

myTinyProg MK2 USB

Erstellt von: Stefan Hoffmann

Inhalt

Einleitung	3
Eigenschaften	3
Entwicklungsumgebung:	3
Grundlagen	4
Hardware myTinyProg MK2 USB	5
Hardwareüberblick	5
Schaltplan myTinyProg MK2 USB	6
Stückliste	7
Realisierungen	7
Auf einer myAVR Laborkarte A	7
Mit anderen AVR Controllern	8
Programmieren und Testen	9

Die Informationen in diesem Produkt werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Die Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen. Für Verbesserungsvorschläge und Hinweise auf Fehler sind die Autoren dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien. Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

Fast alle Hardware- und Softwarebezeichnungen, die in diesem Dokument erwähnt werden, sind gleichzeitig auch eingetragene Warenzeichen und sollten als solche betrachtet werden.

© Laser & Co. Solutions GmbH
Promenadenring 8
02708 Löbau
Deutschland

www.myAVR.de
support@myavr.de

Tel: ++49 (0) 358 470 222
Fax: ++49 (0) 358 470 233

Einleitung

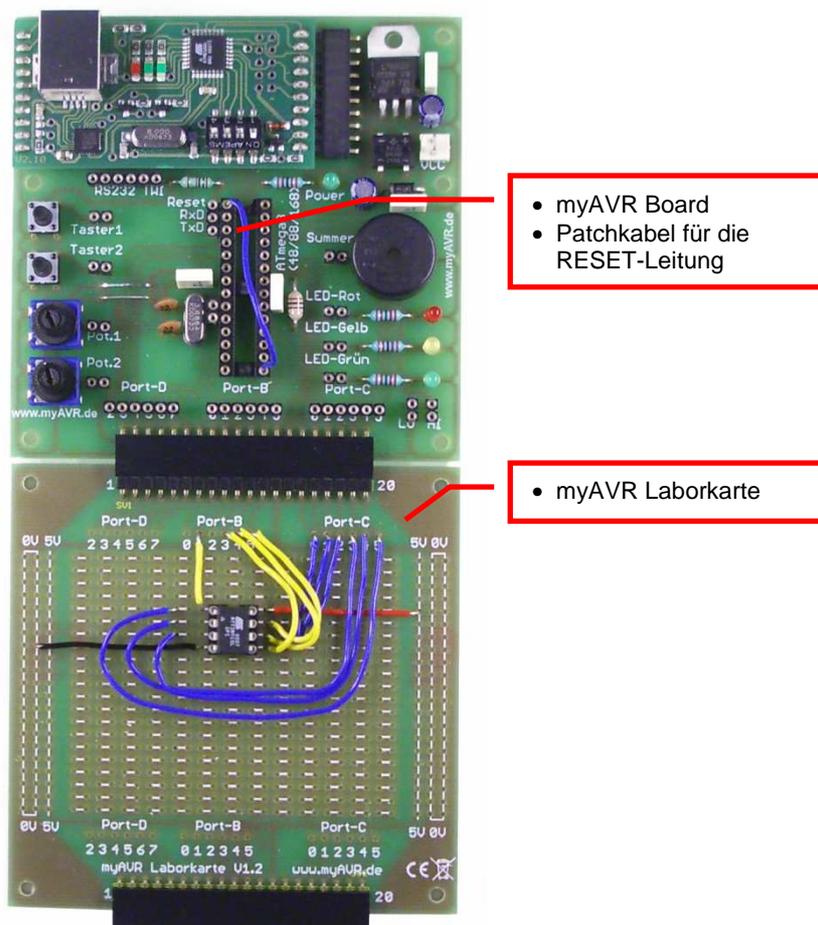
Das vorliegende Projekt myTinyProg MK2 USB beschreibt eine Möglichkeit, das Lern- und Experimentiersystem von myAVR als komfortable Hardware-Umgebung zum Brennen/Programmieren und Testen von ATtiny Controllern in Verbindung mit dem myAVR Board MK2 USB zu benutzen. Das vorgestellte Verfahren funktioniert mit beiden Varianten des myAVR Boards. Die Basis dafür ist eine Laborkarte A und wenige Bauteile wie ein DIP-8-Sockel und eine Steckerleiste; ggf. außerdem ein Quarz und zwei Kondensatoren.

Eigenschaften

- einfache Kommunikation mit dem PC oder Notebook über die USB-Schnittstelle
- Kommunikation über UART, TWI, SPI möglich
- Spannungsversorgung wahlweise über USB-Anschluss oder externe Spannungsversorgung (9-12V)
- 16 MHz Prozessor mit bis zu 256 K Byte Speicher
- Industriestandard-Controller (CP2102) USB zu seriell Konverter (mySmartUSB MK2 und myAVR Stamp PLUS)
- in SMD-Bauweise
- Eingabegeräte: 2 Taster, 2 Analog Geräte Lichtsensor, 4 DIP-Schalter*
- Ausgabegeräte: Speaker, 3 LEDs
- updatefähig über Bootloader
- Highspeed-Programmer (mySmartUSB MK2); separat nutzbar

Entwicklungsumgebung:

- Anschluss: USB Port (virtueller COM Port)
- Programmertyp AVR910/AVR911, mySmartUSB MK2
- Programmiersoftware: SiSy AVR, myAVR Workpad PLUS



Grundlagen

Für das Brennen (Programmieren, Flashen) des fertigen Mikrocontrollerprogramms (*.HEX oder *.BIN) gibt es grundsätzlich zwei Möglichkeiten:

Zum einen kann man ein Programmiergerät verwenden, in das man den Chip einsetzt und programmiert. Dazu muss das Programmiergerät über die parallele, serielle oder USB Schnittstelle an den PC angeschlossen werden. Der Mikrocontroller ist aus dem Zielsystem zu entfernen und auf den entsprechenden Sockel des Programmiergerätes zu stecken. Dann kann das Programm in den FLASH-Speicher des Controllers übertragen werden. War dieser Vorgang erfolgreich, kann der Controller aus dem Programmiergerät entnommen und wieder in das Zielsystem eingebaut werden. Ein solches Programmiergerät ist zum Beispiel der myMultiProg MK1 LPT oder myMultiProg MK2 USB von myAVR.

Eine weitere Lösung für die Programmierung des Mikrocontrollers ist das sogenannte „In System Programming“ (ISP). Dabei muss der Controller nicht aus dem Zielsystem ausgebaut werden, sondern kann direkt im System programmiert und getestet werden. Dafür muss das Zielsystem eine ISP-Schnittstelle bereistellen. Mit einer zusätzlichen Hardware, dem sogenannten ISP-Programmer, der an dem LPT-Port, die COM-Schnittstelle oder dem USB-Port angeschlossen wird, kann aus der Entwicklungsumgebung heraus das Programm direkt in das Zielsystem übertragen werden. Als Experimentierhardware für 28-polige DIL Controller von ATMEL wie dem ATmega8 und Pinkompatible eignet sich das myAVR Board (LPT oder USB). Für AT-MEL AVR Controller mit anderen Pin-Zahlen wie z.B. der Tiny-Serie aber auch andere Mega-Controller eignet sich die hier beschriebene Erweiterung zum myAVR Board auf Basis der Laborkarte A.

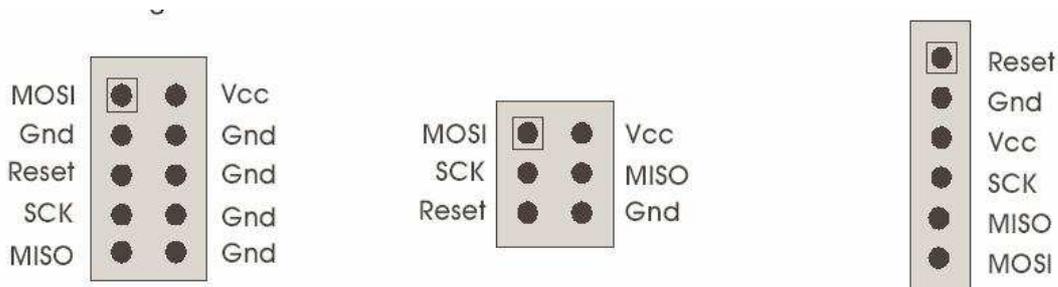


Abbildung: ISP- Anschlüsse, ISP- Standards, Atmel 10-polig, 6-polig, TwinAVR

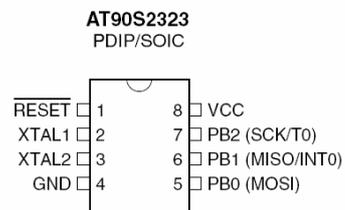
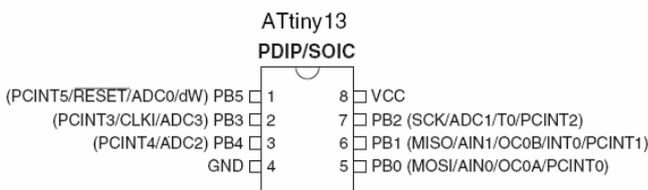
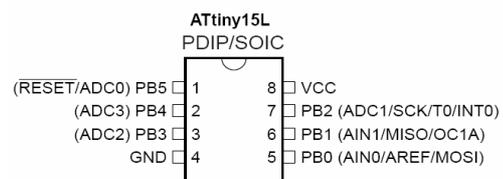
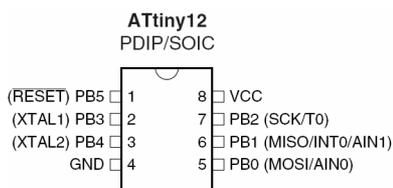
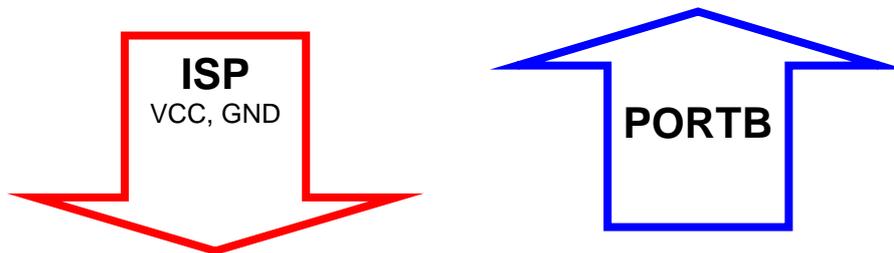
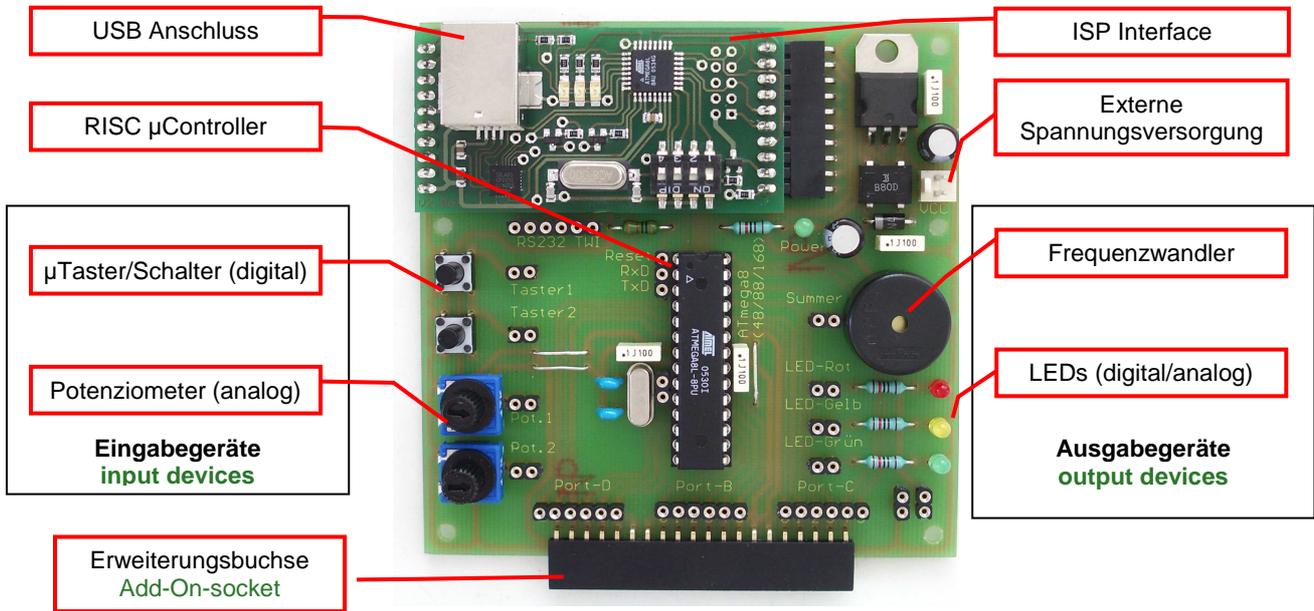
Die folgende Tabelle zeigt die Bedeutungen der einzelnen Signalleitungen der ISP-Schnittstelle. Diese sind die Leitungen des SPI (Serial Processor Interface).

Pin	Name	Comment
SCK	Serial Clock	Programming clock, generated by the in-System programmer (master)
MOSI	Master Out – Slave In	Communication line from In-System programmer (master) to target AVR being programmed (slave)
MISO	Master In – Slave Out	Communication line from target AVR (slave) to In-System programmer (master)
GND	Common Ground	The two system must share the same common ground
RESET	Target AVR MCU Reset	To enable in-System programming, the target AVR Reset must be kept active. To simplify this, the In-System programmer should control the target AVR Reset.
Vcc	Target Power	of target operating at any voltage. The target can have power supplied through the In-System programming connector for the duration of the programming cycle.

Hardware myTinyProg MK2 USB

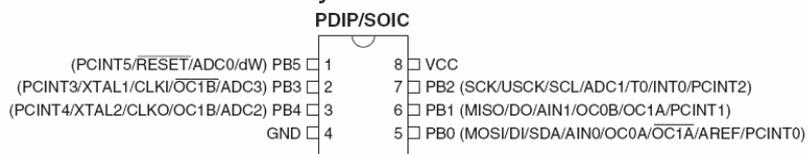
Hardwareüberblick

Das myAVR Board MK2 USB besitzt folgende Struktur:
 USB-Programmer, Peripherie zum Testen von Anwendungen, Erweiterungsbuchse.



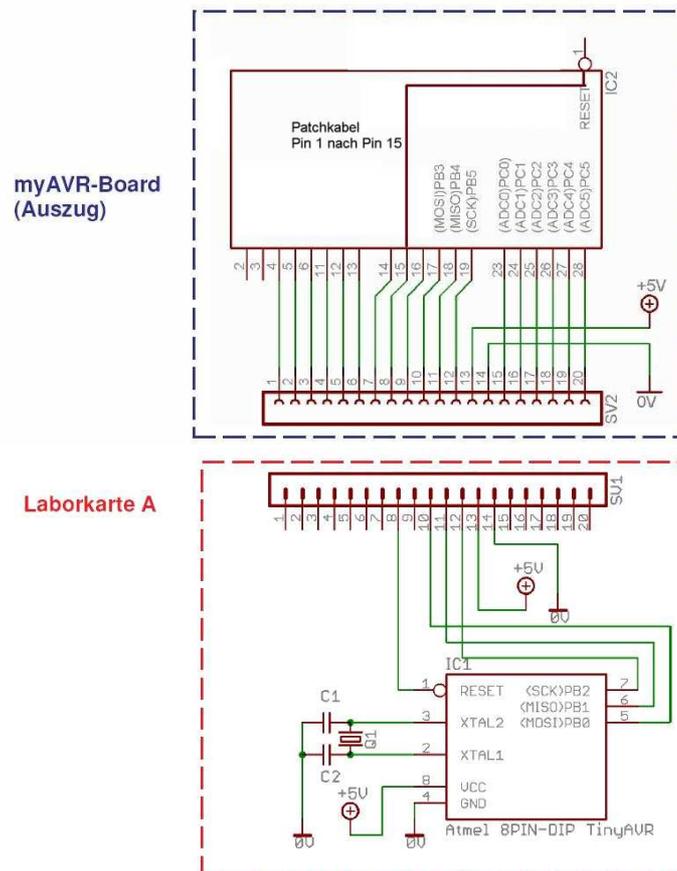
AT90S/LS2323

ATtiny25/45/85



Schaltplan myTinyProg MK2 USB

Die Schaltung soll die Programmierung der ATMEL Controller AT90S2323, ATtiny12, 13 und 15 sowie weiterer AVR-Controller der Tiny-Reihe ermöglichen. Außerdem werden die Möglichkeiten gegenüber dem TinyProg MK1 LPT erweitert. Es soll das Testen und Experimentieren mit dem myAVR Board und den darauf verfügbaren Ein- und Ausgabegeräten möglich sein. Dabei sind die ISP-Leitungen entsprechend der Pin-Belegung der zu programmierenden Tiny-Controller und der Erweiterungsbuchse des myAVR Boards zu verbinden. Ebenso sollten die restlichen Port-Leitungen des Tinsys an die Erweiterungsbuchse zurückgeführt werden, damit die Tiny-Programme auf dem myAVR Board bequem getestet werden können.



Die „Rückführung“ der Portleitungen PB0 bis PB5 zur Stiftleiste an Positionen Port C.0 bis Port C.5 sind hier nicht dargestellt.

Stückliste

Das myAVR Board besitzt folgende Struktur:

USB-Programmer, Peripherie zum Testen von Anwendungen, Erweiterungsbuchse.

Bezeichnung im Schaltplan	Bauteil	Artikelnummer Suchbegriff	Anzahl
IC1	ATtiny12/13/15 DIP	ATTINY 13-20 DIP (www.myavr.de bzw. www.reichelt.de)	1
J1	IC Sockel 8 DIP	GS 8 oder GS 8P (www.reichelt.de)	1
X1	Laborkarte A (Platine)	myAVR Laborkarte (www.myAVR.de)	1
SV1	Stiftleiste 20polig	Stiftleiste 20polig (www.myAVR.de)	1
	Schalt draht ca. 20 Zentimeter (Farbsortiment)	Patchkabel (www.myAVR.de)	

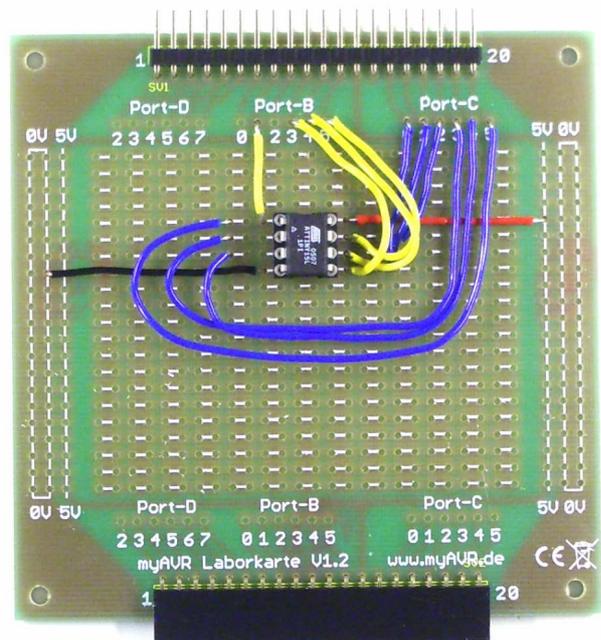
Nach Bedarf für externen Takt:

C1, C2	KERKO 22pF	KERKO 22P (www.reichelt.de)	2
Q1	Quarz 3,6864 MHz	Quarz 3,6 (www.myAVR.de)	1

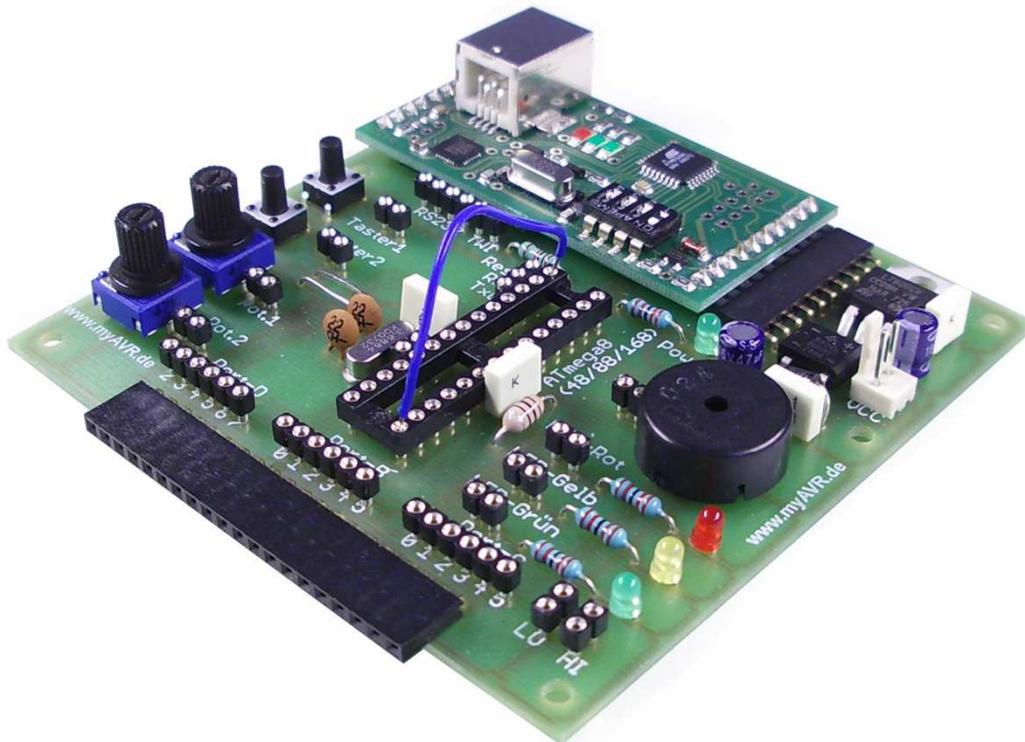
Realisierungen

Auf einer myAVR Laborkarte A

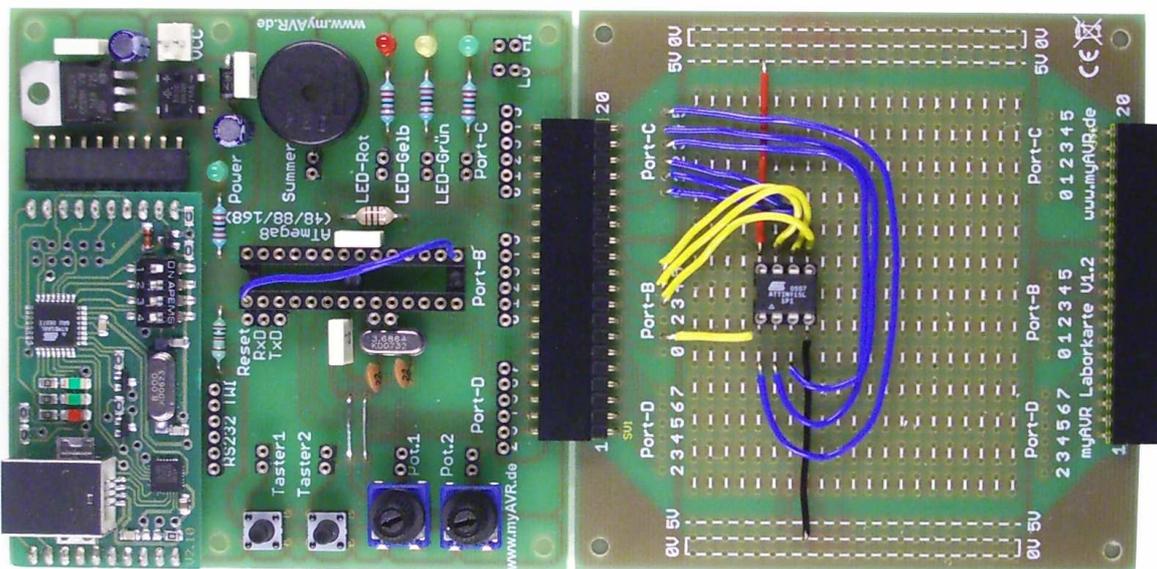
1. Löten Sie die Buchsenleiste und den IC-Sockel auf die Laborkarte. Achten Sie darauf, dass die paarigen Lötäugen der Laborkarte den einfachen Anschluss gewährleisten.
2. Verbinden Sie PIN 8 des IC-Sockels (Plus) mit der Versorgungsspannung (rotes Kabel im Bild) und PIN 4 des IC-Sockels (Minus) mit 0 V (schwarzes Kabel im Bild).
3. Verbinden Sie die PINs 5, 6, 7 des IC-Sockels mit Port B 3, 4 und 5 an der Steckerleiste (gelbe Kabel im Bild).
4. Verbinden Sie PIN 1 des IC-Sockels (Reset) mit Port B1 an der Steckerleiste (gelbes Kabel).
5. Verbinden Sie
 - PIN 5 (PB0) mit Port C.0 der Steckerleiste
 - PIN 6 (PB1) mit Port C.1 der Steckerleiste
 - PIN 7 (PB2) mit Port C.2 der Steckerleiste
 - PIN 2 (PB3) mit Port C.3 der Steckerleiste
 - PIN 3 (PB4) mit Port C.4 der Steckerleiste
 - PIN 1 (PB5) mit Port C.5 der Steckerleiste
(Blaue Kabel im Bild)



6. Stecken Sie ein Kabel von Pin 1 der leeren IC-Fassung auf dem myAVR Board nach Pin 15 derselben IC-Fassung. (Blaues Kabel im Bild)



7. Stecken Sie den Controller in den Sockel der Erweiterungsplatine



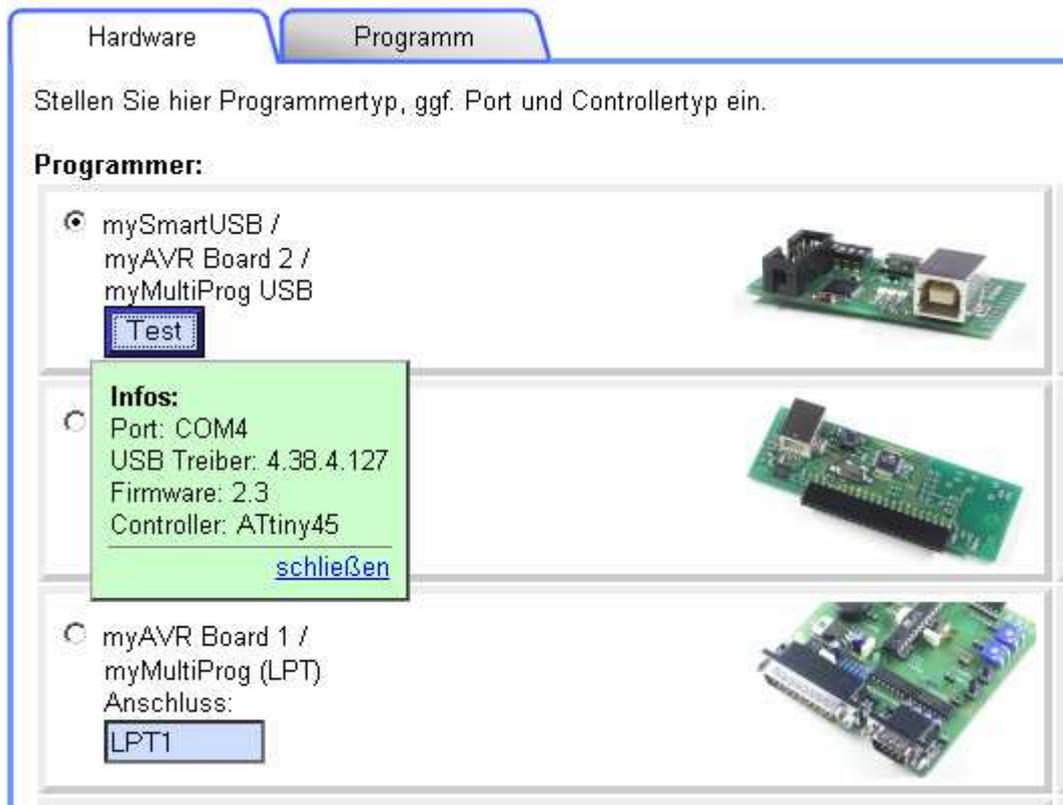
Mit anderen AVR Controllern

Bei AVR-Controllern mit anderen Pin-Zahlen kann analog verfahren werden, wobei Pin B.0 bis Pin B.5 jeweils an Port-C geführt werden können. Weitere Ein/Ausgänge sollten dann auf die Pins von leeren IC-Fassungen auf der Platine geführt werden und dort dann mit Kabeln abgegriffen werden können. Es empfehlen sich hierfür 16 DIL-IC-Fassungen, da die Ports maximal von 0 bis 7 (=8) gehen und daher je IC-Fassungshälfte ein Port zusammengefasst werden kann.

Programmieren und Testen

Sie können nun wie mit dem myAVR Board gewohnt auch 8-polige Controller programmieren/brennen und mit dem myAVR Board und dessen Peripherie testen.

Im Folgenden sehen Sie einen Verbindungsscheck mit einem myAVR Board MK2 USB und dem myTinyProg MK2 USB, der einen ATtiny45 im Sockel hat. Als Software Grundlage diente das myAVR Progtool.



Beim Experimentieren müssen Sie lediglich beachten, dass die aufgedruckten Portbezeichnungen auf dem myAVR Board natürlich nicht den tatsächlich angeschlossenen Ports des ATtiny auf der Laborkarte entsprechen. Sie können aber trotzdem Ihre gewohnte Entwicklungsumgebung (z. B. myAVR Workpad, SiSy AVR oder Bascom) benutzen, müssen kein Kabel vom PC zum myAVR Board umstecken und auch keine spezielle Firmware auf das myAVR-System brennen.

Viel Erfolg und Spaß beim Programmieren und Testen!