1 下载源码

在 ubuntu（本例程使用的版本为 20.04.2）的环境下，输入：

1. **git clone https://github.com/ArduPilot/ardupilot.git**
2. **cd ardupilot**
3. **git submodule init**
4. **git submodule update**

在执行 git submodule update 时若出现报错没有更新完毕子模块，则继续执行该命令，直至更新完毕。

如果需要切换分支可以用 git checkout 分支名，如：

git checkout Copter-4.4.0

2 配置编译环境

在 ardupilot 目录下执行下面的命令安装环境：

Tools/environment_install/install-prereqs-ubuntu.sh -y

```
one-eabi-10-2020-q4-major/bin:/opt/ros/noetic/bin:/home/cwkj/.local/bin:/usr/loc
al/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games
:/snap/bin
+ echo 'Done!'
Done!
+ [[ '' -ne 1 ]]
+ '[' -d .git ']'
+ heading 'Update git submodules'
+ echo '##############################################'
##############################################
+ echo Update git submodules
Update git submodules
+ echo '##############################################'
##############################################
+ cd /home/cwkj/4.2.3/ardupilot
+ git submodule update --init --recursive
+ echo 'Done!'
Done!
+ false
+ echo '---------- ./Tools/environment_install/install-prereqs-ubuntu.sh end ---
-------'
---------- ./Tools/environment_install/install-prereqs-ubuntu.sh end ----------
cwkj@ubuntu:~/4.2.3/ardupilot$
```

配置成功后执行：

. ~/.profile

这里建议使用最新版的 apm 固件的环境安装脚本，因为旧版固件的安装脚本安装完环境后，能编译旧版固件，但不一定能编译新版固件，例如这里使用 4.3.1 版的固件的脚本配置完环境后，能编译 4.3.1 版固件，但在编译 4.3.7 版固件时，却报错：

```
[109/111] ChibiOS: Compiling bouncebuffer.c
[110/111] ChibiOS: Compiling watchdog.c
[111/111] ChibiOS: Compiling ch.cpp

make: *** 没有规则可制作目标“CrashCatcher_armv7m.S”，由“modules/ChibiOS/obj/Cras
hCatcher_armv7m.o” 需求。 停止。
make: *** 没有规则可制作目标“CrashCatcher_armv7m.S”，由“modules/ChibiOS/obj/Cras
hCatcher_armv7m.o” 需求。 停止。
make: *** 正在等待未完成的任务....

Waf: Leaving directory `/home/cwkj/3/ardupilot/build/fmuv3'
Build failed
 -> task in 'ChibiOS_lib' failed (exit status 2):
```

解决办法也比较简单，就是用 4.3.7 版固件的脚本再重新执行一遍，就可以编译 4.3.7 版固件了。

如果编译 master 版本的固件正常，但是在切到 Copter 4.3.7 这个 tag 后，执行 ./waf configure --board fmuv3 时报下面的错

```
env set FLASH_TOTAL=2080768
env set HAS_EXTERNAL_FLASH_SECTIONS=0
env set CHIBIOS_BUILD_FLAGS=USE_FATFS=yes CHIBIOS_STARTUP_MK=os/common/startup/ARMCMx/compilers/GCC/mk/startup_stm32f4xx.mk CHIBIOS_PLATFORM_MK=os/hal/ports/STM32/ST
.mk MCU=cortex-m4 ENV_UDEFS=-DCHPRINTF_USE_FLOAT=1
Traceback (most recent call last):
  File "/home/cwkj/3/ardupilot/modules/waf/waflib/Scripting.py", line 158, in waf_entry_point
    run_commands()
  File "/home/cwkj/3/ardupilot/modules/waf/waflib/Scripting.py", line 251, in run_commands
    ctx = run_command(cmd_name)
  File "/home/cwkj/3/ardupilot/modules/waf/waflib/Scripting.py", line 235, in run_command
    ctx.execute()
  File "/home/cwkj/3/ardupilot/modules/waf/waflib/Configure.py", line 159, in execute
    super(ConfigurationContext, self).execute()
  File "/home/cwkj/3/ardupilot/modules/waf/waflib/Context.py", line 204, in execute
    self.recurse([os.path.dirname(g_module.root_path)])
  File "/home/cwkj/3/ardupilot/modules/waf/waflib/Context.py", line 286, in recurse
    user_function(self)
  File "/home/cwkj/3/ardupilot/wscript", line 445, in configure
    cfg.get_board().configure(cfg)
  File "/home/cwkj/3/ardupilot/Tools/ardupilotwaf/boards.py", line 48, in configure
    self.configure_env(cfg, env)
  File "/home/cwkj/3/ardupilot/Tools/ardupilotwaf/boards.py", line 862, in configure_env
    cfg.load('chibios')
  File "/home/cwkj/3/ardupilot/modules/waf/waflib/Configure.py", line 270, in load
    func(self)
  File "/home/cwkj/3/ardupilot/Tools/ardupilotwaf/chibios.py", line 577, in configure
    setup_canmgr_build(cfg)
  File "/home/cwkj/3/ardupilot/Tools/ardupilotwaf/chibios.py", line 468, in setup_canmgr_build
    cfg.srcnode.find_dir('modules/uavcan/libuavcan/include').abspath(),
AttributeError: 'NoneType' object has no attribute 'abspath'
cwkj@ubuntu:~/3/ardupilot$ Tools/environment_install/install-prereqs-ubuntu.sh -y
```

解决办法也是在切到 Copter 4.3.7 这个 tag 后再执行一下配环境的脚本就可以了

3 编译固件

编译固件前，要配置编译的固件的目标硬件，这里使用的是 pix2.4.8 飞控，所以使用 fmuv3 的固件，配置如下：

./waf configure --board fmuv3

```
Checking for program 'gcc'                      : /usr/bin/gcc
Gtest                                           : STM32 boards currently don't
pport compiling gtest
Checking for program 'arm-none-eabi-size'       : /opt/gcc-arm-none-eabi-10-202
q4-major/bin/arm-none-eabi-size
Benchmarks                                      : disabled
Unit tests                                      : disabled
Scripting                                       : enabled
Scripting runtime checks                        : enabled
Debug build                                     : disabled
Coverage build                                  : disabled
SITL 32-bit build                               : disabled
Checking for program 'rsync'                    : /usr/bin/rsync
'configure' finished successfully (1.139s)
cwkj@ubuntu:~/4.2.3/ardupilot$
```

然后用下面的命令编译四旋翼固件：

./waf copter

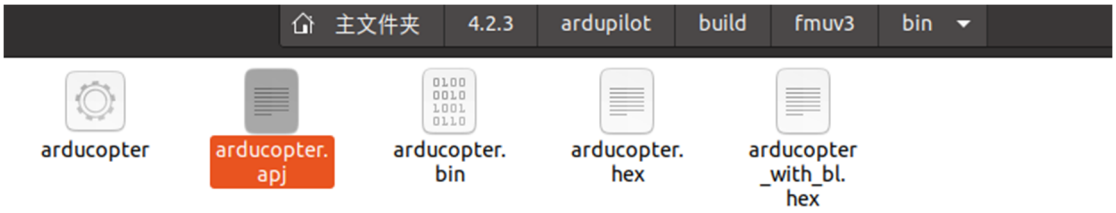
编译成功如下：

```
[891/895] Generating bin/arducopter.bin
[892/895] app_descriptor build/fmuv3/bin/arducopter.bin
No APP_DESCRIPTOR found
[893/895] apj_gen build/fmuv3/bin/arducopter.bin
[894/895] bin cleanup build/fmuv3/bin/arducopter.bin
[895/895] Generating bin/arducopter.hex
Waf: Leaving directory `/home/cwkj/4.2.3/ardupilot/build/fmuv3'

BUILD SUMMARY
Build directory: /home/cwkj/4.2.3/ardupilot/build/fmuv3
Target          Text (B)  Data (B)  BSS (B)  Total Flash Used (B)  Free Flash (B
)
-----------------------------------------------------------------------------
-
bin/arducopter   1568964      3468   193364                1572432          50832
8

Build commands will be stored in build/fmuv3/compile_commands.json
'copter' finished successfully (6m25.623s)
cwkj@ubuntu:~/4.2.3/ardupilot$
```

编译后生成的固件在下图的目录



清除编译：**./waf copter clean**

设置 git 标签：默认的 git 分支处于 master，这个分支时开发者分支，正常使用的话建议稳定版，使用 **git tag** 命令查看所有的 tag

切换成功后，可以使用 **git branch** 命令查看