# AVR ONE!

# **Quick-start Guide**

.....

EVK1100 + Windows®



.....

.....

## Section 1

| Introducti | on           | 1-1   |
|------------|--------------|-------|
| 1.1        | General      | . 1-1 |
| 1.2        | Requirements | . 1-1 |

## Section 2

| Quick-sta | rt guide (short version)   | 2-1 |
|-----------|--|-----|
| 2.1       | Install Hardware and software                                    | 2-1 |
| 2.2       | Create a demonstration project                                   | 2-1 |
| 2.3       | Configure target MCU for a debug session using trace             | 2-1 |
| 2.4       | Start the debug session and configure AVR32 Studio 2.5 for trace |     |
| 2.5       | Start the trace debug session                                    |     |
|           |  |     |

## Section 3

| Hardware preparation |
|----------------------|
|----------------------|

## Section 4

| Software | Installation                                     | 4-1  |
|----------|--|------|
| 4.1      | Download the software                            | 4-1  |
| 4.2      | Download the two installation files to your disk | 4-2  |
| 4.3      | Install AVR32 GNU Toolchain                      | 4-2  |
| 4.4      | Install AVR32 Studio 2.5                         | 4-8  |
| 4.5      | Connect the AVR ONE! to power and USB host       | 4-11 |
| 4.6      | Install AVR ONE! Driver                          | 4-12 |
|          |  |      |

# Section 5

| 5.1 | Connect the AVR ONE! to the EVK1100 | -1 |
|-----|-------------------------------------|----|
| 5.2 | Connect the EVK1100 to power        | -2 |

## Section 6

| Create | demo ap | plication   | 6-1  |
|--------|---------|---|------|
| 6.1    | Start A | VR32 Studio   | 6-1  |
| 6.2    | Configu | re adapter and target                               | 6-2  |
|        | 6.2.1   | Add and configure the adapter (AVR ONE!)            | 6-3  |
|        | 6.2.2   | Configure target board and MCU                      | 6-5  |
|        | 6.2.3   | Target MCU Chip erase                               | 6-7  |
| 6.3    | Create  | a demonstration project                             | 6-8  |
| 6.4    | Configu | re AVR32 Studio for a debug session using trace     | 6-11 |
|        | 6.4.1   | Create a new debug launch configuration             | 6-12 |
|        | 6.4.2   | Configure the target trace module for program trace | 6-13 |
|        | 6.4.3   | Configure the target trace module for data trace    | 6-16 |
| 6.5    | Start a | debug session and configure the debugger for trace  | 6-17 |



i

### Table of Contents (Continued)

|     | 6.5.1     | Add start and stop trace-points       | 6-19 |
|-----|-----------|---------------------------------------|------|
| 6.6 | Start the | e trace debug session                 | 6-23 |
| 6.7 | Modify t  | he code and restart the debug session | 6-26 |

## Section 7

| Firmware | Upgrade                           | 7-1 |
|----------|-----------------------------------|-----|
| 7.1      | Firmware upgrade overview         | 7-1 |
| 7.2      | Firmware version test and upgrade | 7-1 |
| 7.3      | Adapter in use                    | 7-2 |



ii



# Section 1

# Introduction

### 1.1 General

This document contains a quick-start guide describing how to get up and running using the AVR<sup>®</sup> ONE! debugger with AVR32 Studio. In addition to the AVR ONE! debugger, you need the following items:

- AVR32 Studio 2.5 software
- AVR32 GNU Toolchain 2.4
- EVK110x Evaluation board

Software and documents can be found at www.atmel.com/avrone

### 1.2 Requirements

This example was created on a PC running Microsoft<sup>®</sup> Windows<sup>®</sup> XP Professional. For other versions of Windows, the behaviour when installing software and drivers may be slightly different.

Please read the AVR32 Studio 2.5 release notes for information about support for other versions of Windows.



# Quick-start guide (short version)

#### 2.1 Install Hardware and software

- Install the MICTOR38 connector on the EVK1100 board.
- Download and install avr32-gnu-toolchain-2.4.x and AVR32Studio-2.5.x.
- Connect AVR ONE! to power and USB and turn it on.
- Install AVR ONE! USB driver.
- Connect AVR ONE! to the EVK1100 using the MICTOR38 connector.
- Connect the EVK1100 to power and turn it on.
- Start AVR32 Studio.
- Select a suitable workspace folder to contain your projects.
- Exit from the welcome screen to workbench.
- Right-click in the *AVR32 Targets* view and select **Scan Targets**.
- Select the AVR ONE! and click on the *Properties*-tab.
- Select Details-tab. Set MCU to UC3A0512 or UC3A0512ES, depending on what MCU is mounted on your EVK1100 and Board to EVK1100,.
- Right-click on the AVR ONE! in the AVR32 Target view and select Chip Erase. This operation is only needed one time (when the EVK1100 is new).

#### 2.2 Create a demonstration project

- Select File>New>Example.
- Select *EVK1100>Components>DIP204 example*, then **Next**.
- Enter a name for the project, and click **Finish**.
- Right-click on the project in *Project Explorer* view and select **Build Project** (or use Ctrl+B).

#### 2.3 Configure target MCU for a debug session using trace

- When the build process is finished, right-click on the project in the *Project Explorer*-view and select *Debug As>Debug Configurations*.
- In the Debug Configurations-view, select AVR32 Application and click New. A new launch configuration will be created and default values will be filled into all fields.
- Select the *Trace*-tab and click **Enable Trace**.
- Select the preferred trace method. In this case we want **Buffered AUX Trace**.
- Select the preferred action when buffer is full. In this case we choose Break, read out and halt.
- Select Buffer Size. We use **16kB** for a quick test.

Select Debugger tab and tick Stop on startup at: main.

### 2.4 Start the debug session and configure AVR32 Studio 2.5 for trace

- Click the **Debug**-button. Now the program will be loaded into the target, and run until main().
- When the program halts, add at least a trace start-point (Right-click to the left of the source code line in the source code view).

### 2.5 Start the trace debug session

- Click **Resume** (green *Play* button in Debug view) and wait until the program halts.
- You can now look at the trace data in the *Trace*-view.





# **Section 3**

# Hardware preparation

In case you have an evaluation kit without the MICTOR38 connector, you need to install one. In case the connector is already mounted, you can skip this chapter.

To be able to connect to the evaluation board AUX port, you need to solder a connector to the board. The AVR ONE! Kit contains one MICTOR38 connector for this purpose. If you need more connectors for other kits, or your own designs, you can buy more connectors from Atmel, or Tyco Electronics/AMP.

The Tyco Electronics/AMP Part number is 2-5767004-2.

To install the MICTOR38 connector, you only need a fine-tipped soldering iron, a small piece of fine solder (0,3mm is OK), and some extra flux. Also remember to provide proper ventilation to prevent inhaling the fumes from the flux.

The soldering guide shows the EVK1100, but is applicable for all other kits that needs a MICTOR38 connector (like the EVK1101).

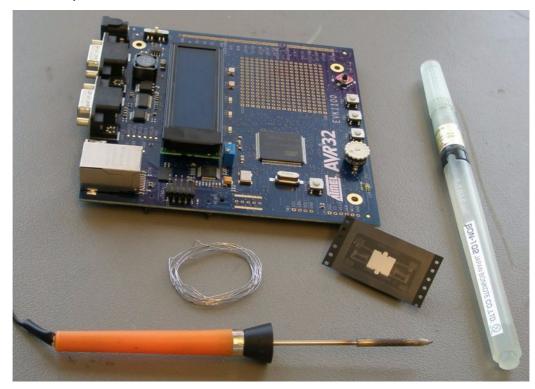


Figure 3-1. Required hardware and tools for installation of Mictor38

Unpack the Mictor38 connector and remove the pick-and-place pad

Figure 3-2. Remove the pick-and-place pad



Place the connector onto the footprint on the evaluation board. Make sure that the guide tab beneath the connector fits into the guide hole in the PCB.

Add a fair amount of flux. The extra flux is very important for a good result. It is also very important to keep the tip of the soldering iron clean while mounting the connector.

Too thick solder, too little flux or solder-blobs on the tip of the soldering iron will give bad connections or short circuits.

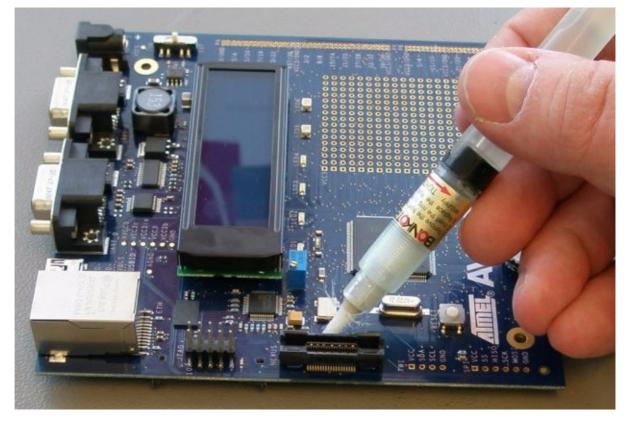
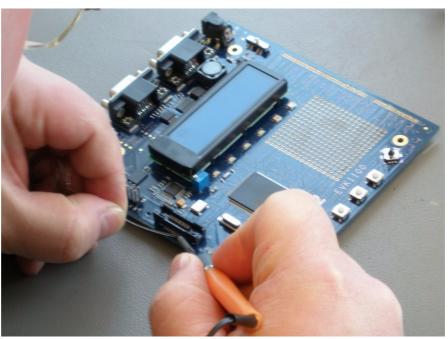


Figure 3-3. Place Mictor38 and apply flux



Make sure that the connector is firmly seated on the footprint, and start by soldering the corners.

*Figure 3-4.* Soldering the corners



When all corners are soldered, check that connector is still firmly seated. It is still possible to push the connector down and re-heat corner pins if you need to adjust a bit.

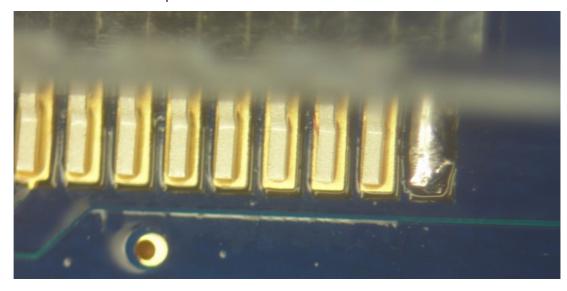
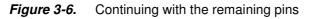
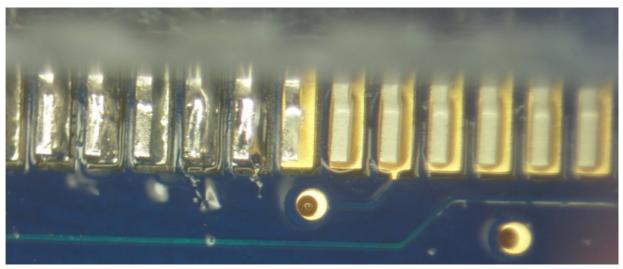


Figure 3-5. Soldered corner pin



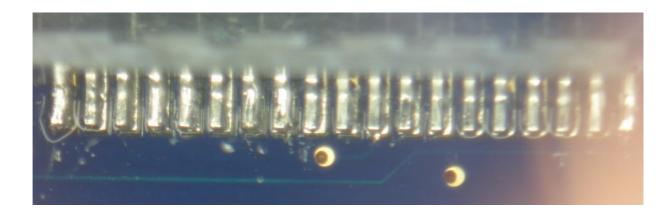
Solder the remaining pins.





After soldering, you should make sure that there are no shorts circuits between pins.







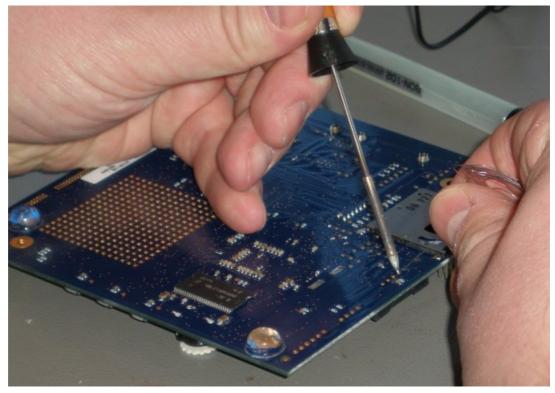
Turn the board and apply flux on the ground pins.

*Figure 3-8.* Apply flux on ground pins



Solder the five ground pins.









# Section 4

# **Software Installation**

#### 4.1 Download the software

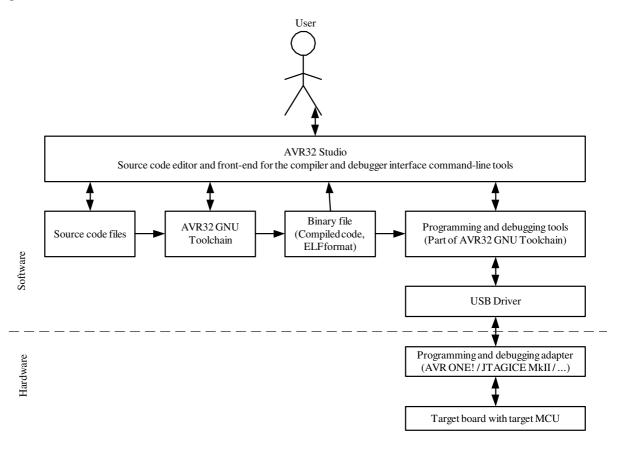
To use the AVR ONE!, you must download and install two software packages:

- avr32-gnu-toolchain-2.4.x.exe
- AVR32Studio-2.5.x.exe

The AVR32 Toolchain is a collection of tools that are required to be able to work with the AVR ONE! It contains command-line tools for controlling the AVR ONE!, and tools to compile code for the AVR32 MCUs.

AVR32 Studio is the front end that uses the AVR32 GNU Toolchain to generate binary code for the target, program the target, and control the debug sessions.

#### Figure 4-1. Tools structure



### 4.2 Download the two installation files to your disk.

The installation files can be found at this location: www.atmel.com/avrone

### 4.3 Install AVR32 GNU Toolchain

If you have any AVR tools connected to the USB hub, turn them off now. Otherwise the USB driver installation may fail.

Double-click on avr32-gnu-toolchain-2.4.x to start the installation process.

Figure 4-2. AVR32 GNU Toolchain installation welcome

| AVR32 Toolchain - InstallShi | ield Wizard  | × |
|------------------------------|--|---|
|                              | Welcome to the InstallShield Wizard for AVR32<br>Toolchain<br>Version: 2.4.2 |   |
| L. Sol and Cold of           | < Back Next > Cancel   |   |

Click Next.



#### Figure 4-3. AVR32 GNU Toolchain License Agreement form

| License Agreement<br>Please read the follow  | ving license agreement carefully   |   |  |
|--|--|---|--|
|  |  |   |  |
| with individual licens<br>individual software p<br>open source softwar<br>Board Support Pack | olchain Package contains oper<br>e agreements. You must ensure<br>ackage's licenses before using<br>e packages are distributed in so<br>age CD-ROM in the source direc<br>S PROVIDED ``AS IS'' AND AN'           | that you comply to the<br>or distributing it. All<br>urce form on the AVR32<br>ctory. | RRANTIES,                                |
| FITNESS FOR A PA<br>IN NO EVENT SHAL<br>SPECIAL, EXEMPLA<br>PROCUREMENT OF                   | OT LIMITED TO, THE IMPLIED<br>RTICULAR PURPOSE ARE EX-<br>L ATMEL BE LIABLE FOR AN'<br>RY, OR CONSEQUENTIAL DA<br>5 SUBSTITUTE GOODS OR SE<br>FORMER AND AND AND AND AND AND AND AND<br>5 SUBSTITUTE GOODS OR SE | (PRESSLY AND SPECIFICALL<br>Y DIRECT, INDIRECT, INCIDE<br>AMAGES (INCLUDING, BUT N    | Y DISCLAIMED.<br>NTAL,<br>OT LIMITED TO, |
| _  | of the license agreement<br>e terms of the license agreemen  | it  | Print                                    |

Select I accept the terms of the licence agreement, then click Next.

Figure 4-4. AVR32 GNU Toolchain installation folder select

| AVR32 Too       | olchain - InstallShield Wizard                                     | X      |
|-----------------|--|--------|
| Installati      | on folder  |        |
|                 | Install AVR32 Toolchain to:<br>C:\\Atmel\AVR Tools\AVR32 Toolchain | Change |
| InstallShield - | < <u>₿</u> ack <u>N</u> ext >                                      | Cancel |

Check that the installation folder is correct and click Next.



#### Figure 4-5. AVR32 GNU Toolchain installer configuration finished

| Ready to Install the Program                               |                           |                    |             |
|--|---------------------------|--------------------|-------------|
| The wizard is ready to begin installation.                 |                           |                    |             |
| Click Install to begin the installation.                   |                           |                    |             |
| If you want to review or change any of your in the wizard. | stallation settings, clic | k Back. Click Cano | el to exit: |
|  |                           |                    |             |
|  |                           |                    |             |
|  |                           |                    |             |
|  |                           |                    |             |
|  |                           |                    |             |
| tallShield   |                           |                    |             |
|  |                           |                    |             |

Click Install.

Figure 4-6. AVR32 GNU Toolchain installation progress indicator

| AVR32 Toolchain - InstallShield Wizard                          |        |
|---|--------|
| Setup Status  |        |
| AVR32 Toolchain is configuring your new software installation.  |        |
| Installing  |        |
| C:\\AVR32 Toolchain\avr32\lib\ldscripts\avr32elf_uc3b0256es.xwr |        |
|   |        |
|   |        |
|   |        |
|   |        |
| InstallShield   |        |
|   | Cancel |

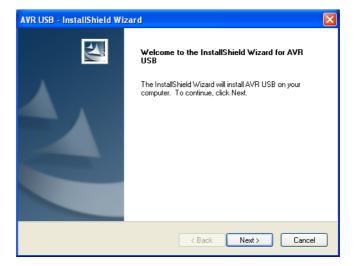
The AVR32 GNU Toolchain is now being installed. As a part of the installation process, USB drivers for all supported programming and debugging adapters are installed.



### Figure 4-7. USB Drivers installation start

| VR32 Toolchain - Ins<br>Setup Status | tallShield Wizard                        | ×      |
|--------------------------------------|--|--------|
| AVE 22 T 1 L 1 1                     | · · · · · · · · · · · · · · · · · · ·    |        |
| AVH32 Toolchain is cor               | figuring your new software installation. |        |
| Installing                           | Installing USB drivers                   |        |
|                                      |  |        |
|                                      |  |        |
|                                      |  |        |
| stallShield                          |  | Cancel |

Figure 4-8. USB Driver installer welcome



Click Next.



### Figure 4-9. USB Drivers licence agreement form

| AVR USB - InstallShield Wizard  | × |
|---|---|
| License Agreement Please read the following license agreement carefully.  |   |
| Welcome to AVR USB drivers from Atmel Corporation.<br>The tools are free of charge and may be freely copied and distributed in its original form.<br>The tools runs under Microsoft Windows 98, Microsoft Windows 2000, Microsoft Windows XP<br>, Microsoft Windows XP 64, Microsoft Windows Vista and Microsoft Windows Vista 64.<br>Copyright © ATMEL Corporation. All rights reserved.<br>AVR is trademark of ATMEL Corporation<br>Windows is a trademark of Microsoft Corporation |   |
| Print     Cancel  | ) |

Select I accept the terms of the licence agreement, then click Next.

Figure 4-10. USB drivers installer configuration finished

| AVR USB - InstallShield Wizard  |                      |
|---|----------------------|
| Ready to Install the Program<br>The wizard is ready to begin installation.                      |                      |
| Click Install to begin the installation.  |                      |
| If you want to review or change any of your installation settings, click Back. (<br>the wizard. | Click Cancel to exit |
| InstallShield   |                      |
| Kack Instru   | Cancel               |

Click Install.



### Figure 4-11. USB Drivers installation progress indicator

| Setup Status                                      |        |
|---|--------|
| The InstallShield Wizard is installing AVR USB    |        |
| Installing  |        |
| C:\Program Files\Atmel\AVR Tools\usb\windrvr6.inf |        |
|   |        |
|   |        |
|   |        |
|   |        |
| stallShield                                       |        |
|   | Cancel |

Figure 4-12. USB Drivers installation complete



Click Finish.





Figure 4-13. AVR32 GNU Toolchain installation complete

Click Finish to complete the AVR32 Toolchain installation process.

## 4.4 Install AVR32 Studio 2.5

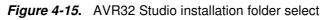
Double-click on the AVR32Studio-2.5.x.exe file to start the installation process.

Figure 4-14. AVR32 Studio 2.5 installer welcome



Click Next.





| Choose D        | dio - InstallShield Wizard<br>Iestination Location<br>Ider where setup will install files. |        |             |    |
|-----------------|--|--------|-------------|----|
|                 | Install AVR32 Studio to:<br>C:\Program Files\Atmel\AVR Tools                               |        | Change      |    |
| InstallShield – |  | < Back | Next > Canc | el |

Check that the installation folder is correct and click Next.

Figure 4-16. AVR32 Studio installer configuration finished

| AVR32 Studio - InstallShield Wizard  | × |
|--|---|
| Ready to Install the Program<br>The wizard is ready to begin installation.   |   |
| Click Install to begin the installation.   |   |
| If you want to review or change any of your installation settings, click Back. Click Cancel to exit<br>the wizard. |   |
| InstallShield —  |   |
| < Back Install Cancel  | ) |

Click Install to start the installation.



| AVR32 Studio - Install<br>Setup Status | Shield Wizard                |        |
|--|------------------------------|--------|
|  |                              |        |
| The InstallShield Wizard               | l is installing AVR32 Studio |        |
|  |                              |        |
| InstallShield                          |                              | Cancel |

Wait for the installation process to complete.

If a suitable Java<sup>™</sup> runtime is not installed, a Java installer wizard will guide you through the installation procedure.

| AVR32 Studio - InstallShield Wizard |  |  |
|-------------------------------------|--|--|
|                                     | InstallShield Wizard Complete<br>The InstallShield Wizard has successfully installed AVR32<br>Studio. Click Finish to exit the wizard. |  |
| -                                   | < Back Finish Cancel   |  |

Tick Create shortcut on desktop if you want a shortcut to be created. Then click Finish.



## 4.5 Connect the AVR ONE! to power and USB host

- Connect the AVR ONE! to power using the supplied power supply.
- Connect the AVR ONE! to the USB host (PC) using the supplied USB cable
- Turn on the AVR ONE! using the power switch next to the power connector

#### Figure 4-19. AVR ONE! connected to power and USB





### 4.6 Install AVR ONE! Driver

When the AVR ONE! is powered up and connected to the PC for the first time, the proper USB driver must be installed. Since the PC is keeping track of the serial number of each USB device, this will happen every time a new AVR ONE! is connected to the PC, even if the driver is the same as for all other AVR ONE!s that have been connected previously. This is a property of the operating system, and is not controlled by any Atmel software installed.

Figure 4-20. "New hardware" notification pop-up

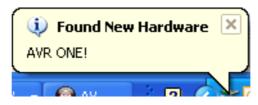


Figure 4-21. AVR ONE! Hardware installation wizard

| Found New Hardware Wizard |   |
|---------------------------|---|
|                           | Welcome to the Found New<br>Hardware Wizard<br>Windows will search for current and updated software by<br>looking on your computer, on the hardware installation CD, or on<br>the Windows Update Web site (with your permission).<br><u>Read our privacy policy</u> |
|                           | Can Windows connect to Windows Update to search for<br>software?<br>Yes, this time only<br>Yes, now and every time I connect a device<br>No, not this time<br>Click. Next to continue.  |
|                           | < Back Next > Cancel  |

When the hardware installation wizard pops up, select No, not this time and click Next.



#### Figure 4-22. Hardware installation wizard configuration



Select Install the software automatically and click Next.

Figure 4-23. Hardware installation in progress

| Found New Hard | Iware Wizard             |
|----------------|--------------------------|
| Please wait wh | nile the wizard searches |
| AVI            |                          |
|                | <u>S</u>                 |
|                |                          |
|                | < Back Next > Cancel     |

Wait for the installation process to complete.







Click Finish.

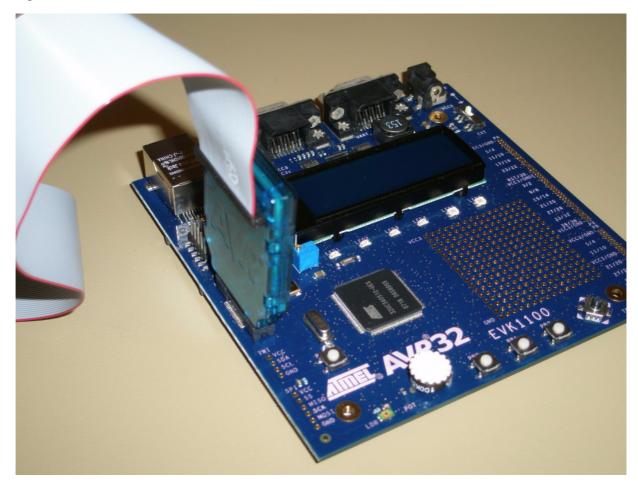




## 5.1 Connect the AVR ONE! to the EVK1100

Connect the AVR ONE! debugger to the EVK1100 evaluation board using the MICTOR38 connector.

Figure 5-1. AVR ONE! connected to the EVK1100

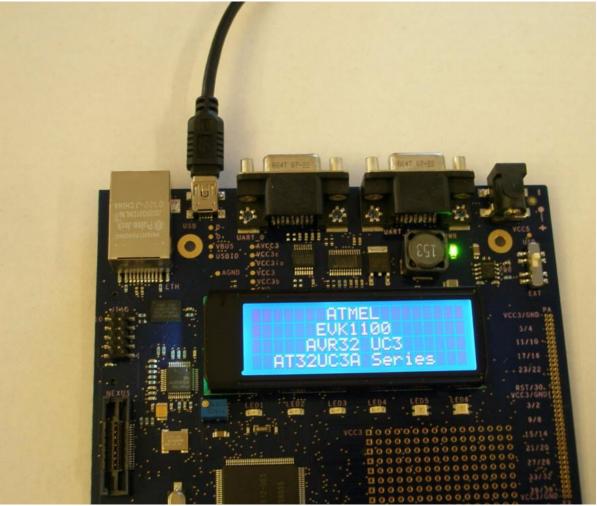


### 5.2 Connect the EVK1100 to power

Connect the EVK1100 to power and turn it on. The easiest way to provide power is to use the supplied USB cable.

Switch it on by setting the power switch to **USB**.

*Figure 5-2.* Powering the EVK1100 using the USB cable







# **Section 6**

# **Create demo application**

### 6.1 Start AVR32 Studio

Start AVR32 Studio. Start-up may take a while (because of all the Java libraries being loaded).

Figure 6-1. AVR32 Studio splash screen



#### Figure 6-2. AVR32 Studio workspace selection

| 🙆 Worksp   | ace Launcher   |           |
|------------|--|-----------|
| Select a w | vorkspace  |           |
|            | lio stores your projects in a folder called a workspace.<br>orkspace folder to use for this session. |           |
| Workspace: | C:\Avr32Studio2_work   | Browse    |
| Use this a | as the default and do not ask again  | OK Cancel |

Select a suitable workspace folder for your project files. If you want to use the same folder for your workspace every time you start AVR32 Studio, you should tick the box before clicking **OK**.

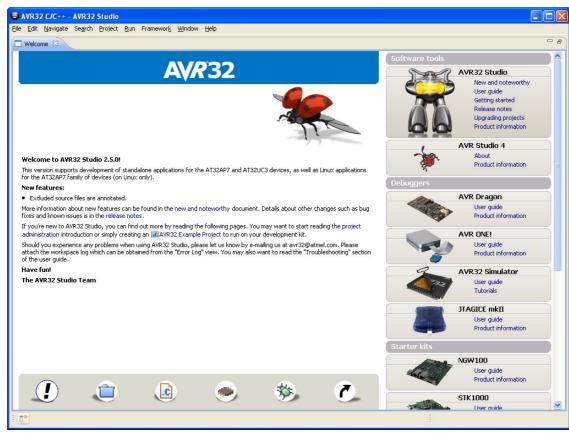


Figure 6-3. AVR32 Studio Welcome view

Exit from the welcome screen to the workbench by clicking on the Close Page icon (Arrow).

### 6.2 Configure adapter and target

Before you can use the AVR ONE! and the EVK1100, you have to tell AVR32 Studio what type of equipment is connected to your PC.

"Target" refers to the MCU on the EVK1100 evaluation board, and "Adapter" refers to the tool connecting the target to the PC (in this case, the AVR ONE!).



#### 6.2.1 Add and configure the adapter (AVR ONE!)

#### Figure 6-4. Scan Targets



Right-click in the AVR32 Target-view and select Scan Targets.

#### Figure 6-5. Available targets

| 🥔 A      | VR32 Targets 🛛  |                 | ê – D   |
|----------|-----------------|-----------------|---------|
|          | Name 💌          | Adapter         | Board   |
|          | AVR ONE!        | AVR ONE!        |         |
| <b>*</b> | AVR32 Simulator | AVR32 Simulator | AVR32 S |
|          |                 |                 |         |
|          |                 |                 |         |
|          |                 |                 |         |
| <        |                 |                 | >       |

Select the AVR ONE!

Figure 6-6. AVR ONE! Selected

| 🥔 A | VR32 Targets 🕺   |                 | 🗳 🗖 🗖   |
|-----|------------------|-----------------|---------|
|     | Name 🔻           | Adapter         | Board   |
|     | AVR ONE!         | AVR ONE!        |         |
| ۲   | AVIS32 Simulator | AVR32 Simulator | AVR32 S |
|     |                  |                 |         |
|     |                  |                 |         |
|     |                  |                 |         |
| <   |                  |                 | >       |



| errors, u warnings, u inros 🔗 🔍   |          |      |          |  |
|---|----------|------|----------|--|
| Problems 🛛 🔲 Properties<br>errors, 0 warnings, 0 infos<br>Description 🔺 | Resource | Path | Location |  |
|   |          |      |          |  |
|   |          |      |          |  |
|   |          |      |          |  |
|   |          |      |          |  |
|   |          |      |          |  |
|   |          |      |          |  |
|   |          |      |          |  |

#### Figure 6-7. Selecting the properties view

Click on the **Properties** tab.

#### Figure 6-8. Properties view

| 🖹 Problems 🔳               | Properties 🛛                   | E Console  | ₹ ~ - 0 |
|----------------------------|--------------------------------|--|---------|
| AVR ONE                    |                                |  |         |
| General<br>Details         | Name:                          | AVR. ONEI  |         |
| Daisy Chain<br>Information | ▼ Binaries<br>AVR32 Studio kee | ps track of the last file used to program a target. The name and date is show below. |         |
| Information                | Binary path:                   |  |         |
|                            | Binary date:                   | Thu Jan 01 01:00:00 CET 1970   |         |
|                            |                                |  |         |
|                            |                                |  |         |
|                            |                                |  |         |

If you have several adapters connected at the same time, this is the place where you can give them unique names. Just type the name you want to use in the **Name** field.



#### 6.2.2 Configure target board and MCU

Select the Details tab.

| Figure 6-9. Setting the board and device ty |
|---|
|---|

| General                    | Details           |     |        |      |       |      |       |      |    |           |     |   |   |     |    |    |    |    |          |              |        |        |   |   |         |          |   |   |     |       |          |       |    |      |    |
|----------------------------|-------------------|-----|--------|------|-------|------|-------|------|----|-----------|-----|---|---|-----|----|----|----|----|----------|--------------|--------|--------|---|---|---------|----------|---|---|-----|-------|----------|-------|----|------|----|
| Details                    | Debugger/programm | er: | A١     | /R · | ONE   | 1    |       |      |    |           |     |   |   |     |    |    |    |    |          |              |        |        |   |   |         |          |   | _ |     | ~     | Dev      | vice: | U  | JC3/ | AC |
| Daisy Chain<br>Information | Clock source:     |     |        | ter  | nal f | RC d | oscil | lato | r  |           |     |   |   |     |    |    |    |    |          |              |        | _      |   |   |         |          |   |   |     | ~     | Boa      | ard:  | E  | VK1  | 10 |
|                            | ▼ Connection      |     |        |      |       |      |       |      |    |           |     |   |   |     |    |    |    |    |          |              |        |        |   |   |         |          |   |   |     |       |          |       |    |      |    |
|                            | Serial number:    | 00  | 000    | 000  | 000   | 15   |       |      |    |           |     |   |   |     |    |    |    |    |          |              |        |        |   |   |         |          |   | _ | _   | _     |          |       |    | _    |    |
|                            | Connection:       | us  | )      |      |       |      |       |      |    |           |     |   |   |     |    |    |    |    |          |              |        |        |   |   |         |          |   | _ | _   | _     |          |       |    | _    |    |
|                            | COM Port:         |     |        |      |       |      |       |      |    |           |     |   |   |     |    |    |    |    |          |              |        |        |   |   |         |          |   | _ | _   | _     |          |       |    | _    |    |
|                            | ▼ Clock           |     |        |      |       |      |       |      |    |           |     |   |   |     |    |    |    |    |          |              |        |        |   |   |         |          |   |   |     |       |          |       |    |      |    |
|                            | JTAG Clock        | 1   | 1      | 1    | 'n    | 1    | C.    | e    | T. | 1         | i.  |   |   | 0   | C. | C. | r. | I. | ı.       | 1            | 9      | 0      | c | 1 | ī       | 1        | 1 | 1 |     |       | 81       | 12    | C  | 11   |    |
|                            | STAG CIOCK        | ¢   | 1      | 1    | ٣     | 1    | •     | 1    | 1  | 1         | 1   | 1 | 1 | 2   | Ċ, | r. | 1  | i. | 1        | 1            | 1      | 2      | ć | ľ | •       | 1        | 1 | 1 |     | 1     | 2        | ¢.    | i. | •    |    |
|                            | aWire Clock       | 1   | 9<br>9 | -    | - 63  | 12   | 1     | 1    | 37 | 25<br>100 | - 1 | 1 | 1 | - 1 | 25 | 1  |    |    | ् ः<br>ः | ( 38<br>( 93 | )<br>/ | 8<br>3 | 1 | 1 | 9<br>20 | 97<br>20 | Ó | 1 | i i | а<br> | 34<br>27 | 0     | 12 | T.   | _  |

Set MCU to UC3A0512 or UC3A0512ES, depending on what MCU is installed on your EVK1100.

Figure 6-10. MCU Markings



To check which type of MCU is mounted on your EVK1100 evaluation board, you can read the part number printed on the MCU. The picture shows the part number printed on an -ES part (-UES suffix).

Set Board to EVK1100.



#### Set MCU Clock source to Crystal.

Adjust the JTAG Clock to a suitable value (Usually 33MHz or less. Max speed depends on target board signal quality). Click **Apply**.

The target and adapter configuration process is now complete.



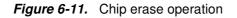
#### 6.2.3 Target MCU Chip erase

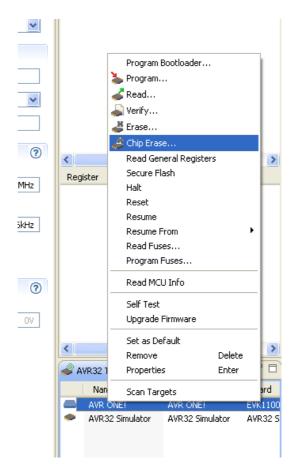
If the EVK1100 evaluation board is brand new, or if it still contains the original demo application (Control Panel Demo), the FLASH lock-bits need to be cleared. Right-click on the AVR ONE! In the *AVR32 Target* view and select **Chip Erase**.

WARNING! This process will erase the original demo application programmed at the factory. After this operation the EVK1100 evaluation board will be completely empty. If you need to keep the original application, you should not perform this operation.

If you would like to use your EVK1100 for this example, it is not difficult to restore the original "Control Panel Demo application". All you have to do is to build the "Control Panel Demo example" enclosed with AVR32 Studio.

You should now perform the Chip Erase operation.







## 6.3 Create a demonstration project

| Figure 6-12. | Create new project |
|--------------|--------------------|
|--------------|--------------------|

| File Edit Navigate | Search Project Run | Framework Window Help     |                   |
|--------------------|--------------------|---------------------------|-------------------|
| New                | Alt+Shift+N        | 📸 AVR32 C Project         | - (               |
| Open File          |                    | 📸 AVR32 C++ Project       |                   |
| Close              | Ctrl+W             | MVR32 C Project From Temp | late              |
| Close All          | Ctrl+Shift+W       | AVR32 Example Project     |                   |
| Save .             | Ctrl+S             | Project                   |                   |
| 🔚 Save As          |                    | 😂 Source Folder           |                   |
| 🦷 Save All         | Ctrl+Shift+S       | 😂 Folder                  |                   |
| Revert             |                    | C Source File             |                   |
| Move               |                    | h Header File             |                   |
| Rename             | F2                 | File                      |                   |
| 🔊 Refresh          | F5                 | E Fuse Settings           |                   |
| Convert Line Deli  | miters To          | 🕂 📑 Other                 | Ctrl+N            |
| 👜 Print            | Ctrl+P             |                           |                   |
| Switch Workspac    | e 1                | oblems 🔲 Properties 🛛     | 🖳 Console         |
| Restart            |                    | AVR ONE!                  |                   |
| import             |                    | - Details                 |                   |
| Export             |                    |                           | AVR ONE!          |
|                    |                    | Chain                     | nmer:             |
| > Launch AVR Stud  | io 4               | Clock source:             | Crystal connected |
| Properties         | Alt+Enter          |                           |                   |
|                    |                    | Connection                |                   |

Create a new project by clicking File>New>AVR32 Example Project.

#### Figure 6-13. Select project example

| 🙆 New E               | xample   |        |
|-----------------------|--|--------|
| Select a<br>Creates a | wizard<br>DIP204 example for the EVK1100.  |        |
| Wizards:              |  |        |
| type filter           | text   |        |
|                       | C3AES Software Framework<br>EVK1100 - APPLICATIONS - Control Panel Demo example<br>EVK1100 - COMPONENTS - Data Flash Memory example<br>EVK1100 - COMPONENTS - DJ204 example<br>EVK1100 - COMPONENTS - SD/MMC Card example<br>EVK1100 - COMPONENTS - SD/MMC Card example<br>EVK1100 - DRIVERS - Analog-to-Digital Converter (ADC) example<br>EVK1100 - DRIVERS - CPU Cycle counter example<br>EVK1100 - DRIVERS - CPU Cycle counter example<br>EVK1100 - DRIVERS - External Interrupt Controller (EIC) example 1<br>EVK1100 - DRIVERS - External Interrupt Controller (EIC) example 2<br>EVK1100 - DRIVERS - External Interrupt Controller (EIC) example 3<br>EVK1100 - DRIVERS - Flash controller (CLS) example 3<br>EVK1100 - DRIVERS - Flash controller (IDS) example<br>EVK1100 - DRIVERS - Interrupt Controller (INTC) example |        |
| 0                     | < Back Next > Finish   | Cancel |

Select EVK1100 - Components - DIP204 example, then Next



### Figure 6-14. New project name

| New Example Project  |        |
|--|--------|
| EVK1100 - COMPONENTS - DIP204 example<br>Creates a DIP204 example for the EVK1100. | 5      |
| Project name: Demo_1   |        |
| Location: C:/Avr32Studio2_work/Demo_1 Choose file system: default                  | Browse |
|  | Cancel |

Enter a name for the project, and click **Finish**.



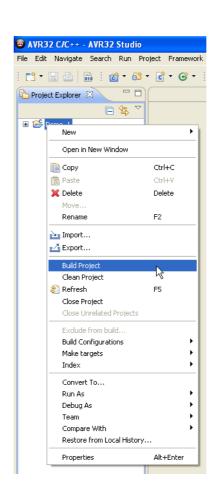


Figure 6-15. Build project

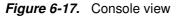
Right-click on the project in **Project Explorer**-view and select **Build Project** (or press CTRL+B).

Figure 6-16. Project build progress

| Build Project    |                                     |
|------------------|-------------------------------------|
| Building project |                                     |
|                  | )                                   |
|                  |                                     |
|                  | Run in Background Cancel Details >> |

Wait for the project build process to finish.







The console shows output from the compiler. Make sure that this ends with a "Build complete ..." message (Except for the "Time consumed" message). If something is not working, you will see error messages in this view.

# 6.4 Configure AVR32 Studio for a debug session using trace

| AVR3        | 2 C/C++ - AVR32 Studio    |                    |            |              |                      |              |
|-------------|---------------------------|--------------------|------------|--------------|----------------------|--------------|
|             | Navigate Search Project   | Run Framework Wi   | ndow       | Help         |                      |              |
| <b>[]</b> - | 🖫 📤   🗟 i ╆ i 💰           | • 🚳 • 🖻 • 🞯        | -          | <b>% -</b> 🛞 |                      | - 🜔 - 💁 -    |
| -<br>Projec | tt Explorer 🛛 🗖 🗖         |                    |            |              |                      |              |
|             | □ 🔄 😜 🏹                   |                    |            |              |                      |              |
| Ð 😂 🖡       |                           |                    |            |              |                      |              |
|             | New                       |                    | •          |              |                      |              |
|             | Go Into                   |                    |            |              |                      |              |
|             | Open in New Window        |                    |            |              |                      |              |
|             | Copy                      | Ctrl+C             |            |              |                      |              |
|             | 💼 Paste                   | Ctrl+V             |            |              |                      |              |
|             | 💢 Delete                  | Delete             |            |              |                      |              |
|             | 🕭 Remove from Context     | Ctrl+Alt+Shift+Dov | /n         |              |                      |              |
|             | Move                      |                    |            |              |                      |              |
|             | Rename                    | F2                 |            |              |                      |              |
|             | 🚵 Import                  |                    |            |              |                      |              |
|             | 🛃 Export                  |                    |            |              |                      |              |
|             | Build Project             |                    |            |              |                      |              |
|             | Clean Project             |                    |            |              |                      |              |
|             | 8 Refresh                 | F5                 |            |              |                      |              |
|             | Close Project             |                    |            |              |                      |              |
|             | Close Unrelated Projects  |                    |            |              |                      |              |
|             | Exclude from build        |                    |            |              |                      |              |
|             | Build Configurations      |                    | •          |              |                      |              |
|             | Make Targets              |                    | •          |              |                      |              |
|             | Index                     |                    | •          |              |                      |              |
|             | Show in Remote Systems vi | iew                |            |              |                      |              |
|             | Convert To                |                    |            |              |                      |              |
|             | Run As                    |                    | <u> </u>   |              |                      |              |
|             | Debug As                  |                    | Þ          | 🥔 1 AVR      | 32 Application       | Alt+Shift+D, |
|             | Profile As                |                    | •          | 💽 2 Loca     | al C/C++ Application | r            |
|             | Team                      |                    |            | Dobus        | g Configurations     |              |
|             | Compare With              |                    | - <b>b</b> | Debug        | y coningurations     |              |

Figure 6-18. Open Debug Dialog

When the build process is finished, right-click on the project in the *Project Explorer* view and select *Debug As>Debug Configurations*.



## 6.4.1 Create a new debug launch configuration

In the *Debug Configurations* view, select **AVR32 Application** and right click and select **New**. A new launch configuration will be created and default values will be filled into all applicable fields.

Select the *Debugger* tab and tick the **Stop on startup at: main** option.

Figure 6-19. Debugger tab

| Debug Configurations   |   |          | X            |
|--|---|----------|--------------|
| Create, manage, and run cor  | nfigurations  |          | Ť.           |
| AVR32 Application      Organization      C/C++ Application      C/C++ Application      C/C++ Remote Applicatic      C/C++ Remote Application      GOB Hardware Debugging      Launch Group | Name:       Demo_1         Main       Debugger       Irac         GDB proxy:       If Start GDB proxy         GDB proxy command:       GDB proxy host:         GDB proxy host:       GDB proxy host:         GDB proxy trace port:       GDB proxy trace port:         If Stop on startup at:       If Verbose mode | e Common | Apply Revert |
| Filter matched 8 of 8 items  |   |          |              |
| ?  |   |          | Debug Close  |



# 6.4.2 Configure the target trace module for program trace

Figure 6-20. Debug configurations, Trace tab

| Debug Configurations  |   | X  |
|---|---|--|
| Create, manage, and run con   | figurations   | TO-  |
| Vpe filter text         AVR32 Application         C/C++ Application         C/C++ Application         C/C++ Postmortem Debug;         C/C++ Remote Application         E C/C++ Remote Application         E C/C++ Remote Application         E C/C++ Remote Application         E GDB Hardware Debugging         Launch Group | Name:       Demo_1         Main       Debugger       Irace         © General:       © Common         © Epable trace       Enable ownership trace         Trace method:       NanoTrace         Data trace options:       O         Data trace range 1       Data trace range 2         Memory access type:       access         Lower boundary:       Address:         Address:       0x0         Wolffer full actions:       Image: Continue running         Buffer full actions:       Image: Continue running         Break, read out and continue       Wrap buffer | AUX trace options:<br>Buffer size:<br>Override auxiliary port selection<br>Auxiliary port pin configuration:<br>WanoTrace options:<br>Break on application buffer access<br>Use NANOTRACE variable<br>Specify size and location<br>Buffer start:<br>OXO<br>Detect<br>Size:<br>32 bytes |
| Filter matched 8 of 8 items   |   | Apply Revert   |
| ?   |   | Debug Close  |

Select the Trace tab and click Enable Trace.



| Figure 6-21. | Preferred Trace method |
|--------------|------------------------|
|--------------|------------------------|

| Debug Configurations   |   |               |               |
|--|---|---------------|---------------|
| Create, manage, and run conf   | īgurations  |               | 10 to         |
| Ype filter text         Pemo_1         C/C++ Application         C/C++ Application         C/C++ Notmortem Debug;         C/C++ Remote Application         GDB Hardware Debugging         Launch Group | Name:       Demo_1         Main       Debugger       Irace         General:       Image: Common intervent i | Size: 32 byta | 0 V<br>access |
| Filter matched 8 of 8 items  |   | Apply         | Revert        |
| 0  |   | Debug         | Close         |

Select the preferred trace method. In this case we want **Buffered AUX Trace**.



| Debug Configurations   |   | X  |
|--|---|--|
| Create, manage, and run con  | nigurations   | Ť.   |
| Vype filter text<br>Vype filter text<br>AVR32 Application<br>C (/C++ Application<br>C (/C++ Application<br>C (/C++ Postmortem Debug;<br>C (/C++ Remote Application<br>C GDB Hardware Debugging<br>Launch Group | Trace mgthod:     Buffered AUX Trace       Data trace options:     Auxiliary port       Data trace range 1     Data trace range 2 | 16kB       uxiliary port selection       ain configuration:       0       tions:       oplication buffer access       rRACE variable |
| Filter matched 8 of 8 items  |   | Apply Reyert   |
| ?  |   | Debug Close  |

Select Buffer Size. We select 16kB for a quick test.

*Figure 6-23.* Buffer full action



Selected the preferred action when buffer is full. In this case we choose Break, read out and halt.



## 6.4.3 Configure the target trace module for data trace

We would like to keep an eye on one of our variables. To do this, we configure a data trace range. In our case, we want a trace message each time the program writes to a variable called display.

Figure 6-24. Memory access type

| Data trace options:      |                   |
|--------------------------|-------------------|
| Data trace range 1 D     | ata trace range 2 |
| Memory access type:      | access            |
| Lower boundary:          | access            |
| Address or file location | write             |
| Upper boundary:          |                   |
| Address or file location | on:0x0            |
|                          |                   |

Set Memory access type to write.

#### Figure 6-25. Data trace boundaries

| Data trace options:                   |
|---------------------------------------|
| Data trace range 1 Data trace range 2 |
| Memory access type: write             |
| Address or file location: 0x0         |
| Address or file location: 0x0         |

Select memory location for lower and upper boundaries.

| Figure 6-26. | Variable address selection dialogue |
|--------------|-------------------------------------|
|--------------|-------------------------------------|

| Address s       | election   |                            |
|-----------------|------------|----------------------------|
| Please select a | n address. |                            |
| Address         | Length     | Name 🔼                     |
| 0xc0            | 4          | _int_line_handler_table_5  |
| 0xc4            | 4          | _int_line_handler_table_6  |
| 0xc8            | 4          | _int_line_handler_table_7  |
| Oxcc            | 4          | _int_line_handler_table_8  |
| 0xd0            | 4          | _int_line_handler_table_9  |
| 0xd4            | 4          | _int_line_handler_table_10 |
| 0xd8            | 4          | _int_line_handler_table_11 |
| 0×dc            | 4          | _int_line_handler_table_12 |
| 0xe0            | 4          | _int_line_handler_table_13 |
| 0xe4            | 12         | _int_line_handler_table_14 |
| 0×f0            | 4          | _int_line_handler_table_15 |
| 0xf4            | 4          | _int_line_handler_table_16 |
| 0xf8            | 4          | _int_line_handler_table_17 |
| 0xfc            | 4          | _int_line_handler_table_18 |
| 0×100<br>0×104  | 4<br>2     | _int_line_handler_table_19 |
| 0×104<br>0×108  | 2<br>32    | pwm_duty                   |
| 0×108           | 2          | pwm_channel                |
| 0x128           | ۷.         | uispiay 🕑                  |
| ?               |            | OK Cancel                  |

Select the start and stop addresses for the data range. Use the Address selection dialogue, or type the addresses.



# Figure 6-27. Configured data trace range

| Data trace options:                   |       |
|---------------------------------------|-------|
| Data trace range 1 Data trace range 2 |       |
| Memory access type: write             | ✓     |
| Address or file location:             | 0x9a4 |
| Upper boundary:                       | 0x9a4 |

# 6.5 Start a debug session and configure the debugger for trace

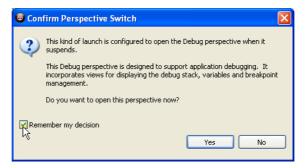
| Figure 6-28. Startin | g a | debug | session |
|----------------------|-----|-------|---------|
|----------------------|-----|-------|---------|

| Image: Second | Create, manage, and run conf   | gurations   |
|---|--|---|
|   | type filter text  type filter text  def value of the second secon | Main       Debugger       Trace       Common         General: |
| Filter matched 8 of 8 items   |  | Apply Reyert  |

Click the **Debug** button. Now the program will be loaded into the target, and run until main().



## Figure 6-29. Switching perspective

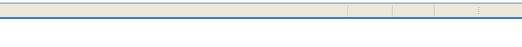


When the debug session starts, AVR32 Studio 2.5 will change to the *Debug* perspective (desktop layout designed for use during debug sessions). You should click **Yes**. To avoid being asked every time you start a debug session, you should also click the **Remember my decision** box before answering **Yes**.

Wait until the target has stopped at the first instruction in the main() routine.

Figure 6-30. Program halted at main()

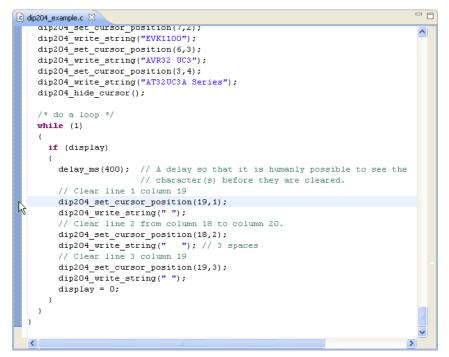
| - Demo_1/src/dip204_example.c - AVR32 Studio  |  | -  |
|---|--|--|
| Refactor Navigate Search Run Project Framework Window Help                              |  |  |
| 』 酉│ 🔜 : ● ▼ 林 ▼ 🔘 ▼ 💁 ▼ : 🖋 : ½ × 徑 - 🤝 - ↔ -  |  | 🖹 🏇 Debug  |
| x 🛛 🗱 🖗 🕨 🖩 🗱 🕺 🗞 🤿 🖄   | 🛤 Variables 🕴 💁 Breakpoints 🕮 AVR32 Regist 🙀   | Expressions 📣 Trace Data 🚻 Registers 🛋 Module  |
| mo_1 [AVR32 Application]  |  | 🦾 📲 📄 💕 💥  |
| vr32gdbserver debugger (11.03.08 16:27) (Suspended)                                     | Name   | Value  |
| High Thread [0] (Suspended)<br>= 1 main() dip204 example.c:318 0x80002470               | 🗄 🥭 spiOptions   | {}   |
| avr32qdbproxy   | E 🥭 DIP204_SPI_GPIO_MAP  | 0x8000b218   |
| avr32-qdb (11.03.08 16:27)  |  |  |
| C:\Avr32Studio2_work\Demo_1\Debug\DIP204_EXAMPLE.elf (11.03.08 16:27)                   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   |  |  |
|   | and a second sec |  |
|   |  |  |
| _example.c 🛛  |  | 🗄 Outline 🛛 🛛 🖓 🙀 💉 🔍  |
| brief main function : do init and loop (poll if configured so)                          |  | poard.h  |
|   |  | 🔜 compiler.h   |
| main(void)  |  | dip204.h   |
|   |  | 🛀 intc.h   |
| atic const gpio_map_t DIP204_SPI_GPIO_MAP =   |  | gpio.h   |
| (DIP204 SPI SCK PIN, DIP204 SPI SCK FUNCTION ), // SPI Clock.                           |  | pm.n<br>cycle counter.h  |
| (DIP204 SPI MISO PIN, DIP204 SPI MISO FUNCTION), // MISO.                               |  | spi.h  |
| (DIP204 SPI MOSI PIN, DIP204 SPI MOSI FUNCTION), // MOSI.                               |  | avr32/io.h   |
| (DIP204 SPI NPCS PIN, DIP204 SPI NPCS FUNCTION) // Chip Select NPCS.                    |  | # GPIO_CHARSET   |
|   |  | # GPIO_BACKLIGHT_MINUS   |
|   |  | # GPIO_BACKLIGHT_PLUS  |
| Switch the CPU main clock to oscillator O   |  | display : unsigned short   |
| _switch_to_oscO(&AVR32_PM, FOSCO, OSCO_STARTUP);  |  | TimeOut : unsigned char  |
| Disable all interrupts.   |  | current_char : unsigned short           •         •           •         *           •         *           •         *    |
| sable global interrupt();   |  | <ul> <li>S dip204_example_PB_int_handler(void) : void</li> </ul>   |
| page Tappar Tuper Mo(),   |  | <ul> <li>S dip201_example_10_int_intitide(void) : void</li> <li>S dip204_example_Joy_int_handler(void) : void</li> </ul> |
|   | >  |  |
|   | 2  |  |
| e 🕴 🖉 Tasks 🔝 Problems 🚺 Memory   |  | 📕 🗶 🔆 🔒 🚮 🗬 🖉 🖻 🕇 📑 📬  |
| VR32 Application] C:\Avr32Studio2_work\Demo_1\Debug\DIP204_EXAMPLE.elf (11.03.08 16:27) |  |  |





#### 6.5.1 Add start and stop trace-points

Figure 6-31. Source code editor



Scroll down to and select line 356 in the file DIP204\_Example.c and then select *Run>Toggle Trace Point*.



## Figure 6-32. Tracepoint (Start)

| Properties for   |  |  |
|------------------|--|--|
| type filter text | Tracepoint   | ⇔ + ⇔ +  |
| Tracepoint       | <ul> <li>Tracepoint is enabled</li> <li>Trigger event:</li> <li>Program counter</li> <li>Data read</li> <li>Data write</li> <li>Data access</li> </ul> | Trigger location:         Location         Range       exclusive         Trigger value compare:         No value comparison         Mask bytes:         Value equals:         Disregard size   |
|                  | Trigger start location:  | Trace operation:   Emit trace message  Start trace  trace tr |
|                  | Address or file location: C:\A   | wr32Studio2_work\Demo_1\src\dip204_example.c,397   |
| 0                |  | OK Cancel  |

Set Tracepoint Configuration values:

- Set Trigger Event to Program Counter
- Set Trace Operation to Start Trace
- Set Tracepoint type to both Program trace and Data trace
- Click OK

This will create a tracepoint that starts both program and data trace when the program counter hits this code line.

Scroll down to and select line 364 in the file DIP204\_Example.c and then select Run>Toggle Tracepoint.



# Figure 6-33. Tracepoint (Stop)

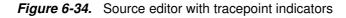
| Properties for   |  |   |
|------------------|--|---|
| type filter text | Tracepoint   | ↔ ↔ ↔   |
| Tracepoint       | <ul> <li>Tracepoint is enabled</li> <li>Trigger event:</li> <li>Program counter</li> <li>Data read</li> <li>Data write</li> <li>Data access</li> </ul> | Trigger location:            • Location          Range         exclusive         Trigger value compare:         • No value comparison         Mask bytes:         • Value equals: |
|                  | Tracepoint type:<br>Program trace<br>Data trace<br>Boundaries<br>Trigger start location:<br>Address or file location:                                  | Trace operation:   Trace message  Start trace  Stop trace  C:\Avr32Studio2_work\Demo_1\src\dip204_example.c,406   |
|                  | Trigger end location:<br>Address or file location:   | Restore Defaults Apply  |
| 0                |  | OK Cancel   |

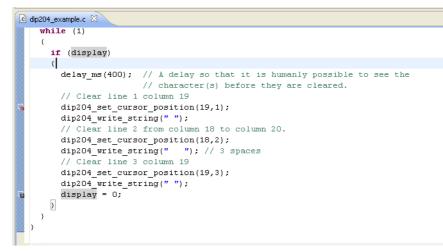
Set Tracepoint Configuration values:

- Set Trigger Event to Program Counter
- Set Trace Operation to Stop Trace
- Set Tracepoint type to both Program trace and Data trace
- Click OK

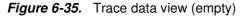
This will create a tracepoint that stops both program and data trace when the program counter hits this code line.







The source editor now has two tracepoint indicators next to the respective code lines.



| (X)= Variables | 💁 Breakpoints | 🚇 AVR32 Regist | ିଙ୍କୁ Expressio | ons | 🤳 Trace 🛛 | Data S | 3  | lili Re | gisters | 🛋 м   | lodules |   |
|----------------|---------------|----------------|-----------------|-----|-----------|--------|----|---------|---------|-------|---------|---|
| Demo_1         |               |                |                 |     |           | 43     | 2  | 8-      | ൻ 💰     | •   @ | 2 🚚     | - |
| Frame a        | #             | Address        | Frame desc      | /*  | Select    | one    | or | more    | prog    | ram   | trace   | f |
|                |               |                |                 |     |           |        |    |         |         |       |         |   |
|                |               |                |                 |     |           |        |    |         |         |       |         |   |
|                |               |                |                 |     |           |        |    |         |         |       |         |   |
|                |               |                |                 |     |           |        |    |         |         |       |         |   |
|                |               |                |                 |     |           |        |    |         |         |       |         |   |
|                |               |                |                 |     |           |        |    |         |         |       |         |   |
|                |               |                |                 |     |           |        |    |         |         |       |         |   |
| <              |               |                | >               |     |           |        |    |         |         |       |         | V |
|                |               |                |                 | <   |           |        |    |         |         |       |         | > |

Click on the Trace Data tab to bring the trace data view to the front.



# 6.6 Start the trace debug session

| Figure 6-36. | Resume debug session |
|--------------|----------------------|
|--------------|----------------------|

| 🏇 Debug 🛛  | 🎽 🏟 💽 🗉  | 🔳 🖓 🖪         | 👁 . e 🖷 | i> 🛒 🎽 | - 0 |
|--|----------|---------------|---------|--------|-----|
| & Demo_1 [AVR32 Application]     & avr32gdbserver debugger (11.03.08 16:27)     & Thread [0] (Suspended)     & I main() dip204_example.c:318 0xt     avr32gdbproxy     avr32gdbproxy     & C:\Avr32Studio2_work\Demo_1\Debug\DIF | 80002470 | .03.08 16:27) |         |        |     |

Make sure that the main() process is still selected in the *Debug* view before pressing the **Resume** button.

Figure 6-37. LCD Display showing original message

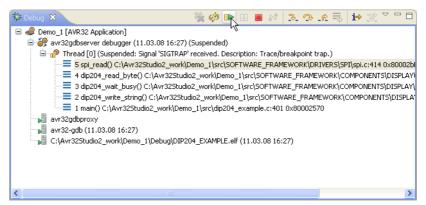


The display should look like this.

Push the joystick button on the EVK1100 evaluation board a few times, until the trace buffer is full and the target stops (6-7 button operations should be enough).



*Figure 6-38.* Target stopped because trace buffer full



#### Figure 6-39. Trace data view (not empty)

| (×)= V | /ariables 💊 Breakpoint | s 💷 AVR32 Registers | 🚱 🕼 Expressions 📣 Trace Data | 1111 F | egisters 🛋 Modules 🛛 🛃 🛃 😸 🖬 🥔 🖉 🖉 🖃 🖛 🗖                |  |  |  |  |
|--------|------------------------|---------------------|------------------------------|--------|---|--|--|--|--|
| Demo   | Demo_1                 |                     |                              |        |   |  |  |  |  |
|        | Frame #                | Address             | Frame description            | ^      | Direct call into src/SOFTWARE_FRAMEWORK/COMPONENTS      |  |  |  |  |
|        | 0x00000000             |                     | Watchpoint hit               |        | {   |  |  |  |  |
|        | 0x00000001             | 0x80002554          | Trace started                |        | <pre>spi_selectChip(DIP204_SPI, DIP204_SPI_NPCS);</pre> |  |  |  |  |
| 2      | 0x00000002             | 0x80002554          | main()                       |        |   |  |  |  |  |
| 2      | 0x0000003              | 0x80003958          | dip204_set_cursor_posi       |        |   |  |  |  |  |
| 1      | 0x00000004             | 0x80003a50          | dip204 select()              |        |   |  |  |  |  |
| 1      | 0x00000005             | ∛ 0x800027fa        | spi_selectChip()             |        |   |  |  |  |  |
| 1      | 0x00000006             | 0x80002872          | spi_selectChip()             |        |   |  |  |  |  |
| Ð      | 0x00000007             | 0x80002888          | spi_selectChip()             |        |   |  |  |  |  |
| Ð      | 0x0000008              | 0x80003a60          | dip204_select()              |        |   |  |  |  |  |
| 1      | 0x00000009             | 0x80003978          | dip204_set_cursor_posi       |        |   |  |  |  |  |
| 1      | 0x0000000a             | 0x80003a80          | dip204_write_byte()          |        |   |  |  |  |  |
| 1      | dx000000b              | 0x80003ab6          | dip204_write_byte()          | ~      | ~   |  |  |  |  |
| 2 345  | 5 trace frames         |                     |                              |        |   |  |  |  |  |

Have a look at the trace data collected by clicking on a trace frame.

| Figure 6-40. | Changing trace view format |
|--------------|----------------------------|
|--------------|----------------------------|

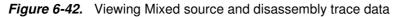
| (×)= \ | /ariables 💊 Breakpoints | AVR32 Register | s 🐗 Trace Data 🛛 👫 Registers 🚍 | \lambda Modu | ules 🛃 🛃 🛃 🛃 🛃 🛃 🛃 🛃                                    |
|--------|-------------------------|----------------|--------------------------------|--------------|---|
| Demo   | 0_1                     |                |                                |              | Set Format to Disassembly                               |
|        | Frame #                 | Address        | Frame description              | <u>^</u>     | Direct call into src/SOFTWARE_FRAMEWORK/COMPONENT       |
|        | 0x00000000              |                | Watchpoint hit                 | -            | {   |
|        | 0x00000001              | 0x80002554     | Trace started                  |              | <pre>spi_selectChip(DIP204_SPI, DIP204_SPI_NPCS);</pre> |
| Ł      | 0x00000002              | 0x80002554     | main()                         |              |   |
| 2      | 0x00000003              | 0x80003958     | dip204 set cursor posi         |              |   |
| L R    | 0x00000004              | 0x80003a50     | dip204 select()                |              |   |
| 1      | 0x00000005              | 0x800027fa     | <pre>spi_selectChip()</pre>    |              |   |
| 1      | 0x00000006              | 0x80002872     | spi_selectChip()               |              |   |
| Ð      | 0x00000007              | 0x80002888     | spi_selectChip()               |              |   |
| Ð      | 0x00000008              | 0x80003a60     | dip204_select()                |              |   |
| 2      | 0x00000009              | 0x80003978     | dip204_set_cursor_posi         |              |   |
| 1      | 0x0000000a              | 0x80003a80     | dip204_write_byte()            |              |   |
| 2      | d0000000x0              | 0x80003ab6     | dip204_write_byte()            | ~            | ~   |
| 1      |                         |                |                                |              |   |

Change the format of the code view by opening the trace format menu (click the small arrow).



| (×)= V   | 'ariables 💊 Breakpoints | AVR32 Registers | 🚱 Expressions 📣 Trace Data 🛛 | 1010 R | egisters 📑 Moo | tules      | 🛃 🎦 🖬          | 6 🔗   🖉 🗊      |         |
|----------|-------------------------|-----------------|------------------------------|--------|----------------|------------|----------------|----------------|---------|
| Demo     | _1                      |                 |                              |        |                |            | Source Only    |                | 1       |
|          | Frame #                 | Address         | Frame description            | -      | Direct cal     | l into sr  | Mixed Source a | nd Disassembly | PONENTS |
|          | 0x00000000              |                 | Watchpoint hit               |        | {              |            | Disassembly Or |                | 1       |
|          | 0x00000001              | 0x80002554      | Trace started                |        | spi_sel        | ectChip(DI | P204_SPI, 1    | DIP204_SPI_1   | WPCS);  |
| 2        | 0x00000002              | 0x80002554      | main()                       |        |                |            |                |                |         |
| 2        | 0x00000003              | 0x80003958      | dip204_set_cursor_posi       |        |                |            |                |                |         |
| 1        | 0x00000004              | 0x80003a50      | dip204_select()              |        |                |            |                |                |         |
| 1        | 0x00000005              | 0x800027fa      | spi_selectChip()             |        |                |            |                |                |         |
| 1        | 0x00000006              | 0x80002872      | spi_selectChip()             |        |                |            |                |                |         |
| Ð        | 0x00000007              | 0x80002888      | spi_selectChip()             |        |                |            |                |                |         |
| Ð        | 0x0000008               | 0x80003a60      | dip204_select()              |        |                |            |                |                |         |
| <b>-</b> | 0x00000009              | 0x80003978      | dip204_set_cursor_posi       |        |                |            |                |                |         |
| 1        | 0x0000000a              | 0x80003a80      | dip204_write_byte()          |        |                |            |                |                |         |
| <b>å</b> | d0000000x0              | 0x80003ab6      | dip204_write_byte()          | ~      |                |            |                |                | ~       |
| 2 345    | trace frames            |                 |                              |        | <              |            |                |                | >       |

Figure 6-41. Set trace view format to Mixed source and Disassembly



| (×)= \     | /ariables 💊 Breakpoints | AVR32 Registers | 🛿 🙀 Expressions 🐗 Trace Data 🛛 🛛 | 0101 R | tegisters 🛋 Modules 🛛 🛃 😤 💰 🥔 🖉 🗐 🖶 🖛 🗆                 |
|------------|-------------------------|-----------------|----------------------------------|--------|---|
| Demo       | 0_1                     |                 |                                  |        |   |
|            | Frame #                 | Address         | Frame description                | ^      | Direct call into src/SOFTWARE_FRAMEWORK/COMPONENT       |
|            | 0x00000000              |                 | Watchpoint hit                   | -      | {   |
|            | 0x00000001              | 0x80002554      | Trace started                    |        | <pre>spi_selectChip(DIP204_SPI, DIP204_SPI_NPCS);</pre> |
| 2          | 0x00000002              | 0x80002554      | main()                           |        | 0x80003a50 stmSP, R7, LR                                |
| 2          | 0x00000003              | 0x80003958      | dip204_set_cursor_posi           |        | 0x80003a54 mov R7, SP                                   |
| 1 <b>2</b> | 0x00000004              | 0x80003a50      | dip204_select()                  |        | 0x80003a56 mov R11, 2<br>0x80003a58 mov R12, -55296     |
| 1          | 0x00000005              | 0x800027fa      | spi_selectChip()                 |        | 0x80003a38 mov R12, -55296<br>0x80003a5c mcall PC[8]    |
| 1          | 0x00000006              | 0x80002872      | spi_selectChip()                 |        | OXODOJAJE MEATI PE[0]                                   |
|            | 0x00000007              | 0x80002888      | spi_selectChip()                 |        |   |
|            | 0x00000008              | 0x80003a60      | dip204_select()                  |        |   |
| L 🕹 .      | 0x00000009              | 0x80003978      | dip204_set_cursor_posi           |        |   |
| 1          | 0x0000000a              | 0x80003a80      | dip204_write_byte()              |        |   |
| L 🔒        | d0000000x0              | 0x80003ab6      | dip204_write_byte()              | ~      |   |
| 2 345      | i trace frames          |                 |                                  |        |   |

Double-click on a trace frame to highlight source code in the source editor.



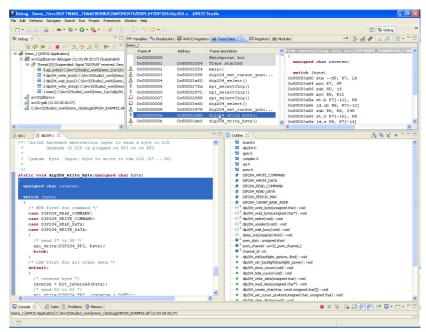
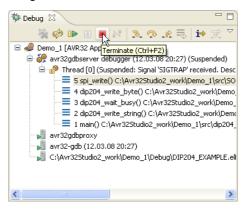


Figure 6-43. Trace frame highlighting source code in the editor

# 6.7 Modify the code and restart the debug session

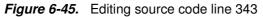
If we want to make changes to our code, we must stop the debug session, edit, rebuild and start the debug session again.

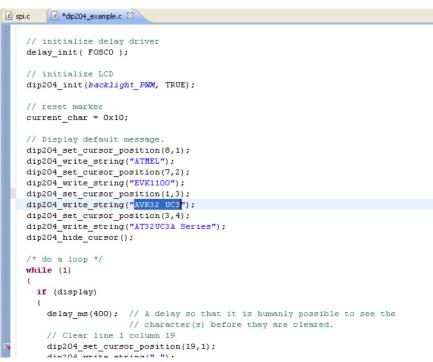
| Figure 6-44. | Terminating the | debug session |
|--------------|-----------------|---------------|
|--------------|-----------------|---------------|



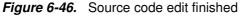


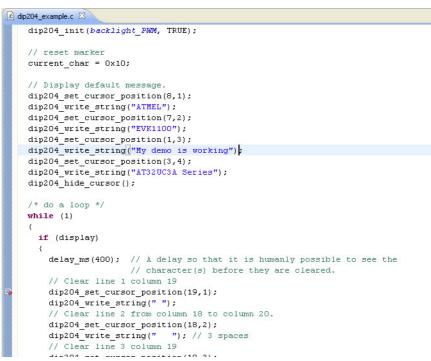
Edit the source code. This example changes the cursor position in line 342 from (6,3) to (1,3), then the text in line 343.











Start a new debug session. AVR32 Studio uses the previous Launch Configuration if you just press the Debug button.



| 🙆 Debug   | g - Demo_   | _1/src/di | p204_e   | xamp     | le.c - A | VR32 S | itudi      | 0 |
|-----------|-------------|-----------|----------|----------|----------|--------|------------|---|
| File Edit | Refactor    | Navigate  | Search   | Run      | Project  | Frame  | vork       | W |
| i 📬 •     | 8 🕹   1     | 🗟 i 🧇     | • 🌴 •    | 0        | - 💁 -    | : 🔗    | 8 ₽        |   |
| 🏇 Debug   |             |           | -0       |          |          |        |            |   |
| S         | 🏚 🔅 🕩       |           | 84   B.  | Ċ,       | e =5     | i⇔ ऱ   | <u>¢</u> ~ |   |
| = 🥔 <     | terminated. | >Demo_1 [ | AVR32 Ap | plicatio | on]      |        |            |   |

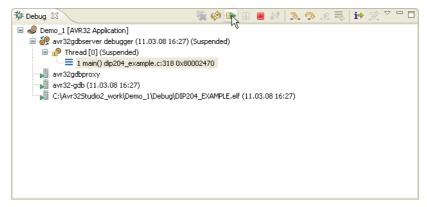
#### Figure 6-48. Save edited source code dialog

| 6 | Save Resource 🛛 🗙                                   |
|---|---|
|   | /dip204_example.c' has been modified. Save changes? |
|   | Yes No Cancel                                       |

Confirm saving the edited source code file. AVR32 Studio2.5 will now rebuild the project and program the target MCU FLASH. The code will run from start to main() and halt.



Figure 6-49. Resume button



Click "Resume" to start the application.

*Figure 6-50.* LCD Display showing edited message

| <b>EXERCISE ATMEL TRADE OF A</b> |
|----------------------------------|
| EVK1100                          |
| My Demo is working!              |
| HI32UC3A Series                  |

The LCD display should now contain the edited message.

Congratulations! You have now created your first AVR32 application and collected real time trace data from the target MCU running your program using the AVR ONE!





# Section 7

# **Firmware Upgrade**

## 7.1 Firmware upgrade overview

The tools (adapters) used to provide the physical connection between PC and target MCU contains firmware. This firmware needs to be compatible with the gnu toolchain and AVR32 Studio installed on the PC.

When AVR32 Studio is started, or when a new adapter is detected, AVR32 Studio will perform a firmware version check to determine if the adapter firmware needs to be upgraded.

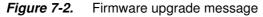
If AVR32 Studio contains a newer firmware than present in the adapter, the adapter will be upgraded.

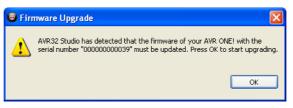
## 7.2 Firmware version test and upgrade

When AVR32 Studio is testing the firmware version of connected adapters, you can see a progress indicator in the status line.

### Figure 7-1. Firmware version test







If the adapter firmware must be upgraded, you will be notified by a pop-up. Click **OK** to continue.

Firmware upgrade progress can be monitored by activating the Progress view.

#### *Figure 7-3.* Firmware upgrade progress

| 🖹 Problems 🔲 Properties 🖳 Console 🙋 Progress 🙁  | * | ~ |  |
|---|---|---|--|
| Automatic Update Search                         |   |   |  |
|   |   |   |  |
| Searching: Contacting Target Management Updates |   |   |  |
| Upgrading firmware                              |   |   |  |
|   |   |   |  |
|   |   |   |  |
|   |   |   |  |
|   |   |   |  |
|   |   |   |  |

A firmware upgrade report can be found in the *Console* view.

*Figure 7-4.* Firmware upgrade report

| Problems 🔲 Properties | 📮 Console 🛛  | 🧵 Progress  |                 | 2 | <u></u> | 1 | - 8 |
|-----------------------|--------------|-------------|-----------------|---|---------|---|-----|
| AVR32 Console         |              |             |                 |   |         |   |     |
| Upgrading AVR ONE!    | FPGA image ' | avr32', ple | ease wait       |   |         |   | ^   |
| Firmware Image        | On disk      | On tool     | Status          |   |         |   |     |
|                       |              |             |                 |   |         |   |     |
| avrone.bin            | 1.1          | 0.20        | UpgradeRequired |   |         |   |     |
| avr32.bin             | 1.1          | 0.e         | UpgradeRequired |   |         |   |     |
|                       |              |             |                 |   |         |   | ~   |
| <                     |              |             |                 |   |         |   | >   |

## 7.3 Adapter in use

The firmware version test is a process that is running in the background. This may cause a situation where the adapter is busy (debug session active) when AVR32 Studio determines that the firmware should be upgraded. In this case, the firmware upgrade process will wait until the adapter is not busy anymore (debug session terminated).

| Figure 7-5. | Firmware | upgrade | process | waiting | for a | adapte | r |
|-------------|----------|---------|---------|---------|-------|--------|---|
|             |          |         |         |         |       |        |   |

| 🖹 Problems 🔲 Properties 📮 Console 🙋 Progress 🕴 | × | $\bigtriangledown$ |  |
|--|---|--------------------|--|
| ZzzUpgrading firmware (Sleeping)               |   |                    |  |
|  |   |                    |  |
|  |   |                    |  |
|  |   |                    |  |
|  |   |                    |  |
|  |   |                    |  |
|  |   |                    |  |

