

2 系列系统时钟跑 1M 注意点

2 系列设置系统时钟跑 RC 1MHz，可通过调用函数 `sys_clock_rc()`，之后需调用 `sysclock_div2(0)` 函数，防止在设置了 RC 前调用了系统 2 分频函数 `sysclock_div2(1)`；如果这样，可能导致系统时钟有时跑 1M，有时跑 500K。例如在 AUX 模式下让其系统时钟跑 RC 1MHz。当退出 AUX 模式后需重新把系统时钟设置到 48M，此时需调用 `sys_clock_pll()` 函数。如下图 1 红色方框设置。

```
090 void aux_function(void)
091 {
092     key_voice_disable = 1;
093     // sys_clock_pll();
094     sysclock_div2(1);
095     flush_low_msg();
096     Disp_Con(DISP_AUX);
097     set_max_vol(MAX_ANALOG_VOL, MAX_DIGITAL_VOL);
098     dac_out_select(DAC_AMUXO);
099     sys_clock_rc();
100     sysclock_div2(0);
101     deal_aux();
102     sys_clock_pll();
103     main_vol_set(0, CHANGE_VOL_NO_MEM);
104 }
```

图 1

当系统设置为 1MHz 后，AD 采用样的波特率需做调整。当系统调用了 `sys_clock_rc` 函数后，会把 `sys_clock_flag` 变量清 0，系统调用了 `sys_clock_pll` 函数后，会把 `sys_clock_flag` 变量置 1，如下图 2。

```
086 #pragma disable
087 void sys_clock_rc(void)
088 {
089     mainclock_select(MAIN_CLK_RC);
090     sd_speed_init(0, 3);
091     SPIBAUD = 0x0;
092     sys_clock_flag = 0;
093 }

#pragma disable
void sys_clock_pll(void)
{
    SPIBAUD = 0x04;
    sd_speed_init(1, 200);
    mainclock_select(MAIN_CLK_PLL);
    sys_clock_flag = 1;
    clock_change_delay = 0;
}
```

图 2

如下图 3 是 `adc_clock` 函数体，可知道，`sys_clock_flag` 置 1 时 AD 采样波特率设置在一个稳定值下，当系统时钟是 RC 1M 下，`sys_clock_flag` 变量清 0，所以的确保 ADCCON 控制器的波特率设置为 1M，BIT4 和 BIT5 要清零。`adc_clock` 函数修改如下图 4。

```
133 void adc_clock(void)
134 {
135     if (sys_clock_flag)
136     {
137         ADCCON |= 0x30;
138     }
139 }
140
```

图 3

```

133 void adc_clock(void)
134 {
135     if (sys_clock_flag)
136     {
137         ADCCON |= 0x30;
138     }
139     else
140     {
141         ADCCON &= ~0x30;
142     }
143 }
144

```

图 4

系统跑 RC 1M 后，按键音要屏蔽。deal_aux 函数里 while 里的如下语句也要屏蔽掉。

```

040
041 //         if(dac_cnt > 20)
042 //         {
043 //             dac_sw(0);
044 //         }
045

```

备注：系统跑在低频率下，对一些 DAC 音量的设置会有影响。