

Andreas Zilker, Stefan Wurdack

Z80-EMUF mit Komfort

Der Z80-EMUF ist ein kompletter Mikrocomputer und eigentlich zu schade, um nur zu Steuerungsaufgaben eingesetzt zu werden. Deshalb soll in diesem Artikel eine Möglichkeit vorgestellt werden, den EMUF mit einer einfachen Tastatur und einer Anzeige auszurüsten und ihn durch die entsprechende Software zum Leben zu erwecken. So ausgerüstet eignet er sich besonders für Entwicklungs- und Meßaufgaben oder einfach als Lehrsystem für diejenigen, die ein erstes Mal ernsthaften Kontakt zu einem Mikrocomputer suchen.

Um die CPU nicht mit der Kontrolle des Displays und der Tastatur zu belasten, wurden zwei „intelligente“ Peripherietreiber verwendet. Der 7218 von Intersil steuert das 8stellige LED-Display (Typ 7218 A mit gem. Anode, B mit gemeinsamer Katode) und der 74C923 von National Semiconductor die Tastatur. Beide Bausteine werden über PIO 2 von der CPU mit Daten (PIO2A) und Steuersignalen (PIO2B) versorgt. Ein direkter Anschluß an den CPU-Bus, der prinzipiell möglich wäre, wurde nicht vorgenommen, damit die Peripherie z. B. auch für den 6504-EMUF ohne Hardwareänderungen verwendbar ist.

Der Aufbau der Hardware

Der Aufbau auf eine Einfach-Europakarte in Fädertechnik ist problemlos möglich (*Bild 1*), wobei nicht mit Entkopplungskondensatoren gespart werden sollte (Multiplex-Störungen des Anzeigentreibers). Beide ICs, speziell der Anzeigentreiber, sind angesichts der gepflegten Preise mit großer Vorsicht zu behandeln. Für die Tastatur wurden Digitast-Mini-Keys verwendet; der 74C923 verdaut sicher auch einfachere Tasten. Außerdem ist noch Platz auf der Platine für ein Kassettenrecorder-Interface [1] und eine einfache V.24-Schnittstelle (*Bild 2*). Der Anschluß an die CPU-Karte erfolgt mit einem 26poligen Flachbandkabel mit Pfostenstecker. Dazu wurde der „intelligenterweise“ nur 20polige PIO-An-

schluß der EMUF-Platine ins Lochrasterfeld herübergezogen und auf 26 Pole erweitert (zusätzlich Versorgungsspannung, RESET und NMI von der CPU). Auf keinen Fall darf man vergessen, die Handshake-Leitung STRB des A-Ports auf GND zu legen. Der EMUF selbst muß mit 4 MHz laufen, sonst stimmen die Zeitkonstanten für die serielle Schnittstelle nicht.

Schon recht komfortable Software

Das Monitor-Programm in *Bild 3* versorgt Tastatur, Anzeigen und Kassettenrecorder. Außerdem wurden einige Utilities wie z. B. MOVE und Displacement-Berechnung angefügt. Als besonderes „Zuckerl“ gibt es noch einen Downloader für Host-Rechner-Kopplung und eine DCF-77-Decodierung für angeschlossenen Empfänger. Der Monitor wurde so modular wie möglich angelegt, wobei alles außer der kurzen Monitor-Hauptschleife als Unterprogramm verwendbar ist. In den Unterprogrammen werden meist nur die parametertragenden Register verändert, die zweite Registerbank der CPU bleibt frei für den Anwender. Stack und Monitor-RAM belegen die Adressen 8000H-80FFH, ab 8100H aufwärts (mögliche RAM-Erweiterung) ist Platz für „Selbstgestricktes“. Die Interrupts NMI und IRQ werden vom Monitor über RAM-Vektoren versorgt. Bei RESET wird in beiden Bereichen (NMIVE Adr. 8000H IRQVE Adr. 8003H) ein kompletter Jump-Befehl

(C3...) zum Breakpoint-Entry des Monitor geladen. Die beiden Adreß-Bytes können nun vom Anwender auf seine Routinen gerichtet werden. So wird der fehlende JP (nn) des Z80 simuliert, wobei alle Register unverändert bleiben. Um die Kommando-Decodierung im Monitor für Erweiterungen zu öffnen, erfolgt am Ende der Monitor-Hauptschleife ein indirekter Sprung nach obigem Schema über den Monitor-Erweiterungsvektor (MONEVE Adr. 8006H). Er wird beim RESET auf die „Error“-Routine des Monitors gerichtet und bewirkt bei einem unbekanntem Kommando eine Fehlermeldung auf dem Display. Der Benutzer kann nun diesen Vektor auf eine selbst definierte Fortsetzung des Kommando-Decoders richten und so z. B. Monitor-Erweiterungen testen, bevor sie ins EPROM kommen.

Ein kommentiertes Source-Listing ist beim Franzis-Software-Service erhältlich.

Funktionen und Kommandos

Die Belegung der Tastatur ist in *Bild 4* gezeigt, die möglichen Monitor-Kommandos sind in der *Tabelle* aufgelistet. Die meisten Kommandos sind dialogorientiert gestaltet und benutzen das 7-Segment-Display zur Textausgabe so gut es eben geht. In der Monitor-Hauptschleife stellen auch die Hex-Tasten Kommandotasten dar, zur Zifferneingabe werden sie erst innerhalb der Kommandos benutzt. Mit Hilfe der Shift-Taste (^) stehen 38 Kommandotasten zur Verfügung. Die Verwendung der Shift-Funktion erfolgt wie beim Taschenrechner (zuerst Shift-, dann Kommandotaste). Die Shift-Funktion wird durch den

Tabelle 1: Die Monitor-Kommandos auf einen Blick

Folgende Tasten sind mit Funktionen belegt:

0	Speicher durchblättern
1	Speicherausschnitt verschieben
2	Speicher füllen
3	Programm seriell senden bzw. auf Kasette speichern
4	Programm von Kasette bzw. serieller Schnittstelle laden
5	Verify von Kasette
6	CPU-Register nach Break anschauen evtl. ändern
7	Continue nach Break
8	Relative Sprungweite berechnen
9	DCF 77 decodieren (St. Min. Sek.)
GO	Start des USER-Programms
MEM	Memory anzeigen

Dezimalpunkt ganz rechts im Display angezeigt. Ein nicht definiertes Kommando bringt „Error“ aufs Display (siehe auch Monitorerweiterung).

Bei der Hex-Eingabe werden die Ziffern von rechts nach links im Display durchgeschoben. Korrektur erfolgt durch Überschreiben. Innerhalb der einzelnen Kommandos wirkt die „+“-Taste wie eine Return-Taste zum Beenden der Zifferneingabe. Der Abschluß des Kommandos und die Rückkehr zum Monitor wird durch „Mon 3.2“ angezeigt. Mittels Reset kann jedes Kommando abgebrochen werden.

Die Monitor-Kommandos im einzelnen

Memory-Display: Taste Mem
Hiermit können Daten im RAM abgelegt werden.

Taste	Anzeige	
Mem	M.0000	Adreßeingabe
Mem	M.aaaa dd	Dateneingabe
^Mem	Mon	Rückkehr zur Monitor-Schleife

Mit der Mem-Taste kann zwischen Adreß- und Dateneingabe hin- und hergeschaltet werden.

Go-Kommando: Taste Go
Go G 0000 Eingabe der Startadresse

Nochmaliges Betätigen der Go-Taste schickt die CPU auf die Reise. Jede andere Kommando-Taste bewirkt Rückkehr zum Monitor. Beim Start des User-Programms werden alle CPU-Register (Bank 1) aus dem Puffer im RAM geladen (siehe Break-Kommando). Ein offenes RET am Ende des Anwenderprogramms führt zurück zum Monitor.

Speicher durchblättern: Taste 0

0	Sta 0000	Eingabe der Startadresse
+	M.aaaa dd	
+	M.aaaa+1 dd	vorwärts blättern
Go	M.aaaa-1 dd	rückwärts blättern
Mem	M.aaaa dd	Einsprung ins normale Mem-Kommando
^Mem	Mon	Rückkehr zum Monitor

Hier kann man z. B. nach einem bestimmten Byte im Speicher suchen, da die „+“- und die „Go“(„-“-Taste mit Autorepeat ausgestattet sind.

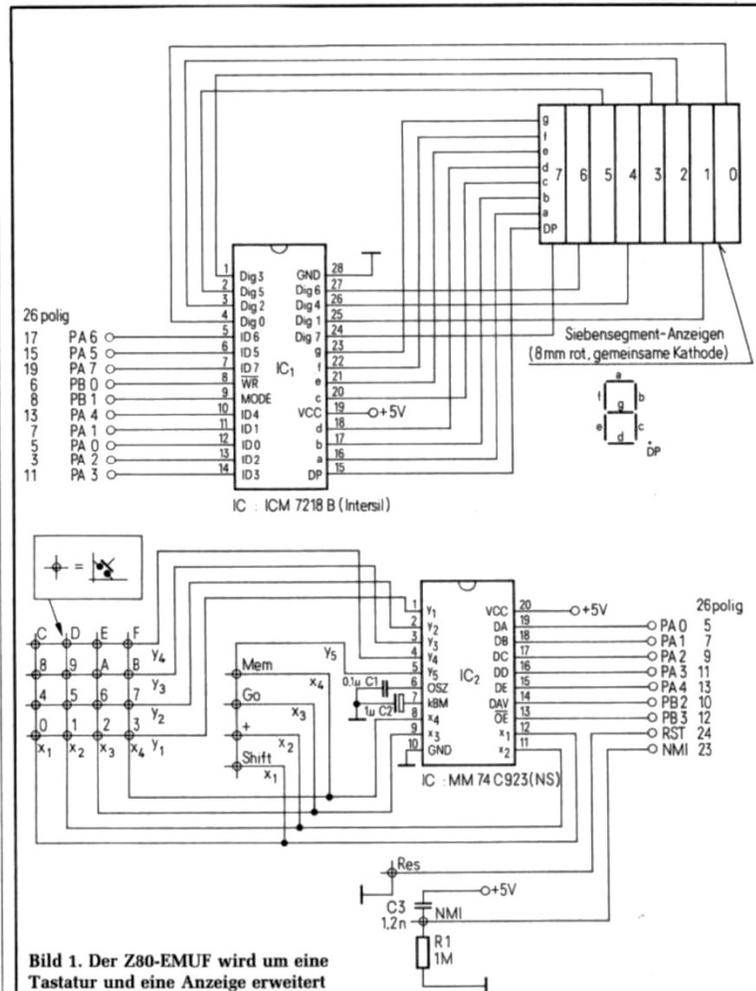


Bild 1. Der Z80-EMUF wird um eine Tastatur und eine Anzeige erweitert

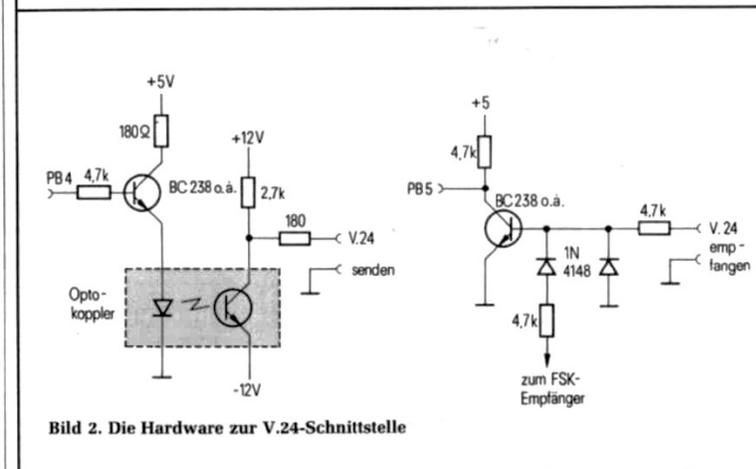


Bild 2. Die Hardware zur V.24-Schnittstelle

MOVE-Kommando: Taste 1
 1 Sta 0000 Startadresse des Blocks eingeben
 + End 0000 Endadresse
 + to 0000 Zieladresse
 + Mon
 Bei Startadresse \leq Endadresse erfolgt eine Fehlermeldung

FILL-Kommando: Taste 2
 2 FIL 0000 Eingabe des „Fill-Bytes“ (nur die beiden rechten Ziffern sind gültig)
 + Sta 0000 Startadresse
 + End 0000 Endadresse
 + Mon

SAVE-Kommando: Taste 3
 3 Sta 0000 Start des Speicherblocks
 + End 0000 Ende des Speicherblocks
 + Esp 0000 Einsprungadresse (siehe unten)
 + Idnr 0000 Eingabe einer vierstelligen Kennnummer des Programmblocks auf Band
 + Sen XX XX „flimmernde“ Sendedaten

Die Einsprungadresse wird mit aufgezeichnet und nach dem Laden des Datenblocks startet der Prozessor an dieser Stelle. Dadurch wird ein Autostart des Programms möglich. Wird als Einsprungadresse 0000 eingegeben bzw. „+“ gedrückt, dann kehrt der Rechner nach dem Laden zum Monitor zurück.

LOAD-Kommando: Taste 4
 4 Idnr 0000 Eingabe der Kennnummer des zu ladenden Programms. Wird 0000 eingegeben bzw. „+“ gedrückt, dann wird das nächste vollständige Programm vom Band geladen.
 + Emp 00 Empfangsbereitschaft des Prozessors

Sobald Daten empfangen werden, verändert sich die Anzeige:

iiii xx iiii Id-Nummer des empfangenen Programms
 xx empfangene Daten nach abgeschlossenem Ladevorgang siehe LOAD-Kommando.

Verify-Kommando: Taste 5
 5 EMP 00 siehe LOAD

Das auf Band geschriebene Programm kann mit dem Speicherinhalt verglichen werden. Wenn keine Fehler aufgetreten sind, erfolgt Rückkehr zum Monitor, andernfalls wird „Error“ gemeldet und die Operation abgebrochen.

Anzeige der CPU-Register: Taste 6
 Diese Funktion ist nur sinnvoll nach einem NMI oder Breakpoint. Nur dann sind die Puffer-Zellen im RAM auf aktuellem Stand.

6 AF aaff Anzeige von Akku und Flags, zu ändern wie Memory-Zellen
 + BC bbcc
 ...
 + IY yyyy
 + Mon

Continue-Kommando: Taste 7
 Auch dieses Kommando sollte nur nach NMI oder Break verwendet werden. Andernfalls kann es zum Absturz des Rechners führen. Die eventuell modifizierten RAM-Zellen werden in die CPU-Register geladen und das Programm beim augenblicklichen Programmzählerstand fortgesetzt.

Displacement-Rechner: Taste 8
 Hiermit kann der Offset für die relativen Sprünge der Z80-CPU berechnet werden.

8 bra 0000 Adresse des JR-Befehls (nicht des Displacement-Bytes) eingeben
 + to 0000 Adresse des Ziel-Labels
 + disp xx Displacement
 Wurde die maximale Sprungweite überschritten, so erscheint die Meldung „too long“ im Display.

DCF-77-Decodierung: Taste 9
 Mit diesem Kommando kann der „Kleine“ nach Anschluß eines einfachen DCF-77-Empfängers an PIO2, PB 6, in eine hochgenaue Uhr verwandelt werden. Der Empfänger sollte TTL-Pegel (Ruhezustand high) liefern. Der EMUF verdaut für log. 0 80...120 ms Absenkung, für log. 1 180...220 ms.

Um den Programmaufwand gering zu halten, decodiert der Rechner nur Stunden, Minuten und Sekunden und zeigt sie auf dem Display an. Nach Start des Programms wartet der EMUF die 59-Sekunden-Marke ab und zeigt 00 00 an. Nach einer weiteren Minute werden auch die Stunden und Minuten ange-

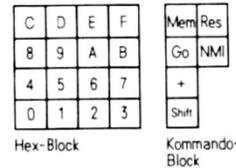


Bild 4. Die Belegung der kleinen Tastatur

zeigt, die maximale Synchronisationszeit beträgt also 1 Minute und 59 Sekunden. Treten während des Empfangs Störungen auf, so erlischt das Display, um sinnlose Zeitanzeigen zu vermeiden; der Rechner versucht erneut zu synchronisieren. Aus diesem Programm kann der EMUF nur mit Reset zurückgeholt werden.

BREAK- und STOP-Befehl

Als BREAK-Befehl wird, wie allgemein üblich, der RST 38H (OFFH) verwendet. Taucht er im Programm auf, so werden die CPU-Register ins RAM gerettet und man landet im Monitor (siehe auch Kommando 6 und 7).

Der NMI der CPU wird als STOP-Befehl interpretiert, um die CPU definiert aus endlosen Schleifen herauszuholen. Die Register werden ebenfalls gerettet. Dieses Kommando funktioniert nur, solange der NMI-Vektor im RAM nicht verändert wird. Zum Entprellen der NMI-Taste wird ein einfaches RC-Glied verwendet (siehe Bild 1).

Im folgenden soll noch ausführlich auf die Möglichkeiten des seriellen Empfangs bzw. Sendens mittels des softwaremäßig implementierten USART's eingegangen werden.

Die Tasten 3 + 4 zum Abspeichern und Laden von Programmen bedienen sich des Datenformats in *Tabelle 2*.

Der Markierzustand der Schnittstelle ist high. Die Daten haben seriell folgende Form:

8 Bit breite Worte, 1 Startbit, 2 Stopbits. Die Übertragungsrates beträgt 300 Baud. Durch Änderung der entsprechenden RAM-Adressen (BITNR 8011H, BAUDCT 800FH) lassen sich Baudraten bis zu 2400 Baud und beliebige (bis 8 Bit) Wortbreiten einstellen. Entsprechende Versuche sind bereits erfolgreich durchgeführt worden. Die Sendung von

einem Startbit und 2 Stopbits ist fest programmiert.

Hierdurch ergibt sich die interessante Möglichkeit, Programme, die dieses Format einhalten, von einem Host-Rechner zu laden und automatisch ausführen zu lassen. Wer dieses Schema nicht einhalten möchte, kann direkt auf die Schnittstellentreiber zugreifen und sein eigenes Übertragungsprotokoll ablaufen lassen.

Der Rechner könnte durch Nutzung der seriellen Schnittstelle und entsprechender Programme völlig ferngesteuert werden. Außerdem gilt es noch zu beachten, daß er nur entweder senden oder empfangen kann. Beides gleichzeitig ist nicht möglich.

Die Aufzeichnung der Programme mittels des (Funkschau-)FSK-Modems funktionierte auf Anhieb. Die Anforderungen in bezug auf Kassetten- und Recorderqualität sind gering. Da keine Umschaltung von Recorderinterface auf die serielle Schnittstelle notwendig ist, versteht es sich von selbst, daß beim Empfang nur eine der beiden Quellen in Betrieb sein darf. Die Anschlüsse der nicht benutzten Schnittstelle müssen offen bleiben.

Tabelle 2: Das Datenformat beim Speichern auf Kassette

gesendete Bytes	Anzahl	
0	1	
A5H	1	Marken für Programmbeginn
55H	1	
Kennnummer	2	Reihenfolge low Byte, high Byte
3CH	1	Startbyte für 1. Datenfeld
Länge	1	Länge des Datenfeldes, max. 256 Bytes/Block
Quelladresse	2	low Byte, high Byte
Datenfeld	Länge	
Checksum	1	(Summe aller gesendeten Bytes pro Block) modulo 256
2. Datenblock		
n-ter Datenblock		
letzter Datenblock		
78H	1	Marke Fileende
Einsprungsadresse	2	low Byte, high Byte

Wichtige RAM-Adressen

8000H NMIVE

Sprungvektor für NMI 'C3 xx xx'

8003H IRQVE

Sprungvektor für Interrupt Mode 1 bzw. RST 38H.
Analog NMIVE

8006H MONEVE

Monitorerweiterungsvektor. Zeigt nach der Initialisierung auf die Error-Routine. Hier kann der Benutzer Fehlermeldungen abfangen, die nach einer nicht definierten Taste in der Monitorschleife ausgegeben werden. Der Akku enthält zu diesem Zeitpunkt 'Tastencode * 2'. 'SLR A' ergibt den Tastencode.

800FH BAUDCT

Zeitkonstante für Baudrate. Durch entsprechende Werte lassen sich die Baudraten einstellen. Folgende Beziehung gilt:

$$1/\text{Baudrate} = (\text{BAUDCT}) * 100 \text{ Mikrosekunden}$$

Folgende Baudraten wurden erprobt:

Baudrate	BAUDCT
75	83H
150	42H
300	21H
600	10H
1200	8H
2400	4H

Nach einem Reset sind 300 Baud eingestellt.

8011H BITNR

Anzahl der Bits pro seriellem Datenwort.

8013H DISBUF

Displaypuffer. Hier stehen die Bitmuster für den aktuellen Displayinhalt (8 Zeichen).

Segment – Bit Zuordnung (s. Bild 1):

Bit	7	6	5	4	3	2	1	0
Segment	DP	a	b	c	e	g	f	d

Wichtige ROM-Adressen und Unterprogramme

011FH Monitorwarmstart

0796H Kommandotabelle

0138H KEYIN Wartet auf Taste. Tastenwert im Akku, berücksichtigt Shift-Taste

0178H GETKEY Tastenfeldabfrage
Taste gedrückt → Carry = 0, A = Tastencode
Taste nicht gedrückt → Carry = 1, A = xx

01C1H HLDISP HL als 4-Digit-Zahl ausgeben. DE ist Pointer auf 1. Digit

01CFH TO-DIBUF Hexzahl im Akku wird als Bitmuster im Displaypuffer abgelegt. DE ist Pointer auf 1. Stelle

0235H HLINCL Eingabe einer 4stelligen Hexzahl in HL. Durchschieben und Echo auf Display (DE als Pointer). Abbruch der Eingabe durch Kommandotaste (> 0FH)

05200 SERSEN Byte in A seriell senden (siehe BAUDCT und BITNR)

054E SEREMP Byte seriell empfangen (siehe oben). Empfangenes Byte im Akku

06B7H DISPOUT Ausgabe des Displaypuffers aufs Display (siehe DISBUF)

0711H FILBUF Zeichenkette in den Displaypuffer schreiben:
Format:
n CC...C 0 ≤ n ≤ 8
n = Anzahl der Textzeichen (Bitmuster)
Register HL zeigt auf ..n"

Literatur

[1] Feichtinger, Herwig: FSK-Demodulator/Oszillator. Funkschau 1978, Heft 14, Seite 698 und Heft 15, Seite 742.