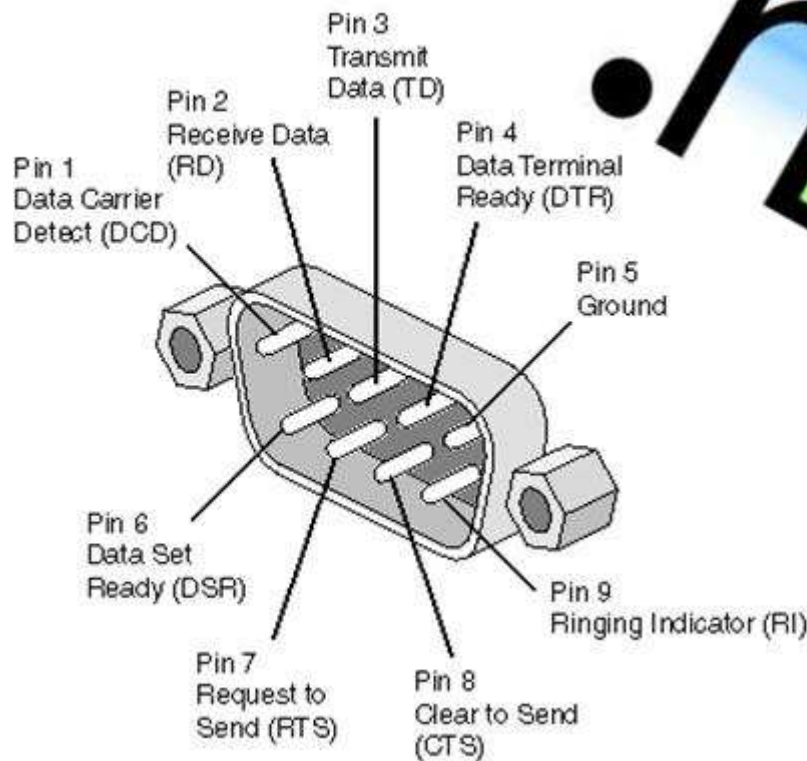


به نام خدا

آموزش استفاده از پورت سریال در دات نت

نویسنده: سید محمد حسینی

تابستان ۸۸



iranled.com

www.iranvig.com

forum.iranled.com

www.iranled.com

این مقاله متعلق به انجمن ایران ویج و سایت iranled است و کپی برداری از آن با ذکر منبع و نویسنده موردی ندارد!.

مدتی بود دنبال یک مقاله بودم که استفاده از پورت سریال (RS232) را در محیط دات نت یاد بگیرم. اما مقاله فارسی که چه عرض کنم حتی خارجی هم پیدا نمی شد! تنها چیزی که پیدا کردم چند مثال و سورس، از msdn و چند سایت دیگر بود. البته مقالاتی در مورد استفاده از mscomm ویژوال بیسیک ۶ هم پیدا شد. ولی دات نت خیلی فرق کرده بود. بالاخره مجبور شدم خودم وقت بذارم و کد بنویسم و به روش آزمایش و خطا، یاد بگیرم. بعد از کلی آزمایش متوجه شدم ریزکاری های مهمی هست که ممکنه دلیل اصلی مشکل خیلی ها با پورت سریال در دات نت همین باشه. خلاصه تصمیم گرفتم اطلاعاتم را هرچند ناچیز بشکل یک مقاله جمع کنم و در اختیار دوستان علاقه مند قرار بدم. امیدوارم مفید باشد.

به امید پیشرفت و سرفرازی ایران عزیز ...

التماس دعا.

به دلیل اینکه اکثر دوستان از vb.net استفاده می‌کنند من آموزش را با سینتکس vb.net می‌نویسم. البته همانطور که می‌دانید کنترل‌های دات نت در vb و ++C و #C فرقی نمی‌کنند و همه از یک کتاب خانه مشترک استفاده می‌کنند و توابع و رویدادها یکسان هستند. فقط سینتکس زبان‌ها فرق می‌کند و هیچ نگرانی در این مورد نیست. من خودم ++C کار می‌کنم و پیشنهاد می‌کنم شما هم از C و ++C به دلیل سازگاری بهتر با میکروکنترلر استفاده کنید.

برای ارتباط با پورت سریال باید کنترل SerialPort را از بخش components پنجره Toolbox به پروژه اضافه کنید. در ابتدا باید خصوصیات ارتباط سریال و فریمبندی اطلاعات را مشخص کنید. این کار را می‌توانید خیلی راحت در قسمت پروپرتی‌های کنترل SerialPort انجام بدین.

PortName : پورت مورد نظر برای برقراری ارتباط. مثل COM1 ، COM2 ،... (معمولا COM1 ولی مال من COM13!)

BaudRate : نرخ ارسال که باید یکی از مقادیر استاندارد مثل ۲۴۰۰ ، ۴۸۰۰ ، ۹۶۰۰ و... باشد. (بطور معمول ۹۶۰۰)

DataBits : تعداد بیت‌های داده که می‌تواند یکی از مقادیر ۵ ، ۶ ، ۷ یا ۸ باشد. (چون معمولا از داده‌های کاراکتری استفاده می‌شود و اطلاعات بایت به بایت ارسال می‌شوند باید ۸ باشد)

Parity : تنظیم بیت پریتهی یا تقارن برای چک کردن دیتا ارسال شده. (معمولا None گذاشته می‌شود)

StopBits : تعداد بیت توقف که این هم معمولا one هست.

البته همانطور که می‌دانید همه این تنظیمات فریمبندی بستگی به طرف دیگه ارتباط سریال دارد و باید یکسان باشد. طرف مقابل می‌تواند میکروکنترلر ، کامپیوتر یا هر سیستم دیگه‌ای با امکان برقراری ارتباط سریال باشد.

برای اینکه ارتباط برقرار شود باید از دستور Open استفاده کنید.

```
SerialPort1.Open()
```

در صورتی که ارتباط با موفقیت برقرار شود تابع IsOpen مقدار True و در غیر اینصورت False برمی‌گرداند.

```
Dim b As Boolean = SerialPort1.IsOpen()
```

برای قطع ارتباط از دستور Close استفاده می‌شود.

```
SerialPort1.Close()
```

ارسال :

Write : این دستور یک رشته به عنوان ورودی می‌گیرد و بصورت کاراکتر به کاراکتر ارسال می‌کند.

```
SerialPort1.Write("forum.iranled.com")
```

WriteLine : این تابع مثل Write عمل می‌کند با این تفاوت که در انتهای رشته کاراکتر خط بعد (تب عمودی یا enter) را هم می‌فرستد.

```
SerialPort1.WriteLine("iranvig")
```

معمولا بجای دستور Write از WriteLine استفاده می‌شود. چون برنامه‌نویسی طرف مقابل که اطلاعات را دریافت می‌کند معمولا بشکلی است که اطلاعات را می‌گیرد تا جایی که به کاراکتر خط بعد برسد و تا آنجا را بشکل یک فرمان مجزای دریافت شده تحلیل می‌کند. دستور بالا در واقع رشته زیر را در زبان C می‌فرستد :

```
"iranvig\n"
```

و تقریبا معادل این دستور در وی بی است :

```
SerialPort1.Write("iranvig" & vbCrLf)
```

ثابت vbCrLf دو کاراکتر به عنوان خط بعد می‌فرستد. به همین دلیل کمی با WriteLine فرق می‌کند. اگر دستور بالا را بعد از دریافت نمایش بدید یک کاراکتر اضافه هم در آخر آن (قبل از خط بعد) دیده می‌شود. بهرحال اگر دو طرف از وی بی استفاده شده است، می‌شود از Write استفاده کرد ولی اگر طرف مقابل میکروکنترلر است و در آن از enter به عنوان پایان دستور استفاده شده، بهتر است از WriteLine استفاده شود.

دریافت :

به محض اینکه اطلاعاتی از پورت سریال برسد رویداد DataReceived فراخوانی می‌شود.

```
Private Sub SerialPort1_DataReceived(ByVal sender As System.Object, ByVal e _  
    As System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
```

```
End Sub
```

تابع این رویداد بخاطر اینکه ممکن است دو طرف با سرعت برابر عمل نکنند، بر روی یک thread با اولویت بالاتر جدا از برنامه اجرا می‌شود. به همین دلیل، درون این تابع شما نمی‌توانید به توابع دیگر یا کنترل‌های برنامه (مثل تکست باکس) دسترسی داشته باشید. در این تابع فقط باید اطلاعات رسیده خوانده شود و پردازش یا نمایش اطلاعات رسیده باید از یک تابع دیگر انجام شود. اگر هم بخواهید از درون این رویداد به thread اصلی برنامه دسترسی داشته باشید باید از روش آدرس‌دهی به توابع یا کنترل‌ها استفاده کنید که این بحث مربوط به این مقاله نمی‌شود و مربوط به برنامه نویسی ویندوز است. (در همین حد می‌گم که ساده ترین روش آدرس‌دهی استفاده از دستور invoke است...) بهرحال می‌توانید از یک تایمر استفاده کنید و متغیر مربوط به اطلاعات رسیده را پردازش کنید.

: ReadByte

این تابع یک بایت (یک کاراکتر) را از ورودی می‌خواند. در واقع هشت بیت ارسال شده را خوانده و بصورت یک عدد دهدهی (دسیمال) برمیگرداند. اگر هشت بیت ارسال شده به عنوان کاراکتر ارسال شده باشند کد اسکی آن دریافت می‌شود.

```
i = SerialPort1.ReadByte() 'i as integer
c = Chr(SerialPort1.ReadByte()) 'c as char or string
```

کد زیر یک کاراکتر را از پورت سریال دریافت می‌کند.

```
Private Sub SerialPort1_DataReceived(ByVal sender As System.Object, ByVal e _
    As System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
    c = Chr(SerialPort1.ReadByte()) 'c as char or string
End Sub
```

تابع ReadChar هم مانند ReadByte عمل می‌کند.

: ReadLine

در صورتی که یک رشته ارسال شده باشد و بخواهیم با ReadByte اطلاعات را بخوانیم هر بار که یک کاراکتر خوانده شد رویداد Recieve دوباره فراخوانی می‌شود و باید کاراکترهای بعد را بخوانیم. در واقع باید یک آرایه از کاراکترها داشته باشیم و آنها را مقداردهی کنیم. بجای این کار می‌شود از تابع ReadLine استفاده کرد. این تابع با یکبار فراخوانی رویداد Receive دیتاهای رسیده را تا انتهای خط می‌خواند و به عنوان یک رشته خروجی می‌دهد.

```
Private Sub SerialPort1_DataReceived(ByVal sender As System.Object, ByVal e _
    As System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
    s = SerialPort1.ReadLine(); 's as string
End Sub
```

نکته مهم در مورد تابع ReadLine این است که حتما باید کاراکتر انتهای خط (کاراکتر خط بعد) هم ارسال شده باشد. در غیر اینصورت منتظر این کاراکتر می‌ماند. یعنی اگر طرف مقابل وی‌بی است باید با دستور WriteLine رشته فرستاده شود. دستورهای ارسال رشته در میکروکنترلرها هم (مانند puts در کدویژن) این کاراکتر ویژه را می‌فرستند و نگرانی در این مورد نیست.

ReadLine یک تابع overload هم دارد. این تابع یک عدد به عنوان ورودی می‌گیرد و از رشته دریافت شده از ورودی، کاراکتر با شماره ورودی تابع را به عنوان خروجی برمی‌گرداند.

```
Private Sub SerialPort1_DataReceived(ByVal sender As System.Object, ByVal e _
    As System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
    ' 'iranvig' was sent
    c = SerialPort1.ReadLine(3) 'returns 'a'
End Sub
```

: ReadTo

این تابع یک رشته به عنوان ورودی دریافت می‌کند و از ورودی تا جایی می‌خواند که به رشته‌ای برابر با رشته ورودی خود برسد.

```
Private Sub SerialPort1_DataReceived(ByVal sender As System.Object, ByVal e _
    As System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
    ' 'iranvig' was sent
    s = SerialPort1.ReadTo("vig") 'returns 'iran' (in first call)
End Sub
```

در استفاده از این تابع باید به این نکته توجه کنید که بعد از خواندن رشته تا مکان مشخص شده، بار دیگر تابع رویداد فراخوانی می‌شود و هنوز ادامه رشته برای خواندن باقی می‌ماند. در واقع تابع به شکلی که در بالا نوشته شده قابل استفاده نیست و باید کدنویسی مناسب با قالب دیتای ارسال شده انجام شود.

(در کمال تواضع اینو بگم یک تابع دیگه هم هست به اسم Read که من هنوز کامل ازش سر در نیاوردم!! ولی معمول ترین و راحتین روش همون ReadLine است)

چند تابع دیگه هم هست که باید بیشتر کار کنید تا یاد بگیرید. مثل BytesToRead که تعداد بایت‌های موجود در بافر ورودی را برمی‌گرداند و DiscardInBuffer که بایت‌های موجود در بافر ورودی را پاک می‌کند.

حرف آخر :

من در زمینه الکترونیک بسیار تازه کار هستم. پس بعید نیست اشکالاتی بر این مقاله وارد باشد. خوشحال می‌شوم هرگونه نقصی در مورد مطالب این مقاله دیدید در فروم ایران ویج مطرح کنید. در صورتی هم که مشکل و سوالی داشتید من و دوستان ایران ویجی در خدمت شما هستیم.

در آخر از دوستان خوبم **t3rip3000** و **ilgar** که با حوصله، من را در تابستان ۸۸ تحمل کردند و با موضوعات مختلف الکترونیک و میکروکنترلر آشنا کردند تشکر می‌کنم. همینطور **ha_60** مدیر ایران ویج و **1nafar** عزیز که از مقالاتش خیلی استفاده کردم و تمام الکترونیکی‌های ایرانویج ممنونم ...