

INSTITUTO FEDERAL

Santa Catarina

Câmpus Tubarão

Timers, Interrupções e Sensores

Curso Superior de Tecnologia em Sistemas Embarcados

Professor: Fernando Silvano Gonçalves

fernando.goncalves@ifsc.edu.br

Maio de 2023

Cronograma

Encontro	Data	Nº Aulas	Conteúdo
1	29-fev.	04	Recepção e Apresentação do Unidade / Apresentação do Plano de Ensino / Avaliação Diagnóstica / Introdução a sistemas embarcados
2	02-mar.	04	Conceitos e Características e Aplicações de Sistemas Embarcados / Histórico de Sistemas Embarcados / Práticas com Arduino
3	07-mar.	04	Microcontroladores, Microprocessadores / Periféricos / Introdução ao Arduino / Introdução ao C
4	14-mar.	04	Introdução à Linguagens de Programação / Comunicação Serial
5	21-mar.	04	Entrada de Dados via Serial
6	28-mar.	04	Linguagem C para Arduino
7	04-abr.	04	Variáveis e Operadores
8	11-abr.	04	Estruturas Condicionais
9	18-abr.	04	Práticas com Arduino e entradas e saídas analógicas e digitais
10	25-abr.	04	Práticas com Arduino e entradas e saídas analógicas e digitais

Cronograma

Encontro	Data	Nº Aulas	Conteúdo
11	02-mai.	04	Avaliação 01
12	09-mai.	04	Avaliação 01
13	16-mai.	04	Correção da Avaliação / Introdução a estruturas de repetição
14	18-mai.	04	Estruturas de Repetição / Comunicação I2C / Display 16x2
15		04	Timers, Interrupções / Sensores (LDR)
16		04	Comunicação I2C / Display LCD / Sensores (Ultrassônico, DHT-22)
17		04	PWM, Ponte H, Atuadores (Servomotor, Motor DC, Relé)
18		04	Comunicação Bluetooth /Práticas com Sensores e atuadores
19		04	Práticas com Sensores e atuadores
20		04	Conselho de Classe / Atividades de Recuperação e Encerramento da UC
		80	

Pauta

- Timers;
- Interrupções;
- Práticas com Timers e Interrupções;
- Sensores;
- Sensor de Luminosidade (LDR);

Timers

Timers

Timers

- ❑ Um timer nada mais é do que um contador que é incrementado a cada intervalo de tempo;
- ❑ Os timers funcionam como um relógio que pode ser usado para contar o tempo, medir a duração de certos eventos, entre outras aplicações;
- ❑ O Arduino Uno possui os timers: **timer0**, **timer1** e **timer2**;
 - ❑ **Timer0** e **timer2** são contadores de 8bits, ou seja, contam de 0 a 255;
 - ❑ **Timer1** é um contador de 16bits, conta de 0 a 65535.

Timers

- ❑ **Timer0:** Utilizado pelo Arduino para funções como `delay()`, `millis()` e `micros()`. Então não se deve utilizar esse timer para evitar comprometer essas funções;
- ❑ **Timer1:** No Arduino UNO esse é o timer utilizado pela biblioteca de controle de servos;
- ❑ **Timer2:** Esse timer é utilizado pela função `tone()`;

Configurando Timers

- ❑ Para facilitar a configuração do timer vamos utilizar a biblioteca disponível em: <http://playground.arduino.cc/Code/Timer1>
- ❑ Primeiro passo é baixar as bibliotecas e colocá-las na pasta “Arduino/libraries”;
- ❑ A partir daí o timer pode ser configurado e utilizado;


```
#include "TimerOne.h"
```

```
int i;
```

```
void setup() {
```

```
    pinMode(8, OUTPUT);
```

```
    pinMode(9, OUTPUT);
```

```
    Timer1.initialize(1000000);
```

```
    Timer1.attachInterrupt(processaTimer);
```

```
    i = 0;
```

```
}
```

```
void processaTimer(){
```

```
    switch(i){
```

```
        case 0:
```

```
            digitalWrite(8, HIGH);
```

```
            break;
```

```
        case 1:
```

```
            digitalWrite(8, LOW);
```

```
            break;
```

```
        case 2:
```

```
            digitalWrite(9, HIGH);
```

```
            break;
```

```
        case 3:
```

```
            digitalWrite(9, LOW);
```

```
            i = -1;
```

```
            break;
```

```
    }
```

```
    i++;
```

```
}
```

Prática do Semáforo com Timer

- ❑ Construa um circuito com 3 leds simulando um semáforo, faça a sua ativação utilizando Timer. Esse deve obedecer os seguintes critérios:
 - ❑ A luz verde deve ficar acesa por 2 segundos;
 - ❑ A luz amarela deve ficar acesa por 1 segundo;
 - ❑ A luz vermelha deve ficar acesa por 3 segundos.
- ❑ Os leds devem ser acesos na sequência correta de funcionamento do semáforo de trânsito;
- ❑ Não deve haver mais de um led aceso ao mesmo tempo.



Prática do Semáforo com Timer II

- ❑ Construa um circuito com 6 leds simulando um cruzamento de semáforos, faça a sua ativação utilizando Timer. Esse deve obedecer os seguintes critérios:
 - ❑ A luz verde deve ficar acesa por 2 segundos;
 - ❑ A luz amarela deve ficar acesa por 1 segundo;
 - ❑ A luz vermelha deve ficar acesa por 3 segundos.
- ❑ Os leds devem ser acesos na sequência correta de funcionamento do semáforo de trânsito;
- ❑ Não deve haver mais de um semáforo aberto ao mesmo tempo.



Interrupções

Interrupções

Interrupções

- Quando estamos executando uma tarefa, algumas vezes temos que interromper a sua execução para resolver outra tarefa importante;
- Quando estamos monitorando um determinado componente, muitas vezes o que nos importa é a sua mudança de estado, como o clique de um botão;
- Nestes casos podemos fazer uma pausa em nossa aplicação e só depois retornar do ponto que se parou.
- Uma interrupção tem dois pontos chave:
 - Condição de interrupção;
 - Função a ser executada;

Interrupções - Sintaxe

`attachInterrupt(digitalPinToInterrupt(pino), ISR, modo);`

- ❑ **Pino:** número do pino para interrupção;
- ❑ **ISR:** função a ser chamada quando detectada a interrupção;
- ❑ **Modo:** modo de ativação da interrupção;
 - ❑ **LOW:** aciona a interrupção quando o estado do pino for LOW;
 - ❑ **CHANGE:** aciona a interrupção quando houver mudança no estado do pino;
 - ❑ **RISING:** aciona a interrupção quando o estado do pino for de HIGH para LOW apenas;
 - ❑ **FALLING:** aciona a interrupção quando o estado do pino for HIGH;

No Arduino UNO somente as portas 2 e 3 estão disponíveis para uso de interrupções.

Interrupções - Exemplo

```
#include "TimerOne.h"
```

```
#define pb 3
```

```
#define led 10
```

```
int ledState;
```

```
int i;
```

```
void interrupt(){
```

```
    Serial.println("Interrupção");
```

```
    i++;
```

```
}
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    attachInterrupt(digitalPinToInterrupt(pb), interrupt, RISING);
```

```
    pinMode(led, OUTPUT);
```

```
    ledState = LOW;
```

```
    i = 0;
```

```
}
```

```
void loop(){
```

```
    if(i){
```

```
        if(ledState == LOW){
```

```
            ledState = HIGH;
```

```
        } else {
```

```
            ledState = LOW;
```

```
        }
```

```
        digitalWrite(led, ledState);
```

```
        i = 0;
```

```
    }
```

```
}
```

Atividade Com Botão e Leds I

- ❑ Crie um circuito com um led e um botão. Faça a leitura do botão por meio de interrupção, ao pressionar o botão intercalar o led entre ligado e desligado;
- ❑ Adicione ao circuito anterior, 3 leds e mais um botão. Configure o segundo botão também para ser reconhecido por interrupção. Ao acionar o botão 1 você deve intercalar os leds 1 e 2 entre ligado e desligado, assim como, ao acionar o segundo botão intercale os leds 3 e 4 entre ligado e desligado. Os dois leds de cada botão não podem estar acesos ao mesmo tempo.

Atividade Com Botão e Leds II

- ❑ Crie um circuito com cinco (5) leds e dois (2) botões.
- ❑ Faça a leitura dos botões por meio de interrupção;
- ❑ Os leds devem iniciar piscando da direita para esquerda a cada 1s.
- ❑ Ao pressionar o primeiro botão você deve alterar o sentido que os leds estão piscando.
- ❑ Ao pressionar o segundo botão, você deve alternar a frequência com que os leds são acionados, sendo respectivamente: 0,25s - 0,5s - 1,0s - 1,5s.

Sensores

Sensores

Sensores

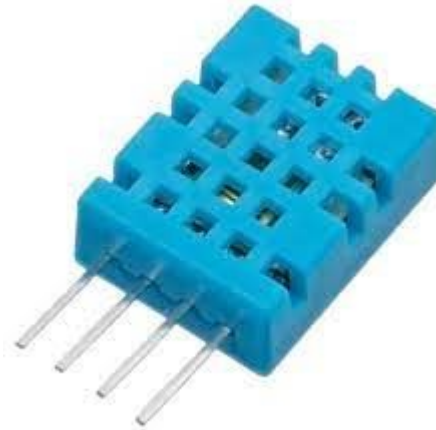
- ❑ Sensores são dispositivos que detectam magnitudes físicas, químicas e biológicas;
- ❑ Compõem o meio da relação entre os processos industriais e os circuitos eletrônicos responsáveis por controlá-los e monitorá-los;
- ❑ Usualmente transformam grandezas físicas em um sinal elétrico de tensão ou corrente;

Sensores

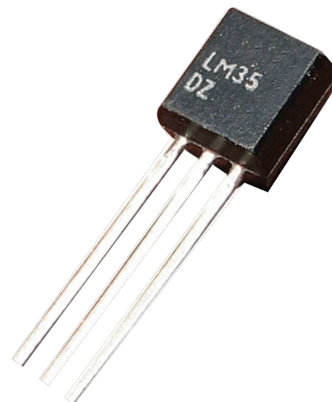
❑ Detectam as seguintes variáveis:

- Variáveis mecânicas;
- Variáveis térmicas;
- Variáveis elétricas;
- Variáveis magnéticas;
- Variáveis ópticas;
- Variáveis químicas ou moleculares.

Sensores



Copyright Sérgio F. Lima



Light Dependent Resistor - LDR

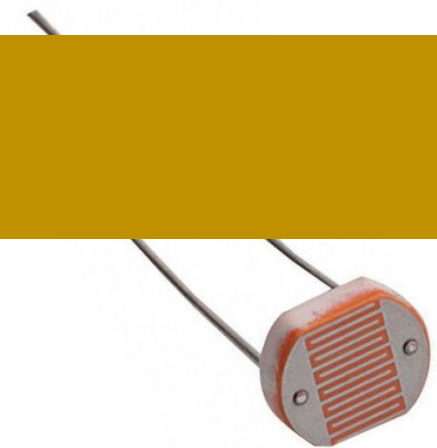
Light Dependent Resistor

Fotocélulas

- ❑ Sensor de luz;
- ❑ Quando a luminosidade está abaixo de um valor de referência aciona a lâmpada ou dispositivo conectado a ela;
- ❑ LDR é responsável por medir a luminosidade.

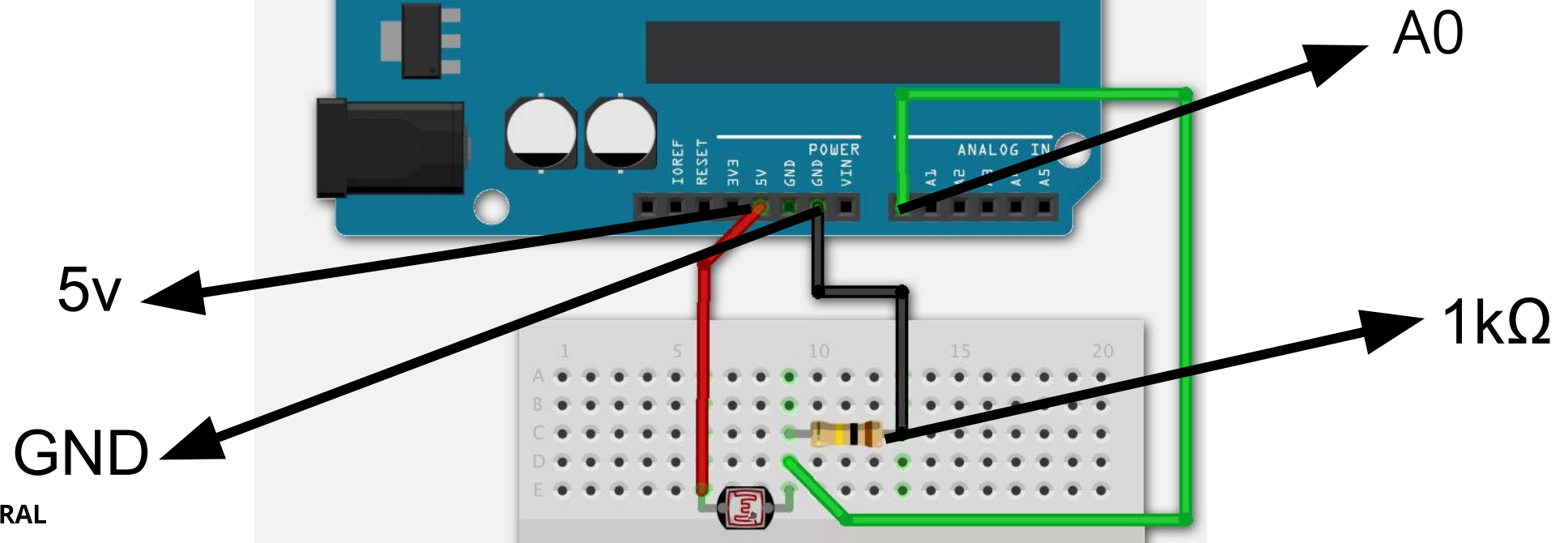


Light Dependent Resistor - LDR



- ❑ Sigla do inglês Light-Dependent Resistor;
- ❑ Quanto maior a luz que incide nesse componente, menor é a sua resistência;
- ❑ Constituído de um semicondutor de alta resistência, que ao receber uma grande quantidade de fótons oriundos da luz incidente, ele absorve elétrons que melhoram sua condutibilidade, reduzindo assim sua resistência.

Light Dependent Resistor - LDR



Realizando Leitura do LDR

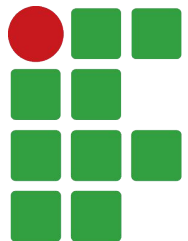
```
#define ldrPin A0 //pino de conexão do LDR
int ldrValue = 0; //Valor lido do LDR

void setup() {
    Serial.begin(9600); //inicialização da porta serial
    pinMode(ldrPin, INPUT); //definição da porta de conexão do LDR como entrada
}

void loop(){
    ldrValue = analogRead(ldrPin); //Leitura do LDR entre 0 e 1023
    Serial.println(ldrValue); //imprime na porta serial a leitura do LDR
    delay(500);
}
```

Praticando com LDR

- ❑ Modifique a prática anterior onde ao acionar um dos botões seja exibido na tela a leitura da luminosidade obtida no LDR;
- ❑ Adicione 2 leds ao circuito e conforme a variação da luminosidade faça o acionamento dos leds. Você deve iniciar com os leds apagados e ir acendendo conforme a variação da luminosidade;



INSTITUTO FEDERAL

Santa Catarina
Câmpus Tubarão

Obrigado!

Fernando Silvano Gonçalves

fernando.goncalves@ifsc.edu.br

se.cst.tub@ifsc.edu.br