[野火]sphinx 文档规范与模版

2019 年 11 月 22 日

为什么使用 sphinx 写文档

CHAPTER 1

关于本项目

本项目的 github 地址：https://github.com/Embdefire/ebf_contribute_guide

本项目的 gitee 地址：https://gitee.com/wildfireteam/ebf_contribute_guide

点击右侧链接可在线阅读本项目文档：《[野火]sphinx 文档规范与模版 》

本文档是关于如何使用 sphinx 编写文档的说明。也可以把本文档作为模版创建其它 sphinx 的文档。

1.1 为什么使用 sphinx 编写文档

使用 shpinx 编写文档有如下优点：

- 使用 sphinx 编写的文档可以方便地制作 html、pdf 等格式，非常方便浏览和转换。
- sphinx 支持 rst 和 markdown 语法，方便共享及开源编辑，使用 git 也方便跟踪。
- 由于 rst 语法比 markdown 语法更强大和方便，我们主要采用 rst 语法编写文档，linux 内核源码文档也是使用 rst 格式编写的。我们的文档也支持 markdown，主要是为了方便不熟悉 rst 的用户参与进来。

CHAPTER 2

关于野火

2.1 开源共享，共同进步

野火在发布第一块 STM32 开发板之初，就喊出 **开源共享，共同进步**的口号，把代码和文档教程都免费提供给用户下载，而我们也一直把这个理念贯穿至今。

目前我们的产品已经包括 *STM32*、*i.MX RT* 系列、*GD32V*、*FPGA*、*Linux*、*emXGUI*、操作系统、网络、下载器等分支，覆盖电子工程应用领域的各种常用技术，其中教学类产品的代码和文档一直保持着开源的姿态发布到网络上，为电子工程师排忧解难，让嵌入式没有难用的技术是我们最大的愿望。

2.2 联系方式

- 官网：http://www.embedfire.com
- 论坛：http://www.firebbs.cn
- github 主页：https://github.com/Embdefire
- gitee 主页：https://gitee.com/wildfireteam
- 淘宝：https://yehuosm.tmall.com
- 邮箱：embedfire@embedfire.com
- 电话：0769-33894118

CHAPTER 3

快速参与本项目（提交 bug 或文档修改）

开源代码和文档最重要的初衷是让大家参与进来，一起来找茬。

无论你是觉得代码写得不够漂亮、有 bug，或是文档小到有错别字，大到有原理性的描述错误，都可以直接通过项目的 github 提交给我们，我们会视具体的情况进行审核处理加入到主线或特性分支。

野火的每个项目的《关于本项目》文件中都包含有项目的 git 仓库地址，通过 git 提交 **issue**、**pull request** 的方式可参与到项目。

我们主要使用 github 仓库进行维护和更新，gitee 仓库用于方便用户快速下载，我们一般只在 gitee 上同步 github 的 master 分支，所以参与项目时请尽量使用 github。

3.1 轻度参与，提交 issue

如果你发现了一些 bug 或小问题，而自己暂时没时间帮助我们修改的话，可以直接在项目的 github 中提交 issue 告诉我们，我们会适时进行处理。

3.2 深度参与，提交 pull request

如果你想深度参与本项目，可以先克隆或 fork 项目的仓库到本地，修改编辑后向我们提交 pull request，我们会审核处理。

CHAPTER 4

TODO 和悬赏任务

本页记录待改进的事项，以后会在此页发布任务清单，完成部分任务可获得 **奖励**，敬请期待！

如果您想参与到项目，可按《*快速参与本项目（提交 bug 或文档修改）*》的说明操作。

4.1 文档任务

此处写具体的任务列表

- PDF 与 readthedocs
- 详细写如何贡献（git issue 和 commit）

4.2 代码开发任务

此处写具体的任务列表

CHAPTER 5

sphinx 的安装与使用

5.1 安装 sphinx

sphinx 官方安装说明：http://www.sphinx-doc.org/en/master/usage/installation.html

readthedoc 官方说明：https://docs.readthedocs.io/en/stable/intro/getting-started-with-sphinx.html

总的来说步骤如下：

- 安装 python3
- 通过 python3 安装 sphinx 包
- 根据项目要求安装项目的 requirements.txt 里的软件包

安装 python3

windows 直接到 python 官网下载安装

Ubuntu 下使用如下指令安装：

```
sudo apt install python3
```

5.2 安装 sphinx

windows 下使用如下指令安装：

```
py -3 -m pip install sphinx
# 国内用户推荐使用清华源安装，使用-i 指定源
py -3 -m pip install -i https://pypi.tuna.tsinghua.edu.cn/simple sphinx
```

Ubuntu 下使用如下指令安装

```
python3 -m pip install sphinx
# 国内用户推荐使用清华源安装，使用-i 指定源
py -3 -m pip install -i https://pypi.tuna.tsinghua.edu.cn/simple sphinx
```

5.3 创建 sphinx 文档

本文档已适配好支持 markdown、默认的 readthedoc 主题等内容。推荐直接使用本文档作为模版，修改 conf.py 文件的配置即可改变项目的名字、主题之类的内容。

使用时，要先根据项目的 **requirements.txt** 安装依赖的 python 包。

```
#Windows 指令，推荐使用 powershell 运行
py -3 -m pip install -r requirements.txt
#Ubuntu 指令
python3 -m pip install -r  requirements.txt
```

5.4 创建全新的 sphinx 文档

若不想使用本工程模版，可以使用如下指令创建全新的文档。

```
sphinx-quickstart
```

按照提示回答问题即可。推荐使用默认的 _build 目录，与 vscode 保持一致。其中提示语言时可以使用这个中文代码：zh_CN

sphinx 默认不支持 markdown 语法，要支持的话请参考本模版的 conf.py 文件配置。

5.5 编译

如果使用了 vscode 的 rst 插件，可以直接，保存 rst 文件后它会自动编译并可预览。

到文档源码所在的 makefile 目录：

```
# 在文档的 makefile 目录下执行
make html
```

在设定的 build 或 _build 的 html 目录下会生成静态的 html 文件。可直接使用这些静态的 html 文件制作网站。

5.6 使用 python 服务器预览

vscode 插件预览有时不够完整，可以在本地开启一个 python 服务器来预览。进入到生成的 _build/html 目录，运行如下指令：

```
# 在生成的 html 目录执行如下指令
#Windows
py -3 -m http.server  8000

#Ubuntu
python3 -m http.server 8000
```

运行指令后，在浏览器中打开 http://localhost:8000 即可查看生成的静态网页。

5.7 清除编译输出

有时 html 文件不会完全达到我们修改 rst 后的效果，这可能是因为之前的旧文件影响，这时可以先清除编译输出再重新编译。

```
# 清除编译输出
make clean
# 重新编译
make html
```

CHAPTER 6

使用 vscode 编写 rst 文件

使用 vscode 编写 sphinx 文档非常方便，可以即时预览和自动编译。

rst vscode 插件说明页：https://docs.restructuredtext.net/index.html

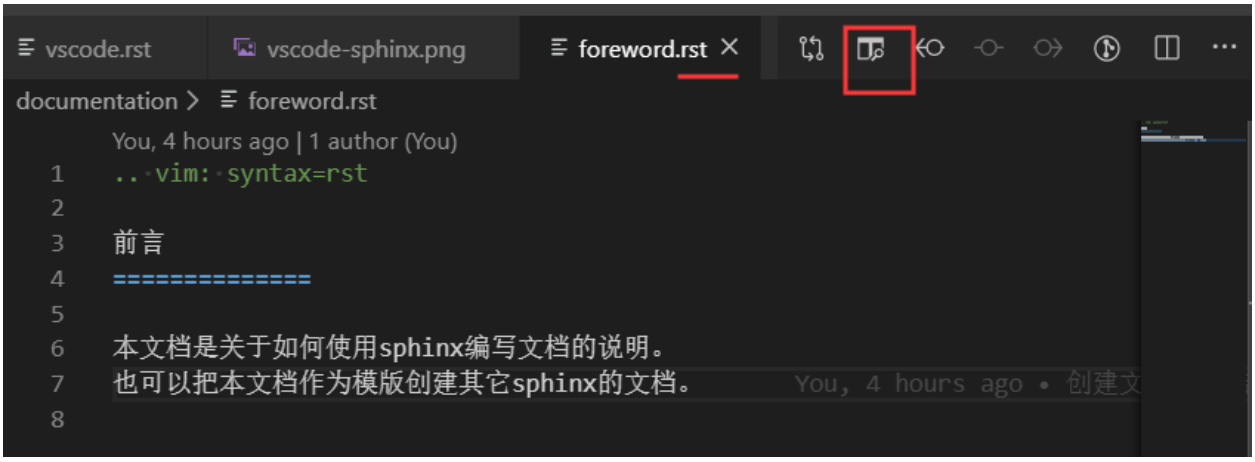
另外，vscode 默认支持 markdown 单个文件的预览。

需要安装的 vscode 插件：

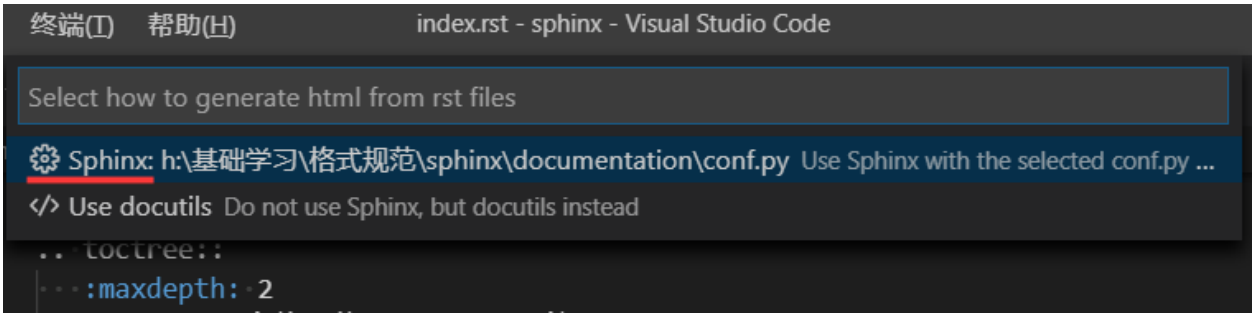
- python
- reStructuredText

安装 python 插件后，要选择使用 python3 的解释器，不要使用 python2。

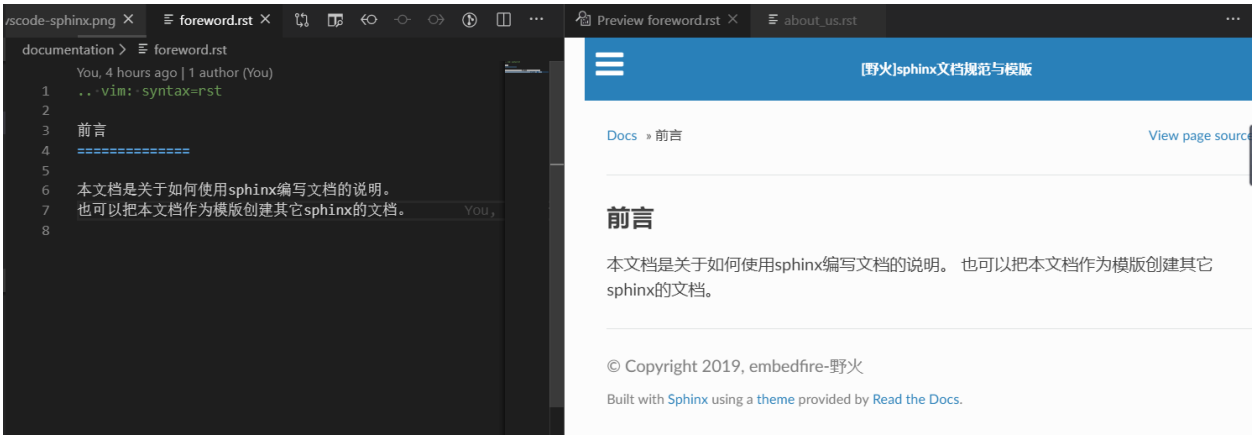
安装 reStructuredText 插件后，打开 rst 文件，点击预览按钮。



会提示使用 sphinx 还是 doctuil 工具，我们选择使用 sphinx。



若配置正确，稍等一会即可在右侧看到预览效果。



CHAPTER 7

基础语法

rst 的基础语法和 markdown 差不多，

可以使用这个在线的 rst 编辑器了解相关语法：rst 在线编辑器

- sphinx 语法官网：http://www.sphinx-doc.org/en/master/usage/restructuredtext/basics.html
- restruct 语法官网：http://docutils.sourceforge.net/rst.html

下面是常用语法：

语法：

```
一级标题
==============================

二级标题
------------------

三级标题
^^^^^^^^^^^^^^^^^^^^^^

四级标题
""""""""""""""""""""

五级标题
**********
```

(下页继续)

(续上页)

```
** 强调 **

* 斜体 *

``monospace，会变色，具体作用不清楚``

无序列表
--------------------------
- hhhhhhhh
- hhhhhhhh
- hhhhhhhh
- hhhhhhhh
- hhhhhhhh
- hhhhhhhh

有序列表
------------------------
支持数字、大小写字母和罗马数字

1. hhhhhhhh
#. hhhhhhhh
#. hhhhhhhh
#. hhhhhhhh
#. hhhhhhhh
#. hhhhhhhh

a. hhhhhhhh
#. hhhhhhhh
#. hhhhhhhh
#. hhhhhhhh
#. hhhhhhhh
#. hhhhhhhh
```

效果:

7.1 一级标题

7.1.1 二级标题

三级标题

四级标题

五级标题

强调

斜体

`monospace，会变色，具体作用不清楚`

7.1.2 无序列表

- hhhhhhhh
- hhhhhhhh
- hhhhhhhh
- hhhhhhhh
- hhhhhhhh
- hhhhhhhh

7.1.3 有序列表

支持数字、大小写字母和罗马数字

1. hhhhhhhh
2. hhhhhhhh
3. hhhhhhhh
4. hhhhhhhh
5. hhhhhhhh
6. hhhhhhhh

a. hhhhhhhh

b. hhhhhhhh

c. hhhhhhhh

d. hhhhhhhh

e. hhhhhhhh

f. hhhhhhhh

CHAPTER 8

超链接与文件引用

8.1 超链接

语法：

```
直接嵌入网址：
`野火公司官网 <http://www.embedfire.com>`_

使用引用的方式把具体网址定义在其它地方： `野火公司官网 `_

.. _ 野火公司官网: http://www.embedfire.com
```

效果：

直接嵌入网址：

野火公司官网

使用引用的方式把具体网址定义在其它地方：野火公司官网

8.2 引用

8.2.1 引用文档

语法：

```
自定义引用文字

:doc:`引用本地的其它 rst 文档,rst 后缀要省略，如 about_us <../about_us>`

使用标题文字
:doc:`../about_us`
```

效果：

自定义引用文字

引用本地的其它 *rst* 文档,*rst* 后缀要省略，如 *about_us*

使用标题文字关于野火

8.2.2 使用标签引用文档

要在被引用的文件头定义标签，如 about_us.rst 文件头写 “about_embedfire” 的标签，具体请查看该文档的源码

语法：

```
:ref:`about_embedfire <about_embedfire>`

:ref:`about_embedfire`
```

效果：

about_embedfire

关于野火

8.2.3 引用非 rst 文档

会呈现出点击后下载文件的效果。

语法：

```
:download:`引用非 rst 的本地文档 <../requirements.txt>`.
```

效果：

引用非 rst 的本地文档.

CHAPTER 9

代码高亮

参考说明：http://www.sphinx-doc.org/en/master/usage/restructuredtext/directives.html#toctree-directive

支持的高亮语言：https://pygments.org/docs/lexers#lexers-for-various-shells

9.1 快速定义代码块

使用简便的预定义高亮语法高亮缩进，默认不带语言说明的都使用 highlight 定义的语言高亮，然后可以直接使用“::”两个冒号代替“code-block”指令快速定义其它代码块，直到下一个 highlight 指令，才会改变语言：

```
.. highlight:: sh

此指令后如下的“::”定义的块都会以 sh 语法进行高亮，直到遇到下一条 highlight 指令。

::

   # 此命令在主机执行
   sudo apt install python
   echo "helloworld,this is a script test!"
```

效果：

```
# 此命令在主机执行
sudo apt install python
echo "helloworld,this is a script test!"
```

9.2 code-block 代码高亮

9.2.1 shell 高亮测试

高亮语法:

```
.. code-block:: sh
   :caption: test
   :name: test333
   :emphasize-lines: 2
   :linenos:

   # 此命令在主机执行
   sudo apt install python
   echo "helloworld,this is a script test!"
```

效果:

列表 1: sh test

```
1 # 此命令在主机执行
2 sudo apt install python
3 echo "helloworld,this is a script test!"
```

9.2.2 C 高亮测试

语法:

```
.. code-block:: c
   :caption: c test
   :emphasize-lines: 4,5
   :linenos:

   #include <stdio.h>
```

(下页继续)

(续上页)

```
   int main()
   {
      printf("hello, world! This is a C program.\n");
      for(int i=0;i<10;i++ ){
         printf("output i=%d\n",i);
      }

      return 0;
   }
```

效果：

列表 2: c test

```c
#include <stdio.h>

int main()
{
   printf("hello, world! This is a C program.\n");
   for(int i=0;i<10;i++ ){
      printf("output i=%d\n",i);
   }

   return 0;
}
```

9.3 literalinclude 直接嵌入本地文件并高亮

9.3.1 嵌入整个文件

直接嵌入文件，包含标题、代码语言、高亮、带编号以及名称方便引用。

插入 C 代码

```
.. literalinclude:: ../../base_code/hello.c
   :caption: ../../base_code/hello.c
   :language: c
   :emphasize-lines: 5,7-12
```

(下页继续)

(续上页)

```
   :linenos:
   :name: hello.c
```

效果:

列表 3: ../../base_code/hello.c

```c
/* $begin hello */
#include <stdio.h>

int main()
{
        printf("hello, world! This is a C program.\n");
        for(int i=0;i<10;i++ ){
                printf("output i=%d\n",i);
        }

        return 0;
}
/* $end hello */

```

插入 shell 代码

语法:

```
.. literalinclude:: ../../base_code/hello_world.sh
   :caption: ../../base_code/hello_world.sh
   :language: sh
   :linenos:
```

效果:

列表 4: ../../base_code/hello_world.sh

```sh
echo "helloworld,this is a script test!"
```

插入 Makefile 代码

语法:

```
.. literalinclude:: ../../base_code/Makefile
   :caption: ../../base_code/Makefile
   :language: makefile
   :linenos:
```

效果：

列表 5: ../../base_code/Makefile

```
1  # 生成的可执行文件名
2
3  hello:hello.o
4          gcc -o hello hello.o
5
6  # 生成规则
7  hello.o:hello.c
8          gcc -c hello.c -o hello.o
9
10 # 伪目标
11 .PHONY:clean
12 clean:
13         rm -f *.o hello
```

9.3.2 嵌入文件的某部分

通过 lines 指定嵌入文件的某些行。

语法：

```
.. literalinclude:: ../../base_code/hello.c
   :caption: ../../base_code/hello.c
   :language: c
   :linenos:
   :lines: 1,3,5-8
```

效果：

列表 6: ../../base_code/hello.c

```
1  /* $begin hello */
2
3  {
```

(下页继续)

(续上页)

```
4         printf("hello, world! This is a C program.\n");
5         for(int i=0;i<10;i++ ){
6                 printf("output i=%d\n",i);
```

9.3.3 文件对比

语法:

```
.. literalinclude:: ../../base_code/hello.c
:diff: ../../base_code/hello_diff.c
```

效果:

```
--- /home/docs/checkouts/readthedocs.org/user_builds/ebf-contribute-guide/checkouts/
↪stable/base_code/hello_diff.c
+++ /home/docs/checkouts/readthedocs.org/user_builds/ebf-contribute-guide/checkouts/
↪stable/base_code/hello.c
@@ -5,7 +5,7 @@
 {
         printf("hello, world! This is a C program.\n");
         for(int i=0;i<10;i++ ){
-                printf("diff output i=%d\n",i);
+                printf("output i=%d\n",i);
         }

         return 0;
```

CHAPTER 10

图片与视频

10.1 图片

图片原文件统一存储在引用文档所在的同级目录的 **media** 文件夹下。

显示图片直接使用 image 指令，无特殊情况的话我们书籍图片要求使用居中方式显示：

语法：

```
.. image:: ../media/rest-syntax/pic-video/logo.png
:align: center
```

效果：

以下的图片显示方式是 word 自动转换的结果，使用这种方式无法以居中形式显示图片，所以我们要修改。

它的原理是使用“||”类似宏的方式替换指令，使 rst 源码看起来更简洁，但不能居中显示。

语法：

```
|logo|，使用"| 宏名 |" 的形式替换文字。

.. |logo| image:: ../media/rest-syntax/pic-video/logo.png
```

效果：

，使用”|”宏名”|”的形式替换文字。

图片还可以使用 width、heigh、scale 等参数，但不推荐使用，设置过 width、height 参数的图片，在页面可以点击查看原图。

语法：

```
.. image:: ../media/rest-syntax/pic-video/logo.png
   :align: center
   :width: 5.63529in
   :height: 0.97222in
```

效果：

10.2 视频

使用 raw 指令插入 html 代码。直接粘贴视频网站的通用 html 代码，不过貌似不能放大。

在 vscode 可能无法预览，直接在浏览器中看即可。

注解： TODO：目前还不知道如何放大，以及点击后到具体的视频网站播放。

10.2.1 哔哩哔哩

语法：

```
.. raw:: html

<iframe src="//player.bilibili.com/player.html?aid=70961112&cid=122951107&page=1"␣
↪scrolling="no" border="0" frameborder="no" framespacing="0" allowfullscreen="true"> </
↪iframe>
```

效果:

10.2.2 优酷

语法:

```
.. raw:: html

<iframe height=498 width=510 src='http://player.youku.com/embed/XMzk2MzQxNTQ3Ng==='␣
↪frameborder=0 'allowfullscreen'></iframe>
```

效果:

CHAPTER 11

特殊提示

特殊提示支持警告、重要、提示、注意等标签，适合做显眼的用途。

- attention
- caution
- danger
- error
- hint
- important
- note
- tip
- warning

语法：

```
.. note:: This is a note admonition.
 This is the second line of the first paragraph.

 - The note contains all indented body elements
   following.
 - It includes this bullet list.
```

(下页继续)

(续上页)

```
.. hint:: This is a hint admonition.

.. important:: This is a important admonition.

.. tip:: This is a tip admonition.

.. warning:: This is a warning admonition.

.. caution:: This is a caution admonition.

.. attention:: This is a attention admonition.

.. error:: This is a error admonition.

.. danger:: This is a danger admonition.
```

效果:

注解: This is a note admonition. This is the second line of the first paragraph.

- The note contains all indented body elements following.
- It includes this bullet list.

提示: This is a hint admonition.

重要: This is a important admonition.

小技巧: This is a tip admonition.

警告: This is a warning admonition.

警告: This is a caution admonition.

注意： This is a attention admonition.

错误： This is a error admonition.

危险： This is a danger admonition.

CHAPTER 12

rst 的指令和角色（directive and role）

- directive 说明：http://docutils.sourceforge.net/docs/ref/rst/directives.html
- role 说明：http://docutils.sourceforge.net/docs/ref/rst/roles.html

rst 使用 directive 和 role 进行特殊操作。

12.1 directive

directive 的格式为：

```
.. directive 名:: 参数
   : 参数:
   : 参数:
```

12.2 role

role 的格式为：

```
:role 名: 内容
```

CHAPTER 13

野火 sphinx 文档规范

开源项目的整体目录：

README	说明文档或软链接至 documentation 中的说明，方便 github 阅读
base_code	配套代码
documentation	配套文档
faq	存储常见问题的文档
about_us.rst	关于我们
_build	文档编译输出目录
conf.py	sphinx 配置文件
contribute	如何参与项目，贡献与投稿的说明
foreword.rst	前言
index.rst	目录
make.bat	
Makefile	
media	文档中使用的图片文件
requirements.txt	文档的 python 依赖
_static	
_templates	
TODO.rst	待完成的内容，发布的任务列表

13.1 图片

文档引用的图片存储在文档源码目录下的 media 文件夹中，按部分、章节及文档分目录存储。

图片要直接使用如下形式, 方便居中：

```
.. image:: media/logo.png
   :align: center
```

不要使用如下形式，如下形式是 docx 转换后的格式，它不支持居中，见到要把它改好。

```
|logo|

.. |logo| image:: media/logo.png
```

13.2 rst 格式检查

使用 vscode 的 rst 插件在编写时就会有格式检查。编写文档时应尽量满足该格式检查的规范。对于 docx 转 rst 后的不规范文档，做到见一个改一个。

make html 时，编译会有提示输出，尽量让它不输出的 warning。

13.3 代码引用

超过 3 行的代码要加上行号、并用 caption 名指明代码片段的名，对于引用的代码文件，使用 caption 指明引用的路径名。

如以下语法：

```
.. literalinclude:: ../../base_code/hello.c
   :caption: ../../base_code/hello.c
   :language: c
   :linenos:
```

CHAPTER 14

使 sphinx 支持 markdown

sphinx 默认是不支持 markdown 语法的，但可以通过插件来支持。

14.1 markdown 基础支持

recommonmark 的 PyPi 说明：https://pypi.org/project/sphinx-markdown-tables/

recommonmark 的 sphinx 说明：http://www.sphinx-doc.org/en/master/usage/markdown.html

recommonmark 的使用说明：https://recommonmark.readthedocs.io/en/latest/index.html

```
pip install recommonmark
```

在 conf.py 添加扩支持：

```
extensions = ['recommonmark']
```

配置后就可以在 sphinx 中使用 markdown 文件了

14.2 markdown 表格支持

要支持 markdown 的表格语法，还需要安装 sphinx-markdown-tables

markdown 表格支持：https://pypi.org/project/sphinx-markdown-tables/

安装：

```
pip install sphinx-markdown-tables
```

使用:

在 conf.py 添加扩展支持:

```
extensions = [
   'sphinx_markdown_tables',
]
```

CHAPTER 15

使用 pandoc 进行文档转换

使用 pandoc 可以方便地对文档的格式进行相互转换。

15.1 安装

pandoc 安装说明：https://github.com/jgm/pandoc/blob/master/INSTALL.md

pandoc 用户手册：https://pandoc.org/MANUAL.html

根据系统按它的说明下载对应的安装包即可，ubuntu 下不要直接用 apt 安装，apt 安装的版本太旧。

在 ubuntu 下可以使用如下命令查看 pandoc 的说明：

```
man pandoc
```

15.2 docx 转 rst

使用 python 安装 rstdoc 扩展包，方便转换：https://pypi.org/project/rstdoc/

docx 转 rst 操作步骤：

- 先把 docx 文档按章节分开，一个章节一个 docx 文档，不然整个文档转换可能太大，导致错误，而且整个文档转换图片或内容不方便处理。
- 使用 rstfromdocx 命令一个个文档转换成 rst

使用如下指令转换

```
#doc1 为要转换的文档
rstfromdocx -lurg doc1.docx
```

命令执行后会生成与文档名相同的目录，目录下是 sphinx 形式的 rest 文档，把 rest 文档复制至新的书籍目录并把后缀改为 rst，media 下的图片也放到目标书籍的同级目录即可。

小技巧: 在 Windows 转换后，生成的文件目录引用使用的路径是 “”，而 sphinx 路径只支持 “/”，所以图片之类的引用路径要注意修改。

15.3 pandoc rst 转 docx

rstdocx 插件据说可以更完美地把 rst 转成 docx，但还不清楚 rstdocx 插件如何把 rst 转成 docx，但用 pandoc 可满足基本要求。

```
pandoc rstfile.rst  -f   rst  -t docx  -o targetdocfile.docx  --data-dir=sourcedir  --
↪reference-doc=module.docx
```

- 其中 rstfile.rst 是源文件，-f 指源格式，-f 指目标格式
- -o 指定的是输出文件
- –data-dir= 指定的是依赖目录
- –reference-doc= 是 docx 模版文件，使用模版文件转换可以把标题之类的格式搞得更规范

reference-doc 模版及说明：https://pandoc.org/MANUAL.html#option–reference-doc

15.4 pandoc docx 转 rst

与 rst 转 docx 类似，输出的图片目录可以使用这个参数设置：

–extract-media=DIR

CHAPTER 16

常见问题

CHAPTER 17

版权说明

野火电子保留本项目的所有版权。

公司组织有生存的压力，因为开源而导致生存不下去，是有 GEEK 精神的人不愿看到的事情。由于我们还不熟悉各种版权条例，所以关于版权的问题还在选择中。

目前我们保留本项目的所有版权，但我们秉承开源的心是不变的。如果你使用本项目不是用于商业目的，基本不需要考虑版权问题。

我们主要担忧的商业目的如下，列出来的意思是如果用于以下目的，我们极有可能会追究版权。

- 同业竞争，目前开发板是我们主要的盈利来源，禁止把本项目用于其它开发板项目。
- 文档出版，项目中包含的文档我们随时会提交至出版社出版，所以我们保留文档出版的权利。如果你只是在自己的文章中使用了本项目文档中的内容，需要注明来源。