

# 驱动开发规范

部门： 第五事业部  
版本： V1.0  
时间： 2014-02-17

## （一） 开发规范

- 1: 熟练掌握模块所涉及的硬件设计，如所用的电，GPIO，CLOCK 等。
- 2: 开发阶段时序一定要符合 spec。
- 3: 关注负责模块的耗电流，包括工作电流，模块加入前后系统底电流变化。
- 4: 新功能和较大的修改需要通过评审。
- 5: 整体功能完成后需由硬件部配合验证时序。
- 6: 对于因为自己代码出现的编译或者运行时的 warning 都需要处理掉。
- 7: 主管及时更新项目表格中的模块调试状态。

## （二）编程规范

**1: 用 #if XXX #endif 的时候需要在 #endif 中指明对应的是哪个 #if 格式如下：(多个#if 时)**

```
void __init msm8226_add_drivers(void)
{
    #if NOT_DO_INIT
        msm_smem_init();
        msm_init_modem_notifier_list();
    #if JUST_FOR_TEST
        msm_smd_init();
    #if NOT_USE_RPM_SPM_PM
        msm_rpm_driver_init();
        msm_spm_device_init();
        msm_pm_sleep_status_init();
    #endif /* NOT_USE_RPM_SPM_PM */
    #endif /* JUST_FOR_TEST */
        rpm_regulator_smd_driver_init();
    #endif /* NOT_DO_INIT */
}
```

**2: 注释掉的代码如果超过两行请使用 #if 0 的方式。如果使用 #if 0 则需要在后面跟上一个标识，标识后面可以跟上注释，并在对应 #endif 中使用此标识：**

```
void __init msm8226_add_drivers(void)
{
    msm_smem_init();
    msm_init_modem_notifier_list();
    msm_smd_init();
    #if 0 //TEST_LABEL: 注释
        msm_rpm_driver_init();
        msm_spm_device_init();
        msm_pm_sleep_status_init();
        rpm_regulator_smd_driver_init();
        qpnp_regulator_init();
        if (of_board_is_rumi())
            msm_clock_init(&msm8226_rumi_clock_init_data);
        else
            msm_clock_init(&msm8226_clock_init_data);
        tsens_tm_init_driver();
    #endif /* TEST_LABEL */
}
```

```

        msm_thermal_device_init();
    }

```

3: 缩进格式保持和源文件一致，如果是新建文件，缩进使用 4 个空格。

4: 对于不能一眼就可出来的功能或者算法，需要写注释说明功能或者原理等。自己写的代码文件要有说明。

5: 对于自己定义的对外接口函数需要对其说明，包括功能，参数，返回值，格式如下：

```

/*****
 * Description: 函数功能、性能等的描述
 * Input: 输入参数说明，包括每个参数的作用、取值说明及参数间关系。
 * Return: 函数返回值的说明
 *****/
*/

```

6: 对于数据结构申明，需要在成员后面注明功能等。

7: 注释的格式要统一起来，[SIMT-姓名-年月日]，姓请采用首字母，名字请使用拼音全称，不采用英文命名规则，采用顺序拼写,比如“周靓炯”。

注释示例：

[SIMT-zliangjiong-20140101]

8: 单行代码也需要用 {}, 比如：

```

if(ret < 0)
{
    return ret;
}

```

9: 命名要有意义，Review 人员需要在 codereview 或者代码评审的时候去关注。

10: 代码行最大长度宜控制在 70 至 80 个字符以内

长表达式要在低优先级操作符处拆分成新行，操作符放在新行之首（以便突出操作符），拆分出的新行要进行适当的缩进，使排版整齐，语句可读。

```

if((very_longer_variable1 >= very_longer_variable12)
    && (very_longer_variable3 <= very_longer_variable14)

```

```
    && (very_longer_variable5 <= very_longer_variable16))  
{  
    DoSomething();  
}
```

**11: 使用统一的 Debug 打印函数，规范打印信息，在 user 版本中把 debug 信息关闭**

**12: 代码里要做输入验证边界检查，不要让别人犯的错导致我们的程序崩溃 (Code Review 人员在 review 时需要关注)。**

**13: 降低函数间的耦合度，并提高函数的独立性以及代码可读性、效率和可维护性 (Code Review 人员在 review 时需要关注)。**