

Verbindungsfreundlich

Es ist allgemein bekannt, daß der VC 20 über eine serielle Schnittstelle verfügt. Weniger bekannt ist, daß er auch mit einem parallelen Interface ausgestattet ist. Beide sollen hier beschrieben werden.

Die Schnittstellen des VC 20 werden von einem 6522-Chip gesteuert. Dieser verfügt über 16 Register, mit deren Hilfe zwei 8-Bit-Datenports, zwei 16-Bit-Timer, ein Schieberegister und verschiedene Steuerleitungen gehandhabt werden können. Für die parallele Schnittstelle benötigen wir nur die Register 0 und 2.

Das Register 0 gibt den Zustand des Datenports B (PB 0 bis PB 7 am User-Port) wieder. Es kann sowohl gelesen als auch geladen werden. Es liegt bei dezimal 37 136 (\$ 9110). Die Pin-Belegung des User-Ports geht aus Tabelle 1 hervor. Mit dem Register 2 können die Leitungen des Datenports individuell auf Ein- oder Ausgabe geschaltet werden. Ist zum Beispiel das Bit 0 gesetzt, so ist die korrespondierende Datenleitung (in diesem Fall PB 0) auf Ausgabe programmiert. Die anderen Leitungen stehen auf Eingabe, weil Bit 1 bis Bit 7 nicht gesetzt sind.

Das Ganze sieht in Basic wie in Listing 1 aus. Etwas schwieriger ist es, die serielle Schnittstelle zu programmieren. Sie hat einen Aus- und einen Eingang. Am User-Port ist CB 2 der Aus- und CB 1 der Eingang. Mit dem Basic-Befehl OPEN 2,2,0, CHR\$(Byte 1) + CHR\$(Byte 2) wird die serielle Schnittstelle eröffnet und programmiert. Die Bezeichnungen Byte 1 und Byte 2 stehen für das Kontroll- und das Befehlsregister. Die beiden Register werden nach den Tabellen 2 (Kontrollregister) und 3 (Befehlsregister) programmiert.

Listing 1. Die Programmierung der parallelen Schnittstelle

```
10 rem parallele schnittstelle
15 rem
20 rem alle leitungen auf ausgabe
25 rem
30 poke37138,255:rem 255 = %11111111
35 rem
40 rem auf jede leitung eine logische ein
50 poke37136,255:rem alle bits gesetzt
60 end
```

Listing 2. Die Programmierung der seriellen Schnittstelle

```
100 rem serielle schnittstelle
105 rem
110 open2,2,0,chr$(6+32+128)+chr$(32+64+
128):printchr$(14);chr$(147);
120 getb$:ifb$=""then160
130 a=0:b=asc(b$):ifb<91andb>64thena=32
140 if b=20thenb=8
150 b=b+a:print#2,chr$(b);
160 get#2,a$:ifa$=""then120
170 a=0:b=asc(a$):ifb<91andb>64thena=128
180 ifb>96thenb=b-32
190 ifb=8thenb=20
200 b=b+a:printchr$(b);:goto120
210 end
```

Wenn ich zum Beispiel das Kontrollregister auf 300 Baud, 7 Daten- und 2 Stoppbits programmieren möchte, muß ich die Bits 1, 2, 5 und 7 setzen. Man rechnet also $2^1 + 2^2 + 2^5 + 2^7 = 2 + 4 + 32 + 128 = 166$ und setzt diesen Wert bei Byte 1 ein. Genauso macht man es mit dem Befehlsregister, also: alle Werte zusammenzählen und in Byte 2 einsetzen.

PIN #	Name	Bemerkung	PIN #	Name
1	GND		A	GND
2	+5V	100mA max.	B	CB1
3	Reset		C	PB0
4	JOY0		D	PB1
5	JOY1		E	PB2
6	JOY2		F	PB3
7	LIGHT PEN		H	PB4
8	CASSETTE SWITCH		J	PB5
9	SERIAL ATN IN		K	PB6
10	+9V	100mA max.	L	PB7
11	GND		M	CB2
12	GND		N	GND

Tabelle 1. User-Port Belegung

Bit	3	2	1	0	Dezimal	Baud-Rate
	0	0	0	1	1	50
	0	0	1	0	2	75
	0	0	1	1	3	110
	0	1	0	0	4	134,5
	0	1	0	1	5	150
	0	1	1	0	6	300
	0	1	1	1	7	600
	1	0	0	0	8	1200
	1	0	0	1	9	1800
	1	0	1	0	10	2400

Bit	6	5	dezimal	Anzahl der Datenbits
	0	0	0	8
	0	1	32	7
	1	0	64	6
	1	1	96	5

Bit	7	dezimal	Anzahl der Stoppbits
	0	0	1
	1	128	2

Tabelle 2. Die Programmierung des Kontrollregisters der RS232-Schnittstelle

Bit	0	dezimal	Handshake
	0	0	3-Draht
	1	1	X-Draht

Bit	4	dezimal	Übertragungsart
	0	0	Vollduplex
	1	16	Halbduplex

Bit	7	6	5	dezimal	Paritätsprüfung
	0	0	0	0	keine Paritätsprüfung kein 8. Datenbit
	0	0	1	32	ungerade Parität
	0	1	1	96	gerade Parität
	1	0	1	160	keine Paritätsprüfung 8. Datenbit immer 1
	1	1	1	224	keine Paritätsprüfung 8. Datenbit immer 0

Tabelle 3. Die Programmierung des Befehlsregisters

Nun ist die serielle Schnittstelle programmiert. Ausgaben macht man über den Befehl PRINT # 2, CHR\$(x), wobei x der ASCII-Code des zu sendenden Buchstabens ist. Lesen kann man die Schnittstelle mit dem Befehl GET # 2, x\$, wobei x\$ das zuletzt gelesene Zeichen ist.

Ein Basic-Programm, das den VC 20 mit anderen Computern, zum Beispiel über ein Telefonmodem, kommunizieren läßt, ist als Listing 2 abgedruckt. Bei diesem Programm können falsche Eingaben mit der DEL-Taste gelöscht werden. Nun viel Spaß mit Ihren Schnittstellen. (Andrej Dvorak/ev)