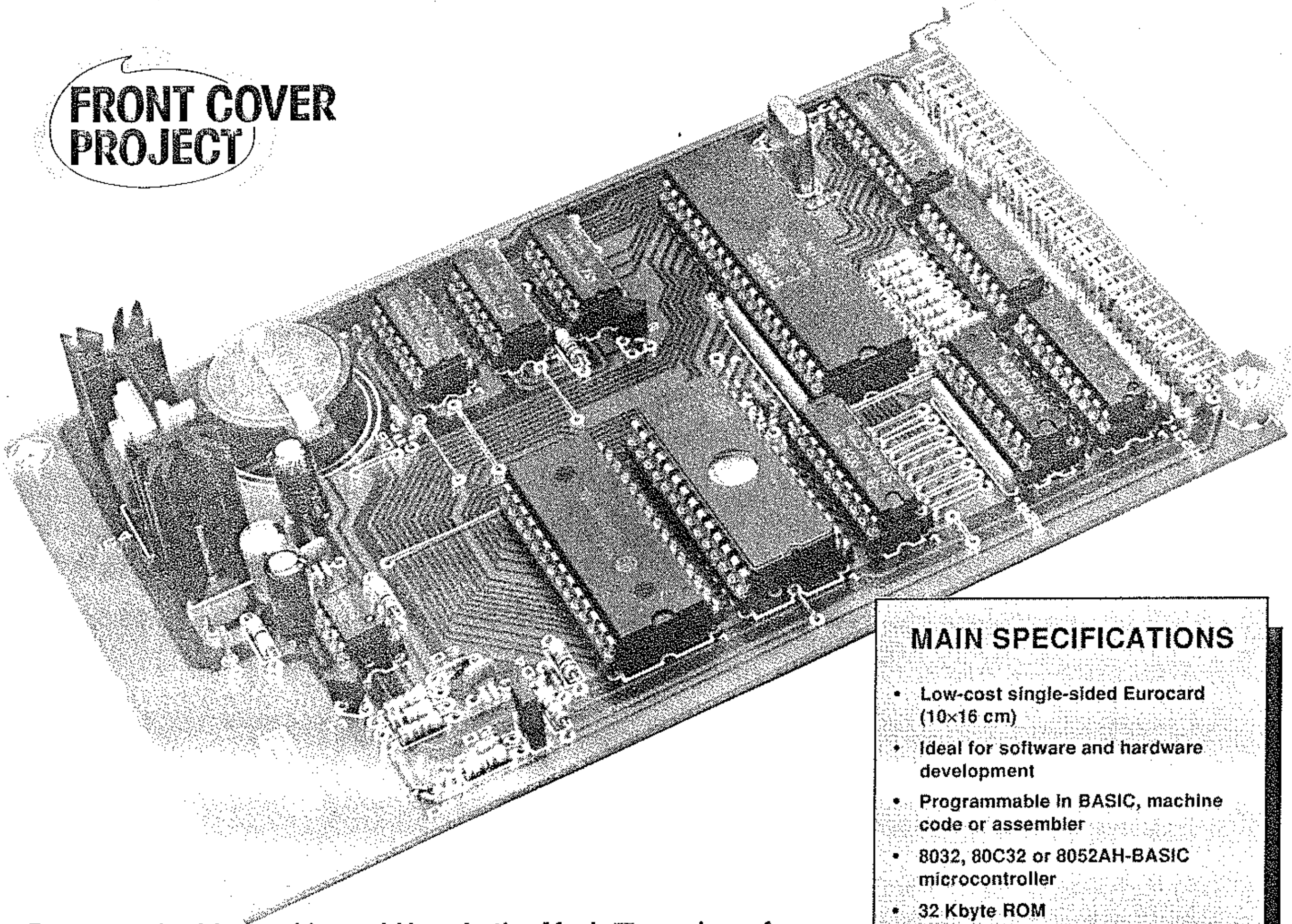
8032/8052 SINGLE-BOARD COMPUTER

FRONT COVER PROJECT



Power to the bits and bytes! Here is the Mark-II version of the popular BASIC computer we published a few years ago. This time we present an even more powerful system with more RAM, more ROM, an on-board EPROM programmer, and many more goodies on a single-sided Euro-size PCB. Ideal for small control applications and software development, the present single-board computer (SBC) can work with Intel's powerful 8032, 80C32 or 8052AH-BASIC processor. The latter has an on-chip interpreter that allows you to program in BASIC with full access to machine code.

MAIN SPECIFICATIONS

- Low-cost single-sided Eurocard (10×16 cm)
- Ideal for software and hardware development
- Programmable in BASIC, machine code or assembler
- 8032, 80C32 or 8052AH-BASIC microcontroller
- 32 Kbyte ROM
- 32 KByte RAM
- Memory backup battery
- Simple power supply
- On-board EPROM programmer
- Simple to use serial interface
- Clock frequency up to 24 MHz (80C32) or 15 MHz (8032 or 8052AH-BASIC)
- Low-cost a-c row 64-way DIN connector for serial interface and hardware extensions

H. Reelsen

MICROCONTROLLERS these days are silent workers in many apparatus, ranging from the washing machine to the video recorder, to mention but two examples in the home. Nearly all of these controllers are mask-programmed and therefore of very little use for hobby applications since the program they execute can not be altered. And even if we could alter the program, the information necessary to do so — an instruction set, an assembler language description and some basic hardware information — is often not available or very difficult to obtain. Also, the last resort of many programmers, a cross-assembler, is long sought but never found. In short, many microcontrollers, powerful as they may be, are not accessible (but bear in mind that they were not designed to be so).

An marked exception to the above 'misery' is the 8052AH-BASIC from Intel. This microcontroller has features that seem to make it more accessible than any other single-chip microcontroller with a reasonable price tag. Consider, for instance, its bit manipulation instructions, its internal BASIC interpreter, its ability to program

EPROMs, or load a BASIC program into memory via a three-wire RS-232C link.

In 1987 we published a single-board computer based on the 8052AH-BASIC (Ref. 1), and it has been a popular project ever since. The computer was not only built and programmed by vast numbers of computer enthusiasts, its design concept was also taken up commercially, witness the sudden influx of 8052-based controller boards and development systems following our publication.

The single-board computer described here is an upgraded version of the 8052AH-BASIC computer. The improvements are basically a larger RAM and ROM space of 32 KByte each, a memory backup circuit with 2 µA RAM data retention current, a 12.5-V EPROM programming voltage supply, and, last but not least, the possibility to use the ROM-less (inexpensive) 8032 or 80C32 microcontroller. A further boon for the home constructor is that the printed-circuit board for the present computer is single-sided and designed with 0.4-mm wide copper tracks for easy reproduction. This allows the cost of the computer to be kept to a minimum. If a double-sided board had been used, it would probably have cost more than all the components on it to build the 8032 version of the computer.

The circuit

The heart of the circuit shown in Fig. 1 is either the ROM-less 80(C)32 CPU or the 8052AH-BASIC processor. The LS (least-significant) group of address lines, A0-A7, is multiplexed with the data, and extracted with the aid of octal latch IC8 when the ALE (address latch enable) signal is logic high. The latched databits are fed direct to the data inputs of the system RAM and ROM, as well as to buffer IC9. The MS (most-significant) group of address lines, A8-A15, is not multiplexed and connected direct to the memories and the address decoders, IC1, IC2 and IC3.

The memory map of the system has the following structure:

- two ROM ranges, one from 0000$_H$ to 3FFF$_H$ and one from 8000$_H$ to BFFF$_H$;
- one RAM range from 0000$_H$ to 7FFF$_H$.
- An I/O range mapped between addresses C000$_H$ and C0FF$_H$.

The structure of the memory map is illustrated in Fig. 2. Wire jumper Br1 when fitted allocates the range from 0000$_H$ to 1FFF$_H$ (a part of the ROM range) to the internal BASIC interpreter ROM in the 8052AH-BASIC processor. The wire jumper is removed when an external ROM (or EPROM) is used, as required in most cases with the 80(C)32.

The ROM and RAM ranges have an overlapping area between 0000$_H$ and 1FFF$_H$. The PSEN signal allows the processor to use this 'shared' area either as program memory or data memory. This means that the memory structure of the 80(C)32 and 8052 is not based on the 'classic' Von Neumann model in which the memory areas allocated to program and data are arranged as contiguous blocks in the address space. Fortunately, the

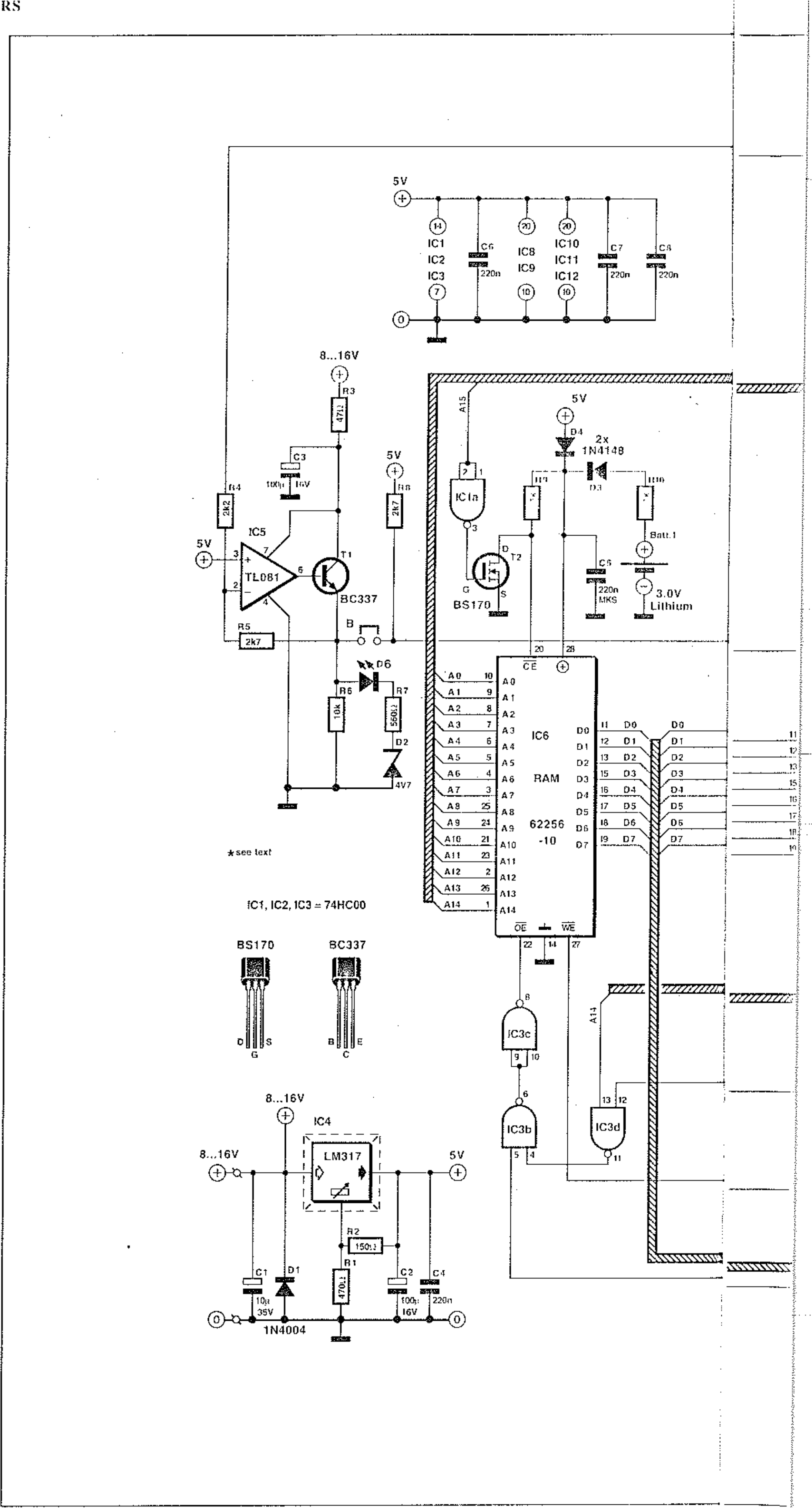
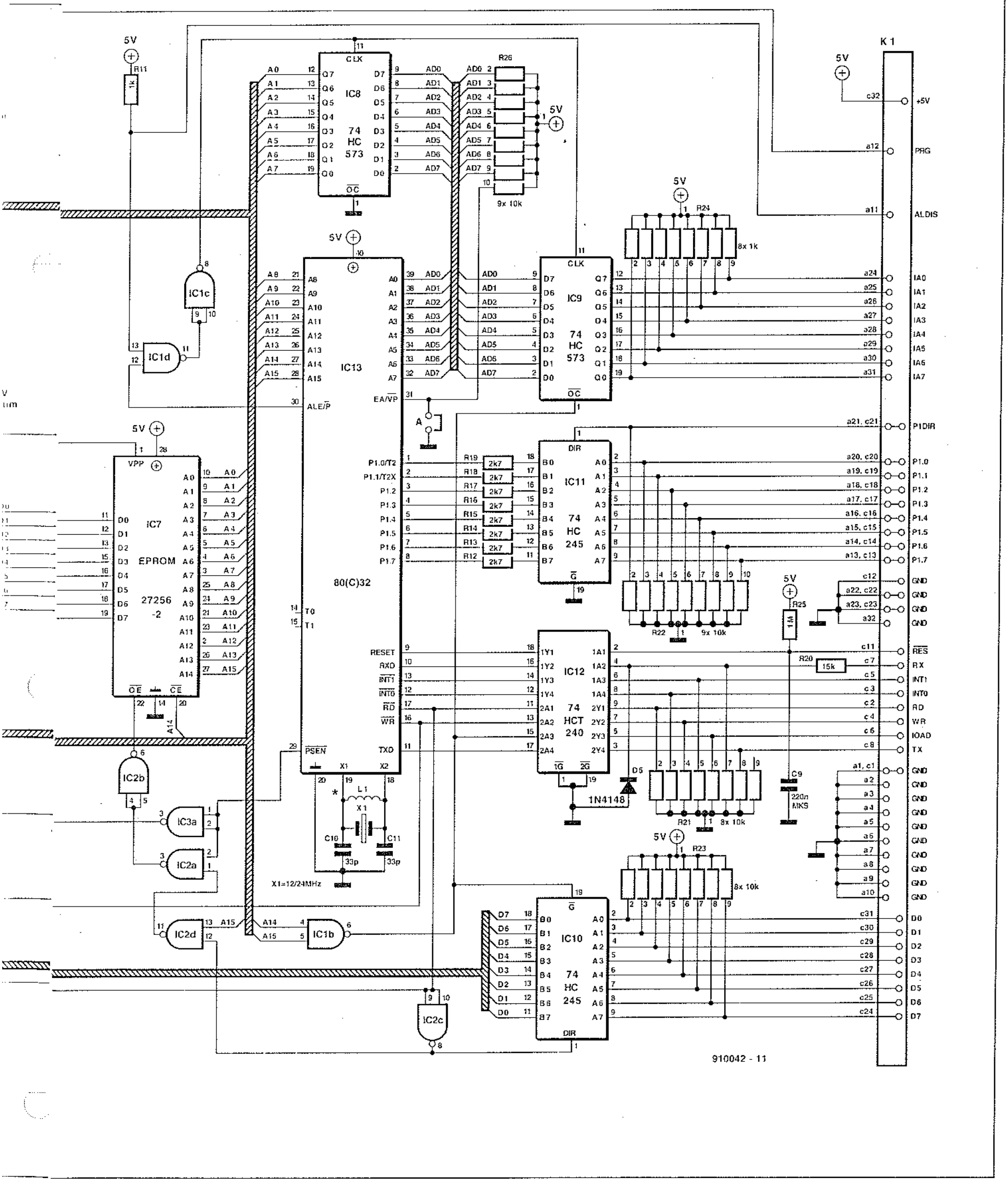


Fig. 1. Circuit diagram of the single-board computer. Note that although a 80C32 CPU is sho you may



; showıyou may also fit a 8032 or 8052AH-BASIC. The latter brings programming in BASIC within easy reach if you have a PC emulating a terminal.

present system does allow for a 'classic' address division. This is achieved as follows: access to the lower 16 Kbytes in the ROM (EPROM) area is only allowed when the CPU supplies the appropriate address, and actuates the PSEN line. The upper 16 KByte range can be selected with the appropriate address and the $\overline{\text{RD}}$ (read) or the PSEN signal. This means that the upper 16 KByte range may function as a data memory or a program memory, which is a requirement for the storage of BASIC programs. The memory division is realized by a 74HC00 (IC$_2$) whose gates are combined to form a kind of decoder.

The lower 16 KByte block in the RAM range between 0000$_H$ and 7FFF$_H$ can only be used as data memory, while the upper 16 KByte block can be used as data memory or program memory because it does not overlap the ROM range. The upper block is therefore suitable for writing and debugging programs written in machine code.

The address decoding of the RAMs is arranged by IC$_3$. Gate IC$_{1b}$ combines address lines A14 and A15 to provide an input/output address strobe, $\overline{\text{IOAD}}$, required for the signalling of access to the address range above C000$_H$, i.e., the range reserved for I/O operations.

Although IC$_9$ latches the lower address byte like IC$_8$, it is controlled differently via its $\overline{\text{OE}}$ input which is connected to the previously mentioned $\overline{\text{IOAD}}$ line. This means that the 8-bit address is latched only when $\overline{\text{IOAD}}$ is actuated. Otherwise, the de-actuated IC outputs are held at +5 V by a resistor array and so represent a value FF$_H$.

The datalines are buffered by an octal three-state bus driver, IC$_{10}$. The $\overline{\text{RD}}$ signal supplied by the controller is inverted by IC$_{2c}$ and determines the data direction (read/write) of IC$_{10}$. To keep the external bus free from all of the data transfer operations performed by the processor, IC$_{10}$ is enabled by $\overline{\text{IOAD}}$ via its $\overline{\text{G}}$ input.

Port 1 is freely available and may be used to convey outgoing as well as incoming data. A bidirectional bus driver Type 74HC245 protects the controller and increases the drive capacity when the port is used as an output. The logic level applied to the P1DIR connection determines the direction of the data. When P1DIR is not connected, or connected to ground, the port functions as an output (write). The input function (read) is selected by making P1DIR logic high. The 2.7 kΩ resistors between IC$_{11}$ and the controller prevent overloading when port 1 is programmed as an output while IC$_{11}$ supplies data as a result of incorrect programming. The resistors limit the processor output current to a safe 2 mA or so in this output-against-output conflict.

A memory backup circuit retains the RAM data when the power is removed from the computer. A PCB-mount 3-V lithium cell supplies the required data retention current to RAM IC$_6$ via resistor R$_{10}$ and diode D$_3$. When the computer is in normal use, diode D$_4$ conducts and D$_3$ isolates the backup battery. If you have a sensitive voltmeter, the data retention current may be measured indirectly as the voltage across resistor R$_{10}$: 1 mV corresponds to about 1 μA of RAM current.

The RAM is automatically switched to its low-power standby state when the power is removed. This function is effected by a MOSFET, T$_2$. As long as the computer is powered by the mains supply, the FET takes the $\overline{\text{CE}}$ input of the RAM low. When the power is removed, the FET arranges the $\overline{\text{CE}}$ input to be effectively connected to the retention voltage via resistor R$_9$. In this manner, the RAM is never allowed to be in a non-defined state (i.e., $\overline{\text{CE}}$ 'floating' while a voltage exists across the supply terminals) which results in a relatively high current drawn from the battery.

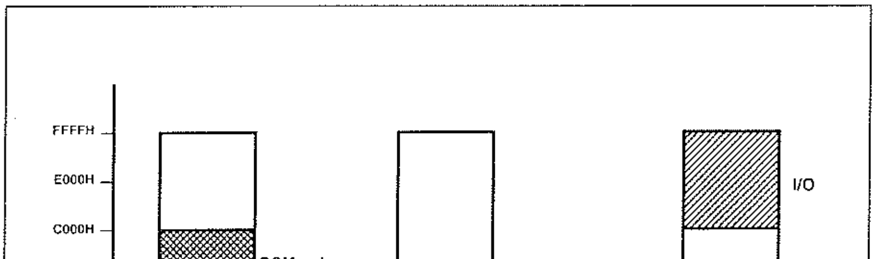


Fig. 2. Parts of the memory space in the system are shared by ROM and RAM. A special kind of memory addressing is therefore required to access the right data or program.

Interfaces

Programming the board in BASIC essentially requires a serial keyboard as an input device, and a serial display as an output device. The two functions are probably best combined into a terminal or a PC running a terminal emulation or general-purpose communication program (see Refs. 2 and 3).

The SBC has an on-board serial interface that works with TTL levels. The RX (received data) and TX (transmitted data) terminals of this interface are available on the DIN-VG64 connector, K$_1$. Although the serial interface of the SBC works basically with TTL levels, there should be no problem in hooking it up to a terminal or PC with an RS232C-compatible input/output. Designed to work with ±12-V signals, most of these interfaces work happily with TTL (0 V/+5 V) signals, although it must be noted that the noise immunity suffers considerably when a relatively long cable is used between the SBC and the PC or terminal.

At the side of the SBC, the signal received at terminal RX of connector K$_1$ is made positive only and limited to a swing of 5 V by R$_{20}$, a resistors in 8-way array R$_{21}$, and diode D$_5$. The RS232C signal is inverted as required by that standard by IC$_{12}$. The serial signal at the TX output of the SBC has a swing of 5 V, and is connected to the RxD input of the terminal. The RX input of the SBC is connected to the TxD output of the terminal or PC.

The two interrupt inputs of the board, INT0 and INT1 (pins c3 and c5 respectively on K$_1$) enable the processor to respond quickly to external events. The use of interrupts is of particular interest when the SBC is at the heart of a computer-controlled system. A software interrupt is acknowledged by the processor with an appropriate servicing routine written by the user.

The SBC may be reset with an external signal via the RES terminal, pin c11 on K$_1$. Since IC$_{12}$ contains inverting buffers, the write (WR), read (RD) and I/O address (IOAD) signals are active-high on connector K$_1$. An example of the use of these signals for a data input/output decoder is shown in Fig. 4.

On-board EPROM programmer

The 8052AH-BASIC processor is capable of storing a BASIC program in EPROM starting from address 8000H. The hardware to do so consists of opamp IC$_5$, transistor T$_1$ and a handful of passive parts. The non-inverting input of the opamp is held at +5 V, while the inverting input is connected to pin a12 (PRG) of connector K$_1$ via a 2.2 kΩ resistor. A low level on PRG takes the opamp output high. Transistor T$_1$ then supplies a programming voltage of about 12.5 V to the Vpp pin of the system EPROM, IC$_7$. When the PRG line is at +5 V, the Vpp pin is at 5 V also.

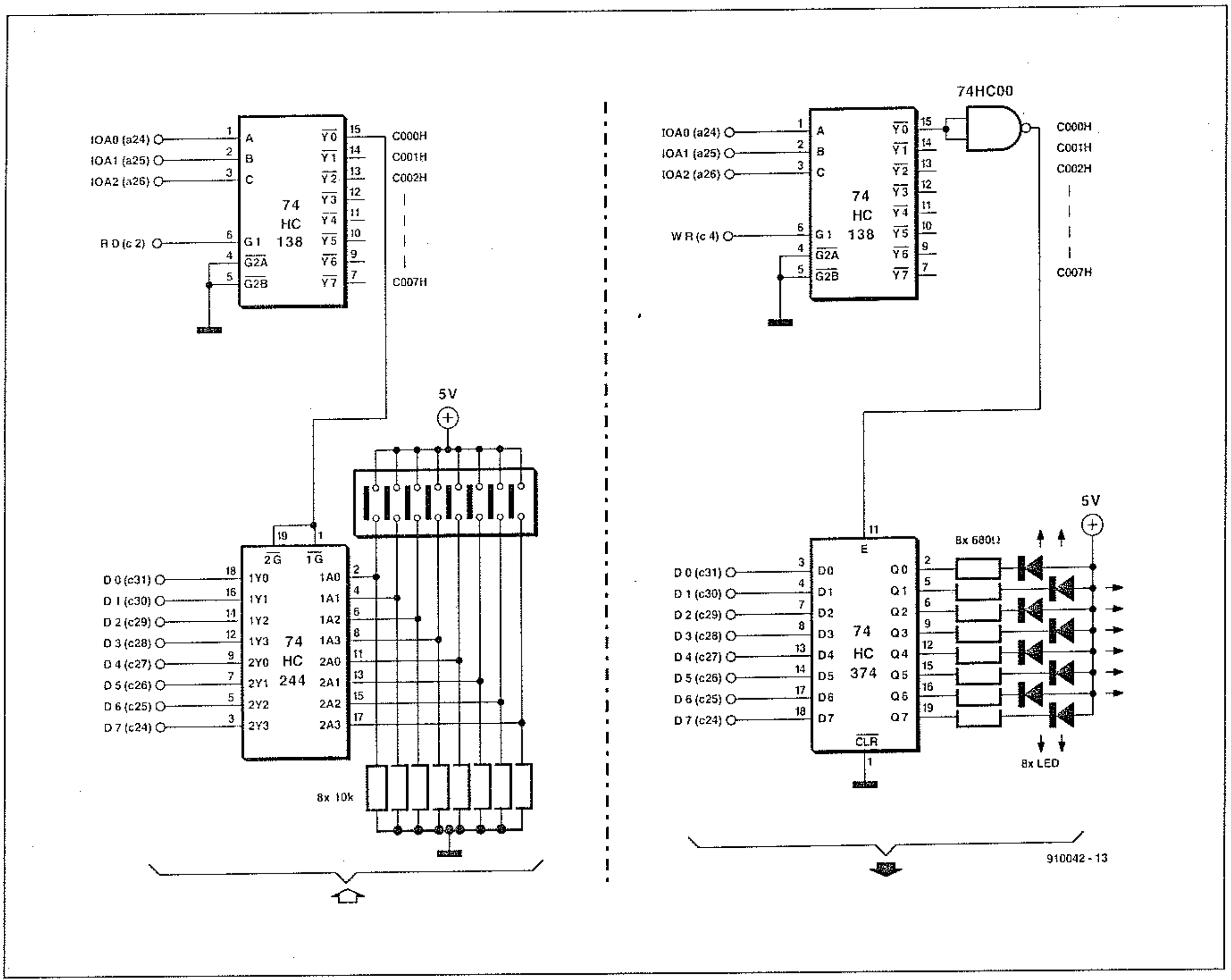


Fig. 4. Illustrating byte-wide input/output for the SBC. Shown here are two suggested external circuits to effect memory-mapped input/output operations with the single-board computer. Both circuits are wired to the system extension connector. The left-hand circuit reads an array of eight push-buttons in a keyboard, while the right-hand circuit is a basic output device that controls eight LEDs.

The ALE signal must be disabled while the EPROM is being programmed. This is achieved by the processor making the ALDIS line logic low. To program an EPROM, connect the PRG terminal to port line P1.4, and the ALDIS line to port line P1.3. The P1DIR terminal is not connected, and jumper Br_2 is best removed.

The programming sequence is indicated by LED D_6, which lights only when the programming voltage is higher than about 6.5 V. Wire jumper Br_2 must not be fitted while D_6 lights.

Which processor?

The choice between a 8052AH-BASIC, a 80C32 or a 80C32 processor is up to you. The latter two do not have an on-chip BASIC interpreter, and can only be programmed in machine code. It is, however, possible to unload the BASIC interpreter from the 8052AH-BASIC, transfer it to EPROM, and run it with a 80(C)32. How this can be achieved is detailed in Refs. 2 and 3. In addition to the information provided in these earlier publications we print another EPROM downloader — see Fig. 4.

The 80C32 can work with much faster clocks than the 8052AH-BASIC: according to the manufacturer, it is capable of operating at a clock of 16 MHz. A couple of our prototypes however worked fine at a clock of 24 MHz. By contrast, the HMOS 8052AH-BASIC threw in the towel at about 15 MHz.

Power supply

The unregulated power supply voltage to the SBC should normally lie between 8 V and 12 V. To enable the 12.5-V stabilizer around IC_1 and T_5 to operate correctly, a minimum supply voltage of about 16 V is required when an EPROM is to be programmed. Provided IC_4 is adequately cooled, the SBC may be powered permanently with 16 V obtained from a simple mains supply: a 12-V transformer, a bridge rectifier and a 1,000 µF capacitor should do the job.

The current consumption of the SBC depends on the ICs fitted. When the CMOS 80C32 is used, you can expect a current consumption between 50 mA and 150 mA. The HMOS CPUs (8032 and 8052AH-BASIC) will require the power supply to deliver more than 300 mA. Depending on the type of EPROM installed (CMOS or non-CMOS), add another 100 mA or so to the current requirement.

Practical use

The microcontroller has its own clock generator which operates in conjunction with a quartz crystal. Remove inductor L_1 when a quartz crystal specified for fundamental frequency resonance is used, as with the 8052AH-BASIC, which will not go faster than 15 MHz or so. Since most crystals of 20 MHz and higher are overtone types, L_1 must be fitted when a 24-MHz type is used with a 80C32 processor. A 1.5-µH inductor then prevents the crystal resonating at its fundamental frequency of 8 MHz.

In case the computer does not function spot on, the power-on reset capacitor, C_9, may have to be changed from 220 nF into a 4.7 µF tantalum type. This may be necessary when the supply voltage rises too slow at power on.

You will need software to program the computer, and to use the computer in conjunction with a terminal. MCS-52 assemblers

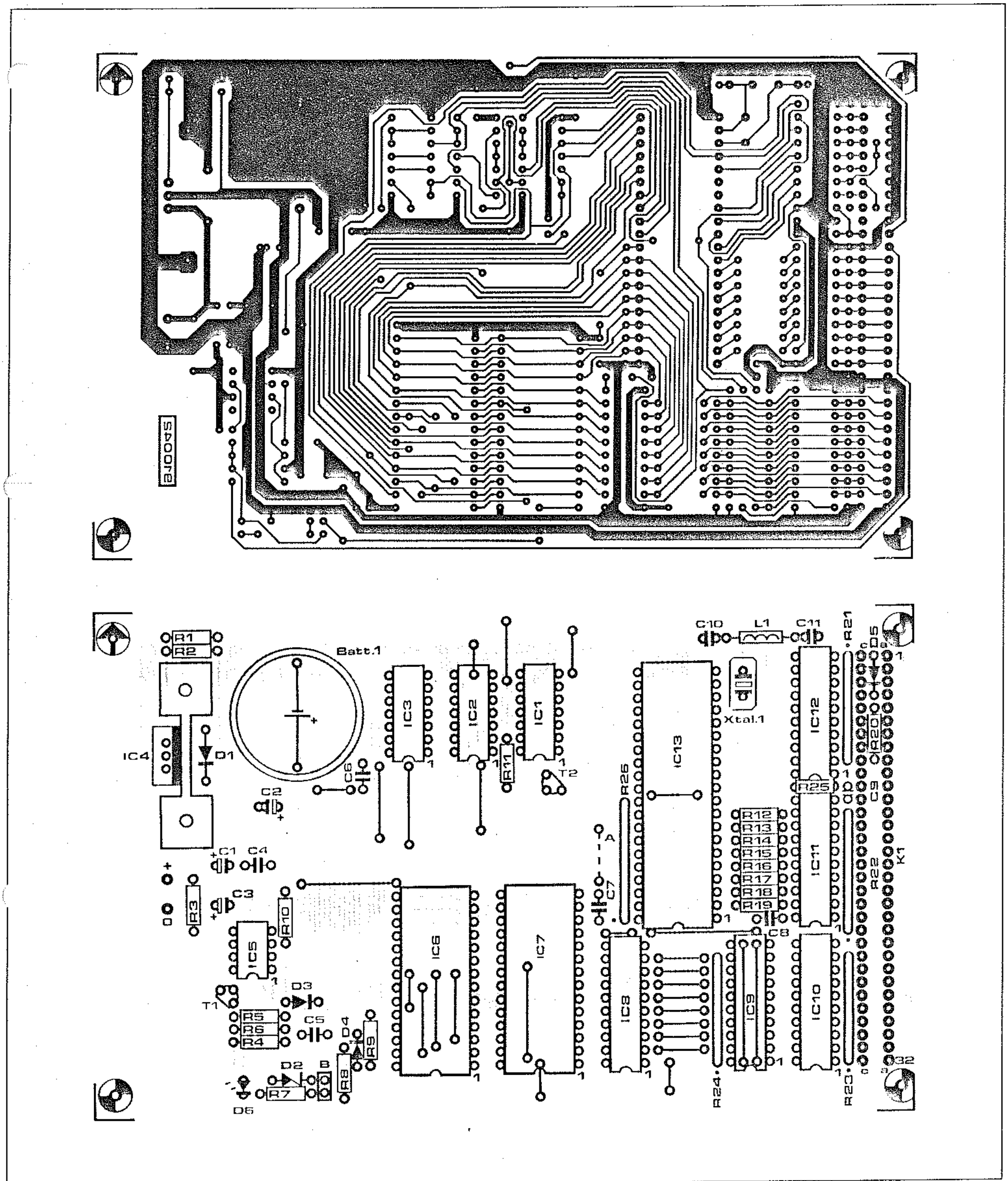


Fig. 5. Track layout (mirror image) and component mounting plan of the *single-sided* printed circuit board for the computer.

and cross-assemblers are available for use on a PC, and the object code may be transferred to the SBC by means of an EPROM. If you want to avoid the hassle of burning EPROMs, debugging the program, erasing the EPROM and programming it again, consider the use of an EPROM emulator which forms a direct link between the PC used to develop the program and the target system, in this case, the SBC.

By virtue of its internal BASIC interpreter, the 8052AH-BASIC is much simpler to get going than the 80C32: connect the terminal or PC to the SBC via RX, TX and ground, set a baud rate of anything between 300 and 19,200, reset the SBC and press the space bar to initiate the automatic baud rate timing. The message

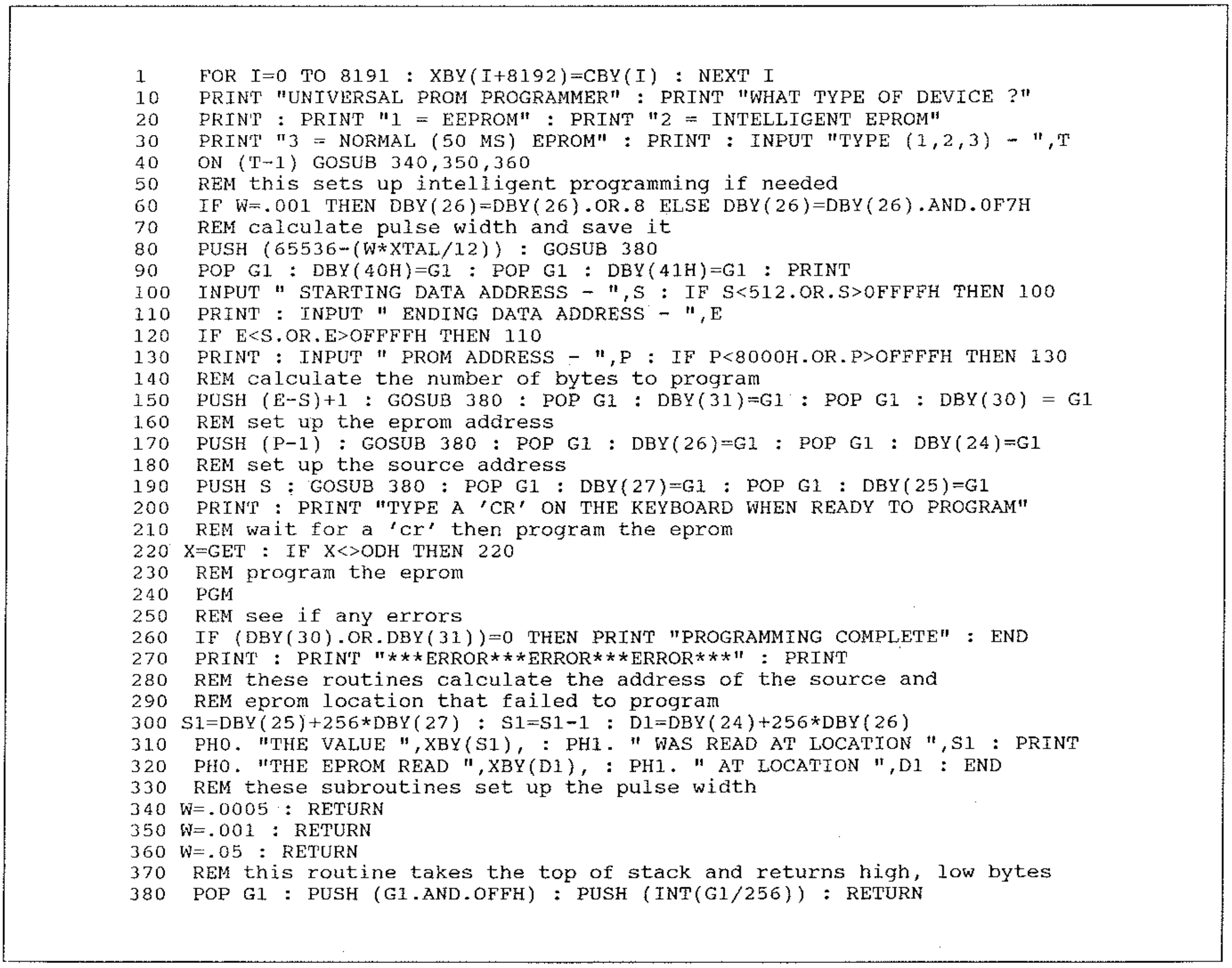
```
*MCS-51 (tm) BASIC V1.1
READY
```

should appear on the console display, and you are ready to start entering or downloading a BASIC program.

One final note: when the 8052AH-BASIC is used, do not suspect a malfunction of the memory backup circuit if you find that your program has disappeared. The BASIC inter-

COMPONENTS LIST

Resistors:		
1	470Ω	R1
1	150Ω	R2
1	47Ω	R3
1	2kΩ2	R4
10	2kΩ7	R5;R8;R12-R19
1	10kΩ	R6
1	560Ω	R7
3	1kΩ	R9;R10;R11
1	15kΩ	R20
2	9-way 10kΩ SIL	R22;R26
3	8-way 10kΩ SIL	R21;R23;R24
1	1MΩ	R25
Capacitors:		
1	10µF 35V radial	C1
2	100µF 16V radial	C2;C3
6	220nF	C4-C9
2	33pF ceramic	C10;C11
Semiconductors:		
1	BC337	T1
1	BS170	T2
1	1N4004	D1
1	4V7 0.4 W zener diode	D2
3	1N4148	D3;D4;D5
1	green LED	D6
3	74HC00	IC1;IC2;IC3
1	LM317	IC4
1	TL081	IC5
1	62256-LP2	IC6
1	27C256-2	IC7
2	74HC573	IC8;IC9
2	74HC245	IC10;IC11
1	74HCT240	IC12
1	80C32 or 8052AH-BASIC	IC13
Miscellaneous:		
1	1µH5 choke	L1
1	64-way a-c row DIN connector with angled solder pins	K1
1	12MHz, 15MHz or 24MHz quartz crystal	X1
1	3V lithium battery (button cell) for PCB mounting	Bat1
1	TO220 style heat-sink	
1	printed-circuit board	910042



```
1     FOR I=0 TO 8191 : XBY(I+8192)=CBY(I) : NEXT I
10    PRINT "UNIVERSAL PROM PROGRAMMER" : PRINT "WHAT TYPE OF DEVICE ?"
20    PRINT : PRINT "1 = EEPROM" : PRINT "2 = INTELLIGENT EPROM"
30    PRINT "3 = NORMAL (50 MS) EPROM" : PRINT : INPUT "TYPE (1,2,3) - ",T
40    ON (T-1) GOSUB 340,350,360
50    REM this sets up intelligent programming if needed
60    IF W=.001 THEN DBY(26)=DBY(26).OR.8 ELSE DBY(26)=DBY(26).AND.0F7H
70    REM calculate pulse width and save it
80    PUSH (65536-(W*XTAL/12)) : GOSUB 380
90    POP G1 : DBY(40H)=G1 : POP G1 : DBY(41H)=G1 : PRINT
100   INPUT " STARTING DATA ADDRESS - ",S : IF S<512.OR.S>0FFFFH THEN 100
110   PRINT : INPUT " ENDING DATA ADDRESS - ",E
120   IF E<S.OR.E>0FFFFH THEN 110
130   PRINT : INPUT " PROM ADDRESS - ",P : IF P<8000H.OR.P>0FFFFH THEN 130
140   REM calculate the number of bytes to program
150   PUSH (E-S)+1 : GOSUB 380 : POP G1 : DBY(31)=G1 : POP G1 : DBY(30) = G1
160   REM set up the eprom address
170   PUSH (P-1) : GOSUB 380 : POP G1 : DBY(26)=G1 : POP G1 : DBY(24)=G1
180   REM set up the source address
190   PUSH S : GOSUB 380 : POP G1 : DBY(27)=G1 : POP G1 : DBY(25)=G1
200   PRINT : PRINT "TYPE A 'CR' ON THE KEYBOARD WHEN READY TO PROGRAM"
210   REM wait for a 'cr' then program the eprom
220 X=GET : IF X<>0DH THEN 220
230   REM program the eprom
240   PGM
250   REM see if any errors
260   IF (DBY(30).OR.DBY(31))=0 THEN PRINT "PROGRAMMING COMPLETE" : END
270   PRINT : PRINT "***ERROR***ERROR***ERROR***" : PRINT
280   REM these routines calculate the address of the source and
290   REM eprom location that failed to program
300 S1=DBY(25)+256*DBY(27) : S1=S1-1 : D1=DBY(24)+256*DBY(26)
310   PH0. "THE VALUE ",XBY(S1), : PH1. " WAS READ AT LOCATION ",S1 : PRINT
320   PH0. "THE EPROM READ ",XBY(D1), : PH1. " AT LOCATION ",D1 : END
330   REM these subroutines set up the pulse width
340 W=.0005 : RETURN
350 W=.001 : RETURN
360 W=.05 : RETURN
370   REM this routine takes the top of stack and returns high, low bytes
380   POP G1 : PUSH (G1.AND.0FFH) : PUSH (INT(G1/256)) : RETURN
```

Fig. 6. Type in this listing if you want to unload the BASIC interpreter from the 8052AH-BASIC CPU, transfer it to EPROM and run it with a 8032 or 80C32. The advantages: lower cost and higher speed. The pulse timing in this program is based on a clock of 11.0592 MHz.

preter on reset runs a memory test that clears all of the RAM content. When the program is transferred to EPROM, you have the option to use the programming mode PROG3, which forces the interpreter to clear the memory up to the value indicated by MTOP. All data above MTOP is retained. ■

References:

1. "BASIC computer". *Elektor Electronics* November 1987.
2. "CMOS replacement for 8052AH-BASIC". *Elektor Electronics* January 1990.
3. "ROM-copy for 8052-BASIC computer". *Elektor Electronics* September 1990.

For further reading:

1. Microcontroller Handbook, Intel Corp. 1984.
2. MCS® BASIC-52 users manual, Intel Corp. 1984, order no. 270010-001.
3. "LCD for 8052 microcontroller". *Elektor Electronics* Supplement July/August 1990.