NCR DECISION MATE V

ECHTZEITUHR
(K803-V001)

Die beiliegenden Seiten zeigen Ihnen, wie Sie diese Leistungserweiterung an Ihren NCR DECISION MATE V anschließen können. Bitte ordnen Sie diese Beschreibung in Ihre Bedienungsanleitung für den NCR DECISION MATE V ein.

Doc:017-0033071

NCR ist ständig bemüht, die Produkte im Zuge der Entwicklung von Technologie, Bauteilen, Soft- und Firmware dem neuesten Stand anzupassen. NCR behält sich deshalb das Recht vor, Spezifikationen ohne vorherige Ankündigung zu ändern.
Nicht alle hier beschriebenen Leistungen werden von NCR in allen Teilen der Welt vertrieben. Nähere Informationen bezüglich eventueller Einschränkungen oder Erweiterungen sowie den aktuellen Stand erfahren Sie von Ihrem Händler oder der nächstgelegenen NCR-Geschäftsstelle.

ECHTZEITUHR
(K803-V001)

INHALTSVERZEICHNIS

Teil 1
INSTALLATION

Teil 2
SOFTWARE

ALLGEMEINE HINWEISE

ACHTUNG: Diese Leistungserweiterung enthält CMOS Bauelemente. Bitte berücksichtigen Sie die üblichen Vorsichtsmaßnahmen für die Behandlung solcher Bausteine:

- *** Vermeiden Sie die Berührung der Kontaktleiste.**
- *** Stellen Sie sicher, daß Ihr NCR DECISION MATE V ausgeschaltet ist, bevor der Adapter eingesetzt oder entfernt wird.**
- *** Falls Sie das Gehäuse öffnen (siehe unten), sollten Sie die Bauelemente oder die Lötseite der Platine nicht berühren.**

Dieser Adapter verwendet (wie alle anderen) eine **IFSEL**-Nummer (**I**nter**F**ace **SEL**ect), um mit dem NCR DECISION MATE V Daten auszutauschen. Insgesamt stehen 10 IFSEL-Nummern zur Verfügung, denen wiederum jeweils acht E/A Ports zugeordnet sind. Die Software zur Echtzeituhr, die unter dem Betriebssystem p-UCSD zur Verfügung steht, verwendet die IFSEL-Nummer 4B (Ports C8 bis CF) als Standardvoreinstellung. Zum Zeitpunkt der Auslieferung sind die Wahlschalter des Adapters auf diesen Wert eingestellt. Falls Sie diesen voreingestellten Wert verwenden möchten, können Sie den Adapter - ohne weitere Vorbereitungen - in einen der Steckplätze **2 bis 6** Ihres NCR DECISION MATE V einstecken (Siehe Abb. 1.1).

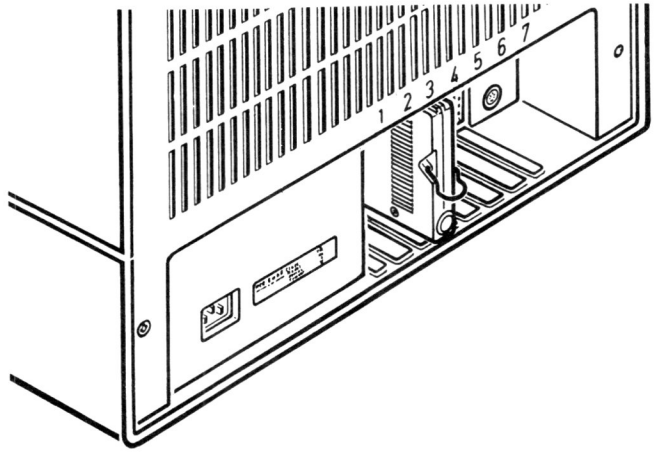

Abbildung 1.1 Anbringung des Adapters

Lesen Sie in diesem Fall bitte den zweiten Teil (Software) dieser Dokumentation.

Falls Sie die eingestellte IFSEL-Nummer verändern möchten, sollten Sie folgendermaßen vorgehen:

EINSTELLUNG DER IFSEL-NUMMER

1. Entfernen Sie den Drahtbügel und die vier Kreuzschlitzschrauben, ohne die Gehäusehälften zu trennen.
2. Drehen Sie den Adapter um (Schraublöcher nach unten) und legen Sie ihn auf eine flache Unterlage. Heben Sie die obere Gehäusehälfte ab, so daß Sie die Bestückungsseite (nicht die Lötseite) der Platine vor sich sehen. Außer den üblichen Vorsichtsmaßnahmen bei der Behandlung von CMOS-Bauelementen sollten Sie beachten, daß der Adapter einen kleinen Akkumulator enthält, um die Uhr mit Strom zu versorgen. Vermeiden Sie daher die Berührung der Platine oder von einzelnen Elementen mit leitfähigen Werkzeugen.
3. Bringen Sie die IFSEL-Wahlschalter (Abb. 1.2) in die gewünschte Stellung. Die Tabelle (Abb. 1.3) zeigt die möglichen Schaltereinstellungen. Um den Adapter in die selbe Lage zu bringen wie die Abbildungen in diesem Text, sollten Sie ihn so drehen, daß der Steckkontakt von Ihnen wegzeigt. Der Akkumulator auf der Platine ist Ihnen dann zugewandt. Der Schalter auf der linken Seite ist der Schalter Nummer 4, der Schalter auf der rechten Seite ist der Schalter B. Falls abweichende Beschriftungen auf dem Schalterblock vorhanden sind, sollten Sie diese nicht beachten. Ein Punkt (●) in der Abbildung 1.3 zeigt, daß der betreffende Schalter an der Seite der Steckerleiste niedergedrückt werden muß (sollte Ihr Adapter mit Schiebeschalter ausgestattet sein, müssen Sie den Schalter in Richtung auf den Steckkontakt schieben). Ein Kreis zeigt, daß der Schalter an der dem Akkumulator zugewandten Seite niedergedrückt bzw. in diese Stellung geschoben werden muß. Die Abbildung 1.4 zeigt die Schalterstellung für die IFSEL-Nummer 4B.
4. Bauen Sie den Adapter wieder zusammen und setzen Sie ihn in eine der Steckfassungen **2 bis 6** Ihres (ausgeschalteten) NCR DECISION MATE V ein (Siehe Abb. 1.1).

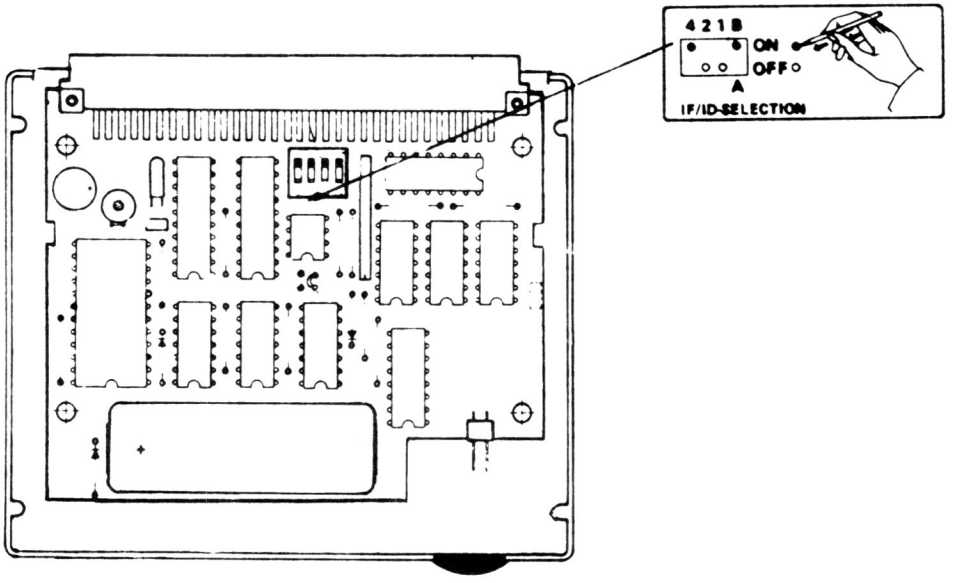

Abbildung 1.2 IFSEL-Schalter

IFSEL	SCHALTER 4 2 1 B	PORT-ADRESSE HEX	DEZ
0A	o o o o	60H - 67H	96 - 103
0B	o o o ●	68H - 6FH	104 - 111
1A	o o ● o	70H - 77H	112 - 119
1B	o o ● ●	78H - 7FH	120 - 127
2A	o ● o o	30H - 37H	48 - 55
2B	o ● o ●	38H - 3FH	56 - 63
3A	o ● ● o	B0H - B7H	176 - 183
3B	o ● ● ●	B8H - BFH	184 - 191
4A	● o o o	C0H - C7H	192 - 199
4B	● o o ●	C8H - CFH	200 - 207

Abbildung 1.3 Mögliche IFSEL-Einstellungen

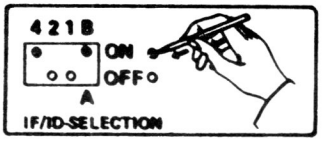

Abbildung 1.4 Beispiel: IFSEL-Nummer 4B

INTERRUPT-SIGNAL IM BEREITSCHAFTSBETRIEB

Die Echtzeituhr läuft auch nach dem Ausschalten Ihres Computers oder nach der Entfernung aus dem Computer weiter. Es besteht die Möglichkeit, die Echtzeituhr auf das Setzen eines Interrupt-Signals in diesem Bereitschaftsbetrieb zu programmieren. Das Interrupt-Signal wird in dem Augenblick erzeugt, in dem die Werte der laufenden Zeitzähler den vorab gespeicherten Werten entsprechen. Dieses Interrupt-Signal wird jedoch nur einmal gesetzt und verbleibt bis zu einer Rücksetzung in diesem Zustand.

Das Interrupt-Signal kann an zwei Kontakten (P2) auf der Platine abgegriffen werden (Siehe Abbildung 1.5):

Anschluß 1: INTERRUPT-SIGNAL (gesetzt=LOW)
Anschluß 2: MASSE

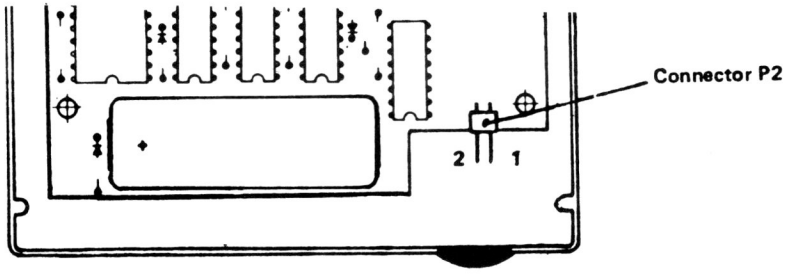

Abbildung 1.5 2-Pol Stiftleiste P2

Diese Stiftleiste ist nach Öffnung des Adapters (Siehe Beschreibung auf Seite 2) zugänglich. Den Anschlußdraht können Sie nach Entfernung des Blindstopfens durch die Gehäuseöffnung herausführen.

Der Ausgang kann z.B. zum Einschalten eines Gerätes durch eine Steuerschaltung verwendet werden. Er ist für die Belastung durch eine TTL-Schaltung entworfen. Es muß ein Abschlußwiderstand (Pull-Up- Widerstand) von 3.6KOhm verwendet werden.

Die softwareseitige Beschreibung dieses Interrupts finden Sie im zweiten Teil dieser Dokumentation.

STROMVERSORGUNG

Der Adapter enthält einen kleinen Akkumulator (3.6V; 60mAh), der die Stromversorgung der Echtzeituhr sicherstellt, wenn der Computer ausgeschaltet oder der Adapter nicht in Ihren NCR DECISION MATE V eingesteckt ist. Der Akkumulator wird automatisch während dem Betrieb des Computers geladen. Die mögliche Zeitdauer der Batterieunterstützung ist stark von der Umgebungstemperatur abhängig, bei konstant 20 Grad Celsius kann von einer Dauer von etwa 180 Tagen ausgegangen werden.

Sie sollten unter keinen Umständen versuchen, den Akkumulator von der Platine zu entfernen.

SOFTWARE

EINLEITUNG

Die Echtzeituhr ist ein Zeitzähler mit programmierbarem Alarmsignal. Die Zeiteinheiten, die von der Echtzeituhr gezählt werden sind Monate, Monatstage, Wochentage, Stunden, Minuten, Sekunden und Sekundenbruchteile.

Der Zeitpunkt für ein Alarmsignal wird in einem Satz von Speichern ("Latches") festgehalten. Durch das Besetzen einzelner oder aller Speicher mit einem "Ignoriere"-Wert ist es möglich, das Alarmsignal in wiederkehrenden Intervallen zu programmieren. Werden z.B. die Speicher für Monat, Tag des Monats und Wochentag auf "Ignoriere" gesetzt und die restlichen Speicher auf 12 Uhr Mittags programmiert, wird jeden Tag um 12 Uhr Mittags das Alarmsignal gegeben.

Außer der Zeitmessung und der Alarmfunktion bietet die Echtzeituhr folgende Leistungen:

Für den Zweck der Synchronisierung der Echtzeituhr steht ein eigenes "GO"-Befehlsregister zur Verfügung. Ein spezielles Bit ermöglicht Assemblerprogrammen eine Kontrolle, ob sich einer der Zähler während eines Lesezugriffes in Umstellung befand. Ein Lesen der Zeitzähler während des Umstellungsvorgangs kann zu einer Ungenauigkeit führen.

Die UCSD-p-Betriebssystemsoftware unterstützt die Echtzeituhr durch eine Reihe von Routinen in den Programmiersprachen BASIC, FORTRAN und Pascal. Weitergehende Information zu diesen Routinen können Sie der Dokumentation des UCSD-p-Betriebssystems entnehmen.

Selbstverständlich kann die Echtzeituhr auch unter den Betriebssystemen CP/M und MS-DOS in Assembler oder BASIC programmiert werden. Die folgenden Seiten geben Beispiele und Hinweise zur Implementierung solcher Routinen in diesen Betriebssystemen.

FORMATUMWANDLUNG

Die Speicher und Zähler der Echtzeituhr enthalten Werte im BCD-Format (Binär kodierte Dezimalzahlen). In diesem Format stellt jedes Byte zwei Stellen der Zahl, jeweils in vier Bit, dar. Die Dezimalzahl "25" wird z.B in einem Byte als 0010 0101 abgelegt, wobei die Ziffer mit der höheren Dezimalwertigkeit ("2") die oberen vier Bits belegt. Da in BASIC die Erkennung von BCD-kodierten Zahlen nicht vorgesehen ist, muß zunächst eine Ersatzdarstellung für die BCD-Zahlen errechnet werden. Für das obige Beispiel findet sich die Dezimalzahl 37, die als 00100101 binär kodiert wird.

Die Gleichung für das Umwandeln von Dezimalzahlen in eine Ersatzdarstellung für BCD-kodierte Zahlen lautet in BASIC:

```
BCD = INT(DEZIMAL/10) * 6 + DEZIMAL
```

Die Gleichung für das Umwandeln der Ersatzdarstellung von BCD-kodierten Zahlen in Dezimalzahlen lautet in BASIC:

```
DEZIMAL = INT(BCD/16) * 10 + BCD MOD 16
```

Beispiel 1:
Umwandlung der Dezimalzahl 43 in eine BCD-kodierte Zahl:

```
BCD = INT(43/10) * 6 + 43
    =      4     * 6 + 43
    = 67
```

Die Ausgabe der Zahl 67 durch das BASIC Programm ergibt dasselbe Bitmuster wie die BCD-Kodierung von 43.

Beispiel 2:
Umwandlung der BCD-kodierten Zahl 67 in eine Dezimalzahl:

```
DEZIMAL = INT(67/16) * 10 + 67 MOD 16
        =      4     * 10 +     3
        = 43
```

Dies ist der wahre Dezimalwert des Bytes 0100 0011, das von BASIC aber als "67" gelesen wird.

Jede BCD-Zahl entspricht der Binärkodierung einer hexadezimalen Zahl, bei der keine Ziffer größer als neun ist. Dies ist von besonderer Bedeutung, falls Sie in Assembler programmieren oder hexadezimale Zahlen als Eingabewerte für ein BASIC-Programm verwenden.

ADRESSIERUNG VON REGISTERN

Die Echtzeituhr verwendet 23 Register. Ein Registerzugriff wird durch zwei zusammengehörige Befehle ausgeführt:
Der erste Befehl wählt einen der sechs Adreßspeicher aus, welche jeweils eine Gruppe von vier (Adreßspeicher 5: drei) Register verwalten. Der nachfolgende Befehl wählt eines der zum angewählten Adreßspeicher gehörigen Register und greift auf dieses zu. Die Abbildung 3.1 zeigt tabellarisch die Funktion der einzelnen Register, die Zugehörigkeit zu den Adreßspeichern und die mögliche Art des Zugriffs.
(R=READ=Lesezugriff; W=WRITE=Schreibzugriff)

BADD stellt die Basisadresse des verwendeten IFSELs dar. Unter der Annahme, daß IFSEL 4B verwendet wird, ist das adressierte Port C8H (Siehe Abb. 1.3). Der zuletzt aktivierte Adreßspeicher bleibt gültig bis zur nächsten Ausgabe an das Port BADD. Dies ermöglicht den Austausch von Daten mit Register, die zu einen Adreßspeicher gehören, ohne der Notwendigkeit einer wiederholten Anwahl des Adreßspeichers. Das nachfolgende Beispiel zeigt diesen Vorgang: Zuerst wird eine Gruppe von vier Registern angewählt, dann werden die Werte für die Register der Sekunden- und Minutenzähler auf die jeweiligen Werte in SEC und MIN gesetzt.

```
OUT BADD,0
OUT BADD+6,SEC
OUT BADD+7,MIN
```

FUNKTION	ADR.SPEICHER	REGISTER	ZUGRIFF
ZÄHLER 1/10000 SEKUNDEN	BADD,0	BADD+4	R/W
ZÄHLER 1/100 SEKUNDEN	BADD,0	BADD+5	R/W
ZÄHLER SEKUNDEN	BADD,0	BADD+6	R/W
ZÄHLER MINUTEN	BADD,0	BADD+7	R/W
ZÄHLER STUNDEN	BADD,1	BADD+4	R/W
ZÄHLER WOCHENTAGE	BADD,1	BADD+5	R/W
ZÄHLER TAGE DES MONATS	BADD,1	BADD+6	R/W
ZÄHLER MONATE	BADD,1	BADD+7	R/W
LATCH 1/10000 SEKUNDEN	BADD,2	BADD+4	R/W
LATCH 1/100 SEKUNDEN	BADD,2	BADD+5	R/W
LATCH SEKUNDE	BADD,2	BADD+6	R/W
LATCH MINUTE	BADD,2	BADD+7	R/W
LATCH STUNDE	BADD,3	BADD+4	R/W
LATCH WOCHENTAG	BADD,3	BADD+5	R/W
LATCH TAG DES MONATS	BADD,3	BADD+6	R/W
LATCH MONAT	BADD,3	BADD+7	R/W
INTERRUPT STATUS	BADD,4	BADD+4	R
INTERRUPT BEFEHL	BADD,4	BADD+5	W
RÜCKSETZEN ZÄHLER	BADD,4	BADD+6	W
RÜCKSETZEN LATCHES	BADD,4	BADD+7	W
UMSTELLUNGSKONTROLLBIT	BADD,5	BADD+4	R
"GO" BEFEHL	BADD,5	BADD+5	W
BEREITSCHAFTSINTERRUPT	BADD,5	BADD+6	W

Abbildung 2.1 Register der Echtzeituhr

DIE STRUKTUR DER REGISTER

Es stehen jeweils acht Register für die Zähler und für die Alarm-"Latches" zur Verfügung. Jedes Register enthält acht Bits und kann somit eine ein- oder zweiziffrige BCD-Zahl enthalten. Unbelegte Bits sind in der Abb. 2.2 durch "-" dargestellt, sie werden bei Lesezugriffen als logisch 0 zurückgegeben und bei Schreibzugriffen ignoriert.

ZÄHLER/LATCH	EINER Bit:D0	D1	D2	D3	ZEHNER Bit:D4	D5	D6	D7
1/10000 SEKUNDEN	-	-	-	-	X	X	X	X
1/100 SEKUNDEN	X	X	X	X	X	X	X	X
SEKUNDEN	X	X	X	X	X	X	X	-
MINUTEN	X	X	X	X	X	X	X	-
STUNDEN	X	X	X	X	X	X	-	-
WOCHENTAGE	X	X	X	-	-	-	-	-
TAGE DES MONATS	X	X	X	X	X	X	-	-
MONAT	X	X	X	X	X	-	-	-

Abbildung 2.2 Bitbelegung der Zähler- und Latch-Register

PROGRAMMIERUNG DER REGISTER

Die Abbildungen 2.1 und 2.2 enthalten alle Informationen, die Sie zum Programmieren der Register benötigen. Nachfolgend finden Sie Beispiele für Zugriffe auf die Zählerregister und eine Beschreibung der Alarm-"Latches". Beachten Sie bitte, daß es nicht erforderlich ist, einen Adreßspeicher wiederholt anzuwählen, wenn nacheinander Zugriffe auf die zugeordneten Register erfolgen.

PROGRAMMIERUNG DER ZÄHLERREGISTER

Das folgende Beispiel stellt den Stundenzähler auf den in der Variable HOUR enthaltenen BCD-Wert:

```
OUT BADD,1          (Wahl des Adreßspeichers)
OUT BADD+4,HOUR (Schreibzugriff auf den
                    Stundenzähler)
```

Das folgende Beispiel liest den BCD-Wert des Stundenzählers in die Variable HOUR ein:

```
OUT  BADD,1          (Wahl des Adreßspeichers)
IN  (BADD+4),HOUR (Lesen des Registerinhalts
                    in die Variable)
```

PROGRAMMIERUNG DER ALARM-"LATCHES"

Die Struktur und die Zugriffsweise auf die Register der Alarm-"Latches" unterscheiden sich nicht von denen der Zählerregister. Die Echtzeituhr vergleicht den Inhalt der Latch-Register mit denen der Zählerregister und löst einen Interrupt aus, sobald alle folgenden Bedingungen erfüllt sind:

* Es wird eine Übereinstimmung von mindestens einem Zähler- / Alarmregister ermittelt.
* Alle Alarmregister, die nicht übereinstimmen, sind auf den Wert "ignoriere" gesetzt.
* Der Alarm-"Latch"-Interrupt ist aktiviert. (Siehe Abschnitt "PROGRAMMIEREN DES INTERRUPTS")

Die Register der Alarm-"Latches" können einzeln auf "ignoriere" gesetzt werden durch die jeweilige Belegung mit der Bitfolge 1100 1100 (Dezimalwert 204). Die Echtzeituhr erkennt diesen besonderen Wert, da keine BCD-kodierte Zahl existiert, bei der die oberen zwei Bits der vier-Bit-Gruppen gesetzt sind. Das folgende Beispiel setzt den Alarm-"Latch" für Minuten auf "ignoriere":

```
OUT BADD,2        (Wahl des Adreßspeichers)
OUT BADD+7,204    (Schreibzugriff auf den
                    Latch:"ignoriere")
```

RÜCKSETZEN DER REGISTER

Um die Zähler- und die Alarmregister einfach auf den niedrigstmöglichen Wert zu setzen, stehen jeweils spezielle Register zur Verfügung. Um die Rücksetzung zu bewirken, müssen die jeweiligen Register (Siehe Abbildung 2.1) mit dem Dezimalwert 255 besetzt werden. Beispiel:

```
OUT BADD,4        (Wahl des Adreßspeichers)
OUT BADD+6,255    (Rücksetzen aller Zählerregister)
```

PROGRAMMIERUNG DER INTERRUPTS

Zusätzlich zum bisher beschriebenen "Alarm"-Interrupt kann die Echtzeituhr auf die Erzeugung von Interrupts in den folgenden Intervallen programmiert werden:

einmal pro 1/10 Sekunde
einmal pro Sekunde
einmal pro Minute
einmal pro Stunde
einmal pro Tag
einmal pro Woche
einmal pro Monat

Die Abbildung 2.3 zeigt die Bedeutung der einzelnen Bits im Interrupt-Befehlsregister. Ein gesetztes Bit zeigt an, daß der jeweilige Interrupt aktiviert ist.

BITPOSITION	DEZIMAL	INTERRUPTQUELLE
7	128	Monat
6	64	Woche
5	32	Tag
4	16	Stunde
3	8	Minute
2	4	Sekunde
1	2	1/10 Sekunde
0	1	Alarm

Abbildung 2.3 Bitpositionen des Interrupt-Befehlsregisters

Zwei Beispiele zeigen die Aktivierung von Interrupts:

```
OUT BADD,4        (Wahl des Adreßspeichers)
OUT BADD+5,1      (Setzen des Bit 0, Alarm)

OUT BADD,4        (Wahl des Adreßspeichers)
OUT BADD+5,24     (Setzen der Bits Stunde
                   und Minute)
```

Ein Interrupt-Statusregister enthält Information, welche Interrupts seit dem letzten Lesen des Statusregisters stattgefunden haben. Die Zuordnung der Bitpositionen bzw. der Dezimalwerte entspricht der Abbildung 2.3. Beispiel:

```
OUT BADD,4            (Wahl des Adreßspeichers)
IN (BADD+4),INTSET    (Einlesen in Variable INTSET)
```

Der Dezimalwert 1 in der Variable INTSET nach dem Lesen des Interrupt-Statusregisters z.B. zeigt an, daß seit dem letzten Lesen ein Alarm-Interrupt ausgelöst worden ist.

DER INTERRUPT IM BEREITSCHAFTSBETRIEB

Dieser Interrupt ist an die Erfüllung folgender Bedingungen geknüpft:

* Die Werte aller Alarm-"Latches", die nicht mit "ignoriere" besetzt sind, stimmen mit den Werten der jeweiligen Zähler überein.
* Der Bereitschafts-Interrupt ist aktiviert.

Der Interrupt ist aktiviert, wenn das niedrigstwertige Bit des zugehörigen Registers (Siehe Abb. 2.1) gesetzt ist. Dieser Interrupt wird vom Interrupt-Steuerregister nicht beeinflußt. Beispiel:

```
OUT BADD,5        (Wahl des Adreßspeichers)
OUT BADD+6,1      (Aktivieren des Ber.Interrupts)
            bzw.
OUT BADD+6,0      (Rücksetzen des Ber.Interrupts)
```

UMSTELLUNGSKONTROLLBIT

Dieses Bit wird in einem eigenen Register (Siehe Abb.2.1) gespeichert und wird von der Echtzeituhr gesetzt, wenn sich eines der Zählerregister gerade in Umstellung befindet. Dieses Bit ist insbesondere von Bedeutung für Assembler-Programme, die ein hohes Maß an Zeit-Einlesegenauigkeit erfordern. Solche Programme sollten dieses Bit abfragen und einen Lesezugriff auf einen Zähler wiederholen, falls dieses Bit gesetzt war.

DER "GO"-BEFEHL

Mithilfe dieses Befehls kann der Ablauf der Echtzeituhr genau gestartet werden. Ein Schreibzugriff auf das Register, das diesen Befehl enthält, bewirkt die Rücksetzung der Zähler 1/10000, 1/100, 1/10 und 1/1 Sekunde. Falls der Wert des Zählers für ganze Sekunden zum Zeitpunkt der Rücksetzung größer als 40 ist, wird der Minutenzähler um eins erhöht. Andernfalls ist der Minutenzähler nicht betroffen.

Der Ablauf der Synchronisierung der Echtzeituhr auf die Zeit 15 Uhr 12 Minuten ist z.B. wie folgt:

1. Setzen der "langsameren" Zähler für Stunden und Minuten auf die Werte 15 bzw. 12.
2. Setzen des Sekundenzählers auf den Wert "0"
3. Wahl des Adreßspeichers: OUT BADD,5
4. Vorausgesetzt, daß der Wert "1" um genau 15 Uhr 12 in den "GO"-Befehlsregister geschrieben wird und seit der Nullsetzung des Sekundenzählers weniger als 40 Sekunden vergangen sind, wird die Echtzeituhr genau auf 15 Uhr 12 synchronisiert: OUT BADD+5,1

Die genaue Einstellung der Echtzeituhr ist nur in Assembler-Programmen möglich, da diese die unmittelbarste Ausführung eines Programmausgabebefehls gewähren. Für Programmierer, die die größtmögliche Synchronisations-Genauigkeit erzielen möchten, ist die Arbeitsfrequenz der Prozessoren des NCR DECISION MATE V von Bedeutung:

8-Bit (Z80A) - 4MHz
16-Bit (8088) - 5MHz

SOFTWAREBEISPIELE

Der folgende Abschnitt zeigt anhand zweier Beispiele die Programmierung der Echtzeituhr in BASIC. Auch Assembler-Programmierern werden diese Beispiele nützlich sein, da die grundsätzlichen Ein-/Ausgabebefehle an die Echtzeituhr auch in der höheren Programmiersprache verständlich sind.

Das erste Beispiel zeigt wie die Uhrzeit und das Tagesdatum eingestellt und abgefragt werden. Außerdem wird ein Test durchgeführt, um die verwendete IFSEL-Nummer zu ermitteln und davon die korrekte Basisadresse abzuleiten und auf der Diskette zu speichern.

Das zweite Beispiel zeigt das setzen der Alarm-Zeit und ermöglicht die Ausgabe eines Alarmtextes. Die Basisadresse wird von der Diskette gelesen.

Die Programme sind unter den Betriebssystemen CP/M und MS-DOS lauffähig.

BEISPIEL 1

ZWECK: STELLEN DER ECHTZEITUHR, ANZEIGEN VON DATUM UND UHRZEIT.

BESCHREIBUNG DES PROGRAMMS:

ZEILENNR.:	KURZBESCHREIBUNG:
50 - 100	ROUTINE ZUR UMWANDLUNG VON DEZIMAL ZU BCD UND ZURÜCK
110 - 130	SPRUNG IN DIE TESTROUTINE DER ECHTZEITUHR (SIEHE 1140)
140 - 210	AUSGABE EINES BILDSCHIRMMENÜS: WAHL ZWISCHEN STELLEN, ABFRAGEN DER ZEIT ODER ENDE DES PROGRAMMS
220 - 380	EINGABE DER ZEIT UND DES DATUMS. EINGABEN AUSSERHALB DES WERTEBEREICHS WERDEN ERKANNT.
390 - 460	DEZIMAL EINGEGEBENE DATEN WERDEN IN BCD VERWANDELT.
470 - 550	DIE UMGEWANDELTEN WERTE WERDEN IN DIE REGISTER DER ECHTZEITUHR EINGETRAGEN.
560 - 700	ZEIT UND DATUM WERDEN VON DER ECHTZEITUHR ABGEFRAGT. EINE UMSTELLUNG DER UHR VON 59 SEKUNDEN AUF 00 SEKUNDEN WIRD ÜBERWACHT. ANSCHLIESSEND WERDEN DIE ERMITTELTEN WERTE VON BCD NACH DEZIMAL UMGEWANDELT. "FLAG" IST EIN SCHALTER FÜR DIE BILDSCHIRMAUSGABE DER ZEIT UND DES DATUMS BEIM ERSTEN SCHLEIFENDURCHGANG .
710 - 820	AUSGABE DES DATUMS UND DER ZEIT IM KORREKTEN FORMAT. WOCHENTAG- UND MONATWERTE WERDEN MIT NAMEN ANGEZEIGT.
830 - 880	ABFRAGE, OB DER ANWENDER DAS PROGRAMM VERLASSEN MÖCHTE.
890 - 980	UMWANDLUNG VOM WOCHENTAG-WERT IN DEN WOCHENTAG-NAMEN.
990 - 1130	UMWANDLUNG VOM MONATS-WERT IN DEN MONATS-NAMEN.

1140 - 1480 DIESES UNTERPROGRAMM KONTROLLIERT, OB DIE ECHTZEITUHR ZUR VERFÜGUNG STEHT. EIN AUSGEWÄHLTES ALARM-"LATCH"-REGISTER WIRD MIT EINEM WERT BESCHRIEBEN. DIE ECHTZEITUHR IST VERFÜGBAR, WENN EINE ANSCHLIESSENDE ABFRAGE DES REGISTERS DENSELBEN WERT ZURÜCKGIBT, DER VORHER GESCHRIEBEN WURDE. FALLS DIE BASISADRESSE DES PORTS FALSCH IST, WIRD EINE TABELLE ALLER I/O PORTADRESSEN ANGEZEIGT. ES KANN DER KORREKTE WERT EINGEGEBEN WERDEN, EINE ABFRAGE FINDET STATT.
DIE STANDARDVOREINSTELLUNG DER PORTADRESSE IST DEZIMAL 200 (HEXADEZIMAL C8 VON IFSEL 4B). DIESE PORTADRESSE WIRD AUF DER DISKETTE GESPEICHERT, SIE WIRD VOM BEISPIELPROGRAMM 2 VON DORT ABGELESEN.

1490 - 1530 ROUTINEN ZUR ZEIGERPOSITIONIERUNG UM EINE KORREKT FORMATIERTE AUSGABE ZU ERZIELEN.

```
10 REM                EXAMPLE - 1
20 REM  SET AND GET TIME TO/FROM REAL TIME CLOCK
30 REM                  K 8 0 3
40 REM  ------------------------------------------
50 REM DEFINE A FUNCTION TO CONVERT FROM DECIMAL TO BCD
60 DEF FNTOBCD(X)=(X\10)*6+X
70 REM
80 REM DEFINE A FUNCTION TO CONVERT FROM BCD TO DECIMAL
90 DEF FNTODEC(X)=(X\16)*10 + X MOD 16
100 REM
110 REM  GOTO RTC AVAILABLE
120 GOSUB 1140
130 REM
140 PRINT:PRINT " FUNCTION SELECTION : "
150 PRINT " 1 - SET TIME TO RTC "
160 PRINT " 2 - GET TIME FROM RTC "
170 PRINT " 3 - EXIT PROGRAM "
180 SEL$=" "
190 INPUT " ",A
200 ON A GOTO 220,550,1540
210 PRINT " ? ";:GOTO 190
220 REM CLEAR SCREEN AND GET TIME FROM KEYBOARD
230 REM ------------------------------------------
240 PRINT CHR$(26);"PLEASE ENTER THE FOLLOWING DATA!"
250 PRINT
```

```
260 INPUT "           MONTH (1-12):";MTH
270  IF MTH<1 OR MTH>12 GOTO 260
280 INPUT "DAY OF WEEK (1-7 SUN=1):";DOW
290  IF DOW<1 OR DOW>7 GOTO 280
300 INPUT "    DAY OF MONTH (1-31):";DOM
310  IF DOM<1 OR DOM>31 GOTO 300
320 PRINT
330 INPUT "            HOUR (0-23):";HR
340  IF HR<0 OR HR>23 GOTO 330
350 INPUT "         MINUTES (0-59):";MIN
360  IF MIN<0 OR MIN>59 GOTO 350
370 INPUT "         SECONDS (0-59):";SEC
380  IF SEC<0 OR SEC>59 GOTO 370
390 REM CONVERT REAL TIME AND DATE FROM DECIMAL TO BCD
400 REM ------------------------------------------------
410 MTH=FNTOBCD(MTH)          'CONVERT MONTH
420 DOM=FNTOBCD(DOM)          'CONVERT DATE
430 DOW=FNTOBCD(DOW)          'CONVERT DAY OF WEEK
440 HR=FNTOBCD(HR)            'CONVERT HOUR
450 MIN=FNTOBCD(MIN)          'CONVERT MINUTES
460 SEC=FNTOBCD(SEC)          'CONVERT SECOND
470 REM SET REAL TIME AND DATE AS ENTERED
480 REM ------------------------------------
490  OUT BADD,1: OUT BADD+7,MTH       'SET MONTH
500              OUT BADD+6,DOM       'SET DAY OF MONTH
510              OUT BADD+5,DOW       'SET DAY OF WEEK
520              OUT BADD+4,HR        'SET HOUR
530  OUT BADD,0: OUT BADD+7,MIN       'SET MINUTES
540              OUT BADD+6,SEC       'SET SECONDS
550 REM
560 REM GET TIME AND DATE FROM RTC
570 REM ---------------------------
580 PRINT CHR$(26)          'CLEAR SCREEN AND CURSOR HOME
585 LI=0 : PO=0 : GOSUB 1520 : PRINT "ENTER 'Q' TO QUIT PROGRAM"
590 FLAG=0
600 REM GET TIME AND PRINT ONCE PER SECOND
610  OUT BADD,0: X=INP(BADD+6): SEC=FNTODEC(X)    'READ SECOND
620 IF SEC=59 THEN ADDSEC=0 ELSE ADDSEC=SEC+1     'SET ROLLOVER CHECK
630  OUT BADD,0: X=INP(BADD+6): SEC=FNTODEC(X)    'READ SECONDS
640 IF SEC<>ADDSEC GOTO 630        'IF A SECOND HAS PASSED, GET THE TIME
650  OUT BADD,0: X=INP(BADD+7): MIN=FNTODEC(X)    'GET MINUTE
660  OUT BADD,1: X=INP(BADD+4): HR=FNTODEC(X)     'GET HOURS
```

```
670  X=INP(BADD+5): DOW=FNTODEC(X)      'GET DAY OF WEEK
680  X=INP(BADD+6): DOM=FNTODEC(X)      'GET DAY OF MONTH
690  X=INP(BADD+7): MTH=FNTODEC(X)      'GET MONTH
700 IF ADDSEC<>0 AND FLAG=1 GOTO 800
710 REM PRINT REAL TIME AND DATE
720 REM ------------------------
730 PRINT CHR$(30): PRINT  CHR$(23)   'CURSOR HOME & CLEAR END OF LINE
740 ON DOW GOSUB 910,920,930,940,950,960,970
750 ON MTH GOSUB 1010,1020,1030,1040,1050,1060,1070,1080,1090,1100,1110,
    1120
760 PRINT "DATE: "; DOW$;", ";MTH$;" ";DOM;
770 LI=2 : PO=40 : GOSUB 1520                    'POS. HR,MIN
780 PRINT USING "TIME: ## : ## :";HR, MIN
790 FLAG=1
800 LI=2 :PO=55:  GOSUB 1520: PRINT CHR$ (23)      'CLEAR END OF LINE
810 LI=2 :PO=55:  GOSUB 1520                       'POS. SEC
820 PRINT  SEC;
830 REM QUIT PROGRAM ?
840 REM ---------------
850 SEL$=INKEY$
860 IF SEL$="Q" GOTO 140
870 GOTO 620
880 REM
890 REM CONVERT DAY OF WEEK NUMBERS TO DAY OF WEEK NAMES
900 REM ------------------------------------------------
910 DOW$="Sunday": RETURN
920 DOW$="Monday": RETURN
930 DOW$="Tuesday": RETURN
940 DOW$="Wednesday": RETURN
950 DOW$="Thursday": RETURN
960 DOW$="Friday": RETURN
970 DOW$="Saturday": RETURN
980 REM
990 REM CONVERT MONTH NUMBERS TO MONTH NAMES
1000 REM --------------------------------------
1010 MTH$="January": RETURN
1020 MTH$="February": RETURN
1030 MTH$="March": RETURN
1040 MTH$="April": RETURN
1050 MTH$="May": RETURN
1060 MTH$="June": RETURN
1070 MTH$="July": RETURN
```

```
1080 MTH$="August": RETURN
1090 MTH$="September": RETURN
1100 MTH$="October": RETURN
1110 MTH$="November": RETURN
1120 MTH$="December": RETURN
1130 REM
1140 REM RTC AVAILABLE ? TEST
1150 REM ------------------
1160 REM
1170 REM DEFAULT BADD = 200 DEC
1180 BADD=200
1190 OUT BADD,3: X=INP(BADD+7)
1200 OUT BADD+7,11
1210 T=INP(BADD+7)
1220 OUT BADD+7,X
1230 IF T=11 THEN GOTO 1430
1240 PRINT  CHR$(26)
1250 PRINT "TABLE FOR BASE PORT ADDRESS (BADD):"
1260 PRINT
1270 PRINT "IFSEL-   BADD    BADD"
1280 PRINT "SWITCH   (HEX)   (DEC)"
1290 PRINT "-------------------"
1300 PRINT "0-A      60      96"
1310 PRINT "0-B      68      104"
1320 PRINT "1-A      70      112"
1330 PRINT "1-B      78      120"
1340 PRINT "2-A      30      48"
1350 PRINT "2-B      38      56"
1360 PRINT "3-A      B0      176"
1370 PRINT "3-B      B8      184"
1380 PRINT "4-A      C0      192"
1390 PRINT "4-B      C8      200"
1400 PRINT
1410 INPUT "INPUT THE BASE PORT ADDRESS (DEC) OF THE RTC (SEE TABLE):",
     BADD
1420 IF BADD=96 OR BADD=104 OR BADD=112 OR BADD=120 OR BADD=48 OR
     BADD=56 OR BADD=176 OR BADD=184 OR BADD=192 OR BADD=200 THEN
     GOTO 1190
     ELSE GOTO 1410
1430 PRINT CHR$(26)
1440 PRINT "RTC AVAILABLE ON PORT ADDRESS (DEC): " ;BADD
1450 PRINT
```

```
1460 BADD$=STR$(BADD) : OPEN "O",#1,"BADD" : WRITE #1,BADD$ : CLOSE #1
1470 RETURN
1480 REM
1490 REM CURSOR POS
1500 REM ----------
1510 REM
1520 PRINT CHR$(27);CHR$(61);CHR$(32+LI);CHR$(32+PO);
1530 RETURN
1540 END
```

BEISPIEL 2

ZWECK: DIESES PROGRAMM ZEIGT DIE UHRZEIT UND DAS DATUM AN UND ERLAUBT DAS SETZEN DER ALARM-"LATCHES". SOBALD EIN ALARMZEITPUNKT ERREICHT WIRD, ERFOLGT DIE AUSGABE EINER MELDUNG.

BESCHREIBUNG DES PROGRAMMS:

ZEILENNR.:	KURZBESCHREIBUNG:
50 - 90	ROUTINE ZUR UMWANDLUNG VON DEZIMALZAHLEN IN BCD-ZAHLEN UND ZURÜCK.
100 - 110	PORTADRESSE (STANDARDVOREINSTELLUNG) WIRD VON DER DISKETTE GELESEN.
120 - 160	RÜCKSETZUNG DER ALARM-"LATCH"-REGISTER DER ECHTZEITUHR.
170 - 500	ZEIT- UND DATUMANZEIGE DER ECHTZEITUHR WIE IN BEISPIEL 1. ZUERST WIRD UM EINE EINSTELLUNG DER ALARMZEIT GEBETEN. DAS VORLIEGEN EINES ALARMS WIRD ABGEFRAGT. BEIM VORLIEGEN EINES ALARMS WIRD EINE NACHRICHT AUSGEGEBEN. UM EINE KORREKT FORMATIERTE AUSGABE ZU ERZIELEN WIRD "PRINT USING" VERWENDET.
510 - 620	ABFRAGE, OB TASTATUREINGABE VORLIEGT DURCH DIE HAUPTSCHLEIFE, DIE SEKUNDEN ANZEIGT. "Q"=BEENDEN DES PROGRAMMS, "A"=ALARM. ZUSÄTZLICH WIRD DER ALARM AKTIVIERT.
630 - 720	UMWANDLUNG VOM WOCHENTAG-WERT IN DEN WOCHENTAG-NAMEN.

730 - 870 UMWANDLUNG VOM MONAT-WERT IN DEN MONAT-NAMEN.

880 - 1170 DIESE ROUTINE WIRD AUFGERUFEN, WENN SIE DEN ALARM STELLEN MÖCHTEN. ZUERST WIRD DER ALARM, FALLS GESETZT, UNWIRKSAM GEMACHT UND DIE WERTE DES VORHERIGEN ALARMS GELÖSCHT. EINE BILDSCHIRMANZEIGE ERBITTET DIE EINGABE DES DATUMS, DER ZEIT UND EINER NACHRICHT. UNKORREKTE WERTE WERDEN ABGEFANGEN. LEEREINGABE VON <CR> BEWIRKT DIE BELEGUNG DER ALARM-"LATCH"-REGISTER MIT "IGNORIERE".

1180 - 1300 DARSTELLUNG DES ALARMZEITPUNKTES IM KORREKTEN FORMAT.

1310 - 1390 UMWANDLUNG DES DEZIMALWERTES IN BCD. "IGNORIERE" BELEGT DIE LATCHES MIT 204 DEZIMAL = 11001100 BINÄR.

1400 - 1490 SETZEN DER ALARMLATCHES MIT DEN ERMITTELTEN WERTEN, AUSGABE DER AKTUELLEN ZEIT UND DES DATUMS

1500 - 1610 UNTERROUTINE ZUR AUSGABE DER ALARM-NACHRICHT UND ERZEUGUNG EINES AKUSTISCHEN SIGNALS.

1620 - 1650 POSITIONIERUNG DER SCHREIBMARKE (CURSOR)

1660 - 1670 LÖSCHEN BIS ENDE BILDSCHIRM

```
10 REM                      EXAMPLE - 2
20 REM  SET THE LATCHES ON THE REAL TIME CLOCK AND WAIT FOR ALARM
30 REM                        K 8 0 3
40 REM  ------------------------------------------------------------
50 REM DEFINE A FUNCTION TO CONVERT FROM DECIMAL TO BCD
60 DEF FNTOBCD(X)=(X\10)*6+X
70 REM
80 REM DEFINE A FUNCTION TO CONVERT FROM BCD TO DECIMAL
90 DEF FNTODEC(X)=(X\16)*10 + X MOD 16
100 OPEN "I",#1,"BADD" : INPUT #1,BADD$ : BADD=VAL(BADD$) : CLOSE #1
110 REM
120 REM LATCH RESET
130 REM ----------
140 OUT BADD,4
150 OUT BADD+7,255
160 REM
```

```
170 REM GET TIME AND DATE FROM RTC
180 REM ------------------------
190 REM
200 PRINT CHR$(26)        'CLEAR SCREEN AND CURSOR HOME
210 LI=0 : PO=0 : GOSUB 1610
220 PRINT "ENTIER 'A' TO SET ALARM, 'Q' TO QUIT PROGRAM "
230 FLAG=0
240 REM GET TIME AND PRINT ONCE PER SECOND
250  OUT BADD,0: X=INP(BADD+6): SEC=FNTODEC(X)     'READ SECOND
260 IF SEC=59 THEN ADDSEC=0 ELSE ADDSEC=SEC+1      'SET ROLLOVER CHECK
270  OUT BADD,0: X=INP(BADD+6): SEC=FNTODEC(X)     'READ SECONDS
280 REM DETECT FOR ALARM
290  OUT BADD,4          'INT. CONTR./STATUS REG.
300 XX=INP(BADD+4)       'READ INT. STATUS REG.
310 IF XX<>0 THEN GOSUB 1500 'DISPLAY THAT INT. HAS OCCURRED
320 IF SEC<>ADDSEC GOTO 270        'IF A SECOND HAS PASSED, GET THE TIME
330  OUT BADD,0: X=INP(BADD+7): MIN=FNTODEC(X)     'GET MINUTE
340  OUT BADD,1: X=INP(BADD+4): HR=FNTODEC(X)      'GET HOURS
350  X=INP(BADD+5): DOW=FNTODEC(X)      'GET DAY OF WEEK
360  X=INP(BADD+6): DOM=FNTODEC(X)      'GET DAY OF MONTH
370  X=INP(BADD+7): MTH=FNTODEC(X)      'GET MONTH
380 IF ADDSEC<>0 AND FLAG=1 GOTO 470
390 REM PRINT REAL TIME AND DATE
400 PRINT CHR$(30): PRINT  CHR$(23)      'CURSOR HOME & CLEAR LINE
410 ON DOW GOSUB 650,660,670,680,690,700,710
420 ON MTH GOSUB 750,760,770,780,790,800,810,820,830,840,850,860
430 PRINT "DATE: ", DOW$;", ",MTH$,DOM;
440 LI=2 : PO=55 : GOSUB 1610           'POS. HR,MIN
450 PRINT USING "TIME: ## : ## :";HR, MIN
460 FLAG=1
470 LI=2 :PO=70:  GOSUB 1610: PRINT CHR$ (23)   'CLEAR END OF LINE
480 LI=2 :PO=70:  GOSUB 1610                    'POS. SEC
490 PRINT USING " ##"; SEC;
500 REM
510 REM QUIT PROGRAM ? OR SET LATCHES ?
520 REM ------------------------------
530 SEL$=INKEY$
540 IF SEL$="Q" GOTO 1680
550 IF SEL$="A" THEN GOTO  880      'ENTRY FOR LATCH
560 REM ENABLE INTERRUPT ON LATCH ALARM
570  OUT BADD,4        'INT. CONTR.& STATUS REG.
580  OUT BADD+5,1      'SELECT INT. CONTR. REG & ENABLE ALARM
```

```
590 REM
600 REM
610 GOTO 250          'WAIT FOR NEXT SEC
620 REM
630 REM CONVERT DAY OF WEEK NUMBERS TO DAY OF WEEK NAMES
640 REM ------------------------------------------------
650 DOW$="Sunday": RETURN
660 DOW$="Monday": RETURN
670 DOW$="Tuesday": RETURN
680 DOW$="Wednesday": RETURN
690 DOW$="Thursday": RETURN
700 DOW$="Friday": RETURN
710 DOW$="Saturday": RETURN
720 REM
730 REM CONVERT MONTH NUMBERS TO NAMES
740 REM ------------------------------
750 MTH$="January": RETURN
760 MTH$="February": RETURN
770 MTH$="March": RETURN
780 MTH$="April": RETURN
790 MTH$="May": RETURN
800 MTH$="June": RETURN
810 MTH$="July": RETURN
820 MTH$="August": RETURN
830 MTH$="September": RETURN
840 MTH$="October": RETURN
850 MTH$="November": RETURN
860 MTH$="December": RETURN
870 REM
880 REM S E T  L A T C H
890 REM ---------------
900 OUT BADD,4 : OUT BADD+5,0 : YY=INP(BADD+4)   'DISABLE ALARM,
    CLEAR INT.
910 LI=8 : PO=0 : GOSUB 1610 : GOSUB 1660   'CURSOR POS. & CLEAR END OF
    SCREEN
920 PRINT TAB(10); " E N T E R   A L A R M  --  CR = Don`t care"
930 PRINT
940 INPUT "DAY OF WEEK (1-7 SUN=1):";LDOW$:LDOW=VAL(LDOW$)
950   IF LEN(LDOW$)=0 GOTO 970
960   IF LDOW<1 OR LDOW>7 THEN  PRINT CHR$(11) CHR$(23);:GOTO 940
970 INPUT "           MONTH (1-12):";LMTH$:LMTH=VAL(LMTH$)
980   IF LEN(LMTH$)=0 GOTO 1000
```

```
990   IF LMTH<1 OR LMTH>12 THEN PRINT CHR$(11) CHR$(23);:GOTO 970
1000 INPUT "    DAY OF MONTH (1-31):";LDOM$:LDOM=VAL(LDOM$)
1010   IF LEN(LDOM$)=0 GOTO 1030
1020   IF LDOM<1 OR LDOM>31 THEN PRINT CHR$(11) CHR$(23);:GOTO 1000
1030 PRINT
1040 INPUT "            HOUR (0-23):";LHR$:LHR=VAL(LHR$)
1050   IF LEN(LHR$)=0 GOTO 1070
1060   IF LHR<0 OR LHR>23 THEN PRINT CHR$(11) CHR$(23);:GOTO 1040
1070 INPUT "         MINUTES (0-59):";LMIN$:LMIN=VAL(LMIN$)
1080   IF LEN(LMIN$)=0 GOTO 1100
1090   IF LMIN<0 OR LMIN>59 THEN PRINT CHR$(11) CHR$(23);:GOTO 1070
1100 INPUT "         SECONDS (0-59):";LSEC$ :LSEC=VAL(LSEC$)
1110   IF LEN(LSEC$)=0 GOTO 1130
1120   IF LSEC<0 OR LSEC>59 THEN PRINT CHR$(11) CHR$(23);:GOTO 1100
1130 PRINT: INPUT " ALARM-MESSAGE (MAX.50):";MESSAGE$
1140 ON LDOW GOSUB 650,660,670,680,690,700,710
1150 ON LMTH GOSUB 750,760,770,780,790,800,810,820,830,840,850,860
1160 GOSUB 1610 : GOSUB 1660    'CURSOR POS. & CLEAR END OF SCREEN
1170 REM
1180 REM DISPLAY ALARM TIME & DATE
1190 REM ------------------------
1200 LI=3 : PO=0
1210 GOSUB 1610 :PRINT CHR$(23)      'CURSORPOS. & CLEAR LINE
1220 PRINT "ALARM:",
1230   IF LEN(LDOW$)=0 THEN PRINT "--,",ELSE PRINT DOW$;",",
1240   IF LEN(LMTH$)=0 THEN PRINT "--",ELSE PRINT MTH$,
1250   IF LEN(LDOM$)=0 THEN PRINT " -",ELSE PRINT " ";LDOM$,
1260 LI=4: PO=61 : GOSUB 1610        'CURSOR POS.
1270   IF LEN(LHR$)=0 THEN PRINT USING "& :";"--",
       ELSE PRINT USING "## :";VAL(LHR$),
1280   IF LEN(LMIN$)=0 THEN PRINT USING " & :";"--",
       ELSE PRINT USING " ## :";VAL(LMIN$),
1290   IF LEN(LSEC$)=0 THEN PRINT USING " &";"--",
       ELSE PRINT USING " ##";VAL(LSEC$);
1300 REM
1310 REM CONVERT DATA TO BCD, DETERMINE (DON'T CARE) LOCATIONS
1320 REM ------------------------------------------------------
1330   IF LEN(LMTH$)=0 THEN LMTH=204 ELSE LMTH=FNTOBCD(LMTH)
1340   IF LEN(LDOW$)=0 THEN LDOW=204 ELSE LDOW=FNTOBCD(LDOW)
1350   IF LEN(LDOM$)=0 THEN LDOM=204 ELSE LDOM=FNTOBCD(LDOM)
1360   IF LEN(LHR$)=0 THEN LHR=204 ELSE LHR=FNTOBCD(LHR)
1370   IF LEN(LMIN$)=0 THEN LMIN=204 ELSE LMIN=FNTOBCD(LMIN)
```

```
1380   IF LEN(LSEC$)=0 THEN LSEC=204 ELSE LSEC=FNTOBCD(LSEC)
1390 REM
1400 REM SET TIMES AND (DON'T CARES) INTO THE RTC LATCHES
1410 REM ------------------------------------------------
1420  OUT BADD,2 : OUT BADD+6,LSEC
1430               OUT BADD+7,LMIN
1440  OUT BADD,3 : OUT BADD+4,LHR
1450               OUT BADD+5,LDOW
1460               OUT BADD+6,LDOM
1470               OUT BADD+7,LMTH
1480 FLAG=0 : GOTO 250      'WAIT FOR NEXT SEC
1490 REM
1500 REM OUTPUT MESSAGE
1510 REM --------------
1520 LI=20: PO=0
1530 GOSUB 1610 :PRINT ;MESSAGE$;
1540 RESTORE
1550 FOR I=1 TO 4
1560 READ TRE,VOL
1570 PRINT CHR$(27);CHR$(77);CHR$(32+TRE);CHR$(32+VOL);
1580 DATA 20,10,30,10,40,00,30,5
1590 NEXT
1600 RETURN
1610 REM
1620 REM CURSOR POSITION
1630 REM ---------------
1640 PRINT CHR$(27);CHR$(61);CHR$(32+LI);CHR$(32+PO);
1650 RETURN
1660 PRINT CHR$(27);CHR$(121)
1670 RETURN
1680 END
```

INFORMATION FÜR ASSEMBLER PROGRAMMIERER

Das Lesen des Interrupt-(Unterbrechungs-)Status-Registers löscht dieses Register. Das Interrupt-Status-Register könnte genau dann gelesen werden, wenn ein Interrupt auftritt. Das kann eine Störung verursachen und der Interrupt könnte vom Lesekommando übersehen werden. Vergewissern Sie sich, daß in Assemblerprogrammen das Interrupt-Status-Register dann gelesen wird, wenn kein Interrupt erwartet wird.

Beispiel:

```
INSTAT    LESE ZÄHLER 1/1000 SEK. IN ALPHA
          LESE UMSTELLUNGKONTROLLBIT (ROLLOVER-BIT)
          WENN UMSTELLUNGKONTROLLBIT = 1, SPRUNG NACH INSTAT
LOOP      LESE ZÄHLER 1/1000 SEK. IN BETA
          LESE UMSTELLUNGKONTROLLBIT
          WENN UMSTELLUNGKONTROLLBIT = 1, SPRUNG NACH LOOP
          WENN ALPHA = BETA, SPRUNG NACH LOOP
          WARTE 0,1 MS.  *
          LESE INTERRUPT STATUS REGISTER
```

* = Nächster Interrupt kann nach 0,9 ms auftreten.