



CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA



Universidade Federal  
de Campina Grande

ANTONIO NILTON BARBOSA JÚNIOR



Centro de Engenharia  
Elétrica e Informática

TRABALHO DE CONCLUSÃO DE CURSO  
DESENVOLVIMENTO DE UMA PLATAFORMA VIRTUAL DE EXPERIMENTOS PARA A  
DISCIPLINA SISTEMAS DE AQUISIÇÃO DE DADOS E INTERFACE



Departamento de  
Engenharia Elétrica



Campina Grande  
2017



ANTONIO NILTON BARBOSA JÚNIOR

DESENVOLVIMENTO DE UMA PLATAFORMA VIRTUAL DE EXPERIMENTOS PARA A  
DISCIPLINA SISTEMAS DE AQUISIÇÃO DE DADOS E INTERFACE

*Trabalho de Conclusão de Curso 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: Instrumentação Eletrônica

Orientador:

Professor Jaidilson Jó da Silva, D. Sc.

Campina Grande  
2017

ANTONIO NILTON BARBOSA JÚNIOR

DESENVOLVIMENTO DE UMA PLATAFORMA VIRTUAL DE EXPERIMENTOS PARA A  
DISCIPLINA SISTEMAS DE AQUISIÇÃO DE DADOS E INTERFACE

*Trabalho de Conclusão de Curso 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: Instrumentação Eletrônica

Aprovado em        /        /

**Professor Avaliador**  
Universidade Federal de Campina Grande  
Avaliador

**Professor Jaidilson Jó da Silva, D. Sc.**  
Universidade Federal de Campina Grande  
Orientador, UFCG

Dedico este trabalho aos meus pais, que com muita dedicação e amor nunca mediram esforços para que eu cumprisse mais essa etapa.

## AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me concedido todos os meios necessários para a conclusão deste curso. Tenho certeza de que sem Ele, eu não teria conseguido nada.

Agradeço aos meus pais Nilton e Dalva, por terem batalhado tanto para que eu tivesse uma formação de qualidade, mas principalmente por me ensinarem os valores que carregarei pro resto da vida. Também aos meus irmãos Diego e Thayse, que sempre compartilharam comigo os momentos de brincadeira e descontração fazendo a minha vida mais feliz.

Agradeço aos meus familiares, em especial aos meus tios Daniel, Martinha, Ducileide e Deta que sempre se preocuparam com meus estudos e ajudaram meus pais quando sozinhos não podiam arcar com as despesas escolares.

Agradeço também a minha noiva Milena, que sempre soube dar uma palavra de incentivo e motivação nos momentos que mais precisei. Que esteve ao meu lado, seja festejando as conquistas ou consolando as decepções. Muito obrigado!

Agradeço aos inúmeros amigos de curso, pelas madrugadas de estudo e pela comemoração das conquistas. Com certeza esta dura caminhada seria mais árdua sem vocês.

Por fim, agradeço aos meus mestres, pela formação que me foi proporcionada, em especial ao Prof. Jaidilson Jó da Silva, por todo apoio e orientação neste trabalho.

*“Arruma a cangalha na cacunda  
Que a rapadura é doce,  
Mas não é mole não.”*

João Alexandre.

## RESUMO

Este trabalho descreve o desenvolvimento de uma plataforma virtual de experimentos utilizando o software LabVIEW. Esta plataforma será integrada com uma bancada experimental já existente que simula um sistema de automação residencial a ser utilizada na disciplina Sistema de Aquisição de Dados e Interface, com o objetivo de auxiliar os alunos no aprendizado da disciplina, aliado a uma aplicação prática da utilização do microcontrolador ADuC842.

Ao longo do texto serão apresentadas algumas informações básicas sobre conceitos que são úteis para o melhor entendimento deste trabalho, bem como de todo o desenvolvimento do instrumento virtual desde o painel frontal até o diagrama de blocos. Por fim, serão apresentadas as conclusões e sugestões de implementações futuras no aparato experimental, e também, em anexo, os guias utilizados para cada conjunto de experimentos da bancada.

**Palavras-chave:** Aquisição de Dados, Instrumento Virtual, Automação Residencial, LabVIEW.

# ABSTRACT

This paper describes the development of a virtual platform of experiments using LabVIEW software. This platform will be integrated with an existing experimental bench that simulates a residential automation system to be used in the discipline Data and Interface Acquisition, with the objective of assisting students in the learning of the discipline, together with a practical application of the use of Microcontroller ADuC842.

Throughout the text will be presented some basic information about concepts that are useful for a better understanding of this work, as well as all the development of the virtual instrument from the front panel to the block diagram. Finally, the conclusions and suggestions of future implementations will be presented in the experimental apparatus, and also, in annex, the guides used for each set of bench experiments.

**Keywords:** Data Acquisition, Virtual Instrument, Residential Automation, LabVIEW.

# LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo da utilização da instrumentação virtual na automação residencial .....	14
Figura 2 - Esquema representativo de exemplo de um instrumento virtual.....	16
Figura 3 - Painel frontal.....	18
Figura 4 - Diagrama de blocos .....	19
Figura 5 - Microcontrolador ADuC842.....	20
Figura 6 - Bancada experimental utilizada .....	21
Figura 7 - Fotografia de um Fotodiodo .....	22
Figura 8 - Resistor Dependente de Luz(LDR).....	23
Figura 9 - Fotografia de um sensor LM35.....	24
Figura 10 - Painel frontal do experimento 1 .....	25
Figura 11 - Diagrama de blocos do experimento 1.....	26
Figura 12 - Parâmetros de configuração do bloco VISA.....	26
Figura 13 - Bloco VISA Read .....	27
Figura 14 - Diagrama de blocos da lógica implementada no experimento 1.....	27
Figura 15 - Painel frontal do experimento 2.....	28
Figura 16 - Diagrama de blocos para aquisição de dados via porta serial.....	29
Figura 17 - Diagrama de blocos da lógica implementada no experimento 2.....	30
Figura 18 - Painel frontal do experimento 3.....	31
Figura 19 - Diagrama de blocos da lógica implementada no experimento 3.....	31

# LISTA DE ABREVIATURAS E SIGLAS

LabVIEW	<i>Laboratory Virtual Instruments Engineering Workbench</i>
MIPS	Milhões de Instruções por Segundo
PWM	<i>Pulse Width Modulation</i>
LDR	<i>Light Dependent Resistor</i>

# SUMÁRIO

1	Introdução.....	12
1.1	Objetivos.....	12
1.2	Estrutura do trabalho.....	13
2	Fundamentação Teórica.....	14
2.1	Automação Residencial .....	14
2.2	Instrumentação Virtual.....	15
2.3	LabVIEW.....	17
2.4	Microcontrolador ADuC842.....	19
3	A plataforma.....	20
3.1	Bancada de Experimentos.....	21
3.1.1	Sensores utilizados .....	22
3.1.1.1	Fotodiodo .....	22
3.1.1.2	Resistor dependente de Luz - LDR .....	23
3.1.1.3	Sensor de Temperatura - LM35.....	24
3.2	Experimento 1 – Monitoramento de lâmpadas .....	24
3.3	Experimento 2 – Controle de temperatura .....	28
3.4	Experimento 3 – Visualização de Sinais PWM .....	30
4	Considerações Finais.....	33
	Referências .....	34
	ANEXO – Guias dos Experimentos .....	36

# 1 INTRODUÇÃO

A coleta de informações do mundo real de maneira a gerar dados que podem ser manipulados por um computador ou um sistema microprocessado é conhecida como aquisição de dados. Estes dados são geralmente analisados, monitorados e armazenados em um computador. Para isto, se faz necessário o desenvolvimento de uma plataforma conhecida por instrumento virtual.

Em um instrumento virtual, o operador tem acesso através de uma interface gráfica na tela do computador por meio de painéis, chaves, botões, etc., ao controle do instrumento conectado ao computador. Sendo assim, na instrumentação virtual, o computador é utilizado tanto para operar o instrumento quanto para conduzir o experimento conforme for conveniente. Nesta visão, a interface associada ao computador deve prover a mesma funcionalidade que a interface que um instrumento real apresentaria ao usuário. Ou seja, através da tela do computador, o usuário deve ser capaz de operar os instrumentos ou conduzir o experimento, tal qual faria usando os controles de um instrumento real.

A proposta deste trabalho é apresentar o desenvolvimento de uma plataforma virtual de experimentos utilizando o software LabVIEW, integrando a plataforma com uma bancada experimental já existente, que simula um sistema de automação residencial a ser utilizada na disciplina Sistema de Aquisição de Dados e Interface, com o objetivo de auxiliar os alunos no aprendizado da disciplina aliado a uma aplicação prática da utilização do microcontrolador ADuC842.

## 1.1 OBJETIVOS

O objetivo principal deste trabalho é o desenvolvimento de uma plataforma virtual de experimentos juntamente com seus respectivos guias para implementação na disciplina Sistema de Aquisição de Dados e Interface, com o intuito de auxiliar os estudantes no aprendizado do conteúdo desta, e demonstrar aplicações práticas para o uso do microcontrolador ADuC842 juntamente com a instrumentação virtual por meio do LabVIEW.

## 1.2 ESTRUTURA DO TRABALHO

Neste primeiro capítulo foi apresentada a introdução e os objetivos deste trabalho, no capítulo 2 foi apresentada uma breve revisão bibliográfica sobre automação industrial, Instrumentação Virtual, software LabVIEW, microcontrolador ADuC842 e sobre a bancada experimental, que visa destacar alguns pontos importantes ao entendimento do trabalho. Depois, no capítulo 3, foram descritos os procedimentos realizados no trabalho desenvolvido. E por fim, o trabalho é finalizado, no capítulo 4, com as considerações finais sobre o projeto e sugestões de implementações posteriores.

## 2 FUNDAMENTAÇÃO TEÓRICA

Estão descritas no decorrer deste tópico, resumidamente, algumas informações básicas sobre automação residencial, instrumentação virtual, ADuC842, software LabVIEW e sobre a bancada experimental utilizada que são úteis para o melhor entendimento deste trabalho.

### 2.1 AUTOMAÇÃO RESIDENCIAL

A automação residencial, também conhecida por domótica, inicialmente é referenciada como uma novidade que as vezes causa perplexidade pelo seu alto grau tecnológico e pela alusão ao futurismo ao mesmo tempo que pode ser compreendida como um símbolo de status e modernidade. Numa visão realista e austera, a automação residencial proporciona o conforto e a conveniência que qualquer ser humano deseja, talvez seja o maior e melhor dos sonhos de consumo almejados (TEZA, 2002).

O objetivo da automação residencial é proporcionar por meio de sistemas tecnológicos integrados, a melhor maneira de satisfazer as necessidades básicas de segurança, comunicação, gestão energética e conforto de uma habitação (JOSÉ et al., 2013). Um exemplo da utilização da instrumentação virtual na domótica é mostrado na figura 1.

Figura 1 - Exemplo da utilização da instrumentação virtual na automação residencial



Fonte: SANDIM, 2017.

Os sistemas de automação desenvolvidos atualmente priorizam o conforto de uma maneira simples e amigável para o usuário. Projetos básicos de automação iniciaram-se com soluções de segurança patrimonial, porém, com as exigências do mercado, estão sendo capazes de interagir com outras tecnologias, sempre tendo como ponto centralizador o processador de automação, para que o usuário tenha o controle de tudo através de centrais de controle como, por exemplo, instrumentos virtuais por meio de computadores (SGARBI, 2007; WORTMEYER et al., 2005).

Um dos problemas dos produtos de ponta lançados no mercado como *Home-Theaters*, DVD (Digital Video Disc), CFTV (Circuito Fechado de Televisão), entre outros, é a total falta de integração entre estes aparelhos. Temos em nossas casas uma quantidade absurda de controles remotos: muitas pessoas possuem aparelhos de ar condicionado, televisores, aparelhos de som, aparelhos de DVD, portões automáticos, ou seja, cada equipamento tem o seu próprio controle isolado. Neste sentido, a automação residencial veio para revolucionar este mercado, trazendo inovação e implantando nos dias atuais uma realidade que antes era vista como ficção científica (WORTMEYER et al., 2005).

## 2.2 INSTRUMENTAÇÃO VIRTUAL

A rápida disseminação dos computadores pessoais nos últimos anos catalisou uma revolução na instrumentação virtual para teste, medição e automação. A instrumentação virtual é fruto da quase onipresença dos computadores nos dias atuais, trazendo benefícios para engenheiros e pesquisadores que buscam aumento de produtividade, precisão e desempenho nas suas atividades (NATIONAL INSTRUMENTS, 2017).

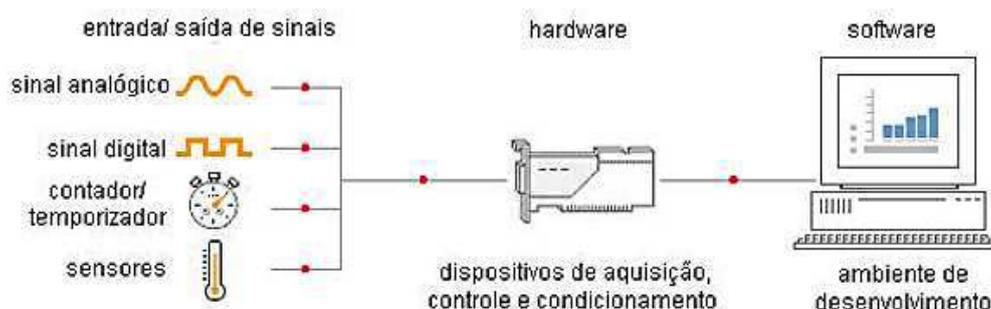
A instrumentação virtual consiste basicamente em um computador equipado com placas de medição e um software específico que juntos permitem desempenhar capacidade idêntica ou superior dos instrumentos tradicionais. A operação de um instrumento virtual envolve a utilização de interfaces gráficas especializadas que permitem uma comunicação fácil com o instrumento ao mesmo tempo que fornece uma representação adequada e flexível dos resultados da medição. A função de aquisição é realizada por hardware, mas o processamento e representação dos dados é feito por software (LOPES, 2007).

Um instrumento virtual é um dispositivo que mostra resultados gráficos a partir de um ambiente de dados, ou a partir de uma unidade em teste, com o objetivo de exibir informações para um usuário com base no tratamento destes dados. Este é um campo interdisciplinar que combina detecção, tecnologias de hardware e software, a fim de criar instrumentos flexíveis e sofisticados para aplicações de controle e monitoramento. O conceito de instrumentação virtual nasceu no final de 1970, quando a tecnologia de processamento de dados possibilitou a elaboração de sistemas de monitoramento de máquinas (ANDRADE; BALTAZAR, 2013).

Embora o computador pessoal e a tecnologia de circuito integrado tenham proporcionado avanços significantes nas últimas duas décadas, foi o desenvolvimento mais aprimorado dos softwares que realmente proporcionou que o computador fosse utilizado para criar instrumentos virtuais, oferecendo condições favoráveis para criar soluções inovadoras e reduzir significativamente os custos quando comparado com a abordagem tradicional. Com instrumentos virtuais, engenheiros e pesquisadores constroem sistemas de medição que se adequam exatamente as suas necessidades, são sistemas definidos pelo usuário, ao invés de estarem limitados pelas funções fixas dos instrumentos tradicionais, que são definidos pelo fabricante (NATIONAL INSTRUMENTS, 2017).

O conceito de instrumentação virtual consiste então em implementar instrumentos tradicionais utilizando o computador, e seus elementos básicos são: sensores (que convertem grandezas de diversos tipos em sinais elétricos), condicionadores de sinais (utilizados para adequar os sinais provenientes dos sensores as limitações do sistema de aquisição de dados), placa de aquisição de dados, computador e *softwares* de desenvolvimento (LOPES, 2007). A Figura 2, apresenta um esquema representativo da arquitetura de um instrumento virtual.

Figura 2 - Esquema representativo de exemplo de um instrumento virtual



Fonte: Catálogo da National Instruments 2006.

Segundo Lopes (2007), o uso de instrumentação virtual em laboratórios didáticos para o curso de engenharia elétrica é importante devido à redução de custos na aquisição de equipamentos para medições, como também no aumento da velocidade para a realização das práticas experimentais. A redução dos custos é possível pela utilização de hardware, havendo possibilidade de fazer simulação de uma ampla diversidade de equipamentos utilizados em laboratórios didáticos tais como o multímetro, osciloscópio, frequencímetro, wattímetro entre outros.

## 2.3 LABVIEW

A instrumentação virtual do presente trabalho é desenvolvida com o auxílio do software LabVIEW (*Laboratory Virtual Instruments Engineering Workbench*), que refere-se a um pacote de *software* desenvolvido pela National Instruments, para ser um ambiente de programação voltado ao desenvolvimento de aplicações, utilizando o conceito de instrumentação virtual. A linguagem de programação utilizada é do tipo visual e chamada de G (de Graphics). O LabVIEW é, então, uma linguagem de programação gráfica, que usa ícones em vez das linhas de comandos para criar suas aplicações. Contrasta com as linguagens de programação, baseadas em linha de comando, em que as instruções determinam a ordem de execução do programa. O LabVIEW usa o fluxo de dados dentro do programa, e este fluxo de dados por meio dos nós no diagrama de blocos determina a ordem de execução do instrumento virtual e das funções (LOPES, 2007).

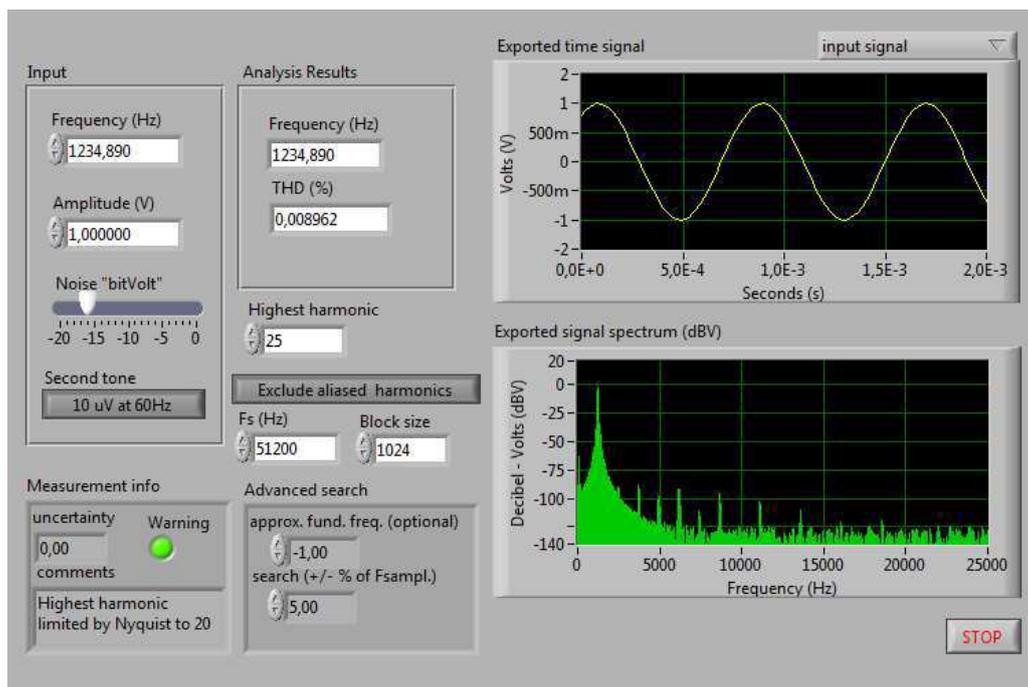
O LabVIEW é uma plataforma de programação extensa e inclui uma infinidade de funcionalidades que vão desde operadores algébricos básicos até componentes avançados de processamento de sinais que podem ser integrados em programas bastante sofisticados e complexos para uso em laboratórios e ambientes industriais. Ele utiliza uma linguagem de programação que permite configurar rapidamente a interface gráfica fazendo com que o utilizador adquira, visualize e registre sinais em simultâneo (MORRIS; LANGARI, 2016; SCOLASTICI, 2011).

Quando se trata de desenvolvimento de interfaces para o usuário, o LabVIEW é uma linguagem muito apropriada, pois interage com o programador por meio de duas telas separadas: o painel frontal e o diagrama de blocos. No painel frontal, os botões, os indicadores e os gráficos podem ser alcançados para controle direto do usuário. Já no

diagrama de blocos, o fluxo de dados e as funções de controle podem ser desenvolvidos através do código fonte gráfico que define a funcionalidade do instrumento virtual.

A parte do instrumento virtual que define a relação com o usuário é o painel frontal. A figura 3 mostra um exemplo de um painel frontal de um analisador de harmônicos desenvolvido pela National Instruments e disponível como exemplo no LabVIEW 11. A interface com o usuário é desenvolvida por meio de controles e indicadores, que são os terminais interativos de entrada e de saída do instrumento virtual, respectivamente. Os controles são botões, chaves seletoras, e outros mecanismos de entrada que simulam mecanismos de entrada no instrumento virtual, fornecendo dados ao diagrama de blocos. Os indicadores são os gráficos, LEDs, e os outros tipos de saída que simulam mostradores no instrumento virtual, apresentando os dados que o diagrama de blocos adquiriu ou gerou.

Figura 3 - Painel frontal

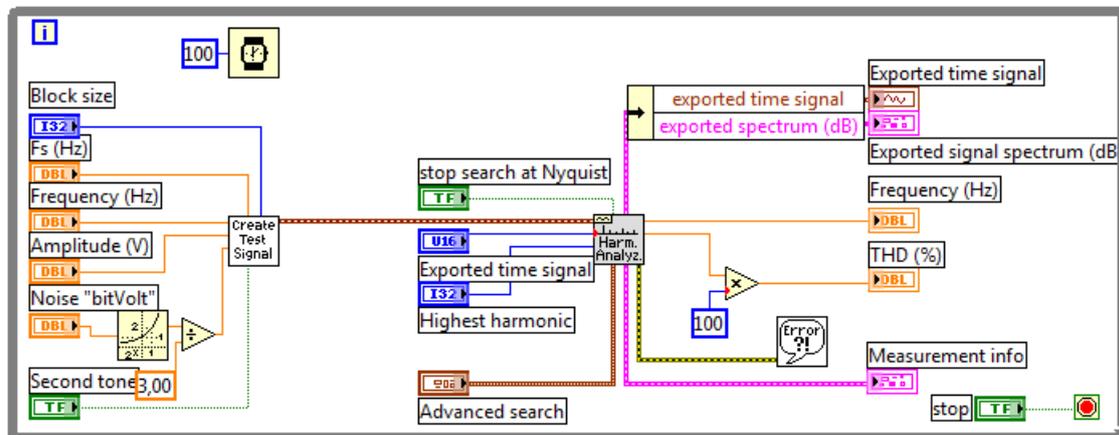


Fonte: Exemplo contido no programa LabVIEW.

Para controlar os objetos do painel frontal, é desenvolvido, através de representações gráficas de funções, o diagrama de blocos. As representações gráficas das funções podem ser operações matemáticas, lógicas ou estruturas de programação (*while loop, for loop, case structure, etc.*) e os objetos do painel frontal são os terminais presentes no diagrama de blocos dos respectivos controles e indicadores.

A figura 4 mostra o respectivo diagrama de blocos implementado com o LabVIEW, pela National Instruments, para desenvolver a lógica de funcionamento do analisador de harmônicos.

Figura 4 - Diagrama de blocos



Fonte: Exemplo contido no programa LabVIEW.

Uma curiosidade é que a designação, instrumentação virtual, foi introduzido em 1990 a partir de empresas que desenvolviam programas para aquisição de dados por meio do LabVIEW (RODRIGUES, 1999).

## 2.4 MICROCONTROLADOR ADuC842

O ADuC, desenvolvido pela Analog Devices, é uma série de microcontroladores destinada a sistemas de instrumentação e aquisição de dados. Foi concebido para aplicações em redes ópticas, sistemas com base remota, instrumentação de precisão e sensores inteligentes, entre outros. Seu núcleo é baseado na arquitetura do 8052 e é compatível com as instruções do 8051, sendo que mais recentemente a Analog Devices tem desenvolvido seus microcontroladores ADuC com um núcleo ARM7 para atingir taxas maiores de MIPS (Milhões de Instruções Por Segundo) e, conseqüentemente, aumentar o desempenho. A programação é feita em linguagem C utilizando-se o compilador KEIL, que é compatível e dispõe de recursos para programação, depuração e simulação não só para a família ADuC, mas para vários outros microcontroladores (COSTA FILHO et al., 2011).

O microcontrolador ADuC842 pode ser observado na figura 5. Ele possui como temporizador interno um cristal de 32 kHz, além de conversores analógico/digital e digital/analógico de até 12 *bits* de resolução. Possui 62 *kBytes* de memória de programa e 4 *kBytes* de memória de dados, com suporte a memória externa de acesso rápido. O ADuC possui também quatro portas, de 8 *bits* cada, onde algumas compartilham funcionalidades secundárias. Possui 12 sistemas de interrupções que podem assumir duas hierarquias de prioridade, alta ou baixa (Analog Devices, 2017).

Figura 5 - Microcontrolador ADuC842



Fonte: o próprio autor.

A partir dos devidos registradores do microcontrolador podem ser acessadas outras funcionalidades que podem ser implementadas digitalmente. Dentre estas estão um sistema contador de tempo, ou eventos; padrões de comunicação serial, sistema *Watchdog*, e saídas para sinal modulado por largura de pulso – do inglês, *pulse width modulation (PWM)* (Analog Devices, 2017).

### 3 A PLATAFORMA

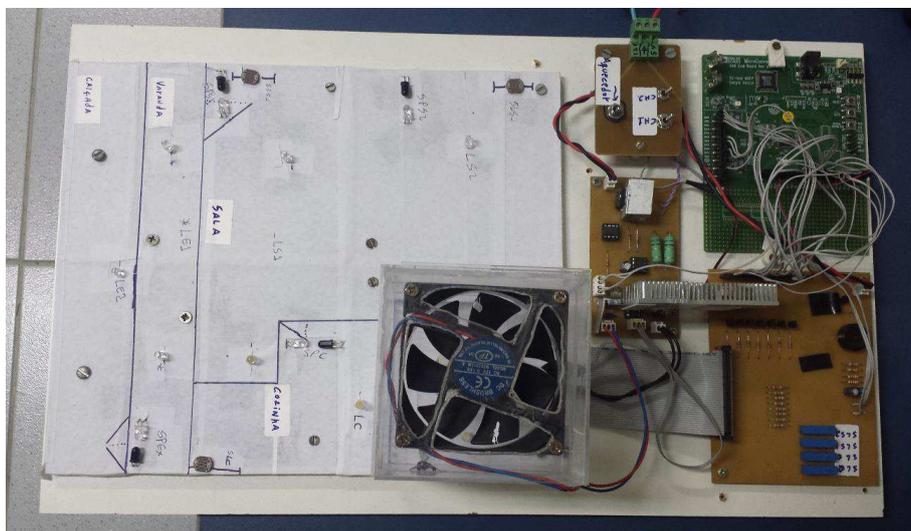
A fim de cumprir o objetivo deste trabalho foi desenvolvida uma plataforma virtual de experimentos que simula um sistema de domótica e utiliza o conceito de instrumento virtual visto anteriormente a partir da utilização do software LabVIEW.

Para isto, foi utilizada a bancada experimental já existente e foram desenvolvidos instrumentos virtuais para os três experimentos já existentes na bancada, mas que não contemplavam uma interface gráfica.

### 3.1 BANCADA DE EXPERIMENTOS

A bancada de experimentos utilizada neste trabalho para integração por meio da instrumentação virtual foi desenvolvida pelo aluno Pedro Merencio Primo Passos em seu estágio curricular obrigatório realizado no Laboratório de Instrumentação Eletrônica da UFCG. A bancada, mostrada na Figura 6, simula um sistema de automação residencial em que podem ser monitoradas e controladas eletronicamente características como iluminação, climatização, monitoramento de segurança e a abertura e fechamento de portas e janelas.

Figura 6 - Bancada experimental utilizada



Fonte: o próprio autor.

A maquete está dividida em quatro áreas que foram denominadas de calçada, varanda, quarto, sala e cozinha. Ela conta com sensores de intensidade luminosa, presença e temperatura para o monitoramento e atuadores como diodos emissores de luz, resistências aquecedoras, ventoinhas e alarme sonoro. Além da placa de desenvolvimento do ADuC842 a bancada possui duas placas *drivers* que fazem a adequação dos níveis de corrente e tensão dos componentes para a conexão com o microcontrolador.

Para o estudo foram desenvolvidos três experimentos com o objetivo de explorar em cada um certas funcionalidades do microcontrolador ADuC842. No primeiro, foram utilizadas as portas digitais de entrada e saída e a comunicação serial via protocolo RS-232. No segundo experimento, utilizou-se os conversores analógico/digital e digital/analógico. E, por fim, no terceiro experimento, tem-se a utilização da saída PWM do microcontrolador.

### 3.1.1 SENSORES UTILIZADOS

Sensores são designados como dispositivos sensíveis à algum tipo de energia do ambiente podendo ser luminosa, térmica, cinética, relacionando informações sobre uma grandeza física que precisa ser mensurada, como: temperatura, pressão, velocidade, corrente, aceleração, posição, etc (WENDLING,2010).

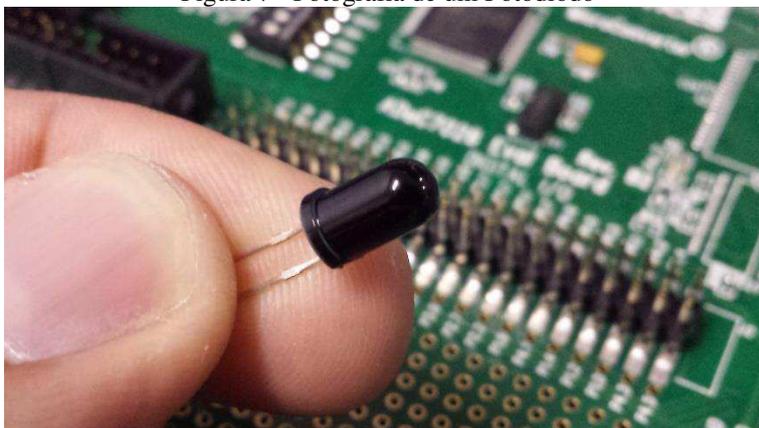
Os sensores quando operam diretamente, convertendo uma forma de energia em outra, são chamados de transdutores. Os de operação indireta alteram suas propriedades, como a resistência, a capacitância ou a indutância, sob ação de uma grandeza, de forma mais ou menos proporcional (Borges & Dores, 2010).

Os sensores que trabalham com luz são mais rápidos que os mecânicos, pois não apresentam inércia e não têm peças móveis que quebram ou desgastam. Os sensores fotoelétricos podem ser empregados numa infinidade de aplicações na indústria e em outros campos. Existem diversos dispositivos sensores que podem ser utilizados como sensores de luz, e sua escolha depende de suas características. São eles os fotoresistores, fotodiodos, fototransistores, entre outros (WENDLING,2010). Na bancada experimental, os principais sensores utilizados foram os fotodiodos, fotoresistores e o sensor de temperatura LM35.

#### 3.1.1.1 FOTODIODO

Os fotodiodos, apresentado na figura 7, são detectores de radiação nos quais o sinal elétrico é produzido pela geração de pares elétron-buraco, causada pela absorção de fótons na região próxima a camada de depleção, que está exposta para permitir a passagem de luz. Com a absorção, um par elétron-buraco é formado e cada portador é acelerado em um sentido diferente pelo campo elétrico da junção (ASSUNÇÃO, 2012).

Figura 7 - Fotografia de um Fotodiodo



Fonte: o próprio autor.

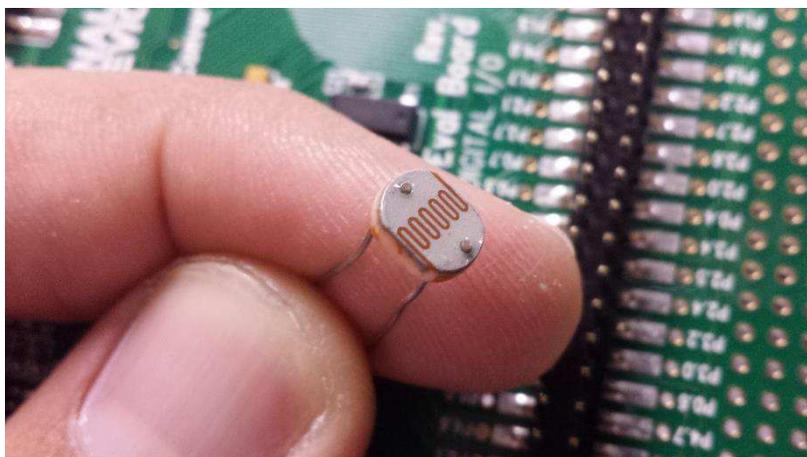
Existem dois modos de operação dos fotodiodos: fotovoltaico e fotocondutivo. No fotocondutivo uma tensão externa é aplicada e, nesse caso, quando o dispositivo é iluminado, uma corrente flui no sentido oposto à corrente aplicada. Na operação fotovoltaica, o circuito está aberto e a iluminação da junção faz surgir a tensão que gerará a corrente elétrica (ASSUNÇÃO, 2012).

### 3.1.1.2 RESISTOR DEPENDENTE DE LUZ - LDR

O LDR (Light Dependent Resistor) ou simplesmente fotoresistor, é um componente eletrônico constituído por Sulfeto de Cádmio - CdS, substância semicondutora muito sensível à luz. Este resistor possui característica sólida, apresentando cor amarelo avermelhado (THOMAZINI e ALBUQUERQUE,2012).

Conforme pode ser observado na figura 8 a seguir, ele é constituído de dois terminais. Sua resistência varia linearmente com a luz ou radiação incidente porém, de forma inversamente proporcional e sua equação característica sempre depende do material utilizado e do processo de fabricação (VASCONCELOS, 2014).

Figura 8 - Resistor Dependente de Luz(LDR)



Fonte: o próprio autor.

Quando a luz incide em determinadas substâncias cujas resistências são alteradas devido à quantidade de luz que recebem, ocorre a liberação de portadores de carga que ajudam a condução da corrente elétrica (THOMAZINI e ALBUQUERQUE,2012).

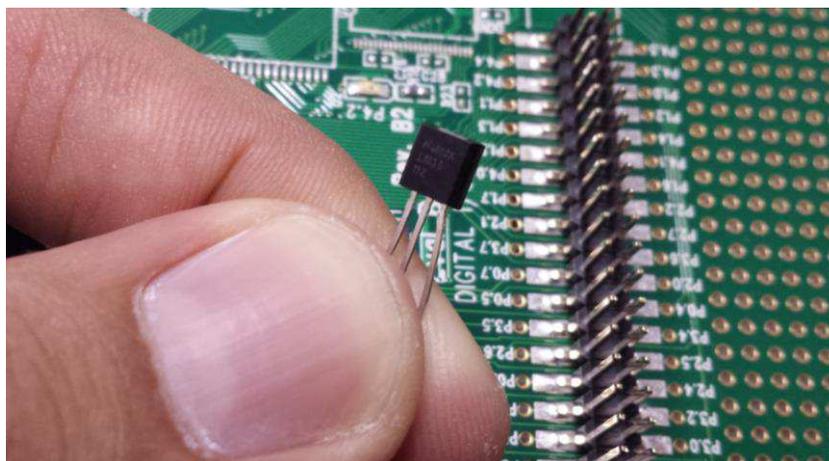
O LDR possui, como material semicondutor, uma região de valência e uma região de condução. A radiação incidente ou luz desloca mais elétrons da camada de valência para a camada de condução quanto maior for a sua intensidade ao passo que a resistência

do LDR diminui e a energia aumenta proporcionalmente com o aumento da quantidade de elétrons liberados e migrados (VASCONCELOS, 2014).

### 3.1.1.3 SENSOR DE TEMPERATURA - LM35

O sensor de temperatura LM35 é um dispositivo de 3 terminais (um de alimentação, um para a tensão de saída e um terminal neutro) que fornece uma tensão de saída diretamente proporcional a temperatura. A tensão de saída é de  $10 \text{ mV}/^\circ\text{C}$ , referenciado a partir de  $0^\circ\text{C}$ . A exatidão do dispositivo é de  $0,5^\circ\text{C}$  para a faixa de  $0$  a  $70^\circ\text{C}$ . O LM35 pode ser alimentado com uma tensão que pode variar entre 5 Volts e 30 Volts(CAMARGO,2008). A Figura 9, apresenta a fotografia de um sensor LM35.

Figura 9 - Fotografia de um sensor LM35



Fonte: o próprio autor.

O LM35 é utilizado amplamente em pesquisas por apresentar baixo custo, precisão e linearidade, podendo ser empregado em diversas atividades onde é necessário o uso de termômetros para o controle de temperatura, como em câmeras frias e/ou ambientes controlados para criação de animais, controle de temperatura de máquinas e aquisição de dados para pesquisas (VOLTAN, 2012).

## 3.2 EXPERIMENTO 1 – MONITORAMENTO DE LÂMPADAS

O experimento de número um da bancada experimental tem por objetivo realizar um primeiro contato com o microcontrolador, bem como entender, configurar e testar a

comunicação serial e os pinos de entrada e saída digital. O experimento consiste em acender ou apagar as lâmpadas de uma casa a partir da entrada do morador em cada cômodo, sempre verificando a necessidade do acendimento ou não das lâmpadas, a partir de sensores instalados próximos as janelas que mensuram se a luminosidade externa vinda por meio dela é suficiente ou não para iluminar o interior do cômodo. Entendido o experimento da bancada, foi desenvolvido o experimento virtual com o auxílio do LabVIEW.

A ideia principal deste experimento foi monitorar o acendimento e apagamento das lâmpadas afim de mostrar no instrumento virtual quais lâmpadas estavam acesas e apagadas no momento. Existem várias ideias que podem ser implementadas juntamente ao monitoramento, como por exemplo o cálculo do custo monetário mensal pela utilização de cada lâmpada ou ainda o acendimento e apagamento da lâmpada remotamente. Porém, este trabalho limitou-se somente ao monitoramento do estado das lâmpadas indicando o seu estado visualmente no instrumento virtual.

Com o auxílio do AutoCAD, foi desenhada a planta baixa de uma casa que corresponde ao layout da bancada afim de ser usada como plano de fundo da plataforma em todos os instrumentos virtuais criados. Na figura 10 a seguir, é apresentado o painel frontal do instrumento virtual desenvolvido que possui luzes indicando o estado atual das lâmpadas da bancada.

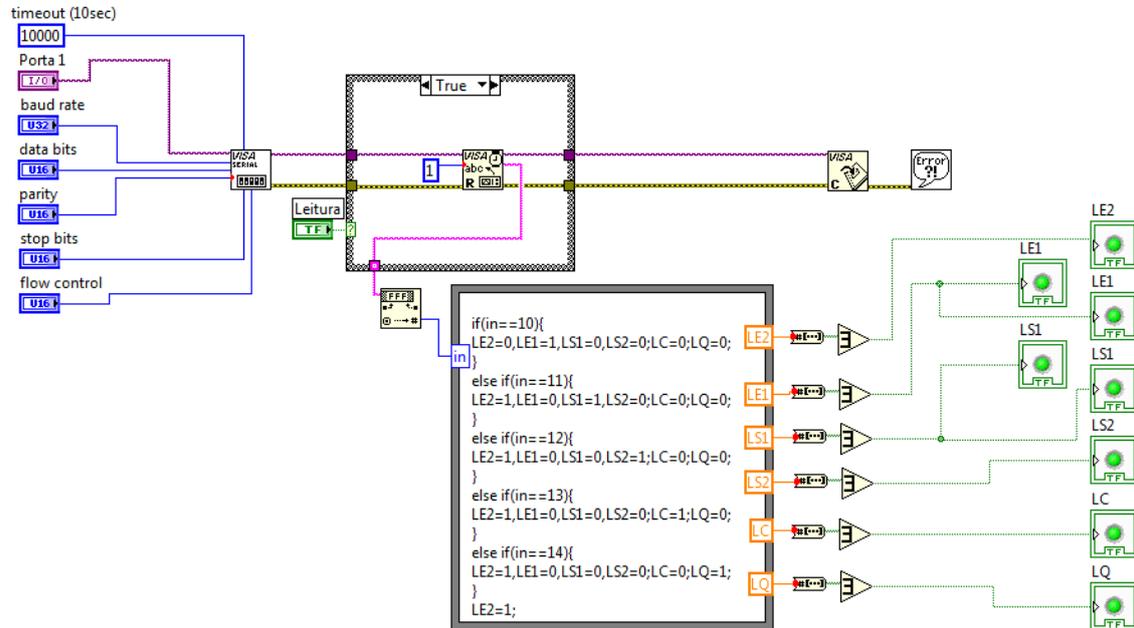
Figura 10 - Painel frontal do experimento 1



Fonte: o próprio autor.

Para o correto funcionamento do painel frontal, foi criado o diagrama de blocos que pode ser visualizado na figura 11 a seguir.

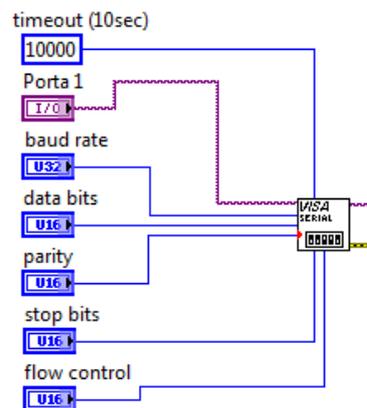
Figura 11 - Diagrama de blocos do experimento 1



Fonte: o próprio autor.

No diagrama de blocos, a recepção serial dos dados vinda do ADuC842 é feita pelo bloco VISA. O bloco VISA *Configure Serial Port* prepara a porta serial para comunicação. Neste bloco são configurados parâmetros como a porta serial a ser utilizada, baud rate, e o número de bits dos registradores. Conforme mostrado na figura 12.

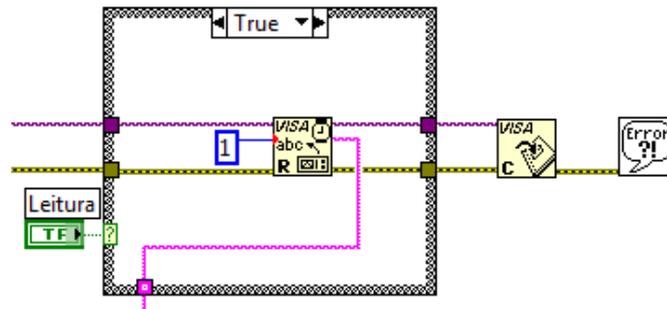
Figura 12 - Parâmetros de configuração do bloco VISA



Fonte: o próprio autor.

Para a realização das leituras, foi utilizado o bloco VISA Read apresentado na figura 13. Na entrada deste bloco deve ser indicado a quantidade de bytes por quadro recebido. Para o experimento em questão, apenas um byte por quadro é suficiente para o envio de informações. Na saída do bloco, tem-se uma string com os dados lidos pela porta serial.

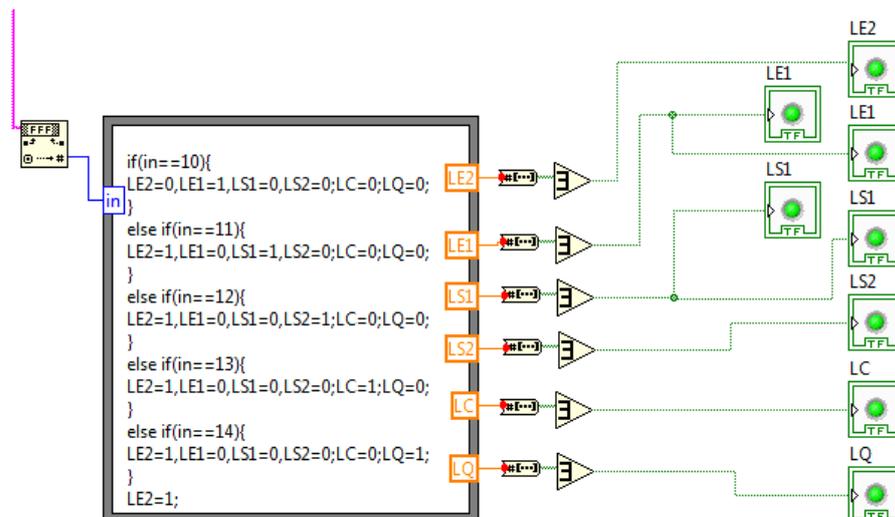
Figura 13 - Bloco VISA Read



Fonte: o próprio autor.

A string com os dados lidos pela porta serial são tratados, afim de que seja concluído o objetivo principal do experimento. Para tanto, necessitou-se utilizar um bloco de conversão de string para um numérico. Após isto, utilizou-se um bloco denominado Formula Node para que fosse definida uma ação para cada possibilidade de leitura. Desta forma, foram utilizados comparadores e nas saídas do bloco Formula Node portas lógicas OR para que o comando de acender ou apagar o led fosse feito no sistema binário.

Figura 14 - Diagrama de blocos da lógica implementada no experimento 1



Fonte: o próprio autor.

Desta forma, para cada estado lido na porta serial, corresponde um estado de acendimento ou apagamento dos led's.

### 3.3 EXPERIMENTO 2 – CONTROLE DE TEMPERATURA

O experimento dois da bancada experimental visa demonstrar a conversão Analógico/Digital (A/D) e Digital/Analógico (D/A) entendendo conceitos como resolução e amostragem. O experimento consiste em controlar a temperatura do cômodo quarto a partir do sensor LM35 deixando-o abaixo de uma temperatura máxima previamente estabelecida. Para tanto, a bancada dispõe de resistências a fio que simulam uma perturbação externa e uma ventoinha para o resfriamento do cômodo quando o mesmo atinge determinada temperatura.

Compreendido o experimento da bancada experimental, desenvolveu-se o experimento virtual para monitoramento da temperatura do cômodo. O objetivo principal foi observar em um termômetro virtual a que temperatura em graus Celsius o cômodo estava exposto e em que momento foi necessário o acionamento da ventoinha para o resfriamento do cômodo representada no instrumento virtual por meio do led ventilador, como pode ser visto na figura 15 a seguir.

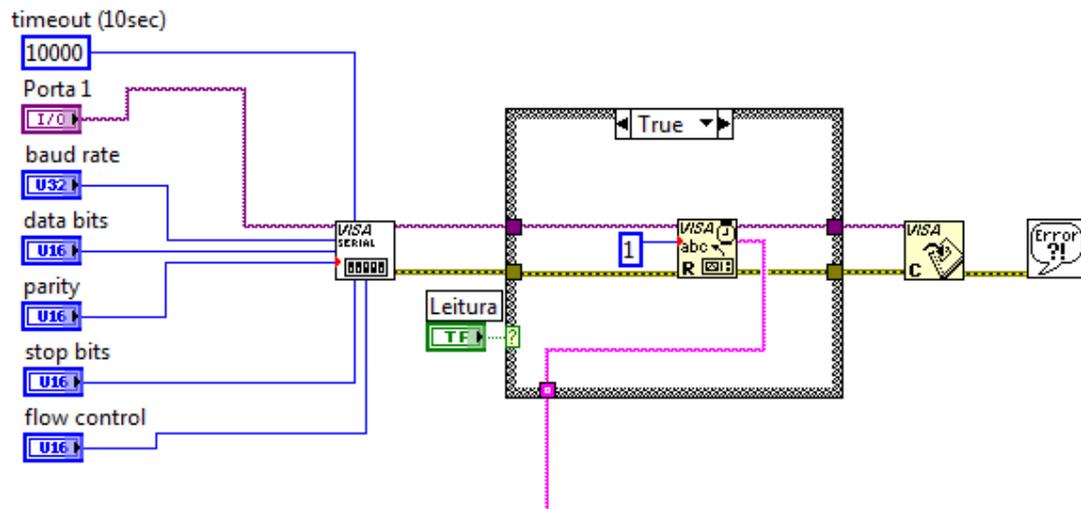
Figura 15 - Painel frontal do experimento 2



Fonte: o próprio autor.

Da mesma forma que no experimento anterior, utilizou-se a planta baixa no painel frontal e os blocos *VISA Configure Serial Port* e *VISA Read* no diagrama de blocos afim de receber os dados enviados pelo ADuC842 pela porta serial.

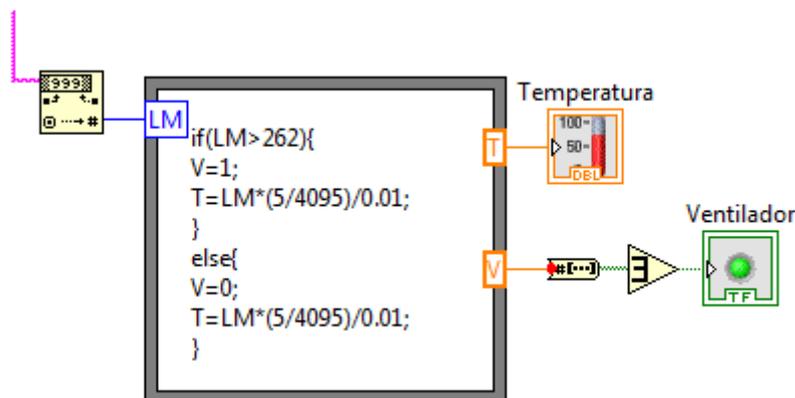
Figura 16 - Diagrama de blocos para aquisição de dados via porta serial



Fonte: o próprio autor.

O sinal recebido da porta serial corresponde à leitura no canal A/D vinda do sensor de temperatura LM35. Como mostrado na figura 17, foi feita a conversão para que fosse obtida a temperatura do cômodo. Para isto, levou-se em consideração que a resolução do conversor é de 12 bits, e que a tensão de referência é de 5 volts. Logo, multiplicando-se o valor lido pela tensão de referência e dividindo-se pela resolução, obtemos o valor de tensão da saída do LM35. Como neste sensor cada grau °C corresponde a 10mV de tensão, dividimos a tensão lida por 0,01. Obtendo assim a temperatura do cômodo.

Figura 17 - Diagrama de blocos da lógica implementada no experimento 2



Fonte: o próprio autor.

Quando a leitura no canal passa de 262, significa que a temperatura está maior que 32°C e portanto o ventilador está ligado. Assim, foi implementado no diagrama de blocos uma condição que faz com que um led seja aceso indicando o acionamento do ventilador.

### 3.4 EXPERIMENTO 3 – VISUALIZAÇÃO DE SINAIS PWM

O experimento de número três da bancada experimental trata da geração de sinais PWM pelo microcontrolador ADuC842. Para este experimento, os sinais foram gerados a partir do *clock* do Oscilador Controlado por Tensão do próprio microcontrolador que é de 16.77MHz. Com o auxílio de duas chaves presentes na placa 1 da bancada experimental, o usuário pode alterar a frequência para 10 ou 20Hz como também o *duty cycle* para 25 ou 75%. Este sinal é colocado na entrada de um *buzzer* e pode-se então observar a diferença para cada uma das quatro configurações possíveis.

O estudo da geração de sinais PWM é indispensável para um conhecimento mais aprofundado do microcontrolador ADuC842. Porém, não encontrou-se uma aplicação para esta função no domínio da domótica. Sendo assim, foi desenvolvido com o auxílio do software LabVIEW um instrumento virtual que contém no seu painel frontal um gráfico mostrando o sinal PWM gerado pelo microcontrolador. Este painel frontal pode ser observado na figura 18 a seguir.

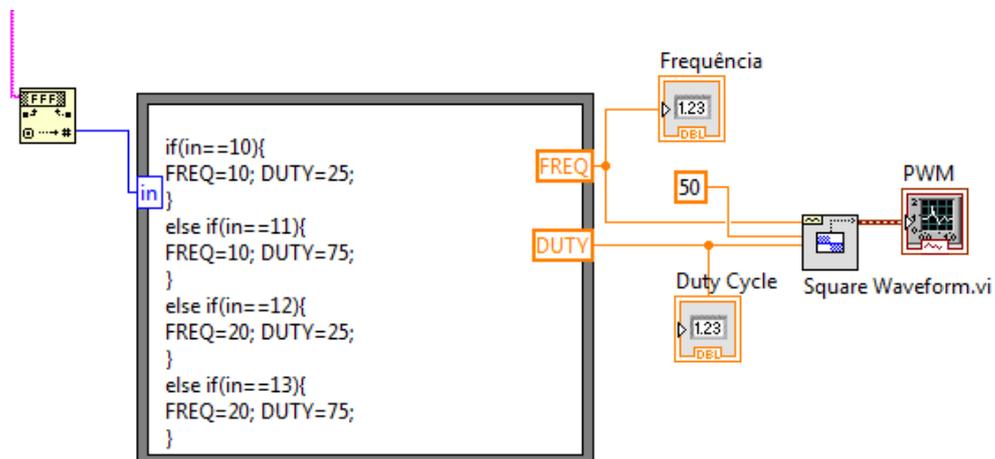
Figura 18 - Painel frontal do experimento 3



Fonte: o próprio autor.

Para o diagrama de blocos, utilizou-se da mesma forma que os exemplos anteriores para obtenção dos dados enviados pelo ADuC842 a partir da porta serial. O sinal recebido da porta serial segue basicamente a mesma lógica do experimento 1. Para cada estado em que as chaves se encontram envia-se uma string que após conversão para número é enviada para um bloco chamado de Formula Node para que seja definida uma ação para cada estado das chaves. Com o objetivo de representar graficamente o sinal PWM, utilizou-se o bloco Square Waveform o qual recebe na entrada os dados de frequência, duty cycle e amplitude e gera na saída um sinal que foi plotado no gráfico denominado PWM.

Figura 19 - Diagrama de blocos da lógica implementada no experimento 3



Fonte: o próprio autor.

Desta forma, conseguiu-se o objetivo do experimento que era representar graficamente o sinal PWM gerado de acordo com o posicionamento das chaves.

## 4 CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvida uma plataforma virtual de experimentos com a utilização do software LabVIEW. Esta plataforma foi integrada com uma bancada experimental já existente no laboratório e que simula um sistema de automação residencial.

A partir da construção dos experimentos, concluiu-se que os alunos terão uma visão mais ampla da instrumentação virtual e da sua integração com sistemas de aquisição de dados. Em consequência, apresentarão maior facilidade no aprendizado do software LabVIEW, bem como do microcontrolador ADuC842 uma vez que os experimentos tratam dos principais conhecimentos básicos destas ferramentas, fazendo com que abra-se um leque de possibilidades para que os estudantes criem novos projetos e se aprofundem no conhecimento da disciplina.

Mesmo que os experimentos já estejam prontos, fica aberta a possibilidade dos estudantes criarem experimentos novos e melhorarem os já existentes uma vez que a área da automação residencial é muito ampla. Sugere-se também implementações futuras para os experimentos, como por exemplo a integração do experimento 3 com um alarme sonoro do sistema de automação, ou a implementação de um visor que mostra o custo monetário mensal pela utilização de cada lâmpada ou ainda o acendimento e apagamento das lâmpadas remotamente no experimento 1.

## REFERÊNCIAS

ANALOG DEVICES. **MicroConverter®, Small Package 12-Bit ADC with Embedded Flash MCU-ADuC814**. Disponível em: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADUC814.pdf>. Acesso em: 10 de Março de 2017.

ANDRADE, D. C. D. G. F. DE; BALTAZAR, P. M. **Estudo e simulação de controle de torque em sistemas servoacionados**. 2013.

ASSUNÇÃO, M. DE O. Efeitos térmicos em fotodiodos de pontos quânticos semicondutores. **Dissertação Programa de Pós-Graduação em Física da Universidade Federal de Uberlândia**. 2012.

BORGES, L. P. e DORES, R. de C. Automação Predial Sem Fio Utilizando Bacnet/Zigbee com Foco em Economia de Energia. **Trabalho de Graduação**. Universidade de Brasília. Faculdade de Tecnologia Curso de Graduação em Engenharia de Controle e Automação, 2010.

CAMARGO, V. L. A. de. Desenvolvimento e Implementação de uma Plataforma para Monitoramento Estrutural Utilizando Sensores Extensométricos Conectados em Rede. Londrina. **Dissertação submetida ao Departamento de Engenharia Elétrica da Universidade Estadual de Londrina**. 2008.

COSTA FILHO, A. A. DA; NETO, J. M. R. DE S.; ROCHA NETO, J. S. SISTEMA DE AQUISIÇÃO E TRATAMENTO DE DADOS USANDO O. **XXXIX Congresso Brasileiro de Educação em Engenharia. Blumenau-SC.**, 2011.

JOSÉ, P.; MURATORI, R.; DAL, H. Capítulo I Automação residencial : histórico, definições e conceitos. p. 70–77, 2013.

LOPES, V. J. S. Instrumentação Virtual Aplicada ao Ensino Experimental de Engenharia Elétrica. **Dissertação para obtenção do título de Mestre em Engenharia Elétrica**, 2007.

MORRIS, A. S.; LANGARI, R. Data Acquisition with LabVIEW. In: **Data Acquisition with LabVIEW**.p. 347–374.

NATIONAL INSTRUMENTS. **Instrumentação Virtual**. Disponível em [www.ni.com/white-paper/4752/pt](http://www.ni.com/white-paper/4752/pt). Acesso em: 15 de fevereiro de 2017

RODRIGUES, C. R. **A Instrumentação Virtual como forma de Integração Entre a Teoria e Prática no Ensino de Medidas Elétricas**. XXIX Congresso Brasileiro de Ensino de Engenharia.1999.

SCOLASTICI, É. F. **Utilização da Instrumentação Virtual em LABVIEW para a Criação da Interface Gráfica do Amplificador Lock-in Utilizado em Estudos Fisiológicos a Partir da Comunicação via Porta Serial.** 2011.

SGARBI, J. A. **DOMÓTICA INTELIGENTE: AUTOMAÇÃO RESIDENCIAL BASEADA EM COMPORTAMENTO. Dissertação de Mestrado em Engenharia Elétrica**, v. CENTRO UNI, n. São Bernardo do Campo, p. 107, 2007.

TEZA, V. R. **ALGUNS ASPECTOS SOBRE A AUTOMAÇÃO RESIDENCIAL - DOMÓTICA. Dissertação de mestrado em Ciência da Computação.** Universidade Federal de Santa Catarina. 2002.

THOMAZINI, D. e ALBUQUERQUE, P. U. B. **Sensores Industriais – Fundamentos e Aplicações.** 5ª Ed. São Paulo: Érica, 2012.

VASCONCELOS, E. R. DE S. **Detector Para Raios X Usando Sensor Ldr Para Mensurar A Energia dos Fótons Emitidos e Testar a Eficiência De Aparelhos Hospitalares. Dissertação de Mestrado. Programa De Pós-Graduação Em Engenharia Biomédica.** Unb - Universidade De Brasília, 2014.

VOLTAN, D. S. **Desenvolvimento de um Sistema Computacional para Monitoramento de Parâmetros Físicos na Área de Irrigação, com Base na Plataforma Labview. Dissertação de Mestrado.** Universidade Estadual Paulista “Julio De Mesquita Filho” Campus De Botucatu, 2012.

WENDLING, M. **Sensores.** Universidade Estadual Paulista - UNESP. Colégio Técnico Industrial de Guaratinguetá “Professor Carlos Augusto Patrício Amorim”. 2010. Disponível em: [http://www2.feg.unesp.br/Home/Paginas\\_Pessoais/Prof\\_MaceloWendling/4---sensores-v2.0.pdf](http://www2.feg.unesp.br/Home/Paginas_Pessoais/Prof_MaceloWendling/4---sensores-v2.0.pdf). Acesso em: 20 de Fevereiro de 2017.

WORTMEYER, C.; FREITAS, F.; CARDOSO, L.; EDUCACIONAL, A.; BOSCO, D. **Automação Residencial : Busca de Tecnologias visando o Conforto , a Economia , a Praticidade e a Segurança do Usuário.** II Simpósio de Excelência em Gestão e Tecnologia. p. 1064–1067, 2005.

# ANEXO – GUIAS DOS EXPERIMENTOS

## Guia do Experimento 1

### Comunicação serial com ADuC842 e Portas I/O

#### Objetivos

Realizar um primeiro contato com o microcontrolador, bem como entender, configurar e testar a comunicação serial e os pinos de entrada e saída digital.

#### Comunicação serial

O microcontrolador ADuC842 possui 3 protocolos de comunicação serial implementados, o protocolo I2C, o protocolo SPI e o mais conhecido e simples de implementação, o protocolo RS232. O bloco de comunicação RS232 é *full-duplex*, o que significa que o sistema pode transmitir e receber simultaneamente. A interface física para rede de dados serial é via os pinos RxD (P3.0) e TxD (P3.1), enquanto que a interface de registradores é composta pelos registradores SBUF e SCON como descrito na Tabela 1 em seguida.

**Tabela 1: SCON**

Nº do bit	Nome	Descrição		
7	SM0	Bits de seleção para o modo de comunicação UART		
6	SM1	SM0	SM1	Seleção do modo de operação
		0	0	Modo 0: Baud rate fixo ( $core\_clk/2$ )
		0	1	Modo 1: UART de 8 bits, Baud rate variável
		1	0	Modo 2: UART de 9 bits, Baud rate fixo ( $core\_clk/32$ ) ou ( $core\_clk/16$ )
		1	1	Modo 3: UART de 9 bits, Baud rate variável
5	SM2	Comunicação em multiprocesso usado nos modos 2 e 3		
4	REN	Setado por software habilita a recepção de dados pela serial		
3	TB8	Usado nos modos 2 e 3 de comunicação para enviar o bit 9		
2	RB8	Usado nos modos 2 e 3 de comunicação para receber o bit 9		
1	TI	<i>flag</i> de interrupção para transmissão serial		
0	RI	<i>flag</i> de interrupção para recepção serial		

#### SBUF

Ambos os registradores de recepção e transmissão são acessados através do registrador SBUF. Escrever nesse registrador qualquer valor significa que este valor estará disponível no barramento de dados da comunicação serial.

#### Definindo Baud Rate

Toda comunicação serial necessita de uma base de tempo para definição de um *baud rate*, no ADuC842 essa base de tempo é gerada pelo uso de um dos três *timers* presentes no microcontrolador. O modo mais simples de gerar uma base de tempo para

essa comunicação é usando o *timer 3*, esse modo é descrito de forma mais detalhada no *datasheet* do ADuC842, mas pode ser resumido da seguinte forma:

Dois registradores são usados para o controle do *timer 3* (T3CON e T3FD). O T3CON é o registrador de controle do *baud rate*, permitindo que o *timer 3* seja usado para gerar a base de tempo para a comunicação serial. O T3FD é a razão de divisão fracionária necessária para obter-se o valor de *baud rate* desejado.

A Tabela 2 com o resumo dos valores mais frequentes de *baud rate*, e os valores necessários a serem armazenados nos registradores, pode ser observada a seguir.

**Tabela 2:** Valores Para T3CON e T3FD e os *baud rates* selecionados.  
Divisão do *clock* do PLL = 0, ou seja, *clock* de 16.77 MHz.

BaudRate->115200	BaudRate -> 57600	BaudRate -> 38400	BaudRate -> 19200	BaudRate -> 9600
T3CON = 0x83;	T3CON = 0x84;	T3CON = 0x84;	T3CON = 0x85;	T3CON = 0x86;
T3FD = 0x09;	T3FD = 0x09;	T3FD = 0x2d;	T3FD = 0x2d;	T3FD = 0x2d;

### Trabalhando com os pinos de entrada/saída

Os pinos de entrada/saída do microcontrolador estão associados a quatro portas (P0, P1, P2, P3). Suas características estão descritas no *datasheet* do ADuC842. As portas P0 e P1 não possuem resistores de *pull up* na sua configuração interna. Tais resistores são importantes para limitar a máxima corrente fornecida pelo pino, de modo que estes trabalham com níveis muito baixos de corrente, tanto fornecidos como drenados.

Os pinos da porta P0 que tem '1' escrito neles ficam com nível flutuante e, portanto, podem ser usados como entrada digital. Quando usado como saída digital, faz-se necessário adicionar resistores de *pull up* para limitar a corrente em 1.6 mA. Esta porta é utilizada quando necessário usar memória externa.

A porta P1 é somente de entrada, digital ou analógica. Por padrão, os pinos desta porta funcionam como entradas analógicas. Para serem usados como entradas digitais é necessário, antes, escrever '0' nos bits de cada pino. Alguns pinos dessa porta possuem função secundária, vide *datasheet*.

Já a porta P2 do microcontrolador pode ser usada como entrada, ou saída digital. Esta porta possui resistores de *pull up* internos. Ela é utilizada junto com a porta P0 no acesso à memória externa. Para ser usada como entrada digital é necessário, antes, escrever '1' no bit do pino desejado. Os pinos P2.6 e P2.7 também possuem a funcionalidade do bloco PWM do microcontrolador. Esta função é selecionada via registrador CFG842.

Por fim, a porta P3 é a que demanda maior atenção quando utilizada, pois todos os seus pinos possuem funções secundárias, como descritas no *datasheet*. Por exemplo, quando da comunicação serial RS-232, os pinos P3.0 e P3.1 são utilizados, independentemente das configurações prévias feitas no código. Seus pinos também possuem resistores internos e para serem usados como entradas digitais precisam ter '1' escrito previamente em seus bits.

Para trabalhar com os pinos individuais de cada porta, é necessário definir seus bits de registro devidamente endereçados. Os pinos da porta P0 vão do endereço 80h até o 87h. Na porta P1 os endereços são de 90h até 97h. Já a porta P2 vai de A0h até A7h. E, por fim, os endereços da porta P3 vão de B0h até B7h.

No código exemplo, ao final deste documento, está demonstrado como realizar esses endereçamentos e como essas variáveis podem ser manipuladas.

### Bancada de experimento

Seguindo o propósito de simulação de um sistema de automação residencial da planta, neste experimento será realizada a simulação do controle de iluminação dos ambientes da casa.

Na planta residencial simulada, as posições dos sensores de presença remetem às entradas de cada cômodo, e a dos sensores de luz, por sua vez, remetem as janelas de cada um. Desta forma, as luzes de cada ambiente são acionadas quando alguém entrar, se não houver iluminação suficiente entrando pela janela. Exceto para o quarto, onde as luzes são acionadas por interruptores, simulados pelas chaves. Neste, há 4 níveis de iluminação: tudo apagado, 1 ponto de luz, 2 pontos de luz, ou tudo aceso. A luz que entra pela janela tida como uma fonte de luz. O controlador então decide, através das posições das chaves, quantas luzes é necessário acionar.

Para este experimento, será usada a placa 01, que deve ser alimentada por uma fonte de tensão contínua de 5 V. Bem como as chaves CH1 e CH2, os quatro LDRs, os quatro foto-transistores e os sete conjuntos de LEDs. Na Tabela 3 seguinte há a indicação sobre em qual porta estará conectado cada dispositivo.

**Tabela 3:** Mapeamento dos dispositivos eletrônicos e as portas do ADuC842.

Bit \ Porta	P0	P1	P2	P3
0	SPS2	SLQ	LS2	
1	SPC	SLC	LQ1	
2	SPS1	SLS1	LQ2	SPE <sub>x</sub>
3		SLS2	LC	
4			LE2	
5			LE1	
6	CH2		LS1	
7	CH1			

Os quatro potenciômetros presentes na placa 01 servem para ajustar o ponto de operação dos LDRs e, assim, adequar-se ao nível de iluminação no ambiente no qual o exercício é realizado.

### Plataforma Virtual

Este experimento virtual, monitora o acendimento e apagamento das lâmpadas da residência simulada na bancada experimental. No computador é possível observar, com o auxílio do *software* LabVIEW, o comportamento das luminárias dependendo da iluminação ambiente e do acionamento dos sensores de presença quando da passagem de alguém pela entrada dos cômodos.

## Programa exemplo

```
// UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG
// DEPARTAMENTO DE ENGENHARIA ELETRICA - DEE
// LABORATORIO DE INSTRUMENTAÇÃO E CONTROLE - LIEC
//
// DISC: SISTEMAS DE AQUISIÇÃO DE DADOS E INTERFACE
// PROF: JAIDILSON JÓ DA SILVA
//
//
// Experimento 1: Portas I/O & Comunicação Serial

//-----//
// Cabeçalhos

#include "aduc842.h" // Cabeçalho do ADuC842
#include <stdio.h> // Cabeçalho de Padrão de I/O
#include <stdlib.h>
#include <intrins.h>

//-----//
// Variáveis //
char entrada=0;

// Saídas
sbit LQ1 = 0xA2; // P2.2 -> Luz quarto 1
sbit LQ2 = 0xA1; // P2.1 -> Luz quarto 2

sbit LS1 = 0xA6; // P2.6 -> Luz sala 1
sbit LS2 = 0xA0; // P2.0 -> Luz sala 2

sbit LC = 0xA3; // P2.3 -> Luz cozinha

sbit LE1 = 0xA5; // P2.5 -> Luz externa 1
sbit LE2 = 0xA4; // P2.4 -> Luz externa 2

// Entradas
sbit CH1 = 0x87; // P0.7 -> Chave 1
sbit CH2 = 0x86; // P0.6 -> Chave 2

sbit SPS1 = 0x82; // P0.2 -> Sensor Presença - sala 1
sbit SPS2 = 0x80; // P0.0 -> Sensor Presença - sala 2

sbit SPC = 0x81; // P0.1 -> Sensor Presença - cozinha

sbit SPEX = 0xB2; // P3.2 -> Sensor Presença externa

sbit SLS1 = 0x92; // P1.2 -> Sensor Luz - sala 1
sbit SLS2 = 0x93; // P1.3 -> Sensor Luz - sala 2

sbit SLC = 0x91; // P1.1 -> Sensor Luz - cozinha

sbit SLQ = 0x90; // P1.0 -> Sensor Luz - Quarto

//-----//
// Prototipos de Funções //
void configura_serial (void); // Configuração da porta serial do uC

void delayls (); // Função para atraso de 1s
//-----//
// Programa Principal //
void main (void)
{
    PLLCON = 0x88; // Clock de 16.77 MHz

    configura_serial(); // configura a porta serial para Baud Rate de 9600 bps

    // Configurando bits das Portas
    SPEX = 1;

    CH1 = 1;
    CH2 = 1;
}
```

```

SPS1 = 1;
SPS2 = 1;
SPC = 1;

SLQ = 0;
SLC = 0;
SLS1 = 0;
SLS2 = 0;

// Inicio

LS1 = 0;
LS2 = 0;
LC = 0;
LQ1 = 0;
LQ2 = 0;
LE1 = 0;
LE2 = 1;

// ----- //
// Loop Infinito

while(1)
{
// Controle das luzes nos corredores e sala

    if(SPEX){
        LS1 = 0;
        LS2 = 0;
        LC = 0;
        LE1 = 1;
        LE2 = 1;
        LQ1 = 0;
        LQ2 = 0;
        printf("a");
        delayls();
    }

    if(SPS1 & SLS1)
    {
        LS1 = 1;
        LS2 = 0;
        LC = 0;
        LE1 = 0;
        LE2 = 1;
        LQ1 = 0;
        LQ2 = 0;
        printf("b");
        delayls();
    }

    if(SPS2 & SLS2)
    {
        LS1 = 0;
        LS2 = 1;
        LC = 0;
        LE1 = 0;
        LE2 = 1;
        LQ1 = 0;
        LQ2 = 0;
        printf("c");
        delayls();
    }

    if(SPC & SLC)
    {
        LS1 = 0;
        LS2 = 0;
        LC = 1;
        LE1 = 0;
        LE2 = 1;
        LQ1 = 0;
        LQ2 = 0;
        printf("d");
        delayls();
    }
}

```

```

        if(SLQ)
        {
            LS1 = 0;
            LS2 = 0;
            LC = 0;
            LE1 = 0;
            LE2 = 1;
            LQ1 = 1;
            LQ2 = 1;
            printf("e");
            delay1s();
        }

    } // fim do loop

} // fim do main

//-----//
// Funções //
//delay 1 segundo
void delay1s()
{
    unsigned long int i=0;
    for(i=1;i<167700;i++){
        _nop_();
    }
}

// Configuração da porta serial do uC
void configura_serial (void)
{
    // Configuração do Timer 3 como gerador de BAUD RATE
    T3CON = 0x86;
    T3FD = 0x2d;

    SCON = 0x50;
    TI=1;

    /*
    Tabela de valores Para T3CON e T3FD e os baudrates selecionados
    Divisão do CLOCK do PLL = 0, ou seja, clock de 16.77 MHz
    | BaudRate -> 115200 | BaudRate -> 57600 | BaudRate -> 38400 | BaudRate -> 19200 | BaudRate -> 9600 |
    | T3CON = 0x83;      | T3CON = 0x84;      | T3CON = 0x84;      | T3CON = 0x85;      | T3CON = 0x86;      |
    | T3FD = 0x09;      | T3FD = 0x09;      | T3FD = 0x2d;      | T3FD = 0x2d;      | T3FD = 0x2d;      |
    |-----|-----|-----|-----|-----|
    */
}

```

## Guia do Experimento 2

### Conversores Analógico/Digital (A/D) e Digital/Analógico (D/A)

#### Objetivos

Realizar o processo de conversão de analógico para digital e vice-versa, entendendo conceitos como resolução e amostragem.

#### Conversor A/D

O Bloco de conversor A/D incorpora: 8 canais multiplexados com resolução de 8 ou 12 bits; alimentação simples; sistema de *Track and Hold* e tensão de referência *on chip*. Todos os componentes deste bloco podem ser facilmente configurados e controlados através de 3 registradores SFR.

O conversor A/D é do tipo aproximações sucessivas e a tensão de entrada analógica deve variar de 0 V até  $+V_{REF}$ . Um circuito gerador de referência de alta precisão já vem embutido no chip e é calibrado na fábrica para uma tensão de 2,5 V. Caso a aplicação em questão necessite, a tensão de referência interna pode ser obtida externamente no pino  $V_{REF}$ . Uma referência externa também pode ser utilizada (Figura 1), e seus valores devem compreender entre 2,3 V a  $AVDD$  (tensão de alimentação da parte analógica do chip do ADUC).

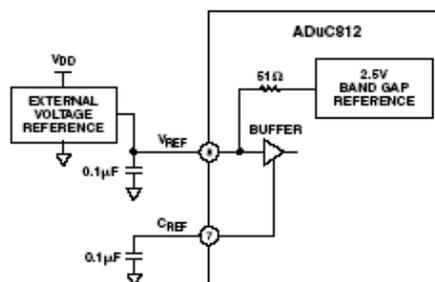


Figura 1 – Utilização de tensão de referência externa para o conversor A/D

Conversões A/D de apenas um passo (uma única conversão) ou de modo contínuo são iniciadas através do *firmware* da aplicação em uso, ou através de um sinal externo aplicado a um pino do ADUC. O Timer 2 também pode ser configurado para gerar gatilhamentos para conversões repetitivas. O modo DMA (*Direct Memory Access*) possibilita a aquisição de um grande volume de dados, de maneira que as conversões são efetuadas e gravadas diretamente na memória externa, sem que haja a interferência do núcleo do microcontrolador neste processo. Este modo de operação permite armazenar até 16 MB de dados coletados.

Na Figura 2, observa-se o diagrama de blocos interno do conversor A/D do ADUC, onde uma chave multiplexadora seleciona qualquer um dos oito canais a ser convertido. O circuito de *Track and Hold* armazena o valor da tensão de entrada do canal selecionado, retendo este valor para que o contador de aproximações sucessivas faça a

conversão A/D. O valor após convertido é armazenado em dois registradores SFR, e um *flag* de fim de conversão é setado.

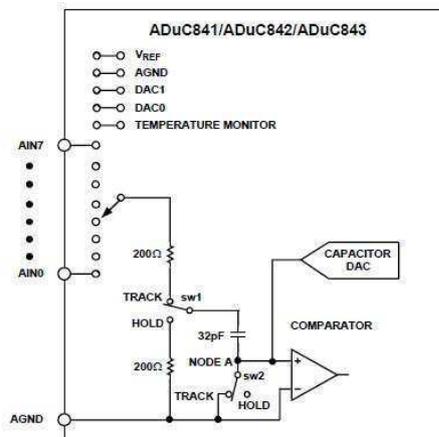


Figura 2 – Estrutura interna do conversor A/D

O procedimento de conversão A/D é todo controlado através de três registradores SFR, são eles:

- **ADCCON1** – ADC Control SFR 1 - responsável pelo controle da temporização de aquisição e de conversão, modos de conversão e modos de regime de consumo.

MD1	MD0	CK1	CK0	AQ1	AQ0	T2C	EXC
-----	-----	-----	-----	-----	-----	-----	-----

MD0 e MD1 – Seleciona o modo de operação do conversor A/D.

MD1	MD0	Modo Selecionado
0	0	Conversor A/D desligado
0	1	Conversor A/D modo normal
1	0	Modo desligado se não está no ciclo de conversão
1	1	Modo de espera se não está no ciclo de conversão

CK0 e CK1 – Ajustam a velocidade de conversão. Um divisor da frequência do clock mestre (clock gerado pelo cristal) de taxa variável é disponibilizado. O usuário pode ajustar a velocidade de conversão, onde a frequência do clock do cristal do microconversor é dividida por 1, 2, 4 ou 8, produzindo o clock do Conversor A/D. Um ciclo de conversão típico precisa de 17 Clocks para ser executado.

CK1	CK0	Divisão do clock mestre
0	0	1
0	1	2
1	0	4
1	1	8

AQ0 e AQ1 – Ajustam a velocidade da aquisição, ou seja, ajusta o tempo em que o circuito de *Track and Hold* adquire o valor da entrada analógica. O tempo é dado por períodos do clock do conversor A/D ajustados em CK0 e CK1.



BUSY – este bit é setado por hardware, e fica em nível alto enquanto o processo de conversão é executado, sinalizando para o usuário que no momento o Conversor A/D está ocupado.

As demais posições do registrador ADCCON3 estão reservadas para uso interno do hardware do bloco do Conversor A/D.

Os três registradores SFR ADCCON, são responsáveis por todo e qualquer ajuste no processo de aquisição e conversão de uma leitura analógica. Depois de concluído o ciclo de conversão no ADUC, o valor convertido (12 bits) é armazenado em outros dois registradores SFR de oito bits chamados de ADCDATAH e ADCDATAL (Figura 3), onde:

ADCDATAH – armazena a parte mais significativa do valor convertido, seu nibble menos significativo, armazena os quatro bits mais significativos do valor convertido, o nibble mais significativo, armazena a identificação do canal onde a leitura foi feita;

ADCDATAL – armazena a parte menos significativa do valor convertido.

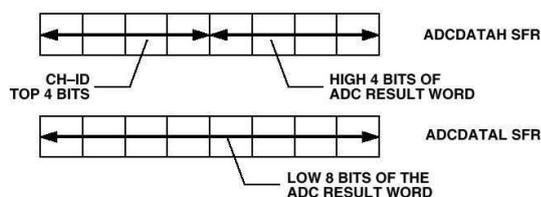


Figura 3 – Armazenamento do valor convertido no ADCDATAH e ADCDATAL

Para garantir uma melhor proteção dos canais de entrada do Conversor A/D, é aconselhável a utilização de *buffers*, além de garantir um melhor acoplamento com a tensão em amostragem. Para uma taxa de amostragem de 200 KHz, um filtro passa baixa deve ser montado após o *buffer*, como mostrado na Figura 4, de forma a eliminar ruídos de alta frequência, que possam produzir erros nas leituras.

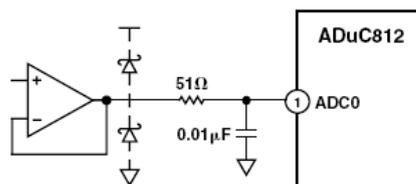


Figura 4 – circuito de acoplamento e rejeição de ruídos

O funcionamento do Conversor A/D dos microconversores da família ADUC é bem parecido, modificando-se entre si algumas poucas características, como número de canais e resolução. Os registradores, valores e procedimentos apresentados nesta seção são baseados no microconversor ADUC 812, que é o primeiro membro desta família, e os demais dispositivos são baseados em sua estrutura. Para ter informações detalhadas sobre qualquer um dos dispositivos da família ADUC, é recomendável estudar em detalhes a folha de dados, de forma que todos os recursos disponíveis sejam utilizados corretamente.

## Conversor D/A

A maioria dos dispositivos da família de microconversores ADUC possui dois conversores D/A de 12 bits. O buffer de saída do Conversor possibilita uma tensão com valores que podem variar de 0 V à VDD (*Rail to Rail*).

O Conversor D/A é formado por uma rede resistiva de  $2^{12}$  resistores de valores iguais. A tensão em cada resistor equivale a um degrau na tensão de saída do Conversor D/A, de forma que uma chave controlada pelo registrador de SFR, que armazena o valor da conversão, seleciona a tensão de saída escolhida (Figura 5).

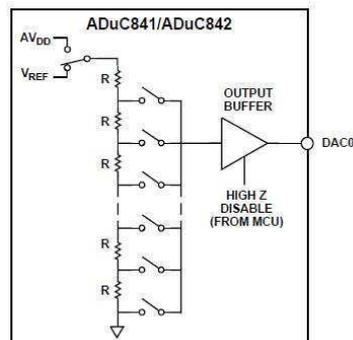


Figura 5 – Estrutura interna do conversor D/A

A tensão de saída do conversor D/A pode ter seus valores variando de 0 V à VDD ou  $V_{REF}$ , onde o usuário pode fazer a seleção de acordo com as necessidades da aplicação.

Da mesma forma que o conversor A/D, o conversor D/A possui registradores SFR de comandos e ajustes de funcionamento, bem como os registradores que recebem os valores a serem convertidos. Os registradores são:

- **DACCON** – *DAC Control SFR* – SFR para controle e ajuste de parâmetros dos conversores D/A.

MODE	RNG1	TNG0	CLR1	CLR0	SYNC	PD1	PD0
------	------	------	------	------	------	-----	-----

**MODE** – Bit de seleção do modo de operação para ambos os conversores D/A para 12 bits ou 8 bits (“0” – modo de 12 bits / “1” – modo de 8 bits);

**RNG1 e RNG0** – *Range select bit* – bit de seleção de escala da tensão de saída do conversor D/A (“0” – escala de 0 V à  $V_{REF}$  / “1” – escala de 0 V à VDD ). RNG0 – Seleção de escala para o conversor D/A 0. RNG1 – Seleção de escala para o conversor D/A 1;

**CLR1 e CLR0** – *DAC Clear bit* – bit para zerar o valor da saída do respectivo Conversor D/A (“0” – Força a saída do conversor D/A para 0V / “1” mantém o conversor D/A funcionando em modo normal).

**SYNC** – Bit de sincronia. Em determinadas aplicações é necessário que o processo de atualização dos conversores D/A seja simultâneo. Em aplicações em sistemas de controle, a amostragem e ação de controle devem estar dentro de um mesmo passo do processo. O bit SYNC permite que os conversores D/A trabalhem sincronizados ou de

maneira independente. Quando em nível lógico “1” os conversores trabalham de maneira independente, e são atualizados a medida que os registradores DACxL/H são preenchidos. Quando em nível lógico “0”, os conversores permitem que os valores sejam escritos nos respectivos SFR, e são atualizados na saída quando o bit SYNC é setado para o nível lógico “1”. O bit SYNC, pode ser comparado à um pino de OE (*output enable*) para os registradores DACxL/H;

PD1 e PD0 – *Power Down bit* – Como os dispositivos da família ADUC são designados para aplicações embarcadas, que por sua vez exigem, um baixo consumo de energia, o usuário pode desligar os conversores D/A quando não estiver utilizando-os. Assim os registradores PD1 e PD2 podem desligar o respectivo conversor D/A quando necessário (“0” – Desliga o conversor D/A / “1” – Liga o conversor D/A);

- **DACxH/L DAC Data Registers** – Registradores SFR que recebem os valores para a conversão para analógico.

DAC0L – Armazena a parte menos significativa do valor a ser convertido pelo conversor D/A 0 (8-bits);

DAC0H – Armazena a parte mais significativa do valor a ser convertido pelo conversor D/A 0, apenas o *nibble* menos significativo é utilizado, já que o conversor D/A é de 12 bits (8-bits);

DAC1L – Armazena a parte menos significativa do valor a ser convertido pelo conversor D/A 1 (8-bits);

DAC1H – Armazena a parte mais significativa do valor a ser convertido pelo conversor D/A 1 (8-bits);

Na Figura 6, pode-se observar que para a proteção das saídas do conversor D/A deve-se utilizar *buffers* externos, possibilitando também uma drenagem de corrente um pouco maior para os sistemas que precisam da tensão de saída do conversor D/A.

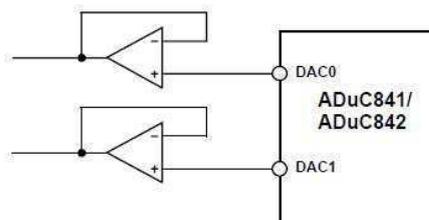


Figura 6 – Utilizando buffer nas saídas do conversor D/A

## Bancada de experimento

Na simulação de automação residencial da bancada, será realizado o controle de temperatura de um ambiente. Um sistema de aquecimento acionado manualmente é tido como uma perturbação externa. Um sensor de temperatura LM35, ligado ao conversor A/D, mede a temperatura do ambiente e um cooler, conectado ao conversor D/A, realiza o resfriamento.

Para este experimento, será usada a placa 02, que deve alimentada por uma fonte de tensão contínua de 12 V que forneça até 3,5 A. Também é utilizado o LM35 para medir temperaturas, o cooler para fazer o resfriamento e os resistores de fio para o aquecimento. Os resistores de fio são acionados diretamente por uma chave própria (Aquecedor) presente na placa de alimentação da bancada.

O circuito driver para o LM 35 está conectado a porta ADC7, que é igual a porta P1.7 do ADuc842. O cooler é acionado pela porta DAC0 do microcontrolador. Uma tensão de referência externa é fornecida pela placa 02.

### **Plataforma virtual**

Este experimento tem por objetivo monitorar a temperatura do cômodo denominado de quarto e mostrar ao usuário por meio de um termômetro virtual com quantos graus °C está o cômodo no momento do experimento e mostrar em que momento o ventilador foi acionado com o objetivo de resfriar o cômodo caso ele ultrapasse uma temperatura previamente estabelecida.

## Programa exemplo

```
// UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG
// DEPARTAMENTO DE ENGENHARIA ELETRICA - DEE
// LABORATORIO DE INSTRUMENTAÇÃO E CONTROLE - LIEC
//
// DISC: SISTEMAS DE AQUISIÇÃO DE DADOS E INTERFACE
// PROF: JAIDILSON JÓ DA SILVA
//
//
// Experimento 2: Conversores A/D - D/A

//-----//
// Cabeçalhos
#include "aduc842.h" // Cabeçalho do ADuC842
#include <stdio.h> // Cabeçalho de Padrão de I/O
#include <intrins.h>

//-----//
// Variáveis //
unsigned char canal_ad;
long valor_ad;
long valor_da;

int cont; i;
long lm35; cooler;
float temperatura; tensaoDA;

//-----//
// Prototipos de Funções //
void configura_serial(void);
void configura_ad(void);
long leitura_ad(char);
void configura_da(void);
void escrita_da(long);
void delayls ();

//-----//
// Programa Principal //
void main()
{
    PLLCON = 0x88; // Clock de 16.77 MHz

    configura_serial();
    configura_ad();
    configura_da();

    //Inicio

    // ----- //
    // Loop Infinito
    while(1)
    {
        lm35 = leitura_ad(7);

        temperatura = lm35*0.122100122100;
        delayls();
        if(temperatura<32)
        {
            cooler = 0;
        }
        else
        {
            cooler = 6*lm35 + 1635;
        }

        if(temperatura > 38)
        {
            cooler = 4095;
        }
    }
}
```

```

        escrita_da(cooler);

        tensaoDA = cooler*0.122100122100;

        printf("%ld\t",lm35);

    } // fim do loop
} // fim do main

//-----//
// Funções //
// Função delay 1 segundo
void delay1s()
{
    unsigned long int i=0;
    for(i=1;i<83850;i++){
        _nop_();
    }
}

// Configuração da porta serial do uC
void configura_serial (void)
{
    // Configuração do Timer 3 como gerador de BAUD RATE
    T3CON = 0x86;
    T3FD = 0x2d;

    SCON = 0x50;
    TI = 1;

    /*
    Tabela de valores Para T3CON e T3FD e os baudrates selecionados
    Divisão do CLOCK do PLL = 0, ou seja, clock de 16.77 MHz
    */


| BaudRate -> 115200 | BaudRate -> 57600 | BaudRate -> 38400 | BaudRate -> 19200 | BaudRate -> 9600 |
|--------------------|-------------------|-------------------|-------------------|------------------|
| T3CON = 0x83;      | T3CON = 0x84;     | T3CON = 0x84;     | T3CON = 0x85;     | T3CON = 0x86;    |
| T3FD = 0x09;       | T3FD = 0x09;      | T3FD = 0x2d;      | T3FD = 0x2d;      | T3FD = 0x2d;     |


    */
}

// Configuração do conversor analógico/digital
void configura_ad(void)
{
    // Configuraçõ do Conversor A/D
    ADCCON1 = 0xC0; // 1100 0000
    // Conversão mais rápida
    // Referência externa
}

// Leitura do canal selecionado do ADC
long leitura_ad (unsigned char canal)
{
    ADCCON2 =( ADCCON2 & 0xF0) | canal; // Seleciona o canal
    cont = 0;
    valor_ad = 0;

    while(cont<16) // Realizando 16 medições consecutivas
    {
        SCONV = 1; // Dispara a conversão

        while (ADCCON3 && 0x80); // Espera a conversão Terminar
        valor_ad += ADCDATAH | ((ADCDATAH & 0x0F) << 8); // Valor do canal
do ADC selecionado

        cont++;
    }

    return (valor_ad/16); // Retorna a média das 16 medições
}

```

```
// Configuração do conversor digital/analógico
void configura_da(void) //Configuração do conversor digital/analógico
{
    DACCON = 0x1D; //      0001 1101
                        //      Modo 12 bits
                        //      Referência externa
                        //      DACx mantido no estado normal
                        //      Dax atualizado quando DACxL é escrito
                        //      ADC0 ON | ADC1 OFF
}

// Escrita no canal DA0
void escrita_da(long valor_da) //Leitura do canal selecionado do ADC
{
    DAC0H = ((valor_da >> 8) & 0x000F);
    DAC0L = (valor_da & 0x00FF);
}
}
```

## Guia do Experimento 3

### Gerador PWM

#### Objetivos

Verificar o funcionamento do gerador de PWM do microcontrolador ADuC842.

#### O Gerador PWM

Alguns dispositivos da família de microconversores ADuC possuem em seu hardware interno, moduladores PWM, que podem ser utilizados diretamente como conversores D/A.

A técnica de modulação PWM (*Pulse Width Modulation* – Modulação em Largura de Pulso) permite que vários níveis de tensão sejam produzidos a partir de um sinal binário, variando apenas a largura do pulso em nível alto. O ciclo de trabalho (*Duty Cycle*) de um sinal binário é a relação entre o tempo de nível alto e o seu período. Utilizando uma interface adequada é possível então fazer o controle do nível de potência entregue para determinado tipo de carga. Os moduladores PWM são largamente empregados na indústria, principalmente em circuitos inversores de frequência para o controle de motores de indução. A possibilidade da manipulação direta das tensões e frequências de um sinal a partir de um sinal binário faz com que um sistema microprocessado utilize a modulação PWM, o que caracteriza uma ferramenta versátil para sistemas de acionamento. Com o uso de filtros sintonizados adequadamente, o nível médio do sinal modulado é recuperado, podendo então ser aplicado diretamente na excitação de dispositivos lineares.

Na Figura 1 apresenta-se uma onda quadrada (os tempos de nível alto e baixo são iguais), onde o seu nível médio é justamente a metade da amplitude máxima do sinal ( $A$ ). O nível médio é alterado justamente variando o tempo de nível alto ( $Ta$ ) e mantendo o período ( $T$ ) do sinal constante, ou seja, variando o ciclo de trabalho. Em (3.1) está definido o Ciclo de Trabalho e em (3.2) e (3.3) a relação entre a tensão média e o Duty Cycle é definida.

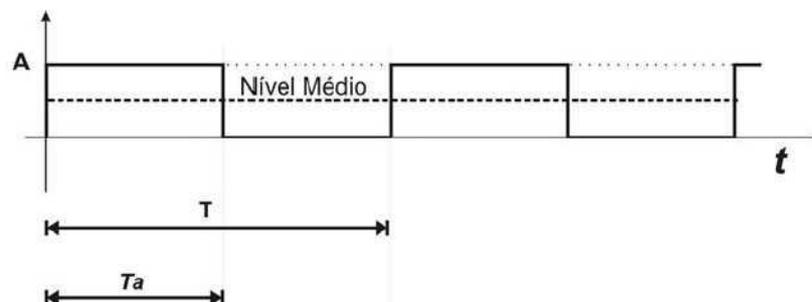


Figura 1 – Representação de uma onda quadrada

$$DutyCycle = \frac{Ta}{T} \cdot 100\% \quad (3.1)$$

$$NívelMédio = A \cdot \frac{Ta}{T} \quad (3.2)$$

Ou então:

$$NívelMédio = A \cdot DutyCycle \cdot 100 \quad (3.3)$$

A modulação PWM precisa do intervalo de um período do sinal da portadora, no caso a onda quadrada que terá seu ciclo de trabalho variado, para representar apenas um nível de amplitude. Devido a esta necessidade o sinal modulado (modulante) em PWM pode ser considerado discreto, onde um período da portadora passa a ser o tempo de amostragem do sinal.

O espectro do sinal PWM (Figura 2) é formado pelas seguintes raia: Raia Fundamental  $F_m$  (modulante), Raia da portadora  $F_s$  (frequência de amostragem PWM), os harmônicos da fundamental ao redor da portadora e os harmônicos da portadora. De acordo com o espectro apresentado na Figura 2 o sinal da modulante é recuperado através de uma filtragem passa baixa sintonizada para selecionar apenas a raia  $F_m$ .

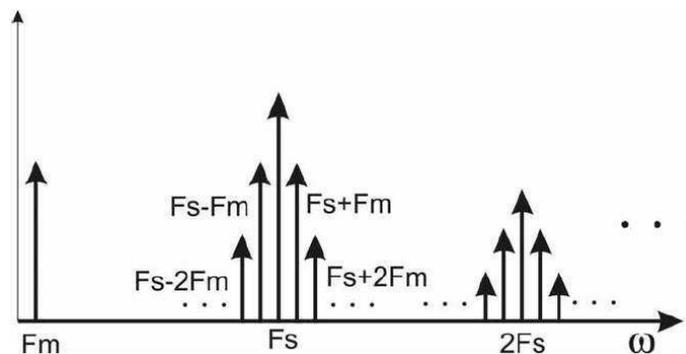


Figura 2 – Espectro de frequência de um sinal modulado em PWM

Ao modular-se um sinal em largura de pulso, quanto maior a frequência de amostragem, melhor será a qualidade da modulação, isto porque, uma quantidade maior de amostras é utilizada para representar o sinal modulado. Quanto maior a frequência da portadora (amostragem), maior será a resolução do sinal representado. O aumento da frequência da portadora, no que diz respeito ao espectro, também tem seus efeitos benéficos, pois a raia  $F_s$  se distancia ainda mais da raia  $F_m$ , o que facilita o processo de recuperação da informação. Nos sistemas de acionamento, por exemplo, devido às características reativas das cargas, as componentes de alta frequência, a partir de certo patamar, podem ser eliminadas sem a necessidade de uma filtragem prévia.

O processo de geração de um sinal PWM é baseado em uma comparação de níveis de valores. Uma tensão de referência de comportamento linear em relação ao tempo deve ser produzida, de forma que sirva de referência para um comparador. Como o sinal PWM possui apenas dois níveis de amplitude, seus valores serão dados por: maior ou menor do que a referência. Um das melhores formas de se gerar uma referência para a modulação

PWM é a partir da comparação do sinal modulante com um sinal dente de serra ascendente. Uma característica muito interessante da onda dente de serra é que o crescimento da amplitude é diretamente proporcional ao período. A frequência da onda dente de serra é a frequência da portadora do sinal PWM, portanto a frequência de amostragem. No diagrama apresentado na Figura 3 ilustra-se a forma como a comparação entre os dois sinais é feita para obter-se o sinal PWM.

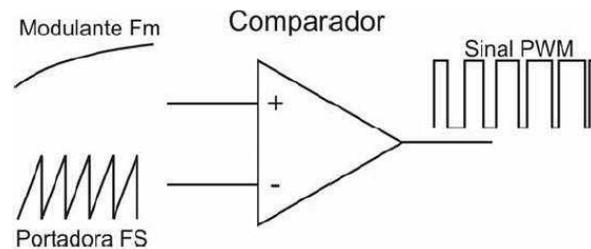


Figura 3 – Comparação dos sinais  $F_m$  e  $F_s$ , produzindo um sinal PWM

Utilizando o diagrama da Figura 3 como referência para geração de um sinal PWM, podemos interpretar o processo de modulação da seguinte forma:

- O sinal de referência dente de serra ( $F_s$ ) deve ser conhecido e bem definido, com valores de amplitudes e frequência fixos;
- O sinal da modulante é aplicado na entrada não inversora (positiva) e então comparado, de forma que cada vez que seu valor for maior que a referência, a saída do comparador apresentará nível lógico alto (Figura 4 A);
- O sinal PWM (Figura 4 B) é produzido em sua saída com os níveis médios variando a cada amostragem (Figura 4 C);

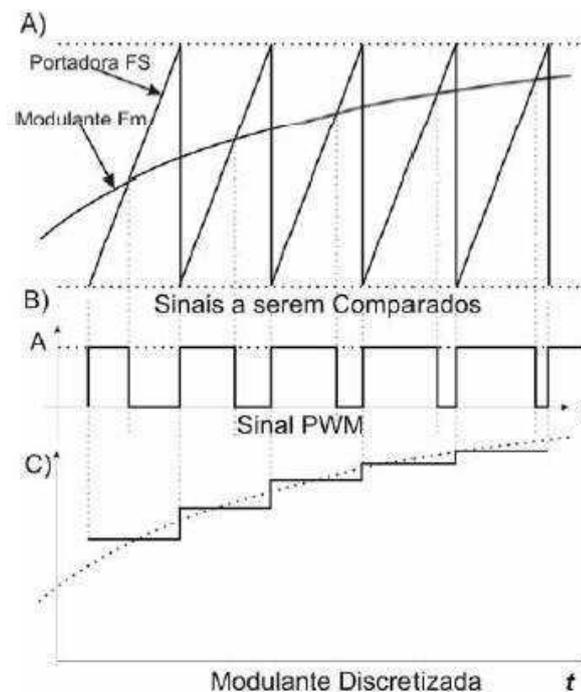


Figura 4 – representação dos passos na modulação PWM por comparação.

A) Sinal da portadora dente de serra e da modulante sobrepostos para comparação;

B) Sinal PWM de saída do comparador;

C) Nível médio do sinal PWM em cada passo de amostra.

Mesmo utilizando um procedimento analógico para explicar a modulação PWM, pode-se lançar mão de um circuito totalmente digital para realizar o mesmo procedimento. Tomando o princípio analógico de comparação para se produzir um sinal PWM, a Analog Devices acrescentou em alguns microconversores da família ADuC um hardware digital apropriado. Neste caso o crescimento do valor da onda dente de serra pode ser interpretado como o crescimento do valor binário de um contador em contagem progressiva. O sinal de referência do modulador PWM digital agora será produzido por um contador, que tem seu período de amostragem igual a um ciclo de contagem completa, desta forma a cada estouro (overflow) do contador, um período do sinal PWM será produzido (Figura 5).

A Resolução do modulador PWM digital dependerá do número ( $n$ ) de bits do contador de referência, desta forma o mesmo pode representar em um período de amostragem  $2^n$  valores diferentes de ciclos de trabalho. Já a resolução no que diz respeito ao número de amostras, depende apenas da frequência da contagem, pois o contador precisa de  $2^n$  incrementos na contagem (ciclos de clock) para gerar um período de amostragem.

O sinal a ser modulado no hardware gerador de PWM digital é interpretado agora como um valor armazenado em um registrador de referência, onde um valor de  $n$  bits deverá ficar retido durante o tempo de um período de amostragem. O tempo mínimo de retenção do valor no registrador de referência deve ser um ciclo completo (contagem) de comparação seja executado.

Da mesma forma que no exemplo analógico (Figura 4), a comparação digital é feita, onde se o valor de referência for maior que o valor corrente da contagem, o nível do sinal de saída é alto. Quando o valor da contagem ultrapassa o valor da referência, o nível do sinal de saída é baixo.

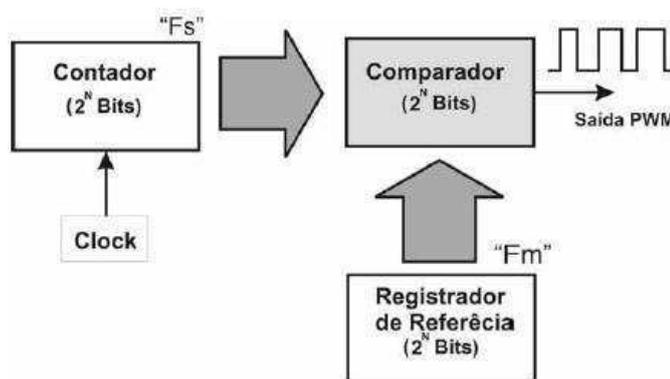


Figura 5 – Diagrama interno de um modulador PWM digital

O modulador PWM pode ser então interpretado como um conversor D/A, pois o valor médio do sinal pulsado poderá variar em  $2^n$  níveis diferentes. Os dispositivos da família ADUC que possuem modulador PWM, são dotados de registradores SFR dedicados para o ajuste de parâmetros de funcionamento e referências. Os modelos possuem duas saídas PWM, onde cada uma delas possui uma referência individual, que pode operar com resolução de 8 ou 16 bits.

No hardware para geração do sinal PWM, o clock da contagem pode ser configurado de várias formas, onde é possível selecionar a fonte do sinal e a divisão de frequência. No microconversores ADUC as fontes de clock para o contador podem ser: o clock do cristal (ex: 11.0592 MHz), o clock do cristal dividido por quinze, um clock externo e o clock do circuito de VCO (Voltage Controlled Oscillator - Oscilador Controlado por Tensão) que é de 16.77 MHz. O circuito de VCO faz parte do hardware de PLL do clock interno do microconversor.

A resolução do gerador PWM é dada de acordo com o modo de contagem do contador interno, 16 ou 8 bits. A escolha da resolução da contagem, bem como modos de operação, seleção de fonte de clock e habilitação do PWM, é feita através do registrador SFR **PWMCON** (PWM Control – Controle do PWM).

Como os microconversores ADUC possuem dois canais para saída PWM, quatro registradores SFR de oito bits são destinados a guardar o valor de referência. Ao trabalhar no modo de 16 bits, são necessários então dois registradores para armazenar um valor de referência (um armazena a parte alta e outro armazena a parte baixa da palavra de referência). Os registradores dos valores de referência são:

**PWM0H** – Armazena o byte mais significativo da referência do canal 0 do PWM;  
**PWM0L** - Armazena o byte menos significativo da referência do canal 0 do PWM;  
**PWM1H** - Armazena o byte mais significativo da referência do canal 1 do PWM;  
**PWM1L** - Armazena o byte menos significativo da referência do canal 1 do PWM.

O diagrama do hardware do gerador PWM dos microconversores da família ADUC (Figura 6) possui um comparador duplo, onde os valores de **PWM0L/H** e **PWM1L/H** são comparados com o valor atual da contagem. Observe que os canais de saída do PWM trabalham sincronizados, pois há apenas um contador, mas o comparador é duplo gerando duas saídas independentes. Os sinais de saída gerados pelo PWM são conectados aos pinos **P2\_6 (PWM0)** e **P2\_7(PWM1)** ou **P3\_4 (PWM0)** e **P3\_3 (PWM1)**. Os pinos de saída selecionados dependem dos ajustes feitos no registrador SFR de configuração geral, que é específico para cada dispositivo da família ADUC. Para o ADUC 832, por exemplo, este registrador é chamado de CFG832.

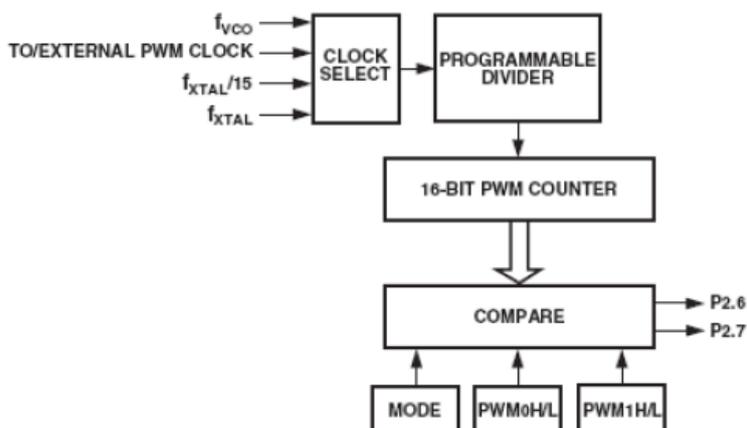


Figura 6 – Diagrama do hardware gerador de PWM digital dos microconversores ADuC

O registrador de controle do gerador PWM é configurado da seguinte forma:

**PWMCON** – PWM Control SFR - responsável pelo controle e ajuste dos parâmetros da geração do sinal PWM.

<b>SNGL</b>	<b>MD2</b>	<b>MD1</b>	<b>MD0</b>	<b>CDIV1</b>	<b>CDIV0</b>	<b>CSEL1</b>	<b>CSEL0</b>
-------------	------------	------------	------------	--------------	--------------	--------------	--------------

**SNGL** – Desliga a saída PWM das portas P2.6 ou P3.4, liberando o pino para funcionamento como entrada e saída digital;

**MD2, MD1 e MD0** – PWM Mod Selector Bits – Bits de seleção do modo de operação do gerador PWM;

<b>MD2</b>	<b>MD1</b>	<b>MD0</b>	<b>Modo de Operação Selecionado</b>
0	0	0	Modo 0: PWM Desabilitado
0	0	1	Modo 1: PWM simples com resolução variável
0	1	0	Modo 2: PWM com resolução de 8 bits com as duas saídas sincronizadas ( <i>Twin Mode</i> – Modo Gêmeo)
0	1	1	Modo 3: PWM com resolução de 16 bits com as duas saídas sincronizadas ( <i>Twin Mode</i> – Modo Gêmeo)
1	0	0	Modo 4: Conversor D/A do tipo $\Sigma - \Delta$ duplo de 16 bits NRZ ( <i>Non Return to Zero</i> – Não Retorna ao Zero)
1	0	1	Modo 5: PWM Duplo de 8 bits
1	1	0	Modo 6: Conversor D/A do tipo $\Sigma - \Delta$ duplo de 16 bits RZ ( <i>Return to Zero</i> – Retorna ao Zero)
1	1	1	Reservado para uso futuro

**CDIV1 e CDIV0** – PWM Clock Divider - Bits de seleção da divisão do clock selecionado para o contador PWM;

<b>CDIV1</b>	<b>CDIV0</b>	<b>Divisão do Clock Selecionado</b>
0	0	Clock Selecionado/1
0	1	Clock Selecionado/4
1	0	Clock Selecionado/16
1	1	Clock Selecionado/64

**CSEL1 e CSEL0** – PWM Clock Selector – Bits de seleção da fonte de clock para o contador PWM;

<b>CSEL1</b>	<b>CSEL0</b>	<b>Fonte de Clock para o Contador PWM</b>
0	0	Clock = $f_{XTAL}/14$
0	1	Clock = $f_{XTAL}$
1	0	Clock = Fonte Externa através do pino P3.4/T0
1	1	Clock = Frequência do VCO = 16.777216 MHz

Os modos de operação do circuito PWM podem ser observados em mais detalhes no manual do dispositivo ADUC em uso. As configurações aqui apresentadas utilizam o ADUC 832 como referência. Desta forma é recomendável que se leia o manual do componente em uso, para que as configurações sejam feitas sem maiores problemas.

### **Bancada de experimento**

O experimento usará o buzzer na placa 01, que deve ser alimentada por uma fonte de tensão contínua de 5 V. As chaves CH1 e CH2 são usadas para mudar a frequência e o duty cycle do PWM, que foi configurado para utilizar a Porta 3 do ADuc842.

### **Plataforma virtual**

Neste experimento será mostrado o gráfico do sinal PWM gerado pelo microcontrolador em tempo real de acordo com o acionamento das chaves. Para cada situação diferente de chaveamento será possível observar tanto a variação na frequência e no dutycycle graficamente quanto em mostradores que indicarão estes valores.

## Programa exemplo

```

// UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG
// DEPARTAMENTO DE ENGENHARIA ELETRICA - DEE
// LABORATORIO DE INSTRUMENTAÇÃO E CONTROLE - LIEC
//
// DISC: SISTEMAS DE AQUISIÇÃO DE DADOS E INTERFACE
// PROF: JAIDILSON JÓ DA SILVA
//
//
// Experimento 3: PWM

//-----//
// Cabeçalhos
#include <ADuC842.h>
#include <stdio.h>
#include <stdlib.h>
#include <intrins.h>

//-----//
// Variáveis
sbit CH1 = 0x87;    // P0.7  -> Chave 1
sbit CH2 = 0x86;    // P0.6  -> Chave 2
int cont;

//-----//
// Protótipos das funções
void configura_serial(void);
void configura_PWM(void);
void delayls ();

//-----//
// Programa principal
void main(void)
{
    PLLCON = 0x80; // Clock de 16.77 MHz
    CFG842 = 0x60; // Transfere PWM pra porta 3.

    configura_serial(); //Configuração da serial
    configura_PWM(); //Configuração do PWM

    cont=0;
    //Inicio

    // ----- //
    // Loop Infinito
    while (1)
    {
        if (CH1 == 0)
        {
            PWM1L = 0x3D;
            PWM1H = 0x25; // PWM1H/L = 9533
                                // Frequência do PWM = (16.777216
MHz/16/9533) = 10 Hz

        }
        if (CH2 == 0)
        {
            PWM0L = 0x4F; // DC = 25%
            PWM0H = 0x09; // PWM0H/L = PWM1H/L * 0.25

            printf("a\n");
            delayls();
        }
        else
        {
            PWM0L = 0xED; // DC = 75%
            PWM0H = 0x1B; // PWM0H/L = PWM1H/L * 0.75

            printf("b\n");
            delayls();
        }
    }
    else
}

```

```

    {
        PWM1L = 0x9E;
        PWM1H = 0x12; // PWM1H/L = 4766
                       // Frequência do PWM = (16.777216 MHz/16/4766) = 20
Hz

        if (CH2 == 0)
        {
            PWM0L = 0xA7; // DC = 25%
            PWM0H = 0x04; // PWM0H/L = PWM1H/L * 0.25

            printf("c\n");
            delay1s();
        }
        else
        {
            PWM0L = 0xF6; // DC = 75%
            PWM0H = 0x0D; // PWM0H/L = PWM1H/L * 0.75

            printf("d\n");
            delay1s();
        }
    }

} // fim do loop

} // fim do main

//-----//
// Funções //
// Função delay de 1 segundo
void delay1s()
{
    unsigned long int i=0;
    for(i=1;i<167700;i++){
        _nop_();
    }
}

// Configuração da porta serial do uC
void configura_serial (void)
{
    // Configuração do Timer 3 como gerador de BAUD RATE
    T3CON = 0x86;

    SCON = 0x52;
    T3FD = 0x2d;

    /*
    Tabela de valores Para T3CON e T3FD e os baudrates selecionados
    Divisão do CLOCK do PLL = 0, ou seja, clock de 16.77 MHz
    | BaudRate -> 115200 | BaudRate -> 57600 | BaudRate -> 38400 | BaudRate -> 19200 | BaudRate -> 9600 |
    | T3CON = 0x83;      | T3CON = 0x84;      | T3CON = 0x84;      | T3CON = 0x85;      | T3CON = 0x86;      |
    | T3FD = 0x09;      | T3FD = 0x09;      | T3FD = 0x2d;      | T3FD = 0x2d;      | T3FD = 0x2d;      |
    |-----|-----|-----|-----|-----|
    */
}

// Configuração da saída PWM
void configura_PWM(void)
{
    PWMCON = 0x9F; // Mode 1
                // PWM Counter = Selected Clock/64
                // PWM Clock = 16.777216 MHz
}

```