



FACULDADE DE TECNOLOGIA DE PORTO ALEGRE– FAQI

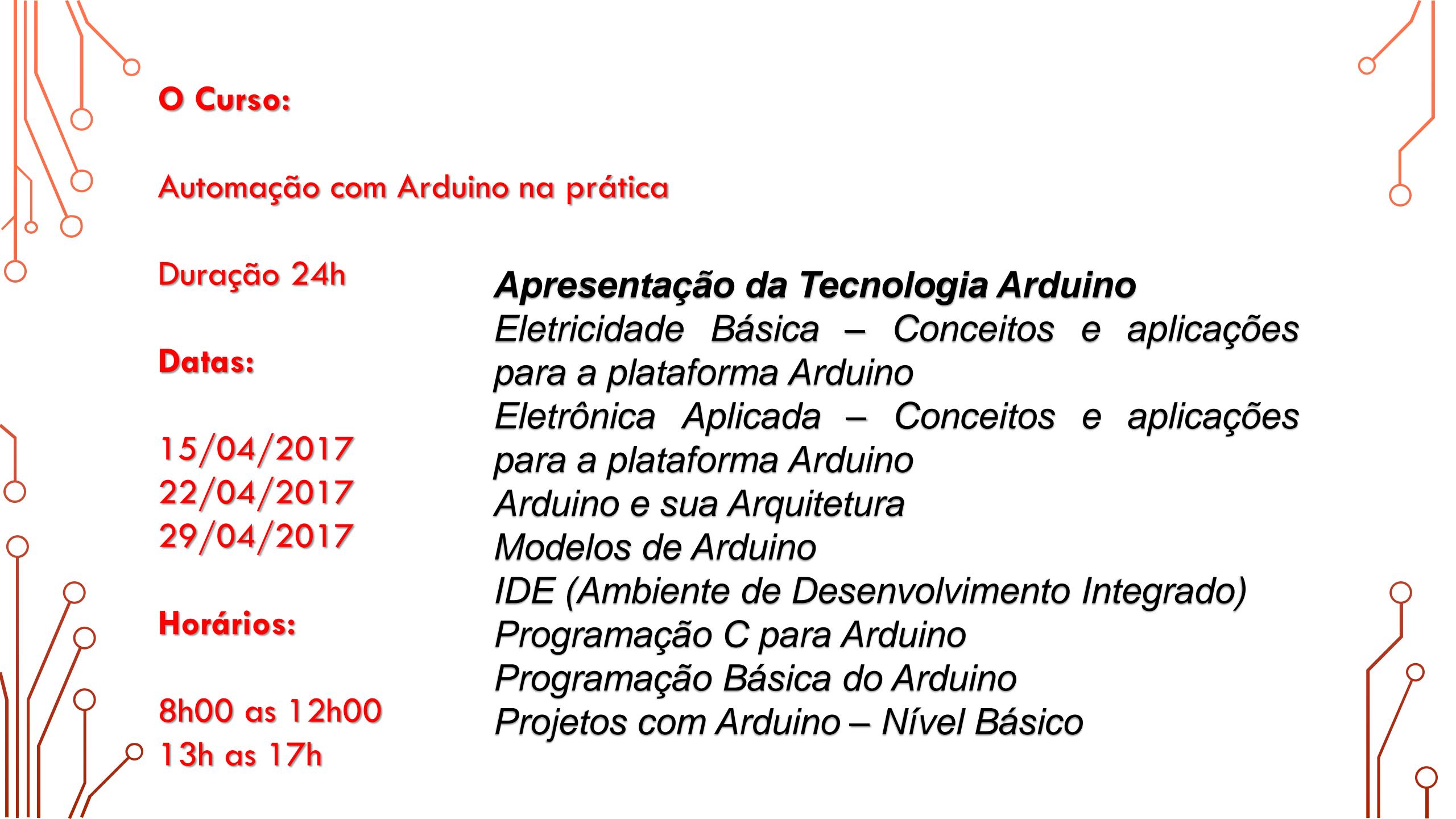


Automação com Arduino na prática



Professor

- **Silvio Cesar Viegas**
- **Doutorando no Ensino de Ciências e Matemática**
- **Mestre no Ensino de Ciências e Matemática**
- **Bacharel em Informática**
- **Email:** scviegas@gmail.com / silvio.viegas@qi.edu.br
- **Curriculum Lattes:** <http://lattes.cnpq.br/5020505141968701>



O Curso:

Automação com Arduino na prática

Duração 24h

Datas:

15/04/2017

22/04/2017

29/04/2017

Horários:

8h00 as 12h00

13h as 17h

Apresentação da Tecnologia Arduino

Eletricidade Básica – Conceitos e aplicações para a plataforma Arduino

Eletrônica Aplicada – Conceitos e aplicações para a plataforma Arduino

Arduino e sua Arquitetura

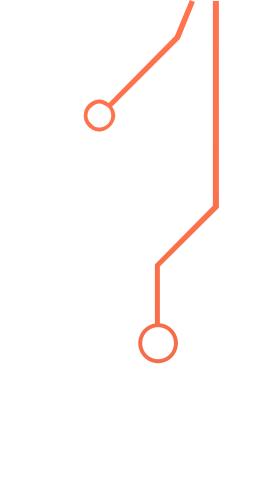
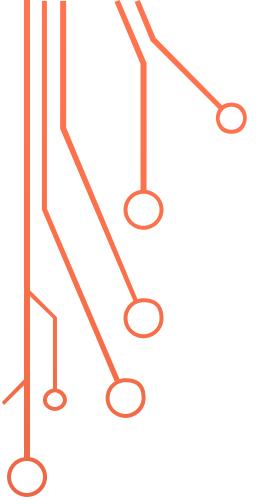
Modelos de Arduino

IDE (Ambiente de Desenvolvimento Integrado)

Programação C para Arduino

Programação Básica do Arduino

Projetos com Arduino – Nível Básico



O Professor:

Silvio Cesar Viegas

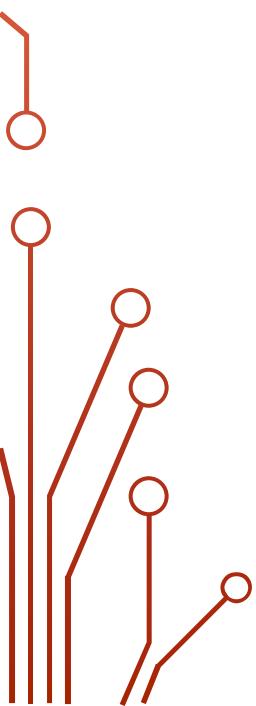
**Doutorando no Ensino de Ciências e Matemática
Pesquisando o uso de tecnologias e sistemas embarcados**

Mestre no Ensino de Ciências e Matemática

Bacharel em Informática

Contato: silvio.viegas@qi.edu.br

Material disponível em: <https://github.com/scviegas/arduino>

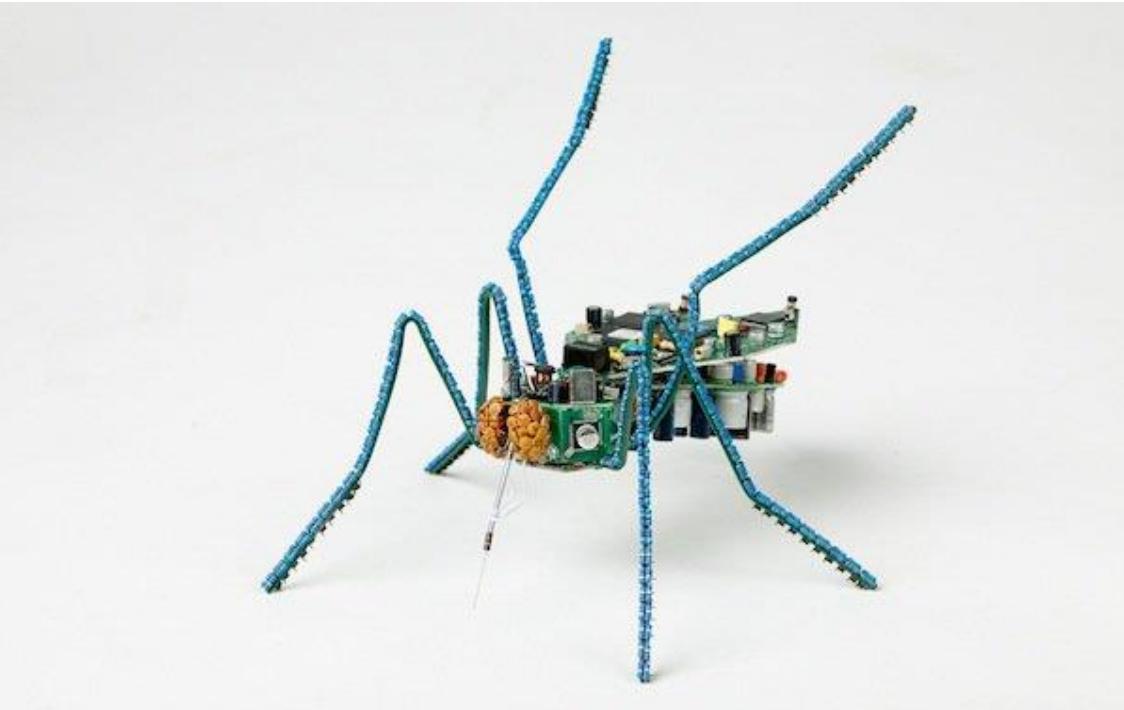




Massimo Banzi says "You don't need anybody's permission to create something great." So what you waiting for?

Eletrônica

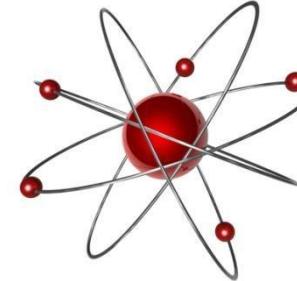
- ▶ A eletrônica pode ser definida como a ciência que estuda formas de controlar a energia elétrica em circuitos elétricos.
- ▶ É um ramo da engenharia que desenvolve soluções aplicando os princípios de eletricidade descobertos pela física.
- ▶ Usa circuitos elétricos formados por condutores elétricos e componentes eletrônicos para controlar sinais elétricos.
- ▶ A eletrônica divide-se em analógica e digital.



Corrente e Tensão

► ÁTOMO

- Composto por prótons, nêutrons e elétrons.
- Os prótons carregam cargas positivas e estão presentes no núcleo do átomo.
- Os nêutrons não carregam carga e assim como os prótons estão presentes no núcleo do átomo.
- Os elétrons carregam carga negativa e orbitam o núcleo do átomo.



Corrente e Tensão

► ÁTOMO



Corrente e Tensão

► ÁTOMO

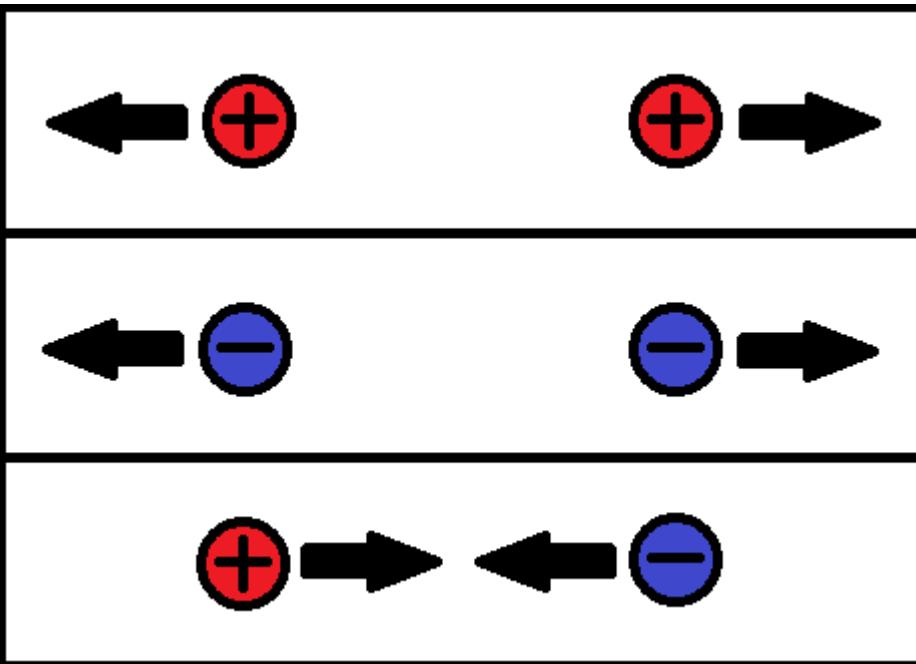
- Quando o átomo possui o mesmo número de elétrons e de prótons é considerado neutro.
- Quando o átomo possui um número maior de prótons do que de elétrons é considerado positivo.
- Quando o átomo possui um número maior de elétrons do que de prótons é considerado negativo.

Ionização é o nome dado quando o átomo ganha ou perde elétrons.

Corrente e Tensão

► Atração e Repulsão

- Corpos com cargas de sinais opostos de atraem e corpos com cargas de mesmo sinal de repelem.



Corrente e Tensão

► CORRENTE ELÉTRICA

- Os elétrons livres movimentam-se de um átomo a outro através de um meio condutor.
- Corrente elétrica é um fluxo de elétrons que circula em um condutor.
- A corrente elétrica (I) é medida em Ampère (A).
- Para os elétrons se moverem de um átomo a outro é necessário haver uma diferença de potencial ou tensão.

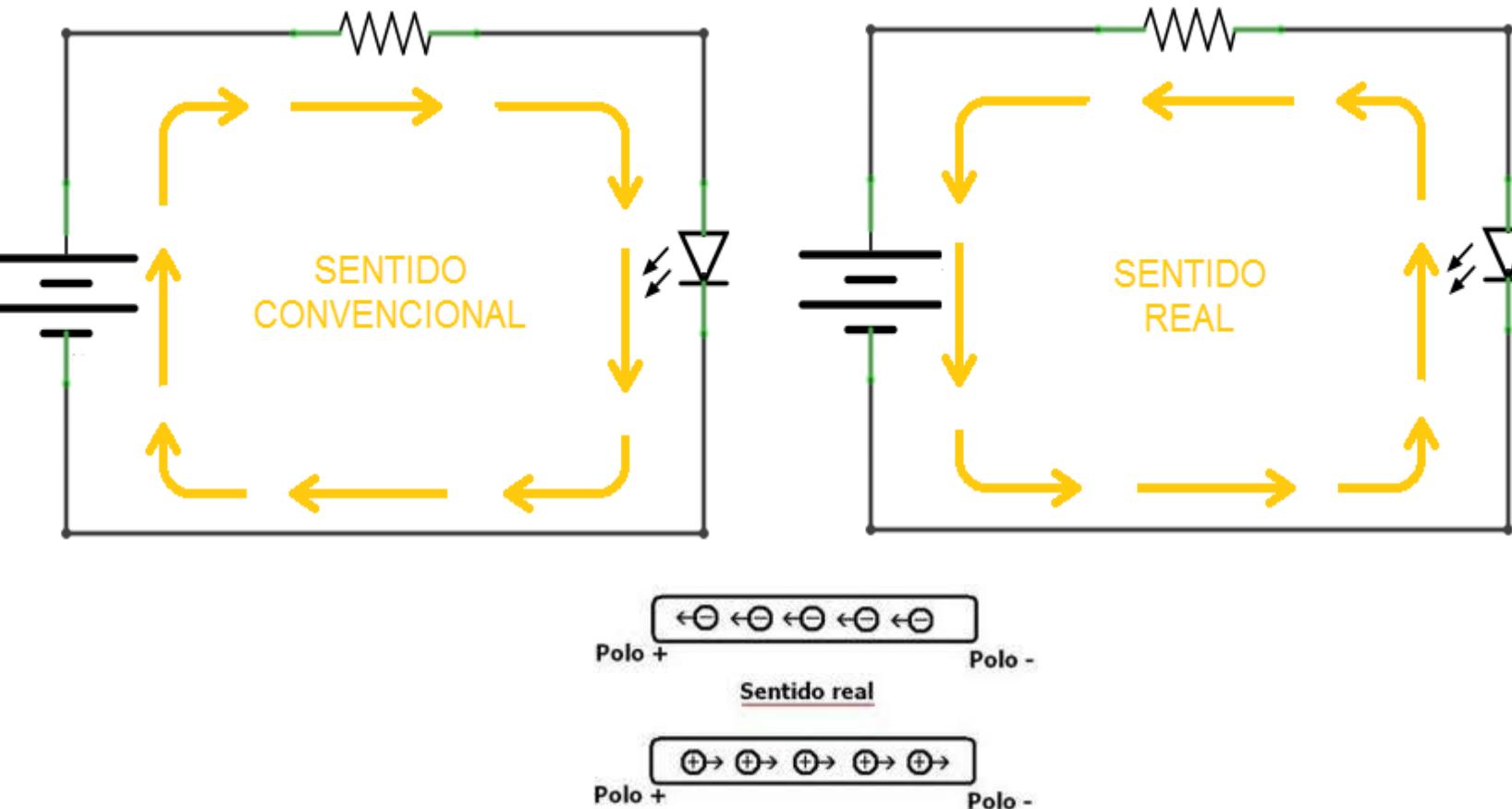
Corrente e Tensão

► SENTIDO REAL VS SENTIDO CONVENCIONAL DA CORRENTE ELÉTRICA

- Em um circuito os elétrons livres se deslocam do polo negativo para o polo positivo. Esse é o sentido real da corrente elétrica.
- Em análise de circuitos, entretanto, costuma- se considerar que os elétrons se deslocam no sentido oposto: do polo positivo para o polo negativo. Esse é o sentido convencional da corrente elétrica.

Corrente e Tensão

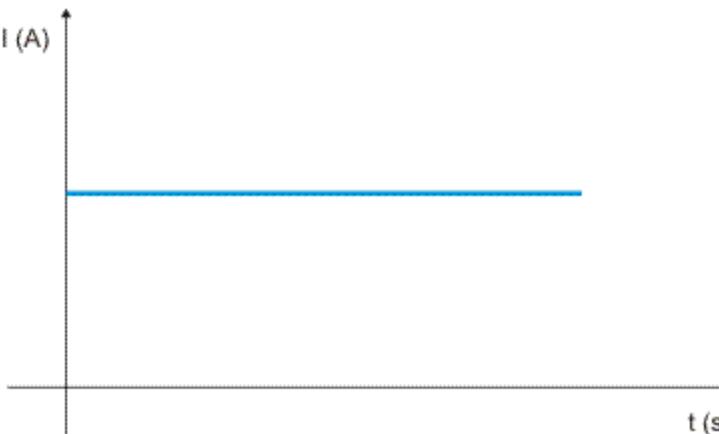
- ▶ Sentido Real vs Sentido Convencional da Corrente Elétrica



Corrente e Tensão

► TIPOS DE CORRENTES ELÉTRICAS

- Corrente contínua
 - Os elétrons se movem sempre no mesmo sentido.
 - Grande parte dos equipamentos eletrônicos trabalha com corrente contínua.

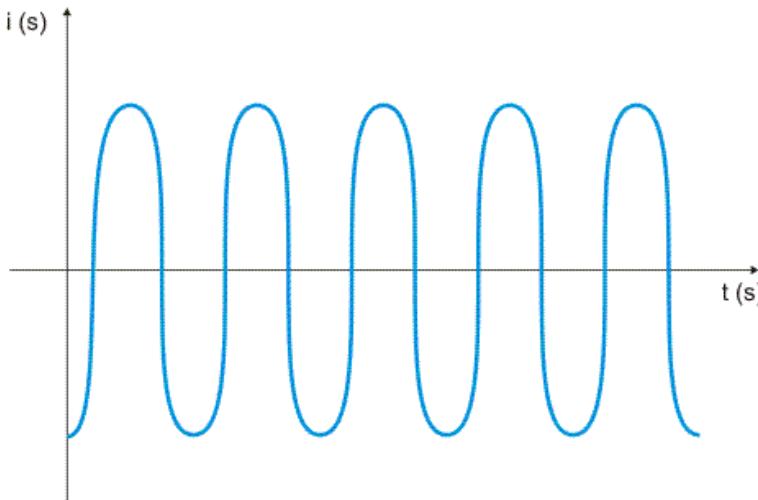


Corrente e Tensão

► TIPOS DE CORRENTES ELÉTRICAS

- Corrente alternada

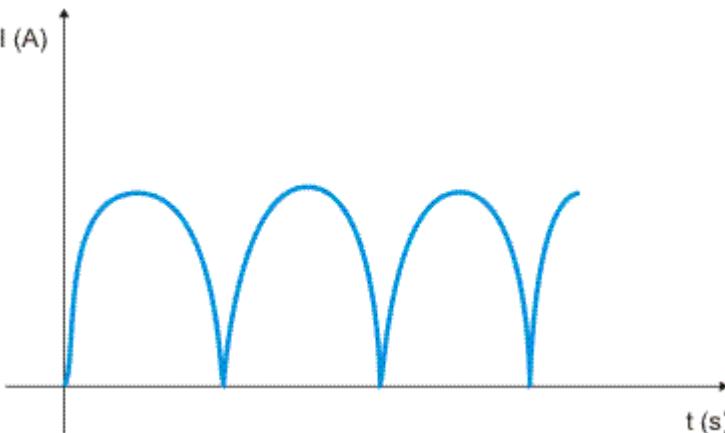
- Na corrente alternada o sentido dos elétrons é invertido periodicamente, ou seja, ora a corrente é positiva, ora é negativa.
- A energia elétrica que chega em nossas casas é do tipo corrente alternada.



Corrente e Tensão

► TIPOS DE CORRENTES ELÉTRICAS

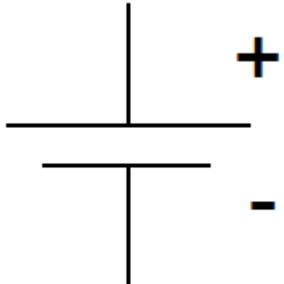
- Corrente pulsante
 - Somente alterna o valor.
 - Corrente resultante da retificação da corrente alternada.



Corrente e Tensão

► TENSÃO ELÉTRICA

- É a força responsável por impulsionar os elétrons em um condutor.
- A tensão é medida em Volts (V).
- Exemplos:
 - Bateria/pilha de 9 volts
 - Tomada de 110 ou 220 volts



Resistência Elétrica

- › Resistência elétrica é uma grandeza que indica o quanto um determinado condutor se opõe a passagem de corrente elétrica.
- › Bons condutores de eletricidade possuem um número maior de elétrons livres, por esse motivo possuem uma baixa resistência elétrica.
- › A resistência elétrica é medida em Ohms e o símbolo é a letra grega ômega – Ω .

Condutores e Isolantes

▶ CONDUTORES

- São materiais que pouco se opõem à passagem de corrente elétrica.
- Possuem baixa resistividade.
- Os elétrons da camada de valência estão fracamente ligados ao núcleo e, assim, quebram facilmente suas ligações com o átomo, tornando-se livres para compor a corrente elétrica.



Condutores e Isolantes

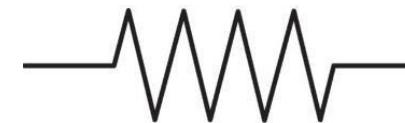
► ISOLANTES

- Fazem muita oposição à passagem de corrente elétrica.
- Possuem alta resistividade.
- Os elétrons da camada de valência estão fortemente ligados ao núcleo e, por isso, precisam de uma energia muito maior para desfazer suas ligações com o átomo. Isso resulta em poucos elétrons livres para compor a corrente elétrica.



Resistores

- ▶ O resistor é um componente eletrônico utilizado para limitar o fluxo de corrente.
- ▶ Os resistores podem ser do tipo fixo ou do tipo variável.
- ▶ Os resistores mais comuns são os de filme de carbono.



Resistores

- ▶ UM RESISTOR FIXO DE FILME DE CARBONO POSSUI EM SEU CORPO FAIXAS COLORIDAS QUE INDICAM O SEU VALOR DE RESISTÊNCIA.



- ▶ Onde:
 - A primeira faixa indica o primeiro número.
 - A segunda faixa indica o segundo número.
 - A terceira faixa indica o multiplicador.
 - A quarta faixa indica a tolerância.

Resistores

- Tabela de cores para a identificação de resistores

Cores	Faixas 1 e 2	Faixa 3	Faixa 4
Preto	0	1	-
Marrom	1	10	1%
Vermelho	2	100	2%
Laranja	3	1000	-
Amarelo	4	10.000	-
Verde	5	100.000	-
Azul	6	1.000.000	-
Violeta	7	10.000.000	-
Cinza	8	-	-
Branco	9	-	-
Ouro	-	-	5%
Prata	-	-	10%
Sem cor	-	-	20%

Resistores

► EXEMPLO:



Primeira faixa = laranja -> 3 Segunda faixa = laranja -> 3 Terceira faixa = marrom -> 10

Resistor de: $33 * 10 = 330 \Omega$

Quarta faixa = ouro -> tolerância de 5% Resistor de:

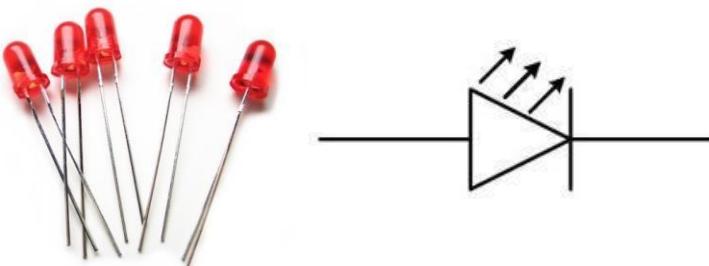
313.5Ω a 346.5Ω

LED

► LED (LIGHT-EMITTING DIODE – DIODO

Emissor de Luz)

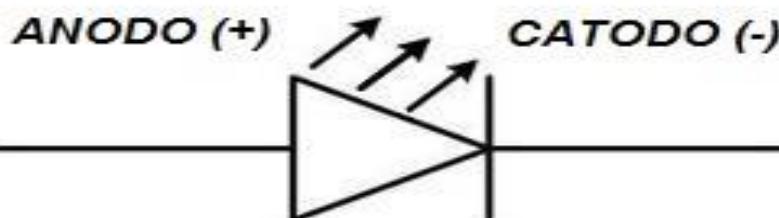
- O LED emite luz visível (amarela, verde, vermelha, laranja ou azul) ou luz infravermelha.
- Deve ser ligado em série com um resistor limitador de corrente.



LED

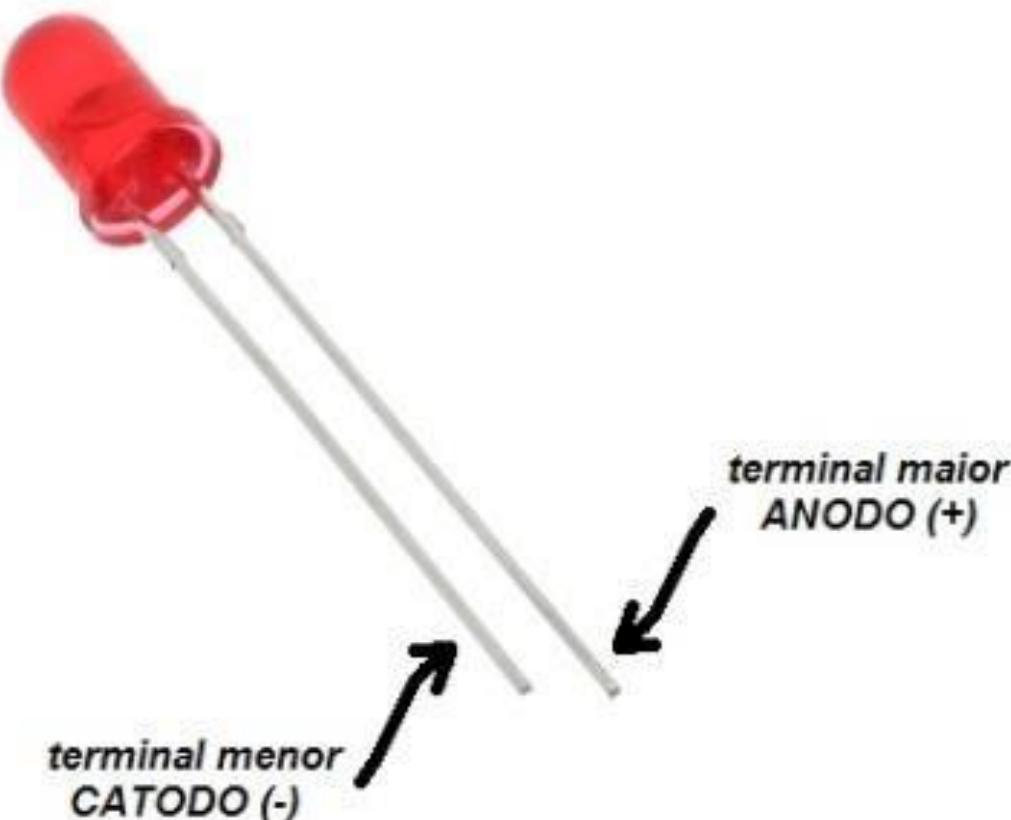
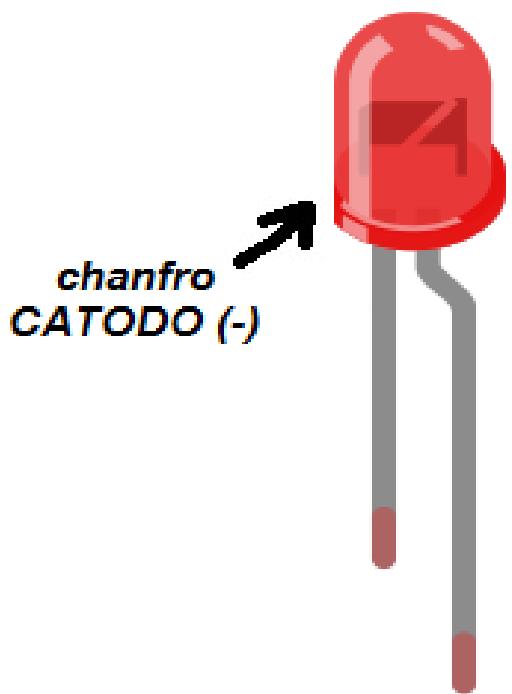
► LED (LIGHT-EMITTING DIODE – DIODO)

- Emissor de Luz)
 - O LED é um exemplo de componente
 - eletrônico polarizado.
 - O posicionamento desse componente em um circuito precisa levar em conta os polos da fonte de alimentação.



LED

► IDENTIFICAÇÃO DOS TERMINAIS



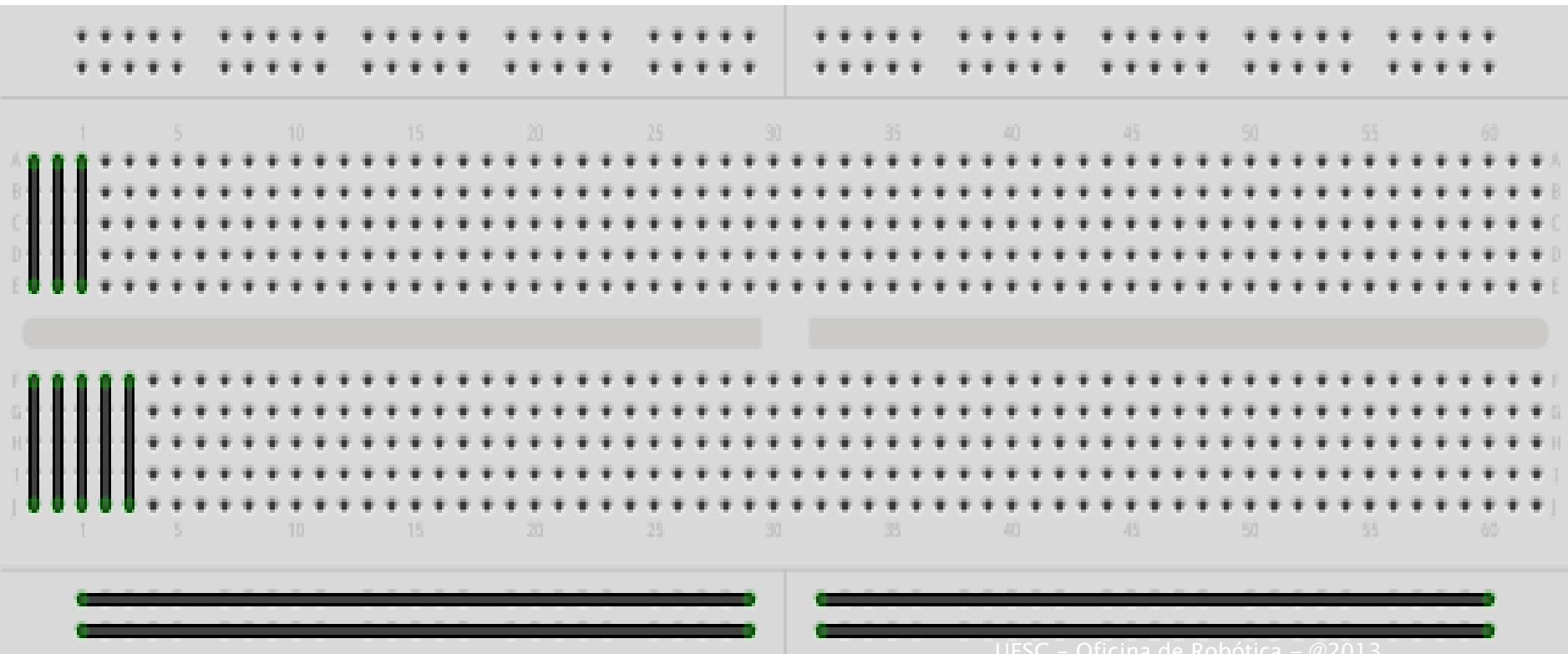
Protoboard (Matriz de Contatos)

- ▶ Ferramenta que auxilia no desenvolvimento de protótipos de circuitos eletrônicos.
- ▶ Torna desnecessária a soldagem de componentes eletrônicos em uma placa.
- ▶ É composta de furos que são interconectados por um material condutor localizado abaixo da camada de plástico.



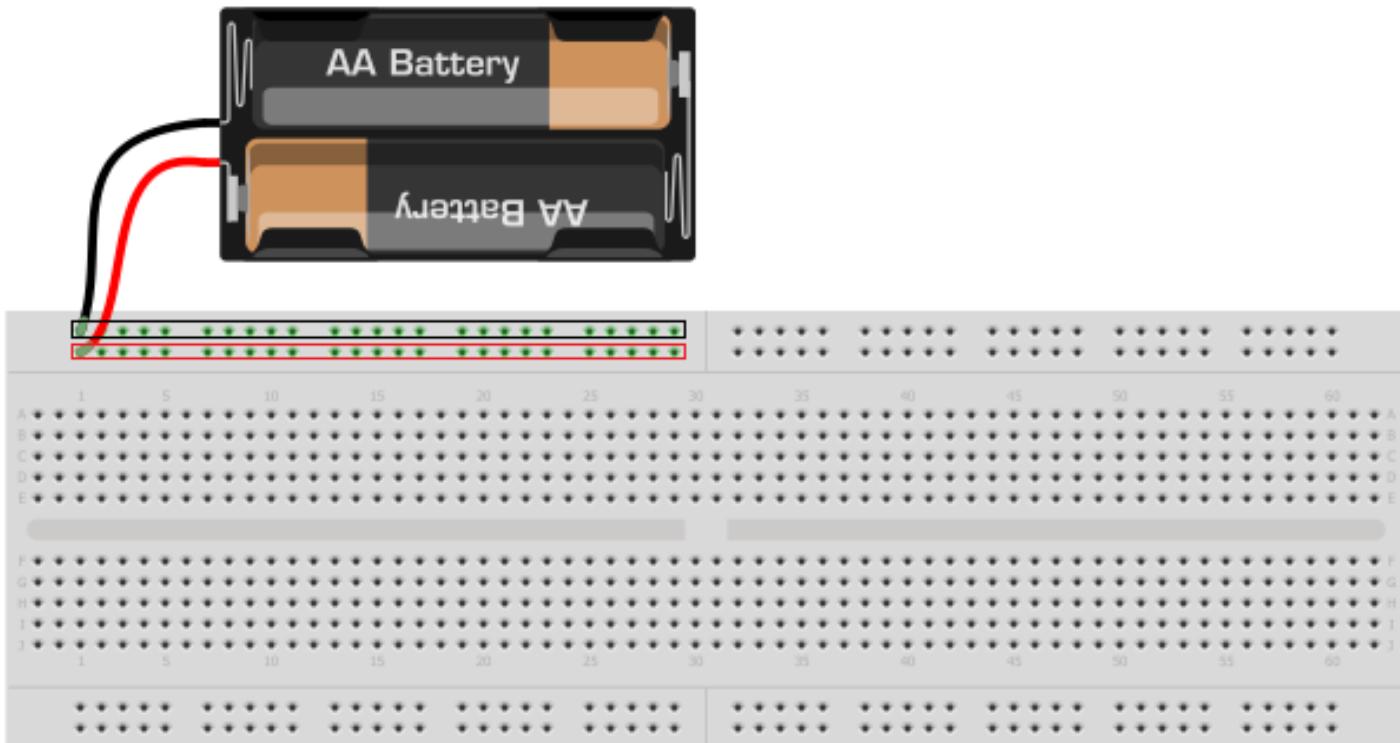
Protoboard (Matriz de Contatos)

- ▶ A figura ilustra a forma como os furos estão interconectados.



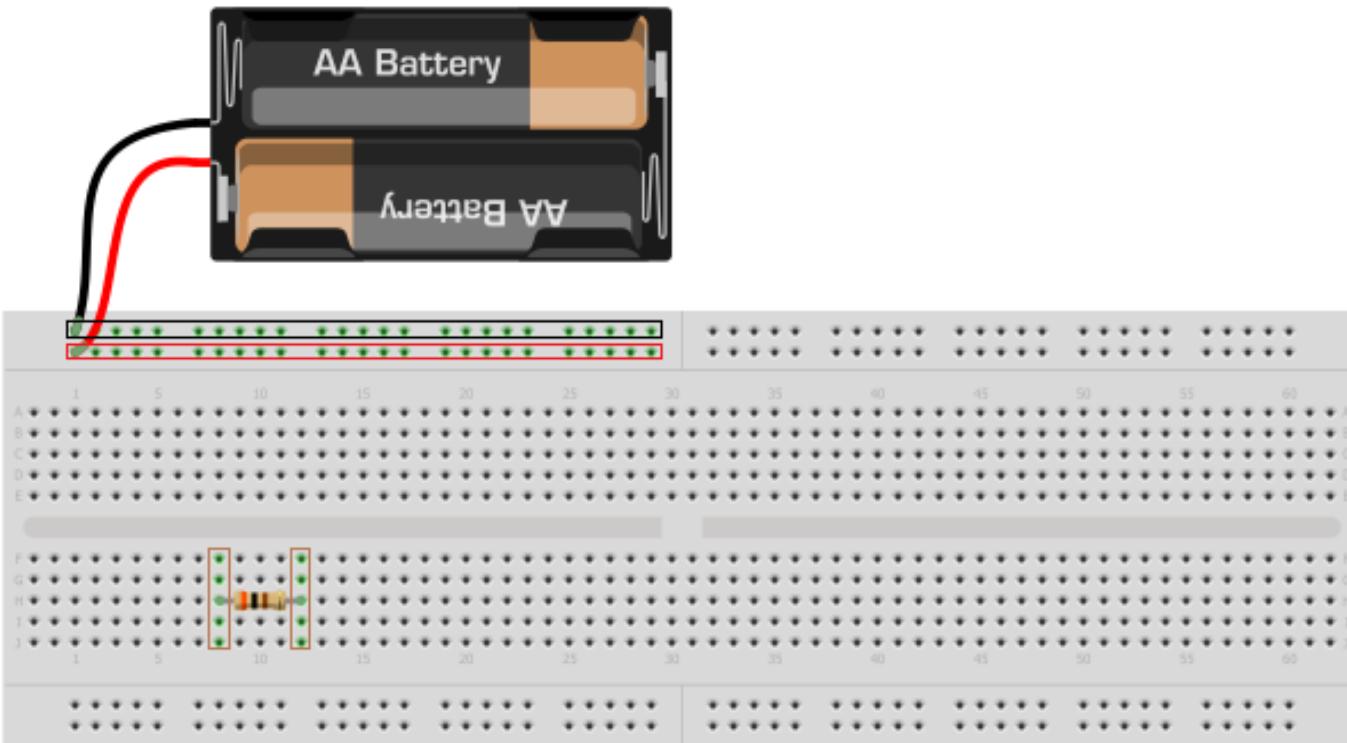
Protoboard (Matriz de Contatos)

- ▶ Passo-a-passo para a montagem de um pequeno circuito
 - Passo 1



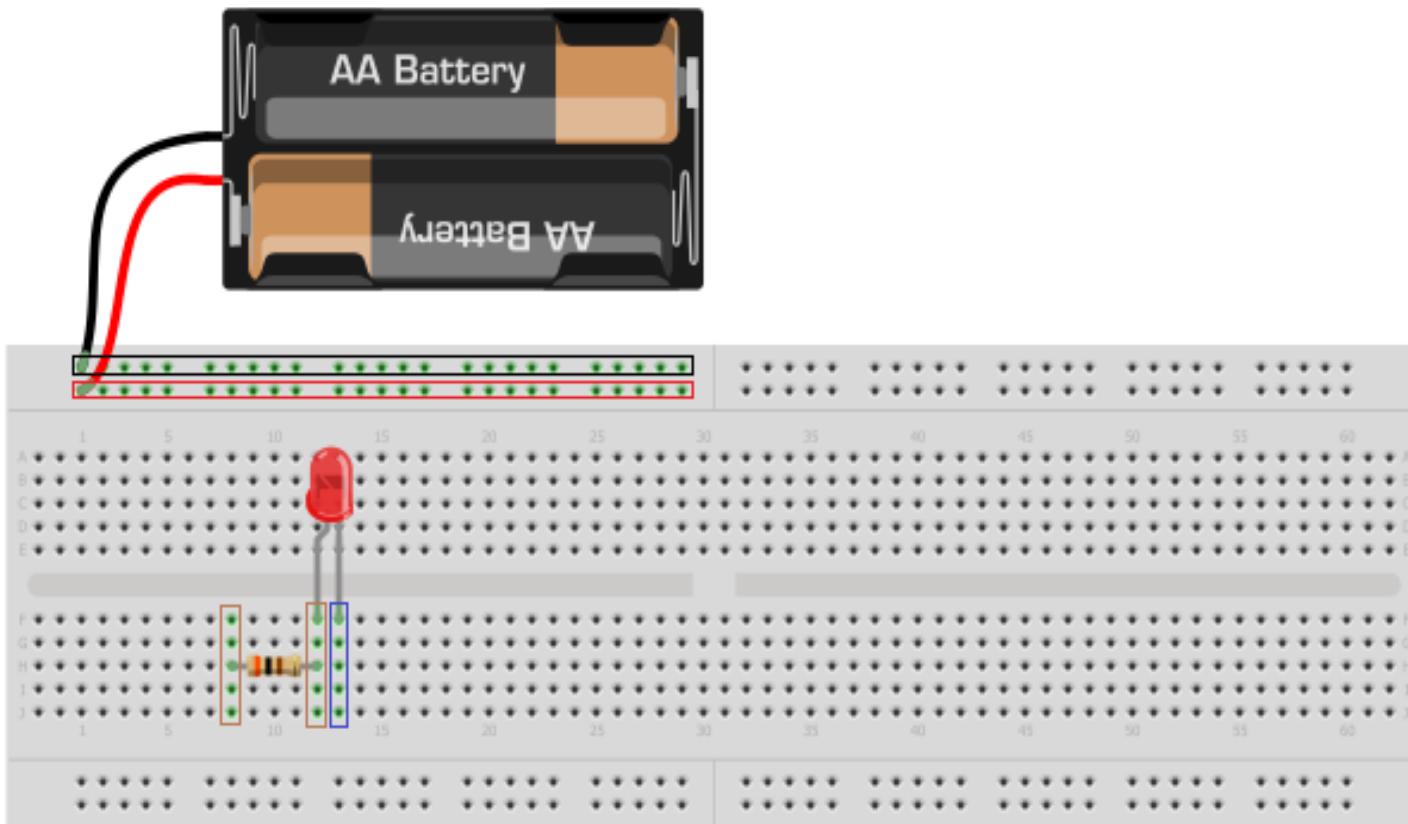
Protoboard (Matriz de Contatos)

- ▶ Passo-a-passo para a montagem de um pequeno circuito
 - Passo 2



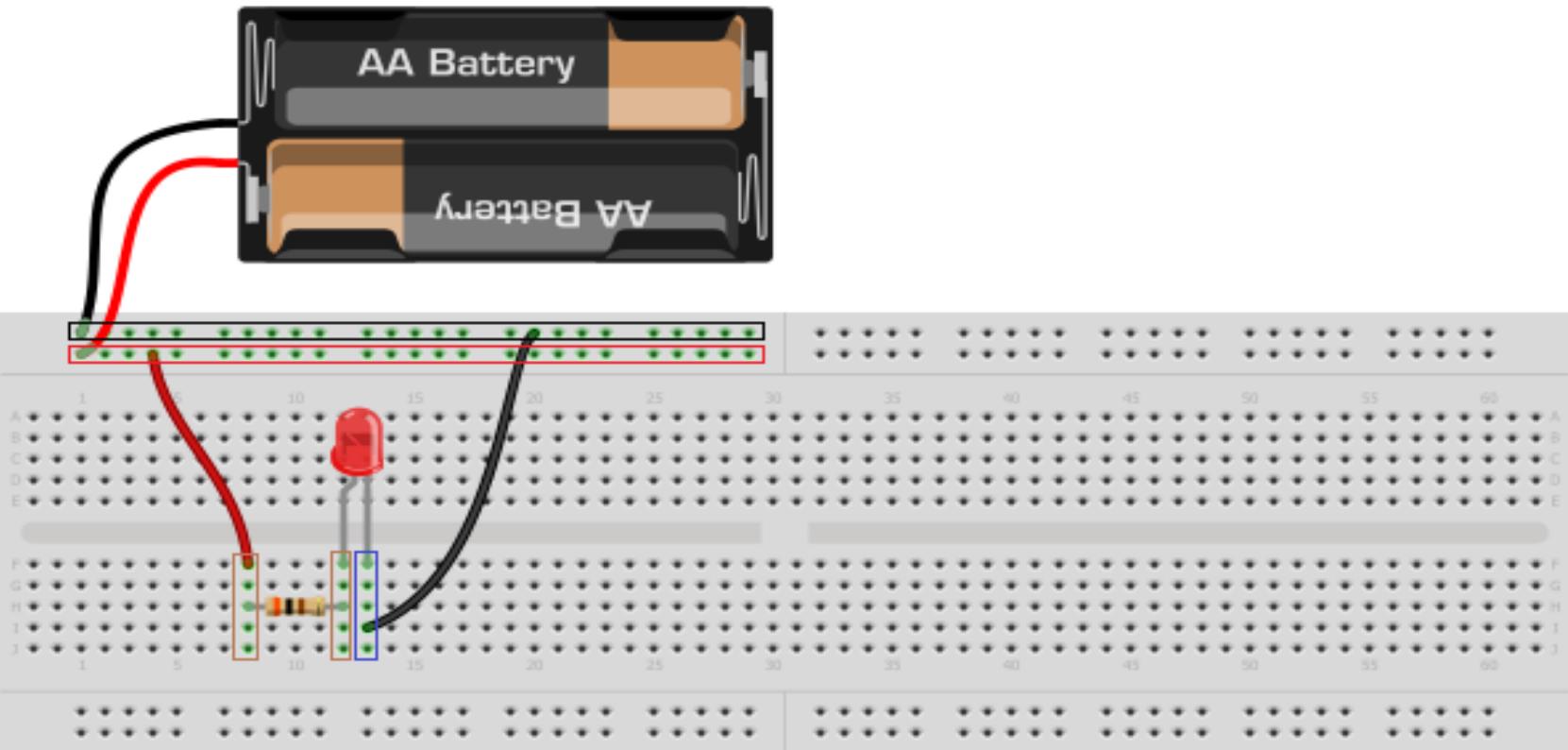
Protoboard (Matriz de Contatos)

- ▶ Passo-a-passo para a montagem de um pequeno circuito
 - Passo 3



Protoboard (Matriz de Contatos)

- ▶ Passo-a-passo para a montagem de um pequeno circuito
 - Passo 4





TM

ARDUINO

Massimo Banzi says "You don't need anybody's permission to create something great." So what you waiting for?

Apresentação

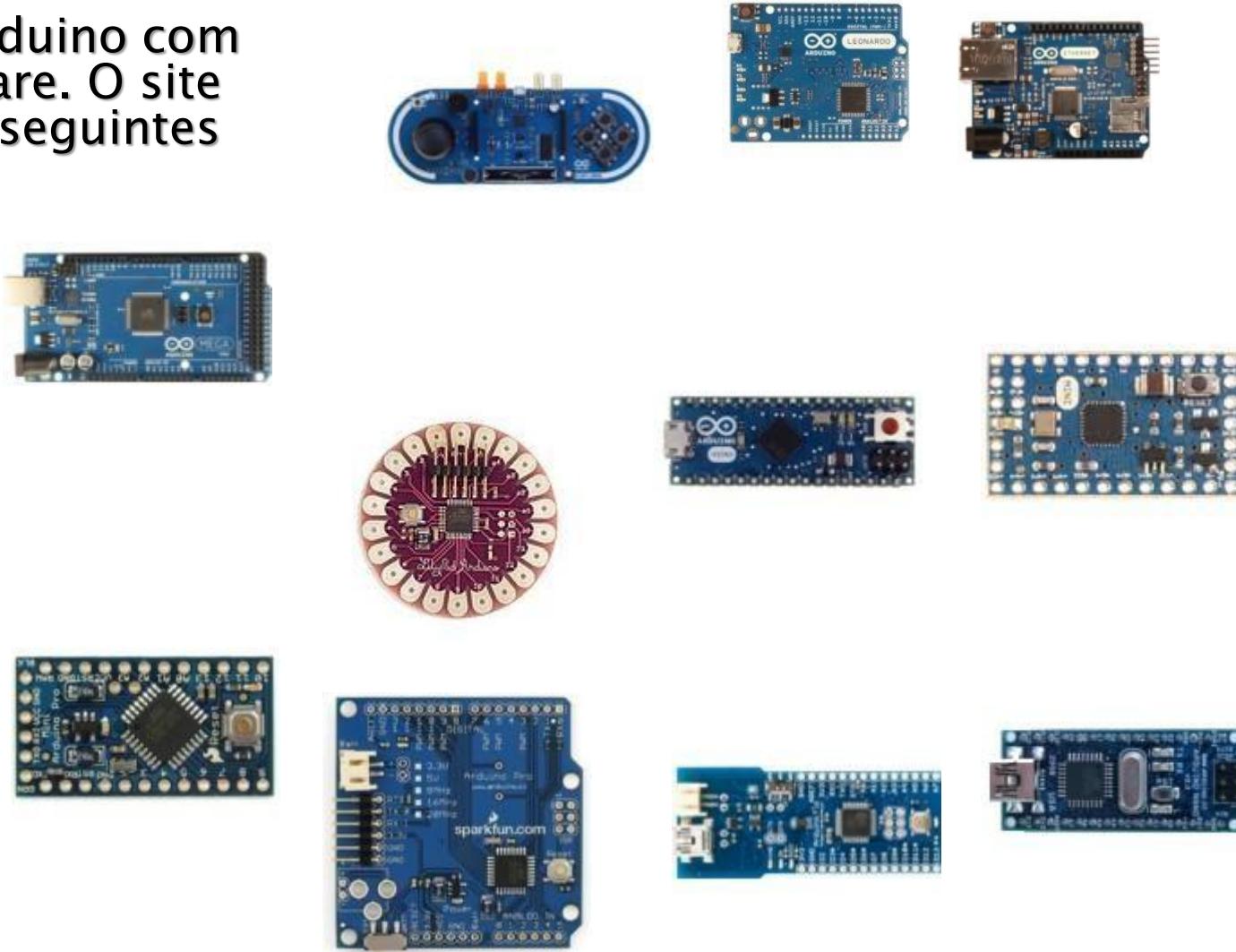
- ▶ O Arduino é uma plataforma utilizada para prototipação de circuitos eletrônicos.
- ▶ O projeto do Arduino teve início em 2005 na cidade de Ivrea, Itália.
- ▶ O Arduino é composto por uma placa com microcontrolador Atmel AVR e um ambiente de programação baseado em Wiring e C++.
- ▶ Tanto o hardware como o ambiente de programação do Arduino são livres, ou seja, qualquer pessoa pode modificá-los e reproduzi-los.
- ▶ O Arduino também é conhecido como plataforma de computação física.

Introdução ao Arduino

► TIPOS DE ARDUINO

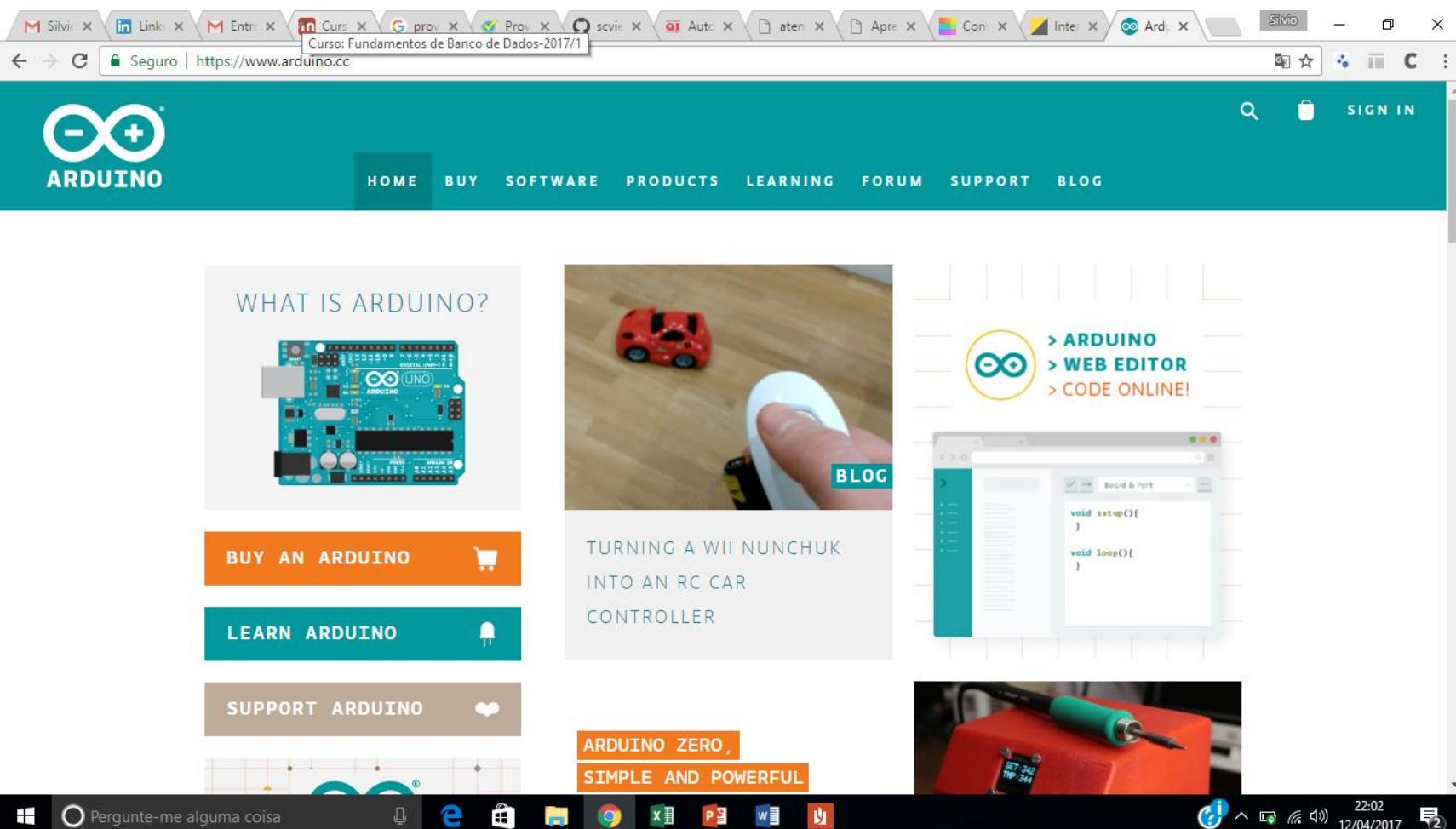
- Existem vários tipos de Arduino com especificidades de hardware. O site oficial do Arduino lista os seguintes tipos:

- Arduino UNO
- Arduino Leonardo
- Arduino Due
- Arduino Esplora
- Arduino Mega
- Arduino Mega ADK
- Arduino Ethernet
- Arduino Mini
- Arduino LilyPad
- Arduino Micro
- Arduino Nano
- Arduino ProMini
- Arduino Pro
- Arduino Fio

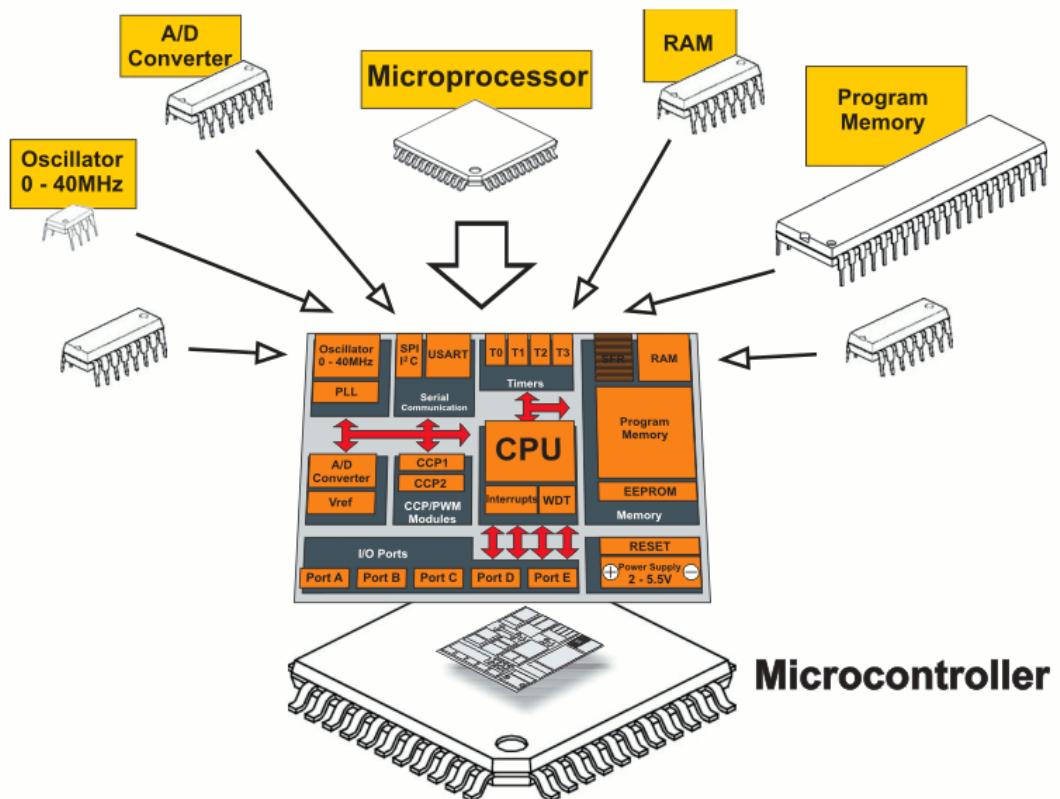


Introdução ao Arduino

► REFERÊNCIAS NA INTERNET: <http://arduino.cc>

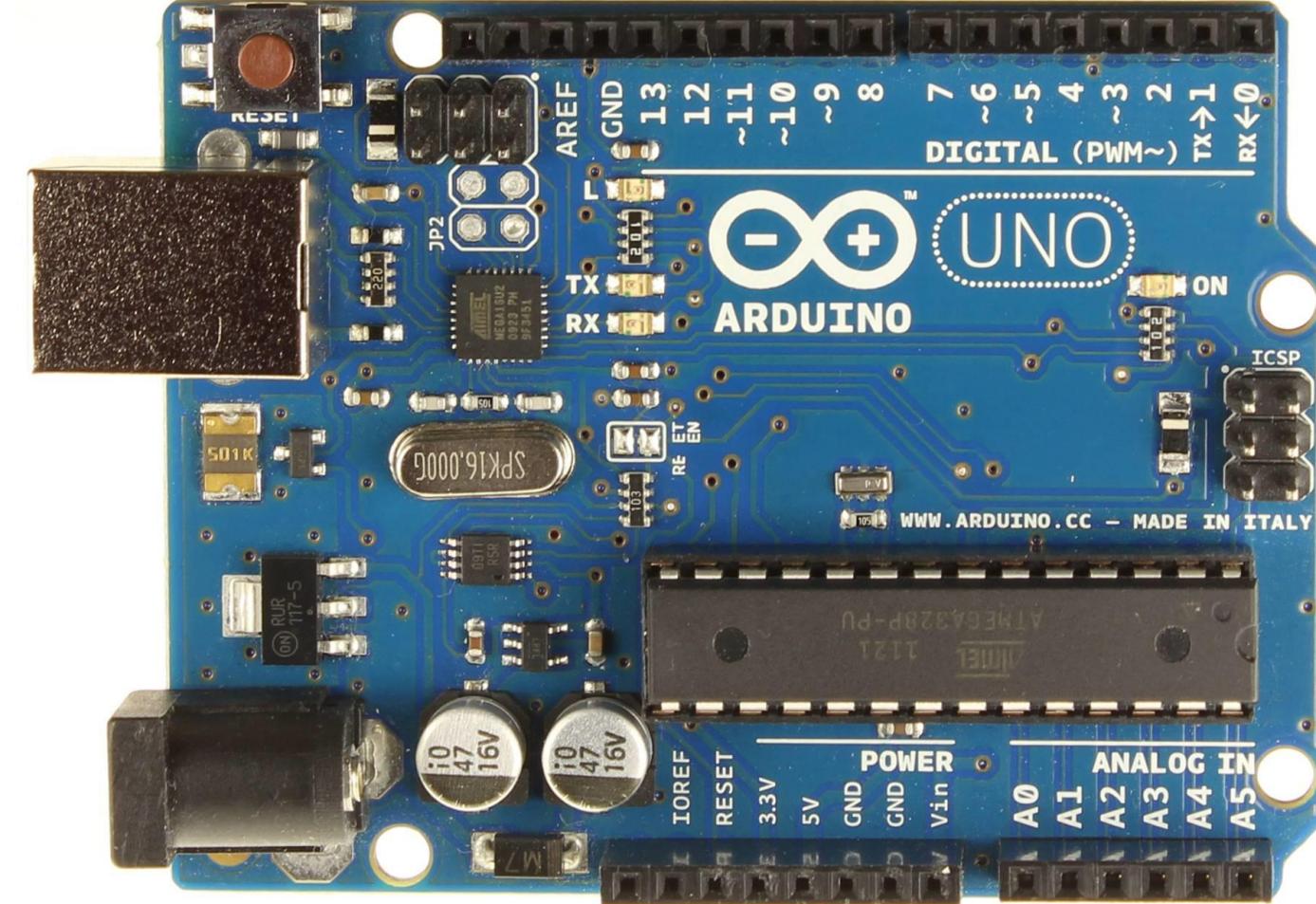


O QUE É MICROCONTROLADOR?

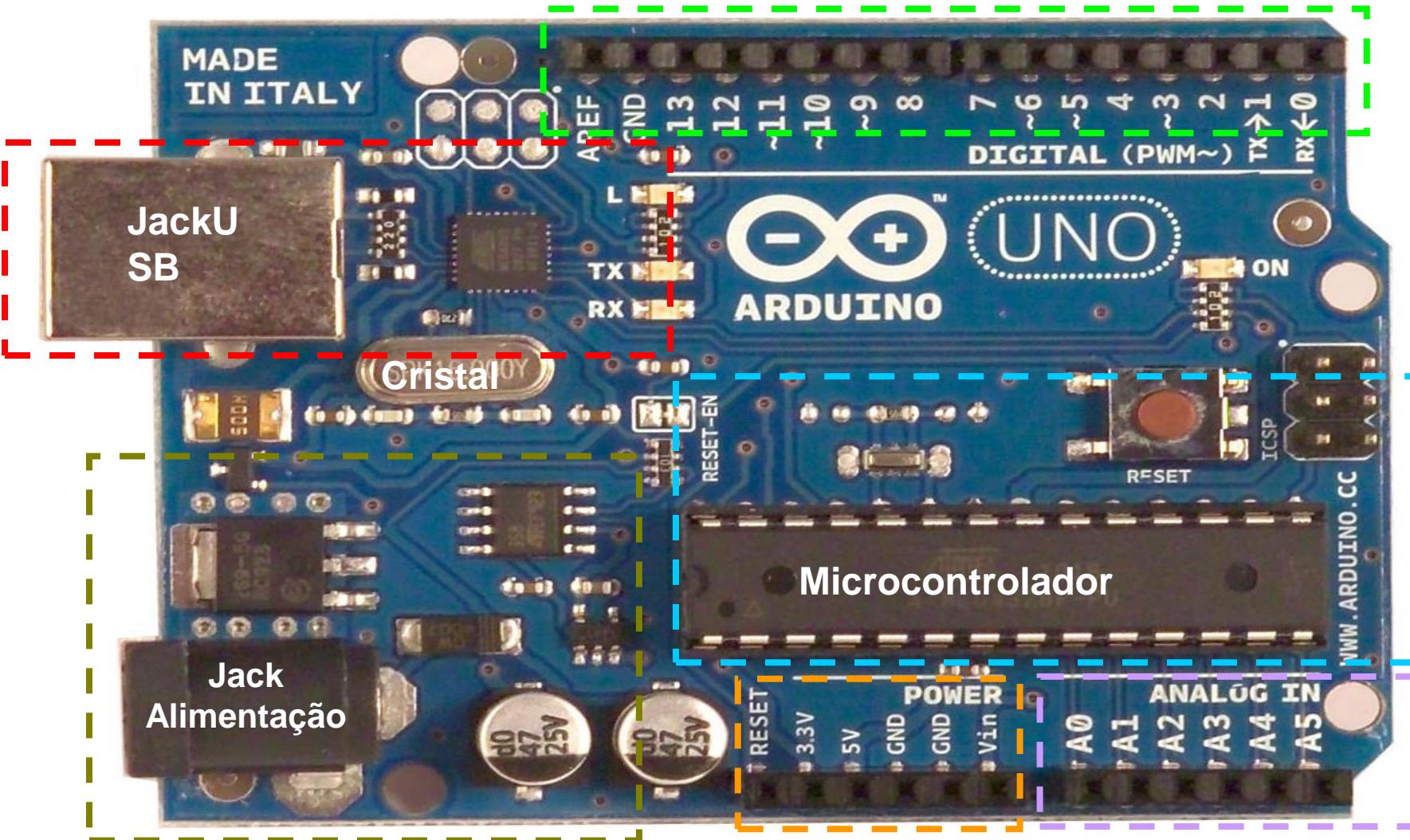


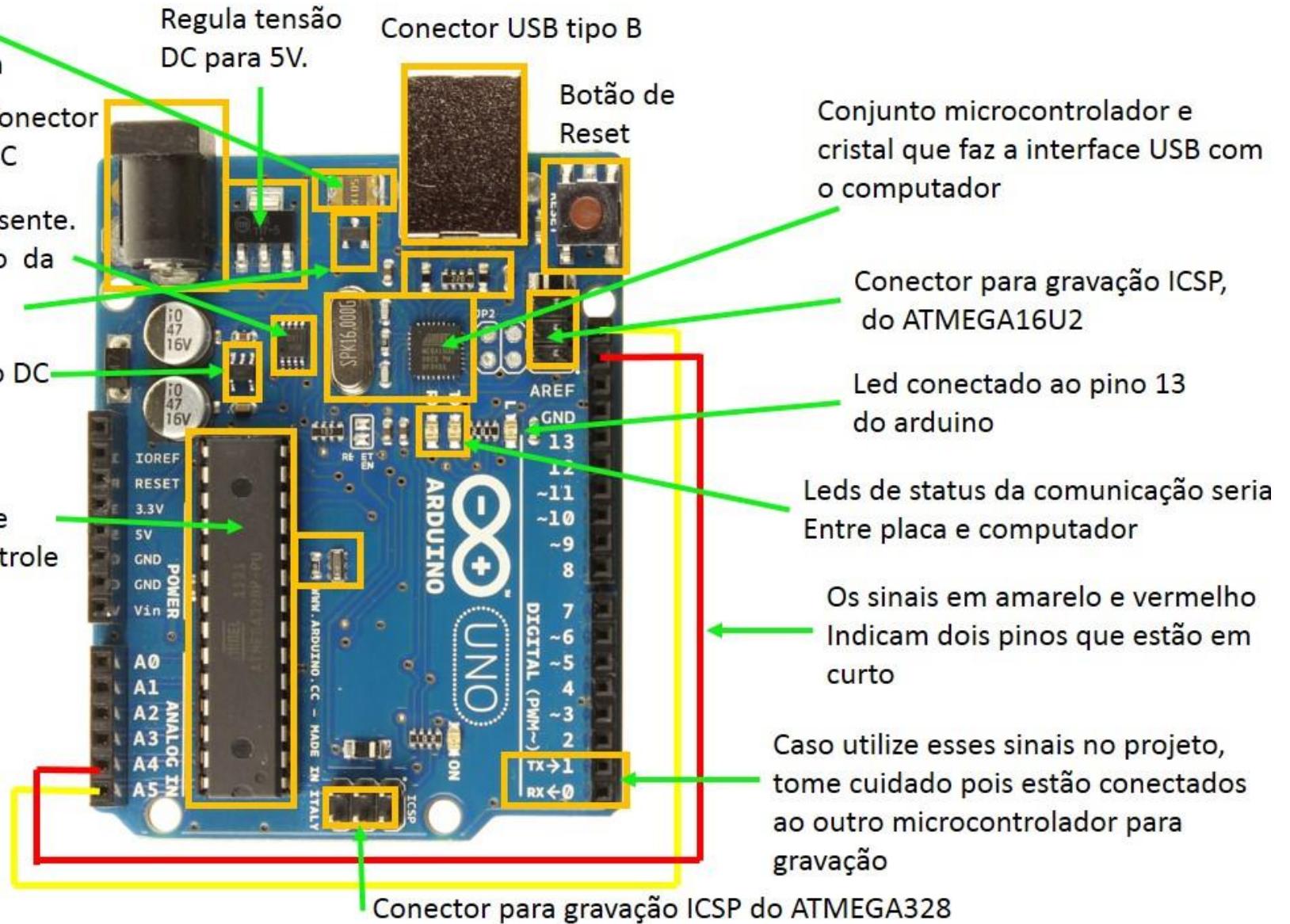
- Um pequeno computador em um único chip
 - Contendo processador, memória, e entrada/saídas
- Tipicamente "**embedded(embarcado)**" dentro de um dispositivo que ele controla
- Um microcontrolador tem tamanho reduzido e baixo custo
- Examples

Arduino UNO



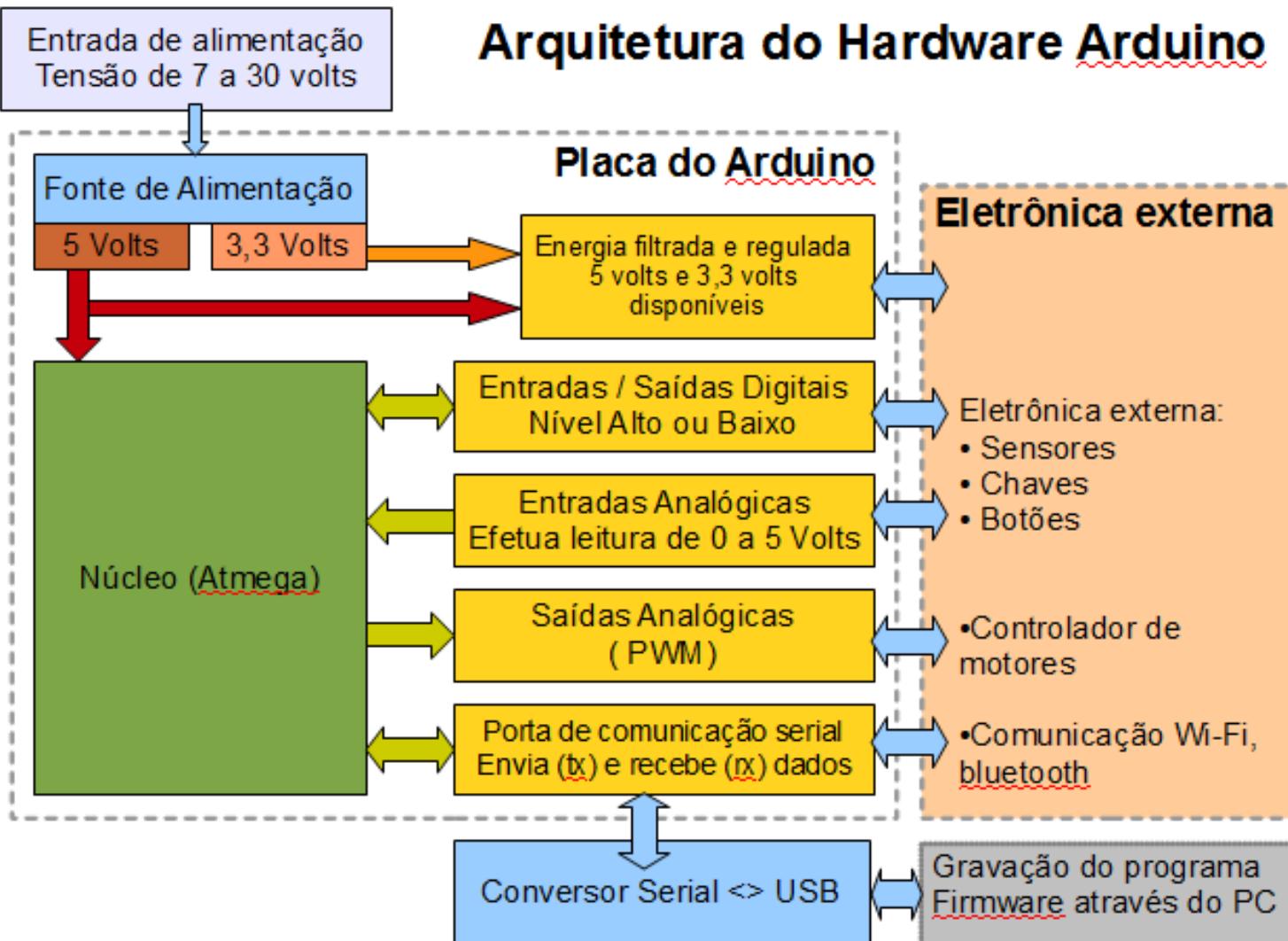
BLOCOS DO ARDUINO





ARQUITETURA DO ARDUINO

Arquitetura do Hardware Arduino



O MICROCONTROLADOR

[datasheet](#)

Arduino function

reset

(PCINT14/RESET) PC6

1

PC5 (ADC5/SCL/PCINT13)

Arduino function

analog input 5

digital pin 0 (RX)

(PCINT16/RXD) PD0

2

PC4 (ADC4/SDA/PCINT12)

digital pin 1 (TX)

(PCINT17/TXD) PD1

3

PC3 (ADC3/PCINT11)

digital pin 2

(PCINT18/INT0) PD2

4

PC2 (ADC2/PCINT10)

digital pin 3 (PWM)

(PCINT19/OC2B/INT1) PD3

5

PC1 (ADC1/PCINT9)

digital pin 4

(PCINT20/XCK/T0) PD4

6

PC0 (ADC0/PCINT8)

VCC

VCC

7

GND

GND

GND

8

AREF

analog reference

crystal

(PCINT6/XTAL1/TOSC1) PB6

9

PB5 (SCK/PCINT5)

VCC

crystal

(PCINT7/XTAL2/TOSC2) PB7

10

PB4 (MISO/PCINT4)

digital pin 13

digital pin 5 (PWM)

(PCINT21/OC0B/T1) PD5

11

PB3 (MOSI/OC2A/PCINT3)

digital pin 12

digital pin 6 (PWM)

(PCINT22/OC0A/AIN0) PD6

12

PB2 (SS/OC1B/PCINT2)

digital pin 11(PWM)

digital pin 7

(PCINT23/AIN1) PD7

13

PB1 (OC1A/PCINT1)

digital pin 10 (PWM)

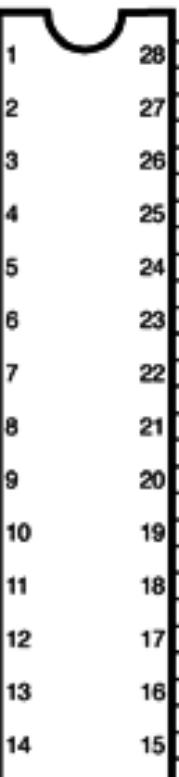
digital pin 8

(PCINT0/CLK0/ICP1) PB0

14

PB0 (OC1A/PCINT1)

digital pin 9 (PWM)



Arduino UNO

► PINOS



Pinos 3V3, 5V e Gnd: dos 6 pinos dessa barra somente os quatro do meio servem para alimentar um circuito externo conectado ao Arduino: o pino de 5V e o terra (os dois pinos Gnd entre 5V e Vin); e o pino 3V3 que disponibiliza essa tensão com uma corrente máxima de 50mA.

Arduino UNO

► PINOS

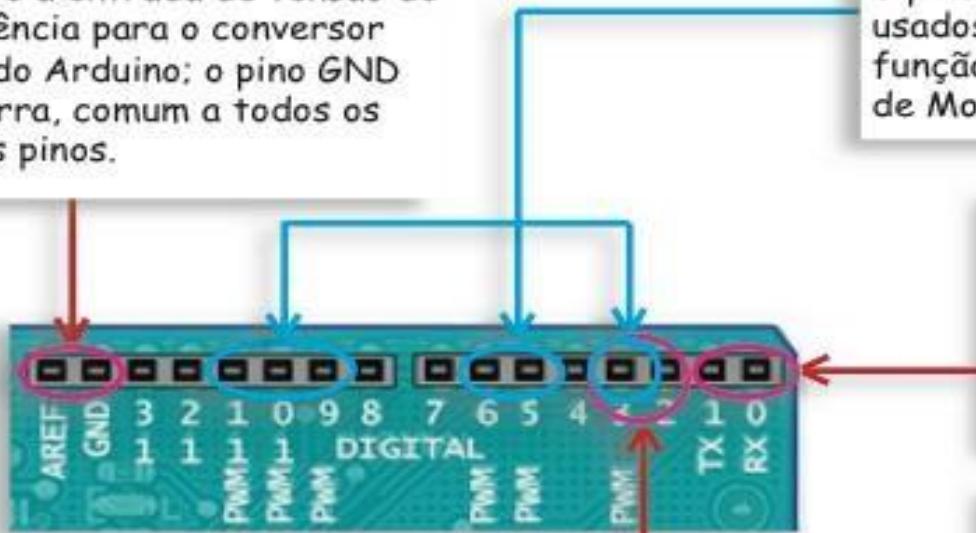


Pinos 0 a 5: esses 6 pinos aceitam tensões entre zero e 5 volts CC que vão ao conversor A/D de 10 bits no microcontrolador do Arduino. O pino AREF, numa das barras de pinos digitais, é a entrada de tensão de referência para esse conversor.

Arduino UNO

► PINOS

Pinos AREF e GND: o pino AREF é a entrada de tensão de referência para o conversor A/D do Arduino; o pino GND é o terra, comum a todos os outros pinos.



Pinos 3, 5 e 6 e 9 a 11 (PWM): 6 pinos dos 14 pinos digitais podem ser usados para gerar sinais analógicos com a função `analogWrite()` utilizando a técnica de Modulação por Largura de Pulso (PWM).

Pinos 0 e 1: os dois primeiros pinos digitais são conectados a USART do microcontrolador do Arduino para comunicação serial com um computador.

Pinos 2 e 3: pinos que chamam uma ISR (Interrupt Service Routine) para tratar uma interrupção com a função `attachInterrupt()` nesses pinos.

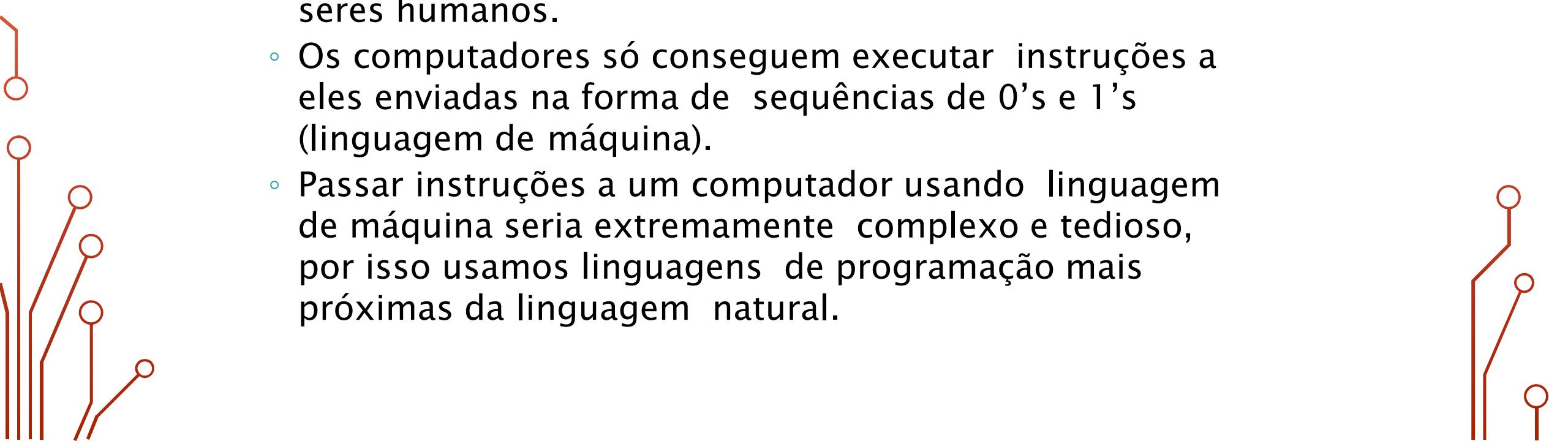
Programação

- ▶ O microcontrolador do Arduino é um computador que segue instruções detalhadas dadas por seres humanos.
- ▶ Para que o Arduino execute determinada tarefa, precisamos “ensiná-lo” a executar essa tarefa, passo a passo.
- ▶ Os humanos passam instruções para o Arduino escrevendo programas.
- ▶ Um programa é uma sequência de instruções codificadas em uma linguagem de programação.



Programação

► LINGUAGENS DE PROGRAMAÇÃO

- Uma linguagem de programação é um meio utilizado para se comunicar com computadores, inclusive o Arduino, de um modo relativamente simples para os seres humanos.
 - Os computadores só conseguem executar instruções a eles enviadas na forma de sequências de 0's e 1's (linguagem de máquina).
 - Passar instruções a um computador usando linguagem de máquina seria extremamente complexo e tedioso, por isso usamos linguagens de programação mais próximas da linguagem natural.
- 

Programação

► ALGORITMO

Bolo de Chocolate simples

Ingredientes

Massa:

1 xícara(s) (chá) de leite
1 xícara(s) (chá) de Óleo de soja
2 unidade(s) de ovo
2 xícara(s) (chá) de farinha de trigo
1 xícara(s) (chá) de achocolatado em pó
1 xícara(s) (chá) de açúcar
1 colher(es) (sopa) de fermento químico em pó

Cobertura:

2 colher(es) (sopa) de manteiga
3 colher(es) (sopa) de achocolatado em pó
3 colher(es) (sopa) de açúcar
5 colher(es) (sopa) de leite



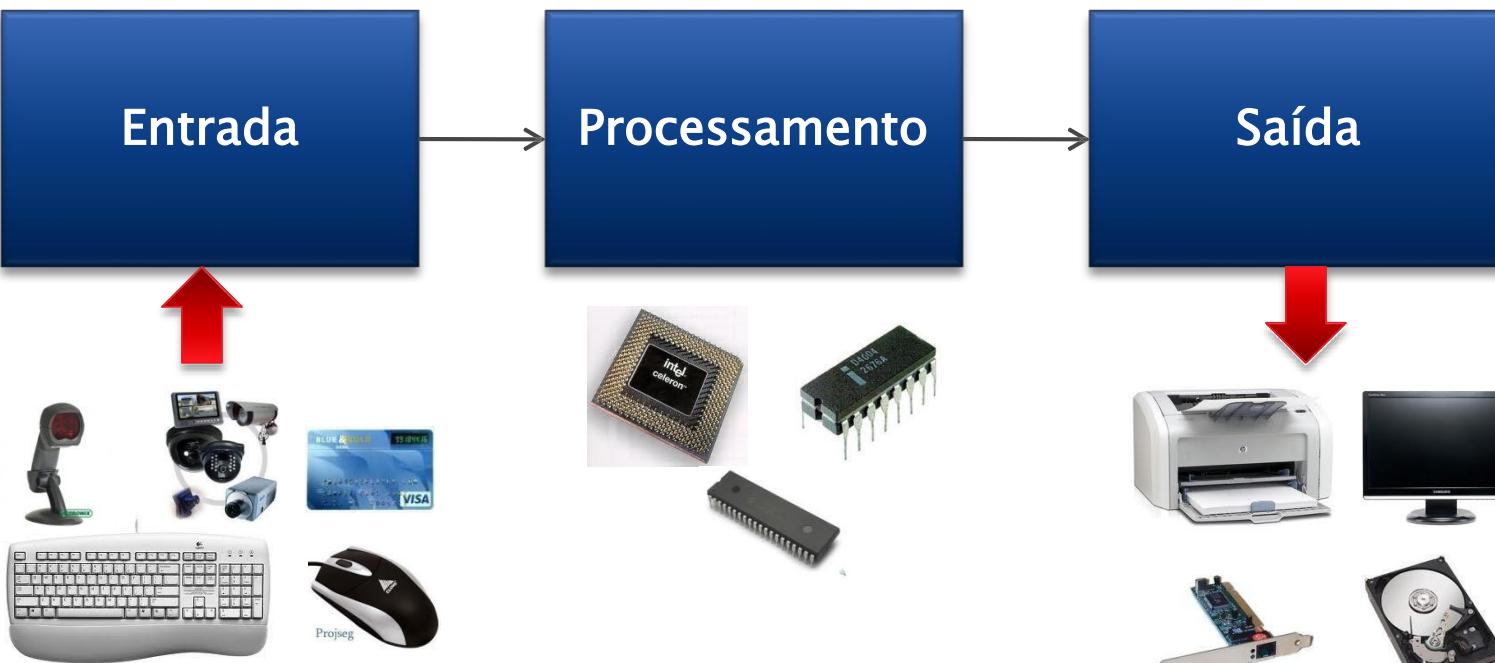
Modo de preparo

Massa: Coloque os líquidos no liquidificador e bata até misturar bem. Coloque os outros ingredientes, sendo o fermento o último. Leve para assar em forno médio, numa forma untada e enfarinhada.

Cobertura: Para a cobertura, misture numa panela a manteiga, o achocolatado, o açúcar e o leite. Leve ao fogo até derreter e a calda ficar homogênea. Cubra o bolo ainda quente, furadinho com garfo.

Programação

PROCESSAMENTO DE DADOS



Ambiente de Desenvolvimento

- ▶ O AMBIENTE DE DESENVOLVIMENTO DO ARDUINO (IDE) É GRATUITO E PODE SER BAIXADO NO SEGUINTE ENDEREÇO:
ARDUINO.CC.
- ▶ AS PRINCIPAIS FUNCIONALIDADES DO IDE DO ARDUINO SÃO:
 - Escrever o código do programa
 - Salvar o código do programa
 - Compilar um programa
 - Transportar o código compilado para a placa do Arduino

SOFTWARE

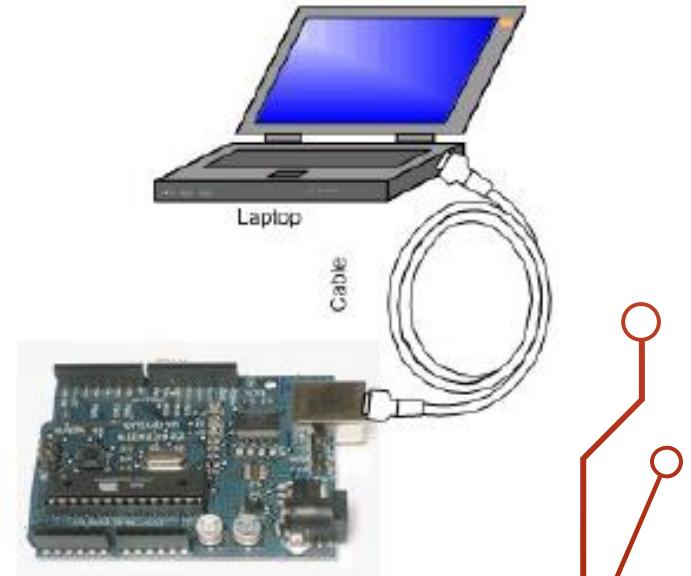
O **1º passo** consiste em efectuar o *download* do respectivo *software* de desenvolvimento, o Arduino IDE 1.0.5, através do site oficial [Arduino](#)

O **2º passo** consiste em descompactar o ficheiro “.ZIP” para uma pasta à sua escolha.

O **3º passo** consiste em ligar a placa Arduino ao computador através do cabo USB e instalar os *drivers* FTDI, para permitir uma conversão de USB para série.

O **4º passo** consiste em configurar a porta série a ser utilizada e qual o tipo de modelo Arduino, que nos encontramos a utilizar. Para tal, necessitamos de abrir o *Software* de desenvolvimento e escolher na barra de separadores a opção “Tools”.

O **5º passo** para a utilização do *Software* consiste em elaborar o seu *Sketch* (*programa*), compilar e, caso não tenha erros, fazer o *uploading* para a placa Arduino.



Ambiente de Desenvolvimento

Arquivo Editar Sketch Ferramentas Ajuda

s4firmware2017

```
// NEW IN VERSION 1.6c (by Jorge Gomez):
// Fixed variable type in pin structure: pin.state should be int, not byte
// Optimized speed of execution while receiving data from computer in readSerialPort()

// NEW IN VERSION 1.6b (by Jorge Gomez):
// Added new structure arduinoPins to hold the pins information:
// - This makes the code easier to read and modify (IMHO)
// - Allows to change the type of pin more easily to meet non standard use of S4A
// - Eliminates the need of having to deal with different kind of index access (ie: states[pin-4])
// - By using an enum to hold all the possible output pin states the code is now more readable
// Changed all functions using old style pin access: configurePins(), resetPins(), readSerialPort(), updateActuator() and sendUpdateActuator()
// Fixed possible overflow every 70 minutes (2e32 us) in pulse() while using micros(). Changed for delayMicroseconds()
// Some minor coding style fixes

// NEW IN VERSION 1.6a (by Jorge Gomez):
// Fixed compatibility with Arduino Leonardo by avoiding the use of timers
// readSerialPort() optimized:
// - created state machine for reading the two bytes of the S4A message
// - updateActuator() is only called if the state is changed
// Memory use optimization
// Cleaning some parts of code
// Avoid using some global variables

// NEW IN VERSION 1.6:
// Refactored reset pins
// Merged code for standard and CR servos
// Merged patch for Leonardo from Peter Maillot (many thanks for this!)
```

Arduino/Genuino Uno em COM3

19:10 14/04/2017

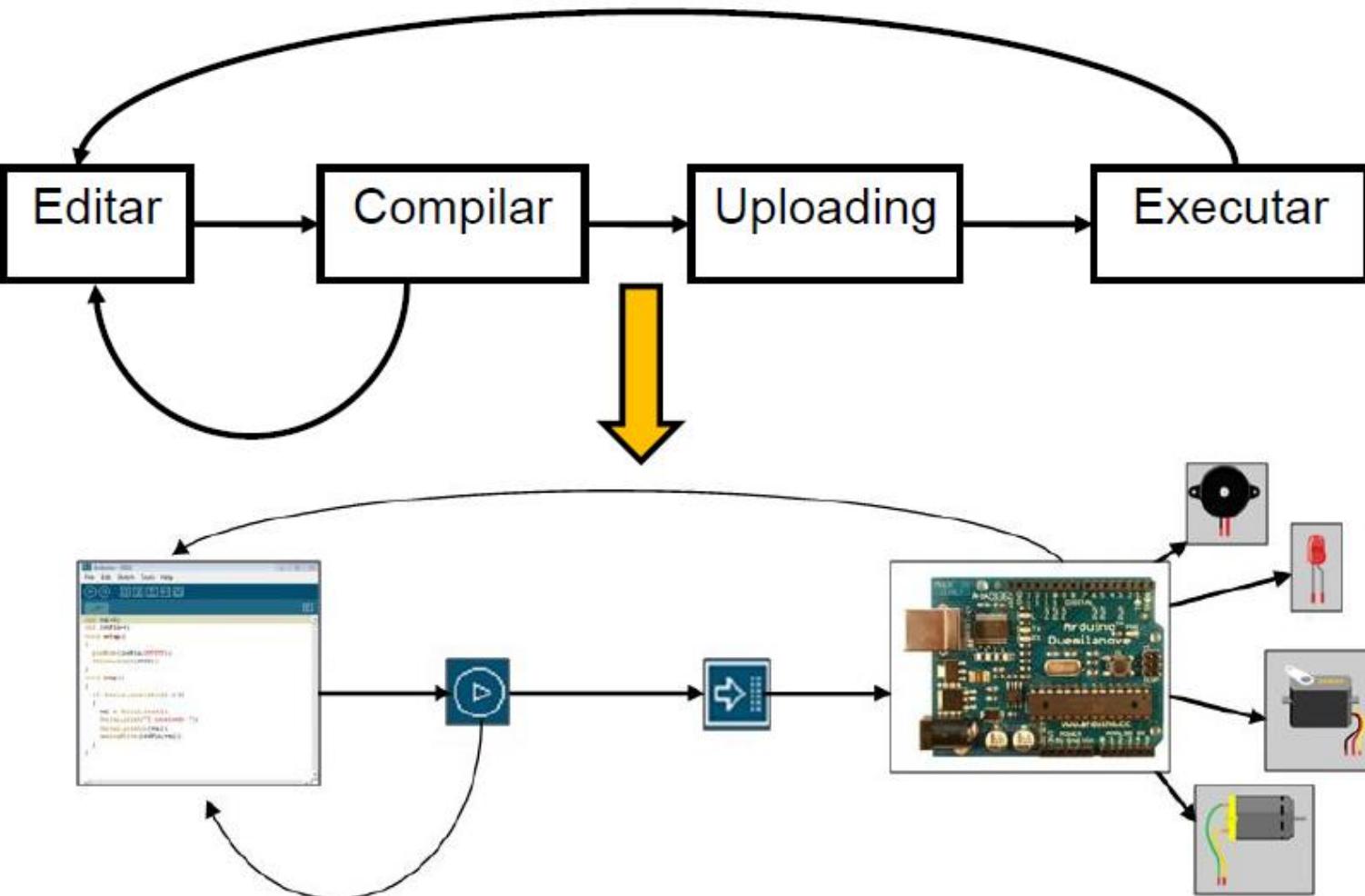
Pergunte-me alguma coisa

ARDUINO IDE



O monitor serial é utilizado para comunicação entre o Arduino e o computador (PC).

CICLO DE DESENVOLVIMENTO



Funções *setup()* e *loop()*

► PRIMEIRO PROGRAMA: BLINK LED

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

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

Funções *setup()* e *loop()*

- ▶ AS DUAS PRINCIPAIS PARTES (FUNÇÕES) DE UM PROGRAMA DESENVOLVIDO PARA O ARDUINO SÃO:
 - *setup()*: onde devem ser definidas algumas configurações iniciais do programa. Executa uma única vez.
 - *loop()*: função principal do programa. Fica executando indefinidamente.
- ▶ Todo programa para o Arduino deve ter estas duas funções.

FUNÇÕES BASE

void setup() - Esta função apenas é executada uma vez e é normalmente utilizada para executar a inicialização de variáveis, a inicialização da utilização bibliotecas, a definição dos pinos (como *input* ou *output*), o início do uso de comunicação série, entre outros. Esta função apenas volta a ser executada novamente ao ser efectuado o *reset* ou quando se desligar e volta a ligar a placa de desenvolvimento *Arduino*.

void loop() - Esta função faz um “loop” sucessivo (como o próprio nome indica), ou seja, todos os comandos existentes no interior desta função são sucessivamente repetidos, o que pode permitir a leitura sucessiva de portas, a leitura sucessiva de parâmetros provenientes de sensores externos e actuar de acordo com as condições estabelecidas.

Declaração de variáveis globais;

```
void setup(){  
  Instrução 1;  
  Instrução 2;  
  (...)  
}  
  
void loop(){  
  Instrução 6;  
  Instrução 9;  
  Função01();  
  (...)  
}
```

Conjunto de instruções apenas executado uma vez,
na inicialização do programa a executar

Conjunto de instruções que é executado em “loop”

Comentários

- ▶ MUITAS VEZES É IMPORTANTE COMENTAR ALGUMA PARTE DO CÓDIGO DO PROGRAMA.
- ▶ Existem duas maneiras de adicionar comentários a um programa em Arduino.
 - A primeira é usando //, como no exemplo abaixo:
 - // Este é um comentário de linha
 - A segunda é usando /* */ , como no exemplo abaixo:
 - /* Este é um comentário de bloco. Permite acrescentar comentários com mais de uma linha */
- ▶ Nota:
 - Quando o programa é compilado os comentários são automaticamente suprimidos do arquivo executável, aquele que será gravado na placa do Arduino.

PROGRAMAR NO ARDUINO

Comentários

Muitas vezes é importante comentar alguma parte do código do programa. Existem duas maneiras de adicionar comentários a um programa em Arduino.

A primeira é usando `//`, como no exemplo abaixo:
`// Este é um comentário de linha`

A segunda é usando `/* */`, como no exemplo abaixo:
`/* Este é um comentário de bloco. Permite acrescentar comentários com mais de uma linha */`

Nota: Quando o programa é compilado os comentários são automaticamente suprimidos do arquivo executável, aquele que será gravado na placa do Arduino.

Constantes

- ▶ UM DADO É CONSTANTE QUANDO NÃO SOFRE NENHUMA VARIAÇÃO NO DECORRER DO TEMPO.
- ▶ DO INÍCIO AO FIM DO PROGRAMA O VALOR PERMANECE INALTERADO.
- ▶ Exemplos:
 - 10
 - “Bata antes de entrar!”
 - -0,58

Constantes

- ▶ A CRIAÇÃO DE CONSTANTES NO ARDUINO PODE SER FEITA DE DUAS MANEIRAS:
 - Usando a palavra reservada **const**
 - Exemplo:
 - `const int x = 100;`
 - Usando a palavra reservada **define**
 - Exemplo:
 - `#define X 100`

Constantes

► NO ARDUINO EXISTEM ALGUMAS CONSTANTES PREVIAMENTE DEFINIDAS E SÃO CONSIDERADAS PALAVRAS RESERVADAS.

► AS CONSTANTES DEFINIDAS SÃO:

- true – indica valor lógico verdadeiro
- false – indica valor lógico falso
- HIGH – indica que uma porta está ativada, ou seja, está em 5V.
- LOW – indica que uma porta está desativada, ou seja, está em 0V.
- INPUT – indica que uma porta será de entrada de dados.
- OUTPUT – indica que uma porta será de saída de dados.

Comentários

► PRIMEIRO PROGRAMA COMENTADO

comentarios

```
/*
 *          OFICINA DE ROBÓTICA - LARM - UFSC
 *
 * Blink Led: Primeiro programa em Arduino.
 *             Pisca um led conectado à porta 13.
 */
// função usada para configurações iniciais
void setup()
{
    pinMode(13, OUTPUT);
}

// principal função do programa - laço infinito
void loop()
{
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);
}
```

Portas digitais

- ▶ O Arduino possui tanto portas digitais quanto portas analógicas.
- ▶ As portas servem para comunicação entre o Arduino e dispositivos externos, por exemplo: ler um botão, acender um led ou uma lâmpada.
- ▶ Conforme já mencionado, o Arduino UNO, possui 14 portas digitais e 6 portas analógicas (que também podem ser utilizadas como portas digitais).

Portas digitais

- ▶ As portas digitais trabalham com valores bem definidos. No caso do Arduino esses valores são 0V e 5V.
- ▶ 0V indica a ausência de um sinal e 5V indica a presença de um sinal.
- ▶ Para escrever em uma porta digital basta utilizar a função `digitalWrite(pino, estado)`.
- ▶ Para ler um valor em uma porta digital basta utilizar a função `digitalRead(pino)`.

Portas digitais

- ▶ AS PORTAS DIGITAIS SÃO USADAS PARA ENTRADA e saída de dados.
- ▶ Para definir se uma porta será usada para entrada ou para saída de dados, é necessário explicitar essa situação no programa.
- ▶ A função `pinMode(pino, estado)` é utilizada para definir se a porta será de entrada ou saída de dados.
- ▶ Exemplos:
 - Define que a porta 13 será de saída
 - `pinMode(13, OUTPUT)`
 - Define que a porta 7 será de entrada
 - `pinMode(7, INPUT)`

Exemplo 1 - Acionamento de LED interno

Começaremos com o exemplo Blink, que já vem no aplicativo. Para encontrar o exemplo clique em Arquivos → Exemplos → Digital → Blink.

O programa tem como objetivo acender e apagar o LED de um em um segundo. Para compilar este exemplo não é necessário de nenhuma outra infraestrutura que não o próprio Arduino.

Exercícios

- ▶ Utilizar o programa anterior pisca LED para comandar o circuito na Protoboard;
- ▶ Ampliar o circuito, utilizando mais portas e controlando mais LEDs;
- ▶ Sincronizar os LEDs, para que sejam habilitados e desabilitados sequencialmente (exercício semáforo);

Exercícios

- ▶ Montar o circuito semáforo de forma de maquete (utilizando material fornecido);
- ▶ Utilizar 2 semáforos sincronizados funcionais (quando semáforo 1 estiver aberto o 2 deve estar fechado)

ARDUINO UNO - CARACTERÍSTICAS

- Microcontrolador: ATmega328
- Tensão de operação: 5V
- Tensão recomendada (entrada): 7-12V
- Limite da tensão de entrada: 6-20V
- Pinos digitais: 14 (seis pinos com saída PWM)
- Entrada analógica: 6 pinos
- Corrente contínua por pino de entrada e saída: 40 mA
- Corrente para o pino de 3.3 V: 50 mA
- Quantidade de memória FLASH: 32 KB (ATmega328) onde 0.5 KB usado para o bootloader
- Quantidade de memória SRAM: 2 KB (ATmega328)
- Quantidade de memória EEPROM: 1 KB (ATmega328)
- Velocidade de clock (cristal): 16 MHz

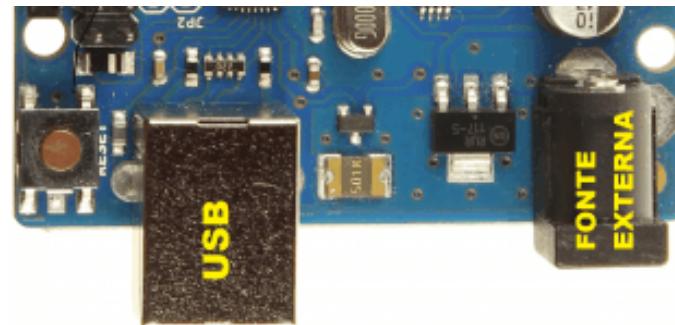
Flash	É aqui que o sketch e o bootloader são armazenados
SRAM	Onde são criadas e modificadas as variáveis ao longo da execução do sketch
EEPROM	Pode ser usada para guardar informação (não volátil)

ARDUINO UNO - ALIMENTAÇÃO

Internamente, o circuito do Arduino é alimentado com uma tensão de 5V.

O Arduino UNO pode ser alimentado pela porta USB ou por uma entrada do tipo “Power Jack” (com o positivo ao centro) através de uma fonte externa DC.

A recomendação é que a fonte externa seja de 7 V a 12 V e pode ser ligada diretamente no conector de fonte ou nos pinos Vin e Gnd.



3,3 V. - Fornece tensão de 3,3V. para alimentação de shield e módulos externos.
Corrente máxima de 50 mA.

5 V - Fornece tensão de 5 V para alimentação de shields e circuitos externos.

GND - pinos de referência, terra.

VIN - pino para alimentar a placa através de shield ou bateria externa. Quando a placa é alimentada através do conector Jack, a tensão da fonte estará nesse pino.

