# 代码注释格式与doxygen文档生成

Longhui Lai | 2025/04/10

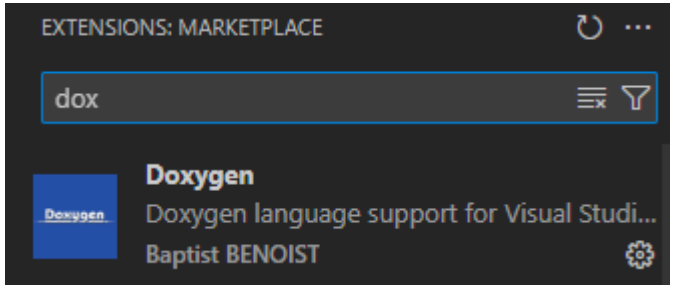

MAHLE

# Doxygen插件使用

在vscode中安装doxygen即可自动生成注释内容，可以在.vscode文件夹的setting.json文件中定义模板，为文件注释和函数注释生成一致的描述。
插入注释的快捷方式可以在setting.json中定义，比如在文件开头或函数前输入“/*”后 回车，即可生成以下模板内容：

文件开头注释：

```
/**
 * @file UTL.c
 * @brief
 * @author Longhui Lai (longhui.lai@mahle.com)
 * @version 1.0.0
 * @date 2025-04-10
 *
 * @copyright Copyright (c) 2025 MAHLE Automotive Technologies (Suzhou) Co., Ltd.
 *
 */
```

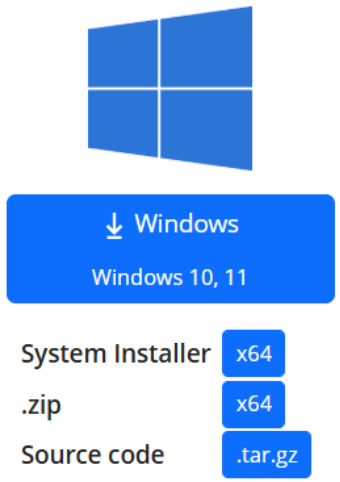


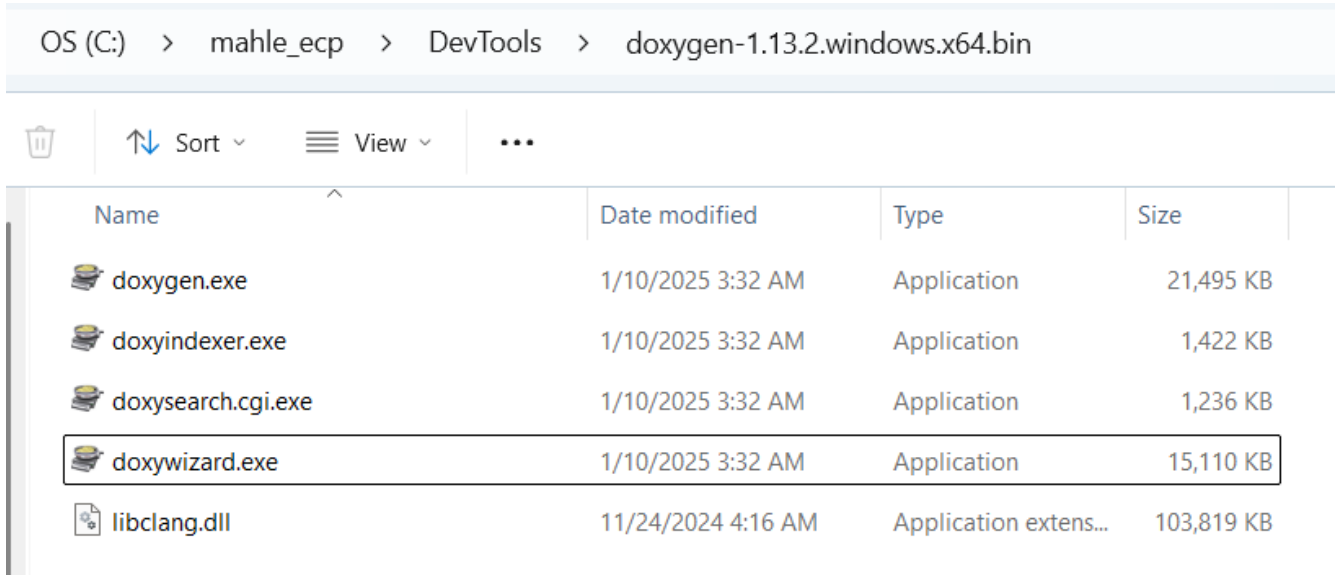

函数注释

```
/**
 * @brief
 * @param  value
 * @param  threshold
 * @return boolean
 */
boolean UTL_CompareFloatIsBelow(float32 value, float32 threshold)
```

# Doxygen软件安装

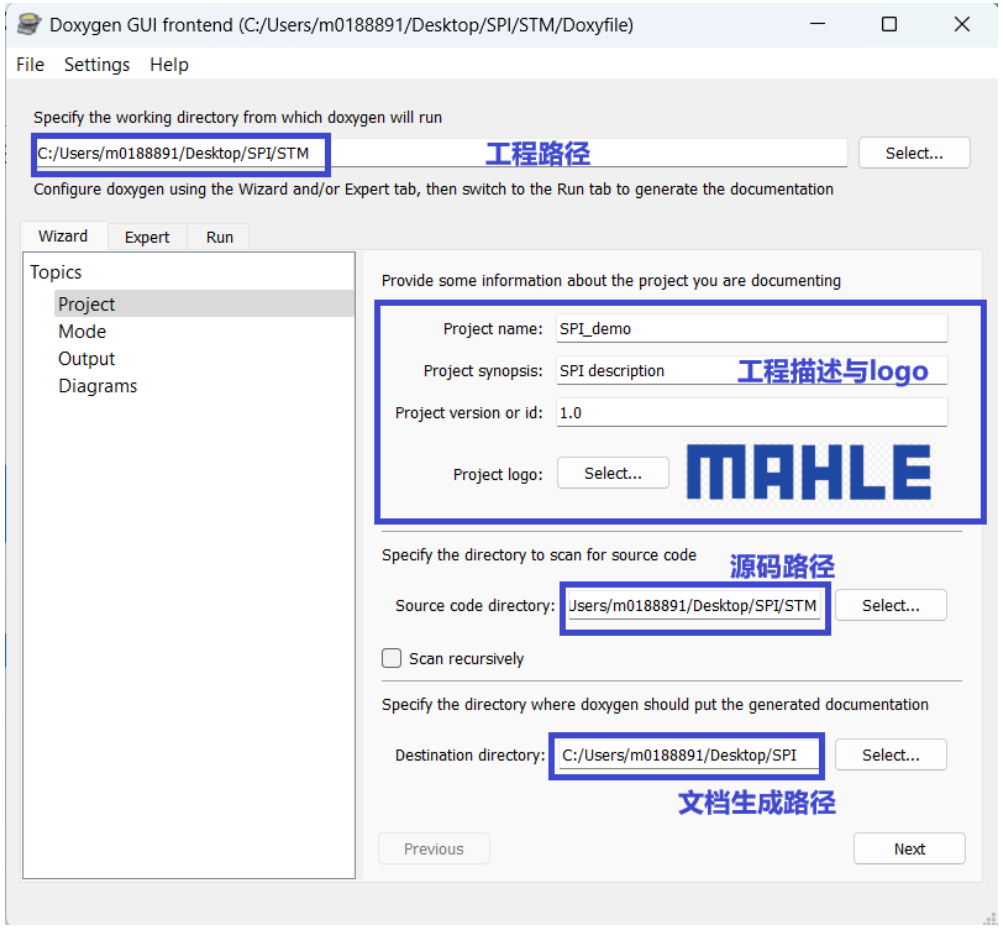
如需生成描述文档，需单独安装doxygen软件，安装方式推荐下载zip解压后，将文件夹路径添加到系统变量，
即可双击doxywizard.exe打开软件。下载链接(Doxygen download)

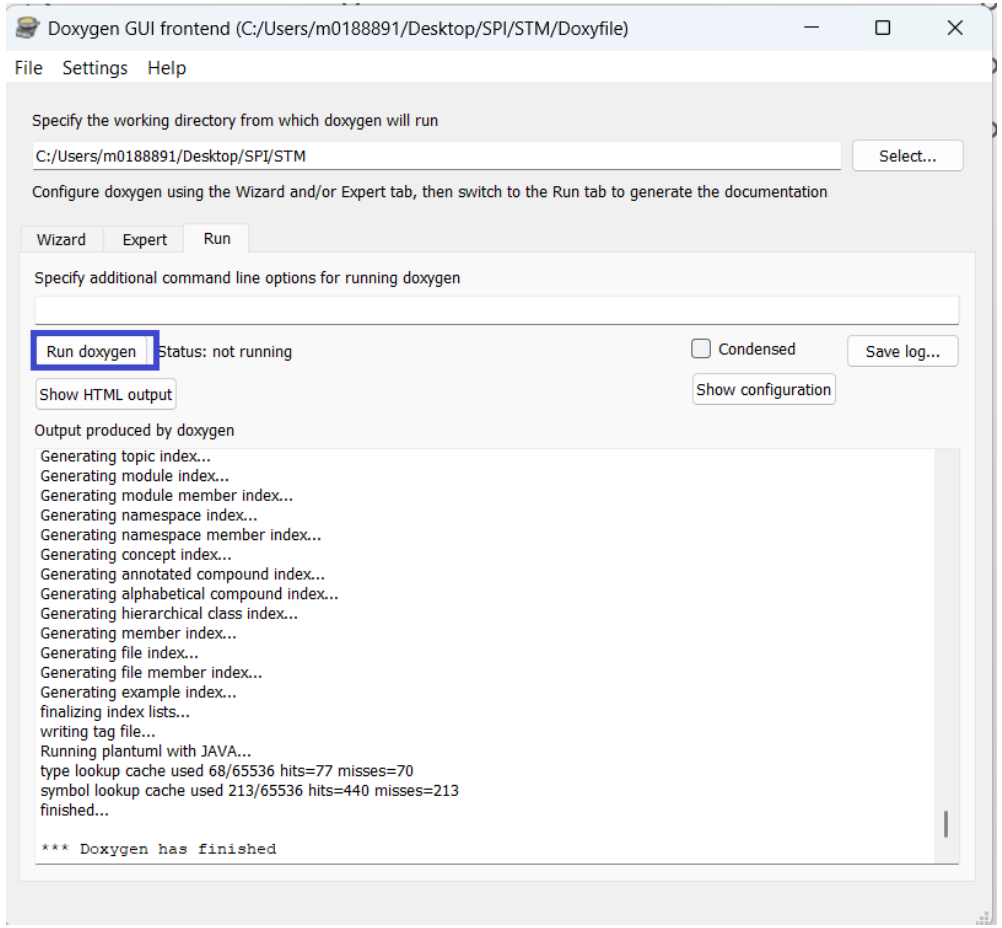


# Doxygen软件使用

简单的文档生成工程配置如图，点击run doxygen即可生成文档描述，点击html\index.html查看详细文件描述
如需更多设置可在expert中单独设置生成文档的效果。



# 文件注释

文件注释包含：@author作者，@version 版本，@data 时间，以上信息在模板中已定义。也可使用更多指令对描述划重点：@note; @warning; @attention…

```
/**
  ******************************************************************************
  * @file    stm32f10x_spi.c
  * @author  MCD Application Team
  * @version V3.5.0
  * @date    11-March-2011
  * @brief   This file provides all the SPI firmware functions.
  ******************************************************************************
  * @attention
  *
  * THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS
  * WITH CODING INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE
  * TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY
  * DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING
  * FROM THE CONTENT OF SUCH FIRMWARE AND/OR THE USE MADE BY CUSTOMERS OF THE
  * CODING INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.
  *
  * <h2><center>&copy; COPYRIGHT 2011 STMicroelectronics</center></h2>
  ******************************************************************************
  */
```

# 文件注释效果

## Detailed Description

This file provides all the SPI firmware functions.

**Author**

MCD Application Team

**Version**

V3.5.0

**Date**

11-March-2011

**Attention**

THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS WITH CODING INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM THE CONTENT OF SUCH FIRMWARE AND/OR THE USE MADE BY CUSTOMERS OF THE CODING INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.

# 函数注释

函数注释通常包含： 函数输入的参数@param，函数返回值@retval，函数的功能@brief；

```
/**
  * @brief  Clears the SPIx CRC Error (CRCERR) flag.
  * @param  SPIx: where x can be
  *   - 1, 2 or 3 in SPI mode
  * @param  SPI_I2S_FLAG: specifies the SPI flag to clear.
  *   This function clears only CRCERR flag.
  * @note
  *   - OVR (OverRun error) flag is cleared by software sequence: a read
  *     operation to SPI_DR register (SPI_I2S_ReceiveData()) followed by a read
  *     operation to SPI_SR register (SPI_I2S_GetFlagStatus()).
  *   - UDR (UnderRun error) flag is cleared by a read operation to
  *     SPI_SR register (SPI_I2S_GetFlagStatus()).
  *   - MODF (Mode Fault) flag is cleared by software sequence: a read/write
  *     operation to SPI_SR register (SPI_I2S_GetFlagStatus()) followed by a
  *     write operation to SPI_CR1 register (SPI_Cmd() to enable the SPI).
  * @retval None
  */
void SPI_I2S_ClearFlag(SPI_TypeDef* SPIx, uint16_t SPI_I2S_FLAG)
{
  /* Check the parameters */
  assert_param(IS_SPI_ALL_PERIPH(SPIx));
  assert_param(IS_SPI_I2S_CLEAR_FLAG(SPI_I2S_FLAG));

    /* Clear the selected SPI CRC Error (CRCERR) flag */
    SPIx->SR = (uint16_t)~SPI_I2S_FLAG;
}
```

# 函数注释文档生成效果

## ◆ SPI_I2S_ClearFlag()

```
void SPI_I2S_ClearFlag ( SPI_TypeDef * SPIx,
                         uint16_t      SPI_I2S_FLAG )
```

Clears the SPIx CRC Error (CRCERR) flag.

**Parameters**

**SPIx** where x can be
- 1, 2 or 3 in SPI mode

**SPI_I2S_FLAG** specifies the SPI flag to clear. This function clears only CRCERR flag.

**Note**
- OVR (OverRun error) flag is cleared by software sequence: a read operation to SPI_DR register (**SPI_I2S_ReceiveData()**) followed by a read operation to SPI_SR register (**SPI_I2S_GetFlagStatus()**).
- UDR (UnderRun error) flag is cleared by a read operation to SPI_SR register (**SPI_I2S_GetFlagStatus()**).
- MODF (Mode Fault) flag is cleared by software sequence: a read/write operation to SPI_SR register (**SPI_I2S_GetFlagStatus()**) followed by a write operation to SPI_CR1 register (**SPI_Cmd()** to enable the SPI).

**Return values**

**None**

常用注释命令可参考 Doxygen 注释语法规范 - schips - 博客园

# 结构体注释

结构体注释在vscode中没有生成指定的模板，需要在内容间自行添加。

```
/**
  * @brief  SPI Init structure definition
  */

typedef struct
{
  uint16_t SPI_Direction;           /*!< Specifies the SPI unidirectional or bidirectional data mode.…
  uint16_t SPI_Mode;                /*!< Specifies the SPI operating mode.…
  uint16_t SPI_DataSize;            /*!< Specifies the SPI data size.…
  uint16_t SPI_CPOL;                /*!< Specifies the serial clock steady state.…
  uint16_t SPI_CPHA;                /*!< Specifies the clock active edge for the bit capture.…
  uint16_t SPI_NSS;                 /*!< Specifies whether the NSS signal is managed by…
  uint16_t SPI_BaudRatePrescaler;   /*!< Specifies the Baud Rate prescaler value which will be…
  uint16_t SPI_FirstBit;            /*!< Specifies whether data transfers start from MSB or LSB bit.…
  uint16_t SPI_CRCPolynomial;       /*!< Specifies the polynomial used for the CRC calculation. */
}SPI_InitTypeDef;
```

# 结构体注释生成效果

## Detailed Description

SPI Init structure definition.

## Member Data Documentation

### SPI_BaudRatePrescaler

uint16_t SPI_InitTypeDef::SPI_BaudRatePrescaler

Specifies the Baud Rate prescaler value which will be used to configure the transmit and receive SCK clock. This parameter can be a value of SPI_BaudRate_Prescaler.

> **Note**
> The communication clock is derived from the master clock. The slave clock does not need to be set.

### SPI_CPHA

uint16_t SPI_InitTypeDef::SPI_CPHA

Specifies the clock active edge for the bit capture. This parameter can be a value of SPI_Clock_Phase

### SPI_CPOL

uint16_t SPI_InitTypeDef::SPI_CPOL

Specifies the serial clock steady state. This parameter can be a value of SPI_Clock_Polarity

### SPI_CRCPolynomial

uint16_t SPI_InitTypeDef::SPI_CRCPolynomial

# 参考文档

更多详细内容可参考已生成的html文件描述进一步了解其他注释功能
SPI\html\ index.html