INFO CWD M

PERSONAL COMPUTER ZX-SPECTRUM

Yu. POMORTZEV

TR-DOS

for professionals and amateurs

Moscow - 1994

PDF version by Deny (Denisenko DA)

e-mail: DenyDA@mail.ru

2007

Page 2

The book contains an in-depth description of the TR-DOS operating system and is intended for for those who write programs in the machine code for computers ZX-Spectrum.

For an initial acquaintance with TR-DOS, the reader should study the brand "User's Guide", which describes in detail the command system and other similar information.

Some of the information in this guide is given below as a brief reference.

AUTHOR

Page 3

CONTENT

Page 4

INTRODUCTION

Recently, with computers ZX SPECTRUM was widely distributed BETA-disk interface with a disk operating system TR-DOS (version 5.0x) firm Technology Research Ltd. This system radically changes the possibilities of the good old SPECTRUM, turning it into a very effective tool.

The shortest characteristics of TR-DOS are as follows:

EASY AND FAST ACCESS - download any program available on the disk, occurs in a few seconds;

WIDE SELECTION OF DISCUVERS - 5.25 "or 3.5", 40 or 80 tracks, single or two-sided (see below);

CAPACITY - Up to 4 drives at a time. Data is stored with double density, which gives a total capacity of up to 2.5 MB;

FLEXIBILITY - the interface is independent, you can use disk drives with other computers;

COMPATIBILITY - the interface can be used with computers SPECTRUM, SPECTRUM + and SPECTRUM 128;

AUTOSTART - automatically launches the BASIC program, "boot" after power-up or reset (SPECTRUM 48 only);

AUTOMATIC CHECK - automatically checks and reconciles the system with drive parameters;

"MAGIC" KEY - thanks to it you can reset to the floppy current program (see below);

SYSTEM SWITCH - allows you to disable the interface for the standard configuration or used for reset;

WORKING WITH FILES - TR-DOS works with all types of files that are available SPECTRUMa - programs (B), codes (C), numeric and Signed arrays (D), and, in addition, with disk files with sequential and random access, whose type is denoted as (#);

SYNTAX - The most simple, using keywords;

TR-DOS is contained in the PROM and uses only 128 bytes of RAM.

BRIEF LIST OF TR-DOS COMMAND

* "A:"
* "B:"
* "C:"
* "D:"
40
80
CAT
CAT # 3
CLOSE #
COPY
COPY s
COPY B
FORMAT
GO TO
ERASE
LIST
LIST # 3
LOAD
INPUT #
MERGE
MOVE
NEW
OPEN #
PEEK
POKE
PRINT #
RANDOMIZE USR 15616
RANDOMIZE USR 15619: REM:
RETURN
RUN
SAVE
VERIFY

Examples:

FORMAT "$ NAME" - format the disc as one-sided.

COPY "new", "old" CODE - copying on one disk.

COPY s "name" - copy to another disk on one drive.

COPY "A: name", "B: name" - copy the file from drive B: to A :.

COPY "B: *", "A: *" - copy all files from the drive A: to B :.

NEW "new", "old" - rename the file (only in DOS).

GO TO "$ name" - start the MAGIC-CODE file with IM2 interrupts.

PEEK "name" address, sector - read sector N from file to memory.

Page 5

POKE "name" address, sector - write from memory to sector file N (N from unit to number, specified in CAT for this file).

OPEN # N, "name", W - open a sequential disk file for writing (WRITE), with the number channel N from 4 to 15.

OPEN # 4, "name", R - the same, but for reading (READ).

PRINT # 4; "TEXT" - output to disk file with channel number 4.

INPUT # 4; T $ - input from a file into a string variable.

OPEN #N, "name", RND, record length - open a **random** -access disk file (recording and reading).

PRINT # 4; record number, T $ - T $ output to a random access file.

INPUT #N; (record number), T $ is the entry of a file entry into a variable.

CLOSE #N - close the disk file with the number N (required!).

Error codes when calling from BASIC

The error codes can be obtained from the DOS variable 23823 or via the BASIC variable (for example, err) as follows:

LET err = USR 15619: REM: The DOS command

Meaning of error codes:

0 - no errors
1 - no file
2 - the file already exists
3 - there is no disk space
4 - the catalog is full (128 files are already written)
5 - overflow of the record number
6 - no disc
7 - error on disk
8 - syntax error

10 - the channel is already open
11 - the disk is not formatted
12 - the channel is not open

Sometimes a design that has the following form INPUT # 9 * USR 15619: .., which If the download fails, the program stops.

DISCS AND DISCONDS

Most often, a 5.25 dual density drive is used with the BETA controller. Drives of high density (1.2, 1.44MB) to use does not make sense. this mode is not supported. If you have a 40-track drive, you will have problems with overwriting, when exchanging programs, because 80-track disks on such a drive read, and 40-track on the 80-track - read, but not written. Let me remind you that the regime The "40" on the 80-track drive does not provide a standard 40-track recording (see p. firm instruction)!

A drive with a double density is called an 80-track drive, although in practice it provides access to 84 ... 86 tracks (depending on the specific drive and floppy disk). AT There is some confusion about the numbering of tracks, because sometimes cylinder numbers are used (pairs of tracks, 0 ... 85) and the side number (0,1; 0-upper), and sometimes - end-to-end numbering (0 ... 171; even numbers are upper). When using TR-DOS, both options are found. Access to tracks 80 ... 85 are possible when calling procedures from ROM; standard DOS commands, it does not supported.

TR-DOS provides 16 sectors per track and 256 bytes per sector. Reduced (in MS-DOS, for example, sector = 512), the sector size has the advantage of storing a large number short files; Disk space is used more sparingly (if the file has, to for example, the length is only 1 byte, on the disk it will still occupy the whole sector). In addition, with The use of random access files increases the speed of operations.

Page 6

This makes it possible to count the number of sectors and the capacity formatted disc (given that TR-DOS uses an external [0] storage path system information about the disk):

40 tracks one-way =	38 * 16 = 624 sectors * 256 = 156 KB
40 tracks double-sided =	79 * 16 = 1264 sectors * 256 = 316 KB
80 tracks one-sided =	79 * 16 = 1264 sectors * 256 = 316 KB
80 tracks double-sided =	159 * 16 = 2544 sectors * 256 = 636 KB
86 tracks (maximum count) =	171 * 16 = 2736 sectors * 256 = 684 KB

This means that 1 track is placed 4 KB, or 4 sectors per 1 KB. In this way, dividing the number of free sectors by 4, you can find out how many kilobytes of free space remains on the disk (recall, 1KB = 1024 bytes).

TR-DOS uses a double recording density, so we get 80 tracks and 16 sectors of 256 bytes per track.

STRUCTURE OF TR-DOS DISCS

All the service information about the floppy disk is located on the zero track. The first eight sectors (No. 0 ... 7) contain a directory of files, the next sector is No. 8 - a system file, in it its parameters are recorded. From track number 1, the main disk space begins, where The files are actually stored. As you can see, the organization of the disk is very simple, in comparison with MS-DOS, so there are difficulties with deleting files, you need to consolidate the disk the MOVE command.

Zero track sectors numbered 9 to 15 are not normally used and with the help of The DOS procedures discussed below can accommodate any of their own information. There is, for example, the original "boot", occupying the main area disk only one sector, which, when written, places its codes in free sectors of zero track (author Yu Vlasov). However, using these sectors, do not write to such a disk with the help of "Magic" - this procedure will spoil sectors with numbers 10, 11 about "Magic" see below).

STRUCTURE OF THE FILE HEADING IN THE CATALOG

```
0 ...... 7 8 9 BDEF
XXXXXXXX X XX XX XXX
│ │ │ │ │ │ └── 1 byte - the number of the initial track
│ │ │ │ │ └──── 1 byte - the number of the initial sector
│ │ │ │ 1 byte - the number of sectors
2 bytes - the length of the file for CODE, and the software part for
│ │ │                                              BASIC, byte
2 bytes - the initial address for CODE, the total length of the file for
│ │                                                BASIC
1 byte - file type (B, C, D, #)
```

8 bytes - the name of the file

The procedure for deleting a file is to replace the first byte of its name with the code 01, after The file is considered deleted and does not appear in the directory. All information on a floppy disk is stored until the MOVE command is executed. If a file is deleted at the end of the directory, The first byte of the name can get a zero value (a sign of free space).

For files of type "DATA", the Length field also represents the length of the array (variable), and the address field (Start) -the initial address of this array at the time of recording, taking into account the 257-byte offset by the size of the clipboard (see below). For disk files (#), the length field contains the length of the filled part of the block, from 0 to 4096 (I remind you that these files are written blocks of 4096 bytes, i.e. 16 sectors, under the same name). The high byte of the address field for the disk file is 32, and the younger file is the block number in the file (0, 1, 2, ...), which in the directory is displayed as 8192, 8193, 8194 ...

On the diskette, the file headers are written in the first 8 sectors (with numbers from 0 to 7) zero track. In addition, it should be noted that the sequence number of the file header in the directory has the following property: the high-order half of the byte corresponds to the sector number in which the headline, and the youngest - the headline number in the sector. This is useful when working with a disk.

Page 7

As you can see, the line number of the BASIC auto-start is not included in the header data. is he is written along with the program, 5 bytes after the end of the program or variable domain, if they are in the ELINE field: after the field separator # 80, the byte #AA and 2 bytes of the line number (junior, senior). Unlike the cassette, autostart from the zero line is not executed.

The "number of sectors" field is used when copying a file, and the "file length" field is used when its loading. By zeroing the number of sectors, you can make the file uncopable (although this protection is also easy and removable).

The diskette directory holds a maximum of 128 files, counting and erased; and since subdirectories in TR-DOS are not created, then for a large number of short files the directory is overflowed, and part space on the diskette remains unused.

STRUCTURE OF THE SYSTEM SECTOR

The system information about the diskette contains the 8th sector of the zero track, more precisely, the end This sector, starting with the 225th byte:

NAME	ADDRESS		APPOINTMENT
BUFF_ADR	0	# 00 # 5025	The beginning of the system sector loaded dynamically the allocated buffer for disk operations.
DCU_SEC	223 224	#DF	Word. Contains the number of sectors on the disk, formatted with DCU (at the beginning is equal to the value in # E5). On the disk the formatted TR-DOS is 0.
FR_S_NEXT	225	# E1 # 5E06	Byte. The number of the next free sector. Initialized to zero.
FR_T_NEXT	226	# E2 # 5E07	Byte. Number of the next free track. Initialized by one.
TYPE DISC	227	# E3 # 5E08	Byte. Floppy type: # 16 - 80 track, 2 side # 17 - 40 track, 2 side # 18 - 80 track, 1 side # 19 - 40 track, 1 side
N_FILES	228	# E4 # 5E09	Byte. Number of files on the disk, including deleted ones. Initialized to zero.
N_FRE_SEC	229 230	# E5 # 5E0A	The Word. Number of free sectors. It is initialized depending on the type of disk: # 09F0 - 80 track, 2 side # 04F0 - 40 track, 2 side # 04F0 - 80 track, 1 side # 0270 - 40 track, 1 side When formatting the DCU program to the maximum number of tracks: # 0A70 - 84 track, 2 side # 0AB0 - 86 track, 2 side
MAIN_BYTE	231	# E7 # 5E0C	Byte. Number of sectors per track. Should be # 10, otherwise the disk is recognized as unformatted.
ZERO	232	# E8	Two bytes of zeros.
BLANK9	234	#EA	9 bytes of spaces (code # 20).
ZERO	243	# F3	1 zero byte.

N_DEL_FIL	244 # F4 # 5E19	Byte. The number of deleted files. Initialized zero.
DISC_TITL	245 # F5 # 5E1A	A field of 8 bytes. The name of the disk after formatting.
ZERO	253 # FD	3 bytes of zeros. End of the main sector.

* NOTE: the address is given from the beginning of the sector in decimal and HEX form, as well as in the temporary DOS buffer, starting at 23845 (# 5D25), where the system sector is read, that

Page 8

it is necessary in the analysis of ROM. For a hexadecimal representation, instead of standard recording, say 5D25h, uses a more familiar users SPECTRUMa - # 5D25.

NOTES ON OTHER TR-DOS VERSIONS

Version 5.03 TR-DOS is faster. This is especially noticeable when performing LIST, MOVE, COPY B. However, to improve performance, we had to abandon testing of drives, in connection with which some old types of drives with low speed moving head with this version will not work. At the command level, the differences between there are no versions, however, the structure of ROM versions 5.01 and 5.03 is very different; 5.04 at addresses more is similar to 5.03. Further, in general, we are talking about version 5.01, which is used by the author; some information about 5.03 is given in the literature (for example, the collection "ZX-Revue" No. 1 for 93 g). I would like to emphasize that the author is not an expert on all versions of TR-DOS. Basically The information given is reliable for other versions, although some differences are also possible.

TR-DOS INPUT POINTS

The way of working with standard TR-DOS commands is given in detail in its proprietary instructions. When programming on BASIC, they are easy and convenient to use. However, when more complex work with SPECTRUM there is a need to contact the operating room system, bypassing BASIC and the TR-DOS interpreter, for example, from a program written in machine codes. The method proposed in the company manual for TR-DOS (imitation of the line BASIC in the codes) is cumbersome and inconvenient. It is much better to apply directly to operating system, located in the "shadow" ROM. For this, TR-DOS exists special opportunities, and on some of them I will stop.

Enabling DOS occurs when moving to one of the addresses in the range 15616-15871 (3D00-3DFF), i.e. there, where in the main ROM there is a table of symbols and, therefore, in normal operation, control is never transmitted there. The scheme controller BETA is made so that when a command is selected in this range of addresses, A trigger that replaces the main ROM with a DOS line, simultaneously becomes available ports of the controller (see attachment). Reverse switching occurs when switching to any address in RAM, so access to DOS is very limited - you can not directly from your programs access neither to its ports nor to ROM, except for the specified range of addresses. However, in this range are all the input points for calling most of the necessary DOS functions.

In fact, there are 23 entry points, but only some of them are applied, because not they are all known. The main input points start with the NOP (00) command, apparently for compensation delay when switching ROM (and possible to run at the following address):

DEC HEX
15616 3D00 - input to DOS from BASIC.
15619 3D03 - call the DOS command from BASIC.
15622 3D06 - communication channel with a disk file - input program.
15629 3D0D - communication channel with a disk file - output program (in the description of the channel 3D0E is used).
15635 3D13 - execution of the command specified in the processor's "C" register (see below).
15652 3D24 - associated with the initialization of the DOS system area - if called the flag "C" is included, 112 bytes are allocated for the DOS system area, and The HL register is set to the beginning of this area (5CC2). When address # 3021 is pre-checked for availability system area.
15663 3D2F - go to any DOS address.

Using the last of these points, you can get full access to TR-DOS ROM. At 15663 (# 3D2F) followed by NOP command should only team RET. She is It allows you to jump to any DOS address, first putting on top of the stack, for example, as follows:

```
            LD       HL, ADR_DOS   ; address in the DOS ROM
            CALL DOS               ; to get on the stack the return address
            ...      ...           ; the continuation of the program
DOS         PUSH HL                ; on the stack address DOS routines
            JP       # 3D2F        ; subroutine call via the entry point
```

This method is good in the case where there is disassembled text TR- program DOS with base addresses corresponding routines. inquisitive readers can be recommended to get yourself a text and understand it (needed To do this, the source code can be written: SAVE "DOS" CODE 0,16383). However, it should to warn that the DOS interpreter written extremely difficult and unintelligible. Process analysis of the disassembled text is very similar to the solution of a crossword puzzle, only to benefit from this activity is somewhat higher. To facilitate the task of decoding the codes mention areas which occupy some of the procedures and the data area:

v5.01	v5.03	PROCEDURES AND DATA
0000-0007	0000-0007	cold start; full system initialization
0008-000F		RST8, initialization, in DE specified memory vertex
0010-0012		RST10, output symbol from the register A, both SOS
0018-001A		RST18, output character string starting from (HL); the last character must be # 00 or> # 7F
0020-0022		RST20, call the program from the main ROM; Address given by two bytes following the RST20
0028-002A		RST28, returns to the current channel address HL given shear set in the register C
0066-0068	0066-0069	jump to the address of the NMI processing "Magic" button
0069-		copies the standard letters A ... U to UDG sets the default values of the system variables
016C-01D2		call DOS from BASICa (through # 3D03, ie 15619)
01C7-		LD A, (HL), and sets the flag Z, if A = # 0D or # 80
01D3-	01D3-	output from a DOS command after; letters # 0D, # 0F, # 10, # 11, # 3D13 subroutine (reserve)
0239-		entrance to the DOS of BASICa (through # 3D00, ie. 15616)
0355-03A0	0360-03AB	dubbed version of TR-DOS
03FB-	0405-	execution of letters # 18 # 3D13 routines
0429-		CAT team
0530-		NEW team
0686-		COPY command
077D-		ERASE command
07F6-0FFF	0800-0FFF	free memory cell (approximately 2 KB #FF)
1000-		command "*"
1003-1017		text "Connected Interface 1"
108D-1138	10A5-1150	system information text to LIST
115D-11A7		printing LIST (register HL) in decimal form
11B6-		LIST command
1348-		COPY s team
1514-		COPY B team
1644-	165C-	execution of letters routines # 08 # 3D13
164C-	1664-	execution of letters routines # 09 # 3D13
1693-		MOVE command
17DD-17DE		Shipment - LDIR, RET (must be set HL, BC, DE)
17E0-		VERIFY command
17E5-		LOAD command
1971-		PEEK team
1975-		POKE command
197D-		MERGE command

1A9C-		SAVE command
1CBC-	1CF0-	execution of letters # 0A routines # 3D13
1CC7-		RETURN command
1D19-		run command
1E01-	1E3D-	execution of letters routines # 05 # 3D13
1E11-	1E4D-	execution of letters routines # 06 # 3D13
1E86-		FORMAT command

1F7D-1F8D	1FB9-1FC9	table
1FAF-	1FEB-	execution of letters # 16 # 3D13 routines
1FBA-	1FF6-	execution of letters # 17 # 3D13 routines
1FC1-		track format
2051-		track inspection
208A-		Track formatting, N-1 in the register E
213C-		OPEN # command
260C-		CLOSE command
26EF-	2739-	execution of letters # 15 # 3D13 routines
271C-27F1	2766-283B	TR-DOS text messages, starting with the "OK"
27F2-	283C-	the transition from the address # 3D13, definition of the command letters
2842-288D	288C-28D7	table subroutine addresses addressable at register C circulation through # 3D13
288E-	28D8-	execution of letters routines # 07 # 3D13
2896-	28E0-	execution of letters # 13 # 3D13 routines
2899-	28E3-	execution of letters # 14 # 3D13 routines
28E5-		when A = 0 - 16 header bytes transfer from HL address # 5CDD, when A <> 0 - from # 5CDD at HL
28A8-	28F2-	execution of letters # 0C routines # 3D13
28B1-	28FB-	execution of letters # 0B routines # 3D13
28C5-	290F-	execution of letters # 0E routines # 3D13
28DC-	2926-	execution of letters # 12 # 3D13 routines
294D-		team "40"
2964-		"80" team
2968-29EA	29B1-2A34	message texts TR-DOS, starting with "* ERROR *"
2A09-2A0B		output to a port - OUT (C), A and RET
2A0C-	2A56-2A06	NMI interrupt request service from the "Magic" key
2D57-		GOTO command
2F79-2F7B		recording command from the register A into VG93
2FA9-2FE7		table
30B3-317B	30FD-31F2	text keywords TR-DOS, starting with the "SAVE"
317E-	2FF3-3031	table
31B3-3C00		free memory location (2637 bytes #FF)
3D00-3D2F	3D00-3D2F	area standard of entries in the TR-DOS
3D8E-	3D98-	execution of letters # 00 # 3D13 routines
3DC1-	3DCB-	execution of letters routines # 01 # 3D13
3E3A-	3E44-	VG93 command execution from the register B, A data register
3E59-	3E63-	execution of letters routines # 02 # 3D13
3EF8-	3F02-	execution of letters routines # 03 # 3D13
3EFC-	3F06-	execution of letters routines # 04 # 3D13
3F09-		reading track (from all service information)
3FB0-		block write procedure - OUTI and RET
3FE6-3FFF	31FD-3C00	free cells

Page 11

Please note that firmware version 5.01 contains more than 4KB of free space, opening up broad prospects for its improvement, is still poorly mastered.

It is easy to notice that the creators of TR-DOS does not adhere to the concept of Sir Sinclair immutability of the ROM, so the various versions of TR-DOS procedures addresses are different. it's desirable bypassing standard entry points (as is done in most decent systems for example, on the IBM PC). If you do use a direct appeal to the ROM, do not be lazy insert in the program check the version of the DOS, for example, using standard procedures # 3D13 (see below.)

```
          LD      HL, # 0361      ; the address is in the DOS ROM to v5.01
          LD      C, # 13         ; Lit. procedures # 3D13 (see. below)
          CALL # 3D13             ; the transfer of 16 bytes from the address in HL in # 5CDD
          CALL comp               ; byte # 5CDD and comparison procedure 8
                                  ; EXAMPL line if coincide, the flag Z = 0
          JP      NZ, EXIT        ; not the same - not version 5.01 - exit
                                  ; or checking to a different version
          ...     ...             ; continued - you can work with DOS ROM
EXAMPL DEFM 'Ver 5.01'            ; a sequence of bytes characteristic
                                  ; for this version (any known)
```

If you do not want to write a comparison routine "comp", 5.01 and 5.03

It can be distinguished in a simpler way:

```
        LD      HL, # 026A      ; the address is in the DOS ROM
        CALL # 01C7             ; A return register byte (HL)
        CP      #AA             ; A byte in register #AA - 5.01
        JP      Z, V501
        CP      # F4            ; A register byte # F4 - version 5.03
        JP      Z, V503
        JP      EXIT            ; not #AA and # F4 - unknown version, output
```

However, it is highly likely that this program with other versions of DOS will get stuck in # 01C7, without reaching "EXIT", so the first test is more versatile.

Unfortunately, in recent years began to appear the program, whose creators We dealt with their version of DOS, and believe that it is the only one in the world. Launch of programs on another computer, as a rule, leads to damage to the disc.

The most important to call TR-DOS from the program in machine code is the point entry with the address 15635 (# 3D13). Developers put on this procedure functions that serious machines (e.g., IBM PC) operate vectors input-system interrupts output. With the address # 3D13 eventually proceeds to the subroutine, which in Depending on the code contained in the register C, by using the table # 2842 [10306 dec] (# 288C [10380] in v5.03), transfers control to the appropriate subroutine. The parameters for these subprograms are transmitted by microprocessor registers or are in system variables TR-DOS (for example, the file header). Below are brief descriptions procedures called depending on the letters (C register contents):

C = # 00 - the command "Restore" VG93: head assigned to the zero track and waits INTRQ signal. Waiting can be interrupted by pressing the key BREAK.

C = # 01 - drive initialization set in register A. If not specified number initialized corresponding variable 23802 ... 23805 (= #FF), it is set equal to # 08, then determines the number of tracks and is recorded in the 23752 ... 23755 (# 00-to 40 tracks, # 80 to 80). Then, a predetermined number of register A recorded in 23798 (temporary disc drive), and 23830 is stored copy of the system register.

C = # 02 - installation of a head to a track number which is set in register A (0 ... 159, i.e., 1st track and sides 0 and 1 correspond to the track numbers 0 and 1).

C = # 03 - Installation of the sector number, the number of which is set in the register A (1 ... 16) - number placed in the # 5CFF (23807).

C = # 04 - setting the buffer address set in the register HL, a system variable # 5D00 (23808).

Page 12

C = # 05 - reading the block of sectors: the buffer is read by HL at B sector, the first of which has the track number E D (if B = 0, read only headlines sectors).

C = # 06 - recording sector unit (options are similar to the previous command) from the buffer HL address on the disk (track D, E sector) B is written sectors.

C = # 07 - the disc directory output to the channel specified in the register A (2-screen printer 3). room the disc is taken from the variable # 5CF6 (23798). It is necessary to duplicate this number in cells 23800, 23801. Pre-performed procedure # 18.

C = # 08 - read the file header in the area of system variables at # 5CDD- # 5CEC. Number of the desired file (from 0 to 127, including remote) is given in register A.

C = # 09 - file header entry from the system area on the disk, the file's location with a number specified in the register A (rename).

C = # 0A - search for files in the named directory and type are specified in the system Variables # 5CDD- # 5CE5; the number of bytes is indicated in variable 23814 (usually equal to 9). If the file is found, its number is returned to the register C and 23838 cells (# 5D1E), 23823 (# 5D0F); if not - in the register C contains #FF, 23838 does not change, in 23823 - the code #FF errors.

C = # 0B - write to the disk CODE file type from the address length HL DE; name (and type?) File It must be in the system # 5CDD- # 5CE4. Checking for the presence of such name is not made. procedure # 18 preliminarily provided.

C = # 0C - write to the disk BASIC-program; file name is set in the system. name check is also not performed. The variable # 5CD1 specified number autostart line (low and high byte). Sometimes, common observation that if the type is not equal to "B", the file is written as "boot"; perhaps for version 5.03? procedure # 18 preliminarily provided.

C = # 0D - out of the TR-DOS (reserve for later expansion).

C = # 0E - downloading files or VERIFY; in this type of header and should be placed in Region # 5CDD- # 5CE5. When the system variable 23801 (# 5CF9) equal to # 00 Loading at #FF - check. To download BASIC - program variable 23824 (# 5D10) must be equal to # 00; if it is equal to # FF, then the downloading the new, the old program is deleted, but the new can not be read. AT Depending on the contents of register A subroutine will operate differently:

A = # 00 - the load address and length is taken from the catalog;

A = # 03 - codes are loaded with HL addresses, length of the uploaded file is determined value DE;

A = # FF - codes are loaded with HL addresses, but the length is the length of the uploaded file recorded on the disc.

C = # 0F ... 11 - out of TR-DOS (a common address for the procedures # 0D, # 0F, # 10, # 11) (in versions 5.01 and 5.03 Reserve for expansion).

C = # 12 - deleted the file name and type that is specified in the # 5CDD- # 5CE5, and removed all files with these parameters. In the # 5D08 saved the first byte of the name, which He was replaced by # 01; in the # 5D07 - number of deleted files.

C = # 13 - information about the file transfer (16 bytes) to the address in HL, in # 5CDD- # 5CEC.

C = # 14 - the transfer of information about a file # 5CDD- # 5CEC address in HL.

C = # 15 - checking the track number which is set in the register D. If found defective sector, will be placed # 5D0F error code equal to # 07, and in the system variable # 5CD6 - the number of defective sectors.

C = # 16 - sets the current drive direction 0 (top), e.g., for a previous procedure. Changes system register #FF (the value of the masked # 5D16 OR # 3C, i.e. 00111100). In other words, the system register load value equal to A OR # 3C.

Page 13

C = # 17 - Sets the current side of the disc 1 (bottom) (the value of the # 5D16 masked AND # 6F). In other words, the system register load value equal to A AND # 6F.

C = # 18 - reads the system controller generates sector and setting the type of diskette.

Thus, through # 3D13 You can perform almost all the necessary disk operations, which may require the usual application. only when writing (eg formatting) Special programs require direct disk appeal to the DOS ROM, but it is impractical to do unless absolutely necessary, as compatibility problem arise.

As an example, a frequent downloader program for monobloc (All parts of the program are written sequentially and combined with BASICom):

```
        LD      HL, 40000       ; load address of the 1st block
        LD      B, 17           ; number of loaded sectors for him
        CALL DOS                ; loading procedure call
        CALL 40000              ; e.g., decompression screen
        LD      HL, 24500       ; download address 2nd (main) unit
        LD      B, 115          ; number of loaded sectors
        CALL DOS                ;
        CALL 24500              ; for example, the decoding program
        JP      START           ; run the downloaded program
DOS     LD      DE, (# 5CF4)    ; address on the disk of the system variable
        LD      C, 5            ; boot sector unit
        JP      # 3D13          ; DOS function call
```

The boot loader uses the letter '# 05 # 3D13 procedures, ie ship the number of sectors specified in register B, to the address specified in HL (for different programs, of course, they will be different). To determine the address on the disk, use the system variable # 5CF4, where after loading each block (including initial BASIC-part) is the address of the next sector of the track. Thus, it is possible to load any number of consecutive blocks, wherein the loading It is much faster, because the drive head does not go to zero for track the parameters of each file; accordingly, the wear decreases diskette. As a rule, units written immediately after the BASIC-loader, and then use any "Disk Doctor" correct number of sectors of the boot loader on the total, as the former blocks are marked as deleted. Besides other advantages, the program-piece harder to crack.

Note that if the block length is not a multiple of the length of the sector (256 bytes), with the latter sector, some "junk" will be loaded after your code block. Therefore, when you want download a predetermined number of bytes, you must use the letter # 0E procedures # 3D13:

```
        LD      HL, NAME        ; the name of the 1st block
        LD      C, # 13         ; carrying the name of (HL) in the DOS variable
        CALL # 3D13             ; DOS function call
        XOR A                   ; address and block length - from the catalog
```

```
          LD      (# 5CF9), A       ; for LOAD
          LD      C, # 0E           ; download the file by its name
          CALL # 3D13               ;
          LD      HL, # 5CE4        ; the last byte of the file name
          INC (HL)                  ; the next unit number
          LD      HL, ADR2          ; work address 2nd block
          XOR A                     ;
          LD      (# 5CF9), A       ;
          LD      A, # FF           ; Block address - in HL, length -
          LD      C, # 0E           ; from the directory
          CALL # 3D13               ;
          JP      START             ; run the downloaded program
NAME      DEFM "block_01C"; name and type of the uploaded file
```

In these examples it is assumed that the downloaded files on the disk, of course, present. This method can be used to download programs only. If it is required loading of the application program when the file name specified by the user, must be carried out check for the requested file. It is also desirable to configure the type of disc (As the floppy disk can be replaced after the start of the program). This can be accomplished, for example, as follows:

```
          LD      C, # 18           ; setting the type of floppy disk
          CALL # 3D13               ; DOS function call
```

Page 14

```
          LD      HL, NAME          ; the name of the requested file
          LD      C, # 13           ; carrying the name of (HL) in DOS
          CALL # 3D13
          LD      C, # 0A           ; File search by name
          CALL # 3D13
          LD      A, C              ; in the "C" Stock No file in the directory
          INC C
          JR      Z, EXIT           ; if C = # FF, the file can not be found - exit
          LD      C, # 08           ; Reading parameter file in DOS peremenennoy
          CALL # 3D13
          XOR A
          LD      (# 5CF9), A       ; for LOAD
          LD      HL, ADR           ; block load address
          LD      A, # FF           ; block address in HL, length - from the catalog
          LD      C, # 0E           ; download the file by its name
          CALL # 3D13
          RET
NAME      DEFM "textnameC"; name and type of the uploaded file
```

MEMORY CARD TR-DOS

At initialization TR-DOS 112 bytes allocated to it, the memory area respectively, since CHANS, moves up:

Standard start BASIC-area: 23755 # 5CCB
Start the BASIC-region with TR-DOS: 23867 # 5D3B
Start the BASIC-area with TR-DOS interface and 1: 23925 # 5D75

In addition, during a read or write sectors, between DOS and area variables CHANS sector creates a buffer - 257 bytes (256 bytes of data + control byte). Wherein a memory area starting from # 5D25 (CHANS-1), shifted up by another 257 bytes thus that the BASIC-area starts with # 5E3C. If RAMTOP set too low, then the call DOS commands can be issued a message "Out of memory", while the rest BASIC works fine; it says not enough memory to allocate 257-byte buffer. allocation buffer must also be considered when reading code to BASIC-program: for example, to load into the zero line after REM, not necessary to specify the address PROG + 5, and PROG + 5 + 257, i.e. # 5E41.

When you transfer the cluster software on a disc, the codes at the zero line are shifted 112 bytes, which requires a corresponding address correction. Simply clean the area under DOS using the procedure of the main ROM:

```
          LD      BC, 112           ; shift amount
          LD      HL, 23755         ; address = standard top BASIC-region
          CALL # 19E8               ; removal unit, ranging from HL, length BC
```

This procedure shifts all BASIC-region, the system correcting values variables.

It should be noted that the 0-th row of the codes can not correctly handle DOS, do not it is recommended to give the command CAT, LIST.

This method of allocation of buffer before CHANS area explains the strange TR-DOS behavior when accessing MS-DOS floppy disk: any disk access begins with Reading 8th sector into a temporary buffer, as well as the MS-DOS sector is twice as long as the table CHANS is jammed - with consequences ...

Address table channel information indicates a system variable CHANS (for at 23631, # 5C4F); BASIC-start program defined variable PROG (cm. 23635, # 5C53).

Page 15

Some system variables TR-DOS v5.01

The brackets [] are values upon initialization, respectively, if the DOS started without disc in the drive / disk / in the course of work.

DEC HEX VALUES

23734 # 5CB6 15 bytes - standard domain CHANS K, S, R channel: 2 bytes - the address inference procedures of the system, the input into the system and channel byte name (R, S, R). AT Unlike standard, modified address in channel R: (23746) = # C9, so that error processing channel address R = # 15C9, instead # 15C4 (analogously to region DOS CHANS - 23858) .Zamena values (# 5CC2) to # C9 (RET) to # C3 (JP?), [2 byte address] leads to the Autostart (at) CODE file after download. When connected INTRFACE1 23734 bytes = 00 #

23734 # 5CB6 [2548,4264 (# 09F4, # 10A8)] K

23739 # 5CBB [2548,5572 (# 09F4, # 15C4)] S

23744 # 5CC0 [3969,5577 (# 0F81, # 15C9)] R

23746 # 5CC2 Byte = # C9.Ispolzuetsya for calling routines from DOS

23752 # 5CC8 bytes - disk mode "A" [# FF / # ??]

23753 # 5CC9 bytes - disk mode "B" [# FF / # ??]

23754 # 5CCA bytes - drive mode "C" [# FF / # ??]

23755 # 5CCB bytes - drive mode "D" [# FF / # ??]
The cells are initialized only in DOS 5.01 [#FF, if the drive does not initialized; ? ?? differently% 00000:
Bit 7 = 1 to DD, = 0 for SD;
bit 1 = 1 for DS, = 0 for the SS;
bit 0 = 0 if DD is used in SD mode]

# 5CCC 23756 bytes - The sector number output CAT team [# 0D]

23757 # 5CCD byte - drive readiness (VG93 status register) [# 80 / # 00]

23758 # 5CCE byte - # 00 read operation, #FF entry operator [# 00]

23759 # 5CCF 2 bytes - WORKSP address when the MOVE, COPY, LIST

23761 # 5CD1 2 bytes - the auto-line number with SAVE BASIC

23764 # 5CD4 bytes - the number of deleted files when the MOVE

23765 # 5CD5 bytes - the number of sectors erased file with MOVE

23766 # 5CD6 bytes - the track number of the erased file with MOVE. to format reset, the number of defective sectors will be at the end (similar to when checking track procedure # 15 Team # 3D13). A method for loading letters # 0E (register A). Raven #FF, if the command is not performed

23767 # 5CD7 2 bytes - address. After initialization, the counter indicates [23771] After write CODE - at the start address in the memory. When called from BASICa comprises the row address from which the call originated. In addition, after checking the type # 5CD7 drive contains a number of tracks (?). By doing format # 5CD7 contains the number to be formatted tracks and # 5CD8- VERIFY flag: at # 5CD8, not equal to 0, the treadmill is not formatted verified

23769 # 5CD9 2 bytes - the address of the symbol for the interpretation of similar CH_ADD. After Initialization indicates [23869]. After recording, the file contains a CODE the length of the recorded file. If you format the variable # 5CDA = # 80, if double-sided floppy disk (otherwise formatted only SIDE1). For letters # 0E stores the file download address

23771 # 5CDB 2 bytes - download counter; after the file operation includes a length loaded / written part of the file (?). With MOVE current comprises the sector number and the track. When LIST # 5CDB byte contains the current number O channel

23773 # 5CDD 16 bytes of the file header; in the same form as in the directory

Page 16

23773 # 5CDD 8 bytes - name of the file
23781 # 5CE5 1 byte - type (B, C, D, #)
23782 # 5CE6 2 bytes - address START to CODE or full length for BASIC
23784 # 5CE8 2 bytes - length of the file or software for BASIC
23786 # 5CEA 1 byte - file size in sectors
23787 # 5CEB 1 byte - number of primary sector file (# 00 ... # 0F)
23788 # 5CEC 1 byte - number of initial file of the track (# 00 ... # 9F)
23789 # 5CED 7 bytes - file settings when copying from the drive to the drive
23789 # 5CED 2 bytes - address START to CODE or full length for BASIC

23791 # 5CEF 2 bytes - length of the file or software for BASIC; in the presence of INTERFACE 1 byte = 01 #
23793 # 5CF1 1 byte - file size in sectors
23794 # 5CF2 1 byte - number of primary sector file (# 00 ... # 0F)
23795 # 5CF3 1 byte - number of initial file of the track (# 00 ... # 9F)
23796 # 5CF4 bytes - the counter loaded sectors (sector number processed controller). After the transactions file contains the following number sector, which is often used when creating downloaders
23797 # 5CF5 byte - Track loadable counter; after operations with files contain the number of the next track
23798 # 5CF6 bytes - the drive time for the operation (0-3) [# 00] (i.e., when a name is given, for example, "B: name"). Addressing DOS uses double-byte numbers so the bytes # 5CF7 = # 00
23799 # 5CF7 bytes - when returning from DOS (15616) is cleared
23800 # 5CF8 bytes - the drive source in step with two files (# 00- # 03) [# 00]; # 5CF9 = # FF, if the channel is open (?)
23801 # 5CF9 byte - drive receiver in step with two files (# 00- # 03) [# 00]; a type operations: LOAD = # 00, VERIFY = # FF [# 00] (if instead of loading commands #FF comparison is performed)
23802 # 5CFA bytes - the movement of the drive head "A"
23803 # 5CFB bytes - the movement of the drive head "B"
23804 # 5CFC bytes - the movement of the head drive "C"
23805 # 5CFD bytes - the movement of the head drive "D" [#FF, if the drive is not initialized; otherwise # 08- # 0B; # 08 - maximum speed]. Used only in DOS 5.01, in other versions, the speed is not changing (only fast), but the value of # 08 is used as an indication of drive initialization
23806 # 5CFE bytes - VG93 controller command given by DOS (see # 3F18.) .For teams Reader # 5CFE = # 80 (% 1000 0000) for write commands # 5CFE = # A0 (% 1010 0000) [# 00]
23807 # 5CFF bytes - the number of sectors + 1 VG93 given by DOS (see # 3F0D.)
23808 # 5D00 2 bytes - DOS buffer address where the goods sector. Indicates address DOS-CHANS; this area of the channels along with the program at startup temporarily shifted by 257 memory cells. When you open a disk file indicates the beginning of BASIC-field (# 5E5F) [# 00 / # 00 / # 5D25]
23810 # 5D02 2 bytes - the temporarily stored register HL. After initialization points for (E_LINE + 1?) [# 5D3D] (saved register HL?)
23812 # 5D04 2 bytes - the temporarily stored register DE [# 317C]
23814 # 5D06 bytes - the number of bytes, the manager searches the file name in the directory (8-name + 1 type) [# 09]
23815 # 5D07 bytes - the result MOVE: # 00 - no files, <> # 00 - OK; remote meter files in the process # 12, the subroutine # 3D13
23816 # 5D08 bytes - saved the 1st letter of the file name when MOVE (ERASE and in procedure

Page 17

12 # 3D13 routines)
23820 # 5D0C byte - a sign of isolation input / output buffer: # 00 - selected, #FF - no [#FF] (see. # 5D00)
23821 # 5D0D bytes - file number when copying from two drives
23822 # 5D0E byte - at #FE DOS message is not displayed, at # FF - sign work BASICa, while another - TR-DOSa

23823 # 5D0F byte - error code TR-DOS when called from BASICa (supra.), as well as the number file in the directory when searching for the letter # 0A # ZD13 procedure [# 00]

23824 # 5D10 high byte errors in the call 15616 or procedures (e.g., letters Reading File # 0E procedures # 3D13); you must first record 00 # [# 00 / # FF]

23825 # 5D11 2 bytes - command-line address TR-DOS: when calling from BASICa indicates CH_ADD, when operating in 15616 - a buffer E_LINE editor [# 5D3C]; Address command line TR-DOS: when called from DOS = E-LINE; from BASIC = CH-ADD

23827 # 5D13 2 bytes - address indicates the highest address of the stack containing the address error handling procedures ERR_SP [# FF54]; copy ERR-SR; if a senior bytes = #AA, performed RUN "boot" command, and in the # 5D19 code #FE

23829 # 5D15 bytes - DOS mode: at # FF - execution of DOS commands from BASICa, at # 00 - work in DOS

23830 # 5D16 bytes - a copy of the system register (operation controller, typically room Drive + # 3C / # 2C cm. Further description of the port #FF)

23831 # 5D17 byte - a sign of initialization DOS: if = # 00, at the entrance to the DOS output saver and executed RUN "boot" command to be = #AA - not (re Input) [# 00 / # AA]; at = #FF, does not give an error when reading the wrong address

23832 # 5D18 used if connected INTERFACE 1, while # 5D18 = # FF [# 00]

23833 # 5D19 bytes - the default drive number (# 00 ... # 03) [# 00]

23834 # 5D1A 2 bytes - the subroutine return address of TR-DOS [# 02C0] (also found # 0201?)

23836 # 5D1C 2 bytes - the stored current value of the stack SP [# FF3E]

23838 # 5D1E bytes - the number of the file in the directory, if it is found [# 00]

23840 # 5D20 last 3 bytes of the initial command line

23843 # 5D23 byte - memory for MOVE (in blocks, min 4KB)

23845 # 5D25 bytes - the beginning of the temporary buffer DOS [#FF]

# 5D26 23846 20 bytes - region CHANS-DOS K, S, R, P channels; similar to the standard of at # 5CB6 (+ printer channel):

23846 # 5D26 [2548,4264 (# 09F4, # 10A8)] K

23851 # 5D2B [2548,5572 (# 09F4, # 15C4)] S

23856 # 5D30 [3969,5577 (# 0F81, # 15C9)] R *

23861 # 5D35 [2548,5572 (# 09F4, # 15C4)] P

23866 # 5D3A byte separator [# 80]

23867 # 5D3B Start the BASIC-program, if not open disk files (see. Below)

* NOTE: The area under CHANS DOS initialization of BASICa (after USR 0) more differs from the DOS initialization reset: the channel address entry procedure R It has the same meaning as without DOS - # 15C4, ie bytes (# 5D32) = # C4, although (# 5CC2) Yet = # C9.

When initializing the standard BASIC-field variables from DOS NSPPC (# 5C44) ... SUBPPC (# 5C47), NXTLIN (# 5C55), MEM (# 5C68), T_ADR (# 5C74) are not initialized (# = 00); PIP (# 5C39) = # 1E, and NMIADR (# 5CB0) = # AA instead of # 00.

It should be noted that many of the addresses used by DOS at different points for different purposes, such as preserving registers et al. described in more detail variables comprising useful information for a programmer.

Page 18

FIELD OF LINKS disk file

If you open a disk file, the next area is allocated in addition:

DEC HEX VALUES

23866 # 5D3A 292 bytes - the channel region due to a disk file:

23866 # 5D3A bayta 2 - write address channel procedure [15630 (# 3D0E)]

23868 # 5D3C 2 bayta - address read from the channel treatments [15622 (# 3D06)]

23870 # 5D3E 1 byte - channel name - "D"

23871 # 5D3F 8 bytes - # F4, # 32, # 33, # 38, # 24, # 01, # 00, # 4C (row Damaged POKE 238 ??, 76 (?))

23879 # 5D47 3 bytes - a file type:
1B, 00, FF - consistent, for recording;
00, 00, 00 - serial, read;
1B, 00, 7F - random access

23882 # 5D4A 16 bytes - file settings from the directory (see # 5CDD: name, type, address. length, number of sectors, location on the disk)

23898 # 5D5A 4 bytes - # 3A, #EC, # 36, # 34 or # 11, # 00, # 00, # 00 (?)
23902 # 5D5E 256 bytes - communication buffer to the file; It contains the current text fragment
24158 # 5E5E byte separator [# 80]
24159 # 5E5F Start the BASIC-program, if one disk file has been opened

Generally speaking, a fairly powerful system disk files (#) slightly used programmers (me, for example, did not come across a single program with its application). AT As an illustration of the opportunities will bring a focus to the disk file, allowing to easily solve the problem of printouts (printers we still weak in common people). So, try to run the following program:

```
10 CLEAR 28000
20 RANDOMIZE USR 15619: REM: OPEN # 4, "printer", W
30 POKE 23580, PEEK 23582: POKE 23581, PEEK 23583
40 RANDOMIZE USR 15619: REM: LOAD "mons3" CODE 40000
50 RANDOMIZE USR 40000
```

Naturally, the floppy must contain downloadable code and sufficient free space for opening the disk file. After starting MONS'a enter the command "T", the question "Printer?" answer "?", and you get on a floppy disk a number of sector-16 blocks "printer #", containing the disassembled text in ASCII format (stress in Unlike the entry in GENS format unsuitable for printing). some inconvenience is the need for post-withdrawal exit the program and close the diskrvy file (CLOSE # 4) to record the rest of the information from the buffer. However, without it you can do if dopechatat at the end of the order of 255 extra bytes.

This method uses a thread switch STRMS field with the channel # 3, the printer ("P") the open channel disk file, in our example # 4. Similarly, one can switch the majority of programs performing printing through the channel # 3. For graphics, Naturally, the program must be configured for the correct printer type. then remains just copy the files to the IBM PC (how to do this, look at later), where they It can be combined with the facilities to edit and print.

Recording with "MAGIC KEY"

With the help of MAGIC buttons almost any program can be recorded on floppy disk at any time, and then run from the same location. No matter how she initially loaded - with magnetic tape or disk drive. On the disc is recorded fully All RAM, and the contents of the CPU registers. Naturally, the disk space when This is used not the most economical way, although many toy with complex loaders or code blocks is longer than 41 KB, the use of completely MAGIC It justified (especially that programs exist to bring MAGIC-files in normal view).

Page 19

MAGIC procedure is based on the NMI, which is mainly ROM is not being used because of an error. The controller BETA pressing MAGIC button switches ROM DOS and generates pulse NMI. Receiving the NMI signal, the processor immediately postponed all other things, puts on the stack the address of the next command (to return), and proceeds to the address # 0066, and then to the subroutine (see. ROM addresses).

To preserve the value registers (AF, BC, DE, HL, IX, IY, BC ', DE', HL ', AF', I, R), their pushed onto the stack. IFF interrupt trigger value is stored together with the register I as a P / V flag. Stack depth is 13 words, and with the work itself recording procedures - 21 word (from SP values at the time of pressing MAGIC). Recording starts by saving the first 512 bytes the display area in 2 sectors zero track (# 0A, # 0B). Then, in this area of the screen system sector is read and floppy organized workspace DOS (stack). Further marks the beginning of a free area on the disk and there is a record of all the RAM, since screen; free space on the disc is not checked. The catalog file is written to the name "@", type "C"; in the address field indicates the address of the stack (SP value). After recording (if It was successful, and if DOS mode has been activated), the program status is restored just like when you start MAGIC-file. MAGIC procedure writes the last cell 23551 printer buffer (# 5BFF) - # C9 (RET), and in the system keyboard scan cell 23552 (# 5C00) and 23553 (# 5C01) respectively # 16 and # 10.

The firm instructions indicate that for some programs the file name must begin with a "$". This symbol means that the startup will set the interrupt mode IM2. The program can not independently determine what type of trap was to write, so you need to check whether the program works with a common name, and if not - to put at the beginning of the name "$" symbol.

To record using the MAGIC is desirable to use a blank disc, as in the case of hardware failure or features of the program, the disc may be damaged. Some programs have not a sufficient depth of the stack (42 bytes), and a stack of the recording itself overwrites program. Sometimes successful entry depends on when you press MAGIC, because at work program stack depth varies; so that if it fails should try again. tested stack area or the appearance of byte # C9 at # 5BFF can be used specifically for protection from the use of MAGIC (try, for example, to record a program MAGICom DCU 2.02).

When downloading MAGIC-file on the screen as well as the recording, organized workspace for DOS. file settings are read from the disk directory. The value of the address field (bytes # 09, # 0A) is stored in the register SP. Depending on the name of the first byte is set type Interrupt IM1 or IM2, if the name begins with a "$" (interruption while prohibited). Next, the stack registers are restored in the reverse order of recording (see. above). simplistically recovery procedure is as follows:

```
POP AF
LD      R, A
POP AF                      ; I interrupt vector and interrupt trigger
LD      I, A
EX      AF, AF '            ; IFF (as the value of P / V flag) stored in the AF '
POP AF                      ; Alternative restoring registers
POP HL
POP DE
POP BC
EXX
POP IY                      ; basic restoring registers
POP IX
POP HL
POP DE
POP BC
EX      AF, AF '            ; recovery of P / V flag, comprising
                            ; value IFF
JP      PO, ADR
EI                          ; enable interrupts if the P / V = 1
POP AF
RET                         ; transition to the address that was at the time of NMI
```

Page 20

There are programs that convert MAGIC-file to normal, with the BASIC - loader for the tape or disk. They used to run a similar procedure.

Formatting features TR-DOS

Formatting carried TR-DOS, it is a simplified version described in application of the standard. The main difference is the absence of the index field, ie, Track begins immediately with the first space (12 bytes), and the identifier of the 1st sector. Besides, in the sector identifier is not set bytes of the disk - it is always equal to # 00, and the side It is determined by hardware. Lack index field - the main reason why the disk with SPECTRUMa can not be read on other computers, for example, IBM PC. Existing IBM program to format the floppy disk in the TR-DOS format (for example, HOFORMAT) create index field, which does not affect the operation of the disk on SPECTRUM, but allows you to read and burn them to IBM (using known NOVETA program). on SPECTRUMe determine that the disk is formatted using HOFORMAT, possible by the fact that net sector is not filled with a null byte, and # F6.

The alternation of sectors usually standard (2: 1), i.e. sector location on 1-9-2-10- track ... -16, but you can format the drive using DCU program mode "fast", when the sector should be in order. When the multi-sector operations is accelerating work with the disk, but if the program handles one sector, it may significantly slow.

Formatting features can be used to create protected floppy disk copy. For example, DISTRIBUTOR reformats the 159-th track, creating in the beginning of sector number # 06 at 128 bytes and a head number equal to 1; then the rest sector non-interleaved (1,2,3 ...), for the 16th sector is also set to head number equal to 1. DCU 2.12 program similar reformats the zero track, creating the beginning #AD with sector number and a length of 128 bytes, and the length of the last (16th) and decreases to 128.

Standard way, any track can be formatted using the procedures from the DOS ROM, for example, as follows:

```
XOR A
LD      (# 5CD8), A     ; to validate the track after formatting
LD      (# 5CDA), A     ; for one-sided format
LD      (# 5CD6), A     ; to reset the number of defective sectors
LD      E, NTRK         ; track number 1 (for a zero NTRK = # FF)
LD      B, 1            ; number formatted tracks
LD      HL, # 208A      ; formatting procedures address
CALL DOS
LD      A, (# 5CD6)     ; The register A - the number of defective sectors
```

```
DOS       PUSH HL               ; appeal to the DOS ROM
          JP     # 3D2F
```

This procedure, to selectively format, can be used program, for example, for recovery of defective diskette (of course, before formatting track it should be considered with all the available information, and after - burn it back). Also of interest is the formatting additional tracks outside of the usual 80-minute (generally not format tracks 2-3).

Non-standard format is much more difficult to implement. For this used procedures # 2F79 (issuing commands to VG93) and # 3FB0 (block entry). On the controller gives the command "track record", but will also need more number procedures for checking disk availability, monitoring results and others. All recordable Information for the whole track, including indexes, blanks, field data should be pre-formed in a memory (of the structure of the service information - see Appendix.).

The track can be read using the procedure # 3F09 (also requires a number of preliminary operations). In addition to the data memory, all service information is read, which is described in detail in the annex; method of formatting does not play a role, so you can read not only the TR-DOS, but also any diskettes, including protected against copying. AT

Page 21

Basically, using the command "Read Tracks" and "track record" may create versatile copier.

When reading a track, consider the following:

✓ With the command "read track" BIS VG93 does not check control amounts so floppy that normally can not be read from the first, can be I read correctly;

✓ gaps between the fields are not synchronized, so they can be read each time the po-different, but still the same as the sequence of bytes; the number of bytes in the gaps may also vary slightly;

✓ sync pulse is recorded in a specific way, so the first byte can be read differently (# 14, # C2), but the next two - always # A1, # A1; so easy identifier detected by sequence # A1, # A1, #FE, and data - at # A1, # A1, FBh;

✓ unformatted track is read in a random sequence of bytes, in which there are no labels mentioned above.

DISC PROGRAM

Through the use of the principles set out in the article, more and more appear programs for SPECTRUMa, adapted to work with TR DOS. Here is a list of the most known:

SYSTEM

ARTIST 2 version for DOS - Rodionov. Graphics editor; has more opportunities than STUDIO, but less convenient.

ART STUDIO v1.5 Developer James Hutchby (OCP) '85 Versions for DOS - N.Rodionov; Kordial '91 (with installation). Convenient editing; It allows you to write the file to former place.

BDJTEST (Beta-disc's test) Developer Starocoltsev Eugen, '90 Shows track numbers on the disc that are not read.

BETA COMMANDER v1.4. Developer Trubinov IU '91 BOOT - shell for File:. start-up, copying, and other Well-decorated renaming functions (including the name of drive) and copy, but there is no information about the files.

CC (Conver commander v3.02R). Developer Yu Kovalevsky, '92 wrapper to work with files, has many features, including built-in "disk doctor" Russified, but too big.

COPY Developer Lebedev SM '91 copier disks A <> B and A-> A; copies a block about 41 KB at a time, by placing the program on the screen.

GENS 4.51 Developer HiSoft v4.1 '87, beta-disk version of the MOA 90. In addition to downloading / recording software allows you to read the source code and implement parts of the assembly will directly on the disc!

DCU 2.12 (Disc control utility V2.12). Developer Piter Ltd, Rodionov '91 Formatting (including 86 tracks) and drive control; installation of distribution labels.

DENCOPY Developer Densoft '89 Copy files to tape type "MAGIC", with loader.

IBM Developer Lebedev SM, Krotov VV (for "Spark"). One of (the programs for copy data from floppy disks to IBM TR-DOS; reads the paths; It supports 2D modes, 1D, 2S, 1S.

MASTER-KEY Developer Kotlarov DV (DVK-soft), TRTI v3.2 '90 One of the best

Programs such as "disk doctor" view and edit the data on the disk, has a search function).

MOA v1.6. Developer MOA Leningrad '90 BOOT-shell for working with files: run, copy, rename, view.

MON7 (Advanced monitor v7.05). Wolf Developer. disassembler programs directly on the disc.

MONS 4 beta. Developer HiSoft '87, IC adaptation '90. Provides excellent the possibility of long-disassembling programs directly to disc.

OPENER Developer VAJ, Lviv, 94g. to download a program and view edit protected BASIC-downloaders.

PASCAL'D (Pascal version HP4TM16) HiSoft 84 Developer, at the adaptation Dereschuck.

PCOPIER2 Developer Rodionov '90 copier "disc-tape" with a selection of files.

RAMPRESS Developer Yuri Vlasov, Lviv, 94g. Converts MAGIC - files into the program with loader for tape and disk. Unlike DENCOPY, only writes one block codes, compressing it. This saves at least 20% of disk space, and thus transformed way at the boot program looks quite acceptable.

TLW2 (The Last Word 2) Developer Saga System & Myrmidon 86 Russian version D.G.S., Kordial '90 text editor with a wide range of commands. Communication with DOS through BASICa line.

TREK A> B Developer Kordial '90 handy copier for those who managed to buy two drives.

WHAM (Music Box). The developer disk version MarkTime '85 Rodionov '90 Music editor. In addition to recording / download the source files, records compiled the music on the disc.

ZX-WORD v1.01. Developer Si-soft & Bg-soft '92, Kharkiv. software package for IBM Text file format. Includes editor ZX-WORD, having habitual IBM users looking but too small font (64 characters per line). To tune in DOS version requires no installation. Also includes:

✓ GENSTEXT, TEXTGENS - Decoders from GENS4 format text and back; There are also for ZEUS and others;

✓ SCRTEXT - transcoder SCREEN $ files in a graphical format printer EPSON, allows you to include text printed patterns;

✓ FORMAT - formatted floppy in MS-DOS 3.0 format;

✓ IBM-COPY - file sharing between IBM and ZX. It allows both read and record on diskette IBM.

An indispensable tool for the programmer are the disc version and MONS4 GENS4. Work with them immediately shows all the advantages of disk systems - design process the program is simple and convenient, as opposed to messing around with a tape recorder. With help Paste command (* F, for example, * F 1: text_ass), you can create a new program of the standard, debugged blocks recorded on the disc. At the assembly to the disc Directive (Eg, A16,5000,1: object) it allows you to create large programs. monitor MONS4B carries out disassembly to disk if the team "T" on request "Text:" enter the name in the form [Disk number: nazvanie_fayla]. The file is written in blocks of 16 sectors under the specified name, after which the appended numbers 00, 01 and the like, the program can be disassembled quite long.

GAME

ACADEMY	- Pete Cooke	86,	disk by	andy Chernikov	
BALLGAME	- Pete Cooke	88,	disk by	N.Rodionov	
BRAIN STORM	- Pete Cooke	87,	disk by	N.Rodionov	'90
COLUMNS	- Piter	91	developer	Kraskovsky & Rodionov	
DRAGON BREED	- Activision	90,	disk by	paul Nosikov	'91
DRAGON SPIRIT	- Tengen	89,	disk by	andy Chernikov	'91
EARTHLIGHT	- Pete Cooke	88,	disk by	N.Rodionov	'90
F-16 COMBATPILOT	- Digital int.	91,	disk by	vasilyev Anton	
FIGHTER BOMBER	- Activision		disk by	V.Belyaev	'91
GHOSTBUSTERS II	- Activision	89,	disk by	AVF Moscov	
GOLDEN AXE	- Sega & Virgin	90,	disk by	vasilyev Anton	
HEROES OF LANCE	- US Gold	88,	disk by	Walsoft	'92
INDIANA JONES2	- Tiertex		disk by	vasilyev Anton	'91
KLAX	- Tengen	90,	disk by	JUTrubinov	'91
LEMMINGS	- DMA design	91,	disk by	andy Chernikov	'92
MICRONAUT 1	- Pete Cooke	88,	disk by	N.Rodionov	'90
MYTH	- Systems3		disk by	andy Chernikov	
OPER.THUNDERBOLT	- Ocean & Taito		disk by	vasilyev Anton	'91
R-TYPE	- Irem corp.	87,	disk by	N.Rodionov	'90
ROBOCOP	- Ocean	87,	disk by	AVF Moscov	
SIM CITY	- Probe	89,	disk by	DEN corp.	'93

Page 23

SKATE CRAZY	- Gremlin		disk by	andy Chernikov	'91
SUPER SCRAMBLE	- Gremlin		disk by	vasilyev Anton	'91
WINTER EDITION	- Epyx	88,	disk by	S. Skorobogatov	'90
ZOLYX	- Pete Cooke	87,	disk by	N.Rodionov	'90

Under the disk version games meant not only the programs that are loaded with disc, even if it used a machine-coded downloader. adaptation involves the presence of at least some of the following features:

1. Selection of the game configuration and loading of the respective blocks (e.g., different character or situation);

2. Resuming the next level (quite common);

3. Record a table of records to disk (available in most adaptation programs Rodionova);

4. Edit the disc and maintain their options game situation (eg, BALLGAME);

5. Recording the deferred game to continue in the future (unfortunately, this opportunity, quite common in the cluster versions, disk implemented rarely for example, in ACADEMY and SIM CITY; apparently intended to be used MAGIC).

It is impossible not to note the contribution of Nikolai Rodionov from St. Petersburg - both in quantity and quality programs.

SPECTRUM AND IBM PC

The issue of exchange of information between the IBM and SPECTRUMom of interest to many. first all this, of course, a variety of texts that would like to enter, view, and print on both computers. This problem is solved in two ways: by means of programs, working with MS-DOS format disks on SPECTRUMe (listed above), and programs, running on PC IBM disc formats BETA TR-DOS.

Full-time job with IBM on diskettes SPECTRUMe presents a number of difficulties. It is necessary, in particular, spend a few kilobytes of RAM for the file allocation table (FAT), and to carry out sophisticated manipulation of the cluster numbers in this table. Available this class of programs allow you to format the SPECTRUMe on MS-DOS floppy disk, copy to them files from a floppy and back, as well as to work with the texts in the alternative encoding Cyrillic (most common). Often, however, the copy function carried out by the simplified algorithm, resulting in problems may arise with fragmented files, and different versions of MS-DOS (even the best of programs that IBM-COPY class Si-soft design writes a file so that MS-DOS v5.0 his can be read only after processing "Norton Disk Doctor"). It should be noted that physical level MS-DOS floppy disk read without problems (using the usual function "Read sector" # 3D13 procedures); difficulties arise precisely from the more complex file system.

As practice shows, it is more convenient to use the emulator program TR-DOS on the IBM PC. The distribution is "ZX-SPECTRUM on the IBM PC" includes:

ftst.exe - work with BETA-disk on IBM PC (TYPE, RENAME ...);
hobeta.exe - it works as NORTON between BETA-drives and IBM PC;
hofrmt.exe - BETA-format drives on the IBM PC;
pctobeta.exe - converts files to the IBM PC BETA-format;
spectrum.exe - emulator ZX-SPECTRUM (additional files);
tapemast.exe - BETA-recorded files in the image of the tape (tape_zx.spc) for the emulator;
teledisk.exe - software copy any disc in the IBM PC, Sydex development; It includes several auxiliary files.

Of these, the most interesting programs is program - shell "HOBETA" firm InterCompex. In appearance and functions performed it resembles the famous "Norton Commander", but works with disks TR-DOS. For IBM copy - Files need to be pre-treated with the program "PCTOBETA", which appends them to the standard BETA-header. To copy the floppy is quite convenient "TELEDISK", copying the entire diskette entirely in one file, and vice versa. fanatics SPECTRUMa

Page 24

can even enjoy a familiar picture on the IBM screen using the emulator program "SPECTRUM" (it is difficult to say who may need it, given that it started Only cluster program, but far from all, and copy them only through diskette) ...

A major shortcoming of the package is that no "HOBETA", or "TELEDISK" not read floppies which are formatted in the usual manner on SPECTRUMe, but only by program "HOFORMAT" (the reason for this can be found in the section on formatting). At this time, a program that allows you to format floppy to SPECTRUMe like "HOFORMAT".

APPLICATIONS

This information is needed when analyzing the DOS ROM to deal with appeals ports and work K1818VG93 LSI, and may be useful when setting up the board controller.

Port address disk controller "BETA"

# 1F record - the command register, reading - the status register VG93;
# 3F register VG93 track;
# 5F register VG93 sector;
# 7F VG93 data register;
#FF recording - system controller register (TM9); mode:

```
7 6 5 4 3 2 1 0
- X - X X X X X
  |   | | | └─┴─ drive number - DS (0 ... 3);
  |   | | └───── reset VG93 - CLR (Reset = 0);
  |   | └─────── head loading - HLT (willingness = 1);
  |   └───────── side of the disc - SIDE (SIDE1, bottom = 0);
  └───────────── technique / recording density (MFM = 0 / FM = 1).
```

#FF reading-readiness HS 93:
bit 6 - DRQ (data request = 1);
Bit 7 - INTRQ (the command = 0).
# 7FFD DOS lock on closing:
(#FD) bit 4 = 0 - blocking, bit 4 = 1 - solution;
(NOTE: the address lines decrypted A01 + A15, but because BASICa OUT 253,0 (16) gives the same effect). The SPECTRUM-128 This port is for control memory banks, and the 4th bit - ROM change. SPECTRUM-48 for this port basically useless, because lock just causes the system to hang when referring to DOS. It's easy to adapt for their own purposes, for example, an additional control bank RAM (not forgetting the lock DOS).

Let me remind you that the ports (except for # FD) are only available in DOS mode, ie, for procedures of DOS ROM, which is very difficult to use. For example, the output of the port, whose number register set C, it can be performed using the subroutine at # 2A09 (for TR-DOS v5.01) through the entry point # 3D2F (see. Above).

DESCRIPTION LSI floppy CONTROLLER 1818 SH 93

The chip communicates with a computer floppy: a head unit to a target track, read / write sector or track checksums (CRC), manages rate of movement of the head and pre-compensation record. It supports two formats Recording: single (FM / FM) and double density (MFM / MFM) and various types of floppy disk drive. BIS K1818VG93 WD1793 is an analogue of the company "Western Digital".

SPECIFICATIONS 1818 SH 93 bK0.348.877-08 TU:

Technology	NMOP
Working temperature	+10 ... + 70 ° C
Supply voltage	5; 12 ± 5%

Page 25

Current consumption, respectively	60 mA; 20 mA
Voltage signals: Ui. "1"	> 2.6
Uin. "0"	<0.8 V
Vout. "1"	> 2.8
Vout. "0"	<0.45 V
Maximum load capacity	100 pF
Power consumption, not more than	500 mW
speed of information exchange:	
- when a single recording density	250 kbit / s
- at the double density recording	500 kbit / s
Clock frequency:	
- for a 133 mm floppy	1 MHz
- 208 mm for the floppy drive	2 MHz
Casing (40-pin DIP)	2.123.40-2

Pin assignment MICROCIRCUITS

No. SYMBOL *	**APPOINTMENT**	
1 BS	substrate (bias voltage control)	
2 / W (/ WE, / WR)	input - the information recording permission data bus selected register	
3 / CS	Input - IC	
4 / R (/ MR, / RD)	input - reading permission information from the register to the data bus	
5.6 A0, A1	address bus - register selection:	
A1 A0	register name	appointment
0 0	reading the status register entry instruction register	Current state current order
0 1	register track	the current track number (00-79)
10	register sector	the current sector number (01-16)
eleven	data register	read / write data
7-14 DB0-DB7 (D0-D7)	8-bit bidirectional data bus	
15 STEP	output pulse for moving the head in one step	
16 DIRC	the direction of movement of the head (1-to center)	
17 SL (EARLY)	pre-compensation output control signal (indicating that pulse data WD to be shifted to the left)	
18 SR (LATE)	pre-compensation output control signal (indicating that pulse data WD to be shifted to the right)	
19 / CLR (/ RST, / RES)	input - reset; It is written in the command register 0000 0011% at time Actions output signal INTRQ = 0, then the command is executed "Recovery", regardless of readiness floppy; into register sector is written, "# 01" to the register track - "# 00"	
20 GND	housing	
21 U1	supply voltage of +5 V	
22 / TEST	decrease the speed of movement of the head when TEST = 0 to 400 ms / step (sm.komandy VG93)	
23 HRDY (HLT)	input - readiness head (head in the working position, i.e. loaded)	
24 CLK	input - the clock signal (for drive 133 mm = 1 MHz)	
25 RSTB (RG)	exit - gate reading, confirming the reception of the data from the floppy - sign preamble (RSTB = 1 after taking 2 bytes "0" when single density and 4 a "0" or "1" - when the double-density)	
26 S (RC, RCLK)	input - playback clock produced from RAWR	
27 / RAWR (/ RA)	Input - pulses read data from the floppy disk	
28 HLD	exit - loading the magnetic head	
29 TR43	output - indicates that the head is on the track with the number> 43 (44 ... 79)	
30 WSTB (WG)	Output - the write enable strobe (= 1 during a write to disk)	
31 WD	output - the data pulses are recorded on floppy disk	
32 CPRDY (READY)	input - drive readiness to perform recording or reading (yet drive is not selected and it gained momentum signal READY = 0; wherein the integrated circuit generates a signal INTRQ)	
33 WF / DE	bidirectional bus: when written to disk (WSTB = 1) operates as input WF - "error recording "and WF = 0 when recording is stopped; when reading from disk (WSTB = 0) works as output DE - "Data resolution", which after loading the head and installation HRDY = 1 is "0". Open-drain output: requires Rload. About 10 ohms connected to the +5 V	
34 / TR00	input - signal "0" is supplied from the disk drive when the head set to zero track	
35 / IP (/ INDEX)	input - index pulse "0" from the disk drive (indicating the start track)	
36 / WPRT	entrance - "0" signal is supplied from the drive if the drive is protected from recording	
37 / DDEN	input - an indication of the recording density, "0" - double, "1" - single	
38 DRQ	Output - the data request; in read mode indicates that the chip is ready to give	

	the next byte read data; in the recording mode - a willingness to take the next byte to write. Open-drain output
39 INTRQ	yield - interrupt request; willingness chip (INTRQ = 0, when finished performing read command or a status register). Open-drain output
40 U2	Supply voltage +12 V

* NOTE: Symbol terminals in different sources differ.

SYSTEM COMMAND CONTROLLER LSI 1818 VG 93

The BETA-interface command register / status has an address # 1F.

The chip provides the 11 teams:

BIN HEX
0000 hvrr 00 ... 0F recovery
0001 hvrr 10 ... 1F search
001t hvrr 20 ... 3F step head (in the same direction)
010t hvrr 40 ... 5F forward
011t hvrr 60 ... 7F ago
100m seca 80 ... 9F sector reading
Sec0 101m A0 ... BF sector record
1100 0e00 C0, C4 reading addresses
1110 0e00 E0, E4 track reading
1111 0e00 F0, F4 recording track (at formatting)
1101 iiii the D0 ... the DF forced interruption

Page 27

Value command bits:
rr - head positioning speed:
r1 r0 T step
0 0 6 ms
0 January 12 ms
1 0 20 ms
1 January 30 ms
(For Fclk = lMrq. When the TEST signal (= 0) at the respective input period is about 400ms and not change)
v - checking the track number after positioning
h - head loading
t - change a track number (a track reg.) after each step
a - the type of address tags (data protection) (0- # FB, 1- # F8)
c - check of the disk number in the ID no. index area
e - delay after loading head (HLD signal) 30 ms
s - drive side
m - multi-sector operation (processed several sectors)
i - Interrupt condition:
i0 - the actuator changes to the "Ready"
i1 - the actuator changes to the "Not Ready"
i2 - on the index pulse
i3 - immediate interruption

The TR-DOS commands usually used for OR # 18 mask.

The "recovery" die performs setting to zero track magnetic disk. If the signal does not appear TR00, the team ends after 256 steps. Always It is performed when the controller is reset, regardless of the drive is ready.

The "Search" - to register the track is the current track number and the register the requested data is recorded. Moving operating until they coincide. It should be flag is set v (bit D2 = 1).

"STEP" command provides a pulse on the STEP issuing movement of the magnetic head one step in the same direction, as "one step forward", "STEP BACK" - change signal DIRC - direction of movement.

The command "READ SECTOR" provides reading sector, which number in the current The track was recorded in the register of the sector. side of the disc is set flag s (0,1). when set flag m - multi-sector operation (sector number is automatically increased until the reached the end of the track and the operation is repeated). sector length in formatting (cm. below). Flag of a - kind of address label data protection. When a = 1 is recorded Byte # F8, allowing the sector erase, when a = 0 - #FB, forbidding. At first read Sector ID (see below.); if an address mark is detected in the status register set a sign "array is not found." If matched track number, sector, and side CRC code, data is read: the next byte in the data register is issued, and accompanied DRQ signal (data ready). The register must be read before taking the next byte; otherwise the flag "is set in the status register loss . Data "at the end of reading is compared to the checksum CRC; if not the same, flag is set "error control code to" multi-sector operation is stopped.

"Record SECTOR" performed like in the previous part identification sector. DRQ signal is generated, requesting the first data byte. Then calculate 22 bytes to double (11 - for single) density - the space between the index area and data. Thereafter, if the DRQ served and the data register received byte strobe is issued and WSTB recorded data, starting with zero bytes and address labels (sm.format). data register should receive the next byte to each data request signal DRQ, the recording rate. If a byte is received, a status register flag is set "loss of data" rather to disk Writes a byte to "# 00". After data is recorded, two byte checksum CRC,

calculated by the controller, the cyclic amount A = X ^ 15 + X ^ 12 + X ^ 5 + 1 and a byte - gap. WSTB signal is removed (set to 0).

The "read address" reads 6 bytes of the index field (including CRC), and their transfer to the computer. If the CRC does not match, the status bit is set "The CRC error", and the reading continues. The command byte from the register track It is stored in the sector register. At the end as usual, set the ready signal INTRQ is reset and a status bit "busy".

The "READING TRACK" reads all the information on the track, including indexed array, the CRC, identifiers, and data gaps. It is not issued strobe read and checked the CRC, allowing you to use this command in diagnostic purposes.

The "TRACK RECORDING" is designed to format the KMT. All Information for this procedure must be formed by a computer including spaces and field indexes and data with all labels. Any sequence of data is written, for except byte # F5- # FE, which are interpreted as control address labels data. At the time of receiving data register bytes # F8- # FE for FM mode (FM) or byte # F5 to MFM mode (MFM), the control code is generated. When the byte code # F7 CRC (2 byte) is written to disk. Thus, when you format these bytes are not present in the spaces, indexed arrays and data (cm. size).

The "FORCED ABORTION" is set for the completion of any executed command. Unlike other commands, it can be written in the command register any time. Execution condition depends on the least significant bits, the team (i0 ... i3): if they are equal to "0", the current command is terminated and INTRQ not produced. When i0 = 1 interrupt is executed after the transition CPRDY signal from low to high when i1 = 1 - on the contrary, from "1" to "0". When i2 = 1 - on receipt of an index pulse (IP), i.e. top track. At i3 = 1 is an immediate interruption command. Once these conditions are met INTRQ signal is generated.

STATUS REGISTER STRUCTURE 1818 VG 93

After the command is written in the status register status byte controller, depending on the result of executing the command:

COMMAND	DISCHARGE OF STATE REGISTER							
	7th	6th	5	4	3	2	1	0
supporting	R	P	H	F	C	T	I	Q
reading addresses	R	0	0	N	C	W	D	Q
sector reading	R	0	A	N	C	W	D	Q
track reading	R	0	0	0	0	W	D	Q
sector record	R	P	E	N	C	W	D	Q
track record	R	P	E	0	0	W	D	Q

The value of flag bits:

R - drive readiness (1 = not ready)
P - write protection (1 = disc is protected)
H - loading head (1 = loaded)
E - write error (error = 1)

A - type AM (1 = Write Erase)
F - search error (1 = position not correct)
N - array is not found (1 = not found)
C - error in the control code CRC (1 = CRC is not true)
T - head in the initial position (zero = 1 lane)
W - loss of data (1 = data is lost)
I - index pulse (pulse = 1)
D - data request (data request = 1)
Q - employed (1 = is the command)

Recording Format CD

EXAMPLE FORMAT DATA SET 1st SECTOR FOR DOUBLE DENSITY

index field

80 bytes # 4E	= Posleindeksny gap (5th space)
12 B # 00	= sinhropromezhutok
3 bytes # F6	= Write C2 - timing AM
1 byte #FC	= Index mark (index marker)
50 bytes # 4E	= First gap

1ST SECTOR

ID of the 1st sector

12 B # 00	= sinhropromezhutok
3 bytes # F5	= A1 post - synchro
1 byte #FE	Address mark identifier =
1 byte nn	= Track number, or a cylinder (# 00- # 4F)
1 byte nn	= Head number (side disk- # 00, # 01)
1 byte nn	= Sector number (# 01- # 10) (not zero!)
1 byte xx	= Length of the sector # 00- # 03: # 00 - 128 bytes # 01 - 256 bytes # 02 - 512 bytes # 03 - 1024 bytes
1 byte # F7	= Write 2 bytes previously computed checksum (CRC) ID
22 bytes # 4E	= Second gap

field data of the 1st sector

12 B # 00	= sinhropromezhutok
3 bytes # F5	= A1 post - synchro
1 byte #FB	= Address label data
NN bytes ??	= Data in accordance with the sector length
1 byte # F7	= Write 2 bytes previously computed checksum (CRC) data
54 bytes # 4E (# 00) third space =	
....	continued to write interrupt (4th space before the start of the index pulse)

Page 30

EXAMPLE FORMAT DATA SET 1st SECTOR FOR SINGLE DENSITY

index field

40 bytes #FF (00)
6 B # 00 = Fifth gap from the beginning of the index pulse
1 byte #FC = Index mark (1st array index)
26 bytes #FF (# 00) = first gap

1ST SECTOR

ID of the 1st sector

6 B # 00 = blank
1 byte #FE = Index data address mark
1 byte nn = Track number (# 00- # 4F)
1 byte nn = Head number (side disk- # 00, # 01)
1 byte nn = Sector number (# 01- # 1A)
1 byte xx = Length of the sector # 00- # 03:
# 00 - 128 bytes
# 01 - 256 bytes
# 02 - 512 bytes
# 03 - 1024 bytes
1 byte # F7 = Write 2 bytes previously computed checksum (CRC) ID

field data of the 1st sector

11 bytes #FF (# 00)
6 B # 00 = Second gap
1 byte #FB = Address label data
NN bytes ?? = Data in accordance with the sector length
1 byte # F7 = Write 2 bytes previously computed checksum (CRC) data
27 bytes #FF (# 00) third space =
.... continued to write interrupt (4th space before the start of the index pulse)

Overhead byte index area

Byte	**appointment**	
	In FM mode	***The MFM mode***
# F5	not allowed	tags # A1 recording calculated CRC
# F6	not allowed	Record label # C2
# F7	is written 2 bytes previously computed CRC	is written 2 bytes previously computed CRC
# F8- # FB	post # F8- # FB with CLK = # C7, calculated CRC	Entry # F8- # FB
#FC	#FC record with CLK = # D7 (index mark before the first array index)	#FC entry (index mark before a first array index)
#FD	Record #FD with CLK = # FF	record #FD
#FE	#FE record with CLK = # C7 (index mark Information at the beginning of the index of the array) CRC is calculated	#FE record (the index data in the tag the beginning of the index of the array)
#FF	#FF record with CLK = # FF	record #FF

See. "V. Kovalenko et al. KR1818VG93 controller LSI for FDD. / Microprocessor tools and systems - 1986 № 3 s.3-8 ".

Page 31

Example 0th track TR-DOS - Disk:

0C first space
E4 (12 bytes) (Not synchronized)
AND# E0 Sinhropromezhutok

	D	# 00 (11 bytes)	(Not synchronized)
	E	#14	clock pulse
	H	# A1	
	T	# A1	
	AND	#FE	AM index
P	F	# 00	cylinder number
E	AND	# 00	head number
R	TO	# 01	sector number
AT	A	# 01	sector length
s	T	#FA	check sum
TH	ABOUT	# 0C	
	R		
		# 4E (22 bytes)	second gap
FROM			
E	P	# 00 (12 bytes)	sinhropromezhutok
TO	ABOUT	#14	clock pulse
T	L	# A1	
ABOUT	E	# A1	
R		#FB	AM data
	D	# 00 (256 bytes)	data
	A	# E1	check sum
	H	# 22	
	H		
	s		
	X		
		# 4E (60 bytes)	the thi
		# 00 (12 bytes)	the fol
		....	(# 09,
			are marked similarly

Original text

контрольная сумма

Contribute a better translation

CONTACTS SOEDENITELYA DRIVE

drive connector is compatible with SHUGART:

```
April 2                                              34
○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○
○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ - contacts "GND"
13                                                   33
```

2 - is not connected (NC)
4 - is not connected (NC)
6 - Selection of the drive "D:" (CS3)
8 - Index (IP)
10 - Selection drive "A:" (CS0)
12 - Select the drive "B:" (CS1)
14 - Select the drive "C:" (CS2)
16 - Head Loading (LOAD HEAD or MOTOR ON)
18 - direction (DIRC)
20 - Step (STEP)
22 - Recording of data (WR DATA)
24 - Write Enable (WR GATE)
26 - Zero track (TRACK 00)
28 - Protecting the disc recording (WPRT)
30 - Read data (RD DATA)

32 - Side 1 (SIDE1)
34 - Drive Ready (RDY)

Yu Pomortsev / Lions 1994 /

Printed in IPK "Moskovskaya Pravda"
Moscow, 1905, 7.
Tire. 10000. Order 4259