BASIC COMPUTER

At the heart of this versatile and simple to build computer for process control and automation applications is Intel's Type 8052AH-BASIC microcontroller.

As already noted in reference [1], the Type 8052AH-BASIC V1.1 is a single-chip microcontroller tailored to data manipulation in intelligent instrumentation, measurement and control systems. Not surprisingly, therefore, the 8052AH-BASIC features an extensive and powerful set of input/output and timekeeping functions.

By virtue of its compactness and ease of programming, the BASIC computer described here is suitable for a wide range of domestic as well as industrial applications. Although not every programmer will applaud the use of BASIC, it can be argued that this is still the most widely known, and often first apprehended, programming language. Moreover, the BASIC interpreter of the 8052AH-BASIC is an advanced version offering instructions like DO-WHILE and DO-UNTIL which enable better structuring of programs than the GOTO statement. Also, variables can be stored and retrieved by means of instructions PUSH and POP. The BASIC interpreter is reasonably fast as compared with competetive 8 and 16 bit systems. In conclusion, the 8052AH-BASIC couples the power and versatility of the 8051 to the qualities of a well-written, reasonably fast, BASIC interpreter.

The computer described is suitable for experimental as well as stand-alone applications. Programs can be written, tested, and debugged by anyone with a reasonable command of BASIC. The microcontroller used is not cheap, probably because of its specialist nature, and the fact that it has hitherto found applications mainly in industrial control systems. None the less, the cost of the 8052AH-BASIC is justifiable considering its impressive potential.

To aid programmers in writing efficient programs, Intel supplies the indipensable *MCS BASIC-52 USERS MANUAL*, which carries reference number 270010-003.

1

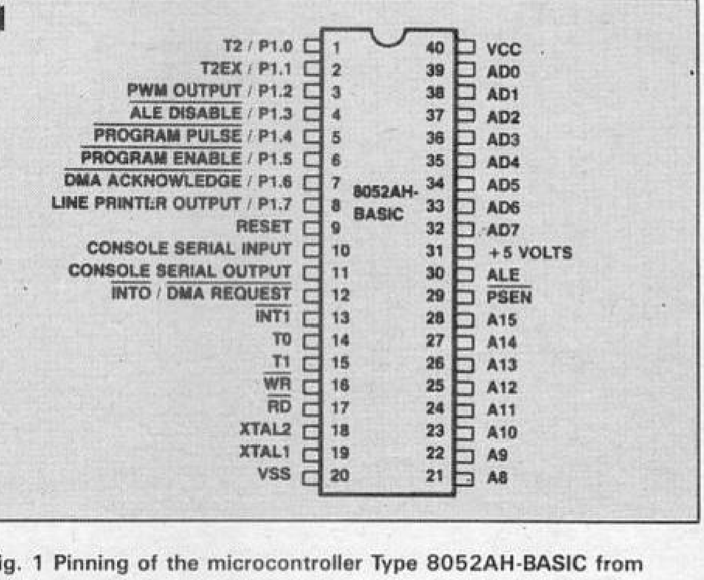


Fig. 1 Pinning of the microcontroller Type 8052AH-BASIC from Intel.

It is important to note that ready-made programs for the BASIC computer are not available. The proposed system is intended primarily for applications where the BASIC programs are not an end in themselves, but where the hardware-software link is readily accessible to enable developing and testing computer controlled systems of a wide variety. Once a program is debugged and known to function satisfactorily, the computer can act as a reliable stand-alone controller.

Features

The computer described features an on-board EPROM programmer, which is controlled direct by the 8052AH-BASIC CPU. This means that the processor can store its own programs in EPROM after debugging and testing. Once it is EPROM resident, the BASIC program is available for direct

2

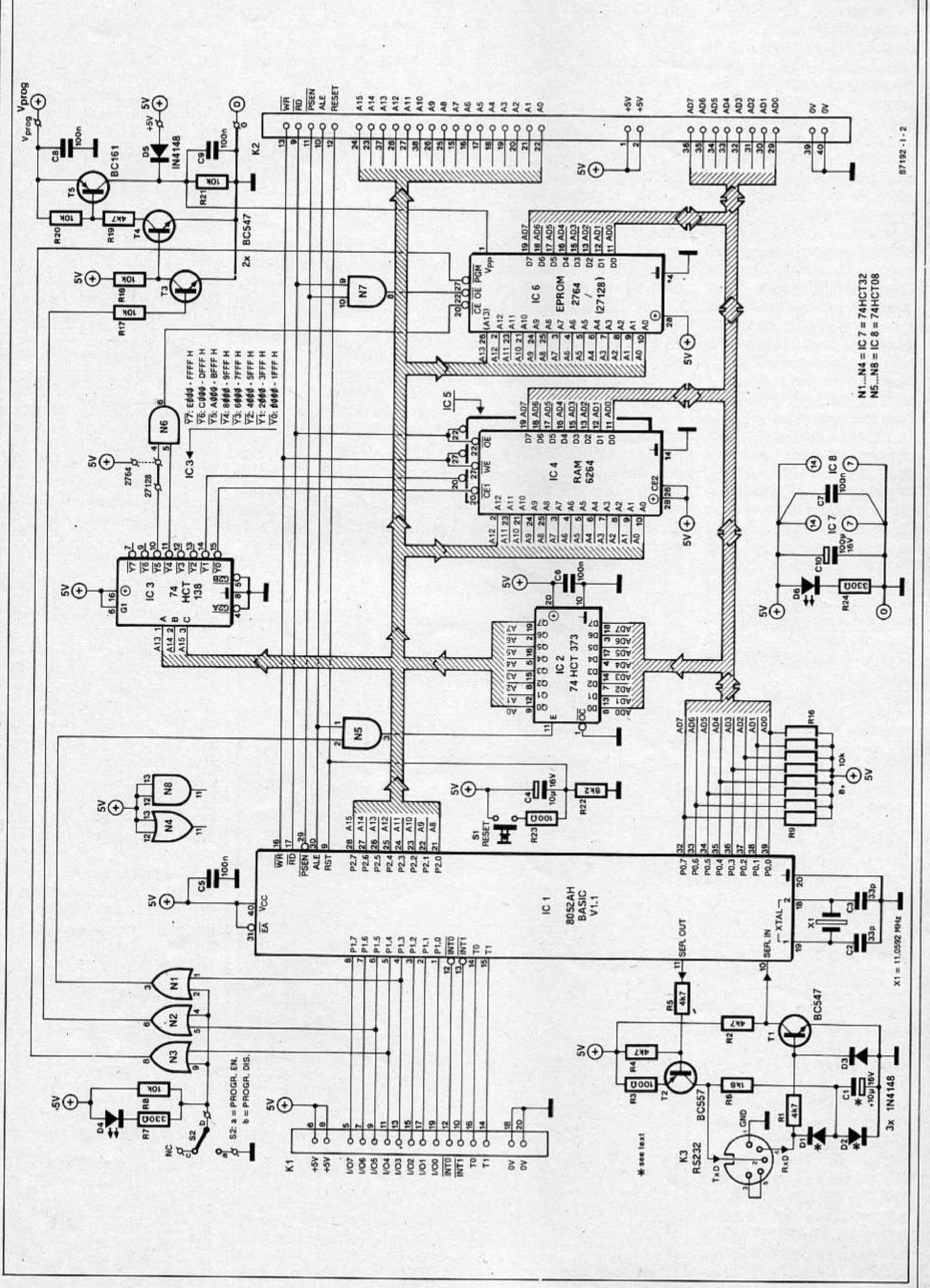


Fig. 2 Circuit diagram of the BASIC computer.

and autonomous execution by the processor. The EPROM contents form the token program listing rather than machine code obtained by a compiling process. The programming of EPROMs on the board is straightforward, and fully supported by BASIC instructions. A single EPROM can hold a number of programs, which can even call each other when necessary.

It should be noted that the BASIC computer has no keyboard and screen of itself. These communication functions are taken over by an external console (terminal), connected to the computer's bidirectional, serial I/O port.

As to the hardware configuration of the proposed BASIC computer, this is characterized by a high degree of flexibility, allowing the user to readily add, say, a UART (*universal asynchronous receiver/transmitter*), an ACIA (*asynchronous communications interface adapter*), a number of PIAs (*peripheral interface adapter*), or other peripheral circuitry such as an alphanumerical display, a sound generator, or a keyboard encoder. The pinning of the 8052AH-BASIC is given in Fig. 1.

The 8052AH-BASIC has a number of powerful timing instructions which, in conjunction with the interrupt statements, special registers, and instruction counters, afford excellent control of time critical I/O applications. A real time clock is also available in the form of function TIME, which offers a resolution of about 5 ms.

The Type 8052AH-BASIC is an 8 bit *microcontroller*, which means that it combines the functions of central processing unit (CPU), and peripheral circuits (I/O; DMA). The chip has an accumulator A, a register B, a status register PSW (*program status word*), an 8 bit stack pointer, a 16 or 2×8 bit data pointer DPTR, 4 8 bit ports for use as an I/O and/or address, data, or command bus, a double serial communication register SBUF, 3 register pairs TH0-TL0, TH1-TL1 and TH2-TL2, which together form the 3 16 bit timers T0, T1 and T2, an intermediate storage register pair RCAP2H-RCAP2L for a number of functions of timer 2, and, finally, an array of registers for various command functions: IP (*interrupt priority*), IE (*interrupt enable*), TMOD, TCON & T2CON for the timers, SCON (*serial control*) and PCON (*power control*).

Circuit description

The circuit diagram of the BASIC computer is given in Fig. 2. The 8 Kbyte BASIC interpreter is internal to the microcontroller, IC_1. EPROM IC_6 holds the user's BASIC programs. The minimum amount of RAM for the 8052AH-BASIC is 1 Kbyte starting at address 0000. In the present application, the RAM area is either 8 Kbyte (0000—1FFF) or 16 Kbyte (0000—3FFF), depending on whether 1 or 2 RAMs Type 6264 are fitted (IC_4; IC_5). Write and read operations are controlled direct by signals $\overline{WR}$ and $\overline{RD}$ respectively.

The memory structure of the 8052AH-BASIC is not in accordance with von Neumann's model: the program memory is distinct from the data memory, which explains the logic combination of signal $\overline{PSEN}$ (*program store enable*: control of read operations in an external program memory) with $\overline{RD}$ in gate N_7 to select the ROM memory area (2764 = 8 Kbyte from 8000 to 9FFF; 27128 = 16 Kbyte from 8000 to BFFF). This does not exhaust all the possible memory configurations for the 8052AH-BASIC, but forms a practical as well as efficient combination—see Fig. 3. In the EPROM programming mode, the microcontroller addresses EPROMs in the memory area starting at address 8000.

Decoder IC_3 divides the memory area in blocks of 8 Kbyte. AND gate N_6 makes it possible to combine 2 block select signals when the EPROM used is a Type 27128. Normally, octal latch IC_2 demultiplexes the data and lower address bytes with the aid of signal ALE (*address latch enable*). In the EPROM programming mode, however, the LS address byte is kept latched much longer than during normal bus cycles.

3

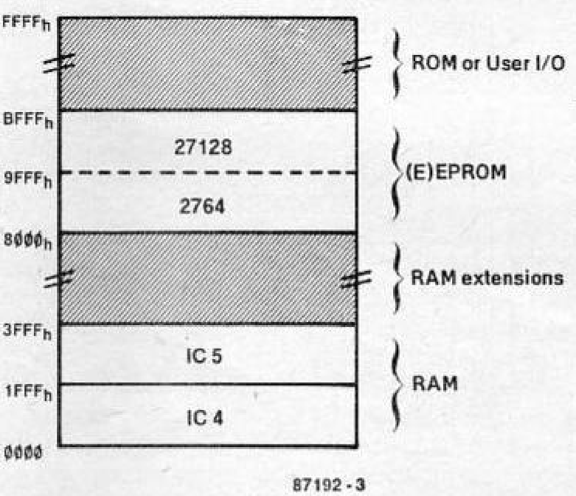


Fig. 3 Memory structure of the 8052AH-BASIC.

Table 1.

Connector K_1:		Connector K_2:			
Pin	Pin	Pin	Pin	Pin	Pin
1 NC	2 NC	1 +5 V	11 $\overline{PSEN}$	21 A1	31 D2
3 NC	4 NC	2 +5 V	12 RESET	22 A0	32 D3
5 I/O7	6 +5 V	3 NC	13 $\overline{WR}$	23 A14	33 D4
7 I/O6	8 +5 V	4 NC	14 NC	24 A15	34 D5
9 I/O5	10 $\overline{INT1}$	5 NC	15 A7	25 A8	35 D6
11 I/O4	12 $\overline{INT0}$	6 NC	16 A6	26 A9	36 D7
13 I/O3	14 T1	7 NC	17 A5	27 A11	37 A13
15 I/O2	16 T0	8 NC	18 A4	28 A12	38 A10
17 I/O1	18 ⊥	9 $\overline{RD}$	19 A3	29 D0	39 ⊥
19 I/O0	20 ⊥	10 ALE	20 A2	30 D1	40 ⊥

Table 1 Pinning of connectors K_1 and K_2.

This also goes for the MS address byte and the dataword—the normal duration of a programming cycle is of the order of 50 ms. The software has no direct control over the length of the ALE pulse, and this is, therefore, inhibited with the aid of N_1, N_5 and the logic low level on CPU output P1.3.

When port 0 is used in the I/O mode, pull-up resistors are required on the open drain outputs. Normally, this port functions as the data & address bus, but operates as an I/O port in the EPROM programming mode.

The TTL levels at the serial output, P3.1, of the microcontroller are converted into the corresponding positive and negative levels for the terminal. Rectifier D_1-D_2-C_1 is connected to the terminal's TXD line to provide the negative supply for TXD driver T_2. Components D_1 and D_2 can be omitted, and C_1 replaced by a wire link, when the terminal accepts and sends pulses with TTL levels.

The connections on the serial I/O connector, K_3, are given in the circuit diagram.

Table 1 shows the pin assignment on connector K_1, which carries the 8 lines of peripheral port P1, interrupt inputs $\overline{INT0}$ and $\overline{INT1}$, and lines T0 and T1, which form the external inputs of the respective timers. Line pairs $\overline{WR}$ and $\overline{RD}$, RxD and TxD, $\overline{INT0}$ and $\overline{INT1}$, and T0 and T1 together form port P3 of the 8052AH-BASIC. Apart from their normal use as I/O lines, the lines on port P1 may be used for special purposes. For example, P1.0 and P1.1 can provide trigger as well as clock pulses for timer T2. This is a standard function of the 8052, and not a particular feature of the BASIC interpreter. Lines P1.3, P1.4 and P1.5 are used for programming the majority of currently available EPROM and EEPROMs Type 2764 and 27128. Output P1.6 is connected to input INT0 for ready implementation of a DMA (*direct memory access*) mechanism. Output P1.7 can act as a direct serial channel for driving, say, a printer, controlled with the aid of commands LIST# and PRINT#. There are more BASIC instructions for port 1: PWM, for example, offers control of the pulsewidth on output P1.2,

while instruction PORT1 enables direct read/write access.
The signal assignment on connector K_2 is shown in Table 1. This connector carries lines AD0...AD7, A0...A15, and the command bus, and so enables ready connection of peripheral extension, or DMA, circuitry. It is possible to halt the processor in the *idle* mode, and so arrange for an external processor or microcontroller to temporarily gain access to the memory in the BASIC computer. The *idle* mode is initiated with the aid of the corresponding BASIC statement, and can be used for switching the microcontroller to the non-active state when no action on its part is required.
The clock oscillator is internal to the 8052AH-BASIC, and merely requires a quartz crystal and 2 capacitors. The indicated crystal frequency of 11.0592 MHz is required to ensure the correct timing for the serial channel, the real time clock, and the EPROM programming pulses. When it is intended to use, say, a 12 MHz crystal, the processor should be informed of this by declaring XTAL=12000000. It should be noted that any oscillator frequency other than 11.0592 MHz may result in reduced accuracy of the counter operations.
The computer is reset and initialized on power up either automatically (R_{22}-C_4) or manually (S_1). Input $\overline{EA}$ (*external address*) is made permanently logic high because the BASIC interpreter is an *internal* memory area.

Programming EPROMs

The (E)EPROM programming facility of the present BASIC computer is, without doubt, one of its most attractive features. It is important to note that the computer is not just an EPROM programmer, but a data handling and storage system that can be customized as required for the application in question. While communicating with the user via the terminal, the 8052AH-BASIC can store edited, debugged and tested BASIC (sub)routines in EPROM to facilitate calling these as "tools" at any time. Before programming is effected, the microsoftware in the 8052AH-BASIC takes care af all the tokenizing of the object program to ensure compact storage. Depending on the programming mode, certain parameters are stored along with the program, and are instantly available when this is loaded and run. These program parameters include the baud rate, variable MTOP, an autoexecute flag, and a flag that enables skipping the memory initialization routine at power-on—this is particularly useful when the RAM is battery powered.
Finally, it is possible to use BASIC for loading an EPROM with an assembler program that is executed automatically after a RESET pulse.

With reference to the circuit diagram, when line P1.5 goes low, transistors T_3, T_4 and T_5 ensure that the programming voltage reaches the Vpp terminal of the EPROM. The programming voltages for a number of EPROMs are listed in Table 2. The microcontroller places the LS address byte onto lines AD0...AD7, then disables ALE by making P1.3 logic low. The address byte remains latched in IC_2 during the remainder of the programming cycle. The MS address byte is placed onto lines A8...A15, and the databyte onto lines D0...D7 of the EPROM to be programmed. Then, output P1.4 is made logic low, and the byte is programmed in the EPROM because $\overline{PGM}$ goes low while Vpp is applied. Instructions PROG and FPROG select a duration of the programming cycle of 50 and 1 ms, respectively. FPROG uses the intelligent programming algorithm, and may require raising the EPROM supply voltage from 5 to 6 V, which is *not* supported by the proposed circuit. Details on the intelligent programming algorithm can be found in reference [2]. In all cases, the duration of the $\overline{PGM}$ pulse is determined by the clock frequency of the microcontroller, and operator XTAL should be defined as as discussed previously. Switch S_2 enables blocking the 3 programming signals. This is done for reasons of security because port P1 can be used for purposes other than programming EPROMs.

Up to 255 BASIC modules can be held in a single EPROM, and each of these can call any of the others. The 8052AH-BASIC automatically assigns a number to each BASIC program before storing this in EPROM. The number is sent to the terminal for the programmer's reference. Loading and running a particular BASIC module is effected with the aid of commands ROM X followed by RUN. Variable X is the number of the relevant module. Modules can be copied from EPROM to RAM by means of command XFER.
The programmer has direct access to an extensive library of routines in the BASIC interpreter. Also, BASIC allows calling external machine code subroutines provided by the user. It should be noted, though, that writing (fast) machine code requires an 8051 assembler, and, of course, considerable experience in working at the assembly code level.

Table 2.

Manufacturer	Type	memory organization	Vpp
AMD	AM2764	8K × 8	21 V
	AM2764A	8K × 8	12.5 V
	AM27128	16K × 8	21 V
	AM27128A	16K × 8	12.5 V
Fujitsu	MBM2764	8K × 8	21 V
	MBM27C64	8K × 8	21 V
	MBM27128	16K × 8	21 V
Hitachi	HN482764	8K × 8	21 V
	HN27C64	8K × 8	21 V
	HN482764P	8K × 8	21 V
	HN4827128	16K × 8	21 V
	HN27128P	16K × 8	21 V
Intel	2764	8K × 8	21 V
	P2764	8K × 8	21 V
	2764A	8K × 8	12.5 V
	27C64	8K × 8	12.5 V
	P2764A	8K × 8	12.5 V
	27128	16K × 8	21 V
	27128A	16K × 8	12.5 V
	P27128A	16K × 8	12.5 V
Mitsubishi	M5L2764	8K × 8	21 V
	M5L27128	16K × 8	21 V
National Semiconductor	NMC27C64	8K × 8	12.5 V
	NMC27CP128	16K × 8	12.5 V
NEC	μPD2764	8K × 8	21 V
	μPD27C64	8K × 8	21 V
	μPD2764C	8K × 8	21 V
	μPD27C64C	8K × 8	21 V
	μPD27128	16K × 8	21 V
	μPD27128C	16K × 8	21 V
Rockwell	R87C64	8K × 8	21 V
	R27C64P	8K × 8	21 V
SEEQ	2764	8K × 8	21 V
	27128	16K × 8	21 V
SGS/ATES	M2764	8K × 8	21 V
Texas Instruments	TMS2564	8K × 8	25 V
	TMS2764	8K × 8	21 V
	TMS27128	16K × 8	21 V
Thomson-CSF	ET2764	8K × 8	21 V
Toshiba	TMM2764	8K × 8	21 V
	TMM2764DI	8K × 8	21 V
	TMM27128	16K × 8	21 V

The type indications as given may be followed by an access time specification.

Table 2 Programming voltages for a number of EPROM types that can be loaded by the BASIC computer.

The practical use and operation of the EPROM programming facility is extremely straightforward. All that is required is to fit an EPROM in the socket for IC_6, apply the correct programming voltage, switch S_2 to PROG. EN, load the BASIC file in RAM, and issue command PROG. The other programming commands, (F)PROG1...(F)PROG6 enable storing auxiliary program information, including the baud rate indicator, and the autoexecute flag. The available options are described in the previously mentioned programming manual from Intel.

Construction

It should be reiterated that the computer described is intended mainly as an aid in developing software and hardware for automated processes and stand-alone, intelligent, controllers or data loggers, where timekeeping is an essential requirement.

The printed circuit board for the BASIC computer is double-sided and through-plated. The component mounting plan is given in Fig. 4.

It is recommended to fit good quality sockets for all ICs. The socket for EPROM IC_6 can be a type with turned pins, although a ZIF (*zero insertion force*) socket mounted as shown in the photograph of the prototype is probably the best solution. Be sure to purchase a microcontroller Type **8052AH-BASIC V1.1**. Connectors K_1 and K_2 are intended for extensions, and need not be fitted as yet. Initially, a single RAM, IC_4, is sufficient, since it offers a memory area of about 7 Kbyte for BASIC programs. Resistors R_9...R_{16} incl. form an 8-way SIL network, but it is also possible to use 8 ordinary resistors, mounted vertically and commoned by a short length of wire connected to +5 V as shown in Fig. 5. The function of the LEDs, D_4 and D_6, is evident from the circuit diagram. The supply and programming voltage are applied to the circuit via soldering pins and mating sockets, insulated with the aid of heat shrink sleeving. *Do not confuse the Vcc and Vpp connections.* The PROG. EN switch, S_2, and the EPROM selector, S_3, may each be replaced by 3 pins and a mating jumper if it is not intended to frequently program EPROMs, or change between a 2764 and a 27128.

EPROM IC_6 is not required to make the circuit function. It is not fitted until it can be pro-

4

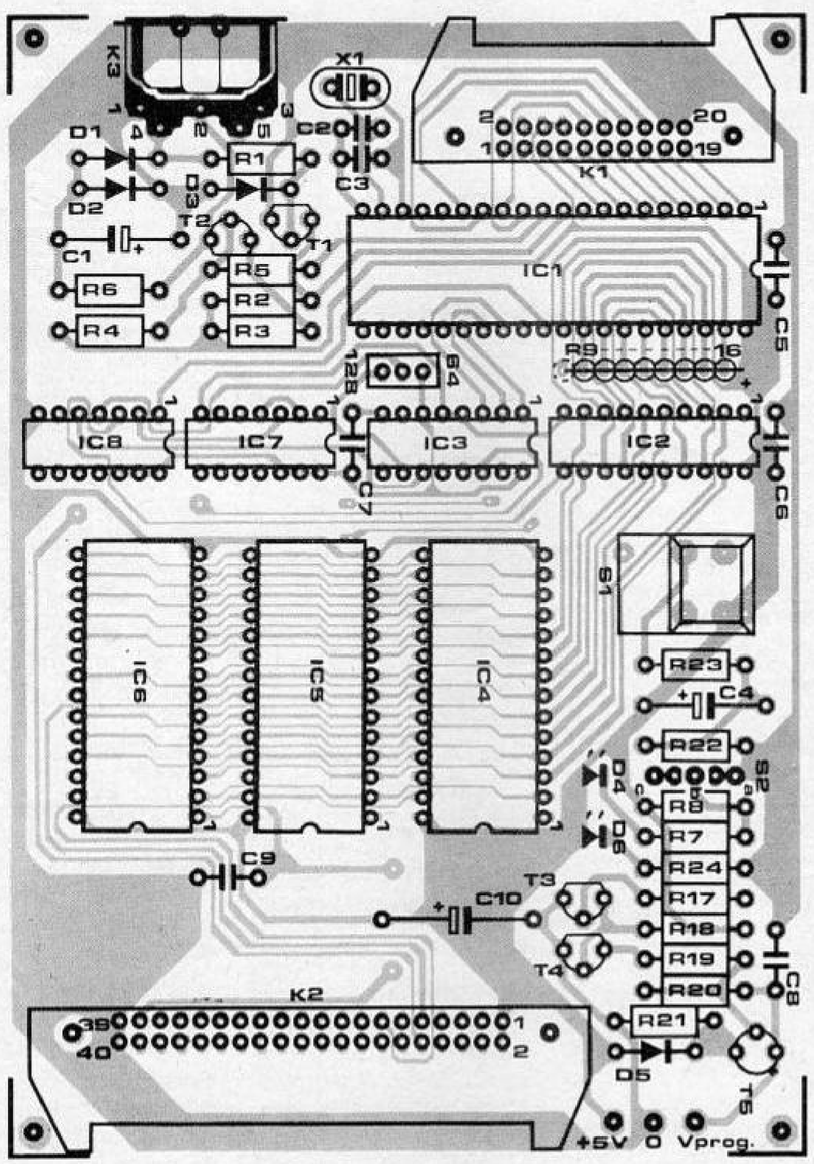


Fig. 4 Component mounting plan for the BASIC computer. The circuit board is available ready-made through the Readers Services.

Parts list

Resistors (±5%):
R_1;R_2;R_4;R_5;R_{19} = 4K7
R_3;R_{23} = 100R
R_6 = 1K8
R_7;R_{24} = 330R
R_8;R_{17};R_{18};R_{20};R_{21} = 10K
R_9...R_{16} = 8-way 10K SIL network, or 8 10K resistors
R_{22} = 8K2

Capacitors:
C_1;C_4 = 10µ; 16 V
C_2;C_3 = 33p ceramic
C_5...C_9 incl. = 100n
C_{10} = 100µ; 16 V

Semiconductors:
D_1;D_2;D_3;D_5 = 1N4148
D_4 = green LED
D_6 = red LED
T_1;T_3;T_4 = BC547
T_2 = BC557
T_5 = BC161
IC_1 = 8052AH-BASIC VERSION 1.1*

IC_2 = 74HCT373
IC_3 = 74HCT138
IC_4;IC_5 = 6264 8Kx8 static CMOS RAM
IC_6 = 2764 or 27128 (see text)
IC_7 = 74HCT32
IC_8 = 74HCT08

Miscellaneous:
S_1 = Digitast SPST push button.
S_2 = miniature SPST switch.
K_1 = 20-way right angled IDC header with side latches.
K_2 = 40-way right angled IDC header with side latches.
K_3 = 5-way DIN socket for PCB edge mounting.
X_1 = 11.0592 or 11.059 MHz, HC18 enclosure.
28-way ZIF socket.
Jumpers and soldering pins as required.
PCB Type 87192 (available through the Readers Services).
Suitable ABS or metal enclosure.
Suitable power supply.

It is regretted that a ready-made front panel for this project is not available.

* Intel distrubutors are listed on InfoCard 505 in the March 1987 issue of *Elektor Electronics*.

The chip is also available from Universal Semiconductor Devices Limited • 17 Granville Court • Granville Road • Hornsey • London N4 4EP. Telephone: (01 384) 9420. Telex: 25157 usdco g. Fax: 01 348 9425.

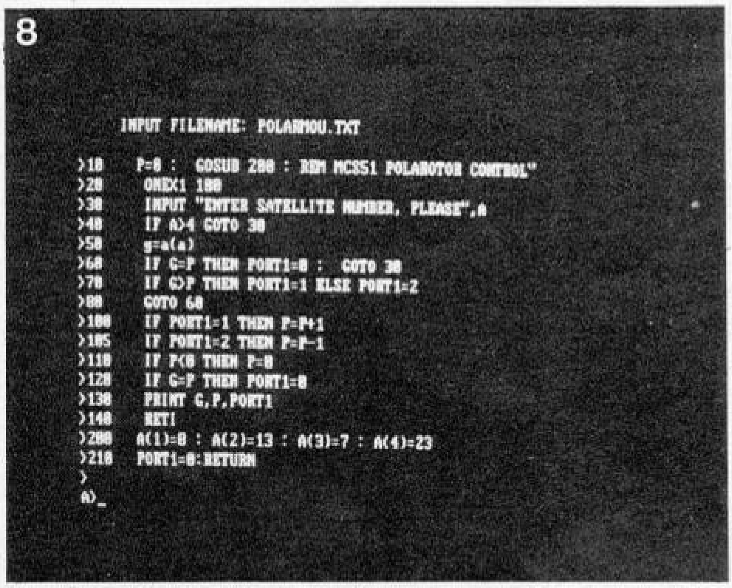


Fig. 8 SENDBAS.COM has completed sending a program to the BASIC computer via the COM1: port on a PC turbo XT. The baud rate is 1200.

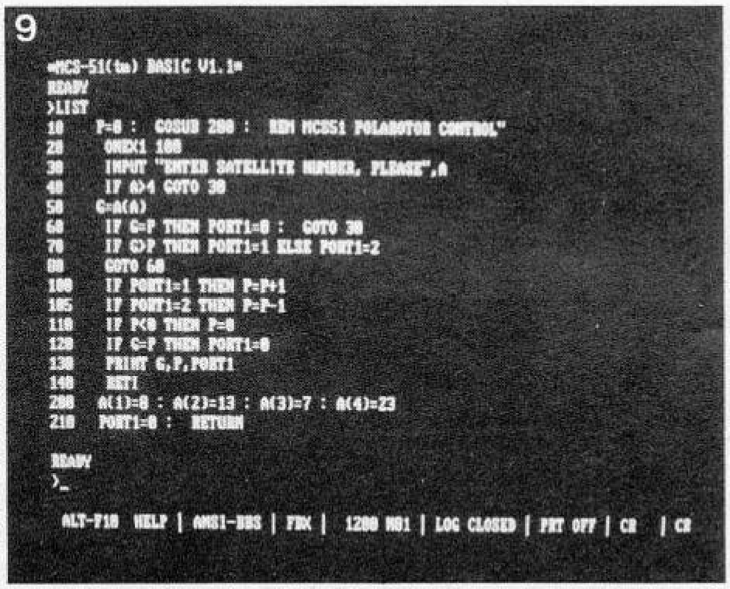


Fig. 9 The BASIC computer is on line again, and has received a program for controlling a polarmount satellite dish position system. Note the system's welcome message at the top of the screen, and the status line of PROCOMM® at the bottom.

SENDBAS.COM was tested in conjunction with PROCOMM® 2.4.2, a versatile communication program for PCs and compatibles. BASIC text files were prepared and stored onto disk in DOS text format using the wordprocessor WORDPERFECT 4.2. Other combinations of communication program and wordprocessor should also work, as long a the files for sending to the BASIC computer are written in DOS text (ASCII) format, i.e., without all the control codes specific to the wordprocessor used. As to the communication program, it is very practical if this offers a SHELL or DOS Gateway command to temporarily switch to DOS, start SENDBAS for loading the updated file, and return to the BASIC computer by means of EXIT. SENDBAS takes over the set baud rate, and awaits the > prompt from the computer before it sends a new line via COMl:. The writing of the file can be seen on the screen. After sending a file using SENDBAS, and EXITing DOS to return to the comms program, type a <CR> when the BASIC computer displays

READY
\>

Type LIST to check the contents of the new program, and run it... The use of SENDBAS.COM on a PC-XT turbo is illustrated in Figs. 8 and 9.

A simple filehandler for the BBC micro is listed in Table 5. This program works in conjunction with the well-known wordprocessor VIEW, the micro's serial outlet and the communication program COMMUNICATOR, set up for VT52 emulation, XON/OFF, and, say, 9600 baud I/O. It is assumed here that the user is thoroughly familiar with these programs, and the way they are called up and exited. Test the communication between the BBC micro and the BASIC computer by pressing RESET and then the space bar as outlined above. Owners of a MASTER micro can avail themselves of the built-in terminal, obviating the need to purchase a separate communication program.

Leave COMMUNICATOR, run BASIC, and enter the listing of Table 5. Run the program. It creates a small machine code routine called PRDR-52 (*printer driver* for 8052AH-BASIC), which is automatically saved onto disk. Select the computer's serial output channel by typing **FX5,2**. Call up VIEW (*W), and load or write the program (i.e. text file) for the BASIC computer. Install PRDR-52 on the serial port by typing **PRINTER PRDR-52** at the command level. The VIEW file is now sent to the BASIC computer at the specified baud rate. The fact that VIEW can not send but complete pages is of no consequence. Leave VIEW and run the terminal emulation program to control the BASIC computer direct.

Th

Table 5.

```
 LIST
 10 MODE 7
 20 FOR ADDRESS=&4200 TO &426B
 30  READ BYTE
 40  ?ADDRESS=BYTE
 50 NEXT ADDRESS
 60 *SAVE PRDR-52 4200 4300 0400 0400
 70 END
 80 DATA &4C.&40.&04.&4C.&0F.&04.&4C.&26.&04.&4C.&3C.&04.&4C.&3D.&04
 90 DATA &48.&8A.&48.&98.&48.&A9.&02.&20.&EE.&FF.&A9.&02.&A2.&01.&20
100 DATA &F4.&FF.&68.&A8.&68.&AA.&68.&60.&48.&8A.&48.&98.&48.&A9.&03
110 DATA &20.&EE.&FF.&A9.&02.&A2.&00.&20.&F4.&FF.&68.&A8.&68.&AA.&68
120 DATA &60.&A0.&00.&60.&8D.&FE.&04.&48.&8A.&48.&98.&48.&AD.&FE.&04
130 DATA &C9.&0D.&F0.&09.&20.&EE.&FF.&68.&A8.&68.&AA.&68.&60.&20.&E7
140 DATA &FF.&20.&E0.&FF.&C9.&0A.&D0.&F9.&20.&E0.&FF.&C9.&3E.&D0.&F2
150 DATA &4C.&52.&04
>
```

87192-T5

Table 5 This program creates PRDR-52, the filehandler for the BBC micro running VIEW and COMMUNICATOR.

References:

(1) *Single-chip microcontrollers. Elektor India*, October 1987.

(2) *MSX extensions — 5: EPROM programmer (2). Elektor India*, May 1987.

Procomm® is a registered trademark of Datastorm Technologies, Inc. • PO Box 1471 • Columbia MO 62205 • USA. BBS: (314) 449-9401 (24 h.).

MCS-51® is a registered trademark of Intel Corporation.

handshaking procedure is shown in Fig. 7.
Table 4 is a hex dump of a simple filehandler for IBM PCs and compatibles. The program is called SENDBAS.COM, and was written by H Peters. It loads (ASCII) BASIC files from disk, and sends these to the BASIC computer via serial port COM1:, in accordance with the previously mentioned prompt-based handshaking arrangement.
The program is loaded and written onto disk with the aid of DEBUG, which can be found on the DOS disk (use version 3.1 or later). Format a new disk, and copy DEBUG.COM onto it. Select the relevant drive, e.g. B:. Follow this instruction if you are unfamiliar with the operation of DEBUG:

DEBUG<CR>

Fill a 256 byte block with nulls:

F 0100 01FF 00<CR>

Name the program:

NSENDBAS.COM<CR>

Ready for entering the 256 bytes:

E 100<CR>

Enter the bytes (*not the addresses*) in Table 4, starting with B4. The first 2-byte address on each line is irrelevant in this case. Use the hyphen for corrections, and the space bar to proceed to the next byte. Type <CR> when the block is complete, and check the screen against the data in Table 4. If necessary, consult the chapter on DEBUG in your DOS manual.

Call up the block pointers:

RCX<CR>

and type

00FF<CR>

after the colon. Do the same with

RBX<CR>

and again

00FF<CR>

Write the .COM file to disk:

W<CR>

Leave DEBUG:

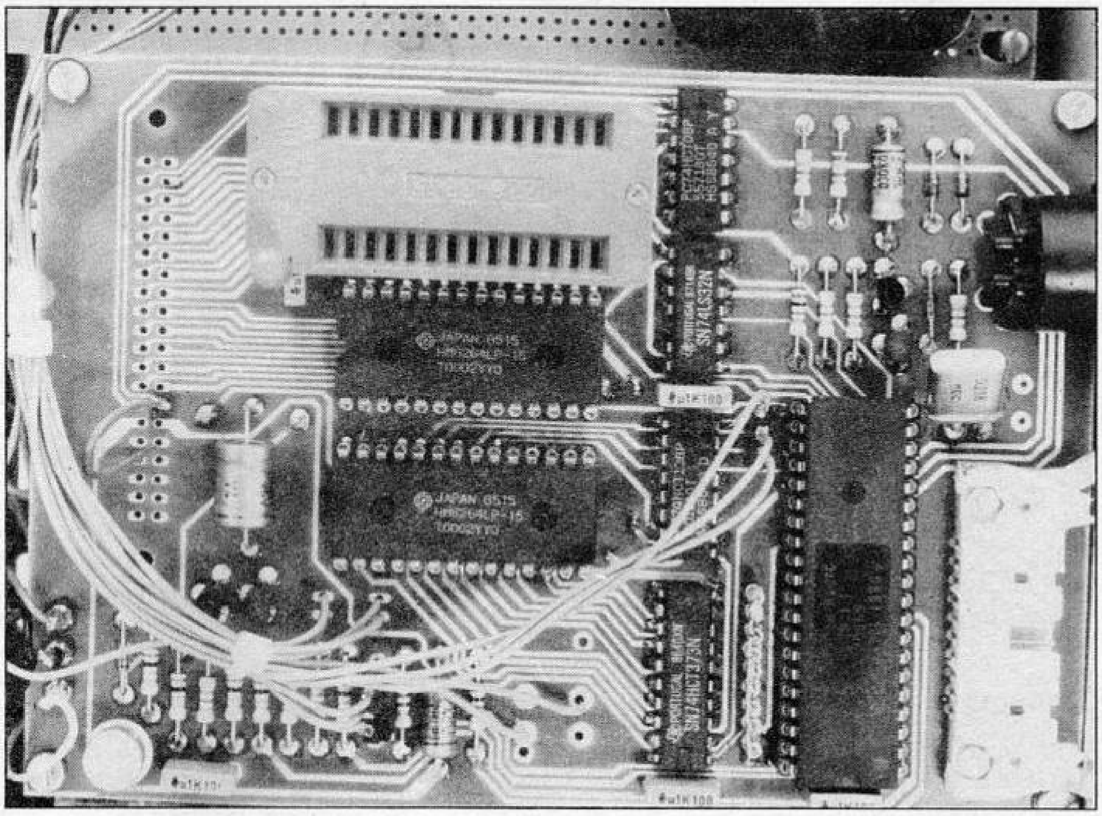
Q<CR>

The PC filehandler is now available on disk, and can be called with command SENDBAS. Test the program: the screen is cleared, and the text **ENTER FILENAME:** is displayed. Type <CR> to return to the DOS command prompt.

Table 3.

COMMANDS	STATEMENTS		OPERATORS	
RUN	BAUD	ONTIME	ADD (+)	DBY( )
CONT	CALL	PRINT	DIVIDE (/)	XBY( )
LIST	CLEAR	PRINT#	EXPONENTIATION (**)	GET
LIST#	CLEAR(S&I)	PRINT@ (V1.1)	MULTIPLY (*)	IE
LIST@ (V1.1)	CLOCK(1&0)	PH0.	SUBTRACT (−)	IP
NEW	DATA	PH0.#	LOGICAL AND (.AND.)	PORT1
NULL	READ	PH0.@ (V1.1)	LOGICAL OR (.OR.)	PCON
RAM	RESTORE	PH1.	LOGICAL X-OR (.XOR.)	RCAP2
ROM	DIM	PH1.#	LOGICAL NOT (.OR.)	T2CON
XFER	DO-WHILE	PH1.@ (V1.1)	ABS( )	TCON
PROG	DO-UNTIL	PGM (V1.1)	INT( )	TMOD
PROG1	END	PUSH	SGN( )	TIME
PROG2	FOR-TO-STEP	POP	SQR( )	TIMER0
PROG3 (V1.1)	NEXT	PWM	RND	TIMER1
PROG4 (V1.1)	GOSUB	REM	LOG( )	TIMER2
PROG5 (V1.1)	RETURN	RETI	EXP( )	XTAL
PROG6 (V1.1)	GOTO	STOP	SIN( )	MTOP
FPROG	ON-GOTO	STRING	COS( )	LEN
FPROG1	ON-GOSUB	UI(1&0)	TAN( )	FREE
FPROG2	IF-THEN-ELSE	U0(1&0)	ATN( )	PI
FPROG3 (V1.1)	INPUT	LD@ (V1.1)	=, >, >=, <, <=, <>	
FPROG4 (V1.1)	LET	ST@ (V1.1)	ASC( )	
FPROG5 (V1.1)	ONERR	IDLE (V1.1)	CHR( )	
FPROG6 (V1.1)	ONEX1	RROM (V1.1)	CBY( )	

Table 3 Overview of the instructions supported by the 8052AH-BASIC.



A close look at the component side of the populated board (prototype version)

Table 4.

```
A>DEBUG SENDBAS.COM
-D 0100 01FF
1E48:0100  B4 00 B0 02 CD 10 8C C8-05 10 00 8E D8 BB ED 00   ................
1E48:0110  53 E8 3B 00 7A 26 E9 94-00 5B 8A 07 43 53 3C 1A   S.;.z&...[..CS<.
1E48:0120  74 F4 3C 0A 74 16 B4 01-BA 00 00 CD 14 B4 02 BA   t.<.t...........
1E48:0130  00 00 CD 14 B4 02 8A D0-CD 21 EB DD B4 02 BA 00   .........!......
1E48:0140  00 CD 14 8A D0 B4 02 CD-21 3C 3E 75 EF EB CA B4   .........!<>u...
1E48:0150  09 BA B0 00 CD 21 B4 0A-BA CB 00 CD 21 BB CC 00   .....!......!...
1E48:0160  8A 07 3C 00 75 03 EB 45-90 BB CD 00 B9 1E 00 8A   ..<.u..E........
1E48:0170  07 3C 0D 74 06 43 E2 F7-EB 05 90 B0 00 88 07 B4   .<.t.C..........
1E48:0180  3D BA CD 00 B0 00 CD 21-8B D8 B4 3F B9 FF FF BA   =......!...?....
1E48:0190  ED 00 CD 21 8B D8 B4 3E-CD 21 B0 20 B4 01 BA 00   ...!...>.!. ....
1E48:01A0  00 CD 14 B0 0D B4 01 BA-00 00 CD 14 C3 5B CD 20   .............[.
1E48:01B0  0D 0A 0A 0A 0A 20 20 20-20 20 45 4E 54 45 52 20   .....     ENTER
1E48:01C0  46 49 4C 45 4E 41 4D 45-3A 20 24 1E 00 00 00 00   FILENAME: $.....
1E48:01D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
1E48:01E0  00 00 00 00 00 00 00 00-00 00 00 00 00 1A 26 00   ..............&.
1E48:01F0  74 09 E8 40 E1 E8 1B F1-E8 BC E1 A1 D6 26 A3 04   t..@..........&..
-
```

87192-T4

Table 4 Hexdump of SENDBAS.COM, the filehandler for PCs and compatibles.

5

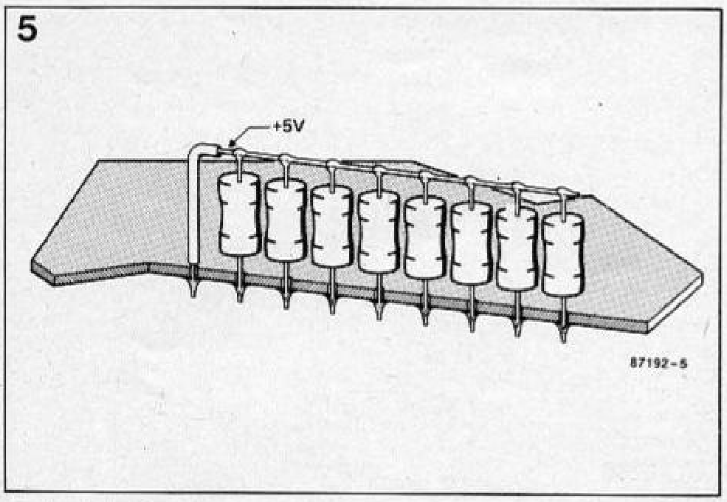


Fig. 5 Showing the use of 8 ordinary resistors instead of a SIL network.

grammed with BASIC modules, and only when the computer is turned off.
The power supply for the BASIC computer can be a simple type with regulated outputs for 5 V (500 mA), and the programming voltage(s).
Initially, the CPU and the memory chips are not fitted while the completed board is fed with Vcc and Vpp. Consult the circuit diagram and carefully check the presence of the supply voltage at all the relevant points. Make sure that there is no short circuit around pin 28 of IC_5, since the programming voltage is carried nearby. Switch off the power, carefully fit the CPU and the RAM(s) with the correct orientation, and switch the power on again.

Communication: the terminal

The serial data format for the BASIC computer is:

8 data bits, no parity, 1 stop bit.

Most terminals, consoles, or terminal emulation programs for computers can support this format.
The 3-wire connection between the BASIC computer and the terminal is shown in Fig. 6. At the terminal side, it may be necessary to hard wire a number of RS232 hanshaking lines—consult the relevant documentation. A solution that works in most cases is to connect the following pins in the 25-way RS232 connector:
4—5—8 and
6—20 (sometimes 6—20—22).
Where — denotes the connection.

The BASIC computer has an internal baud rate timing routine. Press RESET, wait a second or so, and press the space bar on the terminal. The message

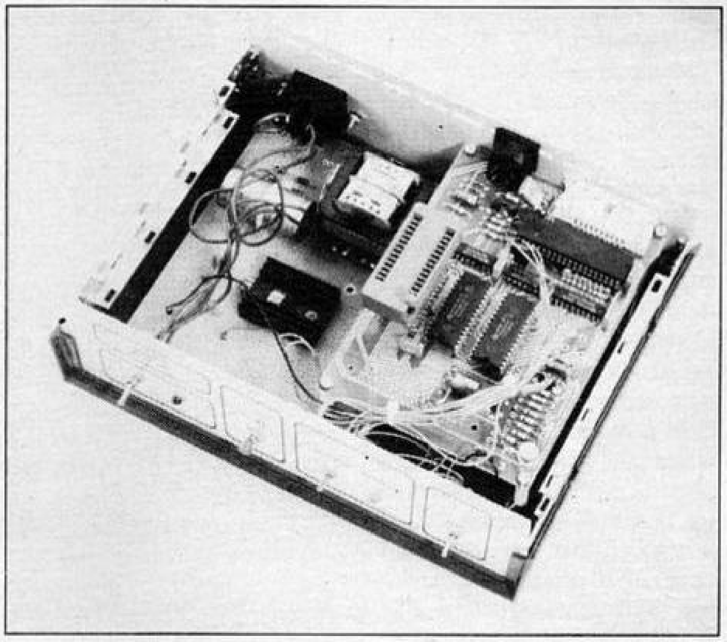
*MCS-51(tm) BASIC V1.1
READY
>

is displayed on the terminal screen, and the BASIC computer is ready to accept commands.
After RESET is pressed, the CPU initializes its internal RAM, and a number of pointers and registers. It then tests, initializes, and determines the size of the external memory area (IC_4 and IC_5). Next, the memory size is stored with the aid of operator MTOP (*memory top*), operator XTAL is defined (default: 11059200), and, finally, the CPU reads the data at address 8000 to check for a valid baud rate definition, programmed in EPROM IC_6. When a baud rate byte is found, it is stored in register T2CON. The computer then skips its automatic baud rate timing routine and operates at the preprogrammed serial speed, obviating the need for the terminal operator to press the space bar after actuating RESET on the BASIC computer.
The maximum baud rate is 38.4 Kbit/s, and timing characters other than 20_H (space) are not accepted.

To verify the correct operation of the system, type

PRINT XTAL,TMOD,TCON, T2CON <CR>

to which the computer replies



Inside view of a prototype of the BASIC computer.

6

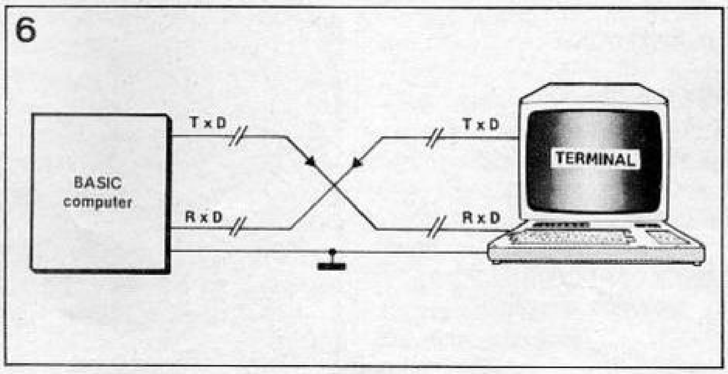


Fig. 6 The 3-wire connection between the BASIC computer and the terminal.

11059200 16 244 52

The system prompt > is displayed to indicate that the computer is ready to accept commands, which are not executed until <CR> is received. Actually, the 8052AH-BASIC starts tokenizing and storing the BASIC commands after receiving a carriage return ($0D_H$). Depending on the length of the line, and the complexity of the command(s), this takes some time, and *new characters must not be sent* until the CPU responds with the prompt, indicating completion of the storage process.

The BASIC computer is probably best programmed and controlled with the aid of a personal micro sporting an RS232 port. As to software, a terminal emulation or communication program in conjunction with a wordprocessor enables efficient editing and downloading of BASIC files. A general flowchart of a serial I/O routine to support the above

7

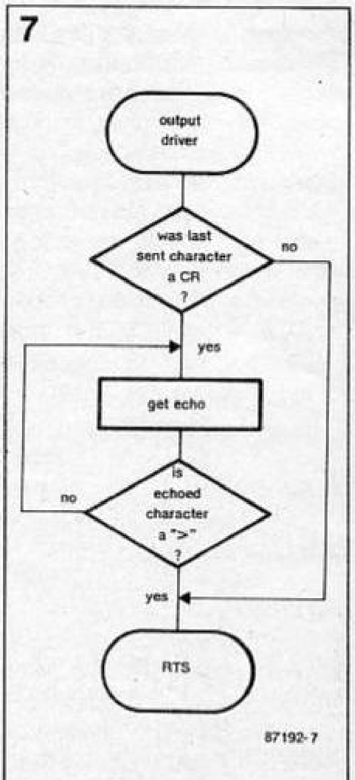


Fig. 7 The sending computer must wait for the > prompt from the BASIC computer before sending a new line of commands.