

This is a brief description of the workflow that produces a music tape that the PDP-1 can play.

(1) Find your sheet music, and break it down into separate lines (voices) of music. The playing program can handle up to four simultaneous notes. As a sample, see the file BWV592-3.pdf which is the third movement of Bach's Organ Concerto in G (BWV 592) which we generally use in demos.

(2) Code each line of music separately, as input to the Music Compiler (also called the Harmony Compiler). The compiler input language is described in a document of which there are two copies, MusicCompiler-a.pdf and MusicCompiler-b.pdf. Read the two copies together, since neither is entirely complete and each has typos which mostly do not coincide.

(3) Type in each voice. For example, see the file BWV592-3.txt which corresponds to the sample piece mentioned above. Assuming you do this in ASCII, you will need to convert it to FIO-DEC, which is the PDP-1 character set and encoding. FIO-DEC is an 8-bit code, documented in the PDP-1 Handbook and the Quick Reference Card. The Handbook (F15D_PDP1_Handbook_Oct63.pdf) can be found on bitsavers.org. Typically, as shown, all four voices can follow each other on one "tape" for convenience. They are separated by the at-sign in ASCII, which represents the FIO-DEC "stop code", octal 13, which is effectively a page break or end-of-tape mark.

(4) Run the compiler, presumably on a PDP-1 simulator such as SIMH. You may need to debug your input. The PDP-1 binary (RIM format) of the compiler is called hc1d.rim. Each voice is compiled by a separate run of the compiler; it is convenient to append their outputs together on one paper-tape image.

The binary form of a voice is traditionally called "intermediate" format. It encodes each individual note as one 18-bit PDP-1 word. The document music_intermediate_format.doc describes its format.

(5) Convert the output file (typically file type .bin) to actual paper tape. The PDP-1 player program (pdp1m16) should be able to read each successive voice, and then merge them (into an internal "final" format, stored in PDP-1 memory) and play it.

The "final" format uses two successive PDP-1 words to encode an event: 6 bits of Soprano pitch, 6 bits of Alto pitch, 6 bits of Tenor pitch, 6 bits of Bass pitch, and 12 bits of duration. The pitch values 0 and 1 both mean silence; other pitches are in the equal-tempered scale, corresponding to the keys of a 61-key organ manual (plus one bonus pitch!)

This format is optimized for the playing program, which outputs the four pitches of an event for its duration, and then moves on to the next event, etc.

(6) Operation of the current playing program, pdp1m16, is shown in the flowchart music_flow16.png, and instructions for running it are in music_instr16.doc.

(7) This description should be more than sufficient to answer questions from visitors. If someone appears very serious about coding up their own music, they will have to get much more information from Ken Sumrall or me (Peter Samson).