

Rockchip

PWM 开发指南

发布版本: 1.0

日期: 2017.02

前言

概述

本文档主要介绍 Rockchip 处理器内 pwm 逻辑模块的功能、常用的接口和内部工作原理，通过背光实例介绍 pwm 模块的开发过程以及注意事项。

产品版本

芯片名称	内核版本
RK3328	3.10
RK3228H	3.10

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

日期	版本	作者	修改说明
2017.02.17	V1.0	wdc	增加 pwm 背光文档说明

目录

1	PWM 概述	1-1
2	驱动文件与 DTS 节点	2-1
3	PWM 连续模式使用配置	2-1
	3.1 背光 DTS 节点.....	2-1
	3.2 背光配置 PWM	2-2

1 PWM 概述

脉冲宽度调制器 (PWM) 是很常见的嵌入式系统一种功能。它提供了生成为脉冲周期波形的控制方法或可以作为数字模拟与一些外部组件的转换器。

RK PWM 脉宽调制模块支持下列功能：

- 4 通道；
- 可配置成 capture 模式；
- 可配置输入波形的高/低极性有效周期；
- 可生成中断在输入波形极性的转折处；
- 32 位高极性捕获寄存器
- 32 位低极性捕获寄存器
- 32 位当前值寄存器
- 可配置成连续模式或一次性模式运行；
- 32 位周期计数器
- 32 位责任登记册
- 32 位当前值寄存器
- 8 位重复计数器为一次性操作的，一次性操作会产生 $N + 1$ 波形， N 是重复计数器的值，在一次性模式运行结束后，生成一个中断。
- 连续模式，不断产生的波形，并不会产生任何中断。

2 驱动文件与 DTS 节点

PWM 驱动文件及 DTS 节点:

drivers/pwm/pwm-rockchip.c

DTS 节点:

```
pwm0: pwm@ff1b0000 {
    compatible = "rockchip,rk-pwm";
    reg = <0x0 0xff1b0000 0x0 0x10>;
    #pwm-cells = <2>;
    pinctrl-names = "default";
    pinctrl-0 = <&pwm0_pin>;
    clocks = <&clk_gates2 8>, <&clk_gates16 6>;
    clock-names = "pwm", "pclk";
    status = "disabled";
};PWM 连续模式使用配置
```

2.1 背光 DTS 节点

PWM 的连续模式使用最多，且以背光使用最为频繁，下面以背光为例子来说明 PWM 如何在代码中使用，以下是背光的 DTS 节点:

```
backlight: backlight {
    compatible = "pwm-backlight";
    pwms = <&pwm0 0 25000>;
    brightness-levels = <
        0  1  2  3  4  5  6  7
        8  9 10 11 12 13 14 15
        16 17 18 19 20 21 22 23
        24 25 26 27 28 29 30 31
        32 33 34 35 36 37 38 39
        40 41 42 43 44 45 46 47
        48 49 50 51 52 53 54 55
        56 57 58 59 60 61 62 63
        64 65 66 67 68 69 70 71
        72 73 74 75 76 77 78 79
        80 81 82 83 84 85 86 87
        88 89 90 91 92 93 94 95
        96 97 98 99 100 101 102 103
        104 105 106 107 108 109 110 111
        112 113 114 115 116 117 118 119
        120 121 122 123 124 125 126 127
        128 129 130 131 132 133 134 135
        136 137 138 139 140 141 142 143
        144 145 146 147 148 149 150 151
        152 153 154 155 156 157 158 159
        160 161 162 163 164 165 166 167
```

```

168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183
184 185 186 187 188 189 190 191
192 193 194 195 196 197 198 199
200 201 202 203 204 205 206 207
208 209 210 211 212 213 214 215
216 217 218 219 220 221 222 223
224 225 226 227 228 229 230 231
232 233 234 235 236 237 238 239
240 241 242 243 244 245 246 247
248 249 250 251 252 253 254 255>;
default-brightness-level = <200>;
enable-gpios = <&gpio1 13 GPIO_ACTIVE_HIGH>;
};

```

2.2 背光配置 PWM

接下来描述如何配置背光参数来调节 PWM 的波形：

A. "pwms = <&pwm0 0 25000>", 很明显看出背光是使用 pwm0 来调节暗度；其中第二个参数“0”表示 index 为 0，因为 pwm0 下面只有 1 个 PWM，所以我们的平台下面都是填 0；而第三个参数“25000”表示为周期为 25000ns，所以频率就是 40k，其他频率配置类似计算。

需要注意的是这里一定要有 3 个参数，否则会出错，无法获取到 PWM 数据。

B. brightness-levels 数组，我们一般以值 255 为一个 scale，所以一般的 brightness-levels 为一 256 个元素的数组；当 PWM 设置为正极性时，从 0~255 表示背光为正极，占空比从 0%~100% 变化，255~0 位负极性，占空比从 100%~0% 变化；当 PWM 设置为负极性时，反之。

当然很多时候有些屏不是一定从 0%~100%，发生黑到亮的变化，比如占空比到达 20% 的时候，屏幕就全黑了，或者占空比到达 80% 的时候，就已经够亮了，那么这时候还把数组配置成 0~255 就不合适，可能你需要只是 50~200 的一个范围，那么这时候应该怎么配置呢？

因为需要 255 为一个 scale，而这个 scale 等于 brightness-levels 数组里面元素的最大值，所以 brightness-levels 数组里一定要有 255 这个数值；

先保证 PWM 的极性一定要为负极性，然后可以将第一个元素改成 255，因为休眠唤醒的时候，会调用第 0 个元素，能把背光关闭；而且为了凑成 256 个元素，需要均匀的重复某些值，我们可以更改成如下的数组：

```

brightness-levels = <
255 180 180 179 178 178 177 177
176 176 175 174 174 173 173 172
172 171 171 170 170 169 169 168
168 167 167 166 166 165 165 164
164 163 163 162 162 161 161 160
160 159 159 158 158 157 157 156
156 155 155 154 154 153 153 152
152 151 151 150 150 149 149 148
148 147 147 146 146 145 145 144
144 143 143 142 142 141 141 140
140 139 139 138 138 137 137 136
136 135 135 134 134 133 133 132
132 131 131 130 130 129 129 128

```

```

128 127 127 126 126 125 125 124
124 123 123 122 122 121 121 120
120 119 119 118 118 117 117 116
116 115 115 114 114 113 113 112
112 111 111 110 110 109 109 108
108 107 107 106 106 105 105 104
104 103 103 102 102 101 101 100
100 99 99 98 98 97 97 96
96 95 95 94 94 93 93 92
92 91 91 90 90 89 89 88
88 87 87 86 86 85 85 84
84 83 83 82 82 81 81 80
80 79 79 78 78 77 77 76
76 75 75 74 74 73 73 72
72 71 71 70 70 69 69 68
68 67 67 66 66 65 65 64
64 63 63 62 62 61 61 60
60 59 59 58 57 56 55 55
54 54 53 53 52 52 51 51>;

```

C. **default-brightness-level** 表示默认的背景光，它存在于开机时候，背光驱动初始化到安卓用户设置下来新的背光这段时间，表示为第 200 个元素的背光亮度。

D. **enable-gpios** 表示背光使能脚，这个根据电路原理图配置即可；有的硬件没有这个背光使能脚，那么将这个配置删除，背光驱动通过配置 **brightness-levels** 数组的第 0 个元素将背光关闭。

E. 如何确定背光灭的 **brightness-level** 值，通过命令行调试背光亮度， `echo xxx > sys/class/backlight/rk28_bl/brightness`。

当 PWM 设置为正极性时，可以通过 `echo xxx > sys/class/backlight/rk28_bl/brightness` 到背光节点，xxx 的范围为 0~255，这时观察亮度变化，如 x 为背光亮度为 0 的点，y 为客户接受的背光最亮的点。这时重新调整 **brightness-level** 表，就可以将数组第一个值改为 x，最大值改为 y，中间值需均匀变化，凑成 256 个元素，且有一个元素值为 255。当 PWM 为负极性时，则反之。