要想让小车直立起来，我们必须首先获得小车的姿态角度，知道小车目前是什么角度，这点通过MPU6050实现。MPU6050可以获得三轴加速度和三轴角速度，通过特殊的算法可以得到三轴角度，通过角度，我们就能知道小车相对于地面的姿态，从而进行相应的控制调整。

（配图mpu6050）

目前，常用的姿态解算方法有四元数法、卡尔曼滤波和互补滤波，具体算法原理比较复杂，这里不做过多介绍，感兴趣的小伙伴可以自行查找资料学习。MPU6050提供了一个DMP的库，底层算法使用了四元数，使用它只要进行驱动的配置和移植就可以获得角度。卡尔曼滤波和互补滤波都是拿MPU6050的原始数据进行计算，需要自己写算法代码，有一定难度。当然现在很多现成的，不懂也可以拿来用，难度已经降低了不少。

（配图 四元数、滤波算法等）

下面我们来看一下MPU6050的驱动移植。

DMP驱动包括6个文件，里边包括了dmp算法的固件和api，我们不需要关心这些算法、函数和api的具体实现，只需要了解它们的使用方法即可。官方是在MSP430平台做的测试，所以我们移植的时候需要将MSP430平台相关的头文件都去掉，如果需要，还要加上新平台的头文件。

我们打开inv\_mpu.c文件，在开始可以看到一段注释，这里就告诉我们移植需要完成的所有工作，只需按照这个顺序做就可以了。

（配图 ）

我们需要定义和实现i2c\_write、i2c\_read、delay\_ms、get\_ms、reg\_int\_cb、labs、fabsf和min函数，DMP库的其他函数就是调用的这些接口。get\_ms、reg\_int\_cb分别是获取时间戳和回调函数，如果用不到可以不定义；labs是标准库函数；fabsf和fabs数学库中有，前者参数是单精度浮点，后者是双精度浮点，这里我们需要的是单精度浮点数。最终它们的定义如下图所示：

（配图）

接下来我们就需要实现具体的函数，我不需要时间戳，所有这里只需实现i2cWrite、i2cRead和delay\_ms。

读写函数的参数列表是确定了的，所以函数原型就是确定的，我们不能随便定义，否则会在编译的时候提示错误。参数列表分别是从机地址、寄存器、数据长度和数据内容。地址使用7位地址模式。

（配图）

这里先使用ioctl函数设置从机地址，然后调用i2c读写函数读写数据。关于树莓派i2c的具体内容可以查阅我的另一篇专栏（附专栏地址）。

延时毫秒函数使用usleep函数实现，usleep是微妙函数，乘以1000实现毫秒延时。（配图）

有了i2c读写函数和延时函数，读取mpu6050的数据就不成问题了，下面要实现的就是读取mpu6050的ID确保通信正常；配置mpu6050，设置它的时钟、设置加速度和陀螺仪的量程等等。之后调用dmp库的api初始化mpu和dmp并读取dmp数据解算姿态角即可。（配图）

ROLL表示横滚角，PITCH表示俯仰角，还有一个YAW表示偏航角，统称为欧拉角，这些定义来自于飞行器。小车可以选择ROLL或者PITCH来保持平衡。（配图）

具体代码可以自行下载驱动研究。

另外，移植过程中肯定会遇到问题，给大家几个调试的参考方法：

1. 网上查找错误信息。很多错误其他人已经遇到过，搜索错误信息将会找到很多相同的问题，在这些问题下可能就会有解决办法；
2. 添加log。在出现问题的代码前后添加log打印，将自己认为可能出问题的数据打印出来看看，通过分析找出问题所在；
3. 注释掉不相干的代码。将与问题不相干的代码去掉，只针对一个点进行测试，这样可以集中精力解决单一问题，避免其他因素的干扰；
4. 使用gdb。gdb是超级强大的调试工具，如果会使用的话会让程序的调试工作效率大大提高。所以建议不会的也学学，可以达到事半功倍的效果；

很多错误非常隐蔽，需要非常有耐心才行，有些错误的提示信息跟实际的问题可能差十万八千里，一定要从错误的表面看到错误的本质。

OK，关于MPU6050的移植就说这么多吧，这里只是给大家一个思路，具体还需要大家实操。

限于本人水平，肯定有遗漏和错误，欢迎大家交流指正。