

Arduino - Primeiros Passos

Por [Fábio Souza](#) - 06/11/2013



ÍNDICE DE CONTEÚDO

1. Introdução ao Arduino
2. Plataforma de desenvolvimento Arduino
3. Hardware do Arduino
4. Software do Arduino
5. IDE do Arduino
6. "Hello World" - Blink
7. Analisando o Código

Arduino é uma plataforma de código aberto (hardware e software) criada em 2005 pelo italiano Massimo Banzi (e outros [colaboradores](#)) para auxiliar no ensino de eletrônica para estudantes de design e artistas. O objetivo principal foi o de criar uma plataforma de baixo custo, para que os estudantes pudessem desenvolver seus protótipos com o menor custo possível. Outro ponto interessante do projeto, foi a proposta de criar uma plataforma de código aberto, disponível para a comunidade o que ajudou em muito no seu desenvolvimento.

1. Introdução ao Arduino

O [site](#) da plataforma Arduino o define como:

“O Arduino é uma plataforma de prototipagem eletrônica *open-source* que se baseia em hardware e software flexíveis e fáceis de usar. É destinado a artistas, designers, hobbistas e qualquer pessoa interessada em criar objetos ou ambientes interativos.

O Arduino pode *sentir* o estado do ambiente que o cerca por meio da recepção de sinais de sensores e pode interagir com os seus arredores, controlando luzes, motores e outros atuadores. O microcontrolador na placa é programado com a linguagem de programação Arduino, baseada na linguagem [Wiring](#), e o ambiente de desenvolvimento Arduino, baseado no ambiente [Processing](#). Os projetos desenvolvidos com o Arduino podem ser *autônomos* ou podem comunicar-se com um computador para a realização da tarefa, com uso de *software* específico (ex: Flash, Processing, MaxMSP).”

2. Plataforma de desenvolvimento Arduino

O Arduino é formado por dois componentes principais: *Hardware* e *Software*.

O *hardware* é composto por uma placa de prototipagem na qual são construídos os projetos.

O *software* é uma IDE, que é executado em um computador onde é feita a programação, conhecida como *sketch*, na qual será feita *upload* para a placa de prototipagem Arduino, através de uma comunicação serial. O *sketch* feito pelo projetista dirá à placa o que deve ser executado durante o seu funcionamento.

3. Hardware do Arduino

Existem [diversas](#) placas oficiais de Arduino e muitas outras não oficiais. Vamos abordar a placa Arduino Uno nesse artigo. A seguir é exibida a placa [Arduino Uno](#) REV3:



Figura 1 - Placa Arduino UNO

Conforme visto na imagem acima a placa Arduino UNO possui diversos conectores que servem para interface com o mundo externo. Vejamos como estão organizados os pinos na placa:

- 14 pinos de entrada e saída digital (pinos 0-13):
 - Esses pinos podem ser utilizados como entradas ou saídas digitais de acordo com a necessidade do projeto e conforme foi definido no *sketch* criado na IDE.
- 6 pinos de entradas analógicas (pinos A0 - A5):
 - Esses pinos são dedicados a receber valores analógicos, por exemplo, a tensão de um sensor. O valor a ser lido deve estar na faixa de 0 a 5 V onde serão convertidos para valores entre 0 e 1023.
- 6 pinos de saídas analógicas (pinos 3, 5, 6, 9, 10 e 11):
 - São pinos digitais que podem ser programados para ser utilizados como saídas analógicas, utilizando modulação PWM.

A alimentação da placa pode ser feita a partir da porta USB do computador ou através de um adaptador AC. Para o adaptador AC recomenda-se uma tensão de 9 a 12 volts.

4. *Software* do Arduino

O software para programação do Arduino é uma IDE que permite a criação de *sketches* para a placa Arduino. A linguagem de programação é modelada a partir da linguagem [Wiring](#). Quando pressionado o botão upload da IDE, o código escrito é traduzido para a linguagem C e é transmitido para o compilador `avr-gcc`, que realiza a tradução dos comandos para uma linguagem que pode ser compreendida pelo microcontrolador.

A IDE apresenta um alto grau de abstração, possibilitando o uso de um microcontrolador sem que o usuário conheça o mesmo, nem como deve ser usado os registradores internos de trabalho.

A IDE do Arduino possui uma linguagem própria baseada na linguagem C e C++.

O Ciclo de programação do Arduino pode ser dividido da seguinte maneira:

1. Conexão da placa a uma porta USB do computador;
2. Desenvolvimento de um *sketch* com comandos para a placa;
3. Upload do *sketch* para a placa, utilizando a comunicação USB.
4. Aguardar a reinicialização, após ocorrerá à execução do *sketch* criado.

A partir do momento que foi feito o *upload* o Arduino não precisa mais do computador: o Arduino executará o *sketch* criado, desde que seja ligado a uma fonte de energia.

5. IDE do Arduino

A IDE pode ser baixada gratuitamente no [site do Arduino](#), onde pode ser escolhida a melhor opção de download conforme plataforma utilizada.

Quando se abre o IDE do Arduino, será exibido algo semelhante à figura abaixo:

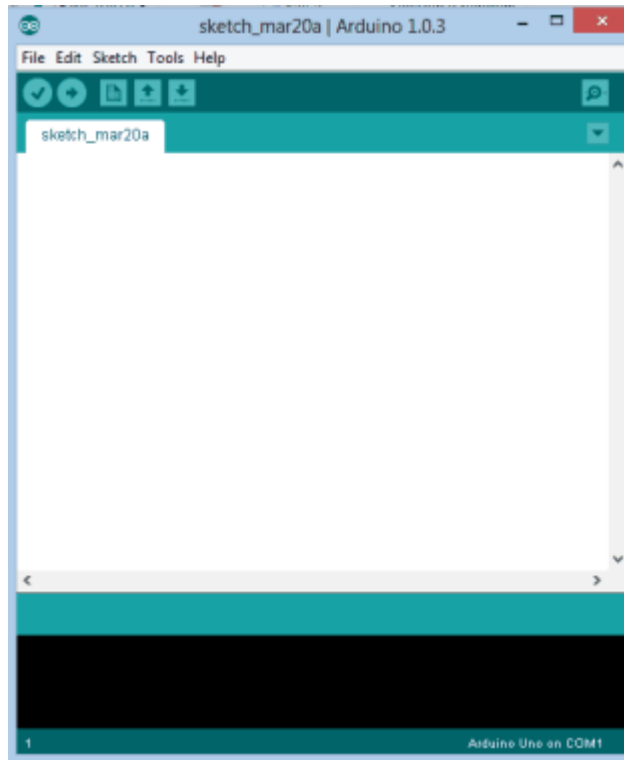


Figura 2 - IDE - Interface de Desenvolvimento Integrado do Arduino UNO

O IDE é dividido em três partes: A *Toolbar* no topo, o código ou a *Sketch Window* no centro, e a janela de mensagens na base, conforme é exibido na figura anterior.

Na *Toolbar* há uma guia, ou um conjunto de guias, com o nome do *sketch*. Ao lado direito há um botão que habilita o serial monitor. No topo há uma barra de menus, com os itens File, Edit, Sketch, Tools e Help. Os botões na *Toolbar* fornecem acesso rápido às funções mais utilizadas dentro desses menus.

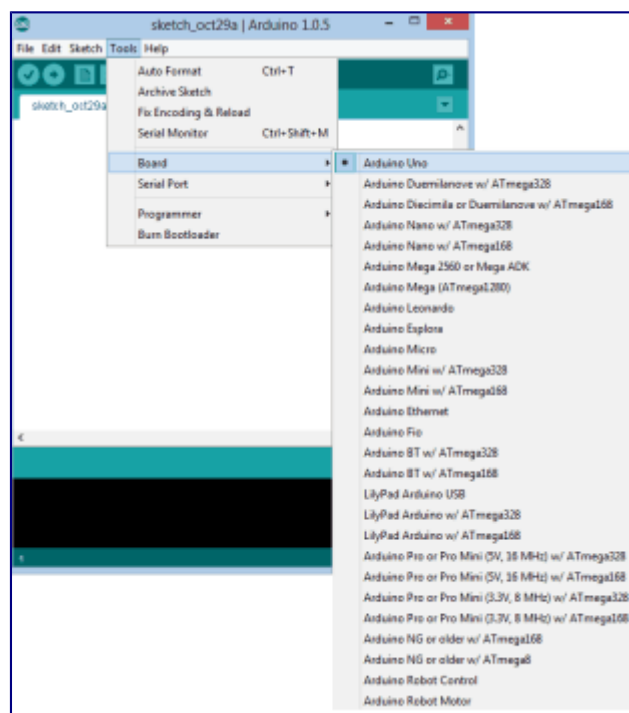
Abaixo são identificados os ícones de atalho da IDE:

- **Verify**
 - Verifica se existe erro no código digitado.

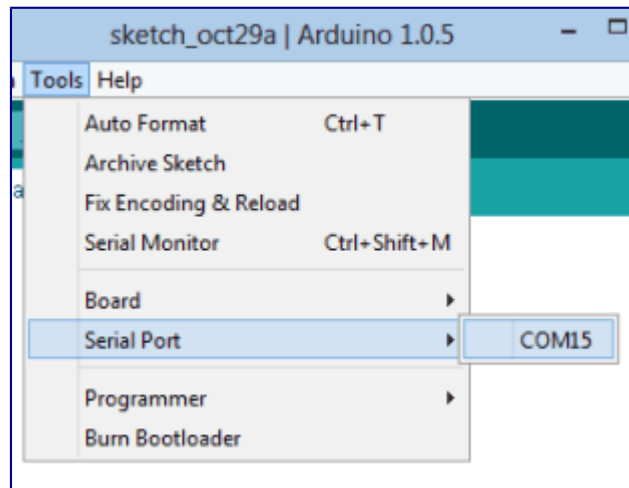
- **Upload**
 - Compila o código e grava na placa Arduino se corretamente conectada;
- **New**
 - Cria um novo *sketch* em branco.
- **Open**
 - Abre um *sketch*, presente no sketchbook.
- **Save**
 - Salva o *sketch* ativo
- **Seria monitor**
 - Abre o monitor serial.

Os demais comandos presentes na barra de menus podem ser consultados através do menu <help><Environment>.

Após a conexão do Arduino ao computador, é atribuído a placa uma COM. A primeira vez que o programa Arduino for executado deve-se selecionar o modelo de placa utilizado, no nosso caso escolheremos Arduino Uno, conforme figura abaixo:



Após a definição do modelo, deve-se selecionar em qual COM a placa foi atribuída:

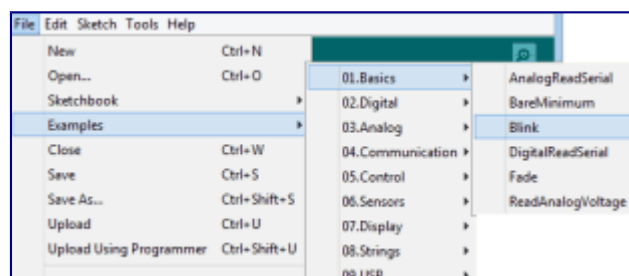


Após estas configurações o ambiente está preparado para uso e pode-se testar qualquer um dos exemplos que acompanham a IDE ou até mesmo com um novo sketch.

6. "Hello World" - Blink

O exemplo mais simples para iniciar a programação do Arduino, que pode ser considerado como o conhecido "Hello World" das linguagens de programação, consiste em acionar um LED através de uma saída digital.

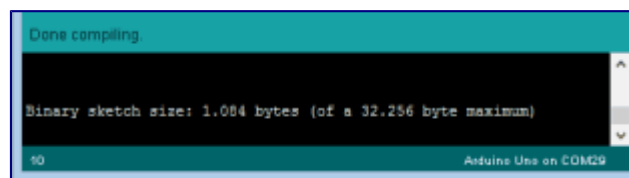
A placa Arduino Uno já possui um Led ligado ao pino digital 13 que pode ser utilizado para o teste, e na IDE podemos carregar o exemplo *Blink*:



Blink

```
1  /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4
5  This example code is in the public domain.
6  */
7
8  // Pin 13 has an LED connected on most Arduino boards.
9  // give it a name:
10 int led = 13;
11
12 // the setup routine runs once when you press reset:
13 void setup() {
14 // initialize the digital pin as an output.
15 pinMode(led, OUTPUT);
16 }
17
18 // the loop routine runs over and over again forever:
19 void loop() {
20 digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
21 delay(1000); // wait for a second
22 digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
23 delay(1000); // wait for a second
24 }
```

Para verificar se o código está correto deve-se clicar no ícone **verify**, após a compilação é exibida uma mensagem de *status* da operação e caso esteja tudo certo será exibida a quantidade de bytes gerados pelo programa:



Para gravar o código na memória flash do microcontrolador é necessário clicar no ícone **Upload**, será transferido o código para a placa e após alguns segundos o LED ligado ao pino 13 começará a piscar em intervalos de 1 segundo.

7. Analisando o Código

O código do exemplo Blink é relativamente simples, porém apresenta a estrutura básica de um programa desenvolvido na IDE Arduino. Inicialmente nota-se que existem duas funções obrigatórias em um programa Arduino, `setup()` e `loop()`.

A função `setup ()` é executada na inicialização do programa e é responsável pelas configurações iniciais do microcontrolador, tal como definição dos pinos de I/O, inicialização da comunicação serial, entre outras.

A função `loop ()` será onde ocorrerá o laço infinito da programação, ou seja, onde será inserido o código que será executado continuamente pelo microcontrolador.

Dentro do *loop* principal está o código que fará o led ligado pino 13 piscar em intervalos de 1 segundo.

A função `digitalWrite(led, HIGH);` coloca o pino em nível lógico 1, ligando o led.

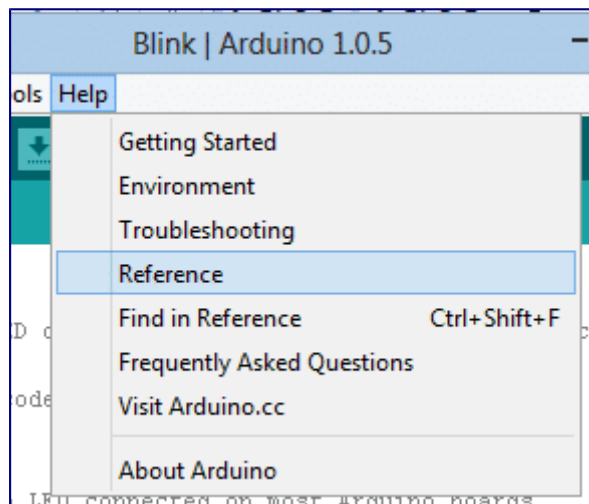
A função `delay(1000);` aguarda o tempo de 1000 ms, ou seja, 1 segundo para que possa ser executada a próxima instrução.

A função `digitalWrite(led, LOW);` coloca o pino em nível lógico 0, desligando o led.

E novamente é esperado 1 segundo com a função `delay();`

O loop é repetido infinitamente enquanto a placa estiver ligada.

A referência da linguagem Arduino pode ser acessada através do menu `<help>`:



Testar outros exemplos básicos, é um bom começo para aprender mais sobre a plataforma Arduino.