ARDUINO
PARA ROBÓTICA

John-David Warren
Josh Adams
Harald Molle

Blucher

John-David Warren
Josh Adams
Harald Molle

ARDUINO PARA ROBÓTICA

Tradução
Humberto Ferasoli Filho
José Reinaldo Silva
Silas Franco dos Reis Alves

Arduino para robótica
Edição original em língua inglesa publicada por Apress,

Edição brasileira em língua portuguesa publicada por Editora Blucher,

Editora Edgard Blücher Ltda.

Blucher

Rua Pedroso Alvarenga, 1245, 4º andar
04531-934 – São Paulo – SP – Brasil
Tel.: 55 11 3078-5366
contato@blucher.com.br
www.blucher.com.br

Segundo o Novo Acordo Ortográfico, conforme 5. ed. do *Vocabulário Ortográfico da Língua Portuguesa*, Academia Brasileira de Letras, março de 2009.

Dados Internacionais de Catalogação na Publicação (CIP)
Angélica Ilacqua CRB-8/7057

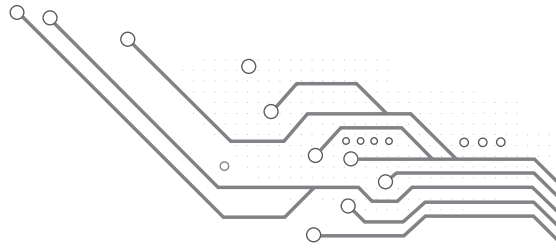
Warren, John-David
Arduino para robótica / John-David Warren, Josh Adams, Harald Molle ; tradução de Humberto Ferasoli Filho, José Reinaldo Silva e Silas Franco dos Reis Alves. – São Paulo : Blucher, 2019.
578 p. : il.

Bibliografia
ISBN 978-85-212-1152-5 (impresso)
ISBN 978-85-212-1153-2 (e-book)
Título original: *Arduino robotics*

1. Arduino (Controlador programável) 2. Robótica 3. Robôs – Desenvolvimento 4. Controle automático – Programas de computador I. Título II. Adams, Josh III. Molle, Harald IV. Ferasoli Filho, Humberto V. Silva, José Reinaldo VI. Alves, Silas Franco dos Reis

15-1528 CDD 629.892

Índice para catálogo sistemático:
1. Arduino (Controlador programável): Robótica



Conteúdo

CAPÍTULO 1

Princípios básicos

O microcontrolador Arduino (Figura 1.1) é como um pequeno centro de comando que fica aguardando suas ordens. Com algumas linhas de código, você pode fazê-lo ligar ou desligar uma lâmpada, ler o valor de um sensor e exibi-lo na tela do computador ou mesmo usá-lo na construção de um circuito caseiro para consertar um eletrodoméstico. Graças à sua versatilidade e ao apoio maciço da comunidade de usuários disponível na internet, o Arduino atraiu uma nova geração de entusiastas de eletrônica que nunca sequer tocaram em um microcontrolador, muito menos programaram um deles.

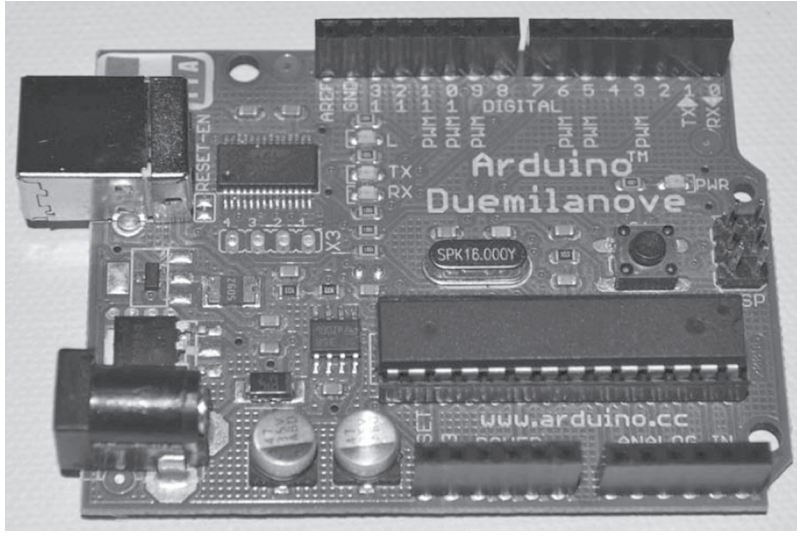


Figura 1.1 Um microcontrolador Arduino Duemilanove.

A ideia básica da comunidade Arduino é criar um ambiente em que *qualquer pessoa* interessada possa participar e contribuir com pouco custo inicial. Uma placa básica do Arduino pode ser encontrada na internet por cerca de 20 dólares, e o *software* necessário para programar o Arduino é baseado em código aberto (que se pode usar e modificar livremente). Você só precisa de um computador e um cabo USB padrão. Além de barato, os criadores do Arduino inventaram uma linguagem de programação (derivada do C++) fácil de aprender, que incorpora várias funções complexas de programação em comandos simples e muito mais fáceis para os iniciantes.

Este livro integra algumas técnicas básicas de construção de robô com a simplicidade do Arduino para criá-los que você poderá modificar e melhorar, compreendendo claramente o processo. A intenção deste livro não é simplesmente "mostrar" como se constrói um robô, mas instruir o construtor iniciante de robôs e inspirar criatividade, para que você possa projetar, construir e modificar seus próprios robôs.

Um obstáculo inevitável que a maioria das pessoas encontra ao construir robôs é o custo. Obviamente, podemos gastar milhares de dólares acrescentando peças de última geração e produtos comerciais caros, mas a maior parte dos entusiastas não tem tempo nem dinheiro para construir um robô desse tipo. Em vista disso, este livro aproveita todas as oportunidades para mostrar como construir uma peça a partir do zero – ou a forma mais barata possível de concretizar esse objetivo. Se qualquer um desses métodos parecer muito complicado, não se preocupe, porque serão indicadas peças substitutas para você comprar.

Entenda que todo projeto apresentado neste livro requer várias tentativas para funcionar – alguns deles podem exigir semanas de "depuração". Posso dizer por experiência própria que, com persistência, acabamos resolvendo o problema mais cedo ou mais tarde – e isso torna a experiência muito mais gratificante.

Descobrir por que um robô não está funcionando requer, com frequência, muito trabalho. Achar erros exige a compreensão de cada etapa do processo, do início ao fim, e a inspeção de cada passo. Quanto mais você tentar, melhor entenderá o processo.

Concluindo, não desanime se algumas das informações deste livro parecerem além de sua capacidade de compreensão. Presumimos que você é iniciante em robótica e programação e nos preocupamos em fornecer um *conhecimento prático* sobre as peças e os códigos utilizados em cada projeto, em vez de sobrecarregá-lo com teoria de eletrônica e instruções complicadas. Antes de iniciar, é melhor assumir a postura positiva de que "você consegue fazer isso" – essa será sua melhor ferramenta.

Para entender melhor o que está ocorrendo dentro de um Arduino, devemos primeiro falar sobre eletricidade e discutir outros conceitos básicos em geral (isto é, componentes eletrônicos e circuitos). Embora os níveis de tensão elétrica encontrados no Arduino (+5 VCC) sejam relativamente inofensivos, se você não souber como a eletricidade funciona, não saberá em qual ponto ela se torna perigosa. Como você verá, os projetos abordados neste livro não utilizam níveis de tensão altos o suficiente para gerar corrente através do corpo, mas ainda assim devemos manusear a eletricidade com cuidado.

ELETRICIDADE

Eletricidade não é nada mais que calor aproveitado. Esse calor pode ser usado para fazer uma variedade de coisas diferentes, como acender uma lâmpada, girar um motor ou simplesmente aquecer uma sala. Quando a eletricidade flui facilmente através de um objeto, ele é chamado de "condutor" (como os fios de cobre). Todo condutor tem uma resistência interna à eletricidade que o impede de transferir 100% da potência. Mesmo o fio de cobre tem uma resistência que reduz o fluxo de eletricidade, gerando calor. Os condutores também têm uma quantidade máxima de potência que eles podem transferir antes de se "sobreaquecerem" (se o condutor for um fio de cobre, isso significa derreter). Com relação à energia elétrica, a potência total também pode ser chamada de calor total.

É por isso que é provável que você veja uma lâmpada ou um micro-ondas com classificação de calor em watts. O watt não é apenas uma medida de calor, mas de potência elétrica.

Alguns dispositivos elétricos (como o Arduino) consomem pouca eletricidade e, portanto, produzem pouco calor. Por isso, nenhuma atenção é dada a essa dissipação de calor. Outros dispositivos são feitos especificamente para transferir grande quantidade de eletricidade (como um controlador de motor) e devem usar dissipadores de calor feitos de metal ou ventoinhas para ajudar a remover o calor do dispositivo. Em ambos os casos, é útil saber determinar a quantidade de calor que um dispositivo elétrico produz para saber como manuseá-la adequadamente.

Analogia elétrica

Geralmente, a eletricidade não pode ser vista (exceto, talvez, em uma tempestade de raios), por isso é difícil entender o que ocorre dentro de um fio quando você liga uma lâmpada ou um eletrodoméstico. Para facilitar a ilustração, considere um sistema elétrico como um tanque de água com um tubo de saída na parte inferior (ver Figura 1.2).

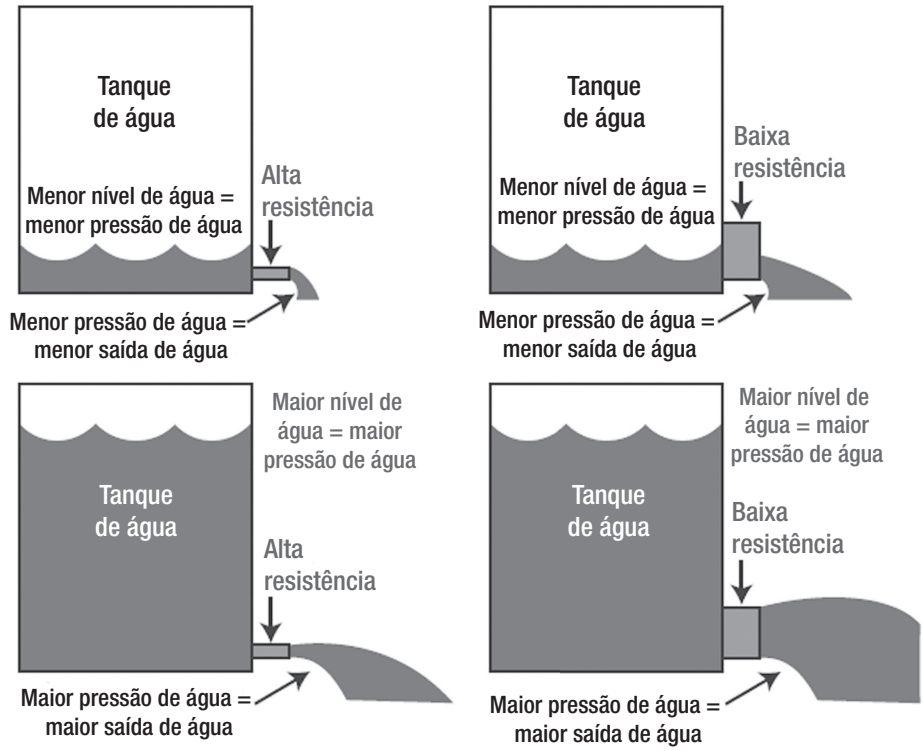


Figura 1.2 Uma analogia para um sistema elétrico.

As quatro imagens mostram como a resistência e a pressão afetam a vazão de água do tanque. Uma resistência maior produz menor vazão de água, ao passo que uma pressão maior produz maior vazão de água. Você pode ver também que, conforme a resistência diminui, um volume bem maior de água sai do tanque, mesmo com uma pressão menor.

Quanto mais água o tanque contiver, mais rapidamente (pressão mais elevada) ela será empurrada pelo tubo de saída. Se não houvesse nenhum tubo de saída, o tanque de água seria simplesmente um reservatório. O fato de haver um tubo de saída no fundo do tanque permite que a água saia, mas apenas a uma taxa determinada pelo diâmetro do tubo. Como o diâmetro do tubo de saída determina a resistência para a água que sai do reservatório, aumentar ou diminuir o diâmetro do tubo de saída aumenta ou diminui inversamente a resistência para a água que sai do tanque (isto é, diâmetro menor = maior resistência = menor vazão de água do tanque).

Tanto o nível (ou pressão) de água quanto a resistência (ou o diâmetro do tubo de saída) podem ser medidos, e utilizando essas medidas você pode calcular a quantidade de água que sai do tanque em um momento específico. A diferença entre a analogia do fluxo de água e o fluxo de eletricidade é que este precisa antes completar seu percurso de volta à fonte para poder ser usado.

Conceitos básicos de eletricidade

Observe que uma pressão de água mais elevada produz uma saída de água mais elevada (desde que a resistência seja a mesma). O mesmo é verdade em relação ao equivalente elétrico da pressão, chamado de "tensão" (V), que representa o potencial de energia que pode ser encontrado em um sistema elétrico. Uma tensão mais alta no sistema terá mais energia para acionar os componentes do sistema. A quantidade de "resistência" (R) encontrada em um sistema impede (reduz) o fluxo de eletricidade, do mesmo modo que a resistência gerada pelo tubo de saída reduz o fluxo de saída de água do tanque. Isso significa que, à medida que a resistência aumenta, a tensão (pressão) também tem de aumentar para manter o mesmo fluxo elétrico na saída. A quantidade de carga elétrica (em coulombs) que passa através de um sistema elétrico a cada segundo é chamada de "corrente" (I) ou "amperagem" e pode ser calculada por meio da tensão, da resistência usando a lei de Ohm. O "watt" (P) é uma medida de energia elétrica que é calculada multiplicando-se a tensão pela corrente. Neste capítulo, falaremos mais profundamente sobre tensão, resistência e corrente. Primeiro, vejamos a relação entre elas: a lei de Ohm.

Segundo a Wikipédia (fonte: `http://en.wikipedia.org/wiki/Ohm's_law`), a lei de Ohm determina que a corrente entre dois pontos de um condutor é diretamente proporcional à diferença de potencial (ou tensão) e inversamente proporcional à resistência entre eles.

Existe uma relação simples entre tensão, resistência e corrente que pode ser calculada matematicamente. Conhecendo qualquer uma das duas das variáveis e a lei de Ohm, você pode calcular a terceira. O watt é uma medida de potência elétrica que está relacionada com a lei de Ohm porque também pode ser calculada com as mesmas variáveis. Veja as fórmulas na Figura 1.3, em que V = potencial, R = resistência, I = corrente e P = watts.

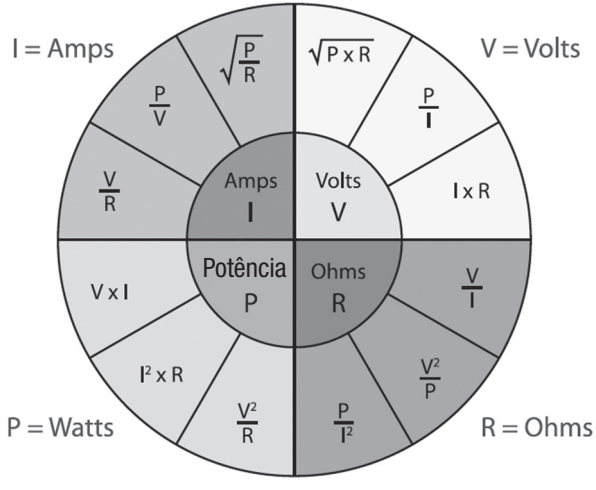


Figura 1.3 Lei de Ohm para calcular potência.

Nota ♦ O gráfico utilizado na Figura 1.3 é cortesia de `www.electronics-tutorials.ws`. Se você estiver interessado em saber mais sobre eletrônica, certamente deve visitar esse site, que tem alguns exemplos e descrições úteis.

As diferentes visões da lei de Ohm incluem o seguinte:

$V = I * R$
$I = V/R$
$R = V/I$

Utilize as seguintes fórmulas para calcular a potência total:

$P = V * I$
$P = I^2 * R$

Você pode encontrar vários outros termos ao trabalhar em um sistema elétrico; falaremos sobre alguns neste livro. Como você deve saber, um sistema elétrico geralmente tem um fio conectado à fonte de energia e um fio comum para completar o circuito. Dependendo do material que você estiver lendo, esses dois elementos podem receber uma denominação diferente. Para ajudar a evitar a confusão que experimentei quando estava aprendendo, a Tabela 1.1 apresenta uma comparação rápida dos vários nomes para os polos positivo e negativo de um sistema elétrico.

Tabela 1.1 Nomes comuns que se referem aos polos positivo e negativo de um sistema elétrico

Tensão de polarização	Terminal polarizado	Fluxo de corrente elétrica	Rótulo esquemático	Nome comum
Positiva	Ânodo	Fonte	VCC	Potência
Negativa	Cátodo	Dreno	VSS	Terra (GND)

Falamos sobre a lei de Ohm e as medições comuns que são utilizadas para descrever as várias propriedades de fluxo de corrente elétrica. A Tabela 1.2 oferece uma lista de unidades elétricas padrão e os respectivos símbolos. Como eles serão usados nos capítulos posteriores deste livro, é bom familiarizar-se com eles.

Tabela 1.2 Termos comuns de medição de eletricidade e respectivos símbolos

Medida	Unidade	Símbolo
Tensão (energia)	Volt	V ou E
Corrente (corrente)	Ampère	I ou A

(*continua*)

Tabela 1.2 Termos comuns de medição de eletricidade e respectivos símbolos (*continuação*)

Medida	Unidade	Símbolo
Resistência	Ohm	R ou Ω
Potência (calor elétrico)	Watt	P ou W
Capacitância	Farad	F
Frequência	Hertz	Hz

Agora falaremos mais sobre as diferentes partes de um sistema elétrico.

Circuitos

O ponto de partida da eletricidade em um sistema elétrico é chamado de "fonte" e, geralmente, refere-se ao terminal positivo da bateria ou à fonte de alimentação. A eletricidade flui através do sistema, da fonte para o dreno, que normalmente é o terminal negativo da bateria ou o fio terra (GND, de *ground*). Para a eletricidade fluir, o circuito deve ser "fechado", o que significa que a corrente elétrica deve voltar ao seu ponto de partida.

O termo "terra" provém da prática de conectar o caminho de retorno de um circuito CA diretamente ao solo usando uma haste de cobre. Você pode notar que a maioria dos medidores elétricos também tem uma haste de aterramento próxima que é presa a um fio que se estende até a caixa de fusíveis. Esse fio terra oferece um caminho para a corrente de retorno sair do sistema. Mesmo que o equivalente CC de GND seja o terminal negativo da bateria, vamos chamá-lo de GND.

Nota ♦ O verdadeiro fluxo de elétrons da corrente elétrica viaja do negativo para o positivo. Porém, a menos que você seja físico, isso não é relevante aqui. Para fins de aprendizagem, adotamos a teoria do fluxo de elétrons convencional, que propõe que a corrente elétrica flui do positivo (+) → negativo (–) em um sistema.

Um sistema elétrico é chamado de "circuito" e pode ser tão simples quanto um cordão de luzes de Natal ligado a uma tomada elétrica ou tão complicado quanto a placa-mãe do seu PC. Agora, pense que em um circuito a eletricidade só flui se algo estiver lá para completar o circuito chamado de "carga" (ver Figura 1.6). Em geral, em um circuito a carga é o dispositivo ao qual você pretende fornecer eletricidade. Pode ser uma lâmpada, um motor elétrico, a serpentina do aquecedor, um alto-falante, a CPU do computador ou qualquer outro dispositivo para o qual o circuito fornecerá energia.

Existem três tipos gerais de circuito: circuito aberto, circuito fechado e curto-circuito. Basicamente, circuito aberto é aquele que está desligado, circuito fechado é aquele que está ligado e curto-circuito é aquele que precisa de reparos (exceto se você tiver usado um fusível). Isso porque um curto-circuito implica que a energia elétrica encontrou um caminho alternativo que não passa pela carga e, portanto, liga o terminal positivo da bateria ao terminal negativo. Isso é invariavelmente ruim e em geral produz faíscas e uma nuvem de fumaça e, ocasionalmente, estalos.

Na Figura 1.4, a lâmpada é a carga do circuito em questão e a chave à esquerda determina se o circuito é aberto ou fechado. A imagem à esquerda mostra um circuito aberto, sem fluxo de eletricidade através da carga, enquanto a imagem à direita mostra um circuito fechado, com fornecimento de energia para a carga.

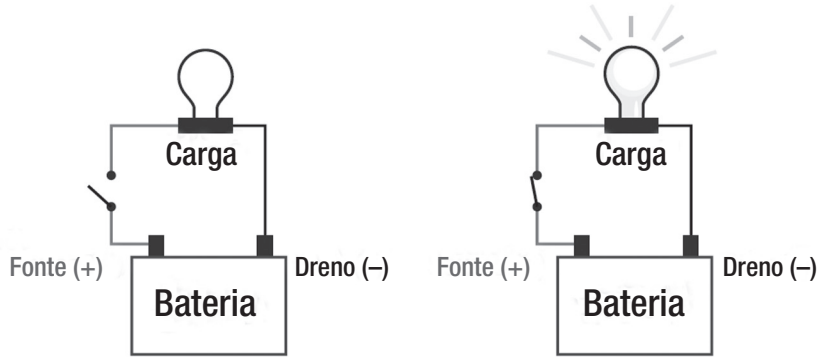


Figura 1.4 Circuitos aberto e fechado.

Medidas elétricas

Sem uma forma de medir os sinais elétricos, estaríamos voando às cegas. Felizmente, existe um dispositivo chamado "multímetro", que é barato e pode facilmente medir tensão, resistência e pequenos níveis de corrente.

Multímetros

Existem diferentes tipos de multímetro com diversas características, mas tudo o que precisamos é de um medidor básico que possa medir os níveis de tensão até cerca de 50 VCC.

Um multímetro convencional pode medir o nível de tensão de um sinal e a resistência de um componente ou do dispositivo que chamamos de carga. Como você pode calcular a corrente com base na tensão e na resistência, isso é tudo de que você precisa para testar um circuito básico. Embora o multímetro digital da Figura 1.5 (à esquerda) custe cerca de 50 dólares, normalmente é possível encontrar um multímetro analógico simples (à direita), que mede tensão e resistência, por menos de 10 dólares. Ambos farão testes básicos e, embora o digital seja mais atraente, na verdade gosto de manter um medidor analógico barato por perto para medir resistência, porque você pode ver a intensidade do sinal com base na rapidez com que a agulha se move para o valor da medida.

Um multímetro padrão tem duas pontas de prova isoladas conectadas à base, que por sua vez são conectadas ao dispositivo que se quer testar. Se você quiser medir a tensão de uma bateria ou circuito, deve colocar a ponta de prova vermelha (conectada aos terminais do multímetro "V, Ω, A") no polo positivo da bateria e a ponta de prova preta (conectada ao terminal COM do multímetro) no polo negativo da bateria ou GND.

Figura 1.5 O multímetro digital Extech MN16a (à esquerda) mede tensões CA e CC, resistência, continuidade, teste de diodo, capacitância, frequência, temperatura e corrente até 10 A. Um multímetro analógico barato comprado em uma loja de eletrônicos do meu bairro (à direita) mede tensão CC e CA, resistência (1 kohm) e corrente até 150 mA (0,15 A). Qualquer um deles consegue diagnosticar um Arduino e também vários outros circuitos, mas não há dúvida de que você precisará de um.

Medição de tensão

A tensão é medida tanto como corrente alternada (CA), que é o tipo encontrado em tomadas elétricas domésticas, quanto como corrente contínua (CC), que é encontrado em baterias. O multímetro deve ser configurado de acordo com o tipo correto de leitura de tensão. Alguns multímetros também têm uma faixa de tensão que é necessário definir antes de fazer a medida. O multímetro analógico da Figura 1.5 (à direita) está definido para 10 VCC, o que efetivamente define a faixa do ponteiro de 0 a 10 VCC.

A tentativa de ler uma tensão muito superior à faixa selecionada pode queimar um fusível. Por isso, você deve sempre usar uma faixa de tensão superior à tensão que será testada. Se não souber o nível de tensão que está testando, selecione a faixa mais alta (300 VCC nesse multímetro) para ter uma ideia. O multímetro digital da Figura 1.5 (à esquerda) tem configurações de tensão CC e CA, mas a faixa é automaticamente detectada e a medida exata de tensão aparece na tela. Procure apenas não exceder os valores nominais máximos de tensão indicados no manual do proprietário.

O nível de tensão de um sinal elétrico também determina se ele é ou não capaz de usar o seu corpo como um condutor. O nível de tensão exata que passa através do corpo humano provavelmente é diferente, dependendo do tamanho da pessoa (dos níveis de umidade, da espessura da pele etc.), mas é possível verificar que, ao tocar acidentalmente em uma tomada de parede com 120 VCA (fio fase) em pé no solo, é produzida uma grande convulsão muscular, mesmo que você esteja usando sapatos com solado de borracha.

Cuidado ♦ Níveis de tensão acima de 40 V podem ser prejudiciais para os seres humanos ou animais de estimação. Lembre-se sempre de desligar a fonte de energia ao trabalhar em seus circuitos e de usar ferramentas isoladas (com cabos de borracha) para testar circuitos. Você não vai querer acabar em uma cama de hospital!

Medição de corrente

A maioria dos multímetros tem um recurso para medir pequenos valores de corrente elétrica (250 mA ou menos), seja CA ou CC. O multímetro digital da Figura 1.5 (à esquerda) pode medir até 10 A de corrente por alguns segundos, enquanto o multímetro mais simples pode medir apenas 150 mA de corrente. Para medir grandes quantidades de corrente (mais de 10 A), você precisa de um sensor de corrente, amperímetro ou grampeamento de voltagem, dependendo da aplicação.

A unidade de medida depende da tensão de funcionamento e da resistência do circuito. À medida que a tensão de operação diminui (descarga das baterias) ou a resistência varia, a corrente elétrica também varia. Em um grande robô em constante movimento, a corrente elétrica muda toda vez que o robô esbarra em uma rocha ou vai por uma superfície levemente inclinada. Isso ocorre porque os motores CC consomem maior corrente quando a dificuldade de locomoção é maior. Uma lanterna de LED consome uma quantidade constante de corrente (cerca de 20 mA a 100 mA por LED) até que as baterias descarreguem.

Você deve ter notado que as baterias são especificadas em ampère-hora (Ah) para refletir a quantidade de corrente elétrica que elas podem fornecer e por quanto tempo. Vagamente, isso significa que uma bateria de 6 V e 12 Ah pode suprir energia para uma lâmpada de 6 V com 1 A de corrente por 12 horas ou essa mesma lâmpada de 6 V com 12 A durante 1 hora. Você pode notar também que as baterias menores (como AA comum) são especificadas em miliampères por hora (mAh). Assim, uma bateria de 2.200 mAh tem uma classificação igual a 2,2 Ah.

Medição de capacitância

Capacitância é a medida de carga elétrica em Farads que pode ser armazenada em um dispositivo. Porém, como 1 Farad é uma quantidade enorme de capacitância, você notará que a maioria dos projetos usa capacitores com valores expressos em microfarads (μF). Um *capacitor* é um dispositivo elétrico que pode reter (armazenar) carga elétrica e fornecê-la a outros componentes do circuito, conforme a necessidade. Embora isso possa lembrar uma bateria, um capacitor pode ser completamente drenado e recarregado várias vezes a cada segundo – a quantidade de capacitância determina a rapidez com que o capacitor pode ser drenado e recarregado.

Alguns multímetros podem medir a quantidade de capacitância entre dois pontos de um circuito (ou o valor de um capacitor), como o Extech MN16a na Figura 1.5. A maioria dos multímetros não mede capacitância, porque geralmente ela não tem grande importância na maioria dos circuitos. Poder testar a capacitância talvez seja útil ao tentar alcançar valores específicos ou testar um capacitor, mas normalmente você não precisará desse recurso em seu multímetro.

Cuidado ♦ Capacitores maiores podem manter uma carga significativa por longos períodos, e tocar os terminais de um capacitor carregado pode causar choque elétrico. Os capacitores encontrados em monitores TRC (tubo de raios catódicos) de computador ou de televisores, os capacitores de motor de arranque e até mesmo os pequenos capacitores encontrados em câmeras descartáveis podem produzir um choque capaz de deixar o braço formigando durante vários minutos e até queimar a pele. É ideal dar um "curto" nos terminais do capacitor com uma chave de fenda isolada antes de tentar manuseá-lo, a fim de descarregar a carga armazenada.

Medição de resistência

A resistência é medida em ohm e nos indica a eficiência com que um condutor transporta eletricidade. O fluxo de corrente e a resistência são inversamente proporcionais. Quando a resistência aumenta, o fluxo de corrente diminui. Assim, um condutor com menor resistência transporta mais eletricidade do que um com resistência maior. Todo condutor tem *alguma* resistência – alguns materiais têm uma resistência tão alta ao fluxo de corrente que são chamados de "isolantes", o que significa que não podem conduzir eletricidade. Quando a eletricidade encontra resistência ao passar por um condutor, transforma-se em calor; por esse motivo, utilizam-se condutores com a menor resistência possível para evitar a geração de calor.

Resistor é um dispositivo elétrico que tem um valor conhecido de resistência medido em ohm e é utilizado para limitar a quantidade de corrente que pode fluir através dele (ver Figura 1.6).

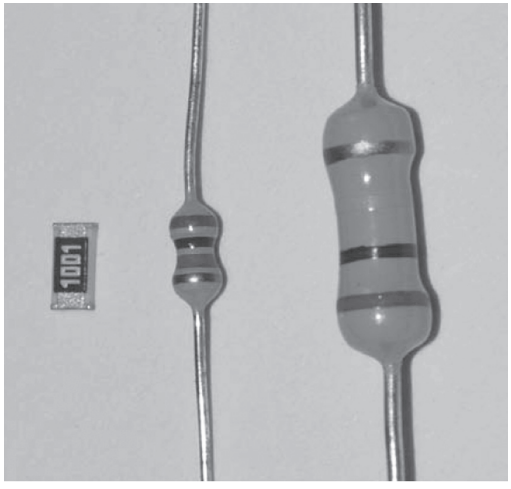


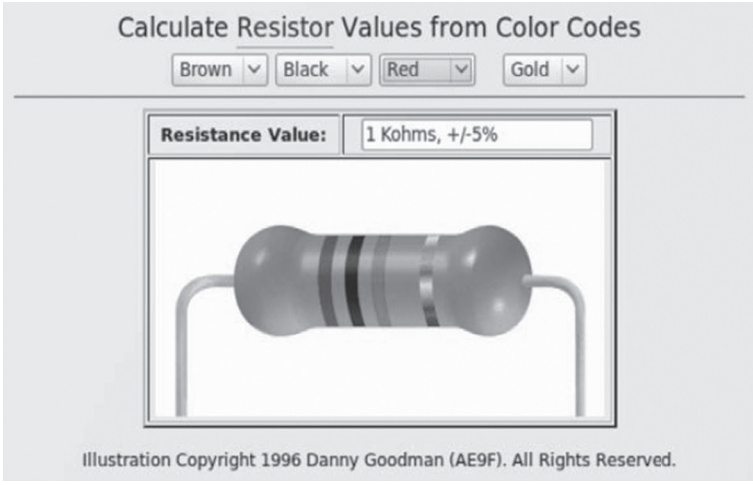
Figura 1.6 Três resistores: resistor de 1/4 W para montagem em superfície (à esquerda), resistor para furo passante de 1/8 W (no centro) e resistor para furo passante de 1/4 W (à direita).

Observe que o resistor de 1/4 W para montagem em superfície (à esquerda) é muito menor que os resistores equivalentes para furo passante, mesmo com a dissipação de potência igual. Normalmente uso resistores para furo passante de 1/8 W porque, embora pequenos, é fácil trabalhar com eles.

É possível utilizar um resistor em série como um componente para limitar a quantidade de corrente elétrica fornecida ao dispositivo e desse modo assegurar que ele se mantenha em um intervalo de funcionamento seguro.

O número sobre o resistor *chip* indica o seu valor em ohm, enquanto as listras codificadas por cor nos resistores para furo passante designam o seu valor. Se você quiser verificar manualmente a resistência de um componente, use o multímetro ajustado para ohm (Ω) – a polaridade não importa, a não ser que você esteja medindo a resistência de um diodo ou transistor.

Uso uma elegante página *web* que permite inserir as cores de cada anel de um resistor para obter o valor da resistência em ohm (ver Figura 1.7). Isso é útil para consultas rápidas quando estamos montando um protótipo ou para identificar o valor de um resistor desconhecido. Visite `www.dannyg.com/examples/res2/resistor.htm`.



Usando este QRcode você será direcionado ao site do Danny Goodman

Imagem usada com permissão de Danny Goodman.

Figura 1.7 A imagem acima mostra a aplicação *web* projetada por Danny Goodman. Tenho essa página marcada em meus favoritos por usá-la muitas vezes para verificar códigos de cor de um resistor desconhecido.

Cálculo da potência de um resistor com a lei de Ohm

Lembre-se de que a qualquer momento uma resistência presente no circuito gerará calor. Por isso, é sempre uma boa ideia calcular a quantidade de calor que será emitida por um resistor (dependendo da carga) para selecionar um resistor com potência adequada. Os resistores são especificados não apenas em ohm, mas também pela potência que podem dissipar (ou eliminar) sem se danificar. As classificações de potência mais comuns são 1/8 W, 1/4 W, 1/2 W etc., e os maiores valores em watts normalmente se referem a resistores maiores, exceto se usarmos componentes para montagem em superfície (ver Figura 1.7).

Para calcular a potência dissipada por um resistor, você precisa saber a tensão do circuito e o valor da resistência em ohm. Em primeiro lugar, precisamos utilizar a lei de Ohm para determinar a corrente que passará pelo resistor. Em seguida, podemos usar a resistência e corrente elétrica para calcular o total de calor que pode ser dissipado pelo resistor em watts.

Por exemplo, se tivermos uma resistência de 1.000 ohm (1 kohm) e uma fonte de alimentação de 12 V, que intensidade de corrente poderá passar através da resistência? E qual deve ser a classificação mínima de potência do resistor?

Primeiro, calculamos a corrente através do resistor por meio da lei de Ohm:

$V = I * R$

$I = V/R$

$I = 12\ V/1.000\ ohm$

$I = 0{,}012\ A$ ou $12\ mA$

Agora usamos a corrente para calcular a potência total (calor): $P = I^2 * R$

$P = (0{,}012\ A * 0{,}012\ A) * 1.000\ ohm$ $P = 0{,}144\ W$

A potência total calculada é de 0,144 W, o que significa que devemos usar um resistor com potência superior a 0,144 W. Como os valores dos resistores comuns geralmente são 1/8 W (0,125 W), 1/4 W (0,25 W), ½ W (0,5 W) etc., podemos usar um resistor com potência de pelo menos 1/4 W (um tamanho mediano) e ainda dissipar com segurança 0,144 W de potência. O uso de um resistor de ½ W não prejudicará nada se você puder ajustar seu tamanho maior no circuito – ele simplesmente transferirá o calor com maior facilidade do que um resistor de 1/4 W com o mesmo valor de resistência.

Agora você provavelmente conseguirá descobrir se suas resistências têm potência adequada. Vamos falar sobre os diferentes tipos de componente de carga.

Osciloscópio

Embora o multímetro seja ótimo para medir tensão, resistência e corrente, às vezes é importante poder ver exatamente o que está ocorrendo em um sinal elétrico. Existe outro dispositivo que é projetado para analisar sinais elétricos, chamado "osciloscópio". O osciloscópio pode detectar padrões ou oscilações repetidas em um sinal elétrico e exibir a forma de onda do sinal em sua tela. Efetivamente, é um microscópio para sinais elétricos. Até recentemente, esses aparelhos eram caros (entre 500 e 5.000 dólares) – alguns osciloscópios de qualidade para hobbistas entraram no mercado por menos de 100 dólares.

O osciloscópio digital DSO Nano (ver Figura 1.8) com código-fonte aberto foi desenvolvido pela SeeedStudio e é também vendido (nos Estados Unidos) pela Sparkfun.com (peça TOL-10244). Tive um osciloscópio desse durante aproximadamente um ano e o usava com frequência porque era fácil trabalhar com ele, pois tinha o tamanho/peso de um celular, e tudo por 89 dólares. Ele tem bateria de lítio recarregável por meio de um cabo mini USB. Tem também um *slot* para cartão de memória para armazenar leituras que podem ser vistas posteriormente em um PC.

Apesar de o osciloscópio ser uma ferramenta valiosa para o diagnóstico de sinais eletrônicos, não é necessário ter um para os projetos deste livro. Você pode se virar com as leituras feitas por um multímetro simples. Há também outras opções de osciloscópio, como um kit DIY da Sparkfun.com por cerca de 60 dólares (peça KIT-09484).

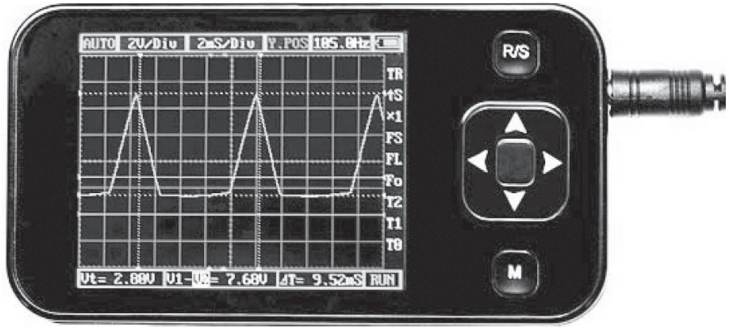


Figura 1.8 O DSO Nano da SeeedStudio (vendido pelo Sparkfun.com) é uma excelente opção para um osciloscópio digital de bolso barato (89 dólares) e cheio de recursos.

Cargas

Em um circuito, a "carga" refere-se a um dispositivo que usa a eletricidade. Existem vários exemplos de "carga", como um motor de corrente contínua, um LED ou uma resistência de aquecimento, e cada uma

dessas cargas cria uma reação diferente no circuito. Por exemplo, uma resistência (em um secador de cabelos ou aquecedor) é simplesmente um fio resistivo enrolado, feito de um metal que se torna vermelho brilhante quando está quente, mas que não se funde, ao passo que o motor elétrico usa eletricidade para energizar um campo eletromagnético em torno de uma bobina, fazendo o eixo do motor mover-se fisicamente. Vamos nos concentrar em dois tipos de carga: indutivas e resistivas.

Cargas indutivas

Se você aplicar energia a um dispositivo e ele criar energia de movimento, é provável que essa carga seja indutiva – o que inclui motores, relés e solenoides. As cargas indutivas criam um campo eletromagnético quando energizadas e, depois que a energia é desligada, geralmente levam algum tempo para perdê-la. Quando a energia é desligada por uma chave, o campo magnético extingue-se e a corrente remanescente retorna aos terminais de alimentação. Esse fenômeno, chamado de força contraeletromotriz, pode danificar os componentes de comutação em um circuito se eles não estiverem protegidos por diodos de retificação.

Cargas resistivas

A carga resistiva utiliza corrente elétrica para produzir luz ou qualquer outra forma de calor, em vez de um movimento mecânico. Nessa categoria incluem-se LEDs, elementos de aquecimento, filamentos de lâmpada, máquinas de solda, ferros de solda e muitos outros. As cargas resistivas usam uma quantidade constante de eletricidade porque a carga não é afetada por influências externas.

Conexões elétricas

Ao construir um circuito elétrico, você deve determinar a tensão de operação desejada antes de selecionar os componentes com os quais vai construí-lo. Apesar de baixos níveis de tensão CA exigirem um transformador, níveis específicos de tensão contínua podem ser conseguidos por métodos diferentes de conexão de baterias. Existem dois diferentes tipos de conexão elétrica: em série e em paralelo.

Conexões em série

Montar um circuito em "série" significa colocar os dispositivos em linha ou um após o outro. Muitas vezes usamos uma ligação em série com baterias para conseguir uma tensão maior. Para mostrar um exemplo desse circuito, usamos duas baterias de 6 V e 10 Ah com o terminal positivo (+) da primeira ligado ao terminal negativo (–) da segunda. Os únicos terminais livres são o terminal negativo (–) da primeira e o terminal positivo (+) da segunda, o que produzirá uma diferença de 12 V.

Quando duas baterias são dispostas em um circuito em série (ver Figura 1.9), a tensão é duplicada, mas a capacidade de ampère-hora permanece a mesma. Assim, as duas baterias de 6 V e 10 Ah trabalham em conjunto para produzir uma única bateria de 12 V e 10 Ah. Essa técnica pode ser útil para alcançar níveis específicos de tensão.

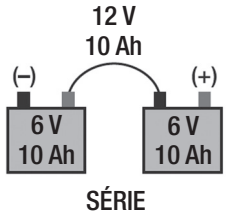


Figura 1.9 Duas baterias idênticas dispostas em um circuito em série produzem o dobro da tensão, mas a mesma capacidade de ampère-hora.

Conexões em paralelo

Montar um circuito em "paralelo" significa colocar todos os terminais comuns juntos. Isso quer dizer que todos os terminais positivos estão ligados entre si e todos os terminais negativos estão ligados entre si. Se colocarmos as duas baterias de 6 V e 10 Ah do exemplo anterior em um circuito paralelo (ver Figura 1.10), a tensão permanecerá a mesma, mas a capacidade de ampère-hora dobrará, resultando em uma única bateria de 6 V e 20 Ah.

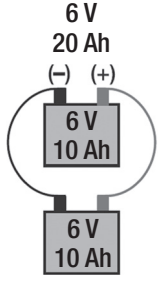


Figura 1.10 Duas baterias dispostas em um circuito paralelo produzem a mesma tensão, mas com o dobro da capacidade de ampère-hora.

Conexão em série e em paralelo

Também é perfeitamente aceitável organizar várias baterias em série e em paralelo ao mesmo tempo para atingir uma tensão e uma razão ampère-hora (ver Figura 1.11). Observe que existem dois conjuntos de baterias de 6 V a 10 Ah dispostas em série para produzir 12 V – e esses conjuntos estão dispostos em paralelo para produzir a mesma tensão, mas com capacidade para 20 Ah.

Para montar conjuntos de baterias, é importante usar baterias com a mesma tensão e capacidade de ampère-hora. Isso significa que você não deve associar uma bateria de 12 V com uma bateria de 6 V para conseguir 18 V.

Em vez disso, use três baterias de 6 V com a mesma capacidade para obter 18 V e evitar carga/descarga desiguais.

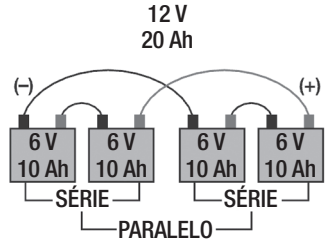


Figura 1.11 Ao montar dois conjuntos de baterias em série e colocá-las em paralelo, você pode criar uma bateria de 12 V com 20 Ah de capacidade usando quatro baterias de 6 V e 10 Ah.

ELETRÔNICA

O campo da eletrônica está relacionado com o controle de fluxo de corrente elétrica através de um circuito por meio, especificamente, de uma chave eletrônica. Antes da invenção da chave eletrônica, os circuitos elétricos eram ligados e desligados por meio de chaves mecânicas, e isso requer um movimento mecânico (isto é, que a chave seja movida manualmente para cima ou para baixo). Apesar de as chaves mecânicas serem perfeitamente aceitáveis e até mesmo preferidas para algumas aplicações, elas são limitadas com relação à rapidez com que podem ser ligadas, em virtude do movimento físico necessário durante o processo de comutação. Mesmo a chave eletromecânica (chamada de relé) não se qualifica como um dispositivo eletrônico, visto que ela usa eletricidade para gerar um movimento mecânico, e com ele realizar a mutação.

A chave eletrônica abstém-se da ação mecânica de comutação por usar uma reação elétrica dentro do dispositivo; portanto, não há componentes móveis. Sem movimento físico, esses dispositivos podem ser comutados muito rapidamente e com uma confiabilidade bem maior. A matéria-prima desses comutadores conduz a eletricidade apenas em determinadas circunstâncias – geralmente deve haver um nível de tensão ou corrente específicos na entrada e na saída do dispositivo para poder abri-lo ou fechá-lo. Quando o dispositivo é ligado, ele conduz eletricidade com um valor especificado de resistência. Quando o dispositivo está desligado, não conduz eletricidade e atua como um isolante. Esse tipo de componente eletrônico é chamado de "semicondutor" porque pode transformar-se em um condutor ou isolante, dependendo das condições elétricas.

Semicondutores

O que determina um circuito "eletrônico" é o uso de semicondutores no lugar de chaves mecânicas, porque eles permitem que os sinais elétricos sejam ligados a velocidades extremamente altas, o que não seria possível com circuitos mecânicos. Existem muitos semicondutores diferentes, e falaremos sobre alguns tipos importantes usados na maioria dos circuitos.

- Diodo: como uma válvula unidirecional para corrente elétrica, esse dispositivo permite que a corrente elétrica passe através dele em apenas uma direção – é extremamente útil por si só, mas também como base para a eletrônica do estado sólido.
- Diodo emissor de luz (*light emitting diode* – LED): esse tipo de diodo emite uma pequena quantidade de luz quando a corrente elétrica passa através dele.
- Resistência dependente da luz (*light dependent resistor* – LDR): esse tipo de semicondutor tem uma resistência que se altera, dependendo da quantidade de luz presente.

- Transistor bipolar de junção (*bipolar junction transistor* – TBJ): trata-se de uma chave eletrônica controlada pela corrente utilizada por suas propriedades de comutação rápida.
- Transistor de efeito de campo de óxido metálico semicondutor (*metal oxide semiconductor field effect transistor* – Mosfet): chave eletrônica controlada por tensão utilizada por suas propriedades de comutação rápida, baixa resistência e capacidade para ser operada em um circuito paralelo. É a base para a maioria dos circuitos amplificadores de potência.

Todos esses dispositivos têm várias camadas de silício positivamente e negativamente carregado, ligado a um *chip* com terminais metálicos condutores expostos para serem soldados no circuito. Como alguns transistores e Mosfets têm diodos embutidos para protegê-los de tensões reversas e força contraeletromotriz, é sempre bom analisar o *datasheet* do componente que você está usando.

Datasheet

Todo dispositivo deve ter seu *datasheet*, que pode ser obtido com o fabricante, geralmente fazendo download no respectivo site. O *datasheet* tem todas as informações elétricas importantes sobre o dispositivo. Os limites superiores, normalmente chamados de "valores máximos absolutos", mostram em que ponto o dispositivo sofrerá danos (ver Figura 1.12). Os limites inferiores (se for o caso) indicam em que nível o dispositivo não responderá às entradas – geralmente, isso não significa que haverá danos no dispositivo, mas simplesmente que ele não funcionará.

FAIRCHILD
SEMICONDUCTOR®

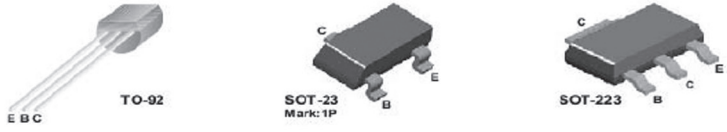
August 2010

PN2222A / MMBT2222A / PZT2222A
NPN General Purpose Amplifier

Features

- This device is for use as a medium power amplifier and switch requiring collector currents up to 500mA.
- Sourced from process 19.

PN2222A **MMBT2222A** **PZT2222A**



Absolute Maximum Ratings * T_a = 25°C unless otherwise noted

Symbol	Parameter	Value	Units
V_{CEO}	Collector-Emitter Voltage	40	V
V_{CBO}	Collector-Base Voltage	75	V
V_{EBO}	Emitter-Base Voltage	6.0	V
I_C	Collector Current	1.0	A
T_{STG}	Operating and Storage Junction Temperature Range	- 55 ~ 150	°C

Figura 1.12 Aqui você pode ver um exemplo de *datasheet* da Fairchild Semiconductor para o transistor 2N2222 NPN. Primeiro são mostrados os encapsulamentos disponíveis e as configurações de pinagem e, em seguida, uma breve lista dos valores máximos absolutos.

Há também uma seção chamada "Características elétricas" para informar em que nível o dispositivo funciona corretamente, em geral mostrando o nível de tensão ou corrente exata que liga ou desliga o dispositivo. Esses valores são úteis para determinar que outros valores de componente (isto é, resistores e capacitores) devem ser selecionados ou se o dispositivo funcionará como pretendido.

O *datasheet* ("folha de dados") costuma fornecer muito mais informações do que você precisa, finalizando com gráficos e dimensões de encapsulamento. Alguns fornecem também recomendações de *layout* de circuitos e sugerem soluções para interfacear o componente com um microcontrolador. Para componentes populares ou comumente usados, você também pode consultar o site do fabricante a fim de obter outros documentos que mostram como eles são usados – estes são chamados de *application notes* ("notas de aplicação"), e podem ser bastante informativos.

Circuitos integrados

Alguns semicondutores têm vários componentes alojados em um mesmo *chip*, caso em que são chamados de circuitos integrados (CIs). Um circuito integrado pode conter milhares de transistores, diodos, resistores e portas lógicas em um minúsculo *chip* (ver Figura 1.13). Esses componentes são oferecidos em encapsulamentos maiores com terminais longos, e versões mais recentes estão sendo feitas em *chips* superpequenos "para montagem em superfície".

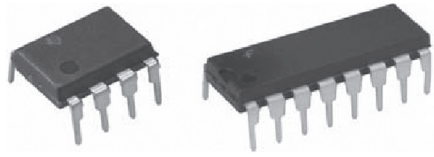


Figura 1.13 Aqui você pode ver um CI Dual Inline Package (DIP) de 8 terminais (à esquerda), e um CI DIP de 16 terminais (à direita). O Arduino Atmega168/328 é um CI DIP de 28 terminais (14 terminais de cada lado).

Encapsulamentos

Usamos semicondutores de diferentes tipos em vários encapsulamentos. O encapsulamento do componente refere-se à forma física, dimensão e configuração dos terminais em que se apresenta. Diferentes encapsulamentos permitem diferentes níveis de dissipação de calor, dependendo do semicondutor. Se você estiver procurando alta potência, os encapsulamentos maiores geralmente dissipam melhor o calor. Para circuitos de baixa potência, normalmente é desejável ter o encapsulamento mais compacto possível. Por isso, tamanhos menores talvez sejam interessantes. Os que mais costumamos usar são o TO-92 e o TO-220 (ver Figura 1.14), que abrigam desde sensores de temperatura até transistores e diodos.

O TO-92 é o menor encapsulamento usado para chaves transistorizadas de baixa potência e sensores. O encapsulamento TO-220 é comumente usado para aplicações de alta potência e é a base para a maioria dos transistores Mosfet de potência, capazes de lidar com cerca de 75 A antes de o metal dos terminais dos componentes danificar-se. O encapsulamento TO-220 tem também uma aba de metal embutida para ajudar a dissipar ainda mais calor do componente e permitir a fixação a um dissipador de calor, se necessário.

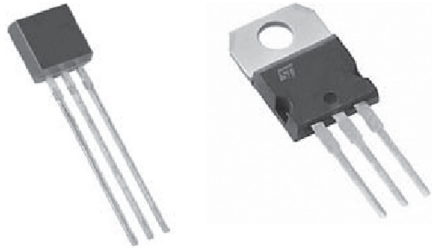


Figura 1.14 O encapsulamento menor (à esquerda), CI TO-92, é usado para reguladores de tensão de baixa potência, transistores de sinal e CIs de sensores. O maior, TO-220 (à direita), é usado para reguladores de tensão mais elevada, chaves de potência Mosfet e diodos de alta potência.

Componentes para furo passante

Ao longo deste livro, procuraremos a forma mais fácil de construir e modificar nossos projetos. Normalmente, isso implica usarmos peças que possam ser substituídas com facilidade, se necessário, e também usarmos peças grandes o suficiente para um iniciante sentir-se tranquilo para soldá-las.

No que diz respeito aos componentes semicondutores, o termo "furo passante" refere-se a qualquer componente cujos terminais transpassam os furos de uma placa de circuito impresso e soldados a uma "ilha" de cobre na parte inferior da placa. Esses componentes geralmente são grandes o suficiente para serem soldados com facilidade a uma placa de circuito impresso (PCI), mesmo para um iniciante. Muitos componentes usam terminais bem maiores que o necessário. Por isso, recomenda-se soldar o componente no devido lugar e no final cortar o excesso na parte inferior de cada um deles para evitar curtos-circuitos no lado de baixo da placa.

Soquetes para CI

"Soquete para CI" é uma base de plástico com contatos metálicos que se destinam a ser soldados à PCI (ver Figura 1.15). O CI é inserido no soquete ao final da soldagem, atenuando o risco de sobreaquecimento do CI durante o processo de soldagem. Isso também é útil para o caso de algo dar errado e provocar um dano no CI. Ele pode ser facilmente substituído sem necessidade de soldagem adicional. Por esse motivo, usamos soquetes para CIs o tempo todo.

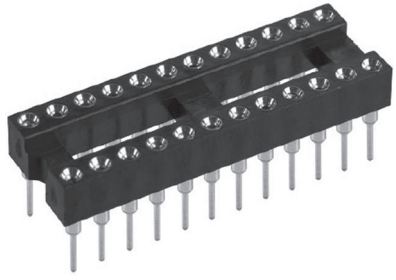


Figura 1.15 Um soquete de CI que deve ser soldado a uma PCI para receber um CI quando o circuito for construído. Como esses soquetes custam menos de 1 dólar cada, podem ser amplamente usados.

Componentes de montagem em superfície

Em vista dos saltos tecnológicos dos fabricantes nos últimos anos, o menor tornou-se o melhor. Isso levou à redução do tamanho dos componentes e CIs, de modo que é possível criar dispositivos menores que fazem a mesma coisa que seus homólogos maiores.

Embora esses dispositivos sejam internamente os mesmos, seus terminais são muito menores, e pode ser um pouco frustrante para um novato tentar soldá-los a uma PCI (ver Figura 1.6, à esquerda, que mostra um resistor de montagem em superfície). A principal diferença entre esses componentes de montagem em superfície e os componentes para furo passante é que os primeiros são soldados na parte superior da placa de circuito impresso sem necessidade de fazer furos. Normalmente, esses componentes ficam colados à PCI e exigem menor espaço de montagem, o que os torna desejáveis para aplicações que demandam economia de espaço.

Alguns componentes de montagem em superfície têm terminais expostos que podem ser soldados normalmente, mas outros têm terminais expostos apenas na parte de baixo do *chip*, o que requer que a solda seja feita em um forno de refluxo para montagem em superfície. Embora essa técnica de soldagem possa ser emulada por um forno elétrico, tentamos escapar da montagem em superfície nos circuitos que construímos neste livro para evitar a dificuldade adicional apresentada por esses componentes.

Nota ♦ No Capítulo 8, como não consegui encontrar uma placa de circuito necessária para completar o projeto deste livro, tive de usar um *chip* de montagem em superfície. Procurei o maior que havia para facilitar a solda, e foi mais fácil do que esperava.

Como já abordamos alguns termos e definições de eletrônica, agora devemos passar para alguns temas específicos do Arduino.

INTRODUÇÃO AO ARDUINO

O Arduino é um microcontrolador AVR programável que oferece um conjunto robusto de recursos, vinte terminais de E/S e um preço acessível – em torno de 30 dólares – para uma placa montada. O Arduino básico conecta-se ao computador por um cabo USB padrão, que oferece uma conexão serial com o PC e a fonte de alimentação de 5 V necessária para operá-lo (dispensa bateria quando se usa o cabo USB).

A equipe do Arduino também desenvolveu um programa para ser executado no computador (disponível para Windows, Mac e Linux) e usado para compilar o código e enviá-lo facilmente à placa Arduino, cujo *chip* adaptador USB (FTDI) permite que o computador a reconheça como um dispositivo serial assim que for conectada. O *software* e os *drivers* mais atuais necessários para a programação podem ser baixados gratuitamente no site `www.arduino.cc`. Confira a seção "Getting started" ("Iniciando") na página inicial do Arduino a fim de ver as instruções passo a passo para a instalação do *software* Arduino em seu sistema operacional:

```
http://arduino.cc/en/Guide/HomePage
```

O *software* do Arduino é considerado um ambiente de desenvolvimento integrado (*integrated development environment* – IDE). Esse é o *software* de programação usado para carregar o código para o microcontrolador Arduino. O IDE contém um editor de texto e um compilador que transforma a

linguagem simplificada de programação do Arduino (que nós escrevemos) em um arquivo binário hexadecimal mais complicado que pode ser enviado diretamente para o microcontrolador.

A linguagem Arduino é uma variante da linguagem de programação C ++, mas usa bibliotecas incorporadas para simplificar as complexas tarefas de codificação e torná-las mais fáceis para os iniciantes. Se você não tem experiência em programação, vai se beneficiar imensamente com as páginas de referência do Arduino. Essas páginas mostram cada um dos comandos do Arduino e como usá-lo com um trecho de exemplo de código. Você pode visitar o site do Arduino para ver essas páginas ou usar a ajuda do Arduino IDE em "Help > Reference" ("Ajuda > Referência"):

```
www.arduino.cc/en/Reference/HomePage
```

Como a linguagem Arduino é um projeto de código aberto (*open source*), ela está sempre sendo melhorada e atualizada. Novas versões do Arduino IDE são lançadas com frequência. Por isso, é melhor atualizar seu sistema com a versão mais recente disponível. Neste livro, a maioria dos projetos usa o IDE 0019-0021, que pode ser baixado da página do Arduino.

Variantes do Arduino

O Arduino é oferecido em formas e tamanhos variados, mas apenas dois modelos usam *chips* completamente diferentes: o padrão e o mega. O padrão é o Arduino básico e refere-se ao *chip* Atmega8/168/328, enquanto o mega é uma placa Arduino distinta, tem mais terminais de E/S e usa um *chip* mais robusto, o Atmega1280. Como o Arduino é um projeto de código aberto, qualquer pessoa pode criar uma nova versão da placa Arduino e distribuí-la como lhe convier. Por esse motivo, vários outros fabricantes criaram "clones" do Arduino que funcionam como o Arduino padrão, mas são feitos por terceiros ou oferecidos em kit de montagem "faça você mesmo".

Há também placas Arduino sem conversor USB embutido. Nesse caso, você deve usar um cabo USB especial de programação (FTDI) para programá-las (ver Figura 1.16, à esquerda). O cabo de programação FTDI custa cerca de 20 dólares na Sparkfun.com (peça DEV- 09718). A vantagem de usar o *chip* FTDI em um cabo de programação separado, em vez de na própria placa Arduino, é que você pode fazer facilmente suas próprias placas do tipo Arduino usando apenas um *chip* Atmega328, um ressonador de 16 MHz e alguns outros componentes fáceis de encontrar. Se você adicionar um conector em barra de pinos, poderá até programar placas Arduino caseiras no próprio circuito (ver Figura 1.16).

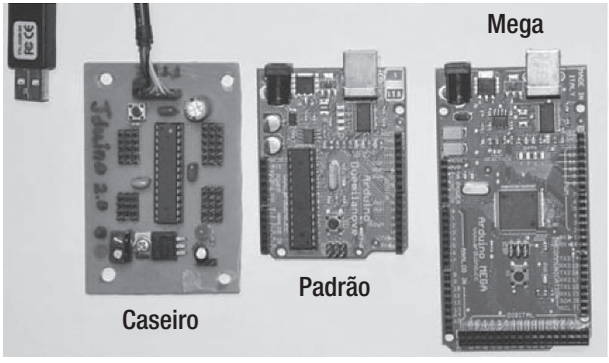


Figura 1.16 Três diferentes tipos de placa.

Depois que comprei o cabo de programação FTDI na Sparkfun.com, envolvi-me em uma inesperada e inspirada temporada de montagens e fiz em torno de 15 clones diferentes do Arduino com diversas configurações de pinos, conectores de parafuso, conectores de R/C, conectores de servomotores e até mesmo algumas placas de extensão que podem ser empilhadas. Apesar de nenhuma das minhas placas caseiras ter a funcionalidade USB *onboard*, várias tinham conector de programação de seis pinos FTDI para permitir programação *in circuit* (no circuito). Por isso, tive de comprar apenas 8 dólares em peças para montar cada placa. Se você gostar de prototipagem, essa é a maneira de trabalhar com bom custo-benefício.

Você pode notar na Figura 1.16 que a placa Arduino caseira tem pouquíssimas peças. Isso ocorre porque há apenas três peças absolutamente necessárias para fazer um Arduino caseiro funcionar: o *chip* Atmega168, um ressonador de 16 MHz e um regulador de tensão de 5 V. Os capacitores, o LED de potência, os pinos do conector e o botão *reset* não são necessários, mas recomendados por motivo de confiabilidade e facilidade de integração em um projeto.

Observe que a versão caseira, à esquerda, usa o mesmo *chip* Atmega168 que o Arduino padrão, mas é programada por meio de um cabo de programação FTDI; a placa no centro é um Arduino Duemilanove padrão; e a última, à direita, é um Arduino mega.

Arduino padrão

O Arduino padrão baseou-se originalmente no *chip* Atmel Atmega8, um microcontrolador de 28 terminais com 20 entradas/saídas (E/S) no total. Dos 20 terminais controláveis, 6 são usados como entradas analógicas, 6 podem ser usados como saídas de PWM, e existem duas interrupções externas disponíveis para uso. O Arduino padrão "roda" em 16 MHz e tem três temporizadores ajustáveis, que poderão alterar as frequências de PWM (discutidas ainda neste capítulo).

Há duas outras variações que são compatíveis em pinagem com esse *chip*, o Atmgea168 e o Atmega328. Um tem maior quantidade de memória *onboard* que o anterior. As versões mais recentes do Arduino padrão vêm com os *chips* mais recentes, o Atmega328, em vez dos *chips* mais antigos – o Atmega8/168. Se você tiver um modelo Arduino mais antigo e quiser atualizar para um *chip* mais recente, com mais memória, poderá comprar um novo *chip* Atmega328 por cerca de 5,50 dólares: basta ligá-lo em seu Arduino (esses *chips* são compatíveis pino a pino e fisicamente iguais). Isso pode vir a ser um problema somente se você tiver um circuito que usa mais memória do que o Atmega8 tem a oferecer, mas esse problema é para usuários mais avançados e projetos maiores.

Uma das principais vantagens desse *chip* é que ele tem encapsulamento de CI para furo passante que pode ser removido da placa Arduino e facilmente montado em uma base de montagens (*breadboard*) ou soldado à placa perfurada de prototipagem para fazer um clone Arduino independente e usá-lo permanentemente em um projeto. O *chip* Atmega328 para furo passante é perfeito para prototipagem, quando usado com um soquete IC 28-DIP.

Nota ♦ Se de alguma forma você destruir um pino em seu Arduino, o problema provavelmente poderá ser corrigido pela substituição do *chip* Atmega168/328 por um novo, que custa 5,50 dólares. Você pode comprá-lo no Sparkfun.com com o *bootloader* do Arduino pré-instalado (peça de código DEV-09217). Já vi isso ocorrer várias vezes, e ainda uso minha primeira placa Arduino!

Arduino mega

O Arduino mega é o *outro* modelo que usa um *chip* Atmega1280 mais robusto, o qual seria como um Arduino padrão ampliado, com um total de 70 terminais de E/S (ver Figura 1.16, à direita). Entre eles, há 16 entradas analógicas, 12 saídas de PWM e 6 interrupções externas disponíveis. O mesmo *software* é usado para todos os modelos Arduino, e todos os comandos na linguagem Arduino funcionam em todos eles.

Esse modelo é oferecido apenas com o Atmega1280 montado em superfície na placa e não pode ser removido, o que limita sua versatilidade em comparação com o Arduino padrão. A princípio, essa placa custava em torno de 75 dólares, mas várias empresas introduziram clones Arduino mega que podem ser encontrados por cerca de 45 dólares. Se você puder adquirir um Arduino extra, é bom ter um à mão se precisar de mais pinos de E/S e não quiser alterar nenhum *hardware*.

Clones

Embora existam apenas dois modelos que usam diferentes *chips* de processamento de base, há um número infinito de clones do Arduino circulando pela internet para montar ou, em muitos casos, comprar. Um clone do Arduino não é uma placa Arduino oficialmente compatível, mas toda placa clone pode ter características próprias e específicas, como configuração de pino, tamanho e função. Tudo o que é necessário para ser compatível com o Arduino é que use o *software* Arduino IDE para carregar o código Arduino.

Existem até mesmo clones que fogem das especificações de *hardware* padrão, mas são compatíveis com o Arduino IDE, como o Arduino Pro Mini, que por padrão opera com 3,3 V e 8 MHz, em vez de 5 V e 16 MHz. Você pode usar qualquer um dos clones do Arduino com o *software* Arduino IDE, mas deve escolher a placa correta no menu Tools (Ferramentas).

Em suma, seja qual for o Arduino que você comprar para começar com este livro, desde que ele leve o nome Arduino, provavelmente funcionará muito bem. Usamos especificamente o Arduino padrão para vários projetos, um Arduino mega para um projeto, um Ardupilot (Arduino com GPS habilitado) para um capítulo e vários clones caseiros do Arduino. Vejamos agora o Arduino IDE para compreender melhor como ele funciona.

Arduino IDE

Supondo que já tenha seguido as instruções para baixar e instalar o Arduino IDE, agora você precisa abrir o programa. A primeira vez que abrir o Arduino IDE no computador, ele perguntará onde você gostaria de colocar seu "caderno de esboços" ou "*sketchbook*" (se estiver usando o Windows ou Linux). Se estiver usando um Mac, seu *sketchbook* será criado automaticamente em usuário/documentos/Arduino. Esse caderno é a pasta em que o IDE armazenará todos os *sketches* que você criar no IDE. Depois que selecionar a pasta Sketchbook, todo o seu conteúdo será exibido no menu File > Sketchbook (Arquivo > Sketchbook).

Ao abrir o IDE, aparecerá uma tela em branco pronta para você digitar código e uma barra de ferramentas em azul na parte superior da tela que fornece botões de atalho para os comandos mais comuns dentro do IDE (ver Figura 1.17). A Tabela 1.3 apresenta uma descrição de cada um.

Figura 1.17 O IDE tem uma barra de ferramentas na parte superior com atalhos para tarefas comuns. Ao usar o IDE, você pode passar o cursor do mouse sobre cada botão para ver a descrição.

Tabela 1.3 Botões da barra de ferramentas do Arduino IDE

	Compilar: Esse botão é usado para verificar a "sintaxe" ou correção do código. Se houver algo rotulado incorretamente ou qualquer variável ainda não definida, você verá um código de erro em letras vermelhas na parte inferior da tela do IDE. Se, no entanto, o código estiver correto, você verá a mensagem "Done Compiling" ("Compilação feita"), assim como o tamanho de seu *sketch* em quilobytes. Esse é o botão que você pressiona para verificar erros em seu código.
	Parar: Caso você esteja executando um programa que está se comunicando com o computador, se pressionar esse botão o programa será interrompido.
	Novo: Esse botão limpa a tela e permite que você comece a trabalhar em uma página em branco.
	Abrir: Esse botão permite que você abra um *sketch* existente a partir do arquivo. Você fará isso quando precisar abrir um arquivo que baixou ou um em que tenha trabalhado anteriormente.
	Salvar: Escolha esse botão para salvar seu trabalho atual.
	Carregar: Esse é um botão mágico que permite que você carregue seu código para o Arduino. O IDE compila o código antes de tentar carregá-lo para a placa, mas sempre pressiono o botão de compilação antes de fazer *upload*. Você pode obter uma mensagem de erro, se tiver com a placa errada selecionada no menu Tools > Board (Ferramentas > Board).
	Monitor Serial: O monitor serial é uma ferramenta para depuração (para descobrir o que está errado). A linguagem Arduino inclui um comando para imprimir valores que são colhidos no Arduino durante a função de *loop* e imprimi-los na tela do computador para que possa vê-los. Esse recurso pode ser extremamente útil se não estiver obtendo o resultado esperado, porque ele pode mostrar exatamente o que está ocorrendo. Usamos esse recurso amplamente para testar o código antes de instalá-lo em um projeto.

O *sketch*

O *sketch*, ou esboço, nada mais é do que um conjunto de instruções para o Arduino realizar. Os *sketches* criados com o Arduino IDE são salvos como arquivos .pde. Para criar um *sketch*, você precisa realizar as três partes principais: declaração de variável, função *setup* e função principal *loop* principal.

Declaração de variável

Declaração de variável é um termo elegante que significa que você precisa digitar o nome de cada entrada ou saída que deseja usar em seu *sketch*. Você pode renomear um pino de entrada/saída do Arduino com qualquer nome (isto é, led_pin, led, my_led, LED2, pot_pin, motor_pin etc.) e pode consultar o pino por esse nome em todo o *sketch*, em vez de pelo número do pino. Além disso, você pode declarar uma variável para um valor simples (não ligado a um pino de E/S) e usar esse nome para se referir ao valor dessa variável. Assim, quando quiser usar o valor da variável no final do *sketch*, lembrará com facilidade. Essas variáveis podem ser declaradas como vários tipos diferentes, mas o mais comum é usarmos um inteiro (int). Na linguagem Arduino, uma variável inteira pode conter um valor que varia de –32.768 a 32.767. Outros tipos de variável serão usados em exemplos posteriores (isto é, *float*, *long*, *unsigned int*, *char*, *byte* etc.) e são explicados, quando usados.

Veja a seguir um exemplo de declaração de variável:

```
int my_led = 13;
```

Em vez de enviar comandos para o pino com esse número do Arduino (ou seja, 13), renomeamos o pino 13 como "my_led". Sempre que quiseremos usar o pino 13, o chamaremos de my_led. Isso é útil quando existem muitas referências a my_led em todo o *sketch*. Se decidirmos alterar o número dos pinos ao qual my_led está atribuído (por exemplo, para o pino 4), você mudará isso uma vez na declaração da variável e, em seguida, todas as referências a my_led conduzirão ao pino 4 – o objetivo disso é facilitar a codificação.

A função setup

Essa função é executada uma vez, sempre que o Arduino for ligado. Esse geralmente é o lugar em que podemos determinar quais das variáveis declaradas são entradas ou saídas por meio do comando pinMode().

```
Example: setup() function:

void setup() {
  pinMode(my_led, OUTPUT);
}
```

Acabamos de usar a função setup() para declarar my_led como saída (OUTPUT precisa estar em CAIXA-ALTA no código). Você pode fazer outras coisas com a função setup(), como ligar a porta serial do Arduino, mas por enquanto isso é tudo.

A função loop

Essa função é o lugar em que o código principal será colocado e executado repetidas vezes e continuamente, até que o Arduino seja desligado. É aí que informamos o que o Arduino deve fazer no processo. Toda vez que o processo atinge o fim dessa função de repetição, volta ao início.

Nesse exemplo, a função *loop* simplesmente pisca o LED de forma intermitente usando a função *delay* (ms) (atraso).

A alteração do primeiro delay(1000) determina por quanto tempo o LED deve permanecer ligado, enquanto a alteração do segundo delay(1000) determina por quanto tempo o LED deve permanecer desligado.

Veja a seguir um exemplo da função loop():

```
void loop() {
                                // início do loop, repita o seguinte:
  digitalWrite(my_led, HIGH);   // mude LED para ligado
  delay(1000);                  // espere por 1 s
  digitalWrite(my_led, LOW);    // desligue o LED
  delay(1000);                  // espere por 1 s
                                // termine o loop e volte para o início
}
```

Se você associar essas seções de código, terá um *sketch* completo. Seu Arduino deve ter um LED embutido para o pino digital 13, portanto esse *sketch* renomeia esse pino como my_led. O LED ficará ligado durante 1.000 ms (1 s) e, em seguida, ficará desligado durante 1.000 ms, indeterminadamente, até que você o desligue fisicamente. É recomendável mudar o tempo de atraso na Listagem 1.1 e fazer o *upload* para ver o que ocorre.

Listagem 1.1 Exemplo Piscando o LED

```
//Código 1.1 - Piscando
// Pisque o LED no pino 13

int my_led = 13;                // declaração da variável my_led

void setup() {
  pinMode(my_led, OUTPUT);      // use o comando pinMode() para atribuir my_led como um pino de↩
                                saída (OUTPUT)
}

void loop() {
  digitalWrite(my_led, HIGH);   // faça my_led igual a HIGH (ligue my_led)
  delay(1000);                  // espere por 1 s (1000mS)
  digitalWrite(my_led, LOW);    // faça my_led igual a LOW (desligue my_led)
  delay(1000);                  // espere por 1 s

}                               // retorne para o início do loop

// end code
```

Você pode copiar esse exemplo de código para a tela do Arduino IDE e pressionar o botão de compilação (ver Figura 1.18). Com o Arduino conectado à porta USB, você poderá pressionar o botão Carregar (*Upload*) para enviar o código para o Arduino. Se você digitar o código manualmente, não

precisará adicionar os comentários, porque eles não serão compilados em código. Esse código não requer nenhuma entrada depois que é carregado, mas você pode alterar o tempo de atraso e recarregá-lo para ver a diferença.

Nota ♦ Você notará que em muitos *sketches* sempre há comentários, que são indicados por meio de duas barras (//), e, em seguida, algum texto. Qualquer texto adicionado após as duas barras invertidas não será convertido em código e serve apenas de referência: *//Isto é um comentário; não será processado como código.*

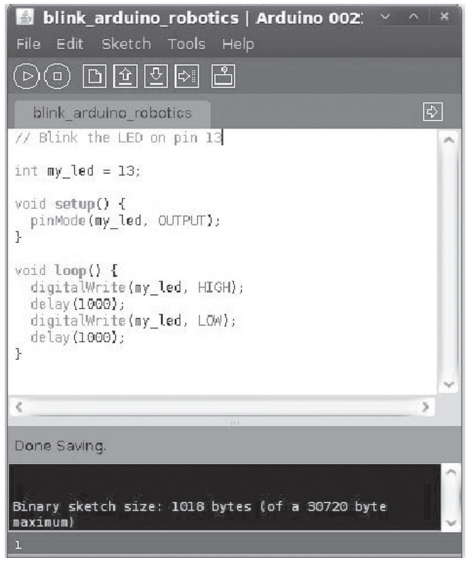


Figura 1.18 Tela do programa Arduino IDE com o exemplo de *sketch* de LED intermitente na Listagem 1.1.

Sinais

Vários são os tipos de sinal que o Arduino consegue ler e escrever, mas eles podem ser diferenciados em dois principais grupos: digital e analógico. Um sinal digital é +5 V ou 0 V, mas um sinal analógico pode ser qualquer tensão linear entre 0 V e +5 V. É possível também ler e gravar sinais de pulso digitais e comandos seriais usando o Arduino e diversas funções incluídas.

Sinais digitais

O Arduino Uno/Diecimila/Duemilanove tem catorze pinos de entrada/saída digitais chamados D0-D13. No Arduino, todo pino digital pode ser configurado como uma ENTRADA ou SAÍDA por meio do comando pinMode() na função setup(). Um sinal digital pode ter somente dois estados no Arduino: ALTO (*HIGH*) ou BAIXO (*LOW*). Isso ocorre independentemente de o sinal digital ser uma entrada ou

uma saída. Quando um pino está em 5 V, é considerado alto; quando está em 0 V ou GND, é considerado baixo.

Entradas digitais

As entradas digitais são úteis se você quiser determinar quando um botão foi pressionado (isto é, um sensor de colisão) e se uma chave está ligada ou desligada ou se quiser ler um pulso em um sensor para determinar seu valor oculto. Para determinar se uma entrada é alta ou baixa, use o comando digitalRead(pin). Um sinal de entrada digital nem sempre pode contar com a disponibilidade total de 5 V. Desse modo, o limite para conduzir um pino de entrada para alto é 3 V, e qualquer valor abaixo desse limiar é considerado baixo.

Os receptores R/C usados têm uma saída de sinais para servo que, em aviões/barcos/carros de modelismo, são pulsos de eletricidade levados para nível alto por um período curto, mas de duração específica, antes de voltarem para um nível baixo. A duração do pulso especifica a posição das alavancas de controle do transmissor R/C. Se você tentar verificar esse tipo de sinal com seu medidor de tensão, não verá o movimento da agulha. Isso porque o pulso é muito curto para ser registrado no medidor, mas qualquer entrada digital no Arduino pode ler um comprimento de pulso como um sinal de servo por meio do comando pulseIn().

Podemos ler informações de uma entrada digital não só com base no fato de ser alta ou baixa, mas com base *no tempo* em que permanece alta ou baixa. O Arduino é eficiente precisamente para medir o comprimento de pulsos elétricos curtos, até cerca de 10 µs! Isso significa que muitas informações podem ser codificadas para uma entrada digital em forma de pulso ou de comando serial.

Saídas digitais

A saída digital é igualmente simples, mas pode ser usada para fazer tarefas complexas. Se você tem um Arduino, já viu o *sketch* "Hello, World" ("Olá, Mundo!"), que simplesmente pisca o LED no pino D13 embutido na placa – esse é o uso mais simples de uma saída digital. Todo pino do Arduino é capaz de fornecer ou de drenar uma corrente de 40 mA em 5 V.

Muitas vezes, a corrente fornecida por um pino do Arduino não é suficiente para ligar nada além de um LED. Desse modo, um conversor de nível ou amplificador pode ser usado para aumentar a tensão e corrente que é ligada e desligada pelo Arduino para um nível compatível para o controle de motores, luzes ou relés. Além disso, os pinos digitais são a base para a transferência serial de dados, que pode enviar vários comandos através de uma única saída digital (Listagem 1.2).

Listagem 1.2 Criação de uma entrada e uma saída digital no mesmo *sketch*

```
// Exemplo de Código: Entrada e Saída
// Esse código vai atribuir uma entrada digital para o pino 2 do Arduino e uma saída digital para o pino 13 do Arduino.
// Se a entrada é HIGH, a saída para o LED será LOW

int switch_pin = 2;      // renomeia o pino 2 do Arduino para "switch_pin"
int switch_value;        // precisamos de uma variável para guardar o valor de switch_pin, que será "switch_value"
int my_led = 13;         // renomeia o pino 13 do Arduino para "my_led"

void setup(){
  pinMode(switch_pin, INPUT);              // diz ao Arduino que switch_pin(pin 2) é um Input
  pinMode(my_led, OUTPUT);                 // diz ao Arduino que my_led(pin 13) é um Output
}
```

```
void loop(){
  switch_value = digitalRead(switch_pin);    // lê switch_pin e guarda o valor em switch_value

  if (switch_value == HIGH){                 // se o valor "for igual (==)" a  HIGH...
    digitalWrite(my_led, LOW);               // ... então desligue o LED
  }
  else {                                     // caso contrário...
    digitalWrite(my_led, HIGH);              // ...ligue o LED.
  }

}
// end code
```

Esse exemplo de código usa uma simples instrução *if* para testar o valor de switch_pin. Você pode ligar um fio (*jumper*) ao pino 2 do Arduino (switch_pin) – conecte a outra extremidade do fio ao GND ou a +5 V, para ver os valores de mudança de LED. Se o valor de entrada for alto, o Arduino definirá o pino my_led em baixo (desligado). Se o valor de entrada for baixo, o Arduino definirá o pino my_led em alto (ligado). Para saber mais sobre as declarações *if/else* com exemplos, consulte as páginas de referência do Arduino em `http://arduino.cc/en/Reference/Else`.

Caso especial: interrupções externas

Ao usar o comando digitalRead() para um pino de entrada no Arduino, você recebe apenas o valor que está disponível no momento exato em que o comando é chamado. No entanto, o Arduino tem capacidade para determinar quando o estado de um pino muda, sem usar o comando digitalRead(). Isso é chamado de *interrupção*. Uma interrupção é um método de entrada que avisa quando o estado de um pino muda, sem você precisar verificar. O Arduino padrão tem duas interrupções externas, nos pinos digitais 2 e 3, ao passo que o Arduino mega tem seis interrupções externas, nos pinos digitais 2, 3, 21, 20, 19 e 18.

A interrupção deve ser iniciada uma vez na instalação e deve usar uma função especial chamada de rotina de interrupção de serviço (*interrup service routine* – ISR), que é executada toda vez que a interrupção é disparada (ver Listagem 1.3). As interrupções podem ser configuradas para serem acionadas quando um pino muda de baixo para alto (subida), de alto para baixo (descida) ou simplesmente toda vez que o pino mudar de estado para uma ou outra direção.

Para melhor ilustrar esse processo, imagine que você está cortando a grama do quintal antes do almoço. Você sabe que o almoço ficará pronto em breve e não quer se atrasar, mas também não quer desligar o cortador de grama a cada cinco minutos para checar. Em vez disso, você pede a quem está cozinhando que vá lá fora e o avise quando o almoço estiver pronto. Dessa forma, você pode continuar cortando a grama sem se preocupar com o almoço.

Você é *interrompido* quando o almoço está pronto (o pino muda de estado) e, depois que você acaba de *comer* (a rotina de interrupção de serviço), pode voltar *a cortar a grama* (o *loop* principal).

Isso é útil porque verificar regularmente o estado de um pino que não muda de estado com frequência pode retardar as outras funções no *loop* principal. A interrupção simplesmente interromperá o *loop* principal durante o tempo necessário para percorrer o ISR e, em seguida, retornar imediatamente ao lugar exato em que parou no *loop*. É possível usar um pino de interrupção para monitorar um sensor de colisão em um robô que precisa parar os motores assim que for pressionado ou usar um pino de interrupção para capturar pulsos em um receptor R/C sem interromper o restante do programa.

A Listagem 1.3 requer o uso de um sistema de rádio R/C. O receptor R/C pode ser alimentado usando os pinos +5 V e GND do Arduino, ao passo que o sinal de R/C deve ser conectado ao pino 2 do Arduino. Se você ainda não tiver um receptor R/C, poderá testar esse exemplo posteriormente.

Listagem 1.3 Usando um pino de interrupção para capturar o comprimento de um pulso R/C

```
// Exemplo de Código – Usando um pino de Interrupção para capturar o comprimento de um pulso R/C
// Conecte o sinal de um receptor R/C no pino 2 do Arduino
// Ligue o trasmissor R/C se estiver usando as duas interrupções no Arduino
// Se receber um sinal válido, o LED no pino 13 está ligado.
// Se nenhum sinal válido for recebido, o LED está desligado.

int my_led = 13;

volatile long servo_startPulse;
volatile unsigned int pulse_val;
int servo_val;

void setup() {
  Serial.begin(9600);
  pinMode(servo_val, INPUT);

  attachInterrupt(0, rc_begin, RISING);    // inicie a interrupção na subida do sinal
}

// declarando a interrupção por rampa de subida
void rc_begin() {
  servo_startPulse = micros();
  detachInterrupt(0);     // desligue a interrupção por rampa de subida
  attachInterrupt(0, rc_end, FALLING); // ligue a interrupção por rampa de descida
}

// declarando a interrupção por rampa de descida
void rc_end() {
  pulse_val = micros() - servo_startPulse;
  detachInterrupt(0);     // desligue a interrupção por rampa de descida
  attachInterrupt(0, rc_begin, RISING); // ligue a interrupção por rampa de subida
      }

void loop() {
  servo_val = pulse_val;  // grave o valor da Interrupt Service Routine calculada

if (servo_val > 600 && servo_val < 2400){
    digitalWrite(my_led, HIGH);    // se o valor está no intervalo do R/C, ligue o LED

  Serial.println(servo_val);
}
else {
     digitalWrite(my_led, LOW);    // Se o valor não está no intervalo do R/C, desligue o LED.
}
    }
```

Esse código Arduino procura qualquer sinal válido de pulso de servo R/C de um receptor R/C ligado ao pino digital 2 do Arduino, que é o lugar em que a respectiva "interrupção externa 0" está localizada. Se for detectado um pulso válido (deve estar entre o comprimento de 600 µS e 2.400 µS), o LED no pino digital 13 ligará. Se não for detectado nenhum pulso, o LED permanecerá desligado.

Como a Listagem 1.3 usa uma interrupção, ela apenas captura os pulsos de R/C quando eles estão disponíveis, em vez de verificar o pulso a cada ciclo de *loop* (*polling*). Como alguns projetos exigem a execução de muitas tarefas diferentes a cada ciclo de *loop* (leitura de sensores, comando de motores,

envio de dados em série etc.), o uso de interrupções pode economizar um valioso tempo de processamento, interrompendo o *loop* principal somente quando algo muda no pino de interrupção.

O único problema que encontrei ao usar as duas interrupções externas do Arduino é que elas estão disponíveis apenas nos pinos digitais 2 e 3, o que conflita com o uso do pino digital 3 como saída de PWM.

Sinais analógicos

Estabelecemos que um sinal digital de E/S deve ser baixo (0 V) ou alto (5 V). As tensões analógicas podem estar em qualquer ponto intermediário (2 V, 3,4 V, 4,6 V etc.) e o Arduino tem seis entradas especiais que podem ler o valor dessas tensões. Essas seis entradas analógicas de 10 bits (com conversores digital/analógico) podem determinar o valor exato de uma tensão analógica.

Entradas analógicas

A entrada está à procura de um nível de tensão entre 0-5 V e transformará essa tensão para um valor de 10 *bits*, ou de 0-1.023. Isso significa que, se você aplicar 0 V na entrada, verá um valor analógico de 0; se aplicar 5 V, verá um valor analógico de 1.023; e nada intermediário será proporcional à entrada.

Para ler um pino analógico, você deve usar o comando analogRead() com o pino analógico (0-5) que gostaria de ler. Uma observação interessante sobre entradas analógicas no Arduino é que elas não precisam ser declaradas como variáveis ou como entradas na configuração. Por meio do comando analogRead(), o Arduino reconhece automaticamente que você está tentando ler um dos pinos A0-A5, em vez de um pino digital.

Um potenciômetro (resistência variável) funciona como um divisor de tensão e pode ser útil como uma saída de uma tensão analógica de baixa corrente que pode ser lida pelo Arduino por meio de uma entrada analógica (ver Figura 1.19). A Listagem 1.4 apresenta um exemplo de leitura de valor do potenciômetro.

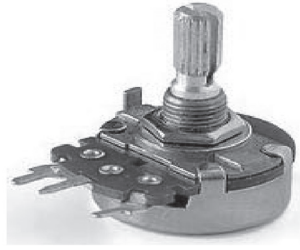


Figura 1.19 Esse potenciômetro de rotação tem três terminais. Os dois terminais externos devem ser conectados ao GND e +5 V, respectivamente (a orientação não importa), enquanto o terminal central deve ser conectado a um pino de entrada analógica no Arduino.

Listagem 1.4 Como ler uma entrada analógica

```
// Exemplo de Código - Entrada Analógica
// Leia um potenciômetro no pino 0 analógico
// E mostre o valor com 10 bits (0-1023) no monitor serial
// Depois de fazer o "upload", abra o monitor serial do Arduino IDE em 9600bps.

int pot_val;     // use a variável "pot_val" para guarder o valor lido do potenciômetro
```

```
void setup(){
  Serial.begin(9600);  // inicie a comunicação serial do Arduino em 9600 bps
}

void loop(){
  pot_value = analogRead(0);  // use analogRead para ler o pino 0 analógico
  Serial.println(pot_val);  // use o comando Serial.print() para enviar o valor para ↵
                            o monitor
}

// end code
```

Copie o código anterior para o IDE e faça o *upload* para o seu Arduino. Esse *sketch* habilita a porta serial nos pinos 0 e 1 do Arduino usando o comando Serial.begin() – você conseguirá abrir o monitor serial no IDE e ver os valores analógicos convertidos do potenciômetro já ajustados.

Saídas analógicas (PWM)

Essa saída não é tecnicamente uma saída analógica, mas é o equivalente digital de uma tensão analógica disponível em um pino de saída. Esse recurso é chamado de modulação por largura de pulso e é uma forma eficiente de fornecer um nível de tensão em algum ponto entre a fonte e GND.

Em eletrônica, o termo PWM é empregado com muita frequência porque é um recurso importante e útil em um microcontrolador. Esse acrônimo significa *pulse width modulation* (modulação por largura de pulso) e é o equivalente digital de uma tensão analógica obtida com um potenciômetro. No Arduino, há seis dessas saídas nos pinos digitais 3, 5, 6, 9, 10 e 11. O Arduino consegue mudar facilmente o ciclo de trabalho ou saída a qualquer momento no *sketch*, por meio do comando analogWrite().

Para usar o comando analogWrite (PWM_pin, velocidade), você deve escrever em um pino PWM (pinos 3, 5, 6, 9, 10, 11). Como o ciclo de trabalho de PWM varia de 0 a 255, você não deve escrever nenhum valor superior ou inferior para o pino. Costumo acrescentar um filtro para garantir que nenhum valor de velocidade acima de 255 ou abaixo de 0 seja escrito em um pino PWM porque isso pode provocar um comportamento instável e indesejado (ver Listagem 1.5).

Listagem 1.5 Como comandar uma saída de PWM

```
// Exemplo de Código – Entrada Analógica – Saída PWM
// Leia o potenciômetro do pin 0 analógico
// A saída PWM no pino 3 será proporcional à entrada do potenciômetro (cheque com um multímetro).

int pot_val;     // use a variável "pot_val" para guardar o valor do potenciômetro
int pwm_pin = 3; // renomeie o pino Arduino PWM 3 = "pwm_pin"

void setup(){
    pinMode(pwm_pin, OUTPUT);
}

void loop(){

    pot_value = analogRead(0);          // leia o valor do potenciômetro no pino 0 analógico ↵
pwm_value = pot_value / 4; // pot_value max = 1023 / 4 = 255

    if (pwm_value > 255){               // filtre para garantir que pwm_value não ultrapasse 255
pwm_value = 255;

    }
```

```
    if (pwm_value < 0){                // filtre para garantir que pwm_value não seja menor que 0
pwm_value = 0;
    }
    analogWrite(pwm_pin, pwm_value);    // escreva pwm_value em pwm_pin
}
// end code
```

Esse código lê o potenciômetro como na Listagem 1.4, mas nesse caso também comanda um sinal de saída proporcional PWM para o pino digital 3 do Arduino. Você pode verificar a saída do pino 3 com um medidor de tensão – ele deve ler de 0 V a 5 V, dependendo da posição do potenciômetro.

Se você tiver um resistor de 330 ohm e um LED disponível, poderá conectar o resistor em série com um dos terminais do LED (apenas verifique se a polaridade do LED está correta) no pino 3 e GND do Arduino, para ver o brilho do LED diminuir e aumentar de 0% a 100% por meio de um sinal digital de PWM. Não podemos usar o LED no pino 13 para esse exemplo porque ele não tem o recurso de PWM.

Ciclo de trabalho

Em um sinal de PWM de 1 kHz, existem 1.000 ciclos ligado/desligado a cada segundo, cada ciclo com 1 ms de duração. Durante cada um desses ciclos de 1 ms, o sinal pode ser alto em parte do tempo e baixo no restante do tempo. Um ciclo de trabalho de 0% indica que o sinal fica baixo durante todo o tempo de 1 ms, enquanto um ciclo de trabalho de 100% fica alto durante todo o tempo de 1 ms. Um ciclo de trabalho de 70% fica alto durante 700 µs e baixo nos 300 µs restantes, para cada um dos 1.000 ciclos por segundo – desse modo, o efeito global do sinal é 70% do total disponível.

O ciclo de trabalho de uma saída de PWM no Arduino é determinado por meio do comando analogWrite (pino, ciclo de trabalho). O ciclo de trabalho pode variar de 0 a 255 e ser alterado a qualquer momento durante o programa – é importante impedir que o valor do ciclo de trabalho ultrapasse 255 ou fique abaixo de 0, porque isso provoca efeitos indesejados no pino PWM.

A maioria dos controladores de velocidade de motor altera o ciclo de trabalho (mantendo a frequência constante) do sinal de PWM que controla as comutações de potência do motor, a fim de alterar a respectiva velocidade. Esse é o método preferido para controlar a velocidade de um motor, porque relativamente nenhum calor é desperdiçado no processo de comutação.

Frequência

A frequência é dada em hertz (Hz) e reflete o número de ciclos (de comutação) por segundo. O ciclo de comutação ocorre num curto período quando a linha de saída passa de completamente alta para completamente baixa. Os sinais do PWM normalmente têm uma frequência definida e alteram o ciclo de trabalho, mas você pode mudar as frequências do PWM do Arduino de 30 Hz até 62 kHz (o que equivale a 62.000 Hz) adicionando uma única linha de código para cada conjunto de pinos PWM.

Em 30 Hz, a linha de saída será alterada de alta para baixa apenas trinta vezes por segundo, o que terá efeitos visíveis sobre uma carga resistiva como um LED, fazendo-o pulsar de ligado para desligado. A frequência de 30 Hz funciona muito bem para uma carga indutiva como a de um motor de corrente contínua, que leva um tempo maior do que é permitido para desenergizar-se entre os ciclos de comutação, gerando uma operação aparentemente suave.

Quanto maior a frequência, menos visíveis são os efeitos de comutação sobre a operação da carga, mas quando a frequência é muito alta os dispositivos de comutação começam a gerar excesso de calor.

Isso ocorre porque, quanto maior a frequência, menor a duração do ciclo de comutação (ver Tabela 1.4), e se o ciclo de comutação for demasiadamente curto, a saída não terá tempo suficiente para mudar completamente de alto para baixo antes de voltar para alto. O comutador na verdade mantém-se em algum ponto entre ligado e desligado, em um estado de condução cruzada (também chamada "*shoot--through*") que gerará calor.

É simples determinar o comprimento total de cada ciclo de trabalho, bastando para isso dividir o tempo pela frequência. Como a frequência determina o número de ciclos de trabalho durante o intervalo de 1 segundo, basta dividir 1 s (ou 1.000 ms) pela frequência do PWM para determinar a duração de cada ciclo de comutação.

Para uma consulta rápida, veja algumas conversões comuns de tempo/velocidade:

- 1.000 ms = 1 s
- 1.000 µs = 1 ms
- 1.000.000 µs = 1 s
- 1.000 Hz = 1 kHz

A Tabela 1.4 mostra todas as frequências disponíveis para os pinos PWM do Arduino e em que pinos cada frequência está disponível.

Tabela 1.4 Lista de frequência *versus* tempo de ciclo de PWM

Frequência de PWM em Hertz	Tempo por ciclo de comutação	Pinos PWM do Arduino
30 Hz	32 ms	9 e 10, 11 e 3
61 Hz	16 ms	5 e 6
122 Hz	8 ms	9 e 10, 11 e 3
244 Hz	4 ms	5 e 6, 11 e 3
488 Hz	2 ms	9 e 10, 11 e 3
976 Hz (1 kHz)	1 ms (1.000 µs)	5 e 6, 11 e 3
3.906 Hz (4 kHz)	256 µs	9 e 10, 11 e 3
7.812 Hz (8 kHz)	128 µs	5 e 6
31.250 Hz (32 kHz)	32 µs	9 e 10, 11 e 3
62.500 Hz (62 kHz)	16 µs	5 e 6

Para mais informações sobre como alterar os temporizadores do sistema para operar em diferentes frequências de PWM, visite o site do *playground* do Arduino:

`www.arduino.cc/playground/Main/TimerPWMCheatsheet`

Exemplo de PWM doméstico

Para simular a frequência e o ciclo de trabalho com cronometragem manual (para fins de aprendizagem e experimentação), associe a Listagem 1.1 (Piscando o LED) e a 1.4 (potenciômetro) para poder alterar a frequência e o ciclo de trabalho de uma saída de pseudo-PWM no pino 13 (o LED embutido). Você só precisa de um potenciômetro ligado ao pino 0, analógico, do Arduino.

Usando cronometragem manual e o LED embutido no pino 13 do Arduino, podemos simular um sinal de PWM em frequências diferentes e com diferentes ciclos de trabalho de 0% a 100%, como mostrado na Listagem 1.6.

Listagem 1.6 Exemplo de pseudo-PWM

```
// Exemplo de Código - Exemplo de Pseudo-PWM (código de um PWM Pulse Width Modulation doméstico)
// Pisque o LED no pino 13 com um ciclo de trabalho variável
// O ciclo de trabalho é determinado pelo valor lido do potenciômetro no pino 0 analógico
// Mude a frequência do PWM reduzindo o valor da variável "cycle_val" para os seguintes valores:
// 10 milliseconds = 100 Hz frequency (fast switching)
// 16 milliseconds = 60 Hz (normal lighting frequency)
// 33 milliseconds = 30 Hz (medium switching)
// 100 milliseconds = 10 Hz (slow switching)
// 1000 milliseconds = 1 Hz (extremely slow switching) - impraticável, mas tente assim mesmo.

int my_led = 13;    // declare a variável my_led
int pot_val;        // use a variável "pot_val" para guardar o valor do potenciômetro
int adj_val;        // use essa variável para ajustar pot_val para um valor variável de frequência
int cycle_val = 33; // Use esse valor para ajustar manualmente a frequência do sinal do pseudo-PWM

void setup() {
  pinMode(my_led, OUTPUT);     // use o comando pinMode() para atribuir my_led como OUTPUT
}

void loop() {
  pot_val = analogRead(0);   // leia o valor do potenciômetro de A0 (retorna um valor entre 0 - 1023)
  adj_val = map(pot_val, 0, 1023, 0, cycle_val); // mapeie o intervalo de entrada ↵
                                                 0 - 1023 em 0 - cycle_val

  digitalWrite(my_led, HIGH);       // faça my_led HIGH (ligado)
  delay(adj_val);                   // continue ligado por este intervalo de tempo
  Write(my_led, LOW);               // faça my_led LOW (desligado)
  delay(cycle_val - adj_val);       // permaneça desligado por este intervalo de tempo

}

// end code
```

A Listagem 1.6 mostra como ajustar o ciclo de trabalho de um LED que pisca a 60 Hz (16 ciclos de comutação por segundo). Esse exemplo de *sketch* funciona apenas para fins didáticos. Como o valor de cycle_val também determina quantos passos estão no intervalo de desvanecimento do LED, você perderá a resolução de ciclo de trabalho à medida que aumentar a frequência. Escolhi 60 Hz para demonstrar uma frequência que é quase igual à das lâmpadas domésticas. Nessa velocidade de comutação, o olho humano não consegue detectar a pulsação, e o LED parece estar firmemente emitindo luz proporcional ao ciclo de trabalho.

Se você quiser aumentar manualmente a frequência do sinal de pseudo-PWM no *sketch* anterior, pode alterar a variável *cycle_val* para algo um pouco maior (frequência menor). Para alterar a frequência

de 60 Hz para 30 Hz, é preciso modificar o tempo de ciclo, mudando a variável *cycle_val* de 16 ms para 33 ms. Além disso, você ainda pode usar o potenciômetro para alcançar os mesmos ciclos de trabalho, mas os resultados serão visivelmente menos regulares. À medida que a frequência de PWM fica abaixo de 60 Hz, você percebe uma pulsação no LED em qualquer ciclo de trabalho (exceto em 100%).

Como já analisamos várias funções básicas do Arduino, falaremos agora sobre os princípios básicos de montagem de circuito.

MONTAGEM DE CIRCUITOS

Uma coisa é conseguir programar o Arduino e testar um circuito elétrico, mas o que pode ocorrer se não for possível encontrar o circuito exato de que você precisa? Talvez seja mais fácil você mesmo montar o circuito. Primeiro, você precisa saber ler um projeto elétrico, que é chamado de esquema. Um esquema elétrico mostra um símbolo universal para cada componente eletrônico (bem como um nome e valor) e uma representação de como ele se conecta com os outros componentes do circuito.

Projeto do circuito

O projeto do circuito pode ser feito em um bloco de notas (no computador) ou em uma folha de papel, mas a reprodução de circuitos feitos à mão pode ser demorada e tediosa. Se você se preocupa em despender pouco tempo ao seu projeto, poderá usar um programa de código aberto ou *freeware* para criar um diagrama esquemático e um projeto de placa de circuito impresso (PCI) para seu circuito. Atualmente prefiro elaborar todo o projeto do meu circuito no computador – mesmo que não esteja pensando em gravar uma PCI com base no projeto, gosto de fazer pelo menos um diagrama esquemático para o circuito.

Existem vários bons programas de computador que podem ser usados para projeto de circuitos. Para os iniciantes, recomendo o programa de código aberto Fritzing, que usa uma biblioteca de componentes muito bem ilustrada para oferecer ao usuário uma sensação visual de como o circuito ficará, bem como um esquema adequado para cada projeto. Existe até uma placa Arduino disponível na biblioteca de componentes para você usar em seus diagramas esquemáticos – usei esse programa para gerar vários diagramas menores e exemplos ilustrativos.

Baixe o Fritzing em: `http://fritzing.org/`

Para usuários mais experientes, o Eagle Cad é um excelente programa de *design* de circuitos que pode ser usado em versões *freeware* ou pagas e tem amplas bibliotecas de componentes e ferramentas de *design* profissionais. Esse programa também é usado em vários capítulos para abrir e imprimir arquivos de projeto de PCI em seu computador.

Baixe o Eagle Cad em: `www.cadsoft.de/`

O Eagle Cad permite criar PCIs confiáveis, compactas e com aparência profissional que são ajustadas para atender exatamente às suas necessidades. Você gastará um pouco mais de tempo na preparação do circuito, mas conseguirá reproduzir facilmente quantas cópias quiser – uma tarefa tediosa se fosse utilizar o método mais simples de fiação ponto a ponto. Não se intimide com todos os botões disponíveis no programa. Ao deslizar o mouse sobre um botão, há uma descrição de sua função. Pense no Eagle como um verdadeiro programa de pintura para aficionados de computador.

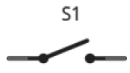
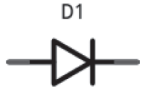
Esse programa é um editor de placa de circuito impresso (PCI) e oferece uma versão gratuita para uso amador (com restrições ao tamanho da placa). Ele permite que você abra, edite e imprima tanto esquemas quanto arquivos de PCI com até duas camadas e uma área de serigrafia de 8,1 cm x 10,1 cm. Não se deixe enganar pela restrição de tamanho: é mais que suficiente para montar qualquer um dos circuitos usados neste livro e em muitos outros. Contudo, se você quiser montar sua placa-mãe para PC ou algo similar, precisará comprar a licença profissional para um tamanho ilimitado de placa de PC.

Falaremos mais sobre o uso de *software* de *design* para criação de circuitos no Capítulo 6. Por enquanto nos concentramos em alguns tipos diferentes de componente e em sua função. Embora existam muitos componentes disponíveis, apenas algumas peças são utilizadas nos projetos apresentados ao longo deste livro. Vejamos algumas imagens, símbolos elétricos e descrições referentes a cada uma.

Diagrama esquemático

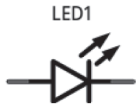
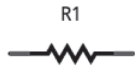
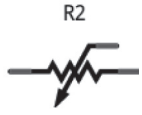
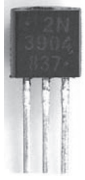
Um diagrama esquemático é uma representação gráfica de um circuito que usa um símbolo padrão para cada componente elétrico com um número que representa seu valor. Isso pode ser útil para assegurar a polaridade adequada e a orientação de cada componente quando ele é colocado no circuito para soldar. Além disso, esse diagrama esquemático pode manter-se o mesmo, ainda que os valores ou o encapsulamento dos dispositivos usados no circuito mudem. Consulte a Tabela 1.5 para examinar alguns componentes e símbolos elétricos comuns encontrados em um diagrama esquemático.

Tabela 1.5 Símbolos de componentes comuns que você pode encontrar ao ler um diagrama esquemático

Componente	Símbolo	Descrição do símbolo
		VCC: símbolo comum para uma bateria como fonte de alimentação. Bateria é um tipo de fonte de alimentação portátil que usa células para armazenar carga elétrica. As células podem ser dispostas em diferentes ordens para produzir níveis específicos de tensão. Aqui é mostrada uma bateria de 9 V geralmente usada em controle remoto ou detector de fumaça.
		Chave: uma chave simples para abrir ou fechar um circuito. Esse tipo de chave, chamado de "chave de contato momentâneo", fecha o circuito quando o botão é pressionado. Ao soltar o botão, o circuito é novamente aberto. Esses botões são usados em circuitos eletrônicos para aplicações de baixa potência.
		Diodo: o símbolo de um diodo, o lado de trás do triângulo (do lado esquerdo), é chamado de "ânodo" ou extremidade positiva, e o terminal com a lista (lado direito) é chamado de "cátodo" ou extremidade negativa. O diodo funciona como uma válvula unilateral – existem muitos tipos diferentes de diodo, e eles também são usados para construir portas lógicas, transistores e praticamente todos os outros tipos de semicondutor. Acostume-se com esse símbolo porque você o verá muitas vezes.

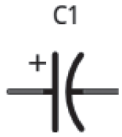
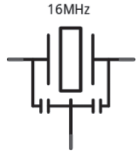
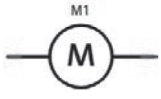
(*continua*)

Tabela 1.5 Símbolos de componentes comuns que você pode encontrar ao ler um diagrama esquemático (*continuação*)

Componente	Símbolo	Descrição do símbolo
		Diodos emissores de luz (LED): são comumente usados em circuitos eletrônicos como luzes indicadoras porque são baratos, consomem pouca corrente, duram muito tempo e são bastante luminescentes para sua dimensão. Tenho um saco com vários LEDs coloridos na minha bancada porque é inevitável que use pelo menos um em cada projeto.
		Resistor: é um componente de fio com uma resistência específica, usada para resistir ao fluxo de corrente através de um circuito ou para um dispositivo do circuito. Os resistores não são polarizados, o que significa que a corrente pode fluir em qualquer direção e não importa a orientação em que eles são instalados.
		Resistor variável (potenciômetro): é o que normalmente você imagina ser um botão de volume. Esse resistor usa um mecanismo deslizante para mover um contator ao longo de um plano de resistência variável linear. Normalmente, existem três terminais em um potenciômetro e os dois externos são ligados a +5 V e GND (em qualquer ordem), enquanto o terminal central é a tensão de saída analógica variável do potenciômetro – o terminal central deve ser ligado à entrada do Arduino ou de outros circuitos de controle.
		Transistor NPN como chave: o transistor é o tipo mais simples de chave digital. Existem muitas variações de transistor, mas usamos apenas os tipos TBJ e Mosfet. Na foto é mostrado um transistor bipolar de junção NPN 2N3904, comumente usado em circuitos eletrônicos com a função de chave para níveis de corrente de até 200 mA. O TBJ em geral é usado como chave de baixa potência e alta frequência que é facilmente controlada por meio de um Arduino. Um transistor tipo N é considerado uma chave do lado de baixo que muitas vezes é usado com o transistor tipo P.
		Transistor PNP como chave: é semelhante ao tipo NPN, mas pode ser usado apenas como uma chave para VCC. Na foto é mostrado um transistor PNP 2N3906, que também consegue comutar cargas de até 200 mA e foi concebido para complementar o tipo NPN 2N3904.

(*continua*)

Tabela 1.5 Símbolos de componentes comuns que você pode encontrar ao ler um diagrama esquemático (*continuação*)

Componente	Símbolo	Descrição do símbolo
		Por meio da combinação de PNP e NPN, você pode criar um circuito de amplificação ou isolação de sinal (ver Capítulo 3 para mais informações).
		Capacitor: é um dispositivo capaz de reter determinada quantidade de carga elétrica que é usada para fornecer corrente para o restante do circuito ou para absorver variações bruscas de tensão para nivelar os sinais. A especificação desse capacitor eletrolítico é de 100 µF e 25 V. Você deve sempre escolher um capacitor com uma especificação de tensão de no mínimo 10 V superior à tensão de funcionamento do sistema. Exceder o limite de tensão pode explodir o capacitor! Alguns capacitores são polarizados e têm um terminal terra específico (indicado por uma listra ou um terminal mais curto), ao passo que os outros não são polarizados e podem ser colocados em ambas as direções.
		Ressonador cerâmico: ocupa o lugar de um cristal e de dois capacitores porque tem dois capacitores embutidos. Basta conectar o terminal central ao GND e os terminais externos aos terminais Xtal1 e Xtal2 do *chip* Atmega168 (em qualquer ordem). Esse dispositivo oferece uma base para todas as funções de temporização no Arduino – tente imaginá-lo como um metrônomo digital.
		Motor: esse símbolo geralmente indica um motor de corrente contínua de dois fios. Esse motor de corrente contínua com caixa de engrenagens acoplada é um pequeno motor de modelismo que pode ser usado em um projeto de robótica. Geralmente são usados dois fios para operar esse tipo de motor, caso em que a inversão da polaridade dos fios inverterá a direção do eixo de saída do motor.
		Regulador de tensão: o regulador de tensão linear LM7805 é útil para converter qualquer entrada de tensão CC de 6 V-25 V em uma fonte de saída regulada de +5 V. Como ele é capaz de fornecer apenas 1 A de corrente, não é recomendável usá-lo para alimentar motores CC em um robô. Porém, ele funciona muito bem em uma placa de prototipagem ou para alimentar o Atmega168 em um circuito Arduino caseiro.
		Terra (GND): esse símbolo universal representa o sinal GND em um circuito. Todo circuito tem um sinal GND, porque é o caminho de retorno que completa o circuito – todos os sinais GND em um circuito devem ser ligados entre si e retornar para o terminal negativo da fonte de alimentação.

Na Figura 1.20, usamos alguns dos símbolos da Tabela 1.5 para um diagrama esquemático de circuito simples, com uma bateria (VCC1), uma chave (S1), um resistor limitador de corrente (R1) e uma luz LED (LED1). No esquema você pode ver os símbolos para cada componente conectado com linhas pretas, indicando uma conexão elétrica. Para ver que aparência um diagrama esquemático tem quando conectado, veja a Figura 1.21.

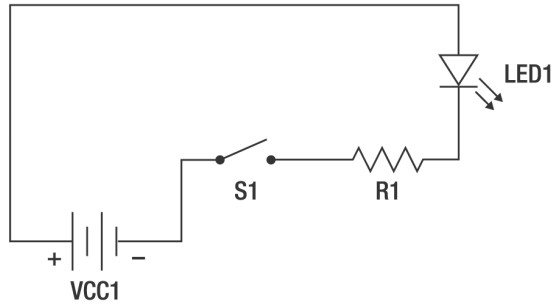


Figura 1.20 Esse diagrama esquemático mostra os símbolos de circuito para quatro componentes diferentes em um circuito simples.

O objetivo do diagrama esquemático anterior é mostrar as conexões elétricas dos componentes de *hardware* mostrados na Figura 1.21. Se tudo estiver conectado como mostrado no diagrama esquemático, o circuito funcionará como pretendido. Isso permite que os usuários montem circuitos sem levar em conta sua dimensão física ou aparência.

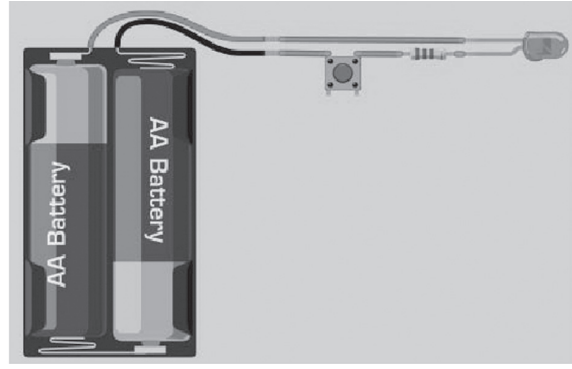


Figura 1.21 Essa imagem é uma ilustração do circuito do diagrama esquemático mostrado na Figura 1.22. Você pode ver a bateria (VCC1), a chave (S1), o resistor limitador de corrente (R1) e a luz vermelha (LED1). Ao pressionar o botão, o circuito é fechado e o LED acende. Ao soltá-lo, o LED apaga.

Prototipagem

Prototipagem é a arte de criar um projeto ou conceito de uma forma crua, concebido não para ser perfeito, mas para testar a viabilidade de uma ideia. Mesmo que você se sinta suficientemente tranquilo com seus cálculos matemáticos para determinar o peso e a velocidade aproximados de seu robô, na

verdade você só saberá como ele funciona depois que o construir e experimentar. É aí que o protótipo vem a calhar.

Você pode construir um protótipo temporário com os materiais com os quais se sentir à vontade (madeira, PVC, metal etc.) Desde que ele seja robusto o suficiente para a montagem temporária de motores e baterias, você conseguirá ter uma boa ideia da velocidade e movimentação do robô real e ajustar correspondentemente o conjunto de engrenagens, a capacidade da bateria ou a tensão do sistema.

A prototipagem está relacionada não apenas com instalação de motores e engrenagens, mas também com projeto, montagem e teste de circuitos eletrônicos. Falamos também sobre algumas ferramentas úteis que tornam os testes e a montagem de circuitos muito mais fáceis.

Base para montagem (*breadboard*)

A base para montagem é uma placa de plástico para experimentos que pode ser comprada na maioria das lojas de produtos eletrônicos por menos de 20 dólares (ver Figura 1.22). É uma ferramenta valiosa para o experimentador de eletrônica porque é possível adicionar ou remover componentes para teste na grade de plástico sem precisar soldá-los. Bases para montagem não podem conduzir grande quantidade de corrente, portanto não devem ser usadas em projetos de alta potência. Contudo, antes de criar um modelo permanente, é recomendável usar uma base para montagem para testar qualquer circuito que você vier a construir.

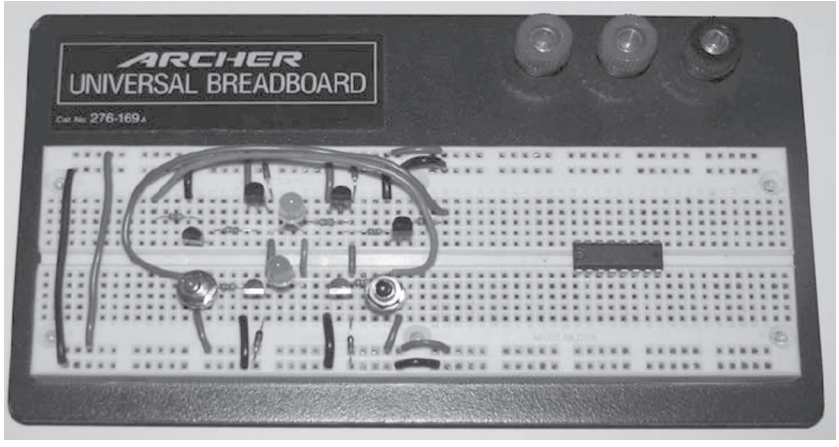


Figura 1.22 Essa é uma base para montagem comum encontrada no Sparkfun.com (peça PRT- 00112) ou em qualquer loja de componentes eletrônicos.

Placa perfurada para prototipagem (placa perfurada)

Assim que seu circuito estiver funcionando em uma base para montagem, você poderá fazer uma cópia réplica para usar como protótipo. Isso pode ser feito facilmente com uma placa perfurada e um ferro de solda. A placa perfurada é uma placa de circuito impresso (PCI) pré-perfurada com espaçamento de ~2,5 mm entre os furos para integrar facilmente a maioria dos componentes através deles

(ver Figura 1.23). Todo furo na placa tem uma ilha de cobre própria para que você possa soldar, e toda ilha é separada da seguinte (exceto em projetos especiais). Esse método requer conexões com fio ponto a ponto, que podem ser entediantes se o circuito for grande, caso em que a melhor solução talvez seja você mesmo gravar sua PCI (ver Capítulo 6). Entretanto, a placa perfurada pode ser uma excelente plataforma para o construtor iniciante de circuito testar uma variedade de protótipos sem a necessidade de projetar e gravar uma PCI adequada.

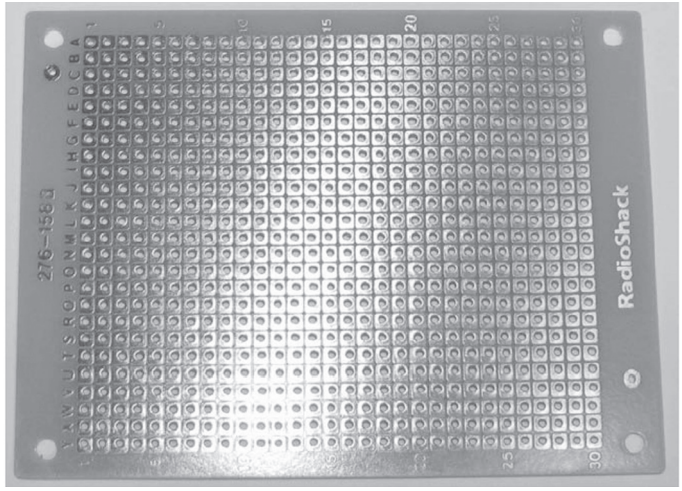


Figura 1.23 Uma placa de prototipagem perfurada convencional com uma ilha de cobre individual para cada furo passante. Você pode construir circuitos completos nesse tipo de placa usando componentes, fio de cobre e ferro de solda elétricos. Como normalmente essa placa custa menos de 5 dólares, ela é útil para protótipos.

Placas de circuito impresso

Depois de verificar se um circuito está funcionando como pretendido em seu protótipo em placa perfurada, pode ser que você queira fazer dez cópias da placa dele para vender ou usar em outros projetos. A fiação manual das dez placas não é apenas tedioso, mas o fio usado em projetos de soldagem ponto a ponto pode quebrar ou se romper, comprometendo a confiabilidade.

Para evitar esse tedioso processo de fiação manual de toda placa de circuito que você vier a montar, uma alternativa é fazer o que é chamado de "placa de circuito impresso" ou, abreviadamente, PCI. A PCI pode ser feita à mão ou em um computador, e requer que você crie um desenho de trilhas de circuito em uma placa de fibra de vidro revestida de cobre (*copper clad*) e corroa o cobre em torno delas (ver Figura 1.24). Todos os fios de uma PCI são as trilhas de cobre criadas no desenho do circuito.

Com um circuito de cobre gravado na placa, você pode soldar os componentes diretamente no cobre – é isso que é chamado de placa de circuito. O Arduino é impresso em uma placa revestida de cobre em ambos os lados e coberta com epóxi azul para proteger as trilhas de cobre contra curto-circuito. Usando materiais fáceis de encontrar, você pode fazer suas placas de circuito impresso em casa em poucas horas (ver Capítulo 6).

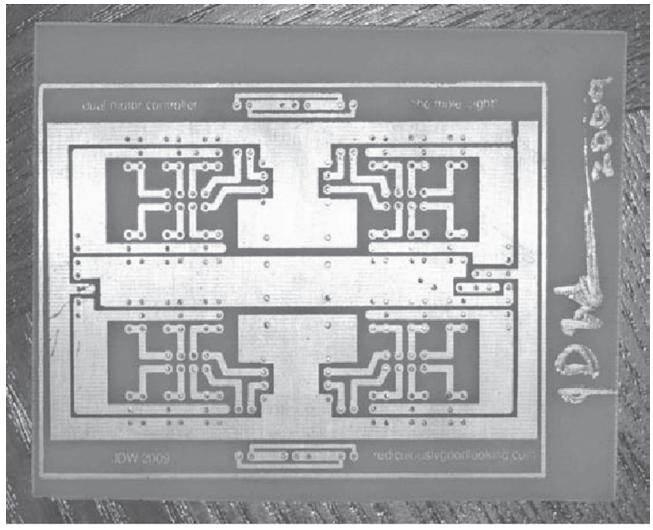


Figura 1.24 A placa de circuito impresso mostrada é um dos meus primeiros controladores de motor caseiros projetados no computador. Usando ao todo 28 chaves Mosfet para acionar dois motores CC, essa é a placa original usada no robô Lawn-bot (de 90,7 kg) apresentado no Capítulo 10.

Antes de finalizar um circuito eletrônico, é preciso soldar cada componente na PCI.

Soldagem

A solda elétrica em geral se refere à fusão de um componente eletrônico em uma PCI por meio de um ferro de solda e de uma solda elétrica, o que oferece uma conexão segura com a PCI. A ideia é fazer com que o terminal do componente e a ilha de cobre da PCI fiquem quentes o suficiente para que a solda se derreta quando os tocar. Por mais tentador que seja, você não deve aquecer apenas o fio de solda com o ferro de solda, porque ele só se fundirá ao terminal do componente e à ilha de cobre se eles também estiverem quentes.

Existem muitos tipos diferentes de solda, mas em conexões elétricas você deve usar uma solda elétrica com núcleo de breu, como mostrado na Figura 1.25. O uso de um fio de solda de menor diâmetro não requer temperatura extremamente alta para fundi-lo às ilhas de cobre e aos terminais dos componentes.

É melhor deixar o ferro aquecer completamente antes de tentar soldar. Não é possível soldar quando o ferro não está quente o suficiente, porque a solda não vai derreter! É recomendável soldar na ilha de cobre apenas o suficiente para preencher completamente os espaços ao redor do terminal do componente, mas não usar muita solda, porque isso cria uma bolha externa que pode tocar na outra ilha de componente ou trilha de cobre.

Você pode obter um ferro de solda por menos de 10 dólares na maioria das lojas de material eletrônico ou na Radio Shack. Embora os ferros sejam adequados para a maioria dos projetos, eles levam algum tempo para aquecer (cerca de dez minutos); além disso, como eles normalmente têm uma ponta grande, é difícil usá-los em lugares reduzidos.

Um ferro de solda de temperatura ajustável, com vários elementos de aquecimento, leva um minuto para aquecer e normalmente tem pontas menores para possibilitar a solda em projetos pequenos ou espaços reduzidos (ver Figura 1.26). É recomendável começar com um desses se você puder arcar com o custo: normalmente eles custam entre 50 e 150 dólares.

Figura 1.25 Um rolo de solda elétrica com núcleo de breu usado para a montagem de circuitos.

Figura 1.26 Usei ferros de solda baratos durante anos, até que minha esposa resolveu comprar um "bom" ferro de solda para mim. O Hakko 936 provavelmente não é perfeito, mas é de longe melhor do que os ferros de solda de 7-10 dólares que antes me faziam perder tempo. Ele aquece em questão de minutos e pode ficar muito mais quente do que um ferro normal, tornando a soldagem muito mais fácil.

Podemos levar algum tempo para nos acostumar com processo de soldagem. Por isso, antes de tentar montar uma PCI, é recomendável comprar uma placa de prototipagem perfurada e usá-la para praticar. Além disso, você pode comprar kits eletrônicos de vários fornecedores que vêm com todas as peças necessárias, PCI e instruções – você só precisará de um ferro de solda e de uma ou duas horas para a montagem. Comprei vários kits quando estava aprendendo a soldar, e eles me proporcionaram uma experiência de aprendizagem prática valiosa e interessante.

Atalhos para o processo de soldagem

Ao soldar placas perfuradas, às vezes pode existir um caminho livre entre duas ilhas com terminais que devem ser conectados. Para facilitar a solda e manter o circuito com todos os fios organizados, podemos usar alguns atalhos para simplificar as conexões (ver Figura 1.27).

- Opção 1 – Agrupamento com solda: você perceberá que, se aquecer ilhas de cobre adjacentes (mas separadas) e aplicar solda, a solda tenderá para ambas as ilhas, removendo espaço entre elas. Isso

porque a solda não gruda na fibra de vidro da PCI se não houver revestimento de cobre. Se você adicionar solda "em excesso" nessas duas ilhas, perceberá que a solda derretida tentará saltar para o outro conjunto de solda derretida na outra ilha. Se tiver cuidado, poderá deixar a solda solidificar-se entre as duas ilhas, criando uma conexão de solda simples. Esse método pode ser útil para criar uma ponte entre duas ou três ilhas adjacentes. Entretanto, esse método não é aceitável para ligações de alta potência, porque a solda não consegue transferir grande quantidade de corrente.

- Opção 2 – Trilhas com fio: você pode usar também um fio de cobre sólido desencapado (16-20 AWG) diretamente nas ilhas de cobre que deseja conectar (ver A, B e D na Figura 1.27). Se a conexão for abranger várias ilhas, é desejável aplicar uma pequena quantidade de solda em cada ilha em que o fio tocar para que ele não se mova depois que o circuito estiver completo. Você pode também dobrar o fio em torno de outros componentes para fazer uma linha curva ou angular. Esse método produz resultados semelhantes a uma trilha de PCI caseira. Como cada fio é ligado diretamente de um terminal para outro, não pode haver fios cruzados de outros componentes na face inferior da placa de circuito impresso. Esse método é aceitável para aplicações de alta corrente, embora um calibre de fio apropriado deva ser usado para a quantidade de corrente a ser transferida.

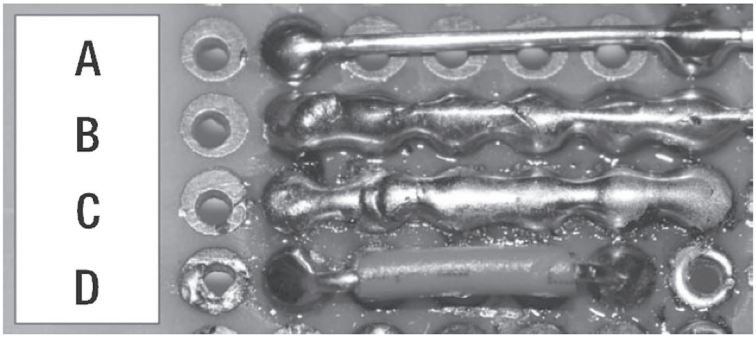


Figura 1.27 Como criar trilhas usando fio de cobre.

A trilha A é um fio desencapado e sem isolamento que é soldado apenas em cada extremidade. A trilha B é um fio desencapado que é soldado em cada ilha de cobre, o que o torna muito mais seguro que a trilha A. A trilha C nem sequer tem fio – é unicamente uma solda que une todas as seis ilhas. A trilha D é um fio que mantém seu isolamento, mas é soldado apenas em cada extremidade. É difícil conseguir a trilha C em mais de duas ou três ilhas e ela não é aceitável para aplicações de alta potência.

CONSTRUINDO UM ROBÔ

A construção real e prática de um robô é minha parte favorita do processo. É aqui que você começa a expressar sua criatividade por meio da concepção e construção de tudo o que você possa imaginar. Esse processo geralmente começa com algumas peças de metal ou madeira e alguns parafusos, porcas, barras roscadas, cola, fita adesiva e qualquer outra coisa que possa encontrar para dar vida a seu robô.

Primeiro você precisa decidir o que quer que seu robô faça e um objetivo (mesmo que seja simplesmente andar sem rumo). Você pode construir um robô autônomo que usa sensores para se orientar ou um robô controlado por rádio que usa entradas como controle. Se você nunca construiu um robô, provavelmente deverá começar por um pequeno. Vários dos capítulos deste livro utilizam servomotores do

tipo usado em modelismo para impulsionar o robô, os quais são conectados facilmente e diretamente ao Arduino sem a necessidade de um controlador de motor. Se incorporar menos peças em um projeto, ficará mais fácil e rápido montá-lo e modificá-lo, se necessário.

Seja o que for, não tente ser perfeito logo de cara. É melhor ter uma ideia satisfatória e um protótipo do que ter apenas um monte de *ideias verdadeiramente boas*. Por melhor que a ideia possa soar enquanto ainda estiver na cabeça, você não sabe se ela realmente funcionará se não a experimentar. Vários robôs deste livro passaram por VÁRIAS estruturas até eu encontrar uma que funcionasse e da qual gostasse. Se seu robô não funcionar como esperado na primeira tentativa, tome nota e tente novamente; é assim que um excelente robô é construído.

Fazer, testar, reformular, interromper, refazer – esse é o ciclo de *design*.

Hardware

Ter as ferramentas certas pode tornar o processo de construção muito mais fácil, mas nem todo mundo tem uma bancada totalmente abastecida. Como as ferramentas boas são caras, é recomendável comprar aquelas que você considera necessárias. Dessa forma você não terá ferramentas que nunca virá a usar.

Ferramentas básicas

Embora muitas ferramentas elétricas sejam em sua maioria opcionais, veja a seguir algumas ferramentas básicas que eu recomendo para começar. Você pode ir tão longe quanto quiser, mas esses itens têm de fazer parte de sua lista (ver Figura 1.28):

- Martelo
- Chaves de boca
- Alicates (padrão e de ponta fina)
- Descascadores e prensadores de fio
- Alicate de pressão
- Chaves de fenda comuns e Phillips
- Trena

Figura 1.28 Meu kit de ferramentas básicas: jogo de chaves de fenda 6 em 1 (centro superior), trena de 25 pés (cerca de 8 m) (canto superior direito), e da esquerda para a direita: martelo, chave inglesa, alicate universal, alicate de ponta fina, descascador de fio, prensador de fio, cortador de fio e alicate de pressão.

Assim que você tiver um conjunto de ferramentas básicas, poderá começar a adquirir ferramentas mais avançadas quando precisar delas (ou quando puder adquiri-las). Provavelmente você precisará também dos seguintes itens para construir todos os projetos deste livro – você não precisa ter todas as ferramentas, mas é imprescindível ter acesso a elas.

- Computador: apesar de normalmente não haver um computador na bancada, você precisará de um para executar o *software* do Arduino e fazer *upload* de código. Esse computador não precisa ser o melhor e mais recente para executar o *software* do Arduino IDE. Praticamente qualquer um com porta USB servirá. Tanto o *software* Arduino quanto o Eagle Cad podem ser usados em Windows, Linux ou Mac.
- Voltímetro (multímetro): esse medidor não precisa ser caro. Geralmente o mais barato que conseguir encontrar medirá tensão CA/CC, resistência e corrente CC em torno de 250 mA. Prefiro um medidor analógico para poder ver todos os movimentos da agulha e como ela está reagindo ao sinal que estou testando – meu medidor digital salta diretamente para a leitura, o que facilita a leitura de valores precisos, mas dificulta mudanças de tensão.
- Furadeira elétrica: você precisará de uma. É possível obter uma furadeira elétrica por 20 dólares em praticamente qualquer loja de ferragens. Se quiser optar por algo melhor, adquira um kit de furadeira de 18 V sem fio por cerca de 75 dólares. Também é útil ter uma furadeira de coluna, se pretende gravar sua PCI. A furadeira de coluna geralmente pode ser comprada por cerca de 60 dólares. Você precisará também de algumas brocas, se estiver pretendendo usar metal.
- Serra: você provavelmente precisará de várias serras, mas o tipo depende da quantidade de trabalho que deseja realizar. A serra mais barata com a qual você pode se virar para cortar metal é o arco de serra, mas os metais espessos exigem um pouco de paciência. Uma serra vaivém (às vezes chamada de serra sabre ou serra tudo) é uma boa opção para o corte de praticamente qualquer coisa, como metal, madeira e PVC. A serra tico-tico é suficiente, se você já tem uma. Apesar de um pouco menos versátil para projetos de robótica, há momentos em que a serra tico-tico pode ser útil.
- Ferro de solda: se você pretende montar qualquer um dos circuitos apresentados neste livro, precisará de um. Lembre-se de manter a ponta sempre limpa com uma esponja molhada ou escova de aço enquanto estiver soldando. Você pode usar um ferro de 7 dólares da Radio Shack, mas eu recomendo um modelo de temperatura controlada e ajustável, se estiver pensando em soldar com frequência. Esse tipo de ferro aquece bem mais rápido e fica muito mais quente, mas pode custar de 50 a 150 dólares.
- Soldador: não é necessário, mas pode ser útil em projetos maiores. Um tipo padrão de arame com alimentação de 110 V é adequado. Lembre-se sempre de usar máscara de solda para evitar danos aos olhos, e nunca olhe diretamente para o arco de solda!

Matéria-prima

Trabalharemos com vários materiais diferentes neste livro, como madeira, metal, plástico e fibra de vidro. É sempre bom usar óculos e luvas de proteção ao cortar qualquer desses materiais. Você pode trabalhar com o material com o qual se sentir mais confortável, mas prefiro metal.

- Madeira: a madeira é o material mais fácil e barato, que também é forte o suficiente para suportar o peso de um grande robô. Por mais tentador que seja usar alguns sarrafos 2×4 para o quadro de um robô, eles tendem a empenar e rachar, o que torna essa ideia menos atraente para um projeto no qual investimos muito tempo. No entanto, a madeira pode ser útil para prototipagem.
- Plástico: gosto de usar folhas de acrílico em vez de vidro para aplicações transparentes e pequenas bases de robô, porque são fáceis de perfurar e atarraxar e podem ser cortadas com uma serra tico-tico. O PVC (tubo) pode ser útil para projetos em que se precisa de pouco esforço. Plexiglas, PVC e muitos outros tipos de plástico podem ser formados ou moldados com um soprador de ar quente.

- Metal: é difícil martelar o metal para construir uma base móvel de robô. O metal extremamente resistente é durável e pode ser unido por meio de soldagem ou parafusos/porcas. O corte é um pouco mais difícil, e exige uma serra de arco (e força no braço) ou uma serra sabre com lâmina de metal com dente fino. Uma vez construída, a estrutura metálica durará anos e não empenará nem deformará. A maioria das lojas de ferragens vende longas barras de aço e outros metais de 122 cm com perfis em ângulos variados por 5-25 dólares, dependendo do tamanho e da espessura.
- Fibra de vidro: é um excelente material para a criação de formas específicas que seriam quase impossíveis com metal ou madeira. Também é extremamente forte e rígida, depois de preparada, bem como à prova d'água. O processo requer a colocação de um tecido de fibra de vidro e, em seguida, a aplicação de duas demãos de resina sobre ele. A fibra leva mais ou menos uma hora para endurecer, mas libera alguns gases fortes. Uma lata de 1 litro de resina de fibra de vidro custa em torno de 25 dólares na maioria das lojas de ferragens (e dura muito tempo), e o tecido de fibra de vidro especial (às vezes chamado de "manta") custa em torno de 5 dólares o pé quadrado (cerca de 0,74 m^2).

Área de trabalho

O ideal é ter um amplo espaço para trabalhar, mas geralmente o espaço depende de nossas condições de vida. Quando eu morava em apartamento, havia peças de projeto espalhadas por toda parte e eu usava a varanda de trás para trabalhar com metais, para desespero de meus vizinhos. Como agora tenho casa, tento manter todos os projetos na garagem e realizo a maior parte dos trabalhos de corte/trituração barulhentos fora, onde há boa ventilação.

Há várias coisas que você deve considerar ao escolher o lugar para montar seus projetos. Esses fatores muitas vezes são ignorados, mas são importantes para sua segurança e daqueles que estão ao seu redor.

- Espaço de teste: como as coisas nem sempre saem de acordo com o planejado, é bom ter muito espaço sempre que estiver testando um robô ativo que possa oferecer riscos físicos para os outros. Vários dos robôs apresentados neste livro são grandes o suficiente para ferir gravemente pessoas e animais de estimação em caso de perda de controle. Não teste robôs grandes em espaços fechados ou perto de pessoas!
- Ventilação: inspirar substâncias contaminantes pode ser prejudicial para os pulmões e o cérebro. No caso da serragem, pode ser apenas desconfortável, mas respirar gases ácidos e corrosivos ou fumaça de solda pode ser um perigo para a saúde. Sempre trabalhe em área bem ventilada ou externa. Se estiver trabalhando com solda, líquidos corrosivos ou fibra de vidro, é aconselhável usar máscara para proteger os pulmões e ventilador para extrair os gases nocivos da área de trabalho.
- Segurança: esteja sempre atento aos seus robôs. É um bom hábito desligar o cabo de energia quando não estiver usando o robô a fim de evitar arranque acidental e possíveis riscos. Não subestime a capacidade dos robôs de provocar danos e destruir (mesmo que essa seja sua finalidade) objetos aleatórios nas proximidades.
- Crianças: se seu espaço de trabalho puder ser acessado por crianças, procure manter o ferro de solda fora do alcance delas e desconectado, guardar qualquer lâmina ou objeto cortante e pequenos componentes que possam ser confundidos com balas e doces em lugar seguro e desligar e desativar a chave de segurança (isto é, a bateria) de qualquer robô que possa ferir uma criança. Vários projetos deste livro usam motores destinados a transportar uma pessoa. Esses motores são potentes o suficiente para provocar danos físicos a pessoas se não for possível controlar o robô. Por esse motivo, sugiro que você mantenha pessoas e animais de estimação a pelo menos 6 m de distância de robôs móveis (a menos que eles já tenham sido testados exaustivamente) por motivo de segurança.

RESUMO

Para recapitular, neste capítulo discutimos primeiro os conceitos básicos de eletricidade, incluindo uma analogia do fluxo de corrente elétrica, propriedades elétricas, circuitos e tipos de conexão. Depois de analisar a eletricidade, falamos sobre eletrônica e semicondutores, *datasheet*, circuitos integrados e encapsulamento de CI.

Em seguida, apresentamos uma breve introdução ao microcontrolador Arduino, incluindo o Arduino IDE, duas principais versões do Arduino (padrão e mega), componentes de um *sketch* e, finalmente, diferentes sinais comuns disponíveis no Arduino.

Com uma breve discussão sobre o projeto de circuitos eletrônicos e alguns dos diferentes tipos de símbolos esquemáticos usados para vários componentes eletrônicos, falamos sobre as ferramentas básicas de que você precisa para os projetos deste livro e os materiais que são usados.

No capítulo seguinte, analisaremos de que forma o Arduino pode ser interligado com uma série de dispositivos diferentes.

Se você quer montar robôs divertidos que podem fazer de tudo, desde seguir uma linha até ajudá-lo em trabalhos manuais – carregando suas ferramentas enquanto você conserta seu carro, por exemplo –, *Arduino para robótica* é o livro certo para você! Os autores vão ensiná-lo a pesquisar onde comprar as peças, como programá-las e como testá-las com segurança.

Você vai aprender fundamentos de Arduino e as características de diferentes tipos de robô. Também vai descobrir métodos de controle e à prova de falhas e aprender a aplicar esses métodos em seu projeto. O livro começa com fundamentos de robótica e movimentos e avança para projetos mais complexos, incluindo um robô explorador com GPS habilitado, um robô que corta grama, um que carrega coisas, um robô de combate e até mesmo uma versão "faça você mesmo" de um robô seguidor de linha.

O que você vai encontrar aqui:

- Tipos de motores e como controlá-los
- Projeto e fabricação de placas de circuito impresso
- Controle e decodificação de R/C
- Sensor autônomo de orientação
- Construção de chassi a partir de vários materiais
- Instruções para diversos projetos de robô

www.blucher.com.br
ISBN 978-85-212-1152-5
9 788521 211525
Blucher

Clique aqui e:

VEJA NA LOJA

Arduino para Robótica

John-David Warren , Josh Adams , Harald Molle

ISBN: 9788521211525

Páginas: 578

Formato: 18 x 23 cm

Ano de Publicação: 2019

Peso: 0.954 kg