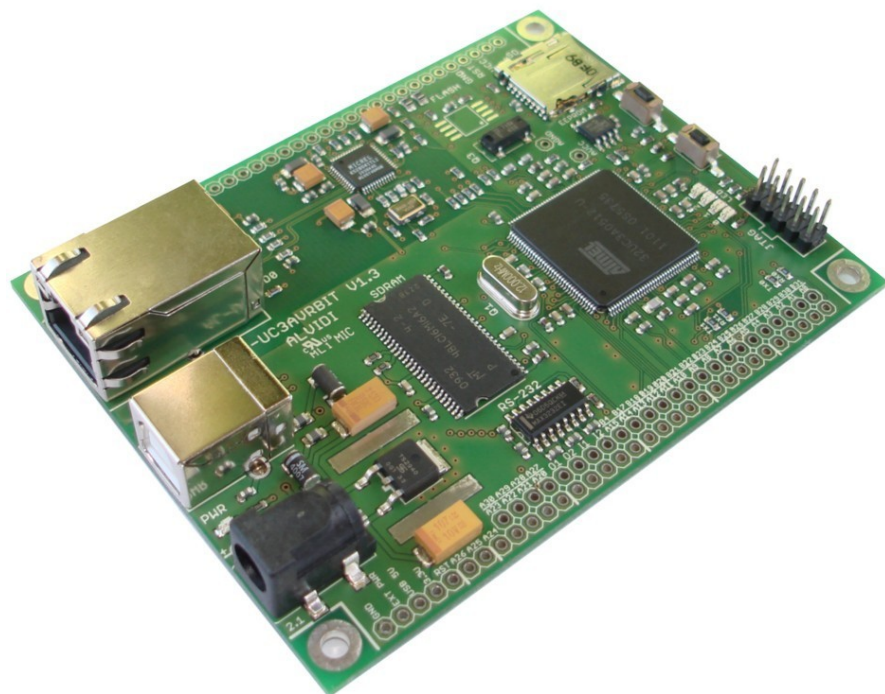


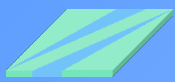
ALVIDI

First Step

***From Download to
the Programming***

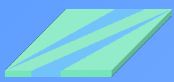
Revision from 27.07.2012





Directory

1.	Instructions	3
2.	Download	4
3.	Installation	5
4.	Atmel Studio 6	6
4.1.	New Project	7
4.2.	Adding the Library	8
4.3.	Compilation of the Project	11
5.	Hardware	12
5.1.	Start the Boot Loader	12
5.2.	Installation of the USB-Driver	13
6.	Programming	14
6.1.	Expanding the Program	14
6.2.	Installation of the Programmers	15
6.3.	Controller Programming	16
7.	Sources	17



1. Instructions

You will find similar information also on the web page of Atmel. In this document we introduce you with the help of our product, AVR32 board, our experiences with AVR32.

This document allows you a quick access to the AVR32 controller's family. It leads you from downloading of the free software and tools of Atmel up to the programming of the controller.

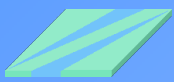
One knows that the first step is very hard. We have done this step and present it in this document.

To make acquaintance with AVR32 you need the following:

- Software
 - Atmel Studio 6.0 (build 1843) Installer - Full
 - FLIP 3.4.5 or higher
- Hardware
 - AVR32-Board AL-UC3AVRBIT mit AT32UC3A0512 Controller or
 - another one AVR32 product
- JTAG-Programmer
 - AVR JTAGICE MKII oder
 - AVR Dragon
- USB-Programmer
 - USB-Boot loader (on each Controller UC3-series installed)

There are two options JTAG & USB-Programmer to program AVR32 controller. As cheaper solution for external Programmer we recommend AVR Dragon. Advantage these JTAG Programmer is the Debug function. As a second option there is the free USB-Boot Loader to choose. How the name already tells, it needs only the USB connection with the computer.

Every controller of the UC3 series is preprogrammed with the USB-Boot Loader, it means that all three above mentioned hardwares can be programmed with an USB cable.



2. Download

You will find the complete software on the web page of Atmel.

- **Atmel Studio 6.0 (build 1843) Installer - Full**

<http://www.atmel.com/tools/ATMELSTUDIO.aspx>

this file contains the development environment for AVR32 - controller. Before you can download this software, you should fill every (*) field. After that appears a window with download link.

- **Atmel Software Framework 3.3.0**

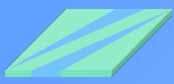
<http://www.atmel.com/tools/AVRSOFTWAREFRAMEWORK.aspx>

this file contains numerous examples, drivers, source codes, ready projects, HTML documentation, software services...

- **FLIP 3.4.5 for Windows (Java Runtime Environement included)**

http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3886

FLIP (**FL**exible **I**n-system **P**rogrammer) supports In system programming of Flash devices through RS232, USB or CAN. This file contains also the driver of USB-Boot Loader.



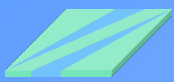
3. Installation

1. As first you have to install **as6installer-6.0.1843.exe**



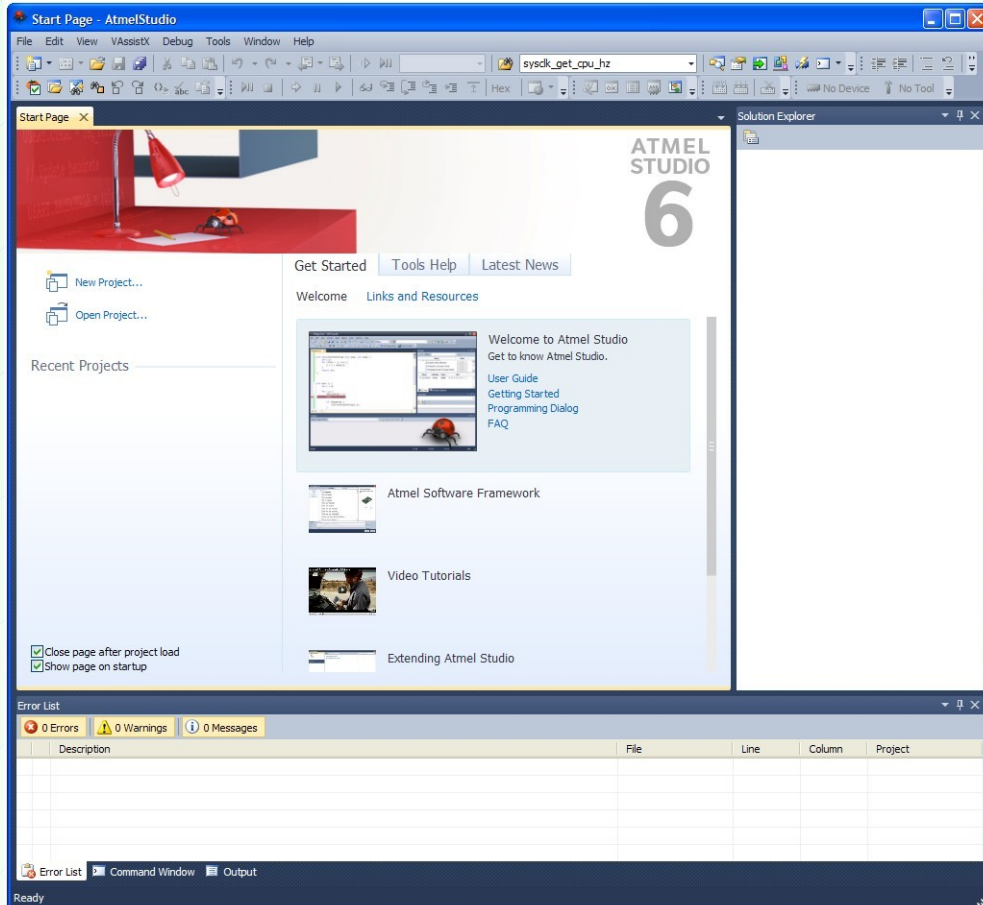
Follow the instructions in the window and install Atmel Studio 6. The installation will take some minutes. Let your Firewall allow the complete installation.

2. Unzip **asf-standalone-archive-3.3.0.zip**, e.g to C:\
3. The last step is the installation of **Flip Installer - 3.4.5.106.exe**

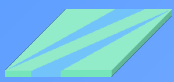


4. Atmel Studio 6

Start AVR32 Studio *Start* → *Programs* → *Atmel* → *Atmel Studio 6.0*

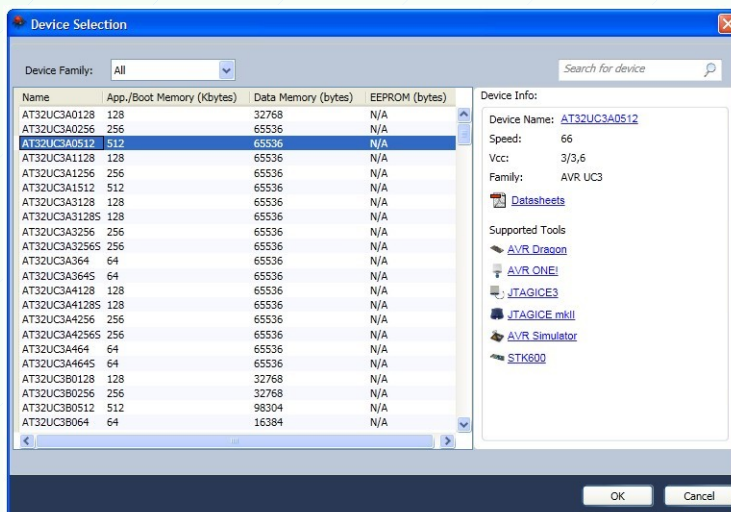
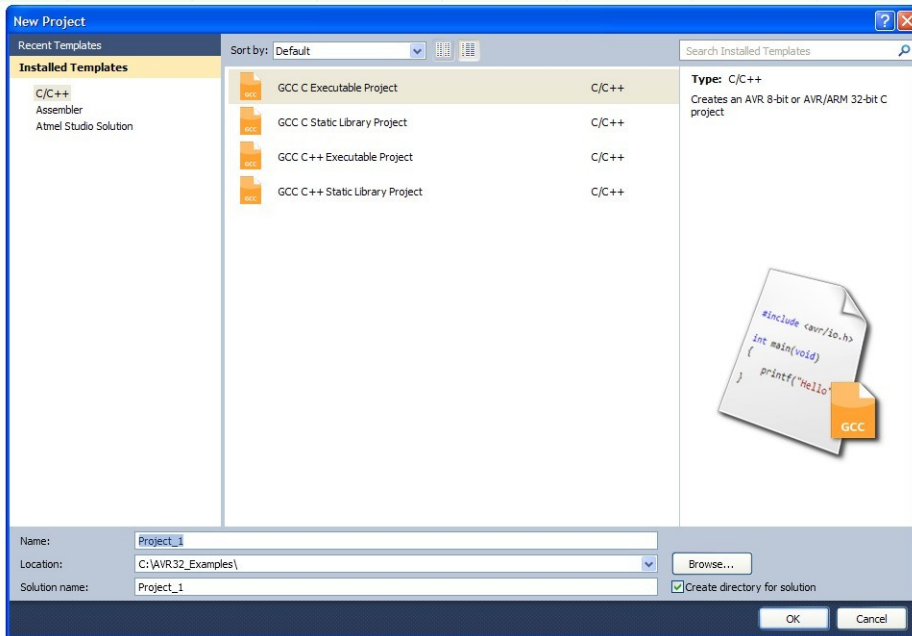


The window "Start Page" welcome you. Here you can get more information about the software, hardware and programmer.



4.1. New Project

Start a new project *File* → *New* → *Project...* In the middle window choose „GCC C Executable Project“. Named your project in the field *Name*, e.g. **Project_1** and save it in field *Location* e.g. **C:\AVR32_Examples**. Finally click the button „OK“. In the next step You have to choose your controller. In our case it is AT32UC3A0512. With the next button „OK“ will be closed the second window. As soon you closed the last window you will get an empty C-File „Project_1.c“ with start mask.



Now we will write a small program in the new created source file. In this program we change periodically the level of the pin 12 on the port X. In our AVR32 Board is this pin with LED0 connected.

```
#include <avr32/io.h>
#include "gpio.h"
#include "sysclk.h"
#include "delay.h"

int main(void)
{
    sysclk_init();
    delay_init(sysclk_get_cpu_hz());

    while(1)
    {
        gpio_tgl_gpio_pin(AVR32_PIN_PX12);
        delay_ms(500);
    }
}
```

Take over the source code illustrated on top in *Project_1.c*

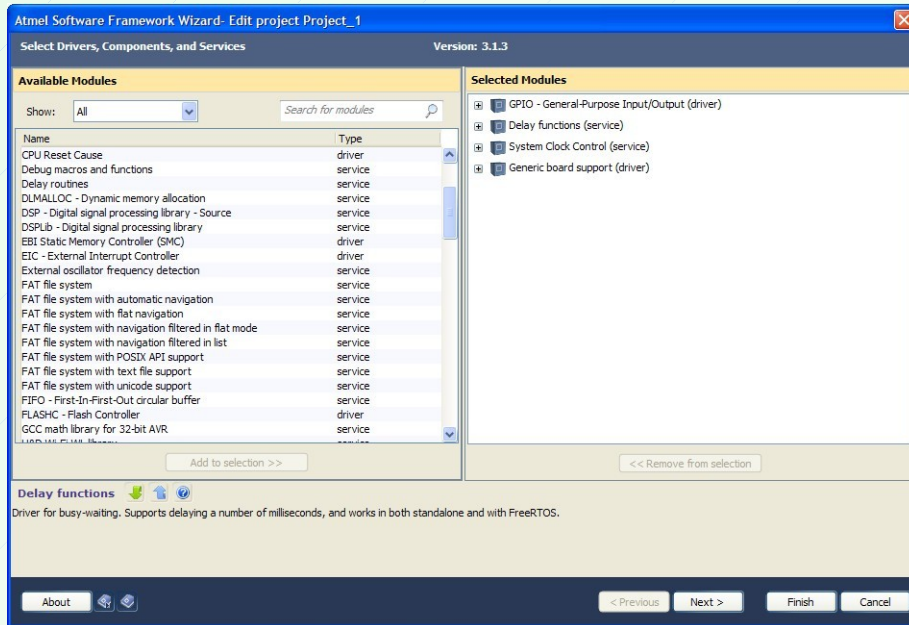
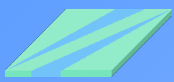
4.3. Adding the Library

After writing our program, we have to include a few library's to our project. This library's you will find *Project* → *ASF Wizard*

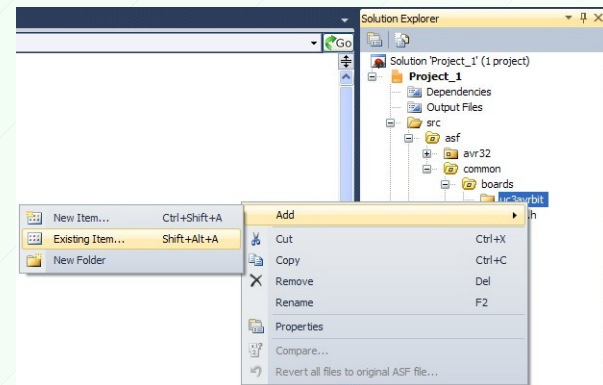
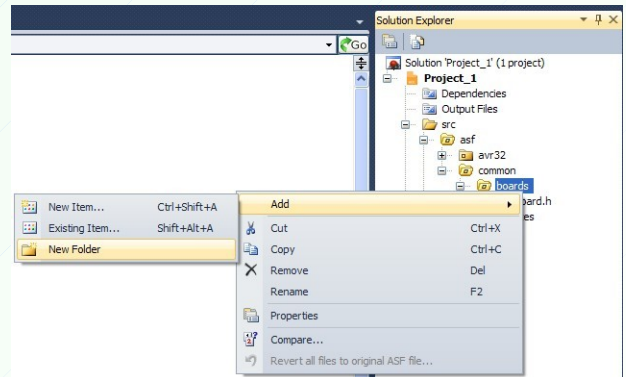
Choose in the left field library's as following :

- GPIO – General-Purpose Input/Output
- Generic board support
- System Clock Control
- Delay functions

With the button „Add to selection >>“ transfer the above choice in the right window. Confirming with the button „Finish“ include the library's to our project.



Create a new folder uc3avrbit in the window **Solution Explorer** under **Project_1** → *src\as\common\boards*. With the right mouse key on the folder boards → *Add* → *New Folder*



With the right mouse key on the folder uc3avrbit → *Add* → *Existing Item...* include following files:

http://alvidi.de/data_sheets/AL_LIB_UC3.zip →

AL_LIB_UC3 → UC3AVRBIT:

[uc3avrbit.h](#)

[led_func.c](#)

[led_func.h](#)

In the library **board.h** under *src\asf\common\boards* change the following lines:

```
#define USER_BOARD          99  //!< User-reserved board (if any).
#define DUMMY_BOARD        100  //!< Dummy board to support board-independent
```

as:

```
#define USER_BOARD          99  //!< User-reserved board (if any).
#define DUMMY_BOARD        100  //!< Dummy board to support board-independent
#define UC3AVRBIT          101  //!< ALVIDI AL-UC3AVRBIT board with AT32UC3A0512
```

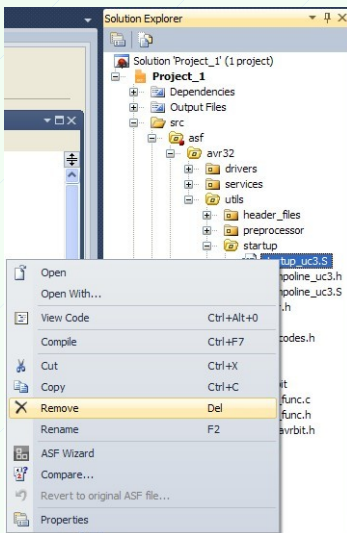
```
#define BOARD                UC3AVRBIT
```

and

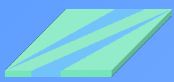
```
#if BOARD == EVK1100
#include "evk1100/evk1100.h"
```

with

```
#if BOARD == UC3AVRBIT
#include "uc3avrbit/uc3avrbit.h"
#elif BOARD == EVK1100
#include "evk1100/evk1100.h"
```



Now we have to make the last change. We should delete the library **startup_uc3.S** under *src\asf\avr32\utils\startup*



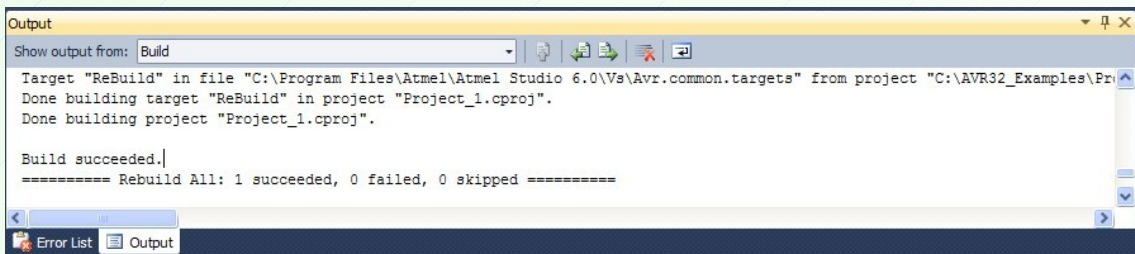
4.3. Compilation of the Project

At first you have to save the complete Project File → *Save ALL*.

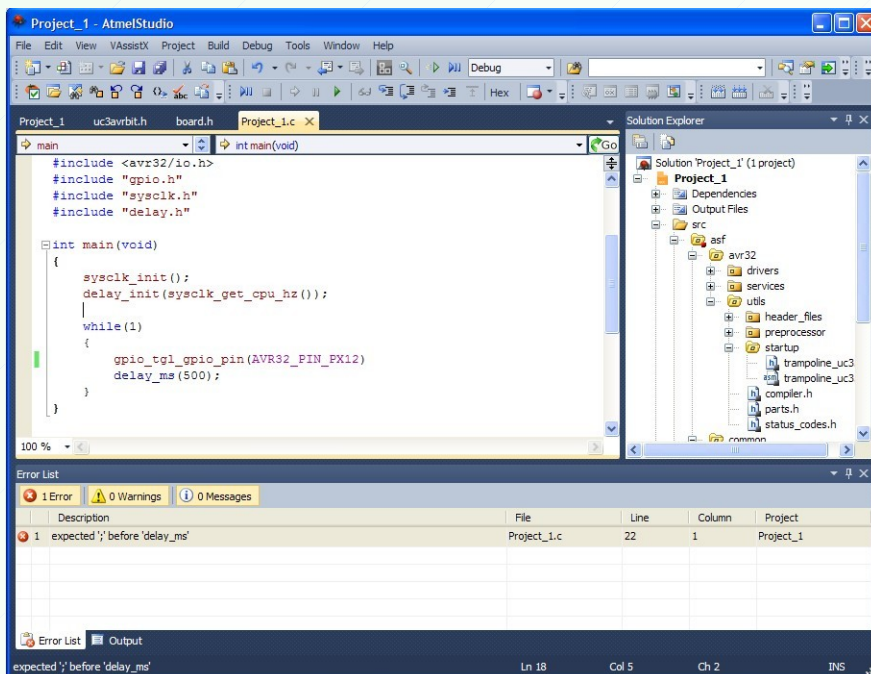
Compile the project:

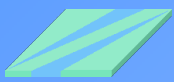
- *Build* → *Rebuild Solution* **or**
- key sequence **[Ctrl]+[Alt]+[F7]** **or**
- right mouse button on **Solution 'Project_1'** → *Rebuild Solution*

You will see the result of the compilation in the window "Output" how it is shown in lower picture.



If during the compilation mistakes were found in the program, they would appear in the window "Error List". The lower picture introduces this case.

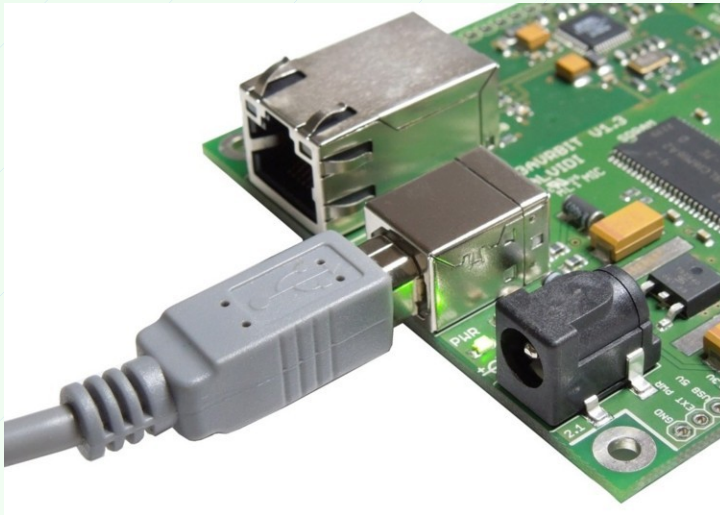




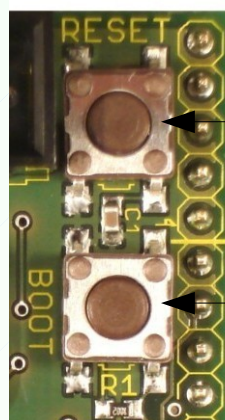
5. Hardware

5.1. Start the Boot Loader

Connect your hardware via USB with computer, e.g. AVR32-Board. As soon the USB connected to the board the green *Power LED* will give light.

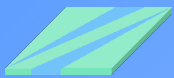


Hold the "BOOT"-key low-pressed and press for a short time the "RESET"-key. Therefore you start the boot loader.



2. press for a short time

1. hold low-pressed



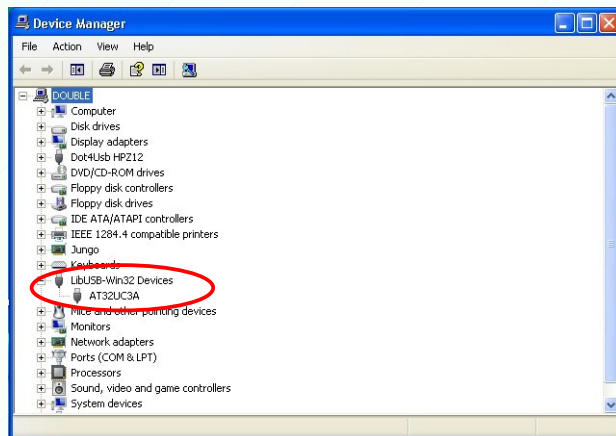
5.2. Installation of the USB-Driver

When the USB connection is available and boot loader has been started you get the picture below.

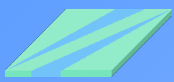


Choose *Install from a list or location (Advanced)* and click the button „Next >“.

Enter in white field the folder of the driver
C:\Program Files\Atmel\Flip 3.4.5\usb and
click the button „Next >“.



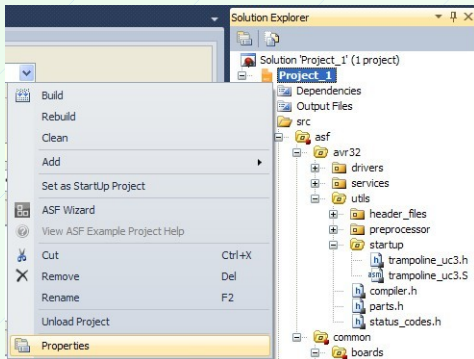
After a successful installation appears the left upper picture. In the right picture *Device Manager* the connected device will be visible under **LibUSB-Win32 Devices**→**AT32UC3A**



6. Programming

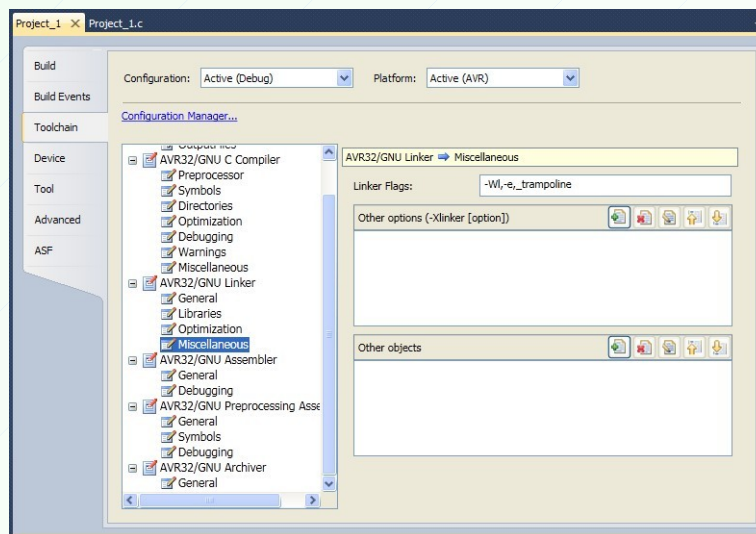
6.1. Expanding the Program

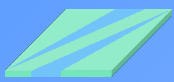
To program with USB-boot loader we must expand properties of our *Project_1* with a line.



Click with the right mouse key in the window *Solution Explorer* on **Project_1** and choose the option **Properties**.

In the field **Linker Flags** under *Project_1* → *Toolchain* → *AVR32/GNU Linker* → *Miscellaneous* write the line `-Wl,-e,_trampoline`





6.2. Installation of the Programmer

You will find in pdf-File AVR32 UC3 USB DFU Bootloader of Atmel, how to program with boot loader. See: <http://www.atmel.com/Images/doc32166.pdf>

This document show us that for programming you are required a Windows program “Command Prompt”. This program you will find under *Start → Accessories → Command Prompt*

With the command line e.g.

cmd.exe /C batchisp -device AT32UC3A0512 -hardware usb -operation onfail abort memory FLASH erase F

the memory area without boot area will be erased and with this command line

cmd.exe /C batchisp -device AT32UC3A0512 -hardware usb -operation onfail abort memory FLASH erase F loadbuffer

C:\AVR32_Examples\Project_1\Project_1\Debug\Project_1.hex program verify start reset 0

the memory area without boot area will be erased, with the file Project_1.hex flashed, verified and after internal reset started the program .

That means, in order to program a controller via USB-boot loader, we have to include the command line above in the window *Command Prompt* every time.

We could simplify that with a self executive file. For that case we need a “Notepad”. This program you will find also under *Start → Accessories → Command Prompt*

In the next steps we explain, how to create such file.

1. Step: open “Notepad” and paste the following line:

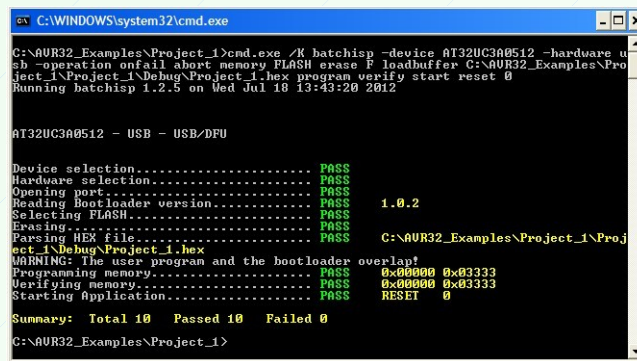
cmd.exe /C batchisp -device AT32UC3A0512 -hardware usb -operation onfail abort memory FLASH erase F loadbuffer

C:\AVR32_Examples\Project_1\Project_1\Debug\Project_1.hex program verify start reset 0

2. Step: save that file e.g. under C:\AVR32_Examples\Project_1 **not** as txt-file **but** as bat-file. We named it **PROGRAM.bat**

6.3. Controller Programming

In order to program AVR32 board, we have as first start boot loader. See chapter 5.1. *Start the Boot Loader*. In 2 steps from the last chapter we created one bat-files. **PROGRAM.bat** will be run each time, if we need it up to date. For an other project you have to change the link in step 3 (**C:\AVR32_Examples\Project_1\Project_1\Debug\Project_1.hex**). The window closed after execution the command line. If you like to let the windows after execution permanent open, you have to change **cmd.exe /C** with **cmd.exe /K** The programming result will be shown in detail in the window Command Prompt (see picture below).



```

C:\WINDOWS\system32\cmd.exe
C:\AVR32_Examples\Project_1>cmd.exe /K batchisp -device AT32UC3A0512 -hardware u
sb -operation onflash abort memory FLASH erase F loadbuffer C:\AVR32_Examples\Pro
ject_1\Project_1\Debug\Project_1.hex program verify start reset 0
Running batchisp 1.2.5 on Wed Jul 10 13:43:20 2012

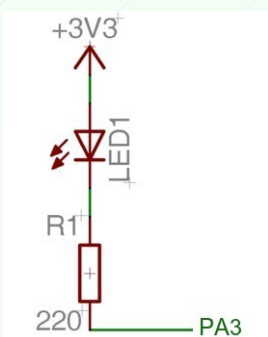
AT32UC3A0512 - USB - USB/DFU

Device selection..... PASS
Hardware selection..... PASS
Opening port..... PASS
Reading Bootloader version..... PASS 1.0.2
Selecting FLASH..... PASS
Erasing..... PASS
Parsing HEX file..... PASS C:\AVR32_Examples\Project_1\Proj
ect_1\Debug\Project_1.hex
WARNING: The user program and the bootloader overlap!
Programming memory..... PASS 0x00000 0x03333
Verifying memory..... PASS 0x00000 0x03333
Starting Application..... PASS RESET 0

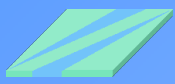
Summary: Total 10 Passed 10 Failed 0
C:\AVR32_Examples\Project_1>

```

Now we have to make sure that the program functions properly. If no LED on the board available, connect on port X pin 12 in a row resistor 220 Ω and a LED like in the lower circuit. If the LED flashes, you have made everything right.



You will find other examples for AVR32 controller series UC3 in the AVR32 UC3 Software Framework. In the folder <C:\asf-2.3.1\avr32\drivers> you will find also the drivers and source code examples of ADC, PWM, RTC, USART...



7. Sources

- **AVR32015: Atmel Studio 5 Video Beispiele**
http://www.atmel.com/microsite/avr_studio_5/
- **AVR32 UC3 USB DFU Bootloader**
<http://www.atmel.com/Images/doc32166.pdf>