

USBASP

Zur In-System-Programmierung (ISP) der AVR Mikrocontroller hatte ich früher den STK200-Dongle von Kanda am Parallelport genutzt.

http://www.mikrocontroller.net/articles/AVR_In_System_Programmieradapter

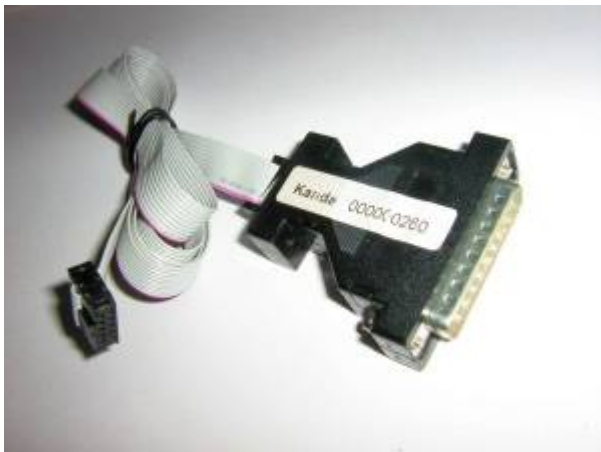
Jetzt benutze ich einen USBASP-Adapter (s. unten).

<https://www.fischl.de/usbasp/>

firmware that gives the USBasp full compatibility with ATmega2560:

<https://github.com/PeterVH/usbasp/tree/v1.06>

STK200-Dongle von Kanda



Dieser AVR-Programmieradapter (Bj 1998) wird am Parallelport angeschlossen. Er enthält lediglich einen 74HC244-Treiber-IC, die Programmierlogik übernimmt vollständig der PC. s.a.

<http://www.mikrocontroller.net/articles/STK200>.

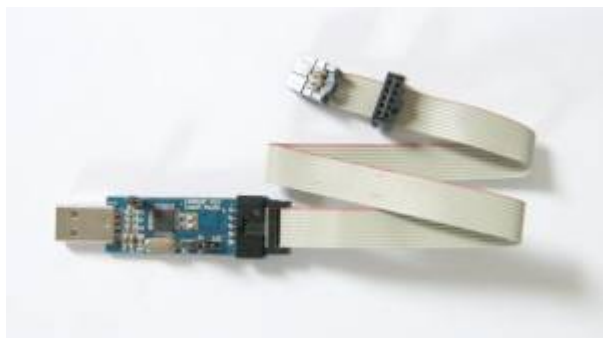
Achtung: Das Kabel ist KEIN 1:1-Kabel, sondern geht über Kreuz. Es ist nur für STK200-Dongle von Kanda gedacht.

Zum Programmieren muss die Schaltung mit dem zu programmierenden Atmel-Controller stromführend sein.

Aufgrund des 74HC244 kann der Controller auch im Betrieb mit dem Programmieradapter verbunden bleiben

Ich hatte mir eine Version namens „USBASP V2.0 Lcsoft Studio“ gekauft. Bei Protostack findet man ein komplettes Userguide mit allen benötigten Unterlagen zu diesem Programmer (AC-PG-USBASP-UG-V2.0).pdf, s.u.).

Achtung: beim beiliegenden Kabel ist rot NICHT Pin1 zugeordnet. Ich habe das für mich korrigiert (Pfofenbuchsen abmontieren und gedreht wieder aufbringen, so dass Pin1 - das kleine Dreieck - am roten Kabel anliegt). Außerdem habe ich zusätzlich eine 6-polige Pfofenbuchse mit aufs Flachbandkabel gebracht. Damit können 6-Pin-ISP-Geräte verbunden werden, wie z.B. Arduino oder Euzebox.



	Kabel rot			6pol. Buchse
MOSI	1	-----+	+-----	1 MISO
VCC	2	---- -----	-----	2 VCC
GROUND	3		+-- -----	3 SCK
TXD	4	+-- -- -----	-----	4 MOSI
RESET	5	----- -- -----	-----	5 RESET
RXD	6		+---	6 GROUND
SCK	7	-----+		
GROUND	8			
MISO	9	-----+		
GROUND	10	-----+		

Unterlagen

s. <http://www.protostack.com/accessories/usbasp-avr-Programmer>

aktuelle Downloads

s. http://www.protostack.com/index.php?main_page=library

User Guide : [http://www.protostack.com/download/Users%20Guide%20\(AC-PG-USBASP-UG-V2.0\).pdf](http://www.protostack.com/download/Users%20Guide%20(AC-PG-USBASP-UG-V2.0).pdf)

Treiber : <http://www.protostack.com/download/USBasp-win-driver-x86-x64-ia64-v1.2.5.zip>

<https://protostack.com.au/download/USBasp-win-driver-x86-x64-v3.0.7.zip>

Der USB-Treiber muss zuerst installiert werden. Im User Guide ist dies ausführlich beschrieben und bebildert. Mit libusb_0.1.12.1\testlibusb-win.exe kann geprüft werden, ob der Treiber richtig installiert ist und der USBASP erkannt wird.

Jumper

Ich habe die beiden Jumper JP2 und JP3 nachgerüstet. (Achtung: Im Original sind die Jumper anders durchnummeriert!)

Mit **JP1** kann die Zielhardware über den USB Port mit Strom versorgt werden (das wird nicht empfohlen, funktioniert aber problemlos). Entweder 5V oder 3.3V, Standard: Offen

JP2 wird nur zum Programmieren des Atmel ATmega8 auf dem USB Atmel SPI Programmieradapter benötigt. Standard: Offen

JP3 wählt die Geschwindigkeit zum Programmieren der Zielhardware aus. Offen (Standard): maximale Geschwindigkeit / Geschlossen: geringe Geschwindigkeit.

Firmware-Update

Khazama(AVRDUDE) gab original immer die Warnung „Error Setting USBASP ISP Clock.“ aus. Abhilfe schafft ein Flashen des USBASP mit der aktuellsten Software und passenden Fuses.

zum Update der Firmware braucht man einen weiteren ISP-Programmierer (z.B. den STK200-Dongle)! Dann JP1 (neue Firmware) und JP2 (langsam) stecken, und Programmieren. Zum korrekten Programmieren gibt es in `usbasp.2011-05-28.tar.gz` geeignete makefiles.

Programmieren der Firmware des USBASP mittels STK200-Dongle von Kanda. Der USBASP braucht dabei Strom, er muss also an einem USB-Port oder ext. USB-Netzteil angeschlossen sein. Außerdem muss JP3 auf +5V gesteckt werden!

```
avrdude -c stk200 -p atmega8 -B 200 -U hfuse:w:0xC9:m -U lfuse:w:0xEF:m  
avrdude -c stk200 -p atmega8 -B 1 -U flash:w:usbasp.atmega8.2011-05-28.hex
```

Hinweise

s.a. <http://www.ulrichradig.de/home/index.php/avr/usb-avr-prog>

Anfängerfehler: target doesn't answer. 1

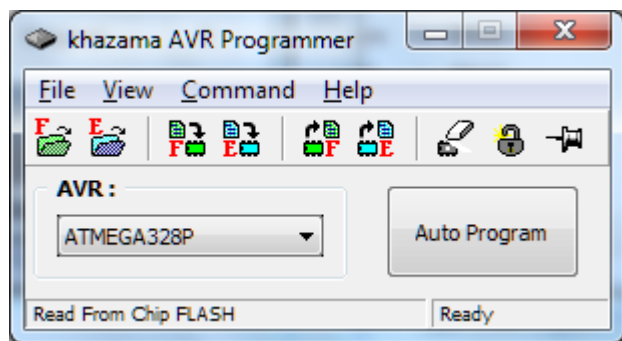
Hier musste JP2 gesteckt werden (bzw. in der Software ein niedrigerer Takt ausgewählt werden), damit überhaupt eine Kommunikation mit dem tiny2313 auf dem Pollin-Spiel zustande kam. Der tiny2313 nutzt in dieser Applikation den internen Taktgenerator, das ist für die schnelle Kommunikation vis USBASP einfach zu langsam gewesen.

Software

AVRDUDE unterstützt natürlich auch USBASP. Download unter <http://download.savannah.gnu.org/releases/avrdude/>. Aktuell f. Windows ist derzeit `avrdude-5.11-Patch7610-win32.zip`.

grafische Oberfläche f. AVRDUDE z.B. **Khazama AVR Programmieradapter**

<http://www.khazama.com/project/Programmieradapter/>



• **eXtreme Burner - AVR** von

<http://extremeelectronics.co.in/avr-tutorials/gui-software-for-usbasp-based-usb-avr-programmers/>

Auch **myAVR_ProgTool** von <http://shop.myavr.de/index.php?sp=download.sp.php&suchwort=dl112> unterstützt alle AVRDUDE-Programmieradapter.

Beispiel **AVRDUDE** Kommandozeilentool

```

;-----
;
; Fehler target doesn't answer. 1
; Lösung: Takt verringern, entweder mit JP3 oder per Software mit -B20
;-----
;
d:\temp\avr>avrdude.exe -c usbasp -p attiny2313
avrdude.exe: error: programm enable: target doesn't answer. 1
avrdude.exe: initialization failed, rc=-1
                Double check connections and try again, or use -F to override
                this check.
avrdude.exe done. Thank you.

d:\temp\avr>avrdude.exe -c usbasp -p attiny2313 -B20
avrdude.exe: set SCK frequency to 32000 Hz
avrdude.exe: AVR device initialized and ready to accept instructions
Reading | ##### | 100% 0.03s
avrdude.exe: Device signature = 0x1e910a
avrdude.exe: safemode: Fuses OK
avrdude.exe done. Thank you.

```

Ändern von USBASP auf USB AVR Lab

Alternative Firmware für USBASP

Für jeden Benutzer von USBASP kann ich nur empfehlen, einen Blick auf ein alternatives Firmware-Set zu werfen: **USB AVR Lab** <http://www.ullihome.de/wiki/USB AVR Lab/index>. Diese Software erhöht die Leistungsfähigkeit des USBASP enorm durch einfachstes Wechseln der Firmware bei gleicher

Hardware. Es ist eine Art „Software Defined“-Werkzeug (daher der Name), das als folgendes arbeiten kann:

- AVR Programmierer, entweder mit original USBASP-Firmware, AVRISPMkII, stk500v2 oder JTAGICEmkII Emulation
- JTAG/OpenOCD Schnittstelle (leider: noch alpha)
- generische USB-Digital-I/O-Schnittstelle
- digitaler Protokollanalysator, unterstützt SPI, UART und I2C-Protokolle
- und (eindrucksvoll): Low-Speed-Digital-Oszilloskop (230 kSps).

s. <http://www.loetlabor.org/USBasp>, Abschnitt AVR ISP mit AVRStudio

zu Sourcen lies <http://www.mikrocontroller.net/topic/234004>

From:

<https://hc-ddr.hucki.net/wiki/> - **Homecomputer DDR**

Permanent link:

<https://hc-ddr.hucki.net/wiki/doku.php/elektronik/usbasp?rev=1604056441>

Last update: **2020/10/30 11:14**

