

CH641 无线充应用常见问题与解决办法

版本: V1.0

<http://wch.cn>

(1) 静态 FOD

无负载上电报错, 或设备移除后报错, 一般是静态 FOD 问题。

可以开启串口, 观察下图所示的函数的输出值 (图 1) 与头文件定义的阈值 (图 2) 进行比较, 如果输出值比阈值低, 可以提高阈值。注意 5V, 9V, 12V 是独立设置的三个阈值。

```

2  /*
3  void Get_Cur_Q_Ref(uint16_t meas)
4  {
5  #if (DE_PRINTF == 2)
6      printf("osc_adc:%d", meas);
7  #endif
8      if(volt_state.bit_fastcharge == 0) Expt.q_value_ref = Q_VALUE_LTHRD;
9      else Expt.q_value_ref = Q_VALUE_HTHRD;
10     if(volt_state.bit_highvol) Expt.q_value_ref = Q_VALUE_HHTRD;
11 }
12

```

图 1 函数的输出值

```

/* Q value of the detected object */
#define Q_VALUE_LTHRD      18
#define Q_VALUE_HTHRD      38
#define Q_VALUE_HHTRD      50

```

图 2 头文件定义的阈值

如不是上述问题, 可以开启串口打印 PING 电流值, 如图 3 所示:

如果待机电流比 PING_0_3A_ISEN 大, 则静态会触发。需要适当修改头文件的 PING_0_3A_ISEN。

其中, 电流计算方法: 电流的 mA 值 = ADC 值 * 4。

```

8  * @return none
9  */
10 void Get_ISEN_State( void )
11 {
12     uint16_t adc_temp;
13     ADC_Set_ISEN();
14     adc_temp = ADC_Start();
15     printf("%d ", adc_temp);
16
17     if( adc_temp > PING_1_6A_ISEN )
18     {
19         Expt.fod_cnt |= 0x80;
20         if(Expt.fod_cnt & 0x40) Expt.fod_cnt = 0x80;
21         Expt.fod_cnt++;
22         if((Expt.fod_cnt & 0x1f) > 1) Expt.fod_cnt |= 0x
23     }
24     else if(adc_temp > PING_0_3A_ISEN)
25     {
26         Expt.fod_cnt |= 0x40;
27         ...

```

图 3 串口打印 PING 电流值

(2) NTC 的使用

NTC 与 LED 复用的方式下，需要注意温度的设定要和硬件相匹配。默认 NTC 为 100K b=3950。

与 NTC 并联的 LED 灯最好选蓝灯，因为蓝灯导通电压比较高，一般 2.5V 左右。

如图 4 所示：

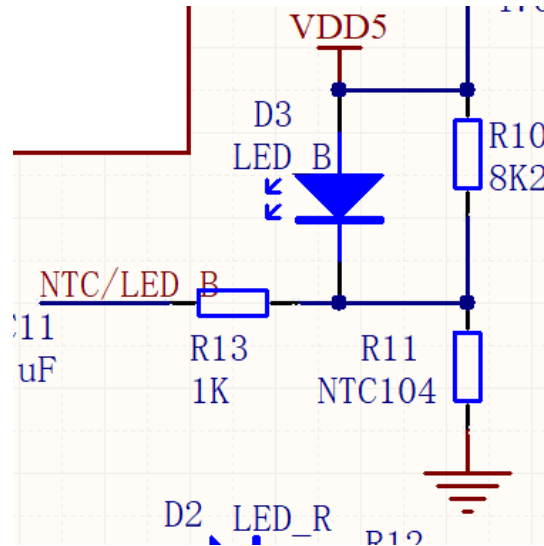


图 4 与 NTC 并联的 LED 灯部分电路示意图

VDD5 电压是 4.8V, D3 的压降大于 2.5V 开始导通。注意浮空测量 NTC 时，D3 不可以有电流，否则会影响 NTC 的测量，即要求引脚浮空时，NTC/LED_B 网络的最低电压为 $(4.8 - 2.5) = 2.3V$ 。随着温度的升高，R11 逐渐减小，NTC/LED_B 的浮空电压也逐渐减小，二极管两端的压差逐渐增大。当 R10=8.2K 时，因为 NTC/LED_B 最低 2.3V，所以 R11 最小值是 7.5K，对应 95 摄氏度。如果温度再高，因为二极管开始过电流，NTC/LED_B 的电流会锁定在 2.3V。不过温度适当超一些，不足以点亮蓝灯 LED，实测 120 摄氏度时，蓝灯也是微微亮，但是建议把过温保护点设置在 95 度以下。

另一方面，随着温度的降低，NTC 阻值开始增大，NTC/LED_B 的浮空电压也逐渐增大，但是为测量温度值，浮空输入电压不可超过 3.3V, R10=8.2K 时，对应的 R11 阻值是 18K，对应温度是 69 度。如果 69~95 度的温度区间不在期望的范围内，可以通过修改 R10，并按照上述流程重新计算。

(3) 死区的设置

```
usercfg.pwm_dead_time = 7;
usercfg.pwm_dt_forword = 4;
```

图 5 死区参数设置

如图 5 所示，如果标准死区时间参数设置为 7，则对应的时间是 $7/48 = 146ns$ ；

pwm_dead_time 是标准死区时间，衡量的是从“关闭上管，到开启下管”这段时间；

pwm_dt_forword 是用来设置从“关下管，到开启上管”这段时间的。

一般情况下，pwm_dt_forword 的时间比 pwm_dead_time 的时间要短，具体时间的设定可以抓取栅极信号来确定。

(4) 动态 FOD 的设置

```

17
18 /* Dynamic FOD related */
19 #define DYN_FOD_ENABLE      1
20 #define MIN_FOD_ISEN        150
21 #define FOD_CHECK_MAXCNT    3
22

```

图 6 动态 FOD 的设置

动态 FOD 的设置是通过计算功率差实现的：

输入功率 $W = U \cdot I$ ；

$U = \text{ADC_V} / 1023 \cdot 3.3 / 0.2$ ；

$I = \text{ADC_I} / 1023 \cdot 3.3 / 80 / 0.01$ ； //按照 80 倍放大，10mR 计算

输出功率 $W = \text{RP} / 128 \cdot X$ ； //X 的取值 5W 时为 5，10W 为 10，15W 为 15

正常工作时需要满足：输入功率 < 输出功率 * efficiency

上述公式合并后，在 5W 下可以转化成： $\text{ADC_V} \cdot \text{ADC_I} < \text{RP} \cdot 121.094 \cdot 5 / \text{efficiency}$

```

ADC_V * ADC_I < RP * 121.094 * 5 / efficiency
5W 55%: ADC_V * ADC_I < RP * 1101 ----> (ADC_V >> 3) * (ADC_I >> 1) < RP * 69
10W 60%: ADC_V * ADC_I < RP * 2018 ----> (ADC_V >> 3) * (ADC_I >> 1) < RP * 126
15W 70%: ADC_V * ADC_I < RP * 2595 ----> (ADC_V >> 3) * (ADC_I >> 1) < RP * 162

```

图 7 动态 FOD 的设置实例

(5) 开启串口调试

```

22
23
24 /* DEBUG UATR Definition Note that
25 #ifndef DE_PRINTF
26 #define DE_PRINTF    0
27 // #define DE_PRINTF  1
28 // #define DE_PRINTF  2
29 #endif
30

```

图 8 串口设置

默认串口功能未开启，串口引脚被 LED_R 占用。如果设置 DE_PRINTF 为 1 或 2，则开启串口，此时 LED_R 自动无效。

(6) 文件修改

一般用户仅需修改 user_proc.c、user_proc.h、logic.c、logic.h、led.c、led.h 这几个文件用来配置或修改灯效，其他文件无需修改。