Guide to BOOK Appendix E Examples:
Victor Lazzarini

MIDI Programming with PortMIDI

These are the examples discussed in the text. All of them require PortMIDI to be downloaded and installed. Instructions for this are found in http://portmedia.sourceforge.net/.

Once PortMIDI is installed, assuming its headers are in `/usr/local/include` and the library file is in `/usr/local/lib` the examples can be built as follows:

Windows

On Windows, with the MINGW/MSYS, the PortMIDI library should be installed in `c:/msys/1.0/local/lib` and headers in `c:/msys/1.0/local/include`. This is needed for the system to sees it at `/usr/local/include` and `/usr/local/lib`. Then you can build a program with just

```
gcc -o pmidiout  pmidiout.c -I/usr/local/include -L/usr/local/lib    -
lportmidi -lwinmm
gcc -o pmidiin  pmidiin.c  -I/usr/local/include -L/usr/local/lib
-lportmidi -lwinmm
gcc -o arpeg  arpeg.c -I/usr/local/include -L/usr/local/lib
-lportmidi -lwinmm
```

`-lportmidi` tells the linker to look for **portmidi** and `-lwinmm` tells the linker to look for **winmm**, which is the Windows low-level library on which portmidi depends.

WARNING: on Windows, using the MINGW/MSYS terminal, there might be a buffering issue with `printf()` and `scanf()` in MIDI programs. Because `printf()` output is buffered, it will only be displayed after the program exits, so the program (shown below) will block waiting for input (`to scanf()`), if a device number is entered the program proceeds. When it ends all messages appear on terminal. A solution is to either run the program from the Windows command line or to double-click on its icon.

OS X

Similarly, with OS X once you have portmidi installed, it's just:

```
gcc -o pmidiout  pmidiout.c -I/usr/local/include -L/usr/local/lib -lportmidi
gcc -o pmidiin  pmidiin.c  -I/usr/local/include -L/usr/local/lib -lportmidi
gcc -o arpeg  arpeg.c -I/usr/local/include -L/usr/local/lib -lportmidi
```

This assumes the header files are installed in `/usr/local/include` and the portmidi library is in `/usr/local/lib`. If the library and headers are installed elsewhere, the `–I` and `–L` options have to be changed to match the location of these files.

To run these programs, just type the command name in the terminal, using "`./`" to make sure you are running the programs you have just built (and not something else in your executable path), eg.:

```
./pmidiin
```

Linux

On Linux once you have portmidi installed, it's just:

```
gcc –o pmidiout  pmidiout.c -I/usr/local/include –L/usr/local/lib -lportmidi
     –lporttime -lasound
gcc –o pmidiin  pmidiin.c  -I/usr/local/include –L/usr/local/lib –lportmidi
      –lporttime -lasound
gcc –o arpeg  arpeg.c -I/usr/local/include –L/usr/local/lib -lportmidi
      –lporttime -lasound
```

This assumes the header files are installed in `/usr/local/include` and the portmidi library is in `/usr/local/lib`. If the library and headers are installed elsewhere, the `–I` and `–L` options have to be changed to match the location of these files. You will need to have all the headers and libraries for alsa too (libasound).

To run these programs, just type the command name in the terminal, using "`./`" to make sure you are running the programs you have just built (and not something else in your executable path), eg.:

```
./pmidiin
```