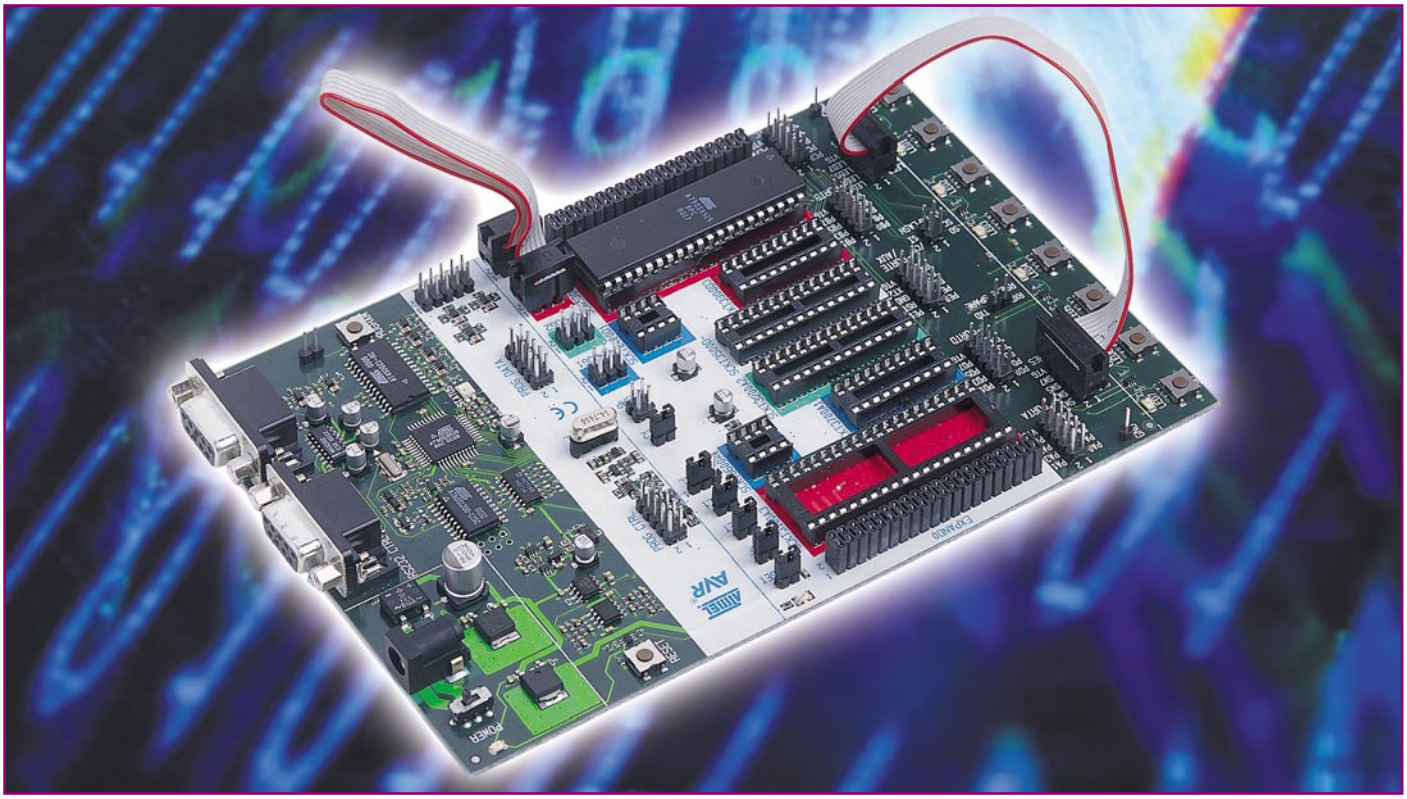

AVR-Grundlagen Teil 4

Nach dem Erstellen des ersten Demoprogramms wollen wir dieses auch ausprobieren und stellen deshalb hier das AVR-Starterkit STK 500 vor. Das erstellte Assemblerprogramm wird mit Hilfe des STK 500 in einen AVR-Mikrocontroller programmiert und getestet.

Universal AVR-Starterkit

Das universelle AVR-Starterkit STK 500 ermöglicht mit dem im letzten Heft vorgestellten AVR-Studio 3.2 (oder neuer) die Programmerstellung, die Simulation von Programmen und die Programmierung aller in Tabelle 1 angegebenen AVR-Mikrocontroller. Hiermit lassen sich also einfache Applikationen schnell erstellen und direkt auf dem Board testen, da auch schon weitere Peripherie wie z. B. ein externer Datenspeicher, eine RS-232-Schnittstelle, Leuchtdioden sowie Taster auf dem Starterkit vorhanden sind und individuell über Steckverbinder an die Ports der entsprechenden Mikrocontroller angeschaltet werden können. Im AVR-Starterkit befindet sich im Auslieferungszustand bereits ein AVR-Mikrocontroller, sodass die Arbeit mit dem Starterkit sofort beginnen kann.

Zunächst wollen wir uns den Aufbau des in Abbildung 1 gezeigten STK 500 kurz ansehen. Die Betriebsspannung von 10 bis 15 V (Gleichspannung, über DC-Hohlstecker, keine Polung vorgeschrieben), die von einem externen Netzteil (min. 500 mA) bereitzustellen und über die DC-Buchse an der linken, oberen Ecke der Leiterplatte einzuspeisen ist. Diese Spannung gelangt über einen Brückengleichrichter auf die Spannungsstabilisierung. Durch diese Gleichrichtung besteht ein sicherer Verpolungsschutz, sodass auch eine negative Spannung die Elektronik des Starterkits nicht beschädigen kann. Über einen mit „POWER" gekennzeichneten Schiebeschalter erfolgt das Einschalten des Kits.

Direkt neben der Spannungsversorgung befinden sich zwei RS-232-Schnittstellen, wovon eine zur Steuerung des STK 500 sowie zur Programmierung des Mikrocontrollers zum Einsatz kommt. Die zweite Schnittstelle kann direkt von den im Kit eingesetzten Mikrocontrollern angesprochen werden, sodass man auch Applikationen, die z. B. mit einem PC kommunizieren, mit dem Starterkit auch ohne weiteren Hardwareeinsatz testen kann.

Des Weiteren befindet sich in diesem Bereich der Platine auch ein 2-Mbit-Flash-Datenspeicher, der über ein serielles Protokoll ansteuerbar ist.

Die Taktversorgung der Mikrocontroller erfolgt entweder über einen zwischen 0 und 3,68 MHz programmierbaren Oszillator oder direkt über einen Quarz (2 bis 20 MHz), für den ein Stecksockel vorhanden ist.

Im weißen Bereich in der Mitte des Starterkits befinden sich acht IC-Sockel zur Aufnahme der verschiedenen AVR-Mikrocontroller, in denen diese auch programmiert werden können. Alle Ports der Mikrocontroller sind auf Stiftleisten geführt,

Tabelle 1:
Vom STK 500 unterstützte Mikrocontroller

ATtiny11	AT90S1200	AT90S4433	ATmega163
ATtiny12	AT90S2313	AT90S8515	
ATtiny15	AT90S2323	AT90S8535	
ATtiny28	AT90S2343	ATmega161	

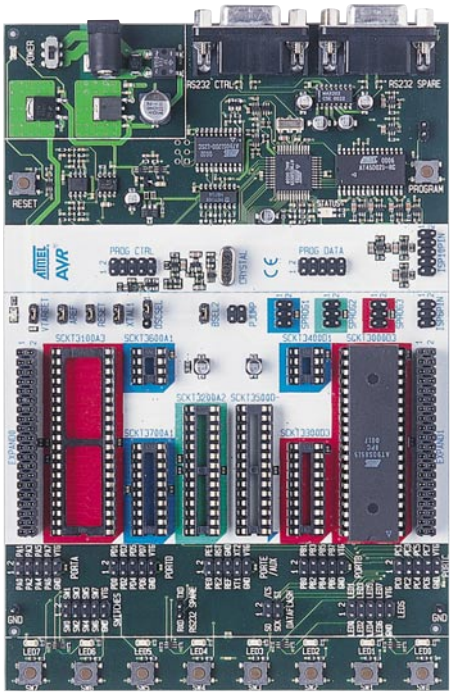

Bild 1: AVR-Starterkit STK 500

die mit „PORT A" bis „PORT E" gekennzeichnet sind, womit man einfache Programme mit Hilfe der hier vorhandenen Ein- und Ausgabeelemente (Taster, LEDs, Speicher, Schnittstelle) oder extern anschließbarer Peripherie auch direkt auf dem Board testen kann. Beim Einsetzen der Mikrocontroller ist unbedingt auf die richtige Polung zu achten. Die Gehäusekerbe am IC muss mit der Kerbe am IC-Sockel übereinstimmen. Des Weiteren darf sich immer nur insgesamt ein Controller auf dem Board befinden.

Auf der unteren Seite des Starterkits sieht man acht Taster sowie acht Leuchtdioden, die, ebenso wie die Ports, über Stiftleisten einzeln zugänglich sind. So besteht die Möglichkeit einer individuellen Zuordnung zu den Ports des Mikrocontrollers. Weiterhin befinden sich hier auch die Anschlussstifte für die RS-232-Schnittstelle und den Flash-Datenspeicher.

Über die Stiftleisten der Ports können, wie bereits angedeutet, nicht nur die auf dem Board vorhandenen Taster und LEDs kontaktiert werden, sondern, über die entsprechenden Verbindungskabel, auch eigene Schaltungen, ohne dass der AVR-Mikrocontroller im Zuge der Programmentwicklung immer wieder zum Programmieren ein- und ausgebaut werden muss. Trotzdem empfielt sich bei häufigem Gebrauch der Einsatz von sogenannten „Nullkraft-" oder „Textool-Sockeln", die direkt auf die integrierten Sockel aufgesteckt werden können und somit die Gefahr des Verbiegens der empfindlichen Prozessorpins sowie des Verschleißes der IC-Fassungen (Ermüden der Kontakte) verringern.

Inbetriebnahme und Funktion

Wir wollen die Inbetriebnahme gleich anhand unseres im letzten Heft diskutierten Programmprojektes betrachten.

Das STK 500 Starterkit wird zunächst an die Betriebsspannung und mittels des mitgelieferten seriellen Kabels an den PC, auf dem das AVR-Studio installiert ist, wie auf Abbildung 2 gezeigt, angeschlossen - aber noch nicht eingeschaltet. Damit das Programm lauffähig wird, ist der (mitgelieferte) Mikrocontroller vom Typ AT90S8515 polrichtig in den rechten 40-poligen Sockel einzusetzen. Jetzt müssen noch die Verbindungen zwischen den LEDs und dem Port A des Mikrocontrollers und die notwendige Verbindung zum Programmieren des Controllers hergestellt werden. Dazu verbindet man zunächst die mit „PORTA" und „LED" gekennzeichneten Stiftleisten mittels einer der mitgelieferten 10-poligen Flachbandleitungen. Die rot eingefärbte Ader der Leitung muss jeweils an Pin 1 des Steckverbinders liegen. Im nächsten Schritt wird eine Verbindung mittels der 6-poligen Flachbandleitung zwischen den Stiftleisten „ISP6PIN" und „SPROG3" hergestellt, wobei auch hier auf gleichsinnige Polung des Kabels (rote Ader an Pin 1) zu achten ist. Hiermit werden die Pins zur „ISP"-Programmierung zugeschaltet. Die Abkürzung ISP steht für „In-System-Programming" und bedeutet nichts anderes, als dass der Mikrocontroller direkt in der Schaltung programmierbar ist, sofern die entsprechenden Pins herausgeführt wurden. Abbildung 3 zeigt die komplette Verkabelung des Starterkits.

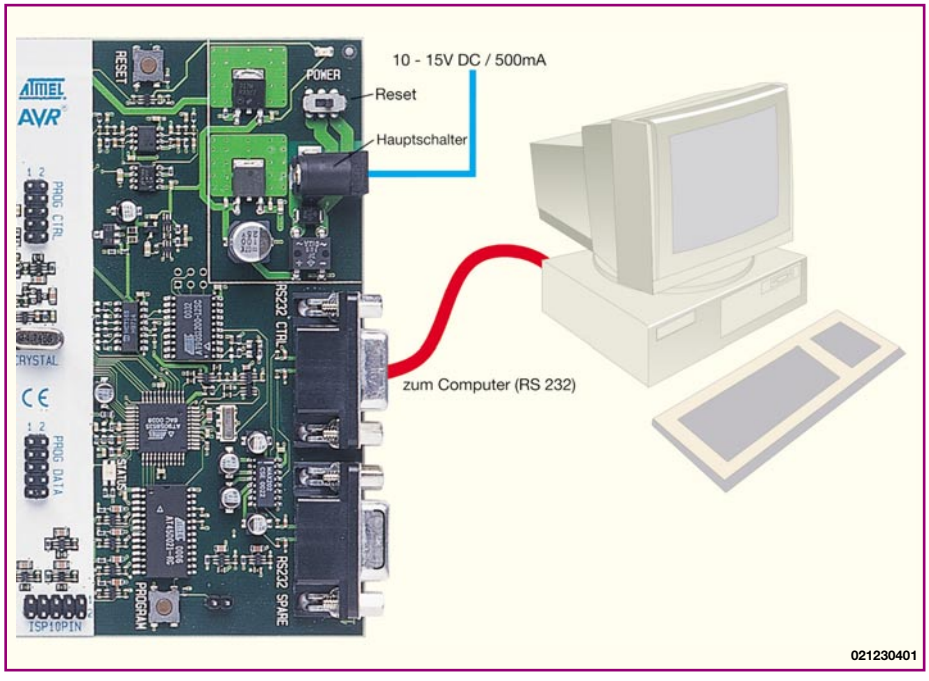

Bild 2: Anschließen des STK 500

Nachdem die vorhergehenden Schritte wie beschrieben durchgeführt worden sind, kann das STK 500 eingeschaltet werden und die Power-LED leuchtet. Beim Einschalten wechselt die Status-LED (links unterhalb des Flash-Speicherbausteins) von rot über orange auf grün und zeigt damit die Betriebsbereitschaft an.

Ausführen eines Programms

Bei der Programmentwicklung im dritten Teil der Artikelserie erfolgte die Implementierung eines Programms, das eine an Port A des AVR-Mikrocontrollers angeschlossene LED zum Blinken bringen soll. Dieses bereits kompilierte (übersetzte) und simulierte Programm wollen wir jetzt mit dem AVR-Starterkit testen. Dazu wird zunächst das AVR-Studio auf dem PC gestartet und über den Menüpunkt „Project→Open" das zuvor erstellte Projekt geöffnet. Um das Programm in den AVR-Mikrocontroller zu programmieren, ruft man den STK-500-Dialog über das Menü „Tools→STK 500" auf. Dieser Dialog enthält sechs Registerkarten (Dialog-Menüs), über die die Steuerung des Starterkits erfolgt. Jedoch soll an dieser Stelle nur auf die beiden wichtigsten Dialog-Menüs genauer eingegangen werden.

Das Register „Program" (Abbildung 4) enthält die Einstellmenüs zur Programmierung der Controller. Dort wird zunächst im Feld „Device" der verwendete Typ (hier: AT90S8515) aus einer Liste aller unterstützten Mikrocontroller ausgewählt. Anschließend stellt man den Programmiermodus im Feld „Programming mode" auf „ISP" ein, da wir ja das Starterkit entsprechend verdrahtet haben.

Jetzt ist im Feld „Flash" der Pfad der

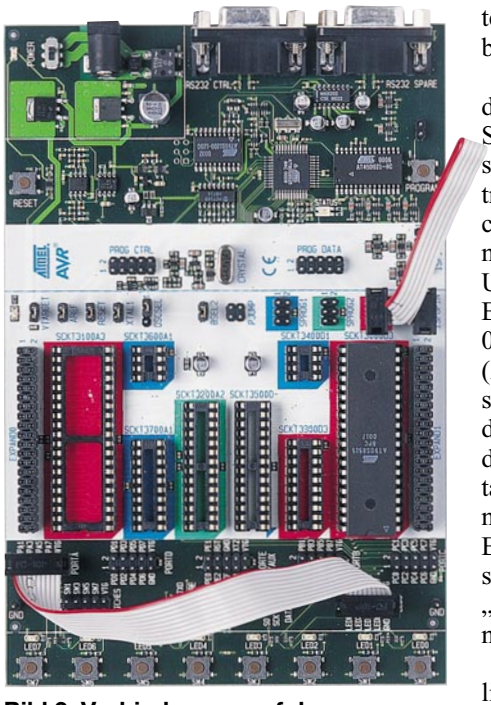

Bild 3: Verbindungen auf dem Starterkit

Hex-Datei, die beim Kompilieren des Projektes erstellt wurde, anzugeben. Im Normalfall enthält das Eingabefeld bereits den richtigen Pfad, wenn man das entsprechende Projekt zuvor geöffnet hat.

Im Anschluss daran kann über den Button „Erase Device" der eingelegte Baustein gelöscht und durch das Anwählen des „Program"-Buttons im Feld „Flash" programmiert werden. Mittels „Verify" ist der Inhalt des Programmspeichers kontrollierbar. Im unteren Bereich des Dialogfeldes werden die durchgeführten Vorgänge, sowie deren Ergebnisse protokolliert. Nach dem Programmieren läuft das Programm auf dem AVR-Starterkit, und die mit „LED0" gekennzeichnete LED blinkt.

Das letzte Feld dieses Registers ist das Feld „EEPROM". Es erlaubt das Programmieren des im Mikrocontroller integrierten EEPROMs auf die gleiche Weise wie beim Programmspeicher erläutert.

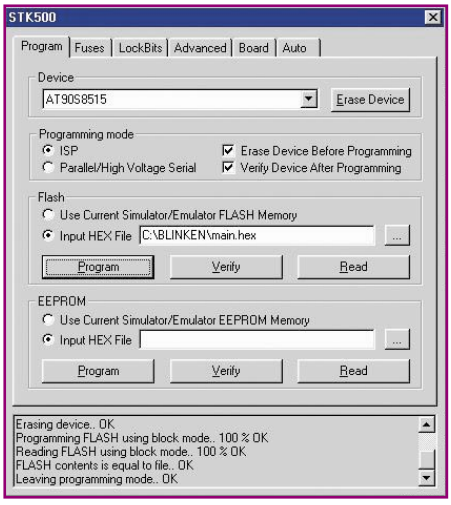

Bild 4: Programmierung der AVR-Controller

Im Register „Board" (Abbildung 5) kann die Einstellung wichtiger Parameter des STK 500 erfolgen. Im Feld „Voltages" stehen Einstellmöglichkeiten für die Betriebsspannungen der eingesetzten Mikrocontroller sowie für die integrierte Spannungsreferenz (z. B. für Analog-Digital-Umsetzer) zur Verfügung. Diese sind im Bereich von 0,0 bis 6,0 V in Schritten von 0,1 V einstellbar. Die Referenzspannung (ARef) kann nicht größer als die Betriebsspannung des Zielsystems gewählt werden, da dies den AVR-Controller beschädigen würde. Über den Button „Write Voltages" werden die entsprechenden Spannungen auf dem AVR-Starterkit eingestellt. Bei Bedarf lassen sich die aktuellen Einstellungen des Starterkits über den Button „Read Voltages" abrufen. So muss man nicht zum Multimeter greifen.

Das Feld „Oscillator and Reset" ermöglicht die Einstellung der Frequenz des Taktoszillators des STK 500. Die Liste enthält zwar nur einige wenige Frequenzen, jedoch ist der Taktoszillator in der Lage, Frequenzen in einem weiten Bereich zu erzeugen. Dazu ist die entsprechende Frequenz einfach in das Eingabefeld zu schreiben. Bei dieser Einstellung kann zwar nicht jeder beliebige Wert ausgewählt werden, jedoch berechnet die Software den nächstmöglichen Wert, der auch direkt angezeigt wird („Closest Attainable Freq."). Soll zum Beispiel eine Frequenz von 0,5 MHz programmiert werden, ist in das Eingabefeld „500000" einzugeben. Die Software schlägt aber einen Wert von 527 kHz vor, da der Taktoszillator die 0,5 MHz nicht genau erzeugen kann. Über den Button „Write Osc." wird die Frequenz auf dem STK 500 eingestellt. Im vorliegenden Beispiel der blinkenden LED ist eine Veränderung des Taktoszillators durch eine Änderung der Blinkfrequenz der LED zu erkennen.

Im Bereich „Revision" wird schließlich die Versionsnummer der Hard- und Firmware des STK 500 angezeigt.

Alle weiteren Registerkarten sollen hier nur kurz vorgestellt werden. Im Bereich „Fuses" befindet sich eine Liste aller setzbaren Sicherungen, die ein Fehlprogrammieren, Überschreiben usw. verhindern sollen, mit den Möglichkeiten zum Auslesen der aktuellen Konfiguration, sowie zum Setzen von neuen Einstellungen.

Ähnlich wie beim Registerbereich „Fuses" sind auf der Karte „LockBits" bestimmte Modi setzbar, in denen eine nochmalige Programmierung bzw. auch Verifikation des eingesetzten Mikrocontrollers gesperrt werden kann. Das Verringern einer einmal gewählten Sicherheitsstufe ist nur durch ein vollständiges Löschen des Bausteins erreichbar.

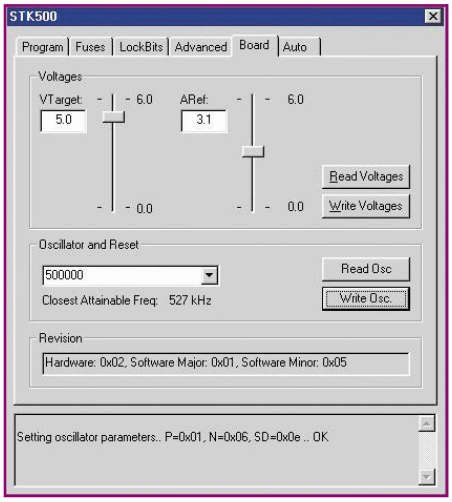

Bild 5: Einstellungen des STK 500

Der Bereich „Advanced" ist in zwei Untergruppen eingeteilt. Als erstes besteht die Möglichkeit, die Signaturbytes auszulesen und zum anderen kann der Kalibrierwert des Oszillators ausgelesen werden, welcher bei der Herstellung des Bausteins ermittelt wurde. Sollte der eingesetzte Baustein nicht über einen internen, abstimmbaren RC-Oszillator verfügen, kann man diesen Punkt nicht auswählen.

Auf der letzten verfügbaren Registerkarte („Auto") ist ein automatischer Ablauf zur Programmierung der Mikrocontroller einstellbar.

Damit sind bereits die wesentlichen Eigenschaften des AVR-STK-500 aufgezeigt. Der gesamte Funktionsumfang in allen Einzelheiten kann der Bedienungsanleitung zum STK 500, die im Internet unter [1] zum Download bereit steht, entnommen werden.

Mit dem gesamten enthaltenen Zubehör stellt das STK 500 eine professionelle Entwicklungsumgebung dar, mit der man die Programmierung sofort beginnen kann. Gerade deshalb ist das Kit inklusive des AVR Studio auch für den privaten Gebrauch zu empfehlen, denn man benötigt zunächst weder irgendwelche Kabel noch zusätzliche Software und ein Controller für die ersten Versuche liegt dem Kit auch schon bei.

Im fünften Teil dieser Artikelserie wollen wir auf weitere interessante Möglichkeiten bei der Programmierung mit AVR-Mikrocontrollern eingehen. ELV

Internet:

[1] Atmel STK 500 User-Guide-Bedienungsanleitung STK 500: http://www.atmel.com/atmel/acrobat/doc1925.pdf

[2] Aktuelle Version des Atmel AVR-Studio: http://www.atmel.com/atmel/products/prod203.htm