

## mySmartUSB MK3

Hardware-Version 3.30

Software-Version 1.19

### Inhalt

Allgemeine Beschreibung.....	3
Überblick.....	3
Programmierung.....	3
Kommunikation.....	3
Eigenschaften.....	4
Technische Daten.....	4
Betriebsdaten.....	4
Mechanische Daten.....	4
Anschlüsse.....	5
Pinbelegung.....	6
8 poliger Wannenstecker.....	6
ISP6 (6 poliger Wannenstecker).....	6
LED-Statusanzeige.....	7
UART-Konfiguration und Baudraten.....	8
Beschreibung des myMode.....	9
Allgemeine Befehle.....	9
Menü: main.....	9
Menü: prog.....	9
Menü: spi.....	11
Menü: lines.....	11
Menü: twi.....	11
Fehler-Codes.....	12
Das mySmartUSB MK3 Protokoll.....	13
Programmereinstellungen.....	14
Programmereinstellungen SiSy AVR (ab 2.18d).....	14
Programmereinstellungen in myAVR Workpad (1.6).....	15
Programmereinstellungen in AVRDUDE (5.8).....	16
Programmereinstellungen in BASCOM (1.11.9.8).....	17
Programmereinstellungen in Codevision (2.04.8).....	18
Programmereinstellungen AVR Studio (4.18.700).....	19
Anwendungsbeispiele.....	20
Allgemeine Sicherheitshinweise.....	20

### Contents

General description.....	3
Overview.....	3
programming.....	3
communication.....	3
Properties.....	4
Technical Data.....	4
Operating Data.....	4
Mechanical Data.....	4
Connections.....	5
pin allocation.....	6
8 pin print plug.....	6
ISP6 (6 pin, print plug).....	6
LED status display.....	7
UART-configuration and baud rate.....	8
myMode-description.....	9
General Commands.....	9
menu: main.....	9
menu: prog.....	9
menu: spi.....	11
menu: lines.....	11
menu: twi.....	11
Error Codes.....	12
The mySmartUSB MK3 protocol.....	13
Programmer settings.....	14
changing setting in SiSy AVR (from 2.18d).....	14
changing setting in myAVR WorkPad (1.6).....	15
changing setting in AVRDUDE (5.8).....	16
changing setting in BASCOM (1.11.9.8).....	17
changing setting in Codevision (2.04.8).....	18
changing setting in AVR Studio (4.18.700).....	19
Exemples of use.....	20
Safety Guidelines.....	20

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.

© SiSy Solutions GmbH  
Promenadenring 8B  
02708 Löbau  
Deutschland

[www.myAVR.de](http://www.myAVR.de)  
[support@myavr.de](mailto:support@myavr.de)

Tel: ++49 (0) 3585 470 222  
Fax: ++49 (0) 3585 470 233

In spite of the great care taken while writing this document the author is not responsible for the topicality, correctness, completeness or quality of the information provided. Liability claims regarding damage caused by the use of any information provided, including any kind of information that is incomplete or incorrect, will therefore be rejected.

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

All trademarks and registered trademarks appearing in this document are the property of their respective owners.

© SiSy Solutions GmbH  
Promenadenring 8B  
02708 Löbau  
Germany

[www.myAVR.com](http://www.myAVR.com)  
[support@myavr.com](mailto:support@myavr.com)

Tel: ++49 (0) 3585 470 222  
Fax: ++49 (0) 3585 470 233

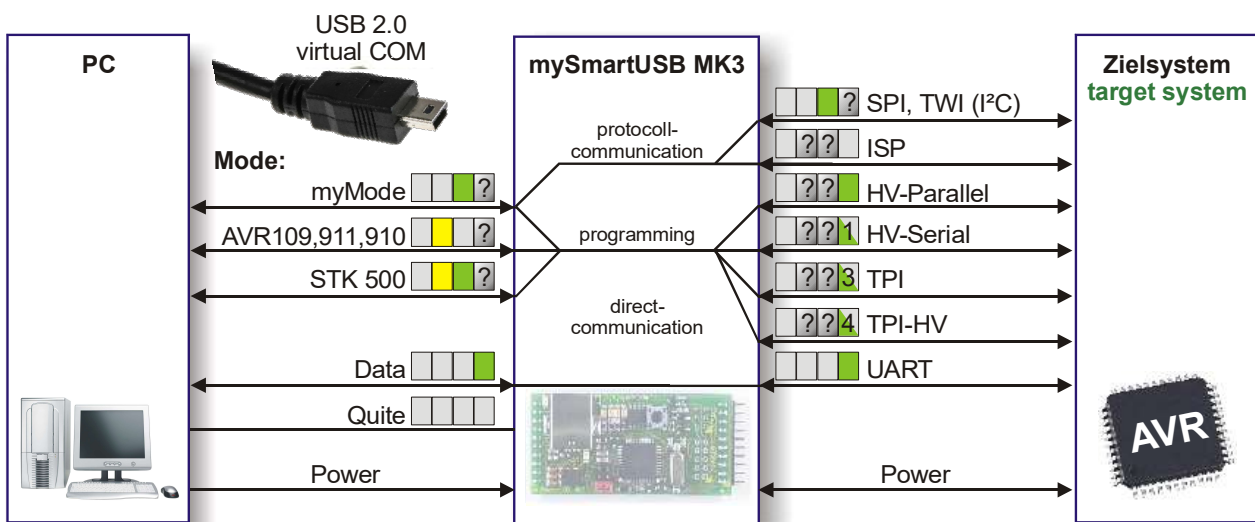
### Allgemeine Beschreibung

Der mySmartUSB MK3 ist ein kompakter Highspeed USB-Programmer für Atmel Mikrocontroller. Sie können mit Hilfe dieses Werkzeugs eine Vielzahl von AVR-Controllern einfach über die USB-Schnittstelle seriell oder parallel programmieren. Neben der Parallelschnittstelle verfügt der mySmartUSB MK3 über einen ISP-Anschluss Wannenstecker mit UART-Option. Die Kommunikationsschnittstelle kann als UART, TWI/I<sup>2</sup>C oder SPI-Bridge fungieren und erscheint auf dem PC völlig transparent als virtueller COM-Port. Für den Anschluss an den PC verwenden Sie ein normales Mini-USB-Kabel. Es besteht die Möglichkeit das Zielsystem mit Spannung zu versorgen. Somit sind für viele Anwendungsfälle keine externen Spannungsquellen nötig.

### General description

The mySmartUSB MK3 is a compact high-speed USB programmer for Atmel microcontrollers. With this tool, you can easily program a variety of AVR controllers serially or in parallel via the USB interface. In addition to the parallel interface, the mySmartUSB MK3 has an ISP port connector with UART option. The communication interface can use as a UART, TWI/I<sup>2</sup>C or SPI bridge and appears completely transparent on the PC as a virtual COM port. To connect to the PC, use a normal mini USB cable. It is possible to supply the target system with voltage. This means that for many applications no external voltage sources are necessary.

### Überblick Overview

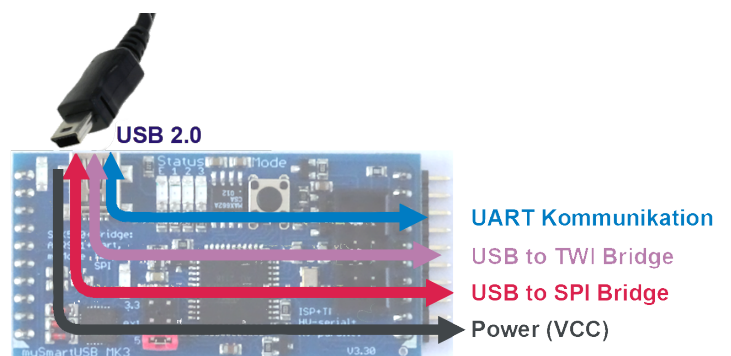
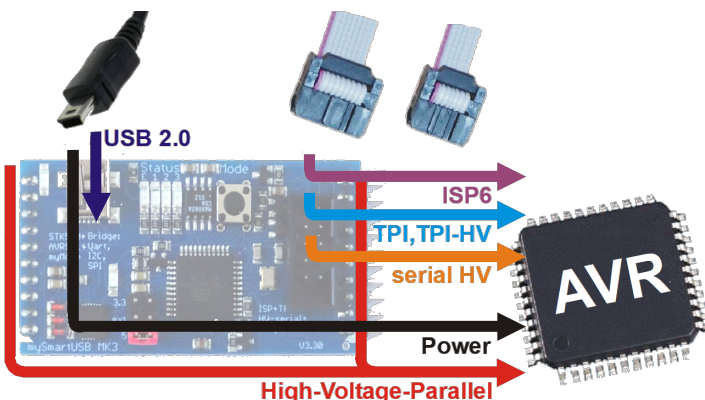


LED Status:  off  on  on  on or off  blink

Auswahl durch Tastendruck: kurz=Mode, lang=Programmier-Mode  
Choice by button: short=mode, long=program-mode

### Programmierung programming

### Kommunikation communication



**Eigenschaften**

- Highspeed AVR-Programmer und Kommunikationsschnittstelle
- sechs Programmiermodi: ISP seriell, parallel, Hoch-Volt seriell, TPI, Hoch-Volt TPI
- Auswahl der Betriebsmodi erfolgt über Software-Kommandos oder den Mode-Taster
- einfache Kommunikation des Mikrocontrollers mit dem PC oder Notebook über USB-Schnittstelle; stellt virtuellen COM-Port zur Verfügung
- umschaltbar zwischen Programmier-, Kommunikations- und Ruhemodus,
- unterstützte Protokolle: STK500, AVR910/911, myMode
- Spannungsversorgung über USB-Anschluss sowie Power-On und Power-Off durch Anwender schaltbar
- detaillierte Zustandsanzeige mit 6 LEDs
- updatefähig über Bootloader
- Taster für Betriebsart
- CP2102 USB 2.0 zu seriell Konverter
- Qualitätsleiterplatte FR4, Industriefertigung, robust

**Properties**

- high-speed AVR programmer and communication-interface
- six program modes: ISP serial, parallel, high voltage serial, TPI, high voltage TPI
- choice of the operational mode by software commands or mode-button
- simple communication between microcontroller and the PC or notebook via USB interface; provides a virtual COM-port
- switchable between programming, communication, sleeping mode and myMode
- supported protocols: STK500, AVR910/911, myMode
- power supply via USB connection as well as switchable power-on and power-off modes by the user
- detailed status indication with 6 LEDs
- updateable via bootloader
- button for control mode
- CP2102 USB 2.0 to serial converter
- quality printed circuit board FR4, industrial production

**Technische Daten**

Betriebsdaten	
Versorgungsspannung	5 V über den USB-Bus
Betriebsstrom	10-20 mA typisch ohne weitere Verbraucher bis 100 mA bei Anschluss an Zielsysteme
Betriebsspannung	5 V
Betriebstemperatur	0 bis 30 °C
Lagertemperatur	-20 °C bis + 70 °C

**Technical Data**

Operating Data	
Supply Voltage	5 V via USB bus
Operating Current	10-50 mA typical without other loads up to 100 mA if connected to target system
Operating Voltage	5 V
Operating Temperature	0 – 30 °C
Storage Temperature	-20 °C – 70 °C

**Mechanische Daten**

Abmessungen Platine (L x B x H)	60 x 30 x 16 mm
Masse	13 g
Rastermaß	2,54 mm
Leiterplattenmaterial:	FR8, 1,5 mm Dicke, 0,35 µm Cu Auflage, zweiseitig, Lötstopmmaske, verzinkt, durchkontaktiert

**Mechanical Data**

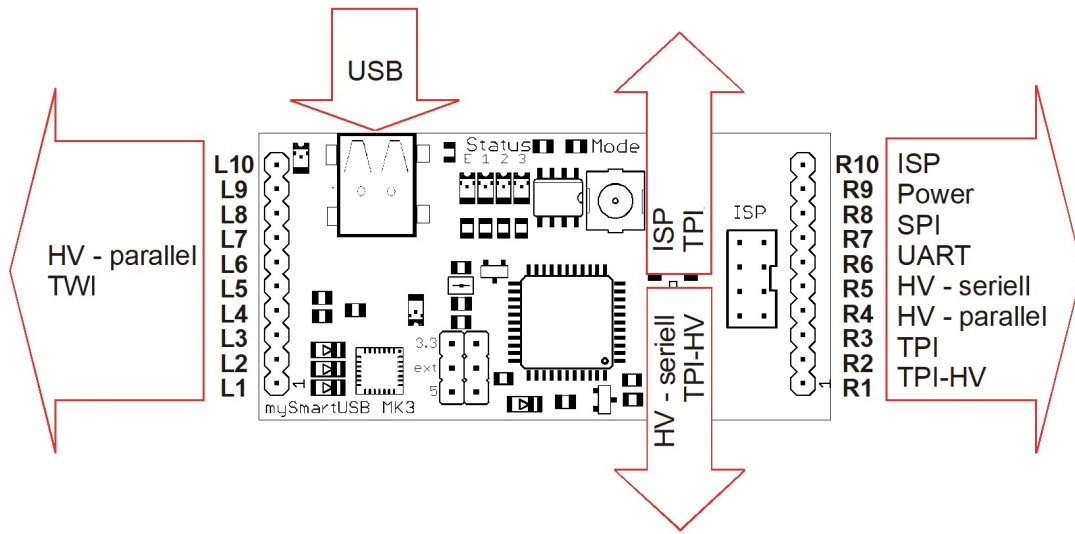
Dimensions of the board (L X B X H)	60 x 30 x 16 mm
Weight	13 g
Grid dimensions	2.54 mm
PCB material	FR8, thickness 1.5 mm, Cu layer 0.35 µm, two-sided, soldering resist mask, tin-plated, plate through

**Anschlüsse**

**Connections**

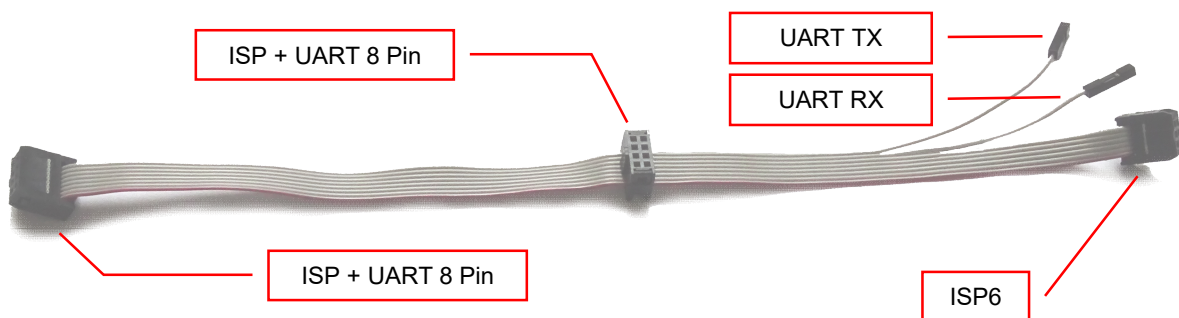
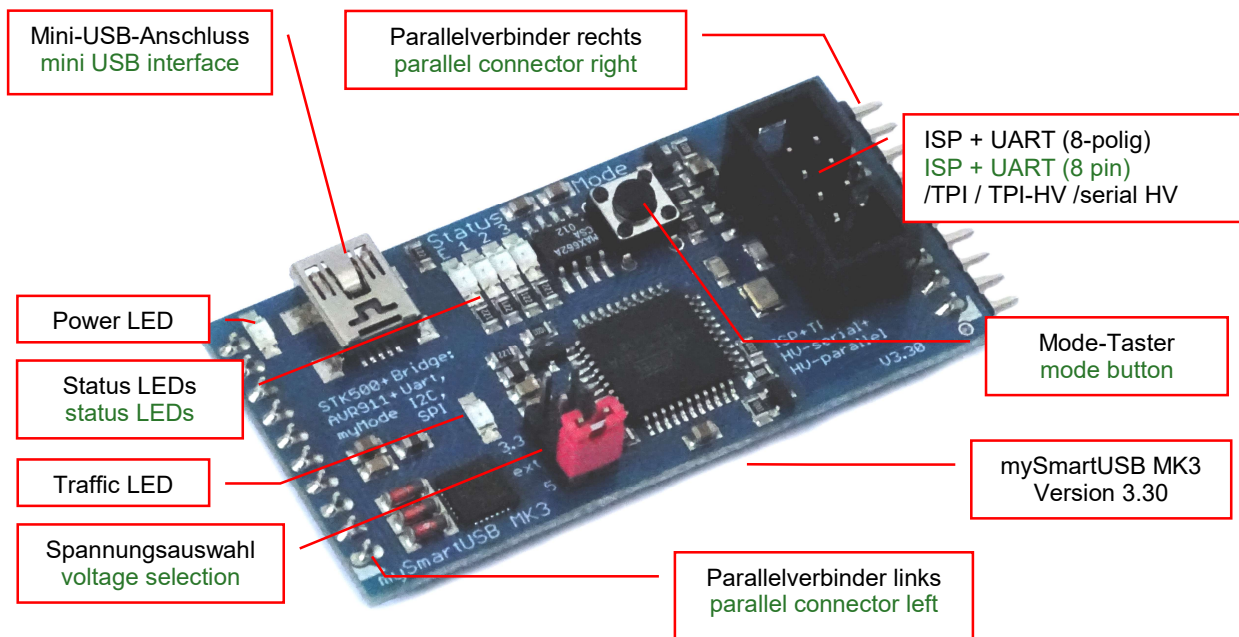
Funktionsschema

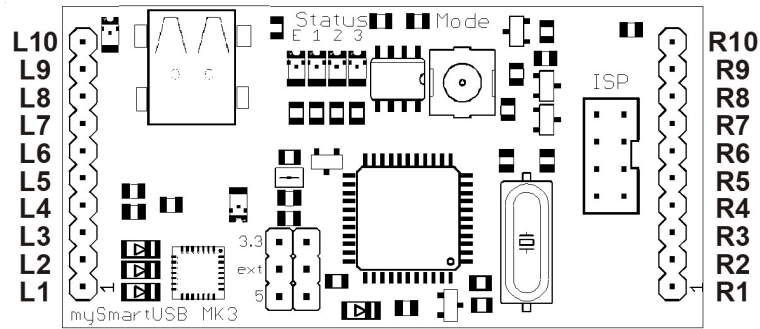
Functional diagram



Aufbau

Structure



**Pinbelegung****pin allocation**

Pin	ISP Programming-Mode	Parallel Programming-Mode	HV-seriell Programming Mode	TPI, TPI-HV Programming Mode	Bridge
L10		READY			I <sup>2</sup> C/TWI-SDA
L9	(rescue clock)	BS2	CLK		I <sup>2</sup> C/TWI-SCL
L8		XTAL1			
L7		DATA6			
L6		DATA5			
L5		DATA4			
L4		XA1			
L3		XA0			
L2		BS1			
L1		DATA7			
R10	GND	GND	GND	GND	GND
R9	VCC	VCC	VCC	VCC	VCC
R8		WR			UART-RxD (PC)
R7		OE			UART-TxD (PC)
R6	Reset	Reset	Reset	Reset	
R5	MOSI	DATA2	SDI		SPI-MOSI
R4	MISO	DATA3	SII	DATA	SPI-MISO
R3	SCK	DATA1	SDO	CLK	SPI-SCK
R2		DATA0	CLK/SCI		SPI-SS
R1		PAGEL			

**8 poliger Wannenstecker****8 pin print plug**

Pin	Pinbelegung / Pin description	
	ISP	TPI, TPI-HV
1	MISO	DATA
2	VCC	VCC
3	SCK	CLK
4	MOSI	-
5	/RESET	RESET
6	GND	GND
7	UART RX	UART RX
8	UART TX	UART TX

Diagram of an 8-pin print plug connector. The pins are numbered 1 to 8 from bottom to top. The connector is shown in a perspective view.

**ISP6 (6 poliger Wannenstecker)****ISP6 (6 pin, print plug)**

Pin	Pinbelegung / Pin description	
	ISP	TPI, TPI-HV
1	MISO	DATA
2	VCC	VCC
3	SCK	CLK
4	MOSI	-
5	/RESET	RESET
6	GND	GND

Diagram of a 6-pin print plug connector. The pins are numbered 1 to 6 from bottom to top. The connector is shown in a perspective view.

LED-Statusanzeige		LED status display			
	mode	LED-Anzeige LED display	Programmier-Mode program-mode	Beschreibung	description
Betriebsmode Operating mode	myMode		In-System-Programmierung In-System-Programming	Im myMode kann der Programmierer durch Befehle über ein Terminalprogramm angesprochen werden (s. Beschreibung des myMode).	In the myMode, the programmer can be controlled by commands, that are sent by a terminal program (commands see myMode-description).
			Hochvolt-parallel High Voltage parallel		
			Hochvolt-seriell High Voltage serial		
			TPI TPI		
			TPI-Hochvolt TPI-High Voltage		
Data-Mode		-	Im Data-Mode leitet der mySmartUSB MK3 alle Daten 1:1 in beide Richtungen weiter.	In the Data-Mode, the programmer directs the data traffic unmodified bidirectional.	
Quiet		-	Der Programmierer verhält sich stumm.	The programmer acts dumb.	
Kompatibilitäts-Mode Compatibility mode	AVR109/911/910		In-System-Programmierung In-System-Programming	In diesem Mode wird durch den mySmartUSB MK3 ein AVR109,911 bzw. 910 Programmierer emuliert.	In this mode, the mySmartUSB MK3 emulates an AVR109, 911 or 910 programmer.
			Hochvolt-parallel High Voltage parallel		
			Hochvolt-seriell High Voltage serial		
			TPI TPI		
			TPI-Hochvolt TPI-High Voltage		
	STK500		In-System-Programmierung In-System-Programming	Wenn sich der mySmartUSB MK3 im STK500-Mode befindet, kann er in den Programmereinstellungen als ein STK500 Programmierer genutzt werden.	In this mode, the mySmartUSB MK3 emulates an STK500 programmer, so in the setup of the IDE must be selected such a programmer.
			Hochvolt-parallel High Voltage parallel		
			Hochvolt-seriell High Voltage serial		
			TPI TPI		
			TPI-Hochvolt TPI-High Voltage		
Bootloader		-	In diesem Modus wird der Bootloader aktualisiert bzw. neu eingeschrieben.	In this mode, the bootloader is updating or reinstalling.	
Error		-	mySmartUSB MK3 hat ein Problem festgestellt und geht für ca. 7 s in den Error-Mode.	mySmartUSB MK3 noticed an error and switch into Error-Mode for circa 7 s.	

= off = on = blink



Die Auswahl des Mode und des Programmier-Mode kann mittels des Tasters erfolgen:  
 kurz tippen = Wechsel des Mode  
 lange drücken = Wechsel des Programmier-Mode

The mode and program-mode can be selected by pushing the button:  
 short tip = change mode  
 hold long = change program-mode

## UART-Konfiguration und Baudraten

## UART-configuration and baud rate

mode	UART	Hinweis / Notes
myMode	500000,8,N,1	
AVR109/911/910	19200,8,N,1	
STK500	115200,8,N,1	
Quiet	500000,8,N,1	nur für mySmartUSB MK3 Protokoll- Kommandos only for mySmartUSB MK3 protocol-commandos
Data-Mode	500000,8,N,1	nur für mySmartUSB MK3 Protokoll- Kommandos only for mySmartUSB MK 3 protocol-commandos
Data-Mode	<b>Data Bits:</b> 5, 6, 7, 8 <b>Stop Bits:</b> 1, 1.5 <sup>(1)</sup> , 2 <b>Parity Type:</b> None, Even, Odd, Mark, Space <b>Baud Rates:</b> 300, 600, 1200, 1800, 2400, 4000, 4800, 7200, 9600, 14400, 16000, 19200, 28800, 38400, 51200, 56000, 57600, 64000, 76800, 115200, 128000, 153600, 230400, 250000, 256000, 460800, 500000, 576000, 921600 <sup>(2)</sup>  <b>Hinweis/Notes:</b> <sup>(1)</sup> nur 5-bit / 5-bit only <sup>(2)</sup> nur 7 oder 8 Datenbits / 7 or 8 data bits only Zusätzliche Baudraten werden unterstützt, vgl. „AN205“ von Silicon Labs Additional baud rates are supported. See “AN205” by Silicon Labs.	



## Beschreibung des myMode

Um die myMode-Befehle nutzen zu können, muss sich der mySmartUSB MK3 im myMode befinden (siehe LED- Statusanzeige).

Für die Kommunikation eignet sich jedes beliebige Terminalprogramm, welches eine Baudrate von 500000 Baud unterstützt. Alle Befehle müssen mit LF (=10, bzw. 0x0A) abgeschlossen werden.

Beachten Sie die exakte Schreibweise und vermeiden Sie unnötige Leerzeichen.

## myMode-description

To use the myMode commands, you need to have the mySmartUSB MK3 in myMode (see LED - Status Display for details).

For the communication, you can use any terminal program, which supports baud rates of 500000 Baud. All commands must be confirmed with LF (=10, resp. 0x0A).

Please note the exact spelling and avoid unnecessary spaces.

Allgemeine Befehle	General Commands
wechselt in angegebenes Menü (prog für Programmieren, main für Hauptmenü)	<pre>m:&lt;menuName&gt; → m:prog ← ok</pre> <p>changes in the specified menu (prog for programming, main for main menu)</p>
listet mögliche Menüs auf	<pre>m:? → m:? ← ok, main, prog, spi, lines, twi</pre> <p>Lists possible menu</p>
listet alle Befehle auf	<pre>? → ? ← ok, burnFlash readFlash ...</pre> <p>Lists all commands</p>
direkte Befehlseingabe, ohne das aktuelle Menü zu verlassen	<pre>@&lt;menuName&gt; &lt;befehl&gt; &lt;para&gt; → @prog ispSpeed scan ← 333333 error ... 294117 ok</pre> <p>Direct command request without leaving the current menu</p>
<b>Menü: main</b>	<b>menu: main</b>
Name, Firmware-Version und Build auslesen	<pre>ver → ver ← ok, mySmartUSB MK3, v 1.08, b1530</pre> <p>read name, firmware-version and build</p>
Reset des Programmers auslösen	<pre>reset → reset ← ok</pre> <p>resets the programmer</p>
in anderen Modus wechseln, z.B. myMode, Data-Mode, avr109/910/911, stk500, quiet Power-On/Off auf angeschlossenem Board	<pre>mode &lt;m d a s q + -&gt; → mode d ← ok</pre> <p>switch into an other mode, e.g. myMode, Data-Mode, avr109/910/911, stk500, quiet Power-on/off on attached board</p>
startet Firmware-Update	<pre>bootloader → bootloader ← ok</pre> <p>starts firmware update</p>
<b>Menü: prog</b>	<b>menu: prog</b>
Das Menü "prog" beinhaltet Befehle für Kommunikation mit dem Zielsystem, u.a. Brenn- und Auslesebefehle.	The menu "prog" includes instructions for communicating with the target system, including Burning and selection commands.
brennt den Flash ab der festgelegten Startadresse mit der angegebenen Länge	<pre>burnFlash &lt;start&gt; &lt;length&gt; → burnFlash 0 300 ← ok, send 256 → &lt;256 Byte Data&gt; ← ok, send 44 → &lt;44 Byte Data&gt; ← ok</pre> <p>burns the flash from the start address with the specified length</p>

liest den Flash von der angegebenen Startadresse mit der festgelegten Länge	<pre>readFlash &lt;start&gt; &lt;length&gt; → readFlash 0 12 ← www.myavr.deok</pre>	reads out the content of the flash with the specified length
Bildet die Summe des Speicherinhaltes von der angegebenen Startadresse mit der angegebenen Länge	<pre>sumFlash &lt;start&gt; &lt;length&gt; → sumFlash 0 12 ← ok, 1209</pre>	sums up the contents of the memory from start to the specified length
brennt den EEPROM ab der festgelegten Startadresse mit der angegebenen Länge	<pre>burnEeprom &lt;start&gt; &lt;length&gt; → burnEeprom 0 262 ← ok, send 256 → &lt;256 Byte Data&gt; ← ok, send 6 → &lt;6 Byte Data&gt; ← ok</pre>	Burn EEPROM from start to specified length
liest den EEPROM von der angegebenen Startadresse mit der festgelegten Länge	<pre>readEeprom &lt;start&gt; &lt;length&gt; → readEeprom 0 20 ← www.myavr.dewww.myavok</pre>	Read EEPROM from start to specified length
Bildet die Summe der Registerinhalte von der angegebenen Startadresse mit der angegebenen Länge	<pre>sumEeprom &lt;start&gt; &lt;length&gt; → sumEeprom 0 12 ← ok, 2057</pre>	sums up the contents of the registers from start to the specified length
Brennt die Fuse- und Lockbits	<pre>burnFuseLock &lt;typ&gt; &lt;byte&gt; → burnFuseLock L 0xff ← ok</pre>	Burns the fuse- and lock-bits
Auslesen der aktuellen Fuse- und/oder Lockbits	<pre>readFuseLock &lt;typ&gt;&lt;typ&gt;... → readFuseLock L123 ← ok, 0xff 0x2a 0xf9 0xff</pre>	Reads out the recent fuse/lock configuration
Schaltet in den Programmiermodus	<pre>enterPm → enterPm ← ok</pre>	Switch into the programming mode
Verlassen des Programmiermodus	<pre>leavePm → leavePm ← ok</pre>	Leave the programming mode
liest die Signatur des Prozessors aus (32 bit Zahl)	<pre>readSig → readSig ← ok, 0x1e9007</pre>	Reads out the signature of the controller (32 bit integer)
festlegen des Programmiermodus i = ISP p = HV-parallel s = HV-seriell t = TPI T = TPI-HV	<pre>progMode &lt;i p s t T&gt; → progMode i ← ok</pre>	Specify the programming mode i = ISP p = HV-parallel s = HV-serial t = TPI T = TPI-HV
Festlegen der ISP-Baudrate (ohne Parameterangabe wird aktuelle Baudrate angegeben)	<pre>ispSpeed &lt;baud&gt; → ispSpeed ← ok, 33222</pre>	Set the ISP baudrate (request without parameters will show current baudrate)
ISP-Baudraten testen	<pre>ispSpeed scan → ispSpeed scan ← 333333 error ... 294117 ok</pre>	scan possible ISP-baud rates
Festlegen der Waits für die parallele Programmierung (ohne Parameterangabe werden aktuelle Waits angegeben)	<pre>parSpeed &lt;waits&gt; → parSpeed 20 ← ok, 20</pre>	determine the waits for parallel programming (without specifying parameters, current waits will be shown)
Geschwindigkeit im parallelen Programmiermodus testen	<pre>parSpeed scan → parSpeed scan ← error 355552 ... ok 182232</pre>	Scans possible parallel-speeds

**Menü: spi**

Das Menü "spi" beinhaltet Befehle für Kommunikation per SPI (Serial Peripheral Interface).  
Es werden jeweils 8 Bit gesendet / empfangen  
(Führende Flanke = steigend = Sample  
SCK = low wenn idle)

**menu: spi**

The menu "spi" includes instructions for the communication via SPI (Serial Peripheral Interface).  
Sending / receiving always 8 bit  
(Leading edge = rising = sample  
SCK = low when idle)

start		
startet die Kommunikation per SPI	→ start ← ok	starts the communication via SPI
send <byte> <byte> ...		
Daten senden	→ send 0x55 12 45	sends data
Antwort = Datenbytes vom Slave	← ok, 0x55 0xa 0xd2	return = databytes from slave
end		
Beendet die Kommunikation per SPI	→ end ← ok	ends the communication via SPI

**Menü: lines**

Das Menü "lines" beinhaltet Befehle für die Konfiguration und Statusabfrage der Anschlussleisten (links und rechts) am Programmer

**menu: lines**

The menu "lines" includes instructions for the configuration and status request of the interface (left and right) from the programmer.

inactive		
setzt alle Lines inaktiv = Input	→ inactive	sets all lines inactive = input
rst <0 1>		
Reset (R6) zurücksetzen (0=low) oder setzen (1=high)	→ rst 1 ← ok, 1	reset (R6) restore (0=low) or set (1=high)
Aufruf ohne Parameter gibt den aktuellen Status zurück	→ rst ← ok, 1	call without parameter returns actual state
11 12 13 14 15 16 17 17 18 19 110 r1 r2 r3 r4 r5 r7 r8 <1 0>		
setzen der Leitung auf 0=low oder 1=high	→ 11 1 ← ok, 1	sets the line to 1=high or 0=low
Aufruf ohne Parameter gibt den aktuellen Status zurück	→ 11 ← ok, 1	call without parameter returns actual state

**Menü: twi**

Das Menü "twi" beinhaltet Befehle für die Kommunikation mittels TWI (I<sup>2</sup>C).  
In geschweiften Klammern ist der Status des Registers TWSR angegeben (siehe TWI Status Codes)

**menu: twi**

The menu "twi" includes instructions for the communication via TWI (I<sup>2</sup>C).  
In cambered brackets the state of the register TWSR is shown (watch TWI state codes).

ima <baud>		
Initialisierung TWI-Master	→ ima ← ok, 50000 {f8}	init TWI-master
Rückgabe passende errechnete Baudrate	→ ima 60000 ← ok, 60240 {f8}	Returns compatible calculated baud rate
sta		
TWI-Start senden	→ sta ← ok, {08}	sends TWI-start
sto		
TWI-Stop senden	→ sto ← ok, {f8}	sends TWI-stop
sla adr <r w>		
setzt Slave-Adresse und Status für nächstes Kommando	→ sla 0x40 r ← ok, {48}	Sets slave-adress and state for next command
r = lesen w = schreiben Adr 0x00...0xFE, Bit0=0	→ sla 0x40 w ← ok, {20}	r = read w = write Adr 0x00...0xFE, Bit0=0

setzen/löschen des bereitgelegten Acknowledge-Bits für die nächste Datenübertragung	ack <0 1> → ack 0 ← ok, {08}	setting/cleaning of the place acknowledge-bits for the next transmission
Bytes per TWI schreiben	wr <byte> <byte> ... → wr 12 45 ← ok, {28}	writes bytes via TWI
x Bytes per TWI lesen	rd <Anzahl Byte> → rd 3 ← ok, 0 255 104 {50}	reads x bytes via TWI
x Bytes lesen + Auto Acknowledge ack=1 vor dem Ersten ack=0 vor dem Letzten	rda <Anzahl Byte> → rda 3 ← ok, 0 255 104 {50}	reads x Bytes + Auto Acknowledge ack=1 before the first ack=0 before the last
TWI-Status ausgeben (beachte TWI Status Codes)	state → state ← ok, 248ok {f8}	returns TWI-state (watch TWI Status Codes)
TWI deaktivieren/beenden	end → end ← ok, {f8}	deactivate/finish the TWI

TWI Status Codes		TWI state codes
Bus-Fehler aufgrund einer unerlaubten START oder STOP-Bedingung	0x00	Bus error due to an illegal START or STOP condition
START Bedingung wurde gesendet	0x08	A START condition has been transmitted
Wiederholter-START-Bedingung wurde gesendet	0x10	A repeated START condition has been transmitted
SLA+W wurde gesendet; ACK wurde empfangen	0x18	SLA+W has been transmitted; ACK has been received
SLA+W wurde gesendet; NOT ACK wurde empfangen	0x20	SLA+W has been transmitted; NOT ACK has been received
Datenbyte wurde gesendet; ACK wurde empfangen	0x28	Data byte has been transmitted; ACK has been received
Datenbyte wurde gesendet; NOT ACK wurde empfangen	0x30	Data byte has been transmitted; NOT ACK has been received
Arbitrage verloren bei SLA+W oder Datenbyte	0x38	Arbitration lost in SLA+W or data byte
Arbitrage verloren bei SLA+R oder NOT ACK Bit	0x38	Arbitration lost in SLA+R or NOT ACK bit
SLA+R wurde gesendet; ACK wurde empfangen	0x40	SLA+R has been transmitted; ACK has been received
SLA+R wurde gesendet; NOT ACK wurde empfangen	0x48	SLA+R has been transmitted; NOT ACK has been received
Datenbyte wurde empfangen; ACK wurde erwidert	0x50	Data byte has been received; ACK has been returned
Datenbyte wurde empfangen; NOT ACK erwidert	0x58	Data byte has been received; NOT ACK has been returned
Eigenes SLA+W wurde empfangen; ACK erwidert	0x60	Own SLA+W has been received; ACK has been returned
Arbitrage verloren bei SLA+R/W als Master; eigenes SLA+W wurde empfangen; ACK wurde erwidert	0x68	Arbitration lost in SLA+R/W as Master; own SLA+W has been received; ACK has been returned
Allg. Sprung-Adresse wurde empfangen, ACK wurde erwidert	0x70	General call address has been received; ACK has been returned
Arbitrage verloren bei SLA+R/W als Master; allg. Sprung-Adresse wurde empfangen, ACK wurde erwidert	0x78	Arbitration lost in SLA+R/W as Master; General call address has been received; ACK has been returned
Vorherig adressiert mit eigenem SLA+W; Daten wurden empfangen, ACK wurde erwidert	0x80	Previously addressed with own SLA+W; data has been received; ACK has been returned
Vorherig adressiert mit eigenem SLA+W; Daten wurden empfangen; NOT ACK wurde erwidert	0x88	Previously addressed with own SLA+W; data has been received; NOT ACK has been returned
Vorherig adressiert mit allg. Sprung-Adresse; Daten wurden empfangen; ACK wurde erwidert	0x90	Previously addressed with general call; data has been received; ACK has been returned
Vorherig adressiert mit allg. Sprung-Adresse; Daten wurden empfangen; ACK wurde erwidert	0x98	Previously addressed with general call; data has been received; NOT ACK has been returned
Eine STOP- oder Wiederholter-START-Bedingung wurde empfangen, obwohl noch als Slave adressiert	0xA0	A STOP condition or repeated START condition has been received while still addressed as Slave
Eigenes SLA+R empfangen; ACK wurde erwidert	0xA8	Own SLA+R has been received; ACK has been returned
Arbitrage verloren bei SLA+R/W als Master; eigenes SLA+R empfangen; ACK wurde erwidert	0xB0	Arbitration lost in SLA+R/W as Master; own SLA+R has been received; ACK has been returned
Datenbyte in TWDR wurde gesendet; ACK wurde empfangen	0xB8	Data byte in TWDR has been transmitted; ACK has been received
Datenbyte in TWDR wurde gesendet; NOT ACK wurde empfangen	0xC0	Data byte in TWDR has been transmitted; NOT ACK has been received
Letztes Datenbyte in TWDR wurde gesendet (TWEA="0"); ACK wurde empfangen	0xC8	Last data byte in TWDR has been transmitted (TWEA= "0"); ACK has been received
Keine wichtigen Statusinformation verfügbar	0xF8	No relevant state information available

**Fehler-Codes**

Bei der Rückgabe der folgenden Werte liegt der dem Code zugeordnete Fehler vor.

**Error Codes**

In return to the following values, the assigned error occurred.

Unbekannter Befehl	error 01	unknown command
Unbekanntes Menü	error 02	unknown menu
Ungültige Daten	error 13	invalid data
Zeitüberschreitung	error 99	timeout
Nicht erfolgreich	error 100	not succeeded
Überprüfung nicht erfolgreich	error 101	incorrect verify
Sonstiger Fehler	error 255	other error

## Das mySmartUSB MK3 Protokoll

Die Firmware des mySmartUSB MK3 verfügt über ein eigenes Protokoll um die zusätzlichen Funktionen des mySmartUSB MK3 anzusprechen. Das mySmartUSB MK3 Protokoll wird über eine Kennung aktiviert die den eigentlichen Kommandos als Präfix vorangestellt werden muss.

**Baudrate:** je nach Mode, z.B. 500000,8,n,1 für myMode, Quiet, Data  
andere Baudraten siehe „UART-Konfiguration und Baudraten“

**Präfix:** "æµ<sup>0123</sup>@"+Kommando als ASCII-Zeichen  
(= 0xE6 0xB5 0xBA 0xB9 0xB2 0xB3 0xA9)

### Kommandos:

+	Board-Power-On
-	Board-Power-Off
r	Reset Board
i	Status wird zurückgegeben
I	ProgMode wird zurückgegeben
m	myMode
p	Programmiermodus
d	Daten-Bypass/ Durchreich-Modus UART
s	STK500-Modus
a	AVR910-Modus
q	Quiet-Modus
v	Versionsnummer wird zurückgegeben
C	Rescue-Clock an
c	Rescue-Clock aus
E	aktueller Mode und Progmode werden in EEPROM geschrieben

**Beispiel:** æµ<sup>0123</sup>@q schaltet den Programmer in den quiet-mode

**Hinweis:** Verwenden Sie die aktuelle Firmware des mySmartUSB MK3.

## The mySmartUSB MK3 protocol

The firmware of the mySmartUSB MK3 uses an own protocol to perform additional tasks. The mySmartUSB MK3 protocol will be activated over a specific character string that has to precede the instructions.

**baud rate:** depending on mode, e.g. 500000,8,n,1 for myMode, Quiet, Data  
other baud rates see "UART-configuration and baud rate"

**prefix:** "æµ<sup>0123</sup>@"+instruction in ASCII code  
(= 0xE6 0xB5 0xBA 0xB9 0xB2 0xB3 0xA9)

### commands:

+	board power on
-	board power off
r	reset board
i	return state
I	return ProgMode
m	myMode
p	programming-mode
d	data-bypass mode
s	STK500-mode
a	AVR910-mode
q	quiet mode (neutral)
v	return version
C	Rescue-Clock on
c	Rescue-Clock off
E	write current mode and Prog-Mode to EEPROM

**Example:** æµ<sup>0123</sup>@q switches the programmer into the quiet-mode

**Notice:** Please use the current firmware of mySmartUSB MK3.

### Programmereinstellungen

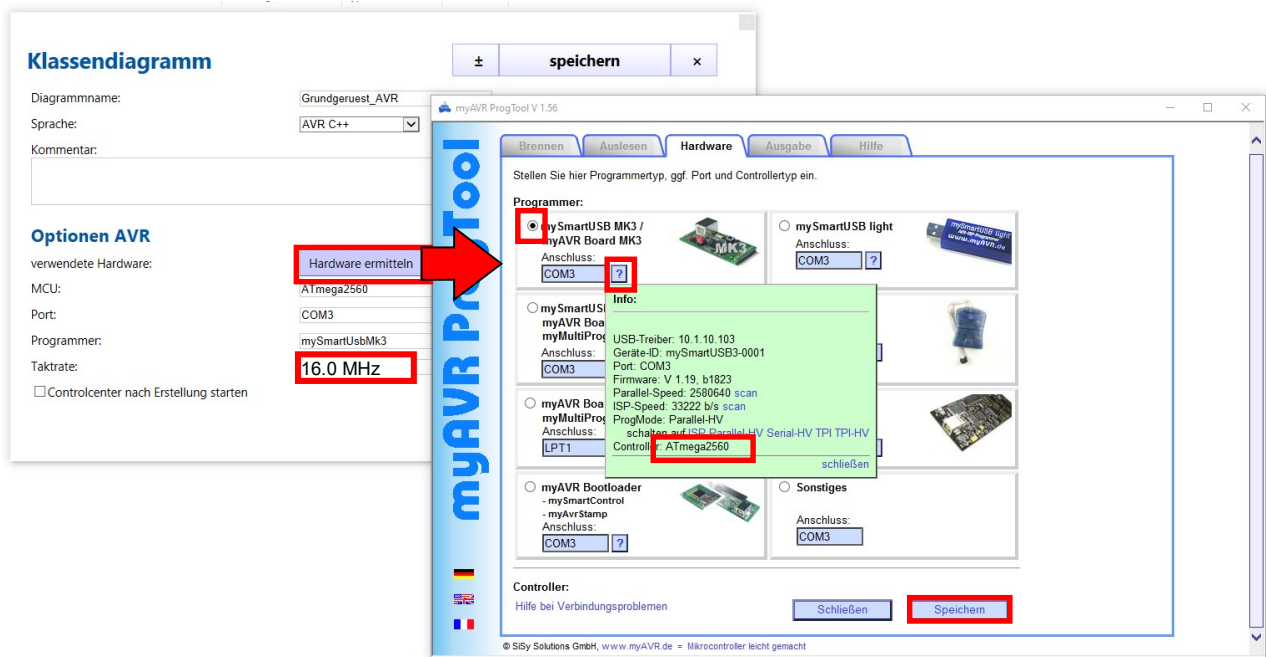
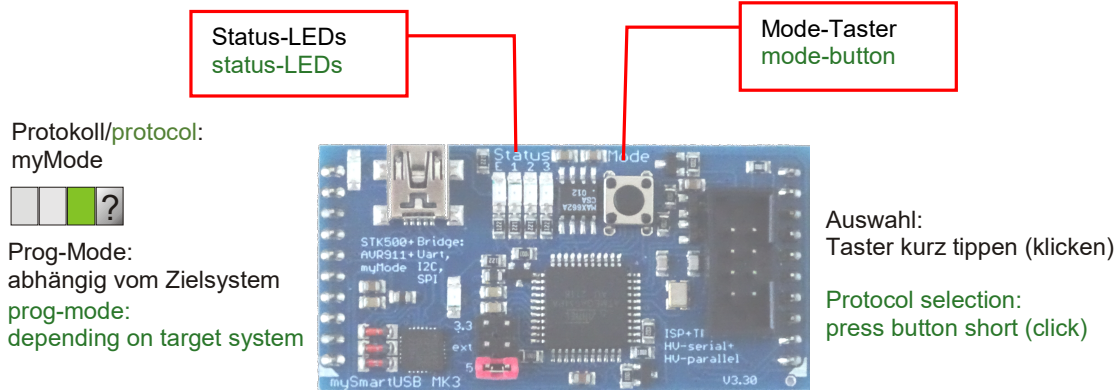
#### Programmereinstellungen SiSy (ab 3.32)

Versetzen Sie den USB-Programmer mySmartUSB MK3 am Besten in den myMode (siehe oben). Starten Sie SiSy und legen Sie ein neues Projekt an. Beim „kleinen Programm“ und beim Klassendiagramm wählen Sie mit der rechten Maustaste „Definieren...“. Im sich öffnenden Fenster klicken auf *Hardware ermitteln*. Im myAVR ProgTool wählen Sie den USB Programmer mySmartUSB MK3 und klicken auf *Test [?]*. Übernehmen Sie die Einstellungen, wenn das Test-Fenster grün ist. SiSy übernimmt den Programmer, den eingestellten Controller und den COM-Port. Klicken Sie auf den erkannten Controller und anschließend auf *Speichern*. Die Taktfrequenz muss manuell eingegeben werden. Für das Beispiel wurde das myAVR Board MK3 einem ATmega2650 verwendet.

### Programmer settings

#### changing setting in SiSy (from 3.32)

It is best to put the mySmartUSB MK3 USB programmer in myMode (see above). Start SiSy and create a new project. For the “small program” and the class diagram, select “Define...” with the right mouse button. In the window that opens, click on Detect Hardware. In the myAVR ProgTool, select the USB programmer mySmartUSB MK3 and click on Test [?]. Apply the settings when the test window is green. SiSy takes over the programmer, the set controller and the COM port. Click the discovered controller and then click Save. The clock frequency must be entered manually. The myAVR Board MK3 and an ATmega2650 were used for the example.



Nach dem *Speichern* überprüfen Sie bitte, dass die Einstellungen korrekt übernommen wurden.

After saving, please check that the settings have been applied correctly.


**Programmereinstellungen in myAVR Workpad (1.6)**

Versetzen Sie den USB-Programmer mySmartUSB MK3 in den myMode (näheres dazu finden Sie im Abschnitt zur Tochterplatine mySmartUSB MK3, sowie der Technischen Beschreibung zum mySmartUSB MK3). Starten Sie myAVR Workpad und wählen Sie im Menüpunkt *Extras/Einstellungen* den mySmartUSB MK3 Programmer aus, klicken Sie auf *Test*. Den gefunden Controller übernehmen Sie, indem Sie diesen anklicken. Schließen Sie die Einstellungen mit der Schaltfläche *Speichern* ab. Für das Beispiel wurde das myAVR Board MK3 256K verwendet.

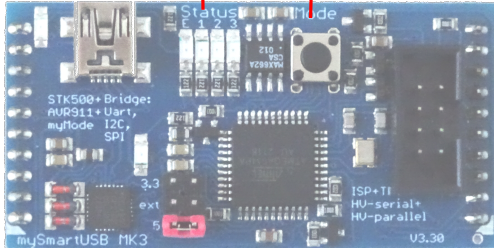
**changing setting in myAVR WorkPad (1.6)**

Put the mySmartUSB MK3 into the myMode compatibility modus (you can find more information about this in the section on the mySmartUSB MK3 daughter board and the technical description for the mySmartUSB MK3). Choose in the menu under *Extras/Einstellungen* the mySmartUSB MK3 programmer and click on the *Test* Button. Click on the controller to take this. In the example was used the myAVR Stamp 256K PLUS with an ATmega2560. To complete the attitudes, use the *Speichern* button.

Protokoll/protocol: myMode



Prog-Mode: abhängig vom Zielsystem  
prog-mode: depending on target system



Status-LEDs status-LEDs

Mode-Taster mode-button

Auswahl: Taster kurz tippen (klicken)

Protocol selection: press button short (click)

**Optionen - myAVR Workpad PLUS**

Hardware Programm

Stellen Sie hier Programmertyp, ggf. Port und Controllertyp ein.

**Programmer:**

<input checked="" type="radio"/> mySmartUSB MK3 / myAVR Board MK3 Anschluss: COM4 ?	<input type="radio"/> mySmartUSB light Anschluss: COM3 ?
<input type="radio"/> mySmartUSB myAVR Board myMultiProg Anschluss: COM5	<input type="radio"/> AVR ISP mk-II Anschluss: usb:48.74 ?
<input type="radio"/> myAVR Board myMultiProg Anschluss: LPT1 ?	<input type="radio"/> AVR Dragon Anschluss: usb:3e.65 ?
<input type="radio"/> myAVR Bootloader - mySmartControl - myAvrStamp Anschluss: COM3 ?	<input type="radio"/> Sonstiges Anschluss:

**Controller:**

Abbrechen Standard laden **Speichern**


**Programmereinstellungen in AVRDUDE (5.8)**

Versetzen Sie den mySmartUSB MK3 Programmierer in den Mode AVR 109,911,910 (näheres dazu finden Sie im Abschnitt zur Tochterplatine mySmartUSB MK3, sowie der Technischen Beschreibung zum mySmartUSB MK3). Starten Sie AVRDUDE über die Kommandozeile. Es werden Ihnen alle verfügbaren Befehle angezeigt. Hier ein Ausschnitt.

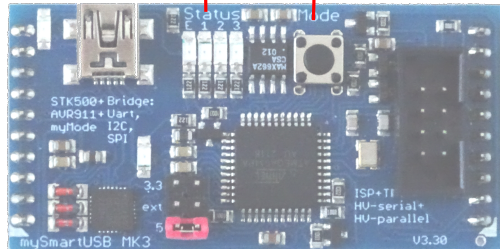
**changing setting in AVRDUDE (5.8)**

Put the mySmartUSB MK3 into the AVR 109,911,910 compatibility modus (you can find more information about this in the section on the mySmartUSB MK3 daughter board and the technical description for the mySmartUSB MK3). After that start the AVRDUDE with the *command line*. That will show you all instructions, which are available. Here a clipping.

Protokoll/protocol:  
STK500



Prog-Mode:  
abhängig vom Zielsystem  
prog-mode:  
depending on target system



Mode-Taster  
mode-button

Auswahl:  
Taster kurz tippen (klicken)

Protocol selection:  
press button short (click)

```

C:\Windows\system32\cmd.exe
C:\>avrdude
Usage: avrdude [options]
Options:
-p <partno>           Required. Specify AVR device.
-b <baudrate>        Override RS-232 baud rate.
-B <bitclock>        Specify JTAG/STK500v2 bit clock period (us).
-C <config-file>     Specify location of configuration file.
-c <programmer>      Specify programmer type.
-D                   Disable auto erase for flash memory
-i <delay>           ISP Clock Delay [in microseconds]
-P <port>            Specify connection port.
-F                   Override invalid signature check.
-e                   Perform a chip erase.
-O                   Perform RC oscillator calibration (see AVR053).
-U <memtype>:r|w|v:<filename>[:format]
                    Memory operation specification.
                    Multiple -U options are allowed, each request
                    is performed in the order specified.
-n                   Do not write anything to the device.
-U                   Do not verify.
-u                   Disable safemode, default when running from a scrip
    
```

Hier sehen Sie einen Beispielbefehl zum Brennen eines Beispielprogramms.

Here you see a command example to burn a program.

```

C:\>avrdude.exe -p ATmega168 -e -c avr911 -Uflash:w:Beispiel.hex:i -P com5
Connecting to programmer: .
Found programmer: Id = "AVR ISP"; type = S
    Software Version = 3.0; Hardware Version = 3.0
Programmer supports auto addr increment.
Programmer supports buffered memory access with buffersize=512 bytes.
Programmer supports the following devices:
    Device code: 0x01
    Device code: 0x09
    
```

```

avrdude.exe: verifying ...
avrdude.exe: 0 bytes of flash verified
avrdude.exe: safemode: Fuses OK
avrdude.exe done. Thank you.
    
```



**Programmereinstellungen in BASCOM (1.11.9.8)**

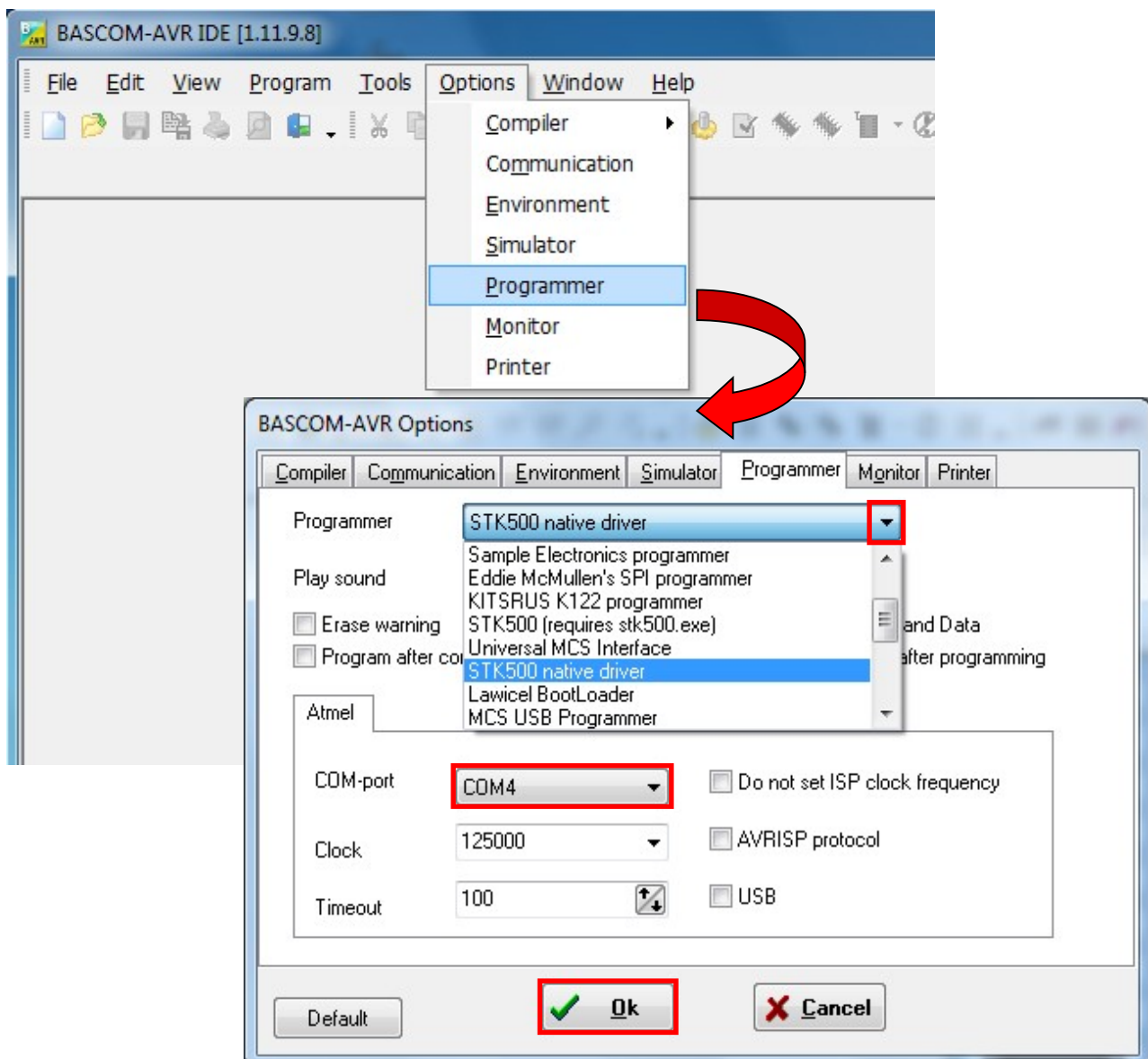
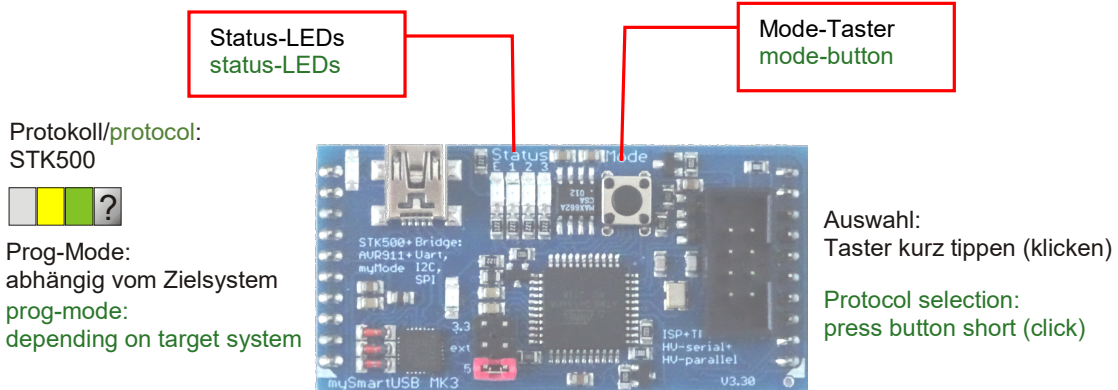
Versetzen Sie den mySmartUSB MK3 Programmierer in den STK500 Mode (näheres dazu finden Sie im Abschnitt zur Tochterplatine mySmartUSB MK3, sowie der Technischen Beschreibung zum mySmartUSB MK3).

Unter dem Menüpunkt *Options/Programmer* können Sie die Einstellungen für den Programmierer vornehmen. Hierzu wählen Sie, wie auf den Bildern ersichtlich, den Modus STK500 extended, sowie den COM-Port aus. Abschließend klicken Sie auf *Ok* und Ihre Einstellungen werden übernommen.

**changing setting in BASCOM (1.11.9.8)**

Put the mySmartUSB MK3 into the STK 500 compatibility modus (you can find more information about this in the section on the mySmartUSB MK3 daughter board and the technical description for the mySmartUSB MK3).

You can carry out the attitudes for the Programmer under the menu item *Options/Programmer*. Then you will select the STK500 extended mode and as well as the respective COM port. In conclusion click on *Ok* and her attitudes were taken on.



**Programmereinstellungen in Codevision (2.04.8)**

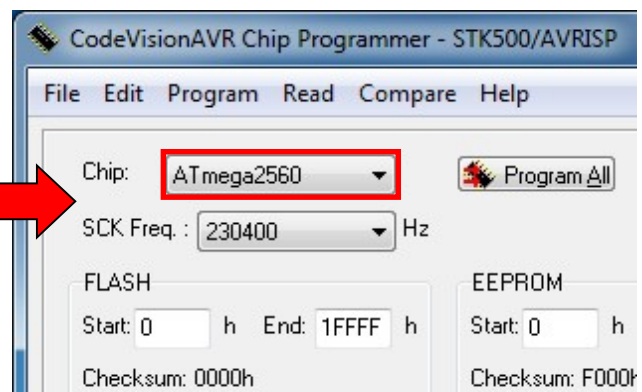
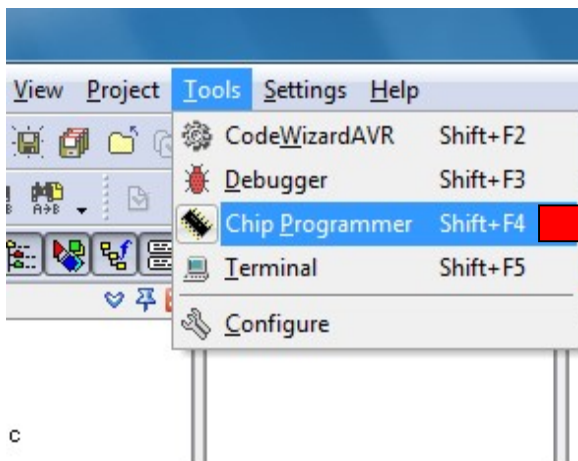
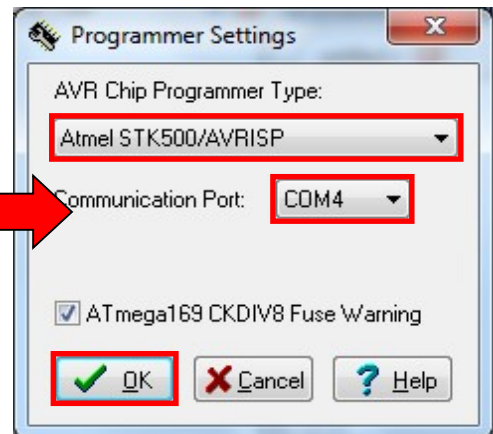
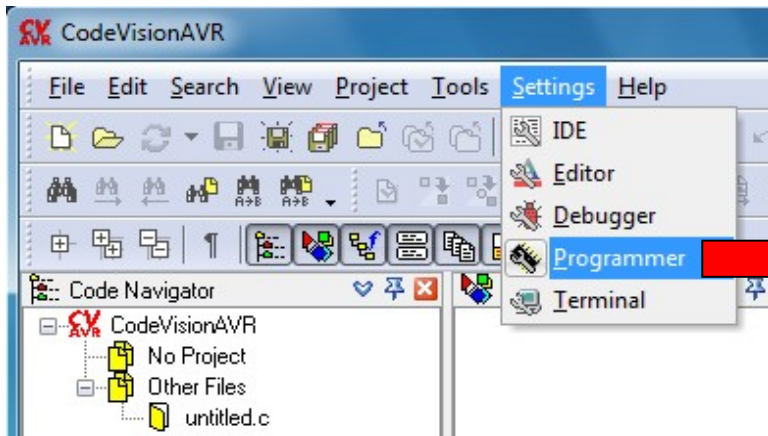
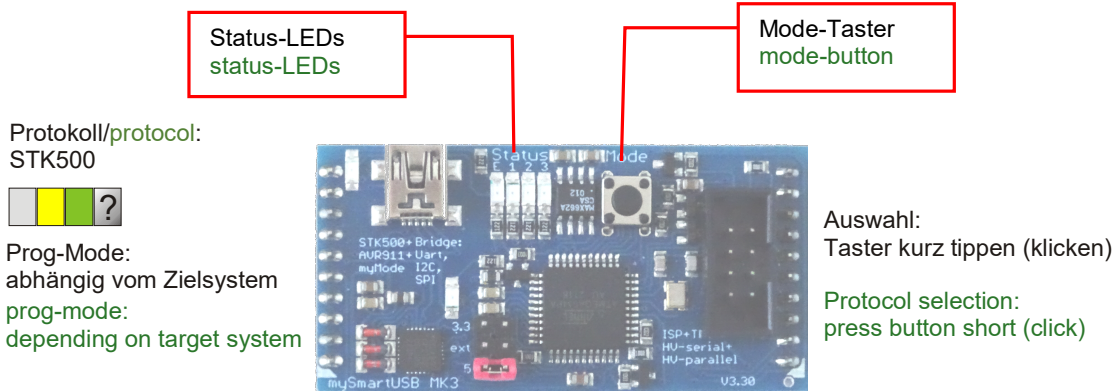
Versetzen Sie den mySmartUSB MK3 Programmer in den STK500 Mode. (näheres dazu finden Sie im Abschnitt zur Tochterplatine mySmartUSB MK3, sowie der Technischen Beschreibung zum mySmartUSB MK3).

Im Menüpunkt *Settings/Programmer* nehmen Sie als erstes die Einstellungen für den mySmartUSB MK3 Programmer vor. Hierzu wählen Sie den Mode STK500, sowie den Port. Bestätigen Sie die Einstellungen. Danach wählen Sie unter dem Menüpunkt *Tools/Chip Programmer* den passenden Prozessor aus, entsprechend dem Prozessor auf der myAVR Stamp und bestätigen ebenfalls. Im Beispiel wurde die myAVR Stamp 256K PLUS mit einem ATmega2650 verwendet.

**changing setting in Codevision (2.04.8)**

Put the mySmartUSB MK3 into the STK 500 compatibility modus. (you can find more information about this in the section on the mySmartUSB MK3 daughter board and the technical description for the mySmartUSB MK3).

After that, you select in the menu Settings/ Programmer and you carry out the attitudes for the mySmartUSB MK3 Programmer. For that you choose the STK500 modus as well as the port. Confirm at the end. After that you configure the Programmer under the menu item Tools/Chip Programmer, according to the processor on the myAVR Stamp and confirm. In the example was used the myAVR Stamp 256K PLUS with an ATmega2650.

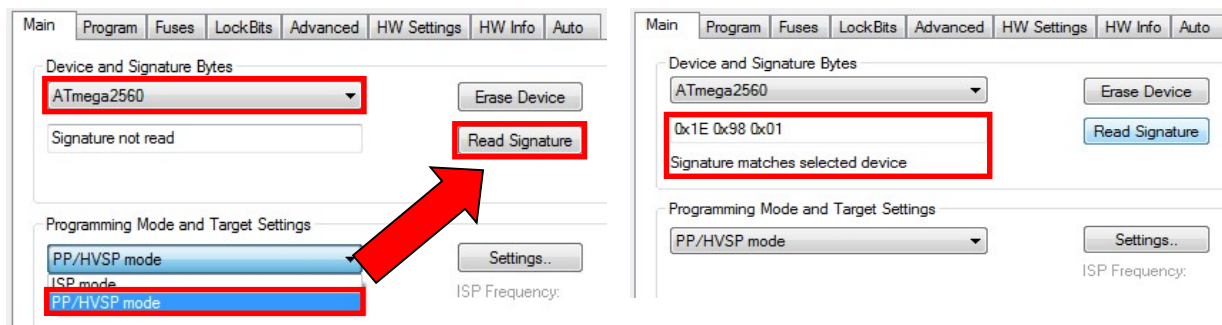
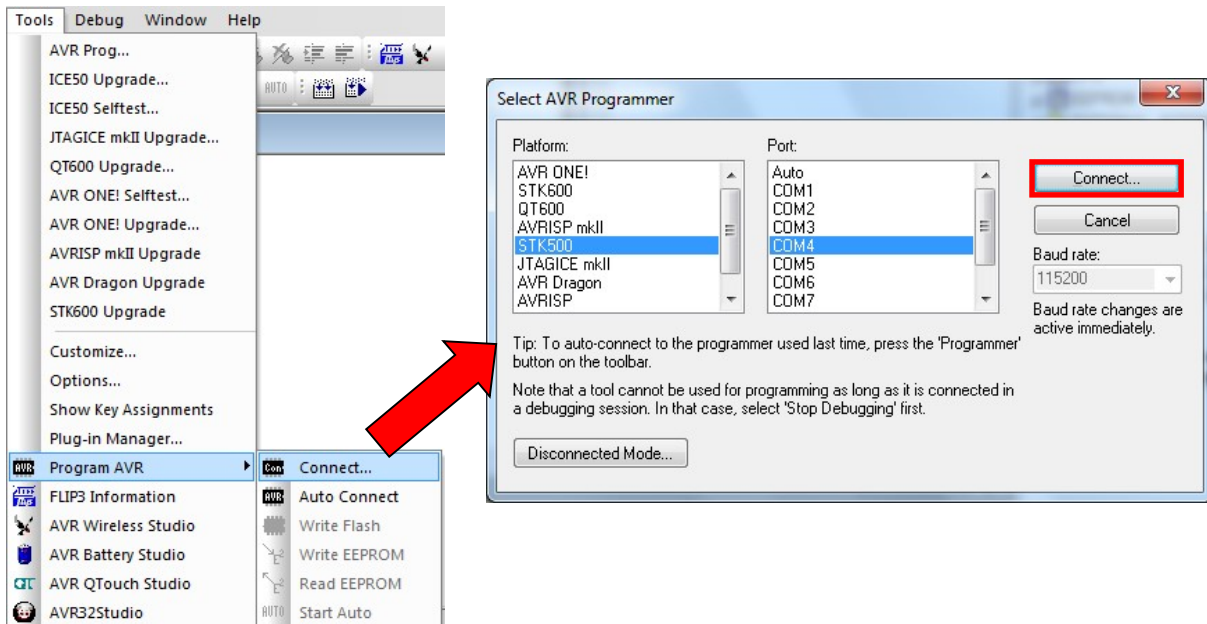
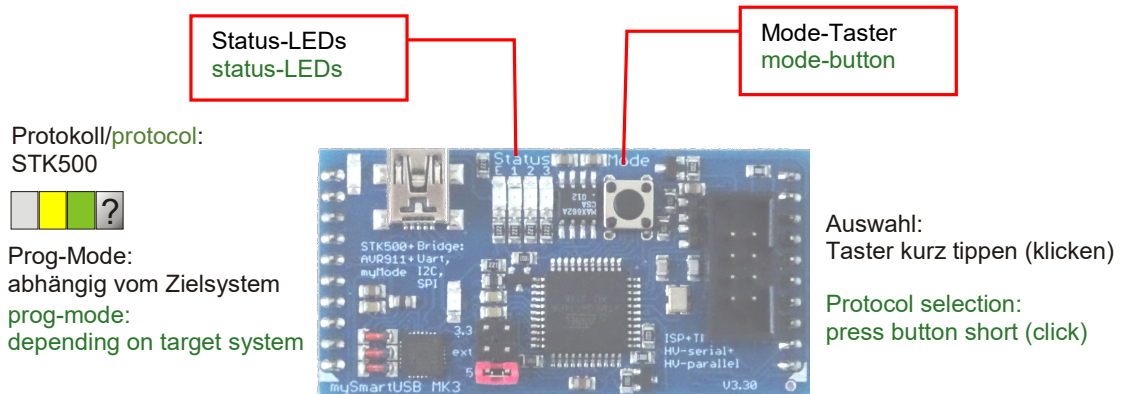


**Programmereinstellungen AVR Studio (4.18.700)**

Schließen Sie das myAVR Board MK3 an das Rechner-system an und versetzen den mySmartUSB MK3 Programmer in den Mode STK500 (näheres dazu, finden Sie im Abschnitt zur Tochterplatine mySmartUSB MK3, sowie der Technischen Beschreibung zum mySmartUSB MK3). Starten Sie AVR Studio und wählen Sie den Menüpunkt *Tools/Programm AVR/Connect...* Wählen Sie hier, unter „Plattform“ STK500, sowie den Port. Mithilfe der Schaltfläche *Connect* wird nun die Hardware konfiguriert. Abhängig von der myAVR Stamp können Sie den Controller auswählen. Im Beispiel wurde die myAVR Stamp 256K PLUS mit einem ATmega2650 verwendet. Anschließend ändern Sie „Programming Mode“ und speichern die Einstellungen über die Schaltfläche *Read Signatur*. Als Ergebnis sehen Sie die Signatur der Hardware (vgl. Bild). Programme können Sie nun in der Registerkarte „Program“ mit Hex-Files laden.

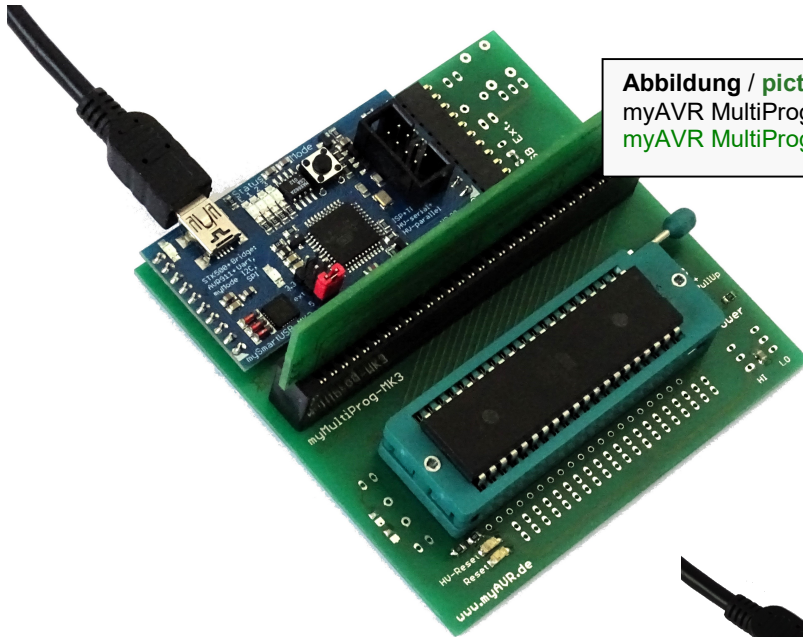
**changing setting in AVR Studio (4.18.700)**

At first you attach the myAVR Board MK3 to the computer system and put the mySmartUSB MK3 into the STK500 modus (you can find more information about this in the section on the mySmartUSB MK3 daughter board and the technical description for the mySmartUSB MK3). Start the AVR Studio and choose in the menu *Tool/Programm AVR/Connect...* Now you select in the platform list “STK500”, your port and press on the “Connect” Button. The Software search the myAVR Board MK3. Addicted from the myAVR Stamp you can select the processor. In the example the myAVR Stamp 256K was used plus with an ATmega2650. After that, you must be changed the “programmer mode” and press the Read Signature button, to safe the settings. The results of that, you see the signature of the hardware (cf. picture). You can now load programs with hex files in the “Program” tab.

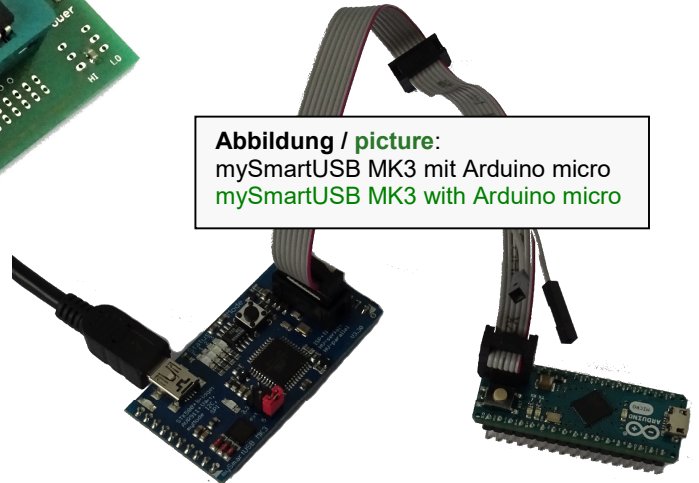


## Anwendungsbeispiele

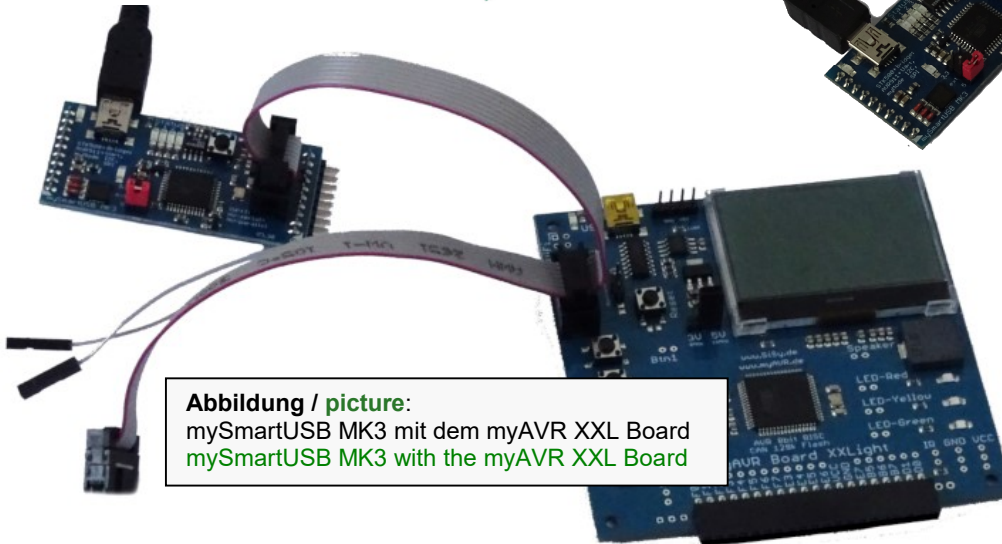
## Exemples of use



**Abbildung / picture:**  
myAVR MultiProg MK3 mit mySmartUSB MK3  
myAVR MultiProg MK3 with mySmartUSB MK3



**Abbildung / picture:**  
mySmartUSB MK3 mit Arduino micro  
mySmartUSB MK3 with Arduino micro



**Abbildung / picture:**  
mySmartUSB MK3 mit dem myAVR XXL Board  
mySmartUSB MK3 with the myAVR XXL Board

### Allgemeine Sicherheitshinweise

Grundsätzlich ist mySmartUSB MK3 nur zum Einsatz unter Lern- und Laborbedingungen konzipiert. Er ist nicht vorgesehen und nicht dimensioniert zur Steuerung realer Anlagen. Bei vorschriftsmäßigem Anschluss und Betrieb treten keine lebensgefährlichen Spannungen auf. Beachten Sie trotzdem die Vorschriften, die beim Betrieb elektrischer Geräte und Anlagen Gültigkeit haben. Wir versichern, dass die Leiterplatte durch den Hersteller getestet wurde. Für fehlerhaften und/oder vorschriftswidrigen Einsatz des Boards übernehmen wir keine Garantie.

### Safety Guidelines

mySmartUSB MK3 is designed for educational and experimental use only. It is not intended and not dimensioned to control real industrial facilities. At correct use there will not occur extremely dangerous voltages. Nevertheless, be aware of general guidelines for using electronic devices. We assure that the PCB has been tested by the producer. For incorrect use and/or application contrary to technical regulations we are not liable.

Die aktuellen Dokumente zum mySmartUSB MK3 finden Sie unter [www.myAVR.de](http://www.myAVR.de) im Downloadbereich.

The current documents for the mySmartUSB MK3 can be found at [www.myAVR.com](http://www.myAVR.com) in the download area.



Abbildungen können vom Inhalt abweichen. Änderungen im Sinne des technischen Fortschrittes behält sich der Hersteller vor.

Images may vary from the content. The manufacturers retains changes in terms of technical advances.