

آموزش مقدماتی AVR

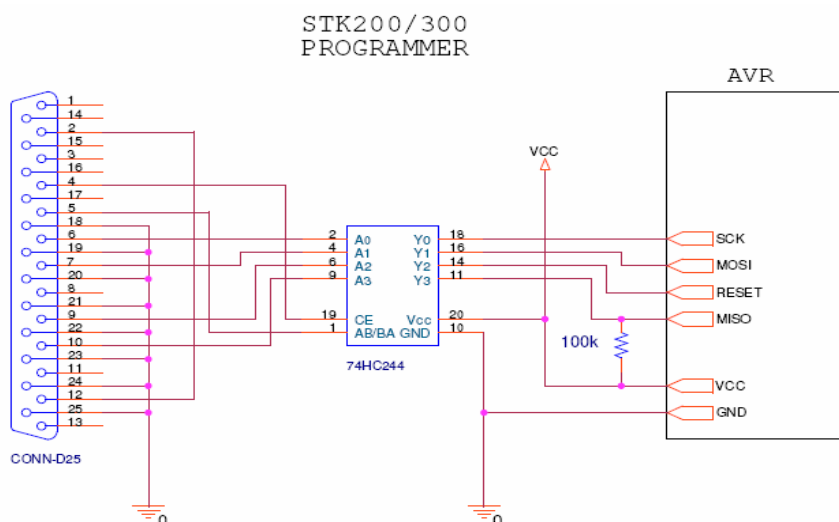
مسئله اسم AVR را زیاد شنیده اید. خیلی علاقمند هستید که با آن کار کنید، پروژه ببندید، ولی تا حالا موفق نشده اید. همین که می خواهید پا پیش بگذارید و شروع کنید با هزار و یک مساله پیش پا افتاده روبرو میشوید. CODEVISION، بهتر است یا BASCOM؟ از کجا آنرا DOWNLOAD کنم؟ چرا نصب نمیشود و ... نگران نباشید خیلی راحت همه اش را به شما آموزش میدهم. آنقدرها هم سخت نیست. در واقع اصلا سخت نیست. اصلا این مقاله برای چنین USER هایی پیش بینی شده.

روال یادگیری همیشه باید STEP BY STEP باشد. یعنی ابتدا چند پروژه ساده و سپس کارهای پیشرفته تر. و حتما هم باید با پروژه یعنی کار عملی شروع کرد. چون تا در عمل دست به کار نشوید هیچ چیزی یاد نمی گیرید.

پروگرامر:

قبل از هر چیز باید یک پروگرامر بسازید. تا موقعی که خودتان یک پروگرامر نداشته باشید نمی توانید سریع پیشرفت کنید. این پروگرامر در واقع فقط یک کابل است که البته یک IC بافر هم به آن اضافه شده. یک سر آن وصل میشود به پورت Printer کامپیوتر شما و سر دیگر به تعدادی از پایه های میکرو AVR. هیچ نیازی ندارد که میکرو را از مدار خارج کنید تا بتوانید آنرا پروگرام کنید. در حالی که در مدار سر جای خود هست می توانید آنرا پروگرام کنید. این نوع پروگرامرها را اصطلاحاً ISP (In system programming) می نامند.

نقشه پروگرامر به صورت زیر است. این پروگرامر به نام STK200/300 معروف است.



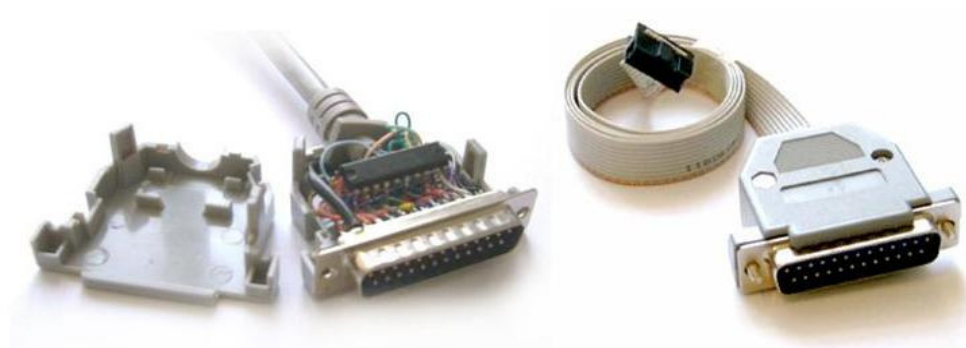
قطعات مورد نیاز:

- 1- کانکتور پورت Printer از نوع male
- 2- آی سی بافر 74HC244
- 3- یک عدد مقاومت 100K
- 4- 190cm کابل 6 رشته

طبق شماتیک مدار باید پایه های 18 تا 25 کانکتور Printer را به هم لحیم و به Ground مدار وصل کنید. تعدادی از پایه های دیگر نیز مطابق شکل به آی سی بافر وصل می شوند. این آی سی را باید در پوشش پلاستیکی کانکتور Printer قرار دهیم و از آن طرف نیز چند پایه آی سی به عنوان سیم های پروگرامر محسوب میشوند که از طریق یک کابل ادامه پیدا میکنند تا به AVR وصل شوند. مقاومت 100K را نیز باید کنار آی سی بافر درون پوشش کانکتور قرار دهیم. 6 سیم خروجی پروگرامر را نیز به یک Pin header وصل می کنیم.

برای ساخت این پروگرامر از یک کابل printer آماده هم میتوانید استفاده کنید. کافی است آی سی بافر را در سری male آن قرار دهید و سری female را نیز جدا کرده و در عوض از pin header استفاده کنید. تعدادی از سیم های کابل نیز بلا استفاده می مانند .

در شکل های زیر یک پروگرامر آماده را می توانید ملاحظه کنید.



نرم افزار:

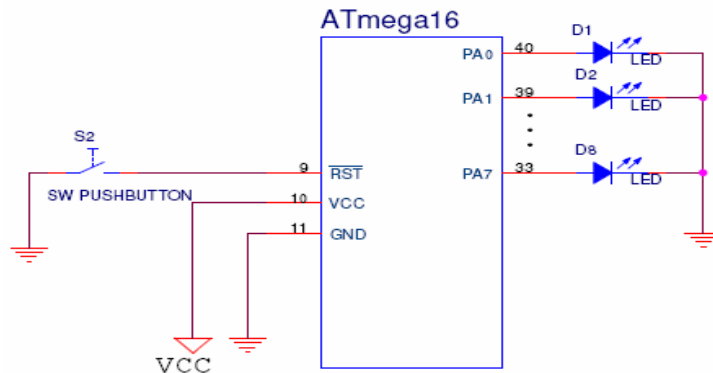
در این مقاله برای برنامه نویسی از کمپایلر CODEVISION استفاده میکنیم. این کمپایلر از زبان برنامه نویسی C استفاده میکند و از نظر من خیلی خوبه. اگر قبلا BACOM کار کردید (که به زبان BASIC برنامه می نویسند) هم میتوانید در این مقاله همراه ما باشید چون همه چیز را از پایه بیان میکنیم. این نرم افزار را می توانید از LINK زیر DOWNLOAD کنید:

<http://www.hpinfotech.ro/cvavre.zip>

البته این LINK نسخه DEMO را در اختیار شما می گزارد که البته بهتر هم هست چون حجم آن نسبت به نسخه اصلی خیلی کمتر هست (1.8 MB در مقابل 8 MB) و تنها محدودیت آن اینستکه تا 2 kb کد را بیشتر اجازه نمیدهد اما برای ما مشکلی ایجاد نمی کند. این نسخه DEMO به راحتی نصب میشود و هیچ دردسری ندارد.

پروژه 1 (LED چشمک زن):

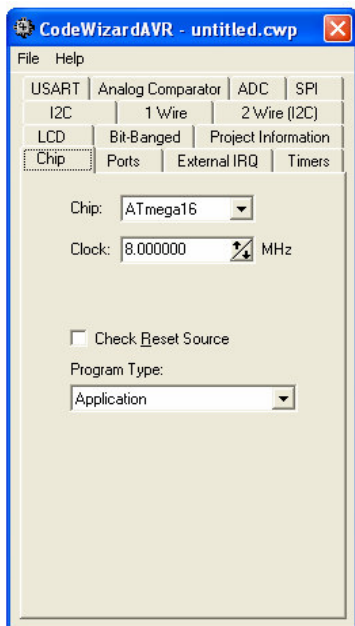
می خواهیم برنامه ای بنویسیم که Led های روی پورت A بطور پیوسته چشمک بزنند، صرفا همین شماتیک مدار به فرم زیر است:



کافی است برای مشاهده خروجی یک یا چند LED را به یک یا چند پایه از پورت A وصل کنید. نیازی هم نیست که مقاومتی با LED ها سری کنیم. فقط (+) و (-) آنها نباید برعکس باشد. (+) و (-) LED مطابق شکل زیر است: همانطور که ملاحظه می کنید پایه (-) کوتاهتر ولی در عوض سر بزرگتری دارد.



کلید S2 نیز Reset مدار است. برای تغذیه 5 ولت VCC هم دو کار می توانید بکنید. یا خروجی یک ترانس 220V:9V را به یک پل دیودی بدهید و از پل دیودی به یک رگولاتور 5V (L7805) وصل کنید (البته خروجی ترانس بیشتر از 9V هم میتواند باشد، بهتر است کاتالوگ L7805 را ببینید) و یا از یک آداپتور 12V استفاده کنید و مستقیما خروجی آنرا به رگولاتور بدهید. من آداپتور را توصیه میکنم، چون راحت تر و ایمن تر است. (+) و (-) سوکت آداپتور را با اهم متر مشخص کنید و (+) را به ورودی رگولاتور (پایه سمت چپ) بدهید، (-) را هم به GND (پایه وسط). خروجی (پایه سمت راست) را با اهم متر چک کنید باید نسبت به GND نسبتا دقیق 5V باشد.



حال به سراغ برنامه نویسی می رویم، یکی از مزایای برنامه نویسی در CODEVISION استفاده از روال Codewizard است. Codewizard یک تولید کننده اتوماتیک برنامه است که به فرم Visual می توان بخشهای مختلف برنامه را در آن مشخص کرد. icon آن که به فرم یک چرخ دنده است در بالای صفحه دیده می شود. با کلیک روی آن پنجره ای به فرم روبرو باز می شود که بسیاری از بخشهای عمده برنامه را در آن میتوان تعیین کرد. مثلا این که پورتهای مختلف بصورت ورودی باشند یا خروجی، از امکانات ارتباط سریال استفاده کنیم یا خیر، پیکره بندی تایمرها چگونه باشد و ...

در این مثال می خواهیم پورت A بصورت خروجی باشد، پس در قسمت Ports تمامی پایه های پورت A را بصورت Out در می آوریم. در قسمت Chip نیز نوع Chip یعنی ATmega16 و فرکانس کریستال، مثلا 8MHz را انتخاب میکنیم. در پایان از منوی فایل Generate, Save and Exit را انتخاب می کنیم و طی سه مرحله که اسم فایل های مختلف پروژه پرسیده میشود، اسم دلخواه خود را وارد کرده و Save می کنیم.

حال پنجره ای را به فرم شکل زیر ملاحظه می کنیم که حاوی متن برنامه است. در این برنامه بخشهای مختلف میکرو نظیر پورتهای، تایمرها، کانترها و ... پیکره بندی شده اند و در انتهای برنامه درون یک حلقه While() فضایی برای برنامه کنترلی شما تخصیص داده شده. ابتدا کمی راجع به پورتهای توضیح می دهیم و سپس به سراغ برنامه ای که باید درون حلقه While() بنویسیم، می رویم.

```

1  /*****
2  This program was produced by the
3  CodeWizardAVR V1.25.0a Evaluation
4  Automatic Program Generator
5  © Copyright 1998-2006 Pavel Haiduc, HP InfoTech s.r.l.
6  http://www.hpinfootech.com
7
8  Project :
9  Version :
10 Date   : 8/13/2006
11 Author : Freeware, for evaluation and non-commercial use only
12 Company :
13 Comments:
14
15
16 Chip type       : ATmega16
17 Program type    : Application
18 Clock frequency : 8.000000 MHz
19 Memory model    : Small
20 External SRAM size : 0
21 Data Stack size : 256
22 *****/
23
24 #include <megal6.h>
25
26 // Declare your global variables here
27
28 void main(void)
29 {
30 // Declare your local variables here
31
32 // Input/Output Ports initialization
33 // Port A initialization
34 // Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
35 // State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
36 PORTA=0x00;
37 DDRA=0xFF;
38

```

در AVR برای هر پورت سه رجیستر اختصاص داده شده مثلا برای پورت A داریم:

1) DDRA که جهت پورت را تعیین می کند: ورودی DDRA=0X00 خروجی DDRA=0XFF
0X نشان دهنده مبنای HEX است.

2) PINA که هنگام نوشتن در پورت استفاده می شود. مثلا کلید


3) PORTA که هنگام خواندن از پورت استفاده می شود. مثلا ارسال DATA روی LED

پس اگر بخواهیم کلیدی را به یکی از پایه های پورت A وصل کنیم، چون قصد نوشتن در پورت را داریم از رجیستر PIN استفاده می کنیم، ضمن اینکه چون پورت باید بصورت ورودی باشد DDRA را نیز برابر 0X00 قرار می دهیم. برعکس مثلا اگر بخواهیم LEDی را روشن کنیم (خواندن از پورت) از رجیستر PORT استفاده می کنیم و DDRA را برابر 0XFF قرار می دهیم. اگر به اوایل برنامه فوق نگاه کنید می بینید که چون پورت A را بصورت خروجی تعریف کردیم DDRA=0XFF است ولی سایر پورتهای بصورت پیش فرض ورودی مانده اند مثلا DDRB =0X00. بر روی خروجی تمامی پورتهای هم در شروع برنامه مقدار 00 ریخته شده که چیز خاصی را نمایش ندهند.

حال به سراغ برنامه ای میرویم که باید درون حلقه While() بنویسیم، باید یک موج مربعی تولید کنیم که موجب چشمک زدن LEDها شود. این برنامه به فرم زیر است:

```
PORTA = 0xff ; // port A = "1"  
delay_ms(20) ;  
PORTA = 0x00 ; // port A = "0"  
delay_ms(20) ;
```

دقت کنید که برنامه دقیقا مطابق فوق نوشته شود زیرا کمپایلر بین حروف کوچک و بزرگ تفاوت میگذارد. بعضی قسمتها مثل نام پورتهای یا پین ها با حروف بزرگ نوشته می شوند و قسمتهایی دیگر نظیر توابع با حروف کوچک نوشته می شوند والا کمپایلر Error میگیرد. برای این منظور می توانید از Help و مثالهای آن استفاده کنید.

تنها یک نکته دیگر باقی می ماند. در زبان C هر تابع زیر مجموعه ای از یک header file است که برای استفاده از تابع باید header file آن در ابتدای برنامه معرفی شود. تابع delay_ms() که بسته به عدد داده شده به آن Delay ایجاد میکند در یک header file به نام delay.h قرار دارد و برای استفاده از آن باید در ابتدای برنامه این header file بصورت `#include<delay.h>` در کنار سایر `#include`ها معرفی شود. پس این عبارت را به ابتدای برنامه اضافه می کنیم، برنامه اصلی را هم درون حلقه While() می نویسیم و به سایر قسمتهای برنامه هم که توسط Codewizard تولید شده کاری نداریم. حال آیکن  را click کنید تا پروژه ساخته و کمپایلر شود. اگر خطایی در برنامه وجود داشته باشد در اینجا مشخص می شود و باید برطرف شود.

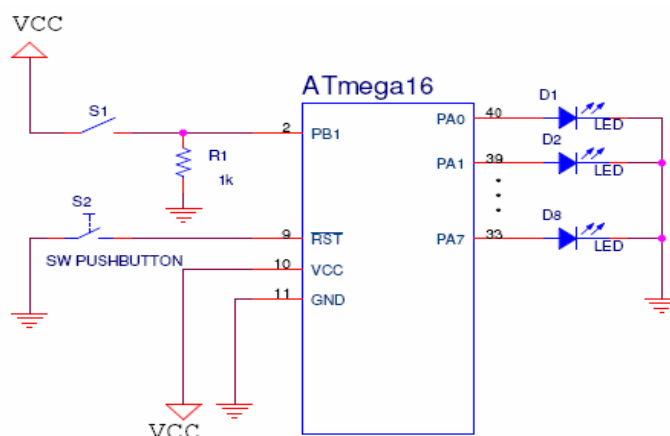
پس از آنکه برنامه کاملا بدون خطا کمپایل شد اکنون هنگام پروگرام کردن IC است. البته ابتدا به منوی `Settings>Programmer` رفته و نوع پروگرامر یعنی `Kanda Systems STK200+/300` و شماره پورت پاراللی که پروگرامر به آن وصل می شود (مثلا LPT1) را مشخص کنید. حال بر روی آیکونی که به شکل یک چیپ الکترونیکی است و آیکن پروگرامر می باشد کلیک کنید. در پنجره باز شده باید نوع چیپ و فرکانس کریستال همان باشد که در برنامه انتخاب کردید. حال از منوی `program>Erase chip` ابتدا اگر برنامه ای قبلا روی چیپ بوده را پاک کنید. سپس از منوی `file>load flash` نام برنامه تان را انتخاب کنید و به منوی `Program>flash` رفته و چیپ را پروگرام کنید. اگر در حین پروگرام کردن errorی پیش آمد نگران نباشید، بر روی No کلیک کنید و کلیه مراحل فوق را از ابتدا اجرا کنید. آنقدر این کار را تکرار کنید تا چیپ بدون error پروگرام شود. قاعدتا الان باید چشمک زدن LED را ببینید، اما یک نکته اصلی وجود دارد که باید همیشه به خاطر داشته

باشید. گاهی اوقات وصل بودن پروگرامر به مدار مانع کارکرد صحیح مدار می شود یا اصلا برنامه اجرا نمیشود. پس چه در این پروژه و چه پروژه های آتی اگر دیدید مدار کار نمی کند کابل پروگرامر را از مدار جدا کنید تا اگر مشکل از این بابت باشد برطرف شود.

****اخطار:** هیچ گاه در پنجره پروگرامر تنظیمات را دستکاری نکنید مگر آنکه کاملا آگاه باشید چکار می کنید چون ممکن است منجر به خراب شدن chip و یا پروگرام نشدن آن شود.


پروژه 2 (اضافه کردن کلید به مدار قبل):

در این پروژه می خواهیم چشمک زدن led ها توسط یک کلید start/stop کنترل شود. شماتیک مدار همانند حالت قبل است فقط کلید S1 اضافه شده :



بیا باید این بار برنامه را کاملا خودمان بنویسیم و به codewizard کاری نداشته باشیم. برنامه به فرم زیر است، سعی کنید برنامه را خودتان تحلیل کنید، من راجع به آن توضیح نمی دهم چون انتظار می رود مطالب ابتدایی را خودتان آگاهی داشته باشید.

```
#include<mega16.h>
#include<delay.h>
void main()
{
  DDRA=0xff ; //Port A = output
  DDRB=0x00 ; //Port B = input
  while(1)
  {
    unsigned char a ;
    a=0x02 ;
    a=PINB&a ;
    if(a= 0x02 )
    {
      PORTA = 0xff ; // port A = "1"
      delay_ms(20) ;
      PORTA = 0x00 ; // port A = "0 "
      delay_ms(20) ;
    }
  }
}
```

چون از codewizard استفاده نکرده اید برای ساخت پروژه باید یک سری کارهای اضافه تر انجام دهید. ابتدا از منوی `file>new` گزینه `source` را انتخاب کنید و در پنجره خالی ایجاد شده برنامه تان را بنویسید و به نام مثلا `kelid` آنرا `save az` کنید. حال مجدداً به منوی `file>new` بروید اما این بار گزینه `project` را انتخاب کنید و در پاسخ به سوال استفاده از `codewizard` ، `No` را کلیک کنید و `project` خود را به یک نام مثلا همان `kelid` ، `save` کنید. در پنجره ای که جدید باز میشود گزینه `Add` را انتخاب کنید و فایل سورس `kelid` را که `save az` کرده بودید به پروژه تان `Add` کنید. اگر این کار را درست انجام دهید در پنجره `navigator` سمت چپ صفحه فایل `kelid.c` در زیر شاخه `project:kelid` قرار میگیرد. در ضمن اگر مراحل کار را به ترتیب انجام ندهید ممکن است پنجره `Add` اتوماتیک ظاهر نشود که از منوی `projrct>configure` می توانید به آن دسترسی پیدا کنید. حال آیکن  را `click` کرده و سپس `IC` را پروگرام کنید. احتمالاً در هنگام پروگرام کردن لازم میشود نوع `chip` و فرکانس کریستال (مثلاً `8.000000Mhz`) را در پنجره پروگرام تنظیم کنید. (مزایای `codewizard` حتماً تا به حال برایتان روشن شده)

چنانچه مدار را درست بسته باشید با قطع و وصل کلید `S1` باید چشمک زدن `LED` کنترل شود. مجدداً گوشزد میکنم اگر مدار کار نکرد ممکن است لازم باشد پروگرام را از مدار جدا کنید.

پروژه 3 (ارتباط سریال با کامپیوتر):

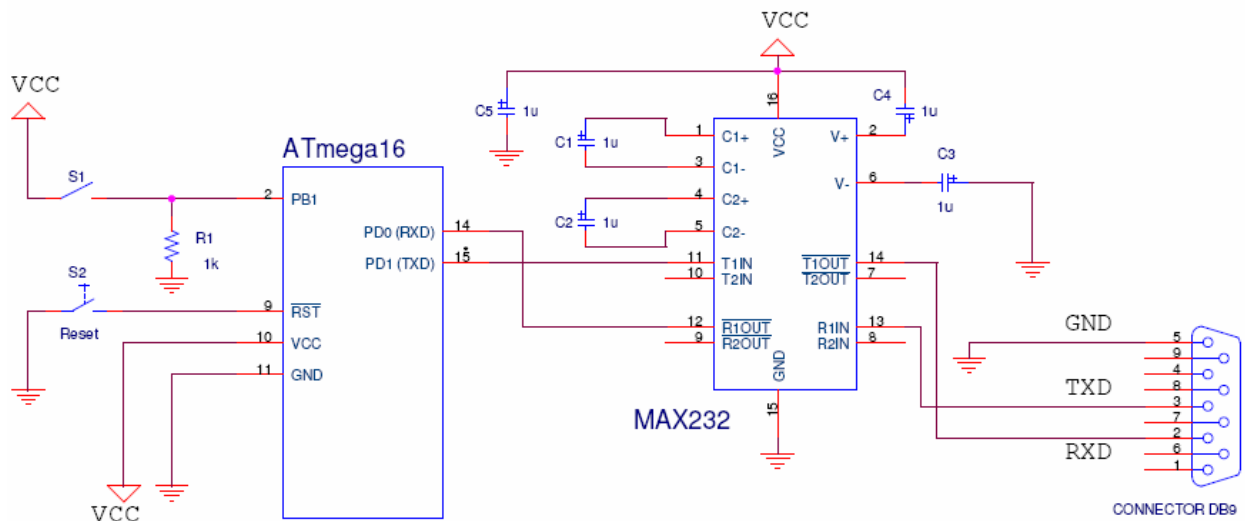
اگر از دو پروژه قبل جواب گرفته اید دیگر می توان مطمئن بود در سخت افزار مدار و پروگرام کردن و ... مشکلی ندارید و کم کم می توان سراغ پروژه های حساب شده تر رفت . اصلاً دو پروژه قبل را برای همین انتخاب کرده بودیم که بر مشکلات ابتدایی فائق بیاییم. برای این قسمت ارتباط سریال را انتخاب کرده ایم. من خود به شخصه پروژه ارتباط سریال را خیلی می پسندم چون اولاً پس از پایان آن احساس می کنید واقعا کاری انجام داده اید و از طرفی با اضافه کردن بخش دیگری نظیر یک سنسور دما و ... به آن میتوانید پروژه های کاملی در حد دوره کاردانی و یا حتی گاه کارشناسی داشته باشید. و از همه مهم تر چنانچه مشتاق پیشرفت در علم باشید این پروژه دید خوبی برای شما ایجاد می کند .

قبل از هر چیز باید راجع به اصول کار کمی توضیح دهیم. در ارتباط سریال اطلاعات را به فرم بسته هایی بنام `Frame` در می آورند. این بسته ها با سرعت استاندارد و یکنواختی ارسال و با همان سرعت دریافت می شوند. در واقع ساختار `Frame` و پارامترهای مربوط به آن به اضافه موارد دیگری پروتکل های گوناگون ارتباط را بوجود می آورند. استاندارد `RS232` یکی از این پروتکل ها است که برای ارتباط سریال در نظر گرفته شده و ما در این پروژه بر اساس آن کار می کنیم. در لینک زیر می توانید اطلاعات خلاصه شده ای در مورد آن و شیوه بکار گیری اش برای ارتباط `PC` و `AVR` ملاحظه کنید:

<http://www.qsl.net/pa3ckr/bascom%20and%20avr/rs232/index.html>

پروتکل `RS232` دارای جزئیات نسبتاً مفصلی است که چون در این مقاله منظور بررسی ارتباطات و شبکه نیست من راجع به آنها توضیح نمی دهم و پیشنهاد می کنم برای فهم بهتر از مراجع دیگر مطالعه ای داشته باشید، صرفاً به برخی از پارامترهای آن که ممکن است بر خوردی داشته باشیم اشاره ای میکنم. این پارامترها به عنوان مثال به فرم `9600,8,N,1` هستند که `9600` اصطلاحاً `Baud Rate` یا سرعت انتقال اطلاعات است، `8` تعداد بیت کلمه، `N` نمایانگر `No Parity` و `1` تعداد `Stop bit` ها است. در ضمن با حداقل ملزومات یعنی فقط سه سیم ارتباط را برقرار میکنیم. سیم `Transmit` یا `(TXD)` ، سیم `Receive` یا `(RXD)` و یک سیم هم به عنوان `Ground`.

شماتیک مدار به فرم زیر است:



در این پروژه شما به حدود 2 متر کابل 3 رشته احتیاج دارید که این کابل از یک طرف به یک کانکتور پورت سریال (com port) از نوع Female 9 پین وصل میشود و از طرف دیگر به یک آی سی واسطه. از آنجا که ولتاژ کاری کامپیوتر بیش از 5V یعنی چیزی حدود 12V است نمی توان از پورت سریال مستقیما به AVR وصل کرد. برای این منظور از یک آی سی واسطه یا بافر بنام MAX232 یا HIN232 (نام تجاری) استفاده میشود که اطلاعات میکرو را برای PC و اطلاعات PC را برای میکرو قابل استفاده و خوانا میکند.

باز هم به سراغ Codewizard بر می گردیم. Icon چرخ دنده ای شکل آن را کلیک کنید. به بخش USART بروید (ممکن است بجای USART عنوان UART0 را ببینید و یا اصلا آنرا نبینید. اگر اصلا آنرا نمی بینید احتمالا به سبب باز بودن پروژه های قبلی شما در صفحه CODEVISION است. پس تمام پروژه ها را ابتدا ببندید.) حال گزینه های Transmitter و Receiver را فعال کنید. Baudrate را برابر 2400 ، Communication parameters را برابر 8 Data, 1 stop, No parity و Mode را Asynchronous انتخاب کنید.

***اخطار:** دقت کنید هیچ چیز خاصی را تیک نزنید یا تغییر ندهید. در مبحث ارتباطات تمامی پارامترهای پروتکل ، نام پورتهای و مواردی از این قبیل باید به دقت تنظیم شوند، چرا که حتی تنظیم نبودن یک مورد به ظاهر کوچک باعث جواب نگرفتن کامل یا جواب نا مفهوم می شود.

در ادامه کار پورت B را هم چک کنید که همان پیش فرض input باشد. Clock را هم 1.000000MHZ انتخاب کنید. حال به منوی File > Generate, Save and Exit رفته و برنامه را تولید و Save کنید. اگر کار را به درستی انجام داده باشید باید انتهای برنامه شما دقیقا مطابق شکل زیر باشد:

```

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 2400
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x19;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

while (1)
{
    // Place your code here
}
}

```

برنامه زیر را درون حلقه While قرار دهید و کمپایل و پروگرام کنید:

```

unsigned char ;
a=0x02 ;
a=PINB&a ;
if(a == 0x02 )
printf("My name is Reza.What's your name?")

```

کانکتور پورت com را به یکی از پورت های com کامپیوتر (مثلا com1) وصل کنید. (پشت case های جدید دو عدد پورت com برای استفاده خارجی قرار داده شده که بالایی com1 و پایینی com2 است.) اگر چه که سخت افزار مدار در این پروژه دو طرفه بسته شده ولی در برنامه فوق اطلاعات فقط از میکرو به PC ارسال شده. با وصل کلید S1 باید عبارت " My name is " Reza.What's your name?" به طور پیوسته به PC شما ارسال شود. برای دیدن این عبارت شما نیاز به نرم افزاری دارید که با امکاناتی که در اختیار شما قرار می دهد بتوانید برنامه ای بنویسید که بتواند کنترل پورتهای PC شما را در اختیار بگیرد و Data های ارسالی از میکرو را بخواند و نمایش دهد مانند Matlab, Visual Basic, ... اما خود Codevision نیز محیطی را در اختیار شما قرار می دهد که برای همین کار طراحی شده بنام Terminal Emulator. این محیط دارای کارایی خوب و حساب شده ای است به گونه ای که ترمینال امولاتور Codevision را در رده ترمینالهای قوی و معتبر قرار داده است و شما نیازی به برنامه نویسی برای کد گشایی Data های ارسالی از میکرو ندارید.

اما ابتدا باید پارامترهای این ترمینال را تنظیم کنید. به منوی Settings>Terminal رفته و پارامترهای زیر را تنظیم کنید:

Port:	Com1
Baud rate:	2400
Data bits:	8
Stop bits:	1
Parity:	none
Emulation:	TTY
Handshaking:	none

سایر پارامترها را هم بصورت پیش فرض بگذارید. دقت کنید پورت com ای را که انتخاب می کنید همان باشد که کابل Serial شما به آن وصل شده. حال بر روی آیکن Terminal که به شکل یک مانیتور کامپیوتر است کلیک کنید. در پنجره باز شده از

میان دکمه های بالا دکمه ای را که مربوط به فرمت نمایش است و میتواند یکی از دو گزینه HEX و ASCII باشد را بر روی ASCII قرار دهید. الان اگر کلید S1 مدارتان ON باشد باید بتوانید Data های ارسالی را مشاهده کنید.

خوب است در اینجا اشاره ای هم به زبان برنامه نویسی matlab برای کاربرد ارتباط سریال داشته باشیم. چرا که با توسعه پروژه مثلا اضافه کردن یک سنسور دما میتوانید data های ناشی از دما را در این نرم افزار خوانده و سپس بصورت گراف رسم کنید. دستوراتی که با کمک آنها می توانید پورت سریال را بخوانید به قرار زیر هستند:

```
s = serial('com1') ;  
set(s,'BaudRate',2400) ;  
fopen(s) ;  
data= fscanf(s)
```

دستورات فوق را در پنجره Command زبان برنامه نویسی matlab کپی کنید تا Data های ارسالی را ببینید. دقت کنید در انتهای سطر آخر سمی کالن نباید بگزارید. در ضمن همزمان پورت com موردنظر شما در اختیار برنامه دیگری نباشد مثلا ترمینال امولاتور Codevision باز نباشد. در پایان کار دستورات زیر را وارد کنید تا پورت بسته شود:

```
fclose(s)  
delete(s)  
clear s
```

در این مقاله سعی بر این بوده که تا جای ممکن مطالب عملی را که معمولا در کتابها نمی توان پیدا کرد توضیح دهیم و مشکلات اولیه ای را که مانع حرکت و Start کار می شوند هموار کنیم تا بلکه شخص بتواند در ادامه خود کار را دنبال کند و پیشرفت نماید