# 编码规范

1. 变量命名规范：
   1. 全局变量， 前缀g\_, 如： g\_global\_var
   2. 局部变量，类的public变量: 小写字母+下划线， 如：local\_var, public\_var
   3. 类的protected变量：下划线作前缀，如\_protected\_var
   4. 类的private变量：下划线\*2作前缀，如\_\_private\_var
   5. 尽量不要使用临时变量，如果要使用临时变量，请以tmp\_int，tmp\_double2这样的方式命名， 尽可能给予注释
   6. 变量不要再局部声明，必须再函数开头统一声明
   7. 变量尽量不要使用缩写，尤其是较长的单词名称，目前允许的缩写名称如下：
      1. avg: average, max: maximum, min: minimum
      2. idx: index (像素，矩阵，向量的下标)，len：length， sz：size
      3. prev: previous, curr: current, next: next
      4. del：delete，mv：move， cmp：compare

h）变量中单词顺序如下（优先级从高到低）：

1）get，set，del，cmp等谓词

2）avg，max，min，prev，curr等形容词

3）idx，len，sz等名词

4）x，y，类名等表示属主的词

I）变量中不要出现多余的形容词，如good，bad，big，small

j) 变量长度尽量不要超过20个字符

2. 函数及类命名规范

a） 类命用camelcase，开头大写，如：

class ExampleClass

b) 全局函数，类的public方法， camelcase，开头小写，如：

void getMaxValue();

ExampleClass::deleteBackElement()

c） 类的protected方法，下划线作前缀

ExampleClass::\_resetStart()

d) 类的protected方法，下划线\*2作前缀

ExampleClass::\_\_findOptimalValue()

e) 类的方法名和全局函数名尽量用全称，缩写规范和顺序规范遵从变量规范，长度不要超过5个单词：

Bad example：ExampleClass::getWifiSwitchStateSerialAsync()

3. 程序设计规范：

遵从面向对象设计规范，可以封装为方法的步骤用独立函数封装，需要内在状态的用类封装，多个类间存在紧耦合的用namespace隔离，只被类调用的数据结构如struct Point{int x；int y；}在类的public部分声明。

使用了template的类放在hpp文件中，和实现放一起。

4. 注释

使用Doxygen注释方式

单行注释: /// 三个下划线

多行注释:

/\*

\* @file FileName 只在文件顶行用

\* @author AuthorName 只在文件顶行用

\* @class ClassName 只在class前用

\* @fn Function(int…) 在类方法或全局函数前用

\* @param ParamName This param is used to… 描述某一个参数

\* @brief: 简介, 可以介绍类的作用或函数等等

\* @note: 函数中用

\* @ref 引用,请看doxygen文档

\*/

每个类和函数的声明都需要进行注释

每个文件都需要文件头

5. 其它

不要使用Tab, 统一使用四个空格进行缩进

花括号如下, 左花括号在同一行:

int testFunction() {

}