

Linux的操作系统I2C驱动架构解说 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022_Linux_E7_9A_84_E6_93_c103_253081.htm 1.Linux的I2C驱动架 Linux中I2C总

线的驱动分为两个部分，总线驱动（BUS）和设备驱动（DEVICE）。其中总线驱动的职责，是为系统中每个I2C总线增加相应的读写方法。但是总线驱动本身并不会进行任何的通讯，它只是存在在那里，等待设备驱动调用其函数。设备驱动则是与挂在I2C总线上的具体的设备通讯的驱动。通过I2C总线驱动提供的函数，设备驱动可以忽略不同总线控制器的差异，不考虑其实现细节地与硬件设备通讯。

1.1. 总线驱动 在系统开机时，首先装载的是I2C总线驱动。一个总线驱动用于支持一条特定的I2C总线的读写。一个总线驱动通常需要两个模块，一个struct i2c_adapter和一个struct i2c_algorithm来描述：

```
static struct i2c_adapter  
pb1550_board_adapter = { name: "pb1550 adapter", id:  
I2C_HW_AU1550_PSC, algo: NULL, algo_data: amp.au1550_algo.  
这个样例给上述总线驱动增加了读写“算法”。通常情况下每个I2C总线驱动都定义一个自己的读写算法，但鉴于有些总线使用相同的算法，因而可以共用同一套读写函数。本例中的驱动定义了自己的读写算法模块，起名叫“Au1550 algorithm”。全部填妥后，通过调用
```

```
: i2c_add_adapter(i2c_adap). 将这两个模块注册到操作系统里，总线驱动就算装上了。对于AMD au1550，这部分已经由AMD提供了。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com
```