

# SIOS-Betriebssystem

## - Kommandos -

(Stand 21.11.96)

Dieser Text wendet sich in erster Linie an Entwickler, die Software für SIOS erstellen wollen. Die Kenntnis der gültigen Interface-Kommandos ermöglicht Lösungen mit beliebigen Programmiersprachen.

## 1 Grundlagen

Nach dem Einschalten befindet sich das SIOS im Basismodus und wartet auf gültige Kommandos. Es gelten folgende Einstellungen:

Serielle Schnittstelle: 19200 Baud, 8 Bits, 2 Stopbits  
Meßbereiche der Analogeingänge A und B: 0...5V  
Meßbereiche der Analogeingänge C und D: 0...2,5V  
Referenzspannung an den Sensorbuchsen C und D: 2,5V  
Interne Zeitbasis: 1 ms  
Interface-Kennung: 10 (Version 1.0)

Neben den SIOS-Kommandos werden auch die Compactbox-kompatiblen Kommandos (81, 176, 210, 211) ausgeführt, so daß alte Programme weiterhin arbeiten. Zusätzlich kann das Interface in einen CompuLAB-kompatibeln Modus versetzt werden. Es liefert dann die Kennung 201 (CompuLAB, Version 2.01). Neben den alten Kommandos der Compactbox werden in der Emulation auch die CAMFACE-Kommandos (64, 72, 58, 69) ausgeführt.

Das SIOS enthält eine Funktionstest ohne angeschlossenen PC: Beim Start muß der digitale Eingang D0 hochgesetzt sein. Dann erscheint das Bitmuster 01010101 an den Ausgängen.

Durch entsprechende Kommandos kann im Basismodus die Baudrate bis 57,6 Kilobaud eingestellt werden. Die Zeitbasis von 1 ms kann für Meßserien auf 0,2ms oder 10 ms verändert werden. Die Referenzspannungen der Sensoranschlüsse werden durch zwei Analog-Kanäle gebildet und sind frei veränderbar. Der Standardwert 2,5V wird mit dem Parameter 128 (= 1/2 Ub) gewählt.

Für jeden der vier Eingänge kann der Meßbereich für 8-Bit-Messungen in weiten Grenzen eingestellt werden, wobei jeweils eine untere und eine obere Grenze durch einen 4-Bit-Parameter festgelegt sind, die zusammen einen Byte-Parameter bilden. Bit 0...3 legt die untere Grenze als Vielfaches von 5V/16 fest, Bit 4...7 ebenso die obere Grenze. Beide Referenzspannungen müssen einen Mindestabstand von 1,25V einhalten. Beispiel: 84h bewirkt den Meßbereich 1,25V-2,5V. Ausnahme: 00h=0.5V. Die durch spezielle Kommandos eingegebenen Meßbereiche überschreiben die Standardauflösungen von 5V, 5V, 2,5V und 2,5V nur für Direktmessungen im 8-Bit-Modus. 10-Bit-Messungen verwenden grundsätzlich den vollen Bereich von 0...5V. Für Serienmessungen erfolgt eine eigene Einstellung der Bereiche im Definitionsblock der Messung.

Serienmessungen werden durch insgesamt 14 Parameter definiert. Dabei werden die Anzahl der Einzelmessungen, die aktiven Kanäle in 8- oder 10-Bit-Auflösung, die Meßbereiche getrennt für jeden Kanal, die Intervallzeit, die Zeitbasis und ein Steuerbyte für Zusatzoptionen angegeben. Die Intervallzeit wird als 16-Bit-Wert als Vielfaches der gewählten Zeitbasis von 0,1ms, 0,2ms, 1ms oder 10ms angegeben. Die schnellste einstellbare Abtastung für beliebige Messungen erfolgt im 0,2-ms-Raster (Abtastrate 5kHz). Für Einkanalmessungen ohne Zusatzoptionen ist auch eine Zeitbasis von 0,1ms erlaubt. Noch schnellere Messungen werden durch nachgeladene Spezialprogramme erreicht.

Beispiel: 112,0,100,1,2,3,4,0,0,128,128,0,6,2,1 stellt ein:

100 Messungen, alle Kanäle, Standardauflösung 8 Bit ,Intervall 10ms, Zeitbasis 1ms, der Digitalausgang D0 zeigt die Dauer der Messung. Die eigentliche Messung wird durch ein eigenes Kommando gestartet. Alternativ stehen Startkommandos mit Triggerung zur Verfügung. Mit "0" kann die Messung vorzeitig abgebrochen werden. Die Unterbrechung entspricht einem Neustart, wobei allerdings alle Meßdaten erhalten bleiben. Nach dem Ende einer Messung sendet SIOS ein Ready-Byte (1).

Nach dem Ende der Messung können die Meßdaten ausgelesen werden. Der Umfang der Daten kann bis zu 30 kB betragen. Das Datenformat der Meßserien ist wie folgt festgelegt: 14 Bytes des Definitionsblocks, dann Meßergebnisse für alle Kanäle geschachtelt, je 1 Byte für 8-Bit, 2 Bytes für 10-Bit-Messungen. Bei 10-Bit ist der im Definitionsblock angegebene Meßbereich ohne Bedeutung und beträgt immer 0-5V. 8- und 10-Bit-Meßkanäle dürfen zusammen verwendet werden.

Das Betriebssystem erlaubt den freien Zugriff auf alle Adressen des Prozessor-internen RAMs, des externen 32K-RAMs und der Spezialfunktionsregister des Prozessors. Es können Programme nachgeladen und gestartet werden. Außerdem können spezielle Speicherkonfigurationen eingeschaltet werden, die sich nur durch Ausschalten wieder verlassen lassen.

SIOS kann durch falsche Kommandos zum Absturz gebracht werden. Empfängt das Interface zufällige Kommandos, dann könnte eine unkontrollierte Aktion eintreten und den weiteren Datenaustausch blockieren. Mögliche Fehlerquellen sind z.B.:

- Die Baudrate wird unkontrolliert verändert.
- Es erfolgt ein ungewollter Programmaufruf, z.B. an einer nicht definierten Adresse.
- Prozessor-Register oder Speicherstellen des internen RAM werden verändert.
- Der geschützte RAM-Bereich 0000h...0025h mit seinen Interruptvektoren (als Programmspeicher gespiegelt bei 8000h...8035h) wird verändert.
- Vor dem Ende einer Messung werden erneute Kommandos geschickt. Das erste Byte unterbricht dann die Messung, das zweite Byte wird als neues Kommando interpretiert und führt zu einer zufälligen Aktion.

## 2 SIOS-Kommandos im Basismodus:

### 2.1 Spezielle SIOS-Optionen:

Senden	Empfangen	Funktion
00h		Interrupt für laufende Messungen und Programme
01h	10	Versionsnummer 1.0 abfragen
02h	15	Datum: Tag
03h	8	Datum: Monat
04h	96	Datum: Jahr
05h, Byte		Baudrate ändern: FFh=57,6kBd FDh=19,2kBd, FAh=9,6kBd usw.
08h, Byte		Meßbereich Kanal 0 ändern
...		00h=5V 80h=2,5V, 40h=1,25V
0Bh, Byte		Meßbereich Kanal 3 ändern

### 2.2 Portausgaben:

Senden	Empfangen	Funktion
-----		



1, Byte		1. Byte
	1	
1, Byte		2. Byte
...		
	1	
0		Ende
61h HiAdr, LoAdr		Programmstart
62h HiAdr, LoAdr, Byte		RAM-Adresse beschreiben
63h HiAdr, LoAdr	Byte	RAM-Adresse auslesen
64h Adr, Byte		Register beschreiben
65h Adr	Byte	Register auslesen
66h SFR, Byte		SFR beschreiben
67h SFR	Byte	SFR auslesen
68h HiAdr, LoAdr		Speicherbereich auslesen
1	Byte	1. Byte
1	Byte	2. Byte
...		
0		Ende

## 2.8 Serienmessungen:

Senden	Empfangen	Funktion
70h, Anzahl-High, Anzahl-low Kanal 1, 2, 3, 4 Meßbereiche Kanal 0...3 Zeit-high, Zeit-low Zeitbasis Optionen		Serie definieren "0" = aus, 129,130,131,132 f. 10-Bit Standard: 0,0,128,128 f. 5V/2,5V Meßzeit = Zeit + Anzahl Kanäle 0=0,1ms,1=0,2ms, 2=1ms, 3=10ms Bit0: D0 zeigt Messung Bit1: D1 zeigt Meßpunkte Bit2: Rampe an Analogausgang A Bit3: Hochzählen an Ausgängen
71h	1	Start der Messung, 1=beendet
72h		Meßwertspeicher auslesen
	Byte	1. Byte
1	Byte	2. Byte
...		
0		Ende
73h Byte	1	Start mit Triggerung, pos. an A0
74h Byte	1	Start mit Triggerung, neg. an A0
75h Byte	1	Start mit Triggerung, pos. an A1
76h Byte	1	Start mit Triggerung, neg. an A1
77h	1	Start mit Triggerung, pos. an D0
78h	1	Start mit Triggerung, neg. an D0

## 3 Compactbox-Kommandos:

SIOS enthält aus Gründen der Kompatibilität den Befehlssatz der alten Compactbox bzw. der RS-Box. Alle alten Programme können daher weiter verwendet werden.

Senden	Empfangen	Funktion
51h Byte		Digitale Ausgabe (81)
B0h		Analogeingang-Init (176)
D2h	Byte	Messung, Kanäle abwechselnd (210)
D3h	Byte	Digitale Eingabe (211)

#### 4 CompuLAB-Emulation:

Senden	Empfangen	Funktion
61h, 10h, 00h		Starten des CompuLAB-Modus

Die Emulation wird als Aufruf eines Programms im EPROM bei Adresse 0100h gestartet. Es gibt keine Rückkehr außer durch Auschalten. Der Interrupt ist gesperrt. Das SIOS kennt nun nur noch die alten RS-Box-Kommandos (81, 176, 210 und 211) sowie die CamFace-Kommandos (64, 72, 60 und 58). Außerdem wird die Interface-Kennung "201" wie beim CompuLab ausgegeben.

Senden	Empfangen	Funktion
51h Byte		Digitale Ausgabe (81)
B0h		Analogeingang-Init (176)
D2h	Byte	Messung, Kanäle abwechselnd (210)
D3h	Byte	Digitale Eingabe (211)
48h Byte		Digitale Ausgabe (72)
3Ch	Byte	Messung, Kanal 0 (60)
3Ah	Byte	Messung, Kanal 1 (58)
D3h	Byte	Digitale Eingabe (211)
01h	201	Identifikation: CompuLAB

Im Normalfall können die alten Compactbox-Kommandos direkt aus dem Standard-Modus aufgerufen werden. Nur die Interface-Kennung und der zweite Befehlssatz des CompuLAB unterscheiden die Emulation vom Basismodus.

## 5 Speicherbereiche und Betriebsmodi

Das SIOS verfügt über vier unterschiedliche Betriebsmodi mit jeweils eigenen Speichermodellen. Nach dem Einschalten befindet es sich im Modus 0 mit dem EPROM im Adreßbereich 0000h bis 7FFFh und dem RAM als Programmspeicher im Bereich 8000h bis FFFFh, gespiegelt als Datenspeicher im Bereich 0000h bis 7FFFh.

0000h - 07FFh    Programmbereich 2K (läuft bei 8000h-87FFh)  
                   800Bh...800Dh und 8023h...8025h reserviert für  
                   Interrupt-Vektoren  
 0800h - 7FFFh    Datenbereich 30K

Der gesamte Datenbereich erscheint bei 8000h - FFFFh als Programmbereich gespiegelt.

Alle Prozessor-Interrupts werden in den RAM-Bereich ab 8000h gelenkt. Das System verwendet den Timer0 (800Bh) und die serielle Schnittstelle (8023h), wobei die Vektoren in den EPROM-Bereich zurückgelenkt werden. Der Anwender darf deshalb die RAM-Bereiche 000Bh-000Ch und 0023h-0025h nicht verändern, außer er möchte die Interrupts für eigene Zwecke einsetzen.

SIOS kennt insgesamt vier verschiedene Betriebsmodi, die zum Entwicklungssystem ES535 kompatibel sind. Es können deshalb Programme in der jeweils passenden Speicheraufteilung nachgeladen werden. Die Umschaltung in die Betriebsmodi 1 bis 3 erfolgt durch Programmsprünge zu den Adressen 7400h, 7800h und 7C00h. Eine Rückkehr in den Modus 0 durch Software ist nicht vorgesehen. Die Umschaltung bleibt gültig, bis SIOS neu gestartet wird. Im Normalfall wird spezielle Software in das RAM geladen, um dann den gewünschten Modus einzuschalten. Dabei ist darauf zu achten, daß beim Laden die Interrupt-Vektoren bei 800Bh und 8023h nicht zerstört werden.

Die Umschaltung in höhere Betriebsmodi erfolgt durch Programmsprünge zu besonderen Adressen: 7400h für Modus 1, 7800h für Modus 2 und 7C00h für Modus 3. Modus 1 und 2 dienen ausschließlich dazu, Programme in den Adreßbereich ab 0000h zu laden und zu starten. Das RAM liegt als Datenspeicher jeweils bei Adresse 0. Während allerdings im Modus 1 das gesamte RAM zugleich als Programm- und Datenspeicher im Bereich 0000h bis 7FFFh angesprochen werden kann, ist es im Modus 2 in zwei 16K-Bereiche aufgeteilt, die als Programm- und Datenspeicher im Bereich 0000h bis 3FFFh parallelliegen und jeweils im Bereich 4000h bis 7FFFh gespiegelt werden.

Modus 3 schaltet ein alternatives Betriebssystem in obereren 16-K-Bereich des EPROMs ein. Dieser Bereich wird derzeit nicht benutzt und ist für künftige Erweiterungen vorgesehen. Das RAM steht im Bereich 0000h bis 7FFFh als reiner Datenspeicher zur Verfügung.

Die folgende Übersicht zeigt noch einmal alle vier Betriebsmodi und die zugehörige Speicheraufteilung:

Modus 0:

Programmspeicher	Datenspeicher	I/O-Bereich
EPROM 0...32K	RAM 0...32K	
RAM 32K...64K (gespiegelt)		32K...64K

Modus 1 (Umschaltung: 7400h):

Programmspeicher	Datenspeicher	I/O-Bereich
RAM (untere Hälfte) 0...16K gespiegelt 16K...32K	RAM (obere Hälfte) 0...16K gespiegelt 16K...32K	32K...64K

Modus 2 (Umschaltung: 7800h):

Programmspeicher	Datenspeicher	I/O-Bereich
------------------	---------------	-------------

RAM	RAM
0...32K	0...32K
-----	
32K...64K	
-----	

Modus 3 (Umschaltung: 7C00h):

Programmspeicher	Datenspeicher	I/O-Bereich
-----		
EPROM	RAM	
16...32K, 0...16K	0...32K	
-----		
32K...64K		
-----		

Die übrigen Ressourcen des Prozessors 80C535 sind wie folgt eingesetzt:

P1 = Digitale Eingänge,  
P4 = Digitale Ausgänge  
P5 = Daten zum DA-Wandler, P3.2...P3.4 = A0,A1,/CS für DA-Wandler  
P3.5 bildet die digitale Datenleitung am Pin 6 der Sensoranschlüsse C/D.  
P4.2 und P4.3 liegen jeweils direkt am Pin 5 der Sensoranschlüsse C/D.

## 6 Einsatz des Makrocompilers MC

Software-Entwickler können den Makrocompiler MC einsetzen, um spezielle Software für das SIOS zu schreiben. MC übersetzt einen Quelltext in ein lauffähiges Maschinenprogramm für beliebige Speicherbereiche. Das SIOS-Betriebssystem wurde komplett mit MC entwickelt. Typische Einsatzbereiche für zusätzlich nachgeladene Software sind:

- Autonom lauffähige Steuerprogramme
- Zeitkritische Programme wie z.B. extrem schnelle Datenerfassung
- Interruptgesteuerte Hintergrundprogramme, z.B. für Signalgeneratoren
- Erweiterungen des Betriebssystems

Um MC-Programme zu schreiben, sind die folgenden Einstellungen im Option-Menü des Makrocompilers einzutragen:

COM	2
Baud	19200
Prozessor	80535
Startseite	129
Startadresse	51
Tabelle	MCSIOS.TAB
Helpfile	MCSIOS.HLP
Download-Protokoll	2
Download-Delay	0

Die Startseite 129 ist zu wählen, um die Interruptvektoren auf der ersten RAM-Seite nicht zu verändern.

Besondere MC-Befehle für SIOS:

RdPORT, WrPORT	Portzugriffe
RdAd	AD-Wandler 8-Bit
RdAd10	AD-Wandler 10-Bit, lowbyte in B
WrDA	DA-Wandler, Kanal in B
WrCOM, RdCOM	Serielle Schnittstelle
Delay	Reitaster-Steuerung
Time0.1ms, Time0.2ms, Time1ms, Time10ms	Zeitbasis für Delay (default: 1ms)

Das folgende Beispielprogramm demonstriert die Verwendung der analogen Ein- und Ausgänge:

```
;Ausgabe der Spannung am Analogeingang A über Analogausgang A
;Invertierte Ausgabe über Analogausgang B
;Aus einem Einphasensignal wird ein Gegentaktsignal gebildet
```

Procedure AnalogRechnen

```
RdAD 0          ;Analogeingan 0 abfragen (8 Bit)
WrMem 1         ;Ergebnis zwischenspeichern
B 0            ;Analogausgang 0
WrDA           ;Analogausgabe Kanal 0
RdMem 1        ;Ergebnis wiederherstellen
NOT            ;Ergebnis invertieren
B 1            ;Analaogausgang 1
WrDA           ;Analogausgabe Kanal 1
EndProc
```

Begin

Loop AnalogRechnen

End

Mit MC können Hintergrundprogramme geschrieben werden, die auf Prozessor-Interrupts reagieren. Das folgende Beispiel demonstriert die Timer-gesteuerte Ausgabe über die digitalen Ausgänge. Das Programm läuft im Hintergrund neben dem Betriebssystem. Anwenderprogramme wie z.B. Do-it für SIOS können gleichzeitig alle SIOS-Kommandos im Vordergrund verwenden. Bei der Programmierung müssen verschiedene Einzelheiten beachtet werden:

- Der Timer-Interrupt darf erst zur Laufzeit von 800Bh auf die eigene Interrupt-Prozedur umgelenkt werden.
- Der Akku A muß gesichert und wiederhergestellt werden.
- Am Ende der eigenen Interrupt-Prozedur muß ein Sprung auf die Timer-Interruptprozedur des Betriebssystems bei 0071h stehen.
- Es dürfen nur die MEM-Speicher 30...47 benutzt werden
- Das Hauptprogramm muß in die Interpreterschleife des SIOS-Betriebssystems bei 07E7h zurückverzweigen.

```
;Interruprgesteuerte Portausgaben im Hintergrund
```

CodeAdress 81h 10h

Procedure TimerInterrupt

```
Inline C0,E0          ;Push Akku
```

```

RdMem 30          ;Zwischenspeicher
A+1
WrMem 30
WrPort           ;Portausgabe
Inline D0,E0     ;Pop Akku
Inline 02,00,71 ;Ljmp 0071h; Timer-Routine
EndInterrupt

Begin
T0.2ms          ;Aufruf alle 0,2ms
WrTCON 0100000b ;Timer 0 stoppen
DPTR 00h 0Bh    ;Daten bei 000B = Programmcode bei 800B
Wr@DPTR 02h     ;02 81 10, Sprung auf Prozedur "TimerInterrupt
DPTR 00h 0Ch
Wr@DPTR 81h
DPTR 00h 0Dh
Wr@DPTR 10h
WrTCON 0101000b ;Timer 0 starten
Inline 02,07,E7 ;ljmp 07E7h, Rücksprung zur Interpreterschleife
End

```