

About

Set app volumes with real sliders! deej is an Arduino & Go project to let you build your own hardware mixer for Windows and Linux

deej.ocks

#audio #go #golang #arduino #gaming #diy #volume-control

- Readme
- MIT license
- Contributing
- Activity
- 5.5k stars
- 70 watching
- 555 forks
- Report repository

Releases 10

deej v0.9.10 - new binding options Latest on Aug 31, 2020

+ 9 releases

Sponsor this project

ko-fi.com/omriharel

Contributors 2

- omriharel Omri Harel
- medusalix Severin von Wnuck-Lipinski

Languages

Go 99.7% Other 0.3%

master 2 Branches 10 Tags

Go to file

Go to file

<> Code

omriharel

Resize community showcase images (w/o history rewrite)

9c0307b · 5 years ago

126 Commits

.github	[maybe broken] move all go code to own...	5 years ago
arduino/deej-5-sliders-vanilla	move arduino sketch to own directory a...	6 years ago
assets	Resize community showcase images (w/...	5 years ago
docs	add contributing.md file	5 years ago
pkg/deej	fix small typo in release notes template	5 years ago
.gitignore	add a few more entries to .gitignore	6 years ago
LICENSE	update readme with go version stuff, ad...	6 years ago
README.md	add contributing.md file	5 years ago
community.md	add builds by ocyrus, bgrier, marcioasf, b...	6 years ago
config.yaml	remove experimental status of deej.unm...	5 years ago
go.mod	add support for advanced target resoluti...	6 years ago
go.sum	start using viper to load app configurati...	6 years ago

README

Contributing

MIT license

deej

deej is an **open-source hardware volume mixer** for Windows and Linux PCs. It lets you use real-life sliders (like a DJ!) to **seamlessly control the volumes of different apps** (such as your music player, the game you're playing and your voice chat session) without having to stop what you're doing.

Join the [deej Discord server](#) if you need help or have any questions!



New: [work-in-progress deej FAQ!](#)

deej consists of a [lightweight desktop client](#) written in Go, and an Arduino-based hardware setup that's simple and cheap to build. [Check out some versions built by members of our community!](#)

[Download the latest release](#) | [Video demonstration](#) | [Build video by Tech Always](#)



Psst! [No 3D printer? No problem!](#) You can build deej on some cardboard, a shoebox or even a breadboard :)

Table of contents

- [Features](#)
- [How it works](#)
 - [Hardware](#)
 - [Schematic](#)
 - [Software](#)
- [Slider mapping \(configuration\)](#)
- [Build your own!](#)
 - [FAQ](#)
 - [Build video](#)
 - [Bill of Materials](#)
 - [Thingiverse collection](#)
 - [Build procedure](#)
- [How to run](#)
 - [Requirements](#)
 - [Download and installation](#)
 - [Building from source](#)
- [Community](#)
- [License](#)

Features

deej is written in Go and [distributed](#) as a portable (no installer needed) executable.

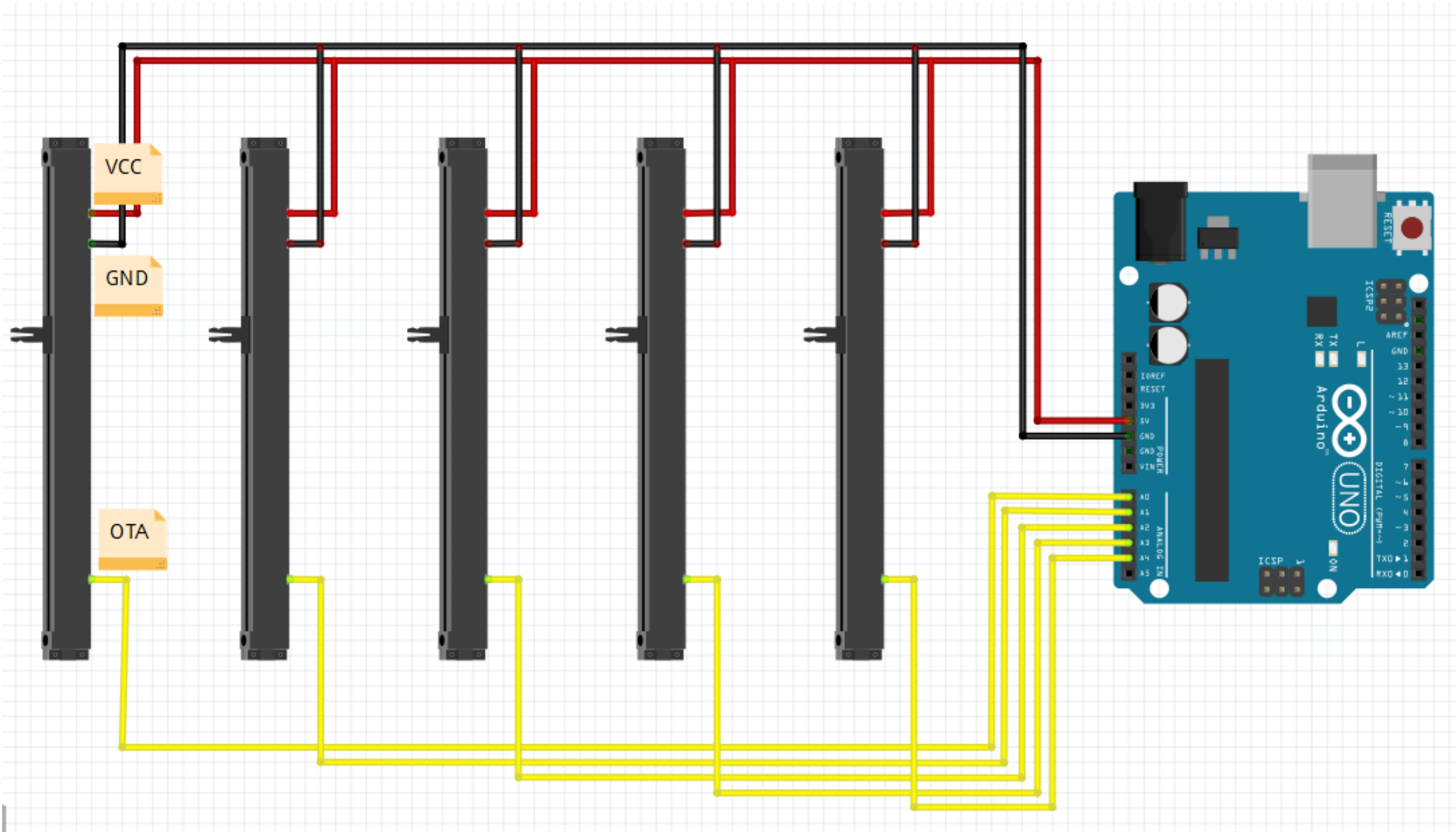
- Bind apps to different sliders
 - Bind multiple apps per slider (i.e. one slider for all your games)
 - Bind the master channel
 - Bind "system sounds" (on Windows)
 - Bind specific audio devices by name (on Windows)
 - Bind currently active app (on Windows)
 - Bind all other unassigned apps
- Control your microphone's input level
- Lightweight desktop client, consuming around 10MB of memory
- Runs from your system tray
- Helpful notifications to let you know if something isn't working

How it works

Hardware

- The sliders are connected to 5 (or as many as you like) analog pins on an Arduino Nano/Uno board. They're powered from the board's 5V output (see schematic)
- The board connects via a USB cable to the PC

Schematic



Software

- The code running on the Arduino board is a [C program](#) constantly writing current slider values over its serial interface
- The PC runs a lightweight [Go client](#) in the background. This client reads the serial stream and adjusts app volumes according to the given configuration file

Slider mapping (configuration)

deej uses a simple YAML-formatted configuration file named [config.yaml](#) , placed alongside the deej executable.

The config file determines which applications (and devices) are mapped to which sliders, and which parameters to use for the connection to the Arduino board, as well as other user preferences.

This file auto-reloads when its contents are changed, so you can change application mappings on-the-fly without restarting deej.

It looks like this:

```
slider_mapping:
  0: master
  1: chrome.exe
  2: spotify.exe
  3:
    - pathofexile_x64.exe
    - rocketleague.exe
  4: discord.exe

# set this to true if you want the controls inverted (i.e. top is 0%, bottom is 100%)
invert_sliders: false

# settings for connecting to the arduino board
com_port: COM4
baud_rate: 9600

# adjust the amount of signal noise reduction depending on your hardware quality
# supported values are "low" (excellent hardware), "default" (regular hardware) or "high" (bad, noisy hardware)
noise_reduction: default
```

- `master` is a special option to control the master volume of the system (*uses the default playback device*)
- `mic` is a special option to control your microphone's input level (*uses the default recording device*)
- `deej.unmapped` is a special option to control all apps that aren't bound to any slider ("everything else")
- On Windows, `deej.current` is a special option to control whichever app is currently in focus
- On Windows, you can specify a device's full name, i.e. `Speakers (Realtek High Definition Audio)` , to bind that device's level to a slider. This doesn't conflict with the default `master` and `mic` options, and works for both input and output devices.

- Be sure to use the full device name, as seen in the menu that comes up when left-clicking the speaker icon in the tray menu
- `system` is a special option on Windows to control the "System sounds" volume in the Windows mixer
- All names are case-insensitive, meaning both `chrome.exe` and `CHROME.exe` will work
- You can create groups of process names (using a list) to either:
 - control more than one app with a single slider
 - choose whichever process in the group that's currently running (i.e. to have one slider control any game you're playing)

Build your own!

Building deej is very simple. You only need a few relatively cheap parts - it's an excellent starter project (and my first Arduino project, personally). Remember that if you need any help or have a question that's not answered here, you can always [join the deej Discord server](#).

Build deej for yourself, or as an awesome gift for your gaming buddies!

FAQ

I've started a highly focused effort of writing a proper FAQ page for deej, covering many basic and advanced topics.

It is still *very much a work-in-progress*, but I'm happy to [share it in its current state](#) in hopes that it at least covers some questions you might have.

FAQ feedback in our [community Discord](#) is strongly encouraged :)

Build video

In case you prefer watching to reading, Charles from the [Tech Always](#) YouTube channel has made [a fantastic video](#) that covers the basics of building deej for yourself, including parts, costs, assembly and software. I highly recommend checking it out!

Bill of Materials

- An Arduino Nano, Pro Micro or Uno board
 - I officially recommend using a Nano or a Pro Micro for their smaller form-factor, friendlier USB connectors and more analog pins. Plus they're cheaper
 - You can also use any other development board that has a Serial over USB interface
- A few slider potentiometers, up to your number of free analog pins (the cheaper ones cost around 1-2 USD each, and come with a standard 10K Ohm variable resistor. These *should* work just fine for this project)
 - **Important:** make sure to get **linear** sliders, not logarithmic ones! Check the product description
 - You can also use circular knobs if you like
- Some wires
- Any kind of box to hold everything together. **You don't need a 3D printer for this project!** It works fantastically with just a piece of cardboard or a shoebox. That being said, if you do have one, read on...

Thingiverse collection

With many different 3D-printed designs being added to our [community showcase](#), it felt right to gather all of them in a Thingiverse collection for you to browse. If you have access to a 3D printer, feel free to use one of the designs in your build.

[Visit our community-created design collection on Thingiverse!](#)

You can also [submit your own](#) design to be added to the collection. Regardless, if you do upload your design to Thingiverse, please add a `deej` tag to it so that others can find it more easily.

Build procedure

- Connect everything according to the [schematic](#)
- Test with a multimeter to be sure your sliders are hooked up correctly
- Flash the Arduino chip with the sketch in [arduino\deej-5-sliders-vanilla](#)
 - *Important:* If you have more or less than 5 sliders, you must edit the sketch to match what you have
- After flashing, check the serial monitor. You should see a constant stream of values separated by a pipe (`|`) character, e.g. `0|240|1023|0|483`
 - When you move a slider, its corresponding value should move between 0 and 1023
- Congratulations, you're now ready to run the deej executable!

How to run

Requirements

Windows

- Windows. That's it

Linux

- Install `libgtk-3-dev`, `libappindicator3-dev` and `libwebkit2gtk-4.0-dev` for system tray support. Pre-built Linux binaries aren't currently released, so you'll need to [build from source](#). If there's demand for pre-built binaries, please [let me know](#)!

Download and installation

- Head over to the [releases page](#) and download the [latest version](#)'s executable and configuration file (`deej.exe` and `config.yaml`)
- Place them in the same directory anywhere on your machine
- (Optional, on Windows) Create a shortcut to `deej.exe` and copy it to `%APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup` to have deej run on boot

Building from source

If you'd rather not download a compiled executable, or want to extend deej or modify it to your needs, feel free to clone the repository and build it yourself. All you need is a Go 1.14 (or above) environment on your machine. If you go this route, make sure to check out the [developer scripts](#).

Like other Go packages, you can also use the `go get` tool: `go get -u github.com/omrihare1/deej` . Please note that the package code now resides in the `pkg/deej` directory, and needs to be imported from there if used inside another project.

If you need any help with this, please [join our Discord server](#).

Community



deej is a relatively new project, but a vibrant and awesome community is rapidly growing around it. Come hang out with us in the [deej Discord server](#), or check out a whole bunch of cool and creative builds made by our members in the [community showcase](#).

The server is also a great place to ask questions, suggest features or report bugs (but of course, feel free to use GitHub if you prefer).

Donations

If you love deej and want to show your support for this project, you can do so using the link below. Please don't feel obligated to donate - building the project and telling your friends about it goes a very long way! Thank you very much.



Contributing

Please see [docs/CONTRIBUTING.md](#) .

License