

# Projetos Eletrônicos

1º Bimestre

## **Introdução ao Arduino**

Março 2016

# Placas de Arduino



## Arduino Mega

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

## Detalhes

- **Entrada e Saída:**

Cada um dos 54 pinos digitais do Mega2560 pode ser usado como entrada ou saída, usando as funções de `pinMode()`, `digitalWrite()`, e `digitalRead()`. Eles operam a 5 volts. Cada pino pode fornecer ou receber um máximo de 40 mA e possui um resistor interno (desconectado por default) de 20-50K?.

- **Comunicação:**

Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Usados para receber (RX) e transmitir (TX) dados seriais TTL.

- **Interruptores:**

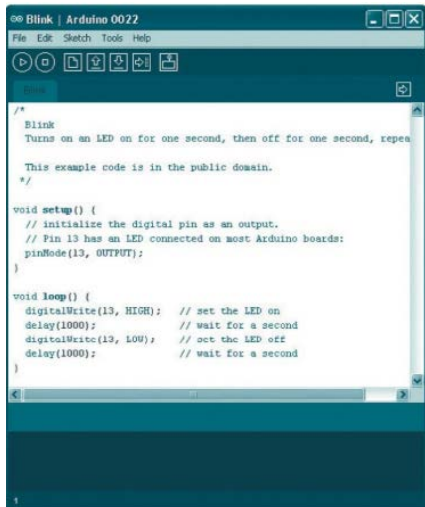
Interruptores externos: 2 (interruptor 0), 3 (interruptor 1), 18 (interruptor 5), 19 (interruptor 4), 20 (interruptor 3), and 21 (interruptor 2).

## Detalhes

- PWM:  
0 a 13. Fornecem saída analógica PWM de 8-bits com a função `analogWrite()`.
- Porta Analógica de Entrada  
16 entradas analógicas, cada uma das quais com 10 bits de resolução (i.e. 1024 valores diferentes)
- LED: 13:  
Há um LED conectado ao pino digital 13. Quando o pino está em HIGH o led se acende.

# Instalando o Software

- Instalar o software do Arduino

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 0022". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for running, saving, opening, and other functions. The main text area contains the following code:

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repea  
  
  This example code is in the public domain.  
  */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);   // set the LED on  
  delay(1000);              // wait for a second  
  digitalWrite(13, LOW);    // set the LED off  
  delay(1000);              // wait for a second  
}
```

[www.arduino.cc](http://www.arduino.cc)

# Ligando o Arduino

- Conecte o Cabo USB e verifica se o computador o reconhece
- Abra o programa do Arduino como administrador



- Vai em Ferramentas e depois em:
  1. Placa: Selecione Arduino Mega (se for esta sua placa)
  2. Porta: Selecione a porta que o computador reconheceu.

# Hello Word

- Vamos executar o primeiro programa. Para isto:
  1. Arquivo > Exemplos > 01.Basics > Blink
  2. Clique em carregar.



The screenshot shows the Arduino IDE interface. At the top, the title bar reads "Blink | Arduino 1.6.7". Below it is a menu bar with "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". A toolbar contains several icons, with a red circle highlighting the "Upload" icon (a right-pointing arrow). Below the toolbar, the text "Blink \$" is visible. The main workspace contains the following code:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);          // wait for a second
}
```

## Resultado:

Led do pino digital 13 irá acender e apagar

# Funcionamento do Código no Arduino

## Setup e Loop

```
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
}
```



# Exercício 1

1. Execute **setup** e faça o pino 13 ser uma saída.
2. Execute **loop** e faça o pino 13 passar para nível HIGH, ou alto (LED acende).
3. Execute um retardo (delay) de meio segundo.
4. Faça o pino 13 passar para nível LOW, ou baixo (LED apaga).
5. Execute novamente **loop**, voltando para o passo 2 e fazendo o nível do pino 13 ser alto, ou HIGH (LED acende).

## Exercício 1

# Resultado:

```
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
}
```

Neste programa o Led vai aparentar estar sempre aceso, pois quando ele apaga a primeira coisa que acontece logo em seguida é que ele é novamente aceso. Isso ocorre muito rapidamente.

## Exercício 1 - Concertado

```
// sketch 03-01
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
}
```

Solução



## Utilizando Variáveis

```
// sketch 03-02
int ledPin = 13;
int delayPeriod = 500;

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

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);
}
```

- Para piscar mais rápido:  
Simplesmente mude delayPeriod = 100;

## Exercícios 2

Faça um Pisca-Pisca utilizando a porta 13 do Arduino que começa a piscar muito rapidamente e aos poucos vai se tornando mais e mais lento

```
// sketch 03-03
int ledPin = 13;
int delayPeriod = 100;

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

void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(delayPeriod);
  digitalWrite(ledPin, LOW);
  delay(delayPeriod);
  delayPeriod = delayPeriod + 100;
}
```

