



CURSO TÉCNICO DE INFORMÁTICA

DISCIPLINA: AUTOMAÇÃO

AULA 01/04/2020

» O que é o Arduino?

O termo *Arduino* é usado para descrever tanto a placa física de Arduino (cujo tipo mais popular é o Arduino Uno) como o sistema Arduino no seu todo. O sistema também inclui o software que deve ser executado no seu computador (com o objetivo de programar a placa) e os shields periféricos que são acoplados à placa de Arduino.

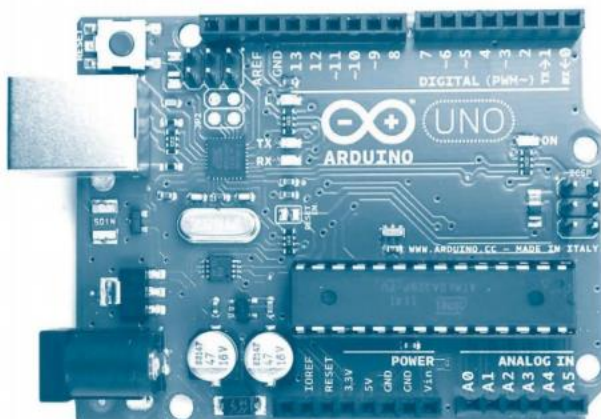
Para usar um Arduino, você também necessita de um computador adequado, que pode ser Mac, PC Windows, PC Linux, ou mesmo algo tão simples quanto um Raspberry Pi. A principal razão da necessidade de um computador é podermos baixar programas para a placa de Arduino. Uma vez instalados nela, esses programas podem ser executados de forma independente.

A Figura 1.1 mostra um Arduino Uno.

O Arduino também pode se comunicar com seu computador através de uma conexão USB. Enquanto o computador estiver conectado, você poderá enviar mensagens em ambos os sentidos. A Figura 1.2 mostra as relações existentes entre o Arduino e seu computador.

Um Arduino é diferente de um computador convencional porque, além de ter muito pouca memória, não contém sistema operacional nem interfaces para teclado, mouse ou monitor. O seu propósito é controlar coisas fazendo interfaces com sensores e atuadores. Assim, por exemplo, você pode ligar um sensor para medir temperatura e um relé para ligar e desligar um aquecedor.

A Figura 1.3 mostra algumas das coisas que você pode ligar a uma placa de Arduino. Além desses, sem dúvida, há muitos outros tipos de dispositivos que você pode conectar à ela.



Um Arduino Uno.

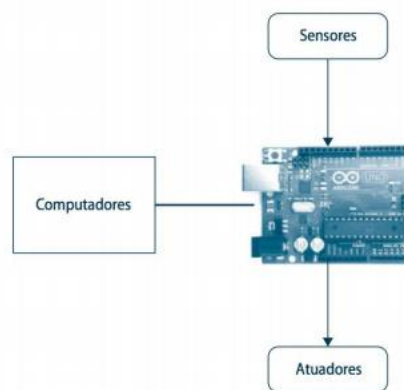


Figura 1.2 O Arduino e seu computador.

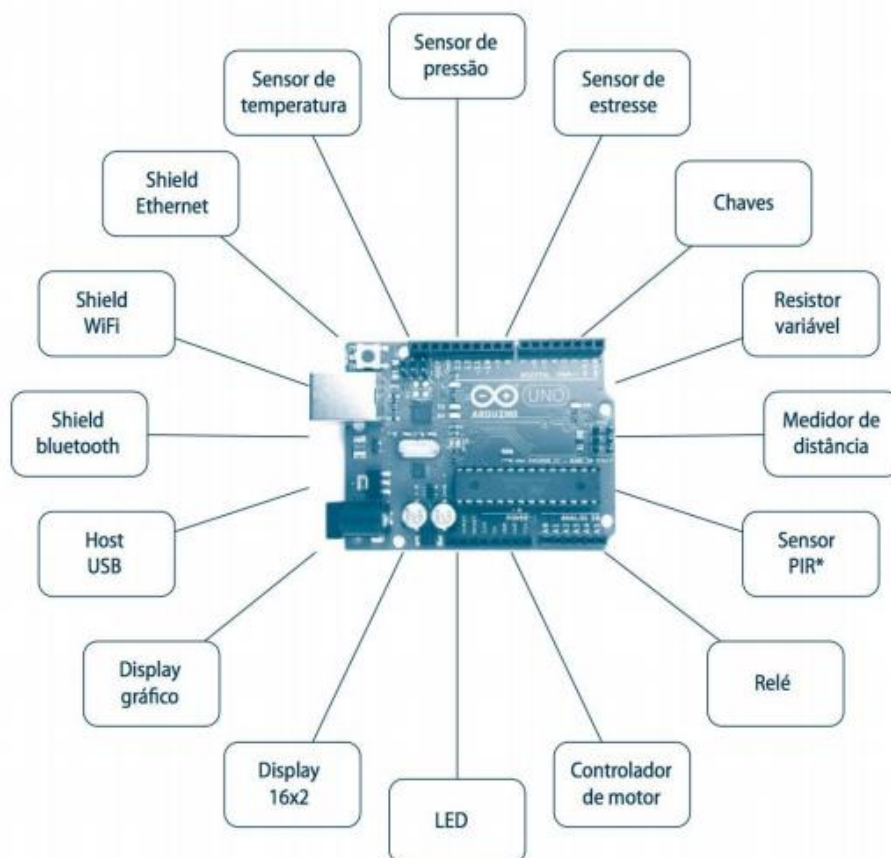


Figura 1.3 Interfaces com um Arduino.

» Instalação e o IDE

O software utilizado para programar o Arduino é denominado *Arduino Integrated Development Environment (IDE)*, que significa ambiente de desenvolvimento integrado de Arduino. Se você for um desenvolvedor de software acostumado a usar IDEs complexos, como Eclipse ou Visual Studio, achará o IDE do Arduino muito simples – e possivelmente buscará encontrar integração de repositórios, preenchimento automático de linha de comando e coisas semelhantes. Se você é iniciante em programação, gostará da simplicidade e facilidade de utilização do Arduino.

>> Instalando o IDE

O primeiro passo é fazer download do software para o seu tipo de computador por meio do site oficial do Arduino: <http://arduino.cc/en/Main/Software>.

Depois de baixar o software, você poderá encontrar instruções detalhadas de instalação para cada plataforma em <http://arduino.cc/en/Guide/HomePage>.

Uma das coisas boas a respeito do Arduino é que, para começar, precisamos apenas de um Arduino, um computador e um cabo USB para conectar os dois. Mesmo a energia elétrica para o Arduino pode ser obtida por meio do cabo USB ligado ao computador.

>> Blink

Para comprovar se o Arduino está funcionando, vamos programá-lo para fazer piscar (*blink*) o LED denominado *L* na placa do Arduino, conhecido simplesmente como LED “L”.

Comece executando o IDE do Arduino no seu computador. Então, a partir do menu File (Arquivo), escolha Examples | 01 Basics | Blink (Figura 1.4).

Em uma tentativa de fazer a programação do Arduino parecer menos assustadora para os não programadores, os programas do Arduino são referidos como *sketches*.

Antes de poder enviar o sketch Blink ao seu Arduino, você precisa dizer ao IDE que tipo de Arduino está usando. O tipo mais comum é o Arduino Uno, e, neste capítulo, estou supondo que é o que você tem. Assim, a partir do menu Tools | Board (Ferramentas | Placa), escolha Arduino Uno (Figura 1.5).

Além de escolher o tipo de placa, você também precisa selecionar a porta à qual a placa será conectada. No Windows, isso costuma ser fácil. Em geral, é a COM4, e provavelmente será a única porta mostrada na lista (veja a Figura 1.6). Entretanto, em um computador Mac ou Linux, geralmente há mais dispositivos seriais listados. O IDE do Arduino mostra primeiro os dispositivos conectados mais recentemente. Assim, a sua placa de Arduino deverá estar no topo da lista.

Para fazer a transferência (upload) do sketch para a placa do Arduino, clique no botão Upload da barra de ferramentas. É o segundo botão da barra, realçado na Figura 1.7.



Figura 1.4 O IDE do Arduino fazendo o carregamento do sketch Blink.

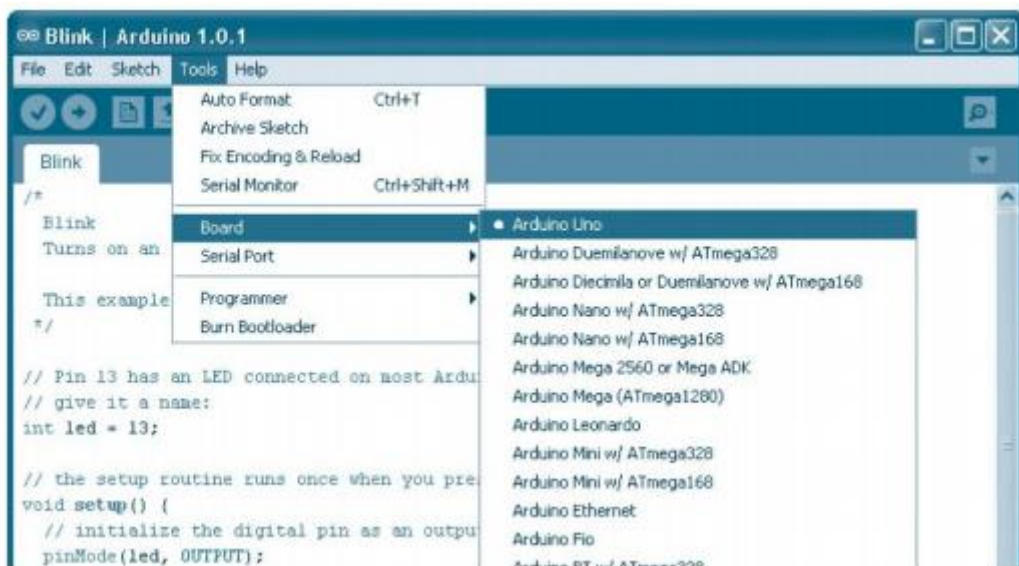


Figura 1.5 Escolhendo o tipo de placa.



Figura 1.6 Selecionando a porta serial.

Logo que você clicar no botão Upload, algumas coisas deverão acontecer. Primeiro, uma barra indicando o progresso da operação aparecerá enquanto o IDE do Arduino compila o sketch (isso significa que ele está convertendo o sketch em uma forma adequada para a transferência). A seguir, no Arduino, os LEDs rotulados como *Rx* e *Tx* deverão piscar por um breve tempo. Finalmente, O LED *L* deve começar a piscar. O IDE exibirá também uma mensagem como "Binary sketch size: 1,084 bytes (of a 32,256 byte maximum)".* Isso significa que o sketch utilizou cerca de 1Kb dos 32Kb de memória flash que estão disponíveis para programas no Arduino.

Antes de começar a programar, vamos examinar o hardware que executará seus programas (sketches) e os recursos que deverão estar disponíveis para seu uso.

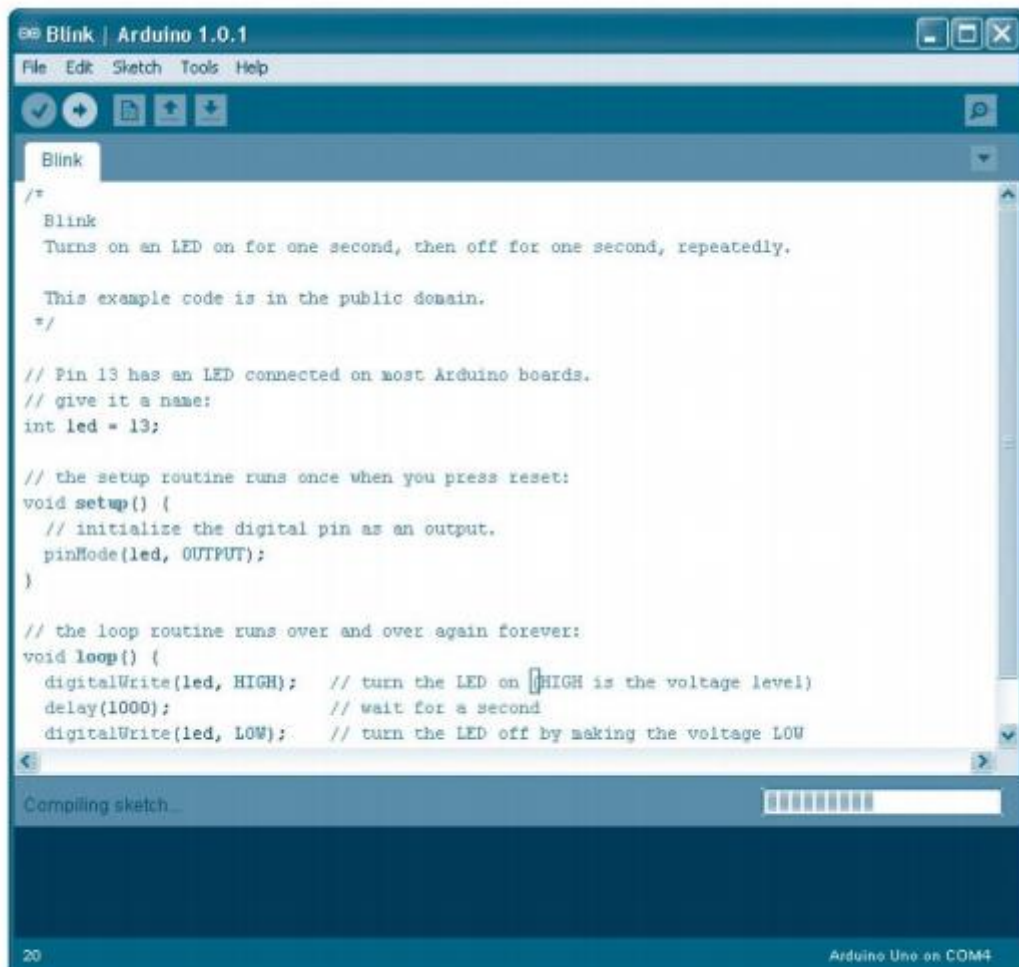


Figura 1.7 Transferindo o sketch Blink.

» Um passeio pelo Arduino

A Figura 1.8 mostra a anatomia de uma placa de Arduino. Começando em cima, próximo do conector USB no canto esquerdo superior, está a chave de Reset (Inicialização). Clicando nela, um pulso lógico é enviado ao pino de Reset do microcontrolador. Isso inicializa a memória do microcontrolador de forma que ele possa começar a execução do programa desde o início. Observe que, no Arduino, qualquer programa armazenado é mantido, porque foi instalado em uma memória flash *não volátil* – isto é, memória que preserva os dados mesmo quando o dispositivo não está energizado.

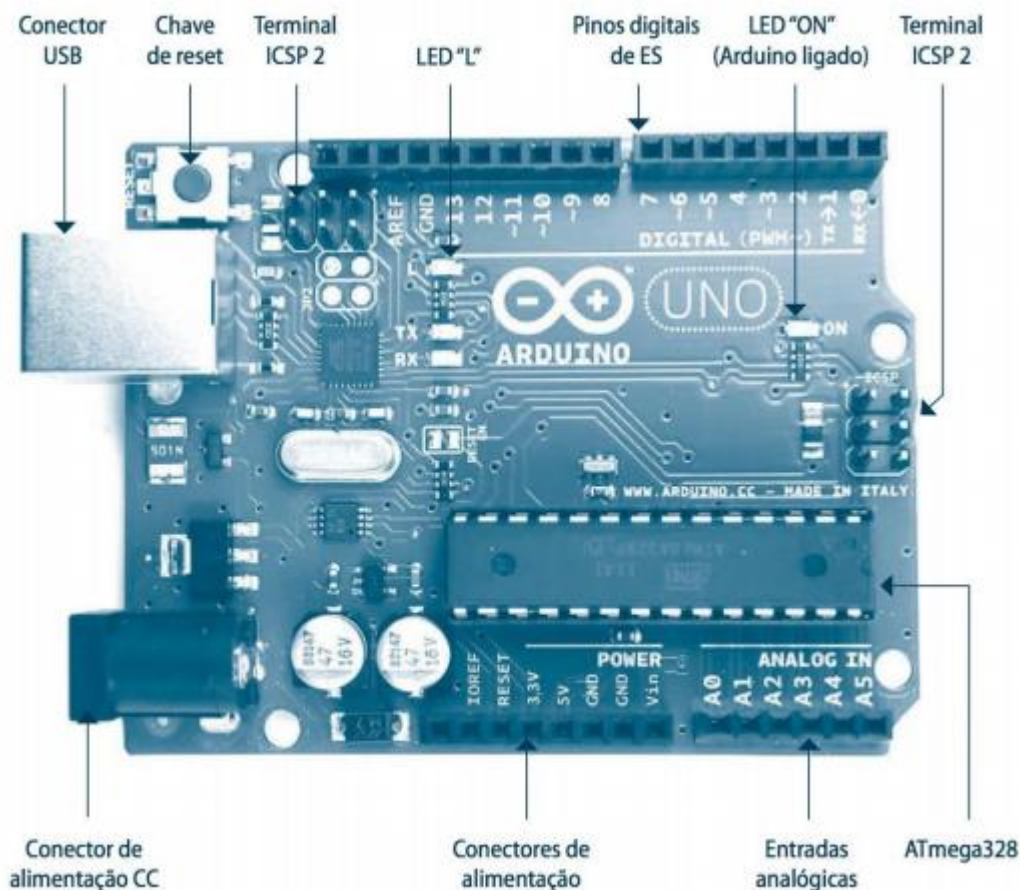


Figura 1.8 Anatomia de uma placa de Arduino.

» Fonte de alimentação

A alimentação elétrica do Arduino pode ser feita tanto através do conector USB como por meio do conector de alimentação CC (corrente contínua) que está mais abaixo. Quando a alimentação é feita através de baterias ou de um adaptador de tensão CC, qualquer valor entre 7,5 e 12V CC pode ser fornecido através do conector de alimentação CC. O Arduino sozinho consome aproximadamente 50 mA. Assim, uma pequena bateria de 9V do tipo PP3 (200mAh) poderá alimentá-lo durante cerca de 40 horas.

Quando o Arduino está ligado, o LED "ON" (à direita no Uno e à esquerda no Leonardo) permanece aceso.

» Conexões elétricas

A seguir, examinaremos os conectores (pinos-fêmeas) na parte inferior da Figura 1.8. Excluindo o primeiro pino, você pode ler os respectivos nomes impressos na sua proximidade.

O primeiro pino, sem identificação, está reservado para uso futuro. O próximo pino, *IOREF*, indica a tensão na qual o Arduino está operando. O Uno e o Leonardo trabalham com 5V. Assim, esse pino estará sempre em 5V, mas você não o usará para nada neste livro. O seu propósito é permitir que shields conectados a Arduinos de 3V, como o Arduino Due, detectem a tensão com a qual o Arduino está operando.

O próximo pino é Reset (Inicialização). Ele tem a mesma função que o botão Reset do Arduino. Como na inicialização de um computador PC, esse conector inicializa o microcontrolador para que ele comece a execução de seu programa desde o início. O conector Reset permite que você inicialize o microcontrolador mantendo esse pino momentaneamente em nível baixo (conectando-o a GND). É muito improvável que você tenha que fazer isso, mas é bom saber que o pino está à disposição.

Os demais pinos fornecem tensões diferentes (3,3, 5, GND e 9), como está indicado. *GND*, ou *ground* (terra), significa simplesmente zero volts. É a tensão de referência à qual todas as demais tensões da placa são relativas.

Os dois pinos GND são idênticos. É útil dispor de mais de um pino GND para ligar coisas. Na parte de cima da placa há outro pino GND.

» Entradas analógicas

A próxima seção de pinos é denominada *Analog In* (Entrada Analógica), indo de 0 a 5. Esses seis pinos podem ser usados para medir os valores das tensões que estão sendo aplicadas neles. Por sua vez, esses valores podem ser utilizados em um sketch. Embora esses pinos sejam de natureza analógica, podem funcionar também como entradas ou saídas digitais. Por default, eles serão considerados entradas analógicas.

» Conexões digitais

Agora vamos passar para os pinos de cima, começando pelo lado direito (Figura 1.8). Temos pinos denominados Digital, indo de 0 a 13. Eles podem ser usados como entradas ou saídas (E/S). Quando usados como saídas, podem ser controlados a partir de um sketch. Se você ligá-los a partir do seu sketch, eles assumirão a tensão de 5V. Se você desligá-los, eles ficarão com 0V. Como nos pinos de alimentação, você deve ser cuidadoso para não ultrapassar a capacidade máxima de corrente.

Esses conectores digitais podem fornecer 40 mA (miliamperes) com 5V – mais do que suficiente para acender um LED comum, mas insuficiente para acionar diretamente um motor elétrico.

» Placas de Arduino

O Arduino Uno (Figura 1.1) é a versão atual da placa original de Arduino. É a placa mais comum, e geralmente é o que as pessoas subentendem quando dizem que estão utilizando um Arduino.

Os outros tipos de placas de Arduino destinam-se a satisfazer necessidades especiais, como a necessidade de mais pinos de E/S (entrada e saída), desempenho mais rápido, placa menor, capacidade para ser costurado sobre tecidos, capacidade de conexão com telefones Android, capacidade para se integrar facilmente em redes sem fio e assim por diante.

Independentemente das diferenças de hardware, cada placa é programada a partir do IDE de Arduino, com variações mínimas nas características de software que podem usar. Quando você aprende a usar uma placa de Arduino, na prática isso significa que você aprendeu a usar todas elas.

Vamos examinar a linha corrente de placas oficiais de Arduino. Há outros Arduinos além dos discutidos aqui, mas em geral tendem a não ser tão populares. Para uma lista completa de placas, acesse o site oficial do Arduino (www.arduino.cc).

» Uno e similar

A Uno R3 é a mais recente de uma série de placas “padrão” que inclui a Uno, a Duemilanove, a Diecimila e a NG. Todas essas placas usam os microprocessadores ATmega168 ou ATmega328, que são muito semelhantes. A diferença entre eles é a quantidade de memória.

Outro Arduino corrente, com o mesmo tamanho e conexões que o Uno R3, é o Arduino Leonardo (Figura 1.9). Como você pode ver, a placa é menos povoada do que a do Uno. Isso acontece porque ela usa um processador diferente. O Leonardo usa o ATmega32u4, que é similar ao ATmega328, mas contém uma interface USB embutida, removendo a necessidade dos componentes extras encontrados no Uno. Além disso, o Leonardo contém um pouco mais de memória, mais entradas analógicas e outros recursos. Ele também é mais barato do que o Uno. Sob diversos aspectos, o seu *design* também é melhor.

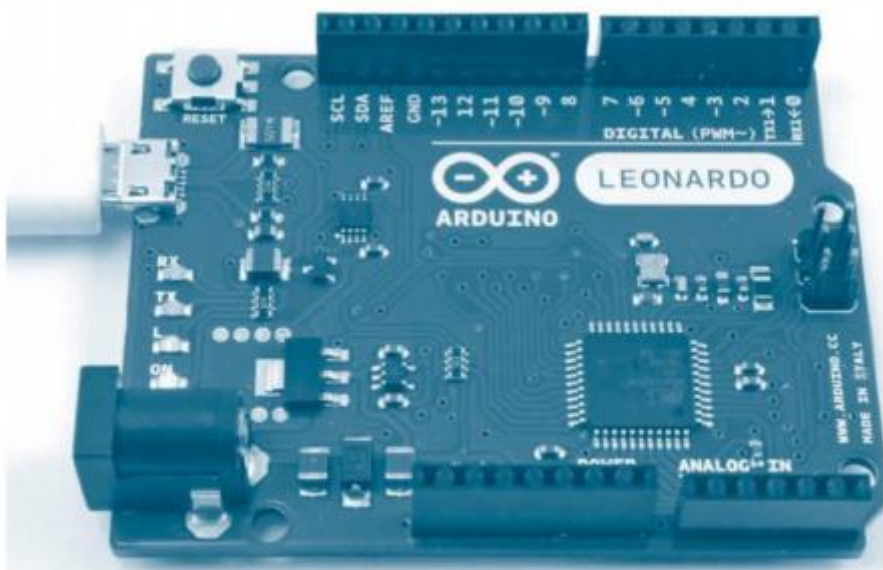


Figura 1.9 O Arduino Leonardo.

Você deve estar se perguntando por que a placa Leonardo não é a placa de Arduino mais popular. A razão é que as melhorias oferecidas pela placa Leonardo se dão ao preço de torná-la ligeiramente incompatível com as placas Uno e suas antecessoras. Alguns shields de expansão (especialmente os mais antigos) não funcionarão no Leonardo. Com o tempo, essas diferenças deixarão de ser um problema e, quando chegarmos lá, será interessante ver se o Leonardo e seus sucessores se tornaram as placas mais populares.

O Arduino Ethernet é um que foi incluído recentemente na família de placas de Arduino. Ele combina as características básicas do Uno com uma interface Ethernet, permitindo que você o conecte a uma rede sem necessidade de incluir um shield Ethernet.

» Placas grandes de Arduino

Algumas vezes, um Uno (ou um Leonardo) não contém pinos de entrada e saída em quantidade suficiente para ser utilizado na aplicação que você está desenvolvendo. Surge então a possibilidade de escolher entre uma expansão de hardware para o Uno ou optar por uma placa maior.

Além de terem os mesmos pinos que um Uno, os Arduinos de tamanho maior têm, na lateral, uma dupla fila de pinos de entrada e saída e, nas bordas de cima e de baixo, diversos pinos extras (Figura 1.10).

Tradicionalmente, uma placa “maior” seria um Arduino Mega 2560. Como todas as placas maiores de Arduino, elas contêm quantidades maiores de cada tipo de memória. Ambos o Mega 2560 e o Mega ADK usam processadores com capacidades similares à do Arduino Uno. Entretanto, considerando tudo, o Arduino Due é

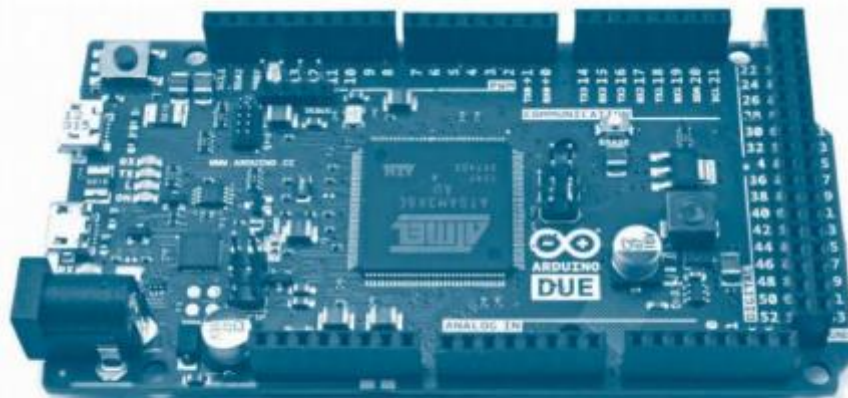


Figura 1.10 O Arduino Due.

o que tem a maior capacidade. Isso vem na forma de um processador de 84 MHz (comparado com os 16 Mhz de um Uno), mas à custa de problemas de compatibilidade. O maior desses problemas é que o Due opera com 3,3V em vez dos 5V da maioria dos Arduinos anteriores. Sem chegar a ser surpreendente, isso significa que muitos shields de Arduino são incompatíveis com o Due.

Entretanto, para projetos mais exigentes, essa placa tem muitas vantagens:

- Muita memória para programação e dados
- Capacidade para gerar música em hardware (conversores digitais-analógicos em hardware)
- Quatro portas seriais
- Duas portas USB
- Interfaces USB host e OTG
- Emulação USB de teclado e mouse

>> Placas pequenas de Arduino

Assim como o Uno é pequeno demais para alguns projetos, ele também pode ser grande demais para outros. Embora as placas de Arduino sejam baratas, os custos poderão ficar elevados se você começar a instalar placas de forma permanente em cada projeto que fizer. Encontram-se à disposição placas de Arduino menores e “profissionais”, que foram projetadas para serem fisicamente menores do que um Uno comum ou para baixar custos. Dessa forma, elas não contêm os recursos que não são necessários na maioria dos projetos.

A Figura 1.11 mostra uma placa de Arduino Mini. Essa placa não tem interface USB. No lugar dela, você precisa de um módulo separado para programar as placas. Assim como a placa Mini, há também Nanos e Micros que já têm USBs embutidas, mas custando mais.

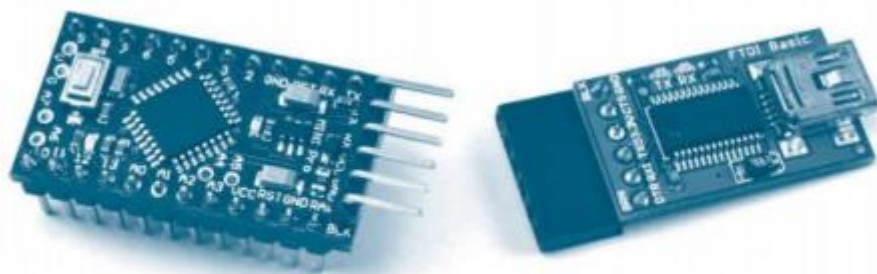


Figura 1.11 Um Arduino Mini e um programador.

>> Placas LilyPad e LilyPad USB

Dois dos Arduinos mais interessantes são o LilyPad (Figura 1.12) e o LilyPad USB, lançado recentemente. As suas placas foram projetadas para serem costuradas em vestimentas usando fios de linha condutora de eletricidade. Encontram-se à disposição também módulos LilyPad similares – para LEDs, chaves, acelerômetros, etc. As placas mais antigas de LilyPad requerem uma interface USB separada, a mesma necessária para o Arduino Mini. Entretanto, essas placas estão sendo gradualmente substituídas pela placa Arduino LilyPad USB, que já vem com um conector USB embutido.

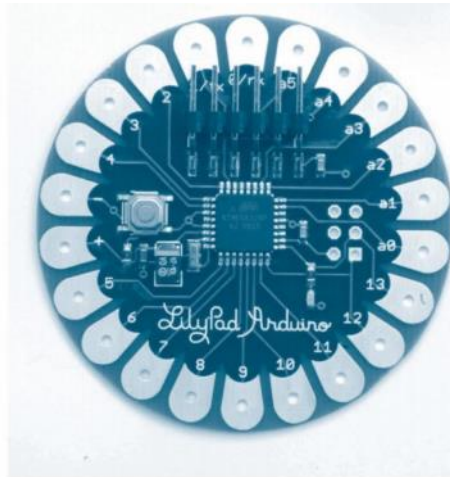


Figura 1.12 Um Arduino LilyPad.

>> Arduinos não oficiais

Assim como dispomos da placa "oficial", há também versões não oficiais e variações no hardware do Arduino, tendo em vista a sua condição de projeto aberto. Os clones fiéis do Arduino são fáceis de serem encontrados no eBay e em outros sites similares. São simplesmente cópias dos arduinos oficiais. O que há de interessante nesses clones é o preço mais em conta. Há também algumas versões interessantes baseadas em Arduino que oferecem características extras.

Dois exemplos que merecem ser examinados desse tipo de placa são

- **EtherTen** Uma placa de Arduino do tipo Ethernet (www.freetronics.com/products/etherten)
- **Leostick** Uma placa do tipo Leonardo, de pouca espessura com conector USB embutido (www.freetronics.com/collections/arduino/products/leostick)

Agora que você tem mais informação sobre o hardware de um Arduino, podemos voltar à sua programação.