آموزش برنامه نویسی

میکروکنترلرهای PIC

به همراه آموزش نرم افزار Proteus

آموزش مفاهیم رباتیک
آموزش کار با حسگرها و ثبات ها
آموزش نصب نرم افزار MPLAB
آموزش نصب نرم افزار Proteus
کار با موتورهای الکتریکی و LCD
پیکربندی PORTA و TRISA
و ...

گروه نویسندگان:

مجید انجیدنی - رضا درودی - سمیه مداح

محمدرضا رضایی - مریم طوطی - بهاره مدیری - زهرا معزز - مجید مرادی

هوالعلیم

آموزش برنامه نویسی

میکرو کنترلرهای PIC

به همراه آموزش نرم افزار Proteus

گروه نویسندگان

مجید انجیدنی – رضا درودی – سمیه مداح

محمد رضا رضایی – مریم طوطی – بهاره مدیری – زهرا معزز – مجید مرادی

گروه کامپیوتر دانشگاه پیام نور نیشابور

زمستان 1388

فهرست مطالب

• مقدمه

با پیشرفت علم و کشیده شدن جوامع به سمت و سوی ماشینی شدن، شاید تصور جهان آینده بدون حضور اجزای الکتریکی تا اندازه ای محال باشد. چرا که علم الکترونیک به عنوان علمی نوین روز به روز بر سیطره خود بر زندگی مردم می افزاید تا آنجا که کارهای روزمره انسان به آن وابستگی انکار ناپذیری پیدا کرده است.
علم رباتیک نیز به عنوان شاخه ای از الکترونیک، این روزها کاربردهای وسیعی پیدا کرده است.
شاید با شنیدن نام روبات، در تصور بسیاری از افراد، تصویر ماشینی انسان نما که قادر به انجام کارهای خارق العاده است، تداعی شود. اما باید بگوییم که گرچه ساخت روبات های انسان نما، یکی از اهداف علم روباتیک محسوب می شود اما تنها محدود به آن نسیت. چرا که امروز شاهد کاربرد رباتیک در صنعت به عنوان قسمت های خودکار دستگاههای مختلف اعم از بازوهای هوشمند، سیستم های کنترل و ناوبری هوشمند و ... هستیم.
در دانشگاهها نیز انواع مختلفی از روباتها طراحی و ساخته می شوند که از آن جمله می توان به روباتهای فوتبالیست ، بسکتبالیست، امدادگر، مسیریاب و ... اشاره کرد.
در نمای کلی، علم روباتیک به دو شاخه اصلی تقسیم می شود که عبارتند از : شبیه سازی نرم افزاری و ساخت سخت افزاری.
در شبیه سازی نرم افزاری که به منظور افزایش توانایی برنامه نویسان در نوشتن دستورالعمل های مورد نیاز ، طراحی شده است، برنامه نویسان، ابتدا برنامه خود را نوشته و سپس در محیط مجازی بر روی روبات های شبیه سازی شده امتحان می کنند.البته این روش به نوبه خود طرفداران خاص خود را دارد و به دلیل هزینه اندک، کاربران بسیاری به آن رو آورده اند.در حال حاضر نیز مسابقات مختلفی در کشورهای مختلف برگزار می گردد که از آن جمله می توان به مسابقات روبوکاپ در قسمت شبیه سازی اشاره کرد.
شاخه دوم رباتیک که ساخت سخت افزاری روباتهاست مبحث ما در این کتاب است. ما در این کتاب سعی داریم با معرفی قطعات مختلف الکترونیکی که در ساخت روبات از آنها استفاده می کنیم، و نیز با آموزش نحوه برنامه نویسی روباتها، مخاطبان را در ساخت روبات یاری کنیم.
ما در این کتاب در ابتدا با معرفی سنسورهای مختلف که در روباتهای گوناگون از آنها استفاده می شود کار خود را آغاز کرده و سپس به آموزش نرم افزار پروتئوس – که شبیه ساز مدارات الکترونیکی می باشد – و همچنین نرم افزار Mplab که محیط برنامه نویسی برای میکروکنترلرهای PIC است . می پردازیم. پس از آن به معرفی قطعات جمله LCD و موتورهای الکتریکی پرداخته و نحوه کار و برنامه نویسی آنها را نیز تشریح خواهیم کرد.
کتاب حاضر، حاصل تلاش گروه رباتیک دانشگاه پیام نور نیشابور است و امید آن داریم که این کتاب بتواند مقدمه ای باشد برای آشنایی شما عزیزان با مباحث علم روباتیک و زمینه ای باشد برای پیشرفت روز افزون شما عزیزان در این شاخه از علم الکترونیک.

رضا درودی
زمستان 87

فصل اول

1. مفاهیم اساسی و عمومی

1-1 نوار منو و نوار ابزارها

سازماندهی منوها و نوار ابزارها در این نرم افزار به گونه ای است که با ویندوز کامپیوتر سازگاری کامل دارد. تمامی عملیات عمومی از قبیل کپی و الصاق و غیره همانند پنجره های ویندوز در این نرم افزار گنجانده شده است و افرادی که با محیط ویندوز آشنایی کامل دارند به سهولت خواهند توانست که از این امکانات استفاده کنند.

1-1-1 نوار منو Menu Bar

این نوار در قسمت بالای پنجره ویرایش (محیط کاری) قرار داشته و این امکان را به کاربر می دهد که رفتار و شکل برنامه یا پنجره را تغییر دهد یا کنترل نماید. تمام فرمان های اجرایی توسط کلیدهای میانبر و آیکون های نوار ابزارها در این منوها گنجانده شده اند.

File View Edit Library Tools Design Graph Source Debug Template System Help

1-1-1-1 منوی File

منوی File دارای قسمت های زیر است که به تعریف چند مورد از آن ها می پردازیم.

New Design
Load Design
Save Design
Save Design As …
Import Section …
Export Section …
Export Graphics

Mail To …
Print …
Print Setup …
Set Area
Exit

- **Load Design**:

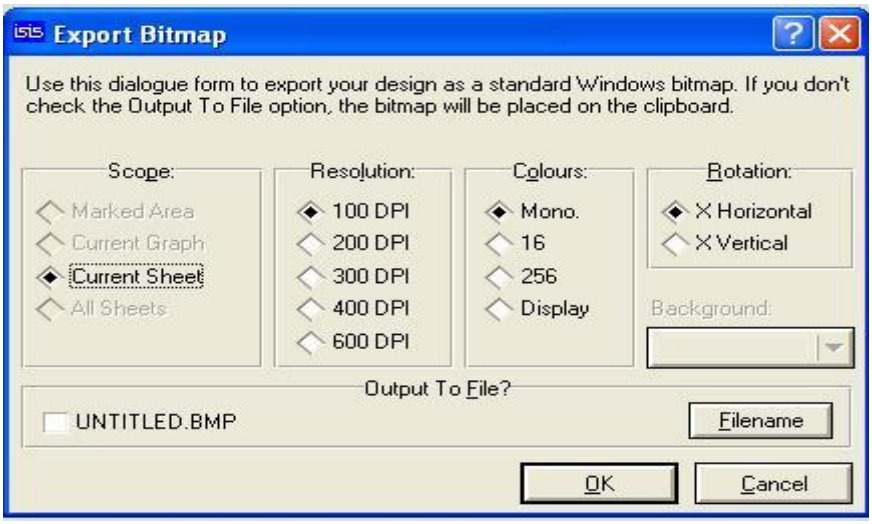
برای باز کردن یک فایل با پسوند مربوط به Preteus می باشد بکار برده می شود.

- **Improve Bitmap**:

برای وارد یک تصویر Bitmap در مدار بعنوان قسمتی از مدار که در نـرم افزارهـای دیگـر انجـام داده باشید از این گزینه می توانید استفاده کنید.

- **Export Graphics** :

برای ذخیره مدار به صورت تصویر می توان از این گزینه استفاده کرد . تصویر خروجی می تواند به صورت یک فایل Bitmap و یا یک فایل DFX که با نرم افزار AtuoCad باز می شود باشد.
اگر بخواهید خروجی مدار به صورت Bimap داشته باشید با کلیک برروی آن پنجره ذیل باز می شود.



شکل 1-1– پنجره ذخیره مدار

قسمت های مختلف این پنجره عبارت است از:
-Scop : برای انتخاب صفحه ای که می خواهید از آن خروجی Bitmap داشته باشید.
-Rosolution: برای انتخاب وضوح تصویر می توانید گزینه مورد نظر را انتخاب کنید.
-Rotation: برای چرخش تصویر خروجی مدار می توانید از این گزینه استفاده کنید.
-Print To File: با استفاده از گزینه File Name سیر و نام فایل خارجی را مشخص می کنید و بـرای ذخیره شدن تصویر در فایل مورد نظر باید Checkbox علامت دار باشد.

1-1-1-2 منوی View

این منوی دارای قسمت های زیر می باشد:

Redraw , Grid , Orgin , X Cursor , Snap 10th ,Snap 50^{th}, Snap 100^{th} , Snap 500^{th} , Pan , Zoom In , Zoom Out, Zoom All, Zoom to Area , Toolbars…

حال به اختصار به تعریف چند گزینه می پردازیم:

- **Redraw: برای Refresh کردن صفحه مورد استفاده قرار می گیرد.**
- **Grid: برای فعال کردن یا غیر فعال کردن تمایش نقاط راهنمای روی صفحه بکار برده می شود که علامت r به معنی فعال بودن آن می باشد.**
- **Origin: در این نرم افزار اطلاعات مختصری از المان موجود در مدار که نشانگر موس بر روی آن است در قسمت پایین علامت موس نمایش داده می شود.**

حال اگر مدار خیلی نزدیک به هم و فشرده باشد و بخواهید تنها اطلاعات قسمت مورد نظر را در نوار وضعیت ببینید با فعال کردن گزینه ی Origin نشانگری ظاهر می شود که می توان ان را به محل مورد نظر انتقال داد و با یک چپ کلیک تنها اطلاعات آن نقطه را در نوار وضعیت در قسمت پایین مشاهده کرد.

- **Snap: در قسمت بعد منوی View گزینه هایی با نام Snap وجود دارد که هرکدام از آنها فاصله های بین نقاط راهنمای روی صفحه را تغییر می دهند.**
- **Pan: با انتخاب این گزینه و یا استفاده از کلید F5 نشانگری ظاهر می شود که به وسیله یک کلیک چپ می توان نقطه ای را به عنوان مرکز تصویر انتخاب کرد.**

1-1-1-3 منوی Edit

این منو دارای قسمت های زیر می باشد:

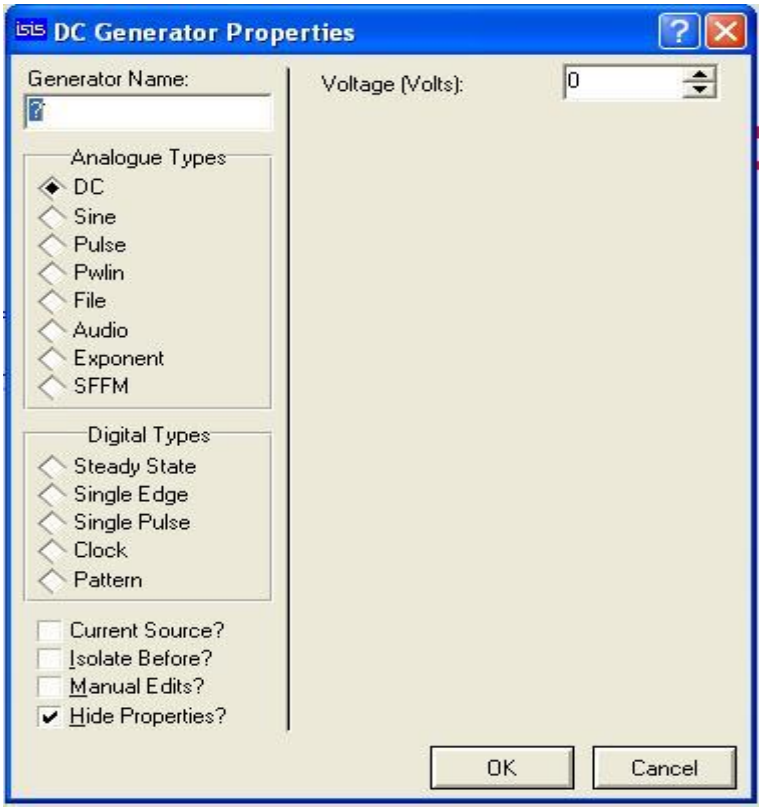
Undo
Redo
Find and Edit Component…
Cut to clipboard
Copy to Clipboard
Past from clipboard
Send to Back
Bring to Front
Tidy

حال به اختصار به تعریف چند گزینه می پردازیم:

Find and edit Component: با انتخاب این گزینه شکل ذیل ظاهر می شودکه با تایپ المان موجـود بر روی صفحه در قسمت Component می توان به پنجره خصوصیات دست یافت.

شکل 1-2– پنجره جستجوی المان ها

در پنجره خصوصیات می توان مقدار المان و نام المان و دیگر خصوصیات المان را تغییر داد. شکل ذیل:



شکل 1-3– پنجره خصوصیات

Past From Clipboard :برای چسباندن قسمت برش داده شده از این گزینه استفاده می شود. بعد از این کار نام المان یا المان های کپی گرفته شده را باید تعریف کنید در غیر اینصـورت در شـروع اجـرای برنامه پنجره خطا ظاهر می شود.

Tidy: اگر قطعات انتخابی در صفحه میز کار استفاده نشده باشد با انتخاب از محل لیست شدن قطعات و با استفاده از گزینه Tidy می توانید آنها را از لیست انتخابی حذف کنید.

شکل 1-4– محل لیست شدن قطعات

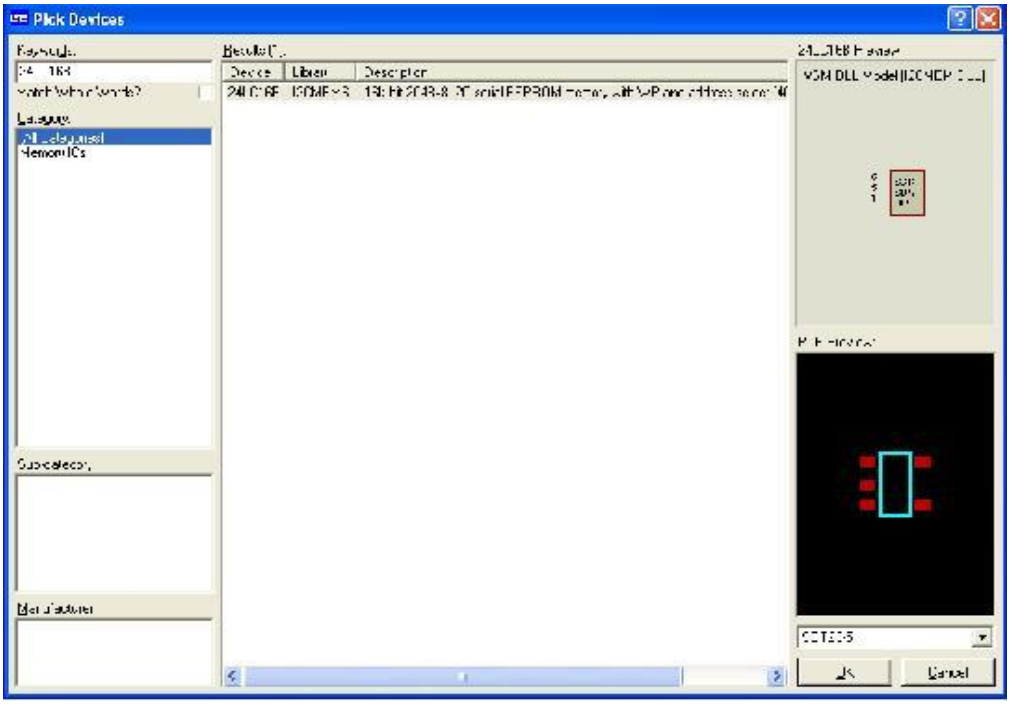
4-1-1-1 منوی Library

دارای قسمت های زیر می باشد:

Pick Device/Symbol
Make Device…
Packaging Tool…
Store Local Object …
Decompose
Compile to Library …
Autoplace Library …
Verify Packaging …
Library Manager

حال به اختصار به تعریف چند گزینه می پردازیم:

Pick Device/Symbol: با استفاده از این گزینه می توان قطعه مورد نظر را برای اضافه کردن به لیست قطعات پیدا کرد. با انتخاب این گزینه شکل زیر ظاهر می شود:

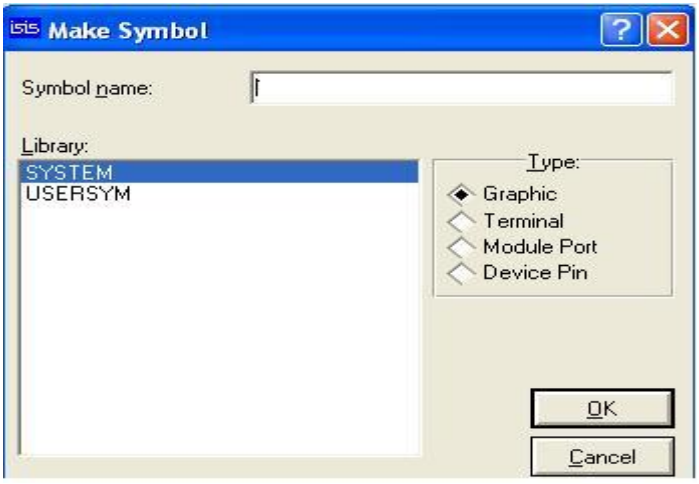


شکل 1-5– پنجره انتخاب قطعه

Make Device: به وسیله این گزینه می توانید یک قطعه را بسازید.

Make Symbol: برای ساخت یک سمبل با استفاده از ابزار طراحی , سمبل مورد نظر را طراحی کنید سپس از این گزینه می توانید سمبل و یا همان اشکا ل گرافیکی طراحی شده را در Library ذخیره کنید.

با استفاده از این گزینه شکل زیر ظاهر می شود. در قسمت Symbol Name نام را وارد کنید و در قسمت Library جایی که می خواهید سمبل در آنجا ذخیره شود را انتخاب کنید. در قسمت Type نیز یکی از انواع سمبل را انتخاب و سپس Ok کنید.



شکل 1-6– پنجره ساخت سمبل ها

Decompose: برای تغییر شمای گرافیکی قطعات استفاده می شود . با انتخاب این گزینه قطعه ی مـورد نظر به اجزای ساخته شده ی اولیه تفکیک می شود و می توانید آنها را تغییر دهید و سـپس بـا اجـرای گزینه **Make Device** دوباره تراشه را به صورت **Package** در آورید.

Verify Packaging: برای اینکه از صحت طراحی **Package** خود مطمئن شوید از این گزینه اسـتفاده کنید. در صورعدم وجود خطا , پیام **No Error Packaging Found** ظاهر می شود.

مراحل طراحی شکل تراشه:

1- به وسیله ابزار طراحی  شکل تراشه را طراحی کنید.

از قسمت **2D Graphic Box** و از لیست موجود , گزینه ی **Component** را انتخاب کنید. سـپس بـا نگه داشتن کلیک چپ موس شکل تراشه را بکشید.

2-حال از قسمت **Device Pin** نوع پایه (**Pin**) را انتخاب کرده و پایه های تراشه را با نگه داشتن کلیک چپ موس طراحی کنید.

3-حال با انتخاب پایه(به وسیله راست کلیک) و چپ کلیک کردن شکل زیر ظاهر مـی شـود. در ایـن پنجره در قسمت **Pin Name** نام پایه ی مورد نظر و در قسمت **Default Pin Number** شـماره پایـه مورد نظر را تایپ کنید.

در قسمت **Electrical Type** نوع پایه را مشخص کنید.

انواع پایه عبارتند از:

PS:پایه بدون مقاومت — **IP**:پایه ورودی

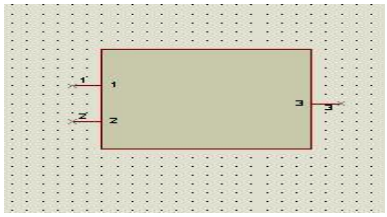
OP:پایه خروجی — **IO**:پایه ورودی/خروجی

Tristate:**TS** — **PU**:پایه **Pull up**

PD: پایه **Pull down** — **PP**:پایه ولتاژ

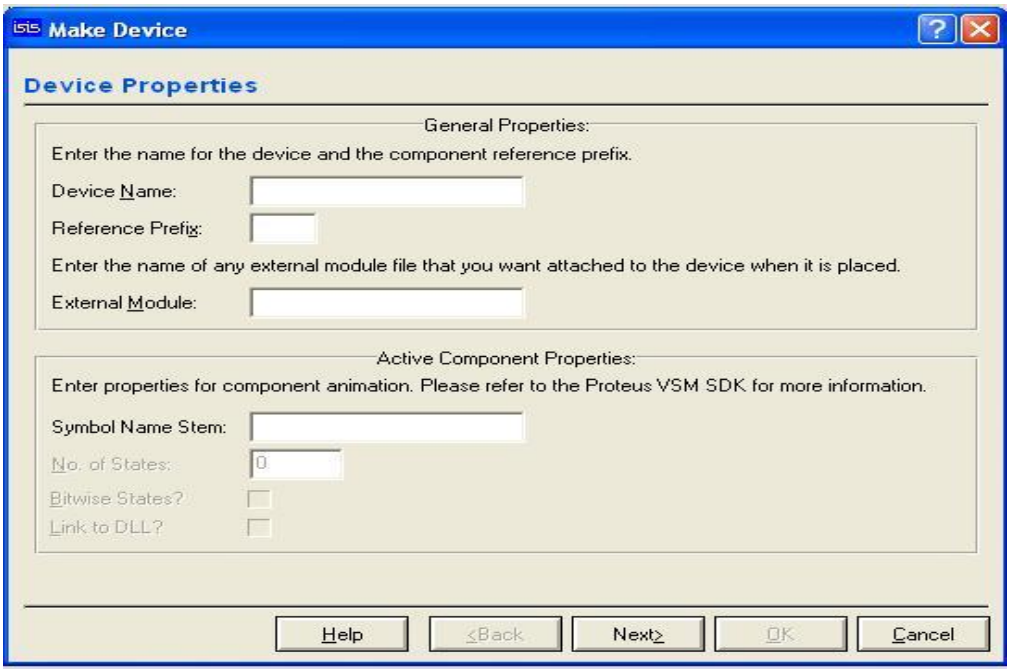
شکل 1-7– پنجره طراحی شکل تراشه

به طور مثال طراحی به صورت زیر انجام شده است:

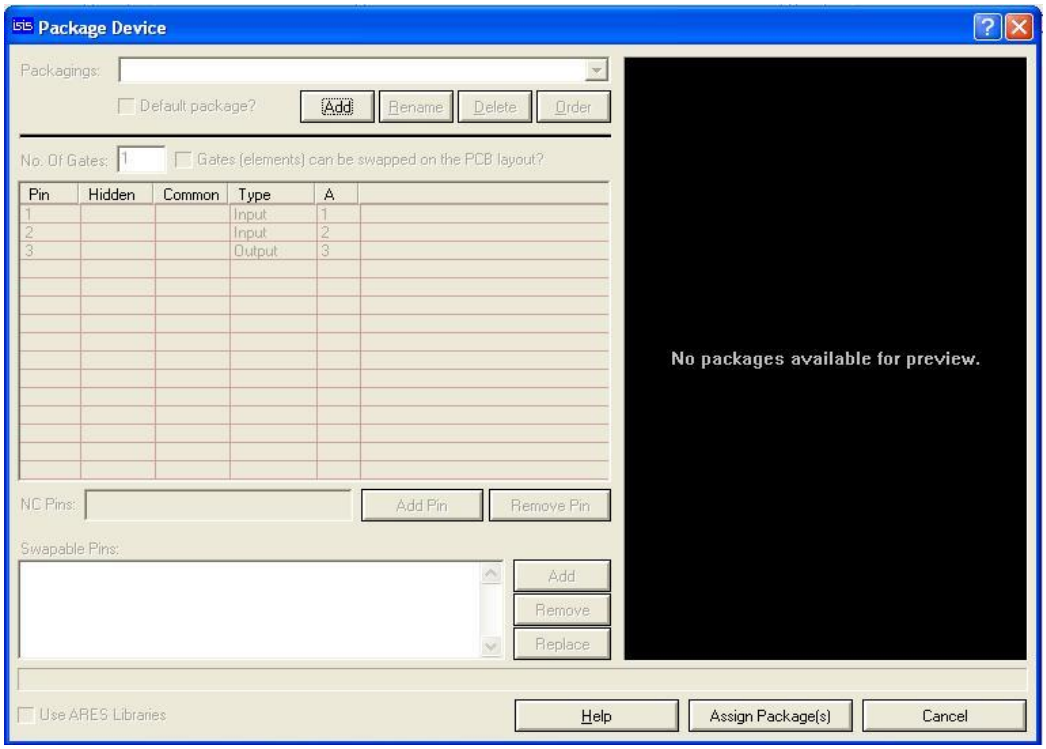


4-حال تمام طراحی را با استفاده از کلیک چپ موس انتخاب کنید (قرمز رنگ می شود) و سپس از منوی Library گزینه Make Device را انتخاب کنید . شکل (1) زیر ظاهر می شود. با زدن کلید NEX پنجره شکل (2) ظاهر می شود. در این پنجره کلید ADD/EDIT ر بزنید تا شکل (3) ظاهر شود.در این پنجره همانطور که می بینید نام ونوع پایه های تراشه نمایش داده شده است . حال برای انتخاب نوع Package , کلید Add را بزنید.پنجره شکل (4) ظاهر می شود. در این قسمت انواع بسته های تراشه وجود دارد. یک مدل از این بسته ها را متناسب با طراحی انجام شده است انتخاب کنید. پس از انتخاب بسته مورد نظر پنجره ی شکل (5) ظاهر می شود.در این پنجره در قسمت Nec Pin پایه هایی که لازم ندارید و زیادی هستند را مشخص کنید حال بر روی Assign Package و Next کردن صفحه ی بعدی صفحه Component Property & Defination ظاهر می شود که در آن می توانید تغییرات مورد نظر دیگررا نجام دهید . با زدن Next می توانید در صفحه ی بعدی , اطلاعات مورد نظر را وارد کنید در این صفحه می توانید فایل help را برای تراشه تعریف کنید.

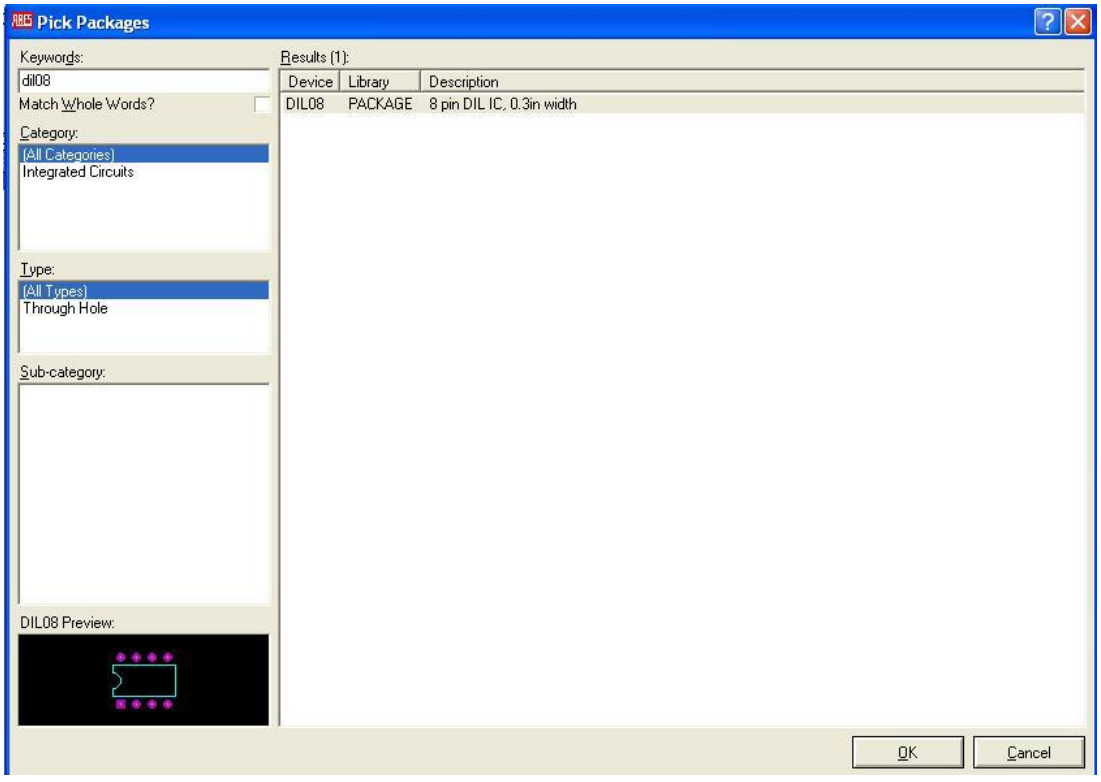
در صفحه بعدی باید مکانی را که می خواهید تراشه در آن قسمت از Library ذخیره شود را انتخاب کنید . و با زدن کلید Ok این مرحله پایان می پذیرد.



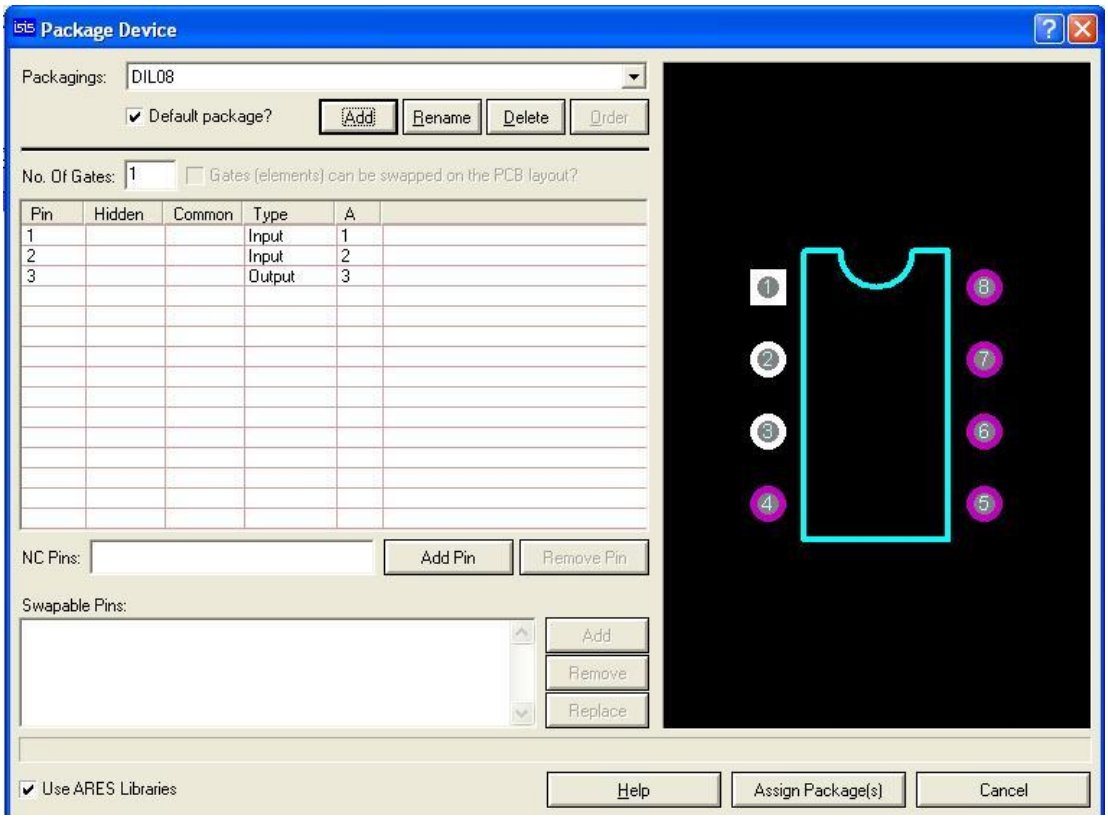
شکل 1-8– پنجره تکمیل طراحی



شکل 1-9– ادامه پنجره طراحی



شکل 1-10– ادامه پنجره طراحی



شکل 1-11–ادامه پنجره طراحی

5–در این مرحله نوبت به طراحی مدار داخلی تراشه می رسد.

تراشه ساخته شده را در جایی که در Library ذخیره کرده اید انتخاب کنید و بر روی صفحه قرار دهید . سپس با یک راست کلیک (برای انتخاب تراشه)و سپس چپ کلیک بر روی تراشه مورد نظر در پنجره ظاهر شده نام تراشه را انتخاب کنید.حال از منوی Design گزینه Go To Sheet را انتخاب کنید و در پنجره اهر شده به زیر شاخه مورد نظر که با نام تراشه است بروید . یک صفحه خالی نمایش داده می شود . در این صفحه می توانید طراحی داخل تراشه را انجام دهید. نکته حئز اهیت این است که تعریف ورودی یا خروجی باید با توجه به نامگذاری انجام شده ی پایه های تراشه صورت گیرد.

حال با استفاد از گزینه Go to Sheet به صفحه اصلی رفته و ورودی و خروجی مورد نظر را اعمال و مدار را اجرا کنید.

5-1-1-1 منوی Tools

دارای قسمت های زیر ی باشد :

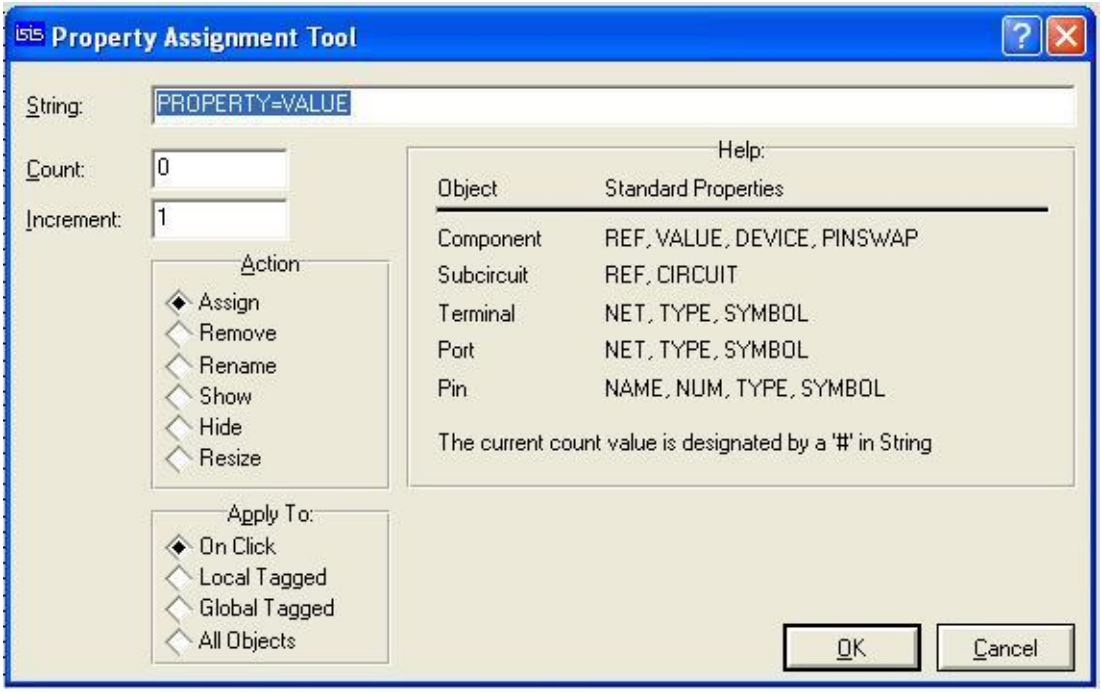
Real Time Annotation

Real Time Snap
Wire auto Router
Search and Tag …
Or Search and Tag …
And Saerch and Tag …
Property Assignment Tool …
Global Annotator …
ASCII Data Import …
Bill of Materiels
Electrical Rule check …
Netlist Compiler …
Model Compiler…
Netlist to AREAS
Backannotate from AREAS

حال به اختصار به تعریف چند گزینه می پردازیم:

Wire Auto Router:در صورت فعال بودن این گزینه , نرم افزار می تواند سیم ها را به طور خـود کـار با زاویه استاندارد 90 درجه رسم می کند و در هنگام سیم کشی کافیست سیم ها را از پایه مبدا به پایـه مقصد وصل کنید.

Property Assignment Tools:به وسیله این گزینه مشخصه های ابزار را تغییر دهید .
در صورت کلیک بر روی ای گزینه شکل زیر ایجاد می شود . در قسمت Action می توان نوع عمل و در قسمتَ Aplly To می توانید مشخص کنید که تغییر بر روی کدام قطعات و بر چسب ها انجام شود.

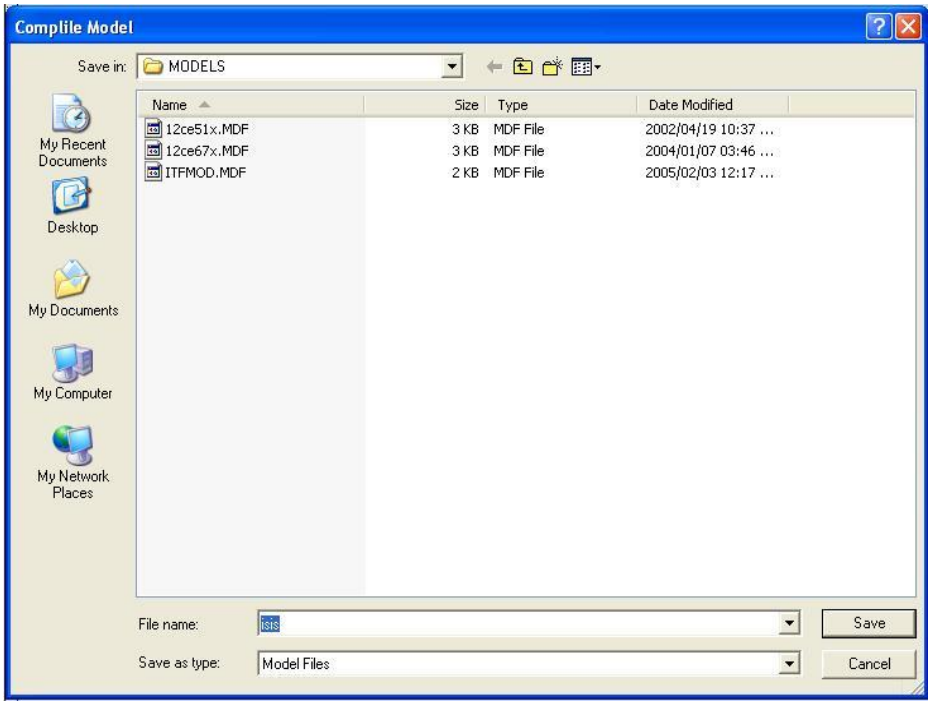


شکل 1-12– پنجره اعمال تغییرات

Bill of Materials:برای تهیه گزارش از مدار مورد استفاده قرار می گیرد.
گزارش گیر از مدار دارای چهار مدل خروجی است :
1-خروجی HTML
2-خروجی ACCII
3-خروجی Compact CSV
4-خروجی Full CSV

Electical Rule Check: این گزینه مدار را از لحاظ قوانین ا لکترونیک چک می کند و در صورت وجود خطا , خطاهای موجود را نمایش می دهد.

Model Compiler: برای انتخاب مدل کامپایلر برنامه , مورد استفاده قرار می گیرد. با کلیک بر روی این گزینه پنجره زیر نمایش داده می شود که می توانید یکی از کامپایلر های موجود را انتخاب کنید.



شکل 1-13- پنجره انتخاب مدل کامپایلر

6-1-1-1 منوی Design

منوی design دارای قسمت های زیر است :

Edit design properties…
Edit sheet properties…
Edit design notes…
New sheet
Remove sheet
Goto sheet…
Zoom to child
Exit to parent
Root sheet1
Root sheet2
Root sheet

حال به اختصار به تعریف چند گزینه می پردازیم :

New sheet:
به وسیله ی این گزینه یک میز کار جدید در همان فایل طراحی باز میشود و می توانید طراحی خود را در آن صفحه انجام دهید بدین صورت که با استفاده از گزینه های input, output و...می توانید بین دو صفحه ارتباط برقرار کنید .
به نامگذاری ورودی و خروجی توجه کنید .

Goto sheet :
بری سوئیچ کردن بین صفحات مورد استفاده قرار می گیرد .
Zoom to child :

برا ی ورود به زیر شاخه ها مورد استفاده قرار می گیرد .

Exit to parent :

برای سوئیچ به صفحه ی بالاتر و اصلی مورد استفاده قرار می گیرد.

1-1-1-7 منوی graph :

منوی **graph** دارای قسمت های زیر است :

Edit graph… ctrl+e
Add trace… ctrl+a
Simulate graph space
Viewlog ctrl+v
Exportdata
Resort est

برای انتخاب یک گراف ابتدا گزینه  را کلیک کنید و سپس از لیست موجود یکی از گراف ها را انتخاب و با نگه داشتن کلیک چپ موس گراف انتخابی را رسم کنید

Add trace:

برای انتخاب پروپ و ورودی و خروجی و رسم نمودار ان ها از این گزینه استفاده کنید .

حال در پنجره **add trace** می توانید پروپ یا ورودی یا خروجی مورد نظر را با توجه به نام ان ها انتخاب کنید .

Export graph:

برای خروجی گرفتن از گراف مورد استفاده قرار می گیرد .

1-1-1-8 منوی source :

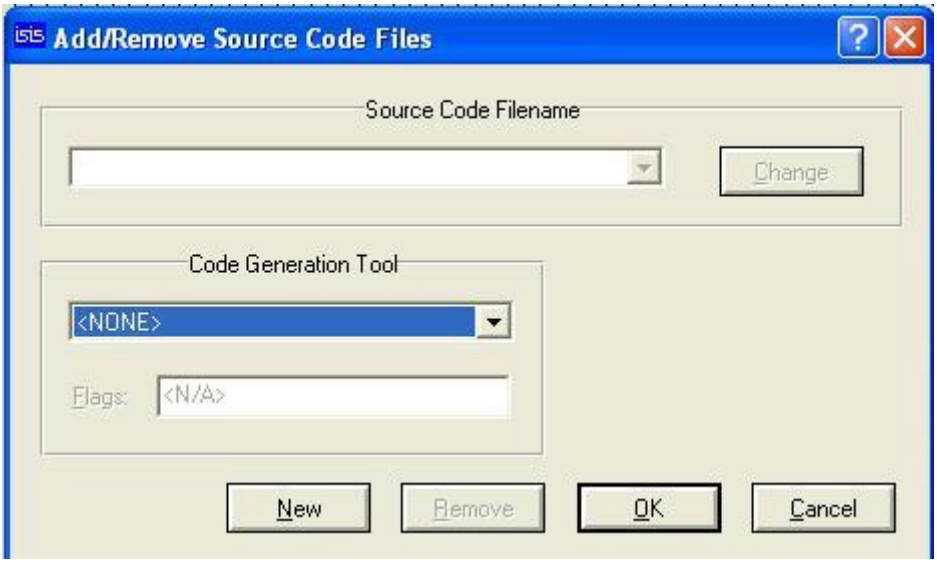
منوی **source** دارای قسمت های زیر است :

Add /remove source files…
Define code generation tools…
Setup external text editor…
Build all

حال به اختصار چند گزینه را به اختصار توضیح می دهیم:

Add/remove source file:

برای اضافه کردن یک **source** استفاده می شود در پنجره ظاهر شده با استفاده از گزینه **new** فایل **source** را وارد کنید و سپس در قسمت **code generation tools** ابزار اسمبلی را با توجه به فایل **source** انتخاب کنید .



شکل 1-14- پنجره اضافه کردن سورس

:Setup external text editor

این گزینه برای انتخاب و تنظیم ویرایشگر متن مورد استفاده قرار می گیرد .

با استفاده از گزینه browse می توانید ویرایشگر مورد نظر خود را انتخاب کنید .

در قسمت service name نام ویرایشگر را باید وارد کنید .

در قسمت open command وsave command بایستی دستور باز شدن برنامه و ذخیره برنامه را وارد کنید .

شکل 1-15- پنجره انتخاب ویرایشگر

1-1-1-9 منوی debug :

منوی debug دارای گزینه های زیر است :

Start/restart debugging ctrl+f12

Pause animation pause
Stop animation shift-break
Step over f10
Step into f11
Step out ctrl+f11
Step to ctrl+f10
Execute f12
Toggle breakpoint f9
Clear all breakpoint ctrl+f9
Reset popup windows
Reset persistent model data
Use remote debug monitor
Tile horizontally
Tile vertically

حال به اختصارچند گزینه را توضیح می دهیم

Srart /restart debugging :

در مدار هایی که با میکرو کنترلر یا قطعاتی که نیازمند برنامه هستند کار می کنید به وسیله ی این گزینه می توانید از مقادیر ثبات ها اگاه شوید و می توانید اجرای خط به خط برنامه را مشاهده کنید .
برای اجرای مجدد برنامه باید دوباره بر روی این گزینه کلیک کنید .
در صورتیکه پس از اجرای start/restart debugging پنجره های مربوط به نمایش خط به خط برنامه و یا پنجره های مربوط به نمایش مقادیر داخلی ثبات ها و ... باز نشد می بایست از منوی debug اقدام به باز نمودن انها کنید توجه داشته باشید که پنجره های مذکور تنها زمانی نمایش داده می شوند که شبیه سازی مدار در حالت pause باشد

Execute :

به وسیله ی این گزینه می توانید مدار را اجرا یا play کنید .

Toggled breakpoint :

برای انتخاب کردن و یا ار حالت انتخاب در اوردن یک یا چند خط (به وسیله ی نقطه قرمزی در کنار انها)از این گزینه استفاده میشود .

Clear all breakpoint :

به وسیله ی این گزینه می توان تمام خط ها را از حالت انتخاب در اورد .

Watch windows :

با استفاده از این پنجره می توانید در مدت شبیه سازی از مقادیر موجود در ثبات ها و حافظه اگاه شوید و دیگر به خط به خط برنامه برای دیدن مقادیر مذکور نیست .
ابتدا مدار را play کنید و سپس از منوی debug گزینه watch window را انتخاب کنید پنجره مربوط باز می شود
با راست کلیک بر روی این پنجره ثبات ها و یا ادرس های حافظه را برا ی نمایش مقادیرشان در زمان شبیه سازی انتخاب کنید

برای انتخاب ادرسهای حافظه و ثبات ها دو راه پیش رو دارید :

1- اگر بخواهید بر اساس ادرس ثبات ها انها را انتخاب کنید باید گزینه (add item (by address را انتخاب کنید پنجره ای ظاهر می شود .

گزینه های موجود در این پنجره عبارتند از :

-memory : حافظه مورد نظر را از این قسمت انتخاب کنید .

حافظه انتخابی به میکرو کنترلر بستگی دارد.

-name: در این قسمت نام آدرس را مشخص کنید .

-address :در این قسمت آدرس مورد نظر را وارد کنید .

-data type :در این قسمت نوع داده ای که در آدرس است را مشخص کنید .

-display format :در این قسمت نحوه ی نمایش داده را مشخص کنید .

2- اگر بخواهید ثبات ها را بر اساس نامشان انتخاب کنید باید گزینه ی (add items(by name را انتخاب کنید پنجره ای ظاهر می شود از این پنجره می توانید نام ثبات مورد نظر را انتخاب نموده وبا دو چپ کلیک ان را در پنجره ی watch window وارد کنید برای خارج شدن از این پنجره گزینه ی done را انتخاب کنید

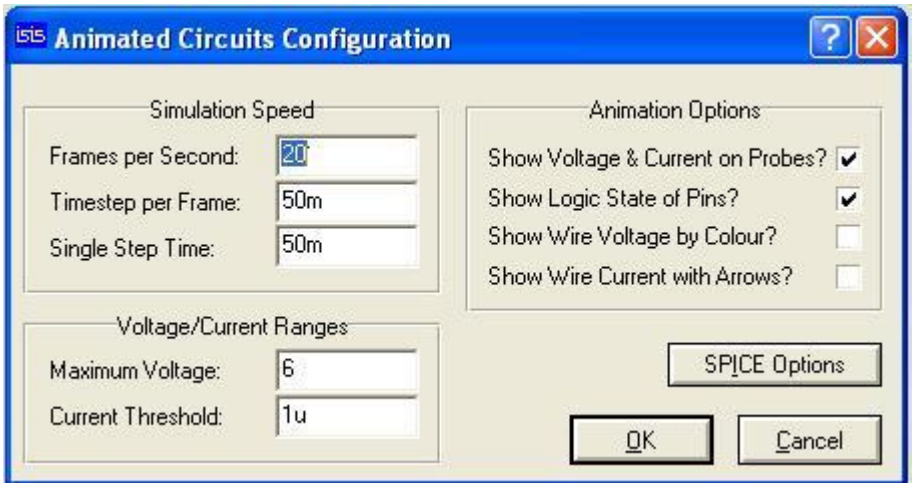
10-1-1-1 منوی system

این منو دارای گزینه های زیر می باشد:

System Info…
Text viewer
Set BOM Scripts …
Set Enviorment …
Set Paths …
Set Property Definations …
Set Text Editor …
Set Animation Options …
Set Simulator Option …
Save Simulator Options …

حال به اختصار به تعریف چند گزینه می پردازیم:

Set Animation Option: با استفاده از این گزینه می توانید تنظیمات مربوط به شبیه سازی مدار را انجام دهید . با کلیک بر روی این گزینه شکل زیر ظاهر می شود:



شکل 1-16– پنجره تنظیمات شبیه سازی مدار

قسمت های مختلف این پنجره عبارتند از:

Simulate Speed: در این قسمت می توانید سرعت شبیه سازی را مشخص کنید.

Votage/Current Option: در این قسمت می توانید دامنه ولتاژ و جریان را مشخص کنید.

Animation Options:این قسمت دارای گزینه های زیر است:

Show Logical State Of Pins?: با فعت بودن این گزینه بر روی پایه های تراشه وضعیت منطقی نمایش داده می شود.

Show Wire Voltage By Color?: با فعال بودن این گزینه ولتاژ سیم ها با رنگ مربوط نمایش داده می شود.

Show Wire Current With Arrow?: با فعال بودن این گزینه جهت جریان سیم ها با فلش نمایش داده می شود.

1-2 نوار ابزارها Tool Bars

نوار ابزار در اکثر پنجره های ویندوز قابل دسترسی می باشد و دکمه های قرار گرفته بر روی آن امکان دستیابی به عملیات متداول و سریع را فراهم می کنند. این نوار ابزارها می توانند به روش کشیدن و رها کردن(Drag and Drop)به چهار گوشه پنجره Proteus جابجا شوند.

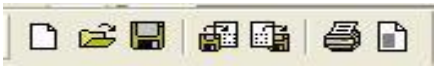
نوار ابزارها در بخش Isis , Proteus به سه قسمت اصلی Mode Selector Command, Orientation تقسیم بندی شده اند که در زیر توضیحاتی برای هر مورد ارایه شده است.

- **نوار ابزارهای دستورات Command Toolbars**

ابزارهایی که به صورت پیش فرض در بالای صفحه نمایش واقع شده اند و مسیری میانبر برای دستیابی به دستورات موجود در منوی File , Edit می باشند را تحت عنوان نوار ابزار دستورات می شناسیم. در این نوار ابزار سری دستورات زیر قرار دارند.

(دستورات فایل-چاپ)File / Print Command

به جای استفاده از منوی فایل می توانید از گزینه های موجود در این نوار ابـزار (در برخـی مـوارد) اسـتفاده کنیـد و اعمالی همچون باز کردن فایل, پرینت گرفتن و ... را انجام دهید.

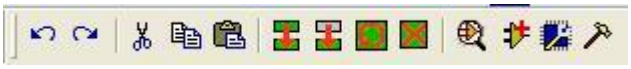


(دستورات نمایش)Display Command

با استفاده از این نوار ابزار می توانید بزرگنمایی و اعمال دیگر را بدون مراجعه به منوی View انجام دهید.

(دستورات ویرایش)...Editing Command

برای سرعت در کپی گرفتن , طراحی تراشه , حذف قطعات و ... به جای استفاده از منوی Edit می توانیـد از گزینـه های مختلف این نوار ابزار استفاده کنید.



(دستورات طراحی)...Design Command

در این نوار ابزار , قطعات و ابزار آلات طراحی بسیاری موجود است. دسترسـی بـه Library و ابـزار الاتـی همچـون اسیلوسکوپ , ولت متر و... از طریق این نوار ابزار ممکن می باشد.

- **نوار ابزارهای انتخاب حالت Mode Selector Toolbars**

این نوار ابزارها در قسمت پایین نوار ابزارهای دستورات و در گوشه چپ صفحه نمایش قرار دارند و عملیات انجام شده بر روی پنجره ویرایش را کنترل می نمایند.این نوار ابزارها به نوبه خود به سه بخش زیر Main Mode, Gadgets, 2D Graphics تقسیم می شوند.

نکته:این نوار ابزارها بر خلاف دیگر نوار ابزارها قابل پنهان شدن نمی باشند.

- **نوار ابزار چرخش(جهت دهی) Orientation Toolbar**

این نوار ابزار چرخش و انعکاس اشیا قرار گرفته بر روی محیط کاری را بر عهده دارد و می تواند جهت آنها را تغییر دهد.

چرخش(Rotation)

انعکاس(Mirror)

3-1 جعبه ویرایش Editing Box

این نوار ابزار به شما این امکان را می دهد که زاویه دلخواه را جهت چرخش اعمال نمایید اما باید به خاطر داشته باشید که ISIS فقط زوایای متعامد را می پذیرد.

زمانی که قطعه ای را انتخاب کردید آیکون های چرخش(Rotation)و انعکاس(Mirror),high light (روشن تر)خواهند شد و به صورت قرمز در خواهند آمد و این نشان می دهد که می توانید زاویه نمایش یک شی در طرح را اصلاح نمایید.

زمانیکه آیکون هاhigh light نمی باشند وظیفه آنها مشخص کردن جهت اشیا جدید اشیایی هستند که می خواهید آنها را در طرح جاگذاری نمایید.(المان هایی که پیش نمایش آنها در پنجره Overview نشان داده می شوند.)

4-1 پنجره اصلی یا ویرایش The Editing Window

زمانیکهISIS را از منوی Start ویندوز انتخاب می کنید.پنجره اصلی ISIS باز می شود که این پنجره از قسمتهای مختلفی از جمله نوار ابزارهاو پنجره ویرایش و پنجره پیش نمایش و Object Selector تشکیل یافته است پنجره Editing قسمت عمده این پنجره را به خود اختصاص می دهد. این پنجره امکان طراحی و ویرایش و شبیه سازی انواع مدارهای آنالوگ و دیجیتال را در اختیار کاربر قرار می دهد. این پنجره با یک Outline (به صورت پیش فرض آبی رنگ) مرزبندی شده است و اگر قطعه ای خارج از این Outline قرار گیرد غیر فعال خواهد بود و کاربر دیگر نمی تواند قطعه مزبور را انتخاب و ویرایش نماید.(توصیه می شود که در صورت قرار گیری قطعه در خارج از Outline از کلید Undo استفاده شود.)

1-4-1بزرگ نمایی و کوچک نمایی(Zoom in/Out)پنجره ویرایش

اگر بزرگ نمایی صفحه به صورتی باشد که نتوانید کل طرح را ببینید و یا به گو نه ای باشد که نتوانید پین های المان ها را به راحتی ویرایش نمایید.بزرگ نمایی پنجره Editing را با استفاده از کلیدF6, F7, F8 به شرح زیر تغییر دهید:

- برای نزدیکتر کردن(Zoom in)کلیدF6 را از کیبرد فشار دهید.
- از کلیدF7 می توانید برای کوچک نمایی استفاده نمایید.
- برای به تصویر کشیدن تمام طرح(Fit Document)کلید F8را از کیبرد فشار دهید.
- برای بزرگ نمایی ناحیه مخصوصی از طرح کلید Shift را پایین نگه داشته و با ماوس جعبه ای به دور ناحیه دلخواه از پنجره ویرایش بکشید.ناحیه مزبور بعد از رها کردن کلید ماوس بزرگ خواهد شد.این روش به روش Shift-Pan موسوم است.

1-4-2 پنجره پیش نمایش

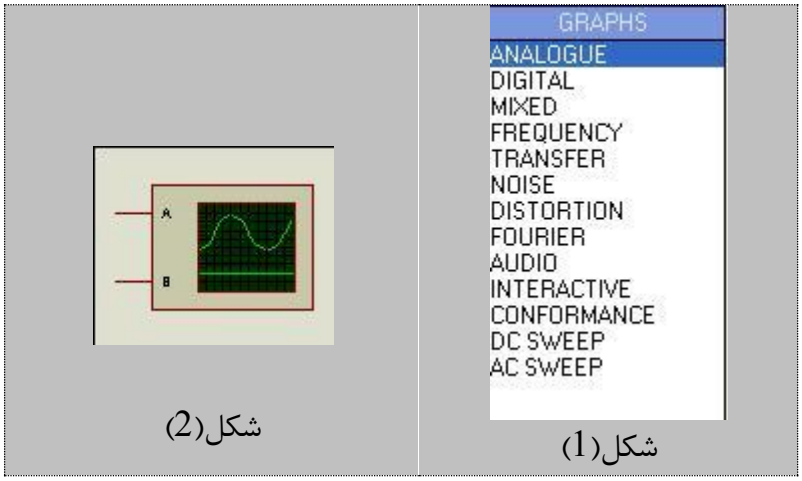
این پنجره یک نمایش خلاصه از تمام طرح را نشان می دهد.در این پنجره رنگ آبی مایل به سبز حاشیه صفحه را مشخص می کند و حاشیه قرمز رنگ ناحیه ای از طرح جاری که چنجره ویرایش قابل مشاهده است را نشان می دهد.

بیشتر اوقات پنجره پیش نمایش برای پیش نمایش الملن انتخابی برای جایگذاری استفاده می گردد.می توانید قبل از جایگذاری الملن جهت دلخواه آن را با جهت مشاهده شده روی پنجره پیش نمایش مقایسه کرده و با استفاده از دستورات mirror وorientation جهت را به جهت مطلوب اصلاح نمایید.

پنجره پیش نمایش پس از جاگذاری المان بطور اتوماتیک پاک خواهد شد.

1-5 نوار ابزار قطعات و پنجره ی آن

ابزار آلاتی که در نوار ابزار طراحی مدار قرار دارند و همچنین قطعات انتخـابی از Library در پنجـره ی نمایش داده شده در شکل (1) قرار می گیرند , شکل آنها در پنجره ی شکل(2) نمایش داده مـی شـود. همانطور که قبلا گفتیم هر قطعه برای استفاده در طراحی باید انتخاب و در لیست قرار گیرد.



شکل 1-17– پنجره انتخاب قطعات

1-6 انتخابگر اشیا (Object Selector)

object selector در سمت چپ پنجره اصلی قرار دارد و تمام قطعات انتخاب شده از کتابخانه ها انواع ژنراتورها میکروکنترلرها و غیره را به صورت لیست شده به نمایش می گذارد.زمانی که می خواهید قطعه ای را در پنجره Editing قرار دهید آن المان باید در پنجره object selector به صورت high light در آید.

1-7 قطعات در ISIS

برنامهPROTEUS با قطعات زیادی پشتیبانی می شود که تمتم این قطعات بر حسب نوع کاربرد و فعال و غیرفعال بودن در کتابخانه های بخصوصی گنجانده شده اند.با استفاده از این قطعات کاربر می تواند عملکرد یک طرح واقعی را قبل از پیاده سازی سخت افزاری در محیط خارج به صورت نرم افزاری آن را شبیه سازی کند.
یکی از کاربردی ترین این کتابخانه ها کتابخانه Micro Processor ها می باشد که مدل تمام میکروپروسسورها و میکروکنترلرها را در خود جای داده است.این مدل ها Active به صورت سازماندهی و برنامه نویسی شده اند و کاربر می تواند همانند محیط خارج این مدلها را برنامه ریزی (Programming)کرده و به کار گیرد.در ادامه نحوه انتخاب المانها و جایگذاری آنها در پنجره ویرایش را مورد بررسی قرار می دهیم.

1-8 جایگذاری المان ها

تمام المان های موجود در محیط ISIS اعم از ژنراتورها و پروب ها و میکروکنترلرها و قطعات و دستگاههای مجازی با یک متد تقریبا یکسانی جایگذاری می شوند. برای فراگیری بهتر در دو مرحله جداگانه زیر نحوه جایگذاری یک المان از سری المان های نوار ابزار و یک المان از سری المان های کتابخانه را توضیح می دهیم.

- تمامی ژنراتورها و پروب ها و میکرو کنترلرها و دستگاههای مجازی که از نوار ابزار در دسترس هستند به روش زیر جایگذاری می شوند مثلا با انجام مراحل زیر کاربر خواهد توانست که دستگاه فانکشن ژنراتور را جایگذاری نماید.

1) آیکون Virtual Instrument را از نوار ابزار Gadget Mode انتخاب نمائید.

2) از بین دستگاههای مجازی لیست شده درObject Selector دستگاه فانکشان ژنراتور را high light نمائید.(پیش نمایشی از دستگاه در پنجره ظاهر خواهد شد.)

3) در یک ناحیه خالی از پنجره Editing چپ کلیک نمائید تا سمبل فانکشن ظاهر شود.

نکته: در مرحله 2 بعد از اینکه پیش نمایشی از قطعه در پنجره Over View ظاهر شد کاربر می تواند با به کار گیری ابزارهای Orientation جهت مناسب را برای المان تعریف کند و یا قبل از جاگذاری المان زمانی که قطعه از ماوس پیروی می کند کاربر می تواند از کلیدهای + و – برای تغییر جهت المان استفاده نمائید.

- برای قرار دادن یک المان مانند Ic از کتابخانه 74HC به روش زیر عمل خواهیم کرد.

1) آیکون p موجود در Object Selector (یا کلید میانبر p) را برای باز شدن پنجره کتابخانه ها فشار دهید.

2) در پنجره ظاهر شده کتابخانه 74HC را انتخاب نمائید.

3) از لیست قطعات کتابخانه 74HC آی سی مورد نظر را دوبار کلیک کرده و پنجره را ببندید (در این مرحله نام Ic مربوطه باید در Object Selector ظاهر شود.)

4) نام Ic انتخابی را از لیست High Light, Object Selector نمائید.

5) در یک ناحیه خالی برای قرار دادن Ic چپ کلیک نمائید.

1-9 نحوه Tagging (انتخاب) ,Untagging (از انتخاب خارج کردن)و ویرایش و حذف المان های قرار گرفته بر روی پنجره Editing

المان هایی که بر روی پنجره ویرایش محیط ISIS قرار می گیرند مانند دیگر نرم افزارهای الکترونیک قابل Tagging و Untagging ویرایش و حذف شدن می باشند. در زیر روشهای مختلف Tagging و Untagging ویرایش و نحوه حذف المان ها را به طور خلاصه شرح می دهیم.

- **نحوه Tagging المان ها**

1) راست کلیک کردن بر روی المان مورد نظر از پنجره ویرایش

2) کشیدن جعبه ای به دور المان با استفاده از کلیک راست ماوس (از این روش می توانید برای انتخاب گروهی از المان ها نیز استفاده نمائید.)

- **نحوه Untagging المان ها**

1) راست کلیک کردن در یک ناحیه خالی از پنجره ویرایش

2) فشار دادن آیکون Redraw از نوار ابزار

- **نحوه ویرایش المان ها**

1) چپ کلیک کردن بر روی المان انتخاب شده

2) فشار دادن کلیدهای میانبر Ctrl + E بر روی المان مورد نظر

3) در دو روش بالا کاربر باید مستقیما بر روی المان مورد نظر اشاره می کرد که این در طرح های پیچیده و فشرده کمی مشکل است. بنابراین روش دیگری که به روش Editing With Name معروف است برای این منظور در نظر گرفته شده است. برای استفاده از این نوع ویرایش مراحل زیر را انجام دهید.

3-1) کلید E را فشار دهید.

3-2) در قسمت نام مرجع (Part ID) نام المان مورد نظر برای ویرایش را تایپ نمائید.

3-3) با فشار دادن کلید کادرok مکالمه ویرایش المان باید ظاهر شود (این روش تمام المان های موجود در صفحه جاری و زیر صفحات (Sub Sheet) را پشتیبانی می کند.)

نکته: بعد از ویرایش صفحه با تمامی محتویاتش از مرکز صفحه باز ترسیم خواهد شد. حتی اگر واقعا نخواهید ویرایشی انجام دهید.

- **نحوه حذف المان**

1) دوباره راست کلیک کردن بر روی المان انتخاب شده (یعنی کلیک راست مضاعف بر روی المان مورد نظر)

2) استفاده از کلید Delete کیبرد.

نکته: هنگام پاک کردن یک المان تمامی سیم های متصل به آن نیز پاک خواهند شد. به جز در حالی که سیم به یک گره وصل شده باشد.

- **کپی کردن المان های انتخاب شده**

برای از باز ترسیم قسمتهای تکراری یک طرح می توان مراحل زیر را انجام داد:

1) المان های مورد نظر را به طور مجزا یا گروهی انتخاب کنید.
2) روی آیکونBlock Copy از نوار ابزار Editing Command کلیک نمائید.
3) جعبه ایجاد شده را به ناحیه دلخواه از پنجره ویرایش کشیده و چپ کلیک نمائید.(المان ها کپی خواهند شد.)
4) در صورت نیاز به جایگذاری چند ین کپی از المان ها مرحله 3 را دوباره تکرار نمائید.
5) برای اتمام عمل کپی در یک ناحیه خالی راست کلیک نمائید.

نکته: زمانی که المان ها کپی می شوند نشان آنها به طور اتوماتیک به وضعیت (?) Un - Annotated در خواهند آمد و تا زمانیکه آنه را برای یادداشت گذاری اتوماتیک Automatic Annotation آماده نکرده اید به همان صورت باقی خواهند ماند.

- **جابجایی المان های قرار گرفته بر روی پنجره ویرایش**

برای جابجایی المان یا المان هایی از یک مکان پنجره ویرایش به مکان دیگر مراحل زیر را دنبال نمائید.

1) المان یا المان های مورد نظر را با استفاده از تکنیک های بیان شده High Light (انتخاب) کنید.
2) روی آیکون Block Move از نوار ابزار Editing Command کلیک نمائید.
3) جعبه ظاهر شده را به مکان دلخواه از پنجره Editing کشیده و چپ کلیک نمایید.

لازم به یادآوری است Isis در صورت ضرورت اتصالات یا قسمتهایی از آنها را که در داخل محدوده انتخاب قرار گرفته اند را بدون باز ترسیم کپی می نماید. اما اتصالاتی را که از مرزهای انتخاب گذشته اند را باز ترسیم نمی کند.

1-10 سیم کشی

شاید هنگام کار با Proteus متوجه آیکونی به نام Wire نشوید. علت این امر را باید در هوشمند بودن Isis بداند. این نرم افزار بطور خودکار تشخیص می دهد که کی می خواهید سیمی را از جایی به جایی دیگر وصل نمائید.

1-10-1 سیم کشی پین های المان ها

به طور کلی Proteus Vsm با دو روش سیم کشی در محیط Isis پشتیبانی می شود این دو روش که تحت عنوان (Wire Auto Router (WAR و Manual Wiriting معروف هستند به طور اختصار در زیر بحث شده اند.

- **سیم کشی اتوماتیک** (Wire Auto Router (WAR

در این قابلیت Proteus Vsm درصد بیشتری از پروسه سیم بندی المان ها را به عهده می گیرد تنها به انتخاب (نقطه) مکان اول اتصال (The First Connection Point) و مکان دوم اتصال (The Second Connection Point) نیاز دارد. یعنی در این روش کاربر تنها پین های اتصالی دو المان را برای Proteus Vsm معرفی می کند و Proteus پس از آن عملیات سیم کشی را به عهده می گیرد.

نکته: ویژگی با استفاده از دستور Wire Auto Router از منوی Tools فعال / غیر فعال می شود.

- **سیم کشی دستی** Manual Wiriting

1) روی نقطه اتصال اول (The First Connection Point) المان کلیک نمائید.

2) بعد از غیر فعال نمودن خاصیت (WAR) در نقاط مختلف پنجره ویرایش برای Fix کردن سیم کلیک نمائید. این روند را تا رسیدن به دومین نقطه اتصال ادامه دهید.

2-10-1 جابجایی بخشی از اتصالات

جهت جابجایی بخشی از اتصالات و سیمها می توان به صورت زیر عمل کرد:

1- در یک محدوده بخشی از اتصالات مورد نظر که می خواهیدجابجا نمائید را انتخاب نمائید.

2- روی آیکون Move کلیک نمائید.

3- بخش های انتخاب شده را می توان در راستای عمودی همانند شکل زیر جابجا نمود.

4- برای خاتمه کار کلیک چپ نمائید.

اگر اشتباهی رخ داده باشد می توانید از Undo استفاده نمائید.

1-11 سبک های متنی و گرافیکی Graphics And Text Styles

Isis این امکان را فراهم کرده که کاربر بتواند طرح شماتیک خود را تغییر دهد مثلا در صورت نیاز می توانید خطوط و بسته های توپر رنگی (Color Fills) و فونتهای متنی و دیگر Effect های لازم را در طرح خود اعمال نماید. سیستم نرم افزار بسیار قدرتمند بوده و به کاربر این اجازه را می دهد که کنترل برخی یا همه جوانب ظاهری طرح را به صورت کلی (Global) در اختیار داشته باشد.

تمام اشیا گرافیکی در Isis (بدنه و المانها و سیمها و نقاط اتصال و غیره) همگی بر اساس یک سبک گرافیکی graphical style کشیده شده اند.

یک سبک گرافیکی یک تشریح کامل از چگونگی ترسیم و پر کردن یک شکل گرافیکی نظیر یک خط – جعبه – دایره و هر چیز دیگری را بیان می دارد و شامل خصوصیات و ویژگی های سبک خط (Line Style) (dashed , dotted , solid) ضخامت و رنگ و سبک پر کردن (Fill Style) رنگ پس زمینه است. به طور مشابه تمامی برچسبها و بلوک های متنی در Isis (بر چسب ترمینالها و نام پین ها و غیره) بر حسب سبک نوشتاری Text Style نوشته می شوند. یک سبک نوشتاری توصیف کامل از چگونگی ترسیم نوشتار توصیف کامل از چگونگی ترسیم نوشتار می باشد و صفات یا ویژگی های ظاهر فونت ها (مانند Arial , Times Roman) ارتفاع کاراکترها و پهنا و رنگ آنها را در بر می گیرد.

در Isis بیشتر اشیا نظیر گرافیک دو بعدی (2D graphic) سیمها بر چسب ترمینال ها و غیره یک Local Style (سبک محلی) خاص خود را دارند و کاربر می تواند بنا به دلخواه خود آن را تنظیم نماید. بعنوان مثال یک سیم می تواند ظاهری متفاوت از سیم های دیگر داشته باشد. عبارت Local در تنظیمات محلی استفاده می شود و آن قطعه را محلی می نماید.

دیگر اشیا مانند پین ها و بدنه زیر مدارات Sub – Circuit bodies معمولا در یک سبک از پیش تعریف شده ترسیم می شوند. بنابراین اشیا فقط می توانند تنظیمات خصوصی Customised مربوط به خود را داشته باشند. بعنوان مثال زیر مدارات می توانند هر ظاهری داشته باشند ولی باید همگی آنها به طور مشابه دیده شوند.

بیشتر اشیا دارای سبک منحصر به فرد خودشان می باشند و امکان تغییر Global style در آنها وجود دارد. به عنوان مثال زمانیکه یک ترمینال را استفاده می کنید سبک بر چسب ترمینال بطور خودکار از سبک TERMINAL LABEL استفاده می نماید و یا زمانیکه می خواهید سیمی را بکار بگیرید سبک گرافیکی بطور خودکار در سبک WIRE تنظیم می شود.

در مورد اشیا گرافیک دو بعدی (2D) قضیه کمی متفاوت است. برای این اشیا نمایشگر Object Selector لیستی از سبکهای گرافیکی قابل دسترسی را نشان می دهد. و جدیدترین شی گرافیکی جاگذاری شده از سبکی که در حال حاضر انتخاب گردیده پیروی می کند.

در این جا ویژگی های جدیدی مطرح می گردد و آن این است که Local Style می تواند برخی یا همه صفات یا ویژگی های Global Style را تعقیب نماید.

مزایای داشتن سبکهای محلی و کلی و توانایی سبکهای محلی برای دنبال نمودن تمام یا برخی از ویژگی های سبک های عمومی به صورت زیر می تواند باشد.

- این اجازه داده می شود که با ویرایش سبک کلی Global Style صورت کلی طرح را اصلاح نمائید.(به جای اینکه تک تک اشیا را اصلاح کنید) همگی این اقدامات به صورت یکجا انجام خواهد پذیرفت.
- این اجازه به شما داده می شود که سمبل های کتابخانه ای تعریف نمائید. این سمبل ها به طور خودکار با ظاهر طرحی که در آن قرار می گیرند هماهنگ خواهند بود.
- می توانید نحوه نمایش بعضی یا همه قطعات و اشیا را ثابت نمائید.(Fix)

به عنوان مثال فرض کنید قطعه جدیدی ایجاد و آن را در کتابخانه ذخیره کرده اید.(این کار توسط Component graphic در سبک Component انجام می گیرد.) زمانیکه این قطعه از کتابخانه به طرح بارگذاری می شود به طور اتوماتیک از سبکComponent پیروی می کند.

12-1 گرافیک های دو بعدی 2D graphic

Isis انواع گرافیک نظیر خطوط وجعبه ها و دوایر و کمانها و متون مقیاس پذیر وسمبلهای مرکب را پشتیبانی می نماید می توانید از این ابزارها برای ترسیم طرح ها و همچنین برای ایجاد قطعات کتابخانه ای جدید استفاده نمایید (نظیر قطعات و سمبل ها و پینها و ترمینالها).

1-12-1 جا گذاری گرافیک دو بعدی

مراحل زیر برای جای گذاری هر نوع گرافیکی بیان شده است.

- **برای جا گذاری خط : (To place a line)**

1- نوار ابزار Mode Selector آیکون Line را انتخاب کنید.

2- سبک گرافیکی دلخواهتان(Graphics style) را که می خواهید خط با آن سبک ترسیم گردد را از Object Selector انتخاب نمائید.

3- برای شروع ترسیم خط چپ کلیک کنید.

4- برای بیان نقطه انتهای خط در آن نقطه چپ کلیک نمائید.

- **برای جا گذاری یک جعبه (To place a box)**

1- نوار ابزار Mode Selector آیکون box را انتخاب کنید.

2- سبک گرافیکی دلخواه را که می خواهید جعبه با آن سبک ترسیم شود را انتخاب کنید.

3- جایی را نشانه بروید که می خواهید گوشه چپ بالای جعبه آنجا باشد.

4- آن را از گوشه بالای چپ به سمت گوشه راست پائینی دراگ نمائید و بعدا کلید ماوس را رها سازید.

- **برای جا گذاری متون گرافیکی (To place a graphics text)**

1- از نوار ابزار Mode Selector آیکون text را انتخاب کنید.

2- سبک گرافیکی دلخواه را انتخاب کنید.

3- از آیکونهای Rotation , Mirror جهت تنظیم جهات متن آن طوری که می خواهید در طرح نشان داده شوند استفاده نمائید.

4- در ناحیه ای از صفحه که می خواهید قسمت پایین و چپ text آنجا باشد نشانه رفته و چپ کلیک نمائید. فرم محاوره ای Edit 2D Graphics text نشان داده خواهد شد.

5- متن دلخواه را در فرم تایپ نموده و همچنین تنظیماتی نظیر Text Size و غیره را در صورت لزوم انجام دهید.

6- جهت جاگذاری متن ENTER را فشار داده یا ok را کلیک نمائید و اگر از جاگذاری متن منصرف شدید از کلید ESC جهت cancel نمودن روندااستفاده نمائید.

- **برای جا گذاری یک سمبل (To place a symbol)**

1- از نوار ابزار Mode Selector آیکون symbol را انتخاب کنید.

2- سمبل دلخواه را جهت جایگذاری از Object Selector انتخاب کنید. اگر سمبل دلخواهتان در اینجا موجود نبود ابتدا باید آن را از کتابخانه سمبلها بر دارید. با فشار p در انتخابگر می توانید فرم symbol library را به نمایش در آورید.

3- از آیکون Rotation , Mirror در صورت لزوم می توانید استفاده کنید.

4- در ناحیه ای از پنجره ویرایش که می خواهید سمبل نمایش داده شود کلیک چپ نمائید. اگر کلید ماوس را پایین نگه دارید می توانید سمبل را دراگ هم بکنید. بعد از رها سازی کلید ماوس سمبل در جایگاهش قرار می گیرد.

1-12-2 ویرایش گرافیک های دو بعدی (Editing 2D graphic)

همه اشیا گرافیک دو بعدی را می شود با روش معمولی ویرایش { یعنی انتخاب آنها با راست کلیک و سپس چپ کلیک روی آنها (بدون حرکت دادن ماوس)} نمود.

همه اشیا گراف دو بعدی به جز 2D graphic text یک جعبه محاوره ای تحت عنوان (Graphics style Editing) را نمایش می دهند واین اجازه را می دهند که خصوصیات محلی مانند مقادیر و دیگر تنظیمات را انجام دهید.

1-13 شروع یک طرح جدید:

دستور New Designتمام صفحه را پاک خواهد کرد ویک صفحه A4 خالی را نشان خواهد داد. (برای استفاده از Template های نرم افزار فرمان New را از منوی فایل احضار کنید.)
فایل طراحی نام UNTITLED.DSN را به خود خواهد گرفت واین نام می تواند با دستور Save Design عوض گردد. شاید بخواهید طرح جدیدی را با نام جدید ایجاد کنید. در این صورت می توانید از دستور Load Design استفاده نمائید و در انتخابگر فایل نام فایل جدید را وارد نمائید.

1-14 بارگذاری طرح Loading a Design

یک طرح می تواند به یکی از سه روش زیر بارگذاری گردد.

- از اعلان Dos به صورت ISIS < my – design >
- با استفاده از آیکون دستور Load Design زمانیکه ISIS اجرا شده باشد.
- به وسیله دابل کلیک نمودن فایل در کاوشگر ویندوز (Windows Explorer)

1-14-1 ذخیره طرح

موقع خروج از ISIS طرحتان را می توانید با استفاده از دستور Save Design ذخیره نمائید.
نکته: دستور Save as این اجازه را می دهد که طرحتان را در یک فایل با نام دیگر ذخیره سازید.

1-15 دستورات Import / Export

دستور Export از منوی فایل یک فایل مقطعی (Section File) از تمامی اشیا انتخاب شده جاری را ایجاد می نماید. این فایل با استفاده از دستور Import می تواند بداخل دیگر صفحات فراخوانی شود.

1-16 خروج از ISIS

زمانیکه بخواهید کار با ISIS را خاتمه دهید باید از دستور Exit از منوی فایل استفاده نمایید یا از کلید میانبر Q استفاده کنید. توجه کنید که اگر در طرح اصلاحی انجام داده باشید آن را حتما ذخیره نمایید.
در این کتاب ما به تعریف محیط نرم افزار Proteuse و چگونگی کار با این نرم افزار می پردازیم.

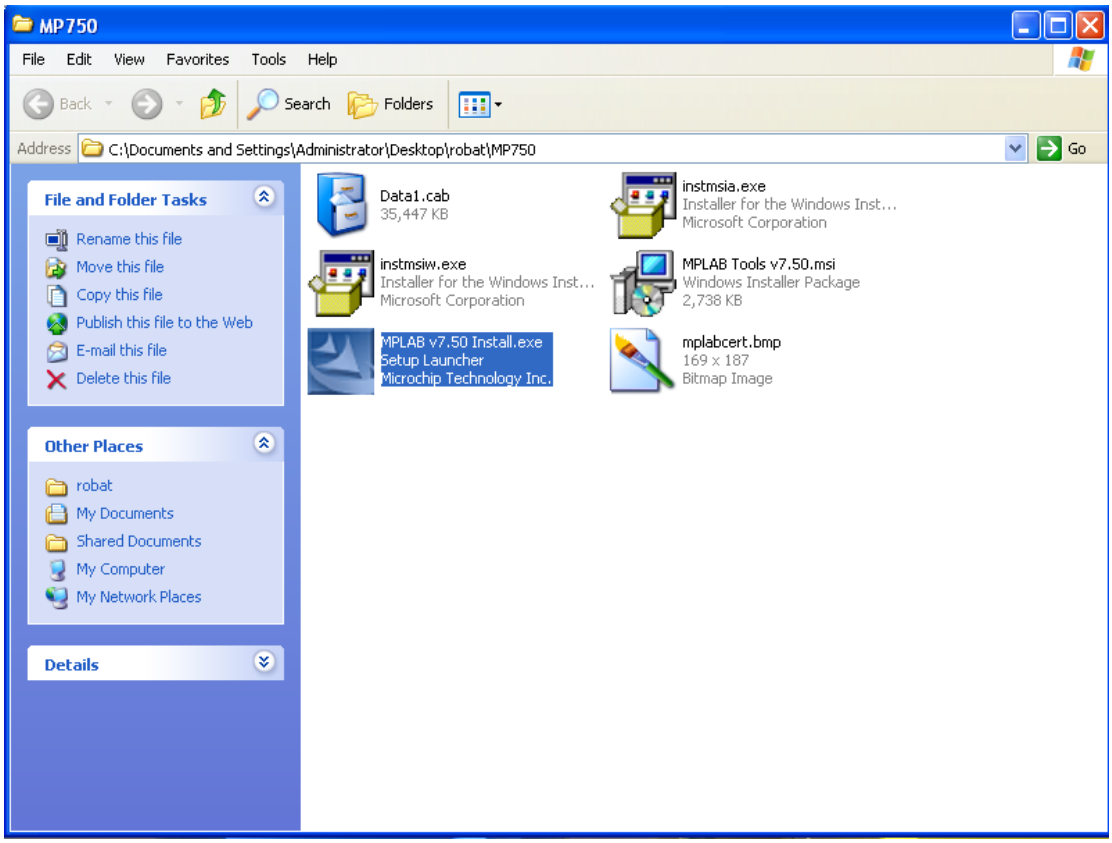
فصل دوم

2. نصب نرم افزار

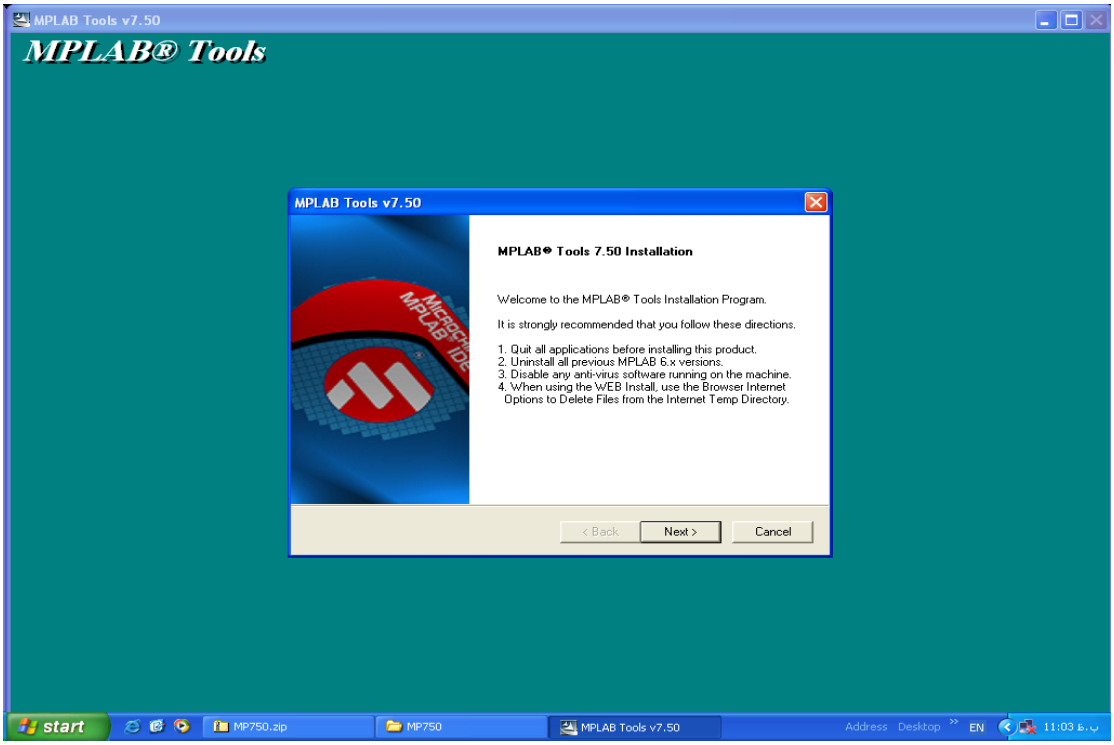
2-1 نصب نرم افزار MPLAB

ابتدا از دیسکت یا CD محتوی نرم افزار، فایل اجرایی آن را (مثلاً در اینجا MPLAB v7.50 Install.exe) را انتخاب می کنیم.

پنجره شکل زیر نمایش می یابد:



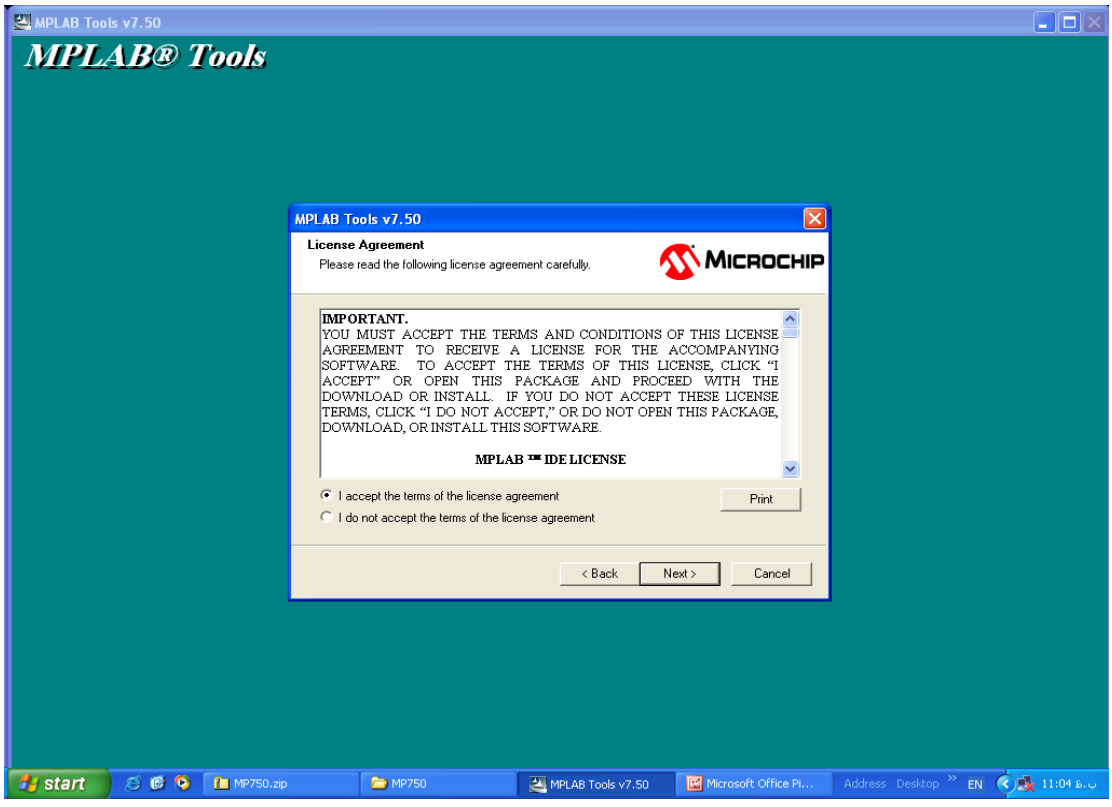
شکل 2-1– پنجره نصب نرم افزار



شکل 2-2– ادامه نصب

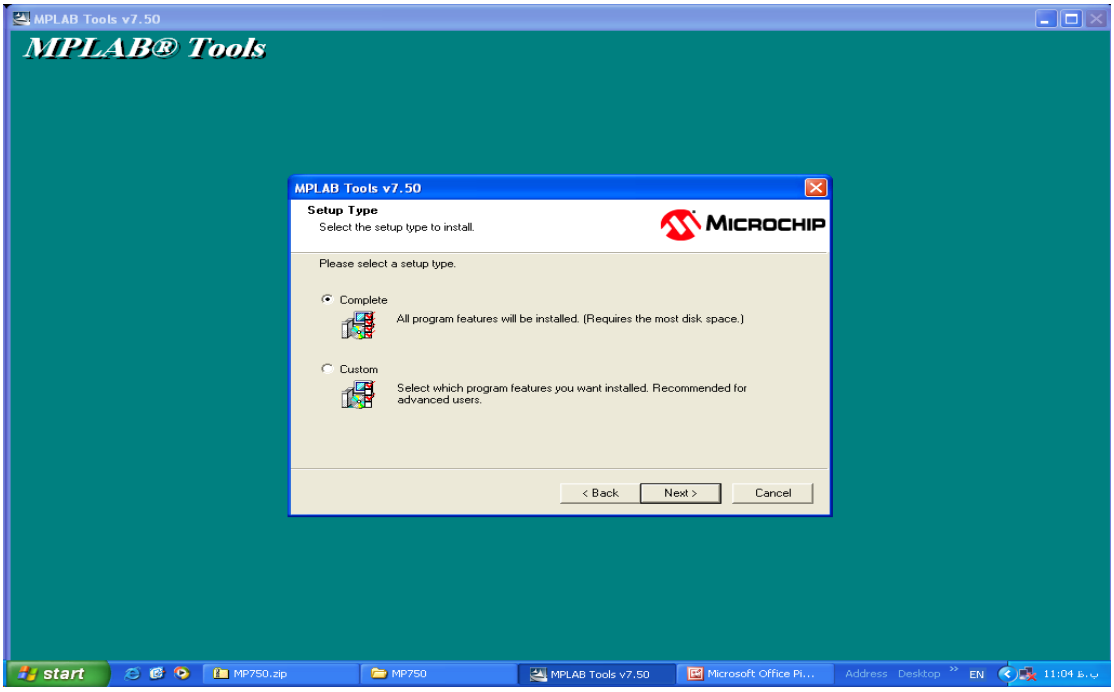
با کلیک روی دکمه Next به صفحه بعد می رویم:

گزینه … I accept the را انتخاب کرده، روی Next کلیک کنید.



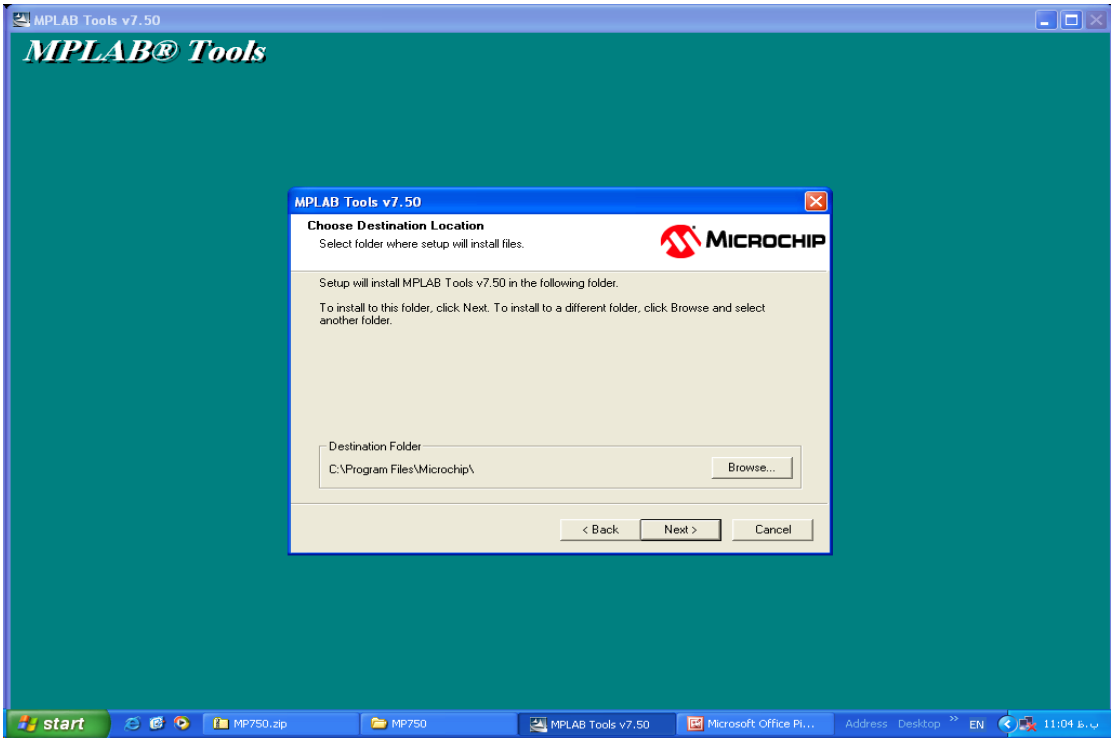
شکل 2-3– قوانین کپی رایت نرم افزار

بعد از انتخاب یکی از انواع نصب روی Next کلیک کنید.

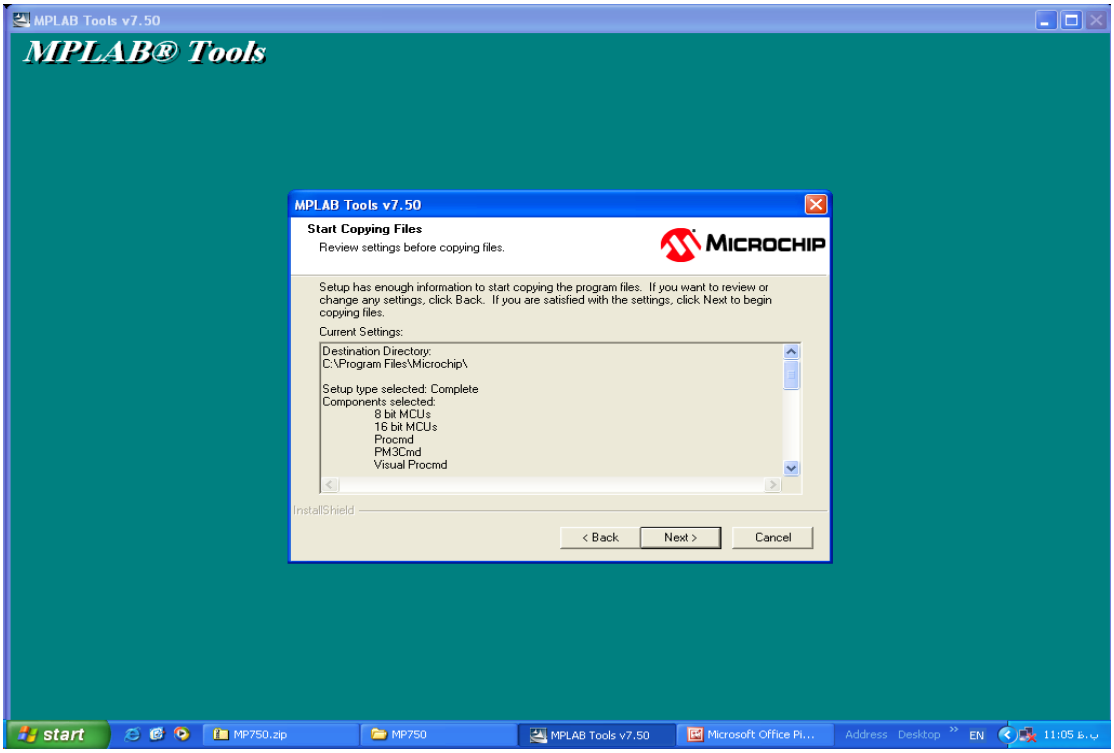


شکل 2-4– انتخاب نحوه نصب

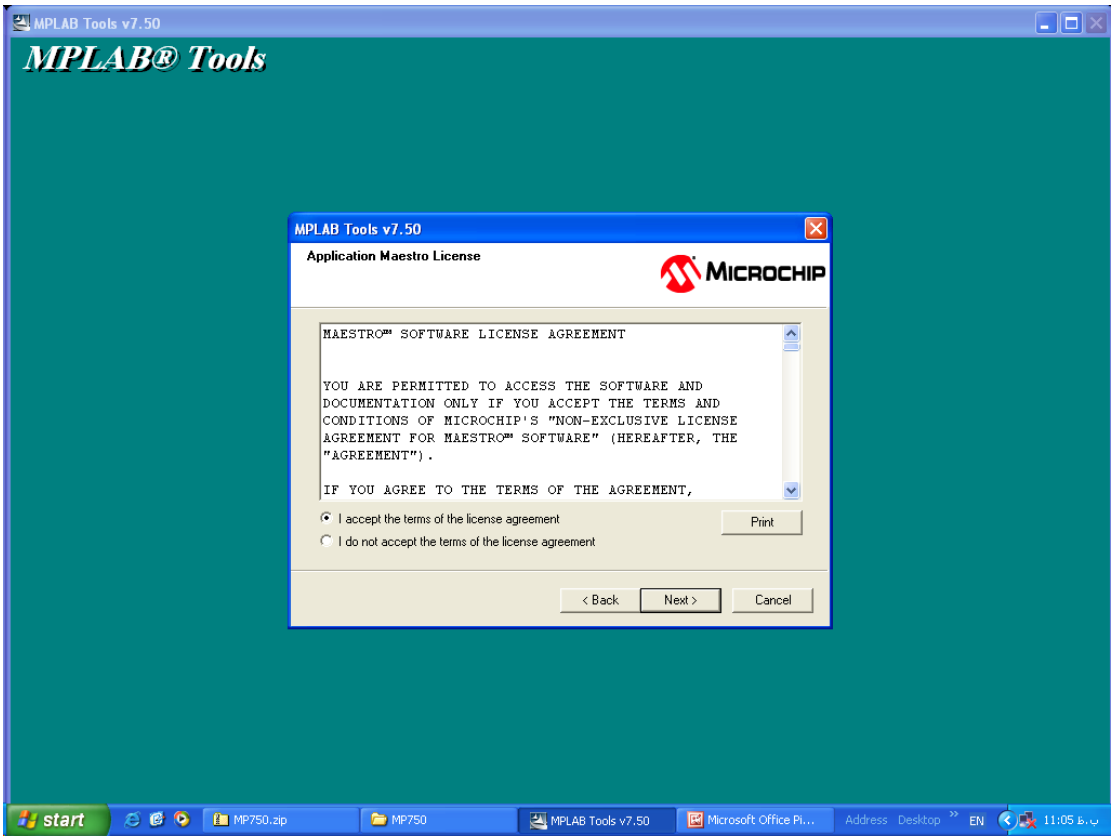
در این پنجره می توانید مسیر نصب برنامه را انتخاب کنید، بعد از انتخاب روی دکمه Next کلیک کنید.



شکل 2-5– انتخاب شاخه مورد نظر برای نصب

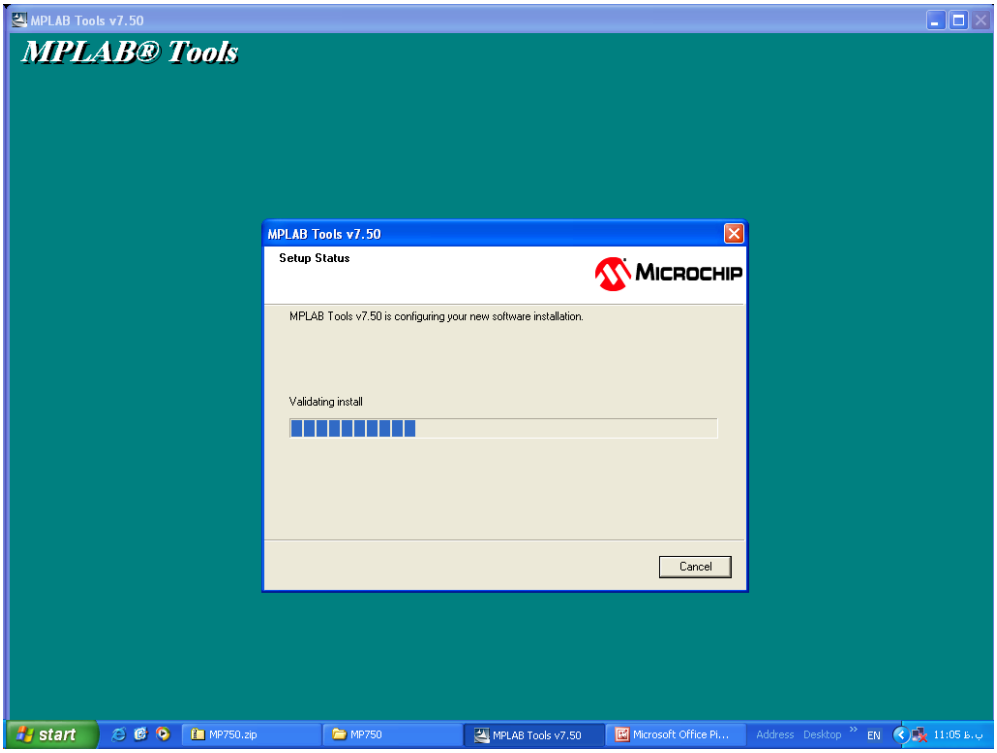


شکل 2-6– نمایش اطلاعات نصب نرم افزار



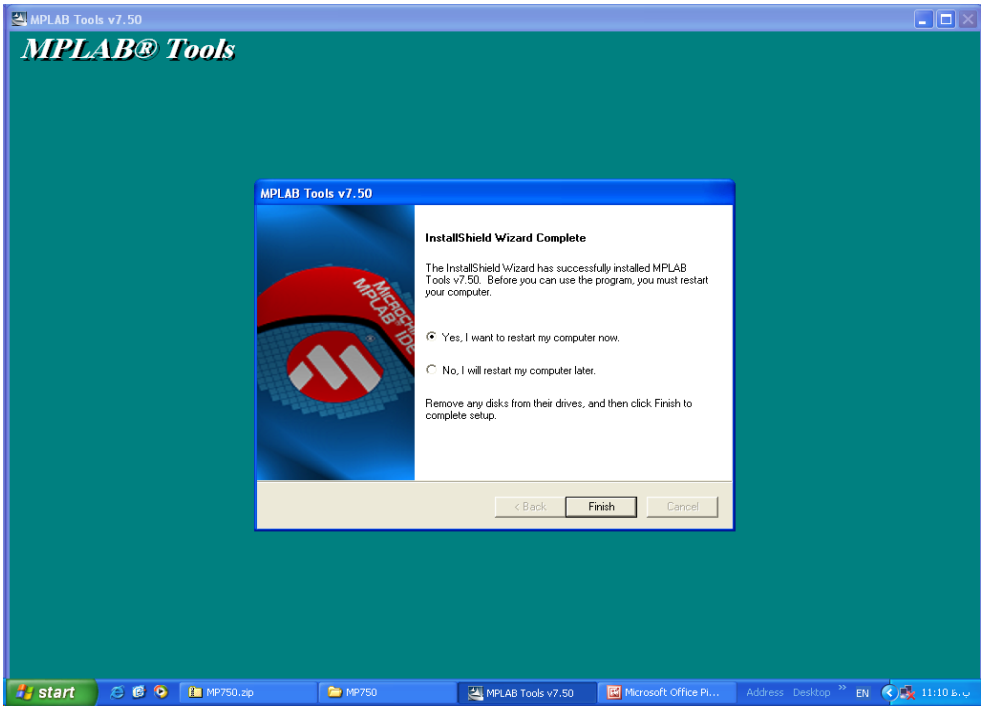
شکل 2-7 – قوانین کپی رایت نرم افزار

کپی کردن فایلهای مربوط به برنامه آغاز می شود، برای رفتن به مرحله بعد روی Next کلیک کنید.



شکل 2-8- نمایش روند کپی کردن فایل ها بر روی سیستم

با ملاحظه این پنجره باید تا اتمام نصب منتظر بمانید.



شکل 2-9– پنجره نهایی نصب

در انتها باید پنجره ای مشابه این پنجره ببینید، که اعلام می کند که نصب برنامه کامل شده است، با انتخاب گزینه Yes, I want… اعلام می کنید که برای تکمیل نصب سیستم Restart شود. روی دکمه Finish کلیک کنید.

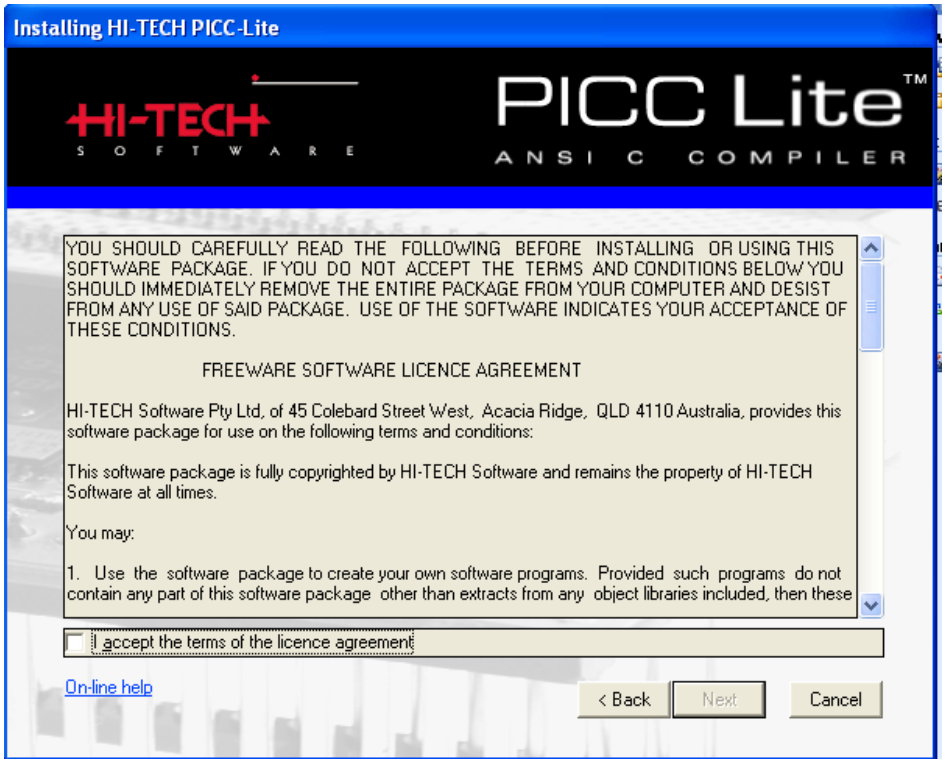
2-2 نصب نرم افزار Picc-Lite:

پس از نصب برنامه MPLAB که یک محیط IDE را برای نوشتن برنامه در اختیارتان می گذارد، بایستی برنامه Picc-Lite را نیز نصب کنید، این برنامه یک اسمبلر، لینکر و کامپایلر است که می تواند به راحتی با برنامه MPLAB کار کند.

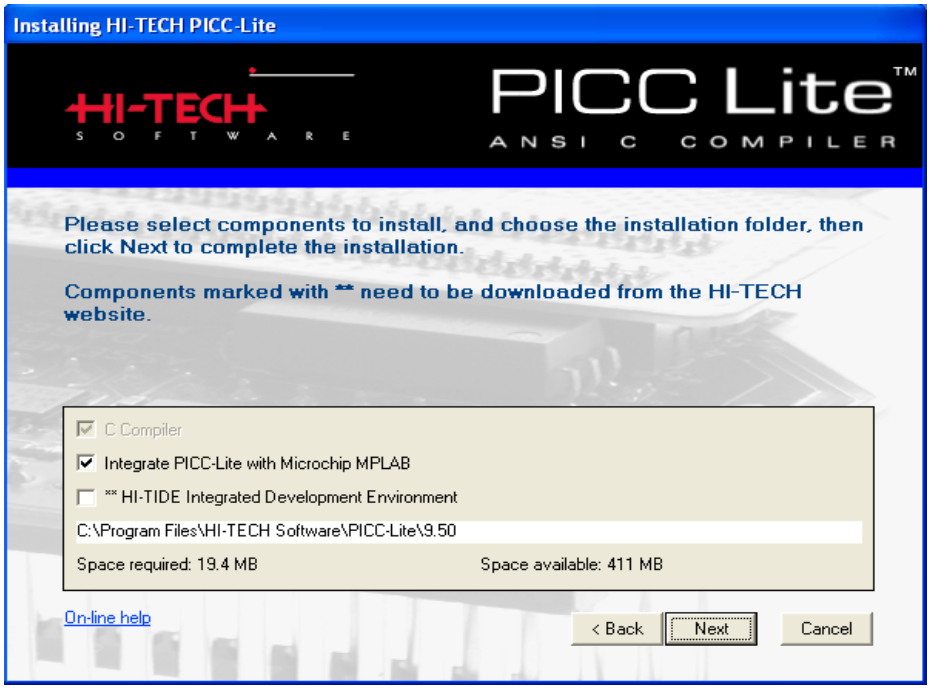
برای نصب این برنامه ابتدا بر روی فایل نصب آن دابل کلیک می کنیم، پنجره ای مطابق شکل زیر نمایش داده می شود:

شکل 2-10 - پنجره نصب PICCLITE

برای ادامه نصب روی next کلیک کنید.

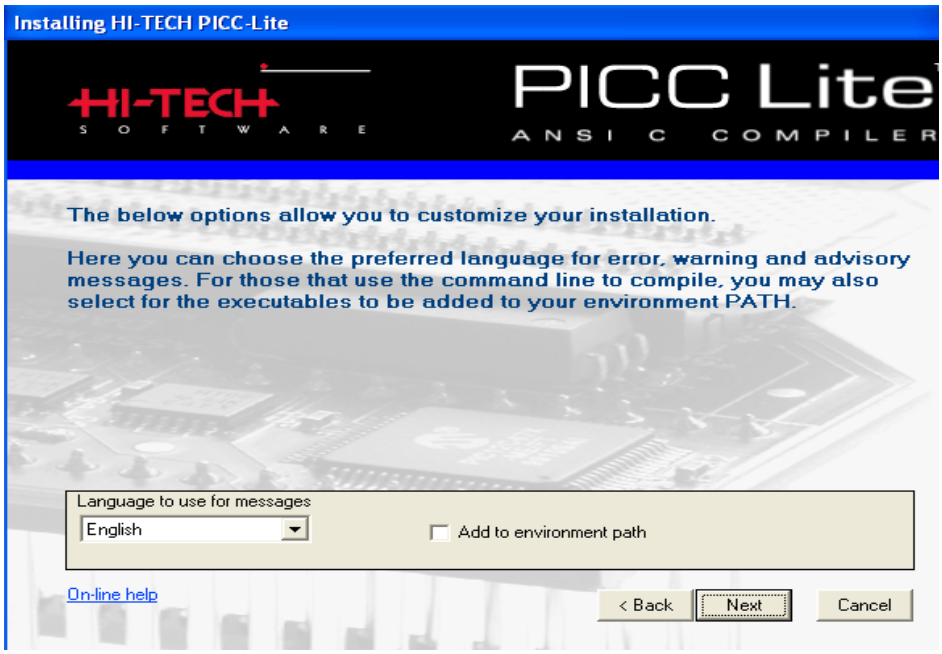


شکل 2-11 – قوانین کپی رایت



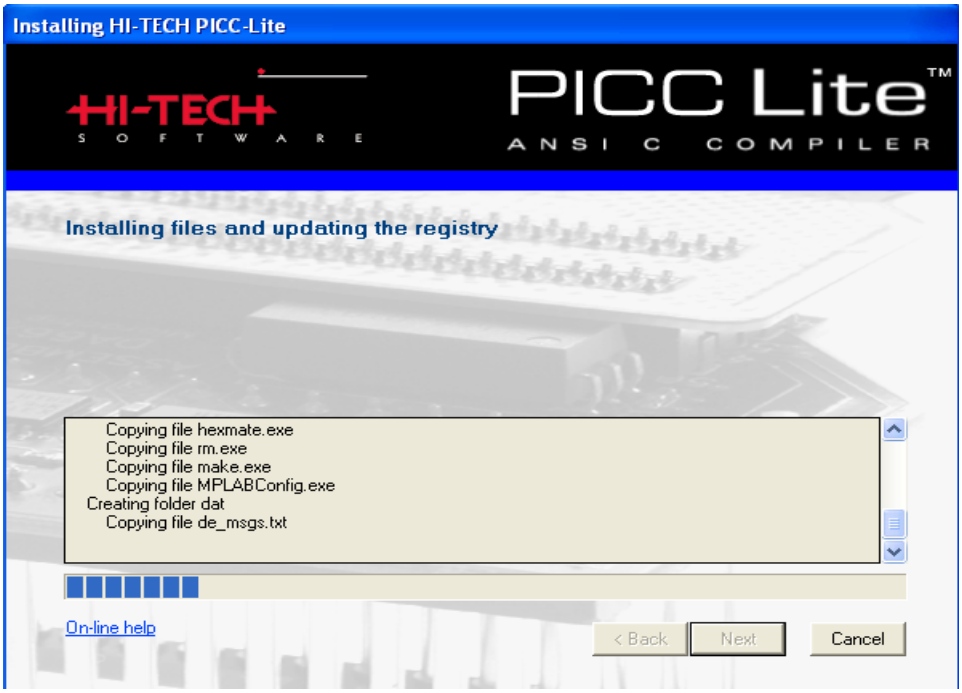
شکل 2-12 – هماهنگ کردن PICC Lite با میکروچیپ های MPLAB

زبان مورد استفاده در این نرم افزار را از این پنجره می توانید انتخاب کنید، به گام بعدی بروید.



شکل 2-13 – انتخاب زبان نصب

برای رفتن به مرحله بعد next را انتخاب کنید.

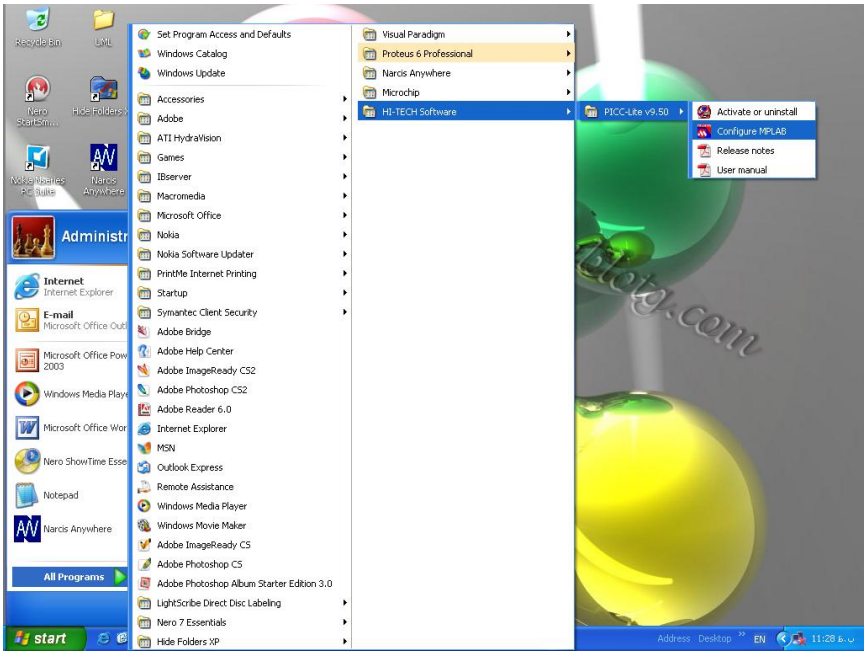


شکل 2-14 – نمایش روند کپی کردن

تا اتمام مراحل کپی منتظر بمانید.

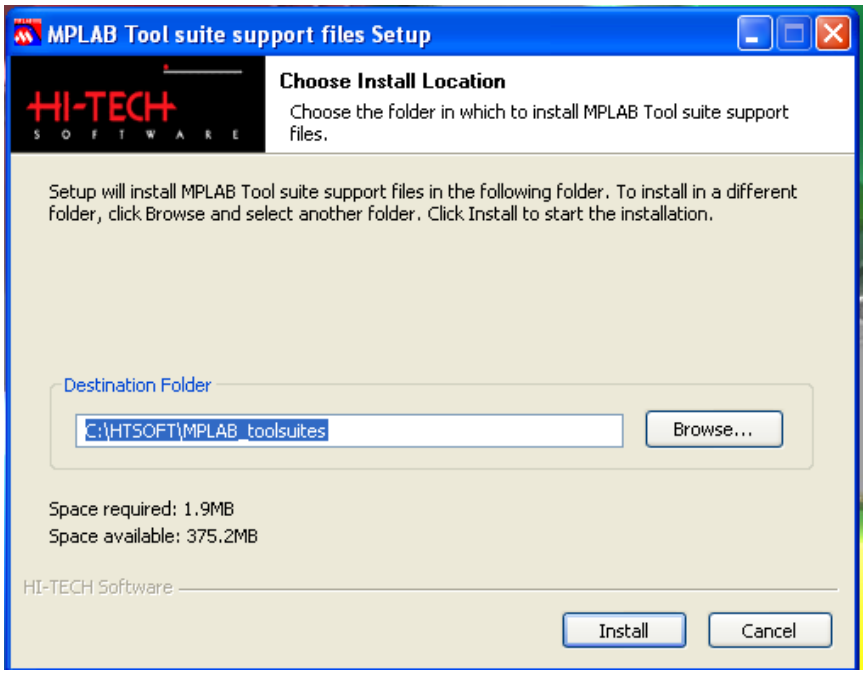
شکل 2-15 - پایان مراحل نصب

حال نصب برنامه به انتها رسیده، اما برای اینکه بتوانید برنامه ها را از داخل محیط MPLAB کامپایل یا اسمبل کنید، باید تنظیمات دیگری هم انجام دهید. برای این کار مطابق شکل زیر عمل کنید.



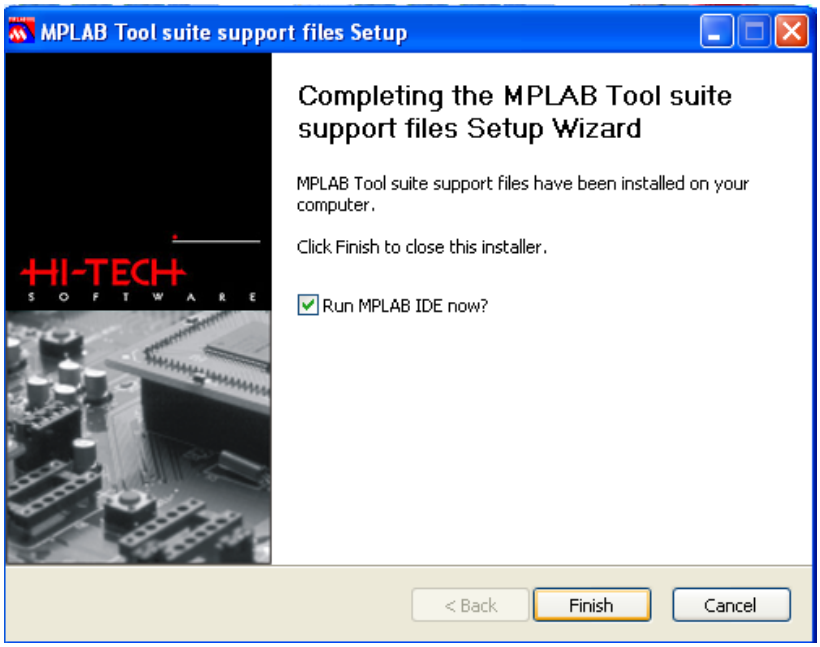
شکل 2-16- محل قرار گرفتن نرم افزار در منوی start

بعد از این انتخاب ویزاردهایی برای تنظیمات Picc-Lite با MPLAB مشاهده می کنید.



شکل 2-17 – تعیین شاخه نصب نرم افزار

در این پنجره در کادر Destination Folder باید مسیر فایلهای تنظیمات را مشخص کنید، برای ادامه روی Install کلیک کنید.



شکل 2-18– صحفه نهایی نصب

با مشاهده این صفحه مراحل نصب به اتمام رسیده، روی Finish کلیک کنید.

2-3 نرم افزار MPLAB:

نرم افزار MPLAB یک محیط IDE یا برنامه نویسی در اختیار کاربران قرار می دهد، که می توانند فایلهای خود را در آن کامپایل کنند. خروجی این نرم افزار فایلهایی با پسوند hex. می باشد که این فایلها می توانند برای program کردن یک pic استفاده شوند.

2-3-1معرفی نرم افزار MPLAB:

محیط IDE نرم افزار MPLAB از چند بخش اساسی تشکیل شده است:

1) نوار منو که بعداً توضیح داده خواهد شد.
2) نوار ابزار، که بصورت پیش فرض نوار ابزار استاندارد، project manager 2 و checksum فعال هستند. (این نوارابزارها بعداً توضیح داده خواهد شد). نوار ابزارها را می توان از منوی view و با انتخاب آنها از زیرمنوی Tools فعال یا غیرفعال کرد.
3) فضای خالی که محل قرارگیری پنجره های مربوط به برنامه است.
4) نوار وضعیت، این نوار پایین ترین نوار در محیط است و همیشه فعال است.

حال به شرح نوار منو که مهمترین نوار در این محیط است می پردازیم:

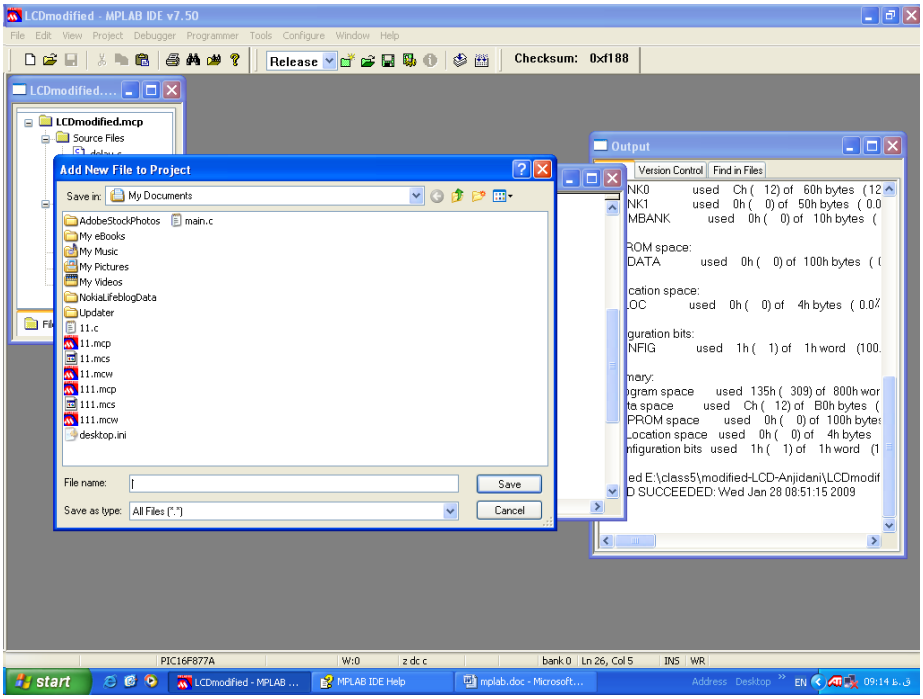
1. File

- New

یک پنجره جدید قابل ویرایش با عنوان "Untitled" برای فایل جدید باز می کند. این پنجره را می توانید ذخیره کنید، زمانی هم که قصد بستن پنجره را داشته باشید میتوانید آنرا ذخیره کرده و نامگذاری کنید.

- Add New File to Project

پنجره ای مطابق شکل زیر نمایش داده می شود، در این پنجره یک نام برای فایل جدید درخواست می شود. اگر بعد از واردکردن نام روی دکمه Save کلیک کنید، فایل جدید به پروژه اضافه شده و یک پنجره برای فایل جدید باز می شود.



شکل 2-19– نمایی از نرم افزار MPLAB

- Open

یک فایل source که قبلاً ذخیره شده را با این گزینه می توانید باز کنید. همچنین امکان انتخاب چند فایل بصورت همزمان در پنجره open را با استفاده از کلیدهای ctrl یا shift را دارید.

- Close

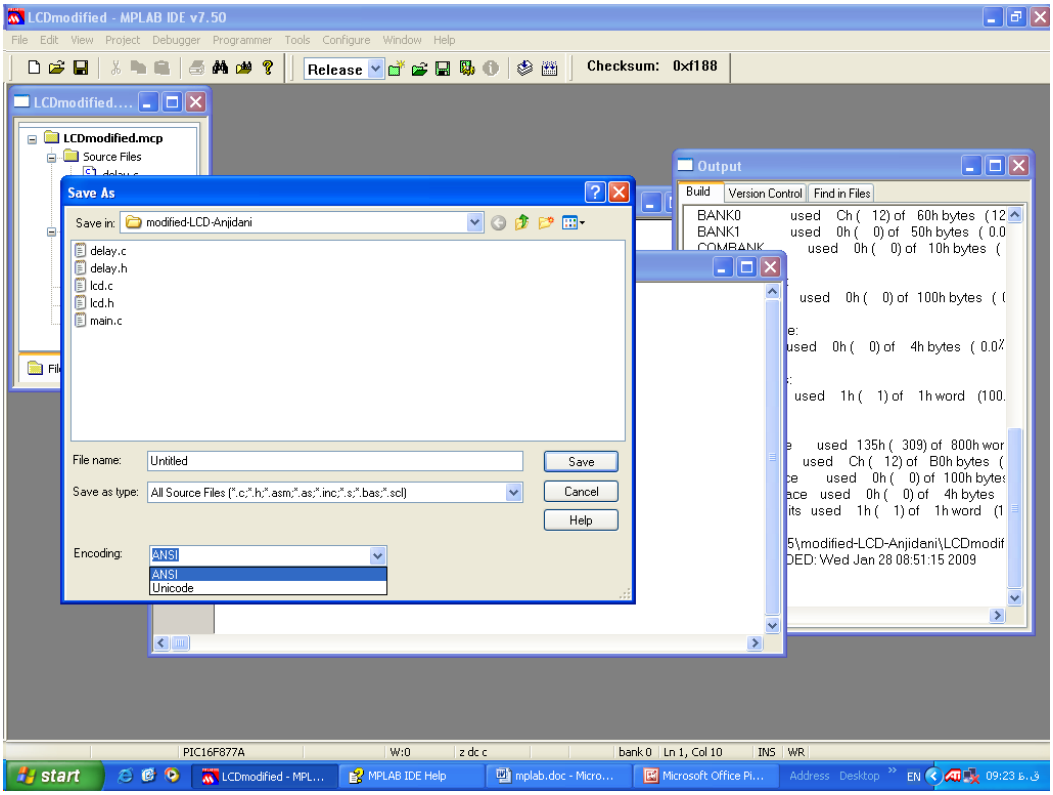
پنجره ویرایشگر فعال خود را با این گزینه می توانید ببندید. اگر از آخرین زمانی که فایل را ذخیره کرده اید، تغییرات جدیدی به آن اعمال کرده باشید، قبل از بستن با پنجره ای برای ذخیره فایل روبرو می شوید.

- Save

با این گزینه می توانید فایل را در محل اصلی خود ذخیره کنید.

- Save As

انتخاب این گزینه باعث می شود پنجره Save As باز شود و بتوانید فایل را با نامی دیگر یـا در یـک مسـیر جدید ذخیره کنید. همانطور که در شکل زیر می بینید، در این پنجره یک لیسـت جدیـد بـا عنـوان "Encoding" وجود دارد. که می توانید encoding مربوط به فایل را از این لیست انتخاب کنید.



شکل 2-20– نمایی از نرم افزار MPLAB

- Save All

تمام پنجره های ویرایشگری که باز هستند را در محل اصلی شان روی دیسک ذخیره می کند.

- Open Workspace

پنجره Open Workspace را برای بازکردن یک Workspace نشان می دهـد، البتـه قبـل از بـازکردن Workspace جدید تمام Workspace های قدیمی بسته می شوند.

- Save Workspace

انتخاب این گزینه Workspace جاری را ذخیره می کند. البته اگـر Configure>Setting را انتخـاب و بعد به تب Workspace بروید، می توانید تنظیمات Workspace را انجام دهید. بعد از بستن این پنجره بصورت اتوماتیک Workspace ذخیره خواهد شد.

- Save Workspace As

پنجره Save As باز می شود و به شما اجازه می دهد تا Workspace خود را با نام جدیـد یـا در مسـیر جدید ذخیره کنید.

- Close Workspace

Workspace خـود را بـا انتخـاب ایـن گزینـه مـی بندیـد، و بـه تنظیمـات ابتـدایی پـیش فـرض بـرای Workspace بازمی گردید.

- Import

با انتخاب این گزینه میتوانید یک فایل اشکالزدا یا یک فایل hex را به پروژه خود اضافه کنید.

- Export

با انتخاب این گزینه می توانید یک خروجی فایل hex از پروژه خود در محیط MPLAB تهیه کنید.

- Print

محتویات پنجره ویرایشگر فعالتان را با این گزینه میتوانید چاپ کنید. پنجره چاپ با این انتخاب باز می شود و به شما اجازه داده می شود که تنظیمات چاپ و پرینتر خود را انجام دهید.

- Recent Files

لیستی از فایلهایی که اخیراً در محیط MPLAB باز شده را می توانید در این لیست ببینید. برای تعیین تعداد فایلهایی که می خواهید در این قسمت نمایش یابد از منوی Configure گزینه Setting را انتخاب کنید، در تب Workspace می توانید این تعداد را تعیین کنید.

- Recent Workspaces

لیستی از Workspace های اخیر را می توانید با انتخاب این گزینه ببینید. مثل قسمت قبلی از منویConfigure گزینه Setting را انتخاب کنید. در تب Workspace می توانید این تعداد را تعیین کنید.

- Exit

با این گزینه برنامه MPLAB بسته می شود.

2. Edit

- Undo

برای بازگرداندن آخرین عمل انجام شده استفاده می شود.

- Redo

برای لغو یک عمل undo استفاده می شود.

- Cut

انتقال موارد انتخاب شده به داخل حافظه

- Copy

تهیه کپی از موارد انتخاب شده در حافظه

- Paste

انتقال از حافظه به مکان مورد نظر

- Delete

پاک کردن موارد انتخاب شده

- Select all

انتخاب تمام موارد

- Find

پیدا کردن یک متن در داخل فایل جاری

- Find Next

پیدا کردن نتیجه بعدی در عمل Find

- Find in Files

پیدا کردن یک متن در داخل فایل (در هر پروژه یا فایل موجود)

- Replace

جایگزینی

- Goto

رفتن به یک خط خاص، به یک تابع خاص، یا یک prototype

- Advanced

حاوی گزینه های ویرایشی، تبدیل متن انتخابی به یک توضیح و ...

- Bookmarks

حاوی گزینه هایی برای ایجاد یک Bookmark، رفتن به Bookmark قبلی و بعدی و پاک کردن Bookmark ها.

- Properties

حاوی تنظیماتی برای پنجره های Editor

3. View

• Project Window

Project Window دربرگیرنده خلاصه ای از اطلاعات در مورد پروژه است. View>Project را بطور متناوب برای ظاهرشدن و مخفی شدن این پنجره کلیک کنید.

تب Files ، فایلها را در سه ساختار نشان می دهد. و تب Symbols کدهای نمونه C و غیره، توابع، متغیرها و .. را نشان می دهد.

• Output Window

با انتخاب View>Output پنجره Output نمایش می یابد. در هر تب این پنجره اطلاعاتی در مورد خروجی برنامه وجود دارد:

- تب Build: پیغامهایی که در هنگام Build کردن پروژه نمایش می یابند، در این پنجره وجود دارد. زبان پیغام های نمایش داده شده در این تب براساس تنظیماتی است که در مسیر زیر ایجاد کرده اید:

  Project>Select Language Toolsuite

  و تنظیمات Build را می توانید از مسیر زیر انجام دهید:

  Project>Build Options>Project

- تب Version Control: اگر سیستم کنترل Version ای در پروژه استفاده شود، در این تب اطلاعات مربوط به کنترل Version نمایش داده می شود.
- تب Find in Files: نتیجه عملیات Project>Find in Project File در این تب نشان داده می شود.
- براساس عملیات انتخابی ممکن است تب های دیگری در این پنجره قابل دسترس باشند.

• MPLAB IDE Toolbars

محیط MPLAB با توجه به خصوصیات و ابزارهایی که شما استفاده می کنید، نوارابزارهای متفاوتی در اختیارتان قرار می دهد. در واقع آیکن های نوارابزار میانبرهایی به روالهای مربوط به آن کار در اختیارتان قرار می دهد.

خصوصیات نوارابزارها:

- با کلیک کردن و درگ کردن روی نوارابزار می توانید آنرا شناور کرده و به هرجا می خواهید بکشانید.
- با کلیک کردن و درگ کردن نوارابزار به بالای میزکار محیط MPLAB میتوانید آنرا به بالای محیط سنجاق کنید.
- با کلیک و درگ کردن نوارابزار می توانید آنرا از محیط MPLAB خارج کنید. (خاموش کردن نوارابزار)
- برای دیدن نام یک آیکن نشانگر ماوس را بر روی یک آیکن نگه دارید.
- بر روی یک نوارابزار راست کلیک کنید تا محتویات آن یا حالت نمایش/مخفی کردن نوارابزار را فعال کنید.

• Call Stack Window

برای ابزارهای کنترلر سیگنال دیجیتال dsPIC (یا DSC)، یک پنجره Stack برای فراخوانی و رفتن به بخش خاصی از کدهای اجرایی قابل دسترس است.

نکته: اگر اجرای برنامه در حین اعلان توابع متوقف شود، توابع فراخوانی شده در این پنجره نمایش داده نمی شود.

توصیه می شود که بهینه سازی کد در زمان استفاده از call stack انجام نشود.

• Disassembly Listing Window

با انتخاب این گزینه کدهای اسمبل شده مربوط به فایل در پنجره ای جدید نمـایش داده مـی شـود. ممکـن است در کدها Breakpoint هایی از قبل تنظیم شده باشد.

اطلاعات اجرایی کد هم ممکن است به فرم نشانه هایی در گوشه پنجره نمایش داده شوند.

• EEPROM Window

در پنجره EEPROM که با انتخاب این گزینه باز می شود، بـرای هـر میکروکنترلـر کـه داده هـای حافظـه EEPROM دارد، داده هـای EEPROM نمـایش داده مـی شـود.اطلاعات هگزادسـیمال داده/upcode ابزارهـای انتخاب شده نمایش داده می شوند. وقتی مقدار ثبات EEPROM تغییر می کند یا پـردازش آن متوقـف شـود، داده نمایش داده شده در این پنجره با آخرین مقدار خود بروزرسانی می شود.

• File Register window

این پنجره تمام ثبات های یک وسیله انتخابی را نمایش می دهد. وقتی مقدار ثبـات فایـل تغییـر کنـد، و یـا پردازشی تشخیص داده شود، داده های این جدول با آخرین مقدار خود بروزرسانی می شود.

نکته: برای بالابردن سرعت اشکالزدایی با ابزارهای یک سخت افزار خاص، این پنجره را ببندیـد. بجـای آن از پنجره های SFR یا Watch استفاده کنید.

پنجره Hardware Stack محتویات Stack سخت افزار را نشـان مـی دهـد. تعـداد سـطوح قابـل دسـترس بستگی به ابزارهای انتخابی دارد.

• LCD Pixel Window

وقتی عملیات LCD فعال شده باشد (SFRs ها در قسـمت پـائین پنجـره) در ایـن پنجـره خروجـی LCD نمایش داده می شود. تمامی پیکسل های LCD (بصورت مربعی) وقتی عملیات LCD غیرفعال است، بصورت پـیش زمینه خاکستری نمایش داده می شوند. زمانی که عملیات LCD فعال شد، پیکسلها به دو صورت سفید (خاموش یـا 0) و خاکستری تیره (روشن یا 1) بنظر می رسند.

• Locals Window

این گزینه باعث می شود تا شما بتوانید متغیرهای اتومات خود را کـه دارای حـوزه محلـی (local) هسـتند ببینید. این گزینه برای پروژه هایی که با زبان های سطح بالا نوشته می شود (مانند C,BASIC و غیره) اعمـال مـی شود.

• Program Memory Window

این پنجره موقعیت ها را در محدوده حافظه برنامه برای پردازش های انتخابی نشان مـی دهـد. اگـر حافظـه خارجی برنامه بوسیله ابزارهای انتخابی ساپورت و فعال شده باشد، آنها در این پنجره نشان داده می شوند.

• Special Function Registers Window

این پنجره محتویات ثبات های عملیاتی خاص(SFRs) را برای پردازش های انتخابی نشان می دهد.

نکته: اگر یک ثبات حافظه داده بصورت فیزیکی روی یک ابزار بکار گرفته نشود، ممکن است در لیست SFR دیده نشود، مانند شبیه سازها، ممکن است به شما اجازه دهد ثبات هایی را که در ابزارهـای حقیقـی وجـود ندارنـد، ببنید مثل prescaler ها.

شکل اطلاعاتی که در این پنجره نمایش داده می شود، برای نمایش SFRs هـا مفیـدتر از پنجـره normal file register است، زیرا هر SFR دربرگیرنده تعداد بیشتری فرمت نمایش است. زمانی کـه یـک شکسـت رخ مـی دهد، محتویات این پنجره با آخرین مقدار ثبات ها بروزرسانی می شود.

نکته: اگر "Freeze Peripherals On Halt" انتخاب شده باشد، بیت های پورت ورودی/خروجی در SFR یا پنجره Watch با یک مرحله بروزرسانی نمی شوند. پین تغییر خواهد کرد، اما درخواست خواندن برای برگردانـدن متغیر جدید بوسیله freeze بلوکه می شود و تا زمان شروع مرحله بعد یا اجرای خط فرمان بروزرسانی نمی شود.

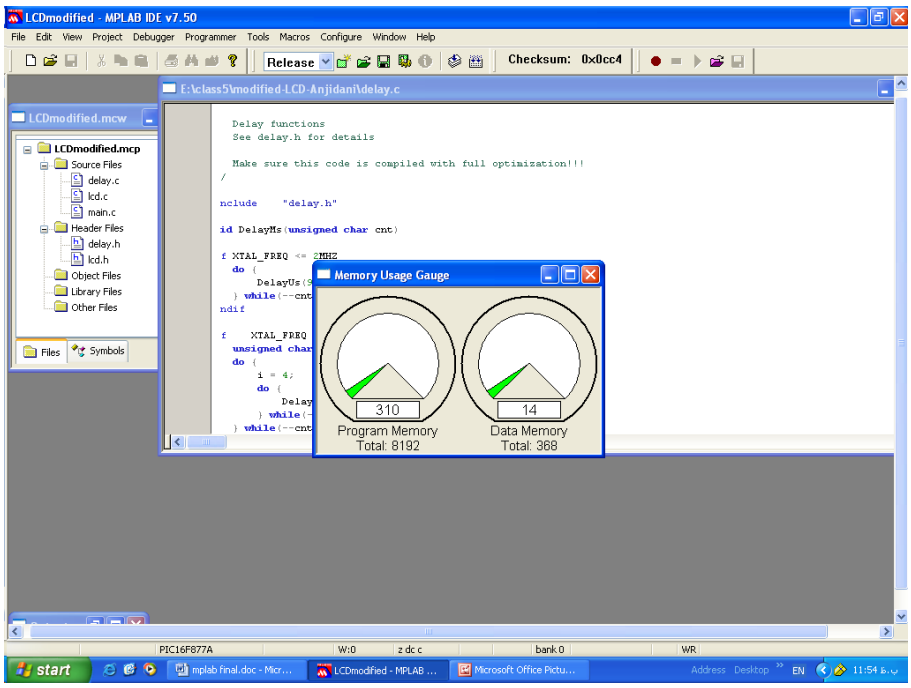
• Watch Window

این پنجره به شما اجازه می دهد نمادهای برنامه (symbol) را در حین اجرای برنامه ببینید. شـما بیشـتر از 4 تب متفاوت (بصورت پیش فرض 4 تب متفاوت) در این پنجره می توانید تنظیم کنید.

• Memory Usage Gauge

این پنجره میزان حافظه ای که در حال حاضر در پروژه فعال در حال استفاده بـرای برنامـه و داده هاسـت را نمایش می دهد.

نکته: شما می توانید از اتصال دهنده ها برای تولید کد برای مقداردهی این مقادیر استفاده کنید. فایـل هـای a*.cof و غیره یا فایلهای اشکالزدای elf باید تولید شوند.



شکل 2-21– نمایشگر حافظه میکروکنترلر

4. Project

- New

با این گزینه می توانید یک پروژه جدید در Workspace ایجاد کنید. با انتخـاب ایـن گزینـه یـک پنجـره برای ایجاد پروژه باز شده و نام پروژه جدید و مسیر آن را از شما می خواهد.

- Open

با انتخاب این گزینه میتوانید یک پروژه موجود را بازکرده و بعنوان یک پروژه فعال روی آن کار کنید.

- Close

این دستور پروژه جاری در Workspace را می بندد. اگر آخرین تغییرات پروژه ذخیره نشده باشد، با گزینه هایی برای ذخیره آن در هنگام انتخاب این فرمان روبرو می شوید.

توجه داشته باشید: بستن یک پروژه به معنای بستن Workspace نیست. برای بستن یک Workspace به ترتیب زیر اقدام کنید:

Edit>Workspace>Close

- Set Active Project

با انتخاب این گزینه پروژه انتخابی تان بعنوان پروژه فعال در Workspace شناخته می شود.

- Quickbuild (filename)

با استفاده از این گزینه Build کردن یک فایل اسمبلی به تنهایی با استفاده از اسمبلر MPASM و بدون نیاز به ساختن یک پروژه (هیچ اتصال دهنده ای) صورت می گیرد. این گزینه در ارتباط با پروژه فعال قابل اعمال است. فایل اسمبلی باید در پنجره فایل باز شده و پنجره باید فعال باشد.

برای انتخاب حالت Quickbuild ، از لیست None را انتخاب کنید.

- Clean

تمام فایلهای واسط در پروژه فعال را پاک می کند، مانند اشیاء، فایلهای hex و فایلهای اشکالزدا. این فایلها مجدداً زمانی که یک پروژه build شود ایجاد خواهد شد.

- Build All

پروژه را با اسمبل کردن/کامپایل کردن تمام فایلها Build می کند. قبل از اینکه فایلی باز نباشد، این گزینه غیرفعال است و با بازکردن یک فایل این گزینه فعال می شود.

- Make

فقط فایلی را که از زمان آخرین اجرای Build، تغییر کرده است، اسمبل/کامپایل می کند. برای فعال شدن این گزینه هم ابتدا باید فایل باز شده باشد.

- Build Options

مشاهده و انجام تنظیمات مربوط به پروژه فعال و فایلهای مجازی (که از پنجره تنظیمات Build استفاده می کنند) با استفاده از این گزینه ممکن است.

- Save Project

پروژه فعال را ذخیره می کند.

- Save Project As

پنجره Save As را برای پروژه فعال باز می کند.

- Add Files to Project

با استفاده از این گزینه می توانید فایلهایی را به پروژه فعال خود اضافه کنید.

- Add New File to Project

بعد از انتخاب این گزینه در یک پنجره محاوره یک نام برای فایل جدید درخواست می شود. بعد از ذخیره کردن، فایل به پروژه اضافه شده و یک پنجره جدید برای فایل جدید باز می شود.

- Remove Files from Project

فایلها را با انتخاب این گزینه می توانید از پروژه فعال خود حذف کنید. فایل از مسیر اصلی خود به این ترتیب حذف نخواهد شد. فقط از پروژه حذف می شود.

- Set Language Tool Locations

با استفاده از این گزینه کادر محاوره ای باز می شود کـه در آن لیسـتی از Toolsuite هـای ثبـت شـده در سیستم با موقعیت آنها (یعنی مسیرشان) وجود دارد. بسته به انتخاب Toolsuite گزینـه مناسـب در ایـن قسـمت انتخاب شود.

- Select Version Control System

با این گزینه می توانید پروژه تان را برای استفاده از یک سیستم کنترل version پیکربندی کنید.

5. Debugger

- Select Tool

یک ابزار برای اشکال زدایی را با این ابزار انتخاب کنید. بصورت پیش فرض گزینه None انتخاب شده است. لیست اشکالزداهایی که مشاهده می شود، بستگی به این دارد که شما کدام ها را روی سیسـتم نصـب کـرده باشـید. ترتیب موارد در این لیست بستگی به نصب دارد.

وقتی یکی از انواع این ابزارها را انتخاب کنید، ویزاردهایی برای فعال شدن این ابزار را می بینید.

- Clear Memory

تمام یا یکی از انواع حافظه های خاصی که در پروژه مورد اسـتفاده قـرار مـی گیـرد، ماننـد حافظـه برنامـه، GPRS، داده، EEPROM یا بیت تنظیمات را پاک می کند.

6. Programmers

در MPLAB از این Microchip programmer ها پشتیبانی می شود:

Programmer Name	Devices Programmed*
MPLAB PM3 Device Programmer**	PIC MCUs, dsPIC DSCs
PRO MATE II Device Programmer** (Obsolete)	PIC MCUs, Memory devices, KeeLoq devices
PICSTART Plus Development Programmer	Most PIC MCUs
PICkit 1 or 2 Development Programmers	Selected PIC MCUs
AN851 Quick Programmer	PIC16F877A, PIC18F452
MPLAB ICD 2 In-Circuit Debugger	PIC Flash MCUs, dsPIC Flash DSCs

برای دیدن لیستی از ابزارهای مورد پشتیبانی در محیط MPLAB میتوانید قسمتهای مربوطه را در فایل راهنمای MPLAB ببینید.

ورژن های خط فرمان این ابزارها از فایل pm3cmd.exe و procmd.exe قابل دسترسی هستند. یک نسخه ویژوالی از این ابزارهای خط فرمان از طریق فایل vprocmd.exe قابل دسترسی است. تمام این فایلها را در مسیر نصب برنامه و در "Programmer Utilities" میتوانید پیدا کنید.

فایلهای راهنما:

از منوی Help گزینه Topic را انتخاب کنید و در پنجره ای که ظاهر می شود در زیر عنوان Programmer لیست زیر را ببینید. با انتخاب هر کدام از این موارد می توانید راهنمایی هایی راجع به هر Programmer ببینید.

- MPLAB PM3
- PRO MATE II
- PICSTART Plus
- AN851 Quick Programmer

- PICkit 2 Programmer

همچنین، در زیر فهرست Debugger می توانید گزینه زیر را انتخاب کنید:

- MPLAB ICD 2

7. Tools

- Data Monitor and Control Interface

Use this interface in support of various applications. See the separate help file for this interface (Help>Topics, "Tools").

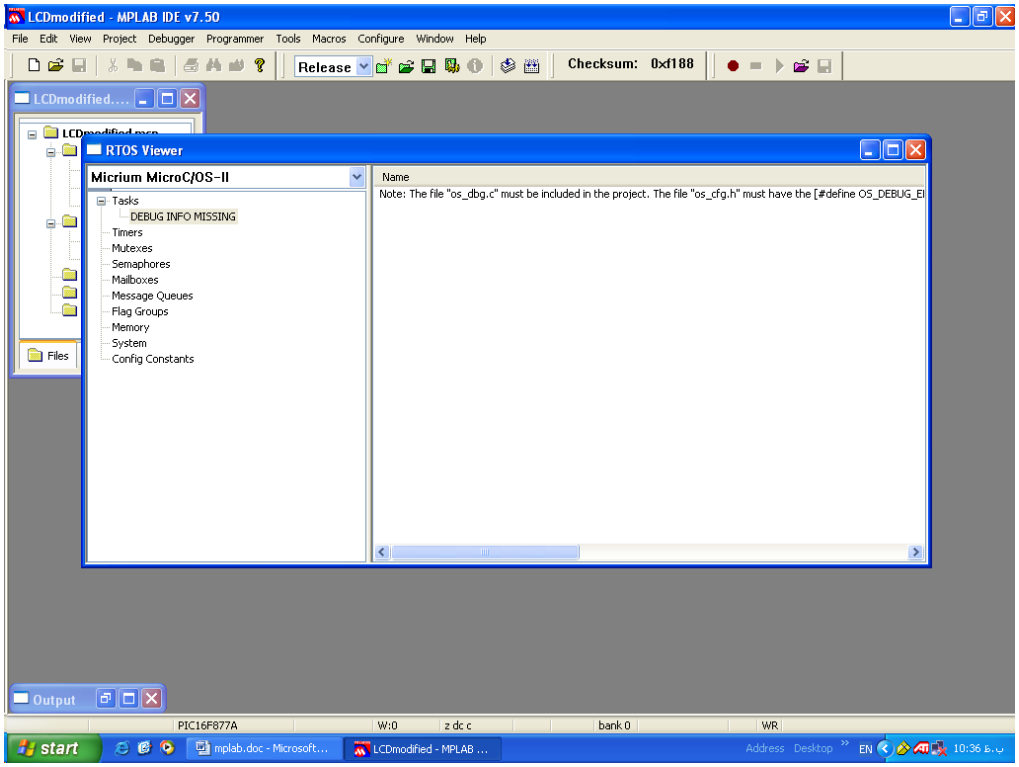
با انتخاب این گزینه پنجره ای با همین عنوان باز می شود، این رابط برنامه های کاربردی متفاوتی را پشتیبانی می کند. برای اطلاعات بیشتر از منوی Help گزینه Topics را انتخاب کرده و در کادر متنی Tools را وارد کنید.

- MPLAB Macros

با این گزینه قابلیت استفاده از ماکرو در محیط MPLAB فعال می شود. و نوارابزار آن در محیط اضافه می شود.

- RTOS Viewer

اگر شما یک سیستم عامل زمان واقعی (یا Real-Time) بر روی سیستمتان نصب کرده اید و در پروژه تان گنجانده شده، می توانید پنجره ناظر را باز کنید.



شکل 2-22- نمایی از نرم افزار MPLAB

- KeeLoq Plugin

با انتخاب این گزینه می توانید یک فایل SQTP را برای ابزار KeeLoq انتخابی ایجاد کنید. برای اینکه بفهمید چطور یک فایل SQTP را برای یک وسیله program کنید، راهنمای programmer را ببینید.

- Visual Initializer

بصورت ویژوالی با انتخاب این گزینه کدهایتان را مقداردهی کنید. برای اینکه ببینید این دستور چطور عمل می کند از Help، Topics را انتخاب کرده و عبارات Tools را وارد کنید.

8. Macros

در این منو گزینه هایی برای ضبط یک ماکرو جدید و ذخیره آن و استفاده از ماکروهای موجود وجود دارد.

- Record Macro:

بعد از انتخاب این گزینه عملیات انجام شده از این لحظه تا زمانی کـه Stop Recording را انتخـاب نکـرده باشید، به عنوان یک ماکرو ضبط می شود.

- Stop Recording:

برای توقف ماکرو، (پایان ضبط ماکرو) از این گزینه استفاده می شود. تا قبل از شروع ضبط یـک مـاکرو ایـن گزینه غیرفعال است.

- Play Macro:

ماکرو ضبط شده را اجرا می کند.

- Open Macro:

یک ماکرو که قبلاً ضبط شده و ذخیره شده را می توانید با این گزینه باز کنید.

- Save Macro:

برای ذخیره ماکرو ضبط شده استفاده می شود.

- Exit Macro:

از ماکرو خارج می شوید.

9. Configure

نکته: تمام گزینه های این منو با توجه به وسیله و ابزارهای اشکالزدایی که انتخاب کرده اید نمایش داده مـی شود.

- Select Device

با این گزینه می توانید ابزارهای مورد استفاده تان را انتخاب کنید. ابزارهای مورد اسـتفاده در فهرسـت هـای Debugger, Programmer, Language and design tools با توجه به ابزار انتخابی شما ست می شوند.

- Configuration Bits

با انتخاب این گزینه در پنجره ظاهر شده می توانید بیت های تنظیمات وسیله (Configuration bits) را تغییر دهید. (با انتخاب از لیست ها) با هر تغییر مقدار Configuration word تغییر خواهد کـرد. تنظیمـات ایـن مقدار، هم در عملیات Debugger و هم در عملیات Programmer تأثیر خواهد داشت.

- External Memory

با این گزینه مشخص می کنید که آیا می خواهید از حافظه جانبی (خارجی) استفاده کنید یا نـه. همچنـین می توانید مقدار استفاده از حافظه جانبی را هم مشخص کنید.

- ID Memory

با انتخاب این گزینه مقدار را به داخل حافظه ID یوزر منتقل می کنید.

- Settings

تنظیمات پیش فـرض بـرای workspace, debugger, program loading, hot keys و projects را در این پنجره می بینید. می توانید تنظیمات را تغییر دهید.

10. window

- Close All

تمام پنجره های باز بسته می شود.

- Cascade

تمام پنجره ها به فرم روی هم افتاده قرار می گیرند، طوری که نوار عنوان هر پنجره کاملاً قابل دیدن باشد.

- Tile Horizontally

تمام پنجره های باز را در اندازه های کوچکتر مرتب می کند بطوری که همه در یک راستای افقی در صـفحه نمایش fit شوند.

- Tile Vertically

تمام پنجره های باز را در اندازه های کوچکتر مرتب می کند بطوری کـه همـه در یـک راسـتای عمـودی در صفحه نمایش fit شوند.

- Arrange Icons

تمام پنجره های iconized را در قسمت بالای محیط مرتب می کند.

- Create Window Set

با این گزینه شکل جاری پنجره ها و ابزارها را در یک فایل ذخیره می کنیـد، کـادر محـاوره ای نمـایش داده می شود تا نام Window Set جدید را در آن وارد کنید. با این ذخیره سازی هیچ اطلاعـات خاصـی در ارتبـاط بـا پروژه ذخیره نخواهد شد. به این ترتیب در دفعات بعد، شما از این چیدمان برای پروژه های بعدی می توانید استفاده کنید.

نکته: اطلاعات داخل یک پروژه در فایل Window Set ذخیره نخواهد شد.

- Window Sets

می توانید با این گزینه یکی از window set هایی که قبلاً تعریف کرده اید را استفاده کنید.

- Destroy Window Set

خوب! برای پاک کردن یک window set هم این گزینه باید استفاده شود.

- Open windows

یک لیست از تمام پنجره های باز در انتهای این منو نمایش داده می شود. برای روی نام پنجـره دلخواهتـان کلیک کنید تا بعنوان پنجره فعال انتخاب شود.

- More windows

ممکن است لیست پنجره های باز شما طولانی باشد، با انتخاب این گزینـه در یـک کـادر محـاوره لیسـتی از تمام پنجره های باز وجود دارد که می توانید به این ترتیب پنجره فعالتان را مشخص کنید.

نکته: اگر نام مسیر خیلی طولانی بود و امکان اسکرول کردن بصورت افقی هم وجود نداشت، تلاش کنید تا header مربوط به ستون مسیر را تغییر اندازه بدهید تا مطمئن شوید که نوار اسکرول قابل دید است.

Help Topics Dialog 4-2

برای بازکردن این پنجره از منوی Help، گزینه Topics را انتخاب کنید. از این کادر محاوره برای نمایش یک فایل راهنما می توانید استفاده کنید. فایلهای راهنما برطبق نوع ابزارهای توسعه دهنده ای که محیط را توصیف می کنند مرتب شده اند، مثل System, Language Tools, Debuggers, Programmers و Other Tools مرتب شده اند. ممکن است در هر لحظه فقط بتوانید یک فایل راهنما را برای بازشدن انتخاب کنید.

بر روی انتخاب خود دابل کلیک کنید یا گزینه مورد نظرتان را با یک بار کلیک انتخاب کرده و Ok را بزنید. برای بستن این پنجره بدون باز کردن هیچ فایل راهنما روی دکمه Cancel کلیک کنید.

2-5 نوارابزارها:

2-5-1 - نوار ابزار استاندارد:

متشکل از دکمه هایی برای عملیات زیر است:

1. new file
2. open file
3. save file
4. cut
5. copy
6. paste
7. print file
8. find
9. find in file
10. help

از گزینه print file برای چاپ پنجره فعال استفاده می شود.

انتخاب گزینه help معادل انتخاب Topics از منوی Help است و با این انتخاب کادر محاوره MPLAB HELP TOPICS باز می شود.

مابقی گزینه ها پیشتر به تفضیل شرح داده شد.

2-5-2 - نوار ابزار project manager 2:

این نوارابزار متشکل از گزینه های زیر است:

1. build configuration

(که یک لیست با دو حالت انتخاب Debug, Release است.)

2. new project
3. open project
4. save workspace
5. build option
6. toolsuite info
7. make
8. build all

که همگی قبلاً بصورت مشروح معرفی شده اند.

2-5-3-نوار ابزار checksum:

این نوارابزار موقعیت اجرای برنامه را در حافظه نشان می دهد.

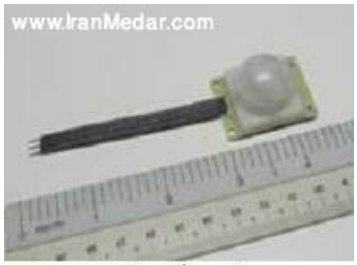
فصل سوم

3. حسگرها:

حسگر یک وسیله الکتریکی است که تغییرات فیزیکی یا شیمیایی را اندازه گیری مـی کنـد و آن را بـه سیگنال الکتریکی تبدیل می نماید. حسگرها در واقع ابزار ارتباط ربات با دنیای خارج و کسب اطلاعـات محیطی و نیز داخلی می باشند. انتخاب درست حسگرها تأثیر بسیار زیادی در میزان کارایی ربـات دارد. بسته به نوع اطلاعاتی که ربات نیاز دارد از حسگرهای مختلفی می توان استفاده نمود:

- **فاصله**
- **رنگ**
- **نور**
- **صدا**
- **حرکت و لرزش**
- **دما**
- **دود و...**

حسگرهاي رطوبت



يك حسگر حركت

حسگرها اطلاعات مورد نیاز ربات را در اختیار آن قرار مـی دهنـد و کمیتهـای فیزیکـی یـا شـیمیایی موردنظر را به سیگنالهای الکتریکی تبدیل می کنند.

3-1 حسگرهای مورد استفاده در رباتیک:

در یک دسته بندی کلی حسگرهای مورد استفاده در رباتها را می توان در یک دسته خلاصه کرد:

3-1-1 – حسگرهای تماسی (Contact)

مهمترین کاربردهای این حسگرها به این شرح می باشد:

- **آشکارسازی تماس دو جسم**
- **اندازه گیری نیروها و گشتاورهایی که حین حرکت ربات بین اجزای مختلف آن ایجاد می شود .**

در شکل یک میکرو سوئیچ یا حسگر تماسی نشان داده شده است. در صورت

برخورد تیغه فلزی به مانع و فشرده شدن کلید زیر تیغه همانند قطع و وصـل شـدن یـک کلیـد ولتـاژ خروجی سوئیچ تغییر می کند.

یک میکرو سوئیچ

- **حسگرهای هم جواری (Proximity)**

آشکارسـازی اشـیا نزدیـک بـه روبـات مهمتـرین کـاربرد ایـن حسـگرها مـی باشـد. انواع مختلفی از حسگرهای هم جواری در بازار موجود است از جمله می توان به موارد زیر اشاره نمود:

- **القایی**
- **اثرهال**
- **خازنی**
- **اولتراسونیک**

یک حسگر اثرهال

- **نوری**

3-1-2- حسگرهای دوربرد (Far awa)

کاربرد اصلی این حسگرها به شرح زیر می باشد:

- **فاصله سنج (لیزو و اولتراسونیک)**
- **بینایی (دوربینCCD)**

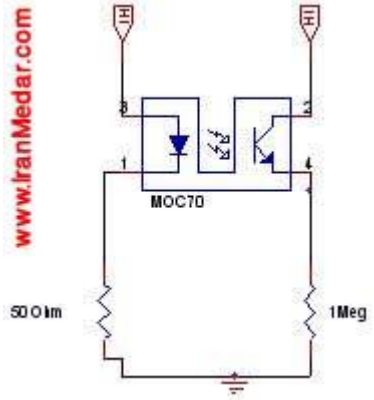
شکل 3-1 فاصله سنج

در شکل یک زوج گیرنده و فرستنده اولتراسونیک (ماورا صوت) نشان داده شده است. اساس کاراین حسگرها بر مبنای پدیده داپلر می باشد.

3-1-3- حسگر نوری (گیرنده-فرستنده)

یکی از پرکاربردترین حسگرهای مورد استفاده در ساخت رباتها حسگرهای نوری هستند. حسگر نوری گیرنده-فرستنده از یک دیود نورانی (فرستنده) و یک ترانزیستور نوری (گیرنده) تشکیل شده است.خروجی این حسگر در صورتیکه مقابل سطح سفید قرار بگیرد 5 ولت و در صورتی که در مقابل یک سطح تیره قرار گیرد صفر ولت می باشد. البته این وضعیت می تواند در مدلهای مختلف حسگر برعکس باشد. در هر حال این حسگر در مواجهه با دو سطح نوری مختلف ولتاژ متفاوتی تولید می کند.

در زیر یک نمونه مدار راه انداز زوج حسگر نوری گیرنده فرستنده نشان داده شده است. مقادیر مقاوتهای نشان داده شده در مدلهای متفاوت متغییر است و با مطالعه دیتا شیت آنها می توان مقدار بهینه مقاومت را بدست آورد.



شکل 3-2 نمای داخلی سنسور نوری

شکل 3-3 سنسور نوری

3-2 سنسور بویایی:

بویایی حسی است که می تواند در جلوگیری از جرائم استفاده شود. از گذشته از حس بویایی سگ ها برای یافتن اجساد ، مواد مخدر یا مواد منفجره و حتی شناسایی افراد استفاده می شده است. دانشمندان و مهندسان دارند بر روی سیستمی کار می کنند که بتواند بو را احساس کند. این تکنولوژی می تواند به جای سگ ها به کار برده شود و وظایف آن ها را انجام دهد.بو مخلوطی از مواد شیمیایی موجود در هوا است. حیوانات و ماشین ها برای بوییدن باید از چندین سنسور متفاوت استفاده کنند. هر سنسور به دسته ی خاصی از مواد شیمیایی حساس است. با اندازه گیری نتایج سنسور ها می توان بو را تشخیص داد.یک سنسور بویایی می تواند از یک کریستال کوارتز با اتصالات الکترونیکی و یک روکش پلاستیکی خاص درست شده باشد. کریستال کوارتز به منظور ایجاد یک لرزه منظم با یک فرکانس دقیق استفاده می شود. روکش پلاستیکی می تواند مواد شیمیایی را جذب کند .

3-3 سنسورهای رنگی:

سنسورهای رنگی ساخته شده در صنعت کار تشخیص رنگ را بر عهده دارند و تقریبا تمام طیف نور مرئی را تشخیص می دهند. این سنسورها بهترین انتخاب برای کاربردهای مقیاس کوچک با عملکرد بالا هستند. مورد

استفاده سنسور رنگ در صنعت مدرن بسیار زیاد است که از آن جمله می توان به کاربرد آن در خطوط تولید کنترل کیفیت (Q.C) ،ماهواره ها، رباتیک ،پزشکی ،صنایع غذایی ، اتومبیل سازی ، و به طور کلی در اتوماسیون سیستمها و سیستمهای اتوماتیک اشاره کرد.

ابزارهای بسیاری نیاز به کنترل رنگ از طریق یک سنسور را دارند به عنوان مثال در صنعت، کدهای رنگی پرینت شده همانند خطوط روی مقاومتها بایستی آشکار شوند تا اجازه عملکرد بالای ذخیره اتوماتیک رابه مقاومت بدهند . نیمی از شرکتهای آلمانی در صنعت غذایی (tetra) جهت کنترل رنگی از color sensor استفاده می کنند.

سه نوع سنسور رنگ شبیه چشم انسان مسئولیت تشخیص رنگ را بر عهده دارند . این سنسورهای رنگی نمایانگر یک طرح کوچک فیلترهایی با کیفیت بالا و خواندن هم زمان سه سطح رنگی می باشند.

شکل 3-4 سنسور رنگی

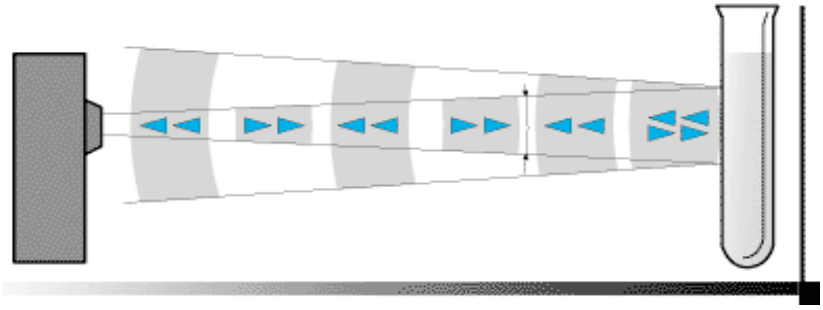
سنسور فوق یکی از انواع سنسورهای JEN color ساخت کارخانه MAZET آلمان و از نوع سنسورهای سه عنصری (3 element color sensor) بوده که قابلیت شناسایی رنگ ها را به تفکیک رنگهای قرمز ، سبز و آبی دارا می باشد.

سنسور مربوط از 3*19 فتو دیود pin سیلیکونی که بصورت حلقه وار روی چیپ فقرار گرفته اند تشکیل شده برای جلو گیری از ایجاد تداخل در بین فتو دیود ها ، هر سکتوری از قسمت دیگر جدا شده برای شناسایی هر رنگی در هر کدام از این فتودیود ها فیلتر مربوط به طول موج رنگ مربوط در نظر گرفته شده است.

سنسورهای التراسونیک:

سنسورهای التراسونیک چگونه کار می‌کنند؟

امواج صوتی با فرکانس‌های بالاتر از فرکانس شنوایی (امواج التراسونیک) را می‌فرستند و امواج بازگشتی را دریافت می‌کنند. از تاخیر زمانی و سرعت صوت در هوا برای تعریف فاصله از هدف استفاده می‌کنند و همچنین می‌توان تنها برای تشخیص هدف و وجود یا عدم وجود آن مورد استفاده قرار گیرد.

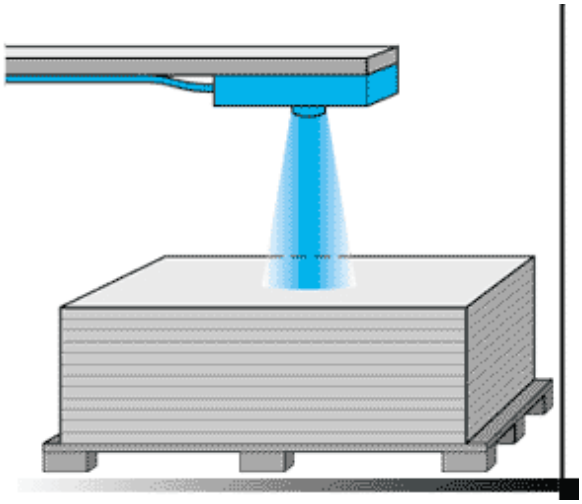


شکل 3-5 عملکرد اولتراسونیک

3-3-1انواع سنسورهای التراسونیک:

- **Ultrasonic proximity sensor with analog output stage**

خروجی‌های جریان و ولتاژ متناسب با فاصله سنسور از هدف هستند.



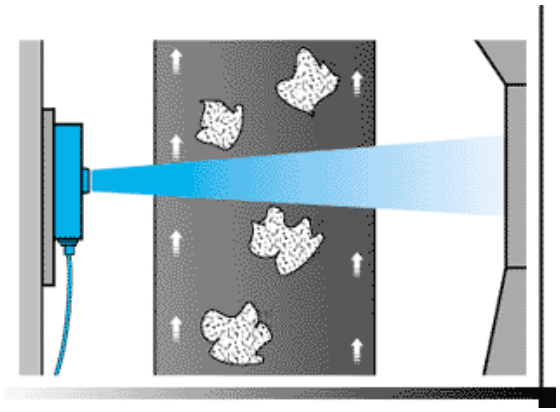
شکل 3-6 تنظیم فاصله سنسور

- **Ultrasonic retro-reflective sensor**

در این حالت از یک قطعه صاف و ثابت یک ماشین به‌عنوان بازتابنده استفاده می شود.

فاصله زمانی بین ارسال و دریافت سیگنال التراسونیک (زمان انتشار) ثابت و شناخته شده است.

وقتی که یک شیئ سیگنال التراسونیک را قطع می کند خروجی فعال می‌شود.

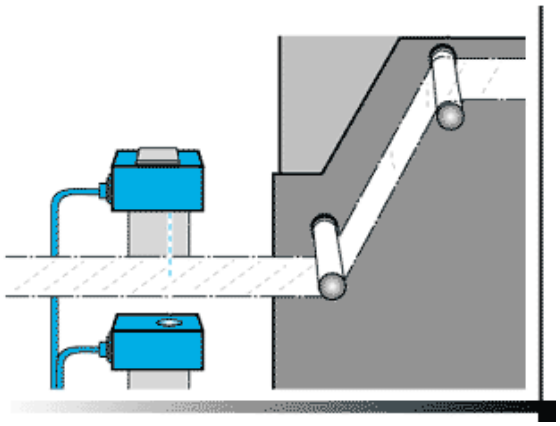


شکل 3-7 قطع سنسور توسط شیء

- **Ultrasonic through beam sensor**

این سنسورها برای کاربردهایی که اشیاء به‌سرعت و پشت سر هم در حرکتند ایدآل هستند.

این سنسورها همچنین زمانی که فرکانس‌های سویچینگ بالا (حدودا 200Hz) مورد نیاز باشد، پیشنهاد می‌شوند.



شکل 3-8 Ultrasonic through beam sensor

3-3-2 کاربرد سنسورهای التراسونیک:

- اندازه گیری زاویه (Angular Measurement)
- مسافت‌یابی (Ranging)
- تستهای غیر مخرب (Non Destructive Test)
- اندازه‌گیری جریان (Flow Metering)
- Non-intrusive medical procedures

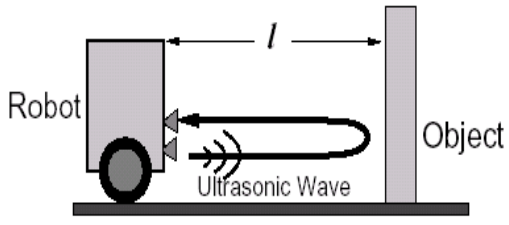
3-3-3مسافت یابی (Ranging):

روش های مسافت یابی:

- Time of Flight Measurement
- Measurement of Phase Difference

3-3-4روش TOF :

در روش TOF یک موج صوتی توسط سنسورهای التراسونیک مسافت‌یاب ارسال شده و فاصله زمانی که طول می‌کشد تا موج صوتی به شیئ برخور کند و به منبع برگردد محاسبه می‌شود.



The principles of the time-of-flight (TOF) method.

شکل 3-9 ارسال و بازگشت سیگنال

3-3-5روش اندازه گیری اختلاف فاز:

اگر یک موج التراسونیک شامل بیش از یک سیگنال باشد، اختلاف فاز بین سیگنال‌ها می‌تواند اندازه‌گیری شود.

روش اختلاف فاز خیلی دقیق است اما دارای این محدودیت است تنها از یک سیگنال با فرکانس خاص به عنوان مثال فرکانس 40 kHz می‌تواند استفاده کند و حداکثر فاصله ای که می‌تواند توسط این روش detect (یافتن) شود به 8mm محدود می‌شود.

3-4 کاربردهای مسافت یابی (Ranging):

Roboticsرباتیک

Auto focus in Camera

PDC (Parking Distance Control)کنترل فاصله پارک کردن

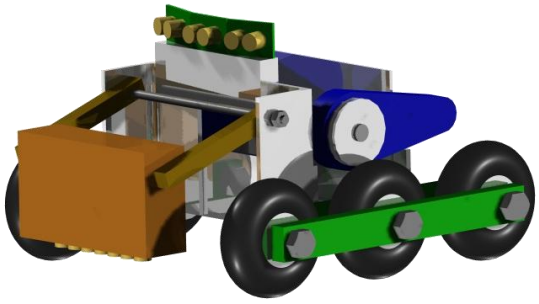
Fishing and underwater application

کاربرد سنسورهای Ultrasonic در رباتیک:

سنسورهای التراسونیک در رباتیک جهت مسافت یابی (Ranging) استفاده می‌شوند.

مسافت یابی در رباتیک عموما بر پایه روش TOF است.

مشکل اصلی در این‌کاربرد تداخل امواج (Crosstalk) است.



شکل 3-10 نمایی از روباتهای دارای سنسور

یکی از مسائل مطرح در رباتیک ایجاد درک نسبت به محیط خارجی برای جلوگیری از برخورد نامطلوب به اشیاء موجود در محیط حرکت است. از سوی دیگر ممکن است نیاز داشته باشیم که ربات بتواند درکی از فاصله ها بدون تماس فیزیکی داشته باشد. برای این منظور از سنسورهای مافوق صوت یا Ultrasonic استفاده میکنند .با وجود اینکه رویکردهای زیادی در این زمینه وجود دارد ولی میتوان آنها را در دو بخش تقسیم بندی کرد. دسته اول شامل ابزارهای انفعالی میباشند ،نظیر سیستمهای فاصله سنجی swept-focus و یا . stereoscopic دسته بعد سیستم های فعال یا Active میباشند نظیر سیستمهای ماکروویو ، لیزر و مافوق صوت .در این مقال ما به معرفی سنسورهای مافوق صوت خواهیم پرداخت. این سنسورها از دو قسمت تشکیل شده است. قسمت اول مدار راه انداز آن را تشکیل میدهد و قسمت دیگر دو قطعه (مبدل)گیرنده و فرستنده آن ، دقیقا مشابه آن قسمت از دزدگیرهایی که در خودروها (مقابل شیشه جلو) نصب میشود. البته دردسر اصلی کار با اینگونه سنسورها مدار راه انداز آن است.البته پکیجهای آماده که کار را بسیار ساده میکنند نیز وجود دارد، مانند مدل مافوق صوت ساخت شرکت Texas Instrumentsکه این سنسورها در برخی دوربینها نیز برای تشخیص فاصله و فوکوس مناسب استفاده میشود .

مکانیزم کلی کار این سنسورها ، فرستادن یک بیم و دریافت انعکاس آن و متعاقبا محاسبه زمان رفت و برگشت. بدین ترتیب میتوان فواص را نیز براحتی با در نظر گرفتن صرعت صوت در دما و فشار محیط ، محاسبه کرد .

قدم بعدی بدست آوردن ماتریسی از موانع موجود در محیط است. اینکار از دو راه ممکن است راه اول جاروب کردن محیط با امواج بصورت مکانیکی میباشد. راه دوم استفاده ازچند مبدل ، با توجه به پیچیدگی محیط ، است. بعنوان مثال میتوان یک مبدل متحرک با رنج زاویه ای بالا در سر ربات ، یک مبدل ثابت در جلو و رو به پایین برای تشخیص گودی، و دو مبدل با زاویه های 45 درجه در چپ و راست را بعنوان یک ترکیب مناسب استفاده کرد .

3-4-1 نمونه ای از کاربرد سنسورهای Ultrasonic در روباتیک:

شکل 3-11– روبات دوچرخه سوار

3-4-1-1 ربات دوچرخه سوار

یک کمپانی ژاپنی اقدام به ساخت یک ربات کوچک نموده که به سادگی دوچرخه سواری می کند.
شرکت Murata Manufacturing در یکی از نمایشگاه های تکنولوژیهای پیشرفته در اطراف شهر توکیو در ژاپن، رباتی را معرفی کرد که قادر است دوچرخه سواری کند.

این ربات که پنجاه سانتیمتر ارتفاع و نزدیک به پنج کیلوگرم وزن دارد، "پسر موراتا" (Murata Boy) نام داشته می تواند با سرعتی حدود 79 سانتیمر در ثانیه حرکت کند.

نسخه قدیمی تر این ربات که دوچرخه سواری می کرد در سال 1990 معرفی شد اما نمی توانست بدون آنکه بیفتد، توقف کند. اما ربات اخیر با تنظیم سرعت و انحراف مرکز جرم خود می تواند تعادل خود را در موقعیت های مختلف از جمله هنگام ایستادن حفظ کند.

3-5 سنسور سونار:

سونار SRF04 سنسور مسافت یاب است که می توان توسط آن ربات را هدایت کرد. شما می توانید ربات خود قادر سازید تا محیط پیرامونش را از طریق مجموعه سنسورهای سونار ببیند.(مانند چشم انسان)

3-5-1 نظریه عملکرد:

یک سنسور سونار از طریق تولید یک صدا مانند رگبار کوتاه اسلحه کار می کند(ping) بنابراین وقتی که صدا به نزدیکترین شیء برخورد می کند، انعکاس صدا(echo) توسط سنسور شنیده می شود.
مانند تصویر زیر:

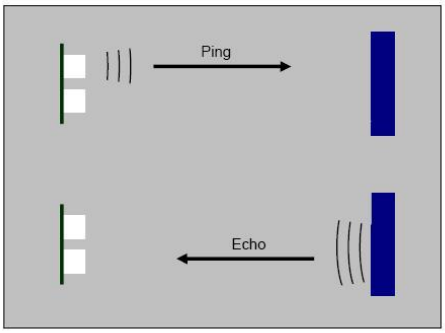


Figure 1 - Sonar Ping and Echo

شکل 3-12– انعکاس صدا

توسط اندازه گیری درست زمان، از لحظه ای که Ping شروع شده تا لحظه ای که Echo به سنسور برمیگردد، مقدار فاصله به نزدیکترین شیء را می توان محسبه کرد. حرکت صدا چیزی در حدود 1116.4 feet/second و یا 340.29 meters/second در سطح دریا، سرعت دارد. فاصله به نزدیکترین شیء را می توان با تقسیم زمان گذشته شده(elapsed time) بر دو برابر سرعت صدا محاسبه کرد.

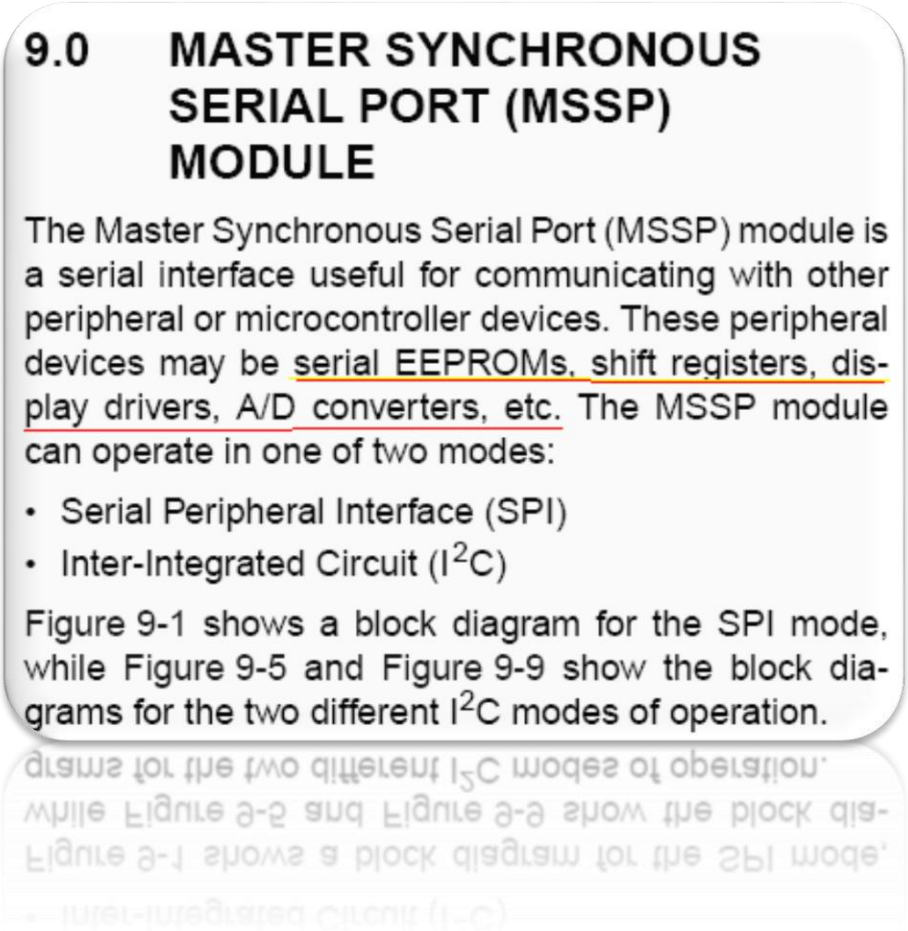
منظور از زمان گذشته شده، زمان بین فرستادن صدا و شنیدن انعکاس(echo) است.

فرمول بدست آوردن فاصله سنسور از شیء به صورت زیر است:

Distance = ElapsedTime / (2* Speed_Of_Sound)

فصل چهارم

4. ثبات های میکروکنترلر و کار با آن ها



9.0 MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc. The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I^2C)

Figure 9-1 shows a block diagram for the SPI mode, while Figure 9-5 and Figure 9-9 show the block diagrams for the two different I^2C modes of operation.

شکل 0-1– MSSP

1-4 پورت سریال همزمان-MSSP

تصویر ۱ برگرفته از دیتا شیت میکرو، MSSP را توضیح می دهد. پورت سریال همزمان "MSSP"، یک واسط سریال مفید برای مبادله بین وسایل جانبی یا ابزار میکروکنتلر است. این وسایل جانبی ممکن است ایپرام[1] های سریال باشد، یا شیفت رجیستر، درایور نمایش، مبدل آنالوگ به دیجیتال و غیره. این پورت در دو حالت کار می کند:

GISTER 12-1: CONFIGURATION WORD

CP1	CP0	DEBUG	—	WRT	CPD	LVP	BODEN	CP1	CP0	PWRTE	WDTE	F0SC1	F0SC0
bit13													bit0

Register: CONFIG
Address 2007h

bit 13-12:

bit 5-4: **CP1:CP0:** Flash Program Memory Code Protection bits (2)
11 = Code protection off
10 = 1F00h to 1FFFh code protected (PIC16F877, 876)
10 = 0F00h to 0FFFh code protected (PIC16F874, 873)
01 = 1000h to 1FFFh code protected (PIC16F877, 876)
01 = 0800h to 0FFFh code protected (PIC16F874, 873)
00 = 0000h to 1FFFh code protected (PIC16F877, 876)
00 = 0000h to 0FFFh code protected (PIC16F874, 873)

bit 11: **DEBUG:** In-Circuit Debugger Mode
1 = In-Circuit Debugger disabled, RB6 and RB7 are general purpose I/O pins.
0 = In-Circuit Debugger enabled, RB6 and RB7 are dedicated to the debugger.

bit 10: **Unimplemented: Read as '1'**

bit 9: **WRT:** Flash Program Memory Write Enable
1 = Unprotected program memory may be written to by EECON control
0 = Unprotected program memory may not be written to by EECON control

bit 8: **CPD:** Data EE Memory Code Protection
1 = Code protection off
0 = Data EEPROM memory code protected

bit 7: **LVP**: Low Voltage In-Circuit Serial Programming Enable bit
1 = RB3/PGM pin has PGM function, low voltage programming enabled
0 = RB3 is digital I/O, HV on MCLR must be used for programming

bit 6: **BODEN**: Brown-out Reset Enable bit (1)
1 = BOR enabled
0 = BOR disabled

bit 3: **PWRTE**: Power-up Timer Enable bit (1)
1 = PWRT disabled
0 = PWRT enabled

bit 2: **WDTE**: Watchdog Timer Enable bit
1 = WDT enabled
0 = WDT disabled

bit 1-0: **FOSC1:FOSC0**: Oscillator Selection bits
11 = RC oscillator
10 = HS oscillator
01 = XT oscillator
00 = LP oscillator

Note 1: Enabling Brown-out Reset automatically enables Power-up Timer (PWRT), regardless of the value of bit PWRTE. Ensure the Power-up Timer is enabled anytime Brown-out Reset is enabled.
2: All of the CP1:CP0 pairs have to be given the same value to enable the code protection scheme listed.

شکل 0-2– کلمه پیکر بندی

[1] ROM قابل برنامه نویسی و پاک کردن

واسط جانبی سریال (SPI)

مدار مجتمع داخلی (I^2C)

4-2 کلمه ی پیکربندی

تصویر۲ کلمه ی پیکربندی را نشان میدهد. این کلمه یا CF، کلمه ای ۱۴ بیتی است وهمچنان کـه در تصـویر ۲ قابل رویت است، هر بیت با نامی مشخص شده اکنون به شرح مهم ترین ها می پردازیم.

بیت ۰-۱: **FOSC1,FOSC0**: Oscillator Selection bits

با چهار حالت برای گزینش نوع و مدل اسیلاتور مورد استفاده است:

- 11 = اُسیلاتور RC
- 10 = اسیلاتور HS فرکانس بالا
- 01 = اسیلاتور XT اسیلاتوری زیر ۴مگاهرتز
- 00 = اسیلاتور LP

بیت ۲: **WDTEN**: Watchdog Timer Enable bit

- 1 = فعال کردن سگ نگهبان
- 0 = غیر فعال نمودن سگ نگهبان

سگ نگهبان را در دو حالت فعال و غیر فعال قرار می دهد. ویژگی سگ نگهبـان بـرای اسـیلاتورRC اختیـاری اسـت و بـدین ترتیـب بـه هـیچ ابـزار خـارجی دیگـری نیـازی نیسـت. بـا توجـه بـه مجـزا بـودن آن از پایـه ی OFC1/CLKIN حتی در صورت از کـار افتـادن سـاعت دو پایـه ی OFC1/CLKIN و OFC2/CLKOUT همچنان به کار خود ادامه می دهد. مثلا در هنگامی که دستور "Sleep" اجرا شده باشد.

در یک عملیات معمولی سگ نگهبان برای بازنشانی وسیله استفاده می شود. اگـر ابـزار در حالـت خـواب باشـد سگ نگهبان باعث بیدار شدن آن و ادامه دادن به عملیات نرمال می شود.

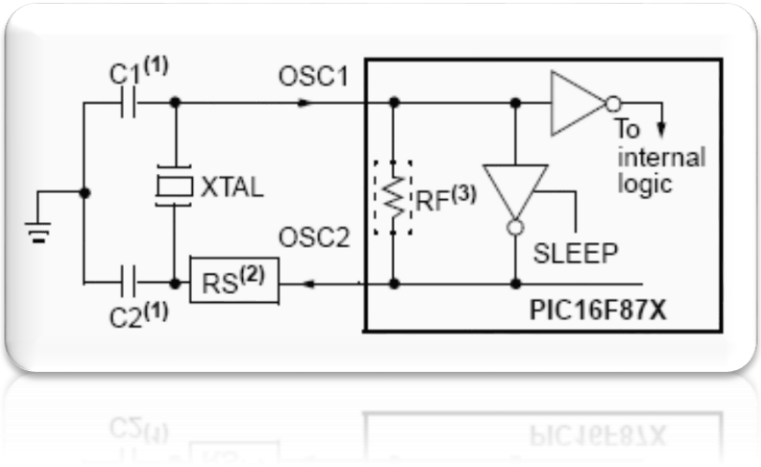
بیت ۳: **PWRTE** : Power-up Timer Enable bit

که در حالت صفر فعال می شود. ابتدا ۷۲ میلی ثانیه منتظر می ماند و سپس یک دستور پیش فرض را اجرا می کند تا بدین ترتیب منابع آماده شوند.

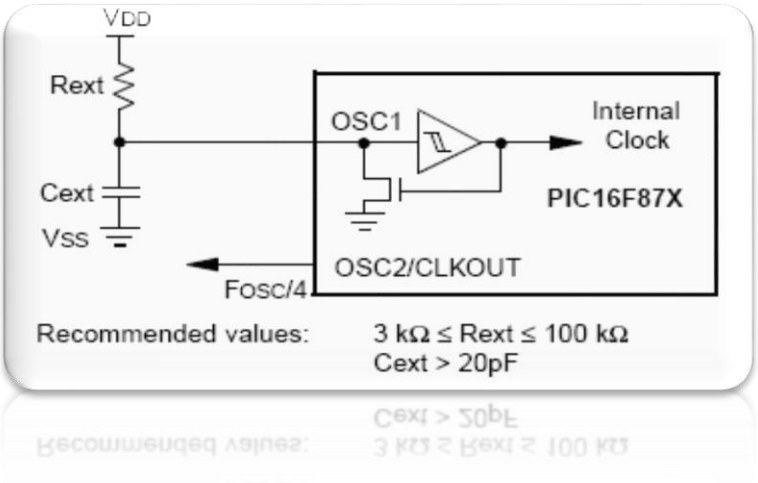
بیت ۶: **BODEN**: Brown-out Reset Enable bit

وقتی ورودی ضعیف باشد بازنشانده می شود. خاصیت Brown-Out Power برای حفاظت حافظه های فلش یا ایپرام در مقابل نوشتن، هنگامی که شرایطی پیش آمده که نباید داده ای روی آن نوشته شود، حیاتی است، یکی از روش های ازپیش ساخته Power-up، می باشد با تاخیر ۷۲ میلی ثانیه ای است.

نکته : برای جلوگیری از بازنشانده شدن میکرو حتما باید ولتاژ موتورها را از میکرو جدا کنید. مثلا استفاده از



شکل 0-3 - XP Oscillator



شکل 0-4 – RC Oscillator

یک باتری قلمی می تواند مفید باشد که با یک IC به ۵ ولت تبدیل می شود.

4-3 آشنایی با کریستال:

تصاویر۳ و ۴ نمایش گر اسیلاتور یا مدار تولید پالس ساعت مورد نیاز برای میکروکنتلر می باشد. اسیلاتور XP برای تولید پالس از کریستال استفاده می کند، البته نوع آن باید با میکروکنتلر تنظیم شود.خازن مورد استفاده در مدار بهتر است، ۲۲ پیکوفارادی انتخاب شود تا برای هر موردی مناسب باشد.

4-4 ثبات پیکربندی وقفه

رجییستر پیکربندی قابل نوشتن و خواندن است، این رجیستر حاوی بیت های فعال ساز و بیت علامت برای سرریز تایمر ، تعویض پورت و وقفه ی خارجی است.

بیت ۷: GIE :Global intrupt enable bit

وقفه ی سراسری در سطح یک فعال می شود. این بیت فعال کننده ی وقفه ی سراسری می باشد.

REGISTER 2-3: INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit7							bit0

R = Read
W = Writa
U = Unim
read
- n= Value

bit 7: **GIE:** Global Interrupt Enable bit
1 = Enables all un-masked interrupts
0 = Disables all interrupts

bit 6: **PEIE**: Peripheral Interrupt Enable bit
1 = Enables all un-masked peripheral interrupts
0 = Disables all peripheral interrupts

bit 5: **T0IE**: TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt

bit 4: **INTE**: RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt

bit 3: **RBIE**: RB Port Change Interrupt Enable bit

شکل 0-5 – پیکربندی وقفه

bit 2: **T0IF**: TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1: **INTF**: RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur

bit 0: **RBIF**: RB Port Change Interrupt Flag bit
1 = At least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state

بیت ۶: Pripheral intrupt enable bit **:PEIE**

- 1 = تمام وقفه های غیر قابل پوشش وسایل جانبی فعال می شوند.
- 0 = تمام وقفه های وسایل جانبی غیر فعال می شوند.

وقفه ی سراسری را فعال می کند، البته در سطح یک فعال می شود.

بیت ۵: **TMR0IE:** TMR0 Overflow Interrupt Enable bit

فعال ساز وقفه ی سرریز شمارنده ی ساعت "TMR0" . فعال شونده در سطح یک.

بیت ۴: **INTE:** RB0/INT External Interrupt Enable bit

بیت فعال کننده ی وقفه ی خارجی و فعال شونده در سطح یک.

بیت ۳: **RBIE**: RB port cheng intrupt enable bit

- 1 = وقفه ی تعویض پورت RB فعال می شود.
- 0 = وقفه ی تعویض پورت RB غیر فعال می شود.

بیت ۲: **TMR0IF:** TMR0 Overflow Interrupt Flag bit

بیت علامت سرریز شمارنده ی ساعت.

- 1 = ثبات زمان سنج سرریز کرده است و باید در برنامه پاک شود.
- 0 = هنوز سرریز نکرده یعنی هنوز از مقداری به آن داده ایم، کمتر است.

بیت ۰: **RBIF:** RB port cheng intrupt flag bit

بیت علامت وقفه ی تعویض پورت RB .

- 1 = وضعیت تمام پایه های RB7: RB4 تغییر می کندو باید در برنامه پاک شود
- 0 = وضعیت هیچ کدام از پایه های RB7: RB4 تغییر نمی کند.

REGISTER 5-1: OPTION_REG REGISTER

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7: **$\overline{\text{RBPU}}$**

bit 6: **INTEDG**

bit 5: **T0CS**: TMR0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4: **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3: **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0: **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

شکل 6-0 – Option Register

5-4 ثبات Option

تصویر ۶ – ثبات "Option" خواندنی نوشتنی است، شامل انواع بیت های کنترلی و پیکربندی TMR0 می باشد. در زیر برخی را معرفی می کنیم:

بیت ۰-۲: **PS2,PS0**: Prescaler Rate Select bits

بیت انتخاب سرعت، جدول زیر نشان گر حالات ممکن انتخاب است.

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

شکل 0-7- سه بیت مقیاس گذاری برای تایمر TMR0 و یا WDT

در حالت معمول تایمر-TMR0 - تا ۲۵۵ می شمارد و سپس دچار سر ریز می شود، می توان آن را با مقدار کم تری تنظیم نمود. روش دیگری برای کنترل TMR0 استفاده از PS می باشد. بر اساس جدول روبرو می توان سرعت آن را تغییر داد نکته ای که در این جا قابل ذکر است این است که برای استفاده از این امکان باید قبلا بیت ۳ یا همان "PSA" را در سطح مناسب قرار داده باشید.

بیت ۳: **PSA**: Prescaler Assignment bit

- 1 = Prescaler را به سگ نگهبان نسبت داده است.
- 0 = Prescaler را به تایمر - TMR0 - نسبت می دهد.

بیت ۴: **TOSE**: TIMER0 Source Edge select bit

بیت انتخاب لبه ی منبع تایمر صفر.

- 1 = افزایش روی پایه ی TOCKI را در لبه ی پایین رونده.
- 0 = افزایش روی پایه ی TOCKI را در لبه ی بالا رونده.

بیت ۶: **INTEDG**: Interrupt Edge Select bit

در صورتی که مایل باشیم وقفه در لبه ی بالا رونده فعال شود می توانیم این بیت را یک قرار دهیم.

مشخص کننده ی حالت فعال شدن وقفه است.

بیت ۷: **RBPU**: PORTB Pull-up Enable bit

این بیت باید مدام درسطح صفر باشد.

REGISTER 2-4: PIE1 REGISTER (ADDRESS 8Ch)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit7							bit0

bit 7: **PSPIE(1):** Parallel Slave Port Read/Write Interrupt Enable bit
1 = Enables the PSP read/write interrupt
0 = Disables the PSP read/write interrupt

bit 6: **ADIE**: A/D Converter Interrupt Enable bit
1 = Enables the A/D converter interrupt
0 = Disables the A/D converter interrupt

bit 5: **RCIE**: USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt

bit 4: **TXIE**: USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt

bit 3: **SSPIE**: Synchronous Serial Port Interrupt Enable bit
1 = Enables the SSP interrupt
0 = Disables the SSP interrupt

bit 2: **CCP1IE**: CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt
0 = Disables the CCP1 interrupt

bit 1: **TMR2IE**: TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0: **TMR1IE**: TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

Note 1: PSPIE is reserved on 28-pin devices; always maintain this bit clear.

شکل 0-8– PIE1 Register

6-4 ثبات PIE1

تصویر۷ نمایان گر ثبات PIE1 می باشد. این ثبات حاوی بیت های فعال سازی غیر قابل رویت برای وقفه های خارجی است. برخی از مهم ترین ها عبارت اند از:

بیت ۶: **ADIE**: A/D Converter Interrupt Enable bit

تبدیل کننده ی آنالوگ به دیجیتال. با استفاده از این بیت نیازی به چک کردن مداوم A/D نیست. برای استفاده از این ویژگی کافیست بیت۶ را در سطح یک قرار دهیم.

بیت ۲: **CCP1IE**: CCP1 Interrupt Enable bit

در این جا، CCP مخفف Capture Compare PWM می باشد که هر پالس PWM یک پالس دلخواه ایجاد می کند.

7-4 خصوصیات خارجی

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I^2C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2

شکل 0-9 – PWM در PIC های مختلف

ابتدا قدری در باره ی خصوصیات خارجی بدانیم، که شامل موارد زیر است.

- Timer0: ۸ بیت شمارنده / تایمر با ۸ بیت پیش مقیاس بندی
- Timer1: ۸ بیت شمارنده / تایمر با ۸ بیت پیش مقیاس بندی، که در طول خواب هم از طریق پالس ساعت کریستال خارجی، افزایش می یابد.
- Timer2: ۸ بیت شمارنده / تایمر با ۸ بیت پیش مقیاس بندی، رجیستر دوره ای و پس مقیاس بند.
- دو مدل Capture, Compare, PWM
  - Capture با ۱۶ بیت، بیشترین رزولوشن در ۵.۱۲ نانوثانیه

- Compare با ۱۶ بیت، بیشترین رزولوشن در ۲۰۰ نانوثانیه
- PWM بیشترین رزولوشن آن ۱۰ بیت است.

- پورت سریال همزمان -SSP .
- USART/SCI با ۹ بیت آدرس.
- پورت موازی پیرو با ۸ بیت طول ، کنترل انتخاب ، خواندن و نوشتن خارجی.(فقط در ۴۰/۴۴ پایه)
- (BOR) Brown-out

هر یک از مدل ها ی (CCP) Capture/Compare/PWM شامل یک ثبات ۱۶ بیتی که می تواند بـه عناوینی که قبلا ذکر رفت به کار رود. هم مدل CCP1 و هم CCP2 در آغاز عملیات، با بروز استثنای واقعـه ی به خصوصی ، یکسان اند. (CCPR1) Capture/Compare/PWM Register1 شامل دو ثبات ۸ بیتـی است: CCPR1L بایت کم ارزش و CCPR1H بایت پر ارزش است. ثبات CCP1CON عملیات CCP1 را

CCPx Mode	CCPy Mode	Interaction
Capture	Capture	Same TMR1 time-base
Capture	Compare	The compare should be configured for the special event trigger, which clears TMR1
Compare	Compare	The compare(s) should be configured for the special event trigger, which clears TMR1
PWM	PWM	The PWMs will have the same frequency and update rate (TMR2 interrupt)
PWM	Capture	None
PWM	Compare	None

شکل 10-0 – INTERACTION OF TWO CCP MODULES

کنترل می کند.

در ادامه به خصوصیات و تشریح بیت های مهم رجیستر PIR1 می پردازیم.

REGISTER 2-5: PIR1 REGISTER (ADDRESS 0Ch)

R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit7							bit0

R = Readable bit
W = Writable bit
- n= Value at POR reset

bit 7: **PSPIF(1):** Parallel Slave Port Read/Write Interrupt Flag bit
1 = A read or a write operation has taken place (must be cleared in software)
0 = No read or write has occurred

bit 6: **ADIF**: A/D Converter Interrupt Flag bit
1 = An A/D conversion completed
0 = The A/D conversion is not complete

bit 5: **RCIF**: USART Receive Interrupt Flag bit
1 = The USART receive buffer is full
0 = The USART receive buffer is empty

bit 4: **TXIF**: USART Transmit Interrupt Flag bit
1 = The USART transmit buffer is empty
0 = The USART transmit buffer is full

bit 7: **SSPIF**: Synchronous Serial Port (SSP) Interrupt Flag
1 = The SSP interrupt condition has occurred, and must be cleared in software before returning from the interrupt service routine. The conditions that will set this bit are:
SPI
A transmission/reception has taken place.
I^2C Slave
A transmission/reception has taken place.
I^2C Master
A transmission/reception has taken place.
The initiated start condition was completed by the SSP module.
The initiated stop condition was completed by the SSP module.
The initiated restart condition was completed by the SSP module.
The initiated acknowledge condition was completed by the SSP module.
A start condition occurred while the SSP module was idle (Multimaster system).
A stop condition occurred while the SSP module was idle (Multimaster system).
0 = No SSP interrupt condition has occurred.

bit 2: **CCP1IF**: CCP1 Interrupt Flag bit
Capture Mode
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare Mode
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM Mode
Unused in this mode

bit 1: **TMR2IF**: TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0: **TMR1IF**: TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

Note 1: PSPIF is reserved on 28-pin devices; always maintain this bit clear.

شکل 0-11 – PIR1 Register

8-4 ثبات PIR1

ثبات PIR1 شامل بیت های کنترلی علامت برای وقفه های خارجی است. نکته این که بیت های علامت وقفه، زمانی نشانده می شوند که شرایط یک وقفه وقوع یابد.

بیت ۷: **PSPIF:** Parallel Slave Port Read/Write Interrupt Flag bit

بیت علامت وقفه ی خواندن/نوشتن درگاه موازی پیرو

- 1 = یک عملیات خواندن یا نوشتن اتفاق می افتد(توسط نرم افزار پاک می شود.)
- 0 = خواندن و نوشتی انجام نمی شود.

بیت ۶: **ADIF:** A/D Converter Interrupt Flag bit

بیت علامت وقفه ی مبدل آنالوگ به دیجیتال

- 1 = یک محاوره ی A/D تکمیل شده.
- 0 = هنوز محاوره کامل نشده.

بیت ۵: **RCIF:** USART Receive Interrupt Flag bit

بیت علامت وقفه ی دریافت USART

- 1 = بافر دریافت USART پر است.
- 0 = بافر دریافت USART خالی است.

بیت ۴: **TXIF:** USART Transmit Interrupt Flag bit

بیت علامت وقفه ی انتقال USART

- 1 = بافر انتقال USART پر است.
- 0 = بافر انتقال USART خالی است.

بیت ۳: **SSPIF:** Synchronous Serial Port (SSP) Interrupt Flag bit

بیت علامت وقفه ی پورت سریال همزمان (SSP)

- 1 = شرایط وقفه ی SSP اتفاق افتاده است و باید قبل از بازگشت از سرویس روتین

وقفه، در نرم افزار پاک شود. شرایطی که این این بیت را می نشانند از این قرارند:

- SPI وقوع یک دریافت یا انتقال
- I^2C Slave وقوع یک دریافت یا انتقال
- I^2C Master
  - وقوع یک دریافت یا انتقال
  - شرایط START که توسط مدل SSP تکمیل شده.
  - شرایط STOP که توسط مدل SSP تکمیل شده.
  - شرایط Restart که توسط مدل SSP تکمیل شده.
  - شرایط Acknowledge که توسط مدل SSP تکمیل شده.

- شرایط START در هنگامی که مدل SSP بیکار است(Multi-Master system).
- شرایط STOP در هنگامی که مدل SSP بیکار است(Multi-Master system).

- 0 = هیچ وقفه ی SSP اتفاق نیافتاده است.

بیت ۲: **CCP1IF**: CCP1 Interrupt Flag bit

بیت علامت وقفه ی "CCP1"

مدل Capture:

- 1 = یک گرفتن یا جمع کردن ثبات TMR1 اتفاق افتاده است.
- 0 = هیچ گرفتن یا جمع کردن ثبات TMR1 اتفاق نیافتاده است.

مدل Compare:

- 1 = یک انطباق ثبات TMR1 اتفاق افتاده است.
- 0 = هیچ انطباق ثبات TMR1 اتفاق نیافتاده است.

برای مدل PWM استفاده نمی شود.

بیت ۱: **TMR2IF**: TMR2 to PR2 Match Interrupt Flag bit

بیت علامت وقفه ی تطبیق TMR2 با PR2

- 1 = تطبیق اتفاق افتاده.
- 0 = با هم منطبق نشده اند.

بیت ۰: **TMR1IF**: TMR2 to PR2 Match Interrupt Flag bit

بیت علامت وقفه ی سر ریز TMR1

- 1 = ثبات TMR1 سر ریز کرده.
- 0 = ثبات TMR1 سر ریزنکرده.

TABLE 12-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF
These values are for design guidance only. See notes at bottom of page.			
Crystals Used			
32 kHz	Epson C-001R32.768K-A		± 20 PPM
200 kHz	STD XTL 200.000KHz		± 20 PPM
1 MHz	ECS ECS-10-13-1		± 50 PPM
4 MHz	ECS ECS-40-20-1		± 50 PPM
8 MHz	EPSON CA-301 8.000M-C		± 30 PPM
20 MHz	EPSON CA-301 20.000M-C		± 30 PPM

Note 1: Higher capacitance increases the stability of oscillator but also increases the start-up time.

2: Since each resonator/crystal has its own characteristics, the user should consult the resonator/crystal manufacturer for appropriate values of external components.

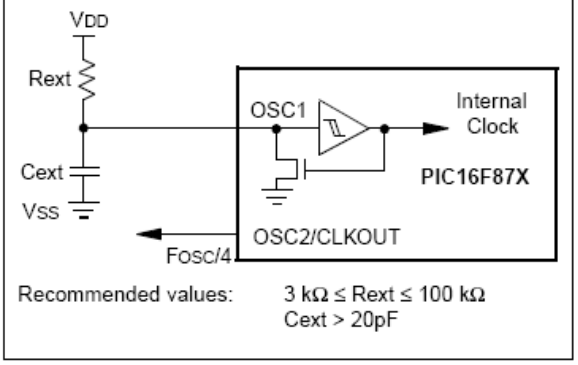
3: Rs may be required in HS mode, as well as XT mode, to avoid overdriving crystals with low drive level specification.

4: When migrating from other PICmicro devices, oscillator performance should be verified.

12.2.3 RC OSCILLATOR

For timing insensitive applications, the "RC" device option offers additional cost savings. The RC oscillator frequency is a function of the supply voltage, the resistor (REXT) and capacitor (CEXT) values, and the operating temperature. In addition to this, the oscillator frequency will vary from unit to unit due to normal process parameter variation. Furthermore, the difference in lead frame capacitance between package types will also affect the oscillation frequency, especially for low CEXT values. The user also needs to take into account variation due to tolerance of external R and C components used. Figure 12-3 shows how the R/C combination is connected to the PIC16F87X.

FIGURE 12-3: RC OSCILLATOR MODE



شکل 12-0 – RC Oscillator

توضیح اشکال ۹ و ۱۰ در انتهای تصویر ۱۰ آمده است.

12.2 Oscillator Configurations

12.2.1 OSCILLATOR TYPES

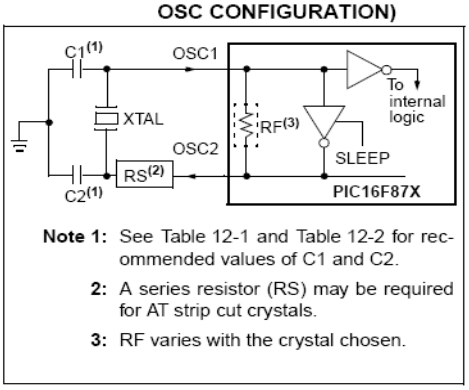
The PIC16F87X can be operated in four different oscillator modes. The user can program two configuration bits (FOSC1 and FOSC0) to select one of these four modes:

- LP Low Power Crystal
- XT Crystal/Resonator
- HS High Speed Crystal/Resonator
- RC Resistor/Capacitor

12.2.2 CRYSTAL OSCILLATOR/CERAMIC RESONATORS

In XT, LP or HS modes, a crystal or ceramic resonator is connected to the OSC1/CLKIN and OSC2/CLKOUT pins to establish oscillation (Figure 12-1). The PIC16F87X oscillator design requires the use of a parallel cut crystal. Use of a series cut crystal may give a frequency out of the crystal manufacturers specifications. When in XT, LP or HS modes, the device can have an external clock source to drive the OSC1/CLKIN pin (Figure 12-2).

FIGURE 12-1: CRYSTAL/CERAMIC RESONATOR OPERATION (HS, XT OR LP OSC CONFIGURATION)



Note 1: See Table 12-1 and Table 12-2 for recommended values of C1 and C2.

2: A series resistor (RS) may be required for AT strip cut crystals.

3: RF varies with the crystal chosen.

FIGURE 12-2: EXTERNAL CLOCK INPUT OPERATION (HS, XT OR LP OSC CONFIGURATION)

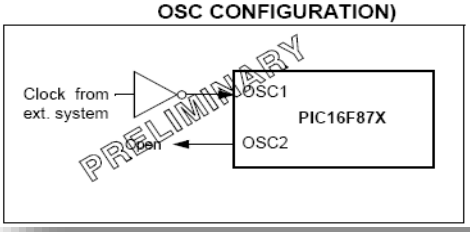


TABLE 12-1: CERAMIC RESONATORS

Ranges Tested:			
Mode	**Freq**	**OSC1**	**OSC2**
XT	455 kHz	68 - 100 pF	68 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF
	16.0 MHz	10 - 22 pF	10 - 22 pF
These values are for design guidance only. See notes at bottom of page.			
Resonators Used:			
455 kHz	Panasonic EFO-A455K04B	± 0.3%	
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%	
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%	
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%	
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%	
All resonators used did not have built-in capacitors.			

شکل 0-13 – Oscillator

4-9 اسیلاتور

در تصویر ۹ یک اسیلاتور RC به نمایش گذاشته شده است.

برای تنظیم زمان کاربردهای غیر حساس، ابزار RC باعث صرفه جویی است. فرکانس اسیلاتور "RC" تابعی از منبع ولتاژ و مقادیر مقاومت ، خازن و دمای عملیات است. علاوه بر این فرکانس اسیلاتور از واحدی به دیگری از پردازش عادی پارامترها متغییر است. حتی ممکن است تفاوت بین فریم هادی خازن ها در مدل بسته بندی های مختلف، باعث تاثیر روی فرکانس اسیلاتور شود. مخصوصا برای خازن هایی با مقدار پایین.

تصویر ۱۰ نشان گر یک اسیلاتور با کریستال است. میکروکنتلر PIC16F87XA در چهار مدل متفاوت اسیلاتورها پردازش کند. با استفاده از برنامه ریزی دو بیت پیکربندی FOSC1 و FOSC0 می توان یکی از این چهار مدل را انتخاب نمود:

- LP کریستال کم قدرت
- XT کریستال/رزوناتور
- HS کریستال/رزوناتور پر سرعت
- RC مقاومت/ خازن

در سه مدل اولی کریستال یا رزوناتور سرامیکی به پایه های OSC1/CLKIN و OSC2/CLKOUT

Osc Type	Crystal Freq	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF

شکل 0-14

متصل می شوند. طراحی اسیلاتور PIC16F87XA نیازمند برش موازی کریستال است. استفاده از یک سری

برش کریستال ممکن است باعث شود فرکانسی خارج از آنچه کریستال منحصرا برای آن ساخته شده بدهد.

REGISTER 6-1: T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit7							bit0

R = Readabl
W = Writable
U = Unimple
read as
- n = Value at

bit 7-6: **Unimplemented:** Read as '0'

bit 5-4: **T1CKPS1:T1CKPS0**: Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value

bit 3: **T1OSCEN**: Timer1 Oscillator Enable Control bit
1 = Oscillator is enabled
0 = Oscillator is shut off (The oscillator inverter is turned off to eliminate power drain)

bit 2: **$\overline{\text{T1SYNC}}$**: Timer1 External Clock Input Synchronization Control bit

TMR1CS = 1
1 = Do not synchronize external clock input
0 = Synchronize external clock input

TMR1CS = 0
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1: **TMR1CS**: Timer1 Clock Source Select bit
1 = External clock from pin RC0/T1OSO/T1CKI (on the rising edge)
0 = Internal clock (FOSC/4)

bit 0: **TMR1ON**: Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

شکل 16-0 – T1CON Register

Mode	Freq	OSC1	OSC2
XT	455 kHz	68 - 100 pF	68 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF
	16.0 MHz	10 - 22 pF	10 - 22 pF

شکل 15-0 – مقایسه ی XT و HS

4-10مدل TIMER1:

مدل Timer1 یک شمارنده ی ۱۶ بیتی است شامل دو ثبات ۸ بیتـی (TMR1H , TMR1L) کـه قابـل نوشتن و قابل خواندن است. این ثبات از 0000 h به FFFF h افزایش می یابد، اگر وقفـه ی TMR1 روی سـر ریز فعال شده باشد، که در بیت علامت وقفه Lache می شود. (PIR1<0>) TMR1IF این وقفه با فعـال/غیر فعال ساز وقفه ی TMR1 تنظیم می شود (PIE1<0>) TMR1IE. تایمر یک می تواند در دو مد زیر کار کند:

- به عنوان یک تایمر
- به عنوان یک شمارنده

مد پردازش با بیت انتخاب ساعت مشخص می شود (T1CON<1>) TMR1CS. در مد تـایمر یـا همـان زمان سنج TMR1 در هر سیکل دستور افزایش می یابد. در مد شمارنده، افزایش در هر لبه ی بالا رونده ی ورودی ساعت خارجی است. بیت کنترل TMR1، (T1CON<0>) TMR1ON است که می توان با آن تایمر یـک را فعال یا غیر فعال کرد.

تایمر۱، یک "ورودی بازنشاندن" نیز دارد. این RESET می تواند با هر دو مد CCP بوجود بیاید. وقتی اسیلاتور Timer1 فعال است (T1OSCEN)، پایه های RC1/T1OSI/CCP2 و RC0/T1OSO/T1CKI بعنوان ورودی منظور می شوند یعنی مقادیر TRISC<1:0> در نظر گرفته نمی شوند و مقدار صفر می گیرند.

4-11 ثبات T1CON: ثبات کنترل TIMER1 (آدرس: 10h)

بیت ۶-۷: بدون استفاده: به عنوان صفر در نظر گرفته می شوند.

بیت ۴-۵: **T1CKPS1,T1CKPS0**: Timer1 Input Clock Prescale Select bits
بیت انتخاب پیش مقیاس بندی ورودی ساعت Timer1

- 11 = 1:8مقدار پیش مقیاس (Prescale).
- 10 = 1:4مقدار پیش مقیاس.
- 01 = 1:2مقدار پیش مقیاس.

- 00 = 1:1مقدار پیش مقیاس.

بیت ۳: **T1OSCEN**: Timer1 Oscillator Enable Control bit
بیت کنترل فعال سازی اسیلاتورTimer1

- 1 = اسیلاتور فعال است.
- 0 = اسیلاتور خاموش شده.

بیت ۲: **T1SYNC:** Timer1 External Clock Input Synchronization Control bit
بیت کنترل همزمان سازی ورودی ساعت خارجی Timer1

هنگامی که TMR1CS = 1:

- 1 = ساعت خارجی را همزمان سازی نکن.
- 0 = ساعت خارجی را همزمان سازی می کند.

هنگامی که TMR1CS = 0:

این بیت نادیده گرفته می شود. تایمر از ساعت داخلی خود استفاده می کند.

بیت ۱: **TMR1CS**: Timer1 Clock Source Select bit
بیت انتخاب منبع ساعت Timer1

- 1 = ساعت خارجی از پایه ی RC0/T1OSO/T1CKI در لبه ی بالا رونده.
- 0 = ساعت داخلی (FOSC/4).

بیت ۰: **TMR1ON**: Timer1 On bit
بیت روشن کردن Timer1

- 1 = Timer1 فعال است.
- 0 = Timer1 متوقف شده.

REGISTER 7-1: T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit7							bit0

bit 7: **Unimplemented:** Read as '0'

bit 6-3: **TOUTPS3:TOUTPS0**: Timer2 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
0010 = 1:3 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2: **TMR2ON**: Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

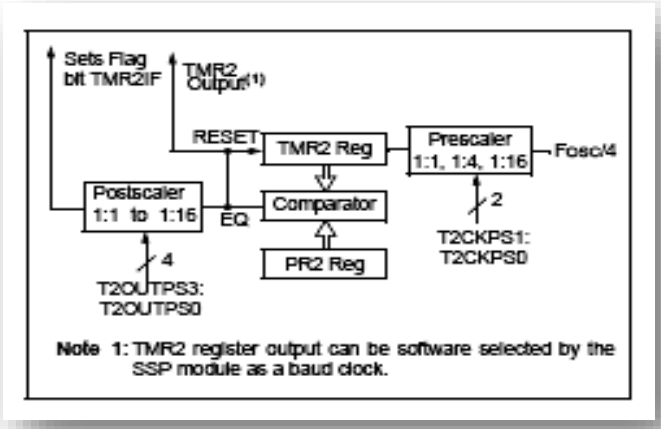
bit 1-0: **T2CKPS1:T2CKPS0**: Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

شکل 17-0 – T2CON Register

4-12مدل TIMER2:

مدل Timer2 یک شمارنده ی ۸ بیتی است با یک پیش و پس مقیاس بند. این می تواند به عنوان یک مبدا زمانی PWM برای مد PWM ازمدل هایCCP، به کا رود. این ثبات خواندنی نوشتنی است و پاک کردن آن فقط از طریق RESET می باشد.

ساعت ورودی (FOSC/4) حالات پیش مقیاس بندی 1:1, 1:4, or 1:16 را با انتخاب بیت کنترل T2CKPS1:T2CKPS0 (T2CON<1:0>) دارد.



شکل 0-18 - تایمر دو

مدل تایمر دو، یک ثبات ۸ بیتی پریود دارد PR2. این تایمر از 00h تا مقدار PR2 افزایش می یابد و سپس برای چرخه ی بعدی به 00h ریست می شود. ثبات PR2 یک ثبات خواندنی نوشتنی است و بعد از بازنشانی به مقدار FF h نشانده می شود.

13-4 ثبات T2CON: ثبات کنترل TIMER2 (آدرس: 12h)

بیت ۷: بدون استفاده: به عنوان صفر در نظر گرفته می شوند.

بیت ۳-۶: **TOUTPS3,TOUTPS0**: Timer2 Output Postscale Select bits

بیت های انتخاب پس مقیاس بند خروجی Timer2

- 0000 = 1:1مقدار پس مقیاس (Post scale).
- 0001 = 1:2مقدار پس مقیاس.
- 0010 = 1:3مقدار پس مقیاس.
- ...
- 1111 = 1:16مقدار پس مقیاس.

بیت ۲: **TMR2ON**: Timer2 On bit

بیت روشن کردن تایمر

- 1 = تایمر روشن است.
- 0 = تایمر خاموش است.

بیت ۰-۱: **T2CKPS1,T2CKPS0**: Timer2 Clock Prescale Select bits

بیت انتخاب پیش مقیاس بند ساعت Timer2

- 00 = پیش مقیاس بند ۱ است.
- 10 = پیش مقیاس بند ۴ است.
- 1x = پیش مقیاس بند ۱۶ است.

فصل پنجم

5. موتورهای الکتریکی

یکی از پرکاربرد ترین مولفه ها در رباتیک، موتورهای الکتریکی می باشند. موتورهای الکتریکی با تبدیل جریان الکتریسیته به حرکت، می تواند در کارهای گوناگونی از قبیل چرخاندن چرخها، پمپ های مختلف و به طور کلی در هر فعالیتی که به حرکت نیاز باشد، به کار رود.

موتورهای الکتریکی انواع مختلفی دارد که پرکاربرد ترین آنها در رباتیک عبارتند از: موتورهای DC، موتورهای پله ای و سرو موتورها. که هر یک را به طور مختصر در ادامه شرح می دهیم.

5-1 موتورهای DC

این نوع موتورها به عنوان اولین موتورهای الکتریکی شناخته می شوند که مکانیسم بسیار ساده ای داشته و به دلیل ایجاد نویز در مدارها و اینرسی حرکتی ، کمتر در رباتیک مورد استفاده قرار می گیرد.سرعت موتور DC به مجموعه ای از ولتاژ و جریان عبوری از سیم پیچهای موتور و بار موتور یا گشتاور ترمزی ، بستگی دارد.

سرعت موتور DC وابسته به ولتاژ و گشتاور آن وابسته به جریان است.

شاید یکی از دلایل استفاده از این نوع موتورها در رباتیک، قیمت مناسب و در دسترس بودن آن می باشد. نمونه ای از یک موتورDC را در شکل مشاهده می نمایید.

شکل 5-1-موتور DC

5-2 موتورهای پله ای

نوع دیگری از موتور ها که تقریبا کاربرد زیادی در رباتیک دارند، موتورهای پله ای یا Stepper Motor ها هستند. این نوع موتور ها به دلیل تسلط بیشتر بر روی آنها نسبت به موتورهای DC از محبوبیت بیشتری در بین سازندگان رباتها برخوردارند.

در کل ، استپر موتور نوعی موتور مثل موتورهای DC است که حرکت دورانی تولید می کند. با این تفاوت که استپر موتورها دارای حرکت دقیق و حساب شده تری هستند .این موتورها به صورت درجه ای دوران می کنند و با درجه های مختلف در بازار موجود هستند .موتورهای پله ای موجود در بازار معمولا در دو نوع ۵ یا ۶ سیم یافت می شود.

موتور دیسک سخت یک نمونه موتور پله‌ای است .کاربرد اصلی این موتورها در کنترل موقعیت است .این موتورها ساختار کنترلی ساده‌ای دارند. لذا در ساخت ربات کاربرد زیادی دارند. بطوریکه به تعداد پالسهایی که به یکی از پایه‌های راه انداز آن ارسال می‌شود موتور به چپ یا راست می‌چرخد.

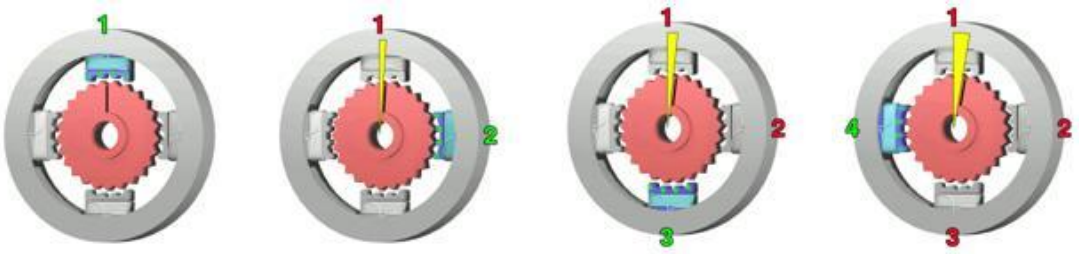
استفاده از موتور پله‌ای مشکلاتی از جمله وزن زیاد، قیمت بالا و قدرت بسیار کم را بدنبال دارد .

5-2-1 اصول کار موتور پله ای

واژه پله به معنی چرخش به اندازه درجه تعریف شده موتور است .مثلاً موتور پله‌ای با درجه ۱.۸ باید ۲۰۰ پله حرکت کند تا ۳۶۰ درجه یا یک دور کامل بچرخ : ۲۰۰X۱.۸= ۳۶۰

یک استپر موتور با درجه ۱۵ فقط باید ۲۴ پله برای یک دور کامل انجام دهد : ۱۵X۲۴=۳۶۰

به این ترتیب هرچه تعداد پله‌های یک موتور بیشتر باشد دقت چرخش آن افزایش می‌یابد .مکانیسم کنترلی موتور پله ای طوریست که امکان کنترل سرعت به سادگی میسر می شود. در شکل زیر نحوه کار یک موتور پله ای را مشاهده می کنید.



شکل 5-2 مراحل کار موتور پله ای

این هم تصویری از یک نوع موتور پله ای موجود.

شکل 5-3 شمای کلی شبکه سلولی

5-3 سروُ موتورها (Servo Motor)

نوع دیگری از موتورها که اخیرا در رباتیک استفاده میشود، سروُ موتورها هستند. این نوع موتورها نوع خاصی از موتورهای DC گیربکس دار هستند که به ربات این امکان را می دهند که در موقعیت خاصی، به اندازه زاویه مشخصی حرکت کرده و یا در لحظه ی مورد نظر متوقف شود. نمونه ای از سرو موتورها را در شکل مشاهده می نمایید.

شکل 5-4- سروُ موتور (Servo Motor)

5-4 درایور L298

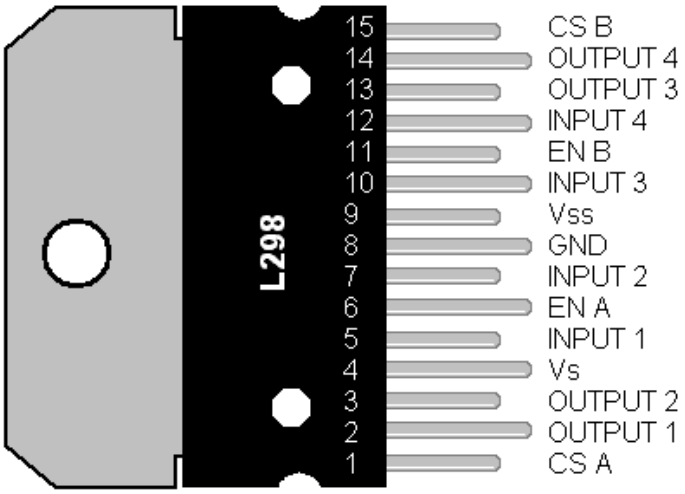
برای راه اندازی بسیاری از قطعات مانند موتورهای الکتریکی پرتوان، پمپ آب و...،معمولاً جریان خروجی المان های الکترونیکی نا کافی بوده و نیاز به تقویت جریان دارد. آی سی L298 برای راه اندازی موتورها بکار می رود که تصویر آن را در زیر مشاهده می نمایید.

شکل 5-5 – درایور L298

همان طور که در شکل می بینید، یک قطعه فلز در پشت این IC تعبیه شده تا از با انتقال گرما از IC به محیط، مانع گرم شدن بیش از حد IC شود. این قطعه Heat sink نام دارد. گاهی برای اطمینان بیشتر از یک Heat sink کمکی نیز استفاده می کنیم، به این صورت که Heat sink به وسیله ی پیچ بهHeat sink خود IC بسته می شود.

این IC یک پایه ی ورودیِ ولتاژ دارد که هر ولتاژی به این پایه وصل شود، مستقیماً به موتور یا هر المانی که به IC متصل شده باشد منتقل می شود. این پایه VPS نیز نام دارد (Variable Power Supply).

ترتیب پایه های این IC در شکل زیر توضیح داده شده است.



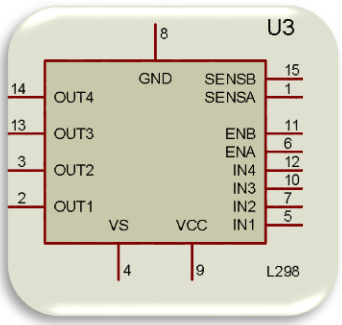
شکل 5-6 ترتیب پایه های L298

این آی سی 15 پایه می باشد. در زیر نحوه ی کار با این 15 پایه به صورت مختصر توضیح داده شده:

- پایه های 1 و 15: این پایه ها "Current sensing" نام دارند و باید هر 2 به – متصل شوند.
- پایه های 2 و 3: همان‌طور که می دانید این آی سی می‌تواند 2 موتور را همزمان و به صورت مستقل از یکدیگر راه‌اندازی و کنترل کند (2 موتور را A , B می‌نامیم). این 2 پایه باید به موتور A متصل شوند. (خروجی برای موتور A)
- پایه‌ی 4: این پایه باید به ولتاژ 5 ولت متصل شود.
- پایه های 5 و 7: این 2 پایه، ورودی برای کنترل موتور A هستند. این 2 پایه باید توسط کاربر یا مدار کنترل کننده‌ی ربات کنترل شوند.اگر این 2 پایه هر 2، 0 یا 1 منطقی باشند، موتور بدون حرکت می ایستد. اگر این 2 پایه به ترتیب 0 و 1 شوند، موتور به یک جهت مشخص می‌چرخد و اگر 1 و 0 شوند(یعنی ورودی برعکس شود)، موتور عکس جهت قبلی خواهد چرخید.
- پایه ی 6 و 11: این 2 پایه به ترتیب فعال ساز موتورهای A و B هستند. برای استفاده از هر 2 موتور باید هر 2 پایه 1 شوند.(برای فعال‌سازی هر موتور باید پایه‌ی مربوط به آن 1 شود).
- پایه ی 8: باید به – متصل شود.

- پایه ی 9: هر ولتاژی بر روی این پایه قرار گیرد برای راه اندازی موتورها استفاده می شود. مثلاً اگر موتورهای شما 12 ولت است، باید این پایه به 12 ولت متصل شود.
- پایه‌های 10 و 12: این 2 پایه، ورودی برای کنترل موتور B هستند. کار با این 2 پایه نیز مانند پایه‌های 5 و 7 (ورودی‌های موتور A) می‌باشد.
- پایه‌های 13 و 14: این 2 پایه باید به موتور B متصل شوند. (خروجی برای موتور B).

لازم به ذکر است که درایور L298 در نرم افزار پروتئوس با شمای زیر نمایش داده می شود:



شکل 5-7– شمای L298 در نرم افزار پروتئوس

5-5 راه اندازی موتور DC

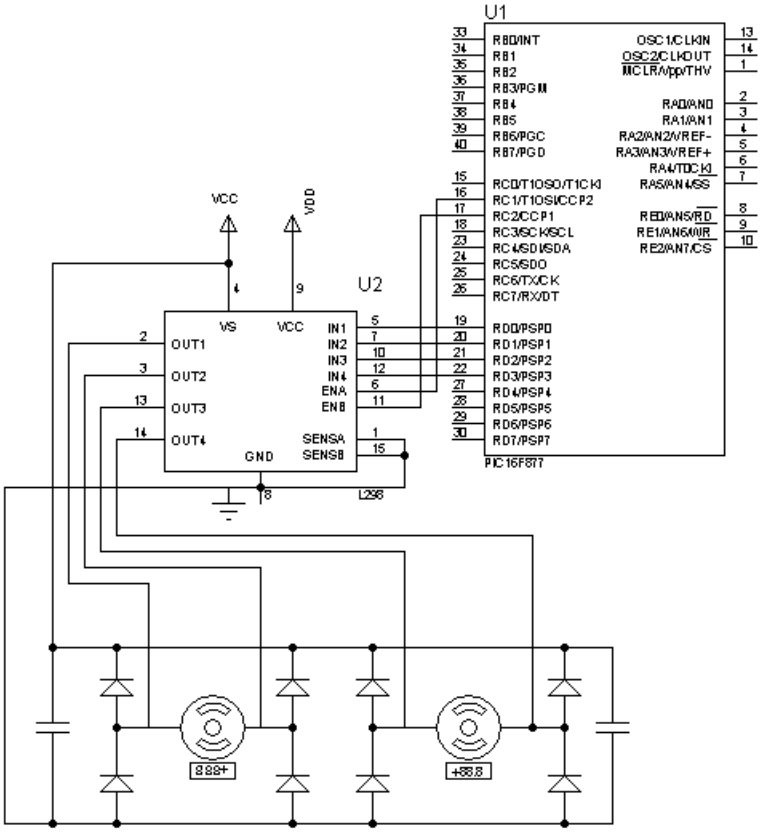
برای آشنایی با نحوه کار با موتورها در رباتیک، ابتدا نحوه اتصال و نوع کار با موتور DC را توضیح داده، سپس مثالی را با استفاده از نرم افزار پروتئوس انجام می دهیم تا بیشتر با نوع کار این نوع موتورها آشنا شویم.
موتورهای DC در نرم افزار پروتئوس با شمای زیر نمایش داده می شوند:

شکل 5-8– شمای موتورDC در نرم افزار پروتئوس

اگر ما دو سر یک موتور DC را به دو سر یک باطری متصل کنیم، موتور با آخرین سرعت خود شروع به چرخیدن می کند. اما ما نیاز داریم که سرعت را تحت کنترل خود درآوریم. یعنی اگر یک موتور DC را در یک ربات استفاده کنیم، نیاز است در بعضی جاها سرعت موتور بیشتر و در بعضی جاها سرعت موتور کمتر باشد. برای اینکار ما از PWM استفاده می کنیم .که توضیح مربوط به آن را در قسمت بعد ملاحظه می فرمایید. اما در اینجا ابتدا نحوه

اتصال پایه های موتور DC به L298 و میکرو را مشاهده کرده و پس از آن توضیحات مربوط به چگونگی کار را مشاهده خواهیم کرد.

ابتدا توسط نرم افزار پروتئوس مداری مطابق مدار موجود در تصویر 9 را ایجاد می کنیم.



شکل 5-9– شمای اتصال موتور DC به L298 و میکروکنترلر

- توجه کنید که در اطراف هر موتور DC از چهار عدد یکسوساز 1N4001 و دو عدد خازن 1nF استفاده کرده ایم. که اینکار بخاطر کم کردن فشار حاصل از نویز ایجاد شده توسط میدان الکتریکی موتورها می باشد که بر روی میکروکنترلر اثری نامطلوب ایجاد می کند.

- همانگونه که در تصویر نیز مشاهده می نمایید، هر درایور L298 می تواند همزمان دو موتور را ساپورت کند. که در این تصویر، خروجی های OUT1 و OUT2 برای موتور سمت چپ و خروجی های OUT3 و OUT4 برای موتور سمت راست مورد استفاده قرار گرفته است.

- پایه های IN1 و IN2 ورودی های موتور سمت چپ هستند. توجه کنید که اگر IN1 برابر با 1 و IN2 برابر صفر باشد، موتور در جهت راست و اگر IN1 برابر با صفر و IN2 برابر با یک باشد، موتور در جهت چپ گردش می کند. و اگر مقدار این دو پایه با هم برابر باشد، موتور بی حرکت می ماند. پایه های IN3 و IN4 نیز همین وضعیت را داشته و بعنوان ورودی های موتور سمت راست استفاده شده اند.

- پایه های SENSA و SENSB ، Current sensing هستند. به این معنی که میزان جریان برگشتی از موتورها را حساب می کنند. به عنوان مثال یک روبات تعقیب خط را در نظر بگیرید. این روبات هنگام حرکت در سرپیچ ها، طبیعتا یکی از موتورها فشار بیشتری را متحمل می شود که همین موضوع باعث می شود که روبات نتواند مسیر مستقیم را طی کند و منحرف می شود. اما اگر ما در طراحی روبات از دو پایه SENSA و SENSB استفاده کنیم، قادر خواهیم بود در هر لحظه، فشار وارد بر هر موتور را تشخیص داده و توسط برنامه ای که به میکرو داده ایم، سرعت چرخش هر موتور را در هر لحظه تعیین کنیم و مانع از انحراف روبات از مسیر مورد نظر شویم.

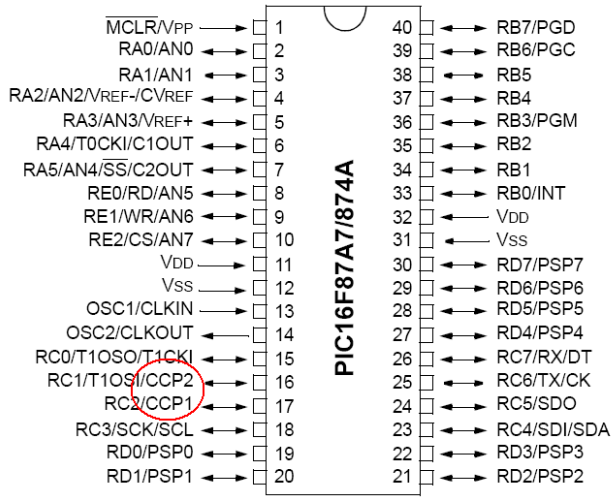
6-5) PWM (Pulse With Modulation):

برای کنترل کردن دور موتورهای DC می شود از میکروکنترلر استفاده کرد. برای این که میکروکنترلر بتواند دور موتور را کنترل کند، دو روش وجود دارد. یکی اینکه ولتاژ موتور را کم و زیاد کند و دیگری به روش پالس دهی می باشد. برای اینکار از روش PWM (مدولاسیون عرض پالس) استفاده می کنیم. در این روش فرکانس ثابت است و از تغییر نسبت صفر به یک استفاده می شود که در اصطلاح به آن Duty-Cycle گفته می شود.

در این روش ما یک موج PWM را بر روی پایه های موتور اعمال می کنیم. هنگامی که سطح ورودی پایه ها، یک است، موتور روشن، و هنگامی که سطح ورودی پایه ها صفر است، موتور خاموش است.

با تغییر نسبت زمان صفر یا یک بودن پالس (همان زمان روشن یا خاموش بودن موتور)، سرعت موتور کم و یا زیاد می شود. که تعیین نسبت صفر و یک بودن پالس ارسالی، بر عهده میکروکنترلر است.

برای این کار میکرو از رجیسترهای CCP1CON,CCP2CON و T2CON استفاده می کند.

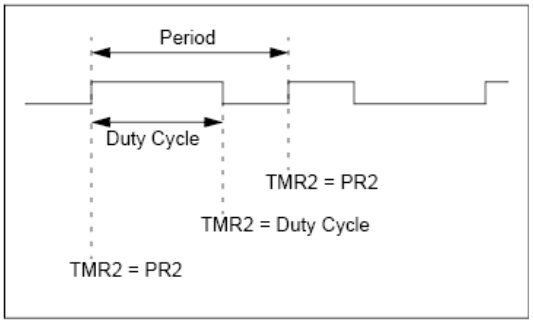


شکل 5-10– پایه های مربوط به PWM در میکروکنترلر

لازم به ذکر است که میکرو پالس PWM ایجاد شده را بر روی پایه های CCP1 و CCP2 (یعنی پایه های 16 و 17 میکرو کنترلر ارسال می کند.)

5-6-1 تنظیم PWM:

گفتیم برای تنظیم PWM از Duty-Cycle استفاده میکینم. همانطور که در شکل زیر مشاهده می نمایید، هر موج، Period مخصوص به خود را دارد. حال با تنظیم ، طول Duty-Cycle می توانیم مدت زمان یک بودن سطح پالس را تعیین کینم. به این نحو که هر چه طول Duty-Cycle بیشتر باشد، مدت زمان روشن بودن موتور بیشتر بوده و در نتیجه سرعت موتور بیشتر خواهد بود.



شکل 5-11– Duty-Cycle و پالس

برای تنظیم Duty-Cycle مراحل زیر را دنبال کنید:

- میزان پریود PWM را با نوشتن مقدار آن در رجیستر PR2 تعیین کنید.هر میکرو می تواند همزمان دو موج PWM تولید کند که مقدار PR2 برای هر دو PWM مشترک است.

  توجه کنید که PR2 فرکانس PWM را تنظیم می کند که در صورتی که بخواهیم فرکانس تغییر کند، باید مقدار PR2 را تغییر دهیم. که فرمول آن به شرح زیر است:

  PWM period=[PR2+1]*4*Tosc*TMR2 prescaler value

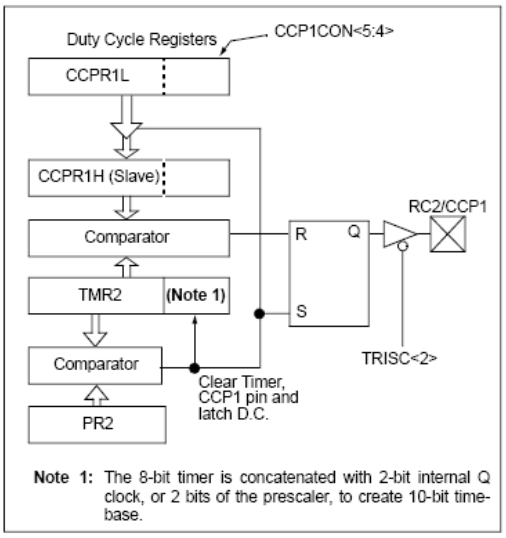
  که Tosc نشان دهنده مقدار اوسیلاتور است.
- میزان Duty-Cycle را با نوشتن مقدار آن در رجیستر CCPR1L و بیت های چهارم و پنجم CCP1CON تعیین کنید. و فرمول آن عبارت است از:

(TMR2 prescale value)*TOSC *PWM duty cycle =(CCPR1L:CCP1CON<5:4>)

- پایه CCP1 یا CCP2 میکروکنترلر را با استفاده از TRISC<2..3> به عنوان خروجی تعیین کنید.
- مقدار TMR2را تعیین کرده و تایمر 2 را با استفاده از رجیستر T2CON فعال کنید.

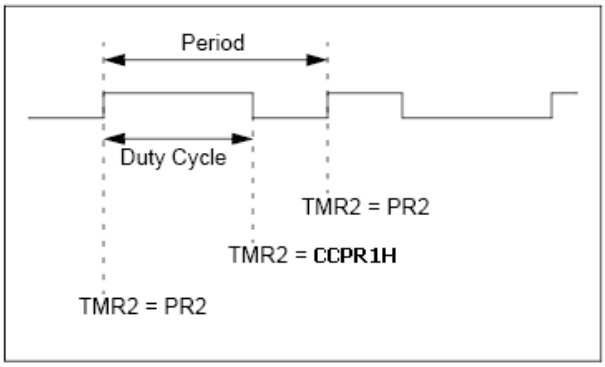
5-7 اساس کار PWM :

ساختمان داخلی میکروکنترلر برای تولید موج PWM از دو مقایسه کننده و یک فیلیپ فلاپ نوع SR تشکیل شده است که این سه عنصر موج PWM را تولید کرده و بر روی پایه های CCP1 و CCP2 میکرو قرار میدهند.
همانگونه که در شکل می بینید پایه S فلیپ فلاپ که set نام دارد، به مقایسه کننده بین TRM2 و PR2 متصل است و پایه R که Reset نام دارد به مقایسه کننده بین TMR2 و CCPR1H متصل است.
با توجه به شکل بالا، در آغاز کار به محض اینکه مقدار تایمر دو با مقدار PR2 برابر شد، Comparator پایینی



شکل 5-12– ساختار داخلی PWM

یک پالس تولید کرده که این پالس سه کار انجام میدهد.
اول مقدار تایمر دو را صفر می کند، سپس به پایه S فلیپ فلاپ رفته و پایه CCP1 را برابر یک (Set) می کند. و در پایان مقدار CCPR1L را در CCPR1H کپی می کند. (دوباره یاد آور می شوم که مقدار CCPR1L ، مشخص کننده اندازه Duty-Cycle است.)
در مرحله بعد، Comparator بالایی، TMR2 و CCPR1H را مقایسه کرده و به محض تساوی مقدار آندو، پالسی را ایجاد کرده که این پالس به پایه R فلیپ فلاپ اعمال شده و پایه CCP1 میکرو را صفر (Reset) می کند.
به این ترتیب Comparator پایینی CCP1 را یک کرده و Comparator بالایی CCP1 را صفر می کند. و موجی شبیه به موج زیر تولید می کند:



شکل 5-13– تولید موج PWM

5-8 کد نویسی مربوط به موتور DC

حال که با نحوه کار موتور DC آشنا شدیم بهتر است به تشریح کد مربوط به این پروژه بپردازیم. برای اینکار ابتدا نرم افزار MPLAB را باز کرده و یک پروژه جدید با نام dcmotor ایجاد کرده و شروع به نوشتن کد زیر در داخل برنامه می کنید.

```
1.  #include <pic.h>
2.      __CONFIG(0x03F72);                  // PIC16F877A Configuration Fuses:
3.                                          //  - XT Oscillator
4.                                          //  - 70 msecs Power Up Timer On
5.                                          //  - Watchdog Timer Off
6.                                          //  - Code Protection Off
7.
8.  static bit TrisL_motor_i1 @ (unsigned) &TRISD*8;     // LED Physical Bits
9.  static bit L_motor_i1 @ (unsigned) &PORTD*8;
10. static bit TrisL_motor_i2 @ (unsigned) &TRISD*8+1;   // LED Physical Bits
11. static bit L_motor_i2 @ (unsigned) &PORTD*8+1;
12. static bit TrisR_motor_i1 @ (unsigned) &TRISD*8+2;   // LED Physical Bits
13. static bit R_motor_i1 @ (unsigned) &PORTD*8+2;
14. static bit TrisR_motor_i2 @ (unsigned) &TRISD*8+3;   // LED Physical Bits
15. static bit R_motor_i2 @ (unsigned) &PORTD*8+3;
16. int R_motorSpeed=200;
17. int L_motorSpeed=255;
18.
19. void main(void)                          // Template Mainline
20. {
21.     OPTION = 0x0D1;                     // Assign Prescaler to TMR0
```

```
22.                                          // Prescaler is /4
23.     CCPR1L = 0;                          // 0% Duty Cycle
24.     CCP1CON = 0b000001111;               // Turn on PWM Mode
25.     CCPR2L = 0;                          // 0% Duty Cycle
26.     CCP2CON = 0b000001111;               // Turn on PWM Mode
27.     PR2 = 255;                           // 26 usec Cycle for 38 KHz
28.     TMR2 = 0;                            // Reset TMR2
29.     T2CON = 0b000000100;                 // TMR2 is On, Scalers 1:1
30.
31.     TRISC1=0;
32.     TRISC2=0;
33.     TrisR_motor_i1=0;
34.     TrisR_motor_i2=0;
35.     TrisL_motor_i1=0;
36.     TrisL_motor_i2=0;
37. ////////Forward left & right motor//////
38.     R_motor_i1=L_motor_i1=1;
39.     R_motor_i2=L_motor_i2=0;
40. ////////////////////////////////
41.     CCPR1L = R_motorSpeed;
42.     CCPR2L = L_motorSpeed;
43. ////////////////////////////////
44.     while (1)
45.     {
46.      }
47. }
```

کد مربوط به موتورهای DC

حال به شرح کد می پردازیم.

- دستورات خط هشتم تا شانزدهم، عبارتی را جایگزین یکی از رجیسترهای میکرو می کند. مزیت این کار در این است که هنگام نوشتن برنامه بر روی پورت های میکرو، برنامه نویس علاوه بر اینکه در هر لحظه می داند که هر پورت چه کاری را قرار است انجام دهد، این امکان را نیز دارد که در صورتی که نیاز داشت وظیفه ی یکی از پورت ها را تغییر دهد، این کار به آسانی صورت گیرد و دیگر نیازی نباشد تا در کل برنامه تک تک قسمت هایی را که از این پورت استفاده کرده است را دوباره تغییر دهد. به عنوان مثال، خط هشتم، عبارت TrisL_motor_i1 را جایگزین TRISD0 می کند و الخ...
- در خط 23 و 25، ابتدا مقدار CCPR1L و CCPR2L را مساوی صفر قرار داده تا در داخل برنامه آنها را مقدار دهی کنیم.

- توسط خطوط 24 و 26، مُد PWM را فعال می کنیم. که این کار توسط مقداردهی ثبات های CCP1CON و CCP2CON صورت می گیرد. برای این کار مقدار این ثبات را برابر 0b000001111 قرار می دهیم.
- در خط 27، مقدار PR2 را برابر 255 قرار می دهیم. همانگونه که گفتیم PR2 نشان دهنده طول موج PWM است که می تواند از 0 تا 255 مقدار دهی شود.
- در خط 28 مقدار شمارنده تایمر دو را مساوی صفر قرار داده و در خط 29، تایمر دو را فعال می کنیم.
- توسط خطوط 31 و 32، پایه های 16 و 17 میکرو را به عنوان خروجی تعیین می کنیم.تا از آنها برای ارسال موج PWM استفاده کنیم.
- توسط خطوط 33 تا 36، پایه های D0 تا D3 میکروکنترلر را نیز به عنوان خروجی تعیین می کنیم.
- توسط خطوط 38 و 39، جهت چرخش هر موتور را تعیین می کنیم.
- توسط خطوط 41 و 42 نیز مقدار Duty-Cycle را مشخص می کنیم.

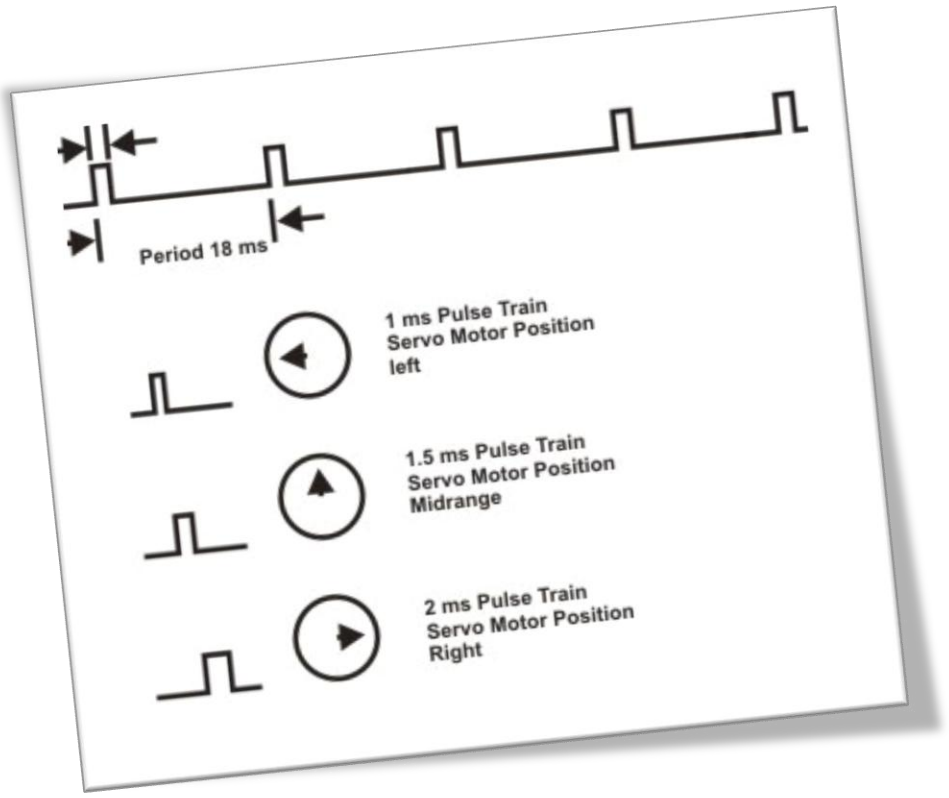
با اجرای برنامه مشاهده خواهیم کرد که هر دو موتور DC در یک جهت شروع به چرخش می کنند. برای تغییر در جهت چرخش موتورها، خطوط 38 و 39 را تغییر دهید و برای تغییر در سرعت موتورها می توانید خطوط 41 و 42 را تغییر داده تا Duty-Cycle تغییر کند.

5-9 راه اندازی موتور Servo

موتورهای سروُ در نرم افزار پروتئوس با شمای زیر نمایش داده می شوند.

شکل 5-14– شمای موتور Servo در نرم افزار پروتئوس

این سروُ موتورها با PWM کار می کنند و دارای سه پایه هستند. پایه اول به 5+ ولت وصل می شود. پایه سوم به زمین متصل می گردد و پایه وسطی به پایه تولید موج PWM میکروکنترلر متصل می گردد.
نحوه چرخش این نوع موتور بدینگونه است که اگر طول Duty-Cycle کمتر از یک میکروثانیه باشد، موتور به چپ می چرخد. اگر طول آن بین یک و دو میکرو ثانیه باشد، موتور در حالت Center قرار می گیرد و اگر طول Duty-Cycle بیشتر از دو میکرو ثانیه باشد، موتور به راست می چرخد.



شکل 5-15– طول پالس ها و اثر آن بر حرکت موتورهای Servo

کد نویسی مربوط به سروُ موتور نیز تقریبا مشابه موتورهای DC است به همین خاطر از نوشتن دوباره آن خودداری می کنیم.

5-10 رجیسترها

5-10-1 :رجیستر CCP1CON / رجیستر CCP2CON

(آدرس 17H/1DH)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7							bit 0

بیت 7-6 :

اجرا نشده : در صورتی که صفر باشد خوانده میشود.

بیت5- 4 CCPxX:CCPxY :

بیت های کم اهمیت PWM

روش تصرف:استفاده نشده

روش مقایسه:استفاده نشده

روش PWM:این بیتها 2تا از کم اهمیت ترین بیتهای دوره کار PWM هستند.

8بیت با اهمیت تر در CCPRxL یافت میشوند.

بیت 3-0 CCPxM3:CCPxM0 :

بیت های انتخاب حالت CCPx

0000 = غیر فعال کردن تصرف/مقایسه/PWM (راه اندازی مجدد ماژول CCPx)

0100 = حالت تصرف ، همه لبه ها پایین رونده

0101 = حالت تصرف ، همه لبه ها بالا رونده

0110 = حالت تصرف ، همه لبه ها پایین رونده 4بیتی

0111 = حالت تصرف ، همه لبه ها پایین رونده 16بیتی

1000 = حالت مقایسه ، تنظیم(یک کردن) خروجی با مطابقت دادن (بیت CCPxIF یک است)

1001 = حالت مقایسه ، پاک کردن(صفر کردن) خروجی با مطابقت دادن (بیت CCPxIF یک است)

1010 = حالت مقایسه ، ایجاد وقفه نرم افزاری با مطابقت دادن (بیت CCPxIF یک اســت،پین CCPx تحت تاثیر قرار نمی گیرد)

1011 = حالت مقایسه ، ایجاد رویدادی خاص (بیت CCPxIF یک است،پین CCPx تحت تــاثیر قــرار نمی گیرد)

CCP1، تایمر (TMR1) را راه اندازی مجدد می کند

CCP2، تایمر (TMR1) را راه اندازی مجدد می کند ویک تبدیل آنالوگ به دیجیتال را شروع مـی کنـد (اگر ما ژول تبدیل آنالوگ به دیجیتال را در اختیار داشته باشد)

11xx = حالت PWM

2-10-5 T2CON : کنترل رجیستر TIMER2 (آدرس 12H)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

بیت 7 :

اجرا نشده : در صورتی که صفر باشد خوانده میشود.

بیت 6-3 TOUTPS3 :TOUTPS0 :

بیت های انتخاب مقیاس بعدی خروجی تایمر2

0000 = مقیاس بعدی 1:1

0001 = مقیاس بعدی 1:2

0010 = مقیاس بعدی 1:3

.

.

1111 = مقیاس بعدی 1:16

بیت 2 : TMR2ON:

بیت روشن کردن تایمر 2

1 = تایمر2 روشن است

0 = تایمر2 خاموش است

بیت 1-0 T2CKPS1:T2CKPS0:

بیت های انتخاب مقیاس قبلی ساعت تایمر2

00 = مقیاس قبلی 1 است

01 = مقیاس قبلی 4 است

1x= مقیاس قبلی 16 است

فهرست علایم :

R =بیت خواندنی

W =بیت نوشتنی

U = بیت اجرا نشدنی ، در صورتی که صفر باشد خوانده میشود

N =مقدار POR

'1'= بیت تنظیم شده

'0'=بیت پاک شده

X=بیت تعریف نشده

فصل ششم

6. ارتباط با کامپیوتر

در بسیاری از کاربردها تبدیلات آنالوگ به دیجیتال نیاز است امروزه مبدل های آنالوگ به دیجیتال (ADC)غالباً به شکل یک مدار مجتمع یافت میشوند.

مبدل های آنالوگ به دیجیتال (ADC) ولتاژهای آنالوگ را به رمزهای دیجیتال تبدیل می کنند.

در این فصل در مورد تولید اطلاعات دیجیتال بوسیله مبدل های آنالوگ به دیجیتال (ADC) شرح مختصری داده می شود .

7. پیکربندی PORTA و TRISA:

PORTA یک 6 بیتی عمومی و پورتی دوطرفه است. TRISA همان ثبات مسیر داده است.

برابر یک قرار دادن TRISA ، پین PORTA را به عنوان یک ورودی قرار خواهد داد (یعنی خروجی را در حالت امپدانس بالا قرار می دهد)

صفرکردن یک بیت TRISA پین PORTA را به عنوان یک خروجی قرار خواهد داد (یعنی محتویات خروجی را در پین انتخاب شده قرار می دهد).

عمل خواندن PORTA ،حالات پین ها را می خواند درحالیکه عمل نوشتن PORTA ،در پورت می نویسد.

تمام عملیات نوشتن در پورت، شامل عملیات (خواندن _تغییر کردن – نوشتن) هستند.

بنابراین نوشتن در یک پورت به طور ضمنی شامل خواندن پین های پورت ،اصلاح کردن مقدار آن و سپس نوشتن در یک پورت داده می باشد.

پین RA 4 برای اینکه مناسب با پین RA4/T0CKI باشد ، با ورودی ساعت ماژول TIMER0 تسهیم شده است.

پین RA4/T0CKI یک ورودی رهاساز اشمیت و یک خروجی جریان باز است .

بقیه پین های PORTA دارای سطوح ورودی TTL(Transistor-Transistor Logic)و تمام قطعات خروجی CMOSمیباشند.

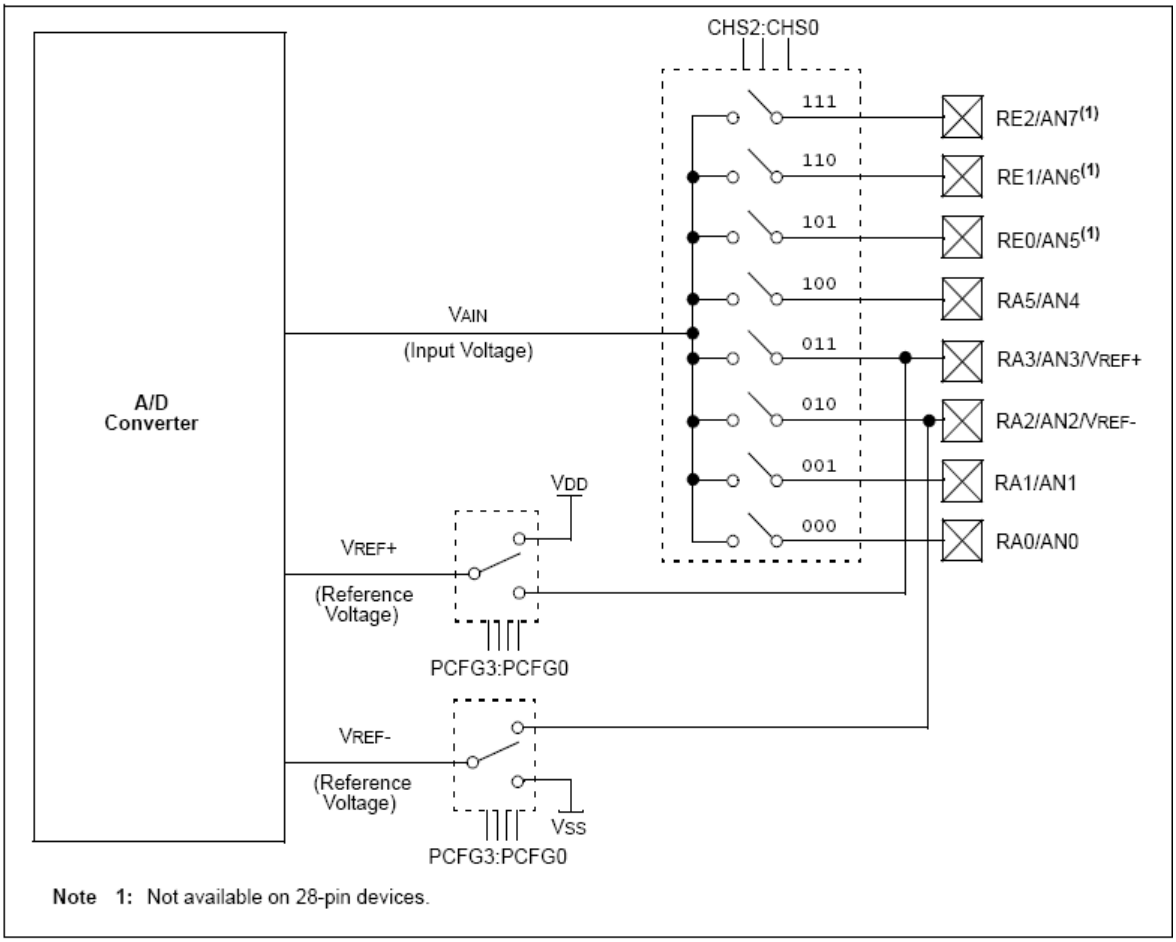
دیگر پین های PORTA با ورودی های آنالوگ و ورودی های VREF آنالوگ، برای تبدیل کننده های آنالوگ به دیجیتال و مقایسه کننده ها ،تسهیم شده اند.

عملیات هریک از پین ها بوسیله یک یا صفر کردن بیت های کنترلی مناسب درثباتهای ADCON1 و یا CMCON انتخاب می شود.

توجه:دریک راه اندازی مجدد (روشن شدن) این پین ها به عنوان ورودی های آنالوگ پیکربندی می شوند و با صفر خوانده می شوند.مقایسه کننده ها در حالت خاموش (دیجیتال)هستند.

رجیستر TRISA مسیر پین های پورت را کنترل می کند، حتی زمانیکه که آنها به عنوان ورودی های آنالوگ استفاده شوند،کاربر باید مطمئن باشد که بیت های رجیستر TRISA راتنظیم کرده است، هنگامیکه که از آنها به عنوان ورودی های آنالوگ استفاده می کند.

FIGURE 11-1: A/D BLOCK DIAGRAM



شکل 7-1– نمودار بلوک تبدیل آنالوگ به دیجیتال

7-1 :رجیستر ADCON0(آدرس 1FH)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit 7							bit 0

بیت7-6 ADCS0:ADCS1 :

بیت های انتخاب ساعت در تبدیل آنالوگ به دیجیتال

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

بیت 5-3 CHS0:CHS2 :

بیت های انتخاب کانال آنالوگ

000= کانال 0 (AN0)

001= کانال 1 (AN1)

010= کانال 2 (AN2)

011= کانال 3 (AN3)

100= کانال 4 (AN4)

101= کانال 5 (AN5)

110= کانال 6 (AN6)

111= کانال 7 (AN7)

توجه : قطعات PIC 16F873A/876A فقط کانال های A/D مابین 0 تا 4 را به کار می برنـد ، کانـال های استفاده نشده کنار گذاشته می شوند ، کانال های استفاده نشده را با این قطعات انتخاب نکنید.

بیت 2 – GO/DONE :

بیت وضعیت در تبدیل آنالوگ به دیجیتال

- **هنگامیکه ADON=1 :**

1 = تبدیل آنالوگ به دیجیتال شروع شده است (یک کردن این بیت، تبدیل آنالوگ به دیجیتال را شروع می کند که به طور خودکار هنگامیکه تبدیل آنالوگ به دیجیتال کامل شود ،به طور سخت افزاری صفر میشود.
0= تبدیل آنالوگ به دیجیتال شروع نشده است.

بیت 1 :

استفاده نشده : با صفر خوانده میشود.

بیت 0 – ADON :

1= روشن کردن ماژول تبدیل آنالوگ به دیجیتال
0= خاموش کردن ماژول تبدیل آنالوگ به دیجیتال و استفاده نکردن جریان عملیاتی

7-2 رجیستر ADCON1(آدرس 9FH)

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

بیت 7– ADFM:

بیت انتخاب روش قالب بندی در تبدیل آنالوگ به دیجیتال
1= همتراز شده از راست ،6بیت پرارزش از آدرس، با صفر خوانده می شوند.
0=همتراز شده از چپ،6بیت کم ارزش از آدرس، با صفر خوانده می شوند.

بیت 6– ADCS2:

بیت انتخاب ساعت در تبدیل آنالوگ به دیجیتال

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

بیت 4-5 :

استفاده نشده : با صفر خوانده میشود.

بیت 0-3 PCFG3:PCGF0 :

بیت های کنترل پیکربندی پورت در تبدیل آنالوگ به دیجیتال

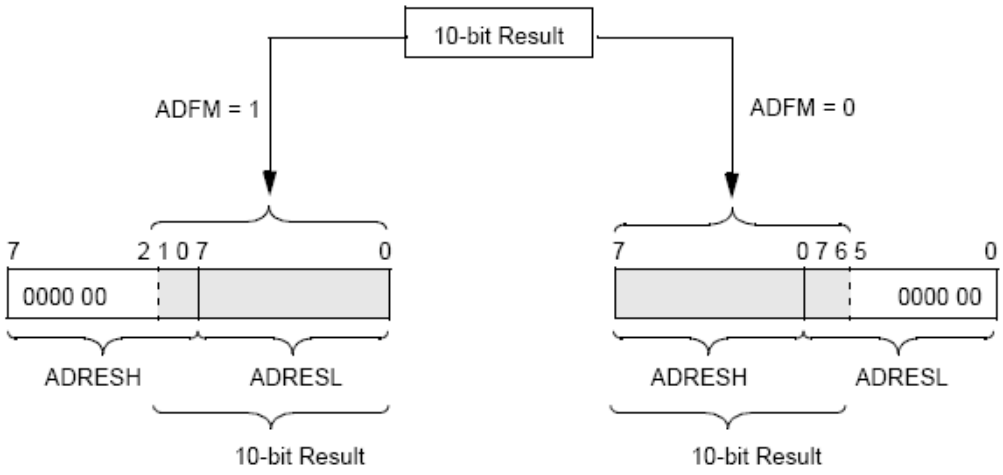
PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C / R
0000	A	A	A	A	A	A	A	A	VDD	Vss	8 / 0
0001	A	A	A	A	VREF+	A	A	A	AN3	Vss	7 / 1
0010	D	D	D	A	A	A	A	A	VDD	Vss	5 / 0
0011	D	D	D	A	VREF+	A	A	A	AN3	Vss	4 / 1
0100	D	D	D	D	A	D	A	A	VDD	Vss	3 / 0
0101	D	D	D	D	VREF+	D	A	A	AN3	Vss	2 / 1
011x	D	D	D	D	D	D	D	D	—	—	0 / 0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6 / 2
1001	D	D	A	A	A	A	A	A	VDD	Vss	6 / 0
1010	D	D	A	A	VREF+	A	A	A	AN3	Vss	5 / 1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4 / 2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3 / 2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2 / 2
1110	D	D	D	D	D	D	D	A	VDD	Vss	1 / 0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1 / 2

A = ورودی آنالوگ

D= ورودی-خروجی دیجیتال

C/R=کانال های ورودی آنالوگ/ منبع ولتاژ در تبدیل آنالوگ به دیجیتال

شکل 7-2-: نتیجه همتراز کردن در تبدیل آنالوگ به دیجیتال



همتراز شده از چپ **همتراز شده از راست**

7-3 ADRESH:ADRESL:

ثبت کردن محتوی 10 بیت از نتیجه تبدیل آنالوگ به دیجیتال،هنگامیکه تبدیل آنالوگ به دیجیتال کامل شد.نتیجه به این جفت رجیستر A/Dبار میشود.

بیت (<ADCON0<2> GO/DONE(صفر میشود و بیت پرچم وقفه در تبدیل آنالوگ به دیجیتال(ADIF) یک میشود.

نمودار بلوک ماژول تبدیل آنالوگ به دیجیتال در شکل 1-11 نشان داده شده است.

بعد از اینکه ماژول تبدیل آنالوگ به دیجیتال به طور مناسب پیکربندی شد،کانال انتخاب شده قبل از اینکه تبدیل آنالوگ به دیجیتال شروع شود ،باید تعیین شود.

کانال های ورودی آنالوگ باید با بیت های TRIS که به عنوان ورودی انتخاب شده اند، مطابقت داشته باشد.

برای تعیین واحد زمان قسمت 1-11 را نگاه کنید.

بعد از اینکه زمان هدف یابی تمام شد ، تبدیل آنالوگ به دیجیتال میتواند شروع شود.

این مراحل برای انجام یک تبدیل آنالوگ به دیجیتال باید دنبال شود:

1- ایجاد پیکربندی در ماژول تبدیل آنالوگ به دیجیتال:

- پیکربندی پین های آنالوگ/منابع ولتاژ و ورودی-خروجی دیجیتال(ADCON1)

- انتخاب کانال ورودی آنالوگ به دیجیتال (ADCON0)
-انتخاب ساعت تبدیل آنالوگ به دیجیتال (ADCON0)
-روشن کردن ماژول تبدیل آنالوگ به دیجیتال (ADCON0)

2- پیکربندی وقفه های تبدیل آنالوگ به دیجیتال (در صورت مطلوب بودن)
- صفر کردن بیت ADIF
- یک کردن بیت ADIE
- یک کردن بیت PEIE
- یک کردن بیت GIE

3-انتظار برای زمان استفاده لازم

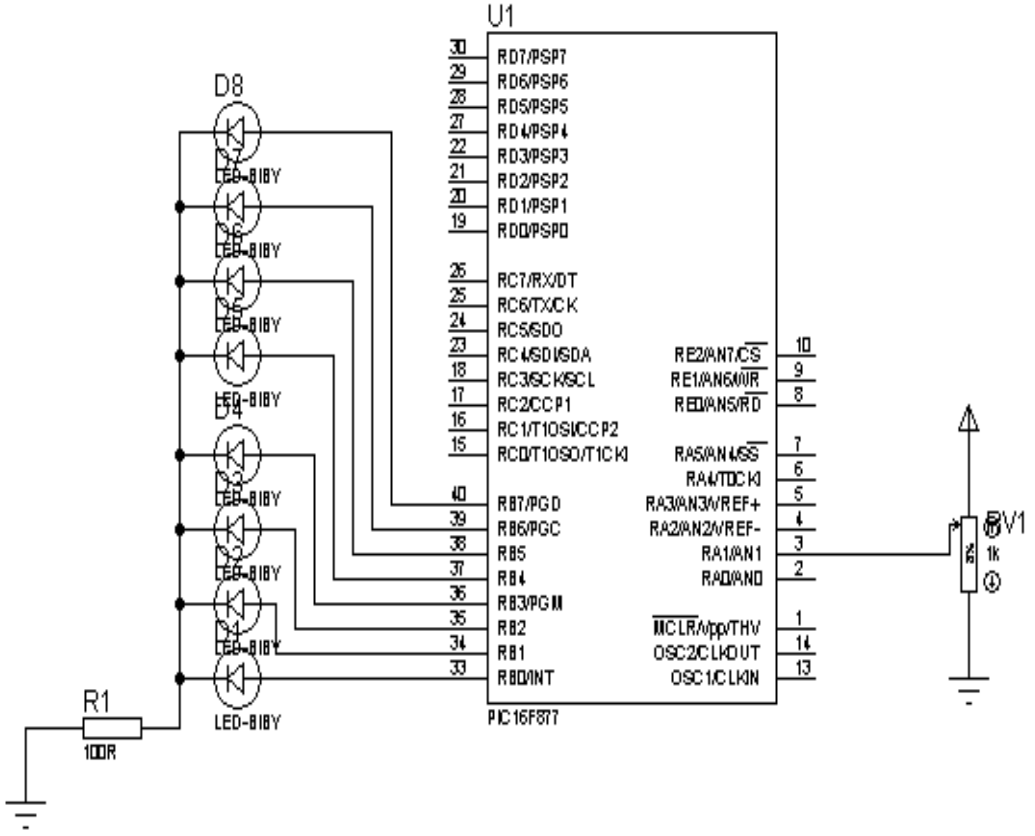
4-شروع کردن تبدیل آنالوگ به دیجیتال
- یک کردن بیت GO/DONE(ADCON0)

5- انتظار برای کامل شدن تبدیل آنالوگ به دیجیتال ،بوسیله هریک از این دو مورد:
-نمونه برداری از بیت GO/DONE برای اینکه صفر شود(بوسیله وقفه ها) یا
- انتظار برای وقفه تبدیل آنالوگ به دیجیتال

6- خواندن نتیجه جفت رجیستر در تبدیل آنالوگ به دیجیتال (ADRESH:ADRESL)
و صفر کردن بیت ADIF اگر لازم باشد.

7- برای تبدیل بعدی اگر لازم بود به مرحله 1 یا 2 بروید.
زمان تبدیل آنالوگ به دیجیتال برای هر بیت به عنوان T_{AD} تعریف شده است.

7-4 مدار تبدیل کننده آنالوگ به دیجیتال در PROTEUS:



شکل 7-3– شماتیک مدار تبدیل کننده آنالوگ به دیجیتال

7-5برنامه تبدیل آنالوگ به دیجیتال :

```
unsigned int adc_read(unsigned char channel)
{
unsigned int result;
ADCON=(channel << 3) + 0X80 ;
//wait for required acquisition time. ( 20 micro sec.)
GODONE =  1 ;
while(GODONE); // wait for conversion complete
result=(ADRESH << 8) + ADRESL;
return result;
}
```

فصل هفتم

8. نمایشگر (LCD)

LCD ها می توانند به برنامه کاربردی شما برحسب مهیا سازی رابط (واسط) مفید برای کاربر موارد زیادی را اضافه کنند .خطا گیری یک برنامه یا فقط ارائه آن یک نگرش فنی تلقی می شود . مرسومترین نمونه ی کنترل کننده ی LCD هیتاچی 44780 است که یک واسط نسبتاً ساده بین پردازنده و یک LCD را مهیا می سازد. استفاده از این رابط اغلب از طریق طراحان و برنامه ریزان بی تجربه صورت نمی گیرد زیرا پیداکردن مستند سازی خوب بر روی رابط کاری مشکل است. آغاز مجدد رابط می تواند یک مشکل باشد و نمایشات آنها گران است. ما بر مبنای LCD ها برای مدتی با هیتاچی 44780 کار کرده ام و باید بگوییم که هیچ یک از این تصورات را قبول نداریم. LCD ها می توانند به سادگی به یک برنامه کاربری افزوده شوند و هزینه ی آنها اغلب از ابزارهای قدیمی تأ مین شده است یا در مازاد مغازه ها با ارزشی کمتر از 1 دلار یافت شده اند . هدف از این صفحه ارائه آموزش مختصری در مورد چگونگی ارتباط با هیتاچی44780 براساس LCD هاست. . ما سعی کرده ایم تمامی اطلاعات لازم برای افزوده سازی موفقیت آمیز LCD به برنامه کاربردی شما را فراهم کنیم. مرسومترین ارنباط دهنده ی به کا رفته برای 44780 براساس LCD ها ، 14pins در یک ردیف، با مراکز پین جداگانه 100 و 0 می باشد. این پین ها تحت جدول سیم کشی شده اند.

Pins	Description
1	Ground
2	Vcc
3	Contrast Voltage
4	"R/S" _Instruction/Register Select
5	"R/W" _Read/Write LCD Registers
6	"E" Clock
7 - 14	Data I/O Pins

جدول 1- پایه های LCD

. احتمالاً شما از این توصیف حدس خواهید زد که ارتباط دهنده یک باس موازی است که برای ما امکان خواندن و نوشتن سریع و آسان اطلاعات از LCD را فراهم می سازد. این فرم موجی ASC 11 بایت را برای صفحه ی LCD خواهد داشت. کدASC 11 که نمایش داده شده است 8 بایت درازا دارد و به LCD ، چهار یا هشت بایت در هر زمان ارسال کرده است. اگر مدل 4 بیت استفاده شده است ، دو nybble داده ها برای آرایش یک 8 بیت کامل انتقال داده شده اند. ساعت E برای آغاز انتقال اطلاعات در مسیر LCD مورد استفاده قرار گرفته است. ارسال

داده های موازی مثل 4 یا 8 بیت ها دو روش اولیه عملکرد می باشند. در حالیکه نگرشات و مدل های ثانویه ای موجود می باشد.

تصمیم گیری در مورد چگونگی ارسال اطلاعات به LCD مهمترین تصمیم اتخاذ شده برای برنامه کاربردی رابط LCD تلقی می شود.

زمانیکه سرعت در یک برنامه کاربردی مورد نیاز است و حداقل 10 (I/O) پین موجود می باشد روش هشت بیت بهترین روش به کار رفته می باشد. روش 4 بیت نیازمند مینیموم شش بیت است. جهت سیم کشی میکرو کنترل کننده برای یک LCD فقط رأس چهار بیت (7- DB4) نوشته شده است.
(R/S) بیت جهت گزینش انتقال اطلاعات یا یک دستورالعمل بین میکروکنترل کننده و LCD استفاده شده است. بیت تنظیم شده است ، پس بایت در موقعیت نشانگر کنونی LCD می تواند خوانده شود و یا نوشته شود. زمانی مجدداً بیت تنظیم می شود که یا یک دستورالعمل به LCD فرستاده می شود یا موقعیت های اجرایی کمترین دستورالعمل read back در نظر گرفته می شود.

در جدول زیر دستورالعمل های مختلف موجود برای استفاده با 44780 نشان داده شده است. توصیفات بیت برای دستورات مختلف به شرح زیر است .

R/S	R/W	D7	D6	D5	D4	D3	D2	D1	D0	Instruction/Description
4	5	14	13	12	11	10	9	8	7	Pins
0	0	0	0	0	0	0	0	0	1	Clear Display
0	0	0	0	0	0	0	0	1	*	Return Cursor and LCD to Home Position
0	0	0	0	0	0	0	1	ID	S	Set Cursor Move Direction
0	0	0	0	0	0	1	D	C	B	Enable Display/Cursor
0	0	0	0	0	1	SC	RL	*	*	Move Cursor/Shift Display
0	0	0	0	1	DL	N	F	*	*	Set Interface Length
0	0	0	1	A	A	A	A	A	A	Move Cursor into CGRAM
0	0	1	A	A	A	A	A	A	A	Move Cursor to Display
0	1	BF	*	*	*	*	*	*	*	Poll the "Busy Flag"
1	0	D	D	D	D	D	D	D	D	Write a Character to the Display at the Current Cursor Position
1	1	D	D	D	D	D	D	D	D	Read the Character on the Display at the Current Cursor Position

جدول 2 – ارسال اطلاعات به پایه های LCD

توضیحات بیت مربوط به دستورات مختلف:

Set Cursor Move Direction

تنضیم جهت حرکت کرسر:

ID : اگر ست شود هر بار مکانمای موس را بعد از هر بایتی که نوشته می شود یک واحد به موقعیتس اضافه میکند

S : زمانی که یک بایت نوشته شود ، موقعیت مکان نما به اندازه ی نوشتن یک بایت جابجا می شود.

Enable Display/Cursor:

D : اگر یک باشد صفحه نمایشگر روشن و اگر صفر باشد صفه نمایش خاموش است.

C: اگر یک باشد کرسر روشن و اگر صفر باشد کرسر خاموش است

B : اگر یک باشد کرسر چشمک می زند و اگر صفر باشد حالت چشمک زدن غیر فعال می شود

Move Cursor/Shift Display:

SC : اگر یک باشد نمایشگر انتقال فعال است و اگر صفر باشد نمایش انتقال غیر فعال است.

RL : اگر یک باشد جهت انتقال به راست است و اگر صفر باشد جهت انتقال به چپ است.

Set Interface Length:

DL : اگر یک باشد طول واسط **8** و اگر صفر باشد طول واسط به **4** تنظیم می شود.

N : اگر یک باشد تعداد خطوط نمایشگر **2** خط و اکر صفر باشد تعداد خطوط نمایشگر یک خط است.

F : فونت کاراکتر ها (اگر یک باشد **5*10** و اگر صفر باشد **5*7** است)

Poll the "Busy Flag:

BF : این بیت زمانی که **LCD** فعال و در حال پردازش می باشد ست می شود.

Move Cursor to CGRAM/Display:

A : محتوا آدرس مورد نظر برای جابجایی

Read/Write ASCII to the Display

D : داده ها

خواندن اطلاعات گذشته بهترین استعمال در برنامه های کاربردی است که اطلاعات بر روی LCD به عقب و جلو حرکت می کنند (مثل برنامه های کاربردی که اطلاعات در بین خطاها مرور می شود) .

Busy Flag می تواند جهت تعیین زمان ارسال آخرین دستورالعمل جهت تکمیل پردازش انتخاب شود.

در اکثر برنامه های کاربردی، تنها خط«R/W» را به زمینه اتصال داده شده است زیرا هیچ چیزی از گذشته را خوانده نشده است، این نشان دهنده ی برنامه کاربردی است زیرا زمانیکه اطلاعات read back است میکروکنترل کننده ی I/O pins باید بین روش های خروجی و ورودی متناوب باشد.

برای اغلب برنامه های کاربردی واقعاً دلیلی جهت خواندن از LCD وجود ندارد. معمولاً «R/W» را به زمینه اتصال می دهیم وفقط برای ماکزیموم مقدار زمان هر دستورالعمل صبر می کنیم. (4/1 msecs برای وضوح نمایش یا حرکت نشانگر/ نمایش موقعیت خانه، 160 uecs برای تمامی دستورات دیگر) همان طور که برنامه ی کاربردی نرم

افزار simpler ایجاد می شود همچنین پین میکروکنترل کننده برای دیگر کاربردها مجاز می شود. LCD مختلف دستورالعمل هایی را در سرعت های متفاوت اجرا می کند و از مشکلات بعدی جلوگیری می کند.

ما تنها کاربرد ماکزیموم تأخیرات ارائه شده در بالا را توصیه می کنیم. در مفهوم گزینه ها ، هرگز نمایش LCD 10×5 را ندیده ایم . این به این معناست که باید همیشه «F» بیت در (set interface instruction) مجدداً معادل با (0) تنظیم شود. قبل از اینکه شما بتوانید دستورات یا اطلاعات را به واحد LCD ارسال کنید واحد باید شروع دوباره شود. برای روش 8 بیت استفاده از ردیف برنامه های عملکردی زیر اجرا شده است :

1- بیشتر از 15 msecs بعد توان به کار گرفته شده منتظر شوید.

2- 30 0×0 برای LCD نوشته شود و 5 msecs برای دستورالعمل جهت تکمیل به تأخیر بیفتد.

3- 30 0×0 برای LCD نوشته شود وusecs 160 برای دستورالعمل جهت تکمیل به تأخیر بیفتد.

4- AGAIN 30 0×0 برای LCD نوشته شود و 160 usecs به تأخیر بیفتد یا Busy Flag را انتخاب کنید

5- ویژگی های کاربردیLCD را تنظیم کنید

نوشتن تنظیم طول رابط

نوشتن 10 0×0 برای خاموشی نمایش

نوشتن1 00×0 برای وضوح نمایش

نوشتن تنظیم جهت حرکت نشانگر، تنظیم نمودن عملکرد بیت های نشانگر

نوشتن ناتوانی نمایش/ نشانگر و نشانگر گزینشی در توصیف چگونگی LCD باید در روش چهار بیت شروع مجدد شود.

ما نوشتن برای LCD در مفهوم nyblle ها را مشخص خواهیم کرد.در ابتدا تنها nybble های منفرد فرستاده شده اند. همانطور که در بالا ذکر شد زمانیکه یک بایت ارسال شده است nybble بالا قبل از nybble پایین فرستاده شده است و E پین هرزمان که چهار بیت ها به LCD فرستاده می شود تغییر موقعیت یافته اند. جهت آغاز با روش چهار بیت:

1- بیشتر از 15 msecs بعد توان به کار گرفته شده منتظر شوید.

2- 30 0×0 برای LCD نوشته شود و 5 msecs برای دستورالعمل جهت تکمیل به تأخیر بیفتد.

3- 30 0×0 برای LCD نوشته شود وusecs 160 برای دستورالعمل جهت تکمیل به تأخیر بیفتد .

4- AGAIN 30 0×0 برای LCD نوشته شود و 16 usecs به تأخیر بیفتد یا Busy Flag را انتخاب کنید.

5- ویژگی های کاربردیLCD را تنظیم کنید.

نوشتن2 0×0 برای LCD جهت ناتوانی روش 4 بیت .

اطلاعات یا دستورالعمل های لازم برای دو Nybble نوشتن نیاز دارند.

#نوشتن تنظیم طول رابط .

نوشتن 00× 01/0× 0 جهت خاموشی نمایش.

نوشتن 01× 00/0×0 برای وضوح نمایش.

نوشتن تنظیم جهت حرکت نشانگر تنظیم نمودن عملکرد بیت ها ی نشانگر .

نوشتن ناتوانی نمایش/نشانگر و نشانگر گزینشی.

زمانیکه آغاز سازی تکمیل شد می تواند LCD با اطلاعات یا دستورالعملهای مورد نیاز نوشته شود . هر کارکتر برای نمایش مثل بایت های کنترل نوشته شده است . به غیر از خط « R/S » که تنظیم است و در زمان آغاز سازی از طریق تنظیم بیت « S/C » در طول دستور حرکت نشانگر / تغییر نمایش ، بعد هر کارکتر به LCD ارسال شده است ، نشانگر ساخته شده برای LCD به موقعیت بعدی پیشرفت خواهد کرد (یا راست یا چپ) . به طور طبیعی بیت « S/C » معادل (1) مطابق با بیت « R/S » در فرمان «حرکت نشانگر / تغییر نمایش» تنظیم شده و از چپ به راست برای کاراکترها نوشته شده است . (با نمایش ویدئویی تله تاپ).

LCD Layout	Top Left Character	Ninth Character	Second Line	Third Line	Fourth Line	Comments
8x1	0	N/A	N/A	N/A	N/A	Single 44780/No Support Chip
16x1	0	0x040	N/A	N/A	N/A	Single 44780/No Support Chip
16x1	0	8	N/A	N/A	N/A	44780 with Support Chip. This is quite rare
8x2	0	N/A	0x040	N/A	N/A	Single 44780/No Support Chip
10x2	0	8	0x040	N/A	N/A	44780 with Support Chip
16x2	0	8	0x040	N/A	N/A	44780 with Support Chip
20x2	0	8	0x040	N/A	N/A	44780 with Support Chip
24x2	0	8	0x040	N/A	N/A	44780 with Support Chip
30x2	0	8	0x040	N/A	N/A	44780 with Support Chip
32x2	0	8	0x040	N/A	N/A	44780 with Support Chip
40x2	0	8	0x040	N/A	N/A	44780 with Support Chip
16x4	0	8	0x040	0x020	0x060	44780 with Support Chip
20x4	0	8	0x040	0x020	0x060	44780 with Support Chip
40x4	U/N	U/N	U/N	U/N	U/N	Two 44780 with Support Chips. Addressing is device specific

جدول 3 -پایه های مربوط به LCD های مختلف

یک ناحیه گیج کننده چگونگی حرکت به محل های متفاوت روی نمایش آنها است و دیگری چگونگی حرکت به خطوط مختلف روی یک نمایشگر LCD می باشد . جدول زیر چگونگی نمایشات مختلف LCD که 44780 منفرد را استفاده می کنند می تواند با آدرس هایی برای موقعیت های خاص کارکتر تنظیم شومند را نشان می هد . LCD های لیست شده موسومترین تنظیمات موجود می باشند و طرح کلی تحت شماره ستون ها از طریق شماره خطوط ارائه شده است . نهمین کارکتر، موقعیت کارکتر نهم بر روی خطوط اولیه است .

اغلب نمایشات LCD دارای 44780 و تراشه ی حمایتی جهت کنترل عملکرد LCD می باشد . 44780 مسئول خارجی و مهیا سازی خطوط کافی کنترل برای کارکتر ششم LCD می باشد . تراشه ی حمایتی I/o را از 44780 جهت پشتیبانی به کارکتر 128 ، LCD ارتقا می دهد. از جدول بالا ، باید ذکر شود که دوکل اولیه ی (1 × 16 و 1 × 8) تنها دارای 44780 می باشد و حاوی تراشه ی حمایتی نمی باشند. پس کارکتر نهم در 1 × 16 ، در آدرس 8 پدیدار نمی شود و در آدرس که برای 2 خط LCD موسوم می باشد نشان داده می شود.

ما از طریق چهار خط (4 × 40) LCD شامل کارکتر 40 شده ایم، زیرا کاملاً رایج است .

به طور طبیعی LCD به عنوان دو نمایش 2 × 40 سیم کشی شده است .

ارتباط دهنده واقعی موسوم به ششمین بیت های عریض با تمامی ارتباط دهنده های چهارم 44780 ، به جز برای پین های E ، تلقی می شود. E,Stribess برای مخاطب قراردادن بین مناطقی ار عرضه های استعمال شده برای دو دستگاه ، استفاده شده اند.

ارتباط دهنده های واقعی و کارکتر نشان داده شده برای این نوع عرضه می تواند بین سازندگان و شماره های قسمت نمایش متغییر باشد . توجه کنید که زمانیکه هر نوع از نمایش متعدد LCD 44780 به کار گرفته می شود شما باید احتمالاً در هر زمان یک نشانگر 44780 را نمایان کنید. برای 44780 نشانگر ها می توانند به عنوان یک تأکید ساده در هر زمانی روشن شوند و «عرضه ناتوان / نشانگر » دستورالعمل LCD استفاده می شود. بیت C تنظیم شده است.

ما استعمال بیت B را به عنوان دلیلی برای بخش کامل کارکتر مربع نمایش داده شده سفارش نمی کنیم .

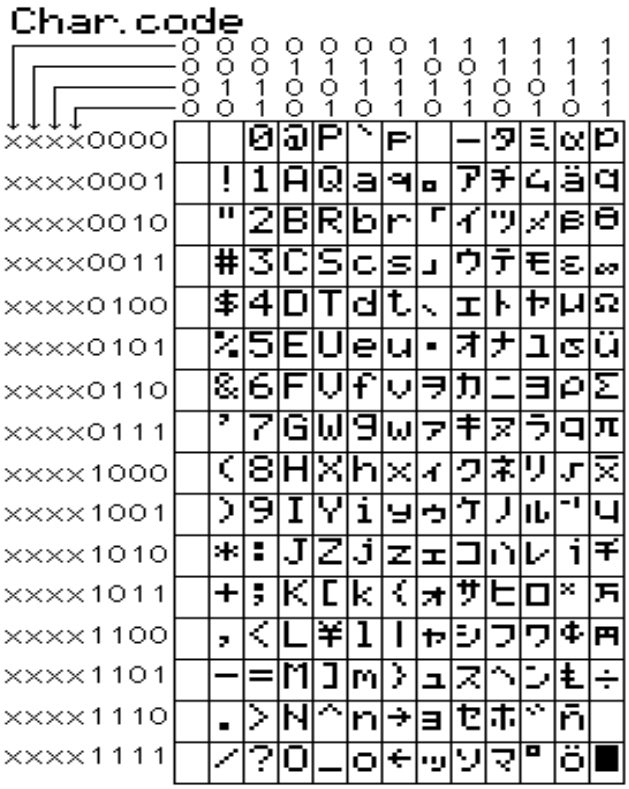
LCD می تواند به عنوان یک نمایش تله تایپ اندیشه شود زیرا در عملکرد طبیعی بعد یک کارکتر به LCD فرستاده شده است . نشانگر درونی یک کارکتر را به سمت راست حرکت داده است .

نمایش واضح و بازگشت نشانگر و LCD برای دستورالعمل های موقعیت خانه جهت تنظیم نشانگر و جهت نمایش استفاده شده است .

برای این دستورالعمل ، بیت 7 ، بایت دستورالعمل با هفتمین بیت های باقی مانده تنظیم شده است . این 7 بیت ها 128 آدرس، را با انطباق ماکزیمم تعداد آدرس های موجود کارکتر LCD ما مهیا می کنند.

جدول بالا جهت تعیین آدرس کارکتر offset روی یک خط ویژه از نمایش LCD استفاده می شود.

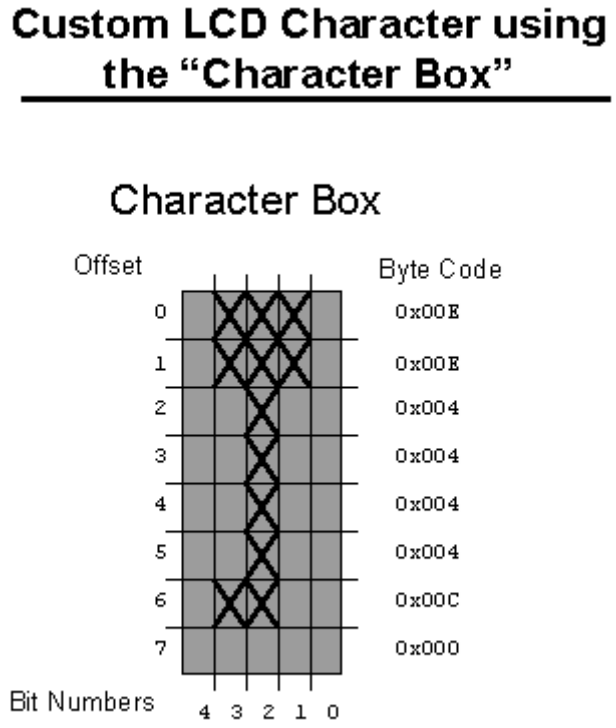
اساساً تنظیم کارکتر موجود در 44780 ASCII می باشد . من واژه اساساً را به کار می برم زیرا برخی کارکترها به طور کامل قاعده ی ASCII را دنبال نمی کنند. کارکترهای کنترل ASCII (0×008×0×01 F) به عنوان کارکترهای کنترل پاسخ داده نمی شود و ممکن است کارکترهای سرگرمی ژاپنی را نمایش دهند. تنظیم کارکتر LCD نشان داده شده در زیر تواضع peer cuwehand و صفحه عالی LCD او تلقی می شود.



جدول 4-کاراکتر های ذخیره شده در حافظه

هشت کارکتر قابل برای برنامه ریزی موجود می باشد که از کدهای 007×0 در 000×0 استفاده می کننـد. آنهـا از طریق اشاره به نشانگر LCD ها برای منطقه کارکتر ژنراتور RAM برنامه ریزی شده است .

هشت کارکتر بعدی نوشته شده برای RAM هر خط از کارکتر قابل برنامه ریزی شده آغاز یافته از رأس تلقـی مـی شود. من مایلم این امر را به عنوان مربع های 5 × 8 در نمودار سمت راست بیان کنم. در بالا، ذکر کـردم کـه اغلـب نمایشات برای هر کارکتر 5 × 7 پیکسل می باشد . بنابراین ردیف اضافی ممکن است گیج کننده باشـد هـر کـارکتر LCD واقعاً دارای 8 پیکسل بلندی است که با ردیف پائین برای تأکید بر نشانگر به طول نرمال استفاده شده است . ردیف پائین برای کارکترهای گرافیک می تواند استفاده شود اگر چه اگر شـما قصـد اسـتفاده از یـک نشـانگر واضـح underscore را دارید و آن را در هر کارکتر دارید من سفارش می کنم که از آن بهره نگیرید. با استفاده ازاینباکس شما می توانید در پیکسل هایی که کارکتر خاص شما را تعریف می کنند طراحی کنیـد و سـپس بیـت هـا را بـرای تعیین چگونگی واقعی کدهای اطلاعاتی ، استفاده کنید .



شکل 8-1- نمایی از یک کاراکتر طراحی شده

زمانیکه ما این کار را انجام دادیم به طور طبیعی از یک تکه کاغذ گراف استفاده کردیم و سپس کدهای هگزادسیمال را برای هر خط نوشتم همانطور که انها را در نمودار راست شده است . برای کاربردهای انیمنت ، ما کارکتر چرخش را برای انیمیشن ها استفاده می کنیم. به سادگی ما کارکترهای مختلف را نشان دادیم . که تنها (دو بیت) به LCD ارسال شده است . اگر انیمیشن از طریق تبیین مجدد کارکترها اجرا شده بود پس 10 کارکتر به LCD ارسال خواهد شد . (یکی برای حرکت به فضای CG RAM هشت کارکتر برای تعریف و یک دستورالعمل بازگشتی برای نمایش RAM) اگر کارکترهای متعدد قصد کاربرد دارند یا بیشتر از هشت عکس برای انیمیشن وجود دارد شما باید در هر زمان کارکتر جدید را مجدداً یادداشت کنید .

کاربری که اطلاعات خطی کارکتر را تعریف کرده است در ناحیه ی CGRAM ، LCD ذخیره شده است . این 64 بایت حافظه در دسترس قرار گرفته است استفاده از دستورالعمل حرکت نشانگر در CGRAM حالت مشابهی از حرکت نشانگر برای یک ادرس ویژه در حافظه با یک اختلاف مهم تلقی می شود

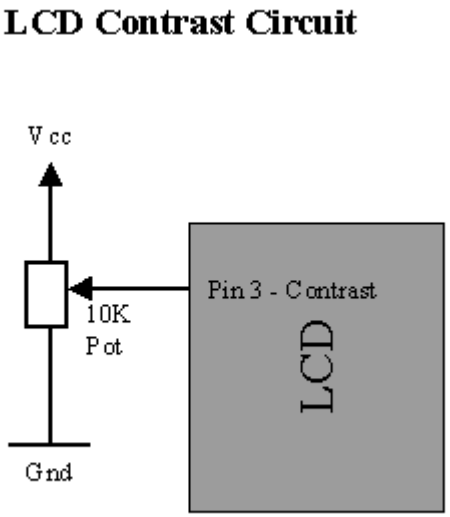
. اختلاف بیان شده این است که هر کارکتر در هشت دوره ی مقدار کارکتر خود آغاز شده است . بـه ایـن معنـا کـه کارکتر 0 قابل تعریف کاربر دارای آغاز اطلاعات خود در آدرس 0 از CGRAM است و کـارکتر در آدرس 8 شـروع شده است و کارکتر 2 در ادرس (10616 0×0) آغاز شده است و ... .

جهت کسب یک مسیر ویژه در طول کارکتر قابل تعریف کاربر افست آن از بالا به آدرس شـروع شـونده اضـافه شـده است . در اکثر برنامه های کاربردی، کارکترها برای همه در یک دوره با کارکتر 0 اولیه نوشـته شـده اسـت . در ایـن مورد دستورالعمل 040×0 برای LCD نوشته شده است که از طریق کارکترهای تعریف شده ی کاربر دنبـال شـده است .

که یک نکته خاص برای الکترونیک هـای wriz (SLI -OEM) کـاربران مـی باشـد . زمانیکـه کارکترهـای جدیـد تعریف شده اند ایده ی خوبی است که اطمینان حاصل کنیم سه بیت بالایی در بایت کارکتر تعریـف شـده ی کـاربر تنظیم یافته اند . زمانیکه دستورالعمل حرکت نشانگر در CGRAM دریافت شـده اسـت ، SLI -OEM بـه روش ویژه ای سوق داده شده است جایی که اطلاعات کارکتر ردیف شمارشگر به روز نشده است در زمان دریافـت کـارکتر جدید این امر حادث می شود. این روش زمانیکه دستورالعمل جدید بـه SLI -OEM ارسـال مـی شـود یـا یـک , ASCII "Backspace" "Carriage Return" "Line Feed" یا "Form Feed"دریافت مـی شـود بسـته شده است .

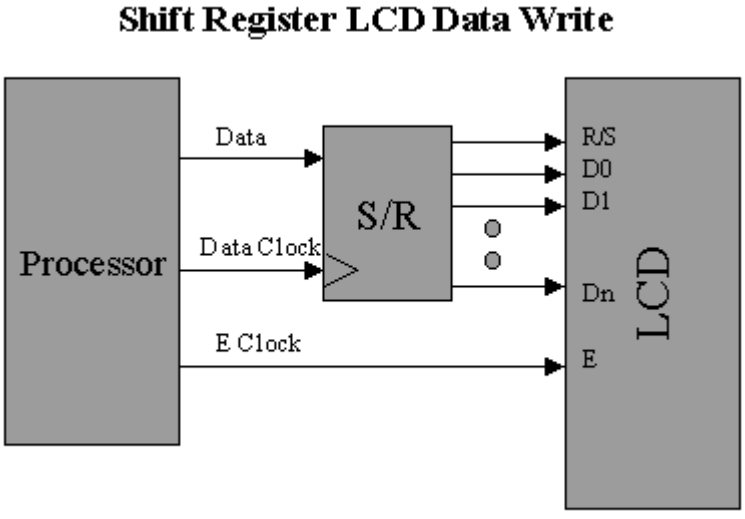
چون تمامی این کارکتر ها معتبر هستند کاربر LCD توضیحات مسـیر کـارکتر را تعریـف مـی کنـد شـما خواهیـد دریافت SLI -OEM اطلاعات را به درستی تفسیر نمی کند. اگر ما علامت مردمالار را برای نمـایش روی OEM-SLI تهیه کرده بودیم از بایت EE 0×0 برای اولین خط به جای 00E×0 استفاده خواهیم کرد.

جنبه ی اخیر LCD برای بحث چگونگی تعیین تباین ولتاژ برای نمایش است . معمولاً من از یک مقسم ولتاژ سـیم کشی شده به عنوان تقسیم کننده ی ولتاژ استفاده می کنم. این امر بین زمینه و VCC که برای تعیـین تبـاین (یـا تاریکی) کارکترها بر روی صفحه ی LCD به کار می رود به آسانی ولتاژ متغییری را فراهم خواهد کرد. ممکن است شما درک کنید که LCD های مختلف با ولتاژهالی پائین تر به طور متفاوتی عمل می کنند که کارکترهـای تاریـک در برخی در ان ها مهیا می شود و ولتاژهای بالاتر چیزهای مشابه دیگری را تهیه می کنند.



شکل 8-2- شمای داخلی تغییر شفافیت نمایشگر

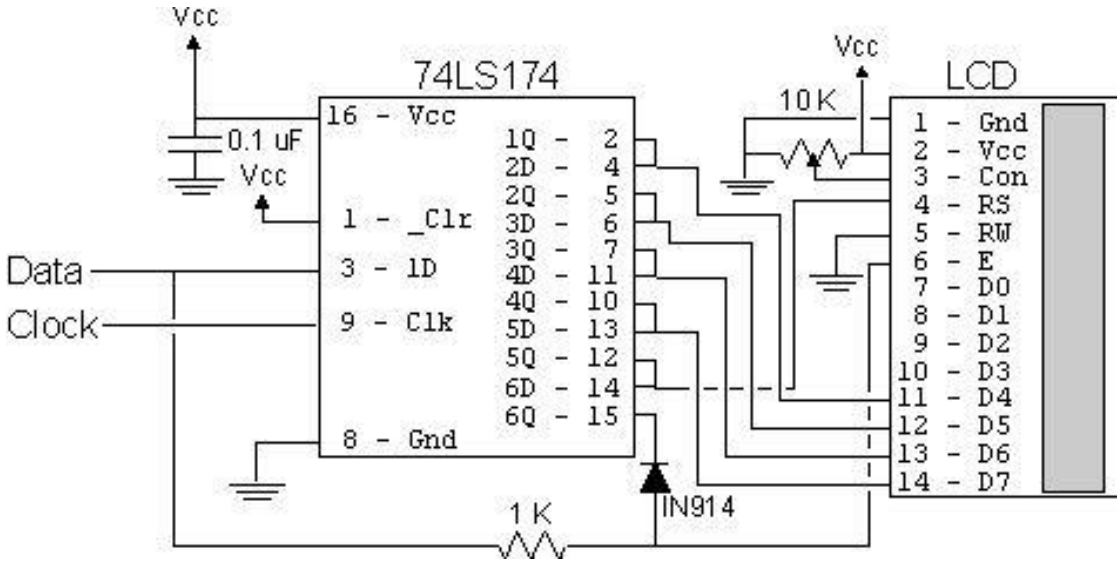
روش های گوناگون متغییری برای سیم کشی LCD وجود دارد . من متوجه شدم که 44780 می تواند با 4 یا 8 بیت ها ارتباط برقرار کند. جهت نمایش تقاضا ها در میکرو کنترل کننده ها ، تغییر رجیستر اغلب جهت کاهش تعداد پین های I /0 به سه مورد استفاده قرار می گیرد. و می تواند از طریق کاربرد مدار نشان داده شود زیا در اطلاعات سریالی ترکیب شده با محتوای تغییر رجیستر جهت تولید E strobe در فواصل مناسب کاهش بیشتری



بدست آورد .

شکل 8-3- شیفت رجیستر نمایشگر برای چاپ اطلاعات

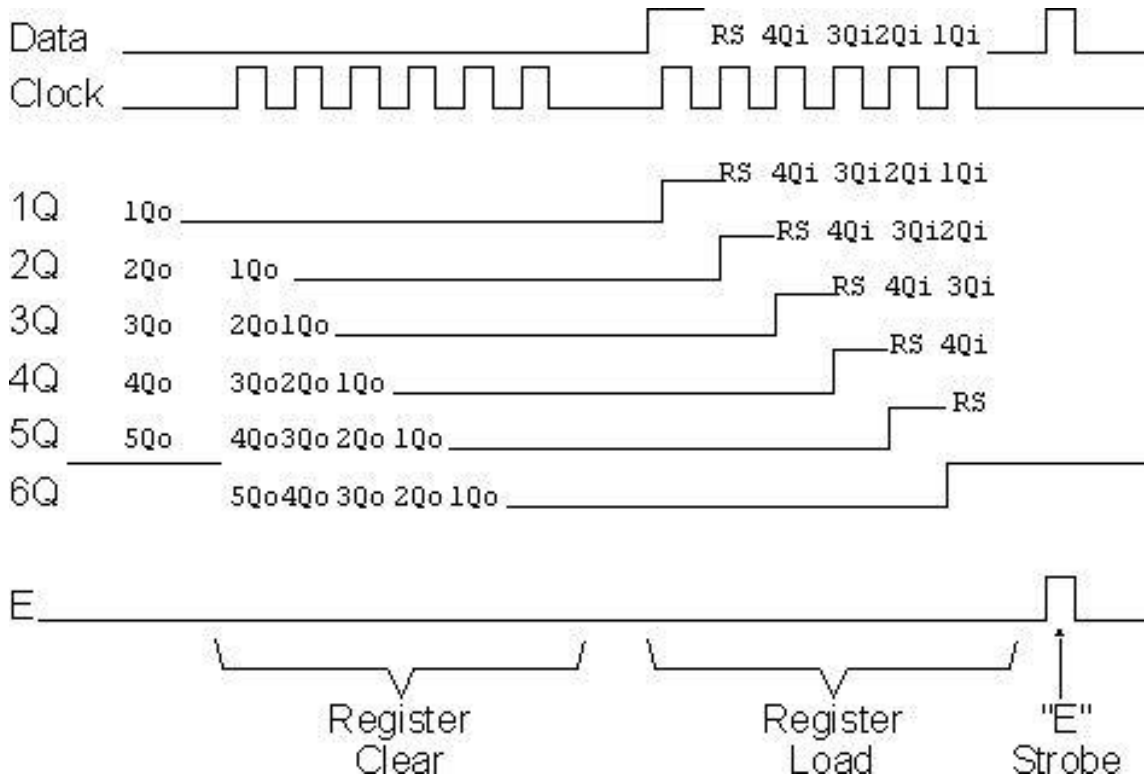
این مدار «AND $_S$ » (مقاوم 1K و دیود 1N 914 را به کـار مـی بـرد) خروجـی ششـم (D- Flip Flop) از 74 LS 174 و اطلاعات و اطلاعات بیت از ابزار نوشتاری برای LCD بـه فـرم E strobe مـی باشـد . . ایـن روش نیازمند یک پین کمتر از سه سیم رابط و یک دستورالعمل کمتر از کد می باشد .



شکل 8-4- مدار اتصال ال سی دی

. ما طبیعتاً یک 74 LS 174 سیم کشی شده به عنوان تغییر رجیستر (همانطور که در نمـودار قیاسـی نشـان داده شده است) را به جای یک رجیستر تغییر Serial- in/parallel-out استفاده کرده ایم .

باید مدار بدون هیچ مشکلی با یک تراشه ی رجیستر Serial- in/parallel-out اختصاصی کار کند ، اما تناقضات تنظیم وقت ممکن است متفاوت باشند. در حالی که 74 LS174 به کار رفته است توجـه کنیـد کـه اطلاعـات روی ارتقا لبه ی سیگنال ساعت (از منطق کم به بالا) قفل شده اند



شکل 8-5- سیگنال های ارسالی

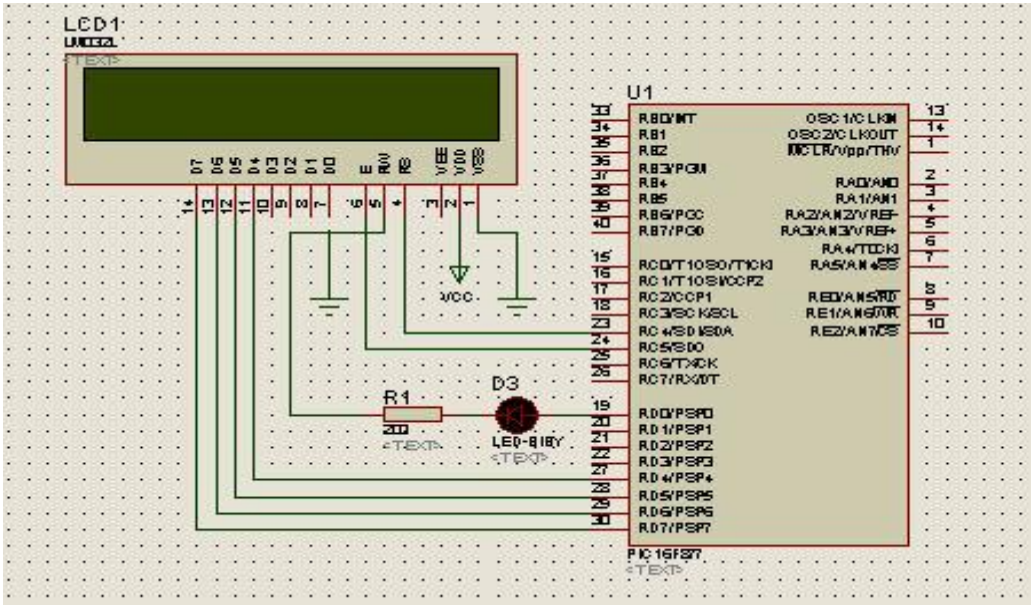
در این نمودار برای سمت راست ، من چگونگی تغییر رجیستر نوشته شده جهت کار این مدار را نشان داده ام . قبل از اینکه اطلاعات برای آن نوشته شده است که از طریق بیت « R/S » و اطلاعات چهار بیت پیروی می شود . چون به طرز صحیحی بارگذاری شده اند مسیر اطلاعات به بیت E Storbe اضافه شده است . بزرگترین اختلافات بین سه سیم و دو سیم این است که تغییر رجیستر باید قبل از اینکه بتواند بارگذاری شود ، وضوح یابد و عملکرد دو سیم نیازمند بیشتر از 2 بار شماره ی چرخش های ساعت جهت بارگذاری چهار بیت برای LCD می باشد ، من این مدار را با PICMICRO ، 8051 و AVR استفاده کرده ام و این امر واقعاً نوشتم LCD را برای یک میکرو کنترل کننده ی را بسیار آسان می سازد .

مزیت مهم استفاده از تغییر رجیستر مثل دو مدار نشان داده شده در اینجا می باشد . اطلاعات برای LCD دارای فقدان زمان بندی حساسی است که با آن مواجه خواهد شد . بزرگترین نمونه برای تماشا کسب اطمینان از زمان بندی E Storbe در تصریح ان می باشد . (به عنوان مثال بیشار از 450 nsecs) بارهای تغییر رجیستر می تواند بدون تأثیر گذاری نوشته واقعی متوقف شوند. این مدار نه تنها با خروجی ها کار نمی کند بلکه در Drain باز نیز

دارای عمل نمی باشد . (چیزی که اکثر مردم را درگیر خود ساخته است) یک نکته در مـوردLCD هـای Storbe E این است که در برخی مستند سازی ها تحت عنوان سطح بالای فعال تعیین شده اند که لبه افت فعال تلقـی مـی شود و به نظر می رسد چگونگی ارتباط wire LCD- 2 بیان شده را ارائه می دهد حتـی اگـر خطـوط پایـانی بـالا باشند در انتها اطلاعات تغییر می یابند. اگر لبه ی افت به کار رفته است (مثل رابط wire- 2) مطمـئن شـوید کـه قبل از اینکه خط E خروجی O باشد حداقل یک تعویق 450 nesecs با هیچ خطوط وضعیت تغییر موجود نباشد .

و در پایان برای کاربردی کردن این فصل یک نمونه برنامه متنی با MPLAB به همراه مدار نمونه آن در نرم افزار PROTEUS گذاشته ایم .

مدار نمونه در نرم افزار PROTEUS:



شکل 8-6- شمای مدار در نرم افزار پروتئوس

```
/*
 *      LCD interface example
 *      Uses routines from delay.c
 *      This code will interface to a standard LCD controller
 *      like the Hitachi HD44780. It uses it in 4 bit mode, with
 *      the hardware connected as follows (the standard 14 pin
```

```
*       LCD connector is used):
*       pic16f877a  Anjidani
*       PORTD bits 4-7 are connected to the LCD data bits 4-7 (high nibble)
*       PORTC bit 4 is connected to the LCD RS input (register select) ,Anjidani
*       PORTC bit 5 is connected to the LCD EN bit (enable)  ,Anjidani
*
*       To use these routines, set up the port I/O (TRISA, TRISD) then
*       call lcd_init(), then other routines as required.
*
*/


#include        <pic.h>
#include        "lcd.h"
#include        "delay.h"

static bit LCD_RS @ ((unsigned)&PORTC*8+4); // LCD Reg. Select ,Anjidani
static bit LCD_EN @ ((unsigned)&PORTC*8+5); // LCD Enable ,Anjidani

//#define LCD_DATA          PORTD (4 high bits)

#define LCD_STROBE ((LCD_EN = 1),(LCD_EN=0))

/* write a byte to the LCD in 4 bit mode */

void lcd_write(unsigned char c)
{
        PORTD = (PORTD & 0x0F) | (c & 0xF0);
        LCD_STROBE;
        PORTD = (PORTD & 0x0F) | (c << 4);
        LCD_STROBE;
        DelayUs(40);
}
```

```
/*
 *      Clear and home the LCD
 */

void lcd_clear(void)
{
        LCD_RS = 0;
        lcd_write(0x1); //-- Clear the Display
        DelayMs(2);
        lcd_write(0x02); //-- Home the display
        DelayMs(2);
}

/* write a string of chars to the LCD */

void lcd_puts(const char * s)
{
        int i;
        LCD_RS = 1;  // write characters
        while(*s)
                lcd_write(*s++);
}

/* write one character to the LCD */

void lcd_putch(char c)
{
        LCD_RS = 1;  // write characters
        lcd_write( c );
}
```

```
/*
 * Go to the specified position
 */

void lcd_goto(unsigned char pos)
{
	LCD_RS = 0;
	lcd_write(0x80+pos);
}

void lcd_scroll(char Direction)
{
	LCD_RS=0; //-- write command
	if(Direction==0)
		lcd_write(0x18); //- S/C=1 R/L=0 => Scroll LEFT
	else
		lcd_write(0x1c); //- S/C=1 R/L=1 => Scroll RIGHT
	DelayMs(1);
}

void lcdprint(unsigned char x,unsigned char *str)
{
	lcd_goto(x);
	lcd_puts(str);
}

void lcd_home(void)
{
LCD_RS=0; //-- write command
lcd_write(0x02); //-- Home Everything
DelayMs(2);
```

```
}

void lcd_cursor(char onoff)
{
	LCD_RS=0;
	if(onoff==0) //- 0000 1DCB
	{ //- D= Display C= Cursor B=Blink
		lcd_write(0x0C); //- D on/off= 1 C on/off=0 B on/off=0
	}
	else
	{
		lcd_write(0x0F); //- D=1 C=1 B=1 => Cursor ON + Blink
	}
	DelayMs(1);
}

/* initialise the LCD - put into 4 bit mode */
void lcd_init()
{
	char init_value;
	init_value = 0x3;
	TRISD4=0;
	TRISD5=0;
	TRISD6=0;
	TRISD7=0;
   //////////////
	TRISC4=0;
	TRISC5=0;
	//////////////
	LCD_RS = 0;
	LCD_EN = 0;
	DelayMs(15);  // wait 15mSec after power applied,
	PORTD = (PORTD & 0x0F) | (init_value << 4);
```

```
        //LCD_DATA = init_value;
        LCD_STROBE;
        DelayMs(5);
        LCD_STROBE;
        DelayUs(200);
        LCD_STROBE;
        DelayUs(200);
        PORTD = (PORTD & 0x0F) | (2 << 4);
        //LCD_DATA = 2;     // Four bit mode
        LCD_STROBE;


        lcd_write(0x28); // 4 bit mode, 1/16 duty, 5x8 font
        lcd_write(0x08); // display off
        lcd_write(0x0F); // display on, blink cursor on
        lcd_write(0x06); // entry mode
}
```

فصل هشتم

USART یک فرستنده-گیرنده همگام- ناهمگام عمومی استکه اطلا عات رقمی موازی را به اطلاعات رقمی متوالی تبدیل می کند.این وسیله سیگنال های کنترل را برای مدم و اطلاعات دست تکانی را تامین می کند و آشکار سازی خطا را امکان پذیر می کند.

USART میتواند در حالت های زیر ایجاد پیکربندی کند:
اسنکرون (ناهمگام) – (به صورت کاملاً دوطرفه)
سنکرون (همگام)مدل اصلی – (به صورت یکطرفه)
سنکرون (همگام)مدل پیرو – (به صورت یکطرفه)
بیت SPEN (RCSTA<7>) و بیت های TRISC<7:6> برای پیکربندی پین های RC6/TX/CK و RC7/RX/DT به صورت فرستنده-گیرنده همگام- ناهمگام عمومی ،باید یک شوند.
همچنین ماژول USART یک ارتباط چندپردازنده ای با توانایی استفاده از 9 بیت آدرس دارد.

رجیستر10-2 :RCSTA: وضعیت دریافت و رجیستر کنترل
(آدرس 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit7							bit0

R = بیت خواندنی
W = بیت نوشتنی
U= استفاده نشده : با صفر خوانده میشود.
-n = مقدار بازنشانی POR

بیت 7 SPEN : بیت فعال سازی پورت سریال
1= پورت سریال فعال شده است (پیکربندی پین های RC6/TX/CK و RC7/RX/DT همانند پین های پورت سریال)
0 = پورت سریال غیر فعال شده است.

بیت 6 RX9 : بیت فعال سازی دریافت 9 بیت

1= انتخاب 9 بیت دریافتی
0= انتخاب 8 بیت دریافتی

بیت 5 SREN : بیت فعال سازی دریافت سیگنال
روش ناهمگام بی اهمیت است.
روش همگام- مدل اصلی :
1= فعال سازی دریافت سیگنال
0= غیر فعال سازی دریافت سیگنال
بعدازاینکه دریافت کامل شد، این بیت صفر می شود.
روش همگام –مدل پیرو در این حالت استفاده نمی شود.

بیت 4 CREN : بیت فعال سازی دریافت متوالی
روش ناهمگام :
1= فعال سازی دریافت متوالی
0= از کار انداختن دریافت متوالی
روش همگام:
1= فعال سازی دریافت متوالی تا هنگامیکه بیت فعالسازی CREN پاک شود(CREN بیت SREN را باطل می کند)
0= از کار انداختن دریافت متوالی

بیت 3 ADDEN : بیت فعال سازی یافتن آدرس
روش ناهمگام 9 بیتی
1= فعال کردن یافتن آدرس، فعال کردن وقفه و بارکردن بافر دریافت، هنگامیکه <RSR<8 یک شده باشد.
0= از کار انداختن یافتن آدرس، همه بایت ها دریافت شده اند وبیت نهم میتواند به عنوان بیت توازن استفاده شود.

بیت 2 FERR : بیت خطای قاب بندی
1= خطای قاب بندی (میتواند با خواندن رجیستر RCREG بهنگام شود وبایت صحیح بعدی را دریافت کند)
0= عدم وجود خطای قاب بندی

بیت 1 OERR : بیت خطای سرریز
1= خطای سرریز (میتواند با صفر شدن بیت CREN پاک شود)
0= عدم وجود خطای سرریز

بیت 0 RX9D : بیت نهم داده دریافت شده(می تواند بیت توازن با شد)

رجیستر10-1 :TXSTA: وضعیت ارسال کردن و رجیستر کنترل

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
bit7							bit0

بیت 7 CSRC: بیت انتخاب منبع ساعت

روش ناهمگام بی اهمیت است.

روش همگام

1= وضعیت اصلی(تولید ساعت درونی بواسطه BRG)

0= حالت پیرو (تولید ساعت بواسطه منبع خارجی)

بیت 6 TX9 : بیت فعال سازی ارسال 9 بیت

1= انتخاب 9 بیت ارسالی

0= انتخاب 8 بیت ارسالی

بیت 5 TXEN : بیت فعال سازی ارسال

1= فعال سازی ارسال

0= ازکار انداختن ارسال

توجه:در حالت همگام، SREN/CREN بیت TXEN را لغو میکند.

بیت 4 SYNC: بیت انتخاب حالت USART

1 = حالت سنکرون (همگام)

0= حالت اسنکرون (ناهمگام)

بیت 3 UNIMPLEMENTED (استفاده نشده):

با صفر خوانده میشود.

بیت 2 BRGH: بیت انتخاب سرعت بالای باود(علامت)در ثانیه

روش ناهمگام :

1= سرعت بالا

0= سرعت کم

روش همگام در این حالت استفاده نمی شود.

بیت 1 TRMT: بیت وضعیت TSR(ارسال شیفت رجیستر)

1= TSR خالی است.

0= TSR پر است.

بیت 0 RX9D : بیت نهم داده ارسال شده(می تواند بیت توازن با شد)

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64(X+1))	Baud Rate= Fosc/(16(X+1))
1	(Synchronous) Baud Rate = Fosc/(4(X+1))	NA

جدول 5: **فرمول سرعت باود(علامت در ثانیه)**

X = مقدار SPBRG (از صفر تا 255)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

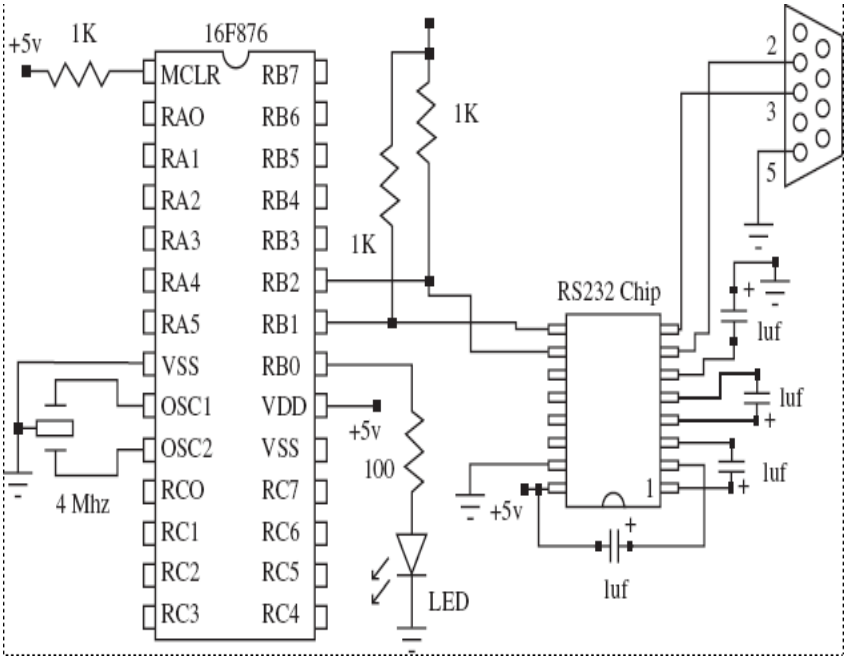
BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.301	0.33	185
1.2	1.202	0.17	51	1.216	1.33	46
2.4	2.404	0.17	25	2.432	1.33	22
9.6	8.929	6.99	6	9.322	2.90	5
19.2	20.833	8.51	2	18.643	2.90	2
28.8	31.250	8.51	1	-	-	-
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	55.930	2.90	0
HIGH	0.244	-	255	0.218	-	255
LOW	62.500	-	0	55.930	-	0

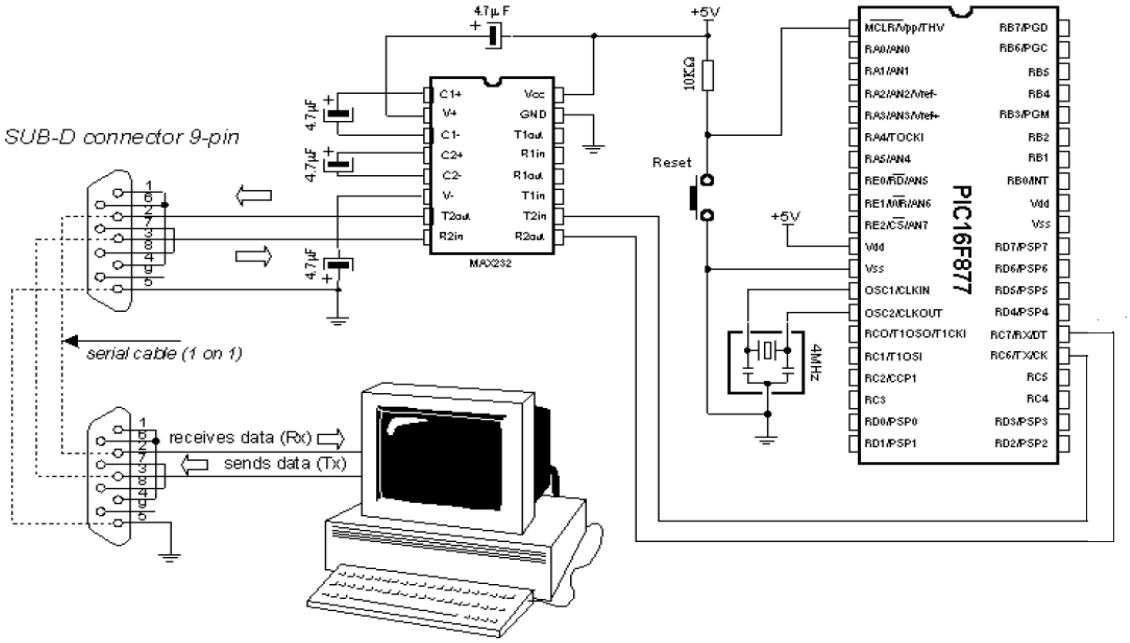
جدول 6: **سرعت باود(علامت در ثانیه) برای حالت ناهمگام (BRGH=0)**

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000		0	625.000	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.203	0.25	185
2.4	2.404	0.17	103	2.406	0.25	92
9.6	9.615	0.16	25	9.727	1.32	22
19.2	19.231	0.16	12	18.643	2.90	11
28.8	27.798	3.55	8	27.965	2.90	7
33.6	35.714	6.29	6	31.960	4.88	6
57.6	62.500	8.51	3	55.930	2.90	3
HIGH	0.977	-	255	0.874	-	255
LOW	250.000	-	0	273.722	-	0

جدول 7: **سرعت باود(علامت در ثانیه) برای حالت ناهمگام (BRGH=1)**





شکل 8-7- شمای یک فرستنده-گیرنده همگام- ناهمگام عمومی و نحوه اتصال آن با کامپیوتر:

هنگام تنظیم کردن یک ارسال ناهمگام(اسنکرون) مراحل زیر را باید دنبال کنید:

1. رجیستر SPBRG را بواسطه سرعت باود مناسب مقدار دهی اولیه کنید.اگر یک سرعت بـاود بـالا نیاز باشد، بیت BRGH را یک کنید.(بخش 10-1)

2.پورت سریال ناهمگام را بوسیله صفر کردن بیت SYNC ویک کردن بیت SPEN فعال کنید.

3.اگر وقفه ها مورد نیاز باشند ، بیت TXIE را یک کنید.

4.اگر 9 بیت داده برای ارسال نیاز دارید ، بیت TX9 را برای ارسال کردن ، یک کنید.

5.ارسال اطلاعات را با یک کردن بیت TXEN فعال کنید که این نیز بیت TXIF را یک میکند.

6.اگر ارسال 9 بیت داده انتخاب شده است ،بیت نهم باید در بیت TX9D بار شود.

7.داده ها را در رجیستر TXREG بار کنید (ارسال داده ها را شروع کنید.)

هنگام تنظیم کردن یک دریافت ناهمزمان(اسنکرون) مراحل زیر را باید دنبال کنید:

1. رجیستر SPBRG را بواسطه سرعت باود مناسب مقدار دهی اولیه کنید.اگر یک سرعت بـاود بـالا نیاز باشد، بیت BRGH را یک کنید.(بخش 10-1)

2.پورت سریال ناهمگام را بوسیله صفر کردن بیت SYNC ویک کردن بیت SPEN فعال کنید.

3.اگر وقفه ها مورد نیاز باشند ،سپس بیت RCIE را یک کنید.

4.اگر 9 بیت داده برای دریافت نیاز دارید ، بیت RX9 را یک کنید.

5.دریافت اطلاعات را با یک کردن بیت CREN فعال کنید .

6.هنگامیکه دریافت اطلاعات کامل شد ،پـرچم بیـت RCIF یـک میشـود و اگـر بیـت RCIE یـک باشد،یک وقفه تولید میشود.

7. اگر هیچ خطایی در مدت دریافت رخ نداده باشد،رجیستر RCSTA را برای بدست آوردن بیت نهـم (اگر فعال شده باشد) بخوانید و تعیین کنید.

8. بیت هشتم داده دریافت شده را بوسیله خواندن رجیستر RCREG بخوانید.

9. اگر هیچ خطایی رخ نداد ، با صفر کردن بیت فعال CREN ،خطا را صفر کنید.

پایان