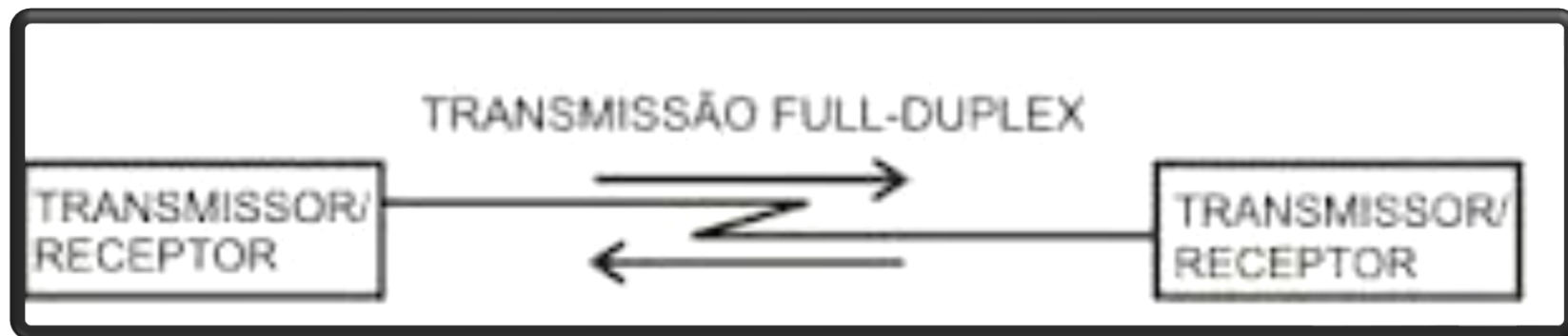


Comunicação Serial

Comunicação bit-a-bit refere-se a um tipo de comunicação em que os dados são transmitidos ou recebidos de forma sequencial, bit por bit, em oposição à comunicação paralela, onde vários bits são transmitidos simultaneamente por vários canais. Na comunicação serial, como a utilizada no Arduino através da USB, os dados são enviados ou recebidos bit-a-bit ao longo de um único canal de comunicação.

A comunicação serial no Arduino feita através da USB é do tipo full-duplex assíncrona.



Programação Arduino

1.Full-duplex: Isso significa que a comunicação pode ocorrer nos dois sentidos simultaneamente. Ou seja, o Arduino pode enviar dados para outro dispositivo e receber dados desse dispositivo ao mesmo tempo. Isso é diferente da comunicação half-duplex, onde os dados são transmitidos em apenas uma direção por vez.

2.Assíncrona: Neste tipo de comunicação, não há um clock global compartilhado entre os dispositivos para sincronizar a transmissão e recepção dos dados.

Em vez disso, é usado um método de sinalização de início e fim de cada caractere (byte) enviado. Por exemplo, no protocolo assíncrono UART (Universal Asynchronous Receiver-Transmitter), amplamente utilizado em comunicações seriais.

Programação Arduino

Essa combinação de full-duplex e assíncrona é comumente vista em comunicações seriais como a realizada pelo Arduino através da porta USB.

Por exemplo, ao conectar o Arduino a um computador via USB, você pode enviar comandos para o Arduino (como leituras de sensores, ativação de saídas, etc.) e também receber dados do Arduino (como respostas, leituras de sensores, etc.) ao mesmo tempo, tornando a comunicação bidirecional possível.

E a comunicação é assíncrona porque não é necessário um sinal de clock global para sincronizar a transmissão e recepção de dados, sendo utilizado um método de sinalização de início e fim de cada byte transmitido.

Utilizando o Monitor Serial

Existem dois tipos de taxa de transmissão importantes:

Bit Rate, ou taxa de bits, é uma medida que indica a quantidade de dados que podem ser transmitidos em um canal de comunicação em um determinado período de tempo. Essa medida é expressa em bits por segundo (bps), indicando quantos bits de informação podem ser enviados ou recebidos em um segundo.

Em termos simples, o Bit Rate é como a velocidade da comunicação de dados. Quanto maior o Bit Rate, mais rápido os dados podem ser transmitidos. Isso é crucial em tecnologias de comunicação, como a internet, telefonia celular, transmissão de vídeo e áudio, entre outras.

Utilizando Monitor Serial

Por exemplo, se você tem uma conexão de internet com uma taxa de bits de 10 megabits por segundo (10 Mbps), isso significa que a sua conexão pode transmitir até 10 milhões de bits de dados em um segundo.

Quanto maior for o Bit Rate, mais rápida será a transferência de dados e melhor será a qualidade da transmissão de informações, como em vídeos de alta definição ou em chamadas de vídeo sem interrupções.

Programação Arduino

O *Baud Rate* é uma medida que indica a taxa de alteração de um sinal em um canal de comunicação em um segundo.

Em termos simples, ele mede quantas vezes um sinal pode mudar de estado em um segundo. A unidade de medida para o Baud Rate é o baud por segundo.

É importante destacar que o Baud Rate difere do Bit Rate, embora muitas vezes sejam confundidos.

Enquanto o Bit Rate mede a quantidade de bits de dados transmitidos em um segundo, o Baud Rate mede a taxa de alteração do sinal.

Para entender melhor, imagine um sinal digital que pode ter dois estados: 0 ou 1.

O Baud Rate determina quantas vezes esse sinal pode mudar de estado em um segundo.

Programação Arduino

Por exemplo, se o Baud Rate é de 1000 bauds por segundo, isso significa que o sinal pode alterar seu estado mil vezes em um segundo.

O Baud Rate é especialmente relevante em comunicações seriais, como em conexões de dispositivos através de portas seriais (RS-232) ou em comunicações via modems.



portas serial (RS-232)

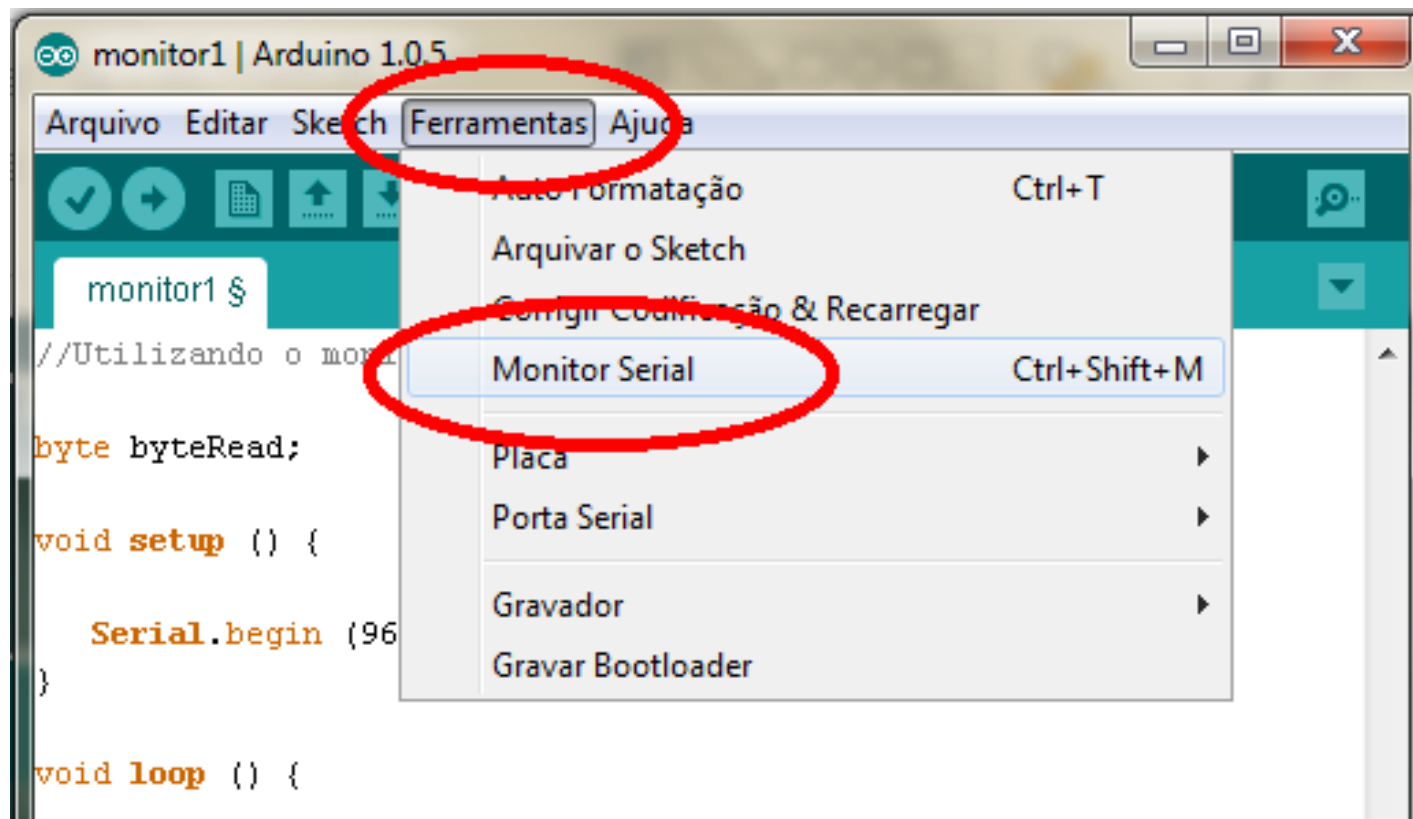
Programação Arduino

É utilizado para determinar a velocidade máxima com a qual os sinais podem ser enviados e recebidos sem erros de comunicação.

É importante notar que o Baud Rate nem sempre é igual ao Bit Rate. Em certos tipos de comunicação, como modems analógicos, vários bits podem ser representados por cada sinal de baud, o que significa que o Bit Rate pode ser maior do que o Baud Rate.

Por exemplo, um modem com Baud Rate de 2400 bauds por segundo pode ter um Bit Rate de 9600 bits por segundo se cada sinal de baud representar quatro bits de dados.

Visualizar Comunicação Serial



Programação Arduino

Serial.begin(9600); Função utilizada para configurar a comunicação serial. “em um Arduino”. A taxa de comunicação Serial [Arduino x PC], padrão é de 9600 bps.

Esta função é sempre escrita dentro do setup.

Serial.available(); Esta função fica lendo constantemente a porta serial do Arduino.

Ela retornará 0 quando nada estiver sendo mandado e um valor maior do que 0 se algo esta chegando da Serial.

Programação Arduino

Serial.read(); Lê o primeiro byte que está no buffer.

Quando retornar -1, significa que não existem dados a serem resgatados.

Serial.write(); Esta função é o oposto de Serial.read(); - escreve na Serial, ou seja, envia algo para o computador e se o computador estiver disponível, receberá a informação.

Serial.println(); Esta função escreve o caractere ASCII

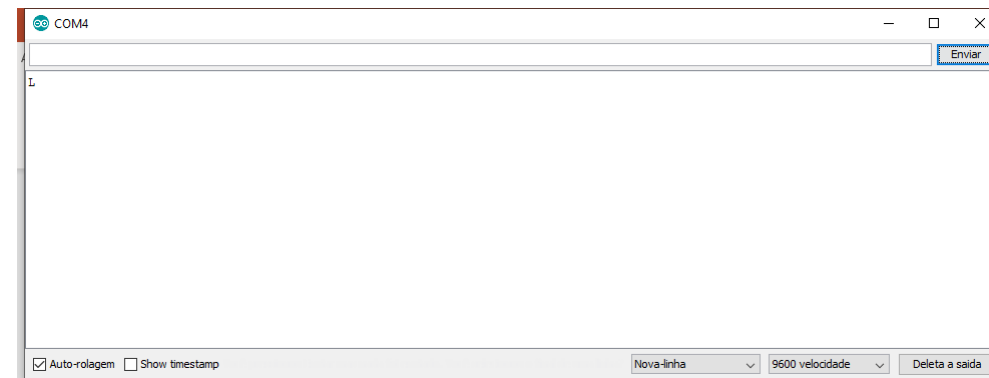
Exemplo

```
byte valordigitado;|
void setup() {
    Serial.begin(9600);

}

void loop() {
    if(Serial.available())
    {
        valordigitado = Serial.read();
        Serial.write(valordigitado);
    }
}
```

Ao executar este código, o que for digitado através do teclado será mostrado no monitor serial.



Programação Arduino

```
int led=13;
char tecla;
void setup() {
    Serial.begin(9600);
    pinMode(led, OUTPUT);
}

void loop() {
    tecla = Serial.read();
    if(tecla=='L')
    {
        digitalWrite(led,HIGH);
        Serial.write(tecla);
    }

    delay(1000);
}
```

Controlar um led
com teclado

Exercícios

1 – Ligar três led's na posição 9,8, 10 quando pressionar uma letra deverá acender o led na posição 9 e 10.

2 – Ligar três led's na posição 8, 9, 10 quando pressionar a letra L deverá acender todos os led's da posição 8, 9 e 10 e quando pressionar a letra D deverá desligar na posição contrária que foi ligado os led's 8, 9 e 10.

3 – Ligar 3 led's na posição 8, 9, 10 quando pressionar a letra K deverá acender todos os led's da posição 8, 9 e 10 quando pressionar a letra P deverá realizar um efeito de pisca pisca nos led's por 10 segundos.

Obs. Entregar o print da tela toda com o esquema e print dos códigos.