基于 CODING 的 aerocar 开发流程

编写：_____

校验：_____

审核：_____

修订页

版本	内容	修订人
V1.0	针对 aerocar 项目的开发流程	黄潇剑

1.环境搭建

1.1 安装 ubuntu20.04

1.2 注册 CODING，进入团队

1.2.1 注册 CODING

参考链接：注册与邀请 - 什么是 DevOps? DevOps 介绍 | CODING DevOps

1.2.2 由团队负责人邀请进入团队

1.3 配置 aerocar

1.3.1 安装 GIT

在根目录下打开终端，输入指令：sudo apt install git

1.3.2 克隆 aerocar 代码

方法一：输入账号和密码
在根目录下打开终端，输入指令：
git clone https://e.coding.net/ifccq/aerocar/AeroCar.git --recursive（需要你的 CODING 账号和密码）
方法二：SSH 公钥
参考链接：https://coding.net/help/docs/repo/ssh/config.html

1.3.3 更新仓库子模块

在根目录下打开终端，输入指令：
cd AeroCar
git submodule update --init --recursive
cd src/drivers/uavcan/libuavcan
git remote add IFC https://e.coding.net/ifccq/aerocar/libuavcan.git（需要你的 CODING 账号和密码）
git fetch IFC
cd mavlink/include/mavlink/v2
git remote add IFC https://e.coding.net/ifccq/aerocar/mavlink_c_library_v2.git（需要你的 CODING 账号和密码）
git fetch IFC
cd Tools/sitl_gazebo
git remote add IFC https://e.coding.net/ifccq/aerocar/sitl_gazebo.git（需要你的 CODING 账号和密码）
git fetch IFC

1.3.4 安装所有工具链

在根目录下打开终端，输入指令：
sudo apt install make
bash ./Aerocar/Tools/setup/ubuntu.sh
sudo apt-get install -y gcc-arm-linux-gnueabihf g++-arm-linux-gnueabihf
sudo apt install python3-pip
sudo apt install ninja-build exiftool ninja-build protobuf-compiler libeigen3-dev genromfs xmlstarlet libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev python3-pip
【提示】：上述安装如若安装过程出现问题，可以通过更改国内软件源或者科学上网，多次运行以免安装不齐全。

1.3.5 重启计算机

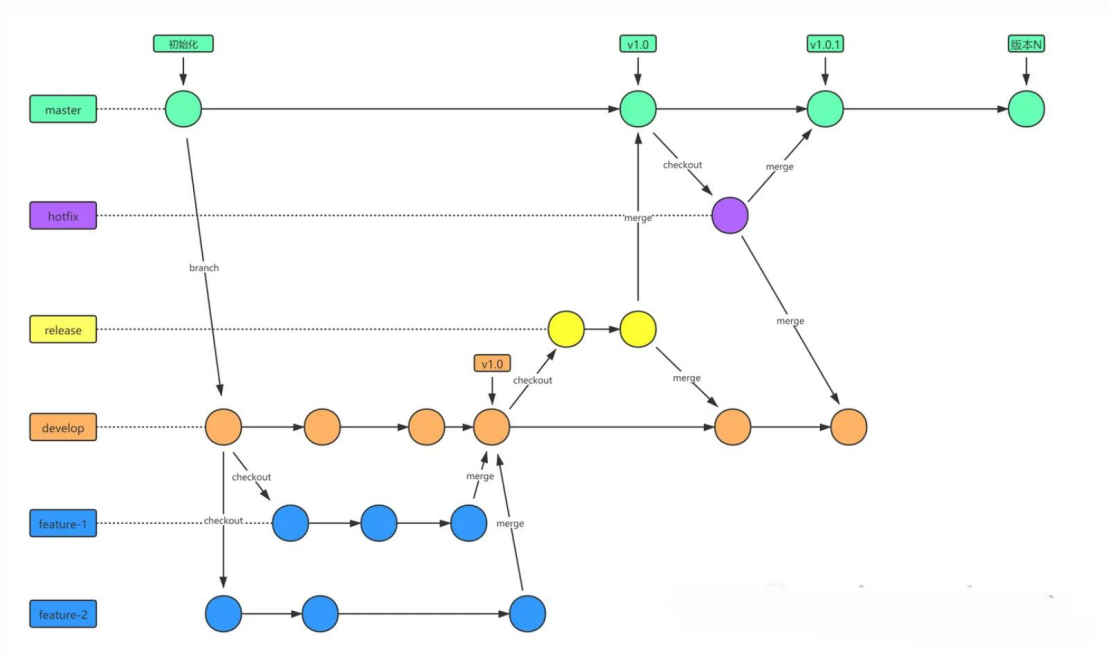
1.3.6 安装 ROS

参照官网教程：http://wiki.ros.org/cn/Installation/Ubuntu

1.3.7 安装 QGC

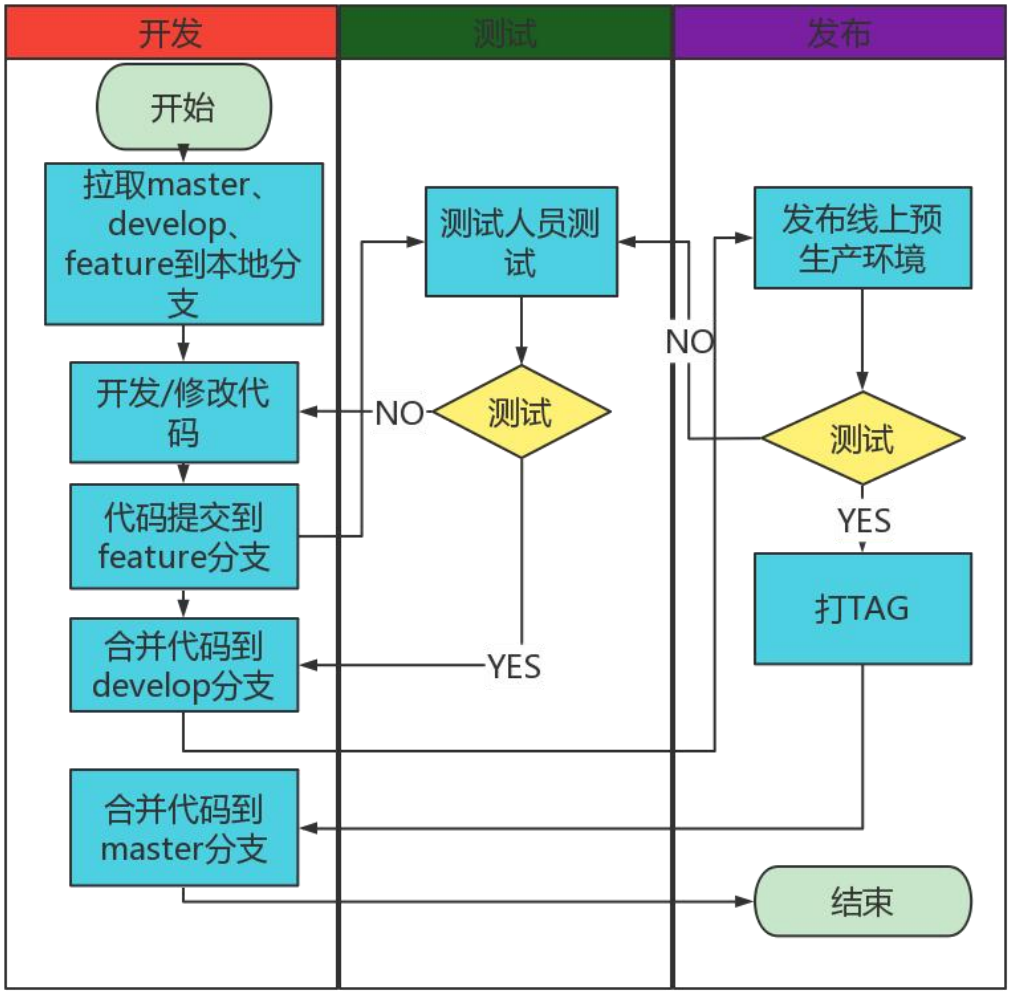
参照官网教程：
https://docs.qgroundcontrol.com/master/en/getting_started/download_and_install.html
1.3.8 配置仿真环境
在根目录下打开终端，输入指令：
cd AeroCar
make px4_sitl gazebo_aerocar_v2

2. 开发流程

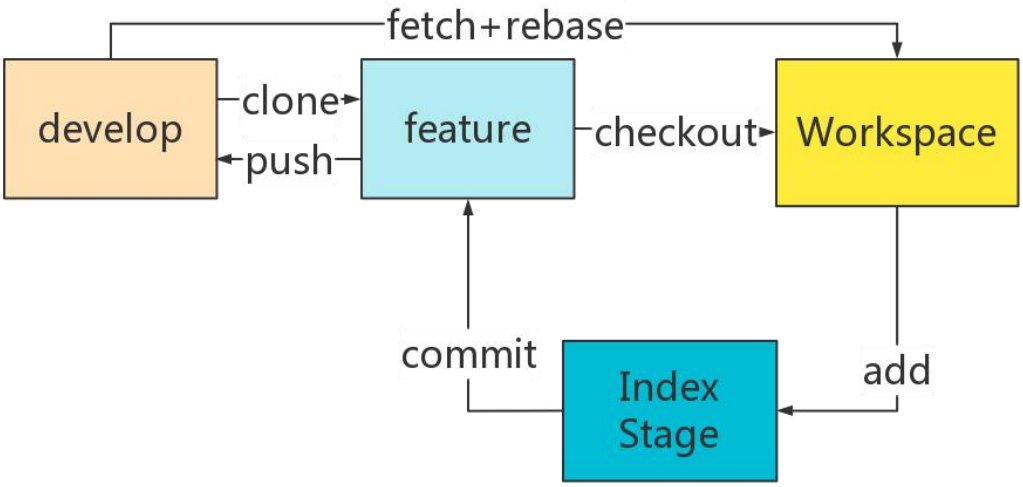


项目开发流程图

(1) master 分支记录了每次版本发布历史和 tag 标记。
(2) develop 分支记录了所有开发的版本历史。
(3) develop 分支仅第一次创建时从 master 分支拉取。
(4) feature 分支是从 develop 分支拉取的分支。
(5) 每个 feature 完成后需合并到 develop 分支。
(6) release 分支是在所有功能开发自测完成后，从 develop 分支拉取的分支。
(7) release 分支一旦创建后，通常不再从 develop 分支拉取，该分支只做 bug 修复，文档生成和其他面向发布的任务。
(8) release 分支测试完成，达到上线标准后，需合并回 master 分支和 develop 分支。
(9) hotfix 分支是在线上出现 bug 之后，从 master 分支拉取的分支。
(10) hotfix 分支测试完成后，需合并回 master 分支和 develop 分支。



GIT 开发流程图

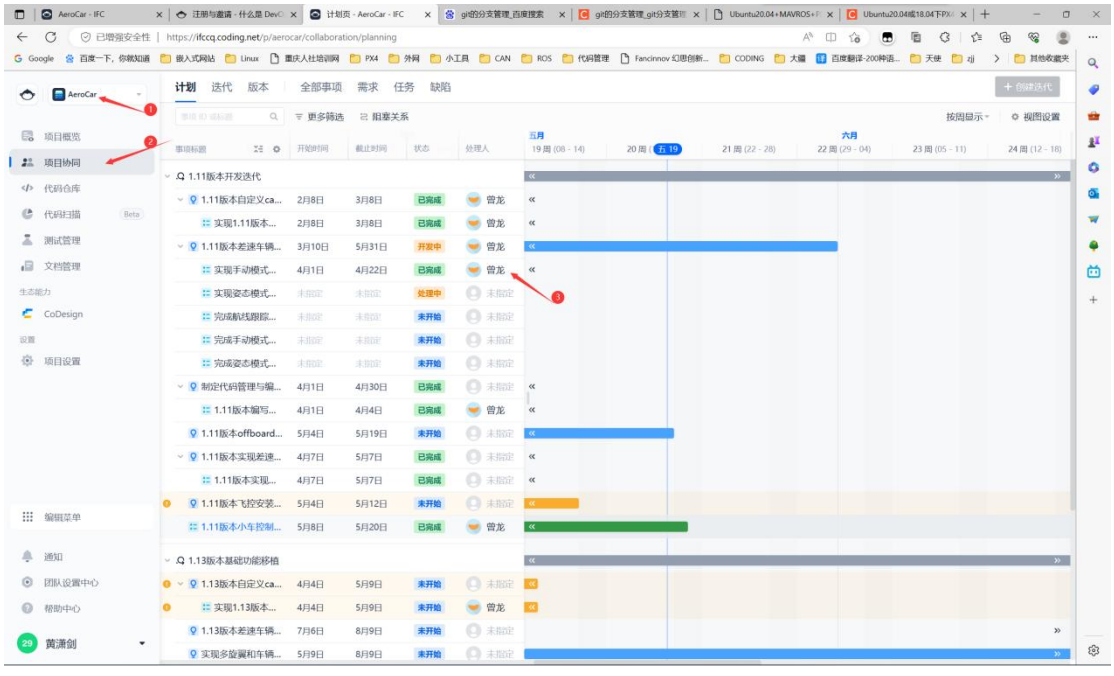


GIT 操作流程图

2.1 分支管理

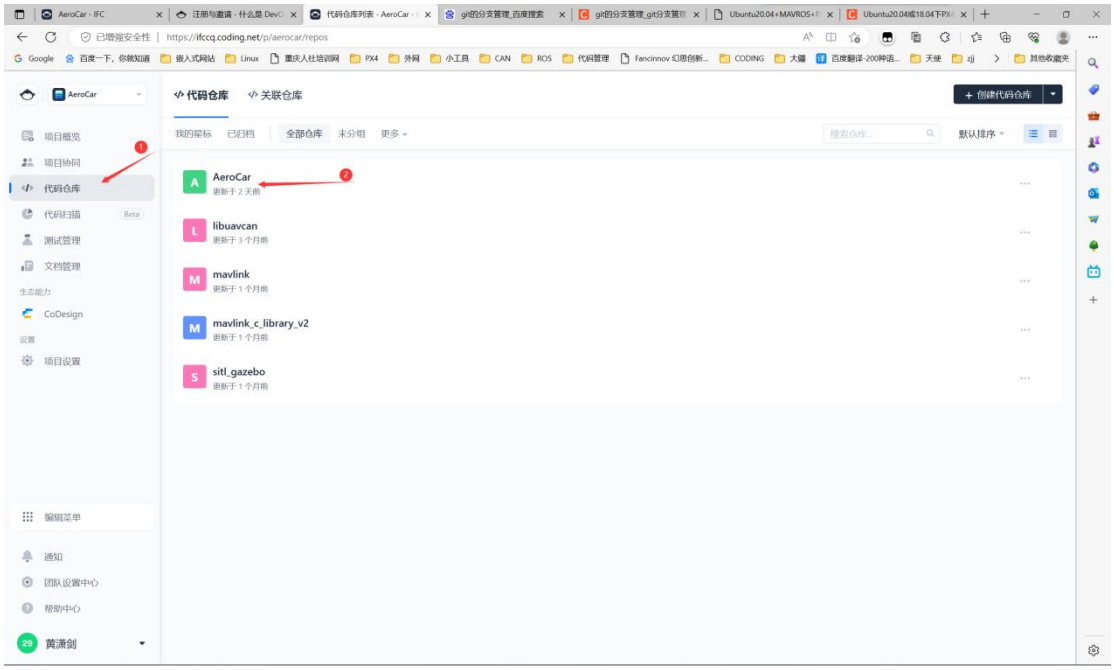
2.1.1 在 CODING 仓库查看自己的任务（feature）

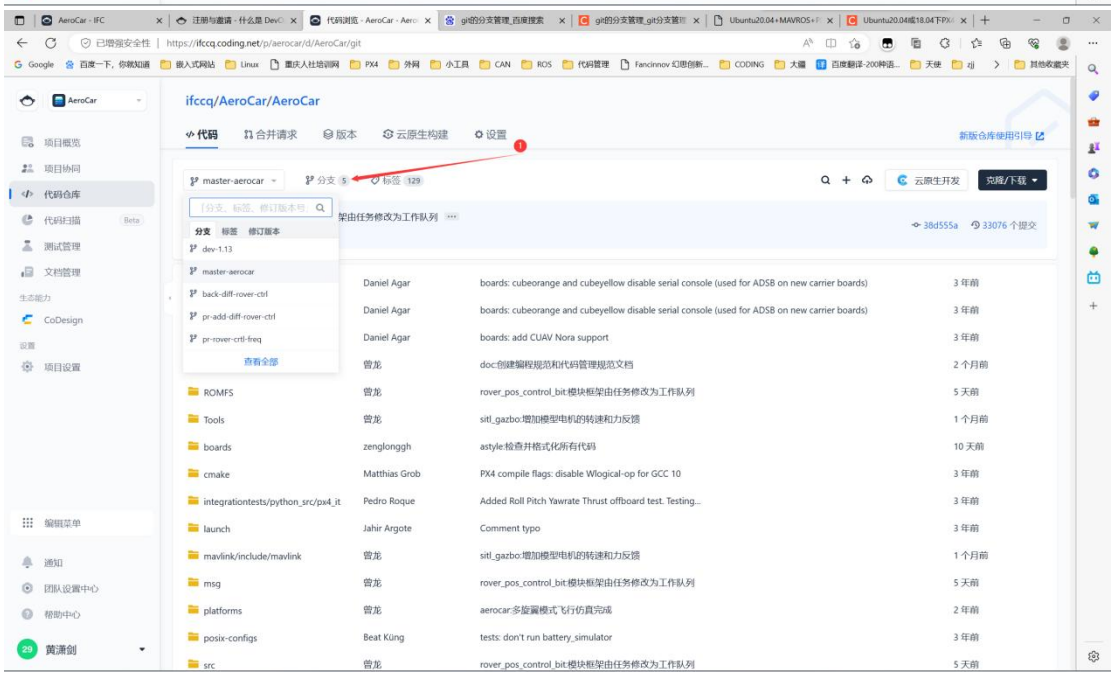
如图：

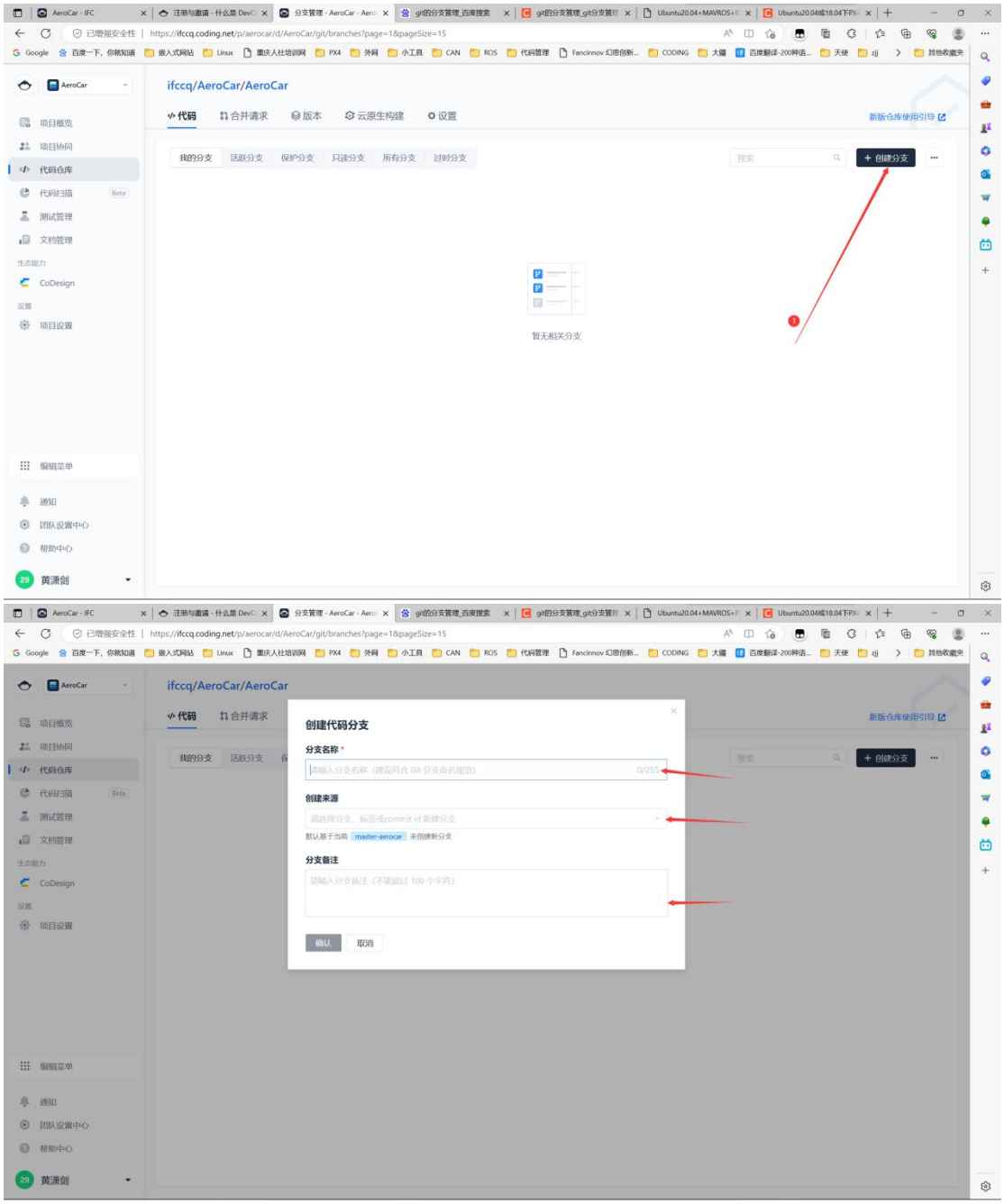


2.1.2 根据任务创建开发分支（develop）

如图：

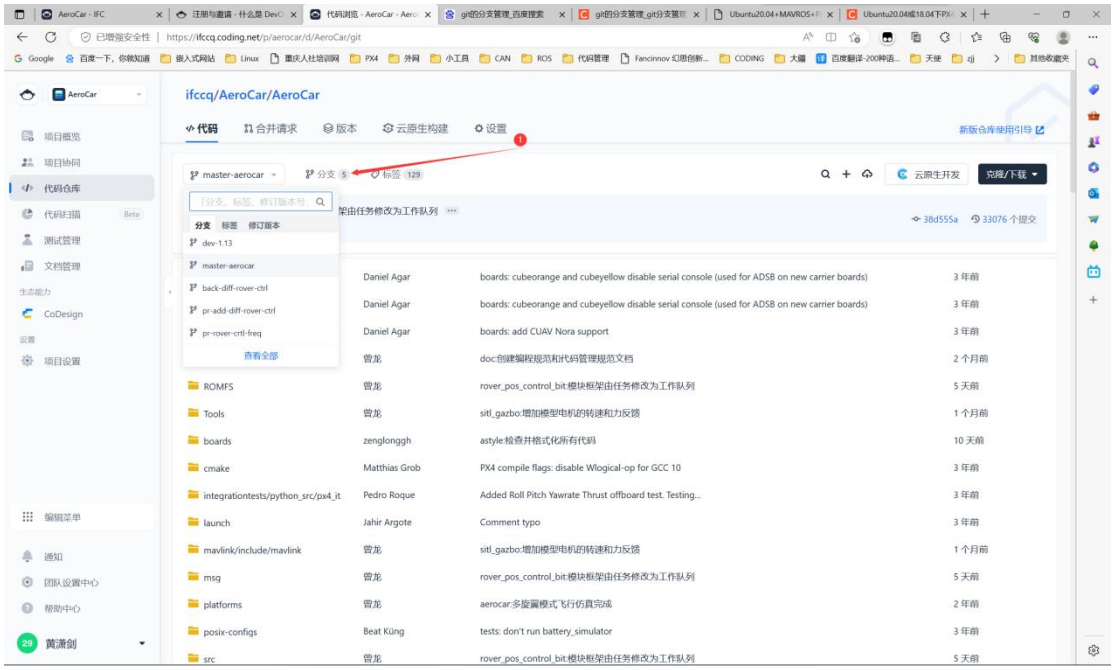

代码仓库
关联仓库
+ 创建代码仓库
AeroCar
libuavcan
mavlink
mavlink_c_library_v2
sitl_gazebo

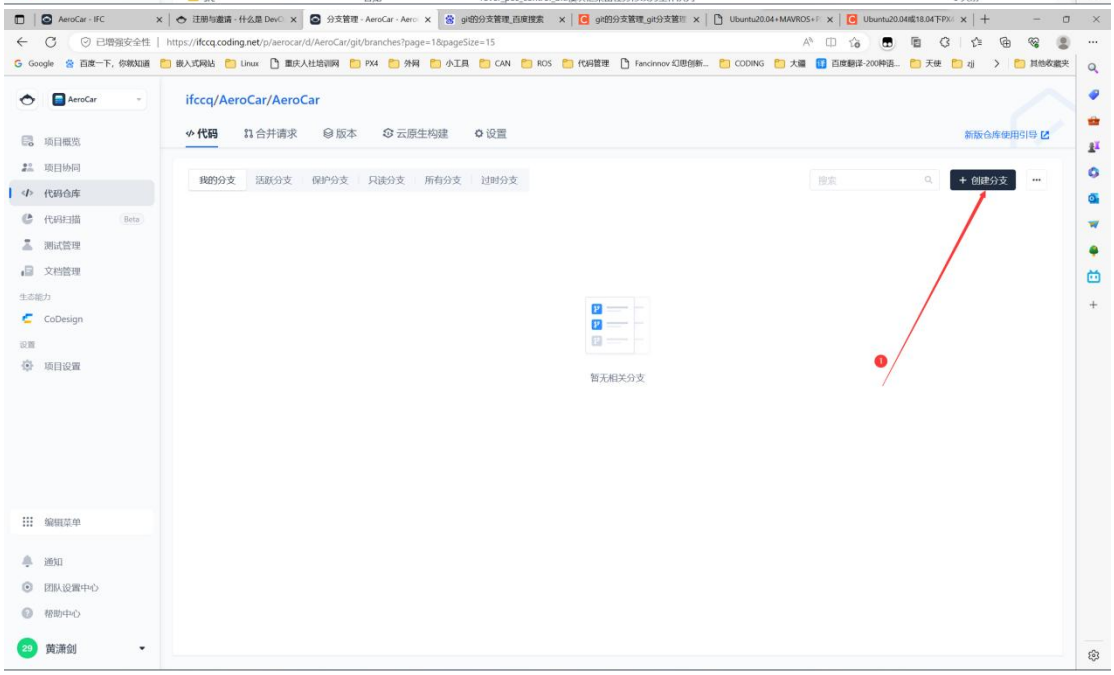

ifccq/AeroCar/AeroCar
代码
合并请求
版本
云原生构建
设置
master-aerocar
克隆/下载
ROMFS
Tools
boards
cmake
launch
mavlink/include/mavlink
msg
platforms
posix-configs
src

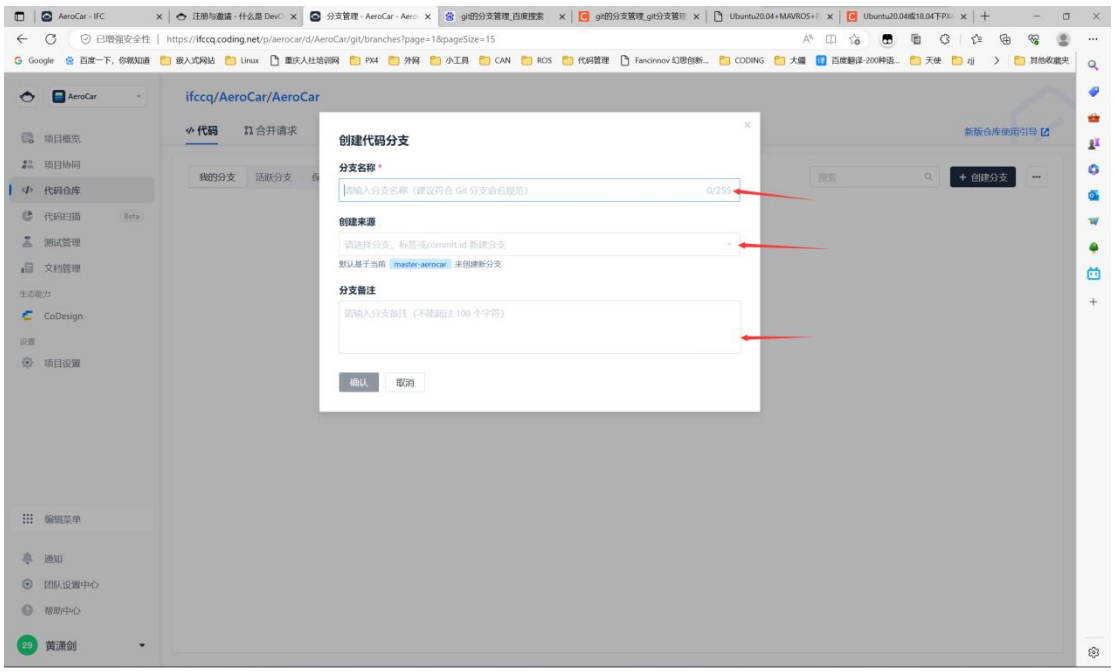


2.1.3 根据任务创建任务分支（test_develop）

选择你创建的开发分支，如图：


ifccq/AeroCar/AeroCar
代码
合并请求
版本
云原生构建
设置
新版仓库使用引导
master-aerocar
分支 5
标签 129
云原生开发
克隆/下载
dev-1.13
master-aerocar
back-diff-rover-ctrl
pr-add-diff-rover-ctrl
pr-rover-crtl-freq
查看全部
38d555a
33076 个提交
ROMFS
Tools
boards
cmake
integrationtests/python_src/px4_it
launch
mavlink/include/mavlink
msg
platforms
posix-configs
src
Daniel Agar
boards: cubeorange and cubeyellow disable serial console (used for ADSB on new carrier boards)
3 年前
boards: add CUAV Nora support
管龙
doc:创建编程规范和代码管理规范文档
2 个月前
rover_pos_control_bit:模块框架由任务修改为工作队列
5 天前
sitl_gazbo:增加模型电机的转速和力反馈
1 个月前
zenglonggh
astyle:检查并格式化所有代码
10 天前
Matthias Grob
PX4 compile flags: disable Wlogical-op for GCC 10
Pedro Roque
Added Roll Pitch Yawrate Thrust offboard test. Testing...
Jahir Argote
Comment typo
Beat Küng
tests: don't run battery_simulator
项目概览
项目协同
代码仓库
代码扫描
测试管理
文档管理
CoDesign
项目设置
编辑菜单
通知
团队设置中心
帮助中心
黄潇创


ifccq/AeroCar/AeroCar
代码
合并请求
版本
云原生构建
设置
新版仓库使用引导
我的分支
活跃分支
保护分支
只读分支
所有分支
过时分支
+ 创建分支
暂无相关分支
项目概览
项目协同
代码仓库
代码扫描
测试管理
文档管理
CoDesign
项目设置
编辑菜单
通知
团队设置中心
帮助中心
黄潇创



2.1.4 更新本地仓库（workspace）

在根目录下打开终端，输入指令：
cd AeroCar
git fetch 仓库名

2.1.5 切换到开发分支

在根目录下打开终端，输入指令：
cd AeroCar
git checkout 分支名

2.1.6 开发分支提交到 CODING 仓库

在根目录下打开终端，输入指令：
cd AeroCar
git add .
git commit -m “更改的内容”
git push

2.1.7 下拉合并主分支

开发完成后，下拉主分支然后手动合并解决冲突
git fetch
git rebase 主分支名

2.1.8 再次下拉合并主分支

因为可能合并的时间较长，远程又有人提交更新了，所以上一次下拉合并完成后接着再来一

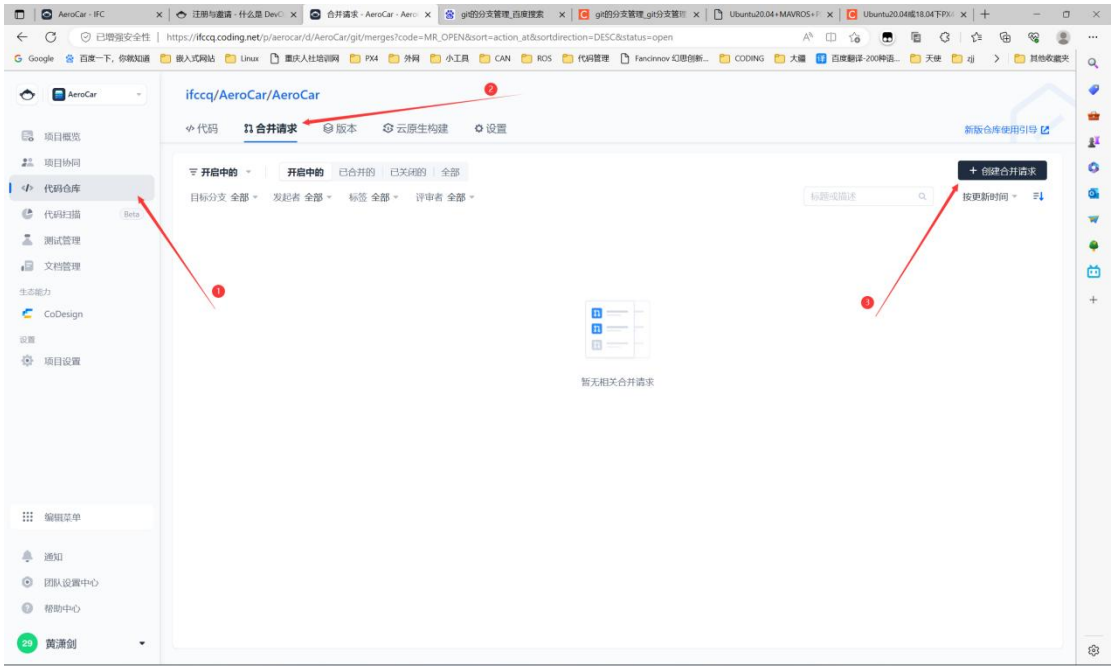
次，避免又有冲突的内容
git fetch
git rebase 主分支名

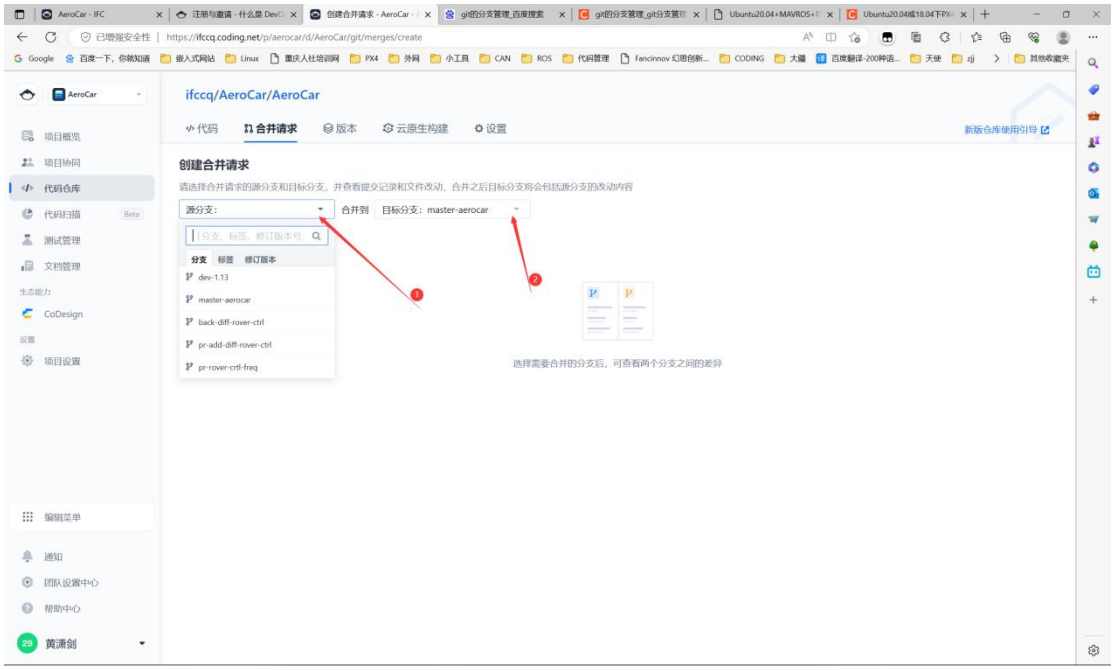
2.1.9 推送到 CODING 仓库

git push 仓库名 开发分支

2.1.10 团队负责人审核与合并，并发布到主分支

完成上述步骤后，按照下图在 CODING 上操作





2.2 代码测试

2.2.1 静态测试

在根目录下打开终端，输入指令：
cd AeroCar
./Tools/astyle/check_code_style_all.sh

2.2.2 仿真测试

2.2.3 实物测试

实物测试测试项目需留存相关测试报告、数据及视频；

2.3 下载固件

2.3.1、清除编译文件

在根目录下打开终端，输入指令：
cd AeroCar
make clean

2.3.2、编译文件

make cuav_nora_default

2.3.3、烧写固件

make cuav_nora_default upload