

Arduino mit ISP programmieren

15. August 2018 – DK8XK
21. August 2018 – REV.1.1 (USBtinyISP)

Begriffe: ISP = In System Programmer,
IDE = Integrated-Development-Environment.

Motivation:

Die mit dem ATmega328P (und verwandten) bestückten Platinen bieten nur eine serielle Hardware Schnittstelle die standardmäßig für die Programmierung verwendet wird. Der weiterentwickelte ATmega328P_B_ bietet u.a. eine zweite, ist aber nur selten und entsprechend teuer angeboten. Ein Beispiel wäre "Pololu". Die Möglichkeit "Softserial" ist bei zeitkritischen Anwendungen wegen der vielen Interrupte kritisch und nicht immer anwendbar. Um die eine serielle Schnittstelle frei zu halten wird das Programmierverfahren mit einem ISP-Interface gewählt.

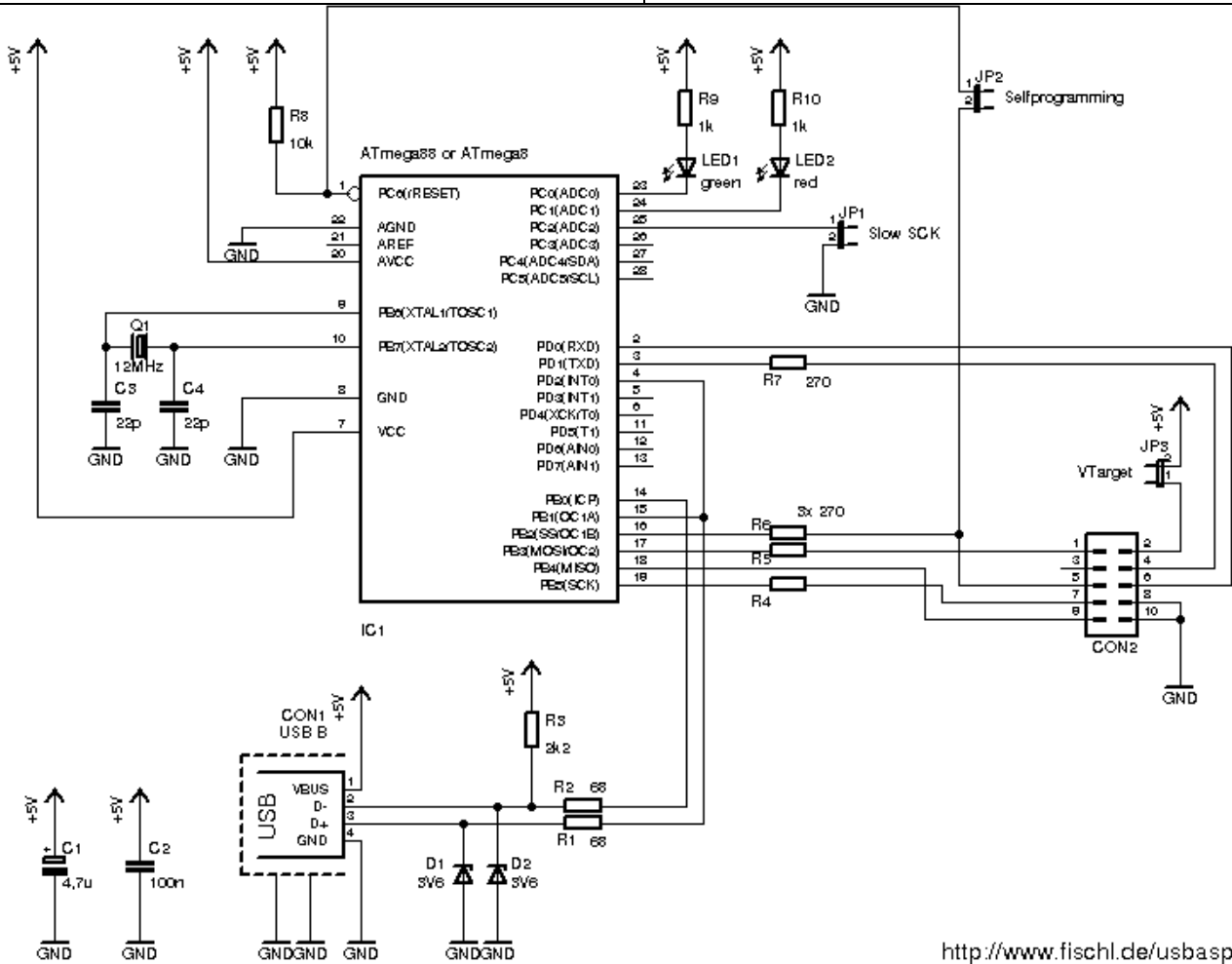
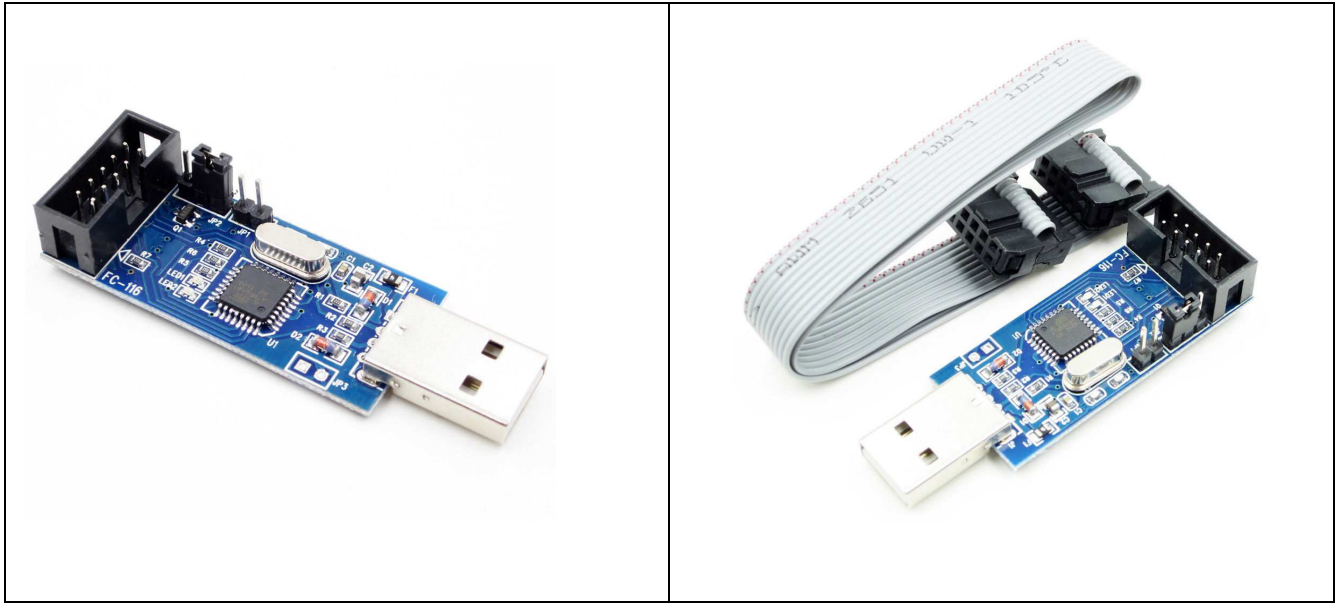
Lösungsansatz:

Eine gute Möglichkeit ist es "Arduino als ISP" zu verwenden. Dabei wird ein Arduino-Board aus der "UNO-Familie" o.ä. zu einem ISP durch das Laden des Beispiel-Sketches 11. "ArduinoISP". An diesen wird dann der zu programmierende Arduino angeschlossen (diverse Hinweise unter dem Suchbegriff "ArduinoISP").

Eine weitere von mir verwendete Lösung ist einen ISP zu verwenden. Ich habe mich für den USBasp entschieden.

Ein freies Projekt: <https://www.fischl.de/usbasp/> . Günstig zu erwerben zB. bei ebay.de unter dem Suchbegriff "USBasp" oder auch "ArduinoISP".

Und so sieht er aus:

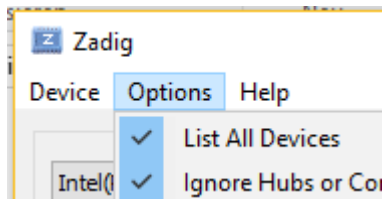


<http://www.fischl.de/usbsp>

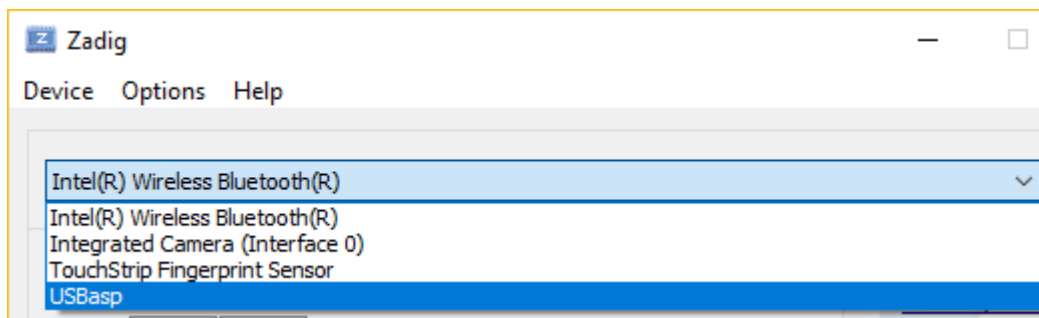
Die Schaltung des USBasp.

Die Einrichtung des ISP:

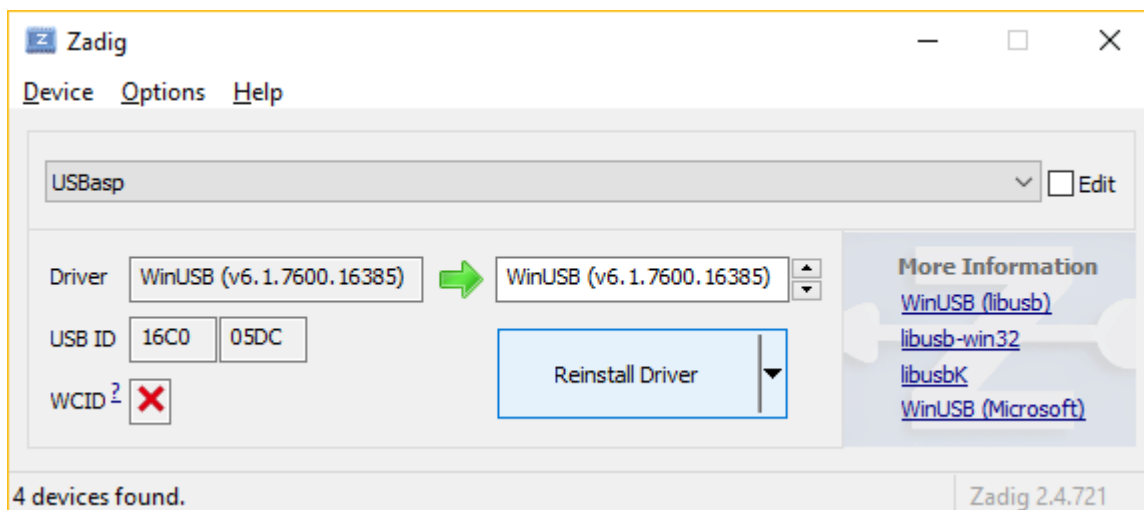
Nach dem Anstecken des USBasp kann Windows nicht den passenden Treiber finden. Hier hilft ein Universalwerkzeug "ZADIG" von hier: <http://zadig.akeo.ie/>. Das heruntergeladene Programm kann ohne Installation gestartet werden.



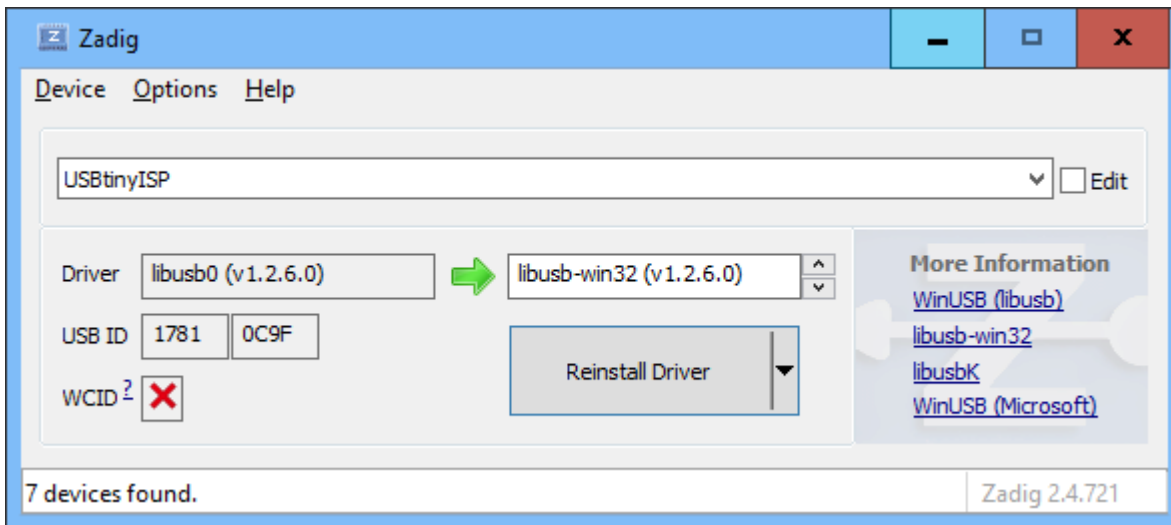
"List All Devices" an hacken und dann USBasp markieren:



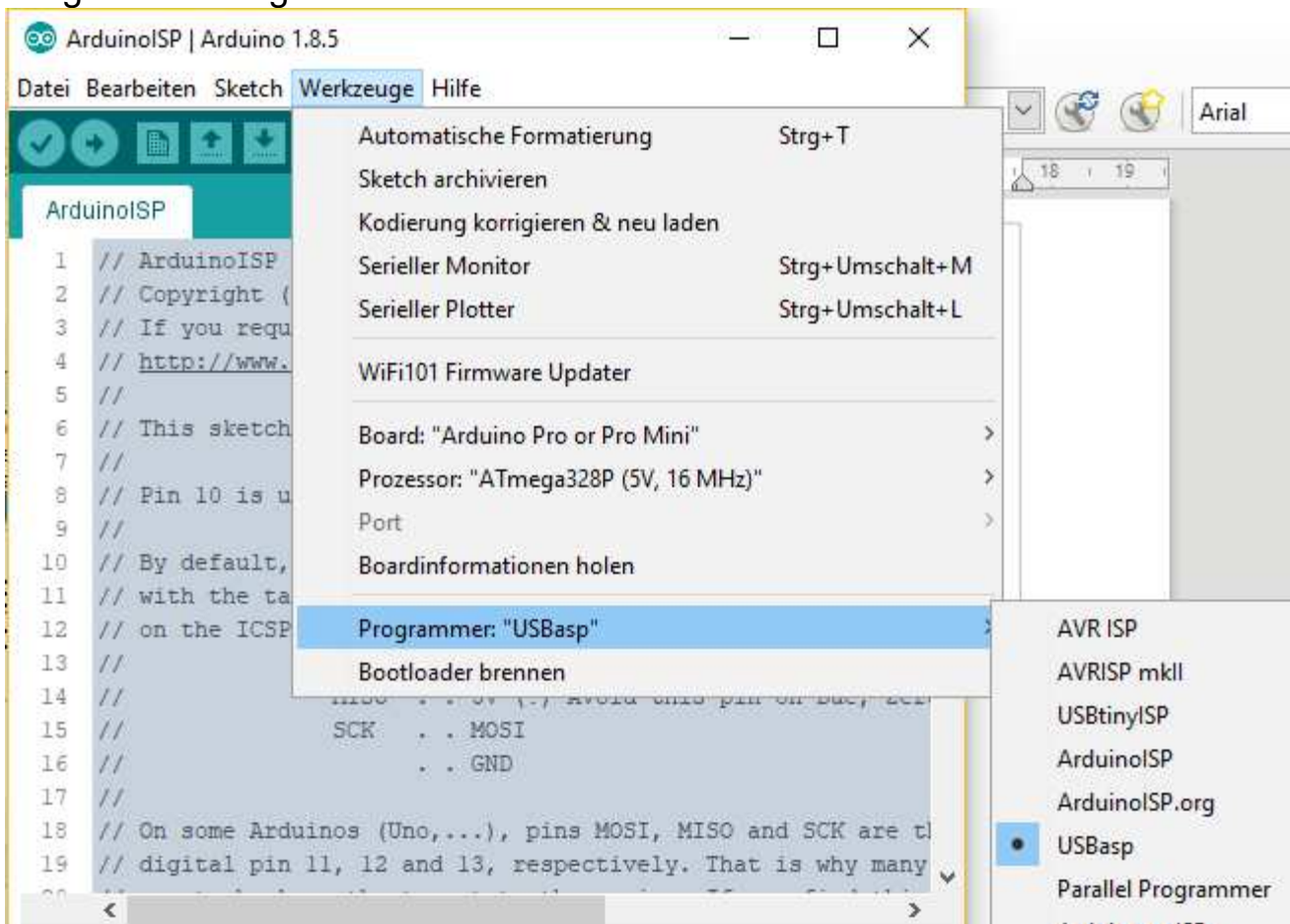
Dann den WinUSB(XXX) – Treiber installieren. Der tat es bei mir. Der vorgeschlagene libusb-win32 wurde von AVRdude nicht gefunden.



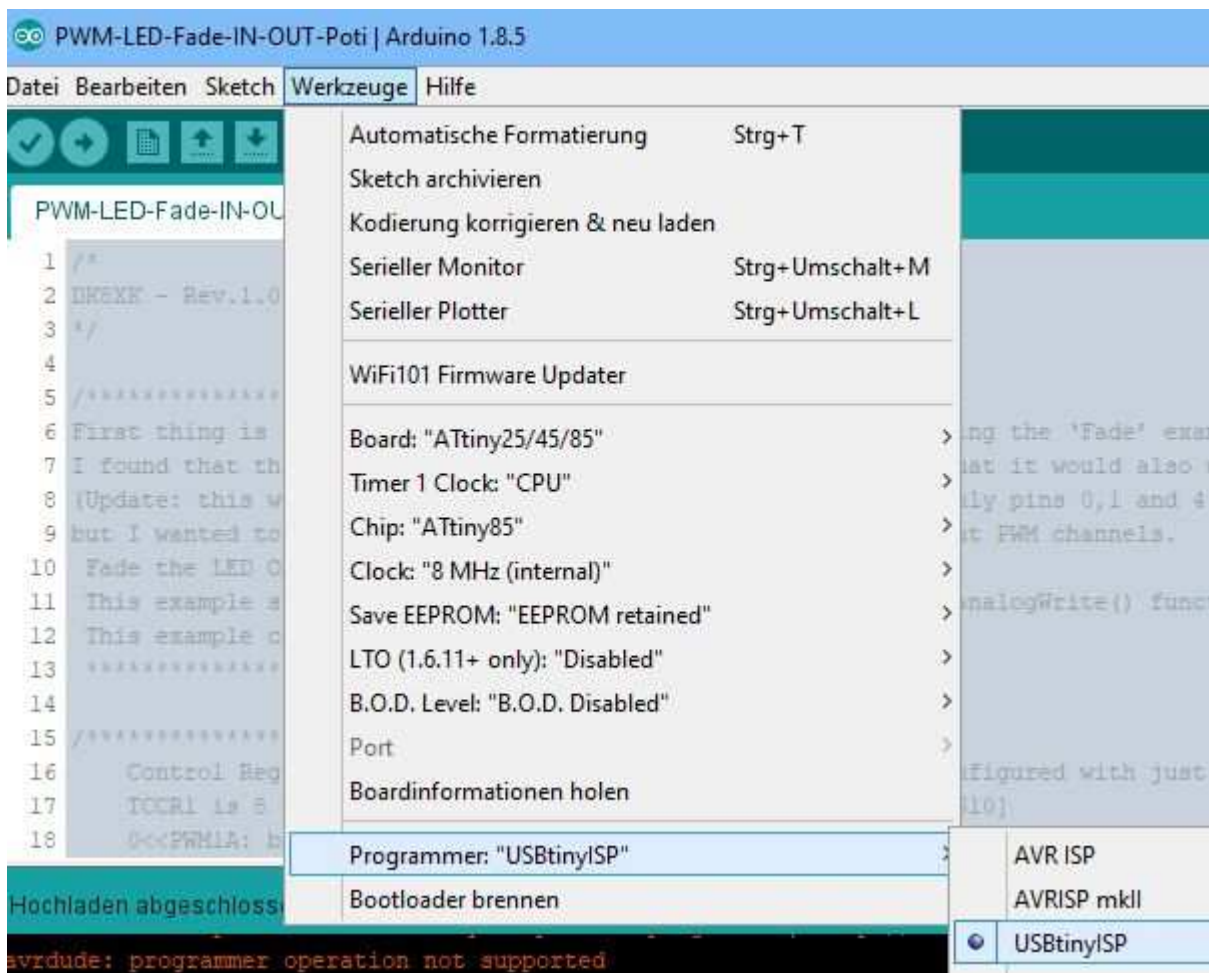
oder USBtinyISP:



Damit ist der USBasp betriebsbereit und kann in der Arduino-IDE als externer Programmierer eingestellt werden.

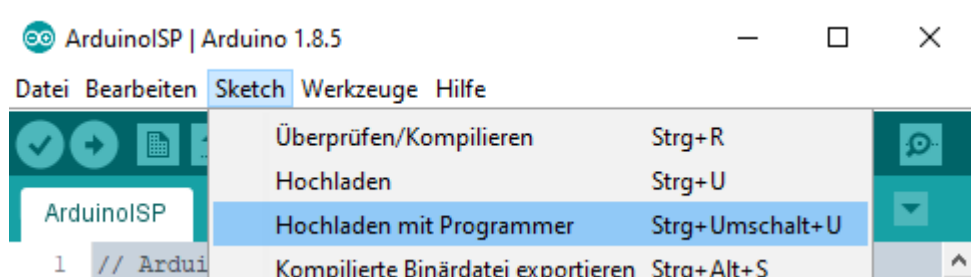


Und so sieht es aus.



Oder für den USBtinyISP:

Später verwende ich zum übertragen des Sketches zum Arduino diesen Befehl:



Entweder mit der Maus klicken oder Strg+Umschalt+U an der Tastatur eintippen.

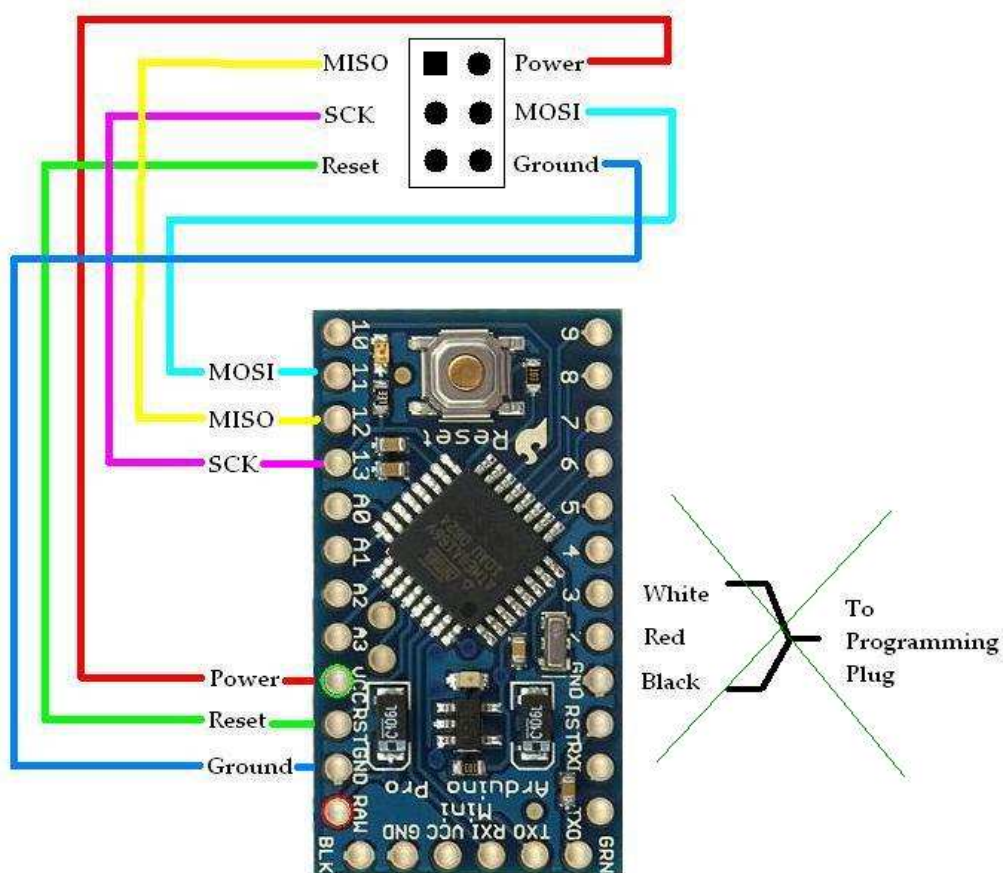
Wichtiger Hinweis:

Vor dem Anstecken des USBasp an den USB-Anschluß des PC müssen erst die Verbindungen zum Arduino mit großer Sorgfalt, damit keine Vertauschung der PINs passiert, hergestellt werden.

Anschlüsse vom USBasp zum Arduino:

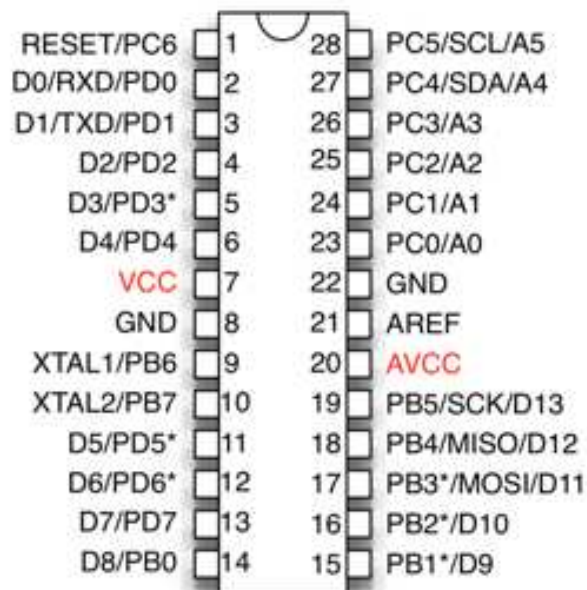
Pin-USBasp	Name	PIN-Arduino	PIN-ATtiny85	PIN-Name-Arduino
1 (4)	MOSI	D11	5	MOSI
5 (5)	PB2 - Reset	Rst	1	Reset
7 (3)	SCK - Clock	D13	7	SCK
9 (1)	MISO	D12	6	MISO
2 (2)	Vcc – 5Volt	Vcc	8	Vcc - Power
10 (6)	GND - 0Volt	Ground	4	GND

Werte in Klammer gelten für den 6-poligen Sockel am ISP falls verfügbar.

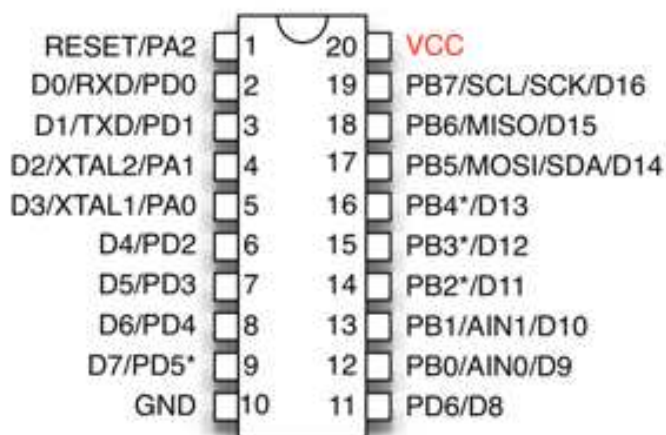


Soll der Arduino nicht vom USBasp sondern anders mit 5Volt versorgt werden die rote Leitung nicht anschließen oder den Jumper auf dem USBasp ziehen.

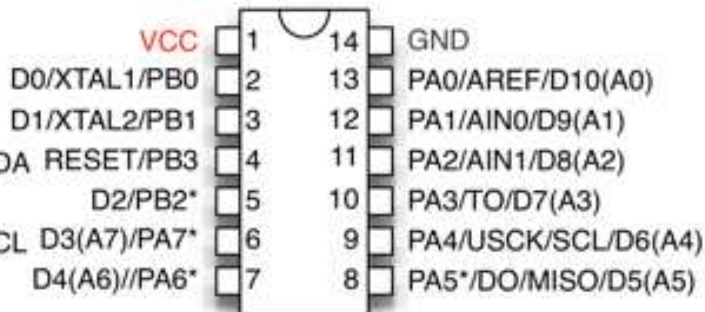
ATMega8/48/88/168/328/Arduino



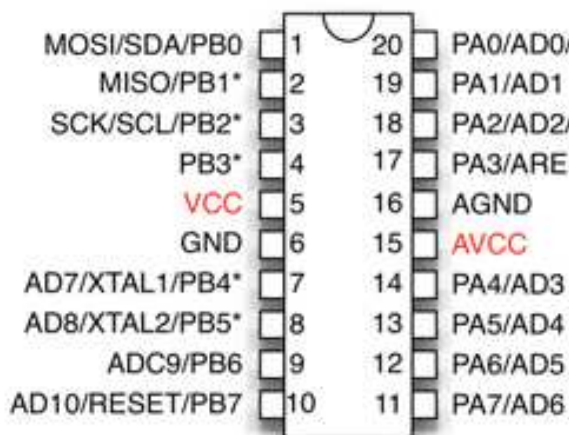
ATTiny2313/4313



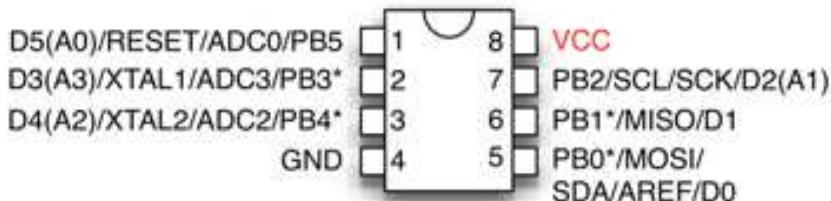
ATTiny24/44/84



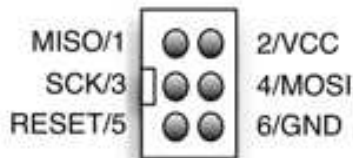
ATTiny26/261/461/861



ATTiny25/45/85/13



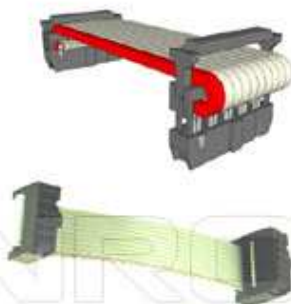
ISP Header



NOTES:

- Arduino pins for ATTinyX313/X5/X4 are from the arduino-tiny project
- PWM pins are marked with (*)

- *ATTiny13 has PWM only on PB0 and PB1
- * No AREF on ATTiny13
- * PB3 and PB4 share the same timer.



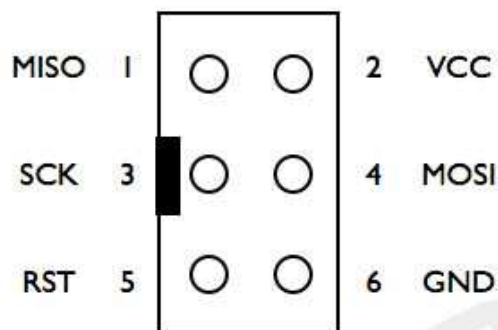
MOSI: 1	□ ○	2 :Vcc
(unused): 3	○ ○	4 :Ground
RESET: 5	○ ○	6 :Ground
SCK: 7	○ ○	8 :Ground
MISO: 9	○ ○	10:Ground

Connector on Programmer

----- FRC10 Cable(Pin#1 arrow) -----

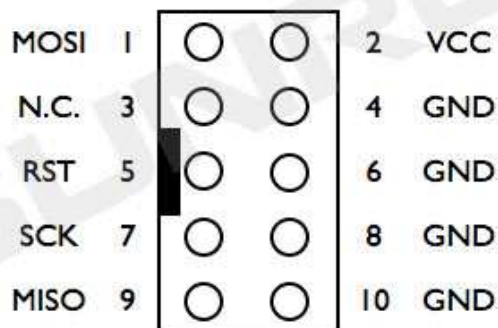
Target board connector
Same as programmer

6 Pin



1	MISO	Master In, Slave Out
2	VCC	Voltage Common Collector
3	SCK	Serial Clock
4	MOSI	Master Out, Slave In
5	RST	Reset
6	GND	Ground

10 Pin



3	N.C.	Not Connected
---	------	---------------

ISP Cable
Bottom View

