G. de Cuyper

floppy-disk interface for the Junior

. . . and other 6502 computers

At present the floppy disk is the most significant mass storage medium for the computer. Considering the method used for recording, it is almost incredible that computer data can be stored on a simple plastic disk at such speed and with such precision. This article will point out everything that has to be taken into account before one single bit can be stored on the plastic disk. The hardware of the floppy disk interface is designed to be universal. Not only Junior Computer fans, but also the owners of a KIM, SYM, AIM-65, ACORN and other computers can use this low-cost interface to extend their computer to a real personal computer. Even an interface for connecting the EPSON printer is provided.

Computer fever broke out here in Europe about eight years ago. The first 8-bit processors were even available at a hobbyist's budget. The great change, however, came in 1976 when Shugart brought the first 8-inch drive onto the market. Until then computer data had to be stored on punched paper tape or inconveniently on magnetic tape at speeds that are considered extremely slow today. The floppy disk revolutionized rapid data interchange between the computer and an external mass storage device. Rapid paper tape readers only achieved rates of up to 15 kilobaud (baud = bit/second) at best. Paper tape punches were barely able to exceed 700 baud. Unless the user is to go to great expense with magnetic tape recording using a cassette recorder, the upper limit is 1200 baud at a tape speed of 4.75 cm/s.

But even floppy disks have their disadvantages. Although the round plastic disks do not cost more than good chrome-dioxide cassettes, the drive unit for floppy disks is fairly expensive. The computer user will have to pay about 120 to 190 pounds for a floppy disk drive. If one considers that two drives are required for convenient operation with a computer, the investment is quite substantial.

The price barrier

Since the prices for floppy disk drives are not likely to drop much lower, it was decided to save costs in another area when implementing a DOS (DOS = disk operating system) for the Junior Computer. We had to make a choice between employing a floppy disk controller and a controller using a few TTL ICs and some software. The 1771 or 1791 from Western Digital or the Motorola 6843 are suitable as controllers. These chips have the disadvantage that they cost between 17 and 35 pounds. A further disadvantage is that not much 6502 software is available on the software market for these controller ICs.

Our objective was to equip the Junior Computer with a powerful disk operating system, without forgetting the KIM, SYM and AIM-65 friends. Hardware for the floppy disk interface was not to exceed the 35 pound limit. We made the following demands of a DOS:

1. The programmer should no longer have to be concerned with absolute addresses in the computer.
2. The DOS should operate together with a Microsoft BASIC. The BASIC interpreter should understand DOS macro commands.
3. The DOS should operate with a convenient debugger. A debugger is a program which allows software to be generated in machine language and tested. It should also be possible to place break points at any locations.
4. An assembler and editor should also be provided and should understand various DOS macro commands.
5. If the programmer makes an incorrect input, the computer should enable immediate analysis of the syntax and operating errors with precise error messages.
6. There should be a lot of good and cheap software available on floppy disks for the DOS:
   - Games programs
   - Bookkeeping programs
   - Programs in BASIC and Assembler
7. The DOS must be easily adaptable to any 6502 computer.
8. The DOS must be capable of generating random files. Random files are data files on the floppy disk into which data are written and which are produced during execution of a BASIC program.

As can be seen from these requirements, we made high demands of the disk operating system. For this reason we chose an operating system that is widely used in the USA and Europe: the DOS is from Ohio Scientific and is known as the 'Ohio Scientific OS-65D Operating System'.

Ohio Scientific also supplies the popular computers 'superboard C1P, C4P and C8P'. The software developed for these computers (and there is a good deal of it) can be easily adapted to the Junior Computer and other 6502 systems by modifying a few bits in the DOS main program (called KERNEL). Two versions of Ohio's DOS are available at present:

1. OS-65D V3.1 consisting of

1

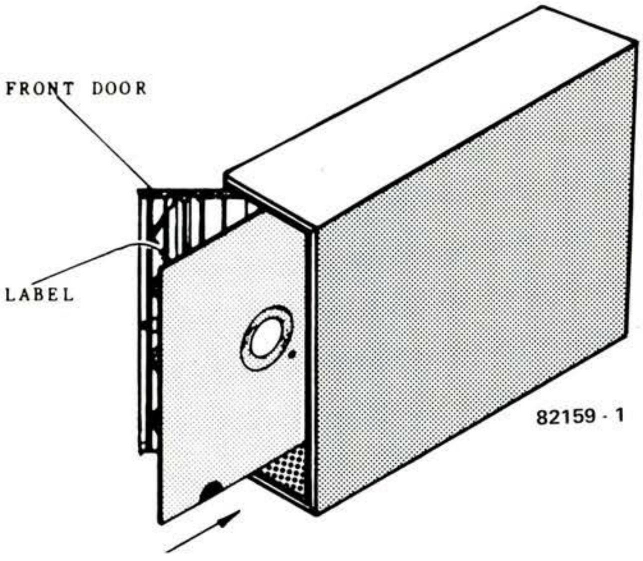


Figure 1 shows how a diskette is inserted into the drive. The label of the diskette faces the door of the drive.

— A 5-inch diskette
— A manual of approximately 75 pages.
— The price of OS-65D V3.1 including manual is approximately 30 pounds — relatively inexpensive.

2. OS-65D V3.3 consisting of
   — Five 5-inch diskettes accommodating various user-support programs (more than 17 utility programs altogether, which greatly facilitate programming).

All programs are written in BASIC and can thus be easily modified by the user if necessary.

— A fresh diskette
— A 250-page manual and detailed instructions for working with the DOS, BASIC and Assembler. Also contained are a BASIC manual and an Assembler manual.
— The price of OS-65D V3.3 including all manuals is approximately 60 pounds. Considering the expensive documentation of OS-65D V3.3, this price is certainly justified to say the least.

We adapted both versions to the Junior Computer and both versions have been working for many months without any problems and to our full satisfaction.

Before being able to work with the extensive DOS from Ohio Scientific, however, a good deal must be known about the operating system. The Ohio manuals are very well written, but one must be familiar with computer techniques to understand the manuals. For this reason we shall try to provide our readers with the necessary knowledge concerning a disk operating system, step by step. First we shall generally describe the manner in which the data are stored on a floppy disk and will then discuss the recording method of Ohio Scientific. A functional description of the mechanical aspects of a floppy disk drive will also be included.

As shown in figure 1, each floppy disk drive has a door at the front. This door must be opened to insert or remove a diskette. The door should not be closed until the deskette has been fully inserted into the drive. Otherwise there is a risk of damaging the diskette. Fitted to the door of the drive is a switch which closes a contact when the door is closed.

2

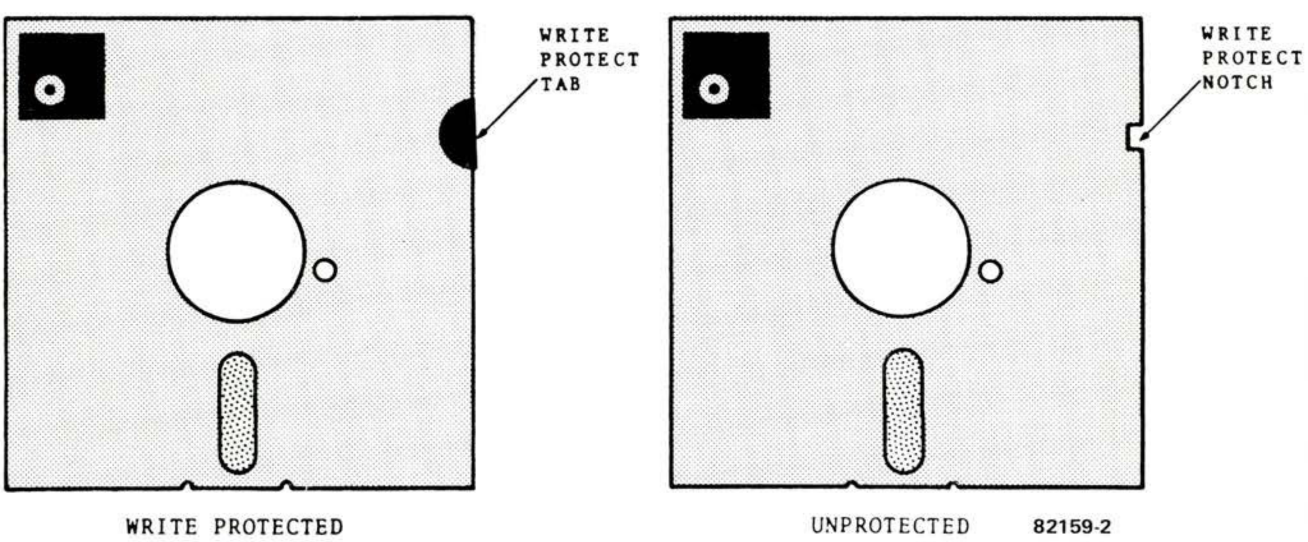
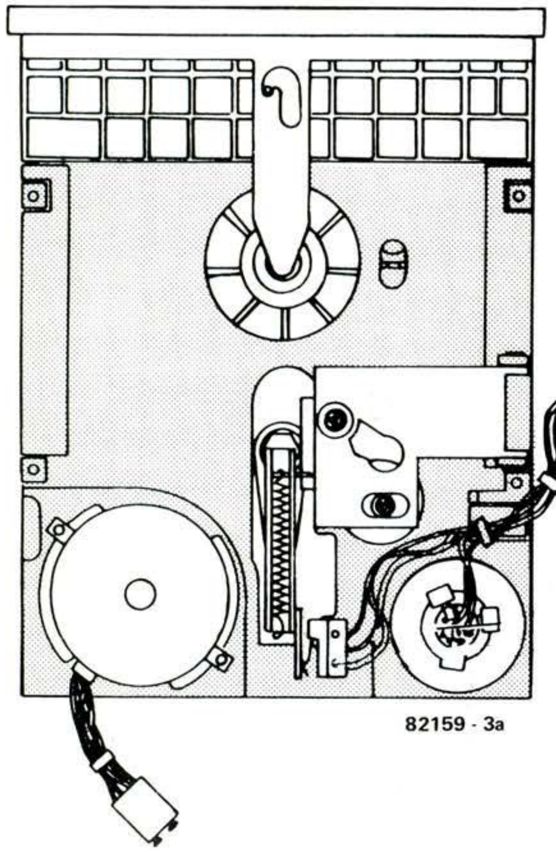


Figure 2. The diskette has a notch on one edge. If this notch is covered by a non-translucent medium, the diskette is write-protected.

3



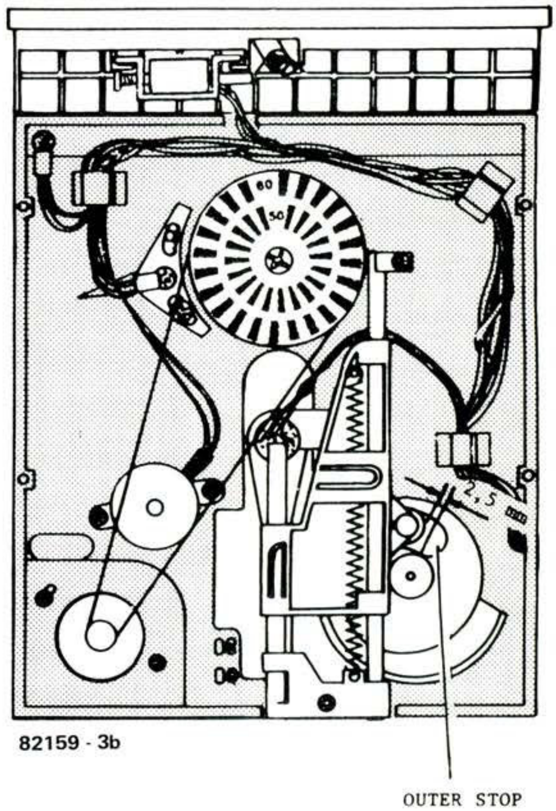


Figure 3a shows a Shugart drive and figure 3b shows a BASF drive.

4

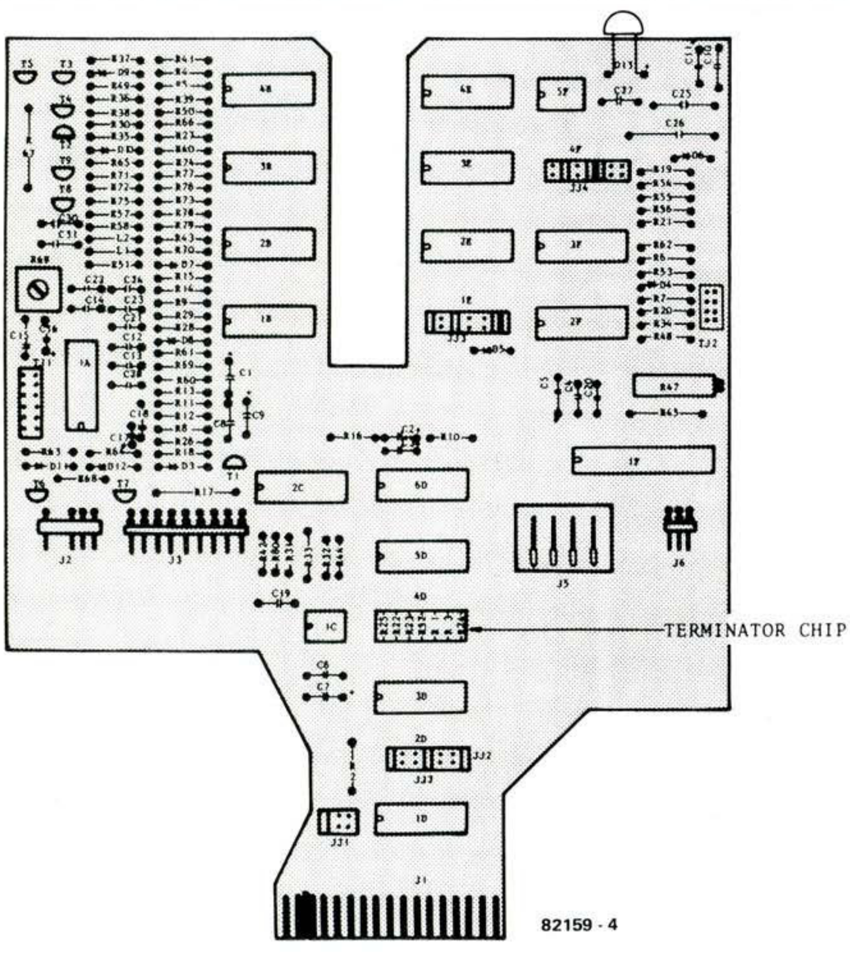


Figure 4. This is the component side of a BASF 6106 drive. The computer transmits and receives data to and from the drive via J1. Plug JJ1 is in position 'A'. The plug of JJ1 must always be in the position shown when a drive is used.

Thus the computer can only write data onto the diskette or read data from it when the door is closed.

Diskettes can be protected against accidental overwriting. As can be seen in figure 2, there is a notch in one side of the diskette. An optoelectronic device in the floppy disk drive monitors this notch to establish whether it is open or covered. If the notch is covered the diskette is protected against accidental overwriting. If the programmer attempts to write data onto a write-protected diskette the DOS issues an error message.

Any floppy disk drive can be utilized in principle. The only condition is that the input/output connector of the drive must be Shugart-compatible. Most 5 1/4-inch drives meet this requirement. We have tested the DOS both with Shugart and BASF drives. The only difference between the two drives is that the read/write head with Shugart is positioned by means of a spindle drive, whilst BASF use a helix. Figure 3a shows the mechanism of a Shugart drive and figure 3b shows that of a BASF drive.

Both drives are equipped with two motors:

— a drive motor and
— a stepper motor.

The drive motor rotates the floppy disk at a constant speed of 360 rpm. The drive motor is connected to an electronic regulator which keeps the rotational speed of the diskette constant, even in the event of load variations. The rotational speed of the diskette can be varied within certain limits on both drives. The second motor is a stepper motor which handles the positioning of the read/write head of the drive. This motor is also connected to electronic control circuitry. The control circuitry is fed with pulses by the computer. Each pulse switches the stepper motor one step further. Another line is connected between the computer and the control circuitry of the stepper motor. The potential on this line determines whether the stepper motor is to move the read/write head outwards from the interior or vice versa.

The drive chassis also contains three other electromechanical components:
As its name implies, the task of the head-load solenoid is to lower the read/write head onto the magnetic surface of the diskette. If the head-load solenoid is not activated, the read/write head is raised from the surface of the diskette by a spring. On the BASF drive the head is rigidly mounted. A felt pressure disk presses the magnetic surface of the diskette against the head.

Two optoelectronic sensors are located on the drive chassis. One of these sensors emits a pulse when the read/write head is over 'track zero' (described in more detail at the end of the article). Track zero is a special recording track on almost all floppy disk drives.
The second sensor monitors the index hole which is punched in the diskette. The index hole (see figure 2) is the absolute zero mark of the diskette or, to put it another way, the 'zero degree mark' on the round plastic disk. The index hole serves to inform the computer when the diskette has made a full revolution. Thus at 360 rpm an index pulse is emitted every 166.66 ms.

To summarize, therefore, a floppy disk drive consists of the following components:

- A stepper motor for positioning the read/write head
- A drive motor to rotate the diskette at a constant speed
- An optoelectronic sensor that checks whether the read/write head is positioned over track zero
- An optoelectronic sensor that establishes whether the diskette is write-protected
- An optoelectronic sensor that monitors the index hole and emits a pulse every full revolution
- A head-load solenoid that lowers the read/write head onto the magnetic surface of the diskette.

Clearly, a good deal of electronic circuitry is required to control all the electromechanical components on the drive chassis. Read/write amplifiers are also required for the head in the drive. These amplifiers can be compared with the recording and playback amplifiers in a conventional tape recorder. However, the read/write amplifier in the drive must be able to process frequencies of approximately 125 kHz, because the baud rate for our floppy disk interface is 125 kilobaud.

All this extensive circuitry is installed in a floppy disk drive and obviously contributes to its high cost. Usually the electronic circuitry in the drive is prealigned. There is therefore no difficulty involved in connecting a drive to the Junior Computer. Figure 4 shows the printed circuit board for a BASF drive. Only two connectors are important to the user: J1 and J5.

Connector J1 is the Shugart-compatible connector of the drive. All control signals of this connector have TTL levels. All control signals emitted by the floppy disk interface are fed to the electronic circuitry in the disk drive via J1.

Connector J5 is also Shugart-compatible and is utilized for the supply of power. The disk drive requires two voltages: 12 V/800 mA and 5 V/300 mA. Since the power consumption of a DOS computer is fairly high, we shall examine the question of power supplies in a future issue of Elektor.

Any user wishing to connect two or more drives to his computer must also

5

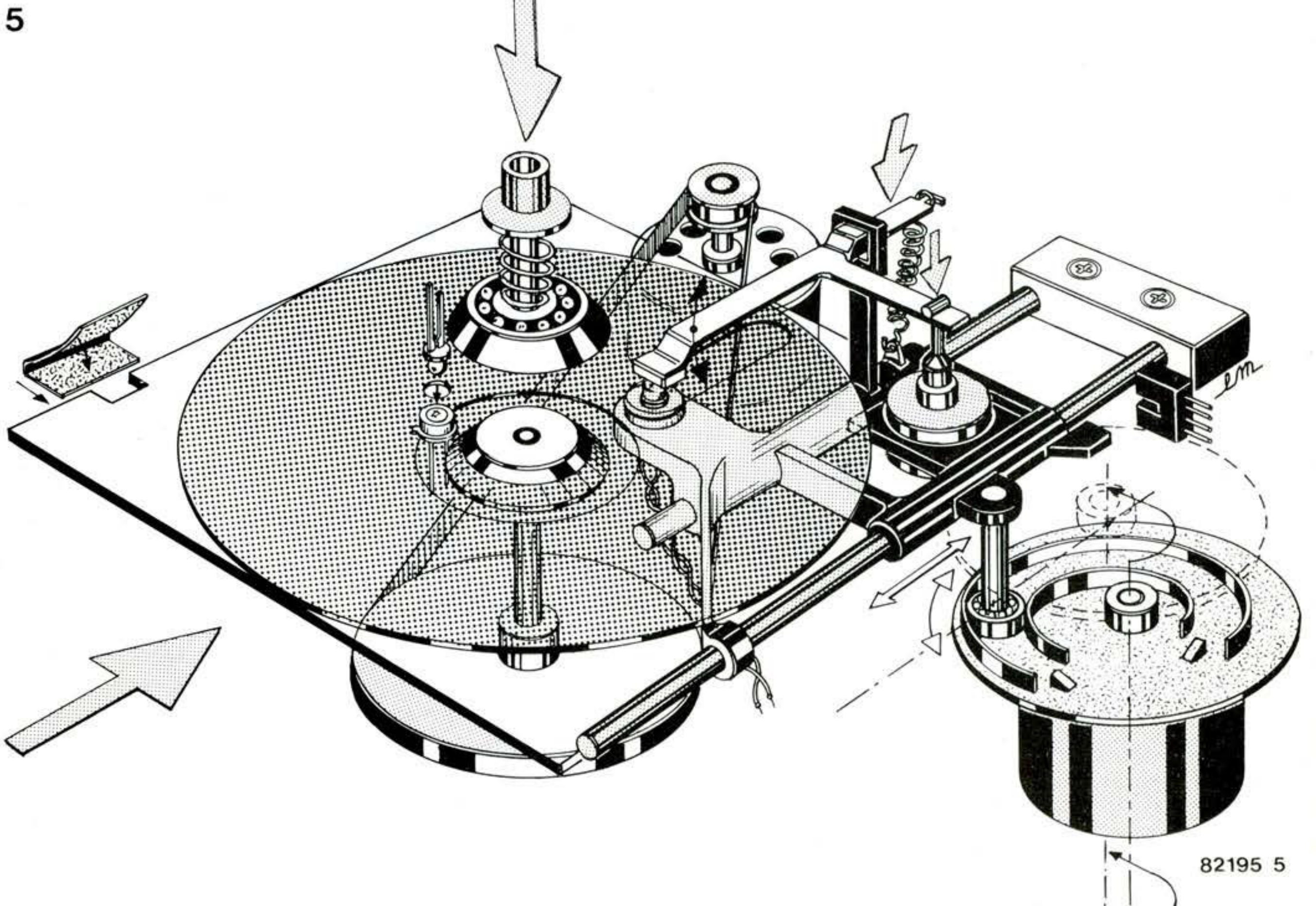


Figure 5. Principle of the mechanism in the drive. All floppy disk drive manufacturers use their own designs for the mechanism. For this reason we can only show the principle here.

take into account connector JJ1 and the terminator chip:
The terminator chip contains eight pull-up resistors and is always located in the last drive, when more than one are connected. When two drives are connected to the Junior Computer, drive A (the first one) has no terminator chip whilst drive B (the last one) contains the terminator chip. If, for example, four drives are to be connected to the Junior Computer, drives A . . . C contain no terminator chips whilst drive D (the last one) contains a terminator chip.
The drive designated as A, B, C or D is selected on JJ1 with the shorting plug. Drieve A is selected with the position shown. The following plug positions are assigned to drives B and C. Drive D does not require a shorting plug because it is clearly identifiable by the terminator chip.
Resistor R69 is for fine adjustment of the read amplifier. This resistor should never be readjusted. The quality of the read signal depends on it (freedom from jitter).

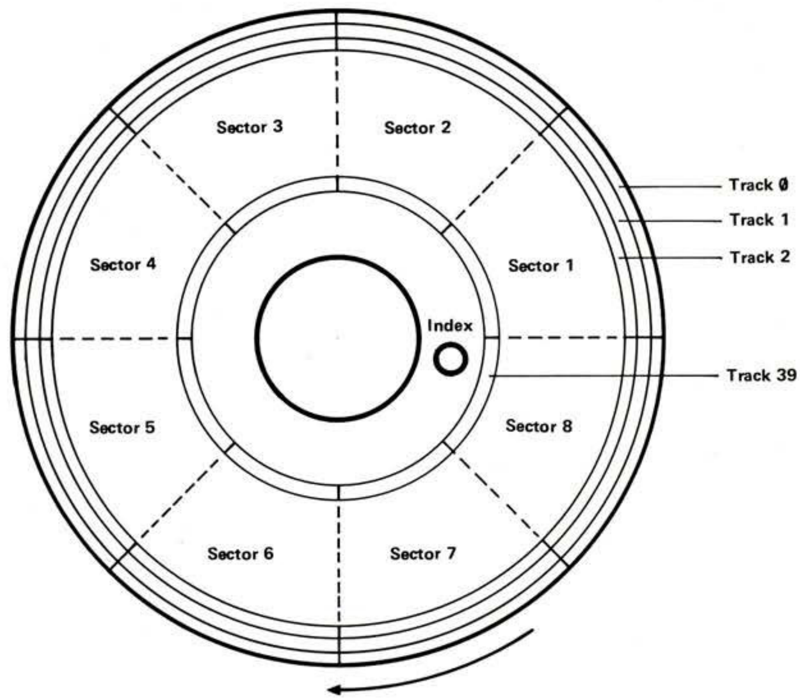
Drive mechanism

Figure 5 shows a sketch of a drive mechanism. The floppy disk is rotated by the drive motor like a record. The head makes contact with the surface of the diskette and converts the magnetic field fluctuations in the gap to an electrical signal. Recording on the diskette and playback from it ('writing and reading') are accomplished using the same principle as for a tape recorder or cassette recorder.
Since there are no grooves on the magnetic surface of the diskette, the head cannot follow a groove as on a music record. The head is therefore positioned over the desired track by a stepper motor. The stepper motor moves the head, which is mounted on a carriage, from track to track. The head can be moved over the surface of the diskette from the exterior to the interior or vice versa by means of the carriage. Since the surface of the diskette becomes worn fairly rapidly, the head is raised from the surface after every read or write operation. If the head is positioned on the surface of the diskette the term used is 'loaded head'. If it is not on the surface, this condition is an 'unloaded head'. The head can be lowered onto the surface of the diskette or raised from it by means of a spring and the head-load solenoid. If the head is continuously positioned on the surface of the diskette, the track will be destroyed after approximately 50 hours of operation. Since the head is normally only loaded for a short period, the service life of a diskette extends to several years.

Sectoring of a floppy disk

We shall now describe the way in which the data are stored on a diskette. With

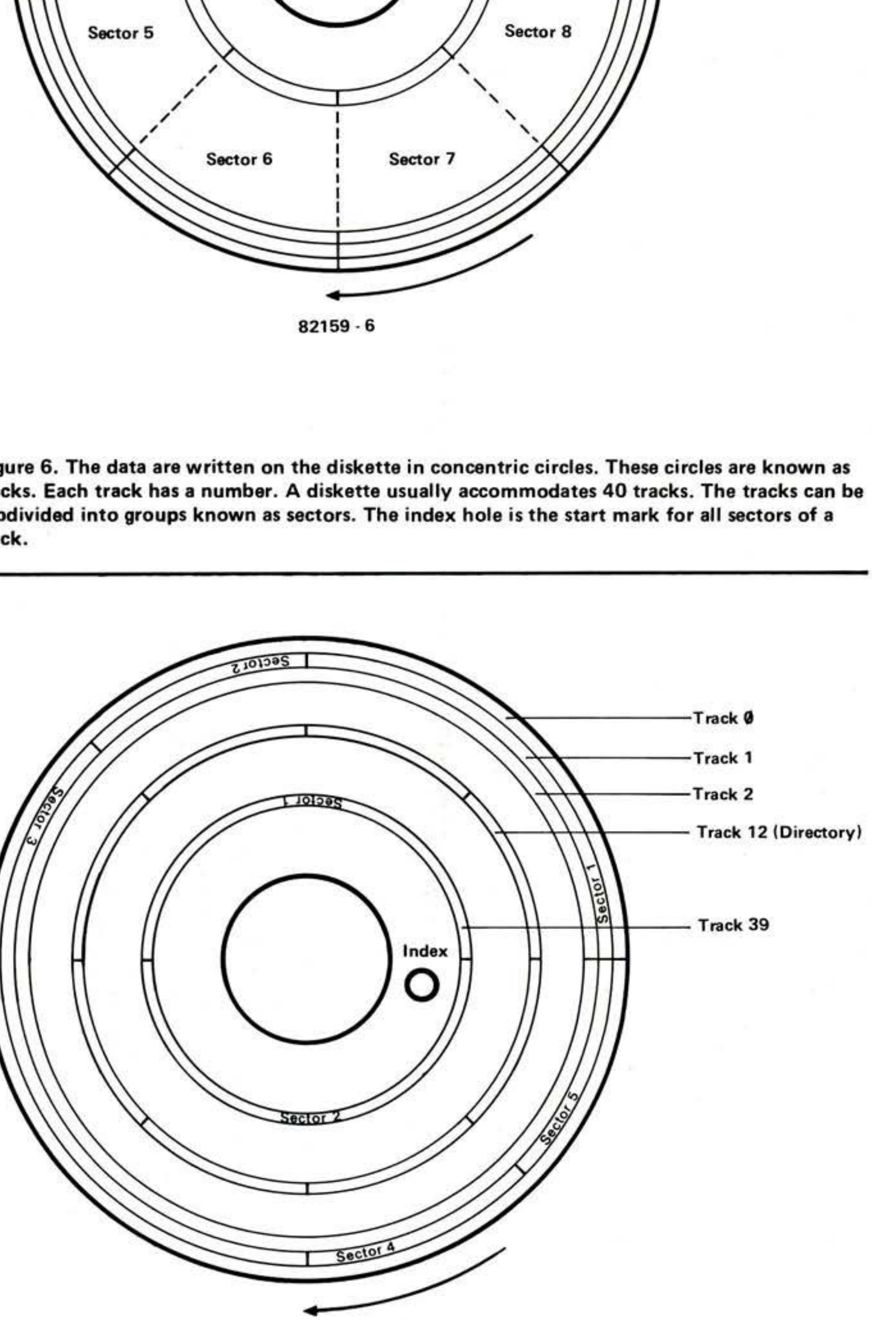
6



82159 - 6

Figure 6. The data are written on the diskette in concentric circles. These circles are known as tracks. Each track has a number. A diskette usually accommodates 40 tracks. The tracks can be subdivided into groups known as sectors. The index hole is the start mark for all sectors of a track.

7



82159 - 7

Figure 7. With soft-sectoring, the sectors can be of different lengths. In the extreme case a track consists of one or eight sectors. Track 12 is reserved for the directory with the Junior Computer.

floppy disks of the type used for the Junior Computer, the data are stored on 40 concentric rings. The width of one ring is only 0.2 mm. The outermost ring of the diskette is track zero. With most disk operating systems (including the OS-65D) this is a reference track for the other tracks on the diskette. Figure 6 shows that the diskette is further subdivided. In addition to the 40 tracks, the diskette is subdivided into sectors. For the sake of simplicity, we have used eight sectors in the example. Sector 1 always comes soon after the index hole. With OS-65D there is always a wait of 1 ms until the index pulse has decayed. Only then are the data written into the corresponding sector of the track.
There are very many formats for sectoring a diskette. The best known format is the IBM-3740 format which is not employed by Ohio Scientific. For this reason we shall not discuss the IBM format but will deal only with Ohio's own format.

The track number and sector number allow a data block written on a diskette to be clearly identified. The diskette in figure 6 has sectors of the same lengths. It is possible, however, to place sectors of different lengths on one track. The minimum data length accommodated in one sector with Ohio Scientific is a 6502 page or 256 bytes. Thus the track number and sector number are the coordinates with which a data block can be found on the diskette in fractions of a second.
Figure 7 shows the sectoring of a diskette with variable sector lengths. This format is also used by the DOS which we have adapted to the Junior Computer. Track zero, the outermost track of the diskette, has a particular write format which will be explained later. Track 1 is subdivided into several sectors: sector 1 contains two pages, i.e. 2-times 256 bytes.
A 45-degree rotation of the diskette corresponds to a data block of 256 bytes or one page. Sector 2 on track 1 is only half as long as sector 1 and only contains 256 bytes. Sector 3 on track 1 contains 5 pages. Thus 5-times 256 bytes are stored in sector 3 on track 1.
It is possible, however, to place only one sector on a track. This is the case with track 2 in figure 7. If only one sector is placed on a track, a maximum of eight pages can be stored per track, i.e. 2048 bytes. Since specific formatting information per sector, i.e. additional bytes which require space, is written on the diskette it is advisable for safety reasons not to write more than seven sectors per track on the diskette.

Track 12 has a special function. This track holds the directory of the diskette. By means of the BASIC interpreter it is possible to store a file in the computer (for example, a BASIC program, a shopping list or a love letter). A file is created in the computer when the programmer presses keys on a terminal and the computer files the information in the memory, key by key.
For any programmer it is difficult or even impossible to make a note of the track and sector in which the program or file is stored on the diskette. For this reason the Ohio DOS offers the facility for assigning names to the programs. A program name or file name may have a maximum of six alphanumeric characters and the first character must be an alphabetic character (A . . . Z). If, for example, you have written a BASIC program for calculating a circle and wish to store this program on a diskette, you can assign the program a name. You could use the name 'CIRCAL' for instance, as an abbreviation for circle calculation. You write the name 'CIRCAL' onto the diskette quite simply by typing:

DISK! 'PUT CIRCAL'.

The computer then 'puts' the program on the diskette. The inverse of the PUT command is the LOAD command:

DISK! 'LOAD CIRCAL'.

This causes the file to be loaded into the computer. We shall explain the commands of the disk operating system later.
Before the computer can write a file onto the diskette or read one from it, the file name must exist in the directory. Ohio supply various system service routines on diskettes to be able to generate file names in the directory.

8

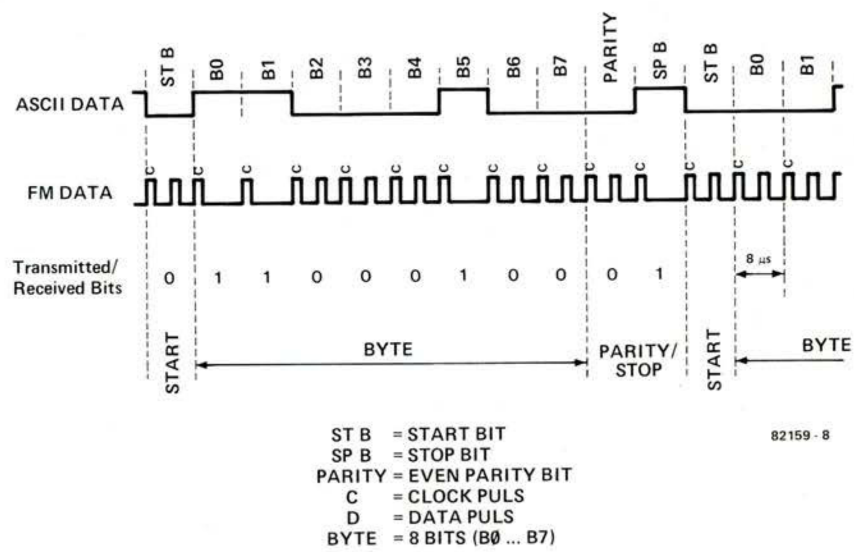
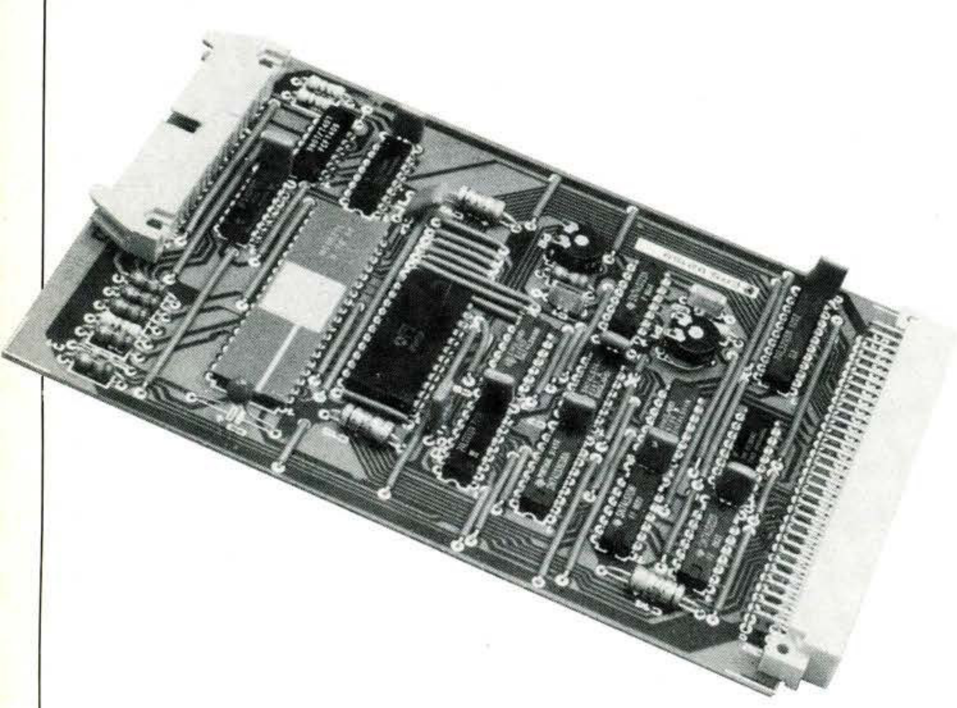


Figure 8. This is the transmission format of the floppy disk interface.

Data pulse to the floppy disk drive

At the end of this article we will demonstrate the electrical signals which the computer sends to the disk drive. Ohio use a very simple transfer format. The data are transferred asynchronously, as with the printer interface of the Junior Computer. Although the printer interface can only handle a maximum of 2400 baud, the floppy disk interface can transfer at 125000 baud. An MC 6850 ACIA, which costs less than 2 pounds, allows this high transfer rate.
The serial data delivered by the asynchronous interface adaptor (ACIA) have the following format:
- One start bit
- Eight data bits
- One even-parity bit
- One stop bit

The even-parity bit is a check bit with which any transfer errors can be traced. This bit is set when the number of set bits in the transferred byte is an even number. Unfortunately the electronic circuitry in the disk drive cannot process the serial signal of the ACIA. For this reason the serial data signal must be converted to a frequency-modulated signal. Figure 8 shows how this conversion takes place. At the start of each data bit, a narrow clock pulse of only a few hundred nanoseconds is generated. If the transmitted bit is a



logic 1, a data pulse 'D' is modulated between two clock pulses 'C'. If the transmitted bit is a logic zero no data pulse 'D' is modulated between two clock pulses 'C'. As can be seen from figure 8, the electronic circuitry of the disk drive is presented with a frequency-modulated voltage for transmitting the data. The time elapsing until a bit has been transmitted is only 8 microseconds. For receiving data from the disk drive the frequency-modulated signal must be converted back to a serial data signal. This task is performed by a data separator which is located in the floppy disk interface.

This brings us to the floppy disk interface itself and we will describe its hardware in detail. The following summary indicates everything that is needed to convert a Junior Computer or any other 6502 computer to a DOS computer:

- At least two dynamic RAM cards are needed (see ELEKTOR, April 1982).

To develop large programs, three RAM cards are required.

- A Junior Computer, consisting of the basic printed circuit board, interface card and a bus PCB with five connectors.
- A floppy disk interface card containing a few TTL ICs, an MC 6850 and an MC 6821.
- One or two (if possible) floppy disk drives which have Shugart-compatible connections:

For example the 5 1/4 inch disk drive from BASF, Shugart, TEAC, etc.

Low-cost, surplus disk drives can also be utilized.

- A power supply unit which delivers the following voltages:

+5 V/5 A
+12 V/2.5 A
+12 V/400 mA
–5 V/400 mA
–12V/400 mA

Hardware of the floppy disk interface

A brief study of the circuit diagram of the floppy disk drive (figure 9) shows that only standard, commercially available components have been utilized. We think we have reason to be pleased with this circuitry: this universal floppy disk interface is the lowest-cost interface available on the market at present. All KIM-1, AIM-65 and SYM owners can upgrade their computers from cassette to floppy disk system. However, before the floppy disk interface can be connected to the computer the user should know how the hardware functions. We shall now turn our attention to these interesting technical details.

Data transfer between computer and floppy disk drive

The principle of data transfer between the computer and floppy disk drive can be described as follows:

- The STEP and the DIR line (outputs)

Via the peripheral interface adapter (PIA) IC5 the read/write head of the disk drive is placed on the desired track. The computer emits stepper-pulses via PB3 which are matched to the electronic circuitry of the drive by driver N18. The read/write head is shifted one step inwards or outwards with each pulse. PB2 of the PIA (IC5) and N19 generate the DIR signal. The logical level on the DIR line determines whether the stepper pulses shift the read/write head outwards from the interior or vice versa.

- The TRO line (input)

The TRO line is an acknowledge line from the drive to the computer. The logic level on this line indicates whether the read/write head is placed over track zero.

- The INDEX line (input)

The Index line is an acknowledge line from the drive to the computer. As explained at the beginning of this article, the index hole on the diskette is a zero mark for a soft-sectored diskette. Whenever the index hole passes a light barrier in the drive a pulse is produced on the Index line.

- The WR.PROT line (input)

The logic level on the WR.PROT line informs the computer whether it is allowed to write on the diskette in the selected drive, or whether the diskette is write-protected. The computer only writes on the diskette when the WR.PROT line is inactive:

- WRITE line (output)

The WRITE line switches the electronic circuitry in the disk drive from the read mode to the write mode. Before this line become active, the computer checks via the WR.PROT line whether the diskette is write-protected. If this is the case, the WRITE line can never become active.

- The SEL1, SEL2, SEL3 and SEL4 lines (outputs)

The computer selects one of four drives via the SEL lines. Normally only SEL1 and SEL2 lines are utilized. Line SEL1 controls drive A and line SEL2 controls drive B. When Ohio software is used a floppy disk drive must always be connected to SEL1.

- The SIDE SEL line (output)

The SIDE SEL line is not used with the Junior Computer and is intended for later extensions. Special drives containing two read/write heads can be controlled via this line. These drives can write on both sides of a diskette and read back the stored information from both sides.

- The WDA (output) and RDA (input) lines

The computer writes the data in serial form into the electronic circuitry of the drive via the WDA line. The computer reads the serial data from the drive via the RDA line. The baud rate on these lines is 125 kilobaud. The data transfer from the computer to the drive can be compared to a simple V24/RS232 serial interface. We became acquainted with this interface in the Junior Computer 3 and 4 books. In that application the data are transferred from the computer to the ELEKTERMINAL or from the ELEKTERMINAL to the computer.

The serial data transmitted by the computer to the drive are written in parallel into the ACIA (IC11) and transmitted serially at the TxD output at a rate of 125 kilobaud.

The serial data from the ACIA cannot be directly written onto the diskette. This requires modulation of the serial data with clock pulses which initiate the start of a data bit, as shown in figure 8. Between two clock pulses, a data pulse is then modulated or not, depending on the logic level of the current data bit.

When the computer reads data from the

9

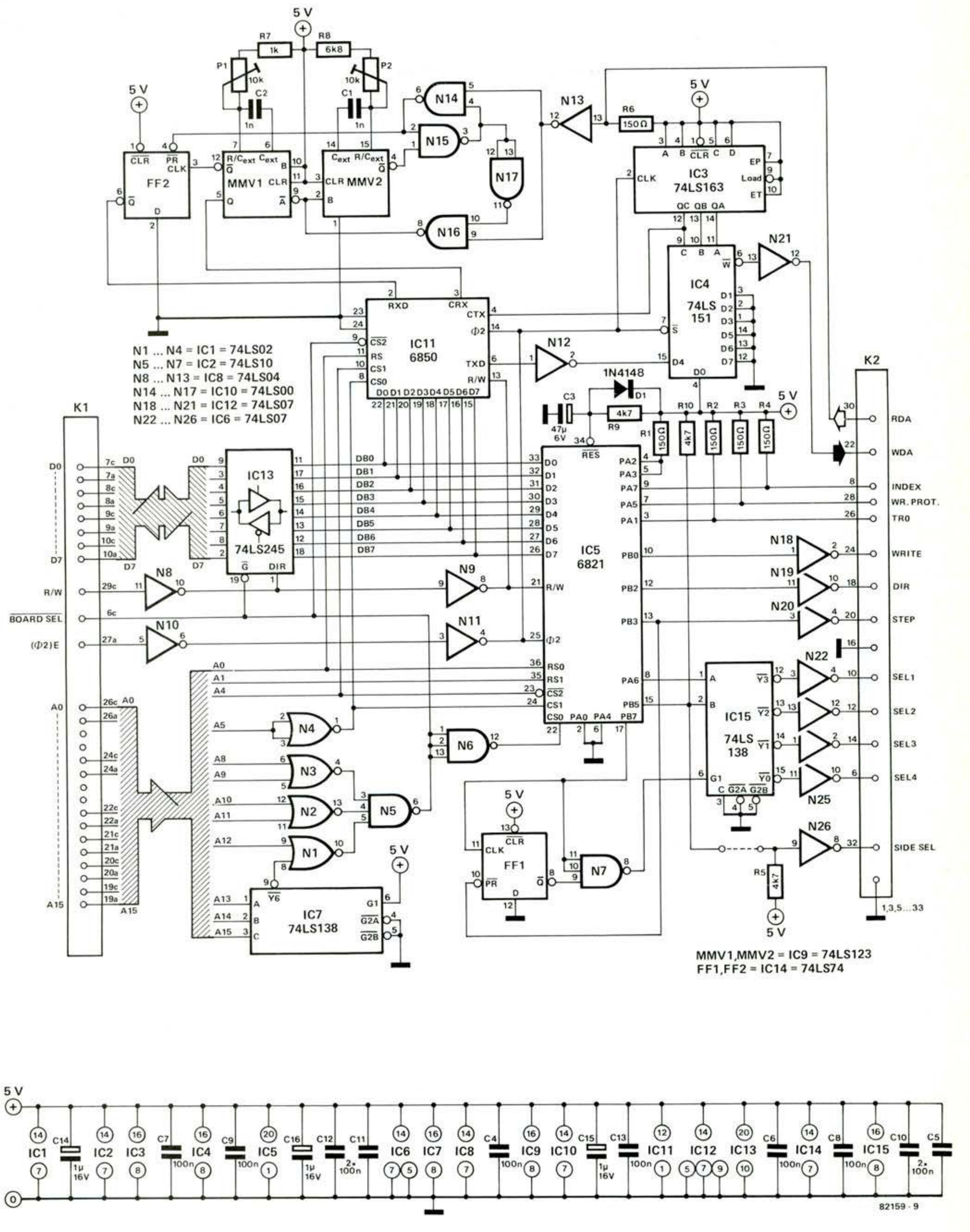


Figure 9. Circuit diagram of the floppy disk interface. Since the data interchange between the computer and drives is controlled by software, one saves the cost of an expensive floppy disk controller. Only standard components are utilised in the circuit.

10

Test Points		Signal
TJ1	1,2	Write Current Signal
	3,5	Read Signal (Preamp. Output)
	6	GND
	7,9	Read Signal (Differentiator Input)
	8	Jitter Voltage
	10	Erase Current T.P.
	11,12	Write Current T.P.
TJ2	1	DISK CHANGE FF/
	2	PWRONRESET/
	3	N.O. TRACK ZERO SWITCH
	4	IN USE- FF
	5	MOTOR ON
	6	TRACK 00
	7	GND
	8	INDEX

POTI	FUNCTION
R 47	Drive Motor Adjust
R 69	Jitter Adjust

Connector		Function
J1		Signal - Interface
J2		Read/Write - Head
J3	1,3	Head Load Solenoid
	2,4	Door Lock Solenoid
	5,6	Write Protect Phototransistor
	7,8	Index Phototransistor
	11-18	Stepper Motor
J5		DC- Connector
J6	2,4	Drive Motor
	3,5,6	Track Zero Switch

Figure 10. WRITE DATA ENCODER diagram. Here are the pulses that are emitted to the drive.

diskette the clock pulses must be separated from the data pulses. This task is handled by a data separator consisting of components N13 . . . N17, the two monostables MF1, MF2 and flip-flop FF2. The clock pulses for the ACIA are at output Q of MF1 and the serial V24/RS232 data are at output Q of FF2. The data signal that was previously in serial form can be read out of the ACIA (IC11) in parallel form by the computer via the data bus.
Since the 1/O chip used for serial data transmission between computer and drive is normally employed for V24/RS232 interfaces, a number of known characteristics are found in the transmitted data pattern:

1. Each byte to be transmitted begins with a start bit and ends with a parity bit.
2. A stop bit is located between two bytes. The stop bit is the inverse of the start bit.

Eleven bits are required to transfer one byte:
One start bit, eight data bits, one parity bit and one stop bit. If no data are being transferred, only stop bits are present (logic 1). By using pseudo-FM-encoding, 22 pulses are written onto the diskette during the transmission of one byte (= 8 bits), which only consists of logic ones (= $FF). The reason is that each bit to be transmitted consists of a clock pulse and a data pulse. The clock pulse is always present, whilst the data pulse is only present when a logic one is being transmitted (see figure 8).
This format allows eight pages of 256 bytes each to be stored on one track, i.e. 8-times 256 bytes = 2 kilobytes. Since a few tracks are required for the directory and system program (= DOS program and BASIC), only 35 tracks available to the programmer on a diskette with a total of 40 tracks. Thus approximately 2 kilobytes times 35 = 70 kilobytes can be stored on a diskette. This is more than adequate for a hobby computer system.

11

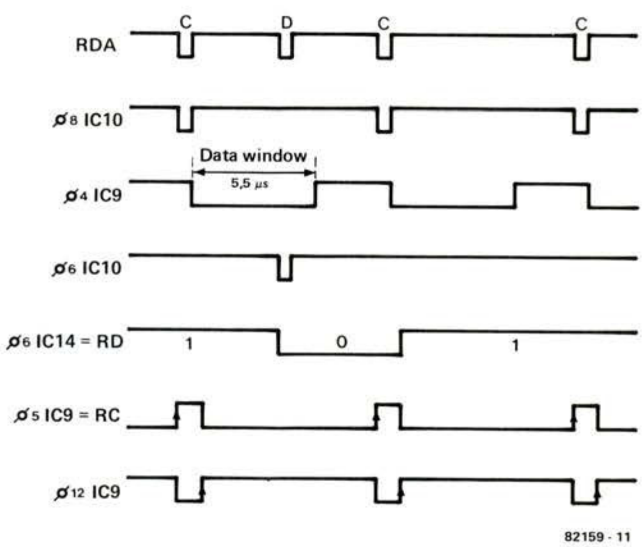


Figure 11. READ DATA SEPARATOR diagram. Here are the pulses arriving from the drive and which are then separated by the data separator.

The circuit diagram in detail

The circuit diagram of the floppy disk interface can be subdivided into several groups which we shall now examine:

a. Address decoding and data buffer
The outputs of address decoder IC7 change their states in 8-kilobyte steps. Output Y6 drives IC1 (N1 . . . N4) which decodes the rest of the address lines. If all inputs of gate N5 are logic ones, its output (pin 6) becomes a logic zero. This output always remains a logic zero between addresses $C000 . . . $C0FF. This signal is needed to activate the data bus buffer in IC13. Additionally, the output is wired to pin 6c of bus connector K1.

12

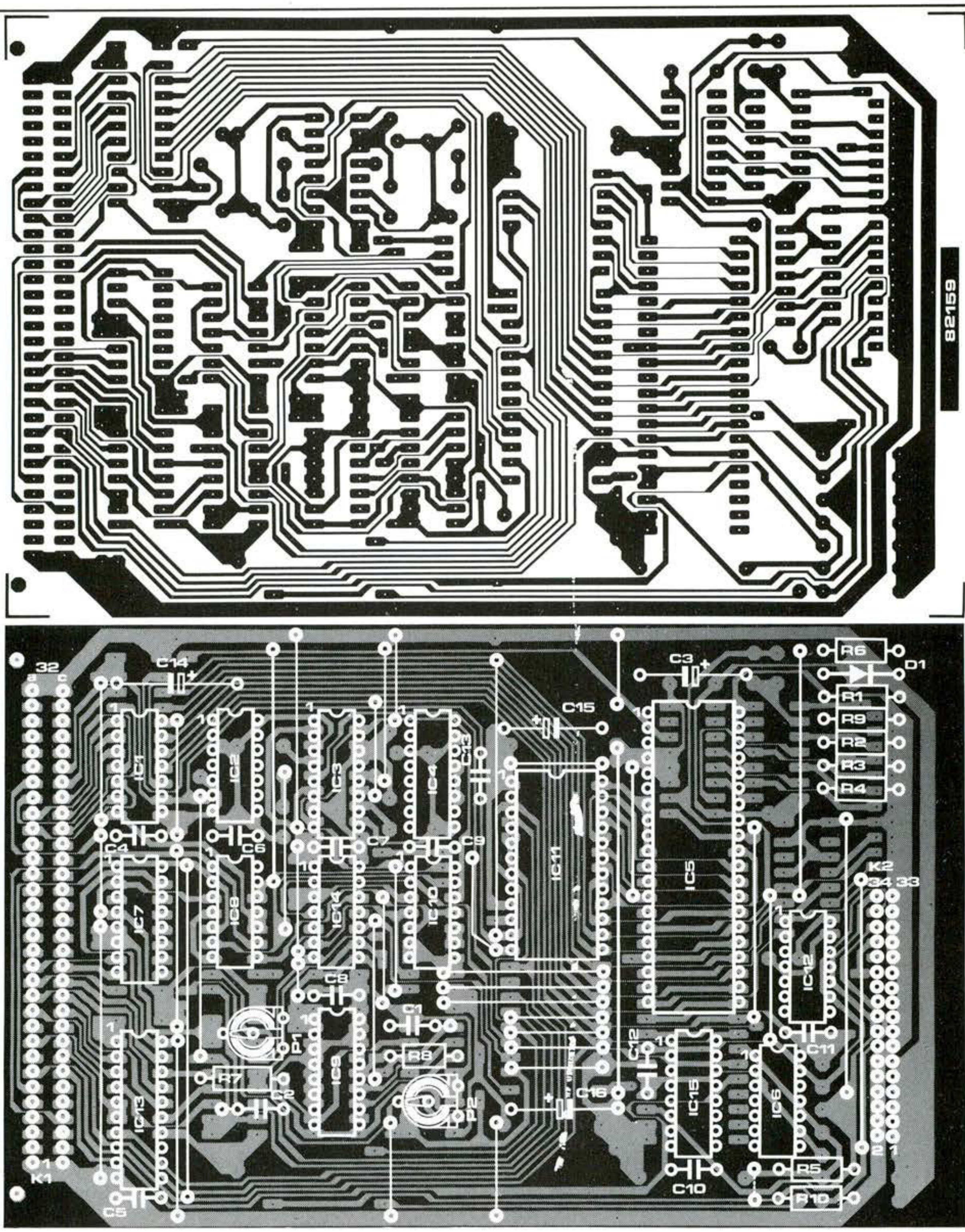


Figure 12. Layout and component overlay of the PCB of the floppy disk interface. Use great care in soldering, because some tracks are very close to each other. Good sockets should be utilised for IC5 and IC11.

Parts list

Resistors:

R1,R2,R3,R4,R6 = 150 Ω
R5,R9,R10 = 4k7
R7 = 1 k
R8 = 6k8
P1,P2 = 10 k trimmer potentiometers

Capacitors:

C1,C2 = 1 n MKT
C3 = 47 μ/6,3 V
C4 . . . C13 = 100 n
C14,C15,C16 = 1 μ/16 V

Semiconductors:

D1 = 1N4148
IC1 = 74LS02
IC2 = 74LS10
IC3 = 74LS163
IC4 = 74LS151
IC5 = 6821
IC6,IC12 = 74LS07
IC7,IC15 = 74LS138
IC8 = 74LS04
IC9 = 74LS123
IC10 = 74LS00
IC11 = 6850
IC13 = 74LS245
IC14 = 74LS74

Additionally:

64-way male connector, rows a and c fitted (Siemens)
Two 34-way female connectors for ribbon cable
Mating plugs for the above for fitting to the PCB, angled
(Molex, Amphenol, ITT-Cannon, etc.)

The direction of the data bus buffer is governed by the R/W signal which is buffered by drivers N8 and N9 on the floppy disk interface PCB. The 02 or E-signal is buffered by drivers N10 and N11.
The inverse C0XX signal activates the PIA (IC5) via the CS0 pin. The other chip select signals are connected with address lines A4 and A5, so that the PIA has a base address of $C000.

b. *Lines between drive and interface*
The outputs to the floppy disk drive are buffered by drivers N18 . . . N26. These drivers have an open collector output. The pull-up resistors are always in the last drive (terminator), as explained at the beginning of this article. The drive is also controlled by the floppy disk interface via drivers with open collectors. The pull-up resistors are R1 . . . R4 and R6.
IC15 multiplexes lines PA6 and PB5 of the PIA, so that four drives can be operated using one 34-way cable. A small modification on the interface allows two doublesided drives to be connected to the computer. This is achieved by connecting the input of N26 to PB5. The connection between PB5 and pin 2 of IC15 must be disconnected in this case.
Multiplexer IC15 is activated via N7. The inputs of N7 are controlled by the head-load output of the PIA (PB7) and the Q-output of FF1. FF1 is set by the step pulse and reset by the leading edge of the head-load pulse. N7 and FF1 are not absolutely necessary. We have made provision for them, however, on the interface PCB because the Ohio software for 8-inch drives has been applied and requires a separate head-load line. Mini-disk drives utilize the select lines to activate the read/write head. To prevent the head from continuously rubbing the surface of the diskette, the select line is activated by the head-load line. This ensures proper treatment of the diskette in the drive.

The port lines of the PIA

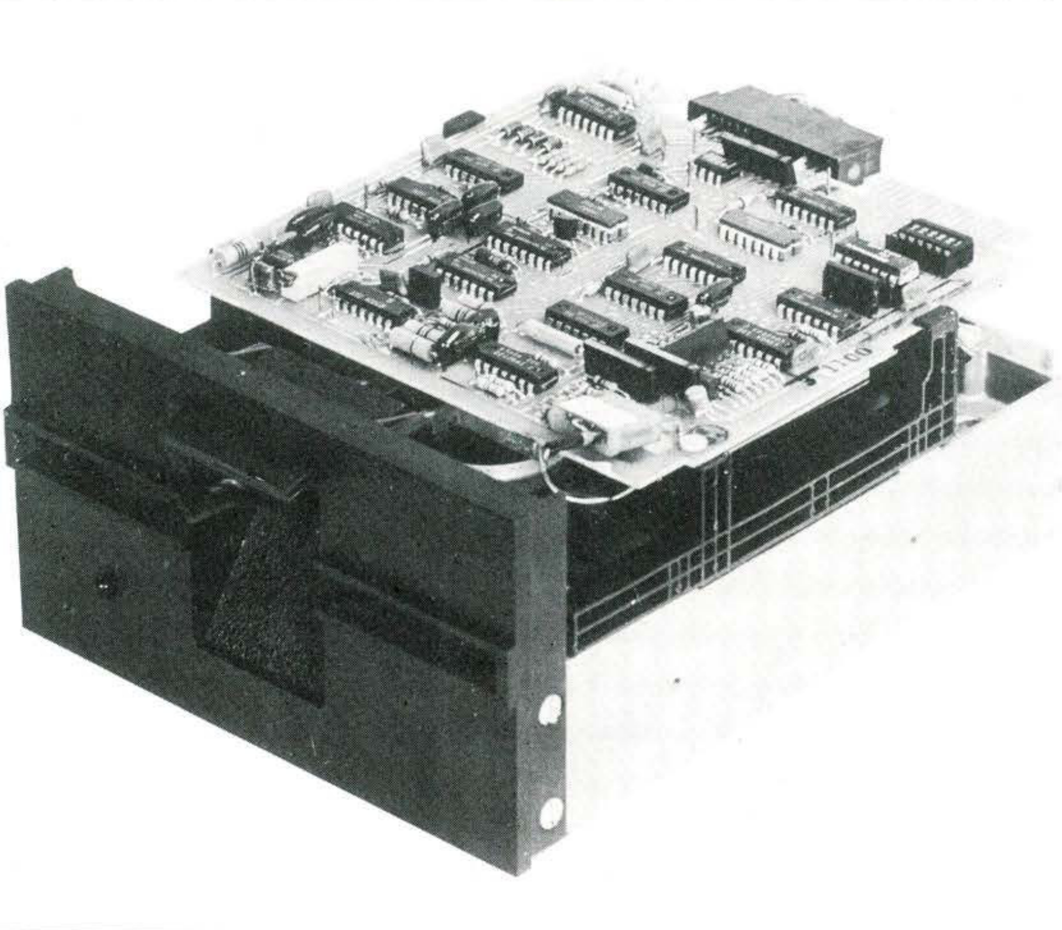
The port lines of the PIA are used as follows:

A-side:	Address $C000; Disk status Port		
PA0:	Drive 0 ready	Input	
PA1:	Track 0	Input	X
PA2:	Fault	Input	
PA3:	Free for user		
PA4:	Drive 1 ready	Input	X
PA5:	Write protect	Input	X
PA6:	Drive select L	Output	X
PA7:	Index pulse	Input	X
B-side:	Address $C002; Disk control Port		
PB0:	Write enable	Output	X
PB1:	Erase enable	Output	
PB2:	Step direction	Output	X
PB3:	Step pulse	Output	X
PB4:	Fault reset	Output	
PB5:	Drive select H	Output	X
PB6:	Low current	Output	
PB7:	Head load	Output	X

All I/O lines are active at logic zero. 'X' = I/O line used.

The electronics for data transmission

Data transmission is primarily handled by the 6850 ACIA (IC11). The computer writes the byte to be transmitted into the transmit register of the ACIA via the data bus. IC11 then shifts the word written in parallel form to the TxD output in serial form. The ACIA receives the serial from the diskette at the RxD input. The clock input for the serial receive signal is designated CRx. If a serial word is read into the ACIA the computer can read it out of the receive register in parallel. We will discuss the register structure of the ACIA in the December issue.
The data to be transmitted are presented to input D4 of data selector IC4 in inverted form (N12). All other data inputs except for D0, are grounded. The select input of IC4 selects the 'E' or 02 signal. Synchronous counter IC3 divides the clock signal by eight and sequentially addresses data selector IC4 with outputs QA, QB, QC. Output 6 of the data selector must always be a logic zero when the 'E' signal is a logic zero and when data input D4 is a logic one. This means that a pulse is always produced at address zero. This pulse is the clock pulse which is repeated every eight microseconds.
When the ACIA (IC11) transmits a logic one, TxD is a logic one and D4 of the data selector (IC4) is a logic zero. The result is as follows:



- During transmission of a logic one, no data pulse appears at the 'W' output of the data selector (IC4).
- During transmission of a logic zero, one data pulse appears between two clock pulses at the 'W' output of the data selector (IC4).
- Each clock or data pulse has a length of only 500 nanoseconds.

The signal at the 'W' output of the data selector forms the coded FM signal which is transmitted via buffer N21 to the floppy disk drive. The WRITE DATA diagram for the write encoder is shown in figure 10.

To be able to read back the data from the diskette, the clock and data pulses must be separated again. After separation, the clock pulses are utilized to shift the serial data pulses into the ACIA at a rate of 125 kilobaud. The separating of clock and data pulses is performed by a data separator consisting of N13, N14 . . . N17, MF1, MF2 and FF2.

Incoming data from the floppy disk drive are inverted by N13. The NAND port N16 is enabled by NAND N17, so that the first clock pulse can trigger both monostables MF1 and MF2. MF1 triggers on the negative edge of the clock pulse whilst MF2 triggers on the positive edge. The Q-output of MF2 should be at logic zero for about 5.5 microseconds, so that N14 is enabled and N16 is disabled. As soon as a data pulse is present between two clock pulses, flip-flop FF2 is set via N14.

The Q-output of monostable MF1 emits a clock pulse of about 1 microsecond to the CRx input of the ACIA. The leading edge of the clock pulse transfers the data bit currently being transmitted to the serial input register of the ACIA. The data bits come from the Q-output of flip-flop FF2. A data pulse on the preset input sets flip-flop FF2. The Q-output then goes to logic zero. The subsequent clock pulse transfers this zero into the ACIA. When MF1 toggles back to the stable state, it clears flip-flop FF2 via the clock input. Figure 11 shows the timing diagram of the READ DATA separator in detail.

Construction and alignment

Construction and alignment of the floppy disk interface are quite simple. All wire links should first be connected on the printed circuit board (figure 12). Since some tracks are very close to each other, soldering requires great care.

13

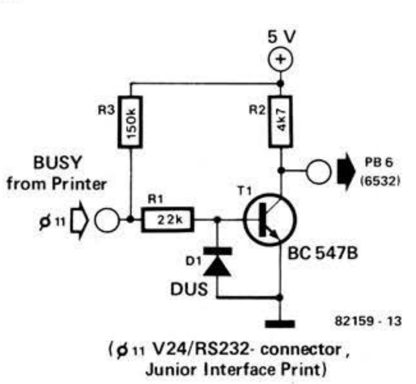


Figure 13. When connecting an EPSON printer with a serial interface, a V24/RS232-to-TTL level converter is required. This little circuit can be wired on the Junior interface PCB in self-supporting fashion. PB5 of the 6532 on the basic PCB of the Junior Computer is used as the BUSY line for the printer.

The resistors, capacitors, diode D1 and the two connectors are then fitted to the board. Trimmer potentiometers P1 and P2 are rotated to their midpoints and soldered into the board. If new ICs are inserted, they can be soldered in directly without sockets. Good sockets should always be employed for the 6850 (IC11) and the 6821 (IC5).

The floppy disk interface should normally work immediately if the two trimmers are set to their midpoints. If, however, fine alignment is still required, the procedure is as follows:

1. Remove the plug from connector K2.
2. Jumper the WDA output to the RDA input on the soldering side of the board using a wire link.
3. Align output Q of monostable MF2 to 5.5 microseconds using an oscilloscope.
4. Monostable MF1 is non-critical and can be aligned to a time of about one microsecond. However, the ACIA and the PIA must then be initialized with a short program. We will go into this in more detail when we discuss the software for the floppy disk interface in the December issue.

EPSON interface

In the Elektor laboratory the Junior Computer operates with an EPSON printer and has therefore been equipped with an interface for the EPSON dot-matrix printer. The following is a description of the necessary interface, for those readers who employ one of these printers.

For connection to the Junior Computer, the EPSON requires a serial interface adapter and not the usual Centronics interface. The commercial price for the serial interface adapter for the EPSON is approximately 23 pounds. The baud rate must be set to 1200 baud by means of the digiswitch provided on the printed circuit board. The ELEKTERMINAL should also run at this rate. The EPSON is connected in parallel to the V24/RS232 output of the ELEKTERMINAL. The EPSON uses the BUSY line to inform the computer whether data can be transmitted to the printer or not. Since cassette control is no longer necessary on the DOS computer, we have used PB5 of the 6532 on the basic PCB of the Junior Computer as the BUSY input. Relay Re2 on the Junior interface PCB can thus be discarded. The green LED (D5) can continue to be used as a transmit data indicator.

Since there is also a V24/RS232 signal level on the BUSY line, conversion to TTL level is required. Figure 13 shows a circuit which can be wired in self-supporting fashion on the component side of the Junior interface PCB.

If no EPSON printer is connected to the Junior, PB5 of the 6532 must be grounded otherwise the computer cannot transmit data. ◼