

[PÁGINA INICIAL](#)[MATERIAL DIDÁTICO](#)[PROJETOS](#)[COMPONENTES](#)[EQUIPE](#)[CONTATO](#)[DOCUMENTOS](#)

[Arduino](#)

[Placa Arduino MEGA 2560](#)[Placa Arduino UNO](#)[Alimentação do Arduino](#)

[Portas](#)

[Portas Digitais](#)[Portas Analógicas](#)[Portas PWM](#)[Comunicação Serial](#)[Biblioteca](#)

Arduino

Arduino é uma plataforma de prototipagem baseada em um microcontrolador (Figura MD1) de placa única, com suporte de entrada/saída embutido (embarcado), projetado para que o processo de utilização da eletrônica seja mais acessível. O *hardware* consiste em um dispositivo desenvolvido pela empresa Atmel Corporation projetado para um microcontrolador Atmel com possibilidade de comunicação diretamente com qualquer computador através de interfaces.

A empresa Atmel Corporation foi adquirida em 2016 pela empresa Microchip Technology Inc. A Microchip comercializa sistemas embarcados programáveis.

Figura MD1 - Microcontrolador



O Microcontrolador é um pequeno computador num único circuito integrado, contendo um núcleo de processador, memória para armazenar programas e periféricos programáveis de entrada e saída.

O Arduino é *open source*, isso significa que os usuários podem criar e modificar sua estrutura sem preocupação com questões de direitos autorais e nem de licenciamento, e com uma grande vantagem que é a facilidade de utilização.

O projeto Arduino foi idealizado por um grupo de cinco pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis, na cidade de Ivrea na Itália em 2005, no *Interaction Design Institute* (Instituto de Design e Interação), com o objetivo de utilizar em projetos escolares. A ideia era facilitar o trabalho dos designers e demais

profissionais que desejavam incorporar recursos tecnológicos nos projetos, sem que eles tivessem conhecimentos de eletrônica e que seus custos fossem menores frente aos sistemas de prototipagem disponíveis na época.

No mercado existem diversos modelos de placas com diferentes configurações de memória interna, quantidade de portas, tamanhos e valores variados, como exemplo é possível citar as seguintes placas: Arduino UNO (Figura MD2), Arduino MEGA 2560 (Figura MD2), Arduino Leonardo, Arduino Due, Arduino Nano. Em geral, a escolha por uma determinada placa demanda da estrutura do projeto que será executado. As placas possuem:

- Microprocessador (responsável pelos cálculos e tomada de decisão);
- Memória RAM (utilizada para guardar dados e instruções, volátil);
- Memória flash (utilizada para guardar o Software, não volátil);
- Temporizadores (timers);
- Contadores;
- Clock, etc.

Detalhes dos diversos modelos disponível em:

<https://www.arduino.cc/en/Main/Products>

Para o desenvolvimento de nossos projetos, utilizaremos o Arduino UNO e o Arduino MEGA 2560, esse último possui mais portas. O Arduino MEGA 2560 é o mais indicado para projetos maiores e mais complexos, sua placa permite controlar um maior número de dispositivos em comparação à placa UNO.

Figura MD2 - Placas Arduino

Arduino UNO



Arduino MEGA 2560



[Clique na imagem para ampliar](#)

O Arduino é a plataforma mais utilizada na área da Educação para o desenvolvimento de projetos de robótica. Os projetos desenvolvidos em Arduino podem melhorar a vida de muitas pessoas, como: sistemas para acessibilidades, seguranças, agricultura, pecuária, dentre outros.

A filosofia de fonte aberta da plataforma Arduino promove o surgimento de diversas comunidades na internet, que disponibilizam gratuitamente informações através de fóruns e materiais didáticos (livros, apostilas e vídeos) sobre a ferramenta, facilitando a pesquisa e promovendo discussões sobre os futuros projetos que serão implementados pelos usuários. Como exemplo: o website de divulgação do Arduino (<https://www.arduino.cc/>), um site com tutoriais (<http://playground.arduino.cc/>) com diversos projetos e; um fórum (<http://forum.arduino.cc/>) para sanar as dúvidas dos usuários.

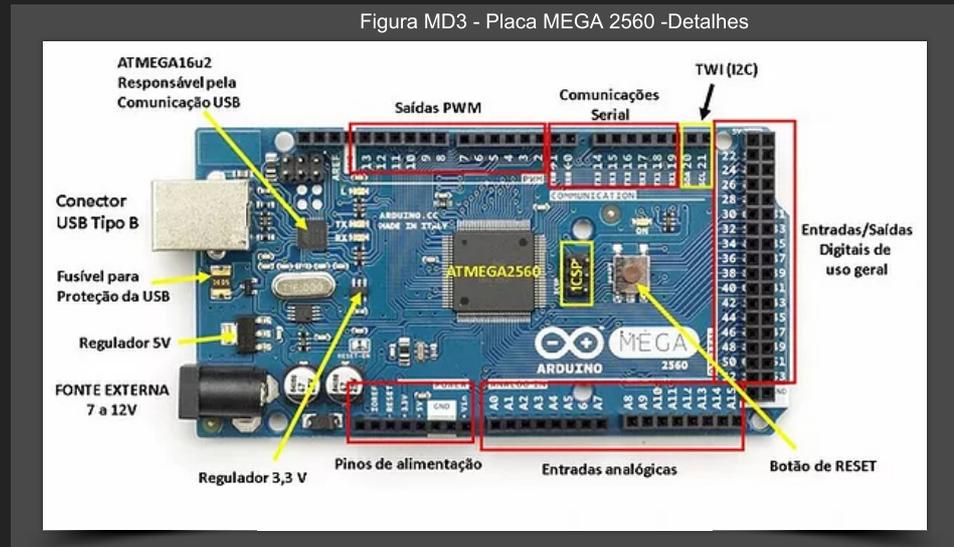
Placa Arduino MEGA 2560

O Arduino MEGA 2560 possui as seguintes especificações:

- Microcontrolador: ATmega2560;
- Voltagem: 5V;
- Entrada de Voltagem (recomendada): 7-12V;
- Entrada de Voltagem (limite): 6-20V;
- Portas Digitais: 54 (15 portas PWM);
- Portas Analógicas: 16 (A0 até A15);
- Memória Flash: 256KB (8KB usado no bootloader*);
- SRAM: 8K (local das variáveis);
- EEPROM: 4K (armazenar dados mesmo sem alimentação);
- Velocidade de CLOCK: 16 MHz;
- Peso: 37g.

Bootloader é nome dado ao gerenciador de inicialização do Arduino. Executado pelo microcontrolador após um Reset (Boot).

Figura MD3 - Placa MEGA 2560 - Detalhes



Placa Arduino UNO

O Arduino UNO possui as seguintes especificações:

- Microcontrolador: ATmega328;
- Voltagem: 5V;
- Entrada de Voltagem (recomendada): 7-12V;
- Entrada de Voltagem (limite): 6-20V;
- Portas Digitais: 14 (6 portas PWM);
- Portas Analógicas: 6 (A0 até A5);
- Memória Flash: 32KB (0,5KB usado no bootloader);
- SRAM: 2K;
- EEPROM: 1K;
- Velocidade de CLOCK: 16 MHz;
- Peso: 25g.

Figura MD4 - Placa UNO - Detalhes





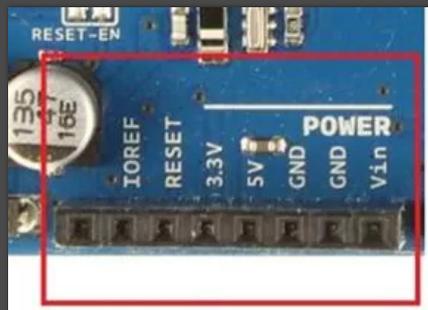
Alimentação do Arduino

Quando conectado via USB (sigla de *Universal Serial Bus*, em português, Porta Universal), o circuito interno do Arduino é alimentado com uma tensão contínua de 5V. Essa porta, além da função de proporcionar energia, permite também a comunicação do Arduino com o computador, assim, podemos enviar e receber dados. Isso é possível com a instalação de driver que emula uma porta serial, através da porta USB.

Caso necessário, é possível alimentar a placa utilizando uma fonte externa ou uma bateria, com saída entre 7V e 12V contínua. Valores inferiores de alimentação podem causar instabilidade nos resultados de processamento do Arduino e valores superiores podem danificar a placa. A fonte pode ser conectada com um plug de 2,1 mm (centro positivo) no conector de alimentação.

A seguir são exibidos os conectores de alimentação (Figura MD5) para conexão de Shields e módulos na placa Arduino MEGA:

Figura MD5 - Alimentação da Placa MEGA



- IOREF - Fornece uma tensão de referência para que shields possam selecionar o tipo de interface apropriada, dessa forma shields que funcionam com a placas Arduino que são alimentadas com 3,3V podem ser adaptar para ser utilizados em 5V e vice-versa;
- RESET – Pino para ser utilizado para um reset (reinicialização) externo da placa Arduino;
- 3,3V - Fornece tensão de 3,3V para alimentação;
- 5V - Fornece tensão de 5V para alimentação;
- GND - Pino Ground (terra);
- VIN - Pino para alimentar a placa externamente.

Portas

No Arduino existem dois tipos de portas: digitais (binárias e PWM) e analógicas. Cada porta tem uma finalidade específica.

Portas Digitais

As portas digitais são utilizadas para trabalhar com dois valores binários de tensão: 0V e 5V. Através destas portas podemos mandar e receber dados. Normalmente nas placas temos um total de 14 portas digitais, numeradas de 0 a 13, sendo que as portas 0 e 1 são para os LEDs RX e TX, respectivamente - os LEDs TX acende quando transmitem dados

e o RX acende quando recebem dados pela porta serial. As portas 0 e 1 estão conectadas diretamente ao sistema de comunicação do Arduino. Os pinos TX (*transmitter*) e RX (*Receiver*) são para transmissão e recepção de dados, respectivamente.

Alguns pinos têm funções específicas: SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estes pinos suportam comunicação SPI (*Serial Peripheral Interface*), que permitem comunicação com diversos componentes, formando uma rede.

As principais funções para manipular as portas digitais são:

- `digitalRead(pino)` – Lê um valor da porta e retorna o valor HIGH quando está em 5V e LOW quando está em 0V;
- `digitalWrite(pino,estado)` – Utilizada para atribuir os valores: HIGH (5V) ou LOW (0V).

Portas Analógicas

As portas analógicas (Figura MD6) podem ler qualquer valor, com precisão, entre 0V e 5V. Entretanto, somente pode ler e não enviar. As portas analógicas são representadas pela letra A antes do número, exemplo: A4. Sua leitura vai de 0 a 1023 (210 = 1024 valores). O 0 representa 0V e o 1023 o 5V. Geralmente essas portas são utilizadas na leitura dos valores de um sensor.

Figura MD6 - Portas Analógicas



Portas PWM

As portas PWM do inglês Pulse Width Modulation (Modulação por Largura de Pulso) se diferenciam das portas digitais, pois podem trabalhar não apenas com as tensões 0V e 5V, mas com uma escala que vai de 0 até 255 (0 = 0V e 255 = 5V). Assim, estas portas permitem obter resultados analógicos através de portas digitais.

Com as portas PWM, podemos: controlar a velocidade de motores; controlar a luminosidade dos leds; gerar sinais analógicos e de áudio, etc.

Na placa Arduino Mega 2560, possuem 15 saídas PWM. Em algumas placas (Figura MD7) as portas PWM são indicadas pelo carácter '~' na frente de seu número.

Figura MD7 - Portas PWM



Comunicação Serial

A comunicação com o Arduino é realizada através da comunicação serial. A porta USB (*Universal Serial Bus*) é utilizada para fazer *upload* de dados no Arduino. Através desta porta é possível também enviar sinais e comandos para o Arduino.

As principais funções são:

- `Serial.begin(velocidade)`
Esta função informa para o Arduino que iremos iniciar a interface serial. O parâmetro velocidade serve para indicar o valor da taxa de transferência. Como padrão utiliza-se o valor 9600 como taxa.
- `Serial.print("mensagem")`
Envia uma mensagem para o monitor serial. Essa mensagem pode ser o estado ou valor de um componente.
- `Serial.available()`
Retorna o número de bytes lidos pela porta serial. Essa função permite identificar algum valor enviado ou recebido.
- `Serial.read()`
Essa função lê os dados que digitamos pelo teclado e envia para o Arduino.
- `Serial.flush()`
Esvazia o buffer de entrada da porta serial. Isto é, qualquer chamada à `Serial.read()` ou `Serial.available()` somente retornarão dados recebidos após a última chamada à `Serial.flush()`.

Biblioteca

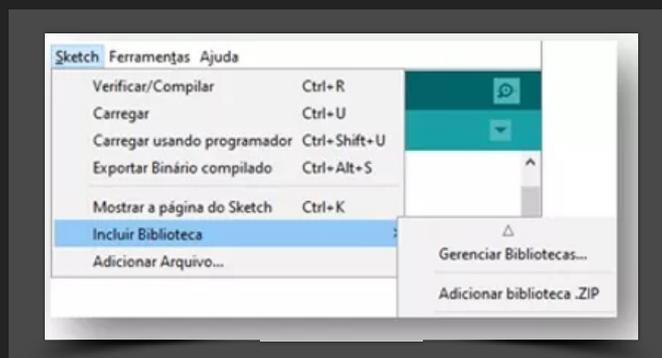
Biblioteca são pequenos programas utilizados no desenvolvimento dos programas. Contém código e dados auxiliares para que esses programas possam executar os dispositivos eletrônicos.

A IDE do Arduino já possui algumas bibliotecas padrões para as funções mais básicas, mas caso queira utilizar alguma outra, utilize o comando `#include <Biblioteca.h>`.

Exemplo:

```
#include <LiquidCrystal.h>
```

Observação: Para importar uma biblioteca externa, utilize a opção "Sketch> Incluir Biblioteca".



[Voltar ao Topo](#) ^

CONTATOS



e-mail: euderfs@gmail.com