


```
0001 SUBROUTINE RXDISK (LUN)
0002 C
0003 C Version: 'V04-000'
0004 C
0005 C*****
0006 C*
0007 C* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0008 C* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0009 C* ALL RIGHTS RESERVED.
0010 C*
0011 C* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0012 C* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0013 C* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0014 C* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0015 C* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0016 C* TRANSFERRED.
0017 C*
0018 C* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0019 C* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0020 C* CORPORATION.
0021 C*
0022 C* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0023 C* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0024 C*
0025 C*
0026 C*****
0027 C
0028 C
0029 C
0030 C AUTHOR BRIAN PORTER CREATION DATE 9-APR-1979
0031 C
0032 C++
0033 C Functional description:
0034 C
0035 C This module displays entries for the RX211 controller. The format
0036 C of the error log buffer is as follows.
0037 C
0038 C +-----+
0039 C | control and status register |
0040 C +-----+
0041 C | device data buffer register |
0042 C +-----+
0043 C | data path number |
0044 C +-----+
0045 C | data path register |
0046 C +-----+
0047 C | final map register |
0048 C +-----+
0049 C | previous map register |
0050 C +-----+
0051 C | special error register |
0052 C +-----+
0053 C |
0054 C |-- extended error information --|
0055 C |
0056 C +-----+
0057 C
```

RXD
AP
ARR
2
LAB
FUN
T
COM
F
/
/
/
/


```

0273 integer*4      soft_status
0274 integer*4      bad_track
0275
0276 logical*1      done
0277
0278 CHARACTER*20    RX_HEADSELCT(0:1)
0279 CHARACTER*7     RX_DENSITY(0:1)
0280
0281 CHARACTER*5     DRIVE_TYPE(0:1)
0282 DATA  DRIVE_TYPE(0)  /'RX01*'/
0283 DATA  DRIVE_TYPE(1)  /'RX02*'/
0284
0285 CHARACTER*17    V1RX2_CS(5:7)
0286 DATA  V1RX2_CS(5)    /'DONE*'/
0287 DATA  V1RX2_CS(6)    /'INTERRUPT ENABLE*'/
0288 DATA  V1RX2_CS(7)    /'TRANSFER REQUEST*'/
0289
0290 character*20    v2rx2_cs(8:9,0:1)
0291 data  v2rx2_cs(8,0)   /'SINGLE DENSITY*'/
0292 data  v2rx2_cs(8,1)   /'DOUBLE DENSITY*'/
0293 data  v2rx2_cs(9,0)   /'LOWER HEAD SELECTED*'/
0294 data  v2rx2_cs(9,1)   /'UPPER HEAD SELECTED*'/
0295
0296 CHARACTER*6     V3RX2_CS(15:15)
0297 data  v3rx2_cs(15)    /'ERROR*'/
0298
0299 CHARACTER*20    V1RX2_ES(0:4)
0300 DATA  V1RX2_ES(0)    /'CRC ERROR*'/
0301 DATA  V1RX2_ES(1)    /'SIDE 1 READY (RX03)*'/
0302 DATA  V1RX2_ES(2)    /'INITIALIZE DONE*'/
0303 DATA  V1RX2_ES(3)    /'DRIVE AC LO*'/
0304 DATA  V1RX2_ES(4)    /'DENSITY ERROR*'/
0305
0306 character*21    v2rx2_es(5:5,0:1)
0307 data  v2rx2_es(5,0)   /'SINGLE DENSITY DRIVE*'/
0308 data  v2rx2_es(5,1)   /'DOUBLE DENSITY DRIVE*'/
0309
0310 CHARACTER*13    V3RX2_ES(6:7)
0311 data  v3rx2_es(6)     /'DELETED DATA*'/
0312 data  v3rx2_es(7)     /'DRIVE READY*'/
0313
0314 CHARACTER*20    V4RX2_ES(10:11)
0315 data  v4rx2_es(10)    /'WORD COUNT OVERFLOW*'/
0316 data  v4rx2_es(11)    /'NON-EXISTENT MEMORY*'/
0317
0318 character*27    v1soft_status(0:0,0:1)
0319 data  v1soft_status(0,0) /'COMMAND WAS SINGLE DENSITY*'/
0320 data  v1soft_status(0,1) /'COMMAND WAS DOUBLE DENSITY*'/
0321
0322 character*26    v2soft_status(4:4,0:1)
0323 data  v2soft_status(4,0) /'DRIVE #0., SINGLE DENSITY*'/
0324 data  v2soft_status(4,1) /'DRIVE #0., DOUBLE DENSITY*'/
0325
0326 character*12    v3soft_status(5:5)
0327 data  v3soft_status(5) /'HEAD LOADED*'/
0328
0329 character*26    v4soft_status(6:7,0:1)

```



```

0558      call linchk (lun,1)
0559
0560      write(lun,140) current_track_address_drive0
0561      format(' ',t40,'CURRENT TRACK #',
140      1 i<compress4 (current_track_address_drive0)>,'.., DRIVE #0.')
0562
0563      current_track_address_drive1 = lib$extzv(24,8,extended_registers(1))
0564
0565      call linchk (lun,1)
0566
0567      write(lun,145) current_track_address_drive1
0568      format(' ',t40,'CURRENT TRACK #',
145      1 i<compress4 (current_track_address_drive1)>,'.., DRIVE #1.')
0569
0570
0571      target_track = lib$extzv(0,8,extended_registers(2))
0572
0573      call linchk (lun,1)
0574
0575      write (lun,150) target_track
0576      format(' ',t40,'TRACK #',i<compress4 (target_track)>,
150      1 '.., TARGET TRACK')
0577
0578
0579      target_sector = lib$extzv(8,8,extended_registers(2))
0580
0581      call linchk (lun,1)
0582
0583      write(lun,155) target_sector
0584      format(' ',t40,'SECTOR #',i<compress4 (target_sector)>,
155      1 '.., TARGET SECTOR')
0585
0586
0587      soft_status = lib$extzv(16,8,extended_registers(2))
0588
0589      call output (lun,soft_status,v1soft_status,0,0,0,'2')
0590
0591      call output (lun,soft_status,v2soft_status,4,4,4,'2')
0592
0593      call output (lun,soft_status,v3soft_status,5,5,5,'0')
0594
0595      call output (lun,soft_status,v4soft_status,6,6,7,'2')
0596
0597      if (definitive_error_code .eq. '150'o) then
0598
0599      bad_track = lib$extzv(24,8,extended_registers(2))
0600
0601      call linchk (lun,1)
0602
0603      write(lun,160) bad_track
0604      format(' ',t40,'SELECTED DRIVE AT TRACK #',
160      1 i<compress4 (bad_track)>,'..')
0605
0606      endif
0607
0608      if (emb$w_hd_entry .ne. 98) then
0609
0610      call uba_datapath (lun,uba_regs(1),uba_regs(2))
0611
0612      call uba_mapping (lun,-1,uba_regs(3))
0613
0614

```



```

0321      qiocode(0,i) = 33
0322
0323      if (qiocode(1,i) .eq. 0) then
0324
0325      qiocode(1,i) = %loc(qio_string)
0326      endif
0327
0328      10      continue
0329      endif
0330
0331      call irp$w_func (lun,emb$w_dv_func,
0332      1 qiocode(0,lib$extzv(0,6,emb$w_dv_func)))
0333
0334      return
0335
0336      end
    
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	225	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	8	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	548	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 \$QIOCOMMON	1247	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated		2028

ENTRY POINTS

Address	Type	Name
0-00000000		RXDISK_QIO

VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008	I*2	EMB\$W_DV_FUNC	2-00000200	I*4	I
3-00000442	CHAR	IOS_ABORT	3-0000034D	CHAR	IOS_ACCESS
3-000003C2	CHAR	IOS_ACPCONTROL	3-000004B3	CHAR	IOS_AVAILABLE
3-00000297	CHAR	IOS_CLEAN	3-00000369	CHAR	IOS_CREATE
3-00000385	CHAR	IOS_DEACCESS	3-00000393	CHAR	IOS_DELETE
3-0000026D	CHAR	IOS_DIAGNOSE	3-00000065	CHAR	IOS_DRVCLR
3-000004CB	CHAR	IOS_DSE	3-000000A9	CHAR	IOS_ERASETAPE
3-00000276	CHAR	IOS_FORMAT	3-00000071	CHAR	IOS_INITIALIZE
3-00000014	CHAR	IOS_LOADMCODE	3-000003A1	CHAR	IOS_MODIFY
3-000003E2	CHAR	IOS_MOUNT	3-00000000	CHAR	IOS_NOP
3-0000009D	CHAR	IOS_OFFSET	3-000000EB	CHAR	IOS_PACKACK
3-000000E0	CHAR	IOS_OSTOP	3-000003EF	CHAR	IOS_RDSTATS
3-00000421	CHAR	IOS_READCSR	3-00000169	CHAR	IOS_READHEAD
3-000002B6	CHAR	IOS_READLBLK	3-0000013F	CHAR	IOS_READPBLK

3-00000200	CHAR	IOS_READPRESET	3-00000195	CHAR	IOS_READTRACKD
3-0000033A	CHAR	IOS_READVBLK	3-0000045A	CHAR	IOS_READWTHBUF
3-00000484	CHAR	IOS_READWTHXBUF	3-0000004D	CHAR	IOS_RECAL
3-0000007C	CHAR	IOS_RELEASE	3-000001AB	CHAR	IOS_REREADN
3-000001B8	CHAR	IOS_REREADP	3-000000CA	CHAR	IOS_RETCENTER
3-000002E6	CHAR	IOS_REWIND	3-000002C9	CHAR	IOS_REWINDOFF
3-000000FC	CHAR	IOS_SEARCH	3-00000024	CHAR	IOS_SEEK
3-00000231	CHAR	IOS_SENSECHAR	3-00000309	CHAR	IOS_SENSEMODC
3-0000021D	CHAR	IOS_SETCHAR	3-000003B8	CHAR	IOS_SETCLOCK
3-00000088	CHAR	IOS_SETCLOCKP	3-000002DD	CHAR	IOS_SETMODE
3-000002ED	CHAR	IOS_SKIPFILE	3-000002FA	CHAR	IOS_SKIPRECORD
3-00000029	CHAR	IOS_SPACEFILE	3-0000010E	CHAR	IOS_SPACERECORD
3-000003D7	CHAR	IOS_STARTDATA	3-000000B4	CHAR	IOS_STARTDATAP
3-00000037	CHAR	IOS_STARTMPROC	3-0000020F	CHAR	IOS_STARTSPNDL
3-00000059	CHAR	IOS_STOP	3-0000000D	CHAR	IOS_UNLOAD
3-0000046B	CHAR	IOS_WRITEBUFNCRC	3-0000011E	CHAR	IOS_WRITECHECK
3-000001E4	CHAR	IOS_WRITECHECKH	3-000003FF	CHAR	IOS_WRITECSR
3-00000153	CHAR	IOS_WRITEHEAD	3-000002A2	CHAR	IOS_WRITELBLK
3-00000247	CHAR	IOS_WRITEMARK	3-00000314	CHAR	IOS_WRITEOF
3-0000012A	CHAR	IOS_WRITEPBLK	3-000001C9	CHAR	IOS_WRITERET
3-0000017E	CHAR	IOS_WRITETRACKD	3-00000326	CHAR	IOS_WRITEVBLK
3-00000448	CHAR	IOS_WRITEWTHBUF	3-00000257	CHAR	IOS_WRTTMKR
AP-00000004@	L*1	LUN	3-000004A1	CHAR	QIO_STRING

ARRAYS

Address	Type	Name	Bytes	Dimensions
2-00000000	I*4	QIOCODE	512	(0:1, 0:63)

LABELS

Address	Label
**	10

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name
	IRPSW_FUNC	I*4	LIB\$EXTZV

COMMAND QUALIFIERS

FORTRAN /LIS=LIS\$:RXDISK/OBJ=OBJ\$:RXDISK MSRC\$:RXDISK

/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
 /DEBUG=(NOSYMBOLS,TRACEBACK)
 /STANDARD=(NOSYNTAX,NOSOURCE FORM)
 /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)
 /F77 /NOG_FLOATING /I4 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19

The image displays a grid of 128 small terminal window screenshots, arranged in 8 rows and 16 columns. Each window shows a different set of system commands and their corresponding outputs on a VAX/VMS system. The outputs vary significantly, including:

- System status and configuration information.
- File listings and directory structures.
- Execution of system utilities such as `RESELECT LIS`, `PUDRIVER LIS`, `PCL11T LIS`, `PCL11R LIS`, `ROLLUP LIS`, `RDISK LIS`, `RXDISK LIS`, and `SB11 LIS`.
- Log files and diagnostic reports.
- System error messages and warnings.
- Network-related information.

The screenshots are small and densely packed, providing a comprehensive overview of the system's operational state and the variety of tasks performed.