

# 目录

- PWM 的参数: .....2
- Kernel 的配置: .....2
  - dts 的配置: .....2
  - 客户硬件举例: .....3
- PWM 的架构.....4
- User mode 对 PWM 的控制: .....5

## PWM 的参数:

Duty\_cycle:

占空比。

Echo 25 > duty\_cycle 表示占空比是 25%。

Period:

Frequency。

Echo 2000 > period

表示 2K HZ 的 frequency 的 pwm 波。

Enable/disable:

使能;

Polarity:

极性。

如果是 normal。那么 duty\_cycle=25%，表示高电平占的比例是 25%。

如果是 inverse，那么就反之。

## Kernel 的配置:

### dts 的配置:

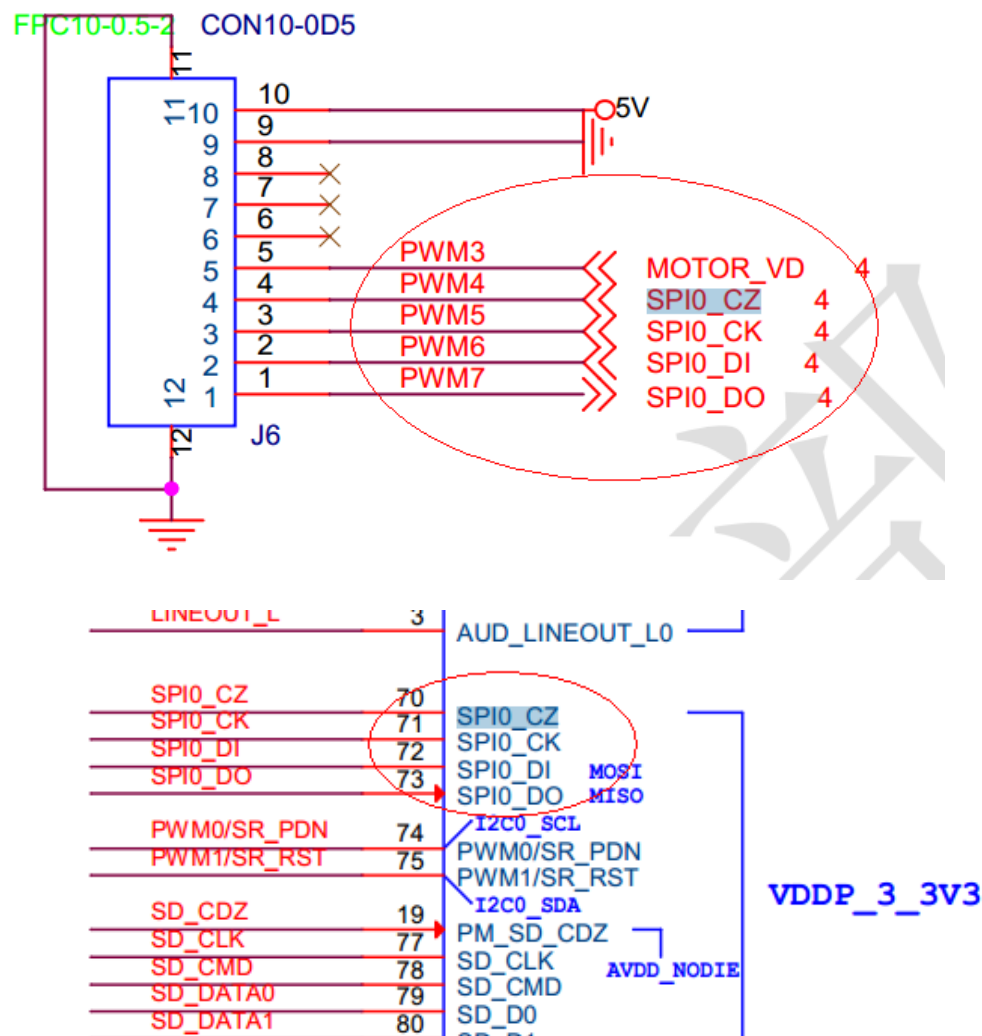
```
pwm {
    compatible = "sstar,infinity-pwm";
    reg = <0x1F003400 0x600>;
    clocks = <&CLK_xtali_12m>;
    npwm = <11>;
    pad-ctrl = <PAD_PWM0 PAD_PWM1 PAD_UNKNOWN PAD_UNKNOWN PAD_UNKNOWN PAD_UNKNOWN PAD_UNKNOWN PAD_UNKNOWN PAD_UNKNOWN PAD_UNKNOWN>;
    status = "ok";
}
```

npwm: 11; 表示 pwm 有 11 组;

Pad-ctrl: 由于是共享 gpio 的。所以当 gpio 被当作 pwm 使用的时候,需要做切 pad 动作。

上述: PAD\_PWM0 / PAD\_PWM1 表示只使能 PWM0/PWM1。

客户硬件举例：

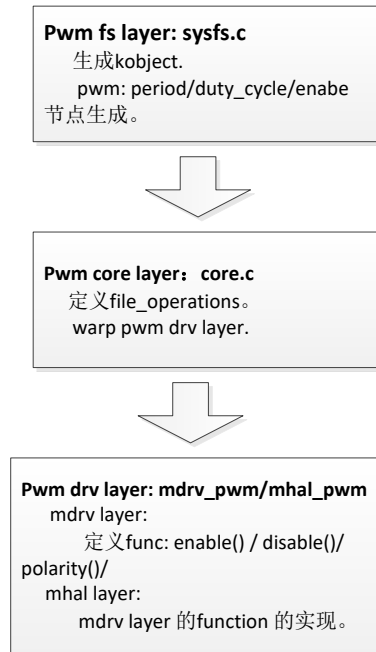


通过查看： drivers\sstar\include\infinity6\gpio.h

```
90 #define PAD_PM_SPI_CS 69
91 #define PAD_PM_SPI_CZ 70
92 #define PAD_PM_SPI_CK 71
93 #define PAD_PM_SPI_DI 72
94 #define PAD_PM_SPI_DO 73
95 #define PAD_PM_SPI_WPZ 74
96 #define PAD_PM_SPI_HLD 75
97 #define PAD_PM_LED0 76
drivers/sstar/include/infinity6/gpio.h
```

所以，只需要再配置好 dts 就可以了。

## PWM 的架构



User mode 对 PWM 的控制:

1. Export PWM number (例如 USB PAD\_PWM0)

*Command:*

```
cd /sys/class/pwm/pwmchip0
```

```
echo 0 > export
```

2. Set period(frequency) / duty\_cycle / polarity / enable

*Command:*

```
cd pwm0
```

```
echo xxxx > period
```

In our driver implementation, xxxx indicates output frequency

ex: echo 2000 > period will generate 2KHz waveform

```
echo xx > duty_cycle
```

ex: echo 25 > duty\_cycle will generate 25% duty\_cycle

```
echo inversed > polarity
```

Inverse output waveform, default is normal

```
echo 1 > enable
```

Enable output waveform

对应 user 层代码:

即:

Open 一个节点;

Write 节点;