

PROJETO ARDROBOTIC

PROGRAMAÇÃO COM ARDUINOS



Arduino

O Arduino é uma plataforma de prototipagem eletrónica open-source, baseada em princípios de flexibilidade e facilidade de utilização, para hardware e software.

Consiste numa placa microcontroladora, programável preparada para receber sinais de sensores e acionar atuadores.

A linguagem de programação é baseada em Wiring (semelhante a C/C++).



Arduino – História

Em 2005, Massimo Banzi e David Cuartielles desenvolveram o Arduino, um dispositivo programável, de fácil utilização, com o objetivo de poder ser utilizado em projetos interativos de arte e design.



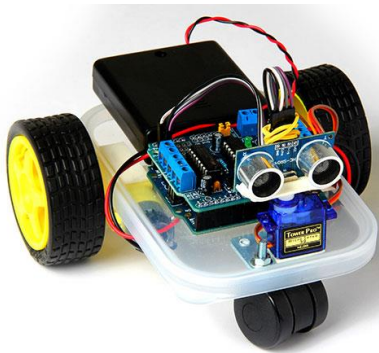
Projetaram então um dispositivo simples, de baixo custo, fácil de conectar a diversos componentes, como relés, motores ou sensores e fácil de programar.



Utilizaram um microcontrolador Atmega de 8 bits e projetaram uma placa com conexões de simples utilização, desenvolveram o firmware do bootloader para o microcontrolador e o ambiente de desenvolvimento integrado (IDE) – Software Arduino.

Um dos seus alunos, David Mellis, ficou responsável pelo desenvolvimento do software **Arduino**, baseado em Wiring.

Arduino – Aplicações



Arduino - Modelos mais comuns

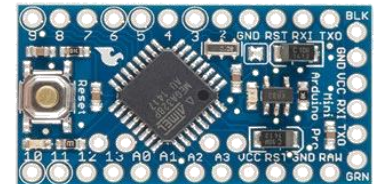
Arduino UNO



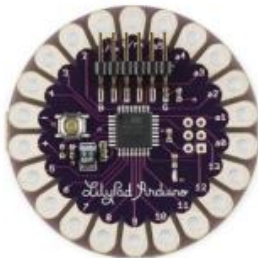
Arduino Mega



Arduino Mini



Lilypad



Arduino Nano



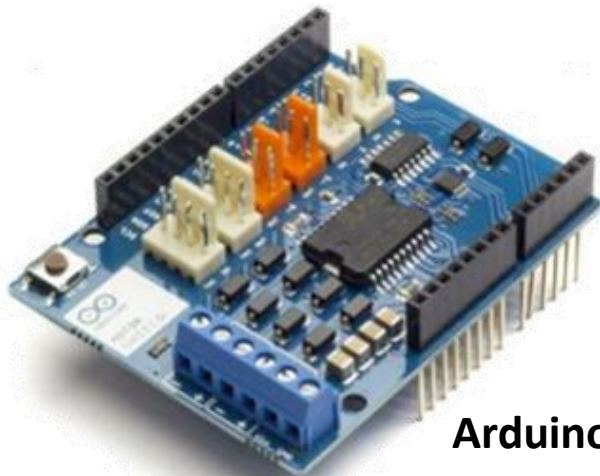
Shields (extensões) do Arduino



Arduino Ethernet Shield



Arduino GSM Shield



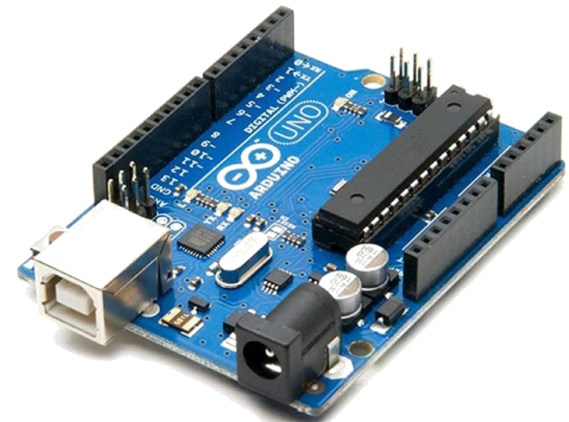
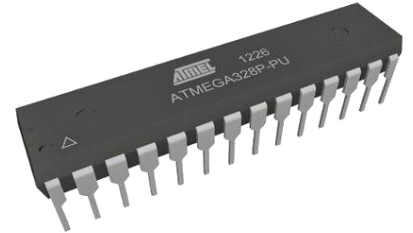
Arduino Motor Shield



Arduino Bluetooth Shield

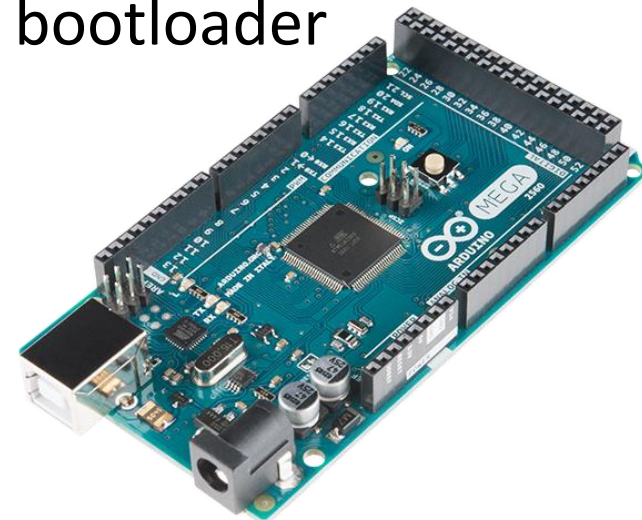
Arduino Uno – Características técnicas

- Micro controlador: ATmega328P
- Tensão de operação: 5V
- Tensão de entrada (recomendada): 7-12V
- Tensão de entrada (limite): 6-20V
- Pinos Digitais de E/S: 14 (6 pins PWM)
- Pinos Analógicos (Entrada): 6
- Corrente máxima por pino de E/S: 40mA
- Memória Flash: 32 kB (ATmega328), 0.5kB usado pelo bootloader
- SRAM: 2 kB
- EEPROM: 1 kB
- Velocidade de relógio: 16 MHz



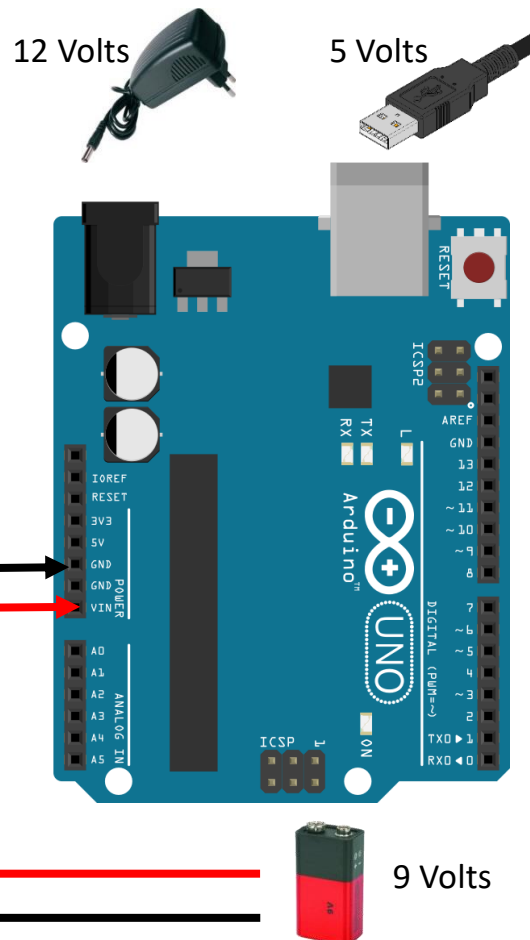
Arduino Mega – Características técnicas

- Micro controlador: ATmega2560
- Tensão de operação: 5V
- Tensão de entrada (recomendada): 7-12V
- Tensão de entrada (limite): 6-20V
- Entradas/Saídas: 54 (15 pins PWM)
- Pins Analógicos (Entrada): 16
- Memória Flash: 256 kB, 8kB usados pelo bootloader
- SRAM: 8 kB
- EEPROM: 4 kB
- Velocidade do Clock: 16 MHz

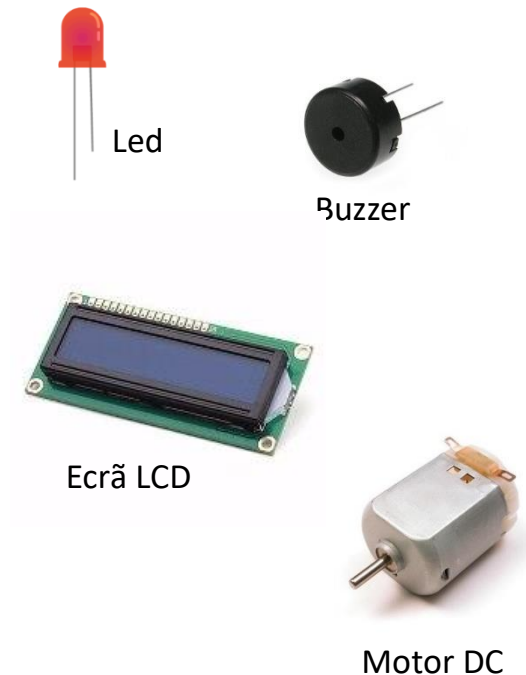


Componentes eletrônicos simples

Input (Ler)



Output (Escrever)



VIN - pino para alimentar a placa através de uma pilha/bateria externa

Explorando o Arduino UNO

3,3V - Fornece tensão de 3,3V para alimentação de shields e módulos externos.

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

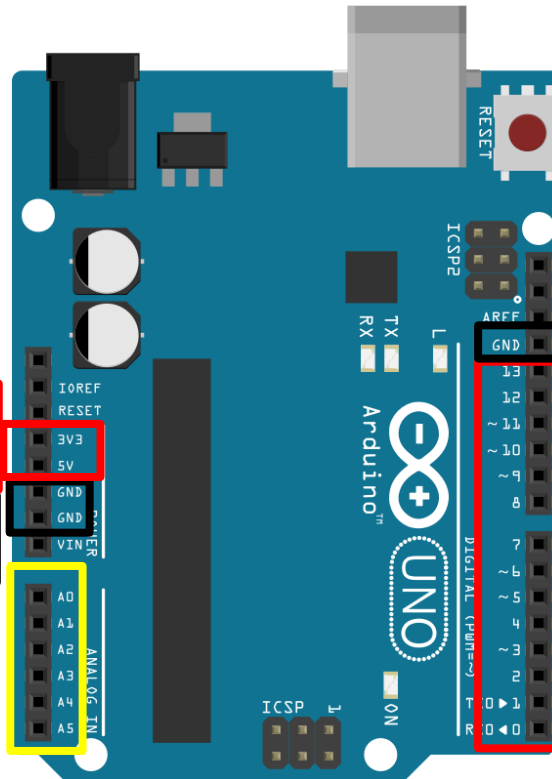
3,3 Volts

5 Volts

GROUND - 0 Volts (GND)

Pinos Analógicos:

A0 – A5 (Input)



GND - pinos de referência, terra.

Pinos Digitais:

0 – 13 (Input / Output)

~3
~5
~6

Exemplo:

PWM

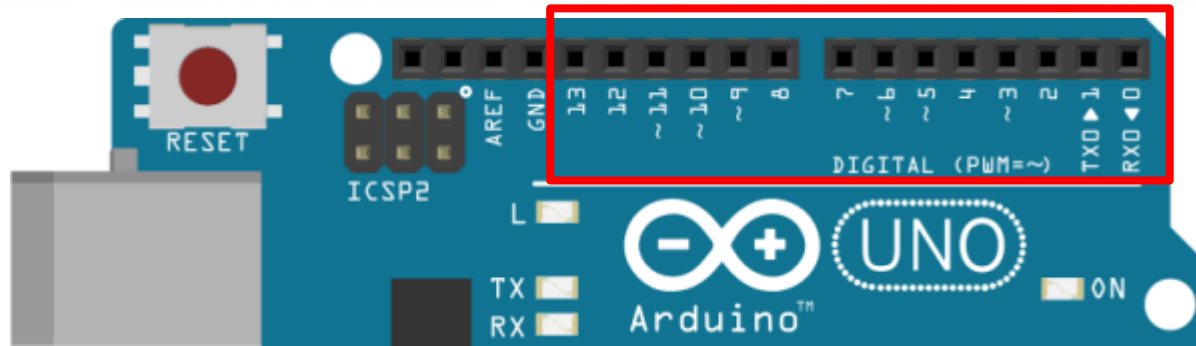
~9
~10
~11



Para variar a velocidade de um motor DC



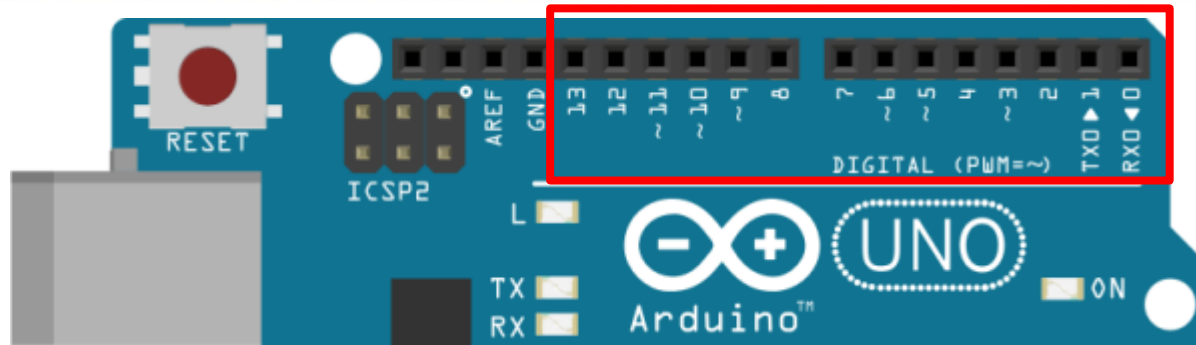
Explorando o Arduino UNO



Pinos digitais operam a **5V**, onde cada pino pode fornecer ou receber uma corrente máxima de **40 mA**. Cada pino possui uma resistência de **pull-up interno (20-50 kOhms)** que pode ser ativada por software.

Digitais: 0 a 13 podem ser usados como entradas/saídas digitais através das função **digitalWrite()** e **digitalRead()** como input ou output através da função **pinMode()**;

Explorando o Arduino UNO

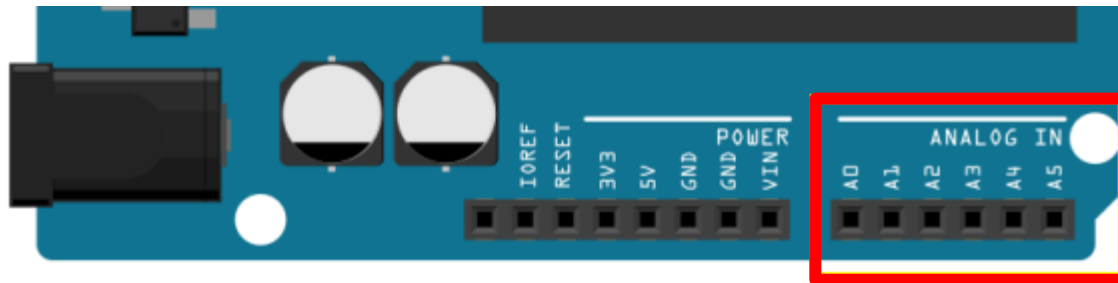


PWM: 3,5,6,9,10 e 11 podem ser usados como saídas PWM de 8 bits através da função `analogWrite()` – Output Analógico;

Serial: 0 (RX) and 1 (TX). Usados para receber (RX) e transmitir (TX);

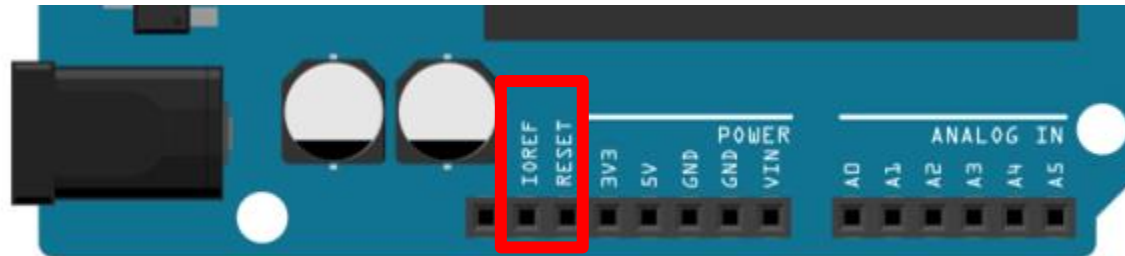
LED 13: Existe um LED embutido no pin digital 13.

Explorando o Arduino UNO



Analógicas: 0 a 5 podem ser usados como entradas analógicas através da função **`analogRead()`**, cada pin disponibiliza 10bits (ou seja valores de **0 a 1023**). Por defeito debitam entre **0 e 5 volts**;

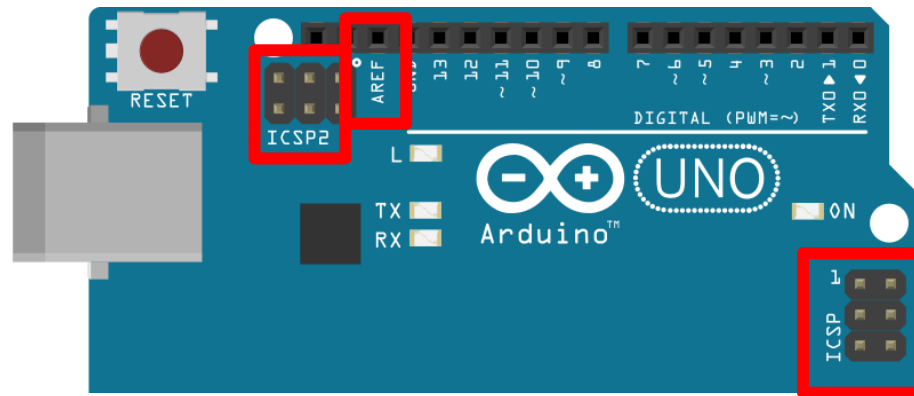
Explorando o Arduino UNO



IOREF: Fornece a tensão de referência com que o microcontrolador funciona, para que os shields possam selecionar a fonte de alimentação apropriada ou para que possam ativar a conversão de tensão para que as saídas funcionem a 5V ou 3,3V.

RESET: pino conectado ao pino de RESET do microcontrolador.

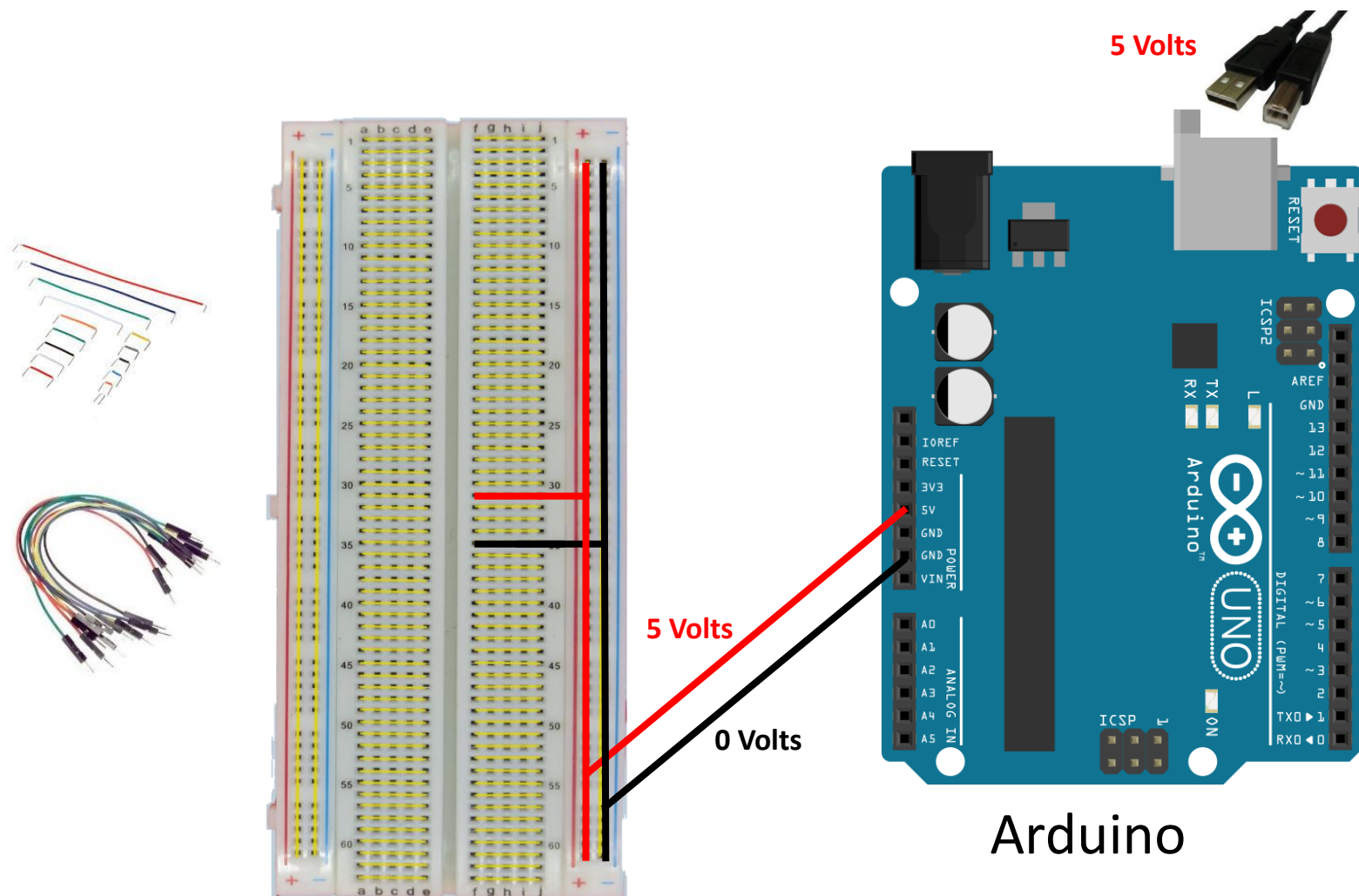
Explorando o Arduino UNO



AREF: Fornece uma tensão de referência para pinos analógicos

ICSP: representa um dos vários métodos disponíveis para a programação de placas Arduino. Normalmente, o bootloader do Arduino é utilizado para programar uma placa Arduino, mas se este estiver ausente ou danificado, podemos recorrer ao ICSP. O ICSP pode também ser usado para recuperar um bootloader ausente ou danificado.

Placa de ensaio (Breadboard)

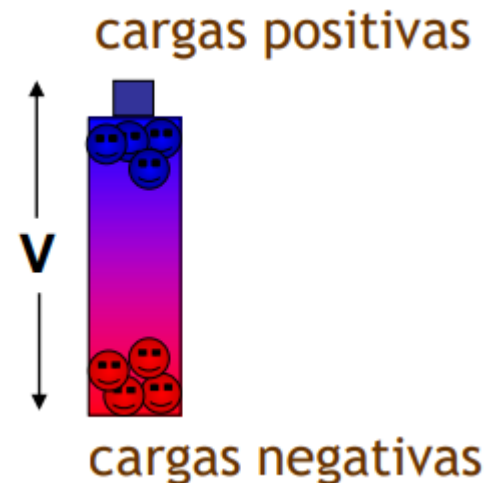


Conceitos Básicos Eletricidade

Diferença de potencial (v - volts)

v - é a unidade de tensão elétrica (diferença de potencial elétrico)

Diferença de potencial ou
tensão.

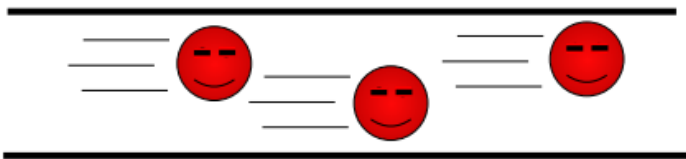


Quanto maior a tensão, mais “força” tem os elétrons

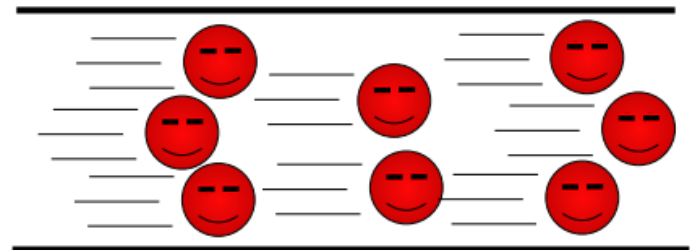
Conceitos Básicos Eletricidade

Corrente Elétrica (i - amperes)

Um ampere (símbolo: A) é definido como a intensidade de uma corrente elétrica constante



Fluxo de elétrons num condutor

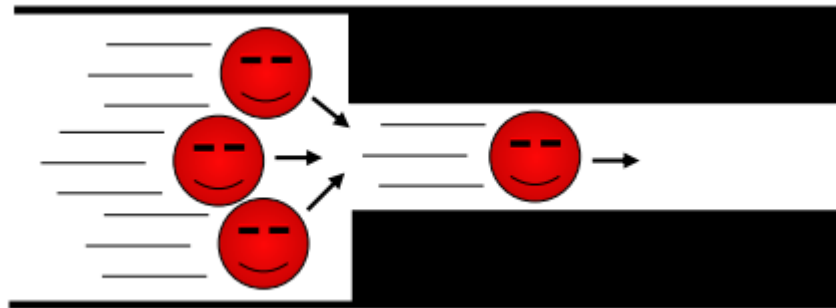


Quanto maior a corrente, maior a “quantidade” de elétrons

Conceitos Básicos Eletricidade

Resistência Elétrica (r - ohm)

O ohm (símbolo: Ω) é a unidade de medida da resistência elétrica

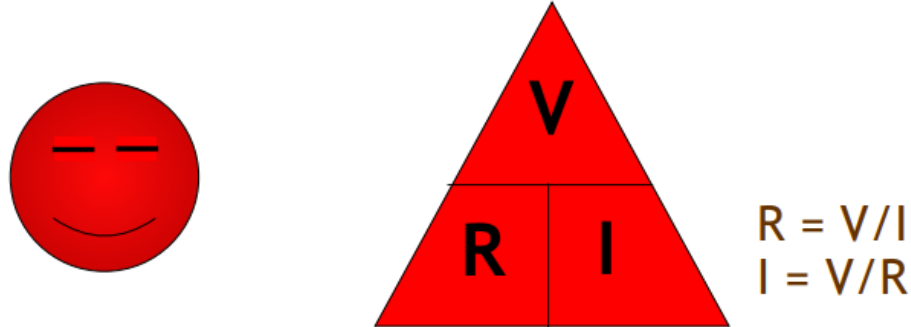


Propriedade do material condutor em reduzir a passagem dos elétrons.

Conceitos Básicos Eletricidade

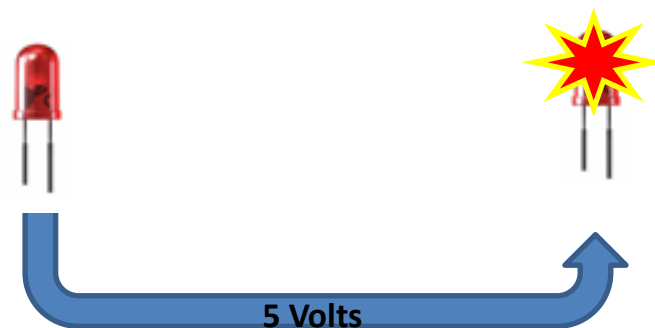
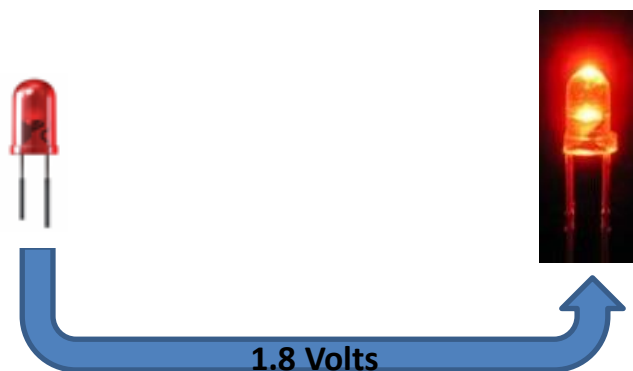
Lei de Ohm

$$V = R \times I$$



A diferença de potencial (V) entre dois pontos de um condutor é proporcional à corrente elétrica (I) que o percorre e à sua resistência (R)

Led – Díodo Emissor de Luz



Led – Díodo Emissor de Luz

Resistência – qual utilizar?

- Tensão de saída de uma porta digital do Arduino = 5v
- Tensão de funcionamento do led vermelho = 1,8v
- Corrente máxima do led = 0,02 A (20mA)

Lei de Ohm:
$$R = \frac{V}{I} = \frac{5V - 1,8V}{0,02} = 160\Omega$$



Não se deve utilizar uma resistência MENOR do que o valor recomendado, pois o led pode queimar.

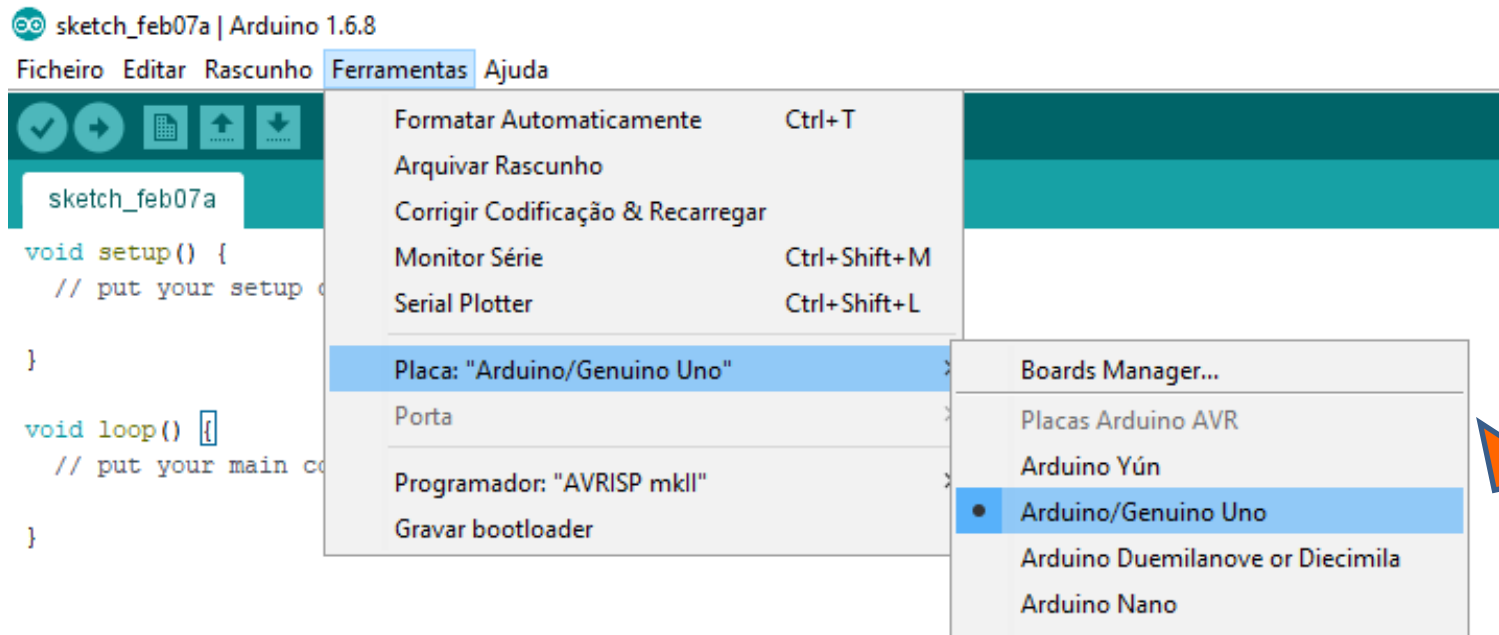
Caso se utilize um resistência MUITO elevada, o led não acenderá.

Ambiente de Desenvolvimento Integrado (IDE)



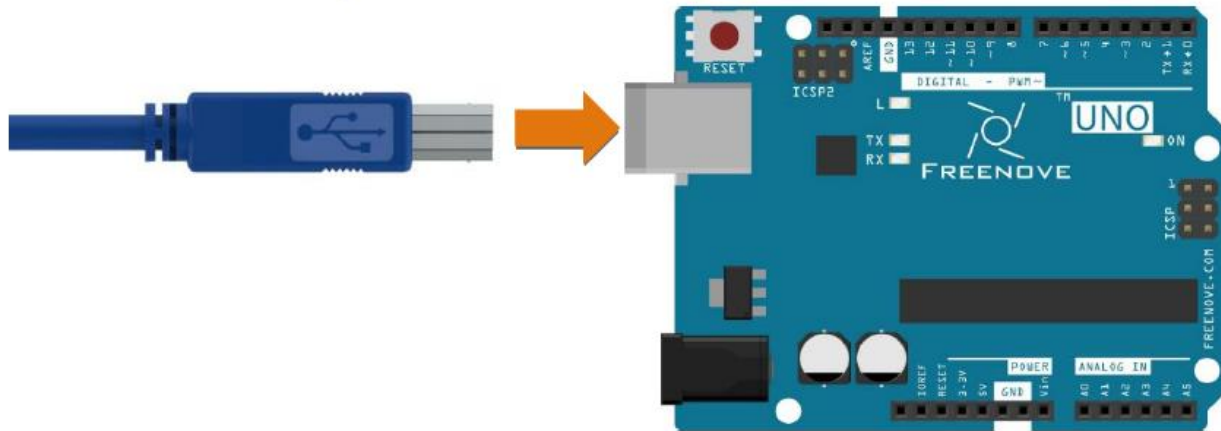
Ambiente de Desenvolvimento Integrado (IDE)

➡ 1º - Selecionar a placa



Ambiente de Desenvolvimento Integrado (IDE)

➡ 2º - Ligar o cabo USB ao Arduino após verificação das ligações



Ambiente de Desenvolvimento Integrado (IDE)



3º - Selecionar a porta

sketch_feb07a | Arduino 1.6.8

Ficheiro Editar Rascunho Ferramentas Ajuda

Verificar código

Enviar código

sketch_feb07a

```
void setup() {  
  // put your setup code here
```

```
}
```

```
void loop() {  
  // put your main code here
```

```
}
```

Formatar Automaticamente Ctrl+T

Arquivar Rascunho

Corrigir Codificação & Recarregar

Monitor Série Ctrl+Shift+M

Serial Plotter Ctrl+Shift+L

Placa: "Arduino/Genuino Uno" >

Port >

Programmer: "AVRISP mkII" >

Gravar bootloader

Serial ports

COM3 (Arduino/Genuino Uno)

Programação - Funções

setup()

Esta função é executada quando o Arduino é ligado. Serve para definir configurações iniciais de portas, comunicações, etc...

loop()

Esta função entra em execução logo após a função setup. O código que está dentro desta função é executado de forma repetitiva, indefinidamente.

delay(parâmetro)

Esta função permite colocar um intervalo em milissegundos entre uma instrução e outra. Recebe como parâmetro um valor inteiro

Programação - Funções

`pinMode(parâmetro 1, parâmetro 2)`

Configura uma porta digital, podendo ela ser de entrada(INPUT) ou saída (OUTPUT). Recebe no parâmetro 1 o número da porta digital, e no parâmetro 2 é definido se a porta é de entrada ou saída.

`digitalWrite(parâmetro 1,parâmetro 2)`

Escreve na porta digital selecionada. Recebe no parâmetro 1 o número da porta digital, e no parâmetro 2 recebe um valor: ligado(HIGH – 5V) ou desligado(LOW- 0V)

Programação - Funções

`analogWrite(parâmetro 1, parâmetro2)`

Escreve numa porta digital PWM. Recebe no parâmetro 1 o número da porta digital PWM e no parâmetro2 recebe um valor entre 0 e 255.

`digitalRead(parâmetro)`


Efetua leituras na porta digital selecionada. Recebe apenas um parâmetro com o número da porta digital. Valores recebidos: 0 ou 1


`analogRead(parâmetro)`

Efetua leituras na porta analógica selecionada. Recebe apenas um parâmetro com o número da porta analógica. Valores recebidos: Entre 0 e 1023

Atividade 1 – Acender um led

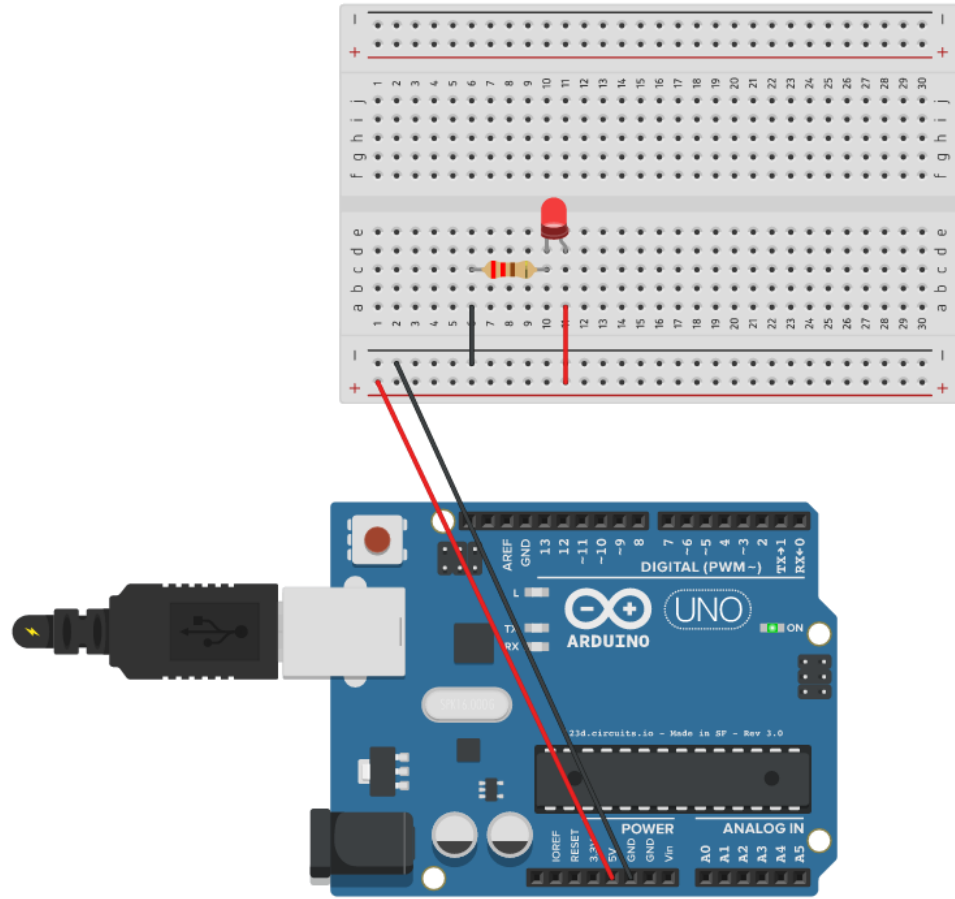
Material necessário:

Led  (Output Digital)

Resistência 

Fios 

Atenção:



Sessão 1



The End