Appendix E

Universal Serial Test Program – Visual Basic Version

Appendix D explained the inner workings of the QuickBASIC version of the Universal Serial Test Program (USTPQB.BAS). In this Appendix, I'll do the same for the Visual Basic version (USTPVB.VBP). Because of the high level of similarity between most of the programming statements, I'll focus on the differences. For a more complete description regarding the statements that remain the same, please consult Appendix D.

PROGRAM ORGANIZATION

Let's begin by reviewing the organizational structure of the USTPVB project as illustrated in Fig. E-1.

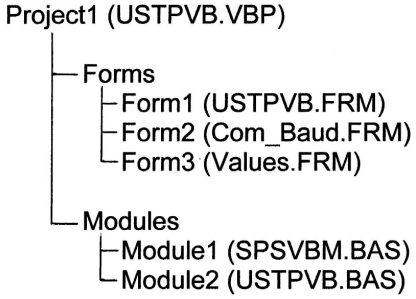

Fig. E-1. Tree-structure for USTPVB.VBP within Visual Basic

There are 3 forms and 2 modules described as follows:

- **Form 1**, with the assigned file name USTPVB.FRM, contains the main body of our program. Thus, much of the content of USTPVB.FRM is taken directly from the QuickBASIC version USTPQB.BAS defined in Appendix D. From the user interaction standpoint, this form also determines if an SMINI is being tested and if not, are the cards being used 24- or 32-bit.

- **Form 2**, with the assigned file name Com_Baud.FRM, is used to define the graphics used for entering the PC's communication port number and the desired baud rate. We created this form back in the Chapter 16 subsection ***Adding a Com Port and Baud Rate Selection Dialog Box with Option Buttons.***

- **Form 3**, with the assigned file name Values.FRM, is used to define the graphics used for reviewing and updating the values for the program default parameters MAXTRIES, USIC Delay (DL), SLOWDIS and INFILTER. We created this form back in the Chapter 16 subsection ***Adding Text Boxes for Display and Data Entry.***

- **Module 1**, with the assigned file name SPSVBM.BAS, contains the 5 Serial Protocol Subroutines (INIT, INPUTS, OUTPUTS, RXBYTE and TXPACK) along with the necessary Public statements to make the subroutines operational.

- **Module 2**, with the assigned file name USTPVB.BAS, contains a listing of the application specific Public variables to be shared between all Forms and all Modules.

When using the Visual Basic Editor, the VB project's tree-structure is displayed in the upper right corner of the screen. Simply click on the form or module that you wish to examine or edit. (Note: If the tree-structure isn't displayed, you can bring it up by clicking on View and then selecting Project Explorer). Because each form can contain graphics as well as code, the Visual Basic Editor provides two buttons directly above the tree-structure. Activate the View Code button and the code for the selected form is displayed. Activate the View Object button and the graphics for the selected form are displayed. Both can be edited using standard editing techniques. The View Object button becomes inactive when viewing a Module because modules can only contain code.

PROGRAM HIGHLIGHTS (USTPVB)

All the files utilized by the USTPVB.VBP project are included on the software disk supplied with this manual. This includes the directly entered coding statements as well as the VB self-generated coding to create the project file and the graphics associated with each of the 3 form files. In the remainder of this appendix we'll examine some of the key programming statements associated with Form1 and Module2. Form2 and Form3 descriptions, as well as their corresponding source code, are provided in Chapter 16.

You can think of Form1, file name USTPVB.FRM, as the "mainline program" because it invokes the other two forms as well as the Serial Protocol Subroutines. The "user entered" code for creating the USTPVB.FRM file is provided at the end of this appendix. You may refer to it periodically as we discuss the program highlights. Then, once you have the user-generated code understood, you might want to take a look at the corresponding disk file, which also includes the VB self-generated code which among other things includes the coding required to support the graphics generation used with Form1.

Comparing the listing at the end of this appendix with the corresponding QuickBASIC listing found at the end of Appendix D, it becomes obvious that the vast majority of the programming statements are identical. In the following discussion, I'll highlight the differences. See Appendix D for a more complete description of the coding that remains unchanged.

In highlighting the differences we'll be applying the 14-step procedure, as derived back in Chapter 16, for converting a QuickBASIC program to Visual Basic. Firstly, the **`Declare Sub`** statements are deleted because the USTPVB.VBP project file defines which files need to be included. Likewise, the **`DIM SHARED`** and **`COMMON SHARED`** statements are deleted because all of this is taken care of for you within the VB version of the Serial Protocol Subroutines.

With the Form1 Declaration section complete, we need to add the **Private Sub Form_Load()** statement prior to our first executable statement which is the **`PNU$(1) = "A"`**. Following the definition of the port display array parameters PNU$( ) and PNM$( ), which stay the same, the VB version initializes the maximum buffer size for MSComm at 50 via the **`MAXBUF = 50`** statement per the discussion in Chapter 15. Then the **`Form1.Show`** command is invoked to activate the Form1 display for printouts on Form1 followed by a **`Cls`** statement to clear the monitor screen.

Next, a VB MsgBox function is used to query the user whether or not the testing is for an SMINI. The program does this by invoking the standard "vbYesNo" button nomenclature. This automatically creates two "buttons," one labeled "Yes" and one labeled "No." Clicking on the Yes button will result in the MsgBox function returning with x = 1 and clicking on the No button will return x = 2. Thus, in this case, all we need to do is check for x = 1 (i.e. the Yes button was pressed), then execute a branch to line label

INSTM to begin initialization processing for the SMINI. Otherwise, the program simply falls through the comparison to line label INSTU, to begin initialization processing for the SUSIC.

To illustrate different programming techniques, I did vary the specific coding used for making the comparison tests after using MsgBox. Rather than applying the `If x = 1 Then...` or `If x = 2 Then...` statements (as we've done previously in the Chapter 16 examples), VB allows the programmer to simply use the more direct comparison `If x = vbYes Then...` and/or `If x = vbNo Then...` statements, thus, circumventing having to remember that x = 1 corresponds to the Yes button and x = 2 the No button.

Not only does VB make data entry easy via button presses, it also circumvents the need to check for invalid entries because the only possible entries are Yes and No, which are both valid. Also, option buttons perform just like "radio station selection buttons," meaning that you can only activate one button. For example, if you press Yes, then change your mind and press No, the latter action cancels the Yes button press. For this reason, Option buttons are frequently referred to as "Radio Buttons."

At the line labels INSTU and INSTM, multiple Print lines are executed, as with QuickBASIC, to define the setup requirements for performing each test. Directly after the printouts, the program sets the Node Definition Parameter (NDP$). For the SMINI case this is easily accomplished with the single statement `NDP$ = "M"`. However, for the USIC/SUSIC case it's necessary to know if 24- or 32-bit I/O cards are being used. Thus, the VB MsgBox statement is employed with the query statement changed to read "Are you using 32-bit cards?" If the No button is pressed the program sets NDP$ = "N" (i.e. 24-bit cards are being used) and if the Yes button is pressed the program sets NDP$ = "X". In both cases the program then branches to read in the desired PC communications port and baud rate at label COMIN.

At this point the VB version is considerably different. The `Form2.Show` statement is invoked to activate Form2 with the file name COM_Baud.FRM. Using this form, the user simply clicks on one option button to select the Com port desired and another option button to select the baud rate. Once the OK button is pressed, the program coding within Form2 places the correct values in the variable locations for COMPORT and BAUD100, prior to closing Form2.

The next difference in the VB program is the use of another MsgBox, with the standard Yes and No buttons, to define whether or not a change in the default settings for MAXTRIES, USIC Delay (DL), INFILTER and SLOWDIS is desired. If the user presses the No button, the program branches to label TESTIN where the program waits for the user to define which testing is to be performed. If the user presses the Yes button, the program falls through the If-Then statement to execute the `Form3.Show` statement which displays Form3 with the file name VALUES.FRM. This form provides the user with a simple procedure to override the default parameter values for the variables MAXTRIES, USIC Delay (DL), INFILTER and SLOWDIS. Once the OK button is pressed, Form3 takes any user entered data and places it into the correct variable location, prior to closing Form3.

It should be pointed out that both the Form2 and Form3 Show statements include the numeral 1 after Show. This signifies to VB that the form being displayed is "Modal." This simply means that no input (keyboard or mouse click) can occur except to objects on the modal form. The program must hide, or unload, a modal form before input to another form can occur. This is precisely what is needed because we want Form1 processing suspended until the requested user input is received.

Following the closure of Form3, the updated default parameters are printed out on Form1. Then, as soon as the program execution reaches line label TESTIN, a conventional print statement is employed to inform the user to select the testing to be performed using the provided pull-down menu. At this point, program execution is suspended awaiting user input.

We created the "Select Test" pull-down menu back in the Chapter 16 subsection ***Creating a Pull-Down Menu.*** Using this rather general purpose graphic provides the program with a simple technique to activate different Events (subroutines) based upon which selection the user makes. If the user selects Wraparound Test, VB activates the **`Mnu_Wrap_Click()`** subroutine and if EXIT is selected the **`Mnu_Exit_Click()`** subroutine. I'll walk through the Output Card Test first (starting at **Private Sub Mnu_Out_Click()**) and then we'll look at the Wraparound Test.

The first step in Output Card testing is to check NDP$ to see if the node is an SMINI. If it is, the program branches to line label OTESTM to initialize the SMINI node set up for Output Card testing. If it's not an SMINI, the program falls through the IF-THEN statement to line label OTESTU for initializing the USIC/SUSIC node. Both sets of node initialization statements follow the exact pattern we've been using throughout this manual.

Once the proper node is initialized, the program reaches line label OTESTR (picked for Output Card Test repeating) where the actual test code is located. The statements following the OTESTR label closely follow what we've written for the previous Output Demonstration Programs in Chapters 7, 12 and 13.

One exception is the use of the IF-THEN-ELSE statement to select if the printout of card-port information is for an SMINI or for a USIC/SUSIC. Also, the delay loop used to allow time to read the Test Card LEDs is changed to use the SLOWDIS factor rather than a "fixed constant" as in previous examples.

The first step in the Wraparound Test, starting at **`Private Sub Mnu_Wrap_Click(),`** is to use MsgBox to determine if every test line is to be printed or simply if only data lines when there's an error are to be printed. A user pressing the No button sets variable PRTIO (used as a print I/O flag), equal to 0 to suppress every line printout. A user pressing the Yes button sets PRTIO equal to 1 to enable every line printout. It's nice to run with every I/O line printed for general testing and to only print error lines when letting a test run for long periods of time (like overnight) for counting the number of random communication errors.

At line label CHKNDP, the program checks NDP$ to see if the node is an SMINI. If it is, the program branches to line label WTESTM to initialize the SMINI node set up for wraparound testing. If it's not an SMINI the program falls through the IF-THEN statement to line label WTESTU for initializing the USIC/SUSIC node. Both sets of node initialization statements follow the exact pattern we've been using throughout this manual.

Once the proper node is initialized, the program reaches line label WTESTR (picked for Wraparound Test repeating), where the actual test code is located. The statements following the WTESTR label closely follow what we've written for the previous Wraparound Display Programs in Chapters 7, 12 and 13. One exception is the addition of the INFILTER delay loop to account for adding input line filtering to input ports. Another exception is the built-in branching based upon the value of the PRTIO flag. When PRTIO equals 1 every I/O line is printed and when PRTIO equals 0 only I/O lines with errors are printed.

That's really all there is to understanding the content of the USTPVB program.

SOURCE LISTING FOR USTPVB.FRM

The remainder of this appendix is devoted to the Form1 and Module2 source code listings. The Form1 listing does not include the VB self-generated statements. The complete file, including the VB self-generated statements, is provided on disk as file USTPVB.FRM.

```
Rem****************USTPVB.FRM************************
   Rem*****      UNIVERSAL CARD TEST PROGRAM         *****
   Rem**COMBINED OUTPUT CARD AND WRAPAROUND TEST PROGRAM**
   Rem**           VISUAL BASIC CALL VERSION            **
   Rem**   FOR USE WITH USIC, SUSIC AND SMINI NODES     **
   Rem**   WRITTEN BY BRUCE CHUBB AND ALAN ANDERSON     **
   Rem**               August 5, 2003                   **
   Rem**************************************************
      Rem**DIMENSION VARIABLES USED IN MAINLINE ONLY
      DefInt A-Z
      Dim PNU$(4) 'Array used to printout port letter for SUSIC/USIC nodes
      Dim PNM$(6) 'Array used to printout card-port for SMINI nodes

Private Sub Form_Load()
   Rem**DEFINE I/O CARD PORTS FOR USIC AND SUSIC NODES
      PNU$(1) = "A": PNU$(2) = "B": PNU$(3) = "C": PNU$(4) = "D"

   Rem**DEFINE OUTPUT CARD PORTS FOR SMINI NODE
      PNM$(1) = "0 A": PNM$(2) = "0 B": PNM$(3) = "0 C"
      PNM$(4) = "1 A": PNM$(5) = "1 B": PNM$(6) = "1 C"

   Rem**DEFINE MAXIMUM BUFFER SIZE WITHIN MSCOMM BEFOR DECLARE PC OVERRUN ERROR
      MAXBUF = 50

   Rem**SET UP TO PRINT FORM1 AND CLEAR SCREEN
      Form1.Show
      Cls

   Rem**CHECK IF TESTING SMINI NODE TO PRINT HARDWARE ASSUMPTIONS**
NODEIN:
      x = MsgBox("Are you testing a SMINI node?", vbYesNo, "CMRI Test")
      If x = vbYes Then GoTo INSTM

INSTU:
   Rem**PRINT OUT HARDWARE SETUP ASSUMPTIONS FOR USIC AND SUSIC**
      Print " "
      Print "******FOR OUTPUT CARD TESTING WITH USIC AND SUSIC*****"
      Print "1) Node's USIC address must be 0 (UA DIP switch all off)"
      Print "2) Output card in card address slot 0 (DIP switch all off)"
      Print "3) Compatible test card to be mounted on output card"
      Print " "
      Print "******FOR WRAPAROUND TESTING WITH USIC AND SUSIC*****"
      Print "1) Node's USIC address must be 0 (UA DIP switch all off)"
      Print "2) Output card in card address slot 0 (DIP switch all off)"
      Print "3) Output card must be standard current sinking configuration"
      Print "4) Input card in card address slot 1 (DIP switch right segment on)"
      Print "5) Wraparound cable to be connected between output and input cards"
      Print "6) Adding a second output card, set to the same card address..."
      Print "   ...slot as the first output card, with an appropriate test..."
      Print "   ...card attached enhances wraparound test debug if a..."
      Print "   ...problem is detected with the input card."
      Print " "

   Rem**DEFINE NODE DEFINITION PARAMETER FOR USIC AND SUSIC APPLICATIONS**
BITIN:
      Print "Note: If using classic USIC node, I/O cards must be 24-bit"
      Print " "
      x = MsgBox("Are you using 32-bit cards?", vbYesNo, "CMRI Test")
      If x = vbNo Then NDP$ = "N": GoTo COMIN
      If x = vbYes Then NDP$ = "X": GoTo COMIN

INSTM:
   Rem**PRINT OUT HARDWARE SETUP ASSUMPTIONS FOR SMINI**
      Cls
```

```
      Print " "
      Print "******FOR OUTPUT CARD TESTING WITH SMINI*****"
      Print "1) Node's SMINI address must be zero (UA DIP switch all off)"
      Print "2) Compatible test card to be mounted on output card 0"
      Print "3) Additional test card can be mounted on output card 1 or..."
      Print "  ...the same test card moved to card 1 once card 0 is tested OK"
      Print " "
      Print "******FOR WRAPAROUND TESTING WITH SMINI*****"
      Print "1) Node's SMINI address must be 0 (UA DIP switch all off)"
      Print "2) Test card to be mounted on output card 0"
      Print "3) Output card 1 must be standard current sinking configuration"
      Print "4) Wraparound cable to be connected between output card 1 and..."
      Print " ...input card 2 (make sure A0 on card 1 connects to A0 on card 2)"

   Rem**DEFINE NODE DEFINITION PARAMETER FOR SMINI APPLICATIONS**
      NDP$ = "M"

   Rem**INPUT DESIRED PC COM PORT AND BAUD RATE**
COMIN:
      Form2.Show 1     'Show Form2 to make COM and Baud rate selections.
                       'The file name is (COM_Baud.FRM). Execution of Form1...
                       '...code stops here to branch to execute Form2 code.
                       'Form2 is automatically closed when the user selects...
                       '...the OK button which then returns control to Form1...
                       '...with the values set for COMPORT and BAUD100.
      Cls
      Print " "
      Print "Use baud rate of 9600 or 19200 advised when testing I/O cards..."
      Print "...as this minimizes the chances that a communication error can..."
      Print "...occur that might be mistakenly identified as a board error"
      Print " "
      Print "The baud rate DIP switch, as well as the UA address DIP..."
      Print "...switch, are read only during the card's power up cycle."
      Print "Therefore, if you change either of these DIP switch..."
      Print "...settings, you must cycle power to the card to activate..."
      Print "...the new switch settings."
      Print " "

ADJPAR:
   Rem**DEFINE INITIALIZED DEFAULT PARAMETERS**
      Print " "
      MAXTRIES = 30000 'Define maximum number of reading input tries for PC
      DL = 0           'Define USIC delay between transmitted bytes to PC
      INFILTER = 0     'Define input filtering delay for wraparound testing
      SLOWDIS = 40     'Define slow display factor to observe test card LEDs

   Rem**PRINT OUT DEFAULT VALUES OF PARAMETERS
      Cls
      Print " "
      Print "DEFAULT PARAMETERS ARE:"
      Print "  MAXTRIES = "; MAXTRIES
      Print "  USIC DELAY (DL) = "; DL
      Print "  INPUT FILTER DELAY (INFILTER) = "; INFILTER
      Print "  SLOW DISPLAY (SLOWDIS) ="; SLOWDIS

   Rem**CHECK IF DESIRE TO CHANGE DEFAULT PARAMETERS**
      x = MsgBox("Do you want to change the default settings?", vbYesNo, "CMRI Test")
      If x = vbNo Then GoTo TESTIN     ' User clicked the No button

   Rem**DISPLAY FORM3 SO USER CAN OVERRIDE DEFAULT PARAMETERS
DFLTIN:
      Form3.Show 1

   Rem**PRINT LIST OF DEFAULT PARAMETERS ON FORM1
```

```
      Print " "
      Print "DEFAULT PARAMETERS HAVE BEEN CHANGED TO:"
      Print "  MAXTRIES = "; MAXTRIES
      Print "  USIC DELAY = "; DL
      Print "  INPUT FILTER DELAY = "; INFILTER
      Print "  SLOW DISPLAY ="; SLOWDIS

TESTIN:
   Rem**AT THIS POINT, BASIC TEST INITIALIZATION INFORMATION IS COMPLETE SO...
   Rem** ...WAIT FOR USER TO SELECT TEST DESIRED VIA PULL-DOWN MENU SELECTION
   Rem**MENU SELECTIONS PROVIDED DIRECTLY ON FORM1 ARE:
      ' Output card test - when selected VB activates subroutine Mnu_Out_Click()
      ' Wraparound card test - when selected VB activates subroutine Mnu_Wrap_Click()
      ' Exit testing - when selected VB activates subroutine Mnu_Exit_Click()

      Print " "    'Print instruction for user to pull-down menu for selecting test
      Print " Select Test from Pull-Down Menu"

   Rem**TERMINATE MAIN PROGRAM SUBROUTINE TO WAIT FOR USER DEFINITION OF...
   Rem**        ...TESTING TO BE PERFORMED VIA MENU SELECTION
End Sub

Private Sub Mnu_Out_Click()

   Rem****************************************
   Rem**BEGIN USER PROGRAM TO TEST OUTPUT CARD**
   Rem****************************************
      Cls
      Print "PERFORMING OUTPUT CARD LED DRIVER TEST"
      Print "SELECT 'EXIT' FROM THE PULL-DOWN MENU TO EXIT"
      Print " "

   Rem**CHECK AND BRANCH TO SEPARATE TEST INITIALIZATION FOR SMINI**
      If NDP$ = "M" Then GoTo OTESTM

   Rem**INITIALIZE USIC AND SUSIC
OTESTU:
      Print "PERFORMING OUTPUT CARD TEST ON USIC OR SUSIC"
      Print " "
      UA = 0       'USIC address
      NS = 1       'Number of card sets of 4
      NI = 0       'Number of input ports
      If NDP$ = "N" Then NO = 3 'Define number of output ports 24-bit card
      If NDP$ = "X" Then NO = 4 'Define number of output ports 32-bit card
      CT(1) = 2    'Card type definition (one output card in slot 0)
      Call INIT    'Invoke initialization subroutine
      t$ = "Node Initialization is complete - check green..." + Chr$(13) + Chr$(10)
      t$ = t$ + "...status LED for 1 second on 1 second off blink rate"
      MsgBox t$, vbOKOnly, "CMRI Test"
      GoTo OTESTR  'Branch to start output card test repeating

   Rem**INITIALIZE SMINI**
OTESTM:
      Print "PERFORMING OUTPUT CARD TEST ON SMINI"
      Print " "
      UA = 0       'USIC address
      NS = 0       'Number of 2-lead yellow aspect oscillate signals
      NI = 3       'Number of input ports
      NO = 6       'Number of output ports
      Call INIT    'Invoke initialization subroutine
      t$ = "Node Initialization is complete - check green..." + Chr$(13) + Chr$(10)
      t$ = t$ + "...status LED for 1 second on 1 second off blink rate"
      MsgBox t$, vbOKOnly, "USTPVB Alert"
```

```
   Rem**BEGIN COMMON CODE OF ALL OUTPUT CARD TESTING**
   Rem**INITIALIZE ALL LEDS TO OFF
OTESTR:
      For I = 1 To NO
         OB(I) = 0
      Next I

   Rem**INCREMENT PORT TO BE TESTED IN A LOOP
      For PN = 1 To NO

   Rem**INCREMENT DISPLAYED BIT NUMBER IN A LOOP
      For N = 0 To 7

   Rem**OUTPUT TEST STATUS TO CRT FOR SMINI AND NON-SMINI
      If NDP$ = "M" Then
         Print "PORT IS   " + PNM$(PN) + "   BIT NUMBER IS  "; N
      Else
         Print "PORT IS   " + PNU$(PN) + "   BIT NUMBER IS  "; N
      End If

   Rem**SETUP TEST LED PATTERN FOR DISPLAY ON TEST CARD
      OB(PN) = 2 ^ N   'Integer 2 raised to the power N shifts...
                       '...the turn-on displayed LED one position...
                       '...to the left each loop through program.

   Rem**OUTPUT LED DISPLAY DATA TO CARD
      Call OUTPUTS    'Invoke transmit data subroutine

   Rem**ADD DELAY TO BE ABLE TO READ LEDs
      DoEvents
      If SLOWDIS > 0 Then
        For I = 1 To SLOWDIS            'Delay by looping to give...
           For IJ = 1 To 10000: Next IJ '...time to visually check...
        Next I                          '...operation of LEDs on test card
      End If

   Rem**COMPLETE LOOPS
      Next N       'Update display to next LED bit position
      OB(PN) = 0   'Last LED within port tested so turn off port LED
      Cls          'Clear screen for next port display
      Next PN      'Update display to next port number

   Rem**REPEAT OUTPUT CARD TEST
      GoTo OTESTR  'All ports completed, so branch back to repeat test
End Sub

Private Sub Mnu_Wrap_Click()
   Rem************************************************
   Rem**BEGIN USER PROGRAM TO PERFORM WRAPAROUND TEST**
   Rem************************************************
      Cls
      Print "PERFORMING WRAPAROUND TEST"
      Print " "

   Rem**CHECK IF DESIRE TO PRINT EVERY I/O LINE OR ONLY ERROR LINES**
CHKPIO:
      t$ = "NOTE: Clicking No to following prints only error lines" + Chr$(13) +
Chr$(10)
      t$ = t$ + "DESIRE TO PRINT EVERY I/O LINE ?"
      x = MsgBox(t$, vbYesNo, "CMRI Test")
      If x = vbNo Then PRTIO = 0: GoTo CHKNDP
      If x = vbYes Then PRTIO = 1: GoTo CHKNDP

   Rem**CHECK AND BRANCH TO SEPARATE TEST INITIALIZATION FOR SMINI**
```

```
CHKNDP:
      If NDP$ = "M" Then GoTo WTESTM

   Rem**INITIALIZE USIC AND SUSIC**
WTESTU:
      Print "PERFORMING WRAPAROUND TEST ON USIC OR SUSIC"
      Print " "
      UA = 0      'USIC address
      NS = 1      'Number of card sets of 4
      If NDP$ = "N" Then NI = 3: NO = 3  'Define no. I/O ports 24-bit cards
      If NDP$ = "X" Then NI = 4: NO = 4  'Define no. I/O ports 32-bit cards
      CT(1) = 6   'Card type definition (output slot 0, input slot 1)
      Call INIT   'Invoke initialization subroutine
      t$ = "Node Initialization is complete - check green..." + Chr$(13) + Chr$(10)
      t$ = t$ + "...status LED for 1 second on 1 second off blink rate"
      MsgBox t$, vbOKOnly, "CMRI Test"
      GoTo WTESTR 'Branch to start wraparound test repeating

   Rem**INITIALIZE SMINI**
WTESTM:
      Print "PERFORMING WRAPAROUND TEST ON SMINI"
      Print " "
      UA = 0      'USIC address
      NS = 0      'Number of 2-lead yellow aspect oscillating signals
      NI = 3      'Number of input ports
      NO = 6      'Number of output ports
      Call INIT   'Invoke initialization subroutine
      t$ = "Node Initialization is complete - check green..." + Chr$(13) + Chr$(10)
      t$ = t$ + "...status LED for 1 second on 1 second off blink rate"
      MsgBox t$, vbOKOnly, "CMRI Test"

   Rem**BEGIN COMMON CODE OF ALL WRAPAROUND TESTING**
   Rem**INITIALIZE ALL OUTPUTS TO ZERO
WTESTR:
      For I = 1 To NO
         OB(I) = 0
      Next I

   Rem**SETUP TO INCREMENT THROUGH ONLY 3 PORTS IF SMINI...
          '...AND THROUGH ALL NO (N-Oh) PORTS IF USIC OR SUSIC
          'Note: During wraparound testing of SMINI...
          '...test program increments test pattern...
          '...through first 3 ports while setting the...
          '...second 3 ports equal to the same pattern.
          'During wraparound testing of SUSIC and USIC...
          '...test program increments test pattern...
          '...through all no ports.
      If NDP$ = "M" Then N3NO = 3 Else N3NO = NO

   Rem**INCREMENT PORT TO BE TESTED IN A LOOP
      For PN = 1 To N3NO

   Rem**INCREMENT TEST PATTERN IN A LOOP
      For D = 0 To 255
      OB(PN) = D
      If NDP$ = "M" Then OB(PN + 3) = D  'For SMINI case also set...
                                  '...second card's port to same bit pattern
   Rem**OUTPUT BIT PATTERN TO OUTPUT CARD
      Call OUTPUTS   'Invoke transmit subroutine
      DoEvents
      For IJ = 1 To 100: Next IJ  'Basic delay to slow program...
                                  '...to read LEDs on output card

   Rem**ADD DELAY TO ACCOUNT FOR INPUT LINE FILTERING
```

```
      If INFILTER > 0 Then
         For I = 1 To INFILTER  'Delay to compensate for input card filtering
            For IJ = 1 To 10    'Stretch length of effective delay
            Next IJ
         Next I
      End If

   Rem**READ BIT PATTERN FROM INPUT CARD
      Call INPUTS  'Invoke receive subroutine

   Rem**OUTPUT TEST STATUS TO CRT
     If PRTIO = 1 Then
       Cls      'Clear screen for next print line
       Print "PORT IS "; PNU$(PN); " DEC OUT IS "; OB(PN); " DEC IN IS "; IB(PN)
     End If

   Rem**COMPARE INPUT TO OUTPUT AND BRANCH TO CONTINUE LOOP IF NO ERROR
      If IB(PN) = OB(PN) Then GoTo CLOOP

   Rem**ERROR FOUND SO PRINT TO CRT
      If PRTIO = 1 Then
       Print "ERROR FOUND IN ABOVE I/O, PRESS YES BUTTON TO CONTINUE TESTING"
       x = MsgBox("Error in above I/O - continue testing?", vbYesNo, "Input Error")
       If x = vbNo Then End
      Else
       Cls      'Clear screen for next print line
       Print "PORT IS "; PNU$(PN); " DEC OUT IS "; OB(PN); " DEC IN IS "; IB(PN)
       Print "ERROR FOUND IN ABOVE I/O, CONTINUING TEST"
      End If

   Rem**COMPLETE LOOPS
CLOOP:
      Next D       'Increment to next bit pattern for port
      OB(PN) = 0   'Zero out port before starting new port
      If NDP$ = "M" Then OB(PN + 3) = 0  'For SMINI case also set...
                                         '...second card's port to 0

      Next PN    'Increment to next port

   Rem**REPEAT WRAPAROUND TEST**
      GoTo WTESTR  'At end of port loops so branch back to repeat test
End Sub

Private Sub Mnu_Exit_Click()

OPSYS:
      Print " "
      Print "TESTING IS BEING TERMINATED BY USER REQUEST"
      End     'Return to system program

End Sub
```

SOURCE LISTING FOR USTPVB.BAS

The source listing for Module2 (USTPVB.BAS) is simply the one REMark and two Public statements:

```
   REM**USER APPLICATION SPECIFIC GLOBALIZED VARIABLES
      Public SLOWDIS      'Slow display factor
      Public INFILTER     'Input filter delay
```