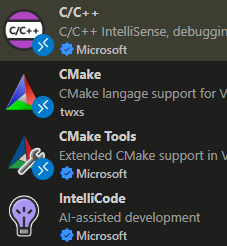
1. 在Ubuntu中安装编译器与cmake：

sudo apt install build-essential cmake ninja-build

1. 安装vscode打开，

安装插件：

1. 将gnu拷贝至任意目录，打开gnu/arm\_linux.cmake，修改

set(CMAKE\_C\_COMPILER gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin/arm-linux-gnueabihf-gcc)

set(CMAKE\_CXX\_COMPILER gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin/arm-linux-gnueabihf-g++)；

在gnu/aarch32/前面加入当前路径，例如：

set(CMAKE\_C\_COMPILER /home/cas/gnu/aarch32/lin/gcc-arm-linux-gnueabi/bin/arm-linux-gnueabihf-gcc)

1. 在vscode中打开工程目录，按F1，输入CMAKE选择CMAKE:Configure，

select\_kit选择unspecified。

文本

描述已自动生成

selet\_cmakelist选择{workspacefolder}/ctl\_sever\_cpp/cmakelist.txt。图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

按F1，输入CMAKE选择CMAKE:Edit User-Local CMAKE Kits，

在[ ] 加入以下语句

{

"name": "arm\_linux\_corss",

"toolchainFile": "/home/cas/gnu/arm\_linux.cmake "

}

在前一个大括号前加“ ，”，如图：

文本

描述已自动生成

其中"toolchainFile"：后面要指向arm\_linux.cmake所在路径。

1. 打开ctl\_sever\_cpp/cmakelist.txt，按ctl+s保存，会自动生成build文件夹。在界面最底部有cmake工具栏，



点击No activate kit，选择arm\_linux\_cross，之后按F7或点击build开始编译

编译的二进制文件在build/bin中