BT60 Guide

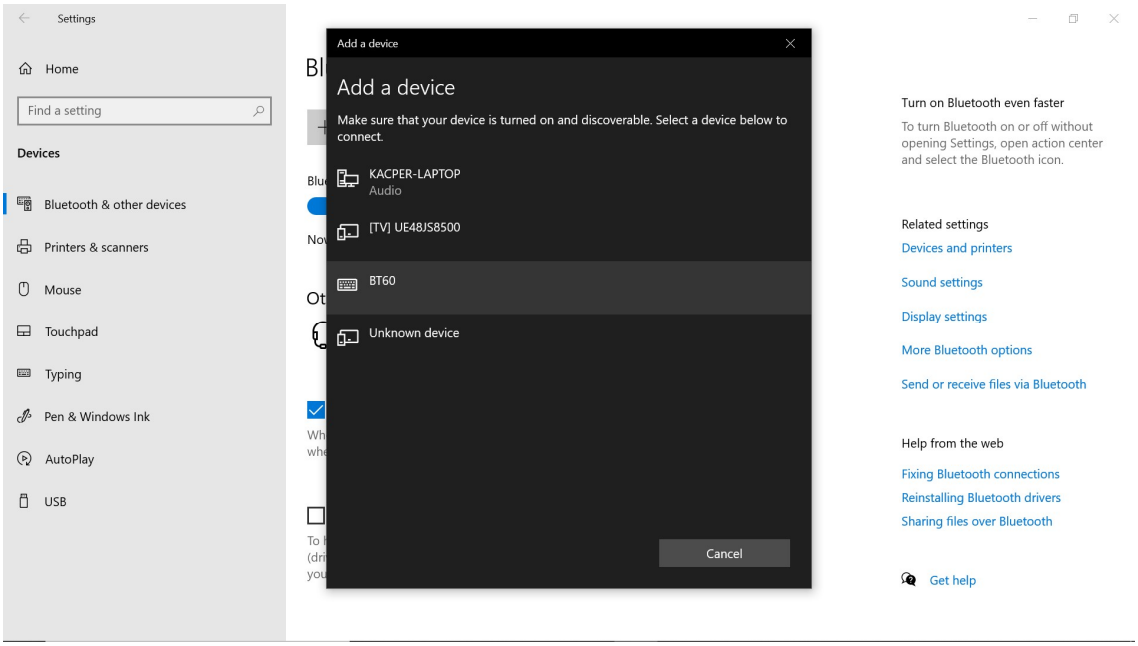
Thank you for supporting Polarity Works and we hope you enjoy your new BT60!

Setting up your BT60

1. Pairing

The BT60 is supplied powered on and ready to pair, if you bought more than one it's recommended to unplug the battery on unintended ones prior to pairing, use a fingernail or small flathead screwdriver to pry the white battery connector out of the socket. DO NOT PULL THE PLUG BY THE WIRES. This can damage the wires/battery. We also recommend unplugging the battery prior to assembling the soldered unit. The connector is keyed and will only plug in one way, forcing it in the wrong way can damage both the connector and the circuitry, if soldering your own battery please ensure the polarity is correct, it is marked on the PCB

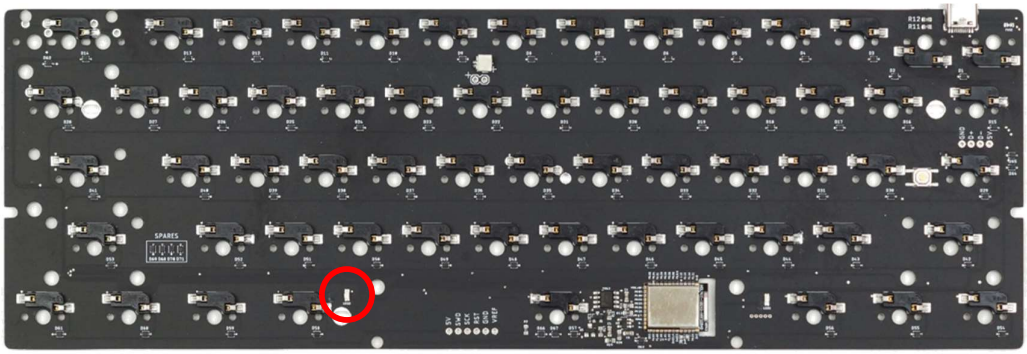
The BT60 is compatible with BT4 or BT5 with BLE. Should your computer lack compatibility we recommend the TP Link UB4A. The drivers aren't great; however, it was rock solid when we were testing each board. To pair in Windows go to Settings – Devices – Add Bluetooth and other devices – Bluetooth. The keyboard advertises as "BT60".



Click on it to pair, it should say "connecting" and a short while later it will say "Your device is ready to go". Hit some keys to make sure it's connected and it should appear with battery charge in the devices section. If you're having trouble check the troubleshooting section.

2. Charging

The BT60 charges whenever connected to USB so it's always ready for when you disconnect. There's a red led on the underside of the board that illuminates when the battery is charging or when the battery is disconnected.



It's dimly visible inside even a metal case. A full charge from flat should take about 3 hours using the included battery, and it should last for months.

Customising your BT60

By default, both the soldered and hotswap versions come with an unmodified ANSI firmware

3. Setting up

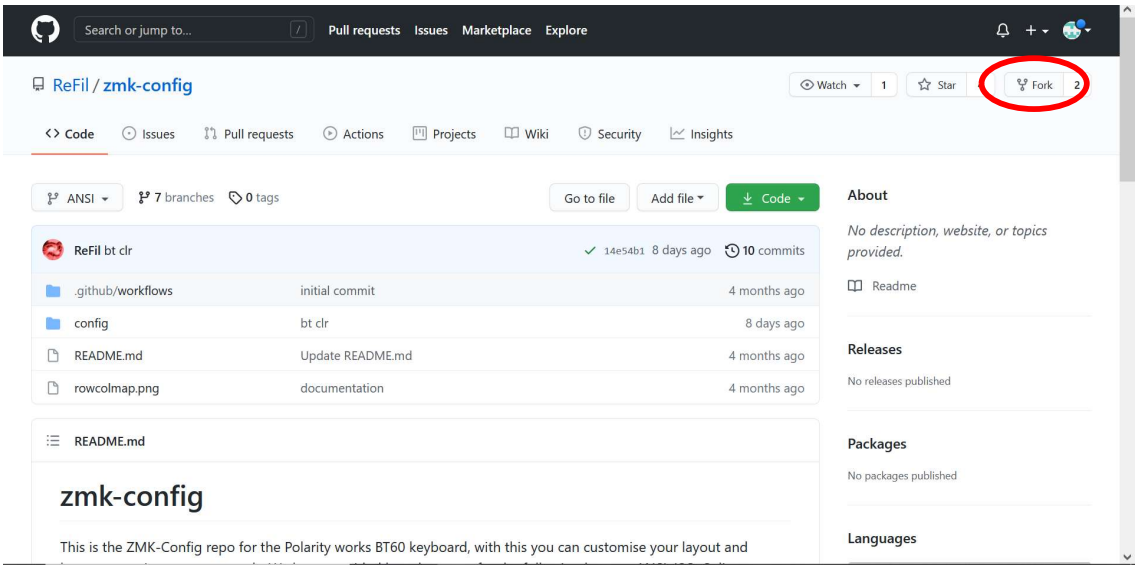
We're using the very versatile ZMK-config system, which allows us to rebuild custom firmware without needing to setup any build environments, saving time, effort and disk space.

a. Create a GitHub account

To setup a GitHub account you need only an email and to solve their CAPCHA puzzle. Follow this link to get to the signup screen: https://github.com/signup.

b. Fork the template repo we provide

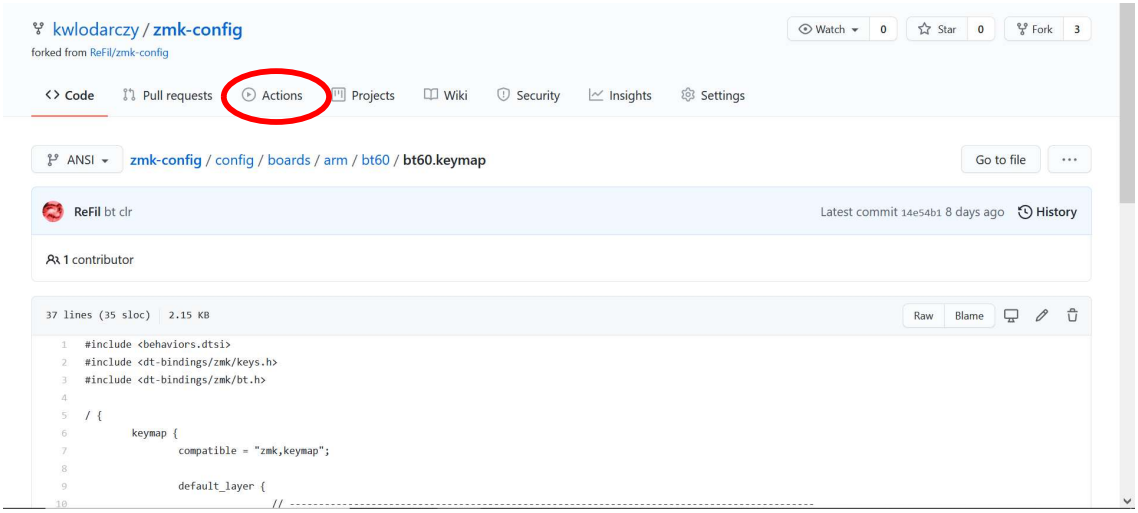
We have example firmware for the ANSI layout of the hotswap and for several common other 60% layouts, follow this link: https://github.com/ReFil/zmk-config, and click the fork button.



In your fork of the repository proceed to click the drop-down menu that says "ANSI" and then select the layout of your choice or hotswap. GitHub has a built-in text editor to handle online editing so you don't need to download anything at all.

c. Enable the build workflows

We have setup GitHub actions to automatically build the firmware in the cloud saving you significant time and trouble setting up a local build environment. Click on the “actions” tab after saving changes and you should see a message saying “workflows aren’t being run on this forked repository”. Click the green “I understand” button to enable the automatic building actions



4. Edit the configuration

When altering the configuration of the BT60 there are two files that are important, the bt60.keymap contains the actual key bindings and the bt60.dts file which defines the physical layout of the keys. They can be found in config/boards/arm/bt60

bt60.keymap

The bt60.keymap file contains all the layers with the keys they correspond to, if you have encoders set up you map them here too. ZMK uses a behaviours principle, whereby each physical keypress triggers a behaviour (denoted by &). Some of these behaviours trigger a keypress event (&kp), or an auxiliary control such as Bluetooth. The full list of behaviours is documented here : https://zmk.dev/docs#! ZMK has all the keycodes documented here: https://zmk.dev/docs/codes/. Each layer needs to have the right number of keys to match the physical layout or else the build will fail.

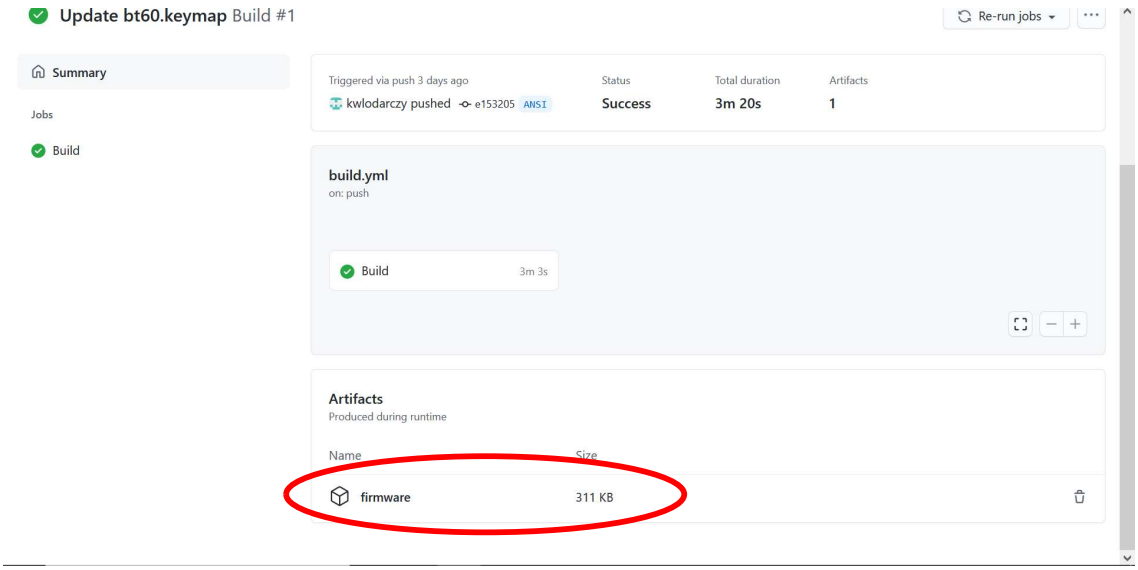
bt60.dts

The bt60.dts file has the physical layout of the keyboard with other configuration options as well. The image in the repo dictates which columns and rows the keys connect to for the various layouts. Of note is the image is 1 indexed whereas in the matrix transform it starts at 0. The matrix transform documentation is here: https://zmk.dev/docs/development/new-shield/#optional-matrix-transform, to add new keys put the RC(X,Y) in the appropriate place in the matrix transform. Invalid keys might not fail to build but will cause unexpected behaviours

When finished editing click the green “save changes” button. This will “commit” and “push” the changes you’ve made to your fork.

5. Building

Click on the “Actions” tab after saving changes and you should see a new action building. There will be an orange progress indicator. Click on it and you will see the progress of the building action. The build process typically takes 10-15 minutes. If the build succeeds you will see a green tick mark in the place of the orange dot, and if it fails there will be a red cross, click on “build” and you’ll see where it failed, typically in the “West Build (bt60)” section. In the first ~10 lines it’ll display the problem, including the line number in the file “bt60.dts.pre.tmp”. This file can be viewed in the drop down below west build (“bt60 DTS File”). Go to the relevant line number and you will usually see the problem, typically either an error in the behaviour definition (capitalisation is important!) or a keycode error. Edit the keymap or dts file to fix the problem and rebuild. Once the build succeeds you’ll be able to download the firmware from the action summary page (it’s under the artifacts section).



6. Flashing

Unzip the firmware.zip file and you will find a file called bt60.uf2. The BT60 uses a variant of the Adafruit nrf52 bootloader which allows software-less flashing. To get your BT60 into bootloader you can use a key combination or press the reset button on the underside of the board twice, we recommend the key combination as the button we chose to maintain case compatibility proved finicky in testing. On the default firmware you hold down the mod button (bottom row, third in from the right) and press the enter key.

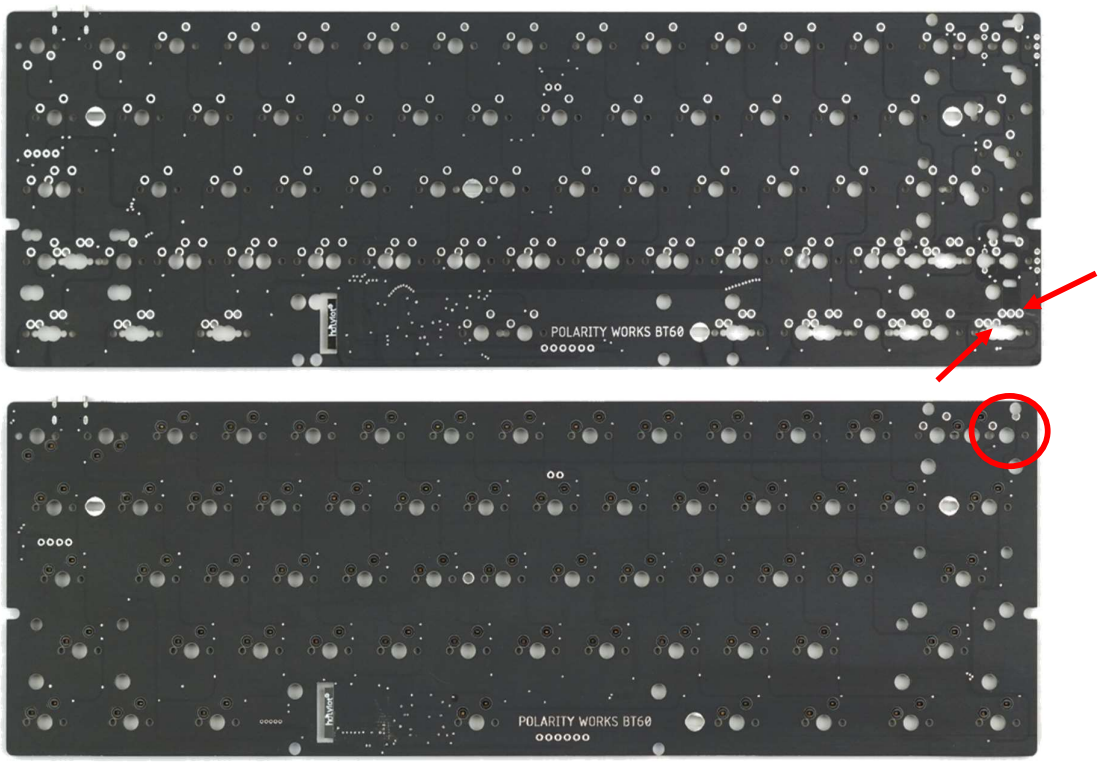
At this point the green led on the underside will start blinking, and the keyboard will show up in windows as a USB drive with the name “BT60” and three files on it. Drag and drop the bt60.uf2 file you downloaded onto the board and let it copy. Sometimes Windows will report an error message, but this doesn’t necessarily mean the flashing has failed. The green led will extinguish once flashing has completed, we recommend unplugging and reconnecting the board and it should reappear in windows as a keyboard. Of note is paired devices are not forgotten between flashes, so don’t unpair the keyboard on your computer after flashing it, or you’ll need to clear Bluetooth devices.

Troubleshooting your BT60

We tried to make the experience as painless as possible but inevitably some people may encounter issues. The best place to get in touch with us if this guide doesn’t help is our discord. Follow the QR code on the business card included in the bag or this link: https://discord.gg/nBq8TShwP2

Bluetooth

Bluetooth LE can sometimes be funky on Windows, occasionally you can get pairing and connectivity issues. If you get an issue pairing first step is to clear the Bluetooth profile on the keyboard, in ZMK this is known as BT_CLR. There are two ways of doing this: on both PCBs it is bound to mod + RCTRL, but we also bound it to some pads on the boards. On the soldered the bottom furthest right key pads and on the hotswap it’s the top right switch pads.



We recommend shorting the pads with tweezers or a paperclip. After running BT_CLR you should ensure the PCB is unpaired from the computer as well, it can cause problems if not. Try pairing again and it should succeed, sometimes you need to hit the clear button a few times or reboot the keyboard by either pressing the reset button (mod + \) or the physical button on the underside or by removing and reconnecting the battery.

Flashing

We've yet to encounter issues, during testing each board was flashed multiple times with different firmwares from several different Linux. If you encounter problems the best bet is try and flash it again, it's most likely a little glitch. The Adafruit nRF52 bootloader is very common and well tested so it's very stable. Other possible causes for flashing errors include a corrupted uf2 file (try flashing an unmodified firmware found on our repo) or a dodgy type c cable (try a different one).

Customising

If you're having trouble with ZMK even when following the docs, you can reach out in our discord or join the official ZMK discord and ask there, the typical issues are slight typos in the keycodes or incorrect behaviours. Capitalisation is very important in the code.