



StarFive
赛昉科技

昉·惊鸿-7110 DevKit电源管 理开发手册

版本：1.0

日期：2023/07/18

Doc ID: JH7110-DGEN-009

法律声明

阅读本文件前的重要法律告知。

版权注释

版权 ©上海赛昉科技有限公司，2023。版权所有。

本文档中的说明均基于“视为正确”提供，可能包含部分错误。内容可能因产品开发而定期更新或修订。上海赛昉科技有限公司（以下简称“赛昉科技”）保留对本协议中的任何内容进行更改的权利，恕不另行通知。

赛昉科技明确否认任何形式的担保、解释和条件，无论是明示的还是默示的，包括但不限于适销性、特定用途适用性和非侵权的担保或条件。

赛昉科技无需承担因应用或使用任何产品或电路而产生的任何责任，并明确表示无需承担任何及所有连带责任，包括但不限于间接、偶然、特殊、惩戒性或由此造成的损害。

本文件中的所有材料受版权保护，为赛昉科技所有。不得以任何方式修改、编辑或断章取义本文件中的说明，本文件或其任何部分仅限用于内部使用或教育培训。

联系我们：

地址：浦东新区盛夏路61弄张润大厦2号楼502，上海市，201203，中国

网站：<http://www.starfivetech.com>

邮箱：

- sales@starfivetech.com（销售）
- support@starfivetech.com（支持）

目录

表格清单.....	4
插图清单.....	5
法律声明.....	ii
前言.....	vi
1. 简介.....	7
1.1. 概述.....	7
1.2. 系统框图.....	7
1.3. 上电时序.....	7
2. 配置.....	11
2.1. 驱动和菜单配置.....	11
2.2. DTS配置.....	11
3. 接口介绍.....	13
3.1. of_regulator_match.....	13
3.2. devm_regmap_init_i2c.....	13
3.3. devm_regulator_register.....	14
4. 休眠和唤醒.....	15
4.1. Hibernation.....	15
4.2. 唤醒.....	15
5. 调试.....	17

表格清单

表 0-1 修订历史.....	vi
表 1-1 上电时序间隔.....	8



插图清单

图 1-1 系统框图.....	7
图 1-2 上电时序.....	8
图 1-3 推荐上电时序.....	10
图 5-1 示例输出.....	17
图 5-2 示例输出.....	17



StarFive
赛昉科技

前言

关于本指南和技术支持信息

关于本手册

本手册主要为SDK开发和移植提供赛昉科技新一代SoC平台——昉·惊鸿-7110电源管理模块的编程基础和调试操作。

受众

本手册主要服务于与电源管理相关驱动程序的开发人员。如果您正在开发其他模块，请与您的销售或支持顾问联系，获取我们在昉·惊鸿-7110上的完整文档。






修订历史

表 0-1 修订历史

版本	发布说明	修订
1.0	2023/07/18	预备版。

注释和注意事项

本指南中可能会出现以下注释和注意事项：

-  **提示：**
建议如何在某个主题或步骤中应用信息。
-  **注：**
解释某个特例或阐释一个重要的点。
-  **重要：**
指出与某个主题或步骤有关的重要信息。
-  **警告：**
表明某个操作或步骤可能会导致数据丢失、安全问题或性能问题。
-  **警告：**
表明某个操作或步骤可能导致物理伤害或硬件损坏。

1. 简介

昉·惊鸿-7110 DevKit电源管理子系统为用户提供了在各种电源模式和配置文件中配置设备的能力，用于功耗测量。用户可以选择在交互（简单）模式或使用高级模式下工作，他们可以在模式中更改源代码中的定义值，以控制应用程序的行为。本章主要是对功能管理子系统、框图和上电时序的介绍。

1.1. 概述

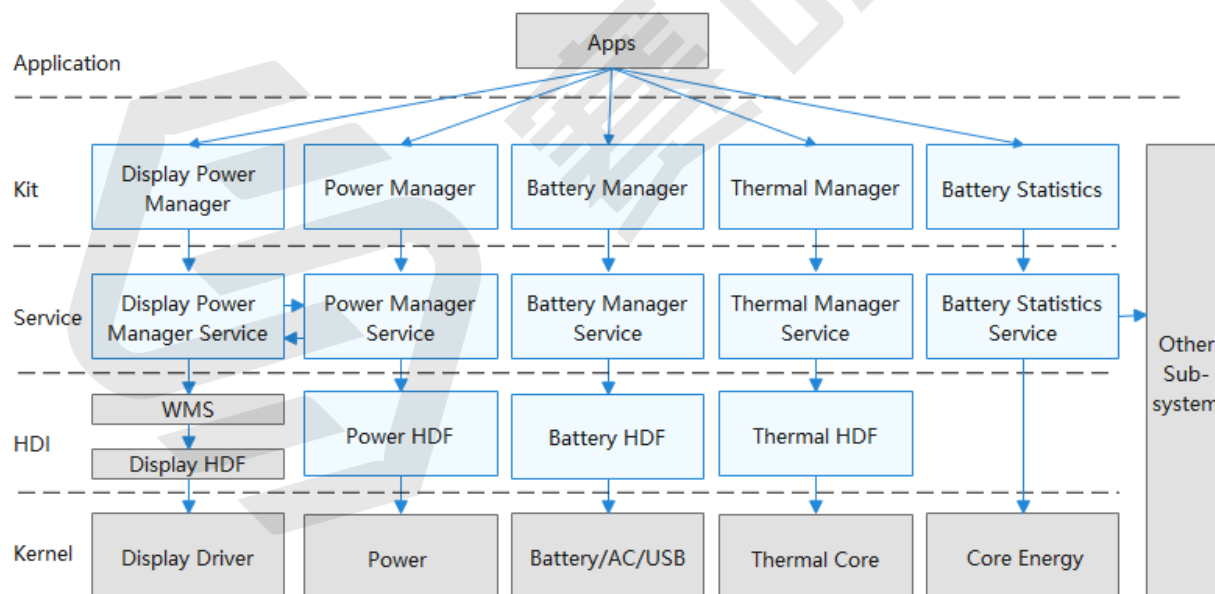
系统中的每个电源一般分为两种类型：DCDC（直流电）和LDO（低压差线性稳压器）。两种电源的总体特点如下：

- DCDC：具有效率高、输入电压范围宽、驱动能力强、静电流低等优点。
- LDO：低压差线性稳压器，为线性的稳压器，仅能使用在降压应用中。也就是输出电压必需小于输入电压。它具有稳定性好、负载响应快、输出纹波小等优点。

1.2. 系统框图

下图为电源管理子系统框图：

图 1-1 系统框图



1.3. 上电时序

昉·惊鸿-7110有3个电源，VDD、AVDL和AVDH。VDD和AVDL都是内核电源，而单独的Pin用于隔离噪声。

推荐的上电时序是从数字内核电压到模拟I/O电压，从低电压电平到高电压电平。该推荐不是限制，而是一个指导建议，因为这样可形成最佳运行情况，即通电期间保持最低的泄漏电流。

下图显示了昉·惊鸿-7110 昉·星光 2及EVB 不同电源组的推荐上电时序。

图 1-2 上电时序

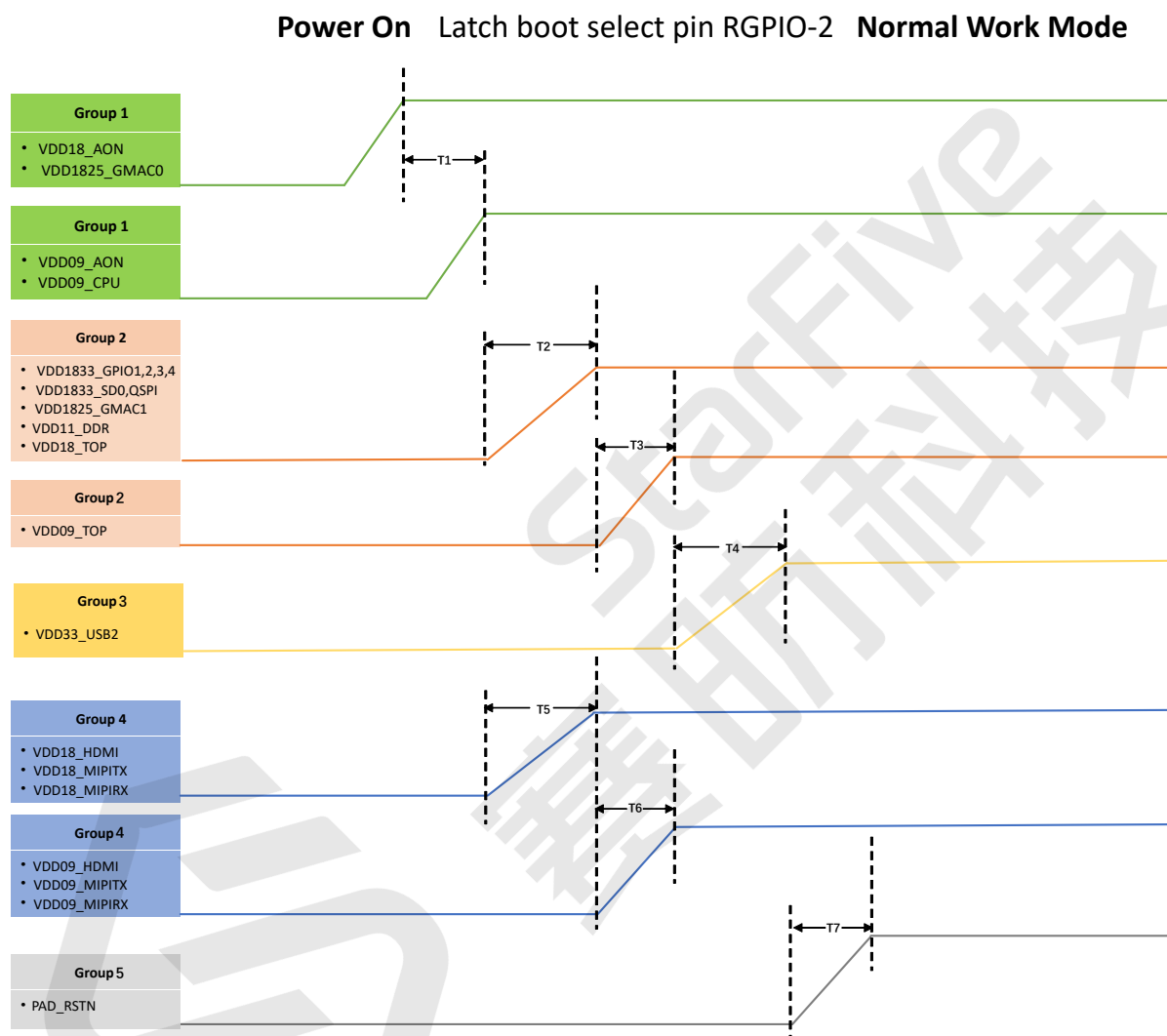


表 1-1 上电时序间隔

顺序	间隔
T1	100 us
T2	300-500 us
T3	50-100 us
T4	300-500 us
T5	300-500 us

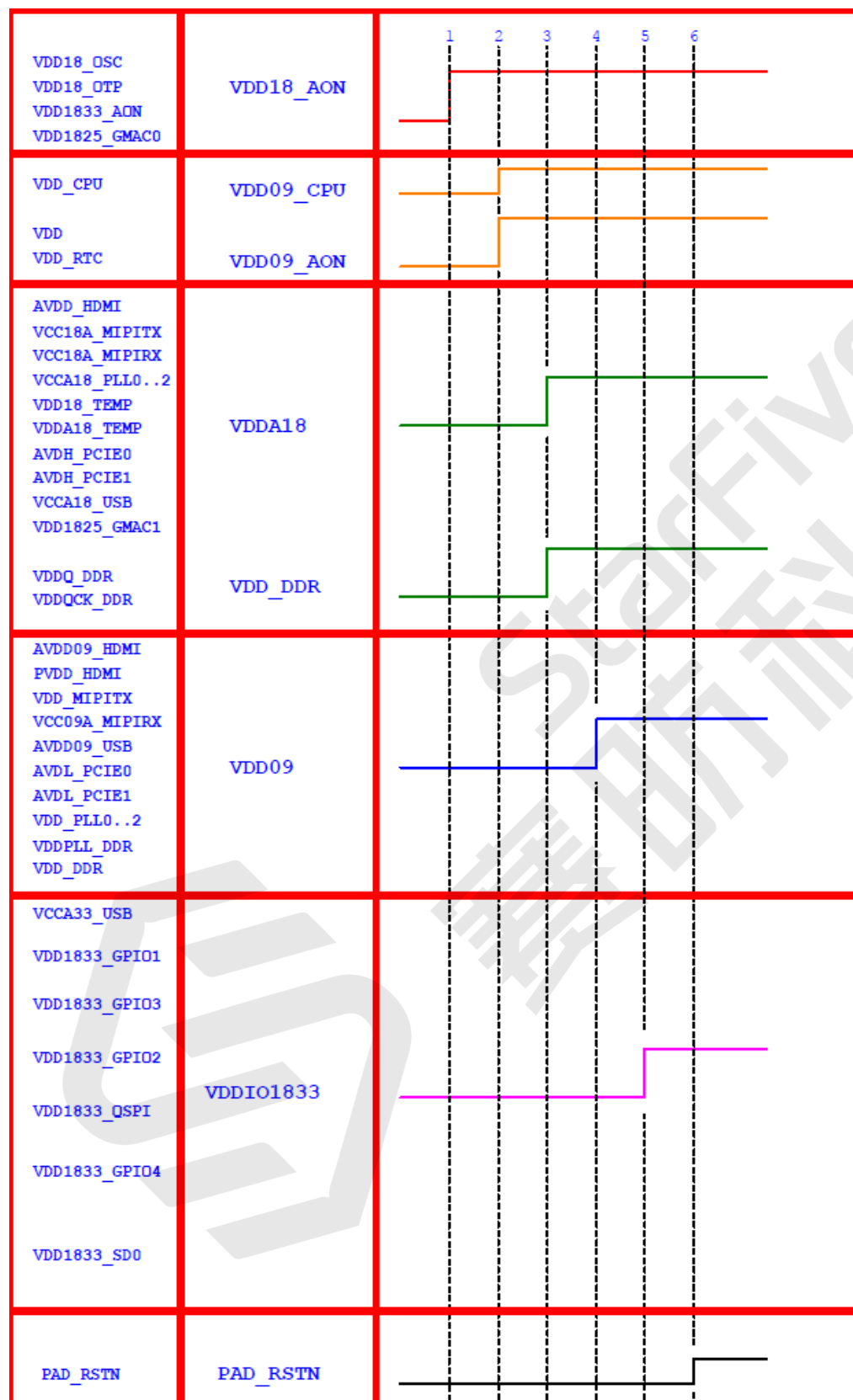
表 1-1 上电时序间隔 (续)

顺序	间隔
T6	50-100 us
T7	>10 ms

下图显示了昉·惊鸿-7110 DevKit不同电源组的推荐上电时序。



图 1-3 推荐上电时序



2. 配置

以下为昉·惊鸿-7110 DevKit电源管理子系统的配置信息。

2.1. 驱动和菜单配置

以下为昉·惊鸿-7110 DevKit的驱动文件、DTS参考文件和菜单配置中相应的宏配置：

- 驱动文件：

```
drivers/regulator/axp15060-regulator.c
```

- DTS文件可参考：

```
arch/riscv/boot/dts/starfive/jh7110-devkits.dts
```

- 菜单配置中相应的宏配置：

```
CONFIG_REGULATOR_APX15060
```

2.2. DTS配置

DTS配置包括：I2C mount、body、regulator、rtc和其他部分。以下代码块为&i2c5和pmic_axp15060: axp15060_reg@36的设备树源代码示例。

```
&i2c5 {
    clock-frequency = <100000>;
    i2c-sda-hold-time-ns = <300>;
    i2c-sda-falling-time-ns = <510>;
    i2c-scl-falling-time-ns = <510>;
    auto_calc_scl_lhcnt;
    pinctrl-names = "default";
    pinctrl-0 = <&i2c5_pins>;
    status = "okay";

    pmic_axp15060: axp15060_reg@36 {
        compatible = "starfive,axp15060-regulator";
        reg = <0x36>;

        regulators {
            cpu_vdd: DCDC2 {
                regulator-boot-on;
                regulator-always-on;
                regulator-compatible = "cpu_vdd";
                regulator-name = "cpu_vdd";
```

```
        regulator-min-microvolt = <500000>;
        regulator-max-microvolt = <1540000>;
    };
    emmc_vdd: BLDO3 {
        regulator-boot-on;
        regulator-always-on;
        regulator-compatible = "emmc_vdd";
        regulator-name = "emmc_vdd";
        regulator-min-microvolt = <1800000>;
        regulator-max-microvolt = <1800000>;
    };
    vcc_3v3: BLDO4 {
        regulator-boot-on;
        regulator-always-on;
        regulator-compatible = "vcc_3v3";
        regulator-name = "vcc_3v3";
        regulator-min-microvolt = <3300000>;
        regulator-max-microvolt = <3300000>;
    };
};
};
};
```

以下提供了对上述代码块中的参数说明。

- **clock-frequency:** 上述时钟的频率。
- **pinctrl-names:** pinctrl的名称。
- **status:** i2c模块的工作状态。
- **compatible:** 兼容性信息，用于连接驱动程序和目标设备。
- **reg:** i2c的寄存器地址。
- **regulators:** 需要被控制的regulator。
- **regulator-boot-on:** 由引导加载程序或固件启用的regulator。
- **regulator-always-on:** 禁止对该regulator进行断电操作。
- **regulator-min-microvolt:** 用户可设置的最小电压。
- **regulator-max-microvolt:** 用户可设置的最大电压。

3. 接口介绍

昉·惊鸿-7110 DevKit 为电源管理引擎提供以下应用接口。

3.1. of_regulator_match

该接口有以下参数。

- 简介:

```
int of_regulator_match(struct device *dev, struct device_node *node,
                      struct of_regulator_match *matches,
                      unsigned int num_matches);
```

- 描述: 在为provider进行探测的过程中, 匹配表的每个组件都将与实际使用的所有regulator进行比较, 并且所有的匹配项都将根据设备节点进行初始化。

- 参数:

- dev: 目标i2c设备
- node: Regulator DTS节点
- matches: 定义的regulator信息
- num_matches: 定义的regulator数量

- 返回值: 匹配的个数

3.2. devm_regmap_init_i2c

该接口有以下参数。

- 简介:

```
# define devm_regmap_init_i2c(i2c, config) \
__regmap_lockdep_wrapper(__devm_regmap_init_i2c, #config, \
i2c, config)
```

- 描述: 该接口用于映射PMIC地址。

- 参数:

- i2c: 目标i2c设备
- config: PMIC regmap信息

3.3. devm_regulator_register

该接口有以下参数。

- 简介:

```
struct regulator_dev *devm_regulator_register(struct device *dev,  
const struct regulator_desc *regulator_desc,  
const struct regulator_config *config);
```

- 描述: 注册regulator

- 参数:

- **dev:** 目标i2c设备
- **regulator_desc:** PMIC提供的regulator信息
- **config:** regulator的配置信息

- 返回值:

- **成功:** 0
- **失败:** 错误代码

4. 休眠和唤醒

本章介绍了昉·惊鸿-7110 DevKit的休眠模式和唤醒方法。昉·惊鸿-7110的休眠为hibernation，即在休眠时将运行状态数据保存在系统的disk中，然后选择合适的方式关闭或重启系统。

4.1. Hibernation

执行以下步骤进入休眠模式：

1. 在昉·惊鸿-7110 DevKit上插入eMMC。
2. 联网下载image.fit文件到昉·惊鸿-7110 DevKit。
3. 内核启动后配置swap分区。
4. 通过执行以下命令启动hibernation：

```
echo disk > /sys/power/state
```

5. 进入休眠模式。

4.2. 唤醒

Hibernation为最省电的休眠方式，在休眠时会关闭所有电源，因此想唤醒昉·惊鸿-7110 DevKit，唯有power键可用。

执行以下步骤唤醒：

1. 在昉·惊鸿-7110 DevKit进入休眠模式并关闭后，联网使用相同的image.fit文件启动开发板。
2. 检查休眠恢复后是否可以恢复内核镜像。

以下为唤醒函数的调用过程：

```
|--late_initcall_sync(software_resume)
| |--swsusp_check # Check is the image is valid of not from
| |swsusp_resume_device device.If it is invalid, the following process
| |will not be excuted, you can exit and proceed with the normal startup
| |process.
| | |--pm_prepare_console
| | |--__pm_notifier_call_chain
| | |--freeze_processes # Freeze the processes of the current user space
| | |--load_image_and_restore
| | | |--lock_device_hotplug
| | | |--create_basic_memory_bitmaps
| | | |--swsusp_read
```

```
| | |--swsusp_close
| | |--hibernation_restore # Restore from the hibernate image
| | | |--resume_target_kernel
| | |--swsusp_free
| | |--free_basic_memory_bitmaps
| | |--unlock_device_hotplug
| |--thaw_processes # Thaw the processes of the user sapce.And at this
point, the process being thawed and restored is a process in
Hibernate.
| |--__pm_notifier_call_chain
| |--pm_restore_console
| |--swsusp_close
```

5. 调试

PMIC中涉及的驱动程序使用并不复杂，重点反映在最终的寄存器设置中。因此，常用的调试方法是通过以下i2c直接检查昉·惊鸿-7110 DevKit的寄存器：

- 读寄存器：

```
I2cget -f -y 5 0x36 addr
```

图 5-1 示例输出

```
# i2cset -f -y 5 0x36 0x1b 0x10
```

- 写寄存器：

```
I2cset -f -y 5 0x36 addr data
```

图 5-2 示例输出

```
# i2cget -f -y 5 0x36 0x1b  
0x10
```