

ATUADORES

TRABALHANDO COM ATUADORES

Se os sensores são fundamentais para capturar as informações do mundo externo, os atuadores são fundamentais para que o Arduino aja sobre ele, seja executando ações mecânicas ou seja externalizando informações (de forma visual, sonora ou háptica). Em conjunto com algum meio de comunicação, essa é a tríade base do IOT. Os atuadores são geralmente mais simples do que os sensores em acionamento e controle, bastando muitas vezes energizá-los (seja com até 5V utilizando alguma das portas do próprio Arduino, seja de forma externa com tensão maiores e através do uso de fontes secundárias e relés). Até mesmo no controle de motores e servos motores (que necessitam de bibliotecas especiais e possuem métodos para controle de passo, velocidade e rotação), a maior dificuldade não é nem no código (muito simples) mas na transferência e controle do movimento produzido, levando a complexidade para o lado da mecânica e não da eletrônica ou da programação. Neste capítulo, conheceremos os principais atuadores, suas utilidades e formas de uso.

Atuadores de ativação simples com até 5V

Muitos dos atuadores do Arduino são de ativação simples, bastando alimentá-los com tensão provida de alguma das portas digitais. Para alguns deles isto basta e para outros, simples métodos nativos são suficientes para o seu controle. Todos estes atuadores utilizam portas digitais e, com isso, a simples mudança de estado da porta geralmente é o único controle necessário.

Leds



O primeiro atuador visto ao longo deste livro e o mais utilizado até aqui, o led é fundamental em praticamente qualquer circuito eletrônico apesar de ser muito simples (seja para mostrar que um circuito está energizado e em funcionamento, seja para mostrar informações sobre estados ou acontecimentos). O seu uso já foi amplamente visto e sabemos que o seu controle é vinculado à

simples mudança de estado de uma porta digital ou, no caso dos leds RGB, à regulação da tensão das portas digitais PWM (por meio da mudança dos seus Duty Cycles).

Aplicações: informação de circuito ligado ou desligado, informações de acontecimentos monitorados, de mudanças de estado de sensores, criação de optoacopladores para isolar 2 circuitos diferente e um ativar o outro com luz, efeitos visuais em geral.

Emissor laser



Este emissor é um gerador simples de laser, com um feixe concentrado de luz (geralmente vermelha, azul ou verde) que pode ser projetado de 1 a 15 Km de acordo com a potência. Os compatíveis com Arduino (e com diversas outras placas de prototipagem), são desenvolvidos para serem alimentados com tensão a partir dos 3,3 V.

Aplicações: sistemas de segurança de áreas em conjunto com LDR para acionar quando o feixe de laser é cortado, controle de circulação em ambientes (entrada e saída ou contagem de pessoas), ativação simples à longa distância de um circuito a partir de outro.

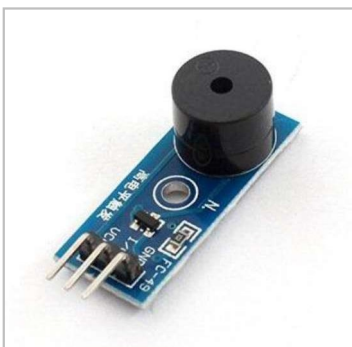
Vibracall



O vibracall nada mais é do que um pequeno motor com uma hélice de peso desequilibrado (bem mais pesada de um lado), gerando uma forte vibração ao girar. Muito comum em smartphones e joysticks de consoles mais modernos, no objetivo de chamar atenção do usuário para algum evento. Seu funcionamento também é bastante simples, vibrando quando alimentado por tensões de 3,3V à 5V, sendo possível seu controle através da mudança de estado de uma porta digital.

Aplicações: substituição dos leds em sistemas adaptados para deficientes visuais (sinais visuais trocados por sinais hápticos), sistemas que necessitem chamar atenção do usuário ou comunicar algo através de uma vibração (ou uma sequência delas).

Buzzer



O buzzer é um tipo de piezo especialmente construído para produzir vibrações com força suficiente para gerar sons. Estes sons são mais simples que os produzidos por um aut falante, geralmente beeps em uma determinada frequência. O buzzer pode ser do tipo ativo (aqueles que contam com um oscilador interno e são capazes deles próprios criarem modulações que gerem os sons) ou passivo (aqueles que não contam com oscilador interno sendo incapazes de gerar sons sem alguém que os entregue um sinal modulado, neste caso, a placa). Estes últimos são os mais comuns em prototipagem.

Aplicações: substituição dos leds em sistemas adaptados para deficientes visuais (sinais visuais trocados por sinais sonoros), sistemas que necessitem chamar atenção do usuário ou comunicar algo através de beeps (ou uma sequência deles), sistemas que necessitem

QI ESCOLAS E FACULDADES

CURSOS TÉCNICOS – EIXO TECNOLOGIA DA INFORMAÇÃO

da reprodução de uma música simples ou uma sequência sonora em resposta de uma ação.

Para utilizar um buzzer passivo no o Arduino, necessitamos de apenas dois métodos: `tone` (que gera o som) e o `noTone` (que interrompe a geração).

```
tone( <porta>, <frequencia> );
```

<porta>
porta digital à qual o buzzer está ligado

<frequencia>
Frequência em hertz de vibração do buzzer

```
noTone( <porta> );
```

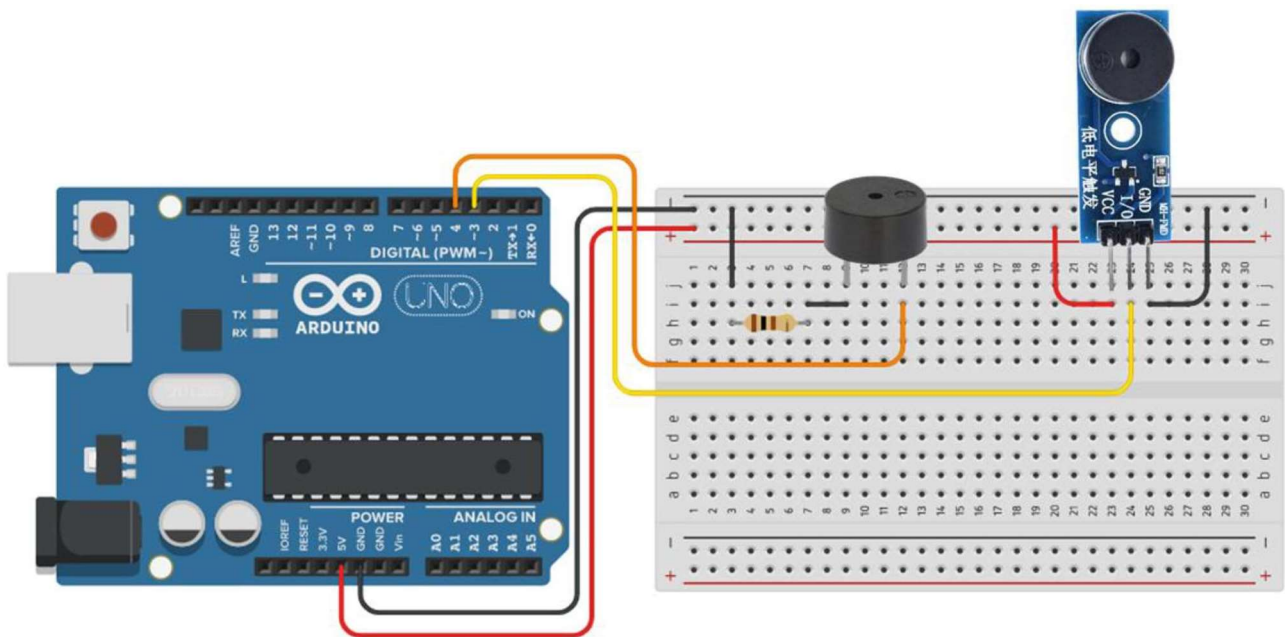
<porta>
porta digital à qual o buzzer está ligado

Quanto à frequência de vibração, cada valor gera uma percepção sonora (som) diferente. Para termos uma referência, algumas tabelas da área da física dos sons relacionam as frequências com as notas musicais produzidas. Abaixo, temos um mapeamento completo das principais notas, dentro do espectro audível para os seres humanos e com a organização por oitavas.

		Frequência em Hz									
Nota musical	-	1° Oitava	2° Oitava	3° Oitava	4° Oitava	5° Oitava	6° Oitava	7° Oitava	8° Oitava	9° Oitava	-
Dó		33	66	132	264	528	1056	2112	4224	8448	16896
Dó #		34,947	69,894	139,79	279,6	559,15	1118,3	2236,6	4473,2	8946,4	17893
Ré		37,026	74,052	148,1	296,2	592,42	1184,8	2369,7	4739,3	9478,7	18957
Ré #		39,237	78,474	156,95	313,9	627,79	1255,6	2511,2	5022,3	10045	20089
Mi	20,79	41,58	83,16	166,32	332,6	665,28	1330,6	2661,1	5322,2	10644	
Fá	22,03	44,055	88,11	176,22	352,4	704,88	1409,8	2819,5	5639	11278	
Fá #	23,33	46,662	93,324	186,65	373,3	746,59	1493,2	2986,4	5972,7	11945	
Sol	24,72	49,434	98,868	197,74	395,5	790,94	1581,9	3163,8	6327,6	12655	
Sol #	26,19	52,371	104,74	209,48	419	837,94	1675,9	3351,7	6703,5	13407	
Lá	27,75	55,506	111,01	222,02	444	888,1	1776,2	3552,4	7104,8	14210	
Lá #	29,4	58,806	117,61	235,22	470,4	940,9	1881,8	3763,6	7527,2	15054	
Si	31,15	62,304	124,61	249,22	498,4	996,86	1993,7	3987,5	7974,9	15950	
Dó	33	66	132	264	528	1056	2112	4224	8448	16896	

Então, para a emissão de um som que reproduza determinada nota, em determinada oitava, basta utilizar o valor da frequência dentro do método `tone`, seguido de um `delay` que garanta sua execução por um determinado tempo e finalmente por um `noTone` para encerrar a sua reprodução.

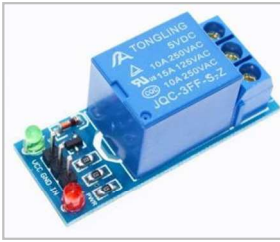
Quanto a ligação eletrônica, é bastante simples. Os buzzers avulsos (sem a placa) devem ter o pino positivo (indicado com um sinal de + ou com uma perna mais longa) ligado à porta digital desejada e o outro ao GND, posicionando em algum dos dois um resistor a partir de 100Ω para reduzir a corrente e garantir que o componente não queime. Já os buzzers encapsulados em uma placa possuem 3 pinos, sendo um deles o GND, o outro o VCC e o outro o pino de sinal que deve ser ligado à porta desejada do Arduino. Estas placas já possuem um resistor interno que garante a redução. Abaixo um exemplo de ligação eletrônica dos dois tipos de buzzer à um Arduino. O primeiro (avulso) ligado à porta digital 4 e o segundo (encapsulado em uma placa) ligado à porta digital 3.



Abaixo, o exemplo de um código com uso do buzzer (ligado à porta digital 3 neste caso) para reproduzir as 8 notas do famoso toque das chamadas telefônicas a cobrar, utilizando o Sol, Lá e Si da quarta oitava e o Dó e Ré da quinta oitava.

```
void setup() {  
  pinMode(3, OUTPUT);  
}  
  
void loop() {  
  tone(3, 395.50); delay(500); noTone(3); //Sol (4ª oitava)  
  tone(3, 444.00); delay(500); noTone(3); //Lá (4ª oitava)  
  tone(3, 528.00); delay(300); noTone(3); //Dó (5ª oitava)  
  tone(3, 444.00); delay(250); noTone(3); //Lá (4ª oitava)  
  tone(3, 498.40); delay(450); noTone(3); //Si (4ª oitava)  
  tone(3, 592.42); delay(300); noTone(3); //Ré (5ª oitava)  
  tone(3, 498.40); delay(250); noTone(3); //Si (4ª oitava)  
  tone(3, 528.00); delay(500); noTone(3); //Dó (5ª oitava)  
  delay(1000);  
}
```

Relé

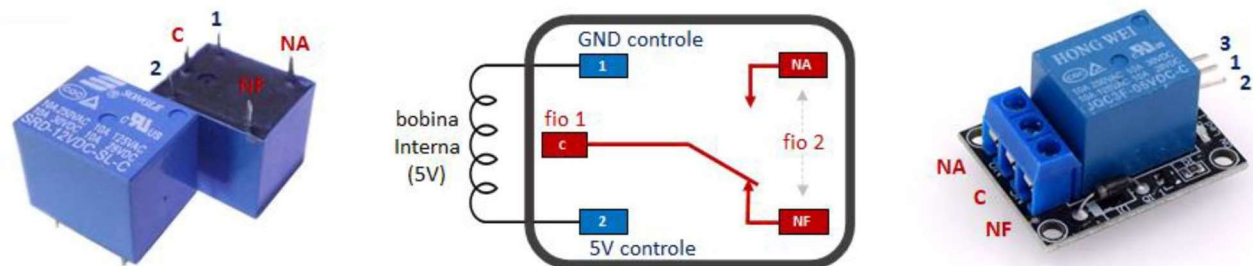


Os relés são um dos componentes mais importantes na automação para ligar e desligar, através de um sinal de 5V, outros componentes e dispositivos que necessitem de tensões maiores do que esta (alimentados por uma fonte de tensão secundária que os sustentem).

Os relés são como interruptores, criando um ponto do circuito que pode ser aberto (interrompendo o fluxo de corrente) ou fechado (permitindo o fluxo de corrente). A diferença é que ao contrário dos interruptores comuns (como a chave de luz, por exemplo) que necessitam de uma força mecânica para alternar entre seus dois estados, os relés são acionados eletronicamente, recebendo uma tensão de 0V ou 5V que movimentam a chave interna para um lado ou para outro.

Aplicações: qualquer sistema que necessite ativar (ligar) ou desativar (desligar) um componente ou dispositivo alimentado por uma fonte secundária em um subcircuito, permitindo assim controlar até mesmo aqueles que necessitem de tensões que o Arduino não pode fornecer (inclusive eletrodomésticos).

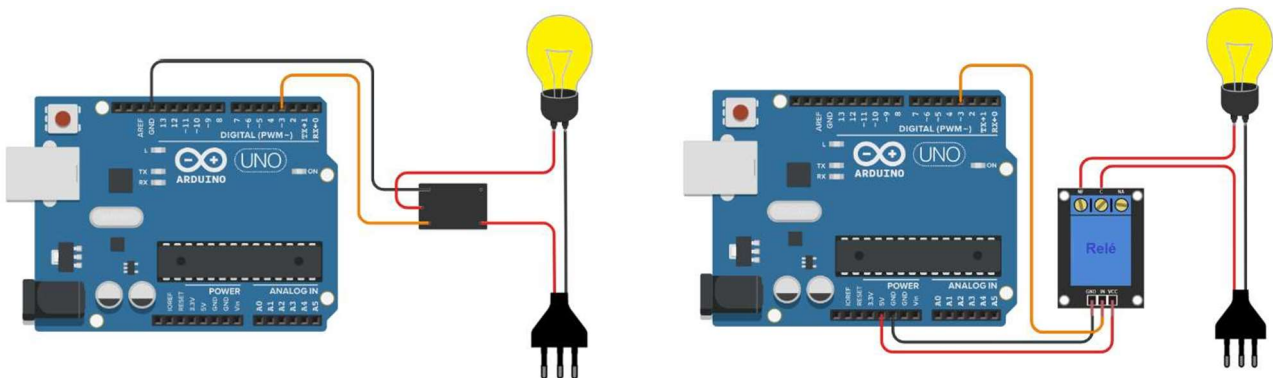
Os relés de estado sólido avulsos, possuem geralmente 5 pinos. Já módulos relé (encapsulados em placas que podem possuir um ou mais relés), possuem pelo menos um pino a mais para garantir potência de acionamento, como será visto abaixo:



Os dois primeiros pinos (1 e 2) são os que vão movimentar a chave interna que funciona como interruptor (através do acionamento de uma bobina) e devem ser ligados a quem vai controlar esta chave do relé (no nosso caso, o Arduino), observando a polarização correta. Já o pino de alimentação extra (3) está presente apenas nos módulos relé e é utilizado para garantir potência no acionamento da chave (5V fixos sendo o pino 2 ligado então à uma porta digital para funcionar como o 5V de controle). Relés avulsos ligados diretamente a placas Arduino muitas vezes não encontram força suficiente para movimentar a chave e necessitam do uso de um transistor em um circuito secundário para

amplificar a corrente, tornando mais complexo seu uso. Já quem vai ser controlado (ligado ou desligado) deve ter um dos fios que o conecta à fonte secundária segmentado (cortado) e um dos lados desse fio deve ser ligado ao pino comum do relé (**C** ou **COM**). Já o outro lado do fio pode ser conectado ou no pino normalmente fechado (**NF** ou **NC**) ou no pino normalmente aberto (**NA** ou **NO**) do relé. A diferença entre eles é que no pino normalmente aberto o contato só é fechado quando uma tensão é aplicada à bobina pelo controle, permanecendo no restante do tempo aberto e sem permitir a passagem de corrente. Já no pino normalmente fechado, a corrente naturalmente flui por ele estar fechando o contato, sendo o fluxo interrompido apenas quando tensão é aplicada à bobina pelo controle, abrindo o contato do circuito. Deste modo, na hora de definir qual deles utilizaremos, devemos dar preferência por aquele que naturalmente garantirá o estado no qual o circuito permanecerá por mais tempo, evitando gasto desnecessário de energia.

Para utilizar relés como controle de ligamento e desligamento de dispositivos externos ao circuito do Arduino, utilizamos um esquema similar ao apresentado abaixo. O único cuidado é que nos relés avulsos de estado sólido do tipo Metaltex (os mais comuns com controle em 5V e possibilitando até 250 V e 10 A no circuito secundário) os pinos são fixos. Já nos módulos relés, cada fabricante define o seu encapsulamento e podemos ter diferença na posição dos 3 pinos.



A lâmpada do esquema apresentado, pode ser substituída pelo dispositivo que desejamos controlar (ligar e desligar através do relé). Sua alimentação, pode ser um plug conectado a uma tomada, mas pode ser também uma fonte de energia, bateria, entre outros. No nosso exemplo, um dos fios da lâmpada está ligado ao pino Comum do relé (**C**) e o outro ao pino Normalmente Fechado (**NF**). Por isso, a lâmpada ficará normalmente ligada e quando o relé for acionado (alimentado com tensão através do pino de controle) ela desligará pois o interruptor se moverá e o circuito antes fechado irá abrir. Se quiséssemos o efeito contrário, com a lâmpada (ou qualquer equipamento que estivermos controlando) normalmente desligada, ligando apenas quando o relé for ativado (alimentado

com tensão através do pino de controle), bastaria ligar um dos fios ao pino Comum (**C**) e o outro ao pino Normalmente Aberto (**NA**). Temos então um circuito de controle de dispositivos externos acionados simplesmente pela mudança de estado da porta digital à qual o pino de controle do relé estiver conectado (em nosso caso, no pino 3).

É importante ressaltar que os módulos podem ser simples, com apenas um relé, ou com múltiplos, independentes entre si. A diferença na pinagem é que para cada relé adicional, temos um pino de de acionamento a mais (geralmente chamado de **IN**), como podemos ver no comparativo abaixo entre um módulo simples e um módulo duplo.

