



# AVR-Grundlagen Teil 6

**In den bisherigen Teilen dieser Serie haben wir uns mit dem Aufbau und den Programmiergrundlagen für die AVR-Mikrocontroller beschäftigt. Jetzt geht es daran, das Erlernete anzuwenden, also Programme in den Prozessor zu laden. Hierfür stellen wir ein preiswertes Tool für das Programmieren gängiger AVR-Mikrocontroller vor.**

## Programmierung im System

Das AVR-Starterkit und andere Spezial-Programmer stellen wohl eine äußerst komfortable Möglichkeit dar, die AVR-Mikrocontroller u.a. auch zu programmieren. Aber diese Geräte sind eben auch relativ teuer und für die einfache Programmierung des Controllers werden ihre vielen Möglichkeiten lange nicht benötigt. Speziell das Starterkit ist ja auch primär für das einfache Erlernen des Umgangs mit dem AVR-Mikrocontroller konzipiert und bietet das entsprechend komfortable Umfeld. Wesentlich einfacher kann es zugehen, wenn bereits eine Anwendungsschaltung fertig ist, die die Spannungs- und Taktversorgung des Controllers ohnehin übernimmt. Dies nutzt unser vom Schaltungsaufwand her einfaches und damit preiswertes Programmiergerät. Es verwendet die „In-System-Programming“-Schnittstelle der AVR-Mikrocontroller, um diese di-

rekt in der Zielschaltung zu programmieren. Hierzu muss und darf der Prozessor nicht aus der Schaltung herausgenommen werden, da die Spannungsversorgung des Programmieradapters aus dem Zielsystem heraus erfolgt. Der für den Programmiervorgang ebenfalls notwendige Takt muss auch durch die Anwendungsschaltung zur Verfügung gestellt werden. Dieses sollte, wie bereits erwähnt, kein Problem darstellen, da für den Betrieb des Prozessors in der Schaltung sowieso eine Taktversorgung zur Verfügung stehen muss.

Die Verbindung mit dem Zielsystem erfolgt über eine kleine, sechspolige Stiftleiste, die über eine Flachbandleitung kontaktiert wird. Für die Kontaktierung in der Anwendungsschaltung ist ein entsprechender Pfostenstecker vorgesehen, der inklusive seiner Verbindungen natürlich in das Platinenlayout der Anwendungsschaltung einzubinden ist.

Der eigentliche Programmiervorgang wird von einer PC-Software gesteuert. Die-

se öffnet die Datendatei, bereitet die Datenpakete entsprechend auf und sendet sie über die RS-232-Schnittstelle des Rechners an den AVR-Programmieradapter. Die Datendatei kann sowohl im Intel-HEX- oder im Binärformat vorliegen, da unsere PC-Software beide Formate verarbeiten kann. Der Vorteil liegt auf der Hand: Oftmals arbeiten billige Programmiergeräte nur mit einem der beiden Formate, so dass dort vorher noch eine entsprechende Umwandlung erfolgen muss.

Auf den Ablauf des Programmiervorgangs sowie die Bedienung und die Funktionen des Programmiergerätes kommen wir später ausführlich zurück. Zunächst soll zum Verständnis der Funktion der Programmieralgorithmus der AVR-Mikrocontroller etwas genauer erläutert werden.

## Der Programmieralgorithmus

Die meisten AVR-Mikrocontroller sind zwar auf verschiedene Arten programmierbar, jedoch beherrscht nicht jeder Typ alle Programmieralgorithmen - meist aus Gründen der jeweiligen Pinanzahl des Controllers.

Grundsätzlich ist zu sagen, dass es für die verschiedenen Controller-Arten auch unterschiedliche Befehlssätze gibt, die hinsichtlich ihrer Parameter und des Ablaufs variieren. Außerdem gibt es einige Funktionsbits, die nur in bestimmten Programmierverfahren beeinflussbar sind.

## Parallele Programmierung

In der Reihe der Methoden soll als erste die parallele Programmierung genannt werden. Hierbei erfolgt die Ein- bzw. Ausgabe der Daten beim Schreiben bzw. Lesen vom Mikrocontroller über 8 Datenleitungen, so dass immer ein komplettes Byte in einem Schritt übertragen wird. Die Ablaufregelung erfolgt über weitere Steuerleitungen.

Bei diesem Algorithmus können die Befehle aus einem oder mehreren Bytes bestehen. Zum Beispiel setzt sich der Befehl zum Löschen des Flash-ROMs nur aus einem Byte zusammen - er ist damit in einem Taktzyklus ausführbar. Dagegen erfordert der Befehl zum Schreiben oder Lesen eines Datenbytes in bzw. aus dem Mikrocontroller weitere wichtige Parameter wie z. B. die Adresse und das Datenbyte an sich, die mit übertragen werden müssen und somit auch mehrere Taktzyklen voraussetzen.

Der Vorteil dieses Verfahrens ist der, dass die Dauer des Programmiervorganges relativ kurz ausfällt, was sich gerade bei Bausteinen mit großem Flash-ROM und entsprechend umfangreichen Programmen bemerkbar macht. Außerdem sind hier alle vorhandenen Funktionsbits (Fuse-Bits,

**Tabelle 1: Befehlssatz am Beispiel des AT90S8515**

Befehl	Befehlsformat				Beschreibung
	Byte 1	Byte 2	Byte 3	Byte 4	
Programming enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Enable serial programming while RESET is low.
Chip erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash and EEPROM memory arrays.
Read Program Memory	0010 <b>H</b> 000	xxxx <b>aaaa</b>	<b>bbbb bbbb</b>	<b>oooo oooo</b>	Read <b>H</b> (high or low) data <b>o</b> from program memory at word address <b>a:b</b>
Write Program Memory	0100 <b>H</b> 000	xxxx <b>aaaa</b>	<b>bbbb bbbb</b>	<b>iiii iiii</b>	Write <b>H</b> (high or low) data <b>i</b> to program memory at word address <b>a:b</b>
Read EEPROM Memory	1010 0000	xxxx <b>xxx</b> a	<b>bbbb bbbb</b>	<b>oooo oooo</b>	Read data <b>o</b> from EEPROM memory at address <b>a:b</b> .
Write EEPROM Memory	1100 0000	xxxx <b>xxx</b> a	<b>bbbb bbbb</b>	<b>iiii iiii</b>	Write data <b>i</b> to EEPROM memory at address <b>a:b</b>
Write Lock Bits	1010 1100	111x <b>x21x</b>	xxxx xxxx	xxxx xxxx	Write Lock bits. Set bits <b>1,2</b> = "0" to program Lock bits.
Read Signature Bytes	0011 0000	xxxx xxxx	xxxx <b>xxbb</b>	<b>oooo oooo</b>	Read signature byte <b>o</b> at address <b>b</b> .

**Legende:**

<b>a</b> address bits high	<b>i</b> data in
<b>b</b> address bits low	<b>x</b> don't care
<b>H</b> 0 - Low byte, 1 - High byte	<b>1</b> Lock bit 1
<b>o</b> data out	<b>2</b> Lock bit 2

Lock-Bits) programmierbar. Die große Anzahl von benötigten Signalleitungen schränkt dieses Verfahren jedoch in Hinsicht auf die Programmierung im Zielsystem ein und ist hauptsächlich für Programmiergeräte mit entsprechendem Sokkel vorgesehen.

### Serielle Programmierung mit erhöhter Spannung

Der zweite Programmieralgorithmus wird im Datenblatt der AVR-Mikrocontroller als „High-Voltage-Serial-Programming“ bezeichnet und hauptsächlich bei den Bausteinen eingesetzt, die für die parallele Programmierung über zu wenig Anschlusspins verfügen. In diesem Programmierverfahren wird an einigen Pins eine höhere Spannung verwendet (11,5-12,5 V), wobei aber auch der Bereich der Versorgungsspannung nach unten hin eingeschränkt ist. Auch bei diesem Verfahren sind die entsprechenden Funktionsbits voll zugänglich.

Die Anzahl der verwendeten Signalleitungen ist in Bezug auf das vorher beschriebene Verfahren zwar bereits eingeschränkt, es hat aber den Nachteil, dass zusätzlich eine höhere Spannung zur Verfügung stehen muss.

Bei der Übertragung der Daten ist es auch nicht möglich, in einem Taktzyklus ein ganzes Byte zu senden, da dieses in serieller Form zum Mikrocontroller geschickt werden muss, was die erforderliche Anzahl der Zyklen erheblich erhöht.

Die Befehlsketten haben auch hier eine variable Länge, die zwischen zwei und vier Bytes pro Befehl variieren kann.

### Einfache serielle Programmierung

Das dritte mögliche Programmierverfahren, bei dem die Daten ebenfalls seriell übertragen werden, ist für die Program-

mierung der Mikrocontroller im Zielsystem anwendbar und kommt auch in dem hier beschriebenen AVR-Programmieradapter APA 100 zur Anwendung.

Die Programmierschnittstelle besteht hier nur aus vier Signalleitungen, was eine Kontaktierung sehr einfach macht. Zusätzlich zu den Signalen ist aber noch die Spannungs- und Taktversorgung bereitzustellen, die vom Zielsystem übernommen wird, da diese dort ohnehin zum Betrieb benötigt wird. Die noch erforderlichen Signalleitungen realisieren den Reset, den Dateneingang, den Datenausgang und den Takt für die Datenübertragung.

Die Befehle des Befehlssatzes haben, im Gegensatz zu den beiden zuvor genannten Methoden, eine feste Länge von vier Bytes, wobei nicht immer alle vier Bytes vorgegeben sind, aber immer vier übertragen werden müssen. Der Befehlssatz des AVR-Mikrocontroller kann mehr oder weniger Befehle aufweisen - je nach Art und Ausstattung des jeweiligen Bausteins. Ein Beispiel für einen Befehlssatz ist in Tabelle 1 zu sehen.

Der Ablauf der Programmierung ist grundsätzlich immer gleich und soll im Folgenden kurz beschrieben werden. Im ersten Schritt erfolgt das Anlegen der Betriebsspannung. Nach einer kurzen Wartezeit wird der Befehl zur Freischaltung des Programmiermodus über die serielle Schnittstelle an den AVR-Baustein gesendet, der das Kommando entsprechend bestätigt. In diesem kann jetzt entweder der komplette Speicherbereich des Mikrocontrollers gelöscht werden oder die Programmierung kann direkt beginnen. Der Programmspeicher und das interne EEPROM der AVR werden byteweise beschrieben. Direkt nach dem Schreiben eines Bytes muss eine kurze Zeit gewartet werden, bis das Flash-ROM den Wert übernommen

hat. Eine weitere Möglichkeit ist die, die zuvor beschriebene Speicherstelle immer wieder auszulesen, bis der entsprechende Wert wieder erfasst werden konnte. Dieser Vorgang wiederholt sich für jede zu beschreibende Speicherstelle.

Nach dem Beschreiben des Speichers kann jedes Byte zum Vergleichen wieder ausgelesen werden, sodass eine volle Kontrolle über den Programmiervorgang möglich ist.

Der genaue Ablauf der Programmierung für die einzelnen Mikrocontroller, sowie die individuellen Befehlssätze sind den entsprechenden Datenblättern zu entnehmen, die unter [1] im Internet verfügbar sind.

Der Flash-Speicher der AVR-Mikrocontroller hat eine Haltbarkeit von bis zu 1000 Schreib-Lösch-Zyklen und weist somit einen erheblichen Vorteil gegenüber den OTP-Typen (OTP - One-Time-Programmable, einmal programmierbar) auf. Das interne EEPROM verfügt über eine Lebensdauer von bis zu 100000 Schreib-Lese-Zyklen und ist damit sehr gut für die Speicherung von schaltungsspezifischen Werten (z. B. Abgleichwerte o. ä.) einsetzbar.

Damit sind die grundlegenden Eigenschaften und Funktionsweisen des Programmiergerätes aufgezeigt. Im nächsten und letzten Teil dieser Artikelserie wird abschließend die Schaltungstechnik und die Bedienung des AVR-Programmieradapters detailliert erläutert. **ELV**

#### Internet:

[1] Datenblätter der AVR-Mikrocontroller  
<http://www.atmel.com/atmel/products/prod200.htm>