Column #75, July 2001 by Jon Williams:

Sounding Off...Again! - Part 2

Last month I introduced the QV306M4-P sound module from Quadravox. I hope, by now, you've had a chance to purchase one (available from Parallax or Quadravox) and play with the demo. I'm sure you'll agree that this product is very easy to use and a lot of fun on top of that.

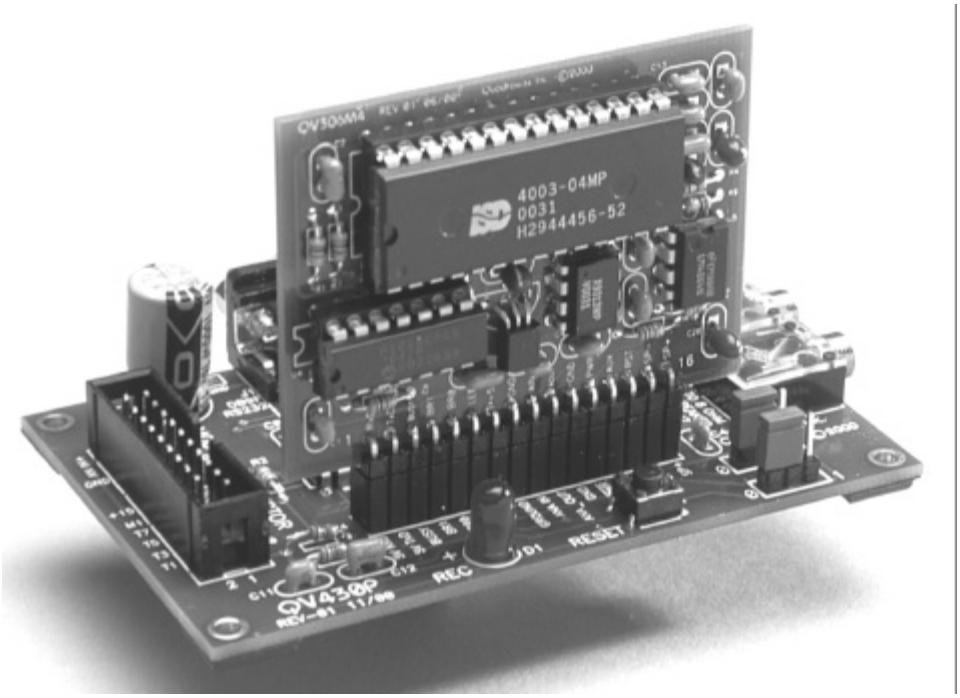
All right, we've seen how easy the pre-programmed version is to use, but what happens when we want to use our own voice or sound effects or music files? That's not a problem at all. The QV306M4 is fully programmable and you can restore the pre-programmed files by downloading them from the Quadravox web site.

Putting New Sounds Into the QV306M4(-P)

Noticeably absent from the QV306M4 is a microphone. This is a deliberate choice. Programming through a microphone allows the introduction of external noises and the annoying clicks and pops of a mechanical switch to enable the microphone, not to mention "dead air" and other timing problems. No thanks.

The QV306M4 will record a line level input on the Ain pin. Commanding the device to start and stop recording is handled through the serial connection.

Figure 75.1: QV430 programming cradle holding the QV306M4 sound module

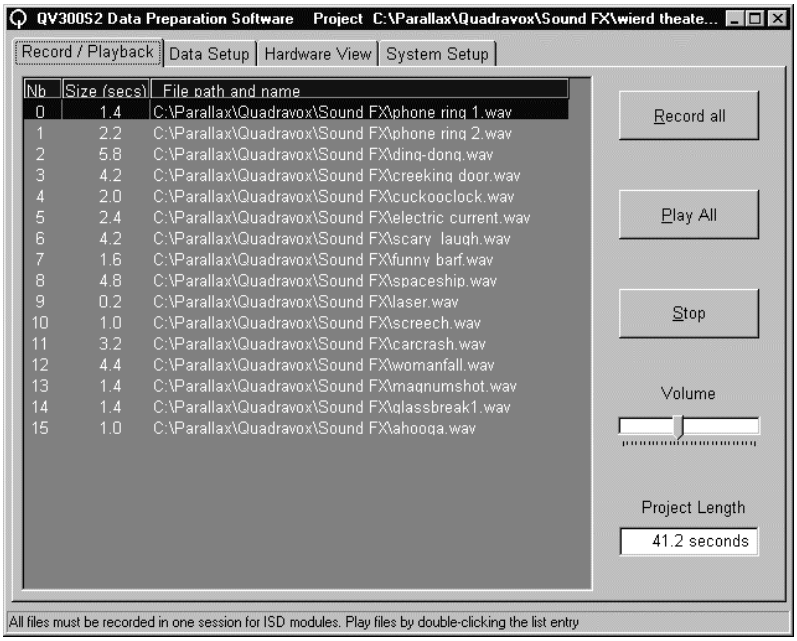


The simplest was to connect the QV306M4 to the outside world for programming is through the (optional) QV430 programming cradle. See Figure 75.1 for the QV306 in it's cradle.

The QV430 provides power, serial level conversion and audio connections for the QV306M4. A serial line is connected to your PC where the device is controlled by the QV300S2 software (available for free download from Quadravox). The audio input connects to your sound card.

The idea behind the QV300S2 software is very simple: you create a list of WAV files to be downloaded into the QV306M4. When you click the "Record All" button, your PC will play the WAV files while the QV300S2 software controls the recording in the QV306M4. It's very easy.

Figure 75.2: Loading files into the QV306 using Quadravox software



Okay, let's do it. Start the QV300S2 software and click on the "Data Setup" tab. If there are existing files listed, click on the "New Project" button to clear the workspace. Next, click the "Add Files" button. This will present a standard Windows file dialog box. You can select one file at a time or, to save time you can hold the Ctrl key will selecting multiple files.

When you've selected your file(s), click on open to place them into your project. Changing the file order is dragging (while holding the left mouse button down) a file to its new location. When you're satisfied with the order of your files, click on "Save Project" to save it to disk. This is shown in Figure 75.2.

To program your QV306M4, place it in the QV430 cradle with its components facing the REC LED. Connect the serial connection to your PC and the SOUND CARD LINE OUT input to the left channel of your PC sound card (this is the tip of a 1/8" mini-plug). Since speed is not a concern while programming, set both baud rate jumpers on the QV430 to 0 (for 2400 baud). You can connect an eight ohm speaker to the QV430, to

listen to your files while programming, but this is not strictly necessary. You will need a nine volt power supply for the QV430.

Now, click on the QV300S2 "System Setup" tab and select QV306 for the module type, set the Serial Port selection to match your PC, set the Baud Rate to 2400 and make sure that "Autonormalize" is selected (this will help smooth the transitions between concatenated files).

Finally, select the "Record/Playback" tab and click "Record All." If you have a speaker connected, you'll hear each file as it's being programmed. Use "Play All" (you'll need a speaker for this) to make sure that everything recorded properly. Once you're satisfied with the recording, you can remove power from the QV430 and move the QV306M4 to your Stamp project.

What I really like about this concept is that I can mix speech and sound effects files. And since the sounds start out on my PC as WAV files, I can tweak and modify them any way I choose. There is an important note here: your WAV files – if they don't meet this spec – should be resampled to 11,025 Hz, 16-bit (Windows PCM), monophonic before downloading into the QV306M4. Personally, I use Sound Forge XP to do the manipulating, and there are myriad freeware and shareware WAV file manipulators on the Internet. Choose what works best for your budget and sound interests.

Look Mom, no Cradle!

Okay, I know what you're thinking: "Dude, this thing is really cool, but I don't want to spend the bucks for a programmer..." Guess what? You don't have to.

Quadravox designed the QV430 for their OEM customers; those folks that have to program a whole bunch of 306s in a big hurry (like the guys who make talking trash cans). We can duplicate the important part of the QV430's functionality with a powered breadboard, a BS2 and a few spare parts.

At its core, the QV430 provides a physical connection for the QV306, a 5-volt power supply for it, an inverter for the serial data stream from the PC and audio connections; one from the PC (audio) in and one to a speaker.

The design of the QV306 makes it easy to plug into a breadboard. Since we also need a 5-volt supply, I used a Parallax Board of Education. The BOE gives me a breadboard, a 5-volt supply and a connection from the PC to the Stamp. This connection will be extended to the QV306 with 22-guage wire.

The code in Listing 75.1 (NOCRADLE.BS2) is loaded into the Stamp. What we're doing, in fact, is using the Stamp as an RS232 to TTL serial converter. This limits what we can do since we don't have a clean connection back from the QV306M4 to the PC (because the Stamp serial port echoes characters). When using this program with QV300S2, only the "Record All" function works (which is all we need to program the QV306M4).

Figure 75.3: Programming the QV306 directly from a BASIC Stamp 2

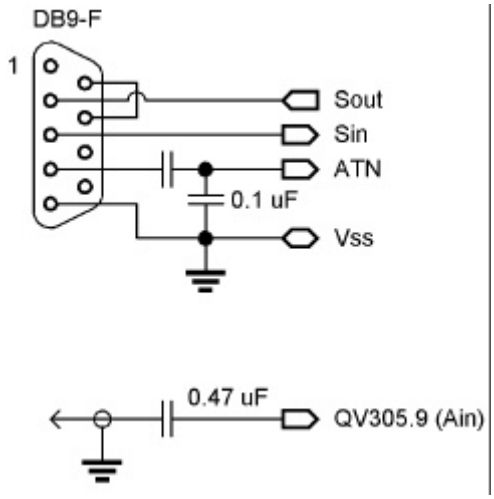
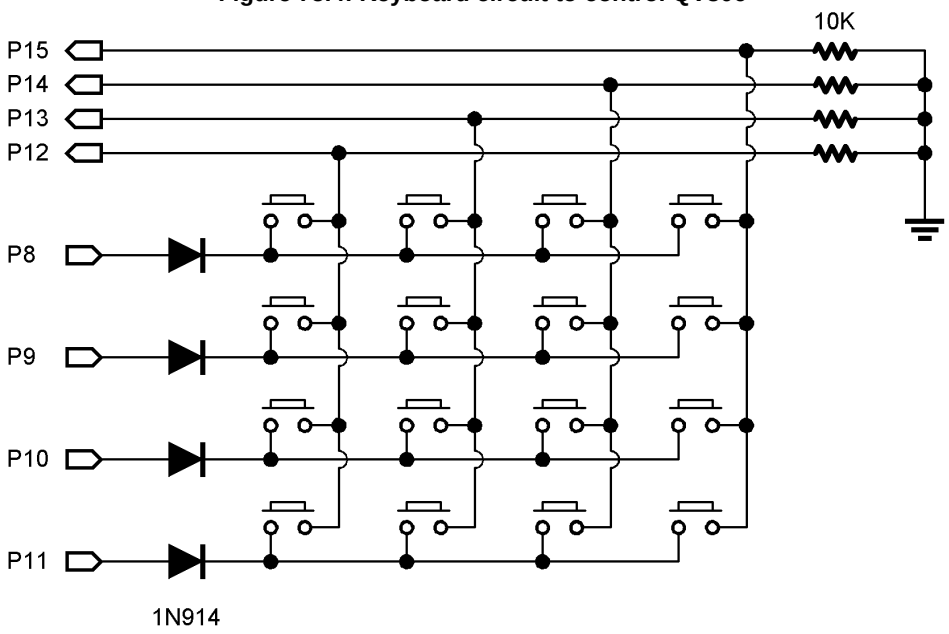


Figure 75.3 shows the schematic for using the BASIC Stamp programming connection as a QV306M4 programmer. If you don't have a BOE, you can easily "roll you own" on a standard breadboard. Just make sure you have the bypass caps in the ATN (DTR) line of the serial connection. If you don't, the program won't run.

Since we can't use the playback functionality of the QV300S2 program with our home-built programmer, the Stamp code will count the number of files downloaded and play them back with the programming software returns the QV306M4 to playback mode. If we don't like what we hear on playback, we make adjustments and click "Record All" again.

Figure 75.4: Keyboard circuit to control QV306



Quiet On The Set!

When I'm not working or playing with Stamps, the focus of my life is on film and television production. Sometimes those two worlds collide. Let me explain.

From time-to-time I work at the local cable access station with a fun guy named Gordon Smith. Gordon does a live show each month called "Weird Theatre." Gordon, the host, plays Mr. Weird, a man of seemingly endless trivia and bad jokes about old horror and sci-fi movies. Of course, he has a beautiful blonde, well-proportioned sidekick. The two of them chat and comment about the movie (while I'm scrambling between cameras – this is a "no budget" show), take calls from those skimming cable access channels that night and generally fill time – we have to put on a two hour show and most of the movies don't run that long.

Gordon loves to play with props while doing his shtick and when working on the last show I had the idea that a sound effects box on the set might be a lot of fun. Radio DJs use them all the time to add appropriate sound-based humor to their programs, it seemed

appropriate for this show. With a Stamp, the QV306M4 and some sort of keyboard, I felt like building one should be a snap. And it is.

The only thing I had to do is add a 4x4 keyboard to the Stamp/QV306M4 circuit. Since I want a lot of room for labels on my box, I decided to use 16 normally-open pushbutton switches.

With the switches, four 1N914 diodes, four 10K resistors and a bit of wire, I had a keyboard. Listing 75.2 (SOUNDFX.BS2) is the code for this project and the circuit is shown in Figure 75.4.

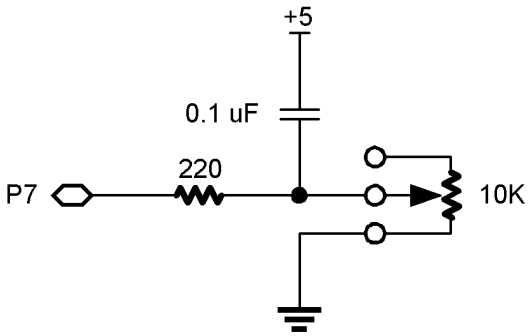
Scanning the keyboard is pretty simple. The program keeps a variable called keyMap that contains the state of the keyboard after the last scan. A "1" corresponds to the key being pressed. The nature of the Scan_4x4 subroutine will allow for any number of keys to be pressed. This particular program only accepts one, using the NCD function to pull out the highest bit. You could easily modify this though, using one button as a "shift" key and decoding the others. This would give you 30 possibilities: 15 unshifted and 15 shifted. I've written the code (just two lines) to handle this, but have removed it with comments. I'll leave the choice to you.

The scan routine starts by refreshing the row output registers. The row outputs are directed through steering diodes that will prevent a short should we make a programming change that allows more than one row to be on at a time. The outer loop activates a row by setting the appropriate direction bit (part of DIRS) to one, then scans all four columns at the same time. The column inputs are read and debounced by ANDing their values to the last scan. If a key is pressed through the entire inner loop, it is considered debounced and will be placed in keyMap. At the end of the routine, the row outputs are cleared and keyMap is returned to the main code.

When keyMap is greater than zero, the QV306M4 is commanded to play the file. Since the busy line is monitored before sending a file, you can press a key while a sound is playing. This will cause the second sound file to be queued (by the code, not the QV306M4), causing it to play as soon as the current sound finishes

After getting the project working, I decided to add a mechanical volume control (Figure 75.5). This is a very simple arrangement using the standard RCTIME circuit. The pot value is read and scaled to the range of 0 to 31, then sent to the QV306M4 if there has been a change. If you decide to leave out the volume control pot, simply change the VolCtrl constant to "No." If you're porting the code to a BS2sx or BS2p, you'll need to measure the maximum raw RCTIME value and update the VMax constant.

Figure 75.5: Volume control for QV306 controlled by BASIC Stamp



This was a fun project. I'm upgrading it to add a programming and audio-in connectors. That way I don't have to pull the QV306M4 from the box to change my sound effects. Of course, I'll be building a remotely controllable version of this box for Halloween – that will be a lot of fun.

There's More Coming

Like I said last month, sound projects – especially those using ISD chips – have generated more e-mail than anything I've ever written about. That hasn't gone unnoticed by Parallax. The sheer coolness of the QV306M4, the ease of use with the BASIC Stamp and with so many customer requests coming, Parallax and Quadravox have joined forces to develop new products. Stay tuned, there's some really cool stuff on the way.

Until next time, happy Stamping.

```
' Program Listing 75.1
' Nuts & Volts -- July 2001

' ----[ Title ]---------------------------------------------------------
'
' File..... NOCRADLE.BS2
' Purpose... Programs QV306M4 without QV430 programming cradle
' Author.... Jon Williams
' E-mail.... jonwms@aol.com
' Started... 22 APR 2001
' Updated... 22 APR 2001

' {$STAMP BS2}


' ----[ Program Description ]-------------------------------------------
'
' This progam facilitiates the programming of the QV306M4 sound module
' without the expense of the QV430 programming cradle.  Since this
' program simply passes seial messages from the PC to the QV306M4, not
' all QV300S2 functions are available.
'
' When started, the program resets the QV306M4 to ensure control.  Once
' ready, the Record LED is flashed briefly.  When the record command is
' received from the QV300S2 software, the Record LED is lit solid.  When
' all files are recorded, the QV300S2 software will send the QV_Direct
' (play mode) command.  When this is detected, the LED is flashed and the
' recorded files are played by the Stamp.
'
' Note: After starting the QV300S2 software, reset the Stamp and allow it
' to get ready before clicking on "Record All."


' ----[ Revision History ]----------------------------------------------
'
' 22 APR 2001 : Version 1 tested and working with QV300S2 "Record All"


' ----[ Connections ]---------------------------------------------------
'
' QV306M4 Connections:
'
'  1 (RxD)            Stamp.P1
'  2 (TxD)            Stamp.P2
'  3 (Busy)           Stamp.P3
'  4 (BR1)            Ground
'  5 (BR0)            Ground
'  7 (+5)             +5 volts
'  8 (Gnd)            Ground
'  9 (Ain)            Analog in from sound card (through 0.47 uF cap)
```

```
' 14 (Reset)          Stamp.P0
' 15 (Sp-)            8 ohm speaker -
' 16 (Sp+)            8 ohm speaker +
'
'
' LED to Stamp.P7 through 220 ohm resistor


' ----[ I/O Definitions ]---------------------------------------------
'
PC_in          CON     16                    ' serial via programming port
RecLED         CON     7                     ' recording LED

QV_RST         CON     0                     ' QV306M4.14
QV_RX          CON     1                     ' QV306M4.1
QV_TX          CON     2                     ' QV306M4.2
QV_BUSY        VAR     In3                   ' QV306M4.3


' ----[ Constants ]---------------------------------------------------
'
N2400          CON     16780                 ' 2400 baud, inverted
T2400          CON     396                   ' 2400 baud, true

QV_Direct      CON     $F0                   ' play mode
QV_Record      CON     $F4                   ' record mode

IsBusy         CON     0


' ----[ Variables ]---------------------------------------------------
'
pcByte         VAR     Byte                  ' command byte from PC
files          VAR     Byte                  ' sound files downloaded
temp           VAR     Byte                  ' general purpose


' ----[ Initialization ]----------------------------------------------
'
Initialize:
  LOW QV_RST                                 ' reset the QV306M4
  PAUSE 100
  HIGH QV_RST
  PAUSE 2000

IsReady:
  files = 0
  HIGH RecLED                                ' blip LED to indicate ready
  PAUSE 500
  LOW RecLED
```

```
' ----[ Main Code ]-----------------------------------------------------
'
Main:
  SERIN PC_in,N2400,[pcByte]                 ' get byte from PC
  SEROUT QV_RX,T2400,[pcByte]                ' pass it to QV306M4

CheckRec:
  IF (pcByte <> QV_Record) THEN CheckEOF
  HIGH RecLED                                ' light LED if record
  GOTO Main

CheckEOF:
  IF (pcByte <> 0) THEN CheckPlay            ' end of sound file?
  files = files + 1                          '  - yes, update counter
  GOTO Main

CheckPlay:
  IF (pcByte <> QV_Direct) THEN Main

IsDone:                                      ' flash LED when done
  FOR temp = 1 TO 10
    TOGGLE RecLED
    PAUSE 100
  NEXT
  LOW RecLED

Playback:                                    ' play recorded files
  IF (files = 0) THEN IsReady
  FOR temp = 0 TO (files - 1)
    GOSUB Say_Phrase
  NEXT

LetPlaybackFinish:
  IF (QV_Busy = IsBusy) THEN LetPlaybackFinish

  GOTO Initialize                            ' do it again


' ----[ Subroutines ]----------------------------------------------------
'
' ************************
' Say an individual phrase
' ************************
'
Say_Phrase:
  IF (QV_Busy = IsBusy) THEN Say_Phrase      ' wait until not busy
  SEROUT QV_RX,T2400,[temp]                  ' say the phrase
  RETURN
```

```
' Listing 2
' Nuts & Volts -- July 2001

' ----[ Title ]---------------------------------------------------------
'
' File..... SOUNDFX.BS2
' Purpose... Sound FX player; uses BS2 and QV306M4
' Author.... Jon Williams
' E-mail.... jonwms@aol.com
' Started... 25 APR 2001
' Updated... 07 MAY 2001

' {$STAMP BS2}


' ----[ Program Description ]-------------------------------------------
'
' Scans a 4x4 matrix keypad and plays corresponding sound effect from
' QV306M4 module.  There are many sound effects sites on the Internet,
' I found several files at:
'
' -- http://soundamerica.com
' -- http://soundresource.net
'
' WAV files need to be resampled to 11,025 Hz, 16-bit (Windows PCM), Mono
' before downloading to the QV306M4


' ----[ Revision History ]----------------------------------------------
'
' 25 APR 2001 : Completed and working
' 03 MAY 2001 : Added forced key release -- prevents unwanted duplicates
' 07 MAY 2001 : Added "shifted" keyboard code (commented out of this demo)


' ----[ Connections ]---------------------------------------------------
'
' QV306M4 Connections:
'
'  1 (RxD)            Stamp.P1
'  2 (TxD)            Stamp.P2
'  3 (Busy)           Stamp.P3
'  4 (BR1)            Ground
'  5 (BR0)            Ground
'  7 (+5)             +5 volts
'  8 (Gnd)            Ground
' 14 (Reset)          Stamp.P0
' 15 (Sp-)            8 ohm speaker -
' 16 (Sp+)            8 ohm speaker +
'
' Volume in           Stamp.P7 (RCTIME: 10K pot, 0.1uF cap)
```

```
'
' Keys.R0             Stamp.P8       |
' Keys.R1             Stamp.P9       +-- OutC
' Keys.R2             Stamp.P10      |
' Keys.R3             Stamp.P11      |
'
' Keys.C0             Stamp.P12      |
' Keys.C1             Stamp.P13      +-- InD
' Keys.C2             Stamp.P14      |
' Keys.C3             Stamp.P15      |


' ----[ I/O Definitions ]---------------------------------------------------
'
QV_RST          CON     0                       ' QV306M4.14
QV_RX           CON     1                       ' QV306M4.1
QV_TX           CON     2                       ' QV306M4.2
QV_BUSY         VAR     In3                     ' QV306M4.3

VolPot          CON     7

Rows            VAR     DirC                    ' keyboard row (outputs)
Row_pins        VAR     OutC
Cols            VAR     InD                     ' keyboard column (inputs)


' ----[ Constants ]---------------------------------------------------------
'
T2400           CON     396                     ' 2400 baud, true
IsBusy          CON     0

NoKey           CON     $FF                     ' flag value for no key
pressed

Yes             CON     1
No              CON     0

VolCtrl         CON     Yes                     ' use volume control POT?
VMax            CON     645                     ' RCTIME val at max volume
VScale          CON     VMax / 31               ' scale value for volume
control

' ** QV306M4 Command Set **

QV_Direct       CON     $F0                     ' QV modes
QV_Stop         CON     $F6
QV_Sleep        CON     $F8
QV_Volume       CON     $FC
QV_Reset        CON     $FD                     ' software reset
QV_Rev          CON     $FE                     ' module revision
QV_Type         CON     $FF                     ' module type
```

```
' ----[ Variables ]-----------------------------------------------------
'
volume         VAR     Word
lastVol        VAR     Byte           ' last volume setting
keyMap         VAR     Word           ' keyboard map (1 = pressed)
colsIn         VAR     Nib            ' keyboard column inputs
phrase         VAR     Byte           ' phrase to play
lastPhrase     VAR     Byte
scan           VAR     Nib            ' row scan control
dbnc           VAR     Nib            ' debounce loop counter
shiftKey       VAR     Bit            ' shift state


' ----[ Initialization ]-------------------------------------------------
'
Initialize:
  LOW QV_RST                          ' reset the QV306M4
  PAUSE 100
  HIGH QV_RST
  PAUSE 2000


' ----[ Main Code ]------------------------------------------------------
'
Main:
  IF (VolCtrl = No) THEN GetKey       ' using volume pot?

GetVolume:
  HIGH VolPot                         ' discharge volume cap
  PAUSE 1
  RCTIME VolPot,1,volume              ' read volume pot
  volume = volume / VScale MAX 31     ' scale volume
  IF (volume = lastVol) THEN GetKey   ' no change, check keyboard
  GOSUB Set_Volume                    ' if changed, send it to QV306
  lastVol = volume                    ' save latest volume

GetKey:
  GOSUB Scan_4x4                      ' scan the keyboard

  ' shiftKey = keyMap.Bit15           ' get shift state
  ' keyMap.Bit15 = 0                  ' remove shift from map

  IF (keyMap > 0) THEN DecodeKey      ' any key pressed?
  lastPhrase = NoKey
  GOTO Main

DecodeKey:
  phrase = (15 * shiftKey) + (NCD keyMap) - 1
  IF (phrase = lastPhrase) THEN Main   ' force key release
```

```
  GOSUB Say_Phrase                    ' say the corresponding file
  lastPhrase = phrase

  GOTO Main


' ----[ Subroutines ]-------------------------------------------------
'
' *****************
' Scan 4x4 keyboard
' *****************
'
Scan_4x4:
  keyMap = 0                          ' clear keyboard map
  Row_pins = %1111                    ' refresh row outputs

  FOR scan = 0 TO 3                   ' row scan
    Rows = 1 << scan                  ' enable a row
    colsIn = %1111                    ' assume press
    FOR dbnc = 1 TO 10                ' debounce
      colsIn = colsIn & Cols          ' scan column inputs
      PAUSE 2
    NEXT
    keyMap.LowNib(scan) = colsIn      ' update keyboard map
    Rows = %0000
  NEXT
  RETURN


' ************************
' Say an individual phrase
' ************************
'
Say_Phrase:
  IF (QV_Busy = IsBusy) THEN Say_Phrase ' wait until not busy
  SEROUT QV_RX,T2400,[phrase]         ' say the phrase
  RETURN


' *****************
' Set module volume
' *****************
'
Set_Volume:
  IF (QV_Busy = IsBusy) THEN Set_Volume  ' wait for Busy to release
  SEROUT QV_RX,T2400,[QV_Volume]
  SEROUT QV_RX,T2400,[volume.LowByte]
  RETURN
```