为什么要用树莓派做平衡车呢？

原因主要有下面几点

1. 我不忍心再让它吃灰了；
2. 我想学习linux的应用开发；
3. 我想学习linux的驱动开发；
4. 树莓派可玩性和可扩展性高；
5. 切实体会下嵌入式linux开发与裸机开发的区别，对比一下它们的优缺点；
6. 看看自己到底能不能做出个东西来。

基于以上原因，我拿出了尘封已久的树莓派开干了。（配图 树莓派3b）

材料清单：

树莓派（3，4，zero应该都行）

MPU6050传感器

TB6612FNG或者L298N电机驱动（两个我都试过，推荐前者）

电机、轮胎和车架（这个某宝有卖，130块，电机一定要带编码器）

5v稳压电源（电流最好2A以上）

12V锂电池

面包板（洞洞板）

排针排座

导线、杜邦线

焊接工具（可选，烙铁、焊锡、松香等等）

Oled显示屏（可选）

超声波传感器（可选）

自己画电路板（可选，好处是整洁紧凑，没有太多连线，稳定可靠）

其他（可选，开关、按键、固定柱、螺丝螺母、电源插座等，这个看实际情况安排）

程序开发：

1. 调通传感器mpu6050
2. 控制电机
3. 获取编码器信号和测速
4. 编写pid控制代码
5. 调通oled
6. 控制超声波
7. 调通蓝牙
8. 调整代码逻辑，优化代码

程序逻辑：

通过制作平衡车，可以学习掌握或者了解的知识点有：

1. linux gpio控制
2. linux i2c
3. linux spi
4. linux 中断
5. linux字符设备驱动
6. linux模块开发
7. Linux蓝牙控制
8. Linux线程
9. Linux信号
10. Linux的系统api，其中有open、read、write、lseek、ioctl、select、poll、gettimeofday、usleep、sleep、mmap等等
11. 数字pid的实现
12. Pid控制参数整定
13. 滤波算法——卡尔曼、一阶互补
14. Makefile编写
15. 驱动移植的方法
16. 电机方向和速度（pwm）的控制方法
17. Linux gdb调试程序的方法
18. 如果自己画板子，电路板原理图与pcb绘制，电路板焊接

整个项目都使用c语言开发，需要对c语言比较熟悉。个人感觉平衡车还算是一个比较综合的项目，资料丰富，难度适中，很适合学了一堆模块却不知道做个什么小项目的小伙伴。本文主要给大家一个大概的印象和思路，具体的细节我会分几篇专栏详细讲解，欢迎大家一起学习、交流。