K E N - D O S

O P E R A T I N G M A N U A L

Voorlopige versie.

d.d. 15-3-84 verbeterde
& uitgebreide versie
o.a. met stack-init modificatie.
zie hiervoor achterzijde.

P

T A B L E O F C O N T E N T S

P

INSTALLATION.

In order to have KEN-DOS working it is necesarry to connect two points on the main PCB by means of the wire provided with the KEN-DOS system-package.
Pin 49 of the X bus has to be connected with pin 8 of the DCE-bus, thus routing the hold-line to the DCE-bus.
This hold-line enables handshaking.
Without this modification it would be impossible for KEN-DOS to work in double-density mode.

SOME ADVICE WHEN MAKING THE CONNECTION.

Remove all cables, connectors etc. from the back-panel of your PC.
Remove the enclosure of your DAI-PC. This is accomplished by removing the four black pins on the right and left side of the enclosure.
Put both tumbs on the keyboard and fold your fingertips on both sides under the edge of the white DAI-PC enclosure.
Gently pull the enclosure upwards and push to the rear.
Be carefull not to damage the hooks on the DCE-bus connector.
Take away the bottom of the enclosure by removing the black pins.(with newer versions of the DAI-PC these pins have been replaced by plastic nuts and bolts).
Carefully lay the main-PCB with component-side down on a soft surface. Put something under the side where the video-board is located to prevent it from damage.

SOLDERING THE WIRE.

Closely examine figure-1
Take the wire provided with your KEN-DOS package and strip 1 mm of the isolation and put some solder on the tip.
Put some fresh solder on pin 49 of the X-bus and pin 8 of the DCE-bus too.
Carefully solder one side of the wire on pin 49 of the X-bus.
Pull the wire tightly past the base-plate of the power-transformer to pin 8 of the DCE-bus. Keep the connection as short as possible.
Cut the wire, allowing 2 mm extra length and remove 1 mm isolation.
Put some solder on this point and carefully solder it to pin 8 of the DCE-bus. The connection is now made.
Check thoroughly for any short-circuits or drops of tin you might have spilled etc.
When you are convinced that everything is in order, remount the main PCB on the bottom of the enclosure.

Now install the provided EPROM-board on the X-bus connector, with the EPROM-sockets facing the keyboard. Note that the EPROM-board is installed slanting backward. DO NOT APPLY FORCE. Pay attention to the pins of the X-bus connector being exactly in line with the connector on the Eprom-board. Look at the back-side as well.
Replace the top of the enclosure.
After reconnecting the cables to the rear-panel, switch your computer on. If everything functions normally, you have probably done a fine job.(Run a few programs to make sure)
When your PC does nothing at all or does not behave the way it should,then re-open the enclosure and check again for short-circuits. Check again if you have soldered the wire at the right pins. If you cannot find anything that should not be the way it is, then disconnect the wire you soldered between the two connectors. Also remove the EPROM-board.
If your computer still does not function normally, the problem must lie elswhere. If your PC works fine with the wire disconnected, reconnect it the proper way and try again. The problem might be in the EPROM-board.

TESTING THE SYSTEM.

Switch-off your computer and connect the drive unit to the DCE-bus at the rear-panel of your DAI-PC.
The connector on the drive-unit cable fits into the DCE-bus connector in one way only.The notch on the drive-unit cable has to be turned upward.
Now switch your computer on.The screen displays the message "KEN-DOS V3.1" or just the cursor is vissible.You might even see an error message with "BREAK".
If you see one of the things mentioned before you can be sure the auto-start is working correctly.
Switch-off your computer and drive unit. Put a blank floppy-disc in drive 0 (the drive on the left). First switch on your drive-unit and then your computer.On the screen appears"KEN-DOS V3.1".
The KEN-DOS system is now ready for testing.

With the system still switched-on and a floppy-disc in drive 0 you enter "INIT0" and press the return key. After a few moments appears the message"TRACKS" on the screen. You now enter either "40" or "80" depending on the drive(s) you installed and press return . The system displays the current DATE wich in this case will be '000080'. You can enter the correct date by pressing the space-bar and typing the current date 'DDMMYY' or simply hit 'RETURN' in wich case no date will be entered.
Then enter "FORMAT" and press return again.

If everything goes well, you'll see a "0" appear on the screen every second or so.If a "5" appears after a rather long period of waiting,something is wrong with the system. If this might happen, refer to the section "TROUBLE-SHOOTING".
If you have installed 40-track drives, 40 "0"'s will appear on the screen.
With 80-track drives, two rows of 40 zero's each, will be dispayed.
When the FORMAT command is finished, the cursor will reappear.
Enter "DIR0" and press return. On the screen information regarding the floppy-disc you have in drive 0 will be displayed.
When you press the space-bar you will hear the drive click and the red led at the front will light-up.
Press return and then enter "TEST0". If no error message appears on the screen, drive 0 is working well.
If you have installed double-sided drive(s) or more then one single-sided drive, you will have to repeat all steps of the test.
When issuing a command you will have to replace the "0" with the number of the drive you want to test.Refer to the section 'DRIVE-UNIT' in the hardware part of this manual to see how KEN-DOS numbers the drives.

After you have completed all tests the system is ready.

HARDWARE.
The KEN-DOS system hardware consists of two parts:
1)EPROM-board with system-software in EPROM
2)Drive unit with controller-board,powersupply,enclosure
and drives (optional)

EPROM-BOARD.
On the board there is room for a maximum of 96Kbyte of EPROM
divided over 6 EPROM's
The smallest EPROM is the 2716 (2Kb), the largest is the
27128(16Kb) EPROM.
Should the latter be installed, then one jumper has to be cut,
while another has to be connected.(refer to fig.2/3).
The EPROM-board is normally configured to accept all types of
EPROM's except 27128.
The EPROM's on the board are placed in memory area #F000-#F7FF.
KEN-DOS also uses a heap at addresses #F900-#FAFF.
On power-on the bankswitch routines are written to this area
of memory. We recommend not to use this memory-area.
Should you do so, you run the risk that the system will "hang"
which will certainly cause loss of all your data in memory.
The memory-banks are switched at address #F900. To be able to
switch to another memory-bank a zero has to be written to location
#0296.If you fail to do so and try to switch banks the system
will crash. You can avoid this problem by switching on the
computer without switching-on the drive-unit.
The operating system resides in EPROM 1,2 and 3.In fact EPROM
3 is reserved for a CP/M bios which will be available soon.
The other sockets are available to the user.It is therefore
possible to put often-used software in EPROM's, which can
then be addressed via the "BANK"command. In this way it is
possible e.g.to load and run DNA within 1 second! Basic programs
can also be put into EPROM's, but occupy relatively much memory.

The EPROM sockets are numbered as follows:
The rightmost socket is no.1 and the leftmost is no.6(fig.2).
Bankno's increase by "8"
256 bytes.EPROM no.1 starts with 00(bank 1).The second bank is
00 + 8 = 08 etc.
EPROM 2 starts with 01
" 3 " " 02
" 4 " " 04
" 5 " " 03
" 6 " " 05

If you want to read from bank 4 which is located in EPROM 1,
you have to write a "0" to memory location #0296.
Then write #18 to memory location #F900.(03+8+8+8=27 = #18)
To return to KEN-DOS write a "0" to #F900 and "2" to $0296.

On EPROM-boards rev.2.0 it is possible to use socket 6 for a read/write device.Situated between sockets 5 and 6 are two small points, closely together, to one of which the R/W signal is routed. By connecting these to points and cutting a trace at the back-side of the board (see fig.1,2 W and X), you can place a 6116 static RAM in socket-6.
It is then possible to read and write to address #F700-#F7FF.
In order to enable a write operation you will have to issue a "DI" assembler-command first.(disable interrupt).
If you fail to do so, a stack-overflow will occur.

Provision has been made on the EPROM-board to enable users to install a keyboard-beep. This will give acoustical feed-back when a keystroke has been accepted by the computer.
An application-note fully describing this feature will follow soon.

THE DRIVE-UNIT.
The drive-unit houses the controller-board. All controll-signals between computer and disk-drives and all data-transfer is handled by this VLSI device.
The controller used by the KEN-DOS system is capable of transfering data between floppy-disk and main memory at a rate of 250 Kbits per second,using MFM technique. Data is checked on CRC errors.
The controller can handle a maximum of 4 double-sided drives.
The KEN-DOS operating system looks at double-sided drives as two single sided drives. Four double-sided drives are therefore handled like 8 single-sided drives.
Drives are numbered as follows:
Drive (0/1),(2/3),(4/5,(6/7)
If you have installed two single-sided drives you can read from them or write to them as drive 0 and drive 2.
Do you have two double-sided drives you can refer to them
as logical drive 0 and 1 physical the first drive
logical drive 2 and 3 physical the second drive.

The system comes complete with connectors for two drives.
Should four drives be connected to the system, a "T" connector has to be pressed on to the cable.

The controller can handle both 40- and 80track drives. The software supports both formats. Allthough it will probably not cause any damage,we do not recommend to connect both kinds of drives to one controller at the same time.
With 80 track drives it is possible (using a utility program) to read 40 track diskettes.The other way around is not possible.
The directory of either format can be red on both systems.

Provisions have been made on the controller-board to support connection of 8" drives.These drives can only be used in single-density mode. For more information on this subject, refer to the 8" application note.

THE FLOPPY DISC.
The floppy-disc used, are of the soft-sectored type and are formated at 5 sectors per track. This makes it possible to store 400 Kbyte of data per disc-side using 80 track drives and 200 Kbyte per side using 40 track drives. In a system with two 80track double-sided drives you can store 1600 Kbyte (1.6 Mbyte) of data.
To read 400 Kbyte of data in a sequential cyclus, KEN-DOS only needs 32 seconds. This is fast enough for animated graphics or wordprocessing using overlay's (paging).
The first 3 tracks on a disk are reserved.
Track zero for the directory and track-2 and track-3 for subdirectories.
Sector 5 of track 2 is used by the "TEST" command.
This means that the user has access to 400 minus 15 Kbytes of storage.
The directory on double-density disks allows 128 entries.
By using sub-directories this can be increased.
Maximum file-length is 250Kbyte.
A file can be overwritten even if the new file is larger than the old one. For sequential files KEN-DOS uses "dynamic file allocation". Random files have to be created before-hand and are of a specific length. It is, however, possible to make a Random-access file larger than initially created. This influences acces-time, allthough this will be hardly noticeable.
To deal with all the above, KEN-DOS uses a "file allocation map (FAM).This map is located on track zero and occupies 512 bytes.
The main directory occupies 4 blocks of 1Kbyte each.

POWERSUPPLY.
The powersupply is dimensioned to provide adequate power for at least 3 disk-drives.When the user wants to install 4 slim-line drives in 1 cabinet, it will be necesarry to mount a fan at the back of the cabinet to avoid heat-problems.
Stack-overflow can occur, caused by spikes on the mains-supply.
We advise to apply a mains-filter.
Without mains-filter you run the risk of loosing data.

THE DRIVES.
All SHUGART-compatible drives can be used, provided track to track steptime is 6 ms.or less.
With longer steptimes the drives can also be used (decreasing system performance), but a modification in the operating system has to be made.

MEMORY MAP
addressen in hex.

0000	DAI system-heap
02EC	start of free RAM
AD50	start of FAM (mode-0)
AF50	start of directory(mode-0)
B350	bottom of screen (mode 0)
C000	start BASIC ROM's
F000	start KEN-DOS
F800	start DAI stack
F900	start KEN-DOS heap/bankselect address
FA50	start KEN-DOS bank-switch routine
FB00	
FFFF	END

IMPORTANT ADRESSES.

#0180 Buffer for driveselect byte

#0296 Inswitch vector; 0=RS232
2=pointer op #02E0

#0297 Pointer to KEN-DOS command-table.
#0298

#02C5 Pointer for disk and/or cassette.
#02EB Do not change these addresses!

#F000 Pointer to initializing-routine which enables KEN-DOS commands to be used in Basic programs.

#F98C On this address "10" is written during start-up.
#F98D Second byte for motor-on time after finishing disc-acces. Is used together with #F98C as count-down buffer.

#FAFE Offset pointers to relocate the 1.5Kbyte needed by the KEN-DOS system as buffer for FAM and directory.
Usually this buffer moves with the lower part of screen memory in various screen modes.
This buffer may be overwritten e.g. by "EDIT"
By putting an offset-value in these addresses (contents is normally 0000), the user himself can determine where the start of the buffer is.
The formula is: D=B-KB-O
D=destination address buffer
B=address bottom of screen (contents of #02A5)
KB=1.5Kbyte (#600)
O=offset-value in #FAFE/#FAFF

FORMAT, BACK-UP, COMPAC, COPY and RND-buffers use parts of memory as specified below:

#8000 Start of FORMAT buffer
#B350

#0B00 Start of BACKUP, COPY and COMPAC buffer.
#B350

#9950 Start of RND-buffer (5 buffers of 1 Kb. each).
#AD50 If only 1 buffer is used, then the startaddress will be #AD50-#400; Using 2 buffers the start address will be #AD50-#800 etc.

GENERAL INFORMATION.

KEN-DOS has two kinds of commands.
The first type can only be used in direct-mode,the second type can be used in direct-mode and in programs.
The first type of command has a "*" behind the command name in the KEN-DOS command table.

Most commands use a filename.
This filename can be used in short-hand notation by placing a "/" behind the name. DOS only looks for a name which confirms with the part before the "/".
On writing to disk a new file is created if no match in name is found. If the name contains a '/' KEN-DOS creates a new file without "/".
If more files are present on the disk with the same first characters in their name,DOS will find only the first one.
We recommend to use this short-hand notation with care.
Example: DLOAD1"KENDOS" may be written as
DLOAD1"KEN/"
In both cases the file KENDOS will be loaded,except when another file starting with "KEN" is present on the disk and comes before KENDOS in the directory.

Drives are generally selected by placing the drive number behind the command. With the basiccommands "LOAD,SAVE,LOADA,SAVEA, R,W,the drive number has to be before the filename. This is caused by the structure of DAI-basic.
Example:the correct syntax is LOAD"1KENDOS" and not LOAD1"KENDOS".
If in disk commands a number is placed before the filename, KEN-DOS will assume this number is a drivenumber.
Example:DLOAD1"2KENDOS"means that the file 'KENDOS' will be loaded from drive-2 and not from drive-1.

If a drive has been selected for read or write operations it will remain selected untill another drive is specified in a read or write operation.
Example:LOAD"3KENDOS" followed by e.g. SAVE"KENDOS" will write the file 'KENDOS' to drive 3.
NOTE.This only applies to DAI-commands and not to KEN-DOS-commands.

Commands must be entered without space(s) between command and following data.
Example:DLOAD1"KENDOS" is correct,but DLOAD1 "KENDOS" is not.

The different filetypes are determined conform DAI concept.
BAS=$30, UTY=$31, ARY=$32, SRC=$33, RND=$34, TXT=$35 and DBS=$36.

In BASIC programs KEN-DOS commands are selected according to the DCR-protocol: CALLM#F000:REM (KEN-DOS command).

If an error report occurs always try the same commands again.

COMMAND TABLE

BAS *
BACKUP
BANK
BUF
CAS
CLOSE
CLR
COM
CODE *
COMPAC
COPY
CPM *
CREATE
DATE *
DCR *
DELETE
DIR *
DISK
DLOAD
DNA *
DSAVE
FIND
FORMAT
FWP *
GET
HELP
INIT *
KEY
KILL
LIB *
LOAD
LOADA
LOCK
LPRINT
NAME
MANUAL *
OPEN
PRT
PUT
R
RCAS
RENAME
RESTORE
SAVE
SAVEA
SPL *
SWAP
TEST *
TIME
UNLOCK
VERIFY
VOICE
W
WCAS

COMMAND SUMMARY

Brackets '()' have no significance.
(D) means DRIVENUMBER unless stated otherwise.
(N) means NAME
(S) means STARTADDRESS
(E) means ENDADDRESS

BAS
Syntax: BAS
Purpose: Is used to relocated a Basic program in memory. Can also be used to change Basic pointers.

BUF
Syntax: BUF(n)"(d)"
n=buffernumber 1-5
d=SET (init buffer-n)
CLR (clear buffer-n)
PRT (print buffer-n)
EDT (edit buffer-n)
Purpose: Used to create, delete, change or edit disc buffers.

CAS
Syntax: CAS
Purpose: Assigns system to cassette read/write.

CLR
Syntax: CLR(D)

Purpose: Changes disc protect-status.

COM
Syntax: COMD(D) of COM(D)"(S)(N)"
S=(+)(file can be run)
(++)(AUTO-run)
(&)(file is command file)
( )(status in question is deleted)
Purpose: Create Auto-start and COM-files.
Reads command-filenames from disc and puts them in extended command table (first three characters only).

CPM
Syntax: CPM
Purpose: Assigns system to 'CP/M'.

DCR
Syntax: DCR
Purpose: Assigns system to DCR-TOS.

DIR	Syntax:	DIR(D) or DIR(D)"A" or DIR(D)"P"
		A=displays all information concerning file.
		P=displays sub-directory(for future use)
		Space-bar : scroll
		Cursor-L : load file
		Cursor-R : load and run file
		Return : return to command mode
	Purpose:	Display directory of disc(D) on screen/printer.
		scherm of printer.
DNA	Syntax:	DNA
	Purpose:	Jump to 'DNA' program. CALLM12000 not necessary.
FWP	Syntax:	FWP
	Purpose:	Jump to FWP-program.
		UT,Z3,G400 not necessary.
GET	Syntax:	GET(D)"(N),n,nn"
		n =recordnumber from 1-256
		nn=buffernumber from 1-5
		Buffers have to be created before-hand
		using 'BUF' command.
	Purpose:	Reads record (n) from file (N).
		Puts data in buffer (nn).
KEY	Syntax:	KEY or KEY"?" or KEY"(D)(N)"
		KEY function is executed
		*= Cursor-up function='HELP'
		(D)(N):Cursor-up function='SAVE'
		(D=Drivenr,N=filename.)
	Purpose:	Create funtion-key for 'HELP' or 'SAVE'.
LIB	Syntax:	LIB or LIB1
	Purpose:	Display commandtable
		LIB = display KEN-DOS commands
		LIB1= display extended commands(COM-files)
PRT	Syntax:	PRT(D)
	Purpose:	Protect disk against 'FORMAT'
PUT	Syntax:	PUT(D)"(N),n,nn"
	Purpose:	Write buffer (nn) to recordnr (n)
		in file (N).(see GET)
SPL	Syntax:	SPL
	Purpose:	Jump to SPL-program

BANK

Syntax: BANK(n)

Purpose: Get data from bank-n and put in memory.
(optional run.)
Data must start at #F010.
#F000=(S)
#F002=(L)
#F004=execute-address or 0000
#F006=:0=bas;1=uty;#FF=no data
Only in start-bank a '0' or '1'.
next bank always '#FF'.
#F007=next banknumber;0=end
#F008=:0=no execution;1=execution
#F00A=totall length Basic-file
#F00C=totall length textbuffer
#F010=start of data
BANK(n)
n=1 startbanknr.=#4
n=2 startbanknr.=#3
n=3 startbanknr.=#5
n=4 startbanknr.=#14
n=5 startbanknr.=#13
n=6 startbanknr.=#15
n=7 startbanknr.=#24
n=8 startbanknr.=#23
n=9 startbanknr.=#25
For more details see 'Hardware'

CODE

Syntax: CODE

Purpose: Entering of lockcode or mastercode
Mastercode permits opening of all 'locks',
except locks on discs which have been
locked on another system.

COPY

Syntax: COPY(D)"(N),(D)"
N=BAS,UTY,ARY,SRC,RND,TXT,DBS or N=filename.
These file-types are reserved names and
should therefore never be used as filename.

Purpose: Copy files from drive (D) to drive (D) or copy
all files of a specific type.

DATE

Syntax: DATE

Purpose: Entering date.
Displays contents of date-buffer.
Date can be entered after pressing
space-bar.

DISK

Syntax: DISK

Purpose: Assigns system to disc read/write.

FIND	Syntax:	FIND(n)"($)" or FIND"($),(S),(E)"
		$=string to be searched
		n=buffernummer
	Purpose:	Search for a string in buffer or in memory.
		If endaddress contains '%'(E%) then string
		address or 'not found'-message is not displayed.
		In both cases :#F800=0 or 1.
		0=not found;1=found
		String address on #F801/#F802.
HELP	Syntax:	HELP
	Purpose:	Displays menu with syntax of some commands.
INIT	Syntax:	INIT(D)
	Purpose:	Initializes drives
		Init is followed by 'Tracks:'
		Entering number of tracks is advisable(80/40).
		Date can be entered hereafter(see DATE).
		All connected drives have to be initialized.
KILL	Syntax:	KILL(D)"(N)"
	Purpose:	Kill a file.
		File has to have delete-status.
		Usefull to free disc-space.
LOCK	Syntax:	LOCK(D) or LOCK(D)"(N)"
	Purpose:	Adding a code to a disc or a file.
		This lock-code prevents reading or
		writing unless the correct pass-word
		has been entered.
NAME	Syntax:	NAME(D)
	Purpose:	Name a disc; max 31 characters.
OPEN	Syntax:	OPEN(D)"(N)"
	Purpose:	Open file for writing.
RCAS	Syntax:	RCAS
	Purpose:	Enable cassette-read; disc-write.
SWAP	Syntax:	SWAP(n)
		n=buffernumber
	Purpose:	Swap data between discbuffer1
		and discbuffer-n.

TEST	Syntax:	TEST(D)
	Purpose:	Checks operation of drive (D). KEN-DOS writes to and reads from selected drive. Data is written to sector-5 of track-2 and then red and compared.
TIME	Syntax:	TIME(D)
	Purpose:	Displays time on screen or writes time to disk.
WCAS	Syntax:	WCAS
	Purpose:	Enables cassette-write;disc-read.
CLOSE	Syntax:	CLOSE(D)"(N)"
	Purpose:	Close specified file for writing.
VOICE	Syntax:	VOICE"($)"
	Purpose:	Send data string to speech-synthesizer.
BACKUP	Syntax:	BACKUP(D)"(D)"
	Purpose:	Copies all data from drive (D) to drive (D).
COMPAC	Syntax:	COMPAC(D)"(D)"
	Purpose:	Copy all files,except deleted files, from drive (D) to drive (D).
CREATE	Syntax:	CREATE(D)"(N),(f),n" f=filetype : BAS,ARY,UTY,SRC,RND,TXT,DBS. n=number of 1kbyte-blocks.
	Purpose:	Create new file or enlarge existing one.
DELETE	Syntax:	DELETE(D)"(N)"
	Purpose:	Delete file
FORMAT	Syntax:	FORMAT(D)
	Purpose:	Format diskette in drive(D)
LPRINT	Syntax:	LPRINT(n) of LPRINT"(S),(E)" n=buffernumber
	Purpose:	Send data from buffer or memory to RS232-port. If endaddress (E) is followed by '$' then get from RS232-port and store in memory.
MANUAL	Syntax:	MANUAL(D)
	Purpose:	Manual read/write of sectors or tracks.
RENAME	Syntax:	RENAME(D)"(N),(N)"
	Purpose:	Rename file.

RESTOR(E)	Syntax:	RESTOR(D)"(N)"
	Purpose:	Un-delete file.
UNLOCK	Syntax:	UNLOCK(D) or UNLOCK(D)"(N)"
	Purpose:	Unlock disc or file.
VERIFY	Syntax:	VERIFY(D) or VERIFY(D)"(N)"
	Purpose:	Verify data on disc in drive(D) or data of file(N) on disc in drive(D).
LOAD	Syntax:	LOAD"(D)(N)"
	Purpose:	Read Basic-file.
SAVE	Syntax:	SAVE"(D)(N)"
	Purpose:	Save Basic-file. If file-name doesn't exist; create new file. (N!)=close file after writing.
DLOAD	Syntax:	DLOAD(D)"(N),(S)"
	Purpose:	Read BAS-file or UTY-file. Adding a '%' to the file name (N) enables writing data to screen-memory.
DSAVE	Syntax:	DSAVE(D)"(N)" or DSAVE(D)"(N),(S),(E)"
	Purpose:	Saves file or creates new file Basic files without (S) and (E) File name with '!'=close file
LOADA	Syntax:	LOADA"(D)(N)"
	Purpose:	Read array.
SAVEA	Syntax:	SAVEA"(D)(N)"
	Purpose:	Save array.
R	Syntax:	R (D)(N)
	Purpose:	Read UTY-file. (N%)=see 'DLOAD'
W	Syntax:	W(S) (E) (D)(N)
	Purpose:	Save UTY-file. (N!)=see 'DSAVE'.