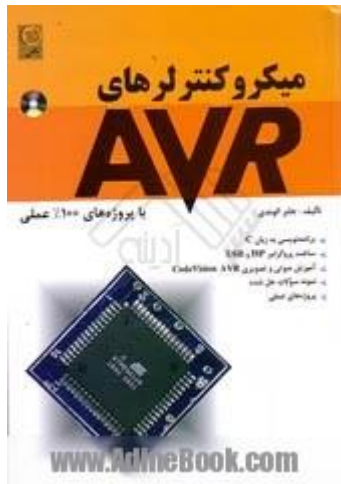


کتاب PDF



میکرو کنترلرهای AVR با پروژه‌های ۱۰۰٪ عملی

تألیف جابر الوندی

منبع درس ریزپردازنده‌ا دانشگاه پیام نور



فایل PDF موجود نسخه RF، در بین سایر فایل‌های PDF ریزپردازنده موجود در سایت‌های دانلود کتاب، کامل‌ترین فایل PDF محسوب می‌شود. چرا که در سایر کتاب‌های PDF مشابه، علاوه بر ناقص بودن صفحات کتاب، نظم و ترتیب خاصی بین صفحات وجود ندارد که این نقص، باعث ایجاد مشکل و سردرگمی برای دانشجویان در هنگام مطالعه کتاب PDF می‌شود.

از همین رو ویرایش RF کتاب پی‌دی‌اف ریزپردازنده را که نواقص فایل‌های قبلی را برطرف کرده، برای استفاده دانشجویان عزیزی که به کتاب اصلی این درس دسترسی ندارند، آماده کرده ایم.

Edited by Aref.b

4123ph@gmail.com



فصل

آموزش نرم افزار CodeVisionAVR

اهداف

۱. آشنایی با محیط برنامه نویسی به زبان C نرم افزار CodeVisionAVR
۲. آشنایی با ابزار CodeWizardAVR نرم افزار کامپایلر C
۳. معرفی پروگرامر ISP و نحوه ی برنامه ریزی میکروکنترلر
۴. آشنایی با محیط اشکال زدایی نرم افزار AVR Studio

۱-۳ نرم افزار CodeVisionAVR

برنامه ای که ما با دستورات می نویسیم صرفاً ارزش یک متن معمولی را دارد و ارزشی برای میکروکنترلر ندارد. برای این منظور هر شرکت سازنده می بایست کامپایلر مخصوص تراشه خود را ارائه دهد و یا اینکه شرکت های طراح کامپایلر باید از قطعه شرکت سازنده حمایت کنند در واقع کامپایلر به معنی تفسیر کننده می باشد و فایل نوشته شده ما را به کدهای قابل فهم برای میکروکنترلر تبدیل می کند و توسط پروگرامر فایل ایجاد شده بر روی تراشه Load می شود.

در بین کامپایلرهای زبان C که برای میکروکنترلرهای AVR از شرکت ATMEL توسط شرکت های سازنده نرم افزار ارائه شده است می توان CodeVisionAVR را یکی از بهترین و قویترین کامپایلرهای زبان C برای میکروکنترلرهای AVR دانست. این نرم افزار طوری طراحی شده است که سرعت اجرای دستورات را بهینه و خیلی خوب از فضای حافظه استفاده می کند. ما در این کتاب به معرفی و کار با این نرم افزار خواهیم پرداخت اما همه ما ملتفت هستیم که آن طور که باید به خوبی یک نرم افزار را در کتاب به طور کامل نمی توان آموزش داد اما در این کتاب روش مدرن و بهتری برای آموزش انتخاب گردیده است. در این فصل نرم افزار کدویژن را به طور کلی قسمت های مهم آن را آموزش داده و نکات تکمیلی و جزئی را به صورت صوتی و تصویری در CD همراه کتاب آموزش می دهیم.

نرم افزار CodeVisionAVR نسخه 1.25.8 سال ۲۰۰۷ میلادی، برای این کتاب انتخاب شده است این نسخه کامل و بدون محدودیت است. البته نسخه سال ۲۰۰۸ میلادی را نیز در CD همراه کتاب قرار داده ایم. شما می توانید آخرین نسخه نرم افزار CodeVisionAVR را از سایت شرکت سازنده آن به نشانی www.hpinfotech.com دانلود کنید اما توجه کنید که به صورت رایگان، نسخه آزمایشی این نرم افزار را در اختیار شما قرار می دهد، که دارای محدودیت در حجم فایل ایجاد شده است.

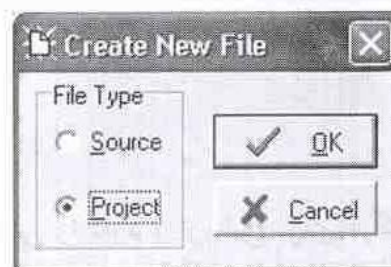
نرم افزار CodeVisionAVR واقع در CD همراه کتاب را نصب کنید پس از اتمام مراحل نصب، محیط نرم افزار را باز کنید و از پیام داده شده گزینه Cancel را انتخاب کنید و فایل Keygen را دابل کلیک نمایید تا پنجره شکل ۱-۳ باز شود. در قسمت Serial Number کلیک راست کرده و گزینه Paste را انتخاب کنید سپس گزینه Make را کلیک کرده و مسیر خواسته شده برای ذخیره فایل License را، پوشه BIN واقع در مسیر نصب نرم افزار تعیین کنید. سپس با اجرای نرم افزار وارد محیط برنامه نویسی شوید. اگر پروژه ای از قبل باز شده بود، از منوی File گزینه Close Project را انتخاب کنید تا آن پروژه بسته شود.



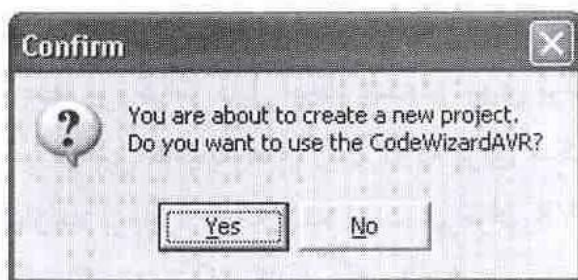
شکل ۱-۳ پنجره نرم افزار KeyGen

ایجاد پروژه جدید

ابتدا از منوی File گزینه New را کلیک کنید تا پنجره شکل ۲-۳ نمایان شود.



شکل ۲-۳ ایجاد فایل جدید



شکل ۳-۳ پنجره Confirm

از پنجره شکل ۲-۳ گزینه Project را انتخاب کرده و OK را تأیید کنید تا پنجره شکل ۳-۳ نمایان شود. در این کادر باز شده سؤال می شود که شما برای ایجاد پروژه می خواهید از کدویزارد استفاده کنید؟ شما گزینه Yes را انتخاب کنید.

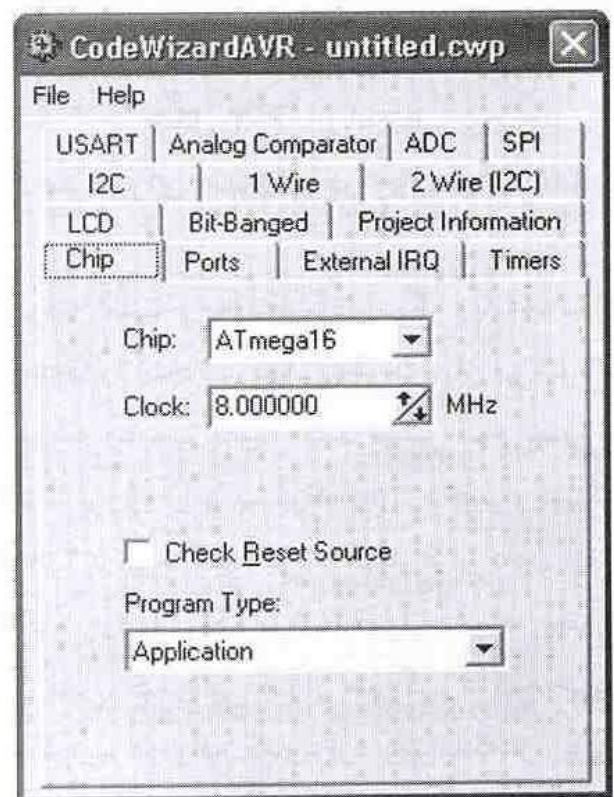
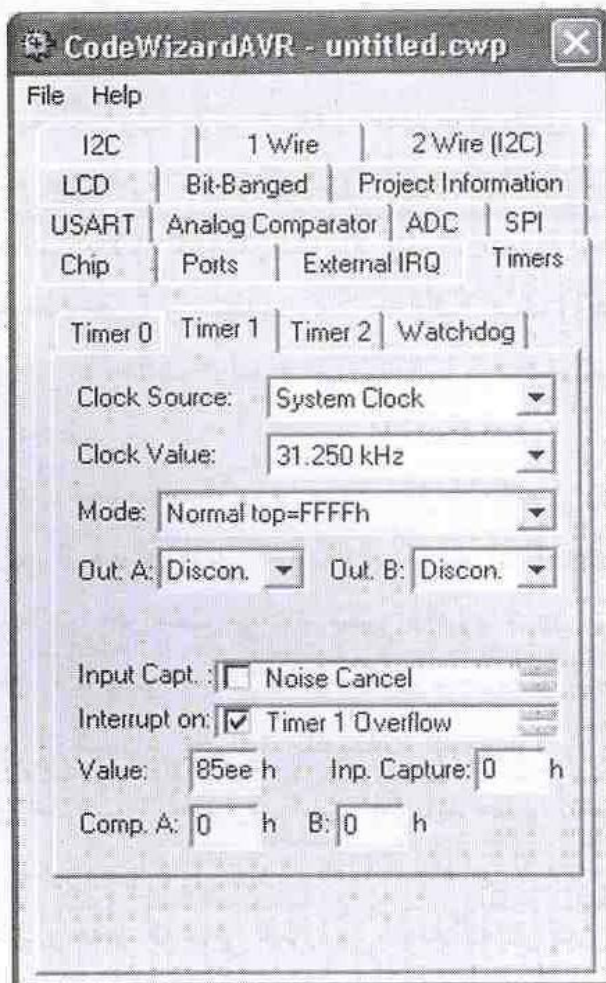
استفاده از کدویزارد (CodeWizardAVR)

پس از انتخاب گزینه YES، پنجره ای مطابق شکل ۴-۳ با زبانه های متعددی باز می شود در واقع کدویزارد به معنی جادوگر کد می باشد و با تنظیماتی که به صورت گزینه ای کاربر انجام می دهد کدویزارد رجیسترهای مربوطه و بدنه ای از توابع را تعیین می کند. از ویژگی های مهم کدویزارد این

است که رجیسترها را برای کاربر با سرعت تنظیم می کند و کدهایی را در اختیار کاربر قرار می دهد و کاربر با کپی کردن آن کدها به برنامه اصلی، به راحتی می تواند از آنها بهره بگیرد.

در زبانه Chip شما باید نوع میکروکنترلر خانواده AVR و کلاک فرکانسی که می خواهید با آن کار کند را تعیین کنید. توجه کنید که فرقی ندارد که شما از کلاک داخلی یا خارجی می خواهید استفاده کنید باید کلاک را حتماً تعیین کنید. زیرا خیلی از توابع به زمان وابسته هستند مانند توابع تأخیر زمانی، توابع تبادل سریال و ... در قسمت Program Type شما باید نوع حافظه را تعیین نمایید. اگر قصد استفاده از Boot Loader حافظه را نداشته باشید آن را بر روی Application قرار دهید.

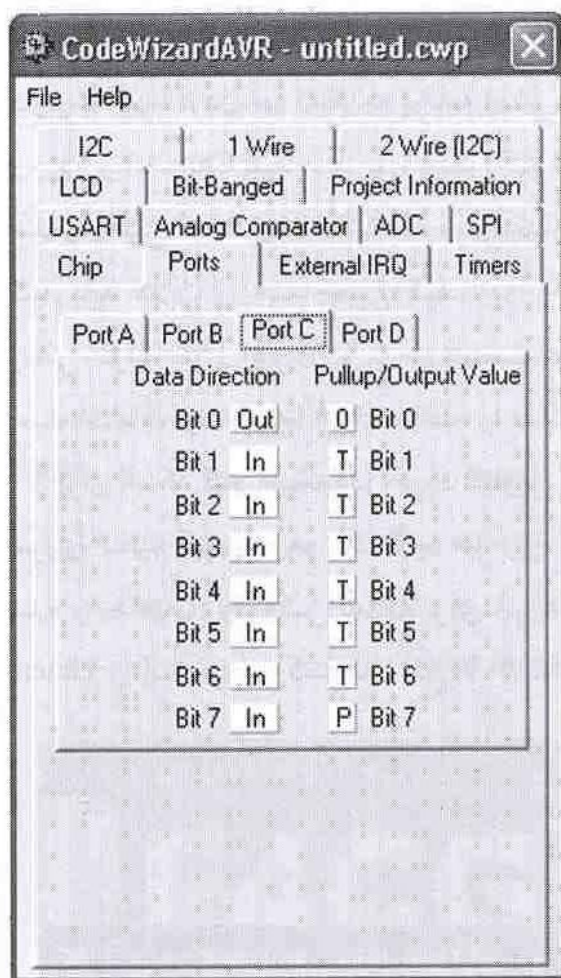
برای اینکه کمی بیشتر در مورد کدویزارد مطلع شوید فرض کنیم که می خواهیم تایمر یک را طوری تنظیم کنیم که هر یک ثانیه سر ریز کند و یک LED که با یک مقاومت 220 اهم به PC.0 متصل شده را خاموش و روشن نماید. برای این منظور بعد از تنظیم تراشه و کلاک در مرحله قبل حال زبانه Timers را انتخاب می کنیم تا شکل ۳-۵ نمایان گردد.



شکل ۳-۵ زبانه تنظیمات تایمرها

شکل ۳-۴ زبانه انتخاب تراشه، توسط کدویزارد

در زبانه Timers تایمر یک را انتخاب می کنیم تا گزینه های تنظیم آن نمایان شود. حال، چون می خواهیم تایمر یک عمل تایمر را انجام دهد در قسمت Clock Source آن را بر روی System Clock قرار می دهیم تا از کلاک سیستم پالس دریافت کند. در قسمت Clock Value تقسیم فرکانسی تایمر را



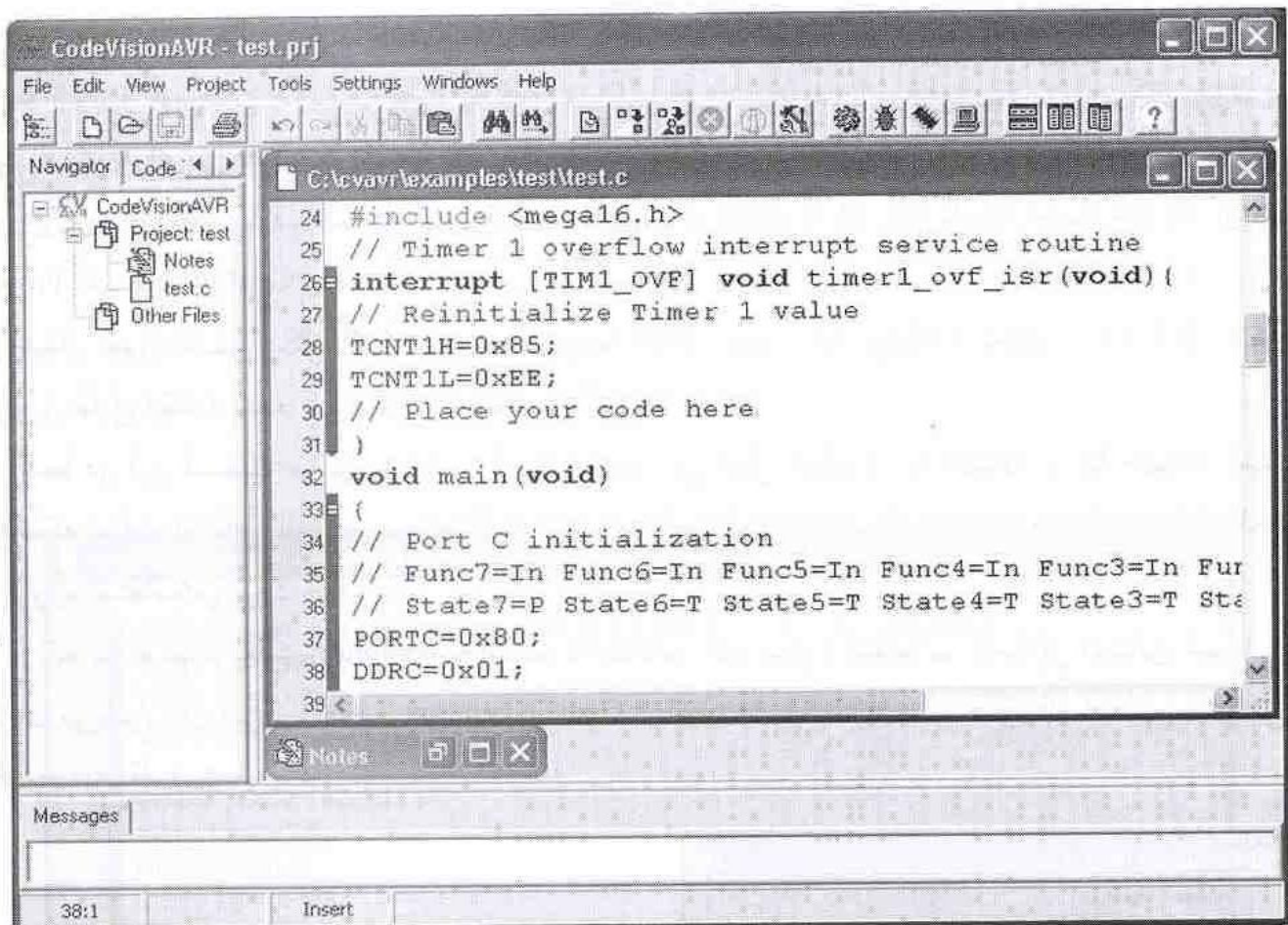
شکل ۳-۶: زبانۀ تنظیم پورت‌ها

طوری قرار می‌دهیم که تایمر با سرعت ۳۲ میکرو ثانیه شمارش کند. در قسمت Mode تایمر را در حالت Normal قرار می‌دهیم. قسمت‌های Out.A و Out.B برای تنظیم خروجی حالت مقایسه‌ای تایمر است که در حالت قطع قرار می‌دهیم و چون از capture تایمر نمی‌خواهیم استفاده کنیم قسمت Input Capt. را رها می‌کنیم و در قسمت Interrupt on گزینه Timer 1 Overflow را تیک می‌زنیم و در قسمت Value مقدار اولیه تایمر را تعیین می‌کنیم. حال تنظیمات تایمر پایان یافته است و زبانه Ports را انتخاب می‌کنیم تا شکل ۳-۶ نمایان شود.

چون می‌خواهیم LED را به PC.0 وصل کنیم، پس Port C را انتخاب می‌کنیم. هر بیت دو بخش تنظیم دارد بخش اول می‌تواند In به مفهوم ورودی و Out به مفهوم خروجی تنظیم گردد و در بخش دوم می‌توان (امپدانس بالا)، P (مقاومت بالا کش)، 0 یا 1 (حالت اولیه خروجی) را تنظیم نمود. ما اولین بیت را در حالت خروجی با حالت اولیه صفر انتخاب کردیم و بیت هفتم را صرفاً برای آموزش در حالت P (Pull-up) قرار داده‌ایم.

حال تنظیمات لازم و مورد نیاز در کدویزارد صورت گرفته است و به تنظیمات دیگر زبانه‌ها نیازی نداریم. از منوی فایل کدویزارد، گزینه Generate, Save and Exit را انتخاب کنید پنجره‌ای باز می‌شود که از شما درخواست می‌کند مسیر ذخیره کردن پروژه را در پوشه مناسب تعیین کنید سپس نام سورس برنامه را با پسوند *.C ذخیره کنید مجدداً از شما می‌خواهد تا پروژه را با پسوند *.prj ذخیره کنید و سپس از شما می‌خواهد تنظیمات کدویزارد را با پسوند *.cwp ذخیره کنید. توجه کنید قرار دادن نام به جای علامت * اختیاری است اما باید مسیر ذخیره این سه فایل را در یک پوشه قرار دهید. بعد از ذخیره این سه فایل پروژه شما ایجاد می‌گردد و در سورس فایل، کدهایی قرار می‌گیرد که خشتی هستند و فقط کدهایی فعال می‌شوند که در کدویزارد تنظیم کردیم در واقع کدویزارد یک بدنه از زبان C با رجیسترهای تنظیم شده در اختیار کاربر قرار می‌دهد بعد از ایجاد پروژه محیط برنامه‌نویسی به صورت شکل ۳-۷ آماده می‌شود.

همان طور که در شکل ۳-۷ می‌بینید محیط این نرم‌افزار بسیار ساده و مناسب می‌باشد و نیازی به توضیحات بیشتر نیست فقط پنجره Messages شاید مجهول باشد. در صورت بروز خطا یا هشدار در موقع کامپایل کردن در این پنجره خطا یا هشدار تعیین می‌شود که با دابل کلیک کردن خط موردنظر مشخص می‌شود.



شکل ۳-۷ محیط برنامه نویسی نرم افزار CodeVisionAVR

- بعد از تکمیل کردن کدهای دیگر توسط کاربر ممکن است با خطاهای احتمالی زیر مواجهه شود:
۱. **no final** : این خطا نشان می دهد که پایان تابع یا دستورات استفاده شده در برنامه نویسی را با علامت **{** تعیین نکردیم در زبان C به ازای هر آکولاد باز شده **{** باید یک آکولاد بسته شده وجود داشته باشد.
 ۲. **missing ""** : این خطا از رشته استفاده شده در یک فرم دستوری است که از علامت دابل کوتیشن استفاده نشده است.
 ۳. **missing ';' :** این خطا نشان می دهد ما در یک خط فراموش کردیم پایان خط از علامت **;** استفاده کنیم در زبان C باید پایان هر خط از علامت نقطه ویرگول استفاده کنیم.
 ۴. **declaration syntax error** : این خطا نشان دهنده خطای گرامری است احتمالاً دستوری را صحیح تایپ نکرده ایم.
 ۵. **invalid expression** : این خطا نشان می دهد یک دستور نوشته شده نامعتبر است به طور مثال شما ممکن است از دستور **while()** استفاده کنید اما داخل پرانتز آن شرطی قرار ندهید.
 ۶. **'data' global variable 'unreferenced'** : این هشدار نشان می دهد که ما از یک متغیر کلی که در برنامه تعریف کردیم اصلاً در برنامه استفاده نکردیم.

سایر خطاها را شما در تجربیاتی که کسب می کنید متوجه می شوید. فقط لازم است بدانید همیشه

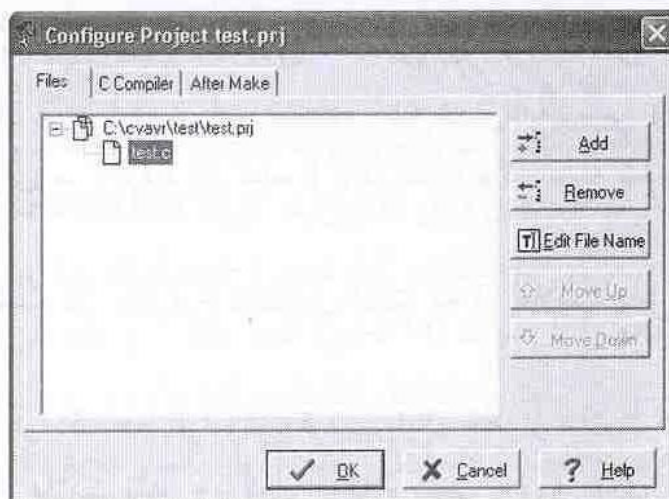
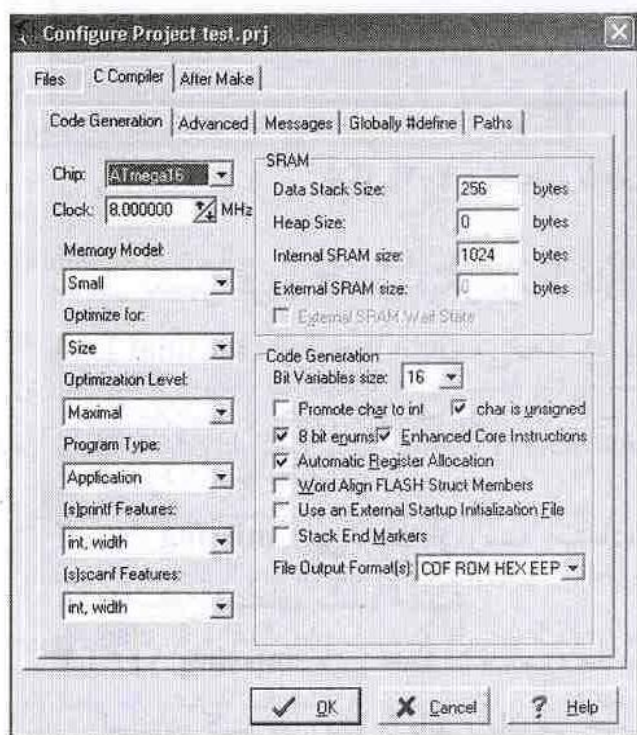
در یک خطای احتمالی رخ داده، نرم افزار یک خطا اعلام نمی کند و ممکن است خطاهای متعددی گرفته شود و همچنین همیشه خطی که خطا در آن رخ داده است را نرم افزار تعیین نمی کند چون خیلی مبهم است. برای بر طرف کردن معایب کافی است یکبار خودتان برنامه را بررسی کنید.

حال می خواهیم با قسمت دیگری که شما در این نرم افزار به آن نیاز دارید، آشنا شوید. از منوی Project گزینه Configure را انتخاب کنید.

همان طور که در شکل ۸-۳ می بینید در زبانه Files مسیر و نام پروژه مشخص شده است. همچنین سورس فایل اضافه شده با نام test.c مشخص گردیده است.

شما در این قسمت می توانید به پروژه خود سورس فایل دیگری را اضافه و یا حذف کنید در موقعیکه برنامه از فایل الحاقی توسط کاربر بهره می گیرد از این قسمت برای اضافه کردن فایل الحاقی به پروژه استفاده می شود.

زبانه پر اهمیت بعدی C compiler است. با انتخاب آن، پنجره تنظیمات کامپایلر مطابق شکل ۹-۳ نمایان می شود. در این زبانه، لبه CodeGeneration از اهمیت بیشتری برخوردار است بنابراین به توضیح این قسمت می پردازیم.

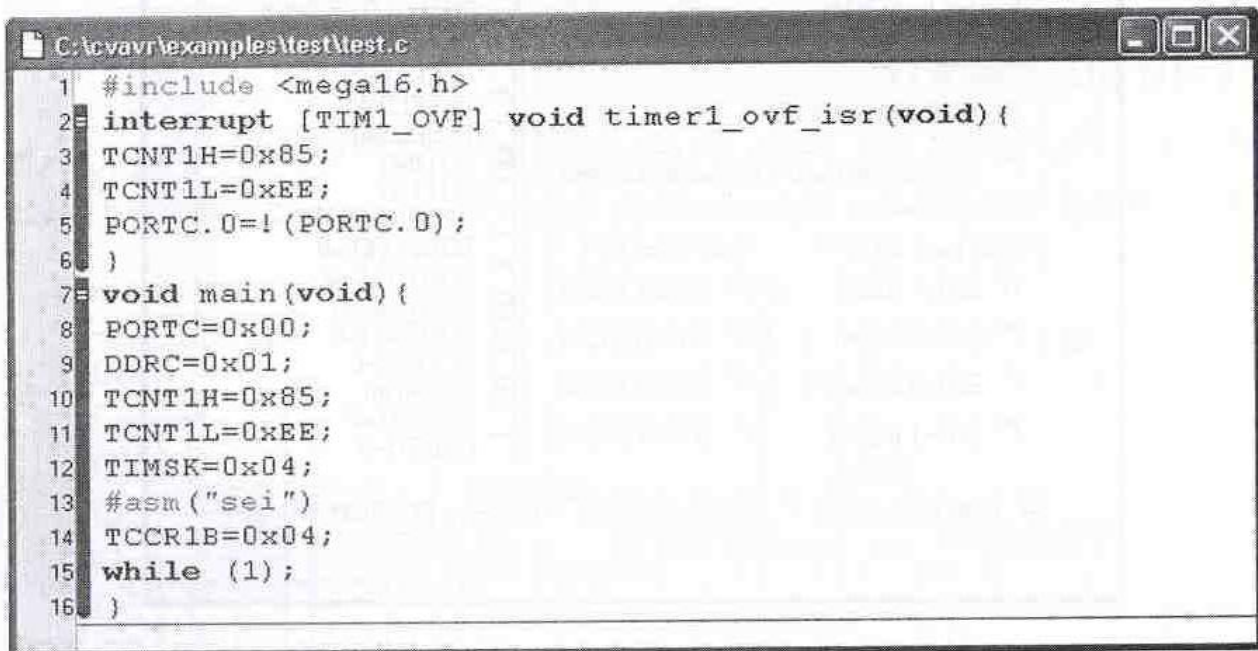


شکل ۸-۳ پنجره پیکربندی پروژه شکل ۹-۳ پنجره Configure Project، قسمت C Compiler

گزینه chip نوع میکروکنترلر استفاده شده و گزینه clock فرکانس کاری را تعیین می کند. گزینه های بعدی به ترتیب مدل حافظه و سطح بهینه سازی و نوع برنامه و همچنین ویژگی های دستورات printf, sprintf, scanf, sscanf را تنظیم می کنند. بخش بعدی مربوط به اندازه حافظه SRAM داخلی است که در آن اندازه کل حافظه SRAM و همچنین میزان فضای استفاده شده توسط اشاره گر پشته تعیین می گردد. در بخش دیگر، تنظیمات مربوط به کامپایلر صورت می گیرد به طور مثال گزینه

Bit Variables size حداکثر متغیر بیتی استفاده شده در برنامه را مشخص می کند. حال که با این قسمت از نرم افزار آشنا شدید می خواهیم کد برنامه را کامل کنیم و همان طور که قبلاً ذکر کردیم یک LED را هر یک ثانیه یکبار توسط وقفه سرریز تایمر خاموش و روشن کنیم برای این منظور از کدهایی که کدویزارد در اختیار ما قرار داده است، استفاده می کنیم و کدهایی که نیاز نداریم و توضیح هستند را پاک می کنیم.

در پنجره شکل ۳-۱۰ کدهای اضافی را پاک می کنیم و در تابع وقفه سرریز تایمر یک، کد $PORTC.0 = ! (PORTC.0)$ را اضافه کردیم تا هر باری که وقفه سرریز تایمر اتفاق می افتد PC.0 معکوس شود و LED چشمک بزند.



```

C:\cvavr\examples\test\test.c
1 #include <mega16.h>
2 interrupt [TIM1_OVF] void timer1_ovf_isr(void){
3   TCNT1H=0x85;
4   TCNT1L=0xEE;
5   PORTC.0=! (PORTC.0);
6 }
7 void main(void){
8   PORTC=0x00;
9   DDRC=0x01;
10  TCNT1H=0x85;
11  TCNT1L=0xEE;
12  TIMSK=0x04;
13  #asm("sei")
14  TCCR1B=0x04;
15  while (1);
16 }
  
```

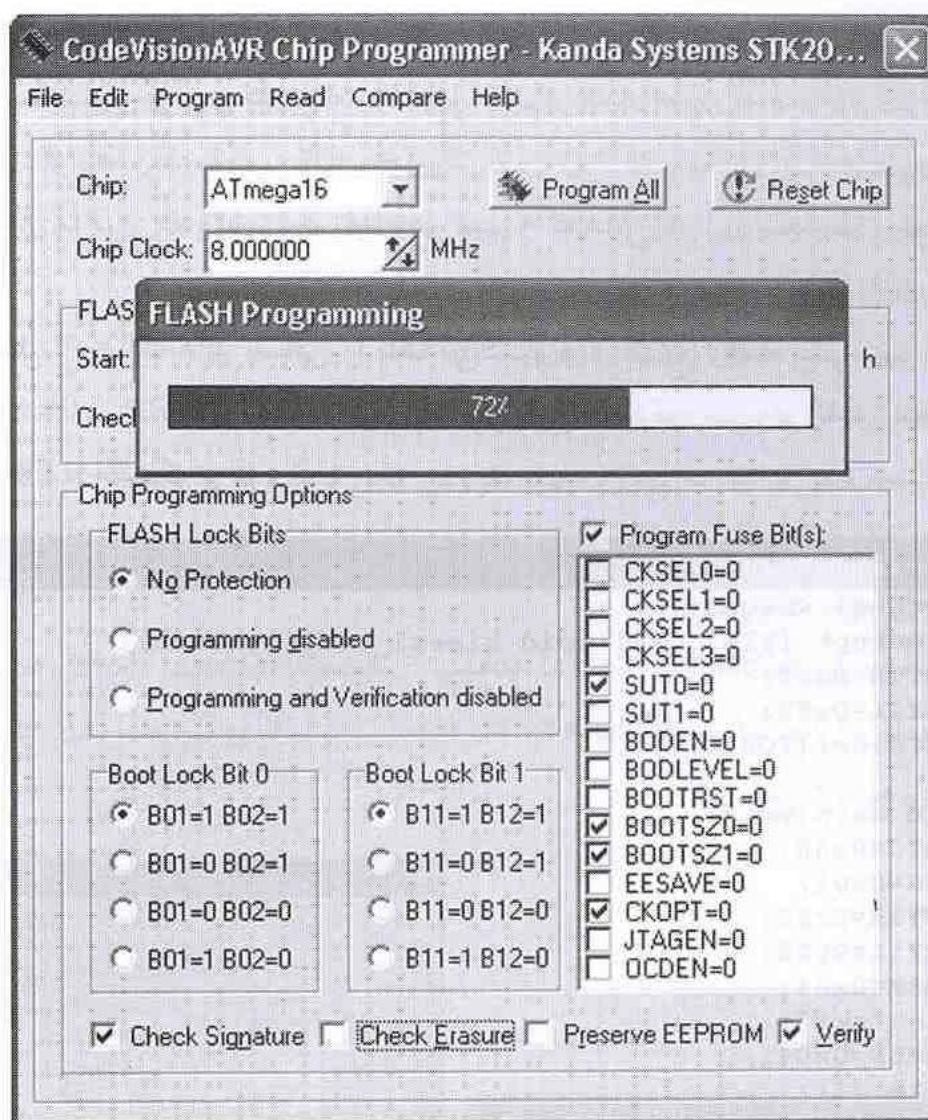
شکل ۳-۱۰ پنجره برنامه تکمیل شده

بعد از کامل کردن کد برنامه باید برنامه را کامپایل کنیم تا کدهای لازم جهت اشکال زدایی و همچنین کد برنامه ریزی (کد هگزاد) تراشه ایجاد شود. برای این منظور از منوی Project گزینه Make را کلیک کنید تا برنامه شما توسط نرم افزار کامپایل شود.

بعد از کامپایل کردن، در صورت نداشتن خطا، اطلاعات مربوط در یک پنجره باز می شود و در اختیار کاربر قرار می گیرد. در این اطلاعات تعداد خط کامپایل شده و همچنین هشدارهای احتمالی مشخص می گردد و در قسمت پایینی این پنجره میزان فضای استفاده شده از حافظه های EEPROM و FLASH میکروکنترلر تعیین می گردد.

برنامه ریزی میکروکنترلرهای AVR :

برنامه را کامل می کنیم و البته در صورت نیاز از اشکال زدایی توسط نرم افزار AVR Studio کمک گرفته و سپس اقدام به برنامه ریزی تراشه می نمائیم. از منوی Tools گزینه Chip Programmer را کلیک کنید. می توانیم یکی از فایل های *.rom یا *.hex یا *.bin که هم نام و هم مسیر با پروژه ایجاد شده اند را از منوی File پنجره شکل ۳-۱۱ و گزینه Load Flash انتخاب کنیم.



شکل ۳-۱۱ پنجره برنامه ریزی میکروکنترلرهای AVR

در پنجره شکل ۳-۱۱ در قسمت Chip نوع میکروکنترلر و کلاک کاری آن تعیین شده است. بخش بعدی Chip Programming Option است که برای قفل کردن حافظه برنامه است با انتخاب گزینه Programming disabled خواندن از حافظه ثابت میکروکنترلر، غیر ممکن می گردد و با انتخاب گزینه Programming and Verification disabled علاوه بر خواندن، بازرسی توسط شبیه ساز سخت افزار (امولاتور) غیر ممکن می گردد در صورتی که تمایل به قفل تراشه دارید باید از منوی Program، گزینه Lock Bits را انتخاب کنید. اما توجه کنید قفل کردن تراشه فقط برای جلوگیری کپی غیر مجاز توسط افراد سودجو است و شما با پاک کردن تراشه قفل آن را نیز پاک می کنید و می توانید مجدداً از میکروکنترلر خود استفاده کنید. توجه کنید اگر از Program All استفاده می کنید باید فیوز بیت های لازم را نیز در قسمت راست این پنجره تنظیم کنید در مورد فیوز بیت های میکروکنترلر ATmega16 در فصل اول به طور کامل توضیح داده شده است. اگر نمی خواهید تنظیمات خاصی انجام دهید می توانید از منوی Program فقط گزینه Flash را انتخاب کنید.

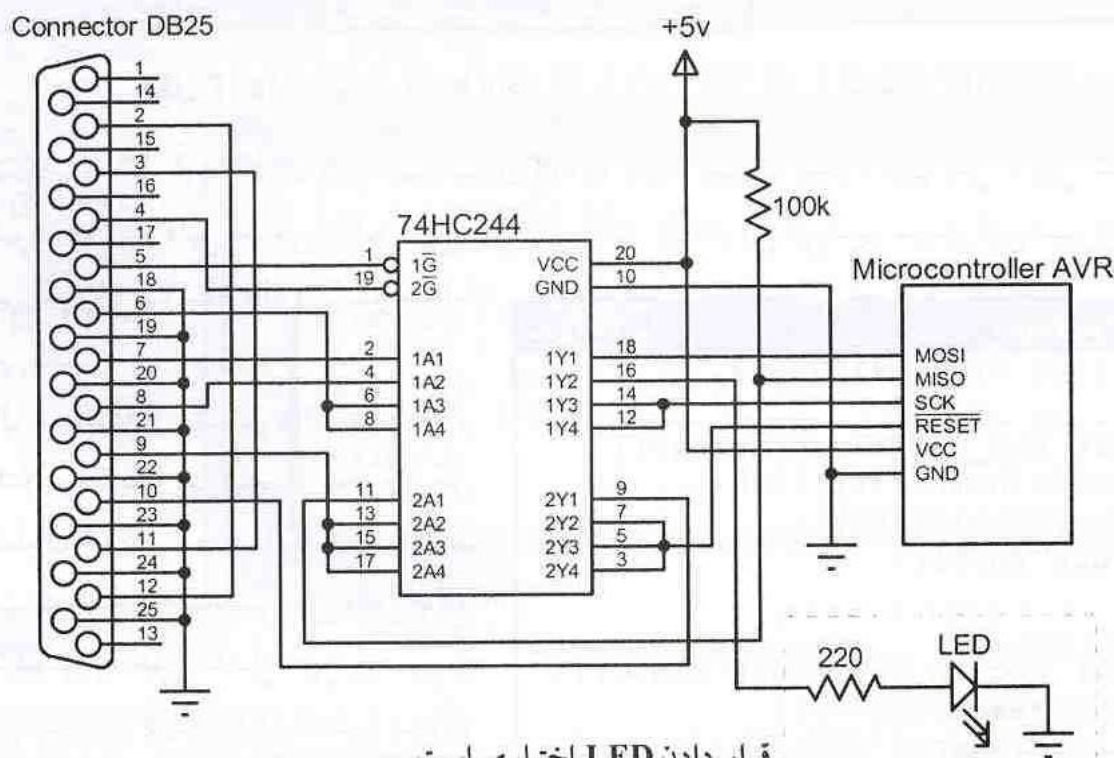
همچنین شما باید نوع پروگرامر خود را قبل از آنکه Chip Programmer را انتخاب کنید، تعیین نمایید. برای این منظور از منوی Settings گزینه Programmer را انتخاب کنید تا پنجره شکل ۳-۱۲ نمایان شود.



شکل ۳-۱۲ پنجره تنظیم پروگرامر

پروگرامر (ISP) STK200+/300

در پنجره باز شده شکل ۳-۱۲ از کادر کشویی قسمت AVR Chip Programmer Type نوع پروگرامر را Kanda Systems STK200+/300 انتخاب کنید سخت افزار این پروگرامر به صورت شکل ۳-۱۳ است.



قرار دادن LED اختیاری است

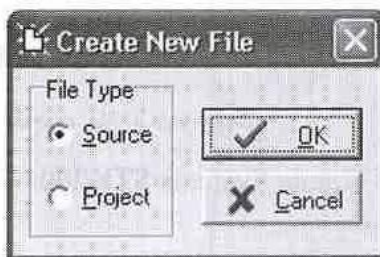
شکل ۳-۱۳ پروگرامر STK200+/300

سخت افزار این نوع پروگرامر که از طریق پورت LPT عمل برنامه ریزی تراشه را انجام می دهد بسیار ساده است شما می توانید با هزینه ای بسیار کم آن را بسازید و یک کابل پروگرامر ایجاد کنید سعی کنید بافر 74HC244 را در یک (کاور) DB25 قرار دهید. اما با استفاده از چسب حرارتی پایه های آن را عایق بندی کنید تا به دیگر پایه های پورت LPT اتصال پیدا نکند. توجه کنید تغذیه ۵ ولت برای بافر از مدار میکروکنترلر شما تأمین می گردد بنابراین تغذیه مناسبی باید در نظر بگیرید. در این کتاب تغذیه میکروکنترلر به صورت ورودی DC و AC در فصل اول آورده شده است. همچنین سعی کنید از

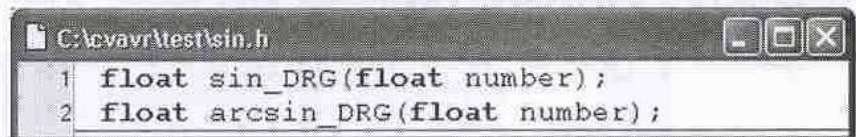
کابل کوتاه و شیلددار استفاده کنید و شیلد کابل را به زمین تغذیه وصل نمایید این عمل در کاهش نویز در موقع پروگرام کردن مؤثر است.

ایجاد کردن فایل الحاقی یا سرآمد

گاهی اوقات برنامه‌ها طولانی و یا تکراری هستند مانند استفاده از اسکن صفحه کلید 4×4 در دو برنامه متفاوت که توابع آنها با هم فرقی ندارد شاید شما نیز تمایل داشته باشید که این برنامه و یا سایر برنامه‌های تکراری را به فایل سرآمد تبدیل کنید و هر زمان که به آنها نیاز داشتید مجبور نباشید مجدداً آن برنامه را بنویسید. ما در این قسمت دو تابع Sin و Arcsin که بر حسب درجه باشند نه رادیان استفاده می‌کنیم و آنها را در یک فایل الحاقی قرار می‌دهیم.



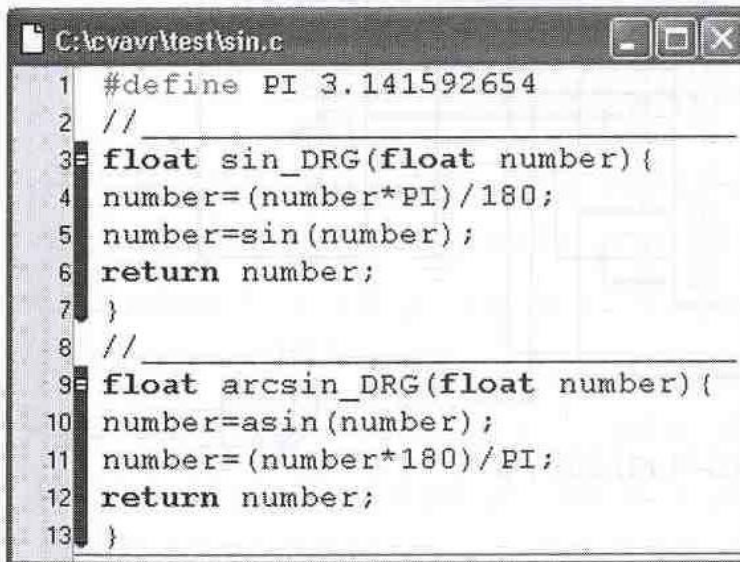
برای شروع ابتدا از منوی File نرم‌افزار گزینه New را کلیک کنید تا شکل ۱۴-۳ نمایان شود.



شکل ۱۴-۳ ایجاد سورس فایل

شکل ۱۵-۳ فایل سرآمد Sin.h

در شکل ۱۴-۳ گزینه Source را انتخاب و OK را تأیید کنید تا پنجره سورس فایل شکل ۱۵-۳ نمایان شود. در این فایل اسامی توابع فایل سرآمد جدید را معرفی می‌کنیم سپس این سورس فایل را در مسیر پروژه با نام sin و پسوند *.h ذخیره می‌کنیم.

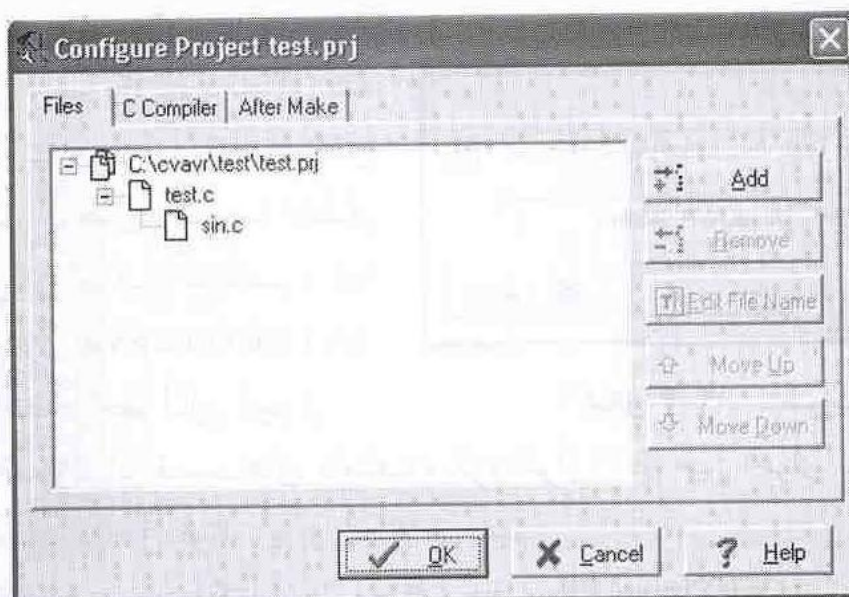


حال مجدداً از منوی File نرم‌افزار گزینه New را کلیک می‌کنیم تا پنجره شکل ۱۴-۳ نمایان شود و مجدداً گزینه Source را انتخاب کرده و در این سورس فایل توابع معرفی شده را به طور کامل می‌نویسیم و سپس این سورس فایل را در مسیر پروژه با نام sin و پسوند *.c ذخیره می‌کنیم. مرحله بعدی اضافه کردن فایل جدید به برنامه است.

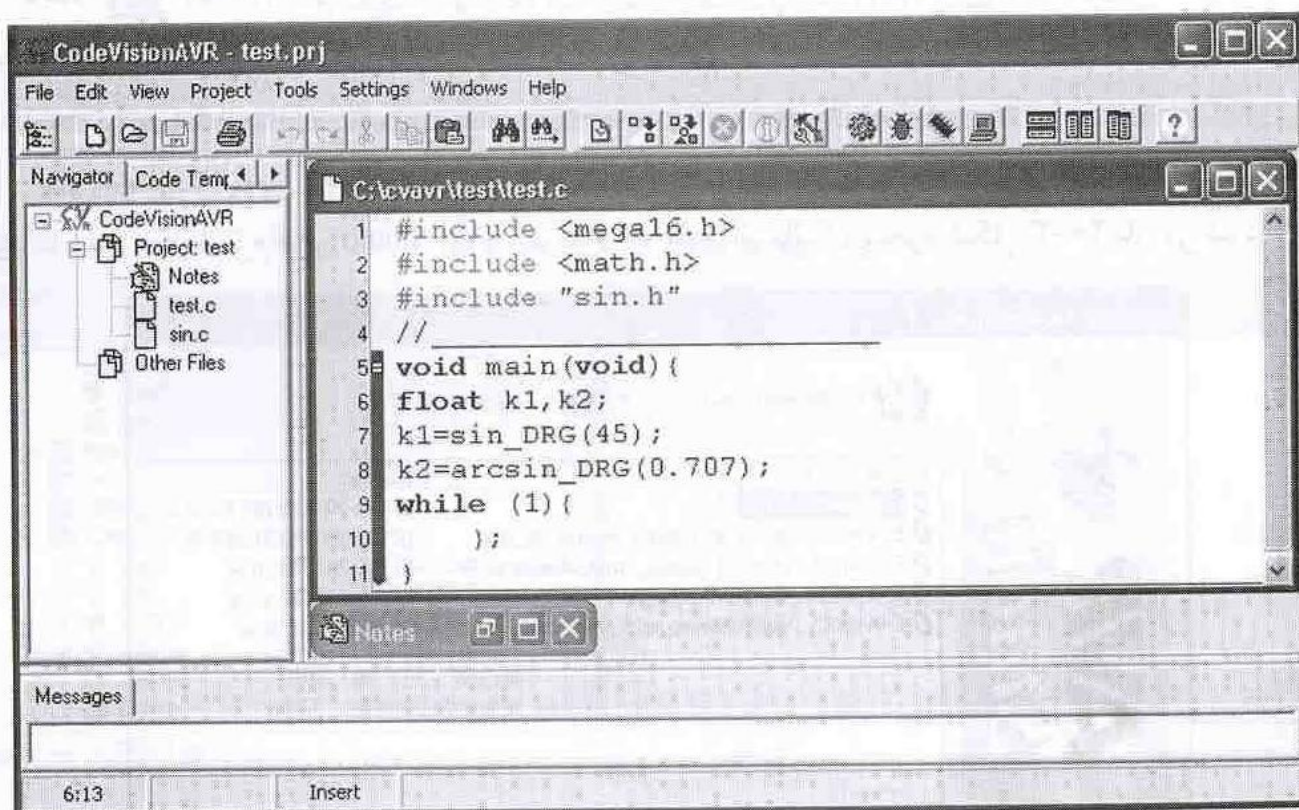
شکل ۱۶-۳ برنامه سورس توابع فایل سرآمد Sin.h

برای این منظور از منوی Project گزینه Configure را انتخاب می‌کنیم تا پنجره شکل ۱۷-۳ باز شود. در این قسمت از گزینه Add استفاده می‌کنیم و فایل Sin.c را به پروژه اضافه کرده و گزینه OK را کلیک می‌کنیم.

حال فایل الحاقی ما ایجاد گردیده است و برای استفاده از آن کافی است با استفاده از دستور `#include "sin.h"` در ابتدای برنامه اصلی معرفی شود.



شکل ۳-۱۷ پنجره Configure Project ، قسمت Files

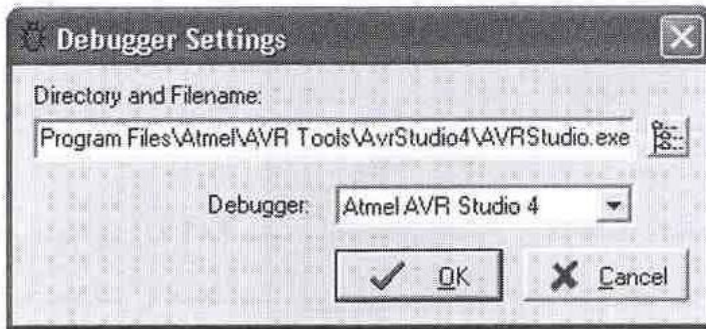


شکل ۳-۱۸ نحوه استفاده از فایل سرآمد ساخته شده در برنامه اصلی

به روشی مشابه می توان فایل کتابخانه ای ایجاد کرد. باید فایل الحاقی را در پوشه INC ، در مسیر نصب نرم افزار و فایل *.C را از منوی File توسط گزینه Convert to Library به فایل کتابخانه ای تبدیل کنید.

کار با نرم افزار اشکال زدایی AVR Studio

در بسیاری از طراحی ها ما نیاز داریم برنامه را اشکال زدایی کنیم. اولین سوالی که در ذهن شما ممکن است رخ دهد این است که ما برنامه را کامپایل کردیم و خطایی هم ندارد پس چه لزومی به

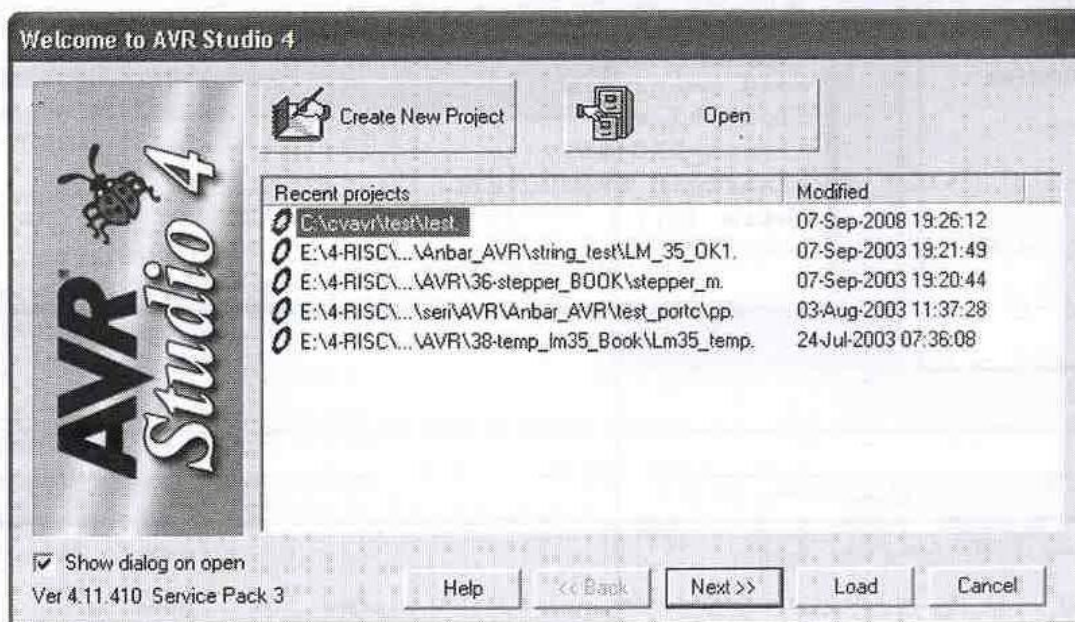


اشکال‌زدایی دارد؟ جواب بسیار واضح است چون ما همیشه نمی‌توانیم بگوییم برنامه‌ای که نوشته‌ایم حتماً درست عمل می‌کند ممکن است عملکرد برنامه، آن طوری که ما انتظار داریم نباشد بنابراین توسط این نرم‌افزار که ارتباط خوبی با نرم‌افزار CodeVisionAVR دارد می‌توانیم برنامه را خط به خط آنالیز کنیم و

شکل ۳-۱۹ تنظیمات اشکال‌زدایی

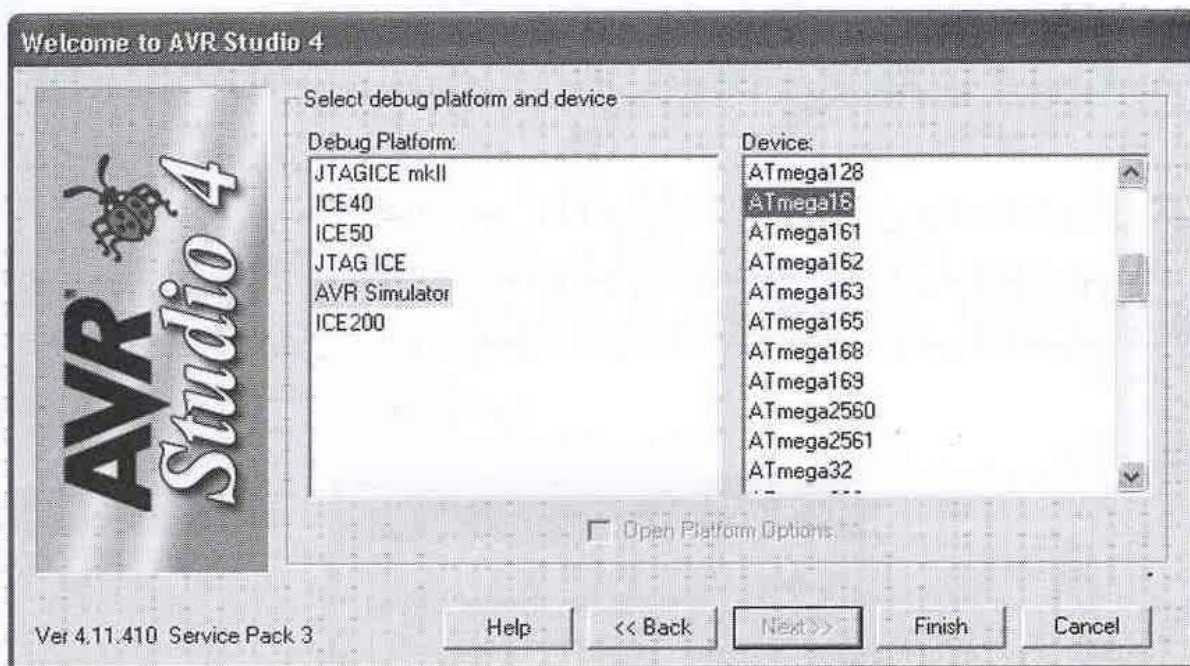
مقداردهی تایمر، وقفه و سایر قسمت‌های مختلف میکروکنترلر را در حین اجرای برنامه مشاهده کنیم. برای کار با این نرم‌افزار مراحل کوتاه زیر را انجام می‌دهیم. نرم‌افزار AVR Studio را نصب کنید و در محیط نرم‌افزار CodeVisionAVR از منوی Settings گزینه Debugger را انتخاب کنید تا پنجره شکل ۳-۱۹ باز شود.

دکمه  (Browse) را کلیک کرده و از مسیر نصب نرم‌افزار AVR Studio در پوشه AvrStudio4 فایل اجرایی AVRStudio.exe را انتخاب کنید و همچنین در قسمت Debugger، نسخه ۴ این نرم‌افزار را انتخاب کنید و دکمه OK را کلیک کرده و پیغام داده شده را نیز تأیید کنید. حال بعد از کامپایل کردن برنامه پروژه مورد نظر خود، از منوی Tools گزینه Debugger را انتخاب کنید تا وارد محیط نرم‌افزار AVR Studio شوید و به طور اتوماتیک پنجره شکل ۳-۲۰ باز می‌شود.

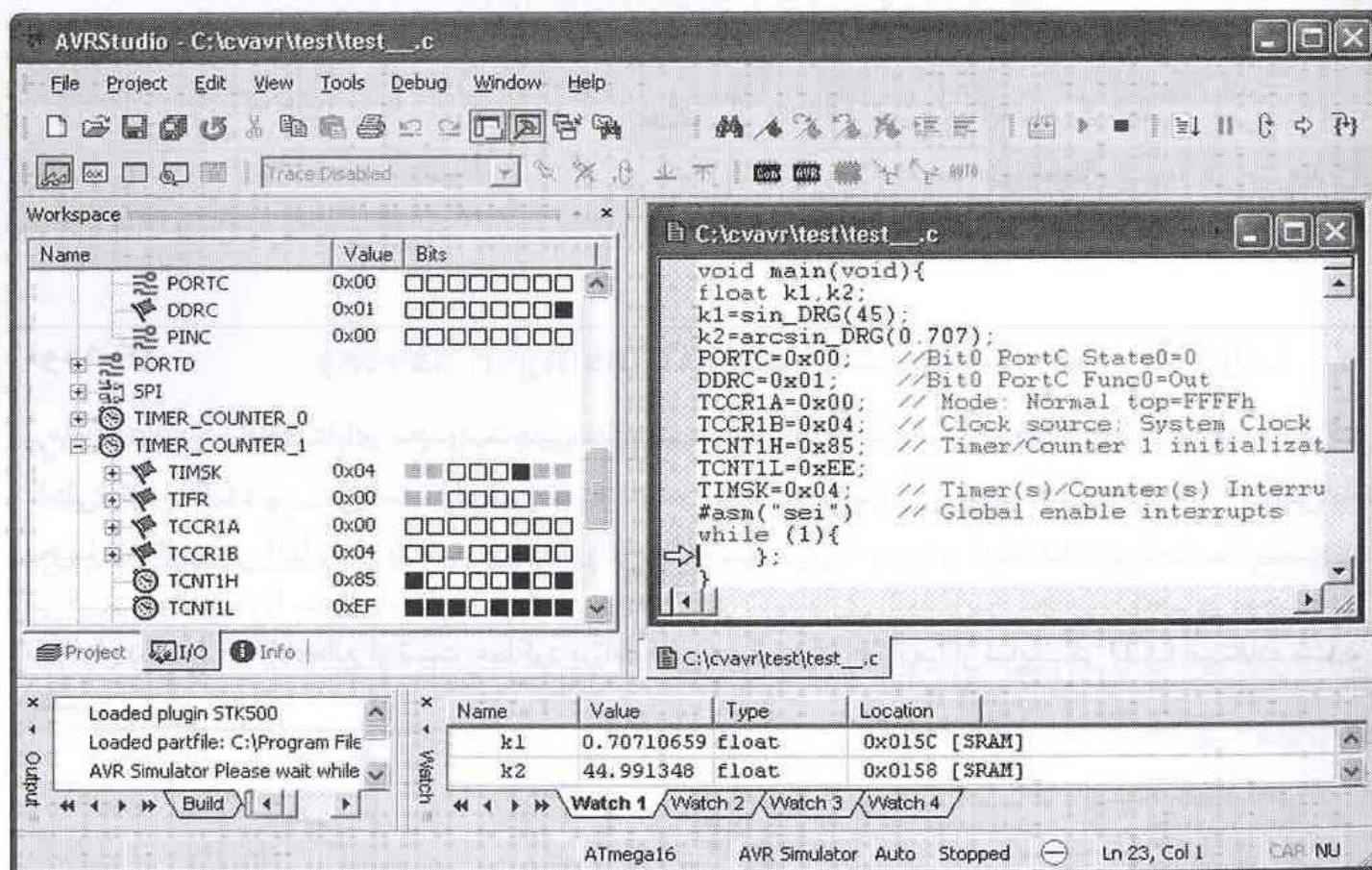


شکل ۳-۲۰ انتخاب فایل اشکال‌زدایی برنامه

همان طور که در شکل ۳-۲۰ مشاهده می‌فرمائید فایل‌هایی که قبلاً مورد آنالیز قرار گرفته‌اند لیست شده‌اند شما باید دکمه Open را کلیک کنید و از مسیر پروژه خود، فایل *.abs یا *.coff را انتخاب کنید اگر نوع فایل *.coff را انتخاب کنید پنجره‌ای مطابق شکل ۳-۲۱ باز می‌شود. در قسمت Debug Platform باید گزینه AVR Simulator را انتخاب کنید و در قسمت Device میکروکنترلر استفاده شده را تعیین کنید و دکمه Finish را کلیک نمائید تا برنامه بارگذاری شود.



شکل ۳-۲۱ انتخاب میکروکنترلر مورد نظر برای شبیه سازی

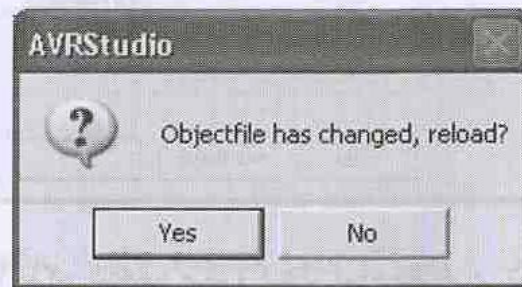


شکل ۳-۲۲ محیط اشکال زدایی نرم افزار AVR Studio

وقتی محیط شکل ۳-۲۲ آماده شد شما می توانید در قسمت Workspace قسمت های مورد نظر مانند رجیسترها و عملکرد CPU و سایر قسمت های سخت افزاری میکروکنترلر را در حین اجرا مشاهده کنید و همچنین یکی از قسمت های پر اهمیت این نرم افزار پنجره watch است که از منوی View

انتخاب می‌گردد. برای اضافه کردن متغیر به پنجره watch کافی است متغیر را انتخاب و بر روی آن کلیک راست کنیم و گزینه Add to watch را انتخاب کنیم در پنجره watch مقدار، آدرس و محل ذخیره‌سازی متغیر مشخص می‌شود.

برای اجرای برنامه می‌توانید از کلید تابعی F11 استفاده کنید. همچنین توجه کنید برای تغییر برنامه شما باید بدون بستن نرم‌افزار AVR Studio به نرم‌افزار CodeVisionAVR برگردید و بعد از تغییرات احتمالی مجدداً برنامه را کامپایل کنید و به محیط AVR Studio برگردید. وقتی به محیط اشکال‌زدایی بر می‌گردید پنجره شکل ۳-۲۳ باز می‌شود.



شکل ۳-۲۳ پنجره بارگذاری مجدد

در این پنجره از شما سؤال می‌شود که آیا در فایل Object تغییری ایجاد کردید و می‌خواهید بارگذاری مجدد انجام دهید؟ شما گزینه Yes را انتخاب کنید تا تغییرات انجام شده در نرم‌افزار CodeVisionAVR بارگذاری گردد.

توجه

نرم‌افزار AVR Studio دارای محدودیت‌هایی نظیر شبیه‌سازی مبدل آنالوگ به دیجیتال، مقایسه‌کننده آنالوگ داخلی، توابع LCD و ... می‌باشد. همچنین توجه داشته باشید که در صورت وجود تأخیر زمانی، قبل از رفتن به محیط اشکال‌زدایی آنها را با علامت // نادیده بگیرید تا شبیه‌سازی را سریعتر انجام دهید و به شما پیشنهاد می‌کنیم پروژه خود را حداقل امکان با نرم‌افزار Proteus نیز شبیه‌سازی کنید. زیرا محدودیت‌های موجود در نرم‌افزار AVR Studio مانع از تست عملکرد برنامه‌هایی می‌شود که در آنها از نمایشگر LCD استفاده شده است.