****

**Clock、Driver**

时间：2021-02-22

汇报人：邵荣峰

华捷艾米（上海）有限公司

目录

[目录 1](#_Toc64902495)

[版本记录 1](#_Toc64902496)

[1. Clock 1](#_Toc64902497)

[1.1 Clock 1](#_Toc64902498)

[1.2 Clock实例 2](#_Toc64902499)

[2. Driver 2](#_Toc64902500)

[2.1 设备 2](#_Toc64902501)

[2.2 驱动初始化与注册实例 3](#_Toc64902502)

版本记录

|  |  |  |
| --- | --- | --- |
| Date | Version | Tester |
| 2020.02.22 | V1.0 | 邵荣峰 |
|  |  |  |
|  |  |  |

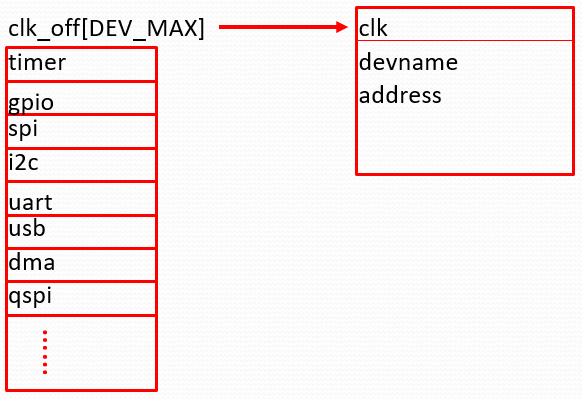
1. Clock
   1. Clock

（1）简介

目前针对Clock模块进行封装管理，各个模块的时钟后续都需要根据此接口进行调用，切勿单独另外实现操作，clock模块主要分三个部分：clock配置、模块clock初始化、clock使能。代码位于\drivers\clock目录下。

1. Clock框架

每个设备的clock模块都是挂在clk\_off[DEV\_MAX]数组下面，每个模块的clock通过结构体clk进行封装，有两个成员变量：devname（设备名称）、address（设备时钟基地址）



1. Clock配置

clock的配置通过clk\_confg（CLKCFG）结构体进行实现，目前还没有完善，此部分主要设置pll分频系数以及目标频率等，需要后续加入。

1. Clock模块初始化

此部分主要对clock配置进行操作初始化，函数为：

void clock\_module\_init(CLKCFG\* clk\_cf)

此部分还需后续进行完善。

1. Clock模块使能

通过clock\_enable(char\* name)函数对指定模块时钟进行初始化，函数返回的是此模块时钟基地址。

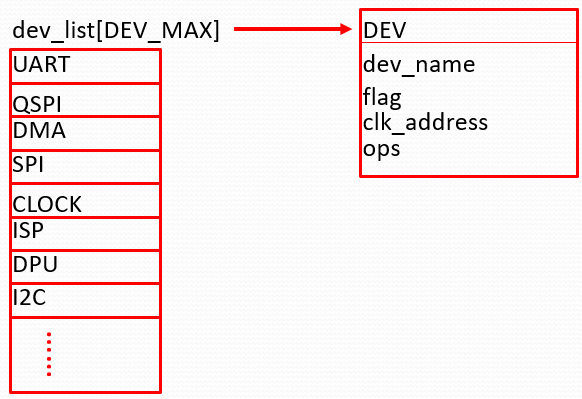
* 1. Clock实例

实例参见2.2节具体介绍。

1. Driver
   1. 设备

（1）device

系统中的设备通过dev\_list数组进行管理，所有的设备都应放到此数组中，具体框图如下：



dev\_list是DEV结构体类型（位于\include\imi\_type.h），其中有四个成员：dev\_name（设备名称）、flag（标志位，设备注册成功则为1）、clk\_address（设备时钟基地址）、ops（设备操作函数，指针类型，指向对应驱动的操作集）

（2）驱动初始化模块

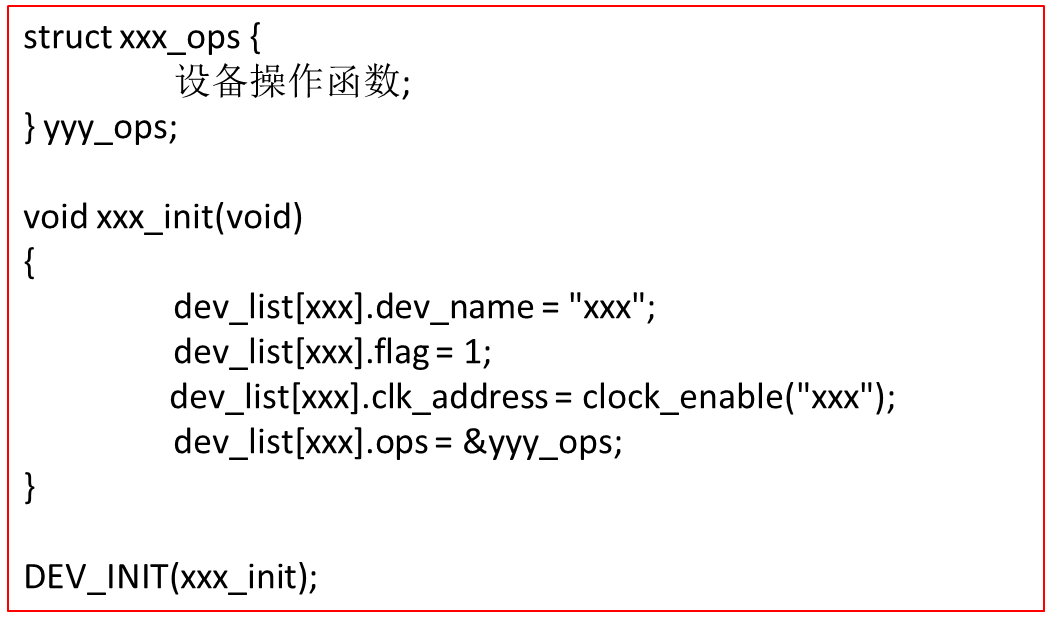
由于对驱动进行了封装，因此驱动初始化过程变得比较简单，主要针对四个成员进行对应初始化。后续各个模块设备具体操作函数都要通过结构体进行封装，然后通过成员ops指向该结构体。部分模块除此之外还需要另行增加额外初始化部分。

（3）驱动注册入口

目前FreeRTOS中的注册的所有device放在.text段中，系统启动时，程序会从段中查找并进行初始化，由于uart模块需要第一个初始化，因此放在.\_INIT\_DEV\_FIRST中，其他模块均放在.\_INIT\_DEV段下，通过DEV\_INIT（func）进行驱动注册。

* 1. 驱动初始化与注册实例

设备驱动程序架构如下：



**以ISP驱动为例：**

（1）、设置ISP对应clock

①在\drivers\clock\clock.h中增加针对isp的clock基地址，如下：

#define ISP\_C 0x23F00000

②对\drivers\clock\clock.c中的clk\_off设置针对isp的.devname以及.address，如下：

.devname = "isp",

.address = ISP\_C,

③clock的使能主要通过clock\_enable()函数进行使能，此处建议将clock使能在模块驱动初始化过程中进行，详细见下面流程。

（2）、ISP驱动初始化

①在\include\dev.h中添加设备列表序号（dev\_list列表中的序号），此处通过宏实现，ISP序号为5（部分驱动已经进行添加，需查看是否添加，若没有对应添加需自行依次添加），即如下：

#define ISP 5

②ISP驱动初始化

对dev\_list[ISP]的四个成员进行赋值，如下：

void isp\_init(void)

{

log\_d("isp init start.\n\r");

isp\_ops.IOCTL = isp\_t\_IOCTL;

dev\_list[ISP].flag = 1;

dev\_list[ISP].clk\_address = clock\_enable("isp");

dev\_list[ISP].ops = &isp\_ops;

}

注：设备初始化的时候需要将其flag标志位置1，即dev\_list[ISP].flag=1，后续可以通过此flag检查某驱动是否进行初始化；clock\_enable("isp")是对ISP的clock进行使能，函数返回值为ISP对应clock的基地址；针对isp对外操作的函数接口放在isp\_ops结构体下面，然后通过ops进行指向。

③驱动的注册

一般在对应驱动代码下通过DEV\_INIT()对驱动进行注册，如下：

DEV\_INIT(isp\_init);

以上即为ISP device的初始化与注册的过程。

华捷艾米（上海）有限公司