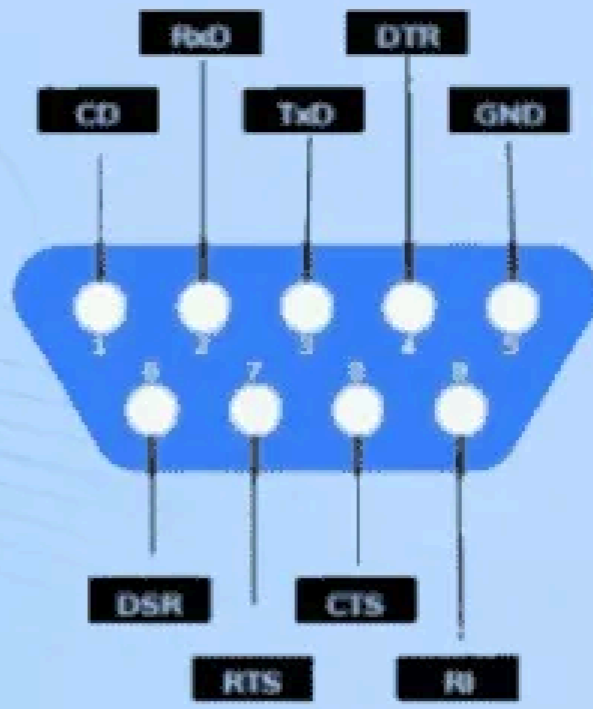


RS232 Serial Port: 7 Facts You Need to Know before Using



概述： RJCP.DLL.SerialPortStream库为C#串口通信提供强大支持，跨平台、高度可定制，集成基础功能如打开、配置串口和数据读写，以及高级功能包括事件处理、超时设置等，极大简化串口开发流程，适用于不同平台和复杂通信需求。

C#串口开发之RJCP.DLL.SerialPortStream库介绍

用途与优点

RJCP.DLL.SerialPortStream库用于C#中串口通信的开发，具有以下优点：

- 跨平台支持：** 适用于Windows、Linux和macOS等多个平台。
- 高度可定制性：** 提供灵活的配置选项，可满足不同串口通信需求。
- 开源社区支持：** 在GitHub上积极维护，可获取最新版本和社区支持。

基础功能

1. 创建 SerialPortStream 实例

```
1 using RJCP.IO.Ports;
2
3 SerialPortStream serialPort = new SerialPortStream("COM1");
```

2. 打开和关闭串口

```
1 serialPort.Open();
2 // 进行串口操作
3 serialPort.Close();
```

3. 配置串口参数

```
1 serialPort.BaudRate = 9600;
2 serialPort.Parity = Parity.None;
3 serialPort.DataBits = 8;
4 serialPort.StopBits = StopBits.One;
```

4. 读取和写入数据

```
1 // 读取数据
2 byte[] buffer = new byte[1024];
3 int bytesRead = serialPort.Read(buffer, 0, buffer.Length);
4
5 // 写入数据
6 byte[] dataToSend = Encoding.UTF8.GetBytes("Hello, Serial!");
7 serialPort.Write(dataToSend, 0, dataToSend.Length);
```

高级功能

1. 事件处理

```
1 // 添加数据接收事件处理程序
2 serialPort.DataReceived += (sender, e) => {
3     byte[] receivedData = new byte[serialPort.BytesToRead];
4     serialPort.Read(receivedData, 0, receivedData.Length);
5     Console.WriteLine($"Received: {Encoding.UTF8.GetString(receivedData)}");
6 };
```

2. 超时设置

```
1 // 设置读取超时时间为100毫秒
2 serialPort.ReadTimeout = 100;
```

3. 自定义流控制

```
1 // 自定义流控制为RTS/CTS
2 serialPort.Handshake = Handshake.RequestToSend;
```

完整示例

以下是一个结合基础和高级功能的完整示例：

```
1 using System;
2 using System.Text;
3 using RJCP.IO.Ports;
4
5 class Program {
6     static void Main() {
7         using (SerialPortStream serialPort = new SerialPortStream("COM1")) {
8             // 基础功能：打开串口、配置参数
9             serialPort.Open();
10            serialPort.BaudRate = 9600;
11            serialPort.Parity = Parity.None;
12            serialPort.DataBits = 8;
13            serialPort.StopBits = StopBits.One;
14
15            // 高级功能：数据接收事件处理
16            serialPort.DataReceived += (sender, e) => {
17                byte[] receivedData = new byte[serialPort.BytesToRead];
18                serialPort.Read(receivedData, 0, receivedData.Length);
19                Console.WriteLine($"Received: {Encoding.UTF8.GetString(receivedData)}");
20            };
21
22            // 发送数据
23            byte[] dataToSend = Encoding.UTF8.GetBytes("Hello, Serial!");
24            serialPort.Write(dataToSend, 0, dataToSend.Length);
25
26            // 等待一段时间以接收数据
27            Console.WriteLine("Waiting for data...");
28            Console.ReadLine();
29        }
30    }
31 }
```





以上示例演示了RJCP.DLL.SerialPortStream库的基础和高级功能，包括打开和配置串口、事件处理、超时设置和流控制等。

源代码获取：公众号回复消息【code: 87745】



重要提示! : 取消关注公众号后将无法再启用回复功能, 不支持解封!

第一步: 微信扫码关注公众号“架构师老卢”

第二步: 在公众号聊天框发送 **code: 87745** , 如:   **code: 87745**   获取下载地址

第三步: 恭喜你, 快去下载你想要的资源吧