

[Skip to main content](#)

r/OpenRGB



Search in r/Op...



Create



r/OpenRGB • 2 yr. ago
Ryuu_Dango

Beginner General Guide to Getting Feker IK75 QMK/VIA Version (and other keebs) Working with OpenRGB

DISCLAIMER: I've only tested this out for my feker ik75 QMK/VIA keyboard, but the steps should be similar for other keyboards. Please try this out at your own discretion.

I originally went down this rabbit hole because of my feker ik75 QMK/VIA version. I wanted to get my feker working with OpenRGB but didn't know where to start since ORGB didn't recognize my keeb right out of the box. I was then given a hex file by discord user Bkzshen from the ORGB discord who got help from Discord user HorrorTroll and that got my feker working with ORGB, however, I wanted to make my own changes to my feker keymap layout and that's where my troubles began. I had issues creating a firmware that would work with ORGB and that's when Discord user MichaelGK helped teach me how to go about creating firmwares compatible with ORGB. I'm making this guide because as I was searching for guides myself before asking the help of the ORGB Discord community, I couldn't find any comprehensible ones that would help a beginner QMK keeb modder such as myself (I have some prior experience working with keychron keebs but it's still kinda new to me) navigate the ORGB git. So I'm hoping this little tutorial helps at least someone out there and makes the whole QMK + ORGB journey a little easier! :) special thanks to Bkzshen and MichaelGK for helping me out and to MichaelGK and HorrorTroll for their amazing ORGB gits of which can be found here:

<https://github.com/MichaelDavidGK/QMK-OpenRGB>

<https://gitlab.com/HorrorTroll/OpenRGB>

Tutorial Beginning

Open the QMK-MSYS application and type into the command prompt

```
git clone -b HorrorTroll --recurse-submodules https://github.com/MichaelDavidGK/QMK-  
OpenRGB .git
```

Make sure you have git installed otherwise this won't work. This will install a file titled 'QMK-Open RGB' in your user space (unless you cd to another file location and it should save there) that will contain all of the files related to OpenRGB compatible keyboards as well as install all submodules.

Next, open the QMK-Open RGB file and navigate to

```
>> keyboards >> <your keyboard model> >> keymaps
```

and create a new file and call it whatever you want. This is where you will paste your json file with your desired keyboard layout and rules.mk file to base your firmware from.

To create your layout json, you can use either the qmk configurator website here:

<https://config.qmk.fm/#/fecker/ik75/LAYOUT> (make sure to select your model of keyboard), or, you can manually alter the json of the default keymap of your keyboard. Once you have your json, move the file to the new folder you just created in

```
QMKG-Open RGB >> keyboards >> <your keyboard model> >> keymaps >> <filename>
```

[Skip to main content](#)[Create](#)`# Encoder enabled``ENCODER_MAP_ENABLE = yes`

We need to make openrgb enable set to yes so add another line above the ENCODER_MAP_ENABLE line so that your code looks like this:

`OPENRGB_ENABLE = yes``# Encoder enabled``ENCODER_MAP_ENABLE = yes`

Then save this file as rules.mk and save it in the same folder where you saved your json file. Right now, your created folder in <keymaps> should have two files in it: rules.mk and your json with your keyboard layout.

Now it is time to compile your hex file that you will use to flash your keyboard with to make it compatible with OpenRGB. Go back to QMK-MSYS and type the following command:

`qmk compile -kb <keybaord> -km <keymap file name>`

For example, if I wanted to write the firmware to my feker IK75, I would write

`qmk compile -kb feker/ik75 -km mykeymap`

This should start the compiling process and once it is done, you should see something along these lines:

`Copying feker_ik75_mykeymap.hex to qmk_firmware folder [OK]``Checking file size of feker_ik75_mykeymap.hex [OK]``* The firmware size is fine - 22230/28672 (93%, 6442 bytes free)`

WARNING: if the hex file firmware size exceeds your keyboard's memory, IT WILL BRICK YOUR KEYBOARD. In my example, you see that the firmware size is 93%. For my feker, the ideal size would be somewhere around 80% or less. In order to reduce the firmware size, scroll down to the bottom of this tutorial for size reduction tips!

Your firmware should save in the 'QMK-Open RGB' folder. Open the qmk_toolbox, choose your firmware from the dropdown menu, plug in your keyboard and set it to bootloader mode (for my feker, I just need to unplug the keeb, hold down the esc key while plugging it back in, and that sets it to the bootloader mode), and then flash your firmware.

To get your keeb working with OpenRGB, open the application. Navigate to the 'Settings' tab and scroll down to the 'OpenRGB QMK Protocol' section. Here, you must add your keyboard manually so that it can recognize your keeb. Write down a name, and find your keyboard's USB VID and USB PID code numbers. You can find this by going to your device manager on your computer and opening the 'Universal Serial Bus Controllers' dropdown menu. Double click the USB device (It might be labeled as USB Composite Device), go to the 'Details' tab and on the dropdown menu, select 'Hardware IDs'. This will display your USB VID and PID values. Type these combinations of numbers and letters into the OpenRGB software and save the information. As an example, the feker's VID is F2E7 and PID is 1226 (this is universal for all fekers). Close the OpenRGB application and reopen it. OpenRGB should now recognize and work with your keyboard! Enjoy! :)

[Skip to main content](#)[Create](#)

In order to reduce the firmware size, you can do the following as listed in this article: <https://get.vial.today/docs/firmware-size.html> and if doing this still results in a high firmware size, follow the below steps.

Some ways to further reduce file size would be to

1. Reduce the number of keymap layouts to 2
2. Disable RGB effects

These are done in your keyboard's config.h file. Navigate to

```
QMK-Open RGB >> keyboards >> <your keyboard model> >> config.h
```

You can do step 1 by writing the following command in the top of the config.h file right below #include "config_common.h"

```
#define DYNAMIC_KEYMAP_LAYER_COUNT 2
```

Next, to disable RGB modes, you simply need to navigate to the RGB matrix section and change the #define to #undef. For example, these are the RGB effects I disabled:

```
#undef ENABLE_RGB_MATRIX_ALPHAS_MODS
```

```
#undef ENABLE_RGB_MATRIX_BAND_SAT
```

```
#undef ENABLE_RGB_MATRIX_BAND_PINWHEEL_SAT
```

```
#undef ENABLE_RGB_MATRIX_BAND_SPIRAL_SAT
```

```
#undef ENABLE_RGB_MATRIX_BAND_SPIRAL_VAL
```

```
#undef ENABLE_RGB_MATRIX_RAINBOW_MOVING_CHEVRON
```

```
#undef ENABLE_RGB_MATRIX_CYCLE_OUT_IN_DUAL
```

```
#undef ENABLE_RGB_MATRIX_CYCLE_SPIRAL
```

```
#undef ENABLE_RGB_MATRIX_RAINBOW_PINWHEELS
```

```
#undef ENABLE_RGB_MATRIX_PIXEL_FLOW
```

```
#undef ENABLE_RGB_MATRIX_PIXEL_FRACTAL
```

Once you do this, save the file and compile the firmware again. This time, you should get a firmware file that has a lower size percentage than before. As an example, this is my size output after making the above changes:

```
Copying feker_ik75_mykeymap.hex to qmk_firmware folder [OK]
```

```
Checking file size of feker_ik75_mykeymap.hex [OK]
```

```
* The firmware size is fine - 22230/28672 (77%, 6442 bytes free)
```

[Skip to main content](#)[Create](#)

Here are screenshots of my tutorial that I initially wrote on Carbon. I posted two copies, one light mode version of the tutorial, and one dark mode version, for those who have a preference.

```
// Open the QMK-MSYS application and type into the command prompt
git clone -b HorrorTroll --recurse-submodules https://github.com/MichaelDavidGK/QMK-OpenRGB

// Make sure you have git installed otherwise this won't work. This will install a file titled 'QMK-OpenRGB' in your user space (unless you cd to another file location and it should save there) that will contain all of the files related to OpenRGB compatible keyboards as well as install all submodules.

// Next, open the QMK-Open RGB file and navigate to
>> keyboards >> <your keyboard model> >> keymaps

// and create a new file and call it whatever you want. This is where you will paste your json file with your desired keyboard layout and rules.mk file to base your firmware from.

// To create your layout.json, you can use either the qmk configurator website here:
https://config.qmk.fm/#/feker/ik75/LAYOUT (make sure to select your model of keyboard), or, you can manually alter the json of the default keymap of your keyboard. Once you have your json, move the file to the new folder you just created in

QMK-Open RGB >> keyboards >> <your keyboard model> >> keymaps >> <filename>

// Next you want to write the rules.mk file. Normally what I do is I open an existing rules.mk file from the default folder under <keymaps> and then copy and paste that into a new file. It should look something similar to this:

# Encoder enabled
ENCODER_MAP_ENABLE = yes

// We need to make openrgb enable set to yes so add another line above the ENCODER_MAP_ENABLE line so that your code looks like this:
OPENRGB_ENABLE = yes

# Encoder enabled
ENCODER_MAP_ENABLE = yes

// Then save this file as rules.mk and save it in the same folder where you saved your json file. Right now, your created folder in <keymaps> should have two files in it: rules.mk and your json with your keyboard layout.

// Now it is time to compile your hex file that you will use to flash your keyboard with to make it compatible with OpenRGB. Go back to QMK-MSYS and type the following command:
qmk compile -kb <keyboard> -km <keymap file name>

// For example, if I wanted to write the firmware to my feker IK75, I would write
qmk compile -kb feker/ik75 -km mykeymap

// This should start the compiling process and once it is done, you should see something along these lines:
Copying feker_ik75_mykeymap.hex to qmk_firmware folder
[0%]
Checking file size of feker_ik75_mykeymap.hex
[0%]
* The firmware size is fine - 22238/28672 (9%, 6442 bytes free)

// WARNING: If the hex file firmware size exceeds your keyboard's memory, IT WILL BRICK YOUR KEYBOARD. In my example, you see that the firmware size is 9%. For my feker, the ideal size would be somewhere around 80% or less. In order to reduce the firmware size, scroll down to the bottom of this tutorial for size reduction tips!

// Your firmware should save in the 'QMK-Open RGB' folder. Open the qmk_toolbox, choose your Firmware from the dropdown menu, plug in your keyboard and set it to bootloader mode (for my feker, I just need to unplug the keyboard, hold down the esc key while plugging it back in, and that sets it to the bootloader mode), and then flash your firmware.

// To get your keyboard working with OpenRGB, open the application. Navigate to the 'Settings' tab and scroll down to the 'OpenRGB QMK Protocol' section. Here, you must add your keyboard manually so that it can recognize your keyboard. Write down a name, and find your keyboard's USB VID and USB PID code numbers. You can find this by going to your device manager on your computer and opening the 'Universal Serial Bus Controller' (or something similar) and then right-clicking on your keyboard and selecting 'Properties'. Under 'Device', go to the 'Details' tab and on the dropdown menu, select 'Hardware IDs'. This will display your USB VID and PID values. Type these combinations of numbers and letters into the OpenRGB software and save the information. As an example, the feker's VID is FEEF and PID is 1226 (this is universal for openrgb application and reopen it. openRGB should now recognize and work with your keyboard! Enjoy:-) )

#####
##### What to do if your firmware size is too big //
In order to reduce the firmware size, you can do the following as listed in this article:
https://get.vial.today/docs/firmware-size.html and if doing this still results in a high firmware size, follow the below steps.

// Some ways to further reduce file size would be to
// 1. Remove unused keymaps
// 2. Disable RGB effects
// These are done in your keyboard's config.h file. Navigate to
QMK-Open RGB >> keyboards >> <your keyboard model> >> config.h

// You can do step 1 by writing the following command in the top of the config.h file right below
#include "config_common.h"

#define DYNAMIC_KEYMAP_LAYER_COUNT 2

// Next, to disable RGB modes, you simply need to navigate to the RGB matrix section and change the #define to #undef. For example, these are the RGB effects I disabled:
#undef ENABLE_RGB_MATRIX_ALPHAS_MODS
#undef ENABLE_RGB_MATRIX_BAND_SAT
#undef ENABLE_RGB_MATRIX_BAND_SPIRAL_SAT
#undef ENABLE_RGB_MATRIX_BAND_SPIRAL_VAL
#undef ENABLE_RGB_MATRIX_RAINBOW_MOVING_CHEVRON
#undef ENABLE_RGB_MATRIX_CYCLE_OUT_IN_DUAL
#undef ENABLE_RGB_MATRIX_CYCLE_OUT_IN_SPIRAL
#undef ENABLE_RGB_MATRIX_RAINBOW_FFMHEELS
#undef ENABLE_RGB_MATRIX_PIXEL_FLOW
#undef ENABLE_RGB_MATRIX_PIXEL_FRACTAL

// Once you do this, save the file and compile the firmware again. This time, you should get a firmware file that has a lower size percentage than before. As an example, this is my size output after making the above changes:
Copy/Run feker_ik75_mykeymap.hex to qmk_firmware folder
[0%]
Checking file size of feker_ik75_mykeymap.hex
[0%]
* The firmware size is fine - 22238/28672 (77%, 6442 bytes free)

// As you can see, my file size had a reduction of 10%, making it safe to flash to my keyboard without bricking it.

// It is now safe to flash your keyboard with this firmware.
```

Light mode version

[Skip to main content](#)[Create](#)

```

open RGB to you user space (unless you cd to another file location and it should save there) that will
contain all of the files related to OpenRGB compatible keyboards as well as install all submodules.

// Next, open the QMK_Open RGB file and navigate to

>>> keyboards >> <your keyboard model> >> keymaps

// and create a new file and call it whatever you want. This is where you will paste your json file
with your desired keyboard layout and rules.mk file to base your firmware from.

// To create your layout json, you can use either the qmk configurator website here:
https://config.qmk.fm/#/feker/ik75/LAYOUT (make sure to select your model of keyboard), or, you can
simply copy and paste the default keymap of your keyboard. Once you have your json, move the file
to the same folder under <keymaps> and then create a new file. It should look
something similar to this:

QMK_Open RGB >> keyboards >> <your keyboard model> >> keymaps >> <filename>

// Next you want to write the rules.mk file. Normally what I do is I open an existing rules.mk file
from the default folder under <keymaps> and then copy and paste that into a new file. It should look
something similar to this:

# Encoder enabled
ENCODER_MAP_ENABLE = yes

// We need to make openrgb enable set to yes so add another line above the ENCODER_MAP_ENABLE line so
that your code looks like this:

OPENRGB_ENABLE = yes

# Encoder enabled
ENCODER_MAP_ENABLE = yes

// Then save this file as rules.mk and save it in the same folder where you saved your json file. Right
now your created folder in <keymaps> should have two files in it: rules.mk and your json with your
desired layout.

// Now it is time to compile your hex file that you will use to flash your keyboard with to make it
compatible with OpenRGB. Go back to QMK-HYS and type the following command:

qmk compile -kb <keyboard> -km <keymap file name>

// For example, if I wanted to write the firmware to my feker IK75, I would write

qmk compile -kb feker/ik75 -km mykeymap

// This should start the compiling process and once it is done, you should see something along these
lines:

copying feker_ik75_mykeymap.hex to qmk_firmware folder
[OK]
Checking file size of feker_ik75_mykeymap.hex
[OK]
* The firmware size is fine - 2228/28672 (9%, 6442 bytes free)

// WARNING: if the hex file's firmware size exceeds your keyboard's memory, IT WILL BRICK YOUR KEYBOARD.
In my example, you see that the firmware size is 9%. For my feker, the ideal size would be somewhere
around 80% or less. In order to reduce the firmware size, scroll down to the bottom of this tutorial
for size reduction tips!

// Your Firmware should save in the 'QMK_Open RGB' folder. Open the qmk_toolbox, choose your firmware
from the dropdown menu, plug in your keyboard and set it to bootloader mode (For my feker, I just need
to unplug the keyboard, hold down the esc key while plugging it back in, and that sets it to the bootloader
mode), and the flash your firmware.

// To get your keyboard working with OpenRGB, open the application. Navigate to the 'Settings' tab and
scroll down to the 'OpenRGB QMK Protocol' section. Here, you must add your keyboard manually so that it
can recognize your keyboard. Write down a name, and find your keyboard's USB VID and USB PID code numbers.
You can find these by going to your device manager and selecting 'USB Controller' and then double click the USB device (It might be labeled as USB Composite
Device), go to the 'Details' tab and on the dropdown menu, select 'Hardware ID'. This will display
your USB VID and PID values. Type these combinations of numbers and letters into the OpenRGB software
and save the information. As an example, the feker's VID is PZP and PID is 3220 (this is universal for
all fekers). Close the OpenRGB application and reopen it. OpenRGB should now recognize and work with
your keyboard! Enjoy! :)
```

```

// What to do if your firmware size is too big //
// In order to reduce the firmware size, you can do the following as listed in this article:
https://get.vial.today/doc/firmware-size.html and if doing this still results in a high firmware size,
follow the below steps.

// Some ways to further reduce file size would be to
// 1. Reduce the number of keymap layouts to 2
// 2. Disable RGB effects
// These are done in your keyboard's config.h file. Navigate to

QMK_Open RGB >> keyboards >> <your keyboard model> >> config.h

// You can do step 1 by writing the following command in the top of the config.h file right below
#include "config_common.h"

#define DYNAMIC_KEYMAP_LAYER_COUNT 2

// Next, to disable RGB modes, you simply need to navigate to the RGB matrix section and change the
directive to #undef. For example, these are the RGB effects I disabled:

#undef ENABLE_RGB_MATRIX_ALPHAS_MODS
#undef ENABLE_RGB_MATRIX_BAND_SAT
#undef ENABLE_RGB_MATRIX_BAND_VAL
#undef ENABLE_RGB_MATRIX_BAND_PINWHEEL_SAT
#undef ENABLE_RGB_MATRIX_BAND_SPIRAL_SAT
#undef ENABLE_RGB_MATRIX_BAND_SPIRAL_VAL
#undef ENABLE_RGB_MATRIX_RAINBOW_MOVING_CHEVRON
#undef ENABLE_RGB_MATRIX_CYCLE_SEQUENTIAL
#undef ENABLE_RGB_MATRIX_CYCLE_DUAL
#undef ENABLE_RGB_MATRIX_CYCLE_SPIRAL
#undef ENABLE_RGB_MATRIX_PIXEL_FLOW
#undef ENABLE_RGB_MATRIX_PIXEL_PINWHEELS
#undef ENABLE_RGB_MATRIX_PIXEL_FRACTAL

// Once you do this, save the file and compile the firmware again. This time, you should get a firmware
file that has a lower size percentage than before. As an example, this is my size output after making
the above changes:

copying feker_ik75_mykeymap.hex to qmk_firmware folder
[OK]
Checking file size of feker_ik75_mykeymap.hex
[OK]
* The firmware size is fine - 2228/28672 (7%, 6442 bytes free)

// As you can see, my file size had a reduction of 10%, making it safe to flash to my keyboard without
bothering it.

// It is now safe to flash your keyboard with this firmware.
```

Dark mode version

[12](#)

9

[Share](#)[Add a comment](#)

Sort by: Best

[Search Comments](#)

vhailorx • 2y ago

[Skip to main content](#)[Create](#)

to get qmk/via working nicely with orgb? The only other reddit post I saw on this subject from several months ago indicated that via didn't even recognize the board at all (which is pretty discouraging when they are charging a ~50% premium for it).

3

Ryuu_Dango OP • 2y ago

So without enabling Open RGB on the via feker model, I can confirm via works with this model! But, once you enable Open RGB, Via doesn't work anymore. That's why I had to modify my keyboard layout this way.

Edit: also, I just used the feker as an example, this method should work for any qmk model keyboard that orgb supports!

1

mad_dog_94 • 2y ago

hey so im doing this with the same gits on my nk87 but when i go to compile it references quantum.h and i get an error because rgb isn't a valid type name, but i havent changed the file at all. i was hoping you could help me out with why this might be happening?

2

johnyeros • 2d ago

Does this work for Feker IK85?

1

ringowu1234 • 2y ago

Hi, I tried following your steps to create hex file. I created my own .json file, put it together with **rules.mk** as you instructed. However when I typed in qmk compile -kb gmmk/pro/rev1/ansi -km RW, it gave me an error saying Invalid keymap.

I tried to figure out what's wrong, grabbed the keymap.c from default and modified as the way I wanted, together with **rules.mk** the compile works, flashed it, and the key definition is exactly the way I want.

However OpenRGB still doesn't detect my QMK ready GMMK Pro, even after entering correct USD VID/PID in OpenRGB (Pipeline, not 0.7)

I think the reason it failed to detect is because I used keymap.c instead of .json file. Is this assessment correct? Or am I missing anything? the QMK Compile just doesn't work when I use .json file.

Any suggestions? Thanks in advance.

1

ringowu1234 • 2y ago • Edited 2y ago



+ Create



The actual solution is here:

If you did "GMK setup" command in QMK MSYS, then the problem is that the "GMK Setup" command downloads another "QMK_Firmware" folder into your user folder. The compile command runs with the default keymap in "QMK_Firmware", not the files I put in "QMK-OpenRGB" as instructed in this guide. Therefore the new bin file is not compiled from my own personal JSON, and is not OpenRGB compatible.

What I did was to delete the "QMK_Firmware", and rename the "QMK-OpenRGB" I git from the guide to "QMK_Firmware", then follow the guide to compile.

↑ 1 ↓ Reply ...



GameOfShadows • 2y ago

Hiya, just tried this out on my Feker IK75. firmware compiled and flashed successfully, added the custom USB PID and VID, but openrgb still doesn't detect it :(

↑ 1 ↓ Reply ...



PooyaHesabi • 1y ago

This is the result

<https://www.youtube.com/shorts/5i7AhIThbyE>

↑ 1 ↓ Reply ...



Optimal_Shopping_414 • 10mo ago

any chance getting it working on keychron v3?

↑ 1 ↓ Reply ...