

# Comunicação serial

## Comunicação serial

A comunicação serial é um dos meios mais simples e eficientes para enviar e receber dados entre o microcontrolador e outros dispositivos, como computadores ou módulos externos. No Arduino, a biblioteca `Serial` facilita esse tipo de comunicação utilizando o protocolo UART.

## Como Funciona

- **Velocidade de Comunicação:** Definida em bauds (ex.: 9600, 115200), representa o número de bits transferidos por segundo.
- **Transmissão e Recepção:** Usa dois pinos — TX (transmissão) e RX (recepção) ou comunicação direta com o computador na IDE por cabo.
- **Formato de Dados:** Tipicamente, cada pacote de dados inclui 8 bits, 1 bit de parada e nenhum bit de paridade.

## Configuração inicial

Antes de utilizar a comunicação serial, é necessário inicializá-la na função `setup` com o comando:

```
Serial.begin(9600);
```

## Envio e recebimento de dados

Para enviar dados do microcontrolador, utiliza-se o comando:

```
Serial.print("Mensagem"); // Envia uma mensagem sem pular linha  
Serial.println("Mensagem"); // Envia uma mensagem e adiciona uma nova linha
```

Para receber dados, é possível verificar se há informações disponíveis com:

```
if (Serial.available() > 0) {  
    char dado = Serial.read(); // Lê um byte disponível  
    Serial.println(dado); // Ecoa o dado recebido  
}
```

# Exemplo completo

```
void setup() {  
    Serial.begin(9600); // Inicializa a comunicação serial  
    Serial.println("Iniciando comunicação serial...");  
}  
  
void loop() {  
    if (Serial.available() > 0) {  
        char recebido = Serial.read();  
        Serial.print("Você enviou: ");  
        Serial.println(recebido);  
    }  
    delay(100); // Pequena pausa para evitar sobrecarga  
}
```

## Aplicações comuns

- Depuração e monitoramento de código.
- Comunicação com módulos Bluetooth, Wi-Fi, ou GPS.
- Envio de comandos para o microcontrolador a partir de um computador ou outro dispositivo.

Compreender a comunicação serial é essencial para criar projetos interativos e para monitorar o comportamento do microcontrolador durante a execução de um programa.

---

Revision #2

Created 2025-02-11 04:09:16 UTC by Arthur Yuji M. Mori

Updated 2025-02-28 00:52:47 UTC by OBSAT MCTI