

Hardware of the DOS Junior

In order to upgrade the Junior Computer to a DOS computer, certain hardware modifications are necessary. To allay any fears, let us point out that no tracks need be cut nor need any mechanical changes be made. All that is required is to solder an IC onto another one in piggyback fashion on the interface p.c.b. of the Junior Computer. In order to connect an EPSON dot matrix printer, an interface is required for the BUSY line. Since this interface only consists of three resistors, one transistor and one diode, it can easily be mounted in self-supporting fashion next to the V24/RS-232 connector.

G. de Cuyper

floppy-disk interface for the Junior

... and other 6502 computers, Part II

the Junior collects its stars and stripes

The second and last of these two articles describes the modifications that must be made to the hardware of the Junior Computer, to be able to run Ohio Scientific software on the Junior or any other 6502 computer. A new EPROM is required, to load software from the diskette during initialization (reset) of the computer. Documentation is available showing the source listing for the monitor program in the EPROM.

Let us take another look in the circuit diagram of the Junior interface p.c.b. (figure 1). NOR gate N33 is replaced by a NAND gate. Line 8K0 or EX is now no longer active in address range \$0000 ... 1FFF. The new address range with the NAND gate is E000 ... FFFF. This addressing affects the memory chips of the Junior Computer as follows:

- 48 K bytes of dynamic RAM are located in address range \$0000 ... BFFF. The advantage of dynamic RAM is that it is cheap and has a low current consumption. Three dynamic RAM cards (see Elektor, April 1982) are sufficient to provide the 48 K byte address range for the RAM.
- The address decoder on the standard Junior Computer p.c.b. (IC6) decodes address range \$E000 ... FFFF. The memory chips on the standard Junior Computer p.c.b. are therefore assigned

the following addresses:

EPROM IC2, type 2708:

\$FC00 ... FFFF

PIA, RAM, TIMER, type 6532:

\$FA00 ... FBFF (documentation)

RAM IC4 and IC5 type 2114:

\$E000 ... E3FF

The memory chips on the Junior interface p.c.b. have the following addresses:

VIA IC1, type 6522:

\$F800 ... F9FF (documentation)

RAM IC2 and IC3, type 2114:

\$E400 ... E7FF

EPROM IC4 and IC5, type 2716:

\$E800 ... F7FF

The second hardware modification concerns the interface for the BUSY line of the EPSON printer. Figure 2 shows how this interface is connected to the Junior. Relay Re1 is discarded. LED D4 becomes the BUSY indicator in this arrangement and operates in parallel with the BUSY lamp on the EPSON.

The drawings of figures 3 and 4 clarify the hardware modifications. After implementation of these modifications it is only necessary to plug the type 2708 (ESS 515) EPROM into the socket on the standard Junior Computer p.c.b. The two EPROMs IC4 and IC5 on the Junior interface p.c.b. (PM and TM) are no longer required, because the input/output programs for printer control are located in the 2708 EPROM. The address space assigned to IC4 and IC5 is now free for user programs. One more item that must not be forgotten is to fit a wire link between soldering points 'R' and 'S' on the Junior interface p.c.b. (WITH).

The Junior Computer has now been upgraded to a DOS computer. Plug at least two or three dynamic RAM cards into the bus card of the Junior Computer. The lines for address decoding on the RAM cards should be wired as follows:

RAM card 1: U - 0

V - 1

X - 2

Y - 3

RAM card 2: U - 4

V - 5

X - 6

Y - 7

RAM card 3: U - 8

V - 9

X - A

Y - B

At this stage, do not yet plug the floppy disk interface into the bus card of the Junior. Power-up the computer as usual and press the <RST> key on the hex keyboard. The display on the Junior should now light up. The command keys <AD>, <DA>, <+> and <GO> have the same significance as previously. Only the <PC> key has a new function. Once the disk operating system has been loaded into the computer, the DOS command interpreter can be called with the <PC> key. But we shall examine this in more detail later.

Table 1 shows the memory map of the DOS Junior Computer. This memory map also applies to all other 6502 com-

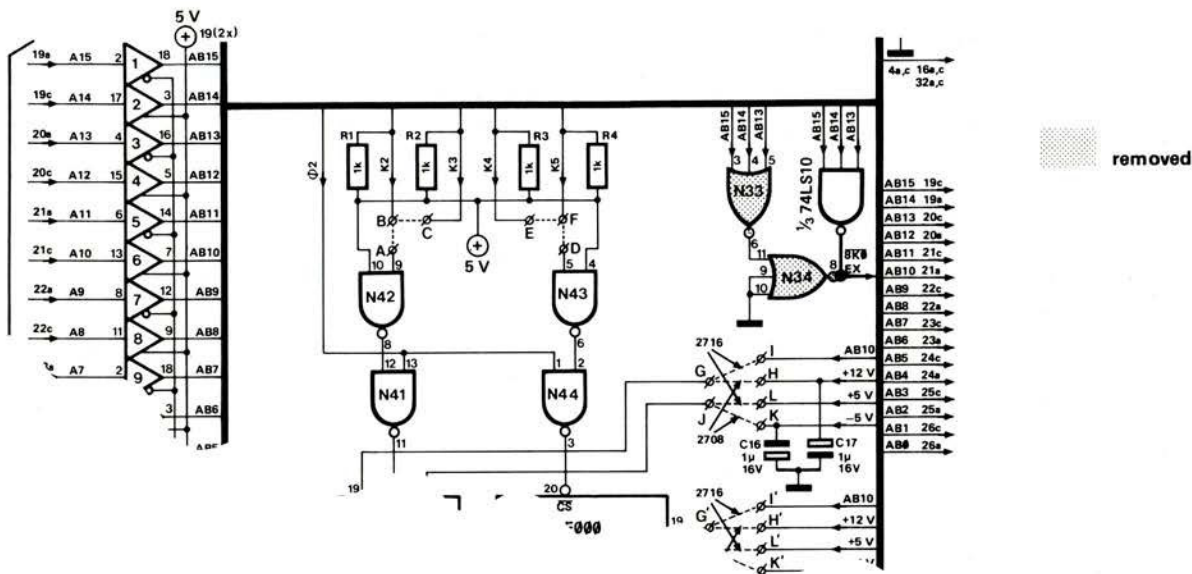


Figure 1. A part of the circuit diagram of the Junior Computer interface card. Gates N33 and N34 are replaced by a 74LS10 NAND gate. This results in a new memory area for the memories on the standard Junior Computer and on the interface card.

puters that are connected to the floppy disk interface. Adresses \$C000 . . . FBFF can vary from one computer to another. The only important detail is that the computer must be able to address at least 32 K bytes of contiguous RAM in the lower address range. If, on a computer other than the Junior, the address range FC00 . . . FFFF should already be assigned, the bootstrap software should be relocated to another memory area. This should be relatively simple using the documentation.

Software of the DOS Junior

The software of the DOS Junior Computer is oriented on the latest standards in computer design. This means that the computer is provided with a minimum of ROM intelligence and as much RAM as possible. The advantages of this computer system become clear: Since the system software, whether BASIC, FORTH, Assembler or a word processor, has to be loaded rapidly from the diskette into the computer, not much space is wasted for the ROM address range. The ROM contains only sufficient intelligence to allow the computer to operate the hex display and keyboard and transmit/receive to and from the Elekterminal. Another function of the ROM intelligence of the Junior is to load track 0 from the diskette into the Junior Computer. Thus BASIC, Assembler etc. are no longer stored in the ROM but are loaded from the diskette into the computer. This is referred to as 'portable software'. The decisive advantage of portable software is that it can be easily modified. Previously, when software errors were detected or when the system was to be updated, it was necessary to plug a new

set of EPROMs into the computer; now with portable software we only need to mount a new diskette on the drive. It is therefore easy to eliminate software errors and the system can be upgraded in keeping with the latest state of the art without difficulty. We make full use of these advantages with the DOS Junior Computer. Let us examine Table 1 once again. Page 0 and the system stack are located in address range \$0000 . . . 01FF. Address space \$0200 . . . 22FF is intended for the 'transient processor'. What is a transient processor? This is the software that makes the computer system operable. If, for example, you are working in BASIC, the transient processor is the BASIC interpreter. If you are working in machine language, then Assembler or Ohio Scientific's Extended Monitor is the transient processor. The system software for controlling the floppy disk drive and the printer, as well as a memory-mapped video display unit takes up about 4 K bytes of RAM. With OS-65D V3.1 the DOS software occupies memory area \$2300 . . . 3278. The work space or memory area in which the programmer's own program is stored begins at address \$327E . . . When you type a BASIC program into the computer it is stored from address \$327E . . . onwards. The data are written into the floppy disk drive from this address onwards. Data which are read from the diskette into the computer are also stored from this address onwards. With OS-65D V3.3 the work space begins at address \$3A7E. The first five bytes of the work space are known as the header. These five bytes contain the following information:

- 1) Start address on the file (2 bytes)
- 2) End address on the file (2 bytes)

BOOTSTRAP RES, IRQ, NMI	FFFF	2708
6532: RAM, TIMER, PIA	FC00	
6522: VIA	FA00	
IC5 Interface Board (2716)	F800	
IC4 Interface Board (2716)	F000	
2k Static RAM	E800	
Memory Mapped VDU	E000	
not used	D800	
Floppy I/O	C100	
Work Space	C000	
System Software DOS	2300	
Transient Processor	8200	
System Stack, Page Zero	0000	

Table 1. The new memory map for the DOS Junior Computer. On account of these extensions the workspace is reduced by 4 K bytes with the OS-65D V3.3. The transient processor overwrites address range \$ 2200 . . . 22FF of the DOS software.

2

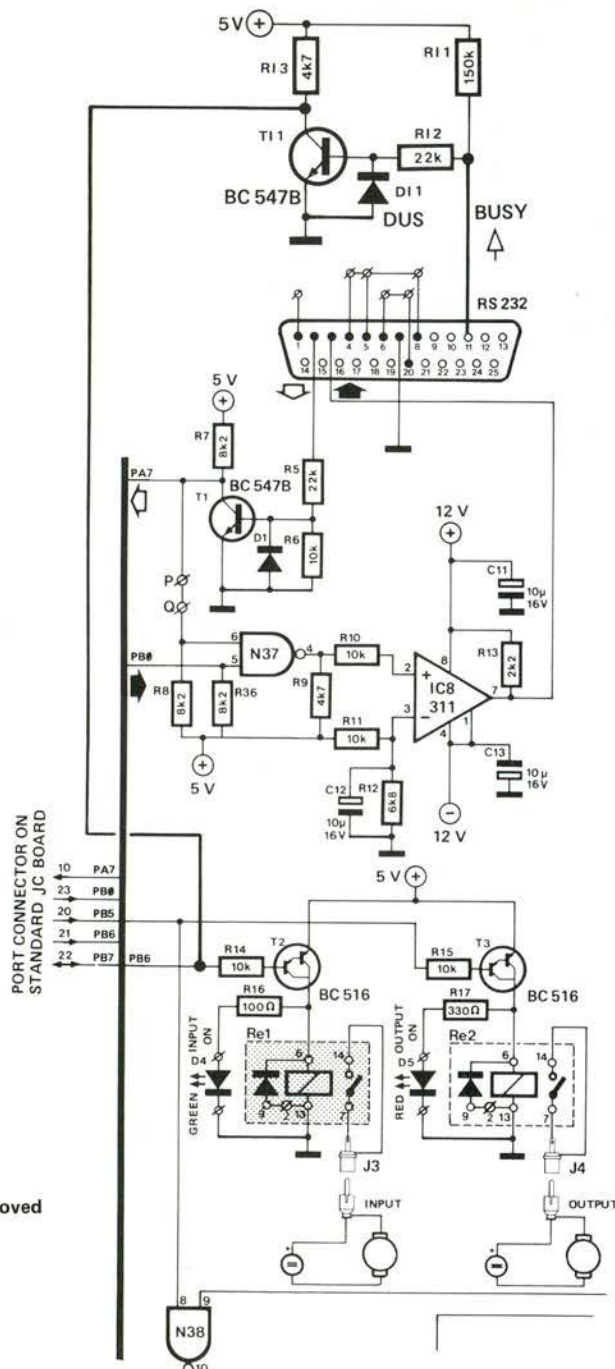


Figure 2. This is how the interface is connected to PB6 for the EPSON printer. If PB6 is a logic 1, the send output (RS-232 connector pin 3) of the computer is inhibited. If no printer is connected, resistor R11 ensures that the send output is enabled.

3

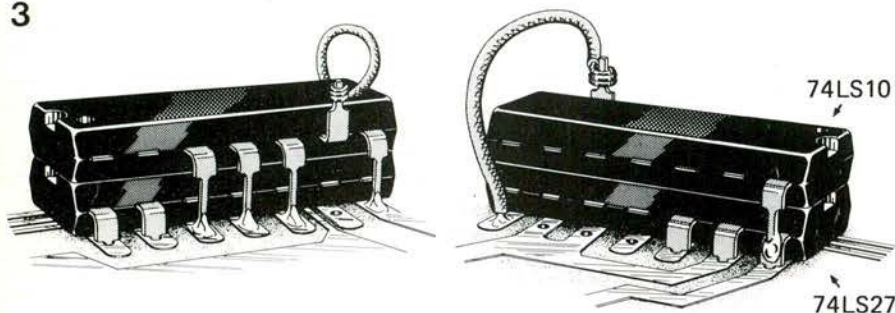


Figure 3. This hardware modification provides the Junior Computer with a new memory arrangement. The drawing shows the 'piggy-back' installation of the ICs.

3) Length of the file in pages (1 byte;
1 page = 256 bytes)

Next to the work space is the 4 K byte page \$DXXX. The memory-mapped video display unit of Ohio Scientific is located in this address range. A video unit of this type differs from the Elekterminal in that the computer can transport data to the monitor screen directly via the data bus. The data interchange between the screen and the computer is much more rapid with the memory-mapped video display unit (VDU) than with a video interface using the V24/RS-232 interface.

2 K bytes of static RAM are located in address range \$E000...E7FF. This RAM was previously located in the range \$0000...07FF. In future we shall use this RAM area for the object code that is generated by the Assembler. You can now assemble a source file with the Assembler, print a listing if desired and instruct the computer to execute directly the machine code generated by the Assembler. Please consult Ohio Scientific's Assembler Manual for further details.

Address range \$E800...FFFF to IC4 and IC5 on the interface card of the Junior and to the two interface chips 6522/6532 and the 2708 bootstrap EPROM. The two EPROMs IC4 and IC5 on the interface card of the Junior are now freely accessible to the programmer. Personal software or programs which one does not wish to load from the floppy disk into the computer can be loaded into the two EPROMs (2716) for long term applications.

The bootstrap EPROM (ESS515)

The bootstrap EPROM is addressed in the range \$FC00...FFFF. It contains only 1 K byte of software. The software in the bootstrap EPROM can be subdivided into eight groups:

1) Hex display monitor. This program is very similar to the original monitor of the Junior. The commands <AD>, <DA>, <GO>, <+> have kept their significance. Only the <PC> key has a different function. The DOS command interpreter can be called directly from the hex keyboard with the <PC> key. The main purpose of the hex display monitor is to be able to modify the software on the Ohio Scientific diskettes. This allows Ohio software to be converted to Junior software.

Important start addresses can be called with the hex display monitor and started with the <GO> key.

- RESBAS * \$FF17
- RESDOS * \$FF34
- VONE * \$FFE2
- VTHREE * \$FFE8

2) Loading the BASIC interpreter from the diskette. You can load BASIC from the diskette into the Junior Computer with the command <AD> FF17 <GO> (RUBOUT). To distinguish between command keys on the hex keyboard and Elekterminal keys, we shall

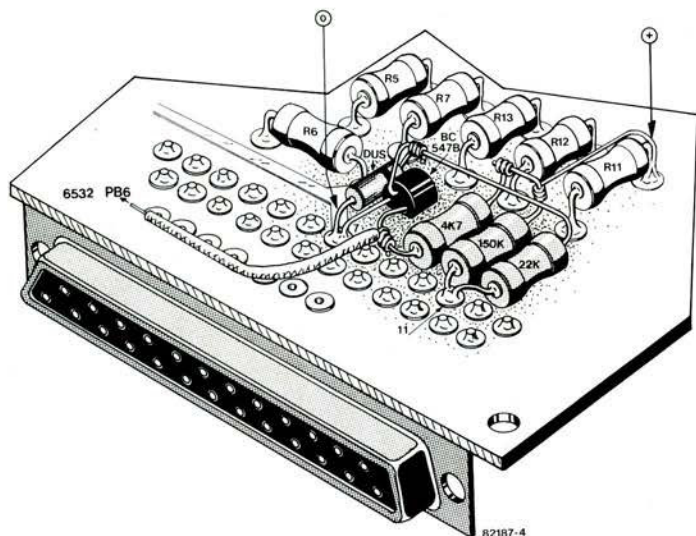


Figure 4. The EPSON interface is placed on the interface card of the Junior Computer as shown here.

write the command keys of the hex keyboard between <> symbols and command keys on the Elekterminal between () brackets in future.

Once BASIC is loaded and the computer has responded on the terminal the transient processor is ready for program interrupts. During output, you can interrupt a BASIC program with the (BREAK) key on the terminal. If the (BREAK) key was pressed during a LIST command, the BASIC interpreter prints 'Break'. If you have started a BASIC program with the RUN command and wish to stop output of the program, simply press the (BREAK) key. The BASIC interpreter responds with 'BREAK IN LINE X'. All program variables and pointers are then saved in the stack. You can restart the program after the interrupt with the CONT statement. The indirect jump vector for the (BREAK) key is automatically set by the computer via address \$FF17 when the BASIC program is started. The break vector is stored in address \$FA7C and \$FA7D.

3) Loading of DOS software from the diskette. In future you can load ELEKTOR's own software into your Junior Computer with the command <AD> FF34 <GO> (RUBOUT). The address DOS is intended for future expansions and 'non-Ohio-software'.

4) Adapting an OS-65D V3.1 Ohio diskette.

An OS-65D V3.1 Ohio Scientific diskette can be adapted to your Junior Computer with the command <AD> FFE2 <GO> (RUBOUT). When the monitor is started at the address VONE * \$FFE2 the following occurs:

- The computer positions the read/write head of the drive on track D.

- The computer reads the data on track 0 and stores these data at address \$2200 ... in the memory.
- The computer positions the read/write head of the drive on track 1.
- The computer reads the data on track 1 and stores these data at address \$2A00 ... in the memory. The entire disk operating system is now loaded in the computer and can be modified by the programmer with the hex keyboard.
- Once track 0 and track 1 have been loaded into the computer a jump is made to the hex display monitor

and the computer responds with *Track 0&1* on the printer.

5) Adapting an OS-65D V3.3 Ohio tutorial disk 5.

An OS-65D V3.3 Ohio Scientific tutorial disk 5 can be adapted to your Junior Computer with the command <AD> FFE8 <GO> (RUBOUT). For track 1 and track 1 the same applies as under 4). However, once track 0 and track 1 have been loaded into the computer the following occurs:

- The computer moves the read/write head of the drive to track 6. The data on track 6, sector 2 are read into the memory from address \$3200 ... onwards. Sector 2 is one page long.
- The data on track 6, sector 3 are read into the memory from address \$0000 ... onwards. Sector 3 is one page long.
- The computer removes the read/write head of the drive to track 13. The data on track 13, sector 1 are loaded into the computer from address \$3A79 ... onwards. Sector 1 of track 13 is eight pages long. When all tracks have been loaded into the computer the message *Track 0&1* is issued, as with OS-65D V3.1, and a jump is made to the hex display monitor.

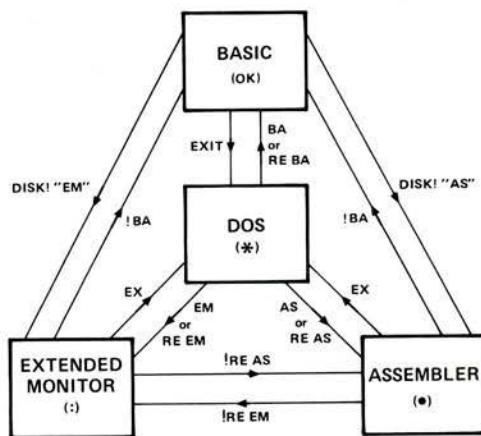
6) The <PC> key

You can leave the hex display monitor and jump to the DOS command interpreter with the <PC> key. Printer I/O is initialized but no new baud rate measurement is made. The computer responds with the prompt A* or B*, etc..

7) Printer input routine.

The printer input routine receives a character from the terminal. The received ASCII character is in the accumulator of the CPU after the return from the input subroutine. Bit 7 of the character is always a logic zero. The contents of the index registers are not affected

5



Transfer of Control Commands

82187-5

Figure 5. The command transfer to the different transient processors: from BASIC to ASSEMBLER, from ASSEMBLER to EXTENDED MONITOR, etc. and vice versa. The 'control centre' is the DOS. It allows the BASIC, ASSEMBLER and EXTENDED MONITOR transient processors to be loaded from the diskette into the computer.

by calling this subroutine. The start address of the printer input routine is: RECCHA* \$FE1B (call: JSR RECCHA).
8) Printer output routine.

The printer output routine sends the character in the accumulator of the CPU to the terminal. The transfer format is:

- one start bit
- seven data bits
- no parity bit
- two stop bits

The contents of the index registers are not affected by calling this subroutine. The start address of the printer output routine is:

PRCHA* \$FEA3 (call: JSR PRCHA).

How does the bootstrap routine operate?

Now that we know what the software in the bootstrap EPROM consists of, we should discuss operation of the bootstrap routine. Once you have entered

```
<RST>
<AD> FF17
<GO>
```

into the computer, the following operations take place:

1) The computer sets the indirect jump vector for the (BREAK) key on the terminal. The NMI vector points to the hex display monitor.

2) The computer initializes the I/O on the floppy disk interface p.c.b. (6821 and 6850) and the serial I/O for the terminal (6532). When all I/O lines have been initialized the computer waits for the RUBOUT character from the terminal, to measure the baud rate. The baud rate is stored in memory cells \$FA5A...FA5D (see floppy disk documentation).

3) Once the baud rate of the terminal has been measured, the computer loads track 0 (= 2 K bytes of software in machine language). The steps required to load track 0 into the computer are as follows:

- Position the read/write head of the drive over track 0. A light barrier in the drive informs the computer when the head is positioned over track 0.
- The computer then emits a head load pulse to the drive. The read/write head is lowered onto the surface of the diskette. The computer then waits for the index hole in the diskette. Another light barrier generates the index pulse as soon as the index hole of the diskette passes the light barrier.

— Once the index pulse has decayed, the control register is set in the ACIA (6850). The transfer format of the ACIA is: one start bit, eight data bits, one even parity and one stop bit.

— The computer reads the first byte from the diskette. This byte is the most significant start address of the memory area in which track 0 is stored (= \$22). The second byte from the diskette is the least significant byte of the start address (= \$00). Both bytes are loaded into the bump pointer (load

```
<RST>
<AD> FFE2
<GO>
```

(RUBOUT)

TRACK 0&1

```
<AD> 2217
<DA> 4C
<+> 40
<+> 22
```

```
<AD> 2245
<DA> 4C
<+> 76
<+> 22
```

```
<AD> 2283
<DA> 4C
<+> A6
<+> 22
```

<PC>

A*CA 0200=13,1

```
A*IN
ARE YOU SURE?Y
A*GO 0200
```

- DISKETTE UTILITIES -

```
SELECT ONE:
1) COPIER
2) TRACK 0 READ/WRITE
? 2
```

- TRACK ZERO READ/WRITE UTILITY -

```
COMMANDS:
Rnnnn - READ INTO LOCATION nnnn.
Wnnnn/gggg,p - WRITE FROM nnnn FOR p PAGES
                WITH gggg AS THE LOAD VECTOR
E - EXIT TO OS-65D
```

COMMAND? W2200/2200,8

- TRACK ZERO READ/WRITE UTILITY -

```
COMMANDS:
Rnnnn - READ INTO LOCATION nnnn.
Wnnnn/gggg,p - WRITE FROM nnnn FOR p PAGES
                WITH gggg AS THE LOAD VECTOR
E - EXIT TO OS-65D
```

COMMAND? E

A*HOME

A*SA 01,1=2A00/8

Table 2. Adapting an OS-65D V3.1 diskette to the Junior Computer (Part 1), if only one drive is available.

Table 3. Adapting an OS-65D V3.1 diskette to the Junior Computer (Part 2).

A*CA 4000=02,1	A*CA 4000=10,1	A*CA 4000=18,1	A*CA 4000=26,1
A*CA 4800=03,1	A*CA 4800=11,1	A*CA 4800=19,1	A*CA 4800=27,1
A*CA 5000=04,1	A*CA 5000=12,1	A*CA 5000=20,1	A*CA 5000=28,1
A*CA 5800=05,1	A*CA 5100=12,2	A*CA 5800=21,1	A*CA 5800=29,1
A*CA 6000=06,1	A*CA 5200=12,3	A*CA 6000=22,1	A*CA 6000=30,1
A*CA 6800=07,1	A*CA 5300=12,4	A*CA 6800=23,1	A*CA 6800=31,1
A*CA 7000=08,1	A*CA 5800=13,1	A*CA 7000=24,1	A*CA 7000=32,1
A*CA 7800=09,1	A*CA 6000=14,1	A*CA 7800=25,1	A*CA 7800=33,1
A*SA 02,1=4000/8	A*CA 6800=15,1	A*SA 18,1=4000/8	A*SA 26,1=4000/8
A*SA 03,1=4800/8	A*CA 7000=16,1	A*SA 19,1=4800/8	A*SA 27,1=4800/8
A*SA 04,1=5000/8	A*CA 7800=17,1	A*SA 20,1=5000/8	A*SA 28,1=5000/8
A*SA 05,1=5800/8	A*SA 10,1=4000/8	A*SA 21,1=5800/8	A*SA 29,1=5800/8
A*SA 06,1=6000/1	A*SA 11,1=4800/8	A*SA 22,1=6000/8	A*SA 30,1=6000/8
A*SA 07,1=6800/8	A*SA 12,1=5000/1	A*SA 23,1=6800/8	A*SA 31,1=6800/8
A*SA 08,1=7000/8	A*SA 12,2=5100/1	A*SA 24,1=7000/8	A*SA 32,1=7000/8
A*SA 09,1=7800/5	A*SA 12,3=5200/1	A*SA 25,1=7800/8	A*
	A*SA 12,4=5300/1		<RST>
	A*SA 13,1=5800/5		<AD> FF17
	A*SA 14,1=6000/8		<GO>
	A*SA 15,1=6800/8		(RUBOUT)
	A*SA 16,1=7000/8		
	A*SA 17,1=7800/8		

```
OS-65D V3.0
OSI 9 DIGIT BASIC
COPYRIGHT 1977 BY MICROSOFT
36225 BYTES FREE
```

OK

Table 4. Adapting an OS-65D V3.1 diskette to the Junior Computer (Part 3).

pointer). Thus the bump pointer points to address \$2200. The computer then reads the third byte from the diskette. This byte indicates the number of pages on track 0 (= \$08).

— The next bytes are 2 K bytes of software in machine language. These bytes are loaded from the diskette into the computer. The memory area is \$2200...29FF.

— Once track 0 has been stored in the memory, the read/write head is raised from the surface of the diskette and a jump to address \$2200 takes place. Starting with this address, the computer finds further instructions concerning the address into which the tracks and sectors are to be loaded. Normally the two K bytes of software on track 1 are loaded into address range

RUN"BEEXEC"

BASIC EXECUTIVE FOR
OS-65D V3.1

JUNE 25, 1980 RELEASE

FUNCTIONS AVAILABLE:

CHANGE- ALTER WORK-
SPACE LIMITS

DIR- PRINT DIRECTORY

UNLOCK- UNLOCK SYSTEM
FOR END USER MODI-
FICATIONS

FUNCTION? UNLOCK

SYSTEM OPEN

OK
RUN

BASIC EXECUTIVE FOR
OS-65D V3.1

JUNE 25, 1980 RELEASE

FUNCTIONS AVAILABLE:

CHANGE- ALTER WORK-
SPACE LIMITS

DIR- PRINT DIRECTORY

UNLOCK- UNLOCK SYSTEM
FOR END USER MODI-
FICATIONS

FUNCTION? DIR

LIST ON LINEPRINTER INSTEAD OF DEVICE # 1 ? NO

OS-65D VERSION 3.0
-- DIRECTORY --

FILE NAME	TRACK RANGE
OS65D3	0 - 12
BEXEC*	14 - 14
CHANGE	15 - 16
CREATE	17 - 19
DELETE	20 - 20
DIR	21 - 21
DIRSRT	22 - 22
RANLST	23 - 24
RENAME	25 - 25
SECDIR	26 - 26
SEQLST	27 - 28
TRACE	29 - 29
ZERO	30 - 31
ASAMPL	32 - 32

50 ENTRIES FREE OUT OF 64

OK

Table 5. 'Trial run' of an adapted OS-65D V3.1 diskette. Utility program BEXEC* is called when the BASIC interpreter has been loaded.

\$2A00 ... 31FF. Track 0 and track 1 jointly contain 4 K bytes of software in machine language, which make the entire disk operating system operable.

— Once the disk operating system (DOS) has been loaded into the computer, the BASIC interpreter is loaded from the diskette into the memory. With OS-65D V3.1 the BASIC interpreter is located on tracks 2...4. With OS-65D V3.3 the BASIC interpreter and various extensions of the editor are located on tracks 2...4, track 6 and track 13. When the interpreter has been loaded a jump is made to the cold-start entry of the BASIC interpreter (\$20E4). The system then responds with the prompt 'OK'.

— The system is not yet ready for the statements LIST, CONT etc.. Even a BASIC file cannot yet be created. The

OK
LIST

10 REM DIRECTORY UTILITY FOR OS-65D VERSION 3.0
20 REM
30 NF=0
40 PN=11897
50 DEF FNA(X)=10*INT(X/16)+X-16*INT(X/16)
BREAK
OK

NEW

OK
RUN"DIR"
LIST ON LINEPRINTER INSTEAD OF DEVICE # 1 ? NO

OS-65D VERSION 3.0
-- DIRECTORY --

FILE NAME	TRACK RANGE
OS65D3	0 - 12
BEXEC*	14 - 14

BREAK IN 11110

OK
CONT
CHANGE 15 - 16
CREATE 17 - 19
DELETE 20 - 20
DIR 21 -
BREAK IN 11100
OK

Table 6. Checking the (BREAK) key with the LIST command and the RUN statement.

	02
	03
	04
	05
	06
	07
	08
	09
	10
	11
	12
	13
	14
	15
	16
	17
	18
	19
	20
	21
	22
	23
	24
	25
	26
	27
	28
	29
	30
	31
	32

A*CA 0200=13,1

A*GO 0200

- DISKETTE UTILITIES -

SELECT ONE:
1) COPIER
2) TRACK 0 READ/WRITE
? 1

- DISKETTE COPIER -

FROM DRIVE (A/B/C/D)? A

TO DRIVE (A/B/C/D)? B

STARTING TRACK? 2

ENDING TRACK (INCLUSIVE)? 32

READY (Y/N)? Y

Table 7. If two floppy disk drives are available (drive A and drive B), an OS-65D V3.1 diskette can be adapted very easily. Copying is performed automatically by the computer.

only BASIC statement that the computer recognizes is RUN. If you wish to generate a BASIC file, load the program BEXEC* into the computer using the command:

RUN 'BEXEC*'

The computer then presents a menu of options. With OS-65D V3.1 select the option 'UNLOCK' and with OS-65D V3.3 select option '9'. The computer responds with the prompt 'SYSTEM OPEN'. Now enter a 'NEW' statement and the work space of the Junior is formatted for your BASIC file.

The DOS and its capabilities

The Junior Computer can easily be converted to various transient processors with Ohio's disk operating system, which is located on track 0 and track 1

of the diskette. Figure 5 shows how the individual connections between the DOS and the processors can be made and cancelled. If, for example, you start the DOS at address \$FF17 the Junior becomes a BASIC computer. If you want to convert the Junior to an Assembler computer, you must enter EXIT in order to leave the BASIC interpreter. After the EXIT command the Junior responds with the DOS prompt A* or B*, etc. on the printer. If you then enter AS or Assembler the Junior converts itself to an Assembler Computer. You can then create an Assembler file and write on the diskette with the DOS. However, you can also assemble the source file and program the object code directly in the EPROM with the ELEKTOR EPROM programmer, without having to type in a single byte. The computer performs this task automatically. For further details regarding operation with Assembler, please consult Ohio Scientific's Assembler Manual. Note that the Assembler and Extended Monitor are not on the diskette in the case of OS-65D V3.1. Only OS-65D V3.3 has the Assembler and Extended Monitor (EM) on the diskette as standard.

Adapting an OS-65D V3.1 diskette for one drive

If you have only one drive, V3.1 can only be adapted to the Junior Computer with a considerable amount of typing. Tables 2, 3 and 4 show how this is done. First we start the system with the hex keyboard. A program with which a V3.1 diskette can be loaded into the Junior Computer to facilitate modification begins at address \$FFE2. When the (RUBOUT) key has been pressed the computer loads track 0 and track 1 into the memory and responds with *TRACK 0&1*. The display lights up, the vectors for the input and output routines are automatically loaded and the programmer can now modify some bytes at addresses \$2217, \$2245 and \$2283. Once the bytes have been modified in accordance with Table 2, a jump to the DOS command interpreter is made using the <PC> key. The DOS responds with the prompt A*.

The command CA 0200-13,1 loads from track 13, sector 1 of the diskette. The data are stored in the computer from address 0200 onwards. The program that we have just loaded into the Junior Computer is a disk copier and track 0 read/write utility. You can write the modified DOS onto track 0 of your diskette with the track 0 read/write utility program. But first you must initialize your diskette. Remove the Ohio diskette from the drive and insert your diskette, then close the door of the drive. Table 3 shows the next steps: type the word IN. This command initializes the diskette. The system asks: 'ARE YOU SURE? '. Your reply is Y(ES). The drive ceases clicking when the formatting in-

Table 8. Adapting an OS-65D V3.3 diskette. BASIC is loaded via a jump to the DOS command interpreter (GO 2276) and copied with the utility program BEXEC*. Finally a 'trial run' is executed to ensure that the diskette was adapted correctly.

```
<RST>
<AD>    FFE8
<GO>
```

```
(RUBOUT)
```

```
*TRACK 0&1*
```

```
<AD>    2217
<DA>    4C
<+>     40
<+>     22
```

```
<AD>    2245
<DA>    4C
<+>     76
<+>     22
```

```
<AD>    2285
<DA>    8E
<+>     C6
<+>     2A
<+>     4C
<+>     B3
<+>     22
```

```
<AD>    2E84
<DA>    4C
<+>     B0
<+>     2E
```

```
<PC>
```

```
A*CA 0200=06,4
```

```
A*GO 0200
```

```
- TRACK ZERO READ/WRITE UTILITY -
```

```
COMMANDS:
```

```
Rnnnn - READ INTO LOCATION nnnn.
Wnnnn/gggg,p - WRITE FROM nnnn FOR p PAGES
               WITH gggg AS THE LOAD VECTOR
E - EXIT TO OS-65D
```

```
COMMAND? W2200/2200,8
```

```
- TRACK ZERO READ/WRITE UTILITY -
```

```
COMMANDS:
```

```
Rnnnn - READ INTO LOCATION nnnn.
Wnnnn/gggg,p - WRITE FROM nnnn FOR p PAGES
               WITH gggg AS THE LOAD VECTOR
E - EXIT TO OS-65D
```

```
COMMAND? E
```

```
A*SA 01,1=2A00/8
```

```
A*
```

```
<RST>
<AD>    FF17
<GO>
```

```
(RUBOUT)
```

```
OS-65D V3.0
OSI 9 DIGIT BASIC
COPYRIGHT 1977 BY MICROSOFT
34177 BYTES FREE
```

```
Ok
RUN"BEXEC**"
```

```
OS-65D Tutorial disk five - Sept. 16, 1981
```

```
1 > Directory
2 > Create a new file
3 > Change a file name
4 > Delete file from diskette
5 > Create blank data diskette
6 > Create data diskette with files
7 > Create buffer space for data files
8 > Single or dual disk drive copier
9 > Enter OS-65D system
```

```
Type the number of your selection
and depress RETURN ? 1
```

```
Directory utility
Directory of which drive ?
Type A,B,C or D and depress RETURN <A> ?
```

```
Do you want to list the directory
to the printer (Yes or No) <No> ?
```

```
-- Directory --
```

```
File name Track range
```

```
-----
OS65D3      0 - 13
BEXEC*      14 - 16
COPIER      17 - 18
CHANGE      19 - 20
CREATE      21 - 22
DELETE      23 - 23
DIR         24 - 24
RANLST      25 - 26
RENAME      27 - 27
SECDIR      28 - 28
SEQLIST     29 - 30
TRACE       31 - 31
ZERO        32 - 33
ASAMPL      34 - 34
```

```
ATNENB      35 - 35
COLORS      36 - 36
MODEM       37 - 38
COMPAR      39 - 39
46 Entries free out of 64
```

```
Depress RETURN to continue ?
```

```
<RST>
<AD>    FFE8
<GO>
```

```
(RUBOUT)
```

```
*TRACK 0&1*
```

```
<PC>
```

```
A*GO 2276
```

```
OS-65D V3.0
OSI 9 DIGIT BASIC
COPYRIGHT 1977 BY MICROSOFT
33921 BYTES FREE
```

```
Ok
RUN"BEXEC**"
```

```
OS-65D Tutorial disk five - Sept. 16, 1981
```

```
1 > Directory
2 > Create a new file
3 > Change a file name
4 > Delete file from diskette
5 > Create blank data diskette
6 > Create data diskette with files
7 > Create buffer space for data files
8 > Single or dual disk drive copier
9 > Enter OS-65D system
```

```
Type the number of your selection
and depress RETURN ? 8
```

```
- Diskette copier -
```

```
Copy from which drive (A/B/C/D) ? A
```

```
Copy to which drive (A/B/C/D) ? A
```

```
What is the last track to be copied (Inclusive) <0-39> ? 39
```

```
Are you ready to start copying (Y/N) ? Y
```

```
Insert master diskette -- press <RETURN> ?
```

```
Reading --
```

```
Insert blank diskette -- press <RETURN> ?
```

```
Initializing --
```

```
Track 01 - 01/08
Track 02 - 01/08
Track 03 - 01/08
Track 04 - 01/08
Track 05 - 01/08
Track 06 - 01/01 - 02/01 - 03/01 - 04/02
Track 07 - 01/08
Track 08 - 01/08
Track 09 - 01/08
Track 10 - 01/08
Track 11 - 01/01 - 02/01 - 03/01 - 04/01 - 05/01 - 06/01 - 07/01
Track 12 - 01/01 - 02/01 - 03/01 - 04/01
Track 13 - 01/08
Track 14 - 01/08
```

```
Insert master diskette -- press <RETURN> ?
```

```
Reading --
```

```
Insert blank diskette -- press <RETURN> ?
```

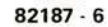
```
Track 15 - 01/08
Track 16 - 01/08
Track 17 - 01/08
Track 18 - 01/08
Track 19 - 01/08
Track 20 - 01/08
Track 21 - 01/08
Track 22 - 01/08
Track 23 - 01/08
Track 24 - 01/08
Track 25 - 01/08
Track 26 - 01/08
Track 27 - 01/08
Track 28 - 01/08
```

```
Insert master diskette -- press <RETURN> ?
```

```
Reading --
```

```
Insert blank diskette -- press <RETURN> ?
```

```
Track 29 - 01/08
Track 30 - 01/08
Track 31 - 01/08
Track 32 - 01/08
Track 33 - 01/08
Track 34 - 01/08
Track 35 - 01/08
Track 36 - 01/08
Track 37 - 01/08
Track 38 - 01/08
Track 39 - 01/05 - 02/02
```



It is considerably easier to adapt V3.1 to the Junior Computer when two

drives are connected. Modify and copy track 0 and track 1 in accordance with Table 2 and Table 3. When these two tracks are both on the diskette, remove your diskette from drive A and mount it in drive B. Now mount the Ohio diskette in drive A. Proceed according to Table 7. The computer now automatically copies all tracks from the Ohio diskette in drive A onto your diskette in drive B, beginning with track 2 and ending with track 32. After this procedure your diskette is adapted to the Junior Computer with the Ohio software.

Adapting an OS-65D V3.3 diskette

The V3.3 diskette is much easier to adapt than the V3.1 diskette. Only one drive is required. Table 8 shows how to proceed. First we load track 0 and track 1 into the computer. Since the DOS is located in the memory, we load the BASIC interpreter with the command GO 2276. The prompt OK shows that the interpreter is loaded. Then we load the utility program BEXEC* with the command:

RUN 'BEXEC*'

The computer prints a menu of options and we select the option '8'. With this option we are selecting the copier. When all the tracks have been copied, some modifications must be made to the disk operating system. Press the <RST> key again and start the program at address \$FFE8. Tracks 0 and 1 are then loaded into the computer again. However, memory area \$2200...22FF is no longer overwritten by the BASIC interpreter. Implement the modifications at addresses \$2217, \$2245, \$2285 and \$2E84 in accordance with table 8. Once these modifications have been implemented, load the track 0 read/write program from track 6, sector 4. Write the modified DOS onto track 0 of the diskette again using this program. The command SA 01,1=2A00/8 saves eight pages of DOS software on track 1. You now have a V3.3 diskette which is adapted to the Junior Computer. You can start BASIC at address \$FF17. The DOS and BASIC interpreter are automatically loaded into the Junior Computer after you have pressed the (RUBOUT) key — see Table 8.

The DOS command interpreter

As already mentioned, the disk operating system has its own command interpreter. We shall now examine the most important commands. If a command is not entered correctly an error message is issued.

Whenever the computer has printed the DOS prompt A* or B* you can enter a DOS command after the prompt. The computer only recognizes the first two upper-case letters of each command. With a SAVE command, for example,

7

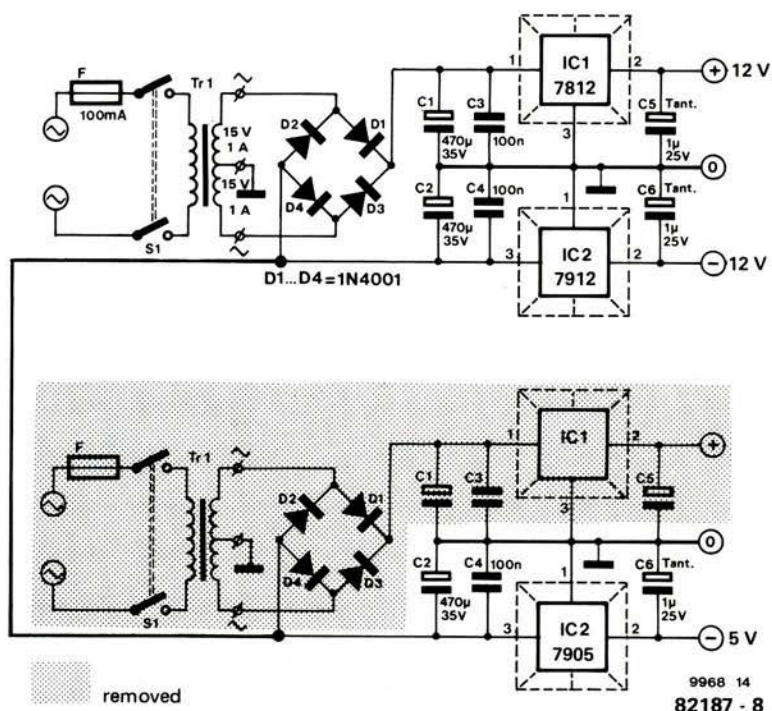


Figure 7. The power supplies for the dynamic RAMs and the printer interface can also be built quite simply with Elektor printed circuit boards (EPS 9968-5a).

it makes no difference whether you enter SA, SAV or SAVE.

Command AS or ASM

The computer loads the Assembler and Extended Monitor from the currently selected drive. Once this program has been loaded in machine language, a jump is made to the cold-start entry of the Assembler. The Assembler is linked with a line-oriented editor.

Command EM

The computer loads the Assembler and Extended Monitor from the currently selected drive. Once this program has been loaded in machine language, a jump is made to the Extended Monitor. This is a program with which a machine language program can easily be checked. The Extended Monitor has its own command interpreter. The most important commands are:

ISTRING

Send 'STRING' to the DOS command interpreter as a command.

aNNNN

Open the memory cell with the address NNNN for execution of the following subcommands:

- (LF) open the next memory cell
- (CR) close the currently addressed memory cell

(D)(D) write the data into the currently addressed memory cell

('') print the ASCII character of the currently addressed memory cell

(/) prepare the currently addressed memory cell for data input

(') open the previously addressed memory cell for data input

BN,LLLL

Set the break point with number N at address LLLL. The numeric range of the break point number N is 1...8.

EN

Eliminate the break point with number N.

A

Print the accumulator contents resulting from the last break point.

C

Start the program after the last break point.

DNNNN,MMMM

Make a memory dump from address NNNN up to and excluding address MMMM.

EX

Leave the Extended Monitor and return to DOS.

FNNNN,MMMM=DD

Fill the memory area beginning with address MMMM and ending with address MMMM-1 with the data DD.

GNNNN
Jump to address NNNN and process the program there.

HNNNN,MMMM(OP)
Call of the hexadecimal arithmetic unit. The arithmetic unit prints the result NNNN (OP) MMMM, where (OP) can be equal to +, -, *, /. Addition, subtraction, multiplication and division of hexadecimal numbers can be executed.

MNNNN=MMMM,LLLL
Move memory area MMMM...NNNN-1 into the memory area that begins at address NNNN.

RMMMM=NNNN,LLLL
The Extended Monitor has a so-called relocater. This allows machine language programs to be moved to another memory area. The computer performs the 'relocating' of all absolute addresses: 'Relocate' the memory area between NNNN...LLLL-1 into the memory area that begins at address MMMM.

The Extended Monitor has some other commands which you can find in the Ohio Manual.

Command BA
The command loads the BASIC interpreter from the currently selected drive. Once the BASIC interpreter has been loaded, a jump is made to the cold-start entry. The interpreter provides information regarding the number of free memory locations in the system and responds with the prompt 'OK'.

Command CA NNNN=TT,S or CALL NNN=TT,S
Load the data from track TT, sector S of the diskette into the computer. Store the data in memory area NNNN... The numeric range for TT is 1...39 and the numeric range for S is 1...8.

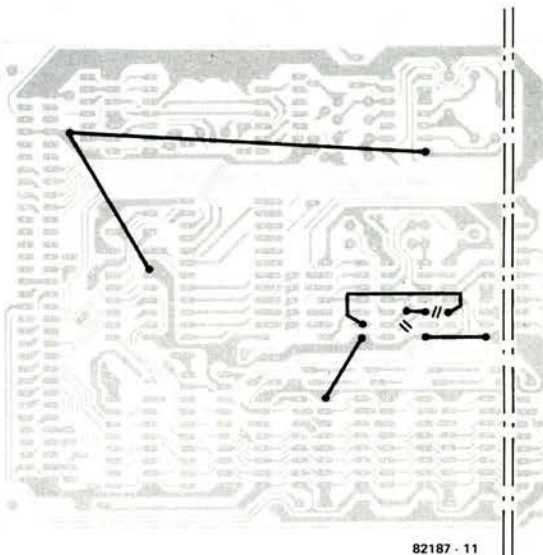
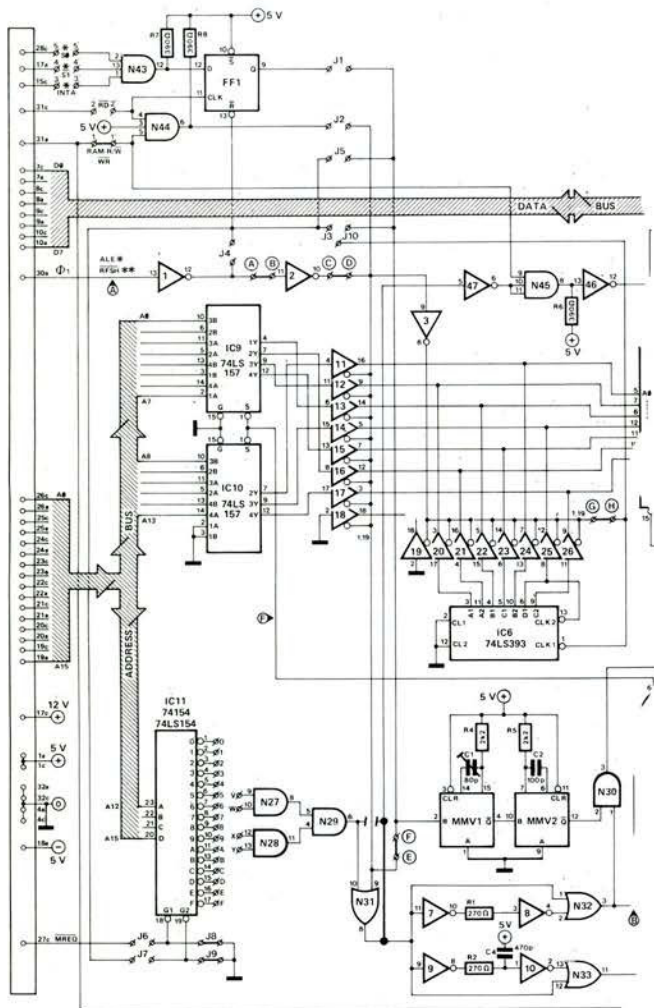
Command SA TT,S=NNNN/8PP or SAVE TT,S=NNNN/P
Save the data in memory area NNNN... for P pages onto the diskette. The track number is TT, the sector number is S and the number of pages to be written

onto the diskette is P. The numeric range for TT is 01...39, for S it is 1...8 and for P it is also 1...8.

Command DI TT or DIR TT
This command allows a check of the numbers of sectors on track TT to be made. The numeric range of TT is 01...39.

Command IN or INIT
The command IN initializes a new diskette on which no data have been written yet. If you wish to erase a recorded diskette, you can do so with the IN command.

Command IN TT:
This initializes track TT only.



82187 - 11

Figure 8. These modifications contribute greatly to the proper functioning of the dynamic RAM card.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
FC00:	A9	1E	8D	83	FA	A9	01	85	FD	A2	FF	9A	78	D8	20	7C	FDF0: 03 8E 59 FA 2C 80 FA 30 FB 20 4F FE 4E 5F FA 6E
FC10:	FC	D0	FB	20	7C	FC	F0	FB	20	7C	FC	F0	F6	20	E1	FC	FE00: 5E FA AD 5E FA 8D 5C FA AD 5F FA 8D 5D FA A2 08
FC20:	C9	13	D0	07	A2	FF	9A	D8	6C	FA	00	C9	10	D0	06	A9	FE10: 20 72 FE 20 2B FE C9 7F D0 B5 60 2C 80 FA 30 FB
FC30:	01	85	FD	D0	14	C9	11	D0	06	A9	00	85	FD	F0	0A	C9	FE20: 8E 61 FA A2 08 20 72 FE 20 81 FE 2C 80 FA 10 09
FC40:	12	D0	09	E6	FA	D0	02	E6	F3	4C	0D	FC	C9	14	D0	F3	FE30: 38 6E 62 FA CA D0 F1 F0 07 18 6E 62 FA CA D0 E8
FC50:	4C	FD	FE	C9	15	10	F2	85	FF	A4	FD	D0	0D	B1	FA	FA	FE40: 20 81 FE AD 62 FA 29 7F 8D 63 23 AE 61 FA 60 18
FC60:	0A	0A	0A	05	FF	91	FA	4C	49	FC	A2	04	06	FA	26	FB	FE50: AD 5A FA 69 01 8D 5A FA AD 5B FA 69 00 8D 5B FA
FC70:	CA	D0	F9	A5	FA	05	FF	85	FA	4C	49	FC	A0	00	B1	FA	FE60: 2C 80 FA 10 EA AD 5A FA 8D 5E FA AD 5B FA 8D 5F
FC80:	85	F9	A9	7F	8D	81	FA	A2	08	A5	FB	20	B8	FC	A5	FA	FE70: FA 60 AD 5C FA 8D 5E FA AD 5D FA 8D 5F FA 38 B0
FC90:	20	B8	FC	A5	F9	20	B8	FC	A9	00	8D	81	FA	A0	03	A2	FE80: 0C AD 5A FA 8D 5E FA AD 5B FA 8D 5F FA 38 AD 5E
FCA0:	00	A9	FF	8E	82	FA	E8	E8	2D	80	FA	88	D0	F5	A0	06	FE90: FA E9 01 8D 5E FA AD 5F FA E9 00 8D 5F FA EA EA
FCB0:	8C	82	FA	09	80	49	FF	60	48	84	FC	4A	4A	4A	4A	20	FEA0: B0 EB 60 8E 60 FA 8D 62 FA AD 82 FA 29 40 D0 F9
FCC0:	CB	FC	68	29	0F	20	CB	FC	A4	FC	60	A8	B9	08	FD	8D	FEB0: AD 82 FA 29 FE 8D 82 FA 20 81 FE A2 07 4E 62 FA
FCD0:	80	FA	8E	82	FA	A0	FF	88	D0	FD	88	8C	82	FA	E8	E8	FEC0: 90 30 AD 82 FA 09 01 8D 82 FA 20 81 FE CA D0 ED
FCE0:	60	A2	21	A0	01	20	A1	FC	D0	07	E0	27	D0	F5	A9	15	FED0: AE 59 FA AD 82 FA 09 01 8D 82 FA 20 81 FE CA D0
FCF0:	60	A0	FF	0A	B0	03	C8	10	FA	8A	29	0F	4A	AA	98	10	FEF0: 7C FA AD 82 FA 29 FE 8D 82 FA 18 90 CD 20 03 FF
FD00:	03	18	69	07	CA	D0	FA	60	40	79	24	30	19	12	02	78	FF00: 4C 51 2A A9 27 8D 82 FA A9 00 8D 80 FA A9 7F 8D
FD10:	00	10	08	03	46	21	06	0E	20	1E	FD	6C	FD	00	20	28	FF10: 81 FA 4A 8D 83 FA 60 A9 2E 8D 7C FA A9 FF 8D 7D
FD20:	FD	20	CF	FD	20	4F	FD	60	A0	00	8C	01	C0	A9	40	8D	FF20: FA A9 00 8D 7A FA A9 FC 8D 7B FA 4C 18 FD A9 03
FD30:	00	C0	A9	04	8D	01	C0	A9	40	8D	00	C0	A2	04	8E	01	FF30: 8D 25 23 60 A9 51 8D 7C FA A9 2A 8D 7D FA A9 00
FD40:	C0	8C	03	C0	88	8C	02	C0	8E	03	C0	8C	02	C0	60	A9	FF40: 8D 7A FA A9 FC 8D 7B FA 4C 18 FD 6C 7A FA 6C 7E
FD50:	FB	D0	F9	A9	02	2C	00	C0	F0	1C	A9	FF	8D	02	C0	20	FF50: FA 20 1E FD A9 28 8D A3 26 A9 01 8D 5E 26 20 BC
FD60:	CE	FD	29	F7	8D	02	C0	20	CE	FD	09	08	8D	02	C0	A2	FF60: 26 A9 2A 85 FF 20 54 27 86 FE 20 67 29 A9 01 8D
FD70:	18	20	BA	FD	F0	DD	A2	7F	8E	02	C0	2D	D7	FC	AD	00	FF70: 21 23 8D 22 23 8D C6 2A 20 C6 29 A9 1A 8D 01 23
FD80:	C0	30	FB	AD	00	C0	10	FB	A9	03	8D	10	C0	A9	58	8D	FF80: 8D 03 23 A9 A2 8D 11 23 8D 12 23 8D 14 23 60 EE 5E
FD90:	10	C0	20	C5	FD	85	FE	AA	20	C5	FD	85	FD	20	C5	FD	FF90: 8D 04 23 8D 12 23 8D 14 23 60 EE 5E 26 A9 06 20
FDA0:	85	FF	A0	00	20	C5	FD	91	FD	C8	D0	F8	E6	FE	C6	FF	FFA0: BC 26 20 67 29 EE 5E 26 A9 00 85 FE 85 FF 20 67
FDB0:	D0	F2	86	FE	A9	FF	8D	02	C0	60	A0	F8	88	D0	FD	55	FFB0: 29 A9 01 8D 5E 26 A9 13 20 BC 26 A9 32 85 FF A9
FDC0:	FF	CA	D0	F6	60	AD	10	C0	4A	90	FA	AD	11	C0	60	D8	FFC0: 74 85 FE 20 54 27 20 67 29 20 61 27 20 73 2D 0F
FDD0:	78	A9	67	8D	82	FA	A9	C0	8D	80	FA	A2	FC	8E	5A	FA	FFD0: 0A 2A 54 52 41 43 4B 20 30 26 31 2A 0D 0A 00 40
FDE0:	A2	FF	8E	5B	FA	EA	A9	7F	8D	81	FA	4A	8D	83	FA	A2	FFE0: 0F 2A 54 52 41 43 4B 20 30 26 31 2A 0D 0A 00 40
																	FFF0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF

Table 9. Hex dump listing for the bootstrap EPROM ESS515. The source listing is available as part of the documentation.

Command SE X or SELECT X

One of four drives can be selected with this command. The computer only operates with one selected drive. X = A, AB, C or D.

Command LO FILE or LOAD FILE

A file with a name can be loaded into the main memory with this command. However, the name of the file must be specified in the directory of the diskette. A file name can be generated in the directory (track 12) with the CREATE utility program. Further details can be found in the manuals supplied with the diskettes. The file name must begin with an alpha character and can be 1...6 characters long.

Command PU FILE or PUT FILE

A file with a name can be written from the main memory onto the diskette with this command. Before the file can be written on the diskette the file name must be specified in the directory.

Commands PU TT and LO TT

With this command a file in the work space of the computer can be written onto the diskette or read from the diskette into the computer without a file name. However, the file in the work space must not be longer than 2 K bytes. It is advisable not to use these two commands because it is easy to overwrite existing software on

the diskette without being warned by an error message.

Command RE or RETURN

With RETURN commands you can return from the DOS interpreter to the present transient processor:

RE AS return to Assembler

RE BA return to BASIC

RE EM return to Extended Monitor

You can find other commands in the Ohio Manual. We shall now end the software adaptation and discussion of the disk operating system and consider some special hardware requirements of the DOS Junior Computer.

Hardware requirements of a 6502 DOS computer

Computers on which a disk operating system is implemented require a good power supply. The power supply chosen for the DOS Junior Computer should have sufficient reserve capacity and should have a pulse-free output. If you already have an extended Junior Computer, you can continue to use the existing power supply. It is only necessary to build a new 12 V/4 A power supply for the drive. If, however, you have not yet got a power supply you can use the circuits in figure 7. These power supplies are intended to power the entire DOS computer; they are easy to build and are reliable. The entire circuit can be accommodated on only four ELEKTOR printed circuit boards: 2-times ESS 82570 and 2-times ESS 9968-5a. Two transformers are required with the following ratings: 9...10 V/

10 A and 15 V/4.4...5 A. Both transformers should be of the toroidal type. Toroidal transformers are more expensive than the laminated type but are lighter and generate less of an interference field. The temperature rise of a toroidal transformer is also limited, even under full load.

A third transformer with 2-times 15 V/1 A provides the supply voltage for the dynamic RAM cards and the printer interface. No particular demands are made of this transformer. With the power supply circuits of figures 6 and 7 you will never have problems with the DOS Junior or any other 6502 computer, because they are generously overrated and short-proof. The power supplies can be accommodated in a flat case of the dimensions 300 mm x 70 mm x 200 mm. Conductors with a cross-section of 1.5 square millimetres should be used for the wiring.

The dynamic RAM card

Experience has shown that the dynamic RAM card 'refuses' to operate with certain 6502 processors. Neither was the card very fond of the 6809 processor so far. For this reason we have made a few modifications to the dynamic RAM card which make it extremely reliable in conjunction with any 6502/6809 system. Figure 8 shows all the modifications that we made to the circuit diagram and p.c.b..

We wish you continued enjoyment with the DOS Junior Computer which is still the lowest-cost and favourite computer for home construction, with a disk operating system.