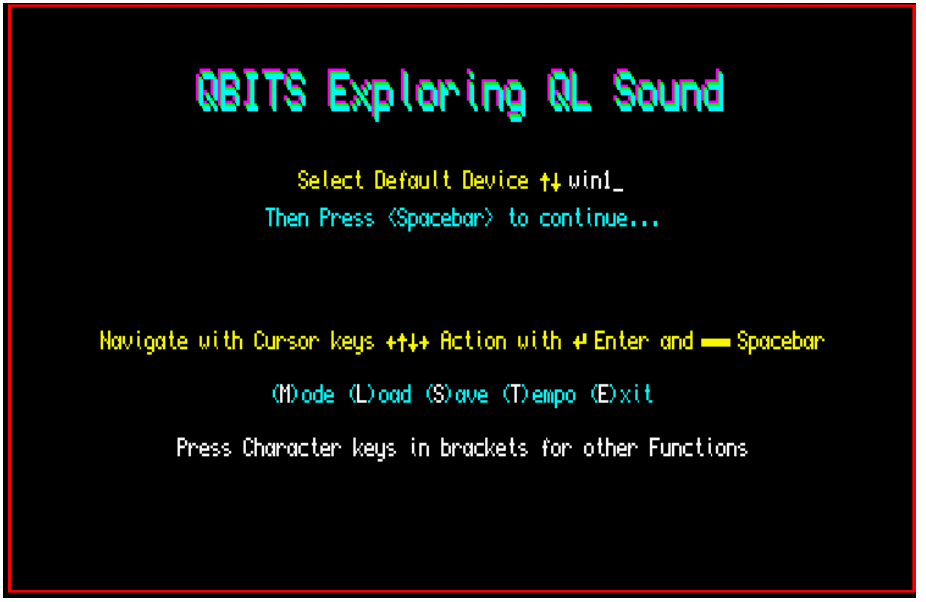
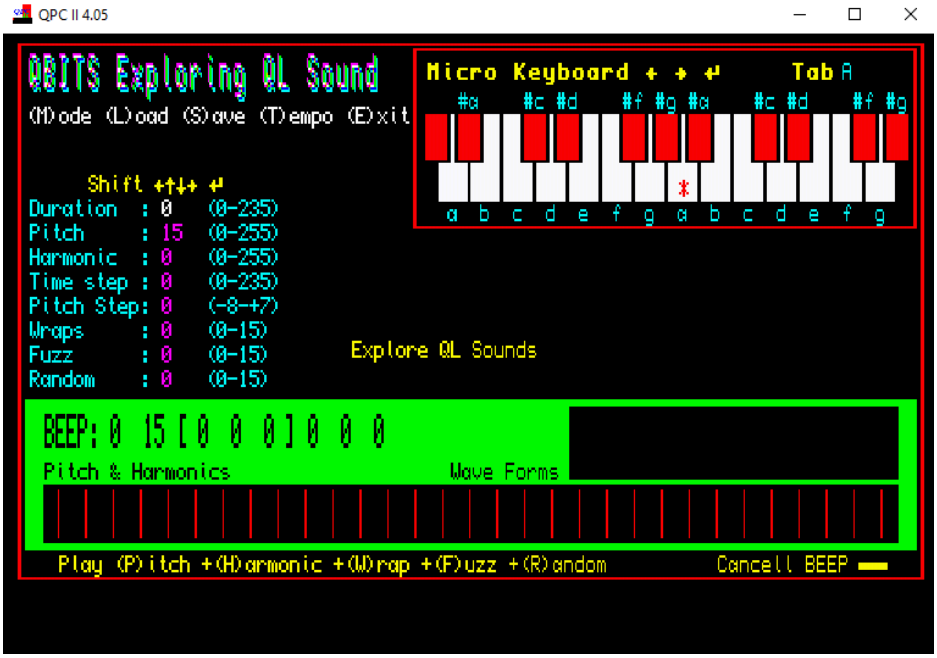
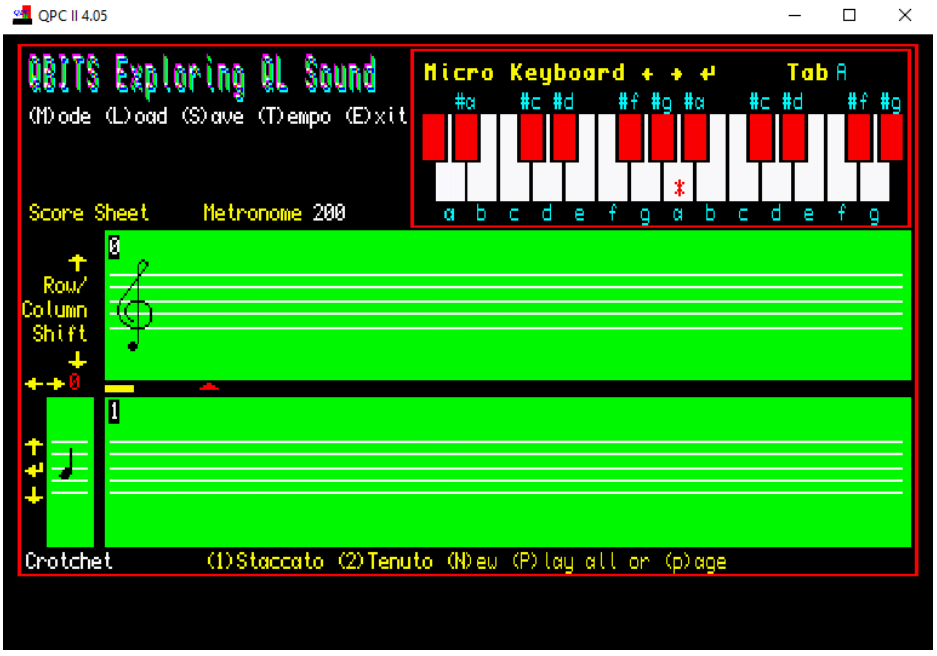
Sinclair QL Retro Computing



Sinclair QL Retro Computing


QPC II 4.05
QBITS Exploring QL Sound
(M)ode (L)oad (S)ave (T)empo (E)xit
Micro Keyboard ← → ↵ Tab A
#a #c #d #f #g #a #c #d #f #g
a b c d e f g a b c d e f g
Shift ←↑↓→ ↵
Duration : 0 (0-235)
Pitch : 15 (0-255)
Harmonic : 0 (0-255)
Time step : 0 (0-235)
Pitch Step: 0 (-8-+7)
Wraps : 0 (0-15)
Fuzz : 0 (0-15)
Random : 0 (0-15)
Explore QL Sounds
BEEP: 0 15 [0 0 0] 0 0 0
Pitch & Harmonics
Wave Forms
Play (P)itch +(H)armonic +(W)rap +(F)uzz +(R)andom
Cancell BEEP


QPC II 4.05
QBITS Exploring QL Sound
(M)ode (L)oad (S)ave (T)empo (E)xit
Micro Keyboard ← → ↵ Tab A
#a #c #d #f #g #a #c #d #f #g
a b c d e f g a b c d e f g
Score Sheet
Metronome 200
Row/
Column
Shift
Crotchet
(1)Staccato (2)Tenuto (N)ew (P)lay all or (p)age

QBITS Exploring QL Sounds

Introduction

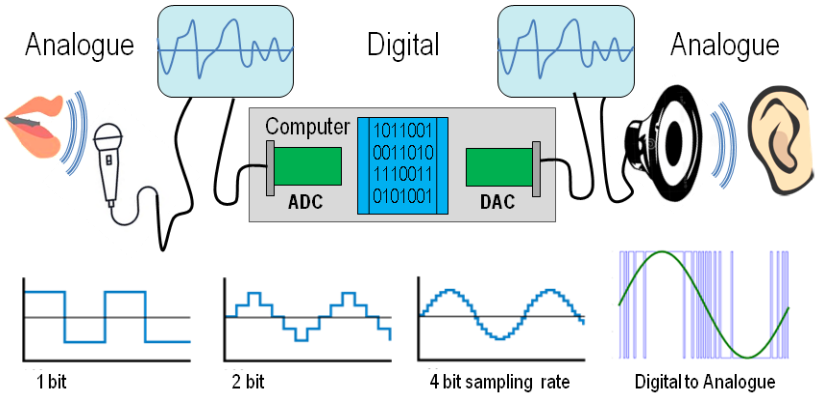
When tinkering in QL SuperBASIC to explore its Sound capabilities I have to consider a few facts. I don't profess in any way to be a skilled programmer although I have in the past as a project manager been involved with computer code development. As to my musical ability, it has been said by family and friends on numerous occasions, be it mostly in polite terms, I'm tone deaf (a misspent youth listening to loud rock maybe). All said and done my grandfather in his day played the piano, performed on stage with various song and dance routines and was a bit of an entrepreneur running a small troop of entertainers. Well entrepreneurial I'd like to think so, but regrettably I have to admit his musical talents were not passed on.

Aspirations

I therefore start with something of a disadvantage yet still curious to find out what could be achieved with the **QL SuperBASIC BEEP** command. Being able to add sounds to Games has been an aspiration that until now has been sidelined. The thud of a dart hitting the board or the whack as a golf ball is shot across the fairway, the background purr of an engine running. Those warning sounds of alarms and sirens associated with imminent danger or an emergency. Gun or laser weapons being fired with that accompanying sound of an explosion. Sounds of interest, musical notes or just quirky noises, whatever your thoughts these accompanying notes on **QBITS Exploring QL Sounds** begin by checking out the basic requirements, original QL hardware and concept behind the **BEEP** command.

Digital Audio

The human ear registers vibrations, sounds heard that are analogue in nature. In sound recording and reproduction systems, digital audio is the encoded representation of these sounds for processing, storage or transmission. The analogue wave length is sampled in regular time slots and the varying amplitude represented as a series of precise numbers. This allows editing and mixing to be introduced with special effects to simulate reverberation, enhancement of certain frequencies or change of pitch.

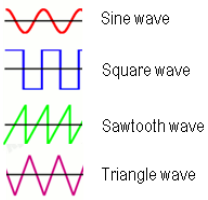


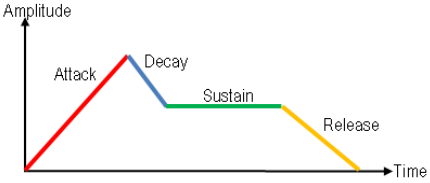
The minimum sample rate of any signal without introducing errors is twice the highest frequency present in the signal and is called the Nyquist frequency or Limit. A Digital representation can be expressed as a number of on/off's, high's & low's or in binary logic of 1's and 0's.

Sound Synthesizer

Electronic synthesizers use basic components that work together to reproduce a sound. Oscillators that generate the waveform and change of pitch, filtering that removes certain frequencies in the wave to change the timbre, an amplifier that controls the signal volume, and varying modulation to create effects.

A pure note or pitch will be in the form of a sine wave. Mixed with sympathetic vibrations enriches the tonal blend of a pitch creating differing waveforms. Timbre is perceived as the quality of these different sounds, the characteristic that represents the pre-conscious identity, based on information gained from the frequency transients, noisiness, unsteadiness, perceived pitch and spread of harmonics in a specific time frame. Really! Wow! Simplistically this means they make distinguishable the same pitch from being played on a piano as opposed to a violin. The main pitch is called the fundamental frequency and the related frequencies the harmonics. The wave envelope is the relationship of amplitude to time, this is called the pattern of Attack, Decay, Sustain and Release **ADSR**.





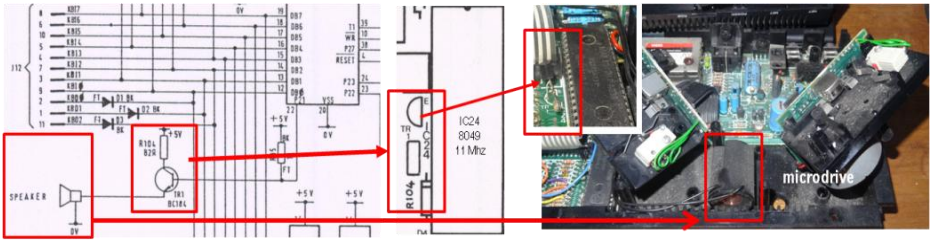
Summarising

A synthesizer needs to generate sound waves of different shapes. Supply more than one sound tone to produce a fundamental frequency and its harmonics. Make the volume of the sound change over time to produce different **ADSR** envelope shapes.

QL Sound Generation

For Sound the QL 68008 CPU (Central-Processor-Unit) communicates with a slave processor 8049 called the IPC (Intelligent-Peripheral-Controller). The **Intel 8049** co-processor chip used by the **Sinclair QL home computer** is designed to work alongside the custom chip ZX8302 ULA (Uncommitted Logic Array), acting as a keyboard buffer / joystick buffer, RS232 receive buffer and as a sound generator directing the output to the internal loudspeaker. The 8049 Chip contains a 2kx8 program memory, a 128x8 RAM memory, 27 I/O lines, 8-bit timer/counter in addition to on board oscillator and clock circuits.

Some QLs have an 8749 chip, which is the EPROM version of the Intel 8049. The 8049 can also be replaced by the Hermes Co-Processor manufactured by Tony Firshman, which provides improvements to sound, serial ports and further eliminates problems of keyboard bounce.

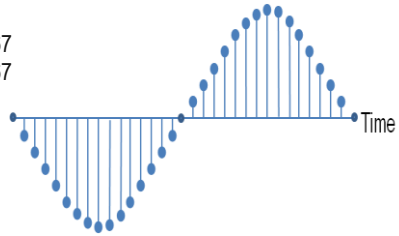


In executing a **SuperBASIC BEEP** instruction the IPC writes to port 2, P21 switching on/off transistor TR1. The emitter follower configuration is used as a voltage buffer amplifier to drive the 60 ohm 23mm loudspeaker situated between the two microdrives.

IPC Communication

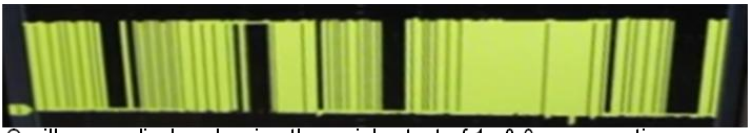
A command sent to the IPC uses the Manager Trap MT.IPCOM in the form of a header describing the command followed by any parameters. For audio output this is Initiate Sound, 8 parameters

8 bits	pitch_1	range 0 to 255
8 bits	pitch_2	range 0 to 255
16 bits	interval between steps	range -32768 to +32767
16 bits	duration	range -32768 to +32767
4 bits	step in pitch	range -8 to 7
4 bits	wrap	range 0 to 15
4 bits	randomness of step	range 0 to 15
4bits	fuzziness	range 0 to 15



Frequency and Amplitude

Two key features of a sound wave is the frequency (how many times the wave vibrates in one second) and is broadly related to the pitch of the sound we hear. The amplitude (volume) of a sound is related to the amount of energy that the sound wave carries. As the frequency increase the shorter the wave length, this is represented by the number of changes within a time frame. The higher the energy, the louder it sounds, the higher a waves amplitude. Digitally the amplitude is represented in binary 1s and 0s as a precise count or weight. A wave form is therefore generated by the number of ones in the binary record processed by the **DAC** on each cycle and related to the sample rate.



Oscilloscope display showing the serial output of 1s & 0s representing a wave

QL Sound Output

The QL Sound is produced by the changing number of switched 1s and 0s in each output cycle fed to transistor TRI via the IPC Port 2 Pin 21. Due to device inherent latency it is assumed this produces an output perhaps more recognisable as an analogue wave. The values to generate this digital to analogue conversion are derived from the BEEP parameters sent as part of the IPC instruction.

The one thing the IPC doesn't appear to have is any separate control over wave amplitude. This gives a partial explanation as to why higher pitches sound louder than their lower counterparts, the strings of 1s and 0s being closer together are going to add to the overall amplitude.

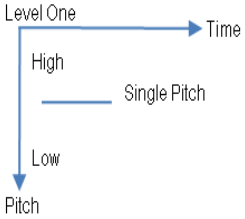
If four instead of just one of the IPC port 2 I/O had been use to further scale the voltage amplitude feeding TR1 Transistor, then an additional parameter could have been added to give 16 magnitudes of amplitude control over the resulting waveform.

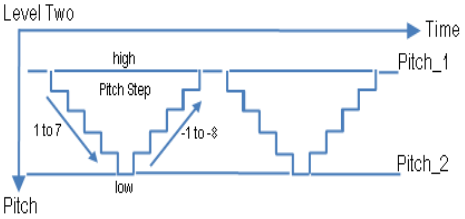
QL Sound Attributes

How does the QL Sound Generator fit the bill? It has two pitches and a method to ramp up and down between the two frequencies producing a range of harmonics. Adding harmonics can build the fundamental frequency from sinusoidal to more that of a squarewave. Then there's Wrap which I believe is intended to create an output similar to a sawtooth wave. There are also the parameters for fuzziness and randomness potentially further changing the output waveform.

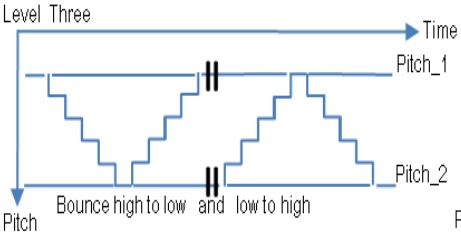
QL Sound Concepts
The QL Guide describes Sound being generated by the IPC (8049) as controlled by specifying a number of parameters, allowing the stage by stage build up of more complex sounds.

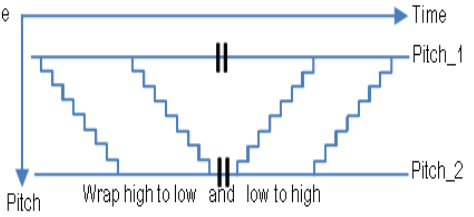
The first level is a single pitch active for a specified time, which is a pure sound at a set frequency. Once the **IPC 8049** has been instructed it will itself carry on for the specified duration. The **BEEP** command with a duration value of zero will run until a following **BEEP** command cancels it out or changes the parameters for a different sound. The duration is allegedly carried out in units of 72 microseconds the range being 1 to 32767 or again from -32768 to -1 (the duration for -1 or 32767 being 2.36 seconds).





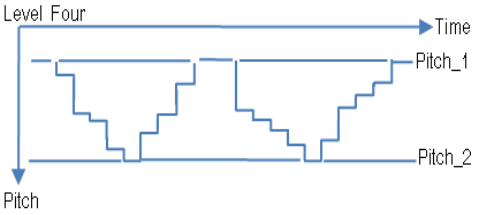
At level two a second pitch is added and the rate at which the sound ramps between the two pitches allegedly can produce semi musical beeps, spiralling or rippling tones, growls, zaps and moans. The number of steps and direction high to low or low to high can be configured.





The third level controls how the sound behaves after reaching one of the pitches. The sound is left to bounce or wrap a number of times. Depending on what step direction this can be high to low or visa verse.

Level four introduces a deviation from the specified step or gradient in moving between pitches with random larger or smaller steps. This random element can generate a wide and unexpected range of sounds.



Fuzzy is **level five** and is described as a further variation that adds changes to the pitch being generated and tends to make the sound more like a buzz.

Note: Directly being able to vary the amplitude of the sound output is not a parameter option.

QL SuperBASIC BEEP Command
The eight parameters listed in the QL User Guide are duration, pitch, pitch_2, grad_x, grad_y, wraps, fuzzy and random. They are grouped as duration + pitch, pitch_2 + grad_x + grade_y, wrap, fuzzy, random and used in different combinations and values to build complex sounds.

Duration
If the minimum time length used by the **IPC (8049)** processor is 1x72 microseconds, what would be the shortest multiple to perceive a sound? One factor is the pitch or frequency and the other the volume of amplitude. Once a sound wave reaches the human ear, the brain can perceive it in around 50 milliseconds. The stapes reflex is where the ear protects itself from very loud noises, here perception can be in as little as 25 milliseconds. In the relationship between hearing a sound and pitch perception this would be in the order of 100ms or slightly less for a higher pitch.

When a program has a number of **BEEP** Instructions to be carried out sequentially there is a concern that the sounds will be overwritten before being fully executed. The length of a sound being an important factor an alternative to the duration parameter is to use the **SuperBASIC PAUSE** command to control when to activate a following **BEEP** instruction to the **IPC 8049** processor. The **PAUSE** command uses multiples of 20 milliseconds for example:- **BEEP 0,3 : PAUSE 100 : BEEP** (will last two seconds).

Pitch
Hearing the sensation of a vibration is commonly referred to as pitch, which is the perceptual property of sounds and allows the ordering of their frequency to be judged as higher or lower. When only the **BEEP** duration and first pitch parameters are used it produces a single fundamental frequency. The **Pitch** range is 0 to 255 where the pitch climbs from its lowest at 255.

Harmonic
The second pitch (pitch_2), I will refer to as the **Harmonic,** add a **Time Interval** (grad_x) and a **Pitch Step** (grad_y), they create a sequence of sound variations ordered by the time duration and number of steps between the main pitch and second pitch. The Time Interval again is in multiple units of 72 microseconds for each note in the sequence. The Pitch Step range is -8 to 7 where step 1 to 7 scales downwards high to low pitch and -8 to 0 starts the sequence scaling upwards from low to a higher pitch. From then on the sequence bounces between the two pitches. A **Harmonic** without a **Time Interval** and/or **Pitch Step** has no affect. Adding a **Pitch** step of 1 when **Harmonic** and **Time Interval** are both 0 identifies the pitch as a high zero. **Harmonic** plus a **Pitch** step with **Time Interval** 0 just changes Main **Pitch** to the **Harmonic**.

Wraps
Wraps repeat the sequence of harmonics produced by the pitch_1, pitch_2, grad_x, grad_y parameters a number of times. Zero continues the bounce affect of the harmonic. Increasing values 1 to 7 creates scaling high to low for the number of Wraps. Scaling 8 to 15 creates Wraps from low to high.

Fuzzy & Random
Fuzzy decreases the purity of the pitch, Random just randomises the steps until little of the original sequence is evident. Both of these have a range 0 to 15, zero has no effect and the active range is more like 8 to 15. Increasing the fuzzy range as said before blurs the pitch to a buzz.

BEEPING
This **SuperBASIC function** detects if the QL hardware is producing a sound and simply returns as true or false. **IF BEEPING THEN BEEP**. If true this will cancel any QL Sound output.

QBITS QL Sound Output

As a starting point in coding for a QL Sound output, I added a few modifications to the **BEEP exerciser** Prog… from **QL SuperBASIC - The Definitive Handbook by Jan Jones**.

Variable	Parameter	Value	Description
d	duration	0 to 235	[0 increments of d*1e4/72]
p	pitch	0 to 255	[0 highest descending to 55]
h	harmonic	0 to 255	[0 highest descending to 255]
t	time interval	0 to 235	[0 increments of t*1e4/72]
s	pitch step	0 to 15	[effective range 1 to 7 low to high 8 to 15 high to low]
w	wrap	0 to 15	[effective range 1 to 15]
f	fuzz	0 to 15	[effective range 8 to 15]
r	random	0 to 15	[effective range 8 to 15]

```
100 REMark QLBeepv1   (QBITS Exploring QL Sounds 2018)

104 MODE 4:BInit : BMenu

108 DEFine PROCedure BInit
110 WINDOW 492,200,8,8:PAPER 7:INK 0:CSIZE 0,0:CLS
112 CURSOR 8,6:PRINT 'Play (P)itch +(H)armonic +(W)rap +(F)uzz +(R)andom'
114 CURSOR 8, 50:PRINT 'Duration      :       (0 to 235)'      :REMark d
116 CURSOR 8, 60:PRINT 'Pitch         :       (0 to 255)'      :REMark p
118 CURSOR 8, 70:PRINT 'Harmonic      :       (0 to 255)'      :REMark h
120 CURSOR 8, 80:PRINT 'Time Step     :       (0 to 235)'      :REMark t
122 CURSOR 8, 90:PRINT 'Pitch Step    :       (-8 to 7)'       :REMark s
124 CURSOR 8,100:PRINT 'Wraps         :       (0 to 15)'       :REMark w
126 CURSOR 8,110:PRINT 'Fuzz          :       (0 to 15)'       :REMark f
128 CURSOR 8,120:PRINT 'Random        :       (0 to 15)'       :REMark r
130 CURSOR 8,134:PRINT 'Edit  use  ←↑↓→  Space cancels BEEP <Esc> quit menu'
132 DIM BPm(7):INK 2:FOR ipm=0 TO 7:BRead
134 END DEFine

138 DEFine PROCedure BMenu
140 INK 0:ipm=0:BPrt
142 REPeat lp
144  CSIZE 0,1:CURSOR 8,24:PRINT 'BEEP: ';d;'  ';p;' [ ';h;'  ';t;'  ';s;' ] ';w;'  ';f;'  ';r;' ':CLS 4
146  k=CODE(INKEY$(-1))                    :REMark Read Keyboard
148  SELect ON k
150    =208:IF ipm>0:BChange -1            :REMark Up
152    =216:IF ipm<7:BChange 1             :REMark Down
154    =192:BPm(ipm)=BPm(ipm)-1:BRead      :REMark Left
156    =200:BPm(ipm)=BPm(ipm)+1:BRead      :REMark Right
158    = 80,112:BEEP d,p                   :REMark (P)itch
160    = 72,104:BEEP d,p,h,t,s             :REMark +(H)armonic
162    = 87,119:BEEP d,p,h,t,s,w           :REMark +(W)rap
164    = 70,102:BEEP d,p,h,t,s,w,f         :REMark +(F)uzz
166    = 82,114:BEEP d,p,h,t,s,w,f,r       :REMark +(R)andom
168    = 32:BEEP
170    = 27:BEEP:STOP
172  END SELect
174 END REPeat lp
176 END DEFine
```

```
180 DEFine PROCedure BPrt
182 CSIZE 0,0:CURSOR 80,50+ipm*10:PRINT BPm(ipm) TO 14:CSIZE 0,1
184 END DEFine

188 DEFine PROCedure BChange(change)
190 INK 4:BPrt
192 ipm=ipm+change
195 INK 7:BPrt
196 END DEFine

200 DEFine PROCedure BRead
202 BPrt: d=INT(BPm(0)*10000/72): t=INT(BPm(3)*10000/72)
204 p=BPm(1):h=BPm(2): s=BPm(4):w=BPm(5):f=BPm(6):r=BPm(7)
206 END DEFine
```

```
Play (P)itch +(H)armonic +(W)rap +(F)uzz +(R)andom

BEEP: 0 1[ 3 10000 7 ]0 0 0

Duration  :0          (0 to 235)
Pitch     :1          (0 TO 255)
Harmonic  :3          (0 TO 255)
Time Step :72         (0 to 235)
Pitch Step:7          (-8 to 7)
Warps     :0          (0 to 15)
Fuzz      :0          (0 to 15)
Random    :0          (0 to 15)
Edit use +↑↓+ Space cancels BEEP <Esc> to quit menu
```

To my untrained ear the above example gives a passable representation of a police panda car siren.

QBITS QL BEEP Parameters

Using **BEEP** parameters with the QL internal speaker arrangement, it has to be said, is more a trial and error process rather than any constructed methodology. However the program supplied here allows setting the various parameters and switching them on sequentially to hear the effects that take place.

Listening to Musical Notes

The next step was to look more deeply into pitch frequency and their harmonics. The frequency range of the human ear can be as low as 20 cycles per second or as high as 20,000 cycles per second (20Hz to 20kHz). The higher the frequency, the higher the pitch – double the frequency and the pitch goes an octave higher. For example 260Hz is approximately middle C on a piano keyboard 720Hz is C an octave higher, 4186Hz is the highest C8 and A0 the lowest key is 27.5Hz. The AC mains hum of 50Hz in Europe is close to the pitch of G1 = 48.99Hz.

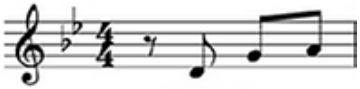
This whole process of relationship between frequencies and their harmonics, short and long time intervals of rising or falling pitches and the wave envelope they produce create tonal quality. In music this can be represented by symbols that allow a range of Notes and the way each is to be played. A Notes differing tone are dependent on the instrument being played. Therefore at this point we take a quick overview of musical representation, Stave, Clefs, Notes, their meaning and relationship.

Identifying a Music Note

In evaluating pitch in musical terms let's begin by identifying the notes and their representation, letters or symbols are used making it easier to write and quicker to read. Pitch classes are represented by letters of the alphabet (A,B,C,D,E,F,G) or more often by the naming convention Do-Re-Me-Fa-Sol-La-Ti.

The letter definition and corresponding notes are:

C –> do
D –> re
E –> mi
F –> fa
G –> so
A –> la
B –> ti (**H** in German)

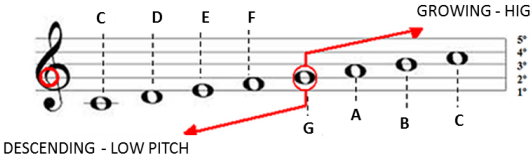


Measure (bar) Separator

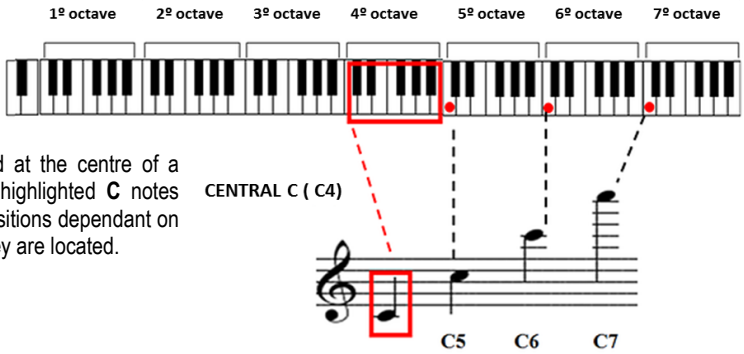
Sheet music registers the harmonic, rhythmic and melodic ideas. The Notes are positioned and written in the form of musical symbols.

Music Stave

The five lines (1st, 2nd, 3rd, 4th and 5th) of a **Stave** are where each line and each space between represent a different note of scale.



To read sheet music - is the sequence of notes, forwards and backwards!



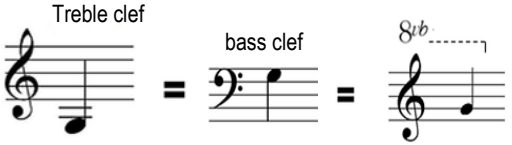
Middle **C (C4)** located at the centre of a piano keyboard. The highlighted **C** notes hold different stave positions dependant on the octave in which they are located.

Ledger Lines

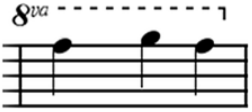
Where the Stave can't handle the representation of the notes for a full range of octaves**,** ledger lines are used. These lines are nothing more than the continuation of the Stave, they are used to represent notes that surpass the bottom or upper limits.

Treble Clef

Musicians throughout history have assigned different positions for their notes. **Clefs** were created as symbols serving to sign the note and the line of reference adopted. The most common Clef for guitar, piano and voice is the **Treble Clef** also known as the **G Clef** because the design of the Clef encircles the second Stave line which is G.



The symbol 8*v* is followed by the letter "*b*", which means "below", "8*va*" would be for octaves above.



Interpretation of the notes (F, G, F) should be played one octave above the position that it is in the Stave.

Accidentals

To show the increase or decrease of a notes pitch by one half step, symbols called Accidentals are used. When these same symbols appear at the very beginning of the music score they are specifying a key signature. They stay in effect for all of the notes of the same pitch for the rest of the measure.

Flats lower the pitch of the note by one half step.

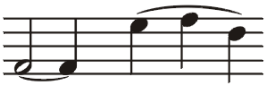
Sharps raise the pitch of the note by one half step.

Naturals cancel out any previous sharps or flats. The pitch returns to normal.

Articulations

These effect how the note is played and include the **slur**, phrase mark, staccato, staccatissimo, accent, sforzando, rinforzando, and legato.

Slurs smoothly connect notes of different pitches. This means to play the notes without breaks.

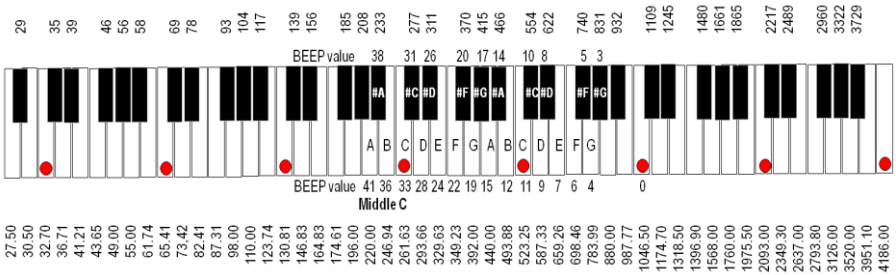


Ties connect notes of the same pitch, forming essentially one longer note.

Key Notes & BEEP values

At this point I thought it might be useful to review the range of piano keys and associated notes or pitches to their related frequencies. Then with a little help from a **QLUB** Prog and again with my untrained ear I cross referenced **BEEP** Pitch values around the middle **C** shown on my chart.

BEEP Pitch_1 = 0 to my untrained ear equates to a C6 or 1046.50Hz.

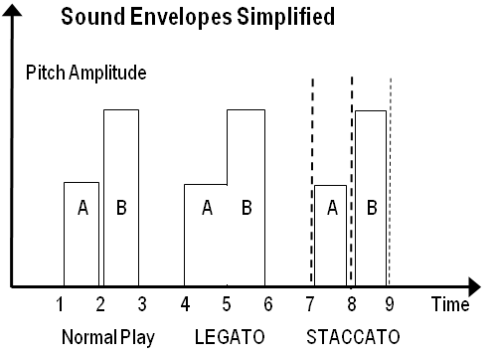


QLUB Music Micro Please!

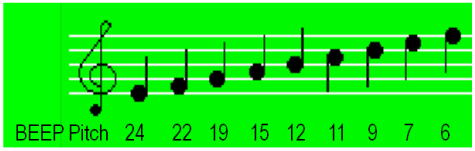
The **QLUB** edition of Mar/April 1985 carried an article with a short program which displayed to screen a **Stave** a **G-Clef** and added **crotchet** symbols to selected pitches. It described musical notes over two octaves that could be reproduced and gave the **BEEP pitch_1** numbered equivalents.

The article displayed the music symbols and their beat values from **Breve** to **hemi-demi-semi-quaver** (8 beats down to 1/16). Also drawn were Simplified **Sound envelopes** showing **Pitch / Amplitude** of Normal Playing time against **Legato** and **Staccato**.

```
100 REMark QLUBMicrov1     (QLUB Music Micro QBITS - 2018)

104 DIM pitch(18)
106 MODE 4:WINDOW 448,200,32,16:PAPER 4:CLS
108 WINDOW#0,448,20,32,216:PAPER#0,7:INK#0,0:CLS#0
110 PRINT#0,'auto or manual ? (a/m)'
112 IF INKEY$(-1)=='m' THEN yourself=1:ELSE yourself=0

116 REPeat loop
118  up=50:across=16:inc=0:CLS
120  Stave:Draw_Clef
122  FOR note=1 TO 18
124    Pick_Note yourself
126    Display_Note
128    pitch(note)=p
130    across=across+8
132  END FOR note
134  Play_Tune
136  CLS#0:PRINT#0,'Another tune ? (y/n)'
138  again$=INKEY$(-1)
140  IF again$=='n' THEN EXIT loop
142 END REPeat loop
144 STOP
```

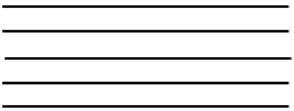



```
148 DEFine PROCedure Pick_Note(yourself)
150 IF yourself
152 REPeat check
154    CLS#0:INPUT#0, 'Note number (1 to 9)';choice
156  ELSE
158    choice=RND(1 TO 9)
160  END IF
162  SELect ON choice
164    =1:p=24:inc=0
166    =2:p=22:inc=1.5
168    =3:p=19:inc=3
170    =4:p=15:inc=4.5
172    =5:p=12:inc=6
174    =6:p=11:inc=7.5
176    =7:p= 9:inc=9
178    =8:p= 7:inc=10.5
180    =9:p= 6:inc=12
182    =REMAINDER :END REPeat check
184  END SELect
186 END DEFine
```



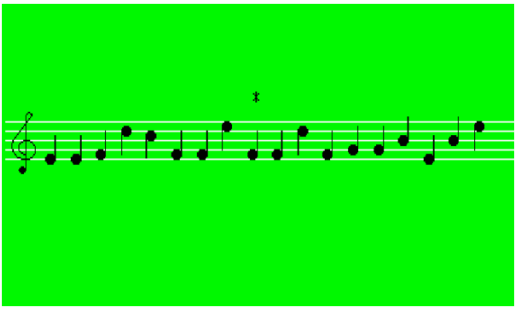
```
190 DEFine PROCedure Display_Note
192 FILL 1:CIRCLE across, up+inc,1.5:FILL 0
194 IF p<12
196   LINE across-1.5,up+inc TO across-1.5,up+inc-8
198 ELSE
200   LINE across+1.5,up+inc TO across+1.5,up+inc+8
202 END IF
204 BEEP-1,p:PAUSE#0:BEEP
206 END DEFine
```

```
210 DEFine PROCedure Stave
212 INK 7
214 FOR ledger=0 TO 12 STEP 3
216   LINE 2,up+ledger TO 165,up+ledger
218 END FOR ledger
220 INK 0
222 END DEFine
```



```
226 DEFine PROCedure Draw_Clef
228 LINE 8,up+1.5
230 ARC_R TO 0,4.5,-PI
232 ARC_R TO 0,-6,-PI TO -3,7,-3*PI/4
234 LINE_R TO 5,7:ARC_R TO -2,0,PI
236 LINE_R TO 0,-18
238 FILL 1:CIRCLE_R -1,0,1:FILL 0
240 END DEFine
```

```
244 DEFine PROCedure Play_Tune
246 CLS#0:PRINT#0,'Press any key to Play!'
248 PAUSE
250 x=116:y=72
252 FOR note=1 TO 18
254   Blip x,y
256   BEEP -1,pitch(note)
258   PAUSE 20
260   Blip x,y:x=note*8+16
262   BEEP
264 END FOR note
266 CLS#0:PRINT#0,'Play again(y/n)'
268 IF INKEY$(-1)=='y':Play_Tune
270 END DEFine
```



```
274 DEFine PROCedure Blip(x,y)
276 INK 4:OVER -1:CURSOR x,y,0,0 :PRINT '*' :OVER 0
278 END DEFine
```

The author of the **QLUB** publication was not given, however later that year 1985; the **QL User** magazine published **James Lucy's QL COMPOSER** which would appear to have some connection.

Bearing in mind the quizzical nature of the QL Sound generator it seemed logical to explore and retain sets of BEEP parameters, each key defined either as a musical note or everyday sound or even weird futuristic effects. To create a Score, with keyboard notes offering differing sounds it now required a selection of Symbols to identify timing and how they might be played.

QBITS Music Score
Having the **Stave** and **G-Clef,** next was determining a Time Signature or **Tempo**. Normal music has a regular pulse or rhythm identified as the **beat**. This is represented by two numbers written after the **Clef** at the beginning of a score to establish the number of beats in each uniformed section or measure. Provided are double **2/2, 2/4, 4/4,** and triple **3/4, 3/8, 6/8** timing options that can be used with the separator bar. The number on top tells you the beats in a measure; the number at the bottom is an indicator to the note combination for each **beat**. For example a **4/4** timing would be four beats to the metre and each beat represented by a Crotchet, but potentially other note combinations, two Minim or a single Semibreve.

QBITS Metronome
The **Beat** timing is calculated against the rate or number to be played per minute. The given range is from 30 to 240, in 10 beat steps. This is then used to calculate an individual **Note** or **Rest** value in determining the duration used with the **PAUSE** Command (*see below*).

For practical use with the **QL Sound System** and to provide a distinct separation between **Notes** or **Rests** from any previously played, a **delay PAUSE** is inserted. For normal playback **80 milliseconds** is chosen as a **standard break**. For **Staccato** (separated) this delay is increase to **120 milliseconds** to make the Note more distinctive and for a **Legato** (lengthened) reduced to **40 milliseconds** so it appears to merge with any following Note. As stated earlier to recognise differing pitches the Human ear requires around 100 milliseconds. Therefore the shortest duration for normal play based on above assumptions for a Semiquaver (a quarter Note) would need to be in the order of 180ms.

Calculating the timing for beats per minute (**bpm**), 60 seconds is multiplied by the **PAUSE** value for one second ie. 60*50=3000. This is then multiplied by the **Note** or **Rest** value (4 to 0.25). The result is then further divided by the Metronome rate. The PAUSE duration for a Crotchet with a max of 240bpm would be (60*50*1)/240=12.5, for a Semiquaver (60*50*0.25)/240=3.125. Clearly a duration **PAUSE** of 3 is only 60ms and not nearly enough time for the human ear to differentiate a change in pitch.

For the **ADSR** of a **Note's** playback in this proposed arrangement the **Attack, Decay, Sustain** is covered by the **BEEP** command with its attributes and set by the following **PAUSE** duration setting. This second cancelling **BEEP** followed by a further **PAUSE** delay creates the **Release** before any following Note.

Accents or **Articulations** explain how each **Note** is to be performed, **Staccato** with the note short and detached, **Tenuto** holding the Note for its full value blending into the next as in playing **Legato**. Another way to extending the duration and by half as much time again, is by placing an **Augmentation Dot** after the Note. For modes of play such as **Staccato** or **Tenuto**/**Legato** and the **Augmentation Dot** the **duration** and **delay** can be adjusted to reflect the change by increasing and decreasing their lengths.

BEEP *d,p,h,t,s,w,f,r.*:**PAUSE** *duration* : **BEEP** :**PAUSE** *delay*

Staccato marked by a dot placed above or below the Note head

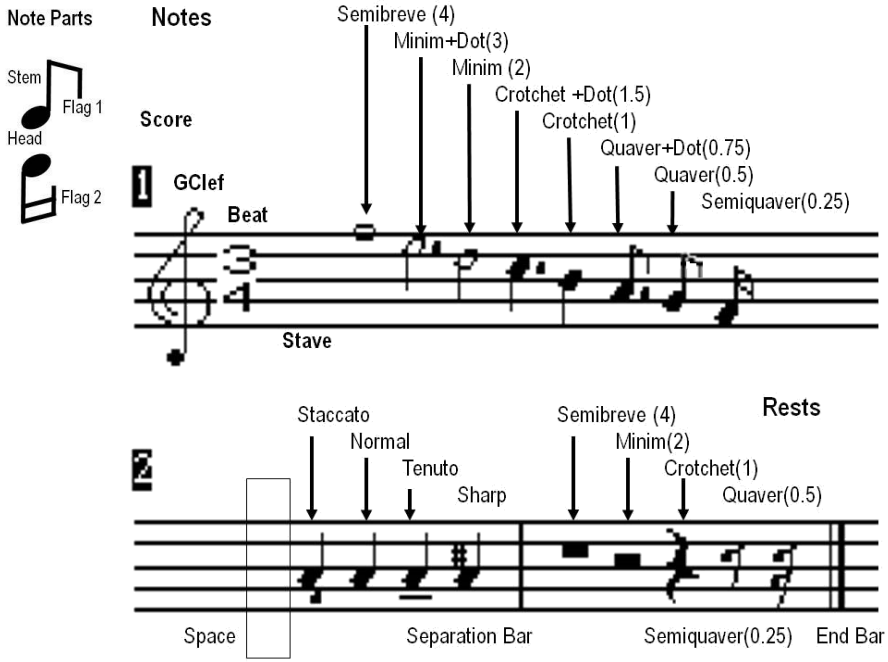
Tenuto marked by a line placed above or below the Note head.

Dot placed after the **note** adds half of the value of the **note** to itself.

As important are the **Rests** where there is no **BEEP** command just a **PAUSE** *duration + delay*. The **Space**, **Separation Bar** and **End Bar** are given a zero time.

QBITS Notes and Scores

The **QBITS Notes** chosen range is from the **Semibreve** down to the **Semiquaver** (a value of 4 beats down to 1/4 of a beat), this includes **Notes** with a **Dot Argumentation**. Then **Rests** to hold equivalent time slots where no music is played. **Sharps** that increase a note by half a pitch step and **Staccato** and **Tenuto** to further emphasise the length of how a note is to be played**.** Other symbols include a **Separation bar** for the measures or metre and an **End bar** to complete a musical score.



QBITS Musical Symbol Generation

Considerations to take into account are a **Note's** positions either below, between or on a **Stave line**, and if additional **ledgers** are required. The **Stave** and components that make up the Musical Symbols **Notes, Rests** etc. use **SuperBASIC** commands that operate with the **Graphics Coordinate System**. Each Symbol of a Score will therefore require progressive positioning along the Stave as well as vertical positioning relative to the scale of a Note being displayed.

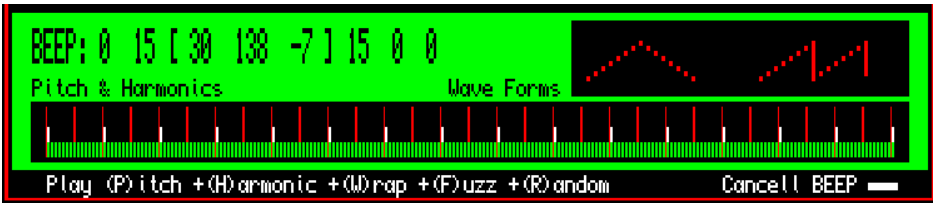
The first requirement is the **Space** which clears a position and redraws the **Stave** lines. The **Space** is a **FILL**ed rectangle drawn with the **LINE** command. Further use of the **LINE** command then displays the **Stave** lines. For example the combination of a **Space** and selected **Beat** value after the **GClef** allows the signature **Beat** to be changed or optionally to show **No Beat**.

The **Separation Bar, End Bar, Semibreve** and **Minim Rests** make further use of the **LINE** and **Fill** commands. The **Crotchet**, **Quaver** and **Semiquaver Rests** required a little more engineering. The actual **Notes** are built up from parts, **Head, Stem Tails** as shown above. **Accidentals #**Sharps, **Articulations, Staccato/Tenuto Dots - Lines** and following **Augmentation Dots** are added as required.

QBITS Menu Considerations
I decided on two (**M)odes** the **BEEP** where changes to the parameters could be explored and assigned as **Notes** or **Sounds** from which entries one could construct a written **Score** for replay. The usual and expected **(L)oad (S)ave** and **(E)xit.** One addition is the **(T)empo** which allows changes to the setting of the **Beat** and **Metronome** values.

'**m**' =0 **BEEP Mode** =1 **Score Mode**

QBITS BEEP Mode
I decided on some enhancement to the **BEEP Prog** given earlier, namely to provide some additional graphics to show how the differing parameters might have cause and effect on the waveform output. The bottom part of the display shows the **Pitch** and **Harmonics** and interspaced sub frequencies created by the **grad_X**, **grad_y** parameters (**QBITS time/step**) used with the **BEEP** command. Derived from the **QL Sound Concepts** this is also shown as the first wave. The second wave represents the assumed effects of the **Wrap** parameter. My hope in exploring the **BEEP** command parameters by listening and identifying their effects will lead to a more methodical way of constructing useful sound outputs.



QBITS Program Arrays & Variable Assignment
For the **BEEP Exploration** with use of the **Micro Keyboard** the option was to load an array so each **Key** is set with **BEEP parameter information**. These would then be available for use as **Notes** in constructing a **Score** for playback.

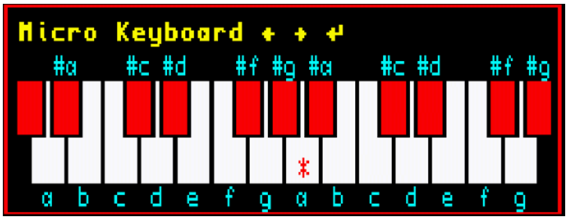
For the Micro Keyboard **Mkey(kg,kn,kp)**
kg 0-1 Micro key Groups **(Note: A/B Groups could be extended)**
kn 0-23 Micro key Number
kp 0-8 Micro Key Parameters **d,p,h,t,s,w,f,r, so**
d duration, **p** pitch, **h** harmonic, **t** time interval, **s** step, **w** warp, **f** fuzzy, **r** random, **so** Stave offset

For the Score Sheet **Score(sl,sn,sp)**
sl 0-9 Score Lines **(Note: 10 x 24 = 240 Symbols can be assigned)**
sn 0-23 Score Note position
sp 0-5 Score Parameters **kg,kn,ds,vn,as,ar**
kg Micro Keyboard Group (A/B) **, kn** Micro key number, **ds** display symbol (0-15),
vn value (4 to 0.25) of **Note** or **Rest** , **as** Spare, **ar** Articulations (**Staccato Tenuto/Legato**)

QBITS Coordinates
For **WINDOW x,y graphic coordination's** the conventions in pervious **QL Sound Progs** used variables **across** and **up**, so for the **Micro Keyboard** this became **ka, ku** and for the **Score Sheet sa, su**. For **Notes** dependant on keyboard location on the **Stave** the offset **so** is added to **su** for the displayed symbol.

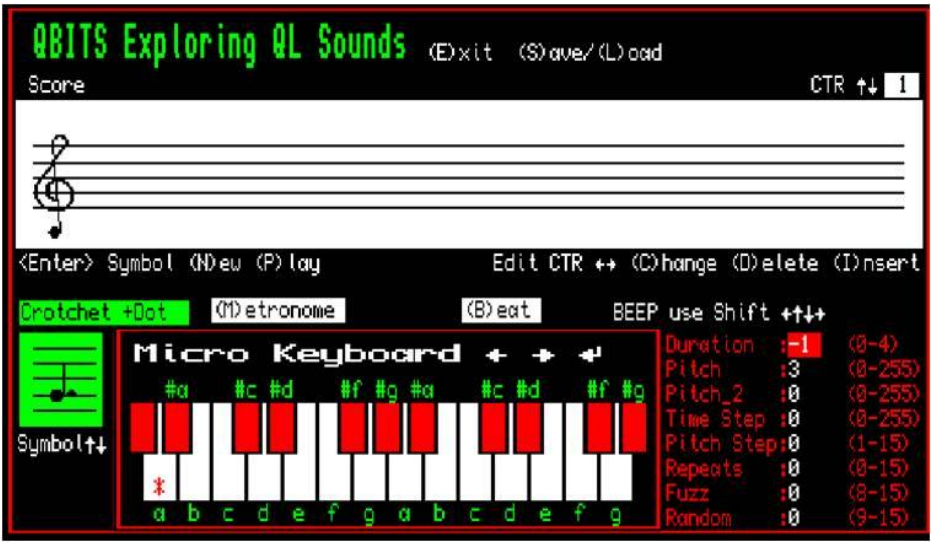
QBITS Micro Keyboard

Reviewing some past Progs aimed at using the **QL BEEP command** I hadn't seen any graphical representation of a keyboard. This display is based on the proposed key values given in the QLUB article described earlier.




```
100 REMark QBSMicrov1 (QBITS Micro Keyboard Graphics 2018)

102 MODE 4:CLS:Init_Keyboard:Select_Key

104 DEFine PROCedure Init_Keyboard
105 OPEN#3,scr_276x74a224x6 :PAPER#3,0:BORDER#3,1,2:CLS#3
106 FOR i=0 TO 13:BLOCK#3,16,36,12+i*18,26,7
107 FOR i=0 TO 14
108   IF i=2 OR i=5 OR i=9 OR i=12                    Note no action (Keys without upper keys)
109   ELSE
110     BLOCK#3,16,20,3+i*18,26,0
111     BLOCK#3,12,19,5+i*18,26,2
112   END IF
113 END FOR i
114 OVER#3,1:CSIZE#3,2,0:INK#3,6
115 FOR i=1 TO 2:CURSOR#3,4+i,4:PRINT#3,'Micro Keyboard ¼ ½'
116 OVER#3,0:CSIZE#3,0,0:INK#3,5
117 CURSOR#3,16,16:PRINT#3,' #a   #c #d   #f #g #a   #c #d   #f #g'
118 CURSOR#3,16,62:PRINT#3,'a  b  c  d  e  f  g  a  b  c  d  e  f  g'
119 END DEFine

121 DEFine PROCedure Select_Key
122 kp=0:ka=16:ku=54
123 OVER#3,-1:CURSOR#3,ka,ku:PRINT#3,'*':OVER#3,0
124 REPeat klp
125 k=CODE(INKEY$(-1))                    :REMark Read Keyboarded
126 SELect ON k
127   =192:IF kp> 0 :Change_Key -1        :REMark Left
128   =200:IF kp<23:Change_Key  1         :REMark Right
129   = 27:CLOSE#3:EXIT klp
130 END SELect
131 END REPeat klp
132 END DEFine

134 DEFine PROCedure Change_Key(change)
135 OVER#3,-1:CURSOR#3,ka,ku:PRINT#3,'*':OVER#3,0
136 kp=kp+change:ka=16
137 IF kp> 2:ka=26
138 IF kp> 7:ka=35
139 IF kp>14:ka=44
140 IF kp>19:ka=53
141 ka=ka+kp*9
142 SELect ON kp:=1,4,6,9,11,13,16,18,21,23:ku=36
143 SELect ON kp:=0,2,3,5,7,8,10,12,14,15,17,19,20,22:ku=52
144 OVER#3,-1:CURSOR#3,ka,ku:PRINT#3,'*':OVER#3,0
145 END DEFine
```

QBITS Exploring QL Sounds
Having identified the basics, **Micro Keyboard**, **BEEP parameters** with a **Graphics update**, the **Score Sheets, Stave, GClef** and the means to select from a range of **musical Symbols**, now was the time to bring this all together in a meaningful and workable display. Early attempts to cram this all in to one screen, left out the **BEEP** Graphics and gave only a small number of spaces to create a score.



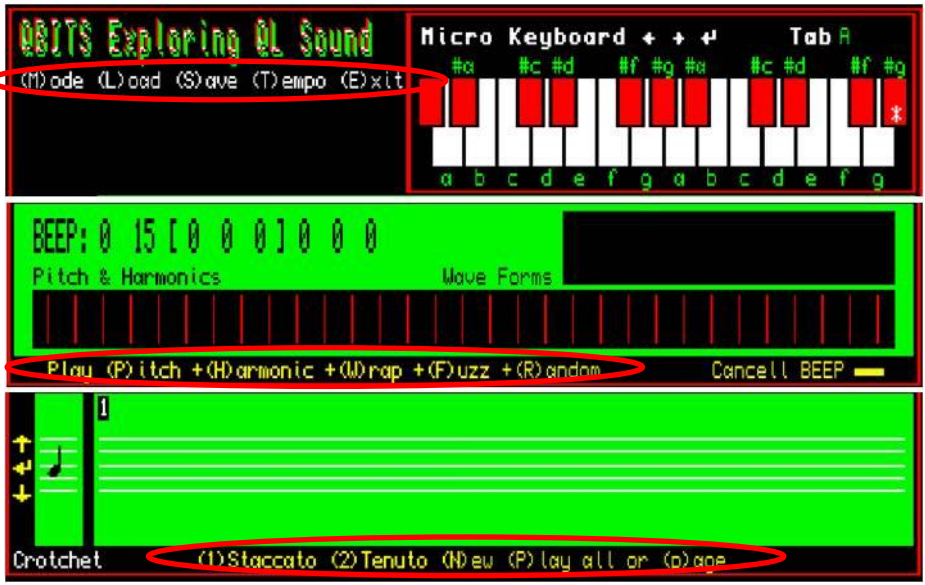
QBITS Layout Design
A review of what was required as mentioned earlier revealed the need for a two Mode approach, one for exploring the **BEEP** parameters and a second one for creating a Score sheet. (see inside Cover)

Providing a **Mode** change brought back the Graphics for exploring the effects of the changes to the **BEEP** parameters. In **Score sheet Mode** the redesign allowed two full lines to be displayed. Expanding a single Score line into multiple ones was not a problem, but displaying two with the ability to scroll up and down between them helps keep a sense of continuity. (see inside Cover)

QBITS Controls
The aim was for the design and layout to display Navigation using the Cursor keys and activation using the Spacebar and Enter key. Further Functions use single Character Keys identified by being within brackets. For example the main menu items (**M**)ode (**L**)oad (**S**)ave (**T**)empo (**E**)xit. For the different Modes bracketed Character Keys identify specific Mode related functions.

Having two Modes and trying to minimise the keys created problems or opportunities depending on your view point. For example the (**P**)lay command in the BEEP Mode returns the Pitch, in Score Sheet Mode (**P**) plays through all the Score Lines and (**p**) just the two current lines displayed on screen.

The Micro Keyboard in both display Modes provides a series of BEEP settings as Musical Notes, day to day sounds or strangely weird, atmospheric, ethereal or having an aesthetic quality. In BEEP Mode the parameters are up dated to the stored array entries. In Score sheet Mode the Musical Note being displayed will be shown located on the relative Stave bars position.



QBITS Exploring QL Sounds Data Files

The (**L**)oad and (**S**)ave allow a file Selection from **QBSDat_0** TO **_9**. The saved Data MKey(kl,kn,ka) is the array for BEEP parameters A/B two sets each holding 0 to 23 entries, then the Score(sl,sn,sa) array with 0 to 9 lines each holding from 0 to 23 entries (potentially values for 240 Notes).

QBITS Program Performance

At nearly 500 lines this should load on a standard QL be it slow to run. For performance a minimal tenfold step up in speed would make it reasonable. For example used with an emulator such as **QL2K** or **SMSQ** or **QPC2** running on an up to date hardware platform, the speed would be in the order of 1000 times faster. If you exit from the **QBITS** opening screen with **<Esc>** key as opposed to the **Spacebar** then changes to the **BEEP** display by use of the Micro keyboard or changes to the BEEP parameters will update the wave form displays automatically without having to press the enter Key.

QBITS Exploring QL Sounds

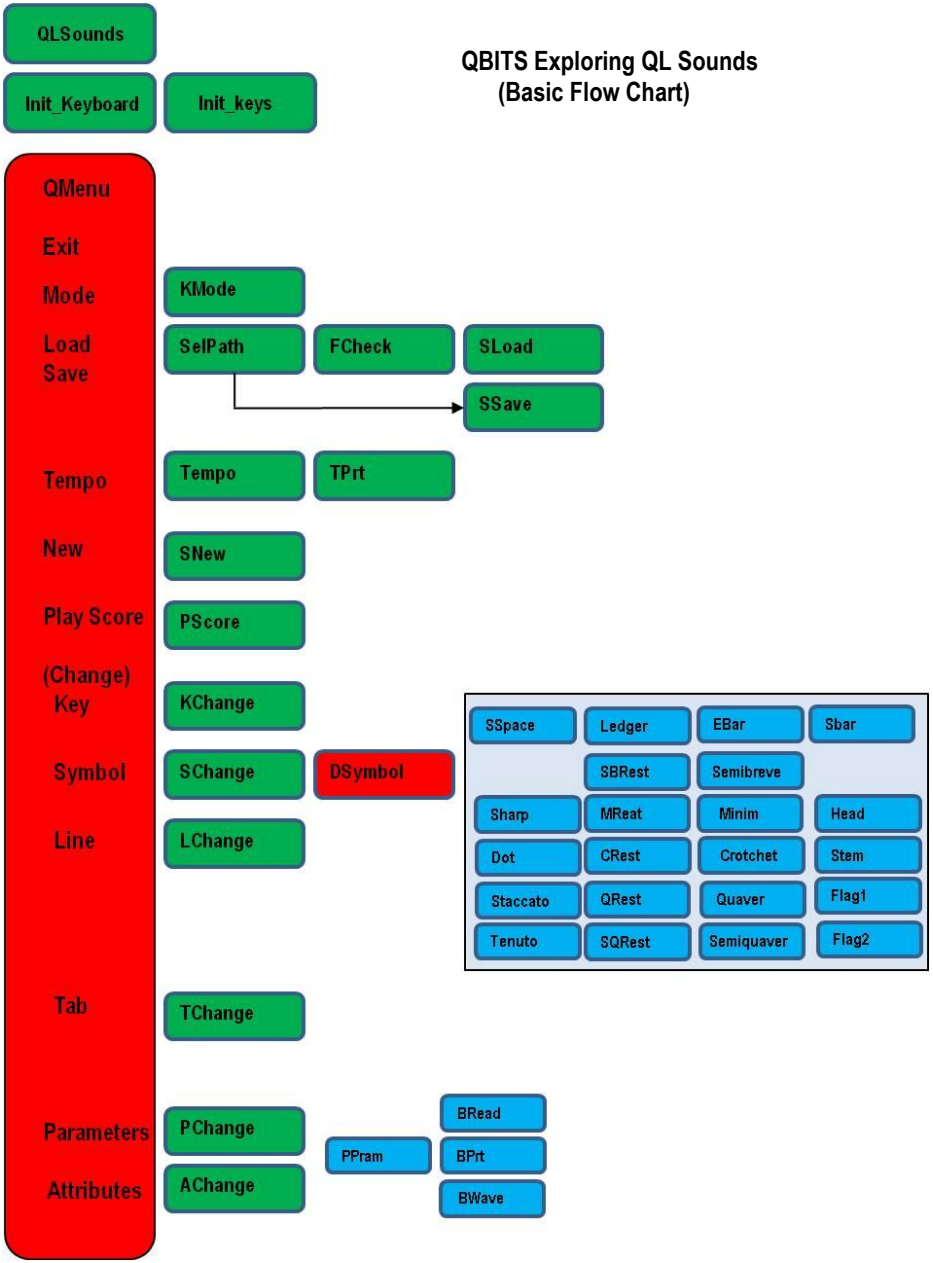
QBSBeepv1	BEEP Prog… (shown on pages 6/7)
QLUBMicrov1	QLUB magazine Music Micro Please (shown on pages 10/11)
QBSMicrov1	Micro keyboard Prog… (shown on page 14)
QBSoundsv3	QBITS Exploring QL Sounds Main Prog… (see page 20)
QBSDat_0	The first data file is supplied with Score samples. Load and enjoy!

QBITS Summary

Although the BEEP parameters hold the possibility for producing a large range of sounds and at a minimal level compatible musical Notes, it lacks the range of controls or mix of a basic electronic keyboard. However, I feel learning more about the QL BEEP command has been an experience worth pursuing and I've enjoyed tinkering with the code in putting together this Prog...

QBITS PROCedures

QLSounds		Init Windows, Welcome Instructions, Select default storage device	
Init_Keyboard		Setup Micro Keyboard display	
Init_keys		Seed Micro Key default configurations	
QMenu		Main Menu	
KMode		**m=0** 'Explore Sounds' : **m=1** 'Score Sheet'	
SLoad		Load from selected filename into array Key(,ln,a)	
SSave		Save to selected filename from array key(l,n,a)	
SelPath		Select device & Data file name **QBSDat_0 to 9**	
FCk		Filename check against **DIR** File List	
Tempo		Set Beat and Metronome values	
KChange(change)		Change key on Micro Keyboard range 0 to 23	
SChange(change)		Change Score Symbol	
SNew		Resets array Score (sl,sn,sa)=0	
NChange(change)		movement of Highlight marker for Score sheets	
SEnt		Enters Symbol on Score sheet and updates Score Array	
PScore		Plays Score (P) all or (p) page	
LChange(change)		Changes lines and actions Score symbol info displayed on page.	
DSymbol		Displays Symbols selected	
SSpace		Clears display position	
Ledger		Add ledger lines when required to extend Stave	
EBar		Draws End Bar	
SBar		Draws Separation bar	
Head, Stem, Hook1, **Hook2**		Draws Parts that make up a Note Symbol	
Semibreve	**SBRest**	Draws Symbol	Beat value of 4
Minim	**MRest**	Draws Symbol	Beat value of 2
Crochet	**CRest**	Draws Symbol	Beat value of 1
Quaver	**QRest**	Draws Symbol	Beat value of ½
Semiquaver	**SQRest**	Draws Symbol	Beat value of ¼
Sharp		Pitch raised by ½ pitch	
Dot		Beat played 1½ times normal length (Beat values 3,1½, ¾)	
Staccato		Notes played Pause = 8/10 of Beat (distinct)	
Tenuto		Notes played Pause = Whole Beat (lengthy)	
GClef		Draws the G-Clef	
Tempo		Select Beat & Metronome values	
TPrt		Screen displsy updates and Prints Tempo selection	
TChange		Toggles between KMkey arrays A & B	
PChange(change)		Selects parameters of BEEP	
PPram		Screen display updates and Prints selected parameter	
BRead		Reads Beep parameters d,p,h,,t,s,w,f,r	
AChange(change)		Updates BEEP attribute checks boundaries	
BPrt		Screen Display Prints BEEP attributes	
BWave		Calculates and updates BEEP wave Info and displays to screen	


QLSounds
Init_Keyboard
Init_keys
QBITS Exploring QL Sounds
(Basic Flow Chart)
QMenu
Exit
Mode
KMode
Load
Save
SelPath
FCheck
SLoad
SSave
Tempo
Tempo
TPrt
New
SNew
Play Score
PScore
(Change)
Key
KChange
Symbol
SChange
DSymbol
Line
LChange
SSpace
Ledger
EBar
Sbar
SBRest
Semibreve
Sharp
MReat
Minim
Head
Dot
CRest
Crotchet
Stem
Staccato
QRest
Quaver
Flag1
Tenuto
SQRest
Semiquaver
Flag2
Tab
TChange
Parameters
PChange
PPram
BRead
BPrt
BWave
Attributes
AChange

```
100 REMark QBSoundsv3            (QBITS Exploring QL Sounds 2018)

102 MODE 4:CLS:QLSounds:Init_Keys:QBITS_Menu

104 DEFine PROCedure QBITS_Menu
105 ac=0:ar=0:sl=0:sn=0:ds=12:bn=1:mn=200
106 mp=1:kl=0:kn=12:ka=143:ku=52:m=1:KMode
107 OVER#3,-1:CURSOR#3,ka,ku:PRINT#3,'*':OVER#3,0
108 :
109 REPeat Mlp
110 IF m=0 AND sx=1:BPrt:BWave
111 k=CODE(INKEY$(-1))
112  SELect ON k
113   =192: IF kn>0      :KChange -1       :BRead:SChange 0  :REMark Cursor Left
114   =200: IF kn<23     :KChange  1       :BRead:SChange 0  :REMark Cursor Right
115   =208: IF m=1 AND ds> 0               :SChange -1       :REMark Cursor Up
116   =216: IF m=1 AND ds<15               :SChange  1       :REMark Cursor Down
117   =196: IF m=0:AChange -1:ELSE         NChange mp,-1     :REMark Cursor Shift Left
118   =204: IF m=0:AChange  1:ELSE         NChange mp, 1     :REMark Cursor Shift Right
119   =212: IF m=0:PChange -1:ELSE         LChange -1        :REMark Cursor Shift Up
120   =220: IF m=0:PChange  1:ELSE         LChange  1        :REMark Cursor Shift Down
121   =  9:  IF kg=0:kg=1:else kg=0:END IF :TChange           :REMark Tab Toggle A/B Grp
122   = 10:  IF m=0::BPrt:BWave:ELSE       SEnt              :REMark Entr Sounds/Score
123   = 27:  BRead:BEEP d,p,h,t,s,w,f,r:PAUSE 50:BEEP        :REMark Test Sound
124   = 32:  IF m=0:BEEP:ELSE IF mp=1:mp=-1:ELSE mp=1:END IF :NChange mp,0    Note: Spacebar
125   =77,109:KMode                                         :REMark (M)ode BEEP/Score
126   =76,108:KLoad:IF m=1:LChange 0                        :REMark (L)oad
127   =83,115:KSave                                         :REMark (S)ave
128   =84,116:IF m=1:Tempo                                  :REMark (T)empo
129   =69,101:EXIT Mlp                                      :REMark (E)xit
130   =78,110:IF m=1:SNew:l=1                               :REMark (N)ew
131   =80,112:IF m=0:BEEP d,p:ELSE       PScore             :REMark (P)itch/(P)lay – (p)
132   =72,104:IF m=0:BEEP d,p,h,t,s                         :REMark +(H)armonic
133   =87,119:IF m=0:BEEP d,p,h,t,s,w                       :REMark +(W)raps
134   =70,102:IF m=0:BEEP d,p,h,t,s,w,f                     :REMark +(F)uzz
135   =82,114:IF m=0:BEEP d,p,h,t,s,w,f,r                   :REMark +(R)andom
136   =49:IF ar=0:ar=1:ELSE ar=0:END IF  :SChange 0
137   =50:IF ar=0:ar=2:ELSE ar=0:END IF  :SChange 0
138  END SELect
139 END REPeat Mlp
140 CLOSE#3:CLOSE#4:PAPER 0:CLS                           :REMark (E)xit
141 PRINT#0,'Bye...':STOP                                 :REMark NEW
142 END DEFine
```

Note: The Main Menu gives access to nearly all functions both in **BEEP Mode** when exploring the parameters or in **Score Sheet Mode** when building a piece of music. Select (**M**)ode and further actions by pressing Keys enclosed by brackets. Navigate by use of the Cursor keys with/without Shift, spacebar and enter keys

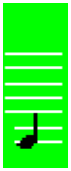
```
144 DEFine PROCedure KChange(change)          Note: Controls the Micro Keyboard Key Changes
145 OVER#3,-1:CURSOR#3,ka,ku:PRINT#3,'*':OVER#3,0
146 kn=kn+change:ka=16
147 IF kn> 2:ka=26
148 IF kn> 7:ka=35
149 IF kn>14:ka=44
150 IF kn>19:ka=53
151 ka=ka+kn*9:ar=0:SChange 0
152 SELect ON kn:=1,4,6,9,11,13,16,18,21,23:ku=36:ac=1
153 SELect ON kn:=0,2,3,5,7,8,10,12,14,15,17,19,20,22:ku=52
154 OVER#3,-1:CURSOR#3,ka,ku:PRINT#3,'*':OVER#3,0
155 IF m=0:INK 3:FOR kp=0 TO 7:PPram
156 IF m=0:kp=0:INK 7:PPram
157 END DEFine

159 DEFine PROCedure SChange(change)          Note: Controls the Selection of Symbol Changes
160 na=10:nu=18:IF m=0:RETurn
161 BLOCK 26,62,14,145,4:BLOCK 100,10,0,208,0:INK 7
162 FOR i=0 TO 12 STEP 3:LINE 7,18+i TO 15,18+i
163 ds=ds+change:DSymbol:INK 7:CURSOR 2,208:PRINT N$
164 END DEFine

166 DEFine PROCedure SNew
167 BLOCK 444,62,46, 76,4:FOR i=0 TO 12 STEP 3:LINE 20,56+i TO 198,56+i
168 BLOCK 444,62,46,145,4:FOR i=0 TO 12 STEP 3:LINE 20,18+i TO 198,18+i
169 FOR l=0 TO 9
170  FOR n=0 TO 23
171    FOR a=0 TO 4:Score(l,n,a)=0
172  END FOR a
173 END FOR l
174 GClef:l=0:CURSOR 48,78:PRINT l:CURSOR 48,146:PRINT l+1
175 sn=0:NChange 1,0:bn=1:mn=200:TPrt
176 END DEFine

178 DEFine PROCedure NChange(mp,change)          Note: Controls position of Note Change
179 sn=sn+change:IF sn<0 OR sn>23:sn=0
180 BLOCK 426,6,64,139,0:INK 2:ma=42+sn*6.5:mu=42
181 BLOCK 20,10,24,134,0:CURSOR 26,134:PRINT sn
182 FILL 1:LINE ma-2,mu TO ma,mu+mp TO ma+2,mu TO ma-2,mu:FILL 0:INK 7
183 END DEFine

185 DEFine PROCedure SEnt                    Note: Enters Selected Note /Symbol into Score
186 IF mp=1:nu=56:l=sl
187 IF mp=-1:nu=18:l=sl+1
188 na=42+sn*6.5:Score(l,sn,0)=kn:Score(l,sn,1)=ds
189 IF ar=1 OR ar=2:Score(l,sn,3)=ar :ELSE Score(l,sn,3)=0
190 DSymbol:Score(l,sn,4)=nv:ac=0:ar=0:SChange 0
191 sn=sn+1:IF sn>23:sn=23:NChange mp,0:ELSE NChange mp,0
192 END DEFine
```

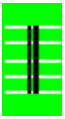
```
194 DEFine PROCedure PScore                    Note: Plays back Score
195 LOCal l:mp=1:nt=kn
196 IF k=112:lmin=sl:lmax=sl+1:ELSE lmin=1:lmax=9
197 FOR l=lmin TO lmax
198  IF l=lmax:mp=-1
199  FOR n=0 TO 23
200   ds=Score(l,n,1):IF ds<3:NEXT n
201   kn=Score(l,n,0)
202   nv=Score(l,n,4):dur=3000*nv/mn:sn=n:NChange mp,0:del=5
203   IF Score(l,n,3)=1:del=8:dur=dur-1
204   IF Score(l,n,3)>1:del=2:dur=dur+2
205   IF ds<8:PAUSE dur+del
206   IF ds>7:BRead:BEEP d,p,h,t,s,w,f,r:PAUSE dur:BEEP:PAUSE del
207  END FOR n
208 END FOR l
209 BEEP:kn=nt:sn=0:ds=12:SChange 0:mp=1:NChange 1,0
210 END DEFine
```

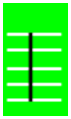
QBSDat_0 Demo file [the extract below from Beethoven's Ninth Symphony + Plus others Lines 2/3 4/5 6/7]

```
212 DEFine PROCedure LChange(change)           Note: Redraws Selected Score Lines
213 sl=sl+change:tn=kn
214 IF sl<0:sl=0
215 IF sl>8:sl=8
216 IF sl<9:CURSOR 48,78:PRINT sl:CURSOR 48,146:PRINT sl+1
217 FOR sn=0 TO 23
218   na=42+sn*6.5
219   nu=56:kn=Score(sl,sn,0):ds=Score(sl,sn,1):ar=Score(sl,sn,3):DSymbol
220   nu=18:kn=Score(sl+1,sn,0):ds=Score(sl+1,sn,1):ar=Score(sl+1,sn,3):DSymbol
221 END FOR sn
222 kn=tn:ds=12:sn=0:NChange 1,0:TPrt
223 END DEFine
```

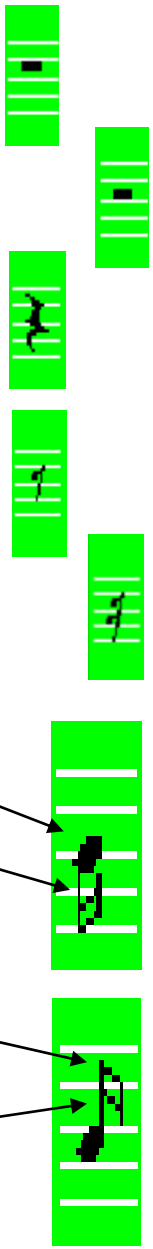
```
225 DEFine PROCedure DSymbol                    Note: Draws Symbol(s) Selected
226 SSpace:INK 0:ac=0
227 IF ds>7:SELect ON kn=1,4,6,9,11,13,16,18,21,23:ac=1
228 IF ds>7 AND MKey(0,kn,8)=-3   :lu=nu-3:Ledger
229 IF ds>7 AND MKey(0,kn,8)=-4.5:lu=nu-3:Ledger:lu=lu-3:Ledger
230 IF ds>7 AND MKey(0,kn,8)=-6   :lu=nu-3:Ledger:lu=lu-3:Ledger
231 IF ds>7:nu=nu+MKey(0,kn,8)
232 SELect ON ds
233  = 0:nv=0    :N$='Space'
234  = 1:nv=0    :EBar          :N$='End Bar'
235  = 2:nv=0    :SBar          :N$='Bar Seperator'
236  = 3:nv=4    :SBRest        :N$='Semibreve Rest'
237  = 4:nv=2    :MRest         :N$='Minim Rest'
238  = 5:nv=1    :CRest         :N$='Crotchet Rest'
239  = 6:nv=.5   :QRest         :N$='Quaver Rest'
240  = 7:nv=.25  :QRest:SRest   :N$='SemiQuaver Rest '
241  = 8:nv=4    :Semibreve     :N$='Semibreve'
242  = 9:nv=3    :Minim :Dot    :N$='Minim+Dot'
243  =10:nv=2    :Minim         :N$='Minim'
244  =11:nv=1.5 :Crotchet :Dot  :N$='Crotchet+Dot'
245  =12:nv=1    :Crotchet      :N$='Crotchet'
246  =13:nv=.75 :Quaver:Dot     :N$='Quaver+Dot'
247  =14:nv=.5   :Quaver        :N$='Quaver'
248  =15:nv=.25 :Semiquaver     :N$='Semiquaver'
249 END SELect
250 IF ac=1:Sharp:ac=0
251 IF ar=1:Staccato
252 IF ar=2:Tenuto
253 END DEFine

255 DEFine PROCedure SSpace
256 LOCal x,y,su:x=na-2.5:y=nu+22:INK 4
257 FILL 1:LINE x,y TO x+6.5,y TO x+6.5,y-34 TO x,y-34 TO x,y:FILL 0
258 INK 7:FOR su=0 TO 12 STEP 3:LINE x,nu+su TO x+7,nu+su
259 END DEFine

261 DEFine PROCedure Ledger
262 INK 7:LINE na-2,lu TO na+5,lu:INK 0
263 END DEFine

265 DEFine PROCedure EBar
266 SBar:FILL 1
267 LINE na+1,nu TO na+1,nu+12 TO na+1.6,nu+12 TO na+1.6,nu TO na+1,nu
268 FILL 0
269 END DEFine

271 DEFine PROCedure SBar
272 LINE na,nu TO na,nu+12.5
273 END DEFine
```



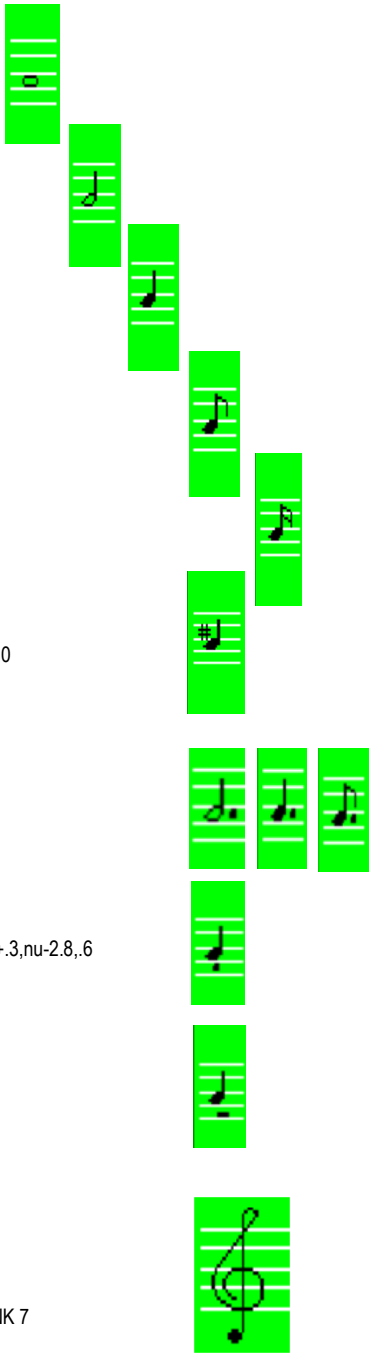




```
DEFine PROCedure SBRest
FILL 1:LINE na-1,nu+7.5
LINE_R TO 3,0 TO 0,1 TO -3,0 TO 0,-1:FILL 0
END DEFine

DEFine PROCedure MRest
FILL 1:LINE na-1,nu+6.2
LINE_R TO 3,0 TO 0,1 TO -3,0 TO 0,-1:FILL 0
END DEFine

DEFine PROCedure CRest
LINE na+2,nu+4.5:FILL 1
ARC_R TO -2,-3,3*PI/4 TO 1,4,-3*PI/4
LINE_R TO -2,1:ARC_R TO 0,4.5,3*PI/4
LINE_R TO 3,-2:ARC_R TO 1,-4.5,3*PI/4:FILL 0
END DEFine

DEFine PROCedure QRest
LINE na+.8,nu+3 TO na+2,nu+9:LINE_R TO -2,-2,
FILL 1:CIRCLE_R 0,.6,.6:FILL 0
END DEFine

DEFine PROCedure SRest
LINE na,nu TO na+1.5,nu+6 TO na-.5,nu+3.5
FILL 1:CIRCLE_R 0,.8,.6:FILL 0
END DEFine

DEFine PROCedure Head
CIRCLE na,nu,1.5,.6,-PI/4
END DEFine

DEFine PROCedure Stem
IF kn>13
 LINE na-1.1,nu-.5 TO na-1.1,nu-6
ELSE
 LINE na+1.2,nu+.5 TO na+1.2,nu+6
END IF
END DEFine

DEFine PROCedure Flag1
IF kn>13
 LINE_R TO 2,1.5  TO 0,2.5
ELSE
 LINE_R TO 2,-1.5 TO 0,-2.5
END IF
END DEFine

DEFine PROCedure Flag2
IF kn>13
  LINE_R TO 0,-1 TO -2,-1
ELSE
  LINE_R TO 0,1 TO -2,1
END IF
END DEFine
```



```
330 DEFine PROCedure Semibreve
331 CIRCLE na,nu,1.4,.7,PI/2
332 END DEFine

334 DEFine PROCedure Minim
335 Head:Stem
336 END DEFine

338 DEFine PROCedure Crotchet
339 FILL 1:Head:FILL 0:Stem
340 END DEFine

342 DEFine PROCedure Quaver
343 Crotchet na,nu:Flag1
344 END DEFine

346 DEFine PROCedure Semiquaver
347 Quaver na,nu:Flag2
348 END DEFine

350 DEFine PROCedure Sharp
351 OVER 1:CURSOR na-2.5,nu+5,0,0:PRINT '#':OVER 0
352 END DEFine

354 DEFine PROCedure Dot
355 FILL 1:CIRCLE na+2.5,nu,.6:FILL 0
356 END DEFine

358 DEFine PROCedure Staccato
359 INK 0:FILL 1
360 IF kn>13:CIRCLE na+1,nu+2.8,.6:ELSE CIRCLE na+.3,nu-2.8,.6
361 FILL 0:INK 7
362 END DEFine

364 DEFine PROCedure Tenuto
365 IF kn>13:LINE na,nu+4:ELSE LINE na,nu-3
366 INK 0:FILL 1
367 LINE_R TO 2,0 TO 0,-.5 TO -2,0 TO 0,.5
368 FILL 0:INK 7
369 END DEFine

371 DEFine PROCedure GClef
372 INK 0:LINE 25,57.5:ARC_R TO 1,4.5,-PI
373 ARC_R TO 0,-6,-PI TO -3,7,-3*PI/4
374 LINE_R TO 5,7:ARC_R TO -2,0,PI
375 LINE_R TO 0,-18:FILL 1:CIRCLE_R -1,0,.8:FILL 0:INK 7
376 END DEFine
```



```
378 DEFine PROCedure Tempo
379 CURSOR 160,52:PRINT 'Beat ⇦ ⇨'
380 CURSOR 188,64:PRINT '⇑⇓⇦':BLOCK 2,4,206,66,7
381 REPeat Tlp
382  k=CODE(INKEY$(-1))
383  SELect ON k
384    =192:IF bn>1:bn=bn -1:TPrt
385    =200:IF bn<7:bn=bn+1:TPrt
386    =208:IF mn<240:mn=mn+10:TPrt
387    =216:IF mn>  30:mn=mn -10:TPrt
388    = 10:EXIT Tlp
389  END SELect
390 END REPeat Tlp
391 BLOCK 48,10,160,52,0:BLOCK 24,10,188,64,0
392 END DEFine
```


```
394 DEFine PROCedure TPrt
395 CURSOR 160,64:PRINT mn;' '
396 SELect ON bn
397   =1:na=32:nu=56:SSpace:RETurn
398   =2:b1$='2':b2$='2'
399   =3:b1$='2':b2$='4'
400   =4:b1$='4':b2$='4'
401   =5:b1$='3':b2$='4'
402   =6:b1$='3':b2$='8'
403   =7:b1$='6':b2$='8'
404 END SELect
405 CSIZE 2,0:INK 0:STRIP 4
406 CURSOR 30,67,0,0:PRINT b1$:CURSOR 30,62,0,0:PRINT b2$
407 CSIZE 0,0:INK 7:STRIP 0
408 END DEFine
```


```
410 DEFine PROCedure TChange
411 IF kg=0:T$='A':ELSE T$='B'
412 CURSOR#3,232,4:PRINT#3,T$:KChange 0
413 END DEFine
```

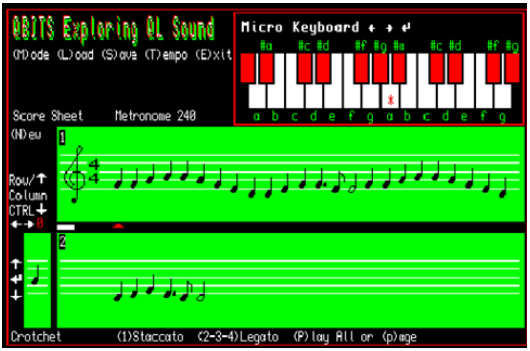
Note: Tab Change


```
415 DEFine PROCedure PChange(change)
416 INK 3:PPram
417 kp=kp+change:IF kp<0 OR kp>7:kp=0
418 INK 7:PPram
419 END DEFine

421 DEFine PROCedure PPram
422 CURSOR 76,62+kp*10:PRINT '  '
423 CURSOR 76,62+kp*10:PRINT MKey(kg,kn,kp) TO 14
424 END DEFine

426 DEFine PROCedure BRead
427 d= MKey(kg,kn,0):p= MKey(kg,kn,1):h= MKey(kg,kn,2)
428  t= MKey(kg,kn,3):s= MKey(kg,kn,4):w= MKey(kg,kn,5)
429  f= MKey(kg,kn,6):r= MKey(kg,kn,7)
430 END DEFine
```

Note: BEEP Parameter Change

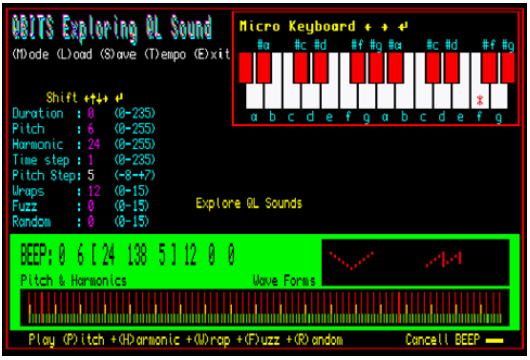
```
DEFine PROCedure AChange(change)                    Note: Attribute Change of BEEP Parameter values
MKey(kg,kn,kp)=MKey(kg,kn,kp)+change:BRead
IF d<0 OR d>235:d=0 :MKey(kg,kn,0)=d
IF p<0 OR p>255:p=0 :MKey(kg,kn,1)=p
IF h<0 OR h>255:h=0 :MKey(kg,kn,2)=h
IF t<0 OR t>235  :t=0  :MKey(kg,kn,3)=t
IF s<-8 OR s>7   :s=0 :MKey(kg,kn,4)=s
IF w<0 OR w>15 :w=0:MKey(kg,kn,5)=w
IF f<0 OR f>15   :f=0  :MKey(kg,kn,6)=f
IF r<0 OR r>15   :r=0  :MKey(kg,kn,7)=r
INK 7:PPram
END DEFine

DEFine PROCedure BPrt
CURSOR#4,8,4:BRead:d=INT(d*10000/72):t=INT(t*10000/72)
PRINT#4,'BEEP: ';d;'  ';p;' [ ';h;'  ';t;'  ';s;' ] ';w;'  ';f;'  ';r;' ':CLS#4,4
END DEFine

DEFine PROCedure BWave                              Note: Graphics to describe BEEP Waveforms
BLOCK 468,24,12,180,0:BLOCK 180,30,300,148,0:STRIP 4:INK 0
CURSOR 12,170:PRINT 'Pitch & Harmonics                Wave Forms'
FOR i=0 TO 452 STEP p:BLOCK 1,20,20+i,182,2
IF h>0 AND s<>0
   bs=SQRT(s*s):h2=INT((h-p)/bs):IF bs>h-p:h2=1
   FOR i=0 TO 450 STEP h :BLOCK 1,12,21+i,190,6
   FOR i=0 TO 450 STEP h2:BLOCK 1, 6,21+i,196,4
END IF
IF s>0
   bs=s:ba=308:bu=156
   FOR i=0 TO bs:BLOCK 2,2,ba+i*4,bu+i*2,2
   ba=308+bs*4:bu=156+bs*2
   FOR i=0 TO bs:BLOCK 2,2,ba+i*4,bu-i*2,2
END IF
IF s<0
   bs=SQRT(s*s):ba=308:bu=156+bs*2
   FOR i=0 TO bs:BLOCK 2,2,ba+i*4,bu-i*2,2
   ba=308+bs*4:bu=156
   FOR i=0 TO bs:BLOCK 2,2,ba+i*4,bu+i*2,2
END IF
IF w<=8 AND w>0
  bw=w:wa=bw*4:wu=156
  FOR i=0 TO bw:BLOCK 2,2,400+i*4,wu+i*2,2
  BLOCK 2,2*bw,400+bw*4,wu,2:BLOCK 2,2*bw,400+bw*8,wu,2
  FOR i=0 TO bw:BLOCK 2,2,400+wa+i*4,wu+i*2,2
END IF
IF w>8
   bw=w-8:wa=bw*4:wu=156+bw*2
   FOR i=0 TO bw:BLOCK 2,2,400+i*4,wu-i*2,2
   BLOCK 2,2*bw,400+bw*4,156,2:BLOCK 2,2*bw,400+bw*8,156,2
   FOR i=0 TO bw:BLOCK 2,2,400+wa+i*4,wu-i*2,2
END IF
STRIP 0:INK 6
END DEFine
```

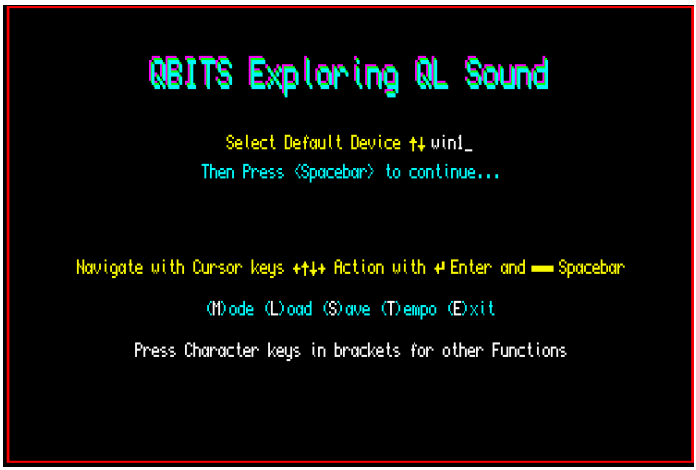
Note: Score Sheet Mode




```
486 DEFine PROCedure KMode
487 IF m=0
488    m=1:INK 7
489    BLOCK 212,166,0,50,0:BLOCK 444,62,46,76,4:BLOCK 444,62,46,145,4
490    FOR i=0 TO 12 STEP 3:LINE 20,56+i TO 198,56+i
491    FOR i=0 TO 12 STEP 3:LINE 20,18+i TO 198,18+i
492    INK 6:CURSOR 4,64:PRINT 'Score Sheet     Metronome ';mn;' '
493    CSIZE 2,0:CURSOR 0,80:PRINT ' ⇑'\\\\' ⇓'\'⇐ ⇒'
494    CURSOR 0,160:PRINT '⇑'\'⇐\'⇓':BLOCK 2,4,10,172,6
495    CSIZE 0,0:BLOCK 16,3,46,140,6:OVER -1
496    CURSOR 0,90:PRINT '  Row/'\'Column'\' Shift':OVER 0
497    CURSOR 100,208:PRINT '(1)Staccato (2)Tenuto (N)ew (P)lay all or (p)age':CLS 4
498    GClef:SChange 0:LChange 0:KChange 0
499 ELSE
500    BLOCK 212,24,0,52,0:BLOCK 490,142,0,76,0:BLOCK 488,62,2,145,4
501    INK 6:CURSOR 180,120:PRINT 'Explore QL Sounds'
502    CURSOR 36,52   :PRINT 'Shift ⇐⇑⇓⇒ ⇐ ':BLOCK 2,4,108,54,6:INK 5
503    CURSOR 4,  62  :PRINT 'Duration  :    (0-235)'  :REMark d
504    CURSOR 4,  72  :PRINT 'Pitch     :    (0-255)'  :REMark p
505    CURSOR 4,  82  :PRINT 'Harmonic  :    (0-255)'  :REMark h
506    CURSOR 4,  92  :PRINT 'Time step :    (0-235)'  :REMark t
507    CURSOR 4,102   :PRINT 'Pitch Step :   (-8-+7)'  :REMark s
508    CURSOR 4,112   :PRINT 'Wraps     :    (0-15)'   :REMark w
509    CURSOR 4,122   :PRINT 'Fuzz      :    (0-15)'   :REMark f
510    CURSOR 4,132   :PRINT 'Random    :    (0-15)'   :REMark r
511    INK 6:CURSOR 380,208:PRINT 'Cancell BEEP':BLOCK 16,3,458,212,6
512    CURSOR 20,208:PRINT 'Play (P)itch +(H)armonic +(W)rap +(F)uzz +(R)andom'
513    m=0:BRead:KChange 0:BPrt:BWave
514 END IF
515 END DEFine
```

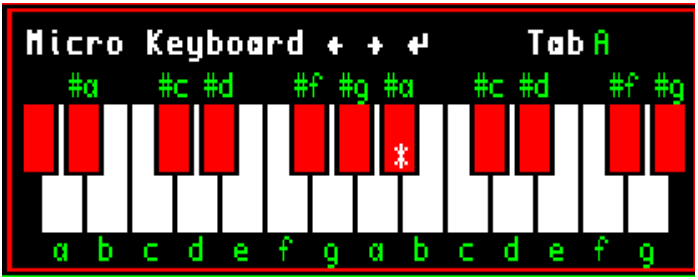
Note: BEEP Mode






```
517 DEFine PROCedure QLSounds
518 OPEN#5,scr_512x256a0x0:PAPER#5,0:CLS#5
519 OPEN#4,scr_284x24a14x150:PAPER#4,4:INK#4,0:CSIZE#4,0,1
520 WINDOW#2,496,220,8,4:PAPER#2,0:INK#2,7:CSIZE#2, 0,0:CLS#2
521 WINDOW#1,496,220,8,4:PAPER 0:BORDER 1,2
522 WINDOW#0,496,30,8,224:PAPER#0,0:INK#0,7:CSIZE#0,0,0
523 DIM Dv$(8,5):sx=0:RESTORE 524:FOR dn=1 TO 8:READ Dv$(dn)
524 DATA 'flp1_','flp2_','win1_','win2_','dos1_','dos2_','nfa1_','nfa2_'
525 CSIZE 2,1:OVER 1
526 FOR i=3 TO 5:INK i:CURSOR 96+i,20+i-2:PRINT 'QBITS Exploring QL Sound'
527 OVER 0:CSIZE 0,0
528 INK 6:CURSOR 156,60:PRINT 'Select Default Device ⇅:dn=3
529 INK 5:CURSOR 138,74:PRINT 'Then Press <Spacebar> to continue...'
530 INK 6:CURSOR 48,120
531 PRINT 'Navigate with Cursor keys ⇦⇧⇩⇨ Action with ⏎ Enter and ▭ Spacebar'
532 BLOCK 2,4,312,122,6:BLOCK 16,3,377,124,6:INK 5
533 INK 5:CURSOR 140,140:PRINT '( )ode ( )oad ( )ave ( )empo ( )xit':OVER 1
534 INK 7:CURSOR 140,140:PRINT ' M     L     S     T     E':OVER 0
535 CURSOR 90,160:PRINT 'Press Character keys in brackets for other Functions'
536 REPeat dlp
537  CURSOR 304,60:PRINT Dv$(dn)
538  k=CODE(INKEY$(5))
539  SELect ON k
540   =208:IF dn<8:dn=dn+1
541   =216:IF dn>1:dn=dn-1
542   = 32:BLOCK 260,30,120,60,0:EXIT dlp
543   = 27:BLOCK 260,30,120,60,0:sx=1:EXIT dlp          Note: Use <Esc> for higher speed QL Platforms
544  END SELect
545 END REPeat dlp
546 device_filename$=' ':CLS
547 CSIZE 1,1:OVER -1
548 FOR i=3 TO 5:INK i:CURSOR i,i-2:PRINT 'QBITS Exploring QL Sound'
549 Init_Keyboard:CSIZE 0,0:OVER 0:SCALE 120,0,0:INK 7
550 CURSOR 2,24:PRINT '(M)ode (L)oad (S)ave (T)empo (E)xit'
551 END DEFine
```

```
553 DEFine PROCedure Init_Keyboard
554 OPEN#3,scr_276x74a224x6 :PAPER#3,0:BORDER#3,1,2:CLS#3
555 FOR i=0 TO 13:BLOCK#3,16,36,12+i*18,26,7
556 FOR i=0 TO 14
557   IF i=2 OR i=5 OR i=9 OR i=12
558   ELSE
559     BLOCK#3,16,20,3+i*18,26,0
560     BLOCK#3,12,19,5+i*18,26,2
561   END IF
562 END FOR i
563 OVER#3,1:CSIZE#3,1,0:INK#3,6
564 FOR i=1 TO 2:CURSOR#3,4+i,4:PRINT#3,'Micro Keyboard ⇦ ⇨ ⏎      Tab'
565 OVER#3,0:CSIZE#3,0,0:INK#3,5:BLOCK#3,2,4,164,6,6
566 CURSOR#3,16,16:PRINT#3,' #a    #c #d    #f #g #a    #c #d    #f #g'
567 CURSOR#3,16,62:PRINT#3,'a  b  c  d  e  f  g  a  b  c  d  e  f  g'
568 END DEFine
```



The **Micro Keyboard** WINDOW now includes the active **Group Array** as **A or B,** which is toggled between by using the **Tab** key.

As with other **QBITS** programs **Load** & **Save** use **SelPath** Procedure to choose from a list of allocated Data filenames and **FCheck** to search the default File **DIR**ectroy returning **NOT found** if undetected.


```
570 DEFine PROCedure SelPath
571 INK 7:file=0:SD$='QBSDat_'
572 REPeat FSel
573   CURSOR 12,36:PRINT 'Select: ';Dv$(dn)&SD$&file;' ⇧⇩ ⏎ (Esc)'
574   BLOCK 2,4,162,38,7:k=CODE(INKEY$(5))
575   SELect ON k
576     =216:IF file>0:file=file-1
577     =208:IF file<9:file=file+1
578     = 10:ck=1:BLOCK 200,10,0,36,0:EXIT FSel
579     = 27:ck=0:BLOCK 200,10,0,36,0:RETurn
580   END SELect
581 END REPeat FSel
582 name$=Dv$(dn)&SD$&file:Gf$=SD$&file
583 END DEFine
```

```
585 DEFine PROCedure FCheck
586 IF ck=0:RETurn
587 BLOCK 200,10,0,36,0:CURSOR 12,36:PRINT 'Searching...'
588 PAUSE 20:DELETE Dv$(dn)&'FList'
589 OPEN_NEW#99,Dv$(dn)&'FList':DIR#99,Dv$(dn):CLOSE#99
590 OPEN_IN#99,Dv$(dn)&'FList'
591 REPeat dir_lp
592  IF EOF(#99)
593    CLOSE#99:CURSOR 12,36:PRINT 'File Not Found...'
594    PAUSE 25:BLOCK 200,10,0,36,0:file=0:ck=0:EXIT dir_lp
595  END IF
596  INPUT#99,fchk$:IF fchk$==Gf$:CLOSE#99:EXIT dir_lp
597 END REPeat dir_lp
598 END DEFine
```


```
600 DEFine PROCedure KLoad
601 SelPath:FCheck:IF ck=0:RETurn
602 CURSOR 12,36:PRINT 'Loading...'
603 OPEN_IN#99,name$
604 FOR kg=0 TO 1
605  FOR kn=0 TO 23
606    FOR kp=0 TO 8:INPUT#99,MKey(kg,kn,kp)
607  END FOR kn
608 END FOR kg
609 INPUT#99,mn\bn :kg=0:kn=12:kp=0:TChange
610 FOR sl=0 TO 9
611   CURSOR 66+sl*6,36:PRINT '.':PAUSE 5
612   FOR sn=0 TO 23
613     FOR sp=0 TO 4:INPUT#99,Score(sl,sn,sp)
614   END FOR sn
615 END FOR sl
616 CLOSE#99:sl=0:sn=0:sp=0:KChange 0:PAUSE 50:BLOCK 200,10,0,36,0
617 END DEFine
```


```
619 DEFine PROCedure KSave
620 SelPath:IF ck=0:RETurn
621 CURSOR 12,36:PRINT 'Saving...'
622 DELETE name$:OPEN_NEW#99,name$
623 FOR kg=0 TO 1
624  FOR kn=0 TO 23
625   FOR kp=0 TO 8:PRINT#99,MKey(kg,kn,kp)
626  END FOR kn
627 END FOR kg
628 PRINT#99,mn\bn :kg=0:kn=12:kp=0:TChange
629 FOR sl=0 TO 9
630   CURSOR 66+sl*6,36:PRINT '.':PAUSE 5
631   FOR sn=0 TO 23
632     FOR sp=0 TO 4:PRINT#99,Score(sl,sn,sp)
633   END FOR sn
634 END FOR sl
635 CLOSE#99:sl=0:sn=0:sp=0:KChange 0:PAUSE 50:BLOCK 200,10,0,36,0
636 END DEFine
```

```
638 DEFine PROCedure Init_Keys
639 REMark Mkey(kg,kn,kp)     =kg(0-1):=kn(0-23):=kp(0-8)
640 REMark Mkey(kg,kn,0 - 1)  =d duration : =p Pitch
641 REMark Mkey(kg,kn,2 - 4)  =h harmonic: =t time: =s Step
642 REMark Mkey(kg,kn,5 - 7)  =w Wrap: =f Fuzzy : =r random
643 REMark Mkey(kg,kn,8)      =so Stave Offset -6 to +13.5 S
644 REMark Score(sl,sn,sp)    =sl(0-9):=sn(0-23):=sp(0-4)
645 REMark Score(sl,sn,0)     =kn Note number(0-23)
646 REMark Score(sl,sn,1)     =ds display symbol =0 to 23
647 REMark Score(sl,sn,2)     Spare
648 REMark Score(sl,sn,3)     =ar Articulation 1=Staccato 2=Tenuto(Legato)
649 REMark Score(sl,sn,4)     =nv Note/Rest value =0 or 4,3,2,1.5,1,0.75,0.5,0.25
650 :
651 DIM MKey(1,23,8),Score(9,23,4)
652 RESTORE 653
653 DATA 41,38,36,33,31,28,26,24,22,20,19,17,15,14,12,11,10,9,8,7,6,5,4,3
654 DATA -6,-6,-4.5,-3,-3,-1.5,-1.5,0,1.5,1.5,3,3
655 DATA 4.5,4.5,6,7.5,7.5,9,9,10.5,12,12,13.5,13.5
656 FOR kn=0 TO 23
657   READ p:MKey(kg,kn,0)=0:MKey(0,kn,1)=p
658 END FOR kn
659 RESTORE 654:FOR kn=0 TO 23:READ so:MKey(0,kn,8)=so
660 DATA 0,82,164,1, 7, 9,0,0     :REMark kn=0
661 DATA 0,76,152,1,-6, 8,0,0     :REMark kn=1
662 DATA 0,72,144,1, 5, 7,0,0     :REMark kn=2
663 DATA 0,66,132,1,-5, 6,0,0     :REMark kn=3
664 DATA 0,62,124,1, 4, 5,0,0     :REMark kn=4
665 DATA 0,56,112,1,-4, 4,0,0     :REMark kn=5
666 DATA 0,52,104,1, 3, 3,0,0     :REMark kn=6
667 DATA 0,48,  96,1,-3, 2,0,0    :REMark kn=7
668 DATA 0,44,  88,1, 2, 1,0,0    :REMark kn=8
669 DATA 0,40,  80,1,-2, 2,0,0    :REMark kn=9
670 DATA 0,38,  76,1, 1, 3,0,0    :REMark kn=10
671 DATA 0,34,  68,1,-1, 4,0,0    :REMark kn=11
672 DATA 0,30,  60,1, 2, 5,0,0    :REMark kn=12
673 DATA 0,28,  56,1,-2, 6,0,0    :REMark kn=13
674 DATA 0,24,  48,1, 3, 7,0,0    :REMark kn=14
675 DATA 0,22,  44,1,-3, 8,0,0    :REMark kn=15
676 DATA 0,20,  40,1, 4, 9,0,0    :REMark kn=16
677 DATA 0,18,  36,1,-4,10,0,0    :REMark kn=17
678 DATA 0,16,  32,1, 5,11,0,0    :REMark kn=18
679 DATA 0,14,  28,1,-5,12,0,0    :REMark kn=19
680 DATA 0,12,  24,1, 6,13,0,0    :REMark kn=20
681 DATA 0,10,  20,1,-6,14,0,0    :REMark kn=21
682 DATA 0, 8,  16,1, 7,15,0,0    :REMark kn=22
683 DATA 0, 6,  12,1,-7,  7,0,0   :REMark kn=23
684 RESTORE 660
685 FOR kn=0 TO 23
686   READ d,p,h,t,s,w,f,r
687   MKey(1,kn,0)=d:MKey(1,kn,1)=p
688   MKey(1,kn,2)=h:MKey(1,kn,3)=t:MKey(1,kn,4)=s
689   MKey(1,kn,5)=w:MKey(1,kn,6)=f:MKey(1,kn,7)=r
690 END FOR kn
691 kg=0:kn=12:CURSOR#3,232,4:PRINT#3,'A'
692 END DEFine
```

Notes on BEEP Parameters
The DATA lines provided for the Micro Keyboard Tab B are experimental sounds. When running the program these can be overwritten as can the Tab A to create a set of personalised sounds.

For Sound construction, **duration** should be considered in steps of 200 milliseconds. For **pitch_1 & pitch_2, fundamental** and **harmonic,** the following pages identify the QL Sound range of frequencies. Typically make **pitch_2** a multiple of **pitch_1** then by increasing the **Time** and/or **Step** (**grad_x, grad_y**) alters the composite sound output by the number of interim frequencies. **Wrap** creates a changing pattern of rising or falling scale.

Note: Table for own use...

	Duration	Pitch	Harmonic	Time	Step	Wrap	Fuzzy	Random	Remark
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									

Pitch vs Frequency on the **Sinclair QL by Marq**

If the **QL BEEP** highest pitch 0 = 1313Hz and lowest pitch 255 is a frequency of 43Hz. Assuming that the formula is of the form *a/(x+c)*, we get approximately the following relationship between frequency (f) and pitch (p): f = 11336.256 / (p + 8.634) and p = 11336.256 / f - 8.634

Playable Notes and their BEEP value:

F1	251
F#1	236
G1	222
G#1	210
A1	197
A#1	187
B1	175
C2	165
C#2	155
D2	146
D#2	137
E2	129
F2	121
F#2	114
G2	107
G#2	101
A2	94
A#2	89
B2	83

(from here; off about 0.5 Hz max.)

C3	78
C#3	73
D3	69
D#3	64
E3	60
F3	56
F#3	53
G3	49
G#3	46
A3	43
A#3	40
B3	37

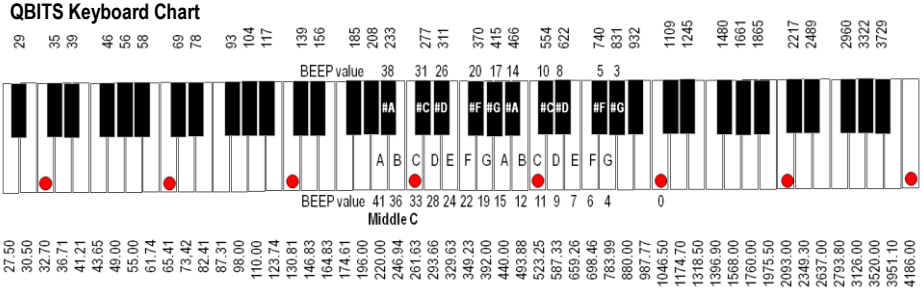
(deviations have grown to 1 Hz+)

C4	35	Middle C 261.63
C#4	32	
D4	30	
D#4	28	
E4	26	
F4	24	
F#4	22	
G4	20	
G#4	19	
A4	17	
A#4	16	
B4	14	

(at this point the notes have very little to do with the periods, but let's keep going for the sake of completeness...)

C5	13
C#5	12
D5	11
D#5	10
E5	9
F5	8
F#5	7
G5	6
G#5	5
A5	4
B5	3
C6	2
D6	1
E6	0

QBITS Keyboard Chart

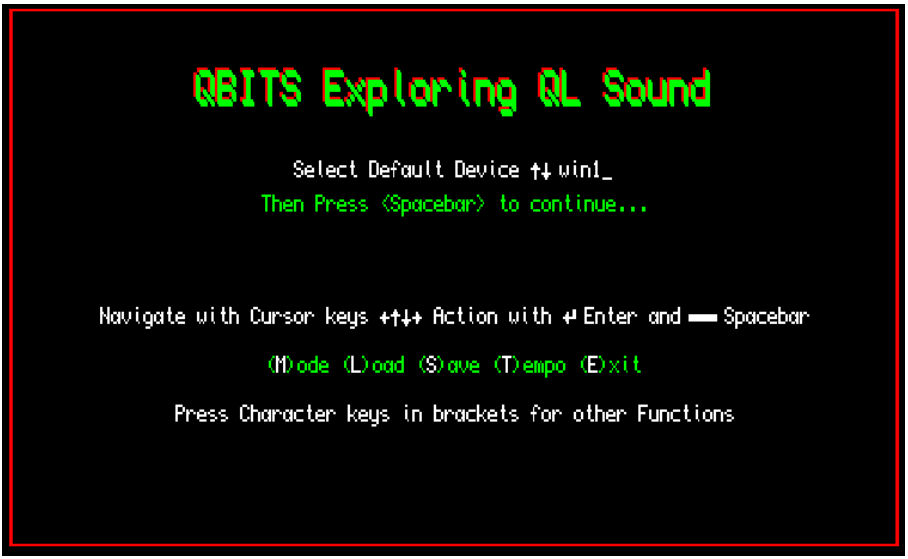


Beep Pitch frequencies supplied by Marq:

QBITS highlighted Octave Notes

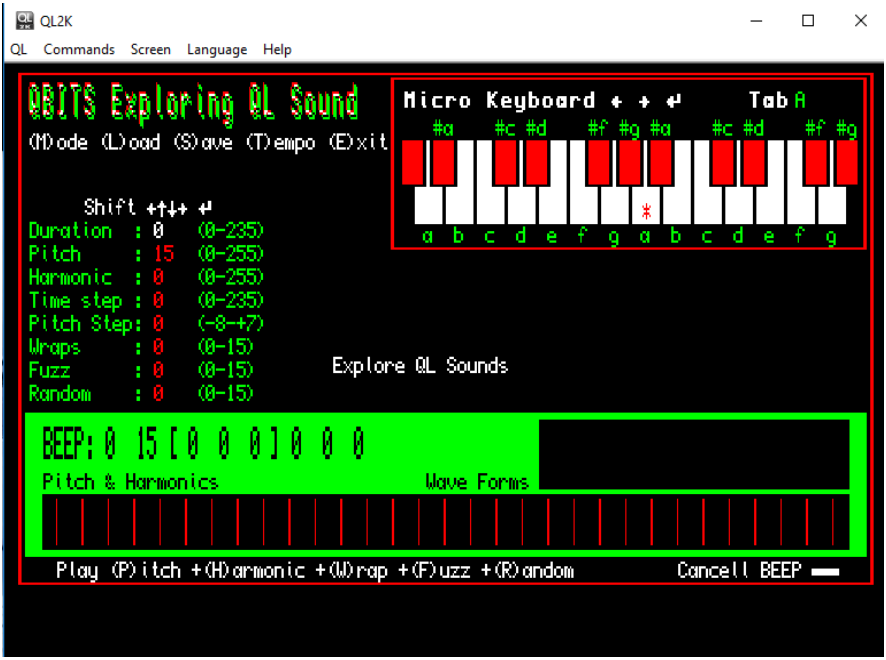
0	1313.00	46	**207.50 G3#**	92	112.65
1	1176.71	47	203.77	93	111.54
2	**1066.05 C6**	48	200.17	**94**	**110.45 A3**
3	974.42	49	**196.69 G3**	95	109.39
4	897.29	50	193.34	96	108.34
5	**831.48 G5#**	51	190.10	97	107.32
6	**774.66 G5**	52	186.96	98	106.31
7	**725.11 F5#**	53	183.93	99	105.32
8	**681.52 F5**	54	180.99	**100**	**104.35 G2#**
9	**642.87 E5**	55	178.15	101	103.40
10	**608.37 D5#**	56	175.39	102	102.47
11	**577.38 D5**	57	172.72	103	101.55
12	**549.40 C5#**	58	170.13	104	100.65
13	**524.01 C5**	59	167.61	105	99.76
14	**500.85 B5**	60	165.17	106	98.89
15	479.66	61	162.80	**107**	**98.04 G2**
16	**460.19 A5#**	62	160.49	108	97.20
17	**442.24 A5**	63	158.25	109	96.37
18	425.63	64	156.07	110	95.56
19	**410.23 G4#**	65	153.95	111	94.76
20	**395.90 G4**	66	151.89	112	93.97
21	382.54	67	149.88	113	93.20
22	**370.06 F4#**	68	147.93	114	92.44
23	**358.36 F4**	69	146.02	115	91.69
24	347.38	70	144.17	116	90.96
25	337.05	71	142.35	117	90.23
26	**327.32 E4**	72	140.59	118	89.52
27	318.13	73	138.87	119	88.82
28	**309.45 D4#**	74	137.19	120	88.13
29	301.22	75	135.55	121	87.45
30	**293.43 D4**	76	133.94	122	86.78
31	286.02	77	132.38	123	86.12
32	**278.99 C4#**	**78**	**130.85 C3**	124	85.47
33	272.28	79	129.36	125	84.83
34	265.90	80	127.90	126	84.20
35	**259.80 C4**	81	126.47	127	83.58
36	253.98	82	125.08	128	82.97
37	**248.42 B4**	83	123.71	129	82.37
38	243.09	84	122.38	130	81.77
39	237.99	85	121.07	131	81.19
40	**233.09 A4#**	86	119.79	132	80.61
41	228.40	87	118.54	133	80.04
42	223.89	**88**	**117.31 A3#**	134	79.48
43	**219.55 A4**	89	116.11	135	78.92
44	215.38	90	114.93	136	78.38
45	211.36	91	113.78	137	77.84

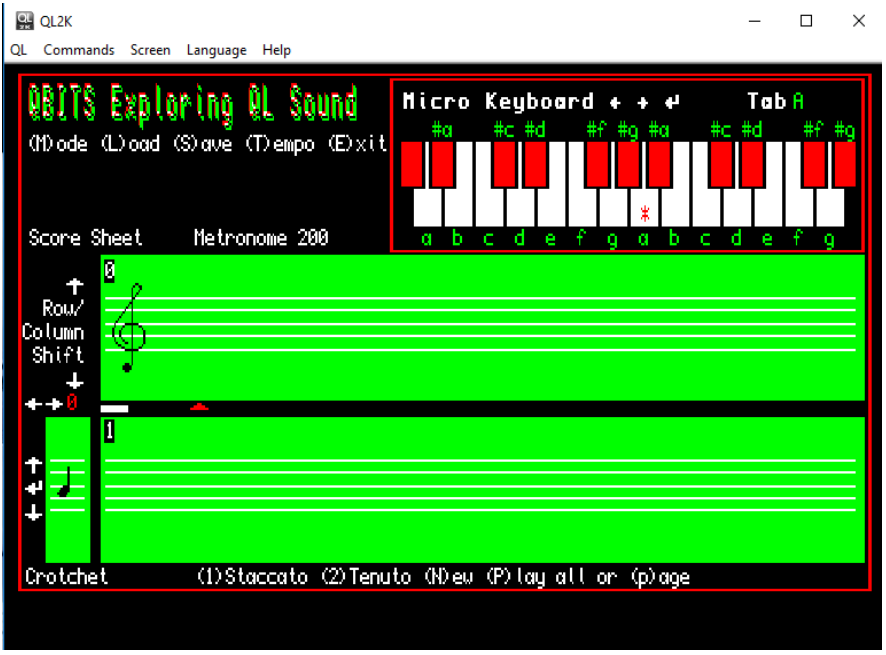
138	77.31	179	60.42	220	49.58
139	76.79	180	60.10	221	49.37
140	76.27	181	59.78	**222**	**49.15 G1**
141	75.76	182	59.47	223	48.94
142	75.26	183	59.16	224	48.73
143	74.76	184	58.85	225	48.52
144	74.27	185	58.54	226	48.31
145	73.79	**186**	**58.24 A2#**	227	48.11
146	73.31	187	57.95	228	47.91
147	72.84	188	57.65	229	47.70
148	72.37	189	57.36	230	47.50
149	71.92	190	57.07	231	47.31
150	71.46	191	56.79	232	47.11
151	71.01	192	56.50	233	46.92
152	70.57	193	56.22	234	46.72
153	70.14	194	55.94	235	46.53
154	69.70	195	55.67	236	46.34
155	69.28	196	55.40	237	46.15
156	68.86	**197**	**55.13 A2**	238	45.96
157	68.44	198	54.86	239	45.78
158	68.03	199	54.60	240	45.59
159	67.63	200	54.34	241	45.41
160	67.22	201	54.08	242	45.23
161	66.83	202	53.82	243	45.05
162	66.44	203	53.57	244	44.87
163	66.05	204	53.31	245	44.70
164	65.67	205	53.06	246	44.52
165	**65.29 C2**	206	52.82	247	44.35
166	64.91	207	52.57	248	44.17
167	64.54	208	52.33	249	44.00
168	64.18	209	52.09	250	43.83
169	63.82	210	51.85	251	43.66
170	63.46	211	51.61	252	43.49
171	63.11	212	51.38	253	43.33
172	62.76	213	51.15	254	43.16
173	62.41	214	50.92	**255**	**43.00 F1**
174	62.07	215	50.69		
175	61.73	216	50.47		
176	61.40	217	50.24		**32.70 C1**
177	61.07	218	50.02		
178	60.74	219	49.80		



Note: For faster **QL platforms** and **Emulators** (QL multiplier x10 to 1000) **Exit** Intro page with <**Esc key**>. The **BEEP** waveforms and parameters will automatically update without having to use the <**Enter key**>.

QBITS Exploring QL Sounds	**Key Commands**	
Cursor Left/Right	Change key on Micro Keyboard	
Cursor Up/Down	Change Displayed Symbol	(Score Mode)
Shift Cursor Up/Down	Change selection of BEEP parameter	(BEEP Mode)
Shift Cursor Left/Right	Change value of selected BEEP parameter	(BEEP Mode)
	Change move Note marker increase/decrease Score line number	(Score Mode)
Spacebar	Cancel active BEEP	(BEEP Mode)
	Toggle upper/lower direction of highlighted Note marker	(Score Mode)
Enter	Activates a Cursor Key Selection	(BEEP Mode)
	Enters selected Note/Symbol on Stave and updates Score array	(Score Mode)
Tab	Toggle between **A/B** Micro Keyboard BEEP parameters	
(**M**)ode	Press **M/m** to toggle between Modes **BEEP** & **Score**	
(**L**)oad	Press **L/l** to Select **QBSDat_0-9** and **Load** file	
(**S**)ave	Press **S/s** to Select **QBSDat_0-9 Save** current BEEP **A/B Sets** & **Score Sheet**	
(**T**)empo	Press **T/t** to change the **Beat** and **Metronome** values	(Score Mode)
(**P**)itch	Press **P/p** to play Beep Duration, Pitch	(BEEP Mode)
+ (**H**)armonic	Press **H/h** add harmonic time interval (grad_x),Step 9grad_y)	(BEEP Mode)
+(**W**)rap	Press **W/w** above plus Wrap parameter	(BEEP Mode)
+(**F**)uzz	Press **F/f** above + Fuzzy parameter	(BEEP Mode)
+(**R**)andom	Press **R/r** above + random parameter	(BEEP Mode).
(**1**)Staccato (**2**)Tenuto	Press **1** or **2** to add Articulations Staccato or Tenuto(Legato)	(Score Mode)
(**N**)ew	Press **N/n** Clears all old Score entries	(Score Mode)
(**P**)lay all or (**p**)age	Press **P** to play all score lines 0-9. Press **p** to play page only	(Score Mode)


QL2K
QL Commands Screen Language Help
QBITS Exploring QL Sound
(M)ode (L)oad (S)ave (T)empo (E)xit
Micro Keyboard ← → ↵ Tab A
#a #c #d #f #g #a #c #d #f #g
a b c d e f g a b c d e f g
Shift ←↑↓→ ↵
Duration : 0 (0-235)
Pitch : 15 (0-255)
Harmonic : 0 (0-255)
Time step : 0 (0-235)
Pitch Step: 0 (-8-+7)
Wraps : 0 (0-15)
Fuzz : 0 (0-15)
Random : 0 (0-15)
Explore QL Sounds
BEEP: 0 15 [0 0 0] 0 0 0
Pitch & Harmonics
Wave Forms
Play (P)itch +(H)armonic +(W)rap +(F)uzz +(R)andom
Cancel BEEP


QL2K
QL Commands Screen Language Help
QBITS Exploring QL Sound
(M)ode (L)oad (S)ave (T)empo (E)xit
Micro Keyboard ← → ↵ Tab A
#a #c #d #f #g #a #c #d #f #g
a b c d e f g a b c d e f g
Score Sheet
Metronome 200
Row/
Column
Shift
Crotchet
(1)Staccato (2)Tenuto (N)ew (P)lay all or (p)age