RK3568 Linux的I2C驱动分析

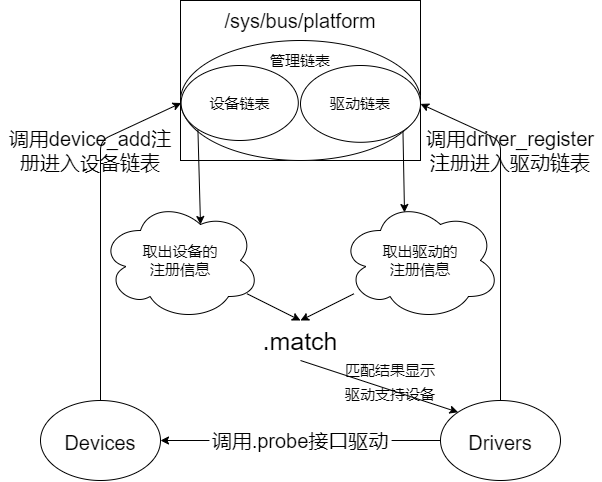
2022-08-09 17:26:37 +0800

# 1. 前言

为了提高代码的重用性，解决驱动代码和设备信息耦合的问题，Linux提出了总线、设备、驱动模型。

总线上管理着两个链表，分别管理设备和驱动。每当要向系统注册一个驱动或设备时，总线负责对新插入的驱动或设备进行匹配，根据匹配结果将新插入的驱动或设备添加到所对应的管理链表中。每当要从系统移除一个驱动或设备时，总线负责将其从所对应的管理链表中删除。

由于并不是所有的设备都能够归属于常见的总线(USB、PCI、I2C、SPI), Linux为了保持设备驱动的统一性，提出了一条虚拟的、抽象出来的总线 platform 。下图展示了Platform平台设备驱动框架的总线、设备、驱动模型。



platform总线、驱动、设备模型

# 2. Linux内核里I2C驱动的体系结构

## 2.1. 组成部分

《Linux设备驱动开发详解》一书第15章《Linux I2C核心、总线与设备驱动》将Linux的I2C体系结构分为核心、总线驱动和设备驱动三部分。

### 2.1.1. I2C核心

I2C核心提供了I2C总线驱动和I2C设备驱动注册和注销的方法，I2C通信方法中上层与具体适配器无关的代码，以及探测设备、检测设备地址的上层代码。

### 2.1.2. I2C总线驱动

I2C总线驱动是对I2C硬件体系结构中适配器端的实现。主要包含I2C适配器的数据结构 i2c\_adapter 、I2C适配器的通信方法(与设备之间)数据结构 i2c\_algorithm 和控制I2C适配器产生通信信号的函数。

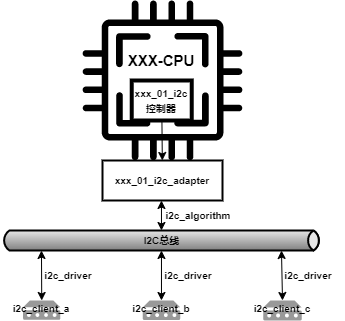
### 2.1.3. I2C设备驱动

I2C设备驱动(客户驱动)是对I2C硬件体系结构中设备端的实现。一般挂载在I2C适配器上。主要包含数据结构 i2c\_driver 和 i2c\_client 。

## 2.2. 体系结构

下图展示了Linux内核里I2C驱动的体系结构。CPU的每个I2C接口都配有一个I2C总线控制器(硬件)，负责控制所对应的一条I2C总线。I2C控制器可通过挂载在其他总线上被CPU控制，也可以直接集成在CPU内部。 i2c\_adapter 是在软件上抽象出来的I2C总线控制器(适配器)，与硬件一一对应。

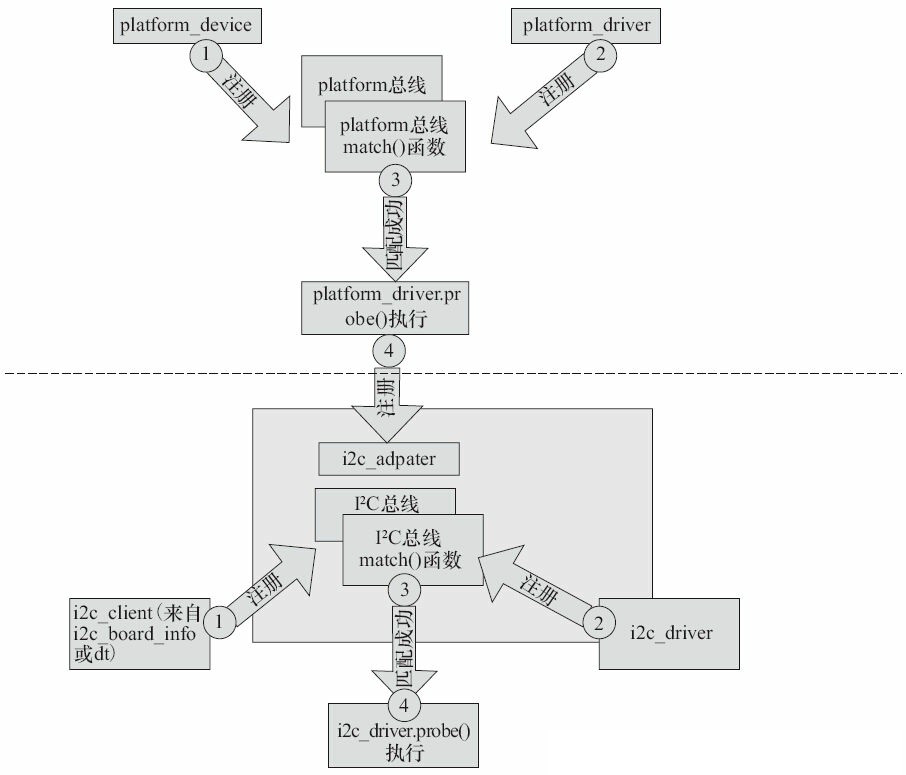
每个I2C设备( i2c\_device )在软件上抽象成 i2c\_client ，挂载到 i2c\_adapter 控制的I2C总线上后与设备驱动( i2c\_driver )匹配使用， i2c\_adapter 则通过根据控制逻辑实现的 i2c\_algorithm 与 i2c\_client 进行通信。



Linux内核里I2C驱动的体系结构

# RK3568 Linux的I2C驱动

RK平台的I2C总线控制器并非直接集成在内部，而是挂载到platform总线上，所以可以将RK3568 Linux的I2C驱动划分为两个部分实现： \* 采用platform驱动框架为I2C适配器设计驱动，以挂载到platform总线。 \* 采用I2C驱动框架为设备设计驱动，以挂载到I2C总线。



Linux内核里I2C驱动的设计框架