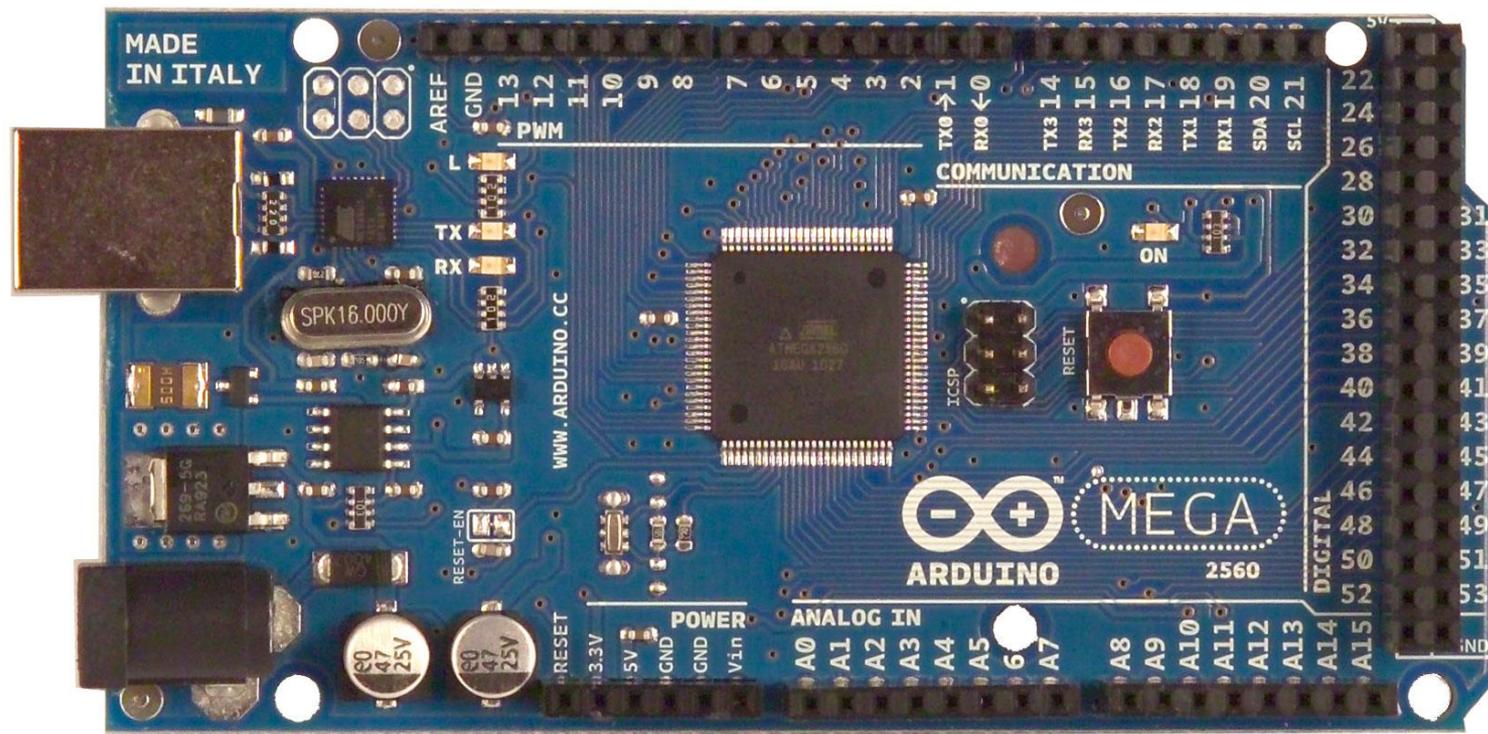


# Robótica com Arduino



# Onde comprar o Arduino?

<http://lojabrasilrobotics.blogspot.com/>

Onde comprar Arduino , Componentes , Robótica , Mecatrônica , Automação , Eletrônica

## Brasilrobotics Loja

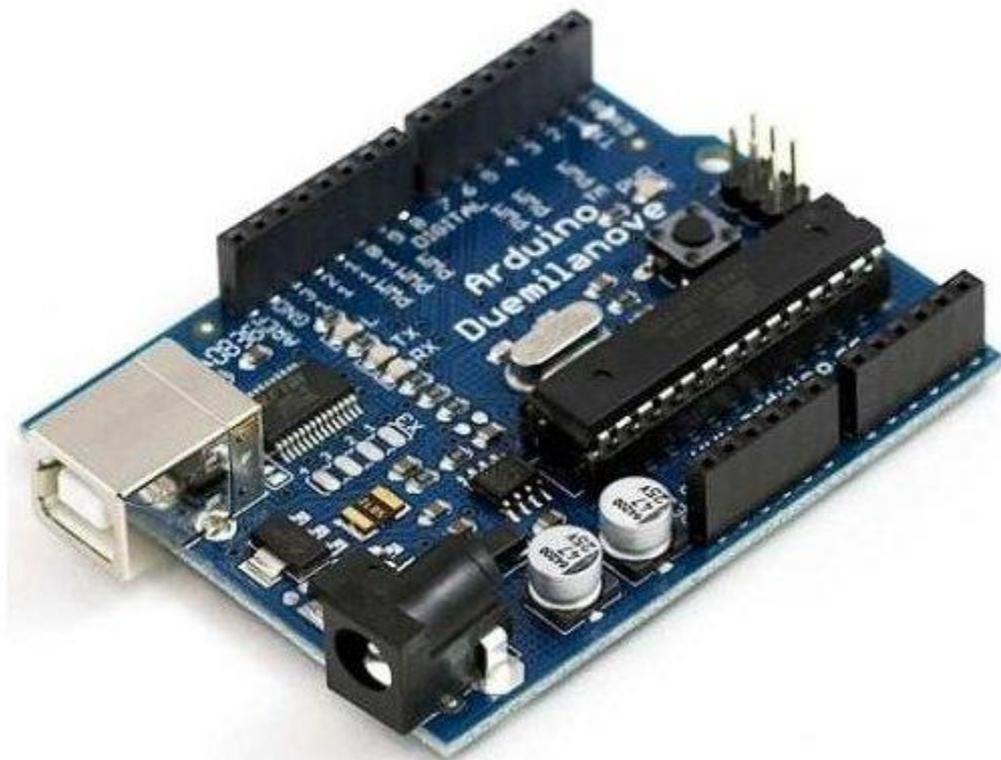


# Onde comprar o Arduino?

<http://www.mercadolivre.com.br/>



[Voltar para a lista](#) | [Eletrônicos, Áudio e Vídeo](#) > [Peças e Componentes Elétricos](#) > [Microcontroladores](#)



## Arduino Duemilanove Ebook



R\$ 99<sup>90</sup>

1 x R\$ 99<sup>90</sup> com Meic

Produto novo

São Paulo (Sao Jose Dos Campos)

34 vendidos [Ver qualificações](#)



### Reputação do vendedor



### Meios de pagamento



# Onde comprar o Arduino?

<http://www.ebay.com/>



Welcome! Sign in or register.

CATEGORIES

ELECTRONICS

FASHION

MOTORS

TICKETS

DEALS

CLASSIFIEDS

Back to search results | Computers & Networking > Other



## UNO 2011 ATMEGA328P-PU AVR MCU Board+ Arduino Cmpatible

Item condition: **New**

Quantity:  2 available

Price: **US \$18.99**

**Buy It Now**

**Add to cart**

Add to Watch list

**BillMeLater** \$10 back on 1st purchase & 6 months to pay  
Subject to credit approval. [See terms](#)

Shipping: **FREE** Standard Shipping | [See all details](#)  
[See details about international shipping here](#)

# Onde comprar o Arduino?

<http://www.sparkfun.com/>



Products	Support	Tutorials	Distributors	About Us	Contact
----------	---------	-----------	--------------	----------	---------

Home | Search Results for *arduino uno* (22 found)

**Refine Search**

arduino uno

- Products
- News
- Tutorials

sort by  showing  per page

	<p>DEV-10356 RoHS ✓ </p> <p><b>Arduino Uno SMD</b></p>	\$29.95	<input type="text" value="1"/>	<input type="button" value="A"/>
--	--	---------	--------------------------------	----------------------------------

	<p>DEV-11021 RoHS ✓ </p> <p><b>Arduino Uno - R3</b></p>	\$29.95	<input type="text" value="1"/>	<input type="button" value="A"/>
---	--	---------	--------------------------------	----------------------------------

# Arduino Nano, Duemilanove ou Mega?

Cada um tem sua aplicação. Cabe ao projetista analisar o melhor hardware para seu projeto.

Um projeto pequeno, pode se beneficiar do Arduino Nano.

Um projeto normal, sem restrição de tamanho, pode usar o Duemilanove

Em um projeto grande, com vários sensores/atuadores, talvez seja melhor trabalhar com o Mega

# Arduino Nano v3.0

Dimensões: 1.85cm largura x 4.32cm comprimento

Pinos I/O Digitais: 14 (6 com PWM)

Entradas analógicas: 8

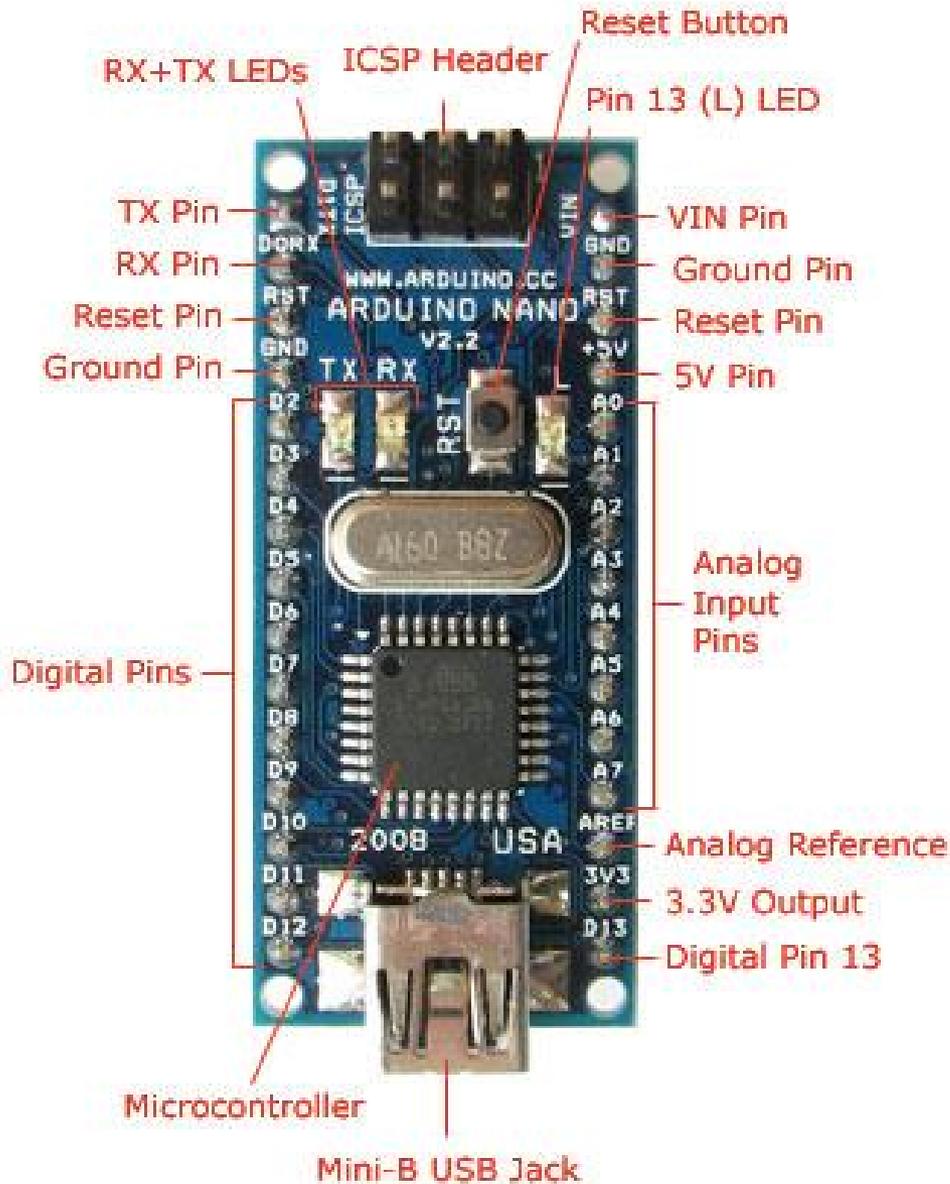
Memória Flash: 32 KB (HD)

SRAM: 2 KB

EEPROM: 512 Bytes

Interrupções Externas: 2

# Arduino Nano v3.0



# Arduino Duemilanove

Pinos I/O Digitais: 14 (6 com PWM)

Entradas analógicas: 8

Memória Flash: 32 KB (HD)

SRAM: 2 KB

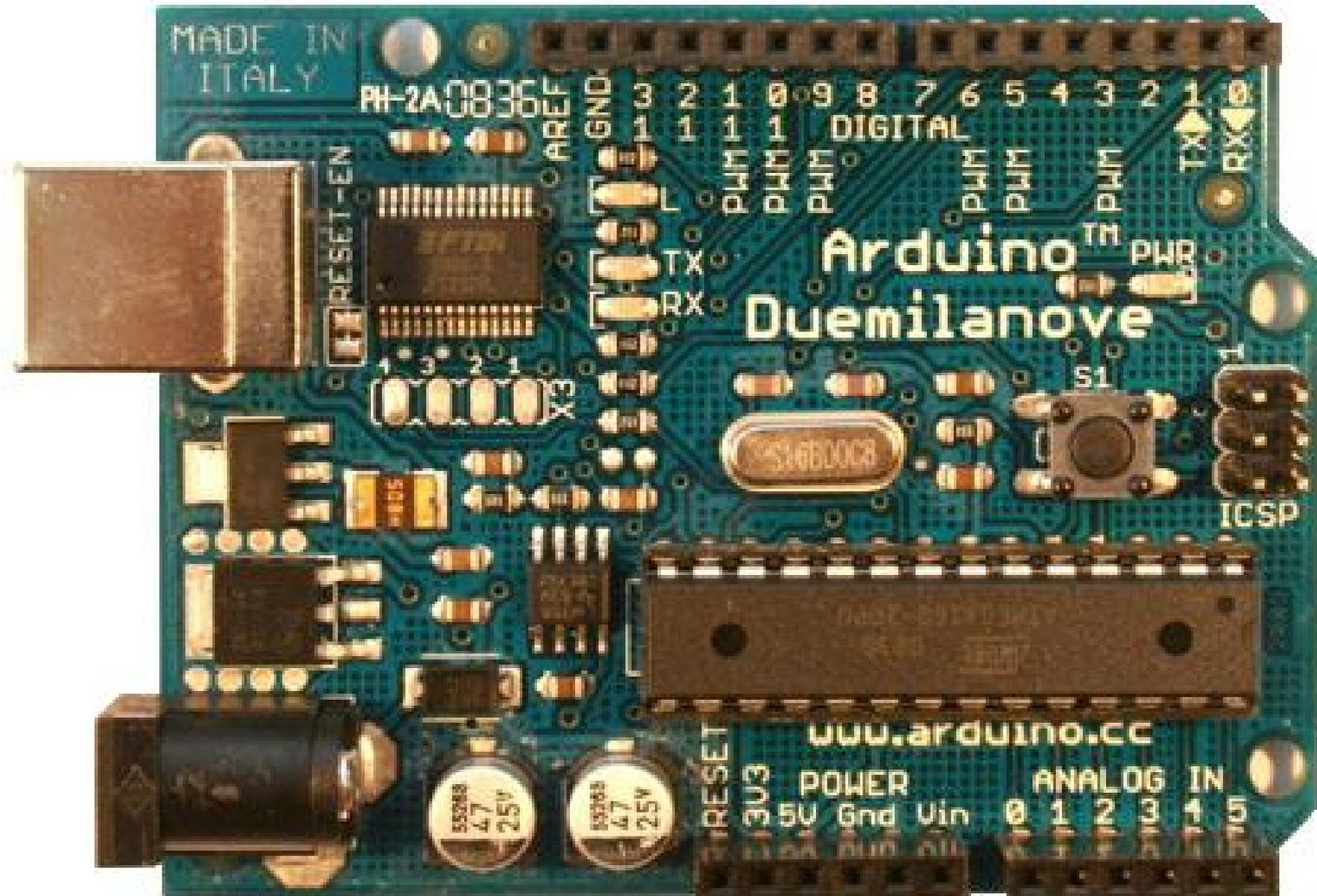
EEPROM: 512 Bytes

Processador: Atmega328

Interrupções Externas: 2

Link oficial: <http://arduino.cc/en/Main/arduinoBoardDuemilanove>

# Arduino Duemilanove



# Duemilanove x Uno

Arduino Uno é a versão de hardware mais recente.

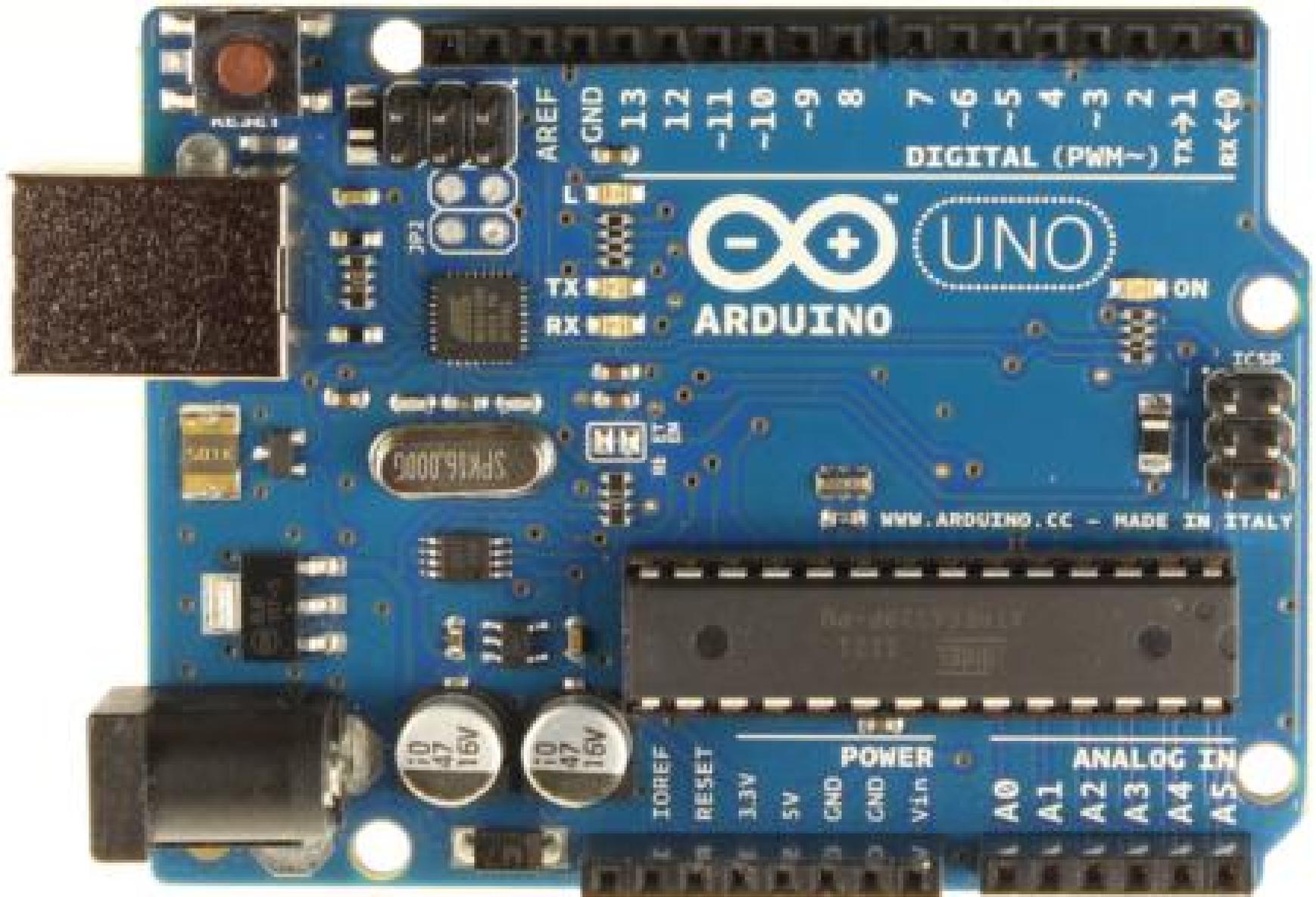
O Duemilanove (2009 em Italiano) usa o chip FT232RL para conversar com o computador.

Já o Uno usa o Atmega16u2 para a comunicação via USB.

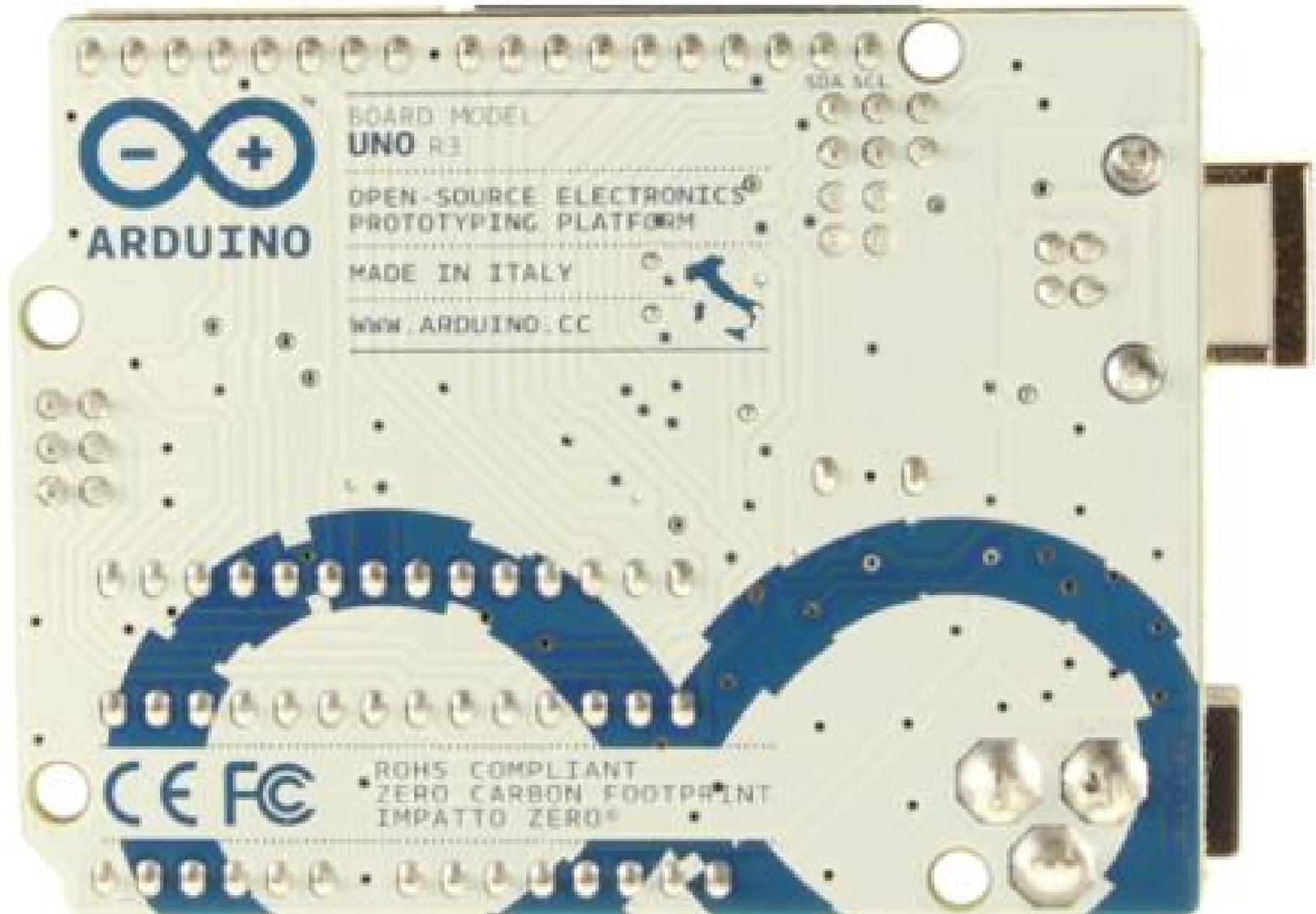
No mais são praticamente iguais

Link oficial Uno: <http://arduino.cc/en/Main/arduinoBoardUno>

# Arduino Uno



# Arduino Uno - parte de trás



# Arduino Mega 1280

Pinos I/O Digitais: 54 (14 com PWM)

Entradas analógicas: 16

Memória Flash: 128 KB (HD)

SRAM: 8 KB

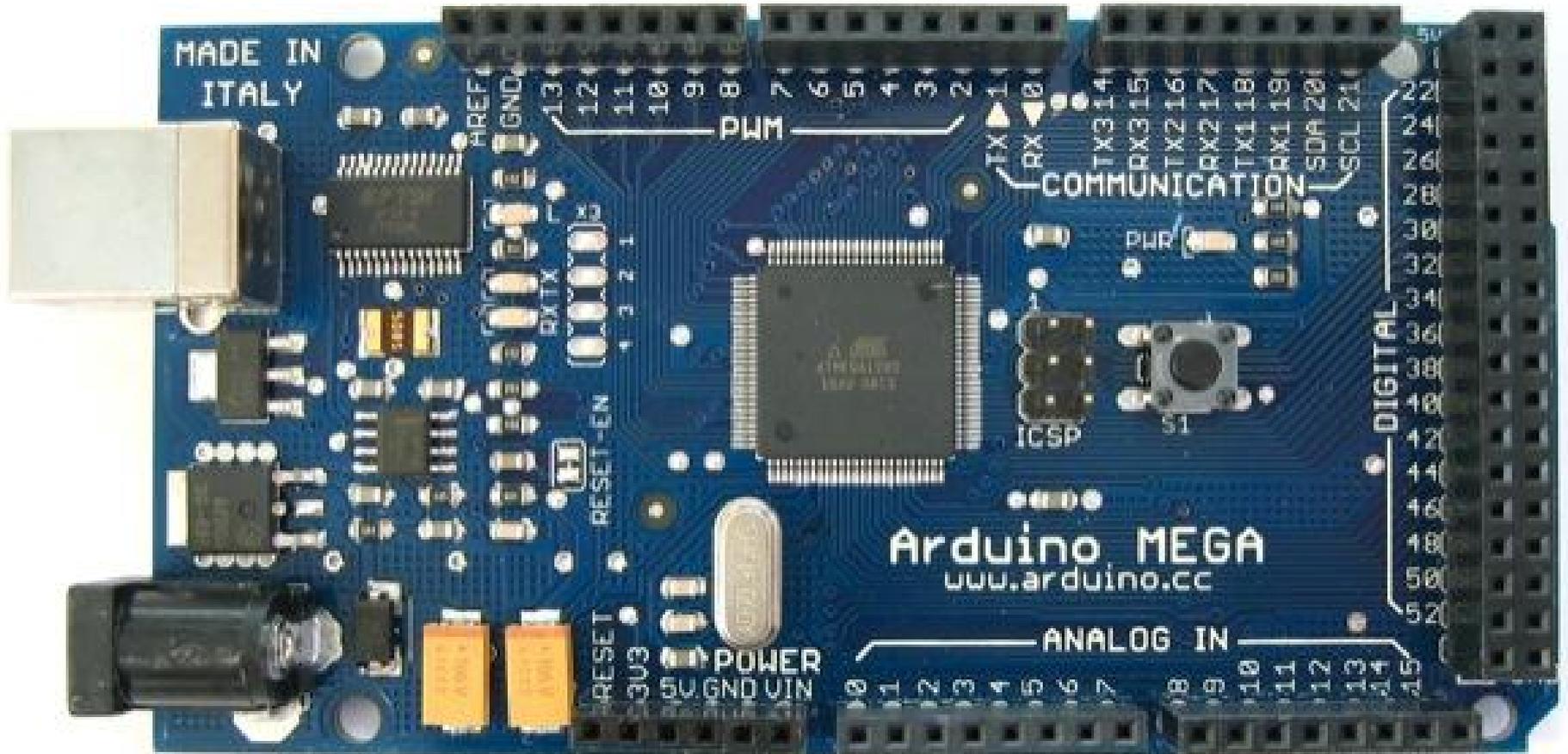
EEPROM: 4 KB

Processador: Atmega328

Interrupções Externas: 2

Link oficial: <http://arduino.cc/en/Main/arduinoBoardDuemilanove>

# Arduino Mega



# Outras Placas

Existem dezenas de placas compatíveis com Arduino.

Vão desde modelos semelhantes ao Uno, até mini placas para desenvolvimento de aplicações onde o espaço é pequeno.

Existem várias outras placas de conexão com Rede, Bluetooth, Zigbee entre outras

Para ver todas acesse o link: <http://arduino.cc/en/Main/Boards>

# LED Piscante - análise de software

// Código fonte para programa do LED Piscante

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

```
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

# Comentários na Programação

Assim como um comentarista de futebol, muitas vezes precisamos de um norte para saber o que está acontecendo quando chegamos de paraquedas em um código.

Na programação, usamos comentários para descrever o que determinado código faz

O compilador ignora tais comentários ao gerar as instruções para o Arduino

# Como comentar?

// Gera um comentário de uma linha

/\*

Inicia comentário para várias linhas.

Isso significa que você pode continuar

Escrevendo até fechar esse comentário

\*/

# Código antes de ser comentado

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

```
void loop() {  
    digitalWrite(13, HIGH);  
    delay(1000);  
    digitalWrite(13, LOW);  
    delay(1000);  
}
```

# Código com comentários

```
/*  
  Código fonte para programa do LED Piscante  
  Liga LED por 1 segundo, e desliga por 1 segundo, repetidamente  
*/  
void setup() {  
  // define o pino 13 como sendo de saída  
  pinMode(13, OUTPUT);  
}  
void loop() {  
  // Envia 5 volts para o pino 13 do Arduino  
  digitalWrite(13, HIGH);  
  // Aguarda 1 segundo (1000 ms)  
  delay(1000);  
  // Envia 0 volts para o pino 13 do Arduino  
  digitalWrite(13, LOW);  
  // Aguarda 1 segundo  
  delay(1000);  
}
```

# A função **setup()**

Quando acordamos depois de um longo sono em um horário e local não habitual, o que acontece?

- Primeiro nos damos conta de que estávamos dormindo
- Depois nos perguntamos, quem sou? Onde estou?
- Por fim, o que devo me preparar para fazer?

# A função **setup()**

Nesta função fazemos as inicializações de pinos que vamos trabalhar (definimos se são de entrada ou saída)

Nela também iniciamos uma comunicação Serial (que será vista em outra aula)

Podemos inserir qualquer código nesta função, mas devemos lembrar que será executado somente uma vez (quando o Arduino acordar)

# Exercício 1

**Recortar** o código da função `loop()` para a função `setup()` e carregar o novo programa para o Arduino.

O que aconteceu?

Testar o botão reset.

# A função `loop()`

Após você acordar, saber quem você é e qual seu propósito, o próximo passo é saber o que você irá fazer durante o dia todo.

Nosso sistema de visão precisa primeiro receber a imagem, invertê-la, transformá-la em pulsos elétricos. Por fim o cérebro a processa e a interpreta.

Esse processo ocorre 24 vezes por segundo, e poderia ser implementado na função `loop()` !

# Função `pinMode()`

Modo Pino. Essa função define como um pino do Arduino irá funcionar. As possibilidades são entrada (`INPUT`) ou saída (`OUTPUT`)

Ela recebe dois parâmetros, ou seja, duas informações.

Primeiro o número do pino que queremos definir

Depois o tipo de modo, `Entrada` ou `Saída`

# Função **digitalWrite()**

Escrita Digital. Escreve um valor digital em um determinado pino do Arduino.

Recebe dois parâmetros, sendo o primeiro o número do pino que desejamos escrever e o segundo o seu estado (**HIGH**, ou **LOW**).

Valores digitais são simples:

**HIGH** -> Coloca 5 Volts no pino definido

**LOW** -> Coloca 0 Volts no pino definido

# Função `delay()`

Atrasar. Simplesmente segura o Arduino para que não faça nada por N milissegundos.

Recebe 1 parâmetro, ou seja 1 único valor que é o tempo (em milissegundos) que o Arduino deve ficar sem fazer nada.

Exemplo para atrasar meio segundo:

```
delay(500);
```

## Exercício 2

Faça um programa que envie 5 Volts ao pino 3 Digital do Arduino.

Com um multímetro faça a medida da tensão neste pino.

Por fim, altere o programa para enviar 0 Volts ao pino 3 Digital e refaça a medida com o multímetro.

# Código fonte do Exercício 2

```
/*  
  Código fonte para programa do LED Piscante  
  Liga PINO (envia 5 Volts ao pino digital 3)  
*/  
void setup() {  
  // Define o pino 3 como sendo de saída  
  pinMode(3, OUTPUT);  
  // Envia 5 volts para o pino 3 do Arduino  
  digitalWrite(3, HIGH);  
}  
void loop() {  
  // Não é necessário nenhum comando aqui  
}
```