www.mohandesyar.com

عنـــــــوان

شمارنده AVR

یا حق

پروژه آموزشی گام به گام راه اندازی شمارنده ۰ – ۹۹

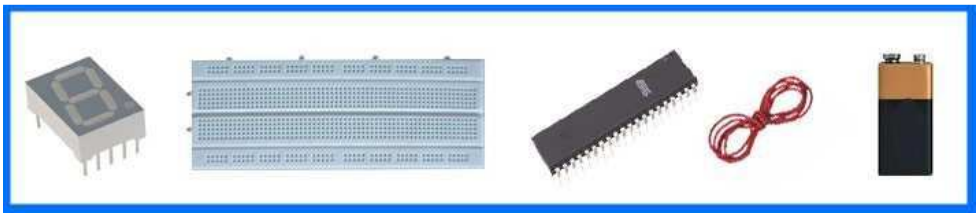
توسط سون سگمنت یا نمایشگر هفت قسمتی

و میکروکنترلر **ATmega16**

• قطعات مورد نیاز

در ابتدا لازم است که قطعات مورد نیاز در این پروژه را با هم مورد بررسی قرار دهیم:

۱- سون سگمنت یا نمایشگر هفت قسمتی یا نمایشگر هفت پارچه (Seven Segment)
۲- میکروکنترلر ATmega16
۳- بِردبورد Bread Board
۴- منبع حدودا ۵ ولتی
۵- سیم



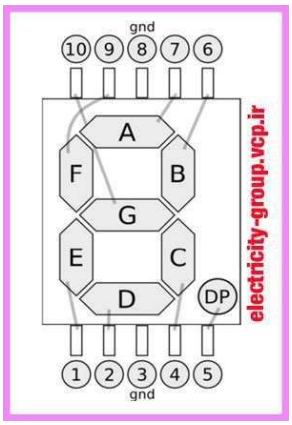
در میان قطعات نامبرده مورد ۱و ۲ نیاز به به بررسی دقیق دارد که در ادامه به این مهم خواهیم پرداخت.

۱- سون سگمنت

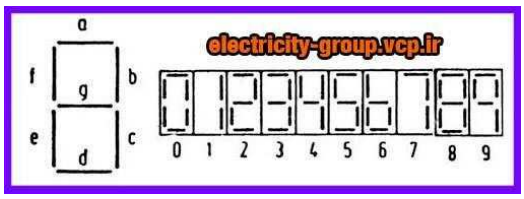
سون سگمنت ها نمایشگر های ارزان قیمتی هستند که جهت نمایش اعداد استفاده می شوند.استفاده از سون سگمنت ها، به جای نمایشگر های LCD به مقدار زیادی هزینه ها را کاهش می دهد.

"سون سگمنت" به فارسی "نمایشگر هفت پارچه" ترجمه شده است. این نمایشگر ۱۰ پایه دارد یعنی ۵ پایه در هر طرف (در شکل زیر می بینید). این نمایشگر از هفت قطعه تشکیل شده است. این هفت قطعه ، هفت دیود نوری(LED) هستند که در کنار هم قرار گرفته اند و شکل ۸ لاتین شده اند. در داخل نمایشگر این LED ها روشن می شوند و قطعه ی مربوط به خود را روشن می کنند.

هفت قطعه ی یک سون سگمنت با حروف a تا g شناخته می شوند. علاوه بر این هفت قطعه ، یک LED کوچک دیگر هم در گوشه ی سمت راست پایین وجود دارد. این نقطه ی کوچک با علامت dp شناخته می شود که مخفف decimal point است. این نقطه گاهی برای نمایش ممیز به کار می رود. (مطابق شکل زیر)

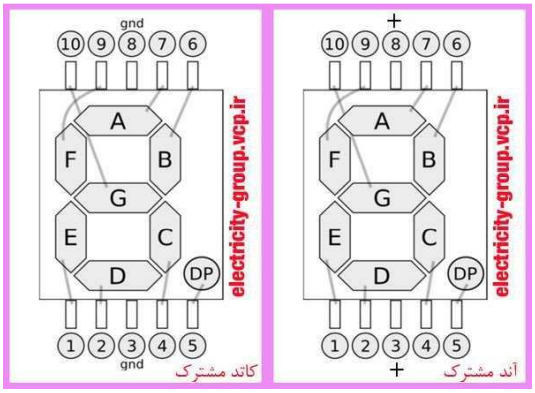


برای نمایش هر یک از ارقام ۰ تا ۹ باید ترکیب مناسبی از هفت قسمت a تا g را روشن کنیم. به شکل زیر دقت کنید که نحوه ی نمایش ارقام را نشان می دهد:



پایه های سون سگمنت:

در سون سگمنت ۱۰ پایه وجود دارد. پایه های وسط در بالا و پایین به منفی (زمین) یا مثبت منبع وصل می شوند، یک پایه هم نقطه است و ۷ پایه ی باقی مانده ، هرکدام مربوط به یکی از لامپ (قطعه) های نمایشگر است. پایه های سون سگمنت را در شکل زیر می بینید:



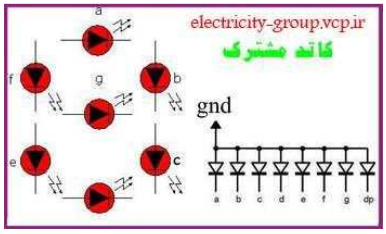
نکته:ابتدا باید یاد بگیریم که در مدار منطقی نام دیگر منفی ، "کاتد" است. همچنین نام دیگر مثبت ، "آند" است. یعنی می توانیم به ترتیب به جای مثبت و منفی بگوییم آند و کاتد.

وقتی برای خرید سون سگمنت می روید ، فروشنده از شما می پرسد: " سون سگمنت کاتد مشترک می خواهید یا آند مشترک؟"

حال ببینیم تفاوت این دو چیست؟

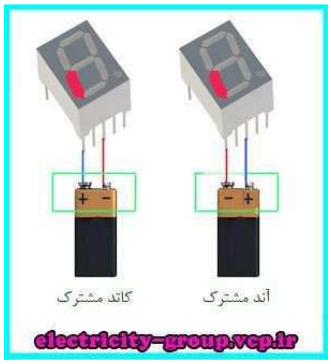
فکر کنید که یک لامپ (LED) چگونه با دو پایه روشن می شود؟ همانطور که می دانید یک سر با پایه بلند تر به سر مثبت و یک سر با پایه کوتاه تر به سر منفی یا همان زمین متصل می گردد. در سون سگمنت در واقع سر منفی(کاتد) همه ی لامپ ها به هم وصل شده و این اتصال مشترک از طریق یکی از پایه های نمایشگر از آن خارج می شود. این پایه ی مشترک نمایشگر ، همان پایه ی وسط است که هم بالا و هم پایین نمایشگر وجود دارد. اگر یکی از دو پایه ی وسط را به منفی متصل نماییم، منفی همه ی لامپ ها تامین می گردد. پس از این که پایه ی مشترک را به منفی وصل کردیم ، به هر پایه ای که مثبت دهیم ، لامپ مربوطه اش روشن می شود.

(۸ پایه: از a تا g و نقطه)



به این نوع نمایشگر که منفی آن مشترک است ، "سون سگمنت کاتد مشترک" می گویند.
نمایشگرهایی که به همین شکل هستند ولی مثبت شان مشترک است. برای کار با آنها باید به پایه مشترک مثبت بدهیم. سپس برای روشن کردن هر قطعه(لامپ) ، به پایه ی مربوطش منفی بدهیم.

که به این نوع نمایشگر که منفی آن مشترک است ، "سون سگمنت آند مشترک" می گویند.



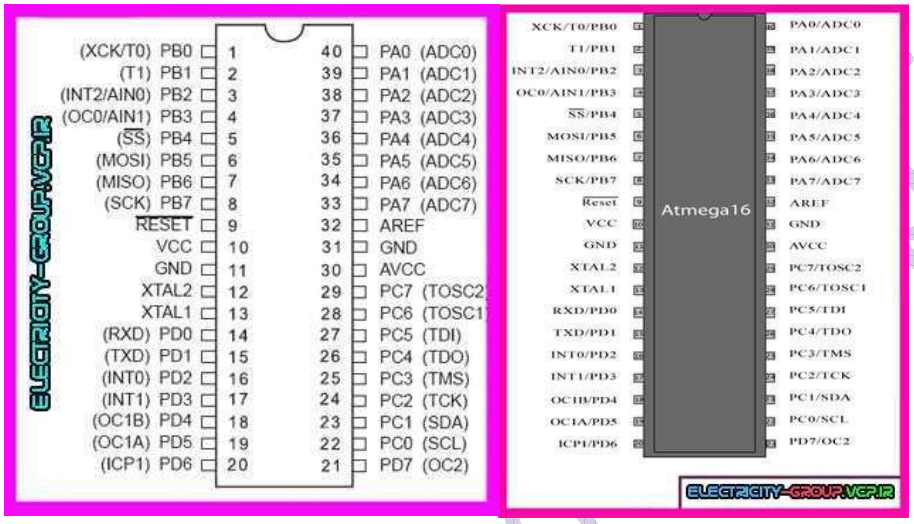
۲- میکروکنترلر ATmega16

میکروکنترلر در زبان فارسی به معنی "ریز کنترل کننده" است. این قطعه در واقع یک کنترل کننده ی مرکزی و یک مرکز تصمیم گیری و هدایت برای مدارهای ماست. این قطعه یک آی سی است که می تواند توسط کاربر برنامه ریزی شود. برنامه ریزی آن نیز توسط زبان های مختلف برنامه نویسی مانند C، اسمبلی و basic انجام می شود.

فقط کافیست تمام ورودی و خروجی های مدارخود را در اختیار میکرو کنترلر قرار دهیم و سپس الگوریتم مورد نظر خود را تحت یکی از این زبان های برنامه نویسی پیاده سازی کرده و میکروکنتلر را برنامه ریزی کنیم، حالا این قطعه به راحتی مدار ما را به طور کامل کنترل می کند.

تصویربالاتصویریک میکروکنترلرATmega16 است. این میکرو کنترلر یک آی سی ۴۰ پایه از خانواده میکروکنترلرهای AVR است و به دلیل ویژگی های خاص و قیمت مناسبش، به عنوان یکی از پرکاربردترین و معروف ترین انواع میکروکنترلرها در جهان شناخته شده است.

پایه های میکروکنترلر ATmega16:



میکروکنترلر ATmega16 دارای ۴ پورت (Port) یا درگاه است. هر پورت دارای ۸ پایه است که می توانند به عنوان ورودی یا خروجی استفاده شوند. در حقیقت این میکروکنترلر دارای ۳۲ پایه برای دریافت اطلاعات و یا صدور دستورات مختلف برای کنترل سایر قطعات است. ۸ پایه ی دیگر نیز وظایف مختلفی بر عهده دارند.

همانطور که در شکل بالا مشهود می باشد به طور خلاصه:

پایه های ۳۳ تا ۴۰ : پورت A

پایه های ۱ تا ۸ : پورت B

پایه های ۲۲ تا ۲۹ : پورت C

پایه های ۱۴ تا ۲۱ : پورت D

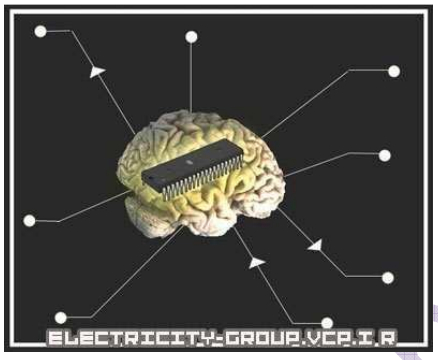
پایه ۱۰ : اتصال به مثبت منبع

پایه ۱۱ : اتصال به منفی منبع (زمین)

به دلیل گستردگی مطالب در ارتباط با میکروکنترلر به همین آشنایی ساده بسنده می کنیم.

• برنامه نویسی و پروگرام میکروکنترلر

همانطور که در بالا گفته شد میکروکنترلر به عنوان مغز مدار در نظر گرفته می شود، یعنی یک کنترل کننده ی مرکزی و یک مرکز تصمیم گیری و هدایت برای مدارهای ماست. بنابراین در هر مداری مجموعه ای از دستورالعمل ها را به زبان خاص میکرو به آن می دهیم تا تصمیم گیری و مدار را هدایت نماید.

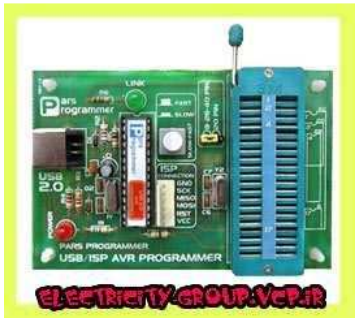


برای برنامه ریزی میکروکنترلر، زبان های مختلفی وجود دارد اما ما اززبان C که یکی از کاملترین زبان های برنامه نویسی روز دنیاست، استفاده می کنیم. اما برنامه ای که در محیط این برنامه نوشته می شود به خودی خود قابل فهم برای میکروکنترلر نمی باشد.

به برنامه ای که توسط کاربر نوشته می شود، Source گفته می شود. این برنامه باید توسط یک نرم افزار، به زبان قابل فهم برای میکروکنترلر تبدیل شود. به این نرم افزار کامپایلر می گویند. به این برنامه ی کامپایل شده نیز، یک Object می گویند. حالا باید این Object توسط نرم افزار دیگری به چیپ (Chip) یا همان آی سی منتقل شود. به این عمل، یعنی انتقال برنامه ی کامپایل شده به چیپ، پروگرام کردن می گویند و به نرم افزاری که این کار را انجام می دهد پروگرامر (Programmer) می گویند. محیطی که ما در آن برنامه ی مورد نظر خود را می نویسیم (تایپ می کنیم) Editor نام دارد. این نرم افزار ما را در میان برنامه نویسی بسیار کمک می کند، مثلاً کلمات رزرو شده و غیر قابل تعویض را با رنگها و فونت های گوناگون برای ما برجسته می کند.

این ۳ برنامه،یعنی**کامپایلر، پروگرامر و ادیتور** ،درقالبنرم افزاری بهنام " Code Vision " توسط شرکت HP به بازار عرضه شده است . کاربر با نصب این نرم افزار بر روی کامپیوتر شخصی خود، در حقیقت هر ۳ برنامه را، به علاوه ی چندین قابلیت و برنامه ی جانبی دیگر را بر روی دستگاه خود نصب کرده است. در واقع Code vision یک بسته‌ی نرم افزاری کامل و جامع برای خانواده ی AVR است که تمام نیازهای نرم افزاری ما را برای کار کردن با میکروکنترلرهای این خانواده برطرف می کند.

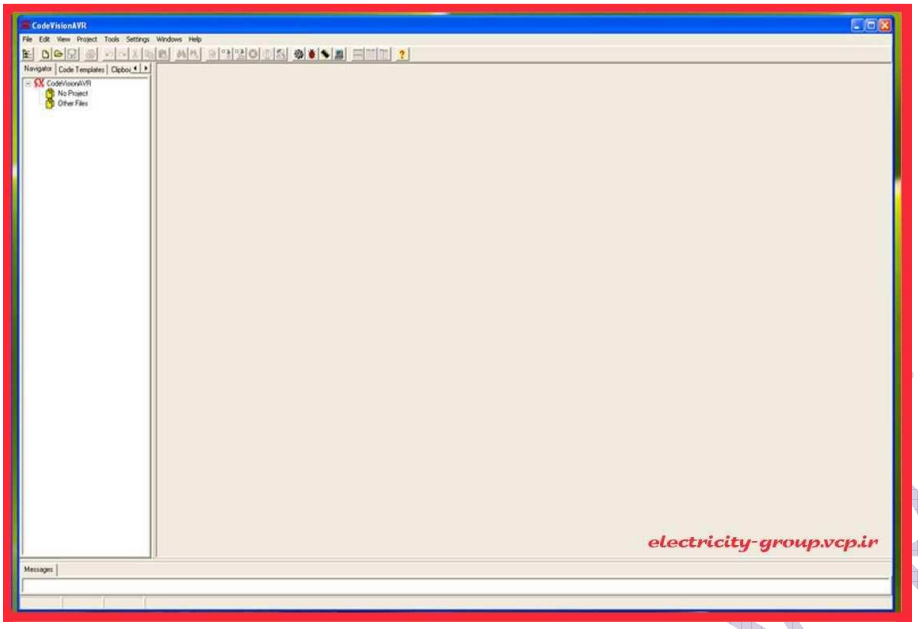
در نهایت برای انتقال برنامه نوشته شده از رایانه به میکروکنترلر یا ریزپردازنده (پروگرام کردن چیپ)، از یک مدار جانبی به نام "Micro controller programmer" استفاده می کنند و میکرو را در آن مدار قرار داده و میکرو باید روی آن مدار پروگرام شود. در شکل زیر یک نمونه پروگرامر قابل مشاهده می باشد.



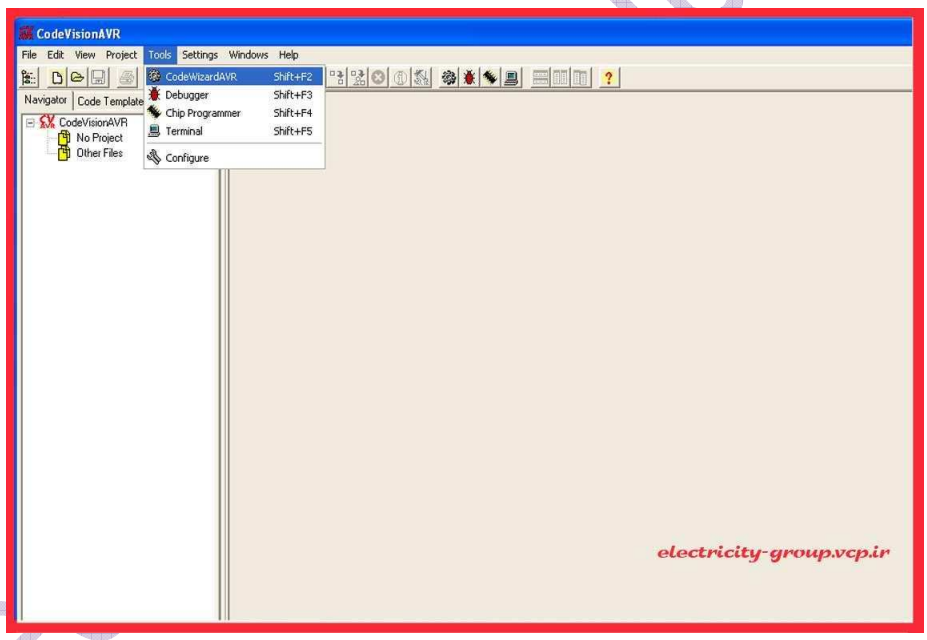
برای پروگرام کردن برنامه توسط پروگرامر معمولا از کابل پرینتر استفاده میگردد که یک نوع کابل موازی یا سریال می باشد. اما برخی پروگرامرها قابلیت پروگرام کردن با کابل USB را نیز دارا هستند. همانطور که می دانید چون USB در دسترس تر و آشنا تر می باشد کار کردن با آن راحت تر است.

---------- شروع به کار با کامپایلر CodeVision ----------

ابتدا برنامه را باز کرده



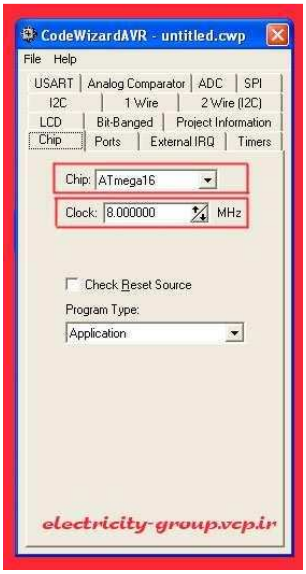
سپس مطابق شکل زیر شا خه ی CodeVisardAVR را باز می کنیم.



پس از کلیک بر روی CodeVisardAVR زیر صفحه شکل زیر باز می گردد.

حال در این "زیر صفحه" باید از قسمت Chip نوع میکروکنترلر که در این پروژه همان ATmega16 می باشد و در قسمت Clock نیز کلاک مورد نظر را وارد نمود که در اینجا کاری به کلاک نداریم و دستخوش تغییر نمی کنیم.

بنابراین :



سپس از نوارهای بالای این زیر صفحه، زیر شاخه Ports را انتخاب می نماییم. سپس پین های دو پورت A و D را به عنوان خروجی انتخاب می کنیم (out) . زیرا میکرو با ارسال مقادیر به نمایشگر هفت قسمتی اعداد مختلف را نمایش می دهد. پورت A را برای سون سگمنت یکان شمارنده و پورت D را برای سون سگمنت دهگان شمارنده انتخاب می کنیم.(شکل های # و ##)

(شکل ###) (شکل ##) (شکل #)

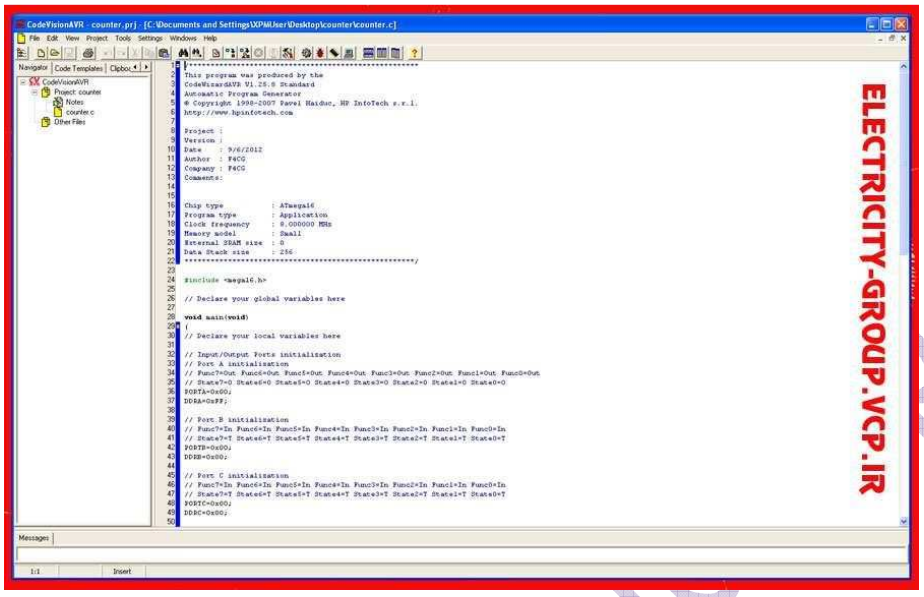
حال که تنظیمات اولیه تمام شده از نوار بالای "زیر صفحه" مسیر زیر را دنبال نمایید. (شکل ###)

File >> Generate, Save and Exit

در صفحات باز شده پروژه را به نام مورد نظر و در مکان دلخواه ذخیره نمایید.

(روند ذخیره در ۳ صفحه متوالی به طور پی در پی تکرار می گردد.)

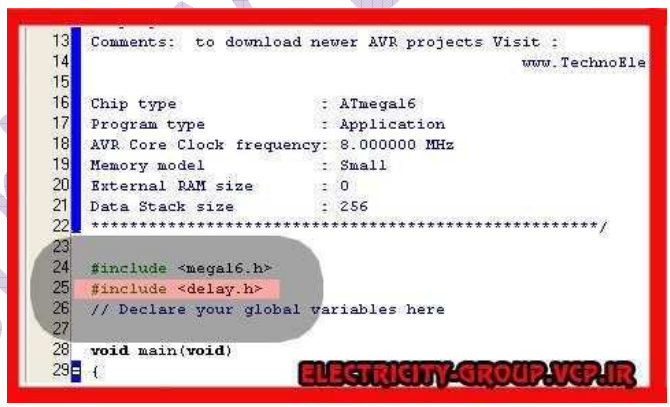
پس از انجام فرایند ذخیره سازی، صفحه ای که در آن کد نویسی و برنامه نویسی انجام می گیرد باز می شود. (شکل زیر)



توجه: در ادامه روش نوشتن کد توضیح داده خواهد شد و پس از آن کدهای نوشته شده را مورد بررسی قرار می دهیم.

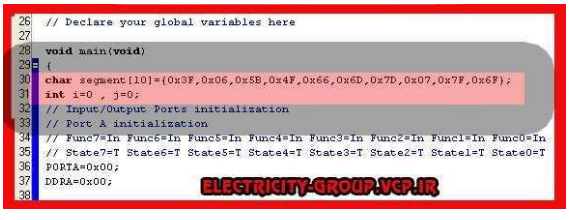
کدهای زیر را در مسیرهای مشخص شده زیر قرار دهید:

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

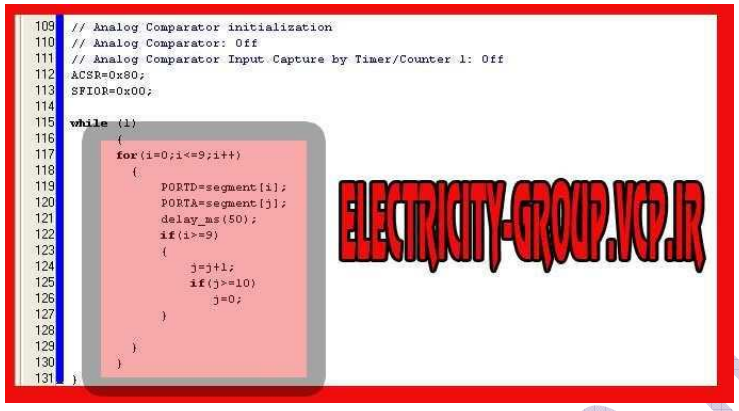



```
char segment[10]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};

int i=0 , j=0;
```



```
for(i=0;i<=9;i++)
{
PORTD=segment[i];
PORTA=segment[j];
delay_ms(50);
if(i>=9)
{
j=j+1;
if(j>=10)
j=0;
        }
}
```

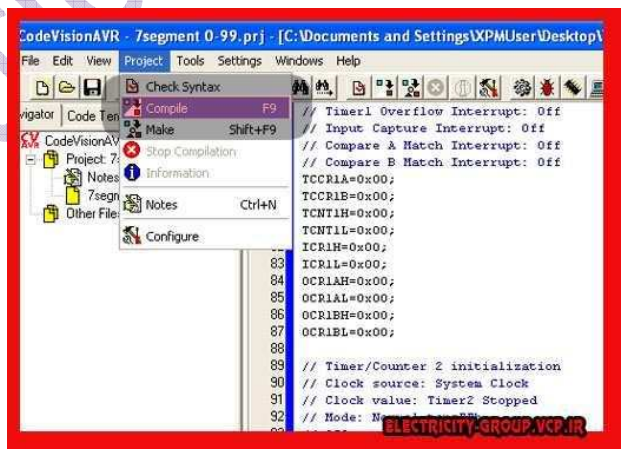


خوب در اینجا کدنویسی برنامه مورد نظر تمام شد. حال باید برنامه نوشته شده را ذخیره کرده و فایل قابل فهم برای میکرو را تولید نماییم.

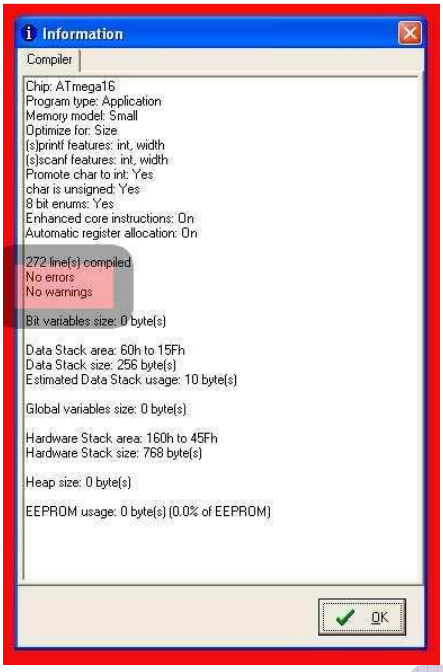
پس برنامه را از منوی File با کلیک بر روی گزینه Save ذخیره نمایید.

در این قسمت ابتدا باید برنامه مورد نظر را چک کنیم تا مشکلی نداشته باشد، به همین جهت مسیر زیر را دنبال نمایید.

Project >> Compile (یاF9)

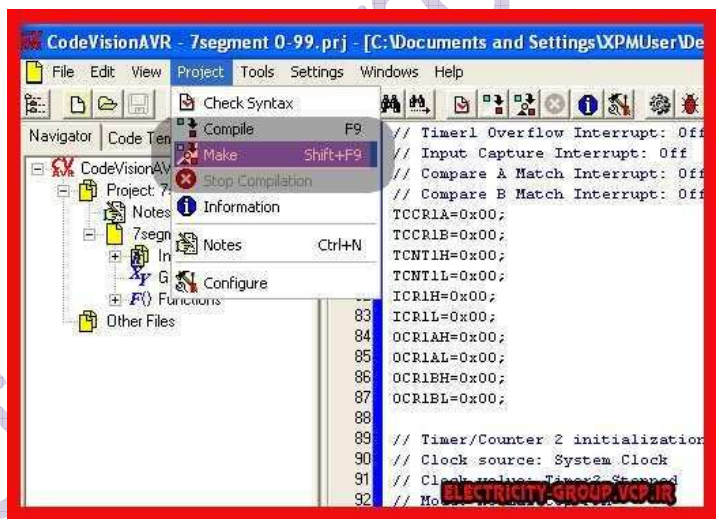


با کلیک بر روی این قسمت در صورتی که برنامه مشکلی داشته باشد متوجه می شویم. اما در صورتی که با پیغام زیر مواجه شوید یعنی برنامه درست بود و بدون مشکل است.



بر روی OK کلیک کنید. و در نهایت برای تولید فایل قابل فهم میکرو مسیر زیر را دنبال نمایید.

Project >> Make



و بر روی OK در صفحه پیغام باز شده کلیک کنید. حال فایل مورد نظر ما که با پسوند hex. در مسیری که پروژه ذخیره شده، تولید شده و باید از طریق پروگرامر به میکروکنترلر انتقال داده شود.

---------- بررسی کدهای نوشته شده ----------

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

با افزودن این کد در واقع کتابخانه مورد نیاز برای قرار دادن دستور ایجاد تاخیر که در آخر برنامه نوشته شده قرار دارد تعریف می گردد.

```
char segment[10]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};

int i=0 , j=0;
```

- این کد در واقع متغییر segment[10] را به عنوان char تعریف می کند.
- عبارات روبروی charsegment[10]متغییرهایی هستند که برابر segment[ ] قرار می گیرند.
- عدد 10نوشته شده در آرگومان segment[10] در واقع نشان دهنده این است که این متغییر توانایی ذخیره 10 عبارت را در خود داراست. و این عبارات در روبروی آن در کروشه به ترتیب تعریف شده اند، یعنی عبارت اول در segment[1] و عبارت دوم در segment[2] و به همین ترتیب تا آخر ذخیره می شوند.

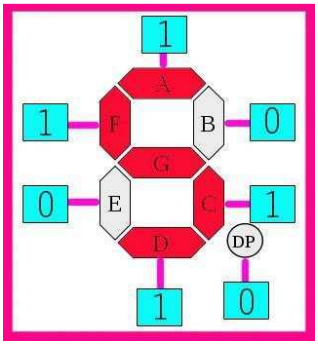
یعنی:

segment[1]=0x3F
segment[2]= 0x06
.
.
.
segment[10]=0x6F

حال وقت آن رسیده تا ببینیم عبارات مذکور ({0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F}) در بالا به چه معنا هستند؟

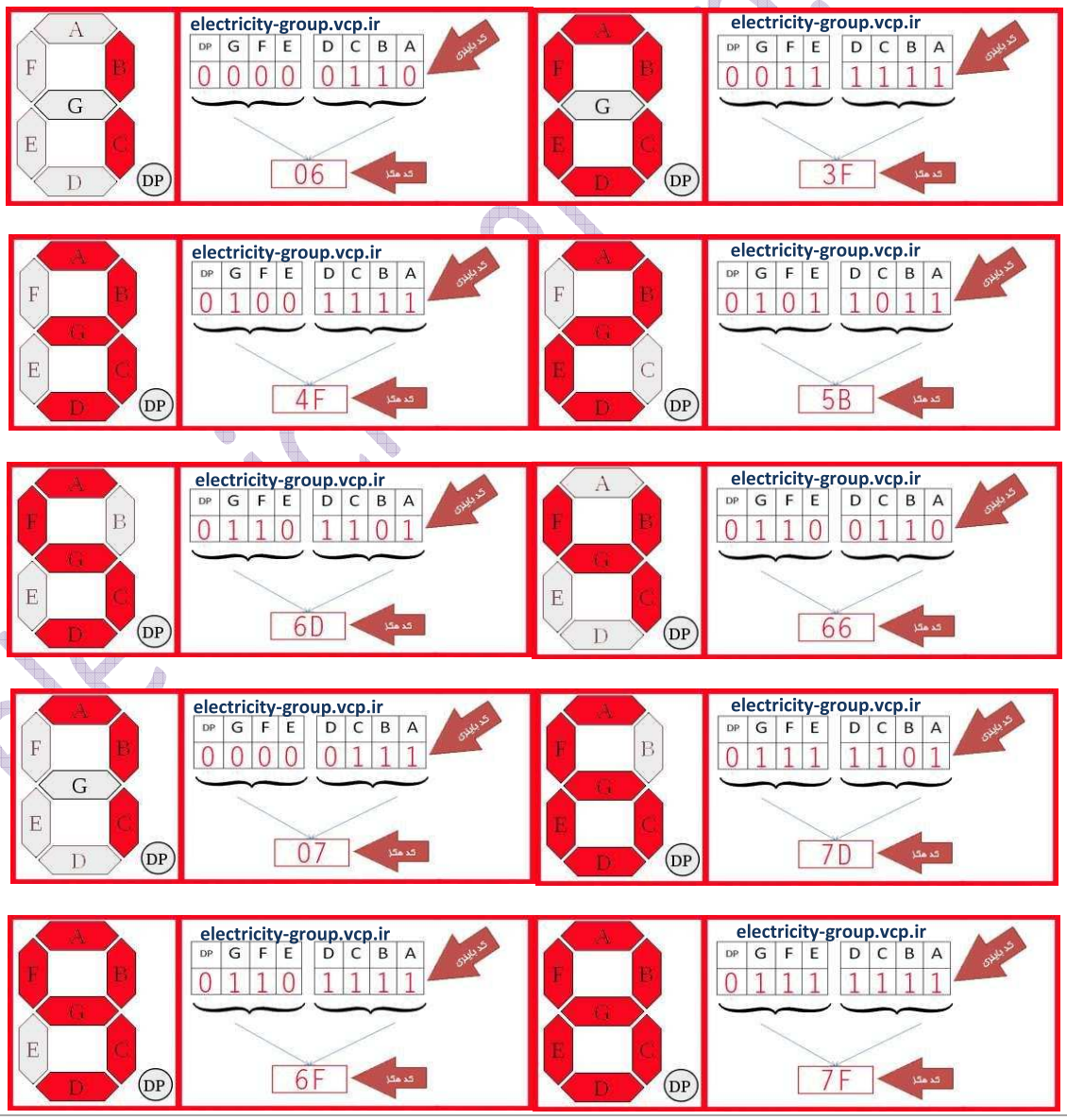
همانطور که گفته شد پایه های سون سگمنت مستقیما به میکرو متصل می گردند و با دریافت عبارت منطقی ۱ چراغ قطعه موردنظر روشن میشود.

به عنوان مثال برای نمایش عدد ۵ روی نمایشگر باید همانند شکل زیر عمل کنیم و باید برای ۵ قطعه روشن عبارت منطقی ۱ و برای سه قطعه خاموش عبارت منطقی ۰ را بفرستیم.



حال باید برای روشن کردن تعدادی از چراغ های سون سگمنت با راهبردی خاص برای نمایش عدد و یا کاراکتری خاص کدی به زبان میکرو تولید کنیم.

اگر پایه های سون سگمنت را به ترتیب زیر از A تا G و نقطه را با DP نامگذاری نماییم برای نمایش هر عدد توسط کدهای باینری (یا هگز) که همان کد قابل فهم میکروست کدهای زیر تولید می گردد.



کدهای مربوط به حالات مورد نیاز را پیدا کردیم. وقت آن رسیده که کدها را در برنامه قرار دهیم. در صورت تعریف کدها به صورت باینری (دستگاه مبنای ۲) در ابتدای کد 0b و در صورتی که کدها به صورت هگز (دستگاه مبنای ۱۶) باشند در ابتدای کد 0x را باید قرار دهیم.

توجه: همانطور که در کدها مشخص شده قبل از هر کد 0x قرار گرفته است،که یعنی ما از کدهای تبدیل شده به صورت hex استفاده می نماییم.

یکی از دلایل استفاده از این سبکِ کد کوچکتر بودن و راحتی کار با آن است، از طرفی احتمال خطا و اشتباه نیز کاهش می یابد.

عدد	کدهای باینری	کدهای هگز
0	0b00111111	**0x3F**
1	0b00000110	**0x06**
2	0b01011011	**0x5B**
3	0b01001111	**0x4F**
4	0b01100110	**0x66**
5	0b01101101	**0x6D**
6	0b01111101	**0x7D**
7	0b00000111	**0x07**
8	0b01111111	**0x7F**
9	0b01101111	**0x6F**

در نهایت در سطرآخر، دو متغییر i و j به عنوان int تعریف و به آنها مقدار اولیه صفر را مقداردهی می شود.

```
for(i=0;i<=9;i++)
{
PORTD=segment[i];
PORTA=segment[j];
delay_ms(50);
if(i>=9)
{
j=j+1;
if(j>=10)
j=0;
        }
}
```

همانطور که در بالا مشخص است در ابتدای کد یک حلقه for تعریف شده است که به ازای i=0 تا i=9 به ترتیب i را یکی یکی افزایش می دهد.

در خط سوم PORTD=segment[i] تعریف شده است که در هر مرحله مقدار i را با تغییر مقدار i در حلقه for در خود جایگزین می نماید و PORTD یعنی یکان عدد دو رقمی مورد نظر را با یکی از مقادیر که همان عددهایی هستند که با کدهای هگز برای segment[i] تعریف کردیم برابر می کند.

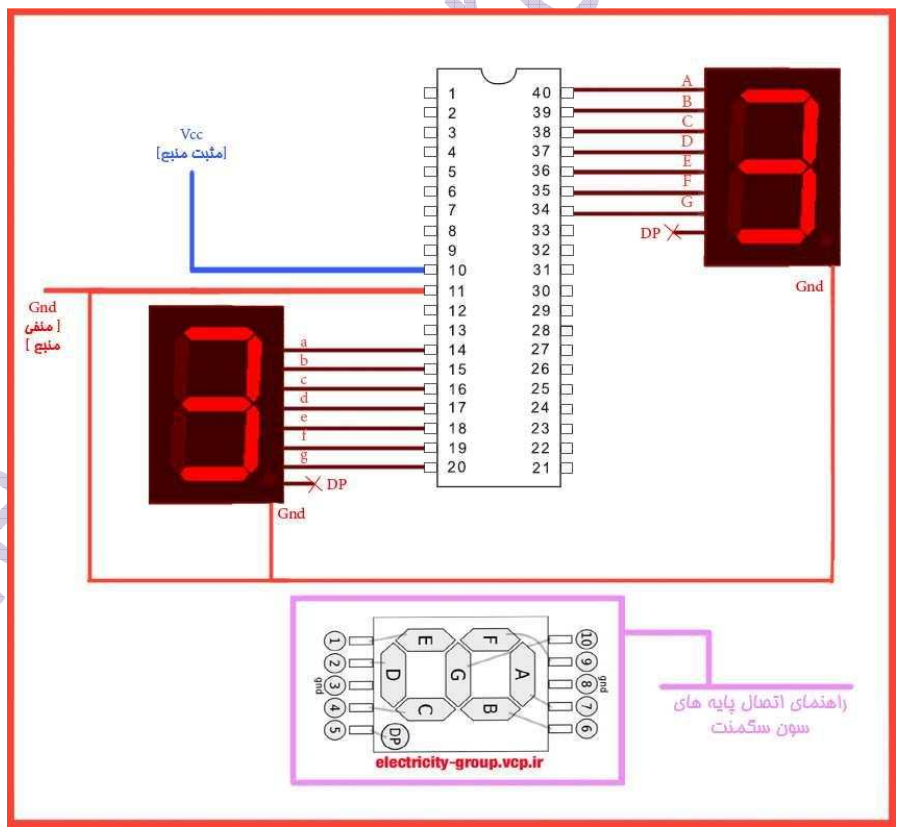
در خط چهارم نیز همان مراحل خط دوم تکرار می شود با این تفاوت که متغییر j است و j همیشه صفر خواهد بو زیرا در ابتدا مقدار دهی اولیه کردیم (int i=0 , j=0;)، مگر شرط خط ششم برقرار شود که اگر i بیش از 9 شود به مقدار j نیز یک واحد افزوده می گردد. در این شرایط j با هر بار شمارش i از ۰ تا ۹ یک واحد افزایش می یابد.

حال باید شرطی برقرار کرد که وقتی j به مقدار ۹ رسید برابر صفر شود و شمارنده دوباره از صفر شروع به شمارش نماید. شرط مورد نظر همان شرطی که در خط ۹ و ۱۰ قرار گرفته است. یعنی به محض آنکه j بیشتر یا مساوی ۱۰ شد برابر صفر قرار گیرد و دوباره از صفر شروع به شمارش نماید.

در این جا بررسی تمام فرایند برنامه نویسی به پایان رسید.

• شروع به کار عملی : بستن مدار

برای بستن مدار از شکل زیر استفاده نمایید.



این هم از چند نمونه عکس بسته شده ی مدار.