



با سلام به شما کاربر عزیز ، امیدوارم مطلبی که در پیش رو دارید برایتان مفید باشد. من با توجه به درخواست مکرر دوستان و همچنین نیاز خودم تصمیم گرفتم اشاره ی کوچکی به بحث ماژول های GSM و نحوه ی برقراری ارتباط میکروکنترلرها با آنها داشته باشم.

در این آموزش از ماژول Sim300cz ساخت شرکت SIMCOM استفاده شده که از طریق یک قاب سیم کارت ۸ پین به سیم کارت متصل شده و از طریق پورت سریال RS232 با میکروکنترلر AVR ارتباط برقرار می کند. به همراه این آموزش یک شماتیک بسیار مناسب برای استفاده از این ماژول و نیز فایل PCB مربوطه به همراه کتابخانه ی Footprint ماژول و قاب سیم کارت ارائه می شود.

توجه داشته باشید که این یک آموزش بسیار مختصر و مفید است که برای آشنایی اولیه کاربران با این نوع ماژول ها و نحوه ی راه اندازی آنها تهیه شده است تا ابهامات اولیه ی کاربران رفع شود. انشالله. در تهیه این آموزش از دیتاشیت سخت افزار ورژن ۲۰۰۸ (SIM300C\_HD\_V2.08) و نرم افزار ورژن ۲ (SIM300\_ATC\_V2.00) استفاده شده است.

صفحه	عنوان
۳	ویژگی های ماژول SIM300cz .....
۳	مختصری راجع به AT Command ها .....
۴	تنظیمات Hyper Terminal .....
۵	توضیحات شماتیک و مدار PCB .....
۶	برقراری ارتباط با کامپیوتر .....
۷	مثال های ارتباط با کامپیوتر .....
۱۱	ارتباط با میکروکنترلرهای AVR .....
۱۲	مثال های ارتباط با AVR .....
۱۲	• کد برنامه ارسال پیام کوتاه برای بسکام .....
۱۳	• کد برنامه ارسال پیام کوتاه برای کدویژن .....
۱۴	• کد برنامه دریافت پیام کوتاه برای بسکام .....
۱۶	• کد برنامه دریافت پیام کوتاه برای کدویژن .....

## ویژگی های ماژول SIM300cz :

در ابتدا به برخی از ویژگی های ماژول SIM300cz اشاره می کنیم.

۱. مناسب برای سه باند یا فرکانس متداول
  - باند EGSM با فرکانس ۹۰۰ مگاهرتز
  - باند DCS با فرکانس ۱۸۰۰ مگاهرتز
  - باند PCS با فرکانس ۱۹۰۰ مگاهرتز
۲. پشتیبانی از طرح های CS-1 ، CS-2 ، CS-3 و CS-4 برای GPRS
۳. قابل استفاده به همراه باتری و دارای مدار داخلی برای استفاده از این قابلیت
۴. پشتیبانی از LCDهای SPI و کی پد برای توسعه پروژه ها
۵. قابلیت به روز کردن برنامه ی داخلی ماژول (Firmware) از طریق پورت جداگانه
۶. دارای پورت سریال RS232 کامل (دارای ۷ خط واقعی)
۷. دارای دو کانال مجزای صوتی ، که هر یک می تواند یک میکروفن و یک بلندگو را پشتیبانی کند.
۸. دارای مد Sleep برای کاهش مصرف توان با جریان مصرفی ۲.۵ میلی آمپر
۹. پشتیبانی از پروتکل TPC/IP
۱۰. قابلیت کار با ولتاژهای ۲.۴ ولت تا ۴.۵ ولت
۱۱. پشتیبانی از پیام کوتاه در مدهای Text ، PDU ، MT ، MO و CB
۱۲. محل ذخیره ی پیام کوتاه فقط حافظه ی سیم کارت می باشد.
۱۳. دارای تایمر ساعت/زمان واقعی RTC با قابلیت برنامه ریزی
۱۴. قابلیت تشخیص خودکار Baudrate بدون هیچگونه تنظیمات اولیه! و ....

## مختصری راجع به AT Command ها :

ممکن است بدانید که برای برقراری ارتباط با این ماژول ها از پورت سریال RS232 و نوعی زبان یا دستور خاص موسوم به AT Command استفاده می شود.

در واقع AT Command ها مجموعه ای از دستورات هستند که از طریق ارسال آنها بوسیله ی پورت سریال می توان رفتار ماژول را کنترل کرد. پیشوند At یا at باید در ابتدای همه ی این دستورات نوشته شود. و برای اجرا شدن هر دستور باید یک کاراکتر Enter نیز اعمال شود. شکل کلی فرآیند اجرای یک دستور و نتیجه ی حاصل از آن در این ماژول مطابق صفحه ی بعد است.

<CR><LF><response><CR><LF>

که در آن CR معرف Enter (Carriage Return) و LF معرف عبارت "سر خط" (Line Feed) و response معرف نتیجه ی حاصل از ارسال دستور است. این نتایج معمولا کد **ناخواسته** یا **Unsolicited** نامیده می شوند ، زیرا هیچگونه درخواستی برای دریافت آنها ارسال نشده است. البته باید توجه داشته باشید که کاراکترهای CR و LF هیچکدام قابل مشاهده نیستند.

چند نکته پیرامون استفاده از AT Command ها :

- ❖ برای استفاده از این دستورات پیشوند AT یا at باید همواره ذکر شود. مثلا AT+CMGS
- ❖ در حالت کلی می توان این دستورات را به سه دسته ی (الف) پایه ، (ب) پارامترهای S و (ج) پیشرفته یا گسترش یافته تقسیم کرد.
- ❖ در حالت پیشرفته که شامل برخی از این دستورات می شود می توان هر دستور را در ۴ حالت زیر بکار برد.

حالت	شکل دستور	توضیح
حالت تست	AT+<x>=?	این حالت تمامی پارامترها و رنج تنظیمات ممکن دستور را بر می گرداند. (در هایپر ترمینال نشان می دهد).
حالت خواندن	AT+<x>?	این حالت تنظیمات و پارامترهای جاری دستور را بر می گرداند.
حالت نوشتن	AT+<x>=<Value>	این حالت تنظیمات کاربر را برای دستور لحاظ می کند.
حالت اجرایی	AT+<x>	این حالت پارامترهای ایجاد شده بوسیله ی پروسه های داخلی ماژول را بر می گرداند.

### تنظیمات Hyper Terminal :

در ابتدا برقراری ارتباط بین کامپیوتر و ماژول را مورد بررسی قرار می دهیم. برای این کار باید پورت سریال ماژول را به پورت سریال کامپیوتر متصل کنیم. این ماژول بطور خودکار می تواند میزان Baudrate (نرخ انتقال داده) پورت سریال کامپیوتر را تشخیص دهد. (بشرطی که میزان Baudrate بین ۱۲۰۰ تا ۱۱۵۲۰۰ باشد) بعبارت دیگر این خاصیت جزء تنظیمات پیشفرض ماژول است که شما می توانید آنرا غیر فعال کنید. سایر تنظیمات پورت سریال ماژول به قرار زیر است.

Data bits = 8 , Stop bits = 1 , Parity = None , Flow control = none

برای برقراری ارتباط بر اساس پورت سریال می توان از نرم افزار آماده ی ویندوز به نام Hyper Terminal استفاده کرد که نحوه ی آماده سازی آن در زیر بیان شده است.

a. ابتدا از مسیر زیر نرم افزار هایپر ترمینال را اجرا کنید :

[Start/All programs/Accessories/communications/Hyper Terminal](#)

b. از منوی File گزینه ی New connection را انتخاب کنید.

- c. در پنجره ی ظاهر شده یک نام و آیکون برای کانکشن خود انتخاب کنید.
- d. در پنجره ی بعدی مهمترین گزینه ، انتخاب شماره ی پورتی است که قصد برقراری ارتباط با آنرا دارید. چنانچه به تکمیل کردن گزینه های Country/region و Area code هم نیاز بود آنها را بترتیب با انتخاب نام کشور ایران و پیش کد تلفن شهرتان کامل کنید. سپس بر روی OK کلیک کنید.
- e. در پنجره ی بعدی باید تنظیمات مربوط به پورت سریال را انجام دهید که انتخاب Baudrate برابر ۹۶۰۰ ، Data bits = 8 ، Parity = None ، Stop bits = 1 و Flow control = none بهترین حالت می باشد. سپس بر روی OK کلیک کنید. (البته در صورت داشتن مهارت در کار با پورت سریال می توانید مقادیر دیگری را نیز اتخاذ کنید ، اما مقادیر ذکر شده تست شده هستند.)
- f. اکنون از منوی File گزینه ی Save As را انتخاب کنید و کانکشن خود را در محل دلخواه خود ذخیره کنید. و پنجره آنرا ببندید.

### توضیحات شماتیک و PCB مدار:

- بهمراه این آموزش یک فایل شماتیک و عکس نیز وجود دارد که نحوه ی اتصال این ماژول به سیم کارت ، منبع تغذیه و پورت سریال را نشان می دهد. مدار مورد نظر را ببندید و به موارد زیر دقت کنید.
- ✓ این مدار فقط برای ارسال SMS طراحی شده است. (سایر پایه های ماژول بکار گرفته نشده اند.)
  - ✓ در این مدار از قاب سیم کارت ۸ پین استفاده شده است.
  - ✓ یک سوراخ برای عبور سیم آنتن از فیبر تعبیه شده است.
  - ✓ برای ایجاد ولتاژ مورد نظر که مقدار کاربردی ۴.۳ ولت می باشد از رگولاتور سوئیچینگ LM2576 استفاده شده است.
  - ✓ در حالت عادی نیازی به استفاده از قطعه ی SMF05C نیست. (فعلا در بازار ایران یافت نمی شود.) بجای این قطعه می توان از دیودهای زنر با ولتاژ شکست ۵ ولت استفاده کرد. وظیفه ی این قطعه محافظت از سیم کارت در برابر ESD یا نویزهای الکترواستاتیکی ( Electro Static Discharge) است که ممکن است در هنگام تماس دست با سیم کارت یا موارد دیگر موجب سوختن آن شود.
  - ✓ در بعضی لحظات ممکن است که ماژول جریانی برابر با ۲ آمپر مصرف کند! بنابراین باید مطمئن باشید که منبع تغذیه ی شما توانایی تامین این جریان را دارد.
  - ✓ در آینده مدار کاملتری از این ماژول بهمرا PCB مربوطه ارائه خواهد شد. همچنین یک فایل آموزشی دیگر نیز درباره ی سایر امکانات ماژول تهیه خواهد شد. انشالله.

## راه اندازی ماژول SIM300cz :

برای روشن کردن ماژول به دو طریق می توان عمل کرد : **روش اول** استفاده از دکمه ی Power می باشد. که با ایجاد یک سطح منطقی صفر به مدت ۱۵۰۰ میلی ثانیه می توان ماژول را روشن نمود. بعد از روشن شدن یک کد **ناخواسته** یا **Unsolicited** که عبارت RDY می باشد از طرف پورت سریال ماژول ارسال می شود. و وضعیت پایه ی STATUS از صفر به ۲.۸ ولت تغییر می یابد و در تمام مدت زمان کار ماژول در همین مقدار باقی می ماند.

**روش دوم** استفاده از روش آلارم یا همان RTC می باشد که در این آموزش به آن نخواهیم پرداخت.

ابتدا سیم کارت را درون قاب سیم کارت قرار داده و سپس ماژول را توسط دکمه ی PWR روشن کنید. پس از روشن شدن ماژول LED نشانگر شبکه شروع به چشمک زدن می کند. حالات مختلف چشمک زدن این LED معرف وضعیت های مختلفیست که در جدول زیر آورده شده اند.

حالت چشمک زدن LED	رفتار یا وضعیت ماژول در شناسایی شبکه
خاموش	ماژول خاموش است.
۶۴ میلی ثانیه روشن/ ۸۰۰ میلی ثانیه خاموش	ماژول شبکه ی سیم کارت را پیدا نکرده است.
۶۴ میلی ثانیه روشن/ ۳۰۰۰ میلی ثانیه خاموش	ماژول شبکه ی سیم کارت را یافته است.
۶۴ میلی ثانیه روشن/ ۳۰۰ میلی ثانیه خاموش	ماژول در حالت ارتباط GPRS قرار دارد.

## برقراری ارتباط با کامپیوتر :

اکنون شما می توانید از طریق کانکشنی که ساخته اید با پورت سریال ماژول ارتباط برقرار کنید. برای این کار کانکشن خود را باز کنید. و سپس مراحل زیر را دنبال کنید.

I. ماژول خود را توسط دکمه ی POWER روشن کنید. و سپس ۳ تا ۵ ثانیه صبر کنید.

**تذکر بسیار مهم :** همواره بیاد داشته باشید که بعد از اولین باری که ماژول را روشن کردید حتما باید پیشوند AT یا at را ۲ تا ۵ مرتبه وارد کنید!

II. اگر ارتباط بین ماژول و کامپیوتر بدرستی برقرار شده باشد شما می توانید پیشوند AT یا at را در محیط هایپر ترمینال تایپ کنید.

III. بعد از تایپ هر دستور و فشردن کلید Enter یک سری کد **ناخواسته** یا **Unsolicited** تولید می شود که از ماژول به پنجره ی هایپر ترمینال وارد می شوند. در حالت Autobauding یا تشخیص خودکار Baudrate بعضی از این کدهای ناخواسته نمایش داده **نمی شوند!** از جمله بعداز وارد کردن پیشوند AT یا at کد ناخواسته ی RDY و +CFUN:1 و +CPIN:READY نمایش داده نمی شوند.

IV. چنانچه در هنگام تایپ کردن در محیط هایپر ترمینال با پیغام **Call Ready** مواجه شدید باید بدانید که مودم شما دائما در حال ریست شدن است و علت این ریست های مکرر قطعا کم بودن میزان جریان در مدار است!

V. پس از اینکه پیشوند At یا at را چند بار وارد کردید طبق جدول SMS می توانید دستورهای مربوط به پیام کوتاه را مطابق توضیحات ذیل بکار ببرید. (برای مشاهده لیست کامل دستورات به دیتاشیت مراجعه کنید.)

#### جدول SMS\*

دستور	توضیح
AT+CMGD	این دستور پیام کوتاه را حذف می کند.
AT+CMGF	توسط این دستور می توان قالب پیام ارسال شونده را تعیین کرد. (Text یا PDU)
AT+CMGL	این دستور پیام ها را از محل ذخیره شده لیست می کند.
AT+CMGR	توسط این دستور می توان پیام مورد نظر را خواند.
AT+CMGS	بوسیله ی این دستور می توان پیام کوتاه ارسال کرد.
AT+CMGW	این دستور یک پیام را در حافظه می نویسد.
AT+CMSS	این دستور یک پیام را از محلی که در آن ذخیره شده ارسال می کند.

\* برای مشاهده لیست کامل دستورات به دیتاشیت مراجعه کنید.

#### مثال های ارتباط با کامپیوتر :

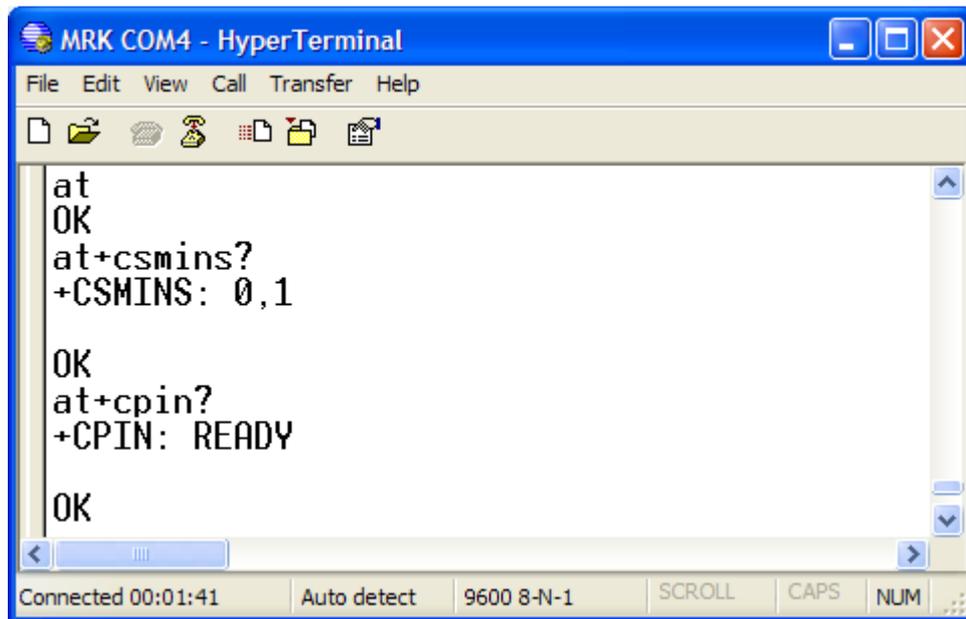
**تذکر :** در هر مثال با تایپ دستورات در پنجره ی هایپر ترمینال و زدن کلید Enter کدهای ناخواسته ای تولید می شود که برای درک بهتر این موضوع عکس هایی از محیط هایپر ترمینال در هر مثال قرار داده شده است. در هر خط از دستورات مفهوم علامت زدن  کلید Enter است.

**مثال ۱ :** دستورات زیر را در محیط هایپر ترمینال تایپ کنید ، با فشردن کلید Enter می توانید از صحت قرار داشتن سیم کارت و مورد تأیید بودن پین کد آن اطمینان حاصل کنید.

at 

at+csmins? 

at+cpin? 



```

MRK COM4 - HyperTerminal
File Edit View Call Transfer Help
at
OK
at+csmins?
+CSMINS: 0,1

OK
at+cpin?
+CPIN: READY

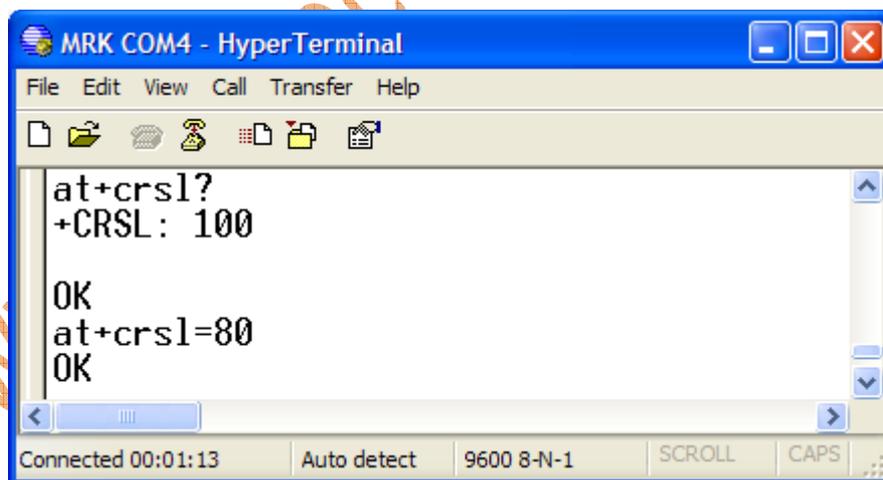
OK
Connected 00:01:41 Auto detect 9600 8-N-1 SCROLL CAPS NUM

```

**مثال ۲:** با وارد کردن دستورات زیر در محیط هایپر ترمینال می توانید ابتدا از میزان بلندی صدای زنگ مازول مطلع شوید و سپس مقدار دلخواه خود را وارد کنید. (این مقدار باید بین ۰ تا ۱۰۰ باشد)

at+crsl? ↵

at+crsl=80 ↵



```

MRK COM4 - HyperTerminal
File Edit View Call Transfer Help
at+crsl?
+CRSL: 100

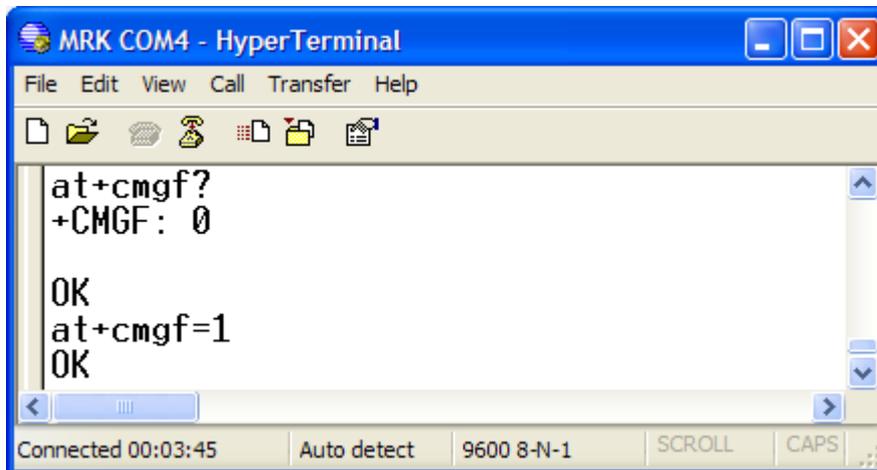
OK
at+crsl=80
OK
Connected 00:01:13 Auto detect 9600 8-N-1 SCROLL CAPS NUM

```

**مثال ۲:** از طریق وارد کردن دستور زیر می توانید قالب پیام کوتاه ماژول را مشاهده کنید و سپس آنرا در حالت Text تنظیم کنید. (صفر برای مد PDU و ۱ برای مد Text می باشد).

at+cmgf? ↵

at+cmgf=1 ↵

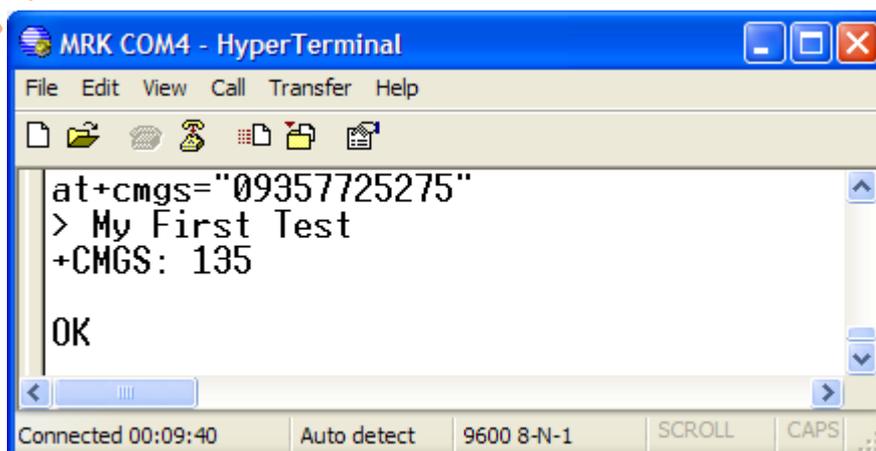


**مثال ۴:** برای ارسال پیام کوتاه دستورات زیر را تایپ کنید و در انتها بجای کلید Enter ، برای ارسال پیام از کلیدهای ترکیبی Ctrl+z استفاده کنید.

at+cmgs="09xxxxxxxx" ↵

>My First Test Ctrl+z

**توضیح:** در این مثال در خط اول دستور ارسال پیام کوتاه به همراه شماره ی همراه مورد نظر تایپ می شود و سپس کلید Enter زده می شود. با این کار نشانگر تایپ متن در محیط هایپر ترمینال به خط بعدی منتقل می شود که این خط بصورت خودکار با کاراکتر > شروع می شود. سپس شما باید در مقابل کاراکتر > متن مورد نظر خود را تایپ کنید و در انتها برای ارسال پیام از کلیدهای ترکیبی Ctrl+z استفاده نمایید.

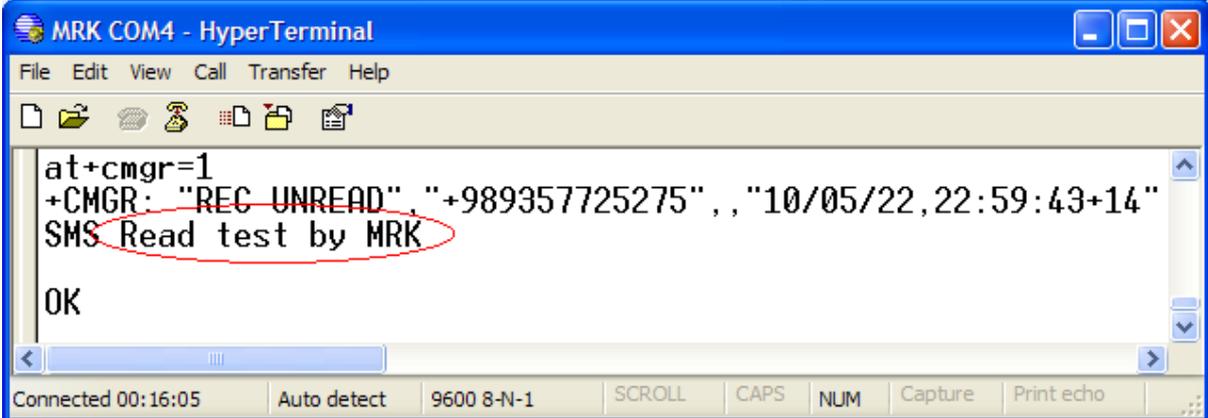


**تذکر:** در مرحله از ارسال پیام چنانچه از ارسال کردن آن منصرف شده باشید می توانید کلید Esc را بزنید. با این کار پروسه ی ارسال پیام از خط جاری لغو می شود.

**مثال ۵:** ابتدا توسط همراه خود یک پیام کوتاه به ماژول خود بفرستید ، سپس با وارد کردن دستورات زیر می توانید پیام رسیده به ماژول را مشاهده نمائید.

**نکته ی مهم:** برای خواندن یا مشاهده کردن پیام کوتاه رسیده ، اندیس مکانی که پیام رسیده در آنجا ذخیره شده است باید مشخص باشد. بعبارت دیگر پیام های کوتاه دریافتی در مکانهایی از حافظه ذخیره می شوند که که هر یک از این مکانها با یک اندیس مشخص شده اند ، در ماژول SIM300CZ این مکانها از ۱ تا ۱۵ را شامل می شوند. برای یافتن شماره ی اندیس مورد نظر هنگامی که به ماژول یک پیام ارسال می کنید به صفحه ی هایپر ترمینال نگاه کنید ، با دریافت پیام کوتاه ، یک کد ناخواسته ایجاد می شود که نشان دهنده ی شماره ی اندیسی از حافظه است که پیام کوتاه مورد نظر در آنجا ذخیره شده است.

at+cmgr=1 ↵



```
MRK COM4 - HyperTerminal
File Edit View Call Transfer Help
at+cmgr=1
+CMGR: "REC UNREAD", "+989357725275", "10/05/22,22:59:43+14"
SMS Read test by MRK
OK
Connected 00:16:05 Auto detect 9600 8-N-1 SCROLL CAPS NUM Capture Print echo
```

**توجه:** همانطور که مشاهده می کنید به همراه پیام کوتاهی که در عکس مشخص شده است ، اطلاعات دیگری از قبیل شماره ی فرستنده ، زمان ، تاریخ و ... نیز وجود دارد. که می توانید برای آگاهی از حذف و یا اضافه کردن این پارامترها و یا آشنایی بیشتر با آنها به دیتاشیت این ماژول مراجعه نمائید. (SIM300C\_ATC\_V2.00)

## ارتباط با میکروکنترلرهای AVR :

در این قسمت به آموزش ارتباط بین میکروکنترلرهای AVR و ماژول های GSM (ماژول نمونه ی SIM300cz) می پردازیم. توجه داشته باشید که اصول ارتباط سایر میکروکنترلرها با این نوع ماژول نیز مشابه همین موارد است.

در اینجا به مثالهایی اشاره می کنیم که به دو زبان C (کامپایلر CodeVision) و بیسیک (کامپایلر Bascom) گرد آوری شده اند تا درک مطلب برای کاربران هر کدام از این کامپایلرها ساده تر شود.

### به نکات زیر توجه کنید :

- ◆ برای ارتباط میکرو و ماژول به هیچ گونه وسیله ی جانبی جز چند رشته سیم نیاز نیست.
- ◆ در این ارتباط وجود حداقل سه اتصال RXD ، TXD و GND ضروریست.
- ◆ از آنجائیکه بین RI در ماژول نمایانگر دریافت پیام کوتاه و یا تماس تلفنی است ، می توان از ارتباط وقفه ای برای راحت تر شدن برنامه نویسی استفاده کرد. بنابراین می توانید پایه ی شماره ی ۹ از پورت سریال ماژول را به یکی از منابع وقفه در میکرو متصل کنید.
- ◆ همانطوریکه از بحث ارتباط با کامپیوتر متوجه شدید برای ارسال پیام نیاز به استفاده از کلیدهای ترکیبی Ctrl+z داریم. از آنجائیکه در برنامه نویسی میکرو فشردن کلید Enter یا Ctrl+z بحثی منتفی می باشد باید به طریق دیگری عمل کرد! و آن نیز فرستادن کد معادل این کلیدها به پورت سریال میکرو می باشد. کد چند کلید و کاراکتر مهم در جدول زیر آورده شده است.

کد هگزادسیمال	کد دسیمال (ده دهی)	کلید یا کاراکتر
00A	10	Line Feed
00D	13	Enter
01A	26	Ctrl+z
01B	27	Escap
022	34	" (double quote)
027	39	' (single quote)

- ◆ برای راحتی هرچه بیشتر و نیز بالاتر رفتن درک کاربران عزیز ، در این مثالها برنامه ها بسیار بسیار ساده نوشته شده اند! سورس این برنامه ها و نیز کد هگز آنها ضمیمه شده اند. که در آنها از میکروکنترلر ATmega8 و کریستال خارجی 6MHz استفاده شده است.

- ◆ نکته ی بسیار مهمی که باید به آن توجه کنید اینست که ساختار زبان برنامه نویسی بیسیک به گونه ایست که در هنگام ارسال یک مقدار به پورت سریال میکروکنترلر و یا

کامپیوتر کاراکتر Enter (عدد ۱۳) بطور پیشفرض ارسال می شود! و این بدان معنیست که در برنامه هائیکه برای میکروکنترلر (و یا بعدها برای کامپیوتر) می نویسید نیازی به ارسال کد کلید Enter نمی باشد! برای درک بهتر این موضوع به مثالها توجه کنید.(سورس همه برنامه ها به فایل آموزشی پیوست شده است)

### مثالهای ارتباط با AVR :

**مثال ۱ :** در این مثال با فشردن کلید متصل به PIND.2 ابتدا یک دستور at به پورت سریال میکرو فرستاده می شود، سپس مد پیام کوتاه به حالت Text ست شده و بعد از آن رشته ای با متن **First AVR Test** به ماژول فرستاده می شود تا ماژول آنرا به شماره ی مورد نظر ارسال کند. برای وارد کردن علامت ( " ) از ارسال کد اسکسی آن یعنی عدد ۳۴ استفاده شده است. همچنین از ارسال عدد ۱۲ برای وارد کردن کد **کلید Enter** و از ارسال عدد ۲۶ برای وارد کردن کد کلیدهای ترکیبی **Ctrl+z** استفاده نمودیم.

### کد برنامه ی ارسال پیام کوتاه برای بسکام :

```
$regfile = "m8def.dat"
$crystal = 6000000
$baud = 9600
'~~~~~ Configurations ~~~~~
Config Pind.2 = Input
'~~~~~ Main program ~~~~~

Do

If Pind.2 = 0 Then

    Print "at"

    Waitms 500

    Print "AT+CMGF=1"

    Waitms 500

    Print "AT+CMGS=" ; Chr(34) ; "09357725275" ; Chr(34)

    Waitms 100

    Print "AVR Test 1" ; Chr(26)

    Waitms 550

End If

Loop

End
```

## کد برنامه ی ارسال پیام کوتاه برای کدویزن :

```
#include <mega8.h>
#include <stdio.h>
#include <delay.h>

char Enter=13;
char double_quote=34;
char Ctrlz=26;

void main(void)
{

PORTD=0x00;
DDRD=0x00;

UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x26;

while (1)
{
    if (PIND.2==0){

        printf("at%c",Enter);
        delay_ms(500);

        printf("AT+CMGF=1%c",Enter);
        delay_ms(500);

        printf("at+cmgs=%c09357725275%c%c",double_quote,double_quote,Enter);
        delay_ms(100);

        printf("AVR Test 1%c",Ctrlz);
        delay_ms(100);

    };
};
}
```

**مثال ۲ :** در این مثال با فشردن کلید متصل به PIND.2 ابتدا دستور AT+CMGR=1 برای خواندن پیام کوتاه از اندیس ۱ حافظه ی ماژول ارسال می شود و سپس متن پیام مورد نظر از طریق ادامه ی برنامه خوانده شده و بر روی LCD نمایش داده می شود.

**تذکر :** LCDهایی با ۱۶ یا ۲۰ ستون برای نمایش کل متن دریافتی مناسب نبوده و شما نمی توانید پیام را بخوبی مشاهده نمائید. برای رفع این مشکل دو راه وجود دارد! **۱.** حذف موارد اضافه ای که به همراه متن پیام کوتاه به میکروکنترلر وارد می شوند ، از طریق ویرایش تنظیمات ماژول (برای اطلاعات بیشتر به دیتاشیت ماژول مراجعه نمائید). **۲.** استخراج متن پیام از بین اطلاعات دریافتی توسط دستورات برنامه نویسی میکروکنترلر.

## کد برنامه ی خواندن پیام کوتاه برای بسکام :

```

$regfile = "m8def.dat"
$crystal = 6000000
$baud = 9600
' ~~~~~ Configurations ~~~~~
Config Lcd = 16 * 2
Config Lcdpin = Pin , Db7 = Portc.0 , Db6 = Portc.1 , Db5 = Portc.2_
, Db4 = Portc.3 , E = Portc.4 , Rs = Portc.5

Config Pind.2 = Input
' ~~~~~ Variable ~~~~~
Dim B As Byte
Dim Sms As String * 250 : Sms = ""
' ~~~~~ initionaliz ~~~~~
Cls
Cursor Off
Home
Lcd "Get SMS..."
Waitms 1000
Cls
' ~~~~~ Main program ~~~~~
Do

If Pind.2 = 0 Then

Print "AT+CMGR=1"
Sms = ""

Do
B = Inkey()

Select Case B

Case 0:

Case 13:
If Sms <> "" Then Exit Do

Case 10:
If Sms <> "" Then Exit Do

Case Else
Sms = Sms + Chr(b)

End Select

Loop

Home
Lcd Sms

End If

Loop
End

```

## توضیح قسمت مهم برنامه :

در حلقه ی Do که در بدنه ی دستور if تعریف شده است توسط دستور ( $B = \text{inkey}()$ ) کد اسکری اولین کاراکتر موجود در بافر ورودی سریال خوانده می شود و بعداز عمل خوانده شدن ، اطلاعات موجود در بافر یک واحد شیفت داده می شوند! بدین معنی که بعداز خوانده شدن کد اسکری هر کاراکتر توسط دستور ( $\text{inkey}()$ ) آن کاراکتر از بافر حذف شده و بجای آن کاراکتر بعدی اش قرار می گیرد. سپس در بدنه ی دستور Select Case B از به هم چسبانیدن کدهای اسکری خوانده شده می توان به یک جمله یا همان رشته (String) دست یافت که در حقیقت حاوی اطلاعات ورودی به میکروکنترلر است. این کار توسط قسمت  $\text{Sms} = \text{Sms} + \text{Chr}(b)$  انجام می شود. برای تبدیل کدهای اسکری خوانده شده به کارکترهای معادل آنها از دستور  $\text{Chr}(b)$  استفاده شده است.

همانطوریکه از قبل می دانیم اطلاعات ارسالی از ماژول SIM300cz دارای قالبی به شکل زیر هستند. بنابراین توسط دستور Select Case B می توان انتهای اطلاعات رسیده از ماژول را مشخص کرد.

`<CR><LF><response><CR><LF>`

برای مشخص کردن انتهای اطلاعات رسیده از ماژول باید منتظر دریافت کدهای اسکری Enter و یا Line Feed باشیم. برای تشخیص این حالت از دستورهایی Case 10 و Case 13 استفاده کرده ایم. اما همانطوریکه در قالب اطلاعات مشاهده می شود این کدها دو بار ارسال می شوند! یکبار قبل از ارسال اطلاعات (response) و بار دیگر بعداز ارسال اطلاعات ، بنابراین آن کد Enter یا Line Feed مورد قبول است که بعداز رسیدن اطلاعات رسیده باشد ، برای تشخیص این حالت از دستور if در خلال حالات Case 10 و Case 13 استفاده شده است. بدین طریق که ، تغییر رشته ی sms (تغییر از حالت Null یا همان sms="") بیانگر دریافت اطلاعات است و کد Enter یا Line Feed رسیده بعداز این تغییر ، همان کد انتهای دستور است. در غیر اینصورت کدهای Enter و یا Line Feed رسیده مربوط به قبل از ارسال اطلاعات هستند.

با دریافت کد اسکری Enter یا Line Feed در انتهای اطلاعات ، برنامه از حلقه ی Do خارج شده و اطلاعات دریافت شده بر روی LCD نمایش داده می شوند.

زمانیکه بافر ورودی خالی شده باشد دستور ( $\text{inkey}()$ ) مقدار صفر را بر می گرداند ، در اینصورت برنامه باید منتظر بماند تا اطلاعاتی دریافت کند ، بنابراین با وقوع این حالت هیچ دستوری اجرا نمی شود و برنامه روال عادی خود را طی می کند. (برنامه از حلقه Do خارج نمی شود).

## کد برنامه ی خواندن پیام کوتاه برای کدویژن :

```

#include <mega32.h>
#include <delay.h>
#include <stdio.h>

#asm
. equ __lcd_port=0x1B ;PORTA
#endasm

#include <lcd.h>

void main(void)
{
char Enter=13;
char sms[250];
char B;

char k=0;
int j=0;
int i=0;

for (k=0;k<250;k++){
sms[k]=0;
};

PORTA=0x00;
DDRA=0x00;

PORTD=0x00;
DDRD=0x00;

UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x26;

lcd_init(16);

lcd_clear();
lcd_gotoxy(0,0);
printf("ATE0%c",Enter);
lcd_putsf("Get sms...");
delay_ms(1000);
lcd_clear();

while (1)
{
if (PIND.2==0){

printf("AT+CMGR=2%c",Enter);

while(j==0){

B=getchar();

switch(B){

case 0:

```

```

        break;

    case 10:
        if(sms[0]!=0){
            j=1;
        };

        break;

    case 13 :
        if(sms[0]!=0){
            j=1;
        };

        break;

    default:
        sms[i]=B;
        i++;

        };
    };

    lcd_puts(sms);
    delay_ms(500);
};
}

```

### توضیح قسمت مهم برنامه :

در این برنامه برای غیر فعال کردن حالت echo (اکو) در مازول از دستور ATE0 استفاده شده است. در حلقه ی `while(j==0)` که در بدنه ی دستور `if` تعریف شده است توسط دستور `( B=getchar)` اسکی اولین کاراکتر موجود در بافر ورودی سریال خوانده می شود و بعد از عمل خوانده شدن ، اطلاعات موجود در بافر یک واحد شیفت داده می شوند! بدین معنی که بعد از خوانده شدن کد اسکی هر کاراکتر توسط دستور `( getchar)` آن کاراکتر از بافر حذف شده و بجای آن کاراکتر بعدی اش قرار می گیرد. سپس در بدنه ی دستور `switch(B)` از به هم چسبانیدن کدهای اسکی خوانده شده می توان به یک جمله یا همان رشته (String) دست یافت که در حقیقت حاوی اطلاعات ورودی به میکروکنترلر است. این کار توسط قسمت `sms[i]=B` انجام می شود.

همانطوریکه از قبل می دانیم اطلاعات ارسالی از مازول SIM300cz دارای قالبی به شکل زیر هستند. بنابراین توسط دستور `Select Case B` می توان انتهای اطلاعات رسیده از مازول را مشخص کرد.

```
<CR><LF><response><CR><LF>
```

برای مشخص کردن انتهای اطلاعات رسیده از مازول باید منتظر دریافت کدهای اسکی `Enter` و یا `Line Feed` باشیم. برای تشخیص این حالت از دستورهای `Case 10` و `Case 13` استفاده کرده ایم. اما همانطوریکه در قالب اطلاعات مشاهده می شود این کدها دو بار ارسال می شوند! یکبار قبل از

ارسال اطلاعات (response) و بار دیگر بعد از ارسال اطلاعات ، بنابراین آن کد Enter یا Line Feed مورد قبول است که بعد از رسیدن اطلاعات رسیده باشد ، برای تشخیص این حالت از دستور if در خلال حالات Case 10 و Case 13 استفاده شده است. بدین طریق که ، تغییر رشته ی sms (تغییر از حالت Null یا همان تماما صفر) بیانگر دریافت اطلاعات است و کد Enter یا Line Feed رسیده بعد از این تغییر ، همان کد انتهای دستور است. در غیر اینصورت کدهای Enter و یا Line Feed رسیده مربوط به قبل از ارسال اطلاعات هستند.

با دریافت کد اسکی Enter یا Line Feed در انتهای اطلاعات ، برنامه از حلقه ی `while(j==0)` خارج شده و اطلاعات دریافت شده بر روی LCD نمایش داده می شوند.

زمانیکه بافر ورودی خالی شده باشد دستور `(getchar)` مقدار صفر را بر می گرداند ، در اینصورت برنامه باید منتظر بماند تا اطلاعاتی دریافت کند ، بنابراین با وقوع این حالت هیچ دستوری اجرا نمی شود و برنامه روال عادی خود را طی می کند. (برنامه از حلقه `while(j==0)` خارج نمی شود.)

در انتها بر خود لازم می دانم که از دوست عزیزم جناب مهندس حمیدرضا مداح بابت راهنماییهای مفیدشان تقدیر و تشکر نمایم.

گرد آورنده : مرتضی رحیمی

[Mrk.dadashi@gmail.com](mailto:Mrk.dadashi@gmail.com)

[www.mrkelectronic.blogfa.com](http://www.mrkelectronic.blogfa.com)

[www.hamidrezamaddah.blogfa.com](http://www.hamidrezamaddah.blogfa.com)

