```
'      SERCURV.BAS      EXAMPLE PROGRAM FOR SERIAL INTERFACE
'      This program works with QuickBasic 4.0/4.5 or QBasic on an IBM PC or compatible
'      with a serial interface. It uses the COM1 communication port and 300 BAUD.
'
'      To use, enter an instrument command or query at the prompt. The command goes to
'      the instrument and any query response displays. "EXIT" exits the program.
'      NOTE: The INPUT instruction in this example accepts no commas as part of an input
'      string. If a comma appears in an instrument command, replace it with a space
'      when entering data from the keyboard.
'
       CLS                                      'Clear screen
       PRINT " SERIAL COMMUNICATION PROGRAM"
       PRINT
       TIMEOUT = 2000                           'Read timeout (may need more)
       BAUD$ = "300"                            'BAUD rate 300 (or 1200)
       TERM$ = CHR$(13) + CHR$(10)              'Terminators are <CR><LF>

       OPEN "COM1:" + BAUD$ + ",o,7,1,RS" FOR RANDOM AS #1 LEN = 256

LOOP1: INPUT "ENTER COMMAND (or EXIT):"; CMD$   'Get command from keyboard
       CMD$ = UCASE$(CMD$)                      'Change input to upper case
          IF CMD$ = "EXIT" THEN CLOSE #1: END   'Get out on Exit
          IF CMD$ = "CURVE" THEN GOTO LOAD      'Load a curve from disk file
       CMD$ = CMD$ + TERM$
       PRINT #1, CMD$;                          'Send command to instrument

       IF INSTR(CMD$, "?") <> 0 THEN            'Test for query
          RS$ = ""                              'If query, read response
          N = 0                                 'Clr return string and count

          WHILE (N < TIMEOUT) AND (INSTR(RS$, TERM$) = 0) 'Wait for response
             IN$ = INPUT$(LOC(1), #1)           'Get one character at a time
             IF IN$ = "" THEN N = N + 1 ELSE N = 0 'Add 1 to timeout if no chr
             RS$ = RS$ + IN$                    'Add next chr to string
          WEND                                  'Get chrs until terminators

          IF RS$ <> "" THEN                     'See if return string is empty
             RS$ = MID$(RS$, 1, (INSTR(RS$, TERM$) - 1)) 'Strip off terminators
             PRINT "RESPONSE:"; RS$             'Print response to query
          ELSE
             PRINT "NO RESPONSE"                'No response to query
          END IF
       END IF                                   'Get next command
       GOTO LOOP1
```

```
'*************************************************************************
'*                          CURVE LOADING PROGRAM                        *
'*  This routine will load a curve. Get here by entering "CURVE" above.  *
'*  NOTE: SPACING OF THE DATA STRING IS VERY CRITICAL.                   *
'*        For this example the string data must be on a single line      *
'*        of an ASCII file using the same format as the attached sample.*
'*************************************************************************
LOAD:  CURVE$ = SPACE$(2000)                    'Preset variable lengths
       DELAY = 2000                             'Delay timer
       PRINT
       PRINT "DOWN LOAD A CURVE"
       PRINT
       INPUT "ENTER DRIVE AND FILE NAME: "; FILE$     'Get file name from KB

       OPEN FILE$ FOR INPUT AS #2               'Open ASCII disk file
       LINE INPUT #2, CURVE$                    'Read disk file into string
       CLOSE #2                                 'Close disk file
       PRINT       'Get curve number from KB
       INPUT "ENTER DESTINATION CURVE NUMBER (11 to 31): "; CURNUM$

          IF MID$(CURVE$, 1, 1) = "X" THEN      'Test for DRC curve format
             CURVE$ = MID$(CURVE$, 5)           'Strip off DRC file header
             CURVE$ = "CURV" + CURNUM$ + CURVE$ 'Add 300 series file header
          ELSE
             CURVE$ = "CURV" + CURNUM$ + "," + CURVE$  'Add file header
          END IF

       PRINT
       PRINT "COMMANDS SENT TO 320"             'Screen prints to show what is sent
       PRINT

       CMD$ = LEFT$(CURVE$, 53)                 'Pick out header and first two points
       PRINT CMD$                               'Send first command string to screen
       CMD$ = CMD$ + TERM$                      'Add terminators
       PRINT #1, CMD$                           'Send curve create cmd to 321/320/330
          FOR Z = 1 TO DELAY: NEXT Z            'Delay

       CHRCOUNT = 54                            'Character count for next curve point
       POINTCOUNT = 3                           'Curve data point counter
```