

Eine Hand wäscht die andere

UniProg: 16-Bit-Controller programmiert 8-Bit-Controller

Christian Perschl

Einleitung

Für viele einfache Steuerungs- oder Regelungsaufgaben reichen die immer noch sehr verbreiteten 8-Bit-Controller der 8051/C500-Familie. Wenn schon einfach, dann wäre eine Single-Chip-Applikation angebracht, d.h. alle Programm- und Datenressourcen im Chip intern, keine externen Speicherbausteine, die das Design oft erheblich verkomplizieren.

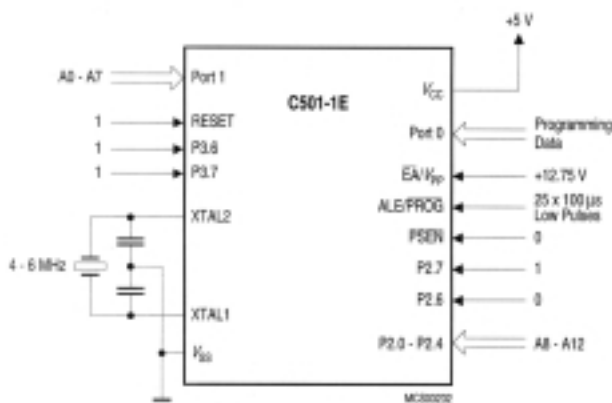
Der C501G-1EP ist ein 80C32 mit internem OTP-Speicher. OTP bedeutet One Time Programmable = genau einmal programmierbar. Für Evaluierungszwecke ist diese Speicherart zwar nicht so geeignet, denn einmal falsch programmiert, läßt sich der interne Speicher im Gegensatz zu einem EPROM- oder FLASH-Baustein nicht mehr löschen oder neu programmieren.

Ist die Designphase abgeschlossen, so bietet jedoch der interne OTP-Speicher eine interessante Variante sowohl für kleinste Stückzahlen als auch größere Produktionen.

Aber wie den internen OTP-Speicher programmieren ?

Programmierung des OTP

Die Programmierung des C501G-1EP geschieht extern durch Anlegen von Daten, Adressen und Steuerleitungen. Zusätzlich muss ein Taktsignal für die interne Programmierhardware angelegt werden.



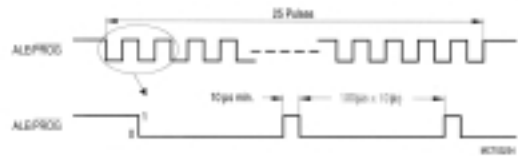
Die zu programmierende Adresse im 8 KByte großen internen OTP wird an Port 1 bzw. den unteren 5 Bits des Port 2 angelegt. Die Daten liegen am Port 0 an. Der Reset-Eingang bleibt während der Programmierung auf logisch 1. Dann werden noch ein paar Steuersignale an den Pins P3.6, P3.7, P2.6, P2.7 und PSEN angelegt.

Ein externer Takt an XTAL1/XTAL2 ist ebenfalls erforderlich.

Die eigentliche Programmierung erfolgt durch Anlegen der Programmierspannung von 12.75 V und der Erzeugung von Programmierimpulsen am Pin ALE/PROG:

Es gibt auch die Möglichkeit, die Programmierung zu überprüfen: Um den Speicher auszulesen, muss am Pin 2.6 logisch 1 anliegen. Zusätzlich muss der ALE/PROG-Eingang und der Vpp-Pin

auf 1 liegen. Nach Anlegen der Adresse können am Port 0 der Inhalt byteweise ausgelesen werden.



Kompliziert? Na ja, zumindest ein bisschen Aufwand. Ich möchte im folgenden eine Komplettlösung vorstellen, die jeder nachbauen kann und kaum Aufwand darstellt. Was wird benötigt? Ein C167-Starterkit, ein bisschen Lötfertigkeit und ein paar Standardbauteile.

UniProg

UniProg ist ein universelles Programmierwerkzeug zur Programmierung des internen OTP (One-Time-Programmable) des C501G-1EP. Dabei wird ein Standard-C167-Starterkit zur Ausführung des Programmieralgorithmus verwendet. Ein Zusatzmodul, welches auf die Starterkit-Verbindungsleiste gesteckt wird, nimmt dann in einem ZIF (Zero Insert Force) – Sockel den zu programmierenden Mikrocontroller auf.



Das auf dem Host laufende Frontend-Programm von UniProg ermöglicht ein komfortables Laden und Editieren von Intel-Hexdateien und eine ebenso einfache Programmierung. Außerdem kann durch das vorgestellte Konzept UniProg einfach für weitere Flash/OTP-Module angepasst werden.

Installation

Folgende Komponenten werden benötigt:

- PC mit Windows 95/98/NT
- C167 Starterkit mit seriellem Kabel und Netzteil
- Aufsteckmodul C501G-1EP
- Softwarepaket UniProg

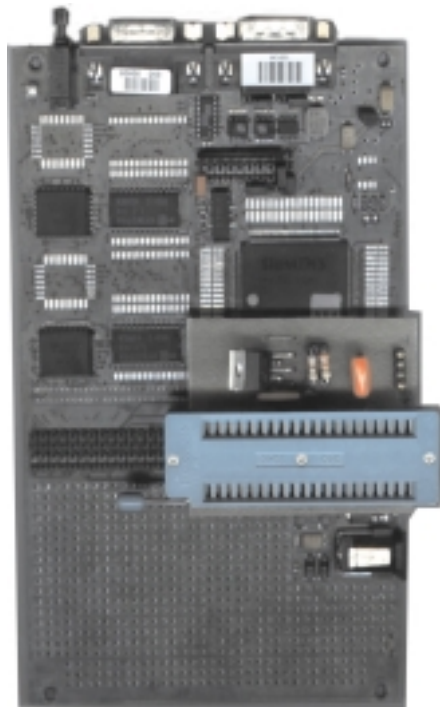
Installation des Softwarepakets

Als erstes muss das Softwarepaket UniProg installiert werden. Dabei wird zuerst das Programm SETUP.EXE aufgerufen.

Bei der nachfolgenden Installationsroutine muss lediglich der Ziel-Pfad angegeben werden.

Inbetriebnahme des Starter Kits

- Der Starterkit muss mit allen Standardeinstellungen in Betrieb genommen werden (alle Jumper wie beim Auslieferungszustand.).
- Sodann muss das Starterkitboard mit dem seriellen Kabel mit dem PC verbunden und an das Netzteil angeschlossen werden.
- Als nächstes muss der Bootstrap-Loader (roter Jumper) aktiviert werden (wenn nicht bereits gesetzt)
- Jetzt kann der Starterkit programmiert werden. Die dem Softwarepaket beiliegende Datei UNIPROG.H86 (ist im Uniprogrammverzeichnis im Unterverzeichnis MCPROG zu finden) muss nun in das externe Flash des Starterkits programmiert werden. Dazu kann z.B. das Phytec-Flashtool oder Minimon verwendet werden.
- Dann muss der Bootstrap Loader wieder deaktiviert werden (roter Jumper aus). Dies ist erforderlich, damit der C167 auf dem Starterkit das soeben programmierte UNIPROG.H86 nach dem Einschalten ausführt.
- Als letztes wird das Uniprogramm-Modul auf die Starterkit-Verbindungsleiste gesteckt. Es empfiehlt sich, ein einmal (richtig) gestecktes Modul nicht wieder herunterzunehmen, da es dadurch leicht beschädigt werden kann. Das Modul muss wie in der folgenden Abbildung eingesteckt werden:



Bedienung

Einstecken des C501G-1EP

Der Zeitpunkt zum Einstecken des zu programmierenden Mikrocontrollers ist an und für sich kein Problem, er kann also entweder vor dem Einschalten oder während des Betriebes – natürlich nicht während des Programmierens – gewechselt werden, da Uniprogramm sich standardmäßig im Lesemodus befindet und daher die meisten I/O-Leitungen auf Eingang geschaltet sind.

Wichtig ist dabei lediglich, dass der C501 mit Pin 1 zur Mitte des Starterkits hin (beim Hebel des ZIF-Sockels) platziert wird:



Anlegen der Programmierspannung

Da der C501G-1EP eine Programmierspannung von 12,75 V benötigt, muss eine Spannung von ca. 15 V wie beschriftet am Adapter angelegt werden. Ein Spannungsregler erzeugt genau die erforderlichen 12,75 V beim Programmieren bzw. 5 V beim Lesen.

Frontend

Uniprogramm enthält ein einfach zu bedienendes Frontend, welches es ermöglicht, den C501-OTP-Speicher zu programmieren, den OTP-Speicher auszulesen oder Inhalte zu vergleichen.

Das Frontend besteht aus folgenden Komponenten:

- Menü und Toolbar: Hier können alle Befehle aufgerufen werden
- Speicheranzeige: Hier wird der aktuelle Inhalt des Frontend-Buffers angezeigt oder geändert.
- Statusfenster: Hier werden Statusmeldungen angezeigt

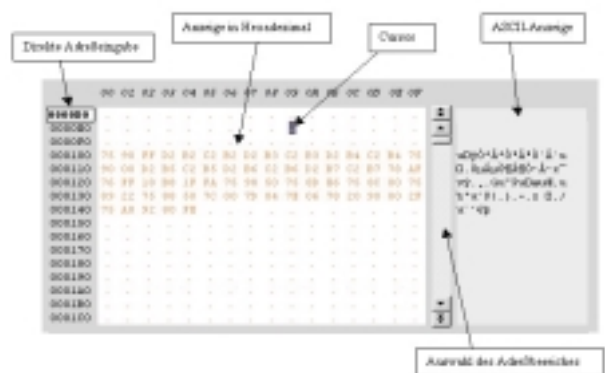


Kommunikation herstellen

Um eine Kommunikationsverbindung herzustellen, muss der Menübefehl *Target - Connect* aufgerufen werden. Alternativ kann auf die Symbolschaltfläche *Connect* (rot) geklickt werden. Im Statusfenster unten sollte dann eine entsprechende Meldung angezeigt werden.

Die Speicheranzeige

Der aktuelle Inhalt des Frontendbuffers wird in der Speicheranzeige angezeigt. Dabei werden immer 256 Bytes angezeigt:



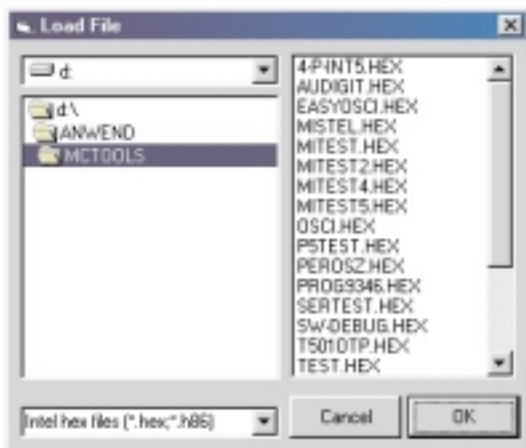
- Welcher Adressbereich gerade angezeigt wird, kann mit dem Rollbalken rechts bestimmt werden.
- Es kann auch direkt eine Adresse angegeben werden, und zwar rechts oben im Feld "Direkte Adresseingabe".

- Mit dem Cursor kann eine bestimmte "Zelle" angefahren werden, entweder mit den Pfeiltasten oder mit der Maus. Das Datenbyte an der Cursorposition kann durch Eingabe eines Hexwertes geändert werden.
- Nicht verwendete Zellen werden mit einem Punkt dargestellt. Verwendete Daten können noch unterschieden werden, ob sie zur Zeit nur im Frontend-Buffer stehen (rot), oder ob sie tatsächlich dem Inhalt des OTP-Speichers entsprechen.
- Zusätzlich wird der aktuelle Adressbereich auch mit ASCII-Zeichen dargestellt (rechts).

Ein Bereich kann mit der Maus selektiert werden: Dazu wird der zu selektierende Bereich mit der Maus gezogen.

Datei laden

Um eine Intel-Hex-Datei in den Frontendbuffer zu laden, muss der Menü-Befehl *File Load* angeklickt werden. Danach erscheint ein Fenster zum Auswählen der Datei:



Nachdem eine Datei ausgewählt wurde, kann diese mit OK geladen werden. Sie wird dann auch gleich in der Speicheranzeige selektiert und dargestellt.

Programmieren des OTP Speichers

Der OTP-Speicher des aktuell im Sockel steckenden Mikrocontrollers kann bytewise programmiert werden. Es wird immer die aktuelle Selektion programmiert!

Die Aktion wird durch den Befehl *Target-Program Selection* im Menü oder das entsprechende Symbol durchgeführt.

Auslesen des OTP Speichers

Der OTP-Speicher des aktuell im Sockel steckenden Mikrocontrollers kann auch bytewise ausgelesen werden. Wie beim Programmieren wird die aktuelle Selektion ausgelesen! Daher muss vor dem Auslesen der gewünschte Bereich selektiert werden.

Die Aktion wird durch den Befehl *Target-Read Selection* im Menü oder das entsprechende Symbol durchgeführt.

Vergleichen

Eine einfache Möglichkeit, den Inhalt des OTP-Speichers mit dem Inhalt des Frontend-Buffers zu vergleichen, bietet der Menübefehl *Target - Compare Selection*: Es muss vor Ausführung des Befehls wiederum der zu vergleichende Bereich selektiert werden (Zur Erinnerung: Dateien werden nach dem Laden selektiert, d.h. durch Aufrufen von *File-Load* und *Target-Compare Selection* wird der Inhalt des OTP-Speichers mit der gewünschten Datei verifiziert).

Ergebnis des Vergleiches: übereinstimmende Datenbytes werden schwarz dargestellt, nicht übereinstimmende rot.

Informationen zur Herstellung des Moduls

Beschaltung

- Die meisten Pins des Moduls (Adressen, Daten, Steuerleitungen) sind mit I/O-Leitungen des C167CR auf dem Starterkit verbunden. Diese werden je nach Bedarf entweder als Eingänge oder als Ausgänge geschaltet.
- Die Belegung der I/O Pins wurde so gewählt, dass sich ein minimaler "Verdrahtungsaufwand" ergibt, also keinesfalls nach den Ports des C167CR (keine "logische" Zuordnung).
- Die Adressleitungen sind immer auf Ausgang geschaltet.
- Ebenso sind die Modusverbindungen immer auf Ausgang geschaltet (Modus: Programmieren/Lesen).
- Die Datenleitungen werden je nach Modus auf Eingang oder Ausgang geschaltet.
- Der Takteingang wird mit einem Digitalsignal von der PLL-Einheit des C167CR versorgt (ca. 5 MHz).

Programmierspannungssteuerung

Da der C501G-1EP eine Programmierspannung von 12,75 V benötigt, muss eine Spannung von ca. 15 V wie beschriftet am Adapter angelegt werden. Der Standard-Spannungsregler L200 (erhältlich bei Conrad Electronics) erzeugt genau die erforderlichen 12,75 V beim Programmieren bzw. 5 V beim Lesen. Zwischen diesen beiden Spannungen umgeschaltet wird mittels I/O-Leitung und Ansteuerung eines Transistors.

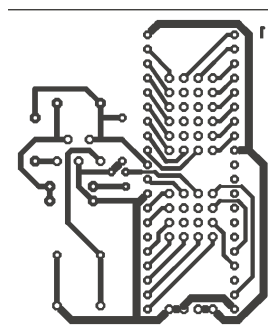
Die Spannungen ergeben sich durch den Spannungsteiler R2/R1 und errechnen sich nach folgender Formel:

$$U = 2,75 * (1 + R2/R1)$$

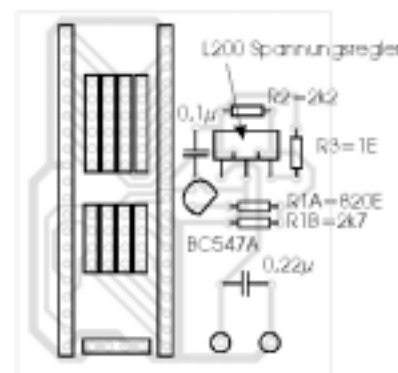
Bei ausgeschalteter Programmierspannung ergibt lediglich R2/R1B den Spannungsteiler (U=5V). Bei eingeschalteter Programmierspannung (U=12,75V) wird R1A parallel zu R1B geschaltet.

Bauanleitung

Zur Selbstbauanleitung werden hier Layout und Bestückungsplan angegeben:



Layout 1:1, Lötseite



Fazit

Es gibt natürlich professionelle Programmiergeräte für den C501-OTP, nur sind diese für den Hobbyelektroniker, Schulen oder einen kleinen Betrieb mit hohen Investitionskosten verbunden. Die hier vorgestellte Lösung "Uniprogramm" ist eine recht günstige Alternative (Starterkit: ca. 2000,- Modul ca. 300,-) zum Selbstbau. Die erforderliche Software ist frei verfügbar.