



8-bit **AVR**[®]
Microcontrollers

Application Note

AVR1900: Getting started with ATxmega128A1 on STK600

1 Introduction

This document contains information about how to get started with the ATxmega128A1 on STK[®]600.

The first three sections contain information about the software (AVR Studio[®], WinAVR and IAR Embedded Workbench[®] for AVR[®]) needed to compile and debug code for ATxmega128A1. The next section contains a brief overview on how to set up the STK600 for ATxmega128A1. The final section shows how to create a project and compile code for ATxmega128A1 using IAR Embedded Workbench or WinAVR, and how to start debugging the code on the STK600 using AVR Studio and JTAGICEmkII.

More Information about where to find the software tools can be found in chapter 7.

Rev. 8107A-AVR-04/08



2 AVR Studio that supports ATxmega128A1

AVR Studio is used to debug code. It contains a simulator and a frontend for the JTAGICE mkII for doing on chip debugging on AVR XMEGA. To have support for STK600 and ATxmega128A1, AVR Studio version 4.14 or later must be installed.

Note: During installation of AVR Studio remember to install the USB driver.

3 Installing IAR Embedded Workbench 5

IAR Embedded Workbench 5.10 (or later) for AVR (IAR EW Evaluation and Full version) has compiler support for ATxmega128A1.

To start the installation use the autorun application, then select “Install IAR Embedded workbench”.

Figure 3-1: Installing IAR EW for AVR.



During installation it is possible to use the same License number and License key as for previous versions of IAR Embedded Workbench.

It's recommended to use the default settings during the installation of IAR. How to start a project in IAR EW is presented in chapter 6.1.

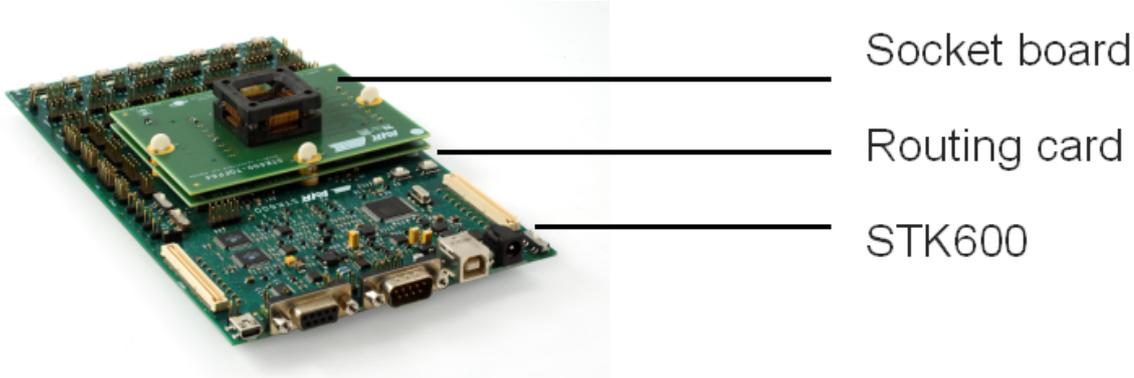
4 Installing WinAVR

Download WinAVR release 20080411 (or later) from Sourceforge. Run the installer and follow the instructions. The avr-gcc toolchain will integrate with AVR Studio automatically. How to start a project in WinAVR is presented in chapter 6.2.

5 Setting up STK600 for ATxmega128A1

STK600 is designed to support all 8-bit AVR devices and AVR32 devices with flash. A system based on socket and routing cards is used to support different package types and pinouts on the STK600 board.

Figure 5-1: STK600 socket and routing system

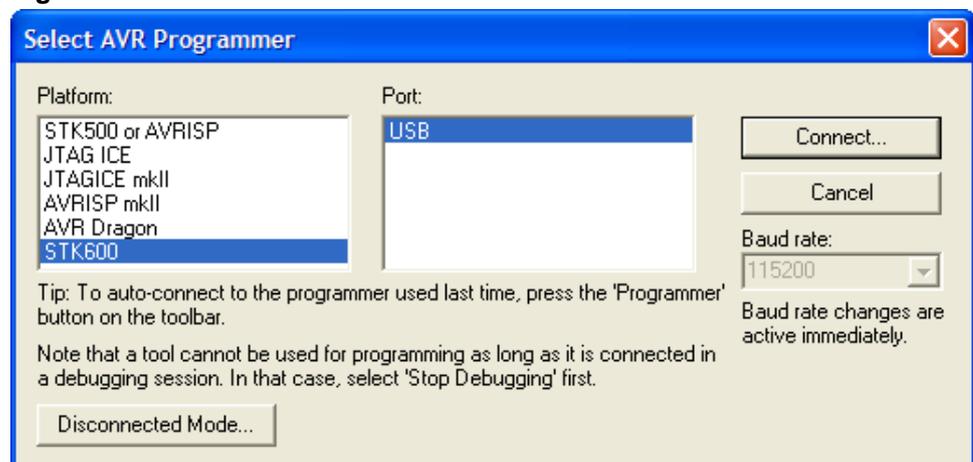


Before using the STK600 and before mounting any socket and routing card it is recommended to check that STK600 contains the newest firmware and that it has the right VTarget voltage. This can be done as described below.

Note: AVR Studio should be installed before connecting to the STK600 to ensure that the drivers are installed.

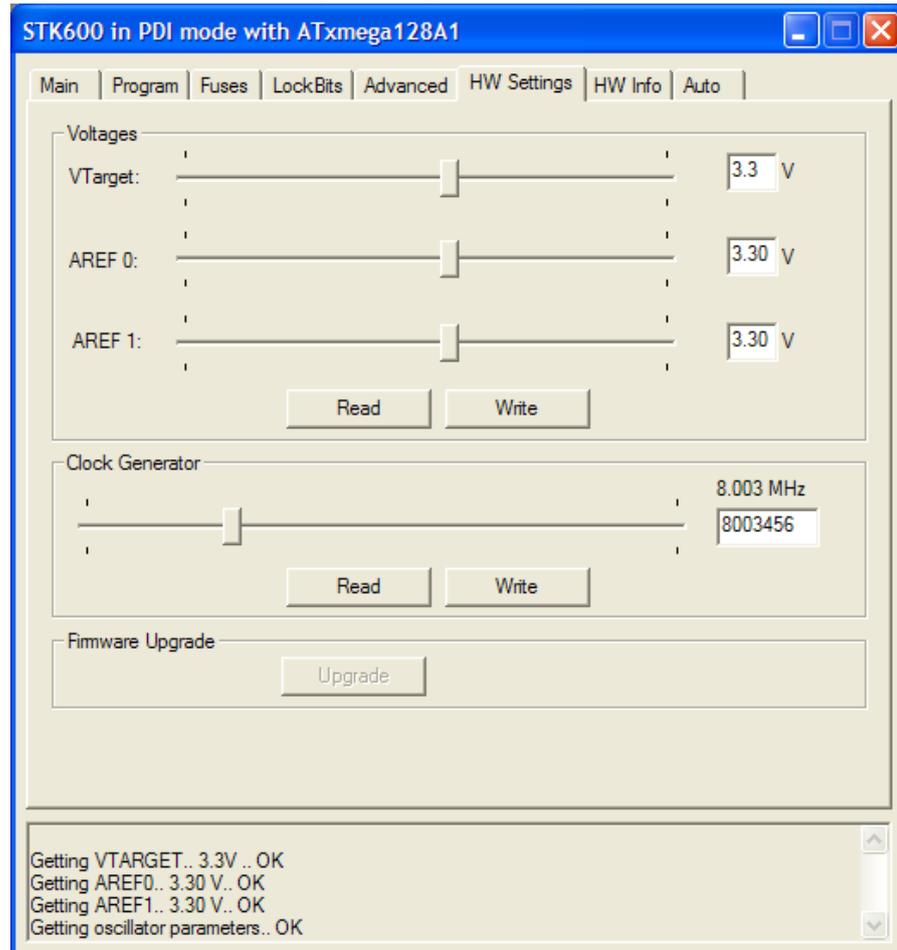
1. Connect a USB cable to the STK600 and turn it on.
2. Windows® will tell you that new hardware is found. Install the driver for STK600.
3. Open AVR Studio and cancel the startup dialog.
4. Start a connection with STK600 by selecting "Tools -> Program AVR -> Connect..." from the menu and select the STK600 and USB as illustrated in Figure 5-2.

Figure 5-2: Connect to the STK600.



1. If a newer firmware version for the STK600 is available AVR Studio will ask if you what to do an upgrade. Follow the instructions given by AVR Studio to do the firmware upgrade.
2. After the upgrade connect to the STK600 again (in the same way as 4.) and open the HW Setting page. Set VTarget voltage between 1.8 V and 3.6 V and write the value to STK600. A read back to verify that the right value was written is recommended.

Figure 5-3: Set the VTarget voltage

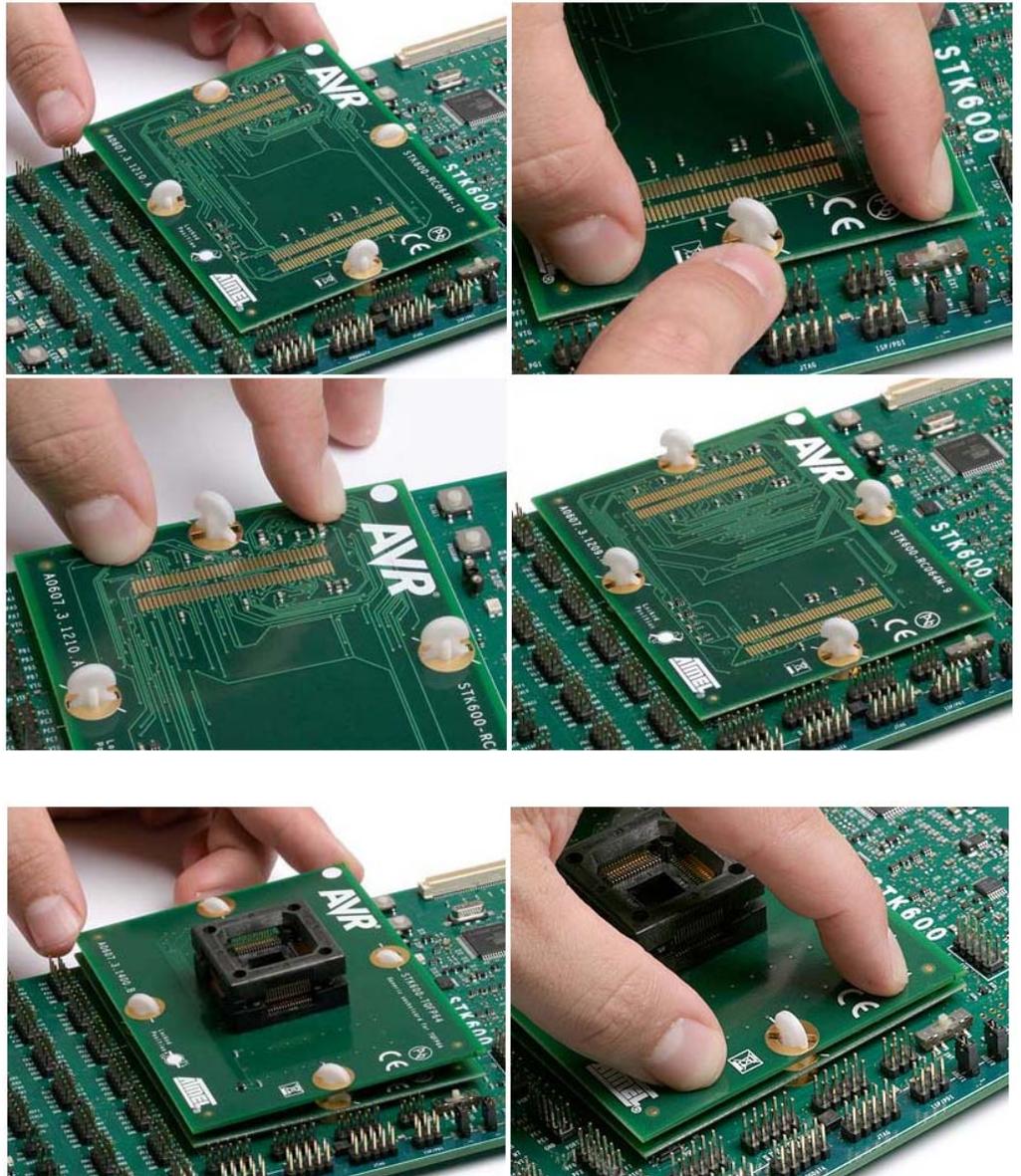


When the newest firmware and the right target voltage is set, turn off the STK600 before mounting the routing card and socket card.

The routing card used for ATxmega128A1 is “**STK600 – RC100X- 13**” and the socket card used for 100 pin devices is “**STK600 – TQFP100**”

How to mount the routing card and socket card to the STK600 is shown in Figure 5-4. It is not possible to mount the routing and socket card in a wrong orientation.

Figure 5-4: How to mount the routing and socket card to the STK600



When mounting the device in the ZIF socket ensure it is mounted in the correct orientation. Pin one on the device is marked with a dot. As seen on Figure 5-5 this corner must be orientated in the same direction as the cut corner on the socket.

Figure 5-5: Mounted device in ZIF socket

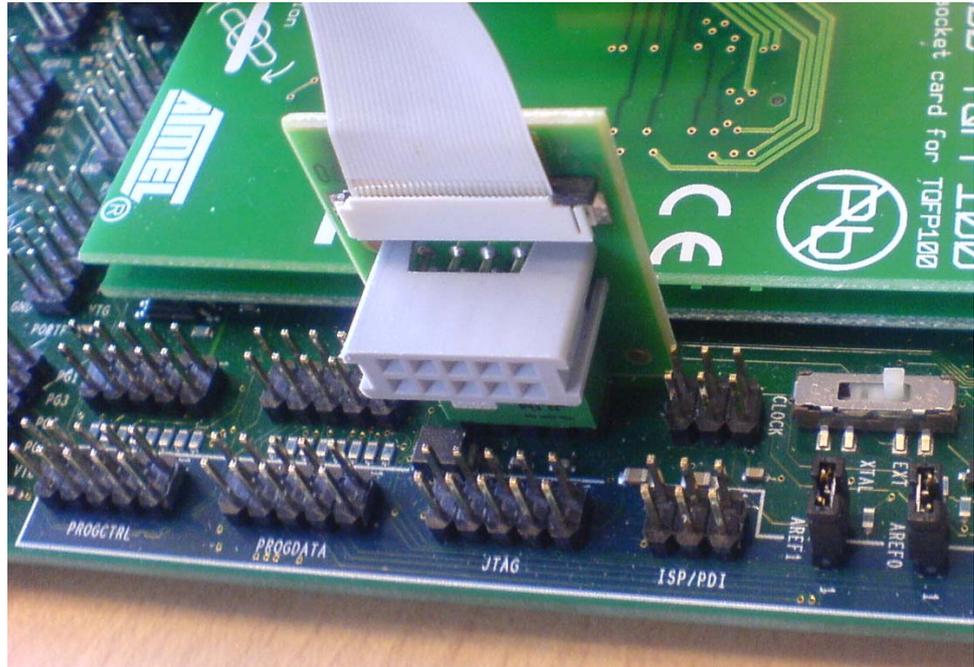


The STK600 can source power through the USB cable. Remember that the power available through the USB cable is limited. If the STK600 is connected to external hardware that consumes more than 300mA, you should use an external power source connected to the DC input socket on STK600. The external power supply should be 10-15V DC with **positive center** connector.

Debugging of the code is done with AVR Studio and JTAGICEmkII. How to do On-Chip Debugging using the JTAGICE mkII to STK600 is described below:

1. Connect the JTAGICE mkII to the STK600 as show below:

Figure 5-6: Connecting JTAGICE mkII to STK600.



2. Turn on power on STK600 and JTAGICE mkII. (If it is the first time JTAGICE mkII is used, Windows will ask you to install the driver.)

More information about STK600 and JTAGICE mkII is available in AVR Studio. (Help -> AVR Tools User Guide).

6 Compiling code and debugging code on chip

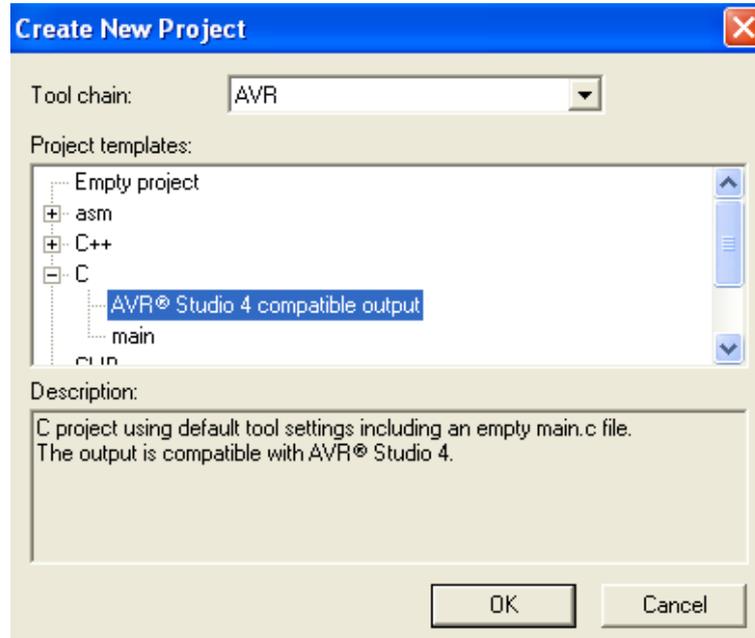
This section describes how to make a project, compile and debug on-chip with both IAR EW and WinAVR. For development with IAR EW read chapter 6.1. For development using WinAVR read chapter 6.2.

6.1 Compiling and debugging with IAR and AVR Studio

To start a project for the ATxmega128A1 device in IAR EW AVR do as described below:

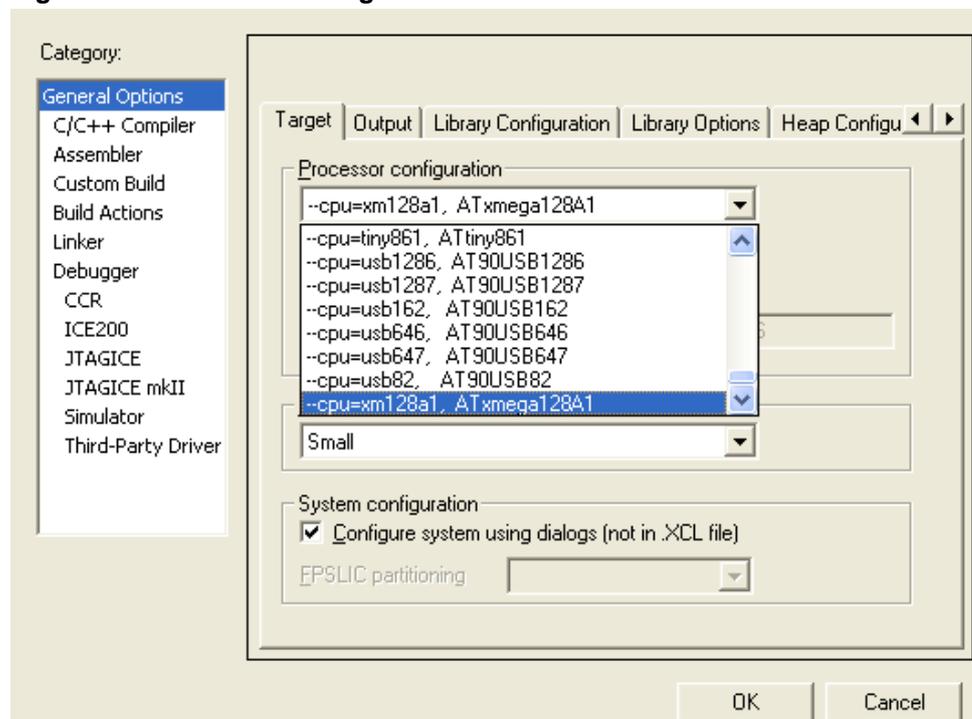
- Open IAR Embedded Workbench
- Select "Project -> Create New Project.." in the menu
- Select the desired coding language with AVR Studio 4 compatible output.

Figure 6-1: Create new project in IAR EW for AVR



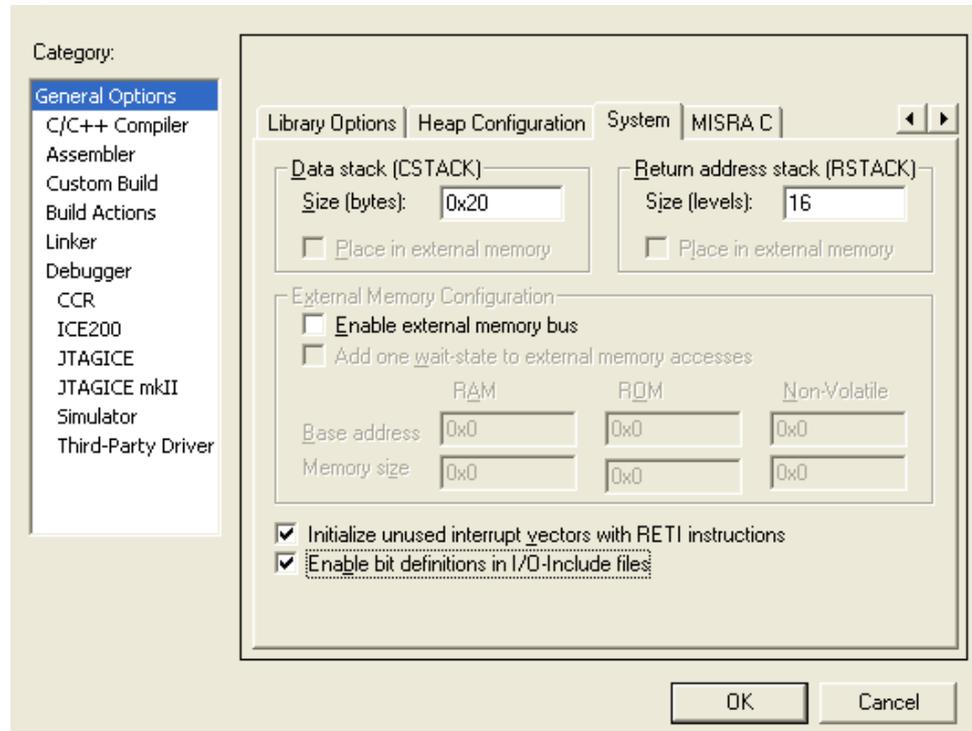
- Save the project in the desired location <project location>.
- Open project options (Project -> Options or ALT+F7)
- In the General Option Category, select the Target tab and set the Processor configuration to ATxmega128A1 as shown in the figure on the next page.

Figure 6-2: Processor configuration.



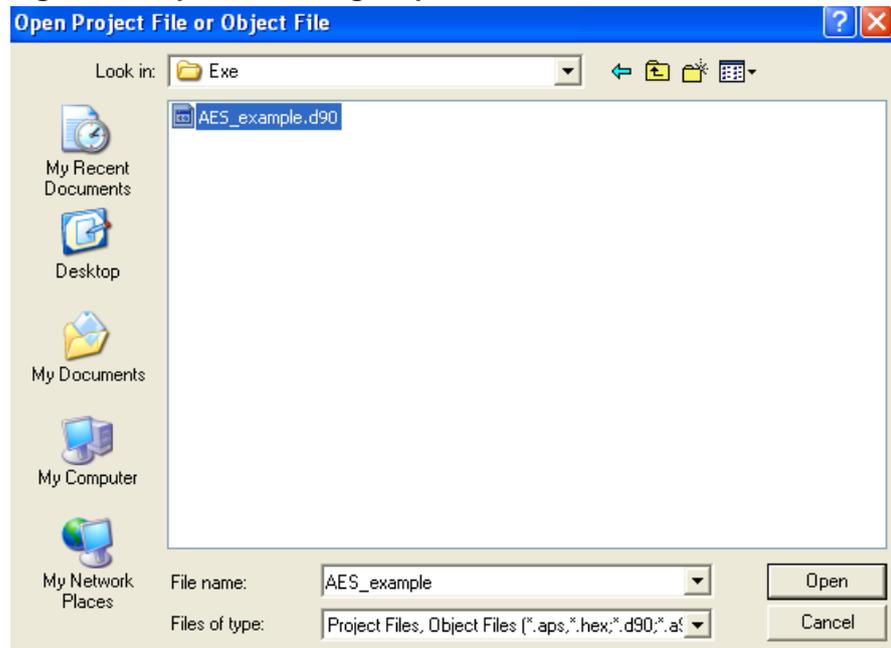
- To enable bit definitions in I/O space, open the system tab and check out the “Enable bit definitions in I/O- Include files” box.

Figure 6-3: Enable I/O bit definitions from include files.



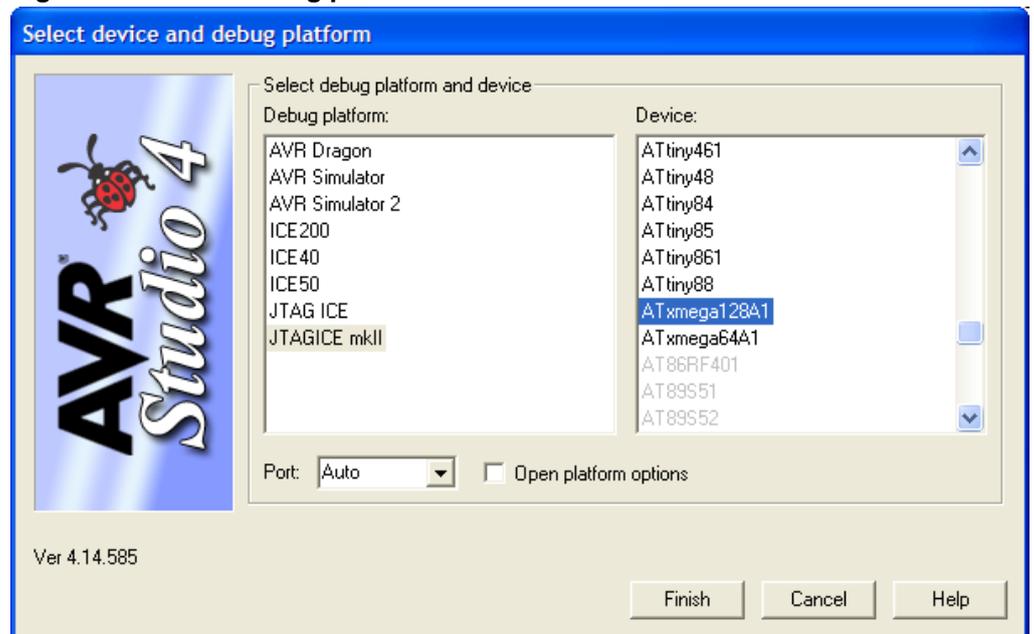
- Include ioavr.h in your files that shall use I/O bit declarations.
- Compile your code pushing the “Compile” button or selecting compile from the Project menu, or simply press “Ctrl+F7”
- Make an output file for AVR Studio by pushing “Make” button or selecting make from the Project menu, or simply press “F7”
- For debugging the program make the connections as described in chapter 5. Open AVR Studio and click on the “Open” button in the startup dialog. Go to the location where you saved your project in IAR. In the project directory, locate the Debug folder, and the “Exe” inside. The “Exe” folder contains the d90 file, which is IAR’s debug output file for AVR Studio. Open this d90 file in AVR Studio.

Figure 6-4: Open IAR debug output file in AVR Studio



- Studio will then generate an AVR Studio Project File that needs to be named and saved.
- Select JTAGICE mkII as debug platform and ATxmega128A1 as device as shown on the next page.

Figure 6-5: Select debug platform.



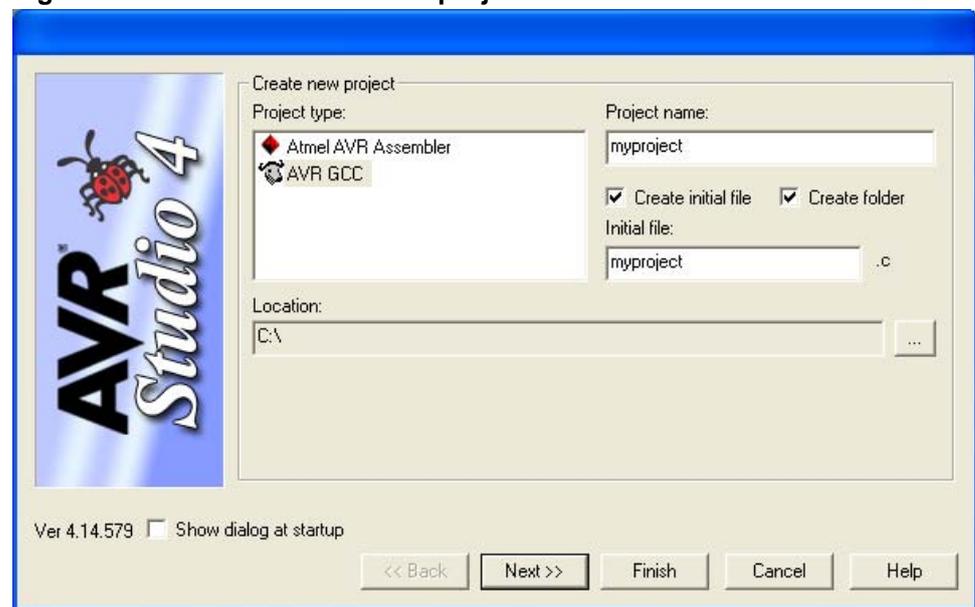
When this is done the debug session is started. More information about AVR Studio and debugging in AVR Studio are found in “AVR Studio User Guide” (Help-> AVR Studio User Guide)

6.2 Compiling and debugging with WinAVR and AVR Studio

To start a project for the ATxmega128A1 device in WinAVR and AVR Studio do as described below:

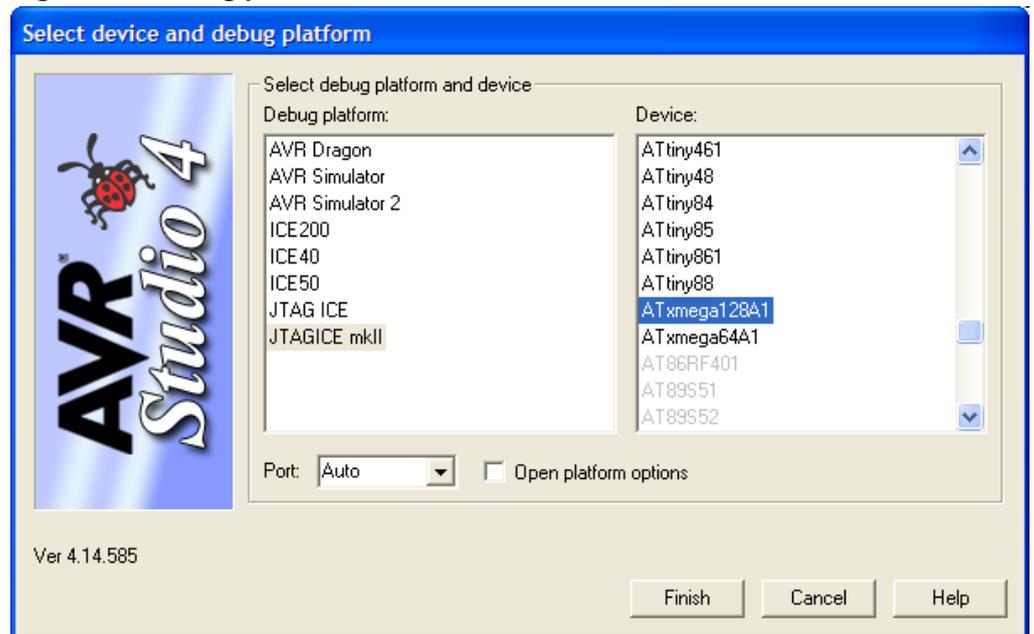
- Open AVR Studio
- Select “Project -> New Project” from the menu
- Select AVR GCC in the pop-up window and type in the project name and file as shown below:

Figure 6-6. Create new AVR GCC project



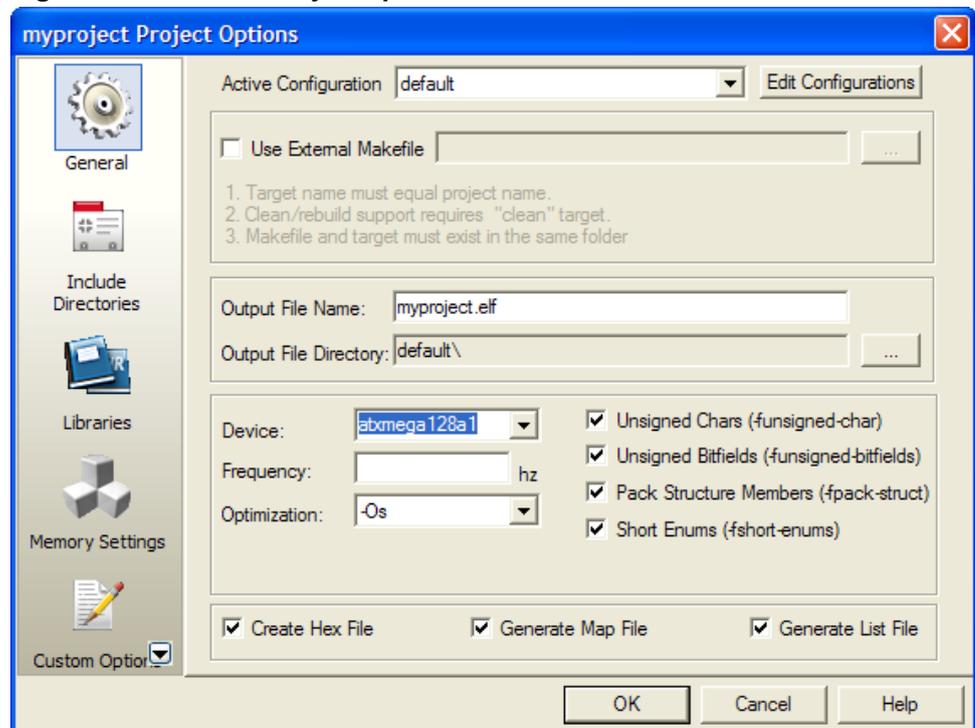
- Press “Next>>” and select “JTAGICE mkII” in the Debug platform, and “ATxmega128A1” as Device:

Figure 6-7. Debug platform and device



- Press “Finish” and the project will be created
- Select “Project -> Configuration Options” from the menu
- In the configuration window, select “atxmega128A1” in the device option and select the wanted optimization level and press the OK button.

Figure 6-8. WinAVR Project options



The project is now set up, and you can write your program in the C file. To compile the program and debug it on the device, make the same connections as described in chapter 5. Compiling your program can be done as follows:

- To compile only the active file, select “Build -> Compile” from the menu or pushing the “Compile” button, or by simply pressing “Alt+F7”.
- To make the whole project, select “Build -> Build” from the menu or by pushing the “Build” button, or by simply pressing “F7”
- The project is now build and you can debug the code on-chip by selecting “Debug -> Run” from the menu or by pushing the “Run” button, or by simply pressing “F5”.
- The two previous steps can be done simultaneously by selecting the “Build -> Build and Run” form the menu or by pushing the “Build and Run” button, or simply by pressing “Ctrl+F7”.

Your debugging session has now started and you can debug the program on-chip. For more information about debugging, read the “Help -> AVR Tools User Guide” in AVR Studio.



7 More Information

For more detailed information about the ATxmega128A1 device, read the datasheet and the XMEGA A Manual available at the Atmel® web pages:

- ATxmega128A1 datasheet:
 - http://www.atmel.com/dyn/resources/prod_documents/doc8067.pdf
- ATxmega A Manual:
 - http://www.atmel.com/dyn/resources/prod_documents/doc8077.pdf
- XMEGA Application Notes:
 - http://www.atmel.com/dyn/products/app_notes.asp?family_id=607#XMEGA%20Microcontrollers

Tools needed for development can be found here:

- AVR Studio
 - http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725
- IAR Embedded Workbench (Evaluation or Full version)
 - <http://www.iar.se/>
 - <http://supp.iar.com/Download/SW/?item=EWAVR-EVAL>
- WinAVR
 - <http://winavr.sourceforge.net/>
 - http://sourceforge.net/project/showfiles.php?group_id=68108



Headquarters

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

International

Atmel Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-
Yvelines Cedex
France
Tel: (33) 1-30-60-70-00
Fax: (33) 1-30-60-71-11

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Product Contact

Web Site
www.atmel.com

Technical Support
avr@atmel.com

Sales Contact
www.atmel.com/contacts

Literature Request
www.atmel.com/literature

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN ATMEL'S TERMS AND CONDITIONS OF SALE LOCATED ON ATMEL'S WEB SITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel's products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

© 2008 Atmel Corporation. All rights reserved. Atmel®, logo and combinations thereof, AVR®, STK®, AVR Studio® and others, are the registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Windows® and others are registered trademarks or trademarks of Microsoft Corporation in the US and/or other countries. Other terms and product names may be trademarks of others.