

UniCEUB - Centro Universitário de Brasília
Faculdade de Tecnologia e Ciências Sociais Aplicadas –
FATECS
Projeto Final

**Utilização de células solares para controle de luminosidade
de ambientes**

Paulo Alexandre Agueda Gobbato
RA: 2041680/3

Monografia Final do Curso de Engenharia de Computação
Orientador: José Julimá Bezerra Junior

Brasília – DF, Novembro de 2009

Utilização de células solares para controle de luminosidade de ambientes

Projeto de conclusão do curso de Engenharia de Computação da Faculdade de Tecnologia e Ciências Sociais Aplicadas, para obtenção do título de Bacharel em Engenharia de Computação, a ser fornecido pelo Centro Universitário de Brasília – UniCEUB.

Orientador: José Julimá Bezerra Junior

Brasília – DF
Novembro de 2009

AGRADECIMENTOS

Ao meu orientador, Professor José Julimá, pela paciência em aguardar resultados.

A minha família, por ter me colocado neste caminho.

Aos meus amigos Henrique, Greg, Gustavo, Rodrigo e Diego pela imensa ajuda e paciência que tiveram.

A Mayra, pelas dezenas de horas de calma e dezenas de idéias que me trouxe.

RESUMO

O presente projeto aborda o desenvolvimento de um protótipo de sistema que tem por objetivo regular automaticamente a intensidade da iluminação de um ambiente com acesso a luz natural. O sistema utiliza uma célula solar como sensor de luminosidade, dois microcontroladores Atmel AT89S8253, para automatizar as tarefas e gerar PWM, além de um ADC0804 para tornar legíveis as variações da célula solar.

Palavras chave: Iluminação, célula solar, PWM, Microcontrolador, LED.

ABSTRACT

This project aims to build a system prototype whose objective is to automatically regulate the intensity of lighting in a room with incoming natural lighting. The system uses two ATMEL AT89s8253 microcontrollers to automate the tasks and generate PWM pulses, and a ADC0804 to convert the data from the solar cell.

Key words: Lighting, solar cell, PWM, Microcontroller, LED.

LISTAS DE FIGURAS

Figura 1 - Esquema do Microcontrolador AT89S8253	17
Figura 2 - Células solares policristalinas de uso comercial	18
Figura 3 - Esquema do Conversor A/D ADC0804.....	19
Figura 4 - Representação de pulsos PWM	21
Figura 5 - Display LCD JHD 162A	22
Figura 6 - Esquema de ligação da fonte 12v	24
Figura 7 - Detalhe das ligações do ADC0804.....	26
Figura 8 - Detalhe das ligações do LCD	27
Figura 9 - Esquema de ligação de um botão	28
Figura 10 - Circuito de chave com transistor	29
Figura 11 - Placa PWM	30
Figura 12 - Placa de LEDs	31
Figura 13 - Esquema base de funcionamento do protótipo.....	32
Figura 14 - Esquema de ligação da placa de controle	61
Figura 16 - Pinos do microcontrolador AT89S8253.....	62
Figura 17 - Pinos do Conversor Analógico-Digital	63
Figura 18 - Pinos e dimensões do módulo JHD 162A.....	64

LISTA DE ABREVIATURAS E SIGLAS

A/D – Analógico / Digital

ADC - Conversor de Analógico para Digital

CAT 5e – Cabo de rede categoria 5e

CI – Circuito integrado

ISP – *In system Programming*

LCD – *Liquid Cristal Display*

LED – *Light Emiting Diode*

LSB – *Least Significant Bit*

MCU – Microcontrolador

MSB – *Most Significant Bit*

PWM – *Pulse Width Modulation*

uC – Microcontrolador

LISTA DE TABELAS

Tabela 1 – Luminosidade do protótipo.....	43
Tabela 2 – Variação de tensão e valores respectivos ao microcontrolador.....	44
Tabela 3 – Comparação com uma lâmpada comum.....	45

SUMÁRIO

AGRADECIMENTOS	III
RESUMO.....	IV
ABSTRACT.....	V
LISTAS DE FIGURAS	VI
LISTA DE ABREVIATURAS E SIGLAS	VII
LISTA DE TABELAS.....	VIII
SUMÁRIO	IX
1. INTRODUÇÃO.....	11
1.1. <i>Motivação</i>	11
1.2. <i>Apresentação do problema</i>	12
1.2.1. Identificação do Problema.....	12
1.2.2. Proposta de solução	12
1.3. <i>Objetivo Geral do Trabalho</i>	12
1.4. <i>Objetivos específicos</i>	12
1.5. <i>Justificativa e relevância do tema</i>	13
1.6. <i>Estrutura do Projeto</i>	13
1.7. <i>Resultados esperados</i>	14
2. REFERENCIAL TECNOLÓGICO	15
2.1. <i>Microcontroladores 8051</i>	15
2.1.1. ATMEL AT89S8253.....	16
2.2. <i>Células Solares</i>	17
2.3. <i>Luz e o conceito de Dia e Noite para um microcontrolador</i>	18
2.3.1. ADC 0804	19
2.4. <i>PWM (Pulse Width Modulation)</i>	20
2.5. <i>Chipset Hitachi HD 44780</i>	21
2.5.1. Módulo LCD JHD 162A.....	22
3. DESENVOLVIMENTO DO PROJETO.....	23
3.1. <i>Preparação do hardware</i>	23
3.1.1. Fontes de 12 e 5v.....	24
3.1.2. Controle do protótipo.....	25
3.1.3. PWM.....	28
3.1.4. Placas de iluminação com LEDs	30
3.2. <i>Rotinas do software</i>	32
3.2.2. Escrita no LCD	34
3.2.3. Conversão Analógica / Digital.....	37
3.2.4. Leitura dos botões	38
3.2.5. Cálculo da necessidade de iluminação	39
3.2.6. Tempos para PWM	40
3.2.7. Interrupção para PWM.....	41
3.2.8. Gravação do software no microcontrolador.....	41
4. RESULTADOS OBTIDOS.....	43
4.1. <i>Análise do uso e sensibilidade do sensor de luz</i>	44
4.2. <i>Análise do funcionamento do PWM</i>	44

4.3. Informações do LCD e aquisição de entrada através de botões	46
5. CONSIDERAÇÕES FINAIS	47
5.1. Dificuldades encontradas.....	47
5.2. Conclusão.....	48
5.3. Propostas futuras.....	49
REFERÊNCIAS BIBLIOGRÁFICAS.....	51
APÊNDICES.....	54
APÊNDICE A - CÓDIGO DO MICROCONTROLADOR:	54
APÊNDICE B - LIGAÇÃO DA PLACA DE CONTROLE:.....	61
ANEXOS	62
ANEXO A – CONFIGURAÇÃO DOS PINOS DO AT89S8253	62
ANEXO B – CONFIGURAÇÃO DOS PINOS DO ADC0804	63
ANEXO C – CONFIGURAÇÃO DOS PINOS DO MÓDULO DE LCD JHD162A	64

1. INTRODUÇÃO

Iluminação de ambientes é fator principal quando se coloca em mesa a qualidade de vida que o ambiente utilizado proporciona ao habitante. Iluminação mal direcionada, ou de intensidade fora da devida, pode causar sérios problemas a quem frequenta o local. [PRODEPA, 2009]]

Iluminação, ao menos em ambientes comerciais tem, por norma, obrigatoriamente que ser calculada para manter um nível mínimo de luminosidade média em toda a área compreendida. [LOPES, ZANLORENZI, COUTO e MINETTI, 2004]

Com a resolução definida pela norma, pode-se garantir ao utilizador do ambiente melhores condições de trabalho e produtividade, além de garantir que sua qualidade de vida não será afetada.

No entanto muitas empresas, seja por falta de investimento ou conhecimento, não controlam o nível de luminosidade no ambiente.

1.1. Motivação

A falta de iluminação pode ser ponto principal na avaliação de ergonomia de um ambiente de trabalho, além de poder se tornar uma grande fonte de tensão e desconforto aos utilizadores, ao invés de ser uma ferramenta útil para melhoria do desempenho ou qualidade de vida. [LOPES, ZANLORENZI, COUTO e MINETTI, 2004]

Em vários ambientes são encontradas luzes ligadas o dia todo para evitar esta fadiga desnecessária, no entanto isso leva a gastos excessivos com energia. Pode gerar ainda excesso de luminosidade, incomodando quem estiver presente no ambiente.

Muitos dos controles automatizados existentes são baseados em horários predeterminados, os quais requerem manutenção a cada horário de verão, além de não levar em conta a quantidade de luz incidida no momento.

1.2. Apresentação do problema

1.2.1. Identificação do Problema

Um dos principais fatores que levam empresas a serem multadas pelos órgãos de fiscalização de ambiente de trabalho é a iluminação existente, muitas vezes precária. [PRODEPA, 2009]

Além disso, um dos principais gastos de um edifício é com iluminação. As empresas têm funcionários responsáveis por apagar e acender a iluminação do edifício, no entanto esse controle não é feito de maneira eficiente. A luz é acesa e apagada em horários predeterminados, normalmente por uma chave geral, e não por um controle de luminosidade. [SOUZA, 2002]

O controle quando feito dessa maneira não equaliza a iluminação necessária com a luminosidade do ambiente no momento. Também não aproveita o máximo da luminosidade natural, pois ela varia de acordo com o tempo, não sendo fixa em horários determinados.

1.2.2. Proposta de solução

A solução descrita neste projeto é a utilização de um controle que não requer entrada de dados manual, exceto pela personalização da luminosidade desejada. Assim funcionando praticamente independente de contato humano, sendo este necessário apenas para manutenção.

1.3. Objetivo Geral do Trabalho

Controlar, através de uma unidade automatizada, a iluminação interna e externa de um ambiente, além de controlar a quantidade de iluminação fornecida para o ambiente de acordo com a iluminação natural provida pelo sol.

1.4. Objetivos específicos

Verificar a luminosidade do ambiente para, com este parâmetro, definir se é dia ou noite. Dependendo do status do parâmetro de luminosidade acender ou não a iluminação externa.

Verificar a luminosidade do ambiente para, com este parâmetro definir qual a quantidade de luz necessária para atender a norma NBR 5413/92.

1.5. Justificativa e relevância do tema

Este tipo de estudo sobre controle de iluminação é relevante, pois não apenas mede a luminosidade do ambiente e acende uma luz, mas sim verifica qual a necessidade real de iluminação do ambiente.

Tomando-se a premissa de que todo o ambiente controlado utiliza algum tipo de iluminação externa, seja por clarabóias, janelas ou tijolos de vidro, pode-se determinar a luminosidade atual e sua necessidade para atender a norma.

Este controle deve ser aplicado principalmente em ambientes de alta incidência de luz, pois permite que a luminosidade externa seja utilizada até o máximo, sem consumo extra de energia elétrica.

1.6. Estrutura do Projeto

O corpo da monografia foi distribuído da seguinte maneira:

Capítulo 1 – Este é o capítulo de introdução ao projeto, nele é abordada a motivação, os objetivos, a justificativa da escolha deste tema e o resultado esperado para o fim do projeto. Também é apresentado o problema e uma proposta de solução

Capítulo 2 – Compõe o referencial tecnológico, no qual são apresentados os conceitos básicos necessários para entendimento do projeto como um todo.

Capítulo 3 – Este é o capítulo que trata do desenvolvimento propriamente dito do projeto. É discutida a montagem do hardware e a criação do software.

Capítulo 4 – Demonstração dos resultados obtidos com testes do protótipo finalizado.

Capítulo 5 – Capítulo de conclusão da monografia, onde são apreciadas as considerações sobre o projeto realizado e suas possíveis melhorias e futuro.

1.7. Resultados esperados

O resultado deste projeto será um protótipo funcional com as funções de controlador automático de iluminação relativo à iluminação necessária.

Espera-se também que o protótipo proporcione a melhorara da qualidade de vida dos utilizadores dos ambientes, além de, se possível, reduzir os gastos mensais das empresas com relação à energia e manutenção dos sistemas de iluminação.

2. REFERENCIAL TECNOLÓGICO

Abaixo são descritos os componentes e conceitos chave deste projeto. Este tópico visa deixar o leitor a par de tudo que é necessário para entender o funcionamento do protótipo.

2.1. Microcontroladores 8051

Um processador e um microcontrolador diferem principalmente por suas funcionalidades. O processador precisa de componentes externos para seu funcionamento, assim como uma placa mãe e memórias, enquanto o microcontrolador embarca em um só pacote todos os componentes necessários para seu funcionamento. [CADT]

Um microcontrolador pode ser considerado um computador em um chip. Mesmo de tamanho tão reduzido realiza todas as funções de um computador, porém com menor desempenho. [SICA, 2006] Eles têm sua principal utilização em projetos dedicados que não tem requerimento de grande volume de armazenamento ou processamento e ainda requerem baixo custo de energia. Mesmo tendo pouco poder, comparado a outras máquinas, são versáteis por abarcar variadas aplicações com baixo custo.

O microcontrolador é projetado para executar certas funções específicas, portanto é necessário dar a ele ordens de funcionamento. Essas ordens, no entanto dependem de estar em conformidade com suas instruções básicas. Essas instruções são dadas normalmente em Assembly, uma linguagem de programação que se aproxima muito da máquina. No entanto, o programador, pessoa que definirá o código utilizado, deve ter conhecimento da linguagem, pois tem de fazer com que o microcontrolador trabalhe de maneira específica.

Por mais que sejam úteis e versáteis, tanto microcontroladores quanto processadores esbarram em uma limitação de desempenho: o clock. O clock é um relógio interno, que limita o número de instruções que podem ser realizadas por certa quantidade de tempo. Quanto maior o clock, mais operações podem ser feitas em um mesmo espaço de tempo. Vale lembrar que no caso de uma aplicação que

requeira altíssimo desempenho pode-se considerar utilizar hardware de lógica fixa. Este, no entanto não demonstra tanta confiabilidade como os dois citados anteriormente. [NICOLSI, 2000]

As características dos microcontroladores 8051 atuais são:

- Utilização de memória EPROM para armazenamento de código;
- Utilização da biblioteca de instruções MCS-51, que contém muitas instruções complexas, fazendo com que a programação do controlador seja simplificada;
- Baixo custo de produção, o que reduz o gasto com fabricação de produtos neles baseados;
- Pode ser programado principalmente em Assembly, C e Basic;
- Hoje, várias versões do 8051 contam com compatibilidade com ISP – In System Programmer – o que torna a gravação de códigos fácil, pois o código pode ser gravado pela porta serial.

2.1.1. ATMEL AT89S8253

O microcontrolador AT89S8253, da ATMEL, é um microcontrolador oito bits de baixa voltagem e é compatível com o padrão MCS-51 de instruções, desenvolvido para toda a família 8051 pela Intel. Suas especificações incluem 2K bytes de memória EPROM, 12K bytes de memória Flash para gravação ISP, 256 x 8 de memória RAM interna e 32 linhas de entrada e saída. [ATMEL, 2008]

A Figura 1 representa em detalhe as portas em que ligações podem ser feitas com o microcontrolador At89S8253, cuja descrição de cada porta é feita na página 25.

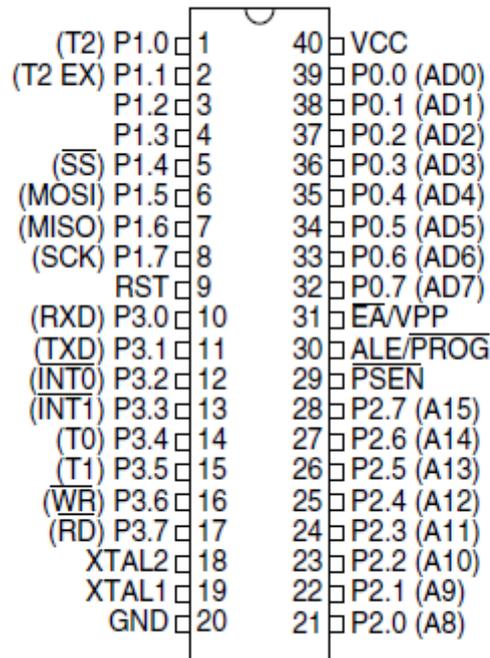


Figura 1 - Esquema do Microcontrolador AT89S8253

2.2. Células Solares

O princípio da utilização de células solares teve raízes na necessidade de geração de energia para estações de comunicação cuja localização era remota. Era uma solução razoável, pois eliminava a necessidade de construção de torres para transmissão de energia. No entanto sua eficiência era baixa, portanto seu uso muito restrito. [SILVA, PÉTRIS, PEREIRA, 2008]

Com a corrida espacial a busca por fontes de energia para satélites foi intensificada. Disso, resultou o desenvolvimento de novas e melhores células solares. Fato que até hoje a fonte de energia mais utilizada no espaço é a solar, através de células fotovoltaicas. [WITTRY, 2005]

O efeito base de Células fotovoltaicas, ou células solares, relatado em 1839 por Edmond Becquerel, diz que a energia é gerada graças à diferença de potencial entre dois pontos de um material semicondutor com capacidade de absorver energia. [INOVAÇÃO, 2008]

Os principais tipos de célula solar são: células de silício cristalino e células de arsênio de gálio. Retirando a diferença do material utilizado, podem-se notar também diferenças no processo de fabricação de cada tipo de célula.

As células de arsênio de gálio são feitas, de apenas um modo, em discos únicos, polidos, dependentes de grande estrutura e alto nível de fabricação. Já as células de silício podem ser fabricadas de dois modos: mono e policristalinas. A fabricação de células monocristalinas depende também de um nível de construção muito elevado, de modo similar as células de arsênio. [INOVAÇÃO, 2008]

O método de produção de células de silício policristalino, como as células mostradas na Figura 2, é mais fácil, no entanto ainda requer material especializado, mas não de tão alto nível quanto das outras, além de não precisar de um ambiente tão controlado. Por isso seu custo é mais baixo. Havia o problema de eficiência, pois as células policristalinas não eram tão eficientes quanto às outras, porém com o desenvolvimento da tecnologia de fabricação, a eficiência foi elevada, de modo que as células policristalinas têm eficiência menor, mas próxima, que células monocristalinas. [INOVAÇÃO, 2008]

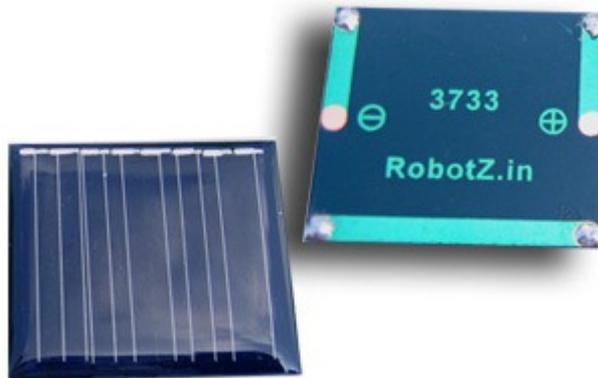


Figura 2 - Células solares policristalinas de uso comercial

2.3. Luz e o conceito de Dia e Noite para um microcontrolador

A luz é a única faixa do espectro eletromagnético que o olho humano pode enxergar. A luz compreende os comprimentos de onda desde 380 nm a 780 nm, e

por isso, para nós tem capacidade de iluminar e colorir. A cor é dada pelos diferentes comprimentos de onda que a luz comporta. [SUNROM, 2008]

Assim como o olho humano, um sensor de luz capta certa faixa desse espectro visível e o transforma em informação. Essa informação é demonstrada através de tensão de saída, mudança de resistência, mudança de cor, forma, ou qualquer outro método que foi construído. O sensor utilizado neste projeto é a célula solar, que varia a tensão de acordo com a luminosidade.

No entanto, o microcontrolador precisa de uma interface para identificar do que se trata essa informação fornecida pelo sensor. Através de um controlador, ou uma porta A/D (Analógico – Digital) é possível interpretar a saída da célula solar. A partir disso basta que o programador defina qual estado do sensor será o estado modelo de noite para que o microcontrolador tenha referência.

2.3.1. ADC 0804

O ADC 0804 é um componente que tem por função converter dados analógicos em dados digitais de 8 bits paralelos. Cada conversão, da entrada analógica, demora 100 microsegundos. [NATIONAL, 1994]

A conversão A/D é feita através do método de aproximações sucessivas, o que leva a precisão de 1 bit (um LSB – Least Significant Bit), onde o último bit apenas é duvidoso quanto a sua exatidão. [NATIONAL, 1994]

Na figura 3 encontra-se a representação das portas do conversor A/D.

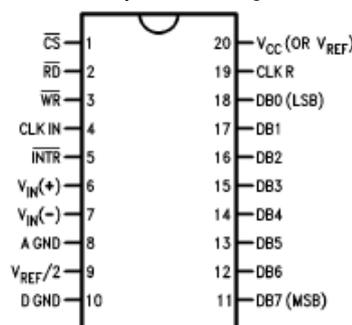


Figura 3 - Esquema do Conversor A/D ADC0804

A aquisição dos dados pode ser feita em [NATIONAL, 1994]:

- Free running mode – Modo livre: O conversor é colocado em modo livre, fazendo a conversão ao fim de cada ciclo. O programador tem disponível uma janela de tempo em que os dados estão disponíveis nas oito portas de saída para fazer a comparação. Após o fim de uma nova conversão estes dados são atualizados.
- Modo simples – O conversor irá fazer a conversão da entrada analógica apenas quando solicitado. O sistema deverá esperar que os dados estejam disponíveis para poder coletá-los. Caso não seja dado nenhum comando para uma nova conversão, os dados permanecerão inalterados na saída do conversor enquanto este permanecer ligado.

2.4. PWM (Pulse Width Modulation)

PWM, ou, do inglês, *Pulse Width Modulation* (modulação por largura de pulso), é um método de controle de sinal baseado em tempo. Tem como princípio básico tratar o sinal através de pulsos, ligando e desligando o componente controlado de acordo com a necessidade. [GHIRARDELLO]

Divide-se o pulso em dois ciclos, sendo um deles ativo, enquanto o outro inativo. De acordo com a proporção entre os dois estados do pulso, tem-se transferência de maior, ou menor, quantidade de energia.[GHIRARDELLO]

Ao contrário de energia transferida constantemente, com PWM pode-se alterar certos parâmetros de um sistema. Em LEDs, por exemplo, pode-se aplicar, por períodos curtos, mais tensão que o nominal, de modo a elevar seu brilho. Isto se dá, em PWM, aumentando a quantidade ativa em um ciclo.

A figura 4 abaixo representa três tipos de ciclos PWM. Neles, a parte alta representa a energia transferida, propriamente dita, enquanto a parte baixa demonstra o tempo que o componente permaneceu desligado.

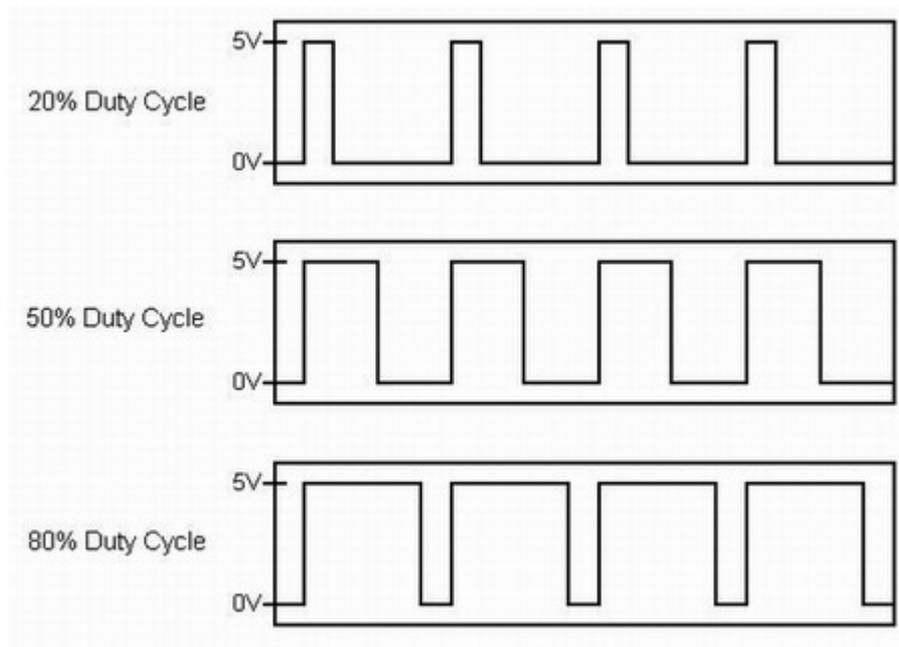


Figura 4 - Representação de pulsos PWM

Cada aplicação tem um limite, caso o ciclo inativo do sinal seja muito longo, o pulso PWM pode ser afetado. Diferentemente de chaves liga/desliga, PWM é algo transparente. A idéia base é que os ciclos desligados não sejam notados pela aplicação. Em motores, que esses não trepidem, em luzes, que estas não pisquem, entre outras aplicações. [GHIRARDELLO]

2.5. Chipset Hitachi HD 44780

É um chipset baseado em matriz de pontos. Pode mostrar caracteres alfa numéricos, além de símbolos e caracteres Kana japoneses, exibindo 240 caracteres diferentes. [HITACHI, 1998]

Suas especificações básicas são [HITACHI, 1998]:

- Suporte a operação com baixa energia – 2,7 a 5,5v
- Forma de controle do cristal líquido A (controle de uma linha)

- Clock interno de 2MHz
- Suporte a quatro ou oito bits
- Memória RAM com capacidade para 80 caracteres
- Reset automático inicializa o controlador após ligamento

2.5.1. Módulo LCD JHD 162A

O display LCD JHD 162A é baseado no chipset acima. Suas especificações diferem apenas na entrada da alimentação, que não suporta 2,7v, mas no mínimo 4,5v para correta alimentação de seu driver. [GHIRARDELLO]

Tem ainda a função de backlight, onde é possível ligar um LED, previamente instalado na fábrica, para iluminar a leitura.

Sua escrita comporta 2 linhas de 16 caracteres, com suporte total ao chipset Hitachi, como demonstrado pela figura 5. [GHIRARDELLO]



Figura 5 - Display LCD JHD 162A

3. DESENVOLVIMENTO DO PROJETO

Neste capítulo é descrita toda a montagem do protótipo, bem como sua programação. Os esquemas de ligação são devidamente demonstrados, assim como todos os cálculos necessários para adequação dos componentes ao projeto.

Há também a demonstração do software implementado no protótipo, o qual permite a correta demonstração de funcionalidade.

3.1. Preparação do hardware

O protótipo é composto de quatro partes: fonte, placa de controle, placa de PWM e iluminação. Destas, a fonte é dividida em duas, a placa de controle é peça única e a iluminação depende da placa PWM, mas estão em módulos separados.

A fonte do sistema tem de ser ligada em 220v, pois é a tensão de uso na cidade de formulação deste projeto. No entanto, com as devidas alterações, pode funcionar também em 110v. Nesta placa constam duas saídas, uma 12v, para alimentação dos LEDs, e uma 5v, para alimentação do restante do sistema.

O controle do protótipo é feito todo pelo microcontrolador, mas em sua placa ainda constam o conversor analógico/digital, a saída para o controle PWM, o display LCD e a conexão para o sensor de luz.

Já na placa PWM constam um microcontrolador, um transistor IRFZ48N e a saída direta para os LEDs. Na placa de iluminação ficam montados os LEDs, efetivamente responsáveis pela iluminação dos ambientes.

3.1.1. Fontes de 12 e 5v

Uma fonte personalizada é necessária ao protótipo, pois as utilizações de energia não são padronizadas. Enquanto todo o sistema trabalha em 5v, os LEDs, responsáveis pela iluminação trabalham em 12v.

Para tal adequação foram preparadas duas fontes distintas em uma mesma placa. As duas são baseadas no mesmo projeto, com as devidas alterações. A primeira fonte utiliza um regulador de tensão LM7812, fornecendo 11v na saída, enquanto a segunda trabalha com o regulador LM7805, fornecendo 5v. Ambas utilizam o mesmo esquema de ligação. Nota-se que o fornecimento correto seria de 12v, porém para tal é necessário utilizar um trafo que entregue mais de 12v na saída.

Os componentes de cada fonte, exceto pelo regulador são iguais, permitindo ainda novas adequações. O esquema da fonte é dado na figura 6:

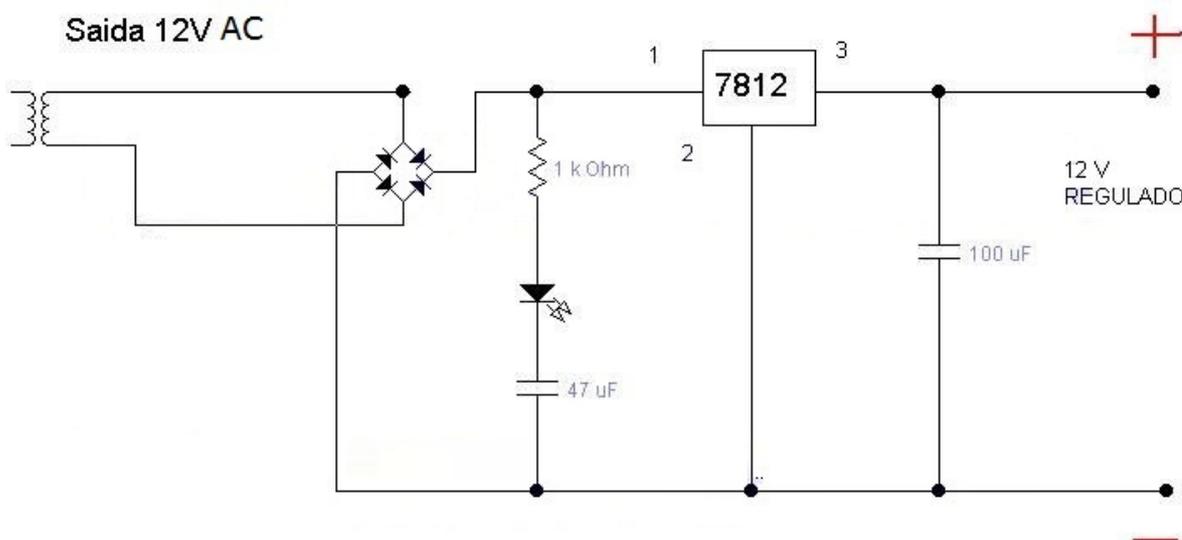


Figura 6 - Esquema de ligação da fonte 12v

No esquema da fonte há um transformador de corrente alternada que reduz 220v para 12v. Após uma ponte de retificação a saída é entregue ao regulador de tensão, que fornece um pólo positivo e um negativo na tensão especificada.

O uso de dissipadores de calor, no regulador de tensão de 12v, é necessário para garantir funcionamento prolongado da fonte.

3.1.2. Controle do protótipo

O controle do protótipo é dado por completo pelo microcontrolador. Para tal ele tem alimentação, proveniente da fonte, além dos outros periféricos. Eles são: um conversor analógico/digital ADC0804, um display LCD baseado no chipset Hitachi HD44780.

Todas as portas do microcontrolador foram utilizadas, sendo:

- A porta P0 é a entrada dos dados provenientes do ADC0804
- A porta P1 é utilizada como controle, sendo P1.0 e P1.1 ligadas aos botões;
- P1.2, P1.3 sendo ligadas aos pinos EN e RS do microcontrolador
- P1.4 a P1.7 ligados aos pinos CS, RD, WR e INT do ADC0804, para controle da conversão analógico-digital
- A porta P2, por completa, é utilizada para definição da saída para o chip responsável pelo PWM
- A porta P3 toda foi usada para transmissão de dados para o LCD, sendo DB0 relacionada à porta P3.0

No segundo microcontrolador são usadas as portas:

- P1 para recebimento de ordem do microcontrolador #01
- P2.2 para ativação do transistor, chaveando o PWM

3.1.2.1. ADC0804

O conversor ADC0804 necessita de quatro portas de controles ligadas diretamente ao microcontrolador. Esse controle, poderia ser descartado, colocando o chip em *free-running mode*, no entanto esta idéia foi descartada, pois o CI poderia perder sincronia com o microcontrolador, alterando seus dados antes que houvesse tempo de ser lido por completo. Desse modo a conversão da informação proveniente da célula solar é feita bastando ao microcontrolador dar a ordem. Para efetivação deste modo de funcionamento os pinos */CS*, */RD*, */WR* e */INT* foram ligados nas portas P1.5, P1.4, P1.7 E P1.6, respectivamente

Este CI não requer clock externo, porém ainda precisa de regulação para o mesmo. Essa regulação se da por conta de uma resistência de 10K ohms e um capacitor de 150pF, conforme o datasheet.

Há ainda o requerimento de tensões de referência. A primeira é o próprio Vcc, que é, neste caso também o Vref, 5v. A segunda é Vref/2, que foi conseguido através de um divisor de tensão com duas resistências de mesmo valor, no caso 100 ohms cada.

A figura 7 descreve as ligações feitas para o ADC0804:

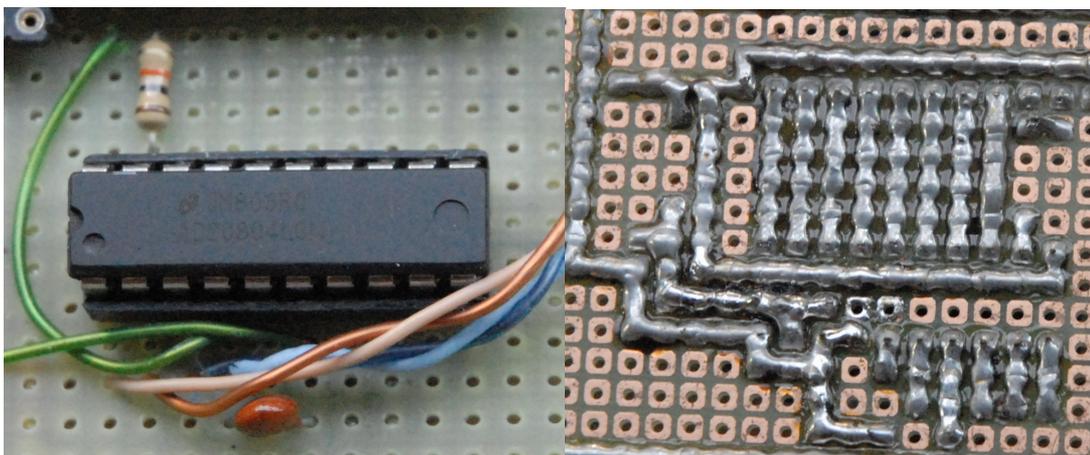


Figura 7 - Detalhe das ligações do ADC0804

3.1.2.2. Display de LCD

Há necessidade de informar ao usuário em que estado a iluminação esta no momento. Para tal é necessário um display de LCD. O escolhido é capaz de informar 16 caracteres por linha, em duas linhas, simultaneamente.

A ligação deste é feita de maneira bastante direta, sendo seus 8 bits de dados ligados diretamente a porta P3 do microcontrolador. Os seus outros 2 bits de controles estão também ligados as portas P1.2 e P1.3, sendo respectivamente os pinos 4 e 6 do display LCD.

O quinto pino, referente ao WR do LCD foi aterrado, pois é utilizada uma rotina de espera para que cada comando seja interpretado pelo LCD, não sendo necessário assim verificar o estado deste pino, ou a indicação de ocupado, na DB7 do módulo.

O contraste da tela é feito com um potenciômetro horizontal de 20K ohms. Essa resistência de ajuste de contraste foi escolhida por dificultar a alteração indevida e permitir novos ajustes caso algo mude na fonte.

A figura 8 mostra que praticamente toda a ligação do LCD foi feita por solda, para evitar mau contato. Mostra as ligações feitas por trás da placa de circuito perfurado e um *jumper* de fio, entre o pino de escrita e o pino terra do LCD.

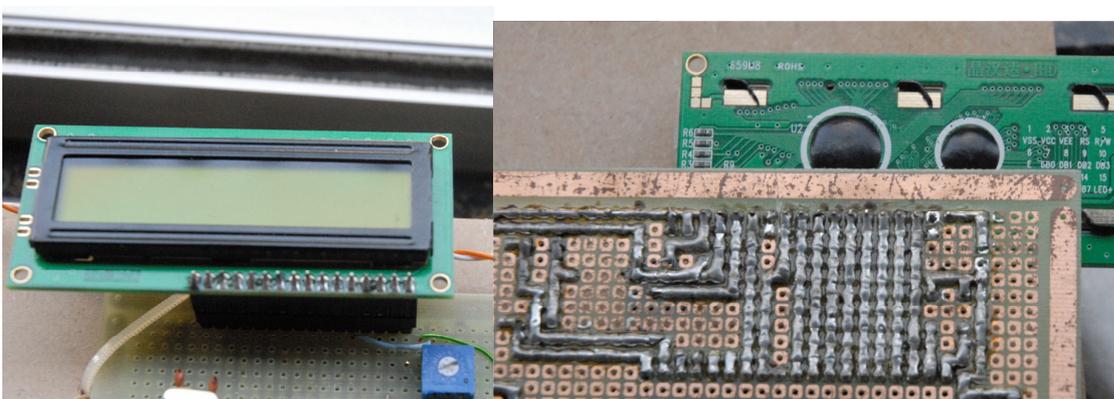


Figura 8 - Detalhe das ligações do LCD

3.1.2.3. Botões *Push-Down*

Os botões do tipo *push down* estão disponíveis para que o usuário possa entrar com sua seleção de luminosidade.

O esquema de ligação utiliza a facilidade de condução de corrente como base. A tensão de 5v vinda da fonte é entregue sempre ao microcontrolador, através de uma resistência de 10K ohms. Caso o botão seja pressionado a ligação é efetuada diretamente com o pino terra do sistema, portanto não chegando ao microcontrolador. É esta variação que é verificada pela porta quando um botão é pressionado.

O esquema desta ligação é visto na figura 9:

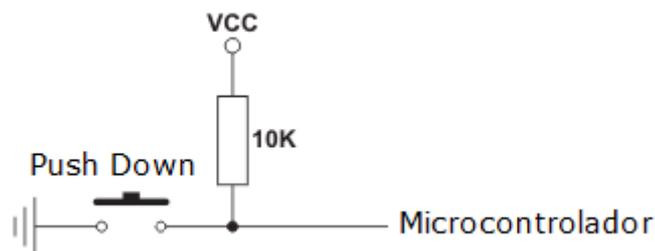


Figura 9 - Esquema de ligação de um botão

3.1.3. PWM

O microcontrolador não tem força suficiente para alimentar tantos LEDs quanto necessários a este projeto, portanto é necessária a ação de fonte externa de energia. Para fazer esta ligação foi escolhida a utilização de PWM, chaveado diretamente da fonte.

Foi feita uma placa com um microcontrolador Atmel AT89S8253, um transistor IRFZ48N e entrada para a fonte 12v. Este transistor, ligado a porta P2.2 do referido microcontrolador libera energia aos LEDs apenas quando acionado.

O funcionamento da chave é feita através de timers no código do programa. Desta maneira, não é obrigatoriamente necessário que o uC tenha geradores PWM incorporados.

A ligação desta placa é feita como demonstrada na figura 10:

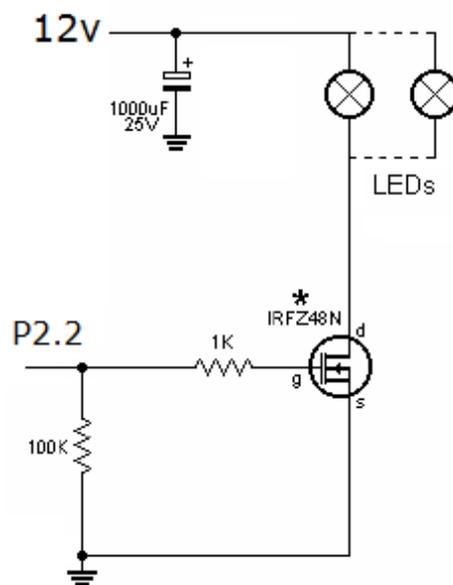


Figura 10 - Circuito de chave com transistor

Na figura 10 nota-se que a entrada de *gate* do transistor foi ligada diretamente no microcontrolador, para que possa ser acionada quando necessário. Assim, a fonte de alimentação que é fornecida aos LEDs é externa.

Na figura 11 encontra-se a ligação acima descrita devidamente implementada. Os conectores usados foram do tipo borne KRE, para facilitar a expansão e/ou manutenção das placas a esta conectada. A alimentação vem através do conector azul centralizado sozinho em uma das laterais da placa. Ali se encontra o pólo positivo e negativo dos LEDs.

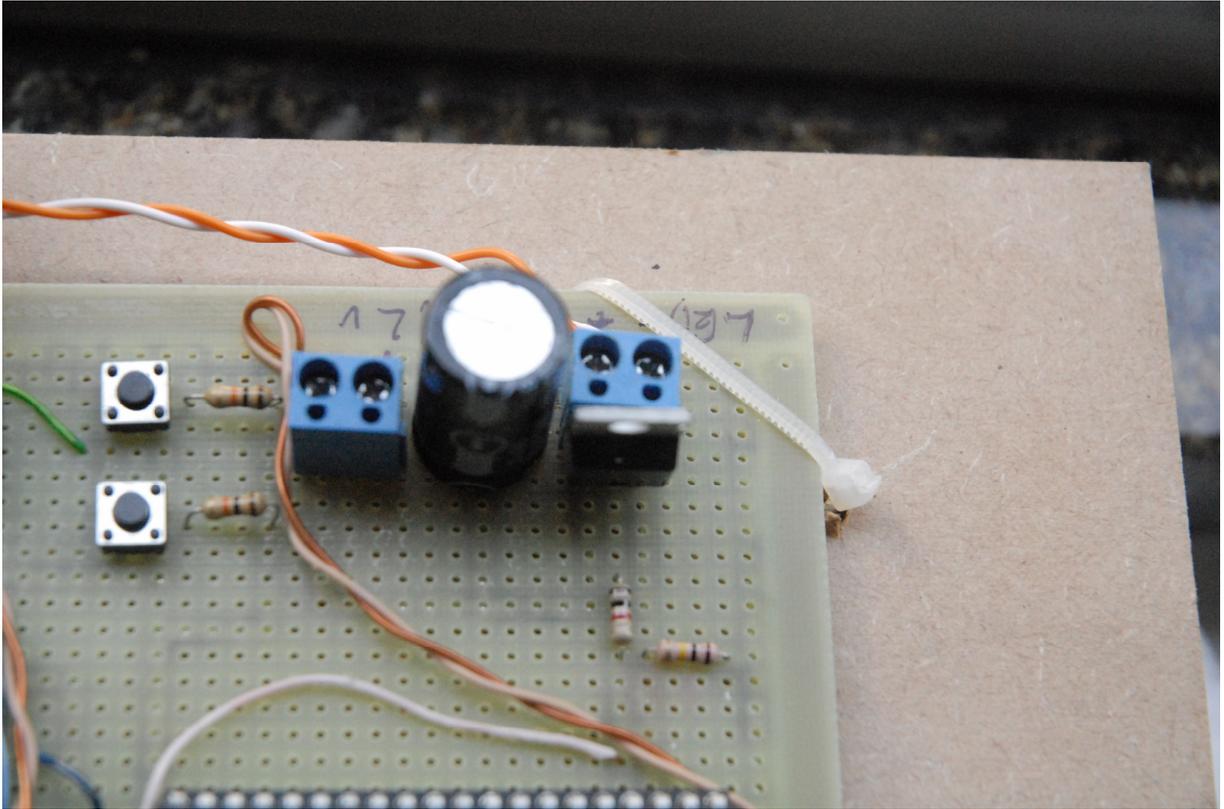


Figura 11 - Placa PWM

Há também o uso de um dissipador de calor no transistor, pois a quantidade de LEDs é limitada apenas pela capacidade da fonte, portanto dependendo da quantidade de LEDs utilizada existe a possibilidade de este componente esquentar bastante.

3.1.4. Placas de iluminação com LEDs

As três placas de LEDs são responsáveis por toda a iluminação artificial do projeto. Os LEDs foram arrumados em grupos de quatro, em série, permitindo grande controle sobre a quantidade de luz incidente sobre o ambiente, bastando apenas adicionar novos, em paralelo.

A figura 12 corresponde a uma das placas de LED utilizadas durante os testes do projeto:

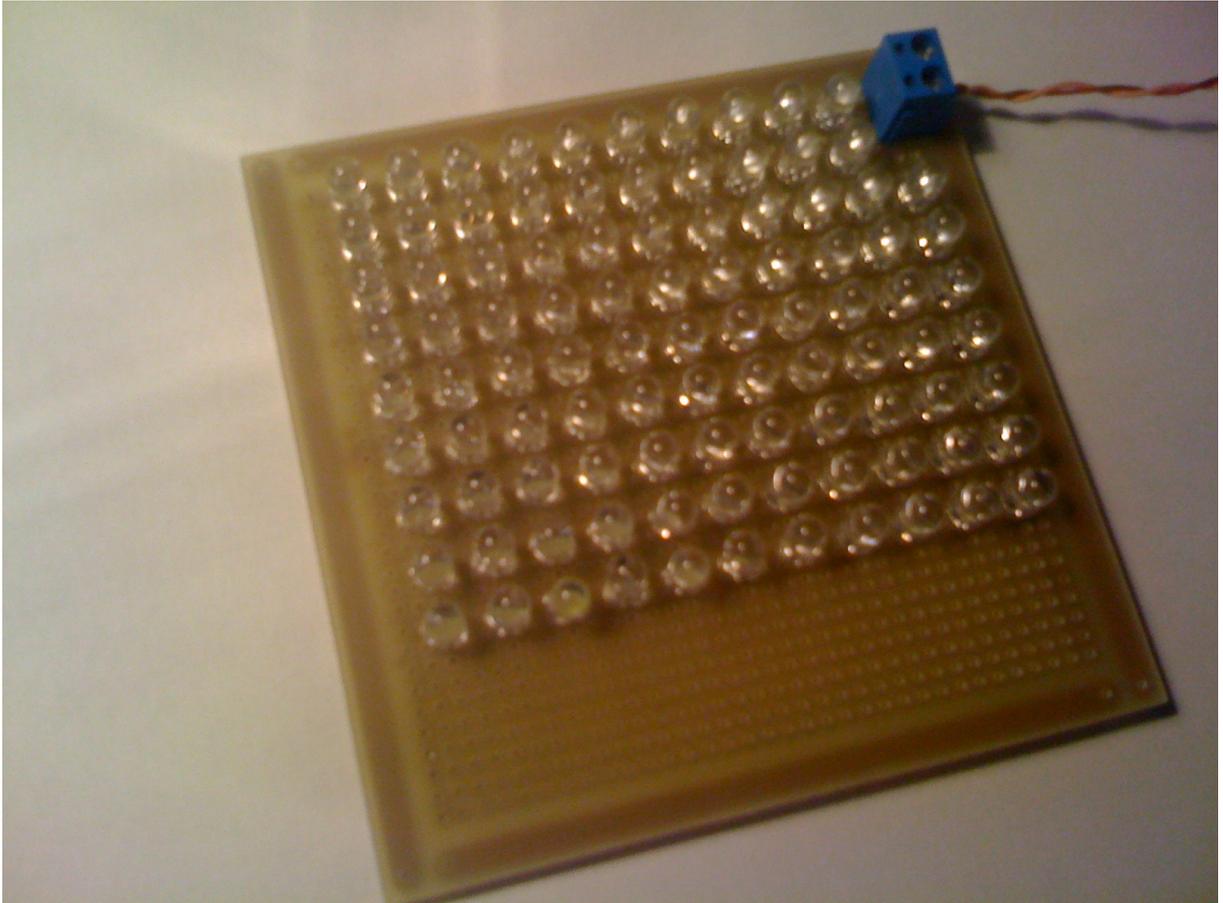


Figura 12 - Placa de LEDs

Os LEDs foram dispostos em séries de três, sem resistores, tendo para cada série um pólo positivo e terra comuns, sendo entregues diretamente ao IRFZ48N.

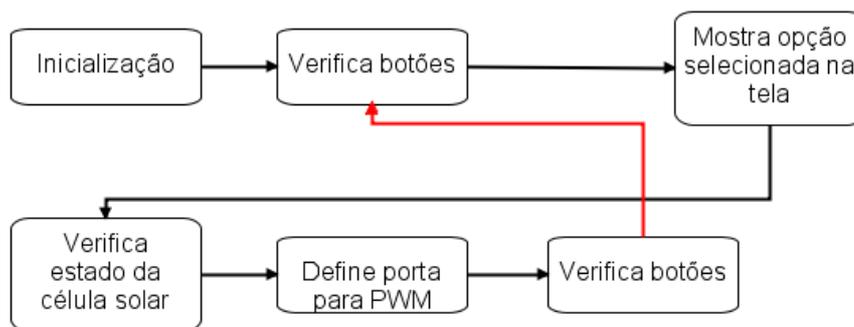
Cada LED escolhido tem capacidade de emitir 18.000 mcds (milicandela) quando utilizado com uma fonte de 5v e resistores de 100 ohms (1/4w). No entanto, com a utilização de PWM esse valor varia, de acordo com a necessidade, pois não há limitação de corrente com resistores.

3.2. Rotinas do software

3.2.1. Principais rotinas do microcontrolador

A utilização do software é baseada em um ciclo infinito de verificações. Estas são feitas a partir de séries de chamadas de funções que retornam ao mesmo ponto, garantindo continuidade ao software.

O software do microcontrolador principal (microcontrolador #01), responsável pela conversão analógico-digital, segue a lógica constante na figura 13.



Autodesk // LABS_

Figura 13 - Esquema base de funcionamento do protótipo

Durante a inicialização encontra-se a rotina de início do LCD, limpeza dos registradores, a mensagem de texto que identifica o responsável pela criação do software e, em seguida, apresenta acesso à primeira função do software, responsável por definir qual a iluminação deve ser utilizada.

O trecho de software a seguir exemplifica essa parte, onde estão as rotinas de mensagens iniciais.

```
org 0000H
mov Rmsg,#00H
mov Rpwm,#01H
mov Radc,#00H
mov adc_val,#00H
mov TMOD,#11H
```

```

        mov    P0,#11111111b
        mov    P1,#11111111b
        mov    P2,#11111111b
        mov    P3,#11111111b

MSG_INICIAL:    lcall   CONF_LCD
                mov     A,#80H
                lcall   SEND_INST
                mov     DPTR,#MSG1
                mov     SIZE,#0EH
                lcall   WR_LCD
                lcall   DELAY2

MSG_INICIAL2:   lcall   CONF_LCD
                mov     A,#80H
                lcall   SEND_INST
                mov     DPTR,#MSG2
                mov     SIZE,#0EH
                lcall   WR_LCD
                mov     A,#0C0H
                lcall   SEND_INST
                mov     DPTR,#MSG3
                mov     SIZE,#0FH
                lcall   WR_LCD

```

Seguindo o programa, são chamadas as funções responsáveis por verificar a pressão dos botões, escrever na tela a mensagem referente à opção selecionada, verificar a tensão da célula solar, calcular a saída para o PWM e um retorno ao início das verificações, respectivamente.

Com o código a seguir ficam resumidas todas as funções do software utilizado pelo microcontrolador.

```

Ver_botao:     ljmp   botao_up
Ver_MSG:       lcall  display_msg
                lcall  start_conv
                lcall  calculo_saida
                lcall  seta_saida
                sjmp  Ver_botao

```

O salto “sjmp Ver_botao” é responsável por fechar o loop, fazendo com que a verificação seja sempre efetuada da mesma maneira.

3.2.2. Escrita no LCD

Para que algo seja escrito no LCD é preciso obedecer a uma rotina, esta é definida durante a inicialização do programa. Após esta ter sido feita, basta chamar as funções de escrita, as quais têm autonomia para apresentar mensagens pré-determinadas.

Dos comandos necessários para escrever no LCD encontramos as funções CONF_LCD, SEND_INST, WR_LCD e DATA_IN. Essas funções são respectivamente responsáveis por inicializar o display, enviar instruções únicas ao chipset do LCD, escrever diretamente na tela e liberar para uso o LCD.

A principal função se refere a inicialização do display, de seu correto funcionamento todas as outras funções, do LCD, são dependentes:

```
CONF_LCD:    mov    A,#38H
              call   SEND_INST
              mov    A,#06H
              call   SEND_INST
              mov    A,#0EH
              call   SEND_INST
              mov    A,#01H
              call   SEND_INST
              mov    A,#0CH
              call   SEND_INST
              ret
```

Nessa função, é descrita a inicialização básica do chipset Hitachi HD44780, desde a seleção do tipo de display até a limpeza de memória, para início de inserção de dados. Os comandos utilizados têm as seguintes referências:

- 38H: Seleção de tipo de display: 2 linhas, 5x7 pontos em cada caractere
- 06H: Modo de entrada de dados
- 0EH: Liga o cursor

- 01H: Limpa a tela e a memória
- 0CH: desliga o cursor

As funções SEND_INST e DATA_IN são responsáveis por preparar e enviar ao LCD o conteúdo de cada dado ou instrução. A função SEND_INST tem capacidade de definir onde o cursor será colocado para início da escrita, assim como limpar a memória e inverter a entrada de dados. Já a função DATA_IN serve para enviar dados a serem escritos no display de LCD. As duas instruções neste código são similares, sendo suas diferenças um bit que é contrário em cada instrução.

```

ENV_INST:      mov  DATA,A
                clr  RS
                setb EN
                call DELAY_01
                clr  EN
                ret

DATA_IN:       mov  DATA,A
                setb RS
                setb EN
                call DELAY_01
                clr  EN
                ret

```

As funções acima se diferenciam por conta da segunda linha de cada função. Nela fica identificado que em uma (bit em zero) o comando enviado é uma instrução a ser executada e na outra (bit em um) o comando enviado é um dado a ser escrito na tela.

No código acima formulado não há verificação da sinalização de estado do LCD (*Busy Flag*). Devido a isso, após cada envio de instrução, é necessária a utilização de uma rotina de espera, para garantir que o comando enviado tenha tempo de ser executado, antes que a memória seja sobrescrita.

A última função prioritária do código de uso do LCD se chama WR_LCD. Ela faz a contagem dos dados a serem enviados para o LCD e utiliza a função DATA_IN para saída de dados.

```
WR_LCD:      mov  A,#0
             movc A,@A+DPTR
             call DATA_IN
             inc  DPTR
             djnz SIZE,WR_LCD
             ret
```

Com o acumulador inicialmente com valor zero, o valor do conteúdo de cada mensagem é enviado caractere a caractere para o LCD. A variável SIZE é a referência sobre o tamanho da palavra a ser enviada. Quando esta é reduzida a zero, o retorno da rotina de envio é finalizado.

3.2.2.1. Mensagens do LCD

As mensagens são programadas no código do programa, não dependendo de nenhuma entrada de dados para serem formadas. São guardadas através da diretiva DB, que aloca espaço na memória do tamanho necessário a mensagem escrita.

As mensagens disponíveis para o LCD são:

```
MSG1:      DB "PAULO GOBBATO "
MSG2:      DB "ESCOLHA OPCA:"
MSG3:      DB "  500 - 600"
MSG4:      DB "500 - 600 - 700"
MSG5:      DB "600 - 700 - 800"
MSG6:      DB "700 - 800 - 900"
MSG7:      DB "800 - 900 - 1000"
MSG8:      DB "900 - 1000  "
```

Nota-se que nessas mensagens, cada espaço em branco conta como um caractere, portanto tem de ser contado como tal, para a variável SIZE.

Nas mensagens enviadas para o LCD a opção escolhida pelo usuário é sempre aquela cuja numeração está no centro da mensagem.

3.2.3. Conversão Analógica / Digital

Com a utilização do ADC0804 é preciso que o microcontrolador prepare a conversão, para que depois possa acessar a porta P0, em que estão ligados o resultado da conversão.

Com os oito bits resultantes da conversão é possível alcançar valores entre 0 e 255. Para iniciar a leitura do sinal resultante da conversão, foi preparada a seguinte função, baseada no comando:

```
start_conv:
            acall conv
            acall read
            ret
```

Este é o comando base da função de leitura do ADC0804, nele são chamadas as funções de início de conversão e de leitura.

A função “conv” é responsável por preparar os bits nas portas relativas ao ADC0804. Desta maneira, a conversão é iniciada, bastando ao microcontrolador aguardar que ela seja finalizada para pegar o resultado. Para tal, seguinte a função de conversão, há uma função de espera, onde a sinalização de fim do ADC é constantemente verificada.

Descrita abaixo, a função “conv”, onde é evidenciada a alteração de estado do chip, seguida pela liberação para a escrita e aguardo da sinalização de pronto vinda do conversor. A verificação do sinalizador é interminável, caso nunca seja ativa esta porta

```
conv:
```

```

        clr cs
        clr adc_wr
        nop
        setb adc_wr
        setb cs
wait:
        jb intr,wait
        ret

```

A função “read”, responsável pela leitura do resultado da conversão move o conteúdo da porta onde o ADC está ligado para um registrador, após o dado estar salvo desativa o ADC0804.

```

read:
        clr cs
        clr adc_rd
        mov a,adc_port
        mov adc_val,a
        setb adc_rd
        setb cs
        ret

```

Após o armazenamento do resultado da conversão, a função retorna ao ponto onde foi parada a “start_conv”, que por sua vez retorna ao topo do software, onde a próxima função será chamada.

3.2.4. Leitura dos botões

A entrada de dados utilizando botões usa entrada lógica inversa. Nesse caso, quando o bit referente a porta é colocado em um, ele salta para a próxima função. Caso o bit seja colocado em zero, quando o botão push down é apertado, é verificado que a porta foi alterada, portanto o registrador referente a mensagem é incrementado ou decrementado.

A função que os descreve utiliza o comando JB, esse comando salta para a função alvo apenas se o bit na porta referida estiver em um.

```

botao_up:      jb  UP,botao_down
               inc Rmsg
               ljmp Ver_MSG

botao_down:    jb  DOWN,botao_fix
               dec Rmsg
               ljmp Ver_MSG

botao_fix:     ljmp Ver_MSG

```

O registrador R1 foi definido Rmsg e é utilizado para contar qual a mensagem que deve ser escrita no LCD.

3.2.5. Cálculo da necessidade de iluminação

A iluminação adquirida pela célula solar tem de ser comparada com a iluminação desejada para que seja possível determinar qual estado de iluminação artificial é necessário para que o ambiente se mantenha iluminado corretamente.

Em todo momento há um valor selecionado para a iluminação desejada, portanto sempre há um alvo de iluminação. Após uma seqüência de testes este valor é verificado, e assim a comparação toma parte.

Dentro da seqüência de teste, o valor fornecido pelo ADC0804 é comparado com valores determinados, em uma tabela de comparação. Caso o valor fornecido seja maior, o bit de CARRY da comparação não é afetado, enquanto quando for menor, é levantado. Isso obrigatoriamente indica que a iluminação no momento é mais alta, ou menor, que a desejada. Esse bit só é levantado nesta situação, portanto caso a iluminação seja mais igual ao desejado, nenhuma ação é tomada.

```

calculo_600:   cjne  Rmsg,#02H,calculo_700
               clr   C
               cjne  Radc,#50H,ver_menor_600
               ljmp  igual

```

```

ver_menor_600:  JC    menor_que_600
maior_que_600:  mov   Rpwm,#02H
                ret
menor_que_600:  mov   Rpwm,#04H
                ret

```

No código acima está uma das verificações, a qual contempla todo o modelo de código. Quando a comparação é dada como maior, a iluminação é imediatamente reduzida em um nível. Quando é dada como menor, incrementada em, também, um nível.

Isso faz com que a iluminação esteja sempre sendo trocada, de acordo com que a iluminação natural seja alterada, mantendo ainda margem de aumento máximo e mínimo.

3.2.6. Tempos para PWM

O cristal de clock utilizado tem capacidade de oscilar em 12MHz, fazendo com que cada instrução de ciclo único gaste 1 μ s para ser efetivada. Desta maneira, para gerar os pulsos responsáveis por acender e apagar os LEDs, foram utilizados as funções de contadores de tempo simples.

De modo a tornar menos aparentes os momentos desligados dos LEDs, foi utilizado o período de 255 μ s, sendo mantido fixo neste valor, durante o estado alto. Nos períodos referentes ao ciclo baixo, onde os LEDs ficam desligados, foi definido outro contador, cujo decremento varia entre 230 μ s e 30 μ s.

```

parte_baixa:  mov   R2,#01H
              mov   Rcalc,#01H
              cjne  Rpwm,#01H,nivel_02
              cpl   PWMPIN
              mov   TL0,#0xB8
              mov   TH0,#0xFC
              reti

```

O cálculo do tempo para cada ciclo do microcontrolador é feito a partir do seguinte, onde o resultado é o tempo tomado por cada instrução simples:

Frequência do cristal / 12 = Frequência máxima

Período = 1 / Frequência Máxima

Frequência máxima = 12MHz /12

Frequência máxima = 1000000Hz

Período = 1/1000000

Período = 1 µs

3.2.7. Interrupção para PWM

Há necessidade de controlar o tempo que deve ser usado para o ciclo baixo do PWM. Com a utilização de uma interrupção de timer, mesmo sem sincronia ou ponto fixo no código, é possível manter o PWM sempre atualizado quanto a qual tempo deve ser utilizado.

ATRO, interrupção escolhida, utiliza o timer 0 e necessita de inicialização para que seja funcional. Esta se dá através do seguinte trecho de código, onde ET indica interrupção por timer, TR0 o timer propriamente dito e EA autorização de uso de interrupções, sendo EA localizada ao fim da parte de boas vindas do programa:

```
setb ET0      ;Libera a função de interrupção timer 0  
setb TR0
```

3.2.8. Gravação do software no microcontrolador

A gravação do software no microcontrolador foi feita utilizando o gravador da placa Kit 8051XN, da fabricante Microgenios. A gravação desta placa é feita utilizando a tecnologia ISP (In system Programming).

O ISP é uma tecnologia baseada em comunicação serial, a qual não requer que o microcontrolador seja retirado da placa. No caso deste projeto, devido à confecção manual das peças, o microcontrolador tem de ser colocado neste kit toda vez que precisa ser gravado.

O software utilizado é o ISP programmer, da própria fabricante, ATMEL, cuja interface de controle é paralela. Para completar a conexão um cabo com pinos especificados pelo fabricante é utilizado.

4. RESULTADOS OBTIDOS

O protótipo foi testado em uma sala, com entrada de iluminação natural, acompanhado de um medidor de luminosidade. Os testes foram efetuados em diversas horas de um dia, portanto com diversas variações de entrada de luz.

Em todas as medições, foi feita antes uma medição de qual a luminosidade no momento, para garantir qual a luminosidade inicial.

A tabela 1 mostra quais foram as luminosidades iniciais e quais as luminosidades alcançadas, com relação ao alvo selecionado:

Tabela 1 – Luminosidade do protótipo

**Luminosidade com protótipo
(em lux)**

Alvo	Inicial	Adquirida*
500	25	480 - 530
600	24	590 - 640
700	22	680 - 720
800	26	770 - 820
900	25	890 - 930
1000	25	1100

Evidenciado na tabela 1, ha variação de luminosidade entre cada uma das faixas predefinidas. Isto é devido às variações constantes de estado do sensor de luz.

4.1. Análise do uso e sensibilidade do sensor de luz

O sensor de luz, baseado em fotocélula policristalina não é completamente linear. Por sua fabricação ser artesanal, não há datasheet que diga qual deveria ser sua resposta, mas com o auxílio de um Luxímetro e um multímetro digital, puderam ser estipuladas as comparações utilizadas no software de controle.

Tabela 2 – Variação de tensão e valores respectivos ao microcontrolador

Decimal	Hexadecimal	Volts	Luminância (lux)
131	83	2,311757	500
137	89	2,417639	600
143	8F	2,523521	700
149	95	2,629403	800
155	9B	2,735285	900
160	A0	2,82352	1000

Escala: 0,017647059 (Para 0 = 0v e 255 = 4,5v)

Com a tabela 2 fica descrito o comportamento da célula perante as situações apresentadas no projeto. Qualquer alteração fora disto é tratada pelo software como os extremos, portanto tomando uma única ação para cada.

A célula solar em conjunto com ADC0804, no entanto não se mostrou simples de trabalhar. A célula é bastante instável, variando de tensão a cada mínima alteração, não permitindo que a iluminação fique constante. A variação é transferida para o ADC, de modo que a comparação efetuada no software é sempre alterada, mesmo que visivelmente não tenha havido diferença na luminosidade.

4.2. Análise do funcionamento do PWM

Com a utilização de *pulse width modulation* foi possível alterar a luminosidade produzida pelos LEDs, sem causar grande impacto ao ambiente, como o barulho de relês, ou LEDs acendendo e apagando bruscamente.

A alteração de brilho é, conforme testes durante pesquisa dos componentes a ser utilizados, o método mais confiável de alterar a iluminação dos ambientes. Com isso também, é possível reduzir a quantidade de componentes utilizados.

Com os tempos utilizados a iluminação ficou duas vezes mais forte entre o mínimo e máximo de luminosidade demandada pelo sistema.

Com relação aos gastos de energia, podemos considerar a tabela 3 que compara a utilização do sistema, baseado em *PWM*, dando seu máximo de iluminação, com uma lâmpada em seu uso nominal:

Tabela 3 – Comparação com uma lâmpada comum

	LED	Lâmpada *
Corrente (mA)	20	113,6
Tensão (V)	3	220
Potência (W)	0,06	24,992
Intensidade (Lux)	18**	220***

* Lâmpada Philips de Iluminação de interiores de 25w, modelo STD-220V25-N. [Philips, 2009]

** Valor aproximado, medido a 1 metro do LED de alto brilho, branco, utilizado neste projeto.

*** Não definida a distância de verificação, no catálogo do fabricante.

Nota-se na tabela acima que a iluminação baseada em LEDs é mais eficiente, promovendo uma melhor relação de gasto de energia por iluminação, sendo necessários 13 LEDs para alcançar a mesma intensidade luminosa.

Com relação ao chaveamento, o componente IRFZ48N é rápido o suficiente para efetuar as trocas de estado entre os tempos definidos pelo timer. No entanto, caso o timer demore demais, a impressão a quem estiver no ambiente é de que as luzes estão piscando. Isto é um problema de cálculo do tempo do PWM, pois este não pode demorar demais entre trocas de estado.

4.3. Informações do LCD e aquisição de entrada através de botões

Os botões têm função de receber entrada de dados desejada pelo responsável pelo sistema, esse desejo é demonstrado no display LCD, na segunda linha.

De acordo com que o contador, atrelado aos botões, é incrementado, a variação é informada na tela. Junto são informados também os dois próximos valores imediatos, ficando disponíveis as seguintes opções na tela:

- 500 – 600
- 500 – 600 - 700
- 600 – 700 – 800
- 700 – 800 – 900
- 800 – 900 – 1000
- 900 – 1000

No quadro acima estão descritas a segunda linha das opções de iluminação diretamente disponíveis. Sendo a primeira, à esquerda, a opção anterior, a central a opção selecionada atualmente, e a direita a próxima opção.

O usuário pode reduzir e aumentar esta necessidade de acordo com as 6 opções disponíveis. Essas opções poderiam ser mais bem definidas caso o conversor analógico-digital tivesse maior resolução, pois a variação verificada entre cada estado da iluminação, útil para o sistema, é muito pequena. Outra opção similar seria a adoção de uma célula solar com maior variação de tensão.

5. CONSIDERAÇÕES FINAIS

Neste capítulo é discutida a conclusão, as dificuldades encontradas durante o trabalho e idéias de continuação e melhorias do projeto.

5.1. Dificuldades encontradas

As dificuldades encontradas foram a aquisição de dados através do ADC0804, que teve seu método de trabalho alterado, o microcontrolador dedicado e a calibração dos tempos de PWM, além do envio de sinais entre microcontroladores de forma fora de padrão.

A interface do ADC0804 apresentou alto nível de complexidade, pois inicialmente foi trabalhado em modo livre, para liberar mais portas do microcontrolador para outras funções. No entanto, sem o devido controle da conversão, os dados muitas vezes eram lidos de forma incorreta, por estar fora de sincronia, forçando o microcontrolador a tomar atitudes incorretas. Este erro não se apresentou durante as simulações, pois o ambiente virtual é completamente controlado e não sofre variações imprevistas.

Utilizando uma fotocélula no sistema ficou claro que a resolução do ADC0804 é baixa para a tarefa a ser executada, pois a variação de tensão dentro a faixa de iluminação proposta é muito pequena. No entanto, o chip foi mantido por ser o único conversor A/D encontrado no mercado de Brasília.

Com relação ao PWM a grande dificuldade foi o microcontrolador escolhido. Como o gravador utilizado tem apenas capacidade de gravação deste tipo de chip o protótipo ficou restrito a apenas este microcontrolador. O AT89S8253 não tem gerador PWM interno, portanto sua resolução de tempo é baixa. Com a resolução de tempo fornecida no protótipo a luminosidade, quando gerada muito baixa, aparenta não ser constante.

A linguagem de programação *assembly* também se mostrou obstáculo, principalmente na comparação da leitura dos dados fornecidos pelo ADC. Toda a programação teve de ser estudada novamente, de modo a tornar possível a confecção do código.

Outra dificuldade, que restringiu o uso de microcontroladores para utilização neste protótipo foi a falta de acesso a variados tipos de gravadores, impossibilitando o uso de chips de arquiteturas diferentes, o que poderia ter provido o protótipo de alta capacidade PWM ou mesmo melhor conversão analógico-digital.

5.2. Conclusão

O objetivo proposto para este projeto, que consiste em um controle de iluminação que não dependa de entrada de dados por um usuário, foi alcançado, cumprindo assim a proposta apresentada.

O protótipo tem seu destaque na construção em peça única, permitindo assim que este possa ser aproveitado e adaptado para qualquer necessidade. No entanto é prejudicado devido às placas terem sido confeccionadas artesanalmente, não utilizando um modelo pronto, no qual bastaria imprimi-lo acima de uma placa de cobre.

Com os conhecimentos adquiridos através do curso de engenharia, foi possível projetar um sistema independente, que beneficia todos que freqüentam o ambiente, além de criar um exemplo de sistema de automação tão necessário para o controle de ambientes.

Com a elaboração deste projeto ficou clara a relação de um engenheiro com seu trabalho, onde não necessariamente ele precisa criar novas tecnologias, mas sim definir um método de uso das tecnologias disponíveis para a melhoria do nível de bem-estar de quem use sua ferramenta.

5.3. Propostas futuras

Em uma nova implementação deste protótipo, ficam como propostas de implementação os seguintes tópicos:

- Introdução de uma célula solar com maior variação de tensão relativa a luminosidade;
- Utilização de microcontroladores AVR ou PIC;
- Utilização de conversores A/D de maior resolução;
- Adição de motor de passo para controle de persianas.

Essas modificações serão extremamente importantes, pois garantirão melhor qualidade ao novo projeto, além de redução do espaço gasto na placa.

Os microcontroladores AVR, em sua maioria, têm integrados dois dispositivos primordiais para uma melhor produção deste projeto. Estes itens são: timer gerador PWM independente e conversor analógico-digital integrado. Alguns modelos têm ainda conversor digital-analógico, para geração, dentre outros, de som.

Um gerador de PWM independente tem maior precisão que o timer utilizado pelo 8051, podendo fornecer uma rede mais extensa de pulsos utilizados.

O conversor analógico digital também fornece uma nova opção de método de uso da célula solar. Quando de seu uso, os dados já estão disponíveis diretamente no microcontrolador, não causando assim dificuldades em adquirir os dados, ou erros de sincronia.

Uma célula solar de maior variação de tensão também é uma possível alteração interessante, pois as variações de tensão podem ser lidas com margens menores de perturbação.

REFERÊNCIAS BIBLIOGRÁFICAS

ABNT - **Norma NBR 5413/92: Norma para iluminância de interiores** – Rio de Janeiro, Brasil - Abril, 1992

ATMEL Corporation – **AT89S8253 Datasheet** – San Jose, Estados Unidos da América – 2008

[CADT] Cientistas Associados Desenvolvimento Tecnológico Ltda. - **Introdução à programação do microcontrolador MSP430** – São Carlos, SP

Comissão Interna de Prevenção de Acidentes – PRODEPA – **Iluminação no Ambiente de Trabalho**. Disponível em: <<http://cipa.prodepa.gov.br/?q=node/64>> Acesso em 12 Set. 2009.

GHIRARDELLO, Ariovaldo - **Apostila Sobre Modulação PWM** – Curso Técnico em Eletrônica – Colégio Politec – Disponível em: <http://www.eletronica.org/arq_apostilas/apostila_pwm.pdf> Acessado em: 30 de Set. 2009

HITACHI – **HD44780U Datasheet** – Japão – 1998

INOVAÇÃO Tecnológica, **Células Solares Policristalinas dão Salto de Eficiência** – Disponível em: <<http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=celulas-solares-policristalinas-dao-salto-de-eficiencia>> Acessado em: 12 Set. 2009

LOPES, Eduardo da Silva; ZANLORENZI, Erinton; COUTO, Luis Carlos; MINETTI, Luciano José. **Análise do Ambiente de Trabalho em indústrias de processamento de madeira na região Centro-Sul do Estado do Paraná**. Periódico Scientia Florestalis – Dezembro de 2004

MICROGENIOS. **Kits Didáticos**. Disponível em:

<<http://www.microgenius.com.br/shop/sessoes.asp?id=28>>. Acesso em: 20 Set. 2009

NATIONAL Semiconductor - **ADC 0804 Datasheet** – Arlington – Estados Unidos da América – Dezembro, 1994

NICOLOSI, Denys E. C. - **Microcontrolador 8051 Detalhado**. - 5 ed. - São Paulo - Érica Ltda, 2004

PHILIPS do Brasil, Catálogo de Produtos – **Lâmpadas Incandescentes** – Disponível em: <http://www.luz.philips.com.br/archives/Standard_out2009.pdf> Acessado em: 30 de Out. 2009

SICA, Carlos. **Sistemas Automáticos com Microcontroladores 8031/8051**. 1.ed. São Paulo - SP: Novatec, 2006

SILVA, Everton Rocha de Paula; PÉTRIS Gustavo Cunha; PEREIRA, Luis Fillipe Couto de Araujo . **Tutorial de Célula Solar**. Universidade Federal Fluminense – Niterói – RJ, Março de 2008

SOUZA, M. B.. **Potencialidade de Economia de Energia Elétrica Gasta em Iluminação através do Aproveitamento da Luz Natural com a Utilização de Sistemas Automáticos de Controle**. In: IX Encontro Nacional de Tecnologia do Ambiente Construído., 2002, Foz do Iguaçu. IX Encontro Nacional de Tecnologia do Ambiente Construído., 2002.

SUNROM Technologies – **LDR 3190 Datasheet** – Estados Unidos da América, Julho de 2008

WITTRY, Jan – **Harnessing the Sun: NASA Studies Advanced Solar Cells on Station** – Cleveland - OH – USA, Maio de 2005, Disponível em:

<http://www.nasa.gov/mission_pages/station/science/FTSCE_MiSSE_feature.html>

Acessado em 13 Set. 2009

APÊNDICES

APÊNDICE A - CÓDIGO DO MICROCONTROLADOR:

```
-----  
;Paulo Gobbato  
;Engenharia de Computação - UniCEUB  
;Projeto de Conclusão de Curso  
;Software do microcontrolador #1  
-----  
                UP            EQU    P1.0            ;Botões  
                DOWN        EQU    P1.1  
                adc_rd      EQU    P1.2            ;Controle do ADC0804  
                cs          EQU    P1.3  
                adc_wr      EQU    P1.4  
                intr        EQU    P1.5  
                RS          EQU    P1.6            ;Controle do LCD  
                EN          EQU    P1.7  
                LCD_DATA    EQU    P3             ;Porta utilizada para o LCD  
                Rlcd        EQU    R0             ;Variavel de tempo do timer 1 (lcd)  
                Rcalc       EQU    R1             ;libera ou não calculo da saída  
                Rint        EQU    R2             ;Libera a parte da interrupção a ser usada  
                SIZE        EQU    R3             ;Local da variavel de tamanho dos caracteres  
                Rmsg        EQU    R4             ;Registradores em uso  
                ;R5 -> Livre  
                Rpwm        EQU    R6  
                Radc        Equ    R7             ;Variavel do ADC  
                PWMPIN      EQU    P2.0  
  
                org         0000h  
                ljmp        start  
  
                org         0Bh  
                ljmp        TIMER_0_INTERRUPT  
  
start:          org         0100H  
Ver_bot:       call        startup  
Ver_MSG:       call        botao_up  
                lcall       start_conv  
                lcall       calculo_saida  
                sjmp        ver_bot  
  
startup:       mov         TMOD,#11H  
                mov         Rmsg,#01H  
                mov         Rpwm,#01H  
                mov         Radc,#00H  
                Mov         P2,#00000000b  
                mov         P0,#11111111b  
                clr         A  
                call        init_timer  
                call        boas_vindas  
                setb        EA  
  
-----  
;Verificação dos botões  
-----
```

```

botao_up:      jb  UP,botao_down
               inc Rmsg
               call display_msg
               ret

botao_down:    jb  DOWN,botao_fix
               dec Rmsg
               call display_msg
               ret

botao_fix:     ret
;-----
;Rotinas de mensagens
;-----
display_msg:   cjne Rmsg,#00H,display_500
               mov Rmsg,#01H

display_500:   cjne Rmsg,#01H,display_600
               lcall CONF_LCD
               mov A,#80H
               lcall SEND_INST
               mov DPTR,#MSG2
               mov SIZE,#0EH
               lcall WR_LCD
               mov A,#0C0H
               lcall SEND_INST
               mov DPTR,#MSG3
               mov SIZE,#0FH
               lcall WR_LCD
               ret

display_600:   cjne Rmsg,#02H,display_700
               lcall CONF_LCD
               mov A,#80H
               lcall SEND_INST
               mov DPTR,#MSG2
               mov SIZE,#0EH
               lcall WR_LCD
               mov A,#0C0H
               lcall SEND_INST
               mov DPTR,#MSG4
               mov SIZE,#0FH
               lcall WR_LCD
               ret

display_700:   cjne Rmsg,#03H,display_800
               lcall CONF_LCD
               mov A,#80H
               lcall SEND_INST
               mov DPTR,#MSG2
               mov SIZE,#0EH
               lcall WR_LCD
               mov A,#0C0H
               lcall SEND_INST
               mov DPTR,#MSG5
               mov SIZE,#0FH
               lcall WR_LCD
               ret

display_800:   cjne Rmsg,#04H,display_900
               lcall CONF_LCD
               mov A,#80H
               lcall SEND_INST
               mov DPTR,#MSG2
               mov SIZE,#0EH
               lcall WR_LCD
               mov A,#0C0H

```

```

    lcall SEND_INST
    mov DPTR,#MSG6
    mov SIZE,#0FH
    lcall WR_LCD
    ret

display_900:  cjne Rmsg,#05H,display_1000
    lcall CONF_LCD
    mov A,#80H
    lcall SEND_INST
    mov DPTR,#MSG2
    mov SIZE,#0EH
    lcall WR_LCD
    mov A,#0C0H
    lcall SEND_INST
    mov DPTR,#MSG7
    mov SIZE,#10H
    lcall WR_LCD
    ret

display_1000: cjne Rmsg,#06H,fix_Rmsg2
    lcall CONF_LCD
    mov A,#80H
    lcall SEND_INST
    mov DPTR,#MSG2
    mov SIZE,#0EH
    lcall WR_LCD
    mov A,#0C0H
    lcall SEND_INST
    mov DPTR,#MSG8
    mov SIZE,#10H
    lcall WR_LCD
    ret

fix_Rmsg2:   cjne Rmsg,#07H,fix_display
    mov Rmsg,#06H
    sjmp display_1000

fix_display:  ret
;-----
;Rotinas de leitura do ADC0804
;-----
start_conv:

begin_conv:  clr a
             call conv           ;Inicio da conversão
             call read          ;Le valor convertido
             ret

conv:        clr cs              ;Limpa cs e wr para preparar o inicio da conversão
             clr adc_wr
             nop
             setb adc_wr        ;Inicia a conversão levantando CS e WR
             setb cs

wait:        jb intr,wait
             ret

read:        clr cs
             clr adc_rd
             mov a,P0           ;le o valor convertido
             setb adc_rd
             setb cs
             mov Radc,a        ;guarda o valor em um registrador
             ret

```

```
;-----  
;Funções de calculo da saída  
;-----
```

```
calculo_saida:  cjne Rcalc,#01H,igual
```

```
                dec Rcalc
```

```
                clr A
```

```
                clr C
```

```
calculo_500:    cjne Rmsg,#01H,calculo_600
```

```
                clr C
```

```
                cjne Radc,#49H,ver_menor_500
```

```
                ljmp igual
```

```
ver_menor_500: JC menor_que_500
```

```
maior_que_500: mov Rpwm,#01H
```

```
                ret
```

```
menor_que_500: mov Rpwm,#03H
```

```
                ret
```

```
calculo_600:    cjne Rmsg,#02H,calculo_700
```

```
                clr C
```

```
                cjne Radc,#50H,ver_menor_600
```

```
                ljmp igual
```

```
ver_menor_600: JC menor_que_600
```

```
maior_que_600: mov Rpwm,#02H
```

```
                ret
```

```
menor_que_600: mov Rpwm,#04H
```

```
                ret
```

```
calculo_700:    cjne Rmsg,#03H,calculo_800
```

```
                clr C
```

```
                cjne Radc,#57H,ver_menor_700
```

```
                ljmp igual
```

```
ver_menor_700: JC menor_que_700
```

```
maior_que_700: mov Rpwm,#05H
```

```
                ret
```

```
menor_que_700: mov Rpwm,#07H
```

```
                ret
```

```
calculo_800:    cjne Rmsg,#04H,calculo_900
```

```
                clr C
```

```
                cjne Radc,#5CH,ver_menor_800
```

```
                ljmp igual
```

```
ver_menor_800: JC menor_que_800
```

```
maior_que_800: mov Rpwm,#06H
```

```
                ret
```

```
menor_que_800: mov Rpwm,#08H
```

```
                ret
```

```
calculo_900:    cjne Rmsg,#05H,calculo_1000
```

```
                clr C
```

```
                cjne Radc,#62H,ver_menor_900
```

```
                ljmp igual
```

```
ver_menor_900: JC menor_que_900
```

```
maior_que_900: mov Rpwm,#09H
```

```
                ret
```

```
menor_que_900: mov Rpwm,#0BH
```

```
                ret
```

```
;Se o valor calculado é igual a 1  
;salta para a função de retorno, sem  
;alterações  
;Rcalc = 0 -> não é efetuado novo calculo  
;até o fim deste ciclo PWM
```

```
;Se o valor desejado for igual a 1, faz a  
;verificação:  
;Limpa o carry bit  
;Caso o resultado da conversão A/D seja  
;igual ao descrito  
;salta para a função igual e mantém o ;  
valor do pwm.
```

```
;Caso seja maior, o carry bit não é  
;setado, reduzindo para o  
;nivel descrito o timer de PWM
```

```
;Caso seja menor, o carry bit é setado,  
;aumentando para o nivel  
;descrito o timer de PWM.
```

```

calculo_1000:  cjne Rmsg,#06H,igual
                clr C
                cjne Radc,#69H,ver_menor_1000
                ljmp igual
ver_menor_1000:JC menor_que_1000
maior_que_1000:mov Rpwm,#0AH
                ret
menor_que_1000:  mov Rpwm,#0CH
                ret

igual:         ret

```

```

;-----
;Rotinas do LCD
;-----

```

```

Boas_vindas:
MSG_INICIAL:  lcall CONF_LCD
                mov A,#80H
                lcall SEND_INST
                mov DPTR,#MSG1
                mov SIZE,#0EH
                lcall WR_LCD
                lcall DELAY_02

```

```

MSG_INICIAL2: lcall CONF_LCD
                mov A,#80H
                lcall SEND_INST
                mov DPTR,#MSG2
                mov SIZE,#0EH
                lcall WR_LCD
                mov A,#0C0H
                lcall SEND_INST
                mov DPTR,#MSG3
                mov SIZE,#0FH
                lcall WR_LCD
                ret

```

```

CONF_LCD:     mov A,#38H      ;seleciona o tipo de display utilizando (5x7 linhas por caractere), 16x2
                call SEND_INST
                mov A,#06H    ;Libera para entrada de dados
                call SEND_INST
                mov A,#0EH    ;Liga o cursor
                call SEND_INST
                mov A,#01H    ;Limpa a tela e a memória
                call SEND_INST
                mov A,#0CH    ;Desliga o cursor
                call SEND_INST
                ret

```

```

SEND_INST:   mov LCD_DATA,A  ;Coloca o conteúdo do acumulador na variavel dados
                clr RS        ;Marca como envio de instrução
                setb EN       ;Coloca em modo de escrita
                call DELAY_01 ;aguarda até a instrução terminar
                clr EN       ;Sai do modo de escrita
                ret

```

```

WR_LCD:      mov A,#0        ;Limpa o acumulador
                movc A,@A+DPTR
                call LCD_DATA_IN ;Chama a função de escrita de dados
                inc DPTR
                djnz SIZE,WR_LCD ;Enquanto a variável de tamanho não for igual a zero: continua
                ;escrevendo até terminar a frase (fim da frase: tamanho = 0)
                ret

```

```

LCD_DATA_IN: mov LCD_DATA,A
                setb RS
                setb EN

```

```

                call DELAY_01      ;espera até a instrução ser terminada
                clr EN
                ret
;-----
;ATRASOS DO LCD
;-----
DELAY_01:      mov TL1,#LOW(65535-24000)
                mov TH1,#HIGH(65535-24000)
                setb TR1
                jnb TF1,$
                clr TR1
                clr TF1
                ret

DELAY_02:      mov Rlcd,#50
TIME:          mov TH1,#HIGH(65535-45000)
                mov TL1,#LOW(65535-45000)
                clr TF1
                setb TR1
                jnb TF1,$
                clr TF1
                clr TR1
                djnz Rlcd,TIME
                ret
;-----
;Mensagens para o LCD
;-----
MSG1:          DB "PAULO GOBBATO " ;14 - 0EH
MSG2:          DB "ESCOLHA OPCAO:"
MSG3:          DB " 500 - 600" ;15 - 0FH
MSG4:          DB "500 - 600 - 700"
MSG5:          DB "600 - 700 - 800"
MSG6:          DB "700 - 800 - 900"
MSG7:          DB "800 - 900 - 1000" ;16 - 10H
MSG8:          DB "900 - 1000  "
;-----
;Inicialização do timer de intrrupções
;-----
Init_timer:    mov tl0,#00h          ;0 segundos
                mov th0,#00h
                setb ET0             ;Libera a função de interrupção timer 0
                setb TR0             ;Libera a correr o timer
                ret
;-----
;Funções para o PWM - Chamada exclusiva pela interrupção de tempo
;-----
TIMER_0_INTERRUPT:
parte_alta:    cjne R2,#01H,parte_baixa ;300us
                mov R2,#00H
                cpl PWMPIN
                mov TL0,#0xF0
                mov TH0,#0xF1
                reti

parte_baixa:   mov R2,#01H
                mov Rcalc,#01H

                cjne Rpwm,#01H,nivel_02 ;400us
                cpl PWMPIN
                mov TL0,#0xB8
                mov TH0,#0xFC
                reti

nivel_02:      cjne Rpwm,#02H,nivel_03 ;370

```

```

cpl PWMPIN
mov  TL0,#0x50;Lower Byte
mov  TH0,#0xFB      ;Upper Byte
reti

nivel_03:  cjne Rpwm,#03H,nivel_04          ;340
            cpl PWMPIN
            mov  TL0,#0xE8;Lower Byte
            mov  TH0,#0xF9      ;Upper Byte
            reti

nivel_04:  cjne Rpwm,#04H,nivel_05          ;310
            cpl PWMPIN
            mov  TL0,#0x80;Lower Byte
            mov  TH0,#0xF8      ;Upper Byte
            reti

nivel_05:  cjne Rpwm,#05H,nivel_06          ;280
            cpl PWMPIN
            mov  TL0,#0x18;Lower Byte
            mov  TH0,#0xF7      ;Upper Byte
            reti

nivel_06:  cjne Rpwm,#06H,nivel_07          ;250
            cpl PWMPIN
            mov  TL0,#0xB0;Lower Byte
            mov  TH0,#0xF5      ;Upper Byte
            reti

nivel_07:  cjne Rpwm,#07H,nivel_08          ;220
            cpl PWMPIN
            mov  TL0,#0x48;Lower Byte
            mov  TH0,#0xF4      ;Upper Byte
            reti

nivel_08:  cjne Rpwm,#08H,nivel_09          ;190
            cpl PWMPIN
            mov  TL0,#0xE0;Lower Byte
            mov  TH0,#0xF2      ;Upper Byte
            reti

nivel_09:  cjne Rpwm,#09H,nivel_10         ;160
            cpl PWMPIN
            mov  TL0,#0x78;Lower Byte
            mov  TH0,#0xF1      ;Upper Byte
            reti

nivel_10:  cjne Rpwm,#0AH,nivel_11         ;130
            cpl PWMPIN
            mov  TL0,#0x10;Lower Byte
            mov  TH0,#0xF0      ;Upper Byte
            reti

nivel_11:  cjne Rpwm,#0BH,nivel_12         ;100
            cpl PWMPIN
            mov  TL0,#0xA8;Lower Byte
            mov  TH0,#0xEE      ;Upper Byte
            reti

nivel_12:  cpl PWMPIN                      ;70
            mov  TL0,#0x40;Lower Byte
            mov  TH0,#0xED      ;Upper Byte
            reti

```

END

APÊNDICE B - LIGAÇÃO DA PLACA DE CONTROLE:

O design simplificado da placa de controle foi feito no software Proteus 7, versão de demonstração. Nele foi gerada a imagem abaixo, que descreve as ligações, porta a porta utilizadas por cada componente lógico da placa principal.

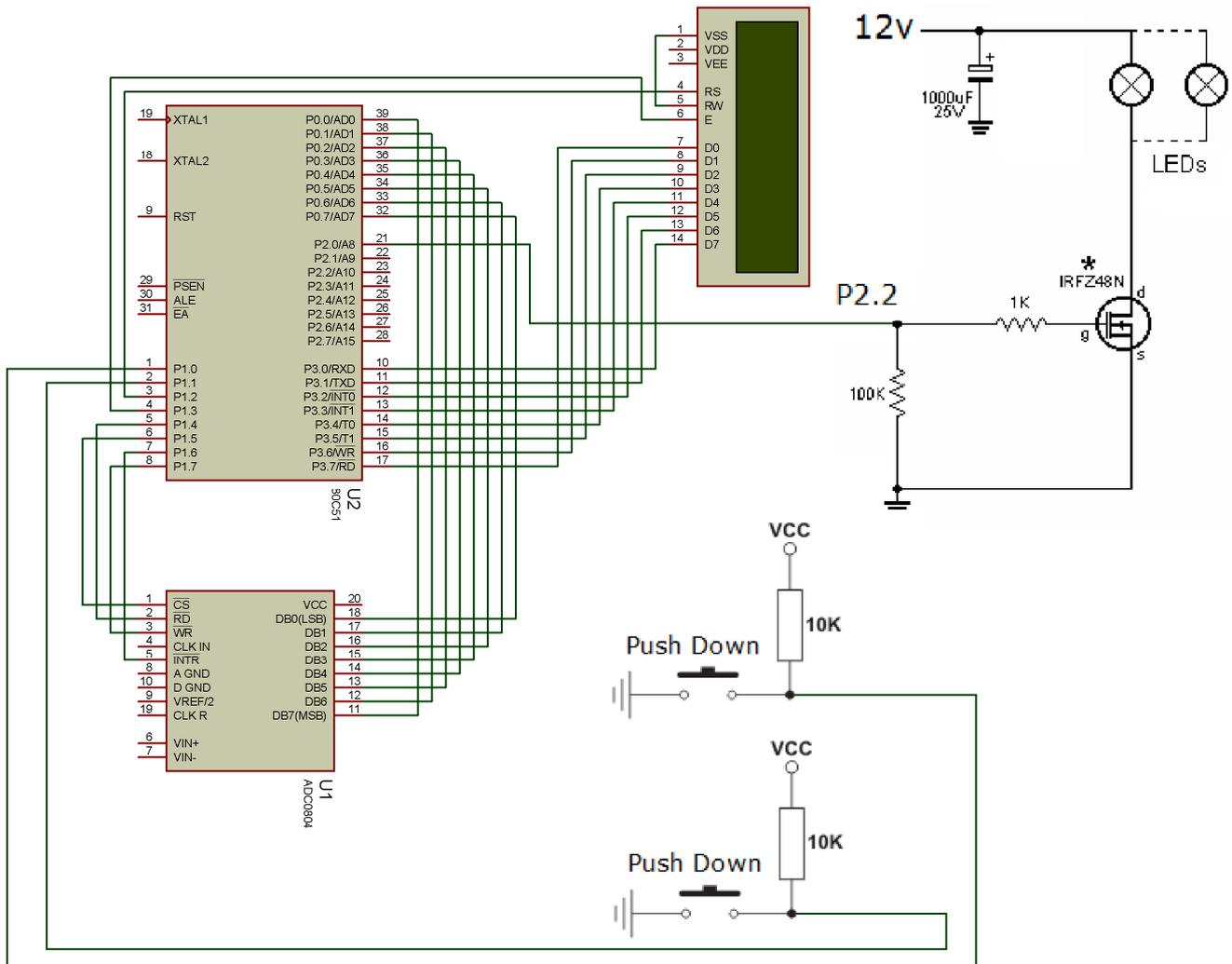


Figura 14 - Esquema de ligação da placa de controle

ANEXOS

ANEXO A – CONFIGURAÇÃO DOS PINOS DO AT89S8253

Retirado diretamente da datasheet

2. Pin Configurations

2.1 40P6 – 40-lead PDIP

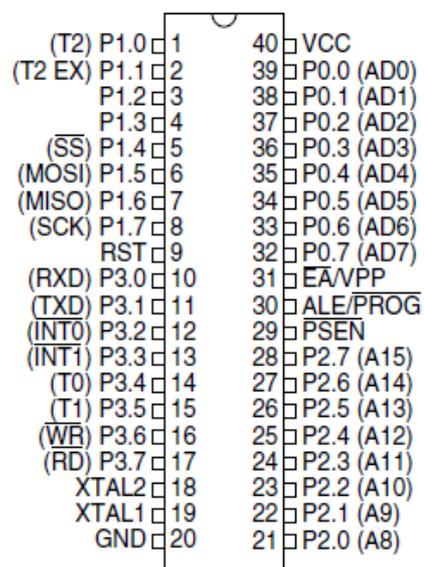


Figura 15 - Pinos do microcontrolador AT89S8253

ANEXO B – CONFIGURAÇÃO DOS PINOS DO ADC0804

Pequenas alterações, com relação ao datasheet, para melhor clareza das conexões.

Functional Description (Continued)

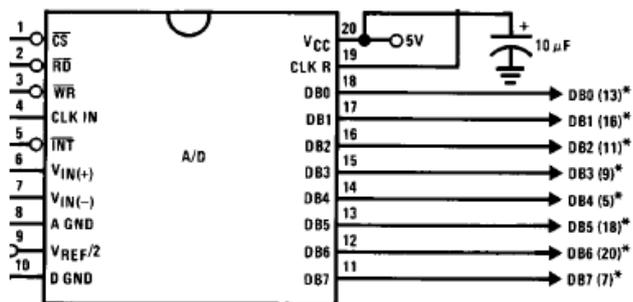
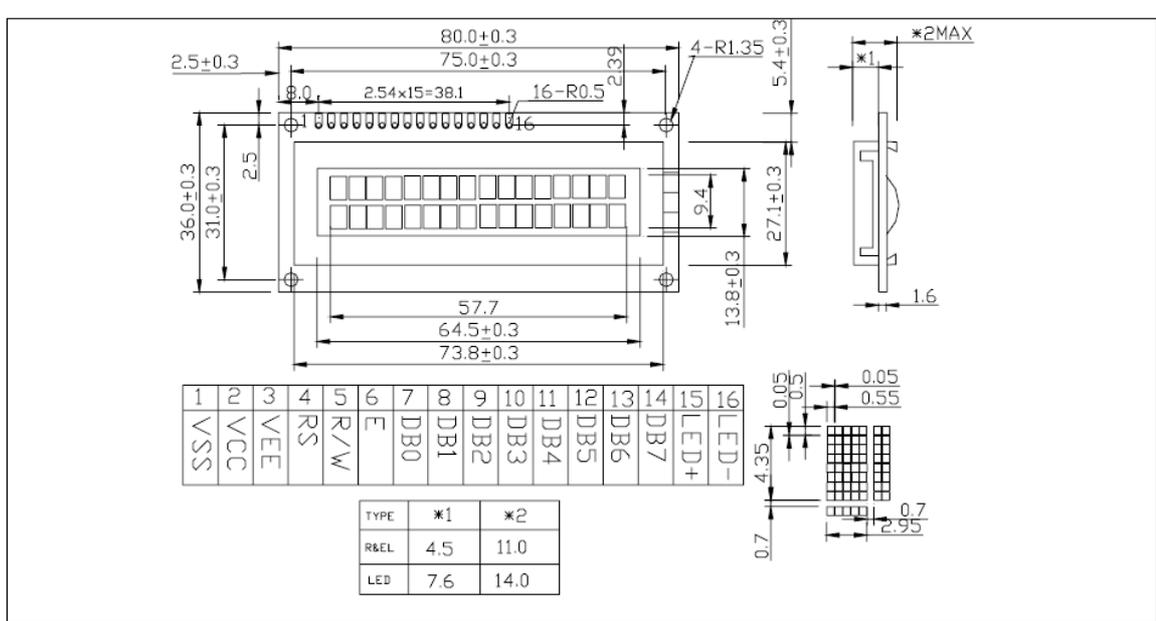


Figura 16 - Pinos do Conversor Analógico-Digital

ANEXO C – CONFIGURAÇÃO DOS PINOS DO MÓDULO DE LCD JHD162A

Retirado diretamente do Datasheet



■ PIN CONFIGURATION

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
VSS	VCC	VEE	RS	R/W	E	DB0	DB1	DB2	DB3	DB4	DB5	DB6	DB7	LED+	LED-

Figura 17 - Pinos e dimensões do módulo JHD 162A