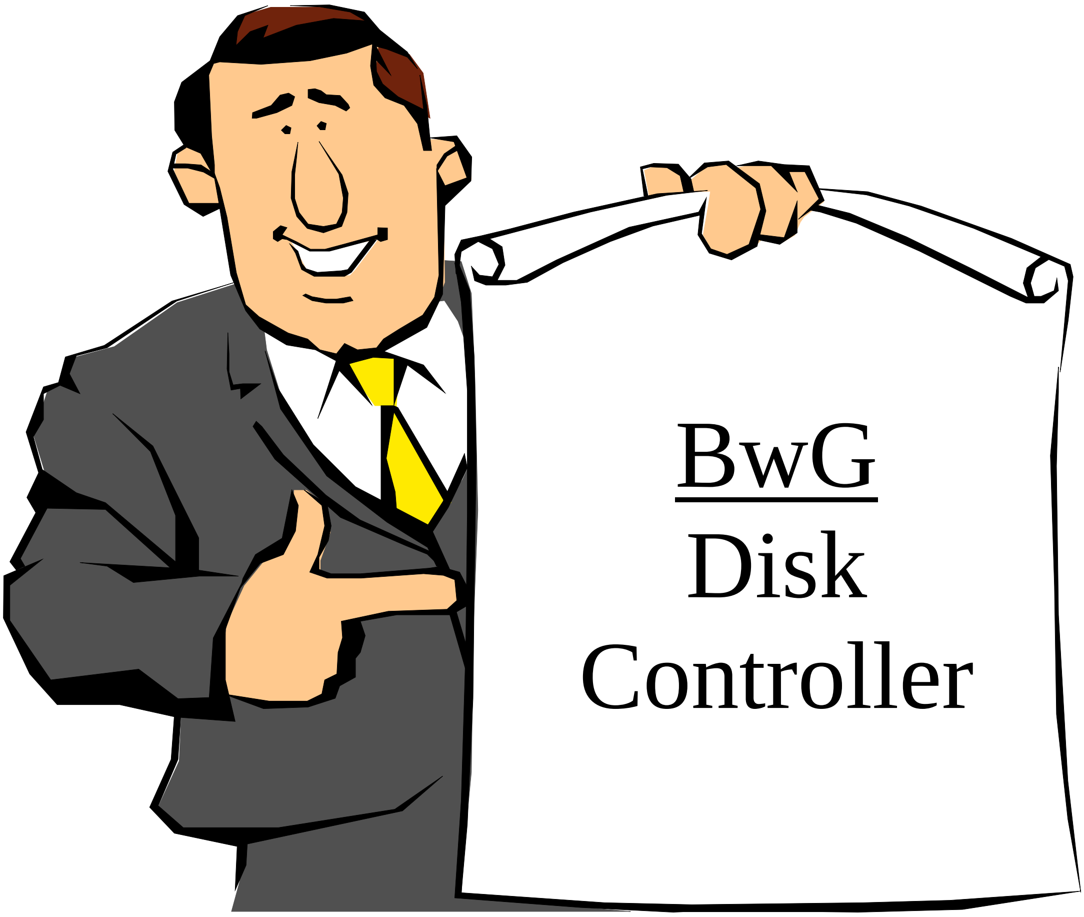

BwG-DSR #4 - v1.18

Description of the DSR

system-99 user-group

Last Manual Edit: 2017-10-17
Translation by Bob Carmany
Actual versions at system-99 user-group

Forward

When we (Michael Becker, Christopher Winter and Harald Glaab) at the end of 1988 decided to produce a new disk controller, we determined to add some new features::

- RAM space to contain diverse flags and buffers
- Sufficient permanent memory (EPROM), that future extensions could be planned
- A real-time clock so that the production and actualization data could be provided.
- DIP-Switches for configuration
- Switching possibility for 4 disk drives
- Drive-Sense line to be able to query the function of the drives

Also for the DSR we provided ourselves with big product requiements:

- Inquiry through the Drive-Sense line to only those disk drives that were connected
- Transfer the Filename-Compare-Buffer into the system RAM so as not to displace the 80-column card in VDP-RAM.
- Head Step Rate set for each disk drive with a DIP-Switch
- Mark the files with a date and time stamp with the production and actualization of the data in the catalog sector that TI intended
- Software-emulation of the CorComp Clock-Card, so that the real time clock could be addresses from Basic
- System subprograms for the support of the clock functions in TMS9900 machine language
- Adjustment of the catalog (OPEN “DSK1.“), so that the date data can be displayed here as well
- Speed increase in cataloging
- Access to the sectors without going through VDP-Ram

and then came all of the others:

After the hardware was completely finished, the DSR not yet, the address logic (with luck in a PAL) was altered so this controller functioned like the Atronic and the DSR was also like the Atronic controllers – with small changes – and could easily be taken over.

We had thought that our Ch. Winter – even with interruptions – told us, that a new DSR for the BWG-Controller, although he did not have the time for it, would not take more than 3 or 4 months but it did not go as planned and has lasted almost 10 years.

New Functions and Features:

Implemented functions existing in the DSR:

- Date and time on disk with disk operations
- Setting of the Step-rate of the disk drives with a DIP-Switch
- Setting the number of drives with a DIP-Switch
- Choice of display of the date and time on the title screen
- Disk catalog in Basic with date and time
- Support of Double-Density-Format of the TI and 16 Sectors per track of the NEC controller chip

Date and Time on the Diskette:

The format of the date and time used on disk by the program MyWord (Geneve) was thoroughly analyzed in May 1987 in Berlin. In the case where this format was modified or completely changed, the BwG controller would store the wrong data on the disk and a false date and time would be displayed.

Restrictions:

Since the date routines are only implemented in the BwG controller, the date and time access to the disk is only written by means of the highlevel-routines in the DSR (DSKx.) or the subroutines >14 and >15.

A date is not added on Ramdisk or ASCSI.

Also we have not found provisions for this in the older Diskmanager I and Diskmanager II modules – but with a sector copy the date and time are also copied.

The Format of the Date and Time:

In the File Descriptor Sector, the sector where the filename resides, is mentioned in publications that Sector-Offset >14 to >1B is a reserved space (Reserved, intended for Date and Time).

These 4 words are specifically for produced time, produced date, actualized time and actualized date.

Byte >14	**Byte >15**	**Byte 16**	**Byte >17**	**Byte >18**	**Byte >19**	**Byte >1A**	**Byte >1B**
Produced Time		Produced Date		Actual Time		Actual Date	

Time:

Highbyte (>14 or >18)								**Lowbyte (>15 or >19)**							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hour					Minute						Second				

There are only 5 bits available for the seconds, so the word for the seconds must be doubled. That means that no odd seconds can be stored (rounded off).

Date:

Highbyte (>16 or >1A)								**Lowbyte (>17 or >1B)**							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Year							Month				Day				

Example Date:

>B2DE = 30. June 1989

Highbyte (>16 or >1A)								**Lowbyte (>17 or >1B)**							
>B				>2				>D				>E			
1	0	1	1	0	0	1	0	1	1	0	1	1	1	1	0
>59 (89)							>06 (06)				>1E (30)				
Year							Month				Day				

Example Time:

>7494 = 14:34:40

Highbyte (>14 or >18)								**Lowbyte (>15 or >19)**							
0	1	1	1	0	1	0	0	0	1	0	1	0	1	0	0
>0E (14)					>22 (34)						>14 (20) times 2 = 40				
Hour					Minute						Second				

Setting the Head Step Rate of the Disk Drive with the DIP-Switch

The Head Step Rate for ALL of the connected disk drives can be set with DIP-Switch 1 between 6ms and 20ms A Head Step Rate of 20ms should only be necessary with older disk drives.

	On (1)	**Off (0)**
Head Step Rate (For all Disk Drives)	20 ms	6 ms

Setting the Number of Disk Drives

The number of disk drives to be connected can be set with DIP-Switches 3 & 4 . That way it is possible to use a disk number of a Ramdisk with a CRU-Address above >1100 to be actually controlled by the disk controller.

Example:

In a disk controller with 2 disk drives (DSK1 & DSK2) connected. The Ramdisk is partitioned into DSK3 and DSK4. The DIP-Switches 3 & 4 are adjusted to 2 disk drives.

Access to DSK1 and DSK2 will reside in the disk controller, access to DSK3 and DSK4 will be handed over to the RAMdisk by the disk controller.

Number of the Disk Drive	**DIP - 3**	**DIP - 4**
1 (DSK 1)	Off (0)	Off(0)
2 (DSK1, DSK2)	On (1)	Off (0)
3 (DSK1, DSK2, DSK3)	Off (0)	On (1)
4 (DSK1, DSK2, DSK3, DSK4)	On (1)	On (1)

So far it is still not possible to pass access with DSK.DISKNAME.FILENAME to the Ramdisk when a disk is not found in a physical disk drive.

Display the Date and Time in the Title Screen

The actual date and time of the real-time clock can be read and displayed on the title screen when DIP-Switch 2 is switched on.

	On (1)	**Off (0)**
Data & Time in the Title Screen	On	Off

Disk Catalog:

The catalog can - if you wish – display the production date and time as well as the actual date and time for each file on the diskette.

If the data record length is not specified with the OPEN command or is not given as 38 bytes, the used disk catalog appears.

If one indicates a data record length of 74 bytes, further information is given.

The data records for each file is covered in the following fields:

Data Type	**Contents**	**Format**
String	Filename	10 Bytes
Variable	Data Type	1...5 (negative = Write protect) 1 = Dis/Fix 2 = Dis/Var 3 = Int/Fix 4 = Int/Var 5 = Program
Variable	Sector Count	0...32767
Variable	Data Length	0...255
String	Produced Time	hh:mm:ss
String	Produced Date	TT.MM.JJ
String	Actual Time	hh:mm:ss
String	Actual Date	TT.MM.JJ

Through changes in the DSR the catalog is developed somewhat faster, sector 1 is read once and placed in the RAM of the BwG.

Example Program

This example program will show the new possibilities of the catalog production (nothing else).

```
100 OPEN #1:“DSK1.“,FIXED 74,INTERNAL,INPUT
110 INPUT #1:A$,A,B,C,X$,X$,X$,X$
120 PRINT A$;B;C
130 PRINT
140 INPUT #1:A$,A,B,C,E_UHRZEIT$,E_DATUM$,A_UHRZEIT$,A_DATUM$
150 IF A$=““ THEN 200
160 PRINT A$;TAB(11);A;TAB(15);B;TAB(20);C
170 PRINT E_UHRZEIT$,E_DATUM$,A_UHRZEIT$,A_DATUM$
180 PRINT „--------------------------„
190 GOTO 140
200 CLOSE #1
```

DSR “CLOCK“

The command format is identical to that of the CorComp Clock Card and the Triple-Tech-Card, so that existing programs will run without alteration.

Unfortunately the comparibility does not extend to the Horizon menu program that addresses the time with “CLOCK“ which only determines if the clock has been implemented in the system. It is not polled through direct access to the clock registers as in the CorComp clock and as has been done with the BwG controller. The patched menu program is supplied with the BwG controller in order that this type of special polling will display the time.

Example Program

```
100 OPEN #1:“CLOCK“
110 INPUT #1:A$,B$,C$
120 CLOSE #1
130 PRINT A$;B$;C$
```

Output for Wednesday, 9. November 1998, 13:12:24 :

```
2         Weekday (Monday = 0, Sunday = 6)
11/09/98  Date in American notation (Month/Day/Year)
13:12:24  Time in 24-Hour-Display
```

If the open file “CLOCK“ is written to, the real-time-clock can be placed in the same format.

```
100 A$ = “2“
110 B$ = “11/09/98“
120 C$ = “13:12:24“
130 OPEN #1:“CLOCK“
140 PRINT A$,B$,C$
150 CLOSE #1
```

Subprogram >1E

This subroutine enables access to the individual sectors on the diskette without going through the VDP-RAM, but writes the data directly to the CPU-RAM from the sector in reverse order..

Delivered Parameters:

>834C	(b)	Disk Drive Number	
>834D	(b)	I/O-Code	>FF = Read
			>00 = Write
>834E	(w)	Pointer to the Buffer in CPU-RAM	
>8350	(w)	Sector number	
>8356	(w)	Pointer to the name length byte in PAB	

Returned Parameter:

>8350 (b)	Returned Error code

Example program

```
       DEF    READ,WRITE
       REF   DSRLNK,VMBW
*
PABAD  EQU   >1000     Addess of the PABs in VDP-RAM
BUFFER EQU   >C000     Address of the Data buffers in CPU-RAM
*
LW     BYTE  >01             Disk drive 1
       EVEN
*
NAME   DATA  >011E           Subroutine name
SEKTOR DATA  150             Sector number
MYWS   BSS   32              Individual Workspace
*
READ   LWPI  MYWS            Load Individual Register
       SETO  R0              R0 = >FFFF, read indicator
RW     MOVB  R0,@>834D       I/O-Code
       MOVB  @LW,@>834C      Disk drive number
       MOV   @SEKTOR,@>8350  Sector number
       LI    R0,BUFFER
       MOV   R0,@>834E       Address of the buffers given
       LI    R0,PABAD
       MOV   R0,@>8356       Pointer to the Namelengthbyte
       LI    R1,NAME         Position of the SBR-Names
       LI    R2,2            2 Byte shift
       BLWP  @VMBW           Create PAB in VDP-RAM
       BLWP  @DSRLNK         Call Routine
       DATA  >A              Routine index
       MOVB  @>8350,@>8350   Check error byte
       JEQ   OK              No error
ERROR  ****  ****            Error handling here
OK     ****  ****            Resume program
*
WRITE  LWPI  MYWS
       CLR   R0              Write indicator
       JMP   RW              Repeat all of the previous actions!
*
END                          End of the Example program
```

Subprogram >1F

The subroutine >1F is what makes access to the date and time in the real-time-clock of the BwG controller possible with machine language.

Here the control of subroutines >10 to >16 is accomplished with DSRLNK.

Delivered Parameters

>834D (b)	I/O-Format	>00	Read	WJJMMTThhmmss
		>01	Read	TT.MM.JJJJ - hh:mm:ss
		>02	Read	hh:mm:ss
		>03	Read	TT.MM.JJJJ
		>04	Read	W (1=Mon, 7=Sun)
		>05	Read	MM-TT-JJJJ – hh:mm:ss
		>06	Read	MM-TT-JJJJ
		>80	Read	Variables Output format (see below)
		>FF	Write	WJJMMTThhmmss
>834E (w)	Pointer to the Buffer in VDP-RAM			
>8356 (b)	Pointer to the Namelengthbyte in PAB			

Returned Parameter

>8350 (b)	Returned Error code

Example Program

```
       REF   DSRLNK,VMBW
       DEF   START
*
PABAD  EQU   >1000           Address of the PABs in VDP-RAM
BUFFER EQU   >0060           Address of the Data buffers (on the screen)
*
NAME   DATA  >011F           Subroutine name
MYWS   BSS   32              Individual Workspace
*
START  LI    R0,>0100        Data format 1 (TT.MM.JJJJ - hh:mm:ss)
       MOVB  R0,@>834D       I/O-Code
       LI    R0,BUFFER
       MOV   R0,@>834E       Address of the buffers
       LI    R0,PABAD
       MOV   R0,@>8356       Pointer to the Namelength byte
       LI    R1,NAME         Position of the SBR-Names
       LI    R2,2            2 Byte shift
       BLWP  @VMBW           Create PAB in VDP-RAM
       BLWP  @DSRLNK         Call Routine
       DATA  >A              Routine index
       MOVB  @>8350,@>8350   Check error byte
       JEQ   OK              No error
ERROR  ****  ****            Error handling here
OK     ****  ****            Resume program
       RT
*
       END
```

Subprogram >1F - Special ouput format >80 (Variable output format)

In this mode it is possible to put all of the date and time information in an arbitrary order and format the output. It is necessary to store an example string in the CPU storage area in the format that you wish to use and it must be ended with a backslash (\).

Delivered Parameters

>834D (b)	I/O-Format	>80	Read	Variable Output format (see below)
>834E (w)	Pointer to the Buffer in VDP-RAM			
>8350 (w)	Pointer to the Format string in RAM			
>8356 (b)	Pointer to the Namelengthbyte in PAB			

Returned Parameter

>834D (b)	Returned Output String Length
>8350 (b)	Returned Error code

Format of the Elements

Format of element	Output	Place
T, D	Day	2
M	Month	2
J, Y	Year	4
j, y	Year	2
h	Hour	2
m	Minute	2
s	Second	2
z	Tenth of Second	1
W	Weekday German	variable
w	Weekday German	2
X	Weekday English	variable
x	Weekday English	2
V	Weekday German uppercase only	2
v	Weekday English uppercase only	2
C	Month name German	variable
c	Month name German	2
E	Month name English	variable
E	Month name English	2
. , -/ : ´ Empty	Pause characters	1
\	String end, **absolutely necessary**	0

Example

Format string	Output
W, T.C J – h:m:s\	Donnerstag, 28.September 2000 – 18:23:40
w M-T-J\	Thursday 09-28-2000
W\	Donnerstag
h:m\	18:23

SBR Calls

The DSR supports the following CALLs in TI.Basic or the HRD-Menu with “K“:

Call Files (x)

With this CALL you can store the number of file buffers in VDP-Ram. For each open file there must be the same number of data buffers or you will get an error message. Possible values for (x) are from 1 to 9.

Call 16SPT

This command switches the controller software to a disk format with 16 sectors per track in Double-Density, for those disk controllers that support it, that was planned for the predecessor of the TI Disk Controller PHP 1240 that never made it out of the beta stage and was never produced or sold..
However, this is the format of the HEXBUS-Disk Controller and the one that MYARC products used so disks in circulation in this format can be run (and are).
The command of 16SPT goes to all of the disk drives that are connected to the BwG Controller. Access to “normal“ Double-Density format is no longer possible and can only be done through a Ramdisk or a diskette with Single-Density..
The 16-Sector format remains in effect until switched back with a CALL 18SPT.
This change affects all programs that use the DSR for reading and writing.. Also formatting for most programs should function since these all use the individual DSR routines.

Call 18SPT

Switches back to the “normal“ Double-Density format (18 Sectors per track)

Technical Appendix

CRU-Mapping Output Bits:

CRU-Adr	CRU-Bit	MM30	Name	Meaning
>1100	>0	>0001	DEN	DSR-Enable
>1102	>1	>0002	MOP	Motor-On-Monoflop-Trigger
>1104	>2	>0004	WAITENA	Activation of the READY Logic
>1106	>3	>0008	-	Currently unused
>1108	>4	>0010	DS1	Drive Select for DSK1
>110A	>5	>0020	DS2	Drive Select for DSK2
>110C	>6	>0040	DS3	Drive Select for DSK3
>110E	>7	>0080	SIDE	>0 for Side 1, >1 for Side 2
>1110	>8	>0100	DS4	Drive Select for DSK4
>1112	>9	>0200	-	Currently unused
>1114	>A	>0400	/DDEN	Double Density Enable
>1116	>B	>0800	A13P	A13 from EPROM
>1118	>C	>1000	-	Currently unused
>111A	>D	>2000	A10P	A10 from RAM
>111C	>E	>4000	/FR	RTC over FDC
>111E	>F	>8000	A14P	A14 from EPROM

CRU-Mapping Input Bits:

CRU-Adr	CRU-Bit	Name	Meaning
>1100	>0	/D4C	0 = DSK4 connected
>1102	>1	/D1C	0 = DSK1 connected
>1104	>2	/D2C	0 = DSK2 connected
>1106	>3	/D3C	0 = DSK3 connected
>1108	>4	DIP 1	DIP-Switch 1
>110A	>5	DIP 2	DIP-Switch 2
>110C	>6	DIP 3	DIP-Switch 3
>110E	>7	DIP 4	DIP-Switch 4

Storage Space:

	Bank 0	Bank 1	Bank 2	Bank 3
EPROM 27C256	>4000 - >5BFF	>4000 - >5BFF	>4000 - >5BFF	>4000 - >5BFF
RAM 6116	>5C00 - >5FDF			
Clock MM58274	>5FE0 - >5FFF			
FDC WD 1773	>5FF0 - >5FFF			

Access Addresses of the Real Time Clock (RTC):

Address	DIR	Meaning
>5FE0	R/W	Control Register
>5FE2	R	Tenth of Second
>5FE4	R/W	One Second
>5FE6	R/W	Ten Seconds
>5FE8	R/W	One Minute
>5FEA	R/W	Ten Minutes
>5FEC	R/W	One Hour
>5FEE	R/W	Ten Hours
>5FF0	R/W	One Day
>5FF2	R/W	Ten Days
>5FF4	R/W	One Month
>5FF6	R/W	Ten Months
>5FF8	R/W	One Year
>5FFA	R/W	Ten Years
>5FFC	R/W	Weekday
>5FFE	R/W	Clock Set & Interrupt Control Register

Access Addresses of the Floppy Disk Controller (FDC):

Address	Direction	Meaning
>5FF0	R	Status Register
>5FF2	R	Track Register
>5FF4	R	Sector Register
>5FF6	R	Data Register
>5FF8	W	Command Register
>5FFA	W	Track Register
>5FFC	W	Sector Register
>5FFE	W	Data Register

Banking:

Bank	CRU-Bit A13P	CRU-Bit A14P
Bank 0	0	0
Bank 1	1	0
Bank 2	0	1
Bank 3	1	1

DIP-Switch 1 (Stepping Rate):

	On (1)	Off (0)
Stepping Rate (For all Disk Drives)	20 ms	6 ms

DIP-Switch 2 (Date and Time in the Title Screen):

	On (1)	Off (0)
Date & Time in the Title Screen	On	Off

DIP-Switch 3&4 (Number of the Disk drives):

Number of the Disk Drive	DIP - 3	DIP - 4
1 (DSK 1)	Off (0)	Off (0)
2 (DSK1, DSK2)	On (1)	Off (0)
3 (DSK1, DSK2, DSK3)	Off (0)	On (1)
4 (DSK1, DSK2, DSK3, DSK4)	On (1)	On (1)

Construction of the New DSR

Changing the DSR requires the following:

- The GAL20V8 (U11) must be exchanged with the address logic ATDC3 as opposed to a new GAL 22V10 with address logic BwG1 (Already done in DSR Version 1.14)
- The link on J2a changed to J2b (Also already done in DSR Version 1.14)
- Exchange the old (U15) ATUx for the new EPROM BwG4
- If necessary adjust the DIP Switches.

DSR-Version display

The actual current version of DSR #4 is

```
BwG-1 ©H.Glaab
25.09.2000 v1.18
```

The version number and version date are stored in every bank in the EPROM at the address >4010 to >402F and can be found with the following small program:

```
*
* Version of the BwG-DSR read and display
*
       DEF START
       REF VMBW,KSCAN
*
START  LI   R12,>1100        CRU-Base of the Disk Controller
*
       LI   R0,>0048         Screen address
       LI   R2,>0010         Text length
       LI   R3,CRUDAT        CRU-Address of the individual Banks
       LI   R4,4             4 Banks
*
LOOP   LDCR *R3+,0           Set the CRU
       LI   R1,>4010         Bank and Copyright
       BLWP @VMBW            Display
       AI   R0,>0020         Screen position
       LI   R1,>4020         Version Date and Number
       BLWP @VMBW            Display
       AI   R0,>0040         Screen position
       DEC  R4               Loop counter
       JNE  LOOP             Loop
*
       LDCR *R3+,0           Turn off CRU
       CLR  R12              R12 report

LOOP2  CLR  @>8374           KeyMode 0
       BLWP @KSCAN           Keypress detect
       CB   @>8375,@HFF      Keycode >FF?
       JEQ  LOOP2            Yes, wait
       CB   @>837C,@EQBIT    Set EQ-Bit?
       JNE  LOOP2            No, wait
       RT                    Return to the E/A-Module
*
CRUDAT DATA >0001  BANK 1
       DATA >0801  BANK 2
       DATA >8001  BANK 3
       DATA >8801  BANK 4
       DATA >0000  AUS

HFF    BYTE >FF    No keypres
EQBIT  BYTE >20    EQ-Bit

*
       END
```

Direct Access of the Date and Time on Diskette

For programmers in TMS9900 machine language here is a small extension to the catalog program SD from Clint Pulley for additional display and actualization of the date and time of the files. It is assumed that the each catalog sector is already in the buffer @DBUF.

```
          ...
       MOV  @DBUF+26,R6      Date of actualization
       JEQ  SD9A             No entry nothing shown
       BL   @SHWDTM
       MOV  R8,R0            Fetch Screen position
       AI   R0,20            Next row
       LI   R1,DATUM         Pointer to buffer
       LI   R2,17            Count of the bytes in buffer
       BLWP @VMBW            Display the data
SD9A      ...

SHWDTM MOV  R11,@SHWDTX+2    Set the return address
       LI   R0,DATUM         Here is where the data is
       MOV  R6,R3            Working copy
       ANDI R3,>1F           Day
       BL   @D2DZ            Change to ASCII
       INC  R0               Skip the period (.)
       MOV  R6,R3
       SRL  R3,5             Month
       ANDI R3,>F
       BL   @D2DZ
       INC  R0
       MOV  R6,R3
       SRL  R3,9             Year
       BL   @D2DZ
       MOV  @DBUF+24,R6      Read the actualized time
       INC  R0               Skip the blank
       MOV  R6,R3
       SRL  R3,11            Hour
       BL   @D2DZ
       INC  R0
       MOV  R6,R3
       SRL  R3,5             Minute
       ANDI R3,>3F
       BL   @D2DZ
       INC  R0
       MOV  R6,R3
       ANDI R3,>1F           Seconds /2
       SLA  R3,1
       BL   @D2DZ
SHWDTX B    @0
*
* Subprogram, switch values to ASCII
*
D2DZ   CLR  R2
       DIV  @DNDC,R2         Value divided by 10
       AI   R2,48            ASCII
       SWPB R2
       MOVB R2,*R0+          Ten place
       AI   R3,48
       SWPB R3
       MOVB R3,*R0+          Ones place
       RT
*
DATUM  TEXT ,00.00.00 00:00:00‘  Display buffer
DNDC   DATA 10                   Divisor
```