# ROM-COPY FOR 8052-BASIC COMPUTER

**In an earlier article on our popular 8052-BASIC computer we showed a way to replace the 8052AH-BASIC microcontroller with the much cheaper 8032 or 80C32 and an external EPROM. This month we take a second look at unloading the BASIC interpreter from the 8052AH-BASIC and transferring it to an EPROM. We also avail ourselves of the opportunity to get to grips with serial communication protocols for the BASIC computer.**

**E. Vermeulen**

Although many of our regular readers will be familiar with the BASIC computer published in Ref. 2, it does no harm to inform others that this computer is among the most popular construction projects of the past few years. The BASIC computer is an extremely versatile single-board microcontroller system based on Intel's 8052AH-BASIC processor. This chip from the MCS52 family of 8-bit processors features an on-board BASIC interpreter which can be accessed via a 3-wire serial link to an external RS232-compatible terminal. In addition to an on-board EPROM programmer, the system puts a number of easily controlled user ports and timers at the programmer's disposal, and is marked by versatility and simple construction. If you are not convinced of this, we suggest you look at the way the BASIC computer is used in the telephone exchange to be published in a forthcoming issue.

The current consumption of the BASIC computer may be reduced considerably by replacing the 8052AH-BASIC processor by the 80C32. Add an external EPROM that contains the machine code for the BASIC interpreter and you have a low-power BASIC computer suitable for battery operation.

```
1000 REM          ************************************************
1010 REM          **       8052 - AH   BASIC V 1.1 (INTEL)      **
1020 REM          **                                            **
1030 REM          **                R O M C O P Y               **
1040 REM          **                                by E Vermeulen  **
1050 REM          **                                            **
1060 REM          **   THIS PROGRAM TRANSFERS THE BASIC         **
1070 REM          **   INTERPRETER IN THE ROM BURNED IN THE     **
1080 REM          **   8052AH-BASIC PROCESSOR TO A 2764 EPROM   **
1090 REM          **   BY USING THE ON-BOARD EPROM PROGRAMMER.  **
1100 REM          **   INTERPRETER  8052 IC:          0000H-1FFFH **
1110 REM          **   RAM ADDRESS (IC5) ON CARD:     2000H-3FFFH **
1120 REM          **   EPROM ADDRESS ON CARD (2764):  8000H-9FFFH **
1130 REM          **-------------------------------------------**
1140 REM          **       8052-AH BASIC   Copyright by INTEL   **
1150 REM          ** For futher information refer to Intel's    **
1160 REM          **         MCS  BASIC - 52 User's Manual      **
1170 REM          **         Order Number : 270010-003          **
1180 REM          **         Intel Corporation (UK) Ltd.        **
1190 REM          **         Pipers Way, Swindon SN3 1RJ        **
1200 REM          **         phone (0793) 696000                **
1210 REM          **                                            **
1220 REM          ************************************************
1230 REM
1240 PRINT"** PART 1: MOVE INTERPRETER TO RAM AT 2000H-3FFFH"
1250 PRINT:PRINT
1260 MTOP=1FFFH:                        REM : RESERVE RAM FROM 2000H
1270 DIM A(15):                         REM : DIMENSION ARRAY 1x16
1280 FOR X=0000H TO 1FFFH STEP 16:      REM : SET ROM ADDRESSING
1290    FOR Y=0 TO 15:                  REM : IN BLOCKS OF 16 BYTES
1300       A(Y)=CBY(X+Y):               REM : READ 1 BYTE ROM, PLACE IN ARRAY
1310       Z=X+Y+2000H:                 REM : COMPUTE CORRESPONDING RAM ADDRESS
1320       XBY(Z)=A(Y):                 REM : WRITE BYTE IN RAM
1330       B=XBY(Z):                    REM : READ IT BACK FROM RAM
1340                                    REM : AND COMPARE WITH BYTE IN ROM
1350                                    REM : ERROR? => REPORT AND STOP
1360       IF A(Y)<>B THEN ?"PROGRAM ERROR ,ROM= ",A,"RAM=",B:END
1370    NEXT Y:                         REM : 16-BYTE LOOP
1380                                    REM : SHOW 16-BYTE LINE ON TERMINAL
1390    PH1. X,"     ",:PH0. A(0),A(1),A(2),A(3),A(4),A(5),A(6),A(7),
1400    PH0. A(8),A(9),A(10),A(11),A(12),A(13),A(14),A(15)
1410 NEXT X:                            REM : LOOP FOR ENTIRE ROM (0000H-1FFFH)
1430 PRINT"** PART2: SET INTERNAL MEMORY FOR PROGRAMMING PROCEDURE"
1450                                    REM : INTERPRETER (size = 8 Kbyte)
1460                                    REM : NOW IN RAM FROM 2000H TOT 3FFFH
1470                                    REM : THE EPROM (also 8 Kbytes)
1480                                    REM : OCCUPIES 8000H TO 9FFFH
1490 DBY(18H)=0FFH:                     REM : TARGET ADDRESS LOW BYTE-1
1500 DBY(19H)=00H:                      REM : SOURCE ADDRESS LOW BYTE
1510 DBY(1AH)=7FH:                      REM : TARGET ADDRESS HIGH BYTE
1520 DBY(1BH)=20H:                      REM : SOURCE ADDRESS HIGH BYTE
1530 DBY(1EH)=00H:                      REM : NUMBER OF BYTES LOW ADDRESS
1540 DBY(1FH)=20H:                      REM : NUMBER OF BYTES HIGH ADDRESS
1550 W=0.05:                            REM : W=WIDTH OF PROGRAMMING PULSE IN SEC
1560 R=65536-W*XTAL/12:                 REM : R=RELOAD VALUE
1570 DBY(40H)=R/256:                    REM : SET VALUE OF R (LOW BYTE)
1580 DBY(41H)=R .AND. 0FFH:             REM : SET VALUE OF R (HIGH BYTE)
1590 DBY(38)=DBY(38) .AND. 0F7H:        REM : RESET BIT 38.3 FOR 50mSEC PULSES
1600 PRINT:PRINT:PRINT SPC(25),"** SETTINGS OKAY **":PRINT:PRINT
1610
1620 PRINT"** PART 3: PROGRAMMING THE EPROM"
1630
1640 FOR R=1 TO 5:PRINT:NEXT R
1650 PRINT"** SWITCH ON THE PROGRAMMING VOLTAGE":PRINT
1660 PRINT"** BE SURE TO APPLY THE CORRECT VOLTAGE (12.5V OR 21 VOLT!":PRINT
1670 FOR R=1 TO 3:PRINT:NEXT R
1680 PRINT"PRESS ENTER TO START PROGRAMMING"
1690 PRINT"OR PRESS ESCAPE TO STOP":PRINT:PRINT
1700 PRINT"PLEASE TYPE <ENTER> OF <ESC>",
1710 K=GET:IF K=0 THEN 1710
1720 IF K=1BH  THEN PRINT:END:      REM ** 1BH is ASCII-code for Esc
1730 IF K<>0DH THEN 1710:           REM ** 0DH is ASCII-code for Enter
1740 PRINT:PRINT:PRINT"BUSY PROGRAMMING EPROM"
1750 PRINT:PRINT"THIS WILL TAKE ABOUT 7 MINUTES":PRINT
1760
1770 PGM                            :REM ** PROGRAMMING INSTRUCTION **
1780 PRINT
1790 PRINT:PRINT"** PART 4: CHECKING INTERNAL POINTERS FOR ERRORS"
1800 PRINT
1810 H=DBY(1AH):L=DBY(18H):HL=H*256+L
1820 IF (DBY(30).OR.DBY(31))<>0 THEN 1830 ELSE 1840
1830 PRINT"INCORRECT PROGRAMMING OF EPROM AT ADDRESS",:PH1.HL:END
1840 PRINT"PROGRAMMING FINISHED   ** NO ERRORS **":PRINT
1850 PRINT
1860 PRINT"** PART 5: DIRECT COMPARISON BETWEEN ROM AND EPROM":PRINT
1870 PRINT
1880 FOR X=0000H TO 1FFFH           :REM ** ADDRESS IN ROM
1890    Y=X+8000H                   :REM ** ADDRESS IN EPROM
1900    A=CBY(X):B=XBY(Y)           :REM ** READ OUT RAM AND EPROM
1910    PRINT"ROM",:PH1.X,:PRINT" =>",:PH0.A," =",
1920    PH0.B,:PRINT" <=",:PH1.Y,:PRINT" EPROM",CR,
1930    IF A<>B THEN PRINT"EPROM ERROR":END
1940 NEXT X:PRINT:PRINT
1950 PRINT"EPROM CORRECTLY PROGRAMMED":PRINT:PRINT
1960 PRINT"PROGRAM FINISHED, BYE !!!"
1970 END
```

This program is available on dis (5¼-inch 360K IBM format) a order number **1441** (see th Readers Services page).

Fig. 1. Type this listing of ROMCOPY.BAS into your favourite word processor.

## Do you copy?

All that is needed to transfer the BASIC interpreter from the 8052AH-BASIC to a 27C64 EPROM is the program listed in Fig. 1. Simply get your original BASIC computer on line, and type in the listing on your terminal or PC. Check what you have on your screen against the printed listing. If this is all right, save the program. Plug an empty 27C64 EPROM in the ZIF socket on the BASIC computer board, apply the correct programming voltage, and RUN the program. The 8052AH-BASIC does the rest, i.e., it transfers its own BASIC interpreter to the EPROM. Some seven minutes later, the EPROM is ready for use in a 80C32-based version of the BASIC computer.

## Procomm and the BASIC computer

Some problems may crop up when the BASIC computer is connected to an IBM PC or compatible running a communication program like Procomm. These problems arise mainly from the absence of any form of hardware handshaking on part of the BASIC computer. Also, while downloading a listing generated on the PC, the BASIC computer needs time to convert the ASCII data into token codes. This operation, carried out by the BASIC interpreter in the 8052AH-BASIC, starts after reception of a carriage return (CR) character from the terminal. During the token conversion, the 8052AH-BASIC does not monitor the serial port. The result is that characters may be lost in the downloading process when the terminal does not wait for the BASIC computer to complete its token conversion operation after every complete line of BASIC instructions. One way of solving the handshaking problems is to reduce the bit rate on the serial link to a value where the time needed for the token conversion is simply not noticed.

## Waiting for 62

A fairly reliable method of ensuring software handshaking is to have the PC wait for the prompt (>) transmitted by the BASIC computer. This can be achieved by programming ASCII value 62 for the pace character in Procomm's ASCII transfer setup menu. First, however, go to the line settings menu (ALT-S) and select option 12 to set the serial data speed and format to 19,200 bits/s, no parity, 8 databits and 1 stopbit. If necessary change the setting as required for the COM port you intend to use. Save the settings.

Next, call up the setup screen by typing ALT-S. Select item 2, the terminal setup. The parameters necessary for the BASIC computer are shown in the screendump in Fig. 2. Return to the setup screen, and select item 6, the ASCII transfer configuration. The settings used by the author are shown in Fig. 3. The pace character (option 3) is set to 62 to force the PC to wait for the BASIC computer after sending a line of BASIC. The character pacing may be left at 0 in most cases. The line pacing is best set to 5 as shown, although this is not critical.

When the pace character is set to 0, Procomm ignores any character returned by the BASIC computer. This setting is still useful, however, when the software handshaking is not used, and the serial link runs at a relatively low baud rate. When software handshaking is not used, some experimenting with the character pacing and line pacing parameters may be required for best results. A lot depends on the length of the cable between the PC and the BASIC computer, and also whether or not this cable is screened. Fortunately, the wait times that may be programmed in Procomm will allow you to find a compromise between fast data transfer and as few as possible errors.

## Uploading the ROMcopy program

Assuming that you have typed the listing in Fig. 1 into your favourite word processor (set to ASCII output), the procedure to send the program to the BASIC computer is as follows. First, save the file as ROMCOPY.BAS on floppy disk or hard disk. Next, run Procomm, and get the BASIC computer on line by pressing the space bar. Type PgUp to enter the upload menu. Select option 7, ASCII PROTOCOL. Type the filename (if necessary, preceded by the drive station and/or the path, e.g. A:\ROMCOPY.BAS) followed by a return. The upload process can be followed on the screen. On completion, type LIST to check the loaded file against the listing. RUN the program and you have your BASIC interpreter available in EPROM after about seven minutes. ■

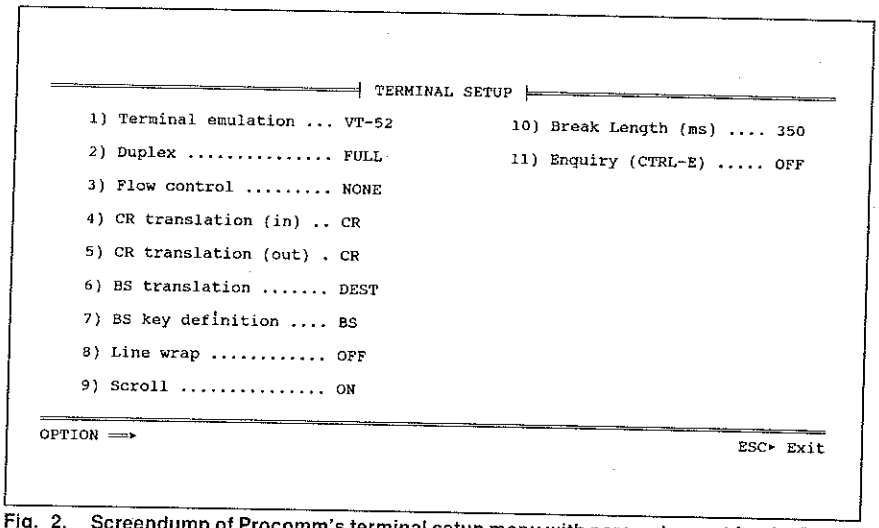


**Fig. 2. Screendump of Procomm's terminal setup menu with parameters set for the BASIC computer.**

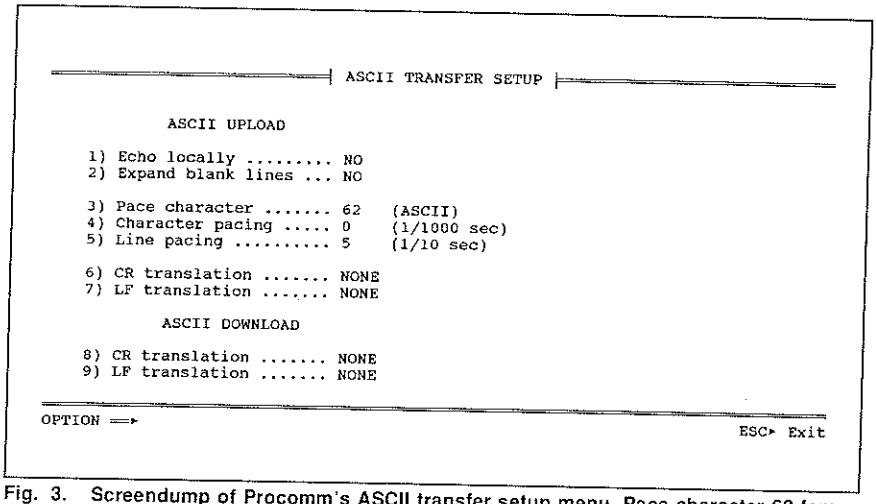


**Fig. 3. Screendump of Procomm's ASCII transfer setup menu. Pace character 62 forces the PC to wait for the prompt transmitted by the BASIC computer.**

**References:**

1. CMOS replacement for 8052AH-BASIC. *Elektor Electronics* January 1990
2. BASIC computer. *Elektor Electronics* November 1987.