

## Neues von myAVR – USB-Programmer mySmartUSB

Die myAVR-Komponenten sowie die Grundlagenlehrgänge von Laser & Co. Solutions erfreuen sich stetig wachsender Beliebtheit, machen Sie doch die ATMEL-AVR-Controller samt verschiedenen Programmierumgebungen auf sehr einfache Weise handhab- und erlebbar. Wir stellen ein sehr universelles Mitglied der myAVR-Familie vor: den Universal-USB/ISP-Programmer mySmartUSB, der gleichzeitig auch ein USB-Kommunikations-Interface für AVR-Mikrocontroller bildet.

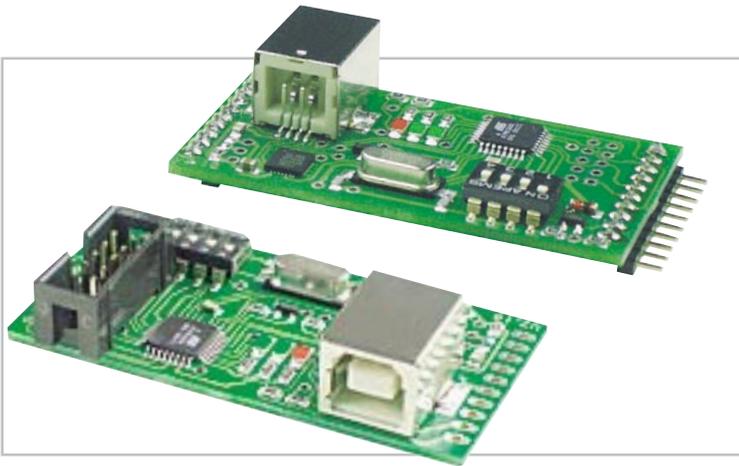
### AVRs per USB programmieren

Die erste Generation der AVR-Programmer wie das bekannte STK 500, die kompakten SPI-Programmer oder auch das im „ELVjournal“ in unserer AVR-Programmierserie verwendete myAVR-Board benötigten noch ein bis zwei serielle und z. T. auch eine parallele Schnittstelle für die Programmierung der verschiedenen AVR-Typen. Allerdings verfügen die beliebten Notebooks kaum noch über einen solchen Port. Hier, wie auch an modernen Desktop-PCs, dominiert heute USB.

Ergo bietet es sich natürlich an, diese beliebte, universelle und in Form von USB 2.0 sehr schnelle Schnittstelle auch für die Programmierung von Mikrocontrollern zu nutzen. Mit dem mySmartUSB hat man gleich mehrere Probleme gelöst. Der kleine USB-Baustein (Abbildung 1) ist so universell ausgelegt, dass er eine ganze Reihe von Aufgaben erledigen kann (sie-

### Technische Daten: mySmartUSB

Interfaces:	USB, ISP, SPI, UART, TWI/I <sup>2</sup> C
Programmer-Kompatibilität:	AVR910/911
USB-Controller:	Industriestandard CP2102
Zielsystem-Spannung:	3,3/5 V
Spannungsversorgung:	USB-Bus-Powered 5 Vdc
Stromaufnahme:	
ohne angeschlossenes Zielsystem:	max. 20 mA
mit angeschlossenem Zielsystem:	max. 100 mA
Max. Stromaufnahme Zielsystem:	80 mA
Abmessungen:	60 x 30 x 12 mm



**Bild 1:** Der mySmartUSB mit und ohne ISP-Wannenstecker, oben mit angelöteter myAVR-2-USB-Anschlussleiste

he Titelgrafik und Abbildung 2), die wir im Folgenden jeweils kurz erläutern wollen. Der Programmer kann als Tochterplatine auf die USB-Version des myAVR-Boards 2 aufgesteckt oder einzeln als AVR910/911-kompatibler Programmer eingesetzt werden. Die Kommunikation mit dem PC erfolgt völlig transparent über einen virtuellen, seriellen Port. Der kleine Programmer ist unter MS Windows (2000/XP/2003/Vista), Mac-OS ab 9 und Linux ab 2.4 einsetzbar. Für die unixbasierten Mac-Betriebssysteme sowie Linux gibt es die kompatible Open-Source-Programmierungsumgebung „AVRDUDE“. Alles darüber findet sich unter [1] und [2].

Über einen vorinstallierten Bootlader lässt sich auch die Firmware für neue AVR-Typen und -Protokolle aktualisieren, so kann rasch auf neue Controllertypen reagiert werden.

Neben dem Einsatz als Programmer kann mySmartUSB aber auch als Bridge-Baustein zwischen dem PC/USB und verschiedenen seriellen Schnittstellen dienen und ist so auch für eigene Applikationen als einfach handhabbare Kommunikationsschnittstelle einsetzbar.

Eine separate Spannungsversorgung ist nicht notwendig, diese erfolgt per USB vom PC aus. Hierüber sind auch die zu programmierenden Zielsysteme mit einem Strombedarf bis 80 mA versorgbar. Dabei ist mySmartUSB sowohl in 5-V- als auch in 3,3-V-Zielsystemen einsetzbar. Für wechselweise Verwendung in beiden Spannungssystemen ist eine Stiftleiste für Jumper-Umschaltung nachrüstbar.

## ISP-Programmierung

MySmartUSB verfügt über eine Standard-ISP-Schnittstelle, die zusammen mit dem mitgelieferten Kabel die In-System-Programmierung (ISP) von AVR-Schaltungen innerhalb deren Einsatzsystem via USB möglich macht. Die ISP-Schnittstelle verfügt

## Übersicht über die Features von mySmartUSB:

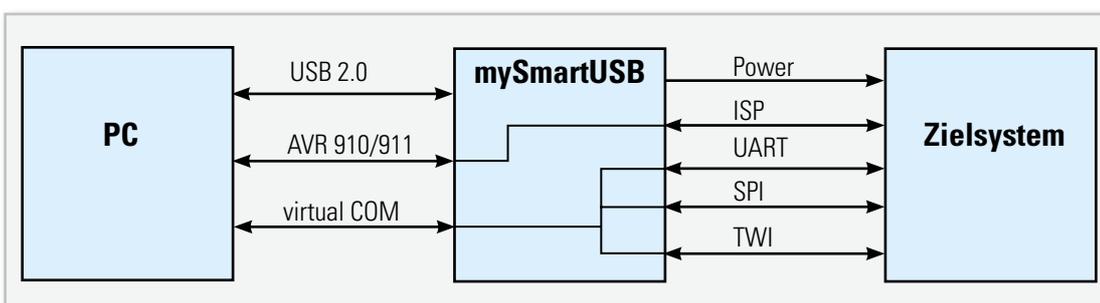
- einfache Kommunikation des Mikrocontrollers mit dem PC über USB-Schnittstelle; stellt virtuellen COM-Port zur Verfügung
- Spannungsversorgung über USB-Anschluss
- über ISP-Adapter Programmierung einer Vielzahl von AVR-Systemen möglich
- ISP-Anschluss mit 10-Pin-ATMEL-Standardbelegung
- ATmega8 mit Firmware nach ATMEL-AN910- und -AN911-Programmierstandard
- schnelle Programmierung über USB 2.0 durch Blockmode
- umschaltbar zwischen Programmier-, Kommunikations-, Ruhemodus und myMode
- USB-zu-TWI-Adapter
- USB-zu-SPI-Adapter
- Power-on und Power-off durch Anwender schaltbar
- Zustandsanzeige (rote/grüne LEDs)
- updatefähig über Bootloader
- Rescue-Mode für „verfusste“ Controller
- DIP-Schalter für Betriebsart, Power, Programmier- und Datenmodus, RESET und Bootloader
- Interface für das myAVR-Board-2-USB
- Industriestandard-Controller (CP2102) USB-2.0-zu-seriell-Konverter

über die 10-Pin-ATMEL-Standard-Belegung (Abbildung 3), kann also unmittelbar an ISP-Schnittstellen von AVR-Systemen angeschlossen werden. Für die Anbindung an die üblichen Programmierungsumgebungen ist lediglich die Installation des mitgelieferten Treibers notwendig.

Die Firmware verfügt über einen automatischen Step-down-Modus für Controller, die auf langsame Taktquellen unter 1 MHz umgeschaltet wurden und die sich bisher nicht mehr über die Standard-ISP-Geschwindigkeit ansprechen ließen. Der Blockmode des AVR911-Protokolls erlaubt über USB 2.0 eine sehr schnelle Datenübertragung und damit eine hohe Programmiergeschwindigkeit. Ein besonders einfaches und zudem kostenlos verfügbares Programmierwerkzeug ist das myAVR ProgTool, eine Auskopplung aus den lizenzpflichtigen AVR-Produkten.

## Kommunikations-Interface

MySmartUSB ist nicht nur als Programmer einsetzbar, sondern auch als universelles Kommunikations-Interface für die



**Bild 2:** Übersicht über die Möglichkeiten von mySmartUSB

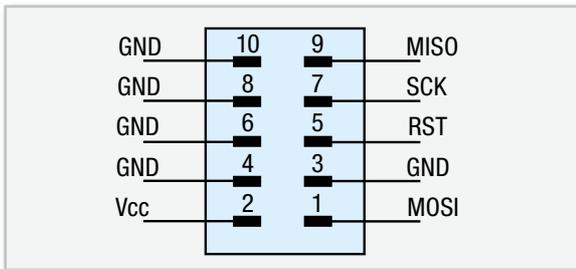


Bild 3: Die Belegung des ISP-Steckers entspricht der 10-Pin-Standard-Belegung von ATMEL

Schnittstellen bzw. Protokolle SPI, UART und I<sup>2</sup>C/TWI (TWI ist kompatibel zu I<sup>2</sup>C, der Name steht für „Two Wire Interface“ und wurde von ATMEL aus lizenzrechtlichen Gründen eingeführt). Abbildung 4 zeigt die Belegungen des Boards für die verschiedenen Schnittstellen. So kann man auch eigene Applikationen über dieses Board via USB ansprechen und es quasi als preiswerten und kompletten USB-Konverter einsetzen. Es ist relativ einfach auch als klassische serielle PC-Schnittstelle (RS232) realisierbar, hier sind lediglich noch eine kleine MAX232-Standard-Applikation mit dem MAX232 und vier Elkos erforderlich.

Auch die beliebten myTWI-Module oder TWI/I<sup>2</sup>C-Sensoren sind mit dem Board ganz einfach per TWI und via USB an einem PC nutzbar.

Über die Freeware „mySmartUSB Terminal“ können die Funktionen eines TWI- oder SPI-Protokolls ausgeführt und genau Schritt für Schritt beobachtet werden – ein ideales Tool für den Einsteiger!

## Erste Hilfe für „verfust“ Controller

Wieder einmal einen Controller durch das versehentlich falsche Stellen eines Fuse- oder Clock-Bits „ruhiggestellt“? Ein solcher Prozessor ist normalerweise nicht mehr programmier- und auch meist nicht einmal mehr erreichbar – er wanderte bisher in den Elektronikschrott. Die aktuelle Firmware (2.5) des mySmartUSB verfügt jetzt über einen zuschaltbaren Rescue-Mode, der ein spezielles Taktsignal (Rescue Clock) erzeugt. Das macht den Controller wieder erreichbar, seine

## Durch mySmartUSB unterstützte Controllertypen:

AT90PWM3	ATmega3290
AT90S1200	ATmega48
AT90S2313	ATmega64
AT90S2343	ATmega644
AT90S4414	ATmega8
AT90S4433	ATmega8515
AT90S8515	ATmega8535
AT90S8535	ATmega88
ATmega103	ATtiny12
ATmega128	ATtiny13
ATmega16	ATtiny15
ATmega161	ATtiny2313
ATmega162	ATtiny24
ATmega163	ATtiny25
ATmega164P	ATtiny26
ATmega168	ATtiny44
ATmega169	ATtiny45
ATmega32	ATtiny84
ATmega324P	ATtiny85
ATmega328P	
ATmega329	

Fuse- und Lock-Bits sind über die Programmierumgebung, z. B. das myAVR Workpad, auslesbar und können richtig eingestellt werden. So ist der Controller wieder gerettet!

Bleibt als Fazit: Mit dem mySmartUSB steht ein äußerst universell handhabbarer USB-Schnittstellen- und Kommunikationsbaustein zur Verfügung, der nicht nur im myAVR-System einsetzbar ist. Durch seinen updatefähigen Bootlader ist mySmartUSB auch zukunftssicher und gehört damit in den „Werkzeugkasten“ jedes Elektroniklers, der sich mit eigenen Mikrocontrollerlösungen beschäftigt.

**Angebote finden Sie im ELV-Hauptkatalog 2009 ab S. 682 und im Internet unter [www.messtechnik.elv.de](http://www.messtechnik.elv.de). **

### Internet:

[1] [www.nongnu.org/avrdude/](http://www.nongnu.org/avrdude/)

[2] [www.mikrocontroller.net/articles/AVRDUDE](http://www.mikrocontroller.net/articles/AVRDUDE)

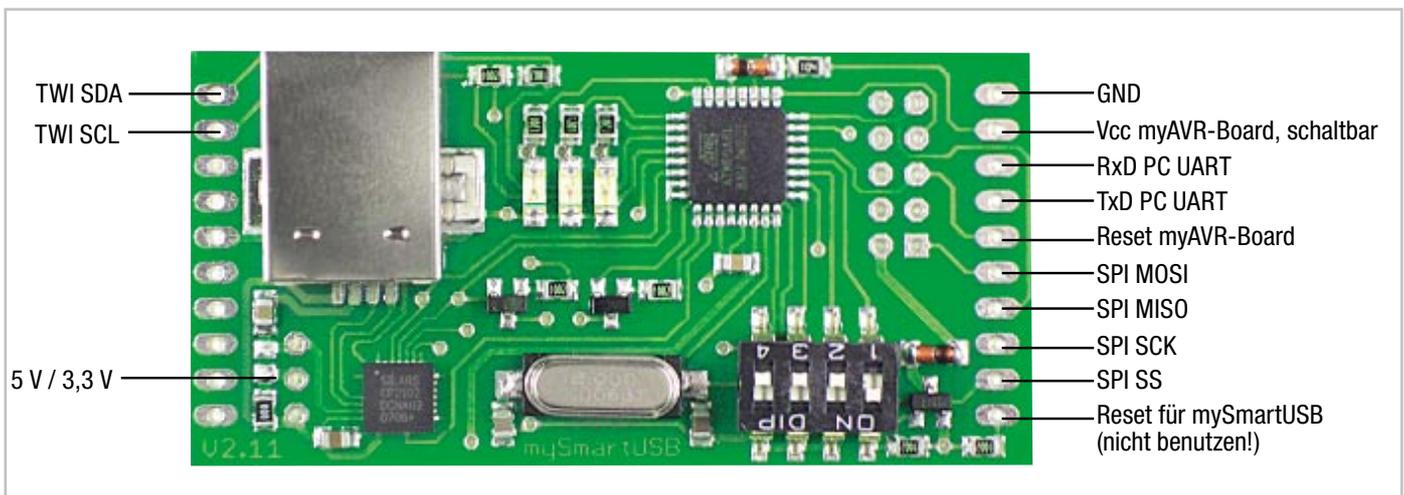


Bild 4: Die Belegung der verschiedenen Schnittstellen