

Драйвер KZ

Полетаев В.М. ВЦ МИЭТ Тел. 534-65-36

Драйвер KZ предназначен для работы с дискетами записанными в формате Электроники-85 (DZ:) на ДВК, оснащенный контроллером с двойной плотностью записи (MY:) и 80-дорожечными накопителями на гибких магнитных дисках.

Драйвер поддерживает одновременную работу как с дискетами в формате DZ, так и в формате MY, причем может быть использован как системный для любого из этих форматов.

SET-параметры драйвера:

SET KZ SYSGEN

- устанавливает параметры генерации драйвера под параметры генерации системы.

SET KZ RETRY=n

- задает число повторов при ошибке ввода/вывода. По умолчанию n=8.

SET KZn DZ

- устанавливает, что привод n будет обрабатываться в формате DZ.

SET KZn MY

- устанавливает, что привод n будет обрабатываться в формате MY.

SET KZ INFORM

- распечатывает краткую информацию о драйвере, SET-параметрах и их текущих значениях.

Все SET-параметры исправляют драйвер и на диске, и в памяти.

КАК ПОЛЬЗОВАТЬСЯ ДРАЙВЕРОМ KZ?

Для того, чтобы пользоваться этим драйвером, нужно:

- Убедиться в том, что у вас есть контроллер с двойной плотностью и 80-дорожечные дисководы (т.е. емкость одной дискеты составляет 1600 блоков).

- Скопировать файл KZ.SYS себе на систему.

- Дать команды:

.SET KZ SYSGEN

.INSTALL KZ

Если все в порядке, то на экран выведется сообщение о версии драйвера.

- Установить необходимые форматы записи для каждого из имеющихся приводов. Для этого необходимо воспользоваться SET-параметрами DZ и MY, например:

.SET KZ0 MY

.SET KZ1 DZ

.SET KZ INFORM

Привод 0 будет работать в формате MY, привод 1 - в формате DZ. Для проверки распечатываются установленные значения.

- Теперь можно работать с устройствами.

ПРИМЕЧАНИЕ 1. Использование одновременно драйверов MY и KZ

НЕДОПУСТИМО. В каждый момент времени должен быть загружен только один из этих драйверов. Пользуйтесь KZ для работы с дискетами в формате MY.

ПРИМЕЧАНИЕ 2. Из-за особенностей контроллера MY дискета с Электронике-85 будет читаться только в том случае, если она была перед этим отформатирована на контроллере MY.

ИСПОЛЬЗОВАНИЕ ДРАЙВЕРА В КАЧЕСТВЕ СИСТЕМНОГО.

Как уже отмечалось, драйвер KZ может работать и как системный драйвер, т.е. с устройства KZ можно загружаться. Загрузка с дискеты в формате MY осуществляется стандартным способом:

```
@172140/000040 37<LF>
172142/XXXXXX <номер привода><CR>
@G
```

Загрузка с дискеты в формате DZ несколько сложнее:

```
@1000/XXXXXX <номер привода><LF>
001002/XXXXXX 0<LF>
001004/XXXXXX 401<LF>
001006/XXXXXX 400<CR>
@172140/000040 1<LF>
172142/XXXXXX 1000<CR>
@G
```

При загрузке драйвер сам определяет, с каким форматом дискеты он работает. Поэтому необязательно, чтобы установленный командой SET формат записи для данного привода соответствовал действительному формату дискеты. Определенный при загрузке формат системной дискеты будет записан в файл драйвера при первом же вызове его по команде SET (кроме .SET KZ SYSGEN).

ПРИМЕЧАНИЕ. Система, записанная на DBK, не будет загружаться на Электронике-85, т.к. на этих машинах стоят разные контроллеры. Для использования такой системы надо скопировать загрузчик из драйвера DZ на Электронике-85.


```

.MCALL .MODULE
MODULE KZ, RELEASE=V01, VERSION=00, COMMENT=<DZ on MY>, AUDIT=YES
.DSABL GBL
.ENABL LC, MCL
.MCALL .PRINT
; Last correction : Date 05/11/87 Time 00:12:41
DZMTRK==79. ; Максимальное число дорожек на DZ
ERLOG== 1 ; \ Необходимо для нормальной работы
TIMBIT==1 ; / SET KZ SYSGEN

```

```

.DRDEF KZ, 52, FILST0!SPFUN0!VARSD0, 800., 172140, 170
BR.DZ= <<DZ-MY>/2>&377

```

```

.ASECT
.=0
DNAM: .RMODULE
RETSPC: .BLKW 4
DVNAME: .ASCII /KZ0:/<11><200>
RETSTR: .ASCII /RETRY=/
RTRYST: .ASCIZ /0/

.ASSUME . LE 40

.=72
NONSYS: .ADDR #INSMES, R0 ; Не системный
.PRINT ; Распечатать через систему
RETURN

.ASSUME . LE 110

.=116
INSMES: .BYTE 15, 12
.NLCSI TYPE=I, PART=ALL
.ASCIZ /(C) /<16>/ВЦ МИЭТ, НОЯБРЬ 1987/<17><200><15><12>

.ASSUME . LE 170

.DRINS KZ
BR NONSYS ; Не системный
.ADDR #INSMES, R0 ; Системный
JSR R1, @#REP-KZBOOT ; Выдать сообщение
MOV @#B0DEVU, R1 ; Номер системного устройства
.ADDR #MYTABL, R1, ADD ; Адрес таблицы типов устройств
CLRB (R1) ; Пусть пока MY:
CMP @#BOTCMD-KZBOOT, #240
BEQ 100 ; Действительно MY:
MOVB #BR.DZ, (R1) ; Нет, это DZ:
100: BR D.GOOD

INFMES: .ASCII <15><12><16>/ ПОЛЕТАЕВ В.М. ТЕЛ. 534-65-36/<15><12><12>
.ASCII <17>/SET-/<16>/ПАРАМЕТРЫ:/<17>/ SYSGEN, INFORM, RETRY, DZ
.BYTE 12, 0

.ASSUME . LE 360 <; Use LINK/NOBITMAP !>
.ASSUME . LE 400

```



```

.SBTTL Set options
.DRSET RETRY 8. O.RTRY NUM
.DRSET MY 100000 O.MYDZ
.DRSET DZ BR.DZ O.MYDZ
.DRSET INFORM 0 O.INF
.DRSET SYSGEN 372 O.SYG

O.SYG: ADD @#54,R3 ; Адрес слова генерации системы
      MOVB (R3),60 ; Сохранить в драйвере

O.GOOD: TST (PC)+
O.ERR: SEC
      RETURN

O.RTRY: CMP R0,R3 ; Не более 8 попыток
      BHI O.ERR
      MOV R0,O.RTRY ; Записать на диск
      ASSUME O.RTRY-KZSTRT LT 1000
      CALL DSTAT ; Получить адрес MYTABL
      BCS O.GOOD ; Драйвер не загружен
      MOV O.RTRY,O.RTRY-MYTABL(R0) ; Исправить в памяти
      BR O.GOOD

O.MYDZ: CMP R1,#4 ; Номер привода 0..3
      BHIS O.ERR
      .ADDR #MYTABL,R1,ADD ; Адрес точки в MYTABL
      ASSUME MYTABL+4-KZSTRT LT 1000
      MOV R3,-(SP) ; Сохранить новое значение
      CALL DSTAT ; Получить адрес загрузки
      MOVB (SP)+,(R1) ; Восстановить новое значение
      BCS 100 ; Если не загружен, то 100
      MOV (R3)+,(R0)+ ; Если загружен, то обновить в памяти
      MOV (R3)+,(R0)+
100: BR O.GOOD

O.INF: CALL INSDAT ; Выдать сообщение
      .ADDR #INFMES,R0
      .PRINT
      CALL DSTAT ; Скопировать из памяти
      .ADDR #MYTABL,R1 ; Адрес таблицы типов устройств
100: MOV #0,R3
      MOV #MY,R0
      TSTB (R1)+
      BEQ 200
      MOV #DZ,R0 ; Имя устройства
200: MOV R0,DVNAME
      MOVB R3,DVNAME+2 ; Номер привода
      INC R3
      .ADDR #DVNAME,R0 ; Распечатать
      .PRINT
      CMP R3,#3 ; От 0 до 3
      BLE 100
      MOVB O.RTRY,RTRYST ; Число попыток
      BISE #0,RTRYST ; В символьный вид
      .ADDR #RETSTR,R0 ; Распечатать
      .PRINT
      RETURN

DSTAT: ; Если драйвер не загружен, то бит C=1
      ; иначе C=0, из драйвера в памяти переписывается
      ; таблица назначений (из-за возможного изменения
      ; ее при загрузке).
      ; R0 -> MYTABL в памяти
      ; R3 -> MYTABL в копии с диска

      .ADDR #DNAM,R0
      .ADDR #RETSPC,-(SP)
      EMT 342 ; DSTAT #RETSPC,#DNAM

```



```

BCS      0.ERR
MOV      RETSPC+4,R0      ; Адрес загрузки
CMP      R0,#1000
BLO      0.ERR
ADD      #MYTABL+4-KZSYS,R0 ; Адрес MYTABL в памяти
.ADDR    #MYTABL+4,R3      ; Адрес MYTABL в копии с диска
MOV      -(R0),-(R3)      ; Переписать из памяти
MOV      -(R0),-(R3)
BR       0.GOOD

```

```

.ASSUME .LT 1000

```



```

.SBTTL  Driver request entry point
.DRBEG  KZ
.ENABL  LSB
        MOV      KZCOE,R1
        MOV      (R1)+,R5          ; Номер блока
        MOVB     (R1)+,R3          ; Код спецфункции
        BITB     #4,(R1)          ; Номер привода от 0 до 3
        BEQ      300
KZERR1: JMP      KZERR
300:    MOVB     (R1)+,R0
        BICB     #^C3,R0
        MOVB     R0,UNIT
        MOV      (R1)+,BUFF        ; Адрес буфера
        MOV      #101,COMAND
        MOV      (R1)+,WCOUNT      ; Счетчик слов
        BPL      100
        MOV      #103,COMAND        ; Команда - запись
        NEG      WCOUNT
100:    .ADDR     #MYTABL,R0,ADD
        MOVB     (R0),(PC)
        BR       MY
.DSABL  LSB

```



```

.SBTTL MY type service
MY:: JSR R0,SPFUN
      .WORD 1600.
      MOV WCOUNT,WCNT
      CLR WCOUNT
      MOV #10,R2
100:  CMP #5000,R5
      BHI 200
      ADD #-5000,R5
200:  ROL R5
      DEC R2
      BGT 100
      MOV R5,R2
      CLRB R2
      SWAB R2
      INC R2
      CMP #10.,R2
      BGE 300
      BISB #4,UNIT
      SUB #10.,R2
300:  BR NEXTMY

```

```

MYTABL: .BYTE 0
        .BYTE 0
        .BYTE 0
        .BYTE 0

```

```

.ASSUME .-KZSTRT LT 1000

```



```

.SBTTL DZ type service
DZ:: JSR R0,SPFUN
      .WORD 800.
      MOV (PC)+,R4
      .BYTE -7.,-10.
400:  CMP #1280.,R5
      BHI 500
      ADD #-1280.,R5
500:  ROL R5
      INCB R4
      BLE 400
      MOVB R5,R1
      ADD R4,R5
      MOV R1,R4
      ASL R1
600:  SUB #10.,R1
      BGT 600
      MOV R1,(PC)+
TRKOFF: .WORD 0
      MOV R5,(PC)+
TRACK:  .BYTE 0
SECTOR: .BYTE 0

NEXT:
200:  MOVB SECTOR,R2
      BGT 400
      SUB #-6,R2
      BGT 300
      ADD #4,R2
      SEC
300:  ROL R2
      ADD TRKOFF,R2
      BGT 400
      ADD #10.,R2
; R2 IS SECTOR NUMBER
400:  MOVB TRACK,R5
      CMPB R5,#DZMTRK+1
      BGT KZERR
      BNE 450
      CLR R5
; R5 IS TRACK NUMBER
450:  MOV #256.,WCNT
      CMP WCOUNT,WCNT
      BHIS NEXTMY
      MOV WCOUNT,WCNT
      .BR NEXTMY

```

```

; Один блок
; А сколько нужно?
; Если больше, то так и оставить
; Иначе столько, сколько нужно.

```



```

.SBTTL Common service
NEXTMY: MOVB R2,SECT
        MOVB R5,TRK
        MOV (PC)+,(PC)+
OdrTRY: .WORD 8.
RETRY: .WORD 8.

REPEAT: .ADDR #BLPAR,R0 ; Адрес блока параметров
        MOV (PC)+,R4
KZCSR: .WORD KZCSR
100: BIT #40,(R4) ; Ожидание бита "завершено"
      BEQ 100
      MOV (PC)+,(R4) ; Команда
COMAND: .WORD 0
200: TSTB (R4) ; Ожидание запроса данных
      BPL 200
      MOV R0,2(R4) ; Адрес блока параметров
      RETURN

      .DRAST KZ,5,KZFIN ; Точка входа по прерыванию
      MOV KZCSR,R4
      BIC #100,(R4) ; Запретить прерывания
      .FORK KZFBLK
      TST (R4) ; R4 и R5 сохраняются
      BPL NEXTBL ; Переход, если нет ошибки
      DEC RETRY ; Уменьшить число попыток
      BGT REPEAT ; Повтор при ошибке
KZERR: MOV KZCQE,R4 ; Выход по ошибке
      BIS #HDERR0,@-(R4)
KZFIN: CLR KZFBLK+2
      .DRFIN KZ

NEXTBL: ADD #512.,BUFF
        SUB #256.,(PC)+
WCOUNT: .WORD 0
        BLE KZFIN

500: INCB SECTOR
      BNE 600
      ADD #-10.*256.+1,TRACK
      ADD #2,TRKOFF
      BLE 600
      SUB #10.,TRKOFF
600: JMP NEXT

```



```

.SBTTL  SPFUN service
SPFUN:  MOV     (R0)+,R2
        TSTB   R3
        BEQ    100
        MOV     (SP)+,R0
        CMPB   R3,#373
        BNE    KZERR
        MOV     R2,@BUFF
        BR     KZFIN
100:    RTS     R0

```

BLPAR:

```

UNIT:   .WORD   0      ; \
BUFF:   .WORD   0      ; \
SECT:   .BYTE   0      ; > Блок параметров для контроллера MY.
TRK:    .BYTE   0      ; /
WCNT:   .WORD   0      ; /
KZFBLK: .WORD   0,0,0,0

```



```

.SBTTL Bootstrap read routine
.DRBOT KZ,ENTRY,READ
.=KZBOOT+14
    .WORD WAIT40-KZBOOT           ; Вектор прерывания по BFT
    .WORD 340
.=KZBOOT+40
ENTRY: JMP BOOT1

.=KZBOOT+100
    .WORD 102
    RTI
WAIT40: BIT #240,@#KZDCSR
    BEQ WAIT40
    RTI

READ:
    MOV @#BDEVU,BCMD
    CALL BINPUT                   ; Считать загрузчик с DZ
    MOV #READ1-KZBOOT,@#BREAD
    CMP @#RMODULE-KZBOOT+6000,RMODULE ; Это DZ:7
    BEQ READ1                     ; Да
    MOV #240,BOTCMD               ; Нет - заслать NOP в BOTCMD

READ1: BIC #^C3,BCMD
    MOV BCMD,R3
    ASL R3
    MOV R1,BCMD+6                 ; Счетчик слов
    MOV R2,BCMD+2                 ; Адрес буфера
BOTCMD: BR DZBOOT

```


.SBTTL Bootstrap from MY

MYBOOT::

```
MOV      #10,R3
200:    CMP      #5000,R0
        BHI      300
        ADD      #-5000,R0
300:    ROL      R0
        DEC      R3
        BGT      200
        SWAB     R0
        INC      R0
        CMPB     #10.,R0
        BGE      400
        BISB     #4,BCMD
        SUB      #10.,R0
400:    MOV      R0,BCMD+4
```

; Номер сектора / Номер дорожки

BINPUT::

```
MOV      #10,R3
500:    MOV      #KZCSR,R4
        BPT
150:    MOV      #1,(R4)+
        BPT
        MOV      #BCMD-KZBOOT,(R4)
        BPT
        TST      -(R4)
        BPL      800
        SOB      R3,150
        BR       BIDERR
800:    RETURN
```


.SBTTL Bootstrap from DZ

.ENABL LSB

DZBOOT::

```
      MOV      (PC)+,R4
      .BYTE    -7.,-10.
100:   CMP      #1280.,R0
      BHI      200
      ADD      #-1280.,R0
200:   ROL      R0
      INCB     R4
      BLE      100
      MOVB     R0,R3
      ADD      R4,R0
      MOV      R3,R4
      ASL      R3
300:   SUB      #10.,R3
      BGT      300
      MOV      R3,TRKOF
      MOV      R0,(PC)+
BTRK:  .BYTE    0
BSEC:  .BYTE    0
500:   MOVB     BSEC,R0
      SUB      #-6.,R0
      BGT      600
      ADD      #4.,R0
      SEC
600:   ROL      R0
      ADD      (PC)+,R0
TRKOF: .WORD    0
      BGT      700
      ADD      #10.,R0
700:   MOVB     BTRK,BCMD+5
      MOVB     R0,BCMD+4
      MOV      #256.,BCMD+6
      CMP      R1,#256.
      BHS      800
      MOV      R1,BCMD+6
800:   CALL     BINPUT
      INCB     BSEC
      BNE      1100
      ADD      #-10.*256.+1,BTRK
      ADD      #2,TRKOF
      BLE      1100
      SUB      #10.,TRKOF
1100:  ADD      #512.,BCMD+2
      SUB      #256.,R1
      BGT      500
      CLC
      RETURN
.DSABL LSB
```

```
BOOT1: MOV      #10000,SP
      BPT
      MOV      @#KZCSR+2,R5
      SWAB     R5
      BIC      #^C3,R5
      MOV      R5,@#BDEVU
      MOV      #2,R0
      MOV      #2000,R1
      MOV      #1000,R2
      CALL     READ
      MOV      (PC)+,@(PC)+
```

```
RMODULE: .RMODULE
      .WORD    BDEVN
      MOV      #READ1-KZBOOT,@#BREAD
      MOV      R5,@#BDEVU
      JMP      @#BBOOT
```



```
BCMD: .WORD 0 ; UNIT\
      .WORD 6000 ; BUFF \
      .BYTE 1 ; SECT > Блок параметров для контроллера MY.
      .BYTE 1 ; TRK /
      .WORD 400 ; WCNT/
      .DREND KZ
      .END
```