



# MANUAL

\$ aaru-bluetooth --help

2022.9



9 797382 919846



# 目录

# [准备材料](#)

# [键码表](#)

# [安装电池：使用HAC-006电池](#)

# [安装电池：使用自备电池](#)

# [蓝牙管理：配对](#)

# [蓝牙管理：删除配对](#)

# [蓝牙管理：优先级设置](#)

# [蓝牙管理：电量报告](#)

# [蓝牙管理：充电](#)

# [蓝牙管理：休眠](#)

# [氛围灯](#)

# [大小写指示灯](#)

# [自定义：准备工作](#)

# [自定义：睡眠时间和全键无冲](#)

# [自定义：键码，通过手动修改\(适合高手\)](#)

# [自定义：键码，通过图形界面修改\(适合新手\)](#)

## AARU无线化指南

搭载蓝牙模块的PCB使AARU拥有无线连接体验，该PCB具有以下特征：

- 热插拔轴体固定方式，适配大门牙 (winkeyless) 或7U空格
- 延迟低至7.5ms的蓝牙5.2无线连接，或1ms的USB有线连接，并自动切换连接方式
- 蓝牙和USB有线支持全键无冲
- 低功耗，在关闭氛围灯时提供长达30日左右的续航<sup>1</sup>，打开氛围灯时提供1小时左右的续航
- 可记忆5组配对，支持各种主流的电脑或手机操作系统
- 具有ModelGauge技术的独立电量计跟踪精确的电池电量
- 通过Github Action自动生成固件，可自定义键码和休眠时间等设置



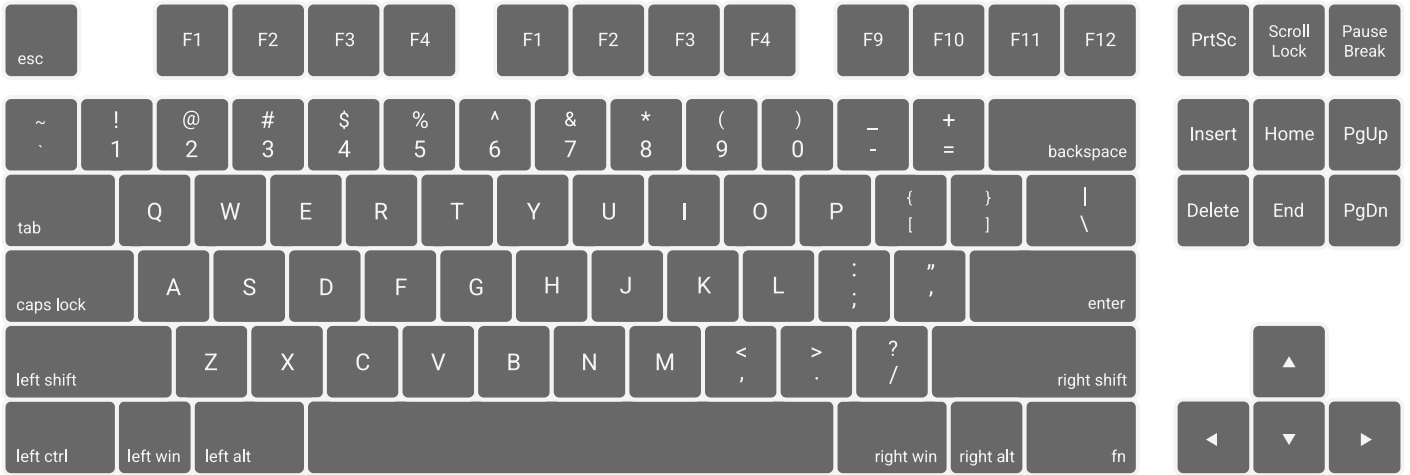
## 准备材料

- AARU
- AARU 蓝牙版本PCB
- 适配大门牙的1.5mm定位板
- HAC-006电池

<sup>1</sup> 测试条件：配对一个设备，关闭氛围灯，每日6h使用时间，从100%放电至0%所耗费的时间

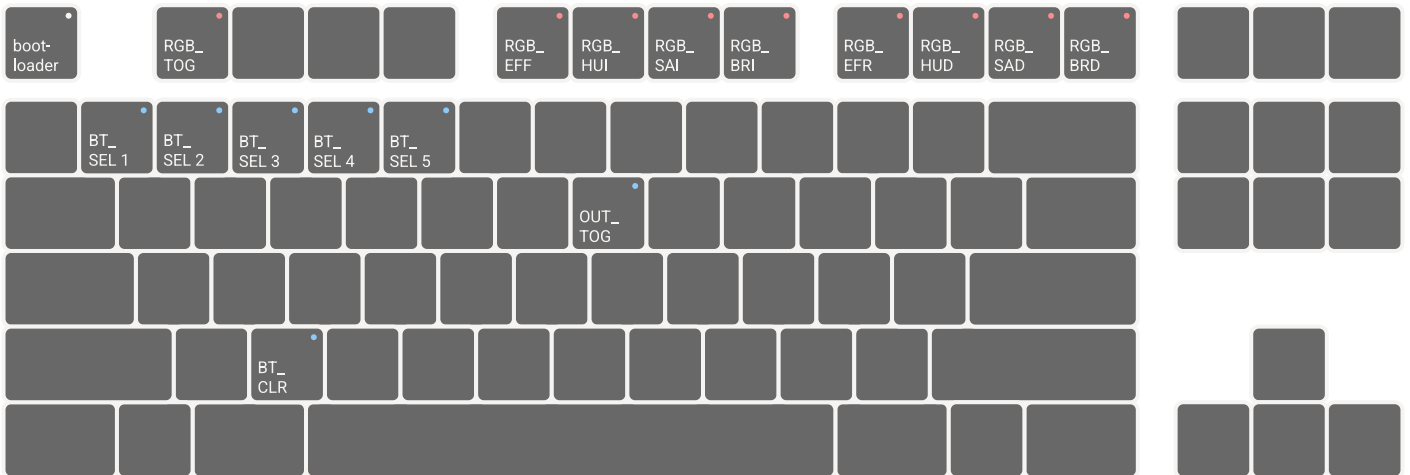
# 键码表

下面的图表绘制了AARU 蓝牙版本PCB支持的布局和出厂键值



第一层

👁️ 可按住'fn'切换至第二层



第二层

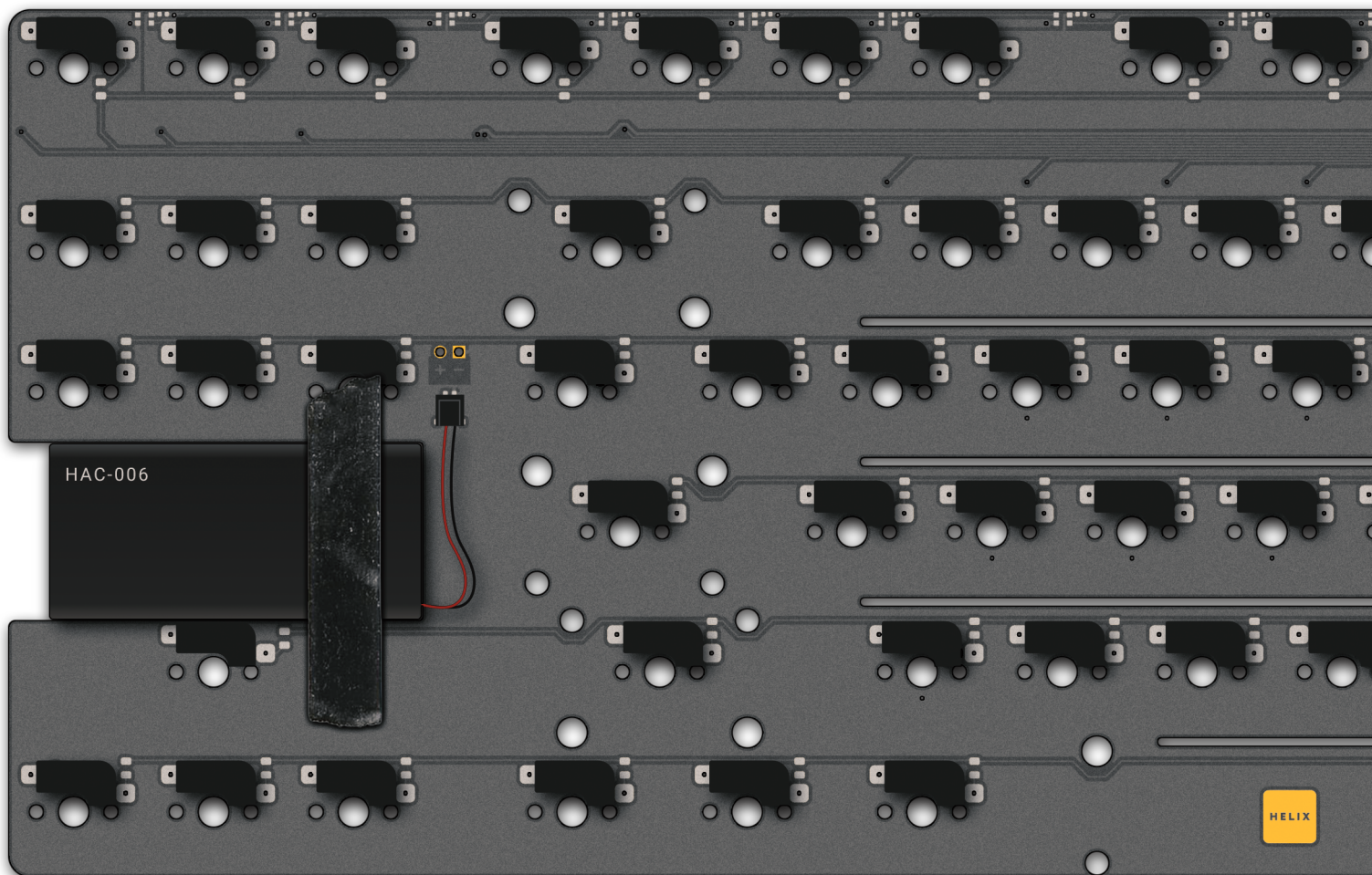
在第二层中，带有 ● 标识的键为灯光相关的控制键  
带有 ● 标识的键为蓝牙相关的控制键  
有关这些控制键的详细说明请参阅后面的小节

## 安装电池：使用HAC-006电池

**【第一步】** 从其他经销商处购买HAC-006电池

**【第二步】** 参考该图片，将电池固定在PCB上。先使用胶带固定电池的位置，再将电池的端子扣在PCB的电池座上

**【第三步】** 将PCB正面的电池开关推至上侧，使电池处于供电的状态。如不使用蓝牙功能，可推至下侧以停止电池供电



电池开关推至下侧时，电池供电会被切断，蓝牙模式将不可用，USB模式依然可用。

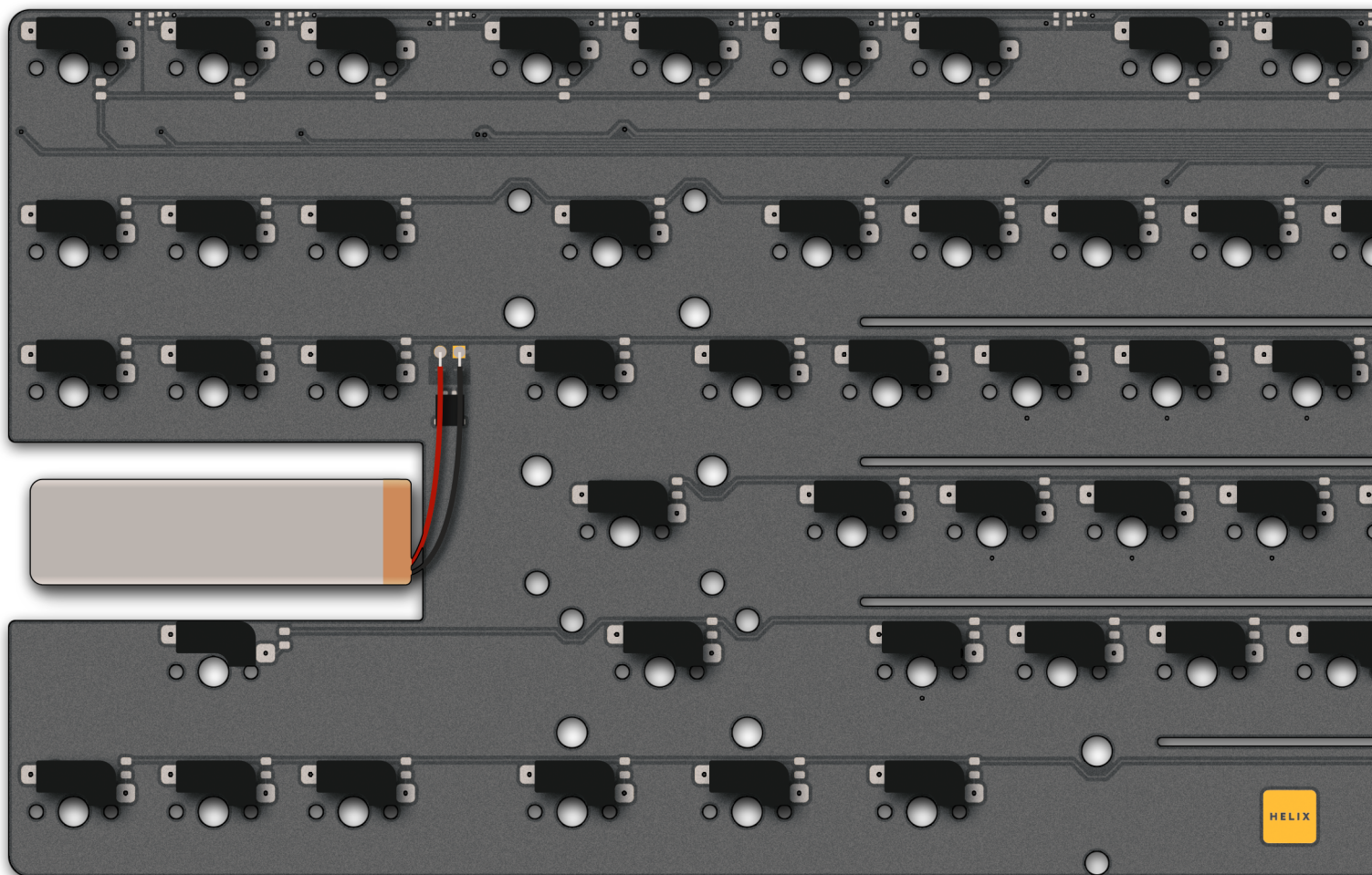
电池开关必须在插入USB数据线前打开，这样AARU才能正常读取到电池状态，否则电池电量将始终显示为100%。

## 安装电池：使用自备电池

【第一步】准备一块3.7V锂电池

【第二步】将电池固定后，把电池的电源线焊接在PCB的电池焊接位上，注意PCB上标示的正负极。

【第三步】将PCB正面的电池开关推至上侧，使电池处于供电的状态。如不使用蓝牙功能，可推至下侧以停止电池供电



电池开关推至下侧时，电池供电会被切断，蓝牙模式将不可用，USB模式依然可用。

电池开关必须在插入USB数据线前打开，这样AARU才能正常读取到电池状态，否则电池电量将始终显示为100%。

## 蓝牙管理：配对

### 将AARU与主机配对：

要通过蓝牙使用AARU，必须先使AARU和主机配对，AARU可以存储5个配对，意味着你可以配对5台不同的主机，然后通过按fn+1~5来在多个不同的主机间切换<sup>1</sup>。

通常情况下，fn+1应配对你最常用的设备，然后以此类推。注意你所配对的主机数越少，AARU的功耗就越低，因此建议不要和那些不常用的设备配对。

配对时不要连接USB数据线，以免AARU自动切换至USB模式。

AARU兼容蓝牙4.0，与具备蓝牙5.0以上的主机配对时可发挥出最佳的性能。



### 配对步骤：

**【第一步】** 准备一台即将配对的主机，可以是电脑、手机等，并确认已打开主机上的蓝牙功能

**【第二步】** 按下AARU上的fn+1或fn+2或fn+3或fn+4或fn+5来选择存储该配对的位置

**【第三步】** 在主机的蓝牙连接界面中搜索名为“AARU”的设备，并连接

### 选中一个配对：

选中一个配对是为了让AARU向该配对中的主机发送按键数据。

要在已经配对的设备中选中一个配对，请按相应的fn+数字键。比如，你曾将手机配对至fn+4，那么应按fn+4来选中该配对。

<sup>1</sup> 和5台主机配对并不意味着按键可以同时被5台主机收到，只有你当前选中的一个配对会收到按键数据



## 蓝牙管理：删除配对

### 删除一个配对：

要删除一个配对，首先需要选中它，然后按fn+x。比如，你将手机配对至fn+4，但现在你想要删除这个配对，则需要先按下fn+4，然后按fn+x。如果要重新配对该手机，则需要手机的蓝牙设置里也删除“AARU”，然后执行上页中的配对步骤。

## 蓝牙管理：优先级设置

AARU具有自动切换USB和蓝牙模式的功能，并且默认优先级为USB优先。当USB连接与蓝牙连接都与主机建立时，AARU会优先选择USB连接。

优先级可以通过按fn+u在USB优先和蓝牙优先中切换。

尤其是当你把AARU插在非电脑端口的充电器上充电时，优先级设置十分有用，因为AARU会误把充电器识别为电脑而试图采取USB连接。

## 蓝牙管理：电量报告

大部分现代设备都支持蓝牙设备的电量显示功能。

Windows在大部分情况下都能正常报告电池电量。

macos从12.0开始支持显示蓝牙设备的电量。

Android系统在大部分情况下都能正常报告电池电量

IOS可从小组件中的控件查看蓝牙设备的电量。

### 如果电量始终显示为100%：

原因一，你可能未插入电池或未将电池开关打开（电池开关推至上侧是打开）。

原因二，正确的上电步骤是在连接USB数据线之前连接电池并打开开关，这样电量计才能正常读取电池数据。如果已经连接电池并打开开关，且发现电量始终显示为100%，那么请先拔出USB数据线（如果有插入的话），然后关闭开关再打开即可。

## 蓝牙管理：充电

给AARU插上USB数据线，即可为AARU充电。充电端口可以是电脑的USB端口、手机充电适配器等。

请注意在充电时，由于连接了USB数据线，因此AARU会自动切换到USB模式，你所连接的其它蓝牙设备会因此而收不到按键数据，如有在充电时有蓝牙连接的需要，可按fn+u手动切换连接为蓝牙模式。

将一块HAC-006电池充满大约需要2小时。

通常你无需将电池充满，将其保持在80%以下的水平对维护电池的健康有所帮助。在有需要的情况下可将其充满。如果在充电时发现已经充至100%，拔出充电线后电量马上掉了2-3%，属于正常情况。

当电池电量降低到1%时，AARU仍可以维持数日的使用，但长期位于低电量可能会影响电池的健康。

## 蓝牙管理：休眠

在键盘无活动的40分钟后，AARU会进入休眠以降低功耗，按任意键即可唤醒，唤醒时间需要大约3秒。休眠时间可修改，请参考后面的章节。

## 氛围灯

AARU配备背部条形氛围灯，灯光特效和HSB等参数可直接由按键设置。AARU会自动记忆你设置的参数更改。

在使用电池供电的蓝牙模式下，打开灯光会迅速消耗大量的电池电量。  
将灯光调成白色，功耗将进一步增加，因为白光由三色合成的。

灯效：呼吸、色相循环、彩虹、电量、单色。

### 按键说明：



这些按键的具体位置请参阅第2页。

下面按从左到右的顺序解释每个按键作用：

**RGB\_TOG**

灯光开关

**RGB\_EFF**

切换到下一个灯光特效

**RGB\_HUI**

灯光色相增加

**RGB\_SAI**

灯光饱和度增加

**RGB\_BRI**

灯光亮度增加

**RGB\_EFR**

切换到上一个灯光特效

**RGB\_HUD**

灯光色相减少

**RGB\_SAD**

灯光饱和度减少

**RGB\_BRD**


灯光亮度减少

## 大小写指示灯

AARU在Caps lock位置拥有一个指示大小写状态的指示灯。指示灯默认为蓝色，可通过用电烙铁连接焊盘的方法修改为红色或绿色。

\*目前，在Windows系统下使用蓝牙连接时指示灯无法工作。

### 修改颜色：

【第一步】在PCB的背面，指示灯旁找到焊盘 

【第二步】用电烙铁清除掉原有的锡，然后按下图重新设置颜色



连接左手边的焊盘，把指示灯设为蓝色



连接中间的焊盘，把指示灯设为红色



连接右手边的焊盘，把指示灯设为绿色

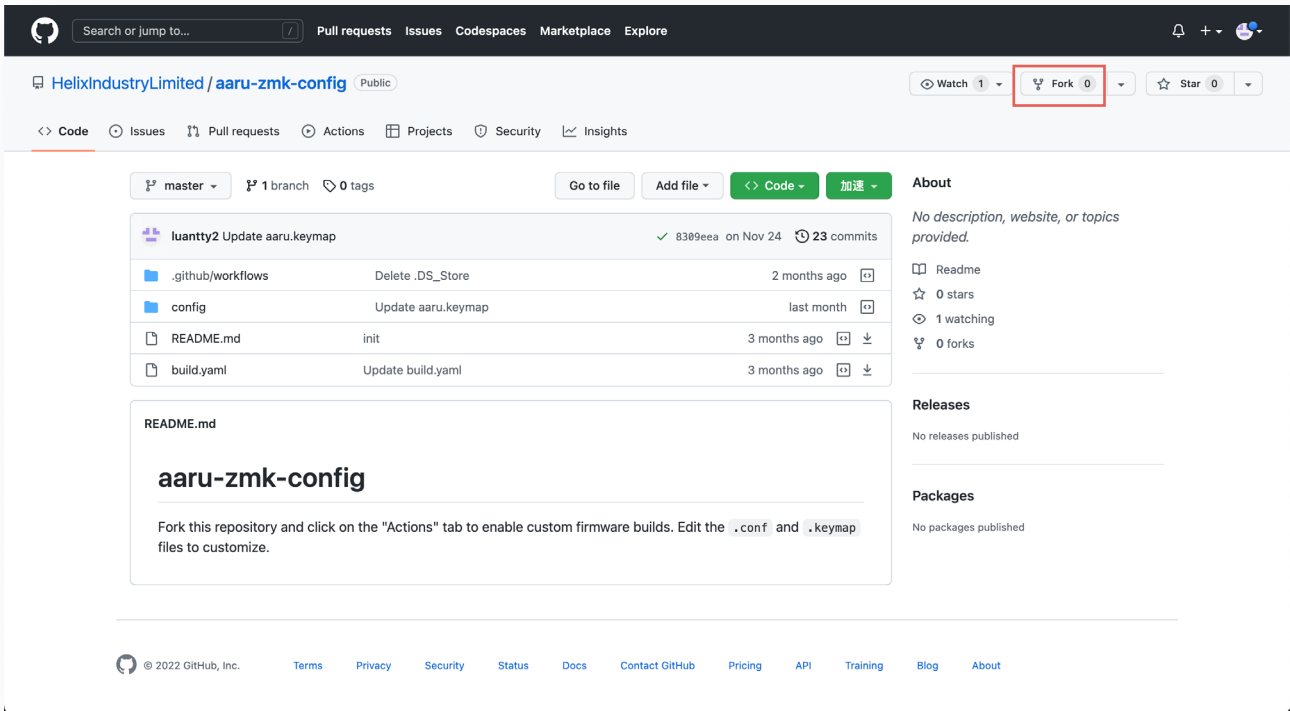


不连接任何焊接，指示灯不会亮起

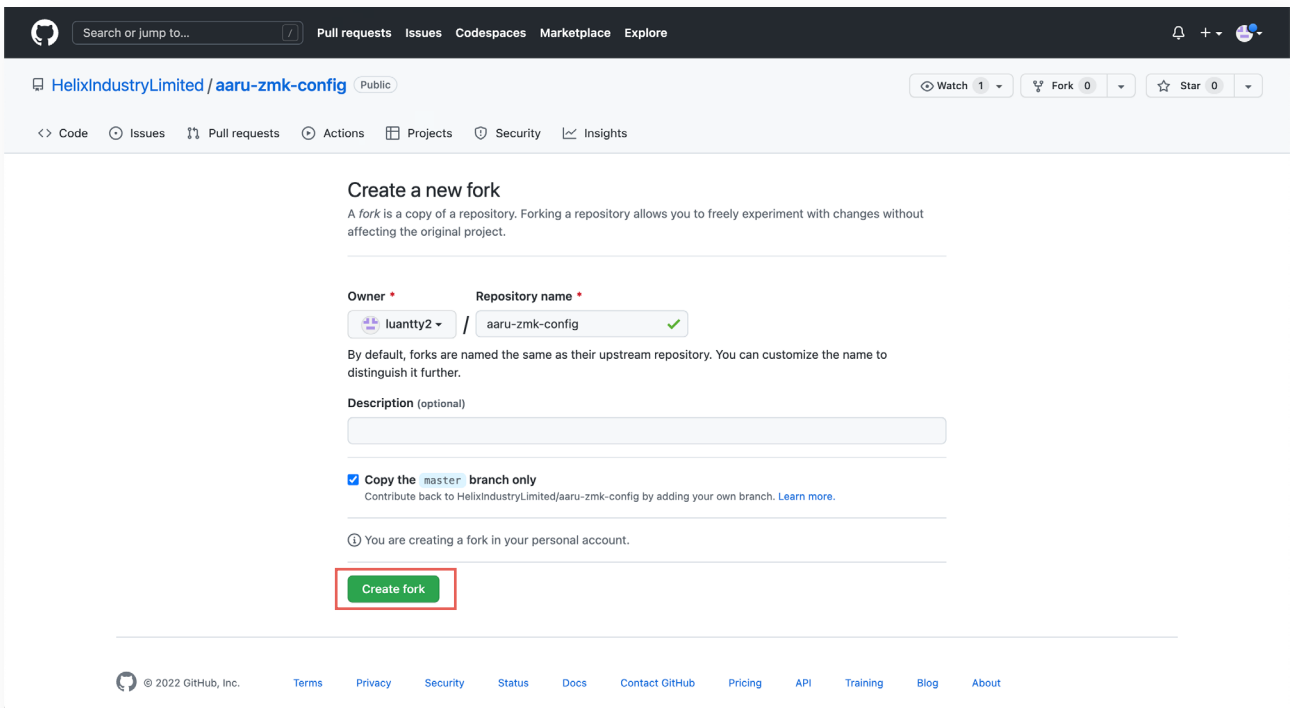
## 自定义：准备工作

【第一步】你需要一个Github账号来完成后面的操作，如果你没有Github账号，请登录github.com点击Sign up注册。

【第二步】浏览器打开github.com/HelixIndustryLimited/aaru-zmk-config，然后点击右上角的Fork。

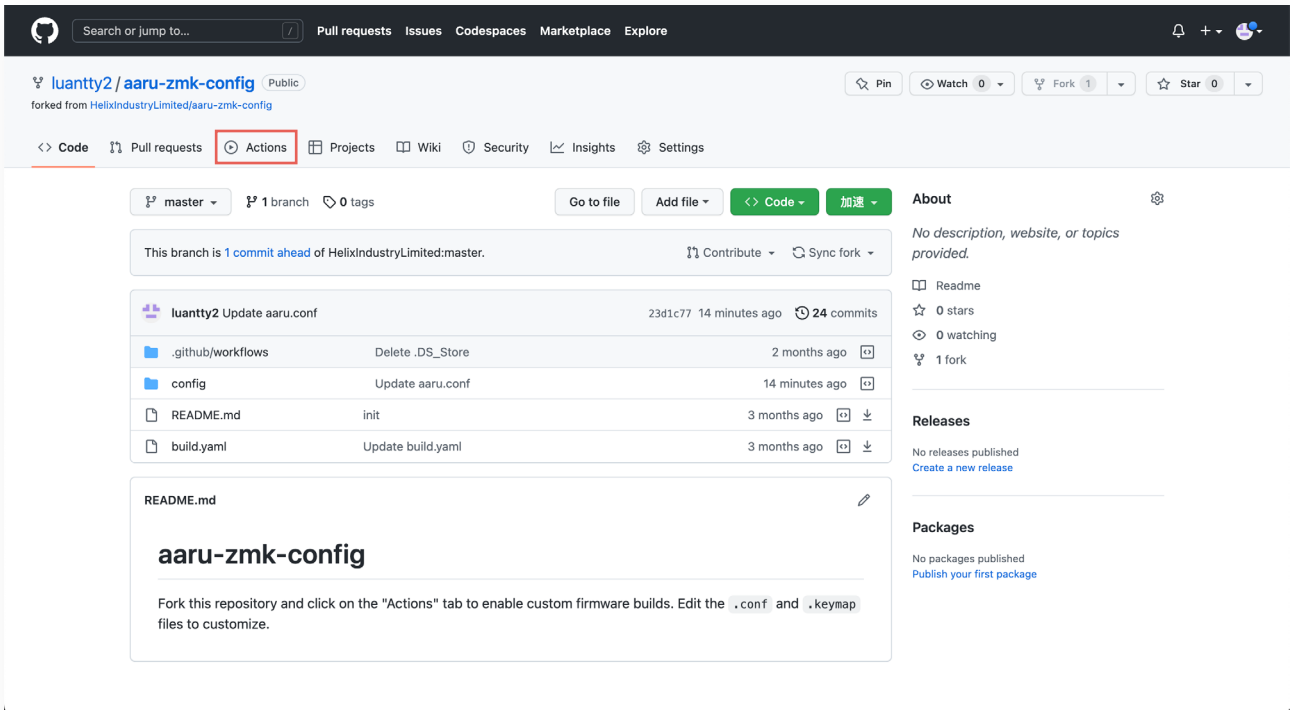


在随后的页面中点击Create fork。

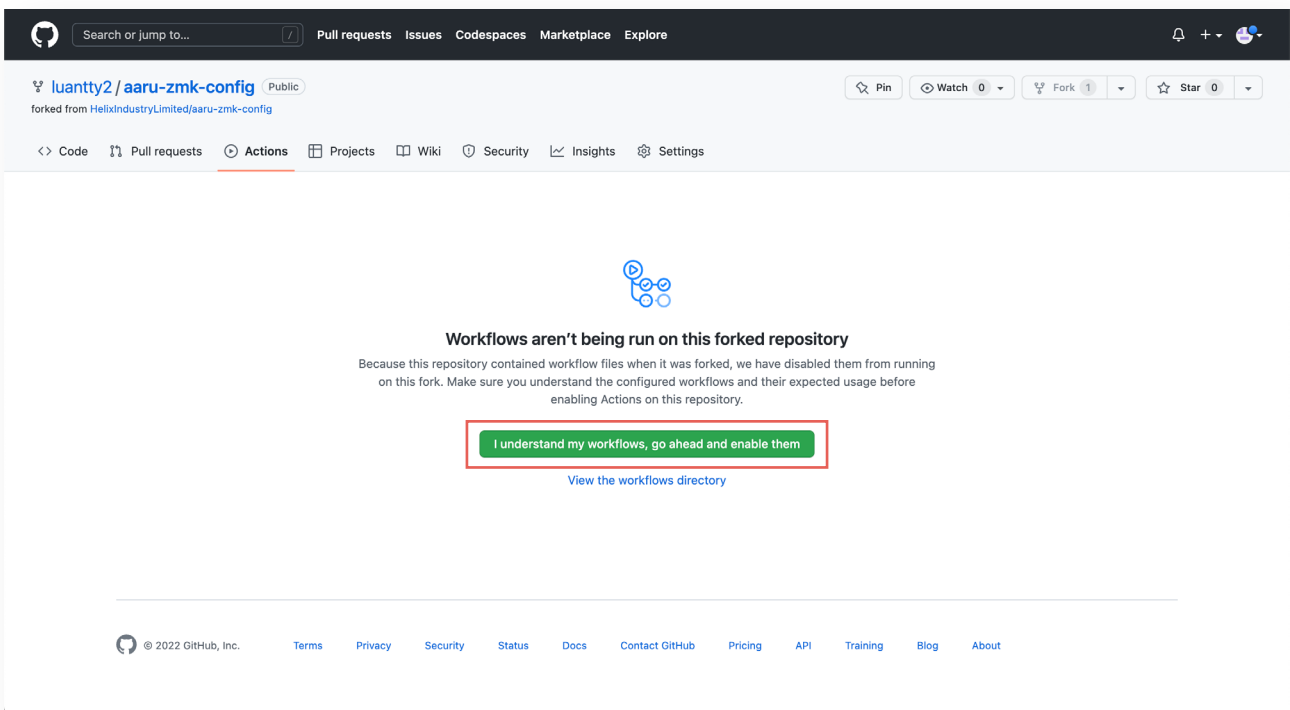


【第三步】浏览器将跳转到你所Fork的aaru-zmk-config存储库，域名为github.com/你的用户名/aaru-zmk-config。将来也可以在Github右上角点击你的头像，再点击Your repositories中找到。

【第四步】点击Actions标签。



【第四步】在该界面中点击I understand...



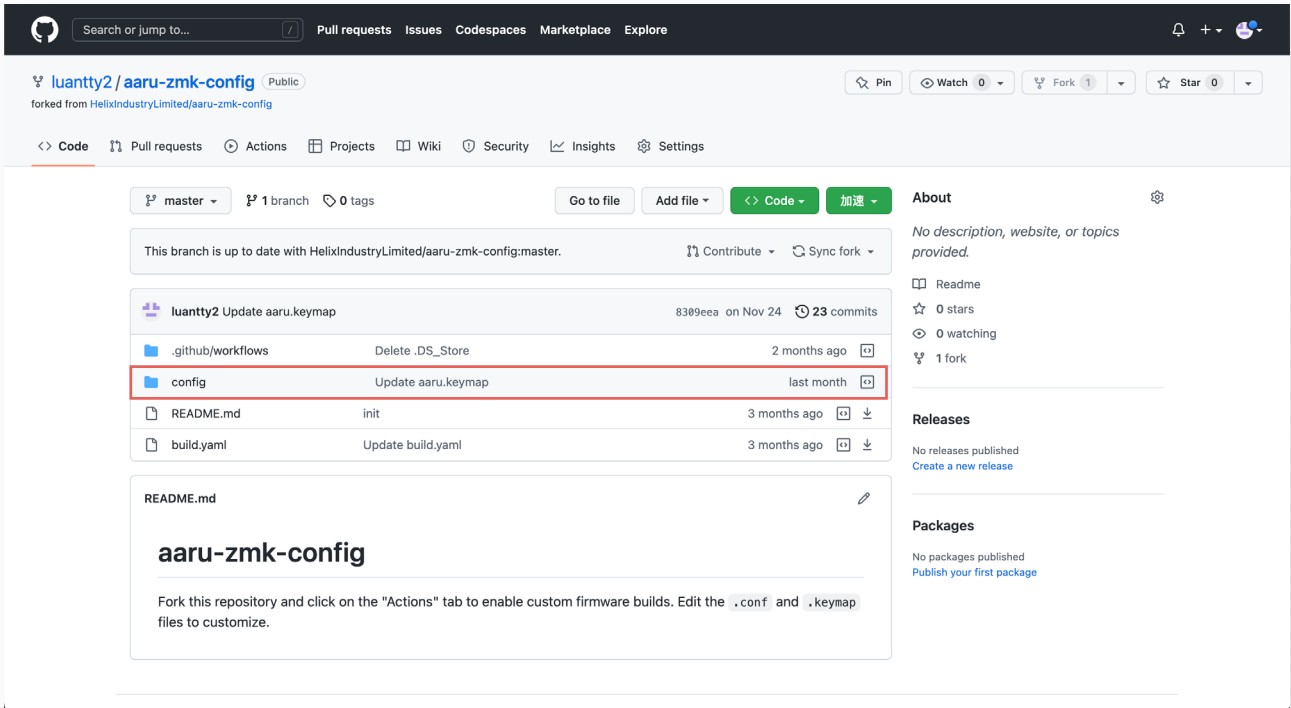
至此，你Fork了一个属于你的aaru-zmk-config存储库，你所有的自定义都将在这个存储库上进行。上述准备工作只需完成一次，未来无需再操作。

## 自定义：睡眠时间和全键无冲

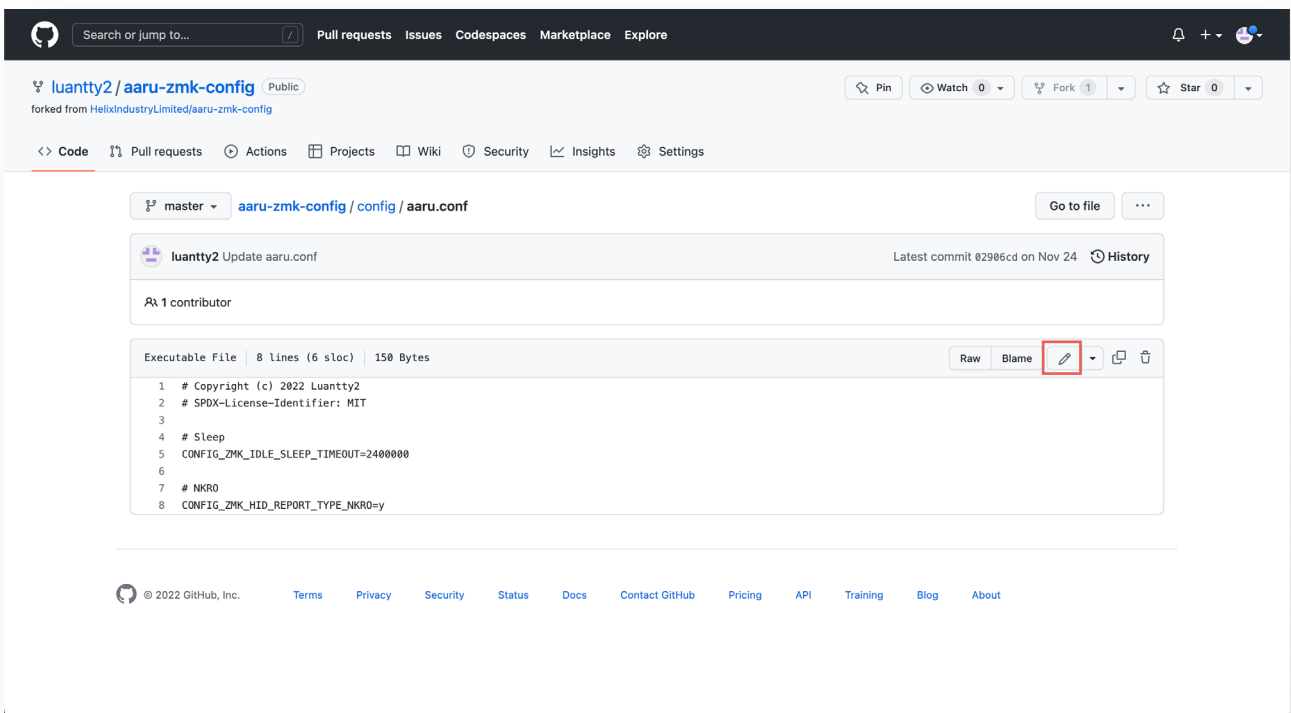
首先，确认你已经完成前篇“自定义：准备工作”中的内容。

【第一步】打开你所Fork的aaru-zmk-config存储库，域名为github.com/你的用户名/aaru-zmk-config。也可以在Github右上角点击你的头像，再点击Your repositories中找到。

【第二步】点击config文件夹，再点击aaru.conf文件



【第三步】点击笔形图标进行编辑

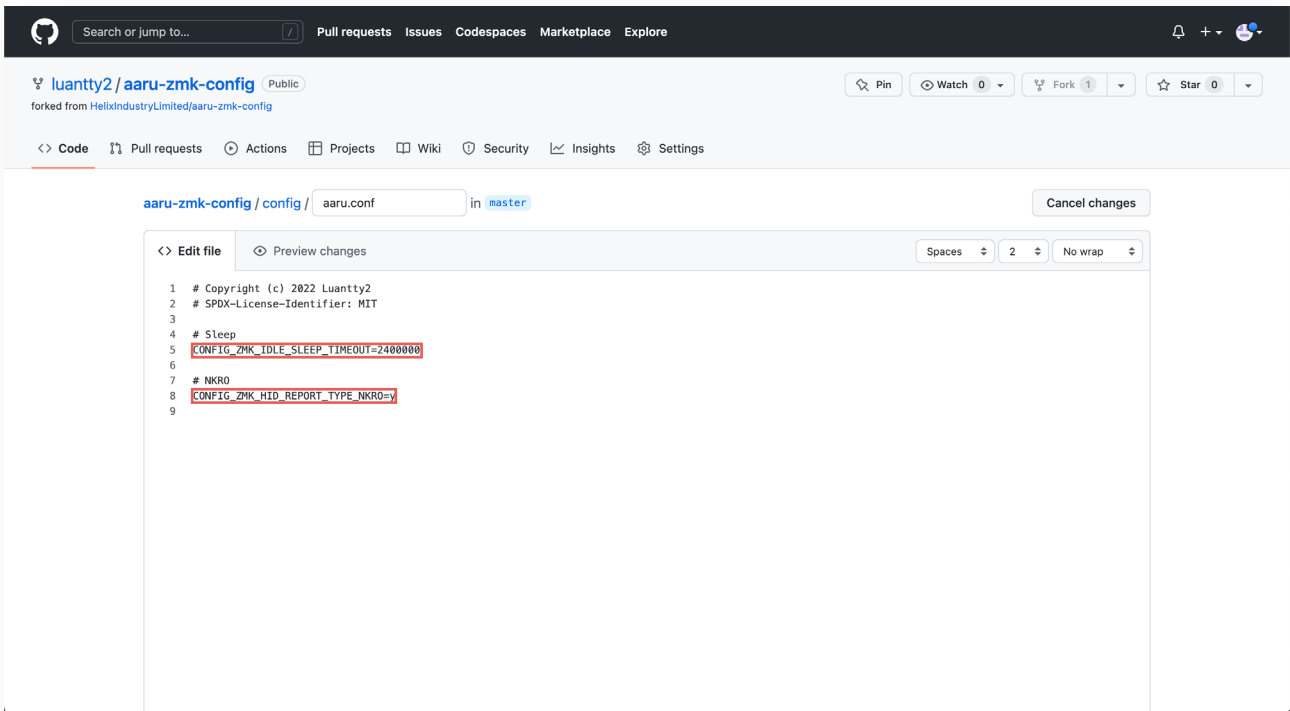


【第四步】对数值进行修改。

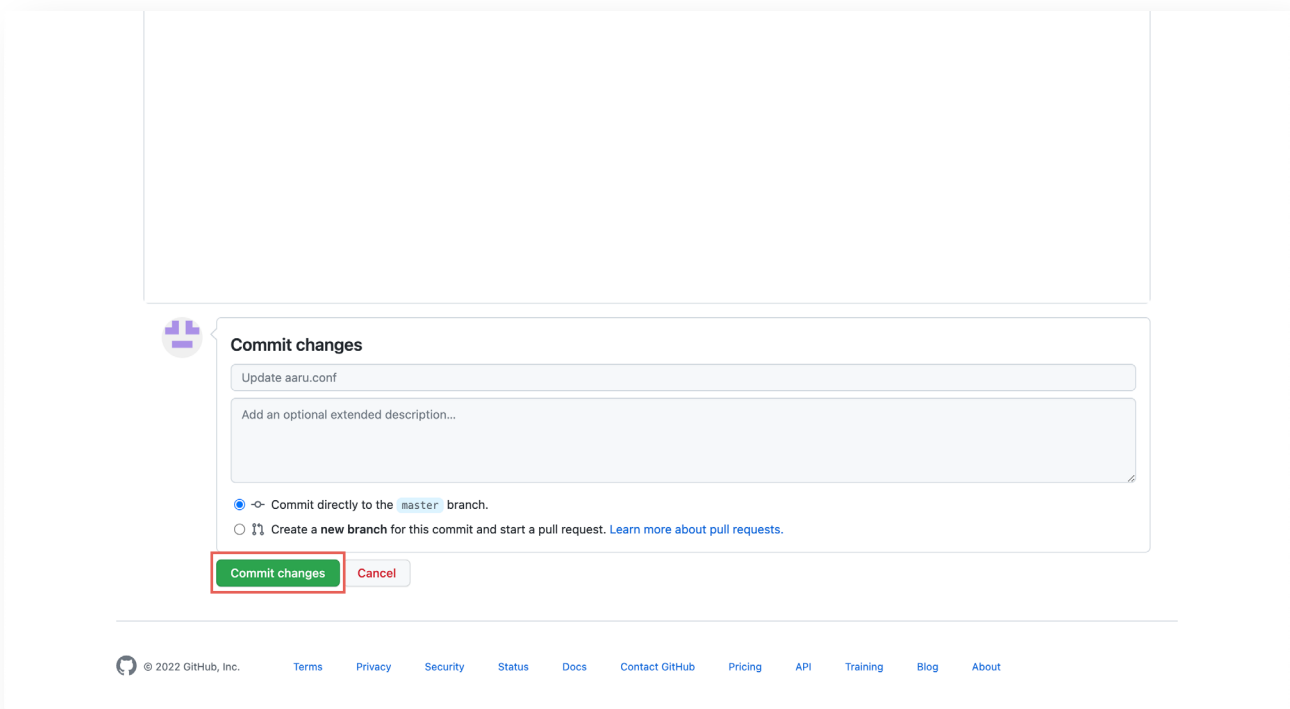
CONFIG\_ZMK\_IDLE\_SLEEP\_TIMEOUT=2400000是睡眠时间设置，以毫秒为单位，默认为2400000，即40分钟。

如果你想修改为3小时，则应修改为10800000。

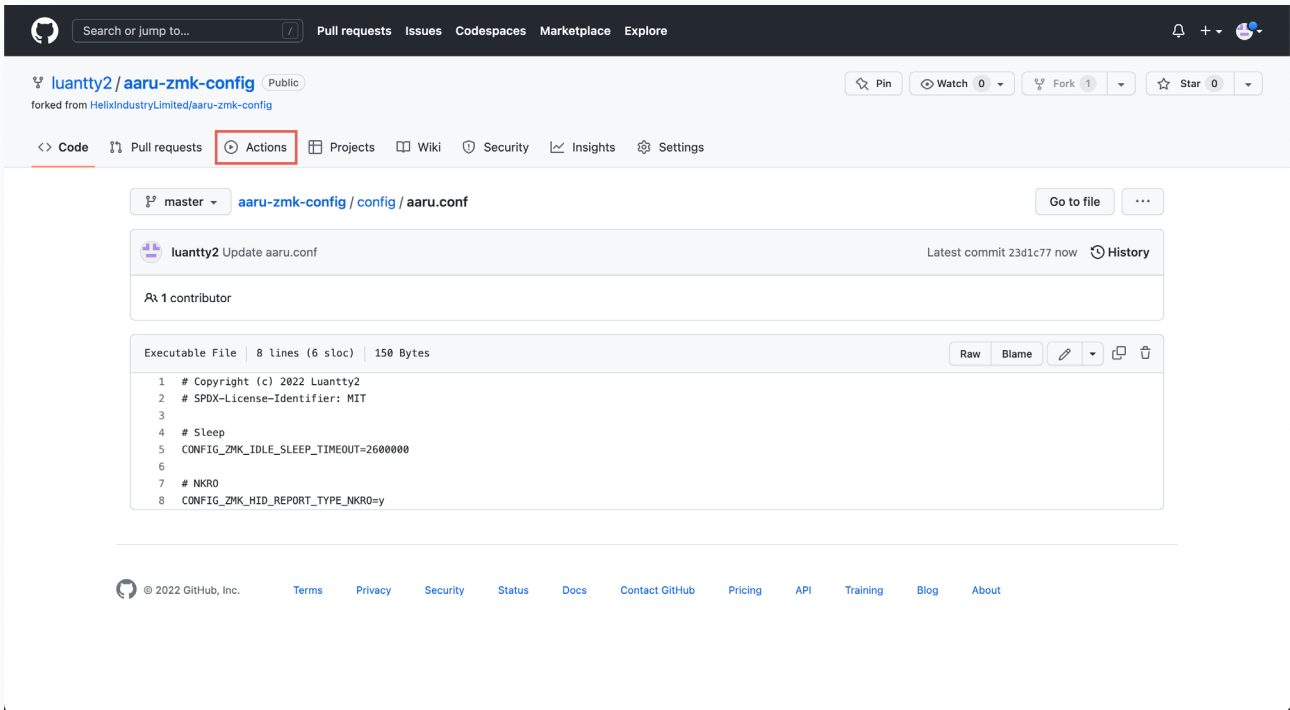
CONFIG\_ZMK\_HID\_REPORT\_TYPE\_NKRO=y是全键无冲设置，默认为y，即全键无冲启用，若想使用六键无冲，请修改y为n。



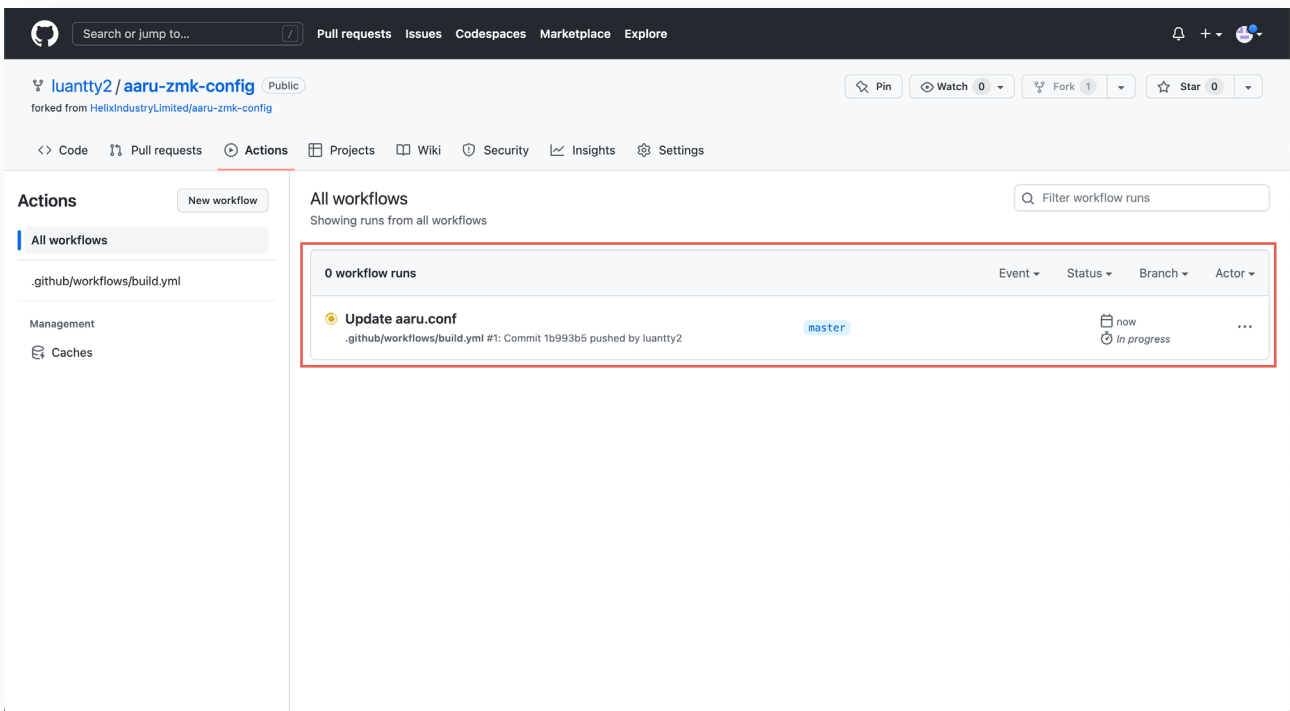
【第五步】修改完成后，向下拉动页面，点击Commit changes保存你的修改。



【第六步】点击Actions标签。

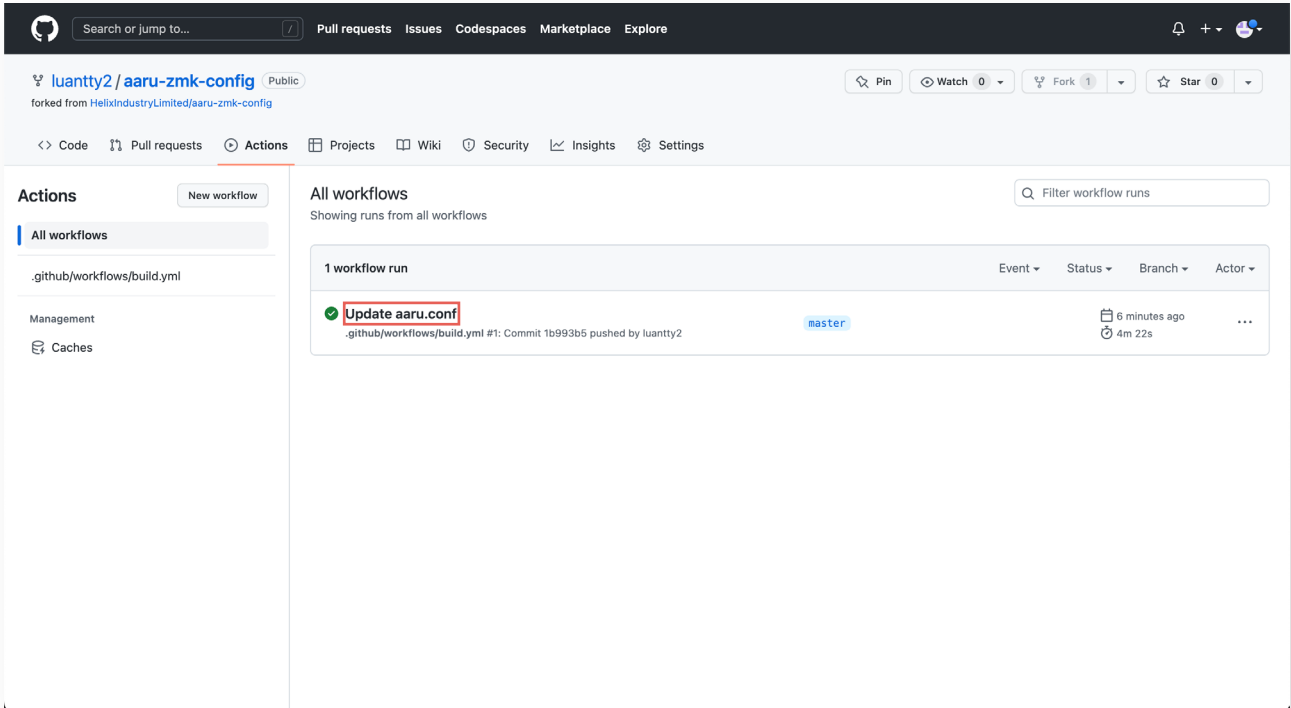


【第七步】你修改了东西，因此可以看到一个自动编译正在运行，请等待3-5分钟，黄色的状态小圆点变为绿色的勾后，提示编译完成。

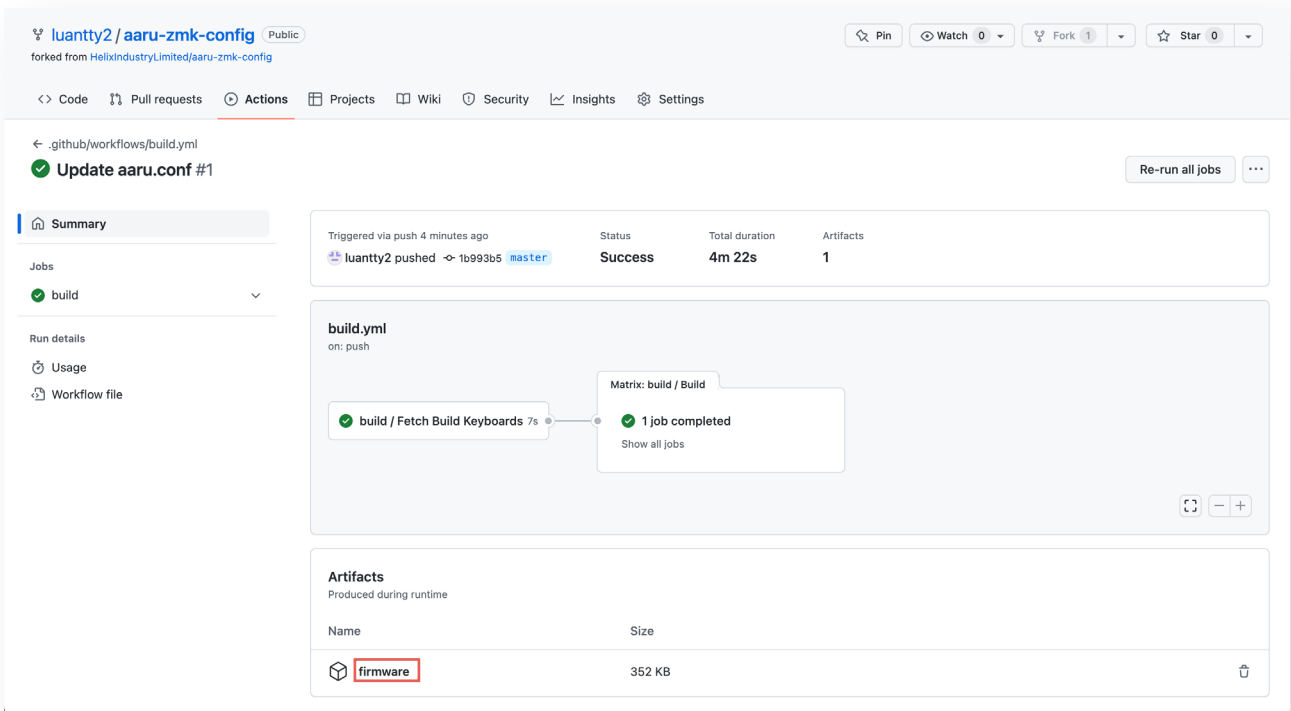




【第八步】在状态从黄色小圆点变为绿色的勾后，提示编译完成，点击标题下载编译完的固件。



【第九步】在此页面中的点击firmware下载固件压缩包，解压后得到aaru-zmk.uf2。



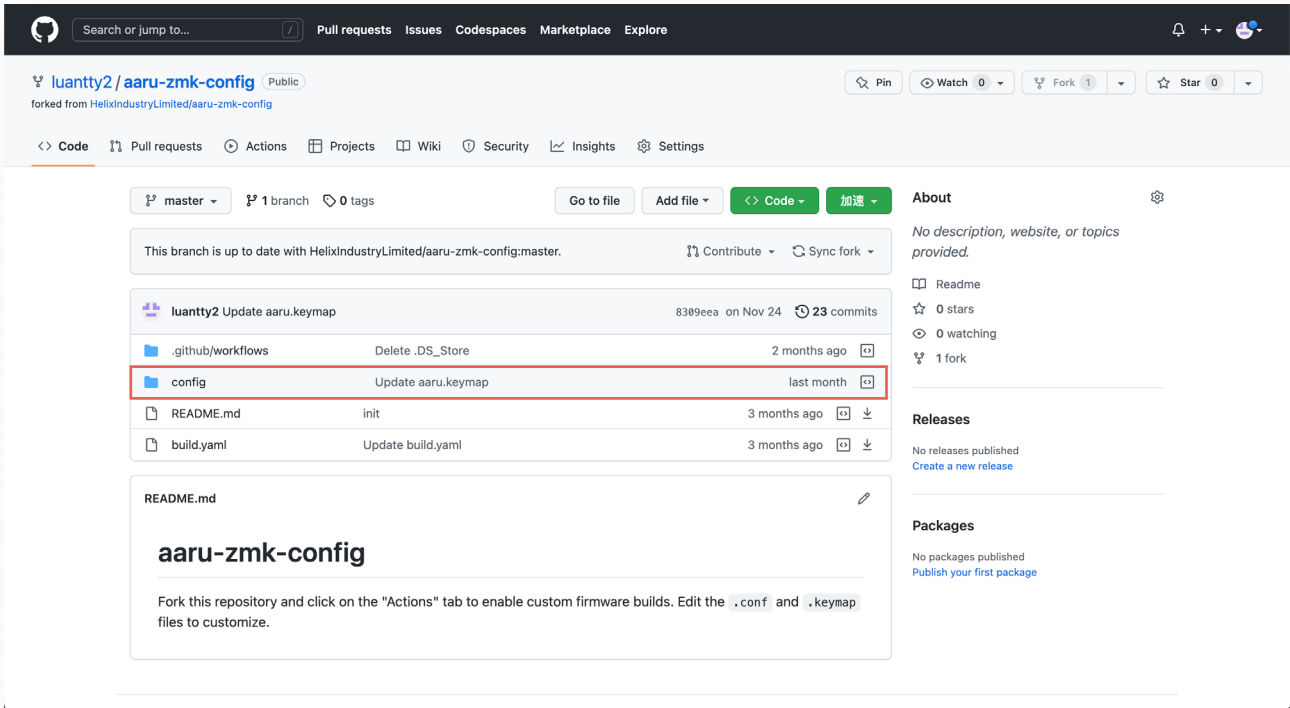
**【第十步】**把AARU用USB数据线连接至电脑，按fn+esc进入刷机模式，你的电脑上将会弹出一个名为NRF52BOOT的U盘，将aaru-zmk.uf2拖入次U盘后，U盘会自动弹出，固件烧录完成。  
新的固件烧录后，请检查你修改的功能是否实现。

## 自定义：键码，通过手动修改(适合高手)

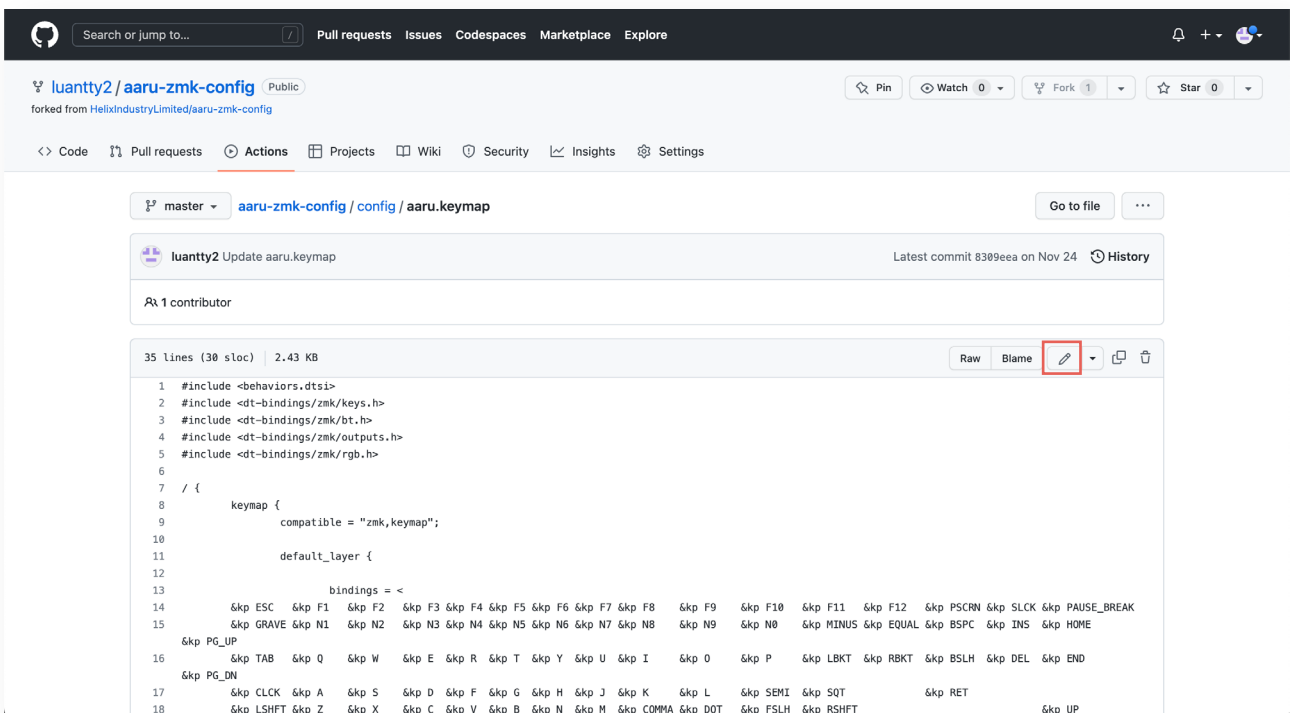
首先，确认你已经完成前篇“自定义：准备工作”中的内容。

【第一步】打开你所Fork的aaru-zmk-config存储库，域名为github.com/你的用户名/aaru-zmk-config。也可以在Github右上角点击你的头像，再点击Your repositories中找到。

【第二步】点击config文件夹，再点击aaru.keymap文件



【第三步】点击笔形图标进行编辑



【第四步】对键码进行修改。

键码为两层，default\_layer为第一层，fn\_layer为第二层。

以default\_layer为例，可以看到：

```
&kp ESC &kp F1 &kp F2 &kp F3 &kp F4 &kp F5 &kp F6 &kp F7 &kp F8 &kp F9 &kp F10 &kp F11 &kp F12 &kp PSCRN &kp SLCK &kp PAUSE_BREAK
&kp GRAVE &kp N1 &kp N2 &kp N3 &kp N4 &kp N5 &kp N6 &kp N7 &kp N8 &kp N9 &kp N0 &kp MINUS &kp EQUAL &kp BSPC &kp INS &kp HOME &kp PG_UP
&kp TAB &kp Q &kp W &kp E &kp R &kp T &kp Y &kp U &kp I &kp O &kp P &kp LBKT &kp RBKT &kp BSLH &kp DEL &kp END &kp PG_DN
&kp CLCK &kp A &kp S &kp D &kp F &kp G &kp H &kp J &kp K &kp L &kp SEMI &kp SQT &kp RET
&kp LSHFT &kp Z &kp X &kp C &kp V &kp B &kp N &kp M &kp COMMA &kp DOT &kp FSLH &kp RSHFT &kp UP
&kp LCTRL &kp LGUI &kp LALT &kp SPACE &kp RALT &kp RGUI &mo 1 &kp LEFT &kp DOWN &kp RIGHT
```

以最左上角的esc键为例，它的键码为 &kp ESC，如果将它修改为&kp UP，它将变成上方向键。所有可以修改的键码可以在ZMK文档上找到。

常规键码必须以&kp开头，在修改键码时，不要将它删除，否则编译会失败。

有一些特殊的键例如 &mo 1，它的意思是按住后跳到第二层(即fn层)，其位置可被修改。

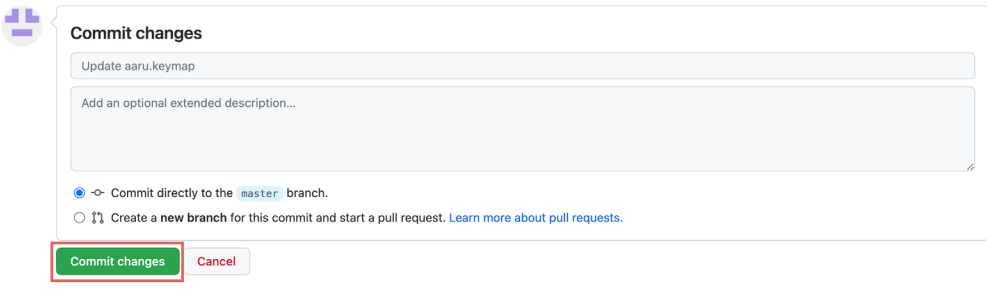
与蓝牙相关的键是以&bt开头的，这些键码位于fn\_layer，其位置可被修改。

与RGB相关的键是以&rgb\_ug开头的，这些键码位于fn\_layer，其位置可被修改。

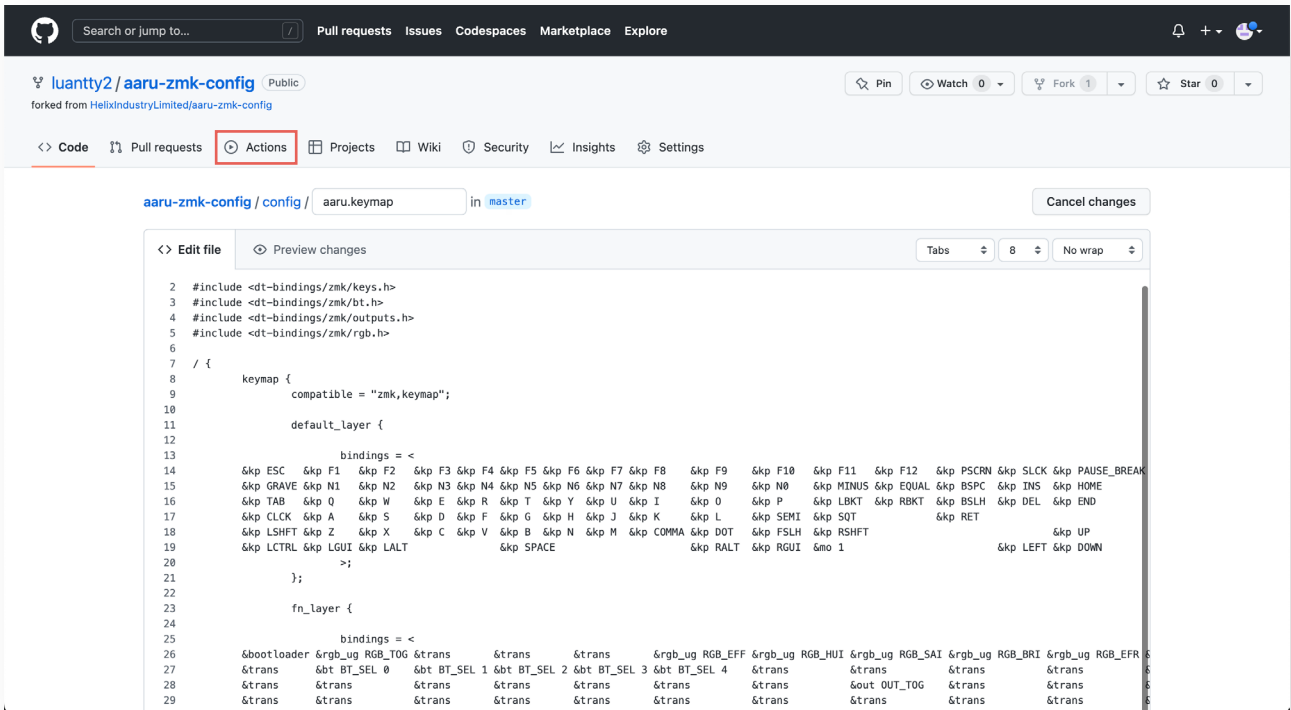
fn\_layer还有大量的&trans键，它的意思是这个键的键码沿用default\_layer的键码。

【第五步】修改完成后，向下拉动页面，点击Commit changes保存你的修改。

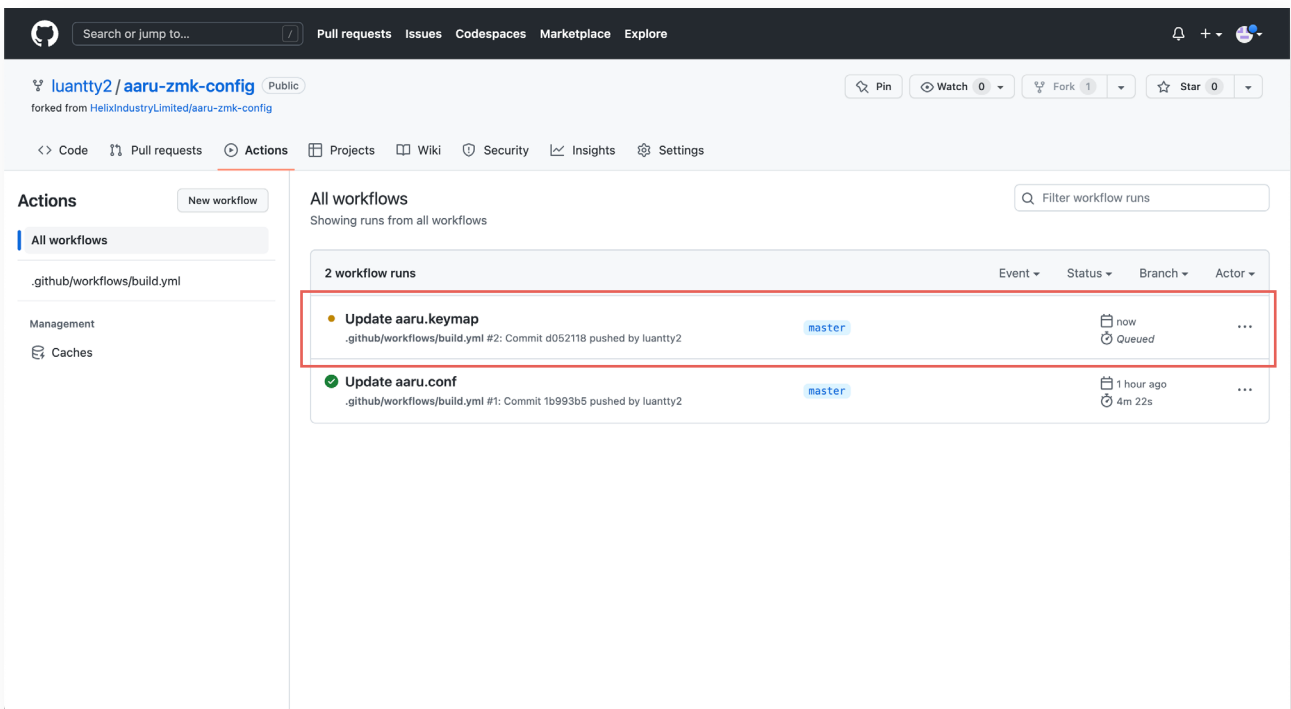
```
12
13     bindings = <
14         &kp ESC &kp F1 &kp F2 &kp F3 &kp F4 &kp F5 &kp F6 &kp F7 &kp F8 &kp F9 &kp F10 &kp F11 &kp F12 &kp PSCRN &kp SLCK &kp PAUSE_BREAK
15         &kp GRAVE &kp N1 &kp N2 &kp N3 &kp N4 &kp N5 &kp N6 &kp N7 &kp N8 &kp N9 &kp N0 &kp MINUS &kp EQUAL &kp BSPC &kp INS &kp HOME
16         &kp TAB &kp Q &kp W &kp E &kp R &kp T &kp Y &kp U &kp I &kp O &kp P &kp LBKT &kp RBKT &kp BSLH &kp DEL &kp END
17         &kp CLCK &kp A &kp S &kp D &kp F &kp G &kp H &kp J &kp K &kp L &kp SEMI &kp SQT &kp RET
18         &kp LSHFT &kp Z &kp X &kp C &kp V &kp B &kp N &kp M &kp COMMA &kp DOT &kp FSLH &kp RSHFT &kp UP
19         &kp LCTRL &kp LGUI &kp LALT &kp SPACE &kp RALT &kp RGUI &mo 1 &kp LEFT &kp DOWN
20     >;
21 };
22
23     fn_layer {
24         bindings = <
25             &bootloader &rgb_ug RGB_TOG &trans &trans &trans &rgb_ug RGB_EFF &rgb_ug RGB_HUI &rgb_ug RGB_SAI &rgb_ug RGB_BRI &rgb_ug RGB_EFR
26             &trans &bt_BT_SEL_0 &bt_BT_SEL_1 &bt_BT_SEL_2 &bt_BT_SEL_3 &bt_BT_SEL_4 &trans &trans &trans &trans &trans &trans &trans
27             &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans
28             &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans
29             &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans
30             &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans
31             &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans &trans
32         >;
33     };
34 };
35 };
36
```

A dialog box titled "Commit changes" with a purple icon on the left. It contains a text input field for "Update aaru.keymap", a larger text area for "Add an optional extended description...", and two radio buttons. The first radio button is selected and labeled "Commit directly to the master branch.", the second is labeled "Create a new branch for this commit and start a pull request. Learn more about pull requests.". At the bottom, there are two buttons: "Commit changes" (highlighted with a red box) and "Cancel".

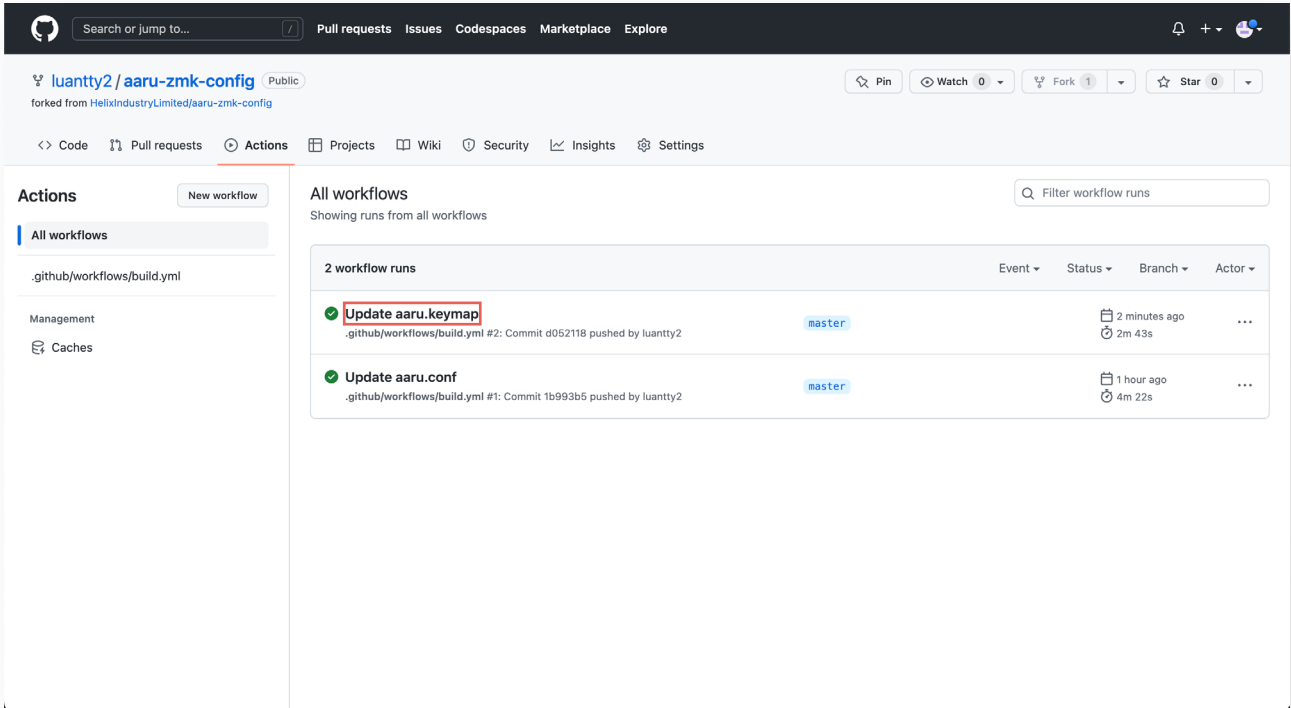
【第六步】点击Actions标签。



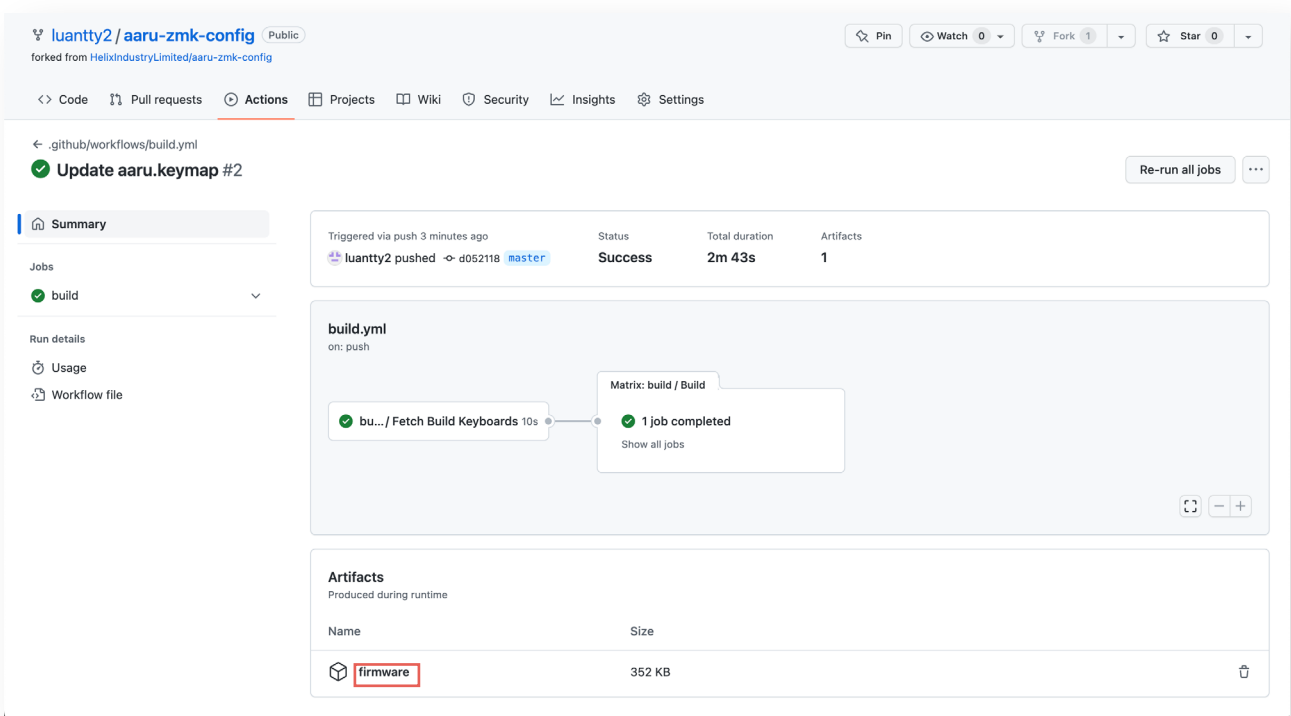
【第七步】你修改了东西，因此可以看到一个自动编译正在运行，请等待3-5分钟，黄色的状态小圆点变为绿色的勾后，提示编译完成。



【第八步】在状态从黄色小圆点变为绿色的勾后，提示编译完成，点击标题下载编译完的固件。



【第九步】在此页面中的点击firmware下载固件压缩包，解压后得到aaru-zmk.uf2。



**【第十步】**把AARU用USB数据线连接至电脑，按fn+esc进入刷机模式，你的电脑上将会弹出一个名为NRF52BOOT的U盘，将aaru-zmk.uf2拖入次U盘后，U盘会自动弹出，固件烧录完成。  
新的固件烧录后，请检查你修改的按键是否实现。

## 自定义：键码，通过图形界面修改(适合新手)

除了手动修改键码外，还可以使用图形界面来修改。

首先，确认你已经完成前篇“自定义：准备工作”中的内容。

【第一步】浏览器打开[nickcoutsos.github.io/keymap-editor](https://nickcoutsos.github.io/keymap-editor)。

【第二步】如下图所示，从左上角的下拉菜单中选择Github。



【第三步】在弹出的一系列窗口中点击按钮。

1

### Authenticate with GitHub

Please log-in to GitHub to continue.

Login with GitHub

Verify your GitHub identity (HelixIndustryLimited)

Know which resources you can access

Act on your behalf

Learn more

Learn more about ZMK Keymap Editor

3

### Welcome, HelixIndustryLimited!

To begin import your own keymaps, you'll need to grant the Keymap Editor app access to the relevant GitHub repositories. Please make sure to only select repositories that are set up for ZMK keyboard configurations.

This app requires read and write access to the repositories you select in order to write to arbitrary `.keymap` files. It also receives all build action events for the repository, which generates quite a bit of traffic. Selecting **All repositories** for the installation will lead to a suspension.

Add Repository

Authorize ZMK Keymap Editor

Authorizing will redirect to <https://zmk-keymap-editor.fly.dev>

Only select repositories

Select at least one repository. Also includes public repositories (read-only).

with these permissions:

- ✓ Read access to actions and metadata
- ✓ Read and write access to code

Install Cancel

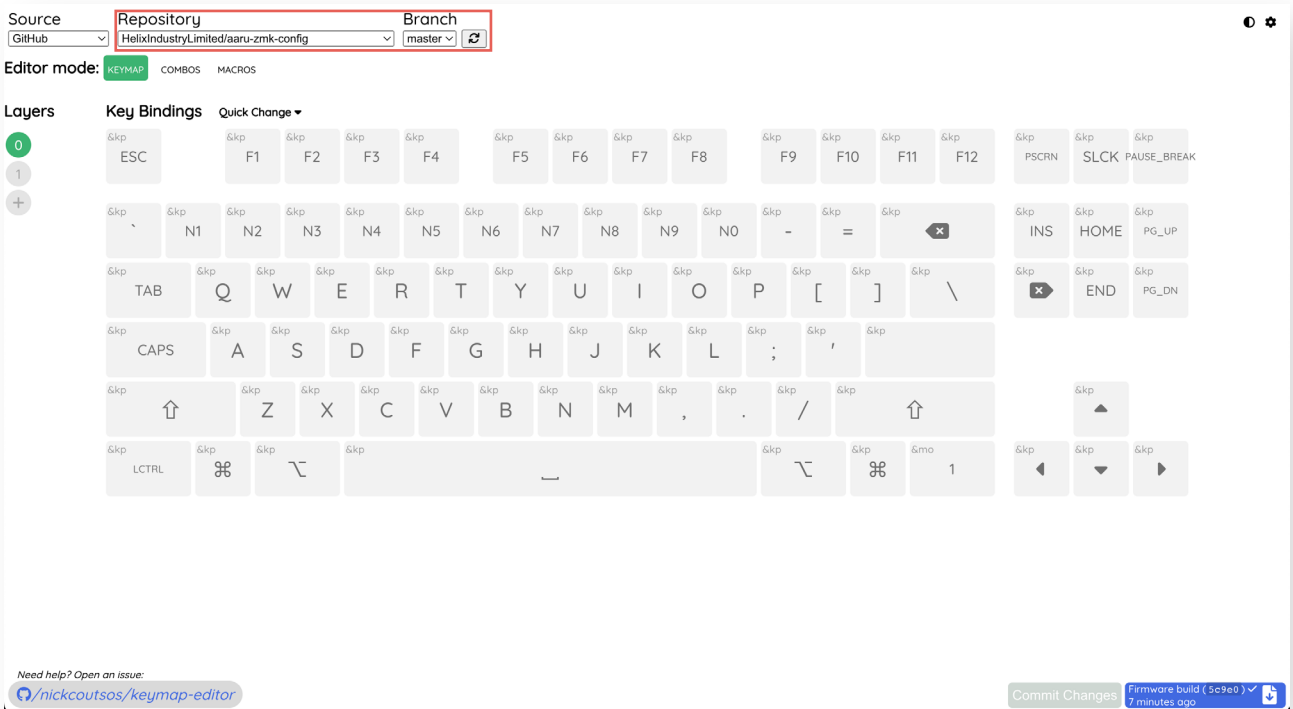
Next: you'll be directed to the GitHub App's site to complete setup.

2

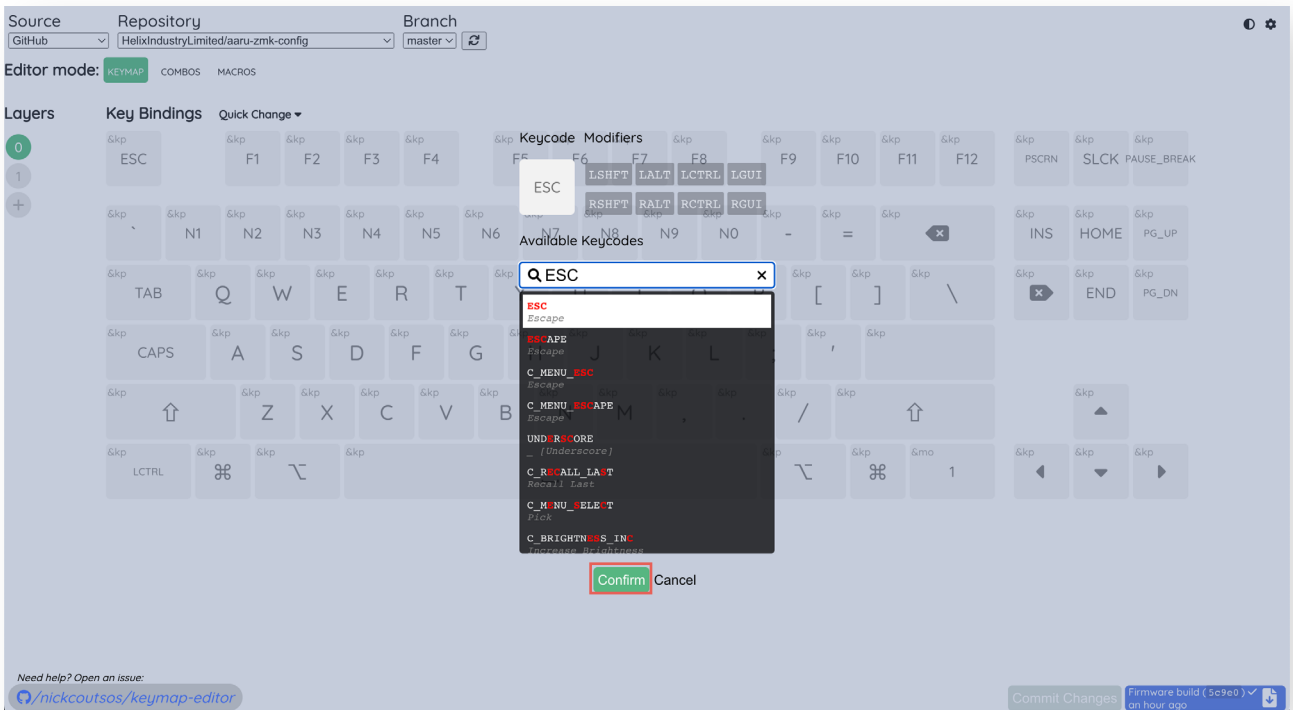
4



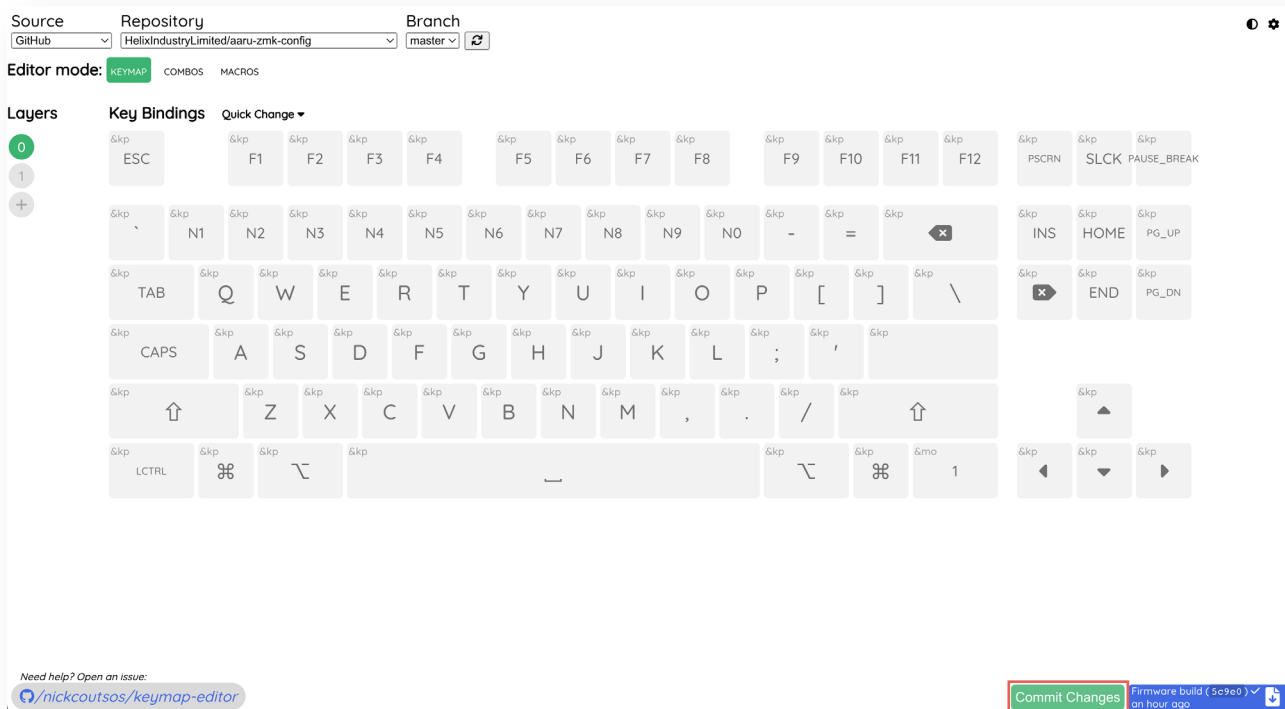
【第四步】如果顺利，你会来到改键的界面，如果没有看到该界面，请确认Repository选项下是否选择了“你的用户名/aaru-zmk-config”



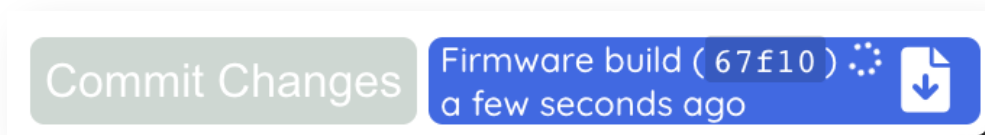
【第五步】点击任意键可查看该处的键值，你可以输入想要的键值来修改它，所有可以修改的键码可以在ZMK文档上找到。  
修改完成后，点击Confirm确认。



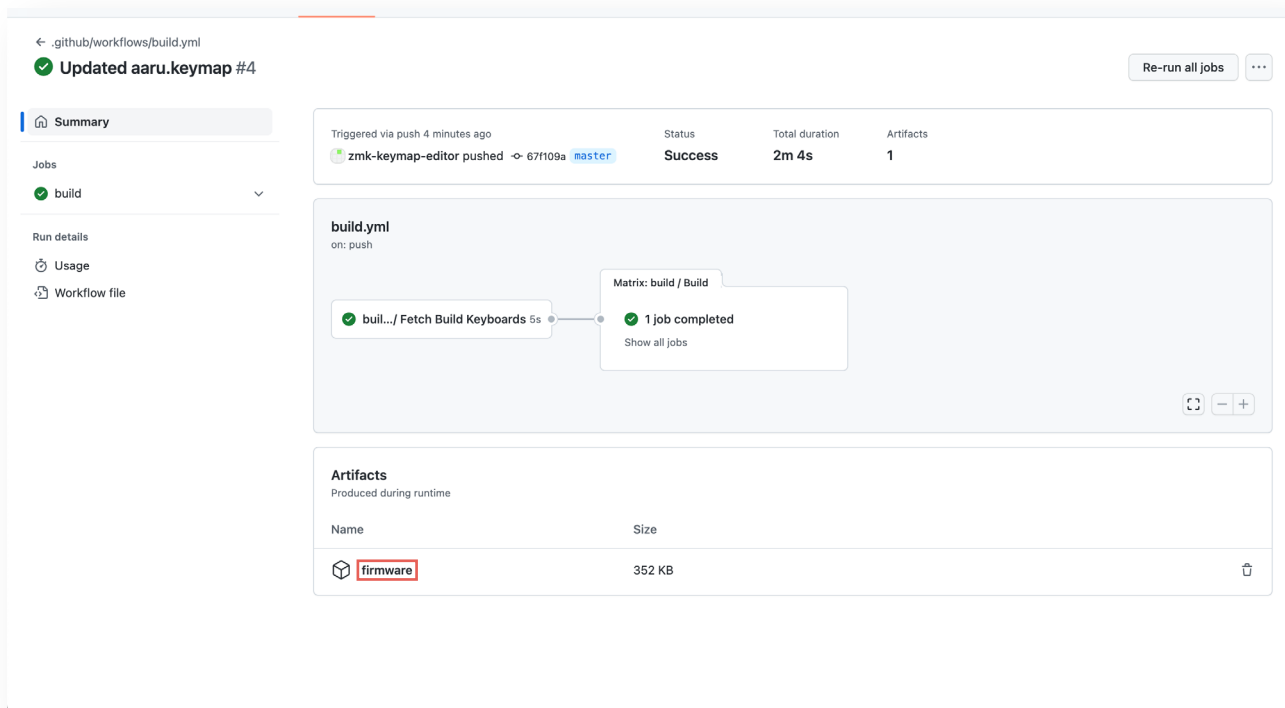
【第六步】所有的修改完成后，点击页面右下角的Commit Changes按钮以保存你的修改。并且在随后弹出的Commit keymap changes窗口中点击commit。



【第七步】在修改保存后，蓝色按钮此时开始出现转圈图形，意味着固件正在被构建，完成后，转圈图形将变成“勾”，提示构建完成。构建大约需要1-5分钟，如果等待很久后依然在转圈，请刷新页面。



【第八步】蓝色按钮出现“勾”后，点击它，打开一个新的页面，点击firmware下载建构好的固件。



**【第九步】**把AARU用USB数据线连接至电脑，按fn+esc进入刷机模式，你的电脑上将会弹出一个名为NRF52BOOT的U盘，将aaru-zmk.uf2拖入次U盘后，U盘会自动弹出，固件烧录完成。  
新的固件烧录后，请检查你修改的按键是否实现。