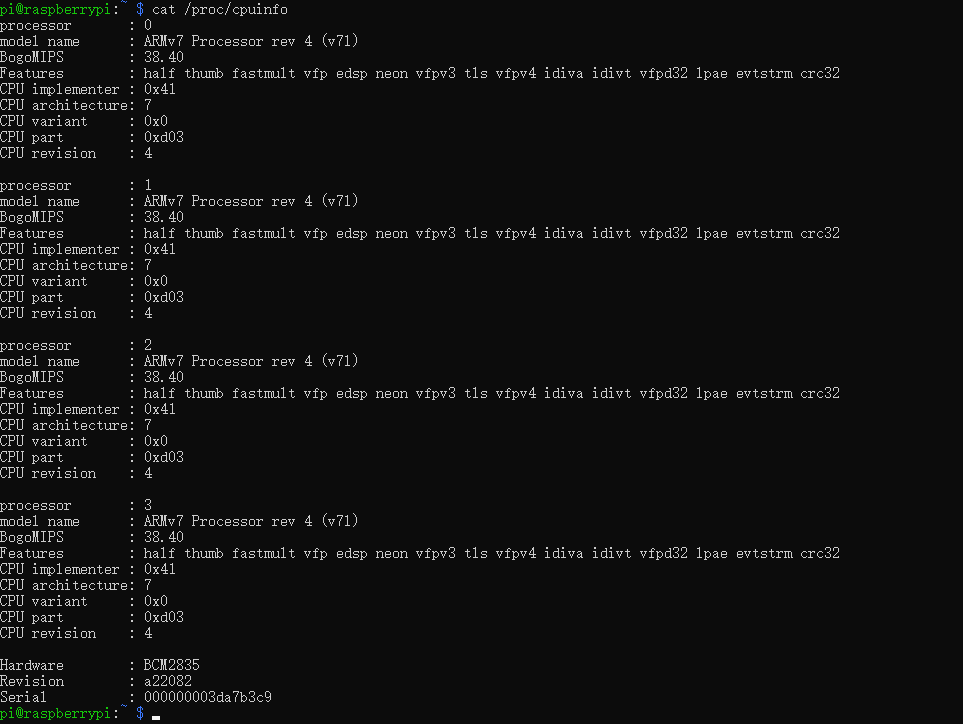
平台：树莓派 bcm2835 Raspberry Pi 3 Model B Rev 1.2





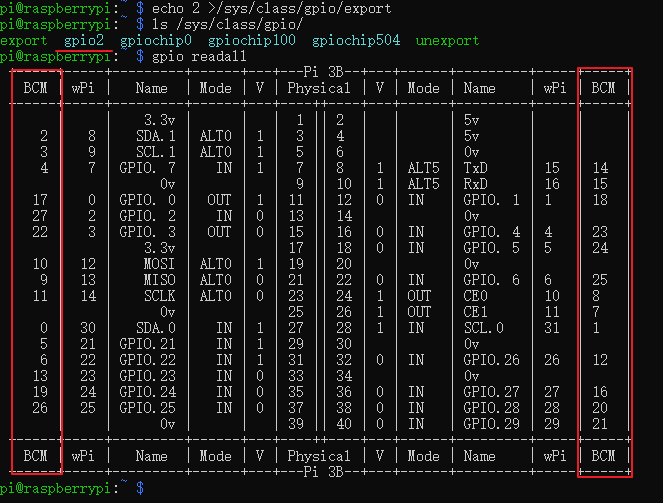
应用层控制gpio目前知道有两种方式——通过gpio设备文件和内存映射

第一种，通过gpio设备文件

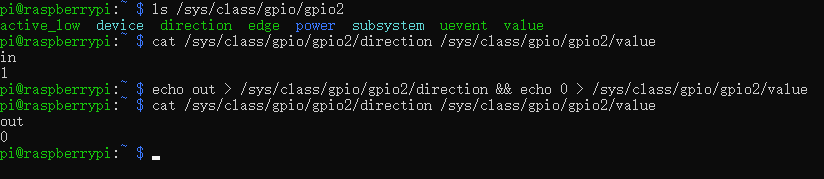
设备文件位于/sys/class/gpio目录下



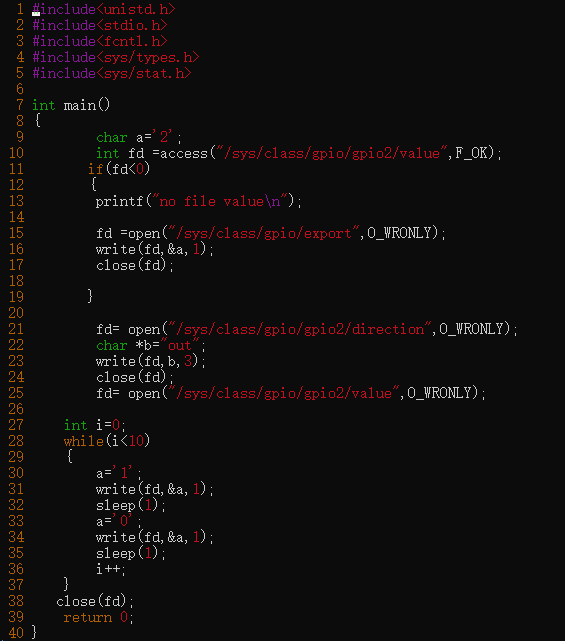
通过向export中写gpio号可以挂载对应的gpio，gpio号是bcm的号，不是wiringPi的。



挂载后在gpio的文件夹下会有配置接口文件，向这些接口文件写命令就能控制gpio了。

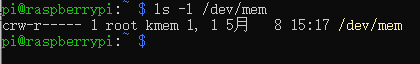


上面用echo这种方式写成shell脚本就可以用脚本控制gpio了。如果通过c语言控制，使用Linux操作文件的API编程即可。

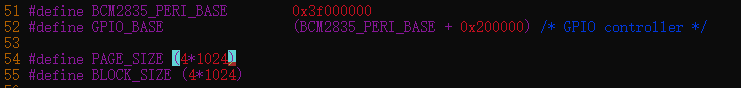


第二种，内存映射

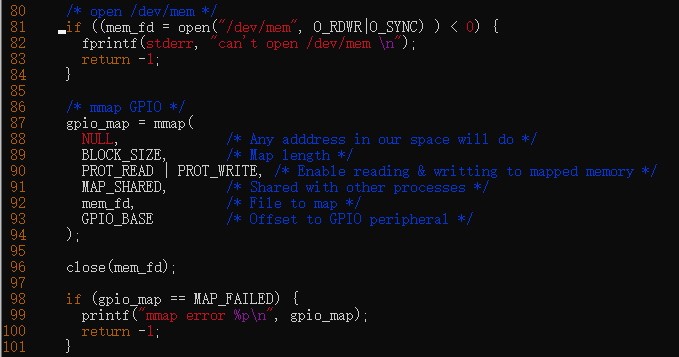
将gpio的地址映射到内存空间，之后所有对这片内存的操作就是对gpio的操作。使用mmap函数实现映射。物理内存的设备文件是/dev/mem。



Bcm2835的外设地址是0x3f000000，gpio的地址是外设地址基础上偏移0x200000。

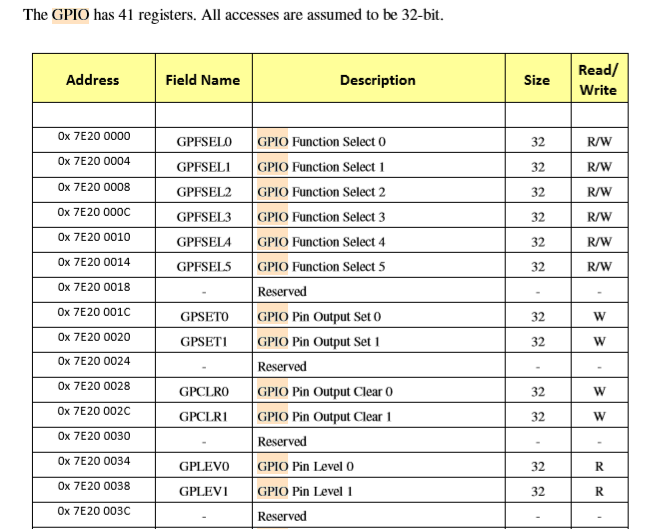


使用open函数可以打开mem设备文件，之后使用mmap函数映射，其返回值就是分配的内存空间的起始地址。

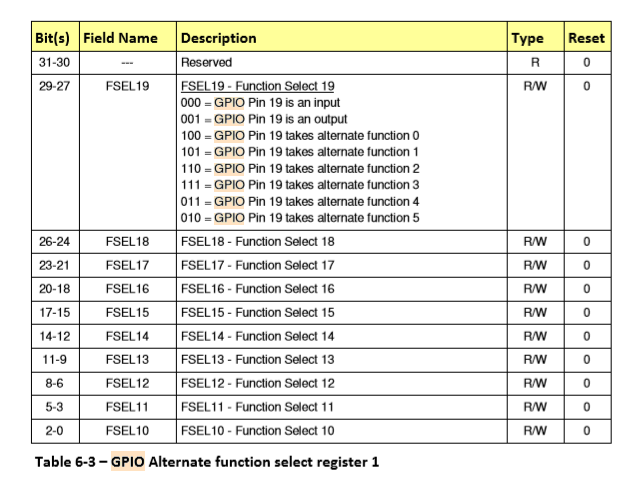


注：必须有管理员权限才能打开mem设备文件，所以运行命令时必须加上sudo。

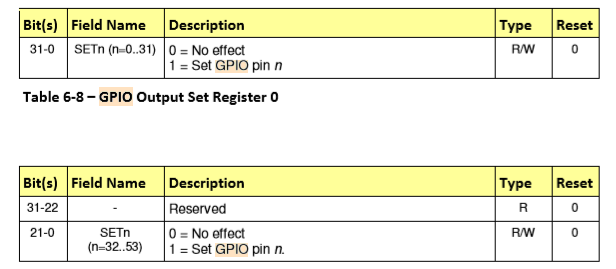
Gpio有41个寄存器，每个寄存器都是32位的。

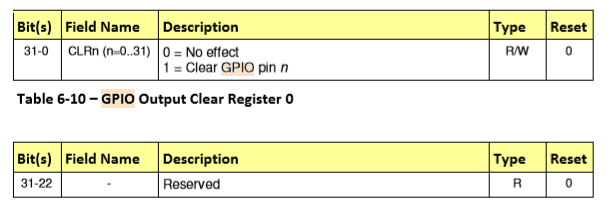


前六个用来设置gpio的功能，前五个寄存器每个设置10个gpio，最后一个设置4个，三个位设置一种功能。

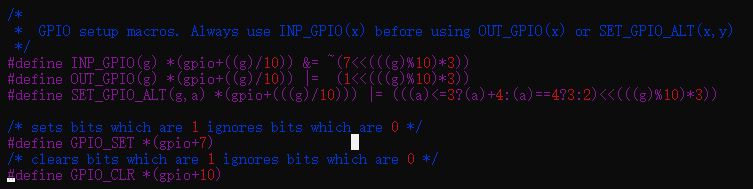


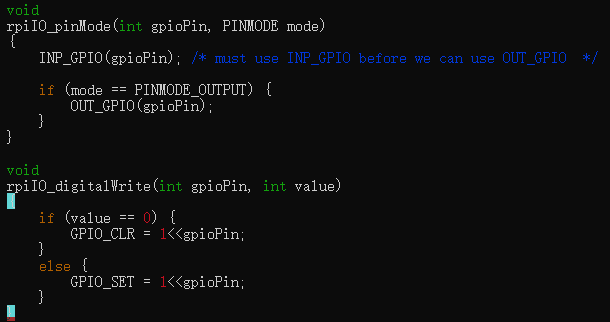
Gpio的设置寄存器是第8和9，清除寄存器是第11和12。一个位对应一个gpio。





根据以上寄存器的布局方式，可以使用下面的代码来设置和控制gpio。





第一种方式比较适合对单独gpio的控制，比如调试和测试，更适用脚本编程；第二种方式适合c代码编程，更加通用方便。

个人水平有限，肯定有错误和遗漏，欢迎大家补充指正。