This is the README file for USBasp.

USBasp is a USB in-circuit programmer for Atmel AVR controllers. It simply

consists of an ATMega88 or an ATMega8 and a couple of passive components.

The programmer uses a firmware-only USB driver, no special USB controller

is needed.

Features:

- Works under multiple platforms. Linux, Mac OS X and Windows are tested.

- No special controllers or smd components are needed.

- Programming speed is up to 5kBytes/sec.

- SCK option to support targets with low clock speed (< 1,5MHz).

- Planned: serial interface to target (e.g. for debugging).

LICENSE

USBasp is distributed under the terms and conditions of the GNU GPL version

2 (see "firmware/usbdrv/License.txt" for details).

USBasp is built with V-USB driver by OBJECTIVE DEVELOPMENT GmbH. See

"firmware/usbdrv/" for further information.

LIMITATIONS

Hardware:

This package includes a circuit diagram. This circuit can only be used for

programming 5V target systems. For other systems a level converter is needed.

Firmware:

The firmware dosn't support USB Suspend Mode. A bidirectional serial

interface to slave exists in hardware but the firmware doesn't support it yet.

USE PRECOMPILED VERSION

Firmware:

Flash "bin/firmware/usbasp.atmega88.xxxx-xx-xx.hex" or

"bin/firmware/usbasp.atmega8.xxxx-xx-xx.hex" to the used controller with a

working programmer (e.g. with avrdude, uisp, ...). Set jumper J2 to activate

USBasp firmware update function.

You have to change the fuse bits for external crystal (see "make fuses").

# TARGET=atmega8 HFUSE=0xc9 LFUSE=0xef

# TARGET=atmega48 HFUSE=0xdd LFUSE=0xff

# TARGET=atmega88 HFUSE=0xdd LFUSE=0xff

Windows:

Start Windows and connect USBasp to the system. When Windows asks for a

driver, choose "bin/win-driver". On Win2k and WinXP systems, Windows will

warn that the driver is is not 'digitally signed'. Ignore this message and

continue with the installation.

Now you can run avrdude. Examples:

1. Enter terminal mode with an AT90S2313 connected to the programmer:

avrdude -c usbasp -p at90s2313 -t

2. Write main.hex to the flash of an ATmega8:

avrdude -c usbasp -p atmega8 -U flash:w:main.hex

Setting jumpers:

J1 Power target

Supply target with 5V (USB voltage). Be careful with this option, the

circuit isn't protected against short circuit!

J2 Jumper for firmware upgrade (not self-upgradable)

Set this jumper for flashing the ATMega(4)8 of USBasp with another working

programmer.

J3 SCK option

If the target clock is lower than 1,5 MHz, you have to set this jumper.

Then SCK is scaled down from 375 kHz to about 8 kHz.

BUILDING AND INSTALLING FROM SOURCE CODE

Firmware:

To compile the firmware

1. install the GNU toolchain for AVR microcontrollers (avr-gcc, avr-libc),

2. change directory to firmware/

3. run "make main.hex"

4. flash "main.hex" to the ATMega(4)8. E.g. with uisp or avrdude (check

the Makefile option "make flash"). To flash the firmware you have

to set jumper J2 and connect USBasp to a working programmer.

You have to change the fuse bits for external crystal, (check the Makefile

option "make fuses").

Software (avrdude):

AVRDUDE supports USBasp since version 5.2.

1. install libusb: http://libusb.sourceforge.net/

2. get latest avrdude release: http://download.savannah.gnu.org/releases/avrdude/

3. cd avrdude-X.X.X

5. configure to your environment:

./bootstrap (I had to comment out the two if-blocks which verify the

installed versions of autoconf and automake)

./configure

6. compile and install it:

make

make install

Notes on Windows (Cygwin):

Download libusb-win32-device-bin-x.x.x.x.tar.gz from

http://libusb-win32.sourceforge.net/ and unpack it.

-> copy lib/gcc/libusb.a to lib-path

-> copy include/usb.h to include-path

cd avrdude

./configure LDFLAGS="-static" --enable-versioned-doc=no

make

Notes on Darwin/MacOS X:

after "./configure" I had to edit Makefile:

change "avrdude\_CPPFLAGS" to "AM\_CPPFLAGS"

(why is this needed only on mac? bug in configure.ac?)

Notes on Linux:

To use USBasp as non-root, you have to define some device rules. See

bin/linux-nonroot for an example.

FILES IN THE DISTRIBUTION

Readme.txt ...................... The file you are currently reading

firmware ........................ Source code of the controller firmware

firmware/usbdrv ................. AVR USB driver by Objective Development

firmware/usbdrv/License.txt ..... Public license for AVR USB driver and USBasp

circuit ......................... Circuit diagram in PDF and EAGLE format

bin ............................. Precompiled programs

bin/win-driver .................. Windows driver

bin/firmware .................... Precompiled firmware

bin/linux-nonroot ............... Linux device rule file

MORE INFORMATION

For more information on USBasp and it's components please visit the

following URLs:

USBasp .......................... http://www.fischl.de/usbasp/

Firmware-only V-USB driver ...... http://www.obdev.at/products/vusb/

avrdude ......................... http://www.nongnu.org/avrdude/

libusb .......................... http://libusb.sourceforge.net/

libusb-win32 .................... http://libusb-win32.sourceforge.net/

2011-05-28 Thomas Fischl <tfischl@gmx.de>

http://www.fischl.de