

Kleiner Monitor ganz groß

MINIMON

Systemmonitor für die Infineon 16-Bit Mikrocontroller C16x

Christian Perschl, Peter Kliegelhöfer



Mikrocontroller

Überall sind sie in Massen zu finden: in Automobilen, Haushaltsgeräten, Computerteilen oder Videorekordern: die **Mikrocontroller**. Sie stellen die „Eierlegenden Wollmilchsäue“ der großen Familie von Integrierten Schaltkreisen dar: Ein Alleskönner soll sämtliche „intelligente“ Aufgaben von der Messwertaufnahme, Abfrage von Sensoren bis zur Signalerzeugung übernehmen. Deshalb sind im Mikrocontroller neben der CPU auch eine Reihe von Peripherieeinheiten (Timer, Standard-Schnittstellen, A/D-Wandler, digitale Ein-/Ausgänge) enthalten.

In allen Bereichen der Anwendungen von Mikrocontrollern (Automotive, Computer Peripheral, Consumer Electronics, Steuerungs- und Regelungssysteme) ist ein immer stärkerer Trend zu **Single-Chip**-Applikationen erkennbar: Nicht zuletzt aus Kostengründen soll dabei auf alle externe Hardware, speziell auf Speicherbausteine verzichtet werden und für die Applikation lediglich interne Speicherressourcen des Controllers wie Flash oder OTP (One Time Programmable) verwendet werden. Ein weiteres Argument für die Verwendung von internen Flash-Speichern ist die Wartbarkeit: Neuere Programmversionen sollen jederzeit und einfach einzuspielen sein, ohne dass nur eine Hardwarekomponente ausgetauscht oder geändert werden muss.

Mikrocontroller der 16 Bit-Klasse sind vielfach für einen Single-Chip-Betrieb ausgelegt: So existieren schon recht lange Derivate mit internem Program-Flash, den Durchbruch schafften diese Bausteine jedoch erst im Laufe der letzten Jahre.

Was ist Minimon ?

Mikrocontroller leben, mehr als alle anderen elektronischen Bausteine, von den verfügbaren Tools. Eine Reihe von Compilern, Programmierutilities, Monitoren und andere Softwareentwicklungstools unterstützt den Entwickler von Mikrocontrolleranwendungen beim Entwurf und Testen des Produktes.

Dieser relativ neue Trend der Single-Chip-Applikationen schafft aber auch für Tools neue Anforderungen: Speziell Tools, die direkt mit der Zielhardware in Kontakt stehen, sollen die vorhandene Hardware möglichst ressourcensparend verwenden und flexibel in bezug auf Lokatierung und Erweiterungen sein. Keinesfalls sollen bestimmte Bereiche vom Monitor blockiert sein. Diese Anforderungen haben zum Universal-Tool „Minimon“ geführt:

Minimon ist ein ressourcensparender Monitor, welcher mittels Bootvorgang in das interne RAM des Mikrocontrollers geladen wird. Er ist nicht auf externe Ressourcen angewiesen und kann jederzeit in einen anderen Speicherbereich verschoben werden. Nicht zuletzt deshalb ist Minimon für Single-Chip-Applikationen prädestiniert.

Trotz der „Winzigkeit“ des Monitors (ca. 400 Bytes) ist der Funktionsumfang beachtlich:

- Beschreiben, Auslesen und Vergleichen von beliebigen Speicherbereichen im Chip
- Setzen und Auswerten von Peripherieregistern
- Laden und Starten von Programmen / Subroutinen

- Programmieren des OnChip Flash-/OTP-Speichers

Ein Frontend sorgt für einen komfortablen Umgang mit diesen Funktionen. Die Voraussetzungen für die Benutzung von Minimon sind: ein System mit Windows 95/98/NT®, Zielhardware mit einem beliebigen C16x 16Bit-Mikrocontroller und eine serielle Verbindung zwischen beiden (z.B. C16x Starterkit, Prototypenboard).

Monitorkonzept

Das Programm Minimon umschließt zwei Komponenten: den Monitor-Kernel, welcher auf der C16x-Zielhardware läuft, und das Frontend unter Windows, mit welchem alle Einstellungen und Aktionen bequem vorgenommen werden können. Kommuniziert wird über ein einfaches Protokoll, welches gut dokumentiert und frei verfügbar ist.



Der Monitor-Kernel ist dafür ausgelegt, im internen RAM des Controllers Platz zu finden. Er ist lediglich ein paar hundert Bytes groß und kann nach dem Booten jederzeit in einen beliebigen anderen Bereich (z.B. externes RAM) verlegt werden. Der Kernel kennt nur ein paar rudimentäre Kommandos wie Speicherblock lesen/schreiben, Prozeduraufruf und Softwarereset. Mit diesen wenigen Grundbefehlen können alle komplexeren Operationen durch das Frontend realisiert werden. Dies bedeutet, dass so viel wie möglich Intelligenz in das Frontend-Programm verlagert wurde und damit der Monitor-Kernel schlank bleibt.

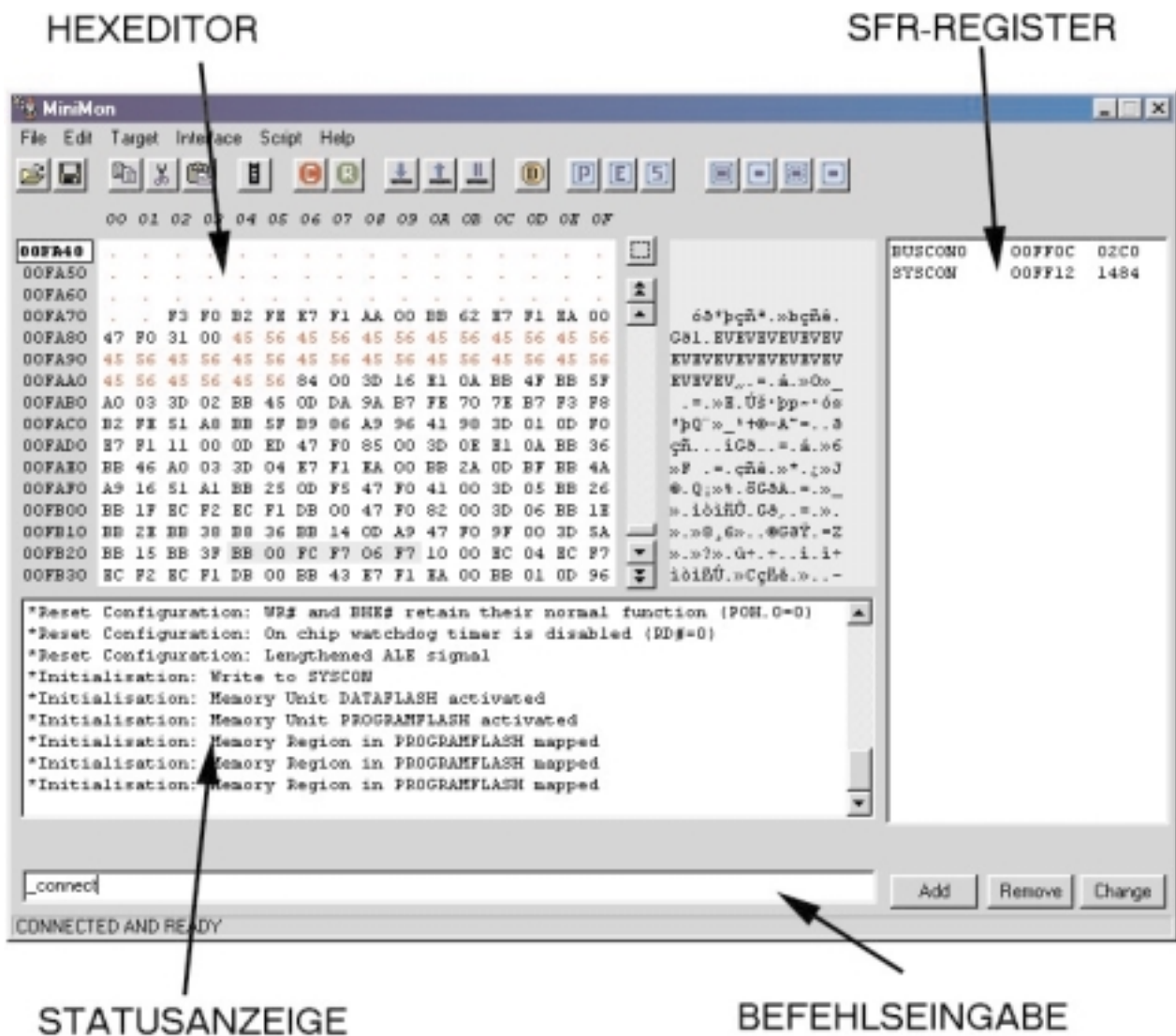
Der Verbindungsaufbau zwischen Kernel und Frontend erfolgt im allgemeinen durch einen Bootvorgang. Es ist allerdings auch möglich, den Kernel dauerhaft im Targetspeicher abzulegen.

Frontend-Funktionen

Das Frontend realisiert eine Reihe von komfortablen Funktionen:

Ein zentraler Bestandteil des Frontends ist der **Hexeditor**. Hier können beliebige Speicherbereiche angezeigt, editiert, Intel-Hex-Dateien geladen und gespeichert werden. Beliebige, auch unzusammenhängende Adressbereiche können selektiert und mit bestimmten Werten gefüllt werden. Der Hexeditor kann auch unabhängig von eventuell vorhandener Targethardware verwendet werden.

Mit dem Hexeditor kooperieren die **Transferfunktionen**. Die im Hexeditor selektierten Speicherbereiche können in den Speicher



der Zielhardware geschrieben oder von dort gelesen werden. Auch eine Vergleichsfunktion steht zur Verfügung, z.B. um den Inhalt des Speichers mit einer Datei zu vergleichen.

Handelt es sich nicht um RAM-, sondern um Flash- oder OTP-Speicherbereiche, so können diese programmiert werden. Die **Flashprogrammierung** stellt ebenfalls eine der Hauptanwendungen von Minimom dar. Der Monitor unterstützt von Grund auf alle internen Flash- und OTP-Typen der Familie. Minimom kann aber jederzeit um eigene Treiber erweitert werden, da sowohl Kommunikationsprotokoll als auch Treiber-Interface offen und dokumentiert sind.

Eine weitere wichtige Funktion von Minimom ist das Laden und **Starten von Applikationen**. Das Programm wird im Intel-Hex Format in den Hexeditor geladen und mit der Transferfunktion „Download“ in den Speicher verfrachtet oder in das Flash programmiert. Danach kann das Programm per Minimom „angesprungen“ werden oder ein Software-Reset durchgeführt werden. Eine interessante dritte Möglichkeit ist die Verwendung des Subroutinen-Interfaces (verwendet von den Flashtreibern). Hier können beliebige Programme als Unterprogramme des Monitors aufgerufen werden. Dies ermöglicht eine geordnete Rückkehr in den Monitor nach Terminierung des Programms. Ideales Einsatzgebiet: kleine Testroutinen für Peripherie oder Hardware.

Ist eine Anwendung einmal gestartet, so kann die (serielle) Kommunikation mit Minimom erfolgen, und zwar mit der **Terminalfunktion**: Empfangene Daten von der Applikation werden in einem Terminalfenster entweder als Hexadezimalwerte oder als

Text angezeigt. Auch das Senden von Hex-Zeichen oder Text zur Applikation ist mit Minimom einfach möglich.

Das „**Systemmonitoring**“ stellt ebenfalls eine der Hauptanwendungsbereiche von Minimom dar. Gemeint ist damit das Initialisieren, Auslesen und Setzen von Peripherieeinheiten. So kann z.B. ganz ohne Programmierung ein Port-Pin gesetzt, ein Timer initialisiert oder ein Wert vom A/D-Wandler eingelesen werden. Auch zur Hardware-Evaluierung läßt sich damit Minimom sehr gut einsetzen. Häufig benötigte SFR-Register können im Minimom-Hauptfenster angezeigt und werden auch laufend aktualisiert.

So nebenbei bietet Minimom auch die Möglichkeit, im Hexeditor selektierte Bereiche zu **disassemblieren**, und so Speicherinhalte besser interpretieren zu können.

Flexibilität

Befehle können bei Minimom entweder über das Menü, über die Symbolleiste oder in der Kommandozeile eingegeben werden. Oft hat man immer wiederkehrende Befehlsfolgen. Da es mühselig und fehleranfällig ist, immer wieder die gleichen Befehle einzugeben, bietet Minimom die Möglichkeit der Ausführung von **Scripts** an: In eine einfache Textdatei werden einfach alle Befehle mit den Parametern hineingeschrieben. Zusätzlich können Scripts auch (begrenzte oder unbegrenzte) Schleifen enthalten.

SIEMENS-3

Inserat

SIEMENS-4

Inserat

Der Monitorkernel ist so konzipiert, dass er nur minimal Speicher benötigt. Das **flexible Speicherkonzept** ermöglicht, dass der Monitorkernel nach dem Starten in jeden beliebigen anderen Bereich verschoben werden kann. Auch die Flash-Treiber sind frei lokatierbar, um hier von vornherein keine Einschränkungen zu machen und eventuell wo anders benötigte Bereiche frei zu halten.

Ein weiteres Feature von Minimon ist das **Initialisierungskonzept**. Beim Start des Monitors wird im Normalfall keinerlei Initialisierung vorgenommen. Um dennoch Register beim Verbindungsstart zu setzen oder bestimmte Befehle aufzurufen, kann deren Initialisierung frei konfiguriert werden. Zum Initialisierungskonzept gehört auch die Konfiguration aller in der Zielhardware vorhandenen Speicherbereiche. Diese können dann aktiviert/deaktiviert oder gemappt werden.

Alle **Einstellungen** werden von Minimon nach dem Beenden gespeichert, und nach dem erneuten Starten ist alles beim Alten. Um zusätzlich das Arbeiten an mehreren Projekten bzw. verschiedenen Zielsystemen zu ermöglichen, können Einstellungen in eigene Dateien gespeichert (Benutzerprofile) oder von dort geladen werden. Zudem werden alle Ein- und Ausgaben im Terminal/Statusfenster protokolliert.

Alle **Quellcodes** der Mikrocontrollerprogramme – Monitorkernel und Flash/OTP-Treiber – liegen bei. Außerdem sind Protokoll und Treiber-Schnittstelle dokumentiert. Dies erleichtert die Erstellung und Anpassung neuer Treiber.

Neue Controllertypen können ebenfalls einfach in Minimon integriert werden. Alle Mikrocontroller-spezifischen Eigenschaften sind in externen Textdateien gespeichert.

Anwendung

Wer mit MiniMon arbeitet, wird schnell feststellen, dass dieses Tool hinsichtlich Funktionalität die Antwort auf verschiedene Anforderungen ist, die sich aus dem täglichen Umgang mit Mikrocontrollern in der Anwendungstechnik ergeben. Mitnichten wird man also auf überflüssige Features stoßen, ebensowenig auf eine überladene Windows-Oberfläche. Beim strukturellen Aufbau wurde auf höchstmögliche Transparenz Wert gelegt, die neuen Usern den schnellen Einstieg in das Tool ermöglicht und erfahrenen Anwendern die Einbindung eigener Funktionen (z.B. Testprogramme oder Memory-Treiber) erleichtert.

Dank der Selbstinstallationsroutine (Setup) werden bei den ersten Schritten mit MiniMon Fehler weitestgehend ausgeschlossen; schließlich ist nichts ärgerlicher, als ein fehlerhaft installiertes Programm. Der Zielpfad kann natürlich frei gewählt werden. Die aktuelle MiniMon Version 2.1.16 beansprucht inklusive Systemdateien ca. 6 MB auf der Festplatte.

Danach wird die Hardware (z.B. C167 Starterkit) an eine der seriellen Schnittstellen des PC angeschlossen, und schon kann's losgehen. Bei der Schnittstellenwahl sollte darauf geachtet werden, dass diese exklusiv für Minimon zur Verfügung steht.

Configure

Initialize register

SYSCON h DPP0 h

BUSCON0 h DPP1 h

ADDRSEL1 h DPP2 h

BUSCON1 h DPP3 h

ADDRSEL2 h

BUSCON2 h

ADDRSEL3 h

BUSCON3 h

ADDRSEL4 h

BUSCON4 h

Generic 1 h h

Generic 2 h h

Addr.

Controller type

Clk rate Hz

Memory

A	M		
X		SFR	: 00FE00-00FFFF
X		ESFR	: 00F000-00F1FF
X		IRAM	: 00F600-00FDFF
		XRAM	: 00E000-00E7FF
		CAN	: 00EF00-00EFFF
X		FLASH	: 000000-03FFFF

Add Edit Remove

Initial command calls

EINIT

Clear Cancel OK

Nach dem Start von Minimon empfiehlt sich, einmalig die Hardwarekonfiguration vorzunehmen: Auswahl des Controller-Typs, eventuelle Initialisierungsregister setzen, Speicherbereiche festlegen. Standardmäßig ist der Typ C167CR und dessen interne Speicherbereiche konfiguriert, sodass man sich in vielen Fällen die Hardwarekonfiguration sparen kann. **Bild 3** zeigt das Hardware-Konfigurationsmenü:

Sodann stellen wir die Verbindung zwischen Frontend und Target her (CONNECT). Der Verbindungsaufbau zwischen Host (PC) und Target (z.B. Phytec C16x Starterkit) geschieht mit Hilfe des für die C16x Familie typischen internen Bootstrapladers (BSL).

Das Booten dauert nur 1-2 Sekunden, es wird der Chip-Identifizierer, die Resetkonfiguration angezeigt und alle Register-Initialisierungen vorgenommen. Zur Initialisierung von Special Function Registern (SFRs) des jeweils selektierten Controller-Derivats stützt sich MiniMon auf standardisierte SFR-Beschreibungsdateien (ASCII-Format), wie sie von Infineon auch an die offiziellen Toolpartner herausgegeben werden.

Als Alternative zum Booten kann der Minimon-Kernel auch in einem externen Speicher dauerhaft programmiert werden. Es erfolgt dann lediglich ein „RECONNECT“ zum Verbindungsaufbau.

MiniMon unterstützt zur Zeit Übertragungsraten von bis zu 28800 Baud, was für Laboranwendungen völlig ausreichend ist. An höheren Baudraten wird jedoch bereits gearbeitet.

Sobald die Verbindung besteht, kann mit der eigentlichen Arbeit begonnen werden. Im folgenden werden einige einfache Anwendungsbeispiele vorgestellt:

Upload

Es soll der Bereich von 0000-0FFF (die ersten 4 kByte) aus dem Targetspeicher geladen und in die Datei TEST.HEX gespeichert werden.

Der gewünschte Speicherbereich wird im Hexeditor markiert und mit dem Befehl UPLoad in das Frontend geladen. Dieser nun geladene Bereich kann dann entweder in eine Datei gespeichert oder disassembliert werden. Ein Script für diese Funktion könnte folgendermaßen aussehen:

```
connect          ; Verbindung herstellen
clearselections ; alle Selektionen entfernen
addselection 0000,0FFF ; Bereich 0000h-0FFFh markieren
_upload         ; Diesen Bereich aus dem
                ; Target-Speicher laden
_save test.hex  ; als Intel Hex Datei test.hex speichern
```

Wollen wir den Bereich disassemblieren, dann rufen wir anstatt dem Befehl Speichern den Befehl Disassemblieren (_disassemble) auf.

Download und Start

Oder wir wollen ein Programm in den Targetspeicher laden und starten. Zuerst wird die Datei geladen. Der geladene Bereich ist jetzt auch selektiert. Mit DOWNLOAD wird das Programm in das Ziel geladen (es muss sich um RAM handeln). Sodann kann das Programm angesprochen oder ein Software-Reset durchgeführt werden.

Der Script

```
connect          ; Verbindung herstellen
_load c:\test\test.hex ; Datei laden und im Hexeditor selektieren
_download        ; Selektion in den Target-Speicher laden
_srst            ; Software-Reset durchführen
```

Handelt es sich um Flash-Speicher, so muss anstelle des Downloads der Programmierbefehl (_iprogram) aufgerufen werden. Dieser löscht bei Bedarf auch alle zu beschreibenden Sektoren.

Hardwaretest

In diesem Beispiel wollen wir an einem Ein-/Ausgabeport des Mikrocontrollers abwechselnd Einsen und Nullen anlegen:

```
_mov DP8,FF ; Port 8 auf Ausgang stellen

DO          ; Schleife
  mov P8,FF ; Ausgänge auf 1 stellen
  _delay 5   ; 5 Sekunden auf 1 lassen
  mov P8,00 ; Ausgänge auf 0 stellen
  _pause     ; Auf Tastendruck warten
LOOP UNTIL 100 ; 100 Durchläufe

mov DP8,00 ; Port 8 auf Eingang stellen
_einit     ; den Befehl EINIT aufrufen
```

Subroutinen-Interface

Für Profis bietet sich außerdem ein ganz besonderes Feature: das Monitor-Interface, mit dem eigene Subroutinen (wie auch Flashtreiber) geladen und aufgerufen werden können.

Testprogramme werden als Subroutinen geschrieben, mit Minimon geladen und aufgerufen (Befehl _CALL). Über Register werden der Routine Parameter übergeben und zurückgegeben. Nach Beendigung der Routine (mit dem RET-Befehl) wird in den Monitor zurückgekehrt. Somit enthält die Subroutine wirklich nur alle für diesen Test spezifischen Befehle, das Booten, die Kommunikation und Darstellung der Ergebnisse wird von Minimon übernommen.

Bei der Anwendung von Minimon sind der Phantasie keine Grenzen gesetzt, und gerade durch die Skriptfähigkeit können viele Test-, Lade-, Programmier- oder Initialisierungsaufgaben einfach und durchschaubar durchgeführt werden. Minimon versteht sich daher durchaus als Gerüst für eigene Aufgaben und Speziallösungen.

Fazit

Minimon ist ein Tool, das es eigentlich schon lange hätte geben müssen. Es bietet für jede C16x-Plattform – speziell aber Single-Chip-Applikationen - ein breites Spektrum an Einsatzmöglichkeiten. Gerade die Vielfältigkeit und hohe Flexibilität macht es aber zu einem Profi-Tool, Anfänger verlieren sicher schnell den Überblick. Die umfangreiche Online-Hilfe und Referenz kann hier jedoch Abhilfe verschaffen.

Minimon ist jedoch kein Ersatz für herkömmliche „Monitor“-Programme. Diese haben als Hauptaufgabenbereich Software-Debugging, wogegen Minimon eher als Hardware-Tool zu klassifizieren ist. Das Programm lebt deswegen auch davon, dass auch die neuesten Typen rasch unterstützt werden und kontinuierlich auf dem laufenden gehalten wird. Deshalb wird Minimon auch ständig weiterentwickelt und gewartet. Es ist auch bereits vielfach bei C16x-Profis auf der ganzen Welt im Einsatz.

Minimon hat noch einen Vorteil: Es ist Freeware und kostet daher nix – außer ein paar Augenblicke, um es aus dem WWW zu laden. Für jeden C16x-Anwender ist daher Minimon ein Muss.

Minimon kann kostenlos heruntergeladen werden unter <http://stud4.tuwien.ac.at/~e9327470/minimon/minimon.htm>