

87-4012

4012 MAJ 83(A) 1 13

CONTENTS

1. Description
 2. Technical data
 3. Jumpers
 4. Real Time Clock registers
 5. I/O commands
 6. Block diagram
 7. Component diagram
 8. Program example
- App.1 ABC800 progr. ex.

DESCRIPTION

The 4012 DataBoard module contains:

- 1: Real Time Clock (RTC) with battery backup.
- 2: Separate Interval clock for program interval timing.
- 3: 7 status switches for program use.

Interval clock:

Interval timing is by regular interrupts or by clock status readings. The on-board crystal oscillator is set by jumpers to give a clock tic each 1 ms, 10 ms or 100 ms. At each tic the interval status (bit 0 in the STAT command) is complemented. If interrupt is enabled on the 4012, the program must reset the interval clock status after each interrupt, using the C2 command, as interrupt is generated by a low level on the clock status bit.

The interval clock is required when using the OS-8 operating system, with interrupt wired to level 0 (I/O-slot pin 5A to pin 13B on control card slot).

INNEHÅLL

1. Beskrivning
 2. Tekniska data
 3. Byglingar
 4. Realtidsklock-register
 5. I/O kommandon
 6. Blockdiagram
 7. Komponentdiagram
 8. Program exempel
- App.1 ABC800 progr. ex.

BESKRIVNING

DataBoard modulen 4012 innehåller:

- 1: Realtidsklocka (RTC) med batteribackup.
- 2: Separat intervallklocka för intervall-tidtagning.
- 3: 7 statusomkopplare för användning från program.

Intervallklockan:

Intervallklockan kan antingen ge regelbundna interrupt eller läsas av som klockstatus. Kristalloscillatorn på kortet sätts av byglingar att ge en signal varje 1 ms, 10 ms eller 100 ms. Vid signal byts nivå på intervallstatus (Bit 0 i STAT). Om interrupt är påkopplat på 4012, måste programmet återställa status med C2-kommandot efter varje interrupt, eftersom interrupt genereras vid låg nivå på intervallstatusbiten.

Intervallklockan krävs för OS-8 med interruptnivå 0 virad till 4012 (I/O-stift 5A till stift 13B på styrkortet).

7 Status Jumpers:

The jumpers in the card position 2F are read by the STAT command, and are for optional use. The jumper no.1 is used by some system program as the Bootstrap loader and the disc selfstart program. The other jumpers can be used to set the year, which is not available in the Real Time Clock chip. The jumpers have the status '0' when closed.

Real Time Clock:

The 4012 includes the MM58174 RTC circuit with battery back-up and is normally in a access-protected mode, opened for reading or writing only during 0.5 ms after an OUT-instruction to the RTC chip.

The battery backup circuit includes +5V power fail detection with hardware protect at power failure. In addition a manual switch is available to give full hardware protection.

Open: (Down, towards LED).
Hardware protected.
Closed: Can be write and read enabled by program.

The Real Time Clock chip has 16 read/write registers for control and timing data from 1/10 of a second up to months, weekday number and a leap year counter. The time is written and read one BCD digit per register.

The battery backup lasts for at least 3 months back-up.

NOTE! Open jumper S4 if the 4012 shall be in storage to disconnect the batteries. The battery should during storage be recharged regularly 2 or 3 times/year to keep the batteries in good condition.

7 Status Byglingar:

Byglingarna i kortposition 2F läses av STAT kommandot och är för valfri användning. Bygling 1 används för vissa systemprogram som Bootstrap och självstartprogrammet. Övriga byglingar kan användas för lagring av årtal, vilket ej explicit finns i Real-Tidsklockan. Bygel sluten ger status "0".

Realtidsklockan:

Klock-kretsen MM58174 används på 4012 och är normalt i access-skyddad mod. Kretsen öppnas för läsning eller skrivning enbart under 0.5 ms efter en OUT-instruktion till RTC-kretsen.

Kretsen har batteri-backup med automatiskt skydd vid spänningsavbrott på +5V - spänningen. Dessutom finns en manuell omkopplare för ovillkorligt skydd.

Öppen: (Ner, mot Lysdioden).
Skrivskydd.
Sluten: Kretsen kan öppnas för skrivning och läsning med en programinstruktion.

Realtidsklock-kretsen har 16 läs/skriv register för styrning och tids-data från 1/10 sekund upp till månadsnummer och veckodag, samt skottårsräknare. Tiden skrivs och läses en BCD siffra i taget.

Batterierna räcker i minst tre månader.

OBS! Öppna bygling S4 när 4012 ligger i lager för att koppla ur batterierna. Batterierna bör under lagringstiden laddas regelbundet 2 eller 3 gånger/år för att hållas i kondition.

TECHNICAL DATA

TEKNISKA DATA

Power supply
Spänningsmatning

+5V +-5% mA

Bus connection
Bussanslutning

On the I/O side of the DataBoard bus.
with Card address 0.

På I/O sidan av DataBoard bussen,
med kortadressen 0.

Bus connector
Buskontakt

B 64 pin Euroconnector plug DIN 41612.
Typ B 64 stifts Eurokontakt, hona, DIN 41612

Size
Storlek

Standard Eurocard 100 * 160 mm.

Bus pin numbering
Stiftsnumrering

Standard. See system manual.

Interval clock
Intervallklocka

1 ms, 10 ms or 100 ms jumper selectable
Interrupt available. On-board crystal.

1 ms, 10 ms eller 100 ms väljs med
byglingar. Interrupt finns. Kristall
på kortet.

Status jumpers
Statusbyglingar

7 jumpers can be read as status bits.
7 byglingar kan läsas som statusbitar.

Real Time Clock
Realtidsklocka

Circuit MM58174 (Nat. Semicond.)
Time: 1/10 second -- months, weekdays.
Leap year control for correct timing also
for the 28/29 of february.

Krets: MM58174 (Nat. Semicond.)
Tid: 1/10 sekund -- månader, veckodags-
siffra. Skottårsräknare för korrekt
datum även för 28/29 februari.

Battery backup
Batteri backup

Battery backup for the Real Time Clock.
Including automatic access protection
at power failure.
Switch for hardware protect.

Batteri backup för Realtidsklockan, med
automatiskt access-skydd vid spänningsfall.
Omkopplare för manuellt skydd.

Batteries
Batterier

Three batteries, VARTA 100 RS, each
1.2 V nominal.

Tre batterier, VARTA 100 RS, vardera
1.2 V nominellt.

=====

This datasheet information is subject to change without notice.

JUMPERS

BYGLINGAR

S1,S2,S3 Close only ONE!!!
 S1 Closed: 100 ms interval.
 S2 Closed: 10 ms interval.
 S3 Closed: 1 ms interval.

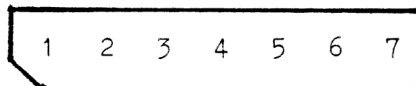
S1,S2,S3 Slut endast EN!!!
 S1 Sluten: 100 ms intervall.
 S2 Sluten: 10 ms intervall.
 S3 Sluten: 1 ms intervall.

S4 Open to disconnect the
batteries.

S4 Öppna för att koppla bort
batterierna.

The status jumpers in the card
position 2F corresponds to the
following status bits.

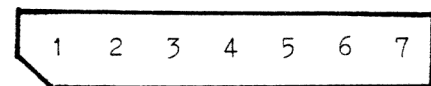
Statusbyglingarna i kort-
position 2F motsvarar följande
statusbitar.



Pin 1

(Bit 0 is interval status)

SW1 Open: Protected.
 Closed: Writing and
 reading can be
 enabled by the
 OUT-command.



Stift 1

(Bit 0 är intervallstatus)

SW1 Öppen: Skyddad.
 Stängd: Skyddet kan
 öppnas med OUT-
 kommandot.

THE REAL TIME CLOCK REGISTERS

REGISTREN I REALTIDSKLOCKAN.

The RTC contains 16 read/write
registers, each containing one
BCD digit or a command.

Kretsen innehåller 16 läs/
skriv-register, vardera med
en BCD siffra eller ett
kommando.

REG ACCESS USE

REG LÄS/SKRIV ANVÄNDNING

0	Write	Prod.Test only.
1	Read	1/10 sec. digit.
2	Read	Units of seconds.
3	Read	Tenth of seconds.
4	WR/RE	Units of minutes.
5	WR/RE	Tenth of minutes.
6	WR/RE	Units of hours.
7	WR/RE	Tenth of hours.
8	WR/RE	Units of date.
9	WR/RE	Tenth of date.
10	WR/RE	Day of week(1-7)
11	WR/RE	Units of months
12	WR/RE	Tenth of months.
13	Write	Leap year counter.
14	Write	Stop/Start(Bit 0)
15	Not used	in 4012 appl.

0	Write	Prod.test enbart.
1	Read	1/10 sekundsiffra
2	Read	Ental sekundsiffra
3	Read	10-tal sekundsiffra
4	WR/RE	Ental minutsiffra
5	WR/RE	10-tal minutsiffra
6	WR/RE	Ental timsiffra
7	WR/RE	10-tal timsiffra
8	WR/RE	Ental datum.
9	WR/RE	10-tal datum
10	WR/RE	Veckodag (1-7)
11	WR/RE	Ental månad
12	WR/RE	10-tal månad
13	Write	Skottårsräknare
14	Write	Stopp/Start(bit 0).
15	Används ej	på 4012 kortet.

For more detailed information on the Real Time Clock chip than given below, please see the product specification for the MM58174 from National Semicond.

För mer detaljerad information om Realtidsklock-kretsen än den kortfattade som ges nedan, hänvisas användaren till MM58174 produktspecifikation från National Semiconductor.

The register 0 shall always be initiated to 0.

Register 0 skall alltid initieras till 0.

The register 1-12 each contain one BCD digit of time data.

Register 1-12 innehåller var sin BCD siffra för tiden.

The seconds registers (1-3) are reset to 0 by writing a start/stop command to reg. 14.

Sekundregistren (1-3) sätts till 0 vid ett start/stop-kommando till reg. 14.

The other time digits are initiated by writing to the corresponding registers.

Övriga tidssiffror initieras genom att skriva till motsvarande register.

The leap year counter (reg.13) shall be initiated by writing a 4-bit value into reg.13. The leap year counter can not be read from the RTC circuit, but is updated automatically each year. Bit 0,1,2,3 in the counter indicate one of 4 possible values:

Skottårsräknaren (reg.13) ska initieras genom att skriva ett 4-bits värde i registret. Skottårsräknaren kan ej läsas från RTC kretsen, men räknas upp automatiskt varje årskifte. Bit 0,1,2,3 anger en av 4 möjliga värden:

Bit 3=1	(Value= 8)	Leap year.	Skottår.
Bit 2=1	(Value= 4)	Leap year+1.	År efter skottår.
Bit 1=1	(Value= 2)	Leap year+2.	2:a året efter skottår.
Bit 0=1	(Value= 1)	Leap year+3.	3:e året efter skottår.

A logic '1', written into bit 0 of the start/stop register 14, will start the clock with the seconds-counters zeroed. A logic '0' stops the clock.

En logisk 1:a, skriven till start/stopp registret 14, startar klockan med sekundräknarna nollställda. En logisk nolla stoppar klockan.

The register access to the RTC circuit on the 4012 is done with a sequence as below.

Läsning och skrivning från/till RTC registren på 4012 görs enligt följande princip.

The direct I/O commands can not be used from BASIC due to the delays in the interpreter. Use the routine in the program example.

Direkta I/O-kommandon kan inte användas från BASIC på grund av fördröjningarna i interpretatorn. Använd rutinen i program exemplet.

Note that if a register is updated during a read operation, the 'read' will return the illegal code '1111'. The user program should re-read the register in this case.

Obs! att den felaktiga koden '1111' returneras om ett register uppdateras under en läsoperation. Användarprogrammet bör då läsa om registret.

Writing:

Skrivning:

- OUT : Address register, open write enable for 0.5 ms and output data to temporary latch.
- C3 : Store data from latch to RTC-register.

- OUT : Adressera register, öppna skrivskyddet för 0.5 ms och lagra data i temporär buffert.
- C3 : Lagra data från buffert till RTC reg.

Reading:

Läsning:

- OUT : Address register.
- C1 : Read first digit into temporary latch. The address is incremented.
- INP : Input digit from 4012 and read next digit into the latch. The address is autoincremented. Continue if the BCD value is correct.

- OUT : Adressera register.
- C1 : Läs första siffran till temporär buffert. Adressen ökas med 1.
- INP : Läs siffra från 4012 och läs nästa siffra till buffert. Adressen ökas med ett varje gång. Forsätt om BCD-värdet är korrekt.

- INP
- INP

etc.

- INP
- INP

etc.

COMMANDS

KOMMANDON

Signal CS A=0. Select card. The 4012 is prewired with ASSEMB OUT 1 the card address 0. The LED on the card FORT80 OUTPUT(1)=A is turned on. PASCAL OUT (1,A) A=0. Välj kort. 4012 har 'noll' som fast kortadress. Lysdioden tänds.

Signal STAT Reads the interval clock status (bit 0) and ASSEMB INP 1 the 7 status jumpers. The interval status FORT80 A=INPUT(1) is complemented each clock tic unless reset. PASCAL A=INP(1) Läser intervallklockstatus (bit 0) och de 7 statusbyglingarna. Intervallstatus växlar vid varje klock-tick om den ej återställs.

Signal C2 Resets the interval clock status to '1', ASSEMB OUT 3 which also resets the interrupt signal. FORT80 OUTPUT(3)=A Återställer intervallklockstatus till '1', PASCAL OUT (3,A) vilket även återställer interruptsignalen.

```

=====
Signal C4          Enable/disable the interval clock interrupt
ASSEMB OUT 5       Data bit 7 = 1 : Enable
FORT80 OUTPUT(5)=A Data bit 7 = 0 : Disable
PASCAL OUT (5,A)   Interrupt is generated when the interval
                  status goes low. The interrupt signal
                  follows the interval status value unless
                  disabled.

```

Sätter på/stänger av interrupt från
intervallklockan. Bit 7=1 : Sätt på.
Bit 7=0 : Stäng av.

Interrupt genereras när intervallstatus
går låg. Interruptsignalen följer intervall-
status om interrupt är påslagen.

```

-----
Signal OUT          a) Select the register address in RTC
ASSEMB OUT 0        with data bits 4-7.
FORT80 OUTPUT(0)=A b) Store one BCD digit (data bits 0-3) in
PASCAL OUT (0,A)    a latch on 4012.
                   c) Enable writing for 0.5 ms. Within this
                   time, a C3 command can be given to store
                   the data to the RTC register.

                   a) Väljer register address i RTC med
                   data bitarna 4-7.
                   b) Lagra databit 0-3 i temporär buffer.
                   c) Öppna skrivskyddet undet 0.5 ms. Under
                   denna tid kan ett C3 kommando ges för att
                   lagra data från buffer till RTC kretsen.

```

```

-----
Signal C3           Store data from the latch to a register
ASSEMB OUT 4        in the Real Time Clock according to the
FORT80 OUTPUT(4)=A earlier selected register address.
PASCAL OUT (4,A)    Lagra data från buffer till ett register
                   i realtidsklockan.

```

```

-----
Signal C1           Read data to an intermediate latch on 4012
ASSEMB OUT 2        from the RTC register, earlier addressed by
FORT80 OUTPUT(2)=X an OUT-instruction. Also increment the
PASCAL OUT (2,X)    address to prepare for the next transfer.
                   Läser data från ett RTC register till en
                   temporär buffer på 4012. Adressen ökas med
                   ett inför nästa överföring.

```

```

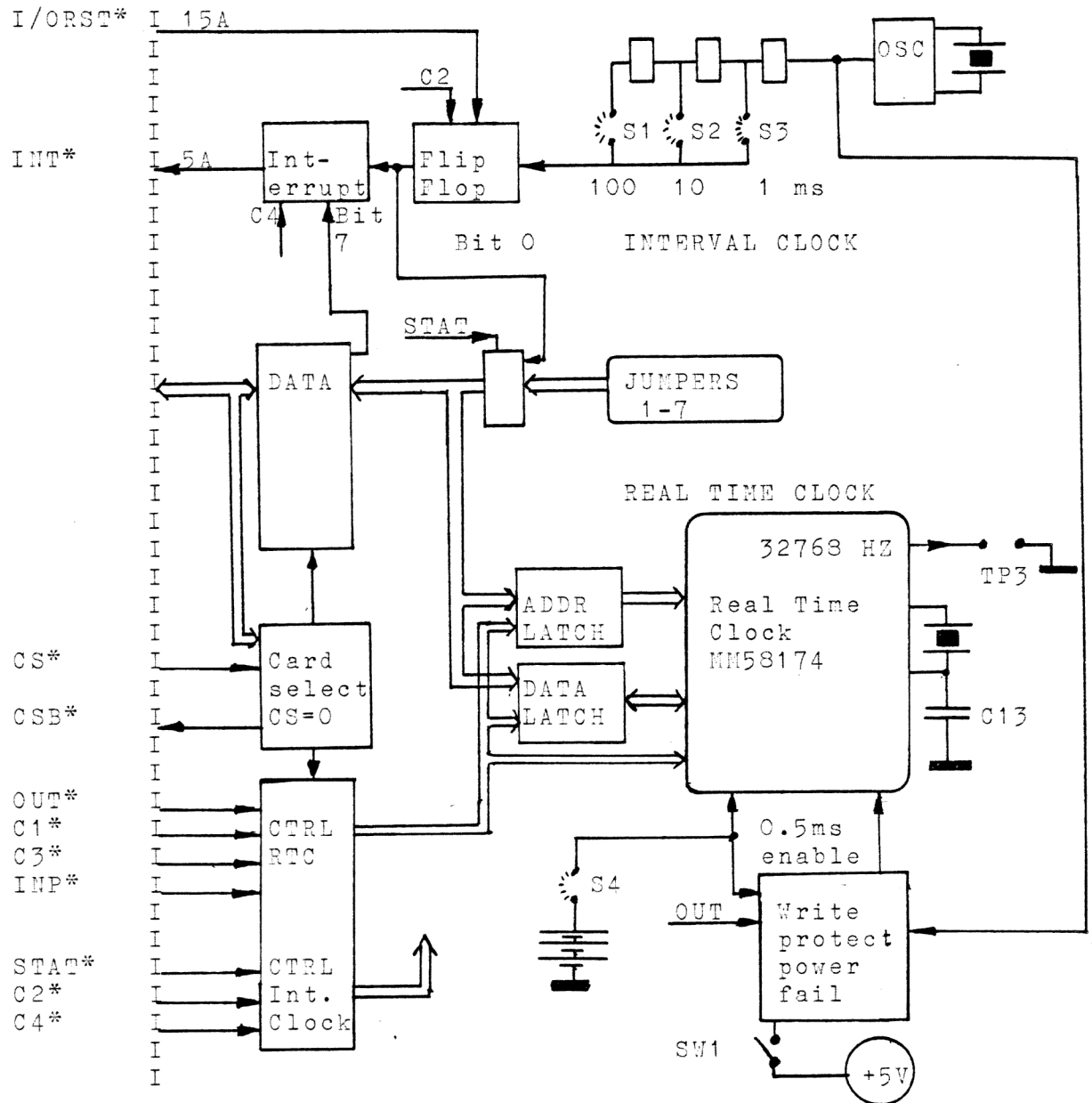
-----
Signal INP          Input data from the latch on 4012 to the
ASSEMB INP 0        CPU. Also initiate the next transfer as the
FORT80 A=INPUT(0)   C1 command.
PASCAL INP(0)       Läs in data till CPU från buffer på 4012.
                   Generera dessutom ett C1 kommando på 4012,
                   för att förbereda för läsning av nästa reg.

```

```

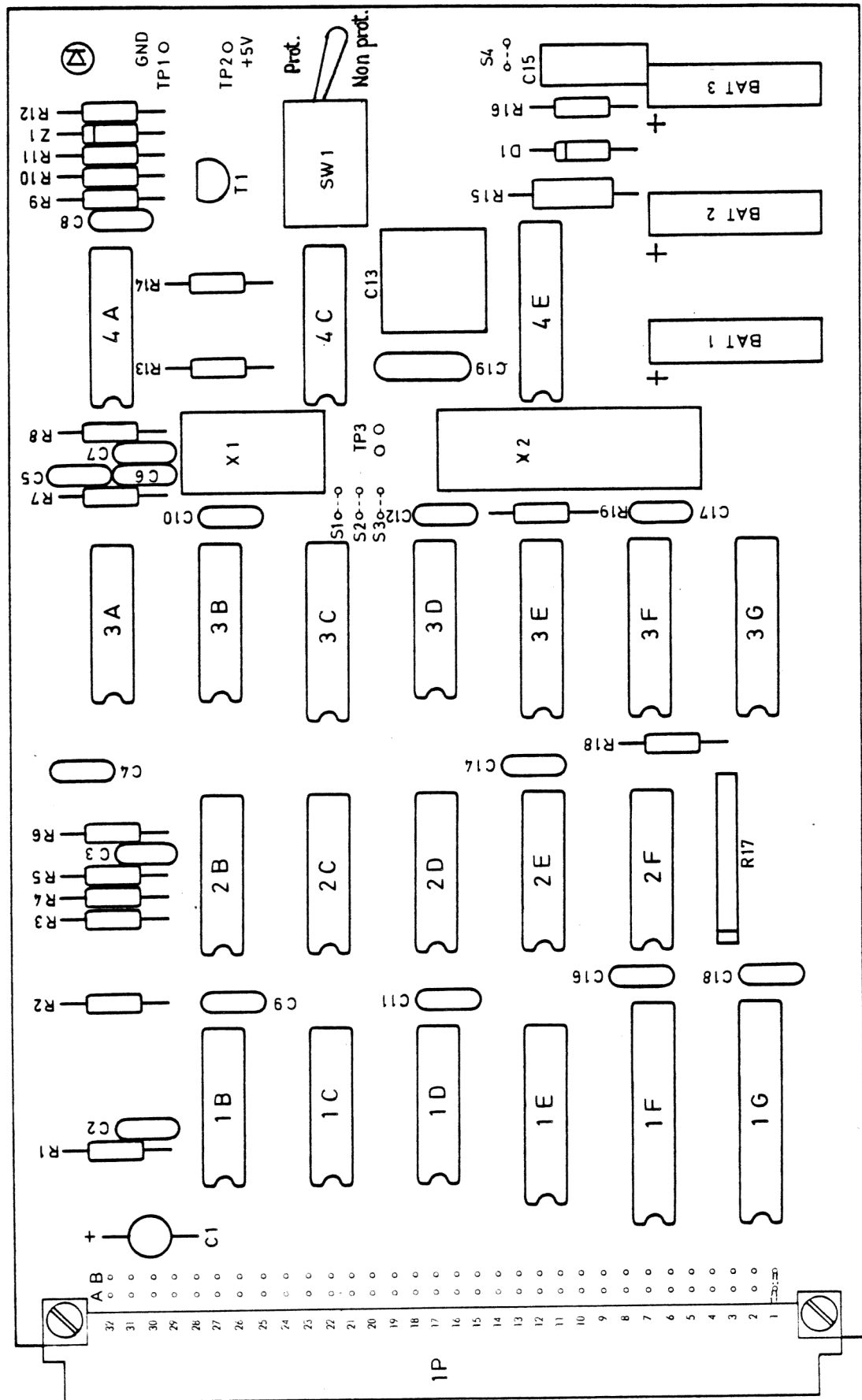
-----
Signal RST          Resets the interval clock status, disables
ASSEMB INP 7        interrupt and dis-selects the 4012 card.
FORT80 A=INPUT(7)   Återställer intervallklockstatus, stänger av
PASCAL INP(7)       interrupt och väljer bort 4012 kortet.
-----

```



COMPONENT DIAGRAM

KOMPONENTPLACERING



PROGRAM EXAMPLE

PROGRAM EXEMPEL

```

=====
File : RFklocka/AscBas
1982-07-01 15.48.16
=====

```

```

=====
10 INTEGER
20 EXTEND
W 30 DIM Månad(12) : DIM Veckodag$(10) : DIM Z$(20)
40 DATA Måndag,Tisdag,Onsdag,Torsdag,Fredag
50 DATA Lördag,Söndag
60 DATA 31,28,31,30,31,30,31,30,31,30,31,30,31
70 PRINT CHR$(26)
80 INPUT "Ställ tiden "+TIME$+" "T$ : IF LEN(T$)<>0 THEN SET TIME T$
G 90 IF RIGHT$(TIME$,18)="00" THEN S$=FNSet$(FNCmos$(TIME$)) ELSE 90 : Nollgenomgång
100 PRINT CHR$(26)
G 110 IF FNctime$=STRING$(15,ASCII("1")) THEN 110 ELSE : CUR(10,10) FNPrint$(FNctime$)
120 GOTO 110
130 DEF FNSet$(U1$) LOCAL A$=20,U$=20
140 :
150 :
160 :
170 :
180 :
190 :
200 :
210 :
220 :
230 :
240 :
250 :
260 :
270 :
280 :
290 :
300 :
310 :
320 :
330 :
340 :
350 :
360 :
370 :
380 :
390 :
400 :
410 :
420 :
=====
A$ är en assemblerutin som ställer tiden
Pekare för assembler rutin är VARPTR(A$)
Pekare för data till 4012 är VARPTR(U$)
Observera att sekunder ej går att ställa
Använd systemklockan för nollgenomgång
Assemblerutinen är som följer
=====
XR      A
DIS     CS
OUT     DATA
OUT     C3
L       A,(DE)
INCD    DE
CI      DE1H
JNZS    LABEL
OUT     DATA
OUT     C3
ENI
RET
=====
A$=CHR$(175,243,211,1,211,0,211,0,211,4,26,19,254,225,32,246,211,0,211,4,251,201)
U$=U1$
U$=CHR$(224)+U$+CHR$(225)
Z=CALL(VARPTR(A$),VARPTR(U$))
RETURN "0"
=====
FNEND

```

RFklocka/AscBas

```

430 DEF FNctime$ LOCAL Timeout,Ctime$=20,A$=60
440 :
450 :
460 :
470 :
480 :
490 :
500 :
510 :
520 :
530 :
540 :
550 :
560 :
570 :
580 :
590 A$=CHR$(235,175,211,1,62,32,211,0,211,2,1,0,11,237,178,201)+SPACE$(40)
600 OUT 1,0
610 Timeout=Timeout+1 : IF Timeout>4 THEN RETURN STRING$(15,ASCII("1")) : ! Läsnings fel.
620 Z=CALL(VARPTR(A$),VARPTR(A$)+20)
630 FOR I=20 TO 30
640 IF (PEEK(VARPTR(A$)+I) AND 15)=15 THEN 610
650 NEXT I
660 FOR I=30 TO 20 STEP -1
670 Ctime$=Ctime$+CHR$((15 AND PEEK(VARPTR(A$)+I))+48) : NEXT I : RETURN Ctime$
680 FNEND

```

=====

RFklocka/AscBas

=====

```

690 DEF FNVeckodag(T$)
700   Dag=VAL(MID$(T$,9,2))
710   IF Dag=Dag1 THEN RETURN Dag2-Dag2/7*7 ELSE Dag1=Dag
720   IF Dag2 THEN Dag2=Dag2+1 : RETURN (Dag2-Dag2/7*7)
730   Dag2=Dag1+5
740   FOR Lopär=1900 TO VAL(MID$(T$,1,4))-1
750     Dag2=Dag2+2+(Lopär/4*4<>Lopär) : NEXT Lopär
760     IF Veckodag$(0)="" THEN FOR Lop1=0 TO 6 : READ Veckodag$(Lop1) : NEXT Lop1
770     IF Månad(1)=0 THEN FOR Lop1=1 TO 12 : READ Q : W=W+Q : Månad(Lop1)=W : NEXT Lop1
780     Dag2=Dag2+Månad(VAL(MID$(T$,6,2))-1)
790     IF Lopär/4*4=Lopär AND Månad>2 THEN Dag2=Dag2+1
800     RETURN (Dag2-Dag2/7*7)
810 FNEND

```

W

```

820 DEF FNCmos$(Q$)
830   Z$="" : Y$=""
840   Z$=CHR$(VAL(MID$(Q$,1,4)) AND 3)
850   FOR S=6 TO 16
860     IF S=8 THEN Z$=Z$+CHR$(FNVeckodag(TIME$)+1)
870     IF S=8 OR S=11 OR S=14 THEN 890
880     Z$=Z$+CHR$(VAL(MID$(Q$,S,1)))
890   NEXT S
900   FOR X=1 TO 10 : Y$=Y$+CHR$(ASCII(MID$(Z$,X,1))+(14-X)*16) : NEXT X
910   RETURN Y$
920 FNEND

```

G

```

930 DEF FNPrint$(Q$)
940   PRINT "Tid från C-mosklocka "+MID$(TIME$,1,4)+"="+MID$(Q$,1,2)+"="+MID$(Q$,4,2):
950   ; "##"+MID$(Q$,6,2)+"-"+MID$(Q$,8,2)+"-"+MID$(Q$,10,2)
960   ; TAB(11) "I dag är det "+Veckodag$(VAL(MID$(Q$,3,1))-1)
970   PRINT TAB(11) "Tid från Systemet "+TIME$
980   RETURN ""
990 FNEND

```

[illegible]

```

5 ! ABC800 CMOS-KLOCKA
10 INTEGER
20 EXTEND
30 DIM MANAD%(12%) : DIM VECKODAG$(10%) : DIM Z$(20%)
40 DATA MANDAG,TISDAG,ONSDAG,TORSdag,FREDAG
50 DATA LÖRDAG,SÖNDAG
60 DATA 31,28,31,30,31,30,31,31,30,31,30,31
70 PRINT CHR$(12%) : ! RENSA SKÄRM
80 INPUT "STÄLL TIDEN "+TIME$+" ANNARS SKRIV IN NY TID "T$ : IF LEN(T$)<>0% THEN S$=FNSETTIME$(T$)
90 !
100 ; "DET GAR EJ ATT STÄLLA SEKUNDER PÅ 4012, SA NU VANTAR PROGRAMMET PÅ"
110 ; "ATT SEKUNDERNA I SYSTEMKLOCKAN SLAR OM TILL 00"
120 ; CUR(10,30) TIME$ : IF RIGHT$(TIME$,18%)="00" THEN S$=FNSET$(FNCMOS$(TIME$)) ELSE 120 ! NOLLGENOMGANG
130 PRINT CHR$(12%) : ! RENSA SKÄRM
140 IF FNCTIME$=STRING$(15%,ASCII("1")) THEN 140 ELSE : CUR(10%,10%) FNPRINT$(FNCTIME$)
150 GOTO 140
160 DEF FNSET$(U1$) LOCAL A$=20,U$=20
170 ! *****
180 ! A$ AR EN ASSEMBLER RUTIN SOM STÄLLER TIDEN
190 ! PEKARE FÖR ASSEMBLER RUTINEN AR VARPTR(A$)
200 ! PEKARE FÖR DATA TILL 4012 AR VARPTR(U$)
210 ! OBSERVERA ATT SEKUNDER EJ GA ATT STÄLLA
220 ! ANDVÄND SYSTEMKLOCKAN FÖR NOLLGENOMGANG
230 ! ASSEMBLER RUTINEN AR SOM FÖLJER
240 !
250 !
260 !
270 !
280 !
290 !
300 !
310 !
320 !
330 !
340 !
350 !
360 !
370 !
380 !
390 ! *****
400 A$=CHR$(175%,243%,211%,1%,211%,0%,211%,0%,211%,4%,26%,19%,254%,225%,32%,246%,211%,0%,211%,4%,251%,201%)
410 U$=U1$
420 U$=CHR$(224%)+U$+CHR$(225%) ! 224=14,0=STOP CLOCK, 225=14,1=START CLOCK
430 Z%=CALL(VARPTR(A$),VARPTR(U$))
440 RETURN "0"

```

O-STÄLL ACC. (REG. 0 SKALL ALLTID INITIERAS TILL 0)

STÄNG AV INTERRUPT

VALJ 4012

SKRIV TILL TEMP. BUFFER PÅ 4012

LAGRA DATA FRÅN BUFFERT TILL RTC-REG.

HÄMTA DATA

PEKA PÅ NÄSTA DATA

SLUT PÅ DATA (225 DECIMALT, SE RAD 420)

FÖRSÄTT TILLS ALLT DATA ÄR FLYTTAT

STARTA KLOCKAN MED SEKUNDER 0-STÄLLDA (E1H)

SKRIV TILL RTC-REGISTER

SÄTT PÅ INTERRUPT

ÅTER TILL BASIC

XR A

DIS

OUT CS

OUT DATA

OUT C3

L A,(DE)

INCD DE

CI OE1H

JNZS LABEL.

OUT DATA

OUT C3

ENI

RET

```

450 FNEND
460 DEF FNCTIME$ LOCAL TIMEOUT%,CTIME$=20,A$=60
470 ! *****
480 ! A$ AR EN ASSEMBLER RUTIN SAMT BUFFERT FÖR DATA FRÅN 4012
490 ! PEKARE FÖR ASSEMBLER RUTINEN AR VARPTR(A$)
500 ! PEKARE FÖR BUFFERT AR VARPTR(A$)+20
510 ! ASSEMBLER MEMONICS FÖR RUTINEN AR SOM FÖLJER
520 ! EXDR DATA PEKARE TILL REG. HL
530 ! XR A O-STALL ACCUMULATORN.
540 ! OUT CS VÄLJ 4012
550 ! LI A,32 BÖRJA MED REG. 2 (ENTAL SEKUNDER)
560 ! OUT DATA SKRIV TILL 4012
570 ! OUT C1 LÄS DATA FRÅN RTC-REG. TILL BUFFER PÅ 4012
580 ! LDI BC,13 ANTAL REGISTER VI VILL LÄSA
590 ! BIMP IR LÄS ALLA REGISTER
600 ! RET ÅTER TILL BASIC
610 ! *****
620 A$=CHR$(235%,175%,211%,1%,62%,32%,211%,0%,211%,2%,1%,0%,11%,237%,178%,201%)+SPACE$(40%)
630 OUT 1%,0%
640 TIMEOUT%=TIMEOUT%+1% : IF TIMEOUT%>4% THEN RETURN STRING$(15%,ASCII("1")) : ! LÄSNINGS FEL
650 Z%=CALL(VARPTR(A$),VARPTR(A$)+20%)
660 FOR I%=20% TO 30%
670 IF (PEEK(VARPTR(A$)+I%) AND 15%)=15% THEN 640
680 NEXT I%
690 FOR I%=30% TO 20% STEP -1%
700 CTIME$=CTIME$+CHR$(15% AND PEEK(VARPTR(A$)+I%))+48%) : NEXT I% : RETURN CTIME$
710 FNEND
720 DEF FNVECKODAG$(T$)
730 DAG%=VAL(MID$(T$,9%,2%))
740 IF DAG%=DAG%+1% THEN RETURN DAG2%-DAG2%/7%*7% ELSE DAG1%=DAG%
750 IF DAG2% THEN DAG2%=DAG2%+1% : RETURN (DAG2%-DAG2%/7%*7%)
760 DAG2%=DAG1%+5%
770 FOR LOPAR%=1900% TO VAL(MID$(T$,1%,4%))-1%
780 DAG2%=DAG2%+2%+(LOPAR%/4%*4%<>LOPAR%) : NEXT LOPAR%
790 IF VECKODAG$(0%)="" THEN FOR LOP1%=0% TO 6% : READ VECKODAG$(LOP1%) : NEXT LOP1%
800 IF MANAD%(1%)=0% THEN FOR LOP1%=1% TO 12% : READ Q% : W%=W%+Q% : MANAD%(LOP1%)=W% : NEXT LOP1%
810 DAG2%=DAG2%+MANAD%(VAL(MID$(T$,6%,2%))-1%)
820 IF LOPAR%/4%*4%=LOPAR% AND MANAD%>2% THEN DAG2%=DAG2%+1%
830 RETURN (DAG2%-DAG2%/7%*7%)
840 FNEND
850 DEF FNCMOS$(Q$)
860 Z$="" : Y$=""
870 Z$=CHR$(VAL(MID$(Q$,1%,4%))) AND 3%)
880 FOR S%=6% TO 16%
890 IF S%=8% THEN Z$=Z$+CHR$(FNVECKODAG%(TIME$)+1%)

```

```

900 IF SZ=8% OR SZ=11% OR SZ=14% THEN 9
910 Z$=Z$+CHR$(VAL(MID$(Q$,SZ,1%)))
920 NEXT SZ
930 FOR XZ=1% TO 10% : Y$=Y$+CHR$(ASCII(MID$(Z$,XZ,1%)))+(14%-XZ)*16%) : NEXT XZ
940 RETURN Y$
950 FNEND
960 DEF FNPRINT$(Q$)
970 ; "TID FRAN C-MOSKLOCKA "+MID$(TIME$,1%,4%)+ "="+MID$(Q$,1%,2%)+ "="+MID$(Q$,4%,2%);
980 ; "#"+MID$(Q$,6%,2%)+ "-" +MID$(Q$,8%,2%)+ "-" +MID$(Q$,10%,2%)
990 ; TAB(11%) "I DAG AR DET " VECKODAG$(VAL(MID$(Q$,3%,1%)))-1%)
1000 ; TAB(1%) "TID FRAN SYSTEMET "+TIME$
1010 RETURN ""
1020 FNEND
1030 DEF FNSETTIME$(T1$)
1040 POKE -17%,VAL(MID$(T1$,3,2)),VAL(MID$(T1$,6,2)),VAL(MID$(T1$,9,2)),VAL(MID$(T1$,12,2%)),VAL(MID$(T1$,15%,2%)),VAL(MID$(T1$,18%,2%))
1050 RETURN T1$
1060 FNEND

```