

## 前言

GitHub Actions 是 Microsoft 收购 GitHub 后推出的 CI/CD 服务，它提供了性能配置非常不错的虚拟服务器环境（E5 2vCPU/7G RAM），基于它可以进行构建、测试、打包、部署项目。对于公开仓库可免费无时间限制的使用，且单次使用时间长达 6 个小时，这对于编译 OpenWrt 来说是非常充足的。不过 GitHub Actions 有一定的使用门槛，首先要了解如何编写 workflow 文件。不过不用担心，博主已经编写好了相关的 workflow 文件模版，只需要按照教程的步骤来操作即可。

## 方案特点

- 免费
- 一键快速编译
- 定时自动编译
- 客制化编译
- 并发编译（可同时进行20个编译任务）
- 无需搭建编译环境（在线make menuconfig生成配置文件）
- 无需消耗自己的计算机与服务器的计算资源（性感E5在线编译）
- 无需担心磁盘空间不足（近60G磁盘空间）
- 无需使用清理文件（内核更新不怕 boom）
- 编译速度快（编译时间1-2小时）
- 编译成功率提升200%（万兆自由网络环境）
- 全新环境（杜绝编译环境不干净导致编译失败）

本解决方案是一个开放平台，任何人都可以基于此打造自己专属的编译方案。

## 项目地址

<https://github.com/P3TERX/Actions-OpenWrt>

支持项目请随手点个 star，让更多的人发现、使用并受益。

## 准备工作

- [GitHub](#) 账号
- 搭建编译环境，生成 .config 文件。（可选）

**TIPS:** 关于编译环境的搭建，推荐去看我之前写的相关文章，Windows 10 可以使用 [WSL](#)，macOS、Linux 可以使用 [Docker](#)。

## 教程更新

- 2021-01-03 新增**源码更新自动编译**使用说明
- 2020-10-11 新**触发方式**使用方法、更新**上传固件到 releases** 页面说明。
- 2020-04-25 更新 **DIY 脚本**说明、添加**自定义 feeds 配置文件**说明
- 2020-04-09 新增**上传固件到 WeTransfer**
- 2020-03-30 新增**上传固件到奶牛快传**

- 2020-02-01 新图文教程
- 2019-12-10 新增 **macOS 编译方案**使用说明
- 2019-12-06 添加 tmate 网页终端链接说明
- 2019-12-05 优化基础使用教程，添加 @lietxia 大佬的图文教程链接
- 2019-12-04 新增**云menuconfig**使用方法
- 2019-12-03 新增**并发编译**使用方法
- 2019-11-30 新增**自定义源码编译**使用方法
- 2019-11-14 全网独家首发

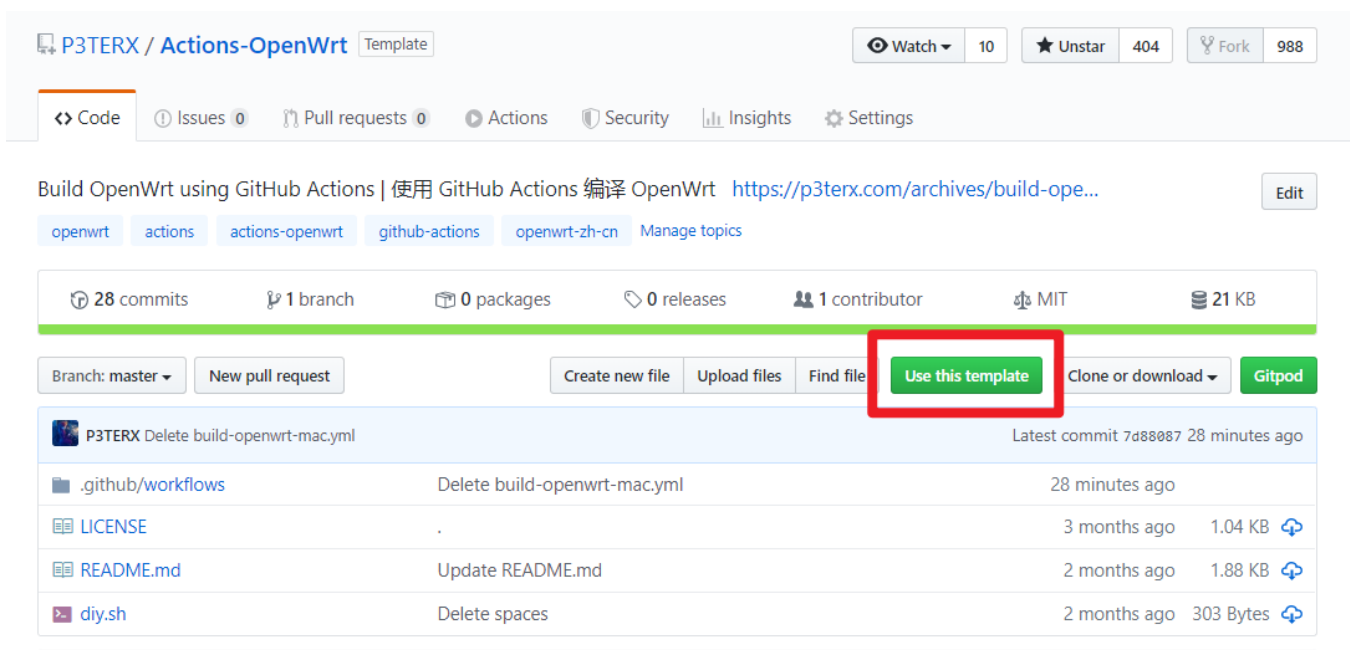
## 基础使用

首先你必须要熟悉整个 OpenWrt 的编译过程，这会让你非常容易的理解如何使用 GitHub Actions 进行编译，即使你没有成功过。因为实际上本地编译近 90% 失败的原因是因为网络问题导致的，中国大陆特色，咱也不敢多说。GitHub Actions 服务器由 Microsoft Azure 提供，拥有万兆带宽，可以使编译成功率大大提升。

- 在自己搭建编译环境中使用 Lean's OpenWrt 源码生成 .config 文件。（或使用直接 SSH 连接到 Actions 进行操作，后面有说明。）

**TIPS:** 方案默认引用 Lean 的源码，因为他的 README 影响了我开始学习编译，也就有了这个项目，而且他的源码非常的优秀。有其它需求可自行修改 workflow 文件，方法后面的进阶使用中有说明。

- 进入 P3TERX/Actions-OpenWrt 项目页面，点击页面中的 Use this template （使用这个模版）按钮。



- 填写仓库名称，然后点击 Create repository from template （从模版创建储存库）按钮。

## Create a new repository from Actions-OpenWrt

The new repository will start with the same files and folders as P3TERX/Actions-OpenWrt.

Owner: P3TERX

Repository name: OpenWrt-x86\_64-firmware ✓ 1

Great repository names are short and memorable. Need inspiration? How about [psychic-train](#)?

Description (optional)

☐ Public  
Anyone can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.

Create repository from template 2

- 经过几秒钟的等待，页面会跳转到新建的仓库，内容和我的项目是相同的。然后点击 Create new file（创建新文件）按钮。

P3TERX / OpenWrt-x86\_64-firmware  
generated from P3TERX/Actions-OpenWrt

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

No description, website, or topics provided. Edit

Manage topics

1 commit 1 branch 0 packages 0 releases 1 contributor MIT 0 Bytes

Branch: master New pull request Create new file Upload files Find file Clone or download Gitpod

P3TERX Initial commit Latest commit 4927efc 1 minute ago

.github/workflows	Initial commit	1 minute ago	
LICENSE	Initial commit	1 minute ago	1.04 KB
README.md	Initial commit	1 minute ago	1.88 KB
diy.sh	Initial commit	1 minute ago	303 Bytes

README.md

- 文件名填写为 .config，把生成的 .config 文件的内容复制粘贴到下面的文本框中。

P3TERX / OpenWrt-x86\_64-firmware  
generated from P3TERX/Actions-OpenWrt

Unwatch 1 Star 0 Fork 0

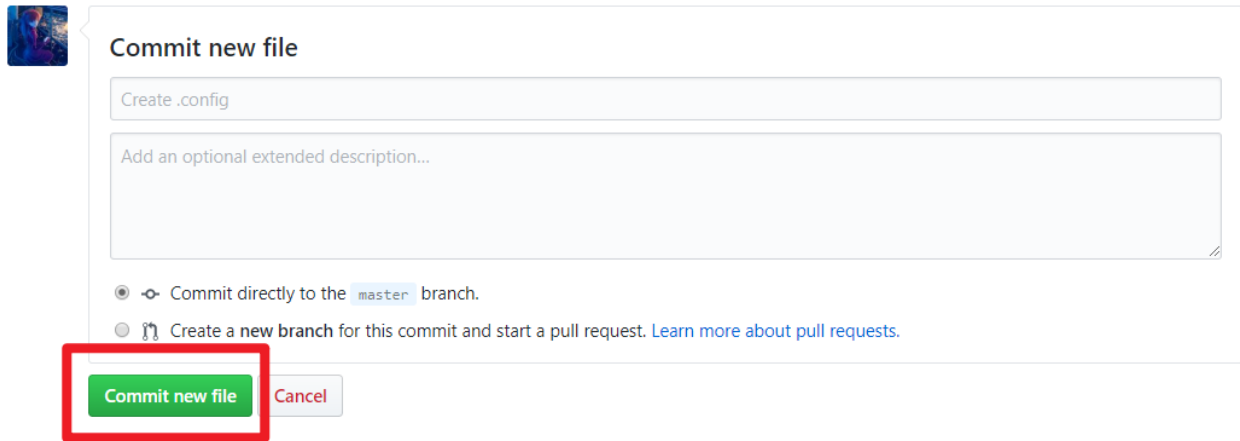
Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

OpenWrt-x86\_64-firmware .config Cancel

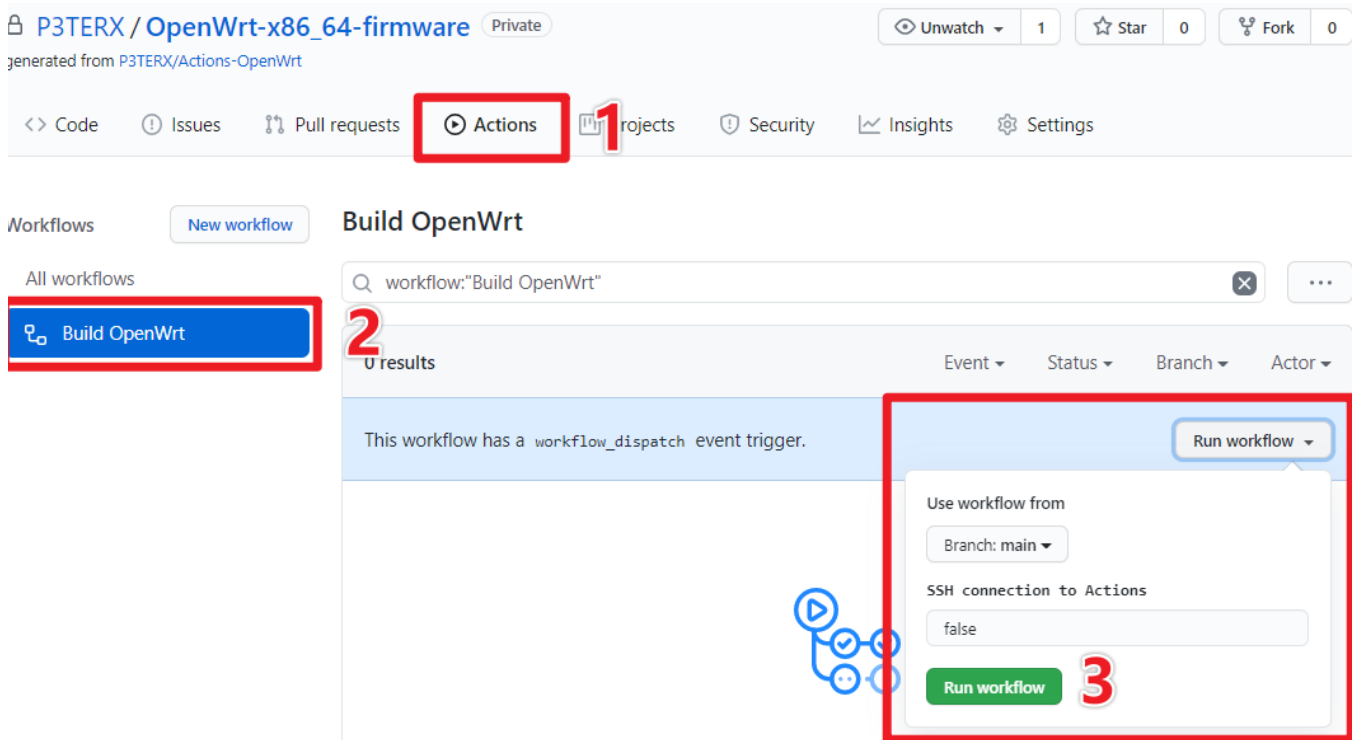
Edit new file Preview Spaces 2 No wrap

```
1 CONFIG_TARGET_x86=y
2 CONFIG_TARGET_x86_64=y
3 CONFIG_TARGET_x86_64_Generic=y
4 CONFIG_ARIA2_ASYNC_DNS=y
5 CONFIG_ARIA2_BITTORRENT=y
6 CONFIG_ARIA2_COOKIE=y
7 CONFIG_ARIA2_LIBXML2=y
8 CONFIG_ARIA2_METALINK=y
9 CONFIG_ARIA2_OPENSSL=y
10 CONFIG_ARIA2_SFTP=y
11 CONFIG_ARIA2_WFRRPC=y
```

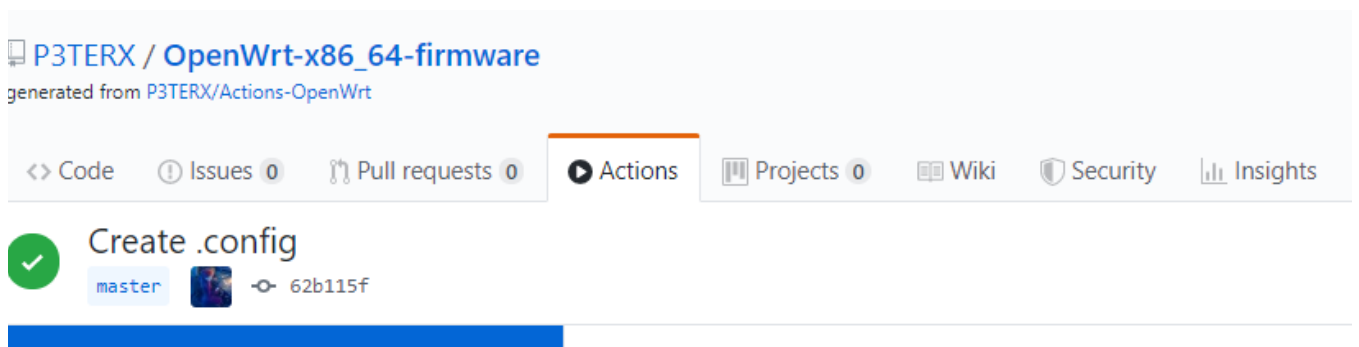
- 翻到页面最下方，点击 Commit new file（提交新文件）按钮。

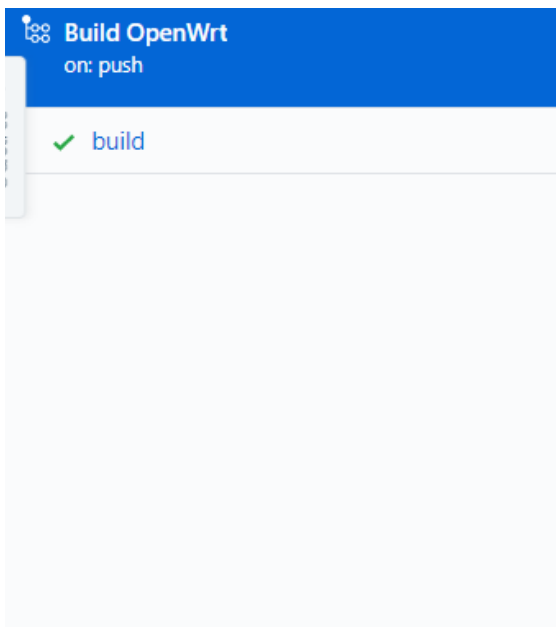


- 在 Actions 页面选择 Build OpenWrt，然后点击 Run Workflow 按钮，即可开始编译。（如果需要 SSH 连接则把 SSH connection to Actions 的值改为 true。其它详情参见进阶使用相关章节）



- 在等待编译完成的过程中，你可以进入这个页面点击右上角的 star，这是对博主最大的支持，而且还可以加快编译速度哦（雾）
- 最后经过一两个小时的等待，不出意外你就可以在 Actions 页面看到已经打包好的固件目录压缩包。





## Build OpenWrt

[This run](#)[Workflow file](#)

1 completed check in 2h 22m 43s

### Artifacts

[OpenWrt\\_firmware](#)

### Annotations

**TIPS:** 如需 ipk 文件可以在**进阶使用**章节找到方法。因为大多数人只需要固件，而且总是有萌新问固件在哪，所以现在默认只上传固件。

## 进阶使用

### 自定义环境变量与功能

► [点击查看](#)

### DIY 脚本

► [点击查看](#)

### 添加额外的软件包

▼ [点击查看](#)

- 在 DIY 脚本中加入对指定软件包源码的远程仓库的克隆指令。就像下面这样：

```
git clone https://github.com/P3TERX/xxx package/xxx
```

- 本地 make menuconfig 生成 .config 文件时添加相应的软件包，如果你知道包名可以直接写到 .config 文件中。

**TIPS:** 如果额外添加的软件包与 OpenWrt 源码中已有的软件包同名的情况，则需要把 OpenWrt 源码中的同名软件包删除，否则会优先编译 OpenWrt 中的软件包。这同样可以利用到的 DIY 脚本，相关指令应写在 diy-part2.sh。

原理是把软件包源码放到 package 目录下，编译时会自动遍历，与本地编译是一样的。当然方法不止一种，其它方式请自行探索。

## 自定义 feeds 配置文件

把 feeds.conf.default 文件放入仓库根目录即可，它会覆盖 OpenWrt 源码目录下的相关文件。

## Custom files (自定义文件)

俗称“files 大法”，在仓库根目录下新建 files 目录，把相关文件放入即可。有关详情请自行搜索了解。

## 自定义源码

默认引用的是 Lean 的源码，如果你有编译其它源码的需求可以进行替换。

▼ 点击查看

编辑 workflow 文件（.github/workflows/build-openwrt.yml），修改下面的相关环境变量字段。

```
REPO_URL: https://github.com/coolsnowwolf/lede
REPO_BRANCH: master
```

比如修改为 OpenWrt 官方源码 19.07 分支

```
REPO_URL: https://github.com/openwrt/openwrt
REPO_BRANCH: openwrt-19.07
```

**TIPS:** 注意冒号后面有空格

## 源码更新自动编译

在检测到源码更新后自动进行编译。

▼ 点击查看

- 创建 Personal access token(PAT)，勾选 repo 权限，这将用于自动触发编译工作流程。

[Settings](#) / [Developer settings](#)

GitHub Apps

OAuth Apps

Personal access tokens

### New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

#### Note

ACTIONS\_TRIGGER

What's this token for?

#### Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).



repo

Full control of private repositories



repo:status

Access commit status



repo\_deployment

Access deployment status



public\_repo

Access public repositories



repo:invite

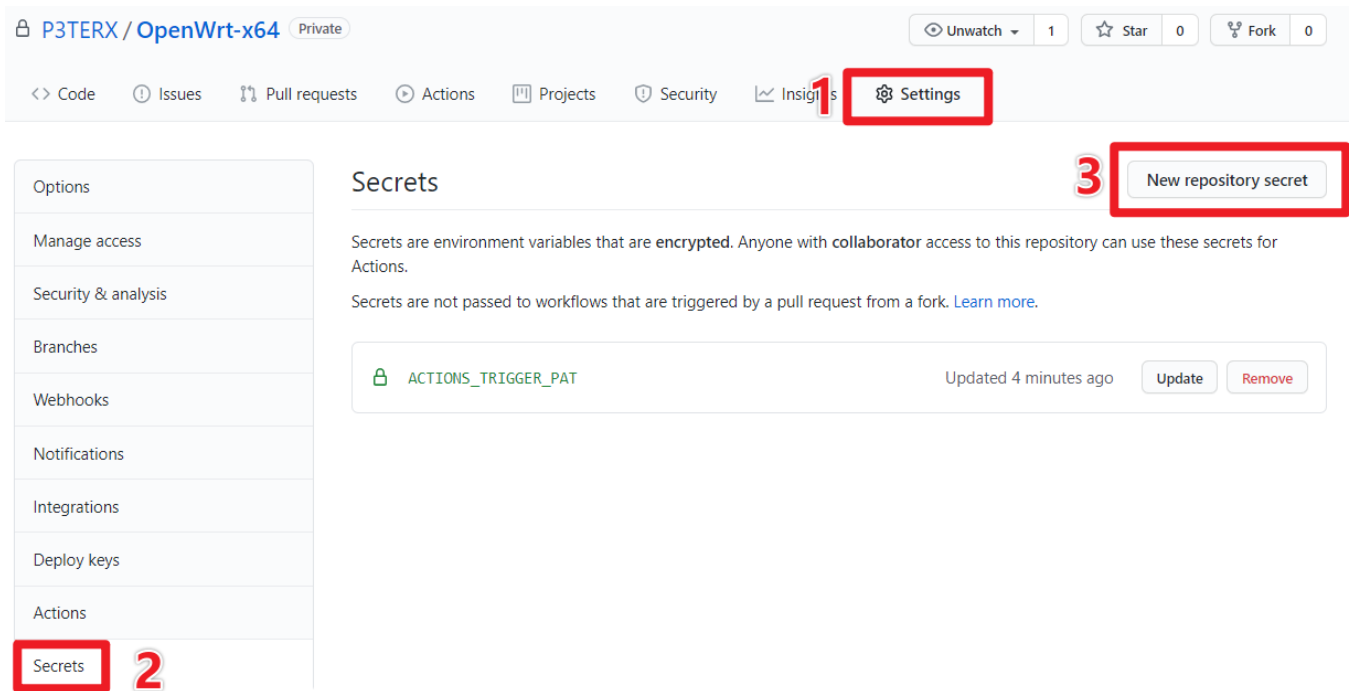
Access repository invitations



security\_events

Read and write security events

- 然后点击自己仓库的 Settings 选项卡，再点击 Secrets。添加名为 ACTIONS\_TRIGGER\_PAT 的加密环境变量，保存刚刚创建的 PAT。



- 在 Actions 页面选择 Update Checker，点击 Run workflow 手动进行一次测试运行。如果没有报错且 OpenWrt 编译工作流程被触发，则代表测试通过。
- 最后编辑 Update Checker 的 workflow 文件（.github/workflows/update-checker.yml），取消注释（删除 #）定时触发相关的部分。这里可以根据 cron 格式来设定检测的时间，时区为 UTC。

```
# schedule:
#   - cron: 0 */18 * * *
```

此外还可以根据实际情况对监视的源码仓库进行更改，如果有多个源码仓库需要监视则多复制几份相应的 workflow 文件。

## 编译多个固件

### 多 repository 方案

通过 [P3TERX/Actions-OpenWrt](#) 项目创建多个仓库来编译不同架构机型的 OpenWrt 固件。

### 多 workflow 方案

基于 GitHub Actions 可同时运行多个工作流程的特性，最多可以同时进行至少 20 个编译任务。也可以单独选择其中一个进行编译，这充分的利用到了 GitHub Actions 为每个账户免费提供的 20 个 Ubuntu 虚拟服务器环境。

#### ▼ 点击查看

假设有三台路由器的固件需要编译，比如 K2P、x86\_64 软路由、新路由 3。

- 生成它们的 .config 文件
- 分别将它们重命名为 k2p.config、x64.config、d2.config 放入本地仓库根目录。

- 复制多个 workflow 文件（.github/workflows/build-openwrt.yml）。为了更好的区分可以对它进行重命名，比如 k2p.yml、x64.yml、d2.yml。此外第一行 name 字段也可以进行相应的修改。
- 然后分别用上面修改的文件名替换对应 workflow 文件中下面两个位置的 .config，不同的机型同样可以使用不同的 DIY 脚本。

```
...
    paths:
      - '.config'
...
    CONFIG_FILE: '.config'
    DIY_SH: 'diy.sh'
...
```

## SSH 连接到 Actions

通过 tmate 连接到 GitHub Actions 虚拟服务器环境，可直接进行 make menuconfig 操作生成编译配置，或者任意的客制化操作。也就是说，你不需要再自己搭建编译环境了。这可能改变之前所有使用 GitHub Actions 的编译 OpenWrt 方式。

### ▼ 点击查看

- 在 Run Workflow 时把 SSH connection to Actions 的值改为 true（或者也可以不修改，而是通过 [webhook](#) 方式发送带有 ssh 触发关键词的请求。）
- 在触发工作流程后，在 Actions 日志页面等待执行到 SSH connection to Actions 步骤，会出现类似下面的信息：

To connect to this session copy-n-paste the following into a terminal or browser

```
ssh Y26QeagDtsPXp2mT6me5cnMRd@nyc1.tmate.io
```

```
https://tmate.io/t/Y26QeagDtsPXp2mT6me5cnMRd
```

- 复制 SSH 连接命令粘贴到终端内执行，或者复制链接在浏览器中打开使用网页终端。（网页终端可能会遇到黑屏的情况，按 Ctrl+C 即可）
- cd openwrt && make menuconfig
- 完成后按 Ctrl+D 组合键或执行 exit 命令退出，后续编译工作将自动进行。

**TIPS:** 固件目录下有个 config.seed 或者 config.buildinfo 文件，如果你需要再次编译可以使用它。

## 上传固件到奶牛快传

奶牛快传是中国大陆的一款临时文件传输分享服务网盘，特点是无限速。因国情所致，中国大陆地区 GitHub 访问速度缓慢，有些小伙伴可能无法正常下载固件，上传固件到奶牛快传是个非常好的选择。



► 点击查看

## 上传固件到 WeTransfer

WeTransfer 是荷兰的一款临时文件传输分享服务网盘，前面提到的奶牛快传实际上师从自它，二者的网站都非常相似。WeTransfer 使用的是 Amazon S3 存储并通过 Amazon CloudFront CDN 全球加速，它在中国大陆的下载体验完全不输奶牛快传，甚至某些情况下要更好。

► 点击查看

## 上传固件到 Releases 页面

GitHub 的 Releases 页面通常用于发布打包好的二进制文件，无需登录即可下载。Artifacts 和网盘有保存期限，Releases 则是永久保存的。

▼ 点击查看

编辑 workflow 文件（.github/workflows/build-openwrt.yml），将环境变量 UPLOAD\_WERANSFER 的值修改为 true：

```
UPLOAD_RELEASE: true
```

编译完成后你可以在 releases 页面找到下载链接。

**TIPS:** 为了不给 GitHub 服务器带来负担，默认保留 3 个历史记录。

## 定时自动编译（已弃用）

▼ 点击查看

**TIPS:** 源码更新是不确定的，定时编译经常是在编译没有变动的源码，无意义且浪费资源，所以不建议使用。

编辑 workflow 文件（.github/workflows/build-openwrt.yml）取消注释下面两行。

```
# schedule:
#   - cron: 0 8 * * 5
```

例子是北京时间每周五下午 4 点（16 时）开始编译（周末下班回家直接下载最新固件开始折腾）。如需自定义则按照 cron 格式修改即可，GitHub Actions 的时区为 UTC，注意按照自己所在地时区进行转换。

## 点击 star 开始编译（已弃用）

▼ 点击查看

点击自己仓库页面上的 Star 按钮开始编译，为了防止产生垃圾记录，所以这个功能默认没有开启。

编辑 workflow 文件（.github/workflows/build-openwrt.yml）取消注释下面两行，后续点击自己仓库上的 star 即可开始编译。

```
# watch:  
# types: started
```

**TIPS:** 字段 started 并不是“开始了”的意思，而是“已经点击 Star”。

**吐槽:** 官方并没有提供一个开始按钮，通过搜索找到过很多奇怪的一键触发方式，但都是通过 Webhook 来实现的。机智的我发现了可以通过点击 Star 来触发，这样就相当于把 Star 当成开始按钮。这个 started 有种一句双关的意思了。

## macOS 虚拟机编译方案（已弃用）

► [点击查看](#)

## 写在最后

博主只是提供基本入门用法和思路，更高阶的玩法还需要小伙伴们自己去发觉。此外希望大家合理使用免费的服务器资源，必要时再编译。让出更多的服务器资源让开发者来充分利用才能产生更多更好的软件，这样大家才能受益。最后感谢 Microsoft 为我们免费提供 GitHub Actions 这样强大的服务。