



# Glove80 Layout Editor Guide

Date: 5 December 2022 DRAFT

# Table of Contents

<b>Introduction</b>	<b>3</b>
Technical requirements	3
Alternative methods to generate customized Glove80 firmware	3
<b>Getting started</b>	<b>4</b>
my.glove80.com	4
Creating an account	4
Starting your first Glove80 Layout	4
<b>Searching for layouts</b>	<b>5</b>
<b>My Layouts</b>	<b>7</b>
<b>Layout</b>	<b>8</b>
Title	8
Tags	8
Layer selector	9
Notes	10
Custom Defined Behavior	10
Key binding & key behaviors	10
Behavior types	11
Keypress behavior: modifier functions	12
&magic pseudo behavior	12
&lower pseudo behavior	12
Bluetooth &bt_0 &bt_1 &bt_2 &bt_3 pseudo behaviors	13
Bluetooth clear all profiles &bt BT_CLR_ALL	13
Sharing layouts	13
<b>Printing layouts</b>	<b>14</b>
<b>Building firmware</b>	<b>15</b>
Loading the firmware onto the Glove80	15
Limits on number of builds	15
<b>Advanced usage: Custom Defined Behaviors</b>	<b>16</b>
Custom Defined Behaviors	16
Key binding for Custom Defined Behavior	17

# Introduction

All of us are different. We have different hand shapes and sizes. We use different applications and we have different habits. We are simply different. Glove80 is designed to adapt to you and your needs, with its highly configurable hardware design and software/firmware as well.

Glove80 Layout Editor is a web application designed to simplify the customization of your Glove80 ergonomic keyboard. With Glove80 Layout Editor you can change the function of each key. It is a complementary service provided by MoErgo, and is subject to its own terms and conditions.

Glove80 Layout Editor is designed to be social, so that the Glove80 user community can share layout and layout ideas. With Glove80 Layout Editor, you can clone a layout shared by another community member, and quickly modify it to make it *your own*.

Glove80 Layout Editor is super easy to use. There is no installation needed. Just create an account and get going within minutes.

## Technical requirements

- Web browser: Chrome, Safari or Firefox
- Device: desktop or laptop

Due to the exceptionally broad range of mobile devices, mobile devices are not officially supported. However we aim to provide a decent level of support, and we do test Glove80 Layout Editor with Chrome browser on Android devices.

## Alternative methods to generate customized Glove80 firmware

If you would prefer not to use a graphical user interface to create your Glove80 layout, the traditional ZMK methods are there for you.

Please see:

- <https://github.com/moergo-sc/glove80-zmk-config> to use the Github action to generate ZMK firmware in the traditional ZMK method
- <https://zmk.dev/docs/user-setup> ZMK documentation on how to generate ZMK firmware, using the standard ZMK method. With Glove80 Layout Editor, you can directly generate the firmware without a Github account, or installing a compiler toolchain.

# Getting started

my.glove80.com

In your favorite browser that is [supported by Glove80 Layout Editor](#), visit <https://my.glove80.com>

That's it. No application installation is required.

## Creating an account

You don't need an account to use Glove80 Layout Editor to design a layout and to generate a ZMK keymap file.

However an account is needed to:

- Save your layouts
- Build firmware within the Glove80 Layout Editor

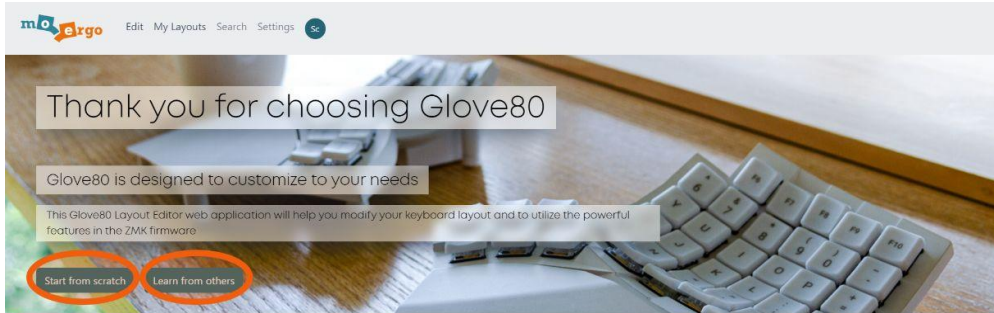
To create a Glove80 Layout Editor account, click on the Sign-Up on the menu bar, or visit <https://my.glove80.com/#/signup>.

## Starting your first Glove80 Layout

From the front page of the Glove80 Layout Editor, you can

- Immediately start editing by clicking on "Start from scratch" button
- Get some inspiration first from the community to see what layouts they have made.
- Use one of the six standard templates to get started:
  - Glove80 factory standard layout
  - Colemak
  - Colemak-DH
  - Dvorak
  - Workman
  - Kinesis

If you were previously using a Kinesis Advantage keyboard (just like all of us who designed and tested Glove80), we have created a layout that mimics Kinesis Advantage's default layout to help you migrate. Over time, you may want to take advantage of Glove80's improved thumb cluster by moving more keys into the thumb cluster, but there is no need to rush; you can do it at your own pace.



Or start with a template

Glove80 factory default

Colemak

Colemak-DH

Workman

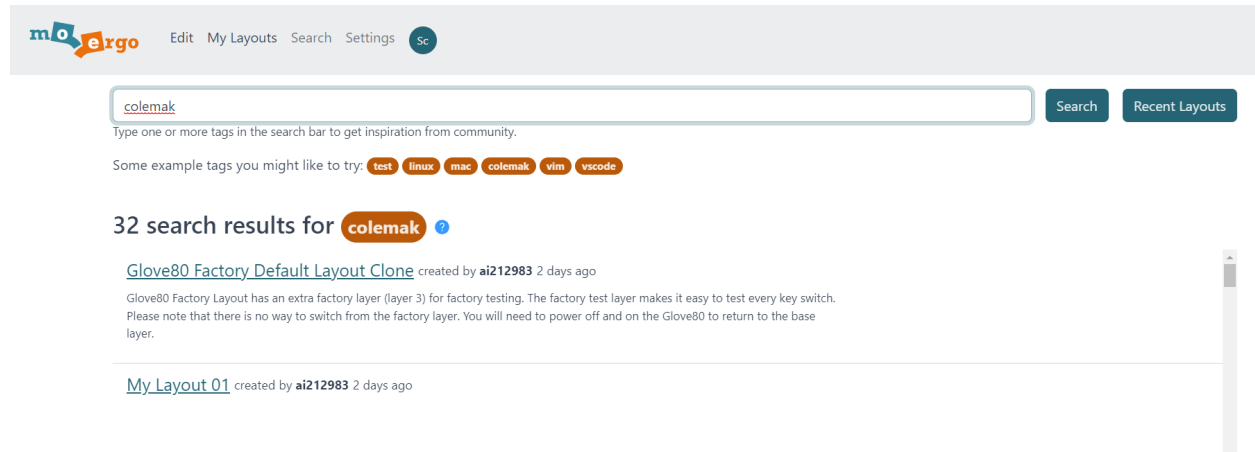
Kinesis

# Searching for layouts

Glove80 Layout Editor provides a convenient place for all of us to share our Glove80 layouts.

We found we often get inspiration by looking at layouts made by other Glove80 users. Sometimes other users have already made a layout that already satisfies your needs.

<https://my.glove80.com/#/search/>



You can type one or multiple search terms, separated by commas, and press on the “Search” button to begin the search. Only layouts with tags that match all of the search terms will show up.

Alternatively, click the “Recent Layouts” button to find all recent layouts.

Please note that a Layout is listed only in the search results or recent layouts after its firmware is successfully built.

# My Layouts

Glove80 Layout Editor keeps a record of your saved layouts. You can retrieve your layouts on any compatible web browser. This functionality is only available if you are logged into your account.

You can reach the My Layouts page by clicking on the “My Layouts” in the menu, or visiting [https://my.glove80.com/#/my\\_layouts/](https://my.glove80.com/#/my_layouts/)

You can have up to 500 saved layouts.

# Layout

When we view a layout, whether it is a template layout, a layout created by another community member or your own saved layout, you will see a screen like below.

The screenshot shows the MoErgo layout editor interface. At the top, there is a navigation bar with 'mo ergo' logo, 'Edit My Layouts Search Settings', and a user profile 'sc'. Below the navigation bar is a title field containing 'Example showing how to define a macro'. A search bar contains the text 'macro custom-defined-behaviours custom example' with 'Tags' to its right. On the left, a 'Layers' selector shows three layers (0, 1, 2) with layer 0 selected. The main area displays a keyboard layout with various keys labeled, including function keys (F1-F10), alphanumeric keys, and modifier keys. Below the keyboard layout, there are two sections: 'Custom Defined Behaviours' and 'Notes'. The 'Custom Defined Behaviours' section contains the following code:

```
/* Macro to emit glove80 */
macros {
  glove80_macro: glove80_macro {
    label = "glove80_macro";
    compatible = "zmk,behavior-macro";
  }
}
```

The 'Notes' section contains the following text:

Layout demonstrating how to define a macro using Custom Defined Behaviours. Layer 1 LH C1R5 is mapped to &glove80\_macro which is defined in Custom Defined Behaviours.

For more information, please see ZMK documentation at

At the bottom of the interface, there are buttons for 'Clone to Edit', 'Build Firmware', and 'ZMK Keymap'. The footer contains links for 'MoErgo Website', 'Terms Of Service', 'Privacy policy', 'Contact us', and 'About this app', along with the copyright notice: '@Copyright 2022 Innworks Development Limited (trading as MoErgo)'.

## Title

A human readable description of the layout.

## Tags

Each layout has one or more tags. Tags are descriptive keywords that help yourself remember and to help other people find your layout. A good tag should describe a key distinguishing attribute of the key layout, perhaps describing the application, OS, game that the layout is designed for.

A tag must start with a lower case, and consists of only lower case, numbers and dashes.

Some good examples of tags include:.



OS:

- linux, mac, android

Layout:

- azerty, colemak, f11-f12

Language:

- japanese, german

Software or games:

- javascript, vim, vscode, dota2

Others:

- macro, custom-defined-behaviors

## Layer selector

Glove80/ZMK supports the concept of multiple layers. When you power on the keyboard, the base layer (Layer 0) will be active. However, you can switch to different layers; thus, the same key can serve multiple purposes. This isn't as confusing as it may sound, it's similar to the Fn key on a laptop. There are two ways of switching layers: You can switch to another layer momentarily (while you hold down a trigger key, à la "Fn") or indefinitely (until you switch layers again).

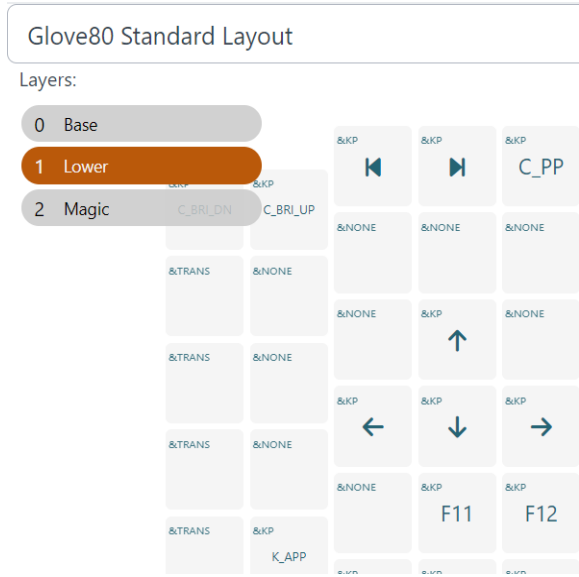
When shipped, Glove80's default key layout has 3 Layers:

- Base layer (Layer 0)
- Lower layer (Layer 1)
- Magic layer (Layer 2)

ZMK documentation for layers are at <https://zmk.dev/docs/features/keymaps>

On Glove80 Layout Editor, the layers are represented on the left hand side. The current layer shown on the layout is the layer highlighted.

When you hover over the circles, the full names of the layers appear and you can select a particular layer to show.



## Notes

This optional section is for yourself and others to better understand the intention of the layout.

## Custom Defined Behavior

This optional section is for injecting your own code into the generated keymap DTSL file. **This is advanced functionality; please use it with extreme caution.**

For more information please see [Advanced usage: Custom Defined Behaviors](#).

## Key binding & key behaviors

Each key in each layer can be bound to a key behavior.



The upper left corner shows the behavior type.

The middle shows the keycode or the behavior parameter, depending on the type of behavior.

## Behavior types

Glove80 Layout Editor currently supports the following ZMK behavior types directly.

- `&kp`: key press
- `&bootloader`: enter bootloader mode
- `&bt`: bluetooth
- `&cap_word`: capitalize words
- `&key_repeat`: repeat key
- `&kt`: key toggle
- `&lt`: layer tap
- `&mo`: momentary layer
- `&mt`: mod tap
- `&none`: do nothing
- `&output`: output selection
- `&reset`: reset keyboard
- `&rgb_ub`: RGB underglow control
- `&sk`: sticky key
- `&sl`: sticky layer
- `&to`: to layer
- `&trans`: transparent

For more information on the ZMK key behaviors, please see ZMK documentation at <https://zmk.dev/docs>

In addition to the above standard ZMK behaviors, Glove80 also supports the following pseudo behaviors:

- `&bt_0`: select bluetooth profile 0
- `&bt_1`: select bluetooth profile 1
- `&bt_2`: select bluetooth profile 2
- `&bt_3`: select bluetooth profile 3
- `&magic`: tap to show RGB indicators. Press to momentarily switch to layer 2 (which is magic layer in the factory layout)
- `&lower`: hold to momentarily switch to layer 1 (lower layer in the factory layout). Double tap to persistently switch to layer 1.
- `Custom`: A special pseudo behavior to allow arbitrary key mapping, to be used with Custom Defined Behavior. For more information please see [Custom Defined Behavior](#).

## Keypress behavior: modifier functions

The keypress (`&kp`) behavior in ZMK has a very powerful but not-well-known capability known as the modifier function. Modifier functions make it possible to add one or more modifiers to a keycode.

ZMK documentation for modifier functions are at <https://zmk.dev/docs/codes/modifiers#modifier-functions>

Glove80 Layout Editor has built-in support for modifier functions. For example if you type LA as the keycode for an `&kp` behavior, you have selected the Left Alt function. You can then add another key or modifier function.

For example, to define Hyper, you can use `&kp LS (LC (LG (LALT)))`. To do so:

1. Select `&kp` as behavior
2. In the keycode slot, select `LS` as keycode
3. In the keycode slot for `LS` modifier function, select `LC` as keycode
4. In the keycode slot for `LC` modifier function, select `LG` as keycode
5. In the keycode slot for `LG` modifier function, select `LALT` as keycode



```
&KP
LS ( LC ( LG
( LALT ) ) )
```

## `&magic` pseudo behavior

This is a special MoErgo-defined pseudo behavior defined as a macro. For a key bound to this behavior,

- A quick tap will activate the LED indicators. Please see the Glove80 User Guide for more information.
- While holding the key, Glove80 will momentarily switch to layer 2 (the magic layer on a standard layout). After key release, Glove80 will automatically revert back to layer 0

On a standard Glove80 layout, LH C6R6 key is mapped to the `&magic` behavior in all layers.

## &lower pseudo behavior

This is a special MoErgo-defined pseudo behavior defined as a macro. For a key bound to this behavior,

- While holding the key, Glove80 will momentarily switch to layer 1 (the lower layer on a standard layout). After key release, Glove80 will automatically revert back to layer 0
- If double tapped, Glove80 will permanently switch to layer 1 (the lower layer on a standard layout)

On a standard Glove80 layout, LH T3 key is mapped to the &lower behavior in the Base Layer (layer 0).

## Bluetooth &bt\_0 &bt\_1 &bt\_2 &bt\_3 pseudo behaviors

These 4 pseudo behaviors switch Glove80 to the respective designated bluetooth device, if the bluetooth device is currently connected.

Underneath the bonnet, these behaviors are implemented as a macro that sets output to bluetooth, and then sets the bluetooth device to the designated bluetooth device.

As an example &bt\_0 is defined as:

```
macros {
  bt_0: bt_profile_macro_0 {
    label = "BT_0";
    compatible = "zmk,behavior-macro";
    #binding-cells = <0>;
    bindings
      = <&out OUT_BLE>,
        <&bt BT_SEL 0>;
  };
};
```

## Bluetooth clear all profiles &bt BT\_CLR\_ALL

The &bt behavior has been extended with an additional parameter BT\_CLR\_ALL

When bound to &bt BT\_CLR\_ALL, the key will initiate clearing all Bluetooth bindings (excluding the binding to the Glove80 right hand).

## Sharing layouts

To share a saved layout simply copy the URL to forward to the person to share with.



# Printing layouts

When you are learning a new layout, it is often useful to print a copy of the layout on paper. Glove80 Layout Editor supports this functionality.

You can print a layer by:

1. Selecting the layer you want to print
2. Using the browser print function to print the layer.  
On a Windows browser this is typically activated by `Ctrl-P`

# Building firmware

This functionality requires logging into your account.

Glove80 Layout Editor provides a quick and simple way to build firmware. Unlike the traditional ZMK approach, there is no need to create a Github repository, nor to install a compiler toolchain on your computer.

To build firmware for a layout you are viewing, click on the Build Firmware button.

To build firmware for a layout you are currently editing, click on the Build Firmware button. Upon successful generation of firmware, a copy of the layout will be automatically saved and you will be redirected to view the saved layout.

## Loading the firmware onto the Glove80

Please see the Glove80 user guide for details.

## Limits on number of builds

To ensure that everyone has a fair opportunity to build firmware through the Glove80 Layout Editor, there are daily and weekly limits on the number of builds a user can build.

It is unlikely that you will ever reach these limits. However if you do reach the limit, please wait a while.



# Advanced usage: Custom Defined Behaviors

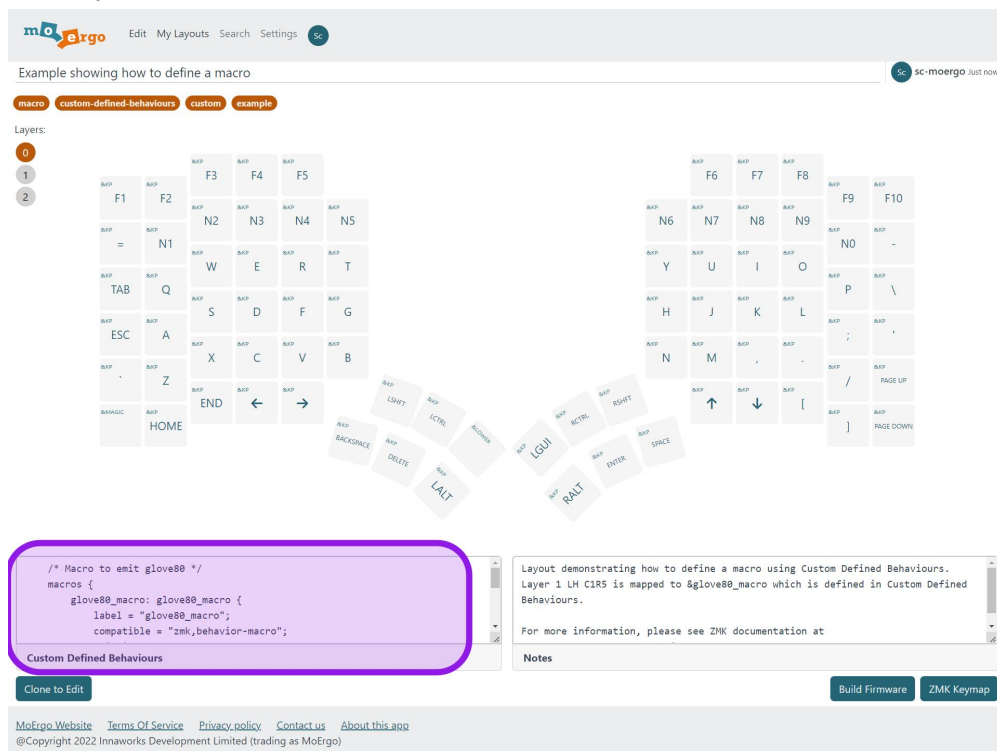
The current version of Glove80 Layout Editor does not support every ZMK feature directly. Missing features include combo and macros.

However these functions can still be supported via Glove80 Layout Editor's Custom Defined Behavior feature.

This is an advanced feature that allows a layout designer to inject text into the keymap DTSI file. It is very powerful and also requires a good understanding of the ZMK DTSI keymap file.

## Custom Defined Behaviors

The layout has a section named Custom Defined Behaviors. This is where you define the text to inject into the keymap DTSI file.



The screenshot shows the Glove80 Layout Editor interface. At the top, there is a navigation bar with the MoErgo logo and options for 'Edit', 'My Layouts', 'Search', and 'Settings'. Below this is a search bar and a user profile 'sc-moergo JUST NOW'. The main area displays a keyboard layout with various keys labeled (F1-F10, N1-N9, A-Z, etc.). A 'Layers' section on the left shows three layers, with layer 0 selected. Below the keyboard layout is a 'Custom Defined Behaviors' section, which is highlighted with a purple box. This section contains a code editor with the following text:

```
/* Macro to emit glove80 */
macros {
  glove80_macro: glove80_macro {
    label = "glove80_macro";
    compatible = "zmk,behavior-macro";
  }
}
```

Below the code editor is a 'Notes' section with the following text:

Layout demonstrating how to define a macro using Custom Defined Behaviours. Layer 1 LH C1R5 is mapped to &glove80\_macro which is defined in Custom Defined Behaviours.

For more information, please see ZMK documentation at

Build Firmware ZMK Keymap

For example, you could try the following text to define a macro named `&glove80_macro`. This macro will “type” `glove80`.

```
/* Macro to emit glove80 */
macros {
  glove80_macro: glove80_macro {
    label = "glove80_macro";
```

```

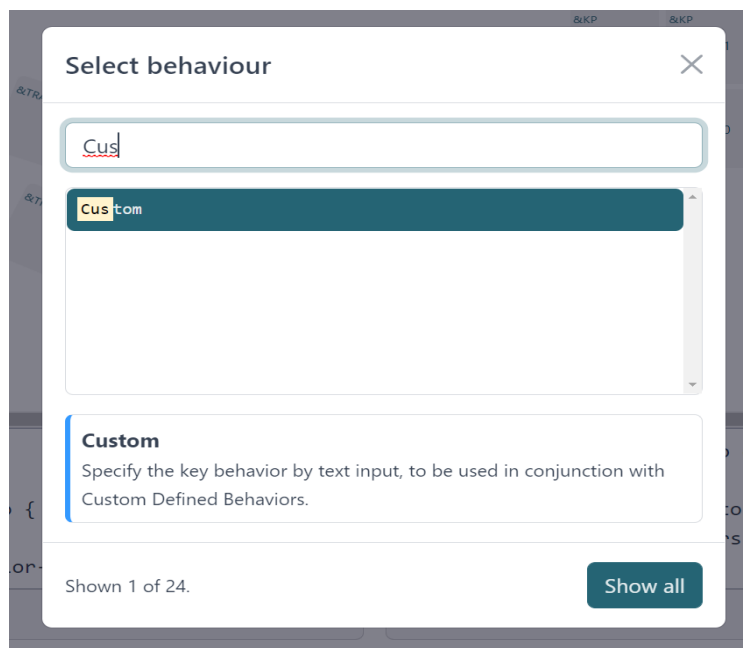
compatible = "zmk,behavior-macro";
#binding-cells = <0>;
wait-ms = <40>;
tap-ms = <40>;
bindings
= <&kp G &kp L &kp O &kp V &kp E>
, <&kp N8 &kp N0>
;
};
};

```

## Key binding for Custom Defined Behavior

To bind a key to a Custom Defined Behavior you have defined:

1. Click on the top left of the key to select the behavior
2. Choose Custom



3. Then in the parameter type the actual binding.  
For example, if you want to bind to a macro named `&glove80_macro`, then type the binding as `"&glove80_macro"`.

