

Hardware

Pieter P

Motorized fader

I used an ALPS Alpine 100mm, 10k Ω motorized fader, part number [RSA0N11M9A0K](#). These are not cheap, I paid around €30 for it. The knob I used is a chromed FK13x25 by [albs](#). Although I cannot seem find the exact part number on <https://www.albs.de/schieb1.htm>, I think [this](#) is the part I ordered. It is important to use a knob with a metal finish, it needs to be conductive for the touch sensitivity to work.

I used a 100nF capacitor to filter some of the noise on the potentiometer.

Motor driver

I used an L293D double H-bridge. It is an absolutely terrible motor driver by today's standards, with bipolar output transistors and a voltage drop of up to 3.6V. While there are much better drivers available, this one is cheap, popular, easy to use, and good enough to drive the small motor of the fader. It includes flyback diodes on the outputs, so you don't have to provide those externally.

If you want something a bit more modern, you could look at the TB6612FNG, for example.

Power supply

The motor I used is rated for 10V. I used a 10V supply rated for 1A, which results in about 8V across the motor (taking into account the voltage drop of the L293). In my experiments, the motor used around 100mA during normal tracking, and up to 350mA when stalling the motor on purpose. According to the spec sheet, they can draw up to 800mA each at 10V, so you have to take this into account when selecting a power supply.

Capacitive touch sensing

You just need an external pull-up resistor of around 500k Ω to 1M Ω . The exact value isn't critical, but it should be high enough to be able to measure the small capacitance of the touch sensitive knob.

Connections

```
43 // ----- Hardware ----- //
44
45 // Fader 0:
46 // - A0: wiper of the potentiometer      (ADC0)
47 // - D8: touch pin of the knob           (PB0)
48 // - D2: input 1A of L293D dual H-bridge 1 (PD2)
49 // - D3: input 2A of L293D dual H-bridge 1 (OC2B)
50 //
51 // Fader 1:
52 // - A1: wiper of the potentiometer      (ADC1)
53 // - D9: touch pin of the knob           (PB1)
54 // - D13: input 3A of L293D dual H-bridge 1 (PB5)
55 // - D11: input 4A of L293D dual H-bridge 1 (OC2A)
56 //
57 // Fader 2:
58 // - A2: wiper of the potentiometer      (ADC2)
59 // - D10: touch pin of the knob          (PB2)
60 // - D4: input 1A of L293D dual H-bridge 2 (PD4)
61 // - D5: input 2A of L293D dual H-bridge 2 (OC0B)
62 //
63 // Fader 3:
64 // - A3: wiper of the potentiometer      (ADC3)
65 // - D12: touch pin of the knob          (PB4)
66 // - D7: input 3A of L293D dual H-bridge 2 (PD7)
67 // - D6: input 4A of L293D dual H-bridge 2 (OC0A)
68 //
69 // If fader 1 is unused:
70 // - D13: LED or scope as overrun indicator (PB5)
71 //
72 // For communication:
73 // - D0: UART TX                          (TXD)
74 // - D1: UART RX                          (RXD)
75 // - A4: I2C data                        (SDA)
76 // - A5: I2C clock                       (SCL)
77 //
78 // Connect the outer connections of the potentiometers to ground and Vcc, it's
79 // recommended to add a 100 nF capacitor between each wiper and ground.
80 // Connect the 1,2EN and 3,4EN enable pins of the L293D chips to Vcc.
81 // Connect a 500k $\Omega$  pull-up resistor between each touch pin and Vcc.
82 // On an Arduino Nano, you can set an option to use pins A6/A7 instead of A2/A3.
83 // Note that D13 is often pulsed by the bootloader, which might cause the fader
84 // to move when resetting the Arduino. You can either disable this behavior in
85 // the bootloader, or use a different pin (e.g. A3 or A4 on an Arduino Nano).
86 // The overrun indicator is only enabled if the number of faders is 1, because
87 // it conflicts with the motor driver pin of Fader 1. You can choose a different
88 // pin instead.
```