Zoom Meetings Physical Mute Button

by elliotmade

If you use zoom meetings for work or school this button is for you!

Press the button to toggle your mute, or hold the button down to leave the meeting (or end it if you are the host).

One great thing about this is that it works even if your Zoom window isn't active... if it's buried under a bunch of spreadsheets and browser windows - no problem - it brings the window to the front and flips your zoom off or on. Quickly un-muting is key to maintaining the impression that you've been paying attention the whole time!

Even better, this all works while you are sharing your screen, so you don't have to do battle with those pesky on-screen controls.

Check the last step for a two-button version that also will toggle your video on and off

https://www.youtube.com/watch?v=apGbelheIzg

Step 1: How It Works

This device simply emulates a keyboard when you plug it into your computer. We take advantage of the built-in keyboard shortcuts for Zoom:

CTRL+ALT+SHIFT brings focus to the Zoom window

ALT+A toggles the state of mute, if you mute is on it turns it off, and if it's off it turns it on

ALT+Q leaves a meeting or ends it if you are the host

These are the keyboard shortcuts for the windows version of the app - I don't have a mac to test this on, but I'm sure a similar thing will work there perhaps with a couple tweaks if they keystrokes are different.

A short press of the button sends CTRL+ALT+SHIFT followed by ALT+A, while a long press sends CTRL+ALT+SHIFT followed by ALT+Q then ENTER.

I used a Digispark clone board (attiny85 microcontroller) and built off of the an example sketch from the Digikeyboard library. I also used this library to deal with the button. I used the Arduino IDE to flash the code below, you will need to add the Digistump boards with the boards manager first.

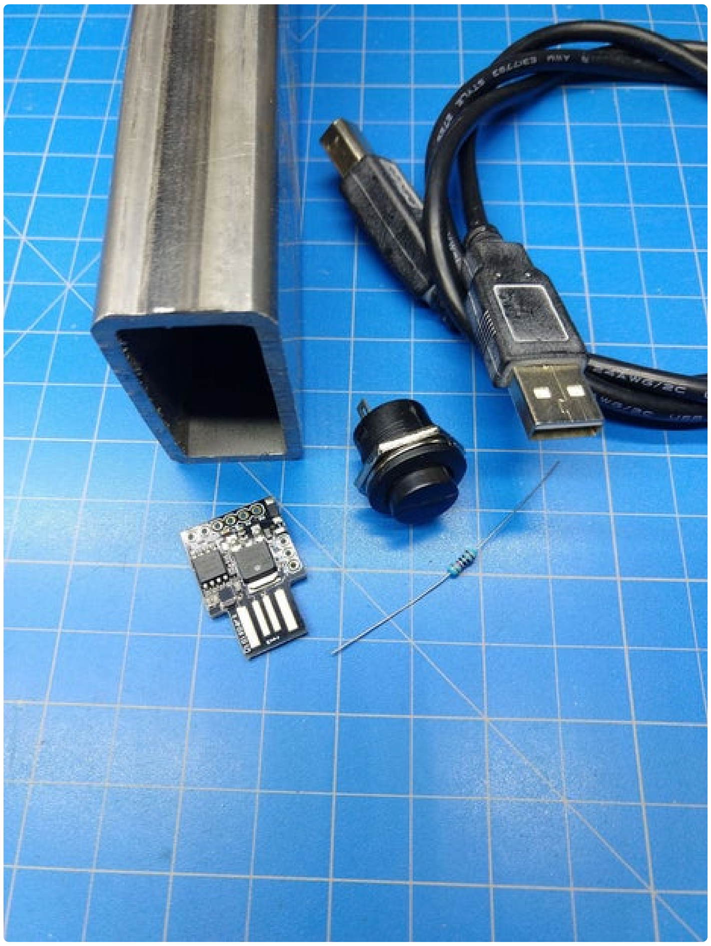
```
//Elliotmade 4/22/2020<br>//https://elliotmade.com/2020/04/23/physical-mute-button-for-zoom-meetings/
//https://www.youtube.com/watch?v=apGbelheIzg
//Used a digispark clone

//this will switch to the zoom application and mute it or exit on long press
//momentary button on pin 0 with pullup resistor

//https://github.com/mathertel/OneButton
//button library
#include "OneButton.h"

int button1pin = 0;

#include "DigiKeyboard.h"

//set up buttons
  OneButton button1(button1pin, true);

void setup() {
  // put your setup code here, to run once:

//set up button functions

  button1.attachClick(click1);
  button1.attachLongPressStart(longPressStart1);

  DigiKeyboard.sendKeyStroke(0);
  DigiKeyboard.delay(500);

}
void loop() {
  // put your main code here, to run repeatedly:
  //monitor buttons
  button1.tick();
}

// This function will be called when the button1 was pressed 1 time (and no 2. button press followed).
void click1() {
  // this is generally not necessary but with some older systems it seems to
  // prevent missing the first character after a delay:
  DigiKeyboard.sendKeyStroke(0);

  // Type out this string letter by letter on the computer (assumes US-style
  // keyboard)
  DigiKeyboard.sendKeyStroke(0, MOD_SHIFT_LEFT | MOD_CONTROL_LEFT | MOD_ALT_LEFT);
  DigiKeyboard.delay(100);
  DigiKeyboard.sendKeyStroke(KEY_A, MOD_ALT_LEFT);

} // click1

// This function will be called once, when the button1 is pressed for a long time.
void longPressStart1() {
  // this is generally not necessary but with some older systems it seems to
  // prevent missing the first character after a delay:
  DigiKeyboard.sendKeyStroke(0);

  // Type out this string letter by letter on the computer (assumes US-style
  // keyboard)
  DigiKeyboard.sendKeyStroke(0, MOD_SHIFT_LEFT | MOD_CONTROL_LEFT | MOD_ALT_LEFT);
  DigiKeyboard.delay(50);
  DigiKeyboard.sendKeyStroke(KEY_Q, MOD_ALT_LEFT);
  DigiKeyboard.delay(50);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);

} // longPressStart1
```

Step 2: Supplies

The core of this is the Digispark microcontroller board and the button, how you assemble this is really up to you. I used a steel tube as the housing for this project because I wanted something with some gravity so that it stays put on my desk. Here is what it took:

- Digispark microcontroller board
- 10k resistor
- Momentary pushbutton
- Wire
- Donor USB cable
- Rectangular steel tube (2" x 1" x 1.5")
- 3mm plywood cut out to fit in the end

I think there are plenty of easy ways to assemble this - you could do it on a breadboard, or 3D print a small housing, laser cut a box, drill a hole in your desk, whatever you want!

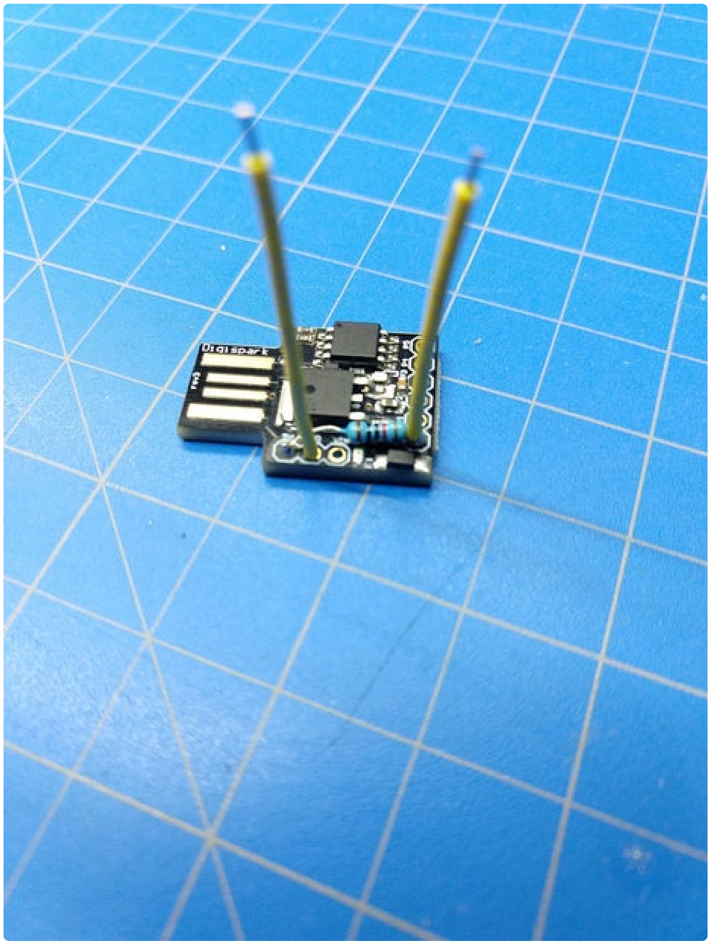


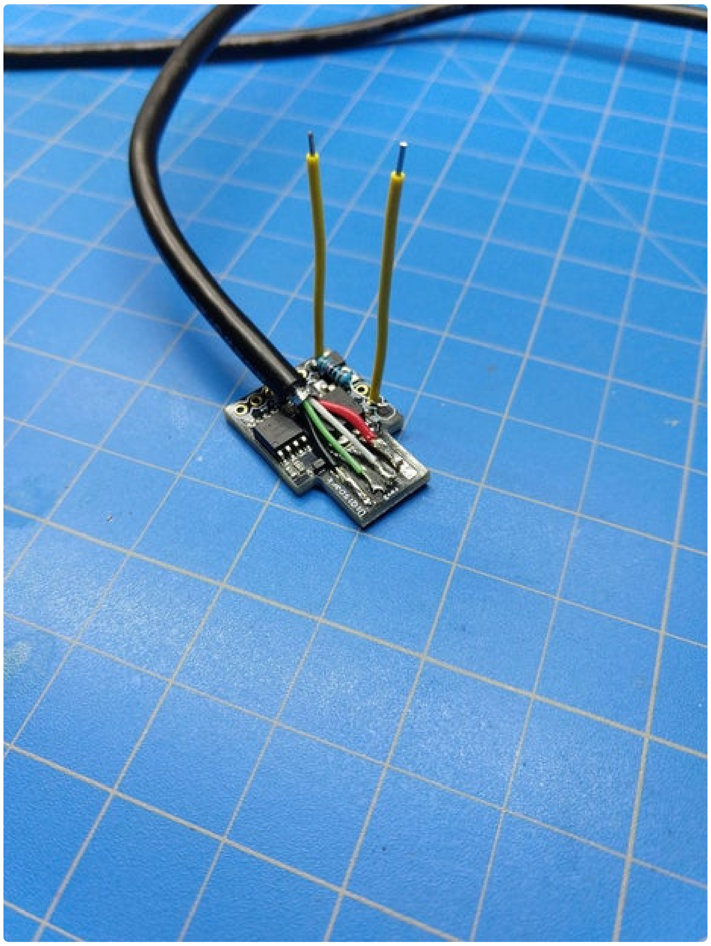
Step 3: Wiring

I included a couple of photos above... if anyone needs a diagram let me know and I can draw it, but it is very simple.

1. 10k resistor between the 5V and P0 pins
2. Wire between GND and one side of the switch
3. Wire between P0 and the other side of the switch

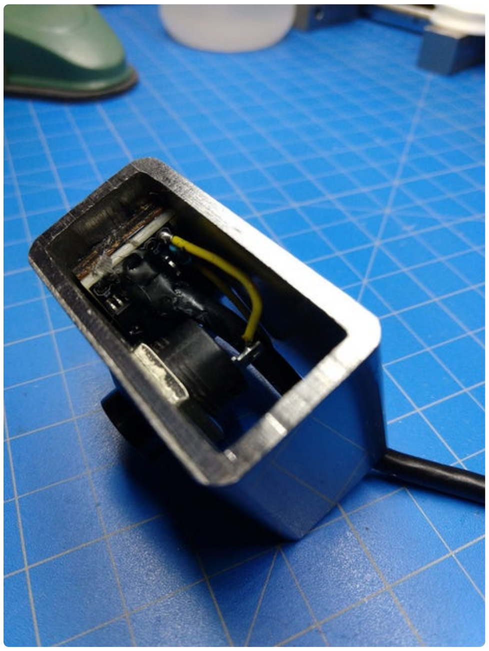
That's all there is to it! You could actually plug it into your computer as is, but I wanted this to be on a wire, so I snipped the end off of an old USB cable and soldered it directly to the pads as shown.

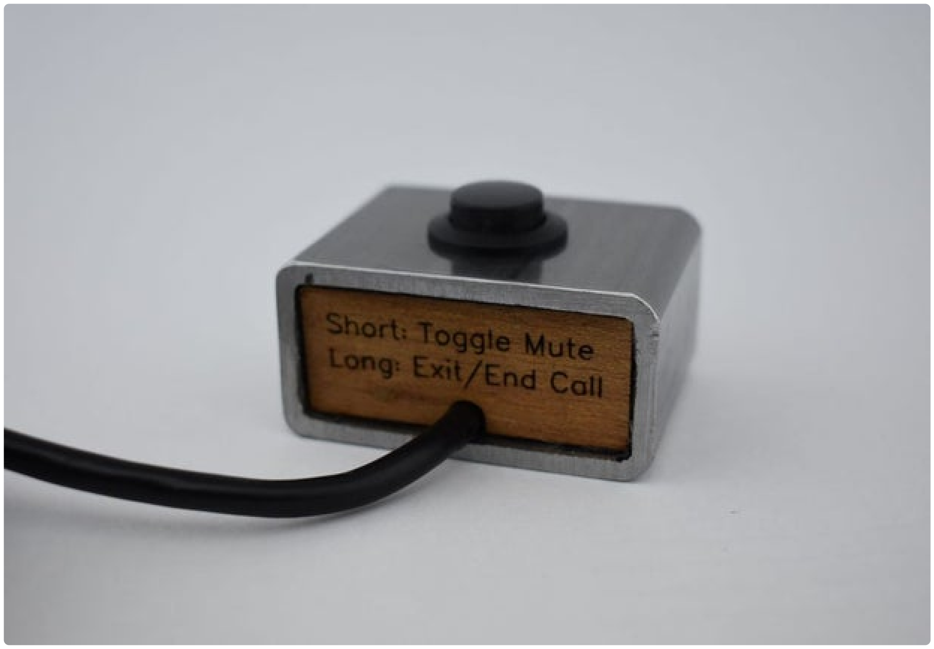




Step 4: Jam It All Together

The photo above doesn't show great detail, but the main idea here is to cram everything into whatever enclosure you decided on. I used hot glue to secure the board and wires inside of the steel tube, then filled in the ends with a small piece of laser cut plywood. The whole thing (except the button) was sprayed with clear coat to prevent rusting, then it was sealed up.





Step 5: Done!

Plug it in to your computer (actually, maybe do this before sealing it up in case you need to troubleshoot the wiring). No drivers are required, it should act like a keyboard right off the bat. Check the video here to see it in action!

I've got a couple extras available on my Etsy store if is something you can't live without.

https://www.youtube.com/watch?v=apGbelhelzg

Step 6: Easy Free Alternative

If you like this idea but don't sit at a desk with room for more stuff, or if you are on the go and don't want to carry something around just to mute yourself, here is an alternative:

Check out autohotkey, I have this script below mapped to my F12 key. It gives me the same mute-toggle function and it's free!

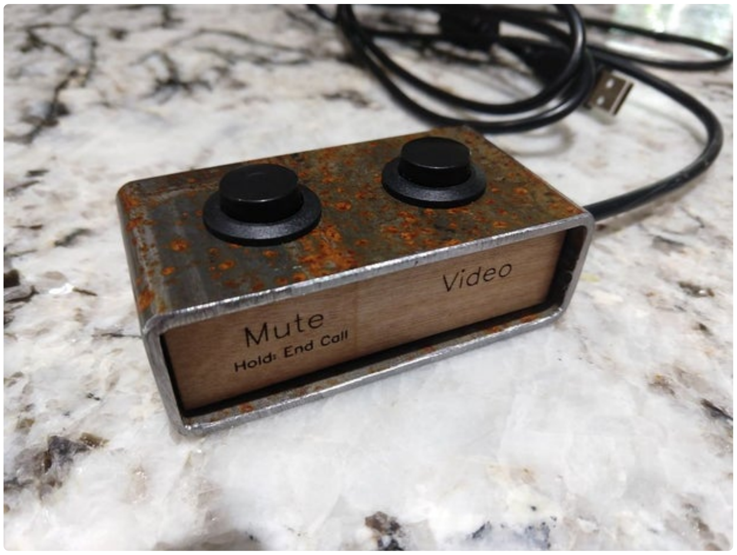
```
#NoEnv  ; Recommended for performance and compatibility with future AutoHotkey releases.
; #Warn  ; Enable warnings to assist with detecting common errors.
SendMode Input  ; Recommended for new scripts due to its superior speed and reliability.
SetWorkingDir %A_ScriptDir%  ; Ensures a consistent starting directory.

f12::
        Send {Ctrl Down}{Shift Down}{Alt Down}{Ctrl Up}{Shift Up}{Alt Up}
        Sleep 100
        Send !a
Return
```

Step 7: Bonus - Two Buttons!

In response to the question below "Do you think you could add a second button for the camera?": Yes! Code is below, picture above. The circuit for this one adds a second switch and pullup resistor on P2.

```
//Elliotmade 4/27/2020
//https://elliotmade.com/2020/04/23/physical-mute-button-for-zoom-meetings/
//https://www.youtube.com/watch?v=apGbelheIzg
//Used a digispark clone

//this will switch to the zoom application and mute it or exit on long press
//momentary button on pin 0 with pullup resistor
//momentary button on pin 0 with pullup resistor

//https://github.com/mathertel/OneButton
//button library
#include "OneButton.h"

int button1pin = 2;
int button2pin = 0;

#include "DigiKeyboard.h"

//set up buttons
  OneButton button1(button1pin, true);
  OneButton button2(button2pin, true);

void setup() {
  // put your setup code here, to run once:

  //set up button functions

  button1.attachClick(click1);
  button1.attachLongPressStart(longPressStart1);
  button2.attachClick(click2);

  DigiKeyboard.sendKeyStroke(0);
  DigiKeyboard.delay(500);

}

void loop() {
  // put your main code here, to run repeatedly:
  //monitor buttons
  button1.tick();
  button2.tick();
}
```

```
// This function will be called when the button1 was pressed 1 time (and no 2. button press followed).
void click1() {
  // this is generally not necessary but with some older systems it seems to
  // prevent missing the first character after a delay:
  DigiKeyboard.sendKeyStroke(0);

  // Type out this string letter by letter on the computer (assumes US-style
  // keyboard)
  DigiKeyboard.sendKeyStroke(0, MOD_SHIFT_LEFT | MOD_CONTROL_LEFT | MOD_ALT_LEFT);
  DigiKeyboard.delay(100);
  DigiKeyboard.sendKeyStroke(KEY_A, MOD_ALT_LEFT);

} // click1

// This function will be called when the button2 was pressed 1 time (and no 2. button press followed).
void click2() {
  // this is generally not necessary but with some older systems it seems to
  // prevent missing the first character after a delay:
  DigiKeyboard.sendKeyStroke(0);

  // Type out this string letter by letter on the computer (assumes US-style
  // keyboard)
  DigiKeyboard.sendKeyStroke(0, MOD_SHIFT_LEFT | MOD_CONTROL_LEFT | MOD_ALT_LEFT);
  DigiKeyboard.delay(100);
  DigiKeyboard.sendKeyStroke(KEY_V, MOD_ALT_LEFT);

} // click2

// This function will be called once, when the button1 is pressed for a long time.
void longPressStart1() {
  // this is generally not necessary but with some older systems it seems to
  // prevent missing the first character after a delay:
  DigiKeyboard.sendKeyStroke(0);

  // Type out this string letter by letter on the computer (assumes US-style
  // keyboard)
  DigiKeyboard.sendKeyStroke(0, MOD_SHIFT_LEFT | MOD_CONTROL_LEFT | MOD_ALT_LEFT);
  DigiKeyboard.delay(50);
  DigiKeyboard.sendKeyStroke(KEY_Q, MOD_ALT_LEFT);
  DigiKeyboard.delay(50);
  DigiKeyboard.sendKeyStroke(KEY_ENTER);

} // longPressStart1
```



Hi whats the file extention name for this code?

It is .ino. I'd recommend opening a new sketch in the Arduino IDE and just pasting this in there (don't forget to install the libraries as well), then it will automatically save the right kind of file for you.

nice project! good luck at the contest:)