Elliott 903 Music

Andrew Herbert
Cambridge
May 2015

Introduction

Like many machines of its era (mid 60s-mid 70s), the Elliott 903 (a.k.a Elliott 920B) had a loudspeaker to give audible feedback on program execution. A running program produced a varying warble, a dynamic stop a high-pitched whistle and a stopped machine, silence (apart from the steady whirr of the fans). Inevitably programs were written to play musical tunes by executing carefully tuned sequence of instructions. (The concept of a "sound card" didn't exist at that time, and although a 903 could be fitted with a digital to analogue convertor, this was not common and providing one to just play tunes for amusement would have been considered profligate.)

As part of my project to collect Elliott 900 software, several music programs have been found. Most have been resurrected sufficiently to be able to test them and make comparisons, the findings of which are reported here.

The Hardware

The audio tone output by the loudspeaker is dependent on the repetition rate of the computer's "VTG" waveform. The VTG pulses control a relaxation oscillator in the computer control panel that in turn drives the loudspeaker. The frequency of the oscillator depends upon the VTG pulse repetition rate, which in turn depends upon the sequence of instructions obeyed and the time they take to execute. VTG is pulsed twice in most instructions, three times by function 6 (collate) and just once by function 14 (shift). Order 1 (add) typically takes 24µsec, order 13 (divide) 80µsec.

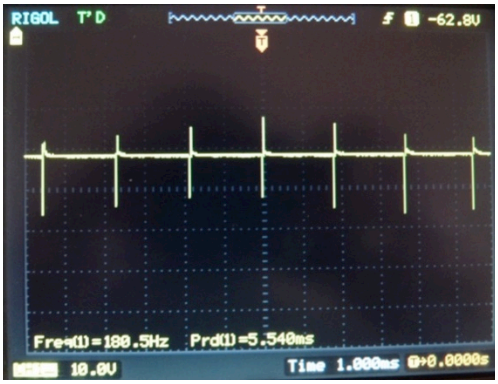
To produce a scale of musical notes, wider spacing of VTG pulses is required than that offered by differences individual instruction times. Salvation lies with instruction 14 n that shifts the accumulator n places left. The duration of the instruction is 22+3nµsecs, i.e., the larger the value of n, the greater the spacing before the first VTG of the next instruction. In the machine specification n is required to be between 0 and 48, higher addresses being reserved for various forms of I/O transfer. In practice, on the 903/920B, n can be as large as 2047 and this is sufficient to allow a musical scale to be constructed.

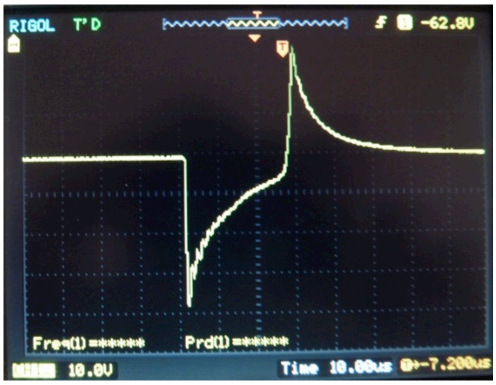
The simplest tone generator is a loop that includes a 14 instruction giving sufficient delay for the burst of VTG pulses to make the control unit oscillator circuit emit the desired note:

```
L     14    NOTE        (Shift NOTE places)
      10    DURATION    (Increment DURATION)
      4     DURATION    (Load DURATION into accumulator)
      9     L           (Jump if negative to L)
```

`DURATION` is assumed to be a store location that contains a negative count of how many loop cycles are required to emit the tone for the required duration. The desired tone is obtained by choosing an appropriate value for `NOTE`.

The two images below show typical oscilloscope traces for the signal sent to the loudspeaker recorded on my 903. The first image shows a train of pulses arising from a loop as described above: the distance between the spikes is the delay due to the 14 instruction. The second shows a single pulse in more detail.





Similar results were obtained from the TNMoC[1] and CCH[2] 903s, although with these there was a lot more noise between the pulses. In particular the TNMoC machine shows significant breakthrough every third cycle.

On my 903, `NOTE`=108 produces A5 (220Hz), `NOTE`=36 A4 (440Hz), `NOTE`=1 A3 (880Hz). Other 903s perform quite differently. The table below show the shift values for a Chromatic scale on my 903, the CCH 903 and the TNMoC 903.

Note	A	A#	B	C	C#	D	D#	E	F	F#	G	G#	A'
Freq	220	233	247	262	277	294	311	330	349	370	392	415	440
AJH	108	99	93	85	77	71	66	60	54	50	45	40	36
CCH	259	234	208	185	169	151	137	123	110	99	90	80	72
TNMoC	303	268	237	207	185	163	144	128	118	103	90	80	70

The table was produced by connecting an oscilloscope to the loudspeaker of each machine and adjusting the shift count (`NOTE`) until a value was found that produced required frequency for each note in the scale. The rate of instruction execution increases (by as much as 10%) as a 903 warms up. It is best to calibrate a machine after it has been computing hard for 30 minutes or more.

It might be though the differences between the machines is simply one of transposition, but lining up the sequences on values of `NOTE` shows this is not so:

[1] TNMoC – The National Museum of Computing on Bletchley Park. A 16K machine.

[2] CCH – Centre for Computing History, Cambridge. An 8K machine.

AJH	108	99	93	85	77	71	66	60	54	50	45	40	36
CCH	110	99	90	80	72	65	58	52	47	41	36	32	27
TNMoC	118	103	90	80	70	62	56	49	42	38	33	28	24

In principle the pulse repetition rate driving the relaxation oscillator can be calculated: PRR = loop time + n * unit shift time, where loop time is the time to execute a loop containing `14 0` and unit shift time is the increment for each additional place shifted. (Nominally loop time is 96µs and unit shift time is 3µs, but variations of up to 10% from these figures have been measured on individual machines).

If the emitted note was proportional to the pulse repetition rate, the repetition rate for notes one semitone apart should be in the ratio $2^{1/12}$. Taking pairs of notes and solving simultaneous equations reveals this is not the case and so the frequency produced by a loop cannot be easily predicted mathematically and so each machine must be tuned by measurement. No wonder then that programs which sound tuneful on one machine sound dreadful on others.

Writing Music

Having found a means to play notes, the next requirement is to devise a means of representing a tune. For each note to be played we need to specify a value for both `NOTE` and `DURATION`, and, if we want to specify absolute durations, we have to calculate the number of loop cycles to be executed, based on the value of `NOTE`.

The most direct input format would be <`NOTE, DURATION`> pairs, but this would require the "composer" to both remember the `NOTE` values and compute correct values for each `DURATION`. Better to have a textual notation giving the name of a note and its duration in some time unit, e.g., `C1` meaning 1 unit of middle C. To cover several scales a notation for octaves is useful – e.g., upper case and lower case letters for a low and high register, or perhaps a symbol such as `^` to specify one octave above, `^^` two octaves above and so on. Similarly a suffix notation for sharps, e.g., # and flats, e.g., `@` could be provided. Other useful things to denote are rests (pauses), slurring (notes that run into one another rather than being separately voiced) and repeated sections.

The Recovered Programs

"MUSIC"

This is a program I remember hearing as a schoolboy when using the Elliott 903 at the Medway and Maidstone College of Technology, and again as a student at Leeds University. It comes with three tunes: Rimsky-Korsakov's "Flight of the

Bumble Bee", Mendelssohn's "Wedding March" and Handel's "Arrival of the Queen of Sheba".

An original source version of this program has yet to turn up, but the Elliott 903 paper tapes at TNMoC included two store dumps of the player and a source has been reconstructed from these. It is believed the dumps originated at Aldenham School.

The program consists of a translator and a player. The translator reads in a textual score and punches tape of integer `<NOTE, DURATION>` pairs as decimal numbers terminated by a halt code. The player then reads the output tape back in; the numbers are stored; and, on reaching a note of -1, the tune is played. At the end of the tune, the player then attempts to read in another tape.

The tune tapes are quite large as the numbers are punched in space-aligned columns.

In the textual input, pitch is expressed as a letter `A-G`, possibly followed by one of more `^` symbols to raise the pitch by one or more octaves, `'` to sharpen by a semitone or `@` to flatten. The duration is an integer representing units of about `10`ms. The letter `S` terminates the input. The letter `R` causes a rest of about 1msec (and is translated to a loop of `14 900` instructions which produces a barely audible growl from the loudspeaker). Thus the beginning of "God Save the Queen" can be coded as:

```
G3G3A3F'2G1A2B3B3C2B1AG3S
```

It would seem the program was generally distributed in binary form and the tunes in numerical rather than textual format. No source for the program has been found, nor any tunes in textual form.

Interestingly, constructing histograms of the notes in the recovered tunes revealed some differences in pitch values for notes than those embedded in the translator part of the recovered program.

Subsequent to the first draft of this article a further version of this program have been found in Terry Froggatt's collection. The original came from an Elliott 903 formerly belonging to Alistair McCowan & Associates in Pontefract, Yorkshire. (McCowan's were part of the infamous Poulson organization.) The tape is labelled "TUNES (VARIOUS), FOR OLD 8K ONLY" suggesting it came from a machine with an additional fast store. The code is positioned differently in store and has some minor coding differences from the Aldenham version but is otherwise identical. The tunes on the tape are, "Men of Harlech", "God Save the Queen", "Old Lang Syne", "Ash Grove", "Land of My Fathers" and "Land Lubbers Lying Down Below".

"Museum of Scotland (MOS) MUSIC"

A music tape was found in a collection of tapes at the Museum of Scotland associated with their stored 903. This tape has just a player part without any translator. It has "The Sailor's Hornpipe" following on from the program.

The program is in the form of a clear store followed by the Elliott "relocatable binary loader", followed by the program in relocatable binary form (both as output by the SIR assembler), followed by the numerical form of the Hornpipe.

The position of the program in store is curious, starting at location 434, a not particularly convenient entry point to set up on the control console. A conventional entry at 32 works as fortunately zeroed locations between 32 and 434 are effectively "no-ops" – instruction 0 just loads the B-register from itself in this context!

The program closely resembles "MUSIC", but the player loop is a little different in that it uses a B-modification to vary the shift (14) instruction rather than patching store.

The pitch values in the Hornpipe tune do not correspond to those used in MUSIC, or indeed any of the other music programs I've found.

"MAESTRO"

This program is elusive – I have found documentation for it, written at Queen Elizabeth College, but no trace of any player. One tune tape, labelled "Cockles and Mussels" has turned up and is in a binary format.

The documentation describes a comprehensive program with facilities for inputting and playing textual music, producing binary tapes and playing them, retuning the player and using the teleprinter as a keyboard for interactive music playing.

The textual notation uses forms like A, BF, CS, AA for A, B flat, C sharp and A an octave above. Durations are specified by integers. "God Save the Queen" is rendered:

```
G3G3A3FS2G1A2B3B3C2B1AG3S
```

When played each note has a built in rest, which can be suppressed by a following `½` character (i.e., slurred). A normal rest is coded by — (minus). Further features allow individual notes to be repeated (←), sections of music to be repeated, e.g., `5[ A2 B2 C2 ]` (5 repeats of the enclosed sequence) and "procedures" declared and called, e.g., `(PPP A2 B2 C2) ... PPP5` (declare procedure `PPP` and call it 5 times).

In the description of how to use the tuning facility, it said that each note is a 50 word subroutine containing a loop of 40 instructions comprising a sequence of `6 1` (collate) and `14 47` (shift) instructions, the pitch of the note being determined by the relative proportion of each. The tuning program allows the proportion to be varied and the total shift count to be increased or decreased. It might be thought this would allow a finer grained control of pitch than the simple loop of the MUSIC program, but the documentation that came with "The Entertainer" tape complains that MAESTRO is badly out of tune. This might just be due to difference in the instruction timing between machines, and indeed as some schools were donated the much slower 920A rather than the faster 903/920B, music tuned for one would indeed play badly on the other.

Unfortunately there is no specification of the binary format so I've not been able to unpick the "Cockles and Mussels" paper tape.

"The MUSIC SYSTEM"

Also found at TNMoC was a short document describing a player called "THE MUSIC SYSTEM", introducing it as an improvement over MAESTRO and giving a table of pitches and unit interval durations, different again from those in the "MUSIC" program. The operating instructions are as for the MUSIC program. A numerical format tape, labelled "The Entertainer" (i.e., Joplin's piano rag used as the theme tune for the film "The Sting") has been found with pitches corresponding to the table in the document.

"ALGOL MUSIC"

In addition to the version of "MUSIC", two Algol 60-based music programs were in Terry Froggatt's "Pontefract" set of paper tapes.

The first program reads in tunes in textual form and punches out a tape of pairs of integers. The input format uses the same convention as MAESTRO, but does not include the facilities for slurring, repeats and procedures. The embedded pitch table is different again from that in other programs. No scores have been found corresponding to the table. This program remains an enigma.

The second program reads in pairs of integers into two vectors in store and then calls a machine code procedure to play the notes. Before reading the notes, a first pair of integers is read, the first of which determines the number of notes in the tune and the second is a scaling factor for the duration of the played notes (like a metronome mark on printed music).

The player follows the familiar form of the MUSIC program, except that B-modification is used to adjust the shift count rather than calculating and patching an instruction into the loop. By default three `14 2047` instructions are executed after each note played, unless the next note is of "pitch" 0, in which case the next note is played immediately (i.e., slurred). This is reminiscent of the description of the MAESTRO player's features. To my ear, the pause between voiced notes makes the music sound stilted.

A number of Christmas carols, folk songs and pop tunes were found in the numerical input format: some were clearly student "experiments", others quite reasonable renditions of the tunes they claimed to be. One tune was identified with the author's name: Karen Senior. Does anyone remember this name from their schooldays?

"TJF MUSIC"

This player has been the easiest to research as the author is well-known as the CCS rapporteur for Elliott computers.

This program is interesting as it was written to run on a 920B with a digital to analogue convertor. If output via the convertor is requested, the program sends a sequence of amplitude values to the DAC that output a sine wave of the desired frequency. If output is via the loudspeaker the frequency value is converted to a shift count for the familiar shift loop. The shift count is merely a calculated value and therefore can only be related to the nominal frequency by external calibration.

The input format consists of upper case letters `A–Z` representing successive whole notes and `a–z` representing the semitone succeeding the corresponding whole note, followed by a digit denoting duration. Thus `a` is A sharp, `b` is B sharp (i.e., C) and so on. It is also possible to specify octaves using + and – suffices, and `$` for sharp. Input is terminated by a halt code. Writing scores as long lines of letters makes them very compact, although not so easy to read as the formats that uses octave, sharp and flat symbols.

"God Save The Queen" can be rendered as:

`G3G3A3F$2G1A2B3B3C2B1AG3` or `G3G3A3f2G1A2B3B3C2B1AG3S`

The program reads in the textual score and stores it in a larger buffer. The text is translated note-by-note on play out. The effect of this is to put a small "rest" between notes rather than slurring them together, making the music sound crisp. The value for each note is calculated using a table of frequencies for notes from which a shift count is calculated using the nominal instruction times for the play out loop. This means that the quality of the play out depends upon how closely the actual instruction times are to their nominal values and whether or not the emitted note is proportional to the generated VTG pulses. On my 903 it is tolerably ok, but on the TNMoC and CCH machines quite badly out of tune.

TJF MUSIC comes with two tunes: one is a large collection of Christmas melodies, the other a tape called "Mandy's Music" containing Clarke's "Trumpet Voluntary" and Bach's "Jesu Soul of Man's Desiring". A tape for "God Save the Queen" also exists, written for Her Majesty's visit to TNMoC.

"AJH MUSIC"

Having explored the recovered music programs I set about making a music program of my own and producing scores for it that play all the recovered tunes. The input is basically TJF MUSIC format, but with the notes pitched an octave lower to give a larger musical range (3 octaves) and with A set to A6 (110Hz). Additional tokens include `*` to double the tempo and `/` to halve it, `&`*n* to reduce the tempo by a factor of 1.*n* and _ to play a "rest".

For the several collected tunes, using either the pitch tables from the relevant player, and/or constructing histograms of the pitches in each tune tape, I wrote a translation program to map to the notation and pitches of my player.

My program has to be modified for each machine it is run on if the music played is to be in tune. The required shift counts for each note have to be determined by measurement and then inserted into the program data. In the case of the CCH and TNMoC machines music has to be transposed to be within shift count values that produce audible notes. There would be some difficulty in assembling a 903 choir given the mismatches and limits in musical range of the various machines.

Conclusions

This investigation of Elliott 903 music playing programs has revealed a surprising amount of (misplaced?) energy invested by 903 users in getting their machines to play barely passable tunes. Common elements included the use of a convenient textual format to capture the notes to be played and the need to tune the player to each different machine. Most of the music systems were very basic, but one included exotic features such as slurred versus voiced notes, repeats and even "procedures". There are some gaps in the record. We have no source or binary code for MAESTRO and the documentation is insufficient to reconstruct it easily. We have an Algol music player with no textual input translator and another vice versa. We have the MOS player but no translator. If any readers can help fill these gaps the author would be pleased to hear from them.

A recording of AJH MUSIC playing classical tunes can be downloaded from:
http://tinyurl.com/903music