

پورت سریال در MATLAB

در رابطه ارتباط با پورتهای سریال MATLAB برنامه ها و دستورات متعدد و موثری دارد. این دستورات رابطه با پورت سریال را خیلی راحت می کند. نوشتن داده، خواندن داده از پورت و. . . از کارهای عادی برای این نرم افزار است اما بهتر است در مورد سریال پورت و پروتکل RS-232 چیزهای را بدانیم

ارتباط سریال چیست؟

ارتباط سریال پروتکل سطح پایین مشترک برای ارتباط بین دو یا چند وسیله است که به صورت معمول یک وسیله کامپیوتر است و وسیله دیگر می تواند یک مودم، پرینتر، کامپیوتر دیگر و یا یک وسیله علمی از قبیل اسیلوسکوپ و یا فانکشن ژنراتور باشد. به عنوان یک اشاره، سریال پورت بایتهایی از اطلاعات را در روش سریال - یک بیت در زمان - ارسال و دریافت می کند. در این ارتباط، فرستنده و یا راه انداز، بیت ها را به صورت پشت سر هم ارسال می کند. در ارتباط بین دو کامپیوتر ممکن است برای هر جهت خط مستقیمی داشته باشیم یا اینکه بر روی یک خط در هر دو جهت داده ارسال شود. وقتی که بیشتر از دو دستگاه به هم متصل شوند، معمولاً دارای یک مسیر مشترک بین آنها هستیم و پروتکل شبکه است که مشخص می کند کدام دستگاه اجازه ارسال دارد.

اتصال دو دستگاه توسط RS-232

درگاه RS-232 آشنا ترین مدار واسط کامپیوتری است. این واسط بر روی همه ی PC ها قرار دارد و همچنین میکروکنترلرهای مختلفی از این واسط بهره می برند. بیشترین کاربرد RS-232 اتصال یک مودم است، اما از دستگاههای دیگری که از این درگاه استفاده می کنند می توان از چاپگرها و واحدهای اکتساب داده، تجهیزات تست و مدارات کنترلی نام برد. می توان از RS-232 برای اتصال بین دو کامپیوتر از هر نوعی استفاده کرد.

امروزه، واسط های سریعتر و پیشرفته تری نسبت به RS-232 ایجاد شده اند، اما همچنان RS-232 محبوبیت خاصی دارد که از دلایل آن می توان به سخت افزار و برنامه ریزی آسان و راحت و ارزان قیمت بودن آن و همچنین موجود بودن آن بر روی بسیاری از دستگاهها نام برد.

یک پروژه ساده برای شروع کار با پورت سریال :

برای شروع کار با پورت سریال بهتر است با یک آزمایش ساده شروع به کار کنیم فرض کنید می خواهیم

داده ای را روی پورت بنویسیم برای این کار بهتر است که یک پورت نری سریال و یک دیود نورانی را آماده کنید .

حال به این صورت عمل کنید : پایه ۵ سریال را به پایه منفی دیود نوری و پایه ۳ سریال را به پایه دیگر دیود نوری متصل کنید حال پورت سریال را به کامپیوتر متصل کنید و در ادامه دستورات زیر را وارد کنید :

دستور: serial:

شئ پورت سریال را ایجاد می کند. مثلا می توان نوشت:

```
s = serial('COM1');
```

که شئ s را ایجاد می کند.

دستور: fopen:

شئ پورت سریال را به وسیله متصل می کند. یعنی در حقیقت پورت را باز می کند.

```
fopen(s)
```

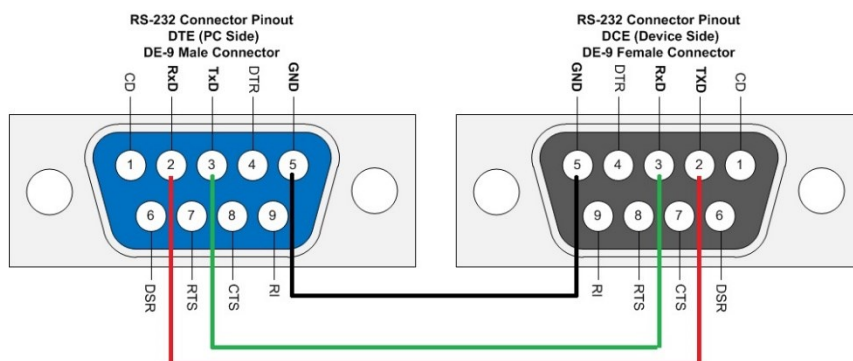
دستور: fwrite:

داده های باینری را در وسیله می نویسد (fwrite(1). در هنگام اجرای این دستور به دیود نوری نگاه کنید که در این هنگام روشن و سپس خاموش می شود.

دستور: fclose:

شئ پورت سریال را از وسیله جدا می کند.

```
fclose(s)
```



مشخصات اجزاء پورت سریال:

مشخصات ارتباطی، خواندن، نوشتن اجزای شئ پورت سریال به طور خلاصه در زیر برحسب الفبا آمده اند:

مشخصات ارتباطی:

- **BaudRate**: سرعتی که بیتها فرستاده می شوند را مشخص می کند. توضیحات: شما BaudRate را بر حسب بیت بر ثانیه پیکربندی می کنید. بیت های ارسالی شامل بیت شروع، بیت های داده، بیت توازن، (اگر به کار روند) و بیت های توقف هستند. هر چند اگر بیت های داده ذخیره شوند BaudRate. سرعتی است که اطلاعات در کانال ارتباطی فرستاده می شود در مفهوم پورت سریال باود ۹۶۰۰ این معنی را می دهد که پورت سریال قادر به ارسال ماکزیمم ۹۶۰۰ بیت در هر ثانیه است. اگر واحد اطلاعاتی یک باود (یک بیت) باشد، baudrate و bitrate با هم یکی هستند اگر یک باود به عنوان ده بیت داده شود، (برای مثال، هشت پالس بیت های داده دو چارچوب بیتهای)، bitrate بهنوز ۹۶۰۰ است ولی baudrate، ۱۰/۹۶۰۰ یعنی ۹۶۰ است. شما همیشه baudrate را بر حسب بیت در هر ثانیه پیکربندی می کنید. بنابراین در مثال بالا baudrate را ۹۶۰۰ تنظیم می کنید. توجه: هم کامپیوتر و هم دستگاه جانبی باید در یک باود تنظیم شوند برای اینکه شما بخواهید خواندن و نوشتن موفق داشته باشید. برای این کار روی آیکون My computer کلیک راست کرده Manage را انتخاب می کنیم در صفحه باز شده Device Manager را انتخاب کرده و از سخت افزار های موجود در سمت راست Ports (Com & Lpt) را انتخاب می کنیم و روی + کلیک می نماییم، روی Communications که مربوط به COM1 است دو بار کلیک کرده و در دیالوگ باز شده در تب Port Setting گزینه Bit per second را روی مقدار مورد نظر تنظیم می کنیم Baudrate. های استاندارد شامل ۱۹۲۰۰، ۳۸۴۰۰، ۵۷۶۰۰، ۱۱۵۲۰۰، ۱۲۸۰۰۰، ۱۱۰، ۳۰۰، ۶۰۰، ۱۲۰۰، ۲۴۰۰، ۴۸۰۰، ۹۶۰۰، ۱۴۴۰۰ و ۲۵۶۰۰۰۰ بیت در هر ثانیه است.
- **Data bits**: تعداد بیت های داده برای ارسال را مشخص می کند. که می توانند پنج، شش، هفت و یا هشت باشند.
- **Parity**: نوع مقابله توازن را مشخص می کند. که خالی، زوج، فرد، علامت و یا فاصله است.
- **Stop bits**: تعداد بیت های بکار رفته شده را برای نمایش انتهای بایت مشخص می کند.
- **Terminator**: نوع کاراکتر پایان رساننده را مشخص می کند. مشخصات نوشتن:
- **Bytes To Output**: شامل تعداد بایت هایی است که در بافر خروجی جریان دارند.
- **Output Buffer Size**: سایز بافر خروجی را بر حسب بایت مشخص می کند.
- **Timeout**: زمان نوشتن برای اینکه عملکرد خواندن و یا نوشتن کامل شود را مشخص می کند.
- **Transfer Status**: اگر یک عملکرد خواندن و یا نوشتن آسنکرون در پروسه باشد را نمایش می دهد.
- **Values Sent**: مجموع تعداد مقادیر نوشته شده در وسیله را نمایش می دهد.

مشخصات خواندن:

- **Bytes Available**: تعداد بایت های قابل دسترسی در بافر ورودی را نمایش می دهد.
- **Input Buffer Size**: سایز بافر ورودی را بر حسب بایت مشخص می کند.
- **Read Async Mode**: مشخص می کند که آیا یک عمل خواندن آسنکرون پیوسته است یا دستی؟
- **Timeout**: در بالا توضیح داده شد.
- **Transfer Status**: در بالا توضیح داده شد.
- **Values Received**: مجموع تعداد مقادیر خوانده شده از وسیله را نمایش می دهد.

توابع پر کاربرد برای کار با پورت سریال:

تابع: fgetl

یک خط متنی از وسیله را می خواند(با رها کردن terminator)

حالات استفاده :

- tline = fgetl(obj)
 - [tline,count] = fgetl(obj)
 - [tline,count,msg] = fgetl(obj)
- Obj:شیء پورت سریال است .
 - Tline:خواندن متن از وسیله ، بدون terminator
 - Count:تعداد مقادیر خوانده شده ، شامل terminator
 - Msg:یک پیغام نمایش داده می شود اگر عمل خواندن با موفقیت نباشد.

تابع: fgets

یک خط متنی از وسیله را می خواند(که شامل terminator می شود).

حالات استفاده:

- tline = fgets(obj)
 - [tline,count] = fgets(obj)
 - [tline,count,msg] = fgets(obj)
- Obj:شیء پورت سریال است .
 - Tline:خواندن متن از وسیله ،با terminator
 - Count:تعداد بایتهای خوانده شده ، شامل terminator
 - Msg:یک پیغام نمایش داده می شود اگر عمل خواندن با موفقیت نباشد.

تابع: fprintf

داده متنی را در وسیله می نویسد.

حالات استفاده:

- fprintf(obj,'cmd')
 - fprintf(obj,'format','cmd')
 - fprintf(obj,'cmd','mode')
 - fprintf(obj,'format','cmd','mode')
- obj:شیء پورت سریال است .
 - Cmd:رشته نوشته شده در وسیله
 - Format:تعیین تبدیل در زبان C

- Mode: مشخص می کند که داده سنکرون یا آسنکرون نوشته شود .

تابع : fread

داده باینری را از وسیله می خواند .

حالات استفاده :

- $A = \text{fread}(\text{obj}, \text{size})$
- $A = \text{fread}(\text{obj}, \text{size}, \text{'precision'})$
- $[A, \text{count}] = \text{fread}(\dots)$
- $[A, \text{count}, \text{msg}] = \text{fread}(\dots)$

- Obj: شیء پورت سریال است .

- Size: تعداد مقادیر برای خواندن

- Precision: تعداد بیت های خوانده شده برای هر مقدار، و تفسیر

بیتها به عنوان کاراکتر ، صحیح و یا مقادیر با ممیز شناور

- A: داده های باینری بازگشت داده شده به وسیله

- Count: تعداد مقادیر خوانده شده

- Msg: یک پیغام نمایش داده می شود اگر عمل خواندن با موفقیت

نباشد.

تابع : fscanf

داده را از وسیله می خواند ، و فرمت متنی است .

- $A = \text{fscanf}(\text{obj}, \text{formatSpec})$
- $A = \text{fscanf}(\text{obj}, \text{formatSpec}, \text{sizeA})$
- $[A, \text{count}] = \text{fscanf}(___)$

- Obj: شیء پورت سریال است .

- Format: تعیین تبدیل در زبان C

- Size: تعداد مقادیر برای خواندن

- A: داده های خوانده شده از وسیله که فرمت آن متنی است .

- Count: تعداد مقادیر خوانده شده

- Msg: یک پیغام نمایش داده می شود اگر عمل خواندن با موفقیت

نباشد.

تابع : length

طول آرایه شیء پورت سریال را تعیین می کند .

- $\text{Length}(\text{obj})$

تابع : size

- $d = \text{size}(\text{obj})$
- $[m, n] = \text{size}(\text{obj})$

- $[m1,m2,\dots,mn] = \text{size}(\text{obj})$
- $m = \text{size}(\text{obj},\text{dim})$

obj: شیء پورت سریال است .

Dim: بعد شیء پورت سریال

D: تعداد سطر ها و ستون ها در شیء

M: تعداد سطر ها در شیء ، یا طول بعد مشخص شده

N: تعداد ستون ها در شیء