

Interrupções

Interrupção em microcontroladores

Virgínia Sátyro | Engenharia de Controle e Automação |2019

# Introdução

Imagine que você esteja estudando e de repente seu telefone toca. Possivelmente, você irá parar o que está fazendo e atenderá o telefonema e, assim que encerrar a chamada, você retornará ao ponto que estava estudando.

Quando estamos executando uma tarefa, muitas vezes precisamos interrompe-la para resolver outra tarefa importante para só depois retornar ao ponto que se parou. Esse é o conceito básico de interrupção, que é utilizado com frequência na programação de microcontroladores.

Uma interrupção tem dois pontos chave:

* Condição de interrupção: é a condição que indica uma interrupção. Ela avisa ao programa que é a hora de executar uma tarefa extraordinária. No caso do exemplo, essa condição é o toque do telefone;
* Função a ser executada: quando algo indica a interrupção, temos que executar uma lista de instruções referentes a essa interrupção. No exemplo dado, a função extraordinária é atender o telefone.

## usando interrupções

Interrupções são úteis para realizar funções automaticamente em programas de microcontroladores, e podem ajudar a resolver problemas de temporização. Tarefas que utilizam interrupção podem incluir a leitura de um codificador rotativo, ou monitorar entradas de dados pelo usuário.

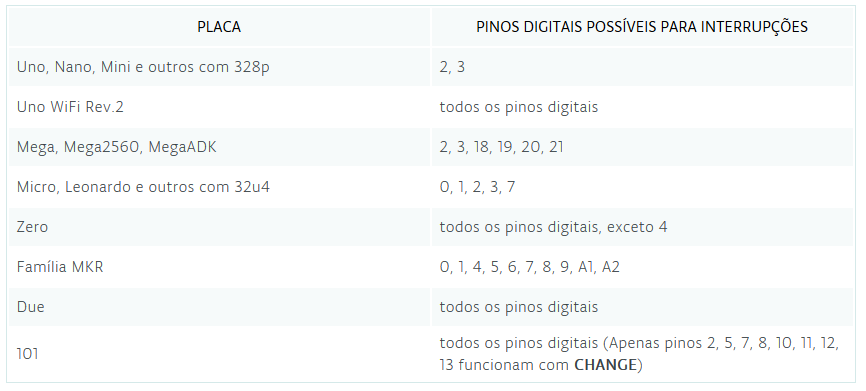
Supondo que queremos ter certeza que um programa sempre captura os pulsos de um codificador rotativo, sem nunca perder um pulso, seria muito complicado criar um programa que pudesse fazer qualquer outra coisa além disso, porque o programa precisa checar constantemente os pinos do codificador, para capturar os pulsos exatamente quando eles ocorrerem. Outros sensores são dinâmicos também, como um sensor de som que pode ser utilizado para capturar um clique, ou um sensor infravermelho para capturar a queda de uma moeda. Nessas situações, usar uma interrupção pode permitir o microcontrolador trabalhar em outras tarefas sem perder a interrupção.

# Arduino

## pinos digitais com interrupções

**attachInterrupt()** – Interrupções externas

O primeiro parâmetro de **attachInterrupt()** é o número da interrupção. É recomentado usar **digitalPinToInterrupt(pino)** para converter o número do pino digital para o número específico da interrupção. Por exemplo, ao utilizar o pino 3, é necessário passar **digitalPinToInterrupt(3)** como o primeiro parâmetro de **attachInterrupt().**



## notas e advertências

Dentro da função associada, **delay()** não funcionará e o valor retornado por **millis()** não será incrementado. Dados recebidos via serial podem ser perdidos enquanto a interrupção é executada. É recomendado declarar como **volatile** quaisquer que você modificar dentro da função associada (?).

## rotinas de serviço de interrupções (isrS)

}

## SINTAXE

* **attachInterrupt(digitalPinToInterrupt(pino), ISR, modo)**; (recomendado)
* **attachInterrupt(interrupção, ISR, modo)**; (não recomendado)
* **attachInterrupt(pino, ISR, modo)**; (não recomendado - Arduino Due, Zero, MKR1000, 101 apenas)

## PARÂMETROS

- **interrupção**: o número da interrupção (int);  
- **pino**: o número do pino *(Arduino Due, Zero, MKR1000 apenas)*  
- **ISR**: a ISR a ser chamada quando a interrupção ocorre; essa função deve não tomar nenhum parâmetro nem retornar nada. Essa função é chamada de rotina de serviço da interrupção ou ISR (do Inglês, interrupt service routine).  
- **modo**: define quando a interrupção deve ser ativada. Quatro constantes estão predefinidas como valores válidos:

* **LOW** acionar a interrupção quando o estado do pino for LOW;
* **CHANGE** acionar a interrupção quando o sempre estado do pino mudar;
* **RISING** acionar a interrupção quando o estado do pino for de LOW para HIGH apenas;
* **FALLING** acionar a interrupção quando o estado do pino for de HIGH para LOW apenas;

Placas Due, Zero e MKR1000 suportam também:

* **HIGH** acionar a interrupção quando o estado do pino for HIGH;

# Fontes:

* <https://www.arduino.cc/reference/pt/language/functions/external-interrupts/attachinterrupt/>
* <https://portal.vidadesilicio.com.br/usando-a-interrupcao-externa-no-seu-arduino/>