

一、 引言

在嵌入式开发中软件分层设计应用的非常普遍，它是一个评判项目设计好坏的重要指标，也是程序员设计能力高低的重要表现。

一个好的项目通常是分层明确且耦合度低。这样设计有助于降低程序设计的复杂度，便于项目后期的功能维护和扩展。

也有利于团队开发，不同的开发者开发相应的层，完成后给出模块的接口，最后组合到一起整体测试，这样会使得项目开发效率非常的高，从而降低成本。

二、 案例分析

下面我们来分析一个STM32程序案例，它是基于freeRTOS编写的。如图1所示。

项目整体分为三个部分：

1. freeRTOS系统源码。
2. 外部硬件接口（port）和外部硬件访问驱动（drivers）。
3. 各种应用实现（APP）。

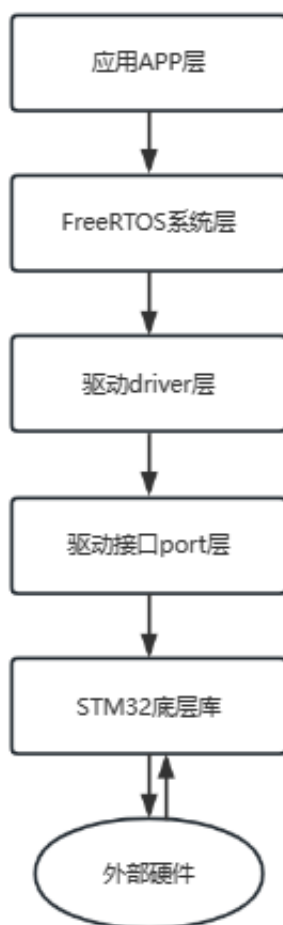


图2.系统框图

三、实例解析

就拿AT24C02存储驱动举例。

它分为两部分：

1. 对AT24C02的业务逻辑操作，包括读、写、擦除等（driver层）。
2. 对AT24C02的通信操作，也就是IIC通信（port层）。

port层是模拟IIC和所需硬件的初始化，如图3所示：

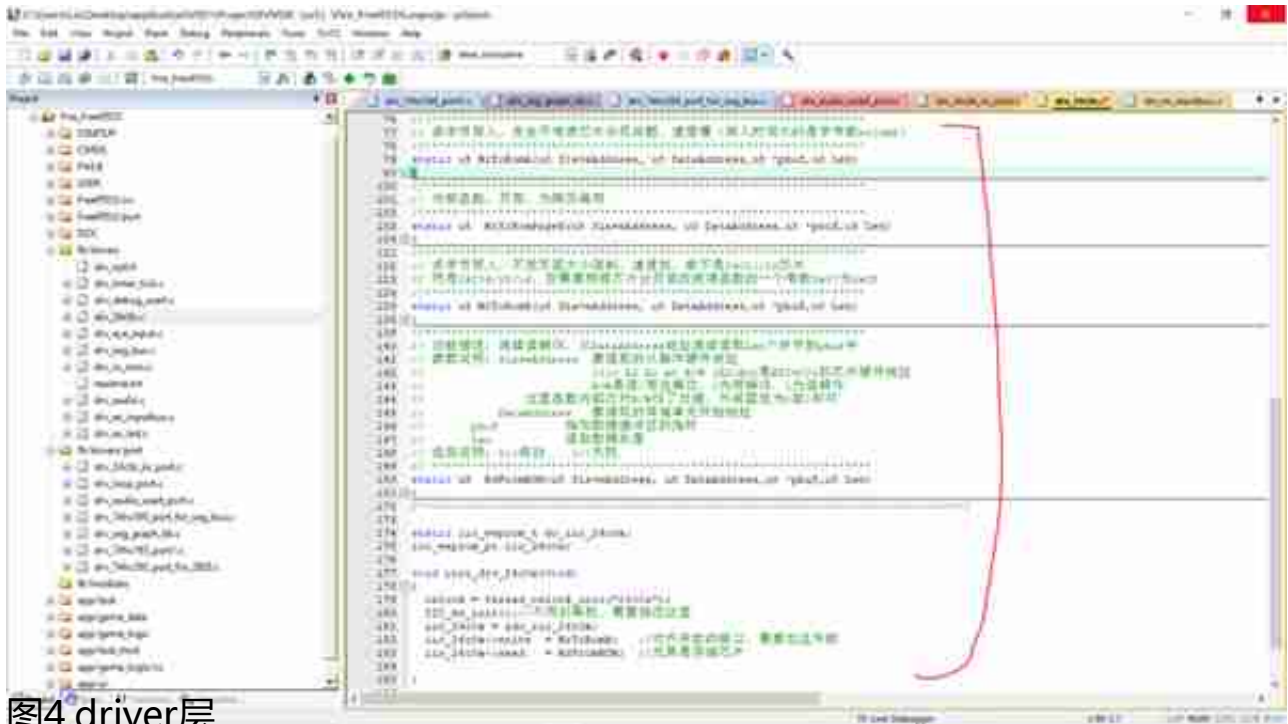


图4.driver层

抽象出的接口类：

```
typedef struct{ u8 (*write)(u8 SlaveAddress, u8 DataAddress ,u8 *pbuf,u8 Len); u8 (*read)(u8 SlaveAddress, u8 DataAddress,u8 *pbuf,u8 Len);}iic_eeprom_t,*iic_eeprom_pt;void init_drv_24c0x(void){ cslock = thread_cslock_init("24c0x"); IIC_hw_init();//????????????? iic_24c0x = &do_iic_24c0x; iic_24c0x->write = WrToRomB; //????????????????? iic_24c0x->read = RdFromROM; //?????????}
```

四、优点：

它的优点就很明显，使用时只需要创建一个iic_eeprom_t类的对象，初始化完成后就可以使用这个存储芯片。如果要换硬件只需要更改port层的管脚定义就好了，驱动层就补用管了。