Arduino Básico

2ª edição

Michael McRoberts

Novatec

Editor: Rubens Prates
Tradução: Joice Elias Costa
Revisão gramatical: Patrizia Zagni
Revisão técnica: Edgard Damiani
Editoração eletrônica: Carolina Kuwabata
Assistente editorial: Priscila A. Yoshimatsu

ISBN: 978-85-7522-404-5

Histórico de impressões:

Março/2015	Segunda edição
Outubro/2013	Quarta reimpressão
Fevereiro/2013	Terceira reimpressão
Agosto/2012	Segunda reimpressão
Janeiro/2012	Primeira reimpressão
Setembro/2011	Primeira edição (ISBN: 978-85-7522-274-4)

Novatec Editora Ltda.
Rua Luís Antônio dos Santos 110
02460-000 – São Paulo, SP – Brasil
Tel.: +55 11 2959-6529
E-mail: novatec@novatec.com.br
Site: www.novatec.com.br
Twitter: twitter.com/novateceditora
Facebook: facebook.com/novatec
LinkedIn: linkedin.com/in/novatec

CAPÍTULO 1

Começando

Desde o início do Arduino Project, em 2005, mais de 500 mil placas Arduino foram vendidas no mundo todo. Sem dúvida, o número de placas-clone não oficiais supera o de placas oficiais, sendo provável que mais de um milhão de placas Arduino e suas variantes tenham sido comercializadas. Sua popularidade não para de crescer e cada vez mais pessoas percebem o potencial sensacional desse incrível projeto open source e sua capacidade para criar projetos interessantes de forma rápida e fácil, com uma curva de aprendizagem relativamente baixa.

A maior vantagem do Arduino em relação a outras plataformas de desenvolvimento de microcontroladores é a sua facilidade de utilização, o que permite que pessoas que não sejam de áreas técnicas possam aprender o básico e criar seus próprios projetos em um período relativamente curto. Artistas, em especial, parecem considerá-lo a maneira ideal para criar obras de arte interativas rapidamente, sem a necessidade de um conhecimento especializado em eletrônica. Há uma imensa comunidade de pessoas usando Arduinos e compartilhando seus códigos e diagramas de circuito para que outros os copiem e modifiquem. A maior parte dessa comunidade tem também muita disposição para ajudar outras pessoas e fornecer orientações. O fórum do Arduino é o lugar ideal para visitar e buscar respostas rápidas.

No entanto, apesar da imensa quantidade de informações disponíveis na Internet para os iniciantes, a maioria delas se encontra espalhada em fontes variadas, dificultando que os iniciantes consigam obter aquelas que procuram. É aqui que entra este livro. Nas páginas que você está prestes a ler, há 50 projetos que foram desenvolvidos para conduzi-lo passo a passo pelo mundo da eletrônica para você aprender a programar o seu Arduino de forma fácil. A melhor maneira de aprender qualquer coisa é arregaçar as mangas e começar a trabalhar. Por isso, este livro não irá aborrecê-lo com páginas e mais páginas de teoria antes de você começar a utilizar o seu Arduino. Sei como é quando você utiliza pela primeira vez o seu Arduino ou qualquer outro gadget novo: você quer conectá-lo, ligar um LED e ver logo as luzes piscando, e não ter que ler páginas e páginas de manuais primeiro.

Como autor, entendo essa empolgação, por isso, logo no início, vamos conectar dispositivos ao nosso Arduino, fazer o upload do código e começar imediatamente. Creio que esta seja a melhor maneira de aprender um assunto e, em especial, no caso da computação física, que é do que se trata o Arduino.

Como utilizar este livro

O livro começa com uma introdução ao Arduino, falando sobre como montar o hardware, instalar o software, fazer o upload de seu primeiro sketch e garantir que o seu Arduino e o software estejam funcionando corretamente. Passamos, então, a explicar o IDE (Integrated Development Environment) do Arduino e como o utilizar, antes de mergulharmos em alguns projetos, progredindo desde coisas muito básicas até tópicos avançados. Cada projeto inicia com uma descrição de como preparar o hardware e qual código será necessário para fazê-lo funcionar. Depois disso, descreveremos o código e o hardware separadamente e explicaremos em detalhes o seu funcionamento. Tudo será exposto em passos claros e de modo fácil de acompanhar. Este livro contém muitos diagramas e imagens que lhe garantirão que você acompanhe o projeto da forma correta.

Você encontrará alguns termos e conceitos que talvez não entenda a princípio. Não se preocupe, pois ficarão mais claros à medida que você for trabalhando nos projetos.

Do que você vai precisar

A fim de ser capaz de acompanhar os projetos deste livro, você vai precisar de diversos componentes. A realização de todos os projetos vai exigir que você, primeiro, adquira muitos componentes. Isso pode ser caro, então sugiro que inicie comprando os componentes para os projetos que constam nos primeiros capítulos e obtenha as peças relacionadas no início das páginas dos projetos. À medida que avançar no livro, você poderá adquirir os componentes necessários aos projetos subsequentes.

Há uma série de outros itens de que você vai precisar ou que poderá considerar úteis. É claro, você deverá obter uma placa Arduino ou uma das muitas placas-clone disponíveis no mercado como Freeduino, Seeeduino (sim, escreve-se assim mesmo, com três eees), Boarduino, Sanguino, Roboduino ou quaisquer outras variantes "duino". Todas são totalmente compatíveis com o IDE do Arduino, com os shields do Arduino e com tudo o mais que você pode utilizar com uma placa

Arduino oficial. Lembre-se de que o Arduino é um projeto open source. Portanto, qualquer pessoa é livre para criar clones ou outras variantes dele. No entanto, se você deseja apoiar a equipe de desenvolvimento da placa Arduino original, adquira uma placa oficial de um dos distribuidores reconhecidos. Para os projetos deste livro, utilizaremos uma placa Arduino Uno, embora qualquer uma das placas Arduino disponíveis funcione da mesma forma.

Você deverá ter acesso à Internet para fazer o download do IDE do Arduino, o software utilizado para escrever o código do Arduino, para fazer o upload dele para a placa e também para fazer o download dos exemplos de códigos que constam neste livro (caso você não queira digitá-los), e ainda de todas as bibliotecas de código que possam ser necessárias para que o seu projeto funcione.

Você também vai precisar de uma mesa bem iluminada ou outra superfície plana para dispor os seus componentes – essa mesa deverá estar próxima ao seu computador ou laptop para permitir que você faça o upload do código para o Arduino. Lembre-se de que você está trabalhando com eletricidade (embora se trate de uma corrente contínua de baixa voltagem) e, portanto, mesas ou superfícies metálicas deverão ser cobertas com um material não condutor, como uma toalha de mesa ou um papel, antes que você espalhe os seus materiais sobre elas. Embora não seja essencial, também poderá ser interessante ter um alicate para cortar fios, um alicate de ponta fina e um desencapador de fios. Também será útil ter um caderno e uma caneta à mão para esboçar diagramas aproximados e desenvolver conceitos e projetos.

Por fim, as coisas mais importantes que você precisa ter são entusiasmo e disposição para aprender. O Arduino foi projetado como uma maneira simples e barata de você se envolver com a eletrônica de microcontroladores, e não há nada difícil demais de aprender desde que você esteja disposto a tentar. Este livro vai ajudá-lo nesta jornada e apresentá-lo a esse hobby empolgante e criativo.

O que é exatamente um Arduino?

A Wikipédia afirma que "o Arduino é um microcontrolador de placa única, projetado para tornar mais acessível o processo de utilização da eletrônica em projetos multidisciplinares. O hardware consiste em um dispositivo open source simples projetado para um microcontrolador Atmel AVR de 8 bits, embora um modelo novo tenha sido projetado para um Atmel ARM de 32 bits. O software consiste em uma linguagem de programação padrão e do bootloader que roda no microcontrolador" (Figura 1.1).

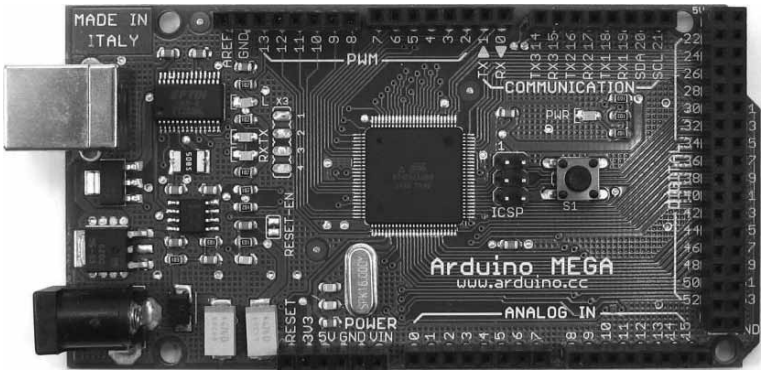

Figura 1.1 – Arduino Mega (imagem de David Mellis).

Para colocar a definição anterior em termos leigos, um Arduino é um computador minúsculo que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos que conectar a ele. O Arduino é o que chamamos de plataforma de computação física ou embarcada. Por exemplo, um uso simples de um Arduino seria para acender uma luz por um determinado período, digamos, durante 30 segundos, depois que um botão fosse pressionado. Nesse exemplo, o Arduino teria uma lâmpada conectada a ele, bem como um botão. O Arduino aguardaria pacientemente até que o botão fosse pressionado. Ao pressionar o botão, o Arduino acenderia a lâmpada e iniciaria a contagem. Depois de contados 30 segundos, apagaria a lâmpada e continuaria no aguardo de um novo apertar do botão. Você poderia utilizar essa configuração para controlar uma lâmpada dentro de um armário, por exemplo.

Esse conceito pode ser estendido de modo que o dispositivo possa detectar quando a porta do armário é aberta ou quando algum outro evento acontece e, automaticamente, acender a lâmpada, desligando-a depois de um determinado intervalo de tempo. Você pode ir além e conectar um sensor infravermelho passivo (PIR) para detectar movimentos e ligar a lâmpada quando é disparado. Esses são alguns exemplos simples de como utilizar o Arduino.

O Arduino pode ser usado para desenvolver objetos interativos independentes ou ser conectado a um computador, a uma rede ou até mesmo à internet para recuperar e enviar dados do Arduino e trabalhar com eles. Por exemplo, pode enviar um conjunto de dados recebidos de sensores para um site, para serem exibidos no formato de um gráfico.

O Arduino pode ser conectado a LEDs, displays de matriz de pontos (Figura 1.2), botões, interruptores, motores, sensores de temperatura, sensores de pressão, sensores de distância, receptores GPS, módulos Ethernet e Wifi ou a qualquer

outro dispositivo que emita dados ou que possa ser controlado. Uma busca na internet revelará uma profusão de projetos em que um Arduino foi utilizado para ler dados de dispositivos ou controlar uma quantidade incrível de dispositivos.

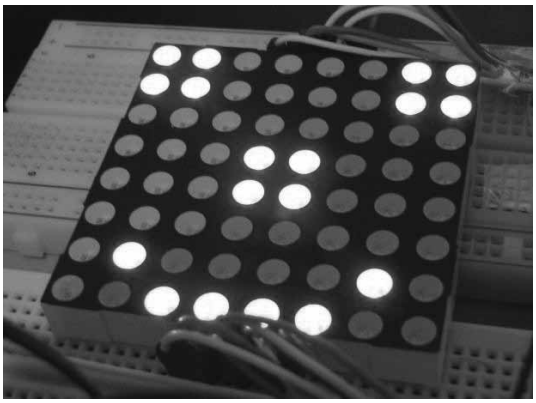

Figura 1.2 – Display de matriz de pontos controlado por um Arduino (imagem cortesia de Bruno Soares).

A placa Arduino é composta de um microprocessador Atmel AVR, um cristal ou oscilador (um relógio simples que envia pulsos de tempo em uma frequência especificada para permitir sua operação na velocidade correta) e um regulador de voltagem de 5 volts (alguns Arduinos podem usar um regulador de voltagem chaveado e outros, como o Due, não são compatíveis com 5 volts). Dependendo do tipo de Arduino que você tem, ele também pode ter uma saída USB que permita conectá-lo a um PC ou Mac a fim de fazer upload ou recuperar dados. A placa expõe os pinos de entrada/saída do microcontrolador para que você os conecte a outros circuitos ou a sensores etc.

Para programar o Arduino (fazer com que ele faça o que você deseja), você utiliza o IDE dele, que é um software livre que lhe permite programar na linguagem que ele entende. No caso do Arduino, a linguagem é baseada em C/C++ e pode até ser estendida por meio de bibliotecas C++. O IDE permite que você escreva um programa de computador, que é um conjunto de instruções passo a passo do qual você, então, deverá fazer o upload para o Arduino. A seguir, o seu Arduino executará essas instruções e interagirá com o que quer que você tenha conectado a ele. No mundo do Arduino, os programas são conhecidos como *sketches* (literalmente, rascunho ou esboço).

O hardware e o software são ambos open source, o que significa que o código, os esquemas, o projeto etc. são abertos e qualquer pessoa pode usá-los livremente para fazer o que desejar. Dessa forma, há muitas placas-clone e outras placas com base no Arduino disponíveis no mercado, ou que podem ser criadas a partir

de um diagrama. Na verdade, nada impede que você compre os componentes apropriados e desenvolva o seu próprio Arduino em uma protoboard ou em sua própria PCB (placa de circuito impresso) feita em casa. A única condição imposta pela equipe do Arduino é que você não pode utilizar a palavra *Arduino*, pois esse nome é reservado à placa oficial. Por isso, todas as placas-clone têm nomes como Freeduino, Roboduino etc.

O Arduino também pode ser estendido com a utilização de *shields* (escudos), que são placas de circuito que contêm outros dispositivos (por exemplo, receptores GPS, displays de LCD, módulos de Ethernet etc.) que você pode conectar à parte superior do seu Arduino para obter funcionalidades adicionais. Os shields também estendem os pinos (os lugares no seu Arduino que você pode usar para entrada ou saída de dados) até o topo de suas próprias placas de circuito e, assim, você continua a ter acesso a todos eles. Você não precisa utilizar um shield se não quiser, pois pode fazer exatamente o mesmo circuito usando uma protoboard, uma Stripboard, uma Veroboard (placas feitas de tiras de cobre em uma matriz para projetos caseiros de soldagem) ou criando a sua própria PCB. A maior parte dos projetos apresentados neste livro é feita com a utilização de circuitos em uma protoboard.

Como os projetos são open source, uma placa clone como a Freeduino pode ser 100% compatível com o Arduino e, portanto, com quaisquer softwares, hardwares, shields etc. Alguns clones são compatíveis em muitos aspectos, mas apresentam diferenças intencionais a fim de suportar recursos especiais. Além disso, o Due (que é um Arduino genuíno) apresenta alguns problemas, como o fato de operar em 3 volts, o que pode não funcionar com todos os shields.

Há muitas versões diferentes do Arduino disponíveis. A mais comum é a versão Uno – lançada em 2010 (atualmente na Revisão 3), que é a placa que você provavelmente encontrará na maioria dos projetos para Arduino na Internet. Você também pode obter versões Due Leonardo, Duemilanove, Mega 2560, Mega ADK, Fio, Arduino Ethernet, Mini, Nano, Lilypad e Bluetooth. As adições mais recentes à linha de produtos são a Arduino Leonardo e a Arduino Due, que é a primeira incursão da equipe do Arduino na utilização de processadores ARM em vez de processadores com arquitetura AVR. O Due tem um processador de 32 bits em vez do habitual processador de 8 bits das outras variantes do Arduino, executa a uma frequência de 84 Mhz e tem 512 KB de memória flash.

Talvez o Arduino mais versátil, e daí o motivo de ser tão popular, seja o Uno (antes do Uno, o Duemilanove era o mais popular). Isso acontece porque ele utiliza um chip-padrão de 28 pinos ligado a um soquete de CI (circuito integrado). A beleza desse sistema é que se você criar algo interessante com um Arduino e depois quiser transformá-lo em alguma coisa permanente, em vez de utilizar uma

placa Arduino relativamente cara, poderá simplesmente usar o Arduino para desenvolver o seu dispositivo e programar o chip, depois poderá retirar o chip da placa e colocá-lo em sua própria placa de circuito no seu dispositivo personalizado. Assim, você terá criado um dispositivo embarcado personalizado, o que é muito bacana. A seguir, com um investimento um pouco maior, você pode substituir o chip AVR do seu Arduino por um novo. O chip deve ser pré-programado com o Arduino Bootloader (software programado no chip para permitir que ele seja utilizado com o IDE do Arduino), mas você pode comprar um AVR Programmer para gravar o bootloader você mesmo, ou comprar um chip já programado, que é oferecido pela maioria dos fornecedores de componentes para o Arduino.

Em relação ao Arduino anterior, o Duemilanove, o mais novo Arduino Uno tem a vantagem de ter um chip USB programável na placa, o que permite que você acesse o chip de tal forma que quando conectar o dispositivo no seu PC, ele vai aparecer como qualquer tipo de dispositivo USB que você desejar, por exemplo, como um teclado, um mouse ou um joystick. Isso permite que você utilize o Arduino como uma interface para criar seus próprios dispositivos USB. No entanto, esse é um recurso avançado e recomendado só para os mais destemidos.

Se você fizer uma busca na Internet por "Arduino", ficará surpreso diante da imensa quantidade de sites dedicados ao Arduino ou sites nos quais alguém utilizou um Arduino para desenvolver um projeto interessante. O Arduino é um dispositivo incrível que lhe permitirá criar tudo, desde obras de arte interativas (Figura 1.3) até robôs. Com um pouco de entusiasmo em relação a aprender como programar um Arduino e fazê-lo interagir com outros componentes, além de ter um pouco de imaginação, você poderá construir tudo o que imaginar.

Figura 1.3 – Instalação de arte Anthros, por Richard V. Gilbank, controlada utilizando um Arduino.

Este livro fornecerá as habilidades necessárias para que você inicie este hobby empolgante e criativo. Com isso, agora que você já sabe o que é o Arduino, vamos ligá-lo ao nosso computador e começar a usá-lo.

Configurando o seu Arduino

Esta seção explicará como configurar o seu Arduino e o IDE pela primeira vez. Serão fornecidas as instruções tanto para Windows quanto para Mac. Caso você utilize Linux, consulte as instruções da seção Getting Started no site do Arduino, em *http://playground.arduino.cc/learning/linux*. Vou também presumir que você esteja utilizando um Arduino Uno (Figura 1.4), Duemilanove, Nano, Diecimila ou Mega 2560 (ou um clone equivalente a estes) e esteja instalando no Windows 7 ou em alguma versão recente do OSX (Lion ou Mountain Lion). Se tiver uma placa diferente, consulte a página correspondente na guia Getting Started do site do Arduino.

Você também vai precisar de um cabo USB (com plug do tipo A para o tipo B), que é o mesmo tipo de cabo utilizado na maioria das impressoras USB modernas. Se tiver um Arduino Nano, em vez disso, vai precisar de um cabo USB A para Mini-B.

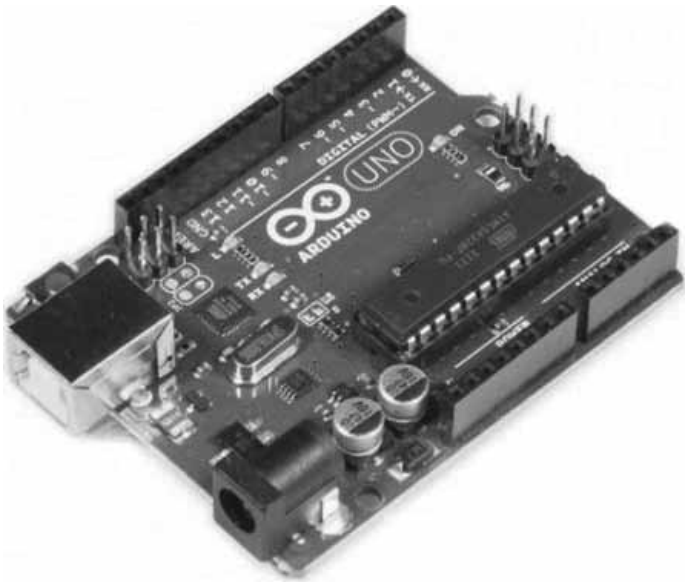

Figura 1.4 – Arduino Uno (imagem cortesia de Earthshine Electronics).

A seguir, você precisa fazer o download do IDE do Arduino. Esse é o software que você vai utilizar para escrever os seus programas (ou sketches) e fazer o upload deles para a sua placa. Para encontrar o IDE mais recente, acesse a página de

download do Arduino em *http://arduino.cc/en/Main/Software* e obtenha a versão apropriada para o seu sistema operacional.

Se tiver um Mac, depois de fazer o download do arquivo Zip, efetue a descompactação do arquivo e, então, verá o ícone do Arduino. Arraste esse ícone para a pasta *Aplicativos* e solte-o nela para instalar o programa. Você deve simplesmente clicar duas vezes no ícone para começar a instalação. No Windows, faça o download do arquivo ZIP e, quando este for concluído, a descompactação do arquivo. A seguir, coloque a pasta descompactada no lugar que preferir, mantendo a estrutura de diretórios intacta.

Agora você precisa conectar a sua placa Arduino antes de instalar os drivers e o software. Conecte o cabo USB ao Arduino e, depois, a outra extremidade a uma porta USB no seu computador. Você vai ver o LED verde (PWR) de alimentação acender na sua placa para mostrar que ele está ligado. Se estiver usando um Mac, não há drivers para instalar. Se estiver usando Windows, então agora o sistema vai tentar instalar os drivers para o Arduino. Vai ocorrer uma falha nessa tentativa automática de instalação e você vai receber uma mensagem dizendo que "device driver software was not successfully installed" ("O driver do dispositivo não foi instalado com sucesso.") (Figura 1.5). Não se preocupe com isso.

Figura 1.5 – Tentativa automática do Windows de instalar não será bem-sucedida. Isso é normal.

Clique no menu Iniciar e, depois, em Painel de Controle. Navegue até Sistema e Segurança, clique em Sistema e, depois, abra o Gerenciador de Dispositivos. Na lista de hardwares logo abaixo de Outros Dispositivos, você deverá encontrar algo semelhante ao apresentado na figura 1.6, onde há um item "Arduino Uno" com um ícone amarelo de aviso.

Clique com o botão direito do mouse sobre o ícone do Arduino Uno e escolha a opção Atualizar Driver (Figura 1.7).

Agora, escolha Procurar o driver no meu computador (Figura 1.8).

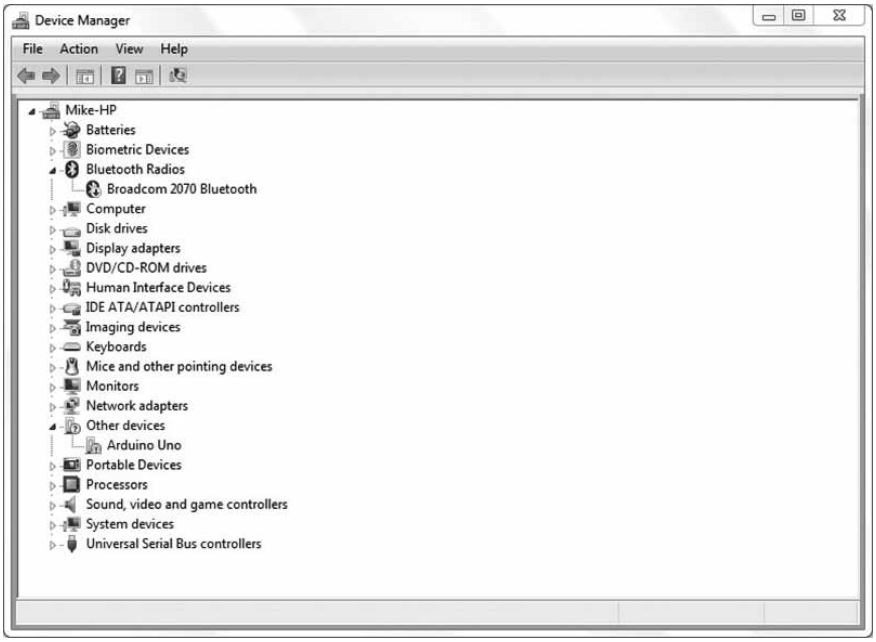

Figura 1.6 – Gerenciador de dispositivos do Windows.

Figura 1.7 – Clique com o botão direito e escolha "Atualizar Driver".

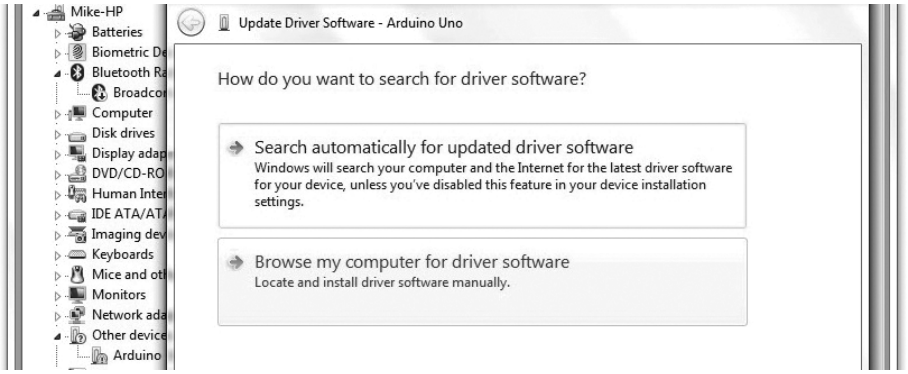

Figura 1.8 – Clique em "Procurar o driver no meu computador".

A seguir, vá até a pasta do driver de instalação do Arduino e, depois, clique no botão **Next** (Avançar). Agora, o Windows concluirá a instalação do driver. Se aparecer uma mensagem que diz "Windows can't verify the publisher of this driver software" (O Windows não consegue verificar o fornecedor deste driver), então clique em **Install this driver software anyway** (Instalar este driver mesmo assim). Se você tiver um Mac, não haverá drivers para instalar.

Agora que os drivers estão instalados, você está pronto para abrir o IDE do Arduino. No Windows, clique duas vezes sobre o arquivo **arduino.exe** que você encontrará dentro da pasta Arduino que você descompactou. No Mac, clique no ícone Arduino dentro da pasta *Aplicativos*. Agora, o IDE vai abrir e apresentar a você um sketch em branco conforme mostra a figura 1.9.

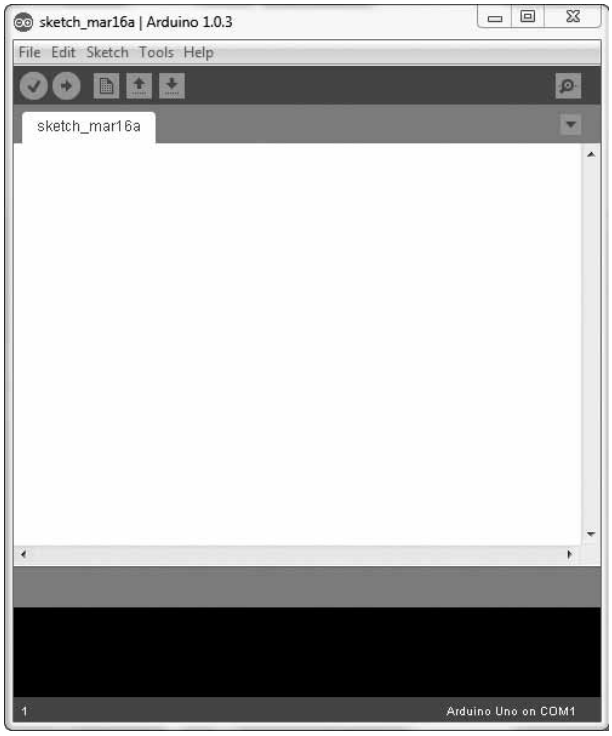

Figura 1.9 – O IDE do Arduino.

A seguir, abra um sketch de exemplo para testar o IDE e o Arduino. Clique em **File**, depois em **Examples**, depois em **01.Basics** e, por fim, em **Blink** (Figura 1.10).

Isso carregará o sketch de exemplo Blink no IDE e você verá algo semelhante ao que aparece na figura 1.11.

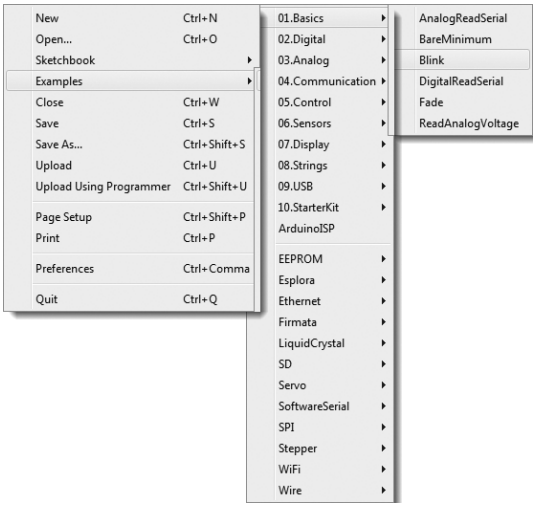

Figura 1.10 – Menu File do Arduino. Escolha o sketch Blink.

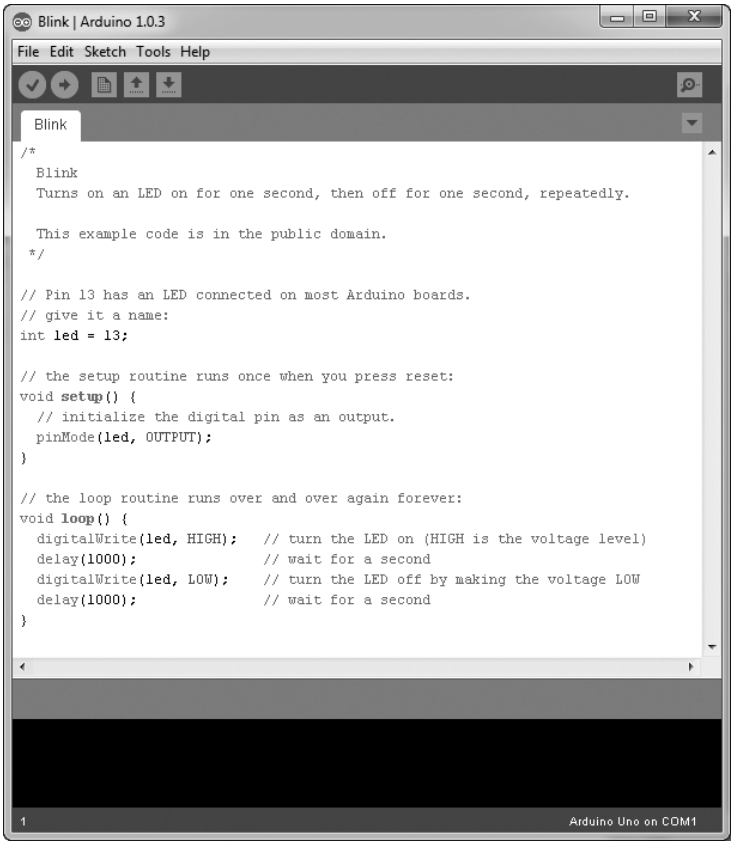

Figura 1.11 – IDE com o sketch Blink carregado.

Depois disso, você terá que escolher a sua placa com base na lista (Figura 1.12) em **Tools > Board**. No caso de um **Arduino Uno**, selecione-o no topo da lista. Se utilizar um Arduino Duemilanove mais antigo ou um clone com um chip Atmega328, terá que selecionar **Arduino Duemilanove or Nano w/ Atmega328**. Se você tem uma placa ainda mais antiga com um chip Atmega168, selecione **Arduino Diecimila, Duemilanove, or Nano w/ ATmega168**, ou você pode ainda ter uma **Leonardo**, **Mega** ou uma **DUE**. Selecione a placa que corresponde à sua.

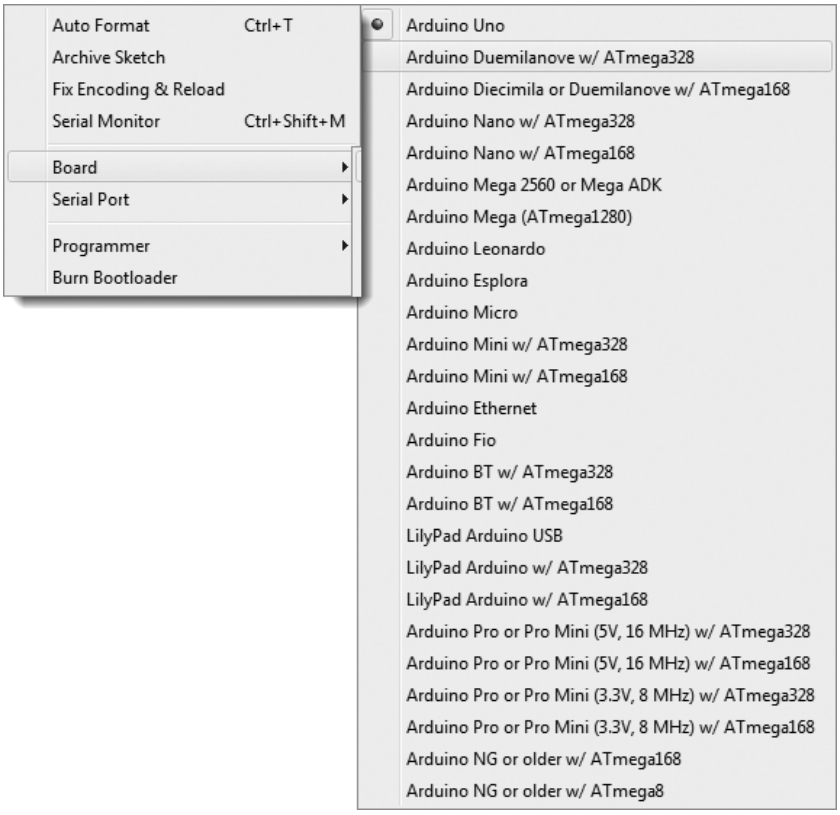

Figura 1.12 – Selecione o tipo de placa que você utiliza.

Selecione o dispositivo serial da placa Arduino a partir de **Tools > Board** (Figura 1.13). Se não tiver certeza de qual é a porta correta, desconecte o Arduino, verifique as portas disponíveis e depois o reconecte para ver qual porta aparece agora (pode ser que você precise fechar e reabrir o menu para que a porta apareça).

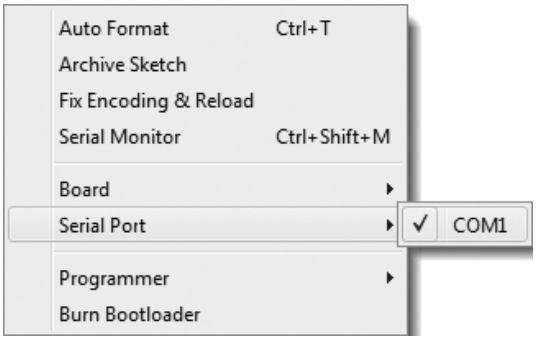

Figura 1.13 – Selecione a porta.

Upload do seu primeiro sketch

Agora que você instalou os drivers e o IDE e tem a placa e a porta corretas selecionadas, pode fazer o upload do sketch de exemplo Blink para o Arduino a fim de testar se tudo está funcionando de forma apropriada antes de avançar para o seu primeiro projeto. Depois de ter carregado o sketch Blink no IDE do Arduino, você pode fazer o upload dele para o Arduino simplesmente clicando no botão Upload (o segundo botão a partir da esquerda que tem uma seta voltada para a direita) e olhar para o seu Arduino (se você tem um Arduino Mini, NG ou outra placa, pode ser necessário apertar o botão reset na placa antes de pressionar o botão Upload). O IDE vai dizer "Compiling sketch..." (Compilando sketch...), o que depois vai mudar para "Uploading...". A seguir, as luzes RX e TX devem começar a piscar para mostrar que os dados estão sendo transmitidos do seu computador para a placa. Quando o upload do sketch tiver sido concluído, a mensagem "Done Uploading" (Upload completo) será exibida na barra de status do IDE e as luzes RX e TX pararão de piscar.

Após alguns segundos, você deverá ver que o LED do pino 13 (o LED pequenino acima dos LEDs TX e RX) começará a piscar, acendendo e apagando em intervalos de 1 segundo. Se isso acontecer, significa que você conectou o seu Arduino, instalou os drivers e o software e realizou o upload de um sketch de exemplo corretamente. O sketch Blink é um sketch muito simples que pisca o LED 13, que é o LED pequenino, laranja e soldado na placa que também está conectado ao pino digital 13 do microcontrolador (Figura 1.14).

Antes de avançarmos para o projeto 1, vamos dar uma olhada no IDE do Arduino para explicar o que cada uma das partes do programa faz.

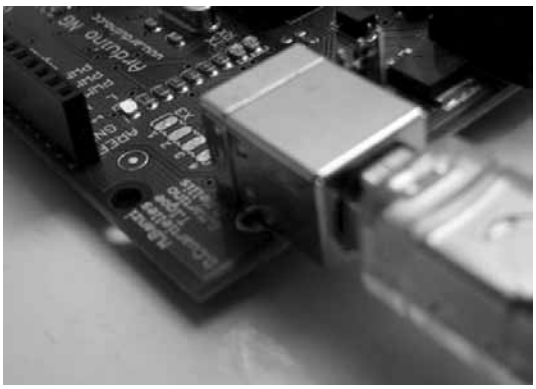

Figura 1.14 – LED 13 piscando.

IDE do Arduino

O IDE (Integrated Development Environment) é o que você vai usar para escrever o código para o seu Arduino, verificá-lo e fazer o upload dele para a sua placa. A versão atual do IDE, 1.x, foi lançada em novembro de 2011. Antes disso, os números da versão Beta foram de 0001 a 0023 e a versão 1.0 foi a primeira versão *release candidate* do software. Na versão 1.0, as extensões dos arquivos dos sketches mudaram de *.pde* para *.ino* para evitar conflitos com o software Processing (o Processing é o projeto no qual se baseia o IDE original). Houve também algumas mudanças importantes feitas na linguagem do Arduino. Se quiser converter o código mais antigo do Arduino em um novo IDE, deverá procurar no site do Arduino e ler a seção de referência sobre como funcionam os comandos, caso ocorram erros com o código mais antigo.

Quando você abrir o IDE do Arduino, encontrará algo muito semelhante à versão do Windows na figura 1.15. Caso esteja utilizando OSX ou Linux, pode haver algumas diferenças pequenas, mas o IDE é praticamente o mesmo, independentemente do sistema operacional utilizado.

O IDE é dividido em quatro partes: o menu de opções no topo do programa (ou no topo da sua tela, no OSX), a barra de ferramentas logo abaixo, o código ou Sketch Window ao centro e a janela de mensagem na base. A barra de ferramentas consiste em seis botões e, abaixo dela, há uma guia ou conjunto de guias com o nome do arquivo do sketch mostrado dentro da guia. Há também outro botão no lado direito que aciona a janela do Serial Monitor.

Ao longo do topo, fica o menu com as opções Arduino (apenas no OSX), File, Edit, Sketch, Tools e Help. Os botões da barra de ferramentas (Figura 1.16) fornecem acesso prático às funções mais utilizadas do menu.

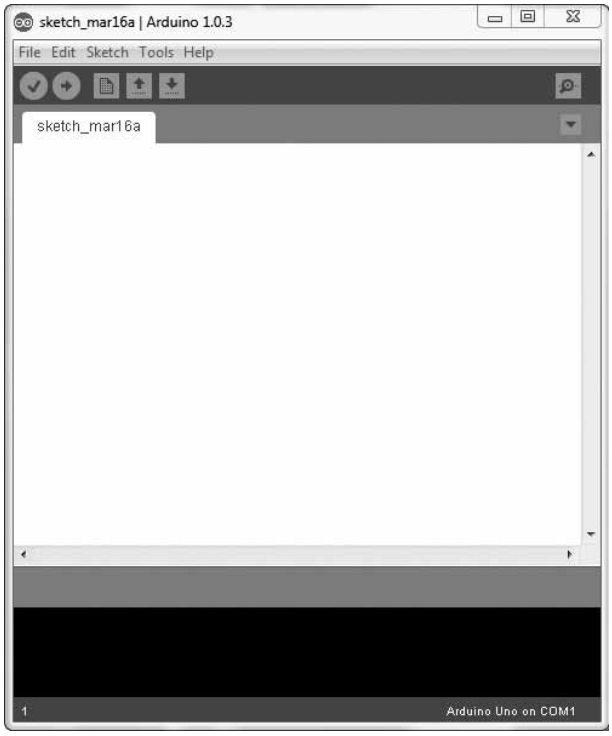

Figura 1.15 – Visual do IDE quando a aplicação abre.

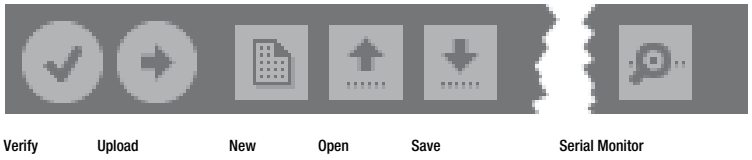

Figura 1.16 – Barra de ferramentas.

Os botões da barra de ferramentas aparecem listados na figura 1.16. As funções de cada um dos botões são apresentadas na tabela 1.1.

Tabela 1.1 – Funções dos botões da Toolbar

Verify	Verifica se há erros no código
Upload	Faz o upload do sketch atual para o Arduino
New	Cria um sketch em branco
Open	Exibe uma lista de sketches para serem abertos no seu Sketchbook
Save	Salva o sketch atual no seu Sketchbook
Serial Monitor	Exibe os dados seriais enviados a partir do Arduino

O botão Verify é utilizado para verificar se o seu código está correto e sem erros antes de você efetuar o upload do código para a sua placa Arduino.

O botão Upload faz o upload do código contido na janela de sketch atual para o seu Arduino. Você deverá se certificar de que selecionou corretamente a placa e a porta (no menu Tools) antes de fazer o upload. É fundamental salvar o seu sketch antes de fazer o upload dele para a sua placa, caso ocorra algum erro estranho que provoque um travamento do seu sistema ou do IDE. Também é aconselhável verificar o código antes de fazer o upload para garantir que não existam erros que precisem ser depurados primeiro.

O botão New vai criar um sketch novo e em branco, pronto para que você insira o seu código. O IDE pedirá que você digite um nome e forneça uma localização para o seu sketch (se possível, tente utilizar a localização-padrão) e, a seguir, abrirá para você um sketch em branco, pronto para receber o seu código. A guia no topo do sketch agora exibirá o nome que você deu ao seu novo sketch.

O botão Open apresentará uma lista de sketches armazenados em seu sketchbook, bem como uma lista de sketches de exemplo que você pode experimentar com diversos periféricos, quando conectados. Os sketches de exemplo são extremamente valiosos para os iniciantes, que poderão utilizá-los com uma base a partir da qual poderão criar seus próprios sketches. Abra o sketch apropriado para o dispositivo que você estiver conectando e modifique o código de acordo com a sua necessidade.

O botão Save salvará o código dentro da janela de Sketch para o seu arquivo sketch. Quando for concluído, você receberá uma mensagem "Done Saving" (Gravação completa) na base da sua janela de código, o que indica que o sketch foi salvo corretamente.

O botão Serial Monitor é uma ferramenta muito útil, em especial para depurar o seu código. O monitor exibe os dados seriais enviados do seu Arduino (USB ou placa Serial). Você também pode enviar dados de volta ao Arduino utilizando o Serial Monitor. Se clicar no botão Serial Monitor, verá uma imagem como a apresentada na figura 1.17.

No canto inferior direito, você pode selecionar a taxa de transmissão (Baud Rate) na qual os dados devem ser enviados de/para o Arduino. A taxa de transmissão (Baud Rate) é a taxa, por segundo, em que as alterações de estado ou bits (dados) são enviadas de/para a placa. A configuração-padrão é 9.600 baud, o que significa que se você quisesse enviar um livro pela linha de comunicação serial (nesse caso, o seu cabo USB), logo seriam enviados 1.200 letras ou símbolos de texto por segundo (9.600 bits/8 bits por caractere = 1.200 bytes ou caracteres – bits e bytes serão explicados posteriormente).

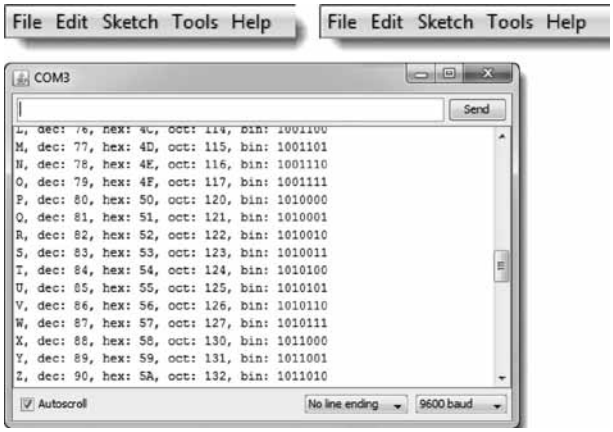

Figura 1.17 – Serial Monitor em uso.

No topo, há uma caixa de texto em branco para que você digite o texto a ser enviado de volta para o Arduino, além de um botão **Send** para enviar o texto escrito dentro desse campo. Observe que o Serial Monitor não poderá receber nenhum dado serial a menos que você tenha preparado o código dentro do seu sketch para enviar dados seriais a partir do Arduino. Da mesma forma, o Arduino não receberá nenhum dado enviado a menos que você o tenha codificado para isso.

Há uma caixa de seleção no canto inferior esquerdo, onde você pode selecionar se deseja ou não que os dados da janela do Serial Monitor rolem automaticamente (autoscroll).

A caixa à esquerda do menu da taxa de transmissão (baud rate) afetará os dados enviados do Serial Monitor de volta para o Arduino. A configuração-padrão é "no line ending" (sem fim de linha), o que significa que quando você inserir os dados na caixa de texto do Serial Monitor e apertar o botão **send**, os dados serão enviados exatamente como estiverem. Se você clicar no menu drop-down, verá que há outras três opções: **Newline**, **Carriage Return** e **Both NL+Cr**. Ao selecionar uma delas, o Serial Monitor acrescentará um código ASCII a Newline, Carriage Return ou ambos ao final de quaisquer dados inseridos na janela do Serial Monitor quando você clicar em **send**. Tenha isso em mente quando estiver processando dados enviados do Serial Monitor de volta para o Arduino.

Por fim, a área central da janela é o lugar onde os seus dados seriais serão exibidos. Na imagem mostrada anteriormente, o Arduino está executando o sketch ASCIITable que pode ser encontrado nos exemplos do submenu **Communication**. Esse programa faz a saída de caracteres ASCII do Arduino via serial (o cabo USB) para o PC, onde o Serial Monitor, então, os exibe.

Assim que estiver hábil na comunicação serial de e para o Arduino, você poderá usar outros programas como Processing, Flash, MaxMSP etc. para realizar a comunicação entre o Arduino e o seu PC.

Vamos utilizar o Serial Monitor posteriormente em nossos projetos quando formos ler dados de sensores e fazer o Arduino enviar esses dados para o Serial Monitor, em formato legível para seres humanos.

Na janela de mensagens na parte inferior do IDE, você encontrará mensagens de erro que o IDE mostrará a você quando tentar se conectar com a sua placa, realizar o upload do código ou verificá-lo.

Abaixo da janela de mensagens, no canto inferior esquerdo, você verá um número. Essa é a linha atual do cursor dentro da janela de código. Se você tiver código em sua janela e mover o cursor para baixo ao longo das linhas de código (utilizando a seta para baixo do seu teclado), verá que esse número aumenta à medida que move o cursor. Isso é útil para encontrar bugs destacados por mensagens de erro.

No topo da janela do IDE (ou no topo da sua tela, se estiver utilizando um Mac), você verá os diversos menus em que poderá clicar para acessar mais itens de menu (Figura 1.18).

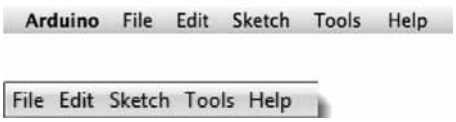

Figura 1.18 – Menus do IDE (parte superior: OSX; parte inferior: Windows).

O primeiro menu (no OSX) é o **Arduino** (Figura 1.19). Dentro dele, há a opção **About Arduino** que, quando acionada, lhe mostrará o número da versão atual, uma lista de pessoas envolvidas na criação desse dispositivo fantástico e algumas informações adicionais. Em PCs com Windows, o item **About Arduino** fica no menu **Help**.

Figura 1.19 – Menu About Arduino.

O próximo menu é o File (Figura 1.20). Aqui, você tem acesso a opções para criar um novo sketch, analisar os sketches armazenados no seu Sketchbook, arquivos de exemplo e opções para salvar o seu Sketch (ou Save As, caso você queira usar outro nome). Você também tem a opção de fazer o upload do seu sketch para o Arduino, fazer o upload utilizando um programador (não utilizaremos esse recurso) ou usar as opções de impressão para imprimir o seu código.

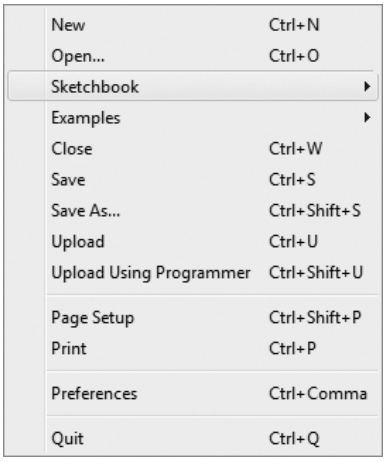

Figura 1.20 – Menu File.

Perto da base fica a opção Preferences, que acionará a janela Preferences na qual você pode alterar as várias opções do IDE, como a localização-padrão de armazenamento do seu Sketchbook etc. Por fim, há a opção Quit, que encerra o programa.

O próximo é o menu Edit (Figura 1.21). Aqui, você encontra opções que permitem recortar, copiar e colar seções de código, selecionar todo o código, bem como determinadas palavras ou frases dentro dele. Você pode comentar o seu código (acrescentando comentários para explicar como ele funciona) e também aumentar ou diminuir sua endentação. Estão também incluídas as opções Undo (desfazer) e Redo (refazer), que são úteis quando se comete algum erro.

O nosso próximo menu é o Sketch (Figura 1.22), que nos dá acesso às funções Verify/Compile e a algumas outras que usaremos mais tarde. Tais funções incluem a opção Import Library que, quando clicada, apresenta uma lista das bibliotecas disponíveis armazenadas em sua pasta de bibliotecas.

Uma biblioteca é um conjunto de código que você pode incluir no seu sketch para aprimorar a funcionalidade do seu projeto. É uma maneira de impedir que você tenha que reinventar a roda ao reutilizar o código que já foi escrito por outra pessoa para vários componentes de hardware comuns que você poderá encontrar no momento em que utiliza o Arduino.

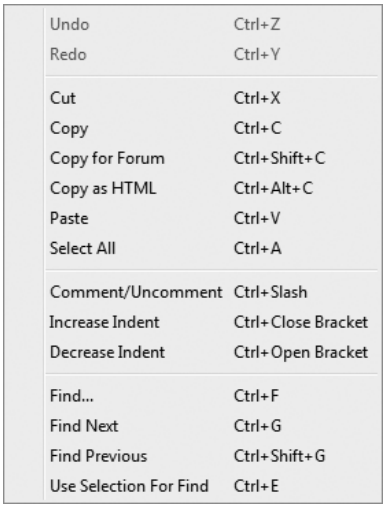

Figura 1.21 – Menu Edit.

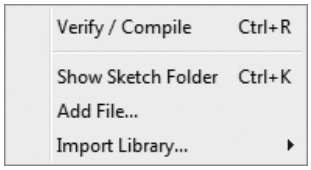

Figura 1.22 – Menu Sketch.

Por exemplo, uma das bibliotecas que você vai encontrar é a Stepper, que é um conjunto de funções que você pode utilizar dentro do seu código para controlar um motor de passo. Alguém já fez a gentileza de criar todas as funções necessárias para controlar um motor de passo e, ao incluir a biblioteca Stepper no seu sketch, podemos utilizar essas funções para controlar o motor como desejarmos. Dessa forma, ao incluir um código comumente utilizado na sua biblioteca, você pode reutilizar esse código várias vezes em projetos distintos e também ocultar do usuário as partes complicadas do código. Mais tarde, falaremos com mais detalhes sobre a utilização das bibliotecas.

Por fim, dentro do menu **Sketch**, você encontrará a opção **Show Sketch Folder**, que abrirá a pasta onde o seu sketch foi armazenado. Além disso, há a opção **Add File**, que permitirá que você adicione outro arquivo-fonte ao seu sketch. Essa funcionalidade permite que você divida sketches maiores em arquivos menores, para depois os adicionar ao sketch principal.

O próximo menu no IDE é o **Tools** (Figura 1.23). Dentro dele, há opções para selecionar a placa e a porta serial que estamos utilizando, como fizemos quando

configuramos o Arduino pela primeira vez. Temos também a função Auto Format, que formata o seu código para melhorar a sua visualização.

Figura 1.23 – Menu Tools.

A opção Copy for Forum copiará o código da janela de sketch, mas em um formato que, quando colado no fórum do Arduino (ou na maioria dos outros fóruns sobre o assunto), seja exibido da mesma maneira que aparece no IDE, acompanhado da colorização da sintaxe etc.

A opção Archive Sketch permitirá que você comprima o seu sketch em um arquivo ZIP e, em seguida, perguntará onde você deseja armazená-lo. A opção Fix Encoding & Reload converte o código criado em versões mais antigas do IDE para o formato mais recente.

O botão Programmer permitirá que você selecione um programador, caso esteja utilizando um dispositivo externo para fazer o upload do código para o seu Arduino ou se deseja gravar o código em um chip no seu próprio projeto. Simplesmente, utilizaremos o cabo USB que adquirimos com o nosso Arduino.

Por fim, a opção Burn Bootloader pode ser utilizada para gravar o Arduino Bootloader (trecho de código do chip para torná-lo compatível com o IDE do Arduino) no chip. Essa opção apenas pode ser utilizada se você tiver um programador AVR e tiver substituído o chip em seu Arduino, ou se tiver comprado chips em branco para utilizar em seu próprio projeto embarcado. A menos que você pretenda gravar muitos chips, em geral é mais fácil e mais barato simplesmente comprar um chip ATmega (Figura 1.24) com o Arduino Bootloader pré-programado. Muitas lojas online oferecem chips pré-programados que podem ser adquiridos a preços bem razoáveis. O chip utilizado no Arduino Uno é um Atmel ATmega328.

O último menu é o Help, onde você pode encontrar os menus de ajuda para descobrir mais informações sobre o IDE ou links para as páginas de referência do site do Arduino e outras páginas úteis.

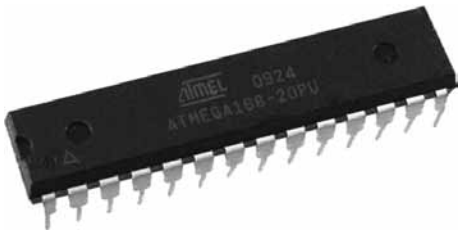

Figura 1.24 – Chip Atmel ATmega, o coração do seu Arduino (imagem cortesia da Earthshine Electronics).

O IDE do Arduino é muito simples. Você aprenderá a utilizá-lo com rapidez e facilidade à medida que avançarmos nos projetos. Conforme se tornar mais proficiente no uso dele e na programação em C (a linguagem de programação que utilizamos para criar código no Arduino), poderá considerar o IDE do Arduino básico demais e talvez deseje utilizar algo com mais funcionalidades. Sem dúvida, muitos programadores especialistas em Arduino, de fato, não utilizam esse IDE e, em vez disso, usam programas de IDE profissionais (alguns são gratuitos), como Eclipse, ArduIDE, GNU/Emacs, AVR-GCC, AVR Studio e até mesmo o XCode da Apple.

Então, agora que você tem o software do seu Arduino instalado, a placa conectada e funcionando e o conhecimento básico de como utilizar o IDE, vamos avançar direto para o projeto 1 – o LED piscante.

Resumo

Neste capítulo, você aprendeu o que é um Arduino, um pouco sobre as diversas variantes dele, o que você pode fazer com ele e quais os componentes básicos que compõem a placa Arduino. Depois, aprendeu a instalar e configurar o software e os drivers do Arduino, a selecionar a porta serial correta e a fazer o upload de um sketch de teste para o seu Arduino para se certificar de que tudo está funcionando corretamente.

Logo depois, abordamos o IDE: como o utilizar e qual a finalidade de cada um dos botões e menus, incluindo a janela Serial Monitor. Esses são os conceitos básicos necessários para entender como configurar o software para que este funcione com o hardware do Arduino. No próximo capítulo, vamos colocar esses conceitos em prática com a utilização do IDE para escrever o nosso código e fazer o upload dele para a nossa placa Arduino.