**yaml配置**

外部库下载链接：

yaml：[GitHub - jbeder/yaml-cpp: A YAML parser and emitter in C++](https://github.com/jbeder/yaml-cpp)[1]

CMAKE: https://cmake.org/download/

1.安装CMAKE，参考 [Windows下CMake安装教程\_u011231598的博客-CSDN博客\_cmake安装](https://blog.csdn.net/u011231598/article/details/80338941?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522164355404616780274160591%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=164355404616780274160591&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-80338941.first_rank_v2_pc_rank_v29&utm_term=WINDOWS+%E5%AE%89%E8%A3%85CMAKE&spm=1018.2226.3001.4187)

图形用户界面, 文本, 应用程序, 电子邮件, 网站

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

选中将CMAKE添加到系统路径

因为在安装过程中选择了自动添加环境，所以安装完后不用手动添加环境了，但是此时必须得重启电脑，变量才能使用，否则无法对Cmake安装进行测试重启电脑后，打开cmd黑窗，输入cmake，能够显示cmake的一些信息即为安装成功,如下图

文本

描述已自动生成

2.登录yaml网址[1]，下载源码并解压

电脑屏幕截图

描述已自动生成

3.开始Build yaml：在源码所在的路径下创建build文件夹，并打开CMAKE，分别选择SORCE CODE路径和BUILD路径，如下图

图形用户界面, 应用程序

描述已自动生成

然后点击下面的Configure按钮等待完成，选择对应的VS编译器版本，笔者为VS2019.出现Configuring done就成功了。

图形用户界面, 文本

描述已自动生成

接下来选中**YAML\_CPP\_BUILD\_SHARED\_LIB（重点）**，在yaml-cpp-master/目录下新建lib文件夹并将CMAKE\_INSTALL\_PREFIX项改成新建的lib目录（不必要，仅方便管理）点击Generate，出现Generating done。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

可以看到build文件夹中出现对应的make结果。

电子设备的屏幕

描述已自动生成

接下来点击Open Project，打开VS,随后分别用Debug模式和Release模式进行项目生成。

图形用户界面, 文本, 应用程序

描述已自动生成

可以看到生成后的文件夹。

文本

描述已自动生成

可以看到Release文件夹下的yaml-cpp.lib标志着Build完成。

（以下内容可跳过：yaml-cpp.lib 包含函数代码本身，在编译时直接将代码加入程序当中，称为静态链接库static link library;

而\*.dll包含了函数所在的DLL文件和文件中函数位置的信息（入口），代码由运行时加载在进程空间中的DLL提供，称为动态链接库dynamic link library。

两种文件的区别：

使用lib需注意两个文件：  
•.h头文件，包含lib中说明输出的类或符号原型或数据结构。应用程序调用lib时，需要将该文件包含入应用程序的源文件中。  
•.LIB文件。

使用dll需注意三个文件：

•.h头文件，包含dll中说明输出的类或符号原型或数据结构的.h文件。应用程序调用dll时，需要将该文件包含入应用程序的源文件中。  
•.LIB文件，是dll在编译、链接成功之后生成的文件，作用是当其他应用程序调用dll时，需要将该文件引入应用程序，否则产生错误（如果不想用lib文件或者没有lib文件，可以用WIN32 API函数LoadLibrary、GetProcAddress装载）。  
•dll文件，真正的可执行文件，开发成功后的应用程序在发布时，只需要有.exe文件和.dll文件，并不需要.lib文件和.h头文件

）

截图里有图片

描述已自动生成

4.引用yaml-cpp库：把刚才下载的yaml-cpp的include目录拷贝到VS项目下

电脑萤幕画面

描述已自动生成

**进行工程配置：以下展示DEBUG模式下的配置**

将生成的lib文件拷贝到新建的lib文件夹中，

图片包含 图形用户界面

描述已自动生成

将dll文件拷贝到VS工程目录（iPPP\_RTK\_cpp.vcxproj所在目录）

电脑软件截图

描述已自动生成

接下来配置VS的项目属性，附加库目录选择lib文件夹。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

附加库依赖项输入yaml-cppd.lib;

图形用户界面, 文本

描述已自动生成

C++常规->附加包含目录选择Include文件夹所在路径

图形用户界面, 文本, 应用程序

描述已自动生成

**进行工程配置：以下展示release模式下的配置,与debug类似，不同在于dll与lib文件名不同，仅展示截图。**

电脑萤幕的截图

描述已自动生成

电脑萤幕的截图

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 电子邮件

描述已自动生成

5.完成yaml的配置，头文件加入yaml.h

图形用户界面, 应用程序, Word

描述已自动生成

6.运行成功，完成配置

图形用户界面, 文本, 应用程序

描述已自动生成