



CENTRO UNIVERSITÁRIO DE BRASÍLIA
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

RICKSON JORDAN PEREIRA DA SILVA MATOS

SENSORES E ATUADORES

BRASÍLIA – DF
2020

RICKSON JORDAN PEREIRA DA SILVA MATOS

SENSORES E ATUADORES

Trabalho de conclusão de disciplina,
apresentado como requisito parcial para
conclusão da disciplina de Sistemas
Embarcados e em Tempo Real – Centro
Universitário de Brasília – UniCEUB.
Orientador: Prof. Aderbal Botelho

BRASÍLIA – DF

2020

RICKSON JORDAN PEREIRA DA SILVA MATOS

Trabalho de conclusão de disciplina,
apresentado como requisito parcial para
conclusão da disciplina de Sistemas
Embarcados e em Tempo Real – Centro
Universitário de Brasília – UniCEUB.
Orientador: Prof. Aderbal Botelho

BRASÍLIA, 06/10/2020

BANCA EXAMINADORA

Prof(A) Aderbal Botelho

BRASÍLIA
2020

Sensores e atuadores

RICKSON JORDAN PEREIRA
DA SILVA MATOS

Resumo

Os sensores são grandes transdutores, isto é, conversores de grandezas físicas em sinais elétricos correspondentes capazes de isolar estímulos externos. Já os atuadores são os componentes que realizam a conversão das energias elétrica, hidráulica, pneumática em energia mecânica. Em geral, os sistemas de tempo real e embarcados costumam utilizar-se de tais dispositivos para controlar todo o fluxo de execução de tarefas, seja periódica ou não, como para aumentar a eficiência de seu sistema. Este trabalho visa a construção de um circuito simples com o Kit Arduino, utilizando-se apenas de uma led e um buzzer que por sua vez, serão controlados de acordo com a intensidade da luz que incide sobre o sensor de luminosidade LDR(fotoresistor).

Palavras-chaves: Sensores, Atuadores, Transdutores, Led, Buzzer, Arduino.

Sensors and Actuators

Abstract

The sensors are large transducers, that is, converters of physical quantities in corresponding signals capable of isolating external stimuli. Actuators are the components that convert electrical, hydraulic, pneumatic energy into mechanical energy. In general, real-time and embedded systems often use such devices to control the entire flow of task execution, whether periodic or not, and to increase the efficiency of your system. This work aims to build a simple circuit with the Arduino Kit, using only a led and a buzzer, which in turn will be controlled according to the intensity of the light that falls on the LDR luminance sensor (photoresistor).

Keywords: Sensors, Actuators, Transducers, Led, Buzzer, Arduino.

SUMÁRIO

1. Introdução.....	5
2. Objetivos.....	6
a. Objetivo Geral	
b. Objetivos Específicos	
3. Fundamentação Teórica.....	7
4. Metodologia.....	9
5. Explicando o código.....	11
6. Considerações Finais.....	15
7. Referências Bibliográficas.....	16

1. INTRODUÇÃO

Os sensores e atuadores estão cada vez mais presentes e integrados aos sistemas de tempo real e embarcados e atuam gerando mais eficiência do sistema e controle do fluxo de execução. O carro, por exemplo, é composto de diversos tipos de sensores e atuadores, tais como: velocímetro(sensor), medidor de combustível(sensor), indicador de combustível baixo(sensor/alarme), acelerador(atuador), câmbio(atuador) e volante(atuador).

Um sistema é classificado como embarcado quando este é dedicado a uma única tarefa e interage continuamente com o ambiente a sua volta por meio dos sensores e atuadores (STUART, 2005). Sistemas de tempo real são sistemas que monitoram e controlam o ambiente, também associados aos sensores que fazem todo controle de dados do ambiente do sistema, e aos atuadores que mudam o ambiente do sistema de alguma maneira conforme a necessidade.

Os sensores geralmente podem ser classificados de diversas formas, como: externos ou internos, de acordo com o modo como gerem a energia envolvida no processo de sensoriamento ou ainda pelo tipo de grandeza que avaliam. Já os atuadores, podem ser classificados de acordo com o tipo de energia que utilizam, como: Hidráulicos, Pneumáticos e Eletromagnéticos.

2. OBJETIVOS

a. Objetivo Geral:

Compreender o funcionamento dos sensores e atuadores através da fundamentação teórica e a experiência prática com os componentes eletrônicos presentes no kit Arduino e testar todas as interações com os seus sensores e atuadores.

b. Objetivos Específicos:

Para que o objetivo geral seja alcançado, será necessário cumprir alguns objetivos específicos, como os citados abaixo:

1. Definir um valor limite de disparo adequado conforme a luminosidade ambiente;
2. Acender a luz do led quando passar valor limite de disparo;
3. Acionar campainha do buzzer quando passar valor limite de disparo;
4. Mostrar valor da luminosidade medido pelo sensor.

3. FUNDAMENTAÇÃO TEÓRICA

Os sensores são dispositivos conversores sensíveis à algum tipo de energia ambiente que pode ser luminosa, térmica, cinética, relacionando sempre informações sobre grandezas físicas que precisa ser mensurada(medida) em sinais elétricos correspondentes. Já os atuadores são componentes capazes de realizar a conversão da energia elétrica, hidráulica, pneumática em energia mecânica.

Um robô, por exemplo, é equipado com sensores capazes de monitorar a velocidade que ele se move, a posição que ele se encontra, a localização de uma peça a ser manipulada, as dimensões da peça, a aproximação de um ser humano, e o impacto com um obstáculo. (MORAES, 2003).

Segundo Isabel Ribeiro (2004), os sensores poder ser classificados como:

Sensores externos que lidam com a observação de aspectos do mundo exterior ao robô. Sensores de contato, de proximidade, de força, de distância, de laser, de ultra-som, de infravermelhos e sensores químicos são exemplos de sensores externos (RIBEIRO, 2004).

Sensores internos que fornecem informação sobre os parâmetros internos do robô, como por exemplo, a velocidade ou sentido de rotação de um motor, ou o ângulo de uma junta. Potenciômetros, codificadores (encoders) e os sensores inerciais (incluindo acelerômetros, giroscópios, inclinômetros e bússolas), são exemplos de sensores internos (RIBEIRO, 2004).

Os sensores também podem ser classificados de acordo com o modo como gerem a energia envolvida no processo de sensoriamento.

Sensores ativos medem através da emissão de energia para o ambiente ou por modificarem o ambiente, como por exemplo sensores laser, sensores de ultra-som e os sensores de contato.

Sensores passivos não emitem energia, mas pelo contrário, recebem energia do ambiente. Um exemplo de sensor passivo são os sensores ópticos que recebem do ambiente a luminosidade necessária para o acionamento dos mesmos (RIBEIRO, 2004).

Outra classificação agrupa os sensores pelo tipo de grandeza que avaliam. Assim, há sensores de distância (laser, ultra-som), sensores de posicionamento absoluto do robô (por exemplos sistemas de GPS), sensores ambientais (que indicam temperatura, umidade), sensores inerciais (que indicam componentes diferenciais da posição do robô como, por exemplo, aceleração ou velocidade) (RIBEIRO, 2004).

Nos atuadores que realizam a conversão de energias, existe uma potência mecânica gerada por eles que é enviada aos elos através dos sistemas de transmissão para que eles se movimentem. Ainda é possível classificá-los de acordo com o tipo de energia que utilizam, a escolha do tipo de atuador mais indicado está diretamente relacionada com essa classificação.

Atuadores Hidráulicos: utilizam um fluido à pressão para movimentar o braço. São utilizados em robô que operam grandes cargas, onde é necessária grande potência e velocidade, mas oferecem baixa precisão (FELIZARDO; BRACARENSE, 2005).

Atuadores Pneumáticos: utilizam um gás à pressão para movimentar o braço. São mais baratos que os hidráulicos, sendo usados em robôs de pequeno porte. Oferecem baixa precisão, ficando limitados a operações do tipo pega-e-coloca (do inglês, pick and place) (FELIZARDO; BRACARENSE, 2005).

Atuadores Eletromagnéticos: motores elétricos (de passo, servos, Corrente Contínua ou Corrente Alternada) ou músculos artificiais, usados em robôs de pequeno e médio porte (FELIZARDO; BRACARENSE, 2005).

4. METODOLOGIA

Definição do problema:

Utilize os componentes indicados e construa o circuito esquemático da Figura. O circuito deve disparar um alarme sonoro quando a luminosidade baixar de um valor determinado no experimento. Para ativar a campanha, considere o envio de um sinal para o pino digital número 8. Conecte o LED no circuito da Figura 5 e escreva o valor da luminosidade medido pelo sensor.

Nesse projeto, ao todo, serão utilizados os seguintes componentes:

- Fotocélula LDR 5mm: Sensor de luminosidade.
- Buzzer 5V: Campanha que dispara um som quando ativada.
- Corrente: $\leq 42\text{mA}$;
- Som de saída: $\geq 85\text{DB}$;
- Frequência de Resonância: $2300 \pm 300\text{HZ}$;
- Temperatura de Operação: $-20^{\circ}\text{C} \sim +45^{\circ}\text{C}$;
- Temperatura de armazenamento: $-20^{\circ}\text{C} \sim +60^{\circ}\text{C}$;
- Sinalizador piezoelétrico de 12mm.
- Resistor Filme de Carbono $10\text{k}\Omega$
- Resistor Filme de Carbono $10\text{k}\Omega$
- Resistor Filme de Carbono 390Ω
- LED

Para resolução do exercício, utilizei o Tinkered para montagem do circuito (conforme Figura 1) com o Arduino UNO e seus respectivos sensores e atuadores citados no problema, como também utilizei a linguagem C++ para elaborar um código capaz de integrar o circuito e realizar todas as ações devidas para resolução. Esse código está hospedado no GIT com os devidos comentários explicando cada parte dele, além de um roteiro dividido por sessões.

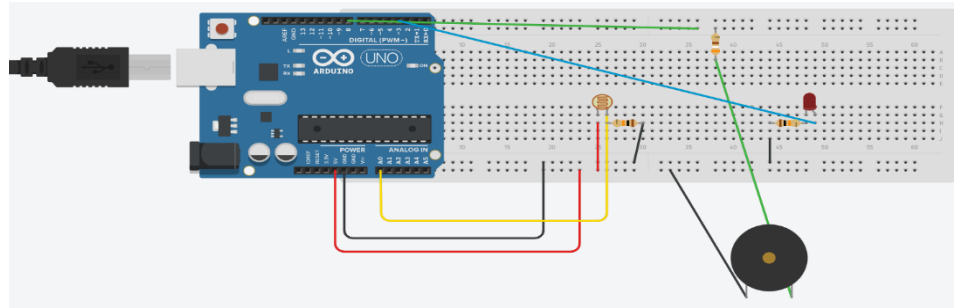


Figura 1

5. Explicando o código

1. Começamos declarando as constantes e as variáveis do projeto.

```
1 //Atribui um "valor limite de disparo" para achar um valor adequado,  
2 //ligando e desligando o LED quando passar a mão sobre o fotoresistor (LDR).  
3 int triggerLimit = 930;  
4  
5 // Ligue o LED ao pino digital 3  
6 const int ledPin = 3;  
7  
8 // Ligue o Piezo no pino 8  
9 const int buzzerPin = 8;  
10  
11 const int freq=5;  
12 // O fotoresistor (LDR) é conectado ao pino analógico A0  
13 int ldrPin = A0;  
14  
15 // Armazena e inicializa o valor de leitura analógica  
16 int ldrValue = 0;  
17
```

1.1. Utilizamos as constantes e variáveis tipo **"int"** que podem contar números inteiros de -32767 até 32767.

1.2. A constante do tipo inteiro nomeada **buzzerPin** refere-se ao buzzer que deverá estar conectado à porta digital 8 do microcontrolador Arduino.

1.3. A constante do tipo inteiro nomeada **ldrPin** refere-se ao Sensor de luminosidade LDR que deverá estar conectado à porta analógica A0.

1.4. A constante do tipo inteiro nomeada **ledPin** refere-se ao Led vermelho que deverá estar conectado à porta 3 do microcontrolador Arduino.

1.5. Declaramos e inicializamos a variável **ldrValue** como 0 (zero).

1.6. Declaramos e inicializamos a constante **freq** como 5 (esse valor deve ser diferente de zero e poderá ser alterado para aumentar ou diminuir a frequência de bips do alarme).

2. Através da estrutura **void setup()**, definimos:

```

18 void setup() {
19     // Inicia a comunicação serial com uma taxa de transmissão de 9600 baud rate
20     Serial.begin(9600);
21
22     // Define o fotoresistor como uma entrada
23     pinMode(ldrPin, INPUT);
24
25     // Define o LED como uma saída
26     pinMode(ledPin, OUTPUT);
27
28     // Define o Piezo como uma saída
29     pinMode(buzzerPin, OUTPUT);
30 }

```

2.1. A função **Serial.begin()** serve para dizer ao Arduino que será coletado dados para o computador a partir da porta serial e o cabo USB. O número entre os parênteses significa qual é a taxa de dados que a placa vai se comunicar com o computador. Utilizaremos a taxa padrão de 9600bps (bits-per-second).

2.2. Observe que portas analógicas não precisam ser definidas, pois por padrão, já são definidas como INPUT. Entretanto, você pode utilizar a linha de código: **pinMode(ldrPin, INPUT);**

2.3. Define-se a constante **ledPin** como saída do controlador Arduino (OUTPUT) conectada à porta digital 3.

2.4. Define-se a constante **buzzerPin** como saída do controlador Arduino (OUTPUT) conectada à porta digital 7.

3. Através da estrutura **void loop()**, obtemos:

```

33 void loop() {
34
35     // Lê o valor atual do fotoresistor
36     ldrValue = analogRead(ldrPin);
37
38     // Se o valor da luminosidade estiver abaixo do valor "limite de disparo",
39     // então o LED liga, caso contrário o LED permanece desligado.
40     if (ldrValue < triggerLimit) {
41         digitalWrite(ledPin, HIGH);
42
43         //Toca o alarme
44         tone(buzzerPin,300); // toca um tom de 300 Hz do piezo
45         delay(30); // espera alguns segundos
46         noTone(buzzerPin); // interrompe o tom do buzzer
47         digitalWrite(ledPin,LOW); // apaga a led
48         delay(ldrValue/freq); /* espera agora a quantidade de milisegundos
49         em ldrValue */
50     }
51     else {
52         digitalWrite(ledPin,LOW); // apaga a led
53         noTone(buzzerPin); // interrompe o tom do buzzer
54     }
55
56     // Imprime as leituras do sensor no monitor serial da IDE do Arduino
57     Serial.print ("Leitura atual do sensor: ");
58     Serial.println(ldrValue);
59     delay(130);
60 }

```

3.1. A variável **ldrValue** receberá os valores lidos e atualizados diretamente pelo pino analógico onde está conectado o sensor LDR, através da função **analogRead()** que faz a conversão de analógico para digital. Esta leitura é feita pelo **ADC** (Analog to Digital Converter - conversor analógico para digital) sem tratamento nenhum. A variável foi definida localmente como tipo inteiro (int), e portanto, vai de 0 a 1023, ou seja, possui $2^{10} = 1024$ valores inteiros (referente à resolução de 10 bits do **ADC** para controladores Arduino UNO, Mega e Leonardo). Assim, quando o sensor LDR não estiver recebendo pouca ou nenhuma luz do ambiente, o valor lido será próximo de zero, e quando sensor receber muita luz, o valor será próximo de 1023, fazendo assim a leitura da luminosidade de um ambiente.

Observação: Nesse circuito, o sensor LDR varia de 0V a 5V (leitura analógica), ou seja, de 0 a 1023 quando convertido em leitura digital através do **ADC** do controlador Arduino.

Quanto mais incidência de luz o LDR receber, menor será a resistência, portanto mais corrente elétrica irá fluir e mais alto será o valor lido. E quanto menos incidência de luz o LDR receber, maior será a resistência, então menos corrente elétrica irá fluir e menor será o valor lido.

3.2. Utilizamos a estrutura condicional **If (ldrValue < triggerLimit)**. Portanto, se a variável **ldrValue** for menor que 930 (número que utilizamos como referência quando o sensor estiver recebendo pouca luz) o led vermelho acenderá e o alarme será acionado, conforme explicação a seguir:

3.2.1. A função **digitalWrite(ledPin, HIGH)** faz com que acenda o led vermelho.

3.2.2. Dispara o alarme, conforme explicação a seguir:

3.2.3.1. A função **tone()** define um tom para o buzzer. Vamos utilizar **tone(buzzerPin,1000)** que gera um tom com frequência de 300Hz. (Você pode alterar este valor definindo tons diferentes para o buzzer).

3.2.3.2. Através da função **digitalWrite(ledPin, HIGH)** acendemos o Led vermelho.

3.2.3.3. Através da função **delay(30)** esperamos 30ms. Este valor define o tempo que o led e o buzzer ficaram ativos, gerando um tipo de bip.

3.2.3.4. Interrompemos o som do buzzer através da função **noTone()**.

3.2.3.5. Através da função **digitalWrite(ledPin, LOW)** apagamos o Led vermelho.

3.2.3. A função **delay(ledrValue/freq)** define o intervalo entre os bips. Quanto maior a incidência de luz sobre o sensor, menor será o valor de **ledrValue** e, portanto, menor o intervalo entre os bips, fazendo com que eles sejam emitidos de forma mais rápida. Quanto menor for a incidência de luz sobre o sensor, maior será o valor lido e, portanto, maior o intervalo entre os bips, portanto, emitidos de forma mais lenta.

3.2.4. Já a constante **freq** faz com que o intervalo entre os bips sejam maiores ou menores, dependendo do seu valor. Quanto maior for o seu valor, menor será o tempo de espera. Portanto os bips serão emitidos de forma mais rápida.

3.3. Escrevemos na tela do **Monitor Serial** o valor da variável **ldrValue** através do comando **Serial.println()**. O comando **println()** diz ao monitor que se deve pular uma linha após escrever o valor definido entre parêntesis.

3.4. Através da função **delay(130)**, definimos um tempo de espera de 130 ms entre cada ciclo do loop.

6. Considerações Finais

Os sensores e atuadores são dispositivos cada vez mais utilizados nos sistemas de tempo real e embarcados, pois possuem papel fundamental na leitura, tratamento, e uso das informações provenientes dos dados coletados.

Nesse trabalho, podemos compreender a importância e relevância do uso desses dispositivos integrados aos sistemas, bem como entender o seu funcionamento e prática aplicado ao mundo real transformando sinais elétricos e força mecânica as informações obtidas do meio físico.

7. Referências Bibliográficas

- BRAGA, Newton C. Como funcionam os encoders mecatrônica fácil. São Paulo, n.50, p.16-19, out. 2009.
- MATIAS, Juliano. Encoders. Mecatrônica Atual, São Paulo, n.3, p.36-42, abr. 2002
- FELIZARDO, Ivanilza Felizardo; BRACARENSE, Alexandre Queiroz. Processos de Soldagem - Processos Mecanizados e Automatizados. Apostila. 2005. Disponível em: <http://ivanilzafe.dominiotemporario.com/soldagem_20.html>. Acessado em 05 de outubro de 2020.
- MORAES, Airton Almeida de Moraes. Robótica. Departamento de Meios Educacionais e Gerência de Educação, Diretoria Técnica do SENAI-SP. Serviço Nacional de Aprendizagem Industrial, 2003. Apostila. Disponível em <<http://www.adororobotica.com/RBSENAI.pdf>>. Acesso em: 05 de outubro de 2020.
- NIKU, SAEED B. Introdução à robótica; análise, controle, aplicações. 2.ed., 2013.
- Ball, Stuart. – “Embedded Microprocessor Systems: Real World Design”, 3rd edition, Editora: MCPros, EUA, 2005.
- OGATA, Katsuhiko. Engenharia de controle moderno. Tradução de Prof. Bernardo Severo. Rio de Janeiro: LTC, 1998. RIBEIRO, M. Isabel. Sensores em robótica, Enciclopédia Nova Activa Multimédia, Volume de Tecnologias, p.228-229, Portugal, 2004.
- ANGELO. Projeto 30 - Sensor de luminosidade LDR com sinalizador de Leds, 2003. Disponível em < <https://bityli.com/dROWnf> >. Acesso em: 05 de outubro de 2020.
- COSTA, Rodrigo. Como ligar um LED utilizando sensor LDR (fotoresistor) com Arduino. Tecnoblog, 2018. Disponível em: < <https://eletronicaparatodos.com/como-ligar-um-led-utilizando-sensor-ldr-fotoresistor-com-arduino//>>. Acesso em: 06 de outubro de 2020.
- GAMA, Kiev. Projeto de Sistemas de Tempo Real. Disponível em:< https://www.cin.ufpe.br/~kiev/IF682/13_Projeto_Sistemas_Tempo_Real.pdf>. Acesso em: 06 de outubro de 2020.