

# User Guide for Keyboards with *nrfmacro*

Suitable for Sweep-Pro, Corne-Pro

Yong

Date: July 23, 2022

## Contents

<b>1</b>	<b>Power and reset buttons</b>	<b>1</b>
<b>2</b>	<b>General picture of Bluetooth connection in zmk</b>	<b>2</b>
2.1	Pairing and Connecting . . . . .	2
2.2	Profiles . . . . .	2
<b>3</b>	<b>Configuration</b>	<b>2</b>
<b>4</b>	<b>Customizing personal logo</b>	<b>3</b>
<b>5</b>	<b>Useful configurations</b>	<b>4</b>
5.1	CONFIG_ZMK_SLEEP and CONFIG_ZMK_IDLE_SLEEP_TIMEOUT . . . . .	4
5.2	CONFIG_ZMK_DISPLAY_FULL_REFRESH_PERIOD . . . . .	4
5.3	Screen margin adjustment . . . . .	4
<b>6</b>	<b>Status on display</b>	<b>4</b>
6.1	connection status . . . . .	5

## Abstract

This document contains two pieces of information:

- the one which already exists in the [official document](#). It is repeated here to emphasize the importance.
- the one which is specific about the keyboards with *nrfmacro* MCU board.

*nrfmacro* is tweaked version of the more well-known *nrfmicro*. They use the same MCU (nrf52840) and share the same circuit design. But *nrfmacro* routes out more pins, thus the board dts is slightly different. Besides this, the display, the encoders and some minor firmware features all need specific configuration which are not available on main branch. For these reasons, a dedicated board called *nrfmacro* is developed and maintained in a [fork branch of zmk](#).

## 1 Power and reset buttons

There're two tactile buttons located on the inner side of each split:

- 'Red' power button: turning on the keyboard by push this button up.

- 'White' reset button: reset the keyboard by push the button once; enter the bootloader mode by pushing this button twice in a short interval

## 2 General picture of Bluetooth connection in zmk

The keycodes related to the Bluetooth connection are described in [the official document](#).

### 2.1 Pairing and Connecting

There are two stages in establishing a bluetooth connection:

- **Pair:** this is a one-time process to bond the keyboard and the host. Security keys are exchanged and stored on both sides so that they will reused when reconnected.
- **Connecting:** this process is repeated everytime when the host and the keyboard are disconnected and need to established the connection again. Only the paired devices can connect with each other.

Correspondingly, there are two reverse stages:

- **Unpair:** the device clears up the security keys associated with the counterpart of an existing pair relationship. After unpair, the device needs to pair with the same device first to communicate with it later.
- **Disconnect:** the device close the current connection with the device in an existing pair relationship. It can connect to the same device again without pairing.

The connection status is shown on the status screen of the keyboard, see Sec. [6.1](#) for details.

**Warning:** Unpairing is a two-fold process. This means that unpairing manipulation needs to be taken out both on the host and the keyboard to remove an existing pair relation. If not, the host can not pair with the keyboard again since old data is not cleared on one side of the connection.

### 2.2 Profiles

There are 5 ble connection profiles in zmk firmware by default, meaning the keyboard can pair with 5 hosts simultaneously. The switching between the profiles is quick and smooth, but the user needs to be aware which profile the keyboard is using currently.

The profiles are indexed from 1 to 5. The current profile is shown on the status screen of the keyboard, see Sec. [6.1](#) for details. User should always check this information first before pair/unpair and connect/disconnect.

## 3 Configuration

The configuration uses the same framework and infrastructure as official zmk board/shield combination. The GitHub workflow is used to build the firmware online. End user needs to create a repository with a defined directory structure (this directory is called [zmk-config](#) in [zmk terminology](#)).

The keymap changes and configuration changes introduced into [zmk-config](#) repository is applied to the firmware by a github workflow, which automatically rebuild the firmware whenever changes are committed to the [zmk-config](#) repository.

**Sweep-Pro** is not officially supported by zmk main branch and some features used in **Sweep-Pro** are not yet merged into the zmk main branch. Thus, instead of creating this repository using the official tool, the `zmk-config` repository for **Sweep-Pro** should be created by forking the [reference repo](#). The reference repo has correctly set up the workflow to point to a forked branch of zmk which supports all the features used by **Sweep-Pro**. For **Corne-Pro**, the reference repo is [https://github.com/ufan/corne-pro\\_config/](https://github.com/ufan/corne-pro_config/).

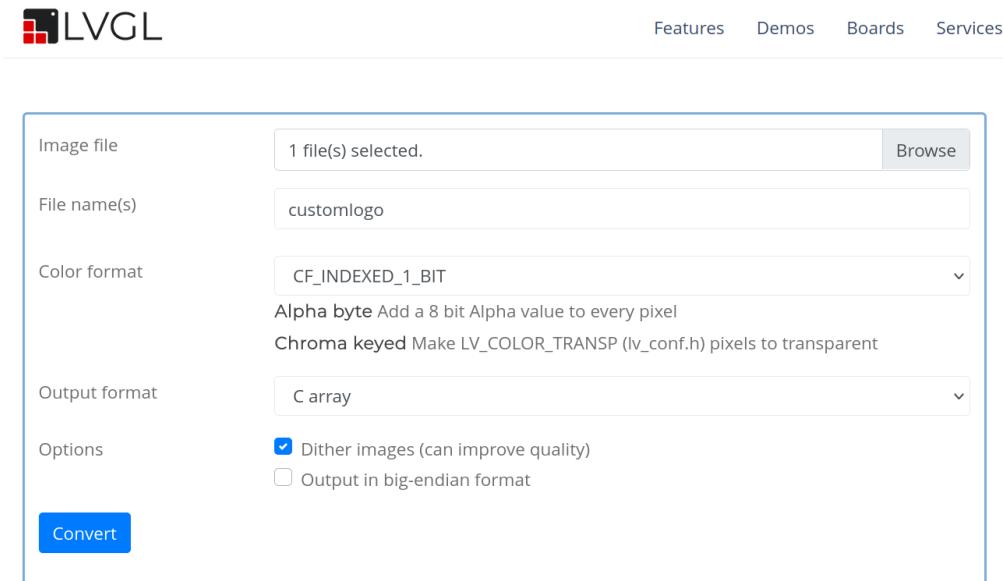
Other than that, it is the same procedure to customize your personal keymap as the official documentation.

## 4 Customizing personal logo

The central part of the epd display on the slave split (i.e., the peripheral side in Bluetooth terminology) can be customized to show an image provided by the end user. Usually it is used to show the personalized logo, either designed or downloaded by the end user. The available area is about 80x70 pixels, be aware of it when designing your logo.

User needs to convert the bitmap logo picture (usually in PNG format) into a C source file named 'customlogo.c', and saved under the `zmk_config` directory ( `${ZMK_CONFIG}/icons/customlogo.c`). The lvgl [online converter](#) is used to achieve this goal, as follows:

1. Open the browser and go to the online converter page
2. Click 'Browse' button and upload the logo image
3. The other options should be set as shown in the following picture:



**Figure 1:** The correct settings for converting an image to be used on epd display

After the above steps, add the following setting to config file of the slave split (for example, `sweepro-ce_right.conf`). Then, commit and push the changes. After flashing the latest firmware, the new logo should show up on the slave display.

```
CONFIG_NRFMACRO_SCREEN_CUSTOM_LOGO=y
```

To put the image in the center horizontally, the width of the logo image should be set to 80. To put the image in the center vertically, `CONFIG_NRFMACRO_SCREEN_BOTTOM_MARGIN` config option (see Sec. 5.3) can be used to adjust the vertical position. Be aware that when adjusting the bottom margin, the mark logo is also affected.

## 5 Useful configurations

### 5.1 CONFIG\_ZMK\_SLEEP and CONFIG\_ZMK\_IDLE\_SLEEP\_TIMEOUT

These options are used together to put the keyboard into deep-sleep mode if the keyboard is idle for a timeout value defined by `CONFIG_ZMK_IDLE_SLEEP_TIMEOUT` (unit: ms).

To wake up the keyboard, just press any key. Be aware that the current running states are not preserved in deep-sleep mode, thus reconnection is needed when waking up. Thus, the delay is quite noticeable, just like powering on the keyboard. It is recommended to set this value above 15 mins.

**Note:** Deep-sleep mode is different from idle-mode in zmk. When the device is in idle mode, the display is not updated but other threads keep running. When in deep-sleep mode, the current process is stopped, there is even no voltage supply to the display. Deep-sleep is a hardware feature, idle-mode is a software feature. The default idle timeout value is 30s.

### 5.2 CONFIG\_ZMK\_DISPLAY\_FULL\_REFRESH\_PERIOD

This option is used to control the time interval to refresh the full screen of the display (unit: s).

It is essential for epd display to have a clear screen all the time. To save power, partial refresh is used under normal circumstances. This means that only the widget which needs to be updated is refreshed. Due to the working principle of epd, faded image may appear on the screen if the corresponding widgets are not updated in a long time. This option solves this issue by refreshing the full screen in a fixed interval.

### 5.3 Screen margin adjustment

The margins of the display screen can be adjusted in pixels using the following config options:

- `CONFIG_NRFMACRO_SCREEN_TOP_MARGIN`
- `CONFIG_NRFMACRO_SCREEN_BOTTOM_MARGIN`
- `CONFIG_NRFMACRO_SCREEN_LEFT_MARGIN`
- `CONFIG_NRFMACRO_SCREEN_RIGHT_MARGIN`

## 6 Status on display

If the keyboard is equipped with a display on each split, then the following information is available as shown in Fig. 2:

- central/left split:
  - 1 battery level (a lightning icon appears when charging)
  - 2 Connection status

- an USB icon appears if the connection channel is established using cable and the output selection is USB
  - an Bluetooth icon associated with a number appears if the USB cable is unplugged or the output selection is Bluetooth when the USB cable is connected. The number indicates the current profile.
- This icon changes with the connection status of the current profile as explained in Sec. 6.1.

3 Current layer (the label is defined in the keymap file, which is customizable by the user)

- peripheral/right split:
  - 4 Status of the connection with the central split
  - 5 battery level (a lightning icon appears when charging)
  - 6 Personal logo image (customizable by the user, see Sec. 4)
  - 7 Keyboard mark logo

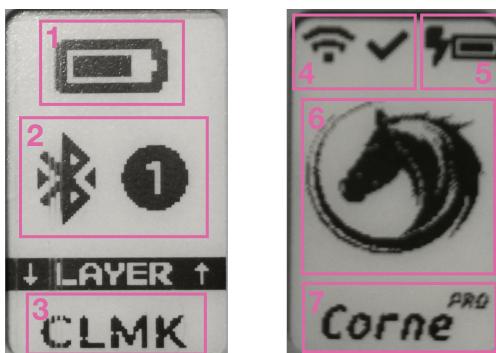


Figure 2: Status screen of the central/left split and peripheral/right split

## 6.1 connection status

Three phases of the Bluetooth connection are shown on the screen of the central split as shown in Fig. 3:

1. **Broadcasting:** the keyboard is ready to be paired on current profile. The keyboard is discoverable on a new host only when it is in this phase.
2. **Connected:** the connection with the host is established using current profile. It is ready to be typed and the output will go through the bluetooth channel
3. **Disconnected:** the pairing with the host exists but the connection failed

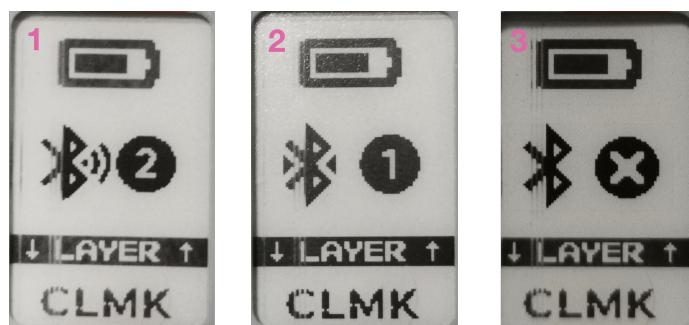


Figure 3: The different phases in Bluetooth connection