make (mainly adding the debian directory) and sent the details back to Gordon. Hopefully he'll pull them into his Git repository one of these days and start the process of getting WiringPi included in the official Raspbian distribution.

For reference and to give a flavour of the process, below are the details of the changes made to the library.

Most of the changes that I needed to make (as opposed to adding new stuff) are in this commit: https://github.com/hamishcunningham/wiringpi/commit/707cf1bc343e07c9c07eb67c55ed93873c2c67c8

All the changes needed are as follows:

I've added a top-level Makefile which does some of what the build script does (just building and installing the two libraries and the gpio command at present). I believe that the build script still works as it used to. The package has to be built on the Pi itself; I tried cross-compiling but didn't get far :-(
In the library Makefiles this line assumes that the binary has been written into DESTDIR, without PREFIX: @ln -sf $(DESTDIR)$(PREFIX)/lib/libwiringPiDev.so.$(VERSION) $(DESTDIR)/lib/libwiringPiDev.so. This doesn't seem to be the case, however... so I've added a PREFIX. Perhaps I've missed something about how make should be invoked, or...?
The package build process requires that DESTDIR is used for everything — so calling "ldconfig" without parameterising it for DESTDIR causes an error. I've changed it to use DESTDIR for now, and also called ldconfig in the .deb's postinst and postrm scripts. This broke the build script's install, so I added this hack to the library Makefiles: @if [x$(DESTDIR) = x/usr]; then ldconfig; fi
The devLib/Makefile does a -I., but this doesn't pick up the headers in DESTDIR, so I've changed INCLUDE there to do so (in the same way the gpio/Makefile does).
The gpio build needs the core library to be built first, so I've put both the bare "make" and the "make install" under the top-level "install" target. Probably non-optimal.
I have changed the gpio install to use the "install" command, which doesn't assume that the process is running as root.
The .deb installs into /usr instead of /usr/local; I believe that Debian mandates that the latter is reserved for a "local administrator" or similar. The examples builds still work without change to the -I or -L flags as /usr is included in the default paths, it seems.
I have added a compressed version of the man page, as that seems to be expected. I've also made it install the man page in /usr/share as that's the normal place on Debian. That might want parameterising in "build" if you want it to target other operating systems. Doing "man gpio" works but complains about lacking the .1 file; no idea why :-(
I added an "install from .deb" para to the INSTALL file.

6. The End

Here endeth the lesson. Debian packaging is a bit of a black art, but with a little effort (and a lot of copying from our elders and betters!) the results are very powerful, stable and long-lasting solutions to the problems of software distribution and installation. Now it's your turn ;-)