
keypads Documentation

Release 1.0

thnikk

Aug 23, 2017

Contents

1 Pages

3

Welcome! This is the documentation for all current models of my keypads.

Here you can find feature lists for models, guides to install drivers, remap the keys, and change your switches. This will be updated with my code so it should always be up to date!

Buyer's Guide

A lot of people have had a lot of questions about my keypads for determining which model and what parts would be better for them, so I'll outline the differences here.

Switches

This may be the most important choice you can make with your keypad. Luckily, you aren't stuck with your choice thanks to the switch changing kit (unless you buy the LED model) so don't feel too stressed out. It's easy to say it's "personal preference," so I'll instead try to give you some objective information that you can base your decision on.

Clicky

These have both a bump and an audible click when the switch is pressed to its actuation point. This is nice for typing (especially touch typing) since it's a clear indication that you've pressed the switch. It's great for avoiding "bottoming out," meaning the switch is fully depressed, which can help reduce fatigue from typing over long periods of time.

However, playing *osu!* is not the same as typing. The game is much more hectic and if you try to feel or listen to the click, you will be pulling your attention away from the game. Some players may be able to take advantage of the click, but I consider it to be very disadvantageous. It would be a nice option for a macro keypad if you want to use it for professional applications, but be mindful if you want to use it with other people around.

Tactile

These offer the same benefit that blues do for typists, but the same downside for *osu!* players. They have the same bump at the actuation point but no audible click. These would also be nice for professional applications and are better in quiet environments.

Linear

Linear switches offer a nice variety in weight (actuation force) and don't have any audible click or tactile bump. I would consider these the best for osu! and gaming use in general.

Actuation Force Table

Switch Type	Actuation Force	Cherry MX	Gateron
Linear	35g	—	Clear
Linear	45g	Red	Red
Linear	65g	Black	Black
Clicky	55g	Blue	Blue
Tactile	50g	Brown	Brown

Cherry MX VS Gateron

One other thing to mention is the difference between Gateron and Cherry switches. Cherry is a German company that's been making these for a long time, but their patent ran out recently so a bunch of other companies started making MX clones. Among those companies is Gateron. They are considerably smoother feeling, where Cherry's switches have a bit of a scratchy feel to them.

Keycaps

DCS

DCS keycaps are what you'll most commonly find on mechanical keyboards. They feature a subtle curve vertically that your finger can rest in and an angle depending on the row. They are comfortable for both typing and osu, but you have to be a bit more precise with your button presses.

DSA

DSA keycaps are a little more rare but offer a completely flat profile across all keyboard rows. This makes it easy to get a whole bunch of blank ones since all 1x keycaps are the same, unlike the DCS keycaps that differ by row. They have an old-fashioned look to them and feature a slight concave curvature on the top of the key. They work well for osu but less for typing on a keyboard since the flatness of them makes them less ergonomic.

Flat

Flat keycaps are the least common by far. I bought a few from pimpmykeyboard.com a year or two ago and loved them, but they were a bit too expensive so I figured nobody would want to pay \$4+ for them. Fast forward to 2017 and I have a bit more experience in 3D printing. Now I can offer them not only as a cost saving measure to both you and myself, but also be happy that they work well and provide a very nice experience in osu.

These offer two major improvements over DCS and DSA keycaps. They expand the surface area of the top of the key to their maximum, giving you more leeway with where you press. They also allow me to have better control of both the travel distance and the max height of the keypad, which means that your hand won't have to rest at as high of an angle which should result in a more comfortable experience over long sessions.

Quickstart Guide

Contratulations on receiving your keypad! This quick guide will walk you through how to use your keypad.

What's in the box:

- Keypad
- Keycaps
- Keychain*
- Cable

*Keychains are given when I have them. You may not get one or it may look slightly different from the picture.

Before use...

There are two things that should be done before you use your keypad.

Installing the Keycaps

To install the keycaps, put them on top of the switches in the correct orientation and push them down.

O-rings come pre-installed on the keycaps so there is no need to isntall them manually. If you don't like the soft feel and reduced travel of the keys, they can be removed.

Remove Tape Film

The tape on the bottom of the keypad will fix it to your desk so it doesn't move around during play. This tape has a film covering it that must be removed to expose its stickiness.

More (Optional)

These are some other things you can do after you get your keypad up and running.

Setting the LED mode

While the basic model's side button is used exclusively as an escape key, it serves an additional purpose on all models with LEDs. By holding the button down and waiting for the LEDs to blink, you can change the LED mode. Wait for it to blink twice for the secondary action, which is for changing the brightness (or the colors on the custom mode.) For more info on LED modes and the side button for your keypad, please visit the models page here:

[Models](#)

Remapping and Reprogramming

If you're unhappy with the default zx(cv) button mapping of your keypad, visit this page to learn how to easily remap the buttons!

[How to remap the buttons on your keypad.](#)

If you'd like to change the functionality of the keypad, I have a reprogramming guide that you can visit [here](#).

[How to reprogram your keypad.](#)

If you wish for any additional functionality, I would appreciate it if you opened an issue on my [GitHub](#).

[GitHub](#)

Changing Your Switches

If you're unhappy with your original choice of switches, a switch changing kit can be purchased from my shop! A guide for how to use the kit can be found [here](#).

[Switch Changing Kit](#)

Keypad Models

Basic Keypad

This is the no bells and whistles model. 2 Keys and a side button. The side button functions as an escape key and offers no additional functionality.

LED Keypad

This is the LED model. It features a white LED on each of the two face keys. The side button functions as escape and is also used to change LED modes and brightness.

Side button functionality

Hold time	Functionality
Less than 0.5s	Escape
0.5 - 1.5s	Change LED mode
more than 1.5s	Change brightness (Press Z to lower and X to raise)

LED modes

Order	Mode
1	Reactive - Turns LEDs on when pressed and fades out when released.
2	Push Off - Turns LEDs off when pressed and fades on when released.
3	Breathe - Fades LEDs in and out.
4	On - LEDs stay on.
5	Off - LEDs stay off.

RGB Keypad

This is the RGB model. It features an RGB LED under each of the two face keys. The side button functions as escape and is also used to change LED modes, brightness, and the colors of the keys on the “custom” LED mode. Since it uses LEDs mounted underneath the switches, Gateron switches are the only option since their semi-transparent housing allows this to work and works perfectly for diffusing the light of the LEDs.

Side button functionality

Hold time	Functionality
Less than 0.5s	Escape
0.5 - 1.5s	Change LED mode
more than 1.5s	Change brightness (Press Z to lower and X to raise)
more than 1.5s (on custom LED mode)	Change color per-key

LED modes

Order	Mode
1	Cycle - Fades through rainbow.
2	Reactive - Turns LEDs on when pressed and fades from red to green to blue to off when released.
3	Push Off - LEDs are normally white and fade from red to green to blue to off when the keys are pressed.
4	Custom - LED color can be set per-key.
5	BPM - LEDs turn white when pressed and change color depending on how much you hit them per second when released.
6	Color Change - Colors cycle in steps per-press per-key.

4K Keypad

This is the 4k model. It’s the same as the RGB model but with 4 keys instead of 2. The side button functions as escape and is also used to change LED modes, brightness, and the colors of the keys on the “custom” LED mode. Since it uses LEDs mounted underneath the switches, Gateron switches are the only option since their semi-transparent housing allows this to work and works perfectly for diffusing the light of the LEDs.

Side button functionality

Hold time	Functionality
Less than 0.5s	Escape
0.5 - 1.5s	Change LED mode
more than 1.5s	Change brightness (Press Z to lower and X to raise)
more than 1.5s (on custom LED mode)	Change color per-key

LED modes

Order	Mode
1	Cycle - Fades through rainbow.
2	Reactive - Turns LEDs on when pressed and fades from red to green to blue to off when released.
3	Push Off - LEDs are normally white and fade from red to green to blue to off when the keys are pressed.
4	Custom - LED color can be set per-key.
5	BPM - LEDs turn white when pressed and change color per-key depending on how much you hit them per second when released.
6	Taiko - Color scheme for Taiko (orange on outside and blue on inside.)
7	Color Change - Colors cycle in steps per-press per-key.

Macropad

By popular request, I made a keypad focused less on crazy lights and more on people that want the ability to quickly switch between button mappings. This model features not only the ability to map each key to up to 3 keys, but also six “pages” of mappings that you can change by holding the side button and pressing one of the six keys. It also has a single RGB LED to show the current page.

Side button functionality

Hold time	Functionality
Less than 0.5s	Escape
More than 0.5s	Change mapping page (1-6 depending on face button pressed)

Default button mapping

Page	Keys
1	A, S, D, Z, X, C
2	Q, W, E, A, S, D
3	blank, up, blank, left, down, right
4	1, 2, 3, 4, 5, 6
5	all blank
6	all blank

Future Tone Controller (coming soon)

This is the small controller I made for Hatsune Miku: Project Diva Future Tone for PS4. There is a mode select button on the front of the controller that allows you to switch between DS4 and arcade mode. DS4 mode maps the buttons to the standard DS4 layout and arcade mode maps the keys in a horizontal layout like you’d get on the arcade cabinet. This makes hold notes a lot more natural without sacrificing speed since the arrow buttons are all adjacent to their corresponding buttons. When the mode is changed, the LEDs change color to match the corresponding layout.

Driver Installation

Windows 8.1 or lower

Before we get started with any remapping or programming, you'll need the driver for the keypad installed. All of my keypads use the same microcontroller, so this procedure will work for any current keypad mode.

Download Drivers

Download and extract the zip to your desktop and open "dpinst-amd64.exe." Follow the wizard and say yes when it asks. That's it!

Windows 10, Mac, and Linux

No driver installation required. You're done, great job!

Remapping Your Keypad

Windows

If you are on Windows 8.1 or lower and you haven't already, follow the driver installation guide first.

Driver Installation

Once the drivers are installed, download and extract Termite. You want to make sure you extract since you need to run it with the .ini file in the same directory.

Download Termite

- Package updated 1/19/17
 - Changed font to monospaced
 - Enabled setting to keep connection active when window isn't focused
 - Enabled word wrap

Upon opening the program, it should immediately connect to the keypad. Follow the directions in the remapper and you're all set!

Make sure to close Termite when you're done, otherwise it may try to reconnect and lock up your keypad.

Mac and Linux

The procedure for Mac and Linux is the same since we'll be using the screen command on both. As we do on windows, you'll first need to find the com port the keypad is using. You can do so by entering this into the terminal:

```
dmesg | grep tty
```

If you just plugged the keypad in, it should be the last thing listed. In my case it's using /dev/ttyACM0. Now we can start serial communications using screen.

```
screen /dev/ttyACM0 9600
```

That's it!

Universal

You can also use the Serial Monitor included in the Arduino IDE. This can be done by downloading the IDE through the Arduino website here:

[Download](#)

Follow the wizard for installation and launch the program. From here you can select the port from Tools > Ports and open the Serial Monitor by clicking the magnifying glass icon in the top right of the window.

Programming Your Keypad

The Easy Way (Web Editor)

Arduino recently introduced their web-based editor, a more accessible version of the Arduino IDE that can run in your web browser. This is great for me because it's a lot easier than installing the program, libraries, copying and pasting code, selecting the port, selecting the board, etc. Don't worry about any of that, since we get to avoid all of it!

Requirements

Browser: Chrome, Firefox, or Safari

Arduino account

Making an Arduino Account

Don't worry, they won't bombard you with emails (yet*) and you don't need to give them any personal information, they just require an account so you can manage your code. You can make your account from this page here:

[Sing Up](#)

You will have to verify your account through your email (don't use a fake one) and then you're done!

Installing the browser plugin

Go to the [Web Editor](#) and you will be asked to install the plugin.

*On the welcome screen there's a box you have to uncheck if you don't want their emails:

Drivers?

If you're on a Windows machine running Windows 8.1 or lower you'll need drivers for your keypad to be recognized. The Arduino plugin should ask you to install them, but if it doesn't, you can follow my driver installation guide here:

[Driver Installation](#)

Code

You're almost done, all that's left is loading the code. Fortunately, they have made this very easy. All you have to do is click on the share link for your corresponding keypad and add to sketchbook. It should then take you to the editor and tell you that Arduino Leonardo has been connected. You can now click on Upload (the right arrow) to upload the

code to the keypad. If it works, you should see “Success: Done Uploading.” If not, make sure the keypad is plugged in and that you installed the drivers.

[Basic](#)

[LED](#)

[RGB](#)

[4K](#)

[4K RGB](#)

[MacroPad](#)

The Hard Way (Desktop IDE)

If you don’t want to use the Web IDE, you can download the tried-and-true desktop IDE instead. This is a little harder and requires a few extra steps, but you don’t need to make an account.

Driver Installation

If you’re on a Windows machine running Windows 8.1 or lower you’ll need drivers for your keypad to be recognized.

[Driver Installation](#)

Installing the Arduino IDE

The latest version of the Arduino IDE can be downloaded and installed from the Ardiono website here:

[Arduino.cc](#)

Configuring the IDE

A few things need to be done before you start. Firstly, you will need the libraries used in the code, otherwise it can’t compile because all of that code will be missing. This used to be a lot more difficult to install, but since version 1.6.5 they added the Library Manager, which allows you to install libraries directly from the IDE. To install libraries, launch the IDE open the Library Manager by going to Sketch > Include Library > Manage Libraries...

My code uses the Bounce2 and NeoPixel libraries, so just search “Bounce2” and “Adafruit NeoPixel” in the search bar and click Install on the result. After they are installed, you can close the Library Manager window.

Verifying and Uploading the Code

Delete the code in the main window of the Arduino IDE. Now, paste in the code for your corresponding keypad from my GitHub here:

[GitHub](#)

Now we need to tell the IDE what it’s trying to compile for and communicate with. Go to Tools > Board and select Arduino Leonardo and go to Tools > Port and select the one that has (Arduino Leonardo) to the right of it.

Now you can click on the right arrow to compile and upload the code. Don’t worry about uploading broken code, if it fails to compile it won’t upload it to the microcontroller. It will ask you to save and you can either click cancel or choose a directory to save to, it will compile either way. If you didn’t get any errors, congratulations, you did it! If you did, make sure you’ve selected the board and the port and that there’s nothing left behind from the blank sketch or that you copied the code correctly.

Making Changes to the Firmware

Understanding the basics

There are some things you should familiarize yourself with before you get started. These are the most important functions/libraries.

`digitalRead()` for reading button inputs.

`analogWrite()` for controlling single-color LEDs.

The `neopixel` library for controlling RGB LEDs.

You should also know...

for loops

switch/case statements

millis timers

I wouldn't recommend trying to change random variables unless you can actually follow the logic and see exactly what they're changing. Some of them may tie in somewhere else and cause unexpected results.

Changing Your Keypad's Switches

Overview

This guide will walk you through swapping the internals of a Cherry MX compatible switch with another one. This is meant to be simpler than replacing the entire switch, which would require resoldering. This still gives the desired effect but lets things like RGB LEDs be mounted on the back of the switch without having to replace them when you want to change the switch.

What You'll Need

-**A switch changing kit:** Fine-point tweezers, Small spudger (for keeping one side of the switch open), Cherry MX compatible switches

-A keypad

Opening the switches on your keypad

Before starting, I'd like for you to take a close look at the switch you want to open. In the four corners, you should see four rectangular indentations. Along the outside edge of these indentations, there are narrow slits. These slits lead all the way down the clamps of the switches, so when you insert something like tweezers into them, it straightens them out and helps you open the switch more easily. With that in mind, we can now move on.

Insert your tweezers into the slits in the rectangular indentations of one side of the switch. There should be a point where you feel them go down a little bit more than if they weren't in the slits. From here you can push the tweezers towards the center of the switch and one side should pop open!

We'll be repeating the same step for the other side, but use the spudger to hold the open side of the switch open, otherwise it will close when you try to open the other side. Now you can remove the top, stem, and spring of the switch.

Opening the loose switches

This is a little simpler than opening the switches on the keypad since you have access to the tabs that hold it together from the underside. Grip the switch from the top and push the tweezers under the two tabs of one side of the switch. Tilt the tweezers up and one side of the switch should be open.

Opening the other side of the switch should also be easier. I use my fingers to firmly grip the open sides of the switch and repeat the first step. With enough pressure, the open side should stay open. With both sides open, we can now remove the stem and spring and insert them into the switch on the keypad!

These parts do have a correct orientation, so make sure the side of the stem that sticks out is facing the metal leaf in the switch. The top also has a correct orientation. The side with the cavity on the top should line up with the leaf (this only fits one way, so don't force it if the switch isn't closing.)

If all went well, your switch should now be changed!

Troubleshooting Guide

This guide will walk you through steps to either get your keypad back up and running or properly diagnose the problem.

Connection

Keypad won't turn on

This is usually caused by a problem with the USB connection. You can tell if your keypad is receiving power by looking for a faint red glow from the power LED inside.

1. Check your cable and make sure it's plugged in all the way on both ends.
2. Try a different cable.
3. Check your micro-USB port to see if it's still connected to the board. In some extreme cases, it might have been ripped off.

USB device not recognized

This can be caused by a number of things, the most common two being a bad USB connection or a corrupted/incorrect bootloader.

1. Follow the steps for the previous issue
2. If you've recently tried uploading new code, open the keypad and quickly short the GND and RST pins twice. This will force it back into bootloader mode and you should be able to reupload your code. Make sure you have the correct board selected.

Shows up in Device Manager but it still isn't working

1. Make sure you don't have a serial monitor open anywhere that may be falsely triggering the keypad to enter remap mode.
2. Try following the reprogramming guide to get the stock code back on the keypad.

The remapper isn't working with Termite

This is most commonly caused by having the Arduino Create plugin running. You can either disable it through the system tray in the bottom right or you can just use the Arduino Create serial monitor.

Mechanical

One of my switches isn't working consistently

Try removing the o-rings from the keycaps to see if they are preventing the keys from being fully depressed.

Programming

It's not compiling and giving me an error

1. Enable verbose output for compilation and uploading under File>Preferences.
2. Make sure you installed the required libraries (bounce2 and neopixel) under Sketch>Include Library>Manage Libraries...

It won't upload

Make sure that you have the correct board and port selected. You need to select Arduino Leonardo under Tools>Boards and the port should be labeled with (Arduino Leonardo) next to it under Tools>Ports.

Remapping (on Remapper 1.1)

The keys I enter aren't mapping properly

1. Make sure you're entering your modifiers first.
2. Make sure you're only entering three keys (including modifiers.)
3. Enter xx (lowercase) after entering a modifier if you don't wish for any additional keys to be mapped to that key. Otherwise, enter the remaining keys.