

# Downloading, Installing and Configuring WinAVR

Author: Colin O'Flynn

Editor: Eric Weddington

ترجمه و تالیف: سامان مشتاق

## فهرست مطالب:

1	.....	1.0	مروری بر WinAVR
1	.....	2.0	دریافت نرم افزار
3	.....	3.0	نصب نرم افزار
4	.....	4.0	MakeFiles
8	.....	5.0	آماده سازی Programmers Notepad
11	.....	6.0	تمرین اولین پروژه
17	.....	7.0	اطلاعاتی بیشتر
18	.....	8.0	واژه نامه

## 1.0 مروری بر WinAVR

WinAVR مانند نرم افزارهای دیگر با نام های مختلف تنها یک ابزار ساده نرم افزاری را در اختیار ما قرار نمیدهد. بلکه Winavr مجموعه کاملتری از ابزارهاست. این ابزارها شامل ( avr-gcc ( the command line compiler و ( avr-libc ( The compiler library that is essential for avr-gcc و ( avr-as ( The assembler و ( avrdude ( the programming interface و ( avr-gdb ( the de-bugger و ( Programmers notepad (editor و ... . این ابزارها همگی برای سیستم عامل Windows پیشبینی شده اند و توسط یک installer قدرتمند نصب میشوند . ( نرم افزاری که فایل های مورد نیاز برنامه در حال نصب را در سیستم عامل شما جایگزاری میکند )

در هنگام رجوع به شماره نسخه , اغلب شماره نسخه کامپایلر avr-gcc را مشاهده می کنید . برای مثال WinAVR حاضر دارای شماره نسخه 3.3 برای کامپایلر-avr-gcc می باشد و این برخلاف استدلال بعضی از استفاده کننده ها است که تصور می کننداز **WinAVR** با شماره نسخه 3.3 استفاده می کنند. درحقیقت تاریخ انتشارنرم افزار است که به عنوان شماره نسخه (Version) نرم افزار مورد استفاده قرار می گیرد. به عنوان مثال WinAVR 20030424 . WinAVR 20030424 یک تاریخ است که در اینجا نمایانگر قدیمی تر بودن این نسخه از WinAVR 20040317 میباشد.

## 2.0 دریافت نرم افزار

سایت **WinAVR** در آدرس زیر قرار دارد:

<http://winavr.sourceforge.net/>

این سایت حاوی اطلاعاتی در مورد **WinAVR** , از قبیل لینک ها و نکته ها و اخبار های مفید می باشد. بهتر است قبل از هر کاری یک سری به این سایت بزنید تا ببینید آیا نکته مهم و قابل توجه ای وجود دارد یا خیر.

حالا شما باید به آدرس زیر بروید:

<http://sourceforge.net/projects/winavr/>

در اواسط صفحه روی لینک ' Download ' برای دریافت آخرین Version ( نسخه ) **WinAVR** کلیک کنید :

Project UNIX name: winavr  
 Registered: 2002-11-27 10:07  
 Activity Percentile (last week): 0%  
 View project activity [statistics](#)  
 View list of [RSS feeds](#) available for this project

**Latest File Releases**

Package	Version	Date	Notes / Monitor	Download
<b>AVR COFF</b>	AVR COFF Beta	May 19, 2003	-	<a href="#">Download</a>
<b>SRecord</b>	1.12	January 28, 2003	-	<a href="#">Download</a>
<b>WinAVR</b>	20030424	April 23, 2003	-	<a href="#">Download</a>

[\[View ALL Project Files\]](#)

**Public Areas**

[Project Home Page](#)

[Tracker](#)

- Bugs ( 0 open / 6 total )  
 Bug Tracking System

- Support Requests ( 0 open / 2 total )

**Latest News**

**WinAVR 20030424 Released**  
 arcanum - 2003-04-28 09:23  
[\[Read More/Comment\]](#)

**WinAVR 20030312 Released**  
 arcanum - 2003-03-12 13:44  
[\[Read More/Comment\]](#)



هنگامی که روی این لینک کلیک می کنید به صفحه پایین وارد می شوید و قسمت هایی را که قبلاً انتخاب شده اند های لایت خواهید دید:

Below is a list of all files of the project. The release you have chosen is **highlighted**. Before downloading, you may want to read Release Notes and ChangeLog (accessible by clicking on release version).





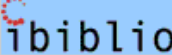









Package	Release & Notes	Filename	Size	D/L	Date Arch.	Type
<b>AVR COFF</b>						
<b>AVR COFF Beta</b>						
		WinAVR-avrcoffbeta-install.exe	1658653		2003-05-19 23:00 14i386	.exe (32-bit Windows)
<b>SRecord</b>						
<b>1.12</b>						
		Srecord-1_12-bin-install.exe	364283		2003-01-28 23:00 1984i386	.exe (32-bit Windows)
		Srecord-1_12-src-install.exe	306290		719i386	.exe (32-bit Windows)
<b>WinAVR</b>						
<b>20030424</b>						
		WinAVR-20030424-bin-install.exe	12417601		2003-04-23 23:00 3479i386	.exe (32-bit Windows)
		WinAVR-20030424-src-install.exe	12302242		958i386	.exe (32-bit Windows)
<b>20030312</b>						
		WinAVR-20030312-bin-install.exe	13239192		2003-03-11 23:00 7866i386	.exe (32-bit Windows)

در اینجا آخرین نسخه دارای تاریخ 20030424 به معنی سال 2003 ماه 04 ( April ) و روز 24 می باشد. شما به WinAVR-XXXXXXX-bin-install.exe که - حروف XXXXXXX به معنی تاریخ مورد نظر برای آن فایل می باشد نیاز خواهید داشت. WinAVR-XXXXXXX-src-install.exe حاوی سورس نرم افزار می باشد که احتمالاً به آن احتیاجی نخواهید داشت.

بعد از کلیک کردن روی فایل مورد نظر از شما خواسته می شود که یکی از **Mirror** ها را انتخاب کنید. **Mirror** سروری است که از طریق آن فایل های مورد نیاز خود را دانلود خواهید کرد. شما می توانید از هر کدام از این **Mirror** ها که می خواهید استفاده کنید. در عین حال پیشنهاد شده تا از **Mirror** ای استفاده کنید که در محدوده شما قرار دارد. چنانچه **Mirror** ای حاوی فایل نبود یا سرعت دانلود آن پایین بود، می توانید مجدداً به این صفحه رجوع کنید و **Mirror** های دیگر را امتحان کنید :

You are requesting file: /winavr/WinAVR-20030424-bin-install.exe  
Please select a mirror

Host	Location	Continent	Download
	Brookfield, WI	North America	 12127 kb
	Reston, VA	North America	 12127 kb
	Chapel Hill, NC	North America	 12127 kb
	Dublin, Ireland	Europe	 12127 kb
	Zurich, Switzerland	Europe	 12127 kb
	Minneapolis, MN	North America	 12127 kb
	Phoenix, AZ	North America	 12127 kb

حال فایل خود را درمحل مشخص ذخیره کنید.

حال که شما **WinAVR** را دریافت کردید ممکن است به بسته های نرم افزاری دیگری نیز , احتیاج داشته باشید. برای مثال برای نسخه 20030324 شما میتوانید فایل **COFF** را که در AVRStudio مورد استفاده قرار می گیرد **Update** ( به روز ) کنید. این کار را به شما پیشنهاد میکنم. برای دانلود کردن این فایل به صفحه فایل ها بروید. در این سند نام این فایل **WinAVR-avrcoffbeta-install.exe** می باشد.

### 3.0 نصب نرم افزار

اول باید روی فایللی که دانلود کردید دابل کلیک کنید. سپس **license** های مختلفی را خواهید دید که باید با همه آنها موافقت کنید. پس از آن محل نصب **WinAVR** را که به صورت پیش فرض مشخص شده قبول کنید.

c:\WinAVR being default.

در نهایت روی دکمه ' install ' کلیک کنید و به **WinAVR** اجازه دهید تا همه چیز را نصب کند! وقتی که نصب انجام شد فایل **README** را مشاهده خواهید کرد. این فایل را در صورت امکان حتماً مطالعه کنید. این فایل حاوی مطالب مهمی در مورد **WinAVR** و راهنمایی استفاده از آن می باشد.

## MakeFiles 4.0

احتمالاً از کامپایلرهای دیگری استفاده کرده باشید و حتماً مشاهده کردید که بیشتر آنها از یک GUI خوب و ساده و یک دکمه ' Compile ' برای رویه کامپایل کردن استفاده می کنند. اما در اینجا کامپایلر avr-gcc از روش دیگری استفاده می کند که در این جهت بسیار قدرتمند نیز ظاهر شده است. به طور واضح تر در این روش از دستورات خط به خط ' command line ' استفاده می شود. به هر حال می توانید آن را برای GUI های مختلفی پایه ریزی کنید و نیز قادرید مطابق با میل خودتان آن را سفارشی کنید.

یک Makefile به کامپایلر می فهماند که چه دستوراتی را انجام دهد. چه فایل‌هایی را کامپایل و یا لینک کند. چه نوع از فایل ها را در خروجی بوجود آورد و همچنین کارهای دیگری همچون پروگرام کردن چیپ. در مجموع یک Makefile می تواند دستوراتی را شامل شود که با اندیشه و کنترل شما کامپایلر را هدایت کند. در این فایل می توانید از متغیرها و نکته های دیگری نیز استفاده کنید. به نام این فایل ( Makefile ) حتماً دقت کنید که در هنگام استفاده از آن مهم می باشد. (در این نام - هیچ پسوندی وجود ندارد ). Makefile با برنامه ای به نام Make عملیات خود را انجام خواهد داد. برای استفاده از آن به مثال زیر توجه کنید.

```
d:\mydir\myproj1> make <CR>
```

( <CR> به معنی فشار دادن کلید ' Enter ' می باشد ). سپس برنامه ( Make ) فایل Makefile را در دایرکتوری تعریف شده پیدا کرده و آن را اجرا می کند. اگر در این دایرکتوری فایل Makefile وجود نداشته باشد با یک خطا روبرو خواهید شد.

یکی دیگر از نکته هایی که باید به آن توجه شود Makefile هایی هستند با Target های مختلف. هر Target وابسته به Makefile شما می تواند کار متفاوتی را - انجام دهد. برای استفاده از این Target ها به آسانی می توانید نام Target را در یک آرگومان پیوست دهید:

```
d:\mydir\myproj1> make <target> <CR>
```

همه Makefile هایی که در AVR استفاده می کنید حداقل دارای دو Target هستند:

```
all
clean
```

' make all ' با ' make ' از لحاظ تاییبی برابرند و در حالت پیش فرض کامپایلر تمام فایل‌های Source شما را کامپایل و فایل hex. را خواهد ساخت. ' make clean ' تمامی فایل‌های موقتی و همچنین فایل‌های hex. و .cof. را پاک می کند. در حقیقت شما را برای یک کامپایل کامل دیگر آماده می سازد.

در هنگام انجام یک پروژه احتمالاً شما بیش از چند ' Source Code ' داشته باشید. اگر همه آنها را کامپایل کنید و بعد یکی از آنها را تغییر دهید در هنگام کامپایل - مجدد توسط برنامه ' make ' حداقل تعداد فایل‌ها کامپایل خواهند شد ( تنها فایل‌ی که تغییر کرده است ) که این در حفظ زمان می تواند بسیار موثر واقع شود مخصوصاً هنگامی که فایل‌های ' Source ' زیادی داشته باشید.

اما در بعضی شرایط نیز مجبور می شوید که یک کامپایل کلی را انجام دهید. برای مثال هنگامی که شما میزان Optimization ( بهینه سازی ) را تغییر می دهید. در این حالت شما فایل‌ی را تغییر نداده اید و تنها با اجرای ' make ' کامپایلر چیزی را تغییر نخواهد داد. شما می بایست ' make clean ' را انجام دهید و بعد از آن ' make '.

خوب است که WinAVR دارای Target زیر نیز هست:

coff

این Target فایل .cof را می سازد که می توانید آن را در AVRStudio برای 'Debug' کردن استفاده کنید. فرمت این Target در حال حاضر قدیمی شده است و در AVRStudio 4.06 و نسخه های قدیمی تر مانند ( AVRStudio 3.x ) استفاده می شود. از این رو این Target با نوع دیگری با نام ( extcoff ) تمدید و به Target های دیگر در Makefile اضافه شده است:

extcoff

این فرمت باید برای AVRStudio 4.07 یا بالاتر استفاده شود. (این فایل نیز دارای همان پسوند .cof می باشد)

Target دیگری که احتمالاً به آن احتیاج خواهید داشت :

program

این Target فایل .hex را که توسط Make ساخته می شود دریافت کرده سپس آنرا در حافظه Flash از چیپ AVR مورد نظر بارگذاری می کند. به این منظور برای آشنایی بیشتر شما پیشنهاد می شود که یک سری به Makefile های نمونه بزنید. شما می توانید یک نمونه را در دایرکتوری WinAVR Sample خود پیدا کنید. (c:\WinAVR\Sample) فایل مورد نظر تنها به اسم Makefile می باشد و بدون هیچ پسوندی. اگر شما COFF به روز شده را نصب کرده اید ممکن است نیاز به فراخوانی Makefile.cof داشته باشید. در آن حالت Makefile را به Makefile.cof و Makefile.old را به Makefile تغییر دهید.

از اینرو چون Makefile ها هیچ پسوندی ندارند شما نمی توانید توسط برنامه دیگری از آنها استفاده کنید. هرچند می توانید توسط یک TEX Editor آنها را ویرایش کنید: بیاید از 'Programmers Notepad' استفاده کنیم. یک 'Shortcut' از این برنامه بر روی Desktop شما موجود می باشد , اما اگر نیست می توانید – آنرا از دایرکتوری pn در محل نصب WinAVR خود پیدا کنید. (c:\WinAVR\pn\pn.exe)

قبل از اینکه ویرایش خود را آغاز کنیم بهتر است اول را جمع به یک موضوع مهم دیگر بحثی داشته باشیم. این موضوع راجع به Tab ها ( فرورفتگی ) در Makefile هاست. تقریباً در هر برنامه دیگری سه یا چهار فاصله ( Space ) معادل یک کاراکتر ( Tab ) شناخته می شوند ( یعنی می توانید به جای چند بار ضربه زدن به کلید ( Space ) از روی کیبورد کلید دیگری را به نام ( Tab ) فقط یک بار بزنید). اما نه در Makefile. کاراکتر Tab نشان دهنده آغاز دستور است و به این صورت شما نباید از آنها در Makefile خود استفاده کنید مگر اینکه قرار است برای مثال برنامه ای را اجرا کنید. یک راه برای مشاهده این تفاوت این است که به منو 'view' از برنامه PN بروید و روی 'mark Whitespace' کلیک کنید تا Check box آن Tick بخورد. در اینجا مثالی از یک ویرایش اشتباه را مشاهده می کنید:

```
234 begin:
235     →@$(BEGIN)
236     ...echo."This will fail horribly"
237
```

من قصد داشته ام که یک دستور 'echo' را در این قسمت اضافه کنم. می توانید با 'mark Whitespace' مشاهده کنید که 'tab' ( فرورفتگی ) توسط یک پیکان بلند و 'Space' ها ( فاصله ها ) نیز توسط نقطه هایی به وضوح مشخص شده اند. در این Makefile از 'Space' هایی در پشت دستور 'echo' استفاده شده که موجب به وجود آمدن یک خطا می شود. همچنین در تصویر پایین عدم وجود خطا را مشاهده می کنید:

```
234 begin:
235     →@$(BEGIN)
236     →echo."This will not fail horribly"
237
```

اگر شما خطاهایی را مبنی بر وجود یک ویرایش غلط مشاهده می کنید, احتمالاً قبلاً از یک 'Tab' درجایی نادرست استفاده کرده اید یا در جایی که نیاز به یک 'Tab' داشته اید از آن استفاده نکرده اید و نیز ممکن است جایی فراموش کرده اید که از فاصله ها (Space) استفاده کنید.

حال می توانیم به سراغ Makefile خودمان برویم و آن را تغییر دهیم. Makefile را در PN باز کنید. در اولین خط باید نوع AVR خود را مشخص کنیم :

# MCU name

MCU = atmega128

برای تغییر این قسمت تنها atmega 128 را پاک کنید و نام پروسسور مورد نظر خود را بنویسید. می توانید لیست کاملی از AVRهایی که پشتیبانی می شوند را به همراه نام هایشان در avr-libc مشاهده کنید. (انتهای این سند را برای لینک مربوطه مشاهده کنید) برای چند نمونه:  
at90s8515 atmega8 attiny22 at90s2313 at86rf401

خط بعدی که به آن احتیاج داریم جایی است برای مشخص کردن نام پروژه. برای مثال می خواهید نام پروژه را myproj1 مشخص کنید که در اینجا باید نام فایل سورس (Source) سی- (فایلی که روال main() در آن قرار دارد) را myproj1.c انتخاب کنید. فایل های hex و COFF هر کدام به ترتیب به نامهای myproj1.hex و myproj1.cof در خروجی ساخته خواهند شد. این خط به شکل زیر نوشته می شود:

# Target file name (without extension).

TARGET = main ← نام فایل اصلی را بدون پسوند وارد کنید

همچنین شما می توانید فایل های سورس C دیگری که پروژه شما از آنها استفاده می کند را مشخص کنید. برای مثال فایل دیگری به نام foo.c دارید که با فایل اصلی (main.c) مرتبط است. ( interface routines ) برای اضافه کردن این فایل چند راه وجود دارد. شما می توانید تنها با اضافه کردن آن به 'SRC=' مانند خط زیر:

SRC = \$(TARGET).c foo.c

یا می توانید با استفاده از تابع += و کمی تغییر :

SRC = \$(TARGET).

SRC += foo.c

و در آخر می توانید از کاراکتر '\ ' استفاده کنید که foo.c در خط بعد قرار می گیرد.

SRC = \$(TARGET).c \

foo.c

اگر فایل های سورس شما اسمبلی هستند می توانید آنها را به 'ASRC=' اضافه کنید. فایل های اسمبلی باید به S. ختم شوند. (S بزرگ) و نه s. (s کوچک). بهتر – است قبل از اولین کامپایل از فایل های خود یک بکاپ (نسخه کپی از اصلی) تهیه کنید. به این دلیل که اگر تصادفاً پسوند فایلی را تغییر دادید و یا با خطا مواجه شدید در حالت 'make clean' فایل های شما پاک خواهند شد.

ASRC = myasmfile.S

همچنین می توانید میزان بهینه سازی (Optimization) را نیز به سادگی تغییر دهید که درجه های مجاز بهینه سازی عبارتند از 0 (بهینه سازی صورت نگیرد) , 1 , 2 , 3 و s. هر کدام از درجه ها با هم متفاوت هستند و درجه 3 همیشه بهترین نمی باشد. هر کدام از این درجه ها وظایف مختلفی را بر روی خروجی برنامه نوشته شده (code) در جهت سرعت بخشیدن یا کم کردن حجم ایفا می کنند. (غیر از درجه 0 که هیچ کاری انجام نمی دهد). اگر از درجه 3 استفاده کنید به احتمال زیاد کد حاصله بسیار بزرگ خواهد شد که البته سریع ترین هم خواهد بود. درجه s یک بهینه ساز برای سایز کد می باشد.

OPT = s

اگر Makefile شما تارگت 'Program' را پشتیبانی کند، شما می توانید اخطیارات دیگری را نیز برای برنامه ریزی AVR در دست بگیرید.

به هر حال این خیلی عمومیت ندارد. برای اطلاعات جزئی تر به تفسیرهای (comments) makefile رجوع کنید. برای مثال Koen Van den Heuvel پیکره بندی یک پروگرامر پارالل را نشان می دهد:

```
Run C:\winAVR\bin\install_giveio.bat to enable the parallel port use in
AvrDUDE. AvrDUDE is the downloader used by WinAVR.

Change the following settings in the make file to make it work with the programming
cable:

AVRDUDE_PROGRAMMER = bascom
AVRDUDE_PORT = lpt1
```

آخرین مطلب اضافه کردن دایرکتوری ها در هنگام کامپایل می باشد. این واقعاً ساده است و شاید لازم باشد کمتر از چند خط به Makefile خود اضافه کنیم. در مرحله اول ببینید آیا خط زیر در Makefile شما وجود دارد یا خیر:

EXTRINCDIRS =

اگر وجود ندارد به موقعیت پرچم های کامپایلر بروید - جایی شبیه خطهای زیر:

```
CFLAGS = -g -O$(OPT) \
-funsigned-char -funsigned-bitfields -fpack-struct -fshort-enums \
-Wall -Wstrict-prototypes \
-Wa,-ahls=$(<:.c=.lst)
```

سپس \$(patsubst %,-1 %,\$(EXTRINCDIRS)) را مانند نمونه زیر به آن اضافه کنید:

```
CFLAGS = -g -O$(OPT) \
-funsigned-char -funsigned-bitfields -fpack-struct -fshort-enums \
-Wall -Wstrict-prototypes \
-Wa,-ahls=$(<:.c=.lst) $(patsubst %,-1 %,$(EXTRINCDIRS))
```

این خط به کامپایلر می گوید که EXTRERAINCDIRS را اضافه کند و قبل از هر جمله یک -I قرار دهد. و این نحوه اضافه کردن یک (include directory) ضمیمه کننده دایرکتوری است. حالا نیاز به محلی برای تعریف متغیر است. پس به بالای Makefile بروید و خط زیر را در آنجا قرار دهید:

EXTRINCDIRS = c:\fool\inc c:\myproj2\inc c:\salvo\inc

این مثال دایرکتوری های c:\fool\inc , c:\myproj2\inc , c:\salvo\inc را در محل ضمیمه (include) اضافه می کند. البته شما احتمالاً به این دایرکتوری ها احتیاج ندارید و می خواهید آنها را از EXTRINCDIRS بردارید پس تنها خط زیر کافی است:

EXTRINCDIRS =

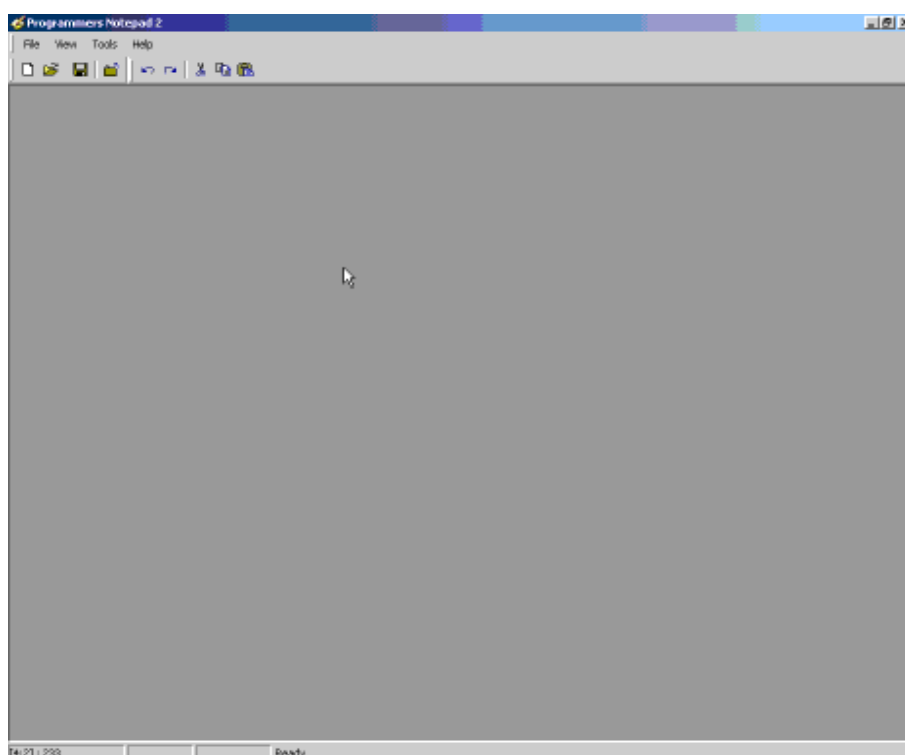
چنانچه این دایرکتوری ها را برای فایل های اسمبلی می خواهید \$(patsubst %,-1 %,\$(EXTRINCDIRS)) را به انتهای پرچم های اسمبلی ('AFLAGS='line) اضافه کنید.

**Makefile** یک ابزار قدرتمند است. راهنمای استفاده از **Make** در دایرکتوری `WinAVR\doc\gnu` ([c:\WinAVR\doc\gnu\make.html](http://c:\WinAVR\doc\gnu\make.html)) در دسترس است. پیشنهاد می‌کنم در صورت امکان حتماً آن را مطالعه کنید.

**Jörg Wunsch** آخرین نکته در مورد پیکره بندی **Makefile** که اگر به نظر خیلی سردرگم کننده می‌رسد: یک برنامه به اسم **Mfile** ساخته است. این برنامه در <http://www.SAX.de/~joerg/mfile/> برای دانلود کردن در دسترس است. این برنامه **Makefile** را برای شما می‌سازد. (**Makefile Generator**) هنگامیکه این برنامه را دانلود کردید آن را در دایرکتوری **WinAVR** آن زیپ 'Unzip' کنید. دو فایل با اهمیت برای شما عبارتند از: 1- **README** که یک فایل **.text** است و می‌توانید با دابل کلیک کردن روی این فایل آنرا در محیط ویندوز توسط برنامه **Notepad** بخوانید. 2- سپس می‌توانید فایل **help** را برای بدست آوردن اطلاعات کافی از این برنامه مطالعه کنید. شما باید کاملاً طریقه استفاده از این برنامه را درک کنید.

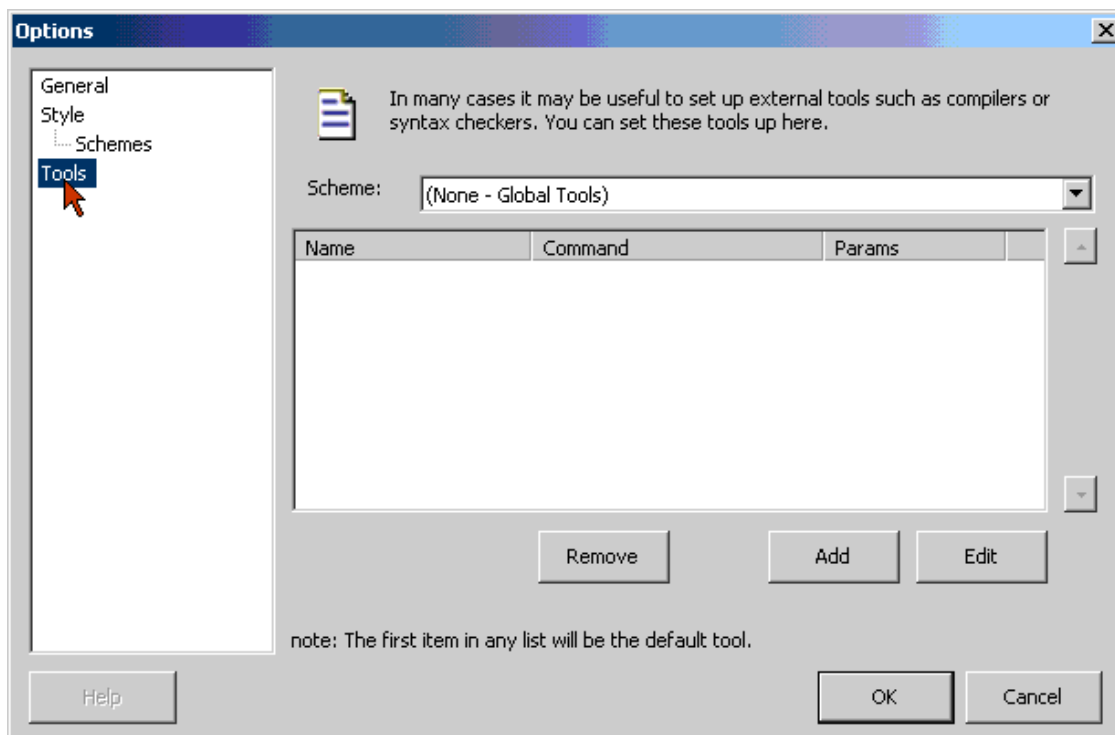
## 5.0 آماده سازی Programmers Notepad

**Programmers Note ( PN )** ابزاری است برای ویرایش برنامه‌هایی که شما می‌نویسید. این برنامه به سادگی پیکره بندی می‌شود و نیازی به دستورات خط به – خط ندارید. (**command line**) که این خود کار کردن با این برنامه را آسان می‌کند. اول از همه ( **PN** ) را توسط شرت کات 'Shortcut' روی دسک تاپ 'Desktop' آن باز کنید. اگر این شرت کات را ندارید از دایرکتوری **WinAVR** ([c:\WinAVR\pn\pn.exe](http://c:\WinAVR\pn\pn.exe)) **pn.exe** را باز کنید.

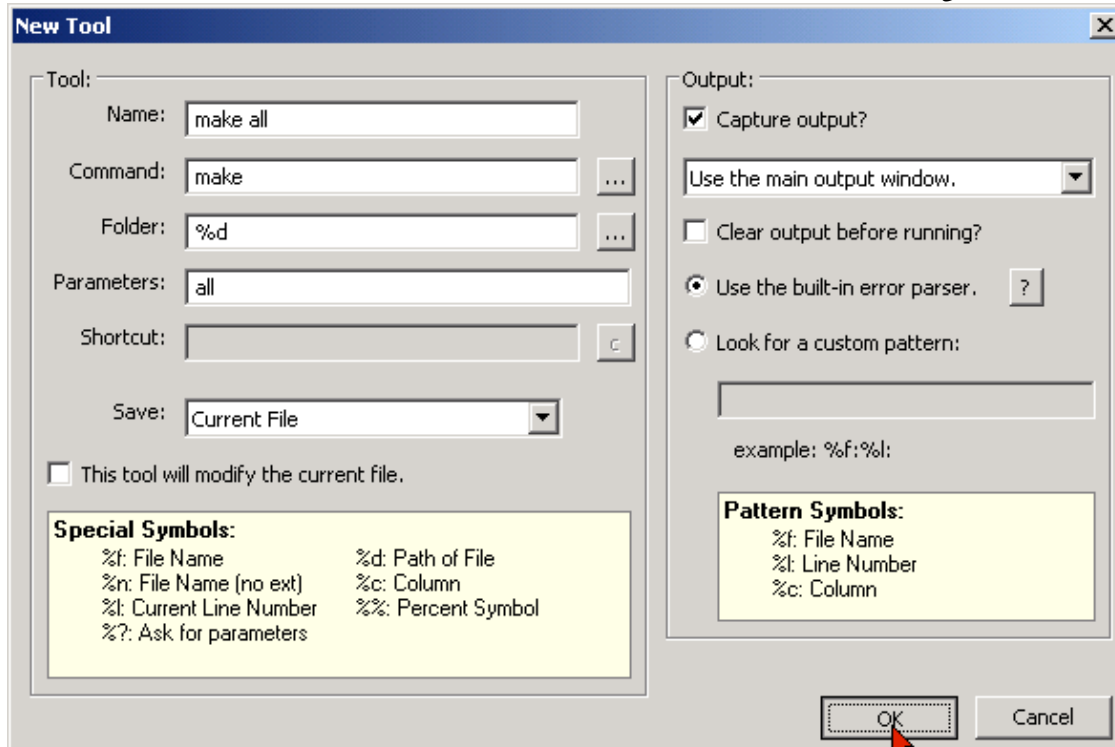


در حال حاضر ممکن است پنجره اصلی را به صورت شکل بالا و خالی مشاهده کنید. از اینرو بپایید **PN** را برای استفاده در **WinAVR** آماده کنیم. از لیست منوها به **Tools → Options** بروید و سپس از پنجره ظاهر شده گزینه **Tools** را انتخاب کنید.





حالا برای اضافه کردن یک ابزار جدید باید روی 'Add' کلیک کنید و بعد از آن اطلاعات مورد نیاز را وارد می کنیم. (مثالها در جدول صفحه بعد هستند) برای مثال 'Make All' من مانند تصویر زیر می باشد:



بباید با این پنجره بیشتر آشنا شویم. Name: برای وارد کردن نام ابزار مورد نظر است که در حقیقت این نام در منوی Tools ظاهر خواهد شد.

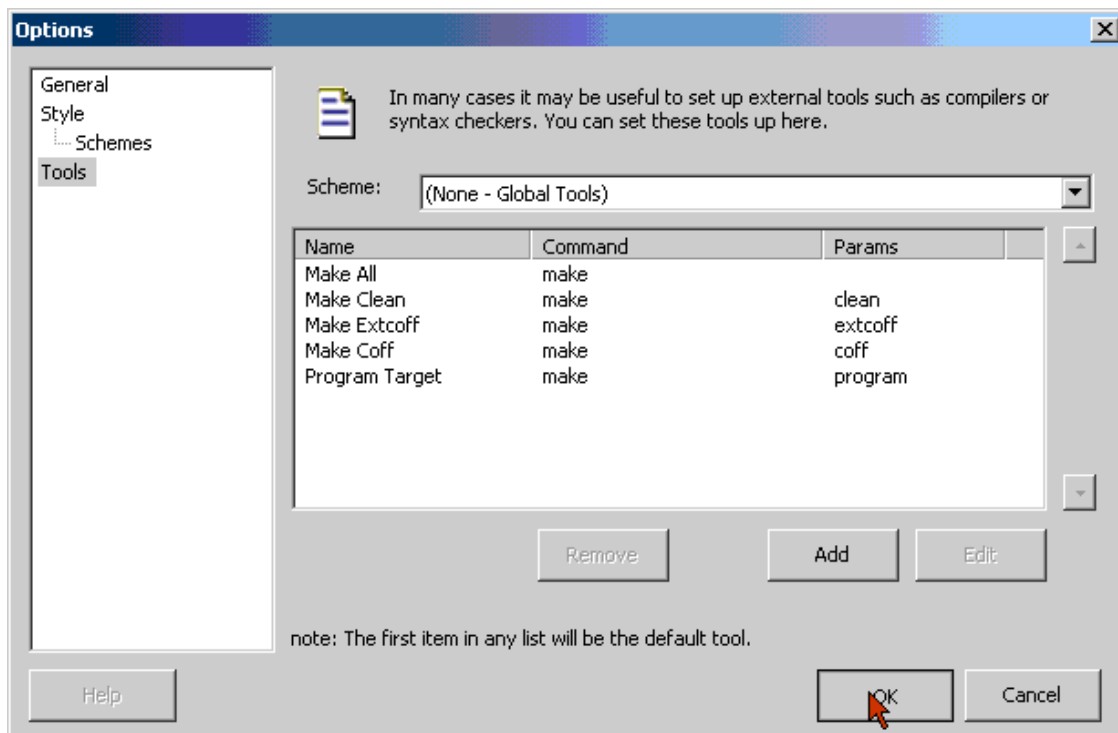
'Command' دستوری است از نوع 'command-line' برای اجرا، که این دستور به همراه پارامترهایی (Parameters) در 'Folder' تایین شده فعالیت خود را انجام می دهد. 'Capture output?' را تیک بزنید و از منوی پایین کشیدنی آن 'Use the main output window.' را انتخاب کنید. این عمل باعث می شود تا در پنجره کوچکی به نام 'Output' که در پایین صفحه مونیتور ظاهر می شود نتیجه 'make all' را مشاهده کنید.

من 'Clear output before running?' را تیک نمی زنم، چراکه با این کار قبل از هر بار اجرا کردن نتیجه کامپایل قبلی که در پنجره 'Output' ظاهر شده است پاک خواهد شد. گزینه 'Save' را نیز می توان به صورت دستی برنامه ریزی کرد که در اینجا فایل جاری قبل از کامپایل شدن ذخیره می شود. به خاطر داشته باشید که اگر 'Make all' را قبل از ذخیره کردن فایل خود انجام دهید، به این معنی می باشد که تغییرات شما کامپایل نشوند. پس با تنظیم این گزینه یک قدم جلوتر هستید.

شما می توانید ابزار خود را مانند من از روی جدول زیر تنظیم کنید:

Name	Command	Folder	Parameters	Capture Output?	This tool will modify..	Save:	Clear Output?	Use Built-in Parser.
Make All	make	%d	all	Main	No	Current File	No	Yes
Make Clean	make	%d	clean	Main	No	None	No	Yes
Make Extcoff	make	%d	extcoff	Main	No	None	No	Yes
Make Coff	make	%d	coff	Main	No	None	No	Yes
Program Device	make	%d	program	Main	No	None	No	Yes

حال 'OK' را بزنید، ابزارهای شما پیکره بندی شده اند.

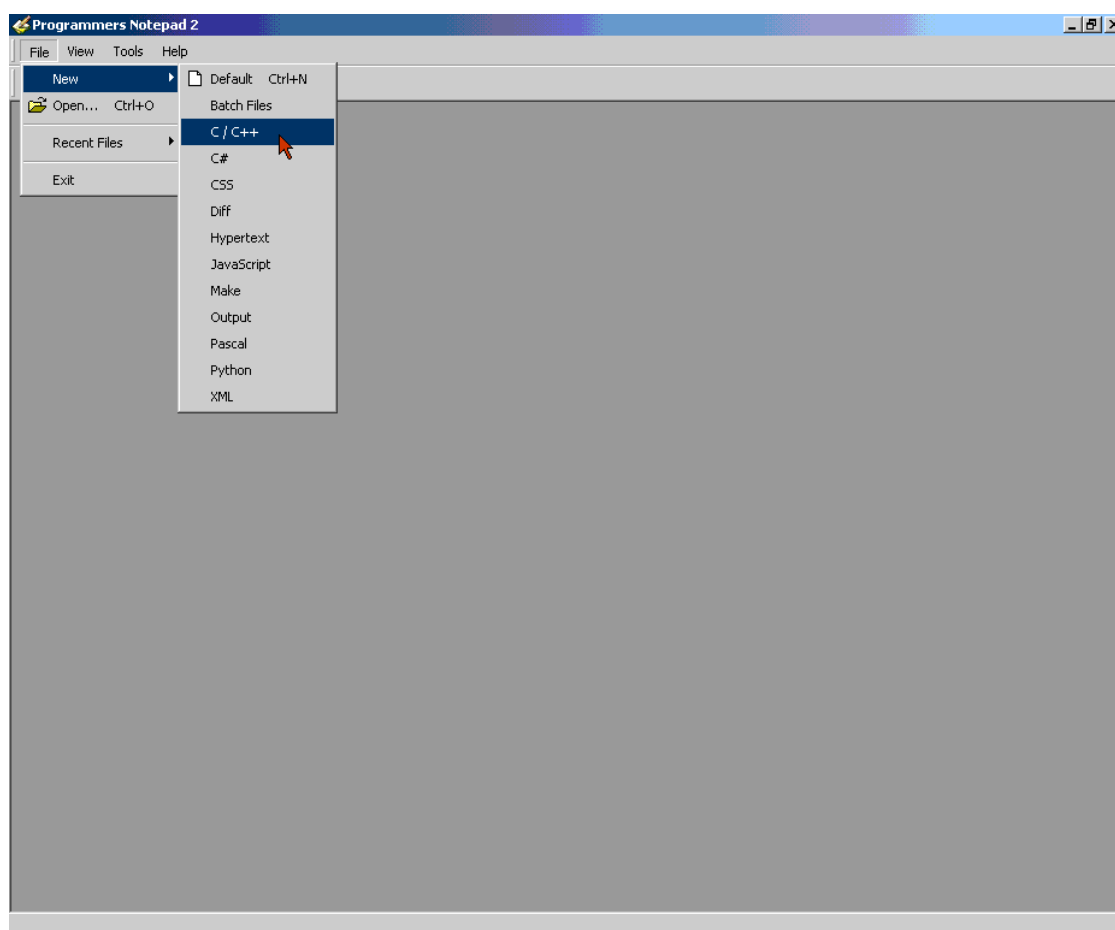


اگر به منو 'Tools' بروید متوجه تغییر چیزی نمی شوید تا زمانی که یک فایل را باز کنید که در این زمان ابزارها ظاهر خواهند شد. در هر صورت پیکره بندی - **Programmers Notepad** را انجام داده اید.

## 6.0 تمرین اولین پروژه

این پروژه ساده به شما نشان می دهد که چگونه در **WinAVR** به صورت واقعی برنامه نویسی کنید. الزاماً شما باید با زبان C و همچنین AVR های Atmel آشنایی داشته باشید.

**Programmers Notepad** را باز کرده و اولین فایل C را آغاز کنید:



در فایل C خود برنامه زیر را بنویسید و در مورد خطاها نگران نباشید زیرا آنها از قبل برای این پروژه پیشبینی شده اند:

```
#include <avr/io.h>
#include <avr/delay.h>

void main (void)
{
    unsigned char counter;

    //set PORTB for output
    DRB = 0xFF;
```

```

while (1)
{

//set PORTB.2 high
PORTB |= 1<<2;

//wait (10 * 120000) cycles = wait 1200000 cycles
counter = 0;
while (counter != 5)
{
//wait (30000 x 4) cycles = wait 120000 cycles
_delay_loop_2(30000);
counter++
}

//set PORTB.2 low
PORTB &= ~(1<<2);

//wait (10 * 120000) cycles = wait 1200000 cycles
counter = 0;
while (counter != 5)
{
//wait (30000 x 4) cycles = wait 120000 cycles
_delay_loop_2(30000);
counter++
}
}

return 1;
}

```

بباید به نکته های مهم این برنامه اشاره ای کنیم. در خط اول ( `#include <avr/io.h>` ) شامل فایل های سر تیر (header files) برای AVR است. این خط تمام رجیستر های مختلف (مانند: `PORTB` , `DDRB` و غیره) AVR ای را که در `Makefile` خود مشخص کرده اید برای کامپایلر تعریف می کند.

خط بعد شامل `delay.h` می باشد، که از تابع های آن در حلقه `While` برای چشمک زدن یک LED استفاده کرده ایم.

خط بعدی قرار است که `PORTB` را به یک درگاه (Port) خروجی تبدیل کند. اگر برگه اطلاعات AVR را بخوانید متوجه می شوید که برای تبدیل `PORTB` به یک درگاه خروجی باید عدد `0xFF` را در رجیستر `DDRB` بنویسید که در این مثال برای آشنایی شما با طرز گزارش خطا در کامپایلر `AVR-GCC` ما عدد `0xFF` را به نادرست در `DRB` نوشته ایم.

`While(1)` در اینجا ایجاد یک حلقه نامنتهی می کند. `While(1)` معادل `While(2==2)` است که همیشه در حال نشان دادن حاصل صحیح می باشد.

برای ست 'SET' کردن پایه ای از `PORTB` روش `PORTB |= 1<<2` را استفاده کرده ایم. به این معنی می باشد که مقدار `PORTB` را گرفته، آن را با عدد 1 که دو مرتبه شیفت به چپ کرده ایم OR منطقی می کنیم.

برای ریست کردن همان پایه از PORTB روش  $(1 < 2) \sim$  PORTB &= استفاده کرده ایم. به این معنی می باشد که مقدار PORTB را گرفته، آن را با معکوس عدد 1 که دو بار به سمت چپ شیفت داده شده است AND منطقی می کنیم. پس اگر 1 را دوبار به سمت چپ شیفت کنیم، نتیجه می شود 00000100 و مکمل آن نیز 11111011 است. تفاو تی نمی کند که شما با چه مقداری از PORTB عمل AND را انجام می دهید. نتیجه xxxxx0xx است ( که x میتواند 1 یا 0 باشد).

در این برنامه چند خطا از قبل فرض شده اند. فایل خود را با نام test1.c در یک فولدر (Folder) در جایی مشخص ذخیره کنید. Makefile را از دایرکتوری C:\WinAVR\sample\Makefile در Programmers Notepad باز کنید (دقت کنید که Makefile هیچگونه پسوندی ندارد) و در فولدر جدیدی که ساخته اید ذخیره کنید. حالا Makefile جدید را باید آماده کنید که این به عهده خود شما گذاشته می شود. بعد از آماده کردن Makefile اقدام به کامپایل کردن برنامه خود می کنیم، برای این منظور به Make All → Tools بروید.

اگر خطایی به صورت زیر مشاهده کردید:

make: \*\*\* No rule to make target `main.o', needed by `main.elf'.

به این علت است که فایل سورس خود را به خوبی معرفی نکرده اید. پس به Makefile خود رجوع کنید تا اطمینان پیدا کنید 'TARGET=Line' شبیه نمونه پایین است:

TARGET = test1

و خط مربوط به 'SRC=' مانند زیر:

SRC = \$(TARGET).c

سپس مجدداً برنامه را کامپایل کنید. شما باید خطاهایی مانند زیر دریافت کنید:

```
test1.c:9: warning: return type of `main' is not `int'
test1.c: In function `main':
test1.c:13: error: `DRB' undeclared (first use in this function)
test1.c:13: error: (Each undeclared identifier is reported only once
test1.c:13: error: for each function it appears in.)
test1.c:28: error: parse error before `}' token
test1.c:40: error: parse error before `}' token
test1.c:45:10: warning: no newline at end of file
make: *** [test1.o] Error 1
```

با اولی که یک هشدار 'Warning' می باشد آغاز می کنیم. این به ما می گوید که نوع برگشتی main از نوع 'int' که باید با استاندارد ANSI C مطابقت داشته باشد نیست. پس باید آنرا در فایل سورس خود تغییر دهید. برای صرفه جویی در زمان کافیسیت که روی خطاها کلیک کنید تا مکان نما به محل قرا گیری خطا پرش کند. مانند تصویر زیر:

The screenshot shows the AVRStudio IDE with a file named `test1.c` open. The code is as follows:

```

2  #include <avr/delay.h>
3
4  void main (void)
5  {
6      unsigned char counter;
7
8      //set PORTB for output
9      DRB = 0xFF;
10
11      while (1)
12      {
13
14          //set PORTB.2 high
15          PORTB |= 1<<2;
16
17          //wait (10 * 120000) cycles = wait 1200000 cycles
18          counter = 0;
19          while (counter != 5)
20          {
21              //wait (30000 x 4) cycles = wait 120000 cycles
22              _delay_loop_2(30000);
23              counter++;
24          }
25
26          //set PORTB.2 low
27          PORTB &= ~(1<<2);
28
29          //wait (10 * 120000) cycles = wait 1200000 cycles
30          counter = 0;
31      }
32  }

```

Below the code editor, the output window shows the compilation errors:

```

.stab      708      0
.stabstr   1151     0
Total     2145

avr-gcc -c -mmcu=at90s8515 -I. -g -O0 -funsigned-char -funsigned-bitfields -fpack-struct -fshort-enums -Wall -Wstri
test1.c:5: warning: return type of 'main' is not 'int'
test1.c: In function 'main':
test1.c:9: error: 'DRB' undeclared (first use in this function)
test1.c:9: error: (Each undeclared identifier is reported only once
test1.c:9: error: for each function it appears in.)
test1.c:24: error: parse error before '}' token

```

بعد از آن خطایی که مبنی بر وجود تعریف نشده `DRB` است را می بینید. به خط مربوطه بروید و مشاهده می کنید که باید `DDRB` را بجای `DRB` تایپ کنید. آن را تغییر دهید.

حال به خطای تجزیه برنامه ( `parse error` ) می رویم که از یک آکولاد ' } ' خطا گرفته که مشاهده می کنید ' } ' ها به درستی قرار گرفته اند. اگر به یک خط بالاتر نگاه کنید متوجه می شوید که در انتهای `Counter ++` سمی کالن ' ; ' فراموش شده است پس آنرا به `Counter ++` تغییر دهید، مشابه آن در حدود 10 خط پایین تر `Counter ++` را نیز به `Counter ++` تغییر دهید. توجه داشته باشید که اغلب خطاها به علت فراموش کردن چیزی در برنامه است (مخصوصاً ' } ' یا ' ; ' ها که موجب بوجود آمدن یک لیست کامل از خطاها می شوند)

آخرین هشدار این است که در آخرین خط از فایل C خود یک خط جدید ( `New-Line` ) باید اضافه کنید. (این خط احتمالی است و ممکن است که از این جهت از شما ایرادی گرفته نشود). باید به پایان فایل C ( آخرین خط ) بروید و کلید ' Enter ' را چند بار بزنید.

حالا مجدداً برنامه را کامپایل کنید و در این زمان برنامه باید بدون خطا کامپایل شود. اگر پروگرام کردن ( `Programmer` ) را در `Makefile` خود پیکره بندی کرده باشید الان زمانی است که می توانید از ' `Make Program` ' استفاده کنید. یا ممکن است فقط بخواهید برنامه خود را شبیه سازی ' `Simulate` ' کنید که در این صورت می توانید از `AVRStudio` استفاده کنید.

وجود تابع های تاخیر زمانی در این برنامه، استفاده کردن از آن را در `AVRStudio` برای شبیه سازی طولانی، بلکه خسته کننده می کند. از اینرو آنرا به برنامه دیگری که از تاخیر ' `Delay` ' در آن استفاده نمی شود تبدیل می کنیم:

```
#include <avr/io.h>
```

```
void main (void)
{

    //set PORTB for output
    DRB = 0xFF;

    while (1)
    {

        //set PORTB.2 high
        PORTB |= 1<<2;

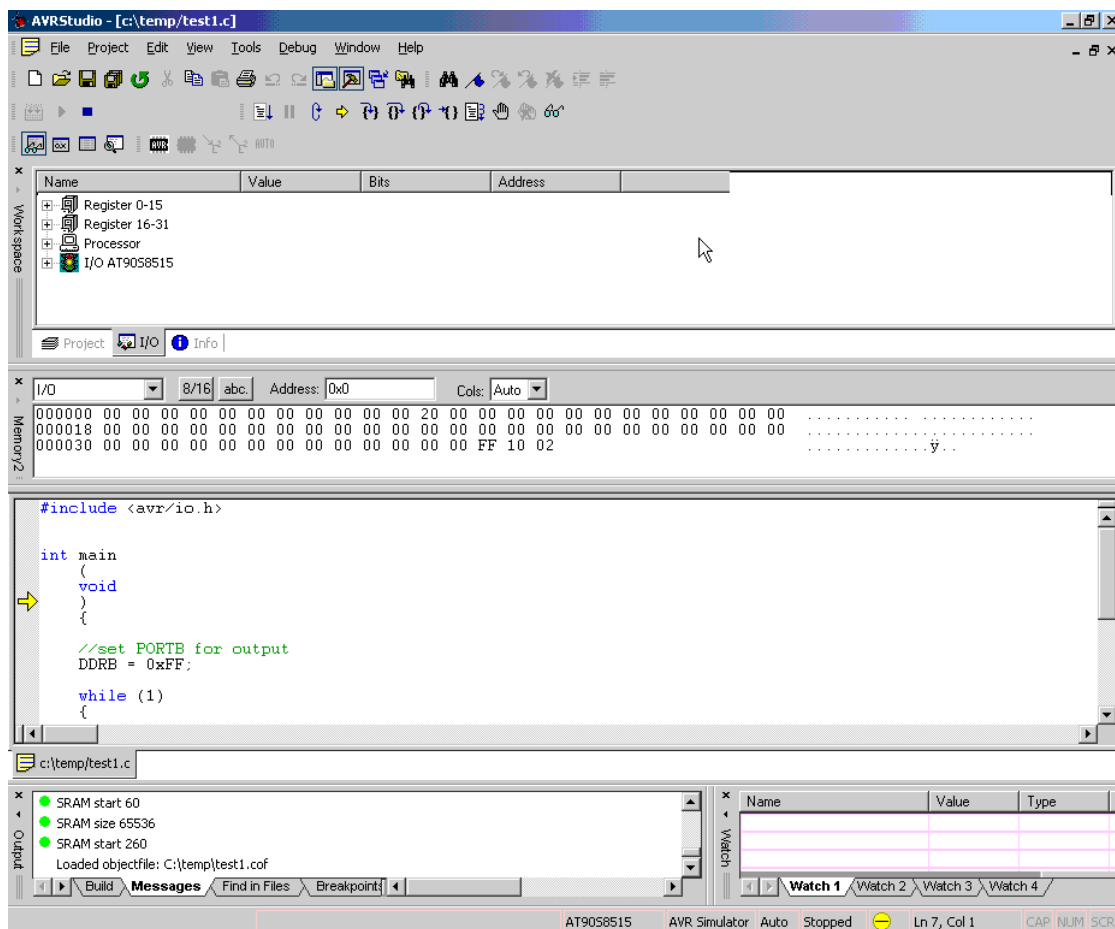
        //set PORTB.2 low
        PORTB &= ~(1<<2);

    }

    return 1;
}
```

فایل خود را 'Save' ذخیره کنید. اگر از AVRStudio 3.x یا AVRStudio 4.06 و پایین تر استفاده می کنید Make Coff و اگر از AVRStudio 4.07 یا بالاتر استفاده می کنید Make Extcoff را انجام دهید. سپس فایل test1.cof را در AVRStudio باز کنید. همچنین باید نوع چیب خود را مشخص کنید که باید همان چیب AVR ای باشد که در Makefile خود انتخاب کردید.

در پایان پنجره ای که ظاهر می شود باید چیزی شبیه تصویر زیر باشد:



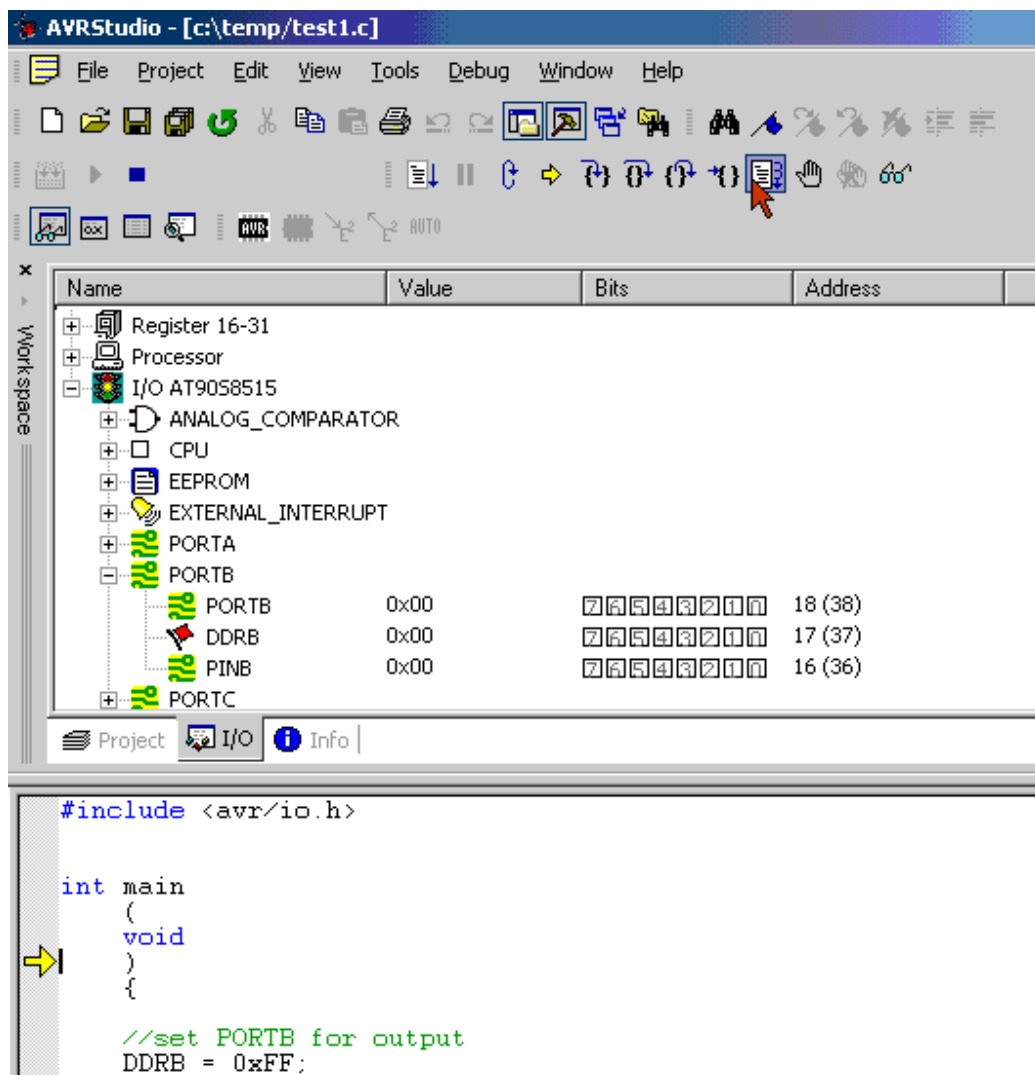
در اینجا کارهای زیادی می توانید انجام دهید که پیشنهاد می شود برای کسب اطلاعات بیشتر در منوهای بالای پنجره **Help → AVR Studio User Guide** را - انتخاب کنید.

اگر بعضی از پنجره ها را نمی بینید یا اتفاقاً آنها را بسته اید می توانید از منو 'View' آنها را پیدا کنید.

حال در پنجره **Workspace** با کلیک کردن روی علامت '+' از شاخه 'I/O' آن را باز کنید و بعد از آن شاخه 'PORTB'.

در آخر کلید **AutoStep (Alt+F5)** را بزنید و مشاهده کنید که **PORTB.2** به 1 و 0 تغییر می کند و برای توقف کلید **Break (Ctrl + F5)** را بزنید. همچنین می توانید از کلید **F11** که **Single-Step** می باشد و مرحله به مرحله روند برنامه را نشان می دهد، استفاده کنید.





## 7.0 اطلاعاتی بیشتر

چیزهای زیادی برای یادگیری **WinAVR** و ابزارهای همراه آن وجود دارد. برای اطلاعات بیشتر می توانید به دایرکتوری [c:\WinAVR\doc](http://c:\WinAVR\doc) بروید. سایت **WinAVR** نیز <http://winavr.sourceforge.net> اطلاعات مفیدی را از قبیل لینکها و اخبار جدید در اختیار شما قرار می دهد. اگر تا بحال به این سایت نرفته اید پیشنهاد می شود حتماً یک سری به آن بزنید.

از مهم ترین اسناد **WinAVR** به **avr-libc** می توان اشاره کرد که در دایرکتوری [c:\WinAVR\doc\avr-libc\avr-libc-user-manual\index.html](http://c:\WinAVR\doc\avr-libc\avr-libc-user-manual\index.html) قرار دارد. شما می توانید آخرین اطلاعات **avr-libc** را در وب سایت **avr-libc** پیدا کنید. <http://www.nongnu.org/avr-libc>

## 8.0 واژه نامه

**ANSI- the American National Standard Institute.** - بطور معمول کامپایلر فوق از استاندارد **ANSI C** بهره می برد. و با اطمینان بالایی می توان برنامه نوشته شده در یک کامپایلر **ANSI C** را در کامپایلر هایی که از این استاندارد استفاده می کنند با حداقل تغییرات به کار برد.

**ASM** - فایل های اسمبلی که برای میکروهای **AVR** ساخته شده اند.

**COFF-Common Object File Format** - که در **AVRStudio** برای اشکال زدایی (**Debugging**) به کار برده می شود. **AVRStudio 4.07** و بالاتر، از فرمت **COFF** استفاده می کند که دارای مقداری نکات بیشتر می باشد. **WinAVR** هردو فرمت **COFF** و **COF** (که برای **AVRStudio 4.06** و پایین تر همچون **AVRStudio 3.x** است) را پشتیبانی می کند. البته برای **COFF** باید بسته نرم افزاری آن را نصب کرده باشید.

**GUI - Graphical User Interface** - قسمت هایی در برنامه که برای ارتباط با کاربر و دریافت دستورات دارای کلید (**Button**) و پنجره (**Window**) هستند و بر خلاف بخشهایی که دستورات شما باید نوشته شوند. (**Command Line**)

**Ihex- The Intel Hex file format** - فایل هگز اینتل است که بیشتر پروگرامر های **AVR** از آن برای پروگرام کردن چیپ استفاده می کنند.

**PN - Programmers Notepad** - یک ویرایشگر مختلط و متنوع برای چندین نوع فایل است که به همراه بسته نرم افزاری **WinAVR** موجود می باشد.