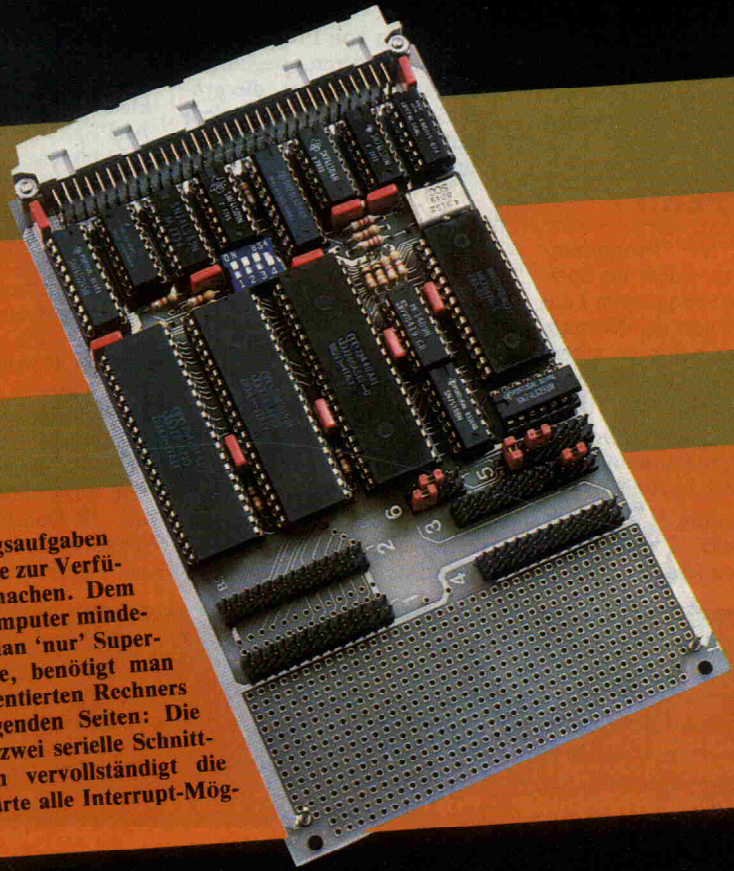


Die I/O-Karte für den ECB-Bus

Datenverkehr

Georg Umbach

Spätestens, wenn man mit seinem Rechner Steuerungsaufgaben bewältigen möchte, muß man sich Gedanken über die zur Verfügung stehende Anzahl von Input-/Output-Ports machen. Dem Gesetz der konstanten Bosheit folgend, wird der Computer mindestens einen Port zu wenig haben. Aber auch wenn man 'nur' Super-Tape auf seinem Rechner implementieren möchte, benötigt man einen freien I/O-Port. Besitzer eines ECB-Bus-orientierten Rechners finden die Lösung ihres Problems auf den folgenden Seiten: Die c't-I/O-Karte bietet vier parallele 8-Bit-Ports und zwei serielle Schnittstellen (Duplex). Ein Zähler-/Zeitgeber-Baustein vervollständigt die Karte. Und für spezielle Anwendungen hält die Karte alle Interrupt-Möglichkeiten der Z80-Bausteine offen.



Die c't-I/O-Karte ist zum Betrieb in Z80-Systemen mit ECB-Bus vorgesehen. Als Schnittstellen-Bausteine stehen ein Z80-CTC (Counter/Timer Circuit — Zähler-/Zeitgeber-Schaltkreis), eine Z80-SIO(0) (Serial Input/Output — serielle Ein-/Ausgabe) und zwei Z80-PIOs (Programmable Input/Output — programmierbare Ein-/Ausgabe) zur Verfügung. Man kann statt der Z80-SIO auch einen Z80-DART (Dual Asynchron Receiver/Transmitter — Zweifacher, asynchroner Empfänger/Sender) verwenden, der preisgünstiger als die SIO ist und für den Betrieb einer seriellen Schnittstelle völlig ausreicht, wenn keine blockweise synchrone Datenübertragung stattfinden soll. Alle I/O-Bausteine können im Interrupt-Betrieb arbeiten, auch wenn man die Karte nur teilweise mit den Z80-Chips bestückt.

Außerdem ist auf der Karte ein

Quarz-Oszillator mit dem notwendigen Frequenzteiler zur Baudratenzeugung untergebracht. Somit kann zum Beispiel der Timer mit den unterschiedlichsten Taktfrequenzen betrieben werden.

Alle Leitungen zum ECB-Bus sind gepuffert, so daß nur eine geringe Belastung des Systembusses stattfindet. Wer auf der Karte zusätzliche Schaltungen aufbauen will, wie zum Beispiel Treiber oder Wandler, hat auf einem Lochrasterfeld von 3,5x9 Zentimetern genügend Platz.

Der CTC-Baustein kann als Timer, gesteuert durch den Systemtakt, oder als extern getriggert Zähler verwendet werden. Alle seine Ein-/Ausgänge sind auf den 26poligen Pfostenstecker J4 geführt. Soll die serielle Schnittstelle (SIO(0)/DART) mit 'exotischen' Baudraten betrieben werden, zum Beispiel mit 45,5 Baud für Funkfernschreiber-

trieb, so kann man den CTC0 und CTC1 für die Erzeugung dieser Baudraten verwenden.

Der Teil B der SIO(0)/DART ist als serielle Schnittstelle ausgelegt. Die Ausgangspegel der SIO werden von einem Baustein 75188 (IC10) in V24-Pegel umgesetzt. Dazu müssen allerdings am Systembus die Spannungen +12V und -12V anliegen, was meistens auch der Fall ist. Stellt das verwendete System diese Spannungen nicht bereit, kann man auf dem Verdrahtungsfeld einen Wandler für +12/-12V aufbauen. Die Umsetzung der V24-Empfangssignale geschieht in dem Baustein 75189 (IC11). Soll die Schnittstelle anstelle von V24-Pegeln TTL-Signale erzeugen und verarbeiten, so ist dies mit einer kleinen Platinenänderung und der Verwendung anderer ICs möglich. Aber dazu später mehr.

Der Sende- und Empfangstakt für die SIO(0)/DART wird mit einem Quarz-Oszillator (4,9152

MHz) erzeugt. Die Frequenz wird binär in einem Zähler 74LS393 (IC9) heruntergeteilt, so daß sie als Takt zur Verfügung steht. Sofern man die Eigenschaft der SIO(0)/DART ausnutzt, für einen seriellen 'Schritt' den 1-, 16- oder 64fachen Takt zu verarbeiten, kann man alle gängigen Baudraten bis hinunter zu 300 Baud erzeugen. Sollte ein von der Frequenz her geeigneter Systemtakt zur Verfügung stehen, so kann der Oszillator entfallen, da alle weiteren Signale dann aus dem Systemtakt abgeleitet werden können.

Alle Ein-/Ausgänge des Teils A der SIO(0)/DART sind auf den Pfostenstecker J4 geführt. Hier kann man sie ungepuffert für besondere Anwendungen (z. B. synchrone Datenübertragung) abnehmen. Damit bei der Verwendung dieses Teils der SIO(0)/DART hier die höhere Interrupt-Priorität liegt, wurde der Teil A des ICs benutzt. Ist er inaktiv, so wird Teil B der SIO(0)/DART interruptmäßig nicht benachteiligt.

Die Ausgänge der zwei Z80-PIOs sind auf 26polige Pfostenstecker geführt (PIO1-J1, PIO2-J2). Um die Karte universell zu halten, sind die PIO-Leitungen ungepuffert.

Von IC zu IC

Die vier Z80-I/O-Bausteine belegen zusammen 16 Port-Adressen. Die Auswahl der einzelnen Ports in den I/O-Bausteinen erfolgt durch die gepufferten Adressen A0/A1. Die Selektierung der einzelnen Bausteine mit je 4 Ports geschieht über die Adressen A3/A2 und den Dekoder 74LS138 (IC3). Mit den DIL-Schaltern (S1—S4) kann man die obere Adressenhälfte (BASE) einstellen. Es ergibt sich eine Adressenverteilung, wie sie Tabelle 1 zeigt.

Basis-Adresse I/O-Karte x0h (DIL-Schalter)

CTC 0	x0h
CTC 1	x1h
CTC 2	x2h
CTC 3	x3h
SIO A Daten	x4h
SIO A Control	x5h
SIO B Daten	x6h
SIO B Control	x7h
PIO 1 A Daten	x8h
PIO 1 B Daten	x9h
PIO 1 A Control	xAh
PIO 1 B Control	xBh
PIO 2 A Daten	xCh
PIO 2 B Daten	xDh
PIO 2 A Control	xEh
PIO 2 B Control	xFh

Tabelle 1. Verteilung der Port-adressen

Wird nun ein Port auf der Karte durch einen I/O-Zugriff der CPU selektiert (M1=1, A7-A4=BASE, IORQ=0), so steht am Komparator 74LS85 (IC4), Pin 6 ein Signal mit dem logischen Pegel 1. Ist gleichzeitig das RD-Signal aktiv, so entsteht durch die Inverter 74LS04 (IC15), das NAND-Gatter 74LS20 (IC12) und das AND-Gatter 74LS08 (IC14) ein Signal mit Pegel logisch 0 für die Richtungsumschaltung des Datenbustreibers 74LS245 (IC1). Das IC ist immer aktiviert und sendet im Normalfall alle Daten, die auf dem Systembus liegen, in den Kartenbus. Somit können alle Z80-I/O-Bausteine den Systembus 'abhören' und

den Befehl RETI (Return from Interrupt) dekodieren.

Die Richtung des Treibers wird nur bei einem I/O-Read oder dem Einlesen eines Interrupt-Vektors umgeschaltet. Dieser Fall tritt ein, wenn ein I/O-Baustein einen Interrupt anfordert (INT=0). Dabei legt dieser Baustein seinen Ausgang IEO auf logisch 0. Durch die 'wired AND'-Verknüpfung der 'Daisy-Chain'-Leitungen liegt auch das Signal IEO der Karte auf logisch 0. Wenn die CPU den geforderten Interrupt annimmt, sendet sie M1 und IORQ mit dem Pegel logisch 0. Liegt das Signal IEI der Karte auf logisch 1 (kein Interrupt einer höher priorisierten Karte), so wird jetzt aus den Signalen M1, IORQ, IEI und IEO über die Inverter 74LS04 (IC15), das NAND 74LS20 (IC12) und das AND 74LS08 (IC14) ein Richtungssignal für den Datenbustreiber mit dem logischen Pegel 0 erzeugt. Gleichzeitig legt der betreffende I/O-Baustein seinen Interrupt-(INT)-Vektor auf den Datenbus, der jetzt zur CPU gesendet wird.

Die I/O-Bausteine haben beim Interrupt folgende Priorität:
CTC — SIO(0)/DART —
PIO1 — PIO2.

Beim Einschalten des Computers oder beim Reset erzeugt ein AND-Gatter des 74LS08 (IC14) aus den Bussignalen CLR und M1 einen Hardware-Reset für die PIOs, die diesen Befehl aus den Signalen M1 aktiv ohne RD oder IORQ aktiv ableiten.

Im Aufbau

Vor dem Bestücken sollte man die Karte einer optischen Prüfung unterziehen. Eventuell vorhandene Fehler sind jetzt noch relativ einfach zu beheben. Das Beseitigen von Fehlern (zum Beispiel Kurzschlüssen) unter einer IC-Fassung ist nach der Bestückung schwierig oder sogar unmöglich.

Da die Bauteile auf der Karte teilweise sehr dicht beieinander liegen, sollte man prüfen, ob die Stützkondensatoren und die vorhandenen Fassungen für CTC, SIO, PIOs zusammenpassen. Verwendet man als Stützkondensatoren Ausführungen mit einer Stärke von 2,5 mm (z. B. WIMA MKS2 0,1µF/50V) und 'anreihbare' IC-Fassungen, ist der Platz für

die Kondensatoren zwischen den I/O-ICs ausreichend. Verwendet man Fassungen mit größeren Maßen, so ist vor der Bestückung zu prüfen, ob die Kondensatoren nicht durch eine andere Bauform mit günstigeren Maßen ersetzt werden müssen.

Für die Widerstände sollte man Bauformen wählen, die einen Durchmesser unter 2,5 mm haben.

Praxis

Zuerst werden die Widerstände eingesetzt. Dabei sollte man einen Pappstreifen von etwa 8 mm Stärke unter die Widerstände legen. Nach dem Verlöten des Bauteils kann man den Streifen entfernen. Dadurch wird verhindert, daß es zwischen den Kappen der Widerstände und den Leiterbahnen zu Kurzschlüssen kommt. Anschließend bestückt man die Karte mit den IC-Fassungen, dem Quarz, dem DIL-Schalter und den Kondensatoren. Danach folgen die Pfostenstecker und die zwei Lötnägel im Verdrahtungsfeld. Beim Stecker J5 können leicht Verwechslungen auftreten: Der Stecker J5 kommt in die 'eckigen' Lötungen. Zuletzt wird die VG-Steckerleiste montiert. Sie sollte unbedingt vor dem Löten mit der Platine verschraubt werden. Die Lötverbindung steht so nicht unter mechanischer Spannung.

Hat man die Karte soweit bestückt, sollte nochmals eine optische Überprüfung erfolgen (vergessene Lötstellen, Lötbrücken). Ist diese Kontrolle positiv verlaufen, kann man den Widerstand zwischen den Lötnägeln messen. Sind alle DIL-Schalter 'OPEN', muß sich ein unendlich hoher Widerstand ergeben. Andernfalls liegt ein Kurzschluß vor.

Anschließend setzt man alle TTL-ICs in ihre Fassungen. Nach dem Anlegen der Versorgungsspannung von 5V darf die Karte etwa 100mA Strom 'ziehen'. Ergibt sich hierbei kein überhöhter Wert, kann man die I/O-Chips und die V24-ICs bestücken. Nach dem erneuten Anlegen der Betriebsspannung sollte sich eine Gesamtstromaufnahme von rund 300mA ergeben. Eine von diesem Wert wesentlich abwei-

chende Stromaufnahme weist auf einen Fehler hin.

Dann wird mit dem DIL-Schalter die Basis-Adresse der Karte eingestellt und das Testprogramm gestartet. Das Programm muß laufend das Zeichen '0' auf dem Schirm darstellen. Setzt man die Brücke J3/3—5, akzeptiert das Programm Zeichen von der Tastatur, die zwischen den Nullen abgebildet werden.

Die Leitung B7 der PIO 2 ist offen, wodurch ein Interrupt generiert wird, den das Programm als '4' auf dem Schirm darstellt. Berührt man mit dem Finger den Stecker J2/21, so werden (gemischt) das Zeichen '0' und überwiegend die '4' ausgegeben. Um diesen Interrupt 'abzuschalten', berührt man mit der Hand gleichzeitig den 'GND-Lötnagel' und den Stecker J2/21. Dadurch gelangt Masse-Potential auf den PIO-Eingang.

Überprüft man die Karte mit einem Oszilloskop, ist folgendes zu beachten: Beim Betrieb der Karte an langen (z. B. 20 Plätze), nicht terminierten Bussen oder auf langen Extendern können, je nach IC 4 (74LS85), in den CS-Impulsen der I/O-Bausteine 'Spikes' auftreten. Dies hat sich bisher in keinem Betriebszustand negativ bemerkbar gemacht. Im übrigen verschwindet dieser Effekt, sobald man die Karte direkt an dem Bus betreibt und dieser gegebenenfalls terminiert ist.

Modifiziert

Bestückt man die Karte nur teilweise mit den I/O-Bausteinen, muß man die Pins IEI und IEO des fehlenden Bausteins mit einer Brücke verbinden, wenn dieser Chip vor anderen I/O-Bausteinen in der 'Daisy Chain' liegt.

Soll die serielle Schnittstelle mit TTL-Pegel arbeiten und gleichzeitig galvanisch von der Karte getrennt sein (zum Beispiel durch Optokoppler), sind folgende Änderungen durchzuführen:

Senden: Das IC10 (75188) entfällt. Die Leitung zu Pin 14 (+12V) ist aufzutrennen, der freie Pin ist mit +5V zu verbinden. Außerdem muß man die Leitung zu Pin 1 trennen

Interrupt, warum und wie?

Es gibt grundsätzlich zwei Verfahren, um einen Computer auf Ereignisse von außen reagieren zu lassen. Bei dem einen, dem Polling-Verfahren, werden die Rechner-Ports oder die Statusregister der Portbausteine zyklisch abgefragt und ausgewertet. Die Reihenfolge ist per Programm vorgegeben, außerdem läuft die Abfrage (normalerweise) nicht weiter, während der Computer auf ein Ereignis reagiert. So entstehen zwangsläufig 'Überwachungs-lücken', in denen kurze Ereignisse verlorengehen können. Ebenso nimmt die Reaktionszeit des Rechners erheblich zu, wenn mehrere 'bedienungsbedürftige' Ereignisse gleichzeitig auftreten. In diesem Zusammenhang kann auch eine niedrige Taktfrequenz der CPU zu Problemen führen, da dieselbe Bedienroutine dann länger dauert als bei einer hohen Taktfrequenz. Wenn die Reaktionszeit kritisch ist, muß man beim Polling zusätzliche Abfragen in die Bedienroutinen einbauen.

Die andere Methode ist, daß der entsprechende Port einen Interrupt (Unterbrechung des

laufenden Programms) anfordert. Damit ist eine schnelle Reaktion auch auf kürzeste Ereignisse möglich, weil die CPU bei jedem Befehl prüft, ob eine Interrupt-Anforderung vorliegt. Dazu besitzt die Z80-CPU zwei Eingänge:

— Der NMI (Non Maskable Interrupt) hat außer bei einer DMA-Anforderung (die CPU gibt die Kontrolle über den Bus an einen anderen Baustein ab) absoluten Vorrang und kann nicht durch Software gesperrt (maskiert) werden.

— Der INT (maskierbarer Interrupt) ist per Programm steuerbar. Dabei kann der Anwender zwischen drei verschiedenen Betriebsarten wählen:

Mode 0: 8080-Modus
Mode 1: Call nach Adresse 0038h
Mode 2: vektorisierter Interrupt

Im 8080-Modus erwartet die CPU, daß der anfordernde Portbaustein oder Interrupt-Controller während des Interrupt-Acknowledge-Zyklus ('Stattgegeben!'), signalisiert durch IORQ = M1 = logisch 0) einen Maschinenbefehl auf den Datenbus schaltet, den die

CPU dann ganz normal ausführt. Üblicherweise ist dies ein RST-Befehl (Ein-Byte-Call mit 'eingebauter' Adresse) oder ein CALL, um die aktuelle Adresse des unterbrochenen Programms zu retten, es kann aber prinzipiell jeder Befehl sein.

Im 'Mode 1' führt die CPU als Antwort auf einen Interrupt immer einen RST 38h aus.

Im 'Mode 2' liest die CPU ebenfalls während des Interrupt-Acknowledge vom Datenbus, diesmal allerdings exakt ein Byte. Dieser sogenannte Interrupt-Vektor, den wiederum der Port/Controller liefern muß, bildet die niederwertige Hälfte einer Speicheradresse (die oberen acht Bit stehen im I-Register der CPU), unter der die Startadresse für die gewünschte Interrupt-Bedienroutine zu finden ist. Die Bedienroutine wird dann als Unterprogramm aufgerufen. Durch die Adressierung über das I-Register erhält man im Speicher eine Tabelle, die bis zu 128 verschiedene Interrupts verwalten kann. Mit einem anderen Wert im I-Register greift die CPU auf eine andere Adreßtafel zu, so erreicht man andere Reaktionen auf dieselben Inter-

rupts. Die Portbausteine der Z80-Familie sind für diesen Interrupt-Modus ausgelegt.

Gibt es nicht nur eine Interrupt-Quelle im System, muß man dafür sorgen, daß auch bei mehreren Interrupt-Anforderungen gleichzeitig immer nur einer der betreffenden Bausteine seinen Befehl/Vektor auf den Datenbus legen kann. Sonst gibt es einen 'Crash' auf dem Datenbus, und die CPU liest 'wirres Zeug'. Man verteilt daher Prioritäten: Auf jeden Acknowledge-Zyklus reagiert von den Bausteinen, die einen Interrupt angemeldet haben, jeweils der höchstpriorisierte.

Dazu bedient man sich einer Prioritäts-Leitungskette, der 'Interrupt Daisy Chain', für die alle interruptfähigen Ports in Reihe geschaltet werden. Die entsprechende Steuerlogik ist in den Z80-Portbausteinen bereits integriert und wird durch die Anschlüsse IEI (Interrupt Enable Input) und IEO (Interrupt Enable Output) repräsentiert. Im Ruhezustand, das heißt, wenn kein Interrupt angemeldet ist, liegt an den IE-Ausgängen derselbe Pegel wie an den IE-Eingängen.

(-12V) und die Pins 1 und 2 verbinden. Je nach gewünschter Polarität des Ausgangssignals ist für IC10 ein 74..00 oder 74..08 einzusetzen. Als Treiber für einen Optokoppler kann man auch ein IC des Typs 74..38 verwenden.

Empfangen: Das IC11 (75189) entfällt. Es sind jeweils die Pins 1 und 3, 4 und 5, 9 und 10 sowie 12 und 13 miteinander zu verbinden. Je nach gewünschter Polarität des Ausgangssignals kann man für IC11 ein IC des Typs 74..00 oder 74..08 einsetzen. Bei Verwendung von Optokopplern im Empfangsweg sind Standard-TTL-ICs anstelle des 74..00 oder 74..08 einzusetzen.

Soft-Test

Das Testprogramm ist für das Betriebssystem CP/M 2.2 geschrieben, kann jedoch nach entsprechenden Modifikatio-

nen auch auf Rechnern mit anderen Betriebssystemen laufen.

Nach dem Start des Programms werden die Sprünge zu den Unterprogrammen CONSOLE, STATUS, CONSOLE INPUT, CONSOLE OUTPUT aus der BIOS-Sprungliste in das Testprogramm einkopiert. Die Routine CONST prüft, ob ein Zeichen von der Tastatur ansteht. Wenn ja, enthält der Akku den Wert FFh, sonst 00h. Das Programm CONIN wartet auf ein Zeichen von der Tastatur und lädt dann das empfangene Zeichen in den Akku. Die Routine CONOUT sendet das Zeichen aus dem C-Register zum Bildschirm. Der Akku enthält später ebenfalls den Character.

Nach diesen Routinen folgt die Initialisierung der Z80-Portbausteine. Die Unterprogramme INIT und INITX senden eine Bytegruppe mit einem Block-I/O-Befehl zu den Port-

bausteinen. Für jedes I/O-IC gibt es eine Tabelle. Das erste Byte eines Tabellenabschnitts gibt die Anzahl der Bytes an, die gesendet werden sollen. Das zweite Byte ist die Portadresse. Die weiteren Bytes dienen zur Initialisierung des Bausteins. Danach folgt der nächste Tabellenabschnitt. Ist das erste Byte eines neuen Tabellenabschnitts 00, so ist das Ende der für diesen Portbaustein gültigen Tabelle erreicht. Die Bedeutung der einzelnen Bytes in der Tabelle ist dem Kommentar im Listing zu entnehmen.

Nach der Initialisierung der Portbausteine arbeitet das Programm eine Schleife ab. Dadurch wird an den Ausgängen B0 bis B6 der PIO 2 ein Rechtecksignal erzeugt. Anschließend wird im Polling (also direkte Abfrage des Ports ohne Interrupt) geprüft, ob ein Zeichen von der SIO(0)/DART empfangen wurde. Falls ein Zeichen ansteht, so wird es aus

der SIO(0)/DART geholt und zum Bildschirm gesendet.

Im Anschluß daran wird die Tastatur abgefragt und ein eventuell eingegebenes Zeichen abgeholt. Ist dieses Zeichen ein '.', so erfolgt ein Warmboot des Systems. Alle anderen Zeichen werden zur SIO(0)/DART gesendet. Zum Test kann man den seriellen Ausgang des ICs mit dem seriellen Eingang verbinden (Brückenstecker J3 3-5). Das Programm fragt dann die Tastatur ab, sendet und empfängt die Zeichen über die SIO(0)/DART und schickt sie dann zum Bildschirm.

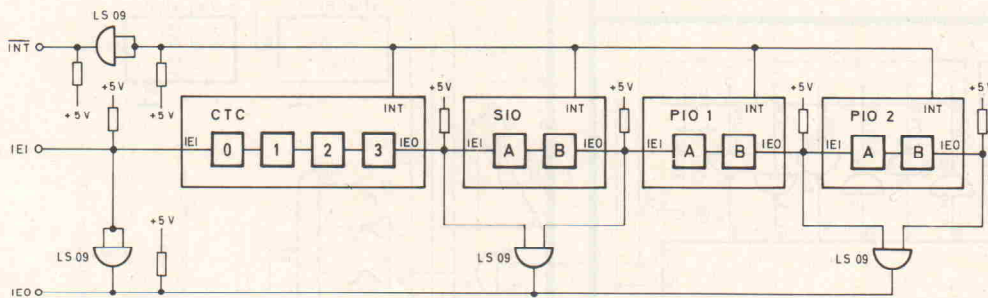
Bei einem Interrupt durch den CTC 0 (erfolgt etwa alle 15 ms bei 4 MHz Systemtakt) wird eine '0' ausgegeben; ein Interrupt durch die PIO 2 B meldet das Programm durch eine '4'.

Die Interrupt-Vektoren müssen auf einer geraden Adresse liegen, da sie bei der Programmie-

Ein Baustein kann nur dann einen Interrupt anfordern, wenn sein IEI auf '1' liegt. Tut er dies, geht sein IEO auf '0' und bleibt so, bis das zugehörige Interrupt-Unterprogramm beendet ist (mit dem Befehl RETI). Damit verhindert er, daß ein in der Kette hinter ihm liegender Port einen Interrupt absetzen kann, bevor er an der Reihe ist. Dieselbe Logik erlaubt es auch, daß ein höher priorisierter Interrupt die laufende Bedienroutine eines 'unwichtigeren' Ports unterbricht. Der umgekehrte Fall ist dagegen nicht möglich. Im 'Fachchinesisch' nennt man das 'Nested Interrupt' (verschachtelte Programmunterbrechung).

Die einzelnen Interrupt-Ebenen innerhalb eines Portbausteins sind genauso organisiert (siehe Abbildung). Ein Problem bei der 'Blümchenkette' — wörtliche Übersetzung von Daisy Chain — sind jedoch die Signallaufzeiten zwischen IEI und IEO: Im ungünstigsten Fall muß der Null-Pegel am IEO des höchstpriorisierten Bausteins die gesamte Kette durchlaufen, um den niedrigstpriorisierten Interrupt zu sperren. Die CPU stellt dafür aber nur eine begrenzte Zeit zur Verfügung, die ohne weitere Maßnahmen für vier Port-ICs ausreicht (bei maximaler Taktfrequenz). Für längere Ketten greift man zu 'vorausschauenden'

Schaltungen (Look-ahead-Logik) mit TTL-Gattern, die jede '0' an irgendeinem IEO sofort und schnell weitergeben (bei der c't-I/O-Karte mit einem LS 09). Mit den Z80-I/O-Bausteinen ist auch Mischbetrieb von Polling und Interrupt möglich, bei der Z80-PIO allerdings eingeschränkt, da ihre Strobe-Eingänge (ASTB und BSTB) nicht per Polling abgefragt werden können (kein Statusregister). Welcher Betriebsart der Vorzug gegeben wird, hängt von der Aufgabenstellung ab — und vom Können des Programmierers, da Fehler in Interrupt-Routinen nicht leicht zu finden sind.



Die Daisy Chain der c't-I/O-Karte

Die einzelnen Interrupt-Ebenen innerhalb eines Portbausteins sind genauso organisiert (siehe Abbildung). Ein Problem bei der 'Blümchenkette' — wörtliche Übersetzung von Daisy Chain — sind jedoch die Signallaufzeiten zwischen IEI und IEO: Im ungünstigsten Fall muß der Null-Pegel am IEO des höchstpriorisierten Bausteins die gesamte Kette durchlaufen, um den niedrigstpriorisierten Interrupt zu sperren. Die CPU stellt dafür aber nur eine begrenzte Zeit zur Verfügung, die ohne weitere Maßnahmen für vier Port-ICs ausreicht (bei maximaler Taktfrequenz). Für längere Ketten greift man zu 'vorausschauenden'

Die einzelnen Interrupt-Ebenen innerhalb eines Portbausteins sind genauso organisiert (siehe Abbildung). Ein Problem bei der 'Blümchenkette' — wörtliche Übersetzung von Daisy Chain — sind jedoch die Signallaufzeiten zwischen IEI und IEO: Im ungünstigsten Fall muß der Null-Pegel am IEO des höchstpriorisierten Bausteins die gesamte Kette durchlaufen, um den niedrigstpriorisierten Interrupt zu sperren. Die CPU stellt dafür aber nur eine begrenzte Zeit zur Verfügung, die ohne weitere Maßnahmen für vier Port-ICs ausreicht (bei maximaler Taktfrequenz). Für längere Ketten greift man zu 'vorausschauenden'

Die einzelnen Interrupt-Ebenen innerhalb eines Portbausteins sind genauso organisiert (siehe Abbildung). Ein Problem bei der 'Blümchenkette' — wörtliche Übersetzung von Daisy Chain — sind jedoch die Signallaufzeiten zwischen IEI und IEO: Im ungünstigsten Fall muß der Null-Pegel am IEO des höchstpriorisierten Bausteins die gesamte Kette durchlaufen, um den niedrigstpriorisierten Interrupt zu sperren. Die CPU stellt dafür aber nur eine begrenzte Zeit zur Verfügung, die ohne weitere Maßnahmen für vier Port-ICs ausreicht (bei maximaler Taktfrequenz). Für längere Ketten greift man zu 'vorausschauenden'

Die einzelnen Interrupt-Ebenen innerhalb eines Portbausteins sind genauso organisiert (siehe Abbildung). Ein Problem bei der 'Blümchenkette' — wörtliche Übersetzung von Daisy Chain — sind jedoch die Signallaufzeiten zwischen IEI und IEO: Im ungünstigsten Fall muß der Null-Pegel am IEO des höchstpriorisierten Bausteins die gesamte Kette durchlaufen, um den niedrigstpriorisierten Interrupt zu sperren. Die CPU stellt dafür aber nur eine begrenzte Zeit zur Verfügung, die ohne weitere Maßnahmen für vier Port-ICs ausreicht (bei maximaler Taktfrequenz). Für längere Ketten greift man zu 'vorausschauenden'

Die einzelnen Interrupt-Ebenen innerhalb eines Portbausteins sind genauso organisiert (siehe Abbildung). Ein Problem bei der 'Blümchenkette' — wörtliche Übersetzung von Daisy Chain — sind jedoch die Signallaufzeiten zwischen IEI und IEO: Im ungünstigsten Fall muß der Null-Pegel am IEO des höchstpriorisierten Bausteins die gesamte Kette durchlaufen, um den niedrigstpriorisierten Interrupt zu sperren. Die CPU stellt dafür aber nur eine begrenzte Zeit zur Verfügung, die ohne weitere Maßnahmen für vier Port-ICs ausreicht (bei maximaler Taktfrequenz). Für längere Ketten greift man zu 'vorausschauenden'

Die einzelnen Interrupt-Ebenen innerhalb eines Portbausteins sind genauso organisiert (siehe Abbildung). Ein Problem bei der 'Blümchenkette' — wörtliche Übersetzung von Daisy Chain — sind jedoch die Signallaufzeiten zwischen IEI und IEO: Im ungünstigsten Fall muß der Null-Pegel am IEO des höchstpriorisierten Bausteins die gesamte Kette durchlaufen, um den niedrigstpriorisierten Interrupt zu sperren. Die CPU stellt dafür aber nur eine begrenzte Zeit zur Verfügung, die ohne weitere Maßnahmen für vier Port-ICs ausreicht (bei maximaler Taktfrequenz). Für längere Ketten greift man zu 'vorausschauenden'

J5	14—16:	Systemtakt teilen für SIO(0)/DART oder
	15—16:	4,9152 MHz teilen für SIO(0)/DART
J5	4—3:	256:1 oder
	5—3:	128:1 oder
	6—8:	64:1 oder
	7—8:	32:1 oder
	10—8:	16:1 oder
	9—11:	8:1 oder
	12—11:	4:1 oder
	13—11:	2:1
J6	5—7	Takt vom Teiler auf SIO(0)/DART A oder
	5—6	Takt vom CTC0 auf SIO(0)/DART A
	8—10	Takt vom Teiler auf SIO(0)/DART B oder
	9—10	Takt vom CTC1 auf SIO(0)/DART B
J6	2—4	Takt vom Teiler auf CTC0
	1—3	Takt vom Teiler auf CTC1

Tabelle 2. Funktion der Brückenstecker

zum Beispiel die Tastatur abzufragen, schaltet das BIOS dann ein EPROM mit Boot-, Input-, Output- und Monitor-Routinen ein. Das RAM an dieser Stelle wird ausgeblendet. Erfolgt jetzt ein Interrupt, so bildet die CPU aus dem I-Register und dem Interrupt-Vektor eine Adresse, unter der in Form von zwei Bytes die Startadresse der Interrupt-Routine liegen soll. Hier aber ist nun das EPROM eingeblenet, und es wird eine falsche Adresse aus der 'Interrupt-Tabelle' geholt; die Folge ist 'Programmabsturz'.

Es ist also sehr wichtig, daß die Interrupt-Vektor-Tabelle und die Interrupt-Routinen so im RAM liegen, daß sie zu keiner Zeit von EPROMs oder anderen Speicherbänken 'verdeckt' werden. Eventuell muß man das Testprogramm in einen anderen Speicherbereich verschieben.

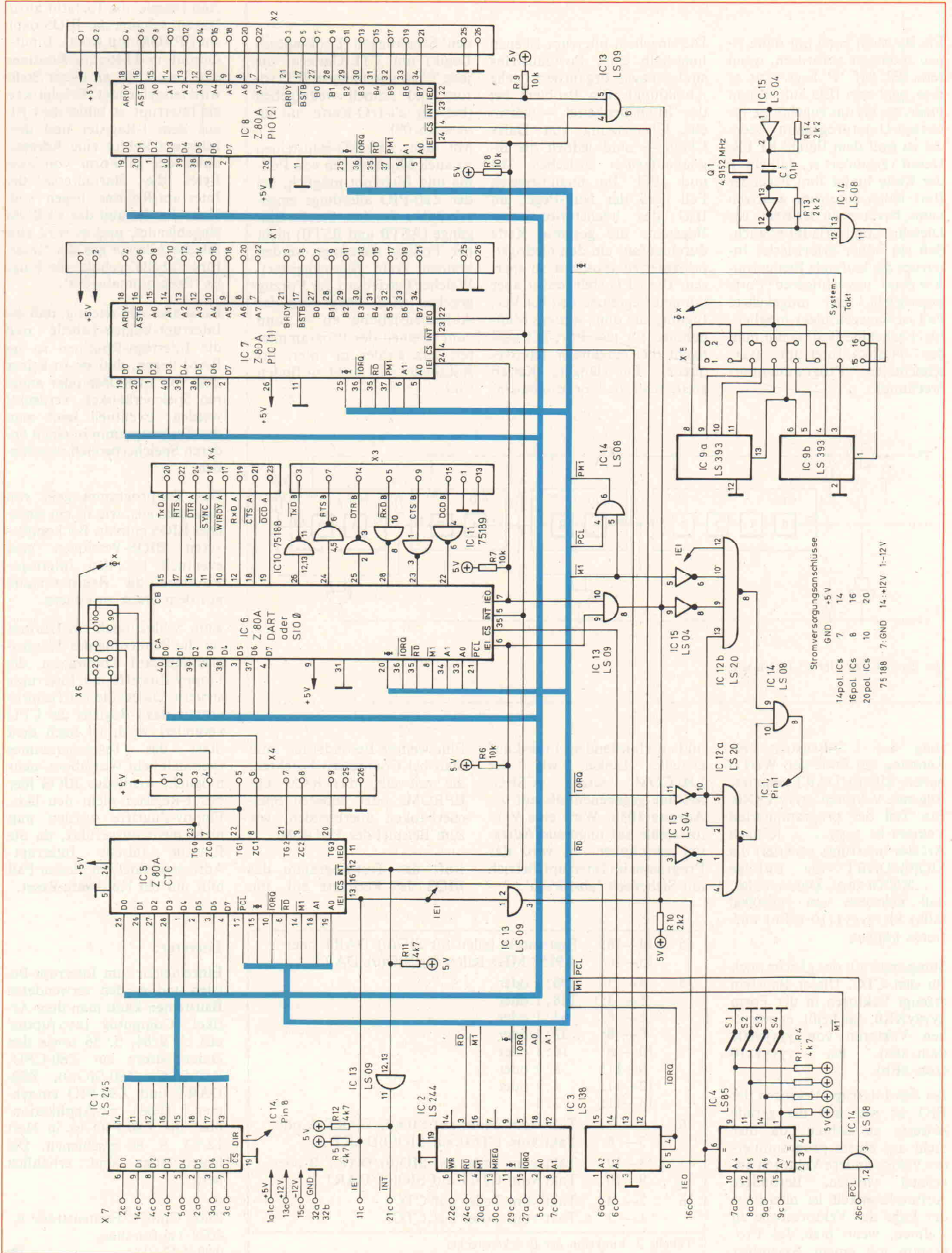
Das Testprogramm geht von Routinen aus, wie sie ein einfaches BIOS enthält. Bei komplexeren BIOS-Versionen sind eventuell bei den Interrupt-Routinen die Registerinhalte auf dem Stack zu sichern.

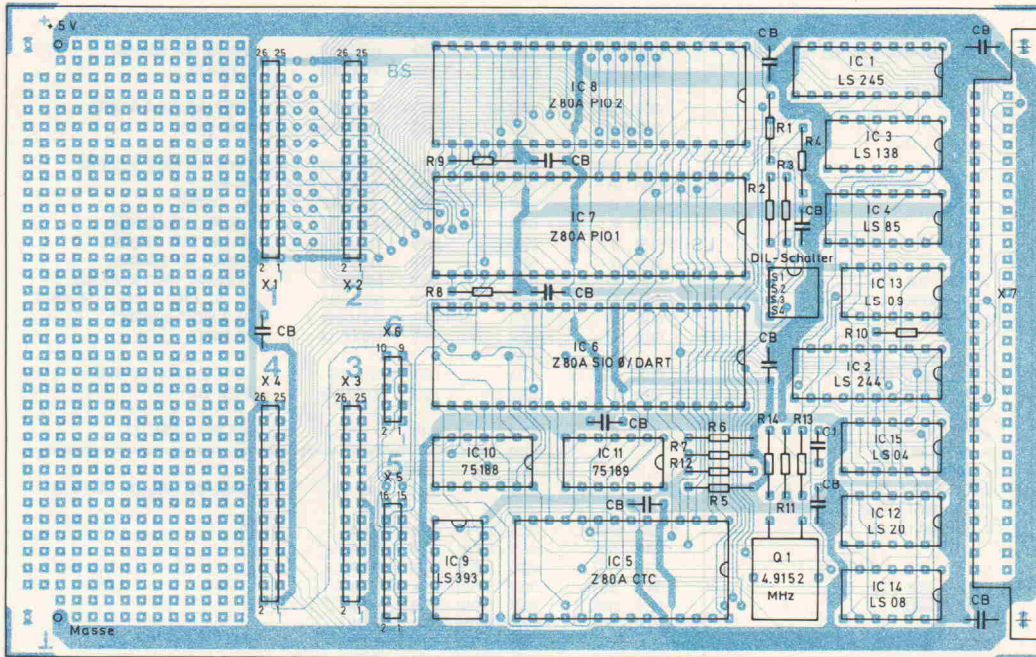
Zum Schluß noch ein Hinweis für die Benutzung des Testprogrammes auf Computern, die Floppy-Zugriffe mit Interrupts steuern. Da bei diesem Testprogramm das I-Register der CPU verändert wird, ist nach dem Start des Testprogrammes eventuell kein Warmboot mehr möglich, wenn das BIOS hier das I-Register nicht neu lädt. Floppy-Zugriffe werden nun nicht mehr ausgeführt, da die Routine falsche Interrupt-Adressen bildet. In diesem Fall hilft nur ein Hardware-Reset.

Literatur:

Einzelheiten zum Interrupt-Betrieb und zu den verwendeten Bausteinen kann man dem Artikel 'Computus Interruptus' aus c't 9/84, S. 56 sowie den Datenblättern zur Z80-CPU, Z80-CTC, Z80-SIO(0), Z80-DART und Z80-PIO entnehmen. Eine 'c't-Applikation' über die Z80-PIO ist in Heft 12/83, S. 86 erschienen. Die Datenblätter sind erhältlich bei:

Zilog GmbH, Eschenstraße 8, 8028 Taufkirchen, 089/6 12 60 46.





Stückliste

Widerstände

R1...5, 4k7
R11,12 4k7
R6...9 10k
R10,13,14 2k2

Kondensatoren

C1 100 nF, RM 5 mm
CB 11 Stützkondensatoren
je 100 nF, RM 5 mm

Halbleiter

IC1 74LS245
IC2 74LS244
IC3 74LS138
IC4 74LS85
IC5 Z80A — CTC
IC6 Z80A — DART oder
SIO(0)
IC7,8 Z80A — PIO
IC9 74LS393
IC10 75188 oder MC1488
IC11 75189 oder MC1489
IC12 74LS20
IC13 74LS09
IC14 74LS08
IC15 74LS04

Sonstiges

Q Quarz 4,9152 MHz, HC18
S1...4 DIL-Schalter, 4polig
X1...4 Pfostenleiste 26polig
X5 Pfostenleiste 16polig
X6 Pfostenleiste 10polig
X7 VG-Steckerleiste, Reihen
a und c bestückt, 64polig

Brückenstecker für Pfostenleisten;
IC-Fassungen: 7x14polig,
2x16polig, 2x20polig, 1x28polig,
3x40polig; 2 Schrauben M 2,5x10
mit Mutter, Platine 'c't-1/O-Karte'.

Bestückungsplan (oben) und
Stückliste der I/O-Karte

```

0002
(0080) 0003 BASE EGU 080H : 4 DIL-SCHALTER
0004
(0080) 0005 CTC EGU BASE+00H
(0080) 0006 CTC0 EGU CTC+0 : ZAEHLER 0
(0081) 0007 CTC1 EGU CTC+1 : ZAEHLER 1
(0082) 0008 CTC2 EGU CTC+2 : ZAEHLER 2
(0083) 0009 CTC3 EGU CTC+3 : ZAEHLER 3
0010
(0084) 0011 SIO EGU BASE+04H
(0084) 0012 SIOA0 EGU SIO+0 : KANAL A DATEN
(0085) 0013 SIOA1 EGU SIO+1 : CONT.
(0086) 0014 SIOB0 EGU SIO+2 : KANAL B DATEN
(0087) 0015 SIOB1 EGU SIO+3 : CONT.
0016
(0088) 0017 PIO1 EGU BASE+08H
(0088) 0018 PIO1A EGU PIO1+0 : PIO 1 A DATEN
(0089) 0019 PIO1B EGU PIO1+1 : B DATEN
(008A) 0020 PIO1AC EGU PIO1+2 : PIO 1 A CONT.
(008B) 0021 PIO1BC EGU PIO1+3 : B CONT.
0022
(008C) 0023 PIO2 EGU BASE+0CH
(008C) 0024 PIO2A EGU PIO2+0 : PIO 2 A DATEN
(008D) 0025 PIO2B EGU PIO2+1 : B DATEN
(008E) 0026 PIO2AC EGU PIO2+2 : PIO 2 A CONT.
(008F) 0027 PIO2BC EGU PIO2+3 : B CONT.
0028
(0100) 0029 ORG 100H : PROGRAMM FUER CPM 80
0030
0100 2A0100 0031 START: LD HL,(00001)
0103 23 0032 INC HL
0104 23 0033 INC HL
0105 23 0034 INC HL
0106 111001 0035 LD DE,CONST
0109 010900 0036 LD BC,00009
010C ED80 0037 LDIR
010E 1809 0038 JR XXXX
0039
0110 C30000 0040 CONST: JP 0 : KBD-STATUS: CHR A=FF
0041 : NO CHR A=00
0113 C30000 0042 CONIN: JP 0 : A: CHR
0116 C30000 0043 CONOUT: JP 0 : CHR. ZUM SCHIRM C1=CHR.
0044
0119 CD5C01 0045 XXXX: CALL INIT
011C FB 0046 EI
0047
011D DB80 0048 LOOP: IN A,(PIO2B) : 7 AUSBAENDE PIO 2 B
011F 3C 0049 INC A : ERZEUGEN RECHTECKE
0120 E67F 0050 AND 07FH : IM BINAR-CODE
0122 D380 0051 OUT (PIO2B),A
0052
0124 CD5201 0053 CALL SIOIN : CHAR. VON SIO ?
0127 2806 0054 JR Z-LOOP1 : NEIN
0129 4F 0055 LD C,A : CHAR. VON SIO AUF SCHIRM
012A F3 0056 DI
012B CD1601 0057 CALL CONOUT
012E FB 0058 EI

```

```

0059
012F CD1001 0060 LOOP1: CALL CONST : CHAR. AUF KEYBOARD ?
0132 B7 0061 OR A
0133 28E8 0062 JR Z-LOOP : NEIN
0135 CD1301 0063 CALL CONIN
0138 FE2E 0064 CP '.' : ABRUCH ?
013A 2006 0065 JR NZ,LOOP2 : NEIN
0066
013C F3 0067 DI
013D 00 0068 NOP
013E 00 0069 NOP
013F C30000 0070 JP 0000
0071
0142 4F 0072 LOOP2: LD C,A
0143 CD4801 0073 CALL SIOOUT : CHAR. VON KEYBOARD ZUM SIO
0146 1805 0074 JR LOOP
0075
0148 DB87 0076 SIOOUT: IN A,(SIOBC) : STATUS LESEN
014A CB57 0077 BIT 2,A : TX SENDEPuffer LEER ?
014C 28FA 0078 JR Z,SIOOUT : NEIN
014E 79 0079 LD A,C : CHAR.
014F D386 0080 OUT (SIOB0),A : SENDEN
0151 C9 0081 RET
0082
0152 DB87 0083 SIOIN: IN A,(SIOBC) : STATUS LESEN
0154 E601 0084 AND 0000001B : RX CHAR. VORHANDEN ?
0156 C8 0085 RET Z : KEIN CHAR.
0157 DB86 0086 IN A,(SIOB0)
0159 E67F 0087 AND 01111111B : BIT 7 = 0
015B C9 0088 RET
0089
015C 218701 0090 INIT: LD HL,CTCIT : CTC 0 INITIALISIEREN
015F CD7C01 0091 CALL INITX
0092
0162 219901 0093 LD HL,SIOIT : SIO INITIALISIEREN
0165 CD7C01 0094 CALL INITX
0095
0168 218401 0096 LD HL,PIOIT : PIO 1 INITIALISIEREN
016B CD7C01 0097 CALL INITX
0098
016E 21C901 0099 LD HL,PIO2T : PIO 2 INITIALISIEREN
0171 CD7C01 0100 CALL INITX
0101
0174 3E02 0102 LD A,HIGH INTVO : CPU INITIALISIEREN
0176 ED47 0103 LD I,A
0178 ED5E 0104 IM 2

```

017A ED4D	0105	RETI		: GGF PORT-BAusteINE AUS INTER. HOLEN
	0106			
017C 7E	0107	INITX: LD	A, (HL)	: BYTE-ANZAHL
017D B7	0108	OR	A	
017E C8	0109	RET	Z	
017F 47	0110	LD	B,A	
0180 23	0111	INC	HL	
0181 4E	0112	LD	C, (HL)	
0182 23	0113	INC	HL	
0183 ED83	0114	OTIR		
0185 18F5	0115	JR	INITX	
	0116			
0187 0380	0117	CTCIT: DEFB	03, CTC0	: 3 BYTE ZUM CTC0
0189 10	0118	DEFB	LOW INTVB	: INT-VEKTOR
018A A7	0119	DEFB	10100111B	: INT-ENABLE
	0120			: RESET
	0121			: ZEITGEBER / 256
	0122			: ZEIT-KONST. FOLGT
018B FF	0123	DEFB	255	: ZEIT-KONST.: 255
	0124			
018C 0281	0125	DEFB	02, CTC1	: 2 BYTE ZUM CTC1
018E 27	0126	DEFB	00100111B	: INT-DISABLE
	0127			: RESET
	0128			: ZEITGEBER / 256
	0129			: ZEIT-KONST. FOLGT
018F FF	0130	DEFB	255	: ZEIT-KONST.: 255
	0131			
	0132			
0190 0282	0133	DEFB	02, CTC2	: 2 BYTE ZUM CTC2
0192 27	0134	DEFB	00100111B	: INT-DISABLE
	0135			: RESET
	0136			: ZEITGEBER / 256
	0137			: ZEIT-KONST. FOLGT
0193 E1	0138	DEFB	225	: ZEIT-KONST.: 255
	0139			
	0140			
0194 0283	0141	DEFB	02, CTC3	: 2 BYTE ZUM CTC3
0196 27	0142	DEFB	00100111B	: INT-DISABLE
	0143			: RESET
	0144			: ZEITGEBER / 256
	0145			: ZEIT-KONST. FOLGT
0197 FF	0146	DEFB	255	: ZEIT-KONST.: 255
	0147			
	0148			
0198 00	0149	DEFB	0	: ENDE CTC-TABELLE
	0150			
0199 0885	0151	SIOIT: DEFB	11, SIOAC	: 11 BYTE ZUM SIO A
019B 18	0152	DEFB	00011000B	: RESET
019C 0100	0153	DEFB	1,00000000B	: REG1: ALLE INT. DISABLE
019E 0200	0154	DEFB	2,00000000B	: REG2: INT, VEC. LEER
01A0 03C1	0155	DEFB	3,11000001B	: REG3: 8 BIT/CHAR RX ENABLE
01A2 04C4	0156	DEFB	4,11000100B	: REG4: TRANSMITTER X&A CLOCK
	0157			: 1 STOPBIT NO PARITY
01A4 056A	0158	DEFB	5,01101010B	: REG5: RECEIVER DTR=0 8 BIT/CHAR
	0159			: TX ENABLE RTS=1
	0160			
01A6 0887	0161	DEFB	11, SIOBC	: 11 BYTE ZUM SIO A
01A8 18	0162	DEFB	00011000B	: RESET
01A9 0100	0163	DEFB	1,00000000B	: REG1: ALLE INT. DISABLE
01AB 0200	0164	DEFB	2,LOW INTVO	: REG2: INT, VECTOR
01AD 03C1	0165	DEFB	3,11000001B	: REG3: 8 BIT/CHAR RX ENABLE
01AF 04C4	0166	DEFB	4,11000100B	: REG4: TRANSMITTER X&A CLOCK
	0167			: 1 STOPBIT NO PARITY
01B1 056A	0168	DEFB	5,01101010B	: REG5: RECEIVER DTR=0 8 BIT/CHAR
	0169			: TX ENABLE RTS=1
	0170			
01B3 00	0171	DEFB	0	: ENDE SIO-TABELLE
	0172			
01B4 058A	0173	PIO1T: DEFB	05, P101AC	: 5 BYTE ZUM P10 1 A CONTROL
01B6 03	0174	DEFB	00000011B	: DIS-INT
01B7 4F	0175	DEFB	01001111B	: BETR. ART BYTE EINGABE
01B8 4F	0176	DEFB	01001111B	: BETR. ART BYTE EINGABE
01B9 CF	0177	DEFB	11001111B	: BETR. ART BIT EIN/AUSS.
01BA 00	0178	DEFB	00000000B	: ALLES AUSS.
	0179			
01B8 0188	0180	DEFB	01, P101A	: 1 BYTE ZUM P10 1 A DATEN
01BD 00	0181	DEFB	00000000B	: DATEN "LOW"
	0182			
01BE 058B	0183	DEFB	05, P101BC	: 5 BYTE ZUM P10 1 B CONTROL
01C0 03	0184	DEFB	00000011B	: DIS-INT
01C1 4F	0185	DEFB	01001111B	: BETR. ART BYTE EINGABE
01C2 4F	0186	DEFB	01001111B	: BETR. ART BYTE EINGABE
01C3 CF	0187	DEFB	11001111B	: BETR. ART BIT EIN/AUSS.
01C4 00	0188	DEFB	00000000B	: ALLES AUSS.
	0189			
01C5 0189	0190	DEFB	01, P101B	: 1 BYTE ZUM P10 1 B DATEN
01C7 00	0191	DEFB	00000000B	: DATEN "LOW"
	0192			
01C8 00	0193	DEFB	0	: ENDE P10-1-TABELLE
	0194			
01C9 058E	0195	PIO2T: DEFB	05, P102AC	: 5 BYTE ZUM P10 2 A CONTROL
01CB 03	0196	DEFB	00000011B	: DIS-INT
01CC 4F	0197	DEFB	01001111B	: BETR. ART BYTE EINGABE
01CD 4F	0198	DEFB	01001111B	: BETR. ART BYTE EINGABE

Das Testprogramm für die I/O-Karte in Z80-Mnemonics