```
wjm 18-apr-1999: Almost finalized PK2K kits (Vs3100-like SCSI support
                 for VAXstation 2000 & microVAX 2000) for various flavors
                 of OpenVMS VAX. Packaging to be improved at some later date.

   ... Subject to change without notice ... partially untested ...

        PK2K_0013-BIN.ZIP       binaries for V5.5-2, plus the README
                                (slight update from the 0012 version).

        PK2K_0013-061BIN.ZIP    binaries for V6.1 (NEW!)

        PK2K_0013-062BIN.ZIP    binaries for V6.2 (NEW!)

        PK2K_0013-071BIN.ZIP    binaries for V7.1 (NEW!)

The README in PK2K_0013-BIN.ZIP applies to all of the above,
with the respective VMS version substituted for "V5.5-2".

        PK2K-BOOT_0013.ZIP      "secondary" SYSBOOT images for booting into
                                a SCSI disk, to be loaded via DUAn: or ESA0:

        KA410W_V23_ROM-0013.PATCH
                                (ASCII!) PATCH command file for improving
                                upon the "KA410-B V2.3" ROM, allowing it
                                to boot from SCSI disks, instead of "MUA0"

   ... Subject to change without notice ... V6.2 and V7.1 not yet tested ...
   ... Not "supported" by anyone ... but very likely to just work ...

w.j.moeller, <moeller@gwdg.de>
```

```
KA410ROM23.BIN /abs /out=KA410ROM23W.BIN /jou=KA410ROM23W

!-------------------------------------------------------------------------------
! "KA410-B V2.3" ROM:
!       - replace "MUA0" TVBTDRIVER by "DKAn" DK2KBTDRIVER
!       - fix bogus 000000C4 SCSI (disk) self-test error
!       - add "T 56" for checking/setting the SCSI host id (default = 0)
!       - try to improve SCSI arbitration logic during self-test (03C6)
!               (no errors observed yet).
!
!-------------------------------------------------------------------------------
! w.j.m. 24-mar-1999 (based upon 2KBTDRIVER.MAR with CDROM = 1)
! mod 29-mar-1999 wjm: (after Vs3100 "STRG V1.3")
!                       fix "000000C4" SCSI self-test problem,
!                       caused by not expecting interrupt due to ATN;
!                       also remove "000000C6" test.
! mod 02-apr-1999 wjm: 2KBTDRIVER.MAR updated to support 'IO410$AB_SCSI';
!                       change rom-id from "KA410-B W" to KA410-W V"
! mod 04-apr-1999 wjm: add "T 56"; try to fix arbitration logic (03C6 error).
!-------------------------------------------------------------------------------
!
def IO410$AL_SIDEX      = 20040004
def IO410$AB_CPMBX      = 200B0038      ! NVRAM, HALT flags
def IO410$AB_SCSI       = 200B00BC      ! NVRAM, SCSI host id (name made up)
def IO410$AB_SCTLS      = 200C0080
!
def ka410rom            = IO410$AL_SIDEX - 4    ! more physical addresses ...
def ka410rom_end        = ka410rom + 040000
def ka410rom_cksum      = ka410rom_end - 4
!
!-------------------------------------------------------------------------------
!
! NOTE: As of 02-apr-1999, this patch affects both the ROM VMB,
!       and the self-test code, so both ids are changed.
!       Use ">>> T 80000050" in order to inspect the VMB version.
!
def vmb_version         = 2005376C - ka410rom
!
replace/ascii/long vmb_version
'V1.2'
exit
'W1.2'
exit
!
def rom_version_410     = 200414B0 - ka410rom
!
replace/ascii/long rom_version_410
'KA41'
'0-B '
'V2.x'
exit
'KA41'
'0-W '          ! ... here
'V2.x'
exit
!
!-------------------------------------------------------------------------------
!-----  part 1a: fix SCSI self-test (000000C4)
!
def t6_unit_c4_loop     = 20044D4F - ka410rom
def t6_unit_c6_start    = 20044D82 - ka410rom
def t6_unit_c6_end      = 20044D92 - ka410rom
def t6_unit_finish      = 20044E1F - ka410rom   ! exit tests here
def delay_316           = 2004C7AF - ka410rom   ! just a random subroutine
!
verify/inst     t6_unit_c4_loop         ! wait for interrupt after DMA MSG_IN
```

```
'       MOVL    B^014(R7),R2'           ! get mask set by interrupt handler
'       BITL    #00000050,R2'           ! expect <6> and/or <4>
'       BNEQ    _4D7A'                  ! if so, done with "C4" test
'       BBCCI   #0,B^014(R7),_4D64'     ! if some other interrupt occurred,
'       TSTB    B^01C(R6)'              ! touch _RESET to acknowledge it
'_4D64: JSB     L^delay_316'
'       AOBLEQ  #000007D0,R1,t6_unit_c4_loop'
'       MOVB    #0C4,B^8(R7)'           ! set ttssmmC4 status
'       BRW     t6_unit_finish'
'_4D7A: BBCCI   #0,B^014(R7),_4D82'     ! if an interrupt occurred
'       TSTB    B^01C(R6)'              ! touch _RESET to acknowledge it
'_4D82: MOVL    B^014(R7),R2'           ! get mask again
'       BBS     #4,R2,t6_unit_c6_end'   ! if <4>(inteop), done with "C6" test
'       MOVB    #0C6,B^8(R7)'           ! set ttssmmC6 status
'       BRW     t6_unit_finish'
exit
!
replace/inst    t6_unit_c4_loop + 4
'       BITL    #00000050,R2'           ! expect <4>(inteop) and <6>(phase chg)
exit
'       BITL    #000000D0,R2'           ! also expect <7>(ATN)
exit
!
replace/inst    t6_unit_c6_start        ! <4> is typically missing ...
'       MOVL    B^014(R7),R2'
'       BBS     #4,R2,t6_unit_c6_end'
'       MOVB    #0C6,B^8(R7)'
'       BRW     t6_unit_finish'
exit
'       nop'                            ! ... so remove this test (cf. Vs3100)
exit
!
!-------------------------------------------------------------------------------
!-----  part 1b: try to fix SCSI self-test arbitration (03C6)
!
def t6_select_3c6       = 20045293 - ka410rom
def t6_select_3c6_end   = 200452A3 - ka410rom
def t6_select_exit      = 2004531D - ka410rom
!
verify/inst     t6_select_3c6
'       movb    (r6),r2'                ! get _CUR_DATA
'       cmpb    r4,r2'                  ! compare 1@hostid to bus data
'       bleq    t6_select_3c6_end'      !* signed test here is definitely wrong
'       movzwl  #03C6,r0'               ! ... return(03C6)
'       brw     t6_select_exit'         ! (can't retry forever w/i self-test)
exit
!
replace/inst    <t6_select_3c6 + 6>
'       bleq    t6_select_3c6_end'      ! fail with (signed) lower id contender,
exit                                    !       7<0<1<2<3<4<5<6
'       bgequ   t6_select_3c6_end'      ! fail with *higher* id contender,
exit                                    !       0<1<2<3<4<5<6<7
!
!-------------------------------------------------------------------------------
!-----  part 2: add "T 56" utility for setting SCSI host id
!
! unused space in KA410-B V2.3 ROM:
!       vmb410_end ... PRA0_bootblock   approx. 320 bytes
!       space after "B PRA0" image      approx. 37 kBytes
!
def PRA0_bootblock      = 20059C00 - ka410rom
verify/long PRA0_bootblock                      ! plausibility check only
+0E7000018
exit
!
examine <pra0_bootblock + 8>                    ! # blocks loaded by "B PRA0"
def pra0_boot_blocks = \
```

```
def free_area = pra0_bootblock + 0200 + <0200 * pra0_boot_blocks>
def free_size = <ka410rom_cksum - ka410rom> - free_area ! size of "patch area"
!
!---
!
! a few random subroutines ...
def cons_output_r0r1    = 20047764 - ka410rom   ! r0 = length, r1 = address
def cons_output7b_r2    = 2004777D - ka410rom   ! output any 7 bit byte in R2
def cons_input_num      = 2004132E - ka410rom   ! [(r0=0) => input a number]
!
def phy_test_50_54      = 20048EC0              ! T 5% get here
def test_50_54          = phy_test_50_54 - ka410rom
def test_53             = 200490FD - ka410rom   ! T 53: prompt for HALT flags
!
def selftest_array      = 20045739 - 8 - ka410rom
def selftest_array_NVR  = selftest_array + <2 * 080>
verify/long/asc <selftest_array_NVR + 8>
'NVR '
'    '
exit
! test_50_54 is reached via a pointer within this area, cf. below.
!
verify/inst test_50_54
'       CLRB    W^009F(R11)'
'       CASEL   R0,#0050,#4'
! 1$:   .WORD   LB_20048EDD - 1$        ! test 50 or 80000050 (status display)
!       .WORD   LB_20049026 - 1$        ! test 51 (q boot device)
!       .WORD   LB_20049098 - 1$        ! test 52 (q boot flags)
!       .WORD   LB_200490FD - 1$        ! test 53 (q halt flags)
!       .WORD   LB_2004914F - 1$        ! test 54 (q language)
! LB_20048ED6:
!       MOVZBL  #0017,R0                ! "?ILL CMD" message code
!       BSBW    S_20047731              ! output a message
!       RSB
exit
!
! we'll model "T 56" after this one ...
verify/inst test_53
'       pushr   #01E'                           ! save R1..R4
'       movab   b^<20049093 + 1 - ka410rom>,r1' ! -> 4 spaces
'       movzbl  b^<20049093 - ka410rom>,r0'     ! R0 := 4
'       bsbw    cons_output_r0r1'
'       movw    @#IO410$AB_CPMBX,-(sp)'
'       extzv   #6,#2,(sp),r0'                  ! R0 := old value
'       addl2   #030,r0'                        ! same in ASCII
'       movl    r0,r2'
'       bsbw    cons_output7b_r2'
'       clrl    r0'
'       bsbw    cons_input_num'                 ! [input a number]
'       tstl    r0'
'       bneq    t53_30$'                        ! [got one]
'       tstl    r1'
'       beql    t53_50$'                        ! [confirmed old value]
'       brb     t53_80$'                        ! [input error]
't53_30$: movl  (r1),r1'                        ! R1 := {input value}
'       clrl    r0'                             ! assume out-of-range error
'       cmpl    r1,#3'                          ! check range (0..3)
'       bgtru   t53_80$'                        ! br if no good
'       insv    r1,#6,#2,(sp)'                  ! ok, update stack copy
'       insv    r1,#2,#2,(sp)'
't53_50$: movl  #1,r0'                          ! success status
't53_80$: movw  (sp)+,@#IO410$AB_CPMBX'         ! re-store stack copy
'       popr    #001E'                          ! restore R1..R4
'       rsb'
exit
!
!---    NEW data & code for "T 56"
```

```
!
def t56_asciz           = free_area
deposit/word/asc <t56_asciz + 1>
'SC'
'SI'
'A:'
'  '
exit
ex/word
def free_area = .                                          ! start of "patch area"
def free_size = <ka410rom_cksum - ka410rom> - free_area ! size of "patch area"
!
deposit/byte t56_asciz                  ! fill in string length
+<free_area - <t56_asciz + 1>>
exit
!
def test_56             = free_area
deposit/inst test_56                    ! (after TEST 53 code, above)
'       clrb    w^009F(r11)'            ! [from test_50_54]
'       pushr   #01E'                   ! save R1..R4
'       movab   b^<t56_asciz + 1>,r1'   ! R1 -> "SCSIA:  "
'       movzbl  b^t56_asciz,r0'         ! R0 := string length
'       jsb     l^cons_output_r0r1'
'       movw    @#IO410$AB_SCSI,-(sp)'  ! fetch NVRAM byte (left shifted by 2)
'       extzv   #2,#3,(sp),r0'          ! R0 := old hostid
'       add12   #030,r0'                ! convert to ASCII
'       movl    r0,r2'
'       jsb     cons_output7b_r2'
'       clrl    r0'
'       jsb     cons_input_num'         ! [input a number]
'       tstl    r0'
'       bneq    t56_30$'                ! [got one]
'       tstl    r1'
'       beql    t56_50$'                ! [confirmed old value]
'       brb     t56_80$'                ! [input error]
't56_30$: movl  (r1),r1'                ! R1 := {input value}
'       clrl    r0'                     ! assume out-of-range error
'       cmpl    r1,#7'                  ! check range (0..7)
'       bgtru   t56_80$'                ! br if no good
'       insv    r1,#2,#3,(sp)'          ! ok, update stack copy
't56_50$: movl  #1,r0'                  ! success status
't56_80$: movw  (sp)+,@#IO410$AB_SCSI'  ! re-store stack copy
'       popr    #001E'                  ! restore R1..R4
'       rsb'
exit
ex/inst
def free_area = .                                          ! start of "patch area"
def free_size = <ka410rom_cksum - ka410rom> - free_area ! size of "patch area"
!
!---    dispatch to test_56
!
def t5x_dispatch        = free_area
def phy_t5x_dispatch    = t5x_dispatch + ka410rom
deposit/inst t5x_dispatch
'       cmpl    r0,#0056'               ! Test 56 ?
'       beql    test_56'                ! br if so
'       jmp     l^test_50_54'
exit
ex/inst
def free_area = .                                          ! start of "patch area"
def free_size = <ka410rom_cksum - ka410rom> - free_area ! size of "patch area"
!
!---    point to NEW "T 5x" dispatcher
!
replace/long <selftest_array_NVR + 0040>
+phy_test_50_54
exit
```

```
+phy_t5x_dispatch
exit
!
!-------------------------------------------------------------------------------
!-----  part 3: within VMB, replace TVBTDRIVER by (own) DK2KBTDRIVER
!
def btab_mua            = 200537D9 - ka410rom
!
def TVBTDRIVER          = 2005630D - ka410rom
def TVBTDRIVER_DRIVRNAME= 200565D7 - ka410rom
def TVBTDRIVER_AUXDRNAME= 200565E4 - ka410rom
def TVBTDRIVER_DEVNAME  = 200565F1 - ka410rom
def TVBTDRIVER_UNIT_INIT= 20056616 - ka410rom
def TVBTDRIVER_ENTRY    = 200566E1 - ka410rom
def TVBTDRIVER_UNIT_DISC= 200576E5 - ka410rom
def TVBTDRIVER_END      = 200576FF - ka410rom   ! ESBTDRIVER starts here
!
def BOO$DRIVER_TBL      = 20058500 - ka410rom   ! ^d10 longwords per driver:
!                                               ! DU, TV("MU"), ES
!
def BTD$K_SCSI_5380_TAPE = 0025 !! "boot device code" associated with MUA0
def BTD$K_SCSI_5380_DISK = 002A !! I'd like to use this for DKAn, however
!                               !! (as of VMS V5.5-2) INIADP410 will map the
!                               !! SCSI port only for BTD$K_SCSI_5380_TAPE.
!
!***
!
replace/word/ascii btab_mua     ! .ascii "MU" -> "DK"
'MU'
exit
'DK'
exit
!
verify/word btab_mua + 2        ! .asciz "A"
+0041
exit
!
replace/word btab_mua + 4       ! maxunit
+0000
exit
+0007
exit
!
verify/word btab_mua + 6        ! boot device type, must match BOO$DRIVER_TBL
+BTD$K_SCSI_5380_TAPE           ! Can't change this (see above comments);
exit                            ! old value is fine, with at least VMS V5.5-2
!
verify/long btab_mua + 8
+IO410$AB_SCTLS
exit
!
!***
!
def drv_base = TVBTDRIVER
!
!@@@@ start DK2KBTDRIVER.PAT  2-APR-1999 13:08:14.48
!
! Data generated from DK2KBTDRIVER.BIN
!
def DK2KBTDRIVER_ENTRY  = drv_base + 0000025C
def DK2KBTDRIVER_DRIVRNAME      = drv_base + 00000050
def DK2KBTDRIVER_AUXDRNAME      = drv_base + 0000005D
def DK2KBTDRIVER_UNIT_INIT      = drv_base + 000000A5
def DK2KBTDRIVER_UNIT_DISC      = drv_base + 00000FA1
def DK2KBTDRIVER_DEVNAME        = drv_base + 0000006A
!
!
```

```
def DK2KBTDRIVER_LENGTH = 000013AB
!
deposit/long drv_base   ! replace previous driver
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
0FFFFFFFF
000000000
0444B440C
045564952
058452E52
04B500C45
052444B32
0452E5256
04B444558
000001B06
006000100
000000000
003060000
008000000
000080600
000000000
00000280A
000000000
006000000
00000000A
02A0A0000
000000000
000000000
09501FC00
00F1282AF
09F0302EF
0200B00BC
001509C50
03CFF71CF
0125564A9
06BCF9107
0281301FF
0648F55D1
018000000
00755D11F
08FD00815
000000908
0559C0450
0FF4CCF01
000648FC5
050550000
064A950B0
0FF3CCF91
013FF35CF
0F88FB0DD
0DB1EA901
```

```
040D05038
08A5754A9
0D008A702
0F2305005
04450E800
0305005D0
050E800E9
08F50D13B
0000001A4
0CA302013
0408FDD00
0F5000F42
08ED5FD6E
0305005D0
050E800C9
08F50D11B
0000001A4
093300512
0D0C91100
0003A988F
0AE305000
00E50E900
008EF00E2
003000000
0D0001930
000045009
000101506
00C000C00
008000000
000000000
000020000
0E8049A30
03F300350
0E0AF7D0A
030FE75CF
050E80554
00A303003
0E8001030
027300350
005C7300A
0300350E8
07D050A1E
0C5307E51
00350E806
09A0A1130
0D2305000
00350E806
09E0A0530
09A52ADAF
082905182
005463050
093F751F5
01214A708
05001D0FA
005518E7D
0FE8CCF7D
030FE1DCF
050E90433
004F63009
0300350E9
090050575
004A7808F
06EF50CDD
0948ED5FD
exit
deposit/long drv_base+00000200
0DD0504A7
```

```
0FCEECFFD
0CF7D2C11
0E3CFFD6E
0E7CFD4FC
0CF5090FC
0CF90FCE3
0D9CFFCFA
0F4CF90FC
0FCD1CFFC
0FCEECF90
090FCC9CF
0CFFCE8CF
0D430FCC1
05050E902
0E8039730
073300350
0051E3008
03050019A
050E90531
005123057
014A70893
067904E13
0B3CFD550
0D70715FC
030FCADCF
0CC300227
0E158F503
0E803E830
03D300650
01050E900
0FC94CFD5
0CFC00A15
08FCFFC96
0FF2031FC
0FC7CCFD0
073CFD058
0CFC25AFC
06FCFFC7A
06FCFD0FC
0CFD055FC
00551FC66
0E903B030
007300350
000023008
08FDDD611
000000054
080CF0091
0D04C12FC
00008708F
001D06E00
0CF05E16E
0E009FC30
0FC28CF07
007DC3036
0CF0400EF
09150FC20
027135001
001F48FD0
0916E0000
01B135003
exit
deposit/long drv_base+00000400
000548FD0
0916E0000
00F135004
001A48FD0
0916E0000
```

```
003135002
0D007A930
0DB05508E
0FC20CF38
0FC1CCFCE
0C0FC19CF
013CF14AC
004ACD0FC
008ACD05A
00CACD058
0CF52D055
009EFFBF6
0F3CF5A15
0CF5AD0FB
06230FBF2
0CF55D001
055D0FBC6
0D0FBC9CF
0FBB4CF5A
0B3CF58D0
0CF58D0FB
0507CFBB6
0FBB0CFD0
0008F7B50
050000002
051D55150
050D60213
0FF8F50D1
015000000
0FF8F9A04
0CF50D050
08FC5FB96
000000200
0CFD15850
0FF8FFB86
01A001FFF
0EACF7D23
0FB54CFFB
054CF5090
06FCF90FB
0FB4CCFFB
0FB69CF90
090FB44CF
0CFFB63CF
02C11FB3C
0FBCECF7D
0D4FB31CF
090FB35CF
0FB31CF50
0FB48CF90
090FB27CF
0CFFB42CF
0CF90FB1F
017CFFB3C
036CF90FB
0FB0FCFFB
0E9012230
0E5307250
00350E801
03006C130
0009A036C
0037F3050
0D47950E9
00BCFD550
0D70715FB
030FB05CF
008930078
```

```
0641314A7
010A72093
05090F413
004A79067
0E08F8A50
050018850
004A75090
004306ED5
004A79002
0E18F8A50
0A7509050
0BF58F504
0E8021430
069300650
01050E9FE
0FAC0CFD5
0CFC00A15
0BBCFFAC2
0FEFE31FA
0FAA8CFD0
09FCFD058
0CFC25AFA
09BCFFAA6
09BCFD0FA
0CFD055FA
00451FA92
0E901DC30
033300350
0FE2E3006
0DF90D611
01150FAA6
0DF509005
0CFD6FA9E
08FB3FA9A
093CF01FF
0D62A12FA
0D5FA89CF
013FA8DCF
0CA1F1920
0FFFE008F
0FA7CCFFF
07D7E567D
056FA6DCF
0F06647DD
0CF15098E
08E7DFA6A
08FCA0556
0FFFFFE00
07DFA5DCF
0EF9E7E56
000000BBF
0FF8FCA56
056000001
047CF56C8
01509EFFA
0A9D05656
exit
deposit/long drv_base+00000600
046DE5750
08FC95767
090000000
067FA2DCF
0FA2CCFDA
0568E7D3A
00CA79405
0501CA790
0FA08CF90
```

```
0A7018867
0F48FD008
0CF000001
0A790FA02
08F935004
008125040
0F9F4CFF5
0009631F1
06EF501DD
0908ED5FD
0935004A7
006135020
008A7018A
0CF8BCB11
05067F9CE
0C7CF5091
090ED1AF9
0935004A7
0E4125020
05004A790
050E08F8A
090500488
0DD04A750
0FD6EF501
0CF898ED5
0A3CFF9A2
0509050F9
004A79067
0E08F8A50
050098850
004A75090
008A7018A
09010A794
08A5004A7
09050E88F
0D004A750
007A1208F
0F978CF00
0A7408F93
0F5071210
0F4F96DCF
0A7901011
08F8A5004
0509050E5
001D004A7
08FD00550
000000054
0517D0550
0018A307E
03050029A
050E9019D
00BCF9E1E
0829A52F9
050829051
0F5001330
00893F751
0FA1214A7
07D5001D0
03005518E
0089304B2
0031214A7
09304A930
01310A720
0675090F1
05004A790
050E08F8A
090500188
```

```
0D504A750
0000D306E
05004A790
050E18F8A
004A75090
004A79005
0E08F8A50
050108850
004A75090
010A72093
0A790FA12
08F8A5004
0509050F0
0D50504A7
018F8DDCF
0FF8F9006
0D6F8DACF
030F8D1CF
0039A00F8
0010B3050
0300350E8
0CFD0043E
09050F8BE
0BBCF4067
0FFB130F8
09A00DB30
0EE305007
00350E800
0D0042130
050F8A1CF
0CF406790
09430F8A0
0408F93FF
0F91210A7
0501CA790
000548FDD
0CFD00000
09150F882
07FCF4000
0910B12F8
079CF4000
0D00312F8
06EE86E01
067CFD509
0300314F8
08ED0001E
05BCFD750
01350E8F8
058CF0091
0ED0C12F8
011CF0400
0031201F8
exit
deposit/long drv_base+00000800
0055001D0
07D7E517D
0CFF86FCF
00830F7F2
00350E8FE
03003AD30
050E8FEC8
003A43003
09A004F30
062305001
04350E900
09E5108D0
052F7D9CF
```

```
093003B30
01214A708
003843003
030826790
051F5FEFF
0BFCF9EEB
0A29052F7
0A2905006
09006A203
09003A250
0905005A2
005A204A2
004A25090
07DFEFC30
03005518E
08F930352
01210A740
003483008
050548F9A
0A7209305
0D0EB1310
0DD055001
010A79051
00302EF51
050915151
090221251
0D00CA751
00003E88F
0F78CCF00
014A70893
0CFF50812
030F5F782
001D0030E
0518ED050
01150D405
0000000F8
000000000
000000000
000000000
000000000
0324B4400
04454424B
045564952
061682052
06E69746C
074612067
06C657220
076697461
064612065
073657264
000002073
020200000
04B324B44
052445442
052455649
073616220
064612065
03D207264
000000020
000000000
0520D0A00
053494745
053524554
052202020
000203A30
000000000
020000000
```

```
031522020
00000203A
000000000
020200000
03A325220
000000020
000000000
020202000
0203A3352
000000000
000000000
020200D0A
020202020
020202020
034522020
00000203A
000000000
020200000
03A355220
000000020
000000000
020202000
0203A3652
000000000
000000000
052202020
000203A37
000000000
00A000000
02020200D
020202020
020202020
03A385220
000000020
000000000
020202000
0203A3952
000000000
000000000
031522020
000203A30
000000000
020000000
031315220
00000203A
000000000
00D0A0000
020202020
exit
deposit/long drv_base+00000A00
020202020
020202020
0203A5041
000000000
000000000
046202020
000203A50
000000000
020000000
050532020
00000203A
000000000
020200000
03A435020
000000020
000000000
0430D0A00
```

```
05542444D
000203A46
000000000
020000000
000000000
000000000
045532020
05F45534E
03A465542
000000020
000000000
000002000
000000000
00D0A0000
054415453
0425F5355
0203A4655
020200000
020202020
020202020
053202020
045534E45
04154535F
03A535554
00A000020
047534D0D
05F4E495F
03A465542
020000020
020202020
020202020
020202020
0534E4553
0534D5F45
04E495F47
00000203A
020202020
020202020
04E455320
0495F4553
05845444E
00000203A
000000000
00D0A0000
052455355
04655425F
04444415F
000203A52
000000000
020000000
055202020
05F524553
05F465542
03A4E454C
000000020
000000000
020202000
020572F52
045545942
0454C5F53
0203A5446
000000000
000000000
047500D0A
04241545F
0505F454C
0203A5254
```

```
000000000
000000000
020202020
053414220
050565F45
000203A4E
000000000
020000000
020202020
055422020
052454646
05254505F
00000203A
000000000
00D0A0000
05F50414D
045444F4D
00000203A
000000000
00D0A0000
043415453
020203A4B
000000000
000000000
020200D0A
020202020
000002020
000000000
00D0A0000
0F438CF9F
0CF6E8EC3
0AE9EFCF3
08FBB7E04
038DB3FFF
0D0081350
000DEAD8F
030005000
0CFD0038F
011CFFCDF
00FCF9EFD
0FCCDCFF4
07D037E30
0CFFCCACF
08FD0FD1F
exit
deposit/long drv_base+00000C00
000000052
09EFCC6CF
0CFFCCACF
0CF9EFCC3
01650FCBB
004004C9F
0CF8ED020
05430FCA3
0A0CF7D03
0FD0FCFFC
094CF8ED0
0034530FC
0FC91CF7D
0D0FD0FCF
0FC85CF8E
07D033630
0CFFC82CF
08ED0FD0F
030FC76CF
0CF7D0327
00FCFFC73
```

```
0CF8ED0FD
01830FC67
064CF7D03
0FD1ACFFC
058CF8ED0
0030930FC
0FC55CF7D
0D0FD1ACF
0FC49CF8E
07D02FA30
0CFFC46CF
08ED0FD1A
030FC3ACF
0CF7D02EB
01ACFFC37
0CF8ED0FD
0DC30FC2B
028CF7D02
0FD25CFFC
01CCF8ED0
002CD30FC
0FC19CF7D
0D0FD25CF
0FC0DCF8E
07D02BE30
0CFFC0ACF
08ED0FD25
030FBFECF
0CF7D02AF
025CFFBFB
0CF8ED0FD
0A030FBEF
0ECCF7D02
0FD30CFFB
0E0CF8ED0
0029130FB
0FBDDCF7D
0D0FD30CF
0FBD1CF8E
07D028230
0CFFBCECF
08ED0FD30
030FBC2CF
0CF7D0273
030CFFBBF
0478FD0FD
0CF000000
0CF9EFBBB
0B8CFFC11
0B0CF9EFB
09F1650FB
02004004C
000478FD0
0A0CF0000
03DCF9EFB
0FB9DCFFC
0FB95CF9E
04C9F1650
0D0200400
00000478F
0FB85CF00
0FC69CF9E
09EFB82CF
050FB7ACF
0004C9F16
08FD02004
000000047
```

```
09EFB6ACF
0CFFC95CF
0CF9EFB67
01650FB5F
004004C9F
087CFD020
0FB45CFF2
07D01F630
0CFFB42CF
0CFD0FCCE
034CFF27A
001E530FB
0FB31CF7D
091FCB4CF
006F265CF
08FB00714
0A6CF3030
063CFD0FC
0FB15CFF2
07D01C630
0CFFB12CF
0CFD0FCBC
004CFF256
001B530FB
0FB01CF7D
0D0FCA2CF
0FB01CF39
0FC73CF9E
09EFAFECF
050FAF6CF
0004C9F16
0CFD02004
0DCCFF26A
0018D30FA
0FAD9CFB0
0B0FCE3CF
0CFFAD4CF
0CFB0FCBF
0A8CFFACF
0CACFB0FC
exit
deposit/long drv_base+00000E00
0FC84CFFA
0F240CFD0
030FAB6CF
0CF7D0167
0D5CFFAB3
0CF2DD0FC
0CF9EFAB3
0B0CFFC5E
0A8CF9EFA
09F1650FA
02004004C
0004B8FD0
098CF0000
070CF9EFA
0FA95CFFC
0FA8DCF9E
04C9F1650
0D0200400
0CFF1C9CF
02430FA73
070CF7D01
0FCABCFFA
0F1BCCFD0
030FA62CF
0CF7D0113
```

```
0B4CFFA5F
0B3CFD0FC
0FA51CFF1
07D010230
0CFFA4ECF
08FD0FCBE
00000004E
09EFA4ACF
0CFFC6DCF
0CF9EFA47
01650FA3F
004004C9F
09FCFD020
0FA25CFF1
07D00D630
0CFFA22CF
0CFD0FCAA
014CFF192
000C530FA
0FA11CF7D
0D0FCB0CF
0CFF185CF
0B430FA03
000CF7D00
0FCBACFFA
0004A8FD0
0FCCF0000
06DCF9EF9
0F9F9CFFC
0F9F1CF9E
04C9F1650
0D0200400
0CFF15DCF
08830F9D7
0D4CF7D00
0FCA2CFF9
0D4CF14D0
08FCF9EF9
0F9D1CFFC
0F9C9CF9E
04C9F1650
0D0200400
066D0565E
030F9AECF
0CF7D005F
08BCFF9AB
0CF12D0FC
0CF9EF9AB
0A8CFFC7A
0A0CF9EF9
09F1650F9
02004004C
09E5205D0
0D05604A6
0F981CF66
07D003230
0CFF97ECF
012D0FC70
09EF97ECF
0CFFC5FCF
0CF9EF97B
01650F973
004004C9F
0D352F520
032313000
036353433
041393837
```

```
045444342
0D00FBB46
051F946CF
09E5208D0
053F941CF
050FC8F79
0E48F7850
08FCA5050
0FFFFFFF0
0AF409050
0F56342CC
00FBAE652
0DB008005
040D05038
0045754A9
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
exit
deposit/long drv_base+00001000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
```

```
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
```

```
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
exit
deposit/long drv_base+00001200
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
```

```
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
000000000
```

wjm 24-feb-199: *TEST* versions of PK2KDRVR (& maybe, assorted patches)

Here is a collection of PK2KDRVR executables
for various VMS versions that I have around,
made _without_ _testing_ them myself.
(I have no Vs2000 running anything but V5.5-2,
where the "official" version has worked very well
over time - for me, that is). So this here is ...

>>> *highly* EXPERIMENTAL software, NO WARRANTIES at all! <<<

It will be difficult for me to fix them,
in case they don't work quite right ...
(but better ask anyway, PLEASE).

I'd appreciate any kind of feed-back,
in case you dare trying them out.

The .ZIP file names in this directory
include the VMS version, e.g.
PK2K-053-TEST.ZIP is for VMS V5.3.

Hacker's note:
Each of the versions here required
some change to the V5.5-2 sources.
DO NOT just assume they'll work for you.

I'll gladly assist you in building PK2K
for VMS versions that I don't have,
provided you'll share the result with me.

Wolfgang J. Moeller, D-37077 Goettingen, Germany <moeller@gwdg.de>

```
wjm 19-aug-1997: PK2KDRVR & assorted patches to system programs, V1.2

Changes over initial version: Add fixes, plus one more check, for several
                              cases of bad register usage in PKNDRIVER.

        *** This is for VMS V5.5-2 only ***

PK2KDRVR is a driver for the Vs2000/uVAX2000 SCSI port
(traditionally known as the "tape controller port").

        >>> EXPERIMENTAL software, NO WARRANTIES at all! <<<

Apart from the restriction mentioned next, it _ought_
to behave just like the Vs3100 SCSI driver (PKNDRIVER);
the SCSI host adapter will have SCSI id 0 (not 6 or 7!).

Known restriction:
        PK2KDRVR won't do data transfers of 16kB or more.

        This has the effect of limiting the block size
        that can be used with SCSI tapes, and also will
        break any program that _attempts_ to read 16kB or more.
        The only VMS program that does so (which I'm aware of)
        is DUMP - see below for a patch.


Disclaimer:
        This is *EXPERIMENTAL* SOFTWARE that theoretically
        *could* not only crash your system, but *could* cause
        CORRUPTION on all media connected to the computer on
        which it's installed.
        (In fact, PK2KDRVR has plenty of code that *attempts*
        to crash the system if a chance for corruption gets
        noticed, but the Vs2000/uVAX2000 hardware has never
        been "qualified" by anyone to work correctly with
        SCSI devices.)

        >>> NO WARRANTIES at all! <<<


Installation & use:

        *** This is for VMS V5.5-2 only ***

        The "binary kit" contains PK2KDRVR.EXE (not spelled PK2KDRIVER
        for quite "technical" reasons) plus 5 patch command files.

        Place PK2KDRVR.EXE in SYS$LOADABLE_IMAGES.

        Use
                $ PATCH @2KSYSGEN.COM
        to create 2KSYSGEN.EXE in the current directory.

        >>> Make sure that no MUA0 shows up, and that TVDRIVER
        >>> (the Vs2000/uVAX2000 magtape driver) is _not_ loaded.

        >>> NOTE that the SCSI host adapter has the SCSI id 0
        >>> (quite different from standard assignement of 6 or 7).

        Use
                $ MCR [dir]2KSYSGEN AUTOCONFIGURE ALL
        to load PK2KDRVR (ought to show up as device PKA0)
        and autoconfigure the SCSI devices, just like on a Vs3100.

        If you're confident enough in the driver that you want
        the machine to auto-configure the SCSI at boot time,
```