

1 下载源码

下载 PX4 源码：

```
git clone -b stable https://github.com/PX4/PX4-Autopilot.git --recursive
```

这里下载的是 stable 版本的固件，即最新稳定版的固件，如果要下载指定版本的固件，则将上面的 stable 换成版本号，如下载 1.11.3 版本的固件，则下载命令为：

```
git clone -b v1.11.3 https://github.com/PX4/PX4-Autopilot.git --recursive
```

用 git clone 下载 px4 PX4-Autopilot 时，如果一些子模块因为网络原因下载失败。可以在网络正常后，在 PX4-Autopilot 路径下执行以下命令继续下载剩余的子模块。

```
git submodule update --init --recursive
```

2 配置编译环境

源码下载完毕后，输入：

```
cd PX4-Autopilot
sudo bash ./Tools/setup/ubuntu.sh
```

如果装的是 PX4 1.12.3 的编译环境，会无法编译 1.13.0 的固件，需要用 1.13.0 的脚本进行配置，如果装的是 1.13.0 的版本的编译环境，则可以编译 1.12.3 的固件

在执行这个脚本的过程中会自动安装 px4 的编译环境和 gazebo 仿真环境，环境安装过程中如果报下载失败的错误，一般是网速问题，可以解决网速问题后重新执行脚本。直到安装成功。如果报以下错误：

http://packages.ros.org/ros/ubuntu xenial InRelease: 由于没有公钥，无法验证下列签名

执行下面命令再重新执行脚本，最后的码，根据不同的提示的进行修改

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys F42ED6FBAB17C654
```

如果报以下错误

**Installing PX4 simulation dependencies E: 无法获得锁
/var/lib/dpkg/lock-frontent - open (11: 资源暂时不可用) E: 无法获取 dpkg 前端锁
(/var/lib/dpkg/lock-frontent)，是否有其他进程正占用它？ E: 无法获得锁
/var/lib/dpkg/lock-frontent - open (11: 资源暂时不可用) E: 无法获取 dpkg 前端锁
(/var/lib/dpkg/lock-frontent)，是否有其他进程正占用它？**

执行以下三条命令再重新执行脚本

```
sudo rm /var/lib/dpkg/lock-frontent
sudo rm /var/cache/apt/archives/lock
sudo rm /var/lib/dpkg/lock
```

如果执行脚本过程中出现以下报错

错误:7 http://packages.ros.org/ros/ubuntu bionic InRelease

则执行下面的命令再重新运行脚本

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys F42ED6FBAB17C654
```

如果执行脚本过程中出现以下报错

```
ka
8
un
(
-----
Command "/usr/bin/python3 -u -c "import setuptools, tokenize;__file__='/tmp/pip-build-9
57yuv4c/pillow/setup.py';f=getattr(tokenize, 'open', open)(__file__);code=f.read().repl
ace('\r\n', '\n');f.close();exec(compile(code, __file__, 'exec'))" install --record /tm
p/pip-wcbiiqej-record/install-record.txt --single-version-externally-managed --compile
--user --prefix=" failed with error code 1 in /tmp/pip-build-957yuv4c/pillow/
mbot@ubuntu:~/PX4-Autopilot$ sudo bash ./Tools/setup/ubuntu.sh
Ubuntu 18.04
```

则执行下面的命令再重新运行脚本

```
sudo apt-get install libjpeg-dev zlib1g-dev
```

直到出现下面提示重启电脑且无报错说明脚本执行完毕

```
libvenge-0.0-0 libsgutils2-2 libssh-4 libsuitesparseconfig5 libvncclient1
libwinpr2-2 libxapian30 libxmlsec1 libxmlsec1-nss lp-solve media-player-info
python3-mako python3-markupsafe syslinux syslinux-common syslinux-legacy
usb-creator-common
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 317 not upgraded.

Relogin or reboot computer before attempting to build NuttX targets
mbot@ubuntu:~/PX4-Autopilot$
```

3 编译固件

安装完环境后编译生成 gazebo 仿真所需的 sdf 文件

```
make px4_sitl_default gazebo
```

如果要编译下载到飞控的代码，需重启电脑编译，以 v5 固件为例

```
make px4_fm-v5_default
```

如果在编译代码时报错

```
ModuleNotFoundError: No module named 'menuconfig' CMake Error at
cmake/kconfig.cmake:6 (message): kconfiglib is not installed or not
in PATH
please install using "pip3 install kconfiglib"
Call Stack (most recent call first): CMakeLists.txt:174 (include)
```

执行下面命令即可解决

```
sudo -H python3 -m pip install kconfiglib
```

如果在编译代码时报错

```
Failed to import genmsg: No module named 'genmsg'
You may need to install it using:
pip3 install --user pyros-genmsg
```

执行下面命令即可解决

```
sudo -H python3 -m pip install pyros-genmsg
```

如果在编译代码时报错

```
Failed to import toml: No module named 'toml'
You may need to install it using:
pip3 install --user toml
```

执行

```
sudo -H python3 -m pip install toml
```

如果在编译代码时报错

```
Failed to import jinja2: No module named 'jinja2'
```

```
You may need to install it using:
```

```
pip3 install --user jinja2
```

执行

```
sudo -H python3 -m pip install jinja2
```

如果在编译代码时报错

```
Failed to import packaging: No module named 'em'
```

则执行下面语句安装软件包

```
sudo -H python3 -m pip install em
```

如果在编译代码时报错

```
Failed to import jsonschema: No module named 'jsonschema'
```

```
You may need to install it using:
```

```
pip3 install --user jsonschema
```

执行

```
sudo -H python3 -m pip install jsonschema
```

如果报

```
Failed to import packaging: No module named 'packaging'
```

```
You may need to install it using:
```

```
pip3 install --user packaging
```

执行

```
sudo -H python3 -m pip install packaging
```

出现以下提示表示安装成功

```
Collecting packaging
  Downloading https://files.pythonhosted.org/packages/05/8e/8de486cbd03baba4deef
4142bd643a3e7bbe954a784dc1bb17142572d127/packaging-21.3-py3-none-any.whl (40kB)
  100% |████████████████████████████████████████| 40kB 180kB/s
Collecting pyparsing!=3.0.5,>=2.0.2 (from packaging)
  Downloading https://files.pythonhosted.org/packages/80/c1/23fd82ad3121656b5853
51aba6c19761926bb0db2ebd9e4ff09a43a3fcc/pyparsing-3.0.7-py3-none-any.whl (98kB)
  100% |████████████████████████████████████████| 102kB 403kB/s
Installing collected packages: pyparsing, packaging
Successfully installed packaging-21.3 pyparsing-3.0.7
```

如果报错的话可以尝试执行下面的命令安装

```
sudo pip install --ignore-installed packaging
```

同理如果缺少 toml/jinja2/numpy 等其他包，安装方法同上，只是把最后的包名换一下。

如果编译时报以下错

```
AttributeError: module 'em' has no attribute 'Interpreter'
```

则执行下面两条命令并重新编译

```
sudo -H python3 -m pip uninstall em
```

```
sudo -H python3 -m pip install --ignore-installed empty
```

如果编译时报以下错

```
ModuleNotFoundError: No module named 'future'
```

执行

```
sudo -H python3 -m pip install --ignore-installed future
```

再重新编译

常见的缺少的包的安装命令如下：

```
sudo -H python3 -m pip install kconfiglib
sudo -H python3 -m pip install jinja2
sudo -H python3 -m pip install jsonschema
sudo -H python3 -m pip install empy
sudo -H python3 -m pip install pyros-genmsg
sudo -H python3 -m pip install packaging
sudo -H python3 -m pip install toml
sudo -H python3 -m pip install numpy
sudo -H python3 -m pip install future
```

如果上面的命令安装后还是报错，可以用下面的命令安装：

```
pip3 install kconfiglib
pip3 install --user empy
pip3 install --user jsonschema
pip3 install --user jinja2
pip3 install --user pyros-genmsg
pip3 install --user packaging
pip3 install --user toml
pip3 install --user numpy
```

如果编译时报错：

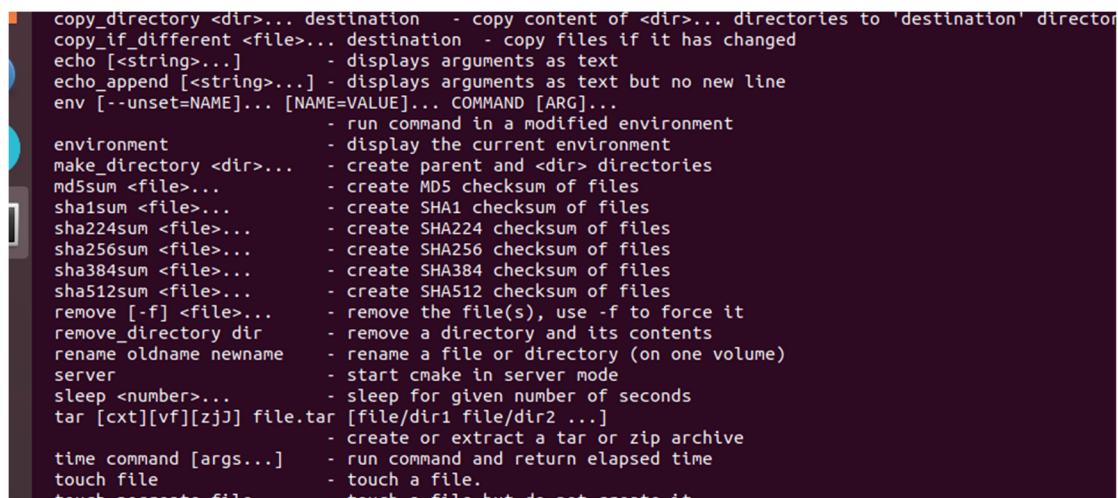
```
CMake Error at CMakeLists.txt:235 (project): The CMAKE_C_COMPILER:
arm-none-eabi-gcc
is not a full path and was not found in the PATH.
```

执行下面命令解决：

```
sudo apt-get install gcc-arm-none-eabi
```

注意编译的时候 Firmware 路径不能有英文的括号，否则编译会报错或者一直循环 config

笔者在编译成功后，利用 **make clean** 清除编译后再次编译会报如下错



```
copy_directory <dir>... destination - copy content of <dir>... directories to 'destination' director
copy_if_different <file>... destination - copy files if it has changed
echo [<string>...] - displays arguments as text
echo_append [<string>...] - displays arguments as text but no new line
env [--unset=NAME]... [NAME=VALUE]... COMMAND [ARG]...
environment - run command in a modified environment
environment - display the current environment
make_directory <dir>... - create parent and <dir> directories
md5sum <file>... - create MD5 checksum of files
sha1sum <file>... - create SHA1 checksum of files
sha224sum <file>... - create SHA224 checksum of files
sha256sum <file>... - create SHA256 checksum of files
sha384sum <file>... - create SHA384 checksum of files
sha512sum <file>... - create SHA512 checksum of files
remove [-f] <file>... - remove the file(s), use -f to force it
remove_directory dir - remove a directory and its contents
rename oldname newname - rename a file or directory (on one volume)
server - start cmake in server mode
sleep <number>... - sleep for given number of seconds
tar [cxt][vf][zjJ] file.tar [file/dir1 file/dir2 ...] - create or extract a tar or zip archive
time command [args...] - run command and return elapsed time
touch file - touch a file.
touch_append file - touch a file but do not create it
```

解决办法是用 **make distclean** 重新清除一下或者将 build 文件夹删除即可。