# 一、I2C驱动框架

## 1.1 裸机下的I2C驱动框架

 首先编写IIC控制器驱动，bsp\_i2c.c和bsp\_i2c.h为IIC外设驱动。向外提供i2c\_master\_transfer函数。

 不管是什么IIC芯片，

 bsp\_3216c.c和bsp\_3216c.h为IIC设备驱动

## 1.2、Linux下的I2C驱动框架

## 1.3、I2C总线驱动

 I2C适配器在内核里面使用i2c\_\_adapter结构体，IIC适配器驱动(控制器)核心就是申请i2c\_adapter结构体，然后初始化，最后注册！

 初始化完成i2c\_adapter以后，使用i2c\_add\_adapter或者i2c\_add\_numbered\_adapter来向内核注册I2C控制器驱动。

 在i2c\_adapter里面有个非常重要的成员变量：i2c\_algorithm，次变量包含了IIC控制器访问IIC设备的API接口函数，需要I2C适配器编写人员实现。

 i2c\_adapter

 -》 i2c\_algorithm

 -> master\_xfer, 次函数就是I2C控制器最终进行数据收发的函数。

 经过查找，I.MX6U的I2C适配器驱动是i2c-imx.c，驱动与设备匹配成功以后，i2c\_imx\_probe

 NXP创建了一个imx\_i2c\_struct结构体，包含I.MX6U的I2C相关属性，这个结构体里面就有i2c\_adapter

struct imx\_i2c\_struct {

 **struct i2c\_adapter adapter;**

 struct clk \*clk;

 void \_\_iomem \*base;

 wait\_queue\_head\_t queue;

 unsigned long i2csr;

 unsigned int disable\_delay;

 int stopped;

 unsigned int ifdr; /\* IMX\_I2C\_IFDR \*/

 unsigned int cur\_clk;

 unsigned int bitrate;

 const struct imx\_i2c\_hwdata \*hwdata;

 struct imx\_i2c\_dma \*dma;

};

 设置i2c\_adapter下的i2c\_algorithm为i2c\_imx\_algo！

static struct i2c\_algorithm i2c\_imx\_algo = {

 .master\_xfer = i2c\_imx\_xfer,

 .functionality = i2c\_imx\_func,

};

 通过I.MX6U的I2C控制器读取IIC或者向IIC设备写入数据的时候最终是通过i2c\_imx\_xfer函数来完成的，

## 1.4、I2C设备驱动

 i2c\_client：表示I2C设备，不需要我们自己创建i2c\_client，我们一般在设备树里面添加具体的I2C芯片，比如fxls8471，系统在解析设备树的时候就会知道有这个I2C设备，然后会创建对应的i2c\_client，

 i2c设备驱动框架，i2c­­\_driver初始化与注册，需要II2C设备驱动编写人员编写的，IIC驱动程序就是初始化i2c\_driver，然后向系统注册。注册使用i2c\_register\_driver、i2c\_add\_driver，如果注销i2c\_driver使用i2c\_del\_driver

## 1.5、添加I2C设备信息

 在设备树中添加，I2C设备挂到那个I2C控制器下就在那个控制器下添加对应的节点：

 i2c\_transfer函数进行I2C数据传输。

# 二、驱动编写与测试

 在I2C1上接了一个AP3216C，UART4\_RXD作为I2C1\_SDA，UART4\_TXD作为I2C1\_SCL

 1、修改设备树，IO、添加AP3216C设备节点

 2、编写驱动框架，I2C设备驱动框架，字符设备驱动框架。

 3、初始化AP3216C，实现ap3216c\_read函数。

 重点就是通过IIC控制器来向AP3216C里面发送或者读取数据。这里使用i2c\_transfer这个api函数来完成IIC数据的传输。

int i2c\_transfer(struct i2c\_adapter \*adap, struct i2c\_msg \*msgs, int num)

 adap：IIC设备对应的适配器，也就是IIC接口，当IIC设备和驱动匹配以后，probe函数执行，probe函数传递进来的第一个参数就是i2c\_client，在i2c\_client里面保存了此I2C设备所对应的i2c\_adapter。

 msgs：就是构成的I2C传输数据。