



Mini get started

by

Torgeir Bjørnvold
24th April 2014



Atmel Studio

Installation

**ATMEL
STUDIO
6**

One Collaborative Studio
With Integrated App Store and
Shared Workspace



- Download, intro, and video's
 - www.Atmel.com/Studio
 - If you have Studio 6.1 installed on your computer, remove it before installing 6.2

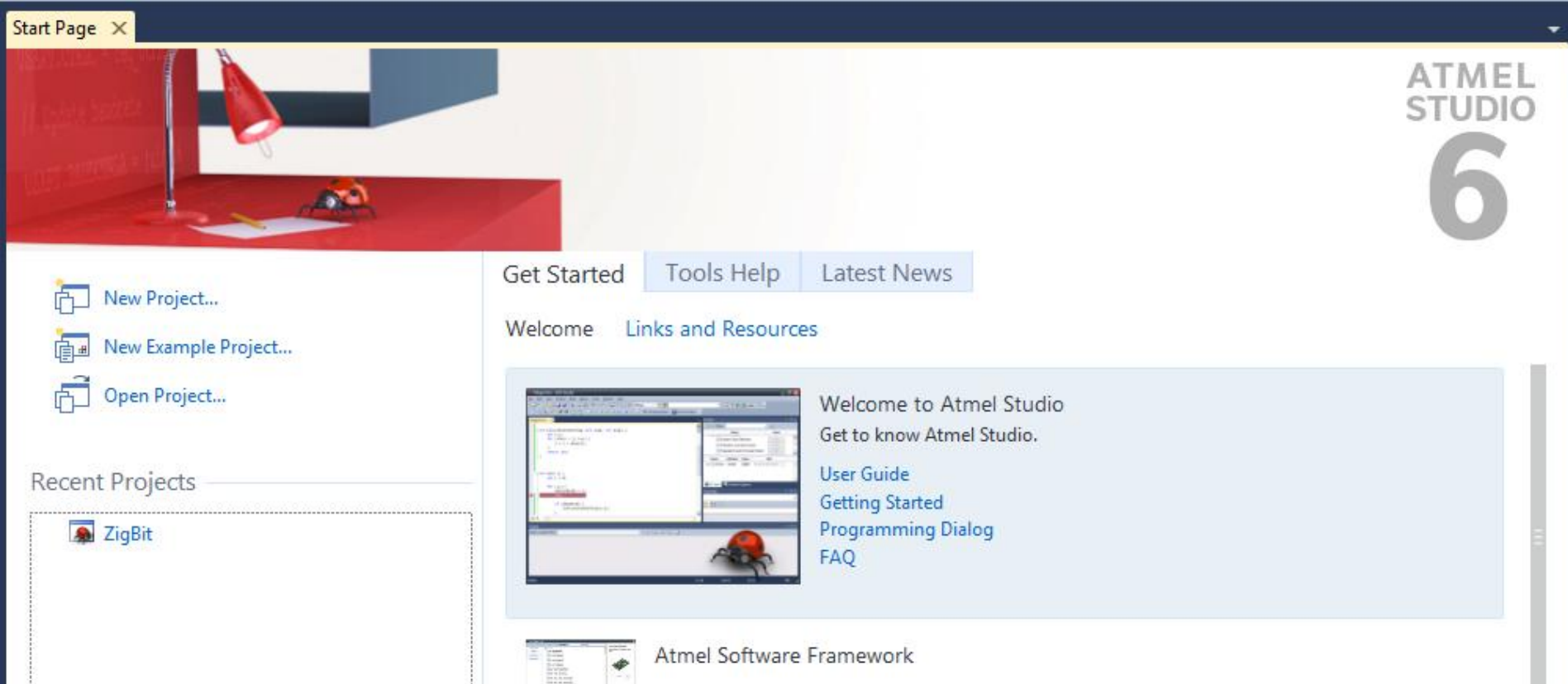
Source code

Download and unzip

- The source code is available as a downloadable ZIP file.
- http://spaces.atmel.com/gf/project/avr_xp_mini/
- Download the source code from Atmel Spaces and unzip it in a directory dedicated for SW projects (e.g. My SW Projects)

Source code

Open project in Atmel Studio

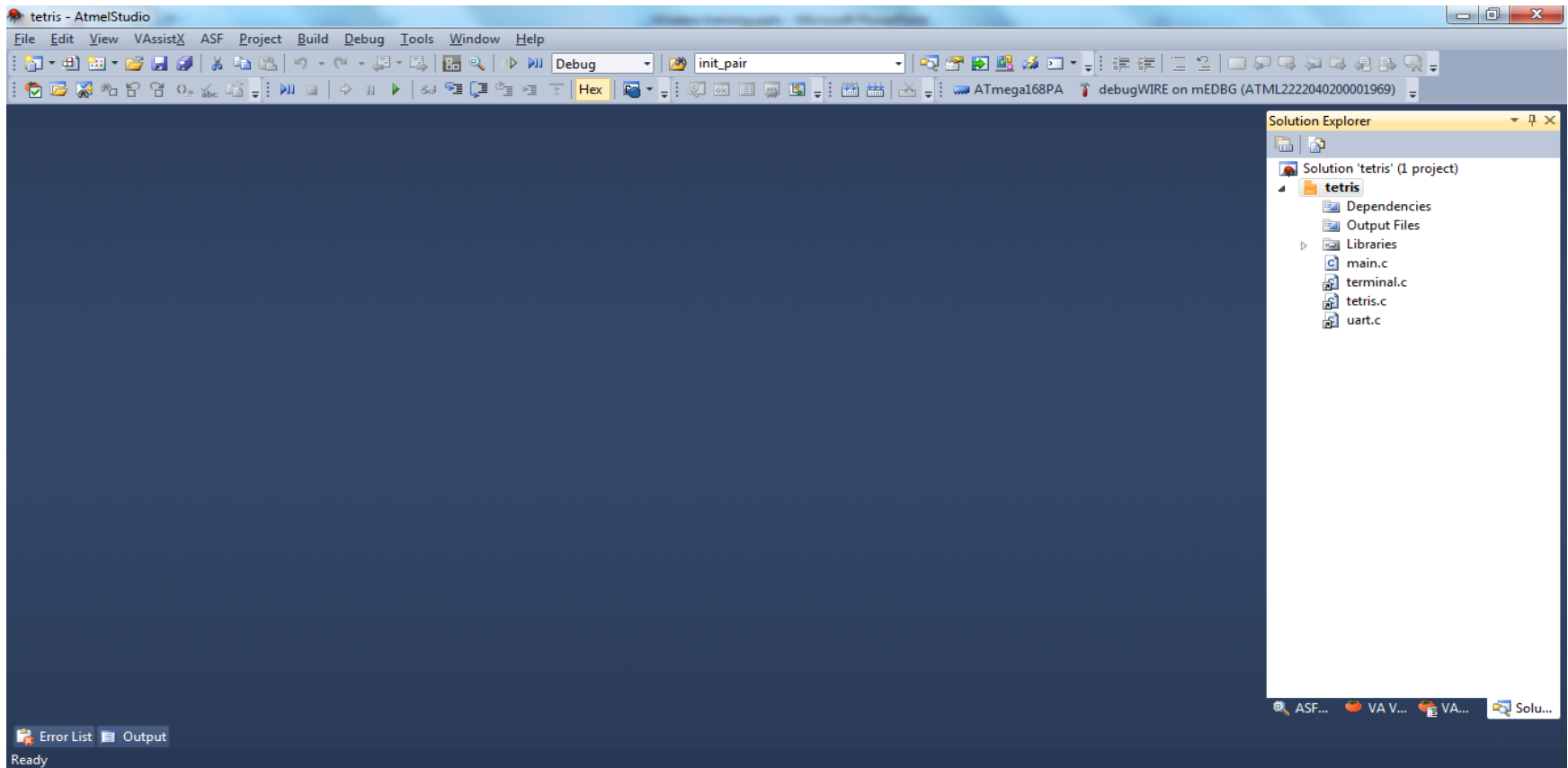


- Click Open Project
 - Brows to the directory where the code is located
 - Select the solution file (*.atsln)

Source code

Open project in Atmel Studio

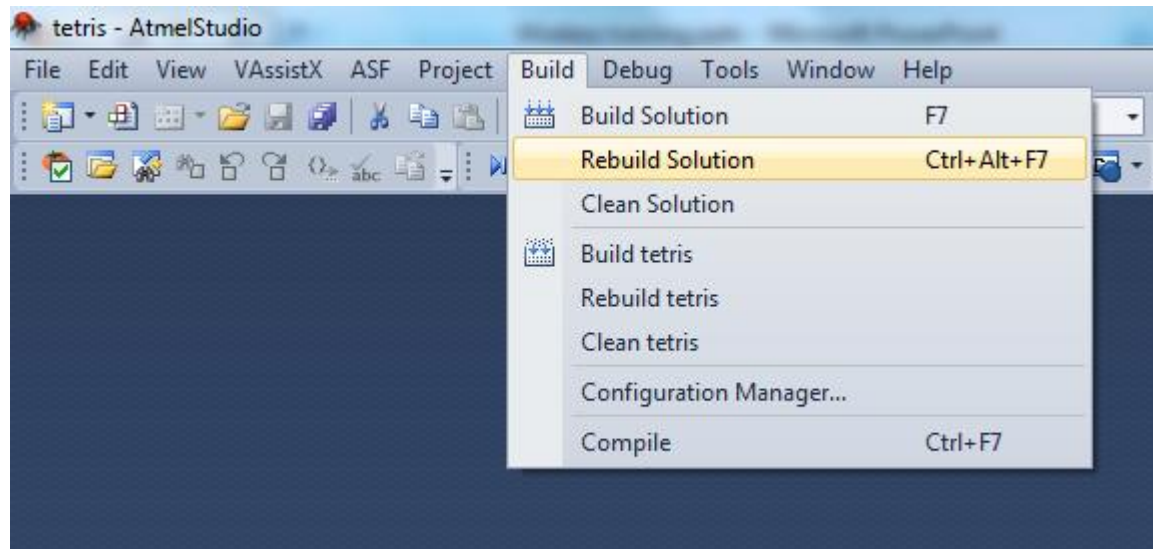
- The SW project is loaded into Atmel Studio and ready to use
- Double – click on the files to open them for editing



Source code

Compile

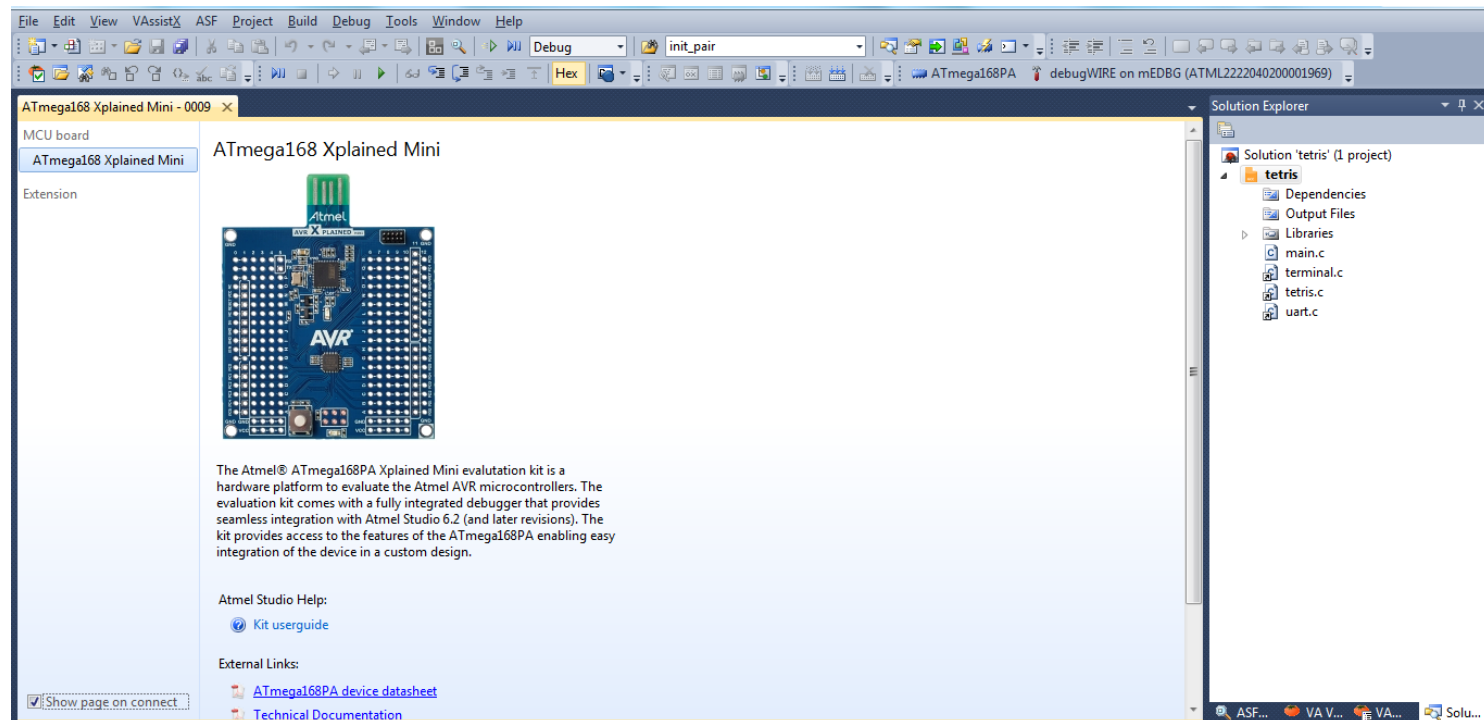
- Compile the code to get hex and elf files



Source code

Connect the Xplained Mini

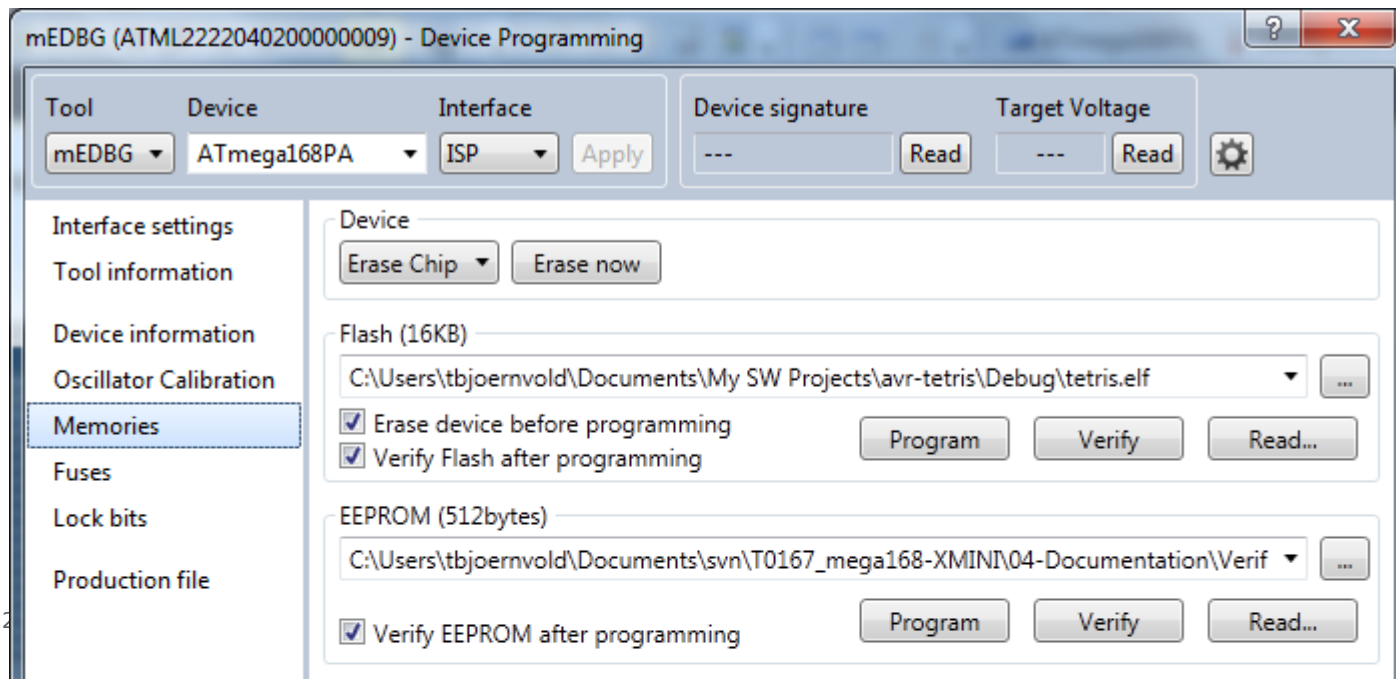
- Connect the Xplained Mini to the PC via the USB cable.
 - The device drivers are automatically installed, unplug the USB and plug in again to get a fresh start this first time



Source code

Program the Xplained Mini

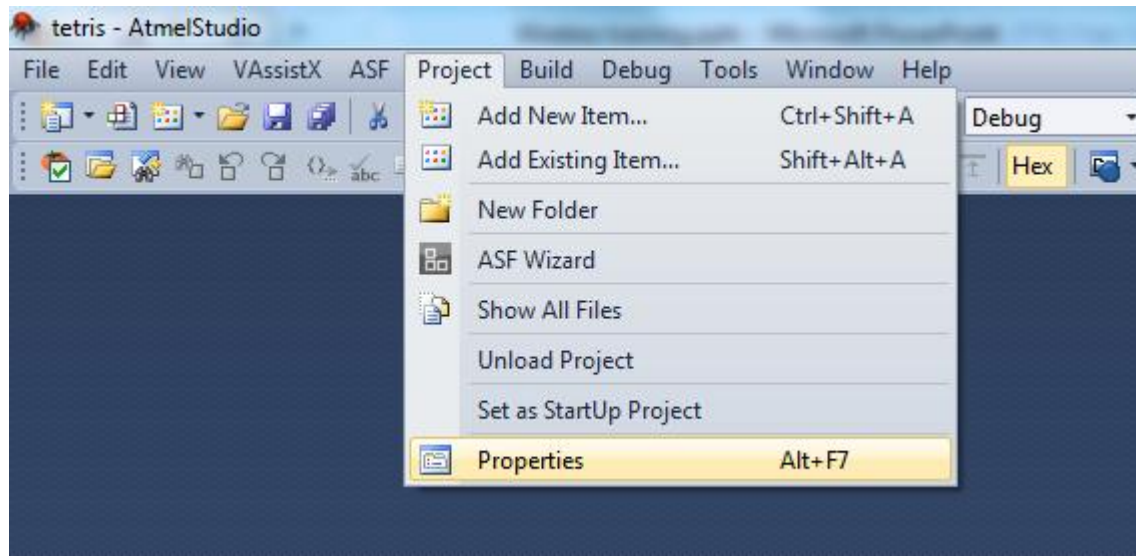
- Program the application
 - Studio – Tools – Device Programming
 - Select Tool for programmin = mEDBG
 - Device automatically selected
 - Apply and click memories
 - Select 256RFR2XplainedPRO.hex and click Program



Debug basic

Select Debugger

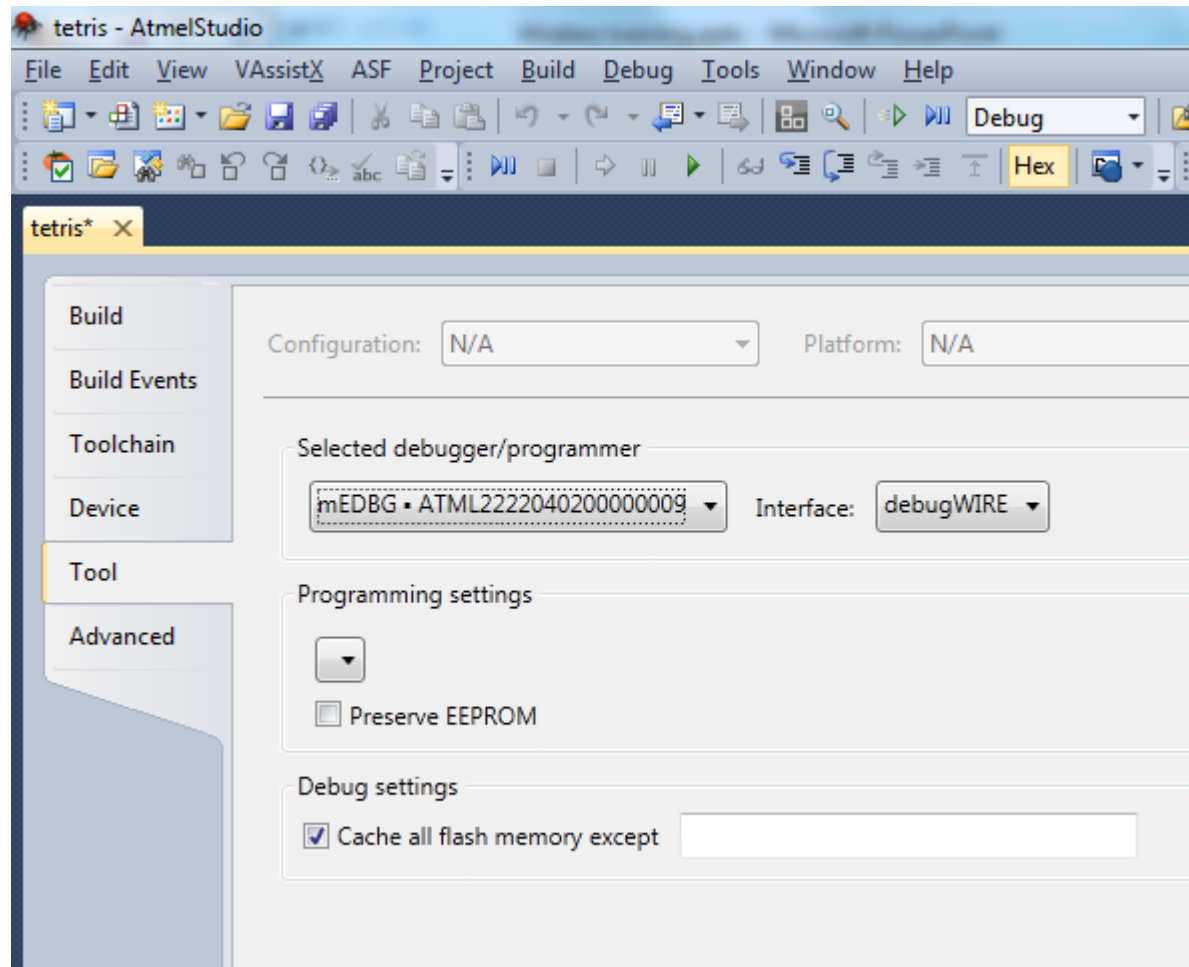
- Project - Properties



Debug basic

Select Debugger

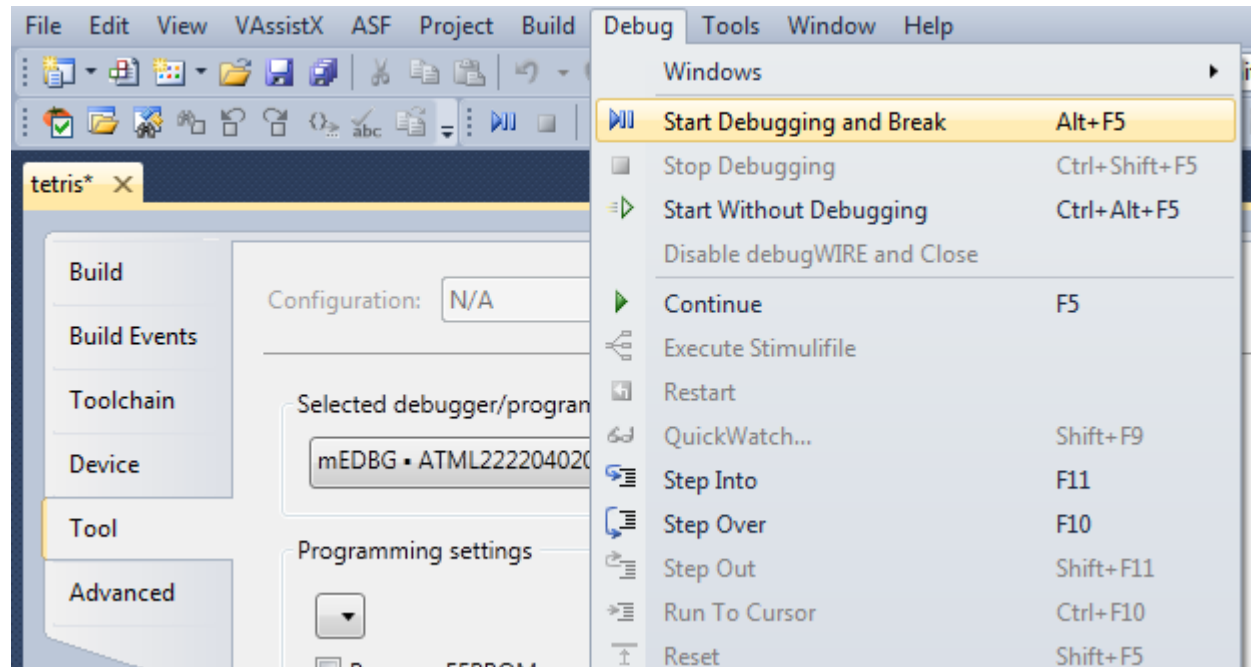
- Project – Properties
- Click the Tool-tab
- Select mEDBG as the debugger



Debug basic

Start debugging

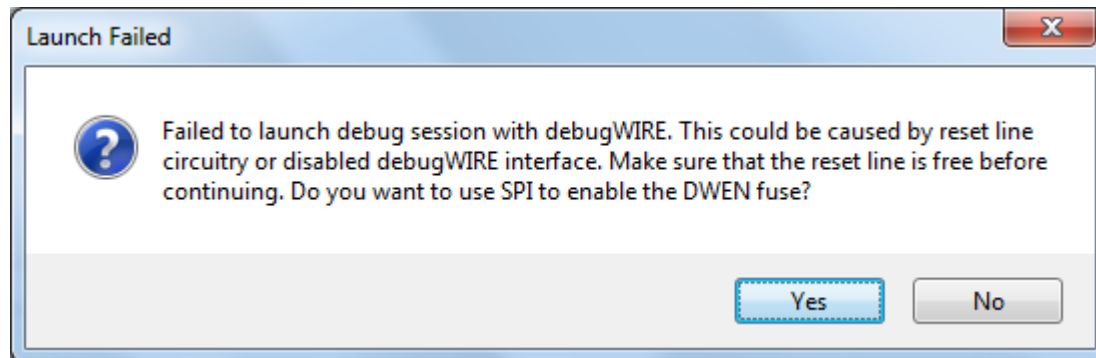
- Debug – Start Debugging and Break



Debug basic

Start debugging

- Debug – Start Debugging and Break
- If the DWEN fuse is not enabled and error message is displayed.
- Click Yes and Studio will use the ISP to set the fuse.



Debug basic

Start debugging

- The debugger is started and break's in main
- You are now ready to start debugging

The screenshot displays the AVR Studio IDE interface. The main window shows the source code for `main.c` with a breakpoint set at the start of the `main` function. The `main` function includes initialization for the timer and UART, and an endless application loop. The `Processor` window on the right shows the current state of the microcontroller, with the Program Counter at `0x00000053`. The `Memory` window at the bottom shows the program flash memory starting at address `0x0000`.

```
main.c x tetris
TIFR1 = 1 << OCF1R;
TIMSK1 = 1 << OCIE1A;
OCR1A = TIMER_TOP_VALUE;
TCCR1B = (1 << WGM12) | (1 << CS12);

uart_init();
terminal_init();
}

int main(void)
{
    unsigned char c;

    board_init();
    application_init();

    /*Turn global interrupts on*/
    sei();

    /*Endless application loop*/
}
```

Processor

Name	Value
Program Counter	0x00000053
Stack Pointer	0x04FD
X Register	0x02E5
Y Register	0x04FF
Z Register	0x12B0
Status Register	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Cycle Counter	0
Frequency	
Stop Watch	
Registers	
R00	0x00
R01	0x00
R02	0x68
R03	0x00
R04	0x6A
R05	0x00
R06	0x81
R07	0x00
R08	0x58
R09	0x00
R10	0x00
R11	0x00
R12	0x00
R13	0x00
R14	0x00
R15	0x00
R16	0x00
R17	0x00
R18	0x00
R19	0x00

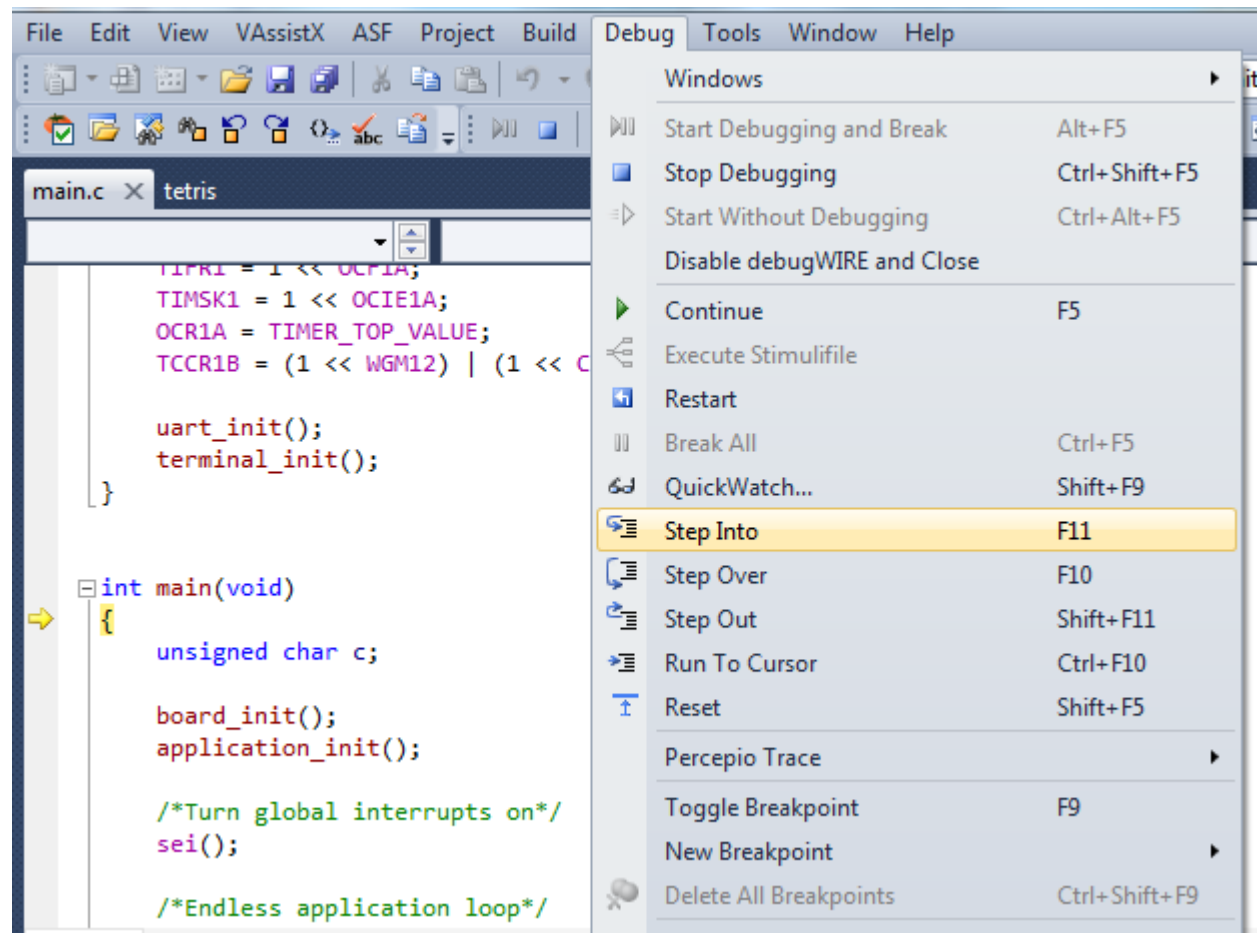
Memory1

Memory:	Address:	Value
prog FLASH	0x0000,prog	
prog 0x0000	0c 94 34 00 0c 94 51 00 0c 94 51 00 0c 94 51 00 0c 94 51	..4..Q..Q..Q..Q
prog 0x0013	00 0c 94 51 00 0c 94 51 00 0c 94 51 00 0c 94 51 00 0c 94	..Q..Q..Q..Q..Q..
prog 0x0026	51 00 0c 94 51 00 0c 94 d3 00 0c 94 51 00 0c 94 51 00 0c	Q..Q..Q..Q..Q..
prog 0x0039	94 51 00 0c 94 51 00 0c 94 51 00 0c 94 51 00 0c 94 c5 04	Q..Q..Q..Q..Q..A.
prog 0x004C	0c 94 f7 04 0c 94 51 00 0c 94 51 00 0c 94 51 00 0c 94 51	Q..Q..Q..Q..Q..Q..
prog 0x005F	00 0c 94 51 00 0c 94 51 00 11 24 1f be cf ef d4 e0 de bf	..Q..Q..Q..Q..Q..Q..
prog 0x0072	cd bf 11 e0 a0 e0 b1 e0 e2 e1 f2 e1 02 c0 05 90 0d 92 ae	Iz.à àààààà.À...'
prog 0x0085	39 b1 07 d9 f7 22 e0 ae e9 b1 e0 01 c0 1d 92 a5 3e b2 07	9t.Ù: "à"étà.À.'¥>..

Debug basic

debug

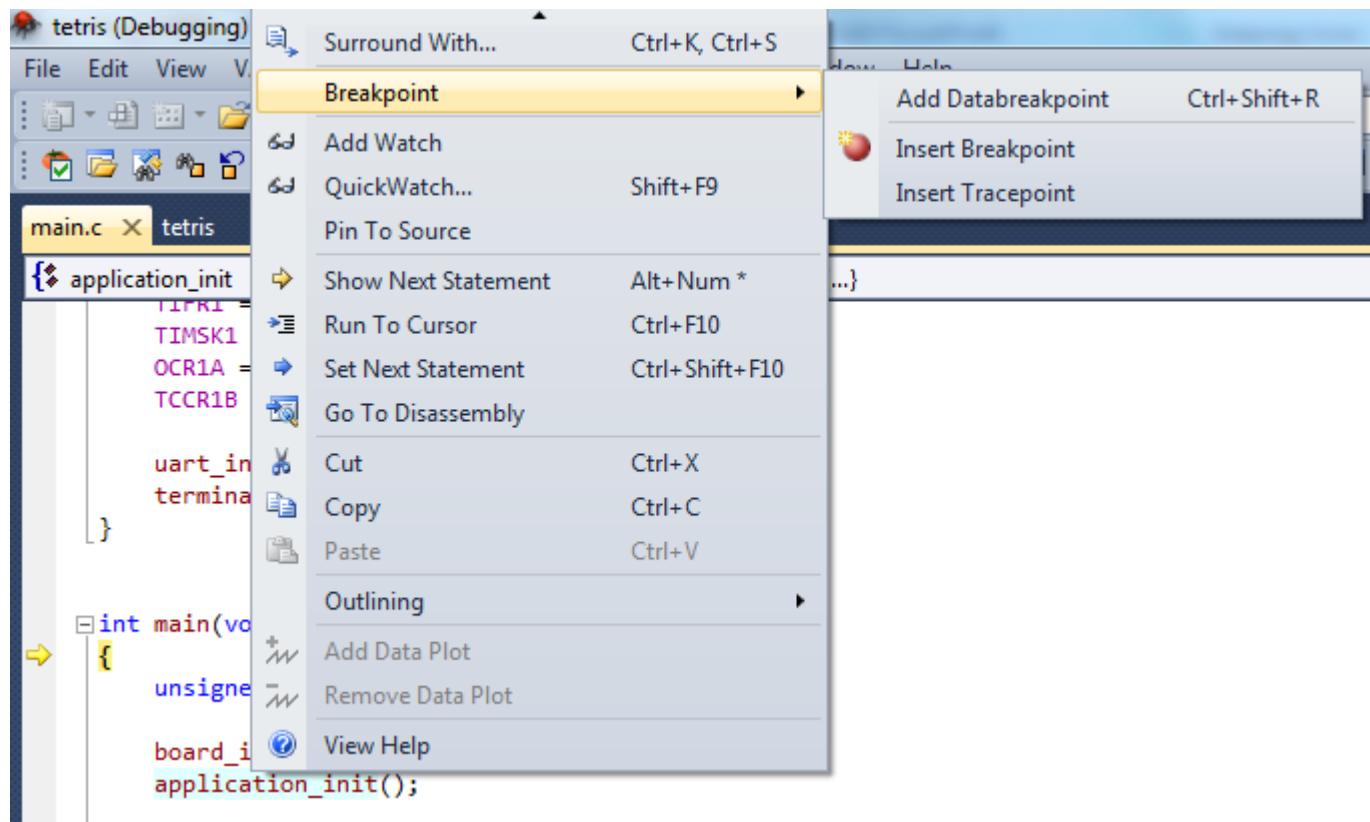
- A number of debug options is available in the debug menu



Debug basic

Break point

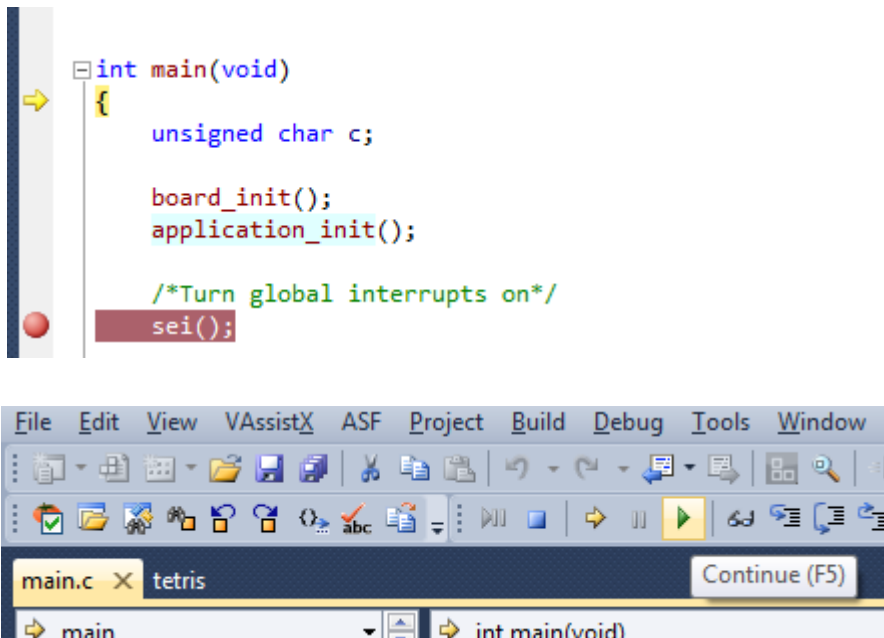
- Go to the line in the source code where you want to insert a breakpoint - right click – select Breakpoint and Insert Breakpoint



Debug basic

Break point

- Go to the line in the source code where you want to insert a breakpoint - right click – select Breakpoint and Insert Breakpoint
- The breakpoint is inserted
- Run to Breakpoint by Clicking "Continue"



Debug basic

Exit debug mode

- Debug – Disable debugWIRE and Close
- The target is returned to normal mode and the DWEN fuse is reset.

