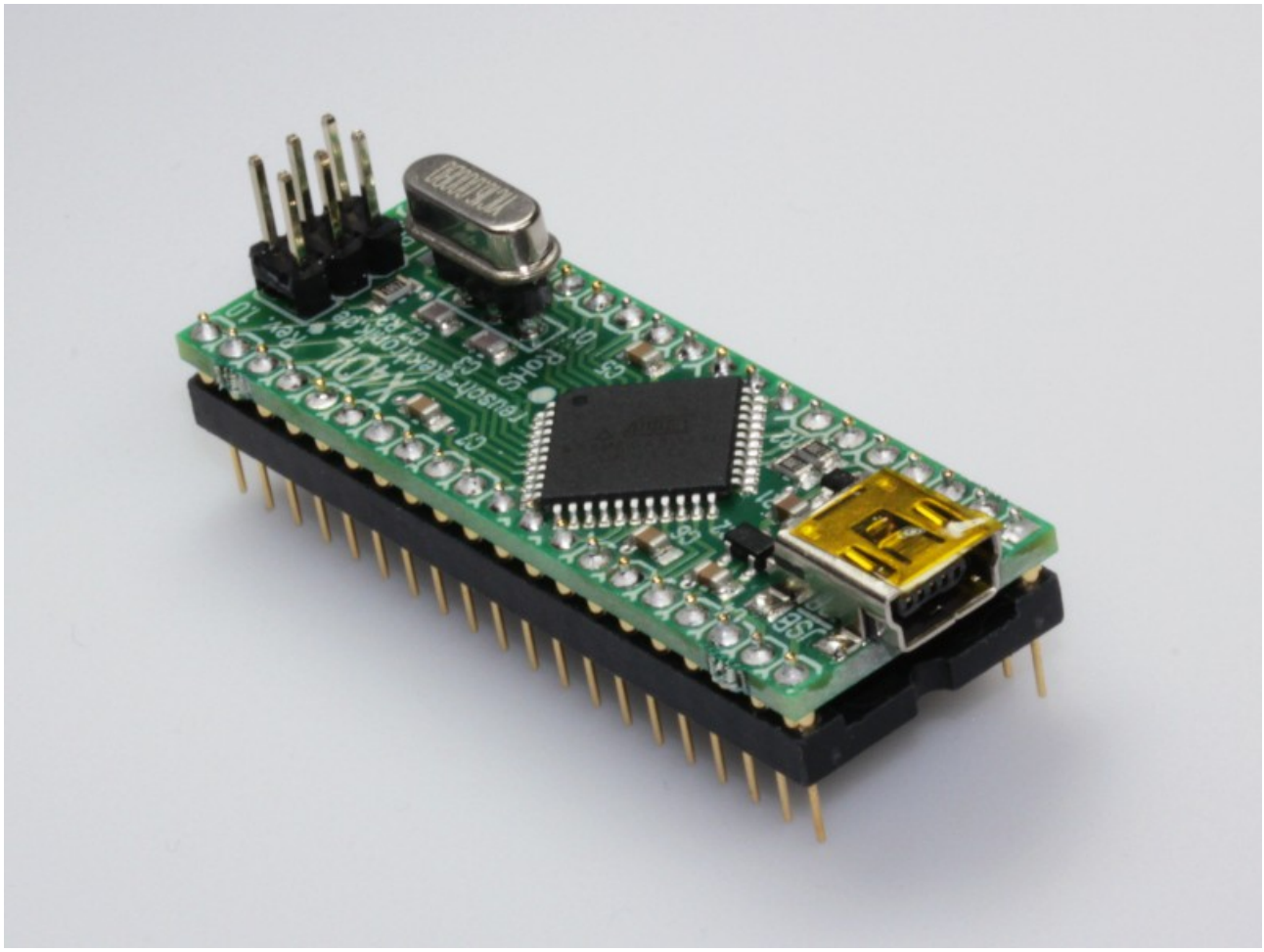
X4DIL

AVR USB Module

Rev. 2.0
Documentation Rev. 11



Reusch Elektronik

© 2012 Reusch Elektronik, Dipl.-Ing. (FH) Rainer Reusch

www.reusch-elektronik.de

http://products.reworld.eu/x4dil.htm

File: X4DIL_Manual
Created: 2012-03-07
Changed: 2012-09-12

Table of Contents

1. Introduction....1
   1.1 Assembly Variations....1
   1.2 Oscillator....1
   1.3 Introduction Example....2
2. Technical Information....3
   2.1 Schematic....3
   2.2 PDI Pin Assignment....4
   2.3 Dual-Inline-Socket Pin Assignment....4
   2.4 Technical Data....5
   2.5 Dimensions....5
3. Application Notes....6
   3.1 Power Supply Options....6
   3.2 Memory Programming....7
   3.3 Reflashing the bootloader....8
4. Concluding Remarks....9
   4.1 Application Software....9
   4.2 Links....9
   4.3 Conformity statement....10
   4.4 Statement and Disclaimer....10
   4.5 Technical Support....10

Please note:

This document refers to revision 2.0 of the X4DIL 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

© 2012 Reusch Elektronik, Dipl.-Ing. (FH) Rainer Reusch

This document is protected by copyright law. It is prohibited to copy or distribute without permission of Reusch Elektronik.

1. Introduction

X4DIL is a tiny interface module for an Atmel AVR micro controller with USB interface in TQFP44 casing. The module fits to an ordinary 40 pin dual-in-line socket. The X4DIL module is helpful in experiments, development and prototyping with USB-AVR 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 ESD protected USB interface (mini USB plug) is on board.

The module contains a 6 pin header for in-system-programming and debugging (Atmel 6 pin PDI standard). These pins are also available by the dual-in-line connector pins.

The X4DIL 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 two assembly versions:

- as “X4DIL-ATXMEGA32A4U”, assembled with ATXMEGA32A4U from Atmel
- as “X4DIL-ATXMEGA128A4U”, assembled with ATXMEGA128A4U from Atmel

The printed circuit board is also available “naked” (not assembled). It is suitable for the following Atmel microcontrollers in TQFP44 casing:

- ATXMEGA16AU4-AU
- ATXMEGA32AU4-AU
- ATXMEGA64AU4-AU
- ATXMEGA128AU4-AU

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 components in wired technology (no SMD). It might be necessary, to remove the two 22pF capacitors (C2 and C3), if a 3 pin ceramic resonator is used.

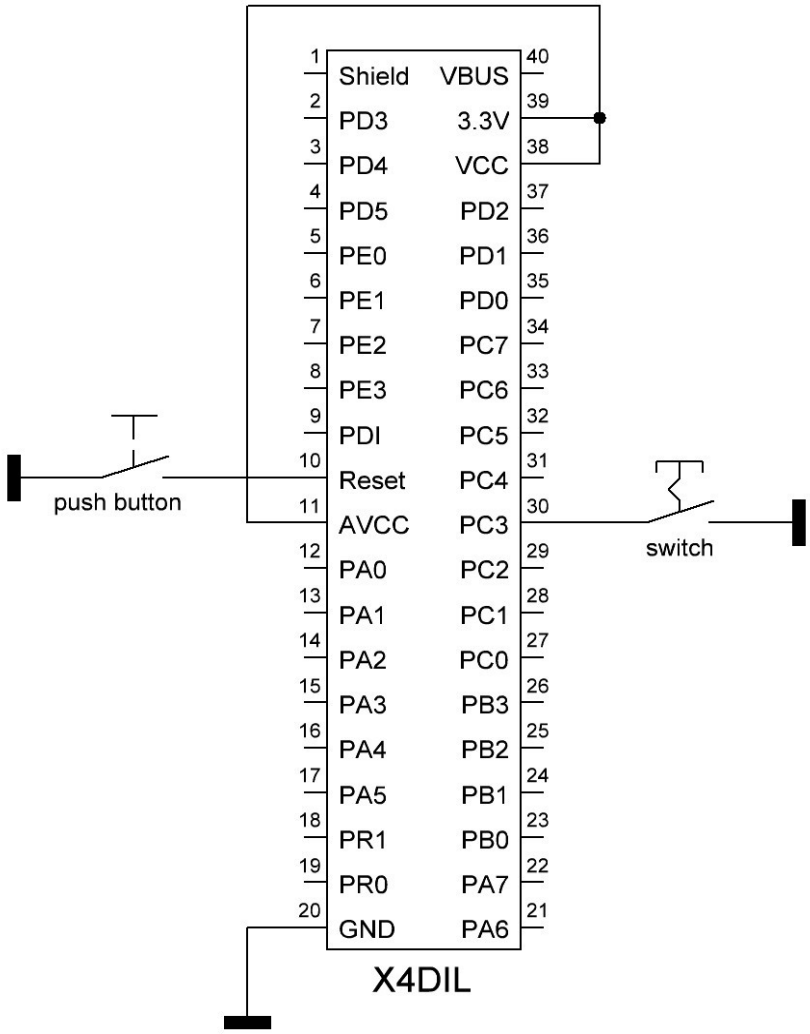
The XTAL pins are also accessible at pin 18 (XTAL1, input) and pin 19 (XTAL2, output). An external clock source can be connected to pin 18.

The controller contains an internal 48MHz RC oscillators. This oscillator can be synchronized with the USB bus clock frequency for usage as clock for the USB engine. Therefor no or any crystal clock frequency can be used to drive the core. If the USB engine is clocked by the crystal, the following frequencies are recommended:

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

1.3 Introduction Example

The wiring example as shown, supplies the module by USB and the onboard 3.3V regulator. Programming can be done via USB, bootloader and “Atmel FLIP”.



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 “Atmel FLIP” to program your firmware application. After programming, the “boot” switch has to be changed to “open” state and a “reset” has to be executed (push of the $\overline{\text{Reset}}$ button). After that or when the device is plugged again, it will run with your programmed firmware.

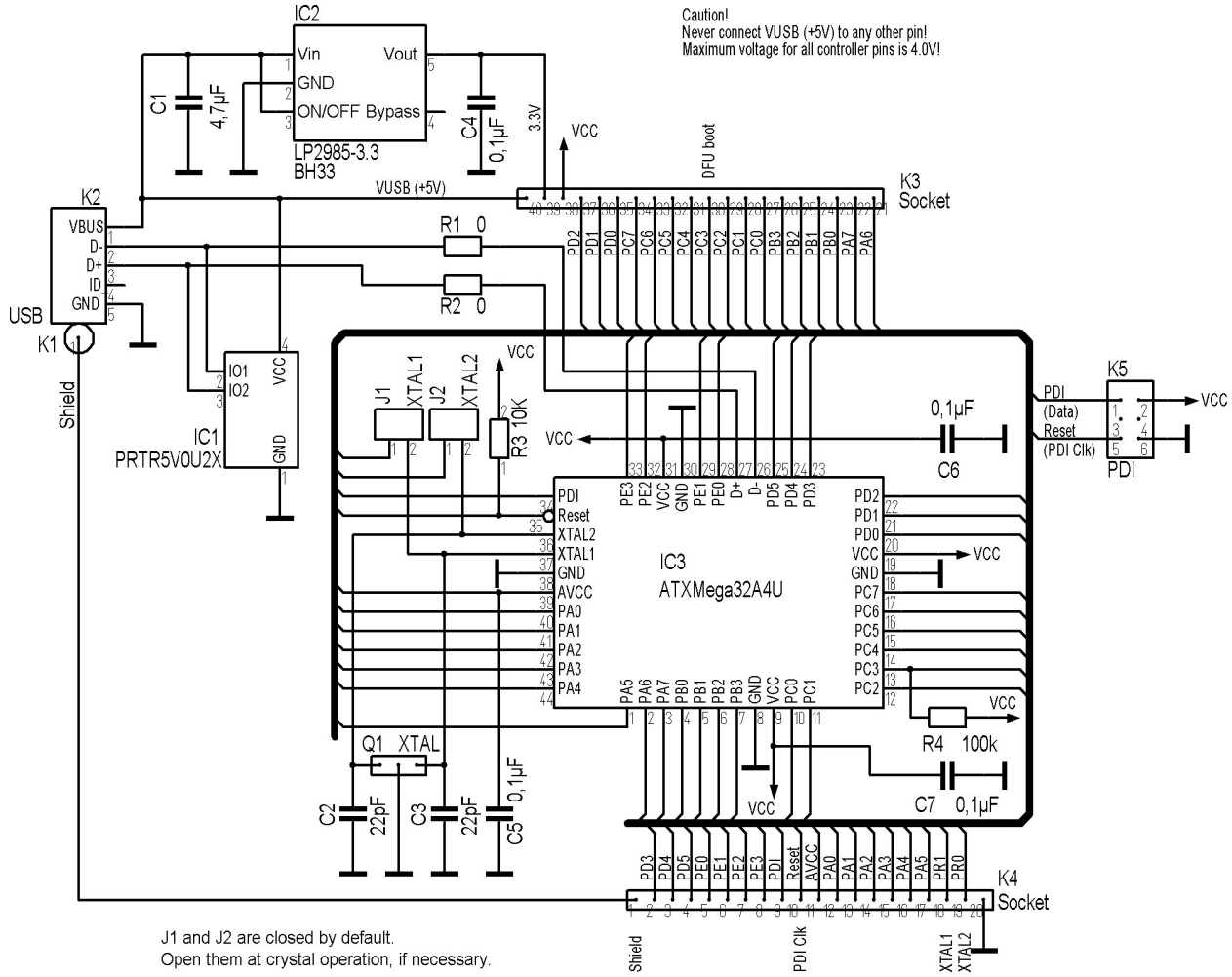
To relaunch the bootloader, close the “boot” switch and push the $\overline{\text{Reset}}$ button once.

Please note:

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

2. Technical Information

2.1 Schematic



Hint:

Bootloader is launched after reset, if DFU bootloader pin is set to high level or kept open (pulled up by R4). Set DFU bootloader pin to low, to launch application.

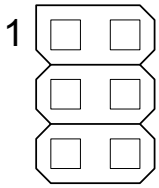
Note:

R4 doesn't exist at PCB revision 1.0!

2.2 PDI Pin Assignment

The in-system programming and debugging adapter is a 6 pin header, which corresponds with the 6 pin Atmel PDI connector. Therefore the usual Atmel in-system programmers (such as “AVR ISP MKII” or “JTAGICE3”) are suitable.

Pin	Function
1	PDI Data
3	NC
5	PDI Clock



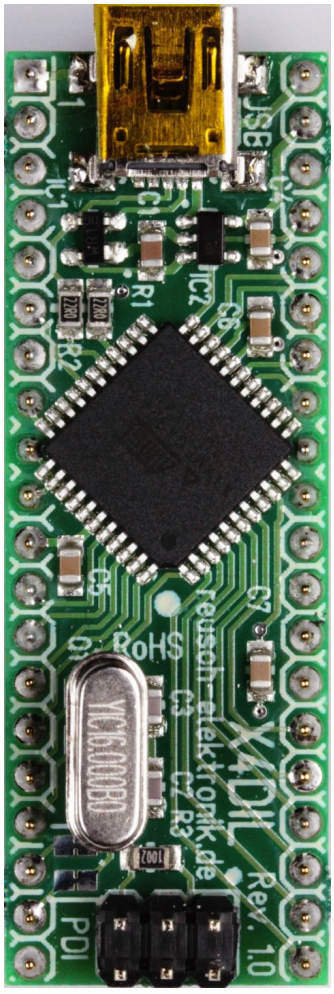
Function	Pin
VCC	2
NC	4
GND	6

Note for AVR JTAG MKII compatible devices (clones, such as the device from Kee Electronics): If it isn't possible, to get a connection to the device, a pull down resistor (about 100kΩ) at the PDI data line may help.

2.3 Dual-Inline-Socket Pin Assignment

The pin assignment of the dual-inline connector.

Pin	Function
1	Shield (USB connector)
2	PD3 (OC0D, TxD0))
3	PD4 (OC1A, $\overline{SS}$)
4	PD5 (OC1B, XCK1, MOSI)
5	PE0 (OC0A, SDA)
6	PE1 (OC0B, XCK0, SCL)
7	PE2 (OC0C, RxD0)
8	PE3 (OC0D, TxD0)
9	PDI (data)
10	$\overline{Reset}$ (PDI clock)
11	AVCC
12	PA0 (ADC0, AC0, AREF)
13	PA1 (ADC1, AC1)
14	PA2 (ADC2, AC2)
15	PA3 (ADC3, AC3)
16	PA4 (ADC4, AC4)
16	PA5 (ADC5, AC5)
18	PR1 (XTAL1, TOSC1)
19	PR0 (XTAL2, TOSC2)
20	GND



Function	Pin
VUSB (5V output)	40
3.3V (output)	39
VCC (3.3V input)	38
PD2 (OC0C, RxD0)	37
PD1 (OC0B, XCK0)	36
PD0 (OC0A)	35
PC7 (OC0DHS, TxD1, SCK, clk_{PER})	34
PC6 (OC0DLS, RxD1, MISO, clk_{RTC})	33
PC5 (OC0CHS, OC1B, XCK1, MOSI)	32
PC4 (OC0CLS, OC1A, $\overline{SS}$)	31
PC3 (OC0D, OC0BHS, TxD0, boot)	30
PC2 (OC0C, OC0BLS, RxD0)	29
PC1 (OC0B, OC0AHS, XCK0, SCL)	28
PC0 (OC0A, OC0ALS, SDA)	27
PB3 (ADC11, DAC1)	26
PB2 (ADC10, DAC0)	25
PB1 (ADC9)	24
PB0 (ADC8, AREF)	23
PA7 (ADC6, AC6, AC0OUT)	22
PA6 (ADC6, AC6, AC1OUT)	21

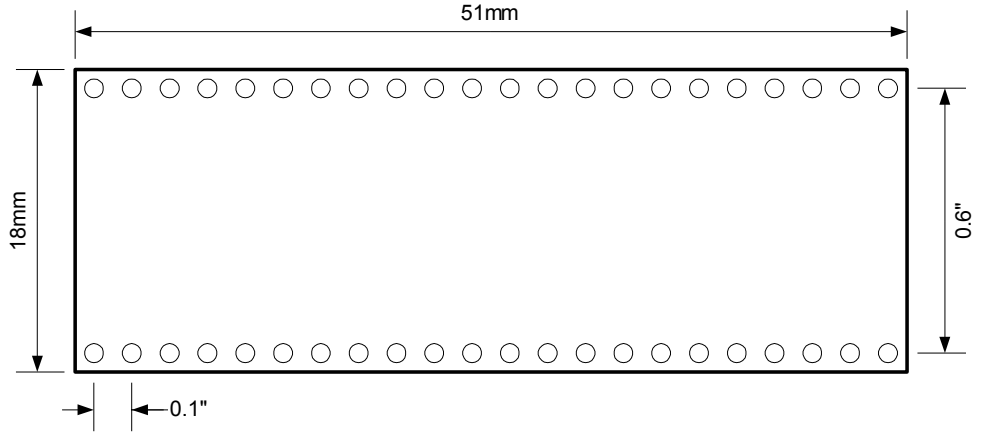
For more information about the alternative pin usage, please refer to the Atmel data sheets. **Warning!** Never connect VUSB (pin 40) to any other pin of the device!

2.4 Technical Data

Supply Voltage	1.6V to 3.6V
Supply Current	<20mA (depends on application)
onboard voltage regulator	3.3V / 100mA
Clock Frequency	up to 32MHz (8, 12 or 16MHz are necessary, if USB engine is clocked by crystal)
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.

2.5 Dimensions



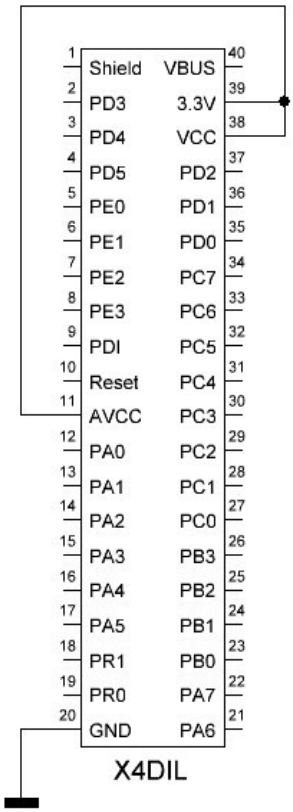
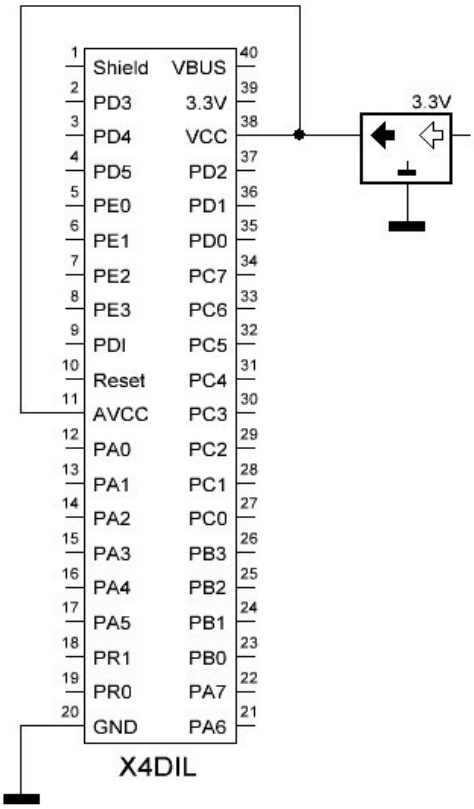
3. Application Notes

3.1 Power Supply Options

The module can be supplied by USB bus power with the on board 3.3V regulator or it can be self powered by an external voltage regulator. The USB supply voltage (+5.0V) is available at pin 40. **Please note:** The controller allows voltages up to 3.6V. Never connect VUSB (pin 40) to any other pin of the module!

For bus powered supply (usage of the onboard voltage regulator), only a few connections are necessary.

For self powered supply a stabilized supply voltage of 3.3V (1.6V to 3.6V) is required. Use the circuitry as shown. Note, that the supply is connected to VCC and AVCC. The 3.3V output (pin 39) is leaved open!

bus powered	self powered
	

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

Please note: The VBUS pin (pin 40) outputs +5.0V from the USB supply voltage. Never connect this pin to any other pin of the X4DIL device! The controller can handle voltages up to 3.6 volts only!

3.2 Memory Programming

The Atmel microcontroller on the X4DIL module is shipped with Atmels DFU bootloader. So there are three ways, to program the flash memory and EEPROM of the microcontroller.

3.2.1 Bootloader and Flip

The USB bootloader and the Software “Atmel Flip” (for *Microsoft Windows* ®) can be used, to program the microcontroller over the USB interface without special programming device. Install the “Flip” software package, before connecting the module in bootloader mode first time. The software requires the *LibUSB-Win32* drivers, which comes with the Atmel Flip installation package.

The USB bootloader will be launched, when the “boot” pin (pin 30 of X4DIL module) is set to low level (0 Volt) and a reset is done (low pulse at pin 10 of the X4DIL module). The module has to be connected to the personal computer (running “Microsoft Windows”) via USB. Note: While the DFU bootloader is running, the module is clocked by the internal 48MHz RC oscillator. That means, no external crystal or clock is necessary for programming.

The module will be detected as Atmel device. The driver installation dialogue will be shown at the first time. Chose “automatic installation”, if the LibUSB-Win32 drivers are already installed. If not, chose “manual installation” and select the driver path of the FLIP installation directory (e.g. “C:\programs\Atmel\Flip”).

Launch the “FLIP” application. Chose the device (ATXMEGA32A4U or 128A4U) and the hex file for programming, finally open an USB connection. Now the microcontroller can be erased, programmed and verified. In a second programming step, the EEPROM can be programmed. Finally push the “start application” button, to close the connection with the device.

To launch the programmed firmware, set the “boot” pin to high level (or keep open) and reset the controller (low pulse at pin 10). It might be necessary, to replug the device (turn off and on the power) for proper USB operation! Note: It isn't possible, to delete the bootloader or modify any fuses with “FLIP”.

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

3.2.2 In-System-Programming

The X4DIL module contains an Atmel PDI connector for in-system-programming. An Atmel PDI compatible programming device (such as “AVRISP MKII”) is required. The advantages of this programming method:

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

Note: This programming method allows a “chip erase”. That means, the bootloader can be erased (the flash memory space can be used for the application). On the other hand, the

device no longer can be programmed with “FLIP”! If a firmware update should be possible later without programmer, the DFU bootloader has to be reprogrammed by this method.

3.2.3 Debugging

The PDI interface is also suitable for debugging. It is a replacement for JTAG. The ATXMEGAxxA4U controllers doesn't have a JTAG interface. The PDI interface can be accessed by the 6 pin header. All pins of this header are also present on the dual in line connector pins (pin 9 and 10).

For debugging, JTAG adapters with PDI support are required. The original Atmel “AVR JTAG MKII”, fully compatible clones (such as the device from Kee Electronics) and Atmels “JTAGICE3” are suitable.

Note for Atmels new “JTAGICE3”: Firmware versions older than V1.34 (delivered until February 2012) doesn't support ATXMEGAxxA4U controllers properly! It is strictly recommended, to update your JTAGICE3 device!

Note for AVR JTAG MKII compatible devices (clones, such as the device from Kee Electronics): If it isn't possible, to get a connection to the device, a pull down resistor (about 100kΩ) at the PDI data line may help.

3.2.4 High Voltage Programming

A “high voltage programming” of the device is not supported!

3.3 Reflashing the bootloader

After a “chip erase” (e.g. when the controller was programmed via PDI) it is necessary, to reflash the DFU bootloader, if further programming should be possible by USB and “FLIP”.

At first, download the DFU bootloader files from the Atmel homepage. Try the following link

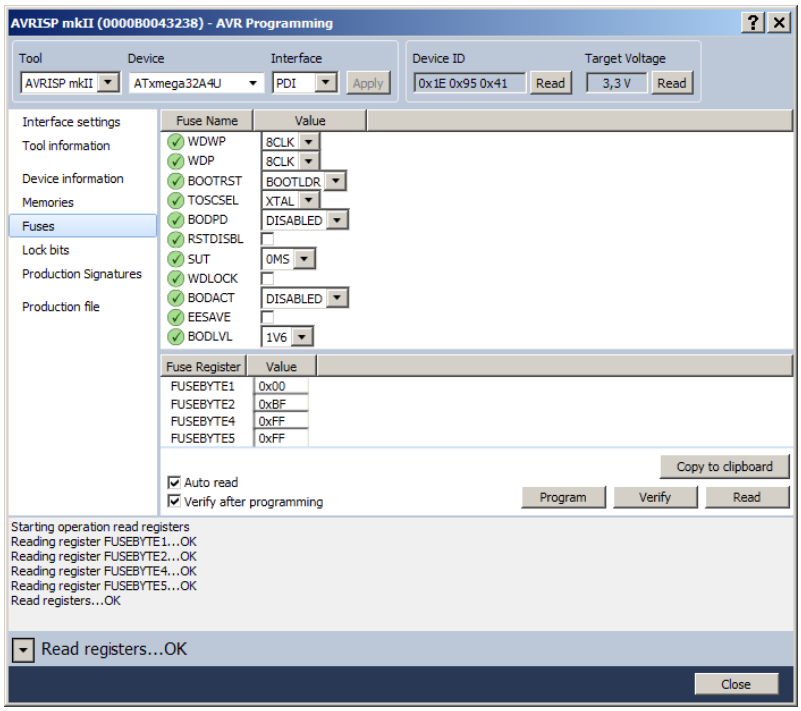
http://www.atmel.com/Images/AVR1916.zip

or search for “ATXMEGA32A4U DFU files” on the Atmel homepage (www.atmel.com). Unzip the file and look for the file “atxmega32a4u_bootloader_usb_dfu_102.hex” (newer files have a higher version number) in the subdirectory “XMEGA_bootloaders_v102\binaries\atxmega32a4u”. For ATXMEGA128A4U use the corresponding file.

Launch AVR Studio (V5.1 or higher) and open the “AVR programming” dialogue. Program the hex file into the flash memory. In a second step, it might be necessary, to modify the fuses. The following screenshot shows the factory default fuse settings, except the fuse *BOOTRST*. This fuse has to be set to “BOOTLDR”, otherwise the bootloader never launches!

After programming the bootloader, it might be necessary, to reset the controller.

Note: The DFU bootloader is running with the internal 48MHz RC oscillator. For this reason, no external crystal or clock generator is necessary, while the bootloader is launched.



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/x4dil.htm

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

Atmel AVR Tools and Software:
http://www.atmel.com/dyn/products/tools.asp?category_id=163&family_id=607&subfamily_id=760

Dean Cameras LUFA (Lightweight USB Framework for AVRs, formerly known as MyUSB):
http://www.fourwalledcubicle.com/LUFA.php

Teensy USB Development Board:
http://www.pjrc.com/teensy/

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/documents.asp?category_id=163&family_id=607&subfamily_id=760

V-USB, Virtual USB Port for AVR microcontroller:
http://www.obdev.at/products/vusb/index.html

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.

4.5 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!