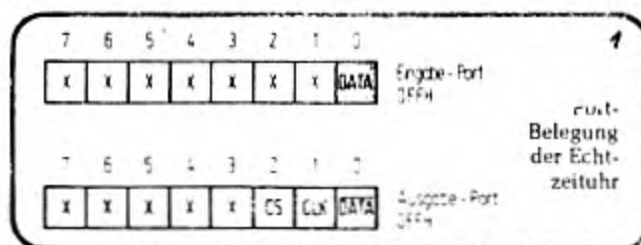


Vorläufiger Vorschlag zum Betrieb der Uhrenplatine

Da das Uhren-IC nur einen Datenein-/ausgang besitzt, müssen die Daten bitseriell verarbeitet werden. Dazu werden sie in Gruppen zu 8 Bit aufgeteilt und im BCD-Code dargestellt. Insgesamt werden dann sieben Bytes zum Auslesen des gesamten Inhalts benötigt. Das geschieht in der Reihenfolge: Stunden, Minuten, Monatstag, Monat, Jahr, Wochentag, Sekunden.

Das Programm besitzt zwei sinnvolle Einsprünge: auf Adresse 103H den Sprung zum Einschreiben eines Datensatzes und auf Adresse 106H den Sprung zum Stellen der Uhr. Die Programme sind als Subroutinen ausgeführt, so daß sie von einem Hauptprogramm aus zum Verarbeiten der Uhrzeit aufgerufen werden können. Nach dem Rücksprung stehen im Falle des Lesens die Uhrzeit und Datum ab Adresse 109H (7 Byte) in der vorher angegebenen Reihenfolge zur Verfü-



gung. Beim Stellen der Uhr müssen diese Speicherstellen entsprechend vorbelegt sein.

Die Routine PULSRD hat die Aufgabe, ein Bit vom Uhren-IC einzulesen. Dazu wird als erstes CLK auf Low gelegt und \overline{CS} ebenfalls. Die Bitzuordnung gibt Bild 1 an. Nach Ablauf einer Wartezeit, da die maximale Taktfrequenz begrenzt ist, wird CLK auf High gelegt, und das Datenbit kann eingelesen werden. Dann folgt wieder eine Wartezeit.

TDI Z80 CPU DISK ASSEMBLER VERSION 2.21
MAIN -

```

.PHFX
.PABS
.LOC 100H
;*****
;* UHRENPORG TEST1 *
;* 000021 ROK *
;*****
;
0100 C3 F01E JMP 0F01EH ;IN MONITOR
0103 C3 0146 JMP READ ;LESEN VON UHR IN BUFFER
0106 C3 0175 JMP WRITE ;SCHREIBEN DER UHR
0109 BUFFER:
;HLKB 8 ;ZEIT
;HRS MIN DATE MONTH YEAR DAY SECONDS

00F0 PORT=0F0H ;PRELIM

0111 WAIT: ;
0111 C5 PUSH B
0112 0600 MVI B,00H ;RESERVE 137*6.5535*00-TX
0114 LP:
0114 10FE DJNZ LP
0116 C1 POP B
0117 C9 RET
;
;
0110 PULSRD: ;LOW
0110 3E04 MVI A,4H ;CS = LOW
011A 03F0 OUT PORT
011C CD 0111 CALL WAIT
011F 3E04 MVI A,4H
0121 03F0 OUT PORT
0123 00F0 IN PORT
0125 F5 PUSH PSW
0126 CD 0111 CALL WAIT
0129 F1 POP PSW
012A C9 RET
012B PULSWR: ;LOW OUTDAT *HIGH HIGHIMP
;IN ANGU LSH 1ST DATA
012B F5 PUSH PSW
012C 3E04 MVI A,4
012E 03F0 OUT PORT
0130 F1 POP PSW
0131 E001 ANI 1
0133 F004 ORI 4
0135 03F0 OUT PORT
0137 CD 0111 CALL WAIT
013A F002 ORI 2 ;TO HIGH
013C 03F0 OUT PORT
013E CD 0111 CALL WAIT
0141 E0FE ANI 0FEH ;HIGH INPE.DANZ
0143 03F0 OUT PORT
0145 C9 RET
;
;
0146 READ:
0146 21 0109 LXI H,BUFFER

```

```

0119      B6B3          MVI B,3
011B      3E84          MVI A,4
011D      D3F0          OUT PORT          ;DREI START ADDR=HIGH
011F      LPI1
011F      AF           XRA A          ;0(1) AUSGEBEN INVERTED
0119      CD 012B      CALL PULSWR
0113      18F1          DJNZ LP1
0115      AF           XRA A
0118      CD 012B      CALL PULSWR          ;0(1) AUSGEBEN READ
                                COMMAND
0119      8EA7          MVI C,7 ;ALLE DATENWERTE
011B      LPH1
011B      86B8          MVI B,8 ;8 BITS JEWEILS
011D      AF           XRA A          ;ERGEBNISBYTE
011E      LP21
011E      F5           PUSH PSW
011F      CD 011B      CALL PULSRD
0112      EA41          ANI 1
0114      8F           RRC          ;NACH BIT 7
0115      5F           MOV E,A
0116      F1           POP PSW
0117      8F           RRC          ;LSB FIRST
0118      B3           ORA E          ;NA.....
0119      1BF3          GINZ LP2
011B      77           MOV M,A
011C      23           INX H
011D      8D           DCR C
011E      28FB          JRNZ LPM
0117      3ERR          MVI A,0
0112      D5F8          OUT PORT
0114      C9           RET
;
0115          WRITE1
;SCHREIBEN NACH UHR

0115      3E84          MVI A,4
0117      D3F0          OUT PORT
0119      21 01E9      LXI H,BUFFER
011C      86B3          MVI B,1
011E      LPI1
011E      AF           XRA A
011F      CD 012B      CALL PULSWR
0112      18FA          DJNZ LP11
0114      3F81          MVI A,1 ;8 NACH UHR
0116      CD 012B      CALL PULSWR          ;INVERTED DATA!
0119      8E87          MVI C,7 ;GESAMT
011B      LPH1
011B      86B8          MVI B,8
011D      7F           MOV A,H ;DATEN HOLEN
011E      23           INX H
011F      LP21
011F      87           RLC          ;MSB FIRST TO BIT 0
0119      F5           PUSH PSW
0111      2F           CMA          ;INVERTED
0112      EA81          ANI 1
0114      CD 012B      CALL PULSWR
0117      F1           POP PSW
0118      1BF5          DJNZ LP22
0119      8D           DCR C
011E      28FB          JRNZ LPM
011D      3E86          MVI A,0
011F      D5F8          OUT PORT
0111      C9           RET
;
;
END

```

Listing des Uhrenprogramms

PULSWR schreibt ein Bit in das Uhren-IC ein. Dazu wird zunächst \overline{CS} auf Low gelegt, dann werden die Daten angelegt. Auch CLK ist zu diesem Zeitpunkt Low. Nach einer Wartezeit wird CLK auf High gelegt, und nach einer weiteren Wartezeit wird der Datenausgang deaktiviert, so daß der Ausgang von LT1 wieder auf Low liegt. Beim Schreiben ist zu beachten, daß die Daten wegen des N1 invertiert an den Uhrenbaustein gegeben werden müssen. Dies geschieht erst in der Routine WRITE. Die Routine READ liest einen kompletten Datensatz aus der Uhr. Dazu wird zunächst \overline{CS} auf Low gelegt. Dann wird der Wert „1“ dreimal in die Uhr eingeschrieben. Dies ist ein 3-Bit-Adresse. Ist sie auf 111, werden alle Daten ausgelesen, bzw. eingeschrieben. Mit anderen Adressen lassen sich gezielt Zeit, Jahr etc. auslesen, doch wurde hier der Einfachheit halber nicht davon Gebrauch gemacht. Als nächstes wird noch einmal ein High-Signal ausgegeben. Dies bedeutet, daß ein Lesevorgang gestartet werden soll. Es werden dann nacheinander 7 Byte eingelesen. Jedes besteht aus 8 Bit, wobei das LSB zuerst ausgegeben wird. Die Daten werden, beginnend bei der Adresse BUFFER, in aufsteigender Reihenfolge abgelegt. Sie werden in gepackter BCD-Darstellung gespeichert. Am Schluß wird \overline{CS} wieder auf High gelegt. Beim Unterprogramm WRITE verhält es sich im Prinzip genauso, nur daß als viertes Bit eine Null eingeschrieben wird, um den Schreibvorgang anzukündigen. Es werden dann 7 Byte an den Uhrenbaustein ausgegeben, wobei jeweils 8 Bit beginnend mit dem MSB invertiert ausgegeben werden. Am Schluß wird wieder \overline{CS} auf High gelegt, um einen nachfolgenden Zugriff zu ermöglichen und synchronisieren zu können. ■

