1-bit music. by dave.V1

1-bit computer music has a bunch of history; going all the way back to the 1940's when mad boffins found that they could do cool stuff using their millions of valves and diodes in their 3 story office building. The term 1-bit music is kinda a generic world wide term - probably origining from these mad fellows turnnig their single mega-watt valve on and off a bunch of times to the them of "I dream of daisy". Who knows.
Sure, the VZ uses 2-bit music. Bits 0 and bit 5 of $6800 (or 26624). Toggle either bit 0 and/or bit 5 at the right frequency, and you have sent a pulse or two to $6800 which then drives the fancy innards of microcode in the ROM to drive logic stuff to drive the piezo speaker. Or that 10" sub sitting in the corner that is wired to the VZ via a capacity coupler. Doof doof. And let the tunes fly.
Depending on the method of the code that is toggling port $6800, essentially you can make various dfiferent sounds, from the BASIC SOUND command, through to (not a full list) polymorphic-sounding sounds, PCM & sampled waves, FM or AY tones, SN76489 compatible sounding tones, SID type music.
Two gentleman that have done wonders in the music scene are Utz and Shiru. Their names do pop up from site to site and, when you go looking, are quite profound coming up in all sorts of webpages and youtubes. Together they have created numerous engine players for the ZX-Spectrum and the ZX-81.
Since the ZX-Spectrum has the greatest potential out there for awesome music already made, its a brilliant start to start from.

One-Bit Forum
http://randomflux.info

Utz listing of players, editors :
http://randomflux.info/1bit/viewtopic.php?id=25

Utz Git-hub source
https://github.com/utz82/ZX-Spectrum-1-Bit-Routines

Shiru
http://shiru.untergrund.net/1bit/

Battle of the Bits (Stuff regarding various engine players)
http://battleofthebits.org/lyceum/View/zxbeep%20(format)

Garvalf
http://garvalf.online.fr/index.php?page=musiques_zx.en

I started off by playing around with Fur Elise , found on www.ticalc.org for the Ti83 calculator (of all things!). After a bit of fiddling the 1-bit sample player, with the Fur Elis sample, actually compiled correctly (this was in a Z88DK C wrapper, so essentially the #ASM code was compiled within C).
Played it through VZEM, and I was both gobsmacked and then hooked. This was sitting on my back verandah on a sunny day in 2013. Time passed and fast forward to March? 2019, when Wham! was successfully assembled and ran, and actually produced a listenable tune.

Anyhoo, first thing is first. Check out this.
Utz - Tutorial : How to write a 1-bit Music routine.
http://randomflux.info/1bit/viewtopic.php?id=21

The ZX spectrum uses Port $FE for its speaker, opcodes being OUT ($FE), A
In a few engines, code also refers to OUT (C), A where C is also typically $FB. And, of course, register A being the data sent out to the port.
I can only assume it is opening and closing the music port.

The ZX81 pretty much appears to use OUT (C), A with OUTI along the way. I havent delved in to find the ZX-80/ZX-81's output port.
Since the VZ does not use a port, but uses a memory location, we need to do a few changes. The VZ uses bit 0 and bit 5, for lo-res text purposes, we need to mask out everything else bar these two bits - otherwise MODE(1) kicks in and displays all sorts of cool / weird effects. We do this by simply adding an AND 33 opcode before our LD ($6800), A

```
Spectrum                        VZ
        blah                            blah
        ld b, frequency                 ld b, frequency
loop:   blah                    loop    blah
        out ($FE), A                    and 33
        blah                            ld ($6800), a
        blah                            blah
        djnz loop                       blah
                                        djnz loop
```

This does introduce a few issues. Most of the engnies have precise T cycles being counted, and with the various methods of toggling the bits, jumping to the "push" or the "pull" cycle (when toggling the speaker) can introduce different number fo cycles per jump routine. That probably doesnt read correctly. Essentially the speaker pull routine really ought to have the exact same number of clock cycles as the routine that does the speaker push. That may read better. By doing a JUMP-TO-THE-PUSH-IF-SOMETHING-happens, there is always going to be a slight difference in timing - even the jump opcode itself introduces one to four more cycles by doing the jump than slipping through the jump. This slight difference can introduce noise.
To fix this, we add useless code : example, such as a NOP operand (a four cycle no-operation opcode that does nothing but chew four T cycles).
By using the precide "useless" code, you can tune in the music and essentially wipe out the noise.

Most of the Spectrum engines are assembled with PASMO assembler. Found here : http://pasmo.speccy.org/
It is a PC executable that creates a Z80 assembled object output file. We then have to add the VZ header to this, by using RBINARY.EXE which is found on a number of VZ sites; should also be under the files section somewhere. This creates out final .VZ file that can be run on an emulator or real hardware.

```
 pasmo file.asm
 rbinary file.obj file-out.vz
```

Emulators that do appear to be lose to benig 100% perfect to real hardware : VZEM, MAME, WINDSEVZ200. Unfortunately it has been found that JVZ200 , JAVA VZ, PC EMULATOR2001 don't appear to be able to successfully re-create the music that these engines can make.
Also in saying this, not all engnies yet work with the emulators or real hardware. I have found that some music engines work on VZEM fine, but not on real hardware as of yet. One such beauty is the very noiseless , brilliant sounding Phase_Squeek.
Others, such as Octonode, Huby, Quadtropic, Ntropic, ZX10 all play on VZEM, MAME and real hardware ultra fine and dandy.
Utz has also advised me that there are a few that are "cycle locked" to the Spectrum, and he assures me that even he would have trouble modifying it to fit the specifics of an alien computer to himself, such as the VZ. One such example is the Pytha engine.

Over to the Z88 C Dev Kit. www.z88dk.org
The fellows doing Z88 have gone and started to release new libraries for all new platforms. Unfortunaetly the VZ has missed out, and if you are a gun-ho C programmer, you too can join the crowd, create or help create the new libraries, and that would be a great win. Anyoo - with these new release libraries for the various platforms, one of Shiru's engines "Tri-Tone" and "Beep-FX" have been added and are being supported. Unfortunately for the life of me, I can not get either to

work , either through Z88DK , or through Utz's listings via asm code. The Tritone engnie does appear to have the largest selection of music available.

Editors & trackers

Cool stuff follows!
Three editors to grab are :
-Beepola
-Bintraker
-1Tracker.
Google these, download, extract and run.
All three are PC executable music trackers , that you can create your tunes on. They all have a few bits of demo music that can be loaded, played and , extracted to an ASM file with or without the relavent music player. All ready to assemble for the ZX Spectrum. For us, we just need to extract and/or modify the engines to suit the VZ.

Listing of all / most music enies for the ZX Spectrum and the ZX-81.

anteater	anteater: converter now correctly handles XMs created by OpenMPT
beepertoy	add beepertoy engine Jun 10, 2016
betaphase	betaphase: improved rest method that does not destroy phase
bm-1	bm-1: add SuperSquare patch Apr 5, 2017
fluidcore	
nanobeep	nanobeep: converter now correctly handles XMs created with OpenMPT
nanobeep2	add nanobeep2 engine Aug 19, 2017
ntropic	ntropic: converter now correctly handles XMs created with OpenMPT
octode2k15	octode2k15: converter now correctly handles XMs created with OpenMPT
octode2k16	add octode2k16 engine May 26, 2016
octodepwm	octode pwm: converter now correctly handles XMs created with OpenMPT
phaserX	PhaserX: fix typo in music.asm Oct 25, 2016
phasesqueek	PhaseSqueek: replace note name "h" with "b" Oct 31, 2016
povver	povver: fix typo in equates file Nov 21, 2016
poww	initial commit Mar 16, 2014
pytha	clear channel counters on note-only data reloads to stop noise leakin…
qaop	converters now correctly handle XMs created with OpenMPT
quattropic	quattropic: converter now correctly handles XMs created with OpenMPT
rawp	converters now correctly handle XMs created with OpenMPT
squeekerplus	squeeker plus: fix ch4 overloads, finally Aug 29, 2016
stringks	add StringKS engine Nov 15, 2018
tritonefx	corrected documentation Sep 22, 2015
vibra	add Vibra engine Jul 18, 2017
wtbeep	ensure pattern read does not leave register set swapped (Shiru)
wtfx	optimized data loading, replaced faulty noise samples, updated docs
xtone	converters now correctly handle XMs created with OpenMPT
yawp	converters now correctly handle XMs created with OpenMPT
zbmod	zbmod: report actual excess size on data overflow

...quickly slapped together on Friday night, 20th Sep, 2019. Sitting in Parkes Caravan Park!