

[Open in app](#)[Sign up](#)[Sign in](#)[Search](#)[Kyria from SplitKB](#)

Separate keymap repo for QMK

Patrick Elmquist · [Follow](#)

4 min read · Sep 2, 2021

[Listen](#)[Share](#)

A while back I started getting into custom mechanical keyboards and the QMK firmware, a really fun side project. One issue I've faced since starting to create my own keymaps and custom functionality is that all my code have been living inside the massive `qmk_firmware` repository which means some extra headache when rebasing to get the latest features. This was especially clear recently when trying to rebase on the latest `master` branch after having cherry-picked a couple of PRs that were now merged. What I would like is to separate my code from the rest of the firmware, since I never change anything outside of my own folders anyway.

To sort this out I'd seen people use separate repositories for their custom code and symlink it when building. Having no experience whatsoever with that kind of black magic I started googling "QMK keymap separate repo" and to my surprise I couldn't find any blog posts about it. So now that I've finally take the time to set it up for myself, I thought I'd share the steps (it was a lot easier than I thought). What we'll be doing is to create a new repository, add QMK as a submodule and create a make file handling the symlinks and build steps.



Lily58 from [Mechboards](#)

First off, here's a link to my 'keymaps' repository on Github, it's what you will end up with when following the steps below:

GitHub - patrick-elmquist/qmk-keymaps: A user repository with my keymaps for QMK keyboards.

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or...

[github.com](https://github.com/patrick-elmquist/qmk-keymaps)

After following a discussion in #keymap-ideas on the [SplitKB Discord](#), I ended up using the following repo as a template for how to set it up:

GitHub - grasegger/kyria-layout

Contribute to grasegger/kyria-layout development by creating an account on GitHub.

[github.com](https://github.com/grasegger/kyria-layout)

Since I have more than one keyboard and a user space folder in `qmk_firmware` I needed to make some adjustments for it to suit my needs. These are the steps I went through:

Create a new Git repository and add QMK as a submodule

```
1 mkdir keymaps && cd keymaps
2 git init
3 git submodule add https://github.com/qmk/qmk_firmware.git
4 git submodule update --init --recursive
5 # this will also create a .gitmodules file
```

[create_repo.sh](https://github.com/grasegger/kyria-layout/blob/main/create_repo.sh) hosted with ❤ by GitHub

[view raw](https://github.com/grasegger/kyria-layout/blob/main/create_repo.sh)

We don't want to commit the build output to the repo, so add a `.gitignore` file with the following:

```
1 obj_*
2 *.hex
3 *.elf
4 *.map
```

[.gitignore](https://github.com/grasegger/kyria-layout/blob/main/.gitignore) hosted with ❤ by GitHub

[view raw](https://github.com/grasegger/kyria-layout/blob/main/.gitignore)

Update March 2023:

Do NOT name your repository (or at least the folder you clone into) `keymaps` as a recent change to QMK can make the builds fail with the following error:

```
qmk_firmware > qmk compile -kb dz60 -km default
Ψ Compiling keymap with gmake --jobs=1 dz60:default

QMK Firmware 0.20.0
make: *** No rule to make target 'dz60:default'. Stop.

| QMK's make format is:
|   make keyboard_folder:keymap_folder[:target]
|
| Where `keyboard_folder` is the path to the keyboard relative to
| `qmk_firmware/keyboards/`, and `keymap_folder` is the name of the
| keymap folder under that board's `keymaps/` directory.
|
| Examples:
|   keyboards/dz60, keyboards/dz60/keymaps/default
|     -> make dz60:default
|     -> qmk compile -kb dz60 -km default
|   keyboards/planck/rev6, keyboards/planck/keymaps/default
|     -> make planck/rev6:default:flash
|     -> qmk flash -kb planck/rev6 -km default
```

QMK look for a folder called `keymaps` to find the provided keymap to compile and it seems having the repo folder called the same can cause conflicts.

Add folder structure

Now lets create the folder structure. In the `qmk_firmware` repo I had `keymaps` for `Lily58`, `Kyria` and have common code for both in a user space folder. So this is this is the structure I use in the new repository:

```
keymaps/
+ kyria/
+ lily58/
+ user/
| .gitignore
` .gitmodules
```

Now transfer any files you already have

in `qmk_firmware/keyboards/<keyboard>/keymaps/` and `qmk_firmware/users/<user>` folders to their new location. **Note:** if you're transferring files to the `user` folder it assumes that you have already set up your user space with the correct files. If you haven't, you can find a guide for how to do it [here](#). Fore me the structure looked something like this after:

```
keymaps/
+ kyria/
|   | keymap.c
|   | config.h
|   ` rules.mk

+ lily58/
|   | keymap.c
|   | config.h
|   ` rules.mk

+ user/
|   | <user>.c
|   | <user>.h
|   ... more user files

| .gitignore
| .gitmodules
```

Create Makefile

When you're done there's only one more thing to do, and that's to configure a `Makefile` to setup the symlinks and build the firmware. I should mention that prior to this I've never written a `make` file, so there might be better ways to do this, but I wanted to make it easy to add more boards.

```

1 USER = pket
2
3 KEYBOARDS = lily58 kyria
4 PATH_lily58 = lily58
5 PATH_kyria = splitkb/kyria
6
7 all: $(KEYBOARDS)
8
9 .PHONY: $(KEYBOARDS)
10 $(KEYBOARDS):
11     # init submodule
12     git submodule update --init --recursive
13
14     # cleanup old symlinks
15     for f in $(KEYBOARDS); do rm -rf qmk_firmware/keyboards/$(PATH_$@)/keymaps/$(USER)
16     rm -rf qmk_firmware/users/$(USER)
17
18     # add new symlinks
19     ln -s $(shell pwd)/user qmk_firmware/users/$(USER)
20     ln -s $(shell pwd)/$@ qmk_firmware/keyboards/$(PATH_$@)/keymaps/$(USER)
21
22     # run lint check
23     cd qmk_firmware; qmk lint -km $(USER) -kb $(PATH_$@) --strict
24
25     # run build
26     make BUILD_DIR=$(shell pwd) -j1 -C qmk_firmware $(PATH_$@):$(USER)
27
28     # cleanup symlinks
29     for f in $(KEYBOARDS); do rm -rf qmk_firmware/keyboards/$(PATH_$@)/keymaps/$(USER)
30     rm -rf qmk_firmware/users/$(USER)
31
32 clean:
33     rm -rf obj_*
34     rm -f *.elf
35     rm -f *.map
36     rm -f *.hex

```

Makefile hosted with ❤️ by GitHub

[view raw](#)

Note: the Makefile assumes that the [QMK CLI](#) is installed.

Now we are pretty much done, to build, just go into the main folder and run:

```

1 make all # or
2 make lily58 # or
3 make kyria

```

script.sh hosted with ❤️ by GitHub

[view raw](#)

Wrapping up



Kyria from [SplitKB](#)

That's all I had to do to set it up. It's been working great, I love to have the code separate from the main repo and also closer together making it easier to navigate while developing.

Qmk

Keymap

Gitsubmodule

Git Repo

Keyboard