

AVR[®] STK500

Hardwarebeschreibung

Haftungsausschluß:

Der Autor dieser Übersetzung kann nicht für die Fehlerfreiheit dieses Textes garantieren und übernimmt keine Haftung für Schäden die durch Übersetzungsfehler sowie mit übersetzten Fehlern direkt oder indirekt an AVR Mikrocontrollern, dem STK500 oder allen anderen in dieser Bedienungsanleitung erwähnten Geräten und Einrichtungen entstehen.

Des weiteren weise ich ausdrücklich darauf hin, daß dieser Text nicht im Auftrag der Firma Atmel entstanden ist, diese also für Fehler und daraus entstandene Schäden nicht Haftbar gemacht werden kann.

Sollten Sie Fehler in diesem Text finden können Sie mir diese per Mail an mikrosoft@yahoo.de mitteilen.

Ebenso Anregungen, Kritik oder Fragen.



Inhaltsverzeichnis

Kapitel 1 • Einführung	3
1.1 Starter-Kit Ausstattungsmerkmale	3
1.2 Unterstützte Controller	4

Kapitel 2 • Inbetriebnahme	5
2.1 Auspacken	5
2.2 System Voraussetzung	5
2.3 Schnell-Start	6
2.3.1 Anschließen der Hardware	7
2.3.2 Programmieren der Ziel-Controller	7

Kapitel 3 • Hardware Beschreibung	9
3.1 Beschreibung der benutzbaren LED's	9
3.2 Beschreibung der Schalter	10
3.3 Verbinden von LED's und Schaltern	11
3.4 Port Anschlüsse	11
3.5 Beschreibung der RS-323 Benutzerschnittstelle	13
3.6 Beschreibung der DataFlash Pins (nur ältere Modelle des STK500!)	14
3.7 Die Controller-Sockel	16
3.7.1 ISP Programmierung	17
3.7.2 High-voltage Programmierung	19
3.7.2.1 Parallele High-voltage Programmierung	20
3.7.2.2 Serielle High-voltage Programmierung	22
3.8 Jumper Settings	23
3.8.1 V_{cc} Einstellung, VTARGET	24
3.8.2 Analoge Referenzspannung, AREF	25
3.8.3 Reset Einstellungen, RESET	26
3.8.4 Takteinstellung, XTAL1 und OSCSEL	28
3.8.5 BSEL2-Jumper	30
3.8.6 PJUMP-Jumper	30
3.9 Erweiterungs-Anschlüsse	31
3.9.1 Signalbeschreibung	33
3.10 Prog Ctrl und Prog Data Anschlüsse	33
3.11 Sonstiges	34
3.11.1 Reset-Button	34
3.11.2 PROGRAM-Button	35
3.11.3 Main Power LED	35
3.11.4 Target Power LED	35
3.11.5 Status LED	35

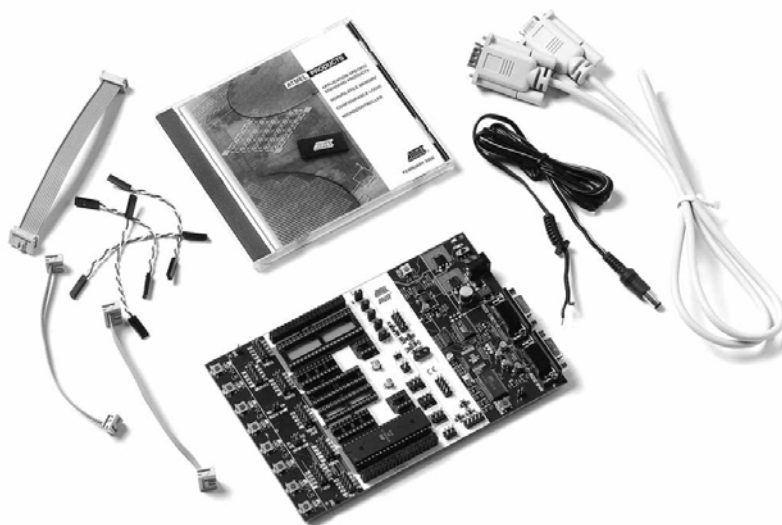
Herzlichen Glückwunsch zum Erwerb des AVR® STK500 Flash Mikrocontroller Starter-Kit. Das STK500 ist ein komplettes Start-Paket mit einer Entwicklungsumgebung für die AVR Flash Mikrocontroller von Atmel. Das Starter-Kit wurde entworfen, um Programmierern einen schnellen Einstieg zum Programmieren und das Erstellen neuer Schaltungen zu ermöglichen.

1.1 Starter-Kit Ausstattungsmerkmale

- AVR Studio kompatibel
- RS-232 Schnittstelle zum PC zur Programmierung und Kommunikation
- Spannungsregler zum Anschluß von 10-15V Gleichstrom
- Sockel für 8-, 20-, 28-, und 40-pin AVR Controller
- Parallele und Serielle High-voltage Programmierung der AVR-Controller
- Serielles In-System Programming (ISP) von AVR's
- In-System Programmierer zum Programmieren von AVR Controllern in externen Schaltungen
- Neu-/Umprogrammierung von AVR Controllern
- 8 Schalter zur freien Benutzung
- 8 LED's zur freien Benutzung
- Alle AVR E/A-Ports sind leicht erreichbar auf Stiftleisten geführt
- Extra RS-232 Schnittstelle zur freien Benutzung
- Erweiterungs-Schnittstelle für Einsteckmodule
- On-Board 2-Mbit DataFlash zum stromlosen Erhalt von Daten (Anm.: Bei neueren STK500 Versionen nicht mehr vorhanden)

Das STK500 wird nur von AVR Studio in der Version 3.2 oder höher unterstützt. Für aktuelle Informationen über dieses und andere AVR Entwicklungswerkzeuge lesen Sie bitte das Dokument „avrtools.pdf“. Die neueste Version von AVR Studio, „avrtools.pdf“ und der Original Bedienungsanleitung (nicht diese Übersetzung!) finden Sie in der AVR Sektion auf der Atmel Website, www.atmel.com.

Abbildung 1.1: STK500



1.2 Unterstützte Controller

Die Systemsoftware unterstützt zur Zeit folgende AVR's in allen Geschwindigkeitsstufen (MHz):

- ATtiny11
- ATtiny12
- ATtiny15
- ATtiny22
- ATtiny28
- AT90S1200
- AT90S2313
- AT90S2323
- AT90S2333
- AT90S2343
- AT90S4414
- AT90S4433
- AT90S4434
- AT90S8515
- AT90S8535
- ATmega8
- ATmega16
- ATmega161
- ATmega163
- ATmega323
- ATmega103⁽¹⁾
- ATmega128⁽¹⁾

¹ Diese AVR's passen nicht in die Sockel des STK500 und können nur mit dem Zusatzboard STK501 oder anderen externen Schaltungen verwendet werden.

2.1 Auspacken

Inhalt des Starter-Kit:

- STK500 Starter-Kit Evaluation Board
- Kabel: 2x 10-Adriges Flachbandkabel für E/A-Ports und parallelen Programmiermodus
1x 6-Adriges Flachbandkabel für In-System Programming
4x 2-Adriges Kabel zum Anschluß des UART und der DataFlash Verbindungen
- RS-232 Kabel mit 9-poligem SubD-Stecker/Buchse
- Stromanschlußkabel (kein Netzteil)
- Atmel CD-Rom mit Datenblättern und Software
- AT90S8515-8PC Beispiel Mikrocontroller

2.2 System Voraussetzung

Die minimalen Hard- und Softwarevoraussetzungen sind:

- 486 Prozessor (Pentium empfohlen)
- 16 MB RAM
- 12 MB freier Festplattenspeicher
- Windows[®] 95/98/2000/ME oder Windows NT[®] 4.0 oder höher
- 115200 Baud RS-232 Anschluß (COM Port)
- 10-15V Gleichstrom Netzteil mit min. 500mA

Das STK500 wird mit einem AT90S8515-8PC AVR im Sockel mit der Bezeichnung SCKT3000D3 geliefert. Die Standard JumperEinstellung verbindet diesen Sockel mit der Taktquelle und dem Spannungsregler die sich auf dem Board befinden.

Der AVR ist mit einem Testprogramm programmiert, das die LEDs blinken läßt. Dieses Testprogramm entspricht dem Beispielcode, der in Kapitel 9 beschrieben wird. Verbinden Sie die LED's und Taster und schalten das STK500 ein, um das Testprogramm im 8515 zu starten.

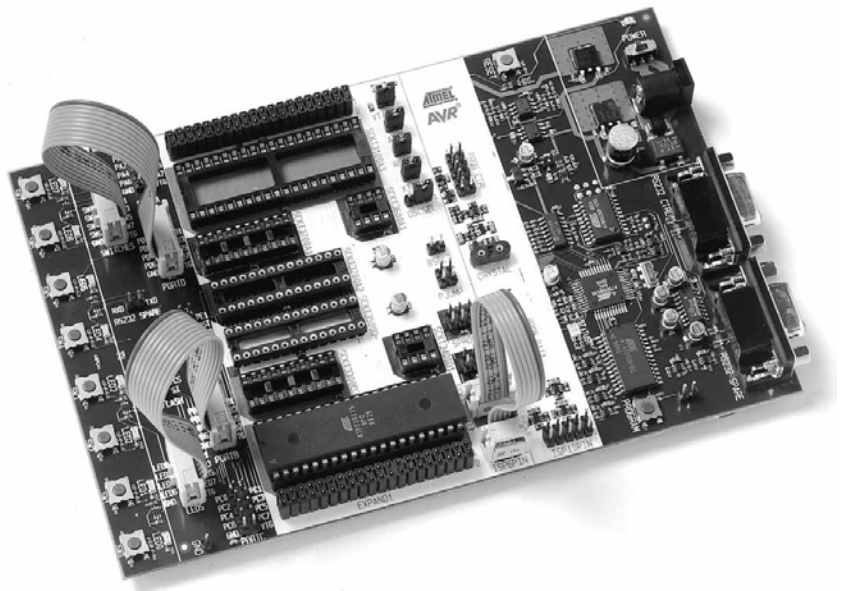
Benutzen Sie dazu die beiliegenden 10-Adrigen Flachbandkabel um die Anschlüsse „PORTB“ mit „LEDS“ und „PORTD“ mit „SWITCHES“ zu verbinden. Die Verbindungen werden in Abb. 2.1 gezeigt.

Benötigt wird außerdem ein externes 10-15V Gleichstrom-Netzteil. In der Eingangsschaltung befindet sich ein Brückengleichrichter und kann mit Positiv- wie auch mit Masse-Mittelkontakt umgehen.

Wenn der Mittelkontakt auf V+ gelegt ist, kann es unter Umständen unmöglich sein das STK500 auszuschalten, da das Board die Masse nicht verbindet. In diesem Fall kann das Board entweder über das RS-232 Kabel oder eine andere Quelle mit Masse verbunden werden.

Schließen Sie nun die Stromversorgung mit einem externen Netzteil an. Über den Power-Schalter kann nun das Board ein- und ausgeschaltet werden. Die rote LED leuchtet, wenn das Board eingeschaltet ist, und die Status-LED wechselt von rot über gelb nach grün. Die grüne LED zeigt an, daß am AVR die Zielspannung V_{CC} anliegt. Das im Programmspeicher des AVR befindliche Programm läuft nun. In diesem Fall (das o.g. Testprogramm) reagiert es auf Schalterdrücken durch Umschalten der dazugehörigen LED.

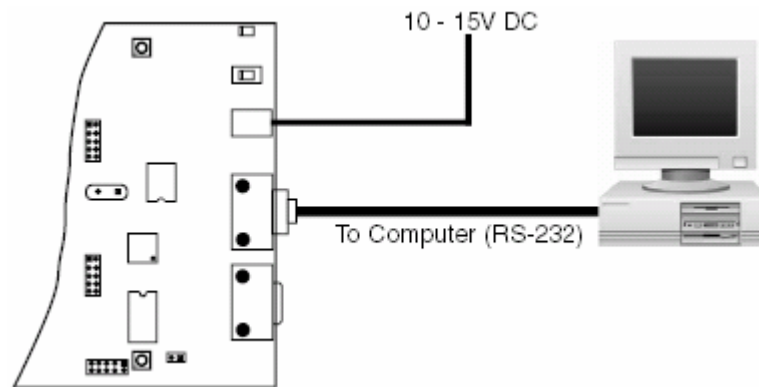
Abbildung 2. 1: Standard Setup des STK500



Das Starter-Kit kann für verschiedene Takt- und Stromversorgungsquellen konfiguriert werden. Eine genaue Beschreibung der Jumper Settings ist in Kapitel 3.8 auf Seite XY erklärt und auch auf die Rückseite des Board gedruckt.

2.3.1 Anschließen der Hardware

Abbildung 2. 2: Verbinden des STK500



Um den AT90S8515 zu programmieren muß der Anschluß „ISP6PIN“ mit „SPROG3“ mit dem beiliegenden 6-Adrigen Flachbandkabel verbunden werden wie es in Abb. 2.1 gezeigt wird. Die Kabelverbindungen zum Programmieren werden in Kapitel 3.7.1 auf Seite XY ausführlich beschrieben.

Verbinden Sie das beiliegende RS-232 (SubD-9) Kabel mit dem „RS232CTRL“ bezeichneten Anschluß auf dem Board und das andere Ende mit einem freien COM Port Ihres PCs wie es in Abb. 2.2 gezeigt wird.

Nun Installieren Sie die AVR-Studio Software auf dem PC. Anweisungen wie die Software installiert und bedient wird finden Sie in Kapitel 5 auf Seite xy. Wenn AVR-Studio installiert und gestartet ist, wird automatisch erkannt an welchem Port das STK500 angeschlossen ist.

2.3.2 Programmieren der Ziel-Controller

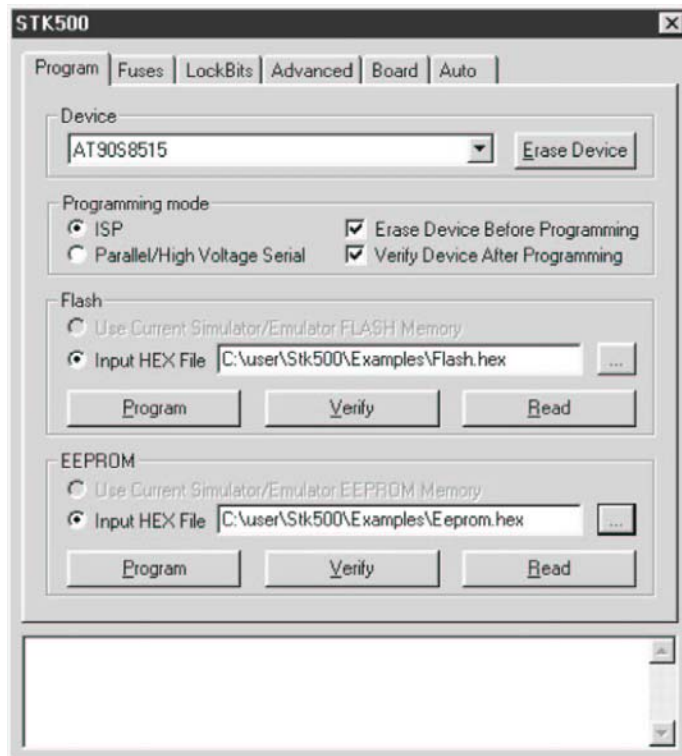
Das STK500 wird von der AVR-Studio Software gesteuert (Version 3.2 oder höher). AVR-Studio ist eine Entwicklungsumgebung (IDE) für die Entwicklung von und Fehlerbeseitigung in AVR-Anwendungen. AVR-Studio beinhaltet ein Projekt-Management Tool, einen Quell-Code Editor, Simulator, eine In-Circuit Schnittstelle und eine ISP-Schnittstelle für das STK500.

Um ein kompiliertes HEX-File in den Programmspeicher des Mikrocontrollers zu übertragen, wählen Sie „STK500“ aus dem „Tools“ Menü in AVR-Studio.

Wählen Sie den Ziel-Controller aus dem Pull-Down Menü des „Program“ Eintrags und wählen sie das HEX-File das übertragen werden soll.

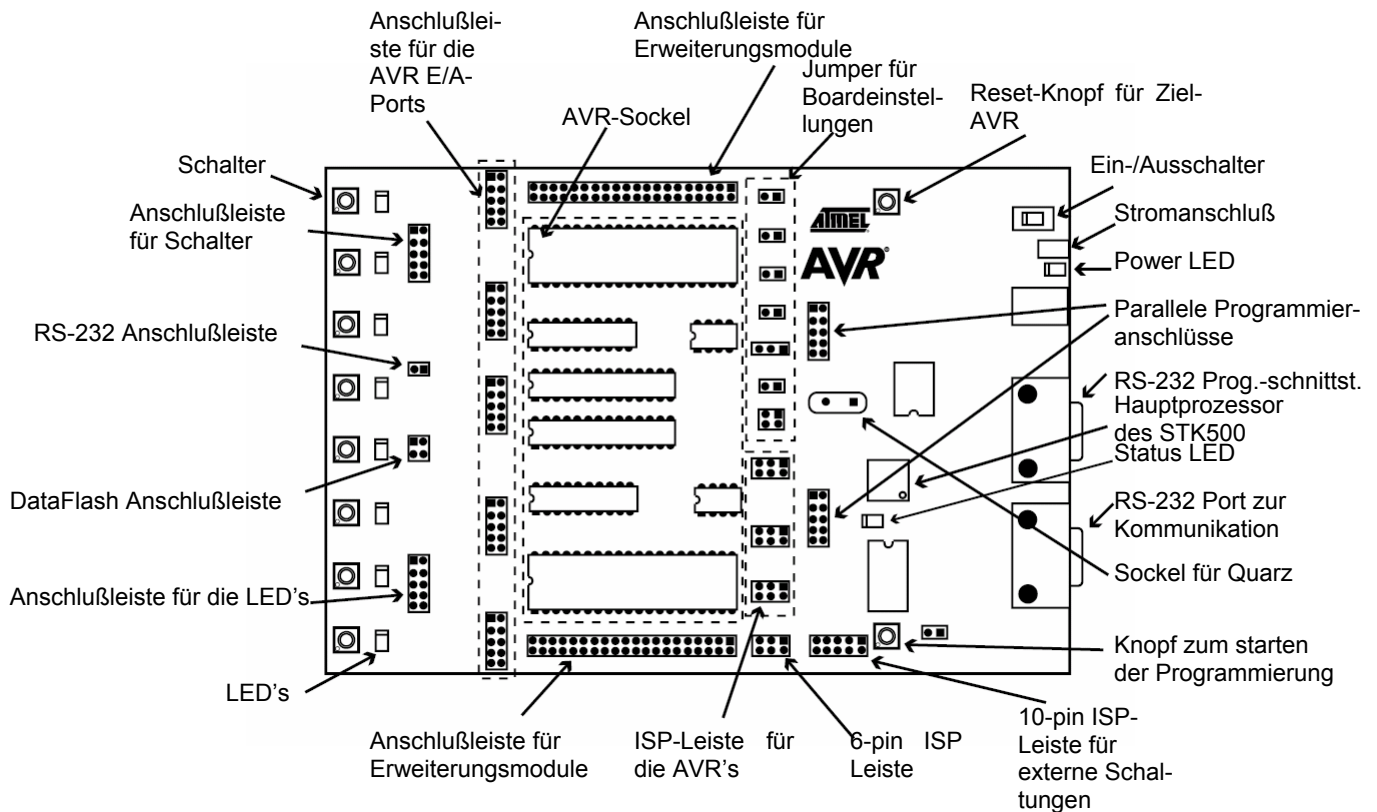
Drücken Sie erst den „Erase“-Button und anschließend den „Program“-Button. Während der Programmierung des Controllers wechselt die Status-LED des STK500 von grün auf gelb. Wenn die Programmierung erfolgreich war wechselt die LED wieder auf grün, andernfalls zeigt sie durch rotes Licht einen Fehler an. Schauen sie hierzu in das Kapitel 7 (Fehlerbehebung) auf Seite XY.

Abbildung 2. 3: AVR-Studio STK500 Programmier-Menü



Eine komplette Beschreibung zur Benutzung der Programmierschnittstelle zum STK500 des AVR-Studio finden Sie im Kapitel 5 auf Seite XY.

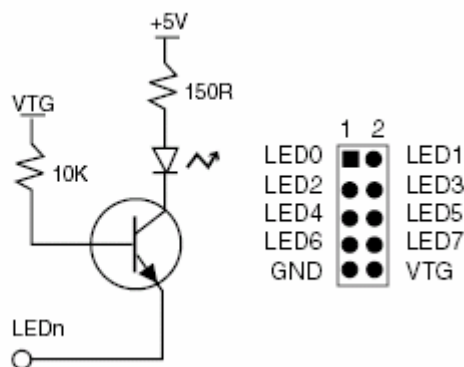
Abbildung 3.1: Komponenten des STK500



3.1 Beschreibung der benutzbaren LED's

Das STK500 enthält 8 gelbe LED's und 8 Druckschalter. Die LED's und Schalter sind an von der restlichen Schaltung separierten Leisten angeschlossen und können mit den beiliegenden 10-Adrigen Flachbandkabeln an die AVR-E/A-Ports angeschlossen werden wie es in Abbildung 3.4 gezeigt ist. Die Kabel sollten direkt an die Anschlußleisten angeschlossen werden und nicht verdreht oder verdrillt werden. Die rote Leitung am äußeren Rand der Kabel markiert Pin 1. Vergewissern Sie sich, das diese Leitung jeweils mit den Pin 1 der Anschlußleisten verbunden ist. Abbildung 3.2 zeigt wie die LED's intern verschaltet sind. Diese Lösung ermöglicht eine konstante Lichtemission der LED für alle Quellspannungen von 1,8-6V.

Abbildung 3. 2: LED's und LED-Anschlüsse



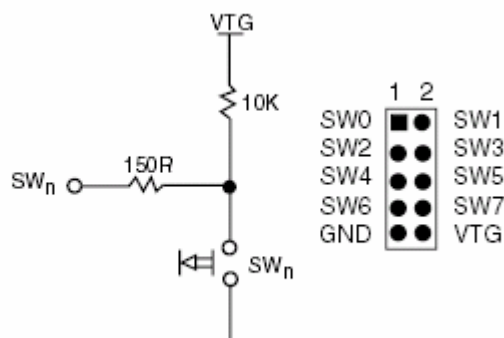
Anmerkung: Die AVR können genug Strom liefern bzw. aufnehmen, um LED's direkt anzusteuern.
Im STK500-Design werden die LED's über einen Transistor mit zwei Widerständen angesteuert, um die LED's unabhängig von der AVR Arbeitsspannung immer mit 5V zu versorgen, bzw. bei fehlender Ansteuerung durch den AVR auszuschalten.

3.2 Beschreibung der Schalter

Die Schalter sind wie in der Abbildung 3.3 gezeigt mit der Anschlußleiste gezeigt verbunden.

Durch das drücken eines Schalters wird der dazugehörige Pin SWx nach Masse gezogen, ein loslassen bzw. ein nichtgedrückter Knopf sorgt für ein Anliegen der Spannung VTG am betreffenden Pin. Die zulässige Spannung beträgt hier $1,8V < VTG < 6,0V$.

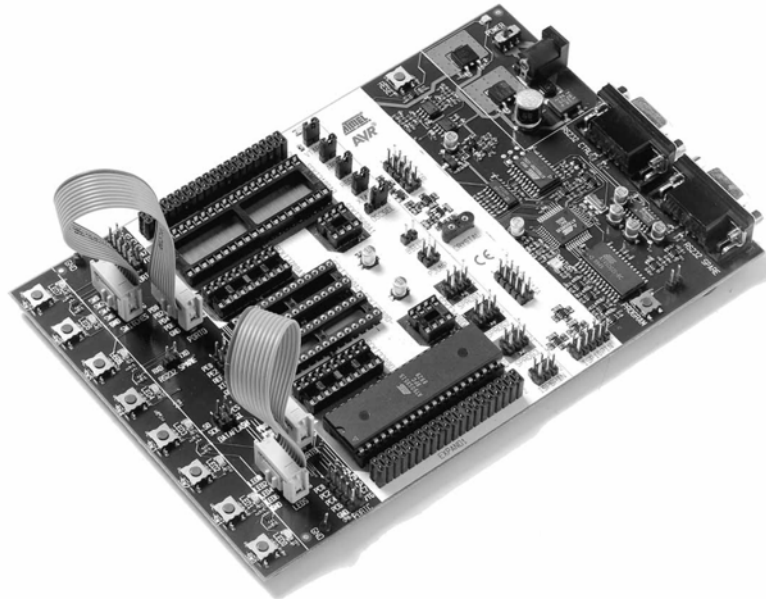
Abbildung 3. 3: Schalter und Schalter-Anschlüsse



Anmerkung: Die AVR's besitzen interne Pull-Up Widerstände, die aktiviert werden können. Damit können bei eigenen Schaltungsentwürfen externe Pull-Up entfallen.
Im STK500-Design wurden die Widerstände dennoch hinzugefügt, so daß eine logische „1“ an den betreffenden SWx-Pins anliegt, solange der betreffende Knopf nicht gedrückt ist. Der 150Ω-Widerstand limitiert den Strom, der durch den AVR-Eingang fließt.

3.3 Verbinden von LED's und Schaltern

Abbildung 3. 4: Verbinden von LED's und Schaltern

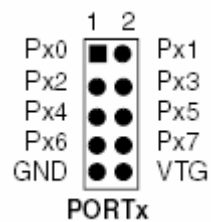


Die Verbindung mit den Schaltern und LED's kann wahlweise mit allen E/A-Ports des AVR durch die 10-Adrigen Flachbandkabel erfolgen. Die Anschlußleisten sind zusätzlich zu den Signal-Leitungen noch mit VTG und GND belegt.

3.4 Port Anschlüsse

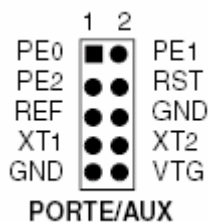
Die Pinbelegung für die E/A-Port Anschlüsse ist in Abbildung 3.5 erklärt. Die quadratische Markierung zeigt Pin 1.

Abbildung 3. 5: Pinbelegung der E/A-Anschlüsse



Der PORTE/AUX-Anschluss hat eine besondere Funktion im Zusammenhang mit den PORTE-Pins. Die Pinbelegung von diesem Anschluß wird in Abbildung 3.6 gezeigt.

Abbildung 3. 6: Pinbelegung des PORTE-Anschlusses



Die speziellen Funktionen dieses Ports sind folgender Tabelle zu entnehmen:

Abbildung 3. 7: PORTE-Anschluss

	ATmega161	AT90S4414/8515
PE0	PE0/ICP/INT2	ICP
PE1	PE1/ALE	ALE
PE2	PE2/OC1B	OC1B

- **REF:** Analoge Referenz-Spannung. Dieser Pin ist mit dem AREF-Pin der Controller mit separatem Eingang für die Analoge Referenzspannung verbunden.
- **XT1:** XTAL 1 Pin. Dies ist das interne Haupt-Taktsignal, das an allen AVR-Sockeln anliegt. Wenn der XTAL1-Jumper nicht verbunden ist, kann hier ein externes Taktsignal eingespeist werden.
- **XT2:** XTAL 2 Pin. Wenn der XTAL1-Jumper nicht verbunden ist, kann hier zusammen mit XT1 ein Quarz angeschlossen werden.

Die Anschlüsse für die LED's und Schalter haben die gleiche Pinbelegung wie die Anschlüsse der E/A-Ports und wurden in Abbildung 3.2 und 3.3 schon gezeigt. Auch hier markiert ein Quadrat Pin 1.

Das STK500 enthält 2 RS-232 Schnittstellen. Ein RS-232 Port ist für den Datenaustausch zwischen STK500 und AVR-Studio reserviert. Die 2. RS-232 Schnittstelle kann zur Kommunikation zwischen AVR und externen Schaltungen oder PC-Programmen genutzt werden. Dafür müssen die UART-Pins des AVR physikalisch mit der RS-232 Schnittstelle verbunden werden.

Der 2-Pin Anschluß mit der Bezeichnung „RS232 SPARE“ kann dazu benutzt werden, um den AVR im Sockel des STK500 mit dem RS-232-Wandlerbausteins zu verbinden. Benutzen Sie dafür eins der beiliegenden 2-Adrigen Kabel. Die Verbindung wird in Abbildung 3.8 gezeigt. Abbildung 3.9 zeigt ein Blockschaltbild der RS-232 Verbindung.

Abbildung 3. 8: Verbindung der E/A-Pins zum UART

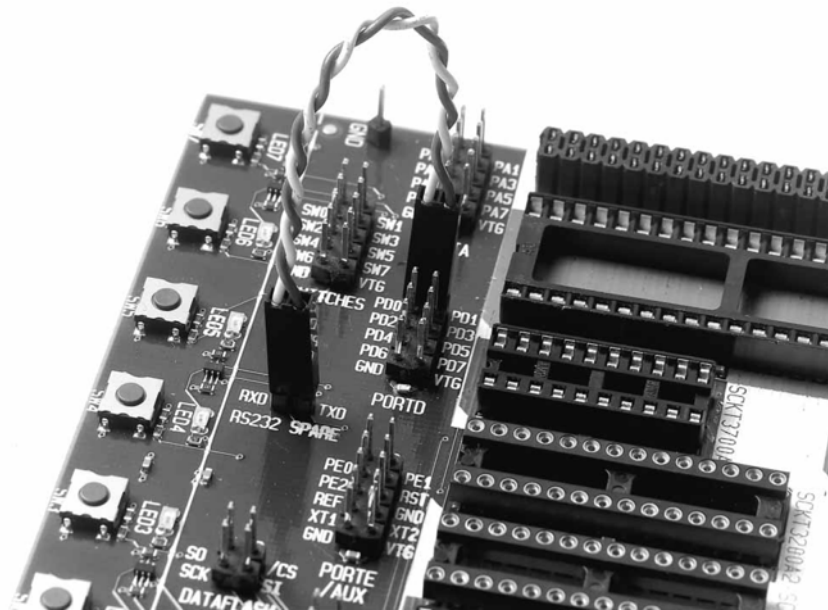
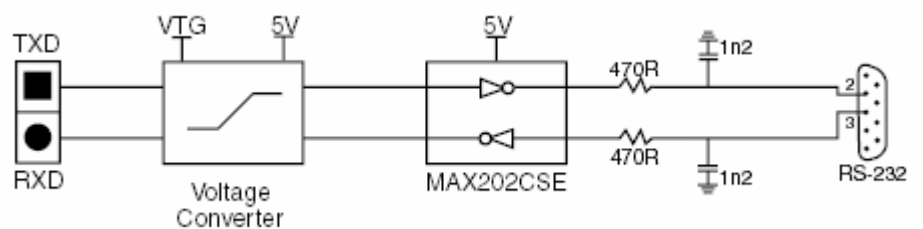


Abbildung 3. 9: Schaltplan der UART-Verbindung



Zum speichern von Daten, die auch beim Ausschalten der Versorgungsspannung erhalten bleiben ist auf dem STK500 (nur ältere Modelle!) ein AT45D021 2-Mbit DataFlash enthalten.

Ein DataFlash ist ein high-density Flash Speicher mit serieller SPI Schnittstelle. Ein detailliertes Datenblatt des Bausteins kann von der Flash Memory Sektion der Atmel CD oder von der Atmel Website bezogen werden.

Der Speicher kann mit den E/A-Pins der AVR-Sockel verbunden werden, indem man den 4-Pin Anschluß mit der Bezeichnung „DATAFLASH“ mit den gewünschten E/A-Pins des AVR-Controllers durch die mitgelieferten 2-Adrigen Kabel verbindet. Das 10-Adrige Flachbandkabel kann auch benutzt werden wenn der DataFlash Baustein mit der Hardware-SPI Schnittstelle der AVR-Controller verbunden wird. Die Verbindung der E/A-Pins wird in Abbildung 3.12 gezeigt. Abbildung 3.13 zeigt ein Blockschaltbild der DataFlash Verbindung. Die Pinbelegungen der SPI-Schnittstelle zeigen Abbildung 3.10 und 3.11.

Abbildung 3. 10: PORTB SPI Pinbelegung (40-Pin AVR's)

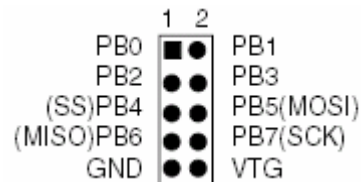


Abbildung 3. 11: PORTB SPI Pinbelegung (28-Pin AVR's)

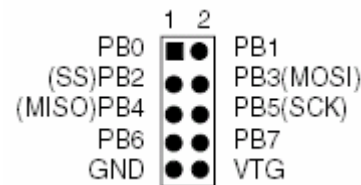


Abbildung 3. 12: Verbindung zum DataFlash für den AT90S8515

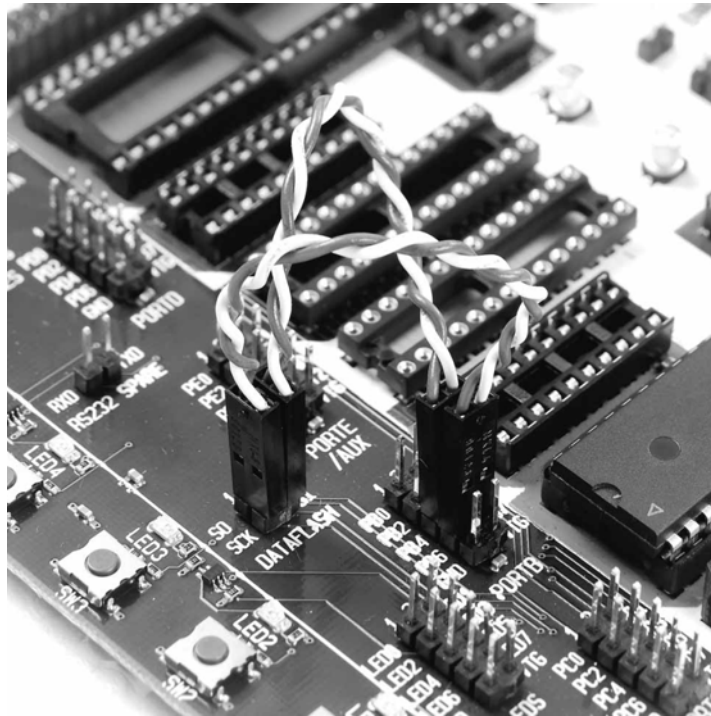
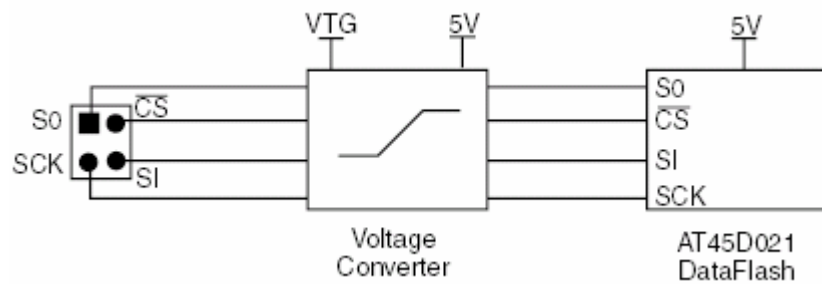


Abbildung 3. 13: Schaltbild der DataFlash Verbindung



Das Programmier-Modul besteht aus den 8 Sockeln in der weißen Fläche die sich in der Mitte des STK500 befindet.

In diese Sockel werden die AVR-Controller zum Programmieren und zum Testen vom Programmieren gesteckt.

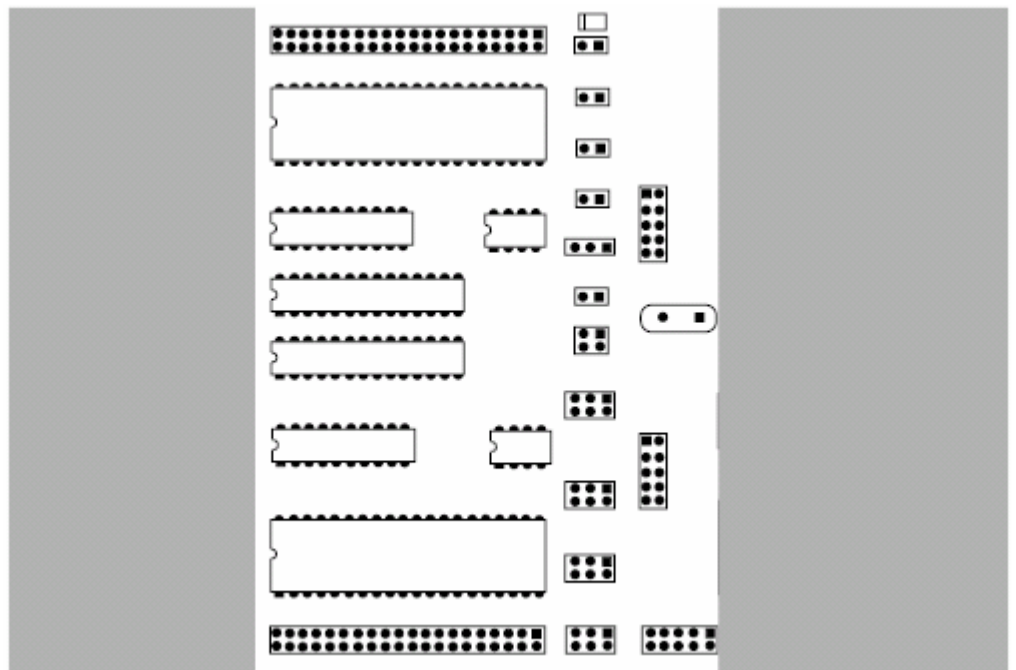
Anmerkung: Es darf nur jeweils ein Controller gleichzeitig im Programmier-Modul stecken!

Die Funktionstüchtigkeit des Flash-Speichers der AVR-Controller wird für 1000 Lösch-/Schreibzyklen garantiert. Die durchschnittliche Lebenserwartung liegt jedoch wesentlich höher.

Anmerkung: Beim einsetzen eines AVR-Controllers in den Sockel achten Sie unbedingt auf die richtige Ausrichtung.

Die Kerbe auf der Seite des Controllers muß mit der Kerbe im Sockel des STK500 übereinstimmen. Wenn der Controller falsch herum eingesetzt wird kann es zu Schäden am Controller und auch am STK500 kommen.

Abbildung 3. 14: Das STK500 Programmier-Modul



Der eingesetzte Controller kann mit AVR-Studio auf zwei verschiedene Arten im System programmiert werden:

1. AVR In-System-Programming bei normaler Betriebsspannung
 2. High-voltage Programming, hier beträgt die Spannung immer 5V.
- 4 Netze (VTARGET, RESET, XTAL1 und AREF) können mit dem Programmiermodul verbunden werden.

Die folgenden Absätze beschreiben, wie die verschiedenen Programmiermethoden verwendet werden. Eine Anleitung für den Einsatz des AVR-Studio zum Programmieren lesen Sie Kapitel 5, „Arbeiten mit AVR-Studio“ auf Seite xy.

3.7.1 ISP Programmierung

Beim In-System-Programming wird das AVR interne SPI (Serial Peripheral Interface = serielle Peripherie-Schnittstelle) benutzt um den Programm-Code in den Flash und/oder EEPROM Speicher des AVR zu laden. ISP benötigt nur V_{CC} , GND, RESET und drei Signal-Leitungen zum Programmieren. Alle AVR's (außer AT90S8534, ATtiny11 und ATtiny28) können per ISP bei normaler Betriebsspannung, normalerweise zwischen 2,1V und 6V, programmiert werden. Es werden keine "Hochspannungsleitungen" benötigt, wobei der ISP-Programmer Flash und EEPROM beschreiben kann. Der Programmer kann auch die Fuse-Bits zum Auswählen der Takteinstellungen, Startverzögerung und der internen Brown-Out-Detection (BOD) der meisten AVR's setzen.

Durch das High-voltage Programming können auch die AVR's programmiert werden, die nicht durch ISP programmierbar sind. Manche AVR's brauchen zum setzen bestimmter Fuse-Bits High-voltage Programming. Lesen Sie dazu den Absatz "High-voltage Programming" auf Seite xy für eine Anleitung wie dieses Feature anzuwenden ist.

Weil die Programmier-Schnittstelle bei den AVR's an unterschiedlichen Pins liegt gibt es drei verschiedene ISP-Anschlüsse auf dem STK500 um die Programmiersignale mit Hilfe des beigelegten 6-Adrigen Flachbandkabels zu den richtigen Pins der AVR's zu leiten. Welche der drei ISP-Schnittstellen des STK500 mit welchen Sockeln verbunden sind wird über eine Farb-Codierung und ein Nummernsystem festgelegt.

Während der ISP-Programmierung muß das 6-Adrige Flachbandkabel ständig mit dem Anschluß mit der Bezeichnung "ISP6PIN" verbunden sein. Beim Programmieren von AVR's im blauen Sockel muß das andere Ende mit dem korrespondierenden grünen Anschluß mit der Bezeichnung "SPROG1" verbunden werden. Analog dazu wird mit dem grünen ("SPROG2") und dem roten ("SPROG3") Sockel verfahren. Die Tabelle in Abbildung 3.15 zeigt, welcher Sockel zu welchem AVR paßt, und welcher SPI-Anschluß zu verwenden ist.

Das 6-Adrige Kabel sollte nicht verdreht oder verdrillt werden. Auch hier markiert eine rote Leitung Pin1, und Sie sollten sich vergewissern, daß diese Leitung auch tatsächlich mit dem entsprechenden Pin1 des SPI-Anschlusses verbunden ist.

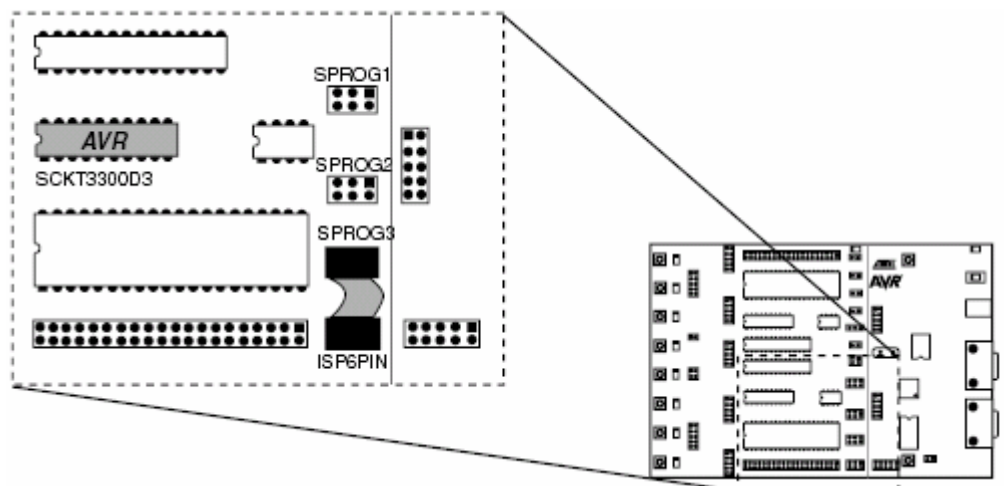
Beim Programmieren von 8-Pin AVR's sollten Sie folgendes Beachten: Pin1 wird bei manchen AVR's sowohl als RESET als auch als PB5 genutzt (ATtiny11, ATtiny12 und ATtiny15). Deshalb ist Pin1 der 8-Pin Sockel SCKT3400D1 und SCKT3400D1 (**Anm.:** hier hat sich wohl ein Fehler in der Originalbedienungsanleitung eingeschlichen. Da mir das STK500 z. Zt. nicht vorliegt kann das hier nicht berichtigen.) mit PB5 verbunden. Das während der ISP-Programmierung benötigte Reset-Signal ist also nicht an diese Sockel geführt. Dies kann jedoch geändert werden, indem für die Programmierung eine Verbindung zwischen RST vom PORTE-Anschluss zwischen PB5 vom PORTB-Anschluss hergestellt wird.

Abbildung 3. 15: AVR-Sockel

AVR	STK500-Sockel	Farbe	Nummer	ISP-Anschluß
AT90S1200 AT90S2313	SCKT3300D3	Rot	3	SPROG3
AT90S2323 AT90S2343 ATtiny12 ATtiny22	SCKT3400D1	Blau	1	SPROG1. RST und XTI von PORTE mit PB5 und PB3 von PORTB verbinden.
ATtiny11	SCKT3400D1	Blau	1	Nur High-voltage
ATtiny28	SCKT3500D-	keine	-	Nur High-voltage
AT90S4414 AT90S8515 ATmega161	SCKT3300D3	Rot	3	SPROG3
AT90S4434 AT90S8535 ATmega16 ATmega163 ATmega323	SCKT3100A3	Rot	3	SPROG3
AT90S2333 AT90S4433 ATmega8	SCKT3200A2	Grün	2	SPROG2
ATtiny15	SCKT3600A1	Blau	1	SPROG1. RST von PORTE mit PB5 von PORTB verbinden.
N/A	SCKT3700A1	Blau	1	Dieser Sockel ist derzeit nicht beschaltet.
ATmega103 ATmega128	STK501 Zusatz-Modul benötigt			

Abbildung 3.16 zeigt ein Beispiel wie ein AT90S2313 per ISP programmiert werden kann. Das 6-Adrige Flachbandkabel ist mit den Anschlüssen ISP6PIN und SPROG3 verbunden, und der AT90S2313 sitzt im Sockel SCT3100D3.

Abbildung 3. 16: ISP-Verbindung am Beispiel des AT90S2313



Es ist nicht nötig das ISP-Kabel nach der Programmierung zu entfernen. Trotzdem können die Port-Pins beim Programmdurchlauf für andere Aufgaben verwendet werden.

3.7.2 High-voltage Programmierung

Bei der High-voltage Programmierung liegt eine Spannung von 12V am RESET-Pin des AVR an. Alle AVR's könne auf diese Art und Weise programmiert werden und müssen dafür nicht aus dem Sockel gezogen werden.

Es gibt zwei verschiedene Methoden zur High-voltage Programmierung: 8-Pin AVR's besitzen eine serielle Programmierschnittstelle, die anderen Modelle werden parallel programmiert. Durch die mitgelieferten Kabel werden die zur High-voltage Programmierung nötigen Verbindungen zu den betreffenden Pins der AVR's hergestellt.

Die Tabelle aus Abbildung 3.17 faßt die Programmiermethoden und die Besonderheiten der High-voltage Programmierung zusammen.

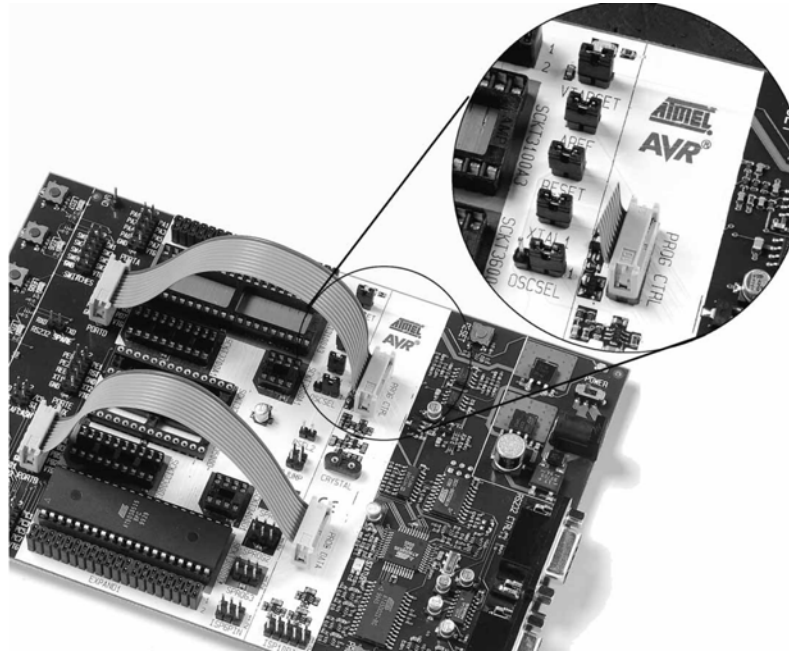
Abbildung 3. 17: Einstellung für High-voltage Programmierung

AVR	STK500-Sockel	Farbe	Nummer	ISP-Anschluß
AT90S1200 AT90S2313	SCKT3300D3	Rot	3	Parallele High-voltage Programmierung. PROG CTRL mit PORTD und PROG DATA mit PORTB wie in Abbildung 3.18 gezeigt verbinden.
AT90S4414 AT90S8515	SCKT3300D3	Rot	3	
AT90S4434 AT90S8535	SCKT3100A3	Rot	3	
ATtiny28	SCKT3500D-	Keine	-	
ATmega161	SCKT3300D3	Rot	3	Wie oben. BSEL2-Jumper ist gesetzt, siehe Abs. 3.8.
ATmega16 ATmega163 ATmega323	SCKT3100A3	Rot	3	
AT90S2333 AT90S4433	SCKT3200A2	Grün	2	
ATmega103 Atmega128	STK501 Zusatz-Modul benötigt			Wie oben. BSEL2-Jumper ist gesetzt, siehe Abs. 3.8.
ATmega8	SCKT3200A2	Grün	2	Wie oben. PJUMP Jumper sind gesetzt, BSEL2 mit PC2 verbunden, siehe Abs. 3.8.5 und 3.8.6.
AT90S2323 AT90S2343 ATtiny12 ATtiny22 ATtiny11	SCKT3400D1	Blau	1	Serielle High-voltage Programmierung
ATtiny15	SCKT3600A1	Blau	1	
N/A	SCKT3700A1	Blau	1	Dieser Sockel ist derzeit nicht beschaltet.

3.7.2.1 Parallele High-voltage Programmierung

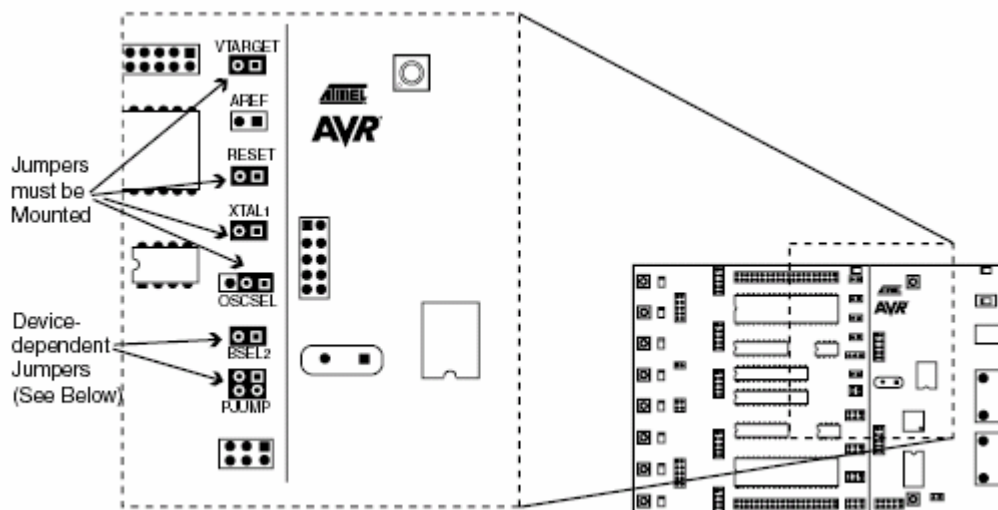
Um im Parallelen High-voltage Modus zu programmieren müssen die Signale zu den E/A-Pins des AVR geleitet werden. Benutzen Sie dazu wie in Abbildung 3.18 gezeigt die mitgelieferten 10-Adrigen Flachbandkabel und verbinden Sie die Anschlüsse PROG DATA mit PORTB und PROG CTRL mit PORTD.

Abbildung 3. 18: Verbindungen für Paralleles High-voltage Prog.



Manche Jumper-Einstellungen müssen für das Parallele High-voltage Programmieren geändert werden wie in Abbildung 3.19 gezeigt wird.

Abbildung 3. 19: Jumper Settings zur High-voltage Prog.



Das Hardware Setup:

1. Schalten Sie das STK500 aus.
2. Den AVR in den dazugehörigen Sockel (siehe Abb. 3.17).
3. Verbinden Sie die Anschlüsse PROG DATA mit PORTB.
4. Verbinden Sie die Anschlüsse PROG CTRL mit PORTD.
5. Setzen Sie den Jumper OSCSEL auf Pin 1 und Pin 2 um den Softwaregeregelten Takt einzustellen.
6. Setzen Sie den Jumper XTAL1 um das Oszillator-Signal zum AVR-Sockel zu leiten.
7. Setzen Sie die Jumper VTARGET und RESET.
8. Beim Programmieren von AT90S2333, 4433 oder ATmega8 setzen Sie beide PJUMP Jumper. Hierfür kann das beiliegende 2-Adrige Kabel anstelle von Jumpern benutzt werden.
9. Beim Programmieren der ATmega16, 163, 161, 128 oder 323 setzen Sie den BSEL2-Jumper. Beim Programmieren des Atmega8 verbinden Sie BSEL2 mit PC2. Hierfür kann das beiliegende 2-Adrige Kabel anstelle von Jumpern benutzt werden.
10. Lösen Sie die Verbindungen mit dem Zielsystem.
11. Schalten Sie das STK500 ein.
12. Stellen Sie sicher das VTARGET zwischen 4,5V und 5,5V liegt, siehe Abs. 5.3.5.1.

Für eine komplette Beschreibung der Jumper Settings lesen Sie Abschnitt 3.8, „Jumper Settings“.

Anmerkung: Entfernen Sie das Hardware Setup zur High-voltage Programmierung bevor Sie das Programm im AVR starten.

3.7.2.2 Serielle High-voltage Programmierung

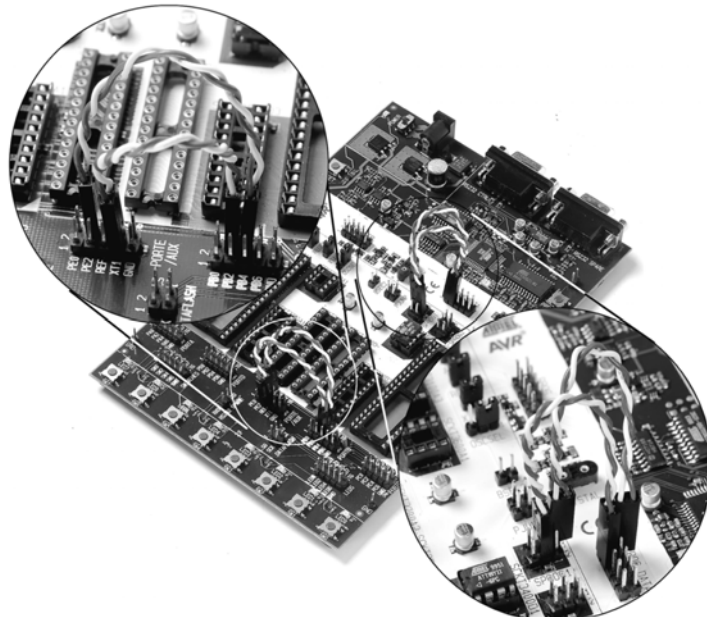
Die 8-Pin AVR's haben zu wenig Pins um Sie parallel zu Programmieren, statt dessen wird hier eine serielle Kommunikation verwendet.

Das bedeutet, daß weniger Leitungen benötigt werden. Das Hardware Setup zur seriellen High-voltage Programmierung ist wie folgt:

1. Schalten Sie das STK500 aus.
2. Den AVR in den dazugehörigen Sockel (siehe Abb. 3.17).
3. Setzen Sie den Jumper OSCSEL auf Pin 1 und Pin 2 um den Softwaregeregelten Takt einzustellen.
4. Setzen Sie den Jumper XTAL1 um das Oszillator-Signal zum AVR-Sockel zu leiten.
5. Setzen Sie die Jumper VTARGET und RESET.
6. Verbinden Sie PB3 mit von PORTB mit XT1 von PORTE. Dies stellt die Verbindung zwischen dem Takt-System und dem zu Programmierenden AVR her.
7. Verbinden Sie PB5 von PORTB mit RST von PORTE. Dies stellt die Verbindung zwischen dem Reset-System und dem zu Programmierenden AVR her.
8. Verbinden Sie PB0, PB1 und PB2 von SPROG1 mit DATA0, DATA1 und DATA2 vom PROG DATA Anschluß.
9. Schalten Sie das STK500 ein, und Sie sind bereit zum Programmieren.

Alle Verbindungen werden in Abbildung 3.20 gezeigt.

Abbildung 3. 20: Verbindung für serielles High-voltage Prog.



3.8 Jumper Settings

Ein Hauptcontroller und die Jumper steuern die Hardware Settings des STK500. Während der normalen Benutzung des STK500 sollten diese Jumper in der Standardeinstellung gesetzt sein. Um andere Einstellungen mit dem Starter-Kit zu nutzen können Jumper entfernt oder umgesetzt werden. Die Bedeutung Jumper Settings und die Anwendung werden im folgenden Abschnitt erklärt, die Standardeinstellung wird in Abbildung 3.21 und 3.22 gezeigt.

Abbildung 3. 21: Standard Jumper Settings

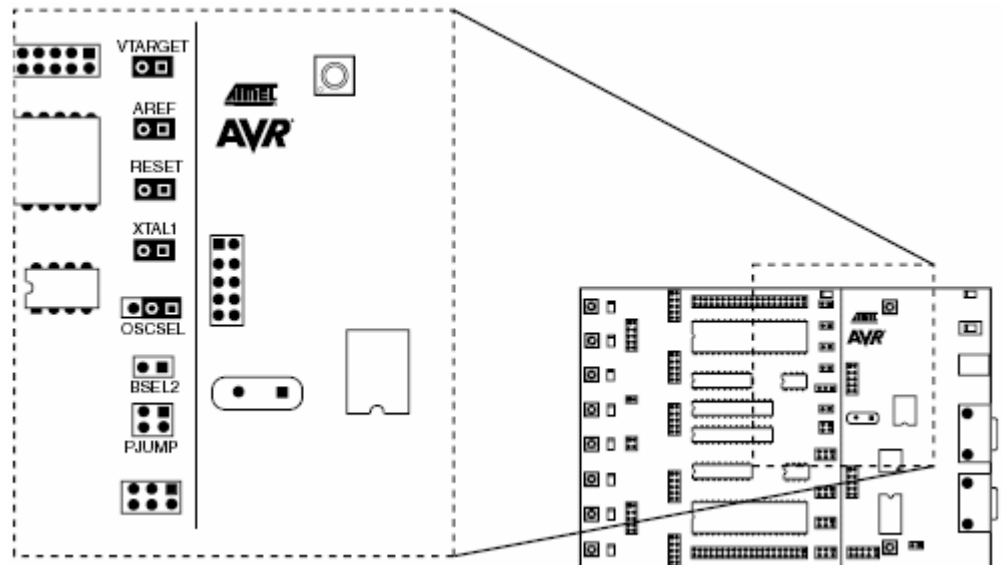


Abbildung 3. 22: Beschreibung der Jumper

Jumper	Beschreibung der Standard Einstellung
VTARGET	On-Board VTARGET Stromversorgung angeschlossen
AREF	On-Board Analog-Referenzspannung angeschlossen
RESET	On-Board Reset-System angeschlossen
XTAL1	On-Board Takt-System angeschlossen
OSCSEL	On-Board Oszillator ausgewählt
BSEL2	Nicht gesetzt. Wird für High-voltage Programmierung der ATmega8, 16, 161, 163, 128 und 323 benötigt
PJUMP	Nicht gesetzt. Wird für High-voltage Programmierung der AT90S2333, 4433 und ATmega8 benötigt

3.8.1 V_{cc} Einstellung, VTARGET

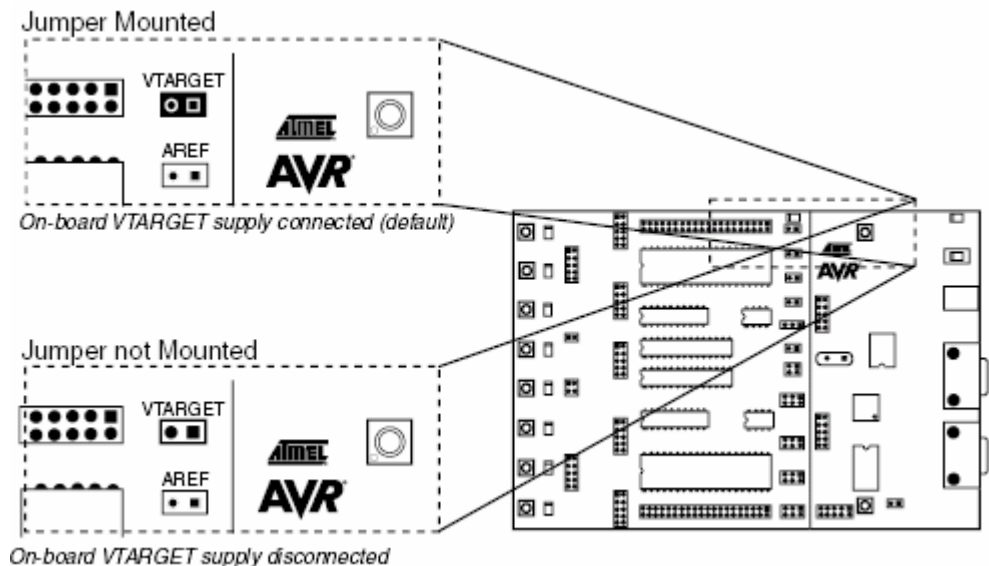
VTARGET liefert die Spannung für die AVR-Controller Sockel. VTARGET kann von AVR-Studio eingestellt werden oder aus einer externen Spannungsquelle versorgt werden. Wenn der VTARGET Jumper gesetzt ist, ist die On-Board Spannungsversorgung mit den Sockeln verbunden. Die On-Board Spannungsversorgung kann über AVR-Studio von 0V – 6V eingestellt werden. Schauen Sie immer ins Datenblatt des verwendeten Controllers bevor Sie VTARGET über AVR-Studio einstellen.

Die On-Board Spannungsversorgung liefert einen Strom von ca. 500mA, siehe hierzu Anhang A in Kapitel 10.

Wenn der VTARGET Jumper nicht gesetzt ist, muß die Arbeitsspannung V_{cc} dem AVR durch eine externe Spannungsquelle über einen der VTG-Pins der Port-Anschlüsse zur Verfügung gestellt werden.

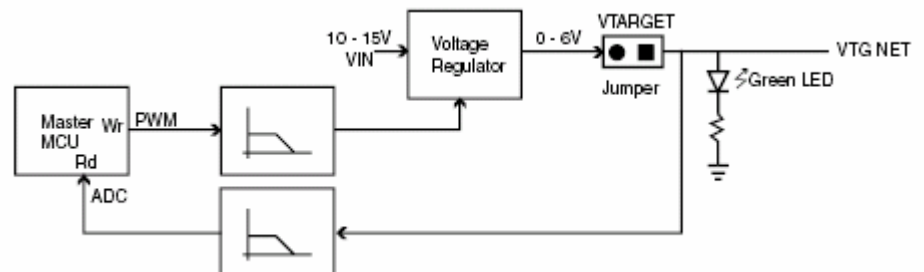
Wenn eine externe Spannungsquelle benutzt wird muß der Benutzer dafür sorgen, daß VTARGET eine höhere Spannung hat als AREF. Verbinden Sie immer auch GND mit der Bord-Masse wenn Sie eine externe Spannungsquelle benutzen.

Abbildung 3. 23: VTARGET Jumper Einstellungsmöglichkeiten



Der STK500-Controller steuert die Spannung an VTARGET durch PWM. Abbildung 3.24 zeigt die interne Beschaltung des VTARGET Signals.

Abbildung 3. 24: VTARGET Beschaltung



Anmerkung: Die grüne LED leuchtet wenn an dem VTG-Netz Spannung anliegt. Ohne VTG kann das Programmier-Modul des STK500 nicht genutzt werden.

Die interne VTARGET Versorgung ist Kurzschluß-Sicher. Wenn durch AVR-Studio VTARGET >0,3V gesetzt ist und der Boardcontroller für mehr als 80ms eine Spannung < 0,3V mißt, werden VTARGET und AREF vom Netz getrennt und die Status-LED fängt hektisch an zu blinken.

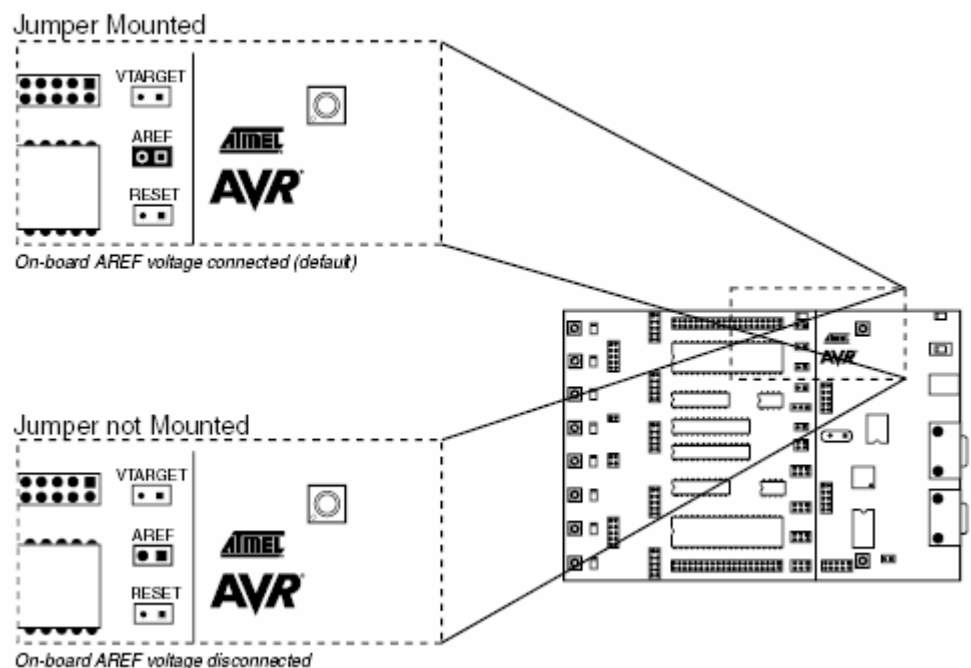
Wenn eine externe Spannungsquelle benutzt wird und abgeschaltet wird interpretiert der Boardcontroller dies fälschlicherweise auch als Kurzschluß. Dies kann dadurch verhindert werden, indem bei Nutzung einer externen Spannungsquelle die interne durch AVR-Studio auf < 0,3V gestellt wird.

3.8.2 Analoge Referenzspannung, AREF

Die analoge Spannung AREF liefert die Referenzspannung für den on-chip A/D-Konverter des AVR. Wenn der AREF-Jumper gesetzt ist, ist die On-Board analog Referenzspannung mit dem AREF-Pin des AVR verbunden. Die On-Board Referenzspannung kann über AVR Studio von 0V bis 6V eingestellt werden, jedoch nicht über VTARGET.

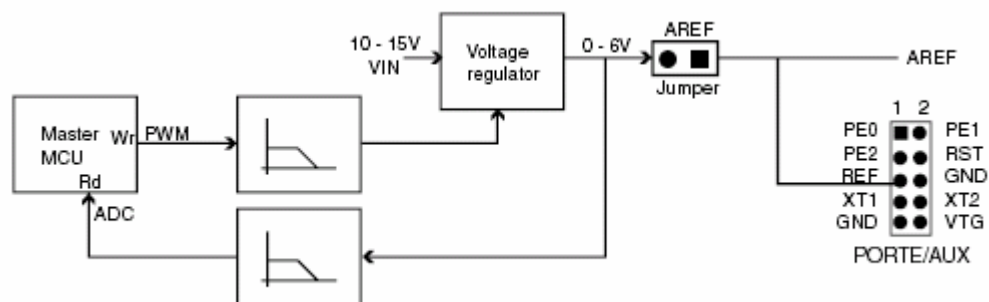
Wenn der AREF-Jumper nicht gesetzt ist, muß dem AVR die Referenzspannung von einer externen Spannungsquelle über die PORTE/AUX-Leiste zugeführt werden (siehe Abbildung 3.6). Abbildung 3.25 zeigt die AREF-Jumper Einstellungsmöglichkeiten.

Abbildung 3. 25: AREF-Jumper Einstellungen



Der STK500-Controller steuert die Referenzspannung durch PWM. Die am AVR anliegende Referenzspannung kann auch am PORTE-Anschluss abgegriffen werden, wobei dieser Anschluß auch als externe AREF-Versorgung benutzt werden kann.

Abbildung 3. 26: Interne AREF-Verbindung



Die durch AVR Studio eingestellte AREF-Spannung kann außerdem als Eingang für den Analog-Comparator oder für die ADC-Eingänge des AVR genutzt werden. Das AVR AREF-Signal kann dann an VTG angeschlossen werden.

Die Bordinterne AREF-Spannung ist kurzschlußsicher. Wenn die AREF-Spannung $>0,3V$ eingestellt ist und für mehr als 80 ms unter $0,3V$ sinkt, schaltet der Bord-Controller AREF ab. In diesem Fall blinkt die Status-LED langsam.

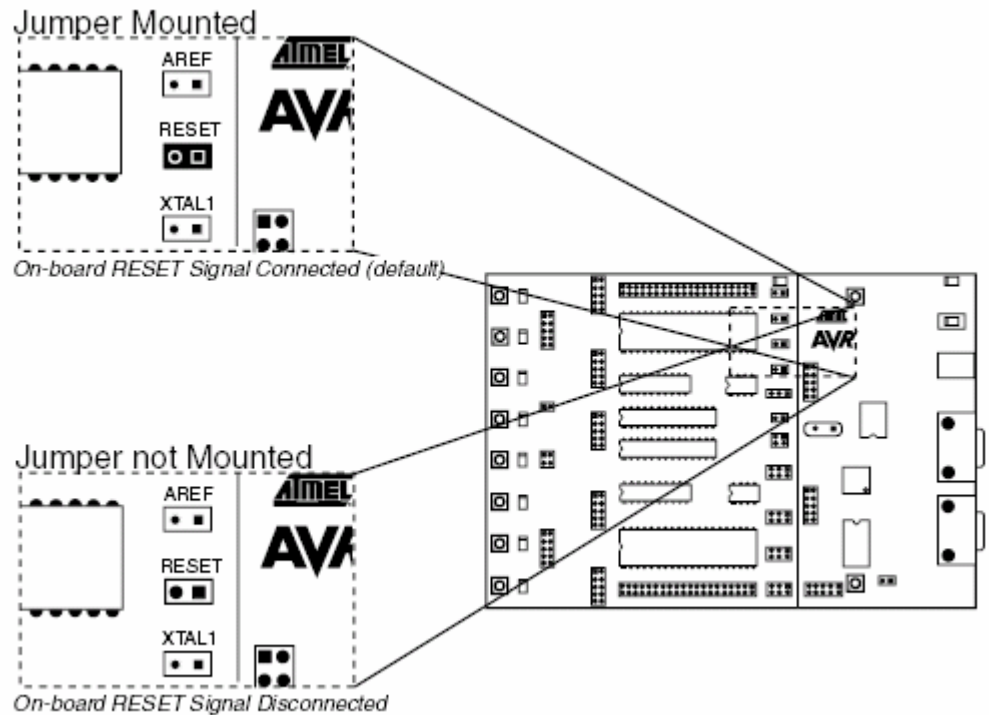
AREF wird auch abgeschaltet (genauso wie auch VTARGET), wenn ein Kurzschluß in VTARGET auftritt, in diesem Fall blinkt die Status-LED schnell.

3.8.3 Reset Einstellungen, RESET

Der RESET-Jumper kontrolliert das Reset-Signal des STK500. Beim ISP Programming kann der Bord-Controller den AVR im Sockel programmieren, ohne sich an der restlichen Schaltung zu stören. Wenn der RESET-Jumper gesetzt ist, wird das Reset-Signal zum AVR durch den Bord-Controller gesteuert. Beim nicht gesetzten Jumper ist der Reset-Pin vom AVR nicht verbunden. Dies ist für externe Reset-Systeme bei Prototypen gedacht.

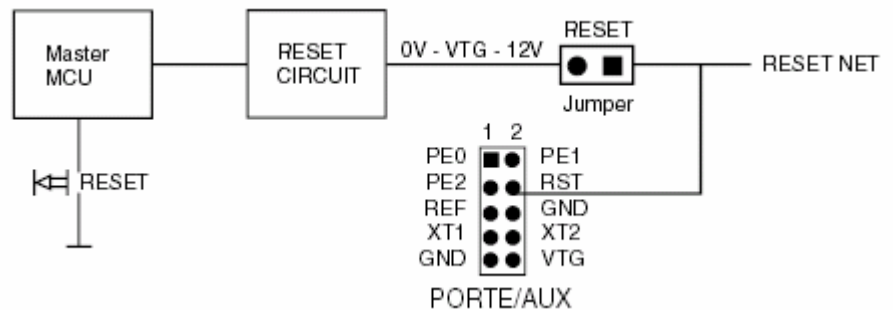
Der RESET-Jumper muß jedoch immer gesetzt sein, wenn der AVR High-voltage programmiert wird. Auch beim Einsatz eines externen Reset-System muß der Bord-Controller beim Programmieren die Kontrolle über die Reset-Leitung haben. Der Reset-Schalter des STK500 ist nicht mit dem AVR verbunden, wenn der Jumper nicht gesetzt ist. Abbildung 3.27 zeigt die RESET-Jumper Einstellungen.

Abbildung 3. 27: RESET-Jumper Einstellung



Der STK500 Bord-Controller steuert das Reset-Signal des AVR. Das Reset-Signal ist auch auf den PORTE-Anschluss geführt, wo ein externes Reset-System angeschlossen werden kann. Abbildung 3.28 zeigt die interne Reset-Verschaltung des Reset-Signals.

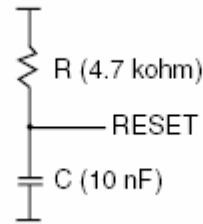
Abbildung 3. 28: Interne Reset-Schaltung



Anmerkung: Beim High-voltage Programming liegen durch die Reset-Leitung 12V am AVR an. Wenn ein externes Reset-System angeschlossen ist und nicht mit dieser Spannung umgehen kann, sollte dieses System vorher vom STK500 getrennt werden.

Wenn ein externes Reset-System angeschlossen ist, wird häufig ein externer Pull-Up Widerstand an die Reset-Leitung angeschlossen. Abbildung 3.29 zeigt einen typischen Reset-Aufbau

Abbildung 3. 29: Externe Reset-Verbindung



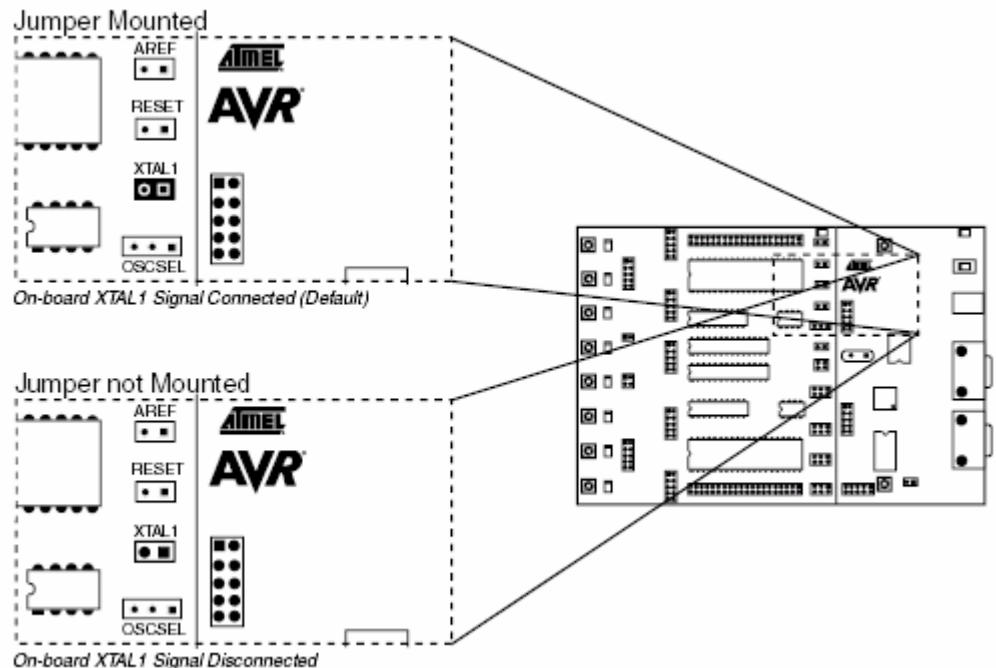
Wenn der Pull-Up Widerstand zu klein gewählt wird (<4,7kΩ) kann das STK500 die Reset-Leitung nicht auf Masse ziehen.

3.8.4 Takteinstellung, XTAL1 und OSCSEL

Das STK500 stellt verschiedene Taktoptionen für den AVR zur Verfügung. Das Setzen der Jumper XTAL1 und OSCSEL kontrolliert den Takt. OSCSEL entscheidet darüber, welches Signal zum XTAL1-Pin des AVR geleitet wird.

Wenn der XTAL1-Jumper gesetzt ist, wird das STK500-interne Taktsystem als Taktquelle des AVR genommen. Wenn XTAL1 nicht gesetzt ist, kann ein externes Taktsignal oder ein Quarz benutzt werden. Abbildung 3.30 zeigt den XTAL1-Jumper

Abbildung 3. 30: XTAL1-Jumper Einstellung



Wenn XTAL1 nicht gesetzt ist kann die externe Taktquelle an den PORTE-Anschluss angeschlossen werden, wie in Abbildung 3.32 gezeigt.

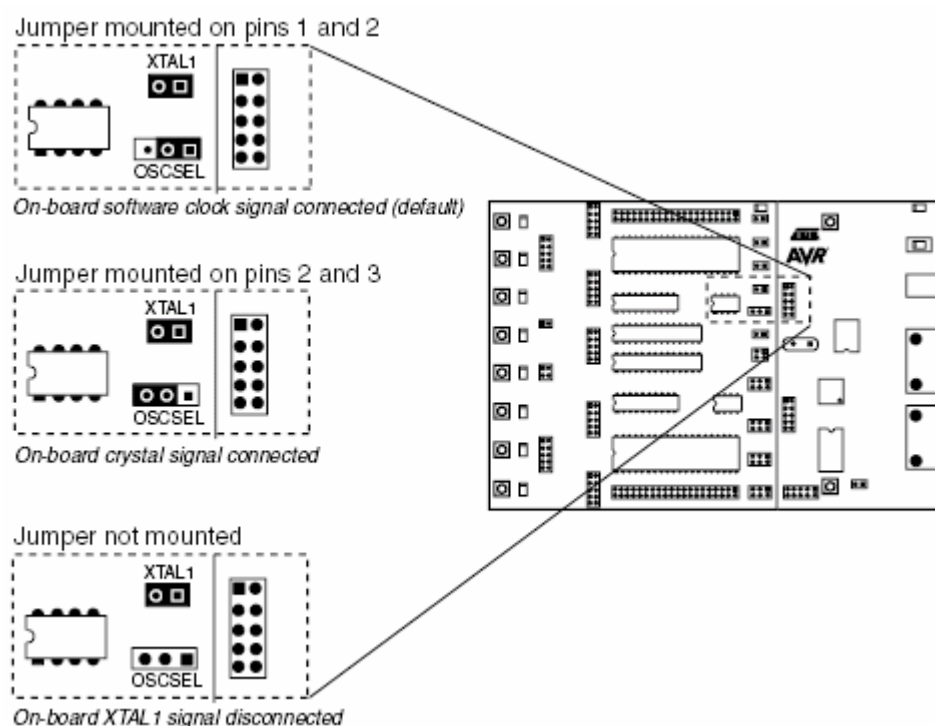
Wenn der XTAL1-Jumper gesetzt ist, liefert die STK500-interne Taktquelle den Takt für den AVR. Die interne Taktquelle kann entweder durch einen Quarz oder durch einen Software-Taktgenerator vom Bord-Controller betrieben werden. Der Software-Taktgenerator kann von 0 bis 3,68 MHz eingestellt werden. 3,68 MHz ist die Standardeinstellung. Kapitel xy auf Seite xy erklärt wie die Taktfrequenz mit AVR Studio eingestellt wird.

Wenn das vom Bord-Controller generierte Taktsignal genutzt wird, müssen die Fuse-Bits des AVR auf „external clock“ als Taktquelle konfiguriert werden. Dies ermöglicht auch eine kurze Startzeit des AVR. Die Fuse-Bits für die Takteinstellungen werden in Kapitel xy auf Seite xy näher erklärt. Nicht alle AVR's können über Fuse-Bits einstellen ob ein Quarz oder ein Oszillator als Taktquelle genutzt wird.

Die interne Taktquelle wird mit dem OSCSEL-Jumper ausgewählt. Abbildung 3.31 zeigt die Einstellmöglichkeiten für OSCSEL.

Der On-Board Oszillator arbeitet mit keramischen Resonatoren oder mit Quarzen zwischen 2 und 20 MHz (**AT-cut, fundamental and parallel resonant crystals**).

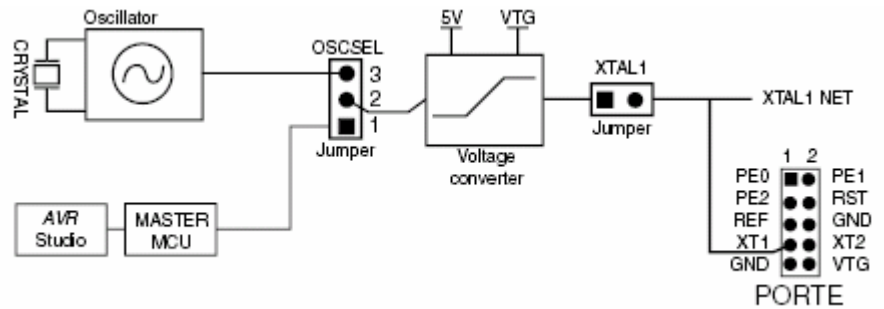
Abbildung 3. 31: OSCSEL-Jumper Einstellung



Wenn der AVR im High-voltage Programming mode programmiert wird, sollte der Jumper OSCSEL auf Pin 1 und Pin 2 montiert sein, um dem Bord-Controller die Kontrolle über das Taktsignal zu überlassen, wie in Kapitel 3.7.2 auf den Seiten 21 und 22 beschrieben.

Anmerkung: In einer realen (eigenständigen) Schaltung, in der nur ein AVR an den Quarz angeschlossen ist, besteht keine Notwendigkeit für einen externen Oszillatoraufbau. Das STK500 hat jedoch 8 verschiedene AVR-Sockel an am selben Taktsystem angeschlossen, wodurch lange Signalleitungen zu den AVR's entstehen. Dies macht es schwierig, einen Quarz am AVR-internen on-chip Oszillator zu betreiben. Der Oszillator auf dem STK500 ist für einen Spannungsbereich von 1,8 – 6,0V ausgelegt.

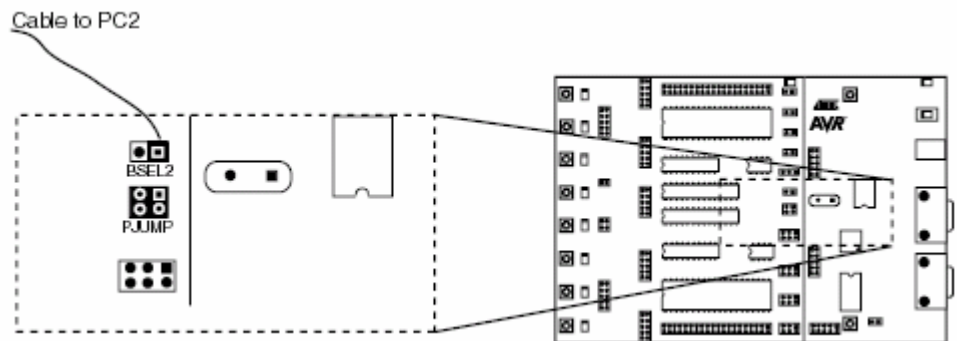
Abbildung 3. 32: XTAL1 und OSCSEL Verbindung



3.8.5 BSEL2-Jumper

Der BSEL2-Jumper ist für die Verbindung des Byte Select 2 Signals beim High-voltage Programming der ATmega8, -16, -161, -163, -128 und -323 zuständig. Der BSEL2-Jumper wird nur beim Programmieren der ATmega16, -161, -163, -128 und -323 gesetzt. Beim programmieren des ATmega8 muß der rechte BSEL2-Pin mit dem PC2-Anschluss auf den Port-Leisten verbunden werden (siehe Abbildung 3.33). Für eine Beschreibung des Byte Select 2 Signals schlagen Sie bitte in den entsprechenden AVR-Datenblättern im Kapitel Programmierung nach.

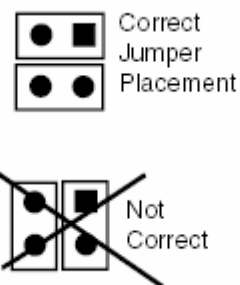
Abbildung 3. 33: BSEL2 Verbindung des ATmega8



3.8.6 PJUMP-Jumper

Die PJUMP-Jumper verbinden die Programmier-Pins der AT90S2333, -4433 und ATmega8 mit den Programmiersignalen beim High-voltage Programming. Die PJUMP-Jumper dürfen nur beim High-voltage Programming der oben genannten AVR's genutzt werden. Beim debuggen, beim High-voltage Programming oder ISP von anderen AVR's dürfen diese Jumper nicht gesetzt sein.

Abbildung 3. 34: PJUMP-Jumper Platzierung



3.9 Erweiterungs-Anschlüsse

Das STK500 hat zwei Erweiterungsanschlüsse (auf jeder Seite des Programmiermoduls einen). Alle AVR E/A-Ports, Programmiersignale und Kontrollsignale des STK500 sind auch auf die Erweiterungsanschlüsse geführt. Dies vereinfacht den Anschluß von Prototypen an das STK500. Die Pinbelegung der beiden Anschlüsse ist in Abbildung 3.36 und 3.37 dargestellt.

Abbildung 3.35: Erweiterungs-Stiftleisten

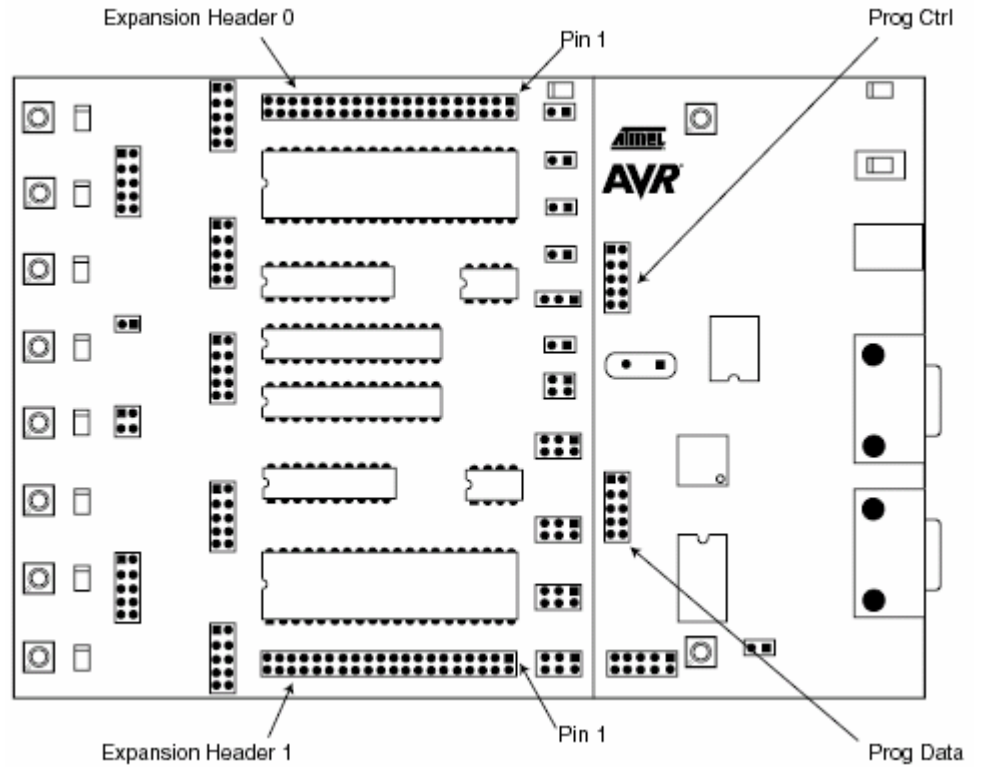


Abbildung 3. 36: Erweiterungsanschluß 0 Pinbelegung

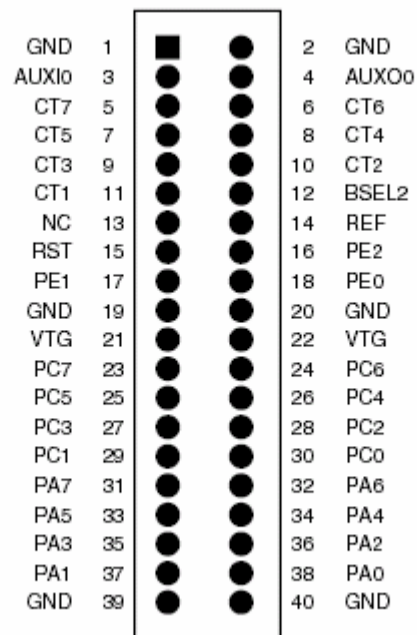
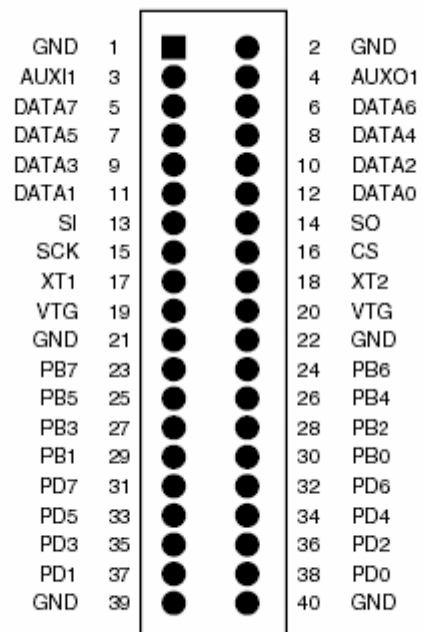


Abbildung 3. 37: Erweiterungsanschluß 1 Pinbelegung



3.9.1 Signalbeschreibung

Die Signale AUXI1, AUXI0, AUXO1 und AUXO0 sind für zukünftige Verwendungsmöglichkeiten gedacht. Verbinden Sie diese Signale nicht mit Ihrer Schaltung.

Die Signale DATA[0:7] und CT[0:7] können auch an den Anschlüssen Prog Data und Prog Ctrl abgegriffen werden. Diese Signale werden in Kapitel 3.10 erklärt.

Das BSEL2 Signal ist vom BSEL2-Jumper herangeführt. Dieser Jumper ist in Kapitel 3.8.5 auf Seite 30 erklärt.

Die Signale SI, SO, SCK u. \overline{CS} sind mit dem DataFlash verbunden. Die Benutzung des DataFlash ist in Kapitel 3.6 auf Seite 14 beschrieben.

NC bedeutet, daß der entsprechende Pin nicht verbunden ist (Not Connected).

Die verbleibenden Signale sind mit denen in Kapitel 3.4 beschriebenen Port-Anschlüssen gleichzusetzen.

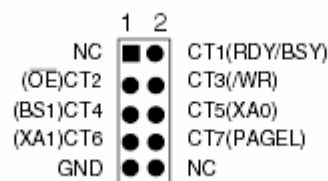
Anmerkung: Die DATA-, CT- und AUX-Signale basieren auf 5V-CMOS-Logik. Es findet keine Pegelkonvertierung zur Angleichung an VTG statt.

3.10 Prog Ctrl und Prog Data Anschlüsse

Die Prog Data und Prog Ctrl Anschlüsse werden für das High-voltage Programming der AVR benutzt. Die Plazierung der Leisten werden in Abbildung 3.35 gezeigt. Während des Programmierens werden die Ctrl-Signale zu PortD des AVR geleitet, die Data-Signale werden auf PortB gelegt (siehe Kapitel 3.7.2 auf Seite 19ff). Die Pinbelegung der Data- und Ctrl-Anschlüsse wird in Abbildung 3.38 und 3.39 gezeigt. Weitere Informationen zum High-voltage Programming können den betreffenden Datenblättern der AVR's entnommen werden.

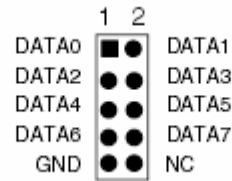
Anmerkung: Die Prog Data und Prog Ctrl Anschlüsse sind direkt (ohne Pegelanpassung) mit dem STK500-Bordcontroller verbunden und führen deshalb immer 5V-Logik Pegel.

Abbildung 3. 38: Prog Ctrl Anschluß Pinbelegung



Die Prog Ctrl Signale werden gewöhnlich für die Steuersignale beim High-voltage Programming genutzt.

Abbildung 3. 39: Prog Data Anschluß Pinbelegung

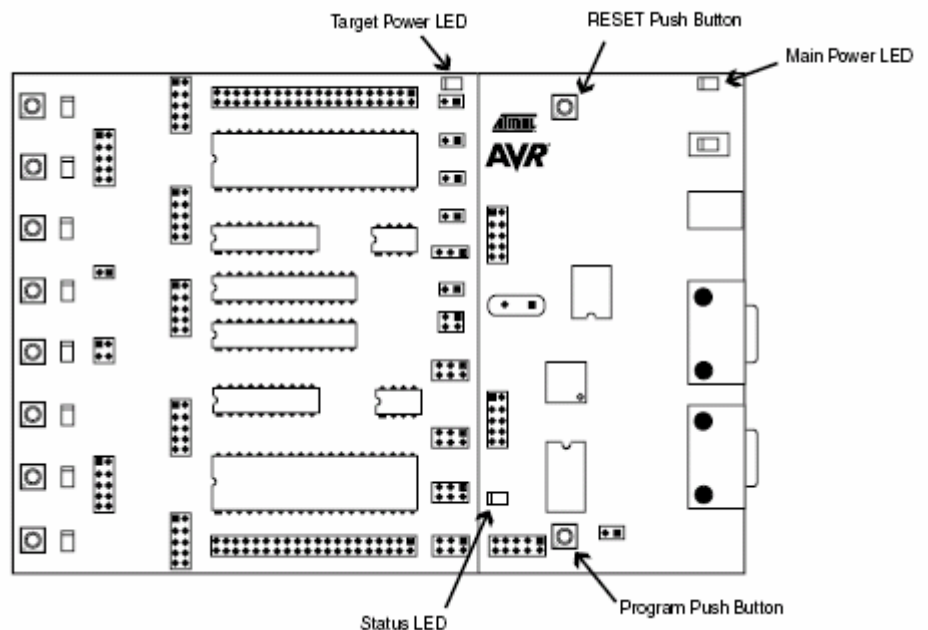


Die Prog Data Signale werden gewöhnlich für den Datenbus beim High-voltage Programming genutzt. Beim ISP wird DATA5 als MOSI, DATA6 als MISO und DATA7 als SCK genutzt.

3.11 Sonstiges

Das STK500 stellt 2 Knöpfe und 3 LED's für spezielle Funktionen und Status-Anzeigen zur Verfügung. Diese Kapitel erklärt diese Eigenschaften. Abbildung 3.40 zeigt die Plazierung dieser Funktionen.

Abbildung 3. 40: Spezielle Funktionen und Status-LED



3.11.1 Reset-Button

Der Reset-Button resettet den AVR. Der Bord-Controller wird durch den Reset-Button nicht tangiert. Wenn der RESET-Jumper nicht gesetzt ist, ist der Reset-Button nicht benutzbar.

3.11.2 PROGRAM-Button

Nachfolgende Versionen des AVR Studio können das Programm des Bord-Controllers aktualisieren. AVR Studio wird dann alte Software-Versionen erkennen und den Flash Programmspeicher des Bordcontrollers updaten. Dafür muß beim Einschalten des STK500 der PROGRAM-Button gedrückt werden. AVR Studio wird dann während des Updates weitere Anweisungen erteilen.

3.11.3 Main Power LED

Die rote Power LED ist direkt mit dem Stromanschluß des STK500 verbunden. Wenn das STK500 mit Strom versorgt wird, leuchtet die LED grundsätzlich.

3.11.4 Target Power LED

Die Target Power LED ist mit dem VCC-Netz (VTG) der AVR-Sockel verbunden. Sie leuchtet, wenn an den Sockeln Spannung anliegt.

3.11.5 Status LED

Die Status LED ist eine 3-Farbige LED. Während des Programmiervorgangs leuchtet sie gelb. Nach dem erfolgreichen Programmieren leuchtet sie grün, im Fehlerfall leuchtet sie rot um anzuzeigen, daß während der Programmierung ein Fehler aufgetreten ist.

Beim Einschalten des STK500 wechselt die Status LED von rot über gelb nach grün um damit anzuzeigen, daß der Bordcontroller bereit ist.