#### ruct i2c client

unsigned short flags;

unsigned short addr;

char name[I2C\_NAME\_SIZE];

struct i2c\_adapter \*adapter;

struct i2c\_driver \*driver;

\_\_\_\_\_

struct list\_head detected;

struct device dev;

q;

struct device dev; int nr;

char name[48];

struct completion dev\_released;

struct mutex userspace\_clients\_lock;

struct list\_head userspace\_clients;

#### struct list\_head u

unsigned int class;

int (\*attach\_adapter) (struct i2c\_adapter \*) \_\_deprecated;

int (\*detach\_adapter)(struct i2c\_adapter \*) \_\_deprecated;

int (\*probe) (struct i2c\_client \*, const struct i2c\_device\_id \*);

int (\*remove)(struct i2c\_client \*);
void (\*shutdown)(struct i2c\_client \*);

int (\*suspend)(struct i2c\_client \*, pm\_message\_t mesg);

int (\*resume)(struct i2c\_client \*);

void (\*alert)(struct i2c\_client \*, unsigned int data);

int (\*command)(struct i2c\_client \*client, unsigned int cmd, void \*arg);
struct device\_driver driver;

const struct i2c\_device\_id \*id\_table;

int (\*detect)(struct i2c\_client \*, struct i2c\_board\_info \*);

const unsigned short \*address\_list;

struct list\_head clients;

#### struct i2c\_algor

struct module \*owner;

struct rt\_mutex bus\_lock;

const struct i2c\_algorithm \*algo;

unsigned int class;

void \*algo\_data;

int timeout;

int retries;

int (\*master\_xfer) (struct i2c\_adapter \*adap, struct i2c\_msg \*msgs, int num); int (\*smbus\_xfer) (struct i2c\_adapter \*adap, u16 addr,

u8 command, int size, union i2c smbus data \*data);

unsigned short flags, char read\_write,

u32 (\*functionality) (struct i2c\_adapter \*);

# 应用/外设 IIC2 IIC Client2 IIC Client1 IIC Driver1 IIC Driver2 IIC Core IIC adapter1 + Algorithm1 IC adapter2 + Algorithm2 IIC Control Driver1 IIC Control Driver2 控制层 BUS IIC Control Devcie1 IIC Control Devcie2 SOC **IIC Control**

# 应用层

最上层的文件节点,每个文件 节点可以当做一个IIC外设

# IIC 核心层

IIC Client 就等价于IIC Device, 通过名字与IIC Driver 匹配,有指针指向IIC adapter,挂在IIC Bus总线上

IIC Driver 是针对不同IIC外设写的驱动程序,调用IIC Core 提供的函数实现读写功能,挂在IIC Bus总线上

IIC adapter 会有指针指向传输的算法,IIC adapter 是控制层的驱动所创建的,算法的实现也是在控制层,挂在IIC Bus总线上

### IIC 控制层

IIC Control Driver 主要的功能包括寄存器的配置、中断注册、传输算法的实现和创建IIC adaper, 当做普通driver 挂在 platform Bus总线上

IIC Control Devcie 会传输一些信息给IIC Control Driver(中断号,寄存器基地址,频率,操作间隔), IIC Control Devcie 可以看是IIC 的channel, 当做普通device 挂在platform Bus总线上

# SOC 控制

IIC Control 是芯片内部的IIC控制模块,可以支持多个channel,每个channel会通过控制层的驱动抽象出一个adapter,通过adapter 的算法来实现数据的传输