Contents

Guide to creating a layout in QMK for the TADA68

The TADA68 can be programmed using Easy AVR which provides a simple gui and can generate the required `.bin` file for the TADA68 but it is not as configurable as QMK nor does it have as many features and keycodes as QMK. Because of this, some people may opt to use QMK for their keyboard if Easy AVR does not have functionality that they require and this guide will outline how to do this.

Step 1: setup QMK

Environment setup

Windows

For Windows, follow the instructions here in the documentation as the method used for QMK on Windows changes every now and then.

Linux

For Linux, the only prerequisite required is `git` so install it via the following commands depending on your distribution:

- Debian/Ubuntu: `apt-get install git`
- Fedora/Redhat/Centos: `yum install git`
- Arch: `pacman -S git`

QMK setup

First, we want to download the QMK files by cloning them, simply run

```
git clone https://github.com/qmk/qmk_firmware.git
```

Now go into the QMK folder with

```
cd qmk_firmware
```

Finally, QMK includes an easy setup script to finish the process so run it by typing the following from the `qmk_firmware` directory

```
./util/qmk_install.sh
```

Now we can test the QMK environment by building a layout, let's try the default TADA one

```
make tada68:default:flashbin
```

Which should give an output like this

If this test build goes well, we're ready to make a layout.

Figure 1: cloning qmk files and running setup

Figure 2: running a test build

Step 2: create/edit a layout

QMK layouts are found in `keyboards/<keyboard>/keymaps/<keymap_name>/`. For example, the default layout for the TADA68 is `keyboards/tada68/keymaps/default/`.

To make your own layout, copy the `default` folder and rename it to something else, `clack` for example which would give us `keyboards/tada68/keymaps/clack/` which will have 3 files in it:

1. `keymap.c`
2. `readme.md`
3. `rules.mk`

```
» qmk_firmware git:(master) ls -l keyboards/tada68/keymaps
total 76
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 amnesia0287
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 ardakilic
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 default
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 fakb
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 fezzant
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 isoish
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 iso-uk
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 maartenwut
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 mattdicarlo
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 mlechner
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 mtdjr
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 pascamel
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 raylas
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 rgb
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 stephengrier
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 trashcat
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 tshack
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 unix
drwxr-xr-x 2 jibreil users 4096 Jun 26 07:09 wamsm_tada
» qmk_firmware git:(master) cp -r keyboards/tada68/keymaps/default keyboards/tada68/keymaps/clack
» qmk_firmware git:(master) ± ls -l keyboards/tada68/keymaps/clack
total 12
-rwxr-xr-x 1 jibreil users 3411 Jun 26 07:18 keymap.c
-rwxr-xr-x 1 jibreil users   90 Jun 26 07:18 readme.md
-rw-r--r-- 1 jibreil users 1132 Jun 26 07:18 rules.mk
» qmk_firmware git:(master) ±
```

Figure 3: copying the default layout

Out of those 3, the `readme.md` file is not required but is present by default, the `rules.mk` file contains advanced build options for QMK and for most users can be left alone. The `keymap.c` file is where the magic happens and in here you can edit the default layout to your own liking. Thorough instructions for editing this can also be found in the QMK docs.

Step 3: build the firmware

The TADA68 is unique to most other mechanical keyboards in terms of putting your own layout on it as you drop a `.bin` file on the keyboard as if it was a regular mass storage device rather than flashing a `.hex` file to the controller.

To generate the `.bin` file, just run the following command within the QMK root directory:

```
make tada68:<name>:flashbin
```

Where `<name>` is the name of the directory you created in step 2, following from our example where the name of the folder was `clack` you would run `make tada68:clack:flashbin`.

If there is an issue with your layout you'll get an error after the make command. If there are no errors, you'll find a shiny new `.bin` file in the QMK root directory, following our example again we'll find `tada68_clack.bin` in the root directory. This is the file you'll need to put onto your board.

Step 4: loading your layout onto your keyboard

To get your layout onto your keyboard, plug it in and press the reset switch, it should show up as a removable device and you'll find a `.bin` file already on it with the default layout, the board only has enough storage for one layout so backup this file somewhere on your computer and replave it with the `.bin` file created in step 3.

When the new layout file is on the board, press escape to get the board out of the bootloader mode and it ***should*** be working on the new layout.

```
» qmk_firmware git:(master) ± make tada68:clack:flashbin

QMK Firmware 0.6.49
WARNING:
 Some git sub-modules are out of date or modified, please consider runnning:
 make git-submodule
 You can ignore this warning if you are not compiling any ChibiOS keyboards,
 or if you have modified the ChibiOS libraries yourself.

Making tada68 with keymap clack and target flashbin

avr-gcc (GCC) 8.1.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Compiling: keyboards/tada68/tada68.c                                [OK]
Compiling: keyboards/tada68/keymaps/clack/keymap.c                  [OK]
Compiling: quantum/quantum.c                                        [OK]
Compiling: quantum/keymap_common.c                                  [OK]
Compiling: quantum/keycode_config.c                                 [OK]
Compiling: quantum/process_keycode/process_leader.c                 [OK]
Compiling: quantum/matrix.c                                         [OK]
Compiling: ./tmk_core/common/host.c                                 [OK]
Compiling: ./tmk_core/common/keyboard.c                             [OK]
Compiling: ./tmk_core/common/action.c                               [OK]
Compiling: ./tmk_core/common/action_tapping.c                       [OK]
Compiling: ./tmk_core/common/action_macro.c                         [OK]
Compiling: ./tmk_core/common/action_layer.c                         [OK]
Compiling: ./tmk_core/common/action_util.c                          [OK]
Compiling: ./tmk_core/common/print.c                                [OK]
Compiling: ./tmk_core/common/debug.c                                [OK]
Compiling: ./tmk_core/common/util.c                                 [OK]
Compiling: ./tmk_core/common/eeconfig.c                             [OK]
Compiling: ./tmk_core/common/report.c                               [OK]
Compiling: ./tmk_core/common/avr/suspend.c                          [OK]
Compiling: ./tmk_core/common/avr/timer.c                            [OK]
Compiling: ./tmk_core/common/avr/bootloader.c                       [OK]
Assembling: ./tmk_core/common/avr/xprintf.S                         [OK]
Compiling: ./tmk_core/common/magic.c                                [OK]
Compiling: ./tmk_core/common/mousekey.c                             [OK]
Compiling: ./tmk_core/common/command.c                              [OK]
Compiling: ./tmk_core/common/backlight.c                            [OK]
Compiling: ./tmk_core/protocol/lufa/lufa.c                          [OK]
Compiling: tmk_core/protocol/usb_descriptor.c                       [OK]
Compiling: ./tmk_core/protocol/lufa/outputselect.c                  [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Class/Common/HIDParser.c       [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/AVR8/Device_AVR8.c        [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/AVR8/EndpointStream_AVR8.c [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/AVR8/Endpoint_AVR8.c      [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/AVR8/Host_AVR8.c          [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/AVR8/PipeStream_AVR8.c    [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/AVR8/Pipe_AVR8.c          [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/AVR8/USBController_AVR8.c [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/AVR8/USBInterrupt_AVR8.c  [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/ConfigDescriptors.c       [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/DeviceStandardReq.c       [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/Events.c                  [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/HostStandardReq.c         [OK]
Compiling: lib/lufa/LUFA/Drivers/USB/Core/USBTask.c                 [OK]
Linking: .build/tada68_clack.elf                                    [OK]
Creating load file for flashing: .build/tada68_clack.hex            [OK]
» qmk_firmware git:(master) ±
```

Figure 4: building the new firmware

Figure 5: new firmware file