Assembly Programming in Atmel Studio 6.2

Step by Step Tutorial

Atmel

Sepehr Naimi

BIHE University

4/3/2015

Contents

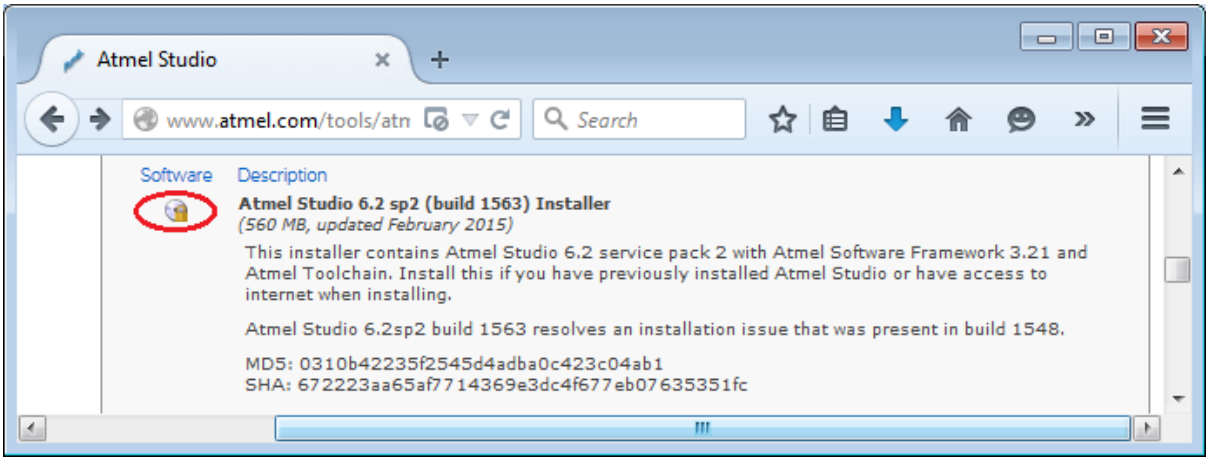
Introduction

This tutorial will teach you how to write, compile, and trace a simple program in Atmel Studio.

Downloading and Installing Atmel Studio

Download the newest version of Atmel Studio from the Atmel website:

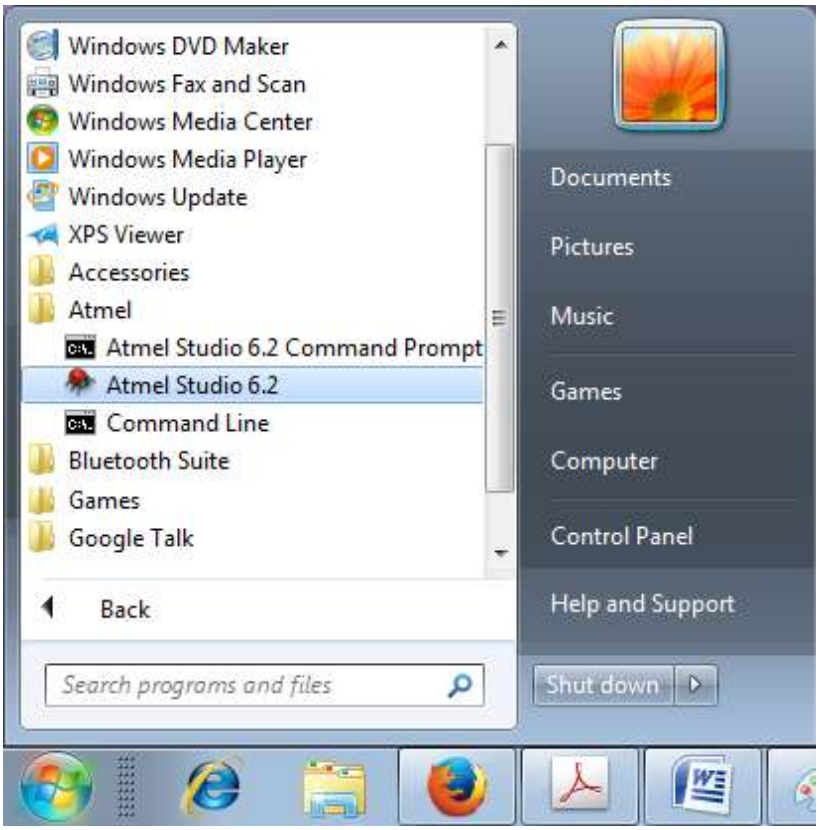
http://www.atmel.com/tools/atmelstudio.aspx



Run the downloaded program to install the Atmel Studio IDE.

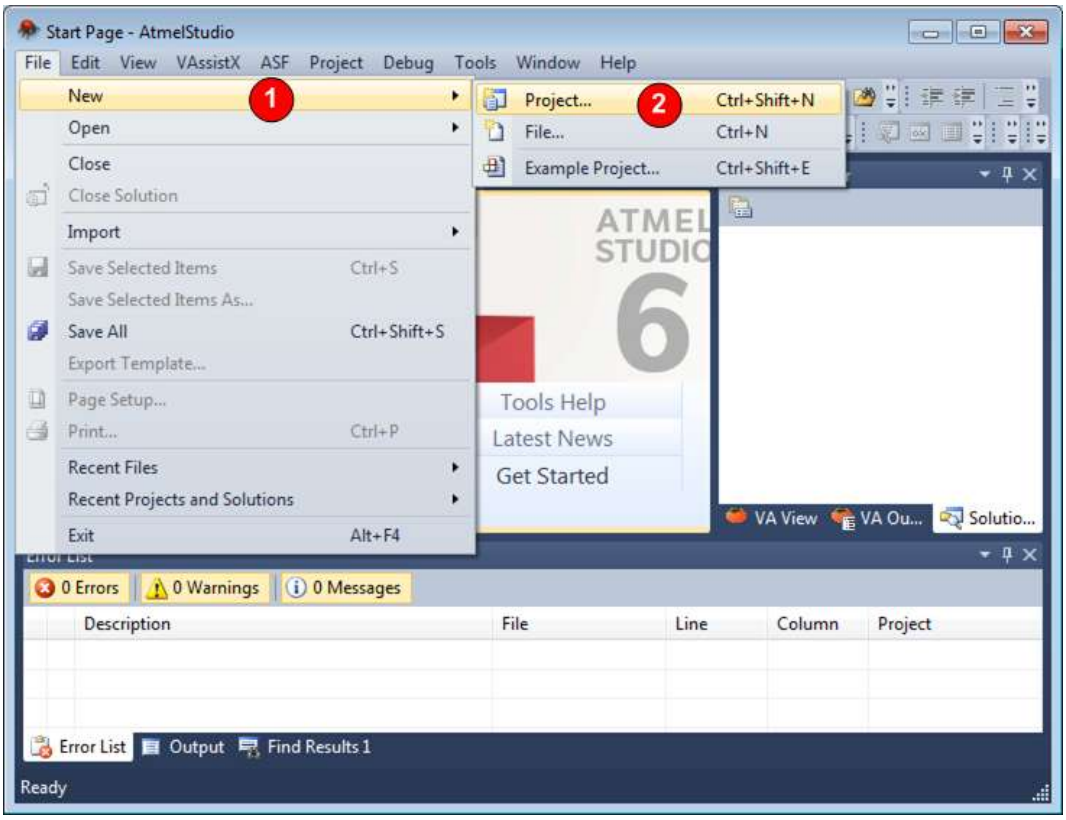
Opening Atmel Studio

Go to the ***Start*** menu and open Atmel Studio.

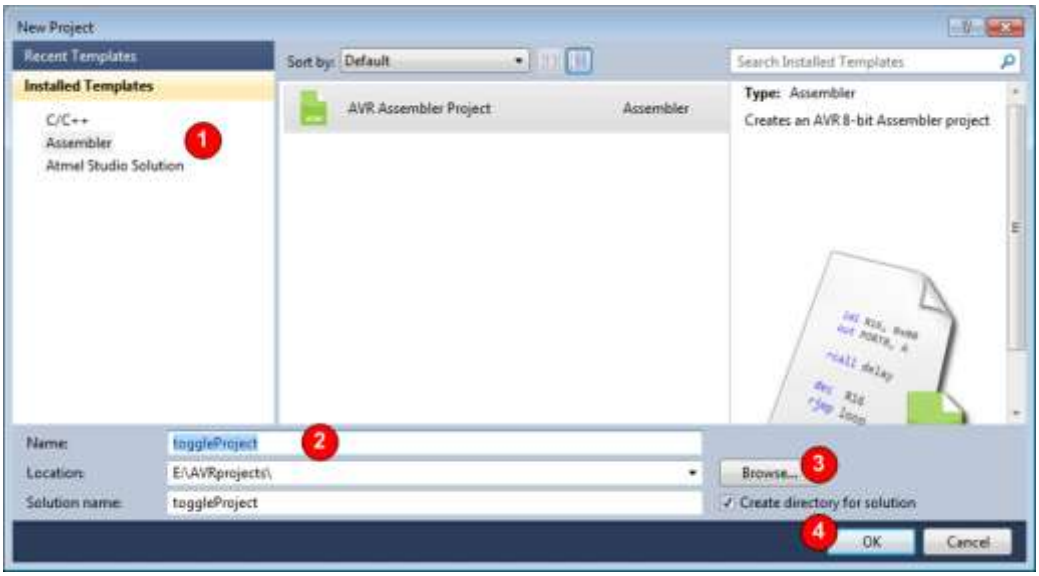


Creating the first project

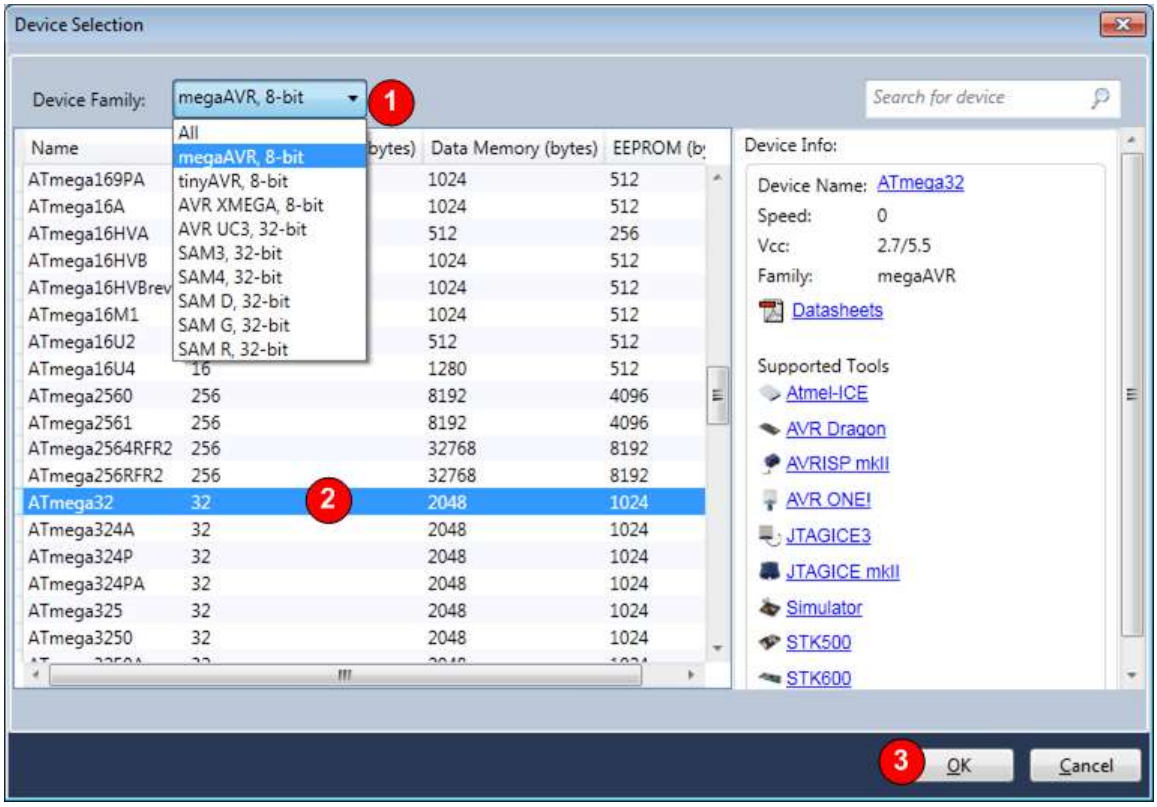
1. Go to the ***File*** menu. Choose ***New*** and then ***Project***.



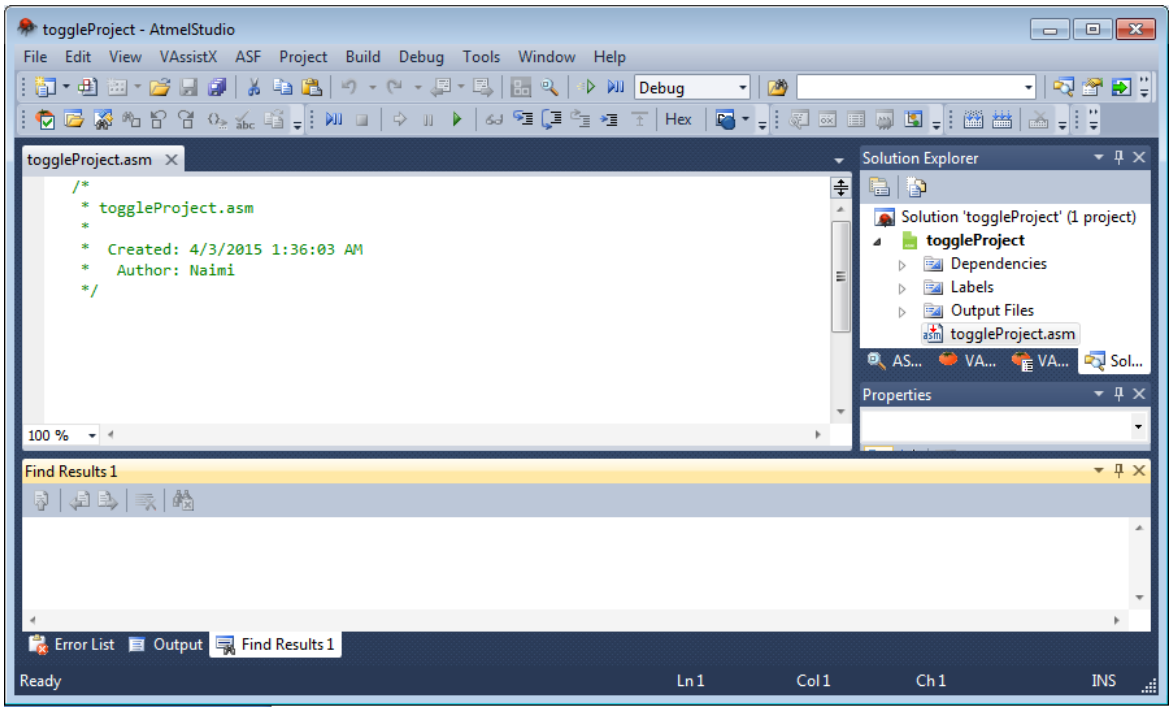
2. In the opened dialog,
    a. Choose ***Assembler***.
    b. Name the project as ***toggleProject***.
    c. Choose the path where you like to save the project by clicking on the ***Browse*** button.
    d. Press ***OK***.



3. In the ***Device Selection*** dialog
    a. Select ***megaAVR*** as the ***Device family***.
    b. Choose ***ATmega32*** (or any other Chips you want to use)
    c. Select ***OK***.



The compiler automatically makes the ***toggleProject*** and adds an assembly file to it.



Writing the first Assembly program

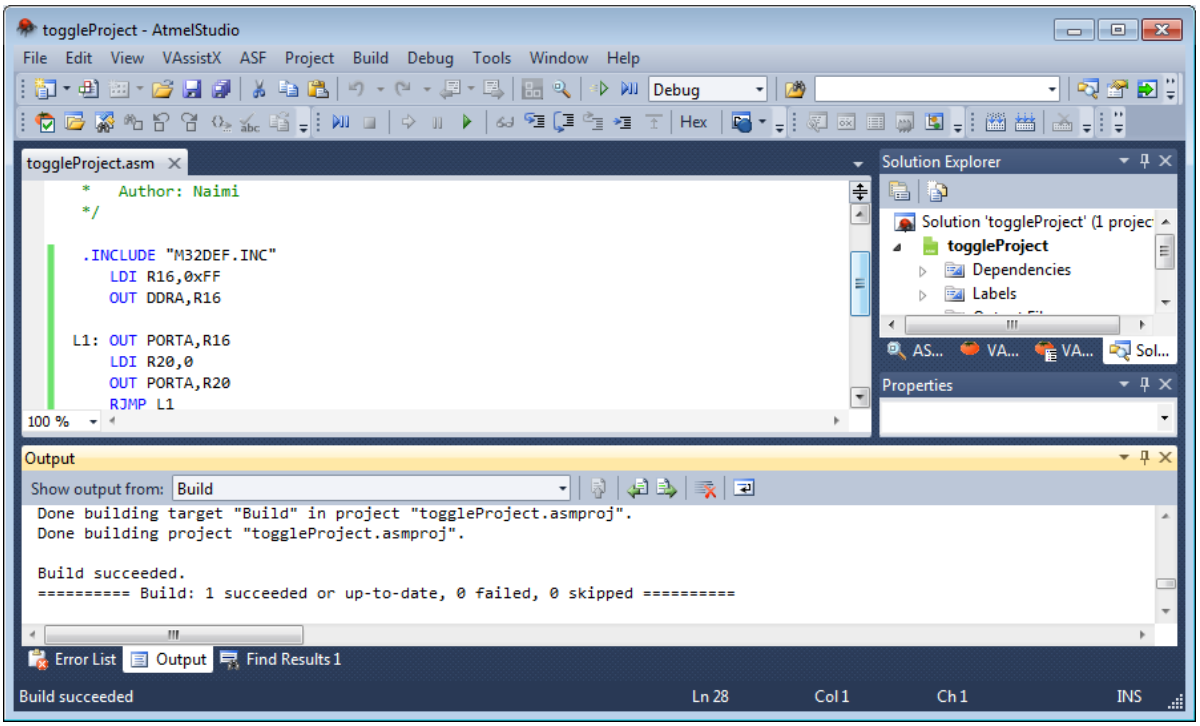
Type the following program.

```
.INCLUDE "M32DEF.INC"
       LDI R16,0xFF
       OUT DDRA,R16

L1:    OUT PORTA,R16
       LDI R20,0
       OUT PORTA,R20
       RJMP L1
```

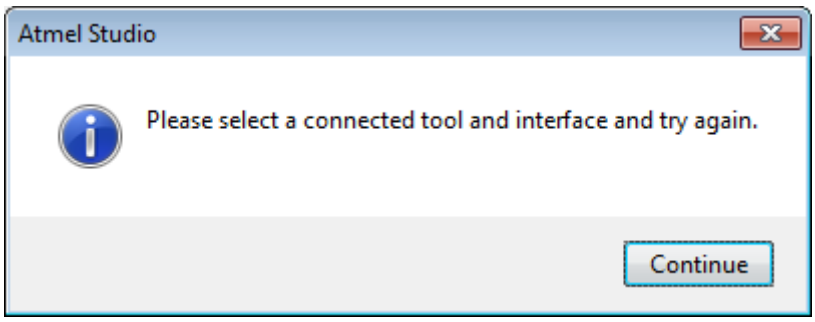
Building

Press ***F7*** to assemble, or choose ***Build Solution*** from the ***Build*** menu. The results of assembling the program are shown in the ***Output*** window.

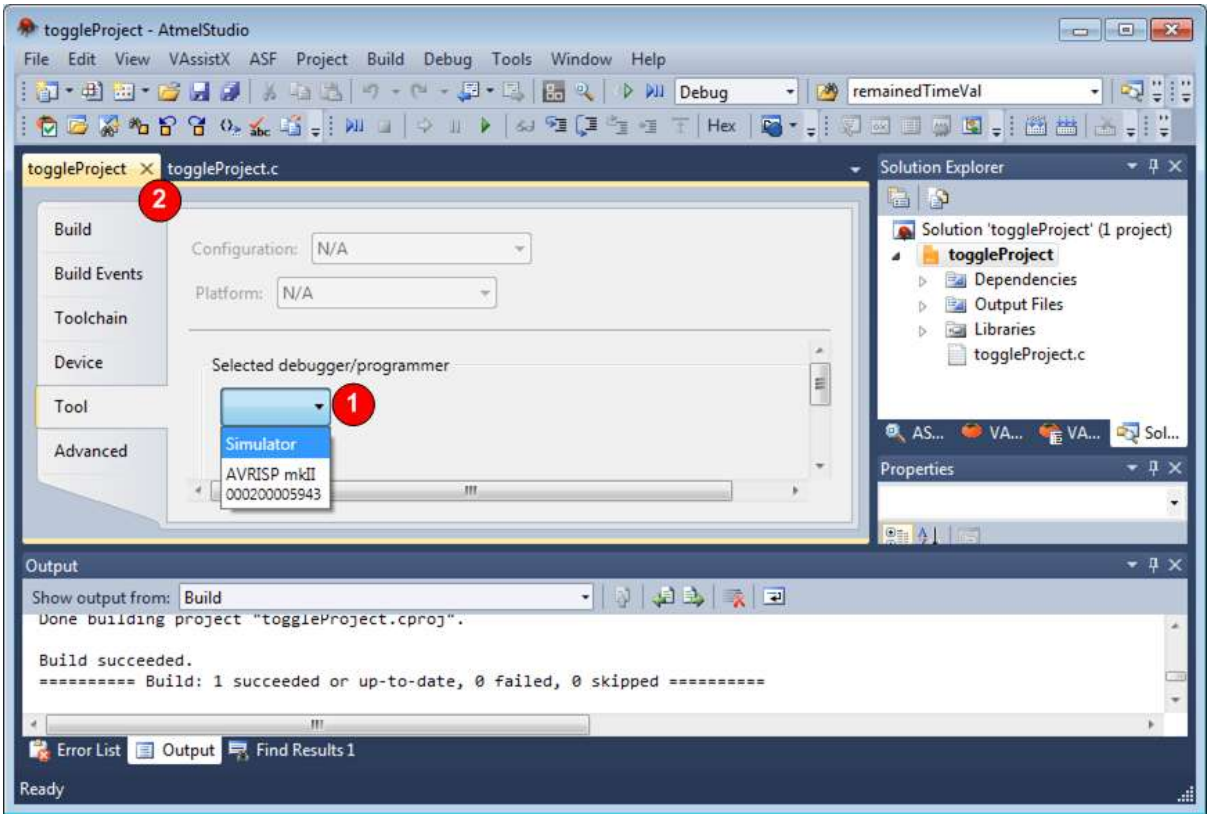


Debugging

1. To start debugging, press ***Alt+F5*** or choose ***Start Debugging*** from the ***Debug*** menu.
2. The following Dialog appears and asks you to select the debugging tool. Press ***Continue***.



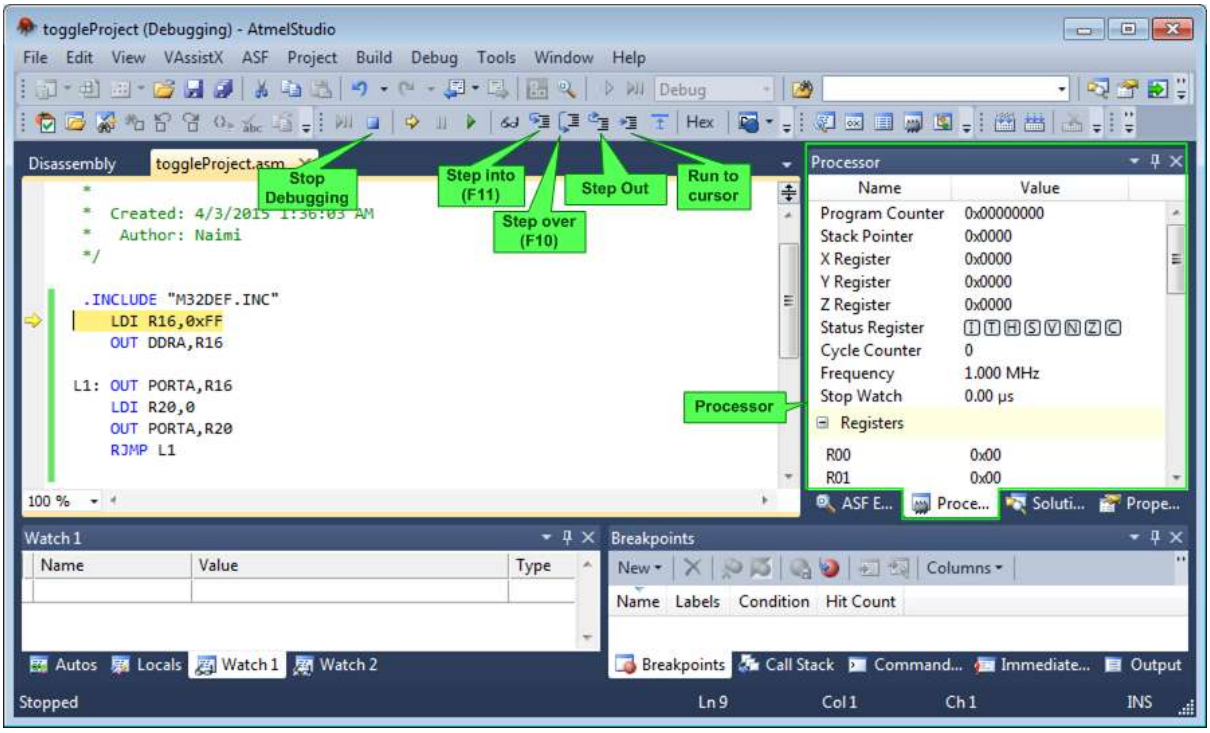
3. In the following window, choose ***Simulator*** as the debugger and then close it by pressing the x next to the ***toggleProject***.



Note: Simulator vs. debugger

Using the simulator you can execute the instructions, and watch the registers and variables.
If you have a debugger, e.g. AVRISP mkII or Atmel-ICE, you can connect a trainer board to your computer. In the case, the microcontroller of the board executes the same instructions, when you trace the program. This facilitates you to check the hardware while monitoring the variables in the IDE.

4. Press ***Alt+F5*** again. A ***Memory*** Watch windows appears which shows the contents of the Flash memory. Close the window.
5. Now a yellow cursor is on the first line of the main program and the IDE is ready to debug the program.



6. To execute the instructions line by line press ***F10*** or click on the ***Step over*** icon.

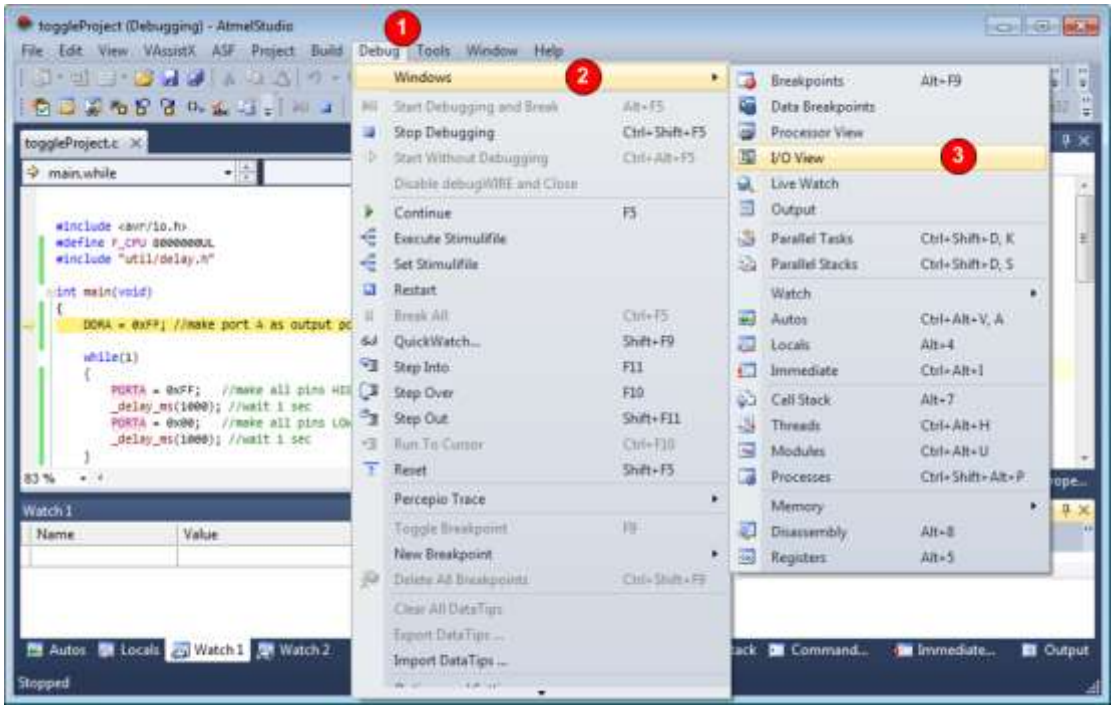
Step Into vs. Step Over
Both ***F10 (Step over)*** and ***F11 (Step into)*** execute one instruction and go to the next instruction. But they work differently when the cursor is on a function call. If the cursor is on the function call, ***Step into*** goes into the first instruction of the function, but ***Step Over*** executes the whole function and goes to the next instruction.

Step Out
If the execution is in a function, you can execute the function to the end by pressing the ***Step Out***.

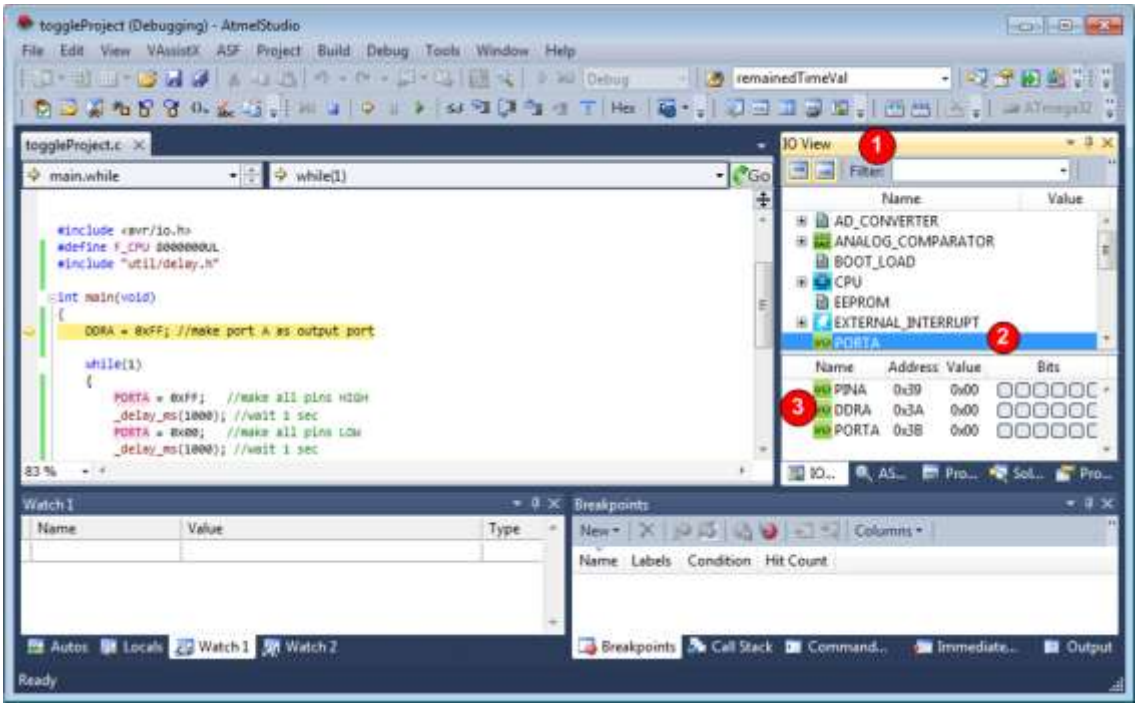
Run to Cursor
You can put the cursor on an instruction and then press the Run to Cursor button. In the case, the program runs until it reaches the instruction which the cursor is on it.

Processor Tab
The Processor tab shows the current values of the CPU registers including R0-R31, SP (Stack Pointer) and PC (Program Counter). You can also change the values of registers by double clicking on their values and typing a new value.

7. To monitor the peripherals, including the I/O ports, click on the ***Debug*** menu, choose ***Windows*** and then ***I/O view***.



8. The ***I/O view*** tab appears on the right hand side which shows the peripherals of the microcontroller, including the I/O ports. Select ***PORTA***. The values of the related registers (PINA, DDRA, and PORTA) will be shown below.

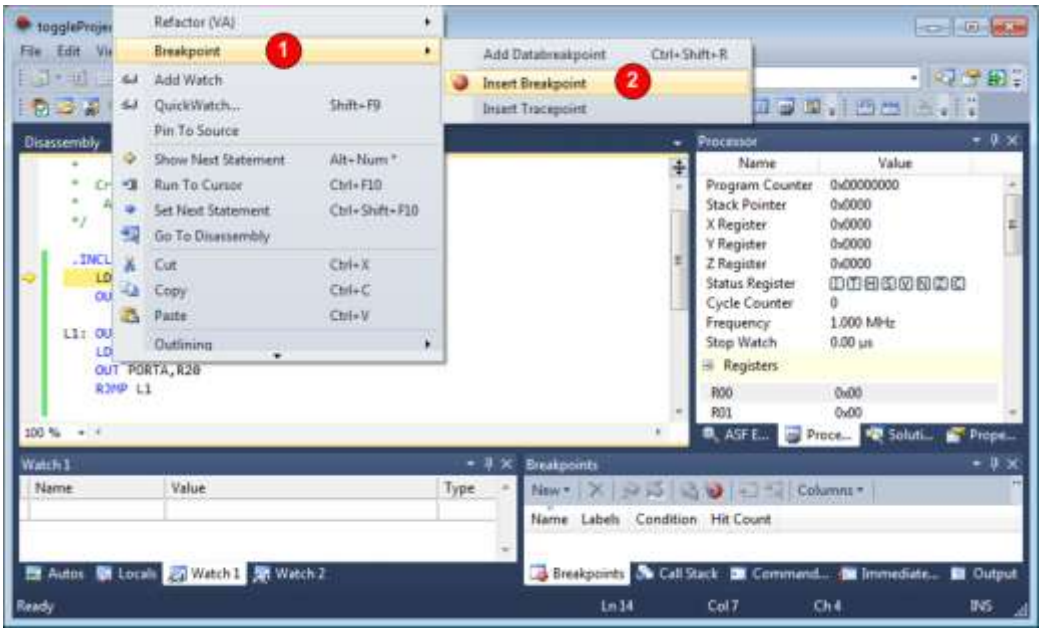


9. Press ***F10 (Step Over)*** a few times and see the PORTA register changes in the ***I/O View***.

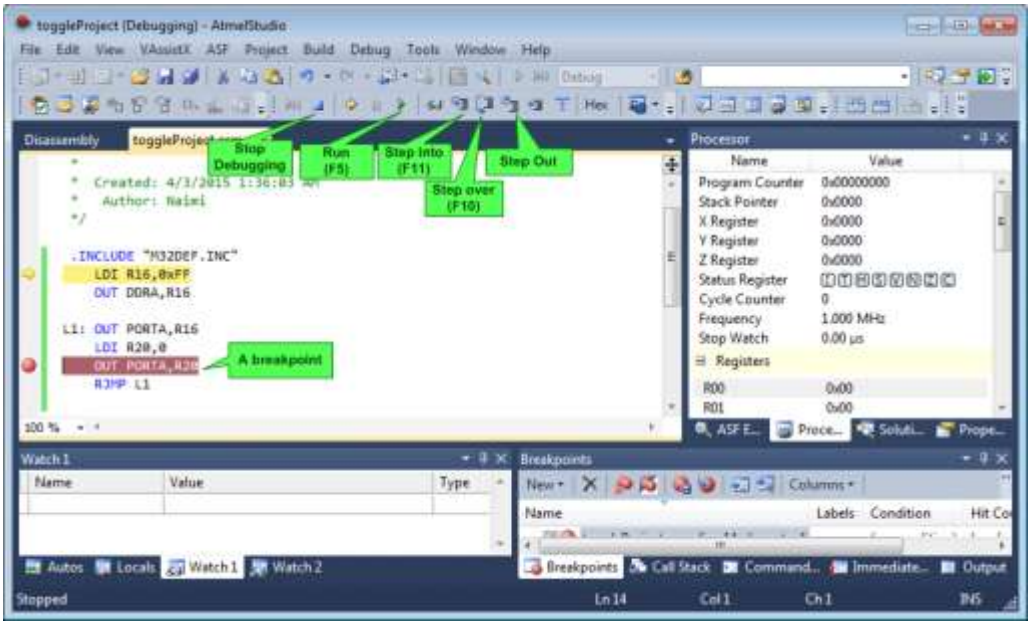
Using Breakpoints

If you want to debug a portion of a program, add a breakpoint to the beginning of this part of the code and press the run button. The IDE runs the program and when it reaches the breakpoint, it stops running and the yellow cursor is shown on the breakpoint line. Below, you see the steps in detail.

1. Right click on the "`OUT PORTA,R20`" instruction. A pop-up menu appears. Choose ***Breakpoint*** and then Insert Breakpoint. A red bullet appears on the hand side of the "`OUT PORTA,R20`" instruction.



2. Press ***F5*** or the ***Run*** button. The IDE runs program until it reaches the Breakpoint. Now, you can continue debugging from the breakpoint using the ***Step into*** and ***Step over*** buttons.



3. Using the ***Stop Debugging*** icon you can stop debugging whenever you want.