

RT-Thread package 开发指南

1. 软件包的开发

1.1 简介

软件包的定义：运行于 RT-Thread 物联网操作系统平台上，面向不同应用领域的软件组件形成一个软件包，由软件包描述信息，软件包源代码或库文件组成。

在制作软件包前，需要先对软件包的功能进行准确的定义，确保 **不要耦合** 与产品业务逻辑有关的代码，提高软件包的 **通用性**。

软件包应该包含以下内容：

- 1、软件包代码和说明文档。
- 2、软件包索引。

例如作为示例的 hello 软件包，这两部分的地址为分别是：

- [软件包代码和说明文档](#)
- [软件包索引](#)

1.2 整理软件包代码和说明文档

制作软件包前需要仔细阅读[示例软件包说明文档](#)，参考示例软件包的文件夹结构。

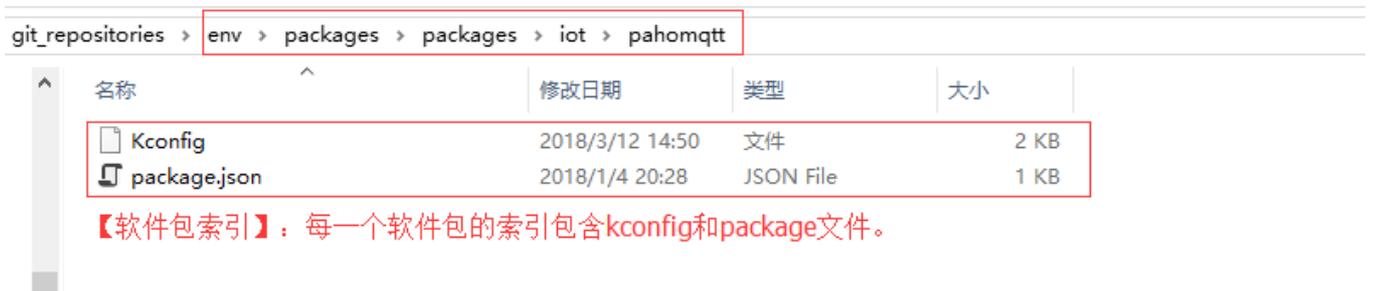
软件包代码和说明文档应该由以下部分组成：

- 软件包的功能代码；
- 根目录下的 SConscript 文件，用于和 RT-Thread 环境一起进行编译；
- 根目录下的 examples 文件夹，用于提供一份使用的例子；
- 如果需要加入 git submodule，则可以直接在根目录下添加；
- 根目录下的 README.md 文件，说明这个软件包的功能；
- 根目录下的 docs 文件夹，放置除了 README 之外的其他文档；

- 如果需要额外的移植代码，可将其放在 port 文件夹下；

1.3 创建软件包索引

软件包索引：指存放在 `env/packages` 文件夹下的软件包 **描述文件**。以 `env/packages/packages/iot/pahomqtt` 文件夹为例，包含内容如图所示：



名称	修改日期	类型	大小
Kconfig	2018/3/12 14:50	文件	2 KB
package.json	2018/1/4 20:28	JSON File	1 KB

【软件包索引】：每一个软件包的索引包含kconfig和package文件。

- **Kconfig 文件**：软件包的配置项，如软件包版本、功能选项等信息。
- **package.json 文件**：存放软件包的名称、简介、各个版本对应的下载链接等信息。

1.3.1 使用索引生成向导

我们可以使用 env 的软件包索引生成向导功能来制作软件包索引，命令为 `pkgs --wizard`，大致流程如下如图所示：

```
Administrator@SUMMERGIFT F:\git_repositories\env\sample
> pkgs --wizard  【制作软件包索引命令】
Welcome to package wizard, please enter the package information.
The messages in [] is default setting. You can just press enter to use default Settings.
Please enter the name of package:
pahomqtt  【1、【输入软件包名】】
version of package, default:
[1.0.0]  【2、【输入组件版本号，可以按下回车使用默认值，后续再更改】】
Please choose a class for your packages.
Enter 1 for iot and 2 for language, following the table below as a guideline.
[1:iot][2:language][3:misc][4:multimedia][5:security][6:system]
1  【3、【输入软件包类别，这次制作的是 iot 类的软件包，输入数字 1，按下回车即可】】
=====> Your package index was made successfully.

Administrator@SUMMERGIFT F:\git_repositories\env\sample
>  【提示软件包索引生成成功】

4、【在生成的文件夹中对软件包索引的信息进行修改】
```

生成内容如下图所示：

git_repositories > env > sample

名称	修改日期	类型	大小
pahomqtt	2018/3/14 15:58	文件夹	
Kconfig	2018/3/14 16:15	文件	1 KB

【本次生成的pahomqtt软件包索引】

git_repositories > env > sample > pahomqtt

名称	修改日期	类型	大小
Kconfig	2018/3/14 15:58	文件	1 KB
package.json	2018/3/14 15:58	JSON File	1 KB
SConscript	2018/3/14 15:58	文件	1 KB

【软件包索引文件夹所需的文件为kconfig文件和packages.json文件。】

SConscript 文件在制作软件包时使用，将其移动到软件包源码文件夹中即可，无需保留

在索引文件夹中。

1.3.2 修改 package.json 文件

- **package.json 文件介绍：**

```
1.  {
2.     "name" : "pahomqtt",
3.     "description" : "a pahomqtt package for rt-thread",    # 软件包描述信
   息
4.     "keywords" : [
5.         "pahomqtt"
6.     ],
7.     "site" : [
8.         {
9.             "version" : "v1.0.0",
10.            "URL" : "https://pahomqtt-1.0.0.zip",          # 根据版本号修改
   软件包压缩包的下载地址
11.            "filename" : "pahomqtt-1.0.0.zip",
12.            "VER_SHA" : "fill in the git version SHA value" # 压缩包形式无
   需填写
13.        },
14.        {
15.            "version" : "latest",                          # latest 版本
16.            "URL" : "https://xxxxx.git",                  # 可以填入 Git
   仓库地址
17.            "filename" : "Null for git package",
18.            "VER_SHA" : "fill in latest version branch name,such as mater" #
   填入 SHA 值或者分支名
19.        }
20.    ]
21. }
```

关于文件中 `URL` 值，每个版本可以填入两种类型：

- **Git**：可以填入 Git 仓库地址和对应版本的 SHA 值。一般 `latest` 版本会在 **SHA** 填入 `master`。
- **压缩包**：指定软件包压缩包的下载地址，此时无需填入 **SHA**。

package.json 是软件包的描述信息文件，包括软件包名称，软件包描述，作者等信息，以及必须的 package 代码下载链接。另外，请务必包含许可证的说明，使用了哪种许可证（GPLv2，LGPLv2.1，MIT，Apache license v2.0，BSD 等）。

修改后的 package.json 大致如下：

```
1.  {
2.     "name" : "pahomqtt",
3.     "description" : "Eclipse Paho MQTT C/C++ client for Embedded
platforms", # 更新了描述信息
4.     "keywords" : [
5.         "pahomqtt"
6.     ],
7.     "site" : [
8.         {
9.             "version" : "v1.0.0",
# v1.0.0 版本
10.            "URL" : "https://github.com/RT-Thread-packages/paho-mqtt.git",
# 更新了 git 仓库地址
11.            "filename" : "paho-mqtt-1.0.0.zip",
12.            "VER_SHA" : "cff7e82e3a7b33e100106d34d1d6c82e7862e6ab"
# 填入了指定版本的 SHA 值
13.        },
14.        {
15.            "version" : "latest",
# 最新版本
16.            "URL" : "https://github.com/RT-Thread-packages/paho-mqtt.git",
17.            "filename" : "paho-mqtt.zip",
18.            "VER_SHA" : "master"
# 填入 master
19.        }
20.    ]
21. }
```

1.3.3 修改 Kconfig 文件

Kconfig 文件内容大致如下：

```
Kconfig
1
2 # Kconfig file for package pahomqtt
3 config PKG_USING_PAHOMQTT ← 【使用软件包的宏定义，和Sconscript文件中的依赖相同】
4     bool "a pahomqtt package for rt-thread"
5     default n
6
7 if PKG_USING_PAHOMQTT
8
9     config PKG_PAHOMQTT_PATH
10        string
11        default "/packages/iot/pahomqtt" 【软件包索引在packages文件夹的存放位置】
12
13    choice
14        prompt "pahomqtt version"
15        help
16            Select the pahomqtt version
17
18        config PKG_USING_PAHOMQTT_V100
19            bool "v1.0.0" ← 【软件包可以选择的版本号，对应不同的下载地址】
20
21        config PKG_USING_PAHOMQTT_LATEST_VERSION
22            bool "latest"
23    endchoice
24
25    if PKG_USING_PAHOMQTT_V100
26        config PKG_PAHOMQTT_VER
27            string
28            default "v1.0.0"
29    endif
30
31    if PKG_USING_PAHOMQTT_LATEST_VERSION
32        config PKG_PAHOMQTT_VER
33            string
34            default "latest"
35    endif
36
37 endif
```

软件包索引中的 Kconfig 文件主要由 menuconfig 命令使用，软件包的一些选项必须在 Kconfig 文件中定义出来，注意事项如下：

- 1、索引向导自动生成的 Kconfig 文件中的内容大多是必须的，可以参考其他软件包修改选项的值，但是**不要删除选项**。
- 2、软件包必须包含一个以 `PKG_USING_` 开头的配置项，这样RT-Thread的包管理器才能将其正确识别。假设这个包的名称叫做SOFTA，那么软件包总选项应该是 `PKG_USING_SOFTA` ；
- 3、和这个SOFTA软件包相关的其他选项，需要以 `SOFTA_` 开头的配置项进行定义，可以是 `SOFTA_USING_A` 或者 `SOFTA_ENABLE_A` 等方式。

- 4、支持 **latest** 版本的软件包也至少需要一个固定版本，以防止在某个时候找不到合适的版本。
- 5、软件包如果还需要更多的配置项，可以搜索 **Kconfig 语法**，并参考已有的软件包来对 Kconfig 文件进行修改。

1.4 上传软件包

软件包可以上传到 git 上或者其他可供下载的地方，推荐使用 git 仓库的方式进行保存，这样方便更新软件包版本。

参考：[RT-Thread 软件包仓库](#)

1.5 测试软件包

- **软件包的下载**：将软件包索引拷贝到 `env\packages\packages` 下对应的位置，然后在 env 中尝试在线下载软件包，测试下载是否成功。
- **软件包的功能**：下载完成后，使用 `scons` 命令重新编译项目，在相应的环境下运行，测试软件包功能是否正确。
- **软件包版本的切换**：尝试在 `menuconfig` 下切换软件包的版本，查看版本切换是否正常。

1.6 向 RT-Thread 提交软件包索引

最后需要将软件包索引通过 PR 流程推送到：<https://github.com/RT-Thread/packages>

[点击这里](#) 了解如何提交 PR。

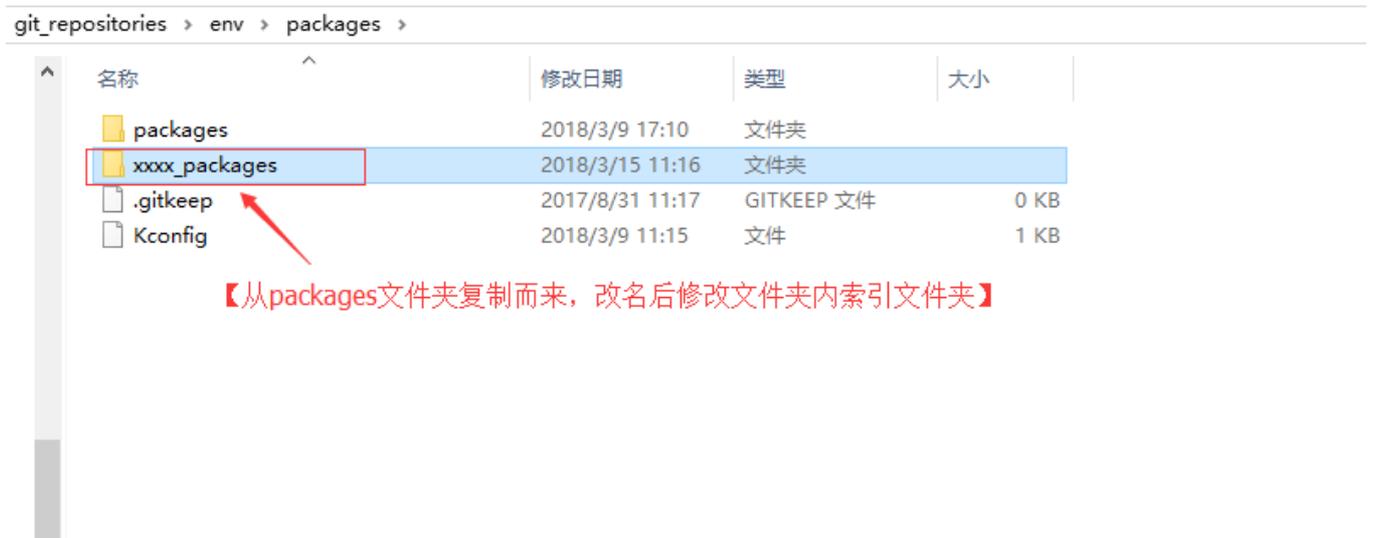
2. 软件包索引源的管理

2.1 简介

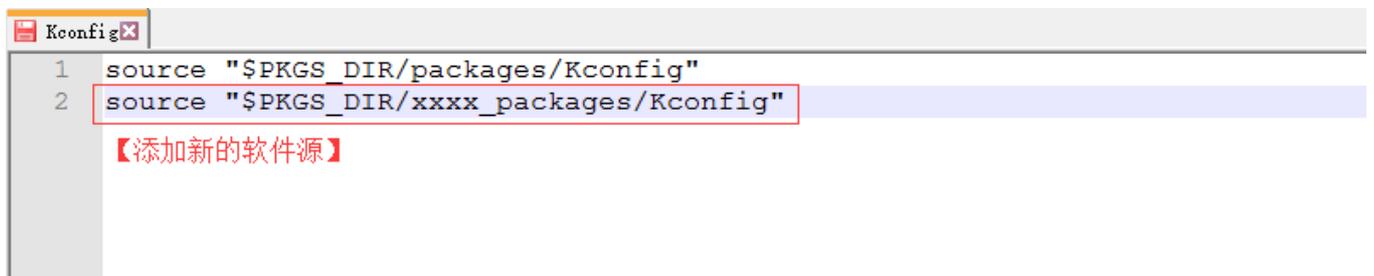
env 可以从多个软件包源来下载软件包，每个源的软件包列表就存放在 `env\packages` 文件夹中，如 `env\packages\packages` 文件夹下就是 RT-Thread 官方的软件包列表。

2.2 添加额外的软件包源

- 复制一份 RT-Thread 官方的 packages 文件夹，修改文件夹名称后，删除该文件夹内不需要的软件包索引，将需要的索引添加进去。



- 更新 `env\packages` 文件夹下的 Kconfig 文件，在 Kconfig 文件中添加软件包源信息。



2.3 删除软件包源

- 删除软件源文件夹；
- 将 Kconfig 文件中相应的源文件夹信息删除。