

```
/**
 * @file      led.h
 * @author    JonesLee
 * @email     Jones_Lee3@163.com
 * @version   V4.01
 * @date      07-DEC-2017
 * @license   GNU General Public License (GPL)
 * @brief     Controller Area Network
 * @detail    detail
 * @attention
 * This file is part of OST.
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 3 as
 * published by the Free Software Foundation.
 * You should have received a copy of the GNU General Public License
 * along with OST. If not, see <http://www.gnu.org/licenses/>.
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 * @htmlonly
 * <span style="font-weight: bold">History</span>
 * @endhtmlonly
 * Version|Author|Date|Describe
 * -----|----|-----|-----
 * V3.3|Jones Lee|07-DEC-2017|Create File
 * <h2><center>&copy;COPYRIGHT 2017 WELLCASA All Rights Reserved.</center></h2>
 */
```

Doxygen快速入门



路明

明君

53 人赞同了该文章

综述

我们在编写代码的时候，最头疼的就属于说明书了，很多代码一边写具体代码，一边写说明书，Doxygen主要解决说明书问题，可以在我们写代码的时候讲注释转化为说明书，Graphviz主要是用于图形展示，html help workshop主要使用生成CHM文档。

1.Doxygen

Doxygen能将程序中的特定批注转换成为说明文件。它可以依据程序本身的结构，将程序中按规范注释的批注经过处理生成一个纯粹的参考手册，通过提取代码结构或借助自动生成的包含依赖图（include dependency graphs）、继承图（inheritance diagram）以及协作图（collaboration diagram）来可视化文档之间的关系，Doxygen生成的帮助文档的格式可以是CHM、RTF、PostScript、PDF、HTML等。

2.graphviz

Graphviz(Graph Visualization Software)是一个由AT&T实验室启动的开源工具包,用于绘制DOT语言脚本描述的图形。要使用Doxygen生成依赖图、继承图以及协作图，必须先安装graphviz软件。

3.HTML Help WorkShop

微软出品的HTML Help WorkShop是制作CHM文件的最佳工具，它能将HTML文件编译生成CHM文档。Doxygen软件默认生成HTML文件或Latex文件，我们要通过HTML生成CHM文档，需要先安装HTML Help WorkShop软件，并在Doxygen中进行关联

见示例效果图。

AHRS_Code_Pro

V4.01

首页

模块

结构体

文件

模块

这里列出了所有模块:

APP

ALGORITHM

NETWORK

Network

MSG

CONTROL

SENSOR

STM32F10x_StdPeriph_Template

BSP

CXX

BMP180

BMX055

CAN

IIC

IO

LED

PWM

SOCKET

SPI

TIME

USART

W5500

Global

Global_Macros

Global_Type

Based_Type_Defined

Coordinate_Transformation_Element

Move_Control_Elementt

XXXX

Describing hardware devices

24CXX

BMP180

BMX055

Controller Area Network

Inter-Integrated Circuit

IO

Light Emitting Diode

Pulse Width Modulation

SOCKET

Serial-Peripheral-Interface

Universal Synchronous/Asynchronous Receiver/Transmitter

W5500

This module defines the structure used for engineering

This module defines the structure used for engineering

This module defines the basic elements, such as symbol data, unsigned data, real and imaginary

Define coordinate transformation elements, such as coordinates, posture, four elements, etc

The module defines the motion control elements, such as position, vector, speed, acceleration, and so on

[详细页码 1 2 3]

知乎 @路明

安装

Doxygen下载 (doxygen-1.8.7-setup.exe) :

stack.nl/~dimitri/doxyg...

graphviz (for windows) 下载:

graphviz.org/Download.....

HTML Help WorkShop (1.32) 下载:

download.microsoft.com/...

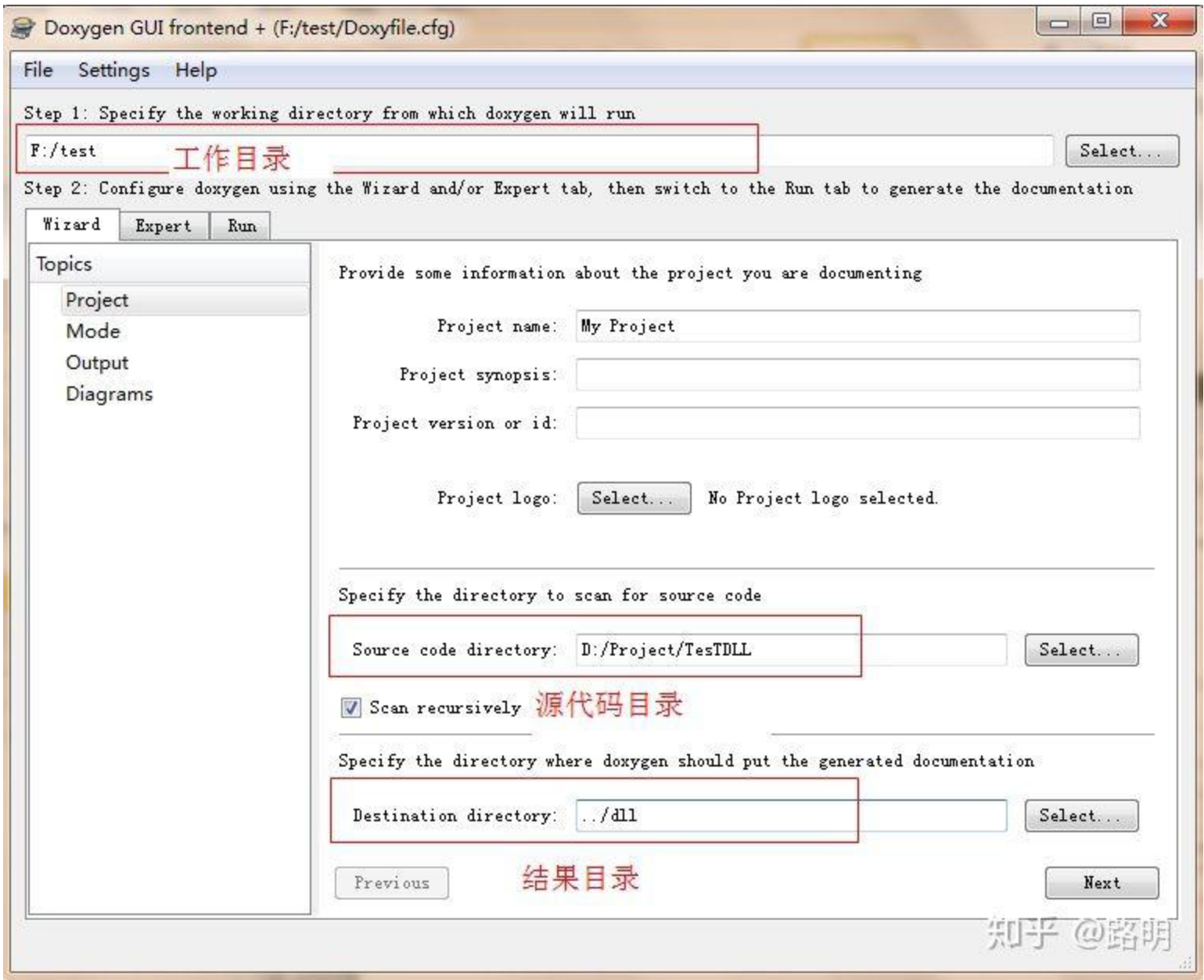
软件安装都选择默认方式，点击下一步直至安装完成。安装完后进行Doxygen配置时需要关联graphviz和HTML Help WorkShop的安装路径。

配置

我们的所有配置都在Doxywizard中进行，生成参考手册是通过运行Doxywizard得到。

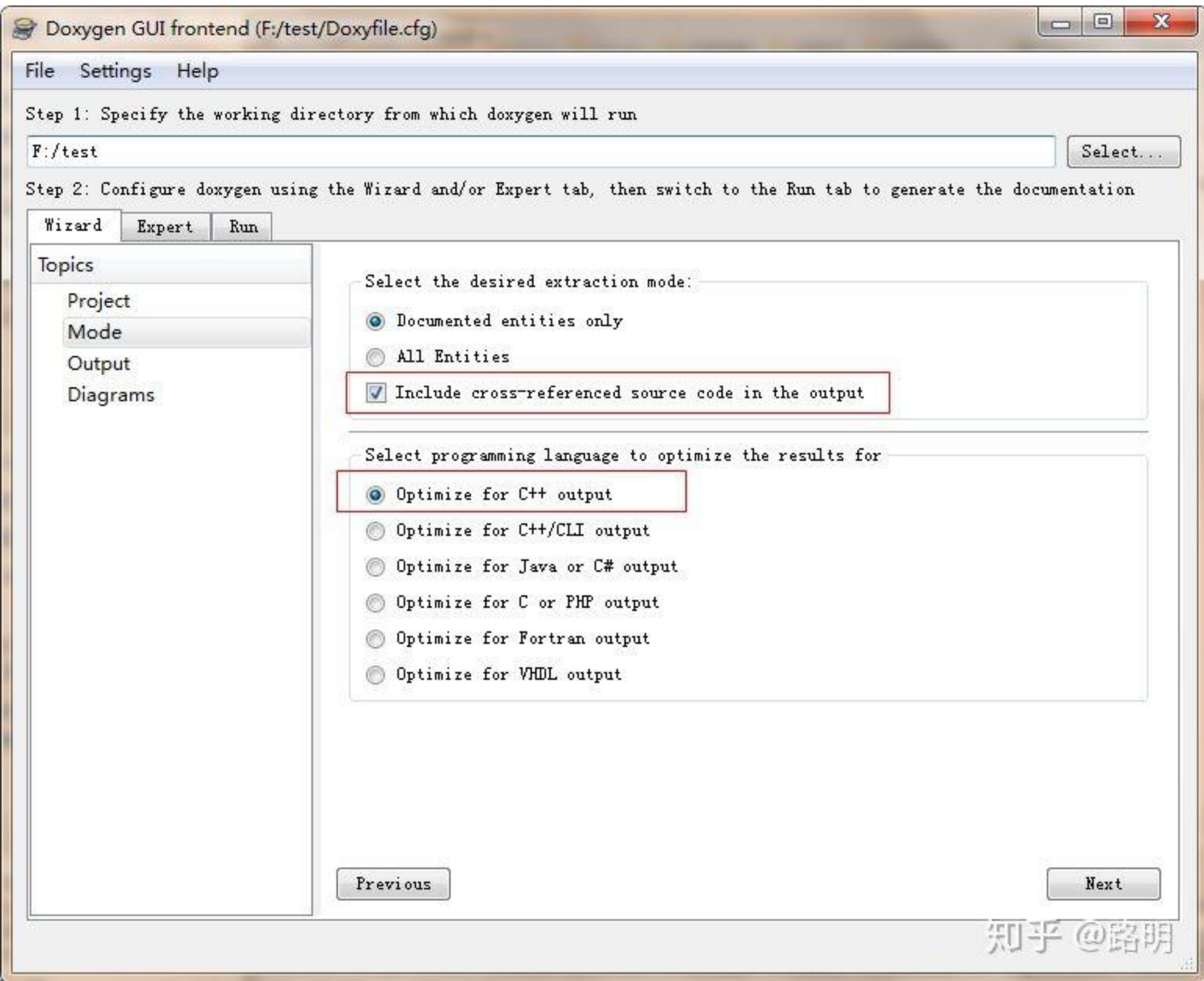
1.Wizard->Project

Wizard->Project最重要的是工作目录，源代码目录，生成参考文件目录三处的设定，其它项目名称、项目简介、版本和标识可以依照实际情况选填。工作目录是新建的一个目录，在配置完成之后可以把配置文件存在这个目录里，每次从这个目录中导入配置文件 (.cfg)， 然后进行说明文档生成。源代码目录和最终的结果目录在每一次运行Doxywizard时都进行设定。



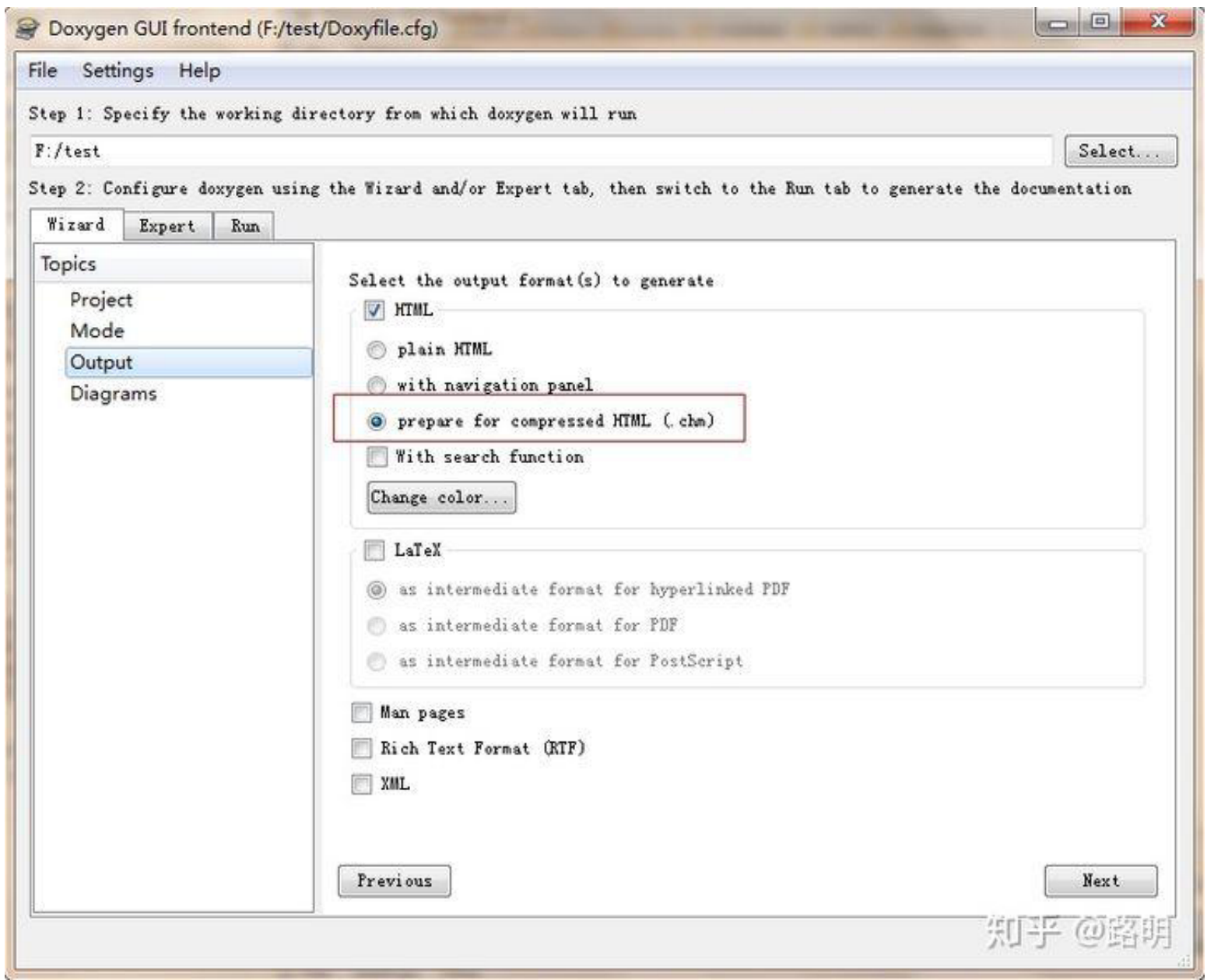
2.Wizard->Mode

选择编程语言对应的最优化结果，按照编程语言选择。



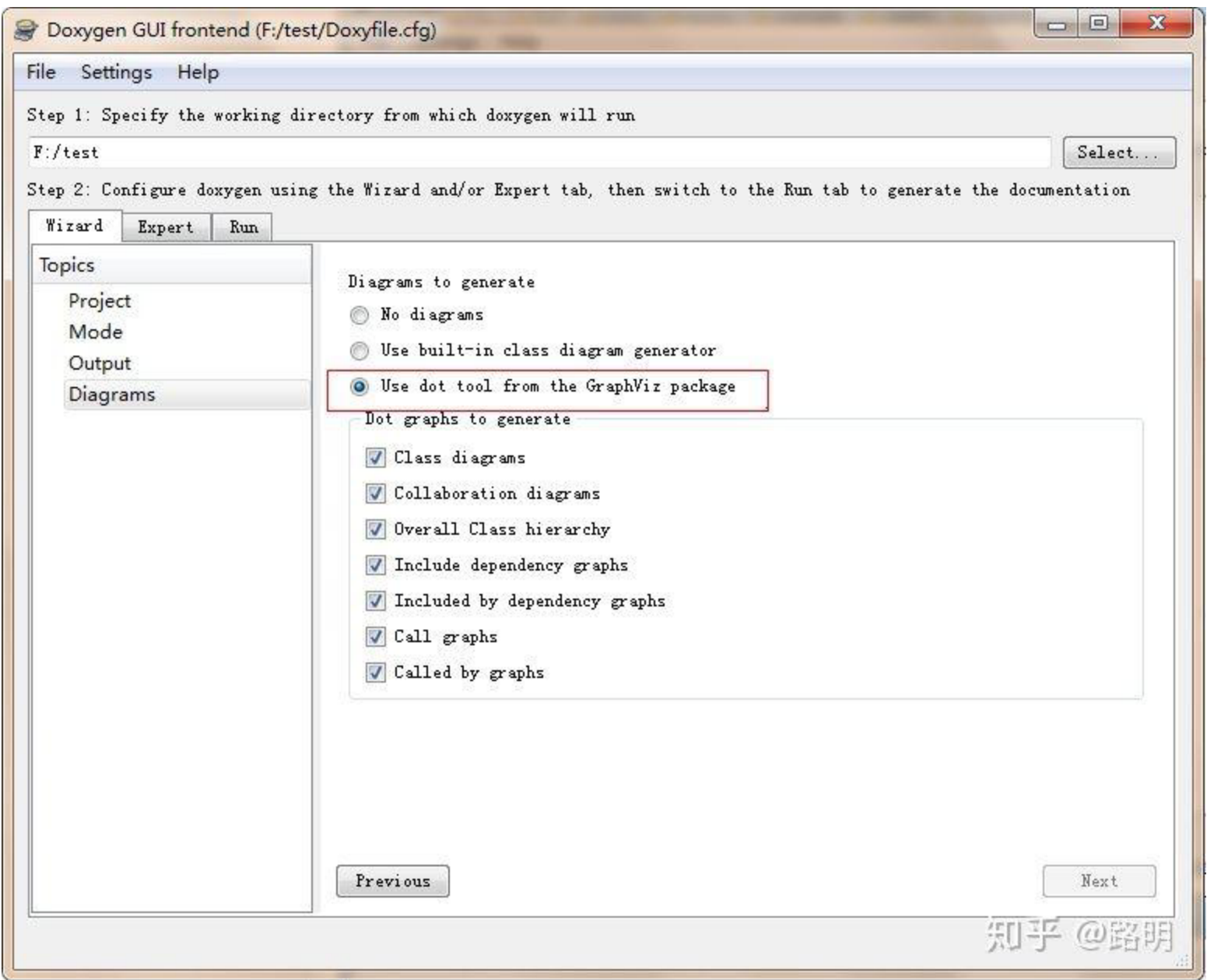
3.Wizard->Output

选择输出格式，选HTML下的(.chm)项，为最后生成chm做准备。由于不需要LaTeX结果，不选此项。



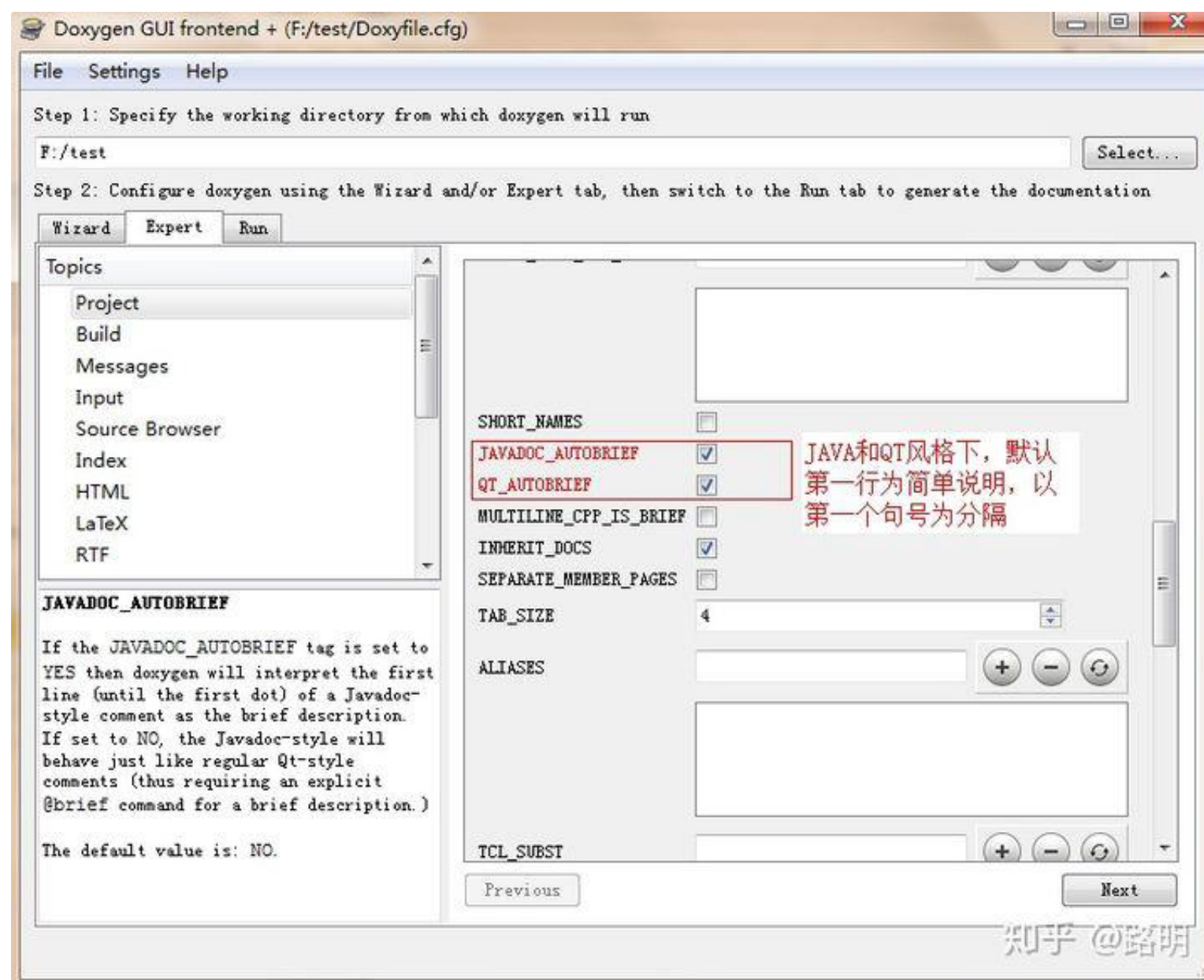
4.Wizard->Diagrams

选择dot tool项，通过GraphViz来作图。



5.Expert->Project

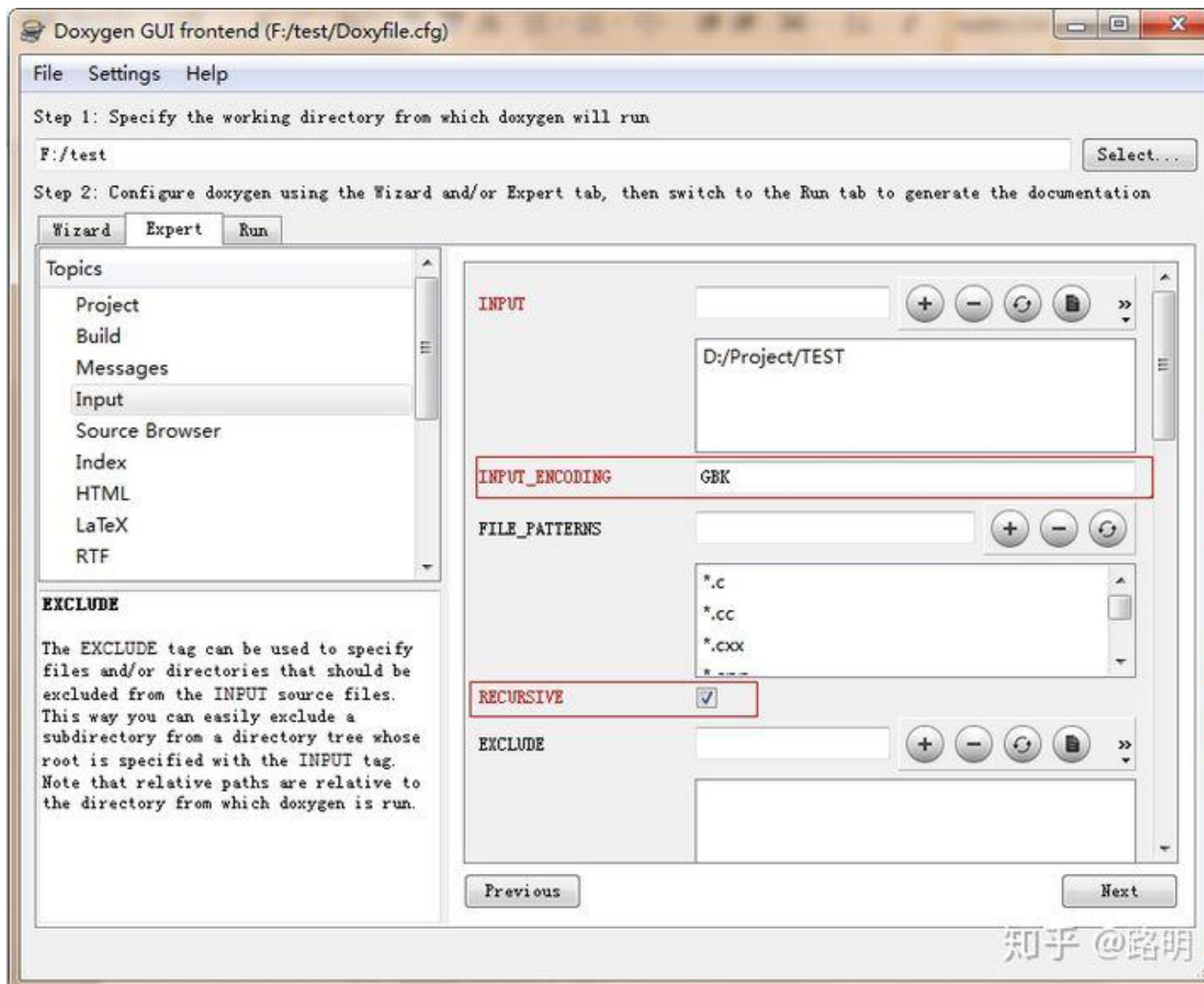
选择输出目录，选着输出语言。如果代码中采用了中文注释，此处选择为中文。
向下拉滑条，看见有JAVADOC_AUTOBRIEF和QT_AUTOBRIEF两个框，如果勾选了，在这两种风格下默认第一行为简单说明，以第一个句号为分隔；如果不选，则需要按照Doxygen的指令@brief来进行标准注释。



6.Expert->Input

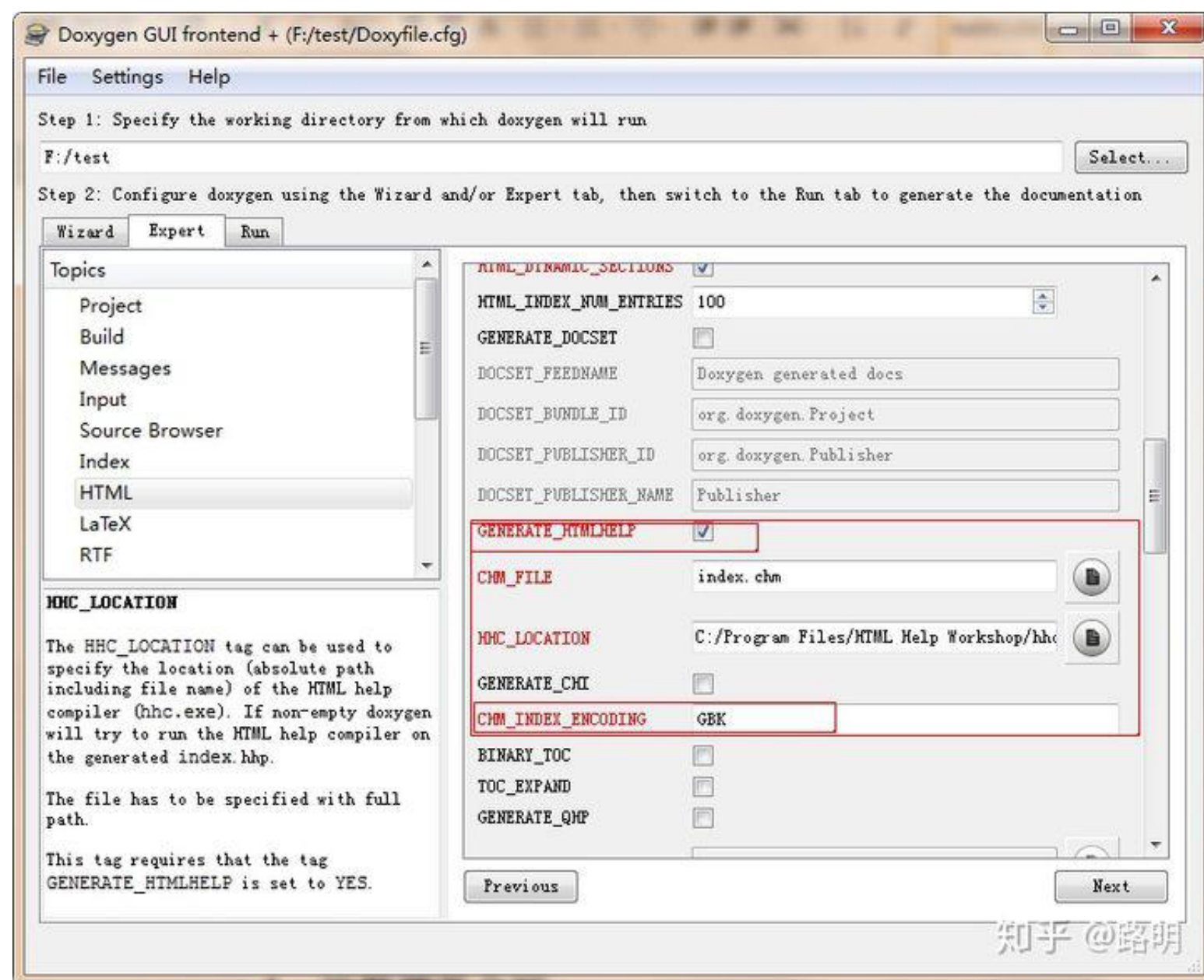
将输入编码方式改为GBK方式，确保输出中不会由于UTF-8方式导致乱码。

最后也是经常遇到的问题就是DoxyGen生成的CHM文件的左边树目录的中文变成了乱码。这个只需要将chm索引的编码类型修改为GB2312即可。在HTML的CHM_INDEX_ENCODING中输入GB2312即可。



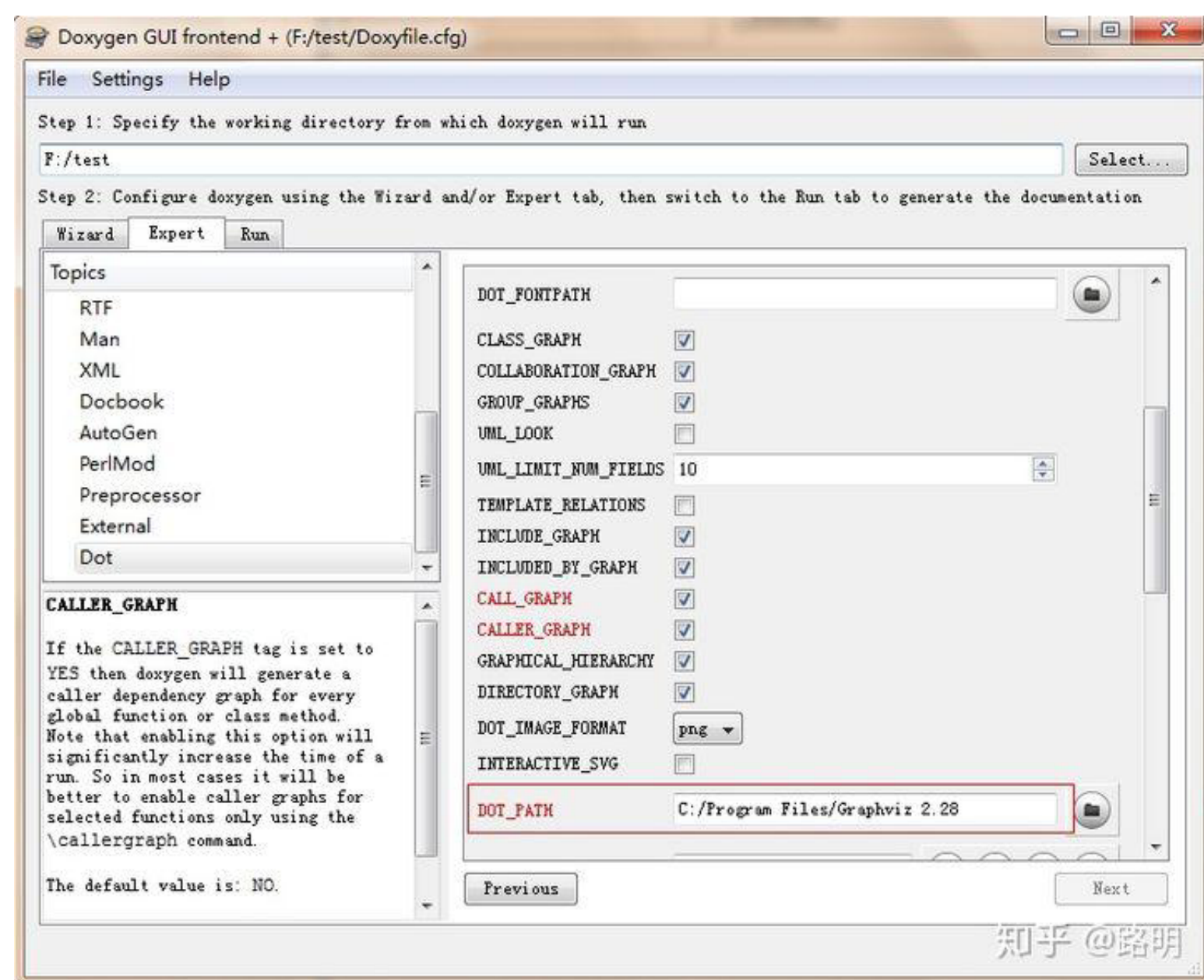
7.Expert->HTML

勾选生成HTMLHELP项，输入生成CHM名称，在HHC_LOCATION中填入HTMLHELP WORKSHOP安装目录中hhc.exe的路径，将chm编码方式改为GBK方式，与第（6）步中的输入编码方式一致。

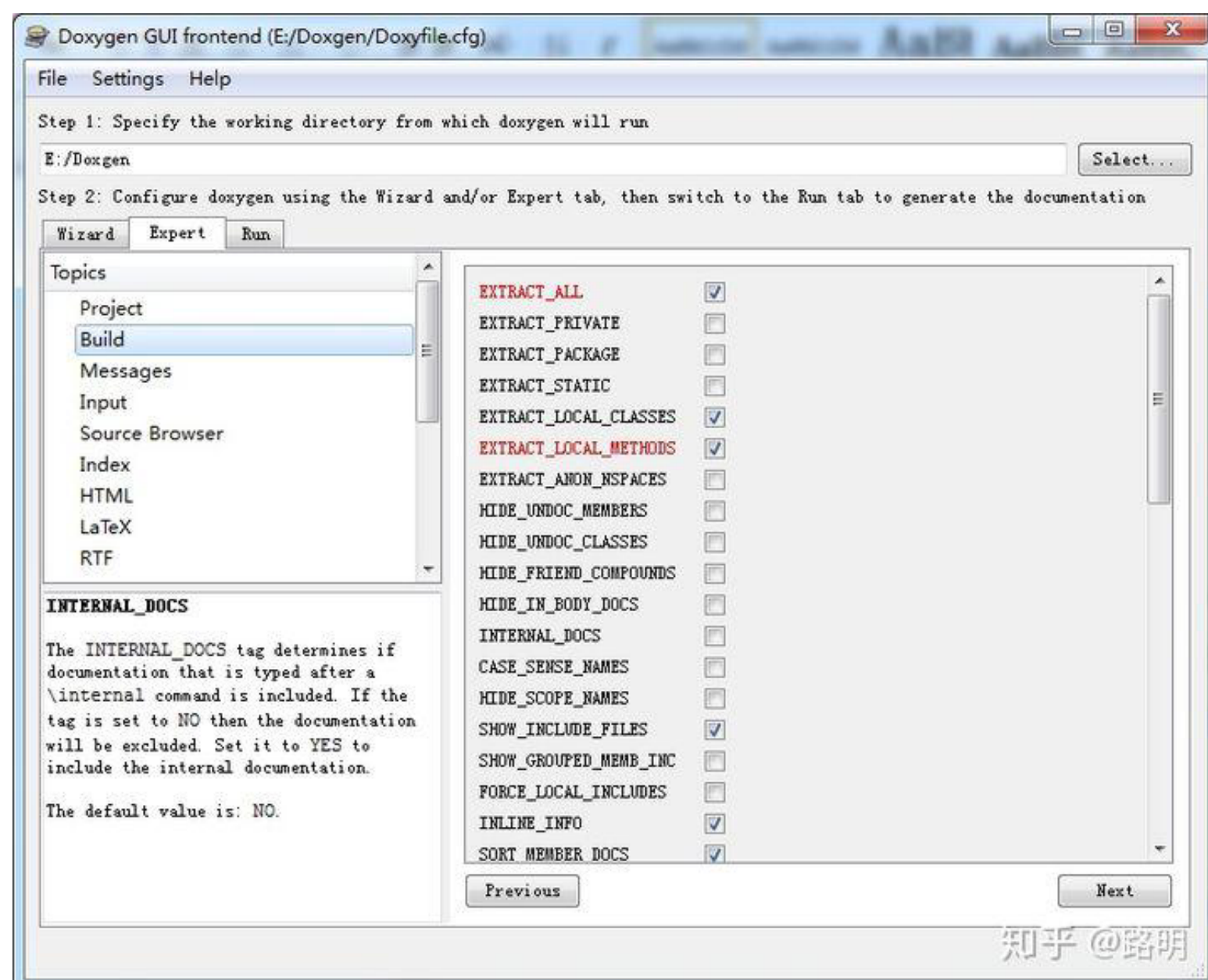


8.Expert->Dot

在Dot_PATH中填写GraphViz的安装路径。

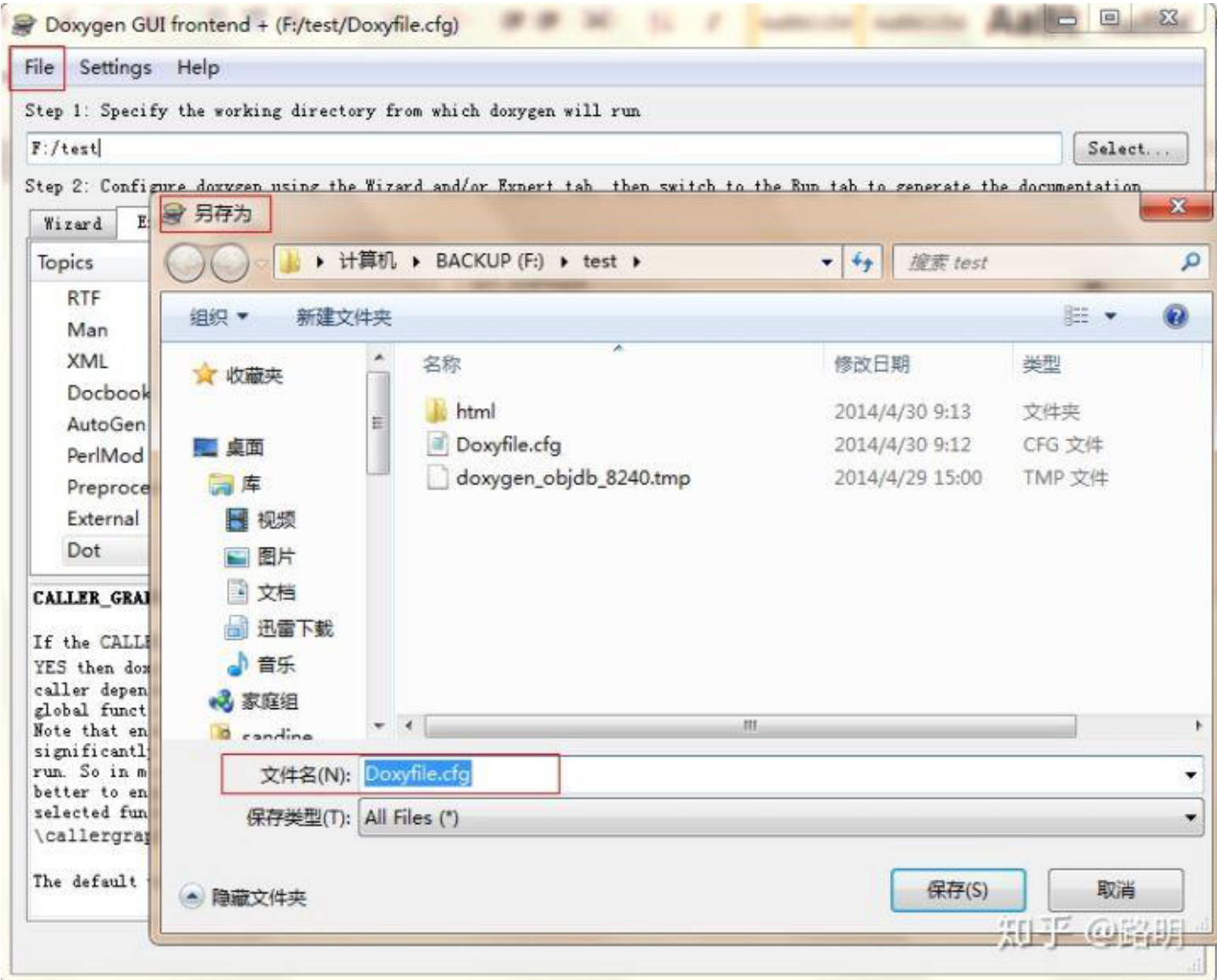


需要在build中配置EXTRACT_ALL和LOCAL_METHODS才能生成所有的变量和函数。



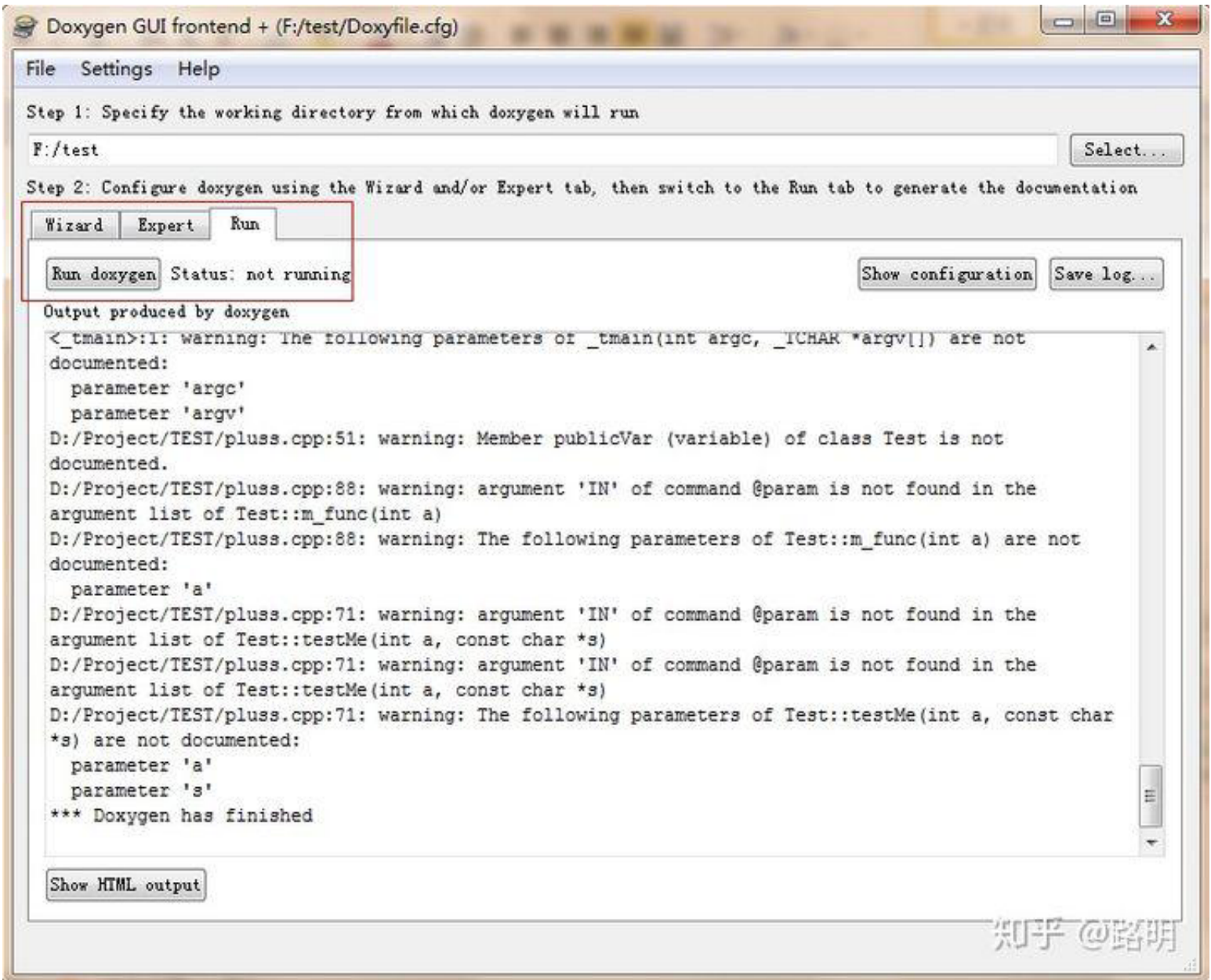
9.存储配置信息

存储配置信息。到上一步Doxygen已经完全配置好，可以在Run中点击运行了，但为了保存以上配置信息，可以将配置好的文件存一个.cfg文件，之后再运行Doxygen时只需要将该文件用Doxygen打开，改变第（1）步中的输入、输出目录及工程的信息再运行。File->Save as, 取一个名，默认为Doxyfile，加.cfg,点击保存。如果需要改变配置文件，改动之后再Save替换之前的配置文件即可。



10.Run->Run Doxygen

即可运行Doxygen，运行完成后在输出目录中的html文件夹中找到index.chm文件即为输入代码的文档说明。



规范

规范综述

简要的说，Doxygen注释块其实就是在C、C++注释块的基础添加一些额外标识,使Doxygen把它识别出来,并将它组织到生成的文档中去。在每个代码项中都可以有两类描述：一种就是brief描述,另一种就是detailed。两种都是可选的，但不能同时没有。简述(brief)就是在一行内简述地描述。而详细描述(detailed)则提供更长,更详细的文档。在Doxygen中,主要通过以下方法将注释块标识成详细(detailed)描述:

JavaDoc风格，在C风格注释块开始使用两个星号'*':

```
/**
 * ... 描述 ...
 */
```

Qt风格代码注释,即在C风格注释块开始处添加一个叹号'!':

```
/*!  
 * ... 描述 ...  
 */
```

使用连续两个以上C++注释行所组成的注释块, 而每个注释行开始处要多写一个斜杠或写一个叹号：

```
///  
/// ... 描述 ...  
///
```

同样的，简要说明（brief）有也有多种方式标识，这里推荐使用@brief命令强制说明，例如

```
/**  
 * @brief 简要注释Brief Description.  
 */  
  
/**  
 * @brief 简要注释Brief Description.  
 */
```

注意以下几点：

- 1.Doxygen并不处理所有的注释，doxygen重点关注与程序结构有关的注释，比如：文件、类、结构、函数、全局变量、宏等注释，而忽略函数内局部变量、代码等的注释。

2.注释应写在对应的函数或变量前面。JavaDoc风格下，自动会把第一个句号"."前的文本作为简要注释，后面的为详细注释。你也可以用空行把简要注释和详细注释分开。注意要设置JAVADOC_AUTOBRIEF或者QT_AUTOBRIEF设为YES。

3.先从文件开始注释，然后是所在文件的全局函数、结构体、枚举变量、命名空间→命名空间中的类→成员函数和成员变量。

4.Doxygen无法为DLL中定义的类型导出文档。

常用指令

指令说明

指令	说明
@file	档案的批注说明。
@author	作者的信息
@brief	用于class 或function的简易说明 eg: @brief 本函数负责打印错误信息串
@param	主要用于函数说明中，后面接参数的名字，然后再接关于该参数的说明
@return	描述该函数的返回值情况 eg: @return 本函数返回执行结果，若成功则返回TRUE，否则返回FLASE
@retval	描述返回值类型
@note	注解
@attention	注意
@warning	警告信息
@enum	引用了某个枚举，Doxygen会在该枚举处产生一个链接eg: @enum CTest::MyEnum
@var	引用了某个变量，Doxygen会在该枚举处产生一个链接
@class	引用某个类，格式：@class [] []eg:@class CTest "inc/class.h"
@exception	可能产生的异常描述 eg: @exception 本函数执行可能会产生超出范围的异常

模块

Modules是一种归组things在分离的page上的方式。组的成员可以是file，namespace，classes，functions，variables，enums，typedefs和defines，但也可以是其他groups。
要定义一个group，应该在一个特殊注释块放置\defgroup。命令的第一个参数应该是唯一标志该group的标签。要将一个entity归为某个group的一个member，在entity前放置ingroup命令。第二个参数是group的title。
要避免在注释中每个member前放置ingroup命令，可以将member用@{和@}封装起来。@{@}标记可以放置group的注释中，也可以在一个独立的注释块使用这些group的标记符号groups也可以嵌套。
如果多次使用一个group标签，将会出错。如果不希望doxygen强行执行唯一标签，可以使用\addtogroup而非\defgroup。运作方式和\defgroup很像，但是如果group已经定义，它默认向已存在的注释中添加一个新的项。Group的title对此命令是可选的，也可以考虑使用它。

```
/*  
 * @defgroup 模块名 模块的说明文字  
 * @{  
 */  
    ... 定义的内容 ...  
/** @} */  
// 模块结尾这样可以在其他地方以更加详细的说明添加members到一个group。
```

如果一个compound（例如一个class或file）有多个members，通常我们希望将其group。Doxygen已经可以自动按照类型和protection级别将这些things归组在一起，但可能你会认为仅仅这样是不够的或者这种缺省的方法是错误的。例如你认为有不同（语法）的类型需要归入同一个group（语意）。

这样定义一个member group：

```
//@{
...
//@}
```

块或者使用。

```
/*@{*/
...
/*@}*/
```

注释实例

1.文件注释

举例说明如下，在代码文件头部写上这段注释。可以看到可以标注一些文本名称、作者、邮件、版本、日期、介绍、以及版本详细记录。

```
/**
 * @file      sensor.c
 * @author    JonesLee
 * @email     Jones_Lee3@163.com
 * @version   V4.01
 * @date      07-DEC-2017
 * @license   GNU General Public License (GPL)
 * @brief     Universal Synchronous/Asynchronous Receiver/Transmitter
 * @detail    detail
 * @attention
 * This file is part of OST.
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 3 as
 * published by the Free Software Foundation.
 * You should have received a copy of the GNU General Public License
 * along with OST. If not, see <http://www.gnu.org/licenses/>.
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 * @htmlonly
 * <span style="font-weight: bold">History</span>
 * @endhtmlonly
 * Version|Auther|Date|Describe
 * -----|----|-----|-----
 * V3.3|Jones Lee|07-DEC-2017|Create File
 * <h2><center>&copy;COPYRIGHT 2017 WELLCASA All Rights Reserved.</center></h2>
 */
```

通过doxygen编译后，打开网页即可显示如下， 以下见名知意，不在讲解。

详细描述

Universal Synchronous/Asynchronous Receiver/Transmitter detail

作者

JonesLee Jones_Lee3@163.com

版本

V4.01

日期

07-DEC-2017 GNU General Public License (GPL)

注意

This file is part of OST.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 3 as published by the Free Software Foundation.

You should have received a copy of the GNU General Public License along with OST. If not, see <http://www.gnu.org/licenses/>.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

History

Version	Auther	Date	Describe
V3.3	Jones Lee	07-DEC-2017	Create File

©COPYRIGHT 2017 WELLCASA All Rights Reserved.

2.类和成员注释

```
/**
 * @class <class-name> [header-file] [<header-name>]
```

```
* @brief brief description
* @author <list of authors>
* @note
* detailed description
*/
```

如果对文件、结构体、联合体、类或者枚举的成员进行文档注释的话,并且要在成员中间添加注释,而这些注释往往都是在每个成员后面。为此,可以使用在注释段中使用'<'标识。

```
int var; /**< Detailed description after the member */
```

对一个类的注释例子如下：

```
class Test
{
public:
    /** @brief A enum, with inline docs */
    enum TEnum
    {
        TVal1, /**< enum value TVal1. */
        TVal2, /**< enum value TVal2. */
        TVal3 /**< enum value TVal3. */
    }
    *enumPtr, /**< enum pointer. */
    enumVar; /**< enum variable. */
    /** @brief A constructor. */
    Test();
    /** @brief A destructor. */
    ~Test();
    /** @brief a normal member taking two arguments and returning an integer value. */
    int testMe(int a,const char *s);

    /** @brief A pure virtual member.
    * @param[in] c1 the first argument.
    * @param[in] c2 the second argument.
    * @see testMe()
```

3.函数注释

直接看案例如下，见明知意，不在赘述。

```
/**
    * @brief          can send the message
    * @param[in]      canx : The Number of CAN
    * @param[in]      id : the can id
    * @param[in]      p : the data will be sent
    * @param[in]      size : the data size
    * @param[in]      is_check_send_time : is need check out the time out
    * @note Notice that the size of the size is smaller than the size of the buffer.
    * @return
    *      +1 Send successfully \n
    *      -1 input parameter error \n
    *      -2 canx initialize error \n
    *      -3 canx time out error \n
    * @par Sample
    * @code
    *      u8 p[8] = {0};
    *      res_ res = 0;
    *      res = can_send_msg(CAN1,1,p,0x11,8,1);
    * @endcode
    */
extern s32 can_send_msg(const CAN_TypeDef * canx,
                        const u32 id,
                        const u8 *p,
                        const u8 size,
                        const u8 is_check_send_time);
```

打开注释网页后展示如下。

• can_send_msg()

s32 can_send_msg (const CAN_TypeDef * canx,
 const u32 id,
 const u8 * p,
 const u8 size,
 const u8 is_check_send_time
)

can send the message

参数
[in] canx : The Number of CAN
[in] id : the can id
[in] p : the data will be sent
[in] size : the data size
[in] is_check_send_time : is need check out the time out

注解
Notice that the size of the size is smaller than the size of the buffer.

返回
+1 Send successfully
-1 input parameter error
-2 canx initialize error
-3 canx time out error

Sample
u8 p[8] = {0};
res_res = 0;
res = can_send_msg(CAN1, 1, p, 0x11, 8, 1);

知乎 @路明

4.枚举注释

直接看案例如下，见明知意，不在赘述。

```
/** bool */  
typedef enum  
{  
    false = 0, /**< FALSE 0 */  
    true = 1   /**< TRUE  1 */  
}bool;
```

枚举类型说明

• bool

enum bool

bool

枚举值	
false	FALSE 0
true	TRUE 1

知乎 @路明

5.全局变量和宏

6.模块注释

Group定义命令的优先级（从高到低）：\ingroup, \defgroup, \addtogroup, \weakgroup。而\weakgroup很像一个有低优先级的\addtogroup。它被设计为实现一个“lazy”的group定义方法：可以在.h文件中使用高优先级来定义结构，在.cpp文件中使用\weakgroup这样不会重复.h文件中的层次结构。在实际使用中，我们可以看到具体的网页展示如下。

首页	模块	结构体	文件	
模块				
这里列出了所有模块:				
				详细规则 1.2
APP				
ALGORITHM				
NETWORK	NETWORK_through_the_w5500			
Network	Network_through_the_w5500			
MSG				
CONTROL				
STM32F10x_StdPeriph_Template				
BSP	Describing hardware devices			
CXX	24CXX			
BMP180	BMP180			
BMX055	BMX055			
CAN	Controller Area Network			
IIC	Inter-Integrated Circuit			
IO	IO			
LED	Light Emitting Diode			
PWM	Pulse Width Modulation			
SOCKET	SOCKET			
SPI	Serial-Peripheral-Interface			
TIME				
USART	Universal Synchronous/Asynchronous Receiver/Transmitter			
W5500	W5500			
Global				
XXXX				
XXXXXXXXXXXX	XXXXXXXXXX			

在图中有个BSP下的LED模块，这个模块就是承载驱动文件LED。具体代码如下，为了显示效果，我把函数的注释删除了。

```
/**
 * @file      led.h
 * @author    JonesLee
 * @email     Jones_Lee3@163.com
 * @version   V4.01
 * @date      07-DEC-2017
 * @license   GNU General Public License (GPL)
 * @brief     Controller Area Network
 * @detail    detail
 * @attention
 * This file is part of OST. \n
 * This program is free software; you can redistribute it and/or modify \n
 * it under the terms of the GNU General Public License version 3 as \n
 * published by the Free Software Foundation. \n
 * You should have received a copy of the GNU General Public License \n
 * along with OST. If not, see <http://www.gnu.org/licenses/>. \n
 * Unless required by applicable law or agreed to in writing, software \n
 * distributed under the License is distributed on an "AS IS" BASIS, \n
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. \n
 * See the License for the specific language governing permissions and \n
 * limitations under the License. \n
 *
 * @htmlonly
 * <span style="font-weight: bold">History</span>
 * @endhtmlonly
 * Version|Auther|Date|Describe
 * -----|----|-----|-----
 * V3.3|Jones Lee|07-DEC-2017|Create File
 * <h2><center>&copy;COPYRIGHT 2017 WELLCASA All Rights Reserved.</center></h2>
 */
/** @addtogroup  BSP
 * @{
 */
/**
 * @brief Light Emitting Diode
 */
/** @addtogroup LED
 * @{
 */
#ifdef __LED_H__
#define __LED_H__
#ifdef __cplusplus
extern "C" {
#endif
#include "bsp.h"
extern s32 led_init(Led_TypeDef led);
extern s32 led_config(void);
extern s32 led_on(Led_TypeDef led);
extern s32 led_off(Led_TypeDef led);
extern s32 led_toggle(Led_TypeDef led);
#ifdef __cplusplus
}
#endif
#endif
/* __LED_H__ */
**/
```

```
* @}  
*/  
/**  
* @}  
*/  
  
/***** (C)COPYRIGHT 2017 WELLCASA All Rights Reserved. *****/
```

前面说明了是C语言的，C语言没有继承一说，在C++中有时候需要展示继承。如图

