



Marcelo Henrique de Oliveira

Rafaela Bárbara Custódio

Thiago Abreu Fiori

TRABALHO ESTRUTURA DE DADOS – GCC 216

LAVRAS - MG

2018

TRABALHO ESTRUTURA DE DADOS

INTRODUÇÃO

Os sensores luminosos são utilizados pra diversas finalidades a partir da placa com microcontrolador e dispositivos de entrada e saída do Arduíno. Os dados obtidos a partir do sensor de luminosidade são possíveis de serem organizados utilizando-se adequadas estruturas de dados para melhor manipulação e tratamento destes valores.

OBJETIVOS

O objetivo do presente trabalho é desenvolver um circuito utilizando um sensor LDR capaz de detectar diferentes luminosidades que ao detectar luz forneça um feedback luminoso da quantidade de luz detectada através de 3 leds, sendo 1 verde, para grandes quantidades de luz recebida, 1 amarelo para valores intermediários e 1 vermelho que será acionado quando o sensor receber pouca ou nenhuma luz.

Os dados oriundos das sucessivas leituras do sensor serão organizados utilizando estruturas de dados que sejam mais adequadas, no caso optamos por utilizar listas encadeadas para armazenar as leituras, as listas são salvas ordenadas em um arquivo, optamos pelas listas devido a sua facilidade de manuseio utilizamos a Tabela Hash para as buscas, escolhemos esta estrutura devido a sua grande velocidade de acesso devido ao mapeamento dos dados na tabela, com tratamento de colisão com endereçamento aberto, onde os dados colididos são armazenados na primeira posição marcada como disponível no arquivo.

REFERENCIAL TEÓRICO

Segundo Santos (2014) o conceito Arduíno surgiu na Itália, em 2005, com objetivo de criar um dispositivo de controle para protótipos construídos de forma menos dispendiosa do que outras soluções disponíveis.

De acordo com Da Fonseca (2011) o kit de desenvolvimento Arduíno é considerado uma plataforma de computação física, onde sistemas digitais, ligados a sensores e a atuadores, são capazes de medir variáveis no ambiente físico, realizar cálculos numéricos e tomar decisões lógicas no ambiente computacional e gerar novas variáveis no ambiente físico.

A parte central do kit é um microcontrolador, o qual pode ser entendido como a evolução de um microprocessador, na qual ele é ampliado com sistemas de temporização, de aquisição e de comunicação.

O kit possui uma linguagem de programação de alto nível similar às linguagens C e C++, preservando muitas características dessas linguagens. O Arduino é acompanhado de um ambiente de desenvolvimento integrado (Integrated Development Environment ou IDE), o qual é um aplicativo de software que oferece várias facilidades para o desenvolvimento de um projeto, de forma integrada.

Nele, é possível editar o programa na linguagem de alto nível, compilá-lo para a linguagem do microcontrolador, carregar o código compilado para a memória do microcontrolador e, através das interfaces de comunicação entre o computador e o Arduino, realizar alguns testes.

O IDE do Arduino pode ser obtido gratuitamente no website do Arduino <<https://www.arduino.cc/en/Main/Software>> .

Uma característica importante é que o IDE do Arduino é multiplataforma, podendo ser instalado em diversos sistemas operacionais diferentes (Windows, GNU/Linux e MacOS).

Segundo Boylestad (1992) o sensor de luminosidade é um equipamento eletrônico capaz de regular o funcionamento de um dispositivo elétrico/eletrônico qualquer, mediante a quantidade de luz irradiada no ambiente.

As principais aplicações do sensor estão relacionadas com a economia de energia sendo devido a isso largamente utilizado em projetos de Iluminação pública, Controle de iluminação em ambientes abertos ou fechados, Ativação de dispositivos eletrônicos de pequeno porte(diretamente) ou Ativação de dispositivos elétricos de médio e grande porte (indiretamente).

A energia luminosa desloca elétrons da camada de valência para a de condução (mais longe do núcleo), aumentando o número destes, diminuindo a resistência.

Um número maior de elétrons na estrutura tem também seu nível de energia aumentado, devido à aquisição da energia entregue pelos fótons. O resultado é o aumento de elétrons livres e elétrons fracamente presos ao núcleo.

Estruturas de Dados

Segundo Laureano (2005) as estruturas de dados são formas otimizadas de armazenamento e tratamento de informações eletronicamente.

As estruturas de dados, na maioria dos casos, baseiam-se nos tipos de armazenamento vistos dia a dia, ou seja, nada mais são do que a transformação de uma forma de armazenamento já conhecida e utilizada no mundo real adaptada para o mundo computacional. Por isso, cada tipo de estrutura de dados possui vantagens e desvantagens e cada uma delas tem sua área de atuação otimizada.

Wang (2004) define a Tabela Hash como uma estrutura de dados especial que armazena as informações desejadas associando chaves tendo como principal objetivo de fazer buscas rápidas e obter o valor desejado a partir de uma chave. A tabela hash é uma generalização da noção mais simples de um arranjo comum, sendo uma estrutura de dados do tipo dicionário. Essas estruturas são especializadas em prover as operações de inserir, pesquisar e remover utilizando uma função, aplicada sobre parte da informação (chave), para retornar o índice onde a informação deve ou deveria estar armazenada. Esta função que mapeia a chave para um índice de um arranjo é chamada de Função de Hashing (Hash function).

Listas são estruturas de dados onde cada elemento contém um elo (endereço) para o próximo elemento da lista. Desta forma a lista pode ser acessada de maneira randômica, podendo ser retirados, acrescentados ou inseridos elementos no meio da lista dinamicamente. Uma lista simplesmente encadeada contém um elo (endereço) com o próximo item da lista.

MATERIAL E MÉTODOS

Para o presente projeto foram utilizados os seguintes materiais:

Arduíno UNO R3

Sensor de Luminosidade LDR 5mm

3 Led's Difusos de 5mm nas cores Vermelho, Amarelo e Verde

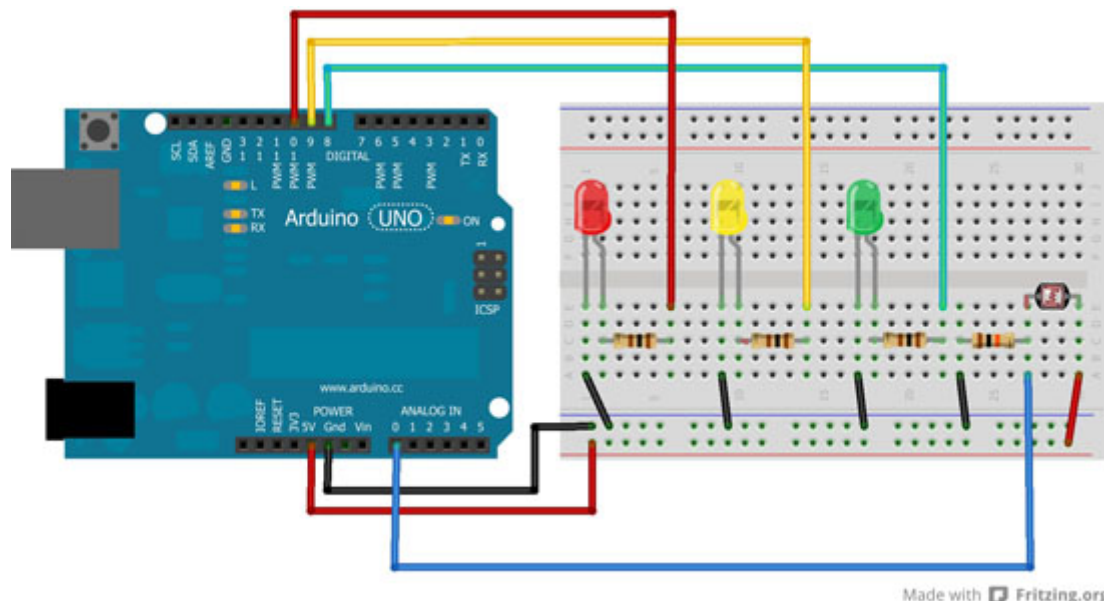
Jumpers – Kit de cabos de ligação macho/macho

Protoboard 830 pontos

3 resistores de 100Ω

1 resistor de 1MΩ

Com o arduíno totalmente desconectado de fontes de energia os componentes do circuito foram conectados a protoboard conforme ilustra a imagem.



Conectou-se o sensor de luminosidade LDR como um divisor de tensão Desta forma, a tensão na entrada analógica do Arduino ficaria alta quando o LDR estivesse sob iluminação e baixa quando o sensor fosse mantido na sombra. O sensor de luminosidade LDR, assim como um resistor comum, não tem polaridade.

Para a calibração do sensor de luminosidade foi definido antes o valor de referência onde utilizamos um Led aceso para conhecer a luminosidade do ambiente onde o sensor estava instalado, para isso utilizamos o seguinte código:

/

Calibrar sensor de luminosidade LDR

*****/

```
const int LDR = 0;
int ValorLido = 0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  ValorLido = analogRead(LDR);
  Serial.print("Valor lido pelo LDR = ");
```

```
Serial.println(ValorLido);  
delay(500);  
}
```

Após a calibração do sensor foi feito o upload para o microcontrolador, em seguida abrimos o monitor serial para fazer a leitura do sensor LDR no ambiente. Os valores de leitura do sensor variaram de zero a 1023 para os ambientes mais iluminados.

Desta forma definimos no código do programa os parâmetros para que a luz verde se acendesse em ambientes com elevada luminosidade, para níveis intermediários o led amarelo era aceso e em ambientes com iluminação reduzida ou ausente acendia-se o led vermelho.

CONSIDERAÇÕES FINAIS

REFERÊNCIAS BIBLIOGRÁFICAS

DA FONSECA, Erika GP; ALEXANDRE, S. **Tutorial sobre introdução a projetos utilizando o kit de desenvolvimento arduíno**. 2011.

SANTOS, Elio Molisani Ferreira. **Arduino: uma ferramenta para aquisição de dados, controle e automação de experimentos de óptica em laboratório didático de física no ensino médio**. 2014.

BOYLESTAD, Robert L. **“Dispositivos eletrônicos e teoria dos circuitos”**, 8ª edição, Editora Prentice Hall, 1992.

LAUREANO, Marcos. **Programando em C para Linux, Unix e Windows**. Rio de Janeiro: BRASPORT LIVROS, 2005.

WANG, X. et al. **Collisions for hash functions** md4, md5, haval-128 and ripemd. IACR Cryptology ePrint Archive, v. 2004, p. 199, 2004.