

Entwicklungsschlüssel

vorgelegt in

Design&Elektronik 02/99

DESIGN & ELEKTRONIK
PRODUKTE UND KNOW-HOW FÜR DEN ELEKTRONIK-ENTWICKLER

Schlüsseltechnologie

In-System-Programmierung und In-System-Debugging von SX-Controllern mit dem SX-Key Entwicklungssystem

Seit der Markteinführung des SX-Controllers von Scenix im letzten Jahr hat die innovative Chipschmiede aus Santa Clara nochmals zugelegt: Der schnellste 8 Bit Mikrocontroller der Welt leistet nun 100 MIPS bei einer Taktfrequenz von 100 MHz. Die Produktion der 100 MHz-Version läuft ab 1. Quartal 1999 (die 50 MHz-Version ist schon länger verfügbar).

Möglich wird das durch eine modifizierte Harvard-Architektur und eine geschickt realisierte Pipeline, so daß eine Verarbeitungsgeschwindigkeit von einer Instruktion pro Taktzyklus erreicht wird. Das Programm wird dabei im internen Flashspeicher (2 K x 12 Bit, 10 ns Zugriffszeit) ausgeführt. Das Konzept der virtuellen Peripherie (VP) sorgt außerdem für eine optimale Flexibilität bei der Nutzung der Chip-Ressourcen. Da der Adreß- und Datenbus nicht extern herausgeführt wird, entspannt sich außerdem die EMV-Problematik beim PCB-Layout für den Designer beträchtlich /RWUR98/. Ausführlichere Informationen zum SX sind in /WUR98/ zu finden.

Mit dieser hohen Rechenleistung und Preisen im Bereich von 5\$ (bei mittleren Stückzahlen) eröffnen sich für diese 8 Bit Mikrocontroller neue und anspruchsvolle Anwendungsgebiete, die bisher ausschließlich von 16- und 32-Bit-Controllern, DSP's oder FPGA's dominiert wurden. Beispiele dafür sind Anwendungen wie digitale Signalverarbeitung (FSK, digitale Filter, Waveform Generatoren usw.), komplexe State-Machine-Verarbeitung sowie Motorsteuerungen, die keinen großen Arbeitsspeicher (RAM) benötigen.

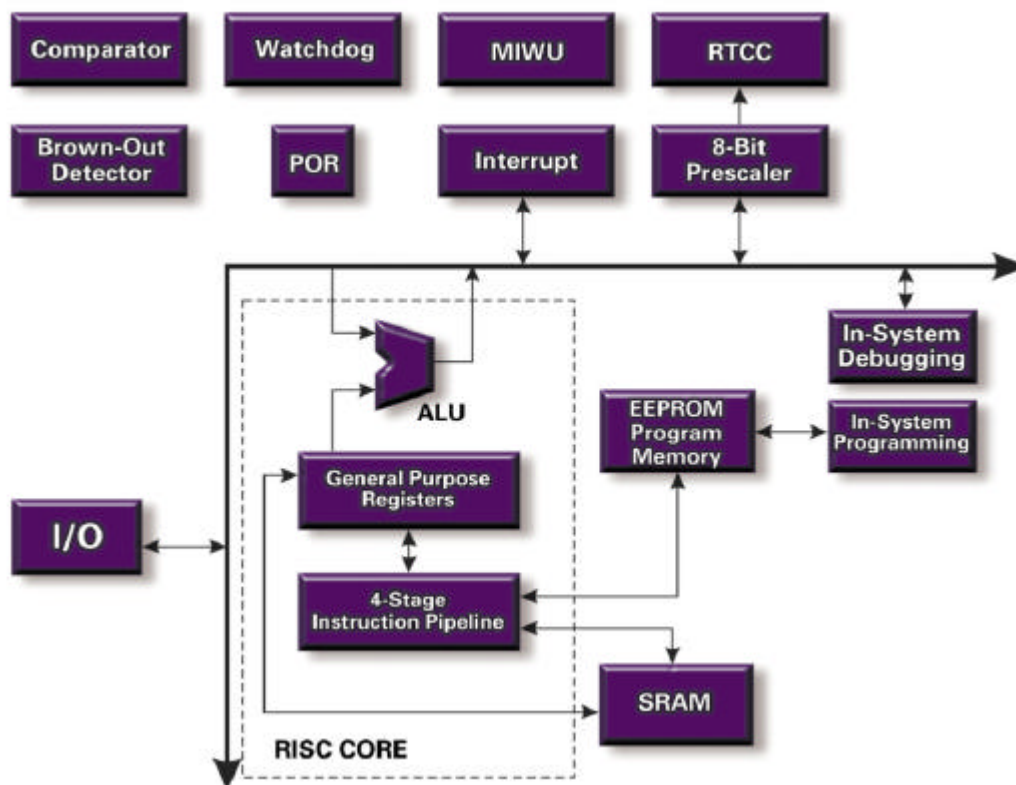


Bild 1: Blockschaftbild des SX (Scenix)

Um den Entwicklern sowohl den schnellen und kostengünstigen Einstieg in diese Technologie zu ermöglichen, als auch professionelle Lösungen im eigenen Produkt umsetzen zu können, bietet die Firma Parallax aus Rocklin - bekannt u.a. durch PIC-Produkte wie die BASIC-Stamp - ein leistungsfähiges Entwicklungswerkzeug für die SX-Controller an. Dabei werden sowohl in technischer Hinsicht als auch in Bezug auf das Preis-Leistungsverhältnis in

dieser Kategorie neue Maßstäbe gesetzt. Wird der Einsatz einer Hochsprache gewünscht, steht außerdem ein optimierender C-Compiler von ByteCraft zur Verfügung.

SX-Key - Der Schlüssel zur erfolgreichen Entwicklung

Das SX-Key Entwicklungssystem besteht im wesentlichen aus einer integrierten Entwicklungsumgebung, die unter Windows 95/98/NT läuft und einem SX-Key Programmier/Debugger-Kabel, das die Verbindung zum Zielsystem herstellt. Die integrierte Entwicklungsumgebung enthält u.a. einen Makroassembler und einen leistungsfähigen Debugger, der in der Funktionalität „richtigen“ Emulatoren gleicht. Allerdings wird dafür keine aufwendige Emulatorhardware benötigt: Das SX-Key Programmier/Debugger-Kabel (Bild 2) wird an die serielle Schnittstelle am PC angeschlossen und mit dem Controller im Zielsystem verbunden - das ist alles.

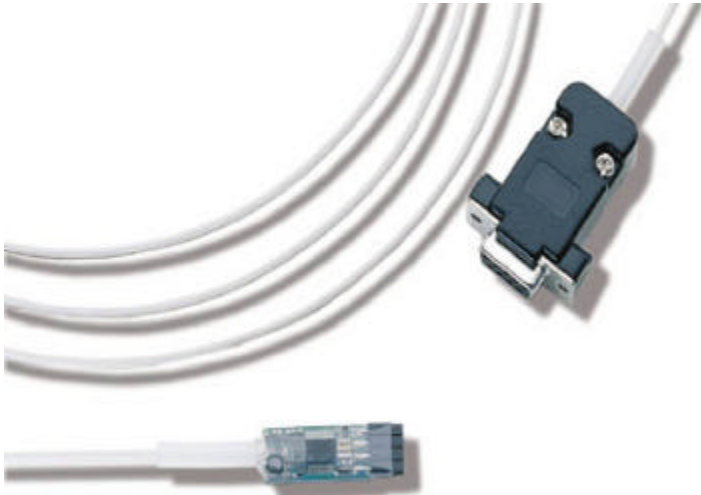


Bild 2: SX-Key Programmier/Debugger-Kabel (Parallax)

Mit dieser „intelligenten Nabelschnur“ wird die In-System-Programmierung und das In-System-Debugging durchgeführt. Kernstück ist hier ebenfalls ein SX, der für das Protokoll, die notwendigen Spannungen und das richtige Timing am zu programmierenden Controller sorgt. Wie im Bild 2 zu erkennen ist, werden nur 4 Signale über einen Pfostensteckverbinder am Zielsystem angeschlossen:

- OSC1
- OSC2
- VDD (Betriebsspannung vom Zielsystem, versorgt auch den SX-Key)
- VSS (Masse)

Die Programmierung und das Debugging werden über 2 Signale durchgeführt: OSC1 und OSC2. Dem Konzept liegt folgende außergewöhnliche Idee zugrunde: Im Programmier/Debugger-Mode wird der externe Taktgenerator (oder Quarz, RC, Keramikresonator) vom Controller abgetrennt und dafür wird der SX-Key an die Oszillatorpins des Controllers angeschlossen. Dieses Prinzip hat gleich mehrere Vorteile:

- Es werden keine zusätzlichen Pins für das Programmieren/Debugging benötigt
- Der SX-Key bestimmt allein das Taktverhalten des Controllers
- Während des Debuggens kann die Taktfrequenz in weiten Bereichen variiert werden
- Single-step-Betrieb und Breakpoints werden unterstützt
- Der Controller zeigt beim Debugging das gleiche (elektrische) Verhalten im Zielsystem

Für den schnellen Einstieg ist optional ein SX-Key Demo-Board (Bild3) oder ein SX QuickProto Board erhältlich. Das SX-Key Demo-Board enthält u.a. Taster, seriellen EEPROM, analoge Ein- und Ausgänge, Piezo-Signalgeber, LED und RS232-Schnittstelle. Zur sofortigen Inbetriebnahme gibt es Softwarebeispiele, die den Einsatz des SX-Controllers und die Nutzung der Peripheriefunktionen demonstrieren.

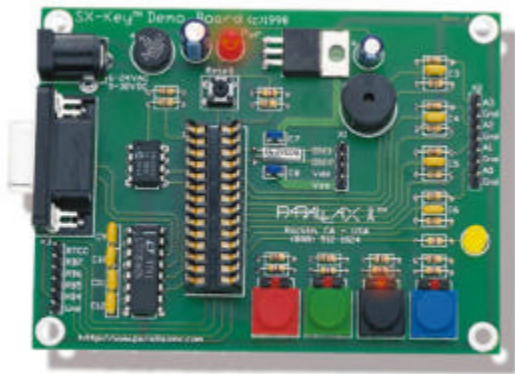


Bild 3: SX-Key Demo-Board (Parallax)

SX-Key Entwicklungsumgebung

Die Installation der SX-Key Entwicklungsumgebung ist denkbar einfach: Die Software wird von der (einzigen!) Diskette in ein gewünschtes Verzeichnis auf den PC kopiert - fertig! Erfreulicherweise erfolgt die Installation ohne Veränderung der Systemdateien im Windows-Verzeichnis...

An Hand des mitgelieferten Beispielprogramms SXDEMO.SRC soll die Arbeitsweise mit der SX-Key Entwicklungsumgebung erläutert werden.

Zuerst ist das SX-Key Programmier/Debugger-Kabel an das Zielsystem (SX-Key Demo-Board) und an den PC anzuschließen und mit Spannung zu versorgen. Dabei muß der Jumper, der den Keramikresonator mit dem Oszillatortreiber des SX-Controllers verbindet, offen bleiben.

Nach dem Programmstart kann sofort mit der Bearbeitung des Quellcodes begonnen werden. Für erste Tests mit dem SX-Key Demo-Board empfiehlt es sich, das Beispielprogramm SXDEMO.SRC in den Editor zu laden und mit RUN/ASSEMBLE zu übersetzen. Mit FILE/LIST TOGGLE kann nach erfolgreichem Assemblerlauf zwischen Quellcode und Programmlisting umgeschaltet werden.

Danach ist die Einstellung der SX-Key Parameter im Menü Configure (Bild 4) vorzunehmen. Neben der Wahl der richtigen seriellen Schnittstelle kann hier noch die Kalibrierung des internen RC-Oszillators vorgenommen werden: Für sehr kostensensitive Applikationen mit unkritischen Timinganforderungen kann auf externe Taktquellen verzichtet werden und ein interner RC-Oszillator stellt den Systemtakt bereit. Um Parameterstreuungen der Bauelemente auszugleichen, kann dieser Takt während der Programmierung auf 4 MHz getrimmt werden. Das Ergebnis (nach der Programmierung) ist im Bild 5 dargestellt.

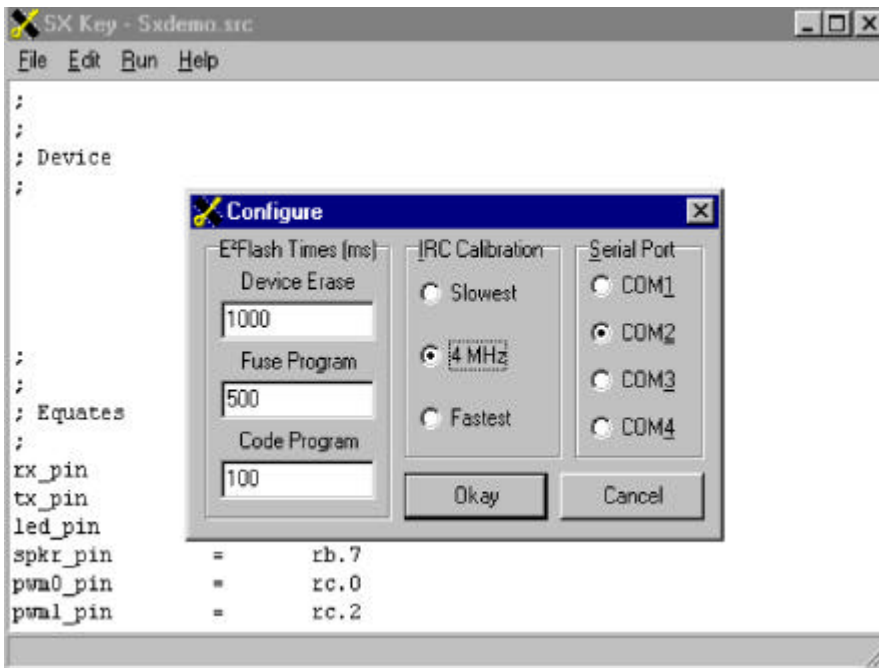


Bild 4: Menü Configure

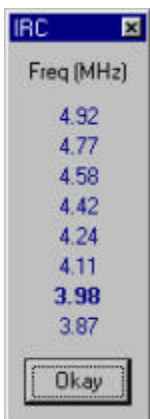


Bild 5: Kalibrierung des internen RC-Oszillators nach der Programmierung

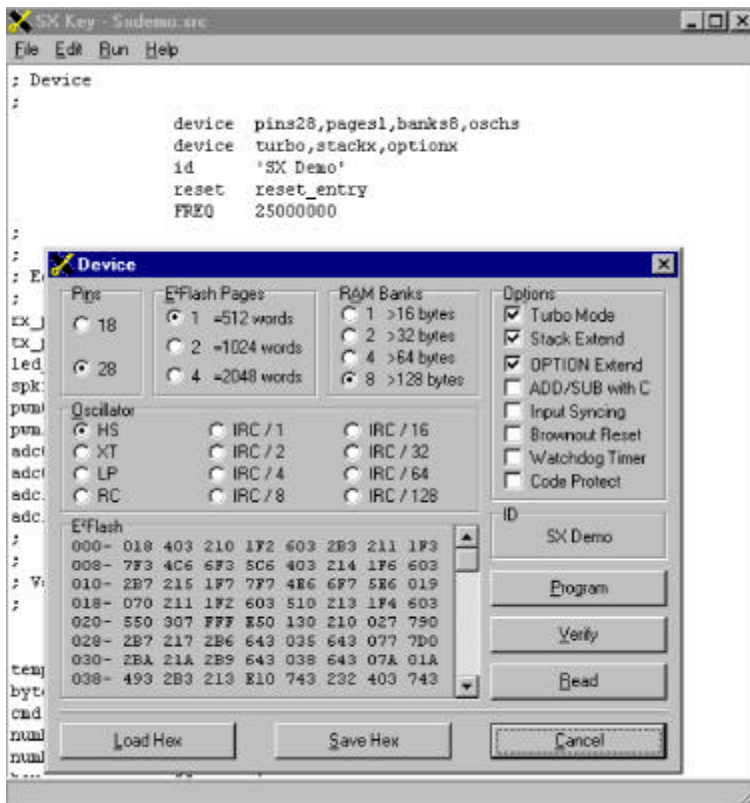


Bild 6: Konfiguration des SX-Controllers im Zielsystem

Im Menü Device (Bild 6) ist nun die passende Konfiguration des SX-Controllers einzustellen. Das betrifft insbesondere folgende Punkte:

- Anzahl der Pins des verwendeten SX-Controllers
- Größe von RAM und Flashspeicher
- Wahl der Oszillatorbeschriftung
- Fuses für spezielle Einstellungen (z.B. Watchdog, Resetcontroller-Schwelle)

Alternativ können diese Einstellungen auch im Quellcode über die DEVICE-Direktive vorgenommen werden und sind damit immer für das Projekt gültig. Außerdem besteht in diesem Menü noch die Möglichkeit, Programme als Hex-File zu laden, abzuspeichern und zu verifizieren.

Bevor der SX-Controller programmiert wird, ist mit RUN/CLOCK noch die Taktfrequenz einzustellen, die vom SX-Key geliefert wird (Bild7). Dabei ist zu beachten, daß die Spezifikation des SX-Controllers im Zielsystem eingehalten wird. (Im vorliegenden Fall ließ sich zwar ein mit 50 MHz spezifizierter SX testweise kurzzeitig mit bis zu 95 MHz betreiben, dies führt aber - wie zu erwarten war - zu erhöhter Wärmeentwicklung und Stabilitätsproblemen und ist natürlich nicht zu empfehlen...) Die Taktfrequenz kann auch während des Betriebes (RUN) verändert oder ein- und ausgeschaltet werden.

Nun kann der SX-Controller mit RUN/PROGRAM programmiert oder mit RUN/RUN erst programmiert und gleich gestartet werden. Nebenbei eröffnet diese einfach zu handhabende Technik der In-System-Programmierung die Möglichkeit des Komponententests auf der Leiterplatte ähnlich dem Boundary-Scan-Testverfahren, da außer dem SX-Key und dem SX-Controller keine weiteren Komponenten zur Inbetriebnahme notwendig sind.



Bild 7: Einstellung der Taktfrequenz

Debugger

Sollte die Applikation nicht sofort das gewünschte Verhalten zeigen, so ist ein Debugger ein sehr willkommenes Werkzeug. Gerade hier liegen die Stärken des SX-Key Entwicklungssystems.

Zum Betrieb des Debuggers sind vorher einige Vorbereitungen vorzunehmen:

- Der Watchdog-Timer muß ausgeschaltet sein
- Mit der RESET-Direktive muß der Programmstartpunkt angegeben sein
- Mit der FREQ-Direktive muß die Taktfrequenz angegeben sein
- 136 Worte freier Programmspeicher muß vorhanden sein

Mit der BREAK-Direktive kann ein Breakpoint für den Debugger an einer beliebigen Stelle im Quellcode plaziert werden. Sehr hilfreich ist auch die WATCH-Direktive, mit der Variablen im Speicher in einer gut lesbaren Form angezeigt werden können. Im folgenden Beispiel bewirkt die Anwendung der WATCH-Direktive die Anzeige der zwei deklarierten 8-Bit Variablen `freq_acch` und `freq_accl` als eine vorzeichenlose 16-Bit Dezimalzahl:

```
...
freq_acch  ds    1
freq_accl  ds    1
           watch  freq_acch,16,udec
...
```

Es sind dabei folgende Anzeigeformate möglich:

Format	Anzeigeformat
UDEC	vorzeichenlose Dezimalzahl
SDEC	Dezimalzahl mit Vorzeichen
UHEX	vorzeichenlose Zahl in Hexadezimaldarstellung
SHEX	Zahl mit Vorzeichen in Hexadezimaldarstellung
UBIN	vorzeichenlose Zahl in Binärdarstellung
SBIN	Zahl mit Vorzeichen in Binärdarstellung
FSTR	Zeichenketten mit fester Länge
ZSTR	Zeichenketten mit variabler Länge, Null-terminiert

Zum Start des Debuggers wird mit RUN/DEBUG der SX-Controller programmiert und anschließend in den Debug-Mode gebracht. Es erscheinen 4 verschiedene Fenster (Bild 8):

Register-Fenster

Es werden die Prozessorregister, die RAM-Bänke und ein Ausschnitt des Programms in Hexadezimaldarstellung (12 Bit) und als Assemblerlisting angezeigt. Sollen während des Debuggens Registerwerte verändert werden, so kann man einfach mit der Maus auf das gewünschte Register klicken und einen neuen Wert eintragen. Die aktuelle Assemblerinstruktion wird jeweils blau hervorgehoben. Die aktuelle RAM-Bank, die über das File Select Register ausgewählt ist, wird heller als die anderen dargestellt. Registerinhalte, die sich seit der vorherigen Instruktion geändert haben, werden an den betreffenden Stellen rot markiert. Ein Breakpoint im Programm wird ebenfalls rot hervorgehoben. Das Setzen und Löschen eines Breakpoints erfolgt einfach durch Klicken mit der Maus auf die betreffende Programmzeile

Code-Fenster

Im Code-Fenster wird der Quellcode angezeigt. Die Markierung von Breakpoint und der aktuellen Assemblerinstruktion erfolgt genauso wie im Register-Fenster. Im Bild 8 fällt auf, daß zwischen dem Breakpoint und der aktuellen Assemblerinstruktion im Code-Fenster keine weiteren Instruktionen liegen. Im Register-Fenster sind jedoch 3 Assemblerinstruktionen an dieser Stelle eingefügt. Das liegt daran, daß die Syntax des Parallax-Assemblers Makros kennt, die dann in mehrere Assemblerinstruktionen aufgelöst werden. Deshalb muß man hier zwischen Quellcode und Assemblercode unterscheiden, obwohl dies bei „normalen“ Assemblern unüblich ist.

Watch-Fenster

Im Watch-Fenster werden die mit der WATCH-Direktive versehenen Variablen im gewählten Format angezeigt und können wie im Register-Fenster auch verändert werden. Im Bild 8 wird z.B. die Variable `freq_acch` im Watch-Fenster als Dezimalzahl und Binärzahl jeweils mit und ohne Vorzeichen dargestellt.

Debug-Fenster

Über das Debug-Fenster erfolgt die Bedienung des Debuggers und es wird der aktuelle Status des SX-Controllers angezeigt. Es stehen folgende Befehle zu Verfügung:

Befehl	Bedeutung
Hop	führt eine Quellcode-Instruktion aus (siehe Code-Fenster)
Jog	führt fortlaufend Quellcode-Instruktionen aus (siehe Code-Fenster) und aktualisiert bei jedem Schritt die Anzeige in den Fenstern des Debuggers
Step	führt eine Assembler-Instruktion aus (siehe Register-Fenster)
Walk	führt fortlaufend Assembler-Instruktionen aus (siehe Register-Fenster) und aktualisiert bei jedem Schritt die Anzeige in den Fenstern des Debuggers
Run	startet das Programm in Echtzeit bis ein Breakpoint erreicht wird oder mit Stop die Programmausführung angehalten wird (ohne laufende Aktualisierung der Fensterinhalte)
Poll	wenn ein Breakpoint existiert, wird ein gestartetes Programm kurz angehalten, es erfolgt eine Aktualisierung der Fensterinhalte des Debuggers und das Programm läuft weiter
Stop	stoppt die Programmabarbeitung
Reset	löst einen Reset am SX-Controller aus

Mit diesen effizienten Befehlen wird die Fehlersuche sehr erleichtert, besonders die Befehle Jog und Walk erlauben eine einfache visuelle Verfolgung des Programmablaufes, da das Programm sozusagen als „Film“ im Debugger abläuft. Während des Debuggings bleibt auch der RTCC (Real-Time-Counter) aktiv, so daß auch das Verhalten von Interruptroutinen (z.B. VP) getestet werden können. Im Bild 8 wird dies im Register-Fenster durch das blau hervorgehobene INT-Symbol angezeigt.

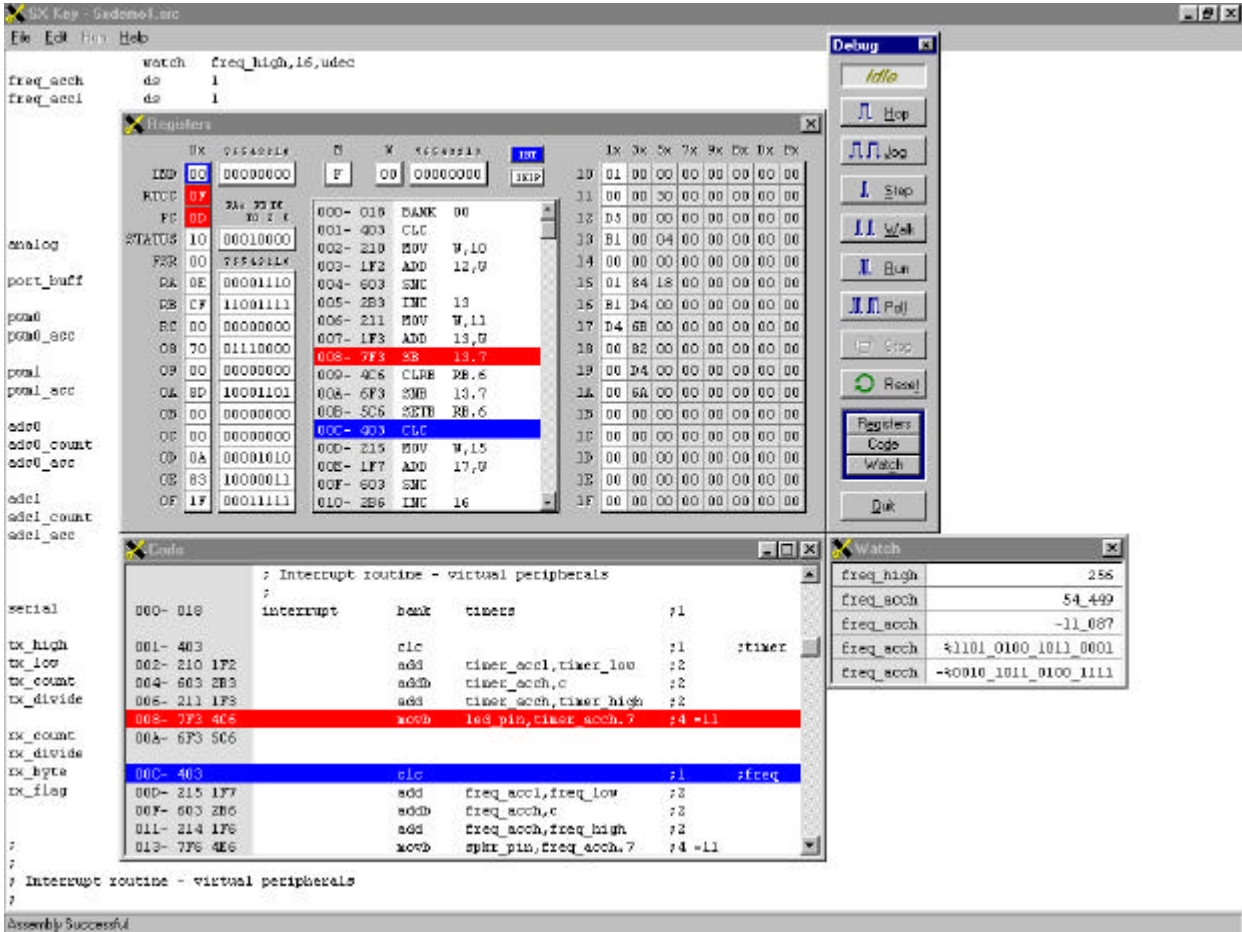


Bild 8: Debugger

Ausblick

Parallax arbeitet kontinuierlich an der Weiterentwicklung des SX-Key Entwicklungssystems. Verbesserungen sind im Hinblick auf eine verstärkte Hochsprachenunterstützung zu erwarten. Neue Programmversionen der SX-Key Software, Hinweise und Errata-Sheets sind im Web bei Parallax kostenlos erhältlich.

Auf der anderen Seite wird das Angebot der virtuellen Peripheriekomponenten weiter anwachsen. Chiphersteller Scenix bietet ebenfalls im Web kostenlose Designtips, Applikationshinweise und Softwaremodule an. Nachstehende Übersicht zeigt die von Scenix verfügbare Software (VP), ohne dabei den Anspruch auf Vollständigkeit zu erheben:

Sinus-Generator über PWM-Ausgang
RS-232 High-speed UART
8 Bit Pulsweitenmodulation
8 Bit Sigma-Delta-ADC
16-Bit Timer
I ² C Master Interface
I ² C Slave Interface
SPI/μWire High-speed Master Interface
SPI/μWire High-speed Slave Interface
Stepper Motor Control
4x4 Keyboard Scan
FSK Generation
DTMF Generation
Real-Time Clock
8-Bit LCD Interface
4-Bit LCD Interface
LED Bar Graph Display
Math Pack (16-32 Bit Integerarithmetik)
FIR-Filter

Außer dem bereits oben beschriebenen SX-Key Demo-Board und dem SX QuickProto Board von Parallax gibt es auch erste Produkte von Third-Party-Herstellern. Die Elektronikladen Mikrocomputer GmbH stellt auf der Embedded Systems Anfang März das Controllermodul „SX HeadStart“ für Prototypenentwicklung und Kleinserien vor. Das Board im Format 80 mm x 100 mm, das für den Einsatz in Steuerungen konzipiert ist, verfügt u.a. über einen SPI-Bus mit 12 Bit ADC (11 Kanäle), RTC, 1k x 8 Bit EEPROM, Analogschalter, Reset-Controller, 2 serielle RS-232 Schnittstellen, 2 RC-PWM Ausgänge sowie Anschluß für LC-Displays.

Aktuelle Informationen zum Thema SX und SX-Key sind auch unter <http://www.elektronikladen.de/sx.html> zu finden.

Literatur

/WUR98/ Wurlitzer, Th.: μ Controller mit virtueller Peripherie, Design&Elektronik, H. 6/98 S.32-34.
/RWUR98/ Reinhold,W.; Wurlitzer, Th.: Sigma-Delta-ADC in Software, Design&Elektronik,
H. 10/98 S.90-99.

Internetlinks:

www.parallaxinc.com
www.scenix.com
www.elektronikladen.de

Autorenkontakt:

Dipl.-Ing. Thomas Wurlitzer
c/o MCT Elektronikladen GbR
Hohe Str. 9-13
04107 Leipzig

eMail: tw@elektronikladen.de
Fax: 0341-2118355

Vertrieb SX-Key in Deutschland:

Elektronikladen Mikrocomputer GmbH
Tel.: 05232-8171
Fax: 05232-86197