

# 代码文档工具 Doxygen 快速入门

TN01010101 V1.00 Date: 2012/07/04

工程技术笔记

类别	内容
关键词	Doxygen; 自动文档; 在线文档; 文档系统; 注释语法; 参考手册
摘 要	Doxygen 是一种开源跨平台的, 以类似 JavaDoc 风格描述的文档系统, 完全支持 C、C++、Java、Objective-C 和 IDL 语言, 部分支持 PHP、C#。注释的语法与 Qt-Doc、KDoc 和 JavaDoc 兼容。Doxygen 可以从一套归档源文件开始, 生成 HTML 格式的在线类浏览器, 或离线的 LATEX、RTF 参考手册。本文通过“step by step”方式对 Doxygen 进行快速入门介绍



## 修订历史

版本	日期	原因
V1.00	2012/07/04	创建文档



## 销售与服务网络（一）

### 广州周立功单片机科技有限公司

地址：广州市天河区北路 689 号光大银行大厦 12 楼 F4

邮编：510630

电话：(020)38730916 38730917 38730972 38730976 38730977

传真：(020)38730925

网址：[www.zlgmcu.com](http://www.zlgmcu.com)

新浪微博：ZLG-周立功（<http://weibo.com/ligongzhou>）



### 广州专卖店

地址：广州市天河区新赛格电子城 203-204 室

电话：(020)87578634 87569917

传真：(020)87578842

### 南京周立功

地址：南京市珠江路 280 号珠江大厦 1501 室

电话：(025) 68123901 68123902

传真：(025) 68123900

### 北京周立功

地址：北京市海淀区知春路 113 号银网中心 A 座  
1207-1208 室（中发电子市场斜对面）

电话：(010)62536178 62536179 82628073

传真：(010)82614433

### 重庆周立功

地址：重庆市石桥铺科园一路二号大西洋国际大厦  
（赛格电子市场）1611 室

电话：(023)68796438 68796439

传真：(023)68796439

### 杭州周立功

地址：杭州市天目山路 217 号江南电子大厦 502 室

电话：(0571)89719480 89719481 89719482

89719483 89719484 89719485

传真：(0571)89719494

### 成都周立功

地址：成都市一环路南二段 1 号数码科技大厦 403 室

电话：(028)85439836 85437446

传真：(028)85437896

### 深圳周立功

地址：深圳市福田区深南中路 2072 号电子大厦 12 楼

电话：(0755)83781788（5 线）

传真：(0755)83793285

### 武汉周立功

地址：武汉市洪山区广埠屯珞瑜路 158 号 12128 室  
（华中电脑数码市场）

电话：(027)87168497 87168297 87168397

传真：(027)87163755

### 上海周立功

地址：上海市北京东路 668 号科技京城东楼 12E 室

电话：(021)53083452 53083453 53083496

传真：(021)53083491

### 西安办事处

地址：西安市长安北路 54 号太平洋大厦 1201 室

电话：(029)87881296 83063000 87881295

传真：(029)87880865

### 厦门办事处

E-mail: [sales.xiamen@zlgmcu.com](mailto:sales.xiamen@zlgmcu.com)

### 沈阳办事处

E-mail: [sales.shenyang@zlgmcu.com](mailto:sales.shenyang@zlgmcu.com)



## 销售与服务网络（二）

### 广州致远电子股份有限公司

地址：广州市天河区车陂路黄洲工业区 3 栋 2 楼

邮编：510660

传真：(020)38601859

网址：[www.zlg.cn](http://www.zlg.cn)

新浪微博：ZLG-周立功（<http://weibo.com/ligongzhou>）



#### 技术支持：

##### CAN-bus：

电话：(020)22644381 22644382 22644253

邮箱：[can.support@zlg.cn](mailto:can.support@zlg.cn)

##### iCAN 及数据采集：

电话：(020)28872344 22644373

邮箱：[ican@zlg.cn](mailto:ican@zlg.cn)

##### MiniARM：

电话：(020)28872684 28267813

邮箱：[miniarm.support@zlg.cn](mailto:miniarm.support@zlg.cn)

##### 以太网：

电话：(020)22644380 22644385

邮箱：[ethernet.support@zlg.cn](mailto:ethernet.support@zlg.cn)

##### 无线通讯：

电话：(020) 22644386

邮箱：[wireless@zlg.cn](mailto:wireless@zlg.cn)

##### 串行通讯：

电话：(020)28267800 22644385

邮箱：[serial@zlg.cn](mailto:serial@zlg.cn)

##### 编程器：

电话：(020)22644371

邮箱：[programmer@zlg.cn](mailto:programmer@zlg.cn)

##### 分析仪器：

电话：(020)22644375

邮箱：[tools@zlg.cn](mailto:tools@zlg.cn)

##### ARM 嵌入式系统：

电话：(020) 22644383 22644384

邮箱：[NXPARM@zlg.cn](mailto:NXPARM@zlg.cn)

##### 楼宇自动化：

电话：(020)22644376 22644389 28267806

邮箱：[mjs.support@zlg.cn](mailto:mjs.support@zlg.cn)

[mifare.support@zlg.cn](mailto:mifare.support@zlg.cn)

#### 销售：

电话：(020)22644249 22644399 22644372 22644261 28872524

28872342 28872349 28872569 28872573 38601786

#### 维修：

电话：(020)22644245



## 目 录

1. 简介.....	1
2. 安装.....	2
3. 运行第一个示例.....	3
3.1    打开图形向导Doxywizard .....	3
3.2    选择配置文件 .....	3
3.3    生成文档.....	3
3.4    浏览文档.....	4
4. 单模块简单示例.....	5
4.1    编写注释 .....	5
4.1.1    有效注释.....	6
4.1.2    定义模块.....	6
4.1.3    加入到模块.....	6
4.1.4    函数注释.....	6
4.2    生成配置.....	6
4.2.1    项目配置.....	6
4.2.2    模式配置.....	7
4.2.3    输出配置.....	7
4.2.4    图表配置.....	8
4.2.5    支持中文.....	8
4.2.6    保存配置文件.....	9
4.3    生成文档并浏览.....	9
5. 多模块示例.....	10
5.1    几个常用标记.....	10
5.1.1    @addtogroup.....	10
5.1.2    @{和@}.....	10
5.1.3    成员注释.....	10
5.2    多模块文档化.....	11
5.2.1    DI模块文档化.....	11
5.2.2    DO模块文档化.....	13
5.2.3    生成文档.....	15
6. 注释的书写建议.....	16
6.1    注释原则.....	16
6.2    对比示例.....	16
7. 生成离线文档.....	18
7.1    CHM格式.....	18
7.1.1    生成CHM索引.....	18
7.1.2    建立CHM工程.....	18
7.1.3    编译CHM工程.....	18
7.1.4    查看CHM文件.....	18
7.2    RTF和word格式 .....	19
7.3    PDF格式 .....	19



8. FAQ.....	20
9. 免责声明.....	21



## 1. 简介

Doxygen是一种开源跨平台的，以类似JavaDoc风格描述的文档系统，完全支持C、C++、Java、Objective-C和IDL语言，部分支持PHP、C#。注释的语法与Qt-Doc、KDoc和JavaDoc兼容。Doxygen可以从一套归档源文件开始，生成HTML格式的在线类浏览器，或离线的LATEX、RTF参考手册。

Doxygen 是一个程序的文档产生工具，可将程序中的特定注释转换成为说明文档。通常我们在写程序时，或多或少都会写上注释，但是对于其它人而言，要直接探索程序里的注释，与打捞铁达尼号同样的辛苦。大部分有用的注释都是针对函数，类别等等的说明。所以，如果能依据程序本身的结构，将注释经过处理重新整理成为一个纯粹的参考手册，对于后面利用或维护这份程序代码的人而言将会减少许多的负担。

对于未归档的源文件，也可以通过配置 Doxygen 来提取代码结构。或者借助自动生成的包含依赖图（includedependency graphs）、继承图（inheritance diagram）以及协作图（collaborationdiagram）来可视化文档之间的关系。

一个好的程序设计师，在写程序时，都会在适当的地方加上合适的注释。如果，能够在撰写注释时，稍微符合某种格式，接着就可以通过一个工具依据程序结构及注释产生出漂亮的文档，这将令程序设计师从繁重的文档编写工作中解脱出来。

Doxygen 就是这样的一个工具。在写注释时，按照它所制订的一些规则编写，接着就可以产生出漂亮的文件了。因此，Doxygen 的使用可分为两大部分，首先是特定格式的注释撰写，第二便是利用 Doxygen 的工具来产生文档。

本文将对 Doxygen 的常用功能做一个入门介绍，更多功能或完整支持请参考 Doxygen 的用户手册。



## 2. 安装

这里只介绍 Windows 二进制安装，后面所有的介绍都是基于 Windows 平台。

可以去 <http://www.doxygen.org/download.html> 页面获取最新版本。截至本文档写作时的最新版本是 1.8.1.1，本文档的写作也是基于 Doxygen 1.8.1.1。

Doxygen 的 Windows 二进制包是自动安装文件，只需按照对话框提示即可。



### 3. 运行第一个示例

首先我们通过运行一个现成的简单示例来熟悉 Doxygen 的操作流程。

假设所需的文件包括源码都在目录“doxygen 快速入门”下面，跟着下面的步骤为文件“src\_single/aw\_di\_single.c”生成文档。

#### 3.1 打开图形向导Doxywizard

打开Doxygen 的图形前端Doxywizard，这是一个向导程序，界面如图 3.1所示。

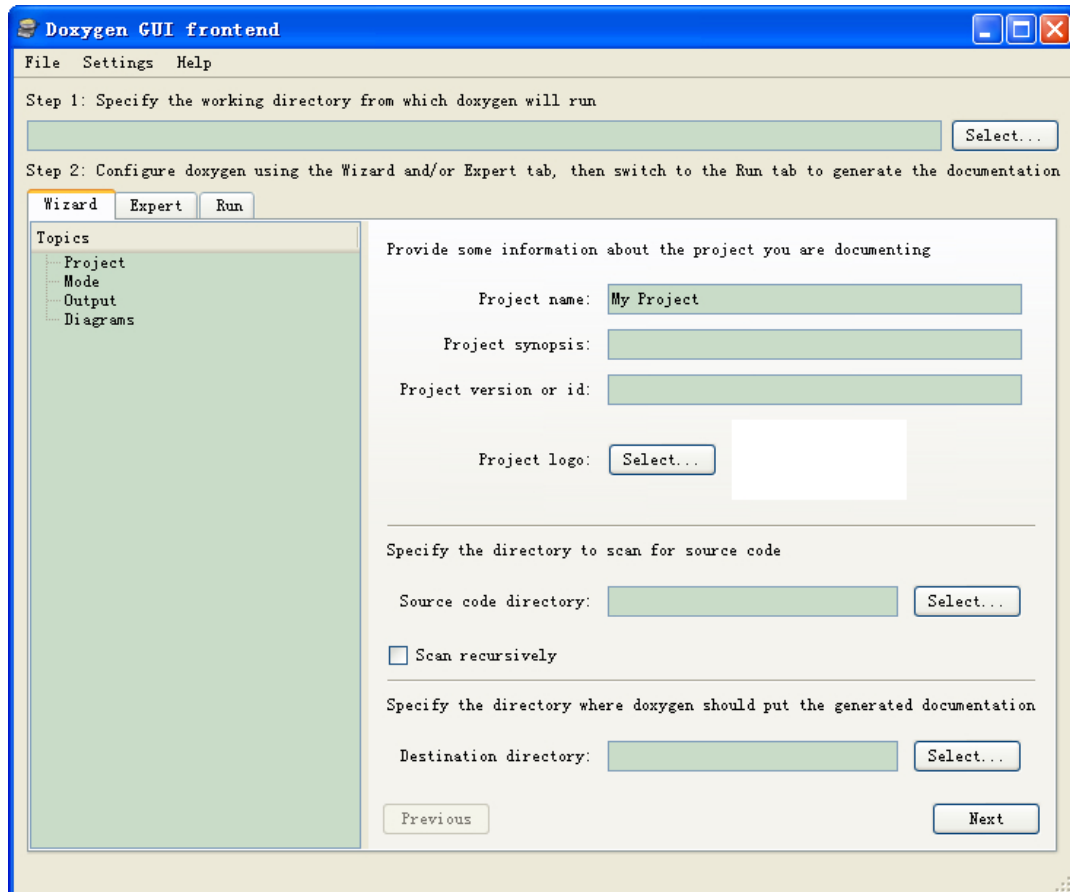


图 3.1 Doxygen 图形前端 Doxywizard

#### 3.2 选择配置文件

选择菜单“file/open...”打开浏览配置文件对话框，浏览文件夹“doxygen 快速入门”并选择文件“doxyfile\_single”，单击按钮“打开”即可。

#### 3.3 生成文档

单击标签页“run”，再单击该标签页下的按钮“Run doxygen”，此时可以在该标签页下的日志窗口中看到 Doxygen 的运行输出信息。

稍等片刻，日志窗口停止输出信息，并在最后一行显示“\*\*\* Doxygen has finished”字样就表示文档已经顺利生成。

默认生成的是 HTML 格式的网页文档，可以看到文件夹“doxygen 快速入门”下多出了一个目录“html”，默认的主页文件为“index.html”。

### 3.4 浏览文档

文档生成后单击标签页“run”左下角的按钮“Show HTML output”就可以用系统默认的浏览器打开生成的文档，如图 3.2所示。

如果使用的是低版本 IE 浏览器，在显示上可能会有些问题，原因是低版本的 IE 浏览器对 HTML 标准支持的不是很好，建议使用 mozilla firefox、google chrome 或 Apple Safari 等对 HTML 标准支持比较好的浏览器。



图 3.2 Doxygen 生成的 HTML 文档界面



## 4. 单模块简单示例

通过图 3.2 可以看到，Doxygen 生成的文档十分漂亮，赏心悦目。

如何使我们的代码能通过 Doxygen 自动生成漂亮文档呢？其实很简单，在编写代码注释时按照 Doxygen 的规矩来就行了，注释即文档，后文中很多地方提及的注释都可以理解成文档。

现以文件“src\_single/aw\_di\_single.c”为例介绍为代码编写可文档化的注释，生成的文档的效果如图 3.2 所示。

### 4.1 编写注释

如何编写 Doxygen 风格注释呢？现以文件“src\_single/aw\_di\_single.c”为例进行简要说明，文件内容如程序清单 4.1 所示。

程序清单 4.1 简单注释示例

文件 1: src\_single/aw\_di\_singe.c

```

1.  /**
2.   * @defgroup aw_di 数字量输入模块
3.   * 数字量输入模块提供简单统一的接口来获取外部数字量输入值
4.   */
5.
6.  /**
7.   * @ingroup aw_di
8.   * @brief 读取索引号为 index 的 DI 输入值
9.   * @details 这个函数读取单个的 DI 输入值
10.  *
11.  * @param index DI 索引值，索引值范围 0 ~ AW_DI_INDEX_MAX
12.  *
13.  * @retval 0 DI 的输入值为 0
14.  * @retval 1 DI 的输入值为 1
15.  * @retval -1 DI 不存在（DI 索引 index 大于 AW_DI_INDEX_MAX）
16.  *
17.  * @note 返回-1 表示操作的 DI 不存在，原因是参数 index 大于 AW_DI_INDEX_MAX，
18.  *       如下面的操作将得到返回值-1：
19.  * @code
20.  * index = AW_DI_INDEX_MAX + 1;
21.  * value = aw_di_get(index);
22.  * @endcode
23.  */
24. int aw_di_get (unsigned index)
25. {
26.     // ...
27.     return -1;
28. }
```

可以看到程序清单 4.1 中的注释比我们通常写的代码注释多了些以“@”开头的标记，

正是这些标记告诉Doxygen如何提取文档以及如何组织文档结构的。下面对这段注释进行解释。

#### 4.1.1 有效注释

第 1 和第 6 行的 “/\*\*” 表示要求 Doxygen 处理这段注释，否则如果写成 “/\*” 或 “/\*\*\*\*” 之类这段注释就会被忽略掉。这样就可以控制哪些注释要出现在文档中供用户阅读，哪些注释仅仅是给代码编写者或维护者阅读。

#### 4.1.2 定义模块

第 2 行的 “@defgroup” 表示定义一个模块，语法是 “@defgroup 模块 ID 模块名”。则 “@defgroup aw\_di 数字量输入模块” 定义了一个名为 “数字量输入模块” 的模块，模块的 ID 为 “aw\_di”。

模块 ID 必须使用英文，这个 ID 的作用是区分不同模块，并可供其他模块或注释引用。

第 3 行是对 “aw\_di” 模块的说明。

#### 4.1.3 加入到模块

第 7 行 “@ingroup aw\_di” 表示把这个注释块加入到模块 “aw\_di” 中。

#### 4.1.4 函数注释

第 8 到 22 行对函数 aw\_di\_get() 开始进行注释。

第 8 行 “@brief” 是一个简要描述，让阅读者快速了解这个函数的功能。

第 9 行 “@details” 是详细描述，这个可以省略。

第 11 行 “@param” 开始描述函数的参数，格式为 “@param 参数名 参数描述”。

第 13 到 15 行 “@retval” 对返回值进行描述，格式为 “@retval 返回值 返回值描述”。

第 17 到 22 行 “@note” 是注意事项。注意事项的描述中的第 19 到 22 行又用到了 “@code” 和 “@endcode”，这两个标记用于插入代码范例，标记之间的文本会当做代码来呈现，对关键字和函数等进行语法着色显示。

## 4.2 生成配置

编写完代码注释后，在使用 Doxygen 为代码生成文档前还需要为工程生成一个配置文件。跟着下面的步骤进行设置，未加说明的配置请留空（使用默认配置）。

### 4.2.1 项目配置

打开Doxygen 的图形前端Doxywizard，选择 “Wizard” 标签页，然后选择 “Topics” 栏下的 “Project” 项，进入项目设置界面，如图 4.1所示。

#### 1. 选择项目路径

单击 “Select...”，浏览并选择 “doxygen 快速入门” 作为当前工作路径。

#### 2. 设置项目名称

在 “Project name:” 后面填项目名，如 “AW DI Single”。

在 “Project synopsis:” 后面填项目简介，如 “AnyWhere DI driver”。

在 “Project version or id:” 后面填项目版本号，如 “V1.0”。

#### 3. 设置源码路径

单击 “Source code directory:” 后面的按钮 “Select...” 选择源码路径，如选择 “src\_single”。

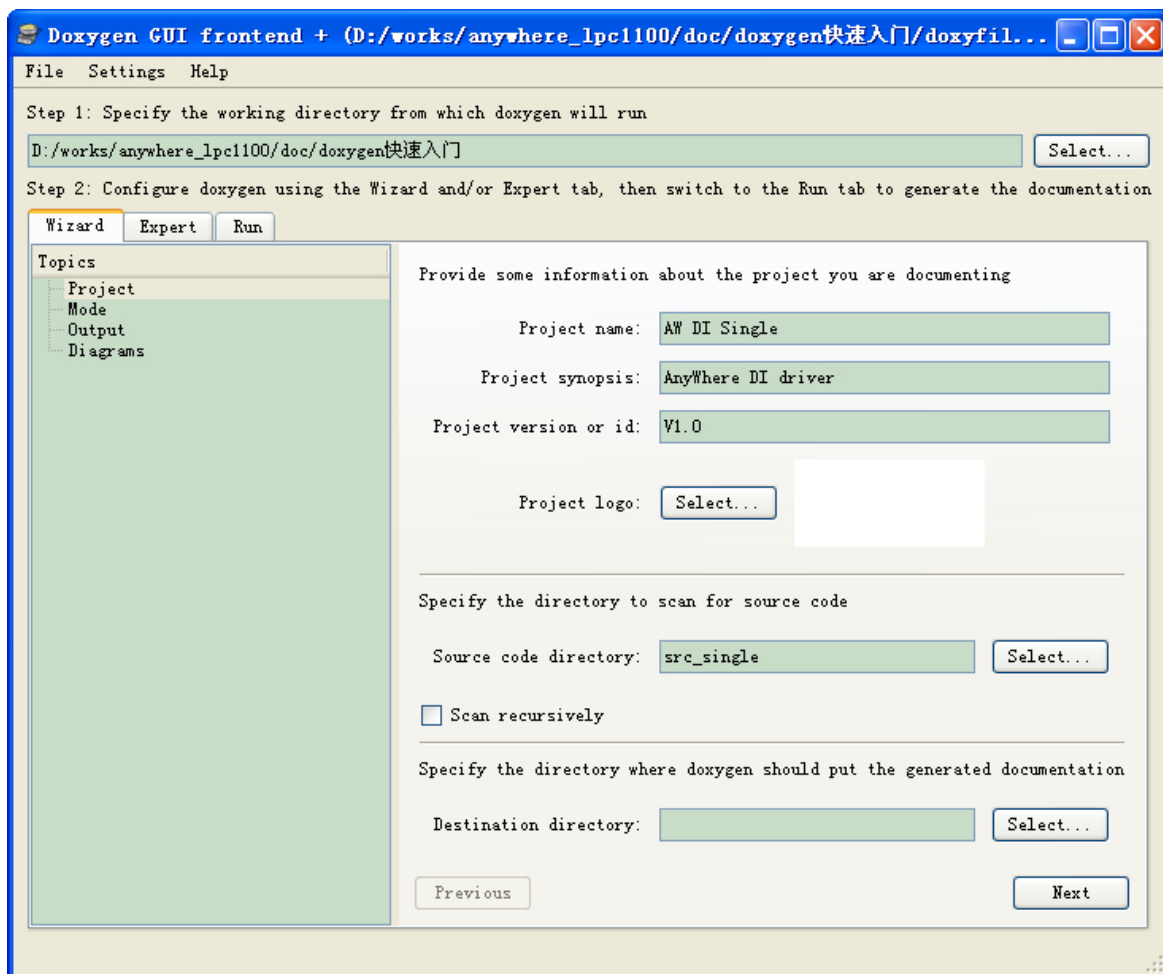


图 4.1 Doxygen 工程设置界面

#### 4.2.2 模式配置

选择“Wizard”标签页，然后选择“Topics”栏下的“Mode”项，进入模式设置界面，如图 4.2所示。选中“Documented entities only”和“Optimize for C or PHP output”。

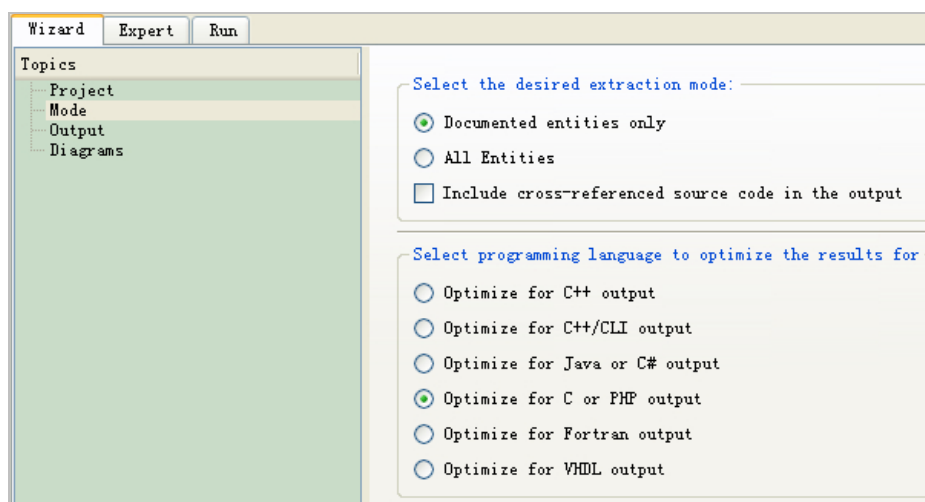


图 4.2 Doxygen 模式设置界面

#### 4.2.3 输出配置

选择“Wizard”标签页，然后选择“Topics”栏下的“Output”项，进入输出设置界面，如图 4.3 所示。

只勾选“HTML”复选框和“with navigation panel”单选按钮，去掉其它所有的勾选项。

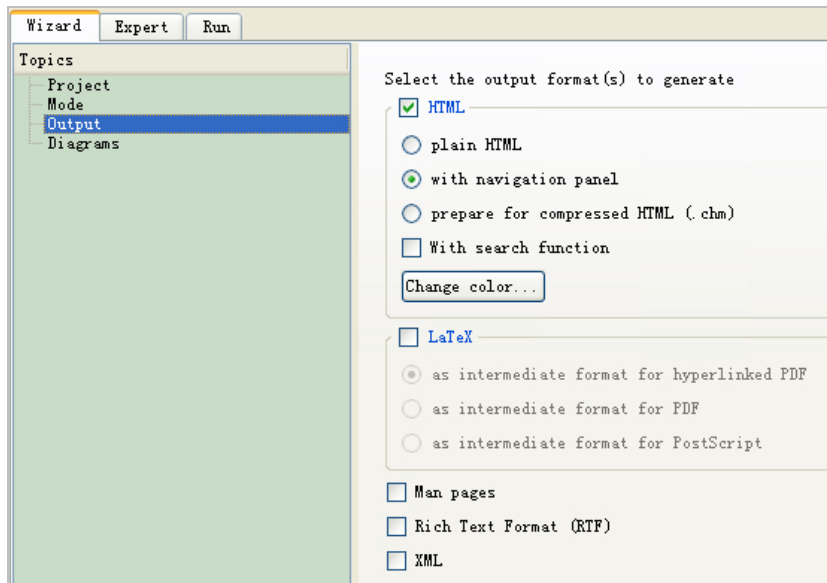


图 4.3 Doxygen 输出设置界面

#### 4.2.4 图表配置

选择“Wizard”标签页，然后选择“Topics”栏下的“Diagrams”项，进入图表设置界面，选择“No diagrams”，将不生成任何图表。关于生成图表和层次调用关系图等将在进阶使用说明中介绍。

#### 4.2.5 支持中文

经过前面的配置 Doxygen 已经可以生成英文的文档了，要支持输出中文还需要做一些额外的配置。

##### 1. 设置输出语言为中文

选择“Expert”标签页进入专家设置界面，然后选择“Topics”栏下的“Project”项，在右边窗口中找到“OUTPUT\_LANGUAGE”下拉框，单击选择“Chinese”。

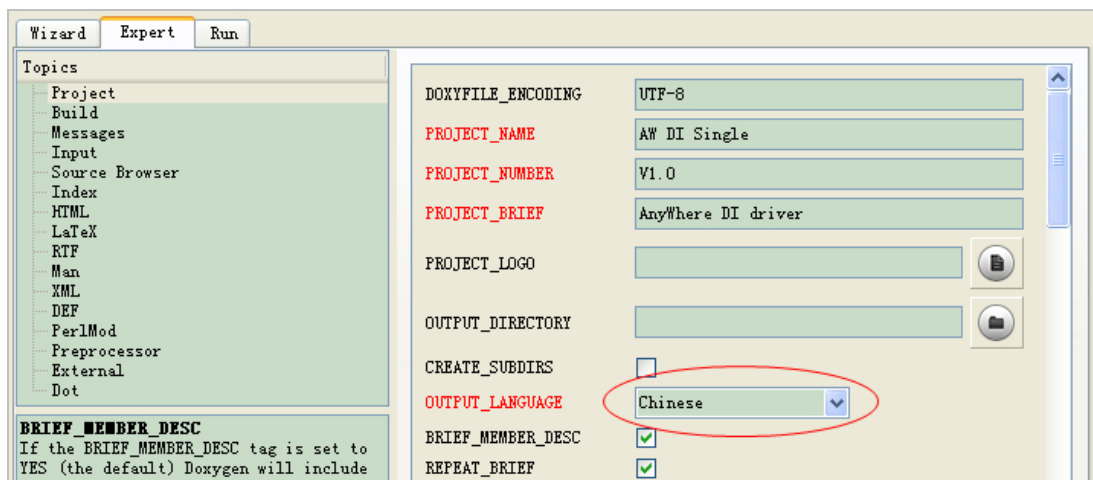


图 4.4 设置 Doxygen 输出中文

## 2. 设置输入编码

选择“Expert”标签页，然后选择“Topics”栏下的“Input”项，在右边窗口中找到“INPUT\_ENCODING”，在后面填上“GBK”。

Windows 下的简体中文用户习惯使用的编码是 GB2312，很多 Windows 下的文本或代码编辑器默认也是使用 GB2312，GBK 是 GB2312 的扩展集，支持更多汉字，并且支持繁体中文。

注：Doxygen 目前不支持自动识别文件编码，请确保所有的源码文件编码相同，否则使用编码转换工具进行转换。编码不统一或编码不能识别会造成输出的文档中有乱码。

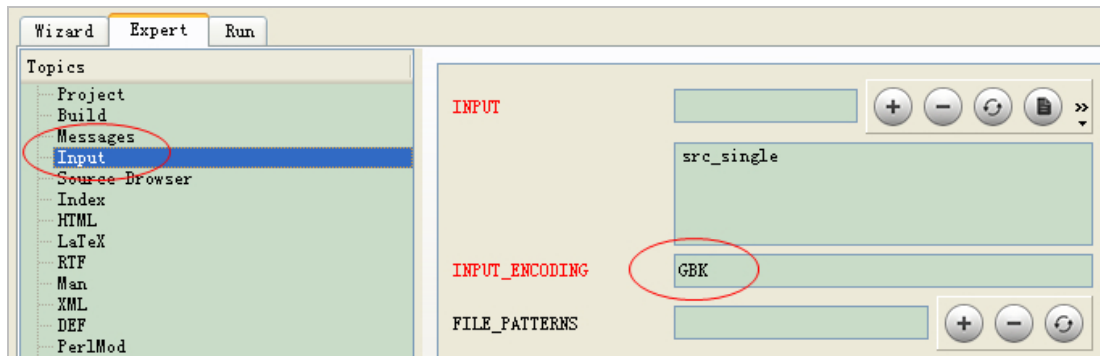


图 4.5 设置 Doxygen 输入编码

### 4.2.6 保存配置文件

至此配置已经完成，选择菜单“File/Save”弹出保存文件对话框，选择当前工作目录，将文件命名为“doxyfile\_single”并保存。

## 4.3 生成文档并浏览

单击标签页“run”，再单击该标签页下的按钮“Run doxygen”，日志窗口将输出大量信息，等信息输出停止时文档就生成完毕。可以看到文件夹“doxygen 快速入门”下多出了一个目录“html”。

文档生成后单击标签页“run”左下角的按钮“Show HTML output”就可以用系统默认的浏览器打开生成的文档。也可以把目录“html”下的文件“index.html”拖到浏览器进行浏览。



## 5. 多模块示例

4介绍了Doxygen基本的注释编写方法和文档生成方法，这一章将介绍如何为多模块的项目有效组织文档化的注释结构。

### 5.1 几个常用标记

在编写程序时我们会把代码划分成不同的“模块”，这样代码层次结构清晰，便于阅读和理解。同样 Doxygen 也提供“模块化”的标记指令来把注释进行归类，生成“模块化”的文档。

在前面的章节其实已经介绍了两个模块化指令：`@defgroup` 和 `@ingroup`。为了便于对后面内容的理解，这里先介绍另外一个常用的模块化标记“`@addtogroup`”，以及模块开始标记“`@{`”和结束标记“`@}`”，还有一个用于结构体、枚举、联合体和类成员注释的标记“`<`”。

#### 5.1.1 @addtogroup

“`@addtogroup`”表示把注释加入到某个模块，语法为“`@addtogroup 模块标签`”。这个和前面介绍过的“`@ingroup`”很类似。

#### 5.1.2 @{和@}

“`@{`”和“`@}`”用于和“`@defgroup`”、“`@addtogroup`”以及“`@ingroup`”组合来标记模块的开始和结束，组合使用方法如程序清单 5.1所示。

程序清单 5.1 组定义组合使用示例

```
1.  /**
2.   * @defgroup group_a 组 a 的名字
3.   * 组 a 的详细描述
4.   * @{
5.   */
6.   .....
7.   /* @} */
8.
9.  /**
10.   * @addtogroup group_b
11.   */
12.  /* @{ */
13.  .....
14.  /* @} */
```

第 4 和 7 行之间的内容都属于模块“`group_a`”，第 12 和 14 行之间的内容被加入到模块“`group_b`”。

注意第 4 行和第 12 行的细微区别，第 14 行另起了一行写“`@{`”，这两种用法都是合法的，依个人习惯选用，不过第 4 行的写法更能体现“`@{`”和“`@}`”之间的内容属于“`group_a`”。

#### 5.1.3 成员注释

对于结构体、枚举、联合体和类，除了对整体注释外，多数情况对他们的成员也需要作出注释，这时使用标记“`<`”跟在成员后面放置文档，如程序清单 5.2所示为成员注释示例。





第 1 行使用对常数宏放置文档，第 3 行对 my\_struct 的成员 idx 放置文档，第 4 行对 my\_struct 的成员 buf 放置文档。

注意到成员注释可以使用 “/\*\*<” 或 “/\*!<”，前者是 JavaDoc 风格，后者是 QT 风格，这个根据个人喜好使用，混合用也可以，不过为了代码风格统一，本文推荐 JavaDoc 风格。

程序清单 5.2 成员注释范例

```
1. #define BUF_SIZE    8        /**< buffer 大小 */
2. struct my_struct {
3.     int    idx;              /**< buffer 索引 */
4.     char   buf[BUF_SIZE];    /**!< buffer 空间 */
5. };
```

## 5.2 多模块文档化

本章使用一个简单的数字量输入输出软件模块来示范如何进行文档化注释，工作路径还是假设为 “doxygen 快速入门”，源码路径为 “src\_multi”。

示例源码分成 2 个文件，DI 模块 “aw\_di.c” 和 DO 模块 “aw\_do.c”。为了简化示例和对应的描述，这个示例没有通常应有的 “.h” 头文件。

DI/DO 软件模块的结构如表 5.1 所示。

表 5.1 DI/DO 模块层次结构

AnyWhere 驱动	DI 模块	DI 配置
		DI 接口函数
	DO 模块	DO 配置
		DO 状态
		DO 接口函数

### 5.2.1 DI模块文档化

DI 模块 “aw\_di.c” 文件内容如程序清单 5.3 所示。

程序清单 5.3 DI 模块文件内容

```
1. /**
2.  * @defgroup aw_driver AnyWhere 驱动
3.  * @brief AnyWhere 驱动包含 DI 和 DO 等驱动模块
4.  */
5.
6. /**
7.  * @addtogroup aw_driver
8.  * @ {
9.  */
10.
11. /**
12.  * @defgroup aw_di AnyWhere DI 模块
13.  * @details DI 模块提供单一接口获取编号 0~AW_DI_INDEX_MAX 之间的数字输入量
14.  * @ {
```

```
15.  */
16.
17. /**
18.  * @defgroup di_config DI 配置
19.  * @{
20.  */
21. #define AW_DI_INDEX_MAX      (8u)    /**< DI 索引最大值 */
22. /* @} */
23.
24. /**
25.  * @defgroup di_func DI 接口函数
26.  * @{
27.  */
28.
29. /**
30.  * @brief 读取索引号为 index 的 DI 的输入值
31.  * @details 这个函数读取单个的 DI 输入值
32.  * @param index DI 索引值，索引值范围 0~AW_DI_INDEX_MAX
33.  * @retval 0 DI 的输入值为 0
34.  * @retval 1 DI 的输入值为 1
35.  * @retval -1 DI 不存在
36.  */
37. int aw_di_get(unsigned index)
38. {
39.     // ...
40.     return -1;
41. }
42.
43. /* @} di_func */
44. /* @} aw_di */
45. /* @} aw_driver */
```

### 1. 定义Anywhere驱动模块aw\_driver

第 1 到 4 行定义了 AnyWhere 驱动模块，这是主模块，用于容纳 DI 和 DO 子模块。

### 2. 包含到aw\_driver模块

因为整个 DI 模块都属于 aw\_driver 模块，所以第 6 到 9 行以及第 45 行使用“@addtogroup”、“@{”和“@}”把整个文件都包含进 aw\_driver 模块。

### 3. 定义DI模块

第 11 到 44 行定义了 DI 模块 aw\_di。第 13 行使用“@details”标记对 DI 模块详细描述。

### 4. 定义DI配置子模块

第 17 到 22 行定义了 DI 配置子模块 di\_config，该子模块嵌套在 DI 模块 aw\_di 中，后面将省掉这种嵌套说明。

如第 21 行所示，该子模块只有一个常数宏定义，并对它使用了成员注释。

## 5. 定义DI接口函数子模块

第 24 到 43 行定义了 DI 接口函数子模块 di\_func。

## 6. DI模块结构图

如图 5.1所示为生成的HTML文档中DI模块的层次结构图，跟表 5.1所期望的结构一致。



图 5.1 DI 模块层次结构

### 5.2.2 DO模块文档化

DO模块“aw\_do.c”文件内容如程序清单 5.4所示。

程序清单 5.4 DO 模块文件内容

```

1.  /**
2.   * @addtogroup aw_driver
3.   * @{
4.   */
5.
6.  /**
7.   * @defgroup aw_do AnyWhere DO 模块
8.   * DO 模块提供单一接口设置编号 0~AW_DO_INDEX_MAX 之间的数字输出量
9.   * @{
10.  */
11.
12. /**
13.  * @defgroup do_config DO 配置
14.  * @{
15.  */
16. #define AW_DO_INDEX_MAX    (10u)    /**< DO 索引的最大值 */
17. /** @} */
18.
19. /**
20.  * @defgroup do_state DO 状态
21.  * @{
22.  */
23.

```

```

24. /**
25.  * @brief DO 输出逻辑状态
26.  */
27. typedef enum aw_dostate {
28.     AW_DOST_LOW  = 0,    /**< DO 状态为逻辑低电平 */
29.     AW_DOST_HIGH = 1    /**< DO 状态为逻辑高电平 */
30. } aw_dostate_t;
31. /* @ } do_state */
32.
33. /**
34.  * @defgroup do_func DO 接口函数
35.  * @{
36.  */
37.
38. /**
39.  * @brief 设置 DO 输出状态
40.  * @param[in] idx    DO 索引，范围 0~AW_DO_INDEX_MAX
41.  * @param[in] state 要设置的 DO 状态
42.  *                  @arg @b AW_DOST_LOW  设置 DO 为逻辑低电平
43.  *                  @arg @b AW_DOST_HIGH 设置 DO 为逻辑高电平
44.  * @return 返回 0 为设置成功，为其它值则设置失败
45.  */
46. int aw_do_set (unsigned idx, aw_dostate_t state)
47. {
48.     //...
49.     return -1;
50. }
51.
52. /* @ } do_func */
53. /* @ } aw_do */
54. /* @ } aw_driver */

```

DO 模块的层次结构和 DI 模块差不多，这里只介绍和 DI 不同的地方。

## 1. 成员注释

为了演示成员注释，DO模块特意增加了子模块do\_state（第 19 到 31 行），该模块下有一个枚举类型aw\_dostate，第 24 到 26 行对枚举类型整体进行简介，第 28 和 29 行分别对其两个成员进行注释。有成员注释的结构体、枚举和类生成的文档看起来如图 5.2 所示。

## 2. 函数注释

第 38 到 45 行展示了更为详细的函数文档，如图 5.3 所示为函数aw\_do\_set的HTML

### 枚举类型说明

#### enum aw\_dostate

DO输出逻辑状态

**枚举值:**

**AW\_DOST\_LOW** DO状态为逻辑低电平

**AW\_DOST\_HIGH** DO状态为逻辑高电平

图 5.2 枚举、结构体等及其成员文档样式

形式文档效果图。

参数注释“@param”后紧跟“[in]”表示输入参数，可以用“[out]”表示输出参数，“[in,out]”表示双向参数。

第 41 到 43 还对参数 state 的有效值使用“@arg”进行了罗列，同时使用“@b”指示参数用粗体显示。

第 44 行使用“@return”对返回值进行描述，而没有使用“@retval”进行罗列，用于返回值太多不方便进行一一罗列的情况。

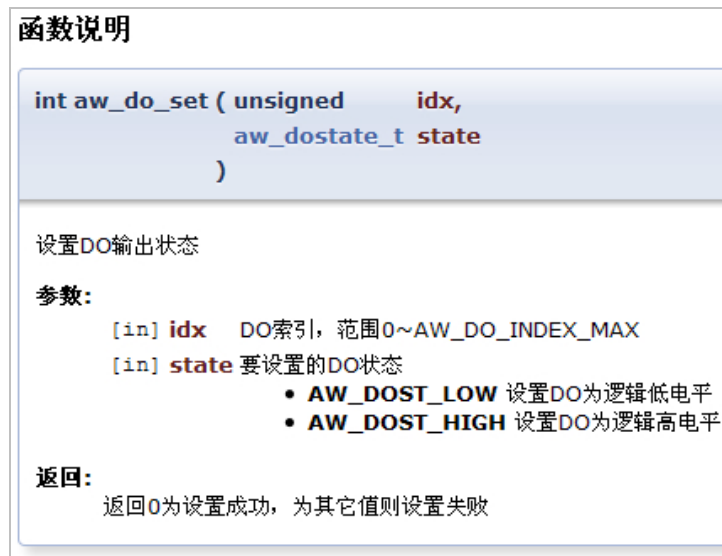


图 5.3 函数 aw\_do\_set 文档

### 5.2.3 生成文档

关于怎么生成文档这里不再赘述，请参考前面 4.2 和 4.3 小节。

## 6. 注释的书写建议

通过前面的介绍，对于 Doxygen 的使用和如何编写文档化注释有了一定的了解。Doxygen 能帮助我们生成格式漂亮的文档，但文档内容的质量还是靠程序员书写的注释来保证。

### 6.1 注释原则

注释应该怎么写，写多还是写少？过多的注释甚至会干扰对代码的阅读。

写注释的一个总的原则就是注释应该尽量用来表明作者的意图，至少也应该是对一部分代码的总结，而不应该是对代码的重复或者解释。对代码的重复或者解释的注释，看代码可能更容易理解。

推荐的写注释的过程是首先使用注释勾勒出代码的主要框架，然后根据注释撰写相应的代码。对各种主要的数据结构、输出的函数、多个函数公用的变量进行详细地注释。对代码中控制结构，单一目的的语句集进行注释。下面是一些写注释时需要注意的要点：

- (1) 避免对单独语句进行注释；
- (2) 通过注释解释为什么这么做、或者要做什么，使代码的读者可以只阅读注释理解代码；
- (3) 对读者可能会有疑问的地方进行注释；
- (4) 对数据定义进行注释，而不是对其使用过程进行注释；
- (5) 对于难于理解的代码，进行改写，而不要试图通过注释加以说明；
- (6) 对关键的控制结构进行注释；
- (7) 对数据和函数的边界、使用前提等进行注释。

这里插入一个题外话，一行代码究竟最长为多少算合理？为了方便阅读兼照顾一些老的低分辨率显示器，以不超过 78 列为宜。如图 6.1 所示为太长的代码行，注释已经超出了编辑窗口，必须拖动下面的滑动条才能阅读右边的注释，但拖到右边阅读注释，左边的代码又有部分被遮住了。

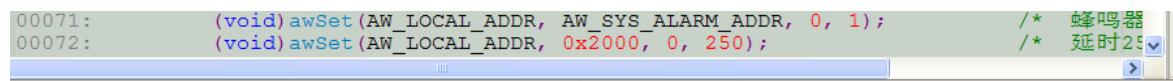


图 6.1 过长的代码行

### 6.2 对比示例

这里引用一个十分简单的示例来对比代码注释的优劣。

如程序清单 6.1 所示为摘自 AnyWhere 示例代码中的一小段蜂鸣器控制代码，每行代码后都有注释，看完每行的注释，再结合“for(;;)”，可以推出作者的意图是控制蜂鸣器以 2Hz 的频率鸣叫。虽然注释写了很多，每行都有，但我们还是要通读代码再推测才知道这段代码的意图。

程序清单 6.1 蜂鸣器示例

```
1.  for(;;) {
2.      (void)awSet(AW_LOCAL_ADDR, AW_SYS_ALARM_ADDR, 0, 0); /* 蜂鸣器停止报警 */
3.      (void)awSet(AW_LOCAL_ADDR, 0x2000, 0, 250); /* 延时 250 毫秒 */
4.      (void)awSet(AW_LOCAL_ADDR, AW_SYS_ALARM_ADDR, 0, 1); /* 蜂鸣器报警 */
```

```
5.      (void)awSet(AW_LOCAL_ADDR, 0x2000, 0, 250);          /* 延时 250 毫秒 */
6.  }
```

如程序清单 6.2所示是程序清单 6.1一个改进版本，首先第 1 到 3 行把Anywhere中晦涩难读的操作封装成有意义的宏，第 4 到 6 行给出了整个代码段“做什么”的解释，简单明了，作者也不必对每行代码进行繁琐的注释。知道了代码段是“做什么”的，再来阅读后面的代码就愉快多了。

程序清单 6.2 改进的蜂鸣器示例

```
1.  #define buzzer_on()  awSet(AW_LOCAL_ADDR, AW_SYS_ALARM_ADDR, 0, 1) /**< 蜂鸣器鸣叫 */
2.  #define buzzer_off() awSet(AW_LOCAL_ADDR, AW_SYS_ALARM_ADDR, 0, 0) /**< 蜂鸣器停止鸣叫 */
3.  #define mdelay(ms)      awSet(AW_LOCAL_ADDR, 0x2000, 0, ms)          /**< 延时 ms 毫秒 */
4.  /*
5.   * 蜂鸣器以 2Hz 频率鸣叫
6.   */
7.  for (;;) {
8.      (void)buzzer_off();
9.      (void)mdelay(250);
10.     (void)buzzer_on();
11.     (void)mdelay(250);
12. }
```



## 7. 生成离线文档

前面介绍的是用 Doxygen 生成 HTML 文档，这个是在线文档，可有时我们需要生成离线的文档随产品一起发布。Doxygen 支持生成离线的 CHM、LaTeX、RTF、XML 和 unix man-page 等格式文档。

### 7.1 CHM格式

CHM是英语“Compiled Help Manual”的简写，即“已编译的帮助文件”。CHM是微软新一代的帮助文件格式，利用HTML作源文，把帮助内容以类似数据库的形式编译储存。

HTML 格式转 CHM 格式可以使用微软的 HTML Help Workshop 软件，假设 HTML 文档已经生成并存放在目录 html 下，跟着下面的步骤生成 CHM 文档。

#### 7.1.1 生成CHM索引

在 Doxygen 的图形界面中选择“Wizard”标签页，选择“Output”项，在右边窗口中勾选“HTML”和“prepare for compressed HTML (.chm)”。

选择“Run”标签页，单击“Run doxygen”即可生成转换 CHM 所需的文件，默认情况下生成的文件都存放在工作目录的 html 目录下面。

#### 7.1.2 建立CHM工程

跟着下面的步骤建立 CHM 工程。

##### 1. 设置工程名

打开 HTML Help Workshop 软件，选择菜单“File/new...”，在弹出的对话框中选择“Project”，单击“OK”按钮，将弹出创建新工程向导。

单击“下一步”，进入目标文件名设置对话框，单击“Browse...”，在弹出的对话框中选择目标文件存放路径和文件名，如设置为 html 目录，文件名为 test.hhp，然后单击“打开”。

##### 2. 设置文件类型

单击“下一步”，进入选择文件类型对话框，勾选所有的复选框。

##### 3. 设置内容表文件

单击“下一步”，进入“Table of contents”对话框，单击“Browse...”，在文件浏览对话框中选择 html 目录下的 index.hhc。

##### 4. 设置索引文件

单击“下一步”，进入“Index”对话框，单击“Browse...”，在文件浏览对话框中选择 html 目录下的 index.hhk。

##### 5. 设置HTML文件路径

单击“下一步”，进入“HTML Files”对话框，单击“Add...”，在弹出的选择文件对话框中选择目录 html 下的所有文件（可使用 Ctrl + A 快捷键），然后单击“打开”。

单击“下一步”，然后单击“完成”。

#### 7.1.3 编译CHM工程

选择菜单“File/Compile...”，在弹出的对话框中单击“Compile”。

#### 7.1.4 查看CHM文件



Windows 已经集成了对 CHM 的支持，只需双击 html 目录下 CHM 后缀名的文件如 test.chm 即可。

如果看到中文变成了乱码，请看 FAQ，这是文件编码引起的，可能要把 HTML 文件转换成中文 Windows 默认的 GBK 编码。

## 7.2 RTF和word格式

RTF (Rich Text Format, 富文本格式) 是由微软公司开发的跨平台文档格式，以纯文本描述内容，能够保存各种格式信息，可以用写字版，Word 等创建。

### 1. 加载配置文件

打开 Doxygen 的图形前端，选择 “File/Open...”，在打开配置文件对话框中选择一个已经存在的配置文件打开。

### 2. 选择RTF输出

选择 “Wizard” 标签页，再选择该页下 “Output” 项，在右边窗口找到复选框 “Rich Text Format (RTF)”，勾上。

### 3. 生成RTF文档

选择 “Run” 标签页，单击 “Run doxygen”，即可在工作目录下看到一个 “rtf” 目录，该目录下生成了一个 RTF 文件 “refman.rtf”。

### 4. RTF转word

使用微软的 word 打开 refman.rtf，然后另存为 word 格式即转换成了 word 文档。

生成的 RTF 或者转换后的 word 格式的文档，其样式并不尽如人意，比起 HTML 格式的效果差很远。目前暂时还没有找到一个好的方法把 html 无损的转换成 word 格式。

## 7.3 PDF格式

PDF 可以由 word 或 HTML 转换得到。word 一般自带了转 PDF 插件，使用十分方便。HTML 转 PDF 就要借助第三方工具，目前暂时也没有找一个好用的工具。

## 8. FAQ

### 1. 为什么在浏览器中看到很多乱码

可能源码文件中有 Doxygen 不能识别的编码，或者你的浏览器没有自动使用 UTF-8 编码进行编码显示。

### 2. 如何对生成文档进行更高级的控制

请参考 Doxygen 的用户手册，这些高级的控制不在本文的讨论范围内。

### 3. 如何把文档中页脚的版权声明替换成自己的

需要自行编写一个 HTML 文件做页脚，如程序清单 8.1 所示为页脚 HTML 的内容形式，假设文件名为“html\_footer”。

选择 Doxygen 图形前端的“Expert”标签页，在“Topics”下选择“HTML”项，在右边窗口中找到“HTML\_FOOTER”项，浏览选择“html\_footer”文件。

程序清单 8.1 定制 HTML 页脚示例

```
1. <hr width="80%">
2. <p><center>广州致远电子有限公司，&copy 版权所有，保留所有权利。
3. 本文档由 Doxygen $doxygenversion 于 $date 自动生成</center></p>
4. </body>
5. </html>
```

### 4. 如何自定义文档样式

自定义文档样式需要自行编写 HTML 头，并在头中指定自己的样式文件（css），然后类似于 FAQ3 那样把“HTML\_HEADER”设置为自己的 HTML 头。



## 9. 免责声明

此应用笔记的著作权属于广州致远电子股份有限公司。任何个人或者是单位，未经广州致远电子股份有限公司同意，私自使用此用户手册进行商业往来，导致或产生的任何第三方主张的任何索赔、要求或损失，包括合理的律师费，由您赔偿，广州致远电子股份有限公司与合作公司、关联公司不承担任何法律责任。

广州致远电子股份有限公司特别提醒用户注意：广州致远电子股份有限公司为了保障公司业务发展和调整的自主权，广州致远电子股份有限公司拥有随时自行修改此用户手册而不需通知用户的权利，如有必要，修改会以通告形式公布于广州致远电子股份有限公司网站重要页面上。