



**Universidade Federal de Campina Grande**  
**Centro de Engenharia Elétrica e Informática**  
**Unidade Acadêmica de Engenharia Elétrica**

PEDRO MERENCIO PRIMO PASSOS

Relatório de Estágio  
Desenvolvido no Laboratório de  
Instrumentação Eletrônica e Controle

Campina Grande, Paraíba  
Dezembro de 2015

PEDRO MERENCIO PRIMO PASSOS

Relatório de Estágio  
Desenvolvido no Laboratório de  
Instrumentação Eletrônica e Controle

*Relatório das atividades 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: Instrumentação Eletrônica

Orientador:  
Professor Jaidilson Jó da Silva, D. Sc.

Campina Grande, Paraíba  
Dezembro de 2015

PEDRO MERENCIO PRIMO PASSOS

Relatório de Estágio  
Desenvolvido no Laboratório de  
Instrumentação Eletrônica e Controle

*Relatório das atividades 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: 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

## **AGRADECIMENTOS**

Primeiramente a Deus.

Agradeço ao professor Jaidilson pela oportunidade oferecida neste estágio e por toda atenção dada durante sua execução. Aos meus colegas de sala e, também, amigos: Ana, Luiz, Maria, Nathalia, Roana e Tony. Agradecimento especial a Simões.

Por fim, agradeço a todos que, na sinceridade da consciência, sabem que me fizeram bem.

## RESUMO

As atividades aqui descritas foram realizadas durante a execução do estágio supervisionado, iniciado em 11 de maio de 2015, e finalizado em dezembro de 2015. Cumprindo, pois, a carga horária mínima de 180 horas. Neste trabalho foi desenvolvida uma plataforma para experimentos didáticos na área de instrumentação eletrônica, mais precisamente, para a disciplina Sistemas de Aquisição de Dados e Interface do curso de graduação em engenharia elétrica da Universidade Federal de Campina Grande. A plataforma foi idealizada para ser utilizada junto ao microcontrolador ADuC842, da Analog Devices<sup>TM</sup>.

**Palavras-chave:** Instrumentação eletrônica, microcontrolador.

## SUMÁRIO

1. Introdução.....	6
2. Desenvolvimento .....	9
2.1 Sobre a maquete .....	9
2.2 Sobre os <i>drivers</i> .....	12
3. Atividades desenvolvidas .....	20
3.1. Bancada experimental .....	20
3.1.1. Os experimentos .....	20
3.1.2. Sobre a bancada .....	21
3.1.3. Sensores e atuadores .....	22
3.1.4. Circuitos drivers .....	23
3.2 Guia do experimento 1 – Comunicação serial com ADuC842 e Portas I/O .....	25
3.3 Guia do experimento 2 – Conversores Analógico/Digital e Digital/analógico ..	31
3.4 Guia do experimento 3 – Gerador PWM .....	42
3.5 Guia do experimento 4 – Temporizadores e interrupções .....	51
4. Conclusão .....	58
5. Referências Bibliográficas .....	59

## 1. Introdução

O Laboratório de Instrumentação Eletrônica e Controle (LIEC), localizado no edifício *Gurdip Singh Deep* (Figura 01), é um dos laboratórios da Unidade Acadêmica de Engenharia Elétrica (UAEE) da Universidade Federal de Campina Grande (UFCG), onde são realizadas pesquisas nas áreas de controle, automação e eletrônica em nível de graduação e pós-graduação. Os projetos relacionados a esses estudos contam com a participação tanto dos professores, como dos alunos vinculados ao departamento [1].



**Figura 01:** Fachada do edifício *Gurdip Singh Deep* na UFCG, sede do LIEC.

O LIEC é subdividido em setores onde cada ramo de pesquisa está concentrado. Dentre estes setores estão os:

- Laboratório de Aplicações Wireless
- Laboratório de Automação Industrial
- Laboratório de Controle e Otimização
- Laboratório de Instrumentação Eletrônica
- Laboratório de Redes Industriais
- Laboratório de RFID
- Laboratório de UltraSom

Destes, o Laboratório de Instrumentação Eletrônica, localizado na sala 08 do bloco, visa o estudo de dispositivos transdutores, circuitos eletrônicos de condicionamento de sinais, microcontroladores, sistemas de interfaces e as possíveis aplicações de engenharia que fazem uso destes tópicos.

A disciplina Sistemas de Aquisição de Dados e Interface pertence à ênfase de Eletrônica do curso de graduação em Engenharia Elétrica da UFCG. A disciplina tem uma carga horária total de 75 horas, incluindo as aulas teóricas e laboratoriais. Seu pré-requisito é a disciplina Instrumentação Eletrônica.

A ementa desta disciplina contém os seguintes tópicos [2]: Microprocessadores e microcontroladores; Aquisição de dados via computador; Circuitos de interface; Circuitos de acionamento; Condicionamento de sinais; Deslocamento de nível; Introdução a filtros digitais; Análise harmônica e espectral; Temporizadores; Protocolos de interfaces; Técnicas de modulação e demodulação com sistemas microcontrolados; Sistemas embarcados.

A disciplina objetiva instruir o aluno sobre como proceder ao desenvolver um projeto eletrônico que faz uso de um microcontrolador, bem como de uma interface virtual onde as grandezas do projeto podem ser monitoradas ou modificadas. Tal procedimento visa respeitar as características de funcionamento dos dispositivos eletrônicos necessários no projeto e, também, as características do controlador.

No seu contexto atual, os tópicos desta disciplina são ministrados em volta do microcontrolador ADuC842 e do software LabVIEW, desenvolvido pela *National Instruments*. As atividades experimentais visam ilustrar como utilizar as funções mais básicas destas ferramentas em uma aplicação de engenharia de maior escopo. Para que, ao final, o aluno possa desenvolver um projeto no qual o microcontrolador realize a aquisição de dados de um sistema de sua definição e o LabVIEW gere a interface entre o sistema e o usuário.

O ADuC842 é um microcontrolador fabricado pela *Analog Devices* (Figura 02), seu núcleo usa a arquitetura 8052 e é compatível com as instruções da 8051. Foi concebido para aplicações em redes ópticas, sistemas com base remota, instrumentação de precisão e sensores inteligentes, entre outros [3].



**Figura 02:** Logomarca da *Analog Devices*, fabricante do ADuC842.



Este microcontrolador, ilustrado na Figura 03, possui como temporizador interno um cristal de 32 kHz, além de conversores digital/analógico e vice-versa de até 12 bits de resolução. 62 kBytes de memória de programa e 4 kBytes de memória de dados, com suporte a memória externa de acesso rápido. O microcontrolador também possui quatro portas, de 8 bits cada, de uso geral, onde algumas compartilham funcionalidades secundárias. Possui 12 sistemas de interrupções que podem assumir duas hierarquias de prioridade, alta ou baixa [3].

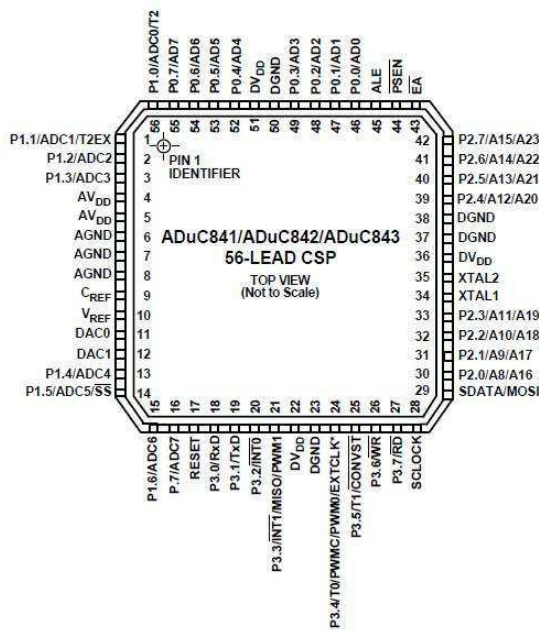


Figura 03: Esquemático do microcontrolador ADuC842;

O ADuC842 também possui outras funcionalidades implementadas digitalmente, que podem ser acessadas através dos devidos registradores. 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)* [3]. Sua programação dar-se pelo programa compilador  $\mu$ Vision 3, desenvolvido pela Keil (Figura 4) [4].



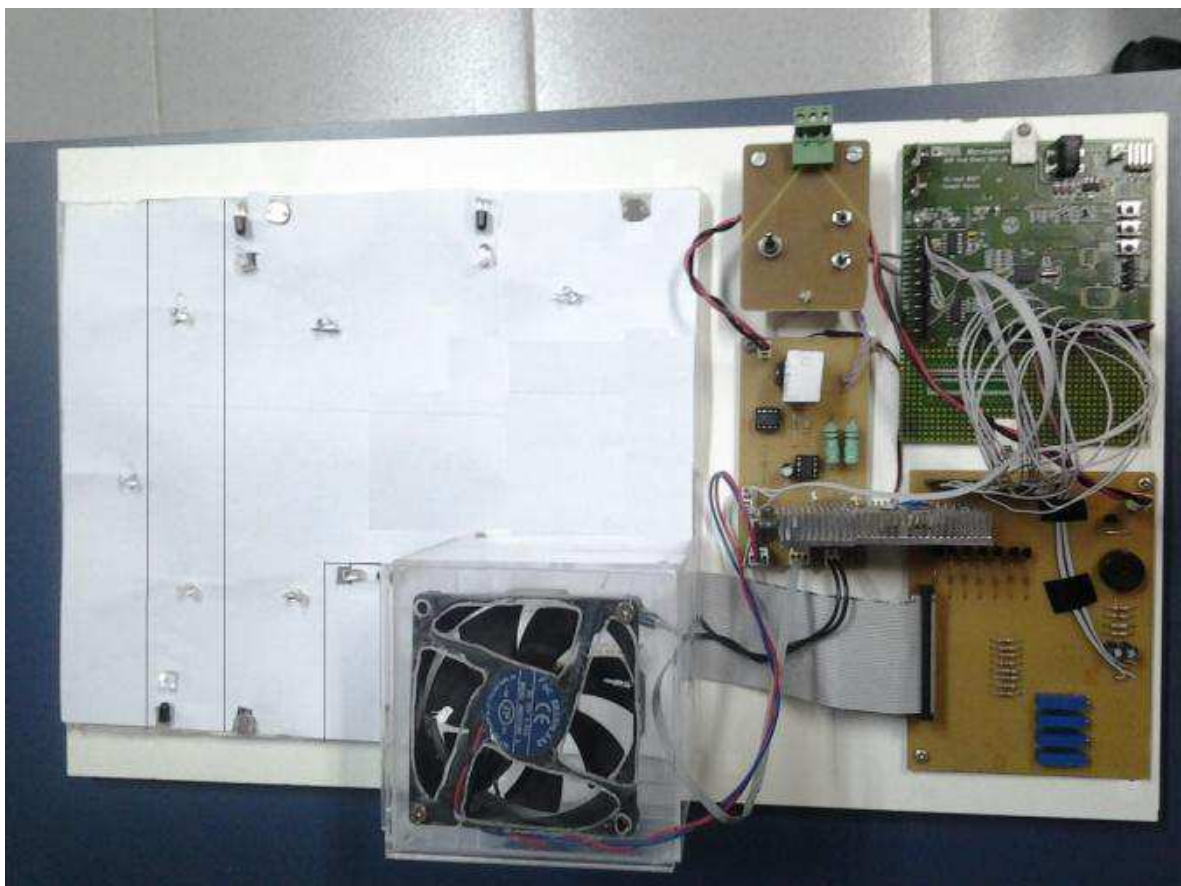
Figura 04: Logomarca da Keil, desenvolvedora do compilador  $\mu$ Vision 3.

## 2. Desenvolvimento

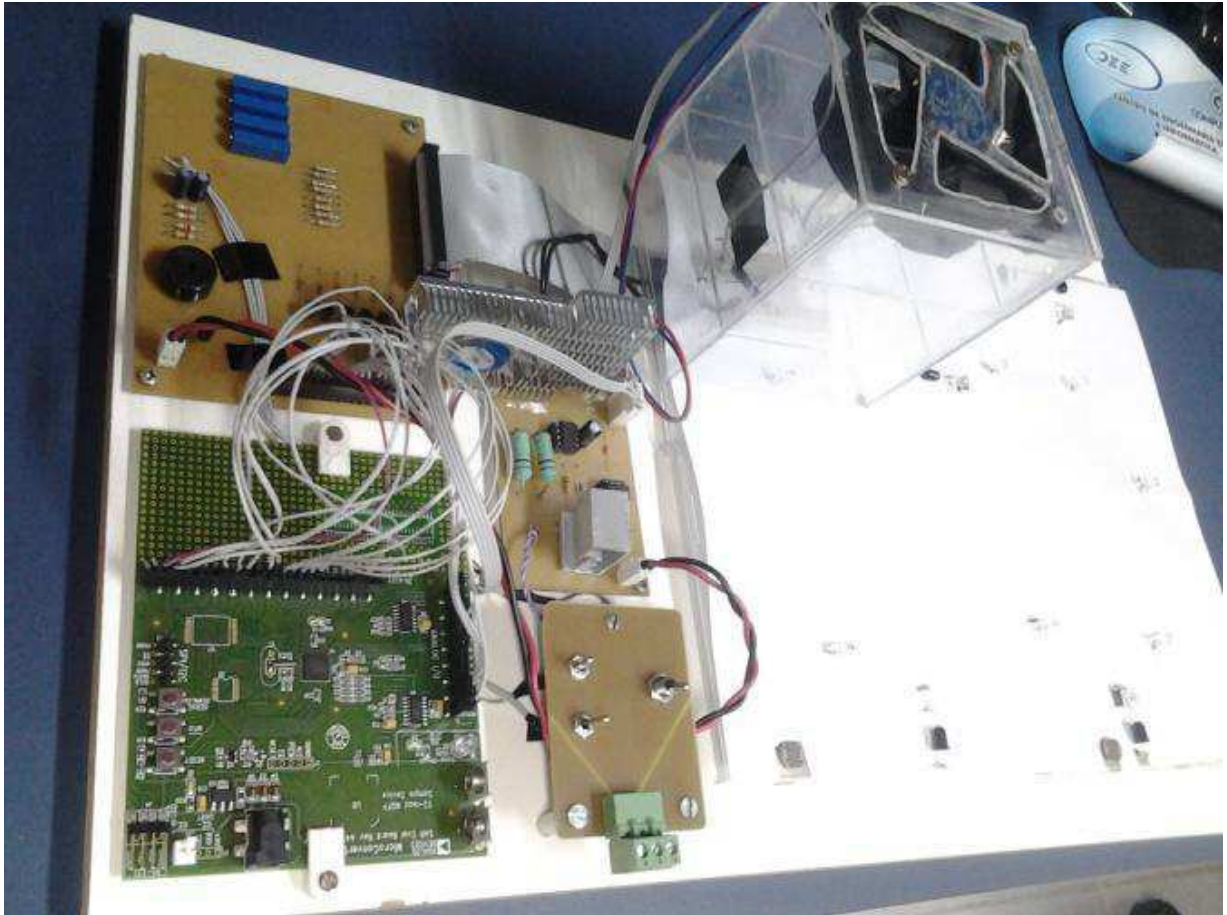
### 2.1. Sobre a maquete

A bancada de experimento foi concebida para simular um sistema de automação residencial. Onde as diversas características de um ambiente habitável podem ser monitoradas e controladas eletronicamente. Tais como a iluminação dos ambientes, a climatização, o abrir e fechar de portas ou janelas e o monitoramento de segurança.

A bancada, ilustrada nas Figuras 05 e 06, conta com uma maquete de modelo reduzido de uma habitação residencial construída de material plástico, sobre uma base de madeira. Nesta base, também se encontram o microcontrolador ADuC842 e as placas de circuito impresso que interligam-no aos dispositivos eletrônicos espalhados pela maquete.



**Figura 05:** Bancada experimental, vista superior.



**Figura 06:** Bancada experimental, vista em perspectiva.

Nesta planta, estão simulados os sistemas de iluminação, temperatura e segurança. Foram utilizados sensores de intensidade luminosa, de presença e de temperatura para o monitoramento. Como atuadores, foram utilizados diodos emissores de luz – do inglês, *light emitting diode (LED)* – resistências aquecedoras, ventoinhas e alarme sonoro.

A maquete está dividida em quatro ambientes denominados: Quarto, Sala, Cozinha e Região Externa, esta ultima subdividida em Varanda e Calçada. Todas as regiões possuem o sistema de iluminação. O quarto possui, também, o sistema de temperatura. E, por fim, a região externa possui o sistema de segurança. Na Figura 07 tem-se a divisão da maquete.



**Figura 07:** Divisão da área da maquete na banca de experimentos.

O sistema de iluminação utiliza resistores dependentes de luz – do inglês, *light dependent resistor (LDR)* – para medir o nível de iluminamento. Há também sensores de presença compostos por LEDs de infravermelho e foto-transistores. Para simular a iluminação do ambiente, utilizou-se LEDs de alto brilho na cor branca.

O sistema de temperatura é composto por resistores de fio para fornecer calor ao ambiente, um sensor de temperatura LM35 e uma ventoinha (*cooler*) para exaustão do ar quente. O sistema de segurança utiliza os sensores de presença e os LEDs da região externa da planta, mais um alarme sonoro (*buzzer*).

Na maquete, a região denominada ‘Quarto’ contém dois LEDs, um LDR, dois resistores de fio de 1 $\Omega$  e 10 W cada, o LM35 e o *cooler*. Os LEDs e o *cooler* são acionados por comandos do microcontrolador. Já os resistores de fio são acionados por um comando manual. A disposição destes dispositivos ao longo da maquete está na Figura 08.

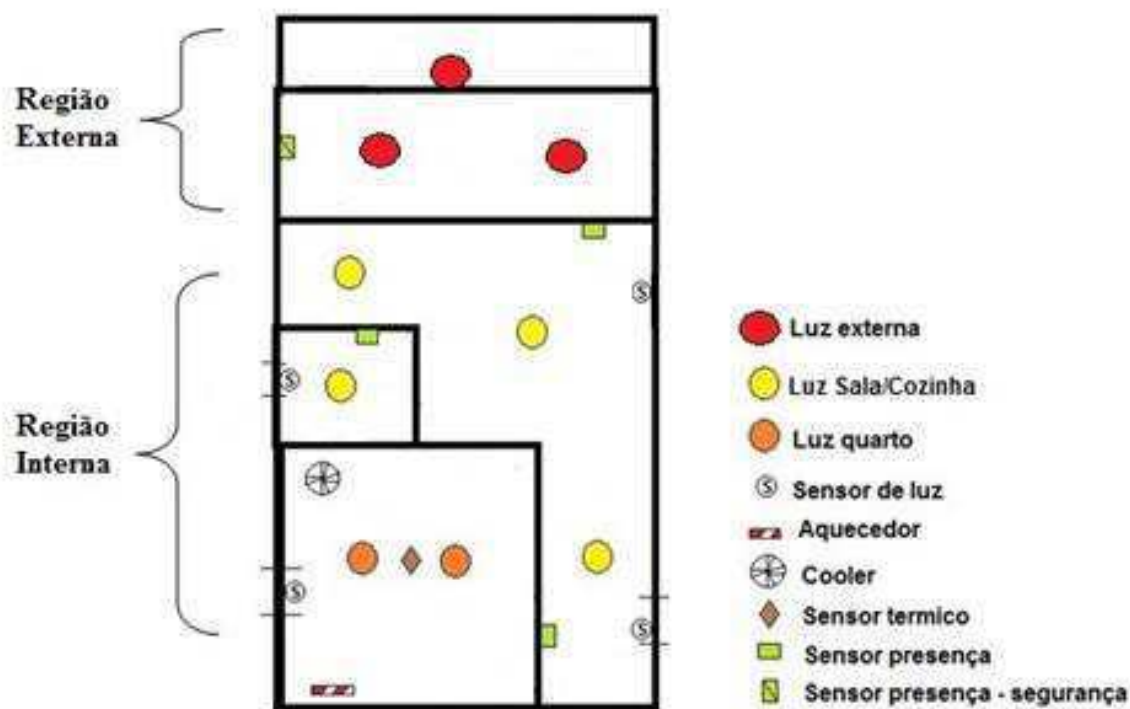


Figura 08: Disposição dos sensores e atuadores na maquete.

A região ‘Sala’ contém dois sensores de presença, dois de iluminação e três LEDs, onde dois desses estão associados em paralelo. A região ‘Cozinha’ contém um LED, um sensor de presença e um de iluminação. E a região externa tem 3 LEDs, dois associados em paralelo e um sensor de presença. O alarme sonoro encontra-se em uma das placas de circuito impresso (*drivers*).

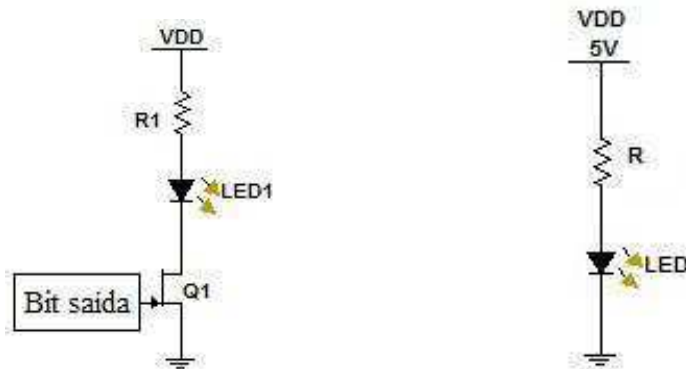
## 2.2. Sobre os *drivers*

Duas placas de circuito impresso foram concebidas para servirem de interface, *driver*, entre o ADuC842 e os dispositivos eletrônicos da maquete. A principal função destas placas é condicionar os níveis das tensões e correntes elétricas para proteção dos dispositivos e, também, para adequar os sinais aos níveis adequados para leituras e escritas realizadas pelo microcontrolador.

A primeira placa *driver* foi desenvolvida para conectar todos os LEDs, LDRs, fototransistores e o *buzzer* ao ADuC842. Projetou-se um circuito eletrônico para cada tipo de dispositivo. Para o acionamento dos LEDs brancos de alto brilho, utilizou-se um transistor bipolar para não haver dreno excessivo de corrente nas portas do microcontrolador. Os LEDs



de infravermelho são mantidos sempre ligados, logo, utilizou-se apenas um resistor para limitar a corrente em cada LED. A Figura 09 traz o esquema elétrico destes circuitos.



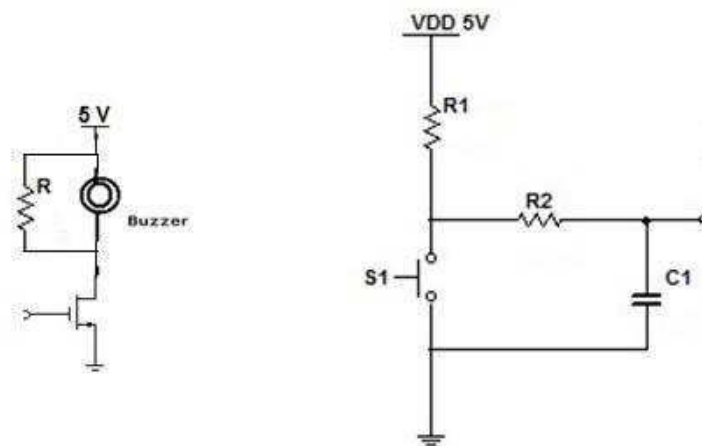
**Figura 09:** Circuitos *drivers* para os LEDs.

Os foto-transistores são utilizados como chaves, logo, são conectados em série a uma resistência. Os LDRs trabalham como divisores resistivos, eles são, pois, ligados em série com potenciômetros para que seus pontos de operação possam ser ajustados ao nível de iluminação do ambiente em que a bancada se encontra. Na Figura 10 encontram-se os esquemas elétricos para o foto-transistor e o LDR.



**Figura 10:** Circuitos *drivers* para o foto-transistor e o LDR.

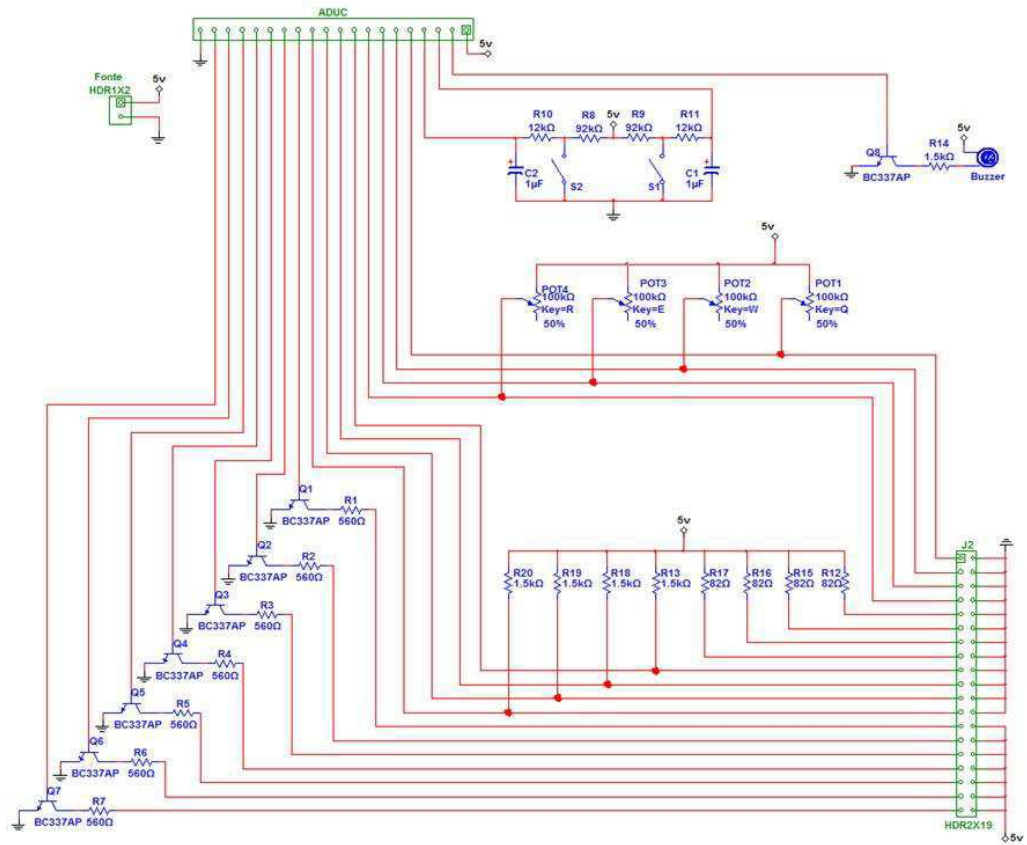
O *buzzer*, por possuir uma alta impedância entre seus terminais, é associado em paralelo com um resistor (Figura 11). Um transistor bipolar faz a ligação do *buzzer* com o ADuC842. Há também duas chaves do tipo liga-desliga, associadas com circuitos RC para filtragem de ruído (Figura 11).



**Figura 11:** Circuitos *drivers* para o *buzzer* e as chaves.

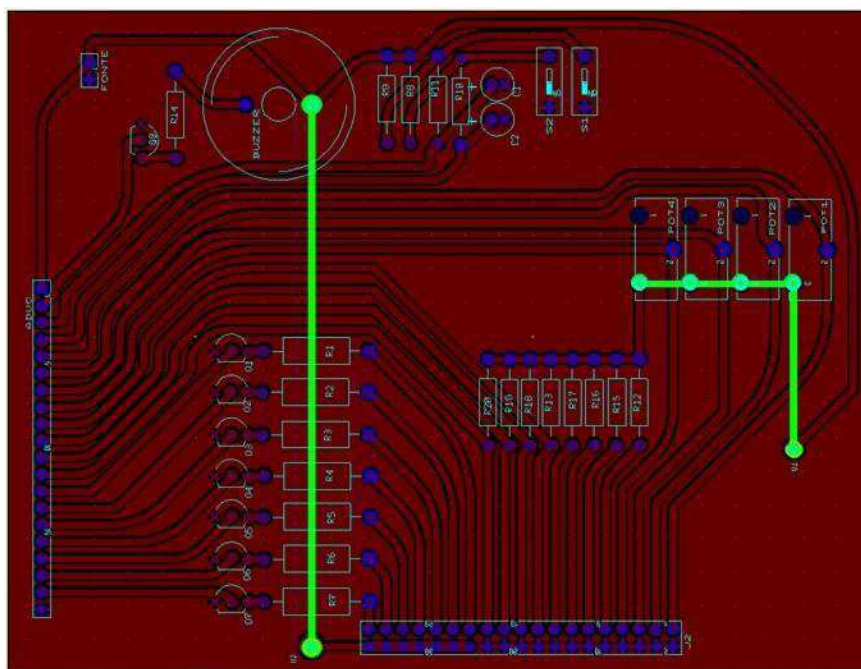
A placa possui duas trilhas de pinos, uma pra sua conexão com as portas do microcontrolador, outra para conexão com os dispositivos na maquete. A placa também possui pinos próprios para receber a alimentação elétrica, que deve ser de 5 Volts DC, com capacidade de fornecer até 0,3 Ampères.

Os circuitos foram projetados e simulados com o auxílio da plataforma Multisim, versão 11.0, desenvolvida pela *National Instruments*. O conjunto com todos os circuitos *drivers* de cada dispositivo foi reunido em um único circuito eletrônico para o desenvolvimento da placa de circuito impresso, cujo esquema elétrico está ilustrado na Figura12.



**Figura 12:** Esquema elétrico da primeira placa driver.

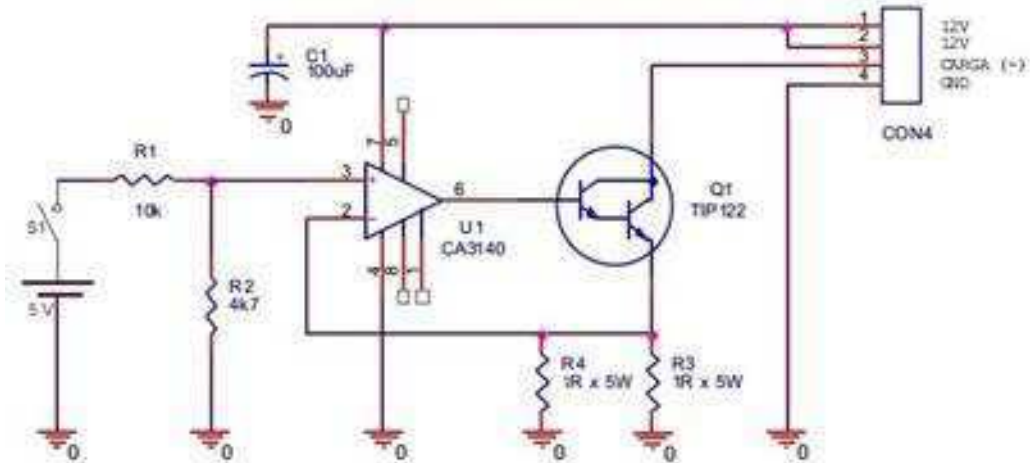
A placa de circuito impresso foi concebida com o uso do Ultiboard 11.0, uma subferramenta do Multisim. Seu esquemático está na Figura 13 seguinte.



**Figura 13:** Esquemático da placa de circuito impresso para a primeira placa driver.

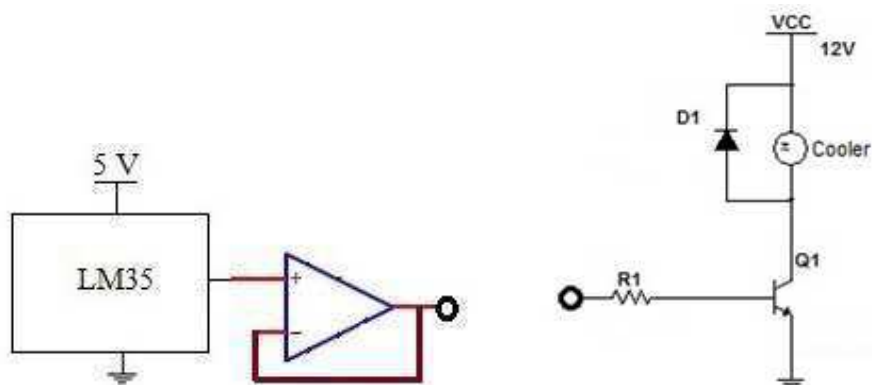


A segunda placa foi projetada para acionar e alimentar os resistores de fio e para realizar a conexão do ADuC842 com o sensor LM35 e o cooler. Assim como na primeira placa, nesta há um circuito *driver* para cada dispositivo a ela conectado. Projetou-se uma fonte de corrente para fornecer cerca de 2 A aos dois resistores de fio que estão associados em série(Figura 14). Um dissipador térmico foi acoplado a este circuito para sua própria proteção. Esta fonte é acionada manualmente por meio de uma chave do tipo liga-desliga.



**Figura 14:** Circuito fonte de corrente.

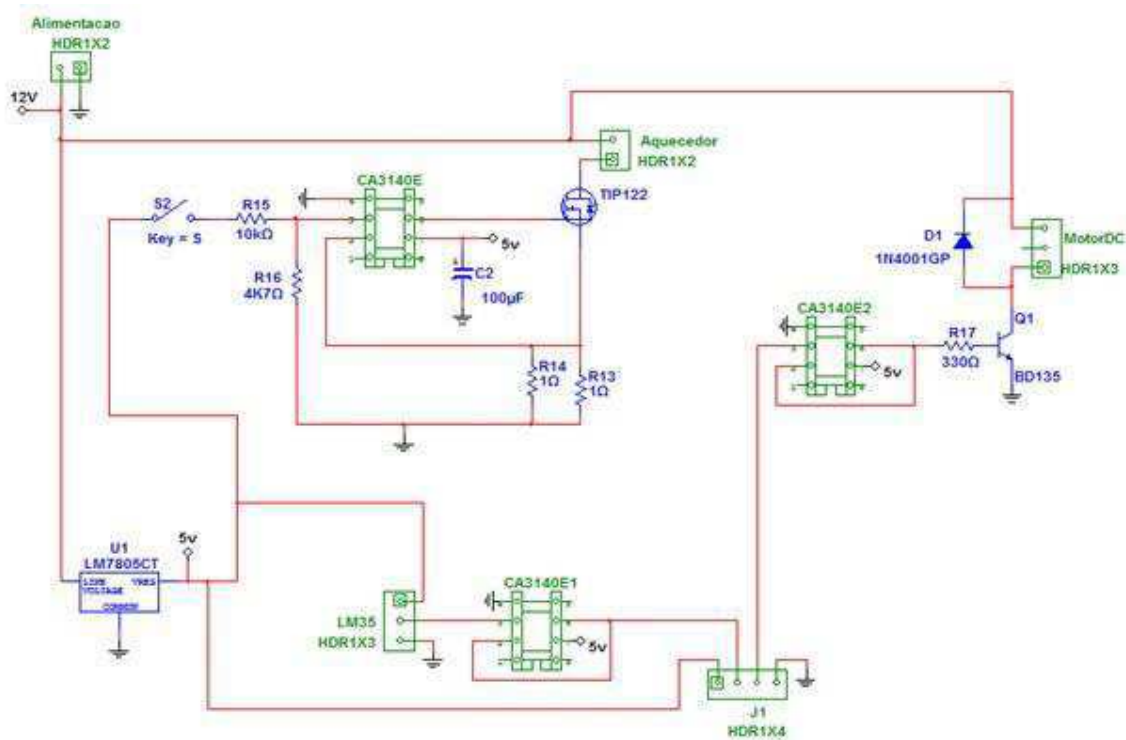
Para o LM35, projetou-se um circuito *buffer* com um amplificador operacional (Figura 15). O *cooler* é conectado em paralelo com um diodo de roda livre, para dissipar sua corrente no desligamento. Ele deve ser alimentado por uma tensão de até 12 V de amplitude. Um transistor bipolar serve como chave para ligá-lo (Figura 15). Que, por sua vez, é conectado ao ADuC842 por meio de um *buffer* semelhante ao usado com o LM35.



**Figura 15:** Circuitos *drivers* para o LM35 e para o *cooler*.

Os buffers e o LM35 trabalham com tensão de 5 V, logo, utilizou-se um regulador de tensão para obter os dois níveis desejados para a placa. Assim sendo, esta placa deve ser alimentada por uma fonte de tensão DC de 12 V, com capacidade de fornecer até 3 A.

Semelhante à primeira placa, os circuitos foram agrupados em um só esquema elétrico e uma placa de circuito impresso foi concebida. As Figuras 16 e 17 trazem as ilustrações do esquema elétrico e o esquemático da placa impressa.



**Figura 16:** Esquema elétrico da segunda placa *driver*.



Nesta mesma porta, nos pinos 6 e 7 estão conectadas as duas chaves da primeira placa. Os quatro LDRs estão conectados aos quatro primeiros bits da Porta 1, que pode funcionar como porta digital, ou analógica[3]. Os sete conjuntos de LEDs da primeira placa estão conectados aos sete primeiros bits da Porta 2. No pino 2 da Porta 3, que pode assumir o papel de gerador de interrupção, está ligado o sensor de presença da região externa da maquete. E o *buzzer*, que é acionado por um sinal PWM, está no pino 4 da Porta 3. O *cooler* é conectado a saída digital/analógica DAC0 do microcontrolador e, por fim, o LM35 é ligado ao pino 7 da Porta 1. A Tabela 01 seguinte traz o mapeamento destas ligações.

**Tabela 01:** Mapeamento dos dispositivos nas portas do ADuC842.

<b>Pino\Porta</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>DACx</b>
<b>0</b>	Sensor de presença	LDR	LED		Cooler
<b>1</b>	Sensor de presença	LDR	LED		
<b>2</b>	Sensor de presença	LDR	LED	Sensor de presença	
<b>3</b>		LDR	LED	Buzzer	
<b>4</b>			LED		
<b>5</b>			LED		
<b>6</b>	Chave Liga-desliga		LED		
<b>7</b>	Chave Liga-desliga	LM35			

### **3. Atividades desenvolvidas**

#### **3.1. Bancada experimental**

A bancada de aquisição de dados objetiva familiarizar o aluno com a ideia do condicionamento de níveis de tensão e corrente para interligação entre o microcontrolador e os sensores e atuadores da planta.

Tal condicionamento faz-se necessário para proteção do próprio microcontrolador, bem como para adequar os sinais para serem processados adequadamente.

Os níveis adequados de tensão e corrente são determinados pelas características do microcontrolador e dos sensores e atuadores. Tais informações podem, geralmente, serem encontradas nos manuais dos fabricantes. De posse disto, projetam-se circuitos eletrônicos que amplifiquem, atenuem, filtrem, ou isolem as grandezas.

##### **3.1.1. Os experimentos**

Serão realizados quatro experimentos, onde cada um demonstrará como utilizar certas funcionalidades do microcontrolador ADuC842[3]. No primeiro serão utilizadas as portas digitais de entrada e saída e a comunicação serial via protocolo RS-232. O segundo experimento traz os conversores analógico/digital e digital/analógico. No seguinte, tem-se as saídas PWM do microcontrolador. E, por fim, os sistemas de interrupções e temporização do ADuC842[5].

Na etapa seguinte desta disciplina, essa bancada também pode ser utilizada para o estudo da plataforma LabView, onde uma interface é criada no computador para a interação com o microcontrolador e a planta.

### 3.1.2. Sobre a bancada

A bancada foi construída para exemplificar como o ADuC842 pode ser utilizado no monitoramento de um sistema, também como devem ser os circuitos eletrônicos interligados a ele. Tais circuitos podem assumir diversas topologias, não limitando-se apenas as utilizadas neste sistema.

Nela está a simulação de um sistema de automação residencial, ou seja, onde as características de uma habitação, tal como a iluminação ou a temperatura, são controladas eletronicamente.

Como ilustrado na Figura 19, o ambiente foi dividido em regiões, semelhante a uma casa. Cada uma possui sua própria iluminação e uma possui um controle de temperatura.

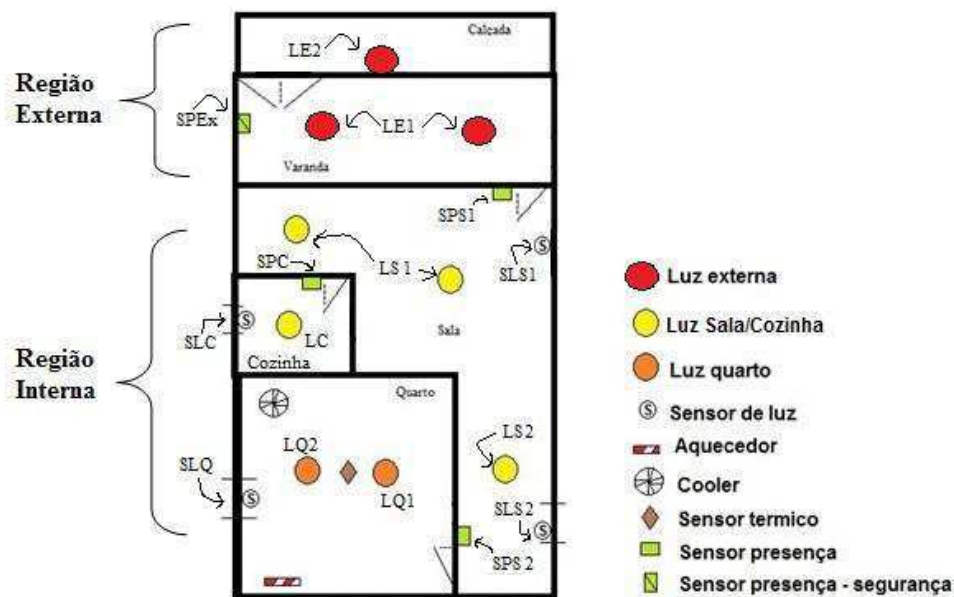


Figura 19: Divisões da bancada de experimentos

A região denominada externa remete à calçada e à varanda. Nela estão as luzes externas LE1 e LE2 e o sensor de presença externo SPEX. A área interna é subdividida em sala, com as luzes, sensores de presença e de luz da sala (respectivamente: LS1, LS2, SPS1, SPS2, SLS1 e SLS2.); quarto, com luzes (LQ1 e LQ2), sensor de luz (SLQ), aquecedor sensor de temperatura e cooler; e cozinha, com luz (LC), sensor de luz (SLC) e sensor de presença (SPS).

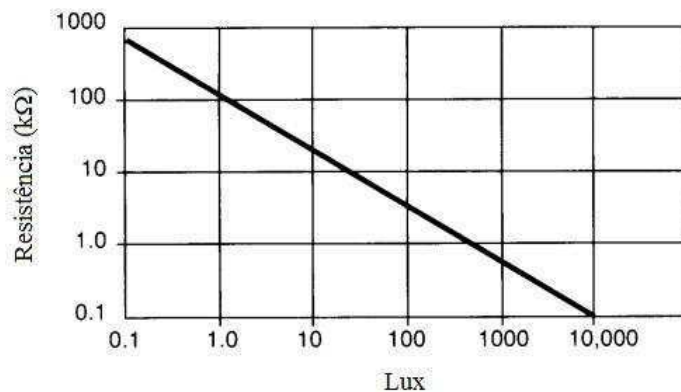
Os sensores de luz e presença indicam quando as luzes devem ser acesas, um sensor de presença é usado para gerar uma interrupção externa, e um sensor de temperatura determina quando acionar a ventilação. Conforme será explicado em cada guia de experimento.

Duas placas de circuito impresso na bancada servem como *drivers* entre o microcontrolador e os dispositivos na planta. A placa 01 será utilizada nos experimentos 1, 3 e 4; e a placa 02 no experimento 2.

### 3.1.3. Sensores e atuadores

Os sensores utilizados são: resistores dependentes de luz, do inglês *light dependent resistor* – LDR; foto-transistores; e o sensor de temperatura LM35.

O LDR é um resistor cujo valor da resistência é inversamente proporcional à iluminância, como ilustrado na Figura 20.



**Figura 20:** Variação da resistência de um LDR em função da iluminância.

O foto-transistor é um dispositivo semicondutor que permite, ou não, a passagem de corrente do coletor para o emissor em função da incidência de luz sobre o mesmo, seja ela no espectro visível, ou infravermelho. Ocorrerá condução quando houver a incidência. Utilizado junto com LED de infravermelho, pode ser considerado como sensor de presença.

O LM 35 é um sensor de temperatura de precisão com resposta linear. Sua saída varia, tipicamente, em 10 milivolts (mV) por grau Celsius (°C), e sua faixa de temperatura está entre -55 °C e 150 °C.

Os atuadores utilizados são: diodos emissores de luz, do inglês *light emitter diode* – LED; Resistores de fio; *cooler* e um *buzzer*.

O LED utilizado é o de alto brilho, na cor branca. Na polarização direta, a tensão entre seus terminais é de aproximadamente 3 V, com uma corrente direta de cerca de 25 mA.

O resistor de fio é uma resistência capaz de suportar altas temperaturas, tipicamente utilizado para aquecimento.

O cooler é um pequeno ventilador, tipicamente utilizado na refrigeração de processadores de computador. É alimentado com até 12 V<sub>DC</sub> e consome uma corrente de até 0,14 A. A velocidade de suas hélices é proporcional ao nível de tensão aplicado.

O *buzzer* é um pequeno alto falante de alta impedância.

### 3.1.4. Circuitos drivers

A placa 01 foi projetada para interligar o ADuc842 com os LEDs, os foto-transistores, os LDRs e o buzzer. Ela é alimentada por uma fonte DC de 5 V e demanda até 0,3 A. Seu esquema elétrico e circuito impresso estão ilustrados, respectivamente, nas Figuras 21 e 22.

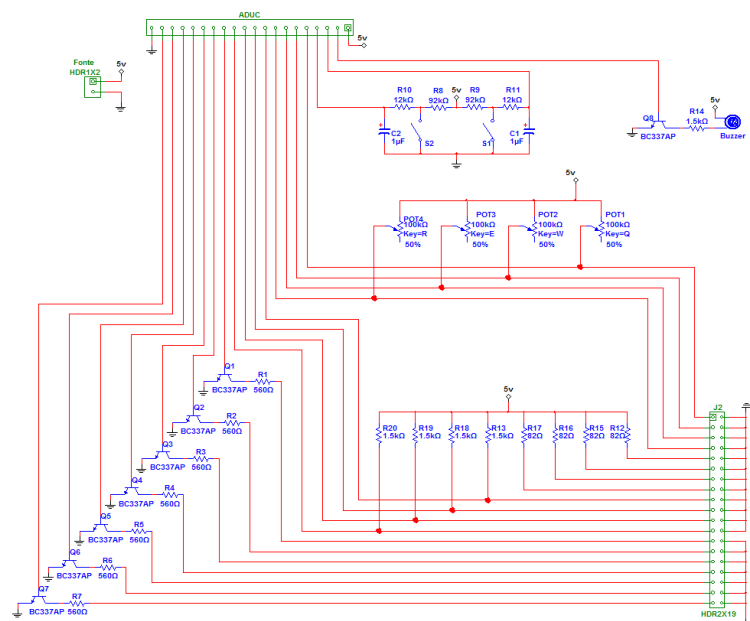
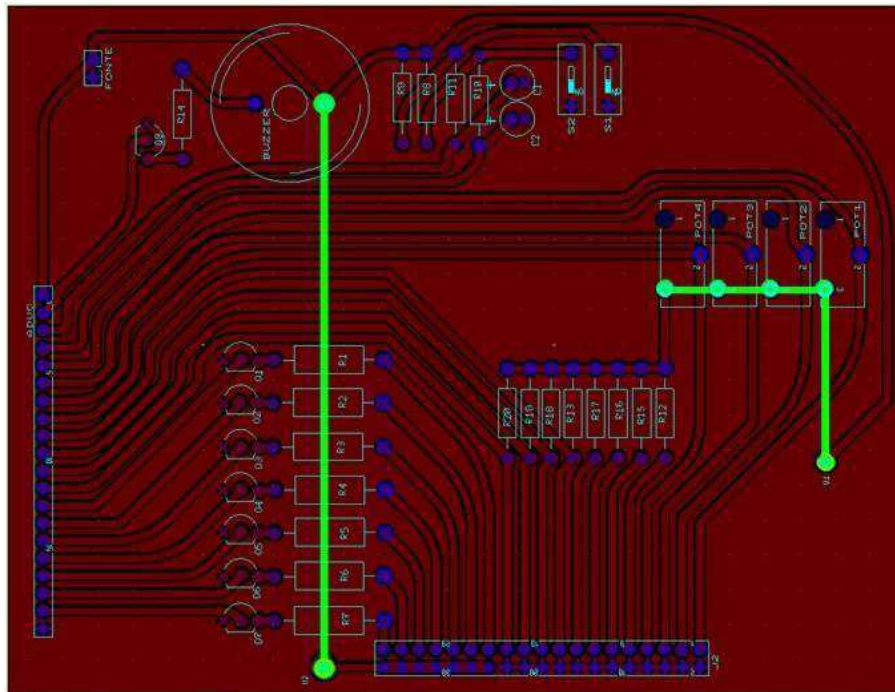


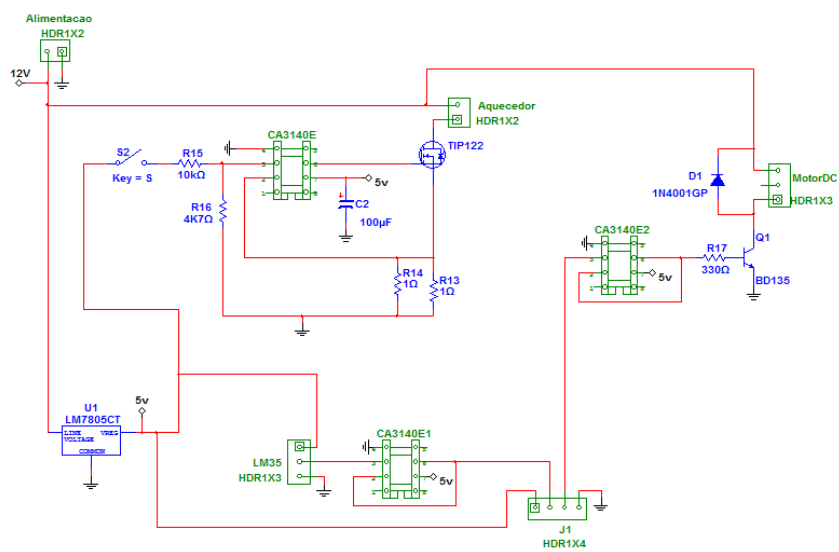
Figura 21: Esquema elétrico da placa 01.



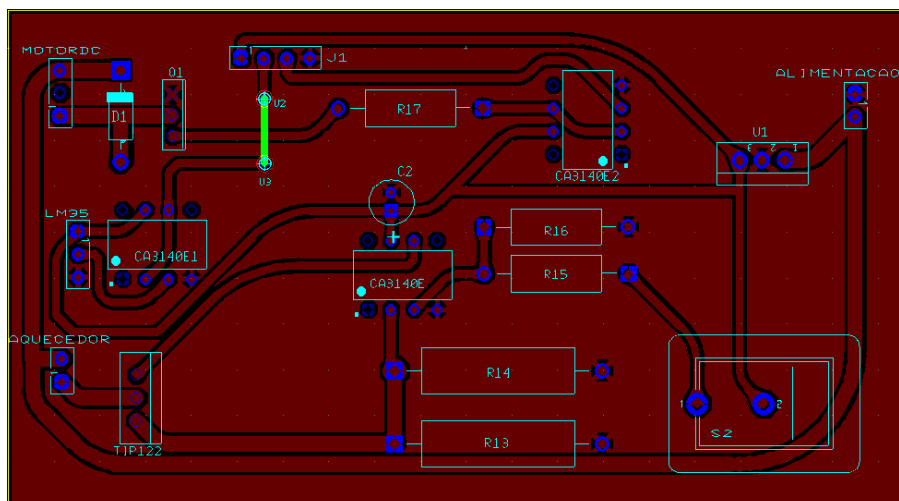


**Figura 22:** Esquemático do circuito impresso da placa 01.

A placa 02 interliga o ADuc842 ao LM35 e ao cooler e aciona as resistências aquecedoras independentemente. Ela é alimentada por uma fonte DC de 12 V e pode demandar até 3 A quando as resistências estão ligadas. O esquema elétrico, bem como o circuito impresso, estão, respectivamente, nas Figuras 23 e 24.



**Figura 23:** Esquema elétrico da placa 02.



**Figura 24:** Esquemático do circuito impresso da placa 02.

Numa terceira placa estão os conectores para alimentação das placas, bem como as chaves CH1 e CH2 da placa 01, e a chave Aquecedor da placa 02. Ambas as placas podem ser ligadas ao mesmo tempo. Pede-se atenção ao nível de corrente drenado pela placa 02 quando os resistores de fio estão ligados.

### 3.2. 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 (Tabela 02) como descrito em seguida.

**Tabela 02: 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 aduc 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 detalhado no datasheet do ADuC, 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 se obter o valor de baud rate desejado.

A Tabela 03 traz um resumo com os valores mais frequentes de baud rate e os valores necessários a serem armazenados nos registradores podem ser observados a seguir.

**Tabela 03:** Valores Para T3CON e T3FD e os baudrates selecionados

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 4 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 0 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 utilizar memória externa.

A Porta 1 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' no bit referente a porta. Alguns pinos dessa porta possuem função secundária, vide datasheet.

Já a Porta 2 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 0 no acesso à memória externa. Para ser usado como entrada digital é necessário antes escrever '1' no bit 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 3 é a que demanda maior atenção quando utilizada, pois todos os seus pinos possuem funções secundárias, como descrito no datasheet. Por exemplo, quando da comunicação serial RS-232, os pinos P3.0 e P3.1 são utilizados, independente das configurações prévias feitas no código. Seus pinos também possuem resistores internos e, para serem usados como entradas digitais precisam ter '1' escrito previamente.

Para trabalhar com os pinos individuais de cada porta, é necessário definir seus bits de registro devidamente endereçados. Os pinos da Porta 0 vão do endereço 80h até o 87h. Na Porta 1 os endereços são de 90h até 97h. Já a Porta 2 vai de A0h até A7h. E, por fim, os endereços da Porta 3 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

Para este experimento, será usada a placa 01, que deve 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. A Tabela 04 tem a indicação sobre em qual porta estará conectado cada dispositivo.

**Tabela 04:** Mapeamento dos dispositivos da planta nas portas do ADuC842.

Bit \ Porta	0	1	2	3
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.

## 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: JAIDILCON 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

//-----//
// Variáveis //

// 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

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

//-----//
// 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;

    printf("\n\nInicio\n\n");

    LS1 = 0;
    LS2 = 0;
    LC = 0;
    LQ1 = 0;
```

```

LQ2 = 0;
LE1 = 0;
LE2 = 0;

// ----- //
// Loop Infinito
while(1)
{
    // Controle das luzes nos corredores e sala
    if(SPS1 & SLS1)
    {
        LS1 = 1;
        LS2 = 0;
        LC = 0;
        printf("Luz Corredor 1 acesa\n");
    }
    if(SPS2 & SLS2)
    {
        LS1 = 0;
        LS2 = 1;
        LC = 0;
        printf("Luz Corredor 2 acesa\n");
    }
    if(SPC & SLC)
    {
        LS1 = 0;
        LS2 = 0;
        LC = 1;
        printf("Luz Sala acesa\n");
    }

    // Controle das luzes do quarto
    LQ1 = (CH2 & SLQ) | (CH2 & CH1);
    LQ2 = CH1 | CH2;

    if(LQ1)
    {
        printf("\nLuz Quarto 1 acesa");
    }

    if(LQ2)
    {
        printf("\nLuz Quarto 2 acesa");
    }

    // Controle das luzes externas
    LE1 = SPEX;
    LE2 = !SPEX;
} // fim do loop
} // fim do main

// -----//
// Funções //

// 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 seleccionados
    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;     |  |


    */
}

```

### 3.3. 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 referencia externa também pode ser utilizada (Figura 25), e seus valores devem compreender entre 2,3 V a  $AVDD$  (tensão de alimentação da parte analógica do chip do ADUC).

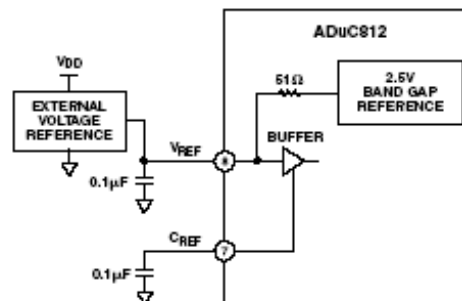
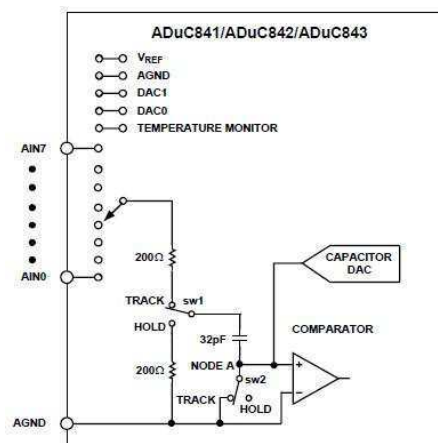


Figura 25: Utilização de tensão de referência externa para o conversor A/D



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 26, 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 26:** 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 (Tabela 05).

**Tabela 05:** Bits do registrador ADCCON1

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

MD0 e MD1 – Seleciona o modo de operação do conversor A/D (Tabela 06).

**Tabela 06:** 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 (Tabela 07). 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.

**Tabela 07:** Divisor para o *clock* mestre.

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 converso A/D ajustados em CK0 e CK1 (Tabela 08).

**Tabela 08:** Ajuste da velocidade de aquisição do conversor A/D

AQ1	AQ0	Número de períodos do clock do conversor A/D
0	0	1
0	1	2
1	0	4
1	1	8

T2C – (*Timer 2 Conversion Bit*) – este bit é setado pelo usuário para habilitar o bit de estouro do Timer 2 disparar a conversão, ou seja, o processo de conversão será acionado automaticamente a cada ciclo programado no Timer 2.

EXC – (*External Trigger Enable bit*) – este *flag* dispara o início da conversão A/D através de um sinal externo, que deve ser dado no pino 23 (CONVST – *Conversion Start*).

• **ADCCON2** – ADC Control SFR 2 – Seleciona o canal utilizado para a conversão e os modos de conversão (Tabela 09).

**Tabela 09:** Bits do registrador ADCCON2

ADCI	DMA	CCONV	SCONV	CS3	CS2	CS1	CS0
------	-----	-------	-------	-----	-----	-----	-----

ADCI – Interrupção por hardware, é setada quando um ciclo de conversão A/D é finalizado;

DMA – Bit de habilitação do modo de aquisição por DMA, deve ser setado pelo usuário quando necessário;

CCONV – (*Continuous Conversion*) este *flag* habilita o modo de conversão contínua, quando setado pelo usuário;

SCONV – (*Start Conversion*) é setado para iniciar o processo de conversão A/D;

CS3, CS2, CS1 e CS0 – (*Channel Select*) – seleciona o canal a ser utilizado na conversão A/D (Tabela 10).

**Tabela 10:** Seleção do canal para conversão A/D

CS3	CS2	CS1	CS0	Canal selecionado
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	1	Sensor de temperatura
1	1	1	1	Parada do DMA

• **ADCCON3** – ADC *Control* SFR 3 – sinaliza para o *firmware* em uso o estado do Conversor A/D (Tabela 11).

**Tabela 11:** Bits do registrador ADCCON3.

<b>BUSY</b>	<b>RSVD</b>	<b>RSVD</b>	<b>RSVD</b>	<b>RSVD</b>	<b>RSVD</b>	<b>RSVD</b>	<b>RSVD</b>
-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------

**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. Após concluído o ciclo de conversão, no ADUC 812 o valor convertido (12 bits) é armazenado em outros dois registradores SFR de oito bits chamados de ADCDATAH e ADCDATAL (Figura 27), 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.

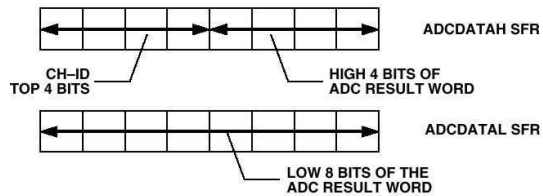


Figura 27: Armazenamento do valor convertido no ADCDATAH e ADCDATAL

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 28, de forma a eliminar ruídos de alta frequência, que possam produzir erros nas leituras.

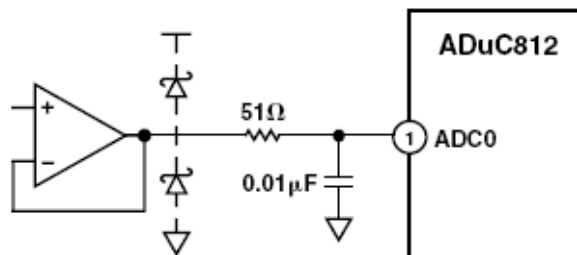


Figura 28: circuito de acoplamento e rejeição de ruídos

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 29).

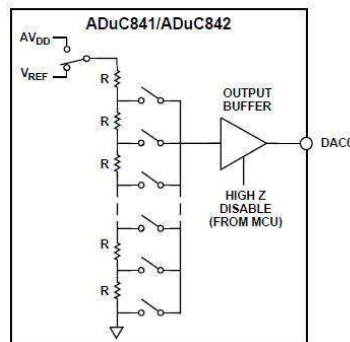


Figura 29: Estrutura interna do conversor D/A

A tensão de saída do conversor D/A pode ter seus valores variando de 0 V à VDD ou VREF, 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 (Tabela 11).

Tabela 11: Bits do registrador DACCON.

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 VREF / “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 SYC 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.

DACOL – 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 30, 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.

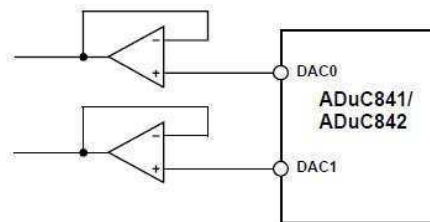


Figura 30: Utilizando buffer nas saídas do conversor D/A

## Bancada de experimento

Para este experimento, será usada a placa 02, que deve ser alimentada por uma fonte de tensão contínua de 12 V que forneça até 3,5 A. Também são utilizados 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 LM35 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.



## 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

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

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

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

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

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

    printf("\n\nInicio\n\n");

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

        tensaoAD = lm35*0.122100122100;

        if(tensaoAD<40)
            cooler = 0;
        else
            cooler = 6*lm35 + 1635;

        if(tensaoAD > 50)
            cooler = 4095;

        escrita_da(cooler);

        tensaoDA = cooler*0.122100122100;

        printf("tensao AD = %.3f mV\t",tensaoAD);

        printf("tensao DA = %.3f mV\n",tensaoDA);

        i = 0;
        while(i<=200000)
            i++;
    }
}
```

```

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

//-----//
// Funções //

// 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)
{
    // Configuração 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 += ADCDATAL | ((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);
}

```

### 3.4. 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 microcontroladores 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 à 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 31 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.

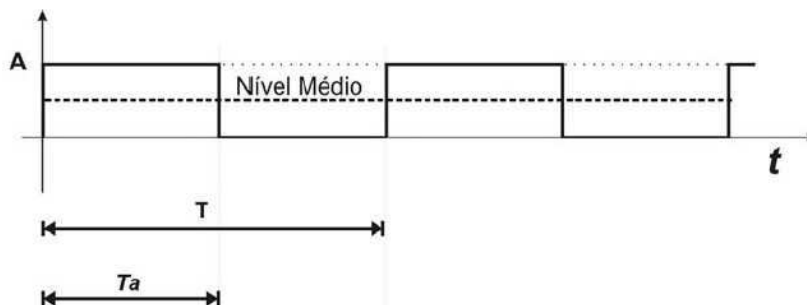


Figura 31: Representação de uma onda quadrada

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

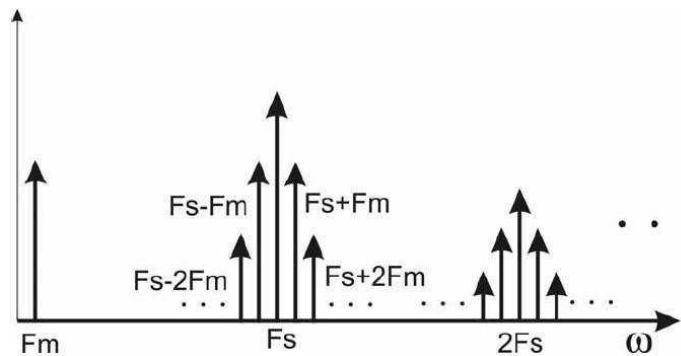
$$NivelMédio = A \cdot \frac{Ta}{T} \quad (3.2)$$

Ou então:

$$NivelMédio = A \cdot DutyCycle \cdot 100\% \quad (3.3)$$

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.

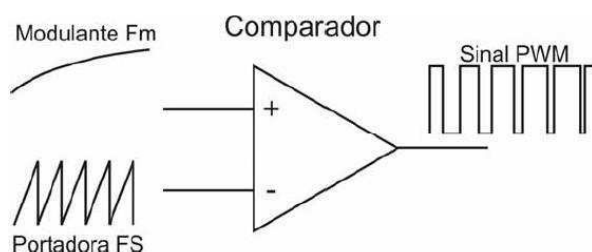
O espectro do sinal PWM (Figura 32) é formado pelas seguintes raia: Raia Fundamental  $F_m$  (modulante), Raia da portadora  $F_s$  (frequência de amostragem PWM), os harmônicos da fundamental ao redor da portadora e os harmônicos da portadora. De acordo com o espectro apresentado na Figura 32 o sinal da modulante é recuperado através de uma filtragem passa baixa sintonizada para selecionar apenas a raia  $F_m$ .



**Figura 32:** 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.

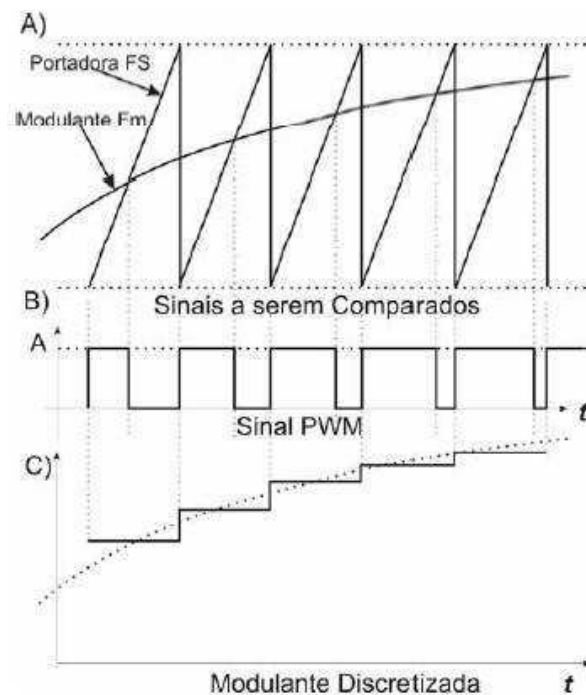
O processo de geração de um sinal PWM é baseado em uma comparação de níveis de valores. Uma tensão de referência de comportamento linear em relação ao tempo deve ser produzida, de forma que sirva de referência para um comparador. Como o sinal PWM possui apenas dois níveis de amplitude, seus valores serão 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 33 ilustra-se a forma como a comparação entre os dois sinais é feita para obter-se o sinal PWM.



**Figura 33:** Comparação dos sinais  $F_m$  e  $F_s$ , produzindo um sinal PWM

Utilizando o diagrama da Figura 33 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 34 A);
- O sinal PWM (Figura 34 B) é produzido em sua saída com os níveis médios variando a cada amostragem (Figura 34 C);



**Figura 34:** 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 35).

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  $2n$  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  $2n$  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 34), 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.

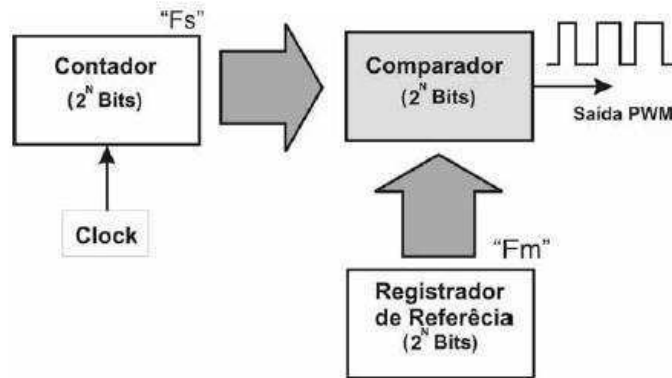


Figura 35: Diagrama interno de um modulador PWM digital

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.

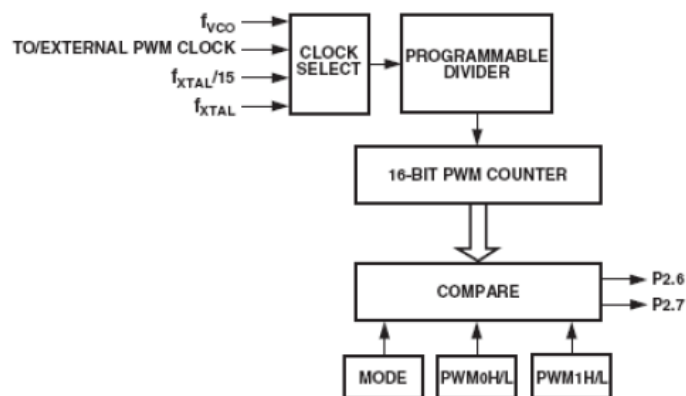
No hardware para geração do sinal PWM, o clock da contagem pode ser configurado de várias formas, onde é possível selecionar a fonte do sinal e a divisão de frequência. No microconversores ADUC as fontes de clock para o contador podem ser: o clock do cristal (ex: 11.0592 MHz), o clock do cristal dividido 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 36) 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 36:** 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 (Tabela 12).

**Tabela 12:** Bits do registrador PWMCON

SNGL	MD2	MD1	MD0	CDIV1	CDIV0	CSEL1	CSEL0
------	-----	-----	-----	-------	-------	-------	-------

**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 (Tabela 13);



**Tabela 13:** Seleção do modo de operação do PWM

MD2	MD1	MD0	Modo de Operação Selecionado
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 (Tabela 14);

**Tabela 14:** Seleção do fator de divisão do clock

CDIV1	CDIV0	Divisão do Clock Selecionado
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 (Tabela 15);

**Tabela 15:** Seleção do clock pro PWM

CSEL1	CSEL0	Fonte de Clock para o Contador PWM
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.

## 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>

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

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

//-----//
// 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

    printf("\n\nInicio\n\n");

    // ----- //
    // Loop Infinito
    while (1)
    {
        if (CH1 == 0)
        {
            PWM1L = 0x3D;
            PWM1H = 0x25; // PWM1H/L = 9533
                        // Frequência do PWM = (16.777216 MHz/16/9533) = 110 Hz

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

                printf("\nFreq = 110 Hz  Duty = 25");
            }
            else
            {
                PWM0L = 0xED; // DC = 75%
                PWM0H = 0x1B; // PWM0H/L = PWM1H/L * 0.75

                printf("\nFreq = 110 Hz  Duty = 75");
            }
        }
    }
}
```

```

else
{
    PWM1L = 0x9E;
    PWM1H = 0x12; // PWM1H/L = 4766
                // Frequência do PWM = (16.777216 MHz/16/4766) = 220 Hz

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

        printf("\nFreq = 220 Hz  Duty = 25");
    }
    else
    {
        PWM0L = 0xF6; // DC = 75%
        PWM0H = 0x0D; // PWM0H/L = PWM1H/L * 0.75

        printf("\nFreq = 220 Hz  Duty = 75");
    }
}
} // fim do loop
} // fim do main

//-----//
// Funções //

// 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 = 0x9B; // Mode 1
                // PWM Counter = Selected Clock/16
                // PWM Clock = 16.777216 MHz
}

```

### 3.5. Guia do Experimento 4 - Temporizadores e interrupções

#### Objetivos

Verificar o funcionamento dos temporizadores para contagem de tempo e dos sistemas de interrupções do ADuC842.

#### **TIC - *Time Interval Counter***

O TIC (Time Interval Counter - Contador de Intervalo de Tempo) é um sistema composto por alguns contadores e registradores conectados em série de forma a contar intervalos de tempo. Com a utilização de um cristal de 32,768 KHz tem-se uma base de tempo precisa, assim pode-se assumir então que o circuito de TIC está associado a um RTC (Real Time Clock – Relógio de Tempo Real), onde o tempo é contado em registradores de segundos, minutos e horas.

A faixa de tempo contada pelo TIC vai de 1/128 segundos a 255 horas, de forma que o usuário pode definir em intervalos de tempo precisos a ocorrência de alguma ação em sua aplicação. Conforme ilustrado na Figura 37, o clock de 32,768 KHz é aplicado ou não aos contadores, dependendo do valor do TCEN (Timer Counter Enable Bit – Bit de habilitação do TIC). O clock então é dividido por 28 no Prescaler de 8 bits (divisor de frequência). Após o Prescaler, um clock com período de 1/128 segundos, faz o registrador contador de Centésimos de segundos (HTHSEC), ser incrementado. Este contador também é um registrador SFR que pode ser acessado a qualquer momento pelo usuário através do firmware. Apesar de não registrar exatamente os centésimos de segundos, os valores de contagem do registrador SFR HTHSEC vão de 0 a 127. A cada estouro de contagem de HTHSEC, que ocorre exatamente a cada segundo, um pulso de clock de mesmo período é produzido para o contador registrador seguinte. O registrador SFR SEC registra então os segundos da contagem e seus valores são de 0 a 59. Da mesma forma a cada estouro da contagem do registrador SFR SEC, um pulso de clock de 60 segundos de período fará o registrador MIN (minutos) ser ativado. O registrador MIN produz então um pulso de clock a cada 60 minutos para incrementar a contagem do registrador contador HOUR. O registrador SFR HOUR faz a contagem de 0 a 23 ou de 0 a 244, dependendo do que o usuário ajustar no TIMECON (TIC Control Register – Registrador de controle do TIC).

Os registradores SFR HTHSEC, SEC, MIN e HOUR, têm seus valores incrementados a cada 1/128 de segundo, segundo, minuto e hora respectivamente, de forma que o usuário dispõe de uma contagem precisa de intervalos temporais. Através do MUX de seleção da base de tempo um clock de 1/128 de segundo, um segundo, um minuto ou uma hora pode ser escolhido para incrementar o contador de intervalo de 8 bits propriamente dito.

Uma interrupção será gerada para o sistema, quando o valor ajustado pelo usuário no registrador SFR de 8 bits INTVAL, for igual ao valor do contador de intervalo. Desta forma uma interrupção com um valor de tempo bem definido pode ficar sendo gerada.

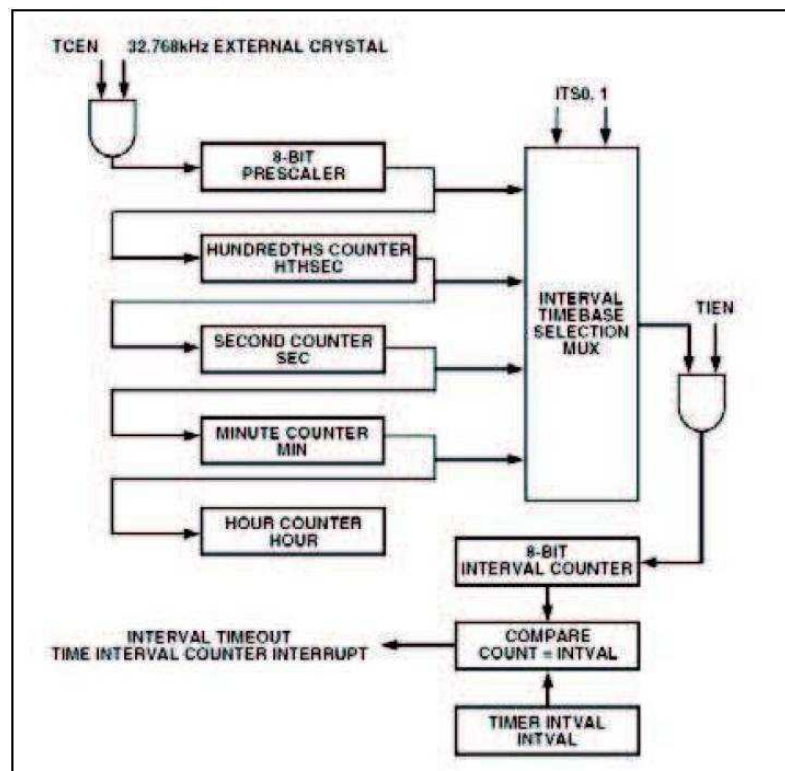


Figura 37: Diagrama de blocos simplificado do TIC

Todo o procedimento de ajuste de funcionamento do TIC é feito no registrador SFR TIMECON que possui o seguinte arranjo (Tabela 15):

- TIMECON – TIC Control SFR.

Tabela 15: Bits do registrador TIMECON

RSVD	TFH	ITS1	ITS0	STI	TII	TIEN	TCEN
------	-----	------	------	-----	-----	------	------

**RSVD** – Reservado para uso futuro;

**TFH** – Twenty-Four Hour Select Bit – Bit de seleção para a contagem do registrador FR HOUR de 0 a 23 (TFH = “1”) ou de 0 a 255 (TFH = “0”);

**ITS1 e ITS0** – Interval Timebase Selection – Bits para a seleção da base de tempo para o contador de intervalo (Tabela 16);

**Tabela 16:** Seleção da base de tempo do temporizador.

ITS1	ITS0	Base de tempo selecionada para o TIC
0	0	1/128 segundo
0	1	Segundos
1	0	Minutos
1	1	Horas

**STI** – Single Time Interval Bit – Bit para a seleção da contagem de apenas um único intervalo de tempo. Quando em nível lógico “1”, o TIC gerará interrupção de apenas uma contagem, pois o bit **TIEN** é zerado. Quando o bit **STI** é zerado a interrupção do TIC é dada a cada intervalo de tempo programado, onde o contador de intervalo é recarregado com o valor programado e uma nova contagem é iniciada;

**TII** – TIC Interrupt Bit – Bit de interrupção do TIC – Seu valor é igual a “1” quando o contador de intervalo tem seu valor igual ao registrador SFR **INTVAL**. O software do usuário deve zerar o bit **TII** após o atendimento à interrupção;

**TIEN** – TIC Bit Enable – Bit de habilitação do TIC. Quando **TIEN** = “0” o TIC fica desabilitado, quando **TIEN** = “1” o TIC está habilitado (Figura 1);

**TCEN** – Time Clock Enable Bit – Bit de habilitação do clock para o TIC e o RTC. Quando o **TCEN** está em “1” o clock será direcionado para os contadores fazendo com que o TIC funcione normalmente. Porém quando em nível lógico “0”, além do clock ficar desabilitado, os registradores SFR **HTHSEC**, **SEC**, **MIN** e **HOUR** (Tabela 17) serão

carregados com os últimos valores ajustados pelo programa da aplicação. Os registradores SFR do TIC podem ter seus valores alterados quando TCEN = "0".

**Tabela 17:** Registradores relacionados com o temporizador

Registrador	Função	Faixa de Contagem
INTVAL	Seleção do intervalo contado pelo TIC	
HTHSEC	Incrementado a cada 1/128 segundos	0 a 127
SEC	Incrementado a cada segundo	0 a 59
MIN	Incrementado a cada minuto	0 a 59
HOUR	Incrementado a cada hora	0 a 23 (THF = "1") 0 a 255 (THF = "0")

## Sistema de interrupções

O ADuC842 possui 9 sistemas de interrupções, que podem assumir dois níveis de prioridade, e são configurados através dos três registradores SFRs seguintes:

- IE – Interrupt Enable Register
- IP – Interrupt Priority Register
- IEIP2 – Secondary Interrupt Enable Register

O registrador IE permite habilitar, ou desabilitar as interrupções desejadas. Já o registrador IP determina o nível de prioridade ('0' ou '1') de cada interrupção. Por fim, o registrador IEIP2 complementa os dois registradores anteriores.

**Obs.:** Consultar o manual do ADuC842 para detalhamento dos bits de cada registrador.

As interrupções com prioridade '1' são resolvidas primeiro e podem parar a execução de uma de prioridade '0'. Caso duas interrupções de mesma prioridade ocorram simultaneamente, a de maior hierarquia será resolvida primeiro. A Tabela 18 traz o nível de hierarquia entre as interrupções.

**Tabela 18:** Hierarquia das interrupções do ADuC842

Source	Priority	Description
PSMI	1 (Highest)	Power Supply Monitor Interrupt.
WDS	2	Watchdog Timer Interrupt.
IE0	2	External Interrupt 0.
ADCI	3	ADC Interrupt.
TF0	4	Timer/Counter 0 Interrupt.
IE1	5	External Interrupt 1.
TF1	6	Timer/Counter 1 Interrupt.
ISPI/I2CI	7	SPI Interrupt/I <sup>2</sup> C Interrupt.
RI + TI	8	Serial Interrupt.
TF2 + EXF2	9	Timer/Counter 2 Interrupt.
TII	11(Lowest)	Time Interval Counter Interrupt.

Para programar a rotina de uma interrupção é necessário saber qual o seu vetor de endereço para, com este valor, definir qual o número de interrupção para a função de correspondente. A lista desses endereços está no manual do microcontrolador.

A plataforma  $\mu$ Vision possui uma tabela de valores próprias pros microcontroladores da família 8051, ao qual está incluso o ADuC842. Esta tabela está replicada na Tabela 19. A sintaxe para a função de interrupção está demonstrada no código exemplo ao final deste documento.

**Tabela 19:** Número de interrupção em função do seu endereço.

Número de Interrupção	Endereço da Interrupção	Número de Interrupção	Endereço da Interrupção
0	0003h	16	0083h
1	000Bh	17	008Bh
2	0013h	18	0093h
3	001Bh	19	009Bh
4	0023h	20	00A3h
5	002Bh	21	00ABh
6	0033h	22	00B3h
7	003Bh	23	00BBh
8	0043h	24	00C3h
9	004Bh	25	00CBh
10	0053h	26	00D3h
11	005Bh	27	00DBh
12	0063h	28	00E3h
13	006Bh	29	00EBh
14	0073h	30	00F3h
15	007Bh	31	00FBh



## Bancada de experimento

O experimento usará a placa 01, que deve ser alimentada por uma fonte de tensão contínua de 5 V. Os LEDs e o sensor de presença da região externa da planta, e o sensor de presença SPS1 da região interna serão usados neste experimento.

O sensor SPEX está conectado ao pino 2 da Porta 3, para gerar a interrupção externa INT0 do ADuC842. Esta interrupção dará início a contagem do temporizador que realiza suas ações a cada intervalo de tempo. O sensor SPS1 desliga as ações do temporizador.

## 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 4: Interrupções

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

//-----//
// Variáveis //
sbit P3_4 = 0xB4; // Led na placa de desenvolvimento
sbit LE1 = 0xA5; // P2.5 -> Luz externa 1
sbit LE2 = 0xA4; // P2.4 -> Luz externa 2
sbit SPS1 = 0x82; // P0.2 -> Sensor Presença - Sala 1

//-----//
// Protótipos de Funções //
void configura_PWM(void);

//-----//
// Programa Principal //
void main (void)
{
    PLLCON = 0x88; // Clock de 16.77 MHz
    CFG842 = 0x60; // Transfere PWM pra porta 3.

    IT0 = 1;
    EX0 = 1;
    EA = 1; // Habilita interrupções

    LE1 = 1;
    LE2 = 1;

    P3_4 = 1;

    SPS1 = 1;
```

```

// ----- //
// Loop Infinito
while (1)
{
    if(SPS1 == 1)
    {
        IEIP2 = 0x40;    // desabilita interrupções do TIC
        PWMCON = 0x8B; // PWM - modo 0 (desligado)
    }

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

// ----- //
// Funções //

// Configuração da saída PWM
void configura_PWM(void)
{
    PWMCON = 0x9B; // Mode 1
                // PWM Counter = Selected Clock/16
                // PWM Clock = 16.777216 MHz
}

// ----- //
// Interrupções //

// Interrupção INT0
void INT_0 () interrupt 0
{
    P3_4 = 0;

    configura_PWM();

    //Configura o TIC
    TIMECON = 0x13; // Interrupção TIC a cada "INTVAL"
    INTVAL = 0x01; // 1 segundo
    IEIP2 = 0x44; // habilita interrupções do TIC
}

// Interrupção TIC
void TIC_int () interrupt 10
{
    PWM1L = 0x9E;
    PWM1H = 0x12; // PWM1H/L = 4766
                // Frequência do PWM = (16.777216 MHz/16/4766) = 220 Hz

    PWM0L = 0x4F;
    PWM0H = 0x9; // DC = 50%

    LE1 = !LE1;
    LE2 = !LE2;

    P3_4 = 1;
}

```

## **4. Conclusão**

Neste estágio realizou-se para o Laboratório de Instrumentação e Controle (LIEC), a concepção e construção de uma bancada de experimentos, inspirada em um sistema de automação residencial, para o estudo das técnicas de instrumentação eletrônica, em especial a disciplina Sistemas de Aquisição de Dados e Interface deste curso de graduação. Além disso, foram elaboradas atividades experimentais a serem realizadas pelos alunos matriculados nesta disciplina.

Para a idealização da bancada foi necessário um conhecimento sobre o projeto de circuitos eletrônicos, oferecidos pela graduação de Engenharia Elétrica da UFCG através de disciplinas tais como Dispositivos Eletrônicos, Eletrônica e Instrumentação Eletrônica. Para a criação das atividades experimentais propostas utilizaram-se conceitos sobre as características de um microcontrolador e sobre programação. As disciplinas desta graduação, tais como Arquitetura de Sistemas Digitais, Técnicas de Programação e a própria Sistemas de Aquisição de Dados e Interface oferecem ao aluno estes conceitos necessários. O que demonstra, em pequena escala, a importância das teorias e técnicas oferecidas ao aluno durante seu curso de graduação.

## 5. Referências Bibliográficas

- [1] <http://liec.ufcg.edu.br/Default.cshtml> Acessado em 01 de dezembro de 2015;
- [2] <https://sites.google.com/a/dee.ufcg.edu.br/cgee/curriculo> Acessado em 01 de dezembro de 2015;
- [3] MicroConverter 12-Bit ADCs and DACs with Embedded High Speed 62-kB Flash MCU. 2003 Analog Devices, Inc.
- [4] <http://www.keil.com/uvision/> Acessado em 18 de maio de 2015;
- [5] Guias experimentais. Disciplina Sistemas de Aquisição de Dados e Interface. Jaidilson Jó da silva.