

# نویز Noise

نشریه تخصصی برق و الکترونیک ECA / سال دوم / شماره ۱۱ / دی - ۱۳۹۱



● نکات طراحی سخت افزار مدارات مبتنی بر AVR

● نرم افزار Jimbo LCD

● ارتباط Labview با انواع PLC از طریق OPC SERVER

● آشنایی با GSM Modem و نحوه ارتباط آن با میکروکنترلر

● کاهش ریبیل گشتاور در موتورهای DC بدون جاروبک

 [www.Noisemagazine.ECA.ir](http://www.Noisemagazine.ECA.ir)



## منشور مجله نویز

### الف- معرفی

ماهنامه نویز، نشریه ای است علمی، خبری، آموزشی که با تکیه بر خلاقیت و نوآوری کارشناسان ایرانی و انجام کار گروهی آنان به مخاطبان خود کمک می کند تا نیازهای خود را در شکل مکتوب در این نشریه محقق شده ببابند.

### ب- اهداف

- ۱- پیشبرد فرهنگ علم گرایی در کشور
- ۲- ایجاد روحیه تلاش و امید در نسل جوان
- ۳- توسعه کمی و کیفی روزافزون به سوی نشریه ای وزین، فراگیر و کم اشکال
- ۴- اطلاع رسانی به روز علوم مرتبط

### ج- اصول و ارزش ها

- ۱- احترام به مخاطب
- ۲- راستی در گفتار و رفتار میان دست اندر کاران نشریه
- ۳- تلاش مستمر برای بهبود
- ۴- استفاده از پتانسیل موجود در قشر تحصیل کرده کشور

### د- حوزه های فعالیت (موضوعات کلیدی)

- ۱- آموزش مبانی علوم برق و الکترونیک
- ۲- آشنایی با تکنولوژی های نوین دنیای فناوری
- ۳- طراحی مدارات آنالوگ و دیجیتال
- ۴- آموزش و معرفی نکات کاربردی پردازنده ها
- ۵- مدارات مجتمع
- ۶- آموزش کاربردی نرم افزارهای تخصصی
- ۷- سیستم ها و مدارات مخابراتی
- ۸- سیستم های قدرت
- ۹- معرفی مشاهیر برق
- ۱۰- موارد کاربردی و عملی علم الکترونیک
- ۱۱- بخش های متنوع مرتبط با موضوعات برق و الکترونیک

\* استفاده از مقاله های مجله، با ذکر مأخذ و رعایت حقوق نویسنده بلامانع است.

\* مجله نویز آماده دریافت آثار و مقالات ارسالی متخصصین و مهندسین است.

\* لطفاً مقاله های خود را بصورت تایپ شده به همراه ضمیمه عکس های مورد استفاده ارسال نمایید.

\* نشریه در ویرایش و اصلاح مطالب رسیده، آزاد است.

\* چنانچه مطالب ارسالی ترجمه است، کپی اصل آن را ضمیمه نماید.

صندوق پست الکترونیکی مجله : noisemagazine.eca@gmail.com



فهرست مطالب :

- تشریح بازده نزولی LED ..... ۲
- ساخت اسیلوسکوپ با استفاده از میکروکنترلر LPC1768 ..... ۴
- در بازکن الکترونیکی خانه ..... ۱۰
- آشنایی با GSM Modem و نحوه ارتباط آن با میکروکنترلر ..... ۲۶
- آموزش طراحی تقویت کننده های ترانزیستوری ..... ۳۲
- نکاتی در مورد طراحی سخت افزار مدارات مبتنی بر AVR ..... ۳۸
- JIMBO LCD ..... ۴۸
- ارتباط LabVIEW با انواع PLC از طریق OPC SERVER ..... ۵۲
- سیستم های متداول شاسی در ساخت ربات ..... ۵۶
- کاهش ریل گشتاور در موتورهای DC بدون جاروبک با استفاده از
- تک سنسور جریان DC ..... ۷۰
- معرفی بخشی از تجهیزات مورد استفاده در سیستم های کنترلی مبتنی
- بر PLC ..... ۷۸
- پروتکل ارتباط سریال ..... ۹۰
- Edwin Howard Armstrong ..... ۱۰۸
- Transistor Wars ..... ۱۱۰

مترجم: پویا محمدیان [pooya\\_b150@yahoo.com](mailto:pooya_b150@yahoo.com)

نشریه بازده فزونی LED

ایده جدید، نظریه ها را به هم نزدیک و محققان را از هم دور می کند.

جامعه LED درگیر یک بحث طولانی و ممتد درباره منشأ یک پدیده به نام droop شده است که وقتی جریان بالا می رود، بازدهی امیترهای (منتشر کننده های) آبی و سفید کاهش می یابد. حل این معما منجر به طرح معماری LED بر اساس پدیده droop خواهد شد که نور یکپارچه درخشان تر و ارزان تر تولید می کند. اخیراً بحث پدیده droop با رقابت تئوری های بسیاری که می خواهند حمایت گروه های متخصص الکترونیک نوری را کسب کنند گرم شده است. اکنون یک نظریه پرداز در آزمایشگاه های ملی سانديا در آلبوکرکی راهی را برای ترسیم روش های انتشار نور با یکدیگر پیدا کرده است. ونگ چاو رفتار LED را از ساختار نواری آنها محاسبه می کند، این رفتار انرژی هایی را که حاملان شارژ می توانند در درون قطعه داشته باشند و احتمال پیدایش حامل هایی با آن انرژی ها را مشخص می کنند.

هر LED علیرغم رنگ آن با تزریق الکترون ها درون قطعه از یک طرف و راندن آنها به نقطه مثبت مقابل (حفره ها) در سمت دیگر عمل می کند. هر دو نوع از حامل ها در یک شیار باریک که به عنوان چاه کوانتوم شناخته می شود برخورد می کنند. اینجا آنها به دام می افتند و از طریق جاذبه الکترو استاتیک به یکدیگر می چسبند و با ترکیب، نور منتشر می کنند.

در LED قرمز این چاه های کوانتوم دقیقاً مانند همان ها در کتاب های مکانیک کوانتومی مقدماتی می باشند و احتمال پیدا کردن الکترون ها و حفره ها در این شیارها بسیار است. اما LED آبی و یا سفید با میدان های الکتریکی داخلی قوی دچار مشکل می شوند. چاو شرح می دهد وقتی که جریان به آرامی از LED عبور می کند، الکترون ها و حفره ها به ندرت به حد نصاب لازم جهت انتشار نور می رسند. میدان الکتریکی آنها را از هم جدا می کند.

در یک مدل همانطور که جریان بالا می رود، تعداد الکترون ها و حفره ها بیشتر شده و بخشی از آن میدان الکتریکی داخلی را متعادل می کند. اگر چه حتی اغلب حاملان حالا در چاه هستند، پدیده droop هم آنجا قرار می گیرد، زیرا میدان الکتریکی داخلی هنوز آنقدر قوی هست که الکترون ها را به یک سمت از شیار و حفره ها را به سمت دیگر براند که باعث تاخیر در انتشار نور شود. در این مدل حتی بالا بردن بیشتر جریان تأثیر میدان داخلی را خنثی می کند، بازدهی انتشار را افزایش می دهد و منجر به بهبود droop، پدیده ای که در تضاد با آنچه در LED واقعی است، می شود.

LED های سفید و آبی پر از نقص در شبکه های کریستالی شان هستند که در چاه های کوانتوم نیرات گالیوم ایندیم آنها متداول تر است. زمانی که چاو این جزئیات را در مدل خودش در نظر می گیرد که می تواند بهبود پدیده droop را نقض کرده و رفتار واقعی این قطعات را تقلید کند. در این حالت، مدل چاو دو نظریه برجسته برای علت نواقص پدیده droop در LED و باز ترکیب حامل ناکارآمد در چاه های کوانتوم را یکی می کند. اما او به سختی یک قدم پیش تر رفت. او معروفترین تئوری droop از تمام باز ترکیب های موثر که یک فعل و انفعال بدون انتشار نور از سه حامل شارژ را راندن یک الکترون و یا یک حفره به سطح انرژی بالاتر می باشد را بوجود آورد. بنابراین پیش گویی های نظری چاو برای بازدهی LED گرایش های اصلی دیده شده در نتایج آزمایشی را بازتاب می دهد.

به عقیده اومیت اوزگور از دانشگاه کامنولث ویرجینیا، کار چاو جلب نظر می کند. او بر این باور است، یکی از نقاط قوت این تئوری شکل دادن پدیده شناخته شده فیزیکی واقعی که در طی انتشار نور رخ می دهد می باشد. به طور ساده مدلی که برای شرح رفتار LED و در ارتباط با droop استفاده می شود، در عمل موفق نمی شود. اوزگور اکنون از چاو می خواهد که داده های آزمایشی از یک محدوده قطعات واقعی را با مدل او تطبیق بدهد.

مانوس کیوپاکیس و کریس ون د وال از دانشگاه کالیفرنیا در سانتا باربارا از منتقدان کار چاو هستند. آنها اشاره می کنند که در مدل چاو، droop به دما حساس است و بحث می کنند که این یافته با نتایج آزمایشی متناقض است.

محقق سانديا مخالفت می کند و مدعی است که تناقضی وجود ندارد زیرا مدل چاو آنقدر که تیم دانشگاه کالیفرنیا می گویند به دما حساس نیست. او همچنین اشاره می کند که کار یورگ هایدلر و همکارانش از دانشگاه آریزونا می تواند رفتار LED را در دماهای مختلف با یک مدل شبیه مدل چاو که شامل نواقص است تکرار کند.

آنچه واضح است اینکه بحث droop هیچ نشانه ای از فروکش کردن ندارد. هنگامیکه چاو نظریه های برجسته را با یکدیگر جمع می کند در متحد کردن نظریه پردازان مسئول موفق نیست.

منبع:

IEEE Spectrum

## آنچه شما در داندلود سنتر ECA خواهید یافت :

### مقالات علمی، آموزشی

مقالات علمی و آموزشی در تمام زمینه های برق و الکترونیک و علوم مرتبط  
آموزش زبان های برنامه نویسی، طراحی مدارات، تکنولوژی های نوین و ...  
آموزش نرم افزارهای تخصصی برق و الکترونیک



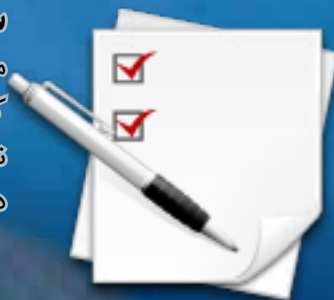
### پروژه های تکمیل شده

پروژه های کاملاً عملی و تست شده به همراه توضیحات کامل در زمینه های تخصص برق و الکترونیک  
میکروکنترلرهای AVR، PIC و ARM  
مدارهای مجتمع خطی، مدارات آنالوگ، کنترل، مخابرات و ...



### سوالات کنکور

مجموعه سوالات کنکورهای کارشناسی ناپیوسته، کارشناسی ارشد و دکترا  
نمونه سوالات آزمون های ورودی و استخدامی  
دانشگاه های سراسری، آزاد و پیام نور



### جزوات دانشگاهی

مجموعه جزوات درسی در مقاطع مختلف و دانشگاه های مختلف کشور  
گزارش کارهای آزمایشگاهی  
جزوات کنکور و درسی برترین اساتید کشور



### کنفرانسی های داخلی

مقالات معتبرترین کنفرانس های داخلی کشور  
کنفرانس های کشوری و بین المللی برگزار شده در دانشگاه ها و ارگان های دولتی



دانلود ECA  
www.Download.ECA.ir



# ساخت اسیلوسکوپ با استفاده از میکروکنترلر LPC1768



هدف از انجام این پروژه، ساخت یک اسیلوسکوپ ساده با استفاده از میکروکنترلر LPC1768 می باشد. برای سهولت کار با این تراشه از برد آموزشی NXP LPC 1768 استفاده نموده ایم ولی در صورت نیاز، این پروژه با استفاده از ماژول LCD ۳،۲ اینچ و هدربرد LPC1768 نیز قابل پیاده سازی است.

عملکرد پروژه

با استفاده از پتانسیومتر روی برد (کانال ۵ مبدل ADC میکروکنترلر) ولتاژ پایه تغییر می کند و ما این تغییرات ولتاژ را بر روی LCD ۳،۲ اینچ موجود بر روی برد، نمایش می دهیم. در صورتی که جامپر شماره JP12 مربوط به ADC را از روی مدار بردارید، می توانید با اعمال سیگنال خارجی آنالوگ مورد نظر به پایه P1.31 میکروکنترلر، نمودار سیگنال مورد نظر را نمایش دهید. به دلیل سرعت بالا، به برنامه تاخیر اضافه نموده ایم تا تغییرات سیگنال به راحتی قابل نمایش باشند. کلید Int0 جهت Pause نمودن نمودار در نظر گرفته شده تا ضمن آموزش عملکرد Int خارجی، بتوان برای نگه داشتن نمایش سیگنال رئی صفحه نمایش و ایجاد امکان محاسبه مقدار سیگنال توسط کاربر استفاده نمود.

کلید ۱ برای انتخاب Time Devision و Volt Devision در نظر گرفته شده است و با فشردن متوالی این کلید میتوان یکی از این ۲ مورد را انتخاب نمود.  
از JoyStick برد برای بالا و پایین نمودن میزان Time Div و Volt Div اسیلوسکوپ استفاده می شود و میزان موارد انتخابی به صورت لحظه ای بر روی LCD نمایش داده می شود.

شرح برنامه

جهت آشنایی با دستورهای مورد استفاده به صورت بلوک بلوک این موارد توضیح داده می شوند:

```
void EINT3_IRQHandler()
{
    LPC_GPIOINT->IO2IntClr |= (1 << 10);
    if(pause) pause=0;
    else pause = 1;
}
```

زیر برنامه مربوط به اینترپت پایه Int0 است که جهت Pause و Resume کردن نمودار تعریف شده است.

```
if (!(LPC_GPIO2->FIOPIN & (1<<11))) { // Volt
    if (selected==0) selected=1;
    else selected=0;
}
if ((LPC_GPIO1->FIOPIN & (1<<29))==0) // UP
{
    if (selected==0)
    {
        if(volt_div<=4) volt_div*=2;
    }
    else if(time_div<=4) time_div*=2;
}
else if ((LPC_GPIO1->FIOPIN & (1<<26))==0) // DOWN
{
    if (selected==0)
    {
        if(volt_div>1) volt_div/=2;
    }
    else if(time_div>1) time_div/=2;
}
```

این بخش برای انتخاب و تغییر مقادیر Volt Div و Time Div می باشد. متناسب با انتخاب هر یک از این موارد، کلید های Up و Down جوی استیک می تواند مقدار را بالا و پایین بیاورد.

```
LCD_ShowNum(220,16,volt_div,1,16);
LCD_ShowNum(220,32,time_div,1,16);
POINT_COLOR = RED;
if (selected==0)
{
    LCD_DrawLine (220,31,227,31);
}
else {
    LCD_DrawLine (220,47,227,47);
}
POINT_COLOR = Black;
```



با این بلوک مقدار Volt Div و Time Div بر روی LCD نمایش داده می شود و متناسب با انتخاب کاربر، زیر مورد انتخابی خط قرمز (به نشانه انتخاب) کشیده می شود.

```
LPC_GPIO1->FIODIR  &= ~((1<<25)|(1<<26)|(1<<27)|(1<<28)|(1<<29)); LPC_GPIO2->FIODIR
&= ~((1<<11)|(1<<22)); // KEY1 = Volt Division  KEY2 = Time Division
```

```
LPC_GPIO2->FIODIR    &= ~(1 << 10);  /* PORT2.10 defined as input    */
LPC_GPIOINT->IO2IntEnF |= (1 << 10);  /* enable falling edge irq    */
```

```
NVIC_EnableIRQ(EINT3_IRQn);
```

این بخش از تابع main برنامه جهت انتخاب و معرفی پایه های مورد استفاده به صورت ورودی و خروجی و فعال سازی وقفه INT0 می باشد.

```
LCD_Clear(White);
POINT_COLOR=Blue;
for (x=0;x<4;x++)
{
    LCD_DrawLine(x,0,x,319); // y origin line
    LCD_DrawLine(0,x,219,x); // x origin line
}
POINT_COLOR=Black;
for (x=1;x<7;x++) {
    LCD_DrawLine(x*40,0,x*40,319); // y line
    LCD_DrawLine(0,x*53,239,x*53); //x line
}

LCD_ShowString(160,0,"www.ECA.ir");
LCD_ShowString(160,16,"Volt=");
LCD_ShowNum(220,16,volt_div,1,16);
LCD_ShowString(160,32,"Time=");
LCD_ShowNum(220,32,time_div,1,16);
LCD_SetTextColor(Red);
for (x=0;x<320;x++) {
```

بخش فوق مربوط به حلقه اصلی نمایش نمودار بر روی LCD می باشد که طبق برنامه، پس از هر بار تکرار تابع کل صفحه پاک شده و به صورت ۶\*۶ جدول بندی می شود. قسمت پایین دستورات مربوط به نمایش Time Div و Volt Div و عبارت www.ECA.ir در گوشه تصویر می باشد.

```
while(pause);
```

این دستور، برنامه را هنگام فشردن دکمه Pause وارد حلقه می کند و باعث مکث نمودار می شود.

```
ADC_Data = 0;
for(i = 0;i < 8; i++)
{
    ADC_Buf  = ADC_Get();
    ADC_Data += ADC_Buf;
}
ADC_Data = (ADC_Data / 8);
```

# ساخت اسیلوسکوپ با استفاده از میکروکنترلر LPC1768

مقدار ADC ۸ بار توسط این دستورات خوانده میشود که این عمل به دلیل کاهش نویزپذیری اوسیلوسکوپ می باشد.

```
ADC_Data = (ADC_Data * 240 * volt_div)/4096;
```

پس از خوانده شدن مقدار ADC و با استفاده از روابط ریاضی موجود در دستور بالا، مقدار ADC از حالت ۱۲ بیتی به رنج مناسب LCD تبدیل شده و برای نمایش در LCD ارسال می شود.

```
if (ADC_Data<240) {  
    LCD_PutPixel(319-ADC_Data,x);  
    LCD_PutPixel(318-ADC_Data,x);  
    LCD_PutPixel(320-ADC_Data,x);  
    LCD_PutPixel(319-ADC_Data,x+1);  
    LCD_PutPixel(319-ADC_Data,x-1);  
}
```

این بخش جهت نمایش پیکسل محاسبه شده بر روی LCD می باشد. برای اینکه پیکسل نمایش داده شده پررنگ-بزرگ باشد ۴ پیکسل نزدیک به پیکسل اصلی نیز روشن می شوند.

```
for (j=0;j<time_div;j++) {  
    delay(18);  
    if(++dly>25) update_settings(),dly=0; // update Settings every 450ms  
}
```

این حلقه جهت ایجاد تاخیر برای اعمال Time Div و آپدیت وضعیت کلیدها در بازه زمانی 450ms تعریف شده است.

## PIC USB PROGRAMMER



جهت استفاده از میکروکنترلرهای PIC ساخت شرکت میکروچیپ شما به یک پروگرامر احتیاج خواهید داشت. روش های متعددی جهت پروگرام کردن میکروهای PIC از قبیل استفاده از پورت پرینتر و یا سریال وجود دارد، ولی در دنیای امروزی این پورت ها به علت کم بودن موارد مصرف دیگر بر روی رایانه های شخصی نصب نبوده و حتی بر روی رایانه های همراه نیز دیگر وجود نداشته و شما می بایست با خرید وسایل جانبی این پورت ها را نصب نمایید. ولی در عوض بر تعداد پورت های USB موجود بر روی رایانه ها افزایش یافته و امروزه رایانه ای بدون پورت USB موجود نمی باشد. این پروگرامر نیز بر اساس تکنولوژی USB طراحی گشته و به این معنی بوده که شما می توانید در هر رایانه از آن استفاده نمایید. در زیر خلاصه ای از مزایای این پروگرامر لیست شده است.

### Device Programming Support

- PIC10
- PIC12
- PIC16
- PIC18
- PIC24
- dsPIC30
- dsPIC33
- PIC32
- KEELOQ® HCS
- 11 Series Serial EEPROM
- 24 Series Serial EEPROM
- 25 Series Serial EEPROM
- 93 Series Serial EEPROM
- MCP250xx CAN

- o هماهنگی با پورت های USB2.0
- o توانایی پروگرام نمودن تمامی میکروکنترلرهای PIC با بسته بندی DIP از طریق یک زیف سوکت
- o پشتیبانی از طریق نرم افزارهای مورد تأیید شرکت میکروچیپ
- o بدون نیاز به تغذیه خارجی (تغذیه از طریق پورت USB)
- o سوکت icsp به منظور پروگرام کردن انواع میکروکنترلرهای pic خارج از پروگرامر
- o قابلیت شناسایی و DETECT خودکار مدل میکروکنترلر
- o قابلیت به روز رسانی Firmware به صورت خودکار
- o قابلیت پروگرام کردن میکروهای ۳.۳ ولتی
- o بدون نیاز به نصب درایور
- o سازگاری با تمامی سیستم عامل ها (Linux/Mac OS X /Windows)
- o دارای لایه محافظ زیر بورد
- o ۱۲ ماه گارانتی تعویض

توضیحات تکمیلی : <http://eshop.eca.ir/link/620.php>



# برد آموزشی NXP ARM LPC1768



برد آموزشی Cortex-M3 LPC1768 بر اساس NXP LPC1768 متعلق به شرکت NXP بوده که بر اساس هسته نسل دوم میکروکنترلرهای ARM Cortex-M3 طراحی گشته است. این هسته برای کاربردهای سیستم های نهفته (embedded) در سرعت های بالا، توان مصرفی کم و پردازش ۳۲ بیتی طراحی گشته است. از جمله اهداف طراحی این نسل از پردازنده ها می توان به مصارف اندازه گیری، ارتباطات صنعتی، کنترل موتورهای صنعتی، سیستم های هوشمند و رباتیک اشاره نمود.

برد آموزشی Cortex-M3 LPC1768 دارای پروگرامر آنبرد (On Board) جیلینک (Jlink) بوده و از تجهیزات USB2.0 و ۲ اینترفیس CAN و همچنین RS-485 پشتیبانی می کند. این برد حاوی مثال های آموزشی بسیار زیادی جهت فراگیری کامل میکروکنترلرهای ARM سری LPC17XX می باشد.

## NXP ARM LPC1768 DevBoard Cortex-M3 3. 2" touchscreen LCD, with JLink V8 On board

### Features:

- Powerful MCU core: Cortex-M3
- processing rates of up to 100MHz, and includes a support eight areas of memory protection unit (MPU)
- Built-in Nested Vectored Interrupt Controller (NVIC)
- 512KB on-chip Flash program memory, to support in-system programming (ISP) and In-Application Programming (IAP)
- 64KB SRAM for high-performance CPU through the instruction bus, system bus, data bus access
- AHB multi-layer matrix with 8-channel general purpose DMA controller (GPDMA)
- support the SSP, UART, AD / DA, Timer, GPIO, etc. , can be used for memory to memory transfers
- standard JTAG test / debug interface and debug serial line and serial line tracking port options
- Emulation trace module supports real-time tracking
- 4 low-power modes: sleep, deep sleep, down, deep power-down
- single 3.3V power supply (2.4V - 3.6V)
- non-maskable interrupt (NMI) input
- On-chip integrated power-on reset circuit
- built-in system tick timer (SysTick), easy operating system migration.

### A wealth of on-board resources:

- 2-way RS232 serial interface (using the straight-through serial cable, serial port support them all the way ISP download)
- 2 Road, CAN bus communication interface (CAN transceiver: SN65VHD230)
- RS485 communication interface (485 transceiver: SP3485)
- RJ45-10/100M Ethernet network interface (Ethernet PHY: DP83848)
- DA output interface (USB sound card can do experiments on-board speaker and speaker output driver)
- AD input (adjustable potentiometer input)
- color LCD display interface (which can take 2.8-inch or 3.2-inch color TFT 320X240)
- USB2.0 Interface, USB host and USB Device interface.
- SD / MMC card (SPI) interface (available with FAT12, FAT16, FAT32 file system)
- IIC Interface (24LC02)
- SPI serial FLASH interface (AT45DB161D)
- 2 user keys, 2 function keys and INT0 Reset button, 8 LED lights
- 1 Five keys to enter the joystick (Joystick)
- serial ISP download (no need to set jumpers), the standard JTAG download, simulation debugging interface.
- integrated USB emulator: emulator debug features such as online support, support KEIL, IAR and other mainstream development environment.
- optional external 5V power supply or USB 5V input supply.
- leads all the IO, user-friendly external circuit to the secondary development of other applications.

توضیحات تکمیلی : <http://eshop.eca.ir/link/853.php>

# LPC1768 Cortex-M3 Header Board



**Cortex**  
Intelligent Processors by ARM®

هدربرد (برد راه انداز) میکروکنترلرهای CORTEX M3 با امکانات اولیه جهت کار با میکروکنترلرهای LPC1768 شرکت فیلیپس. با توجه به ۱۰۰ پایه بودن میکرو، دو ردیف پین هدر در اطراف برد قرار گرفته که بر روی ۲ بردبرد چسبیده به هم قابل قرارگیری است. از دیگر مزایای این برد راه انداز، تأمین ولتاژ مورد نیاز میکرو از طریق پورت USB به همراه کلید قطع وصل می باشد. یکی از امکانات ویژه این برد قابلیت پروگرام نمودن میکرو از طریق پورت USB می باشد. در این روش شما هیچ احتیاجی به سخت افزار خاصی نداشته و فقط از طریق وصل نمودن کابل USB به رایانه می توانید میکروکنترلر را پروگرام نمایید. جهت پروگرام کردن این میکرو می توانید از ۲ طریق ارتباط پورت USB و یا پورت JTAG اقدام نمایید.

❖ حداقل مدار جهت راه اندازی میکروکنترلر LPC1768

❖ بدون نیاز به پروگرامر، مجهز به بوت لودر USB

❖ دارای خروجی تمام پایه های ورودی خروجی به ترتیب شمارشی

❖ دارای کانکتور full speed USB 2.0

❖ امکان نصب مستقیم برد بر روی بردبرد

❖ دارای خروجی ولتاژهای ۳.۳ و ۵ ولت

❖ دارای کلید قطع و وصل تغذیه

❖ امکان فعال و غیر فعال کردن پورت های تمامی امکانات جانبی میکرو، از قبیل

USB، JTAG، DEBUG، VREF و ...

❖ دارای خروجی کانکتور JTAG استاندارد

❖ دارای دکمه RESET

❖ کریستال 32.768KHZ برای راه اندازی RTC داخلی

❖ کریستال ۱۲ مگاهرتز برای راه اندازی میکرو کنترلر

توضیحات تکمیلی : <http://eshop.eca.ir/link/876.php>



❖ امکان فعال و غیر فعال کردن پورت های تمامی امکانات جانبی میکرو، از قبیل USB، JTAG، DEBUG، VREF و ...

❖ دارای خروجی کانکتور JTAG استاندارد

❖ دارای دکمه RESET

❖ کریستال 32.768KHZ برای راه اندازی RTC داخلی

❖ کریستال ۱۲ مگاهرتز برای راه اندازی میکرو کنترلر

توضیحات تکمیلی : <http://eshop.eca.ir/link/877.php>



## LPC2368

## Header Board

❖ حداقل مدار جهت راه اندازی میکروکنترلر LPC2368

❖ بدون نیاز به پروگرامر، مجهز به بوت لودر USB

❖ دارای خروجی تمام پایه های ورودی خروجی به ترتیب شمارشی

❖ دارای کانکتور full speed USB 2.0

❖ امکان نصب مستقیم برد بر روی بردبرد

❖ دارای خروجی ولتاژهای ۳.۳ و ۵ ولت

❖ دارای کلید قطع و وصل تغذیه

❖ امکان فعال و غیر فعال کردن پورت های تمامی امکانات جانبی میکرو، از قبیل USB، JTAG، DEBUG، VREF و ...

❖ دارای خروجی کانکتور JTAG استاندارد

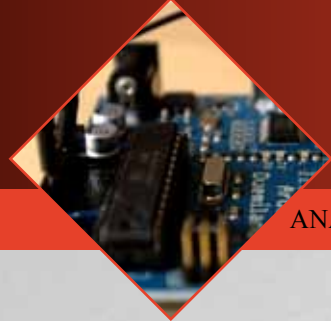
❖ دارای دکمه RESET

❖ کریستال 32.768KHZ برای راه اندازی RTC داخلی

❖ کریستال ۱۲ مگاهرتز برای راه اندازی میکرو کنترلر

توضیحات تکمیلی : <http://eshop.eca.ir/link/877.php>





# درباز کن الکترونیکی خانه

همراه با ساعت، تاریخ شمسی و اندازه گیری دمای بیرون



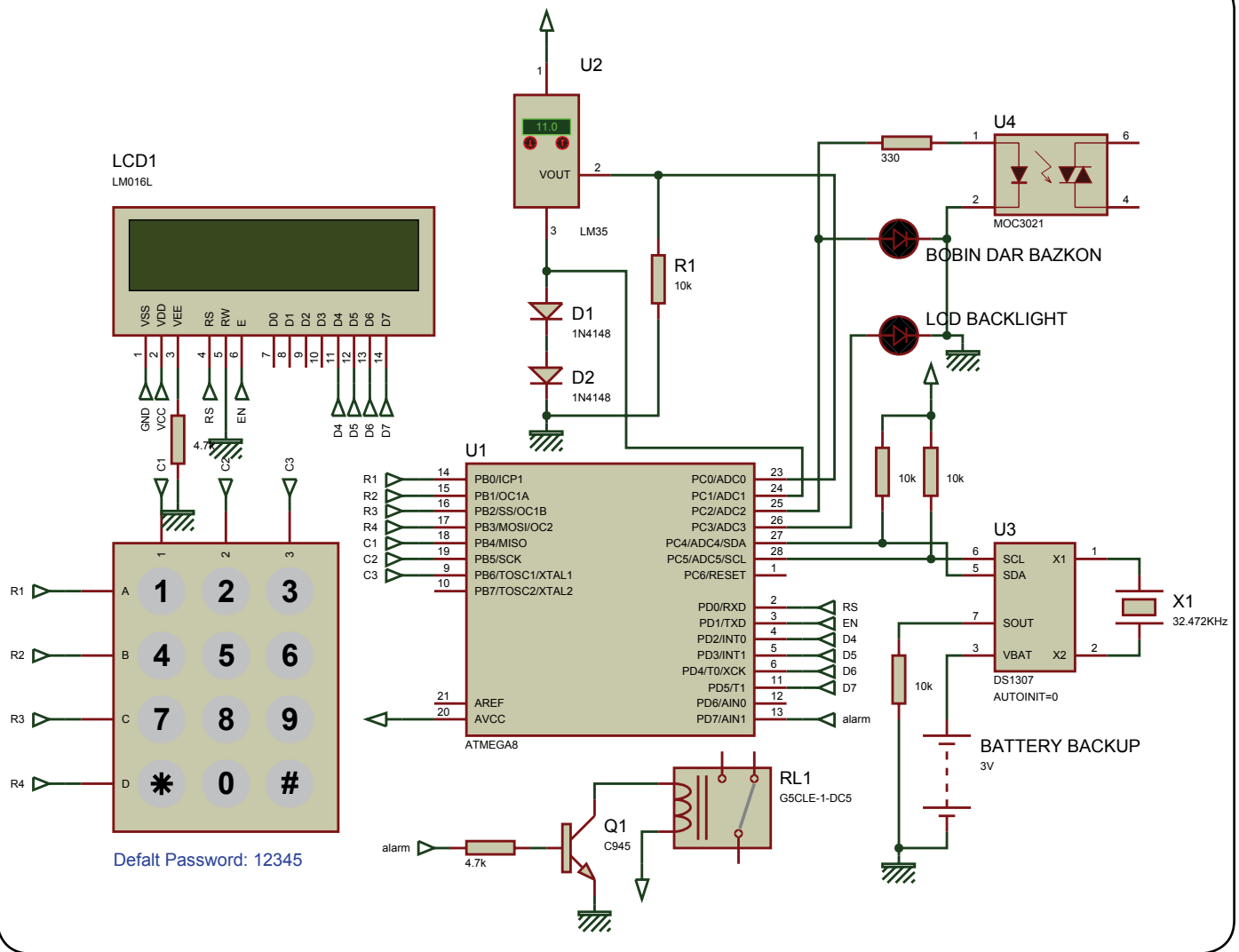
picpars@gmail.com  
www.picpars.com

نویسنده: سید محسن قاسمیان

این پروژه در اصل یک قفل امنیتی است که می توان جلوی درب اصلی منزل (در کنار آیفون) آن را نصب کرد. از امکانات آن می توان به موارد زیر اشاره نمود:

نمایش دما از ۵۵- تا ۱۲۵ درجه سانتیگراد، نمایش ساعت دقیق، نمایش تاریخ شمسی همراه با روز هفته به صورت فارسی، نمایش تعداد فرصت های باقیمانده برای وارد کردن رمز صحیح و قفل شدن سیستم به مدت ۱۵ دقیقه در صورتی که ۴ بار رمز اشتباه وارد شود. امکان تغییر رمز و ست کردن ساعت و تاریخ و...

هدف از انجام این پروژه آشنایی با IC ساعت DS1307 و کار با پروتکل I2C است که جهت ارتباط با این IC استفاده شده است. نحوی خواندن و وارد کردن اطلاعات به DS1307 و همچنین توابع تبدیل تاریخ میلادی به شمسی و سایر توابع وابسته. مدیریت زمان در میکروکنترلر و سرکشی به تمامی وظایف بدون ایجاد کوچکترین وقفه زمانی و همچنین نوشته برنامه به صورت توابع و فراخوانی تو در توی آنها را می توان از جمله ویژگی های آموزشی این پروژه به شمار آورد.



شکل ۱: شماتیک مدار

## اصول کار سیستم

هنگام اولین راه اندازی مدار، ابتدا از شما درخواست می شود که به عنوان Admin یک رمز برای سیستم انتخاب کنید، که بعد از وارد کردن یک رمز ۵ رقمی سیستم پیغام ذخیره سازی یا عدم ذخیره سازی به شما می دهد که بعد از فشار دادن کلید # سیستم آماده راه اندازی می شود. در صورتی که این مرحله را انجام ندهید هر بار که سیستم روشن شود از شما درخواست ساختن رمز می کند. بعد از طی کردن این مرحله میکرو درخواستی برای DS1307 مبنی بر ارسال متغیرهای ساعت و تاریخ می کند، در صورتی که تاریخ سیستم کوچکتر از ۱۳۹۰ باشد میکرو به صورت پیش فرض تاریخ و ساعت مشخصی را در DS1307 ست می کند و این IC تا زمانی که VCC مدار برقرار باشد راه اندازی می شود. حال اگر باتری پشتیبان در مدار باشد و VCC قطع شود آنگاه DS1307 به طور خودکار تغذیه خود را از باتری تهیه می کند و به کار خود ادامه می دهد. بنابراین بعد از روشن شدن مجدد سیستم میکرو عمل ست کردن را دیگر اجرا نخواهد کرد و وارد منوی اصلی می شویم. در اینجا دما، ساعت، تعداد فرصت های باقی مانده، تاریخ شمسی و روز هفته نمایش داده می شود. برای باز کردن درب ابتدا کلید # را فشار می دهیم سپس از شما درخواست رمز عبور می شود، در صورتی که رمز صحیح باشد اپتوکوپلر به مدت ۵۰۰ میلی ثانیه تحریک شده و در نتیجه برق ۱۲ ولت AC بوبین دربازکن تحریک می شود. (ولتاژی که اپتوکوپلر سوئیچ می کند باید AC باشد)

در هر مرحله که درخواست رمز می شود اگر هنگام وارد کردن رمز اشتباهی کردید می توانید از کلید \* برای پاک کردن اطلاعات وارد شده استفاده نمائید. برای تغییر رمز سیستم وقتی که در منوی اصلی هستید کلید \* را فشار دهید که در این حالت باید ابتدا رمز قبلی را وارد کنید و بعد هم رمز جدید، در صورتی که رمز قبلی صحیح باشد پیغامی مبنی بر ذخیره رمز جدید به شما می دهد که در صورت تایید کردن (#) رمز جدید جایگزین خواهد شد.

برای تنظیم کردن تاریخ و ساعت سیستم در منوی اصلی کلید ۱ را فشار دهید، در اینجا از شما رمز عبور درخواست می شود در صورتی که رمز صحیح باشد به منوی تنظیم ساعت خواهید رفت که باید به این فرم اعداد را وارد کنید، مثلاً برای وارد کردن عدد ۸ دقیقه باید ۰۸ وارد کنید و نکته مهم این است که تاریخ را باید برحسب میلادی وارد کنید. بعد از وارد کردن این ۶ پارامتر از شما سوال می شود که ذخیره انجام شود یا خیر. (# بله، \* خیر)

در صورتی که تا ۱۵ ثانیه هیچ کلیدی را فشار ندهید LED پس زمینه LCD خاموش خواهد شد و بعد از آن با فشار دادن هر کلیدی مجدد روشن خواهد شد. پین PD.7 که alarm نام دارد در صورت نیاز می توانید با آن یک رله را تحریک کنید و خروجی های رله را به شاسی زنگ آیفون وصل کنید. در این صورت اگر کسی به دنبال پیدا کردن رمز عبور باشد در صورتی که ۴ بار رمز اشتباه وارد کند قبل از قفل شدن



سیستم ابتدا با ریتم خاصی سه مرتبه بوق آیفون منزل شما را به صدار می آورد و شما متوجه این امر خواهید شد.

## برنامه نویسی C

### معرفی متغیر ها و کاربرد آنها

در ابتدا یک سری تعریف ها و فراخوانی توابع مختلف نوشته شده است و بعد هم متغیر های ما که همه از نوع سراسری هستند یعنی در هر تابعی می توان از محتوای آنها استفاده کرد که به شرح زیر می باشد:

```

eeprom unsigned char prog, pass[5];
char buffer[32];
unsigned char len_pass=5, pass_in[5], old_pass[5], key, op=0;
signed temp;
unsigned char err_cunt, err_cun_num=4, wait_scan_keypad=15; //second
unsigned int lcd_bl;
unsigned char scan[4]={0XFE,0XFD,0XFB,0XF7};
unsigned char arrkey[16]={
1,2,3,15,
4,5,6,14,
7,8,9,13,
10,0,11,12};
    
```

متغیر prog برای ساختن اولین رمز توسط Admin است و بعد مقدار آن به ۸۰ تغییر می کند تا این تابع دیگر هرگز اجرا نشود. آرایه پنج عضوی pass رمز ۵ رقمی مورد نظر در آن قرار می گیرد و هر دوی این متغیرها در حافظه EEPROM میکرو تعریف شده اند. متغیر buffer برای نگهداری اطلاعات ارسالی به نمایشگر است. len\_pass طول رمز عبور را مشخص می کند که در اینجا ۵ رقم می باشد که باید برابر با دو آرایه زیر باشد.

pass\_in و old\_pass آرایه ای است ۵ عضوی برای نگهداری رمزهای وارد شده توسط کاربر است. Key ارزش کلید فشار داده شده در آن قرار می گیرد. op که مقدار پیش فرض آن صفر می باشد برای عملیات خاص در نظر گرفته شده که بیشتر در تابع تغییر رمز مورد استفاده قرار گرفته است. Temp متغیر علامتدار برای نگهداری مقدار دمای خوانده شده می باشد. err\_cunt تعداد دفعات رمزهایی که اشتباه وارد می شود را می شمارد. err\_cun\_num مجوز تعداد دفعات وارد کردن رمز توسط کاربر است که بصورت پیش فرض ۴ بار است. wait\_scan\_keypad مدت زمانی است که اگر کاربر در این زمان تعیین شده (پیش فرض ۱۵ ثانیه) رمز خود را وارد نکند سیستم تقاضای او را نادید گرفته و به منوی اصلی می رود. lcd\_bl یک شمارنده می باشد و در صورتی که تا زمان مشخص شده ای کلیدی فشار داده نشود LED پس زمینه LCD خاموش می شود. آرایه scan و arrkey برای اسکن کیبورد و یافتن مقدار کلید فشار داده شده است.

### توابع موجود در فایل اصلی

در زیر توابع استفاده شده در برنامه به صورت کلی توضیح داده خواهند شد :

### تابع خواندن دما به صورت وقفه

```

unsigned int read_adc (unsigned char adc_input){
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
delay_us (10);
ADCSRA|=0x40;
while ( (ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}
    
```

این تابع به صورت وقفه کار می کند و دارای یک ورودی adc\_input است که تعیین می کند از کدام کانال مقدار را دریافت کند بعد از اتمام زمان تبدیل مقدار تبدیل شده به دیجیتال که در ADCW است برگشت داده می شود تا سایر عملیات روی آن انجام شود.

```
void get_temp (){
signed int a,b;
float t1,t2;
b=read_adc (0);
a=read_adc (1);
t1= ( (b*2. 56)/1023)*100;
t2= ( (a*2. 56)/1023)*100;
temp=t1-t2;
}
```

در این تابع دو مقدار آنالوگ از دو کانال ۰ و ۱ دریافت می‌شود که کانال صفر مقدار دمای سنسور را می‌خواند و از کانال یک هم مقدار ولتاژ دو سر زنر ها را می‌خوانیم و با کم کردن این دو مقدار از یکدیگر دمای اصلی در temp قرار می‌گیرد. عمل تبدیل به سانتیگراد هم خیلی واضح است با ضرب و تقسیم مقادیر خوانده شده بدست می‌آید، مقدار ۲,۵۶ به این علت است که ولتاژ مرجع داخلی ۲,۵۶ ولت انتخاب شده است. همانگونه که می‌دانیم رفتار یک دیود زنر در دما محیط با ولتاژ دو سر آن رابطه دارد بنابراین از این خاصیت همراه با سنسور LM35 برای اندازه گیری دمای منفی استفاده شده.

```
void display (){
get_temp ();
sprintf (buffer,"%d\ndfC %d:%d:%d\n%d/%d/%d",temp,hour,minute,second,jy,jm,jd);
lcd_clear ();
lcd_puts (buffer);
week_name ();
cunter_err_pass (err_cunt);
}
```

این تابع متغیرهای مهم و سراسری را با فرمت مشخص شده در متغیر buffer آماده سازی کرده و سپس برای نمایشگر ارسال می‌شود. توابع week\_name و cunter\_err\_pass به ترتیب روز هفته را به صورت فارسی آماده کرده و تابع بعدی تعداد دفعات باقی مانده برای وارد کردن رمز عبور را آماده و روی نمایشگر چاپ می‌کند.

```
void lcd_back_light (){
lcd_bl++;
if (lcd_bl>=250) lcd=0;
else lcd=1;
}
```

با هر بار فراخوانی این تابع یک واحد به متغیر lcd\_bl افزوده می‌شود، زمانی که به عدد ۲۵۰ رسید LED مربوطه خاموش می‌شود و غیر این صورت روشن می‌شود. این تابع، خود در تابع keypad هر ۱۰۰ میلی ثانیه اجرا می‌شود بنابراین ۲۵۰×۱۰۰ حدوداً ۲۵ ثانیه خواهد شد یعنی اگر تا ۲۵ ثانیه هیچ کلیدی فشار داده نشود LED مربوطه خاموش می‌شود. نکته: متغیر lcd\_bl با هر بار فشار دادن یک کلید مقدار آن در keypad صفر خواهد و به اصلاح زمان شارژ می‌شود.

```
unsigned char keypad (unsigned char wait){
unsigned char r,c,k=0,j;
unsigned int i,l;
DDRB=0X0F;
PORTB=0XFF;
unsigned char keypad (unsigned char wait){
```



```

unsigned char r,c,k=0,j;
unsigned int i,l;
DDRB=0X0F;
PORTB=0XFF;
l= (wait*1000)+1000;

for (i=0; i<l; i++){
for (r=0; r<4; r++){
c=4;
PORTB=scan[r];
delay_us (10);
if (c1==0) c=0;
if (c2==0) c=1;
if (c3==0) c=2;
if (c4==0) c=3;

if (! (c==4)){
k=arrkey[ (r*4)+c];
while (c1==0);
while (c2==0);
while (c3==0);
while (c4==0);
delay_ms (100);
lcd_bl=0;
return k;
}
}
delay_ms (1);
j++;
if (j==100) {lcd_back_light (); j=0;}
}

k=100; //time out "Not press any key"
return k;
}

```

وظیفه این تابع اسکن صفحه کلید به صورت ماتریسی است که دارای یک پارامتر ورودی به نام wait است. تابع صورت پیش فرض به مدت ۱ ثانیه صفحه کلید را اسکن می کند و بر می گردد حالا اگر هنگام فراخوانی این تابع مثلا عدد ۱۵ را بفرستیم آنگاه ۱+۱۵ برابر ۱۶ ثانیه صفحه کلید را اسکن خواهد کرد. نکته مهم این است که اگر در این مدت زمان هیچ کلیدی فشرده نشود تابع مقدار ۱۰۰ را برگشت خواهد داد که در ادامه برنامه ما با عدد ۱۰۰ متوجه خواهیم شد که زمان وارد کردن اطلاعات پایان یافته و کلیدی فشرده نشده بنابراین از عملیات مربوطه صرف نظر می کنیم. در اینجا فقط به این نکته اشاره خواهیم کرد که تابع از دو حلقه تو در تو تشکیل شده است که حلقه داخلی وظیفه اسکن پورت را بر عهده دارد و حلقه بیرونی مدت زمان اسکن را مشخص می کند، به این صورت که حلقه بیرونی حلقه داخلی را هر ۱ میلی ثانیه یک بار اجرا می کند. در صورتی که کلیدی فشرده شود شرط داخل حلقه داخلی اجرا می شود و مقدار واقعی کلید از آرایه انتخاب می شود، همچنین lcd\_bl صفر شده تا زمان روشن بودن LED شارژ شود و همان جا مقدار k برگشت داد می شود و CPU به ادامه برنامه بازگشت داده خواهد شد.

تابع بازنشانی متغیر های مهم

```

void reset_var (){
unsigned char i;
for (i=0; i<len_pass; i++){
pass_in[i]=255;
old_pass[i]=255;
}
}

```

وظیفه این تابع بازنشانی و پر کردن کردن متغیر مهم سیستم با عدد FF می باشد.

## تابع نمایش پیغام ذخیره سازی

```
unsigned char masage (){\n  lcd_clear ();\n  lcd_gotoxy (5,0);\n  lcd_putsf ("SAVE?");\n  lcd_gotoxy (0,1);\n  lcd_putsf ("NO (*)");\n  lcd_gotoxy (10,1);\n  lcd_putsf ("YES (#)");\n  key = keypad (wait_scan_keypad);\n  return key;\n}
```

در مکان های مختلفی از برنامه نیاز است که از کاربر سوالی مبنی بر انجام ذخیره سازی پرسیده شود، این تابع این عمل را انجام می دهد و شروع به اسکن صفحه کلید می کند و مقدار کلید فشرده شده را بازگشت خواهد داد که در ادامه برنامه در صورتی که کلید # فشرده شده بود عملیات ذخیره سازی انجام می شود و اگر کلید \* بود عملیات متوقف خواهد شد.

## تابع دریافت رمز عبور

```
void get_pass (){\n  unsigned char i,x;\n  res_get_pass:\n  lcd_clear ();\n  _lcd_write_data (0X0F);\n  if (op==0) lcd_putsf ("Input Password");\n  if (op==1) lcd_putsf ("Old Password");\n  if (op==2) lcd_putsf ("New Password");\n  for (i=0; i<len_pass; i++){ \n    x=i+5;\n    lcd_gotoxy (x,1);\n    key = keypad (wait_scan_keypad);\n    if (key==10) goto res_get_pass;\n    if (key==100 || key==11) break;\n    pass_in[i]=key;\n    sprintf (buffer,"%d",pass_in[i]);\n    lcd_puts (buffer);\n    delay_ms (150);\n    lcd_gotoxy (x,1);\n    lcd_putsf ("*");\n  }\n}
```

در این تابع ابتدا مکان نمای نمایشگر به صورت چشمک زن خواهد شد و در ادامه با توجه به مقدار متغیر op پیغام مربوطه روی نمایشگر نشان داده خواهد شد و بعد از آن به تعداد len\_pass از کاربر رمز عبور را دریافت خواهد کرد. همچنین مقدار متغیر key برای اعداد (\*), ۱۰ و (#) ۱۱ و (Time out) ۱۰۰ چک خواهد شد. در صورتی که مقدار ۱۰ باشد رمز وارد شده پاک شده و به برجسب res\_get\_pass پرش خواهد کرد و مجددا شما می توانید رمز عبور را وارد کنید. و در صورتی که مقادیر ۱۱ یا ۱۰۰ باشد حلقه شکسته (break) خواهد شد تابع به اتمام می رسد. با هر بار وارد کردن یکی از ارقام به مدت ۱۵۰ میلی ثانیه عدد نمایش داده می شود و سپس به جای آن کاراکتر "\*" نمایش داده خواهد شد.

## تابع ساختن رمز اولیه بعد از پروگرام کردن

```
void new_pass (){\n  unsigned char i;
```

```

lcd=1;
if (prog!=80){ // Create a first password
sprintf (buffer,"Admin: Create a password");
lcd_clear ();
lcd_puts (buffer);
delay_ms (3000);
get_pass ();
i=masage ();

if (i==11){
for (i=0; i<len_pass; i++){
pass[i]=pass_in[i];
}
prog=80; // burn fuse for a first get password
}
}
reset_var ();
}

```

این تابع تنها یک بار اجرا می‌شود و بعد از ساختن اولین رمز عبور دیگر اجرا نخواهد شد. بعد از پروگرام کردن میکرو محتوای EEPROM عموماً FF است، در هر صورت عدد ۸۰ نیست پس این نشان می‌دهد که هنوز رمز اولیه ساخته نشده است که بعد از ساختن آن این متغیر برابر ۸۰ خواهد شد و در راه اندازی بعدی سیستم این تابع دیگر اجرا نخواهد شد. اصول کار این تابع نیز شبیه تابع دریافت رمز عبور می‌باشد که در بالا توضیح داده شد.

#### تابع چک کردن رمز عبور

```

void check_pass (unsigned char open){ //open: 1=> rel=1
unsigned char i,c=1;
for (i=0; i<len_pass; i++){
if (pass_in[i]!=pass[i]){
c++;
break;
}
}

if (c==1){
if (open==1){
rel=1; //cunter for wait invalid password
lcd_clear ();
lcd_putsf ("Opened the door");
lcd_gotoxy (4,1);
lcd_putsf ("WELCOME");
delay_ms (500);
rel=0;
delay_ms (2000);
}
err_cunt=err_cun_num;
op=50; //op 50 ==> password is true and chane time date
}else{
lcd_clear ();
lcd_putsf ("INVALID PASSWORD");
delay_ms (1000);
err_cunt--;
}
reset_var ();
}

```



این تابع دارای یک پارامتر ورودی به نام open می باشد در صورتی که مقدار آن ۱ باشد بعد از چک کردن رمز عبور عملیات مربوط به تحریر ایتوکوپلر انجام شده و درب باز می شود. اصول کار تابع به این صورت می باشد که رمز عبور وارد شده با رمز عبور ذخیره شده مقایسه می شود و در صورتی که یکی از آنها مطابقت نداشته باشد یک واحد به متغیر C اضافه می شود، بعد از اتمام این مرحله محتوای C تست می شود و اگر مخالف مقدار پیش فرض ۱ بود متوجه خواهیم شد که یکی از ارقام رمز عبور اشتباه است بنابراین پیغام INVALID صادر می شود و یک واحد از err\_cunt کم می شود اما در صورتی که مقدار C تغییر نکند یعنی رمز ورودی صحیح بود و درب با توجه به شرط گفته شده باز می شود و متغیر err\_cunt مجدداً با مقدار پیش فرض برای کاربر بعدی بازنشانی می شود همچنین op=۵۰ می شود تا اگر این تابع برای تغییر ساعت و تاریخ فراخوانی شده بود مجوز تغییر داده شود. و در آخر هم متغیرهای مهم بازنشانی می شوند.

## تابع تغییر رمز عبور

```
void change_pass (){\n    unsigned char i,c=1;\n    op=1; //get old password\n    get_pass ();\n    for (i=0; i<len_pass; i++){ \n        old_pass[i]=pass_in[i];\n    }\n    op=2; //get new password\n    get_pass ();\n\n    for (i=0; i<len_pass; i++){ \n        if (old_pass[i]!=pass[i]){ \n            c++; \n            break;\n        }\n    }\n\n    key = masage ();\n    if (key==11){// key==11 save is OK Go to cheking...\n\n    if (c==1){\n        for (i=0; i<len_pass; i++){ \n            pass[i]=pass_in[i];\n        }\n        lcd_clear ();\n        lcd_putsf ("Successfully !");\n        delay_ms (3000);\n        err_cunt=err_cun_num; //cunter for wait invalid password\n    }else{\n        lcd_clear ();\n        lcd_putsf ("ERROR! Wait... ");\n        delay_ms (5000);\n        err_cunt--;\n    }\n\n    }\n    reset_var (); key=0;\n}
```

در تابع ابتدا op=1 می شود و رمز قبلی را از کاربر دریافت کرده سپس مجدداً op=2 می شود و رمز جدید را از کاربر دریافت می کند، سپس رمز قبلی را با رمز ذخیره شده مقایسه می کند و پیغام ذخیره سازی را نمایش می دهد در صورتی که # فشرده شود نتیجه مقایسه رمز را بررسی می کند اگر C برابر ۱ باشد رمز صحیح بوده و در نتیجه رمز جدید جایگزین رمز ذخیره شده خواهد و پیغام Successfully نمایش داده می شود. اما در صورتی که C مخالف ۱ باشد یعنی رمز قدیمی اشتباه است و پیغام ERROR نمایش داده می شود و یک واحد از err\_cunt کم می شود. و در آخر هم متغیرهای مهم بازنشانی شده و سیستم وارد منوی اصلی می شود.

```
void change_time (){
unsigned char digit[2],newtime[6]; //0h,1m,2s 3y,4mo,5d
unsigned char x,y=0,j,locate[6]={5,8,11,5,8,13};
get_pass ();
check_pass (0);
if (op==50){ //op 50 ==> password is true and chane time is alow
sprintf (buffer,"Time --:--:--\nDate --/--/20--");
lcd_clear ();
lcd_puts (buffer);
_lcd_write_data (0X0F);

for (x=0; x<=5; x++){ //get 6 number
digit[0]=digit[1]=0;
if (x>2) y=1; // Go to next row
for (j=0; j<=1; j++){ //get 2 digit from keypad
lcd_gotoxy (locate[x],y);
digit[j]=keypad (wait_scan_keypad);
if (j==0) digit[0]*=10; //Digit format xx => digit[0]x
newtime[x]=digit[0]+digit[1]; //چسپاندن دو عدد به هم و تشکیل عدد دورقمی
if (digit[j]!=100){ //chek for time out keypad
sprintf (buffer,"%d",newtime[x]);
lcd_puts (buffer);
}else goto endTime; //esc
}
}
key=masage ();
if (key==11){
gregorian_week_day (newtime[3],newtime[4], (newtime[5]+2000)); //calculate week day
rtc_set_time (newtime[0],newtime[1],newtime[2]); //hour, min, sec //0h,1m,2s
rtc_set_date (week,newtime[3],newtime[4],newtime[5]); //week_day, day ,month, year //3y,4mo,5d
}else if (key==10) goto endTime;
}
endTime:
key=op=0;
}
```

این تابع نسبت به تابع های دیگر کمی پیچیده تر است و نیاز به تمرکز بیشتری دارد. کلا وظیفه این تابع دریافت اعداد دورقمی مربوط به ساعت، ماه و سال... و ست کردن آنها در DS1307 می باشد. متغیر digit یک آرایه دو عضوی است که عدد دو رقمی ورودی در آن قرار می گیرد. newtime آرایه ۶ عضوی برای نگهداری اطلاعات ساعت، دقیقه، ثانیه، روز، ماه و سال می باشد. locat نیز مکان های نوشتن اطلاعات ورودی روی LCD می باشد.

ابتدا check\_pass فراخوانی می شود تا هر کسی که رمز عبور را دارد بتواند ساعت تنظیم کند، بنابراین اگر رمز عبور صحیح باشد op=50 و مجوز تغییرات صادر می شود، ابتدا کاراکتر هایی روی نمایشگر چاپ می شود سپس در حلقه داخلی ۶ بار اطلاعات دو رقمی دریافت کرده و در نهایت سوال ذخیره سازی مطرح می شود و با تأیید آن اطلاعات جدید جایگزین خواهد شد. اما در حلقه for چه اتفاقاتی رخ میدهد؟ ابتدا هر دو عضو digit صفر می شود، و بعد از آن شرطی برای  $x < 2$  گذاشته شده است که در صورت برقرار شدن این شرط مکان به سطر دوم انتقال می یابد. حال وارد حلقه for داخلی می شویم و مکان نما به سطر ی که y دارد و مقداری که x دارد از آرایه locate[x] محل مکان نما در مختصات x انتخاب شده و به آنجا منتقل می شود، بعد از آن کیبورد فراخوانی می شود و مقدار کلید فشار داده شده در آرایه digit قرار می گیرد حال اگر  $j=0$  بود اولین رقم در ۱۰ ضرب شده و در عضو اولی آریه قرار می گیرد و در صورتی که  $j=1$  بود عمل ضرب انجام نمی گیرد و مستقیماً در عضو دومی آرایه قرار خواهد گرفت بنابراین به این صورت ما یک عدد صحیح دور رقمی دریافت کردیم، حال اگر time out از سوی کیبورد نداشته باشیم عدد دریافت شده روی نمایشگر چاپ می شود. اما اگر time out داشته باشیم قسمت دوم این شرط یعنی else goto endTime اجرا شده و تابع به کار خود پایان می دهد بنابراین هیچ عمل بروزرسانی در DS1307 انجام نخواهد گرفت. با فرض اینکه اطلاعات صحیح وارد شده و time out نیز نداریم در ادامه برنامه سوال ذخیره سازی مطرح شده و با تأیید آن مقادیر جدید در حافظه RAM DS1307 جایگزین شده و به اصطلاح آپدیت می شود. نکته آخر اینکه اطلاعات جدید سال، ماه و روز را وارد تابع gregorian\_week\_day

```
void main (void){
unsigned char a,b;
lcd=1;
lcd_init (16);

PORTC=0x00;
DDRC. 2=1;
DDRC. 3=1;
PORTD=0x00;
DDRD. 7=1;

err_cunt=err_cun_num;
new_pass ();

i2c_init ();
rtc_init (0,0,0);

ACSR=0x80;
SFIO=0x00;
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x87;
```

تابع main که تمامی توابع نوشته شده در اینجا فراخوانی و کنترل می شوند، در ابتدای این تابع تا قبل از شروع حلقه بینهایت پیکره بندی ها و تنظیمات اولیه صورت می گیرد. که ابتدا LED نمایشگر روشن می شود و بعد از آن نمایشگر به صورت ستون های ۱۶ تایی پیکره بندی می شود. در ادامه پین هایی از پورت ها به صورت خروجی تعریف می شود و بعد از آن نیز شمارنده فرصت های باقی مانده برای وارد کردن رمز با مقدار پیش فرض err\_cun\_num بارگذاری شده و تابع new\_pass برای چک کردن اینکه آیا رمز اولیه ساخته شده یا نه فراخوانی می شود. در قسمت بعد پروتکل I2C و خود IC DS1307 جهت راه اندازی پیکره بندی اولیه می شوند. و در آخر هم مبدل آنالوگ به دیجیتال با تنظیمات اولیه آماده راه اندازی و نمونه برداری می شود.

از اینجا به بعد CPU وارد حلقه بینهایت شده و توابع را به صورت مداوم فراخوانی می کند.

```
while (1){
lcd_back_light ();
get_rtc ();
key=0xFF;op=0; //restr this var
if (err_cunt==0){ //wait for error
lcd_clear ();
lcd_gotoxy (3,0);
lcd_putsf ("ERROR... ");
lcd_gotoxy (0,1);
lcd_putsf ("You Don't Input");
alarm=1; delay_ms (500); alarm=0; delay_ms (300);
alarm=1; delay_ms (500); alarm=0; delay_ms (300);
alarm=1; delay_ms (2000); alarm=0;

for (a=0; a<15; a++){
for (b=0; b<60; b++){ // 1 min
lcd_gotoxy (15,0);
lcd_putsf ("|");
delay_ms (300);
lcd_gotoxy (15,0);
lcd_putsf ("/");
delay_ms (300);
```



```

}
}
err_cunt=err_cun_num;
}
display ();
key = keypad (0);

if (key==1) change_time ();
if (key==10) change_pass ();
if (key==11){
get_pass ();
if (key!=100) check_pass (1); //if: no time out then open=1
}
}
}

```

در حلقه بینهایت تابع lcd\_back\_light برای کنترل نور نمایشگر به صورت مداوم فراخوانی می شود. و بعد هم تابع get\_rtc برای بروز رسانی کردن متغیرهای ساعت و تاریخ فراخوانی می شود. متغیرهای key و op را ریست می کنیم، متغیر مهم ما یعنی err\_cunt ابتدا چک می شود و در صورتی که برابر با صفر شده بود سیستم به مدت ۱۵ دقیقه قفل شده و هیچ عملی انجام نخواهد داد، این قسمت برای امنیت بیشتر قرار داده شده تا که اگر کسی رمز عبور را نمی داند و بخواهد با امتحان کردن تعدادی رمز عبور تصادفی به رمز عبور اصلی دست یابد، نتواند بیشتر از ۴ مورد را امتحان کند و پس از ۴ مرتبه سیستم با ریتم خاصی بوق آیفون شما را به صدا در خواهد آورد و قفل خواهد کرد که البته بعد از سپری شدن این زمان err\_cunt دوباره با مقدار پیش فرض بارگیری خواهد شد.

بعد از این شرط تابع display فراخوانی شده تا نمایشگر را تازه کند. بعد از آن تابع keypad (0) فراخوانی می شود که در اینجا چون صفر ورودی تابع است تنها به مدت یک ثانیه کیبورد اسکن شده و بر می گردد حالا مقدار key را مورد مقایسه قرار می دهد و هر کدام یک از شرط ها که برقرار شود توابع خاص و مربوط به آنها فراخوانی می شوند.

#### توابع موجود در فایل هدر

متغیرهای سراسری این هدر

```

typedef unsigned char byte;
unsigned char g_days_in_month[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
unsigned char j_days_in_month[] = {31, 31, 31, 31, 31, 31, 30, 30, 30, 30, 30, 29};
char buffer[32], week;
unsigned char hour=0, minute=0, second=0;
unsigned char year=0, month=0, day=0;
long int y,m,d;
long int jy,jm,jd;
long int gy,gm,gd;

```

byte همان unsigned char می باشد که توسط دستور typedef تغییر نام داده شده است. g\_days\_in\_month و j\_days\_in\_month آرایه هایی هستند که تعداد روزهای ماه های شمسی و میلادی در آن قرار گرفته اند. سایر متغیرها هم مربوط به ساعت و تاریخ می باشند، آنهایی که با حرف ز شروع می شوند شمسی اند و آنهایی که با حرف g شروع می شوند میلادی اند. برخی از متغیرهای سراسری که می خواهیم به صورت مشترک در اینجا هم از آنها استفاده نمائیم باید مجدداً به صورت سراسری تعریفشان کنیم.

#### تابع تقسیم ساده

```

long int div (long int a, long int b){
return (int) (a / b);
}

```

این تابع دو عدد a و b را گرفته و آنها را برهم تقسیم می کند، سپس نتیجه را برگشت می دهد.

```
void gregorian_to_jalali (long int g_y, long int g_m, long int g_d){
long int g_day_no,j_day_no,j_np;
unsigned int i;
gy = g_y-1600;
gm = g_m-1;
gd = g_d-1;
g_day_no = 365*gy+div (gy+3,4)-div (gy+99,100)+div (gy+399,400);

for (i=0; i < gm; i++)
    g_day_no += g_days_in_month[i];
if (gm>1 && ( (gy%4==0 && gy%100!=0) || (gy%400==0)))
    /* leap and after Feb */
    g_day_no++;
g_day_no += gd;
j_day_no = g_day_no-79;
j_np = div (j_day_no, 12053); /* 12053 = 365*33 + 32/4 */
j_day_no = j_day_no % 12053;
jy = 979+33*j_np+4*div (j_day_no,1461); /* 1461 = 365*4 + 4/4 */
j_day_no %= 1461;

if (j_day_no >= 366) {
    jy += div (j_day_no-1, 365);
    j_day_no = (j_day_no-1)%365;
}

for (i = 0; i < 11 && j_day_no >= j_days_in_month[i]; i++)
    j_day_no -= j_days_in_month[i];
jm = i+1;
jd = j_day_no+1;
}
```

این تابع دارای سه پارامتر ورودی شامل سال، ماه و روز است که برحسب میلادی می باشد. با توجه به قوانین تبدیل سال میلادی به سال شمسی این تابع یک سری ضرب و تقسیم ها را انجام می دهد تا این عمل تبدیل صورت گیرد. تابع تبدیل خیلی دقیق می باشد حتی سال های کبیسه را نیز محاسبه می کند. این تابع را از سورس سیستم مدیریت محتوای نیوک Nuke که به زبان PHP نوشته شده بود استفاده کرده ام و آن را به زبان C میکروکنترلر تبدیل کردم. کلا از لحاظ دستوری چیز خاصی ندارد، همانطور که مشاهده می کنید از ضرب و تقسیم استفاده شده، تنها سوالی که پیش می آید این است که این اعداد ثابت از کجا آمده اند؟ همانطور که اشاره شده بر طبق قوانین ریاضی تبدیل سال ها به یکدیگر است.

```
void gregorian_week_day (long int g_y, long int g_m, long int g_d){
long int gy,gm,gd,g_day_no;
unsigned int i;
gy = g_y-1600;
gm = g_m-1;
gd = g_d-1;
g_day_no = 365*gy+div (gy+3,4)-div (gy+99,100)+div (gy+399,400);
for (i=0; i < gm; i++)
    g_day_no += g_days_in_month[i];
if (gm>1 && ( (gy%4==0 && gy%100!=0) || (gy%400==0)))
    /* leap and after Feb */
    g_day_no++;
g_day_no += gd;
week = (g_day_no + 5) % 7 + 1;
}
```

این تابع نیز از سورس همان سیستم نیوک گرفته شده است، اصول کار آن به این صورت است که سال و ماه و روز میلادی را دریافت کرده و پس از انجام یک سری عملیات ریاضی روز هفته را مشخص می کند. مثلاً شما یک تاریخی مربوط به ۱۰۰ سال پیش یا آینده را به این تابع بدهید به شما خواهد گفت که آن روز چند شنبه است.

## تابع تعریف کاراکتر خاص

```
void define_char (byte flash *pc,byte char_code){
byte i,a;
a= (char_code<<3) | 0x40;
for (i=0; i<8; i++) lcd_write_byte (a++,*pc++);
}
```

این تابع که یکی از توابع معروف Help نرم افزار کدویژن است، برای شما کاراکترهای خاص را که روی حافظه فلش به صورت آرایه ۸ عضوی تعریف کرده اید را به کمک دستور lcd\_write\_byte در LCD برای شما به نقش می کشد. شما می توانید با نرم افزار هایی که برای اینکار طراحی شده اند اشکال خاص خود را به صورت پیکسلی رسم کرده و سپس کدهای ۸ بایتی آن را در حافظه فلش میکرو تعریف کرده و با استفاده از توابع زیر آنها را روی LCD رسم کنید. که در اینجا ما کلمه فارسی "شنبه" و "جمعه" را تعریف کرده ایم.

## تابع کاراکتر فرصت های باقی مانده

```
void cunter_err_pass (unsigned char err_cunt){
if (err_cunt==4){
define_char (char_4err,6);
lcd_gotoxy (15,0);
lcd_putchar (6);
}

if (err_cunt==3){
define_char (char_3err,6);
lcd_gotoxy (15,0);
lcd_putchar (6);
}

if (err_cunt==2){
define_char (char_2err,6);
lcd_gotoxy (15,0);
lcd_putchar (6);
}

if (err_cunt==1){
define_char (char_1err,6);
lcd_gotoxy (15,0);
lcd_putchar (6);
}
}
```

این تابع هم به این صورت عمل می کند که ابتدا متغیر err\_cunt را مورد مقایسه قرار می دهد و با توجه به مقدار آن یکی از شرط ها برقرار شده و آن کاراکتر خاص را در مکان مربوطه رسم می کند. ساده تر بگوییم چوب خط های باقی مانده کاربر را می کشد.

## تابع نمایش فارسی روز هفته

```
void week_name_FA(){
if(week!=5){
define_char(char_sh,0);
define_char(char_nb,1);
define_char(char_h,2);
lcd_gotoxy(12,1);
lcd_putchar(2);
}
```



```
lcd_putchar(1);  
lcd_putchar(0);  
lcd_gotoxy(15,1);  
}else{  
    define_char(char_j,4);  
    define_char(char_m,5);  
    define_char(char_h,2);  
    lcd_gotoxy(12,1);  
    lcd_putchar(2);  
    lcd_putchar(5);  
    lcd_putchar(4);  
}  
}
```

در اینجا نیز عدد روز هفته که در week قرار دارد را تست می کند که آیا امروز جمعه است یا روزهای دیگر. همانطور که می دانید همه روزهای هفته ما با کلمه شنبه شروع می شود به جزء روز جمعه، پس این تابع روز را شناسایی کرده و یکی از کلمات شنبه یا جمعه را به صورت فارسی در مکان های مشخص شده نمایشگر رسم می کند.

## تابع انتخاب روز هفته و نمایش آن

```
void week_name (){  
    week_name_FA ();  
    switch (week){  
    case 1:  
        lcd_putchar ('2'); //"دوشنبه"  
        break;  
    case 2:  
        lcd_putchar ('3'); //"سه شنبه"  
        break;  
    case 3:  
        lcd_putchar ('4'); //"چهارشنبه"  
        break;  
    case 4:  
        lcd_putchar ('5'); //"پنج شنبه"  
        break;  
    case 5:  
        //"جمعه"  
        break;  
    case 6:  
        lcd_putchar (' '); //"شنبه"  
        break;  
    case 7:  
        lcd_putchar ('1'); //"یکشنبه"  
        break;  
    }  
}
```

بعد از نوشتن روز شنبه یا جمعه توسط تابع week\_name\_FA حال باید عدد روز قبل از کلمه شنبه نوشته شود که این کار هم توسط این تابع انجام می گیرد. که البته برای روز شنبه کاراکتر خالی ارسال می کند و برای بقیه روزها عدد آن روز را چاپ می کند.

## تابع دریافت تاریخ و ساعت

```
void get_rtc (){  
    rtc_get_time (&hour,&minute,&second);  
    rtc_get_date (&week,&day,&month,&year);  
    y=year+2000; m=month; d=day;  
    gregorian_to_jalali (y,m,d);  
}
```

```
gregorian_week_day (y,m,d);
```

// در صورتی که باتری بک آپ وجود نداشته باشد یا تغذیه ساعت قطع شود، مقدار دهی اولیه خواهد شد

```
if (jy<1390)
```

```
{
```

```
//if not battery backup ==> set&reset RTC
```

```
rtc_set_time (12,0,0); /* set time 12:00:00 */
```

```
rtc_set_date (2,29,11,11); /* set date Tuesday 29/11/2011 */
```

```
}
```

```
}
```

وظیفه این تابع نیز دریافت اطلاعات از DS1307 می باشد، که ابتدا اطلاعات را به صورت میلادی دریافت کرده سپس با افزودن عدد ۲۰۰۰ به متغیر سال آنها را برای تبدیل به شمسی ارسال می نماید و بعد هم روز هفته را محاسبه می کند. و در آخر هم شرطی برای سال شمسی گذاشته ایم که در صورتی که از ۱۳۹۰ کوچکتر باشد این احتمال دارد که IC هنوز تنظیم نشده است و یا تغذیه آن و باتری پشتیبان آن خراب شده است بنابراین یک ساعت و تاریخ پیش فرض در آن ست می شود تا شروع به کار کند. نکته: در صورتی که برای اولین بار DS1307 را به مدار وصل می کنید برای راه اندازی و شروع بکار آن ابتدا باید حتما، تاریخ و ساعت را تنظیم کنیم و در آن ست کنیم تا شروع بکار کند. سال پیش فرض اغلب IC ها به شمسی معادل ۱۳۷۸ می باشد.

## LabVIEW 2010 SP1 Professional Full AddOns



نرم افزار Lab View که مخفف عبارت Laboratory Virtual Instrument Engineering Workbench می باشد، یک زبان برنامه نویسی گرافیکی می باشد که به صورت گسترده ای برای کاربرد های مختلفی در صنایع، تحصیلات، آموزش و تحقیقات آزمایشگاهی به عنوان یک مدل استاندارد برای جمع آوری و پردازش داده ها و همچنین وسیله ای جهت کنترل و شبیه سازی ابزارهای مجازی در آمده است. این برنامه یک نرم افزار قدرتمند و قابل انعطاف جهت تجزیه و تحلیل سیستم های اندازه گیری است.

عملکرد نرم افزار Lab View کاملاً از طبیعت ترتیبی و زنجیره ای موجود زبان های برنامه نویسی متنی متداول و مرسوم مجزاست و یک محیط گرافیکی را برای کاربر فراهم ساخته است. در این راه از تمامی ابزارهای لازم جهت جمع آوری، پردازش و تحلیل داده ها و نمایش نتایج استفاده می شود. به کمک این زبان برنامه نویسی گرافیکی که با "G" نشان داده می شود، در برنامه ی نوشته شده، از یک نمودار بلوکی استفاده می شود و سپس این نمودار به کدهای ماشین تبدیل می گردد. این نرم افزار برای موارد بی شماری از کاربردهای علمی و مهندسی، ایده آل و عملی است و به شما کمک می کند تا مسائل و مشکلات موجود در برنامه نویسی را در مدت زمان کوتاهی حل کنید.

گسترده گی Lab View در زمینه آزمایشگاه ها دارای ابعاد مختلفی است. به عنوان مثال در صنایع گوناگون در مواردی که باید اندازه گیری هایی از قبیل دما انجام گیرد، می توان از این نرم افزار استفاده نمود. این دما ممکن است دمای یک کوره، یک سیستم سرد کننده مانند یخچال، یک محیط گلخانه ای، یک اتاق و یا یک دیگ سوپ باشد. علاوه بر دما می توان موارد دیگری نظیر فشار، نیرو، جابجایی، کشش، pH و... را نام برد. از رایانه های شخصی به همراه این نرم افزار می توان ابزار های اندازه گیری حقیقی را در محل های گوناگون به صورت مجازی شبیه سازی کرد. در این مجموعه می توانید مجموعه کاملی از Add-On و Modules و Toolkits های موجود برای این نرم افزار را دریافت و استفاده کنید.

این محصول در قالب **هفت DVD** ارائه گشته است.

توضیحات تکمیلی: <http://eshop.eca.ir/link/805.php>

## برد مبدل SMD به DIP – نوع ۶۴ پایه



قابل نصب آی سی های ۱۶-۲۴-۳۲-۴۰-۴۴-۴۸-۵۶-۶۴ پایه

مورد استفاده در آی سی های با پکیج های QFP، TQFP، QFN، MLP، MLF

قلع اندود شده و بدون نیاز به قلع قابل نصب بر روی بردبرد

توضیحات تکمیلی: <http://eshop.eca.ir/link/617.php>

# AT91SAM7X256 Header Board



- امکانات اولیه جهت راه اندازی میکروکنترلر AT91SAM7X256
- بدون نیاز به پروگرامر
- دارای خروجی تمام پایه های ورودی خروجی
- امکان نصب مستقیم بر روی بردبرد
- دارای خروجی ولتاژهای ۳.۳ و ۵ ولت
- امکان استفاده از پروگرامر SAM-BA
- دارای پورت JTAG
- امکان کنترل پایه های TST, JTAGSEL و ERASE با استفاده از دیپ سوئیچ
- دارای ۱۲ ماه گارانتی تعویض

توضیحات تکمیلی : <http://eshop.eca.ir/link/601.php>

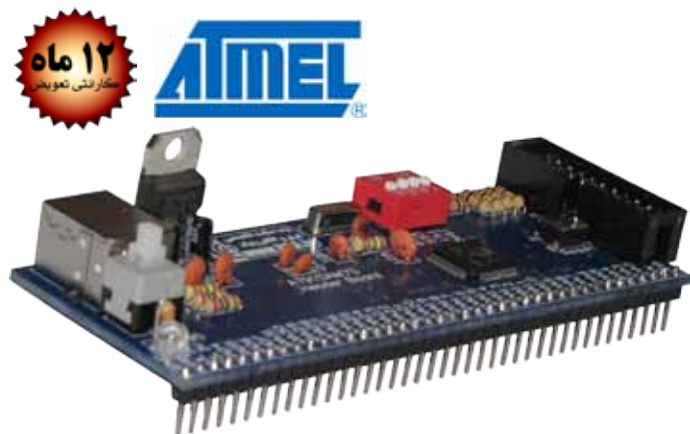
# AT91SAM7S256 Header Board



- امکانات اولیه جهت راه اندازی میکروکنترلر AT91SAM7S256
- بدون نیاز به پروگرامر
- دارای خروجی تمام پایه های ورودی خروجی
- امکان نصب مستقیم بر روی بردبرد
- دارای خروجی ولتاژهای ۳.۳ و ۵ ولت
- امکان استفاده از پروگرامر SAM-BA
- دارای پورت JTAG
- امکان کنترل پایه های TST, JTAGSEL و ERASE با استفاده از دیپ سوئیچ
- دارای ۱۲ ماه گارانتی تعویض

توضیحات تکمیلی : <http://eshop.eca.ir/link/641.php>

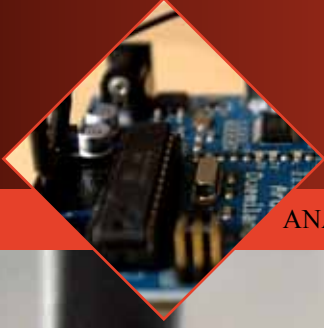
# AT91SAM7S64 Header Board



- امکانات اولیه جهت راه اندازی میکروکنترلر AT91SAM7S64
- بدون نیاز به پروگرامر
- دارای خروجی تمام پایه های ورودی خروجی
- امکان نصب مستقیم بر روی بردبرد
- دارای خروجی ولتاژهای ۳.۳ و ۵ ولت
- امکان استفاده از پروگرامر SAM-BA
- دارای پورت JTAG
- امکان کنترل پایه های TST, JTAGSEL و ERASE با استفاده از دیپ سوئیچ
- دارای ۱۲ ماه گارانتی تعویض

توضیحات تکمیلی : <http://eshop.eca.ir/link/519.php>





# Projects

ANALOG AND DIGITAL Projects

## پروژه های عملی



آشنایی با GSM Modem و نحوه ارتباط آن با میکروکنترلر

### GSM Modem چیست ؟

در واقع GSM Modem دستگاهی در اختیار ما قرار می دهد. به وسیله GSM اینترنت همراه (GPRS) استفاده کرد و حتی شرکت های زیادی در دنیا انواع GSM Modem برای اتصال مستقیم به پورت USB کامپیوتر طراحی از جمله موارد استفاده از GSM Modem استفاده از آن در تماس گرفته و یا پیامک ارسال می کند و حتی می توان بوسیله پیامک دزدگیر را خاموش و روشن کرد و یا آسانسورهایی که در صورت خرابی به صورت خودکار با شرکت سرویس دهنده آسانسور تماس می گیرند و آنها را مطلع می کنند و...

### انتخاب یک GSM Modem مناسب

بسیاری از GSM Modem ها بوسیله پورت سریال دستورات مورد نظر را دریافت می کنند و تقریباً تمام میکروکنترلرهای دنیا پورت سریال (UART) را پشتیبانی می کنند. در اینجا GSM Modem انتخابی ما برای آموزش ماژول SIM900 است که علاوه بر فراوانی در بازار ایران، قیمت مناسب و کارایی بالایی دارد. ابعاد این مودم بسیار کوچک است که استفاده از آن را در دستگاه ها به مراتب ساده تر می کند. در حال حاضر برای سیستم های میکروکنترلری بهترین گزینه استفاده از SIM900 است.

### آشنایی با سخت افزار SIM900

#### الف) منبع تغذیه برای SIM900

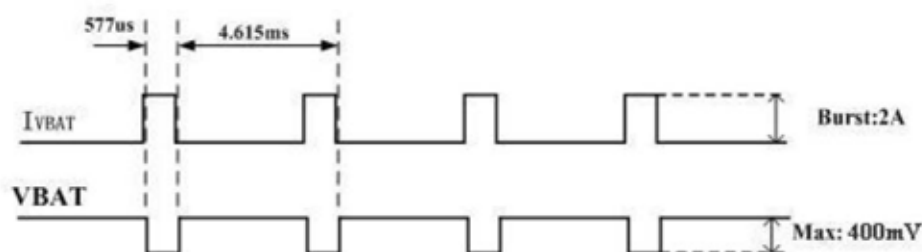
این ماژول برای کار نیاز به یک منبع با ولتاژ بین ۳٫۴V تا ۴٫۵V دارد. همچنین مقدار متوسط جریان مصرفی آن در حالت بیکاری ۲۲mA و در حالت کار (مانند ارسال پیامک) حداکثر ۴۰۰mA است.

# آشنایی با GSM Modem و نحوه ارتباط آن با میکروکنترلر



نکته مهمی که در اینجا وجود دارد پیک بالای جریان لحظه ای این ماژول است. در شکل زیر شکل موج جریان این ماژول را مشاهده می کنید. مطابق این شکل موج ماژول SIM900 به صورت لحظه ای جریانی تا حدود ۲A مصرف می کند که در پیک جریان ولتاژ تغذیه نباید بیش از ۴۰۰mV افت کند، در غیر این صورت ماژول ریست می شود.

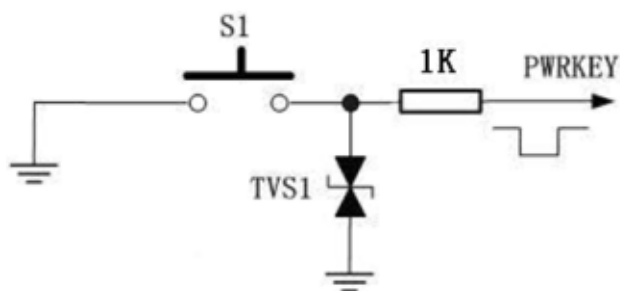
برای تامین این پیک جریان استفاده از دو خازن تانتالیوم موازی با ظرفیت های ۱۰۰nF و ۱۰۰uF در مسیر تغذیه ماژول الزامی است. همچنین برای مدار رگولاتور می توان از مدار روبرو استفاده کرد



شکل موج جریان ماژول SIM900

## ب) روشن کردن SIM900

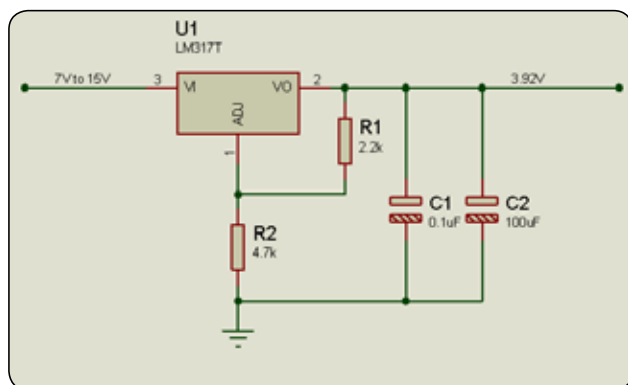
پس از اعمال ولتاژ به این ماژول برای روشن کردن آن بایستی پایه PWRKEY را برای مدت حداقل ۱ ثانیه صفر کنیم و سپس دوباره آن را یک کنیم. برای خاموش کردن ماژول نیز باید همینطور عمل کنیم.



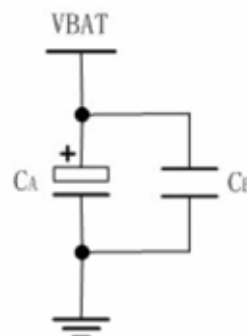
نحوه روشن کردن SMI900

## ج) اتصال SIM900 به میکروکنترلر

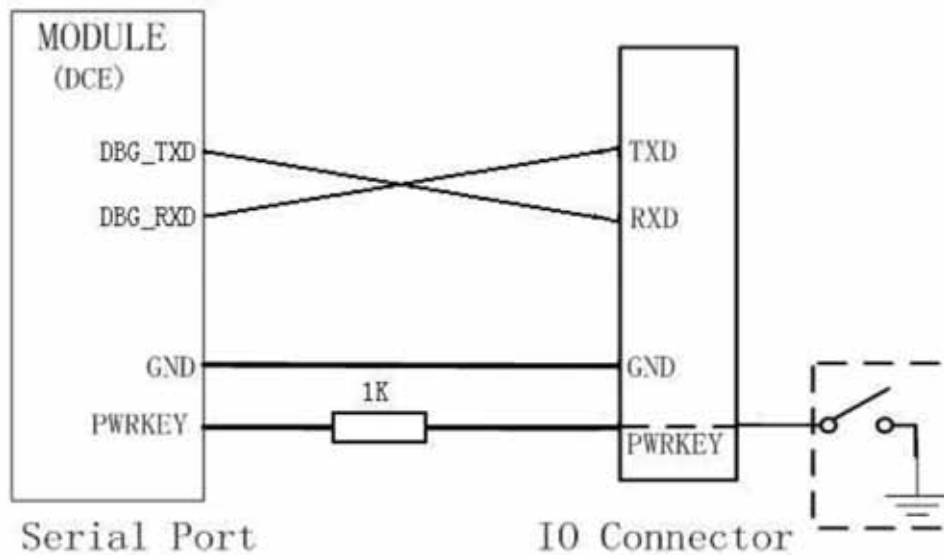
همانطور که گفته شد ماژول SIM900 از طریق پورت سریال (UART) ارتباط برقرار می کند. در شکل زیر حداقل سیم بندی مورد نیاز برای ارتباط این ماژول با میکروکنترلر را مشاهده می کنید.



مدار رگولاتور پیشنهادی



خازن های موازی جهت تامین پیک جریان



حداقل سیم بندی مورد نیاز برای ارتباط ماژول با میکروکنترلر

همانطور که می دانید پروتکل ارتباط سریال UART دارای چندین مشخصه است:

- تعداد بیت های هر فریم که در اینجا ۸bit است.
- تعداد Stop bit ها که در اینجا ۱ است.
- نوع Parity که در اینجا غیر فعال است.
- سرعت ارتباط که می تواند هر یک از مقادیر ۱۲۰۰bps ، ۲۴۰۰bps ، ۴۸۰۰bps ، ۹۶۰۰bps ، ۱۹۲۰۰bps ، ۳۸۴۰۰bps ، ۵۷۶۰۰bps ، ۱۱۵۲۰۰bps باشد. ماژول سرعت ارتباط را به طور اتوماتیک تشخیص می دهد.

همانطور که گفته شد ماژول SIM900 این قابلیت را دارد که سرعت ارتباط را به طور خودکار تشخیص دهد، برای این کار ابتدا باید در ابتدا یک کاراکتر 'A' به ماژول ارسال کنیم.

## دستورات ماژول SIM900

ماژول SIM900 به صورت کاراکتری دستورات را دریافت می کند و به همان صورت هم به آنها پاسخ می دهد. در انتهای هر دستور باید دو کاراکتر CR و LF که معادل عددی آنها به ترتیب ۱۰ و ۱۳ است ارسال شوند. این دو کاراکتر در جدول ASCII که پیش از این در رابطه با آن بحث شده است تعریف شده اند. به این شکل ارتباطی بین یک ماژول و CPU پروتکل ATcommand می گویند. ماژول پس از دریافت هر دستور پاسخ آن را ارسال می کند.

- دستور AT

این دستور برای تست سلامت ارتباط با ماژول می باشد. ماژول در پاسخ به این دستور OK پاسخ می دهد.

AT  
OK

- دستور ATI

ماژول در پاسخ به این دستور مدل و ورژن خود را ارسال می کند.

ATI  
SIM900 R11.0  
OK

- دستور AT+CSQ

ماژول در پاسخ به این دستور دو عدد را باز می گرداند. عدد اول قدرت سیگنال آنتن را با عددی بین ۰ تا ۳۲ برمی گرداند و عدد دوم نشان دهنده جزئیاتی است که در اینجا برای ما اهمیتی ندارد.

AT+CSQ  
+CSQ: 14,0  
OK

- دستور AT+CBC

ماژول در پاسخ به این دستور سه عدد را برمی گرداند. عدد اول نشان دهنده در حال شارژ بودن باتری است. عدد دوم میزان شارژ باتری را به صورت عددی بین ۰ تا ۱۰۰ نشان می دهد و عدد سوم ولتاژ باتری را بر حسب میلی ولت نشان می دهد.

AT+CBC

# آشنایی با GSM Modem و نحوه ارتباط آن با میکروکنترلر

+CBC: 0,100,4338

OK

- دستور AT+CMGF (Select sms message format)

این دستور برای انتخاب فرمت نوشتن و خواندن پیامک در SIM900 است. این ماژول با دو فرمت Text و PDU می‌تواند پیامک‌ها را سرویس دهد. هر چند فرمت PDU کاملتر است ولی بسیار پیچیده بوده و در اینجا به دلیل سادگی فقط با فرمت Text کار می‌کنیم. برای انتخاب فرمت Text کافی است دستور AT+CMGF=1 را به ماژول بفرستید. ماژول در پاسخ OK بر می‌گردد.

AT+CMGF=1

OK

- دستور AT+CMGS (Send SMS)

این دستور در فرمت Text و PDU رفتار متفاوتی دارد که در اینجا فرمت Text آن را تشریح می‌کنیم. ابتدا باید شمار گیرنده را در انتهای دستور وارد کنیم :

AT+CMGS="09191368415"

در پاسخ این دستور ماژول کاراکتر '<' را بر می‌گرداند که به این معنی است که ماژول آماده دریافت متن پیامک است :

AT+CMGS="09191368415"

> Hello SIM900. i am student.

پس از وارد کردن متن پیام بایستی کاراکتر 0x1a را ارسال کنیم. ماژول با دریافت این کد شروع به ارسال پیامک می‌کند و در صورت عدم موفقیت کلمه ERROR را بر می‌گرداند و در صورت ارسال موفق OK پاسخ می‌دهد :

AT+CMGS="09121234567"

> Hello SIM900. i am student.

+CMGS: 145

OK

- دستور AT+CMGL (List SMS messages)

این دستور برای خواندن پیامک‌های موجود در سیم کارت است و به چند شکل می‌توان از آن استفاده کرد :  
ماژول در پاسخ به این دستور متن تمام پیامک‌ها را با زمان دریافت و سایر مشخصاتشان بر می‌گرداند :

AT+CMGL="REC UNREAD"

+CMGL: 2,"REC UNREAD","+989191368415","E57A@","11/09/28,19:30:01+14"

Hello

This is mostafa.

OK

- دستور AT+CMGD (Delete SMS messages)

این دستور برای حذف پیامک‌های موجود در سیم کارت است و به دنبال آن دو عدد باید ارسال کنیم.  
عدد اول را index می‌نامیم و عدد دوم را delflag که این دو عدد عملکرد دستور را کنترل می‌کنند. در جدول زیر نحوه عملکرد دستور را بر اساس مقادیر مختلف delflag مشاهده می‌کنید.

## SIM900a - GSM/GPRS ماژول

Featuring an industry-standard interface, the SIM900 delivers GSM/GPRS 850/900/1800/1900MHz performance for voice, SMS, Data, and Fax in a small form factor and with low power consumption. With a tiny configuration of 24mm x 24mm x 3 mm, SIM900 can fit almost all the space requirements in your M2M applications, especially for slim and compact demands of design.

توضیحات تکمیلی :

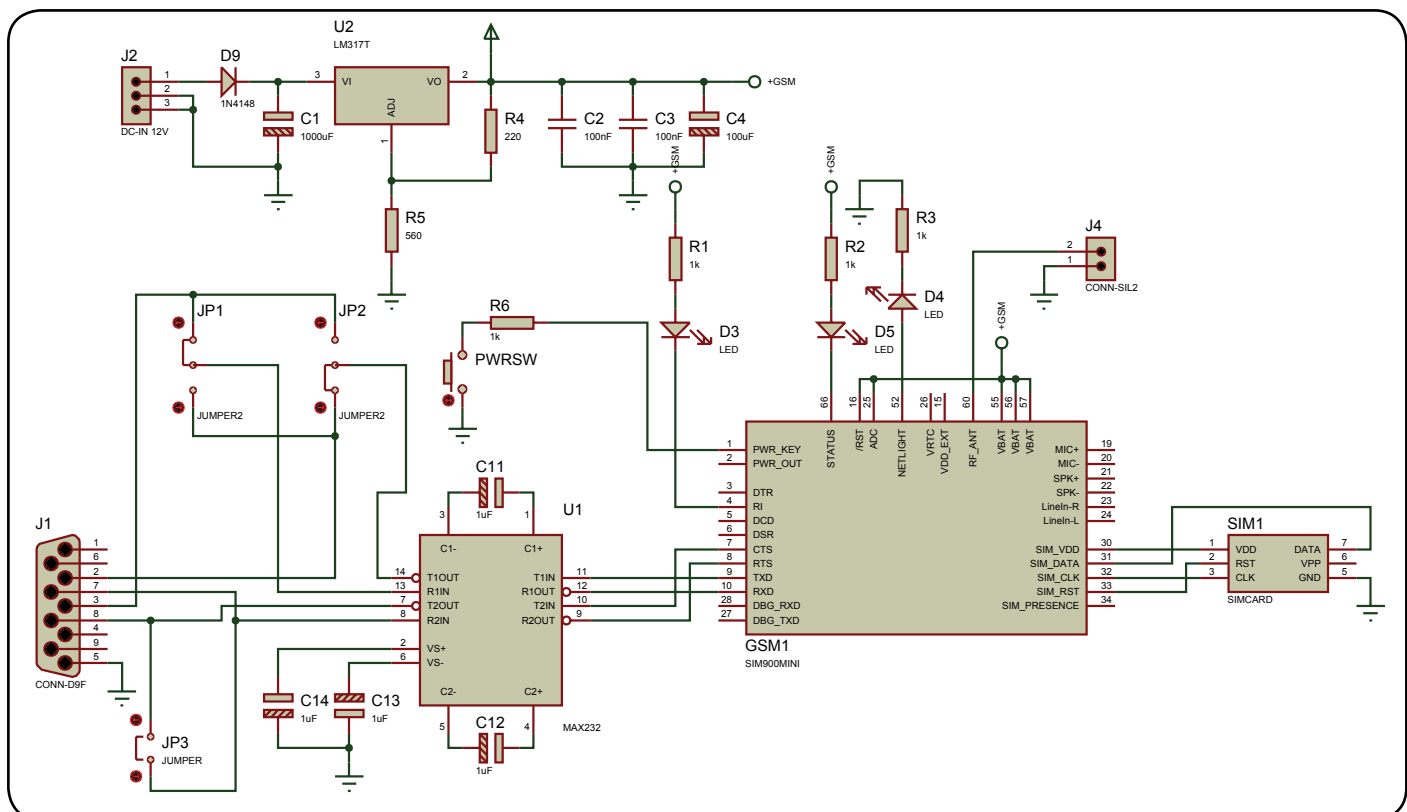
<http://eshop.eca.ir/link/850.php>





دستور	عملکرد
AT+CMGL="REC UNREAD"	خواندن تمام پیامک های دریافتی خوانده نشده.
AT+CMGL="REC READ"	خواندن تمام پیامک های دریافتی خوانده شده.
AT+CMGL="STO UNSENT"	خواندن تمام پیامک های ذخیره شده اما ارسال نشده.
AT+CMGL="STO SENT"	خواندن تمام پیامک های ذخیره شده و ارسال شده.
AT+CMGL="ALL"	خواندن تمام پیامک ها بدون قید و شرط.

عملکرد	دستور نمونه	Delflag
پاک کردن پیامک از خانه شماره index حافظه : به عنوان مثال دستور زیر پیامک موجود در خانه شماره ۲۷ حافظه سیم کارت را پاک می کند	AT+CMGD=27,0 OK	0
حذف تمام پیامک های دریافتی خوانده شده از حافظه.	AT+CMGD=0,1 OK	1
حذف تمام پیامک های دریافتی خوانده شده و ذخیره شده ارسال شده.	AT+CMGD=0,2 OK	2
حذف تمام پیامک ها غیر از پیامهای دریافتی خوانده نشده.	AT+CMGD=0,3 OK	3
حذف تمام پیامک ها بدون قید و شرط.	AT+CMGD=0,4 OK	4



مدار کلی یک GSM مودم نمونه با استفاده از ماژول Sim900 و ارتباط از طریق درگاه سریال رایانه

# پروگرامر و دیباگر J-LINK V8.0

## قابلیت آپدیت



برخی از ویژگی های مهم این دستگاه :

- ارتباط با کامپیوتر از طریق پورت USB 2.0
- توانایی پروگرام و دیباگ در کامپایلرهای مختلف (IAR, Keil, ...)
- پشتیبانی کامل از تمام میکروکنترلرهای ARM7/ARM9/ARM11 (Atmel, Philips, Cortex-M0/M1/M3, ...)
- پشتیبانی از Serial Wire Debug (SWD)
- پشتیبانی از Serial Wire Viewer (SWV)
- قابلیت شناسایی اتوماتیک قطعات
- سرعت JTAG تا 12MHZ
- سرعت دانلود تا 720 Kbytes/second
- سرعت DCC تا 800 Kbytes/second
- سازگاری کامل با IAR Embedded Workbench IDE
- بدون نیاز به تغذیه (تامین تغذیه از طریق پورت USB)
- قابلیت نمایش JTAG signal و قابلیت اندازه گیری target voltage
- پشتیبانی از دستگاههای متعدد
- دارای سیستم plug and play
- دارای کانکتور JTAG ۲۰ پین استاندارد
- رنج ولتاژ کاری بالا از 1.2V تا 3.3V (قابلیت کار تا 5V)
- دارای کابل USB و کابل ریبون ۲۰ پین
- دارای سیستم تغذیه هوشمند (محافظت در برابر اضافه بار)
- قابلیت استفاده از JTAG به صورت ۱۴ پین
- پشتیبانی از Memory viewer
- پشتیبانی از تمامی نرم افزارهای Flash programming
- پشتیبانی از سیستم Flash DLL
- پشتیبانی از Software Developer Kit (SDK)
- پشتیبانی از Embedded Trace Buffer (ETB)

## سازگاری کامل با ویندوز ۷

توضیحات تکمیلی : <http://eshop.eca.ir/link/506.php>



### آموزش طراحی تقویت کننده های ترانزیستوری



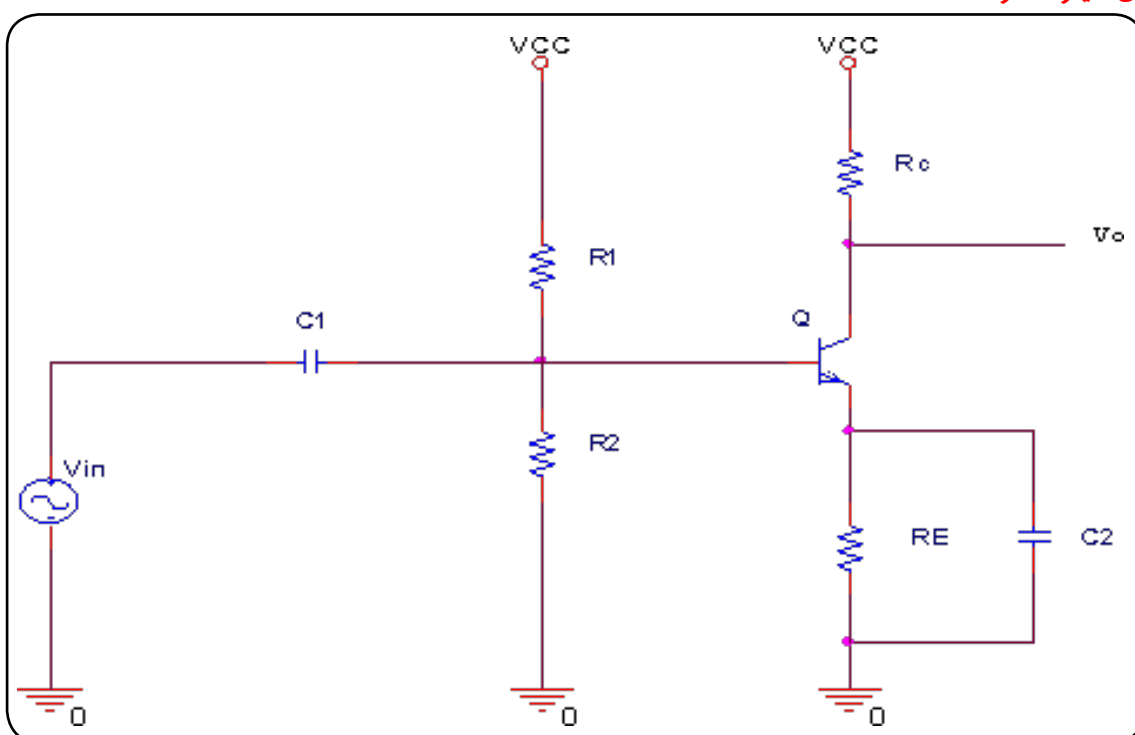
Alireza2s2003@gmail.com

نویسنده: علیرضا صمدی

تقویت کننده ها جزء جدایی ناپذیر مدارات الکترونیکی هستند. اکثر مدار های الکترونیکی دارای حداقل یک یا چند طبقه تقویت کننده برای تقویت سیگنال های ورودی به آنها هستند. عمده ترین قطعات برای تقویت سیگنال های ورودی به تقویت کننده ها آپ امپ و ترانزیستور هستند. هر دوی این قطعات برای تقویت ولتاژ یا جریان ورودی به آنها استفاده می شود. اما محدودیت عمده ی آپ امپ که عبارت است از پهنای باند کمتر آن نسبت به ترانزیستور، باعث شده اکثر تقویت کننده های موجود در مدارات از نوع ترانزیستوری باشد.

تقویت کننده های ترانزیستوری BJT به سه نوع عمومی امیتر مشترک، بیس مشترک و کلکتور مشترک تقسیم می شوند. نوع انتظاری که ما از عملکرد یک تقویت کننده داریم تعیین کننده نوع تقویت کننده ی مورد استفاده خواهد بود. گین ولتاژ یا جریان بالا، مقاومت ورودی زیاد و مقاومت خروجی کم. اینها عناوینی هستند که نوع تقویت کننده ی مورد استفاده را مشخص می کند. در این شماره از مجله به توضیح نحوه ی طراحی هر کدام از تقویت کننده های یاد شده در بالا پرداخته و پارامتر های مختلف آنها از جمله گین ولتاژ، گین جریان، مقاومت های ورودی و خروجی آنها را با یکدیگر مقایسه کرده و بر حسب کاربرد مورد انتظار بهترین تقویت کننده برای هر کدام از کاربردها معرفی می شود. همچنین در مورد ایراد یابی و عدم تطبیق نتایج تئوری و عملی طراحی ها نیز بحث خواهد شد.

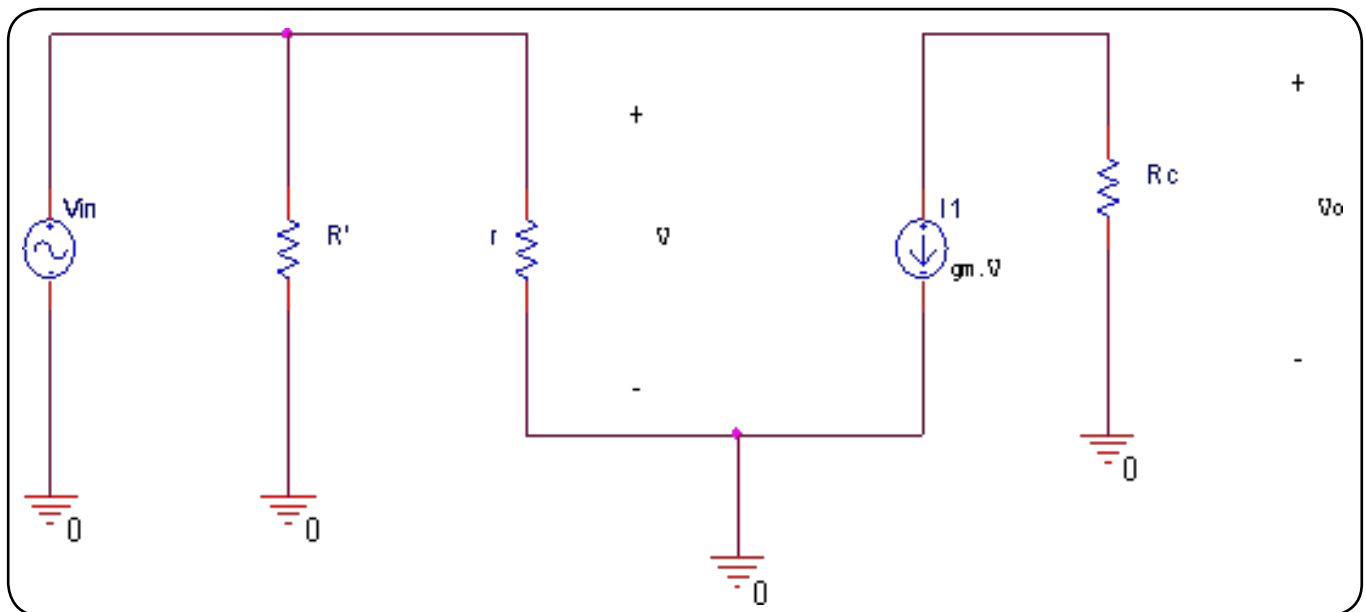
#### تقویت کننده ی امیتر مشترک



شماتیک کلی یک تقویت کننده ی امیتر مشترک

# آموزش طراحی تقویت کننده های ترانزیستوری

در حالت کلی برای طراحی تقویت کننده چند پارامتر به عنوان معلومات مسئله در نظر گرفته شده و با استفاده از آنها پارامترهای دیگر محاسبه می شوند. عموماً در تقویت کننده ها،  $V_{CC}$ ،  $ICQ$ ،  $A_v$ ،  $V_B$ ،  $S$  یا  $AI$ ، پارامترهای معلوم هستند که مقادیر المانهای دیگر بر حسب آنها محاسبه می شود.  $V_B$  ولتاژ تعیین ترانزیستور است که در تقویت کننده ها معمولاً برابر  $1/3V_{CC}$  یا  $1/2V_{CC}$  در نظر گرفته می شود و  $S$  تغییرات حرارتی ترانزیستور است که برابر  $R_1 || R_2 / R_E$  است. برای شروع طراحی تقویت کننده امیتر مشترک مدل  $\pi$  سیگنال کوچک آن رسم می شود :



مدل  $\pi$  سیگنال کوچک تقویت کننده امیتر مشترک

$$g_m = I_{CQ} / V_T$$

$$r = \beta / g_m$$

$$R' = R_1 || R_2$$

با KVL و KCL بر روی این مدل  $V_0 / V_i = A_v$  به صورت زیر به دست می آید.

$$A_v \sim g_m R_c$$

در تمامی تقویت کننده ها برای پیدا کردن مقاومت ورودی به صورت زیر جریان  $I_i$  را از ورودی  $V_i$  خارج کرده که به این صورت مقاومت ورودی به صورت زیر به دست می آید :

$$R_{in} = \frac{V_i}{I_i} = \frac{\beta R' (R_E + \frac{1}{g_m})}{R' + \beta (R_E + \frac{1}{g_m})}$$

با توجه به فرمول بالا مشاهده می شود هر چه قدر مقاومت های بایاس  $R_1$  و  $R_2$  بیشتر باشد مقاومت ورودی نیز بیشتر می شود که مورد دلخواه ما است. تا جایی که به ازای  $B (R_E + 1/g_m) \gg R'$  مقاومت ورودی تقریباً برابر  $B (R_E + 1/g_m)$  خواهد بود.

برای پیدا کردن مقاومت خروجی، در خروجی مدار  $V_{out}$  قرار داده و جریان  $I_{out}$  را از آن خارج می کنیم در این صورت مقاومت خروجی مدار برابر خواهد بود با :

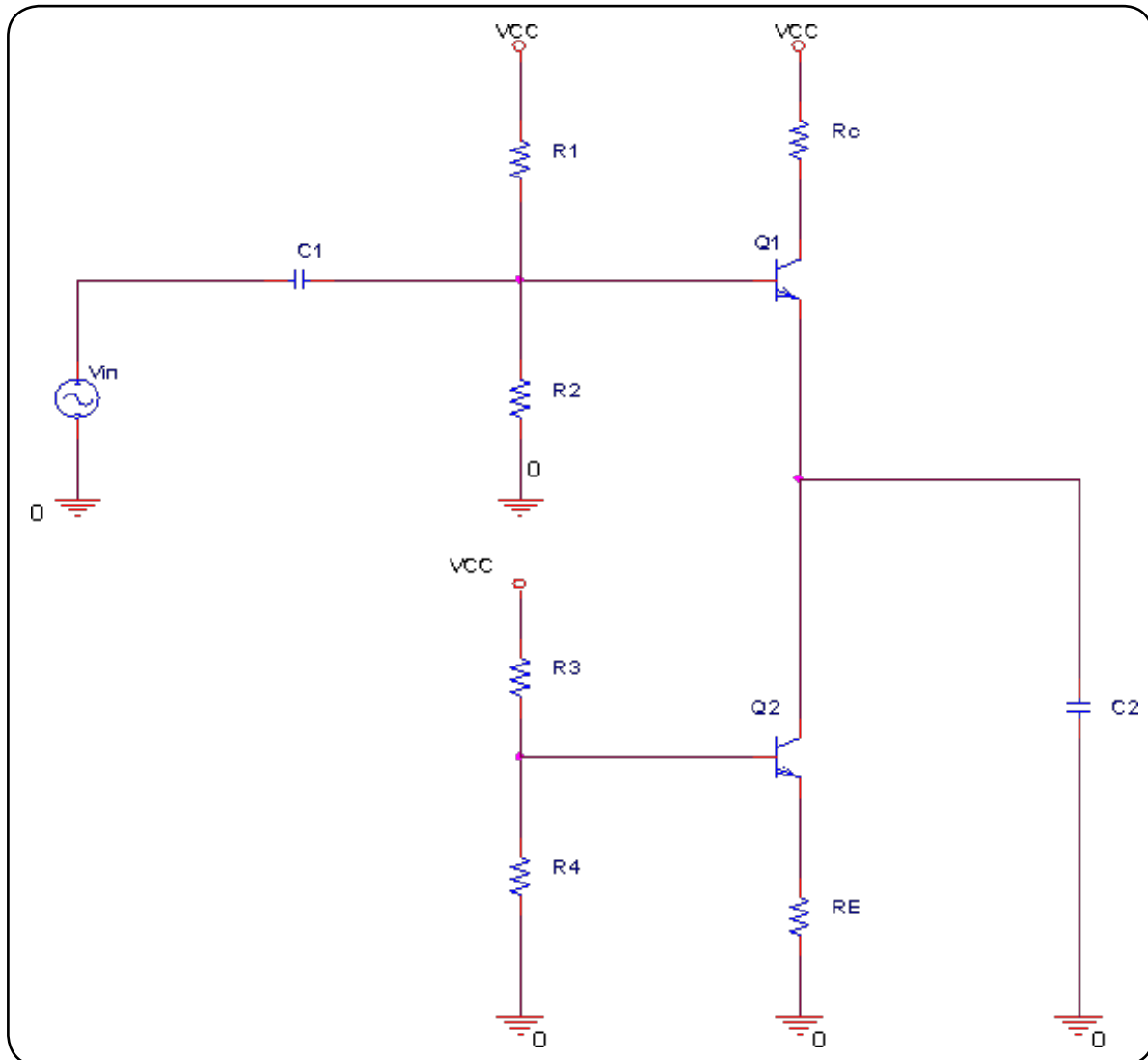
$$V_{out} / I_{out} = R_0 \sim R_c$$

هر چقدر مقاومت خروجی کمتر باشد این امر مورد دلخواه ما خواهد بود.



## تقویت کننده های امیتر مشترک با منبع جریان

گاهی اوقات به جای مقاومت امیتر منبع جریان قرار داده می شود. این کار باعث افزایش مقاومت ورودی، پایداری جریان کلکتور، پایداری گرمایی و..... می شود.



نحوه ی طراحی تقویت کننده امیتر مشترک با منبع جریان

همانند قسمت قبل معلومات مسئله  $S$ ،  $V_B$ ،  $AV$ ،  $ICQ$  هر دو ترانزیستور خواهد بود. همچنین در طراحی منبع جریان یک درجه ی آزادی دیگر نیز وجود دارد که می توان آن را بر روی مقاومت های با یاس  $R_3$  و  $R_4$  اعمال نمود. به عنوان مثال می توان به یکی از این دو مقاومت مقدار داده و مقاومت بعدی بر حسب آن حل شود. با فرضیات بالا طراحی این تقویت کننده با یک مثال توضیح داده می شود.

فرض کنید هدف طراحی یک تقویت کننده با گین ۱۰۰، جریان نقطه ی کار  $V_U = 9V$ ،  $I_{mA} = 1$  و  $S = 10$  است در این صورت داریم :

$$g_m = I_{CQ} / V_T \sim 40 \text{ mV}$$

$$AV = g_m \cdot R_c \implies R_c = 100 / 40 \text{ m} = 2.5 \text{ K} \sim 2.7 \text{ K}\Omega$$

$$V_{B1} = \frac{1}{2} V_{CC} = 4.5 \implies \frac{R_2}{R_1 + R_2} \times 9 = 4.5 \text{ v} \implies R_1 = R_2$$

می توان با استفاده از درجه آزادی مقدار این دو مقاومت را تعیین کرد مثلاً هر دو برابر ۱ کیلو اهم البته اگر در طراحی ها مقدار مقاومت ورودی مشخصی مد نظر ما باشد با استفاده از آن بدون استفاده از درجه آزادی مقادیر این دو مقاومت تعیین شود.

$$V_{B2} = \frac{1}{3} V_{CC} = 3 \text{ v} \implies R_1 = 2 R_2$$

$$R_2 = 1 \text{ K} \implies R_1 = 2 \text{ k} \sim 2.2 \text{ K}\Omega$$

# آموزش طراحی تقویت کننده های ترانزیستوری

$$V_{B2} = 3 \text{ V} \Rightarrow V_{E2} = 2.3 \text{ V} \Rightarrow R_E = 2.3 / I_{CQ} = 2.3 \text{ K}\Omega \sim 2.2 \text{ K}\Omega$$

به این ترتیب تمامی مقادیر المان های تقویت کننده امیتر مشترک با منبع جریان محاسبه شدند.

چند نکته بسیار مهم:

۱- برای اینکه ترانزیستور Q2 به عنوان منبع جریان عمل کند می بایستی جریان از کلکتور به امیتر آن جاری شود این امر هم مستلزم اینست که  $V_{CE2} > 0$  باشد یعنی  $V_{C2}$  از  $V_{E2}$  بیشتر باشد. واضح است که با توجه به اینکه  $V_{E1}V_{B1} - 0.7 = V_{C2}$  و  $V_{E2} = V_{B2} - 0.7$  برای

اینکه  $V_{CE2} > 0$  باشد می بایستی  $V_{B2} < V_{B1}$  انتخاب شود. به همین دلیل در فرضیات مسئله  $V_{B1} = \frac{1}{2}V_C$  و  $V_{B2} = \frac{1}{3}V_C$  انتخاب شد.

۲- در کاربردهای عملی همانند مثالهای قبل معمولاً از یک منبع تغذیه (VCC) استفاده می شود. ممکن است در مسائل از دو منبع تغذیه نیز استفاده شود. در این صورت نیز تمامی این مراحل طراحی به قوت خود باقی خواهد ماند و تنها می بایستی در محاسبه ولتاژ امیتر ترانزیستور یا ترانزیستورها با دقت بیشتری محاسبه شود.

۳- خازن های کوپلاژ C1 و بایپس C2 دارای مقادیری در حدود  $10\mu$  تا  $100\mu$  خواهند بود. خازن C1 برای جدا سازی DC منبع تغذیه از مقاومت های با یاس استفاده می شود. هر چه فرکانس ورودی کمتر باشد بهتر است مقدار این خازن بیشتر انتخاب شود تا در نقش مقاومت در مقابل ورودی عمل نکند. همچنین خازن C2 بایپس کردن مقاومت امیتر در حالت AC را دارد تا گین تقویت کننده کاهش نیابد. برای بایپس شدن کامل مقاومت امیتر نیز می بایستی با توجه به فرکانس ورودی مقدار مناسبی برای این مقاومت انتخاب شود. البته همانطور که گفته شد مقادیر  $10\mu$  تا  $100\mu$  برای این دو خازن محدوده ی مناسبی می باشند.

## ایراد یابی عملی تقویت کننده ی امیتر مشترک

گاهی اوقات بعد از طراحی کامل یک تقویت کننده (در اینجا امیتر مشترک) نتایج به دست آمده در عمل تفاوت های عمده ای با نتایج حاصل در تئوری دارند در اینجا تک تک عوامل این امر بررسی می شوند.

### ۱- اطمینان از صحیح بودن نتایج تئوری

بار دیگر محاسبات خود را در تئوری چک کنید و از صحیح بودن آنها اطمینان کامل به دست آورید. اصول طراحی تقویت کننده های ترانزیستوری به نوعی است که گاهی اوقات کوچکترین اشتباه در تئوری ممکن است به عدم کارکرد مدار منجر شود.

### ۲- کم بودن گین مدار

گاهی اوقات گین مدار یا به عبارت دیگر خروجی آن کمتر از حد انتظار است. به عنوان مثال ولتاژ خروجی کمتر از میزان به دست آمده در تئوری است.

در صورت اطمینان از بند ۱ یعنی طراحی صحیح مقادیر المان ها این امر می تواند به دلایل زیر اتفاق بیافتد :

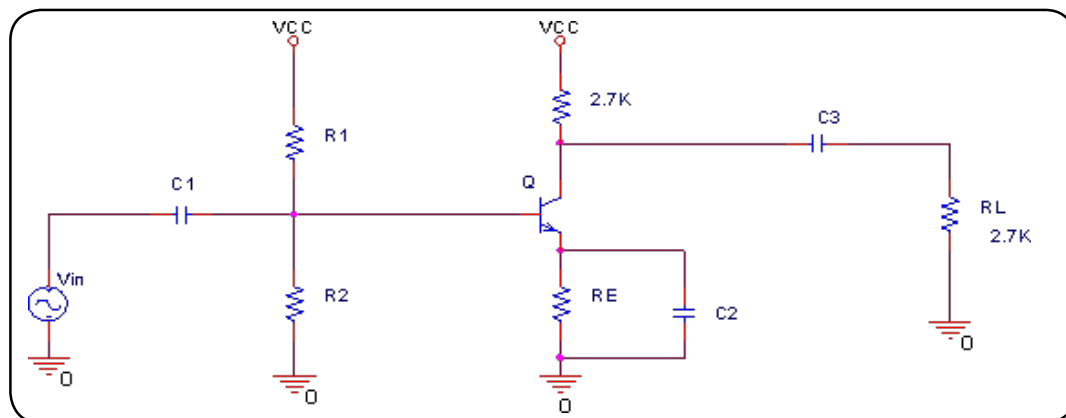
### الف) انتخاب گین بیش از حد مجاز

ترانزیستور های مورد استفاده ما دارای یک مقدار حداکثر تقویت هستند که با B آنها ارتباط مستقیم دارد. به عنوان مثال اگر ترانزیستوری ولتاژ  $10\text{ mV}$  را با بهترین طراحی ممکن حداکثر  $100$  برابر بتواند طراحی کند یعنی خروجی آن برابر  $1\text{ V}$  شود، دیگر نمی توان نوعی از طراحی را برای آن در نظر گرفت که به ازای ولتاژ  $10\text{ mV}$  ورودی دارای خروجی برابر  $2\text{ V}$  شود به عبارت بهتر بیشتر از حداکثر گین قابل استفاده برای یک ترانزیستور مشخص را نمی توان از آن ترانزیستور انتظار داشت. این حداکثر تقویت به عواملی همچون B ترانزیستور و پهنای باند مشخص شده برای آن ترانزیستور بستگی دارد. همچنین برای بدست آوردن این مقدار تقویت می بایستی در طراحی همه ی این موارد در نظر گرفته شود.

### ب) بالا بودن مقاومت خروجی تقویت کننده

گاهی اوقات در هنگامی که خروجی تقویت کننده (در اینجا ولتاژ مد نظر است) به بار وصل نشده تمامی مطلوبات ما بخوبی در خروجی دیده می شود و نتایج عملی با تئوری تطبیق کامل دارد.

اما زمانی که باری به خروجی تقویت کننده وصل می کنیم، خروجی به یکباره افت کرده و از مقدار قبلی خود فاصله می گیرد. دلیل اصلی این مشکل بالا بودن مقاومت خروجی تقویت کننده است. برای درک بهتر مسئله به شکل زیر توجه کنید :



بالا بودن مقاومت خروجی تقویت کننده

همانطور که در شکل مشخص است مقاومت خروجی تقویت کننده (که برابر  $R_c$  است) برابر  $2.7k$  است که یک بار  $2.7k$  به خروجی تقویت کننده نیز بسته شده است. واضح است که در حالت  $ac$  بار  $2.7k$  موازی مقاومت کلکتور  $2.7k$  شده و مقدار معادل این دو برابر  $1.35k$  اهم خواهد شد. یعنی می توان گفت در زمانیکه بار  $2.7k$  به خروجی تقویت کننده بسته می شود همانند این است که باری در خروجی قرار ندارد ولی مقاومت کلکتور  $2.7k$  را برداشته و یک مقاومت  $1.35k$  به جای آن قرار می دهیم که با توجه به  $A_v = g_m R_c$  چون  $R_c$  کاهش پیدا کرده در نتیجه گین مدار و باطبع خروجی آن کاهش پیدا می کند. این امر نشان می دهد برای اینکه در هنگام اتصال بار به خروجی تقویت کننده افت ولتاژی در خروجی روی ندهد می بایستی مقدار بار حداقل  $10$  برابر بیشتر از مقاومت خروجی تقویت کننده ( $R_c$ ) باشد تا در هنگامی که با آن موازی می شود مقدار  $R_c$  را کاهش ندهد. این امر را می توان به صورت دیگری هم بیان کرد.

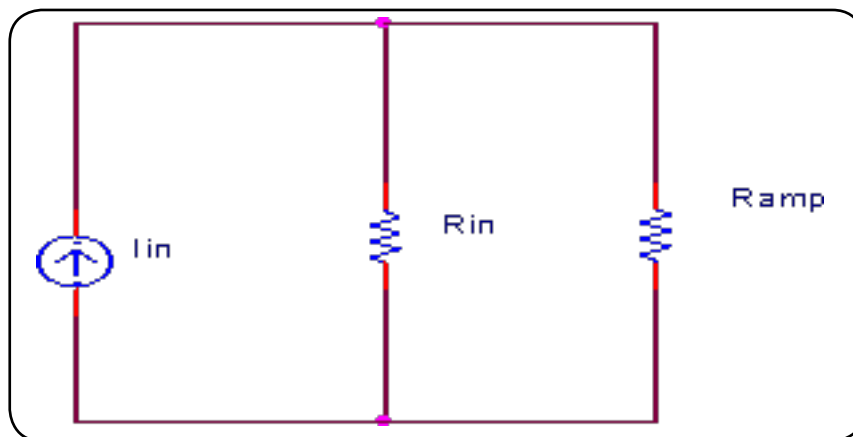
هر چقدر مقاومت خروجی تقویت کننده یا همان  $R_c$  کمتر باشد در هنگام اتصال بار به خروجی آن اثر بارگذاری کمتر شده به نحوی که هر چقدر مقاومت خروجی آن کمتر باشد موازی شدن بار و  $R_c$  مقدار  $R_c$  را کاهش نمی دهد در نتیجه  $A_v$  تقویت کننده ثابت خواهد ماند.

پس به طور خلاصه می توان گفت در طراحی تقویت کننده ها (در اینجا امپتر مشترک) می بایستی طراحی به نحوی صورت گیرد که مقاومت خروجی تقویت کننده تا حد امکان پایین باشد تا در هنگام اتصال بار به خروجی آن، مقدار خروجی کاهش پیدا نکند.

نکته مهم این است که ممکن است مقاومت خروجی تقویت کننده کم باشد ولی باری که به آن متصل می شود مقدار کمی را دارا باشد و افت ولتاژی در خروجی ایجاد کند. پس با توجه به مقاومت خروجی تقویت کننده باید یک حداقل مقدار برای بار متصل به تقویت کننده تعریف کرد. این حداقل مقدار برابر خواهد بود با  $10 R_0 \leq R_L$

### ۳) پایین بودن مقاومت ورودی تقویت کننده

گاهی اوقات پیش می آید که وقتی یک تقویت کننده اعمال می شود در هنگام اتصال آن به تقویت کننده ولتاژ ورودی به ناگهان افت پیدا می کند. این امر در اکثر مواقع به دلیل پایین بودن مقاومت ورودی تقویت کننده به وجود می آید به عنوان مثال شکل زیر دقت کنید:



پایین بودن مقاومت ورودی تقویت کننده

اگر  $R_{in}$  مقاومت درونی سیگنال ورودی و  $R_{amp}$  مقاومت ورودی تقویت کننده باشد واضح است اگر  $R_{amp}$  نسبت به  $R_{in}$  مقدار بزرگی نداشته باشد سیگنال ورودی به طور کامل به تقویت کننده نرسیده و توسط  $R_{in}$  مصرف می شود. در حالتی هم که ورودی ولتاژ باشد باعث افت ولتاژ در ورودی تقویت کننده می شود. این کار معمولاً در زمانی اتفاق می افتد که ورودی تقویت کننده ولتاژ، یکی از گره های مدار است که می بایستی تقویت شود و در هنگام اتصال آن گره به ورودی تقویت کننده به عنوان سیگنالی که می بایستی تقویت شود ولتاژ گره به صورت ناگهانی افت پیدا می کند.

پس به طور خلاصه هر چه قدر مقاومت ورودی تقویت کننده بیشتر انتخاب شود به همان اندازه افت ولتاژ در ورودی تقویت کننده کمتر شده و سیگنال به صورت کاملتری به تقویت کننده می رسد.

مطالب گفته شده نحوه طراحی تقویت کننده ها در حالت کلی و به صورت تخصصی نحوه طراحی تقویت کننده امپتر مشترک است. در شماره های بعدی به توضیح نحوه طراحی تقویت کننده های بیس مشترک و کلکتور مشترک پرداخته در نهایت نحوه عملکرد این سه تقویت کننده با یکدیگر مقایسه خواهد شد و تقویت کننده مناسب برای هر منظور معرفی خواهد شد.

# برد Mini2440 + 3.5" LCD 1GIG

**1GB Flash**



این برد بر اساس توانایی ها و قابلیت های میکروپروسسور S3C2440 طراحی شده است. این برد در اصل Embedded MotherBoard می باشد، از این رو که با فرکانس کاری ۵۳۳ مگاهرتز و داشتن پورت های همچون سریال، USB، صدا، شبکه، حافظه SD/MMC، VGA و... توانایی های یک مادر برد را دارا می باشد.

همچنین امکان نصب سیستم عامل های Windows CE، NET و Linux بر روی مادر برد به راحتی امکان پذیر می باشد. داشتن نمایشگر LCD رنگی ۳.۵" با دقت ۳۲۰\*۲۴۰ پیکسل به همراه صفحه حسگر لمسی و امکان اتصال برد مبدل VGA برای نمایشگر های CRT و LCD معمولی از دیگر قابلیت های جذاب این برد می باشد. ابعاد این برد ۱۰\*۱۰ سانتی متر می باشد. ابعاد بسیار کوچک این برد جهت کاربردهای جیبی (Pocetabl) و

فضاهایی که نیاز به یک مادر برد با سایز بسیار کوچک است، بسیار مناسب می باشد.

این برد نیاز به یک تغذیه ۵ ولت تک داشته و با توجه به عدم تلفات حرارتی فاقد هرگونه فن و سیستم خنک کننده می باشد. پورت ها و خروجی ها :

:: یک عدد پورت شبکه (RJ45) با سرعت 10/5 Mbps

:: سه عدد پورت سریال

:: یک عدد پورت HOST USB

:: یک عدد پورت USB DEVICE

:: یک عدد سوکت کارت های حافظه SD/MMC

:: یک عدد خروجی استریو صوت

:: یک عدد میکروفون

:: یک عدد کانکتور ۱۰ پین JTAG

:: چهار نمایشگر LED

:: شش عدد کلید فشاری همه منظوره

:: یک عدد باز روی خروجی PWM

:: یک عدد پتانسیومتر روی ورودی A/D

:: یک عدد کانکتور ۲۰ پین به منظور اتصال دوربین

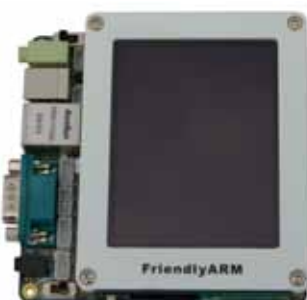
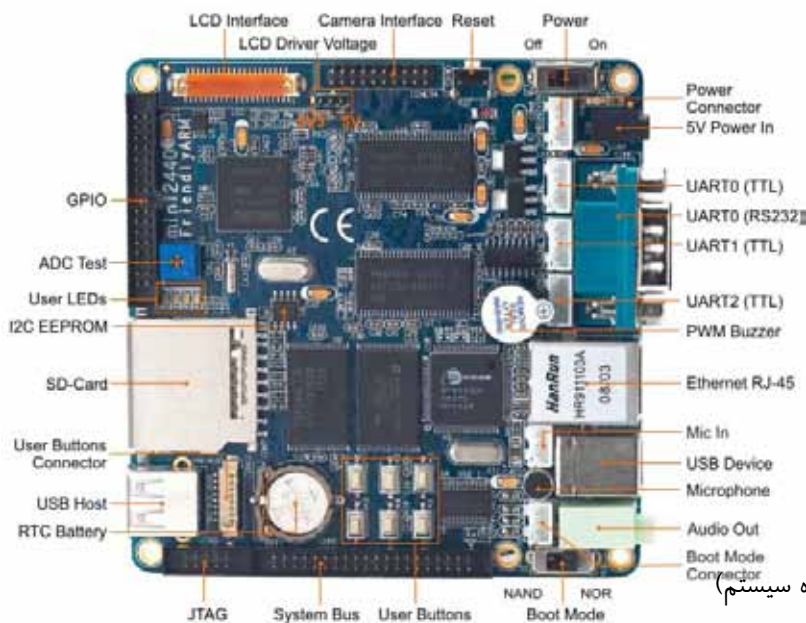
:: یک عدد باتری پشتیبانی حافظه RTC روی برد

:: تغذیه ورودی ۵ ولت به همراه کلید و نمایشگر تغذیه

:: یک عدد کانکتور ۴۰ پین به منظور اتصال LCD

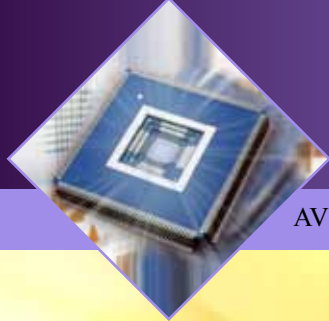
:: دو عدد کانکتور توسعه (۳۴ پایه I/O و ۴۰ پایه گذرگاه سیستم)

:: یک عدد کلید ریست میکرو پروسور



توضیحات تکمیلی : <http://eshop.eca.ir/link/851.php>





نکاتی در مورد طراحی سخت افزار

## مدارات مبتنی بر AVR

amirali.b@gmail.com

مترجم: امیرعلی بلورچیان

## AVR Hardware Design Considerations

این مقاله به منظور پاسخ دادن به سوالات و مشکلاتی که در حین طراحی مدارات مبتنی بر میکروکنترلر AVR مواجه می شوید گردآوری شده است. برای اینکه بتوان توسط میکروکنترلر AVR مداراتی عملیاتی و صنعتی طراحی نمود، در این مقاله به جای پرداختن به جزئیات، مشکلات معرفی و جهت رفع آنها راهکارهای عملی ارائه گشته است. اطلاعاتی که در ادامه مطالعه خواهید نمود، مجموعه ای از اطلاعات جالب منتشر شده توسط شرکت Atmel می باشد.

### منبع تغذیه

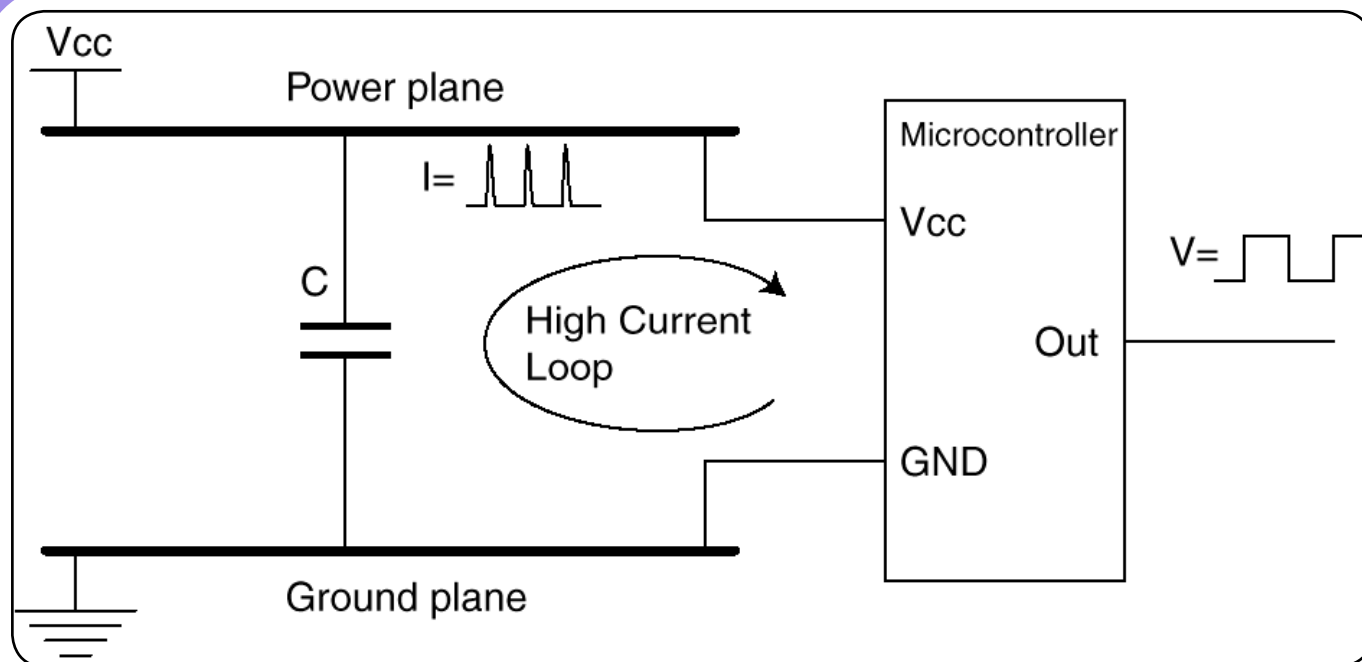
هنگام طراحی منبع تغذیه برای مدارات AVR باید دو مورد کلی در نظر گرفته شود. حفاظت ESD و انتشار نویز. هر دو موضوع در دستورالعمل AVR040 آورده شده است و خلاصه ای کوتاهی از آن ها را در اینجا مطرح می کنیم.

### تغذیه های دیجیتال

با مطالعه دیتاشیت میکروکنترلرهای AVR ممکن است خواننده گمراه شده و گمان کند که منبع تغذیه بخش مهمی نیست. این قسمت رنج ولتاژ بسیار وسیعی داشته و فقط چند میلی آمپر جریان می کشد. ولی این را باید بدانید که در همه ی مدارات دیجیتال، منبع جریان ارزش قابل توجهی دارد. جریان در پیک های کوچک و در جهت لبه های ساعت کشیده می شود و اگر خطوط I/O تعویض شوند، پیک ها بالاتر خواهند بود. اگر همه هشت خط I/O یک پورت همزمان تغییر مقدار بدهند، پالس های جریان در خطوط تغذیه قدرت می توانند تا چند صد میلی آمپر بالا بروند. اگر خطوط I/O استفاده نشوند، پالس ها در حد چند نانو ثانیه خواهند بود.

این نوع جریان ها نمی توانند در خطوط منبع تغذیه به مدت طولانی بوجود آیند؛ منبع اصلی این نوع جریان، خازن دکوپلینگ (decoupling) است.

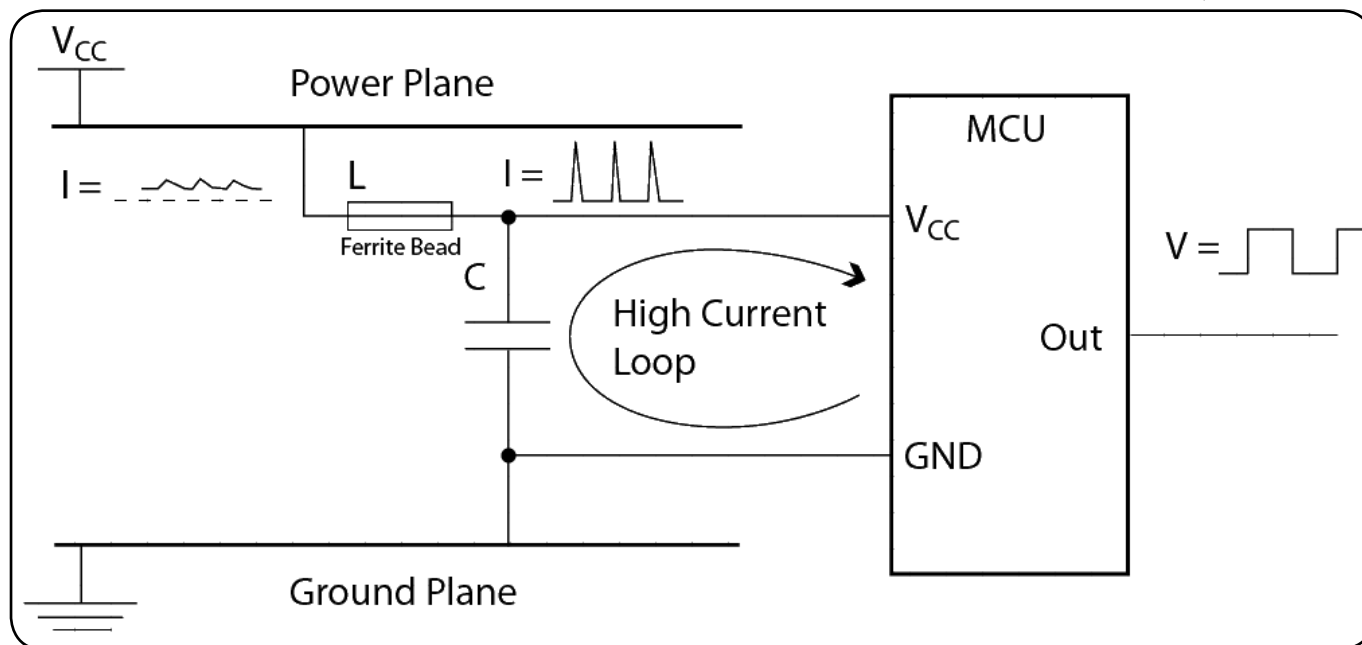
شکل ۱ نشان دهنده ی دکوپلینگ ناکافی است. خازنی که در فاصله دوری از میکروکنترلر قرار گرفته، باعث ایجاد حلقه ی جریان بالا می شود.



شکل (۱): decoupling نادرست

سطوح منبع و زمین به عنوان قسمتی از حلقه جریان بالا در این قسمت هستند که نتیجه‌ی آن پخش ساده‌ی نویز به سایر قطعات برد و افزایش نویز در برد می باشد. کل سطح زمین همانند یک فرستنده به جای عبور جریان بالا به انتشار نویز کمک می کند. پس بهترین حالت این است که خازن دکوپلینگ دقیقاً در همان قسمتی باشد که پایه های زمین و تغذیه وصل شده اند و به همین علت این خازن را بصورت مستقیم از پشت برد به پایه تغذیه و زمین وصل می کنند.

شکل (۲) مکان قرار گرفتن بهتر خازن ها را نشان می دهد. خطوطی که قسمتی از حلقه با جریان بالا هستند جزو خطوط جریان منبع تغذیه و زمین نبوده و این بسیار مهم است زیرا در صورتی که این دو از هم جدا نباشند نویز بسیار زیادی منتشر خواهد شد. بعلاوه، شکل بهبود یافته تری از دکوپلینگ را نشان می دهد. فریت بید در کاهش نویز سوچ منبع نقش مهمی دارد. جهت کاهش افت ولتاژ بر روی فریت بید، می بایست مقاومت آن بسیار پایین باشد.



شکل (۲): مکان قرار گرفتن بهتر خازن های دکوپلینگ

در حالت کلی دکوپلینگ میکروکنترلرهای AVR که پایه های زمین و تغذیه آنها کنار هم قرار گرفته اند (مانند Atmega8535) خیلی بهتر از میکروکنترلرهای AVR است که پایه های آن ها بصورت استاندارد و متضاد (مانند Atmega8514) در دو طرف میکرو قرار گرفته اند. این مشکل را می توان با استفاده از پکیج TQFP میکروکنترلرها رفع نمود. در این حالت دکوپلینگ خازن ها به صفر نزدیک می شود. میکروکنترلرهایی که دارای چند جفت پایه منبع و زمین هستند، می بایست هر جفت پایه به خازن دکوپلینگ خود متصل شوند. برای تثبیت نمودن تغذیه کلی مدار نیز باید از یک خازن تانتالیوم چند میکروفارادی استفاده شود.

## تغذیه آنالوگ

قطعات AVR که ADC داخلی دارند، پایه ولتاژ آنالوگ جداگانه‌ای با نام AVCC دارند. این منبع ولتاژ جداگانه جهت کاهش نویز ایجاد شده

توسط مدارات آنالوگ در مدارهای دیجیتال می باشد. برای داشتن دقت بالا در ADC، ولتاژ آنالوگ باید به صورت جداگانه دکوپله شود و این کار به صورت مشابه در ولتاژ مدارات دیجیتال انجام می شود. AREF نیز می بایستی دکوپله شود، مقدار معمول این خازن  $10\text{ nf}$  است. اگر زمین آنالوگ (AGND) جداگانه ای موجود باشد، زمین آنالوگ باید از زمین دیجیتال کاملاً جدا شود. در اینصورت زمین های دیجیتال و آنالوگ فقط در یک نقطه (منشا GND) به هم وصل می شوند. برای مثال همگی به GND منبع تغذیه وصل می شود.

## اتصال پایه ی Reset به AVR

پایه ی Reset در میکروکنترلرهای AVR، Activelow است و Low کردن پایه Reset به صورت خارجی باعث Reset شدن AVR می شود. به طور کلی Reset کردن ۲ کاربرد دارد:

(۱) برای آزاد کردن تمام پایه ها از طریق Tri-State کردن تمام پین ها به استثنای پایه Xtal، بازنشانی تمام رجیسترهای I/O و صفر کردن شمارنده پروگرام.

(۲) برای ورود به حالت پروگرام شدن.

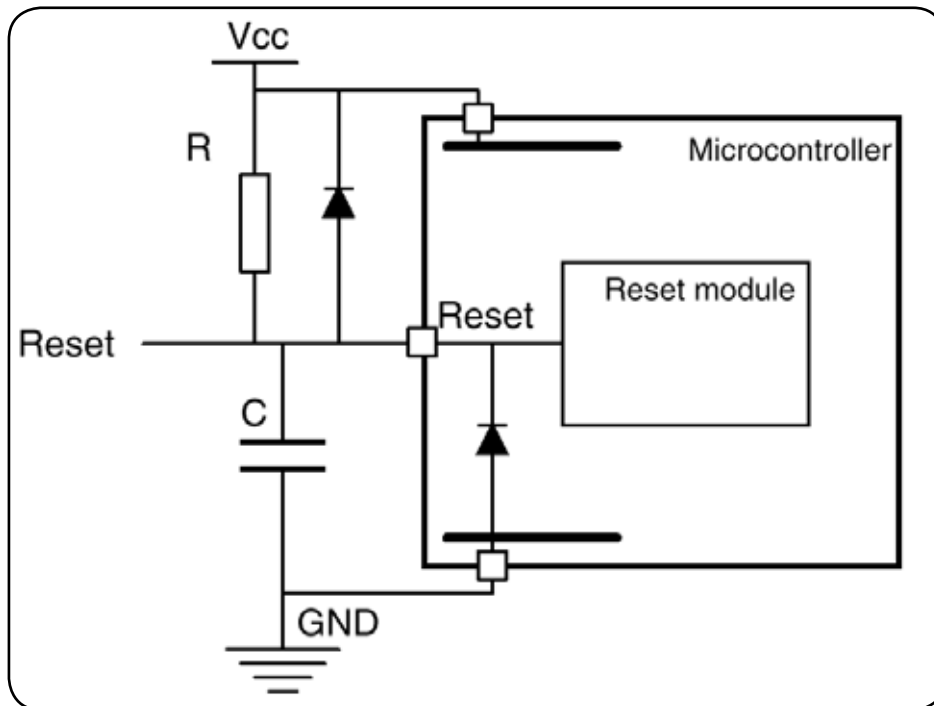
بعلاوه امکان دارد که با اتصال پایه Reset به ولتاژ بسیار بالا میکروکنترلر به حالت پروگرام شدن High-Voltag/Parallel وارد شود که عبارت ولتاژ بسیار بالا چیزی در حدود  $11.5 - 12.5$  ولت است.

خط Reset دارای مقاومت داخلی Pull-Up است ولی اگر محیط دارای نویز باشد، این مقاومت داخلی قابل اطمینان نخواهد بود. پایه Reset هم برای پروگرام نمودن High Voltag و هم برای پروگرام کردن معمولی از طریق یک مقاومت Pull-Up مورد استفاده قرار می گیرد.

با اتصال مقاومت Pull-Up امکان پروگرام نمودن میکروکنترلر با ولتاژ بالا یا پایین معمولی ایجاد می شود. این مقاومت Pull-Up اطمینان می دهد که ناخواسته ولتاژ پایین نخواهد رفت. در حالت تئوری مقاومت Pull-Up می تواند هر اندازه ای باشد، اما اگر AVR توسط STK500/AVRISP پروگرام شود، مقاومت Pull-Up نباید به قدری باشد که پروگرامر قادر به Low کردن پایه Reset نباشد. مقاومت Pull-Up پیشنهاد شده در این حالت  $4.7\text{ K}$  است.

در حالت دیباگ، برای اینکه دیباگ میکروکنترلر به خوبی عمل کند مقاومت pullup نباید کوچکتر از  $10\text{ K}$  باشد. پیشنهاد ما برای محافظت بیشتر پایه Reset از نویز زیاد، اتصال یک خازن از پایه Reset به زمین است. البته این کار در میکروکنترلر AVR الزامی نیست، چون در داخل AVR فیلتر پایین گذاری جهت محافظت قرار داده شده است و در نتیجه قرار دادن یک خازن اضافی محافظت مضاعف است. البته قرار دادن این خازن در حالت دیباگ و استفاده از PDI توصیه نمی شود.

در صورتی که از پروگرامر High Voltag استفاده نمی کنید، پیشنهاد می شود یک دیود محافظ ESD از Reset به VCC وصل شود. بعلاوه یک دیود زener می تواند ولتاژ Reset را نسبت به زمین محدود کند. برای محیط های با نویز بالا، دیود زener توصیه می شود. تمامی قطعاتی که جهت حفاظت استفاده میشوند، باید حداکثر امکان نزدیک به پایه Reset وصل شوند. شکل ۳ نشان دهنده ی مدار پیشنهاد شده در پایه Reset است.



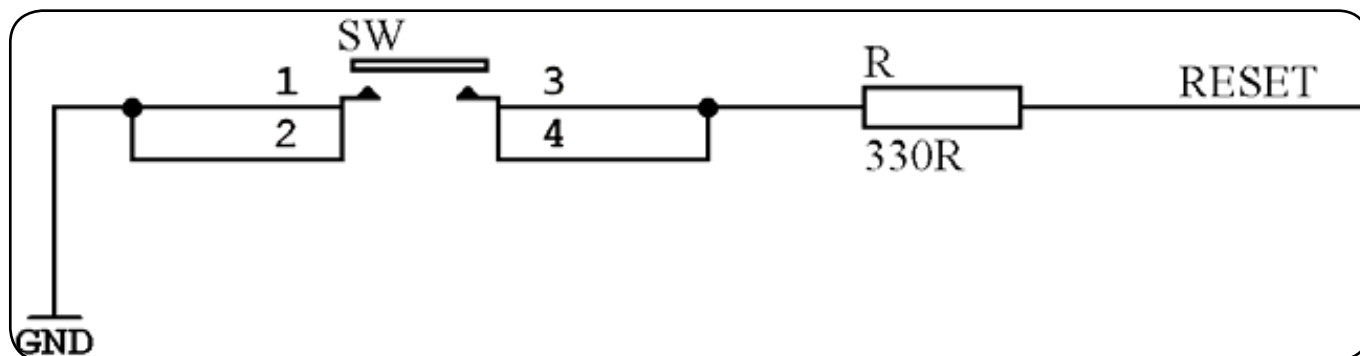
شکل (۳): مدار پیشنهاد شده در پایه Reset جهت محافظت در برابر نویز محیط

## سوئیچ های خارجی Reset

در صورت اتصال سوئیچ خارجی به پین Reset می بایست یک مقاومت با آن سری شود. سوئیچی که فشار داده می شود می تواند پیک بالایی را ایجاد کند. این کار می تواند باعث ایجاد Spike هایی در حدود ۲ تا  $10\text{ ms}$  شده و تا تخلیه خازن به طول خواهد کشید. ترک های PCB قادر به ایجاد اندوکتانس های کوچکی هستند که با توجه به رابطه  $V_i = L \cdot di/dt$  می توانند ولتاژ زیادی را ایجاد نمایند. این مقاومت سری در

# نکاتی در مورد طراحی سخت افزار مبتنی بر AVR

مسیر سوئیچ و Reset می تواند این مشکل را برطرف کند.



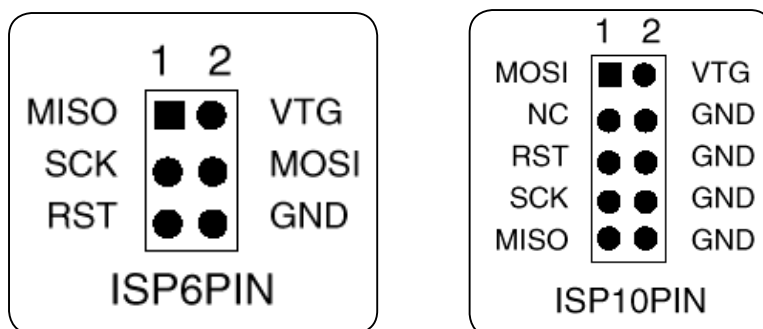
شکل (۴): اتصال سوئیچ به پایه Reset

## اتصال خطوط ISP

میکروکنترلرهای AVR دارای یک یا چند اینترفیس برای پروگرام کردن توسط سیستم (ISP) هستند. این خطوط برنامه ریزی، Flash، EEPROM، Lock Bits و اکثر Fusebit ها در تمام AVR ها (به استثنای ATtiny11 و ATtiny28) به کار می روند. بعضی از اینترفیس های ISP می توانند برای دیباگ کردن نیز مورد استفاده قرار گیرند. برای همین پیشنهاد می شود که همواره در دستگاه های طراحی شده، پورت ISP استاندارد را جهت مصارف آینده در دسترس قرار دهند.

## اینترفیس پروگرام کردن SPI

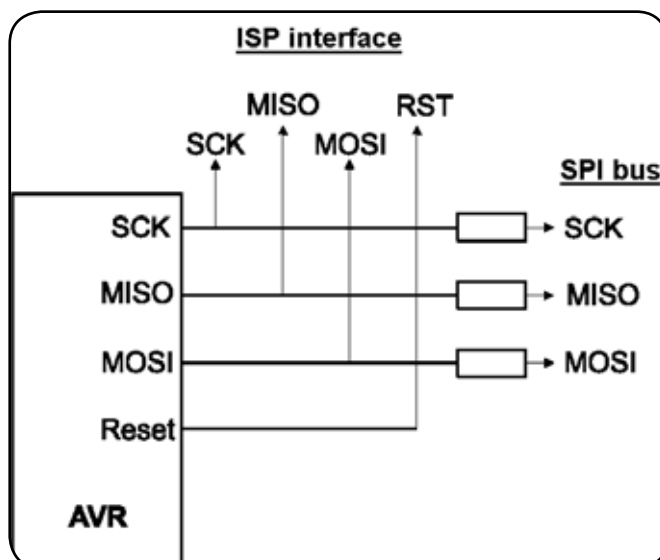
در قطعاتی که از SPI برای ISP استفاده شده است، این پایه ها معمولاً بر روی پین های SPI معمولی قرار می گیرند، در غیر این صورت روی پین هایی با کارایی های دیگر می توانند باشند. دو اتصال شش سیمه و ده سیمه استاندارد توسط پروگرامر Atmel ISP در شکل ۵ نشان داده شده اند. علاوه بر خطوط اطلاعات (MISO و MOSI)، کلاک (SCK) ولتاژ مرجع (VTG)، زمین و Reset نیز در این اتصالات تعبیه شده اند.



شکل (۵): کانکتورهای ISP استاندارد

## نحوه استفاده از پایه های پروگرامر ISP در موارد دیگر

هنگام اتصال دستگاه به ISP، پروگرامر باید در برابر تمام موارد بجز AVR اصلی محافظت شود برای همین بهتر است خطوط SPI مانند اینترفیس ISP طراحی شوند. (شکل ۶)



شکل (۶): اتصال خطوط SPI به اینترفیس ISP



# نکاتی در مورد طراحی سخت افزار مبتنی بر AVR

دقت شود که AVR هیچوقت خطوط SPI را درایو نخواهد کرد مگر اینکه پایه Reset به نشانه ورود به حالت پروگرام کردن فعال شود.

## اینترفیس JTAG

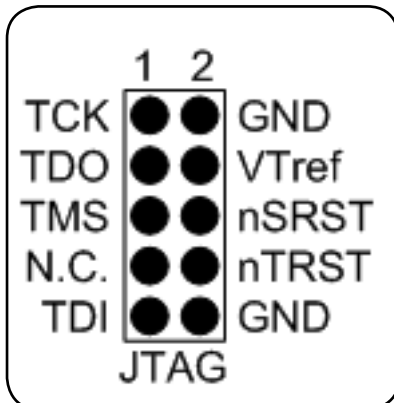
بعضی میکروکنترلرها دارای اینترفیس JTAG هستند که هم برای پروگرام کردن و هم برای دیباگ می توانند استفاده شوند. خطوط JTAG با ورودی های آنالوگ تداخل داشته و در نتیجه پروگرامر JTAG باید این خطوط را کنترل کند. پروگرامرهایی مانند JTAGICE MKII قدر به کنترل بارهای مقاومتی هستند ولی می بایست از بارهای خازنی دور کرد.

## نحوه استفاده از پایه های پروگرام JTAG در موارد دیگر

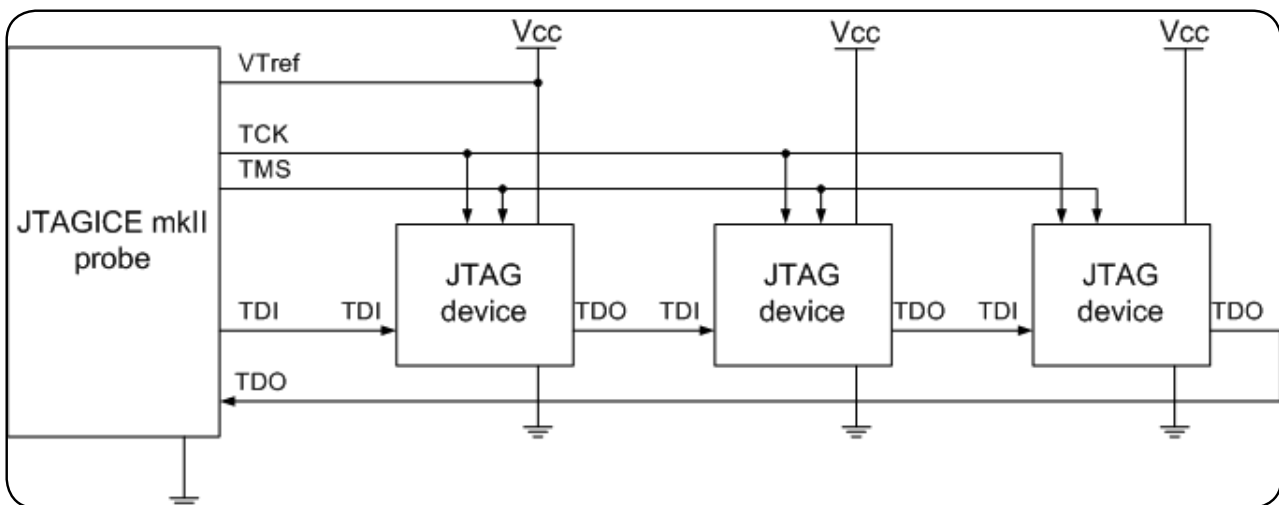
با ساخت یک زنجیره JTAG، یک پروگرامر JTAG می تواند بر مبنای اینترفیس ISP تعداد قطعات زیادی را پوشش دهد.

اگر پایه های JTAG برای مصارف دیگری نیز مورد استفاده قرار گیرند، مقاومت های حفاظتی همانطور که در شکل ۸ نشان داده شده مورد نیاز خواهد بود.

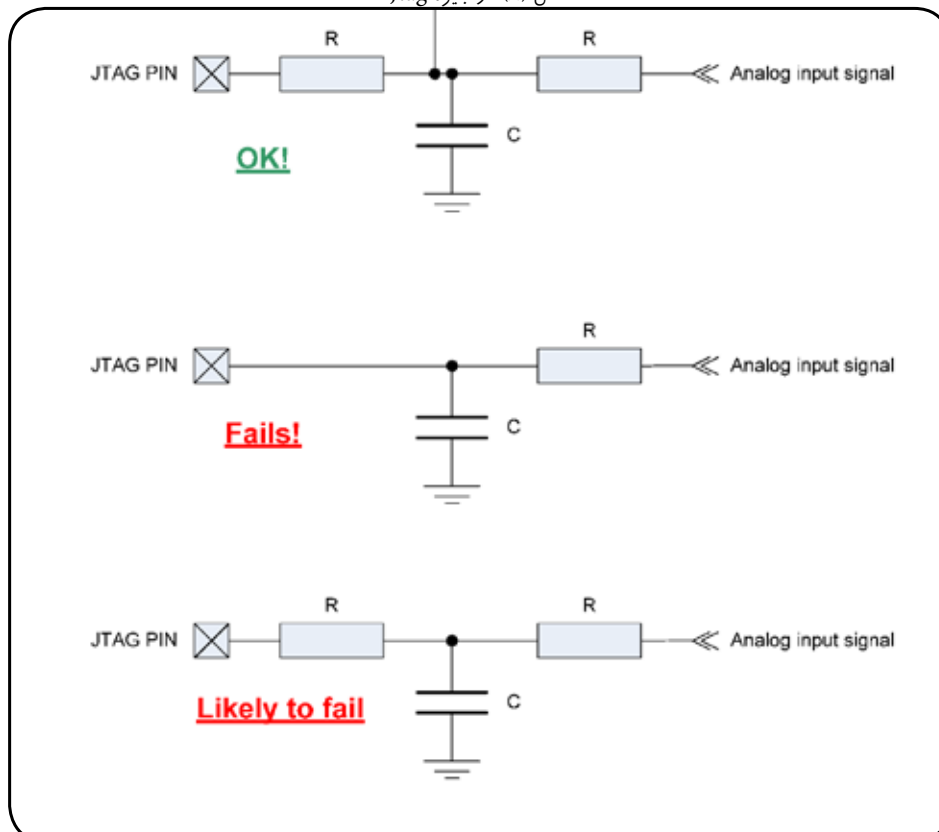
برای مثال اگر برای ورودی آنالوگ مورد استفاده قرار گیرند باید از فیلترهای آنالوگ در ورودی استفاده کرد.



شکل (۷): کانکتور JTAG استاندارد

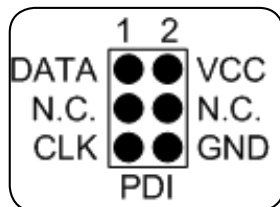


شکل (۸): زنجیره Jtag



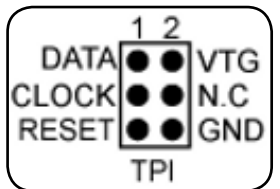
شکل (۹): روش های صحیح و غلط استفاده از فیلترهای موجود بر پایه های Jtag

## اینترفیس PDI



اینترفیس پروگرام و دیباگ PDI یک اینترفیس ۲ سیمه است که با میکروکنترلرهای خانواده AVR-XMEGA معرفی شده است. کانکتور استاندارد PDI در شکل ۱۰ نشان داده شده است. تنها ۲ پین PDI\_DATA و Reset که PDI\_CLK نامیده میشوند، مورد استفاده قرار گرفته است.

## اینترفیس TPI



Tinu Programming Interface برای میکروکنترلرهای TinyAVR طراحی گردیده و دارای کمترین تعداد پین ممکن می باشد. تنها ۳ پین Reset و TPICLK و TPIDATA برای پروگرام کردن مورد نیاز هستند.

## غیر فعال کردن پایه Reset

پایه Reset می تواند با تنظیم رجیستر RSTDISBL تبدیل به پایه I/O شود. این عامل باعث غیر فعال شدن سایر عملکردهای Reset می شود و برای بازگرداندن آن نیاز به پروگرامر High Voltage است.

شکل (۱۰) کانکتورهای استاندارد

## استفاده از کریستال و تشدید کننده سرامیکی (Ceramic Resonator)

بسیاری از میکروکنترلرهای AVR می توانند با منابع کلاک مختلف کار کنند. منابع کلاک خارجی عبارتند از CLOCK و نوسان ساز RC کریستال و تشدید کننده سرامیکی. استفاده از کریستال و تشدید کننده های سرامیکی در برخی از طراحی ها باعث ایجاد برخی از مشکلات می شوند، که علت این امر عدم درک کامل منابع کلاک می باشد. در ادامه با منابع کلاک کریستال و تشدید کننده های سرامیکی آشنا خواهیم شد.

## انتخاب منبع کلاک در AVR

منابع کلاک در تمام خانواده AVR از طریق فیوز بیت ها تنظیم می گردد، به جز XMEGA که این تنظیمات از طریق نرم افزار قابل انتخاب است. دقت داشته باشید که فیوزبیت ها هنگام Erase کردن میکرو پاک نمی شوند. کلاک های مربوط به این بحث عبارتند از :

- “Ext. low-frequency crystal”
- “Ext. crystal oscillator”
- “Ext. ceramic resonator”

## کلید در مورد کریستال و تشدید کننده های سرامیکی

نوع معمول کریستال های استفاده شده برای AVR، کریستال تشدید کننده موازی AT-CUT است. تشدید کننده های سرامیکی بسیار شبیه کریستال تشدید کننده موازی AT-CUT است ولی می توان گفت که مدل ارزاتر و کم کیفیت تر آن می باشند. تشدید کننده های سرامیکی دارای Q-Value کمتری هستند که هم مزیت و هم ایراد آنها به حساب می آید. به سبب Q-Value پایین، فرکانس نوسان ساز برای تشدید کننده سرامیکی به آسانی می تواند به فرکانس دلخواه تغییر یابد. اما همین عامل باعث حساسیت بسیار بالا به دما و تغییرات بار شده است که باعث تغییرات ناخواسته ی فرکانس می شود. مزیت اصلی تشدید کننده سرامیکی به خاطر شروع سریع تر نسبت به کریستال است. در حالت کلی کریستال و تشدید کننده سرامیکی متمایز نیستند و کلمه مشابه نوسانگر برای هر دو مورد استفاده می شود.

	Ceramic resonator	Quartz crystal
Aging	±3000ppm	± 10ppm
Frequency tolerance	± 2000-5000ppm	± 20ppm
Freq. temperature characteristics	± 20-50ppm/°C	± 0.5ppm/°C
Frequency pullability	± 100-350ppm/pF	± 15ppm/pF
Oscillator rise time	0.01-0.5ms	1-10ms
Quality factor (Qm)	100-5000	10 <sup>3</sup> -5·10 <sup>5</sup>

جدول ۱

## میزان خازن کریستال پیشنهادی

برای محاسبه خازن های مورد استفاده برای کریستال های AVR بر اساس روابط ریاضی به مقاله AVR042 شرکت Atmel مراجعه نمایید. در این بخش میزان خازن های کریستال برای حالت های مختلف پیشنهاد می شوند. وقتی از ext. crystal oscillator استفاده می کنیم، کریستال ها باید با فرکانس نامی ۴۰۰ kHz و بالا تر باشد. در این حالت خازن هایی در رنج ۲۲-۳۳pF بهینه خواهند بود.

وقتی از ext. low frequency crystal استفاده می کنیم، میکرو برای استفاده از کریستال ۳۲,۷۶۸kHz آماده می شود و در این حالت خازن های داخلی AVR می توانند جوابگو باشند. با تنظیم فیوزبیت CKOPT می توان خازن داخلی را بر پایه های XTAL1 و XTAL2 فعال نمود. مقدار این خازن ها در حدود ۲۰P است ولی می تواند بیشتر هم باشد. در صورت استفاده از کریستال ۳۲,۷۶۸kHz و در صورتی که زمان آماده سازی بیشتری مورد نیاز نباشد، خازن های داخلی کافی هستند ولی در غیر اینصورت باید از خازن های خارجی استفاده نمود. در صورت استفاده از ext. ceramic resonator حتماً توصیه می شود که برای محاسبه میزان خازن از روابط ریاضی موجود در دیتاشیت قطعه استفاده نمایید.

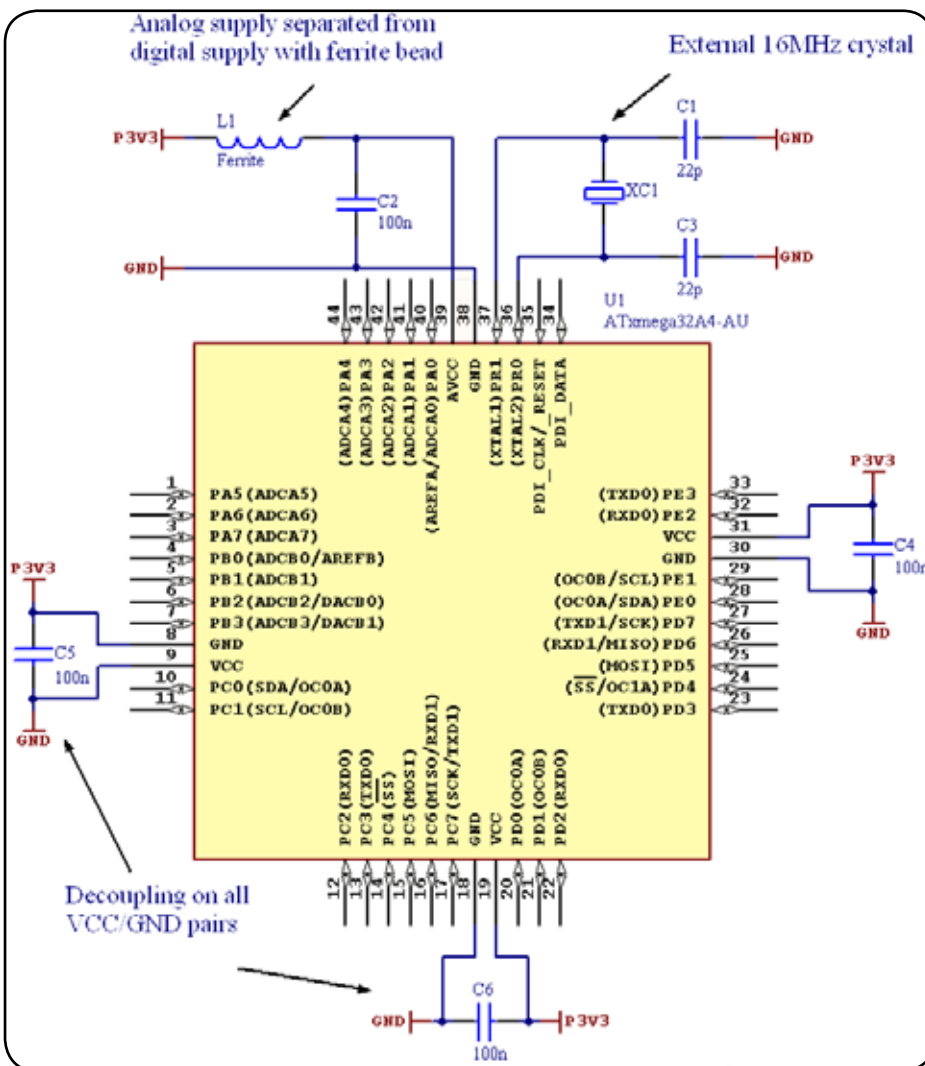
## کریستال RTC

بسیاری از AVR ها امکان استفاده از کلاک های ناهماهنگ در ساختار Timer یا Counter را دارند. کریستال ۳۲,۷۶۸kHz می تواند به پایه های Toscx میکروکنترلر AVR متصل شود. در برخی از AVR ها مدار اسیلاتور داخلی با Counter حالت واقعی استفاده می شود. در بار خازنی حدود ۲۰PF ایجاد می کند که برای کریستال معمولی ۳۲,۷۶۸KHZ اختصاص داده شده است. برای اطلاع در مورد خازن و مقدار آن به دیتاشیت مراجعه کنید. در صورتی که مقدار خازن های داخلی اعمال شده به کریستال خوب نباشند می توانید از خازن های خارجی استفاده کنید.

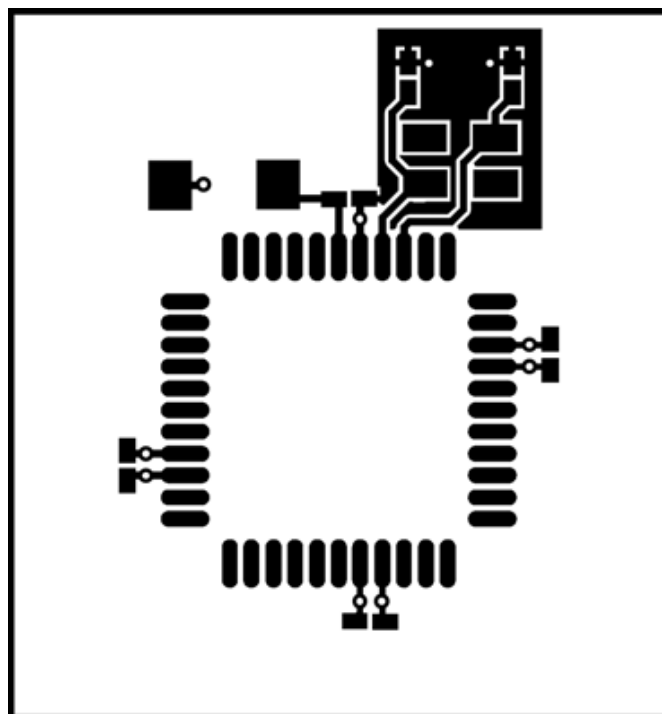
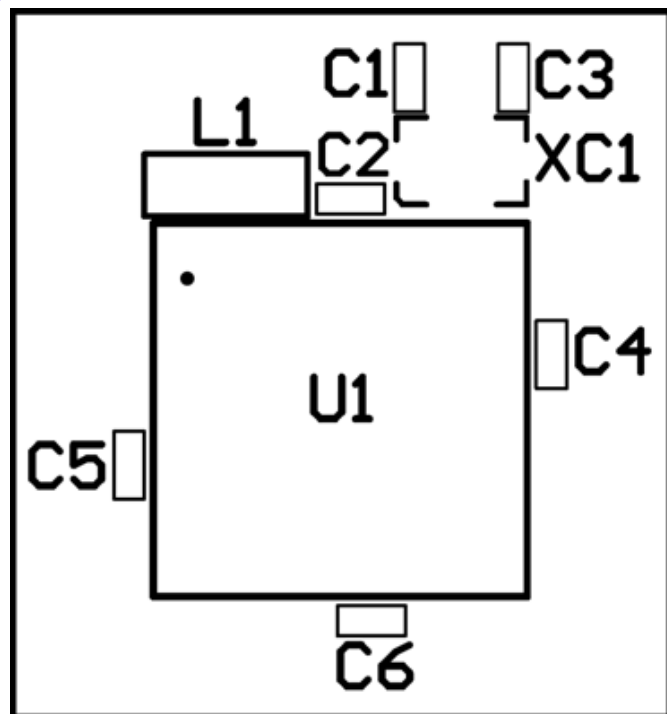
## مدار چاپی - PCB

در پایان، می بایست دقت زیادی برای قرار دادن تشدید کننده ها بر روی PCB داشته باشید. تا آنجایی که امکان دارد می بایست تشدید کننده ها نزدیک به میکروکنترلر بوده و اطراف آن را بوسیله اتصال زمین احاطه کنید.

شکل ۱۱ و ۱۲ شماتیک و pcb نهایی از یک میکروکنترلر ATxmega32A4 را نشان می دهد. در این شماتیک و PCB تمامی موارد ذکر شده جهت حفاظت میکروکنترلر از نویز رعایت گشته است.



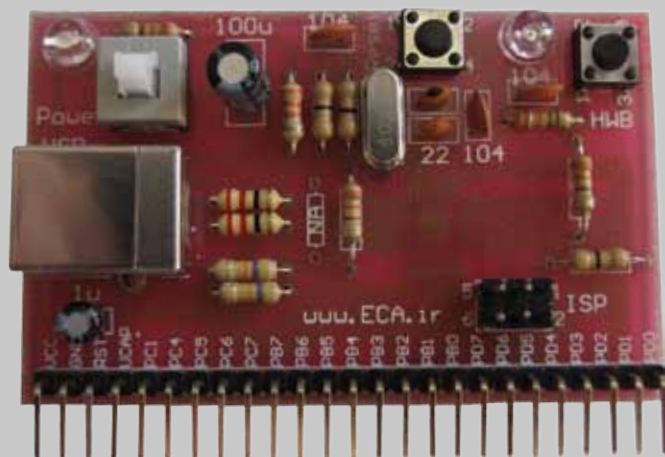
شکل (۱۱)



Atmel AVR042: AVR Hardware Design Considerations

منبع :

## AT90USB162 Header Board



هدربرد (برد راه انداز) میکروکنترلر AT90USB162 با امکانات اولیه جهت راه اندازی این میکروکنترلر. یکی از مزیت های سری AT90USB نسبت به دیگر میکروهای خانواده AVR داشتن یک کانکتور full speed USB 2.0 بر روی میکرو است. همچنین برای این میکرو ها شما نیازی به پروگرامر نداشته و فقط از طریق یک کابل USB و ارتباط برد به رایانه می توانید میکرو را پروگرام نمایید. همچنین تغذیه میکرو نیز از طریق پورت USB تأمین گشته و در پروژه های کوچک نیازی به منبع تغذیه خارجی نخواهید داشت.

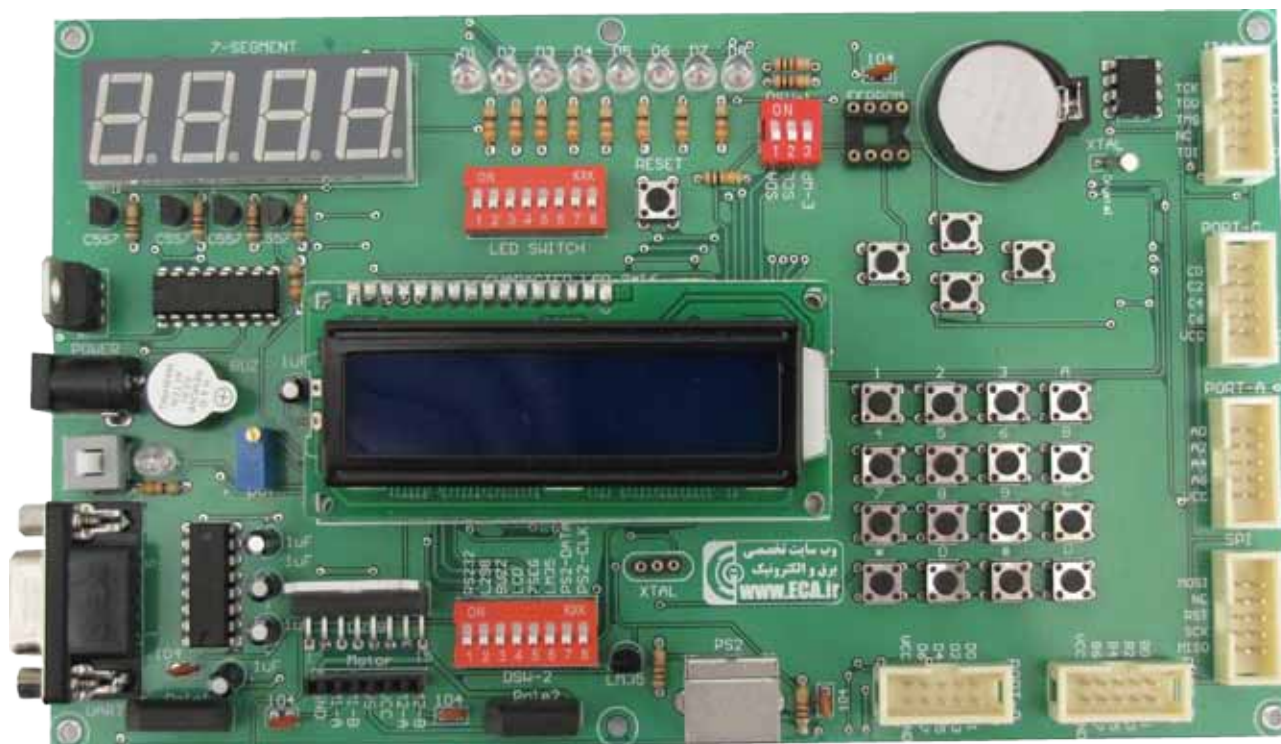
### خلاصه مشخصات برد راه انداز AT90USB162

- \* حداقل مدار جهت راه اندازی میکروکنترلر AT90USB162
- \* قابلیت پروگرام کردن از طریق پورت USB
- \* بدون نیاز به پروگرامر
- \* دارای کانکتور full speed USB 2.0
- \* دارای خروجی تمامی پایه های ورودی خروجی
- \* امکان نصب مستقیم بر روی بردبرد
- \* بدون نیاز به منبع تغذیه خارجی
- \* دارای کلید قطع و وصل تغذیه
- \* امکان استفاده از نرم افزار FLIP
- \* دارای کلید ریست
- \* دارای ۱۲ ماه گارانتی تعویض

توضیحات تکمیلی : <http://eshop.eca.ir/link/852.php>



# AVR Microcontrollers Training Board



## امکانات دستگاه :

۱. میکرو کنترلر ATMEGA 32A
۲. LCD کاراکتری 2x16
۳. درایور موتور L298
۴. پورت سریال RS232
۵. پورت PS2
۶. ۸ عدد LED
۷. 7SEGMENT ۴ تایی
۸. IC ساعت DS1307 و باتری بک آپ
۹. EEPROM
۱۰. بازر
۱۱. صفحه کلید 4x4
۱۲. شستی برای استفاده از اینترنت
۱۳. سنسور دما LM35
۱۴. پروگرامر STK200/300
۱۵. خروجی تمامی پورت ها به صورت BOX 2x5
۱۶. پورت ISP و JTAG جهت پروگرام کردن و دیباگ نمودن میکروکنترلر

توضیحات تکمیلی : <http://eshop.eca.ir/link/748.php>

# ماژول SIM900TE-C



یکی از مشکلاتی که در هنگام کار کردن با SIM900 وجود دارد، نوع پکیج آن بوده که به راحتی نمی توان از آن در مدارات مختلف استفاده نمود. در این ماژول تمامی قطعات اولیه جهت درایو SIM900a بر روی خود ماژول قرار داده شده است و پایه های SIM900a را بصورت پین به پین توسط یک پین هدر به بیرون منتقل کرده است. همچنین جهت سهولت کارکرد ماژول، آنتن بصورت یک کانکتور مادگی کوچک بر روی ماژول قرار داده شده و در صورت نیاز می توانید از پشت برد نسبت به لحیم کاری آن اقدام نمایید.

SIM900C/SIM900-TE-C is pin to pin compatible with SIM300C and SIM340C.

We offer our client an excellent quality range of SIM900-TE-C GSM/GPRS Module. SIM900-TE-C GSM/GPRS Module are widely known for its durability and quality.

## General features

- \* Quad-Band 850/ 900/ 1800/ 1900 MHz
- \* GPRS multi-slot class 10/8
- \* GPRS mobile station class B
- \* Compliant to GSM phase 2/2+
  - Class 4 (2 W @850/ 900 MHz)
  - Class 1 (1 W @ 1800/1900MHz)
- \* Control via AT commands (GSM 07.07 ,07.05 and SIMCOM enhanced AT Commands)
- \* SIM application toolkit
- \* Supply voltage range : 3.1 ... 4.8V
- \* Low power consumption: 1.5mA(sleep mode)
- \* Operation temperature: -40°C to +85 °C

## Specifications for Fax

- \* Group 3, class 1

## Specifications for Data

- \* GPRS class 10: max. 85.6 kbps (downlink)
- \* PBCCH support
- \* Coding schemes CS 1, 2, 3, 4
- \* CSD up to 14.4 kbps
- \* USSD
- \* Non transparent mode
- \* PPP-stack

## Specifications for SMS via GSM/GPRS

- \* Point to point MO and MT
- \* SMS cell broadcast
- \* Text and PDU mode

## Software features

- \* 0710 MUX protocol
- \* embedded TCP/UDP protocol

- \* FTP/HTTP( available at late July ,2010)
- \* FOTA ( available at July ,2010)
- \* MMS (available at July ,2010)
- \* Embedded AT (available at Q3,2010)

## Specifications for Voice

- \* Tricodec
  - Half rate (HR)
  - Full rate (FR)
  - Enhanced Full rate (EFR)
- \* Hands-free operation (Echo suppression)
- \* AMR
  - Half rate (HR)
  - Full rate (FR)

## Interfaces

- \* Interface to external SIM 3V/ 1.8V
- \* analog audio interface
- \* RTC backup
- \* SPI interface (option)
- \* Serial interface
- \* Antenna pad
- \* I2C
- \* GPIO
- \* PWM
- \* ADC

## Compatibility

- \* AT cellular command interface

## Certificates:

- \* CE
- \* FCC
- \* ROHS
- \* PTCRB
- \* GCF
- \* IC
- \* ICASA
- \* TA
- \* REACH \* AT&T

توضیحات تکمیلی : <http://eshop.eca.ir/link/SIM900TE-C.php>

## JIMBO LCD



Size : 2.8 inch  
Resulation : 240\*320  
Driver : ili9325



Size : 3.2 inch  
Resulation : 240\*320  
Driver : SSD1289

LCD های گرافیکی همواره دارای جذابیت زیادی بوده و تنوع در سایز و کیفیت باعث گسترش کاربرد این LCD ها شده است. مشکل عمده ای که در استفاده از این LCD ها وجود دارد، پیچیده بودن استفاده از این LCD ها توسط میکروکنترلرهای مختلف است. از این رو وب سایت ECA بر آن شد تا با طراحی یک نرم افزار جامع مشکل برنامه نویسی انواع LCD های گرافیکی را حل نموده و از طرفی به موازات این طرح اقدام به تولید و وارد نمودن ماژول های مناسبی برای این LCD ها نموده است.

نرم افزار JIMBO LCD نرم افزاری یکپارچه و بر مبنای ماژول های مورد تایید ECA طراحی شده است و قصد داریم که به مرور، متناسب با بازخوردی که کاربران این نرم افزار ارسال خواهند کردند و همچنین با ساخت و معرفی ماژول LCD های جدید، سعی در تکمیل هرچه بیشتر نرم افزار داشته باشیم.

### معرفی بخش های نرم افزار

همانگونه که در شکل صفحه بعد مشاهده خواهید نمود، برنامه از ۴ بخش اصلی تشکیل یافته که به طور خلاصه معرفی خواهند شد.

#### بخش شماره ۱

منوی اصلی برنامه شامل گزینه های زیر می باشد:

**منوی File**، شامل گزینه های :

**Open Sample** : جهت استفاده از نمونه برنامه های از پیش نوشته شده

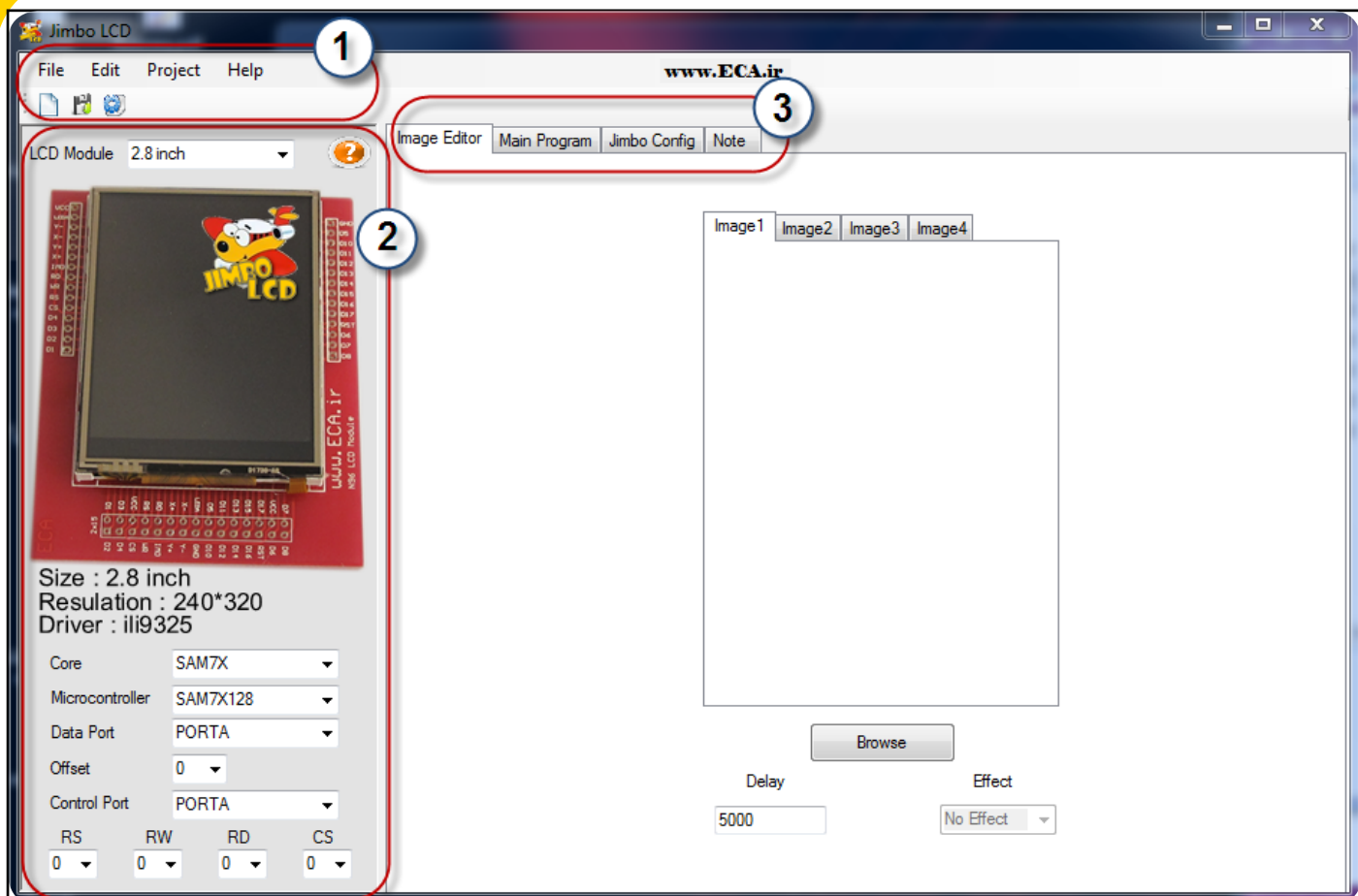
**Page Setup** : جهت تنظیم نمودن تنظیمات صفحه برای پرینت گرفتن

**Print** : پرینت گرفتن از صفحات نرم افزار

**Exit** : خروج از نرم افزار

**منوی Edit** شامل گزینه های Font, Undo, Redo, Select All, Copy, Cut که به دلیل عمومی بودن این امکانات توضیحات اضافی داده نمی شود.





**منوی Project** شامل گزینه های :

New Project : جهت ایجاد پروژه جدید

Open Project : برای باز نمودن پروژه های ذخیره شده

Save Project : ذخیره نمودن تنظیمات و کدهای ایجاد شده جهت استفاده در کامپایلر

Generate : جهت تولید کدهای مورد نیاز میکروکنترلر برای نمایش عکس ها

Clean Up : برای پاک نمودن کدهای تولید شده

**منوی Help**، جهت استفاده از راهنمای نرم افزار و موارد مرتبط

## بخش شماره ۲

این بخش در واقع قسمت Wizzard برنامه بوده و متناسب با تنظیمات این بخش کد نهایی برای شما ساخته خواهد شد.

**LCD Module** : انتخاب ماژول LCD مورد نظر. (با انتخاب علامت سوال موجود در جلوی این بخش، میتوانید توضیحات کاملتری در مورد انواع ماژول های موجود مطالعه نمایید.)

**Core** : انتخاب هسته و خوانواده میکروکنترلر

**MicroController** : انتخاب میکروکنترلر

**Data Prot** : توسط این قسمت می توانید یکی از پورت های میکروکنترلر را جهت تبادل اطلاعات با LCD انتخاب نمایید. این پورت می بایست حداقل دارای ۱۶ پایه دیتا بوده و بصورت متوالی به پایه های LCD متصل شوند. به عنوان مثال بین های P0.0 تا P0.15 میکروکنترلر باید بصورت متوالی به پایه های D0 تا D15 ماژول LCD متصل شوند.

**Offset** : توسط این گزینه شما می توانید پایه های دیتای انتخاب شده میکروکنترلر را به تعداد دلخواه شیفت دهید. به عنوان مثال در صورتی که شما پورت A را جهت دیتای LCD انتخاب کرده باشید، در صورتی که Offset را عدد ۲ انتخاب نمایید، پایه های دیتا از بین P0.2 میکروکنترلر آغاز می گردد.

**Control Prot** : در این قسمت پورتی از میکروکنترلر را جهت اتصال به بین های کنترلی LCD مشخص می نمایم.  
RS - RW - RD - CS : در این قسمت بین مورد نظر از میکروکنترلر را می توانید برای پایه های کنترلی LCD انتخاب نمایید.

نکات قابل توجه:

- مقدار Offset را می بایست طوری انتخاب نمایید که بعد از پایه مورد نظر ۱۶ عدد بین دیتا برای LCD در دسترس باشد.
- جهت انتخاب پایه های کنترلی LCD دقت لازم را داشته باشید تا با پایه های دیتای LCD تداخل نداشته باشند.



## بخش شماره ۳

تب های مربوط به انتخاب عکس، افکت و کد تولید شده در این بخش نمایش داده میشوند.  
 تب Image Editor: انتخاب عکس ها، انتخاب افکت برای تغییر عکس ها به هنگام تغییر تصاویر، میزان تاخیر بین ۲ نمایش عکس از طریق این بخش قابل تنظیم هستند.  
 تب Main Program: پس از Generate شدن برنامه، کدهای بخش main در این قسمت نمایش داده شده و قابل ویرایش خواهد بود.  
 تب Jimbo Config: تمامی تنظیماتی که در بخش ۲ (مراجعه به صفحه قبل) اعمال می گردد، پس از Generate در این قسمت نمایش داده شده و قابل ویرایش می باشد.  
 تب Note: جهت نوشتن توضیحات و نکات اضافی برای پروژه در نظر گرفته شده است.

در این قسمت به عنوان مثال می خواهیم یک پروژه با نرم افزار JIMBOLCD ایجاد نماییم.  
**تنظیمات مربوط به نرم افزار Jimbo LCD**

- ۱- از منوی File یک پروژه جدید ایجاد نمایید.
- ۲- در قسمت Wizzard، مشخصات LCD و میکروکنترلر را تنظیم نمایید.
- ۳- فایل عکس را انتخاب نمایید.
- ۴- با استفاده از گزینه Generate، فایل پروژه ساخته خواهد شد.
- ۵- فایل را از طریق گزینه Save Project ذخیره نمایید

**تنظیمات مربوط به نرم افزار Keil**

- ۱- یک پروژه جدید در مسیر فایل ذخیره شده Jimbo LCD ایجاد نمایید.
- ۲- میکروکنترلر مورد نظر (یکسان با مورد انتخابی در پروژه Jimbo) ایجاد نمایید تا فایل StartUp به پروژه اضافه گردد.
- ۳- اضافه نمودن Jimbo.Lib و main.c به پروژه از مسیر زیر :

Project ==> Manage ==> Components,... ==> Add Files

۴- اضافه نمودن فایل system\_LPC17XX.c به پروژه (این مرحله فقط برای خانواده LPC17XX می باشد)

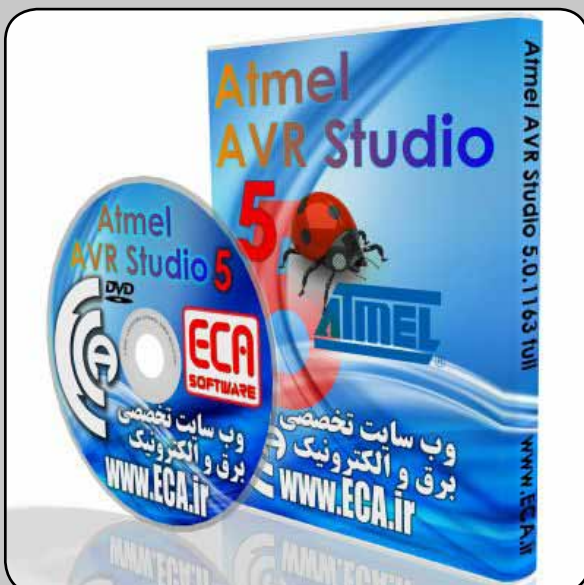
Add Form Keil\ARM\Startup\NXP\LPC17xx\

۵- Build کردن پروژه و پروگرام نمودن میکرو

در نهایت به این نکته تاکید می شود که این نرم افزار فعلا نسخه آزمایشی (Beta) می باشد و سعی داریم به سرعت میکروکنترلرهای تحت پشتیبانی و همچنین امکانات نرم افزار را افزایش دهیم و در این راه نیازمند راهنمایی های استفاده کنندگان محترم هستیم.

صفحه مربوط به نرم افزار : [http://eshop.eca.ir/link/jmbo\\_lcd.php](http://eshop.eca.ir/link/jmbo_lcd.php)

## Atmel AVR Studio 5.0.1163 full



برای کار با میکروکنترلرهای AVR کامپایلرهای متفاوتی به زبانهای سی، اسمبلی، بیسیک و پاسکال وجود دارد. یکی از این کامپایلرها، کامپایلر AVR Studio می باشد که متعلق به شرکت Atmel تولید کننده های میکروکنترلرهای AVR است. این کامپایلر که از زبانهای سی و اسمبلی پشتیبانی می نماید به دلیل قدرت در برنامه نویسی، مجانی بودن و مهمتر از همه کامپایلر مرجع بودن برای میکروکنترلرهای AVR همواره جزو برترین کامپایلرها بوده است.

شرکت اتمل در تاریخ ۱ مارس ۲۰۱۱ به طور رسمی و بعد از عرضه دو نسخه بتا، نسخه رسمی شماره ۵ این کامپایلر را عرضه کرد. در این نسخه تغییرات گوناگونی صورت گرفته است.

توضیحات تکمیلی : <http://eshop.eca.ir/link/956.php>

## ماژول "3.2 LCD به همراه تاج اسکرین



ماژول فوق دارای یک LCD ۳.۲ اینچی به همراه تاج اسکرین بوده و قابلیت راه اندازی با میکروهای سری Xmega و ARM را دارا می باشد. با توجه به رزولیشن بالا و وضوح تصویر مناسب از این ماژول می توان در امور آموزشی و صنعتی استفاده نمود. از ویژگی های مهم این ماژول قابلیت راه اندازی در ۲ مد ۸ بیتی و ۱۶ بیتی از طریق مدار مبدل موجود بر روی برد می باشد. یک عدد سوکت MMC نیز جهت امکان اتصال کارت های حافظه به مدار قرار داده شده است. چیپ کنترلر این ماژول SSD1289 می باشد.

### Parameters:

**Resolution: QVGA 240 X 320**

**Size: 3.2"**

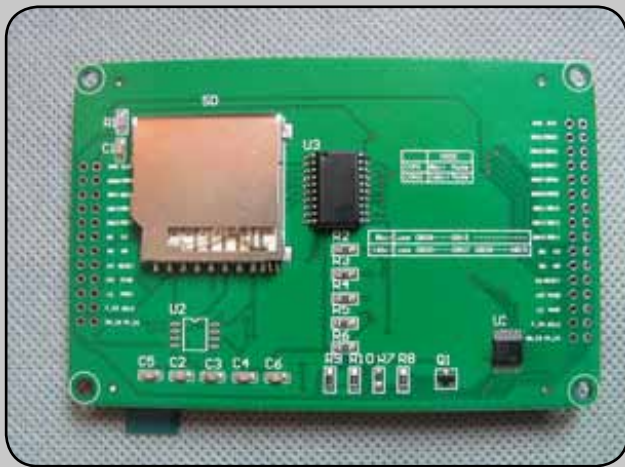
**Controller: SSD1289**

**Touch Screen: 4-wire resistive**

**Pin: 37PIN pitch 1mm**

**Backlight: 5 LED parallel**

توضیحات تکمیلی : <http://eshop.eca.ir/link/897.php>



## ماژول LCD N96 به همراه تاج اسکرین



این ماژول جهت کارکرد راحتتر با LCD N96 ساخته شده و تمام پایه های مورد استفاده این LCD و همچنین پایه های تاج اسکرین بیرون کشیده شده است.

ماژول فوق قابلیت اتصال بر روی برد را داشته و همچنین می توان بر روی برد های دیگر نصب و استفاده شود.

LCD گوشی N96 چینی یا ELT240320 یک نمایشگر رنگی ۵\*۷ سانتی متری (با اندازه تصویر ۳۲۰ \* ۲۴۰ پیکسل) می باشد که توسط آن می توانید تصاویر و متون رنگی خود را به نمایش در آورید.

توضیحات تکمیلی : <http://eshop.eca.ir/link/619.php>



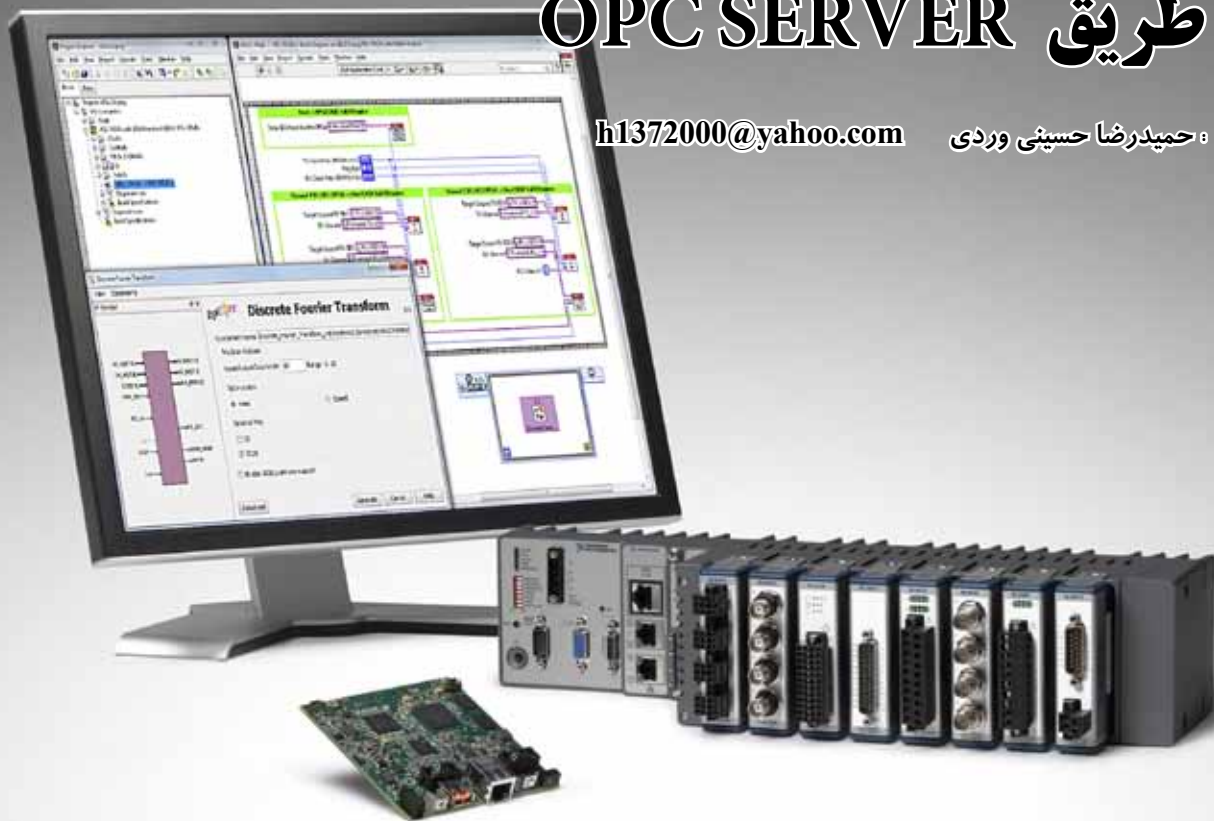
# NI LabVIEW

DESIGN , ANALYSIS AND SIMULATION

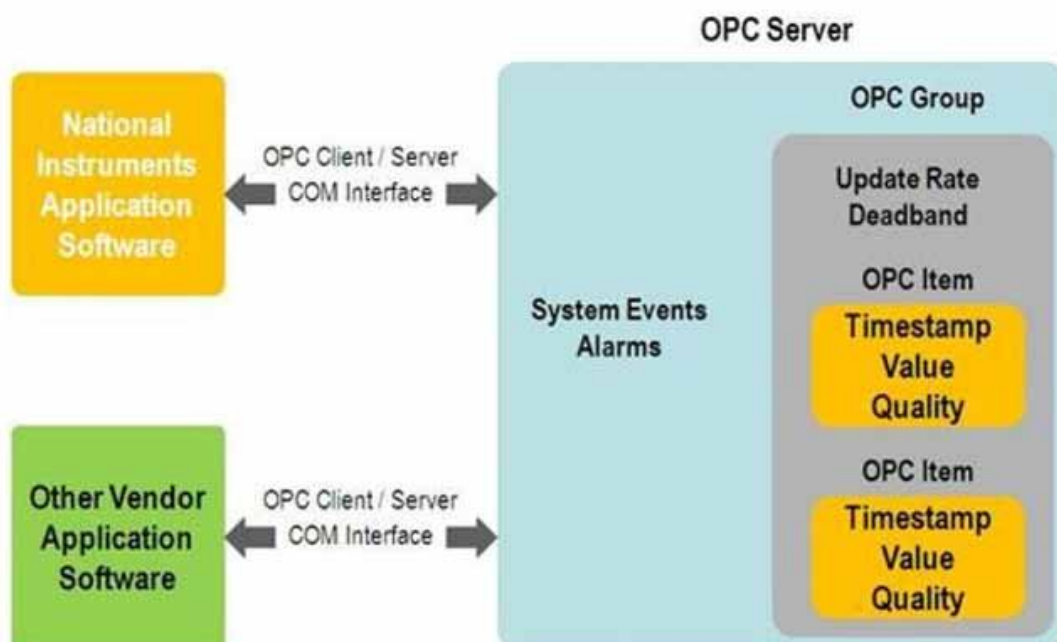
## ارتباط LabVIEW با انواع PLC از طریق OPC SERVER

h1372000@yahoo.com

نویسنده : حمیدرضا حسینی وردی



نرم افزار LabView توانایی ارتباط با هر گونه plc را از راه های گوناگون دارد. OPC یک راه استاندارد برای برقراری ارتباط بین دستگاه های کنترلی و HMI ها می باشد. OPC SERVER ها برای تمامی PLC ها و PAC ها موجود می باشند. در این تمرین با نحوه ارتباط بین PLC شرکت زیمنس و نرم افزار Labview از طریق OPC و ارتباط کابلی MPI آشنا خواهید شد.





# ارتباط LabVIEW با انواع PLC از طریق OPC SERVER

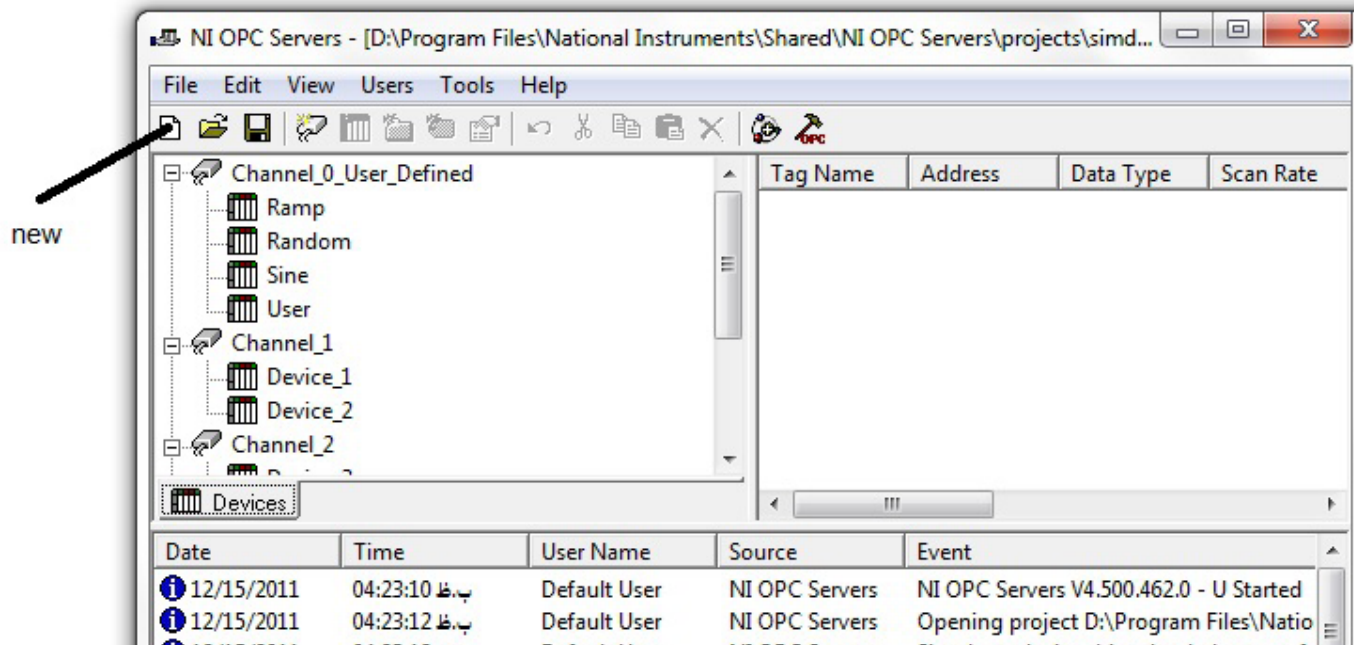
## برقراری ارتباط بین LabView و Siemens PLC

در این تمرین قصد داریم یک سیستم اسکادا شامل Siemens S7300 PLC و یک OPC Client بر پایه LabView ایجاد نماییم. فرآیند ایجاد سیستم اسکادا را به صورت زیر مرحله به مرحله انجام می دهیم.

۱ - ایجاد یک OPC SERVER از طریق نرم افزار NI OPC SERVERS :

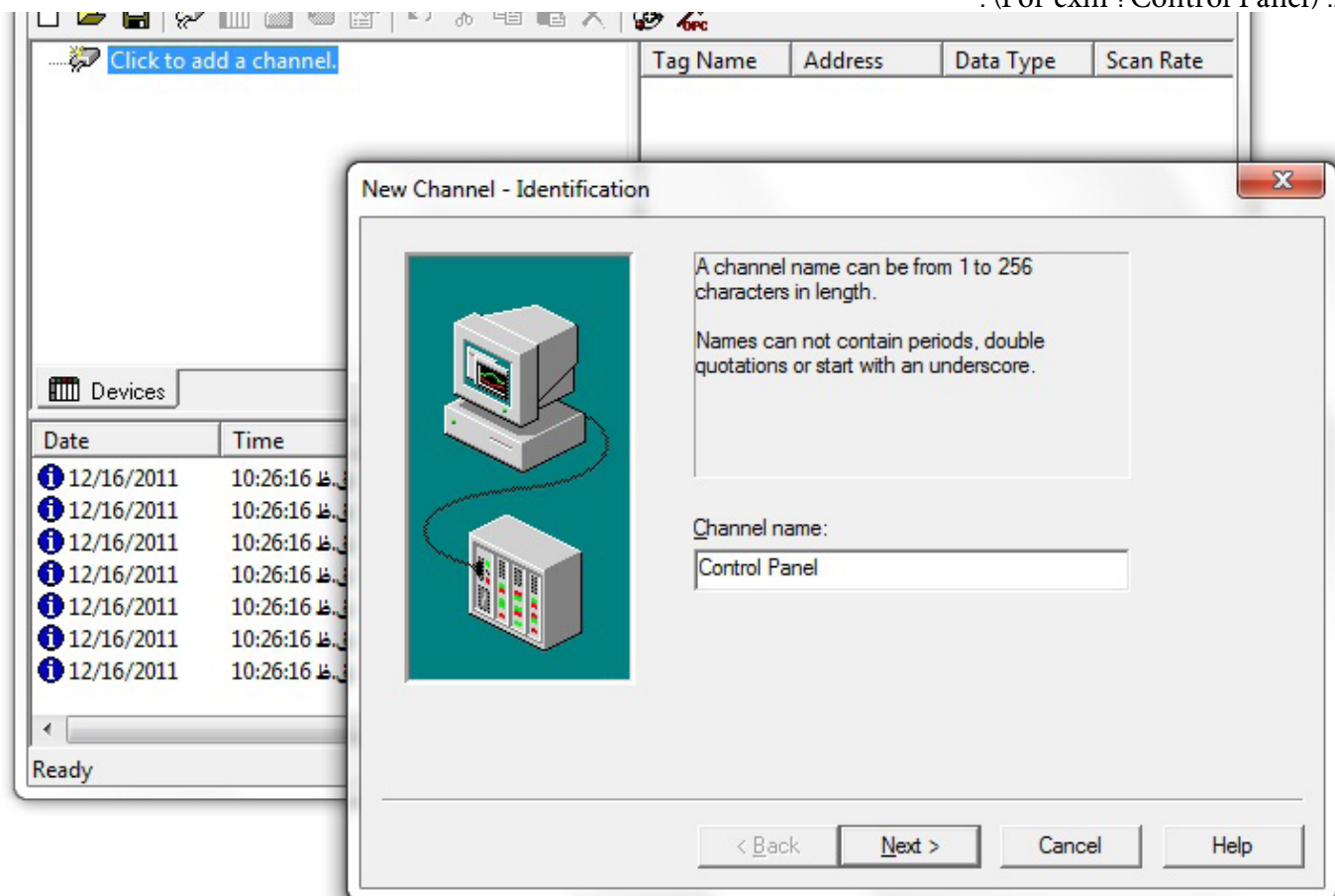
Ni opc server را از آدرس زیر باز کنید و یک صفحه جدید را با زدن دکمه new ایجاد کنید :

- Start\Program\National instruments\Ni opc servers\Ni opc servers



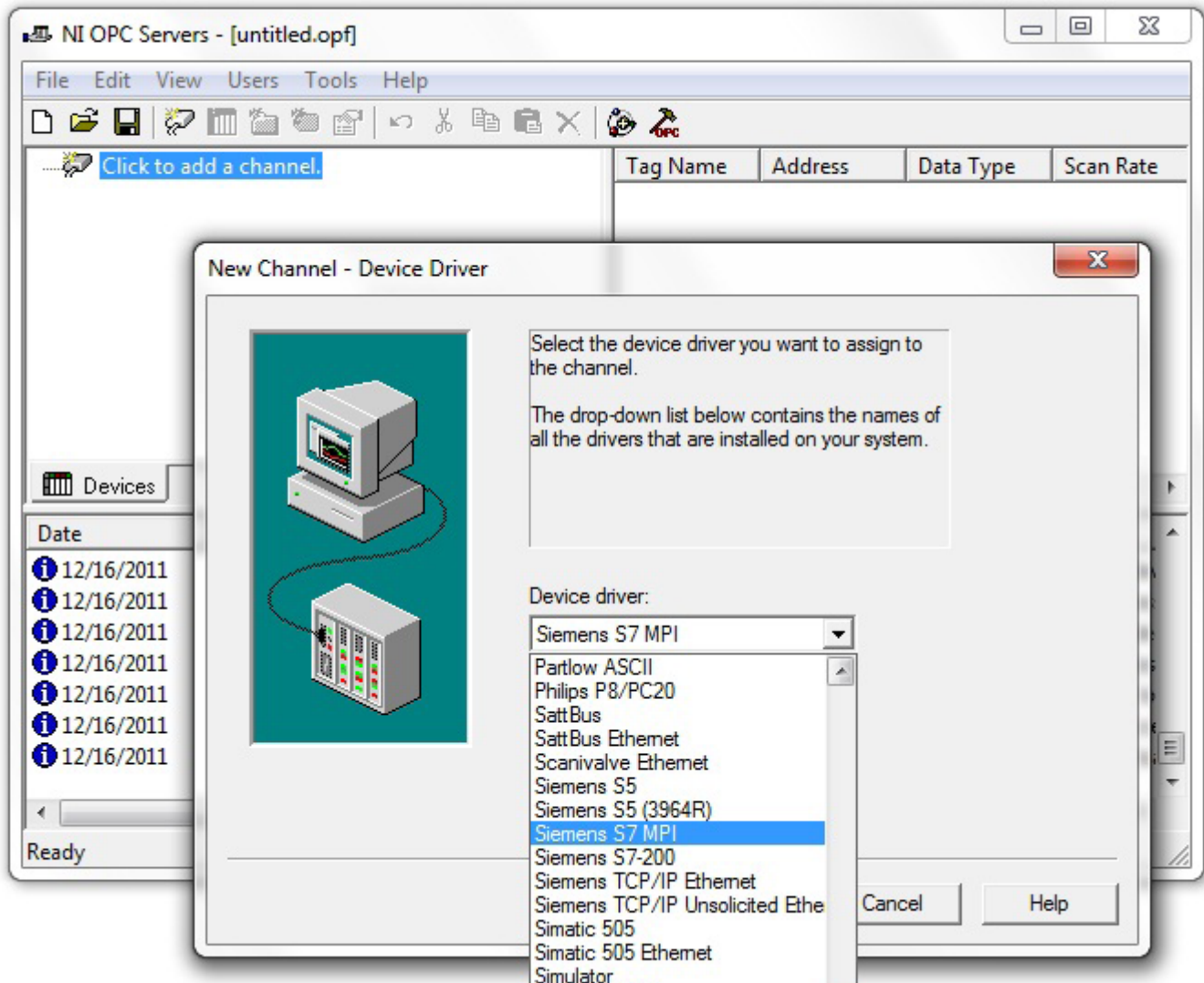
اولین مرحله تعیین نوع کانال ارتباطی بین plc و labview می باشد. هر دستگاه کانال ارتباطی مشخصی را برای خود دارد و لزوم تمامی دستگاه ها دارای کانال ارتباطی یکسان نیستند.

بر روی گزینه Click to add a channel کلیک کنید. در پنجره باز شده و در قسمت Channel name یک نام برای کانال خود انتخاب کنید. (For exm : Control Panel).

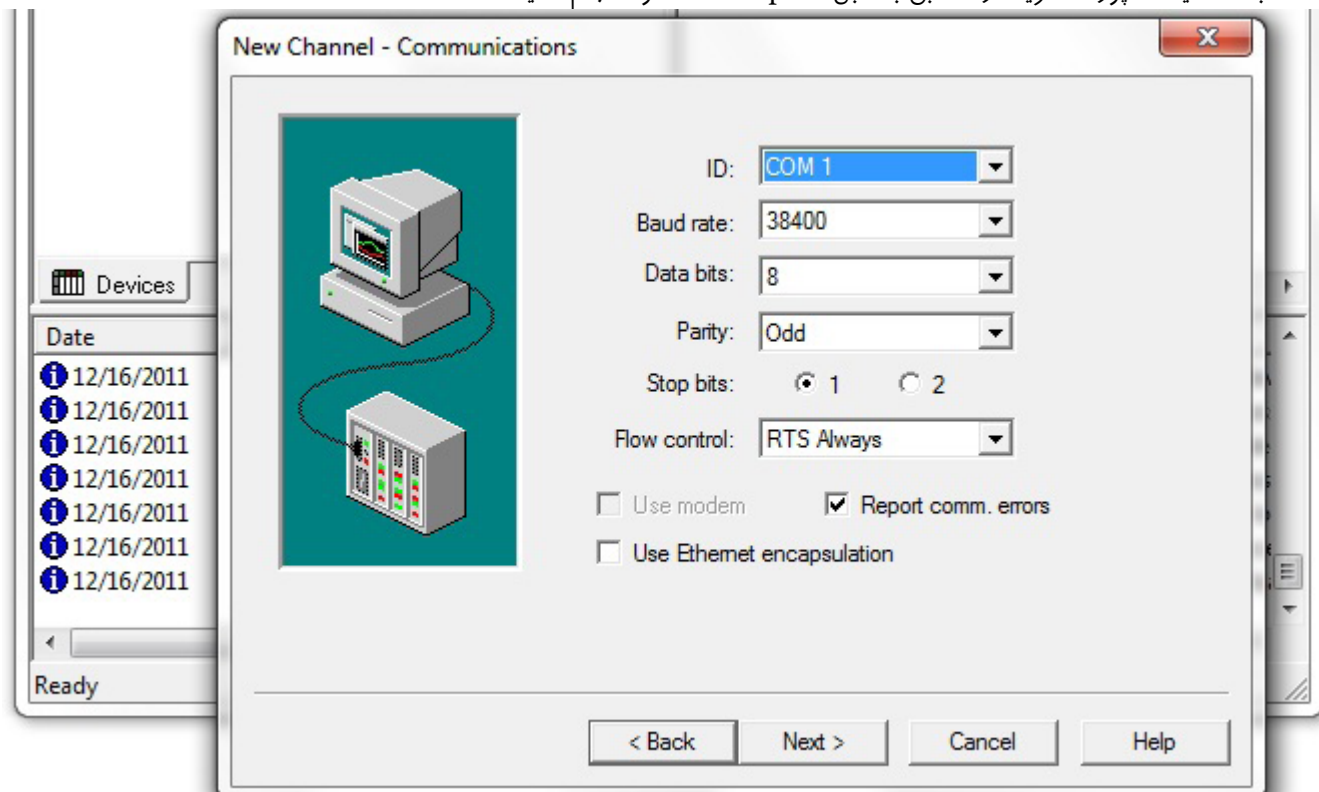




در صفحه باز شده بعدی و از لیست Device Drivers ها گزینه Siemens S7 MPI را انتخاب نمایید.



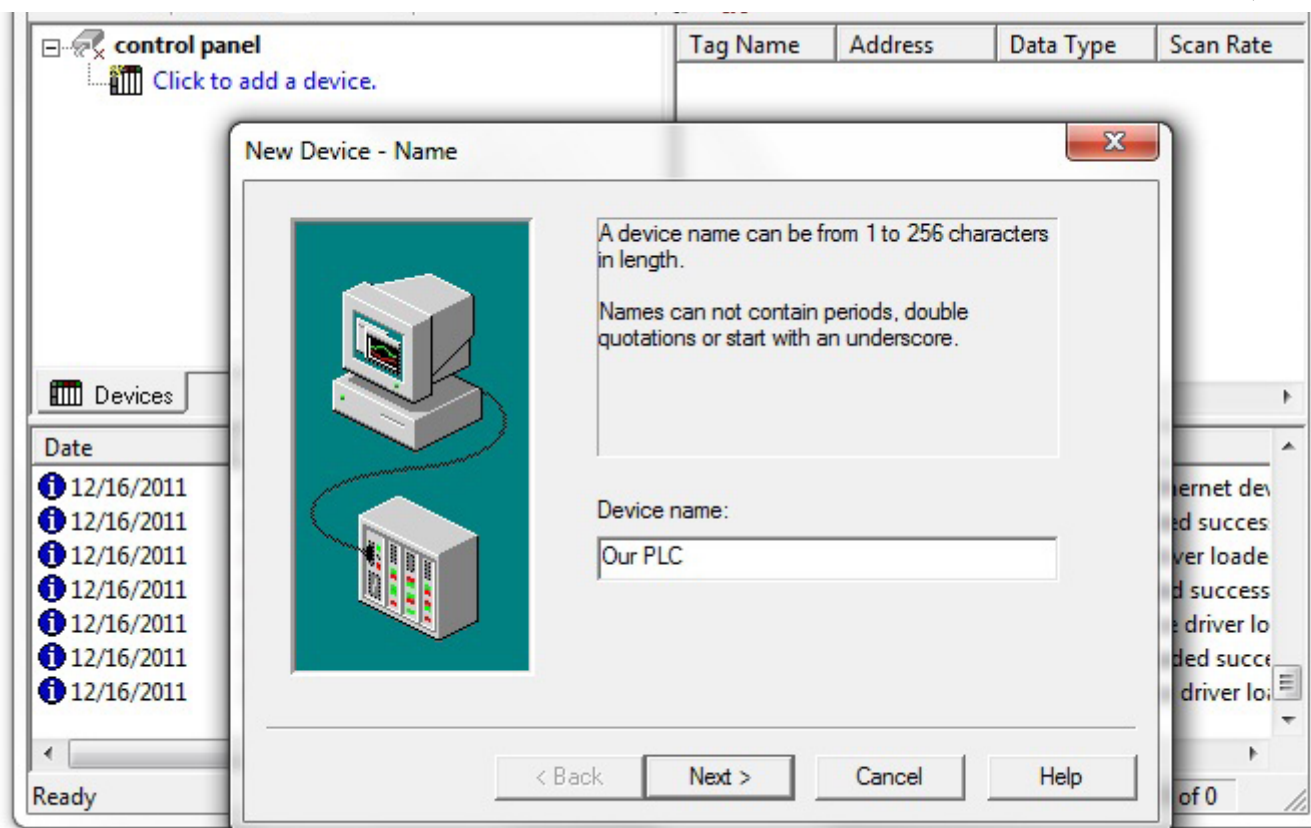
در صفحه بعد تنظیمات پورت سریال را مطابق با کابل PC adaptor خود انجام دهید.



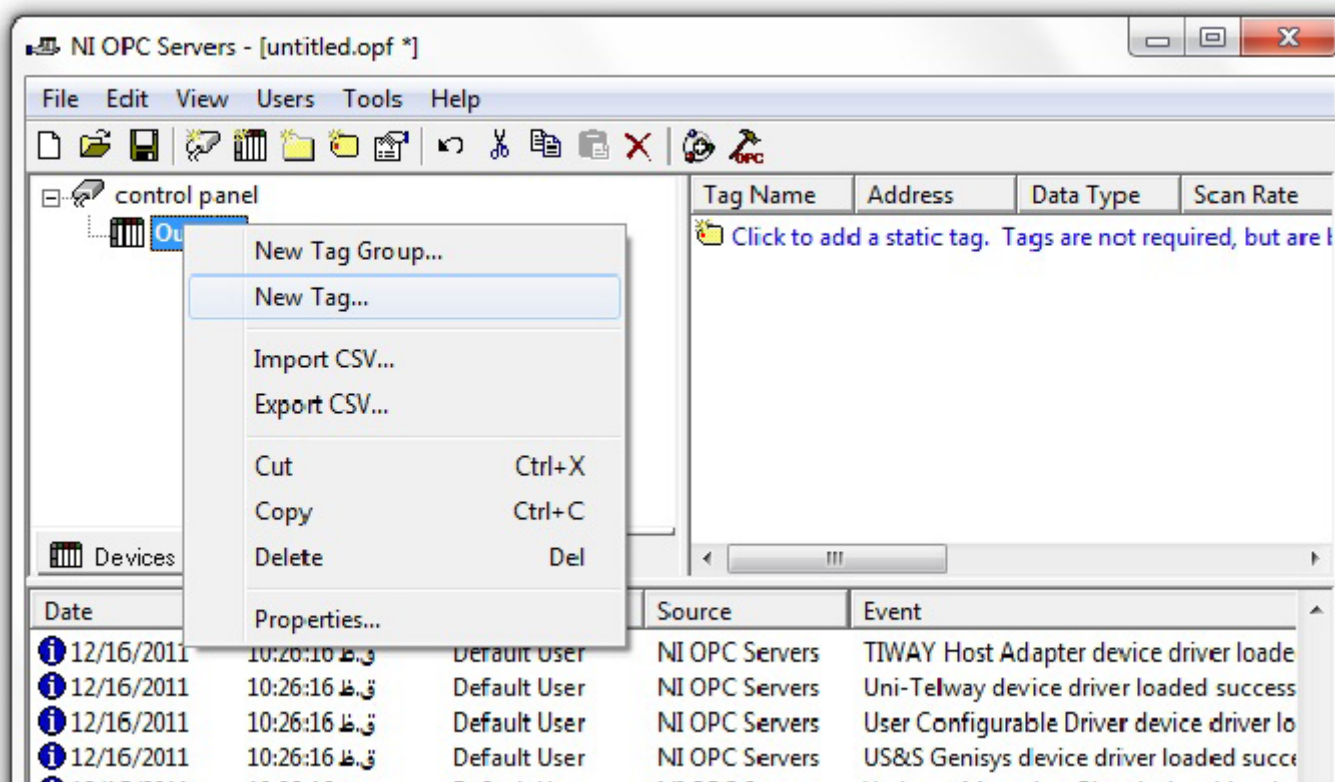
# ارتباط LabVIEW با انواع PLC از طریق OPC SERVER

صفحات بعدی را بدون تغییر در پارامترها تأیید نموده و در نهایت بر روی گزینه finish کلیک نمایید.  
در این مرحله باید یک دستگاه به سیستم اضافه نماییم.

بر روی گزینه Click to add a device کلیک کنید و در قسمت Device name یک نام برای دستگاه خود انتخاب کنید. (For exm : Our PLC)

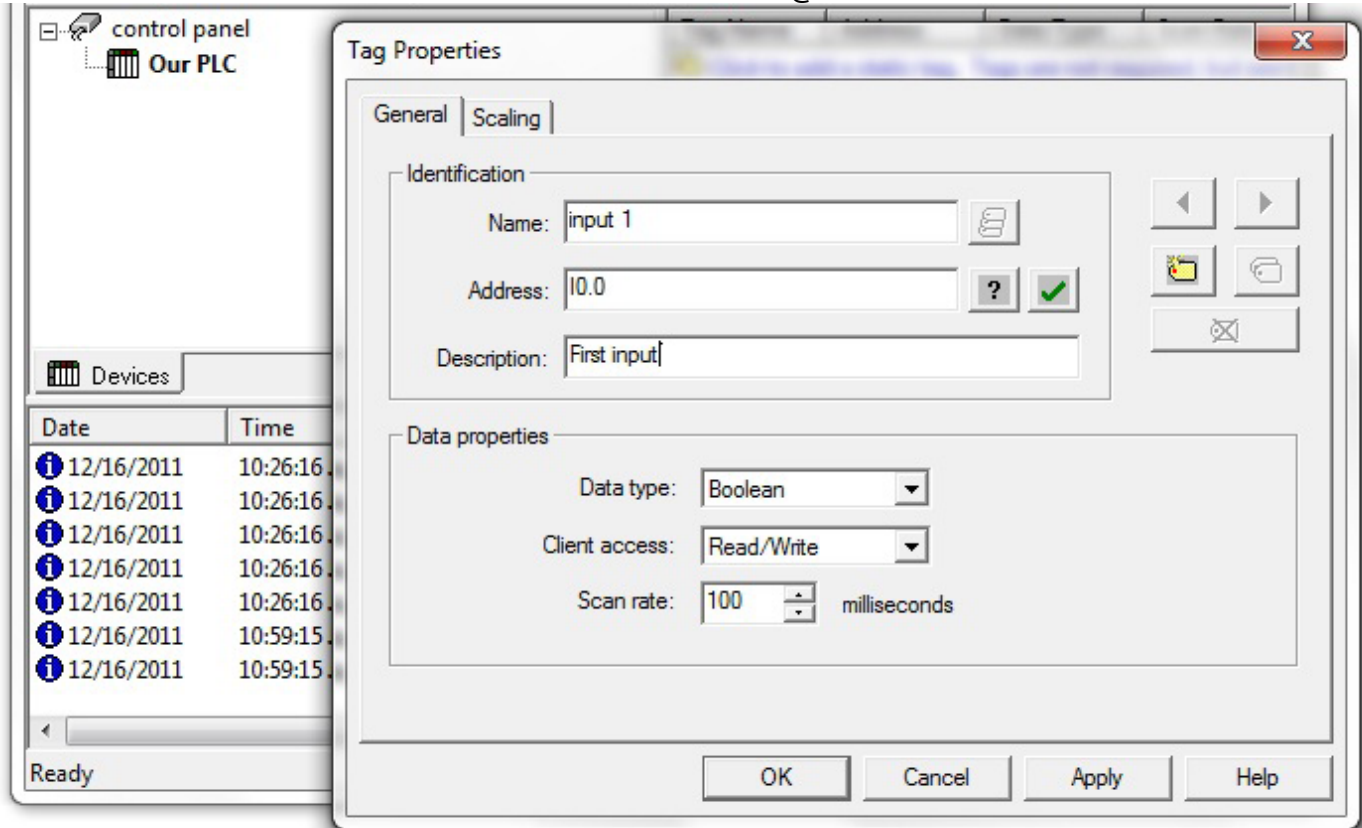


صفحات بعدی را بدون تغییر در پارامترها تأیید نموده و در نهایت بر روی گزینه Finish کلیک کنید.  
پس از انجام مراحل بالا مشاهده می کنیم که یک کانال با یک دستگاه در صفحه opc server ایجاد شده است.  
پس از معرفی دستگاه باید تگ ها را برای سیستم تعیین کنیم.  
برای ایجاد تگ ها بر روی دستگاه در پنجره سمت چپ محیط OPC کلیک راست نموده و گزینه New Tag را انتخاب نمایید.



در معرفی تگ ها باید استاندارد برنامه نویسی دستگاه را رعایت نماییم. برای مثال زمانی که دستگاه مورد نظر plc زمینس باشد باید تمامی تگ ها را مطابق با ورودی ها، خروجی ها، و حافظه بکار گرفته شده در برنامه plc تعیین کنیم. پنجره Tag Properties دارای یک زیر شاخه با نام General که شامل معرفی تگ (Identification) و تعیین نوع دیتا (Data properties) می باشد و یک زیر شاخه با نام Scaling که برای اسکیل بندی داده ها مطابق با نیاز شما می باشد.

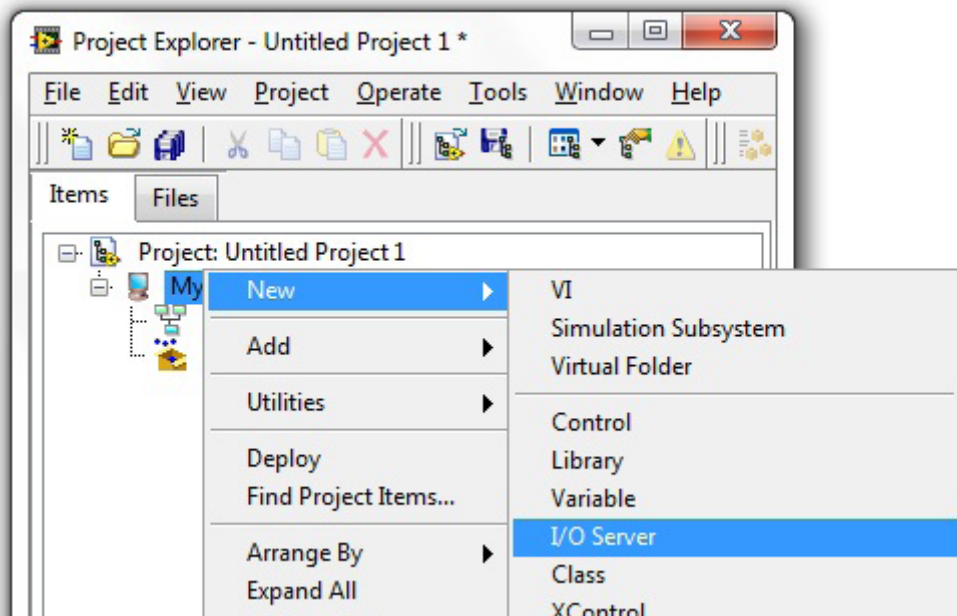
در پنجره Tag Properties و در قسمت Identification ابتدا یک نام برای تگ درج کنید (ex : input1)، سپس نوع تگ را با توجه به نوع داده مشخص کنید (EX : IO.0) و در آخر می توانید توضیحاتی برای این تگ درج کنید. دقت نمایید که پس از درج تگ در قسمت Address، با زدن علامت تیک سبز رنگ از صحت تگ درج شده اطمینان حاصل نمایید.



تا این مرحله کار معرفی تگ ها به پایان رسید. بعد از این نوبت به ارتباط این تگ ها و LabView می باشد. برای ارتباط بین تگ ها ی ایجاد شده و LabView باید متغیرهای اشتراکی در LabView ایجاد کرده و تگ ها را به آن نسبت دهیم. مراحل زیر را برای معرفی تگ ها به LabView انجام دهید :

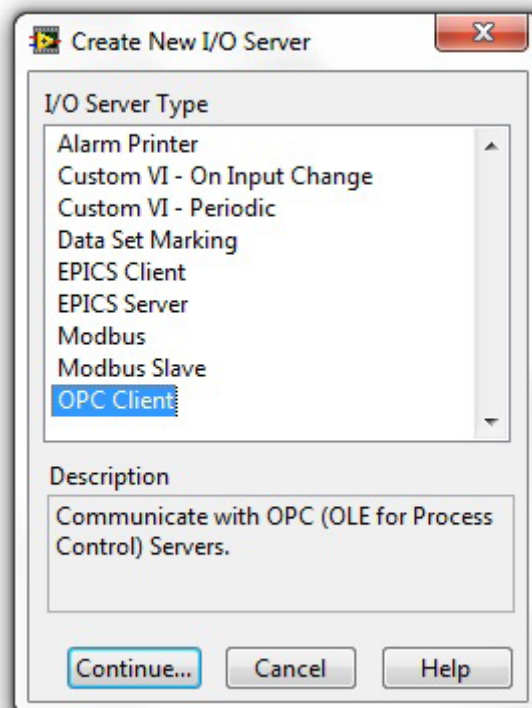
ابتدا در پنجره Getting start LabView یک new Projects ایجاد می کنیم.

در پنجره Project روی گزینه My computer کلیک راست کرده و از شاخه new گزینه I/O server را انتخاب می کنیم.

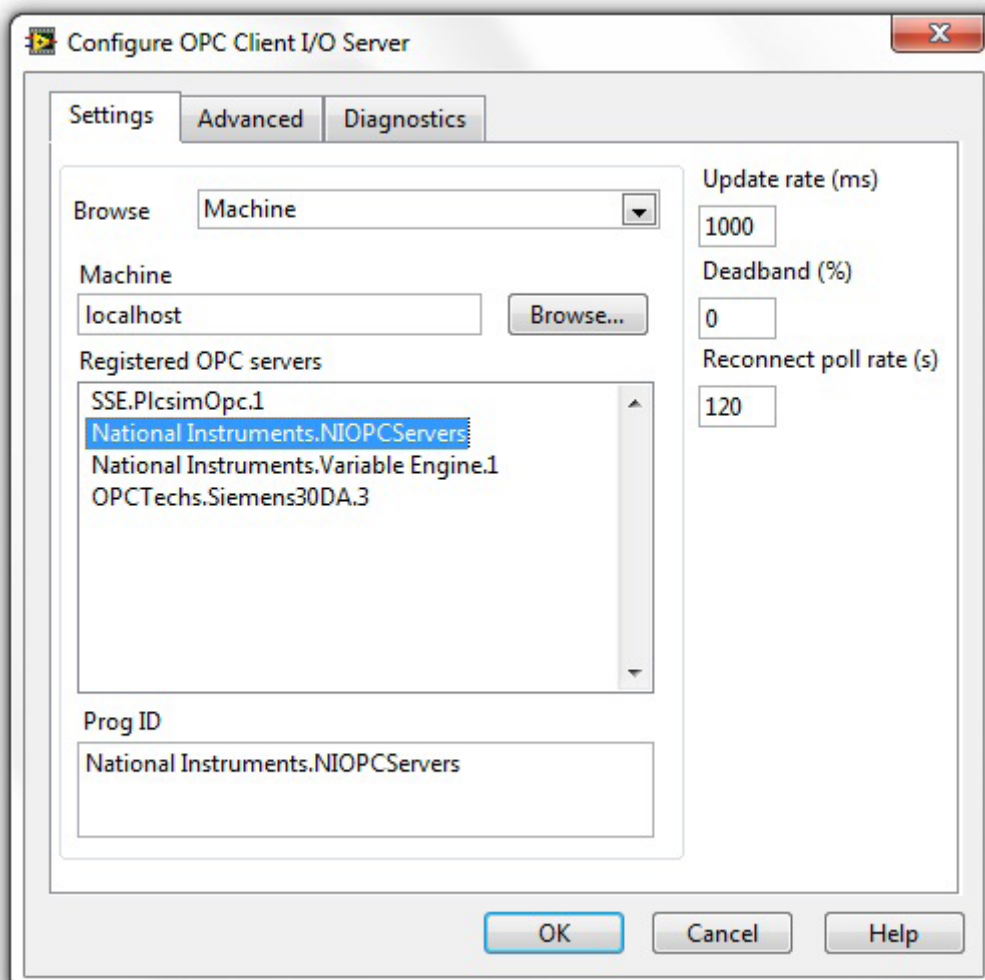




در پنجره باز شده گزینه opc client را انتخاب نمایید.

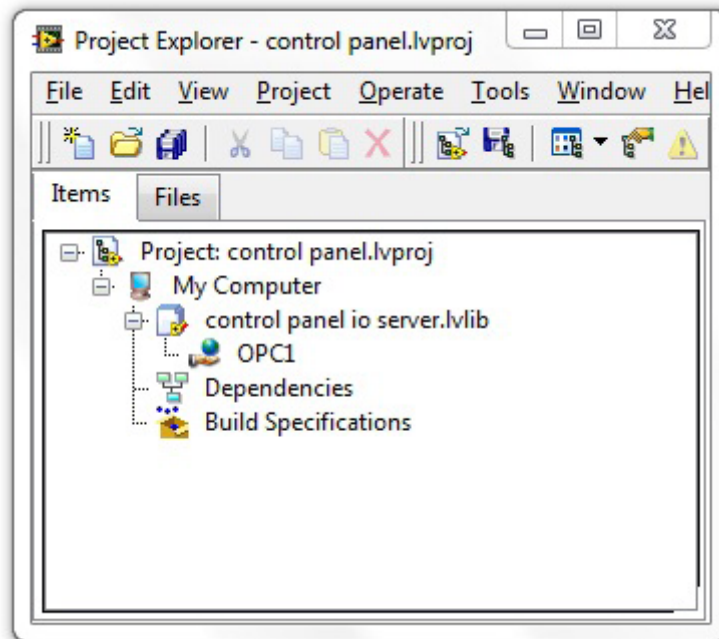


پس از انتخاب گزینه opc client پنجره ای باز می شود که تمامی opc server هایی که در کامپیوتر شما نصب است را نمایش می دهد. اگر LabVIEW و تولکیت های مربوطه به درستی در سیستم شما نصب شده باشند باید شاهد گزینه national instrument.NIOpcservers در پنجره Registered opc server باشید. گزینه national instrument.NIOpcservers را انتخاب کنید.

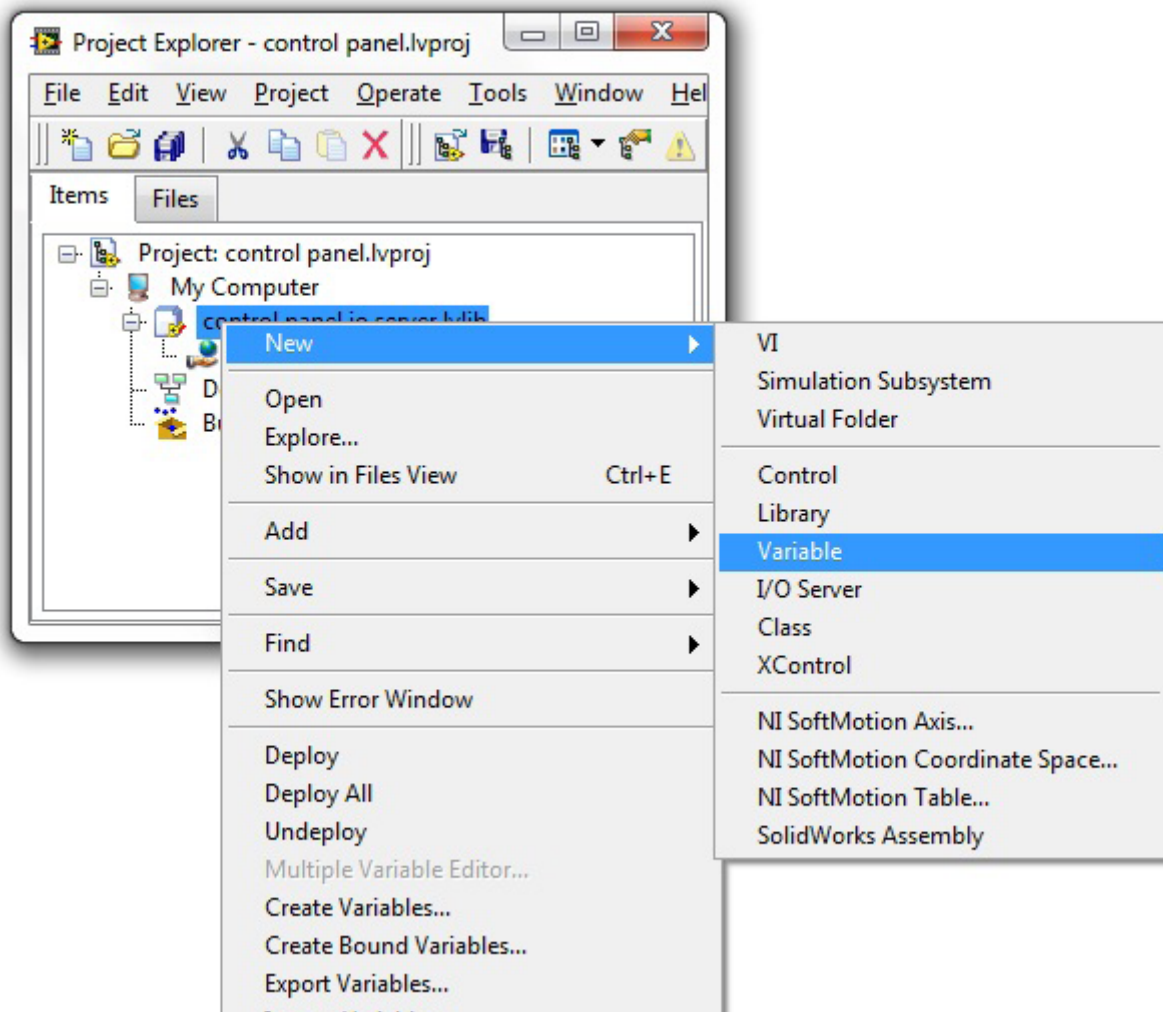




حال مشاهده می کنیم که یک زیر شاخه با عنوان Untitled Library<sup>۱</sup> به شاخه my computer اضافه شده است که دارای یک آیتم به نام OPC1 می باشد.  
از شاخه file گزینه save all را انتخاب نموده و در پنجره اول نام control panel و در پنجره بعدی نام control panel io server را برای ذخیره پروژه وارد نمایید .

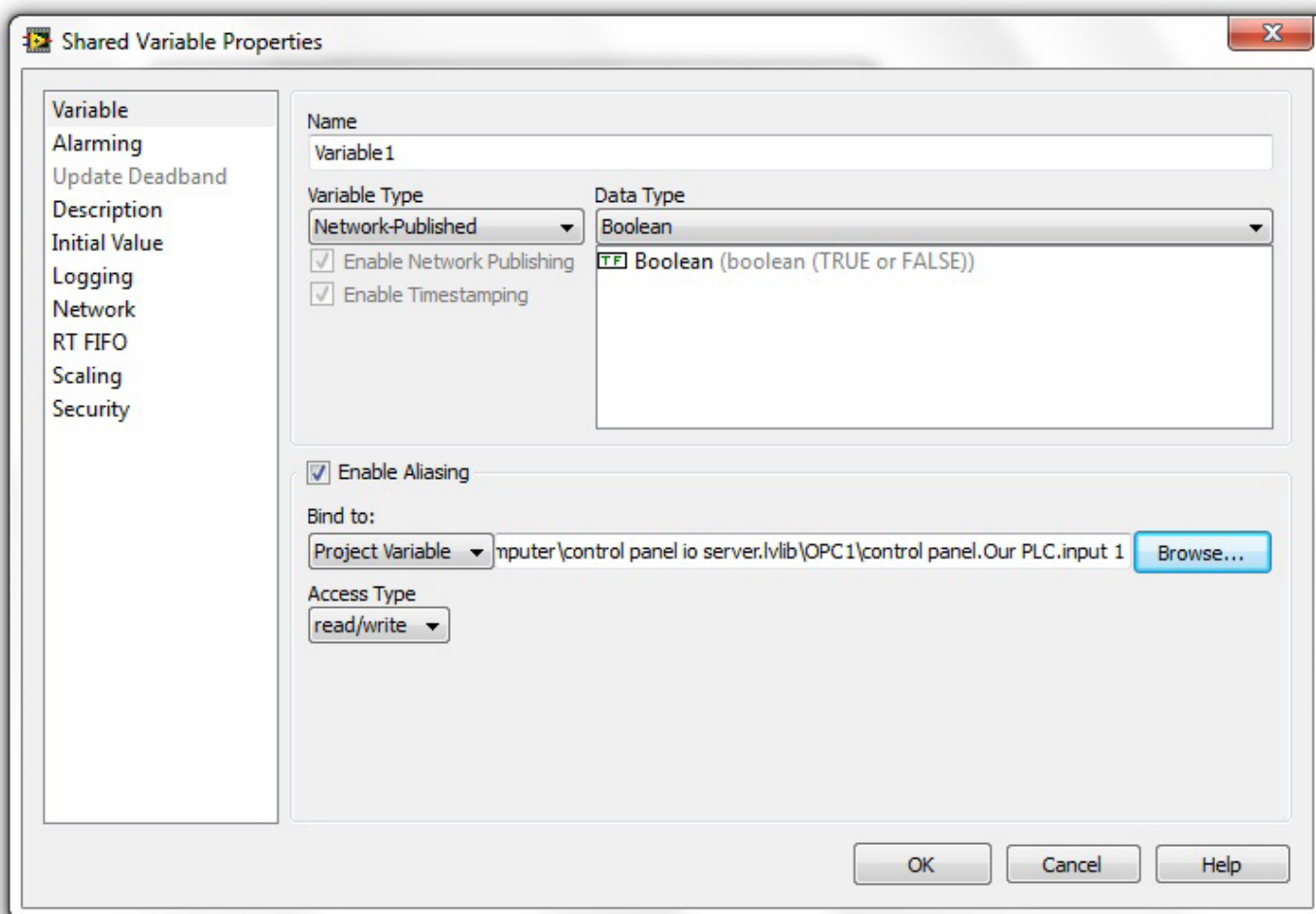


پس از معرفی opc server مربوطه به پروژه نوبت به ایجاد متغیرهای اشتراکی می باشد.  
بر روی گزینه control panel io server کلیک راست نموده و از شاخه new گزینه variable را انتخاب نمایید.

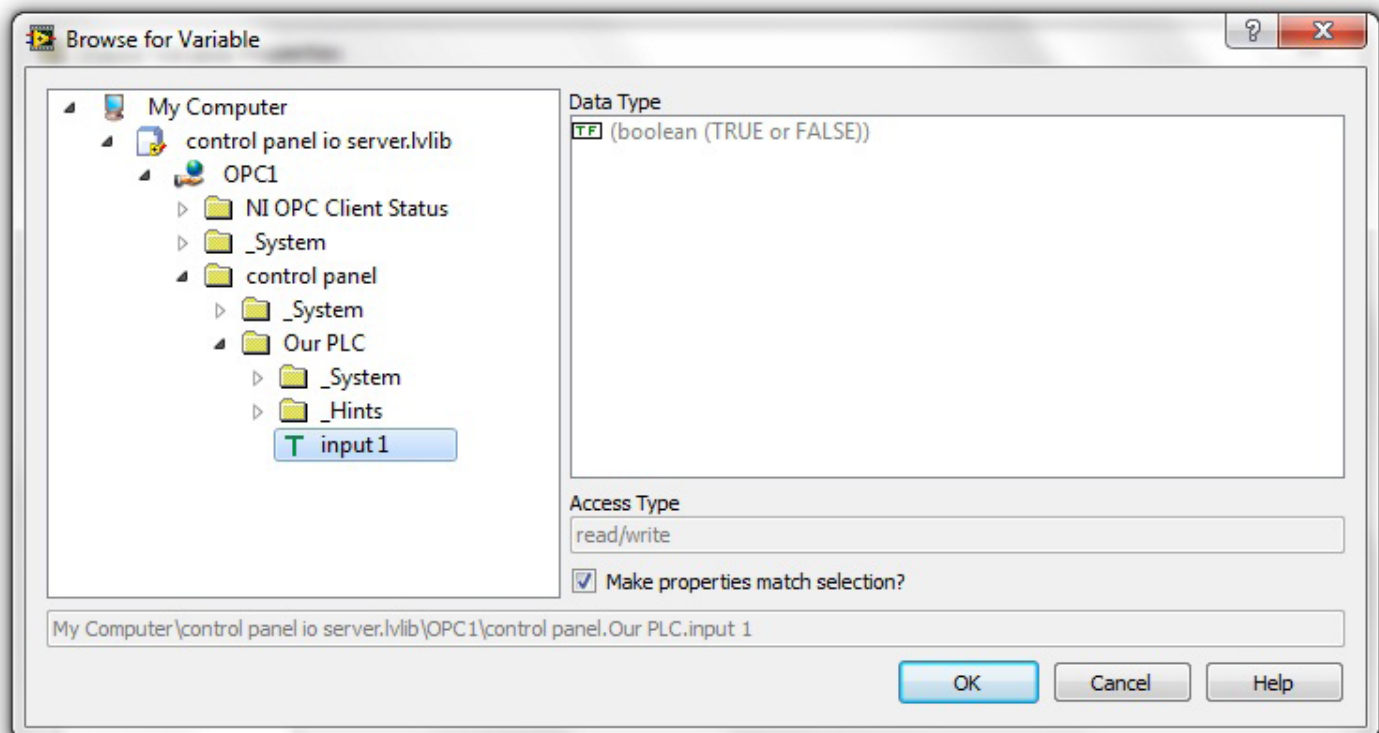


# ارتباط LabVIEW با انواع PLC از طریق OPC SERVER

عنوان پنجره باز شده shared variable properties می باشد. برای افزودن تگ های ایجاد شده در توسط برنامه ni opc server به برنامه خود ابتدا گزینه enable aliasing را فعال کنید. سپس گزینه Browse را انتخاب نمایید.

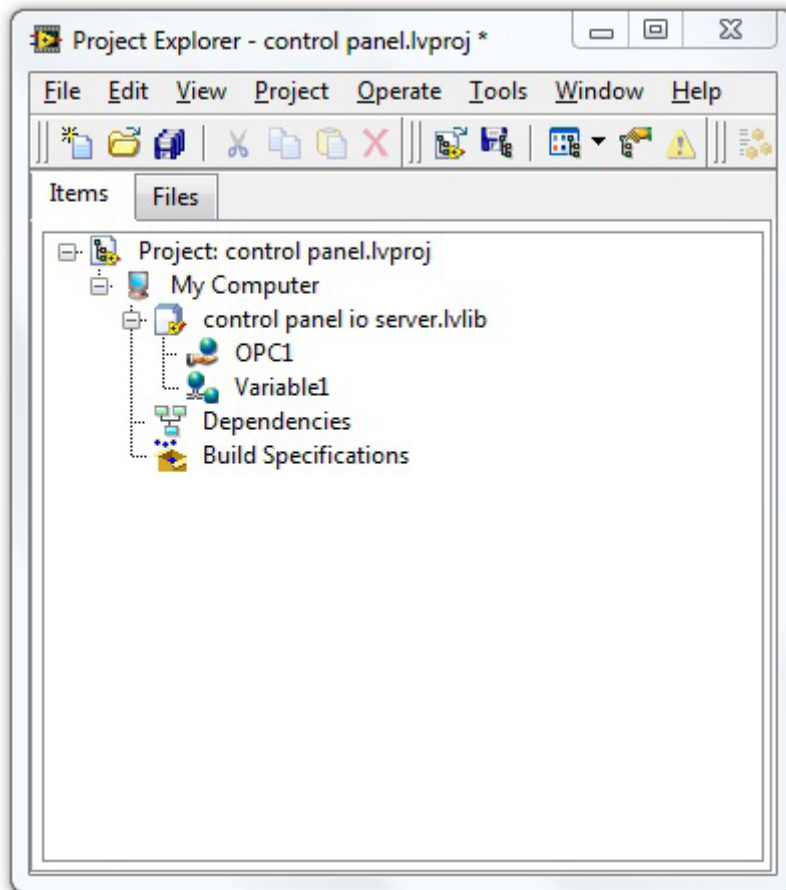


در پنجره باز شده (Browse for variable) زیر شاخه های my computer را تا رسیدن به تگ ها مطابق با شکل باز کنید. همان طور که مشاهده می نمایید کانال و دستگاه ایجاد شده در مراحل قبل در این پنجره مشاهده می شوند.

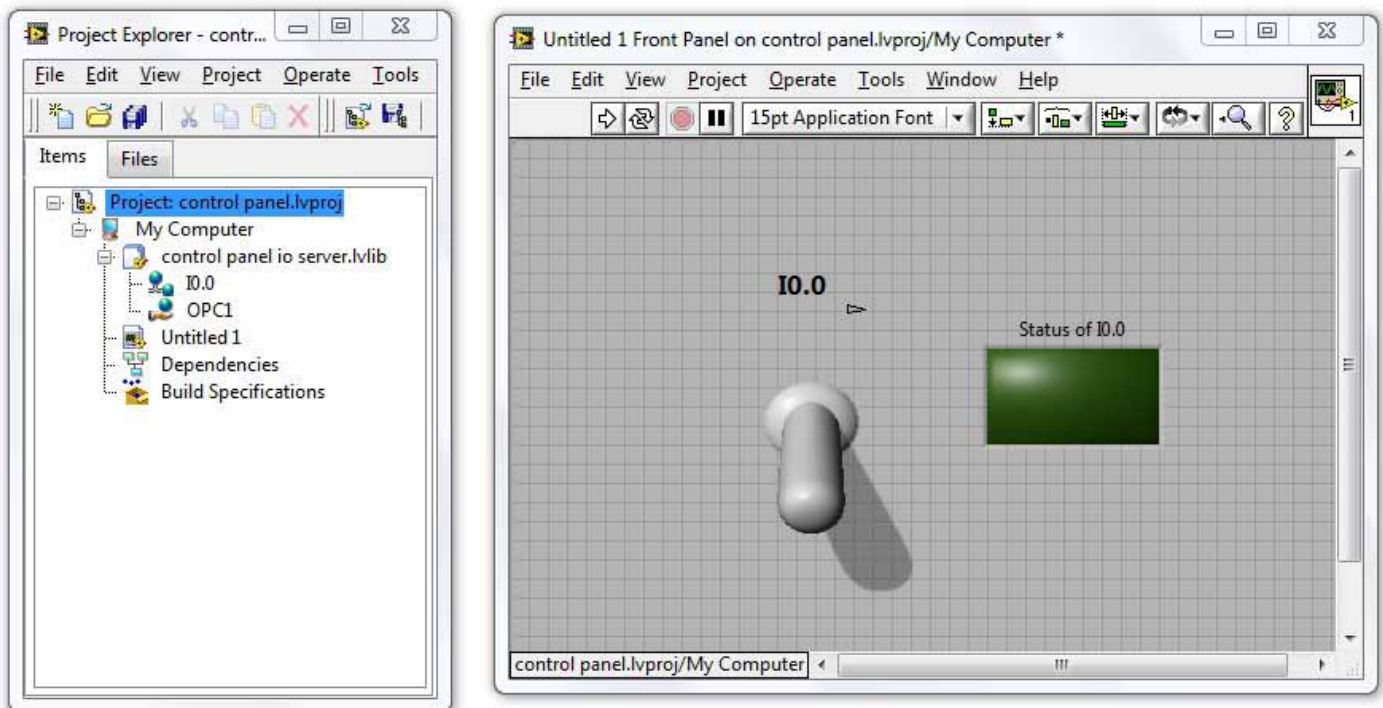


حال تگ های مورد نظر را انتخاب نمایید و در آخر دکمه OK را کلیک نمایید. در پنجره مشخصات متغیرها نیز ابتدا یک نام برای متغیر یا تگ درج نمایید و با زدن دکمه OK از پنجره خارج شوید. مشاهده می کنید که یک متغیر اشتراکی با نام مشخص انتخابی شما به پروژه اضافه

شده است. برنامه را دوباره ذخیره نمایید.



حال با ایجاد یک New Vi متغیرهای ایجاد شده را به درون صفحه پانل از طریق موس درگ نمایید و برای متغیرهای کنترلی یک نشان دهنده و یا برعکس ایجاد نمایید.



شما در حالت کلی نیاز به برنامه نویسی ندارید و با اجرای برنامه می توانید شاهد تغییرات در متغیرها باشید. ولی می توانید با هماهنگی با برنامه اصلی که در دستگاه نوشته شده است مقادیر را در LabView تغییر داده و نتیجه را در plc مشاهده نمایید. در حالت پیشرفته و کاربردی باید از توابع تولکیت DSC استفاده نمود. مراحل گفته شده در این آموزش مراحل کلی ارتباط با plc می باشد. در آموزش بعدی از تولکیت DSC در برقراری ارتباط استفاده خواهیم کرد.



# AVR Ethernet Board



در این دستگاه از تراشه ENC28J60 ساخت شرکت microchip استفاده شده است. این تراشه تمامی شرایط مورد نیاز برای اتصال به شبکه TCP/IP (لایه فیزیکی، mac و...) را یکجا فرا هم می کند. نحوه برقراری ارتباط این تراشه با میکرو کنترلر به وسیله پروتکل پر سرعت SPI می باشد. در این دستگاه علاوه بر خود تراشه ENC28J60، یک میکرو کنترلر ATMEGA32A به همراه یک LCD شستی و... قرار داده شده است. همچنین جهت سهولت در امر برنامه نویسی و یادگیری نحوه استفاده از این برد، فایل آموزشی به همراه کتابخانه های لازم جهت برنامه نویسی و پروژه نمونه در CD محصول قرار گرفته است.



## امکانات و مشخصات برد:

- \* میکرو کنترلر ATMEGA32A
- \* کنترلر شبکه ENC28J60 10Mbit
- \* ۴ عدد شستی مکان نما
- \* دارای LCD کاراکتری ۱۶\*۲
- \* امکان اتصال حافظه EEPROM
- \* کانکتور ISP برای پروگرام کردن برد
- \* کانکتور خروجی برای دو پورت A و C میکرو کنترلر
- \* نمایش ارسال و دریافت اطلاعات شبکه از طریق LED

توضیحات تکمیلی : <http://eshop.eca.ir/link/829.php>

# Digital ICs Tester



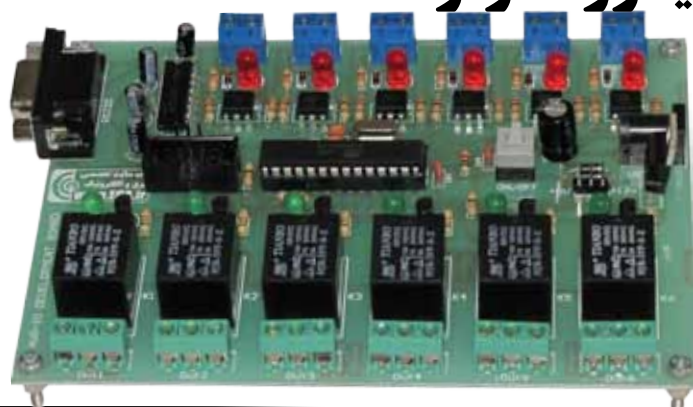
یکی از مشکلات موجود بر سر راه علاقمندان و طراحان الکترونیک که همیشه باعث اتلاف وقت بسیار زیادی می شود، خرابی و معیوب شدن قطعات الکترونیکی بوده که هیچ گاه از ظاهر قابل تشخیص نبوده و فقط با تست قطعه معیوب می توان از خرابی آن آگاه شد. این دستگاه یک تست کننده انواع IC مخصوص آزمایشگاه ها و جزئی لاینفک از تجهیزات طراحان حرفه ای می باشد که دارای امکاناتی جهت تست انواع IC های پر کاربرد خانواده های TTL و CMOS و ULN و... می باشد و می تواند در کمترین زمان ممکن قطعات را تست و سالم بودن یا خراب بودن تک تک گیت ها و یا کلیت قطعه را مشخص نماید.

قطعات قابل تست دستگاه :

- « IC های پر کاربرد سری 74XX
- « IC های پر کاربرد سری 40XX
- « IC های پر کاربرد سری 45XX
- « IC های پر کاربرد سری ULN 20XX
- « IC های پر کاربرد سری ULN 28XX

توضیحات تکمیلی : <http://eshop.eca.ir/link/633.php>

## برد توسعه ورودی خروجی میکرو کنترلر AVR



برد توسعه AVR جهت استفاده از پورت های ورودی و خروجی میکرو کنترلرهای AVR طراحی شده است.

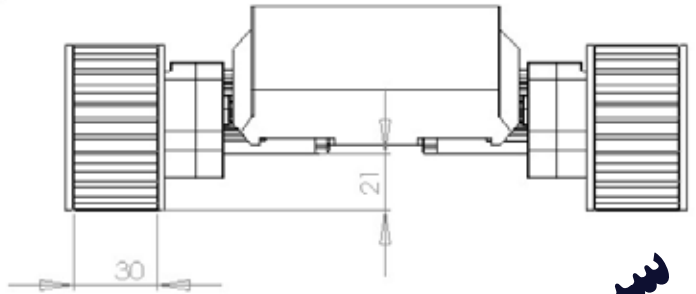
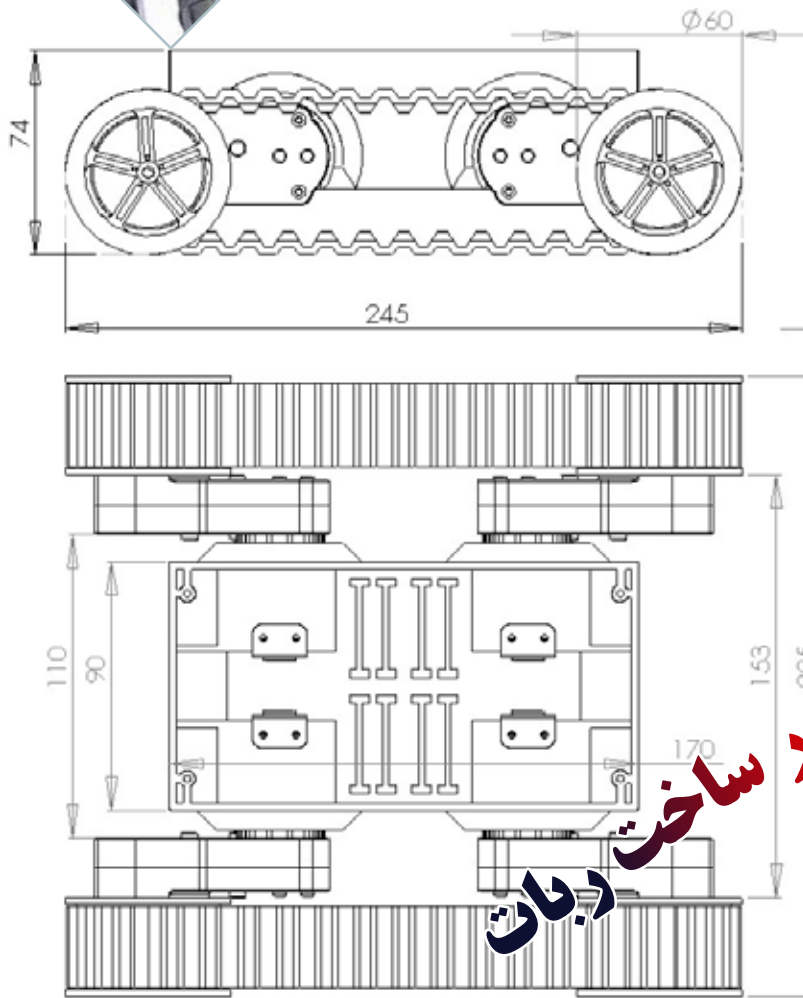
### کاربردها :

مصارف آموزشی، جهت آشنایی و تسلط بر امکانات میکرو کنترلرهای AVR  
استفاده در پروژه های هوشمندسازی خانه و سایر مصارف کنترلی در محیط های صنعتی به عنوان دستگاه های کنترلی مانند PLC  
استفاده در پروژه های اینترفیس آموزشی و کاربردی توسط پورت سریال  
همانک با نرم افزارهای برق و الکترونیک مانند MATLAB و LABVIEW و...



توضیحات تکمیلی : <http://eshop.eca.ir/link/621.php>





سیستم های متداول شاسی در ساخت ربات

مقصود از شاسی، یک یا چند قطعه است که تمامی قطعات اصلی ربات بر آن سوارند، ریسک عمده ی بخش مکانیک ربات ها معمولا ناشی از کیفیت طراحی و ساخت آن است. کوچکترین اشتباه در طراحی و ساخت شاسی، در آزمایش های اولیه ی ربات نمایان می شود. در این صورت سازندگان ربات ناگزیرند آنرا تقویت کرده یا اینکه شاسی را با طراحی جدید بسازند. این امر مستلزم افزایش زمان و هزینه های ساخت ربات خواهد شد. در صورت وجود محدودیت در نقدینگی، توقف پروژه دور از انتظار نخواهد بود. در صورت عدم ظهور مشکل در آزمایش های اولیه این مسئله در طول عمر ربات ظهور می یابد و ربات در فواصل زمانی کوتاه باید تعمیر شود. این امر موجب عدم اعتماد خریداران ربات به سازندگان خواهد شد.

سه عامل تاثیرگذار در طراحی شاسی ربات عبارتند از:

وزن ربات: عموما طراحان و سازندگان ربات به دلایل متعددی تلاش در کاهش وزن ربات دارند. طراحی شاسی از این مسئله تاثیر می پذیرد. مقاومت مکانیکی: شاسی می بایست تحمل وزن قطعات سنگین مانند بازو و یا پن تیل و استقامت مطلوب در برابر تنش های وارده به ربات را داشته باشد.

تعامل شاسی با سایر قطعات: در طراحی شاسی می بایست نحوه ی اتصال قطعات به شاسی مورد توجه قرار گیرد.

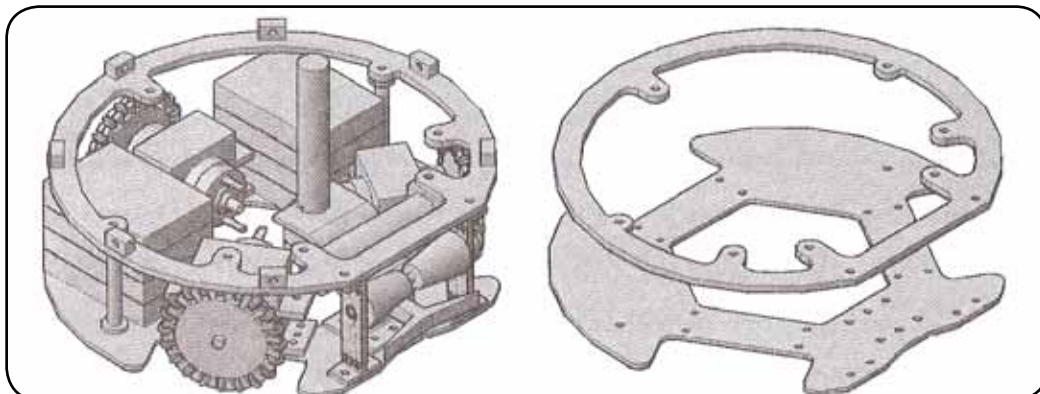
توجه بیش از اندازه به هر یک از این موارد، موجب کاهش کیفیت در موارد دیگر خواهد شد. این امر چالشی را پیش روی طراح قرار می دهد و با هر اقدام کوچک در طراحی، تمامی موارد بالا متاثر خواهند شد.

برای درک روشنتر این روند طراحی دستکش کوه نوردی را تحلیل می کنیم: در طراحی این دستکش ها، چنانچه ضخامت دستکش افزایش یابد، حرارت دست های کوهنورد بهتر حفظ می شود از طرفی با دستکش ضخیم گره زدن طناب دشوار است. در صورت کاهش ضخامت دستکش، کوهنورد با سهولت بیشتری طناب را گره می زند. اما حرارت دست های خود را از دست می دهد بنابراین باید ضخامت بهینه ای برای دستکش، انتخاب شود که امری دشوار است. این موضوع در طراحی شاسی ربات به شکل بسیار پیچیده ای ظاهر می شود.

در ادامه، انواع مکانیسم های متداول در ساخت شاسی ربات بررسی خواهد شد:

## شاسی صفحه ای

این سیستم متداول ترین سیستم ساخت شاسی در ربات متحرک است. در این روش ورقه ای فلزی یا غیر فلزی را با عملیات خم کاری یا ماشین کاری یا مجموع این دو به شکل مطلوب در می آورند و تمامی قطعات ربات را روی آن سوار می کنند.

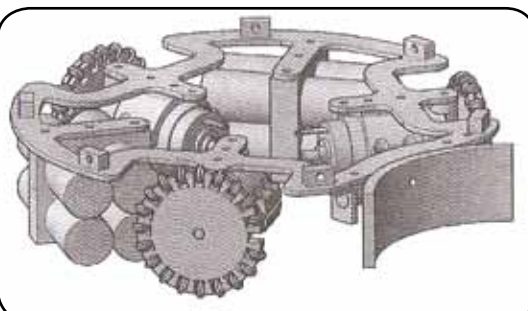


شاسی این ربات فوتبالیست سایز کوچک در دو طبقه ساخته شده است. شاسی پایین سوراخی هشت وجهی وجود دارد که علاوه بر کاهش وزن ربات، موجب تسهیل مونتاژ قطعات می شود. ارتفاع این ربات پنج سانتی متر و قطر آن هفده و نیم سانتی متر است. شاسی این ربات به کمک تکنیک وایرکات توسط نگارندگان ساخته شده است.

استفاده از این روش در ربات هایی با اندازه های کوچک کمک می کند که بدون استفاده از تکنیک های متداول ساخت ربات های متوسط و بزرگ مانند جوشکاری و استفاده از پیچ و مهره، شاسی ربات را بسازیم.

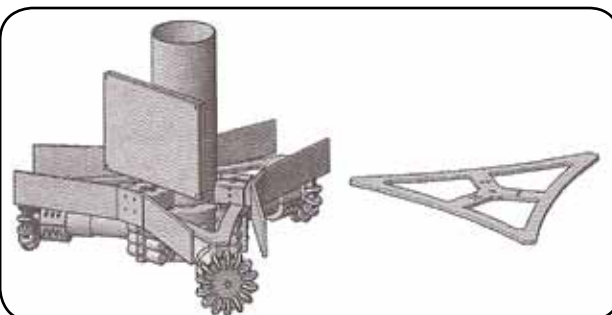
در ربات های متوسط و بزرگ، استفاده از این ورق ها به دلیل استقامت مطلوب می بایست با احتیاط بیشتری صورت گیرد. برای آنکه با مشکلی رو به رو نشویم بهتر است از ورقه های ضخیم بهره گرفته یا ورقه ی مورد استفاده را تقویت نماییم. برای انجام این امر می توانیم کناره های ورق ها را خم یا نک بزنیم، یا با اتصال قطعاتی فلزی مانند پروفیل در مناطق حساس، آنرا تقویت کنیم. برای اتصال قطعات به صفحات از جوشکاری استفاده نکنید. زیرا این عمل موجب بروز انبساط و انقباض

طراحی مناسب شاسی این ربات موجب سبکی آن شده است. وزن این ربات چیزی در حدود پانصد گرم است. این ربات ایلینگ توسط نگارندگان طراحی شده است.



های شدید شده و خمیدگی (تاب) ورق را در پی خواهد داشت. استفاده از ورق در شاسی ربات های متوسط و بزرگ سبب افزایش وزن آن شده و شاهد افول بسیاری از خصوصیات ربات خواهیم بود. شاسی صفحه ای برای ربات های متوسط و بزرگی مانند ربات های حمل بمب و ربات های فوتبالیست سایز متوسط مناسب نیست. چرا که نیروهای مکانیکی با اندازه ی بزرگ و شوک های مکانیکی مداوم، حاصل از سرعت بالا و تصادم های پی در پی ربات را به چالش دعوت می کنند.

در طراحی این ربات فوتبالیست سایز متوسط، از دو قسمت بالا و پایین شاسی استفاده شده است. با توجه به محیط پر تنش فعالیت این ربات بهتر است این شاسی از مواد مقاومی مانند تیتانیوم ساخته شود و ورق به کار رفته در آن قطری مناسب داشته باشد. این ربات توسط نگارندگان طراحی شده است.



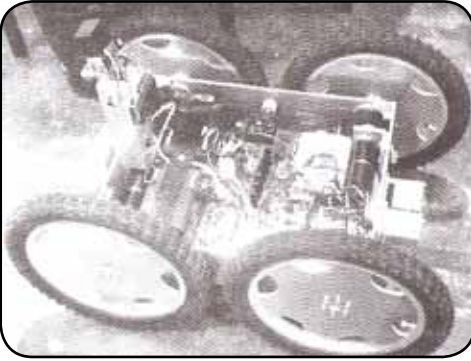
بسیاری از طراحان که برای اولین بار به طراحی چنین مکانیسمی می پردازند، واقعیتی بدیهی را نادیده می گیرند. هر صفحه ی فلزی دو سمت بالا و پایین دارد که قطعات ربات را می توان به هر دو سمت آن وصل کرد. اما بیشتر طراحان بنابر عادت یا پیش فرض های ذهنی نادرست، همه ی قطعات را در سطح بالایی شاسی نصب می کنند. این روش طراح را با کمبود فضا مواجه کرده و زمان طراحی را طولانی می کند یا آنکه

ناگزیر می شود از سیاست های انقباضی مانند دو طبقه کردن شاسی ربات بهره ببرد.

## شاسی جعبه ای

شاسی جعبه ای کمتر از دیگر روش ها در ساخت ربات های متحرک مورد استفاده قرار می گیرد. در این روش جعبه ای فلزی یا غیر فلزی ساخته یا خریداری می شود و تمامی قطعات ربات به این جعبه وصل می شود. به دلیل وضعیت هندسی، مقاومت مکانیکی جعبه های مکعب مستطیل بسیار مطلوب است.

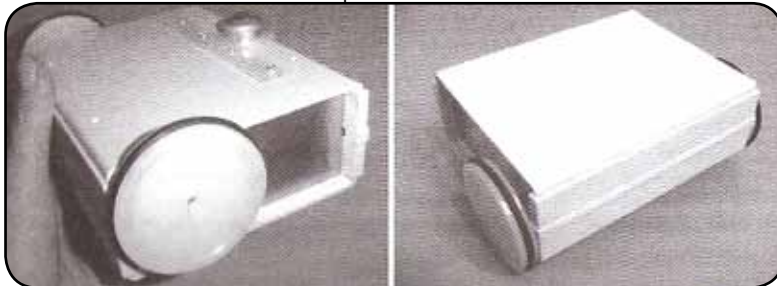
در گوشه های این جعبه ، چهار نبشی نصب شده که علاوه بر افزایش کیفیت ساخت ، اتصال درپوش ربات را نیز ممکن می سازند. این ربات روبوتو نام دارد و در رقابت های سال ۲۰۰۳ ایتالیا ، مقام نخست لیگ امدادگر را کسب کرده است.



هنگام استفاده از شاسی جعبه ای، به دلیل گستردگی بدنه ی داخلی جعبه، براحتی می توانید محلی را برای اتصال قطعات مکانیکی ربات خود بیابید. بنابراین انجام فرایند طراحی در این مکانیسم با سهولت بیشتری نسبت به روش های دیگر صورت می پذیرد. عموماً پیاده سازی اینگونه شاسی ها، طراح و سازنده را از تجربه ریسک، مصون می دارد. این روش در صورت تمایل به ساخت رباتی ضد آب و مقاوم در برابر حریق کارآمدترین روش است.

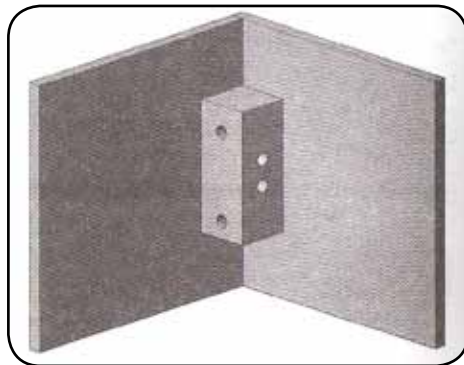
با توجه به مزایای بیشمار این ربات، علل عدم مطلوبیت این سیستم کدامند؟ وزن بالای این سیستم در حدی است که لازم است از سیستم

محرکه ای پرتوان و پر مصرف برای راه اندازی ربات استفاده شود. این امر موجب تخلیه سریع انرژی باتری ها می شود. همچنین در زمان ساخت نیز به دلیل وزن بالا، حمل و نقل ربات با مرارت غیر قابل تصویری صورت می پذیرد. مساله ی سوم که توفیق این ربات را کاهش داده است، اختلال در سیستم مخابراتی بی سیم ربات به دلیل فلزی بودن جعبه ی آن است. زیرا جعبه به صورت شیلدی مخابراتی عمل می کند.



این ربات با جعبه های موجود در بازار توسط نگارندگان ساخته شده است.

به طراحانی که اولین تجربه ی ساخت ربات را سپری می کنند توصیه می شود از این مکانیسم بهره ببرند. در صورت تمایل به ساخت ربات تحقیقاتی کوچک و مقرون به صرفه، می توان جعبه ای کوچک را با بهایی مناسب خریداری و بر اساس خصوصیات آن، ربات را طراحی کرد.



جعبه های فلزی آماده برای ربات های کوچک مناسبند، اما استقامت لازم جهت ایفای نقش در ربات های بزرگ و متوسط را ندارند. برای ساخت جعبه ای مقاوم و مطلوب توصیه می شود چند ورق آلومینیوم با ضخامت های مناسب (پنج تا ده میلیمتر برای ربات اندازه ی متوسط) را بریده و به کمک قطعه ای واسط با زاویه ی نوذ درجه به یکدیگر پیچ و مهره کنید و سپس فرایند جوشکاری آرگون را انجام دهید. در هنگام جوشکاری بهتر است یک خالجوش در ابتدای مسیر این فرایند را تا زمان تکمیل جعبه ادامه دهید، سپس به تعداد نیاز، نبشی به گوشه ها جوش داده و پس از آن جوشکاری را به طور کامل انجام دهید.

برای حفظ حالت مطلوب جعبه، پس از جوشکاری اجازه دهید قطعه در هوای آزاد خنک شود و از آب استفاده نکنید. برای ارتقای کیفیت ظاهری، جوشکاری ها را از داخل جعبه انجام نمونه ای از قطعه ی واسطه که پس از زدن دو خال جوش در بالا و پایین مسیر جوشکاری مسیر جوشکاری، از قطعه کار جدا می شود. یک سانتیمتری به نتیجه ی مطلوب خواهد رسید.

## شاسی ستونی

اصول کار این شاسی، به شاسی اتومبیل شباهت دارد. در این روش، تمام قطعات ربات به چندین ستون متصل به هم، نصب است. در طراحی این نوع شاسی باید موارد بیشماری لحاظ شود. بنابراین، این نوع شاسی بیشتر از انواع دیگر، طراح را در فضای ریسک قرار می دهد. طراحی این نوع شاسی در مقایسه با انواع دیگر بسیار دشوارتر است. به طراحانی که زمان کافی در اختیار ندارند یا مواد اولیه موجود در بازار را به خوبی نمی شناسند یا در فرایند طراحی صبور نیستند، توصیه می شود از این مکانیسم پرهیزند.

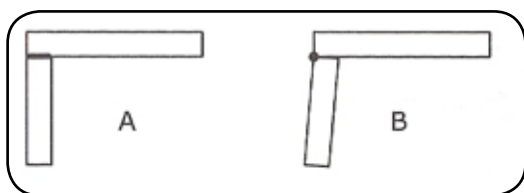
مهمترین مزیت این نوع شاسی وزن بسیار کم آن در مقایسه با دیگر مکانیسم هاست. در صورت استفاده از لوله یا پروفیل برای ساخت، لوله ها مسیر امنی برای عبور سیم ها خواهند بود و این سیم کشی مخفی موجب ارتقای جذابیت بصری ربات شده و نمایی حرفه ای تر می یابد. پیاده سازی سیستم تعلیق در این گونه ربات ها آسانتر است.

برخی نکات مهم که می بایست در طراحی این ربات باید مورد توجه قرار گیرد عبارتند از:

۱- روش های متعددی برای جوشکاری فلزات وجود دارد. در بیشتر این روش ها ممکن است قطعه یا خط جوش، انبساط و انقباض شدیدی داشته باشد. این مساله در مورد تکنیک هایی مانند جوشکاری آرگون که حرارت موضعی شدیدی ایجاد می کند، نمود بیشتری دارد. انقباض و انبساط، نیرو های بسیار قدرتمندی را بر قطعه تحمیل کرده که موجب بروز تنش در قطعه شده و منجر به تغییر شکل آن خواهد شد. در شکل های زیر، دو روش مقابله با شاسعترین معضلات فرایند جوشکاری بیان شده است که نگارنده در سالین گذشته، در حساسترین مقاطع زمانی با

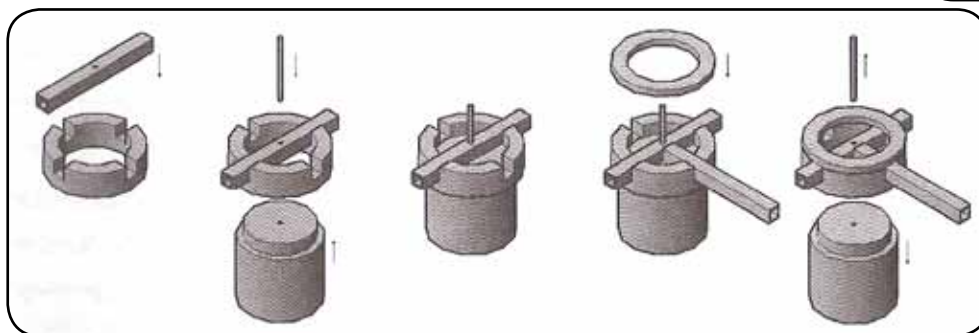
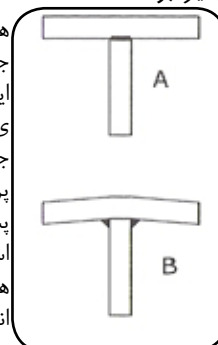


آنها در گیر بوده است.



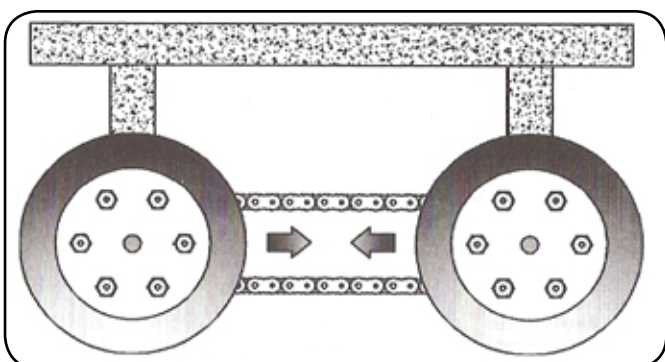
انقباض حاصل از جوشکاری موجب معیوب شدن قطعه ی  $\beta$  شده که تعمیر و جوشکاری مجدد آن دشوار است. بنابراین توصیه می شود قطعه را همانند تصویر A جوشکاری کنید.

هنگام سرد شدن محل جوشکاری ، نقطه جوش ها منقبض می شوند. نیروی حاصل از این انقباض موجبات تغییر شکل قطعه (قطعه ی  $\beta$ ) را فراهم می کند. این تغییر شکل در جوشکاری برخی از اقلام مانند لوله ها و پروفیل ها بسیار شدید بوده و ترمیم قطعه پس از جوشکاری بسیار دشوار و تقریباً ناممکن است. برای پیشگیری بهتر است جوشکاری همانند شکل A صورت می گیرد و خط جوش اندکی از لبه ها فاصله داشته باشد.



ساختار این فیکسچر به کمک سوراخ روی قطعه امکان می دهند که قطعه را دقیقاً در محل خاصی ثابت نگه داشت. به کمک این ابزار جوشکاری در دو طرف قطعه امکان پذیر است. این فیکسچر ، حلال بسیاری از مشکلات نگارندگان بوده است و بهره برداری از آن توصیه می شود. به جای فیکسچر از گیره ی دوقلوی نود درجه ی تجاری نیز می توان استفاده کرد.

اگر شاسی از قطعات فولادی ساخته شده باشد، بهتر است پس از ساخت با روش های عملیات حرارتی عملیات تنش زدایی بر روی شاسی صورت پذیرد.

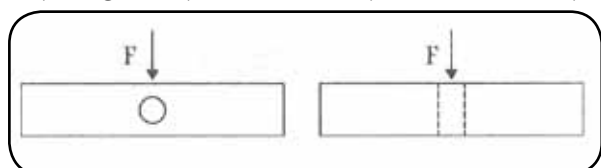


سفتی بیش از اندازه ی زنجیر ، موجب خمیدگی شاسی می شود. این موضوع در مورد شنی های ربات حادثر بروز می کند. هر دو مورد ، از مشکلات بزرگ نگارنده بوده است.

۲- در بسیاری از ربات ها، مجموعه هایی سنگین مانند بازو یا پن تیل (قسمت نگهدارنده دوربین) یا باتری های بسیار سنگین به ربات متصل می شوند. این مجموعه ها، نیرو های بسیار زیادی را به شاسی وارد می کنند، خصوصاً هنگامی که ربات شرایط حرکت در دست انداز را تجربه می کند. در گیر شدن ربات با دست انداز ها، عامل ایجاد پارامتر شتاب در راستای عمود بر زمین می شود. این امر موجب می شود. مراکز سنگین وزن، شرایط دشواری را به شاسی تحمیل کنند. بنابراین در طراحی باید به استقامت شاسی در محل های اتصال ادوات توجه شود.

۳- در ساختمان ربات، نیرو های متعددی وجود دارند که برخی از آنها از دید طراحان مخفی می ماند و اثرات مخرب آنها پس از ساخت ربات، بحران ساز می شوند.

عموماً طراحان به دلیل زمان ناکافی یا فقدان تخصص لازم برای تحلیل های مقاومت مصالح در شاسی ربات، تاثیرات نیرو ها بر موارد را به صورت ذهنی می سنجند. این امر موجب می شود در بسیاری از موارد طراح بیش از اندازه به مواد و ساختار اعتماد کند. برای پرهیز از بروز چنین مسایلی، بهتر است از مواد یا ساختار هایی که بیش از میزان مورد نیاز استقامت دارند استفاده شود. برای شفاف شدن این امر مثالی از مسایل عمرانی را مطرح می کنیم. فرض کنید که در طراحی یک تیر افقی، تیر آهن هجده پاسخگوست، اما برای پرهیز از اشتباه از تیر آهن بیست بهره برید. ابعاد تیر آهن اندکی تغییر کرده اما ویژگی های مکانیکی آن بهبود چشمگیری یافته است. در عرصه ی مکانیک نیز با اندکی تغییر در ابعاد، می توان ویژگی های مکانیکی را متحول کرد.



۴- توصیه می شود در طراحی، محل قرارگیری سوراخ ها بگونه ای باشد که به ویژگی های مکانیکی شاسی خدشه ای وارد نشود.

در ساخت این خانواده از شاسی ها می توان از مواد سودمند متنوعی بهره برد. با توجه به جهت نیروی وارد شده به این دو قطعه ، سوراخ قطعه ی سمت راست تاثیر بسیار بدی در استحکام خمشی به جا می گذارد اما تاثیر منفی سوراخ قطعه ی سمت چپ ناچیز است.



پروفیل های استنلس استیل: ماشین آلات صنایع غذایی و صنعت ساختمان، دوشاخه ی اصلی مصرف کننده ی عمده ی این ماده اولیه اند. پس این پروفیل ها را می توان در هر زمانی با تنوع زیادی در بازار یافت. البته به دلیل مصرف بیشتر این کالا در تهران و شهرهای همجوار، یافتن این اقلام در خارج از استان تهران دشوار است و در صوت یافت شدن، تنوع آن بسیار کم است.

پروفیل های استنلس استیل در ابعاد بسیار متنوعی تولید می شوند. اما سطح مقطع بیشتر پروفیل های موجود در بازار ایران مربع است. پروفیل های استنلس استیل محصول شکل دهی ورق استیل بوده و ضخامت این ورق ها بسته به ابعاد پروفیل بین یک تا سه میلیمتر است.

پروفیل های مذکور را می توانید در مقطع  $10 \times 10$ ،  $20 \times 10$ ،  $30 \times 30$ ،  $40 \times 40$ ،  $50 \times 50$  و  $80 \times 80$  و  $100 \times 100$  میلیمتر به راحتی یافت می شوند. البته اندازه های دیگری از این پروفیل ها در بازار موجود است که مصارف محدودی داشته و نمی توان به وجود همیشگی آنها در بازار اعتماد کرد.

چند ویژگی برجسته موجود برتری این ماده نسبت به سایر مواد می شود:

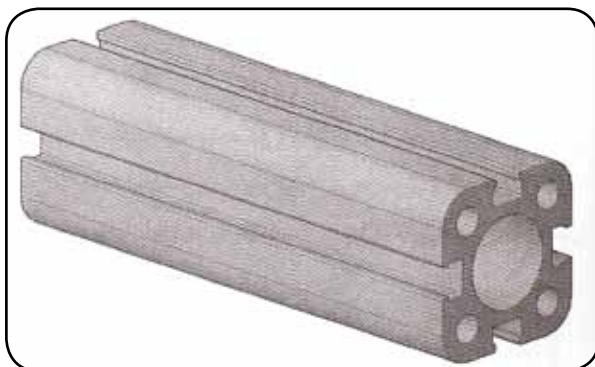
۱- الاستیسیته بالای آن موجب می شود که تنش ها درون شاسی مستهلک شده و از شکست یا تغییر شکل شاسی و قطعات دیگر پیشگیری شود.  
۲- چکش خوار است و هنگام بروز تنش های قدرتمند، دچار شکست نشده و فقط اندکی تغییر شکل می یابد که قالباً جزئی بوده و تاثیر آن مانع ایفای نقش ربات نمی شود.

۳- اکسید نمی شود و بنابراین دوام و زیبایی ربات افزایش می یابد.

۴- برخی از انواع آن فاقد خاصیت مغناطیسی بوده که این امر موجب کاهش نویز سنسور های قطب نمای دیجیتالی می شود.

به دلیل انبارش نامناسب این پروفیل ها، بسیاری از آنها دچار خمیدگی می شوند. بنابراین هنگام خرید از سلامت پروفیل ها مطمئن شوید. آلومینیوم: از پروفیل ها با مقاطع تو پر این فلز می توان برای ساخت شاسی بهره گرفت. این پروفیل ها در ساخت در و پنجره کاربرد وسیعی دارند. بنابراین تامین آن در تمامی شهرها امکان پذیر است. روش رایج برای اتصال قطعات آلومینیومی، پیچ و مهره و پرچ است که البته استفاده از قطعات واسط، این امر را آسان می کند.

جوشکاری آرگون و جوشکاری  $CO_2$  نیز از روش های رایج اتصال قطعات آلومینیومی به یکدیگر است که در صورت لزوم می توان از آنها کمک گرفت. در صورت تمایل به ارایه ی خروجی با کیفیت بهتر است از جوشکاری آرگون بهره مند شوید.

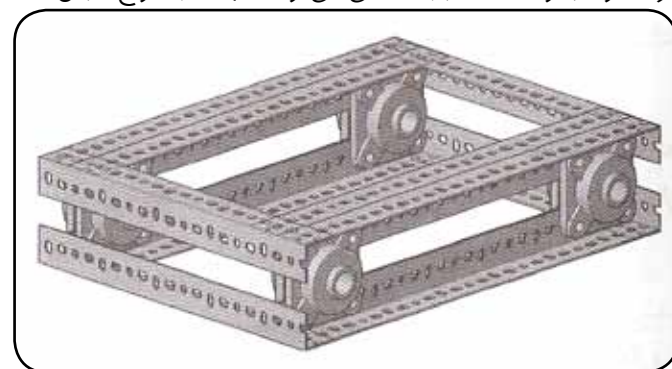


برخی از مقاطع آلومینیومی که به صورت استاندارد در بازار وجود دارد، کاربرد های ویژه ای دارند.

یکی از این مقاطع استاندارد پروفیل های دکوراسیون داخلی است که سازندگان تجهیزات اداری، تولید کننده و مصرف کننده ی آنها، این مقاطع استاندارد دیگر پروفیل های ویژه ی تجهیزات پنوماتیک است. برخی از اقلام آن در بازار وجود دارد و نمایندگی های معتبر شرکت های پنوماتیک بین المللی مقاطع ویژه آنرا وارد می کنند.

مزیت استفاده از این مقاطع آماده، سهولت کاربرد آنهاست. به کمک قطعات این مقطع در پنوماتیک کاربرد داشته و به کمک شیار های روی آن می توان چند تکه از آنها را به یکدیگر متصل کرد. سوراخ های داخل آن واسطی که در این صنعت وجود دارند براحتی می توان، قطعات را به یکدیگر جهت سیم کشی یا اتصال به قطعات دیگر مفید هستند. متصل کرد.

پروفیل های فولاد ساختمانی: این پروفیل ها با تنوعی باورنکردنی در سراسر کشور وجود دارند. و به راحتی می توان آنها را با انواع روش های



نمونه ای از قفسه های انبار که برای ساخت شاسی به کار گرفته شده است.

جوشکاری قوس الکتریکی به یکدیگر متصل کرد. البته به کارگیری این فلز موجب افزایش وزن ربات خواهد شد. پس از جوشکاری باروش های متداول جوشکاری قوس الکتریکی، کیفیت جوش را با دقت بررسی کنید. مدول های ساخت قفسه های انبار: مدول های قفسه های انبار را به راحتی می توان با زاویه ۹۰ درجه به یکدیگر متصل و چهارچوبی با طرح دلخواه ایجاد کرد.

علاقه مندان به این روش معمولاً با اتصال یک جفت پاتاقان به این مدول ها شافت های خود را به شاسی متصل می کنند. اتصال این مدول ها معمولاً با پیچ و مهره امکان پذیر است و به جوشکاری نیازی نیست.

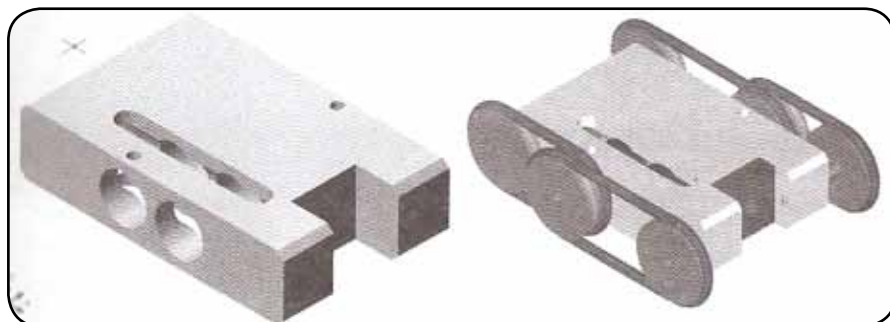
لوله های پلی پروپیلن: این لوله ها در سالیان اخیر کاربرد گسترده ای در تاسیسات ساختمان یافته اند. بنابراین می توان آنها را با بهای مناسب در سراسر کشور تهیه کرد.

# سیستم های متداول شاسی در ساخت ربات

در این روش ابتدا لوله ها را به کمک اره دستی یا قیچی مخصوص به اندازه های دلخواه بریده سپس به کمک اتوهای مخصوص لوله کشی، قطعات پلاستیک را که ممکن است زانوی لوله ی سه راهی یا اقلام دیگر باشند ذوب کرده و به یکدیگر جوش دهید. توصیه می شود قبل از اجرای این روش با متخصص تاسیسات مشورت کرده و اجرای این کار را نیز به مجریان تاسیسات ساختمان بسپارید. وجود اندازه های بی شمار در قطر لوله ها این فرصت را برای شما پدید می آورد که به مقاومت مکانیکی مطلوب خود دست یابید. اما در صورتی که پس از ساخت شاسی به ویژگی های مطلوب مکانیکی دست نیافتید، می توانید پس از ساخته شدن شاسی، با پر کردن لوله ها یا پوشاندن آنها با فایبرگلاس، شاسی را تقویت کنید. تقویت لوله ها قبل از متصل کردن به یکدیگر فرایندی اشتباه است.



چند نما از شاسی ربات قالبی و نحوه ی جای گیری قطعات در درون آن



نمونه ای از شاسی قالبی به تنهایی و زمانی که قطعات به آن متصل می شوند. این شاسی توسط نگارندگان طراحی شده است.

این ربات دو طبقه بوده که طبقه ی پایین آن روی شاسی ورقه ای و طبقه ی بالای آن روی شاسی ستونی بنا شده است. این ربات در رقابت های رباتیک سال ۲۰۰۳ ایتالیا در رشته ربات امدادگر فعالیت داشته و در سال های قبل از آن در رقابت های فوتبال شرکت کرده است.



چند ستون افقی آنرا تقویت کند. همچنین ممکن است ربات روی شاسی جعبه ای بنا شده باشد و بازوی آن ساختار ستونی داشته باشد.

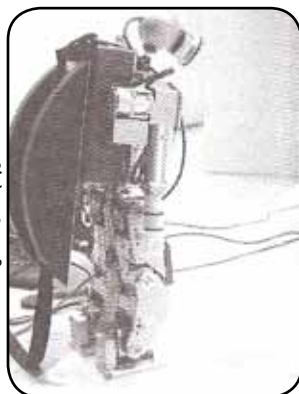
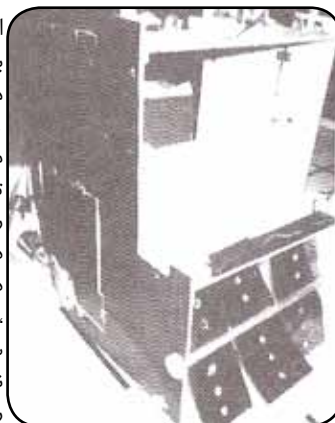
## شاسی های ترکیبی

گاهی ممکن است طراح بخواهد دو یا سه سیستم شاسی را با یکدیگر به کار گیرد. برای مثال ممکن است طبقه ی اول ربات با شاسی صفحه ای و طبقه ی دوم آن با شاسی جعبه ای ساخته شود. در جای دیگر ممکن است طراح شاسی صفحه ای داشته باشد و یا

## ربات های فاقد شاسی مشخص

در بسیاری از خانواده های ربات ها، ساختار ربات به گونه ای است که نمی توان قطعه ی مشخصی را به عنوان شاسی شناخت، در حقیقت چندین قطعه ی اصلی وجود دارد که نقش شاسی بین آنها تقسیم شده است.

این ربات بر اساس ساختار رشته ای بنا شده است. این ربات برای رقابت های رباتیک دانشگاه تهران در سال ۱۳۷۹ ساخته شده است. در این مسابقه ربات می بایست چند عدد توپ، بطری نوشابه و قالب صابون را از یکدیگر جدا کرده و هر یک را درون سیدی مخصوص قرار می داد. در ساختمان این ربات سیستم حرکتی، آسانسور، جدا کننده، سه مخزن و سیستم جمع آوری توپ وجود داشت. نگارندگان این کتاب در ساخت این ربات مشارکت داشتند.



یکی از ربات های شرکت کننده در رقابت های سال ۲۰۰۳ ایتالیا که فاقد شاسی مشخص است.

بارزترین رباتی که در این دسته قرار می گیرد، ربات انسان نماست. بازو های رباتیک نیز عموماً فاقد ساختار مشخصی به نام شاسی هستند. همچنین بسیاری از ربات های دیگر نیز به دلیل وجود برخی از محدودیت های طراحی، فاقد شاسی مشخص اند. ممکن است اتصالات اینگونه ربات ها به صورت رشته ای باشد. بدین معنی که قطعه ی A به B متصل و B نیز به C متصل شده است و این رشته ادامه دارد. ربات های مار نیز از این الگو پیروی می کنند. البته بسیاری از سیستم های دیگر نیز بر اساس این الگو بنا شده اند.

منبع : کتاب دانش کاربردی، طراحی و ساخت روبات نویسدگان: پویا حیرتی - ابوذر آقاجانی - محمد چیت سازان

## انواع چرخ ربات



توضیحات تکمیلی : <http://eshop.eca.ir/link/7576.php>

## انواع موتور گیربکس های ربات



توضیحات تکمیلی : <http://eshop.eca.ir/link/7588.php>



# برد آموزشی ربات تعقیب خط



## امکانات برد آموزشی ربات تعقیب خط :

- ❖ دارای میکروکنترلر ATMEGA16 و هماهنگ با سایر میکروکنترلرهای ۴۰ پایه سری ATMEGA
- ❖ نمایشگر LCD کاراکتری ۱۶\*۲
- ❖ تغذیه مجزا برای راه اندازی موتورها و برد کنترلی
- ❖ امکان درایو دو موتور با حداکثر جریان ۲ آمپر و رنج ولتاژ ۵ تا ۴۸ ولت توسط درایور L298
- ❖ وجود دیودهای محافظ برای حفاظت از درایور موتور
- ❖ دارای خروجی سریال TTL و RS232
- ❖ قابلیت اتصال مستقیم به پورت COM رایانه
- ❖ امکان اتصال ماژول های وایرلس نظیر ماژول های RF و بلوتوث
- ❖ کانکتور ISP جهت پروگرام نمودن میکروکنترلر
- ❖ خروجی پورت B جهت کاربردهای تکمیلی
- ❖ قابلیت اتصال ۸ سنسور IR
- ❖ سازگاری با انواع سنسورهای IR موجود
- ❖ ۸ عدد LED جهت نمایش حالات سنسورها
- ❖ برد مجزا برای اتصال سنسورها
- ❖ امکان چینش سنسورها متناسب با مسیر پیست
- ❖ قابلیت نصب بر روی شاسی دلخواه

## محتویات :

- ❖ برد ربات تعقیب خط
- ❖ برد سنسور
- ❖ چهار عدد سنسور ON2253
- ❖ یک عدد شاسی پلکسی در ابعاد ۱۰x۱۰ سانتی متر
- ❖ چهار عدد اسپیسر و چهار عدد پیچ و مهره جهت اتصالات مکانیکی
- ❖ CD همراه محصول شامل :
- ❖ شماتیک و اطلاعات پایه ای برد آموزشی
- ❖ معرفی و آموزش کامل ربات تعقیب خط
- ❖ آموزش برنامه نویسی میکروکنترلرهای AVR
- ❖ کامپایلرها و نرم افزارهای مورد نیاز
- ❖ برنامه های نمونه جهت راه اندازی کامل ربات

توضیحات تکمیلی : <http://eshop.eca.ir/link/855.php>





# POWER

Power Systems , Generation , Transmission and Distribution

# برق قدرت



کاهش ریدیل گشتاور در موتورهای DC بدون جاروبک با استفاده از تک سنسور جریان IDC

نویسنده: بهزاد سلمانی  
 behzad\_engineer@yahoo.com

مقاله در مورد موتورهای DC بدون جاروبک با استفاده از تک سنسور جریان IDC

در این مقاله یک مطالعه وسیع بر روی کاهش تغییرات ریدیل گشتاور تولید شده در موتورهای DC بدون جاروبک فقط با تک سنسور جریان DC link- فراهم شده است. در این قبیل کنترل ها تغییرات ریدیل گشتاور که جزء تکنیک های جلوگیری کننده اغتشاش می باشد تاثیر کاربردی در سرعت پائین در سرعت بالا دارد. تکنیک جبران سازی گشتاور بر پایه یک استرژژی ورودی و خروجی جریان به فازها و فاصله تغییرات که با استفاده از یک روش کنترلی توان موثر تغییرات را ثابت نگه داشته و آن را برابر می سازدو به حالت تعادل می رساند ایجاد شده است. حال با افزایش جریان، روش کنترل جریان ریدیل ها را متوقف می سازد. روش کنترلی، کار متوقف سازی ریدیل ها و افزایش جریان و گشتاور در اینورتر در مدت زمان مورد نظر را بر عهده دارد. اثرات گفته شده به سهولت توسط شبیه سازی در روش کنترلی قابل دسترسی می باشد.

## مقدمه

در این مقاله موتورهای DC بدون جاروبک (BLDC) back-EMF (BLDC) دوزنقه ای که ناشی از افزایش قدرت می باشد بررسی شده اند که به علت روش کنترلی آسان، استفاده وسیعی دارند. علاوه بر این، موتورهای DC بدون جاروبک با ساخت و به کارگیری یک سنسور جریان DC برای تنظیم جریان فازها از میان هر سه فاز موتور زمانیکه کنترل جریان استاتور با زاویه ۱۲۰ درجه می باشد طراحی گردیده است که این کنترل بر پایه تک سنسور سیگنال جریان انجام وظیفه می کند. تغییرات ریدیل گشتاور معمولا در کنترل کاهش ریدیل جریان فازها در ناحیه کموتاسیون جریان رخ می دهد. به عنوان مثال تغییرات ریدیل گشتاور در موتورهای DC بدون جاروبک دوزنقه ای که شامل ثابت نگه داشتن گشتاور در رنج سرعت پائین و کاهش ریدیل گشتاور در رنج سرعت بالا است را نشان می دهد. برای کنترل موتورهای DC بدون جاروبک با سنسورهای جریان سه فاز تحقیقات فراوانی مربوط به تغییرات ریدیل گشتاور وجود دارد که خارج از بحث این مقاله می باشد. روش هایی برای کاهش تغییرات ریدیل در موتور BLDC با تک سنسور جریان در مقالات وجود دارد. کنترل ولتاژ تک فاز با نسبت توان خواسته شده در مدت تغییرات برای کاهش ریدیل گشتاور تنظیم شده است. در روش overlap سوئیچینگ، با تمام شدن زمان مورد نظر، خارج شدن سوئیچ ها با تاخیر در همان زمان می

# کاهش رپیل گشتاور در موتورهای DC بدون جاروبک

باشد. یک کنترل جریان با یک تغییرات فعال فقط در طی مدت تغییرات کاهش پیک جریان در عدم تغییر فازها ظاهر می شود. اگرچه مطابق این قانون و قاعده محدودیت تاثیرات در کاربرد درخواست متقاضیان در حساسیت پارامترهای موتور نشان داده شده است اما عدم رضایت مجریان در تمامی رنج سرعت مورد نظر به چشم می خورد. این مقاله یک تحلیل جامع و وسیع در روش متوقف سازی تغییرات رپیل گشتاور در موتور BLDC فقط با تک سنسور سیگنال جریان مهیا کرده است. در اینجا از تاثیرات و تغییرات کوچک رپیل گشتاور صرف نظر می شود. یک جریان کنترل شده و ثابت افزایش را در کنترل خود دارا می باشد. طرح کنترلی پیشنهاد شده تغییرات جریان رادر ورود و خروج به فازها در مدت زمان تغییرات متعادل می کند. این تکنیک جدید تولید رپیل گشتاور در سرتاسر رنج مورد نظر را کاهش می دهد.

## کنترل جریان

### مد نرمال

شکل ۳ جریان DC-link و جریان های هم تراز آنها را زمانیکه در اینورتر سوئیچ های S1 و S2 در مد PWM عمل می کند نشان می دهد. طبق شرایط سوئیچینگ سوئیچ های اینورتر معادله ولتاژ را شرح داده و switch on و switch off را در مدت عملکرد این مد توضیح می دهد.

$$V_{ab} = V_d = 2R \cdot i_d + 2L \left( \frac{di_d}{dt} \right) + e_{bc} \quad (1)$$

$$V_{bc} = -V_d = -2R i_d - 2L \left( \frac{di_d}{dt} \right) + e_{bc} \quad (2)$$

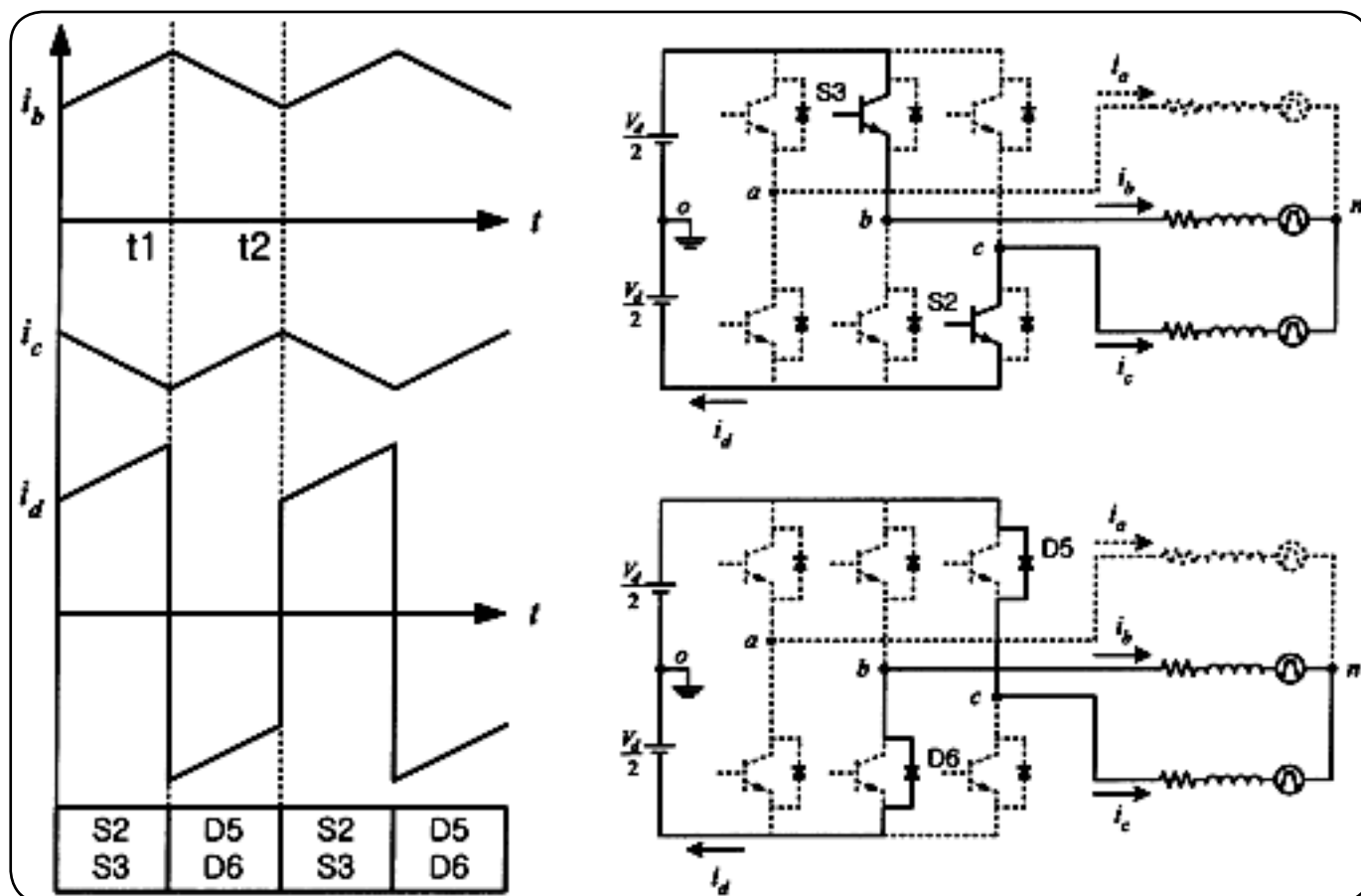
ترکیب (۱) و (۲) بوسیله معرفی وظیفه سوئیچ S معادله بازآرایی شده ولتاژ را نتیجه می دهد.

$$S \cdot V_d = 2R \cdot i_d + 2L \left( \frac{di_d}{dt} \right) + e_{bc} \quad (3)$$

جائیکه S علامتگذاری شده است ۱ برای سوئیچ روشن و ۱- برای سوئیچ خاموش و E اندازه back-EMF معنی می دهد. وظیفه سوئیچینگ S و جریان DC-link مطابق معادله تنظیم شده برای کنترل جریان به صورت ثابت بر پایه اندازه گیری جریان DC-link مشاهده شده است. قابل ملاحظه است در این معادلات که اطلاعات در مورد back-EMF ضروری است شامل همه شکل موج نمی شود. اما اندازه EMF شامل قسمت صاف بالای شکل موج بازگشت EMF می باشد. خروجی سیگنال کنترل شده و ثابت شده بیانگر:

$$V^* m(k) = k_1 \cdot |i_d(k-1)| + k_2 \cdot (i_d(k-1) - |i_d(k-1)|) + 2E(k-1) \quad (4)$$

که K1 و K2 و T3 و K مشخص کننده 2R و 2L/TS و نمونه گیری فوری در لحظه مورد نظر بترتیب انجام شود به معنی یک مرجع با ارزش می باشد. سیگنال خروجی استفاده شده برای یکنواخت کردن سیگنال در نمونه بلوک PWM و نتیجه برآیند در ولتاژ مربوط به استاتور را نشان می دهد.

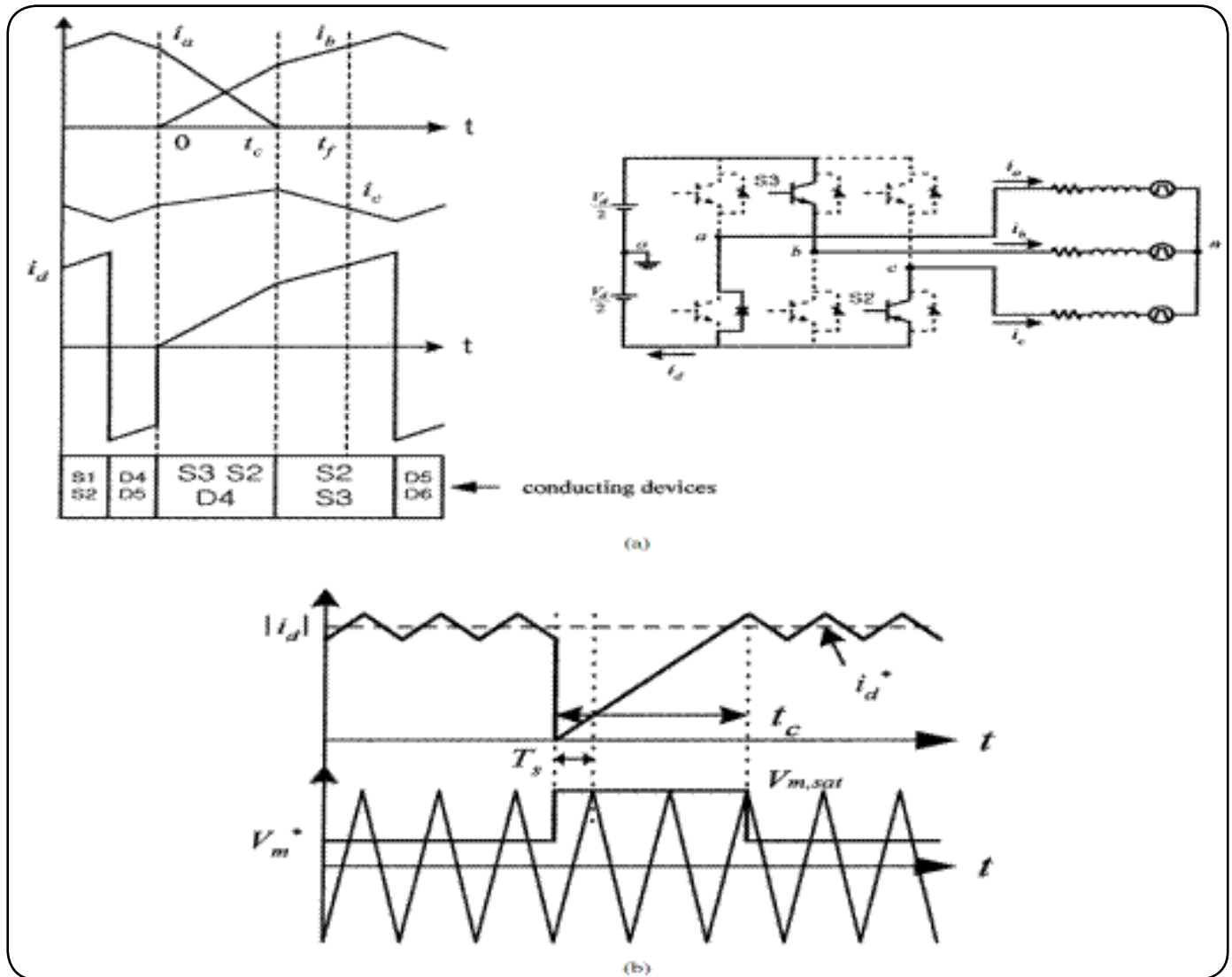


شکل ۳ عملکرد مد نرمال

## مد تغییرات

زمانیکه کنترل کننده معادله ثابت و بدون تغییرات در مد نرمال بکارگیری شده باشد بی درنگ آن چیزی که در مد تغییرات خواسته شده نتایج شکل موج آن شبیه به مد قبل می باشد که در شکل ۴ نمایش داده شده است. این شکل نشان دهنده این است که کنترل dc-link جریان نمی تواند در مدت زمان تغییرات دقیقاً کامل شده باشد. از DC-Link جریان نمی توان مانع کنترل بسته شدن مقدار جریان اصلی در داخل

نمونه گیری در مدت  $T_s$  متناظر با ترمینال و مقدار ولتاژ اصلی  $V_{xm}$  که به یک سطح اشباع حتمی می رسد شد. مطابق این نتایج هنگامیکه حادثه ای از قبیل یک حادثه اشباع رخ می دهد کنترل کننده جریان تحریک کننده تغییرات ریل گشتاور در مدت زمان مورد نظر را کنترل کرده است جریان تثبیت شده و پائین آمدن اعمال نفوذ آن در عدم تغییرات شکل موج فاز جریان بدیهی و متناظر با تغییرات ریل گشتاور بر روری گشتاور تولید شده می باشد. این مشکلات می تواند با استفاده از کاهش تغییرات ریل ارائه شده در این قسمت کاهش پیدا کند و متوقف شود. در این قسمت تثبیت جریان کنترل شده که مقدم بر سایر قسمت ها می باشد فراهم شده است. در اینجا کاهش تولید تغییرات ریل گشتاور باورکردنی نمی باشد. زیرا کنترل کننده اشباع در فاصله تغییرات به سر می برد. برای رفع این مشکل یک تکنیک کنترلی جبران کننده برای سیگنال خروجی اشباع شده پیشنهاد شده است و برای دو جریان تغییر یافته در جریان فازها این قسمت توضیح داده شده است.



شکل ۴ عملکرد مد کموتاسیون

## A. ناحیه سرعت پائین ( $V_d > 4E$ )

در شکل ۵ توالی سوئیچینگ و شکل موج مربوط به فازهای جریان در فاصله تغییرات در ناحیه سرعت پائین نمایش داده شده است. خطوط ناپیوسته شکل موج فاز جریان و الگو سوئیچینگ  $S3-S2-S1$  زمانیکه کنترل کننده جریان متوقف شده باشد را نشان می دهد که اشباع شده می باشد و در اول این قسمت توضیح داده شده است. یک روش پائین آوردن زمان در حال افزایش وارد شده جریان ورودی فاز  $i_b$  توضیح تکنیکی برای تغییرات نابرابر زمان های تغییرات دو فاز جریان می باشد. این خطوط ناپیوسته شکل موجهای مورد نظر در الگوی PWM را نشان می دهد که  $S3-S2-S1$  بوسیله توان نسبی  $D_{low}$  برای تغییرات خواسته شده در اینورتر تنظیم شده است. خروجی ولتاژ اینورتر کنترل شده به وسیله الگو PWM وظیفه برابر کردن  $S3-S2-S1$  در تنظیم توان موثر  $D_{low}$  در برابر کردن جریان ورودی فاز  $i_b$  و جریان خروجی فاز  $i_a$  را بر عهده دارد. ولتاژ فاز برابر شده در فاصله تغییرات بیانگر معادله ۵ و ۷ می باشد. که ولتاژ طبیعی بدست آمده در این تغییرات در مدت زمان این تغییرات برابر است با:

$$-V_d/2 = R.i_a + L \frac{di_a}{dt} + e_a + V_{no} \quad (5)$$

$$S.V_d/2 = R.i_b + L \frac{di_b}{dt} + e_b + V_{no} \quad (6)$$

$$-S.V_d/2 = R.i_c + L \frac{di_c}{dt} + e_c + V_{no} \quad (7)$$

$$V_{no} = -V_d/6 - (e_a + e_b + e_c)/3 \quad (8)$$

افزایش جریان فاز A می تواند بیانگر:

$$dia/dt = 1/L(-V_d/2 - e_a - V_{no}) = -(V_d + 2E)/3L \quad (9)$$

افزایش جریان فاز B مطابق سوئیچینگ محاسبه می شود.

$$dib/dt = 2(V_d - E)/3L, \text{ at } s=1 \quad (10)$$

$$dib/dt = -(V_d + E)/3L, \text{ at } s=-1 \quad (11)$$

معادله متناظر برای توضیح سوئیچینگ S در برقراری ۱ در فاصله  $D_{low} T_s$  و ۱- در فاصله  $(1 - D_{low}) T_s$  می تواند با استفاده از تکنیک حد متوسط فضای ناحیه مورد نظر مرتب شود.

$$dib/dt = V_d(3D_{low} - 1) - 2E/3L \quad (12)$$

هدف برابری جریان وارد شده به فاز با جریان خروجی از فاز در فاصله تغییرات می باشد. از این رو توان نسبی  $D_{low}$  به صورت زیر محاسبه می شود:

$$D_{low} = 2/3 + 4E/3V_d = 1/3(2 + 4E/V_d) \quad (13)$$

## B. ناحیه سرعت بالا ( $V_d > 4E$ )

مشابه در ناحیه سرعت بالا در شکل ۶ نمایش داده شده است یک روش کند پائین آوردن زمان خروجی جریان خروجی  $i_b$  استراتژی برابر ساختن تغییرات نا متناسب زمان دو تغییرات جریان فازها می باشد. با مراجع به شکل ۶ مطابق معادله می توان با کاهش رپل گشتاور در ناحیه سرعت بالا کنترل لازم را انجام داد بدین صورت که زمانیکه S می باشد و به صورت زیر بیان می شود:

$$sV_d/2 = R.i_a + Ldia/dt + e_a + V_{no} \quad (14)$$

$$V_d/2 = R.i_b + Ldib/dt + e_b + V_{no} \quad (15)$$

$$-V_d/2 = R.i_c + Ldic/dt + e_c + V_{no} \quad (16)$$

$$V_{no} = S.V_d/6 - (e_a + e_b + e_c)/3 \quad (17)$$

زمانیه ۱S- می شود معادلات به صورت زیر نوشته می شوند:

$$dia/dt = (V_d - 2E)/3L \quad (18)$$

$$dib/dt = (V_d - 2E)/3L \quad (19)$$

$$dia/dt = -(V_d + 2E)/3L \quad (20)$$

$$dib/dt = 2(V_d - E)/3L \quad (21)$$

از نتایج معادلات ۱۷ و ۲۰ تکنیک متوسط فضای مورد نظر به صورت زیر حاصل می شود:

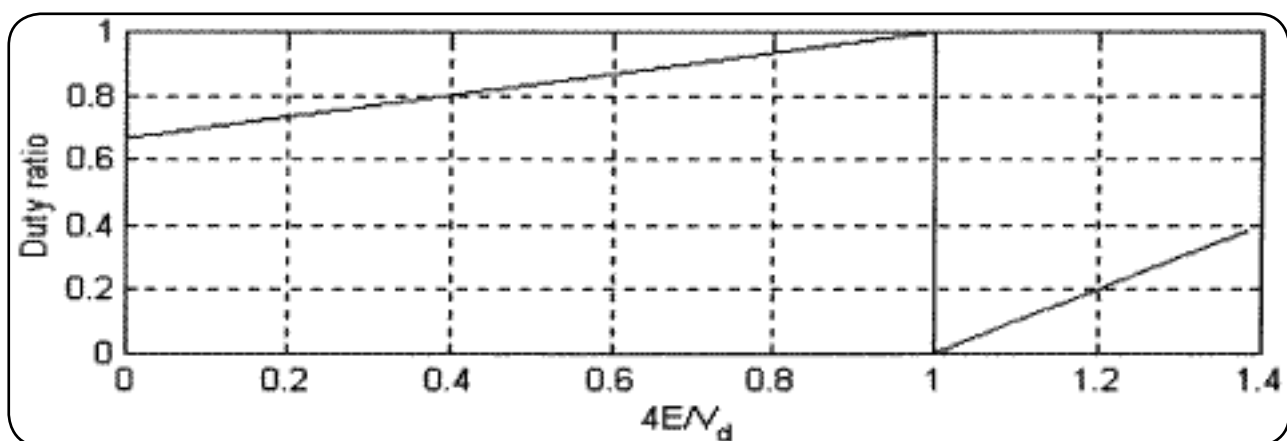
$$dia/dt = -V_d(2D_{high} - 1) - 2E/3L \quad (22)$$

$$dib/dt = V_d(2 - D_{high}) - 2E/3L \quad (23)$$

سرانجام نتایج بدست آمده بر حسب D به صورت زیر مرتب می شود:

$$D_{high} = (4E/V_d) - 1 \quad (24)$$

با جستجو و تامل در معادلات ۱۲ و ۲۳ نسبت تغییرات اینورتر در مدت زمان معین هیچ ارتباطی با اندوکتانس موتور ندارد. ارتباط میان نسبت توان موثر و بی ثباتی  $4E/V_d$  طرح ریزی شده بوسیله معادلات ۲۳ و ۲۴ می باشد که در شکل ۷ نمایش داده شده است.



شکل ۷ نسبت D برای جبران سازی کموتاسیون

## نتایج شبیه سازی

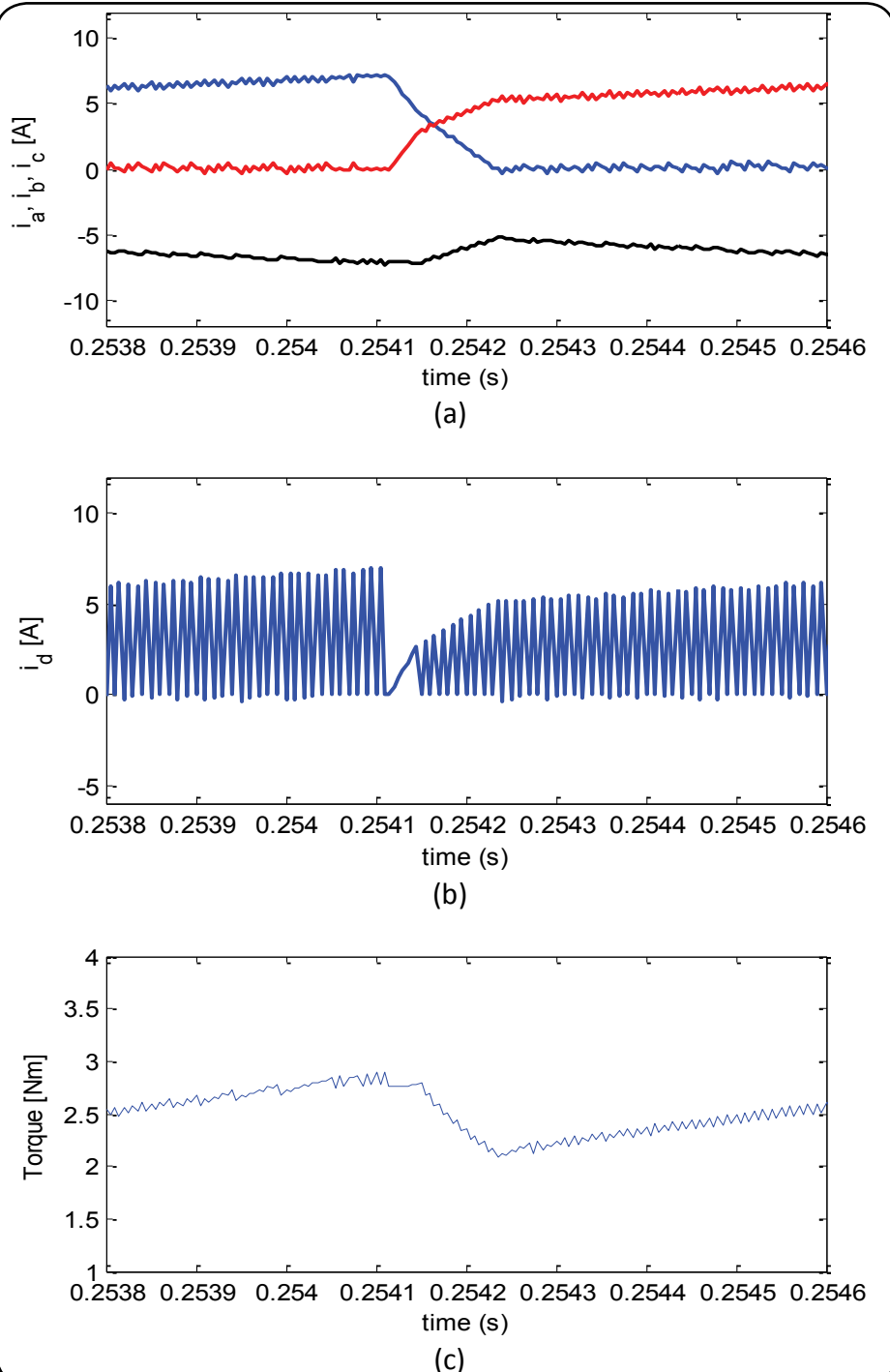
در این مقاله برای نشان دادن کاهش رپل گشتاور در موتور BLDC و همچنین کمتر شدن اعوجاج جریان در ناحیه کموتاسیون موتوری با مشخصات جدول ۱ انتخاب شده و در نرم افزار MATLAB شبیه سازی شده است.

در شکل ۸ جریان های سه فاز موتور در لحظه کموتاسیون کلیدها، جریان لینک DC اینورتر و گشتاور برای سرعت نامی و نصف بار کامل نشان داده شده است.



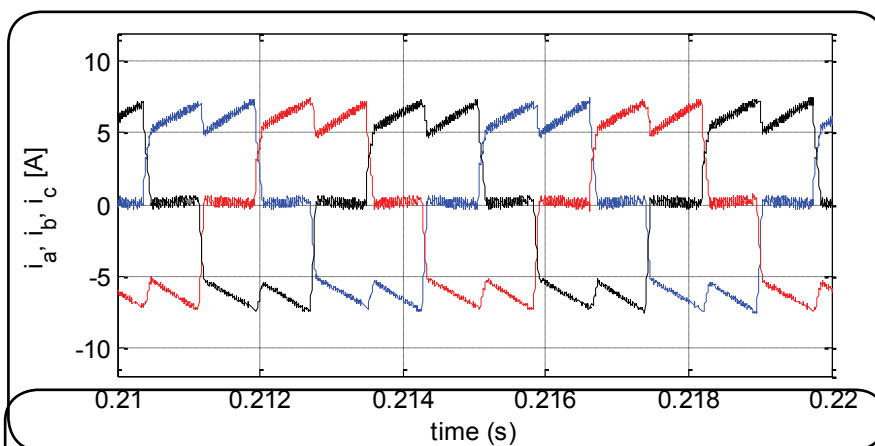
جدول ۱: پارامترهای موتور BLDC

۰.۳۸۸	مقاومت استاتور
۰.۰۰۱۷۲ H	اندوکتانس استاتور
۳۰۰ V	ولتاژ اینورتر
۰.۴	ثابت گشتاور
۰.۰۰۰۲	اینرسی موتور
۸	تعداد قطبها
۵ N.m	گشتاور نامی
۳۲۰۰ rpm	سرعت نامی

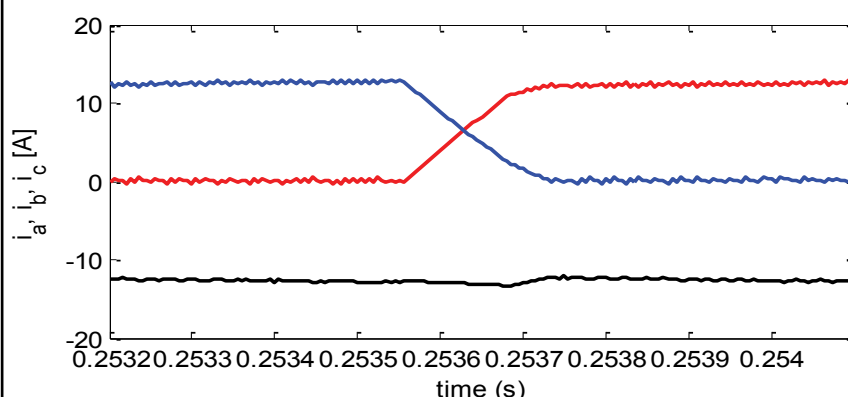


شکل ۸: تغییرات جریانهای موتور، جریان اینورتر و گشتاور در لحظه کموتاسیون برای نصف بار کامل موتور

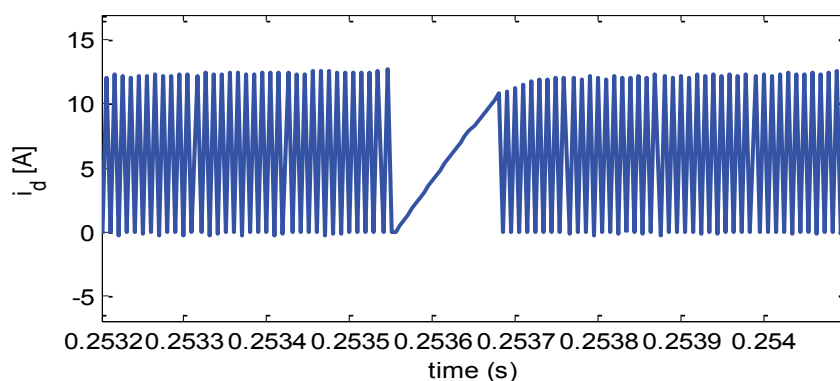
برای این حالت جریان های سه فاز موتور برای چند سیکل در شکل ۹ نشان داده شده است. طبق این شکل می بینیم که در بارهای کوچک رپل جریانه ها در لحظات کموتاسیون و بالتبع آن رپل گشتاور زیاد است.



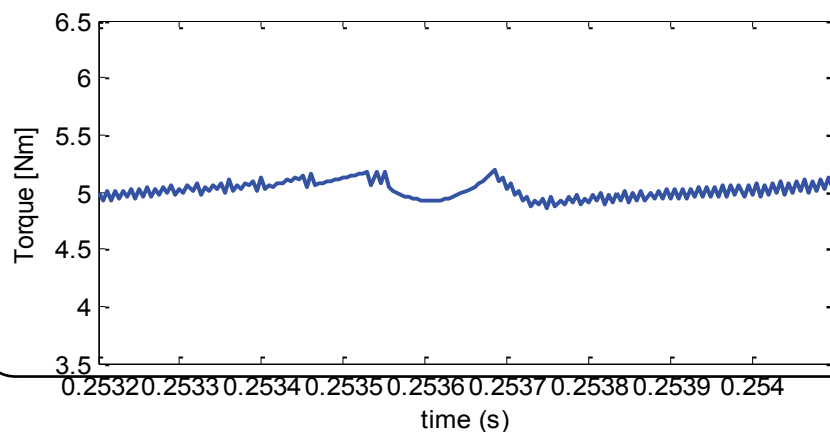
شکل ۹: جریانه های موتور برای نصف بار کامل



(a)



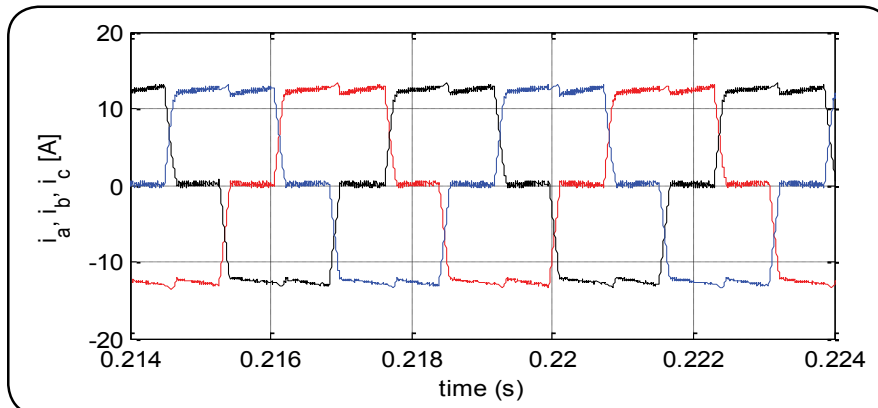
(b)



(c)

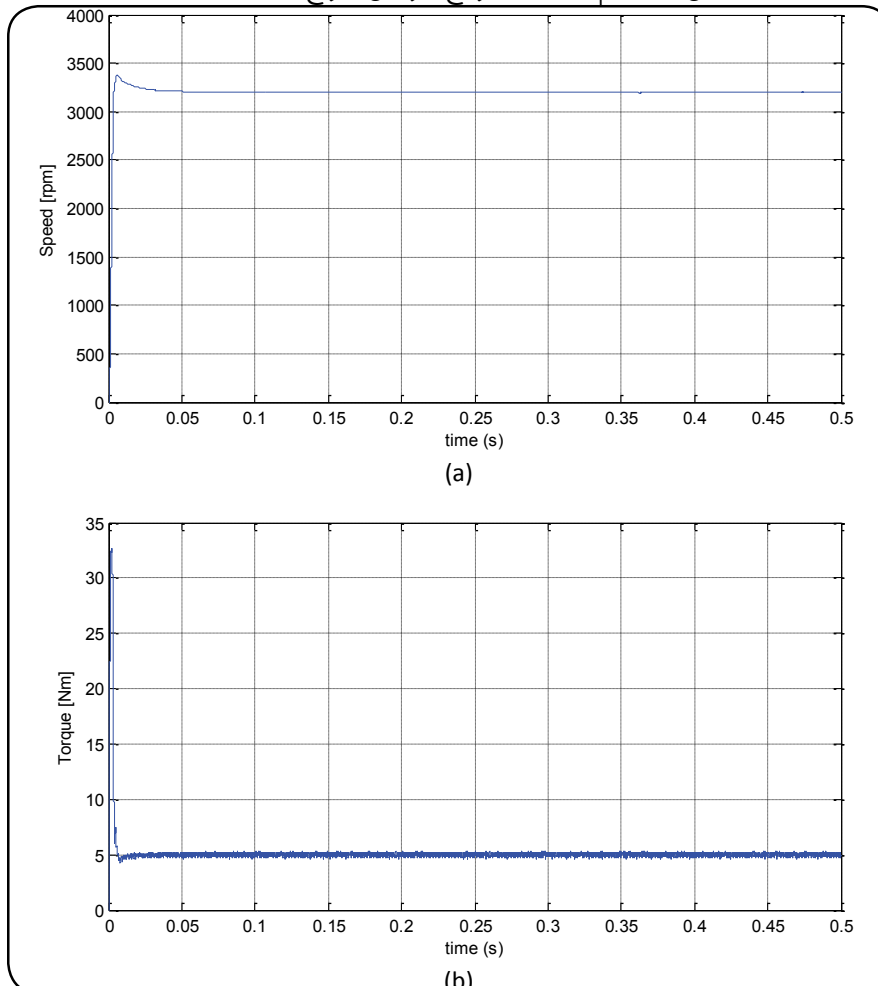
شکل ۱۰: تغییرات جریانه های موتور، جریان اینورتر و گشتاور در لحظه کموتاسیون برای بار کامل موتور

در شکل ۱۰ جریانهای سه فاز موتور در لحظه کموتاسیون کلیدها، جریان لینک DC اینورتر و گشتاور برای سرعت نامی و بار کامل ۵ N.m نشان داده شده است. برای این حالت نیز جریانهای سه فاز موتور برای چند سیکل در شکل ۱۱ نشان داده شده است. طبق این شکل می بینیم که در بار نامی رپل جریانها در لحظات کموتاسیون و بالتبع آن رپل گشتاور خیلی کم است.



شکل ۱۱: جریانهای موتور برای بار کامل

در آخر نمودار سرعت و گشتاور موتور در سرعت و بار نامی از لحظه راه اندازی موتور تا حالت ماندگار آن در شکل پایین نشان داده شده است. طبق این شکل سرعت دقیقاً با یک جهش اولیه کم به مقدار مرجع خودش سریع رسیده است که دارای زمان نشست خیلی کمتری است.



شکل ۱۲: سرعت و گشتاور موتور برای بار کامل

- منابع:
- 1-Joong-Ho Song, Member, IEEE, and Ick Choy "Commutation Torque Ripple Reduction in Brushless DC Motor Drives Using a Single DC Current Sensor" IEEE TRANSACTIONS ON POWER ELECTRONICS, VOL. 19, NO. 2, MARCH 2004
  - 2\_Jong-Woo Choi, Member, IEEE, and Sang-Cheol Lee"Antiwindup Strategy for PI-Type Speed Controller" IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 56, NO. 6, JUNE 2009
  - 3-K.-W. Lee, J.-B. Park, H.-G. Yeo, J.-Y. Yoo, and H.-M. Jo, "Current control algorithm to reduce torque ripple in brushless dc motors," in Proc. Int. Conf. Power Electron., 1998, pp. 380-385.

**فروشگاه تخصصی  
برق و الکترونیک**









[www.eShop.ECA.ir](http://www.eShop.ECA.ir)

**پذیرش نمایندگی فعال**  
از تمامی فروشگاه های برق و الکترونیک سراسر کشور

**فروشگاه تخصصی  
برق و الکترونیک ECA**

[www.eShop.ECA.ir](http://www.eShop.ECA.ir)

**مرکز دانلود ECA**

« مقالات علمی »  
« کنفرانس های داخلی »  
« سوالات کنکور »  
« پروژه های تکمیل شده »  
« و... »

[www.Download.ECA.ir](http://www.Download.ECA.ir)

**طراحی، تولید و واردات کلیه بردها و قطعات  
مربوط به سیستم های Embedded**

**ARM7/ARM9/ARM11  
(Atmel, Philips, Cortex-M3/  
A9/A15/...)**

[www.eShop.ECA.ir](http://www.eShop.ECA.ir)



وب سایت تخصصی برق و الکترونیک (ECA) افتخار دارد با بیش از ۱۱۵۰۰۰ عضو، یکی از بزرگترین وب سایت های تخصصی ایران را تشکیل داده و پاسخگوی نیاز دانشجویان، متخصصان، محققان و صنعتگران عرصه برق و الکترونیک کشور باشد. بی شک یکی از اهداف این انجمن، برقراری ارتباط بین صنعت و جامعه می باشد. لذا از تمام شرکت ها، کارخانجات و موسسات صنعتی علمی آموزشی دولتی و خصوصی، تقاضا مندیم در صورت تمایل به عقد قرارداد تبلیغاتی و یا قبول اسپانسری برای مجله تخصصی نویز از طرق زیر با ما در ارتباط باشند.

تلفن : ۰۴۱۱-۵۵۷۱۲۶۱  
فکس : ۰۴۱۱-۵۵۳۹۷۶۹  
ایمیل : [adver.eca@gmail.com](mailto:adver.eca@gmail.com)





## معرفی بخشی از تجهیزات مورد استفاده در سیستم های کنترلی مبتنی بر PLC

aminima2@gmail.com

تهیه کننده: امین شیخ نجدی

### ترانسمیترها و سنسورها

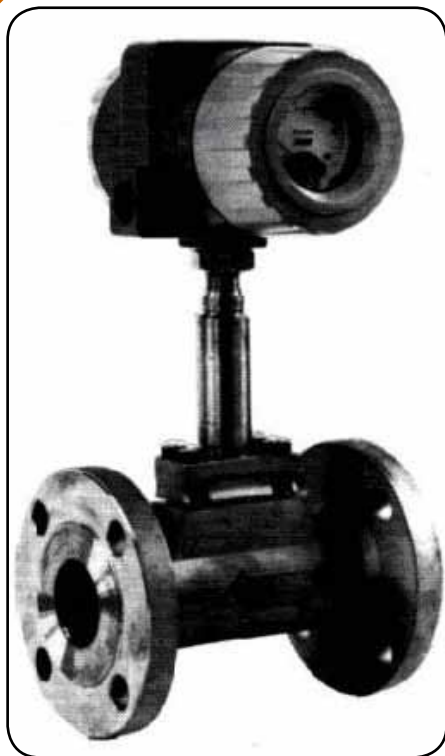
سنسورها و ترانسمیترها اجزای یک پروسه صنعتی هستند که کاربردهای فراوانی در پروسه های متنوع دارند. یک سنسور، قطعه ای است که به پارامترهای فیزیکی نظیر حرکت، حرارت، نور، فشار، الکتریسیته، مغناطیس و دیگر حالات انرژی حساس است و در هنگام تحریک آنها از خود عکس العمل نشان می دهد. ترانسمیتر نیز وسیله ای است که یک سیگنال الکتریکی ضعیف را دریافت کرده و به سطوح قابل قبول برای کنترلرها و مدارهای الکترونیکی تبدیل می کند، مثلاً یک حلقه فیدبک سیگنالی در سطح میکروولت یا میلی ولت یا میلی آمپر تولید می کند و این سیگنال ضعیف می تواند با عبور از ترانسمیتر به سیگنالی در سطوح صفر تا ده ولت و یا ۴ تا ۲۰ میلی آمپر تبدیل شود. ترانسمیترها عموماً از قطعاتی مثل op-amp برای تقویت و خطی کردن این سطوح ضعیف سیگنال استفاده می کنند.

وقتی یک سنسور همراه یک ترانسمیتر باشد، سنسور پارمتر را اندازه می گیرد و مقدار تعیین شده را به ترانسمیتر تحویل می دهد، سپس ترانسمیتر آن را به یک سیگنال الکتریکی قابل درک برای کنترلر و صد البته قابل ارسال توسط سیم های فلزی، تبدیل می کند.

دما:

دما از پارامترهای بسیار مهم در پروسه های صنعتی است. در حقیقت کمتر پروسه کنترلی بدون کنترل دما میتوان در نظر گرفت. این پارامتر معمولاً در ایران با مقیاس درجه سانتیگراد اندازه گیری میگردد. و با توجه به رنج دمائی و نوع کاربرد از روشهای گوناگونی برای اندازه گیری آن استفاده میگردد. این روشهای را میتوان به صورت تماسی و غیر تماسی دسته بندی کرد. روش تماسی در حقیقت از تماس یک سنسور که میتواند ترموکوپل یا ترمورزیستنس (PT100) باشد استفاده کرد. این روش اندازه گیری در مدلهای آنالوگ بیشتر به روش بیمتال یا Gas-filled که از یک حباب و لوله موئین درست شده است انجام میشود.

روش اندازه گیری غیر تماسی بیشتر بر اساس مقدار تابش فرسرخ که جسم هدف از خود به محیط ساطع میکند استفاده میشود این دماسنج ها به پیرو متر یا ترمومتر تفنگی یا لیزی مشهور هستند. دلیل استفاده از واژه لیزری استفاده از باریکه نور لیزر برای مشخص شدن هدف می باشد.



**Pressure trasmitter**

ترانسمیتر فشار یکی از پر کاربرد ترین تجهیزات و ادوات ابزار دقیق در کنترل و مانیتورینگ پروسه های مختلف صنعتی است. این تجهیز علاوه بر این که در کنترل فشار و نمایش مقادیر کاربرد دارد می تواند به عنوان ترانسمیتر مقدار سطح مایعات و همچنین در مدل های دیفرنسیالی که همان ترانسمیتر اختلاف فشار نام دارد در مانیتورینگ فلو و فلومتری به کار می آید. این تجهیزات در مدل های قلمی ارزان قیمت و همچنین نمونه های اسمارت با پروتکل هارت ارائه می گردند.





## Inductive Proximity Switch

کلید های مجاورتی القائی به دسته ای از پراگسیمیتی سوئیچها گفته میشود که به صورت کلیدهای غیر مکانیکی در اثر مجاورت با فلزات تحریک گردیده و حضور شیء را حس میکند. این کلیدها از پر مصرف ترین نمونه کلیدهای مجارتي بوده و در صنایع و کاربردهای مختلف مورد استفاده قرار میگیرد. برای انتخاب این نمونه سنسور القائی بایستی به فاصله سنس، قطر سنسور، نوع خروجی، تعداد خروجی و نوع تغذیه توجه داشت.



## photoelectric sensor

سنسور نوری الکترونیکی از پرتو نور برای تشخیص و آشکارسازی حضور اشیاء استفاده میکنند. این تکنولوژی در بعضی موارد جایگزین ایده آلی برای سنسورهای پراگسیمیتی در مواقعی که فاصله حسگر از شی قابل توجه باشد یا جنس آن غیر فلز باشد. این سنسورهای نوری معمولاً به دسته های Diffusion, Reflection, Through beam می گردند.



## Capacitance Sensor

سنسورهای خازنی نظیر دیگر سنسورهای غیر تماسی با نزدیک شدن به اشیاء خصوصاً غیر فلزات خروجی سوئیچ ترانزیستور PNP و یا NPN می دهد. با نزدیک شدن اجسام به سنسور خازنی دی الکتریک خازن تعبیه شده در سنسور تغییر پیدا کرده و مدار داخلی با مقایسه به مقدار مرجع یک خروجی باز یا بسته میدهد. از سنسور خازنی می توان به عنوان مولد پالس به منظور کنترل وضعیت برنامه ماشین آلات برای شمارنده ها و آشکارسازی تقریباً تمام مواد فلزی و غیر فلزی استفاده کرد. برای مثال در کنترل سطح مواد پودری یکی از پر کاربردترین سنسور

ها، سنسور خازنی است. سنسورهای خازنی در مدل های مکعبی و استوانه ای تولید می گردد و بسته به مکانیزم تولید و ابعاد می تواند در فواصل مختلف وجود اجسام را حس کند

## Laser Sensor

سنسورهای لیزری جزء سنسورهای نوری محسوب میگردند ولی از پرتو نور لیزر جهت تشخیص جسم و یا حتی فاصله دقیق آن استفاده میکند. این سنسورهای در انواع گوناگون نظیر Through beam laser sensor, Reflective و Through beam نیاز به یک قسمت فرستنده و یک قسمت گیرنده میباشد و در مدل Reflection به کمک یک قسمت آینه شکل پرتو لیزر به فرستنده باز تابانیده میشود و در مدل های دیگر قسمت فرستنده و گیرنده در یک تجهیز قرار گرفته اند. از سنسورهای لیزری در بسیاری از موارد که فاصله قابل توجه بین سنسور و جسم به همراه گرد و غبار و یا شرایط بد محیطی وجود دارد میتوان استفاده نمود.



## Magnetic Switch

سنسورهای مگنت زمانی که در میدان مغناطیسی قرار میگیرند عمل میکنند و خروجی آن به صورت نرمال کلوز یا این و در بعضی مدل های با قابلیت ماندن در آخرین وضعیت میباشد یعنی با عبور از کنار میدان مغناطیسی با تغییر وضعیت خواهند داشت از بسته به باز و یا بالعکس ولی بعد از دور شدن از میدان در وضعیت آخرین وضعیت خود خواهند ماند. مگنت سوئیچها مانند دیگر پراگسیمیتی سوئیچها یا سنسورهای مجاورتی در دسته Non contact و یا غیر تماسی قرار میگردند که این خاصیت باعث افزایش طول عمر و بازده مگنت سوئیچ میشود و نویزهای حاصل





از لغزش قسمت های متحرک برهم، نیز در مگنت سوئیچ ها حذف شده است. سنسور های مغناطیسی به دلیل دارا بودن ساختار قابل اطمینان، در محیط های آلوده صنعتی، چرب و روغنی بخوبی عمل میکنند و به همین علت در اتومبیل و کاربرد های مشابه بسیار مفید هستند



## فلومتر

فلومتر یا دبی سنج وسیله ایست که حجم مواد عبوری را نسبت به زمان معرفی میکند. در بیشتر صنایع از جمله صنایع نفت و پتروشیمی و... دانستن مقدار دقیق فلوی عبوری سیالی مانند گاز و یا نفت حیاتی است. توجه به این نکته که میزان کل فلو که معرف جرم ماده مصرفی است در بسیاری موارد نظیر پروسه های صنعتی کاربرد دارد. همچنین اندازه گیری دقیق فلو به دلیل ملاحظات مالی در فروش و خرید حجم مشخصی از ماده و کالا مانند کنتور گاز خانگی و تبدلات نفتی و... اهمیت ویژه ای دارد. تکنیکهای مختلفی برای اندازه گیری فلو به کار می رود که میتوان به پاره از متداولترین آنها در زیر اشاره کرد.

روش جابجائی مثبت Positive Displacement

اوریفیس پلیت Orifice Plate

روش توربینی Turbine

روتامتری (variable area)

آلتراسونیک با پدیده دوپلر Ultrasonic: Doppler

آلتراسونیک با روش زمان انتقال و... Ultrasonic: Transit Time, Time of Travel, Time of Flight

ونتوری Orifice Plate

مغناطیسی Magnetic

ورتکس Vortex Shedding

روتامتر (Variable Area) Rotameter

فلومتر مخصوص کانالهای باز Weir and Flume Open channel

کوریالیس Coriolis

گرمائی Thermal

پیتوت Pitot Tube



## Level meter



سطح سنجی یا اندازه گیری ارتفاع سطح سیال مورد اندازه گیری از پائین مخزن تا گاز یا بخار یا ماده دیگری که روی آن را پوشانده است تعریف می گردد. نمونه های بسیاری از این سطح سنجی و کنترل سطح مایعات حتی در زندگی روزمره استفاده می گردد نظیر سطح مخزن بنزین وسیله نقلیه کولر آبی و... در زیر به تعدادی از تکنیک های رایج ارتفاع سنجی سیالات در صنعت اشاره گردیده است که با توجه به نوع سیال، فشار آن، دما، و شرایط دیگر از قبل چند فازی بودن، و نوع خروجی تجهیز و... از روشهای سطح سنجی مختلف میتوان استفاده کرد :

روش سایت گلاس

مدل شناوری Floater

تجهیزات جابجا شونده (displacement)

ترک تیوب

استفاده از روش اندازه گیری فشار و اختلاف فشار

روش لوله حباب سازها یا سیستم purge

سطح سنج آلتراسونیک

ارتفاع سنج راداری

لول متر خازنی

## Ultrasonic Sensor

سنسور آلتراسونیک یا ماوراء صوت یکی دیگر از سنسورهای غیر تماسی و مجاورتی یا پراگسیمیتی می باشد در کاربردهای گوناگون آشکار سازی اجسام تا اندازه گیری فاصله یا سطح سنجی به کار میرود. به طور معمول سنسورهای آلتراسونیک با ارسال یک پالس صوتی کوتاه در فرکانس فراصوت به سمت هدفی که این پالس را منعکس می کند و دریافت و شناسائی این امواج به شکل یک ترانسپور عمل کرده و در مدلهایی که فاصله را محاسبه می کنند با اندازه گیری اختلاف زمانی ارسال و دریافت پالس می توانند به فاصله یاب تبدیل شوند. سنسور آلتراسونیک را در بازار به شکلهای گوناگون و برای کاربردهای مختلف میتوان یافت. سنسورهائی با نحوه مختلف نصب، پیکربندی، IP و فرکانس متفاوت. البته سنسور ها و ترانسمیتر های مختلف دیگری نیز وجود دارد که در اینجا امکان معرفی تک تک آنها نیست.



## Humidity Transmitter

ترانسمیتر رطوبت برای ارسال اطلاعات مربوط به رطوبت و نقطه شبنم به اتاق کنترل یا PLC یا جهت رکورد گیری و کنترل مورد استفاده می گیرد. خروجی ترانسمیتر رطوبت بسته به انتخاب می تواند خروجی میلی آمپر و یا ولتاژ باشد و با توجه به نیاز و مورد اندازه گیری رنج ۰ تا ۱۰۰ درصد را در بر می گیرد. همچنین با توجه به حساسیت اندازه گیری رطوبت دقت دستگاه می تواند متفاوت باشد. با توجه به این در اندازه گیری رطوبت به اندازه گیری دما نیاز می باشد معمولا ترانسمیترهای رطوبت دارای ورودی سنسور دما نیز می باشند.



## Load cell

از این ترانسدیوسر برای اندازه گیری وزن استفاده می شود.



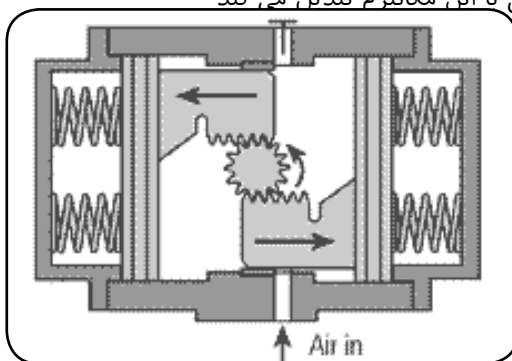
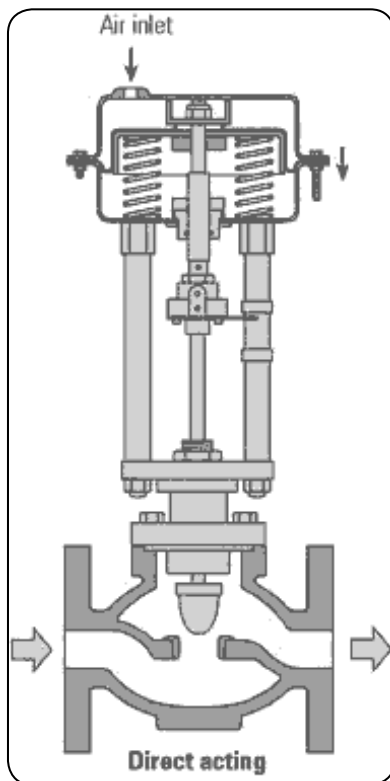
## Limit switch

برای آگاهی از رسیدن یک وسیله مکانیکی به ابتدا یا انتها ی مسیر حرکت خود از limit switch ها استفاده می شود.



## Cotrol valve Actuator

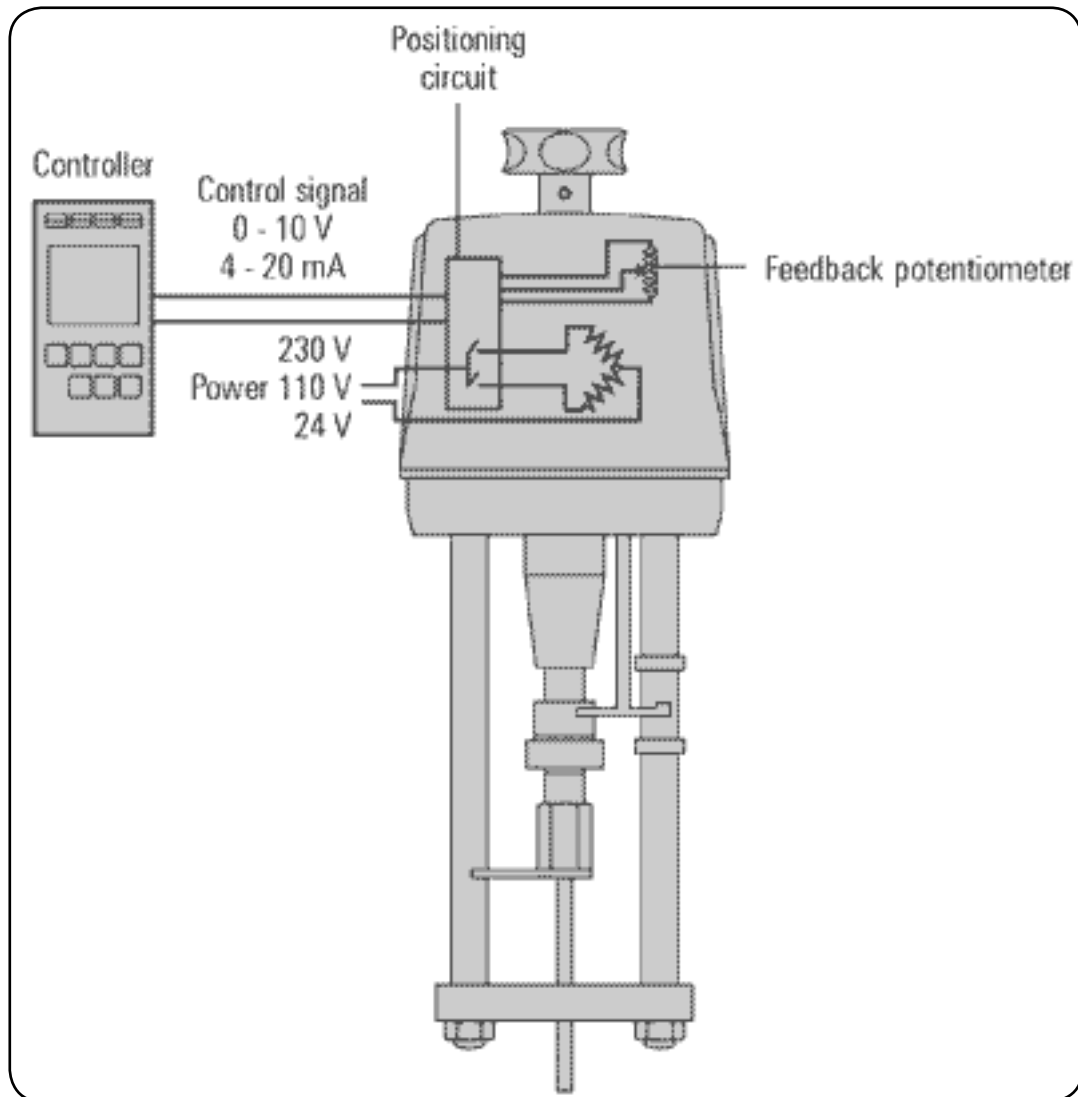
اکچوئیتور یا عمل کننده در یک لوپ کنترلی به صورت ساده به ابزار اطلاق می شود که یکی از صورتهای انرژی را به حرکت از نوع دورانی یا طولی تبدیل می کند و باعث اعمال نیرو می شود به عبارت دیگر اکچوئیتور تجهیز الکتریکی یا پنوماتیکی یا هیدرولیکی یا دستی است که انرژی را تبدیل به حرکت دورانی یا طولی می کند. در کنترل ولو یا شیر کنترل شما برای به حرکت در آوردن ساقه ولو و باز و بسته کردن آن به صورت خودکار و از راه دور از طریق یک لوپ کنترل نیاز به یکی از انواع این عمل کننده ها خواهیم داشت. برای نمونه رایجترین نوع عمل کننده یا اکچوئیتور پنوماتیک و از نوع دیافراگمی است که از قدیمی ترین، ساده ترین و ایمن ترین نمونه های اکچوئیتور بوده و انرژی پنوماتیک ناشی از پالس هوایی که معمولا ۳ تا ۱۵psi است را با هدایت پشت دیافراگم به حرکت طولی برای باز و بسته کردن گلاب ولو یا شیرهایی با این مکانیزم تبدیل می کند



از دیگر نمونه های عمل کننده های پنوماتیکی میتوان به مدل های روتاری یا گردنده برای کنترل ولوهای با محور گردنده برای مثال شیر پروانه ای یا تویی نام برد. عمل کننده های پیستونی نیوماتیکی نیز گونه ای از عمل کننده های پنوماتیکی مشابه به دیافراگمی می باشند.

در نمونه های دیگر شاهد استفاده از عمل کننده های الکتریکی و موتوری هستیم که با برق مستقیم یا متناوب تک فاز یا سه فاز و به کمک جعبه دنده طراحی شده انرژی الکتریکی را به دورانی یا گشتاور متناسب و تحت زاویه ای که برای باز و بسته کردن ولو نیاز می باشد تبدیل میکند. عمل کننده های الکتریکی بسته به فیدبکی که از وضعیت باز و بسته بودن ولو می گیرند و مکانیزم طراحی می توانند حالت های باز یا بسته تنها یا شرایط بینابین آن مورد استفاده قرار گیرند. در نمونه های باز و بسته از دو لیمیت سوئیچ برای کنترل موقعیت باز و بسته و جلوگیری از فشار بیش از حد به ولو استفاده می کنند و در نمونه های تدریجی از فیدبکی که از یک پوزیشنر پتانسیومتری میگیرند میتوانند کنترل صحیحی بر موقعیت والو داشته باشند.





**Pneumatic Actuator**

عمل کننده های پنوماتیکی و بادی جزء لاینفک سیستم های اتوماسیون امروزی می باشند. در بسیاری موارد در صنعت نیاز به جابجایی و انتقال نیرو از طریق باد میباشد که این کار وظیفه این اکچوئیورهای می باشند.



## PC Base Recorder



رکورددهای پی سی بیس از یک کارت خروجی یا اکسترنال و کابل رابط به همراه کامپیوترهای معمولی یا لپ تاپ به علاوه نرم افزار مربوطه تشکیل شده اند. در نمونه های پیشرفته تر این رکوردرها شما با آدرس دهی به هر کارت یا ماژول میتوانید به کانال های ورودی افزوده یا حتی خروجی های آلارم بگیرید. در این حالت شما بایستی کارتها را به کمک کابل مربوطه به صورت شبکه به هم متصل کنید و البته در آن صورت پروتکل صنعتی مشخصی نیز به کمک شما خواهد آمده تا به صورتی کاملاً اقتصادی یک سیستم رکوردینگ یا ثبت اطلاعات راه اندازی کنید.

## Multifunction Calibrator



کالیبراتورهای مولتی فانکشن در حقیقت برای کالیبره چندین پارامتر نظیر دما، جریان، ولتاژ و مقاومت طراحی گردیده اند به صورتی که قابلیت کالیبره تمام سنسورهای رایج دما نظیر انواع ترموکوپل و RTD را داشته و همچنین می توانند انواع ترانسیمترهای فشار و اختلاف فشار را کالیبره کنند.

## نرم افزاری های جانبی و مرتبط با SIMATIC Manager

A7 SCL: زبان برنامه نویسی سطح بالا می باشد که با زبان ST ذکر شده و در استاندارد IEC1131-3 تطبیق دارد و برای PLC های S7-300 و S7-400 و C7 به کار می رود. قابل ذکر است که این نرم افزار در نسخه Professional Step7 موجود می باشد.

HiGraph SV: برای کنترل ترتیبی ( برای مطالعه بیشتر در رابطه با کنترل می توانید به مطلبی در مورد مقدمه ای بر کنترل مراجعه کنید) بصورت گرافیکی، با ابزارهای پیشرفته و در PLC های S7-300 و S7-400 و C7 به کار می رود.

S7 GRAPH: برنامه نویسی بصورت گرافیکی است که برای کنترل ترتیبی به کار می رود و با زبان SFC مندرج در استاندارد IEC1131-3 تطبیق دارد و برای PLC های S7-300 و S7-400 و C7 به کار می رود. این نرم افزار در نسخه Step7 Professional موجود می باشد.

S7 PLCSIM: سیمولاتور نرم افزاری است برای تست برنامه زمانی که PLC در دسترس

نیست به کار می رود. این نرم افزار در نسخه Step7 Professional موجود می باشد.

CFC : توسط این نرم افزار برنامه نویسی بصورت گرافیکی توسط یکسری بلوکهای از پیش تعیین شده طراحی و انجام می شود. این نرم افزار را باید جدا گانه تهیه کرد و برای S-300، S7-400 و F/H Systems کاربرد دارد.

S7 PDIAG : ابزاری عیب یابی است که برای PLC های S7-300 و S7-400 به کار می رود. این نرم افزار در نسخه Step7 Professional موجود می باشد.

TeleService : برای ارتباط با plc از طریق خط تلفن به کار می رود. وقتی که plc توسط آداپتور خاص (tc) به مودم متصل باشد. با استفاده از کامپیوتر بصورت Remot می توان انرا از هر نقطه ای برنامه نویسی و رفع عیب کرد.

DOCPRO : برای مستند سازی بکار می رود با استفاده از آن می توان پس از اتمام پیر بندی و برنامه نویسی نقشه های Warning و متن برنامه را با فرمت تهیه و چاپ کرد.

Standard PID Control : ابزار کمکی برای طراحی کنترل کننده های PID است که برای PLC های S7-300 و S7-400 و C7 به کار می رود.

Fuzzy Control : برای کنترل قازی است و در مواردی بکار می رود که توصیف ریاضی پروسه مشکل یا ناممکن باشد. در برخی موارد ترکیب این روش با لوپ های pid نتیجه بهینه را برای کنترل سیستم به همراه دارد.

Modular PID Control : ابزاری است که برای طراحی لوپ های کنترل پیچیده به کار می رود و داری فانکشن ها و بلوک های از قبل طراحی شده می باشد.

Neurosystem : شبکه عصبی مورد استفاده در سیستم کنترل را می توان با این ابزار طراحی کرد و آموزش داد.

PRODAVE MPI : برای پردازش ترافیک دیتا در شبکه MPI بین سیستم های S7 و M7 و C7 به کار می رود.

SIMATIC ProTools : ابزار پیکر بندی است که برای سیستم های کنترل اپراتوری و بخش مانیتورینگ مربوط به C7 به کار می رود.

SIMATIC Wincc : نرم افزاری برای طراحی سیستم مانیتورینگ است.

مراجع :

[www.esud83.mihablog.com](http://www.esud83.mihablog.com)

[www.abzaran.com](http://www.abzaran.com)

[www.nicmehr.com](http://www.nicmehr.com)

[www.raymonco.com](http://www.raymonco.com)

[www.farasoniro.com](http://www.farasoniro.com)

[www.yektavara](http://www.yektavara)

## Multi AVR Programmer



این پروگرامر علاوه بر استفاده از تکنولوژی USB جهت اتصال به رایانه دارای امکانات و ویژگی های دیگری بوده که می تواند محدوده وسیعی از احتیاجات شما را برآورده سازد. امکان تست LCD، پروگرام کردن میکرو های معیوب، رنج وسیع از میکرو های قابل برنامه ریزی، هماهنگی با انواع سیستم عامل ها و..... از امکانات ویژه این پروگرامر است که در ادامه تمامی آنها معرفی می شوند.

- بدون نیاز به تغذیه خارجی
- مجهز به ۲ پورت USB جهت برطرف نمودن کمبود جریان احتمالی پورت کامپیوتر
- مجهز به کانکتور ISP خروجی
- پروگرام کردن تمام میکروها فقط از طریق یک ZIF سوکت
- مولد پالس ساعت جهت بازیابی میکروهایی که فیوز بیت آنها به اشتباه تغییر داده شده است.
- سازگاری با سیستم عامل های (Linux / Mac OS X / Windows)
- پشتیبانی از تمامی میکروکنترلرهای خانواده AVR
- پشتیبانی از میکروکنترلرهای سری AT89Sxx
- پشتیبانی از حافظه های سریال سری 24Cxx
- توانایی تست LCD های کاراکتری

توضیحات تکمیلی :

<http://eshop.eca.ir/link/489.php>



# اشتراک مجله نویز



**www.NoiseMagazine.ECA.ir**  
**www.Noisemagazine.ECA.ir**

## اشتراک نسخه چاپ شده مجله نویز

لث اشتراک ۶ شماره مجله نویز ۱۷۰۰۰ تومان  
 لث اشتراک ۱۲ شماره مجله نویز ۳۲۰۰۰ تومان

\*لث اشتراک مجلات در تعداد بالا برای دانشگاه ها و مراکز علمی، آموزشی دارای تخفیف ویژه می باشد.

لطفا پس از واریز مبلغ، اطلاعات خود (نام-آدرس پستی-شماره تماس) را به آدرس ایمیل [noisemagazine.eca@gmail.com](mailto:noisemagazine.eca@gmail.com) ارسال نمایید تا نسبت به ارسال نسخه پرینت شده مجله نویز اقدام شود.

شماره حساب ها و درگاه پرداخت الکترونیک: <http://eca.ir/payment>





## پروتکل ارتباطات سریال

در ارتباطات سریال، بین فرستنده و گیرنده تنها یک سیم داده (و یک سیم زمین) وجود دارد. فرستنده بیت های داده را به صورت پشت سر هم (سریال) روی این سیم قرار می دهد و در سمت گیرنده نیز بیت ها یک به یک دریافت شده و کنار هم قرار می گیرند. با توجه به اینکه برای ارتباطات سریال تنها یک جفت سیم ضروری است، این نوع ارتباطات برای تجهیزات مخابراتی و ارتباطات راه دور کاربرد وسیعی دارد. مثلاً ارتباطات در شبکه (با کابل CAT-5 در فواصل چند متری) و اینترنت (خط تلفن در فواصل چند کیلوتری) از این نوع ارتباط استفاده می کنند (تصور کنید اگر ارتباطات اینترنتی با اتصال موازی انجام می شد، تمام معادن مس دنیا هم برای تامین این همه سیم کافی نبود!).

در پروتکل ارتباطات سریال چند موضوع مهم باید مورد توجه قرار گیرند:

- نحوه هماهنگی سرعت سریال و دریافت داده ها بین فرستنده و گیرنده.
- نحوه ی قاب بندی اطلاعات (نحوه مشخص کردن آغاز و پایان واحد اطلاعاتی، طول واحد اطلاعاتی و...)
- مکانیسم کشف خطا
- نحوه کنترل جریان داده

### هماهنگی سرعت بین فرستنده و گیرنده

ارتباطات سریال را می توان مانند نوار نقاله ای در نظر گرفت که یک کارگر در یک سمت آن، جعبه هایی را روی نقاله قرار می دهد و کارگری در سمت دیگر باید جعبه ها را از روی نقاله بردارد. بدیهی است سرعت کار هر دو کارگر باید یکسان باشد. اگر کارگر اول با سرعت بیشتری کار کند، کارگر دوم فرصت برداشتن جعبه ها از روی نقاله را پیدا نمی کند. اگر سرعت کارگر اول کمتر باشد، کارگر دوم که سریعتر کار می کند بارها به نقاله مراجعه می کند بدون اینکه جعبه ی جدیدی رسیده باشد.

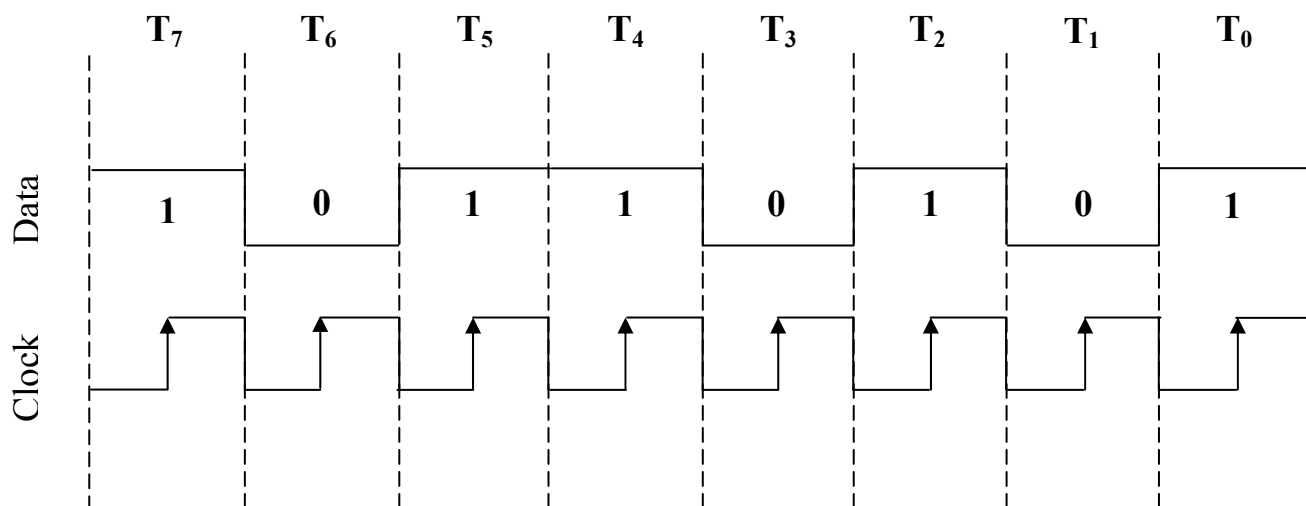
در ارتباطات سریال، چون فرستنده داده ها را بیت به بیت روی خط می گذارد و گیرنده نیز بیت ها را یکی یکی از روی خط برمی دارد، برای یک ارتباط بی نقص باید سرعت فرستنده و گیرنده یکسان باشد. وگرنه داده های ارسالی به درستی در گیرنده دریافت نمی شوند. بنابراین یکی از موارد مهم در برقراری یک ارتباطات سریال، توافق بر سر سرعت ارتباطات بین فرستنده و گیرنده است.

سرعت ارتباطات سریال با واحد بیت بر ثانیه (bps) سنجیده می شود و نرخ بیتی نامیده می شود.

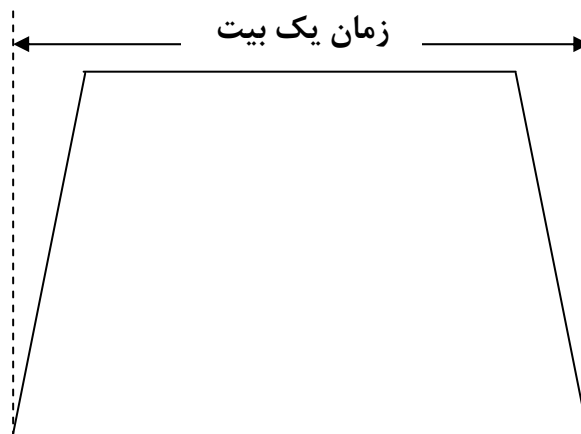
پورت سریال کامپیوتر از سرعت های مختلفی مانند ۱۱۰، ۳۰۰، ۲۴۰۰، ۹۶۰۰ و... تا ۹۲۱۶۰۰ بیت در ثانیه حمایت می کند. بر اساس نحوه هماهنگی بین فرستنده و گیرنده بر سر سرعت انتقال داده ها، دو نوع ارتباطات سریال همگام و غیر همگام تعریف می شود.

## ارتباط سریال همگام

مثال قبلی در مورد نوار نقاله و دو کارگر را در نظر بگیرید. دیدیم که سرعت کارگر اول در قرار دادن جعبه ها باید با سرعت کارگر دوم در برداشتن جعبه ها یکسان باشد. یک روش برای همگام کردن دو کارگر این است که کارگر اول چند لحظه پس از قرار دادن جعبه روی نقاله، زنگی را به صدا درآورد تا کارگر دوم را متوجه کند. کارگر دوم نیز با شنیدن صدای زنگ متوجه می شود که وقت برداشتن جعبه فرا رسیده است. در روش ارتباط سریال همگام، فرستنده به همراه داده ها، یک سیگنال ساعت (CLK) نیز ارسال می کند. در مواقعی که گیرنده باید خط را بخواند، فرستنده یک لبه بالارونده روی سیگنال ساعت ارسال می کند. گیرنده به سیگنال ساعت توجه می کند و هرگاه لبه ی بالا رونده ای روی آن باشد، خط داده ها را خوانده و بیتی که روی آن است را ثبت می کند. شکل بعد این نوع ارتباط را نشان می دهد. پیشتر در مورد اصلی کامپیوتر یک پورت ۲۵ پین سریال وجود داشت که از این روش برای ارتباط استفاده می کرد. ارتباط صفحه کلید با کامپیوتر نیز از این نوع است. در حال حاضر دو استاندارد ارتباط سریال همگام به نام های BISYNC و SDLC وجود دارد.



همانطور که در شکل بالا دیده می شود، فرستنده لبه بالا رونده را در (وسط) زمانی که بیت روی خط است روی سیگنال ساعت قرار می دهد. شکل زیر دلیل این موضوع را نشان می دهد:



همانطور که دیده می شود، بیشترین پایداری ولتاژ بیت روی خط داده در (وسط) زمان قرار داشتن بیت روی خط داده است.

## ارتباط سریال غیر همگام

در این نوع ارتباط، دیگر خط CLK بین فرستنده و گیرنده وجود ندارد. بنابراین تنها راهی که برای تطابق سرعت بین فرستنده و گیرنده می ماند این است که قبل از آغاز ارتباط، نرخ داده بین فرستنده و گیرنده مورد توافق قرار گیرد. این نوع ارتباط سریال کاربردهای فراوانی دارد. پورت COM کامپیوترهای فعلی که ۹ پین دارد، پورت سریال میکروکنترلر ها، ارتباط دستگاه کنترل از راه دور با تلویزیون و... از جمله این کاربردها هستند.

## قاب بندی در ارتباط سریال همگام

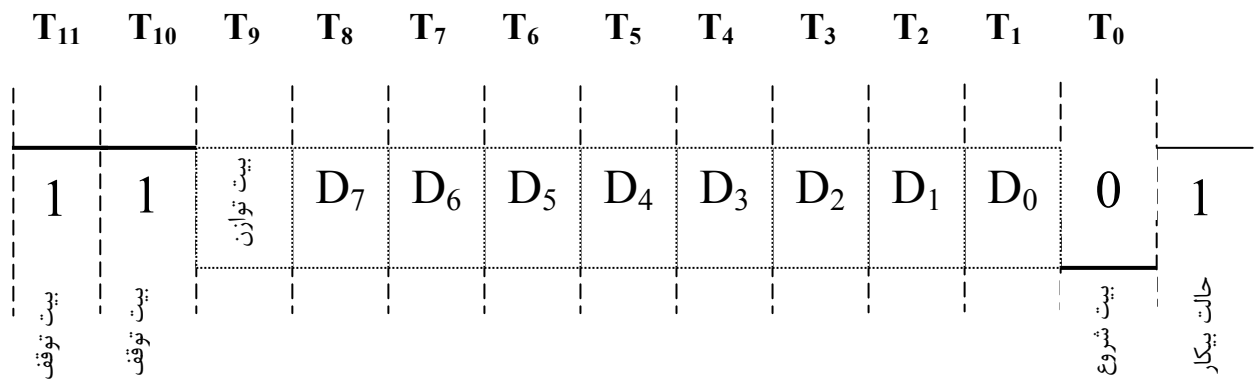
در ارتباط سریال همگام، چون زمان خواندن هر بیت اطلاعات از روی خط داده دقیقاً مشخص است، فرستنده می تواند هر تعداد بیت را پشت سر هم ارسال کند (خواهیم دید که در ارتباط غیر همگام این گونه نیست) به بیان دیگر طول بسته اطلاعاتی می تواند بزرگ باشد. ابتدا و انتهای قاب یا بسته اطلاعاتی با الگوهای خاصی که به ترتیب Preamble و Postamble نام دارند مشخص می شود. در واقع گیرنده مرتباً خط داده را می خواند تا الگوی Preamble را دریافت کند. سپس تا وقتی که الگوی postamble را دریافت نکرده خط را می خواند و

اطلاعات روی آنرا ثبت می کند.

در بعضی طرح های ارتباط سریال همگام، بعد از الگوی Preamble تعداد بایت های اطلاعاتی گنجانده شده در بسته توسط فرستنده به گیرنده ابلاغ می شود. گیرنده نیز بعد از دریافت Preamble و اطلاعات کنترلی ابتدای بسته، به همان تعدادی که ذکر شده بایتهای اطلاعاتی را از خط می خواند.

## قاب بندی در ارتباط سریال غیر همگام

همانطور که ذکر شد، در ارتباط غیر همگام سیگنال ساعت وجود ندارد و نرخ ارسال داده ها باید قبل از آغاز ارتباط بین فرستنده و گیرنده مورد توافق قرار گیرد. به همین لحاظ قاب ارتباط سریال غیر همگام با قاب سریال همگام متفاوت است. شکل زیر یک قاب داده در ارتباط سریال غیر همگام را نشان می دهد:



خط ارتباطی سریال غیر همگام در حالت عادی (بیکار) حاوی (یک) است. هنگامی که فرستنده می خواهد ارسال اطلاعات را آغاز کند، باید به نحوی گیرنده را مطلع کند. به همین لحاظ در ابتدای قاب داده، یک بیت (صفر) به نام بیت شروع ارسال می شود تا گیرنده از تغییر (یک) به (صفر) متوجه آغاز ارسال اطلاعات شود.

به دنبال بیت شروع، تعدادی بیت داده با شروع از کم ارزش ترین بیت (LSB) ارسال می گردد. این تعداد می تواند از ۵ بیت (صفحه کلید های کوچک غیر اسکی) تا ۸ بیت (سیستم های مهمول انتقال داده) باشد. بیت بعدی، بیت توازن برای کشف خطا است که می تواند از نوع توازن زوج یا فرد باشد. قاب می تواند حاوی بیت توازن نباشد.

در انتهای بسته دو بیت (یک) توسط فرستنده ارسال می شود که به بیت توقف معروفند. ارسال این بیت های توقف به دو هدف انجام می شود: اول اینکه به گیرنده فرصت کافی برای پردازش بسته دریافتی داده شود. هدف دیگر این است که خط را به حالت عادی (حالت بیکار که حاوی (یک) است) برگرداند تا آغاز بسته بعدی به کمک بیت شروع که صفر است بتواند مشخص شود. تعداد بیت های توقف می تواند ۱ یا ۱/۵ یا ۲ بیت باشد که معمولاً به تعداد بیت های داده وابسته است. در سیستم های قدیمی تر که کند بودند از دو بیت توقف استفاده می شد. در سیستم های امروزی استفاده از یک بیت توقف رایج است.

نکته تامل برانگیز در ساختار قاب ارتباط سریال غیر همگام، تعداد بیت های داده است که به حداکثر هشت بیت محدود شده است (البته این تعداد می تواند از ۵ تا ۸ بیت متغیر باشد). چون در ارتباط غیر همگام سیگنال ساعت وجود ندارد، فرستنده و گیرنده باید بر سر یک سرعت به توافق برسند و گیرنده با فرض آن سرعت خط داده را بخواند.

فرض کنید سرعت توافق شده ۹۶۰۰ بیت در ثانیه است. این بدان معناست که فرستنده در هر ثانیه ۹۶۰۰ بیت ارسال می کند. بنابراین زمان ارسال هر بیت ۱/۹۶۰۰ ثانیه است. وقتی گیرنده بیت شروع را دریافت می کند، متوجه آغاز یک بسته اطلاعاتی می شود. پس از سپری شدن زمانی که بیت شروع روی خط قرار دارد (۱/۹۶۰۰ ثانیه)، گیرنده باید بیت های بعدی را به موقع (یعنی وسط زمانی که بیت روی خط قرار دارد) از روی خط بخواند. بنابراین گیرنده پس از سپری شدن زمانی که بیت شروع روی خط قرار دارد (۱/۹۶۰۰ ثانیه) به اندازه نصف زمان یک بیت (۱/(۹۶۰۰×۲) ثانیه) صبر می کند و سپس بیت D0 را می خواند، ۱/۹۶۰۰ ثانیه بعد بیت D1 را می خواند، ۱/۹۶۰۰ ثانیه بعد بیت D2 را می خواند و ... تا سرانجام به انتهای بسته برسد.

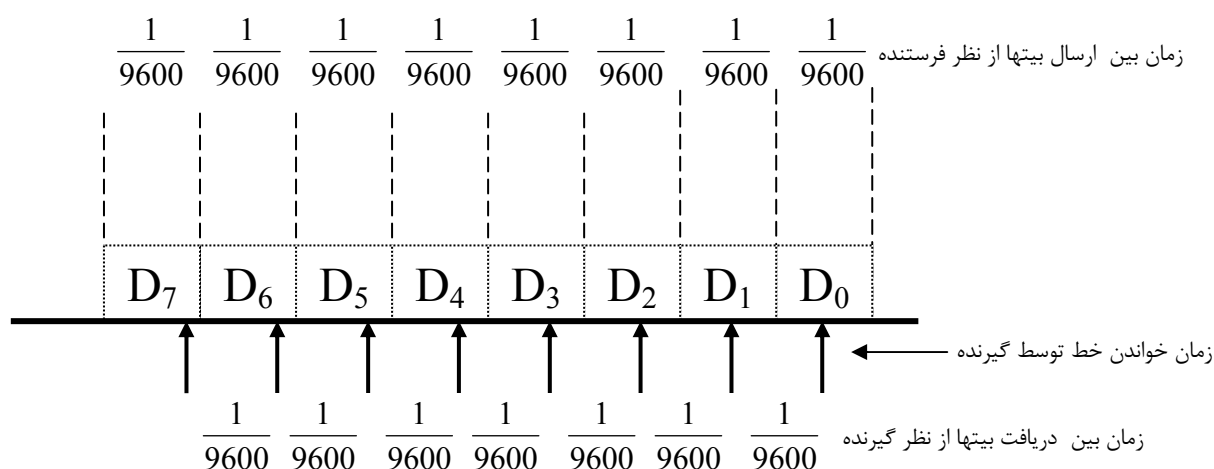
اگر همه چیز طبق برنامه بالا پیش برود، باید فرستنده در هر ثانیه ۹۶۰۰ بیت ارسال و گیرنده این ۹۶۰۰ بیت را به درستی دریافت کند. پس چرا طول قسمت داده قاب اطلاعاتی غیر همگام محدود به ۸ بیت شده است؟

مشکل اینجاست که همه چیز طبق برنامه جلو نمی رود!

فرستنده (با ساعت خودش) یک بیت می فرستد، ۱/۹۶۰۰ ثانیه صبر می کند، بیت بعدی را می فرستد، ۱/۹۶۰۰ ثانیه صبر می کند، بیت بعدی را می فرستد، ۱/۹۶۰۰ ثانیه صبر می کند و ....

گیرنده (با ساعت خودش) یک بیت را از روی خط می خواند، ۱/۹۶۰۰ ثانیه صبر می کند، بیت بعدی را از روی خط می خواند، ۱/۹۶۰۰ ثانیه صبر می کند، بیت بعدی را از روی خط می خواند، ۱/۹۶۰۰ ثانیه صبر می کند، بیت بعدی را از روی خط می خواند و ....

مساله همین جاست که ساعت فرستنده با ساعت گیرنده یکسان نیست. در واقع سیستم های مبتنی بر پردازنده برای تولید پالس ساعت از اجزاء الکترونیکی طبیعی (مانند کریستال) استفاده می شود که پیدا کردن دو جزء کاملاً یکسان در طبیعت غیر ممکن است. به همین لحاظ زمان ۱/۹۶۰۰ ثانیه در فرستنده با زمان ۱/۹۶۰۰ ثانیه در گیرنده متفاوت است و همین اختلاف زمان باعث ایجاد مشکل می شود. شکل زیر نشان می دهد اگر سرعت گیرنده سریعتر از ساعت فرستنده کار کند، چه مشکلی ایجاد می شود:



گیرنده و فرستنده توافق کرده اند زمان بین بیت ها ۱/۹۶۰۰ ثانیه باشد و هر کدام بر اساس ساعت خود این زمان را اندازه می گیرند. همانطور که می بینید بعد از خواندن هر بیت، لحظه خواندن خط توسط گیرنده کمی زودتر اتفاق می افتد. بعد از دریافت ۸ بیت اطلاعات، لحظه خواندن خط توسط گیرنده از وسط زمانی که بیت روی خط دارد فاصله زیادی پیدا کرده و اگر تعداد بیت های داده بیشتر شود امکان دارد گیرنده زمانی خط را بخواند که هنوز بیت روی خط نیامده است.

## مقایسه بین ارتباطات همگام و غیر همگام

هرچند هم اکنون بیشتر ارتباطات سریال از نوع غیر همگام هستند، اما مقایسه ای بین ارتباطات همگام و غیر همگام مفید است: در ارتباط همگام طول بسته اطلاعاتی محدود نیست (مگر اینکه محدودیت هایی از ناحیه لایه های بالاتر ارتباطی اعمال شده باشد). در ارتباط همگام، فرستنده در حین ارتباط می تواند با تغییر سیگنال ساعت، سرعت ارتباط را تغییر دهد (هرچند این کار معمول نیست) اما در ارتباط غیر همگام سرعت باید از ابتدا بین فرستنده و گیرنده مورد توافق قرار گیرد. سربار اطلاعات در ارتباط غیر همگام بالاتر است. در این نوع ارتباط برای ارسال ۸ بیت اطلاعات باید یک قاب ۱۲ بیتی ارسال شود که ۴ بیت آن سربار است. بنابراین تنها ۶۶٪ از ۸/۱۲ پهنای باند خط استفاده می شود. در ارتباط همگام تعداد بیت های داده بیشتر است و به کارایی بیشتری در استفاده از خط ارتباطی می انجامد. در ارتباط غیر همگام نیازی به سیگنال ساعت (CLK) نیست که این موضوع در کاهش هزینه ارتباط موثر است. به علاوه در ارتباطات بیسیم (مانند دستگاه کنترل از راه دور تلوزیون که از سیگنال های مادون قرمز استفاده می کند یا دزدگیر رادیویی) ارسال دو سیگنال داده و ساعت منطقی به نظر نمی رسد.

## کنترل جریان داده

پیشتر دیدیم فرستنده باید حجم اطلاعات ارسالی را کنترل کند تا به گیرنده فرصت پردازش اطلاعات بدهد. این موضوع را کنترل جریان نامیدیم. آیا کنترل جریان داده بحثی مجزا از تطابق سرعت بین فرستنده و گیرنده است؟ به بیان دیگر اگر سرعت فرستنده و گیرنده با هم تطابق داشته باشد، باز هم نگرانی در مورد کنترل جریان داده وجود دارد؟ پاسخ مثبت است. برای مثال فرض کنید شخصی اطلاعاتی برای شما می خواند تا یادداشت کنید. آن شخص آنقدر تند صحبت کند که شما اصلاً متوجه حرف های او نشوید. این مشکل عدم تطابق سرعت بین فرستنده و گیرنده است. امام در حالتی دیگر ممکن است آن شخص طوری صحبت کند که شما متوجه حرف های او بشوید، اما آنقدر اطلاعات را به سرعت بخواند که شما موفق به یادداشت کردن اطلاعات نشوید. این مشکل عدم کنترل صحیح جریان داده است. بنابراین در ارتباط سریال، هم تطابق سرعت بین فرستنده و گیرنده و هم راهکاری برای کنترل جریان داده لازم است. در ارتباطات سریال غیر همگام دو نوع کنترل جریان داده وجود دارد. یکی روش سخت افزاری است که پس از معرفی درگاه سریال کامپیوتر



به آن خواهیم پرداخت.

روش دیگر کنترل جریان داده به شیوه نرم افزاری است. در این روش، هرگاه بافر گیرنده در شرف پر شدن باشد و نتواند اطلاعات جدید را ثبت کند، یک کاراکتر ویژه به نام Xoff برای فرستنده ارسال می‌کند. فرستنده با دریافت این کاراکتر، ارسال اطلاعات را متوقف می‌کند و صبر می‌کند تا گیرنده آمادگی خویش برای دریافت اطلاعات جدید را با ارسال کاراکتری ویژه به نام Xon اعلام کند. این روش کنترل جریان داده به Xoff/Xon موسوم است.

## مشخصات ارتباط غیر همگام

به صورت خلاصه در ارتباط غیر همگام باید مشخصات زیر بین فرستنده و گیرنده مورد توافق قرار گیرد:

نرخ داده (مثلاً ۹۶۰۰ یا ۱۹۲۰۰ بیت در ثانیه)

تعداد بیت های داده (۵ یا ۶ یا ۷ یا ۸ بیت در ثانیه)

نوع توازن به کار رفته (زوج، فرد، بدون توازن، همیشه یک، همیشه صفر).

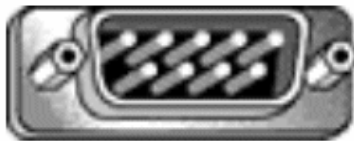
تعداد بیت های توقف (۱ یا ۱/۵ یا ۲ بیت)

روش کنترل جریان داده (سخت افزاری، Xoff/Xon یا بدون کنترل جریان داده)

## پورت سریال کامپیوتر

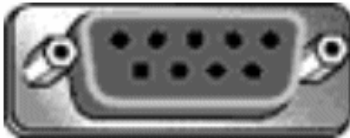


1 2 3 4 5



6 7 8 9

5 4 3 2 1



9 8 7 6

اگر به پشت برد اصلی کامپیوتر خود نگاه کنید، یک پورت با ۹ پایه مانند شکل مقابل می بینید که به پورت سریال، پورت COM یا در اصطلاح فنی الکترونیک پورت DB9 معروف است. به کمک این پورت، می توان یک ارتباط سریال غیر همگام با مشخصات قابل تغییر داشت.

در ادامه به بررسی سیگنال های پورت سریال کامپیوتر می پردازیم. با شناخت این سیگنال ها دو مطلب را خواهیم آموخت. اول چگونگی عملکرد مودم های خارجی در تعامل با کامپیوتر و دوم نحوه کنترل یک دستگاه جانبی به کمک پورت سریال کامپیوتر.

پیش از توضیح عملکرد پایه ها، توضیحی راجع به نحوه شماره گذاری پایه ها لازم است. پورتهای پشت کامپیوتر مشاهده می کنید، دارای ۹ میله است که سیگنال ها را در اختیار کاربر قرار می دهد. شماره گذاری این حفره ها به صورت مقابل است.

معمولاً برای استفاده از این پورت، قطعه ای به کار می رود که به جای میله دارای حفره است و طوری روی پورت سریال کامپیوتر نصب می شود که میله ها، درون حفره ها قرار می گیرند. این قطعه نیز به نام DB9 معروف است. شکل مقابل را ببینید.

سر کابل مودم خارجی (یا دستگاه های دیگر مانند اسکنر که به پورت سریال متصل می شوند) نیز چنین شکلی دارد.

دقت کنید که در پورت اصلی کامپیوتر، پایه بالا سمت چپ و در قطعه دومی که روی پورت نصب می شود، پایه بالا سمت راست شماره یک است.

چون وقتی قطعه ی بالا روی پورت سریال نصب می شود، سمت چپ آن با سمت راست پورت مقابل هم قرار می گیرند. پشت این قطعه در محل هر میله، جایی برای لحیم کردن سیم تعبیه شده تا براحتی بتوانید از سیگنال های پورت سریال استفاده کنید.



## استاندارد ارتباط سریال کامپیوتر

در سال ۱۹۶۰ سازمان صنایع الکترونیک (EIA) برای استاندارد کردن ارتباط سریال بین وسایل مختلف (از جمله کامپیوتر ها) پروتکلی پیشنهاد کرد که به EIA232 یا RS232 معروف شد. بخش سخت افزاری این پروتکل شامل سیگنال های داده و کنترلی است که در ادامه با آن آشنا خواهیم شد.

## ولتاژ های منطقی در RS232

قبلا آموختیم که اگر ولتاژی بین صفر تا  $0/8$  ولت باشد از آن به (صفر منطقی) و اگر بین ۳ تا ۵ ولت باشد از آن به عنوان (یک منطقی) برداشت می شود. چون RS232 قبل از TTL استاندارد شد، سطوح ولتاژ منطقی آن با TTL متفاوت است. در RS232 (که پورت سریال کامپیوتر نمونه ای از کاربرد متفاوت آن است)، ولتاژی بین ۳- ولت تا ۲۵- ولت به عنوان (یک) و ولتاژی بین ۳+ تا ۲۵+ ولت به عنوان (صفر) منطقی در نظر گرفته می شود.

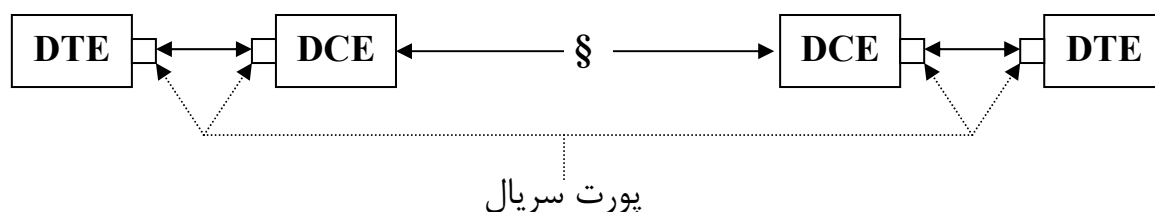
توجه کنید چون معمولاً تراشه هایی با استاندارد TTL به پورت سریال کامپیوتر متصل می شود (مانند تراشه های مدیریت ارتباط سریال یا میکروکنترلر ها)، به تراشه ای نیاز داریم که ولتاژ های RS232 را به TTL و بالعکس تبدیل کند. این تراشه ها Line Driver نامیده می شوند که MAX232 و MAX233 نمونه ای از آن هستند.

یکی از مزایای پورت سریال این است که برای دستگاه های کم مصرف که به آن متصل شوند، می تواند تغذیه مورد نیاز را تامین کند. جریان خروجی پورت سریال کامپیوتر بین ۱۰ تا ۲۰ میلی آمپر و مقاومت ورودی ۱۰ کیلو اهم است.

## سیگنال های RS232

چون پورت سریال از ابتدا برای اتصال مودم خارجی به کامپیوتر استاندارد شد، سیگنال های این پورت نیز با نامهایی که در ارتباط کامپیوتر و مودم به کار می روند نامگذاری شده اند. برای فهم این سیگنال ها ابتدا به نحوه برقراری ارتباط بین دو کامپیوتر به کمک مودم خارجی می پردازیم.

شکل زیر این ارتباط را نشان می دهد:



منظور از DTE هر دستگاهی است که بخواهد اطلاعاتی ارسال یا دریافت کند. کامپیوتر یا میکروکنترلر مثالی از DTE هستند. DCE دستگاهی است که وظیفه آن دریافت داده ها از DTE و ارسال آن روی خط داده یا دریافت داده ها از روی خط انتقال و ارائه آن به DTE است. مودم مثال خوبی از DCE است.

برای ارسال داده ها از DTE اولی به DTE دوم روند زیر دنبال می شود.

- ارسال داده ها از DTE اول به DCE اول
- ارسال داده ها از DCE اول به DCE دوم
- ارسال داده ها از DCE دوم به DTE دوم

اکنون ببینیم پورت سریال چگونه به عنوان رابط بین DTE و DCE عمل می کند. جدول زیر سیگنال های پورت سریال کامپیوتر را نشان می دهد:

شماره پایه	عملکرد	جهت سیگنال (از دید کامپیوتر)	نوع سیگنال
۱	Data Carrier Detect (DCD)	ورودی	کنترل
۲	Received Data (RxD)	ورودی	داده
۳	Transmitted Data (TxD)	خروجی	داده
۴	Data Terminal Ready (DTR)	خروجی	کنترل
۵	Ground (Gnd)	-	-
۶	Data Set Ready (DSR)	ورودی	کنترل
۷	Request To Send (RTS)	خروجی	کنترل
۸	Clear To Send (CTS)	ورودی	کنترل
۹	Ring Indicator (RI)	ورودی	کنترل

سیگنال‌های فوق را می‌توان به سه دسته کلی تقسیم کرد:

**الف) سیگنال‌های داده**

**RxD پایه**

کامپیوتر از طریق این پایه، اطلاعات را به صورت سریال از مودم دریافت می‌کند.

**TxD پایه**

کامپیوتر از طریق این پایه، اطلاعات را به صورت سریال برای مودم ارسال می‌کند.

**Gnd پایه**

زمین پورت سریال است که باید با زمین DCE مشترک شود.

**ب) سیگنال‌های اعلام آمادگی ارتباط**

**DCD پایه**

وقتی مودم با مودم طرف مقابل ارتباط برقرار می‌کند و سیگنال حامل معتبر روی خط کشف می‌کند، از طریق این پایه مراتب را به کامپیوتر سمت خودش اطلاع می‌دهد.

**DTR پایه**

وقتی کامپیوتر آماده ارتباط با مودم است این پایه را فعال می‌کند.

**DSR پایه**

وقتی مودم آماده ارتباط با کامپیوتر است، این پایه را فعال می‌کند.

**RI پایه**

وقتی مودم سیگنال زنگ (Ring) را روی خط ارتباطی با مودم دیگر دریافت می‌کند، از طریق این پایه به کامپیوتر اطلاع می‌دهد تا به خط جواب دهد.

**ج) سیگنال‌های دست‌دهی برای کنترل جریان داده**

**RTS پایه**

هنگامی که کامپیوتر می‌خواهد برای مودم یک بایت بفرستد، با فعال کردن این پایه از مودم درخواست اجازه می‌کند.

**CTS پایه**

وقتی مودم سیگنال RTS را از کامپیوتر دریافت کرد، در صورت آمادگی برای دریافت داده، سیگنال CTS را برای کامپیوتر ارسال می‌کند.

**مودم پوچ (Null Modem)**

در توضیحات بالا دیدیم چگونه می‌توان دو DTE را به کمک دو DCE به هم متصل کرد. حال فرض کنید می‌خواهیم دو DTE (مثلاً دو کامپیوتر، دو میکروکنترلر یا یک کامپیوتر و یک میکروکنترلر) را به یکدیگر متصل کنیم، اما DCE (مثلاً مودم) در اختیار نداریم. در این حالت باید با توجه به عملکرد سیگنال‌های RS232، طوری بین پورت‌های سریال دو DTE سیم‌بندی کنیم که تصور کنند از طریق DCE به هم متصل هستند، در حالی که عملاً DCE بین آنها وجود ندارد. به این اتصال، مودم پوچ گفته می‌شود.

سیم‌بندی بین پورت‌های سریال برای شبیه‌سازی وجود DCE چگونه باید باشد؟

به عنوان مثال DTE اول می‌خواهد برای DTE دوم اطلاعات ارسال کند. اگر DCE وجود داشته باشد:

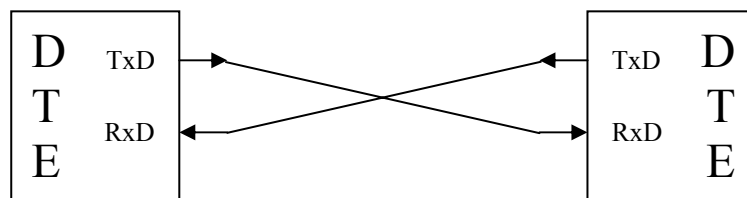
• DTE اول از طریق خط TxD خودش اطلاعات را برای DCE خودش ارسال می‌کند.

• DCE اطلاعات را برای DCE مقابل ارسال می‌کند.

• DCE مقابل اطلاعات را برای DTE دوم می‌فرستد که از طریق سیگنال RxD پورت سریال DTE دریافت می‌شود.

حال که DCE وجود ندارد، بدیهی است که باید خط RxD هر DTE به خط TxD در DTE مقابل و بالعکس متصل شود.

شکل زیر ساده‌ترین نوع مودم پوچ را نشان می‌دهد:



بدیهی است زمین هر دو DTE باید مشترک باشد. به همین لحاظ سیگنال Gnd دو پورت سریال به هم متصل می‌شود.

تکلیف سیگنال‌های دست‌دهی چیست؟

گفتیم هنگامی که DTE می‌خواهد برای DCE یک بایت بفرستد، با فعال کردن این پایه از وی درخواست اجازه می‌کند. وقتی DCE سیگنال RTS را از DTE دریافت کرد، در صورت آمادگی برای دریافت داده، سیگنال CTS را برای DTE ارسال می‌کند. حال که DCE وجود ندارد که DTE بخواهد از آن اجازه بگیرد، یعنی این اجازه باید همیشه داده شود. به همین لحاظ در ساختار مودم پوچ، RTS هر DTE به CTS آن متصل می‌شود. یعنی هرگاه DTE برای اجازه گرفتن از DCE سیگنال RTS را فعال کند، بلافاصله سیگنال CTS خودش فعال می‌شود و DTE

Pin No.

5 GND  
3 TxD  
2 RxD  
7 RTS  
8 CTS  
1 CD  
6 DCR  
4 DTR  
9 RI  
COM port  
(Computer1)



GND  
TxD  
RxD  
RTS  
CTS  
CD  
DCR  
DTR  
RI  
COM port  
(Computer2)

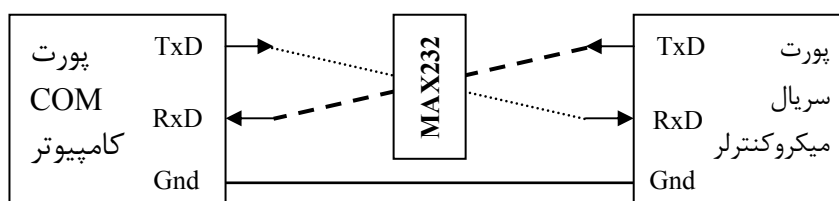
تصور می کند که اجازه از سوی DCE صادر شده است.

شکل مقابل ساختار کامل مودم پوچ برای ارتباط بین پورت سریال دو کامپیوتر را نشان می دهد.

کابل CAT-5 که در سیم کشی شبکه LAN معمول است، برای برقراری این ارتباط مناسب می باشد. چون ۴ زوج سیم و یک سیم بدنه (شیلد) دارد که می توان هر ۹ اتصال را به کمک آن برقرار کرد. به کمک دو قطعه DB9 که روی پورت سریال دو کامپیوتر نصب شوند و مقداری سیم CAT-5 می توان ارتباط مودم پوچ را بین دو کامپیوتر برقرار کرد. در این حالت به کمک ارتباط نرم افزاری مناسب (که بعد ها خواهیم دید) می توان ارتباط داده ای و حتی شبکه بین دو کامپیوتر برقرار کرد.

## اتصال میکروکنترلر به کامپیوتر

اکثر میکروکنترلرها دارای کنترل کننده ارتباط سریال غیر همگام (UART) هستند. برای ارتباط بین میکروکنترلر و کامپیوتر می توان از ساختار مودم پوچ به صورت زیر استفاده کرد:



شرح برنامه میکروکنترلر برای ارتباط سریال در حوصله بحث ما نیست. تنها به یاد داشته باشید که مشخصات ارتباط غیر همگام (نرخ داده، تعداد بیت های داده، نوع توازن به کار رفته، تعداد بیت های توقف، روش کنترل جریان داده) در کامپیوتر و میکروکنترلر باید یکسان باشد.

## نرم افزار کامپیوتر برای ارتباط با پورت سریال

اکنون اتصال سخت افزاری بین کامپیوتر و DTE دیگری (کامپیوتر دیگر یا میکروکنترلر یا...) برقرار شده است. می خواهیم در کامپیوتر برنامه ای بنویسیم که ارتباط سریال را با DTE مقابل برقرار کند. یعنی اطلاعات را به صورت سریال ارسال و دریافت کند.

برای برقراری ارتباط نرم افزاری در ابتدا باید ارتباط سریال پیکربندی شود. یعنی نرخ داده، تعداد بیت های داده، نوع توازن به کار رفته، تعداد بیت های توقف و روش کنترل جریان داده مشخص شود. این مشخصات عینا در نرم افزار DTE طرف مقابل برای پیکربندی پورت سریال آن نیز باید به کار رود.

نرم افزار ارتباط سریال در کامپیوتر می تواند به کمک زبانهای سطح بالا مانند C یا Delphi یا Visual C++ یا... نوشته شود که در ادامه مثال هایی در DOS و ویندوز خواهیم داشت.

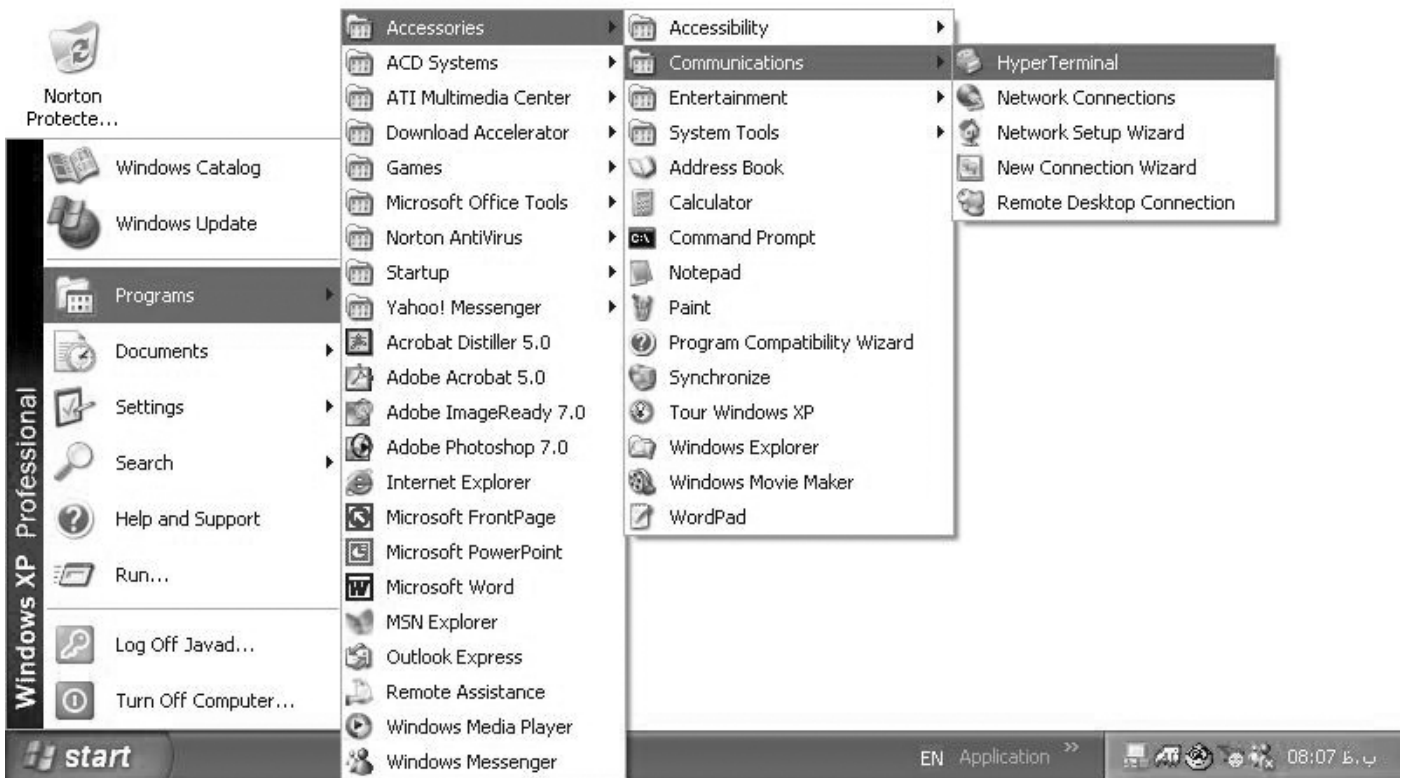
اما راه ساده تری نیز وجود دارد. ویندوز دارای نرم افزاری به نام Hyper Terminal است که می تواند ارتباط های مختلف از جمله ارتباط سریال را به راحتی برقرار کند. ابتدا با این نرم افزار آشنا می شویم.

## نرم افزار Hyper Terminal

این نرم افزار در نسخه های مختلف ویندوز وجود دارد. برای استفاده از آن (در ویندوز XP) به ترتیب زیر عمل کنید

- از منوی Start ویندوز به مسیر Programs/Accessories/Communications بروید و گزینه Hyper Terminal را انتخاب کنید





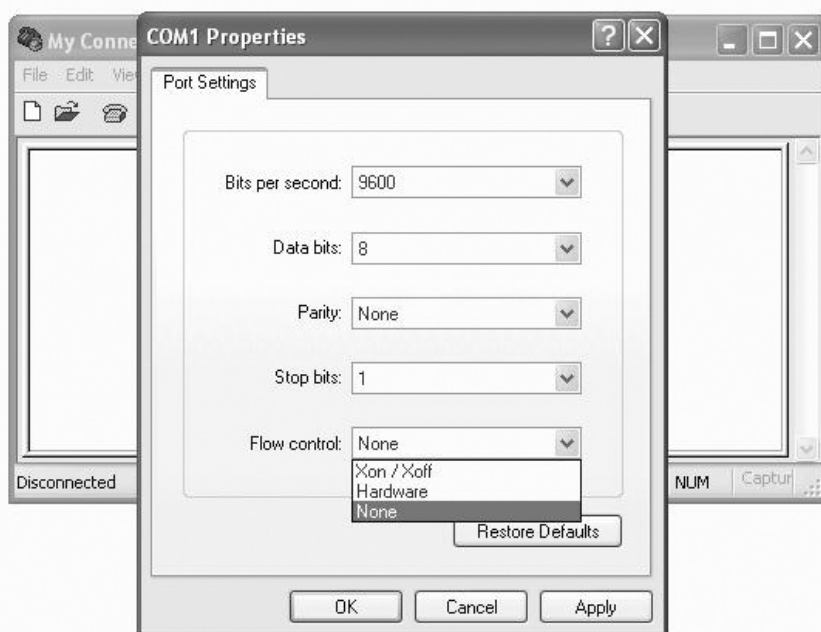
• مرحله بعد شکلی مانند زیر دارد که در آن باید نام ارتباط را انتخاب کنید. نام My Connection در شکل زیر برای ارتباط انتخاب شده است.



• در مرحله بعد باید وسیله ارتباطی را انتخاب کنید. این وسیله ارتباطی می تواند کارت شبکه، کارت مودم یا پورت های سریالی باشد. پورت COM1 را انتخاب کنید.



توجه کنید که کامپیوتر دارای یک یا دو پورت سریال سخت افزاری (COM1 و احتمالاً COM2) است که برای اتصال تجهیزات خارجی به کار می‌روند. به علاوه تعدادی پورت سریال داخلی (COM3، COM4 و...) نیز وجود دارد که برای اتصال کارت مودم‌های داخلی استفاده می‌شوند. در بعضی متون به این پورتهای سریال، یوژت‌های مجازی گفته می‌شود.



در ارتباط بالا نرخ داده برابر ۹۶۰۰ بیت در ثانیه، تعداد بیت‌های داده برابر ۸ بیت، بدون بیت توازن، با یک بیت توقف انتخاب شده است. توجه کنید که پورت سریال کامپیوتر از نرخ‌های ارتباط داده محدودی مانند ۱۱۰، ۳۰۰، ۶۰۰، ۱۲۰۰، ۲۴۰۰، ۴۸۰۰، ۹۶۰۰، ۱۹۲۰۰ بیت در ثانیه حمایت می‌کند و دستگاه متصل به این پورت باید نرخ داده موردنظرش را این لیست انتخاب کند. برای کنترل جریان داده می‌توان از روش‌های سخت افزاری یا نرم افزاری (Xon/Xoff) استفاده کرد. پس از این مراحل، وارد محیط Hyper Terminal می‌شوید. از این پس هرچه در این محیط تایپ کنید، با مشخصات تنظیم شده از پورت سریال کامپیوتر به بیرون فرستاده می‌شود. توجه کنید آنچه تایپ می‌کنید در محیط Hyper Terminal نشان داده نمی‌شود. از سوی دیگر، هر اطلاعاتی به صورت سریال با مشخصات تنظیم شده به پورت سریال کامپیوتر برسد، در محیط Hyper Terminal نمایش داده می‌شود. نحوه نمایش این اطلاعات در Hyper Terminal قابل تنظیم است. در حالت پیش فرض، هر عددی دریافت شود، کاراکتری که کد اسکی آن برابر آن عدد است در محیط Hyper Terminal نشان داده می‌شود.

## استفاده از پورت سریال در زبانهای سطح بالا

نرم افزار Hyper Terminal تنها برای برقرار کردن ارتباط داده ای ساده به کمک ابزارهای ارتباطی کامپیوتر به کار می‌رود. چنانچه بخواهید یک ارتباط هوشمند برقرار کنید (مثلاً نمودار تغییرات دمای اتاق در طول یک شبانه روز را روی کامپیوتر مشاهده کنید یا یک ربات را به کمک کامپیوتر کنترل کنید) دیگر استفاده از Hyper Terminal به کار نمی‌آید و باید به کمک برنامه ای که در کامپیوتر می‌نویسید این کار را انجام دهید.

اکثر زبان‌های سطح بالا از ارتباط با پورت سریال حمایت می‌کنند. برای نوشتن برنامه ارتباط سریال در هر زبانی باید موارد زیر مشخص شود:

- چگونه پورت سریال باید پیکربندی شود؟ به بیان دیگر ویژگی‌های ارتباط سریال مانند نرخ داده، تعداد بیت‌های داده، نوع توازن به کار رفته، تعداد بیت‌های توقف و روش کنترل جریان داده باید به چه صورتی مشخص گردد؟
- ارسال داده به پورت سریال به چه صورت انجام می‌شود؟
- چگونه می‌توان از دریافت کاراکتر توسط پورت سریال مطمئن شد؟
- دریافت داده از پورت سریال به چه صورت انجام می‌شود؟

در ادامه به سوالات بالا در زبان C پاسخ می‌دهیم. توجه داشته باشید که همانطور که قبلاً گفته شد، داستان ارتباط با پورت ای ورودی / خروجی کامپیوتر (از جمله پورت سریال) در DOS و ویندوز متفاوت است. آنچه در ادامه می‌آید، برقراری ارتباط د C تحت DOS را نشان می‌دهد. برای برقراری ارتباط با پورت‌ها در ویندوز، به نحوه ارسال و دریافت داده‌ها در زبان Visual C++ خواهیم پرداخت.

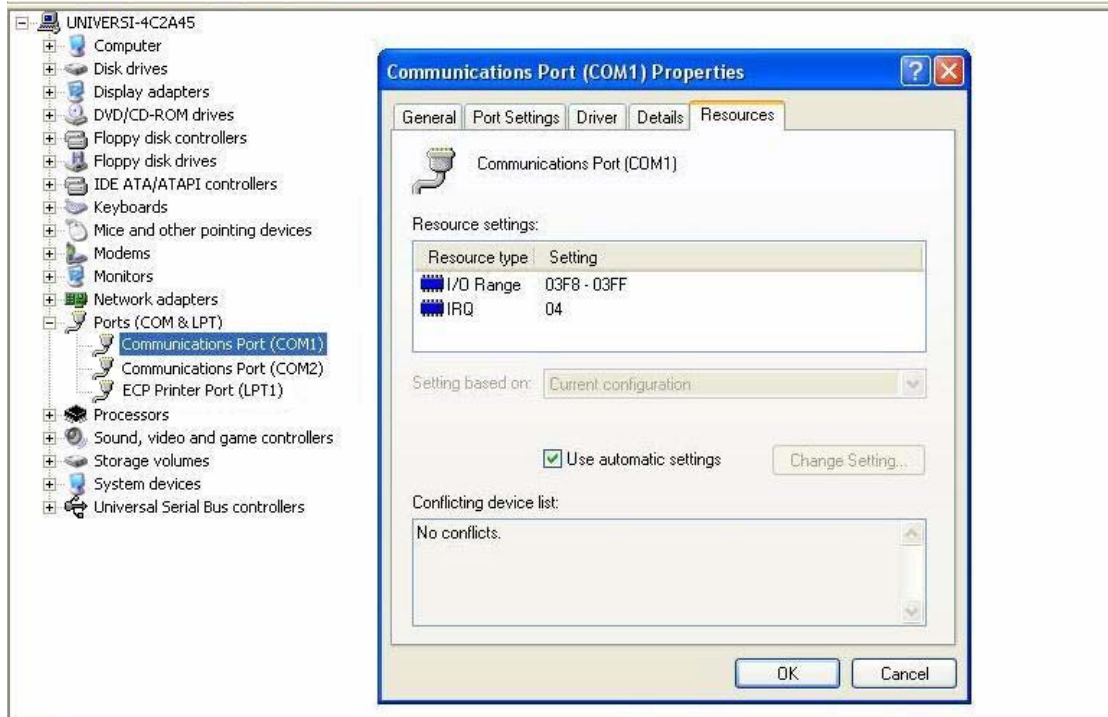
## ثبات‌های پورت سریال

مانند پورت موازی که دارای یک مجموعه ثبات است، هر پورت سریال کامپیوتر نیز یک مجموعه ای متشکل از هشت ثبات برای مدیریت ارتباط سریال دارد. آدرس پایه این ثبات‌ها برای پورت‌های COM مختلف را در جدول روبرو می‌بینید:

پورت COM	آدرس پورت
COM1	03F8-03FF
COM2	02F8-02FF
COM3	03E8-03EF
COM4	02E8-02EF

در ویندوز می‌توانید پورت های سریال سخت افزاری کامپیوتر خود و نیز آدرس های تخصیص داده شده به ثبات های ویژه را ببینید. برای این کار:

- از Control Panel گزینه System را انتخاب کنید.
- به سر برگ Hardware بروید و گزینه Device Manager را انتخاب کنید.
- حال در قسمت Ports می‌توانید مشخصات پورت های کامپیوتر خود را ببینید. شکل زیر، گزینه Ports مربوط به یک کامپیوتر نوعی را نشان می‌دهد.



پورت COM	آدرس پایه ورودی/خروجی ثباتهای پورت
COM1	0040:0000 – 0040:0001
COM2	0040:0002 – 0040:0003
COM3	0040:0004 – 0040:0005
COM4	0040:0006 – 0040:0007

همانطور که دیده می‌شود، هشت ثبات با شروع از 03F8 h به پورت COM1 اختصاص داده شده است.

به کمک BIOS نیز می‌توان آدرس مبنای ثبات های ورودی/خروجی پورت سریال را دانست. در واقع این آدرس ها از آفست 0000 ذخیره می‌شود. جدول مقابل را ببینید.

مثلا دستور D 0040:0000 L8 در نرم افزار Debug، آدرس پایه پورت های سریال نصب شده در کامپیوتر را نشان می‌دهد. قبل با نرم افزار Debug و فرمان های آن آشنا شدیم.

به کمک برنامه نویسی C نیز می‌توان آدرس های ذخیره شده در این بخش حافظه را مشاهده کرد. برنامه ریز، آدرس پورت COM1 و در صورت عدم نصب پورت مزبور، پیغامی را نمایش می‌دهد.

```
#include <STDIO. H>
#include <DOS. H>
void main (){
    unsigned int far *fptr;
    fptr = (unsigned int far *) 0x00000400;
    if (*fptr > 0)
        printf ("COM1 I/O base address = %X",*fptr);
    else
        printf ("COM1 not found");
}
```

جدول زیر ثبات های پورت سریال و آدرس آنها نسبت به آدرس مبنا (Base) را نشان می دهد:

آدرس	عملکرد	خواندنی یا نوشتنی	نام	بیت DLAB
Base+0	بافر ارسال اطلاعات	نوشتنی	-	0
	بافر دریافت اطلاعات	خواندنی	-	0
	قسمت پایین نرخ داده	خواندنی/نوشتنی	-	1
Base+1	فعال کننده وقفه	خواندنی/نوشتنی	IER	0
	قسمت پایین نرخ داده	خواندنی/نوشتنی	-	1
Base+2	شناسایی وقفه	خواندنی	IIR	-
	کنترل کننده FIFO	نوشتنی	FCR	-
Base+3	کنترل کننده قالب اطلاعات	خواندنی/نوشتنی	LCR	-
Base+4	کنترل کننده مودم	خواندنی/نوشتنی	MCR	-
Base+5	نمایشگر وضعیت خط	خواندنی	LSR	-
Base+6	نمایشگر وضعیت مودم	خواندنی	MSR	-
Base+7	ثبات حافظه پنهان	خواندنی/نوشتنی	-	-

مثلا اگر آدرس مبنای پورت سریال COM1 برابر 03F8 h باشد، در برنامه کامپیوتر برای ارسال داده به این پورت و نیز دریافت داده از آن باید از ثبات واقع در آدرس 03F8 h استفاده کنیم (به خط اول جدول توجه کنید) با نوشتن داده ۸ بیتی در این ثبات، تراشه UART آن را به صورت مناسب قالب بندی کرده و از پورت سریال به بیرون ارسال می کند. وقتی نیز بسته ای توسط پورت سریال دریافت می شود، تراشه UART داده را از دل آن بسته بیرون کشیده و در ثبات فوق قرار می دهد. توجه داشته باشید که داده های ارسال و دریافتی در بافر های جداگانه ای از نوع FIFO ذخیره می شوند.

## پیکربندی ارتباط سریال

گفتیم پیش از آغاز ارتباط، ویژگی های ارتباط سریال نرخ داده، تعداد بیت های داده، نوع توازن به کار رفته و تعداد بیت های توقف باید مشخص شود. برای مشخص کردن نرخ داده باید عدد ویژه ای (وابسته به نرخ داده) را در ثبات های اول و دوم قرار بدهید. در این حالت باید بیت DLAB (بیت شماره ۷ ثبات کنترل کننده قالب اطلاعات) یکی باشد. برای تعیین ویژگی های دیگر ارتباط سریال باید بیت های ثبات کنترل کننده قالب اطلاعات مقدار دهی شوند:

DLAB	Break	Parity	Parity	Parity	Stop	Data	Data
<div> <div> 000: بدون توازن 001: توازن فرد 011: توازن زوج 101: توازن همیشه «یک» 111: توازن همیشه «صفر» </div> <div> 0: یک بیت توقف 1: دو بیت توقف </div> <div> 00: ۵ بیت داده 01: ۶ بیت داده 10: ۷ بیت داده 11: ۸ بیت داده </div> </div>							

با (یک) شدن بیت قطع (Break)، فرستنده ارسال اطلاعات را قطع می کند.

## پیکربندی پورت سریال در C (اسمبلی)

در سیستم عامل DOS، وقفه 14H برای ارتباط با پورت سریال تعریف شده است که شامل توابعی برای پیکربندی پورت سریال و ارسال و دریافت داده با این پورت است. برای پیکربندی پورت سریال در زبانهای تحت DOS مانند اسمبلی یا C تحت DOS، از تابع شماره ۴ این وقفه استفاده می کنیم. با مقدار دهی ثبات های مختلف پردازنده و اجرای این تابع، می توان بدون نیاز به درگیر شدن با ثبات های پورت سریال به پیکربندی دلخواه برای ارتباط سریال دست یافت.



جزئیات این تابع در ذیل آمده است:

AH=04

AL=00H (توقف)، H بدون توقف (01)  
 DX = شماره پورت (COM1=0, COM2=1,....)  
 BH=00H (بدون توازن)  
 H توازن فرد (01)  
 H توازن زوج (02)  
 H توازن فرد پایدار (03)  
 H توازن زوج پایدار (04)  
 BL=00H (یک بیت توقف)  
 H بیت توقف برای کلمه 5 بیته (01/5)  
 H بیت توقف برای کلمه ها بزرگتر از 5 بیت (02/2)  
 CH=00H (کلمه 5 بیته)  
 H کلمه 6 بیته (01)  
 H کلمه 7 بیته (02)  
 H کلمه 8 بیته (03)  
 CH=00H (110 bps)  
 01 (150 bps)  
 02 (300bps)  
 03 (600 bps)  
 04 (1200 bps)  
 05 (2400 bps)  
 06 (4800 bps)  
 07 (9600 bps)  
 08 (19200 bps)

تابع مقابل در زبان C پورت سریال را با نرخ داده ۹۶۰۰ بیت در ثانیه، کلمه ۸ بیته، یک بیت توقف و بدون استفاده از بیت توازن پیکربندی می کند.

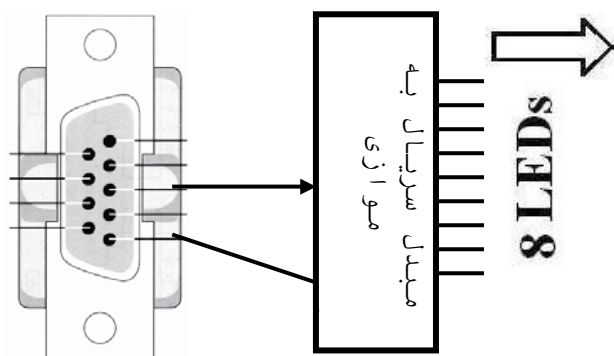
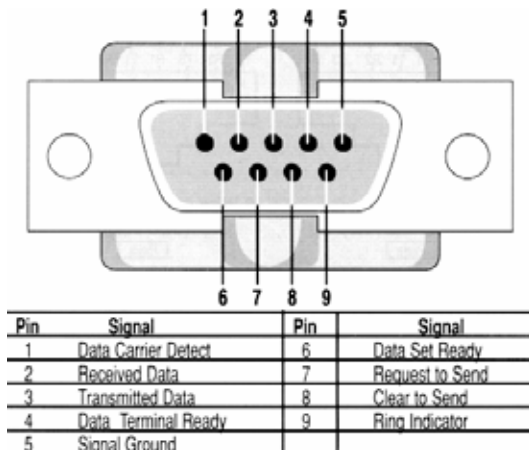
```
void InitialSerialPort (void){
asm {
mov ah,04h
mov al,00h
mov bh,00h // No Parity
mov bl,00h // 1 Stop Bit
mov ch,03h // 8 bit data
mov cl,07h // 9600 bps
mov dx,00h // port COM1
int 14h
}
```

## ارسال داده به پورت سریال

برای ارسال داده به پورت سریال، مشابه با آنچه قبلا دیدیم از تابع `outportb` استفاده می شود. البته تابع شماره ۱ وقفه 14h نیز برای این منظور می تواند به کار رود. در ادامه یک مثال عملی را بررسی می کنیم.

## مثال ارسال داده ها به کمک پورت سریال

پایه های پورت سریال را قبلا دیدیم. شکل روبه رو خلاصه ای از آنچه گفتیم و باید در مثال های طراحی سیستم به کار ببریم را نشان می دهد.



فرض کنید می خواهیم سیستمی طراحی کنیم که اعداد صفر تا ۲۵۵ را روی هشت عدد LED نمایش دهد. برای اینکار، از پورت سریال کامپیوتر به صورت مقابل استفاده می کنیم.

همانطور که می بینید، پایه TxD پورت سریال کامپیوتر به پایه ی RxD مبدل سریال به موازی (مثلا یک میکروکنترلر) متصل شده است. این مبدل، اطلاعات را به صورت سریال از کامپیوتر دریافت و پس از تبدیل به اطلاعات موازی روی هشت LED نمایش می دهد. برنامه را در شکل مقابل می بینید.

این برنامه، اعداد صفر (00000000) تا ۲۵۵ (11111111) را با تاخیر ۵۰۰ میلی ثانیه به پورت سریال کامپیوتر ارسال می کند. پس از رسیدن متغیر خروجی به ۲۵۵، مجددا مقدار آن صفر می شود. برنامه با فشار یک کلید به پایان می رسد.

```
#include <STDIO. H>
#include <CONIO. H>
#include <STDLIB. H>
void InitialSerialPort (void){
asm {
mov ah,04h
mov al,00h
mov bh,00h
mov bl,00h
mov ch,03h
mov cl,07h // 9600 bps
mov dx,00h
int 14h
}
}
void main (void){
unsigned char data = 0;
InitialSerialPort ();
while (!kbhit ()){
outportb (0x3F8,data);
delay (500);
if (data == 255)
data = 0;
else
data++;
}
}
```

## دریافت داده از پورت سریال

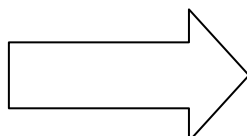
فرض کنید دستگاهی به پورت سریال کامپیوتر متصل است که اطلاعات را به صورت سریال برای کامپیوتر ارسال می کند. می خواهیم در کامپیوتر برنامه ای بنویسیم که این اطلاعات را دریافت کند. به نظر می رسد دستور inportb کافی باشد:

Data = inportb (0x3f8);

این دستور غلط نیست. یعنی اطلاعات را از پورت COM1 (به آدرس 03F8h) می خواند و در متغیر data ذخیره می کند. اما به کار بردن این دستور به تنهایی صحیح نیست. بلکه: باید ابتدا از دریافت داده توسط پورت سریال مطمئن شوید و سپس ادامه داده فوق را بخوانید. تابع 3 وقفه 14h برای خواندن وضعیت پورت به کار می رود:

AH = 03

DX = شماره پورت



AH = بایت وضعیت پورت

AL = بایت وضعیت مودم

بعد از اجرای این تابع، بایت وضعیت پورت سریال در ثبات AH قرار می گیرد که بیت های آن به شرح زیر هستند:

0	1	2	3	4	5	6	7	شماره بیت
1								وقت تمام
	1							ثبات جابجایی ارسال
		1						ثبات نگهداری ارسال خالی
			1					توقف شناسایی شد
				1				خطای قاب شناسایی شد
					1			خطای توازن شناسایی شد
						1		خطای عقب ماندن
							1	داده رسیده آماده

```
int RecvData (){
unsigned char status;
asm {
mov ah,03h
mov dx,00h
int 14h
mov status,ah
}
status = status & 128;
if (status == 0)
return 0;
else
return 1;
}
```

به بیت شماره ۷ دقت کنید. وقتی تابع ۳ وقفه 14h اجرا شود، با بررسی بیت شماره ۷ ثبات AH می‌توان دانست آیا داده‌ای دریافت شده است یا نه؟ اگر پورت سریال داده‌ای دریافت کرده باشد، این بیت (یک) و در غیر اینصورت (صفر) خواهد بود. تابع مقابل بیت دریافت داده را بررسی می‌کند و در صورت آماده بودن داده ارزش منطقی صحیح (یک) و در غیر اینصورت ارزش منطقی غلط (صفر) را باز می‌گرداند. این تابع، بایت وضعیت صفحه کلید که در ثبات AH قرار دارد را در متغیر status کپی می‌کند. سپس این متغیر را با عدد 128 (10000000) AND می‌کند. با این کار، چنانچه بیت شماره ۷ متغیر فوق (صفر) باشد، حاصل AND صفر بوده و مقدار (صفر) (ارزش منطقی نادرست) به عنوان خروجی تابع برگردانده می‌شود. در غیر اینصورت حاصل AND غیر صفر است و مقدار (یک) (ارزش منطقی درست) به

عنوان خروجی به تابع برگردانده می‌شود. بنابراین اگر خروجی تابع RecvData درست باشد یعنی داده‌ای دریافت شده و آماده خوانده شدن است.

اکنون به کمک تابع inportb می‌توانید داده را بخوانید. تابع شماره ۲ وقفه 14h نیز برای این منظور قابل استفاده است.

## مثال دریافت داده‌ها به کمک پورت سریال

می‌خواهیم سیستمی طراحی کنیم که دمای اتاق را هر ثانیه یک بار خوانده و روی نمایشگر کامپیوتر نمایش دهد. برای این کار، باید از سنسور دما استفاده شود. خروجی سنسور دما که یک کمیت آنالوگ است به کمک تراشه ADC به کمیتی دیجیتال تبدیل می‌شود.

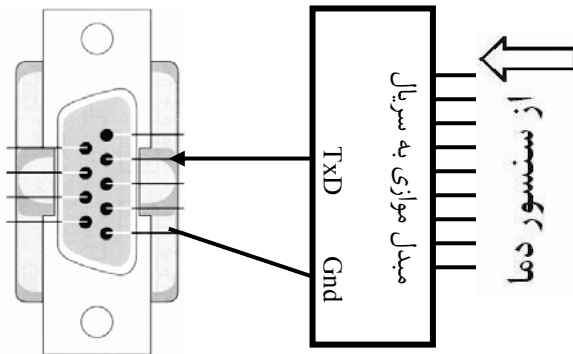
مثلاً با استفاده از یک ADC هشت بیتی (مانند تراشه ADC804)، خروجی آنالوگ سنسور دما به یک عدد دیجیتال ۸ بیتی تبدیل می‌شود. اکنون برای ارسال این عدد هشت بیتی (که در خروجی تراشه ADC به صورت موازی قرار دارد) به پورت سریال کامپیوتر، به یک مبدل سریال به موازی (مثلاً میکروکنترلر) نیاز داریم. همانطور که در شکل بالا می‌بینید، پایه TxD (ارسال) مبدل موازی به سریال به پایه Rx (دریافت) پورت سریال کامپیوتر متصل و زمین‌های دو طرف مشترک شده است.

اکنون باید در کامپیوتر برنامه‌ای بنویسیم که اطلاعات را از پورت سریال دریافت کرده و نمایش دهد. برنامه زیر را ببینید.

```
#include <STDIO. H>
#include <CONIO. H>
#include <STDLIB. H>
void InitialSerialPort (void){
asm {
mov ah,04h
mov al,00h
mov bh,00h
mov bl,00h
mov ch,03h
mov cl,07h // 9600 bps
mov dx,00h
int 14h
}
}

int RecvData (){
unsigned char status;
asm {
mov ah,03h
mov dx,00h
int 14h
mov status,ah
}
status = status & 128;
if (status == 0)
return 0;
else
return 1;
}
```

```
void main (void){
unsigned char data;
InitialSerialPort ();
while (1){
while (RecvData () == 0 && !kbhit ());
if (kbhit ()) exit (0);
data = inportb (0x3F8);
printf ("%X\n",data);
}
}
```



به این خط برنامه توجه کنید:

```
While (RecvData () == 0 && !kbhit ());
```

تا زمانی که کاراکتری دریافت نشده (خروجی تابع RecvData صفر است) و نیز کاربر کلیدی را فشار نداده باشد، برنامه در همین نقطه متوقف می‌ماند. با نقض شدن یکی از این دو شرط، برنامه از این خط عبور کرده و ادامه می‌دهد. در خط بعدی، بررسی می‌شود که آیا در اثر فشردن کلیدی از خط قبلی عبور کرده ایم؟ اگر اینگونه باشد، برنامه به پایان می‌رسد. در غیر اینصورت، به کمک تابع inportb داده دریافتی خوانده شده و در دستگاه شانزده تایی روی نمایشگر کامپیوتر نشان داده می‌شود. به صورت خلاصه، برنامه بالا تا زمانی که کاربر کلیدی را فشار نداده باشد، اطلاعات دریافتی از پورت سریال را خوانده و روی نمایشگر کامپیوتر نشان می‌دهد.

## استفاده از پورت سریال به شیوه وقفه

فرض کنید می‌خواهیم در کامپیوتر برنامه ای بنویسیم که داده هایی را از پورت سریال کامپیوتر خوانده و نمایش دهد. یک راهکار این است که مرتباً بررسی کنیم آیا داده ای را در پورت سریال دریافت شده یا خیر و در صورت مثبت بودن پاسخ، آن را بخوانیم و نمایش دهیم (روش سرشماری Polling) این راه ساده است. اما وقت را بیهوده هدر می‌دهد. برنامه در هر ثانیه هزاران بار پورت سریال را کنترل می‌کند که آیا داده ای در آن دریافت شده است یا خیر؟ که به خصوص در مورد ارتباطاتی که حجم انتقال داده ها در آن زیاد نیست به صرفه نمی‌باشد. حداکثر سرعت ارتباط در این روش بدون از بین رفتن داده ها ۳۴,۷Kbps است.

راه دیگر این است که از پورت سریال در حالت وقفه استفاده کنیم. در این حالت، وقتی داده ای توسط پورت سریال دریافت شود وقفه ای رخ می‌دهد. می‌توانید با باز نویسی زیر برنامه پاسخ به آن وقفه (ISR)، واکنش لازم در قبال دریافت آن داده (مثلاً نمایش آن در کامپیوتر) را نشان دهید. با این کار دیگر نیازی نیست مرتباً پورت سریال را بررسی کنید. بلکه هرگاه داده ای دریافت شد، وقفه ای رخ خواهد داد. با این روش حتی در کامپیوتر های قدیمی سرعت به راحتی تا ۱۱۵,۲Kbps بالا می‌رود.

تغییر ISR یک وقفه در کامپیوتر چندان پیچیده نیست و می‌توان آن را به کمک برنامه های اسمبلی یا سطح بالا (مانند C) انجام داد. در کتاب های زبان اسمبلی در بخش برنامه های مقیم در حافظه را به پورت سریال اختصاص داده است. در شکل زیر که مربوط به یک کامپیوتر نوعی است و قبلاً شیوه مشاهده آن در ویندوز را دیدیم، وقفه (IRQ4) (INT 0C h) به پورت سریال اختصاص داده شده است.



وقفه IRQ4 به پورت های COM1 و COM3 و وقفه IRQ3 به پورت های COM2 و COM4 اختصاص دارد.

اما چگونه باید از سیستم خواست که با دریافت یک کاراکتر اعلام وقفه کند؟ مقدار دهی ثبات فعال کننده وقفه (از مجموعه ثبات های پورت سریال) این کار را انجام می‌دهد. بیت های این ثبات را ببینید:

0	0	0	0	IN	ER	TBE	RxRDY
---	---	---	---	----	----	-----	-------

اگر بیت RxRDY «یک» باشد، دریافت داده باعث ایجاد وقفه می‌شود.

اگر بیت TBE «یک» باشد، خالی شدن بافر ارسال داده باعث روی دادن وقفه می‌شود.

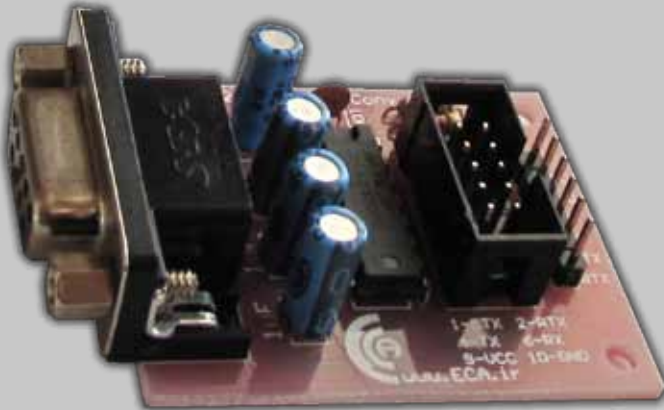
بیت ER در رابطه با وقفه وضعیت خط و بیت IN در رابطه با وقفه وضعیت مودم عمل می‌کند.



دستور 1, 0x3F8+1 (outportb); وقفه دریافت داده را فعال می کند. از این پس با دریافت داده توسط پورت سریال وقفه ای رخ می دهد.

منبع : کتاب اتصال ورودی/خروجی به پورت سریال - جواد راستی

## ماژول RS232 to TTL



یکی از پروتکل هایی که میکروکنترلر توسط آن می تواند با رایانه ارتباط برقرار کند استفاده از پروتکل سریال می باشد. سطح ولتاژ بیشتر میکروکنترلرها TTL بوده و به همین علت بصورت مستقیم قابل اتصال به پورت COM رایانه نمی باشند. به همین جهت بین پورت RS232 رایانه و میکروکنترلر از مداراتی استفاده می نمایند تا این مشکل را رفع نمایند.

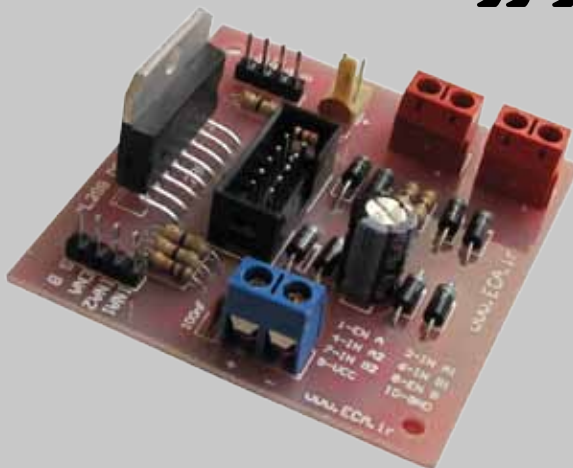
ماژول RS232 to TTL اتصال تمامی میکروکنترلرها به پورت سریال را ساده می نماید.

ولتاژ راه اندازی ماژول 5V DC بوده و تمام سیگنال های Rx، Tx، RTS، CTS و DTR با اینترفیس TTL قابل دسترسی هستند. قابل راه اندازی با تمام Baud Rate های استاندارد.

توضیحات تکمیلی :

<http://eshop.eca.ir/link/87106.php>

## درایور راه انداز موتور L298



یکی از کارهای که می توان با میکروکنترلر انجام داد کنترلر سرعت و یا تغییر جهت انواع موتورهای الکتریکی می باشد. میکروکنترلرها به علت کم بودن جریان خروجی به تنهایی قادر به راه اندازی موتورهای الکتریکی نبوده و از درایورها جهت کنترل موتورهای الکتریکی استفاده می نمایند. یکی از درایورهای پرکاربرد جهت استفاده برای موتورهای الکتریکی درایور L298 است که قادر به کنترل 2 موتور با ولتاژ 50V و حداکثر جریان 2A می باشد. ماژول فوق با استفاده از درایور موتور L298 طراحی گردیده و توانایی کنترل همزمان 2 موتور DC را دارد.

حداکثر جریان قابل تامین برای هر موتور : 2A

حداکثر ولتاژ قابل تامین هر موتور : 50V

قابل راه اندازی با تمام میکروکنترلرها

توضیحات تکمیلی :

<http://eshop.eca.ir/link/87106.php>

## LabVIEW 2011 Professional +AddOns



نرم افزار Lab View که مخفف عبارت Laboratory Virtual Instrument Engineering Workbench می باشد، یک زبان برنامه نویسی گرافیکی می باشد که به صورت گسترده ای برای کاربرد های مختلفی در صنایع، تحصیلات، آموزش و تحقیقات آزمایشگاهی به عنوان یک مدل استاندارد برای جمع آوری و پردازش داده ها و همچنین وسیله ای جهت کنترل و شبیه سازی ابزارهای مجازی درآمده است. این برنامه یک نرم افزار قدرتمند و قابل انعطاف جهت تجزیه و تحلیل سیستم های اندازه گیری است.

عملکرد نرم افزار LabView کاملاً از طبیعت ترتیبی و زنجیره ای موجود زبان های برنامه نویسی متنی متداول و مرسوم مجزاست و یک محیط گرافیکی را برای کاربر فراهم ساخته است. در این راه از تمامی ابزارهای لازم جهت جمع آوری، پردازش و تحلیل داده ها و نمایش نتایج استفاده می شود. به کمک این زبان برنامه نویسی گرافیکی که با "G" نشان داده می شود، در برنامه ی نوشته شده، از یک نمودار بلوکی استفاده می شود و سپس این نمودار به کدهای ماشین تبدیل می گردد. این نرم افزار برای موارد بی شماری از کاربردهای علمی و مهندسی، ایده آل و عملی است و به شما کمک می کند تا مسائل و مشکلات موجود در برنامه نویسی را در مدت زمان کوتاهی حل کنید.

توضیحات تکمیلی :

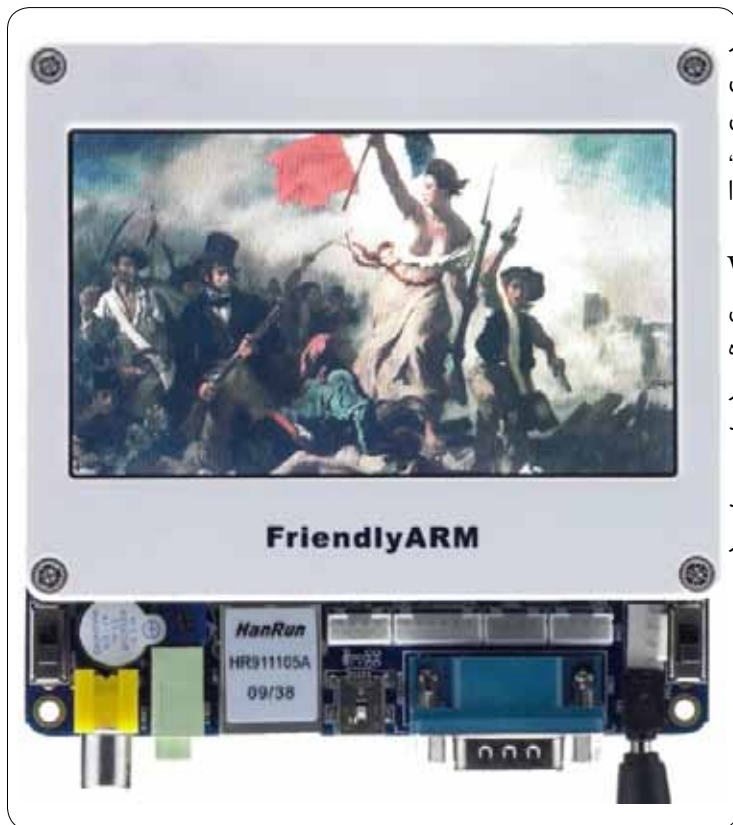
<http://eshop.eca.ir/link/953.php>

# برد Mini6410 + 4.3" LCD

این برد بر اساس توانایی‌ها و قابلیت‌های میکروپروسسور S3C6410A-ARM 11 طراحی شده است. این برد در اصل Embedded MotherBoard می‌باشد، از این رو که با فرکانس کاری ۵۳۳ مگاهرتز و داشتن پورت‌های همچون سریال، USB، صدا، شبکه، حافظه SD/MMC، VGA و... توانایی‌های یک مادر برد را دارا می‌باشد.

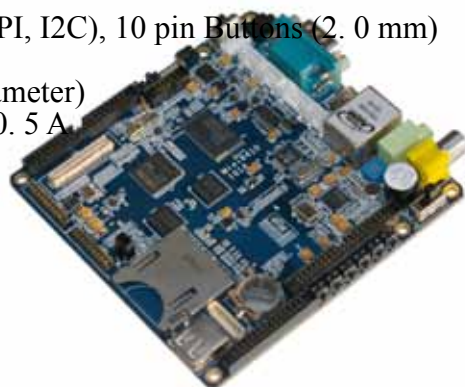
همچنین امکان نصب سیستم عامل‌های Windows CE، NET و Linux بر روی مادر برد به راحتی امکان پذیر می‌باشد. داشتن نمایشگر LCD رنگی ۴.۳" با دقت ۴۸۰\*۲۷۲ پیکسل به همراه صفحه حسگر لمسی و امکان اتصال برد مبدل VGA برای نمایشگرهای CRT و LCD معمولی از دیگر قابلیت‌های جذاب این برد می‌باشد.

ابعاد این برد ۱۱\*۱۱ سانتی متر می‌باشد. ابعاد بسیار کوچک این برد جهت کاربردهای جیبی (Pocotabl) و فضاهایی که نیاز به یک مادر برد با سایز بسیار کوچک است، بسیار مناسب می‌باشد..



## Specification

- Dimension: 110 x 110 mm
- CPU: 533 MHz Samsung S3C6410A ARM1176JZF-S with VFP-Unit and Jazelle (max freq. 667 MHz)
- RAM: 128 MB / 256 MB DDR RAM, 32 bit Bus
- Flash: 256 MB / 1GB NAND Flash
- EEPROM: 256 Byte (I2C)
- Ext. Memory: SD-Card socket
- Serial Ports: 1x DB9 connector (RS232), total: 4x serial port connectors
- IR: Infrared Receiver
- USB: 1x USB-A Host 1. 1, 1x miniUSB Slave/OTG 2. 0
- Audio Output: 3. 5 mm stereo jack
- Audio Input: Condenser microphone
- Ethernet: RJ-45 10/100M (DM9000)
- RTC: Real Time Clock with battery
- Beeper: PWM buzzer
- Camera: 20 pin (2. 0 mm) Camera interface
- TV Output
- LCD Interface: 40 pin (2. 0 mm) and 41 pin (1. 0 mm) connector for FriendlyARM Displays
- Touch Panel: 4 wire resistive
- User Inputs: 8x push buttons and 1x A/D pot
- User Outputs: 4x LEDs
- Expansion: 40 pin System Bus, 30 pin GPIO, 20 pin SDIO (SD, SPI, I2C), 10 pin Buttons (2. 0 mm)
- Debug: 10 pin JTAG (2. 0 mm)
- Power: regulated 5V (DC-Plug: 1. 35mm inner x 3. 5mm outer diameter)
- Power Consumption: Mini6410: 0. 25 A, Mini6410 + 4. 3" LCD: 0. 5 A
- OS Support
  - o Windows CE 6
  - o Linux 2. 6
  - o Android
  - o Ubuntu



توضیحات تکمیلی : <http://eshop.eca.ir/link/6410.php>

## Edwin Howard Armstrong

رادیو FM را «ادوین هوارد آرمسترانگ» اختراع کرد. اما سال ها طول کشید تا این باند رادیویی به صورت گسترده مورد استفاده قرار گیرد و ردیفی از اعداد نشان گر موج را در گیرنده های کوچک خانگی به خود اختصاص دهد. خود آرمسترانگ نیز زندگی حرفه ای پر فراز و نشیبی را گذراند و برای به کرسی نشاندن حرف خود زحمات بسیار کشید و صدمات فراوان دید. ماجرای او و رادیو FM، شباهت عجیبی به داستان نیکلا تسلا و اختراع خود رادیو دارد. ادوین آرمسترانگ در ۱۸ دسامبر ۱۸۹۰ در شهر نیویورک به دنیا آمد و مدرک لیسانس خود را در رشته مهندسی برق از دانشگاه کلمبیا دریافت کرد. نبوغ او در زمینه دانش الکترونیک که با آینده بینی خاصی تلفیق شده بود، دیدگاهی بسیار فراتر از زمانه زندگی اش به او داده بود. آرمسترانگ جز رادیو FM، اختراعات دیگری نیز داشت، به گونه ای که امروزه او را پدر بزرگ رادار و دنیای ارتباطات ماهواره ای می دانند. زمانی که او دانشجوی سال سوم دانشگاه کلمبیا بود، Regenerative circuit را اختراع کرد و در سال ۱۹۱۴ به ثبت رساند. در ۱۹۱۸ نوعی گیرنده موسوم به Super Heterodyne و در ۱۹۲۲ نیز «مدار ابرباز تولیدگر» (Super regenerative circuit) را به ثبت رساند. اما بسیاری از اختراعات او بعدها مورد ادعای دیگری قرار گرفت که در دادگاه ها رأی را به نفع خود تغییر دادند و امتیاز اختراعات او را به نام خود به

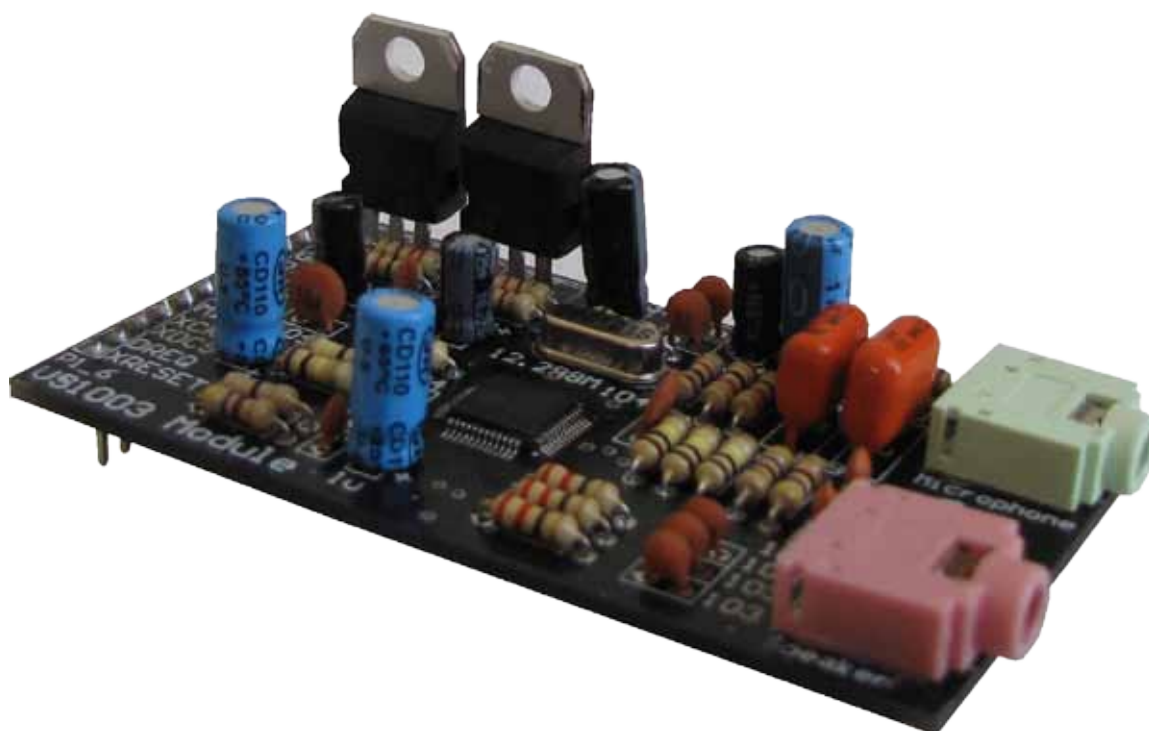
ثبت رساندند. بین سال های ۱۹۲۲ تا ثبت حق اختراعات خود بود: از یک و شرکت وستینگهاوس، و از سوی خلاء سه قطبی) و شرکت AT&T. که ۱۲ سال ادامه داشت، طولانی ترین بود. دادگاه عالی آمریکا حق ثبت مدار فارست داد که او هم آن را به شرکت دادگاه را ناشی از ناآگاهی اعضای آن هنگام که این دادگاه در جریان بود، که همان رادیو FM باشد به سرانجام ثبت شد. گیرنده های رادیویی FM بر آن روزها بودند، صدایی بسیار صاف برای اثبات کارایی و کیفیت رادیو FM. FCC (کمیته ارتباطات فدرال، مسئول داشت و آن را واداشت تا یک باند تا ۴۹ مگاهرتز، به آن اختصاص دهد. پیش از جنگ جهانی دوم و در میانه FM پرتوان را در ایالت های نیوانگلند شد. آرمسترانگ سفری را برای اثبات خیلی بهتر از AM است، آغاز کرد و رادیو با فناوری FM، حق امتیاز خود رادیو آمریکا برای اختصاص فرکانس



های الکترومغناطیسی به صنعت نو پای تلویزیون (که خود متولی آن بود)، FCC را حسابی تحت فشار قرار داده بود. دیوید سرنف (مدیر کل RCA) در نهایت توانست FCC را وادارد تا طیف رادیو FM را از ۴۲ تا ۴۹ مگاهرتز به ۸۸ تا ۱۰۸ مگاهرتز (همان که امروزه هست) منتقل کند و گستره فرکانس های ۴۰ مگاهرتزی را برای تلویزیون (که برای پخش صدای خود به باند FM نیاز داشت) خالی کند. در نتیجه این کار، همه گیرنده های رادیویی FM آرمسترانگ یک شبه بی مصرف شدند در حالی که جایگاه قوی رادیو AM شرکت رادیو آمریکا، قوی تر از قبل شد. شبکه رادیویی آرمسترانگ به این ترتیب در برابر انتقال به فرکانس های بالاتر نتوانست دوام بیاورد. اغلب متخصصان امروزه بر این عقیده اند که این تصمیم FCC، بهره برداری از فناوری FM را چندین دهه عقب انداخت. علاوه بر این، RCA در نهایت با ادعای حق ثبت فناوری FM به نام خود، در دادگاه برنده شد و آرمسترانگ دیگر نتوانست از رادیوهای FM و دستگاه های تلویزیون که در آمریکا فروخته می شد حق امتیاز بگیرد. نابود شدن «شبکه یانکی» و دعوا بر سر حق امتیاز، آرمسترانگ را بی پول کرد و روان او را فرو پاشاند. در این شرایط، آرمسترانگ در ۳۱ ژانویه ۱۹۵۴ با پرت کردن خود از پنجره آپارتمان خودکشی کرد. ماریون، همسر دوم و حالا بیوه او، دعوای حق ثبت FM را علیه RCA دوباره به جریان انداخت و سرانجام در سال ۱۹۶۷ برنده شد. اما چندین دهه از مرگ آرمسترانگ گذشت تا رادیو FM توانست از لحاظ گستره پخش به پای AM برسد، و باز بسیار گذشت تا رادیو FM برای ایستگاه های پخش به صرفه و سودآور شد. آرمسترانگ در سال ۱۹۱۷ به عنوان اولین برنده مدال افتخار IRE (انجمن مهندسان رادیو که امروزه «انجمن مهندسان برق و الکترونیک» IEEE خوانده می شود) برگزیده شد. در سال ۱۹۴۲ مدال ادیسون AIEE (انجمن مهندسان برق آمریکا) را «برای کمک های برجسته خود به هنر ارتباطات الکترونیکی» دریافت کرد و در سال ۱۹۸۰ نیز نام او به «تالار افتخار مخترعان» در آمریکا اضافه شد.



# ماژول ضبط و پخش صدا VS1003



VS1003 یک تراشه همه کاره دکودر MP3 و WMA می باشد. این تراشه قابلیت ضبط صدا با کیفیت بالا و در باند وسیع را داراست. همچنین می تواند MIDI را هم از طریق فایل و هم با استفاده از MIDI keyboard سریال استاندارد پخش نماید.

امکانات ماژول :

قابلیت اتصال به تمام میکروکنترلرها از طریق رابط SPI  
تامین ولتاژهای مورد نیاز تراشه و راه اندازی کل ماژول با تک ولتاژ ۵ ولت  
دارای خروجی هدفن به صورت استریو  
دارای ورودی میکروفن به صورت مونو  
ایزوله نمودن مناسب در برابر نویز و تغییرات ولتاژ، به جهت بالا بودن کیفیت ضبط و پخش

امکانات تراشه :

Decodes MP3 (MPEG 1 & 2 audio layer III) (CBR +VBR +ABR);  
WMA 4.0/4.1/7/8/9 all profiles (5-384kbit/s);  
WAV (PCM + IMA ADPCM);  
General MIDI / SP-MIDI files  
Encodes IMA ADPCM from microphone or line input  
Streaming support  
Bass and treble controls  
Operates with a single 12 - 13 MHz or 24 - 26MHz clock  
Internal PLL clock multiplier  
Low-power operation  
High-quality on-chip stereo DAC with no phase error between channels  
Stereo earphone driver capable of driving a 30 ohm load  
Separate operating voltages for analog, digital and I/O  
5.5 KiB On-chip RAM for user code / data  
Serial control and data interfaces  
Can be used as a slave co-processor  
SPI flash boot for special applications  
UART for debugging purposes  
New functions may be added with software and 4 GPIO pins

توضیحات تکمیلی : <http://eshop.eca.ir/link/973.php>



## Transistor Wars

### Rival architectures face off in a bid to keep Moore's Law alive

In May, Intel announced the most dramatic change to the architecture of the transistor since the device was invented. The company will henceforth build its transistors in three dimensions, a shift that—if all goes well—should add at least a half dozen years to the life of Moore's Law, the biennial doubling in transistor density that has driven the chip industry for decades.

But Intel's big announcement was notable for another reason: It signaled the start of a growing schism among chipmakers. Despite all the great advantages of going 3-D, a simpler alternative design is also nearing production. Although it's not yet clear which device architecture will win out, what is certain is that the complementary metal-oxide semiconductor (CMOS) field-effect transistor (FET)—the centerpiece of computer processors since the 1980s—will get an entirely new look. And the change is more than cosmetic; these designs will help open up a new world of low-power mobile electronics with fantastic capabilities.

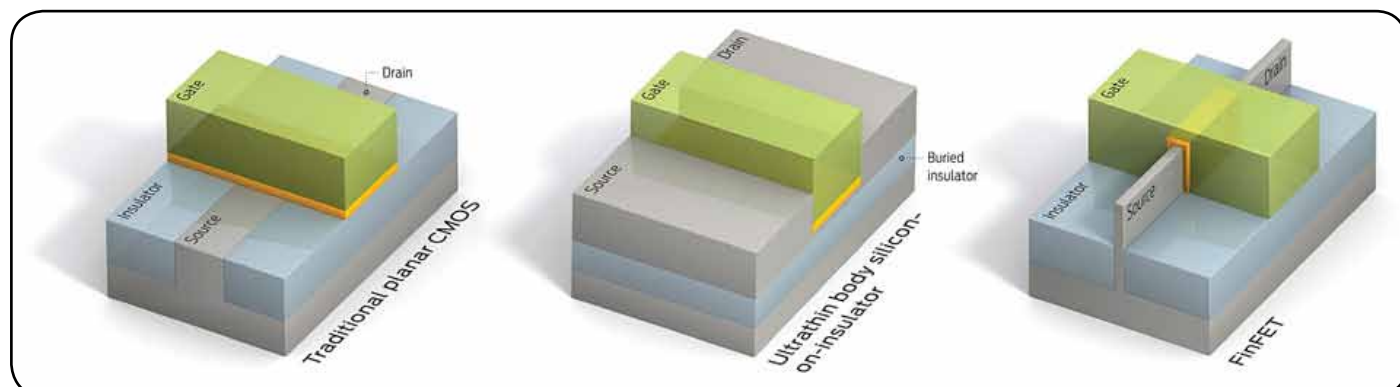


There's a simple reason everyone's contemplating a redesign: The smaller you make a CMOS transistor, the more current it leaks when it's switched off. This leakage arises from the device's geometry. A standard CMOS transistor has four parts: a source, a drain, a channel that connects the two, and a gate on top to control the channel. When the gate is turned on, it creates a conductive path that allows electrons or holes to move from the source to the drain. When the gate is switched off, this conductive path is supposed to disappear. But as engineers have shrunk the distance between the source and drain, the gate's control over the transistor channel has gotten weaker. Current sneaks through the part of the channel that's farthest from the gate and also through the underlying silicon substrate. The only way to cut down on leaks is to find a way to remove all that excess silicon. Over the past few decades, two very different solutions to this problem have emerged. One approach is to make the silicon channel of the traditional planar transistor as thin as possible, by eliminating the silicon substrate and instead building the channel on top of insulating material. The other scheme is to turn this channel on its side, popping it out of the transistor plane to create a 3-D device. Each approach comes with its own set of merits and manufacturing challenges, and chipmakers are now working out the best way to catch up with Intel's leap forward. The next few years will see dramatic upheaval in an already fast-moving industry.

**Change is nothing new** to CMOS transistors, but the pace has been accelerating. When the first CMOS devices entered mass production in the 1980s, the path to further miniaturization seemed straightforward. Back in 1974, engineers at the IBM T. J. Watson Research Center in Yorktown Heights, N.Y., led by Robert Dennard, had already sketched out the ideal progression. The team described how steadily reducing gate length, gate insulator thickness, and other feature dimensions could simultaneously improve switching speed, power consumption, and transistor density.

But this set of rules, known as Dennard's scaling law, hasn't been followed for some time. During the 1990s boom in personal computing, the demand for faster microprocessors drove down transistor gate length faster than Dennard's law called for. Shrinking transistors boosted speeds, but engineers found that as they did so,

they couldn't reduce the voltage across the devices to improve power consumption. So much current was being lost when the transistor was off that a strong voltage—applied on the drain to pull charge carriers through the channel—was needed to make sure the device switched as quickly as possible to avoid losing power in the switching process.



**ELIMINATING EXCESS:** In the next few years, traditional planar CMOS field-effect transistors [left] will be replaced by alternate architectures that boost the gate's control of the channel. The UTB SOI [center] replaces the bulk silicon channel with a thin layer of silicon mounted on insulator. The FinFET [right] turns the transistor channel on its side and wraps the gate around three sides. © 2011 IEEE Spectrum magazine

By 2001, the leakage power was fast approaching the amount of power needed to switch a transistor out of its "off" state. This was a warning sign for the industry. The trend promised chips that would consume the same amount of energy regardless of whether they were in use or not. Chipmakers needed to find new ways to boost transistor density. In 2003, as the length of transistor channels dropped to 45 nanometers, Intel debuted chips bearing devices made with strain engineering. These transistors boasted silicon channels that had been physically squeezed or pulled to boost speed and reduce the power lost due to resistance. By the next "node"—industry lingo for a transistor density milestone—companies had stopped shrinking transistor dimensions and instead began just squeezing transistors closer together. And in 2007, Intel bought Moore's Law a few more years by introducing the first big materials change, replacing the ever-thinning silicon oxide insulator that sits between a transistor's gate and channel with hafnium oxide.

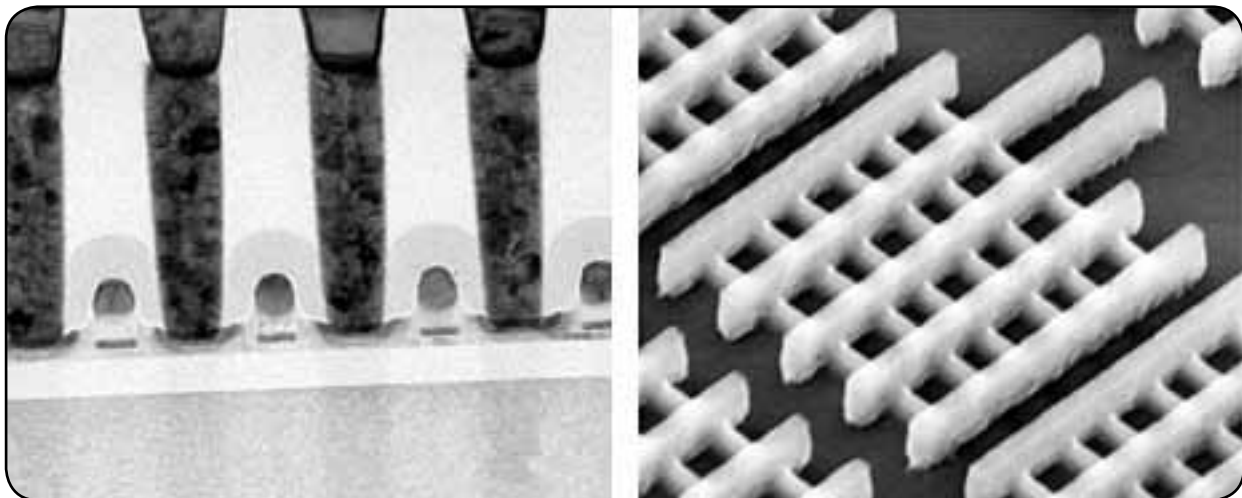
This better-insulating material helped stanch a main source of leakage current—the tunneling of electrons between the gate and the channel. But leakage from the source to the drain was still a huge problem. As companies faced the prospect of creating even denser chips with features approaching 20 nm, it became increasingly clear that squeezing together traditional planar transistors or shrinking them even further would be impossible with existing technology. Swapping in a new insulator or adding more strain wouldn't cut it. Driving down power consumption and saving Moore's Law would require a fundamental change to transistor structure—a new design that could maximize the gate's control over the channel.

**Fortunately, over the course** of more than 20 years of research, transistor designers have found two very powerful ways to boost the effectiveness of the transistor gate. As the gate itself can't get much stronger, these schemes focus on making the channel easier to control. One approach replaces the bulk silicon of a normal transistor with a thin layer of silicon built on an insulating layer, creating a device that is often called an ultra-thin body silicon-on-insulator, or UTB SOI, also known as a fully depleted SOI.

A second strategy turns the thin silicon channel by 90 degrees, creating a "fin" that juts out of the plane of the device. The transistor gate is then draped over the top of the channel like an upside-down U, bracketing it on three sides and giving the gate almost complete control of the channel. While conventional CMOS devices are largely flat, save for a thin insulating layer and the gate, these FinFETs—or Tri-Gate transistors, as Intel has named its three-sided devices—are decidedly 3-D. All the main components of the transistor—source, drain, channel, and gate—sit on top of the device's substrate.

Both schemes offer the same basic advantage: By thinning the channel, they bring the gate closer to the drain. When a transistor is off, the drain's electric field can take one of two paths inside the channel to zero-voltage

destinations. It can propagate all the way across the channel to the source, or it can terminate at the transistor's gate. If the field gets to the source, it can lower the energy barrier that keeps charge carriers in the source from entering the channel. But if the gate is close enough to the drain, it can act as a lightning rod, diverting field lines away from the source. This cuts down on leakage, and it also means that field lines don't penetrate very far into the channel, dissipating even more energy by tugging on any stray carriers.



DOWN AND UP: A cross section of UTB SOI transistors [left] and a micrograph of an array of FinFET transistors

The first 3-D transistor was sketched out by Digh Hisamoto and others at Hitachi, who presented the design for a device dubbed a Delta at a conference in 1989. The UTB SOI's roots extend even further back; they are a natural extension of early SOI channel research, which began in the 1980s when researchers started experimenting with transistors built with 200-nm thick, undoped silicon channels on insulating material.

But the promise of both of these thin-channel approaches wasn't fully appreciated until 1996, when Chenming Hu and his colleagues at the University of California, Berkeley, began an ambitious study, funded by the U.S. Defense Advanced Research Projects Agency, to see how far these designs could go. At the time, the industry was producing 250-nm transistors, and no one knew whether the devices could be scaled below 100 nm. Hu's team showed that the two alternate architectures could solve the power consumption problems of planar CMOS transistors and that they could operate with gate lengths of 20 nm—and later, even less.

The FinFET and the UTB SOI both offer big gains in power consumption. Logic chip designs typically require that a transistor in its on state draw at least 10 000 times as much current as the device leaks in its off state. For 30-nm transistors—about the size that most chipmakers are currently aiming for—this design spec means devices should leak no more than a few nanoamperes of current when they're off. While 30-nm planar CMOS devices leak about 50 times that amount, both thin-channel designs hit the target quite easily.

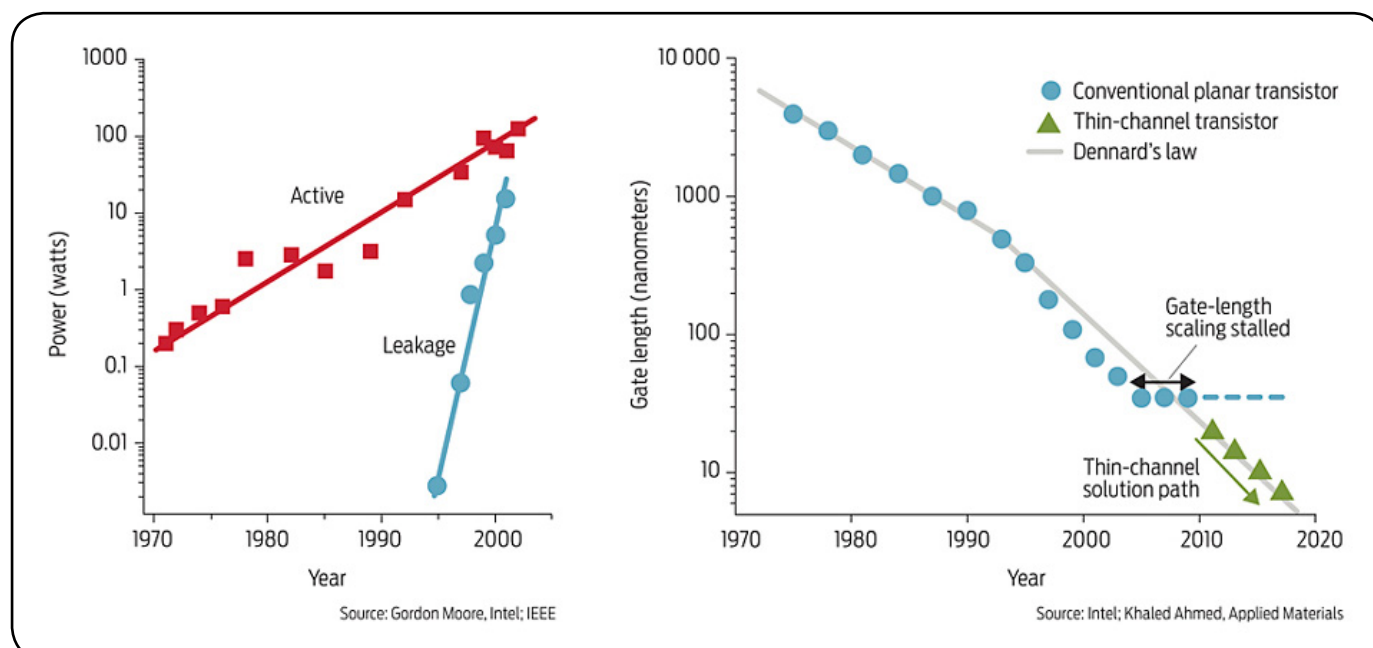
But the two architectures aren't entirely equal. To get the best performance, the channel of a UTB SOI should be no more than about one-fourth as thick as the length of the gate. Because a FinFET's gate brackets the channel on three sides, the 3-D transistors can achieve the same level of control with a channel—or fin—that's as much as half as thick as the length of the transistor gate.

This bigger channel volume gives FinFETs a distinct advantage when it comes to current-carrying capacity. The best R&D results suggest that a 25-nm FinFET can carry about 25 percent more current than a UTB SOI. This current boost doesn't matter much if you have only a single transistor, but in an IC, it means you can charge capacitors 25 percent faster, making for much speedier chips. Faster chips obviously mean a lot to a microprocessor manufacturer like Intel. The question is whether other chipmakers will find the faster speeds



meaningful enough to switch to FinFETs, a prospect that requires a big up-front investment and an entirely new set of manufacturing challenges.

The single biggest hurdle in making FinFETs is manufacturing the fins so that they're both narrow and uniform. For a 20-nm transistor—roughly the same size as the one that Intel is putting into production—the fin must be about 10 nm wide and 25 nm high; it must also deviate by no more than half a nanometer—just a few atomic layers—in any given direction. Over the course of production, manufacturers must control all sources of variation, limiting it to no more than 1 nm in a 300-millimeter-wide wafer.



Shrinking returns: As transistors got smaller, their power demands grew. By 2001, the power that leaked through a transistor when it was off was fast approaching the amount of power needed to turn the transistor on [left], a warning sign for the chip industry. As these Intel data show, the leakage problem eventually put a halt to the transistor scaling [right], a progression called Dennard's law. Switching to alternate architectures will allow chipmakers to shrink transistors again, boosting transistor density and performance.

This precision is needed not only to manufacture the fin; it must also be maintained for the rest of the manufacturing process, including thermal treatment, doping, and the multiple film deposition and removal steps needed to build the transistor's gate insulator and gate. As an added complication, the gate oxide and the gate must be deposited so that they follow the contours of the fin. Any process that damages the fin could affect how the device performs. The resultant variation in device quality would force engineers to operate circuits at a higher power than they're designed for, eliminating any gains in power efficiency.

The unusual geometry of the FinFET also poses challenges for doping, which isn't required but can help cut down on leakage current. FinFET channels need two kinds of dopants: One is deposited underneath the gate and the other into the parts of the channel that extend on either side of the gate, helping mate the channel to the source and drain. Manufacturers currently dope channels by shooting ions straight down into the material. But that approach won't work for FinFETs. The devices need dopants to be distributed evenly through the top of the fin and the side walls; any unevenness in concentration will cause a pileup of charges, boosting the device's resistance and wasting power.

Doping will get only more difficult in the future. As FinFETs shrink, they'll get so close together that they will cast "shadows" on one another, preventing dopants from permeating every part of every fin. At Applied Materials' Silicon Systems Group, we've been working on one possible fix: immersing fins in plasma so that dopants can migrate directly into the material, no matter what its shape is.

Because UTB SOI devices are quite similar to conventional planar CMOS transistors, they are easier to manufacture than FinFETs. Most existing designs and manufacturing techniques will work just as well with the



new thin-silicon transistors as they do with the traditional variety. And in some ways, UTB SOIs are easier to produce than present-day transistors. The devices don't need doped channels, a simplification that can save planar CMOS manufacturers some 20 to 30 steps out of roughly 400 in the wafer production process.

But the UTB SOI comes with its own challenges, chiefly the thin channel. The requirement that UTB SOI channels be half as thick as comparable FinFET fins makes any variations in thickness even more critical for these devices. A firm called Soitec, headquartered in Bernin, France, which has been leading the charge in manufacturing ultrathin silicon-on-insulator wafers, is currently demonstrating 10-nm-thick silicon layers that vary by just 0.5 nm in thickness. That's an impressive achievement for wafers that measure 300 mm across, but it will need to be improved as transistors shrink. And it's not clear how precise Soitec's technique—which involves splitting a wafer to create an ultrathin silicon layer—can ultimately be made.

Another key stumbling block for UTB SOI adoption is the supply chain. At the moment, there are few potential providers of ultrathin SOI wafers, which could ultimately make manufacturers of UTB SOI chips dependent on a handful of sources. Intel's Mark Bohr says the hard-to-find wafers could add 10 percent to the cost of a finished wafer, compared to 2 to 3 percent for wafers bearing 3-D transistors (an estimate from the SOI Industry Consortium suggests that finished UTB SOI wafers will actually be less expensive).

**Going forward, we expect** that chipmakers will split into two camps. Those interested in the speediest transistors will move toward FinFETs. Others who don't want to invest as much in a switch will find UTB SOIs more attractive.

UTB SOI transistors have an additional feature that makes them particularly appealing for low-power applications: A small voltage can easily be applied to the very bottom of a chip full of UTB SOI devices. This small bias voltage alters the channel properties, reducing the electrical barrier that stops current flowing from the source to the drain. As a result, less voltage needs to be applied to the transistor gates to turn the devices on. When the transistors aren't needed, this bias voltage can be removed, which restores the electrical barrier, reducing the amount of current that leaks through the device when it's off. As Thomas Skotnicki of STMicroelectronics has long argued, this sort of dynamic switching saves power, making the devices particularly attractive for chips in smartphones and other mobile gadgets. Skotnicki says the company expects to release its first UTB SOI chip, which will use 28-nm transistors to power a mobile multimedia processor, by the end of 2012. That said, few companies have committed to one technology or the other. STMicroelectronics—as well as firms such as GlobalFoundries and Samsung—is part of the International Semiconductor Development Alliance, which supports and benefits from device research at IBM and is investing in both FinFETs and UTB SOIs. Exactly how the industry will split up and which design will come to dominate will depend on decisions made by the biggest foundries and how quickly standards are developed. Reports suggest that Taiwan Semiconductor Manufacturing Co., which dominates bespoke manufacturing in the chip industry, will begin making 14-nm FinFETs in 2015, but it's not clear whether the company will also support UTB SOI production. Switching to FinFET production requires a substantial investment, and whichever way TSMC swings, it will put pressure on other manufacturers, such as GlobalFoundries, United Microelectronics Corp., and newcomers to the foundry business such as Samsung, to choose a direction.

Also still unclear is how far each technology can be extended. Right now it looks like both FinFETs and UTB SOIs should be able to cover the next three generations of transistors. But UTB SOI transistors may not evolve much below 7 nm, because at that point, their gate oxide would need an effective thickness of 0.7 nm, which would require significant materials innovation. FinFETs may have a similar limit. In 2006, a team at the Korea Advanced Institute of Science and Technology used electron-beam lithography to build 3-nm FinFETs. But crafting a single device isn't quite the same as packing millions together to make a microprocessor; when transistors are that close to each other, parasitic capacitances and resistances will draw current away from each

switch. Some projections suggest that when FinFETs are scaled down to 7 nm or so, they will perform no better than planar devices.

Meanwhile, researchers are already trying to figure out what devices might succeed FinFETs and UTB SOIs, to continue Moore's Law scaling. One possibility is to extrapolate the FinFET concept by using a nanowire device that is completely surrounded by a cylindrical gate. Another idea is to exploit quantum tunneling to create switches that can't leak current when they're not switched on. We don't know what will come next. The emergence of FinFETs and UTB SOIs clearly shows that the days of simple transistor scaling are long behind us. But the switch to these new designs also offers a clear demonstration of how creative thinking and a good amount of competition can help us push Moore's Law to its ultimate limit—whatever that might be.

## About the Authors

Khaled Ahmed is an IEEE senior member and a technology strategist in the Silicon Systems Group at Applied Materials, where Klaus Schuegraf is chief technology officer. After years of working on ways to manufacture smaller and smaller bulk silicon transistors, both are excited about the rapid development of the new, thin-channel alternatives. "From a device physics perspective, thin is definitely in," Ahmed says.

## Keil Full Pack 2011

نرم افزار KEIL یکی از برترین کامپایلر های مخصوص پردازنده های ARM می باشد. این محصول حاوی یک کامل این نرم افزار شامل نرم افزارهای MDK-ARM V4.22a ، C251 V5.07b ، C51 V9.05 و C166 V7.05 سال 2011 می باشند.

این محصول در قالب یک یک کامل به همراه لایسنس معتبر ارائه گشته است.

توضیحات تکمیلی :

<http://eshop.eca.ir/link/959.php>

## مجموعه 16 ساله 1995~2010 elektor

این مجله یکی از معروفترین و معتبرترین منابع اطلاعاتی برای تمامی مهندسين برق و الكترونیک بوده و همراه به چند زبان مختلف انتشار می یابد . این مجله پرفروشترین و پرمحتواترین مجله الكترونیک در سطح بین المللی بوده که توانسته میلیونها مخاطب را به خود جذب کرده و هر ماه تعداد کسیری از متخصصین این رشته منتظر انتشار نسخه جدید این مجله هستند . محتوای مجلات شامل انواع مدارات الكترونیک و طراحی آنها به همراه مدارات ۱۰۰٪ عملی و جذاب ، اخبار و نوآوری های جدید در زمینه الكترونیک ، مقالات علمی ، نکات کاربردی و فنی و هزاران مطلب دیگر در الكترونیک می باشد . این مجموعه آرشيو ۱۶ ساله این مجلات از سال ۱۹۹۵ تا پایان سال ۲۰۱۰ بوده و به جهت دارا بودن هزاران مدار کاربردی ، مطالب علمی و .... در زمینه الكترونیک می تواند برای شما همانند یک بانک اطلاعاتی عظیم و متنوع در زمینه الكترونیک باشد .

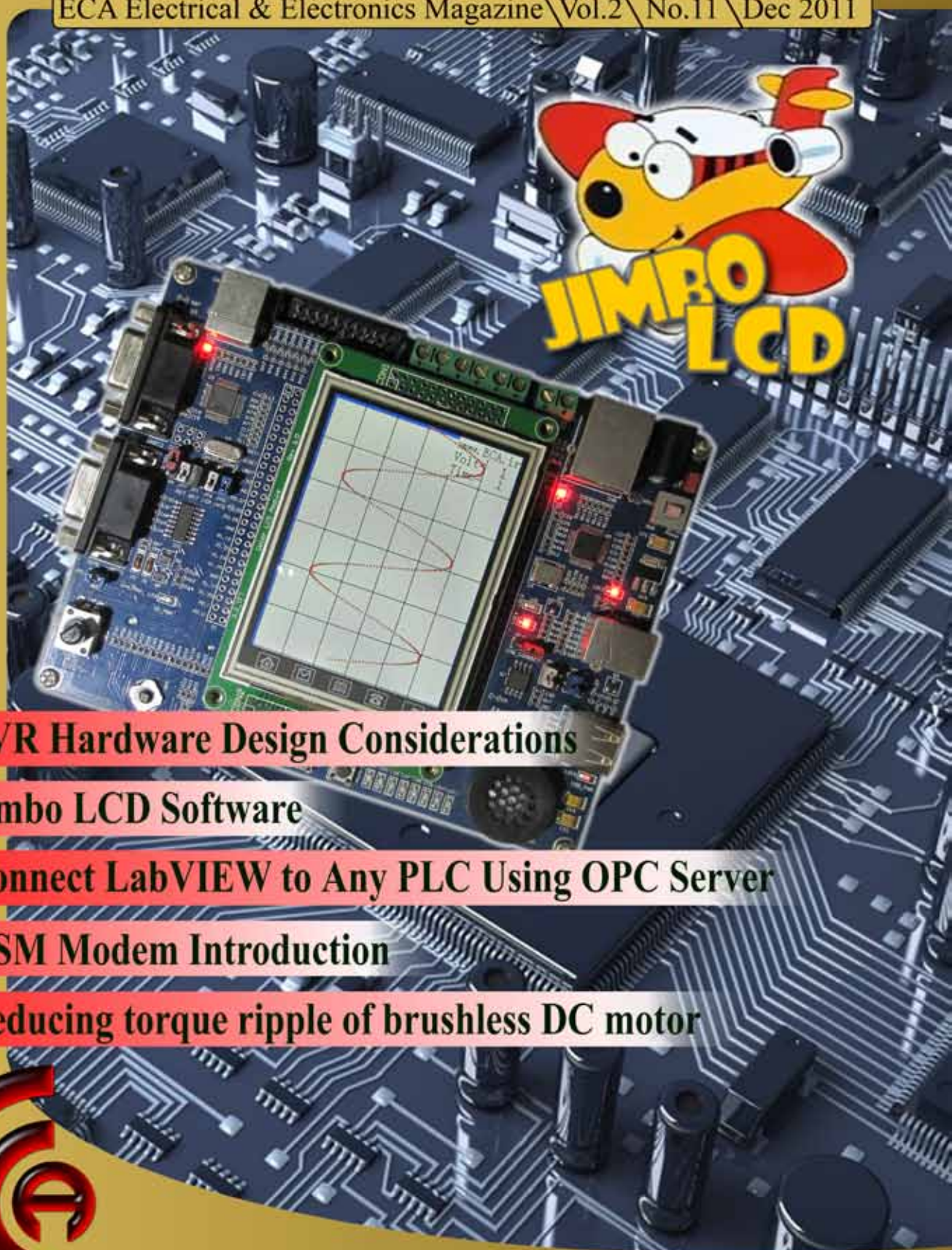
توضیحات تکمیلی :

<http://eshop.eca.ir/link/955.php>



# نویز Noise

ECA Electrical & Electronics Magazine Vol.2 No.11 Dec 2011



- AVR Hardware Design Considerations
- Jimbo LCD Software
- Connect LabVIEW to Any PLC Using OPC Server
- GSM Modem Introduction
- Reducing torque ripple of brushless DC motor



[www.Noisemagazine.ECA.ir](http://www.Noisemagazine.ECA.ir)