

"SIO am LC-80ex für Einsteiger"

Hardware

Die SIO hat (wie auch die besser bekannte PIO) zwei Kanäle sowie jeweils zugehörige Steuer- und Datenregister und ist per Chipauswahl-Signal selektierbar. Aufbau und Verdrahtung des LC-80ex bestimmen die Adressierung der SIO :

Kanalauswahl	Signal A/B	=> an Adressleitung A0
Steuer-/Daten-Registerauswahl	Signal C/D	=> an Adressleitung A1
Chipauswahl	Signal /CE	=> an Adressleitung A5

Damit ergeben sich folgende Peripherieadressen (Mehrdeutigkeiten durch unvollständige Dekodierung weggelassen):

Datenregister Kanal A	0DCh
Datenregister Kanal B	0DDh
Steuerregister Kanal A	0DEh
Steuerregister Kanal B	0DFh

Wichtig ist auch die Einstellung des Systemtaktes mit JP 9:

3686,4 kHz

1843,2 kHz

921,6 kHz **Vorzugseinstellung Originaltakt**

Den baudratenbestimmenden Takt erzeugt die CTC (Peripherieadresse 0ECh...) aus dem Systemtakt. Hier wird beispielhaft Kanal 0 der CTC benutzt. Dazu ist **JP 12 entsprechend zu setzen**.

Alle Signalleitungen der SIO sind bei Bedarf per Steckleiste erreichbar. Die 9polige SUB-D-**Buchse 1** enthält die Signale von Port B (RxD, TxD, RTS, CTS) mit RS-232-Pegel. Die Buchse hat eine Standardbelegung (Pin2 = RxD, Pin 3 = TxD,...). Damit muss für den Anschluss an einen PC ein **Nullmodemkabel** verwendet werden.

Software

Für einen Einstieg in die recht vielfältigen und komplexen Funktionsweisen der SIO wird das einfachste Modell gewählt:

- nur 1 Kanal (B) betrachtet
- asynchroner Modus
- Pollingmethode

Das "Steuerregister" der SIO besteht in der Praxis aus mehreren Einzel-Registern. Konkret sind dies für jeden Kanal je drei Lese-Register (RR0...2) und acht Schreib-Register (WR0...7). Diese Register müssen nun geeignet beschrieben und gelesen werden, um die SIO zur gewünschten Funktion zu bringen. Erst dann können über die Datenregister die eigentlichen Daten byteweise gesendet/empfangen werden.

Da von der CPU immer nur auf eine Adresse (z.B. 0DFh) zugegriffen werden kann, benötigen sowohl das Lesen als auch Schreiben eines Einzelregisters (Ausnahme Register 0) zwei aufeinanderfolgende Befehle:

- Beim ersten Zugriff ist u.a. ein "Zeiger"-Byte im Register WR0 des Kanals zu setzen. Die unteren 3 Bit des geschriebenen Wortes adressieren dabei das gewünschte Register.
- Im zweiten Zugriff gelangen (von der SIO organisiert) die von der CPU gesendeten Daten in das so ausgewählte Register bzw. werden von dort gelesen.

Auf die Darstellung der Register-Tabellen mit den kompletten Inhalten und Möglichkeiten wird hier verzichtet, es kommen nur ausgewählte Werte zur Anwendung/Erläuterung.

Beispiel: 1200 Baud, 8 Datenbits, 1 Stopp-Bit, keine Parität, kein Handshake

Vorbereitung Empfänger+Sender:

INIT:	LD A, 18h	18h = 000 <u>II</u> 000
	OUT (0DFh), A	+--- Register WR0 +----- Reset Kanal
	NOP	kurze Pause
	LD A, 04h	Register WR4 auswählen
	OUT (0DFh), A	(=gemeinsame Parameter für Senden + Empfang)
	LD A, 44h	44h = <u>0I</u> 00 <u>0I</u> 00
		+--- keine Parität +----- 1 Stoppbit +----- Vorteiler x16
	OUT (0DFh), A	Register WR4 schreiben
	LD A, 03h	Register WR3 auswählen
	OUT (0DFh), A	(=Empfangsparameter und Steuerbits)
	LD A, 0C0h	0C0h = <u>II</u> 00 0000
		Empfänger auf <u>8 Bit</u> einstellen
	OUT (0DFh), A	Register WR3 schreiben
	LD A, 05h	Register WR5 auswählen
	OUT (0DFh), A	(=Sendeparameter und Steuerbits)
	LD A, 60h	60h = 0 <u>II</u> 0 0000
		Sender auf <u>8 Bit</u> einstellen
	OUT (0DFh), A	Register WR5 schreiben
	RET	

Baudrateneinstellung

Für die serielle Übertragung benötigt die SIO bei Kanal B am Pin27 (RxTxCB) einen Takt. Woher dieser kommt, ist zweitrangig. Dieser Takt bestimmt aber direkt die Baudrate!

Bei der Berechnung der nötigen Taktfrequenz ist zu beachten, dass diese vom **oben gewählten SIO-Vorteiler** abhängig ist. Allgemein wird meist der Teilerwert von **x16** verwendet, obwohl in der asynchronen Betriebsart alle verfügbaren Werte (x1, x16, x32, x64)¹ zulässig sind. Bei x16 gilt: Der SIO-Schiebetakt an RxTxCB muss 16x höher als die erstrebte Baudrate sein.

Für gängige standardisierte Baudraten sind damit folgende Taktfrequenzen an RxTxCB nötig:

- 9600 Baud: 153,6 kHz.
- 4800 Baud: 76,8 kHz
- 2400 Baud: 38,4 kHz
- 1200 Baud: 19,2 kHz

Dieser Takt kann einfach aus dem Systemtakt gewonnen werden, wenn dieser eine "baudfreundliche" Frequenz aufweist. Das Herunterteilen erfolgt dann günstigerweise mit der meist sowieso vorhandenen CTC.

1 Ein SIO-Vorteilerwert "x1" ermöglicht zwar theoretisch mit gleichem Systemtakt höhere Baudraten, erfordert aber andere Randbedingungen (lt. [1]: "...Wenn die x1-Taktrate ausgewählt worden ist, muss die Bitsynchronisation extern erfolgen...") ??? Vermutl. müsste dazu Rx-C-Takt des Empfängers aus dem Bitstrom an Rx-D gewonnen werden (z.B. mit PLL). Oder man verwendet eine zusätzliche Leitung (Tx-C-Takt des Senders)!?

Vorbereitung der CTC:

Für den Zweck der Bereitstellung Schiebetaktes aus dem Systemtakt ist die CTC wie folgt zu programmieren:

PCTC:	LD A, 07h	kein INT, Zeitgebermode, Vorteiler=16, ZK folgt, Kanalreset => Kanal 0
	OUT (0ECH), A	
	LD A, 03h	Zeitkonstante ZK=3 an CTC Kanal 0
	OUT (0ECH), A	
	RET	

Die CTC liefert nun am Ausgang CTO0 den gewünschten RxTxCB-Takt. Mit einem Systemtakt von 921,6 kHz ergibt sich eine Taktfrequenz von 19,2 kHz und zusammen mit dem oben gewählten SIO-Teiler x16 eine **Baudrate von 1200**.

Inwiefern höhere Baudraten am LC-80ex mit der originalen geringen Taktfrequenz von 921,6 kHz nutzbar/sinnvoll sind, bleibt hier offen und ist eigenen Experimenten überlassen.

Für andere Baudraten sind Systemtakt und/oder CTC-Programmierung entsprechend zu ändern. Die Berechnungsvorschrift dazu:

$$\text{Baudrate [Bits/s]} = \frac{\text{Systemtakt [Hz]}}{\text{Vorteiler CTC} * \text{Zeitkonstante CTC} * \text{Vorteiler SIO}}$$

Da es mehrere Unbekannte in der Gleichung gibt, muss man unter Beachtung der möglichen Werte der einzelnen Größen ggf. etwas probieren.

Lässt man die CTC nicht im Zeitgebermodus laufen sondern verwendet den Zählermodus, so kann der Faktor "Vorteiler CTC" entfallen. Damit sind höhere Baudraten möglich. Allerdings muss dann der zu teilende Takt der CTC extern zugeführt werden (Brücke Systemtakt an JP9 <=> CTC-Eingang Kanal 0 an JP12).

Kanal B ist nun für Senden und Empfang vorbereitet. Die INIT-Routine sowie die Programmierung der CTC sind nur einmal am Programmbeginn nötig (sofern diese Einstellwerte nicht geändert werden müssen).

Empfangsroutine:

Damit die SIO tatsächlich auch empfangen kann, müssen wir den Wert 0C1h ins Register WR3 schreiben. Theoretisch ist nur **Bit0** dafür zuständig, aber die vorher ins Register WR3 geschriebenen Werte in den übrigen Bits dürfen nicht überschrieben werden. Der Empfänger ist nun freigegeben.

Mit der Pollingmethode erfolgt der Empfang so:

- (1) Schreiben einer 0 ins Register WR0, anschließendes Auslesen des Statusregisters RR0
- (2) Testen von Bit0 (= ob ein Zeichen empfangen wurde)
 - wurden keine Daten empfangen, weiter mit (1)
 - liegen Daten vor, dann das empfangene Byte aus dem Datenregister B lesen und verarbeiten, weiter mit (1)

EMPF:	LD A, 03h	Register WR3 auswählen
	OUT (0DFh), A	
	LD A, 0C1h	0C1h = II00 000I
	OUT (0DFh), A	<u>Empfang freigeben</u> (übrige Bits wie oben eingestellt lassen!)
	NOP	warten, bis SIO fertig
	NOP	
	LD HL, 2000h	Anfangsadresse für Ablage
E1:	SUB A	Register 0 auswählen
	OUT (0DFh), A	
	IN A, (0DFh)	Lesen Register RR0 = Status (bit0=I: empfangenes Zeichen im Puffer)
	RRA	rotieren bit0->carry
	JP NC, E3	Carry gesetzt? nein=> keine Daten => M5
	IN A, (0DDh)	Carry war gesetzt, nun Daten von SIO lesen!
E2:	LD (HL), A	Verarbeitung des Zeichens: Ablage im RAM
	INC HL	nächster Ablageplatz
	JR E1	weiter mit Polling
E3:	...	andere Aufgaben, wenn kein Zeichen empfangen,
	...	z.B. Test auf Timeout
	JP E1	weiter mit Polling

Der bedingte Sprung nach E3 erlaubt die Ausführung anderer Aufgaben, ehe wieder zu Adresse E1 gesprungen und mit dem Polling fortgefahren wird. Wurde ein Zeichen empfangen, so muss der Programmierer das nun ab Adresse E2 verarbeiten.

Bei der "Verarbeitung" der Daten sollte man sich (sofern möglich und sinnvoll) zunächst auf ein einfaches Ablegen im Speicher beschränken. Erst nach Beendigung des Empfangs werden sie weiter "behandelt". Damit geht man zeitkritischen Problemen aus dem Wege...

Für ein ggf. nötiges Sperren des Empfangs ist der Wert 0C0h (Bit0=0) in Register WR3 zu schreiben.

Ein Abbruch der Polling-Empfangsschleife kann am LC-80ex mittels RESET (Rücksprung in den Monitor) erfolgen. Der bis dahin im RAM abgelegte Inhalt bleibt dabei dort erhalten. Ein so per SIO vom PC auf 2000h geladenes Programm kann dann mit ADR 2000 EX gestartet werden.

Senderoutine:

Auch das Senden muss zunächst erst freigegeben werden, indem der Wert **68h** ins Register WR5 geschrieben wird. Ehe nun ein Byte gesendet werden kann, ist zu prüfen, ob der Sendepuffer leer ist. Das erfolgt durch Adressieren und Lesen von Register RR0. Diesmal ist Bit2 zu prüfen. Wenn high, dann weiter prüfen, bis Bit2 = low wird. Erst dann kann ein Byte gesendet werden.

SEND:	LD A, 05	Register WR5 auswählen
	OUT (0DFh), A	
	LD A, 68h	68h = 0II0 I000
	OUT (0DFh), A	<u>Senden freigeben</u>
S1:	SUB A	Register 0 auswählen
	OUT (0DFh), A	
	IN A, (0DFh)	Register RR0 lesen (= Status holen)
	BIT 2, A	Test: Sendepuffer geleert?
	JR Z, S1	nein => weiter testen
	LD A, D	ja => zu sendendes Byte (liegt in D) laden
	OUT (0DDh), A	Daten senden
	RET	

Man legt also das auszugebende Byte in CPU-Register D und führt einen CALL SEND aus. Streng genommen ist es aber nicht nötig, jedes mal erst den Sender freizugeben, wenn ein Zeichen gesendet werden soll. Es sollte in der Folge also auch ein CALL S1 genügen.

Für ein ggf. nötiges Sperren des Sendens ist der Wert **68h (Bit3=0)** in Register WR5 zu schreiben.

Praktischer Einsatz

1. Mit einem Nullmodemkabel (!) den LC-80ex an einen COM-Port des PC anschließen.
2. Am PC eine Terminalsoftware starten (z.B. HTerm).
3. Am PC die Terminalsoftware auf korrekten COM-Port und Baudrate einstellen
4. Am LC-80ex "Senden" starten => am PC müssen die gesendeten Zeichen erscheinen.
5. Am LC-80ex "Empfang" starten" => die am PC gesendeten Zeichen müssen am LC-80ex erscheinen.

Anmerkungen

- SIO heißt nichts anderes als "serial input output". Oft wird jedoch auch der Schaltkreis selbst so benannt. Im deutschsprachigen Raum stellt sich dann die Frage, ob nun "männlich oder weiblich" :-). Bei meinen Recherchen im Internet fand ich oft "**die** SIO". Ich verwende das hier (analog zu CPU, PIO, CTC) auch so...
- Die o.a. Befehlsfolgen wurden nur zur Erläuterung so detailliert aufgeführt. In der Praxis kann man auch z.B. mit Tabellen und OTIR-Anweisungen arbeiten.

Quellen

- [1] RFT: "Mikroprozessorsystem der II. Leistungsklasse", Technische Beschreibung SIO U856
- [2] Manual zum V.24-Modul des KC85
- [3] http://www.sbprojects.net/projects/izabella/html/z80_sio_programming_hints.html