

Z8-Entwicklungstools

Nach der Vorstellung des Z8-Emulatorboards im "ELVjournal" 3/99 geben wir eine Übersicht über die umfangreiche und komfortable Entwicklungsumgebung für diese Mikrocontroller-Reihe.

Das Developer Studio

Hat man von der dem Emulator-Kit beiliegenden CD-ROM "Development Software" (für Windows 95/98/NT) aus das komplette Setup ausführen lassen, finden sich in der Programmgruppe "ZILOG" das ZILOG Developer Studio (ZDS), das DSPICE- und das ZPROG-Tool, wie bereits im "ELVjournal" 3/99" beschrieben.

Das GUI ist von einer getrennt beiliegenden Diskette zu installieren, ebenso bei Bedarf der Assembler ZMASM.

Schwerpunkt unserer Softwarevorstellung soll das ZDS bilden, da dieses wohl auch das meistbenutzte Tool des Softwarepaketes sein wird.

Vor allem aus Platzgründen beschränken wir uns jedoch auf die globale Beschreibung des Funktionsumfangs, die "Bedienungsanleitung" des ZDS allein würde den Rahmen dieses Artikels sprengen, zumal die Programmierarbeit denn doch schon

ZiLOG Developer Studio - [Seriell]					
<u>🛛 F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject <u>B</u> uild	<u>T</u> ools <u>W</u> indow <u>H</u> elp			<u>_ 8 ×</u>	
D 🛩 🖬 X h C # 4	? 🛛 🕸 🖀 🎽 📳		🗒 🧶 🕒 (ř 47 (ř 🔍 🗐	□ 🖙 📗 🎖 🔶 📚 🛠 💥	
X		jr	nz, ISR_4_SEND	; nicht 0> IRET	
📾 PC-10	;** Senden bee	nden			
- Source Files		cp	_ANZAHL,_S_ZEIGER	; Zeiger > Anzahl ?	
		jr	nc,ISR_4_S_N	; nein> nächstes By	
			TWD #4 TO STOP	minute states	
SerieiLasm		IU	G FLAC 4. F SFNDFN	; Timer Scop	
trigger.asm		ir	_S_FERG, #"F_SENDEN	, Senderrag roschen	
📄 Befehl.asm	:** nächstes B	vte	INC LOUGH		
🚊 🗠 😋 Dependencies	ISR 4 S N	ld	S DATA,0 S ZEIGER	; Byte in Data	
📄 konstant.h	•	cp	S ZEIGER, ANZAHL	; letztes zeichen ?	
🔲 seriell.h		jr	z,ISR_4_S_N3	; ja> senden	
B triager h					
		cp	_S_DATA,#SOH	; SOH ?	
bereni.n		jr	z,ISR_4_S_N1	; ja> Sonderbehandlı	
		cp	_S_DATA, #DLE	; DLE ?	
		jr	Z, 15R_4_5_N1	; ja> Sonderbehandli	
		ép ár	_S_DATA,#EOI	; EOI /	
	:** Sonderbyte	j⊷ s ver⊮a	lten	, nein , senaen	
FileView			0 DI 10 //P DI P		
≚trigger.o - 0 error(s)	, 0 warning(s)				
Befehlasm	0 monoing(a)				
Linking	0 warning(s)				
PC-IO.hex - 0 error(s)	, 0 warning(s)				
				-	
Build Debug A Find A OTP					
Ready			📃 🛒 Ln	53, Col 1	

Bild 1: So präsentiert sich der ZDS mit seiner komfortablen Oberfläche



mehr als nur einige Grundkenntnisse vor-

aussetzt. Startet man das ZDS, erscheint in zunächst 3 Fenstern eine äußerst komfortable Oberfläche, die starke Ähnlichkeiten zu

	;** nächstes	Byte	
	ISR_4_S_N	ld	_S_DATA,0_S_ZEIGER
•		cp	_S_ZEIGER,_ANZAHL
		jr	z,ISR_4_S_N3

Microsofts "Visual Studio" aufweist. Programmierer werden dieses Erscheinungsbild begrüßen, wird ihnen doch die Umstellung nicht schwerfallen.

Das linke der drei Hauptfenster verwaltet die Projektliste.

Hier sind übersichtlich die eigentlichen Projektdateien und die sogenannten Dependencies (Abhängigkeiten) des aktuellen Projekts aufgelistet, so daß man besonders bei mehreren Teilprojekten und längeren Programmen mit vielen Dependencies eigentlich kaum einmal die Übersicht verlieren kann.

Das rechte (größte) Fenster ist das Editorfenster, hier erscheint das Quellfile.

Schließlich finden wir unten das Resultfenster, in dem die Auswertungen z. B. von Compiler-Läufen, Debugger-Läufen usw. erscheinen.

Abbildung 1 zeigt die gesamte Bedienoberfläche mit einem Beispielprojekt.

Windows-üblich erreicht man über die Menüzeile alle Menüs und dazu direkt über die darunterliegende Symbolzeile zahlreiche oft benötigte Funktionen, was die flinke Arbeit mit dem Programm sehr erleichtert. Mit dem Datei-Fenster (File) eröffnen sich die üblichen Möglichkeiten zum Öffnen, Speichern, Neuerstellen, Drucken und Verwalten von Projekten.

Überdas Edit-Menü sind sowohl allgemein übliche Editiertätigkeiten am Quelltext wie Undo, Cut, Copy, Paste usw. als auch Such-und Ersetzungsfunktionen, Sprünge zu einer bestimmten Programmzeile und schließlich das Setzen von Breakpoints über einen mehrstufigen Dialog möglich.

In Abbildung 2 wird u. a. dieser Dialogweg angezeigt.

Wer das "MS Visual Studio" bereits

kennt, wird hier wohl sofort versuchen, den Breakpoint auf die dort übliche Weise in die Laufleiste links des Editorfensters zu setzen, was sich auch als möglich erweist (Abbildung 3).

> Bild 3: Breakpoints sind auch bequem direkt setzbar Set Breakpoint"-

Da erscheint auch der "Set Breakpoint"-Button in der Symbolleiste fast schon überflüssig. Das View-Menü erlaubt die Gestaltung der Arbeitsoberfläche, also die Auswahl, welche Fenster aktuell geöffnet sein sollen.

Einzige Besonderheit ist hier die Refresh-Möglichkeit, um z. B. die Anzeige im Projektfenster nach Änderungen, etwa dem Erstellen neuer Dependencies, zu aktualiseren.

Der nächste Menüpunkt, das Projekt-Menü, macht zunächst die individuelle Einstellung von Assembler, Linker, Editor, Debugger und OTP-Link möglich.

Des weiteren wird hier die Möglichkeit geboten, Projektteile zu ergänzen, also nachzuladen, oder auf einfache Weise zu entfernen.

Schließlich erfolgt hier über "Target" die Auswahl des zu "behandelnden" Prozessors und des angeschlossenen Emulatorboards aus einer umfangreichen Bibliothek, wahlweise geordnet nach Applikationen oder Prozessorfamilien.

Daneben sind auch die Daten des gewählten Prozessortyps aufrufbar, so daß man sich Nachschlagen nach Daten weitgehend ersparen kann.

Abbildung 4 vermittelt einen Eindruck vom Angebot der Database.

Die Option "Emulator Configuration" erlaubt die Einstellung der Betriebsart des angeschlossenen Boards (bei Zilog "ICE-BOX" genannt), die Auswahl der seriellen Schnittstelle und der Datenübertragungsrate.

Im Menü "Build" findet man alle zum Assemblieren/Compilieren, Aktualisieren und Debuggen wichtigen Funktionsaufrufe (Abbildung 5).

Wichtig ist hier auch der Connect-Aufruf, also das Herstellen der Verbindung

ZiLOG MCU Database	×	
Selection by	ОК	
Family C Application	Cancel	
Master Selection:	Chip Data	
128	Project Build Tools Window	<u>H</u> elp
Project Target: Emulator: Z86E02 Z86C1200ZEM	<u>T</u> arget <u>E</u> mulator Configuration	
Zilog Microcontroller Database	Add to Project	beenden
General Properties IceBoxes Chip Registers Other F	Remove from Project DEL	cp
	<u>S</u> ettings	Assembler
Chip Name: Z86E02		<u>L</u> inker
Description: Standard Z8		<u>E</u> ditor
		<u>D</u> ebugger
		<u>0</u> TP
Address Spaces Start Length BELLE 0 64		
XDATA 0 0		
ROM 0 512	Bild 4:	
	Hier ist die A	uswahl des
	Prozessors l	bequem möglich,
OK Abbrechen	Übernehmen ale Bibliothe wichtigen Da	k zeigt auch alle
	wichtigen Da	iten auf einen Blick.

- [Reaction.s]			
<u>Build T</u> ools <u>W</u> indow	<u>H</u> elp		
Assembly/ <u>C</u> ompile	Ctrl+F7		- 7 4
<u>B</u> uild	F7		
<u>R</u> ebuild All	Alt+F7		
Update All Dependenc	cies		
ICo <u>n</u> nect			
Start <u>D</u> ebug	۱.	<u>G</u> o	
		Step Into	

Bild 5:

Im Menü "Build" findet man alle zum Assemblieren/Compilieren, Aktualisieren und Debuggen wichtigen Funktionsstufen.

zum Emulator-Board nach vorheriger Konfiguration.

Ist die Verbindung zum Emulator-Board hergestellt, kann der Code über die Option "Start Debug" von hier aus in den Emulator heruntergeladen werden.

Schließlich bietet das Menü "Tools" die Möglichkeit der Einstellungen für die OTP-Programmierung, eines Firmware-Upgrades des Emulator-Boards und der individuellen Konfiguration der Toolbar.

Bleiben noch die Menüs "Window" und "Help", die im ersten Fall die Windowsüblichen Anordnungen aller geöffneten Fenster (Projekte) und deren Verwaltung und zum zweiten die sehr umfangreiche Online-Hilfe zum gesamten Programmsystem einschließlich Hilfe für den Kontakt zu Zilog anbieten.

Die Anleitung zum Makro-Assembler selbst findet man im übrigen auch als gedrucktes, ausführliches Handbuch im Emulator-Kit vor.

ZILOG ICEBOX : Z8CCP00ZEM : z8em_CCP.bsc : COM2:19200

<u>F</u>ile ⊻iew <u>0</u>TP! <u>E</u>dit <u>W</u>indow <u>H</u>elp

Tracked Address:

7 Z8 Re

Abschließend zur Erläuterung der Bedienoberfläche des ZDS sei nochmals auf die komfortable Möglichkeit hingewiesen, über die Toolbar zahlreiche Befehle, z. B. zum Debuggen, direkt über die entsprechenden Buttons zu erreichen.

Auch der Service des Ergebnisfensters, hier alle Ergebnisse der verschiedenen Test- und Übersetzungsläufe gewissermaßen zwischenspeichern und nacheinander abrufen zu können, soll nicht

unerwähnt bleiben.

Run to Cursor

Download Code

Reset

Reset+Go

Damit ist die Beschreibung der komfortablen Oberfläche des ZDS bereits abgeschlossen.

GUI und Z 8 ICEBOX

Hat man sich mit dem ZDS schon einmal beschäftigt, wird man schnell feststellen, daß man des GUIs für den Kontakt zwischen dem eigentlichen Makroassembler ZMASM und dem Emulatorboard kaum bedarf.

GUI und ICEBOX bilden zwar zusammen mit dem ZDS eine Software-Einheit, können jedoch auch separat betrieben werden. Sie sind quasi der "Vorläufer" des

<u>- 🗆 ×</u>

ein Debug-Werkzeug, das es erlaubt, maschinennah mit dem Prozessor zu kommunizieren.

Auch hier ist es möglich, Assembler-Files zu laden, zu editieren, zu speichern und zum Emulatorboard herunterzuladen, den Status des Prozessors (Register, Flags etc.) zu ermitteln und zu verändern, Assemblerfiles zu debuggen usw.

Im Konfigurationsfenster des GUIs (Abbildung 7) kann man ähnlich wie im "Target-Menü" den Prozessortyp konfigurieren.

Hier hat man, das ist der entscheidende Vorteil dieses Werkzeugs, die Möglichkeit, auch Prozessoren zu konfigurieren, die

onfiguration		×		
Expanded Register Ban	ks D3 □ 04 □ 05 □ 06 □	07		
	DB 🗖 OC 🗖 OD 🗖 OE 🗖	OF		
Selected Ports				
Port0 Port1	Port2 Port3	3		
Port4 Port5 Port6				
- MicroController	70 DOM C:			
Z86C04/F04				
	O.5K ⊙1K O2	2К		
Emulation Mode	O4K O8K O1	16K		
C Z8 Family MCU	O 24K O 32K O 0	64K		
C Digital P3 C Analog P3 Actual ROM Size: 1K				
© CCP Family MCU © SPI Emulation	OSP ROM Size	ВК		
	OK Can	cel		

Bild 7: Die Prozessor-Konfigurationsbox des GUI

ZDS und je nach Vorliebe des Benutzers statt diesem einsetzbar.

CEBOX (Abbildung 6) ist

sich eventuell noch nicht in der mitgelieferten Konfigurationsbibliothek befinden.

00 01 02 03 04 05 06 07 06 03 0A 05 0C 0D 0C 07 0- 0 00 00 00 00 00 00 00 00 00 00 00 00 0	OTP Programming Options Reserved Reserved Reserved	O4 Topmark:	Standard BLANK CHECK VERIFY	ren" Instrument da dieser doch r blere Möglichke blieren, Compi gen bietet.	, dem ZDS greifen nehr und komforta eiten für das Assem lieren und Debug
10 FF	CheckSum	Reserved OTP <u>C</u> HECKSU	Select an ICEBOX	K: The ICEBC	IX is connected to:
ZET EPK 000C NOP	Command Status Device Serialization Method C Sequential C Pseudorandom © None	RAM CHECKSU Serial Number Size C 1-Byte C 2-Byte Serial Number Serial Number C Dec	286144002EM 286C12002EM 286C15002EM 286C27002EM 286C50002EM 286C62002EM 286C84002EM 286C93002EM 286C93002EM 286C19002EM 286K15002EM 286L71002EM	Communic COM1 COM3 Baud <u>Baud</u> C 28800	etion <u>Port</u> © COM2 © COM4 © 19200 © 57600
Bild 6: Der kleine Bruder" de	286E04 1850IC: Use 286E04 20550P: Use	286E0700ZDP 286E0800ZDH	286018002EM 287022002EM 289175002EM 289C65002EM 289C67002EM		