador AD1CSSL seleciona os canais que serão incluídos para a sequência de varredura.

#### 4.5.4 MODO DE OPERAÇÃO DO CONVERSOR A/D

Utilizando os registradores pode-se configurar o modo de operação do módulo de acordo com as necessidades do projeto. A seguir são listadas as características particulares para a operação do conversor A/D no Kit:

- Dois canais de entrada analógica (AN0 e AN1);
- Faixa de tensão de entrada de 0 a 3,3 V;
- Dados digitais de saída como inteiros sem sinal;
- A amostragem do canal seguinte começa automaticamente após o término da amostragem do anterior;
- Frequência de amostragem superior a 225 kHz.

Partindo dessas necessidades, configuram-se os registradores de forma a atendê-las com o valor mais próximo possível. A configuração do módulo A/D é realizada pela função `iniciar_ADC()` na programação do microcontrolador. Na Figura 4.11, mostra-se por meio de um fluxograma como é feita a inicialização do conversor.

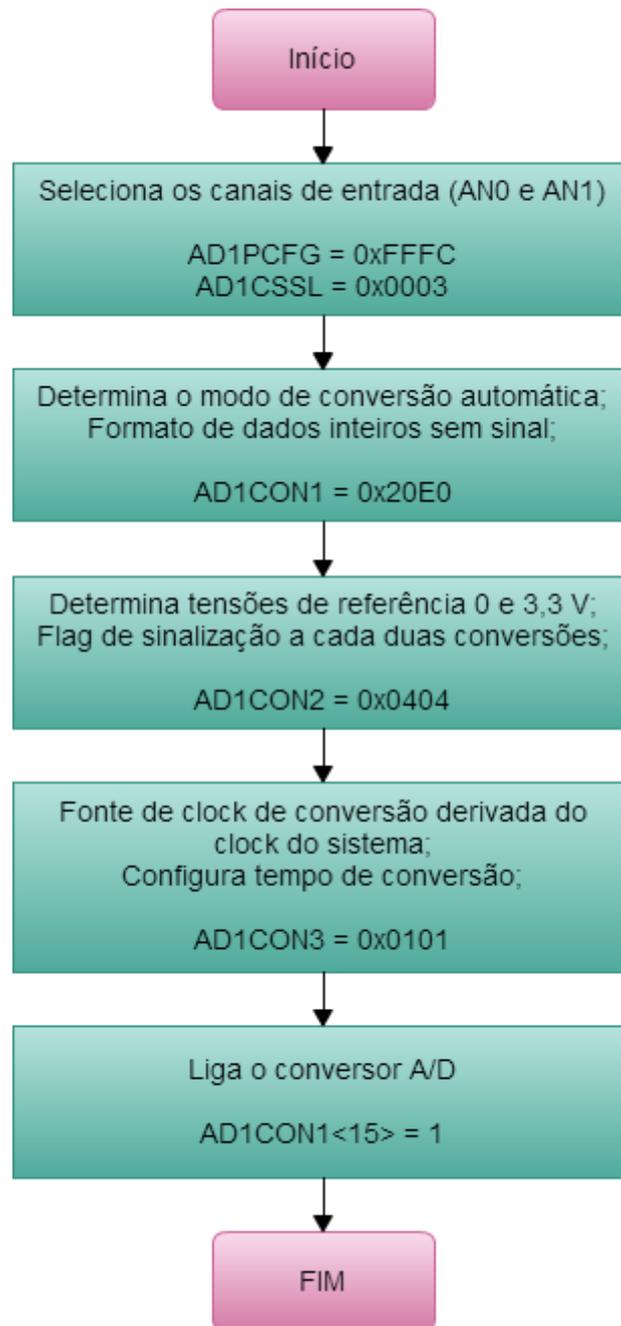


Figura 4.11. Fluxograma da função de inicialização do módulo ADC do PIC24.

O conversor foi configurado para operar com o menor tempo de conversão possível e que mantenha a confiabilidade da conversão. A seguir, é apresentado o cálculo do tempo de conversão utilizado.

O período do clock de conversão  $T_{AD}$  é dado por:

$$T_{AD} = T_{CY} \cdot (ADCS + 1), \quad (4.5)$$

onde  $T_{CY}$  é o período de um ciclo de instrução do processador e é dado por (4.6).

$$T_{CY} = 2 \cdot T_{OSC}, \quad (4.6)$$

onde  $T_{OSC}$  é o período da fonte de clock do processador.

Como ADCS (AD1CON3<7:0>) foi configurado igual a 1 e substituindo (4.5) em (4.6), temos que:

$$T_{AD} = 2 \cdot T_{OSC} \cdot (1 + 1) = 4 \cdot T_{OSC}. \quad (4.7)$$

Para realizar uma conversão completa são necessários 12  $T_{AD}$  mais o tempo para estabilizar o amplificador S/H, que foi configurado para  $1 \cdot T_{AD}$  por meio de SAMC (AD1CON3<12:8>). Logo, o período de conversão é

$$T_{CONV} = 13 \cdot T_{AD} = 52 \cdot T_{OSC}. \quad (4.8)$$

Como são dois canais de entrada, o período de amostragem é então:

$$T_{AMOST} = 2 \cdot T_{CONV} = 104 \cdot T_{OSC} = \frac{104}{F_{OSC}}. \quad (4.9)$$

A frequência do oscilador  $F_{OSC}$  é 32 MHz, logo o período de amostragem é igual 3,25  $\mu$ s. Entretanto, este ainda não é o período de amostragem correto, pois entre varreduras consecutivas do conversor ocorrem instruções de máquinas que aumentam um pouco este valor.

Para fazer a leitura do conversor A/D utiliza-se a função `ler_conversor_AD()`. Esta função inicializa a varredura dos dois canais, isto é, a conversão do sinal da entrada AN0. Depois do término desta, é iniciada automaticamente a conversão do sinal de AN1. O programa fica parado até o término da segunda conversão quando é gerado um *flag* (AD1IF) informando que a varredura foi completada, neste instante o programa sai da função e as amostras digitais podem ser lidas através da leitura dos registradores ADC1BUF0 e ADC1BUF1.

O fluxograma desta função é mostrado na Figura 4.12. A biblioteca que contém as funções em C para a conversão de sinais analógicos está disponível no Apêndice III.

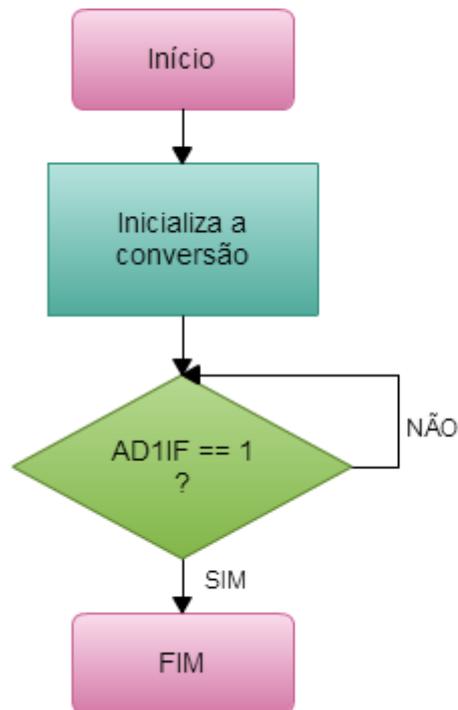


Figura 4.12. Fluxograma da função de leitura dos canais ADC do PIC24.

## 4.6 SISTEMA DE COMUNICAÇÃO

Além de converter os sinais analógicos para um formato digital, o DAQ deve transferir os dados para um computador. Como dito anteriormente, o PIC24 utilizado apresenta um módulo UART que é uma interface de comunicação serial assíncrona. Assim, a função de comunicação do *hardware* de aquisição com o PC é fundamentalmente implementada por este sistema.

### 4.6.1 COMUNICAÇÃO SERIAL ASSÍNCRONA

Em interfaces de comunicação assíncrona não há linhas de sincronização (*clock*), tipicamente apenas duas linhas de dados são utilizadas: TX e RX para entrada e saída, respectivamente (JASIO, 2007). Para realizar a sincronização entre o transmissor e o receptor, cada caractere enviado é precedido por um bit de inicialização (*start bit*) com valor lógico "0" e sucedida por um bit de parada (*stop bit*) com valor lógico "1". Na comunicação assíncrona, o intervalo de tempo entre duas palavras enviadas sucessivamente é variável, por isso a qualificação assíncrona, durante este intervalo a linha de comunicação fica no estado lógico "1".

#### 4.6.2 FORMATO DAS MENSAGENS DE DADOS

Para que uma mensagem enviada por um dispositivo transmissor seja entendida corretamente, ela deve ser apresentada de acordo com algumas regras pré-determinadas. Numa comunicação, os dados são usualmente organizados em um formato particular, junto com informações adicionais, para que a mensagem seja efetivamente enviada, recebida e entendida pelo receptor.

No caso da comunicação serial assíncrona, as informações adicionais são o bit de paridade e os bits *start* e *stop*, mencionados anteriormente e que são mostrados na Figura 4.13. Esta figura apresenta o formato típico de uma mensagem de dados de uma comunicação serial assíncrona. Os bits entre o *start* e o de paridade são a informação propriamente dita.

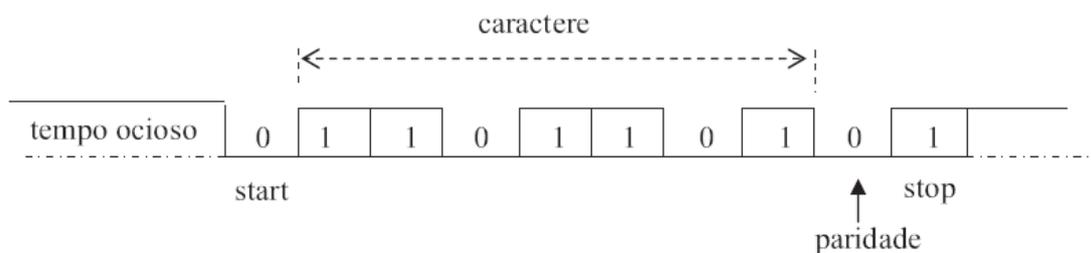


Figura 4.13. Formato típico de uma mensagem de dados serial assíncrona.

#### 4.6.3 VELOCIDADE DE TRANSMISSÃO DE DADOS

Taxas de transferência de dados são usualmente medidas em bits por segundo ou bps. Isto é uma indicação da quantidade de dados úteis que tem sido transmitido para o receptor. Entende-se por dados úteis os bits relacionados com a informação, por exemplo, na comunicação serial assíncrona (Figura 4.13), os dados úteis correspondem aos sete bits que estão entre os bits de *start* e o de paridade, ou seja, esses dois bits mais o de *stop* não são considerados dados úteis.

O *baud rate* pode ser considerado a taxa física em que bits de dados podem ser transferidos e recebidos corretamente em uma interface de comunicação. O *baud rate* é usualmente utilizado em valores padrões como: 1200, 2400, 4800, 19200, 38400, 57600, 115200 e 230400 *baud*. Taxa de transferência de dados e *baud rate* são termos utilizados normalmente de forma intercambiável, mas eles notadamente não são iguais. No exemplo a seguir, pode-se ver a diferença.

Exemplo: Observando a Figura 4.13, se cada bit ocupar um período de  $1\text{ ms}$ , o baud rate poderá ser calculado como:

$$\frac{1\text{ bit}}{1\text{ ms}} = 1000\text{ baud.} \quad (4.10)$$

Já a taxa de transferência, que envolve apenas os dados úteis, é calculada da seguinte forma:

$$\frac{7\text{ bits de dados}}{10\text{ ms}} = 700\text{ bits por segundo (bps).} \quad (4.11)$$

Portanto, é fácil observar que a taxa de transferência de dados do exemplo supracitado é 30% menor que seu correspondente valor de *baud rate*.

#### 4.6.4 INTERFACE UART

O principal componente de uma porta serial assíncrona é o módulo UART. Este componente realiza a conversão serial/paralelo dos dados recebidos de um dispositivo periférico e a conversão paralelo/serial dos dados recebidos da CPU de um PC ou microcontrolador (TEXAS INSTRUMENTS, 2015).

A CPU pode ler o status completo da UART a qualquer tempo durante o funcionamento operacional. Informações de status incluem o tipo e condições das operações de transferência sendo realizadas, assim como situações de erro que ocorram na comunicação. O módulo adiciona o *start* bit, o *stop* bit e o bit de paridade para o dado serial transmitido e detecta estes bits do dado serial recebido.

"A interface UART é talvez a interface mais antiga usada no ambiente de controle embarcado" (JASIO, 2007, tradução nossa). Hoje em dia encontrar uma porta serial assíncrona em novos computadores, especialmente em *notebooks*, tem se tornado um desafio.

Mesmo com a queda de sua popularidade e as características e desempenho evidentemente melhores da interface USB, as interfaces de comunicação serial assíncrona ainda estão resistindo no mundo de aplicações embarcadas devido à sua grande simplicidade e baixo custo de implementação.

#### 4.6.5 MÓDULO UART DO MICROCONTROLADOR

A UART do PIC24 é um canal de comunicação assíncrona *full-duplex* que se comunica com dispositivos periféricos e PCs usando protocolos como RS-232, RS-485, LIN e IrDA®. *Full-duplex* quer dizer que a transmissão e recepção dos dados podem ser realizadas simultaneamente. As principais características deste módulo UART são:

- Transmissão *full-duplex* de dados de 8 ou 9 bits através dos pinos U1T1 e U1R1;
- Um ou dois bits de parada;
- *Hardware* de controle de fluxo;
- Gerador de *baud rate* integrado com *prescaler* de 16 bits;
- Taxa de *baud rate* de 15 bps até 1 Mbps;
- *Buffer* de transmissão de quatro posições, FIFO (*First-In-First-Out*);
- *Buffer* de recepção de quatro posições, FIFO (*First-In-First-Out*);
- Interrupções de transmissão e recepção;
- Codificador e decodificador IrDA.

Um diagrama de bloco simplificado da UART é mostrado na Figura 4.14. Na nossa aplicação não são utilizados os blocos IrDA e *Hardware Flow Control*. O *Baud Rate Generator* é o *hardware* que implementa o *clock* para conversão serial/paralela. Os blocos UART1 *Receiver* e UART1 *Transmitter* são os *hardwares* que realizam a recepção e transmissão de dados, respectivamente.

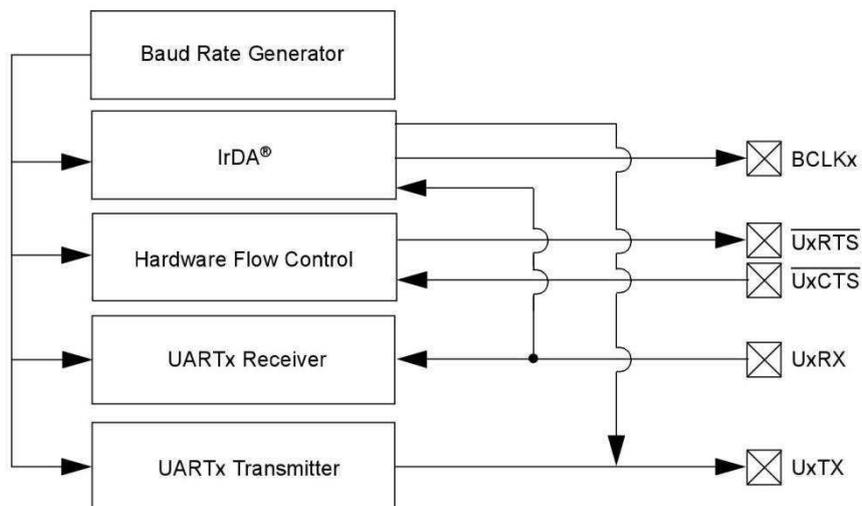


Figura 4.14. Diagrama de blocos simplificado do módulo UART do PIC24.

#### 4.6.6 REGISTRADORES DO MÓDULO UART

Os registradores utilizados para operar o módulo UART são os seguintes:

- U1MODE: registrador de configuração do modo de operação;
- U1STA: registrador de controle e status;
- U1RXREG: registrador de recepção;
- U1TXREG: registrador de transmissão;
- U1BRG: registrador para o gerador de baud rate;
- RPOR0 e RPINR18: seleção dos pinos de transmissão e recepção.

De acordo com o funcionamento desejado da comunicação do Kit, configuram-se estes registradores de modo a obter as seguintes características para a comunicação: *Baud rate* de 230400, transmissão e recepção pelos pinos RP1 e RP2, respectivamente, oito bits de dados, sem paridade e um *stop* bit. Na Figura 4.15, tem-se o fluxograma da função `iniciar_UART()` que configura o modo de operação da UART.

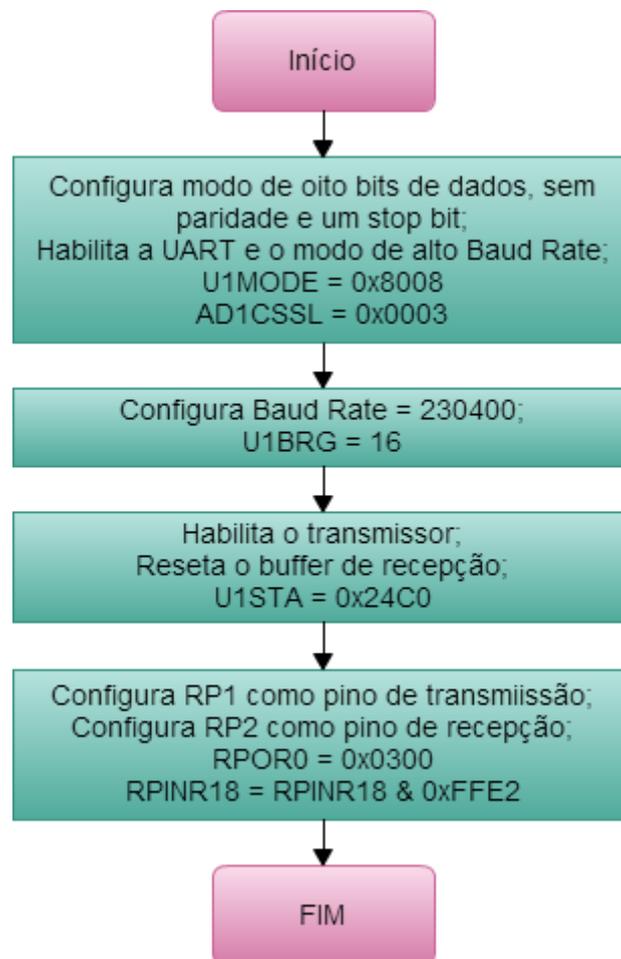


Figura 4.15. Fluxograma da função de configuração da UART.

Determina-se o valor que é escrito no registrador U1BRG para obter um *baud rate* de 230400 da seguinte forma:

$$U1BRG = \frac{F_{CY}}{4 \cdot \text{Baud Rate}} - 1, \quad (4.12)$$

onde

$$F_{CY} = \frac{F_{OSC}}{2}. \quad (4.13)$$

Portanto,

$$U1BRG = \frac{\frac{32000000}{2}}{4 \cdot 230400} - 1. \quad (4.14)$$

$$U1BRG \cong 16,36. \quad (4.15)$$

$$U1BRG = 16. \quad (4.16)$$

Para realizar a transmissão de um caractere no programa utiliza-se a função `put_UART()`. Quando esta função é chamada, ela verifica se o *buffer* de transmissão está cheio, se estiver, ela aguarda até ter espaço para colocar outro dado no *buffer*. Assim que o *buffer* estiver disponível o dado é colocado nele por meio da escrita no registrador U1TXREG. O fluxograma desta função está mostrado na Figura 4.16.

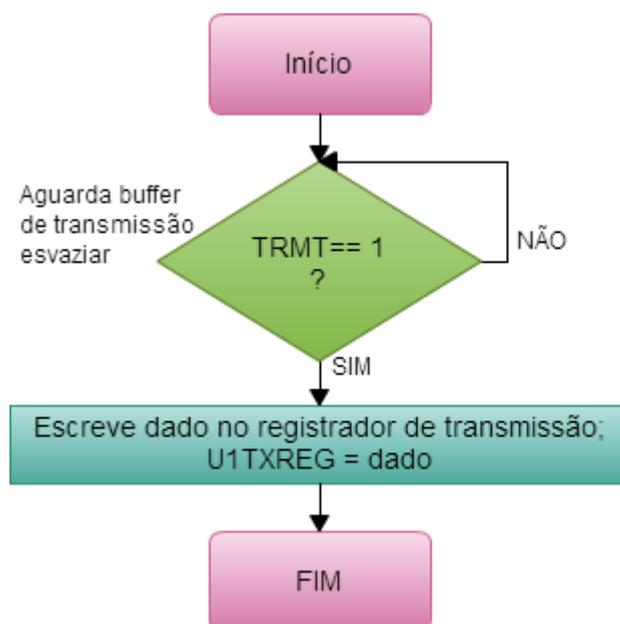


Figura 4.16. Fluxograma da função de envio de dados.

A função utilizada para receber dados é a `get_UART()`. O seu funcionamento, ilustrado no fluxograma da Figura 4.17, é realizado da seguinte maneira: quando `get_UART()` é chamada no programa, verifica se há dado para ser lido no *buffer* de recepção, se não houver, o microcontrolador fica esperando até que chegue pelo menos um dado no buffer. Assim que o dado está disponível, ele é capturado pela função através da leitura do registrador U1RXREG.

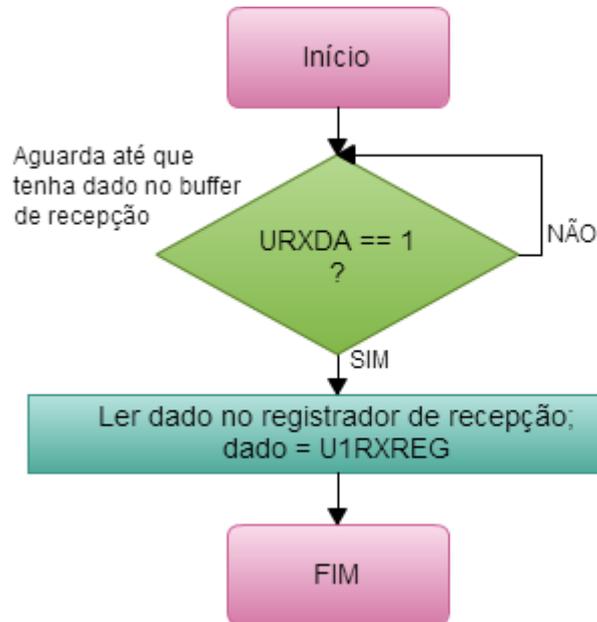


Figura 4.17. Fluxograma da função de recepção de dados.

#### 4.6.7 CONVERSOR USB/RS-232

Como é utilizado o protocolo RS-232 por meio da UART para realizar a comunicação no Kit, e a porta de comunicação utilizada neste protocolo não está presente na maioria dos computadores modernos, principalmente em *notebooks*, foi utilizado o componente FT232RL (Figura 4.18) da fabricante de *chips* FTDI, para resolver este impasse.



Figura 4.18. Chip conversor USB/RS-232.

Este dispositivo consiste em um conversor USB/RS-232, possibilitando assim a utilização das portas USB dos computadores atuais mesmo diante da utilização do protocolo de comunicação serial RS-232. O FT232RL apresenta as seguintes características principais:

- Pode ser utilizado em várias plataformas como o Windows e o Linux;
- Os drives estão disponíveis gratuitamente;
- Apresenta alta velocidade de comunicação.

Na Figura 4.19, tem-se o esquema de ligação do conversor FT232RL realizando a ponte entre o microcontrolador e o conector USB.

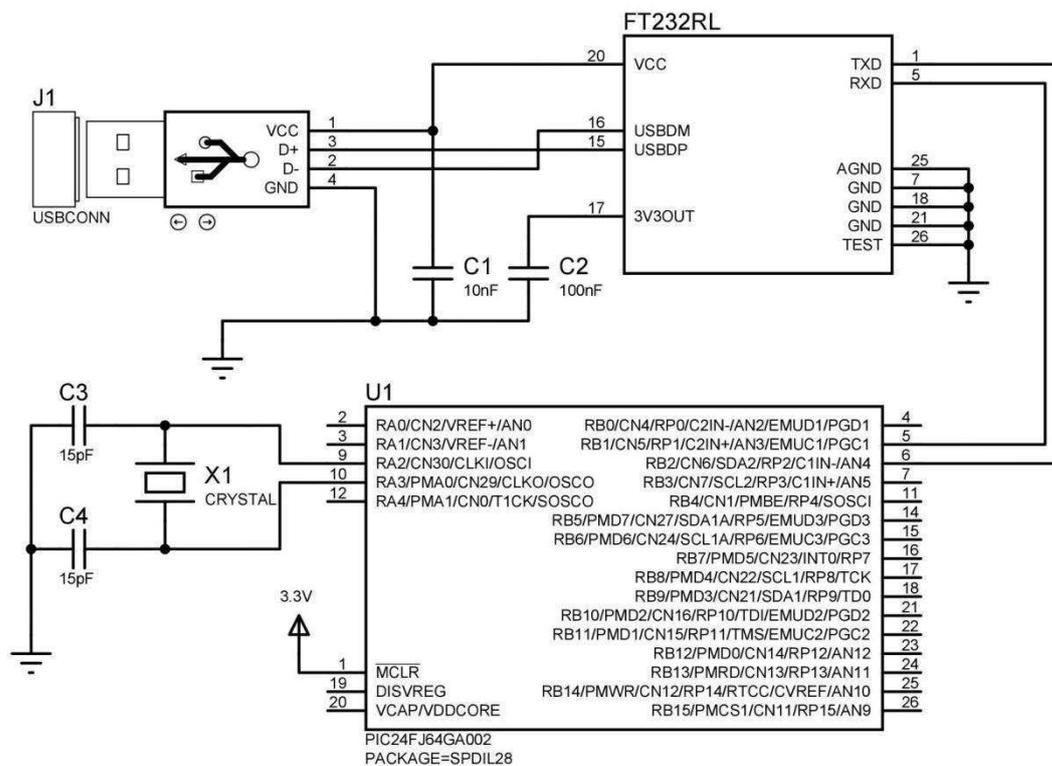


Figura 4.19. Esquema de ligação do FT232RL no Kit.

## 4.7 PROGRAMAÇÃO DO MICROCONTROLADOR

Levando em consideração tanto as funções que serão utilizadas para a conversão A/D quanto a comunicação definida, foi criado um algoritmo que controla a amostragem dos sinais de entrada e envia os dados para a interface gráfica presente no

PC. Para maximizar a taxa de amostragem e conseguir realizar a aquisição simultânea de dois sinais de até 15 kHz, foi definido que primeiramente é realizado um conjunto de aquisições que é guardado num *buffer*, para em seguida ser enviado para a interface onde o sinal é reconstruído.

Com o tempo para conversão de duas amostras, uma de cada canal, calculado na Seção 4.5.4 como 3,25  $\mu$ s, adicionado com os ciclos de máquina necessários para as operações de guardar o dado e incrementar a posição no *buffer* de amostras, foi obtido um período de amostragem aproximadamente igual a 3,82  $\mu$ s ( $T_a$ ). Para possibilitar a visualização de até quatro ciclos de um sinal periódico de 1 kHz ( $f_{min}$ ) e a realização da função *trigger* na interface gráfica, foi estabelecido para o tamanho do *buffer* de amostras um valor de 2200 posições. Com o auxílio da Figura 4.20, obtemos este valor da seguinte forma:

$$T_{buffer} = n \cdot \frac{1/f_{min}}{T_a} = 4 \cdot \frac{1/1000}{0,00000382} \cong 1047, \quad (4.15)$$

onde

$T_{buffer}$  = número de posições no *buffer* para um sinal;

$n$  = número de ciclos do sinal amostrados;

Como são dois canais o valor é multiplicado por dois. Por garantia, extrapolou-se o tamanho do buffer para  $T_{buffer} = 2200$ .

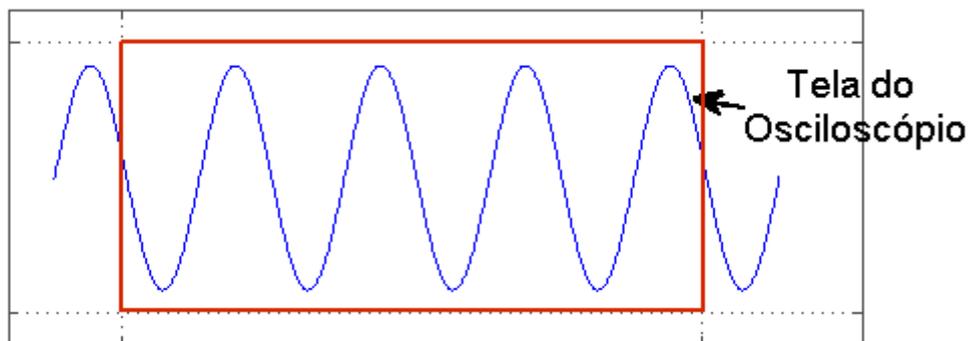


Figura 4.20. Número mínimo de ciclos de um sinal amostrado pelo Kit.

Com as premissas para a implementação do algoritmo e o tamanho do *buffer* definido, foi feito o código-fonte da programação do microcontrolador (Apêndice IV), seguindo o fluxograma da Figura 4.21.

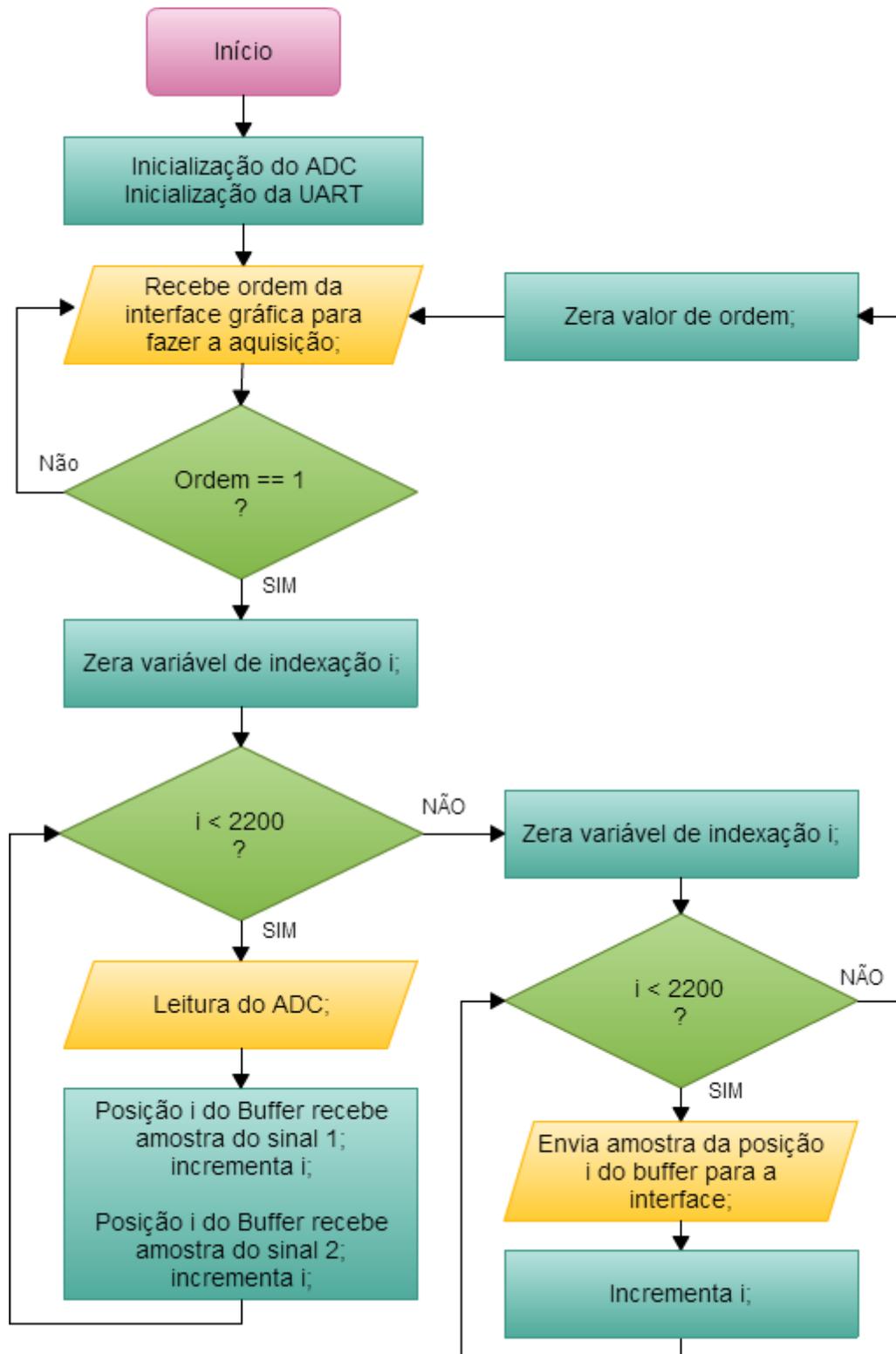


Figura 4.21. Fluxograma da programação do microcontrolador.

## 5 PROJETO DA PLACA DE CIRCUITO IMPRESSO E MONTAGEM

Depois de projetado o *hardware* de aquisição de dados e de posse dos componentes, foi feita uma primeira montagem do sistema em um *proto-board* e, em seguida, realizado ajustes no circuito. Com o sistema funcionando corretamente, o projeto da placa de circuito impresso (PCI) foi elaborado, levando em consideração o encapsulamento de cada componente presente no circuito. O *layout* foi realizado no *software* PROTEUS, na interface para projeto de PCI, ARES. Na Figura 5.1, tem-se o layout da PCI que foi utilizada para o suporte dos componentes do *hardware* do Kit Educacional.

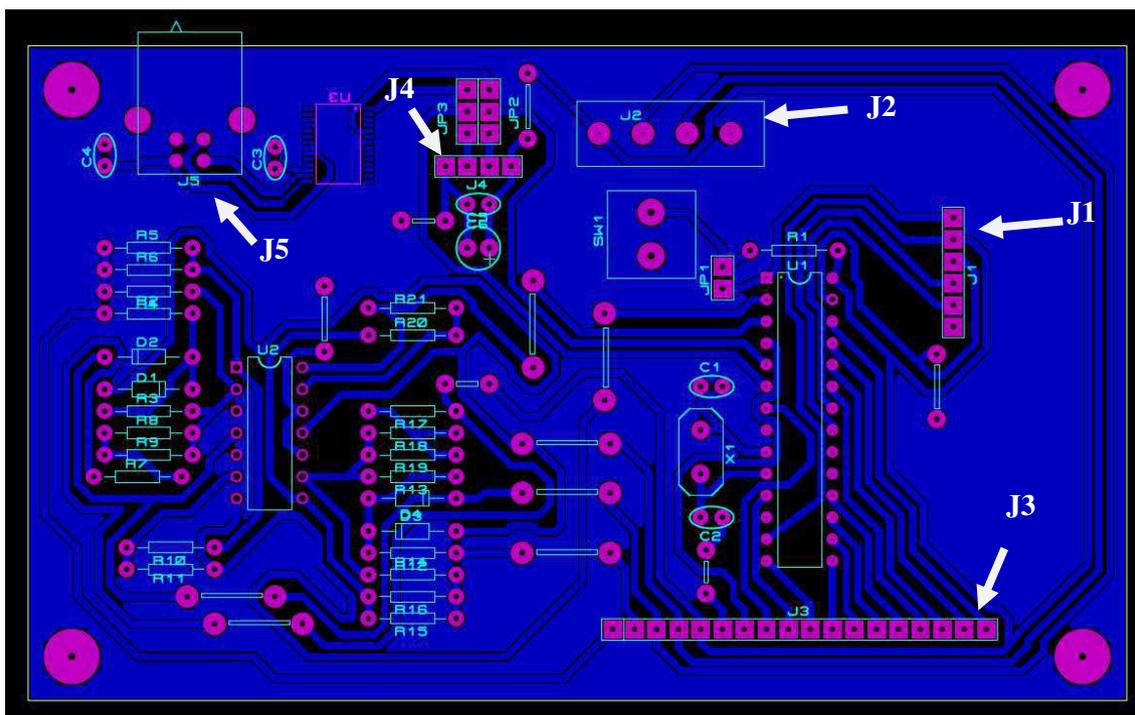


Figura 5.1. *Layout*, no *software* PROTEUS, da placa de circuito impresso do Kit Educacional.

Pode-se observar a presença de cinco conectores, J1 a J5. O primeiro é um barramento para ligação do programador de PIC para realizar a gravação do *firmware* do microcontrolador. O conector J2 é para fonte de alimentação do sistema que deve ter as tensões de +3,3, +12 e -12 V. Os dois pinos, da esquerda para direita, do barramento J3 são os dois canais de entrada analógica. Os pinos restantes deste conector estão

conectados aos pinos do microcontrolador que não foram utilizados, podendo no futuro serem acessados para expansão do sistema. O quarto conector foi colocado para que se possa utilizar uma placa conversora USB/Serial externa, caso ocorra defeito no FT232 da placa. O J5 é um conector USB tipo B para realizar a comunicação com o computador.

Como não havia equipamentos adequados para a fabricação da PCI na Datasonic e também para liberar tempo para desenvolver a interface, a confecção da placa foi um serviço contratado. Com a placa fabricada foram soldados os componentes e realizado testes. Os resultados dos testes são apresentados no capítulo 7. Na Figura 5.2, vê-se a face com cobre da PCI, já com os componentes soldados.

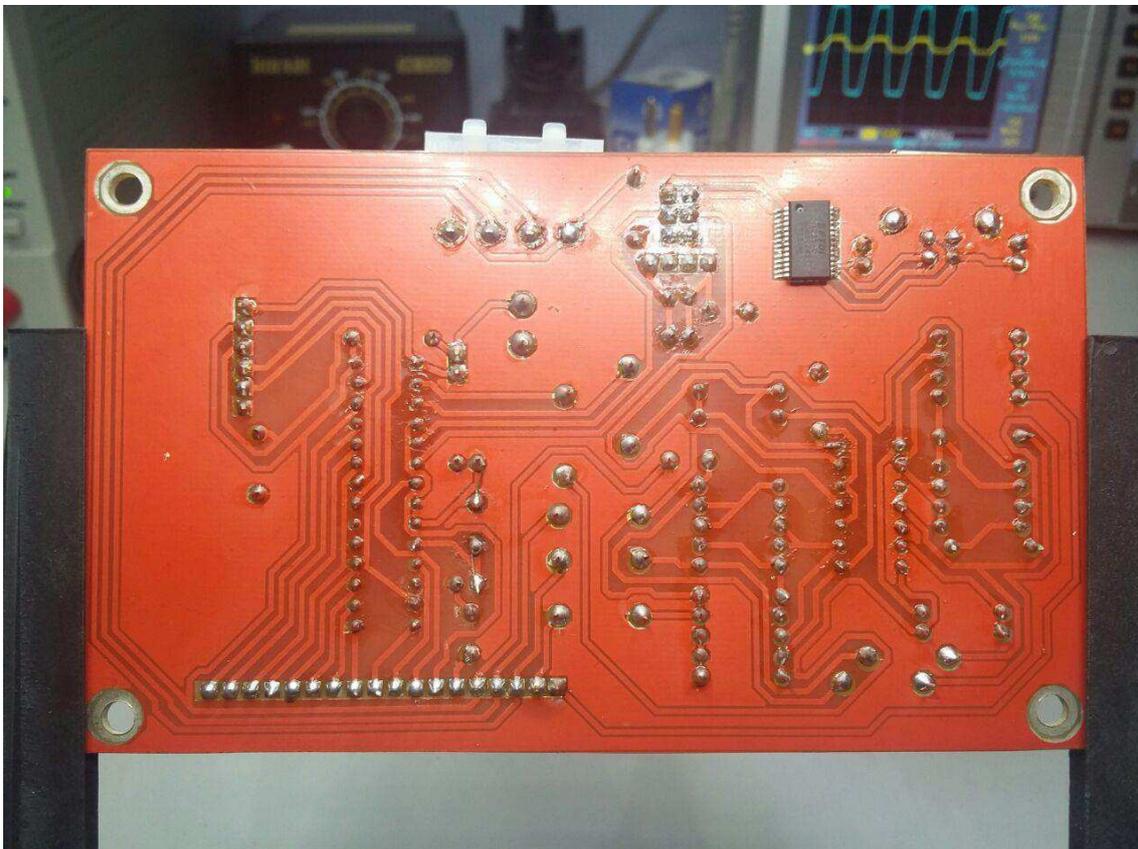


Figura 5.2. PCI do Kit Educacional.

## 6 INTERFACE GRÁFICA

O DAQ não trabalha sem *software*, pois este é executado no computador permitindo que o *hardware* seja transformado num sistema completo de aquisição de dados, analisando, visualizando e controlando sistemas.

Este primeiro enfoque do Kit está direcionado para grandezas envolvidas com os conceitos de circuitos elétricos, ou seja, sinais elétricos, principalmente na forma senoidal. Portanto, a interface foi desenvolvida com direção para esta abordagem baseando-se nos controles de um osciloscópio.

A interface foi criada no *software* MATLAB, que é uma ferramenta bastante difundida na área de engenharia. Além disso, é adequada para implementar e testar soluções com facilidade e precisão sem se preocupar com detalhes específicos de linguagem de programação. Na Figura 6.1 tem-se uma imagem da interface e neste capítulo será explicada a sua concepção e os algoritmos dos controles do osciloscópio virtual e de reconstrução dos sinais.

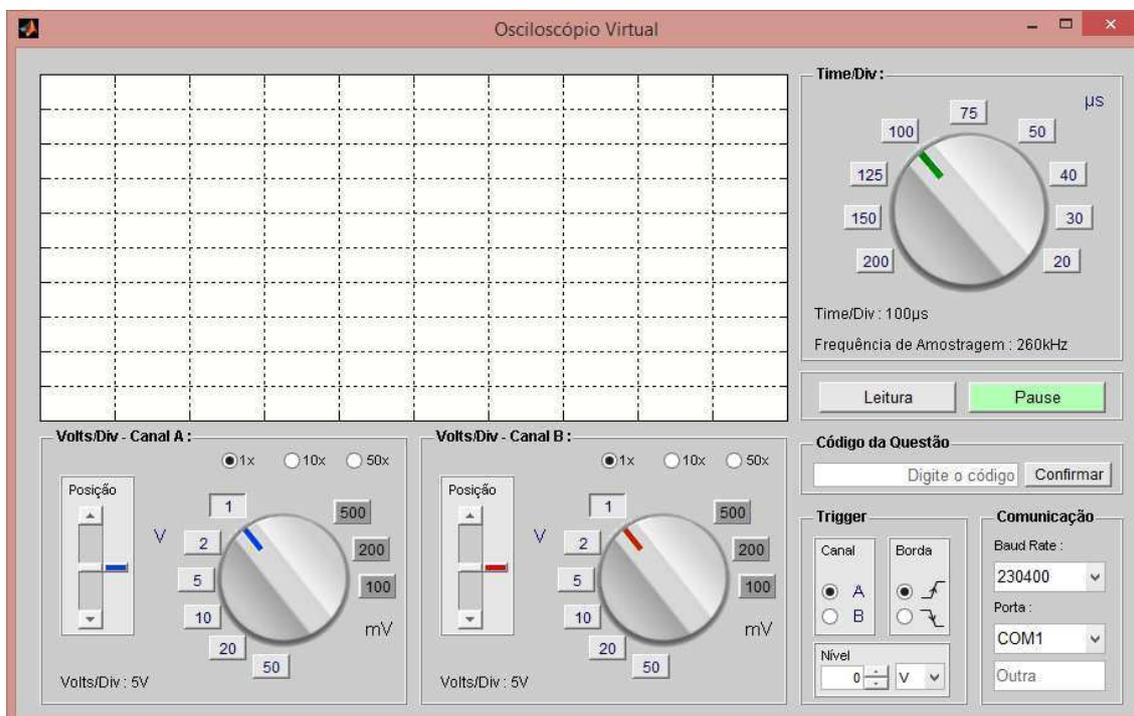


Figura 6.1. Interface gráfica do Kit Educacional.

## 6.1 GRÁFICO

Na interface ilustrada na Figura 6.1 pode-se perceber a presença de um gráfico no quadrante esquerdo superior que simula a tela de um osciloscópio. Por meio dele é possível visualizar em tempo real os sinais elétricos adquiridos pelo DAQ. Esta tela é dividida em 10 segmentos tanto no eixo vertical como no horizontal que facilita a medição de características dos sinais tais como: amplitude, frequência e diferença de fase entre os sinais. O valor de cada divisão é configurado por meio de controles que são explicados mais a frente.

## 6.2 PAINEL PARA CONFIGURAR COMUNICAÇÃO

Tem-se na interface um painel onde é possível configurar parâmetros da comunicação serial entre a interface e o DAQ. Estes parâmetros são o *Baud Rate* e a porta de comunicação serial (COM) do PC em que o *hardware* está conectado. Os valores *default* são 230400 e COM1, mas deve-se configurá-los de acordo com as especificações do DAQ. Só é possível atuar neste painel quando a interface está pausada.

## 6.3 ALGORITMO PARA RECONSTRUIR OS SINAIS

Depois da comunicação configurada, ao se pressionar o botão "Leitura" a interface irá enviar uma ordem ao DAQ para que ele realize a aquisição de dados e envie as 2200 amostras. A interface ao receber os dados realiza uma conversão das amostras digitais para um formato em números decimais que representam o valor analógico dos sinais. Na sequência, as amostras do canal 1 são separadas das do canal 2. Como pode acontecer durante a etapa de aquisição a captura de ruídos de alta frequência sobrepostos aos sinais de interesse, foi adicionada à interface um algoritmo de filtragem passa-baixa digital, removendo assim as componentes indesejadas. Em seguida, é realizada a função *trigger* do osciloscópio onde é determinado a partir de qual amostra o sinal será plotado no gráfico. Antes de plotar os sinais, são realizadas interpolações cúbicas entre amostras consecutivas para diminuir o efeito de digitalização dos sinais

como ilustrado na Figura 6.2. Finalmente as amostras interpoladas são plotadas no gráfico e os sinais podem ser vistos na tela do osciloscópio virtual. Este procedimento se repete várias vezes até que o botão de "Pause" seja pressionado. Na Figura 6.3, tem-se o fluxograma que sequencia estes passos.

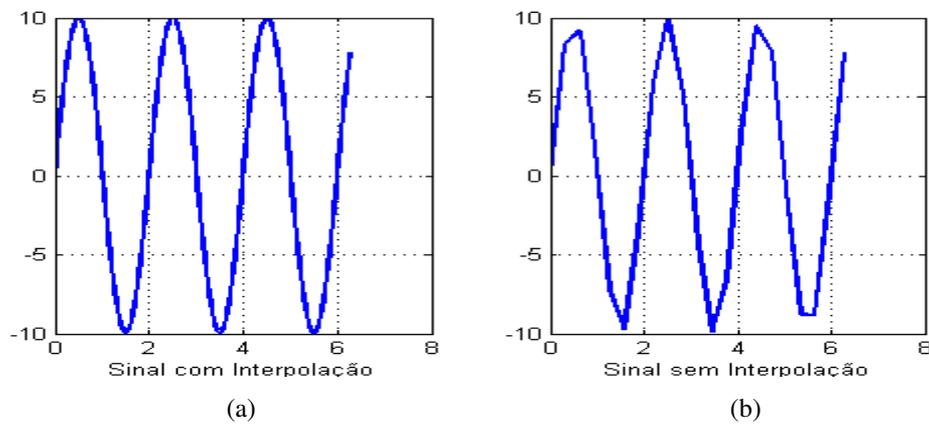


Figura 6.2. Efeito da interpolação.

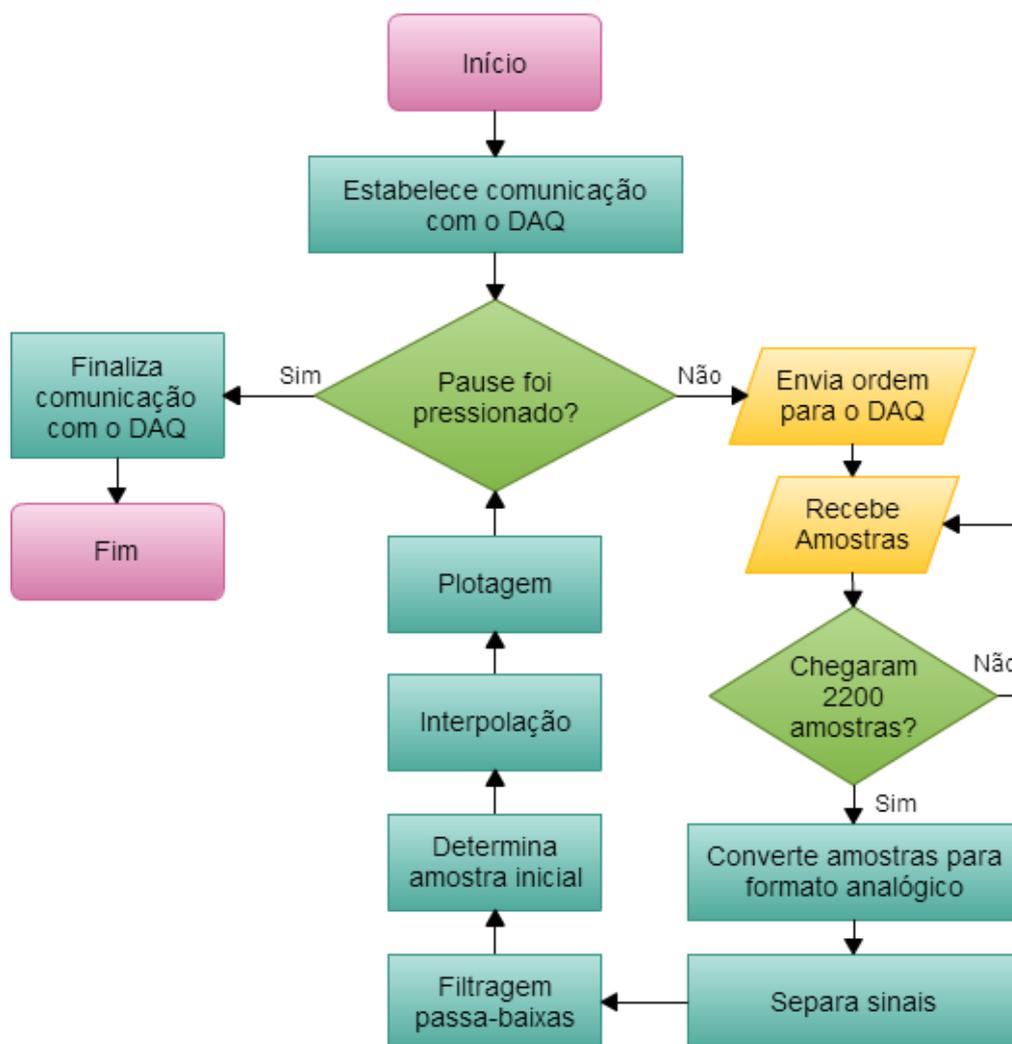


Figura 6.3. Fluxograma da rotina que é executada quando o botão "Leitura" é pressionado.

## 6.4 PAINÉIS DE CONTROLE DA ESCALA DE TENSÃO

Há dois painéis de controle da escala de tensão na interface, um para o canal 1 e outro para o canal 2. Nestes painéis é possível controlar o valor das divisões verticais do gráfico de 100 mV até 50 V/Div. Também é possível deslocar o sinal no eixo vertical por meio de um controle de posição e aplicar um ganho digital de 1, 10 ou 50 vezes. Com este ganho pode-se representar sinais de até 500 V de amplitude.

## 6.5 PAINEL DE CONTROLE DA ESCALA DE TEMPO

O painel Time/Div realiza uma operação similar ao controle da escala de tensão, mas neste caso é controlado o valor das divisões do eixo horizontal do gráfico. Os valores disponíveis variam na faixa de 20 a 200  $\mu$ s que possibilitam a visualização de ciclos completos de sinais dentro da faixa de frequência de 1 a 15 kHz.

## 6.6 PAINEL DE CONTROLE DO *TRIGGER*

O *trigger* tem a função de definir o ponto onde os sinais começam a serem plotados no gráfico. Isto permite que o sinal fique mais estável no gráfico melhorando a visualização e facilitando as medições. No painel de controle é permitido escolher entre os sinais do canal 1 e 2 como fonte do *trigger* e o nível. Desta forma, o sinal do canal selecionado será plotado no gráfico iniciando-se da amostra que apresenta o valor mais próximo do nível configurado. Outro controle presente é a seleção entre as bordas de subida ou descida, isto é, selecionar se o sinal será plotado inicialmente por uma região crescente ou decrescente da sua forma de onda.

## 7 TESTES E RESULTADOS

Para confirmar o funcionamento do *hardware* do Kit junto com a interface desenvolvida, foram realizados testes observando simultaneamente o sinal físico, por meio de um osciloscópio, e o digital pela interface gráfica. Os controles disponíveis no osciloscópio virtual também foram averiguados. Na Figura 7.1, tem-se a montagem do Kit com os cabos do osciloscópio analógico, da fonte de alimentação, da fonte de sinais e da comunicação USB, conectados para realizar a bateria de testes que é apresentada mais adiante.

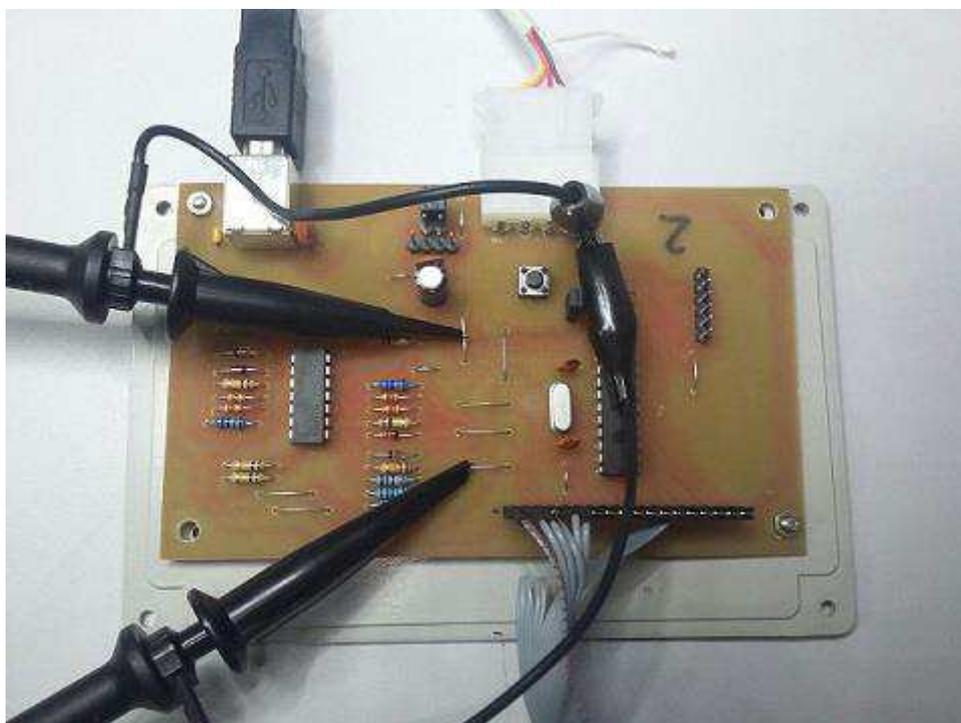


Figura 7.1. Montagem de testes do Kit.

### 7.1 TESTE DO CIRCUITO DE CONDICIONAMENTO

O primeiro teste foi referente ao circuito de condicionamento e de proteção das entradas do microcontrolador. Na Figura 7.2, onde tem-se sinais físicos na tela do osciloscópio analógico, verificou-se o correto funcionamento do circuito de condicionamento. Foi aplicado um sinal seno de 10 V de amplitude (azul) em um dos

canais de entrada do Kit e ele foi convertido para um sinal entre 0 e +3,3 V na entrada do PIC, como vê-se no sinal amarelo, confirmando o funcionamento deste subsistema.

A operação do circuito de proteção é conferida na Figura 7.3, utilizando para tal um sinal com amplitude acima dos 10 V especificado. Conforme esperado, pode-se observar o correto funcionamento dos circuitos ceifadores analisando o aspecto quadricular das formas de onda na figura. A tensão aplicada na entrada do PIC ficou limitada em 0 e 3,3 V comprovando a proteção das entradas analógicas.

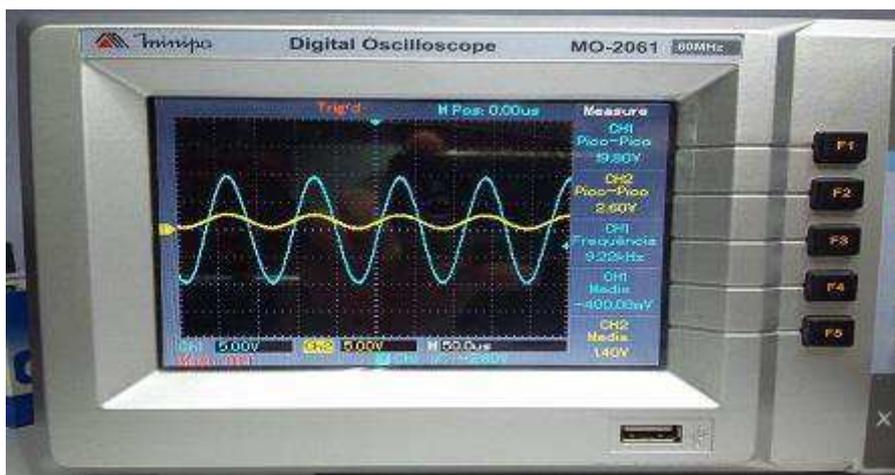


Figura 7.2. Atuação do circuito de condicionamento.

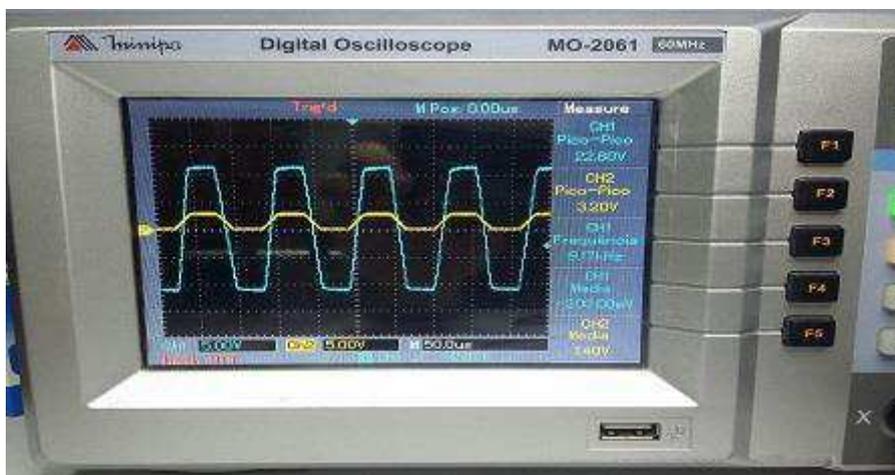


Figura 7.3. Atuação do circuito de condicionamento e proteção.

## 7.2 VERIFICAÇÃO DA RECONSTRUÇÃO DO SINAL

Este teste foi realizado para examinar a reconstrução, por parte da interface gráfica, dos sinais adquiridos pelo *hardware* do Kit. Foram aplicados sinais de amplitude e frequência conhecidas nos dois canais de entrada. Para o canal A foi usado um sinal seno de amplitude de 5 V e frequência 9,35 kHz, e no canal B um sinal com a

mesma frequência e amplitude de 10 V. Estes sinais físicos são apresentados na Figura 7.4.

Desta forma, os valores medidos na interface devem apresentar valores próximos. Na Figura 7.5, cada semiciclo do sinal do canal A (azul) preenche uma divisão vertical enquanto que no canal B preenche duas. As escalas de volts/div de ambos os canais estão selecionadas com valor 5 V, portanto, as amplitudes medidas na interface foram aproximadamente 5 e 10 V para os sinais dos canais A e B, respectivamente. A frequência medida foi de 9,5 kHz, pois os sinais preencheram 1,4 divisões horizontais para um valor da escala time/div de 75  $\mu$ s.

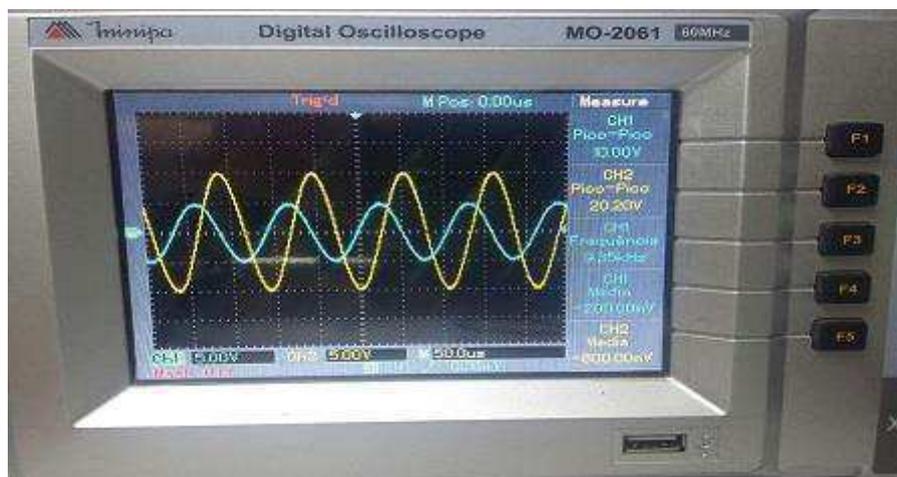


Figura 7.4. Sinais de entrada para o teste de reconstrução de sinais.

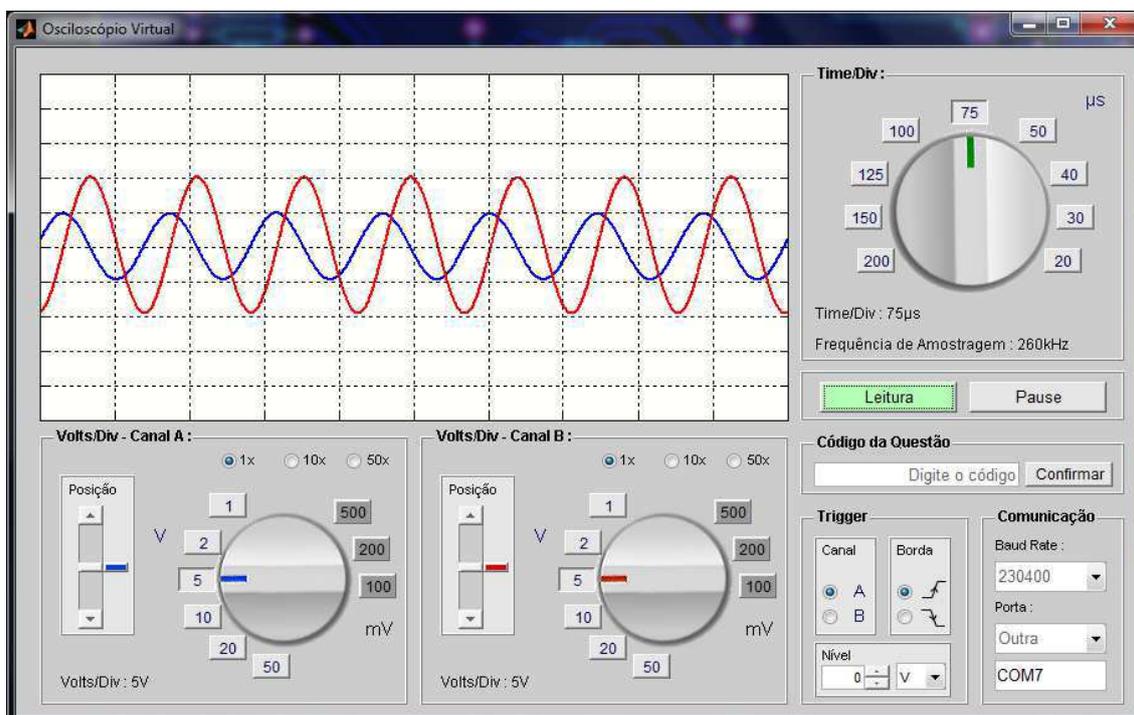


Figura 7.5. Sinais reconstruídos na interface gráfica.

### 7.3 TESTE DE CONSERVAÇÃO DA DIFERENÇA DE FASE

Uma grandeza de grande relevância em estudos de circuitos CA é a diferença de fase entre os sinais. Neste teste é comprovado que na reconstrução dos sinais pela interface a diferença de fase é conservada. Nas Figuras 7.6 e 7.7, observa-se este acontecimento utilizando dois sinais defasados de aproximadamente  $180^\circ$ . Já nas Figuras 7.8 e 7.9, tem-se sinais defasados de alguns poucos graus.

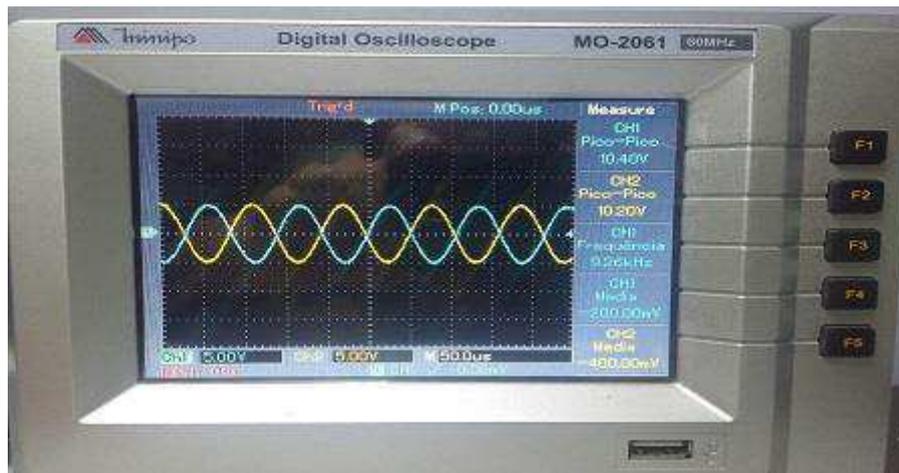


Figura 7.6. Sinais físicos defasados de  $180^\circ$ .

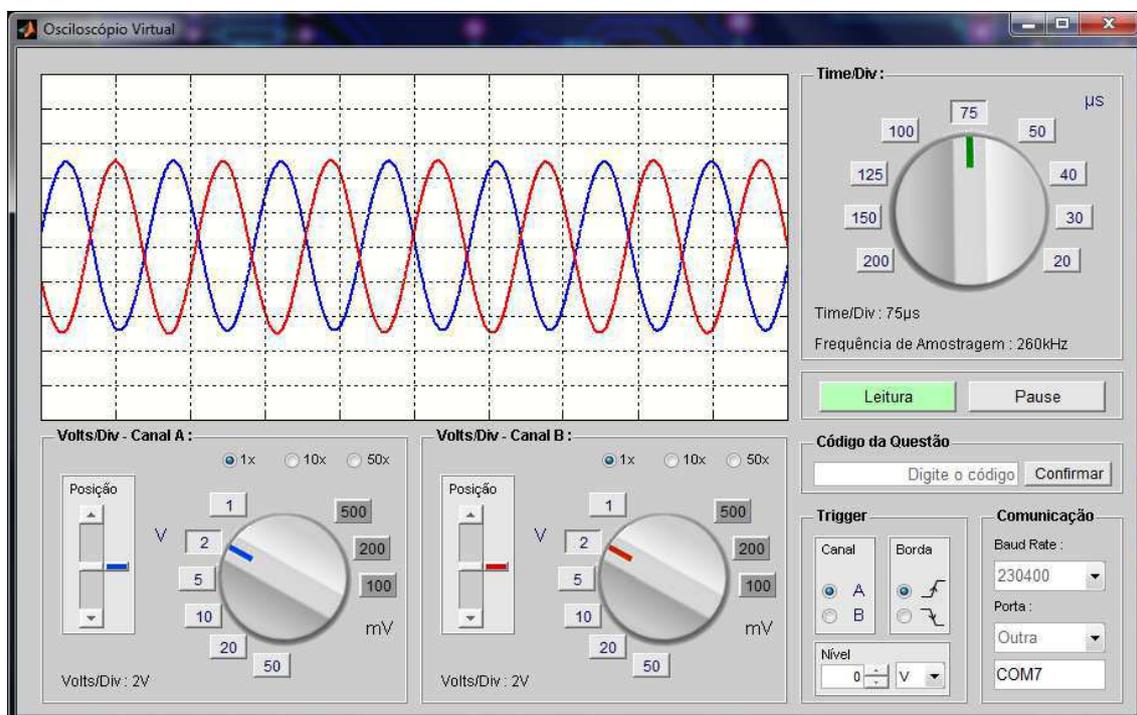


Figura 7.7. Sinais defasados de  $180^\circ$  reconstruídos na interface.

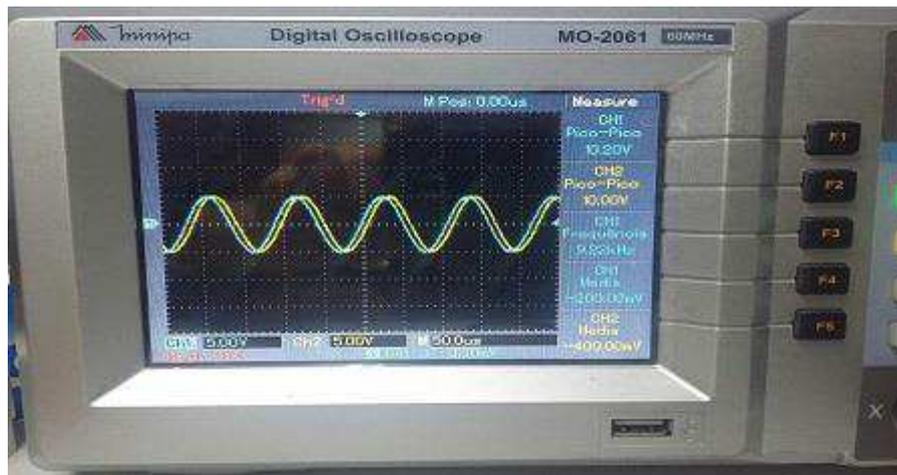


Figura 7.8. Sinais físicos defasados de  $24^\circ$ .

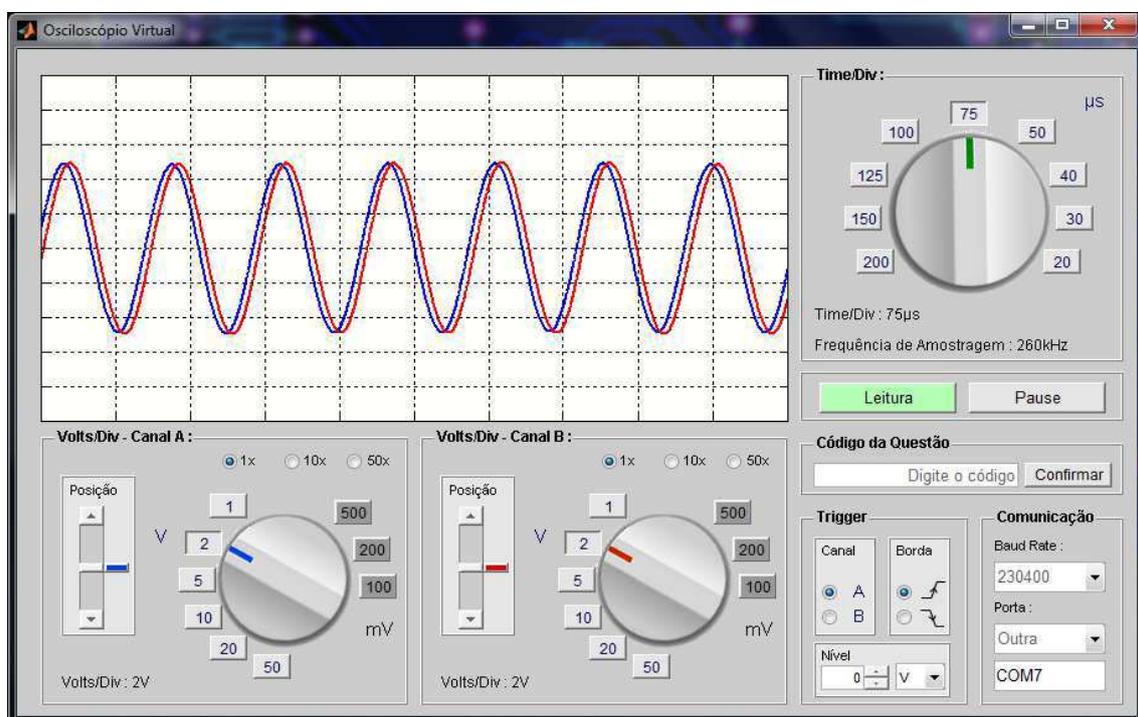


Figura 7.9. Sinais defasados de  $24^\circ$  reconstruídos na interface.

## 7.4 TESTE DA FUNÇÃO TRIGGER DA INTERFACE

Neste teste foi verificado se os botões do painel de controle do *trigger* da interface gráfica funcionam conforme o esperado. Primeiramente foi realizada a aquisição de sinais utilizando o canal A como fonte de *trigger* e a borda de subida selecionada. Na Figura 7.10, observa-se que o sinal do canal A (azul) começa a ser plotado do valor zero, que foi o valor selecionado para o nível do *trigger*, e o semiciclo positivo é o primeiro a ser visualizado. Na sequência do teste, foi selecionada a borda de

descida e o canal B como fonte do *trigger*. Na Figura 7.11, vê-se que os sinais foram plotados a partir do semiciclo negativo do sinal do canal B (vermelho).

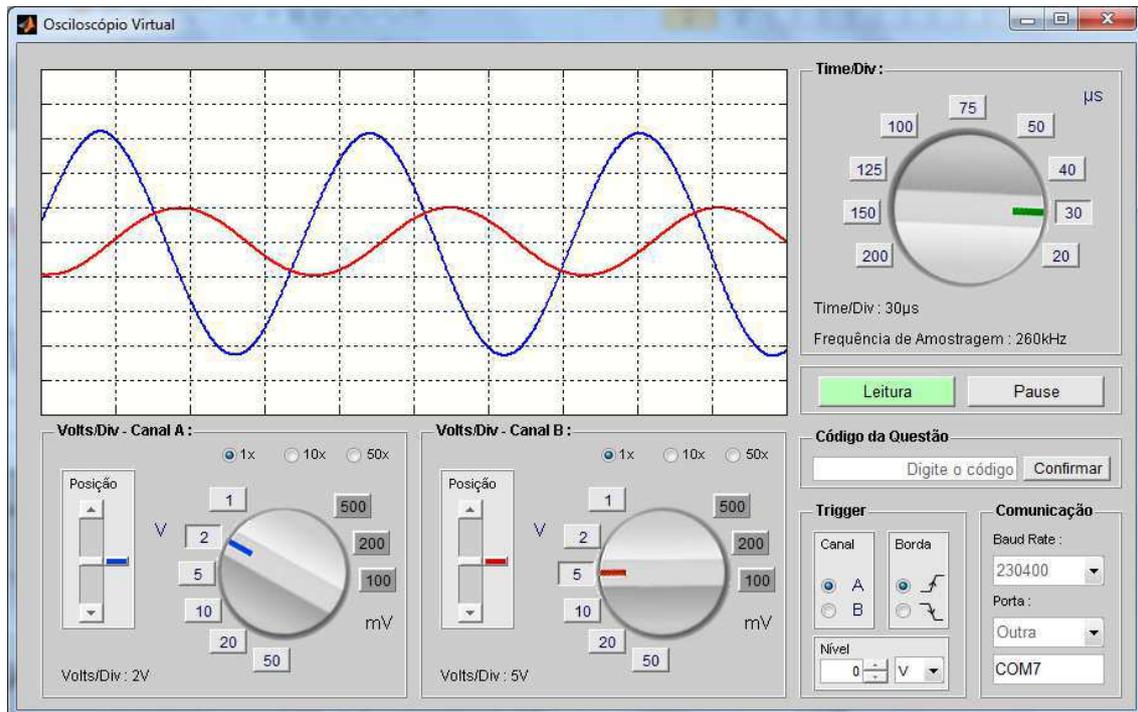


Figura 7.10. Atuação do *trigger* configurado com borda de subida e o canal A como fonte.

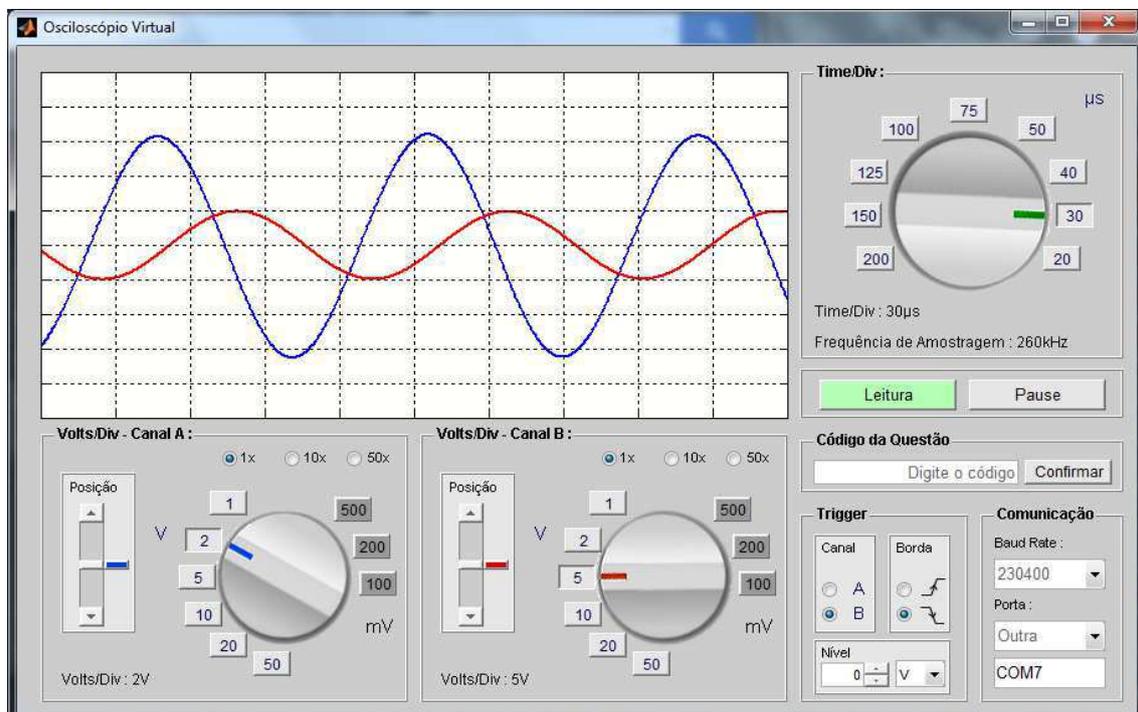


Figura 7.11. Atuação do *trigger* configurado com borda de descida e o canal B como fonte.

## 7.5 TESTE DAS FUNÇÕES GANHO E POSIÇÃO

Finalizando os testes, foram verificadas as funções que aplicam um ganho digital e que modificam a posição vertical dos sinais. Como pode-se ver na Figura 7.12, o sinal do canal A apresenta uma amplitude de 200 V, bem maior que o máximo permitido na entrada do Kit. Na Figura 7.13, checkou-se o controle de posição onde o sinal do canal A foi deslocado para uma porção superior e o do B para uma parte inferior do gráfico.

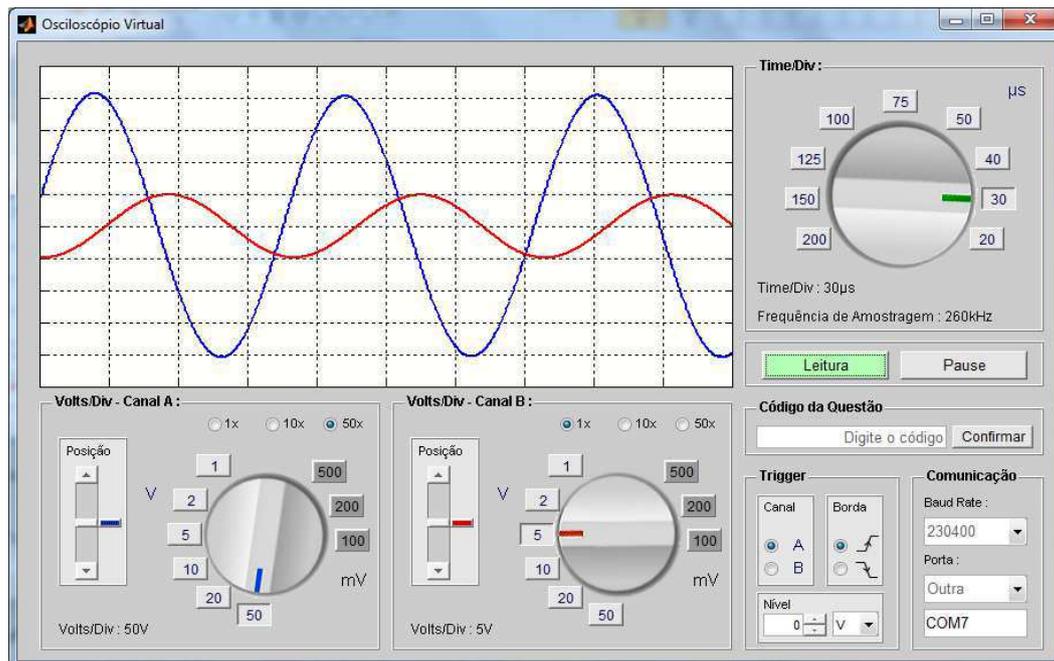


Figura 7.12. Verificação da função ganho da interface gráfica.

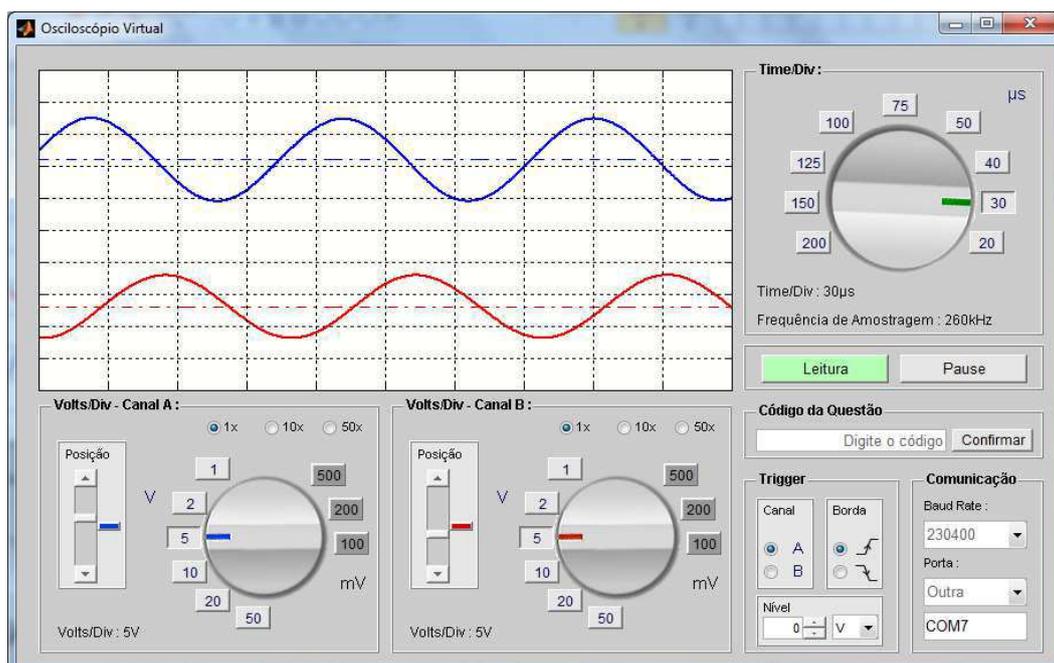


Figura 7.13. Verificação do controle de posição dos sinais no gráfico.

## 8 CONCLUSÃO

Ficou evidenciado pelos resultados mostrados na seção anterior que foi obtido êxito nas atividades desenvolvidas e no objetivo do estágio que era o projeto de um sistema completo de aquisição de dados, isto é, *hardware e software*.

Ressalta-se que a atividade profissional realizada no estágio como desenvolvedor de sistemas embarcados foi bastante importante para consolidar, por meio da prática, muitos dos conhecimentos teóricos estudados no curso de Engenharia Elétrica que, devido à ementa do curso, não são comprovados por meio de experimentos.

Constata-se também que o estágio serviu para firmar habilidades técnicas do estagiário, tais como: montagem, soldagem, utilização de osciloscópio e gerador de sinais. Ademais, contribuiu de forma direta no aumento da perícia em *softwares* de simulação de circuitos, de modelagem matemática e de desenvolvimento de interfaces de usuário, habitualmente utilizados em setores de projetos de sistemas eletrônicos.

As disciplinas cursadas na universidade que mais ajudaram no estágio foram: Sistemas de Aquisição de Dados e Interface, Processamento Digital de Sinais, Eletrônica e Arquitetura de Sistemas Digitais. Entretanto, esta última poderia ter ajudado ainda mais se os microcontroladores fossem estudados nela em mais baixo nível, isto é, na manipulação direta dos registradores.

Para finalizar, destaca-se a experiência adquirida no estágio ao se realizar tarefas com prazos curtos de entrega, trabalhar em uma equipe multidisciplinar e conviver com profissionais de várias áreas tais como: tecnologia da informação, desenho industrial, gestão de vendas e engenharia elétrica.

## BIBLIOGRAFIA

FTDI CHIP. **FT232R USB UART IC Datasheet**. Version 2.10. 2010.

JASIO, L. D. **Programming 16-bit Microcontrollers in C: Learning to Fly the PIC 24**. Elsevier, 2007. 379 p.

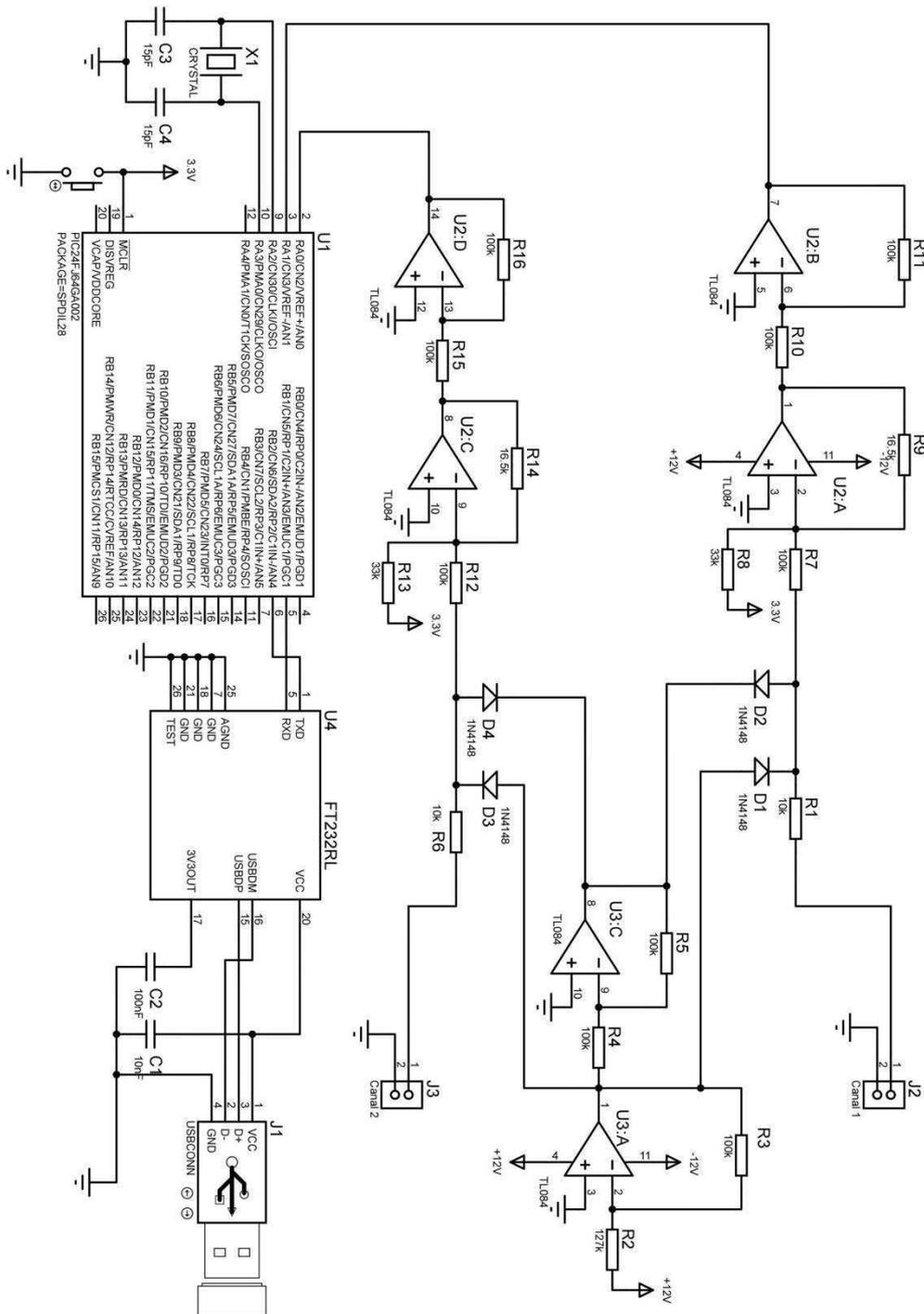
MICROCHIP. **PIC24FJ64GA004 Family Data Sheet: 28/44-Pin General Purpose, 16-Bit Flash Microcontrollers**. 2008.

MORRIS, A. S. **Measurement & Instrumentation Principles**. 3rd ed. Butterworth-Heinemann, 2001. 475 p.

PARK, J.; MACKAY, S. **Practical Data Acquisition for Instrumentation and Control Systems**. Elsevier, 2003. 407 p.

TEXAS INSTRUMENTS. **PC16550D**. rev. C. 2015.

# APÊNDICE I – ESQUEMA ELÉTRICO DO *HARDWARE*



## APÊNDICE II – LISTA DE COMPONENTES DO

### *HARDWARE*

<b>Tipo</b>	<b>Referência</b>	<b>Descrição na Placa</b>	<b>Quantidade</b>	<b>Valor</b>
CI	PIC24FJ64GA002	U1	1	-
CI	TL084	U2, U3	2	-
CI	FT232R	U4	3	-
Resistor	-	R1, R6	2	10 k $\Omega$
Resistor	-	R2	1	127 k $\Omega$
Resistor	-	R3- R5, R7, R10-R12, R15, R16	9	100 k $\Omega$
Resistor	-	R8, R13	2	33 k $\Omega$
Resistor	-	R9, R14	2	16,5 k $\Omega$
Capacitor	-	C1	1	10 nF
Capacitor	-	C2	1	100 nF
Capacitor	-	C3, C4	2	15 pF
Cristal	-	X1	1	32 MHz
Diodo	1N4148	D1-D4	4	-
Conector	USB tipo B - Fêmea	J1	1	-
Chave	Push-Button	SW1	1	-

# APÊNDICE III – BIBLIOTECA DE FUNÇÕES EM C

## UTILIZADAS NA PROGRAMAÇÃO DO PIC24

```

//*****
//
// kit_library.c
//
// Biblioteca de funções utilizadas na programação do microcontrolador do KIT
//
// Autor: Márcio Herson de Aguiar
//
//*****

//*****
// Endereço dos registradores utilizados
/* UART */
#word U1MODE = 0x0220
#word U1STA = 0x0222
    #bit TRMT = 0x0222.8
    #bit URXDA = 0x0222.0
#word U1TXREG = 0x0224
#word U1RXREG = 0x0226
#word U1BRG = 0x0228
#word RPOR0 = 0x06C0
#word RPINR18 = 0x06A4
    #bit DOZEN = 0x0744.11

/* CONVERSADOR AD */
#word AD1CON1 = 0x0320
    #bit ADON = 0x0320.15
    #bit SAMP = 0x0320.1
    #bit DONE = 0x0320.0
    #bit ASAM = 0x0320.2
#word AD1CON2 = 0x0322
#word AD1CON3 = 0x0324
#word AD1CHS = 0x0328
#word AD1PCFG = 0x032C
#word AD1CSSL = 0x0330
#word ADC1BUF0 = 0x0300
#word ADC1BUF1 = 0x0302
#word ADC1BUF2 = 0x0304
#word IFS0 = 0x0084
    #bit AD1IF = 0x0084.13
//*****

//*****
// Funções

// Função para inicializar o conversor A/D
void iniciar_ADC(void)
{
    AD1PCFG = 0xFFFC; // AN0,AN1 e AN2 selecionado como canal de leitura
                    // analógica
    AD1CSSL = 0x0003;
    AD1CON1 = 0x20E0; // Determina conversão automática e
                    // Formato de saída como número inteiro
                    // Não opera no modo ocioso

```

```

AD1CON2 = 0x0404; // Determina tensões de referência como Vdd e Vss,
                // Interrompe a cada três conversões
AD1CON3 = 0x0101; // Seleciona clock externo como clock de conversão AD,
                //tempo de conversão e amostragem
                // configura sample time = 1Tad
                // clock de conversão A/D como 2*Tcy
ADON = 1; // Liga conversor AD
}

// Função para realizar a conversão A/D
void ler_conversor_AD(void)
{
    AD1IF = 0;
    DONE = 0;
    ASAM = 1;
    while (!AD1IF); // Espera o processo de conversão terminar
    ASAM = 0;
}

// Função para inicializar a UART
void iniciar_UART(void)
{
    DOZEN = 0; // desabilita o DOZE --> necessário para calcular o baud
                //corretamente
    U1MODE = 0x8008; // configuração (8-bits data, no parity, um stop bit)
                // habilita a UART
                // habilitar alto Baud Rate
    U1BRG = 16; // Baud Rate 230400
    U1STA = 0x24C0; // habilita o transmissor
                // reseta o buffer de recepção e esvazia RSR
    RPOR0 = 0x0300; // associa RP1 com U1TX
    RPINR18 = RPINR18 & 0xFFE2; // associa RP2 com U1RX
}

// Função para enviar byte
void put_UART(int16 dado)
{
    while (!TRMT); //aguarda o buffer de transmissão esvaziar
    U1TXREG = dado; //escreve dado no registrador de transmissão
}

// Função para receber byte
int16 get_UART(void)
{
    while (!URXDA); //espera até que tenha um dado no buffer de recepção
    return U1RXREG; //retorna o dado do buffer de recepção
}
//*****

```

# APÊNDICE IV – CÓDIGO FONTE DA PROGRAMAÇÃO DO MICROCONTROLADOR

```

//*****
#include <24FJ64GA002.h>
#include "kit_library.c"

#FUSES NOWDT           //No Watch Dog Timer
#FUSES NOJTAG          //JTAG disabled
#FUSES NOPROTECT      //Code not protected from reading
#FUSES NOWRT          //Program memory not write protected
#FUSES NODEBUG        //No Debug mode for ICD
#FUSES ICSP3          //ICD communication channel 3
#FUSES NOIOL1WAY      //Allows multiple reconfigurations of
                        //peripheral pins
#FUSES NOWINDIS       //Watch Dog Timer in Window mode
#FUSES WPOSTS1        //Watch Dog Timer PostScalar 1:1
#FUSES NOIESO         //Internal External Switch Over mode disabled
#FUSES PR             //Promiary Oscillator
#FUSES NOOSCIO        //OSC2 is general purpose output
#FUSES HS             //High speed Osc (> 4mhz)
#FUSES I2C1SELD
#use delay(clock=32000000) // Maior Clock escolhido - 32MHz (HS - High
                        //Speed)

#word TRIS_B = 0x02C8
#word TRIS_A = 0x02C0

// Constantes
#define TAM_BUFFER 2200

// Variáveis
unsigned int16 i=0, ordem=0, buffer[TAM_BUFFER];

void main() // Função principal
{
    //Inicializações
    TRIS_B = TRIS_B | 0x0004;
    TRIS_A = TRIS_A | 0x0003;
    iniciar_ADC();
    iniciar_UART();

    while (1) // Laço contínuo
    {
        ordem = get_UART(); // espera ordem para enviar os dados do buffer

        if(ordem == 1)
        {
            // OBTENÇÃO DAS AMOSTRAS DOS SINAIS REAIS
            i = 0; // Variável de indexação é zerada
            while (i < TAM_BUFFER) // Laço perdura até última posição do buffer
            {
                ler_convertor_AD();
                buffer[i] = ADC1BUF0; // A posição corrente do buffer é preenchida
                                    //com o resultado da leitura no canal 0
                i++; // Próxima posição
                buffer[i] = ADC1BUF1; // A próxima posição é preenchida com a
                                    //leitura do canal 1
            }
        }
    }
}

```

```
        i++;
    }
    // TRANSMISSÃO
    i = 0;
    while (i < TAM_BUFFER) // Envio das amostras do sinal lido pelo
                            //canal_0
    {
        buffer[i] = buffer[i] >> 2; // Compressão de 10 bits para 8 bits
        put_UART(buffer[i]); // envia a amostra da posição i do buffer
        i++; // Próxima posição do buffer com amostra do canal_0
    }
    ordem = 0; // zera flag para poder esperar uma nova ordem de envio
}
}
}
//*****
```