

# 80C537-Mikrocontroller-board

Peter Pramberger

DSK340://80C537

## Allgemeines

Die Schaltung ist auf einer einseitigen Platine im Europaformat untergebracht. Die erforderliche zweite Verdrahtungsebene ist mittels Drahtbrücken realisiert und daher leicht nachbaubar.

Das CMOS-RAM des Mikrocontrollerboards kann mittels Batterie oder Akku gepuffert werden. Es sind alle Ports ausgenommen Port 8 (4 A/D-Eingänge) herausgeführt und die zwei seriellen Schnittstellen auf V24 umgesetzt. Die Stromversorgung ist integriert, so daß nur eine Einspeisung von 8-12V Wechselspannung erforderlich ist.

Bei Verwendung des 'KELL'-Monitors kann ein 'HEX'-File über die S1-Schnittstelle downgeladen werden. Der Monitor (EPROM) schaltet sich dann auf die Adresse 8000h, wodurch das downgeladene Programm dann im RAM steht (inklusive Interrupt-Vektoren).

## Bestückung

Zunächst sollte anhand der Stückliste überprüft werden, ob alle Bauteile vorhanden sind. Außerdem sollten alle Leiterbahnen auf der Platine auf Kurzschlüsse überprüft werden. Falls notwendig, sollte die Kupferseite der Platine mit Lötack überzogen werden, um die Leiterbahnen vor Korrosion und Beschädigung zu schützen.

## Drahtbrücken

Die Drahtbrücken mit den Nummern 1 und 2 werden abhängig vom verwendeten Mikrocontroller eingelötet. Wie auf dem Bestückungsplan angeführt, ist bei der Verwendung des SAB 80C537 die Drahtbrücke 1 einzulöten, während Drahtbrücke 2 weggelassen wird. Bei Verwendung des SAB 80C517A ist die Drahtbrücke 2 zu bestücken, während Drahtbrücke 1 weggelassen wird.

Bevor die übrigen Drahtbrücken eingelötet werden, noch ein kurzer Hinweis. Das Mikrocontrollerboard kann in zwei verschiedenen Modi betrieben werden:

- Monitorbetrieb:** Hier steht im EPROM das Monitorprogramm, welches die Kontrolle und Steuerung des Mikrocontrollers über den PC erlaubt. Es werden nur die Drahtbrücken zusätzlich eingelötet, die mit einem Kreuz markiert sind. Dadurch kommt der für den Monitorbetrieb notwendige Adreßwechsel zustande. Das auszuführende Programm wird über die Schnittstelle S1 downgeladen und steht dann im RAM.
- Applikationsbetrieb:** Diese Betriebsart wird dann eingesetzt, wenn die Platine für eine Anwendung programmiert wurde (Applikation). Dabei wird das fertige Programm in ein EPROM gebrannt. Hier sind die Drahtbrücken mit einem Kreuz wegzulassen und statt dessen die Drahtbrücken, die mit 1a bzw. 1b markiert sind, einzulöten.

Zum Schluß werden die noch verbliebenen Drahtbrücken eingelötet.

## Taster

Zunächst sind die zwei an der abgeflachten Seite der Taster liegenden Anschlüsse abzuwickeln (siehe Bestückungsplan). Anschließend werden die Taster, mit der abgeflachten Seite zu einer der Breitseiten der Platine zeigend, eingelötet.

## Übrige Bauteile

Als nächstes sind die IC-Sockel an der Reihe. Sie sollten gleich richtig herum eingelötet werden, um späteren Verpolungen der IC's vorzubeugen. In weiterer Folge sind dann die Kondensatoren (auf Polung achten!), Widerstände, Dioden, Leuchtdioden (Kathode!) und sonstige Bauteile einzulöten. An manchen Stellen kann es aus Platzgründen notwendig sein, die Bauteile auf der Unterseite der Platine anzulöten (z.B. Widerstand beim Sicherungshalter). Zuletzt werden die IC's in die Fassungen gesteckt. Abschließend sollte dem Spannungsregler ein Kühlkörper spendiert werden, um ihn vor Überlastung zu schützen.

Die grüne Leuchtdiode zeigt die Betriebsbereitschaft an, während die rote Leuchtdiode Unterspannung (bzw. RESET-Zustand) anzeigt.

## Schnittstellen

Die Mikrocontroller-Platine besitzt zwei Schnittstellen, die auf V24 umgesetzt sind. Schnittstelle S0 ist frei verfügbar. Schnittstelle S1 wird im Monitorbetrieb für die Datenübertragung zwischen Mikrocontroller und PC verwendet. Im Applikationsbetrieb ist sie frei verfügbar.

Vorsicht bei Verwendung der Pins 3.0 und 3.1! Diese dürfen nur verwendet werden, wenn die Schnittstelle S1 nicht in Betrieb ist.

## Neuer Baudraten-Generator

Die SAB 80C537-Mikrocontroller mit Stepping Code 'CA' oder höher beinhalten einen neuen Baudraten-Generator. Dieser liefert bessere Ergebnisse in Bezug auf die Genauigkeit der erzeugten Baudrate bei Verwendung anderer Taktfrequenzen als 11,059 MHz.

Der Baudraten-Generator besteht aus einem freilaufenden 10-Bit-Zähler mit automatischem Reload. Nach einem Überlauf wird der Timer aus den Registern S0RELL (AAh) und den ersten zwei Bit von S0RELH (BAh) bzw. S1RELL (9Dh) und den ersten zwei Bit von S1RELH (BBh) automatisch nachgeladen.

Der Baudraten-Generator ist in den Modes 1,3 (für Schnittstelle S0) sowie in den Modes A,B (für Schnittstelle S1) verfügbar.

## Berechnung des Reload-Wertes

Der Reload-Wert errechnet sich nach folgenden Formeln:

$$\text{Baudrate} = \frac{2^{\text{SMOD}} \times \text{Taktfrequenz}[\text{Hz}]}{64 \times (2^{10} - \text{Reload})} \quad \text{Baudrate} = \frac{\text{Taktfrequenz}[\text{Hz}]}{32 \times (2^{10} - \text{Reload})}$$

für Schnittstelle S0

für Schnittstelle S1

Um für beide Schnittstellen die selben Werte verwenden zu können, wird SMOD immer auf 1 gesetzt, wodurch die zweite Formel gilt.

Die Tabelle enthält die Reload-Werte für die gängigsten Taktfrequenzen.

Baudrate	Reload-Wert		
	11,0592 MHz	12 MHz	16 MHz
1200 Baud	2E0h	2C8h	260h
2400 Baud	370h	364h	330h
4800 Baud	3B8h	3B2h	398h
9600 Baud	3DCh	3D9h	3CCh
19200 Baud	3EEh	3ECh	3E6h

## Programmierung

Dieser Abschnitt befaßt sich mit der Programmierung des neuen Baudraten-Generators. Für Detailfragen ist das Siemens-Mikrocontroller-Handbuch zu Rate zu ziehen, dort ist auch detailliert die Programmierung der Schnittstellen aufgeführt.

Die Programmierung des Baudraten-Generators gestaltet sich recht einfach. Zunächst ist der richtige Betriebsmodus der entsprechenden Schnittstelle einzustellen:

Mode 1 oder Mode 3 bei Schnittstelle S0,  
Mode A oder Mode B bei Schnittstelle S1.

Anschließend ist der Reload-Wert aufzuteilen: Die letzten acht Bit (2 Hex-Stellen) werden in SxRELL geschrieben, die übrigen zwei Bit (1 Hex-Stelle) nach SxRELH. Der Timer wird durch Setzen von Bit BD (DFh) gestartet.

## Beispiel

Die Schnittstelle S0 soll im Mode 1 mit 1200 Baud übertragen, die Schnittstelle S1 im Mode B mit 9600 Baud für den Monitorbetrieb. Der Mikrocontroller wird mit 16 MHz getaktet. Gesucht ist die Initialisierung der Schnittstellen.

## Aus Tabelle

1200 Baud bei 16 MHz → Reload = 260h  
 9600 Baud bei 16 MHz → Reload = 3CCh.

S0CON	DATA	098H	; Festlegen der Register-Adressen
S0RELL	DATA	0AAH	
S0RELH	DATA	0BAH	
S1CON	DATA	09BH	
S1RELL	DATA	09DH	
S1RELH	DATA	0BBH	
PCON	DATA	087H	
BD	BIT	0DFH	

```

InitSerial: MOV S0CON, #01111010B ; Schnittstelle S0 initialisieren
            ORL S0RELH, #002H ; Reload-Wert programmieren
            MOV S0RELL, #060H ; fuer 1200 Baud

            MOV S1CON, #10111010B ; Schnittstelle S1 initialisieren
            ORL S1RELH, #003H ; Reload-Wert programmieren
            MOV S1RELL, #0CCH ; fuer 9600 Baud

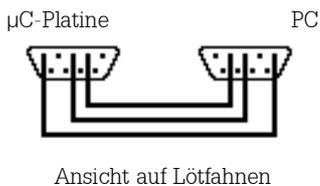
            ORL PCON, #10000000B ; SMOD = 1
            SETB BD ; Start der Baudraten-Generatoren
    
```

Achtung! Beim Programmieren des Reload-Wertes muß unbedingt **zuerst SxRELH** und erst **danach SxRELL** beschrieben werden, sonst wird der Timer nicht korrekt programmiert!

## Monitorbetrieb

Neben der Mikrocontroller-Platine wird noch ein serielles Verbindungskabel und die 'KEIL'-Software benötigt. Unter anderem befindet sich darunter ein Programm ('MON51'), welches den Mikrocontroller vom PC aus steuern kann (Haltepunkte setzen, Registerinhalt anzeigen,...).

### Seriell Kabel



Benötigt werden zwei 9-polige RS232-Buchsen. Das Kabel benötigt nur drei Adern (TxD, RxD, GND), welche einfach 1:1 durchverbunden werden, wobei auf einer Buchse die Zählrichtung der Anschlüsse (Sicht auf die Lötflächen) gegenüber der anderen Buchse entgegengesetzt verläuft.

## Monitor

Zunächst wird die Mikrocontroller-Platine an ein Netzgerät (Stecker-netzteil, Trafo, etc.) angeschlossen. Danach wird die Platine mittels des seriellen Verbindungskabels an einen seriellen Port des PC's angeschlossen.

Das Monitorprogramm wird mit dem Aufruf von 'MON51' gestartet (Syntax siehe Abbildung).

```

MS-DOS MON51 V1.05
(C) Franklin Software Inc./KEIL ELEKTRONIK GmbH 1991,1993

*** USAGE: MON51 [ Controls ] <cr>

Controls:
-----
COM1: | 1 specifies Serial Interface COM1:
COM2: | 2 specifies Serial Interface COM2:
COM3: | 3 specifies Serial Interface COM3:
COM4: | 4 specifies Serial Interface COM4:
INT14 | I use bios interrupt 14 for serial I/O
NOINT | N use direct port access for serial I/O

BaudRate (bps) transfer speed in bits per second

Example: >MON51 BAUDRATE (2400) COM2:
    
```

Wenn die serielle Verbindung fehlerfrei aufgebaut wurde, dann zeigt das Zeichen '#' die Bereitschaft des Monitors an. Jetzt können die nachfolgend beschriebenen Funktionen ausgeführt werden.

### Befehlsübersicht

Auf der nächsten Seite sind die Befehle und deren Syntax abgebildet. Sie werden im Anschluß noch genauer erklärt.

#### Speicher

Bietet verschiedene Möglichkeiten, den Inhalt der Speicher zu manipulieren.

### Anzeigen

Zeigt den Inhalt eines Speicherbereichs an. Abhängig vom zu betrachtenden Speichertyp unterscheiden sich die Befehle im letzten Buchstaben.

Syntax: Dx [Startadresse] [Endadresse]  
 Bsp.: DC 0000h 07FFh

Zeigt den Code-Speicher von Adresse 0000h bis Adresse 07FFh an.

### Verändern

Verändert den Inhalt des Speichers. Die Eingabe wird jeweils mit 'ESC' beendet.

Syntax: Ex [Startadresse]  
 Bsp.: DX 8005h

Verändert den Inhalt des XDATA-Speichers von Adresse 8005h.

### Füllen

Füllt einen angegebenen Speicherbereich mit einem Zahlenwert.

Syntax: FILLx [Startadresse] [Endadresse] [Wert]  
 Bsp.: FILLX 8000h 87FFh 00

Füllt den Speicherbereich 8000h bis 87FFh im RAM mit dem Wert 00.

### Utility

Bietet verschiedene Bearbeitungsfunktionen.

### Assemblieren

Bietet die Möglichkeit, Code direkt im Speicher zu bearbeiten. Mit 'ESC' wird die Eingabe beendet.

Syntax: A [Startadresse]  
 Bsp.: A 0100h

Bearbeitet den Code ab Adresse 100h.

### Disassemblieren

Zeigt ab der angegebenen Adresse Maschinencode als Assemblerbefehle an.

Syntax: U [Startadresse] [Endadresse]  
 Bsp.: U 0100h 0105h

Zeigt den Code von Adresse 100h bis Adresse 105h an.

### Register

Zeigt oder bearbeitet den Inhalt der (angegebenen) Register.

Syntax: X [Register]  
 Bsp.: X R0

Verändert den Inhalt von R0.

### Programmausführung

Bietet verschiedene Möglichkeiten zur Programmsteuerung.

### Ausführen

Führt ein zuvor geladenes Programm aus.

Syntax: G [Startadresse] [Haltepunkt-Adresse]  
 Bsp.: G 0100h

Führt ein Programm ab Adresse 100h aus.

### Schrittweise Ausführung

Führt ein Programm im Einzelschrittmodus aus.

Syntax: T [Anzahl] ... Einzelschritt  
 P [Anzahl] ... Prozedur-Schritt  
 Bsp.: T 10

Führt die nächsten zehn Befehle aus.

### Haltepunkte

Diese Befehle verwalten die Einstellungen der Haltepunkte.

### Deaktivieren

Deaktiviert einen vorhandenen Haltepunkt.

Syntax: BD [Nummer]  
 Bsp.: BD 3

Deaktiviert den 4. Haltepunkt (Zählbeginn bei 0).

**Aktivieren**

Reaktiviert einen vorhandenen Haltepunkt wieder, nachdem er deaktiviert wurde.

Syntax: BE [Nummer]  
Bsp.: BE 5

Aktiviert den 6. Haltepunkt wieder.

**Löschen**

Löscht einen vorhandenen Haltepunkt.

Syntax: BK [Nummer]  
Bsp.: BK 2

Löscht den 3. Haltepunkt.

**Setzen**

Setzt einen neuen Haltepunkt.

Syntax: BS [Adresse]  
Bsp.: BS 0100h

Setzt an der Adresse 100h einen Haltepunkt.

**Auflisten**

Listet alle vorhandenen Haltepunkte auf. Die jeweilige Nummer am Beginn der Liste gibt die Haltepunkt-Nummer an.

Syntax: BL

**Dateien**

Diese Befehle sind zuständig für das Laden und Speichern von Dateien.

**Laden**

Lädt eine 'HEX'- bzw. 'OBJ'-Programmdatei in den Speicher der Mikrocontroller-Platine.

Syntax: LOAD [Datei.HEX] oder  
LOAD [Datei.OBJ]  
Bsp.: LOAD Port.HEX

Lädt die Datei 'Port.hex' in den Speicher des µC-Boards

**Speichern**

Speichert im Speicher stehenden Code in eine 'HEX'-Datei.

Syntax: SAVE [Datei.HEX] [Startadresse] [Endadresse]  
Bsp.: SAVE Port.HEX 0000h 0500h

Speichert den Adreßbereich von 0000h bis 0500h in die Datei Port.hex.

**Symbole**

Lädt Symboldefinitionen in den Speicher.

Syntax: LS [Symboldatei]

**Hilfe**

Die Hilfe kann durch die Eingabe des Befehls 'HELP' angezeigt werden.

```
MS-DOS MON51 V1.05
(C) Franklin Software Inc./KEIL ELEKTRONIK GmbH 1991,1993

INSTALLED FOR PC/XT/AT (COM LINE 2) USING HARDWARE INTERRUPT SERVICE
*** MONITOR MODE ***

#HELP
memory display modify fill utility
bit: >DB range >EB address >FILLB range value >A address - assemble
code: >DC range >EC address >FILLC range value >U range - disassemble
data: >DD range >ED address >FILLD range value >X [register] - dis/chg
idata: >DI range >EI address >FILLI range value
xdata: >DX range >EX address >FILLX range value
pdata: >DP range >EP address >FILLP range value

program execution breakpoint(s) program load/save
>G [address] [,breakadd] - go >BD bp - disable >LOAD file - load hex/obj
>T [count] - trace step >BE bp - enable >SAVE file range- save hex
>P [count] - procedure step >BK bp - kill >LS file - load symbols
>BL - list

>HELP - display menu
```

**Literatur**

PCNEWS *edit* -49, Seite 86: Mikrocontrollerboard für 80c537

**Bauteil-Liste**

	Name	Anz.	Benennung
Widerstände	R1	1	Kohleschicht, 3k3, ±5%, 0,25W
	R2	1	Kohleschicht, 3k3, ±5%, 0,25W
	R3	1	Kohleschicht, 10k, ±5%, 0,25W
	R4	1	Kohleschicht, 1k, ±5%, 0,25W
	R5	1	Kohleschicht, 1k, ±5%, 0,25W
	R6	1	Kohleschicht, 1k, ±5%, 0,25W
	R7	1	Kohleschicht, 3k3, ±5%, 0,25W
Kondensatoren	C1	1	Keramik, 22p, ±2%, 100V
	C2	1	Keramik, 22p, ±2%, 100V
	C3	1	Aluminium, 10u, ±20%, 16V
	C4	1	Tantal, 10u, ±20%, 25V
	C5	1	Tantal, 10u, ±20%, 25V
	C6	1	Tantal, 10u, ±20%, 25V
	C7	1	Tantal, 10u, ±20%, 25V
	C8	1	Keramik, 10n, ±20%, 63V
	C9	1	Keramik, 10n, ±20%, 63V
	C10	1	Keramik, 10n, ±20%, 63V
	C11	1	Keramik, 10n, ±20%, 63V
	C12	1	Aluminium, 2200u, ±20%, 35V
	C13	1	Keramik, 100n, ±20%, 63V
	C14	1	Tantal, 1u, ±20%, 35V
	C15	1	Keramik, 100n, ±20%, 63V
	C16	1	Aluminium, 4700u, ±20%, 10V
	C17	1	Keramik, 100n, ±20%, 63V
	C18	1	Aluminium, 4700u, ±20%, 10V
Dioden	D1	1	Silizium, 1N4004, 1A, 400V
	D2	1	Silizium, 1N4004, 1A, 400V
	D3	1	Silizium, 1N4004, 1A, 400V
	D4	1	Silizium, 1N4004, 1A, 400V
	D5	1	Silizium, 1N4148, 0,1A, 75V
	D6	1	Zenerdiode, 5V1, 1,3W
	D7	1	Leuchtdiode, Rot, 5mm
ICs	IC1	1	MAX232 (RS232)
	IC2	1	SAB 80C537-N16 (16 Mhz µC)
	IC3	1	74HCT573 (Bus-Treiber)
	IC4	1	74HCT00
	IC5	1	74HCT00
	IC6	1	KM62256 (SRAM)
	IC7	1	KM62256 (SRAM)
	IC8	1	AM27C256 (EPROM)
	IC9	1	L387A (Spannungsregler)
Transistoren	T1	1	FET NPN, BS170, 60V, 0,63W (SIPMOS)
	T2	1	FET NPN, BS170, 60V, 0,63W (SIPMOS)
Quarz	Q1	1	Quarz, 16 MHz (HC49U)
Taster	TA1	1	Miniatur-Printtaster, Rund, (Schließer)
	TA2	1	Miniatur-Printtaster, Rund, (Schließer)
Sicherung	SI	1	Glasrohr, 315mA, T, 5x20mm, (+ Halter)
Schnittstellen	DB9	2	RS232, 9-polig, Stecker Abgewinkelt, für Printmontage
IC-Sockel		1	Socket, PLCC84,
		3	Präzisionssockel, 28-polig,
		1	Präzisionssockel, 20-polig,
		1	Präzisionssockel, 16-polig,
		2	Präzisionssockel, 14-polig,
Steckverbinder		1	Steckerleiste, 26-polig
		1	Steckerleiste, 20-polig
		1	Steckerleiste, 10-polig
		1	Steckerleisten-Wanne, 26-polig
		1	Steckerleisten-Wanne, 20-polig
		1	Steckerleisten-Wanne, 10-polig