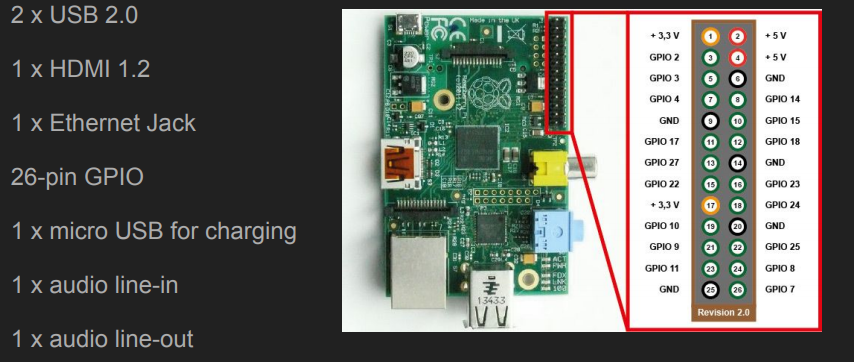
# Hardware



Build Cross Compile

# Get the Platform

1. Create a directory to store everything.

2. Pull the Raspberry Pi toolchain from [***https://github.com/raspberrypi/tools***](https://github.com/raspberrypi/tools)

3. Check you have the directory raspberrypi-tool fully downloaded.

That’s it! You will be using the tools in this directory to build your RaspberryPi applications. This folder contains not just the “building” tools, but also all the standard libraries and system calls that could be needed by your application – we will be using several parts of this directory later.

# Get the libraries

Most non-trivial programs you will build may require standard libraries, or external libraries not yet present on your system to be linked to. In most cases, simply downloading the relevant libraries and placing them in a lib directory was all that was needed. However, this is a bit more complicated for cross-compiling since you can’t reference an Intel compiled library for and ARM binary. There are ways to copy your library files directly onto your main development machine and reference them that way – however, that means you are still dependant on compiling something on your Pi

For this example I’m going to be using the standard pthreads library, a locally-compiled open-source wiringPi library, and the current C11 standard with our compiler.