



Column #74, June 2001 by Jon Williams:

Sounding Off! Again - Part 1

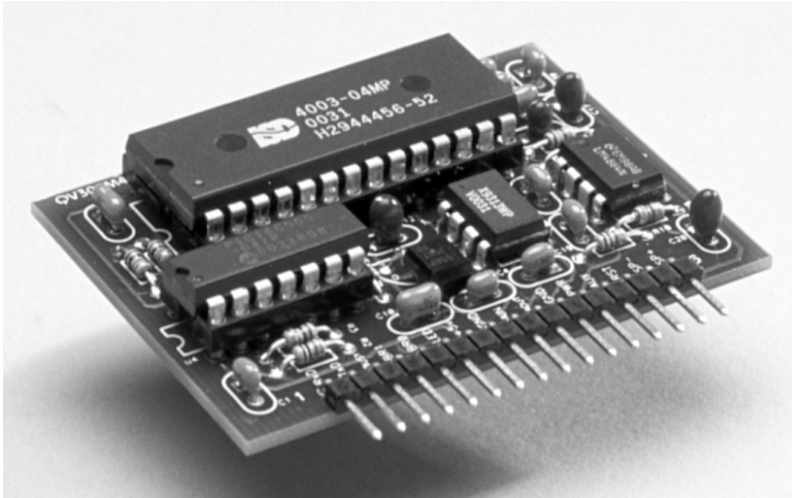
Of the twenty-something articles I've written for Nuts & Volts, none has generated as much e-mail as my articles on using the ISD2560. Any why not? Adding sound to BASIC Stamp projects is a lot of fun, and now it's a whole lot easier.

I've said it before: the Dallas metroplex is a great place. A recent discovery for me is a local company called Quadravox. These guys, all former engineers with Texas Instruments, know their sound and speech stuff. They know it really well. So much so that these are the folks who create the development tools for ISD (now part of Winbond). Since leaving TI, they have focused on making ISD devices easy to use and integrate into OEM products.

The Quadravox QV306M4-P

Of its product line, the Quadravox QV306M4-P is probably the most interesting offering for the Stamp user. The Quadravox guys think so too – their documentation includes a BS2 demonstration (that we'll expand on here).

Figure 74.1: QV306M4-P with ISD chip, audio amplifier and serial controller



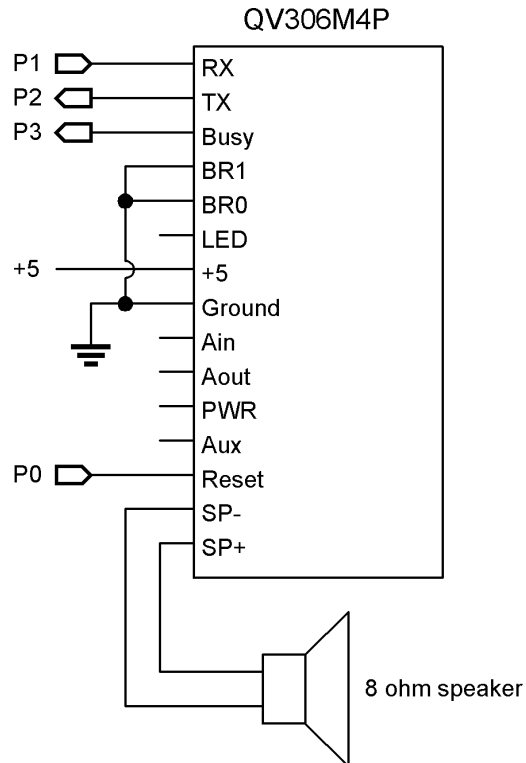
The QV306M4-P is a small (2.3" x 1.7") PCB that holds an ISD4003-04, an audio amplifier and a serial controller (programmed PIC) that makes interfacing the device a snap. The QV306M4-P connects to the outside world through a 16-pin SIP header. Figure 74.1 shows a picture of the QV306MV-P.

The heart of the unit is, of course, the ISD4003-04. This gives the QV306M4-P the ability to store up to four minutes of high-quality sound. The "-P" designation at the end indicates that the unit comes pre-programmed from Quadravox. There are 240 pre-programmed sound files; professionally-recorded numbers, letters and technical terms.

Playing any of these "files" is as simple as sending the file number (0 to 239) through a serial connection. That's it! Tough, huh? Volume control and low-power (sleep) mode is handled serially as well. The QV306M4-P is a Stamp-users dream. Since the QV306M4-P doesn't have a multi-byte serial buffer, there is a busy line to indicate that the device is currently playing. We'll monitor the busy line when creating complex phrases.

Okay, let's just hook it up and try it out as shown in Figure 74.2. The connections to the Stamp – four of them – are easy. To hear how good the unit sounds, you'll need to connect a decent-quality, enclosed speaker. If you use a small speaker that is not enclosed, you'll be robbed – trust me on that.

Figure 74.2: Connecting the QV306M4P to the BASIC Stamp requires only four connections and a decent-quality enclosed speaker

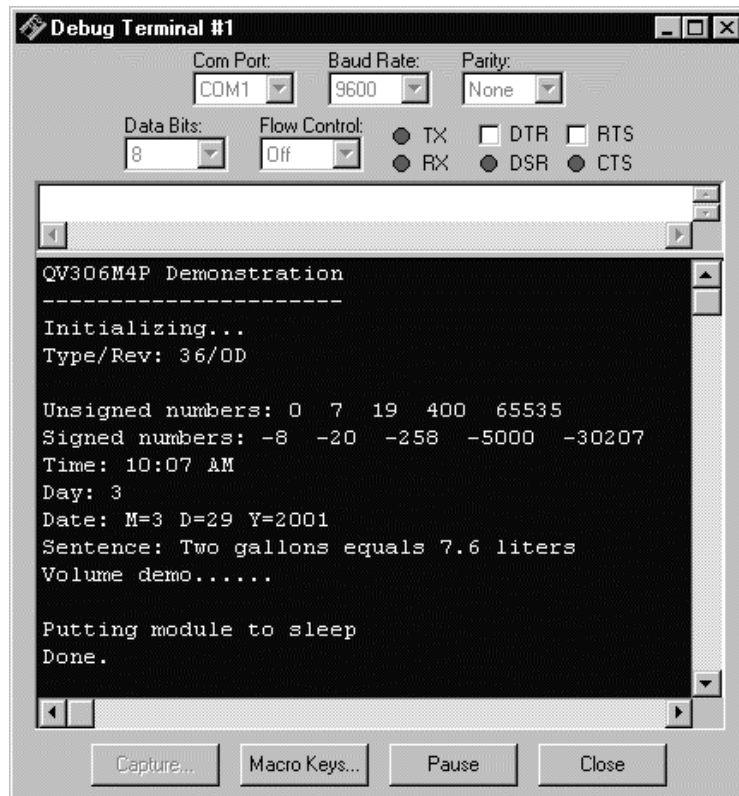


The onboard amplifier is capable of delivering 300 mW, so a nice speaker will give you a room full of sound. I picked up a little 8-ohm bookshelf speaker from Radio Shack and it works great.

Program Listing 74.1 (QV306M4P.BS2) is the updated demo. This code takes advantage of the preprogrammed sound files in the QV306M4-P. Don't let the length of the listing fool you; this program is deceptively simple.

We start by initializing the QV306M4-P. This is done by momentarily dropping the RESET line. When raised, we'll pause a bit to allow the QV306M4-P to count the number of stored sound files. If we ever send a file that is out-of-range, the QV306M4-P will play its first file.

Figure 74.3: Demo BASIC Stamp code in DEBUG screen being run on the module



The first step in the demo is to display the device and revision type. Each command sends a single byte and expects one back from the QV306M4-P. The subroutines for these functions include a serial timeout so that they “don’t” hang if you choose not to connect the serial input. Once we’ve displayed the device type, it’s time to start talking.

As I indicated earlier, the QV306M4-P comes preprogrammed with 240 sound files. The first set of files are numbers that, when properly concatenated (chained together), will cause the device to say just about any number you wish. This is convenient for talking devices like clocks and thermometers.

Since the Stamp uses 16-bit numbers, I created a couple of subroutines to say them. The first to be demonstrated is called Say_Unsigned. This subroutine will “say” (literally) any number between zero and 65,535. A small loop in the main body feeds a table of numbers to exercise all the elements of the Say_Unsigned subroutine.

The nature of Say_Unsigned is to parse the number passed to it and then say the parts just as we would. If our number is 65,535, the routine will start by saying “Sixty five thousand.” The routine separates thousands, hundreds and then the rest. It is supported by another routine, Say_Val_XX, which will say any value between 0 and 99.

Say_Val_XX checks to see if the value passed is 20 or higher. The reason for this is that every number lower than 20 is a unique file. Anything 20 and higher is created by saying the tens value, then the ones. Notice that all the organization of the sound files (“zero” is file 0, etc.) makes this routine easy to use. Should you decide to custom-program your QV306M4-P, it’s a good idea to keep the numbers as they are so you don’t have to fix these subroutines.

Saying a negative number is just as easy. In the BASIC Stamp, a signed number is negative when bit 15 is set to one. In this case, the other bits represent the two’s compliment value of the number. If bit 15 is set, the QV306M4-P will say “negative,” then covert the value to positive with the ABS function and pass it on to Say_Unsigned to finish up.

The next part of the demo says a time; in this case using the 12-hour, AM/PM format. By taking advantage of Say_Val_XX, this routine is fairly straightforward. The only thing we need to accommodate is the way we say time values that have minutes less than ten. In that case, we tend to say “oh-six” if the time is six minutes past the hour.

Say_Time12 does a quick range check, announces “The current time is...” then says the hours using the Say_Val_XX subroutine. It then checks to see if the minutes are less than ten. When this is the case, the QV306M4-P will say “oh” and then the minutes value. When the minutes is ten or greater, we skip the “oh” and say the value with Say_Val_XX. Since “AM” and “PM” are stored in consecutive locations, at bit of math is all that is required to select the correct phrase based on this one-bit value.

Moving on, we’ll use the QV306M4-P to say the date. By now you’ve got the hang of how this works, so I’m not going to rehash the details. The only note here is that when the century value is 19 and the year value is less than ten, we have to insert the “oh” before the year, just as we did with the time.

Finally, there will be projects where we’ll want to create a complex phrase or sentence from the fragments that are included in the QV306M4-P. The method I prefer is to build the complex phrase in a DATA table, then play each phrase in order. Since the valid range for phrase files is zero to 239, any value greater than 239 will cause the Say_Sentence subroutine to terminate. I chose to use \$FF as the terminating value in the

Column #74: Sounding Off! Again - Part 1

demo. To play a stored sentence, we need to put the starting location in the variable, `addr`, then call `Play_Sentence`.

One of the coolest aspects of the QV306M4-P is the digitally controlled amplifier. The onboard controller actually modulates the amplifier to eliminate the clicks and pops that seem inherent with the ISD chip. The last portion of the demo repeats the word “Hello” through the volume range of the circuit.

Next Time...

Well, with our long code listing, that’s about all we can squeeze in this month. Next month, we’ll continue working with the QV306M4 by putting our own sounds into it and creating a sound effects sbox for fun. Be sure to visit the Quadravox or Parallax web sites and download their excellent documentation.

Jon Comes Out of the Closet

In April I had the opportunity to attend the Embedded Systems Conference in San Francisco. It was a lot of fun; there were some really neat things to see. One of my greatest pleasures was the very cool opportunity to some of the readers of this column. That was fun. It’s one thing to get an e-mail, quite another to meet that person face-to-face. I strive to keep my writing easy-going and accessible, so it was a pleasure to meet a few of you that felt like you “know” me. And now you do.

Well, that works both ways. I had the opportunity to Steve Ciarcia, well-known electronics guru and publisher of Circuit Cellar magazine. Like many of you, I’ve been reading Steve’s materials and learning from him for a long time. Meeting him was a genuine thrill. Imagine my surprise when he told me he knew who I was – from this column, no less! Way cool.

And then it happened.... With a furrowed brow, Steve looked right at me and said, “But you don’t actually work for Parallax, do you?” (At the time I was standing in the Parallax booth wearing Parallax attire.) “Yes,” I replied, “I’ve been working for Parallax since August.”

In a nutshell, Steve took exception (or raised an exception for your hardcore programmers) to me not coming out early on and telling you that Parallax had hired me. So now I’m doing it. Does my employment with Parallax change the way I feel about Stamps and how I approach this column? Of course not!

And now I get to work “on the inside” with Parallax’s incredibly talented engineering staff to make Stamps more fun to use.

Okay, there you have it – I’m out. I work for Parallax now (Applications Engineer). I still live in Dallas and I’m still working part-time as an actor (commercials, mostly, but recently worked two days on the Matthew Perry comedy, “Servicing Sara”). I’ve never tried to hide my Parallax employment, I just didn’t think it was necessary to wave (set?) a flag. Parallax did make an announcement to its distributors. Perhaps I should have said something here sooner.

This column is – and always has been – about programming BASIC Stamps. And through an incredible bit of good luck, I happened to land a job that pays me to do that.

Column #74: Sounding Off! Again - Part 1

```
' ----[ Title ]-----
'
' File..... QV306M4P.BS2 (Program Listing 76.1)
' Purpose... Quadravox QV306M4P (preprogrammed) Demonstration
' Author.... Michel Stella (Quadravox) and Jon Williams
' E-mail.... jonwms@aol.com
' Started...
' Updated... 29 MAR 2001

' {$STAMP BS2}

' ----[ Program Description ]-----
'
' This program demonstrates the Quadravox QV306M4P, preprogrammed
' sound module. The "P" version contains 240 professionally recorded
' phrases (numbers, letters and technical words).

' ----[ Revision History ]-----
'
' 29 MAR 2001 : Updated by Jon Williams
'               - improved numeric parsing algorithms
'               - added signed number capability
'               - added stored sentence subroutine

' ----[ Connections ]-----
'
' QV306M4P Connections:
'
' 1 (Rx)D)          Stamp.P1
' 2 (Tx)D)          Stamp.P2
' 3 (Busy)          Stamp.P3
' 4 (BR1)           Ground
' 5 (BR0)           Ground
' 7 (+5)            +5 volts
' 9 (Gnd)           Ground
' 14 (Reset)        Stamp.P0
' 15 (Sp-)          8 ohm speaker -
' 16 (Sp+)          8 ohm speaker +

' ----[ 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
```



```

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

' ----[ Vocabulary ]-----
'
_zero          CON    0          ' message # in QV306M4P
_one           CON    1
_two           CON    2
_three         CON    3
_four          CON    4
_five          CON    5
_six           CON    6
_seven         CON    7
_eight         CON    8
_nine          CON    9
_ten           CON    10
_eleven        CON    11
_twelve        CON    12
_thirteen      CON    13
_fourteen      CON    14
_fifteen       CON    15
_sixteen       CON    16
_seventeen     CON    17
_eighteen      CON    18
_nineteen      CON    19
_twenty        CON    20
_thirty        CON    21
_forty         CON    22
_fifty         CON    23
_sixty         CON    24
_seventy       CON    25
_eighty        CON    26
_ninety        CON    27
_hundred       CON    28
_thousand      CON    29
_million       CON    30
_billion       CON    31

```

Column #74: Sounding Off! Again - Part 1

_2000	CON	32	' "two thousand"
_am	CON	33	
_pm	CON	34	
_a	CON	35	
_alpha	CON	36	
_b	CON	37	
_bravo	CON	38	
_c	CON	39	
_charlie	CON	40	
_d	CON	41	
_delta	CON	42	
_e	CON	43	
_echo	CON	44	
_f	CON	45	
_fox	CON	46	
_g	CON	47	
_golf	CON	48	
_h	CON	49	
_hotel	CON	50	
_i	CON	51	
_india	CON	52	
_j	CON	53	
_juliet	CON	54	
_k	CON	55	
_kilo	CON	56	
_l	CON	57	
_lima	CON	58	
_m	CON	59	
_mike	CON	60	
_n	CON	61	
_november1	CON	62	
_o	CON	63	
_oscar	CON	64	
_p	CON	65	
_papa	CON	66	
_q	CON	67	
_quebec	CON	68	
_r	CON	69	
_romeo	CON	70	
_s	CON	71	
_sierra	CON	72	
_t	CON	73	
_tango	CON	74	
_u	CON	75	
_uniform	CON	76	
_v	CON	77	
_victor	CON	78	
_w	CON	79	
_whiskey	CON	80	
_x	CON	81	
_xray	CON	82	

Column #74: Sounding Off! Again - Part 1

_y	CON	83
_yankee	CON	84
_z	CON	85
_zulu	CON	86
_monday	CON	87
_tuesday	CON	88
_wednesday	CON	89
_thursday	CON	90
_friday	CON	91
_saturday	CON	92
_sunday	CON	93
_january	CON	94
_february	CON	95
_march	CON	96
_april	CON	97
_may	CON	98
_june	CON	99
_july	CON	100
_august	CON	101
_september	CON	102
_october	CON	103
_november2	CON	104
_december	CON	105
_amp	CON	106
_amps	CON	107
_bars	CON	108
_bit	CON	109
_bits	CON	110
_black	CON	111
_blue	CON	112
_brown	CON	113
_byte	CON	114
_bytes	CON	115
_cable	CON	116
_celcius	CON	117
_centimeter	CON	118
_centimeters	CON	119
_cents	CON	120
_chip	CON	121
_connector	CON	122
_cubic	CON	123
_degrees	CON	124
_divided_by	CON	125
_dollars	CON	126
_down	CON	127
_equals	CON	128
_fahrenheit	CON	129
_farads	CON	130
_feet	CON	131
_foot	CON	132
_gallons	CON	133

Column #74: Sounding Off! Again - Part 1

_gigahertz	CON	134	
_go	CON	135	
_gold	CON	136	
_good_aft	CON	137	' "good afternoon"
_good_morn	CON	138	' "good morning"
_goodbye	CON	139	
_gram	CON	140	
_grams	CON	141	
_gray	CON	142	
_green	CON	143	
_hello	CON	144	
_hertz	CON	145	
_hour	CON	146	
_hours	CON	147	
_inch	CON	148	
_inches	CON	149	
_indoor_tmp	CON	150	' "indoor temperature"
_is	CON	151	
_key	CON	152	
_kilobit	CON	153	
_kilobits	CON	154	
_kilobyte	CON	155	
_kilohm	CON	156	
_kilometer	CON	157	
_kilometers	CON	158	
_kiloohms	CON	159	
_light	CON	160	
_liters	CON	161	
_mega	CON	162	
_megabytes	CON	163	
_megaohms	CON	164	
_megohms	CON	165	
_meter	CON	166	
_meters	CON	167	
_micro	CON	168	
_micron	CON	169	
_microns	CON	170	
_midnight	CON	171	
_mile	CON	172	
_miles	CON	173	
_milli	CON	174	
_minus	CON	175	
_minute	CON	176	
_minutes	CON	177	
_noon	CON	178	
_oclock	CON	179	
_of_Merc	CON	180	' "of Mercury"
_of_Water	CON	181	
_ohms	CON	182	
_orange	CON	183	
_pascals	CON	184	

Column #74: Sounding Off! Again - Part 1

_per_hour	CON	185	
_per	CON	186	
_percent	CON	187	
_pico	CON	188	
_pink	CON	189	
_plus	CON	190	
_point	CON	191	
_pound	CON	192	
_pounds	CON	193	
_psi	CON	194	
_purple	CON	195	
_rpm	CON	196	
_red	CON	197	
_second	CON	198	
_seconds	CON	199	
_silver	CON	200	
_square	CON	201	
_start	CON	202	
_stop	CON	203	
_switch	CON	204	
_tan	CON	205	
_temp_is	CON	206	' "temperature is"
_t_cur_t_is	CON	207	' "the current time is"
_t_currnt	CON	208	' "the current"
_t_outside	CON	209	' "the outside"
_t_speed_is	CON	210	' "the speed is"
_the	CON	211	
_thee	CON	212	
_times	CON	213	
_up	CON	214	
_volt	CON	215	
_volts	CON	216	
_white	CON	217	
_wire	CON	218	
_yard	CON	219	
_yards	CON	220	
_yellow	CON	221	
_y_spd_is	CON	222	' "your speed is"
_the_date	CON	223	
_is_more	CON	224	
_pls_wait	CON	225	' "please wait"
_please	CON	226	
_is_less	CON	227	
_thank_you	CON	228	
_than	CON	229	
_and	CON	230	
_are_closed	CON	231	
_are_down	CON	232	
_are_off	CON	233	
_are_on	CON	234	
_are_open	CON	235	

Column #74: Sounding Off! Again - Part 1

```

_are_up      CON      236
_is_closed   CON      237
_is_down     CON      238
_is_off      CON      239

' ----[ Variables ]-----
'
phrase       VAR      Byte      ' phrase to say
command      VAR      Byte      ' command to QV306
addr         VAR      Word
result       VAR      Word
test         VAR      Nib
temp         VAR      Word

qvNumber     VAR      Word
qvDigit      VAR      Byte      ' 0..99

qvDay        VAR      Nib       ' 0 = Monday, 6 = Sunday
qvDate       VAR      Byte      ' 0..31
qvMonth      VAR      Nib       ' 1 = Jan, 12 = Dec
qvYear       VAR      Byte      ' 0..99
qvCentury    VAR      Bit       ' 0 = 19xx, 1 = 20xx

qvHours      VAR      Nib       ' 1..12
qvMinutes    VAR      Byte      ' 0..59
qvAMPM       VAR      Bit       ' 0 = PM, 1 = PM

qvVolume     VAR      Byte      ' 0..31

' ----[ EEPROM Data ]-----
'
Sentence1     DATA    _two,_gallons,_equals,_seven,_point,_six,_liters,$FF

' ----[ Initialization ]-----
'
Splash_Screen:
  DEBUG CLS
  DEBUG "QV306M4P Demonstration",CR
  DEBUG "-----",CR
  DEBUG "Initializing...",CR

Initialize:
  LOW QV_RST      ' reset the QV306M4
  PAUSE 100
  HIGH QV_RST
  PAUSE 2000      ' let QV306 count phrases

```

```

' ----[ Main Code ]-----
,
Show_Device:
  GOSUB QV_GetType
  DEBUG "Type/Rev: ",HEX2 result, "/"
  GOSUB QV_GetRevision
  DEBUG HEX2 result,CR,CR

Unsigned_Demo:
  DEBUG "Unsigned numbers: "
  FOR test = 0 TO 4
    LOOKUP test,[0,7,19,400,65535],qvNumber
    DEBUG DEC qvNumber," "
    GOSUB Say_Unsigned
    PAUSE 500
  NEXT
  DEBUG CR
  PAUSE 2000

Signed_Demo:
  DEBUG "Signed numbers: "
  FOR test = 0 TO 4
    LOOKUP test,[-8,-20,-258,-5000,-30207],qvNumber
    DEBUG SDEC qvNumber," "
    GOSUB Say_Signed
    PAUSE 500
  NEXT
  DEBUG CR
  PAUSE 2000

Time_Demo:
  qvHours = 10
  qvMinutes = 7
  qvAMPM = 0
  DEBUG "Time: ",DEC qvHours,":",DEC2 qvMinutes," "
  DEBUG "A"+(qvAMPM * 15),"M",CR
  GOSUB Say_Time12
  PAUSE 2000

Date_Demo:
  qvDay = 3
  qvDate = 29
  qvMonth = 3
  qvYear = 1
  qvCentury = 1
  DEBUG "Day: ",DEC qvDay,CR
  DEBUG "Date: M=",DEC qvMonth," D=",DEC qvDate
  DEBUG " Y=",DEC (19 + qvCentury),DEC2 qvYear,CR
  GOSUB Say_Day_And_Date
  PAUSE 2000

```

Column #74: Sounding Off! Again - Part 1

```
Sentence_Demo:
  DEBUG "Sentence: Two gallons equals 7.6 liters",CR
  addr = Sentence1
  GOSUB Say_Sentence
  PAUSE 2000

Volume_Demo:
  DEBUG "Volume demo"
  phrase = _hello
  FOR qvVolume = 5 TO 30 STEP 5
    GOSUB Set_Volume
    GOSUB Say_Phrase
    DEBUG "."
  NEXT
  PAUSE 2000

  DEBUG CR,CR,"Putting module to sleep",CR
  SEROUT QV_RX,T2400,[QV_Sleep]
  DEBUG "Done."

END

' ----[ Subroutines ]-----
'
' *****
' Get device information
' *****
'
QV_GetType:
  result = 0
  SEROUT QV_RX,T2400,[QV_Type]
  SERIN  QV_TX,T2400,2000,NoType,[result]
NoType:
  RETURN

QV_GetRevision:
  result = 0
  SEROUT QV_RX,T2400,[QV_Rev]
  SERIN  QV_TX,T2400,2000,NoRev,[result]
NoRev:
  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

' *****
' Say a sentence
' - "addr" holds starting EE address of sentence
' - speaks until phrase > 239
' *****
'
Say_Sentence:
    READ addr,phrase                      ' get a phrase from EEPROM
    IF (phrase > 239) THEN Done_Talking    ' $FF is EOM flag

QV306_Busy1:
    IF (QV_Busy = IsBusy) THEN QV306_Busy1  ' wait for Busy to release
    SEROUT QV_RX,T2400,[phrase]              ' say the phrase
    addr = addr + 1                         ' point to next phrase
    GOTO Say_Sentence                       ' say it

Done_Talking:
    RETURN

' *****
' Say an unsigned number
' *****
'
Say_Unsigned:
    temp = qvNumber

Say_Number:
    qvDigit = temp / 1000                   ' get thousands
    IF (qvDigit = 0) THEN Say_100s
    GOSUB Say_Value_XX
    phrase = _thousand
    GOSUB Say_Phrase

Say_100s:
    qvDigit = temp // 1000 / 100           ' get hundreds
    IF (qvDigit = 0) THEN Say_10s
    GOSUB Say_Val_0TO19
    phrase = _hundred
    GOSUB Say_Phrase

Say_10s:
    qvDigit = temp // 100                   ' get tens and ones
    IF (qvDigit = 0) & (qvNumber > 0) THEN Say_Num_Done
    GOSUB Say_Value_XX

Say_Num_Done:
    RETURN

```

Column #74: Sounding Off! Again - Part 1

```
' *****
' Say a signed number
' *****
'
Say_Signed:
  IF (qvNumber.Bit15 = 0) THEN Say_Unsigned
  phrase = _minus
  GOSUB Say_Phrase
  temp = ABS(qvNumber)
  GOTO Say_Number

' *****
' Say a 2-digit number
' *****
'
Say_Value_XX:
  IF (qvDigit < 20) THEN Say_Val_0TO19      ' skip if less than 20
  phrase = (qvDigit / 10) + _eighteen      ' calculate 10s message #
  GOSUB Say_Phrase                        ' say 10s
  qvDigit = qvDigit // 10                  ' calculate 1s message #
  IF (qvDigit = 0) THEN Say_Val_Done      ' skip if zero

Say_Val_0TO19:
  phrase = qvDigit
  pause 1000
  GOSUB Say_Phrase

Say_Val_Done:
  RETURN

' *****
' Say 12-hour time
' *****
'
Say_Time12:
  IF (qvHours < 1) OR (qvHours > 12) THEN Skip_Time
  IF (qvMinutes > 59) THEN Skip_Time

  phrase = _t_cur_t_is                    ' "The current time is..."
  GOSUB Say_Phrase

Say_Hours:
  qvDigit = qvHours
  GOSUB Say_Value_XX

Say_Oh:
  IF (qvMinutes > 9) THEN Say_Minutes
```

```

    phrase = _o                                ' as in 3:09 (three-oh-nine)
    GOSUB Say_Phrase

Say_Minutes:
    qvDigit = qvMinutes
    GOSUB Say_Value_XX

Say_AMPM:
    phrase = _am + qvAMPM
    GOSUB Say_Phrase

Skip_Time:
    RETURN

' *****
' Say the day and date
' - no range checking on date
' - 1900 = "19 hundred"
' - 190x = "19-oh-x"
' *****
'

Say_Day_And_Date:
    IF (qvDay > 6) THEN Say_Month                ' skip day if out-of-range
    phrase = qvDay + _monday
    GOSUB Say_Phrase
    PAUSE 250

Say_Month:
    phrase = _january + qvMonth - 1
    GOSUB Say_Phrase

Say_Date:
    qvDigit = qvDate                            ' 0..31
    GOSUB Say_Value_XX

Say_Year:
    IF (qvCentury = 1) THEN Year2K
    phrase = _nineteen
    GOSUB Say_Phrase

Check_Year00:
    IF (qvYear > 0) THEN Check_Year0X
    phrase = _hundred
    GOSUB Say_Phrase
    GOTO Year_Done

Check_Year0X:
    IF (qvYear > 9) THEN Say_YearXX
    phrase = _o                                ' say "oh"
    GOSUB Say_Phrase

```

Column #74: Sounding Off! Again - Part 1

```
GOTO Say_YearXX

Year2K:
  phrase = _2000
  GOSUB Say_Phrase

Say_YearXX:
  qvDigit = qvYear
  GOSUB Say_Value_XX

Year_Done:
  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,[qvVolume // 32]        ' volume range is 0..31
  RETURN
```