AVR Stick

AVR USB Module with Auto Reset

Rev. 1.0
Documentation Rev. 10

Reusch Elektronik

www.reusch-elektronik.de

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

File:	AVR_Stick_Manual
Created:	2010-09-10
Changed:	2010-12-09

Contents

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

AVR Stick is a tiny interface module in stick format, containing an Atmel AVR micro controller with USB interface. The device is designed for USB applications, which doesn't need any connectivity to peripherals. For simple interfacing, some controller I/O ports are available on pads. The stick comes in a compact size, suitable for an also available compact casing. It contains an USB-A mail plug for direct plugging into an USB hub.

The stick contains an 8 bit RISC controller from Atmel, either *AT90USB162* or *ATMEGA32U2*. This controller is clocked with 16MHz. The USB communication confirms USB 2.0 specification. The controller supports "full speed" (12MBit/s).

The stick will be delivered with the original Atmel bootloader. It can be programmed by the Windows ® software "Atmel Flip" via USB. For special purposes, an Atmel compatible ISP interface is available on solder pads for "in system" programming (a special programming device, such as Atmel AVRISP MKII, is required).

The AVR Stick is very easy to use, because it contains all necessary and time critical components.

1.1 Assembly Variations

The module is available in two assembly versions:

- as "AVR-Stick 162 AR", assembled with AT90USB162 from, with *Auto Reset*
- as "AVR-Stick U2 AR", assembled with ATMEGA32U2, with *Auto Reset*

Both assembly variations are also available without *Auto Reset* Feature.

1.2 Usage

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 upload your firmware application. After programming, the device has to be unplugged from USB (turn off power supply) and the "boot" jumper ($\overline{\text{HWB}}$ port) has to be changed to "open" state.

When the device is plugged again, it will run with your uploaded firmware.

To reenter the bootloader, close the "boot" jumper ($\overline{\text{HWB}}$ port) and replug the stick. Only assembly version without *Auto Reset (AR)* feature: It is necessary, to reset the controller. For this reason, short the $\overline{\text{Reset}}$ (pin 5) and the *Gnd* pin (pin 6) of the ISP connector K4.

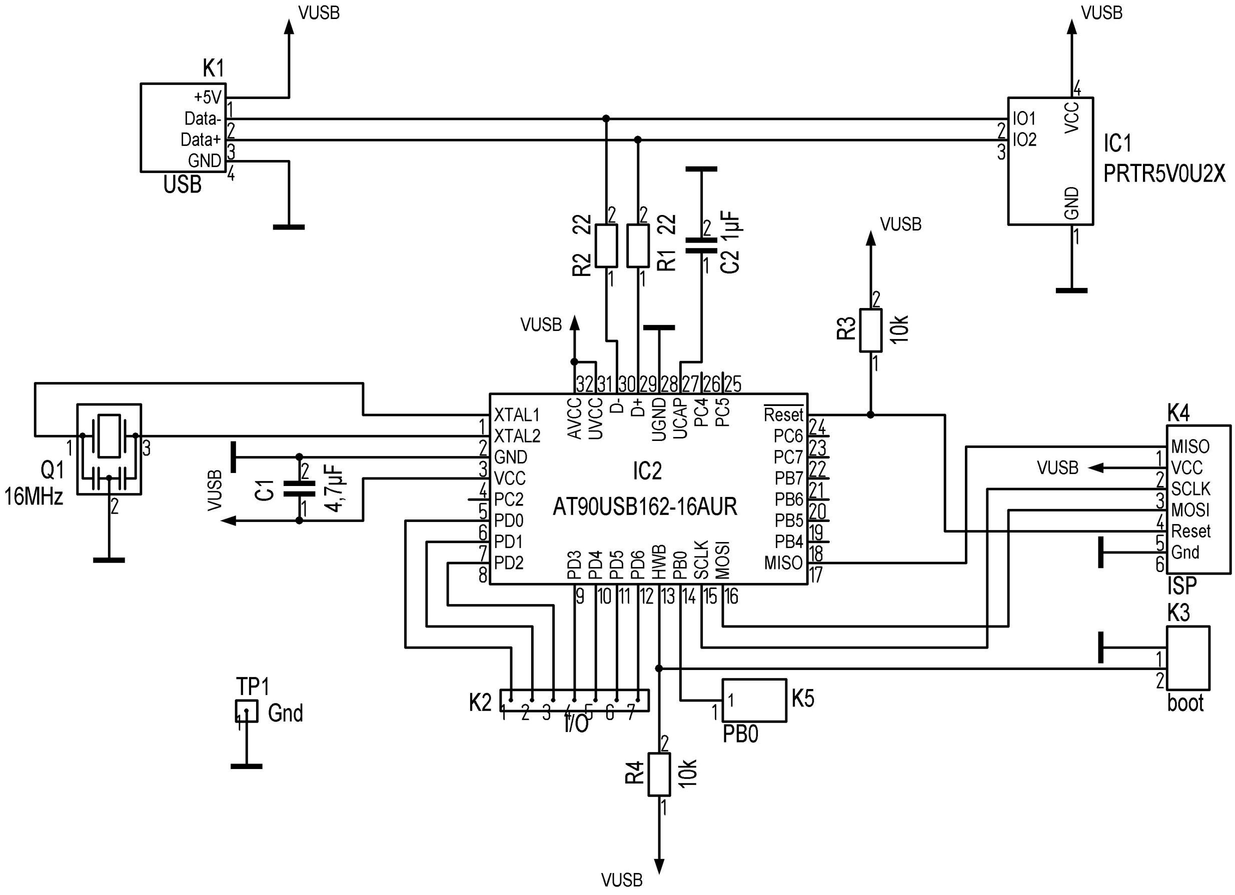
Please note:

Setting or removing the "boot" jumper while stick is plugged, doesn't have any effect! You have to reset the controller, either through replugging or shorten the reset pad with ground for a moment.

The device also can be programmed via *ISP* interface and ISP programmer. For more information read the application note chapter, please.

2. Technical Information

2.1 Schematic



Only assembly version with *Auto Reset (AR)* feature: A ceramic capacitor with 100nF is connected between pin 5 (Reset) and pin 6 (Gnd) of the ISP connector (K4).

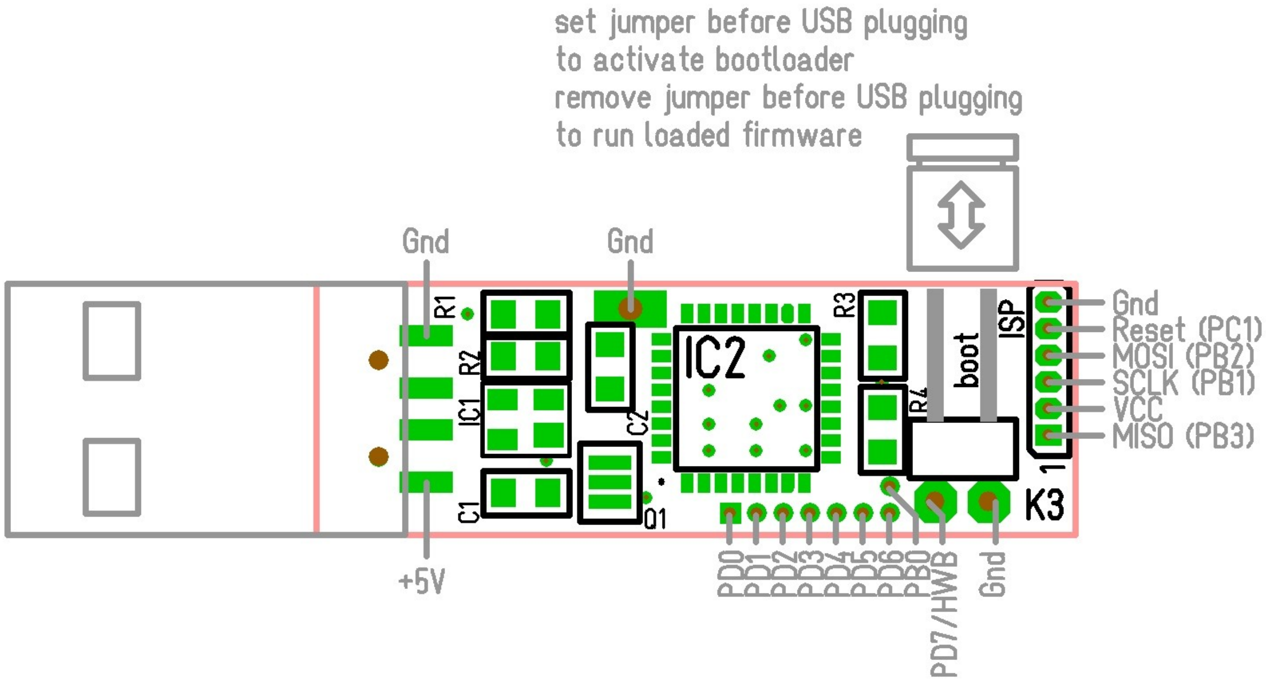
2.2 ISP Pin Assignment

The in-system programming adapter is a 6 pole pad array (K4), which corresponds in order with the 6 pin Atmel ISP connector. Therefore the usual Atmel in-system programmers (such as "AVR ISP MKII") are suitable.

Pin	Function	Direction	Atmel ISP connector
6 *	Gnd		6
5 *	Reset	input	5
4	MOSI	input	4
3	SCLK	input	3
2	VCC	output	2
1	MISO	output	1

*) Only AR version: A 100nF capacitor is connected between this two pins.

2.3 Board of the Stick



2.4 Alternative Pin Functions

The following table shows the alternative function of each available pin of the controller.

Port	Alternative Functions	Connector
PD0	OC.0B, INT0	K2-1
PD1	AIN0, INT1	K2-2
PD2	RXD1, AIN1, INT2	K2-3
PD3	TXD1, INT3	K2-4
PD4	INT5, AIN3	K2-5
PD5	XCK, AIN4, PCINT12	K2-6
PD6	$\overline{\text{RTS}}$, AIN5, INT6	K2-7
$\overline{\text{HWB}}$	PD7, $\overline{\text{CTS}}$, AIN6, T0, INT7 *	K3-2
PB0	$\overline{\text{SS}}$, PCINT0	K5
PB1	SCLK, PCINT1	K4-3
PB2	PDI, MOSI, PCINT2	K4-4
PB3	PDO, MISO, PCINT3	K4-1
$\overline{\text{Reset}}$	PC1, dW **	K4-5

*) *HWB* fuse has to be cleared via ISP programming. Disables ability of firmware upload via USB!

**) *RSTDISBL* fuse has to be set via ISP programming. Disables ability of programming via ISP!

2.5 Technical Data

Supply Voltage	4.75V to 5.25V (USB supply voltage range)
Supply Current	<30mA (depends on application)
Clock Frequency	16MHz (±0.5%)
PCB dimensions	36,2mm x 12mm
Stick in casing dimensions	61.5mm x 16.2mm x 8.7mm
Stick in casing weight	5 gramms
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.

3. Application Notes

3.1 Memory Programming

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

3.1.1 Bootloader and Flip

The USB bootloader, flashed as factory default, 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 executed, when the $\overline{\text{HWB}}$ pin (K3 closed) is set to low level (0 Volt) and a reset is done (pin 5 and 6 of K4 shorten for a moment). This manual reset is not necessary for the *Auto Reset (AR)* featured version of this device. The module has to be connected to the personal computer (running “Microsoft Windows”) via USB.

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. Start the “Flip” application. Chose the device (ATMEGA32U2 or AT90USB162) and the hex file for programming. Finally open an USB connection in “Flip”. Now the microcontroller can be erased, programmed and verified.

The $\overline{\text{HWB}}$ pin has to be set to high level (K3 opened), *before* the loaded application will be started. It is 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

3.1.2 In-System-Programming

The AVR Stick contains an Atmel ISP connector for in-system-programming. An Atmel ISP 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”
- no bootloader required
- the only way, to reflash the bootloader!

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

Note for *Auto Reset (AR)* featured device: The *in system programming* may not work properly, caused by the 100nF capacitor connected to reset pin. It might be necessary, to remove this capacitor, if problems occurs.

3.1.3 Debug Wire

The alternative function for the $\overline{\text{Reset}}$ pin is the “debug Wire” pin. The dW pin can be used for programming and debugging within the development environment (AVR Studio). A JTAG device with “debug Wire” support is required.

Note for *Auto Reset (AR)* featured device: This programming method can't be applied!

3.1.4 High Voltage Programming

Because not all necessary pins of the controller are available, a high voltage programming isn't supported.

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/avrstick.htm

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

Atmel AVR Tools and Software:
http://www.atmel.com/dyn/products/tools_mcu.asp?family_id=607

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/datasheets_mcu.asp?family_id=607

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 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!