

فهرست

- ۱- دستور کار جلسه اول : نحوه ی استفاده از مجموعه ی آزمایشگاهی مدل AZ-SH AP8 ۳
- توانایی کار با مجموعه ی آموزشی ۳
- ویژگی های مشترک در کل مجموعه : ۶
- نحوه راه اندازی و استفاده از مجموعه آموزشی مدل AZ-SH AP8 ۱۰
- ماجول های تعبیه شده در برد ۱ ۱۲
- ماجول های قرار داده شده در برد ۲ ۱۷
- ماجول های قرار داده شده در برد ۳ ۱۹
- نحوه ی راه اندازی و استفاده از پروگرامر سه کاره AVR ۲۲
- جواب سوال در مورد مجموعه ی آموزشی (دانشجوکن باید به این سوالات پاسخ دهند) : ۳۲
- ۱- ماجول منبع تغذیه دستگاه را بچها کرئج ؟ ۳۲
- ۲- ماجول منبع تغذیه این دستگاه چند نوع ولتاژ میتواند تولئج کند ؟ ۳۲
- ۳- میکروکنترلر های مورد استفاده در این مجموعه از کدام ولتاژ استفاده خواهند کرد ؟ ۳۲
- ۴- آلی میتواند از ولتاژ ۱۲ ولت برای تغذیه ی میکرو کنترلر استفاده کرد ؟ چرا ؟ ۳۲
- ۵- کاربرد ولتاژ های متقارن چیست ؟ ۳۲
- ۶- تعداد ماجول های قرار داده شده در روی بردهای اصلی مجموعه چند تا است ؟ ۳۳
- ۷- ماجول های قرار داده شده در روی برد شماره ۱ (برد A) کدام ها هستند ؟ ۳۳
- ۸- ماجول ابزار دقیق را بر روی مجموعه بچها کرئج ؟ ۳۳
- ۹- بر روی ماجول ابزار دقیق چند نوع سنسور حرارتی نصب شده است ؟ آنها را بچها کرده و بگوئج از کدام نوع هستند و چرا از آن نوع سنسور ها استفاده شده است ؟ ۳۳
- ۱۰- چند نوع عملگر بر روی مجموعه قرار دارد ؟ نام ببرئج و در مورد تفاوتهای آنها با یکدیگر و نئج کاربر های هر کدام با دوستانتان صحبت کرئج ۳۳
- ۱۱- به نظر شما در این برد از چند روش برای ارتباط بئج ماجول های مختلف استفاده میشود ؟ دلیلی استفاده از هر یک را توضئج ده بئج : ۳۳
- ۱۲- ماجول اصلی در برد شماره ۱ را بچها کرئج و بگوئج که برای پروگرام کردن میکروکنترلر AVR از کدام سوکت استفاده میشود ؟ ۳۴
- ۱۳- کانکتور PORTA بر روی ماجول اصلی را بچها کرده و بگوئج که این پورت چگونه به میکروکنترلر متصل میباشد و از این کانکتور در کجا میتواند استفاده کرد ؟ ۳۴



- ۱۴- فرض کنید میخواهیم از LED های قرار داده شده در برد شماره ۱ استفاده کنیم . چه کاری را بای انجام دهیم ؟ ۳۵
- ۱۵- فرض کنید بخواهیم میکروکنترلر را با استفاده از واسط RS232 به دنگی خارج ارتباط دهیم . از کدام ماچول استفاده خواهیم کرد ؟ مراحل کار را شرح دهیم . ۳۵
- ۱۶- LED قرار داده شده در ماچول مربوط به I/O سمت راست (در برد شماره ۱) روشن نیست . این به چه معنی است ؟ ۳۵
- ۱۷- فرض کنید میخواهیم از LCD کاراکتری قرار داده شده در مجموعه استفاده کنیم و اطلاعاتی را بر روی آن به نمایش در آوریم . چه کارهایی بای انجام دهیم ؟ ۳۶
- ۱۸- نقشه ی مربوط به ورودی و خروجی سمت راست را از روی نقشه های مجموعه پیدا کنیم و در مورد آن با دوستانتان صحبت کنیم . ۳۷
- ۱۹- پروگرامر مورد استفاده در این دستگاه از کدام پورت ها برای برنامه ریزی میکروکنترلر مورد نظر استفاده میکند ؟ ۳۸
- ۲۰- در کامپیوترهایی که دارای پورت موازی و سرطال و USB هستند کدام یک از پورتها برای استفاده در پروگرامر بهتر هستند ؟ ۳۹
- ۲۱- نحوه ی تنظیم پروگرامر در نرم افزار مورد استفاده خود را شرح دهیم : اگر از BASOM AVR استفاده میکنیم و بی اگر از CODEVISION AVR استفاده میکنیم توضیح دهیم که چگونه میتوانیم از پروگرامر موازی استفاده کنیم . ۳۹
- آزمایش ۳: کار با پورتهای ورودی و خروجی ۴۱
- آزمایش ۴: کار با دستورات شرطی و انتخابی ۵۰
- آزمایش ۵: نحوه ی کار با LCD های کاراکتری ۵۵
- آزمایش ۶: کار با کلید مائرسی ۴*۴ ۶۳
- آزمایش ۷: کار با سرویس های وقفه در میکروکنترلر ۶۶
- آزمایش ۸: کار با مبدل آنالوگ به دیجیتال داخلی میکروکنترلر ۷۱
- آزمایش ۹: کار با تایمر های میکروکنترلر ۷۶
- آزمایش ۱۰: راه اندازی و استفاده از پورت سرطال میکروکنترلر ۸۶
- آزمایش ۱۱: کار با موتور پله ای ۹۰
- آزمایش ۱۲: کار با موتور DC ۹۷

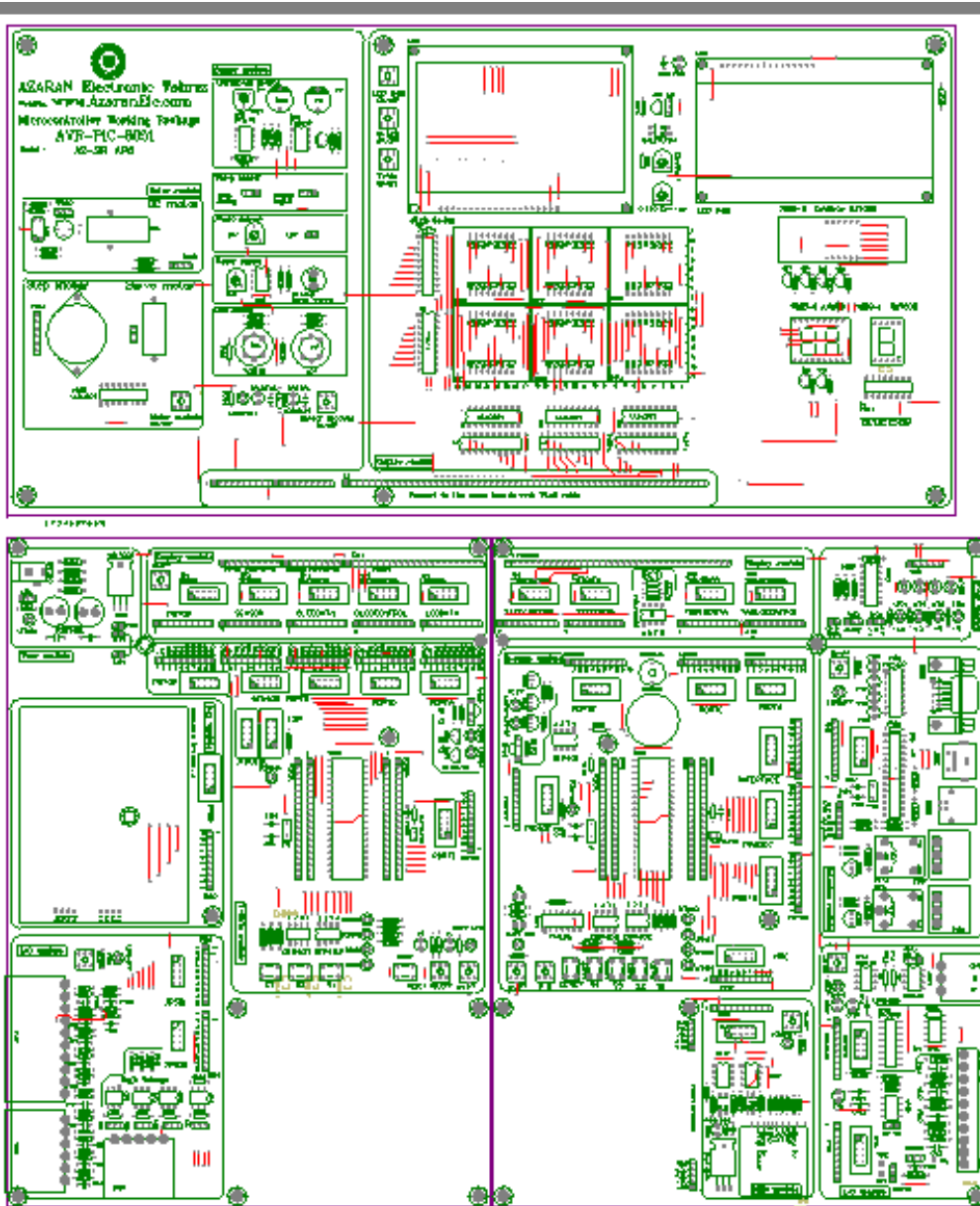
۱- دستور کار جلسه اول : نحوه ی استفاده از مجموعه ی آزمایشگاهی مدل AZ-SH AP8

توانایی کار با مجموعه ی آموزشی

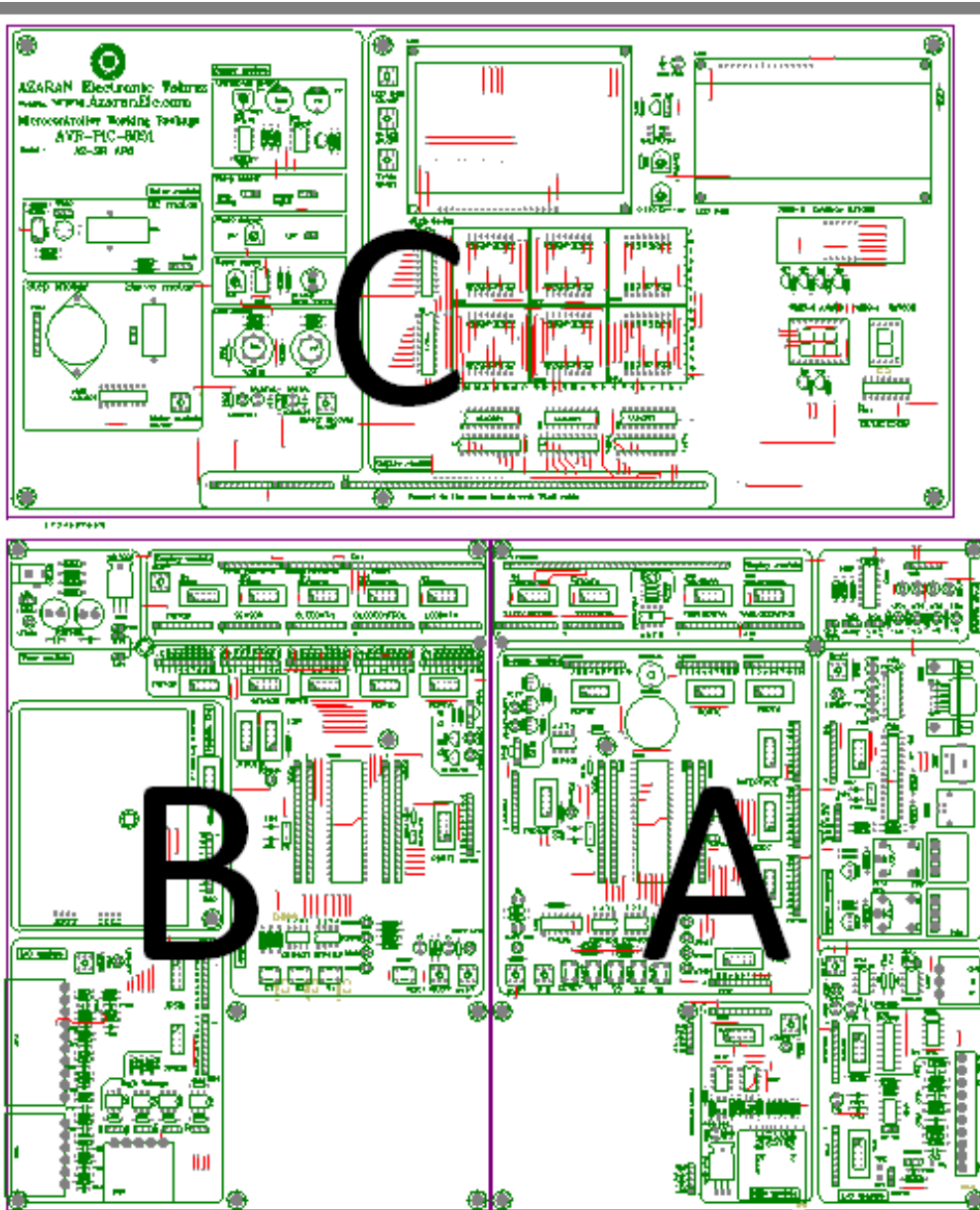
در اولین قدم دانشجو باید توانایی استفاده از مجموعه آموزشی میکروکنترلر مدل AZ-SH AP8 را داشته باشد. این مجموعه به صورت کاملاً حرفه ای طراحی شده است که بیشترین قابلیت را در اجرای انواع آزمایشها دارا میباشد. برای استفاده بهینه از تمام امکانات و قابلیت های این دستگاه دانشجویان باید ابتدا به طور کامل با این مجموعه آشنایی پیدا بکنند. این آزمایش برای این منظور در نظر گرفته شده است.

دانشجویان بعد از انجام این آزمایش و پاسخ به سوالات مربوطه میتوانند به ادامه ی آزمایشات بر روی میکروکنترلر پردازند. این آزمایش برای تمام میکروکنترلر های مشابه میباشد و دانشجویان حتماً باید این آزمایش را انجام دهند. ابتدا برای این که بتوانید آزمایش هایی را که پیش رو دارید بر روی مجموعه آموزشی انجام دهید نکاتی را باید در مورد ساختار این مجموعه ی آموزشی بدانید.

این مجموعه آموزشی از ۳ عدد برد اصلی تشکیل شده است که هر یک میتوانند به صورت مستقل عمل نمایند. عکس زیر نمای این بردها را در کنار یکدیگر نمایش میدهد.



برای سادگی توضیح دادن آزمایش کارها این بردها را نام گذاری میکنیم. برد پایین سمت راست را board A و برد پایین سمت چپ را board B و برد بالایی را board C می نامیم.



این بردها یا به نام های انگلیسی و یا با استفاده از شماره نام گذاری میشوند .

عبارت برد شماره ۱ و برد A هر دو به یک معنی است .

عبارت برد شماره 2 و برد B هر دو به یک معنی است .

عبارت برد شماره 3 و برد C هر دو به یک معنی است .

بر روی هر یک از بردها سخت افزارهای خاصی قرار داده شده است که با استفاده از هر یک میتوانید آزمایشهای خاصی را انجام دهید. برای مثال نمایشگرها کلاً در برد شماره ۳ قرار داده شده است و تجهیزات ارتباطی در برد شماره ۱ قرار داده شده است.

برد های شماره ۱ و شماره ۲ بردهای اصلی هستند چرا که در روی هر یک از آنها یک عدد میکروکنترلر AVR قرار داده شده است. این کار برای توانایی مانور بیشتر در انجام آزمایشها با میکروکنترلر میباشد. برد شماره ۳ نیز برد کمکی برای این بردهای اصلی میباشد. که در روی این برد نیاز به انجام کار خاصی نخواهید داشت. در این برد شما تنها نتایج حاصل از آزمایشات خود را خواهید دید. البته ماجول ابزار دقیق هم در این برد قرار داده شده است که با استفاده از آنها میتوانید آزمایشات مربوط به سیستمهای اندازه گیری و ابزار دقیقی خود را انجام دهید.

ویژگی های مشترک در کل مجموعه :

هر سه برد این مجموعه به صورت ماجول وار طراحی شده است. بدین معنی که تمام ادوات و عناصری که در روی برد قرار داده شده است به صورت گروهی در یک مکان قرار داده شده اند تا بکارگیری و استفاده از آنها راحت تر باشد. به عنوان مثال ماجول ابزار دقیق - ماجول نمایشگر - ماجول اینترفیس (ارتباطی).

ارتباطات بین این ماجولها از طریق استاندارد خاصی برقرار میشود. دو نوع ارتباط برای این ماجولها در نظر گرفته شده است. یکی از طریق کانکتورهای IDC ، ۵*۲ که با استفاده از کابل های رابط FLAT با یکدیگر ارتباط برقرار میکنند. ارتباطاتی که بدین روش بین بردها در نظر گرفته شده است همواره ثابت میباشد و تمام مواردی که در حین کار با انواع ماجولها احتمال وقوع آنها باشد در این ارتباطات پیش بینی شده اند و هیچ گونه نیازی به تغییر در این اتصالات ندارید و میتوانید آزمایشهای خود را به راحتی انجام دهید. اما برای مواقعی که در حالت خاص نیاز به ایجاد اتصالات خاصی داشته باشد میتواند از پین هدرهایی که در کنار هر یک از کانکتورها قرار داده شده است استفاده کنید.

هر یک از ماجولها با استفاده از یک خط سفید از همدیگر جدا شده اند و هیچگونه ارتباطی با ماجولهای دیگر ندارند و حتی برای تغذیه ی هر یک از ماجولها یک کلید ON/OFF قرار داده شده است که در صورتی که بخواهیم از ماجول استفاده کنیم آن را روشن میکنیم در غیر این صورت باید آن ماجول را خاموش کنیم.

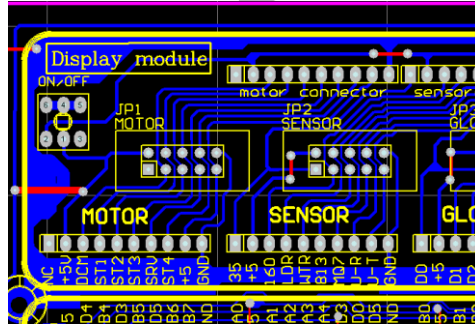
در هر ماجول یک یا چند کانکتور 5*2 IDC قرار دارد که برای ارتباط با ماجول از این کانکتورها استفاده کنید. در داخل ماجول اصلی میکروکنترلر تعداد بیشتری ماجول قرار دارد. تعداد بیشتر کانکتورهای ارتباطی در ماجول اصلی قدرت مانور بیشتری به شما در انجام پروژه های خود داد. در کنار هر کدام از این کانکتورها نام مربوط به آن نوشته شده است که در صورتی که این کانکتور به یک پورت کامل از میکروکنترلر متصل شده باشد نام آن پورت در کنار آن نوشته شده است و در صورتی که پایه های آن برای کاربردهای خاصی طراحی شده باشند نام آن پایه ها در کنار پین هدر مربوط به آن

کانکتور نوشته شده است. اگر هم نام پین ها بر روی مارکاز برد نوشته نشده باشد با رجوع به نقشه های مربوط به مجموعه از اتصالات آنها مطلع شوید.

در مورد برد شماره ۳ باید گفت که این برد هم مانند دو برد دیگر به صورت ماجول وار طراحی شده است اما کانکتور های 5*2 IDC آنها در روی برد های شماره ۱ و ۲ قرار داده شده است. این کانکتور ها در بالای برد و در کنار ماجول های منبع تغذیه ی مربوط به هر دو برد قابل مشاهده هستند. شما برای این که بتوانید از ماجول های برد شماره ۳ استفاده کنید باید با استفاده از این کانکتور ها به پایه های آنها دسترسی پیدا کنید. به عنوان مثال پایه های LCD کاراکتری از طریق کانکتور 5*2 IDC به نام LCDDATA بر روی برد شماره ۲ قابل دسترسی هستند و همین طور بقیه ی عناصر طراحی شده در روی برد شماره ۳ از این طریق در دسترس شما قرار داده شده اند. برای مثال شما کانکتور مربوط به سنسور ها را پیدا کنید؟

بله! همانطور که حدث زدید این کانکتور به نام JP2 – SENSOR در بالای برد شماره ۲ قرار دارد.

کلید روشن و خاموش مربوط به برد شماره ۳ نیز در قسمت کانکتورهای برد شماره ۳ در برد شماره ۲ قرار دارد.



در مواقعی که بخواهید از برد شماره ۳ استفاده کنید بایستی تغذیه ی آن را با روشن کردن این کلید وصل کنید. در گوشه ی بالا سمت چپ ماجول نمایشگر ۳ عدد کلید ON/OFF برای قطع و وصل تغذیه ی مربوط به LCD ها و تابلو روان قرار داده شده است. در مواقعی که نخواهید از LCD و یا تابلو روان استفاده کنید توسط این کلید ها تغذیه ی مربوط به آن ها را قطع کنید.

در قسمت مربوط به 7-SEGMENT ها ۳ نوع نمایشگر هفت قسمتی در نظر گرفته شده است. ۲ تایی – تکی نمایشگر های ۴ تایی و ۲ تایی از طریق کانکتور های روی برد در دسترس شما قرار داده شده است تا بتوانید با استفاده از میکروکنترلر برای آنها برنامه نویسی کنید. اما 7-SEGMENT تکی در قالب یک مبدل BCD27SEG در نظر گرفته شده است. بدین حالت که با استفاده از یک DIP سوئیچ چهار تایی که در بالای برد شماره ۱ در قسمت کانکتور های برد شماره ۳ قرار دارد میتوانید داده های A-B-C-D در مبنای BCD را وارد کرده و معادل آن را در مبنای 7-SEG بر روی نمایشگر 7-SEG تکی مشاهده کنید. این حالت نیاز به هیچ گونه تنظیمات خاصی ندارد و شما می توانید بعد از متصل کردن تغذیه دستگاه این کار را انجام دهید.

در مورد منبع تغذیه هم باید گفت از دو روش میتوانید تغذیه مورد نیاز برای انجام آزمایش هایتان را فراهم کنید.

۱- منبع تغذیه ی اصلی دستگاه ۲- استفاده از پورت USB کامپیوتر

در مواقعی که آزمایشی که قصد انجام آن را دارید یک آزمایش ساده ای است که در آن از کمترین عناصر بهره برداری میکنید و نیز از پروگرامر ISP مربوط به میکروکنترلر AVR استفاده میکنید، میتوانید از همان تغذیه ی پورت USB کامپیوتر که به همراه کابل پروگرامر وارد برد میشود استفاده کنید. به عنوان مثال اگر در برد شماره ۱ و در ماجول اصلی از کانکتور ISP برای برنامه ریزی کردن میکروکنترلر AVR استفاده میکنید دیگر نیازی به روشن کردن کلید ON/OFF کل دستگاه ندارید و اگر هم در حین کار نیازی به استفاده از ماجول های دیگر داشتید (به جز مدار خروجی آنالوگ قرار داده شده ماجول ورودی و خروجی سمت راست، ب ه علت نیاز به تغذیه دابل) می توانید با فشار دادن کلید ON/OFF مربوط به ماجول اصلی تغذیه ی این ماجول را به خط تغذیه انتقال دهید، در این صورت میتوانید با روشن کردن هر یک از ماجول ها با استفاده از کلید ON/OFF تغذیه آنها را نیز تامین کرده و آزمایش خود را انجام دهید.

۲- البته در تمام حالات میتوانید از منبع تغذیه ی خود دستگاه استفاده کنید و مطمئن باشید که هیچ مشکلی از نظر جریان مورد نیاز برای انجام آزمایشات خود نخواهید داشت .

DRAFT

برای کسب اطلاعات بیشتر شما را به مطالعه مطالب زیر دعوت میکنیم :

نحوه راه اندازی و استفاده از مجموعه آموزشی مدل AZ-SH AP8

در این مجموعه تمام ادوات و تجهیزات مورد نیاز برای انجام پروژه های مختلف با میکروکنترلرهای خانواده AVR – PIC – 8051 در نظر گرفته شده است . از جمله آنها می توان به نمایشگرهای مختلف ، ورودی و خروجی های آنالوگ و دیجیتال ، سنسور و موتور و اشاره کرد.

اساس طراحی این مجموعه به گونه ای می باشد که بتوان تمامی آزمایشهای مورد نیاز را بر روی تمامی میکرو کنترلرها و حتی سایر تراشه های سفارشی انجام داد . در این مجموعه روی هر دو برد اصلی قسمتی برای بردهای میکرو در نظر گرفته شده و عناصر و تجهیزات ضروری برای کار میکرو کنترلر در این ماجولها طراحی شده است . برای استفاده از انواع مختلف تراشه ها بوسیله مجموعه ، هسته اصلی بردها (میکروکنترلرها) بصورت قابل تغییر طراحی شده است و در کنار سوکت میکروکنترلرها پین هدرهایی تعبیه گردیده که توسط آنها می توان ماجولهای مربوط به تراشه های PIC , 8051 را به برد اصلی متصل نمود. . بعد از اتصال این ماجولها (به همراه تجهیزات اصلی مجموعه ارائه شده است) از تمام امکانات و تجهیزات مجموعه برای میکروکنترلرهای PIC , 8051 می توان استفاده نمود.

برای کارایی بهتر مجموعه تجهیزات و مدارهای جانبی بصورت ماجولهای مستقل و مجزا از هم طراحی شده اند و پایه های ورودی و خروجی آنها از طریق کانکتورهای استاندارد در دسترس کاربران قرار گرفته است . هر ماجول دارای یک کلید روشن و خاموش اختصاصی بوده و روشن بودن آن بوسیله یک LED مشخص می شود . این ماجولها عبارتند از :

ماجول INTERFACE (واسط کامپیوتری) شامل:

الف (پورت سریال

ب (مبدل USB به سریال

ج) PS2

د) RLY

ماجول I/O ورودی و خروجی شامل:

الف (ورودی آنالوگ و خروجی آنالوگ

ب) ورودی دیجیتال - خروجی دیجیتال

ماجول حافظه I2C MEM – SPI MEM - MMC CARD

ماجول نمایشگر شامل :

الف) نمایشگر LCD گرافیکی

(ب) نمایشگر LCD کاراکتری

(ج) نمایشگر 7-SEG

(د) نمایشگر تابلو روان ۲۴*۱۶ تایی

ماحول ابزار دقیق شامل :

(الف) سنسور دما (دیجیتال و آنالوگ)

(ب) سنسور رطوبت

(پ) سنسور نور

(ج) سنسور گاز

(د) سنسور اولتراسونیک

ماحول کنترل و راه اندازی موتور شامل :

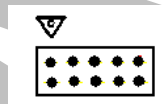
(الف) موتور DC

(ب) STEP MOTOR

(ج) SERVO MOTOR

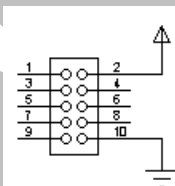
همانگونه قبلاً گفته شد تمام پایه های مربوط به این ماحول ها از طریق باکسهای IDC و نیز پین هدرهای کنار آن در دسترس قرار دارند، در مورد این کانکتورها و برای استفاده از آنها نکات زیر باید در نظر گرفته شود:

۱- این کانکتورها ۱۰ پایه دارند که در ۲ ردیف ۵ تایی چیده شده اند و پایه شماره ۱ توسط یک مثلث به شکل زیر مشخص شده است.



۲- در تمامی کانکتورهای IDC و پین هدرهای (نظامی) مربوطه در روی برد ، پایه شماره ۲ به ولتاژ 5 V + و پایه شماره ۱۰ به زمین (GND) متصل شده است.

۳- در مورد پین هدرهای کنار IDC نیز مورد قبل رعایت شده است. یعنی پایه شماره ۲ به ولتاژ 5 V + و پایه شماره ۱۰ به زمین GND متصل شده است. پایه شماره ۱ از پین هدر نیز توسط عدد ۱ که بر روی فیبر مدار چاپی به رنگ سفید نوشته شده است مشخص شده است.



- ۴- از این کانکتورها و پین هدرها برای اتصال ماجولهای مختلف به یکدیگر استفاده می شود . برای اتصال کانکتورهای 5*2 IDC می توان از کابل های FLAT با اندازه های مختلف متعلق به مجموعه استفاده کرد .
- توضیح : کابل FLAT بصورت مسطح و شامل چندین رشته سیم است ، که در این مورد از نوع ۱۰ رشته ای است که در دو سر آن سوکتهای مادگی 5*2 نصب شده است . (هر مادگی دارای یک زائده می باشد که مانع نصب برعکس سوکت می گردد) ، برای اتصالات بین کانکتورهای IDC استفاده میشود .
- ۵ - از پین هدرها در زمانی که اتصالات جدید علاوه بر اتصالات پیش فرض طراحی شده نیاز باشد استفاده می کنیم که قدرت مانور بیشتری را جهت استفاده کاربران فراهم می کند .
- ۶ - برای آگاهی از اتصالات پین هدرها و IDC ها به نقشه ی شماتیک مربوطه مراجعه کنید .

تذکر مهم :

- * هرگز خروجی های تغذیه را به همدیگر اتصال کوتاه نکنید .
- * ماجول هایی را که از آنها استفاده نمی کنید ، خاموش کنید . این کار را توسط کلید های ON/OFF مختص هر ماجول انجام دهید . (اگر ماجول روشن باشد led مربوطه نیز روشن میشود) .
- * هرگز خروجی های تغذیه را با سیم (یا با هر وسیله ی رسانای دیگری) به نقاط و پایه های مختلف آزاد و در دسترس بردها و مخصوصا " ماجول اصلی میکروکنترلر اتصال ندهید . چرا که باعث سوختن قطعات و عناصر بردها خواهد شد .
- * هرگز قطعات و عناصر نصب شده بر روی برد را به طرفین خم و راست نکنید . چرا که این عمل در دراز مدت باعث میشود که قطعات از پایه بشکنند و دیگر قابل استفاده نخواهد بود .

برای هریک از ماجول ها توضیحاتی در زیر آورده شده است که حتما " قبل از استفاده از آنها این مطالب را به دقت بخوانید . در کل مجموعه ۳ عدد برد بزرگ قرار دارد که از این پس به برد سمت راست برد شماره ۱ و به برد سمت چپ، برد شماره ۲ و به برد بالایی، برد شماره ۳ خواهیم گفت .

ماجول های تعبیه شده در برد ۱

منبع تغذیه مجموعه

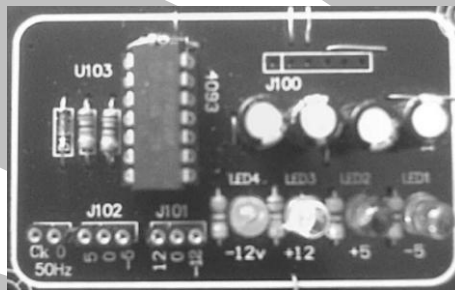
می تواند ولتاژهای متقارن ۵ و ۱۲ ولت را تولید نماید .

(این ولتاژها عبارتند از +12V , +5V , GND , -5V , -12V , CLK) .

برق شهری 220 V از طریق کابل به سوکت تعبیه شده روی کیف متصل می شود ، پس از روشن نمودن کلید ON/OFF مجموعه (در کنار سوکت برق) که روشن یا خاموش بودن منبع تغذیه را تعیین می کند در گوشه سمت راست برد نشانگرهای LED مربوط به ولتاژهای مختلف منبع تغذیه وجود یا عدم وجود این ولتاژها را در مجموعه نشان می دهند . در خروجی منبع تغذیه یک سیگنال موج مربعی با فرکانس 50HZ که از موج سینوسی برق شهر درست شده در نظر گرفته شده است (CLK) . نقشه مدار منبع تغذیه در ضمیمه نقشه ها قرار داده شده است .

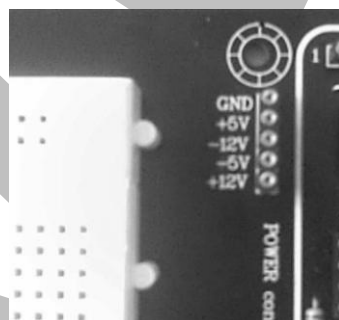
منبع تغذیه دستگاه میتواند جریانی تا حد ۱ آمپر را تامین کند . ولتاژهای مورد نیاز تمام ماجول ها به طور پیش فرض از طریق برد مدار چاپی تامین شده است و شما نیاز به اتصالات جدیدی ندارید . اما برای مصارف خاصی که احتمالاً در پروژ ه های خود با آنها مواجه شوید ولتاژهای فوق در خروجی ماجول منبع تغذیه در دسترس قرار داده شده اند .

عکس : ماجول منبع تغذیه



از طرف دیگر در کنار BREAD BOARD نیز این ولتاژها در دسترس قرار داده شده اند که میتوانید برای استفاده از آنها در روی BREAD BOARD با استفاده از تکه سیم کوچک مخصوص BREAD BOARD آنها را بر روی BREAD BOARD سیم کشی کنید .

عکس : تغذیه برد برد



ماجول اصلی

این ماجول ، ماجول اصلی می باشد که میکرو کنترلر در روی آن قرار دارد . عناصر اصلی برای راه اندازی میکرو کنترلر و نیز تجهیزات لازم برای انجام آزمایشهای اولیه در روی این ماجول قرار داده شده اند . این عناصر عبارتند از کریستال های خارجی ، مدار RESET ، LED ها ، میکروسوییچ ها ، بازر ، پیزو ، فرستنده و گیرنده مادون قرمز و... میباشند که نقشه ی آنها در ضمیمه نقشه ها موجود میباشند .

از طرف دیگر تمام پایه های میکرو کنترلر از طریق پین هدر های دو طرف آرسی در دسترس هستند و چون تمام ارتباطات استاندارد در مجموعه از طریق IDC BOX ها انجام می پذیرد ، کانکتور های ارتباطی مخصوصی برای هر یک از ماجول ها در کنار آن ماجول طراحی شده است .

نقشه ی اتصالات این کانکتور ها در قسمت نقشه ها آورده شده است . با توجه به این که در برقراری ارتباط از طریق کابل FLAT بین کانکتور های IDC BOX ، اتصالات بین پایه های ۱ تا ۱۰ به صورت مستقیم برقرار میشود ، از این رو در کنار این کانکتور ها پین هدر های ۱۰ تایی نیز قرار داده شده این که دقیقاً به همان پایه های IDC ها متصل هستند و در مواقعی که نیاز باشد اتصالاتی غیر از حالت استاندارد انجام دهید از این پین هدر ها استفاده کنید . پایه شماره ۱ از پین هدر ها توسط حرف ۱ در روی مارکاژ برد ها مشخص شده است .

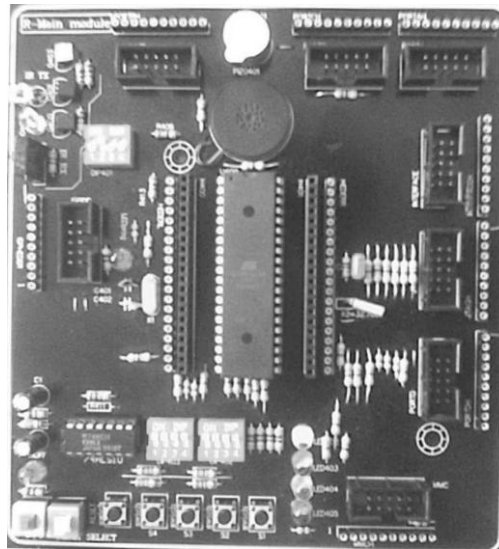
توجه مهم:

نحوه ی استفاده از مدار Reset

با توجه به این که برد های این مجموعه برای میکرو کنترلر هایی با سطح ولتاژ ریست مختلف طراحی شده است لذا یک کلید دوطرفه برای انتخاب حالت ریست میکرو کنترلر در نظر گرفته شده است . مثلاً میکرو کنترلر AVR دارای ریست LOW ACTIVE میباشد اما میکرو کنترلر PIC دارای ریست HIGH ACTIVE میباشد . کلیدی در کنار کلید ON/OFF مربوط به ماجول اصلی قرار دارد و تغییر حالت ریست را بر عهده دارد . اگر این کلید در حالت فشار داده شده قرار داده شود حالت صفر فعال انتخاب میشود و اگر در حالت غیر فعال (بیرون) قرار داشته باشد حالت HIGH ACTIVE فعال میشود .

پس برای میکرو کنترلر AVR کلید را در حالت پایین قرار دهید .

عکس : ماجول اصلی ۱



ماجول اینترفیس

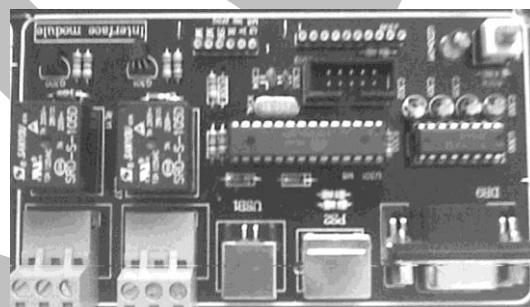
این ماجول شامل پورت‌های ارتباطی برای ایجاد ارتباط بین میکروکنترلر و دستگاه‌های خارجی می‌باشد. این پورت‌ها عبارتند از USB و سریال و ارتباط PS2 و خروجی‌های رله.

برای ایجاد ارتباط با پورت سریال کانکتور مادگی DB9 را از طریق کابل RS232 به کامپیوتر متصل کنید. پایه‌هایی از این کانکتور که باید به میکروکنترلر متصل شوند از طریق کانکتور IDC مربوط به ماجول اینترفیس در دسترس هستند.

برای استفاده راحت‌تر از پورت USB و نیز آشنایی با نحوی ایجاد ارتباط با پورت USB مبدل USB2COM با میکروکنترلر AVR طراحی شده است. پایه‌های TXD, RXD مربوط به این مبدل از طریق پایه‌های IDC BOX در دسترس هستند.

برای استفاده از این ماجول کلید ON/OFF مربوط به آن را فشار دهید. LED مربوط به ماجول روشن می‌شود.

عکس: ماجول اینترفیس



ماجول ورودی خروجی سمت راست

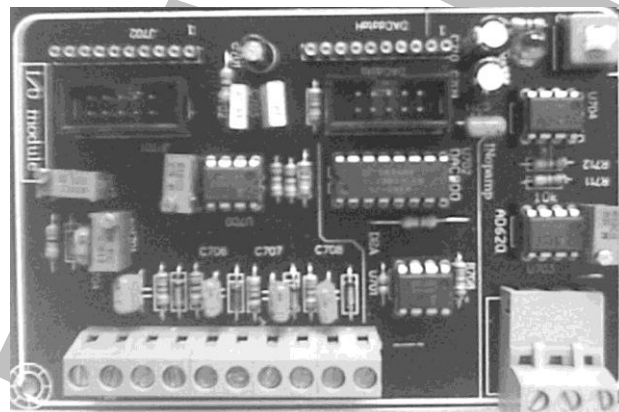
این ماجول که با نام I/O MODULE در روی برد اول مشخص شده است از مدارهای زیر تشکیل شده است:

- روی مدار OP-AMP ابزار دقیق

- خروجی آنالوگ (ورودی دیجیتال ۸ بیتی و خروجی آنالوگ بین ۰ تا ۵ ولت)
- خروجی مدار مبدل سیگنال PWM به ولتاژ آنالوگ
- ورودی و خروجی ساده (دارای مدار محافظ)

برای این که بتوانید از این ماجول استفاده کنید باید آن را روشن کنید . برای این کار از کلید ON/OFF استفاده کنید . در این ماجول ولتاژ های تغذیه ۵ ولت و ۱۲ ولت و -۱۲ ولت استفاده میشود . این ولتاژها (۱۲ و -۱۲) برای تولید خروجی آنالوگ توسط آیسی DAC800 استفاده میشود .

عکس : ماجول I/O MODULE

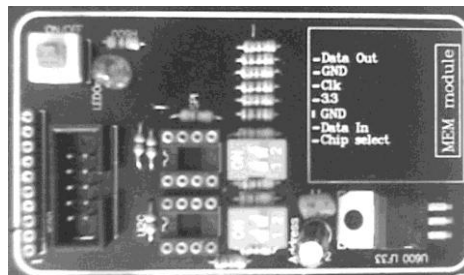


ماجول حافظه

این ماجول سخت افزار لازم برای ارتباط با حافظه های دائمی را فراهم میکند . این حافظه ها عبارتند از حافظه های سری 24CXXX و 25CXXX و نیز مموری کارت MMC CARD . آیسی های سری 24CXXX از پروتکل I2C برای ارتباط با میکرو کنترلر استفاده میکنند و آیسی های سری 25CXXX و MMC CARD ها از پروتکل SPI برای ارتباط با میکرو کنترلر استفاده میکنند . در این ماجول یک سوکت برای قرار گیری آیسی های خانواده 24CXXX (۸ پایه) به نام UI2C در نظر گرفته شده است و یک سوکت برای قرار گیری آیسی های خانواده 25CXXX (۸ پایه) به نام USPI در نظر گرفته شده است . از طرفی دیگر یک سوکت برای جای گیری MMC CARD در زیر برد نصب شده است تا مموری کارت در آن قرار گیرد . بعد از قرار دادن آیسی های فوق در سوکت مربوطه ، کانکتور J600 را به کانکتور MMC که در ماجول اصلی قرار دارد متصل کنید . برای اطلاع از اتصالات بین میکرو و حافظه ی مربوطه به نقشه های برد مراجعه کنید . در روی ماجول مموری دو عدد دیپ سوئیچ قرار داده شده است . دیپ سوئیچ سمت چپی برای تعیین آدرس تراشه UI2C استفاده میشود . (اگر هر دوی کلید های دیپ سوئیچ سمت چپی در بالا قرار داشته باشد آدرس برای تراشه ی قرار داده شده در سوکت UI2C برابر صفر خواهد شد .)

دیپ سوئیچ سمت راستی برای تعیین وضعیت پایه های WP و HOLD به کار میروند . به این ترتیب که کلید ۱ از دیپ سوئیچ سمت چپ برای تعیین وضعیت HOLD و کلید ۲ از آن برای تعیین وضعیت پایه ی WP استفاده میشود .

عکس : ماجول مموری



ماجول های قرار داده شده در برد ۲

ماجول منبع تغذیه

یک منبع تغذیه ساده برای مواقعی که می خواهیم برد ها را از یکدیگر به طور جدا گانه استفاده بکنیم در گوشه سمت چپ برد ۲ قرار داده شده است که باید با استفاده از اتصال یک آداپتور خارجی به جک آداپتور آن ولتاژ مورد نیاز خود را تولید کنید. به یاد داشته باشید که خروجی آیسی رگولاتور ولتاژ به شماره ۷۸۰۵ در این منبع تغذیه به خط تغذیه +۵ کل مدار متصل نیست و این کار را باید شما در مواقعی که به آن نیاز دارید انجام دهید. برای این کار بین سمت راست از کانکتور شماره J202 را به پایه شماره ۲ از یکی از پین هدر های ۱۰ پایه کنار IDC BOX ها متصل کنید. (برای اطلاع از شماره های پین هدر های ۱۰ تایی به قسمت ماجول اصلی در برد ۱ مراجعه کنید).

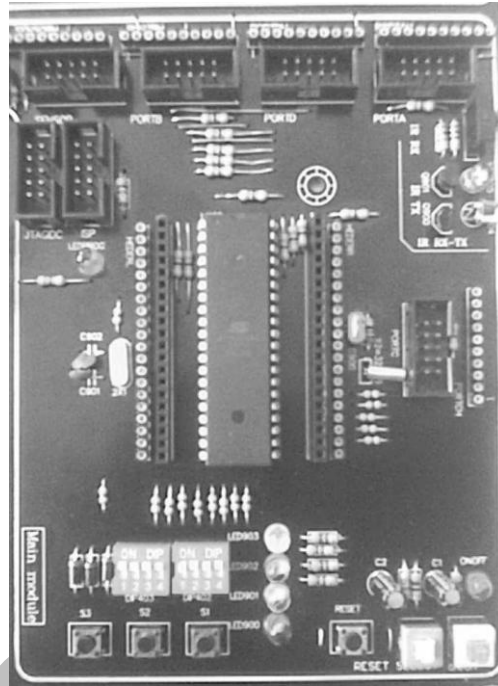
عکس : ماجول منبع تغذیه در برد ۲



ماجول اصلی

در برد دوم نیز همانند برد اول یک ماجول اصلی برای قرار گیری میکروکنترلر تعبیه شده است. همانند برد اول تمامی عناصر مورد نیاز لازم و ضروری برای راه اندازی میکروکنترلر و اجرای آزمایش های اولیه در این ماجول تعبیه شده است. برای اطلاع از نقشه ی این عناصر به نقشه های مربوطه مراجعه کنید. مدار ریست مربوط به ماجول اصلی در برد ۲ نیز همانند ماجول اصلی در برد ۱ میباشد.

عکس : ماجول اصلی در برد ۲



ماجول کیبورد ماتریسی

در این ماجول یک عدد کیبورد 4×4 قرار داده شده است و نحوه ی دسترسی به پایه های کیبورد توسط کانکتور IDC 5×2 و یا از طریق پین هدر 10 تایی میباشد .
برای اطلاع از اتصالات این ماجول به نقشه ی آن مراجعه کنید .

عکس : ماجول کیبورد

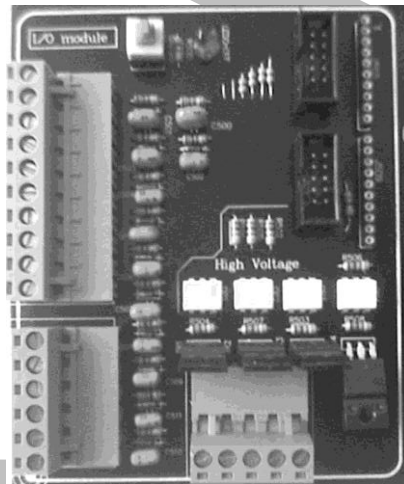


ماجول I/O برد ۲

در روی برد ۲ نیز ماجول ورودی و خروجی در نظر گرفته شده است که دارای ۱۲ عدد ورودی و خروجی محافظت شده ساده و ۴ کانال خروجی قدرت میباشد .

خروجی های قدرت توسط اپتوتریاک و تریاک طراحی شده است. که برای راه اندازی انواع کنتاکتور مورد استفاده قرار میگیرند.

برای اطلاع از نقشه ی اتصالات آنها به ضمیمه ی نقشه ها مراجعه کنید.



عکس: ماجول IO MODULE برد ۲

ماجول های قرار داده شده در برد ۳

ماجول نمایشگر

در ماجول نمایشگر تعداد متنوعی از نمایشگر های رایج در صنعت الکترونیک در نظر گرفته شده اند. عبارتند از نمایشگر های LCD و نمایشگر های 7-SEGMENT و نمایشگر تابلو روان را نام برد.

نمایشگر LCD: دو نوع LCD کاراکتری و LCD گرافیکی در نظر گرفته شده است. بر روی LCD های کاراکتری میتوان اطلاعات مورد نیاز را به صورت حروف انگلیسی و یا هر کاراکتری که بتواند در یک مربع ۵*۷ پیکسلی قرار گیرد به نمایش در آورد. اما این مربع ها از همدیگر فاصله دارند و در بین آنها یک فضای خالی دیده میشود. از این رو LCD های گرافیکی به وجود آمده اند که در روی آنها میتوان هر نوع نوشته و حتی تصویر دلخواهی را به نمایش در آورد. LCD گرافیکی که ما در این برد از آن استفاده کرده ایم از نوع ۱۲۸*۶۴ پیکسلی میباشد. بنابراین می تواند یک عکس و یا یک نوشته با فونت بزرگ را به راحتی در درون خود جای دهد. LCD های گرافیکی از این نظر هیچ محدودیتی ندارند و میتوان هر نوع نوشته ای را در روی آنها به نمایش در آورد.

برای هر یک از LCD های کاراکتری و گرافیکی یک کلید ON/OFF قرار داده شده است که در مواقعی که از LCD ها استفاده میکنید آن را روشن کنید و در زمانی که از آن استفاده نمی کنید LCD را خاموش کنید (تغذیه آن را قطع کنید) تا هم انرژی اضافه مصرف نکند و هم طول عمر آن افزایش یابد. در مورد LCD گرافیکی یک عدد DIP

SWITCH ۲ تایی به نام BACKLIGHT در نظر گرفته شده است که برای روشن شدن چراغ پشت LCD باید از آن استفاده کنید .

در قسمت نمایشگر های هفت قسمتی (7-SEGMENTS) ۳ نوع نمایشگر قرار داده شده است:

نمایشگر ۴ تایی کاتد مشترک

نمایشگر ۲ تایی آنند مشترک

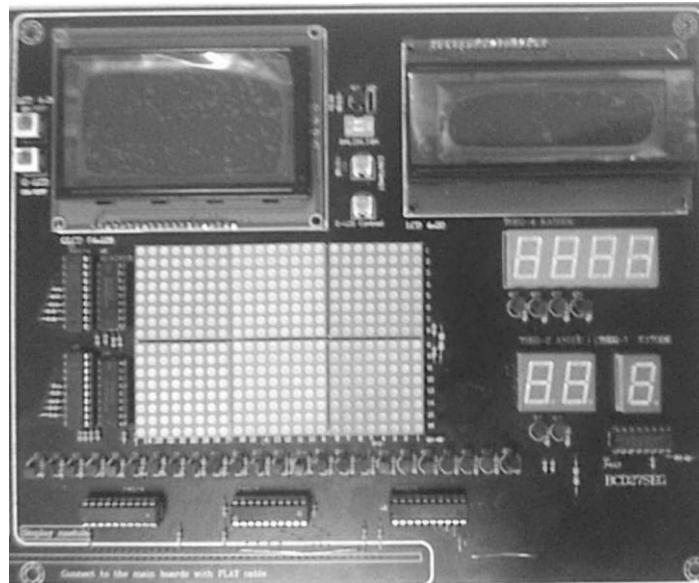
نمایشگر ۱ تایی آنند مشترک

نمایشگر ۱ تایی آنند مشترک برای نمایش اعداد دیجیتالی که در مبنای باینری توسط دیپ سوئیچ تعبیه شده در قسمت BCD انتخاب میشوند به کار میرود . وظیفه ی تبدیل این کدهای باینری به کدهای 7-SEGMENT توسط آی سی شماره ۷۴۴۷ انجام میپذیرد .

در قسمت تابلو روان نیز یک تابلو روان بزرگ در نظر گرفته شده است تا هر برنامه نویسی را راضی کند . ابعاد این تابلو روان ۱۶*۲۴ می باشد که داده های مربوط به سطر و ستون آنها توسط آی سی های ۷۴۵۷۴ تامین میشود . همان طور که میدانید آی سی های ۷۴۵۷۴ آی سی های LATCH (نگهدارنده) میباشند . یعنی اگر در ورودی DATA آنها یک داده قرار داده شود و سپس یک پالس با لبه بالارونده به ورودی CLOCK آن اعمال شود داده ورودی را در خروجی قرار داده و آن را نگه میدارد و هرگز اطلاعات در خروجی آن تغییر نمی کند حتی تغییری در داده های ورودی روی دهد و یا داده ها از ورودی برداشته شود . مگر آن که یک بار دیگر یک پالس فعال ساز به ورودی CLOCK آن اعمال شود .

از این خاصیت این تراشه استفاده شده است تا بتوانیم تابلو روانی را طراحی کنیم تا دارای کمترین خط داده و کنترلی باشد . به عنوان مثال در این تابلو روان تنها تعداد ۸ خط داده و ۵ خط کنترل لازم داریم و بس . در صورتی که اگر از روشهای معمولی برای دسترسی به سطر و ستون های تابلو روان استفاده میکردیم به تعداد ۱۶ خط داده و ۲۴ خط کنترلی ستون ها نیاز داشتیم .

پایه های این نمایشگر ها از طریق کابل FLAT به برد اصلی انتقال یافته و در روی آن برد از طریق کانکتور های IDC و به همراه پین هدر های ۱۰ تایی در دسترس هستند . برای اطلاع از نحوه ی این اتصالات به نقشه های مربوطه مراجعه کنید .



عکس : ماجول نمایشگر

ماجول ابزار دقیق

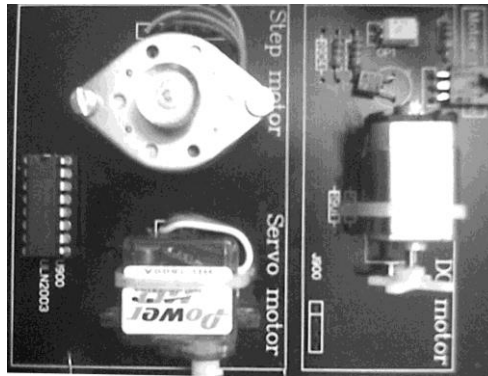
در ماجول ابزار دقیق تعداد ۵ نوع سنسور پر مصرف در نظر گرفته شده است. این سنسور ها عبارتند از سنسور التراسونیک - سنسور حرارت به دو مد آنالوگ و دیجیتال - سنسور رطوبت - سنسور نور - سنسور گاز (دو نوع گاز) بعضی از این سنسور ها در کنار خود دارای یک پتانسیومتر میباشند که قابلیت تنظیم مقادیر داده خروجی را به ما میدهند . نقشه ی این ماجول در ضمیمه ی نقشه های بردها آورده شده است .



ماجول عملگر

در ماجول عملگر ۳ نوع موتور رایج و پر کاربرد در نظر گرفته شده است. این موتور ها عبارتند از , DC-MOTOR , STEP MOTOR , SERVO MOTOR در ورودی این موتور ها بافر های مربوط به هر کدام قرار داده شده است . بافر ULN2003 برای STEP MOTOR و SERVO MOTOR در نظر گرفته شده و مدار راه انداز مربوط به DC MOTOR توسط اپتو کوپلر طراحی شده است .

عکس : ماجول موتور



برای هر سه ماجول نمایشگر و ابزار دقیق و عملگر یک کلید ON/OFF در نظر گرفته شده است که برای این که بتوانید از این ماجول ها استفاده کنید آن را روشن کنید. LED روی برد ۳ نمایشگر وجود و یا عدم وجود تغذیه در برد ۳ است.

نحوه ی راه اندازی و استفاده از پروگرامر سه کاره AVR

به همراه مجموعه یک دستگاه سه کاره برای پروگرامر و دیباگر میکروکنترلر AVR ارائه شده است که شامل یک پروگرامر USB ASP برای پروگرام کردن AVR میباشد که از پورت USB برای اتصال به کامپیوتر استفاده میکند و یک دستگاه JTAG ICE که برای پروگرام کردن و DEBUG کردن میکروکنترلر AVR استفاده می شود این دستگاه از پورت SERIAL برای ارتباط با کامپیوتر استفاده میکند و یک دستگاه پروگرامر STK200/300 که برای پروگرام کردن میکروکنترلر AVR به کار میرود. این پروگرامر از پورت PARALLEL برای ارتباط با کامپیوتر استفاده میکند. برای راه اندازی و استفاده از پروگرامر توضیحات زیر را به دقت مطالعه کنید:

پروگرامر موازی (STK 200/300)

اولین قسمت از دستگاه پروگرامر، پروگرامر STK 200/300 است که از پورت موازی (LPT) کامپیوتر استفاده می کند و مدار تشکیل دهنده آن یک آی سی بافر با شماره 74HC244 است و نقشه مداری آن نیز ارائه شده است. خروجی پین های پروگرامر از طریق کانکتور 5*2 IDC قابل دسترس می باشد. و جهت برنامه ریزی میکرو بایستی این کانکتور را بوسیله کابل FLAT به کانکتور 5*2 IDC پروگرامر، که در کنار میکرو قرار دارد متصل کرد. این کانکتور به نام ISP IDC مشخص شده است. این پروگرامر برای میکروکنترلر های AVR میباشد. و توسط تمام نرم افزار های پروگرامر پشتیبانی میشود.

برای شناسایی این پروگرامر توسط نرم افزار باید در قسمت مربوط به تنظیمات PEGRAMMER گزینه ی AVR STK200/300 را انتخاب کنید.

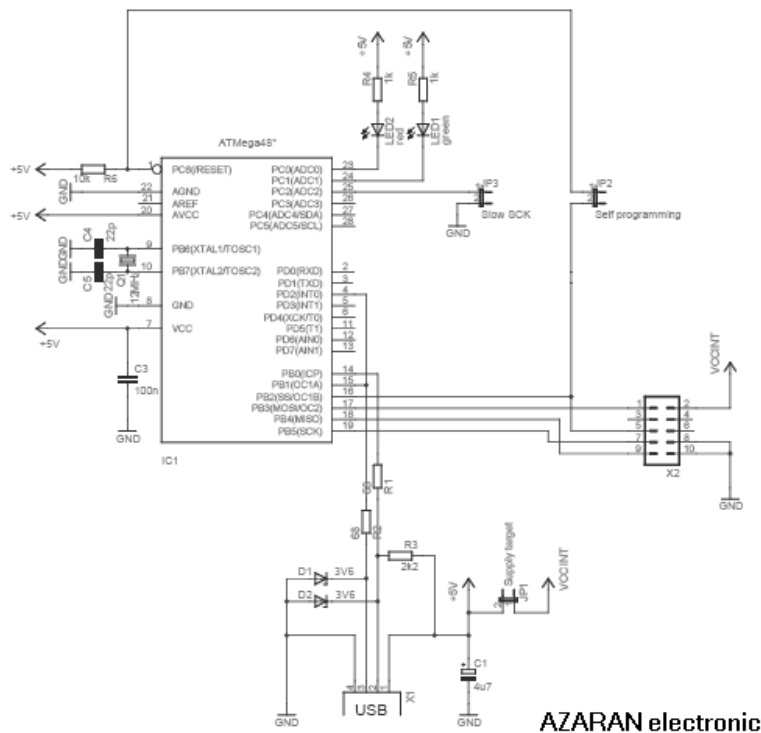
JTAG ICE ((پروگرامر از طریق پورت سریال - دیباگر از طریق پورت سریال))

دومین قسمت از دستگاه پروگرامر JTAGICE میباشد که برای تحلیل و بررسی برنامه نوشته شده به همراه اجرا روی میکرو میباشد برای استفاده از این قسمت از دستگاه به ضمیمه JTAG ICE مراجعه کنید.

پروگرامر USB

سومین قسمت دستگاه یعنی پروگرامر USB-ASP همانند پروگرامر STK 200/300 دارای خروجی ISP می باشد. خروجی پروگرامر USB-ASP از طریق کانکتور 5*2 IDC در دسترس می باشد. برای برنامه ریزی میکرو در این بخش نیز همانند مورد قبل باید کانکتور آن را بوسیله کابل FLAT به کانکتور کنار میکرو متصل کرد. نقشه پروگرامر USB-ASP در زیر آورده شده است.

پروگرامر USB-ISP دارای یک برنامه اختصاصی میباشد که باید با استفاده از آن میکروکنترلر را پروگرامر کنید. نام این نرم افزار PROGISP.EXE میباشد که یک نسخه از آن در CD ارائه شده به همراه مجموعه قرار دارد. برای راه اندازی و استفاده از پروگرامر ISP مطالب زیر را به دقت مطالعه کنید.

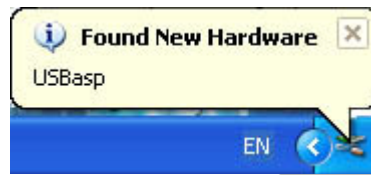


این پروگرامر یکی از جدیدترین انواع پروگرامرهای USB می باشد که سرعت آن بالا بوده و سخت افزار ساده ای دارد. در این مقاله نحوه ی نصب درایور پروگرامر و چگونگی کد نویسی در بسکام و کدویژن و راهنمای PROGISP ارائه شده است:

مراحل نصب پروگرامر USB در ویندوز XP

پروگرامر را به پورت USB کامپیوتر یا لپ تاپ خود متصل کنید.

شکل زیر در گوشه سمت راست صفحه نمایش مشاهده خواهد شد.



اگر با پیغام خطای زیر مواجه شدید به قسمت عیب یابی مراجعه کنید.



در مرحله ی بعد طبق شکل عمل کرده کلید NEXT را بزنید.

CD مربوط به دستگاه رادرون CD ROM قرار داده و گزینه INSTALL THE SOFTWARE را انتخاب کرده و NEXT را کلیک کنید تا ویندوز به صورت خودکار اقدام به نصب پروگرامر کند. و یا گزینه INSTALL FROM LIST OR SPECIFIC LOCATION را انتخاب کنید و در پنجره ی بعد مسیر درایور پروگرامر را وارد کرده و آن را نصب نمایید.





با نمایش پنجره ی بالا پروگرامر به طور کامل نصب میشود .

مراحل نصب پروگرامر در ویندوز vista

پروگرامر را به بورت USB کامپیوتر و یا لب تاب خود متصل کنید . به مسیر CONTROL PANEL\SYSTEM\DEVICE MANAGER رفته و در قسمت OTHER DEVICE بر روی USBASP راست کلیک کرده و گزینه ی UPDATE DRIVER SOFTWARE را کلیک کنید .



در پنجره ی بعدی در قسمت SEARCH FOR DRIVER SOFTWARE درایور CD-ROM را انتخاب کرده و CD دستگاه را درون CD-ROM قرار داده و بر روی NEXT کلیک کنید . در پنجره ی بعدی گزینه ی INSTALL THIS DRIVER SOFTWARE ANYWAY را انتخاب کرده و در آخر بر روی CLOSE کلیک کنید تا پروگرامر به طور کامل نصب شود . با نصب پروگرامر بر روی کامپیوتر خود حالا نحوه ی ایجاد فایل HEX و ریختن آن در میکرو را توضیح میدهیم .

راهنمای استفاده از پروگرامر

دو عدد جامپر بر روی پروگرامر وجود دارد که عملکرد آنها به این صورت است :

حالت قطع جامپر ها در سمت بالاست (به طرف پورت USB) و حالت وصل و یا فعال جامپر ها در سمت پایین . یعنی اگر جامپر را قطع کنید باید جامپر ۲ پایه را به پین های پایینی از پین هدر ۳ پین مربوطه متصل کنید .

کلید SLOW CLOCK :

نکته : این کلید دارای دو حالت FAST CLOCK و SLOW CLOCK میباشد .

اگر جامپر وصل باشد یعنی حالت SLOW CLOCK فعال است . و اگر غیر فعال باشد و یا جامپر را برداشته باشید یعنی حالت FAST CLOCK فعال است . اگر این کلید در حالت SLOW CLOCK باشد شما میتوانید میکروکنترلر را با هر فرکانس پالس ساعتی پرو گرم کنید ولی سرعت پرو گرم کردن میکرو در این حالت سریع نیست .

اگر این جامپر در حالت قطع باشد (FAST CLOCK) شما فقط میتوانید میکرو کنترلر هایی را که فرکانس پالس ساعت آنها بیشتر از 1,5 مگا هرتز تنظیم شده باشد برنامه زری کنید . در این حالت سرعت پرو گرم کردن میکرو بسیار سریع خواهد بود .

جامپر M8-R :

در موقع استفاده از پروگرامر باید قطع باشد و در صورتی که بخواهید برنامه میکروی داخل پروگرامر را UPDATE کنید حتما " باید این جامپر وصل باشد .

خلاصه ی مطالب

پس برای استفاده از سرعت بالای برنامه ریزی این پروگرامر باید فیوز بیت های مربوط به کریستال بالا تر از ۱.۵ مگا هرتز قرار داده شود و کلید کلاک را بر روی FAST CLOCK قرار دهید .

حال اگر مطالب را متوجه شده اید در قسمت بعد چگونگی استفاده از نرم افزار قدرتمند PROGISP و چگونگی تغییر فیوز بیتها با این نرم افزار گفته خواهد شد .

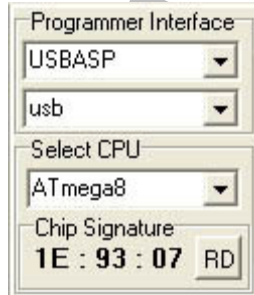
راهنمای استفاده از نرم افزار progisp

- بعد از نصب پروگرامر و نرم افزار PROGISP 1.6.6 , PROGISP را اجرا کنید .

- میکروکنترلر را با استفاده از کابل FLAT به پروگرامر متصل کنید . برای این کار کانکتور 5*2 IDC در روی ماجول میکروکنترلر و در کنار میکرو کنترلر که به نام ISPIC مشخص شده است را به کانکتور IDC خروجی پروگرامر متصل کنید .

- کابل USB مربوط به پروگرامر را به کامپیوتر متصل کنید .

طبق شکل بعد در قسمت PROGRAMMER INTERFACE گزینه ی USBASP و USB را انتخاب کرده و در قسمت SELECT CPU نوع میکروکنترلر خود را انتخاب کنید .



کلید (جامپر) کلاک بر روی پروگرامر را در حالت SLOW CLOCK قرار دهید .

کلید ERASE رازده تا حافظه فلش داخل میکروکنترلر پاک شود .

اگر با پیغام خطای ERROR : COULD NOT FIND USB ASP DEVICE برخورد کردید بدین معنی است که پروگرامر توسط برنامه شناسایی نمی شود و از قرار گرفتن پروگرامر در درون پورت USB و همچنین درست نصب کردن درایور پروگرامر مطمئن شوید .

اگر باز هم با این خطا مواجه شدید به قسمت عیبیابی مراجعه کنید .

بعد از مراحل بالا و بازدن دکمه ی ERASE در گوشه ی سمت چپ برنامه عبارت CHIP ERASE SUCCESSFULLY با رنگ آبی نمایش داده میشود و همزمان ۲ LED سبز رنگ روی پروگرامر و LED قرمز رنگ روی برد میکرو یک لحظه روشن خواهند شد .

حال طبق شکل زیر ، روی دکمه ی READ کلیک کرده تا فیوز بیت میکرو خوانده شود .



گزینه ی DEFAULT فیوز بیتها ی پیش فرض میکروکنترلر را در نظر میگیرد .

گزینه ی WRITE پس از تغییر فیوز بیتها آنها را به میکروکنترلر اعمال میکند .

بعد از زدن کلید READ فیوز بیتهای CKSEL3..0 را در قسمت LOW FUSE BITS بخوانید .

در این حالت با یکی از اشکال زیر مواجه خواهید شد :

(۱)



این حالت ، حالت پیش فرض میکروکنترلر میباشد و بیانگر این است که فرکانس کلاک کاری میکرو بر روی نوسان ساز RC کالیبره شده داخلی و با مقدار 1MHZ میباشد .

(۲)

0	CKSEL3
0	CKSEL2
1	CKSEL1
0	CKSELO

در این حالت فرکانس کاری میکرو بر روی نوسانساز RC کالیبره شده داخلی و با مقدار 2MHz میباشد. چون فرکانس بالای 1.5MHz میباشد شما برای برنامه ریزی میکرو کنترلر میتوانید کلید مربوط به کلاک رابر روی FAST CLOCK قرار دهید تا میکرو با سرعت بالا پروگرام شود.

(۳)

0	CKSEL3
0	CKSEL2
1	CKSEL1
1	CKSELO

در این حالت فرکانس کاری میکرو بر روی نوسانساز RC کالیبره شده داخلی و با مقدار 4MHz میباشد. چون فرکانس بالای 1.5MHz میباشد شما برای برنامه ریزی میکرو کنترلر میتوانید کلید مربوط به کلاک رابر روی FAST CLOCK قرار دهید تا میکرو با سرعت بالا پروگرام شود.

(۴)

0	CKSEL3
1	CKSEL2
0	CKSEL1
0	CKSELO

در این حالت فرکانس کاری میکرو بر روی نوسانساز RC کالیبره شده داخلی و با مقدار 5MHz میباشد. چون فرکانس بالای 1.5MHz میباشد شما برای برنامه ریزی میکرو کنترلر میتوانید کلید مربوط به کلاک رابر روی FAST CLOCK قرار دهید تا میکرو با سرعت بالا پروگرام شود.

(۵)

1	CKSEL3
1	CKSEL2
1	CKSEL1
1	CKSELO

در این حالت فرکانس کلاک کاری میکرو بر روی نوسانساز کریستالی خارجی با فرکانس بالا تنظیم شده است. بدین معنی که هر کریستالی که بیشتر از 1.5MHz است را میتوانید بین پایه های XTAL1, XTAL2 میکرو قرار دهید، با آن فرکانس کلاک کار خواهد کرد. چون فرکانس بالای 1.5MHz میباشد شما برای برنامه ریزی میکرو کنترلر میتوانید کلید مربوط به کلاک رابر روی FAST CLOCK قرار دهید تا میکرو با سرعت بالا پروگرام شود.

(۶)

1	CKSEL3
0	CKSEL2
0	CKSEL1
1	CKSEL0

در این حالت فرکانس کلاک کاری میکرو بر روی نوسان ساز کریستال خارجی با فرکانس پایین تنظیم شده است. بدین معنی که هر کریستالی که کمتر از 1.5MHZ است را میتوانید بین پایه های XTAL1 , XTAL2 میکرو قرار دهید و میکرو با آن فرکانس کلاک کار خواهد کرد. این حالت کلاک برای اتصال کریستال مخصوص ساعت با فرکانس 32.768KHZ در نظر گرفته شده است. در این حالت چون فرکانس زیر 1.5MHZ است برای پروگرم کردن میکرو باید جامگر مربوط به کلاک بر روی SLOW CLOCK قرار بگیرد. در غیر این صورت میکرو پروگرم نمیشود. حال فیوز بیت میکروی شما هر کدام از این ۶ مورد باشد میتوانید با کلیک بر روی صفر و تغییر آن به یک یا بالعکس یک منع یا اندازه فرکانس پالس ساعت میکرو را تغییر داده و با کلیک کردن بر روی W آن تغییر را به میکرو اعمال کنید.

حال نوبت به ریختن فایل هگز در حافظه ی فلش میکرو میرسد. بعد از نوشتن برنامه و کامپایل آن و تولید فایل با پسوند HEX با کلیک بر روی کلید LOAD FLASH TEST.HEX را انتخاب کرده و از منوی COMMAND گزینه ی WRITE FLASH را کلیک کنید بعد از این انتخاب در قسمت پایین نرم افزار پروسه ی پروگرم کردن میکرو نشان داده میشود و عبارت FLASH WRIT SUCCESSFULLY نمایش داده میشود.

برنامه ریزی حافظه EEPROM هم شبیه به همین مرحله است. اگر برنامه ریزی این مرحله کمی طول کشید طبق مطالب قبل به علت روشن بودن کلید در حالت SLOW CLOCK و زیر 1.5MHZ بودن فرکانس کاری میکرو کنترلر میباشد.

نکات مهم مربوط به پروگرامر

در موقع ارسال اطلاعات از طریق کامپیوتر به پروگرامر LED3 بر روی پروگرامر روشن خواهد شد و نشان دهنده ی وضعیت پروگرامر میباشد .

CPU در داخل پروگرامر قابلیت آپدیت شدن دارد .

توجه مهم :

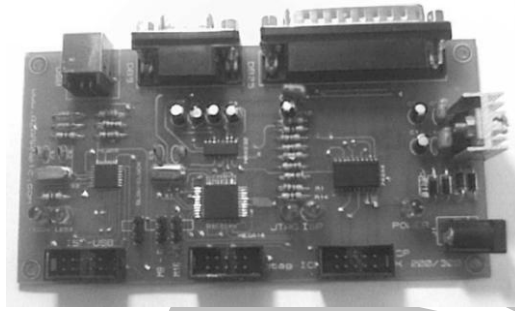
در این مدار چون از تغذیه USB استفاده میکنید ، دقت کنید اتصال کوتاه در مدار رخ ندهد .

عیب یابی :

اگر میکروکنترلر به هر دلیلی پروگرامر نمیشود و یا با کار با نرم افزار USBASP به مشکل برخورد ه اید موارد زیر را چندین بار و به دقت بررسی کنید :

- ۱ - بعد از نصب پروگرامر مجدداً " کامپیوتر خود را راه اندازی کنید ؛
- ۲ - در قسمت PROGRAMMER INTERFACE در نرم افزار PROGISP حتماً USB و USBASP انتخاب شده باشند ؛
- ۳ - سعی کنید پروگرامر را به USB های پشت کیس و بدون کابل کمکی وصل کنید ؛
- ۴ - دقت کنید میکروکنترلری را که میخواهید پروگرامر کنید جزء لیست باشد ؛
- ۵ - از سالم بودن میکروکنترلر مطمئن شوید ؛
- ۶ - فعال بودن یکی از حالات کلاک RC داخلی یا کریستال خارجی فرکانس بالا یا فرکانس پایین ؛
- ۷ - یکسان بودن نام میکروکنترلر در برنامه با میکروکنترلر روی برد ؛
- ۸ - سالم بودن پورت USB کامپیوتر ؛
- ۹ - از نصب بودن درایور پروگرامر بر روی ویندوز اطمینان حاصل نمایید و طبق شکل صفحات قبل مراحل نصب را انجام داده باشید ؛
- ۱۰ - در صورتی که پروگرامر توسط برنامه شناسایی نمیشود یک بار دیگر قرار گیری پروگرامر را درون پورت USB بررسی کرده و مطمئن شوید که LED قرمز پروگرامر روشن است (LED4) و همچنین از سالم بودن میکروکنترلر داخل پروگرامر مطمئن شوید . برای این کار با استفاده از پروگرامر STK200/300 میکرووی مربوط به USBASP را شناسایی کنید . اگر نرم افزار (BASCOM) میکرو را شناسایی کرد بدین معنی است که میکروکنترلر درون پروگرامر سالم میباشد . (البته برای انجام این کار باید جامپر مربوط به RESET را برای آی سی ATMEGA8 فعال کنید برای این کار جامپر را طبق شکل زیر قرار دهید .)

عکس: پروگرامر



AZARAN

جواب سوال در مورد مجموعه ی آموزشی (دانشجویان باید به این سوالات پاسخ دهند) :

دانشجویان باید بعد از مطالعه ی مطالب بالا بتوانند به سوالات زیر پاسخ بدهند :

۱- ماجول منبع تغذیه دستگاه را پیدا کنید ؟

جواب : در گوشه سمت راست بالای برد شماره ۱ قرار دارد که بر روی آن تعداد ۴ عدد LED همواره وضعیت ولتاژ های مختلف را نمایش میدهد و اگر هر یک از ولتاژ ها قطع شود با استفاده از LED مربوطه به این مسئله پی خواهیم برد .

۲- ماجول منبع تغذیه این دستگاه چند نوع ولتاژ میتواند تولید کند ؟

این ماجول با استفاده از یک ترانس سر وسط میتواند ولتاژ های متقارن ۵ و ۱۲ را تولید کند . یعنی میتواند ولتاژ -۵ و ۵+ و ۱۲- و ۱۲+ را تولید کند . در ضمن در خروجی این ماجول فرکانس 50HZ برق شهر نیز قرار دارد . دامنه ی این سیگنال ۵ ولت میباشد . از این سیگنال در بسیاری از موارد و پروژه ها میتواند کاربرد داشته باشد . به عنوان مثال برنامه نویسی یک فرکانس کانتر و

۳- میکروکنترلر های مورد استفاده در این مجموعه از کدام ولتاژ استفاده خواهند کرد ؟

همان طور که میدانید میکروکنترلر های خانواده AVR – PIC – 8051 از ولتاژ ۵ ولت برای تغذیه استفاده میکنند . البته میکروکنترلر های AVR سری L میتوانند با ولتاژ های کمتر نیز کار بکنند .

۴- آیا میتوان از ولتاژ ۱۲ ولت برای تغذیه ی میکروکنترلر استفاده کرد ؟ چرا ؟

خیر - زیرا باعث سوختن و از بین رفتن میکرو میشود .

۵- کاربرد ولتاژ های متقارن چیست ؟

برای تغذیه ی Op-Amp های قرار داده شده در بخش های مختلف مدار و نیز تغذیه آرسی DAC800 و نیز برای پروژه هایی که شما بر روی برد برد میندید ، در اختیار قرار داده شده است تا محدودیتی در این رابطه نداشته باشید .

۶- تعداد ماجول های قرار داده شده در روی بردهای اصلی مجموعه چند تا است ؟

۱۴ عدد

۷- ماجول های قرار داده شده در روی برد شماره ۱ (برد A) کدام ها هستند ؟

ماجول منبع تغذیه - ماجول اینترفیس - ماجول ورودی خروجی سمت راست - ماجول اصلی میکروکنترلر -

ماجول مموری کارت - بخشی از قسمت کانکتور های ماجول display

۸- ماجول ابزار دقیق را بر روی مجموعه پیدا کنید ؟

در برد شماره ۳ (برد بالایی) - وسط

۹- بر روی ماجول ابزار دقیق چند نوع سنسور حرارتی نصب شده است ؟ آنها را پیدا کرده و بگویید از

کدام نوع هستند و چرا از آن نوع سنسور ها استفاده شده است ؟

۲ نوع - دیجیتال و آنالوگ - به خاطر عمومیت این سنسور ها و کاربرد فراوان این سنسور ها .

۱۰- چند نوع عملگر بر روی مجموعه قرار دارد ؟ نام ببرید و در مورد تفاوت های آنها با یکدیگر و نیز کاربر

های هر کدام با دوستانان صحبت کنید .

۳ نوع موتور - DC , step motor , servo motor

۱۱- به نظر شما در این برد از چند روش برای ارتباط بین ماجول های مختلف استفاده میشود ؟ دلیل

استفاده از هر یک را توضیح دهید :

از ۲ روش - یکی ارتباط از طریق کانکتورهای IDC و کابل های Flat - دیگری از طریق ارتباط با استفاده از پین

هدر ها .

در روش استفاده از باکس هدر ها ما همواره از یک سری از ارتباطات ثابت استفاده خواهیم کرد که خود دارای

محاسنی از قبیل

راحتی استفاده - کاهش خطای انسانی - سرعت بیشتر در اجرای پروژه

روش استفاده از پین هدر ها نیز برای این هدف در مجموعه در نظر گرفته شده است که در مواقعی که بخواهیم اتصالات خاصی ، به غیر از اتصالات پیش فرض بین عناصر مختلف برقرار کنیم از این روش استفاده کنیم .

۱۲- ماجول اصلی در برد شماره ۱ را پیدا کنید و بگویید که برای پروگرام کردن میکروکنترلر AVR از کدام سوکت استفاده میشود ؟

از سوکتی که به نام ISPIDC مشخص شده است .

۱۳- کانکتور PORTA بر روی ماجول اصلی را پیدا کرده و بگویید که این پورت چگونه به میکروکنترلر متصل میباشد و از این کانکتور در کجا میتوان استفاده کرد ؟

نحوه ی اتصال این کانکتور به پورت میکرو به این شکل است

شماره پایه	نام پایه از
کانکتور	میکروکنترلر
۱	PORTA.0
۲	VCC
۳	PORTA.1
۴	PORTA.2
۵	PORTA.3
۶	PORTA.4
۷	PORTA.5

PORTA.6 ۸

PORTA.7 ۹

GND ۱۰

از این کانکتور و از این پورت برای اتصال تمامی ماجول ها میتوان استفاده کرد اما با توجه به این که این کانکتور به پورت A متصل شده ، بیشتر در ماجول هایی که با ولتاژ آنالوگ سرو کار دارند استفاده میشود .

۱۴- فرض کنید میخواهیم از LED های قرار داده شده در برد شماره ۱ استفاده کنیم . چه کاری را باید انجام دهیم ؟

به نقشه مراجعه میکنیم . در این نقشه میبینیم که LED ها با استفاده از Dip-switch به میکرو کنترلر متصل شده اند . پس باید برای استفاده از LED ها این Dip-switch را در وضعیت ON قرار دهیم .

۱۵- فرض کنید بخواهیم میکروکنترلر را با استفاده از واسط RS232 به دنیای خارج ارتباط دهیم . از کدام ماجول استفاده خواهیم کرد ؟ مراحل کار را شرح دهید .

از ماجول اینترفیس - ابتدا کانکتور ۲*۵ قرار داده شده در ماجول اینترفیس را به کانکتوری که با همین نام در ماجول اصلی برد شماره ۱ قرار دارد متصل کرده و سپس کلید on/off هر دو برد را فشار میدهیم تا ولتاژ تغذیه ی آنها تامین شود .

۱۶- LED قرار داده شده در ماجول مربوط به I/O سمت راست (در برد شماره ۱) روشن نیست . این به چه معنی است ؟

۱- یعنی کلید مربوط به روشن و خاموش آن در حالت روشن قرار ندارد .

۲- اگر کلید را در حالت روشن قرار دادید (در حالت پایین) ولی باز هم چراغ روشن نیست یعنی این که تغذیه وصل نیست .

۳- ممکن است که چراغ سایر ماجول ها روشن میشود اما فقط چراغ ماجول ورودی/خروجی سمت راست روشن نمیشود:

در این حالت شما در حال استفاده از تغذیه USB میباشد. با توجه به این که تغذیه پورت USB ، 5 ولت میباشد ولی تغذیه ماجول ورودی / خروجی شمت راست ۱۲ ولت میباشد ، بنابراین واضح است که برای این که بتوانید از ماجول ورودی / خروجی سمت راست در برد شماره ۱ استفاده کنید باید از تغذیه ی خود دستگاه استفاده کنید .

۱۷- فرض کنید میخواهیم از LCD کاراکتری قرار داده شده در مجموعه استفاده کنیم و اطلاعاتی را بر روی آن به نمایش در آوریم . چه کارهایی باید انجام دهیم ؟

۱- ابتدا باید به نقشه مراجعه کنیم :

در نقشه نشان داده شده است که پایه های LCD توسط کانکتور IDC 2*5 در اختیار کاربر قرار گرفته است . در این زمان شما باید نحوه ی برقراری ارتباط با LCD را بلد باشید . همانطور که میدانید LCD توسط ۴ بیت داده و ۳ بیت کنترل با میکرو کنترلر ارتباط برقرار میکند . حال شما با استفاده از یکی از کابل های FLAT قرار داده شده در کنار تجهیزات مجموعه ی آموزشی میتوانید این کانکتور IDC 2*5 را به یکی از کانکتور های IDC 2*5 قرار داده شده در ماجول اصلی بردهای شماره ۱ و یا برد شماره ۲ متصل کنید . بعد از این اتصال پایه های LCD شما به پایه های میکرو کنترلر متصل میشود . فرض کنید این کانکتور را به پورت A در میکرو کنترلر AVR اتصال داده اید . نقشه ی اتصالات بین میکرو و LCD به این صورت خواهد بود .

بیت RS : پورت A.0

بیت RW : پورت A.1

بیت E : پورت A.2

بیت DB4 : پورت A.3

بیت DB5 : پورت A.4

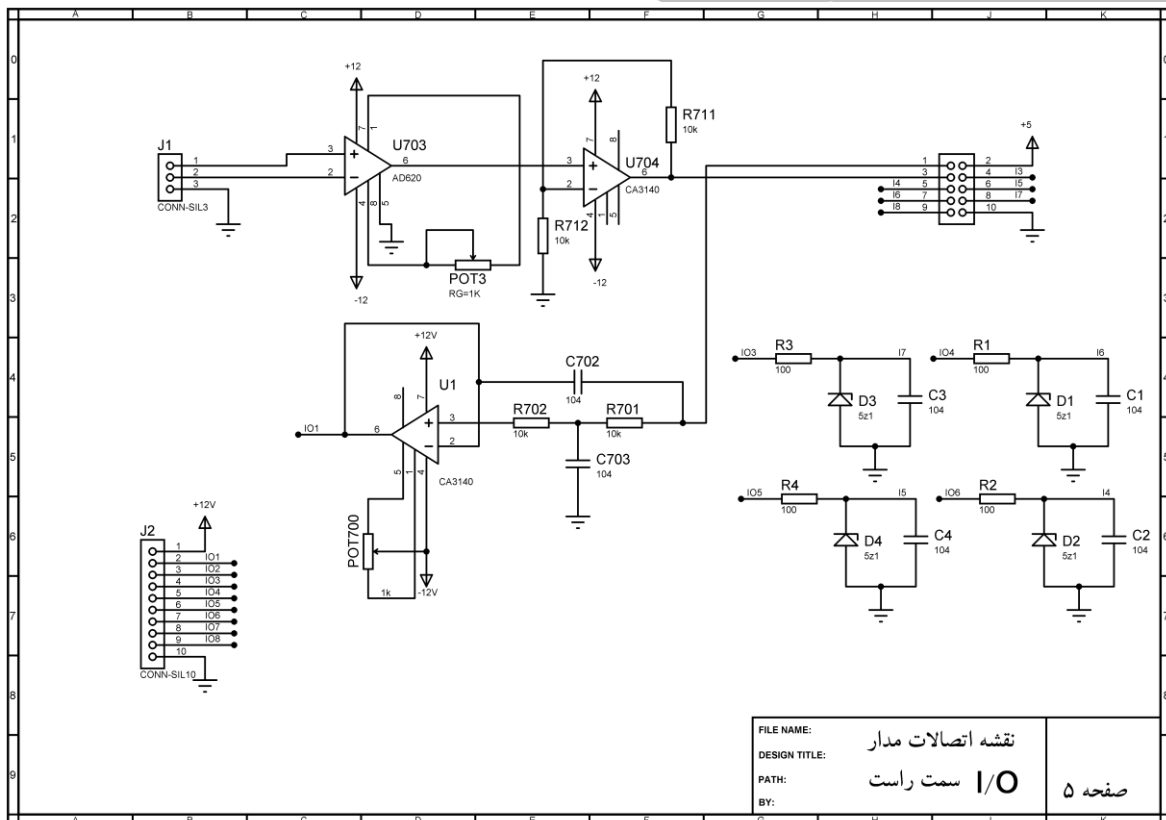
بیت DB6 : پورت A.5

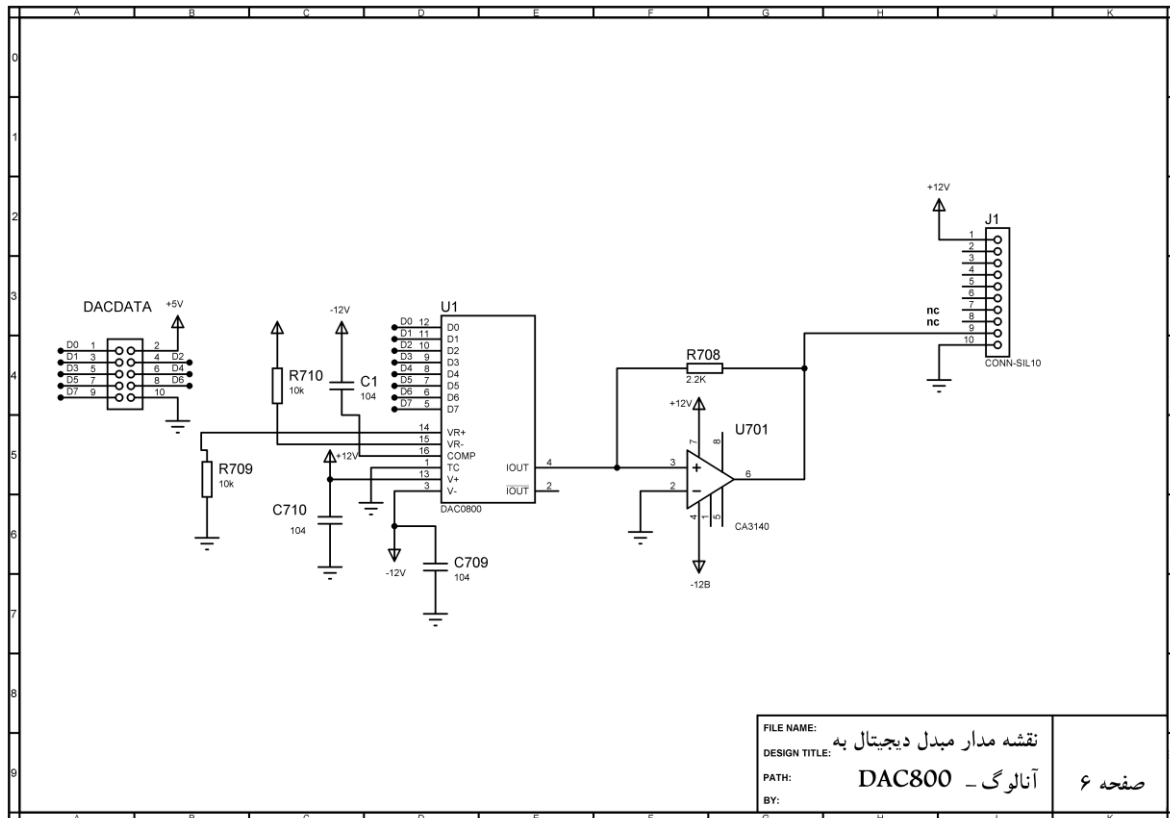
بیت DB7 : پورت A.6

حال با استفاده از این سخت افزار شما باید برنامه ای بنویسید تا با استفاده از این سخت افزار به برقراری ارتباط با

LCD بکند . (بقیه ی مراحل نرم افزاری است)

۱۸- نقشه ی مربوط به ورودی و خروجی سمت راست را از روی نقشه های مجموعه پیدا کنید و در مورد آن با دوستانتان صحبت کنید .





۱۹- پروگرامر مورد استفاده در این دستگاه از کدام پورت ها برای برنامه ریزی میکروکنترلر مورد نظر استفاده میکند ؟

برای برنامه ریزی میکرو کنترلر در پروگرامر AVR ارائه شده به همراه مجموعه از سه پورت USB – SERIAL – PARALLEL استفاده میکند که پروگرامر پورت پارالل به نام STK 200/300 نامیده میشود و پروگرامر پورت سریال به نام JTAG ice شناخته میشود و برای دیباگ کردن برنامه نوشته شده به کار میرود ، و پروگرامر پورت usb به نام usb asp شناخته میشود .

نرم افزارهای BASCOM و CODEVISION از پروگرامر STK 200/300 استفاده میکنند . برای استفاده از پروگرامر سریال (jtag ice) باید از نرم افزار avr studio استفاده شود و برای استفاده از پروگرامر usb باید از نرم افزار usbPROG استفاده کنید که این نرم افزار در CD همراه مجموعه قرار داده شده است . برای این

پروگرامر اطلاعات مفصلی در بخش آموزشی جزوه آورده شده است که برای یادگیری نحوه ی کار با این پروگرامر حتما" مطالب فوق را مطالعه کنید .

۲۰- در کامپیوتر هایی که دارای پورت موازی و سریال و USB هستند کدام یک از پورتها برای استفاده در پروگرامر بهتر هستند ؟

در هر حال فرقی نمیکند که از کدام پورت و یا پروگرامر استفاده کنید چرا که اطلاعات در هر حالت بر روی آی سی برنامه ریزی میشود و فرقی از این نظر با همدیگر ندارند . اما مسئله این است که راحتی کار با کدام یک بیشتر خواهد بود ؟

در استفاده از نرم افزار پروگرامر و در هنگام برنامه ریزی کردن میکرو مهمترین مسئله این است که با استفاده از همان نرم افزار کامپایلر بتوانیم میکرو را برنامه ریزی کنیم ، و چون ما از نرم افزار bascom برای برنامه نویسی میکرو کنترلر avr استفاده میکنیم و این نرم افزار از پورت پارالل و پروگرامر stk200/300 برای برنامه ریزی میکرو کنترلر پشتیبانی میکند بهتر است از این پروگرامر استفاده کنیم .

۲۱- نحوه ی تنظیم پروگرامر در نرم افزار مورد استفاده خود را شرح دهید: اگر از BASOM AVR استفاده میکنید و یا اگر از CODEVISION AVR استفاده میکنید توضیح دهید که چگونه میتوانید از پروگرامر موازی استفاده کنید .

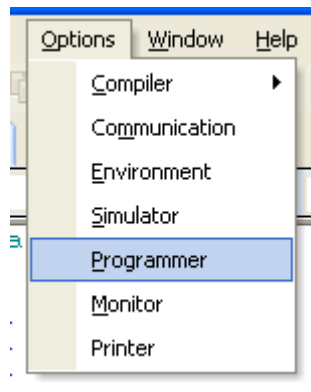
برای استفاده از پروگرامر موازی باید مراحل زیر را به دقت و قدم به قدم انجام دهید :

۱- با استفاده از کابل رابط DB25 سوکت DB25 نرگی در روی برد پروگرامر را به پورت موازی پشت کامپیوتر متصل کنید .

۲- کابل USB قرار داده شده در روی برد را نیز با استفاده از کابل USB به پورت USB کامپیوتر متصل کنید . این کار برای تامین تغذیه ی مورد نیاز برد پروگرامر و نیز برد مجموعه ی آموزشی مورد استفاده قرار خواهد گرفت .

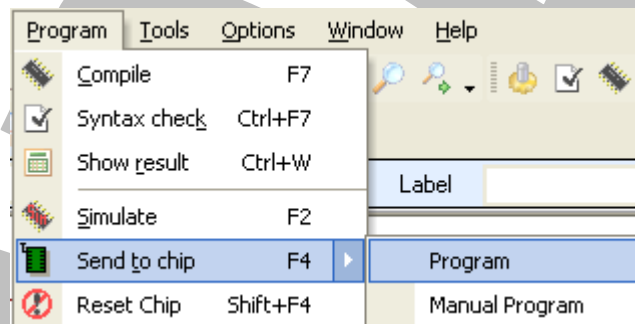
۳- نرم افزار بسکام را اجرا کنید .

۴- در نرم افزار بسکام از منوی option بر روی گزینه ی programmer کلیک کنید .



۵- در پنجره‌ی ظاهر شده در قسمت مربوط به پروگرامر گزینه STK 200/300 را انتخاب کنید و سپس بر روی OK کلیک کنید.

بعد از این برای پروگرام کردن میکرو از طریق نرم افزار BASCOM باید از منوی



گزینه ی Program / Send to chip را فشار دهید. در این قسمت میتوانید میکروی خود را برنامه نویسی کنید.

جواب آزمایش کار :

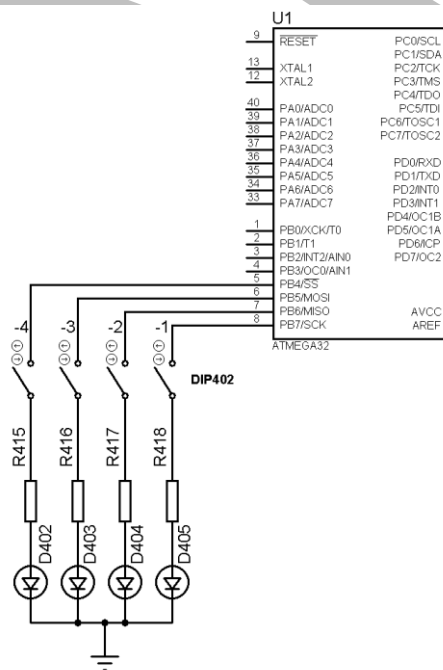
آزمایش ۳: کار با پورت‌های ورودی و خروجی

۱-۳) برنامه ای بنویسید تا LED405 را به فاصله های زمانی ۵۰۰ میلی ثانیه روشن و خاموش کند.

نقشه ی اتصالات مربوط به LED ها در شکل زیر نشان داده شده است . برای این که بتوانید از LED ها استفاده کنید باید کلید های DIP402 را در حالت ON قرار دهید .

تجهیزات مورد نیاز :

یک عدد میکروکنترلر AVR به شماره ATmega 32 - پروگرامر - کابل برق - کابل رابط IDC به طول بیشتر از ۵۰ سانتی متر



مراحل کار :

برای شروع به کار مراحل زیر را با دقت انجام دهید .

- ۱ - نرم افزار BASCOM را اجرا کنید .
- ۲ - از منوی File گزینه ی New را انتخاب کنید .
- ۳ - یک فایل جدید باز میشود . در این فایل نام میکروکنترلر مورد استفاده را وارد کنید . نام میکروکنترلر را با استفاده از دستور `$regfile="m32def.dat"` وارد کنید .

- ۴ - فرکانس کاری میکروکنترلر را با استفاده از دستور `$crystal=8000000` بر روی ۸ مگاهرتز قرار دهید.
- ۵ - پورت B از میکروکنترلر را به صورت خروجی تعریف کنید.
- ۶ - در درون یک حلقه ی `do ... loop` برنامه ی مربوط به تغییر حالت وضعیت پورت متصل شده به LED را که در این مثال `portb.7` میباشد را بنویسید.
- ۷ - بعد از آن مدت زمان تاخیر برنامه را قصد دارید در هر بار روشن و یا خاموش بودن LED صبرشود را وارد کنید. این مدت زمان را با دستور `waitms 500` وارد کنید.

```
$regfile = "m32def.dat"
$crystal = 8000000
```

```
Config Portb = Output
```

```
Do
```

```
Toggle Portb.7
Waitms 500
```

```
Loop
```

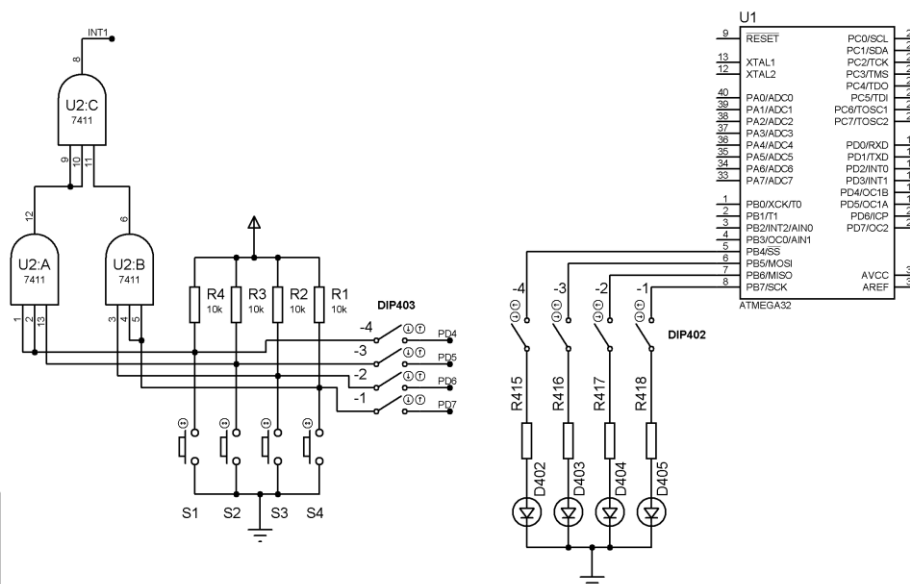
```
End 'end program
```

بعد از نوشتن این خطوط برنامه در نرم افزار BASCOM دکمه ی کامپایل را فشار داده و یا از منوی `program` گزینه ی `compile` را فشار دهید ، سپس میکرو را پروگرام کنید . در این هنگام باید LED شروع به چشمک زدن بکند .

۲-۱) برنامه ای بنویسید اگر کلید S1 فشار داده شده باشد LED1 را روشن کند .

نقشه ی شماتیک مربوط به کلید ها و چراغ ها را در زیر مشاهده میکنید . باید برنامه ای بنویسیم تا ورودی را بخواند و بر اساس آن تصمیم بگیرد که باید LED405 را روشن بکند یا نه . برای این کار باید ابتدا پایه ی مربوط به کلید را به صورت ورودی تعریف بکنیم و نیز پایه ی مربوط به LED را به صورت خروجی تعریف بکنیم .
تجهیزات مورد نیاز :

یک عدد میکروکنترلر AVR به شماره ATmega 32 - پروگرامر - کابل برق - کابل رابط IDC به طول بیشتر از ۵۰ سانتی متر



مراحل کار :

برای شروع به کار مراحل زیر را با دقت انجام دهید :

۱- نرم افزار BASCOM را اجرا کنید .

۲ -از منوی File گزینه ی New را انتخاب کنید .

۳ -یک فایل جدید باز میشود . در این فایل نام میکروکنترلر مورد استفاده را وارد کنید . نام میکروکنترلر را با

استفاده از دستور `$regfile="m32def.dat"` وارد کنید .

۴ -فرکانس کاری میکروکنترلر را با استفاده از دستور `$crystal=8000000` بر روی ۸ مگاهرتز قرار دهید

- ۵- پورت B.7 از میکروکنترلر را به صورت خروجی تعریف کنید .
۶- پورت D.4 از میکروکنترلر را به صورت ورودی تعریف کنید .
۷- در درون یک حلقه ی `do ... loop` برنامه ی مربوط به خواندن اطلاعات از پورت ورودی و قرار دادن آن در پورت خروجی را بنویسید .

```
$regfile = "m32def.dat"
$crystal = 8000000
```

```
Config Pinb.7 = Output
Config Pind.4 = Input
```

```
Do
```

```
Portb.7 = Pind.4
```

```
Loop
```

```
End 'end program
```

در این برنامه همواره در داخل حلقه ی `DO...LOOP` مقدار پایه ی متصل شده به کلید را می خوانیم و آن را در خروجی پایه ی مربوط به `LED` قرار میدهیم . بنابراین اگر کلید مربوطه فشار داده شده باشد مقدار این پایه به صفر تغییر وضعیت میدهد و در غیر این صورت مقدار آن با توجه به وجود مقاومت بالا کشنده در سطح منطقی ۱ قرار دارد . این برنامه جواب مسئله نیست . چرا که در صورت فشردن کلید `LED` را خاموش میکند . آیا میدانید چگونه میتوان این وضعیت را بر عکس کرد ؟

بله : با استفاده از یک دستور `NOT` به این صورت میتوان این فرایند را معکوس کرد .

```
Portb.7 = not Pind.4
```

این دستور را خودتان به برنامه اضافه کنید و آن را بررسی کنید.

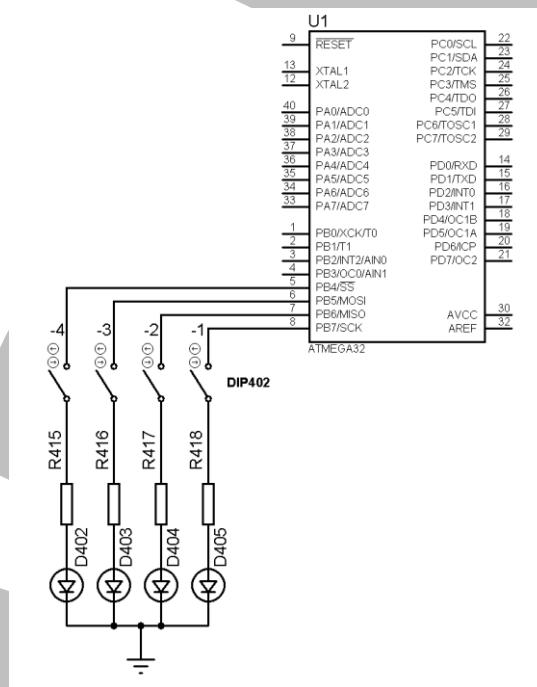
بعد از کامپایل کردن برنامه آن را به داخل میکرو پروگرام کرده و برنامه را به صورت عملی تست کنید .

۳-۳) برنامه ی یک چشمک زن ۴ تایی را با استفاده از میکرو کنترلر AVR بنویسید .

تجهیزات مورد نیاز :

یک عدد میکرو کنترلر AVR به شماره ATmega 32 - پروگرامر - کابل برق - کابل رابط IDC به طول بیشتر از

۵۰ سانتی متر



مراحل کار :

برای شروع به کار مراحل زیر را با دقت انجام دهید .

۱ - نرم افزار BASCOM را اجرا کنید .

۲ - از منوی File گزینه ی New را انتخاب کنید .

۳ - یک فایل جدید باز میشود . در این فایل نام میکرو کنترلر مورد استفاده را وارد کنید . نام میکرو کنترلر را با استفاده از دستور "\$regfile="m32def.dat" وارد کنید .

۴ - فرکانس کاری میکرو کنترلر را با استفاده از دستور "\$crystal=8000000" بر روی ۸ مگاهرتز قرار دهید .

۵ - پورت B از میکرو کنترلر را به صورت خروجی تعریف کنید .

۶ - در درون یک حلقه ی do ... loop برنامه ی مربوط به تغییر حالت وضعیت پورت های متصل شده به LED را که در این مثال portb.4 تا portb.7 میباشد را بنویسید .

۷- بعد از هر دستور مربوط به تغییر وضعیت پایه های فوق یک دستور تاخیر به کار ببرید. این تاخیر به این دلیل استفاده میشود تا LED ها با سرعتی روشن و خاموش شوند که بتوانیم این فرایند را به وضوح مشاهده کنیم. این مدت زمان را با دستور `waitms 500` وارد کنید.

```
$regfile = "m32def.dat"
$crystal = 8000000
```

```
Config Portb = Output
```

```
Do
```

```
Portb=0
Set portb.4
Waitms 500
```

```
Portb=0
Set portb.5
Waitms 500
```

```
Portb=0
Set portb.6
Waitms 500
```

```
Portb=0
Set portb.7
Waitms 500
```

```
Loop
```

```
End 'end program
```

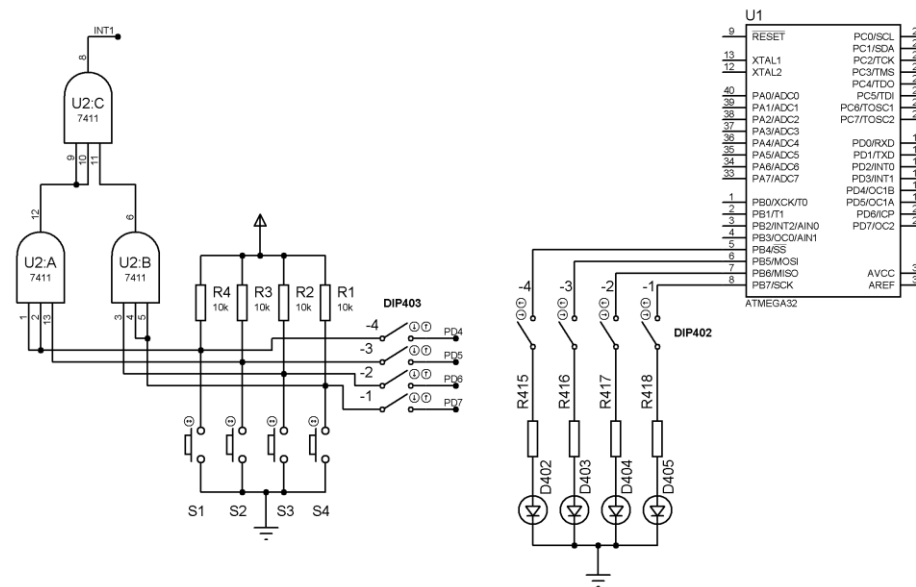
بعد از نوشتن این خطوط برنامه در نرم افزار BASCOM دکمه ی کامپایل را فشار داده و یا از منوی `program` گزینه ی `compile` را فشار دهید، سپس میکرو را پروگرام کنید. قبل از تست برنامه `dip switch` های شماره `DIP 402` را در وضعیت `ON` قرار دهید. در این هنگام باید LED شروع به چشمک زدن بکند.

۳-۴) برنامه ای بنویسید تا اگر کلید S1 فشار داده شود زمان روشن بودن چراغ ها افزایش یابد.

تجهیزات مورد نیاز:

یک عدد میکروکنترلر AVR به شماره ATmega 32 - پروگرامر - کابل برق - کابل رابط IDC به طول بیشتر از

۵۰ سانتی متر



مراحل کار:

برای شروع به کار مراحل زیر را با دقت انجام دهید.

۱ - نرم افزار BASCOM را اجرا کنید.

۲ - از منوی File گزینه ی New را انتخاب کنید.

۳ - یک فایل جدید باز میشود. در این فایل نام میکروکنترلر مورد استفاده را وارد کنید. نام میکروکنترلر را با

استفاده از دستور `$regfile="m32def.dat"` وارد کنید.

۴ - فرکانس کاری میکروکنترلر را با استفاده از دستور `$crystal=8000000` بر روی ۸ مگاهرتز قرار دهید

۵ - پورت B از میکروکنترلر را به صورت خروجی تعریف کنید.

۶ - در درون یک حلقه ی `do ... loop` برنامه ی مربوط به تغییر حالت وضعیت پورت های متصل شده به LED

را که در این مثال `Portb.4` تا `Portb.7` میباشد را بنویسید.

۷ - بعد از هر دستور مربوط به تغییر وضعیت پایه های فوق یک دستور تاخیر به کار ببرید. این تاخیر به این دلیل

استفاده میشود تا LED ها با سرعتی روشن و خاموش شوند که بتوانیم این فرایند را به وضوح مشاهده کنیم.

برای این که بتوانیم مدت زمان تاخیر را تغییر دهیم باید از یک متغیر تعریف شده در حافظه SRAM استفاده کنید. با تغییر عدد قرار داده شده در این متغیر مدت زمان تاخیر نیز به مدت فوق ایجاد میشود. این کار را با دستور `waitms waiting` انجام دهید.

```
$regfile = "m32def.dat"
$crystal = 8000000
```

```
Config Portb = Output
Config pind.4 = input
Config pind.5 = input
```

```
Dim waiting as word
```

```
Do
```

```
If pind.4 = 1 then
  Waitms 25
  If pind.4 = 0 then
    Incr waiting
    If waiting >=500 then
      Waiting = 0
    endif
  endif
endif
```

```
endif
If pind.5 = 1 then
  Waitms 25
  If pind.5 = 0 then
    If waiting <> 0 then
      Decr waiting
    endif
  endif
endif
```

```
endif
```

```
Portb=0
Set portb.4
Waitms waiting
```

```
Portb=0
```



```
Set portb.5
```

```
Waitms waiting
```

```
Portb=0
```

```
Set portb.6
```

```
Waitms waiting
```

```
Portb=0
```

```
Set portb.7
```

```
Waitms waiting
```

```
Loop
```

```
End
```

```
'end program
```

بعد از نوشتن این خطوط برنامه در نرم افزار BASCOM دکمه ی کامپایل را فشار داده و یا از منوی program گزینه ی compile را فشار دهید ، سپس میکرو را پروگرام کنید . قبل از تست برنامه dip switch های شماره DIP 402 را در وضعیت ON قرار دهید . در این هنگام باید LED شروع به چشمک زدن بکند .

دستور کار جلسه چهارم

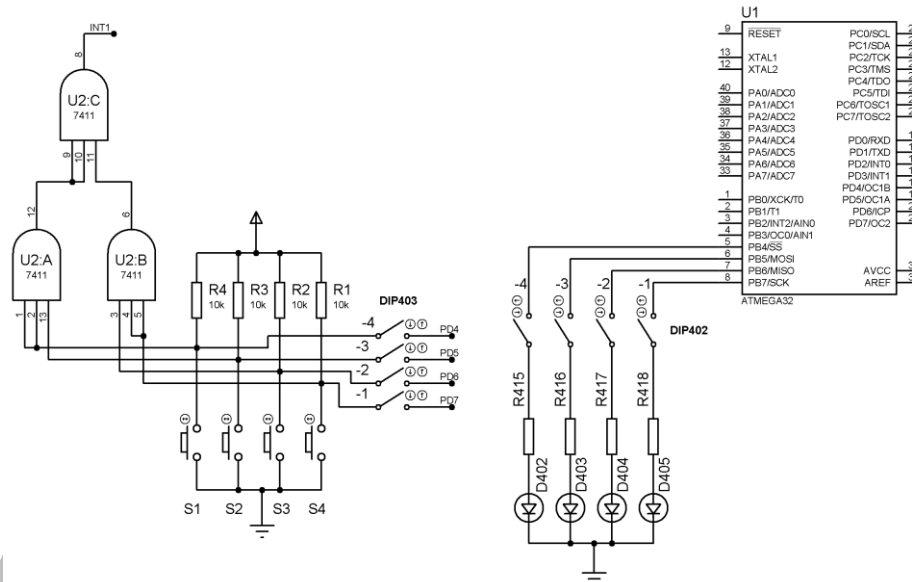
هدف از انجام آزمایش: فراگیری کار با دستورات حلقه - پرش و شرطی زبان بیسیک

آزمایش ۴: کار با دستورات شرطی و انتخابی

۴-۱) برنامه ای بنویسید تا اگر کلید S1 و S2 با هم فشار داده شده باشند LED402 (آبی) روشن شود.

تجهیزات مورد نیاز:

یک عدد میکروکنترلر AVR به شماره ATmega 32 - پروگرامر - کابل برق - کابل رابط IDC به طول بیشتر از ۵۰ سانتی متر



همان طور که در نقشه مشخص شده است کلید های S1 و S2 به پایه های PIND.4 و PIND.5 متصل شده اند.

برای این که بتوانیم از این پایه ها استفاده کنیم باید آنها را به صورت ورودی تعریف کنیم. در ضمن LED402 به پایه

PORTB.4 متصل شده است. برای این که بتوانیم از این LED نیز استفاده کنیم باید پایه ی مربوطه را به صورت

خروجی تعریف کنیم. سپس باید با استفاده از دستورات شرطی بررسی کنیم که کدام یک از کلید ها فشار داده شده اند که

اگر ۲ تای آنها (S1 و S2) با هم فشار داده شده باشند LED402 را روشن کنیم.

برنامه باید به صورت زیر نوشته شود.

```
$regfile = "m32def.dat"
$crystal = 8000000
```

```
Config Portb = Output
config portd= INPUT
PORTB=0
```

```
do
```

```
If Pind.4 = 00 And Pind.5 = 0 Then
```

```
  Set Portb.4
```

```
Else
```

```
  Reset Portb.4
```

```
End If
```

```
Loop
```

```
End
```

```
'end program
```

این برنامه را در نرم افزار **BASCOM** نوشته و سپس کامپایل کرده و در حافظه ی میکرو ریخته و اجرا کنید .
برای این که بتوانید از کلید ها استفاده کنید باید کلید های **DIP403** را در وضعیت **ON** قرار دهید . همچنین باید این
کار را برای **LED** ها با استفاده از **DIP402** انجام دهید .



۲-۴) برنامه ای بنویسید که اگر کلید S1 فشار داده شده باشد LED402 روشن شود. اگر کلید S1 و S2 با هم فشار داده شوند LED403 روشن شود. S1 و S2 و S3 با هم فشار داده شوند LED404 روشن شود و اگر هیچ کلیدی فشار داده نشده است LED405 روشن شود.

این تمرین برای آشنایی شما با دستورات مربوط به انتخاب طراحی شده است. برای مثال دستور SELECT CASE این امکان را برای شما فراهم میکند تا بتوانید از بین چند کار که باید انجام دهید یکی را به درستی انتخاب کنید. ساختار دستور SELECT CASE را در جزوه ی آموزشی همراه مجموعه یاد گرفتید. اکنون شروع به انجام آزمایش میکنیم. باید پورتهای ورودی و خروجی مورد استفاده را تعریف کنیم.

```
$regfile = "m32def.dat"
$crystal = 8000000
```

```
Config Portb = Output
config portd= INPUT
PORTB=0
```

```
do
```

```
If Pind.4 = 0 Then
```

```
Portb = &B00010000
```

```
Elseif Pind.4 = 0 And Pind.5 = 0 Then
```

```
Portb = &B00100000
```

```
Elseif Pind.4 = 0 And Pind.5 = 0 And Pind.6 = 0 Then
```

```
Portb = &B01000000
```

```
Elseif Pind.4 = 1 And Pind.5 = 1 And Pind.6 = 1 And Pind.7 = 1 Then
```

```
Portb = &B10000000
```

```
End If
```

```
Loop
```

```
End
```

```
'end program
```

۳-۴) برنامه ی شمارنده ی ۴ بیتی با استفاده از LED405 – LED402 بسازید .

برنامه ای بنویسید که در هر ثانیه مقدار یک متغیر را در حافظه SRAM یک واحد افزایش دهد و ۴ بیت کم ارزش داده ی مربوط به آن را توسط LED ها ی روی برد نشان دهد .

برای انجام این آزمایش باید یک متغیر (از نوع بایت در حافظه ی Sram) تعریف کنید . سپس باید برنامه را طوری بنویسید تا مقدار این متغیر دائما " در حال افزایش باشد . (با یک تاخیر زمانی مشخص) سپس مقدار ۴ بیت کم ارزش این متغیر را توسط LED ها به نمایش در آورید .

```
$regfile = "m32def.dat"
```

```
$crystal = 8000000
```

```
Config Portb = Output
```

```
Dim Var As Byte
```

```
do
```

```
Portb.4 = Var.0
```

```
Portb.5 = Var.1
```

```
Portb.6 = Var.2
```

```
Portb.7 = Var.3
```

```
Incr Var
```

```
waitms 500
```

```
Loop
```

```
End 'end program
```

این برنامه را در نرم افزار **BASCOM** نوشته و سپس کامپایل کرده و در حافظه ی میکرو ریخته و اجرا کنید .
برای این که بتوانید از **LED** ها استفاده کنید باید کلید های **DIP402** را در وضعیت **ON** قرار دهید .

دستور کار جلسه ی پنجم

هدف از انجام آزمایش : فراگیری کار LCD های کاراکتری

آزمایش ۵: نحوه ی کار با LCD های کاراکتری

۱-۵) برنامه ای بنویسید تا در روی LCD عبارت "Hello" نمایش داده شود .

تجهیزات مورد نیاز :

یک عدد میکروکنترلر AVR به شماره ATmega 32 - پروگرامر - کابل برق - کابل رابط IDC به طول بیشتر از ۵۰ سانتی متر - کابل رابط IDC به طول ۲۰ سانتی متر

برای این که بتوانید از LCD کاراکتری در محیط برنامه نویسی بسکام استفاده کنید باید ترتیب اتصالات بین میکرو و LCD را از طریق محیط نرم افزار Bascom تنظیم کنید . دقت کنید اگر این تنظیمات را به درستی انجام ندهید LCD کار نخواهد کرد .

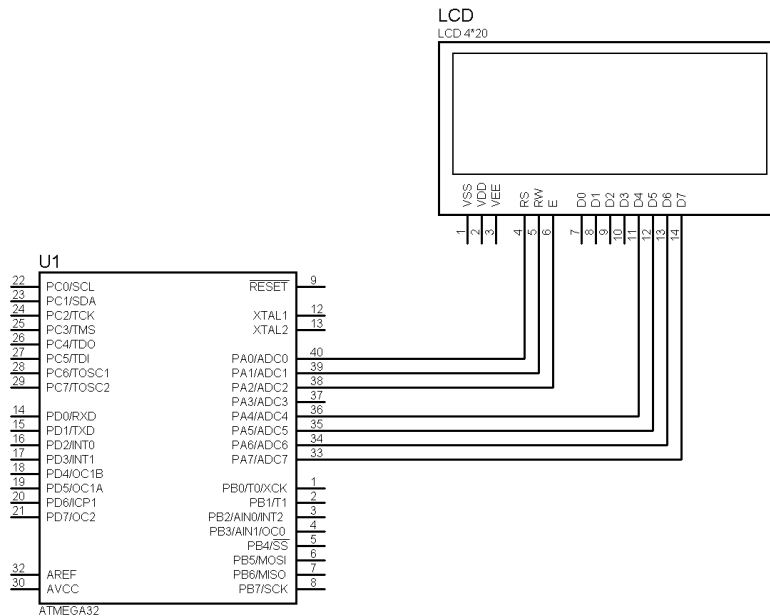
نحوه ی انجام این کار در جزوه ی تئوری آورده شده است .

همان طور که میدانید در این مجموعه LCD کاراکتری در یک برد مستقل از برد اصلی میکروکنترلر قرار داده شده است .

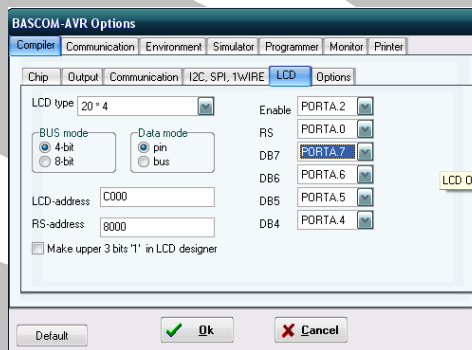
برای ایجاد ارتباط بین میکروکنترلر و LCD باید یکی از کانکتورهای مربوط به پورتهای میکروکنترلر را که با نام کامل پورت مربوطه بر روی برد مشخص شده است را با استفاده از کابل رابط IDC به کانکتور مربوط به LCD بر روی قسمت کانکتورهای ماجول نمایشگر که در برد ۲ قرار دارد و با نام LCDDATA مشخص شده است متصل کنید .

در این مثال فرض شده است که این ارتباط را از طریق PORTA برقرار کرده اید . برای این کار کانکتور IDC که در گوشه ی سمت راست بالا در ماجول R-Main Module قرار داده شده است و با نام PORTA بر روی مارکاژ

مشخص شده است را با استفاده از کابل IDC ۱۰ رشته که به همراه مجموعه ارائه شده است ، به کانکتور LCDDATA در گوشه ی سمت راست بالا در برد ۲ متصل کنید . بعد از این اتصال نقشه ی اتصالات مربوط به میکرو و LCD به این صورت خواهد بود .



شکل بالا نحوه ی اتصالات بین LCD و PORTA نمایش میدهد .



برای انجام تنظیمات مربوط به اتصالات LCD از منوی Option/Compiler گزینه LCD را کلیک کنید . تصویر بالا نحوه ی تنظیم اطلاعات مربوط به پایه های متصل شده بین میکرو و lcd را از طریق نرم افزار BASCOM نمایش میدهد . بعد تنظیم آنها دکمه ی Ok را فشار دهید .

حال شروع به برنامه نویسی برای میکروکنترلر میکنیم . تنظیمات اصلی را همانند برنامه های قبلی انجام دهید . تعریف آی سی مورد استفاده - فرکانس کاری و نیز پورت های ورودی و خروجی . که در این برنامه برای پورت مربوط به LCD نیازی به تعریف ورودی و یا خروجی نمیباشد .

```
$regfile = "m32def.dat"
$crystal = 8000000
```

```
Config Pina.1 = Output
```


Do

```
Reset Porta.1
Home
Cls
Lcd "Hello"
Waitms 1000
```

Loop

```
End 'end program
```

این برنامه را در محیط نرم افزار وارد کنید و سپس مراحل پروگرام کردن را طبق آزمایشات قبلی انجام دهید .
در این برنامه دقت بکنید که چرا دستور `reset porta.1` را قبل از دستورات مربوط به LCD نوشته ایم . علت این امر این است که چون پایه ی RW از LCD به میکرو کنترلر متصل شده است. پایه ی RW از LCD مربوط به کنترل نوشتن و یا خواندن از LCD میباشد. در مواقعی که می خواهیم در روی LCD اطلاعاتی را بنویسیم باید این پایه به زمین متصل باشد و چون که ما در این قسمت میخواهیم بر روی LCD اطلاعاتی را بنویسیم باید با استفاده از دستورات میکرو کنترلر این پایه را به زمین وصل کنیم .



۲-۵) برنامه ای بنویسید تا مقدار یک متغیر (برای مثال متغیر a از نوع byte) را در هر ثانیه یک واحد افزایش دهد و آن را در سطر دوم و ستون پنجم از LCD نمایش دهد .

جواب: برای انجام این کار تنظیمات مربوط به میکرو و فرکانس کاری و LCD را همانند آزمایشات قبلی انجام دهید . در این مثال باید یک متغیر در حافظه Sram تعریف کنید تا بتوانید مقدار آن را تغییر داده و بر روی LCD نمایش دهید . دستوری که با استفاده از آن میتوانید به LCD فرمان دهید تا در مکانی خاص از LCD مطالب مورد نظر شما را بنویسید دستور Locate x,y میباشد .

```
$regfile = "m32def.dat"
$crystal = 8000000
```

```
Config Pina.1 = Output
```

```
Dim A As Byte
```

```
Do
```

```
Incr A
If A >= 255 Then
  A = 0
End If
```

```
Reset Porta.1
Cls
Locate 2 , 5
Lcd "Hello"
```

```
Waitms 1000
```

```
Loop
```

```
End 'end program
```

این برنامه را در محیط نرم افزار وارد کنید و سپس مراحل پروگرام کردن را طبق آزمایشات قبلی انجام دهید.

DRAFT



۳-۵) برنامه ای بنویسید تا عبارت نوشته شده روی LCD به سمت چپ به چرخش در آید .

جواب : برای انجام این آزمایش باید نحوه ی کار با دستورات چرخشی را بدانید .

دستور shiftlcd right تمام اطلاعات نوشته شده بر روی LCD به سمت راست و دستور Shiftlcd left کل

اطلاعات نوشته شده بر روی LCD را به سمت چپ شیفت میدهد . (انتقال میدهد) .

برنامه را به این صورت مینویسیم :

```
$regfile = "m32def.dat"
```

```
$crystal = 8000000
```

```
Config Pina.1 = Output
```

```
Dim A As Byte
```

```
Reset Porta.1
```

```
Cls
```

```
Locate 2 , 5
```

```
Lcd "Hello"
```

```
Do
```

```
Waitms 1000
```

```
Reset Porta.1
```

```
Shiftlcd Left
```

```
Loop
```

```
End
```

```
'end program
```

در این برنامه یک بار اطلاعاتی بر روی LCD نوشته میشود و سپس در هر ثانیه یک بار این اطلاعات به چرخش در می آید

. این برنامه را در محیط نرم افزار وارد کنید و سپس مراحل پروگرام کردن را طبق آزمایشات قبلی انجام دهید .

۴-۵) برنامه ی یک شمارنده را طوری بنویسید تا تعداد دفعات فشرده شدن کلید S1 بشمارد و مقدار آن را همواره روی LCD به نمایش در آورد.

سوال : بعد از این که اطلاعاتی که در سطر دوم نوشته شده اند بعد از خروج از انتهای سمت چپ سطر دوم از ابتدای سطر چهارم وارد lcd میشود ؟

برای نوشتن این برنامه دو راه دارید . ۱- استفاده از دستورات معمولی برنامه نویسی ۲- استفاده از وقفه برای نوشتن این برنامه در برد ۱ از هر دو راه میتوانید استفاده کنید . در این مثال قصد داریم از برنامه نویسی معمولی استفاده بکنیم .

برای انجام این کار لازم است تا با دستور DEBOUNCE در زبان بیسیک آشنا باشید . دستور DEBOUNCE برای تعریف کردن یک کلید به میکروکنترلر به کار میرود . شکل کلی این دستور به صورت زیر میباشد .

DEBOUNCE Px.y , state , label [, SUB]

PX,Y : پایه ای از میکروکنترلر میباشد که کلید به آن متصل شده است .

STATE : وضعیت منطقی ای است که در صورت فشرده شدن کلید ، در روی پایه تولید میشود .

LABLE : نام زیر برنامه ای است که در صورت فشرده شدن کلید باید اجرا شود .

SUB : گزینه اختیاری است که اگر نوشته شود به معنی این است که بعد از اجرای زیر برنامه برگرد و دستورات بعد از دستور DEBOUNCE را اجرا کن .

```
$regfile = "m32def.dat"
```

```
$crystal = 8000000
```

```
Config Pina.1 = Output
```

```
Dim A As Byte
```

```
Do
```

```
Debounce Pind.4 , 0 , L1 , Sub
```

```
Loop
```

```
L1:
```

```
Incr A
```

```
If A = 255 Then
```

```
    A = 0
```

```
End If
```

```
Reset Porta.1
```

```
Cls
```

```
Locate 2 , 5
```

```
Lcd A
```

```
Return
```

```
End
```

```
'end program
```

آزمایش ۶: کار با کیبورد ماترسی ۴*۴

۱-۶) برنامه ای بنویسید تا همواره کیبورد ۴*۴ ماترسی را اسکن کند و در صورتی که کلیدی از آن فشار داده شود شماره ی آن کلید را روی LCD نمایش دهد.

برای انجام این آزمایش باید نحوه ی کار کیبورد های ماترسی را بدانید. نحوه ی اسکن کیبورد های ماترسی بسیار در فهم مسائل مربوط به برنامه نویسی به شما کمک میکند. در این جا ما قصد نداریم به این بحث پردازیم. چون در زبان برنامه نویسی بیسیک با استفاده از نرم افزار BASCOM برای میکروکنترلر AVR دستورات بسیار قدرتمند تری برای کار با این گونه صفحات ماترسی وجود دارد.

دستور CONFIG KBD = PORTX: با استفاده از این دستور میتوانیم به میکروکنترلر تعریف کنیم که کیبورد ماترسی به کدام یک از پورت های میکرو کنترلر متصل شده است.

دستور GETKBD(): با استفاده از این دستور میتوانیم عدد فشرده شده از صفحه کلید را بخوانیم. تنظیمات مربوط به میکرو - فرکانس کاری و LCD را همانند آزمایشات قبلی انجام دهید. پس برنامه را به این شکل مینویسیم:

```
$regfile = "m32def.dat"
$crystal = 8000000
```

```
Config Kbd = Portb
```

```
Dim A As Byte
```

```
Do
```

```
A = Getkbd()
```

```
If A <> 16 Then
```

```
Reset Porta.1
```

```
Cls
```

```
Lcd A
```

```
End If
```

```
Loop
```

End

'end program

۲-۶) برنامه ای بنویسید تا در صورت فشار داده شدن هر یک از کلید های صفحه کلید علامت درج شده بر برجسب کیبورد را روی LCD کاراکتری نمایش دهد.

برای انجام این آزمایش باید مشابه آزمایش قبلی کیبورد را راه اندازی و برنامه مربوط به خواندن صفحه کلید و نشان دادن آن بر روی صفحه LCD را بنویسید. در این حالت خواهید دید که اگر کلید ۱ از روی صفحه کلید فشار داده شود دستور Getkbd() عدد ۰ را برمیگرداند و اگر کلید ۲ فشار داده شود دستور Getkbd() عدد ۱ را برمیگرداند. و بقیه الی آخر (طبق جدول زیر)

حال برای این که با توجه به عدد برگردانده شده از دستور Getkbd() بدانید که نام کلید مربوطه چیست باید از دستور Lookup استفاده کنید. عملکرد دستور Lookup بدین صورت میباشد که یک آدرس جدول و شماره ردیف را به عنوان ورودی دریافت میکند و با توجه به آن عدد یک آرگومان را از روی جدول برمیگرداند.

A=Lookup(a,data)

A: خروجی آرگومان از روی جدول

a: ورودی شماره ردیف

Data: نام برجسب آدرس مربوط به برجسب روی کیبورد

عدد خوانده شده	عدد روی کیبورد
0	۱
1	۲
2	۳
3	F1
4	4
5	5
6	6
7	F2
8	7
9	8
10	9
11	F3
12	stop
13	0
14	start
15	Meno

برنامه ی مربوط به lookup را به این صورت به برنامه قبلی اضافه میکنیم:

```
$regfile = "m32def.dat"  
$crystal = 8000000
```

```
Config Kbd = Portb
```

```
Dim A As Byte
```

```
Do
```

```
A = Getkbd()
```

```
If A <> 16 Then
```

```
    A = Lookup(a , Table(
```

```
        Reset Porta.1
```

```
        Cls
```

```
        Lcd A
```

```
    End If
```

```
Loop
```

```
End                                    'end program
```

```
Table:
```

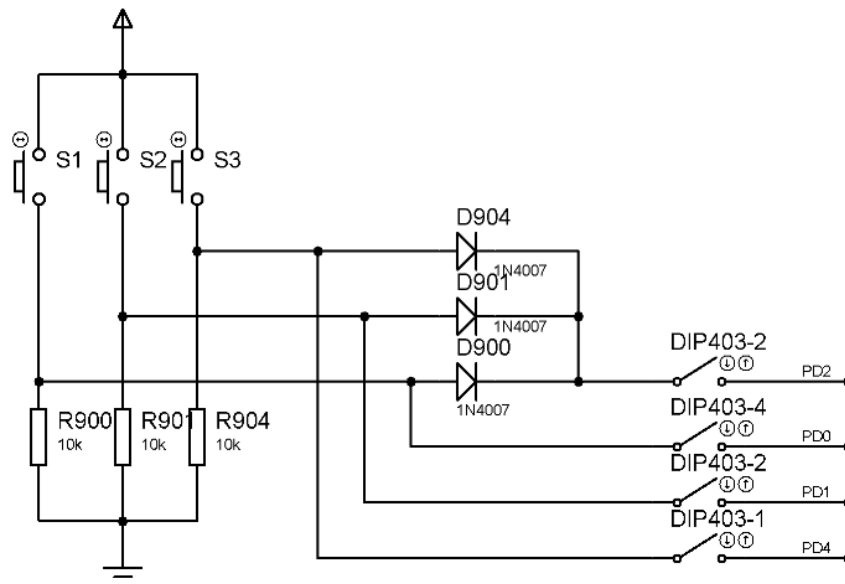
```
Data 1 , 2 , 10 , 3 , 4 , 5 , 11 , 6 , 7 , 8 , 12 , 13 , 0 , 14
```

آزمایش ۷: کار با سرویس های وقفه در میکروکنترلر

۷-۱) برنامه ای بنویسید تا در صورت وقوع وقفه خارجی روی LCD عبارت "int0" نوشته شود. آزمایش را در برد ۲ انجام دهید.

برای انجام این آزمایش در برد ۲ باید ابتدا LCD را همانند آزمایشات قبلی پیکره بندی کنید. در این مثال LCD را بر روی پورت A پیکره بندی کنید.

برای استفاده از وقفه ی شماره 0 میکروکنترلر Atmega32 بر روی برد ۳ عدد میکرو سوئیچ قرار داده شده است که با استفاده از گیت AND دیودی به پایه ی INTO متصل شده اند. نقشه ی شماتیک مربوط به میکرو سوئیچ های برد ۲ در زیر آورده شده است.



نقشه اتصالات طیت AND دیودی در برد شماره ۲

همان طور که در این نقشه مشخص شده است در صورت فشرده شدن هر یک از میکرو سوئیچ ها پایه ی مربوطه از میکروکنترلر به VCC متصل میشود. و از هر یک از پایه های مربوط به میکرو سوئیچ ها با استفاده از یک دیود به پایه ی pd2 , (int0) متصل شده است. پس اگر هر یک از میکرو سوئیچ ها فشرده شود یک پالس باله ی بالا رونده بر روی پایه ی INTO ایجاد میشود. حال باید برنامه ی مربوط به int0 را طوری نوشت که به لبه ی بالا رونده حساس بوده و در زیر برنامه ی مربوط به INTO برنامه ی مربوطه را اجرا بکند.

```
$regfile = "m32def.dat"
$crystal = 8000000
```

```
Config Int0 = Rising
```

```
On Int0 Int0_isr  
Enable Int0  
Enable Interrupts
```

```
Do
```

```
Loop
```

```
Int0_isr:
```

```
Clc
```

```
Home
```

```
Lcd "int0"
```

```
Return
```

```
End
```

```
'end program
```

۷-۲) برنامه ی شمارنده را که قبلا " نوشته بودید حال با استفاده از سرویس وقفه راه اندازی کنید . مزایای استفاده از وقفه به جای برنامه نویسی در روتین اصلی برنامه چیست ؟

این آزمایش را هم در برد B انجام دهید .

برای این که بتوانید با استفاده از وقفه ی میکرو کنترلر یک شمارنده طراحی بکنید باید : ابتدا وقفه ی میکرو کنترلر را راه اندازی کرده و سپس در زیر برنامه ی مربوط به وقفه عملیات افزایش و یا کاهش مقدار شمارنده انجام دهید .

مراحل راه اندازی وقفه و LCD را همانند آزمایشات قبلی انجام دهید .

حال باید قسمتی به برنامه اضافه شود تا این که در زیر برنامه وقفه (یعنی هر بار که کلید فشار داده شده باشد) مقدار یک متغیر تعریف شده در حافظه ی SRAM را یک واحد افزایش دهد . این کار بسیار ساده است . تنها با اضافه کردن یک دستور

INCR var

این کار انجام میشود . نام متغیر را count1 تعریف می کنیم . آن را از نوع long انتخاب میکنیم تا این که بتوانیم ارقام بیشتری را بشماریم .

```
$regfile = "m32def.dat"
```

```
$crystal = 8000000
```

```
Config Int0 = Rising
```

```
On Int0 Int0_isr
```

```
Enable Int0
```

```
Enable Interrupts
```

```
Dim Count1 As Long
```

```
Do
```

```
Loop
```

```
Int0_isr:
```

```
Incr Count1
```

```

Cls
Home
Lcd "counter=" ; Count1
Return
End                                     'end program

```

۳-۷) برنامه را طوری تغییر دهید تا شمارش را هم به صورت افزایشی و هم به صورت کاهشی انجام دهد .
توجه :

حال بعد از کامپایل کردن برنامه و پروگرام کردن برنامه بر روی میکرو ، میکرو شروع به اجرا کردن برنامه میکند .

اگر دقت کرده باشید با فشار دادن هر یک از کلیدها شمارنده عمل میکند . علت این امر چیست ؟

جواب : این بدان علت است که ما در این برنامه در زیر برنامه ی مربوط به وقفه ی شماره ۰ برنامه ی مربوط به شمارنده را

نوشته ایم . این در حالی است که با فشار دادن هر یک از کلیدها به دلیل AND بودن این کلیدها یک پالس باله ی بالا رونده به پایه ی مربوط به وقفه ارسال میشود و باعث اجرای برنامه شمارنده میشود .

به نظر شما برای حل این مشکل چه کاری باید انجام داد ؟

جواب : باید در زیر برنامه مربوط به وقفه اصلاحاتی را انجام داد و برنامه را طوری نوشت که قبل از انجام عملیات شمارش

بررسی بکند و بداند که کدام یک از کلیدها فشار داده شده است . برای این کار باید وضعیت پایه های ورودی میکرو را که

به به همان میکرو سوئیچ ها متصل شده است را بررسی بکند . اگر وضعیت این پایه ها ۱ باشد به این معنی است که کلید

مربوط به آن پایه فشار داده شده است و اگر وضعیت این پایه ۰ باشد به این معنی است که آن کلید فشار داده نشده است .

حال این برنامه را به برنامه ی قبلی اضافه میکنیم و برنامه را طوری اصلاح میکنیم تا این که بتواند شمارش را هم به صورت

افزایشی و هم به صورت کاهشی انجام دهد . به اینصورت که اگر کلید S1 فشار داده شود شمارنده افزایش یافته و اگر

کلید S2 فشار داده شود شمارنده ۱ واحد کاهش یابد .

```
$regfile = "m32def.dat"
```

```
$crystal = 8000000
```

```
Config Portd = Input
```

```
Config Int0 = Rising
```

```
On Int0 Int0_isr
```

```
Enable Int0
Enable Interrupts
```

```
Dim Count1 As Long
```

```
Do
```

```
Loop
```

```
Int0_isr:
```

```
If Pind.0 = 1 Then
```

```
    Incr Count1
```

```
Elseif Pind.1 = 1 Then
```

```
    Decr Count1
```

```
End If
```

```
Cls
```

```
Home
```

```
Lcd "counter=" ; Count1
```

```
Return
```

```
End
```

```
'end program
```

این برنامه توانایی شمارش به صورت منفی را دارد یعنی اگر ارقام از ۰ پایین روند با علامت منفی نمایش داده خواهند شد .

آزمایش ۸: کار با مبدل آنالوگ به دیجیتال داخلی میکروکنترلر

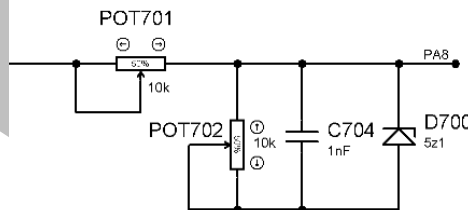
۸-۱) مبدل آنالوگ به دیجیتال داخلی میکرو راه اندازی کنید و برنامه ای بنویسید تا عدد متناظر با ولتاژ ورودی را روی LCD نمایش دهد. با استفاده از ولتاژ مرجع برابر با VCC

برای انجام این آزمایش شما باید با ساختار مبدل آنالوگ به دیجیتال داخلی میکرو و نیز دستورات مربوط به راه اندازی و استفاده از آن آشنا باشید. اگر در این زمینه اطلاعات کمی دارید بهتر است ابتدا به کتاب تئوری مربوط به میکروکنترلر AVR به زبان BASIC مراجعه کنید و بعد از یادگیری مطالب مربوط به آن، آزمایش را انجام دهید.

برای انجام این کار باید LCD را بر روی پورتهای غیر از PORTA از میکروکنترلر متصل بکنید. مثلاً به PORTC علت این امر این است که در این آزمایش به PORTA برای کار با مبدل آنالوگ به دیجیتال میکروکنترلر نیاز داریم.

این آزمایش را در برد A انجام دهید تا بتوانید از ورودی های آنالوگ مربوط به ماژول I/O سمت راست استفاده بکنید. (I/O module R)

پایه ی Ain از کانکتور ورودی مربوط به ماژول I/O module R را با استفاده از تکه سیم کوچکی به ولتاژ +۱۲ ولت که در همان کانکتور قرار داده شده است متصل کنید. نقشه ی این قسمت از مدار به این صورت زیر میباشد.



با استفاده از دو عدد پتانسیومتر مولتی ترن 10Kohm یک ولتاژ متغیر به روی پایه ی ۹ از کانکتور J701 و J702 تولید میشود. برای این که بتوان این ولتاژ را اندازه گیری کرد باید این کانکتور را با استفاده از یک کابل رابط IDC به PORTA از میکروکنترلر متصل کرد. بعد از این اتصال این پایه (خروجی ولتاژ متغیر از ورودی AIN) به پایه ی PA7 متصل میشود. حال باید برنامه را طوری بنویسید تا ورودی آنالوگ را از روی پایه ی PA7 خوانده و مقدار آن را بر روی LCD نمایش دهد.

مراحل زیر را با دقت انجام دهید.

۱- نرم افزار BASCOM را اجرا کنید.

۲- از منوی File گزینه ی New را انتخاب کنید.

۳ - یک فایل جدید باز میشود . در این فایل نام میکروکنترلر مورد استفاده را وارد کنید . نام میکروکنترلر را با استفاده از دستور "\$regfile="m32def.dat" وارد کنید .

۴ - فرکانس کاری میکروکنترلر را نیز وارد کنید . این کار را با دستور \$crystal=8000000 انجام دهید .

۵ - تنظیمات LCD را در منوی Option/Compiler/Lcd انجام دهید .

تنظیمات مربوط به مبدل آنالوگ به دیجیتال را انجام دهید .

این کار با دستور زیر انجام میشود .

Config adc=single , prescaler = auto , reference=avcc

۷- مبدل آنالوگ به دیجیتال را با استفاده از دستور زیر راه اندازی کنید .

۸- حال دستور خواندن از ورودی آنالوگ مربوط به Porta.7 را بنویسید . این دستور مقدار ورودی آنالوگ را از

پورت مربوطه خوانده و در متغیر Analog قرار میدهد .

Analog = getadc(7)

در این دستور منظور از عدد ۷ همان بیت ۷ از پورت A میباشد .

۹- بعد از این مرحله دستور مربوط به نوشتن عدد موجود در متغیر Analog بر روی LCD را بنویسید . این دستور

دستور LCD Analog میباشد .

۱۰- برای این که مطالب نوشته شده در روی LCD همواره در یک قطعه از روی LCD نوشته شود باید قبل از دستور

LCD Analog آدرس محل نوشتن این عدد را وارد کنیم . این دستور ، دستور Home میباشد . این دستور باعث

میشود مکان نما به آدرس سطر ۱ و ستون ۱ از LCD انتقال داده شود و در نتیجه اطلاعات در این نقطه از LCD نوشته

شود .

```
$regfile = "m32def.dat"
```

```
$crystal = 8000000
```

```
Config Adc = Single , Prescaler = Auto , Reference = Avcc
```

```
Start Adc
```

```
Dim Analog As Word
```

```
Do
```

```
Analog = Getadc(7)
```

```
Home
```

```
Cls
```



```
Lcd "analog=" ; Analog" " ;
Waitms 250
```

```
Loop
```

```
End 'end program
```

این برنامه را نوشته و کامپایل کرده و سپس میکروکنترلر را پروگرام کنید.

۲-۸) برنامه ی قبلی را طوری اصلاح کنید تا مقدار واقعی ولتاژ ورودی را روی LCD به نمایش در آورد. (طراحی یک ولتمتر از مقدار ۰ تا ۵ ولت)
برای انجام این آزمایش باید فرمول تبدیل آنالوگ به دیجیتال را بدانید. این کار به این شکل انجام میگیرد که:
ولتاژ ورودی از ۰ تا مقدار ولتاژ مرجع به عددی بین ۰ تا ۱۰۲۳ تبدیل میشود. و برای ولتاژهای بیشتر از ولتاژ مرجع مقدار ۱۰۲۳ برگردانده میشود.

البته لازم به ذکر است که ولتاژ ورودی به پایه مبدل آنالوگ به دیجیتال نباید از مقدار ۵ ولت زیاد تر باشد.

این نحوه ی تبدیل را میتوان با جدول تناسب زیر نمایش داد.

مقدار ولتاژ برابر ولتاژ مرجع	۱۰۲۳
مقدار ولتاژ ورودی	عدد خروجی از ADC

برای تبدیل عدد خوانده شده به مقدار واقعی ولتاژ روی پایه ی ورودی مبدل آنالوگ به دیجیتال باید از فرمول زیر استفاده کرد:

$$V = (\text{getadc} \text{ * عدد برگردانده شده توسط دستور }) / 1023$$

عددی که این فرمول به دست میدهد دقیقاً همان ولتاژی است که بر روی پایه میکروکنترلر وجود دارد.

مثال: فرض کنید عدد خوانده شده از دستور `getadc(7)` مقدار ۲۵۰ باشد. با فرض ولتاژ مرجع برابر با ۵ ولت مقدار واقعی ولتاژ در ورودی `porta.7` چقدر میباشد.

$$V = (\text{getadc} \text{ * عدد برگردانده شده توسط دستور }) / 1023$$

$$V = (250 * 5) / 1023 = 1.22$$

پس ولتاژ موجود بر روی پایه ی `porta.7` برابر است با ۱.۲۲.

حال باید این فرمول را به برنامه اضافه کنید.



همان طور که میدانید نوشتن محاسبات ریاضی در برنامه نویسی به زبان basic دقیقاً همانند نوشتن به زبان ریاضی میباشد

پس برنامه ای که باید به برنامه ی قبلی اضافه شود به این صورت نوشته میشود .

قبل از نوشتن این محاسبه باید یک متغیر از نوع single در برنامه اضافه کنیم . علت این که این متغیر جدید را از نوع

single تعریف میکنیم این است که عدد حاصل از این محاسبه دارای اعشار میباشد و اگر متغیر را از نوعی تعریف

میکردیم که نمیتوانست ممیز را در خود جای دهد ، اعداد بعد از ممیز را حذف میکرد . در نتیجه عدد خروجی دقت خود را

از دست میداد .

Dim v as single

خطوط برنامه ای که باید به برنامه ی قبلی اضافه شود .

V = analog * 5

V = v / 1023

برنامه ی کامل :

```
$regfile = "m32def.dat"
```

```
$crystal = 8000000
```

```
Config Adc = Single , Prescaler = Auto , Reference = Avcc
```

```
Start Adc
```

```
Dim Analog As Word
```

```
Dim V As Single
```

```
Do
```

```
Analog = Getadc(7)
```

```
V = analog * 5
```

```
V = v / 1023
```

```
Home
```

```
Cls
```

```
Lcd "Vin=" ; V ; " Volt "
```

```
Waitms 250
```

```
Loop
```

End

'end program

۳-۸) در برنامه قبلی مقدار دقت اندازه گیری ولتاژ ورودی چه مقدار است ؟
در این آزمایش سعی کنید مقدار این دقت را دو برابر کنید .

دقت مبدل آنالوگ به دیجیتال منظور حداقل ولتاژی است که یک مبدل آنالوگ به دیجیتال میتواند به مقدار دیجیتال تبدیل بکند . در مبدل آنالوگ به دیجیتال داخلی میکروکنترلر AVR تعداد بیت مبدل ۱۰ بیت میباشد . که حداکثر عددی که میتواند در خود جای دهد ۱۰۲۳ میباشد . پس می توان نتیجه گرفت که ولتاژ ورودی بین ۰ تا ولتاژ مرجع را به عددی بین ۰ تا ۱۰۲۳ تبدیل میکند . پس محدوده ی ولتاژ ورودی را به ۱۰۲۳ قسمت مساوی تقسیم میکند . پس اگر ولتاژ مرجع ۵ ولت باشد حداقل ولتاژی که میتواند اندازه گیری کند برابر $5/1023$ میباشد که برابر ۴.۸ میلی ولت میباشد . بدین صورت که اگر ولتاژ ورودی ۰ باشد عدد خروجی ۰ خواهد بود و اگر ولتاژ ورودی ۴.۷ میلی ولت باشد که خروجی باز هم صفر است ولی اگر ولتاژ ورودی ۴.۸ میلی ولت باشد خروجی عدد ۱ را نشان میدهد .

حال مشخص شد که اگر بخواهیم میزان دقت مبدل آنالوگ به دیجیتال را افزایش دهیم باید مقادیر ۲ کمیت را تغییر دهیم .
«مقدار حداکثر مبدل آنالوگ و یا مقدار ولتاژ مرجع»

در بین این دو کمیت تنها میتوان مقدار ولتاژ مرجع را تغییر داد. پس اگر این ولتاژ را کاهش دهیم مقدار حداقل ولتاژ قابل اندازه گیری توسط مبدل آنالوگ به دیجیتال کاهش پیدا میکند و می توان گفت که دقت افزایش پیدا کرده است .

آزمایش ۹: کار با تایمر های میکروکنترلر

۹-۱) برنامه ای بنویسید تا تایمر ۱ را با حداقل فرکانس کاری راه اندازی کرده و مقدار آن را همواره بر روی LCD نمایش دهد.

نحوه ی کار با تایمر ها را در جزوه ی تئوری به دقت مطالعه کنید.

برای انجام این آزمایش مراحل کاری را به صورت زیر انجام دهید.

۱- میکروکنترلر مورد استفاده را با استفاده از دستور `$regfile = "m32def.dat"` تعریف کنید.

۲- برای راه اندازی تایمر با حداقل فرکانس کاری (همان طور که در صورت مثال ذکر شده است) میکروکنترلر را با

فرکانس کاری ۱ مگاهرتز راه اندازی کنید. با استفاده از دستور `$crystal=1000000` این کار را انجام دهید.

۳- تنظیمات مربوط به LCD را همانند آزمایشات قبلی انجام دهید.

۴- حال با استفاده از دستور مربوط به تنظیم و پیکره بندی `Timer`، تنظیمات مربوط به تایمر را انجام داده و آن را با

حداکثر نرخ تقسیم (prescaler) ممکن تنظیم کنید.

۵- تایمر را راه اندازی کنید.

۶- در درون برنامه ی اصلی در حلقه ی `do ... loop`، دستور نوشتن مقدار تایمر را بر روی LCD بنویسید.

```
$regfile = "m32def.dat"
```

```
$crystal = 1000000
```

```
Config Timer1 = Timer , Prescale = 1024
```

```
Start Timer1
```

```
Do
```

```
Cls
```

```
Home
```

```
Lcd Timer1
```

```
Waitms 5
```

```
Loop
```

```
End
```

```
'end program
```

این برنامه را نوشته و کامپایل کرده و بر روی میکرو کنترلر برنامه ریزی کنید .

در روی LCD مقدار تایمر ۱ نشان داده میشود که با فرکانس ۹۷۶.۵۶۲۵ هرتز شروع به شمارش میکند . با توجه به این که مقدار حداکثر تایمر برابر ۶۵۵۳۵ میباشد بعد از رسیدن به این مقدار دوباره از ۰ شروع میشود .
نحوه ی محاسبه ی فرکانس کاری تایمر :

$$F_t = F_m / \text{prescale}$$

F_t نرخ تقسیم , F_m فرکانس کاری میکرو , فرکانس کاری تایمر

$$F_t = 1000000 / 1024 = 976.5625 \text{ Hz}$$

۲-۹) برنامه ای بنویسید تا تایمر ۱ را با حداقل فرکانس کاری راه اندازی کرده و در هر بار سرریزی تایمر وضعیت یک LED را بر عکس کند. (استفاده از وقفه ی تایمر)

همان طور که در مثال قبلی به آن اشاره شد تایمر در حال شمارش بعد از رسیدن به مقدار حداکثر خود به مقدار حداقل خود (که در این مثال از تایمر ۱ با مقادیر حداقل ۰ و حداکثر ۶۵۵۳۵ میباشد) باز میگردد . در حین این بازگشت یک وقفه به cpu ارسال میکند که نام این وقفه ، وقفه ی سرریزی تایمر ۱ میباشد . از این وقفه میتوان در برنامه استفاده کرد و از آن برای انجام بسیاری از فرایندها به دقت زیادی نیاز دارد استفاده کرد . برای این که بتوان از این وقفه استفاده کرد باید ابتدا یک آدرس برای آن مشخص کرد . برنامه ای که باید در صورت ایجاد این وقفه اجرا شود در این آدرس نوشته میشود . (منظور از آدرس یک برچسب میباشد) . بعد از اتمام برنامه ی مورد نظر باید دستور `return` نوشته شود .

دستوری که آدرس زیر برنامه ی وقفه ی سرریزی تایمر را مشخص میکند دستور زیر میباشد .

On ovf1 adress

دستور فعال سازی وقفه دستور زیر میباشد .

Enable ovf1

باید دستور فعال سازی کل وقفه ها نیز صادر شود . این دستور دستور زیر میباشد .

Enable interrupts

برنامه :

```
$regfile = "m32def.dat"
$crystal = 1000000
```

```
Config Pind.4 = Output
```

```
Config Timer1 = Timer , Prescale = 1024
```

```
On Ovf1 Adress
```

```
Start Timer1
```

```
Do
```

```
Loop
```

```
Adress:
```

Toggle Portb.4

Return

End

'end program

AZARAN



۳-۹) برنامه ای با استفاده از تایمر ۱ میکروکنترلر بنویسید تا در هر ۱ ثانیه ۱ بار LED1 را روشن و خاموش کند.

برای نوشتن این برنامه نیز همانند آزمایش قبلی باید از زیر برنامه ی وقفه ی سرریزی تایمر استفاده کنیم. در این آزمایش از ما خواسته شده است تا به فاصله های ۱ ثانیه LED را روشن و خاموش کنیم. برای این کار باید با استفاده از فرمول زیر سرریزی تایمر ۱ را محاسبه کنیم.

$$F_t = F_m / \text{prescale}$$

$$F_{ovf} = F_t / \text{timer max}$$

نرخ تقسیم = prescale - فرکانس کاری میکرو Fm - فرکانس کاری تایمر Ft

مقدار حداکثر تایمر = timer max - فرکانس سرریزی Fovf

برای مقادیر آزمایش قبلی فرکانس سرریزی تایمر ۱ برابر است با:

$$F_t = 1000000 / 1024 = 976.5625$$

$$F_{ovf} = 976.5625 / 65535 = 0.0149$$

یعنی فرکانس سرریزی تایمر ۱ با فرکانس کاری ۱ مگاهرتز و نرخ تقسیم ۱۰۲۴ بابر است با ۰.۰۱۴۹ که زمان تناوب آن برابر خواهد بود با ۶۷.۱۰۷۸۴. که این عدد بیانگر این است که تایمر ۱ در هر ۶۷.۱۰۷۸۴ ثانیه یک بار سرریز میشود.

در این آزمایش از ما خواسته شده است تا با استفاده از وقفه ی سرریزی تایمر ۱ زمان ۱ ثانیه را تولید بکنیم. برای افزایش فرکانس سرریزی و یا همان کاهش زمان تناوب آن باید مقدار نرخ تقسیم را کاهش دهیم. مقدار نرخ تقسیم را از ۱۰۲۴ به ۲۵۶ کاهش میدهم و دوباره محاسبات قبلی را انجام میدهم.

$$F_t = 1000000 / 256 = 3906.25$$

$$F_{ovf} = 3906.25 / 65535 = 0.0596$$

$$T_{ovf} = 1 / 0.0596 = 16.77$$

همان طور که مشاهده میشود باز هم این مقدار خیلی بزرگ است. حال مقدار نرخ تقسیم را از ۲۵۶ به ۱ کاهش میدهم و دوباره این محاسبات را انجام میدهم.

$$F_t = 1000000 / 1 = 1000000$$

$$F_{ovf} = 1000000 / 65535 = 15.25$$

$$T_{ovf} = 1 / 15.25 = 0.06 \text{ s}$$

مشاهده می کنید که در هر ثانیه ۱۵.۲۵ بار سرریز میشود. برای این که بتوانیم این ۱۵ بار در ثانیه را به ۱ بار در ثانیه تبدیل کنیم باید یک مقسم فرکانس نرم افزاری در زیر برنامه ی وقفه ی تایمر ۱ بنویسیم. این مقسم باید این طوری نوشته شود که هر بار زیر برنامه ی وقفه ی سرریزی تایمر اجرا شد مقدار یک متغیر را یک واحد افزایش دهد. در واقع با این کار ما تعداد دفعات سرریزی تایمر ۱ را میشماریم. حال کافی است بعد از همین شمارش مقدار آن را بررسی کنیم و ببینیم مقدار آن به ۱۵ رسیده است یا نه؟ اگر مقدار آن برابر با ۱۵ شده بود به این معنی است که تایمر ۱۵ بار سرریز شده است. همان طور که



قبلاً محاسبه کردیم تایمر با فرکانس کاری فوق در هر ثانیه ۱۵ بار سرریز میشود. پس شمارش ۱۵ در زیر برنامه ی سرریزی تایمر به معنی گذشت ۱ ثانیه میباشد.

در این روش اختلاف تنها ۰.۲۵ بار در ثانیه است که بعد گذشت ۴ ثانیه ۱ بار بیشتر سرریزی اتفاق میافتد که برای جبران آن باید بعد از ۳ بار شمارش از ۰ تا ۱۵، چهارمین دفعه شمارش را تا ۱۶ انجام دهد.

در کاربردهای دقیق باید فرکانس کلاک میکرو و مقدار نرخ تقسیم را طوری تنظیم کنیم تا این عدد قسمت اعشاری نداشته باشد.

```
$regfile = "m32def.dat"
$crystal = 1000000
```

```
Config Pind.4 = Output
```

```
Config Timer1 = Timer , Prescale = 1
On Ovf1 Adress
Start Timer1
```

```
Dim Del As Byte
```

```
Do
```

```
Loop
```

```
Adress:
```

```
Incr Del
```

```
If Del >= 15 Then
```

```
Toggle Portb.4
```

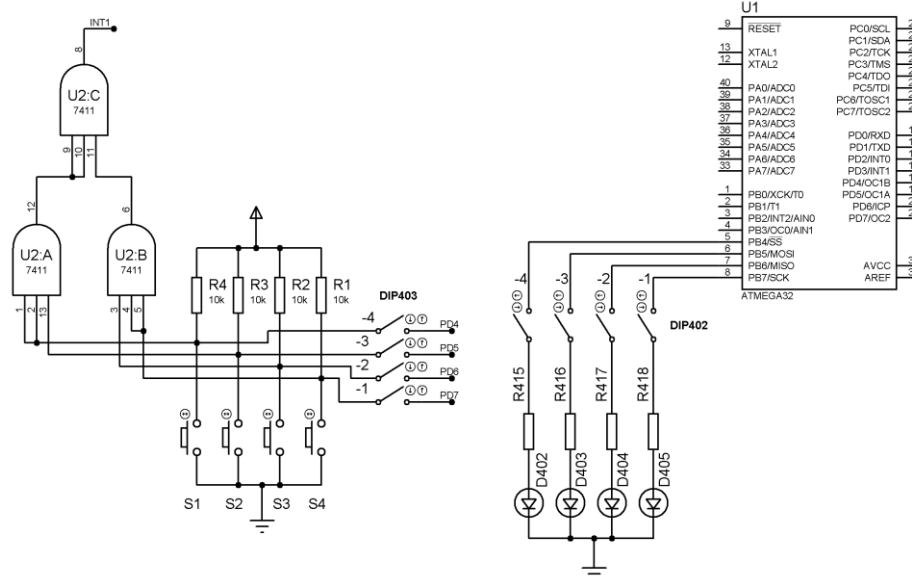
```
End If
```

```
Return
```

```
End
```

```
'end program
```

۴-۹) برنامه ای بنویسید تا با استفاده از تایمر ۱ مدت زمانی را که کلید S1 در حالت فشار داده شده قرار دارد را نشان دهد.



همان طور که در نقشه مشخص شده است کلید S1 به پایه ی PD4 از میکرو کنترلر متصل شده است و زمانی که فشار داده میشود وضعیت این پایه از ۱ به ۰ تغییر میکند. پس باید در حلقه ی اصلی برنامه وضعیت این پایه را بررسی کنیم و اگر در مقدار آن تغییری ایجاد شد تایمر را راه اندازی کند و زمانی که مقدار آن به حالت اول برگشت تایمر را متوقف کند و مقدار تایمر را در روی LCD نمایش دهد.

پس برنامه را به این شکل مینویسیم که در حلقه ی اصلی Do Loop با استفاده از دستور IF وضعیت پایه را بررسی میکنیم که اگر برابر با ۰ باشد تایمر ۱ را راه اندازی کند و سپس منتظر باشد تا مقدار این پایه به سطح منطقی ۱ باز گردد.

```
If pind.4= 0 then
    Start timer1
    bitwait pind.4 , set
    Stop timer0
    Cls
Home
Lcd timer1
Endif
```

برنامه ی زیر را بنویسید :

```
$regfile = "m32def.dat"
$crystal = 1000000
```

Config Pind.4 = Input

Config Timer1 = Timer , Prescale = 1

Do

If Pind.4 = 0 Then

Start Timer1

Bitwait Pind.4 , Set

Stop Timer1

Cls

Home

Lcd "Time=" ; Timer1 ; " "

End If

Loop

End

'end program



۵-۹) برنامه ای با استفاده از تایمر ۲ میکروکنترلر بنویسید تا این تایمر را به صورت آسنکرون راه اندازی کرده و با استفاده از کریستال خارجی ۳۲.۷۶۸ کیلو هرترت زمان ۱ ثانیه را تولید کند (این ۱ ثانیه را با استفاده از چشمک زدن LED1 نشان دهد)

یکی از ویژگی های بارز میکروکنترلر AVR این است که با استفاده از تایمر ۲ میتوان یک تایمر آسنکرون ساخت که حتی در زمان هایی که میکروکنترلر در حالت بی کاری قرار دارد نیز کار می کند و میتواند زمان بسار دقیقی را ارائه دهد. برای این که بتوان از این تایمر آسنکرون استفاده کرد باید یک کریستال ۳۲.۷۶۸ کیلو هرترت به پایه های TOSC1 , TOSC2 از میکروکنترلر متصل باشد. در سخت افزار بردهای مجموعه ی آزمایشی این کریستال ها در نظر گرفته شده اند. شما فقط کافی است تا برنامه ی مربوط به آن را بنویسید. دستور مربوطه برای راه اندازی این تایمر دستور زیر میباشد.

`Config clock = soft , gosub = sectic`

با این دستور در واقع تایمر ۲ بصورت تایمر آسنکرون راه اندازی میشود و شروع به کار میکند و در هر ۱ ثانیه ۱ بار سرریز میشود. بعد از هر بار سرریزی به زیربرنامه ی sectic رفته و برنامه های نوشته شده در آن زیر برنامه را اجرا میکند. دقت این ۱ ثانیه ی تولیدی توسط این تایمر بسیار دقیق میباشد. محاسبات مربوط به این تایمر نیز به این صورت میباشد: این تایمر آسنکرون از تایمر ۲ استفاده میکند و همانطور هم که میدانید تایمر ۲، ۸ بیتی است و ۲۵۶ حالت دارد. به صورت پیش فرض نرخ تقسیم فرکانس آن ۱۲۸ در نظر گرفته میشود. فرکانس کاری این تایمر در حالت آسنکرون ۳۲.۷۶۸ کیلو هرترت میباشد.

پس:

$$F_{timer} = 32.768\text{kHz} / 128 = 256$$

$$F_{ovf2} = 256 / 256 = 1$$

مشاهده میشود که تعداد دفعات سرریزی این تایمر در هر ثانیه ۱ بار میباشد. این عدد بسیار دقیق میباشد چرا که هیچ گونه رقم اعشاری ندارد. پس بسیار مناسب است برای انجام کارهایی که در آنها به فرکانس دقیق 1Hz نیاز است. برنامه ی این آزمایش بصورت زیر نوشته میشود:

`$regfile = "m32def.dat"`

`$crystal = 1000000`

`Config Pinb.7 = Output`

`Config Clock = Soft , Gosub = Sectic`
`Enable Interrupts`

`Do`

Loop

Sectic:

Toggle Portb.7

Return

End

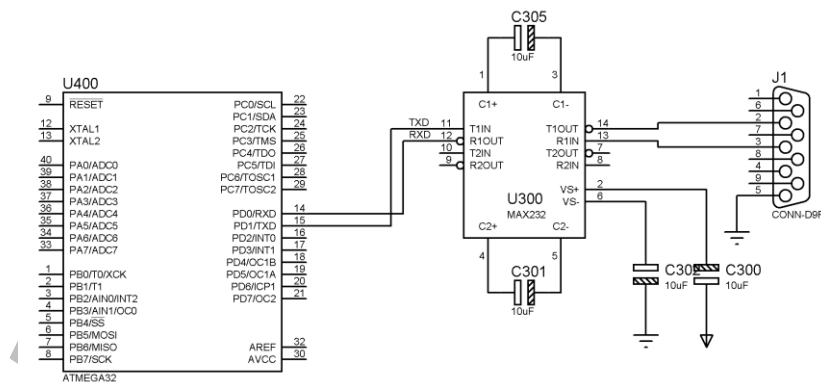
'end program

DRAFT

آزمایش ۱۰: راه اندازی و استفاده از پورت سریال میکروکنترلر

۱-۱) پورت سریال میکرو را با نرخ انتقال ۹۶۰۰ راه اندازی کنید و عدد ۲۰ را در هر ثانیه یک بار به خروجی پورت سریال ارسال کنید.

اگر در مورد مباحث تئوری مربوط به پورت سریال و نحوه ی ارسال اطلاعات به پورت سریال اطلاعات کافی ندارید بهتر است ابتدا به جزوه تئوری مربوط به میکرو کنترلر AVR به زبان بیسیک مراجعه کنید سپس به انجام این آزمایش پردازید. برای انجام این آزمایش باید از ماچول Interface استفاده کنید. این ماچول بر روی برد ۱ از مجموعه ی آزمایشگاهی در نظر گرفته شده است. ابتدا باید این کانکتور BOX2*5 مربوط به این ماچول را که با نام J302 مشخص شده است را به کانکتور BOX2*5 به نام interface که در ماچول اصلی در برد ۱ قرار دارد متصل کنید. بعد از این اتصال نقشه ی پورت سریال بصورت زیر در می آید.



حال برای این که بتوانید از ماچول interface استفاده بکنید باید کلید On/Off مربوطه را فشار دهید تا تغذیه ی این ماچول تامین شود.

در این آزمایش حتما "از برد شماره ۱ استفاده کنید چرا که فرکانس کریستال خارجی که برای U400 در برد ۱، 11.0592Mhz در نظر گرفته شده است که این فرکانس کلاک برای ارتباط با پورت سریال بهترین فرکانس میباشد. چرا که کمترین درصد خطا را دارد. بنابراین اطلاعات ارسال شده به پورت سریال بدون هیچ گونه خطایی به مقصد میرسند. برای ارسال اطلاعات بین میکرو و کامپیوتر از نرخ انتقال اطلاعات ۹۶۰۰ بیت در ثانیه استفاده کنید. برای این کار باید با استفاده از دستور $\$baud = 9600$ استفاده کنید. توجه داشته باشید که این عدد باید در دستگاه فرستنده و گیرنده برابر باشد. اگر کوچکترین اختلافی در بین این دو عدد در فرستنده و گیرنده وجود داشته باشد دستگاه گیرنده توانایی خواندن اطلاعات از پورت سریال را نخواهد داشت. برنامه را به صورت زیر مینویسیم.

```
$regfile = "m32def.dat"
```

```
$crystal = 32059200
```

```
$baud = 9600
```

```
Do
```

```
Print 20
```

```
Waitms 1000
```

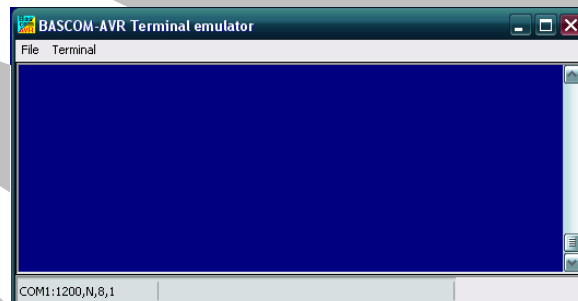
```
Loop
```

```
End
```

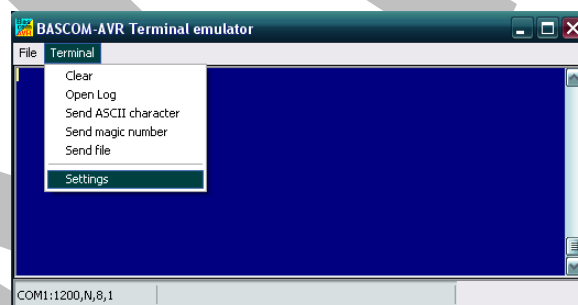
```
'end program
```

حال بعد از نوشتن این برنامه و کامپایل کردن و برنامه ریزی کردن میکروکنترلر میکرو شروع به ارسال اطلاعات به پورت سریال میکند .

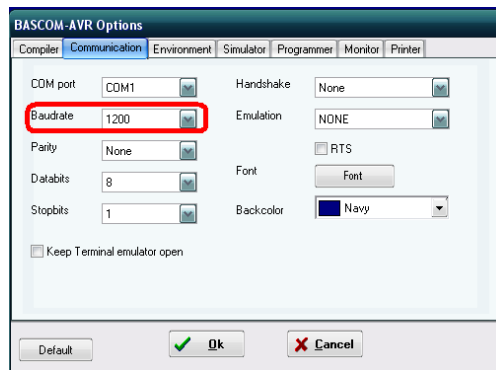
برای این که بتوانید اطلاعات دریافت شده را در کامپیوتر مشاهده کنید باید محیط **terminal emulator** را باز کنید . برای این کار از منوی **tools** بر روی گزینه ی **terminal emulators** کلیک کنید .



پنجره ی **terminal emulator** ظاهر میشود . در این پنجره تمام اطلاعاتی که از پورت سریال وارد کامپیوتر می شود نمایش داده میشود . برای تنظیم نرخ انتقال اطلاعات در ثانیه از منوی **terminal** گزینه ی **settings** را انتخاب کنید .



حال در این پنجره عدد مقابل **baudrate** را به ۹۶۰۰ تغییر دهید .



سپس دکمه ی OK را فشار دهید. حال اگر همه ی مراحل را به درستی انجام داده باشید باید عدد ۲۰ را که از میکروکنترلر ارسال میشود را در درون کادر آبی رنگ مشاهده کنید.

۲-۱۰) پورت سریال میکرو را در حالت گیرنده فعال کنید و اطلاعات ارسالی از پورت سریال کامپیوتر را دریافت کرده و بر روی LCD نمایش دهید.

(این آزمایش ها را با نرم افزار Terminal Emulator کامپیوتر انجام دهید)

نکته ی قابل توجه در مورد محیط terminal emulator این است که زمانی که این محیط فعال است با فشار دادن هر یک از کلید های صفحه کلید کامپیوتر کد اسکی مربوطه به پورت سریال کامپیوتر ارسال میشود. سرعت ارسال این کد به روی پورت سریال نیز برابر با مقدار عدد تنظیم شده در گزینه ی Baudrate میباشد.

حال باید در میکروکنترلر برنامه ای بنویسید تا داده های دریافت شده از پورت سریال را بر روی نمایشگر LCD نمایش دهد.

این کار را با دستور waitkey() انجام میدهیم. این دستور منتظر میماند تا داده ای از پورت سریال دریافت گردد. بدین معنی که تا زمانی که داده ای از پورت سریال دریافت نگردیده است دستورات بعد از این دستور اجرا نخواهند شد.

```
$regfile = "m32def.dat"
$crystal = 32059200
```

```
$baud = 9600
```

```
Dim Indata As Byte
```

```
Do
```

```
Indata = Waitkey()
```

```
Cls
```

```
Home
```

```
Lcd Indata
```

```
Loop
```

```
End
```

```
'end program
```

آزمایش ۱۱: کار با موتور پله ای

۱-۱) برنامه ای بنویسید تا با استفاده از میکروکنترلر AVR، step motor را به صورت نیم پله راه اندازی کرده و به صورت ساعت گرد موتور پله ای را به چرخش در آورد.

موتور پله ای STEPER MOTOR

موتور پله ای وسیله پرمصرفی است که پالس های الکتریکی را به حرکت مکانیکی تبدیل میکند. در کاربردهایی همچون راه انداز های دیسک سخت، چاپگر های مغناطیسی و رباتیک، از موتور پله ای برای کنترل موقعیت استفاده میشود. هر موتور پله ای دارای یک هسته متحرک مغناطیسی دائمی است که روتور یا شفت هم خوانده میشود و بوسیله ی یک بخش ثابت به نام استاتور احاطه شده است. معمول ترین موتور های پله ای، چهار سیم استاتور دارند که با سر وسط جفت شده اند. این نوع موتور ها معمولاً "به موتور پله ای چهار فاز معروفند. موتور پله ای که در این برنامه استفاده شده است دارای ۶ سیم است: ۴ سیم برای چهار سیم پیچ استاتور و دوسر دیگر که به ولتاژ تغذیه متصل میگردند. با اعمال رشته هایی از تغذیه یا پالس به هر سیم پیچ استاتور، روتور خواهد چرخید. رشته های مرسوم متفاوتی موجودند که هر یک دقت متفاوتی را دارا هستند. جدول زیر یک رشته چهار پله نرمال را نشان میدهد

باید توجه داشت که اگر چه میتوان با هر یک از رشته های موجود در جدول فوق آغاز کرد ولی به محض شروع باید ترتیب رعایت شود.

زاویه پله

خلاف	سیم پیچ	سیم پیچ	سیم پیچ	سیم پیچ	# پله	در جهت
جهت	D	C	B	A		ساعت
ساعت	۰	۰	۱	۱	۱	↓
	۰	۱	۰	۰	۲	
	۰	۰	۱	۰	۳	
	۱	۰	۰	۰	۴	

موتور را نشان

زاویه پله مشخصه

میدهد که به ازاء هر پله موتور چند درجه چرخش دارد . جدول زیر بعضی از زوایا را برای انواع موتور ها نشان میدهد .

زاویه پله ی موتور	تعداد پله در دور
۰.۷۲	۵۰۰
۱.۸	۲۰۰
۲.۰	۱۸۰
۲.۵	۱۴۴
۵.۰	۷۲
۷.۵	۴۸
۱۵	۲۴

پله در ثانیه و دور در دقیقه (RPM)

رابطه بین RPM یا دور در دقیقه ، پله در دور (SPT) و پله در ثانیه (SPS) از رابطه زیر بدست می آید :

$$SPT = RPM * SPS / 60$$

یک چرخش کامل

موتور پله ای مورد استفاده در این پروژه ، دارای زاویه پله ۱.۸ درجه میباشد و برای یک دور چرخش کامل نیازمند ۲۰۰ پالس میباشد ($360 / 1.8 = 200$) . با شمارش پالسها ، میکروکنترلر AVR میتواند موقعیت موتور را تنظیم کرده و آن را کنترل کرد .

راه اندازی نیم پله

تحرك نیم پله موجب میشود که دقت موتور دو برابر شود . در این حالت با توجه به موتور استفاده شده نیاز به ۴۰۰ پالس میباشد جدول رشته نیم پله صفحه بعد منطق مورد نیاز برای برنامه را نشان میدهد . وقتی به انتهای جدول رسیدید ، این روند را تکرار کرده و دوباره از ابتدای جدول شروع کنید .

برنامه موتور پله ای

در این پروژه برای راه اندازی موتور پله ای از راه انداز ULN2003 استفاده شده است. این درایور ۱۶ پایه ، دارای ۷ راه انداز کلکتور باز است و بیشترین جریان خروجی آن برای هر پایه 500mA می باشد. جدول زیر چهار نوع از خانواده ULN را نشان میدهد.

ULN2001A	General PURPOSE , DTL , TTL , Pmos ,CMOS
ULN2002A	14-2 5 V , CMOS
ULN2003A	7 TTL , CMOS
ULN2004A	6 – 15V , CMOS , PMOS

جدول چهار نوع از خانواده ULN

(آی سی ULN2803 همانند آی سی ULN2003 میباشد با این تفاوت که دارای ۱۸ پایه میباشد و دارای ۸ کانال کلکتور باز برای راه اندازی موتور پله ای می باشد.)

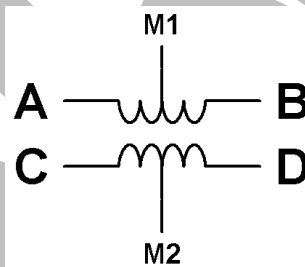
راه اندازی موتور پله ای توسط مجموعه آموزشی میکرو کنترلرهای AVR سری AP8-AZ-SH برای راه اندازی موتور پله ای در این برد از آی سی ULN2003 استفاده شده است که ورودی های این آی سی در کانکتور JP1 که با نام Motor مشخص شده است. برای این که بتوانید به پایه های آی سی ULN2003 دسترسی داشته باشید باید این کانکتور را به یکی از کانکتورهای 5*2box که به پورت های میکروکنترلر متصل است ، وصل کنید. این کار را با یک کابل رابط IDC انجام دهید.

خلاف جهت ساعت	# پایه	سیم پیچ A	سیم پیچ B	سیم پیچ C	سیم پیچ D	جهت ساعت
	1	1	0	0	0	
	2	1	1	0	0	
	3	0	1	0	0	
	4	0	1	1	0	
	5	0	0	1	0	
	6	0	0	1	1	
	7	0	0	0	1	

1 0 0 1 8

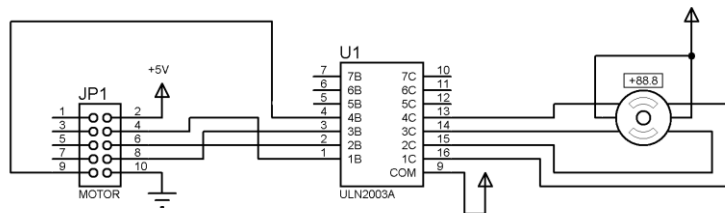
نحوه ی تشخیص پایه های تغذیه و کنترلی مربوط به step motor :

همان طور که در شکل زیر مشخص شده است ، موتور پله ای دارای ۶ پایه میباشد که از ۲ عدد بوبین تشکیل شده است . این بوبین ها سه سر میباشد که دو سر کناری برای کنترل موتور به کار میروند و سر های میانی برای تغذیه ی بوبین . برای تشخیص پایه های فوق از اهم متر استفاده کنید همان طور که در شکل مشخص میباشد پایه ی مشترک هر یک از بوبین ها با سر های کناری مقاومت یکسانی را از خود نشان میدهند در حالی که دو سر کناری مقاومت معادل با دو برابر سر مشترک را نشان میدهند . و از طرف دیگر تنها ۳ عدد از پایه ها با هم و سه پایه ی دیگر با یکدیگر ارتباط دارند . بنابراین میتوان سیم های دو بوبین را از همدیگر تشخیص داد و نیز سر های وسطی هر یک از بوبین ها را نیز تشخیص داد . پایه های A-B را نیز باید به صورت عملی تست کرده و پیدا کنید .

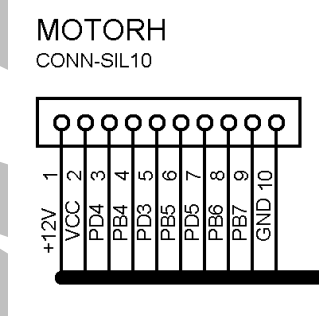


مدار مربوط به موتور پله ای در برد شماره ۳ بصورت شکل زیر میباشد .

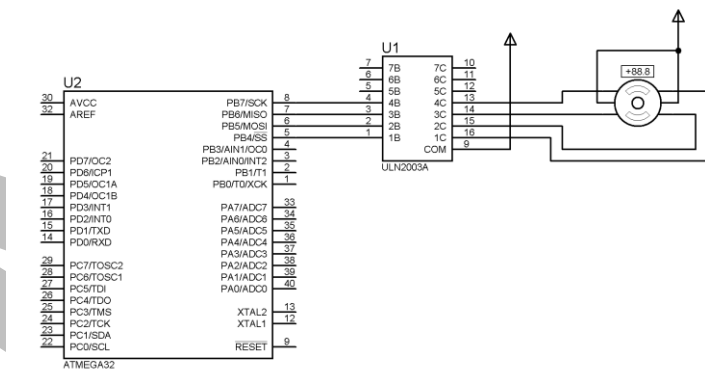
نکته :
لازم به ذکر است که موتور پله ای و آرسی مربوطه در روی برد شماره ۳ قرار دارد اما کانکتور JP1 در روی برد شماره ی ۲ قرار دارد .



بعد از اتصال کانکتور Jp1 به کانکتور Motor بر روی ماجول اصلی در برد ۲ اتصالات بین میکرو و مدار موتور پله ای به صورت زیر در می آید . در زیر اتصالات مربوط به کانکتور Motor نشان داده شده است .



همان طور که در شکل نشان داده شده است پایه ی ۱ از این کانکتور به +12 متصل شده است . پایه ی ۲ به VCC و پایه ی ۳ به PD4 و الی آخر .





بعد از این اتصال مدار به صورت شکل بالا در می آید . حال باید داده های مربوط به پورت های اتصالی به موتور پله ای را تغییر دهید . برنامه ی موتور پله ای یکی از آسان ترین برنامه های کاربردی میکروکنترلر میباشد . تنها کاری که از نظر برنامه نویسی باید انجام دهیم این است که داده های قرار داده شده در خروجی پورت را با سرعت دلخواه تغییر دهیم . سرعت تغییر داده های روی پورت تعیین کننده ی سرعت چرخش موتور پله ای خواهد بود .

حال باید طبق جدول های ارائه شده در بالا پایه های PB4 تا PB7 را تغییر دهید . یک نمونه برنامه در زیر آورده میشود .

```

$regfile = "m32def.dat"
$crystal = 1000000
Config Portb = Output
Dim Motor As Byte
Motor = 1
Do
  Portb= 0
  Set Portb.4
  Waitms 100
  Portb = 0
  Set Portb.5
  Waitms 100
  Portb = 0
  Set Portb.6
  Waitms 100
  portb = 0
  Set Portb.7
  Waitms 100
Loop
End

```



۲-۱۱) برنامه ای بنویسید تا با استفاده از میکرو کنترلر AVR ، Step motor را به صورت تمام پله راه اندازی کرده و آن را این با به صورت خلاف ساعت گرد راه اندازی کنید . تفاوت راه اندازی به صورت نیم پله و تمام پله را توضیح دهید .

```

$regfile = "m32def.dat"
$crystal = 1000000
Config Portb = Output
Dim Motor As Byte
Motor = 1
Do
  Portb = &B00010000
  Waitms 100
  Portb = &B00110000
  Waitms 100
  Portb = &B00100000
  Waitms 100
  Portb = &B01100000
  Waitms 100
  Portb = &B01000000
  Waitms 100
  Portb = &B11000000
  Waitms 100
  Portb = &B10000000
  Waitms 100
  Portb = &B10010000
Loop
End

```

در راه اندازی به صورت نیم پله تعداد پله در دور موتور پله ای دوبرابر میشود پس در نتیجه دقت موتور دو برابر میشود . تنها تفاوت آن با راه اندازی تمام پله در داده هایی است که بر روی پایه های موتور پله ای چرخش داده میشود .

دستور کار جلسه دوازدهم

هدف از انجام آزمایش : فراگیری کار با موتور پله ای

آزمایش ۱۲ : کار با موتور DC

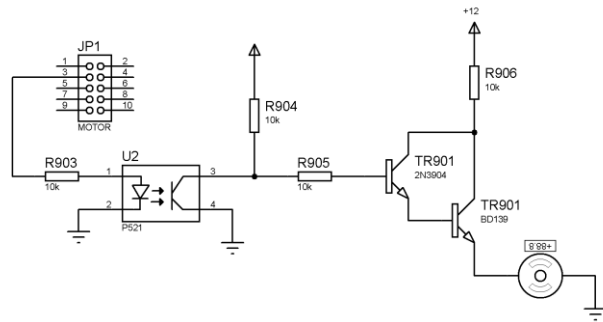
در این آزمایش قصد داریم شما را با نحوه ی راه اندازی موتور DC آشنا کنیم . موتور های DC یکی از پرکاربرد ترین موتور ها هستند که روز به روز نیز کاربرد آنها افزایش پیدا میکند . برای این که بتوانید سرعت موتور DC را تنظیم کنید چند راه دارید :

- ۱ - کنترل جریان
- ۲ - کنترل فرکانس کاری موتور
- ۳ - کنترل از طریق پالس PWM

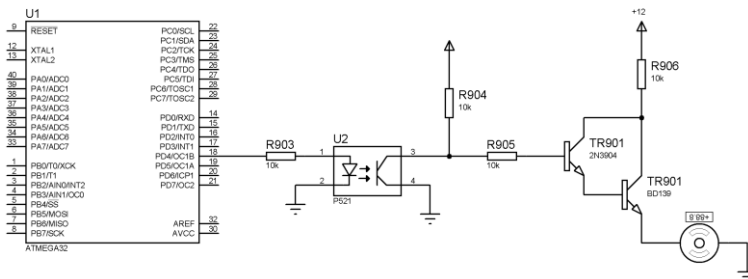
بهترین روش برای کنترل دور موتور DC استفاده از پالس PWM میباشد . در این روش جریان و ولتاژ کاری موتور ثابت نگه داشته میشود و موتور با یک فرکانس ثابت راه اندازی میشود و تنها عاملی که تغییر پیدا میکند عرض پالس فرکانس مربوطه میباشد .

در مجموعه ی آزمایشی میکرو کنترلر مدل AZ-SH AP8 برای انجام این آزمایش یک عدد موتور DC به همراه مدار راه انداز مربوطه در نظر گرفته شده است . توجه داشته باشید که اساس کار موتور های DC یکسان است و اگر شما بتوانید کار با این موتور را یاد بگیرید دیگر تفاوتی برای شما در اندازه و قدرت موتور نخواهد داشت .

برای انجام این آزمایش باید کانکتور j1 در قسمت کانکتور های برد شماره ی ۳ که در بالای برد شماره ۲ قرار دارد را به کانکتور Motor که در ماجول اصلی برد شماره ۲ قرار دارد متصل کنید . این اتصال را با استفاده از کابل های رابط IDC انجام دهید . بعد از این اتصال ، ارتباطات بین میکروکنترلر و مدار درایور موتور DC به صورت زیر خواهد بود .



نقشه ی مدار مربوط به موتور DC



نقشه ی اتصالات بین میکرو و مدار موتور DC (بعد از اتصال کانکتور jp1 به کانکتور Motor با استفاده از کابل رابط IDC)

همان طور که در نقشه ی بالا نیز مشخص شده است بعد از این اتصال پایه ی ورودی اپتوکوپلر توسط یک مقاومت به پایه ی Portd.4 متصل شده است . حال برای این که بتوانیم موتور پله ای را راه اندازی کنیم باید به این پایه • منطقی اعمال کنیم . برای این کار باید ابتدا پایه ی پورت را به صورت خروجی تعریف کرده و مقدار آن را برابر با یک قرار دهیم . با صفر کردن این پایه خروجی اپتوکوپلر از حالت اشباع خارج شده و ولتاژ ورودی بیس ترانزیستور TR901 بالا می رود و در نتیجه زوج دارلینگتون شامل از دو ترانزیستور جریان را به سوی موتور DC هدایت میکنند .

اگر بخواهید موتور با حداکثر سرعت خود حرکت کند باید خروجی این پایه را همواره در سطح منصفی • قرار دهید . برای این که بتوانید سرعت موتور را کنترل کنید باید یک پالس با پهنای متغیر به این پایه اعمال کنید . در این آزمایش قصد داریم این پالس را توسط دستورات ساده ی waitms و set و reset انجام دهیم . در این برنامه دو عدد دستور wait دو عدد reset , set قرار دارد . در واقع یک بار موتور را روشن میکنیم و بعد از یک زمان کوتاهی دوباره آن را روشن می کنیم و بعد از مدت زمان کوتاهی روشن بودن دوباره خاموش میکنیم . این مدت

زمان های روشن بودن و خاموش بودن تعیین میکند که موتور با چه سرعتی به چرخش درآید . پس با تغییر این زمانها میتوان سرعت موتور را کنترل کرد .

```
$regfile = "m32def.dat"
$crystal = 1000000
Config Portd = Output
Do
  Set Portd.4
  Waitms 10
  Reset Portd.4
  Waitms 15
Loop
End
```

در این برنامه برای این که بتوان سرعت چرخش موتور را کنترل کرد تنها کافی است تا زمان تاخیر های موجود برای `reset` و افزایش داد .

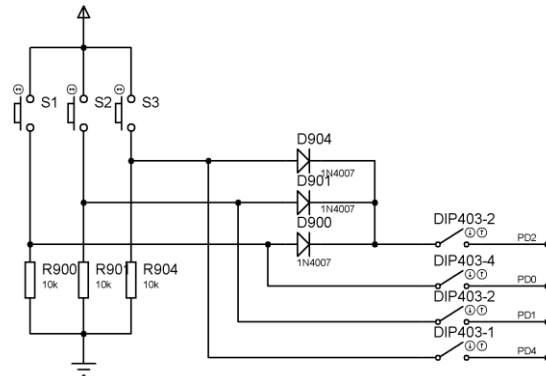
این برنامه میتواند دور موتور را کنترل کند اما یک اشکال اساسی دارد ؟

اشکال این برنامه این است که وقت `CPU` را حدر میدهد برای رفع این مشکل باید از تایمر استفاده کرد و با استفاده از تایمر، یک پالس `PWM` تولید کرد تا این پالس `PWM` وظیفه ی کنترل دور موتور را بر عهده بگیرد و از حلقه ی اصلی برنامه بتوان برای کار های دیگری استفاده کرد .

برای همین منظور پایه ی اتصالی به موتور `DC` از پایه های خروجی `PWM` مربوط به تایمر ۱ استفاده شده است . این پایه `oc1b` نام دارد . این پایه خروجی مقایسه کننده ی شماره ۲ ی تایمر ۱ میباشد . حال برنامه ای بنویسید الف : تا در این پایه یک پالس `PWM` داشته باشیم ؟

ب : با استفاده از کلید های روی برد بتوان عرض پالس این سیگنال را کنترل کرد و کلید `S1` برای افزایش و کلید `S2` برای کاهش آن ؟

جواب : نقشه ی کلید های نصب شده به میکرو کنترلر در برد شماره ۲ به صورت زیر میباشد .



بعد از این که برنامه مربوط به تایمر را به صورت **pwm** راه اندازی کردید ، تایمر شروع به تولید پالس **pwm** میکند . برای این که بتوانید پالس **pwm** فوق را تنظیم کنید باید در صورت فشار داده شدن کلید **S1** مقدار متغیر **pwm1b** را یک واحد افزایش دهید و در صورت فشار داده شدن کلید **S2** مقدار متغیر **pwm1b** را یک واحد افزایش دهید . دستوری که با استفاده از آن میتوانید فشار داده شدن کلید **s1** و یا **s2** را تشخیص دهید ، دستور **debounce** میباشد .

DEBOUNCE Px.y , state , label [, SUB]

Px.y	A port pin like PINB.0 , to examine.
State	0 for jumping when PINx.y is low , 1 for jumping when PINx.y is high
Label	The label to GOTO when the specified state is detected
SUB	The label to GOSUB when the specified state is detected

در این دستور ابتدا باید نام پایه ای از پورت را که کلید به آن متصل شده است را وارد کنید . سپس باید کمیتی را که در صورت فشار داده شدن کلید به پایه اعمال می شود . بعد از آن نام زیر برنامه ای را که باید بعد از فشار داده شدن کلید اجرا شود را وارد کنید و بعد از آن کلمه ی **SUB** را وارد کنید . این در صورتی است که بخواهید از زیر برنامه ی مربوطه برگردید و ادامه ی برنامه را از بعد از دستور **debounce** ادامه دهید . اگر نخواهید برنامه به مکان فعلی برنگردد می توانید این گزینه را وارد نکنید .

```
$regfile = "m32def.dat"
$crystal = 1000000
Config Portd = Output
Config Timer1 = Pwm , Prescale = 256 , Noise Cancel = 1 , Capture Edge = _
```

Falling , Clear Timer = 1 , Compare B = Toggle , Pwm = 8

Do

Debounce Pind.0 , 1 , Lable1 , Sub
Debounce Pind.1 , 1 , Lable2 , Sub

Loop

Lable1:
Incr Pwm1b
Return

Lable2:
Decr Pwm1b
Return

End