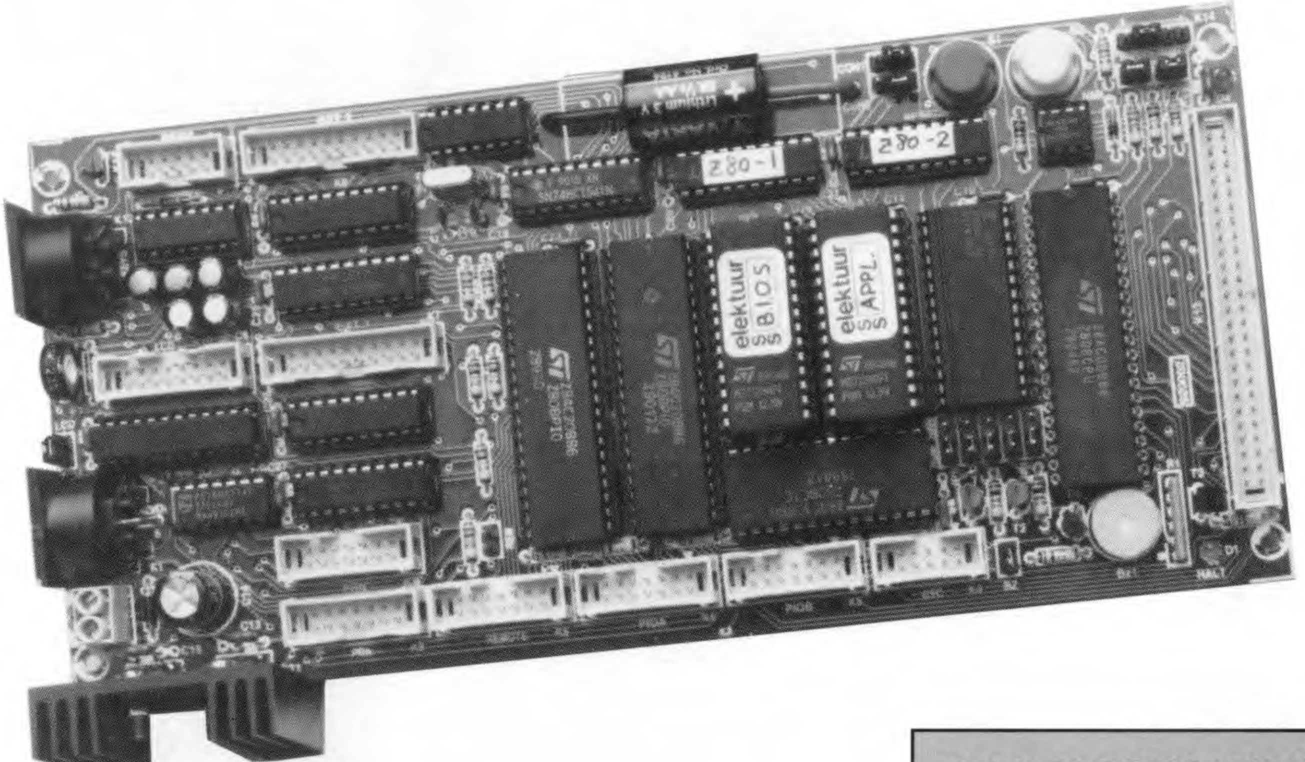
MULTI-PURPOSE Z80 CARD

PART 1: SYSTEM DESCRIPTION



Although microcontrollers are now firmly established, we are pretty sure that the present Z80 processor card will appeal to many of you. Easy to use in combination with such options as a liquid crystal display and an infra-red remote control, and supported by a BIOS that takes the hassle out of I/O programming, this is the nineties-style way of dealing with an 'evergreen' 8-bit microprocessor.

Design by A. Rietjens

THIS Z80 card is easy to use for a wide range of applications thanks to its solid base formed by a number of Z80-family ICs. Remarkably, the multi-purpose character of the card is not compromised in any way by the I/O options available. These options provide functions that normally call for the soldering iron to be switched on because you have to build them yourself, not even mentioning the effort that goes into writing suitable control software. The hardware and software proposed here ensures, in the best possible way, that non-used functions do not interfere with the ones that are used, or can be adapted easily for other purposes. An example of this is a PC-XT keyboard that may be used as an input device with the present card.

Block diagram

As may be expected, a system as outlined above requires quite a bit of electronics. To keep you from losing track at this point already, have a look at the block diagram in Fig. 1. The Z80-CPU is used alongside two Z80-PIOs and one Z80-CTC. Together with the memory, these ICs form the heart of the Z80 card, which is completed with the usual I/O decoding and memory addressing logic. The latter supports the use of bank switching, so that up to 128 Kbyte may be addressed.

As you can see in the block diagram, there is no shortage of I/O and interfacing capacity: RS232, a parallel printer and a display are all catered for. The card receives

MAIN SPECIFICATIONS

Hardware:

- Z80B-CPU running at 5 MHz
- 32 I/O lines, min. 8 and max. 16 for internal use
- 4 timers
- Up to 64 Kbyte RAM and 64 Kbyte ROM or EPROM
- 8-bit A-D/D-A converter
- Standardized RS232 serial interface; all standard baud rates between 50 and 38,400
- Centronics-compatible parallel printer interface
- Two connections for 'universal I/O interface' extension cards
- On-board watchdog
- Input device: PC/XT keyboard or RC5 infra-red receiver
- Connection for LC display with up to 2×40 characters
- On-board battery backup

Software:

- BIOS available to control and test all card functions
- BIOS is MSX-compatible
- Built-in test routines

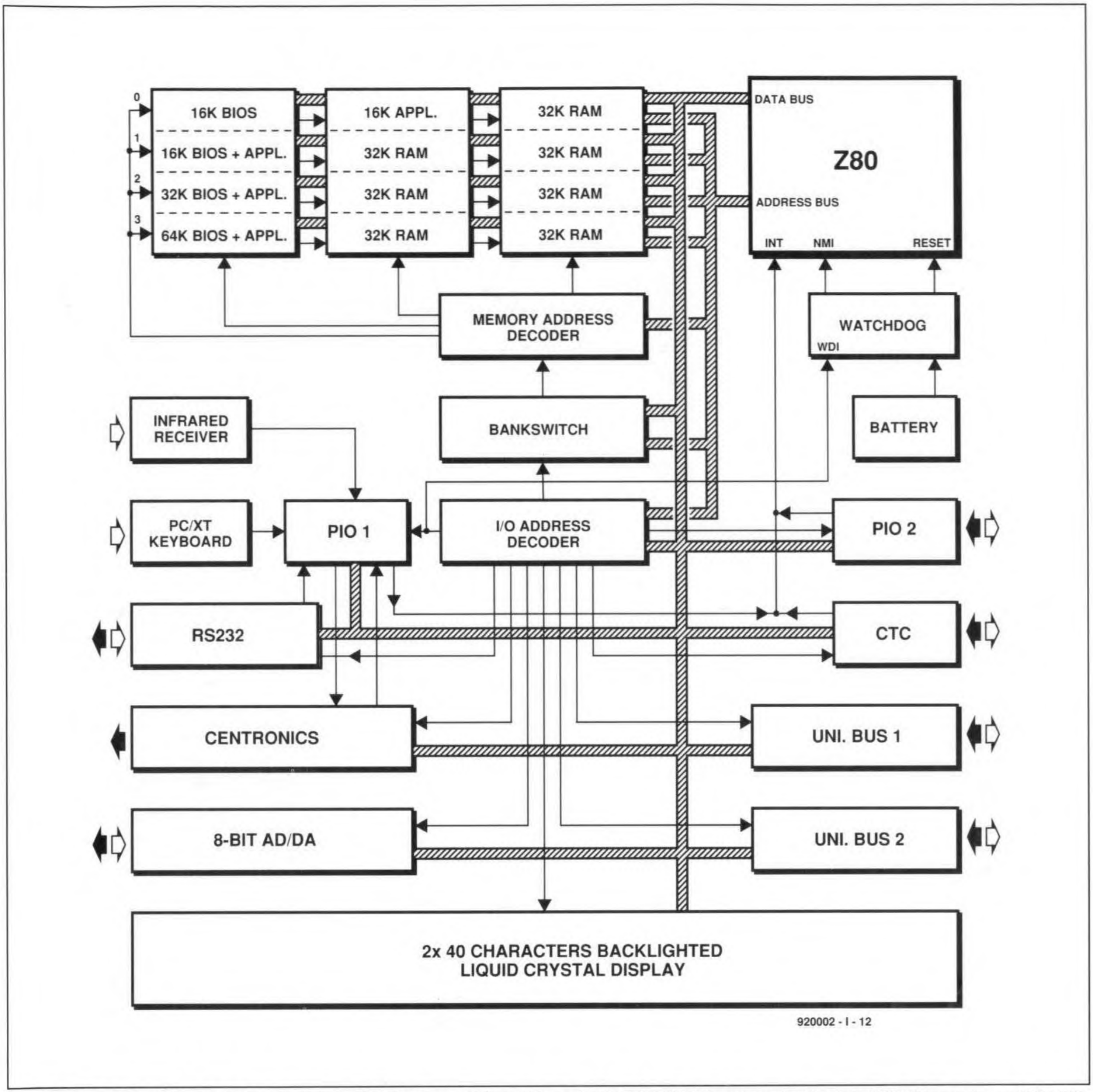


Fig. 1. This block diagram clearly shows the structure of the Z80 card. The Z80-CPU is supported by two Z80-PIOs and one Z80-CTC. Together with the memory, these four ICs form the heart of the system.

data either from a terminal via the RS232 link, or more directly via an XT-compatible keyboard or any RC-5 compatible infra-red remote control. Apart from the digital interfaces, the card also offers an analogue interface in the form of an 8-bit ADC/DAC.

Those of you who require even more I/O capacity will be pleased to find two universal buses that carry the (buffered) databus, a select line and two address lines. This extension bus is readily connected to any peripheral device or card that does not require more than four addresses in the I/O range. Examples of cards that can be connected are the relay card for the universal bus (Ref. 1) and the opto interface card for the universal bus, to be published in a future issue.

The Z80 card has a watchdog that serves to signal power supply failures. When such a failure occurs, it arranges for the 'current state of affairs' to be stored in time by issuing a non-maskable interrupt (NMI). It also serves to re-initialize the card by means of a reset after a software crash, and to switch between the battery and the power supply to prevent data loss when the system is switched on and off.

Memory structure

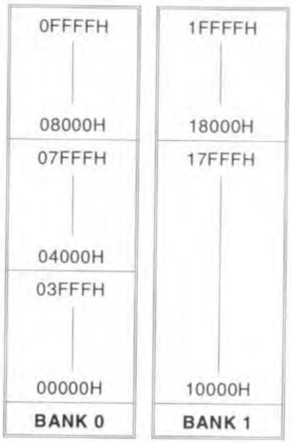
To operate the Z80 requires an external memory in the form of an EPROM or a RAM. As shown in the block diagram, the present card offers four memory configurations. The standard system configuration consists of two 16-KByte PROMs and one 32-KByte RAM. One EPROM contains the application program, the other the system BIOS (basic input/output system). The basic software available in the system allows an application program (stored in the first EPROM) to be started automatically. The other memory configurations allow the BIOS to be combined with user software run from a 27128, 27256 or a 27512, with a 64-KByte RAM in parallel. Further information on the exact memory and address allocations for each configuration may be found in Figs. 2a and 2b.

The Z80 BIOS

A BIOS is basically a program structure that enables the basic hardware and software in a

CON0	0	1	0	1	
CON1	0	0	1	1	
BANK0	32K ROM 32K RAM	16K ROM 32K RAM	32K ROM 32K RAM	32K ROM 32K RAM	
BANK1		32K RAM	32K RAM	32K RAM 32K ROM	
IC1	27128	27128			00000H - 03FFFH
			27256		00000H - 07FFFH
				27512 1/2	00000H - 07FFFH
				27512 1/2	18000H - 1FFFFH
IC2	27128				04000H - 07FFFH
		43256	43256	43256	10000H - 17FFFH
IC3	43256	43256	43256	43256	08000H - 0FFFFH

920002-I-13



920002-I-14

Fig. 2. (left) The memory configuration is determined by the position of jumpers CON1 and CON2, and the associated ICs. (right) Bank switching is used to enable the processor to address up to 128 Kbyte of memory.

```
                  0059H
;***********************************************************
;name:            ldirmv
;function:        load increment repeat into memory from video
;
;input:           BC=length, DE=ram address
;                 HL=display character pointer
;output:          none
;changes:         AF, BC, DE
;***********************************************************
```

920002-I-15

Fig. 3. Example of BIOS routine documentation.

microprocessor system to be upgraded without having to scrap or rewrite existing software. In most cases, the BIOS is a list of start addresses that are the same for each version of the software. A specific function of the system may be called by calling such a start address. The start address contains a jump to the real address of the subroutine required. Thus, the BIOS user must keep in mind:

- the function of the subroutine;
- the call address;
- the variables required, i.e., the use and content of certain registers;
- how the registers contents are changed upon returning from the subroutine.

Only when all these features are known, the BIOS becomes 'transparent', and the user need not have a detailed knowledge of the exact operation of the subroutine.

A BIOS subroutine may be described as shown in Fig. 3. This definition provides all the data on the subroutine required to use it without knowing exactly how it works.

Memory and I/O address decoders

The selection signals of the main components in the Z80 system must be decoded to realize different memory configurations while avoiding conflicts. The configuration chosen and the bank switching information together affect the memory addressing. The configuration is selected before the system is started, and the bank switching is arranged by software. However, provision is made in a hardware to ensure that bank switching is possible only if allowed in the configuration used.

For those of you who do not know what bank switching is all about, the following explanation. Since the Z80 can only address 64 Kbytes of memory, certain provisions must be made if we want it to access a larger memory. In this case, we wish to address 128 KByte, divided into blocks of 32 Kbyte each. This means that there are four possibilities to select a total of 64 Kbytes of memory. Hence, two bits are required to implement bank switching — each bit selects two blocks of 32 Kbyte. When switching such a bit, we must take care not to switch off the block currently used by the CPU. Fortunately, the BIOS contains routines that make bank switching smooth and easy, as required, for instance, to access the RAM 'alongside' the BIOS EPROM in configurations 1, 2 and 3. The bank switching information is stored in a latch that can be written to via I/O addressing.

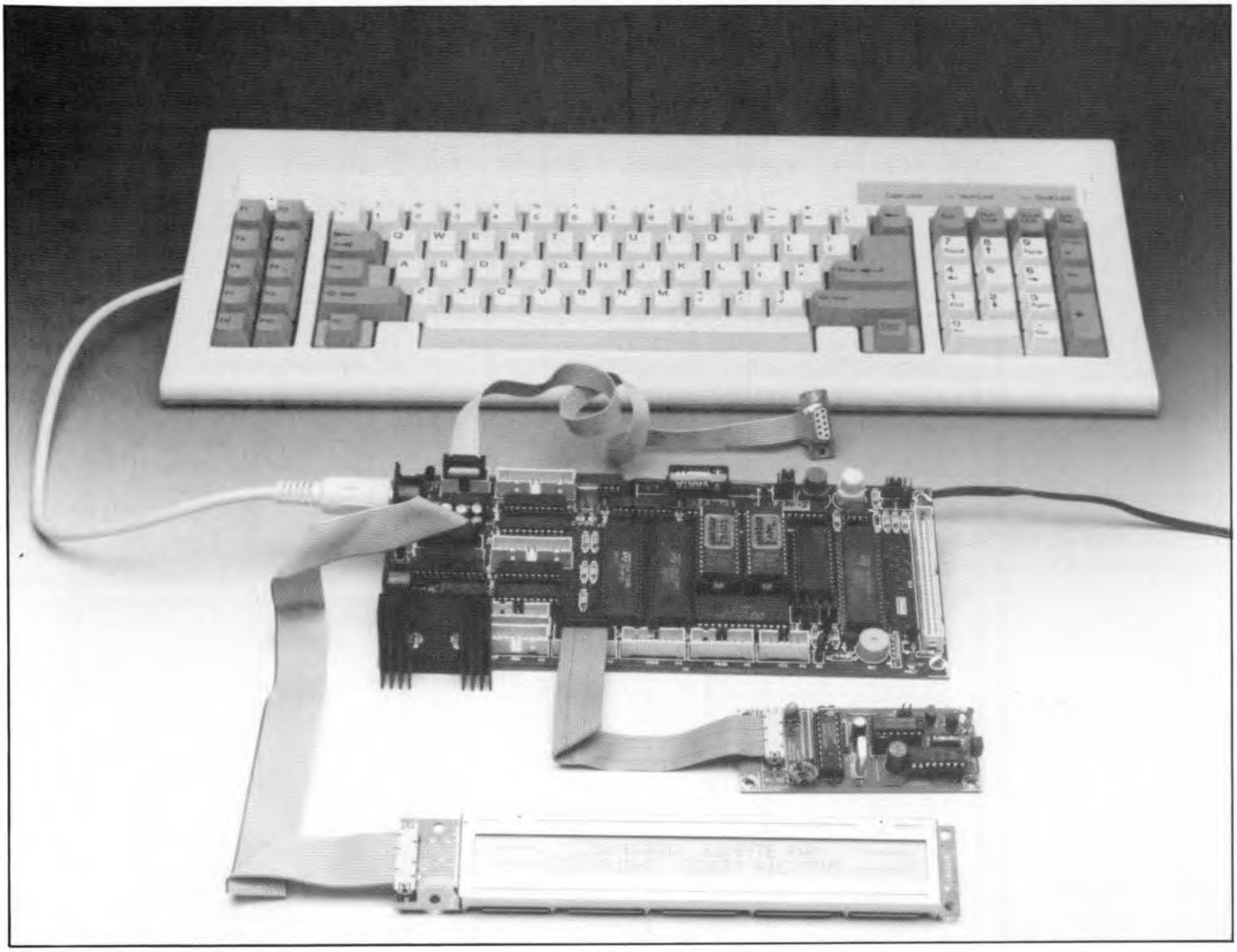
PIOs: parallel I/O

The Z80 card has two PIO (parallel input/output) ICs, each of which contains two 8-bit ports. The PIOs are initialized to bit input/output mode, which means that each bit may be used individually as an input or an output. PIO1 is partly used for internal functions, while PIO2 is available to the user, and is programmed as an input (this can be changed as required via the BIOS).

One port of PIO1 is used for internal signals, and the other to decode the signals supplied by the infra-red receiver. When the IR receiver is not used, the port is, of course, free for other applications. Among the functions of the other port in PIO1 are Centronics handshaking, decoding the PC-XT keyboard, and detecting interrupt signals issued by the RS232 interface. Evidently, these port functions can not be redefined via the BIOS.

CTC: four timers

The Z80-CTC on the present card contains three counters/timers. Of these, timer 3 is used to generate interrupts at 10-ms intervals during which time-dependent functions



can be completed. A software 'hook' is provided to extend this interrupt routine with your own software. A hook is the software equivalent of a road diversion. The system area of the memory contains addresses that are filled with return instructions after the card is switched on. A number of BIOS routines start with a call to one of these addresses. Normally, this address contains a 'return' instruction to the BIOS routine. However, by replacing this return with a call to a user routine, the program can be diverted to an extension of the BIOS routine, which is thus 'hooked' to the basic one. Five addresses are available for each hook, which is sufficient to place a call and a return. If a jump instruction is used at the hook address, the extension subroutine does not return to the hook and the basic BIOS routine, which is then simply not executed. The diskette supplied with this project (order code ESS 1711) contains an example of the use of a hook.

Returning to the functions of the timers, the interrupt routine for Timer 3 counts down the 'on' time of the on-board buzzer, so that the software need not wait for this. Timer 0, Timer 1 and Timer 2 in the CTC are free for your own use.

Keyboard and IR control

Parallel input to the Z80 card is furnished either by a PC-XT keyboard or the RC-5 code infra-red receiver described in Ref. 2. The PC keyboard is connected to the board via its curly cord and 5-way DIN plug. The Z80 card automatically detects the parallel input device at power-on.

Any type of RC-5 compatible IR transmitter may be used. The push-button with the number '1' on it is defined as the escape (ESC) key, while the other buttons are assigned an ASCII value equal to their code plus 32. The key definitions are stored in RAM, which allows them to be readily changed. The jumper marked REM was origibnally designed into the circuit to select remote control data tables. Its function has been scrapped, however, leaving it free for your own programming experiments.

RS232 and Centronics interfaces

The RS232 and Centronics interfaces on the Z80 card enable it to be controlled from a distance, and to print data respectively. The RS232 interface is suitable for full-duplex communication with a terminal (or a PC running communication software). The interface gives the Z80 card the function of DCE (data communication equipment) which means, among others, that the Z80 card will only 'do' something via the RS232 if so requested. The software contains routines that allow parts of the memory to be read or written via the RS232 interface. It is also possible to adapt the baud rate and the transmission format. The interface supports all standard baud rates between 50 and 38,400.

Watchdog and battery backup

As already mentioned, the watchdog has a number of functions on the Z80 card. To begin with, it ensures the minimum required length of the CPU reset pulse when the card is switched on. In addition, the watchdog monitors the unregulated and the regulated supply voltages, and arranges the switching between the 5-V supply and the battery, and vice versa. The watchdog has an input that continually checks if the card has not 'crashed'. If a crash occurs, the watchdog resets the card. The latter function of the watchdog will be particularly valued with

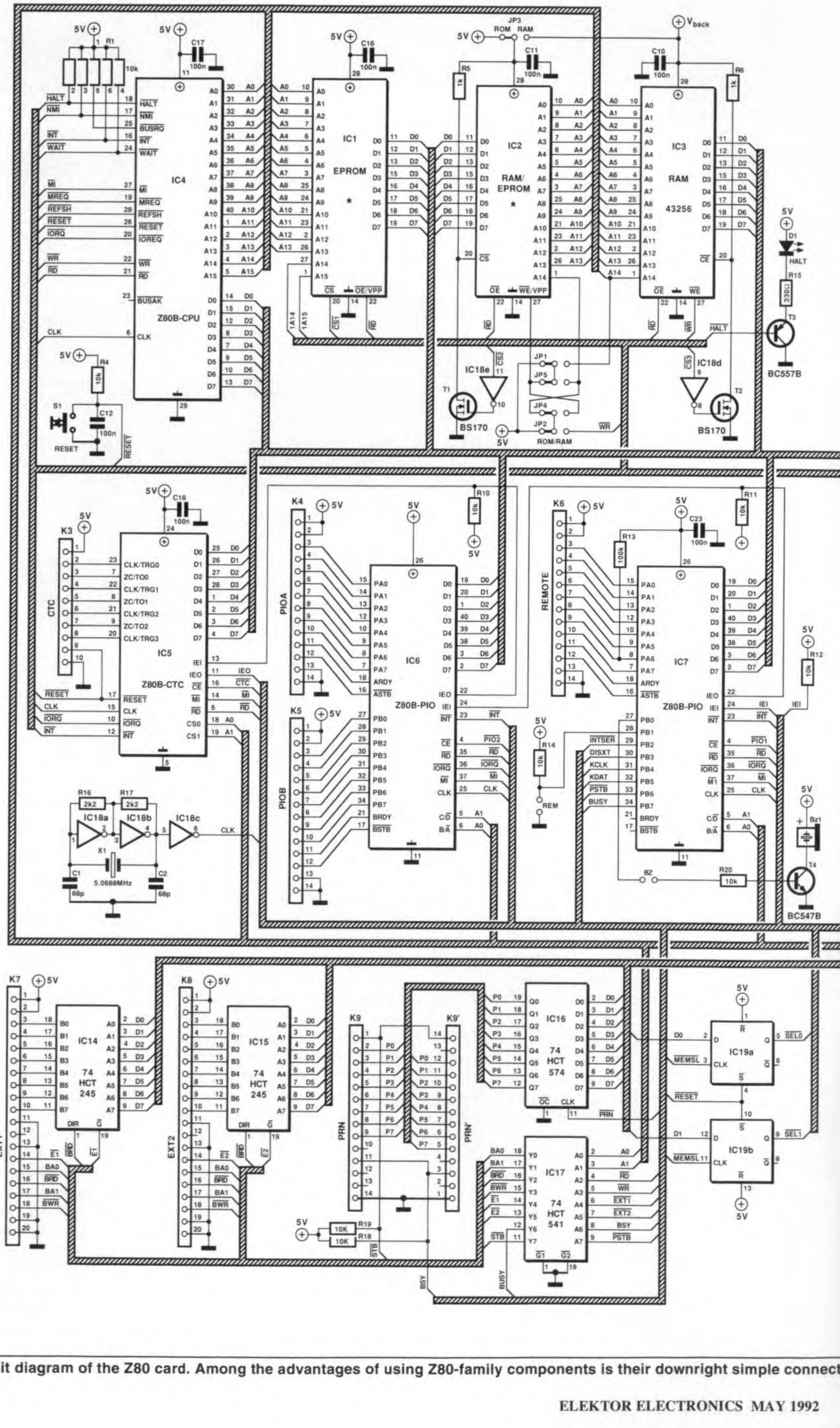
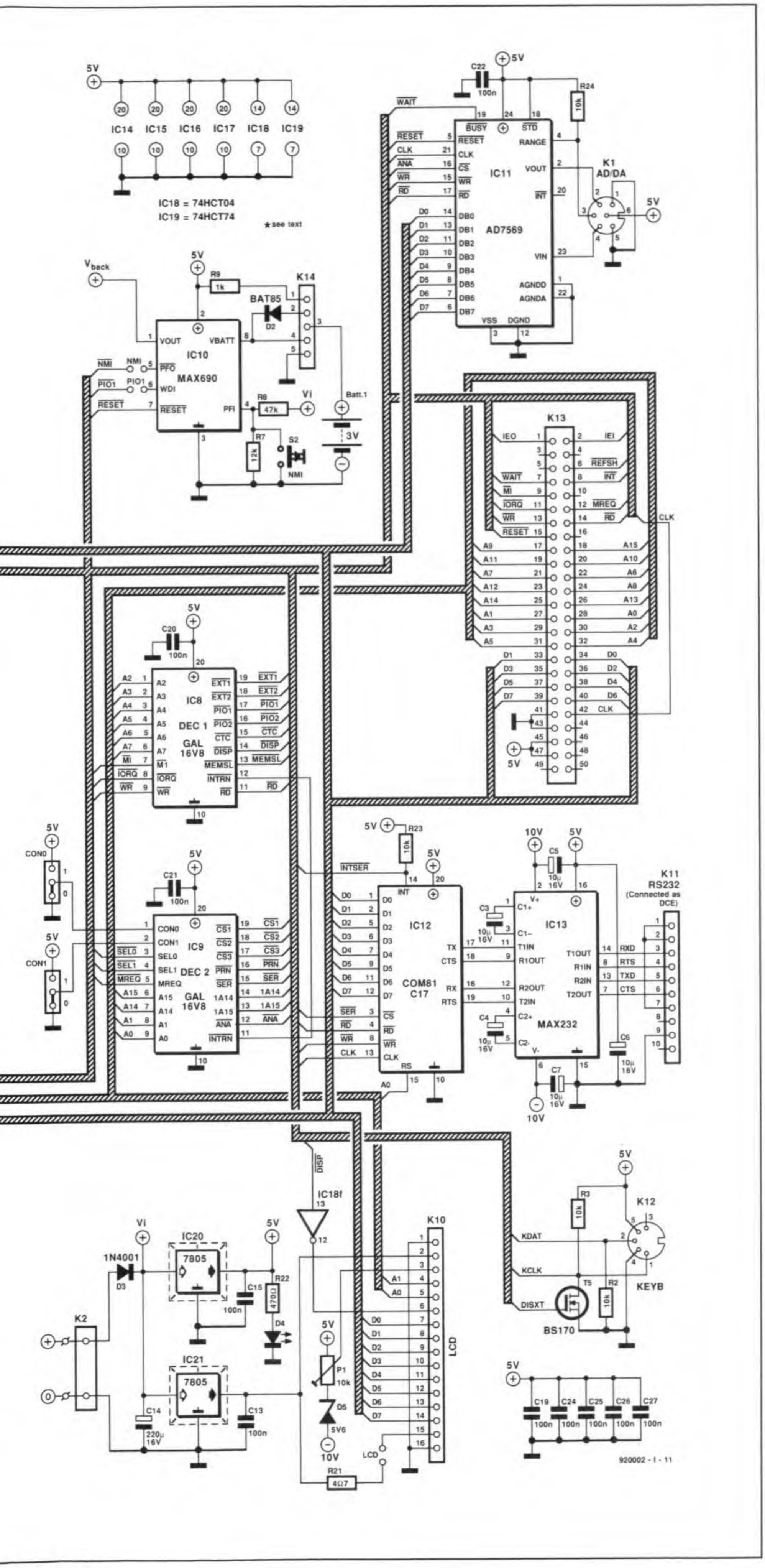


Fig. 4. Complete circuit diagram of the Z80 card. Among the advantages of using Z80-family components is their downright simple connect



nd low cost.

measurement and control applications that run unattended.

Liquid crystal display

The Z80 card offers the possibility of connecting a liquid crystal display (LCD) of the normal or back-lighted type. Although LCDs with up to 80 characters are supported, the preferred type has 2×40 characters. However, one-line and four-line LCDs may be connected also. Nearly all of these intelligent LCDs are based on the HD44780 display controller from Hitachi, and have basically the same connections, albeit that the pins are sometimes arranged differently.

Circuit description

After a rather lengthy tour along the various functions shown in the block diagram of the Z80 card, it is time to see how these functions take on their practical shape.

The circuit diagram is given in Fig. 4. In the upper left-hand corner of the diagram we find the Z80B-CPU. A 6-MHz processor is used here because the system clock frequency is 5.0688 MHz. This frequency is used by the serial interface circuit to derive the standard baud rate series, starting at 50 bits/s.

The memory and I/O components are seen to the right of the Z80B-CPU and below it respectively. The I/O ICs are, of course, also B-versions because of the system clock frequency. The advantages of using Z80 family I/O components are basically that they are inexpensive, widely available, and extremely easy to implement. Essentially, once all the system and data lines of the components are connected, only the 'select' lines remain. Further, we must give some thought to the priority level assigned to each source that can generate an interrupt. This is arranged via the IEO (interrupt enable output) and the IEI (interrupt enable input) terminals, which are connected into a chain. The priority order is defined as follows: (1) IC7 (PIO1 for internal use); (2) IC6 (PIO2 for external use); IC5 (CTC).

In the basic software, only POI1 and the CTC generate interrupts. The PIO interrupts originate from the RS232 port and/or the PC/XT keyboard. The CTC generates an interrupt every 10 ms, during which, among others, the IR keyboard is checked. Without the pull-up resistor at pin 8, the keyboard buffer would be filled with random characters if the IR remote control is not used. Resistor R23 has a similar function, and prevents unwanted interrupts if IC12 is not fitted.

The connection of the EPROM(s) and the RAM(s) is not entirely straightforward because of the different memory configurations that are allowed. Position IC2 can hold either a RAM or an EPROM, which requires jumpers JP1 to JP5 to be set accordingly. The circuit diagram shows the jumpers set to the 'EPROM' positions (memory configuration '0').

Position IC1 accommodates one of three

EPROM types: the 27128, 27256 or 27512, so that address lines A14 and A15 need to be given their appropriate level. These lines are controlled by the memory address decoder located in IC9, a GAL Type 16V8. The configuration is determined directly by the setting of connectors CON0 and CON1. These are pre-set to give the memory configuration shown in Fig. 2a, and determine whether or not address lines A14 and A15 are passed. The GAL also arranges the selection of the three memory components, which, apart from A14 and A15, also depends on the memory configuration and the SEL0 and SEL1 signals furnished by IC19. SEL0 and SEL1 allow you to switch between BANK0 and BANK1 in blocks of 32 KByte (see Fig. 5).

Transistors T1 and T2, and resistors R5 and R6, ensure that the RAMs do not remain selected at power-down, so that their data is available again when the system is switched on. The power supply connections of IC2 and IC3 are connected to pin 1 of IC10, which takes care of the battery backup switch-over function.

The MAX690 watchdog (IC10) switches between 5 V and the battery voltage, U_{batt}, as soon as the voltage at pin 2 drops below (U_{bat}t–50 mV), or rises above (U_{batt}+70 mV). Further, watchdog types MAX690 and MAX694 generate a reset if the supply voltage drops below 4.65 V. Those of you who want a wider margin are advised to use the MAX692, which issues a reset at 4.4 V. Finally, the watchdog supplies a defined reset pulse at power-on (MAX690 and MAX692: 50 ms; MAX694: 200 ms). The input voltage to the card is monitored with the aid of R7 and R8. If the voltage at pin 4 of the watchdog drops below 1.3 V, output $\overline{\text{PFO}}$ goes low. Provided the NMI jumper is installed, this low level can trigger a non-maskable interrupt that allows the current situation to be saved before the supply voltage drops below 4.65 V, and IC10 resets the card. This option is supported only by a software hook coupled to the NMI.

Apart from monitoring the supply voltages, the watchdog is also capable of checking if the Z80 card is still running. The watchdog timer monitors the WDI input, to which a signal must be applied that changes at least every 1.6 ms. If this signal fails, the watchdog resets the card. The WDI input may be connected to the selection signal of IC7 via jumper PIO1. Since, if the BIOS is used, the CTC generates an interrupt every 10 ms, and the associated subroutine addresses IC7, the presence of the selection signal is a good sign that the card is still running.

A second GAL, IC8, contains the address decoding logic for the I/O circuits. Here, the advantage of a GAL is a drastic reduction of the chip count for an address decoder that allows the I/O ICs to be addressed fully and without 'image' areas elsewhere in the memory. An address overview of the I/O components is given in Fig. 6.

Memory select	SEL0	SEL1
00000H - 07FFFH	1	x
08000H - 0FFFFH	x	1
10000H - 17FFFH	0	x
18000H - 1FFFFH	x	0

920002-I-17

Fig. 5. Memory selection by means of the SEL0 and SEL1 lines.

I/O addresses	
EXT1	0FCH - 0FFH
EXT2	0F8H - 0FBH
PIO1	0F4H - 0F7H
PIO2	0F0H - 0F3H
CTC	0ECH - 0EFH
DISP	0E8H - 0EBH
RS232	0E6H - 0E7H
PRN	0E5H
ANALOG	0E4H
MEMSEL	0E0H - 0E3H

920002-I-18

Fig. 6. Addresses of the I/O components that are realized with the aid of GALs.

The analogue interface is built around an AD7569 8-bit A-D/D-A converter. This IC was chosen because it is simple to interface, and reasonably fast: the D-to-A and A-to-D conversion times are 1 µs and 2 µs respectively. Since the $\overline{\text{BUSY}}$ signal (which is low during the A-to-D conversion) is connected to the $\overline{\text{WAIT}}$ terminal of the CPU, the ADC can be read simply with an IN instruction. Hence, the instruction

IN A,(ANALOG)

directly provides the input voltage as a hexadecimal value in the accumulator (provided, of course, that ANALOG equals the I/O address of the A-D/D-A converter), without the need to arrange the timing for this read operation. Writing to the DAC is equally simple: instruction

OUT ANALOG,A

puts the desired voltage on the output.

Depending on whether the range input is high or low, the input and output voltage range is from 0 V to 2.5 V, or 0 V to 1.25 V. The range input is held high via R24, and can be made low by connecting pin 3 of K1 to ground.

The RS232 interface consists of two ICs: a 20-pin UART (universal asynchronous receiver/transmitter) Type COM81C17, and a level converter Type MAX232. The COM81C17 contains everything to set up a serial interface quite easily. The system clock frequency used enables all standard baud rates between 50 bits/s and 38,400 bits/s to be programmed. In addition, the UART arranges the handshaking protocol on the serial link, which does away with the need for any software equivalents. The step-up converters contained in the MAX232 ensure RS232 signal levels of +10 V and –10 V, which will work in most, if not all, applications. Connector K11 is wired such that the Z80 card forms a DCE that is readily connected to a DTE (data terminal equipment; a computer in most cases) via a 9-way flatcable.

At this point we have nearly completed our tour along the main components in the circuit diagram. Connector K12 serves to hook up a PC/XT keyboard. Such keyboards are widely available at very low prices from PC surplus outlets. If you happen to have a PC XT/AT type with automatic switching, the Z80 card ensures that it is set to PC XT mode. The keyboard is reset by software following a hardware reset. This is done via transistor T5.

The contrast of the LCD connected to K10 is controlled via preset P1. A back-lighted display may be used in low ambient light conditions. Because of the possible need for a back-light supply, and to keep display multiplexing noise away from the processor, the LCD is powered separately by IC21, while the rest of the circuit is powered by IC20. The back-light supply depends on the LCD type used. There are types that require a supply voltage of 5 V (replace R21 with a wire link), and types that require a certain current (in which case R21 must be given an appropriate value). The back-light connections are pins 15 and 16 of connector K10. The jumper marked 'LCD' provides a simple way of switching the back-light on and off.

The printer datalines are furnished by latch IC16, while the control signals to and from K9 and K9' are buffered by IC17. This IC also buffers the control signals to the two external bus connections. The databases are buffered by IC14 and IC15. The pinning of connectors K7 and K8 is compatible with the universal bus (Ref. 1).

That concludes the description of the Z80 card as far as its concept is concerned. Next month we will tackle the construction and testing of the card, as well as making use of the associated test software contained in the BIOS EPROM. ❑

References:

1. "Relay card for universal I/O interface". *Elektor Electronics* November 1991.

2. "Universal RC5-code infra-red receiver". *Elektor Electronics* January 1992.

CORRECTIONS AND UPDATES

Digital Audio/visual system (Multi-purpose Z80 card)

May and June 1992
An extensive description of a modification to the memory backup circuit on the Multi-purpose Z80 card is available free of charge through our Technical Queries service.

FM stereo signal generator

May 1993
Capacitors C17 and C19 should have a value of 33nF, not 3nF3 as indicated in the circuit diagram and the parts list of the multiplex generator.

Workbench PSU

May 1993
The polarity of capacitor C15 is incorrectly indicated on the PCB component overlay (Fig. 5a), and should be reversed. The circuit diagram (Fig. 2) is correct.

Transformer TR2 is incorrectly specified in the circuit diagram (Fig. 2) and in the parts list. The correct rating of the secondary is 2 × 12V/5A. Also note that the secondary windings are connected in series to give 24 V.

Audio DAC

September 1992
The polarity of capacitors C25 and C58 is incorrectly indicated on the component overlay of the D-A board (order code 920062-2), and should be reversed.

U2400B NiCd battery charger

February 1993
The value of resistors R17 through R27 should be 2.7kΩ, not 12.7kΩ as stated in the parts list.

VHF/UHF receiver

May 1993
In Fig. 4, the connections to ground of the AF amplifier outputs, pins 5 and 8, should be removed. The amplifier outputs are connected to the loudspeaker only. The relevant printed circuit board is all right.