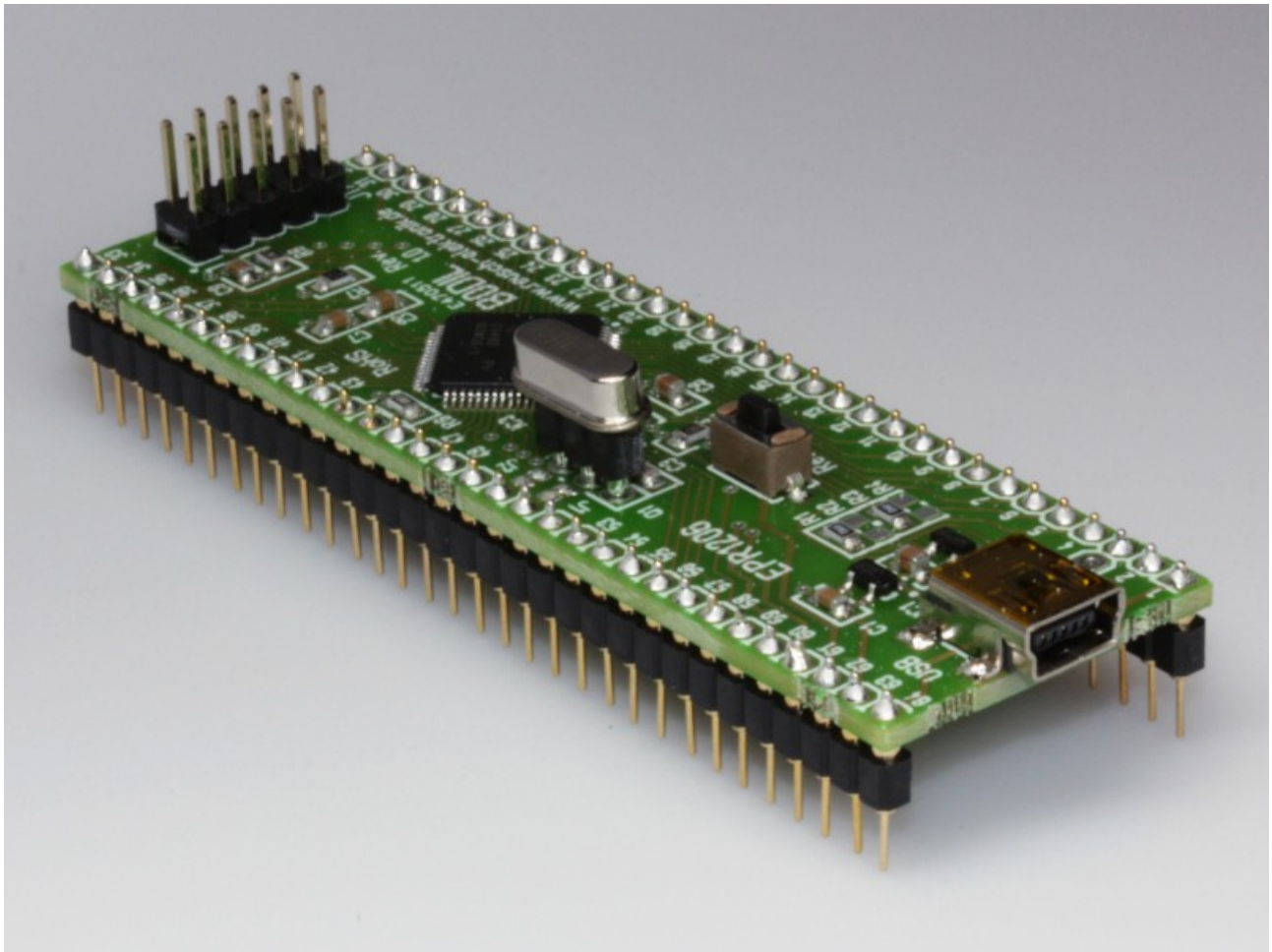
B0DIL

AVR32 USB Module

Rev. 1.0
Documentation Rev. 3



Reusch Elektronik

www.reusch-elektronik.de

http://products.reworld.eu/b0dil.htm

File: B0DIL_Manual
Created: 2012-08-16
Changed: 2012-08-16

Table of Contents

Please note:

This document refers to revision 1.0 of the B0DIL module. If you are using a device with another revision number, please refer to the corresponding documentation!

Reusch Elektronik
Dipl.-Ing. (FH) Rainer Reusch

Blumenstr. 13
D-88097 Eriskirch
Germany

Phone: +49-7541-81484
Fax: +49-7541-81483
E-Mail: info@reusch-elektronik.de
Homepage: www.reusch-elektronik.de

1. Introduction

B0DIL is a tiny interface module for an Atmel AVR32 microcontroller with USB interface in TQFP64 casing. The module fits to an ordinary 64 pin dual-in-line socket. The B0DIL module is helpful in experiments, development and prototyping with USB-AVR32 micro-controllers.

The module contains the crystal oscillator. A socket is used, so it is easy to replace the crystal or ceramic resonator to get the required frequency.

The USB interface (mini USB connector) is on board.

The module contains a 10 pin header for in-system-programming and debugging (Atmel 10 pin JTAG standard). These pins are also available by the dual-inline connector pins.

The B0DIL module is easy to use, because it contains all necessary and time critical interfaces. On the other hand, it gives full flexibility in power supplement and connectivity!

1.1 Assembly Variations

The module is available in one assembly version:

- as “B0DIL-AT32UC3B0512”, assembled with AT32UC3B0512 from Atmel

The printed circuit board is also available “naked” (not assembled). It is suitable for the following Atmel microcontrollers with TQFP64 casing:

- AT32UC3B064
- AT32UC3B0128
- AT32UC3B0256
- AT32UC3B0512

1.2 Oscillator

A 3 pin socket is used for the oscillator. This makes the module suitable for a

- crystal (HC49)
- 2 pin ceramic resonator
- 3 pin ceramic resonator

All devices in wired technology (no SMD). It might be necessary, to remove the two 22pF capacitors (C3 and C4), if a 3 pin ceramic resonator is used.

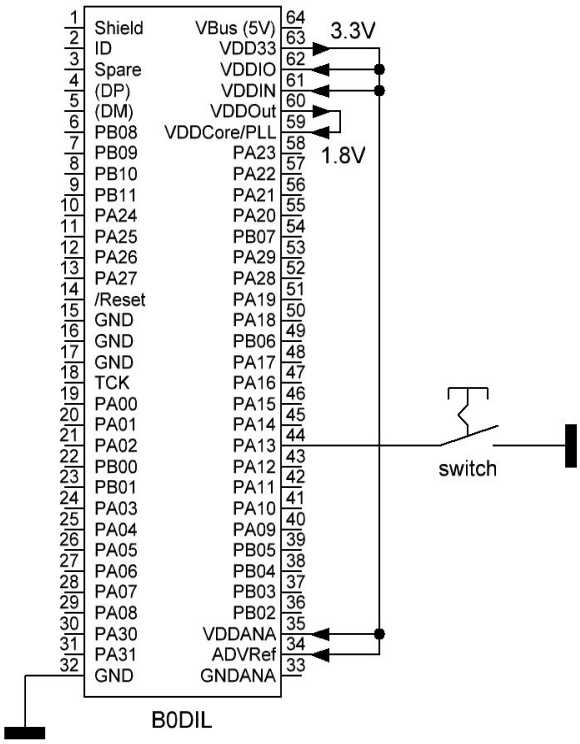
For correct USB function, the following frequencies are required:

- 8.000MHz
- 12.000MHz (delivery standard)
- 16.000MHz

Note: The pins of oscillator 0 (XIN0, XOUT0) are available at the DIL pins 50 and 51). While the microcontroller is operating with osc 0, this two pins can't be used as GPIO ports. J1 and J2 are solder pads, which are closed as factory default. These pads connect the oscillator with the DIL pins. J1 and J2 can be opened for disconnection (use a cutter).

1.3 Introduction Example

The wiring example as shown, supplies the module by USB and the onboard voltage regulator. Programming can be done via USB interface, bootloader and “Atmel batchisp”. For details refer to chapter 3.3.2.



Without firmware, the module always will enter the bootloader. It will respond as “Atmel device” at the personal computer. At the first plugging, it is asked for a driver installation. You have to use the LibUSB-Win32 drivers, which are coming with the “Atmel FLIP” installation package.

After the driver installation, the device can be accessed by “batchisp” to program your firmware application. This command line tool is coming with the “Atmel FLIP” installation.

After programming, the device has to be resetted (push reset button) with open switch (high level) at PA13, to launch the programmed firmware.

To relaunch the bootloader, close the switch (low level) at PA13 and push the $\overline{\text{Reset}}$ button for a moment.

Please note:

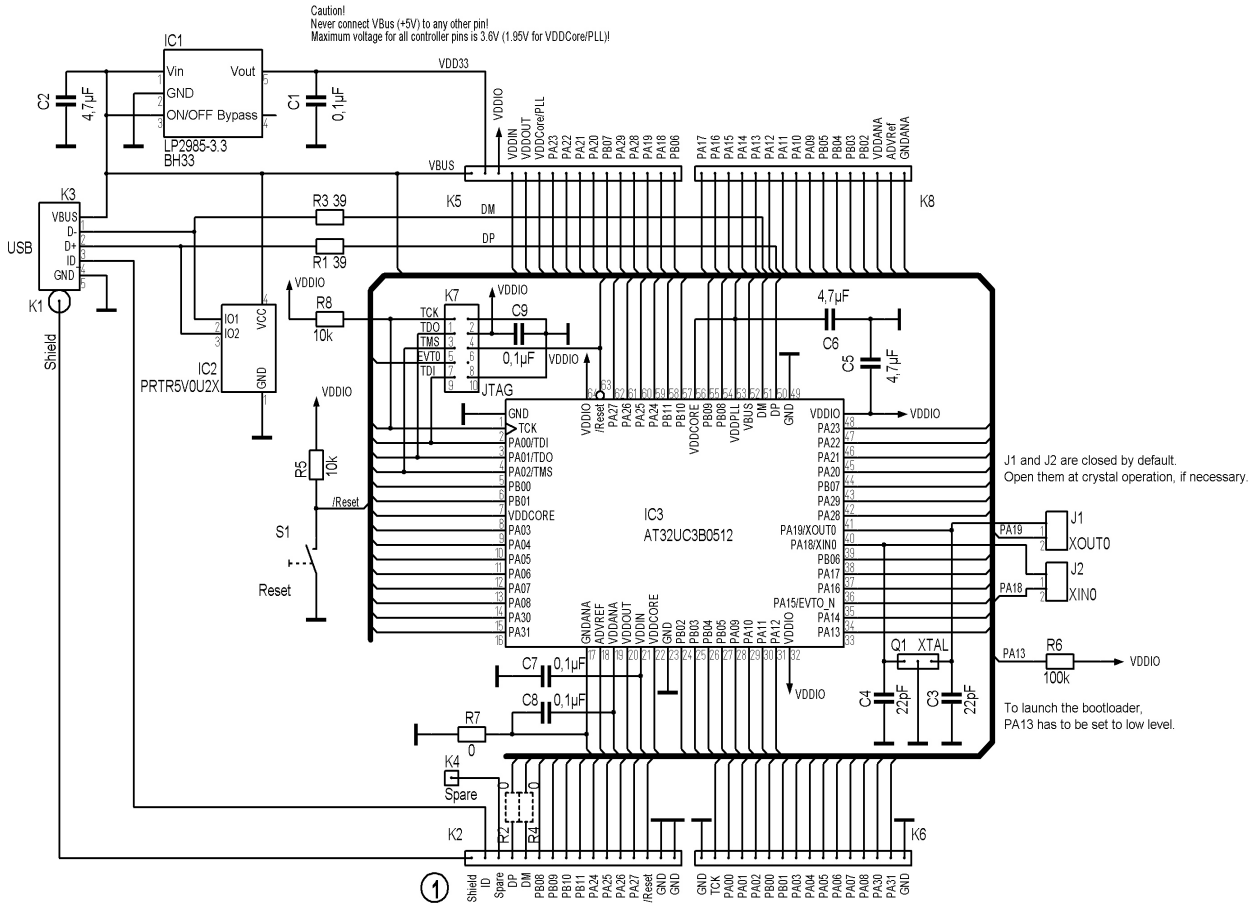
There are more options, to supply the device. The device also can be programmed via *JTAG* interface and *JTAG* programmer. For more information read the application note chapter, please.

Warnings:

If the B0DIL module is connected to USB, pin 64 outputs the USB supply voltage of approximately 5.0V. **Never** connect VDDIO (pin 62), VDDIN (pin 61) or VDDCore/PLL (pin 59) to VBus (pin 64)! Maximum supply voltage for the controller is 3.6V (1.95V for core and PLL)!

2. Technical Information

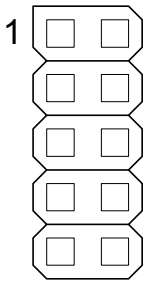
2.1 Schematic



2.2 JTAG Pin Assignment

The JTAG programming adapter is a 10 pin header, which corresponds to the 10 pin Atmel JTAG connector. Therefore the usual Atmel in-system programmers/debuggers (such as “AVR JTAG MKII”) are suitable.

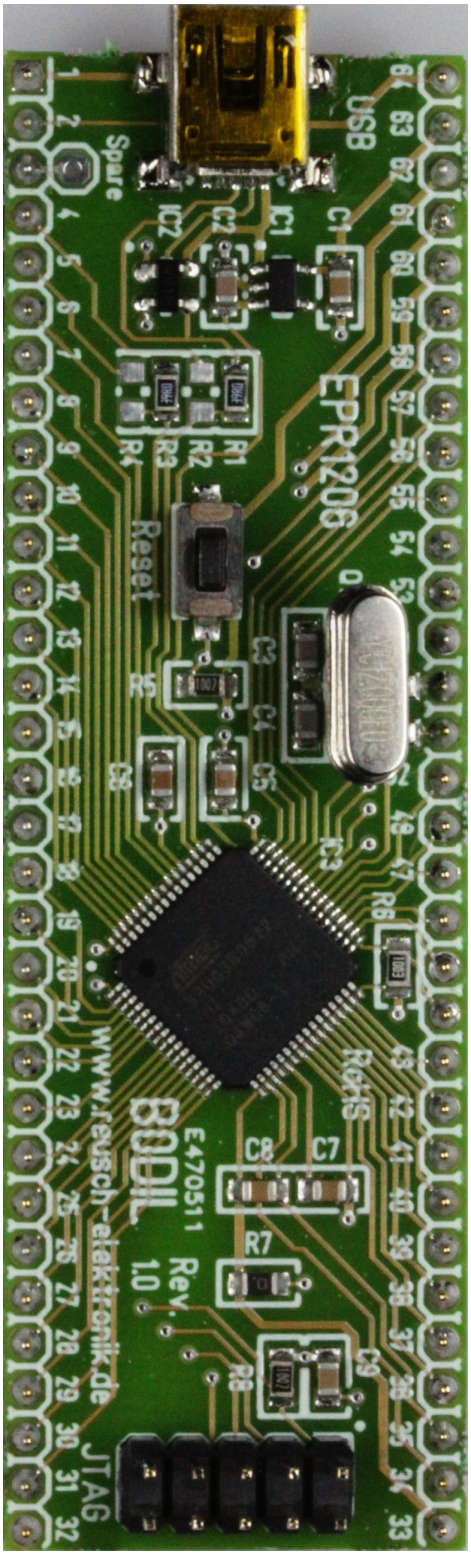
Pin	Function
1	TCK
3	TDO
5	TMS
7	EVT0
9	TDI



Function	Pin
Gnd	2
VDD (3.3V)	4
$\overline{\text{Reset}}$	6
	8
Gnd	10

2.3 Dual-Inline-Socket Pin Assignment

Pin	Function
1	Shield (USB)
2	ID (USB connector)
3	Spare
4	(DP) *
5	(DM) *
6	PB08
7	PB09
8	PB10
9	PB11
10	PA24
11	PA25
12	PA26
13	PA27
14	$\overline{\text{Reset}}$
15	GND
16	GND
16	GND
18	TCK
19	PA00
20	PA01
21	PA02
22	PB00
23	PB01
24	PA03
25	PA04
26	PA05
27	PA06
28	PA07
29	PA08
30	PA30
31	PA31
32	GND



Function	Pin
VBus (5V output)	64
3.3V output	63
VDDIO (input)	62
VDDIN (input)	61
VDDOUT (1.8V output)	60
VDDCore/PLL (1.8V input)	59
PA23	58
PA22	57
PA21	56
PA20	55
PB07	54
PA29	53
PA28	52
PA19 (Xout 0)	51
PA18 (Xin0)	50
PB06	49
PA17	48
PA16	47
PA15	46
PA14	45
PA13	44
PA12	43
PA11	42
PA10	41
PA09	40
PB05	39
PB04	38
PB03	37
PB02	36
VDDANA	35
ADVRef	34
GNDANA (GND)	33

*) DP and DM are not connected to pins by default (R2 and R4 not assembled).

For more information about the alternative pin usage, please refer to the Atmel data sheets.

Warning! Never connect VBus (pin 64) to any other pin of the device!

2.4 Power Supply Considerations

The B0DIL module contains a 3.3V regulator. The input is connected to VBus (pin 64), the output is available thru VDD33 (pin 63). This output can be used to supply the I/O of the microcontroller (VDDIO, pin 62) and external circuitry. Please note: The maximum output current of this voltage regulator is 100mA! If more current is needed, an external voltage regulator has to be used (refer to application note chapter).

The core and the PLL (VDDCore/PLL, pin 59) of the microcontroller requires a supply voltage of 1.8V. For this reason, the microcontroller contains a 1.8V regulator. The input of this regulator (max. input voltage 3.6V!) is available thru VDDIN pin (pin 61). This pin can be connected with VDD33 (pin 63). Connect the 1.8V output (VDDOUT, pin 60) of this regulator to the core/PLL supply input (VDDCore/PLL, pin 59). Of course, an external 1.8V regulator also can be used.

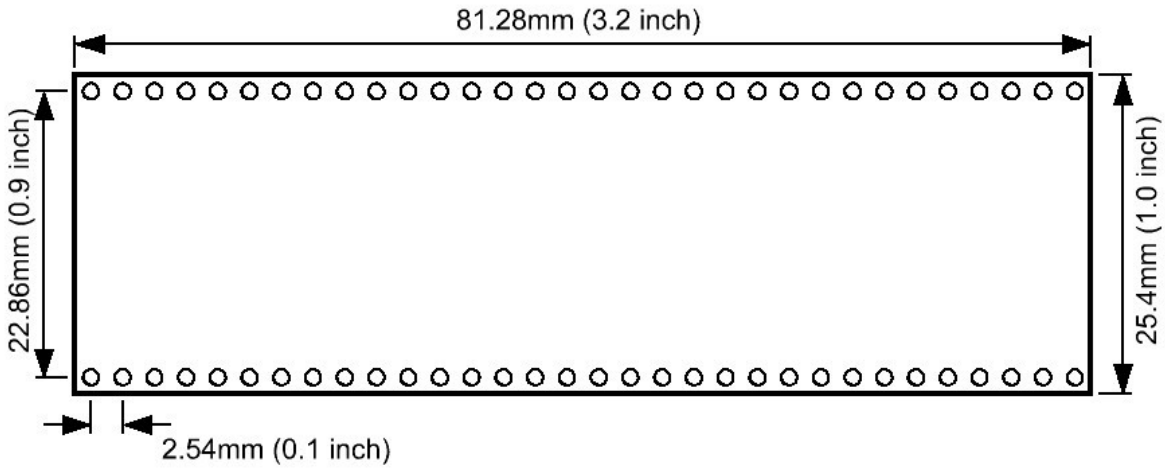
For more information refer application note chapter, please.

2.5 Technical Data

Supply voltage (I/O, 1.8V regulator)	3.0V to 3.6V, 3.3V typically
Core and PLL supply voltage	1.65V to 1.95V, 1.8V typically
Supply current	<30mA (depends on operation frequency and application)
Clock frequency	8MHz, 12MHz or 16MHz (required for USB compatibility)
Operation temperature	-25°C to 70°C
Storage temperature	-25°C to 85°C

For further information related to the microcontroller, please refer to the Atmel data sheet of the microcontroller (doc32059.pdf).

2.6 Dimensions



3. Application Notes

3.1 Power Supply Options

The module can be supplied by USB bus power or it can be self powered.

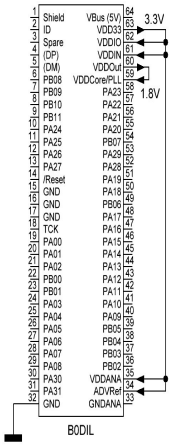
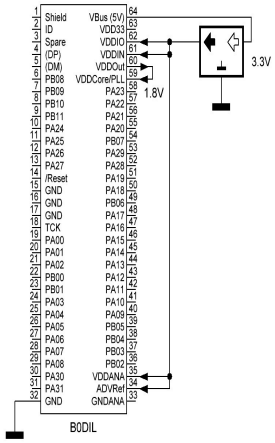
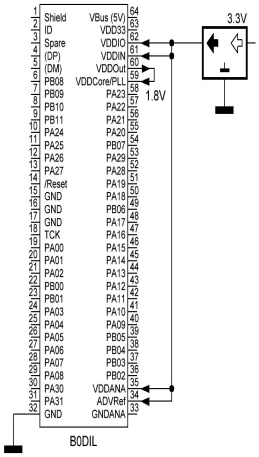
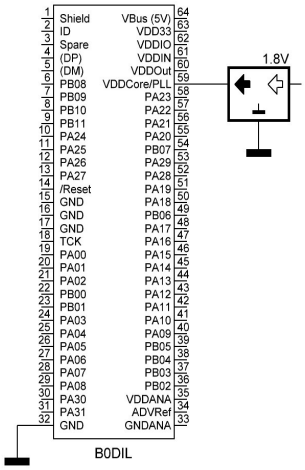
For bus powered supply with 5V, only a few connections are necessary. The onboard 3.3V regulator is used.

For bus powered supply with higher current demand (max. 500mA), an external 3.3V (3.0V to 3.6V) regulator has to be used.

For self powered supply with higher current demand (gt. than 500mA), this 3.3V regulator is connected to the external supply voltage source.

The core and the PLL of the microcontroller also can be supplied by an external voltage regulator. Usually this is not necessary.

Note: VDDANA (pin 35) has to be supplied, otherwise the ports PA03 up to PA08 doesn't reach full high level and current consumption increases!

bus powered, internal regulator	bus powered, external regulator
	
self powered	external core / PLL supply
	

For further information please refer to the Atmel data sheet of the microcontroller.

3.2 Memory Programming

The Atmel microcontroller contains an USB bootloader as factory default. So there are two ways, to program the flash memory of the microcontroller:

- programming via USB interface, bootloader and *batchisp*
- use of in-system-programming with JTAG programmer

3.2.1 Launching the Bootloader

The B0DIL module comes with the factory default DFU bootloader, which allows programming and upgrading of firmware without programming device. This bootloader will always be launched after a reset, because it is placed at address 0x80000000, the first flash memory address, were executing starts after a reset. The upper two bits of the general-purpose fuse bits and a state of a defined port determines, if the bootloader continues or a jump to the application firmware is done.

Without programmed application firmware (delivery default), the microcontroller always will launch the bootloader.

If an application firmware is programmed, the port PA13 has to be kept in low state (connection with ground) while reset is present, to launch the bootloader (after reset). A high state (or open, internal pullup resistor) on PA13 will launch the programmed firmware.

Note: Port PA13 as the launching pin for the bootloader is factory default. The port and its state can be changed by reprogramming the *flash user page*. For detailed information refer Atmel documentation *doc7745.pdf*.

Important hint: The microcontroller has to run with oscillator 0 (Osc0) and a frequency of 8, 12 or 16MHz! Otherwise a proper USB communication is not possible!

3.2.2 Programming with *batchisp*

“batchisp” is a command line tool for automatic device programming. It comes with the software package of “Atmel FLIP”. *batchisp* will be found in the subdirectory *bin* of your FLIP installation path. Usually the path environment variable is set to this directory. So *batchisp* can be called from any directory without the requirement to pretend the hole path. Detailed information about the usage of this command line tool will be given in the help of FLIP (launch FLIP and call the help over menu).

Example

This examples separates the programming commands into a script file (e.g. named *flash_write.txt*). The *batchisp* command will be written into a batch file (e.g. *flash_write.bat*).

The contents of *flash_write.bat*:

```
batchisp -cmdfile "flash_write.txt"
```

The contents of *flash_write.txt*:

```
-device AT32UC3B0512
-hardware USB
-operation erase f loadbuffer "myfile.elf" program verify start reset 0
```

“myfile.elf” is the firmware file to be programmed. Pretend path, if necessary. Hex files can also be used.

The “Atmel Flip” installation package can be downloaded from the Atmel homepage. Search for “FLIP” or try the following link:

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

3.2.3 In-System-Programming

The B0DIL module contains an Atmel JTAG connector for in-system-programming and debugging. An Atmel JTAG compatible programming device (such as “AVRJTAG MKII”) is required. The advantages of this programming method:

- full control over the microcontroller, all memories and fuses
- comfortable integration into “AVR Studio”
- no bootloader required
- suitable for any clock frequency
- debugging

Note: For the first programming by JTAG an “erase device” is necessary, to reset the fuses. This will also erase the factory default bootloader! That means, the device no more can be programmed with “batchisp”, until the bootloader (or another) is reflashed!

3.2.4 JTAG Interface

The pins 18 to 21 and 47 of the B0DIL module represents the JTAG interface. The usage of this pins in the application firmware is restricted! Any connected circuitry shouldn't have any influence, otherwise reprogramming my fault! For debugging the JTAG interface is used continuously. It is not recommended, to use this pins for other applications.

3.2.5 Reprogramming the Bootloader

AVR Studio 5.0 doesn't support a reprogramming of the bootloader. Doc7745.pdf contains a description how to do this manually (not recommended for inexperienced users). For the following steps you need an installed *AVR32 Studio V2.6.0* or later. Proceed the following steps:

- Connect JTAGICE MKII or any other programmer/debugger.
- Start AVR32 Studio.
- In the AVR target window right click and click ‘scan targets’. Right click on the target and select properties.
- After updating the details of debugger, device and board in properties window, right click in AVR target and select ‘Program Bootloader’.
- Complete the steps for programming the bootloader with default settings.

By completing the above steps, you will be able to load the USB DFU bootloader V1.0.3, the user page configuration word and the fuse bit settings.

Now you can flash your software (built with AVR Studio 5 or AVR32 Studio) with *batchisp*.

Please note: AVR32 Studio V2.6.0 will install version 1.0.3 of the bootloader. It is strictly recommended, to use version 1.0.2 or newer releases!

4. Concluding Remarks

4.1 Application Software

You will find programming and application sample software on the product homepage. Please visit the following URL:
http://products.reworld.eu/b0dil.htm

You will find additional information and tools on the following URLs.

Atmel AVR Tools and Software:
http://www.atmel.com/dyn/products/product_card.asp?category_id=163&family_id=607&subfamily_id=2138&part_id=4176

4.2 Links

Visit the following links for more information and downloads about this product, used software modules and tools.

Reusch Elektronik homepage for electronic products:
http://products.reworld.eu

Atmel Corporation, AVR data sheets:
http://www.atmel.com/dyn/products/product_docs.asp?category_id=163&family_id=607&subfamily_id=2138&part_id=4176

WinAVR, GCC development environment:
http://winavr.sourceforge.net/

LibUSB-Win32, USB drivers for Microsoft Windows ®:
http://libusb-win32.sourceforge.net/

4.3 Conformity statement

This evaluation board/kit is intended for use for **engineering development, demonstration or evaluation purposes only** and is not considered by *Reusch Elektronik* to be a finished end-product fit for general consumer use.

This evaluation board/kit does not fall within the scope of the European Union directives regarding electromagnetic compatibility, restricted substances (RoHS), recycling (WEEE), FCC, CE or UL, and therefore may not meet the technical requirements of these directives or other related directives.

4.4 Statement and Disclaimer

This electronic device is designed under best known engineering guidelines. It confirms the appropriate design rules. No warranty or liability is given for adherence, assured properties, or damages which might be caused by the usage of this hardware or the accessory software.

Note: This equipment has been tested and found to comply with the limits for a Class B digital device. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy, and if not installed and used in accordance with the instruction

manual, may cause interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

4.4 Technical Support

We hope you'll understand, that technical support by phone is not possible. If You have any questions, technical problems or You need information about other products from *Reusch Elektronik*, please contact us by e-mail or Fax.

E-Mail: support@reusch-elektronik.de

FAX: +49-7541-81483

We are able to handle inquiries in english and german language.

Don't hesitate to contact us, if You have demand for custom specific solutions!