# 一、嵌入式网络简介

## 1.1、嵌入式网络硬件接口

### 1.1.1、什么是“网卡”？

### 1.1.2、嵌入式下网络方案

1、SOC 内部没有MAC

2、SOC 内部有SOC

## 1.2、MII/RMII接口

1、MII接口

2、RMII接口

## 1.3、MDIO接口

## 1.4、RJ45网络接口

100M和1000M接口区别。

## 1.5、I.MX6ULL ENET接口简介

# 二、PHY芯片详解

## 2.1、 PHY基础知识

## 2.2、LAN8720详解

# 三、Linux内核网络驱动框架

## 3.1、net\_device结构体

net\_device->const struct net\_device\_ops \*netdev\_ops; 网络设备的操作集

net\_device->onst struct ethtool\_ops \*ethtool\_ops;

构建net\_device，

1、申请net\_device。

net\_device，alloc\_netdev\_mqs.

对于以太网，申请函数就是alloc\_etherdev和alloc\_etherdev\_mqs。

alloc\_etherdev\_mqs函数是对alloc\_netdev\_mqs

alloc\_netdev\_mqs函数 有个参数为ether\_setup

对于can，alloc\_candev

## 3.2、net\_device\_ops结构体

ndo\_open和ndo\_stop。

ndo\_start\_xmit函数 用于最终的网络数据发送，

## 3.3、sk\_buff结构体

网络驱动里面很重要好的一个东西，保存网络数据，

要申请和释放。alloc\_skb或netdev\_alloc\_skb申请

kfree\_skb释放

1、发送数据dev\_queue\_xmit函数

2、netif\_rx函数接收

## 3.4、网络NAPI处理机制

# 四、I.MX6ULL 网络驱动简介

## 4.1、I.MX6ULL网络设备树

ENET1网络复位引脚为SNVS\_TAMPER7，ENET2为SNVS\_TAMPER8

## 4.2、I.MX6ULL网络驱动源码简析

ndev->netdev\_ops = &fec\_netdev\_ops;

ndev->ethtool\_ops = &fec\_enet\_ethtool\_ops;

netif\_napi\_add(ndev, &fep->napi, fec\_enet\_rx\_napi, NAPI\_POLL\_WEIGHT);

ret = devm\_request\_irq(&pdev->dev, irq, fec\_enet\_interrupt,

0, pdev->name, ndev);

## 4.3、MII\_BUS

fep->mii\_bus->read = fec\_enet\_mdio\_read;

fep->mii\_bus->write = fec\_enet\_mdio\_write;

struct fec\_enet\_private->mii\_bus

1、先使用mdiobus\_alloc申请mii\_bus。相当于fep->mii\_bus有效了

of\_mdiobus\_register函数的两个重要的功能，1个是通过mdiobus\_register来向内核注册mii\_bus，另一个是通过of\_mdiobus\_register\_phy向内核注册phy设备

调用phy\_device\_register函数向内核注册phy\_device

## 4.4、网络数据收发

4.4.1、数据发送

.ndo\_start\_xmit = fec\_enet\_start\_xmit,

4.4.2、数据接收

中断，fec\_enet\_interrupt，轮训为fec\_enet\_rx\_napi。

# 五、PHY驱动

phy\_driver为PHY驱动，注册phy\_driver

**通用的PHY驱动genphy\_driver**

# 五、网络驱动实验测试

# 六、单网卡使用