

Doxygen 可以根据我们的代码注释生成我们需要的文档，这个文档明显比自己整理的要好看的多有结构的多，另外它还能生成每个文件的调用关系图、每个类的继承关系图、每个函数的调用关系图、每个文件夹的包含关系图……

1 项目注释

项目注释块用于对项目进行描述，每个项目只出现一次，一般可以放在 `main.c` 主函数文件头部。对于其它类型的项目，置于定义项目入口函数的文件中。对于无入口函数的项目，比如静态库项目，置于较关键且不会被外部项目引用的文件中。

项目注释块以 `/** @mainpage` 开头，以 `*/` 结束。章节可自行添加。

```
1  /**
2  * @mainpage 项目名
3  * @section 项目详细描述
4  * @section 功能描述
5  * @section 用法描述
6  * @section 固件更新
7  */
```

2 文件注释

一般在每一个文件开头加入版权、作者、时间等描述。文件注释描述了该文件的内容，如果一个文件只声明，或实现，或测试了一个对象，并且这个对象已经在它的声明处进行了详细的注释，那么就没必要再加上文件注释，除此之外的其他文件都需要文件注释。

```
1  /**
2  * @file 文件名
3  * @brief 简介
4  * @details 细节
5  * @author 作者
6  * @version 版本号
7  * @date 年-月-日
8  * @copyright 版权
9  */
```

3 类的注释

每个类的定义都要附带一份注释，描述类的功能和用法，除非它的功能相当明显。如果类的声明和定义分开了 (例如分别放在了 `.h` 和 `.cpp` 文件中)。此时，描述类用法的注释应当和接口定义放在一起，描述类的操作和实现的注释应当和实现放在一起。

```
1  /**
2  * @brief 类的详细描述
3  */
```

4 函数注释

基本上每个函数声明处前都应当加上注释，描述函数的功能和用途，对一些必要参数进行解释说明。只有在函数的功能简单而明显时才能省略这些注释 (例如，简单的取值和设值函数)

```
1  /**
2  * @brief 函数描述
3  * @param 参数描述
4  * @return 返回描述
5  * @retval 返回值描述
6  */
```

5 常量/变量注释

通常变量名本身足以很好说明变量用途，某些情况下，也需要额外的注释说明。

5.1 常量/变量上一行注释

```
1  /// ... 注释 ...
2  变量
3  //! ... 注释 ...
4  变量
5  /*! ... 注释 ... */
6  变量
7  /** ... 注释 ... */
8  变量
```

5.2 常量/变量后注释

```
1  变量 /*!< ... 注释 ... */
2  变量 /**< ... 注释 ...*/
3  变量 //!< ... 注释 ...
4  变量 ///< ... 注释 ...
```

6 枚举类型注释

枚举可看做类的一种特殊形式，添加简单说明，Doxygen 会根据其后面的 enum 解析出枚举类型。

```
1  /**
2  * @brief 简要说明
3  *
4  * @details 详细说明
5  */
6  enum 枚举类型名
```

```
7 {
8     值1,/**< ...值1的注释... */
9     值2 /**< ...值2的注释... */
10 };
```

7 宏注释

```
1 /**
2  * @brief 宏简要说明
3  *
4  * @details 宏详细说明
5  */
6 #define ... ...
```

8 包含一段代码的注释

```
1 /// @code
2 要包含的代码内容
3 /// @endcode
```

9 分组注释

表明注释块是一组类、文件或名称空间的记录。这可以用于对类、文件或名称空间进行分类，并记录这些类别。还可以将组用作其他组的成员，从而构建组的层次结构。

```
1  /*! \defgroup mygrp This is my group.
2  *   这里为 mygrp 组添加记录
3  *   @{
4  *   */
5
6  /*! A function */
7  void func()
8  {}
9
10 /*! @} */
```

10 特殊命令简介

命令	说明	语法
@author	作者	author { list of authors }
@brief	简介	brief { brief description }
@bug	缺陷, 链接到所有缺陷汇总的缺陷列表	bug{ description }
@copyright	版权	copyright { copyright description }
@code .. @endcode	在注释中开始说明一段代码, 直到 @endcode 命令结束	
@date	年-月-日	date { date description }
@deprecated	弃用说明。可用于描述替代方案, 预期寿命等	deprecated description
@details	细节	details { detailed description }
@file	文件名	file [< name >]
@link	连接 (与 @see 类库, { @link www.google.com })	link < link-object >
@mainpage	主页信息	mainpage [(title)]
@param	参数	param [(dir)] < parameter-name > { parameter description }
@pre	用来说明代码项的前提条件	pre { description of the precondition }
@relates	通常用做把非成员函数的注释文档包含在类的说明文档中	relates [< name >]
@return	描述返回意义	return { description of the return value }
@retval	描述返回值意义	retval < return value > { description }
@see	参考	see { references }
@since	通常用来说明从什么版本、时间写此部分代码	since{ description }
@throw	异常描述	throw < exception-object > { exception description }
@todo	待处理	todo { paragraph describing what is to be done }
@version	版本号	version { version number }
@warning	警告信息	warning { warning message }

以上仅列出部分命令, 详情请查看 doxygen 官方文档

<https://www.doxygen.nl/manual/commands.html>