



Application Note

Product: 8052 Microcontrollers

Simulating PROGx command for BASIC program storage in EEPROM

Date: 9/27/91

Not all 8052 microcontrollers support BASIC's PROG and PROGx commands, because they lack the hardware necessary to perform the programming functions for EPROMs. Now, you can use EEPROMs on all of Micromint's 8052 microcontrollers by simply appending the following lines to the end of your program.

EEPROMs have the advantage of NOT requiring special programming voltages (as do EPROMs) and an EEPROM does NOT require erasure under special UV lamps. The erasure is done electrically, prior to writing the data. The only disadvantage is that an EEPROM's byte write mode is considerably slower than intelligent EPROM programming algorithms. Unlike an EPROM which can be written, read, and verified immediately, the EEPROM's programming algorithm reads back the last data written with the high-bit inverted until the self-timed write is completed. This requires rereading the data until the device is timed out, pausing for 10ms prior to the read. It is this delay that causes the normal EPROM programming commands to fail with a "programming error" message.

EEPROMs are available from many manufacturers.

8K EEPROM – 2864/28C64 or 32K EEPROM – 28256/28C256

Samsung	TI	INTEL	SEEQ
ATMEL	MICROCHIP	EXEL	FUJTSU
NEC	XICOR		

DO NOT apply any programming voltage. Applying a programming voltage will permanently damage the EEPROM device.

If you are using the BCC52 or BCC52C, you can use 8K EEPROMs.

If you are using the BCC52CX, you can use 32K EEPROMs.

If you are using the RTC52, you can use 8K or 32K EEPROMs.

Make sure the jumpers associated with the socket in which you are using the EEPROM are configured for RAM. For autostarting, the EEPROM must be located at 8000H. If you have contiguous RAM in the system up through 7FFFH, your EEPROM will be trampled upon unless you set mtop to below 8000H after initially powering up the board. Also, make sure you use the proper PROGx command to save MTOP if your RAM is contiguous, which prevents clobbering the EEPROM on autostart.

The BCC52 and BCC52C can be write protected by removing jumper JP8. An onboard pullup will keep the RD/*WR line to float. A pullup resistor must be added between pins 27 and 28 on the memory socket holding the EEPROM to ensure write protection. Write protecting an EEPROM is NOT essential, only an added security measure.

Storage of a BASIC program in the ROM (RAM) area is as follows:

8000H – ASC value of 'X' in the PROGx command
8001H – MSB of RCAP2 reload value (baud rate)
8002H – LSB of RCAP2 reload value (baud rate)
8003H – MSB of MTOP (present value)
8004H – LSB of MTOP (present value)

8010H – '55H' indicating a BASIC program follows
8011H on up – BASIC program (exact copy of code starting at 200H in low RAM)

Append this to your debugged program in lower RAM and executed it with a 'GOTO 10000' command.

```

9999  END
10000 PRINT "HIT 'p' TO MOVE YOUR BASIC PROGRAM INTO EEPROM"
10010 PRINT "(HIT 'Q' TO QUIT WITHOUT EXECUTING PROG COMMAND)"
10020 G=GET
10030 G=GET
10040 IF (G<>50H.AND.G<>70H) THEN GOTO 10250
10050 V=55H : D=8010H : GOSUB 10210
10060 PRINT "LEN=",LEN " PRINT "TIME=",LEN/30" SECONDS"
10070 FOR X=200H TO (200H+LEN)
10080 V=XBY(X) : D=X+7E11H : GOSUB 10210
10090 NEXT X : PRINT
10100 PRINT "HIT '1-6' TO DO A PROG# TO 8000H"
10110 PRINT "(HIT 'Q' TO QUIT WITHOUT EXECUTING PROGX COMMAND)"
10120 V=GET
10130 V=GET : IF V=0 THEN 10130
10140 IF (V<31H.OR.V>36H) THEN GOTO 10250
10150 D=8000H : GOSUB 10210
10160 V=INT(RCAP2/256) : D=8001H : GOSUB 10210
10170 V=RCAP2-(INT(RCAP2/256)*256) : D=8002H : GOSUB 10210
10180 IF V<32H THEN GOTO 10250
10190 V=INT(MTOP/256) : D=8003h : GOSUB 10210
10200 V-MTOP-(INT(MTOP/256)*256) : D=8004H : GOSUB 10210
      : GOTO 10250
10210 XBY (D)=V : Z=0 : IF V=0DH THEN PRINT ".",
10220 IF (XBY(D)<>V.AND.Z<5) THEN Z=Z+1 : GOTO 10220
10230 IF (Z=5) THEN PRINT "ERROR @", D
10240 RETURN
10250 END

```

Here's whats happening.

Line 9999 – stops execution of your program

Line 10000 – This is the beginning of **PROG** simulation

Line 10050 – writes the '55H' header to 8010H

Line 10070-10090 – copies the BASIC program (including itself from the 200H low RAM area to 80011H

Line 10100 – This is the beginning of **PROGx** simulation

Line 10150 – writes the ASC value of x from the PROGx command to 8000H

Line 10160-10170 – writes the MSB and LSB of RCAP2 baud rate reload value to 8001H and 8002H

Line 10190-10200 – writes the MSB and LSB of MTOP to 8003H and 8004H

Line 10210 – This is the beginning of EEPROM write routine

Line 10220 – attempts to verify data 5 times

Line 10230 – if data not verified print an ERROR message