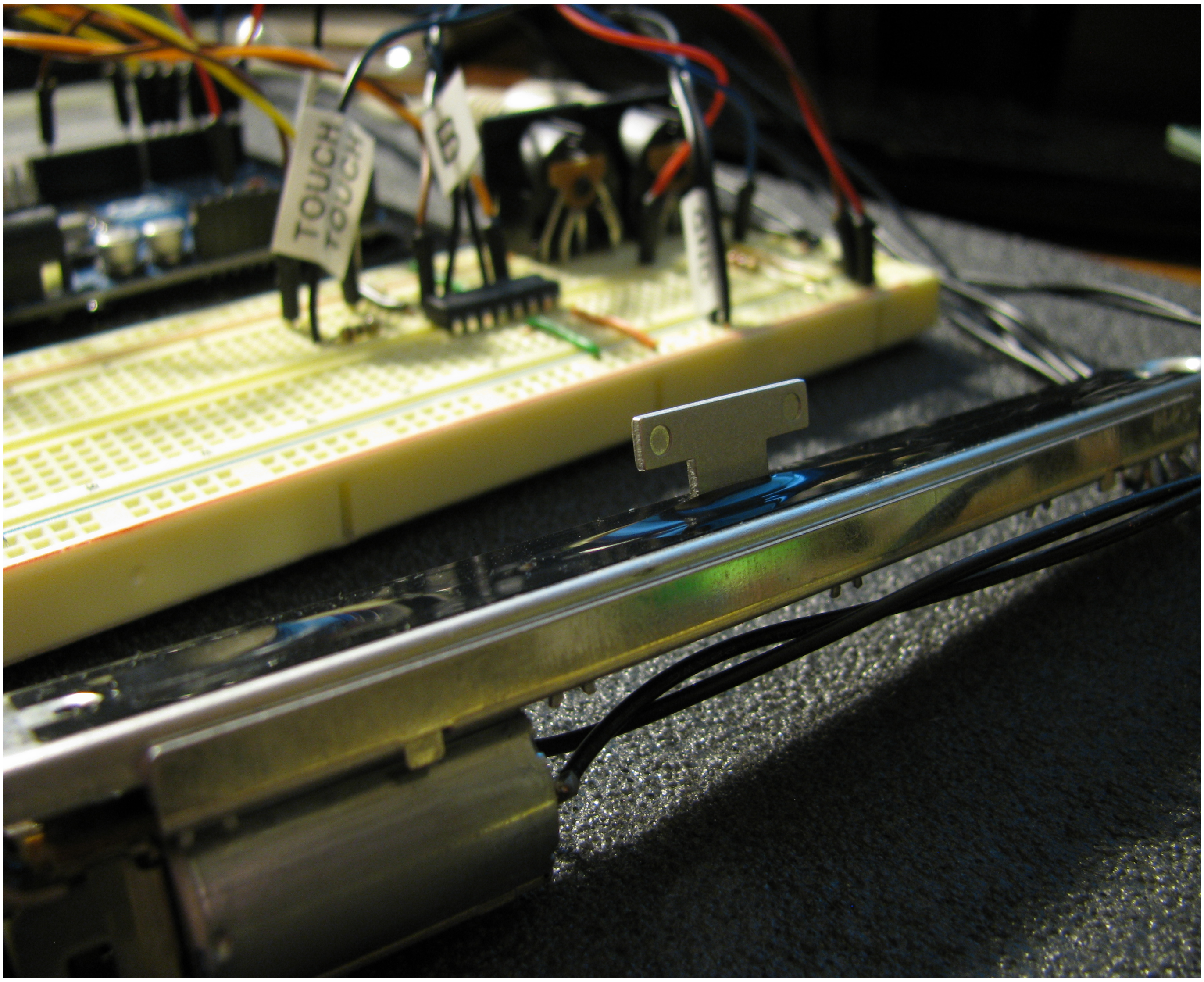
Cody Hazelwood

Motorized Faders and the Arduino

May 15, 2012

As an audio engineer, one fun thing to do is to switch between layers on a digital console as people walk by. Moving faders are a relatively simple concept, but they seem to amaze people. I have to admit, they are pretty cool. But more important than the coolness factor is the amount of flexibility they provide for mixing.

To begin my journey into the world of audio-related programming and hardware development, I decided to do a project to help me understand the technology behind moving fader control. The first part of this project is to make a motorized fader function with the Arduino platform. I'm going to give you step by step instructions to build this part of the project, including the parts I used.



Parts Needed:

- Arduino Uno
- Motorized Fader
- Breadboard
- H-Bridge
- External 9-10 V DC Power Supply
- 1 M Resistor
- 10 k Resistor
- 10 µF Capacitor
- Potentiometer

Motor Control:

The first step in understanding how to control the fader is understanding how to control the motor on the fader. The fader contains a DC motor with two terminals. If power is applied in one direction, the motor goes one way, and if you reverse the power, the motor will go the other way. Using a a digital out pin on the Arduino will give us control of the motor in one direction, but there is no way reverse the direction of the motor without help.

The device that will allow us to control the direction that power is applied to the motor is called an H-Bridge. An H-Bridge is a network of 4-6 transistors that has 2 control pins, 2 output pins, a chip power pin, a motor power pin, and ground. Based on the values of the control pins, the motor will either go forwards, backwards, or it will stop.

The logic table to your right shows exactly which control values are needed to use the H-Bridge to control the motor. As shown in the table, if the control pins are different, the motor will turn, and if they are the same, then the motor will stop. I would recommend wiring the enable pin to +5 V so you don't have to waste an Arduino pin.

EN	1A	2A	FUNCTION
H	L	H	Right
H	H	L	Left
H	L	L	Stop
H	H	H	Stop

Touch Sensitivity:

The next important thing to deal with is touch sensitivity. Motorized faders generally use a capacitance sensing circuit to determine if someone is touching the fader. Luckily, there is an Arduino library called CapSense that makes this very easy. Using a 1 M resistor and 2 digital pins on the Arduino, you can easily detect when the fader is touched. The following code shows a simple example that will light up the LED on the Arduino when the fader is touched:

```
#include "CapSense.h"

// Setup your pins and set pin 13 for output
CapSense touchLine = CapSense(touchSendPin, touchReceivePin);

void loop() {
    if (touchLine.capSense(30) > 700) {
        digitalWrite(13, HIGH);
    } else {
        digitalWrite(13, LOW);
    }
}
```

Being able to detect touch is important for two reasons. The most important one is to protect your motor. Pulling on the fader while the motor is trying to move it can strain the motor, so by using touch sensitivity, you can tell the motor to stop while the fader is being touched. Another reason is for MIDI control. If you are using the fader to write automation, it's important to inform the DAW that you are touching the fader, so it will know when to start or stop writing automation.

Reading Fader Position:

The easiest part of this project is reading the position of the fader. It works exactly like a potentiometer (because a fader is a potentiometer, just in a different layout). To read the position, connect the bottom of the fader to ground, the top of the fader to +5 V from the Arduino, and connect the wiper to an Analog In pin. From there, you can use analogRead to get the fader's position (a 10 bit value between 0 and 1023).

Summary:

This project was a fun way to experiment with controlling a fader with the Arduino. There are many improvements that could be made to this project such as adding a full calibration curve for the fader or using PWM (pulse width modulation) to make the fader move slower and more accurately for small movements. The next post will use this project and add MIDI control.

If you have any questions, feel free to comment and I'll answer them as soon as possible!

Schematic
Breadboard Layout
Code
Mackie Control MIDI Map
Video

96 Comments 1 **Login**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Sort by Best ♡

Peter Schröder • 3 years ago

Hello Cody,

unfortunately I can't find the capsense.h library anymore.
do you know where I can find it?

greeting peter

1 ^ | ⌄ • Reply • Share ›

codyhazelwood Mod ➦ Peter Schröder
• 3 years ago • edited

https://github.com/PaulStof...

Looks like the library has renamed from CapSense to CapacitiveSensor as well, so you may need to make a few code adjustments.

^ | ⌄ • Reply • Share ›

Peter Schröder ➦ codyhazelwood • 3 years ago

Hello Cody,

thank you very much. I have adapted the code. I don't know if it's so neatly programmed. But it works.
The unpleasant thing about it is that the motor fader runs very jerkily. Similarly bad as the former Yamaha O2R ;-)

I don't know if it's the fader. I use this one:

ALPS RSAON11M9 Motor-Fader 10 kΩ 0.5 W linear

But I suspect it's the sketch. I don't know if it can be improved from a programming point of view. Or maybe someone has an idea how you can dampen the start and stop phases with electronics?

^ | ⌄ • Reply • Share ›

codyhazelwood Mod ➦ Peter Schröder

codyhazelwood Mod ➦ Peter Schroder • 3 years ago

The proper way to fix it would be to implement a PID controller loop to reduce the motor's speed as it approaches where it should stop.

As a "quick fix" you could reduce the voltage on VCC1, or you could change the `10` in `void updateFader(int position)` to a larger number like 15 or 20. This padding gives the fader a little more room to overshoot without needing to correct itself.

I also remember having to mess with the sensitivity on the capsense module based on the type of fader and the fader cap (if any) being used. I can't give you any good guidance there, because it's been a while since I did the project.

Final thing to check - make sure your grounds are clean. I had trouble with odd behavior when running the Arduino off of my computer's USB port.

⌃ | ⌄ • Reply • Share ›

Jeffrey Harris • 3 months ago

Curious what it takes to scale this up to 8 faders... or 20 faders!?

⌃ | ⌄ • Reply • Share ›

codyhazelwood Mod ➦ Jeffrey Harris • 3 months ago

A good power supply for all of the motors, an encoder to read from multiple faders, and some sort of PWM driver to drive multiple H-bridges from a limited number of pins would work.

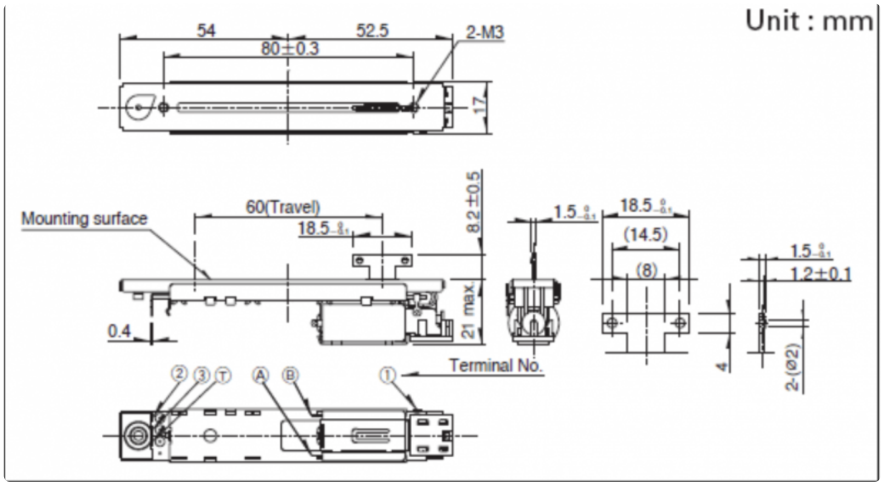
⌃ | ⌄ • Reply • Share ›

cftfsxnwuhbk • a year ago

Hello Cody,
I've recently purchased a fader from ALPS: RS60N11M9A0E
I read the code that you wrote and they make perfect sense.
I'm however, struggling to connect the fader.
"To read the position, connect the bottom of the fader to ground, the top of the fader to +5 V from the Arduino, and connect the wiper to an Analog In pin. “
I really have no idea what you been by the "bottom" and the "top"

top

according to the seller I bought from pin 1,2,3,T each stands for ground, input, output, and Touch sensitivity.
I've also attached the document from ALPS. Could you explain the pin based on your code?
Thank you so much in advance.



^ | ˅ • Reply • Share ›

codyhazelwood Mod ➦ cftfsxnwuhbk • a year ago

Hi,

Based on your description, I think it would be like this:
- Ground -> Ground
- Input -> +5V (VCC1)
- Output -> Arduino Analog Input 0
- Touch -> Arduino Digital Input 7 and then R2 (1MΩ) would connect D7 and D8

You can always use a multimeter to test, too. Set your meter to measure resistance, place the fader towards the bottom (about 25% of the way from the bottom), and test the pins. Ground to VCC1 (+5V) should measure 10k since your fader is a 10k fader, ground to A0 (Output) should be about 2.5k (or less than 5k), and VCC1 to A0 (Arduino Analog Input) should be about 7.5k (or more than 5k).

^ | ˅ • Reply • Share ›

cftfsxnwuhbk ➦ codyhazelwood • a year ago

Thank you so much!!!!! It worked flawlessly. I just have one more question that do I have to use metal caps for touch sensitivity? I imagine if I use plastic ones arduino wouldn't be able to read the capacitance due to insulation of plastic? Thx again

^ | ˅ • Reply • Share ›

codyhazelwood Mod ➦ cftfsxnwuhbk
• a year ago

I believe there are some plastic caps that have a layer of aluminum below the plastic, but I can't remember where I saw them. Metal would definitely be easiest and likely would feel a little higher quality.

^ | ˅ • Reply • Share ›

cftfsxnwuhbk ➦ codyhazelwood
• a year ago

Thank you so much! Unfortunately I think I've purchased log(Audio) pots instead of linear ones... With Arduino 's 10 bit I'm not getting accurate analog readings...I'm thinking about reading resistance and maybe multi-map them to somehow mimic a linear one? Do you have any other ideas that I might be able to make the pot as close as possible to a linear one? Please let me know. Thank you...
PS where do you get the linear slider pots with motors? Can't seem to get them from where I'm living...

^ | ˅ • Reply • Share ›

NadeemRajesh • 4 years ago

Hi Cody,
I have got a slide potentiometer which doesnot have the touch sense in it.
https://www.mouser.in/datas...

I have connected the circuit as per your instruction. I am able to calibrate where the code does get the value for max and min value of fader.

After that the moment I turn the pot...the code gets stuck..the values in Serial monitor doesnot move at all.

My code is as follows

```
const int motorDown = 5; //Declaring where our module is wired
const int motorUp = 6;
const int wiper = 0; //Position of fader relative to GND (Analog 0)
const int pot = 3;
const int pot = 5;
double faderMax = 0; //Value read by fader's maximum
position (0-1023)
double faderMin = 0; //Value read by fader's minimum
position (0-1023)

void setup() {
Serial.begin(57600);
pinMode(5, OUTPUT);
pinMode(6, OUTPUT);
calibrateFader();

}

void loop(){
move_fader();

}

void move_fader(){
int pot_state = analogRead(pot);
int wiper_state = analogRead(wiper);
Serial.println(pot_state);
Serial.println(wiper_state);

if (pot_state < wiper_state - 10 && pot_state > faderMin) {
digitalWrite(motorUp, HIGH); //Switch between this HIGH and
LOW to change direction
while (pot_state < wiper_state - 10) {}; //Loops until motor
is done moving
digitalWrite(motorUp, LOW);
}
else if (pot_state > wiper_state + 10 && pot_state <
faderMax) {
digitalWrite(motorDown, HIGH); //Switch between this HIGH
and LOW to change direction
while (pot_state > wiper_state - 10) {}; //Loops until motor
is done moving
digitalWrite(motorUp, LOW);
}
}

void calibrateFader() {
digitalWrite(motorUp, HIGH);
delay(500);
digitalWrite(motorUp, LOW);
faderMax = analogRead(wiper);
digitalWrite(motorDown, HIGH);
delay(500);
digitalWrite(motorDown, LOW);
faderMin = analogRead(wiper);
}
```

^ | ⌄ • Reply • Share ›

codyhazelwood Mod ➦ NadeemRajesh • 4 years ago

Try commenting out everything below the `Serial.println` in `move_fader()`. Make sure the values are printing correctly and that moving the fader up actually moves the values up. It's possible that you have the wires reversed.

Keep in mind that without the touch line, you cannot change the value with the fader - you can only change it with something external (like the potentiometer you're using).

^ | ⌄ • Reply • Share ›

Sebastian Coudijzer • 5 years ago

hello Cody.
Im looking for the nominal tention of the capacitor? voltage of motor?
Thanks and keep u the good work!

^ | ⌄ • Reply • Share ›

codyhazelwood Mod ➦ Sebastian Coudijzer • 5 years ago

Hi Sebastian, I'm not sure what you mean by the nominal tension of the capacitor? The values for the components are listed on the schematic and in the blog posts. Here is the spec sheet for a fader/motor similar to one that I used: http://www.alps.com/prod/in... Keep in mind that it could vary depending on which one you choose to use.

^ | ⌄ • Reply • Share ›

Shuptuu • 5 years ago

Hi,
I have an issue with Touch Sensitivity. I'm using an Alps RSAON11M9 motorized fader, with touch sense track, and a teensy LC.
What I have done is I connected the "T" terminal (the touch sense track) to a touch pin on the Teensy, and add a 1Mohm resistor between this pin and an output pin directly on the teensy.
Could somebody confirm I'm doing well? Is there anything else to do? Something to connect to the ground? Or to not connect to the ground?

to the ground?
Thanks for your help

Reply • Share ›

codyhazelwood Mod ➦ Shuptuu • 5 years ago

I haven't used a Teensy before, but it sounds like you have it connected correctly.

You don't need to connect ground for touch to work, but it will be more stable if you do connect it to earth. Using a USB cable takes care of that for you. If you start reading the values from the fader, then you'll need to connect one side of the wiper to ground and one side to V+.

Also, here's a pretty good article I found on Capactive touch sensing with a teensy. Just connect to T instead of to a sheet of metal. https://www.pjrc.com/teensy...

Reply • Share ›

Shuptuu ➦ codyhazelwood • 5 years ago

Hi,
Thanks for your answer. And thanks for the link
That I already saw before but reading again, I realized I missed something important: they say Teensy LC do not require the CapacitiveSensor library. To read capacitance on touch pins, simply use touchRead(pin).
Well, I gonna try this!
Thanks again

Reply • Share ›

Shuptuu ➦ Shuptuu • 5 years ago

Hi,
Got it!!
Well, I first tried using the dedicated touchRead(pin) command: same problem.
And then I discovered that the metal shield on the fader MUST NOT be connected to the ground!!!
Which is not so easy when you have a front panel in metal, connected to the ground somewhere else.... ;-)
So found a way to isolate the fader and now it works perfectly!
it was snicky!!

Reply • Share ›

codyhazelwood Mod ➜ Shuptuu • 5 years ago

Nice job! That's interesting, but a good catch. I figured that the fader's touch line would have been isolated from the chassis ground... but maybe not. :)

• Reply • Share ›

Tobias Rüther • 5 years ago

Hi Cody,
you didnt make my comment public, but im sure you read it. Just wanted to tell you that i fixed my problem with the buzzing fader with connecting the pin1 of the l293d (enable input) to another digital out of my arduino and wrote another if-else statement:

```
if (potiValOld != potiSmooth){
digitalWrite(motorEnablePin, HIGH);
.
.
.
//your motor driving code
.
.
.
}else{
digitalWrite(motorEnablePin, LOW);
}
```

• Reply • Share ›

codyhazelwood Mod ➜ Tobias Rüther • 5 years ago

Sorry, sometimes I don't have time to look at the comments for a few days. I'm glad you found the issue. Feel free to submit a Pull Request on GitHub if you want.

• Reply • Share ›

Tobi • 5 years ago

Hi Cody,

thanks for your work, everything is clearly and my fader works fine on the first run. I have one problem: I used an ALPS RSA0N11M9 Motorfader which is 10K. Sometimes i get a buzz-sound from the fader when its not moving. Buzzing stops if i touch the fader but after releasing my finger it starts again.
I smoothed the pot and fader values (and monitored them) so im sure this is not the problem.

• Reply • Share ›

Joe Below • 5 years ago

I have no experience with any of this and it's making my head spin. You keep saying easy easy easy but still this project is very daunting. This article was found to help find a solution for controlling everything my guitar sound is wired to in my DAW. I think I'm closer to hot gluing a pre made controller before I delve into this, sadly.

⌃ | ⌄ • Reply • Share ›

codyhazelwood Mod ➦ Joe Below • 5 years ago

Hi Jon, sorry that the project seems so daunting. When writing the article, I was assuming a basic working knowledge of Arduino, MIDI, and basic electronics. If it seems too difficult, try getting an Arduino starter kit and completing several of the demo projects. Once you understand how the Arduino works and how to assemble a few basic circuits, come back to this project, and I don't think you'll find it to be as difficult.

1 ⌃ | ⌄ • Reply • Share ›

Joe Below ➦ codyhazelwood • 4 years ago

Hey thanks for the support.

⌃ | ⌄ • Reply • Share ›

Juan Manuel Segovia • 5 years ago

Nice! If i want 8 faders , can i use the same arduino? And When i load mix session in protools for example, have the fader automaticaly recall?, Sorry for my english

⌃ | ⌄ • Reply • Share ›

codyhazelwood Mod ➦ Juan Manuel Segovia • 5 years ago

Yes, you could use the same arduino, but you'd need some additional hardware to make it work. You'd also need a bigger power supply to drive 8 motors. Try one, then two, then scale up to 8 to help you work out the kinks.

Auto recall should work fine on any DAW, as the DAW will send out the faders position when the project loads. This project won't work with Pro Tools though, because Pro Tools doesn't support Mackie HUI. You would need to emulate Mackie Control with SysEx messages for that.

⌃ | ⌄ • Reply • Share ›

Juan Manuel Segovia ➦ codyhazelwood • 5 years ago thank you master!

^ | ˅ • Reply • Share ›

kdogus77 • 5 years ago

Nice work on this project! Would you happen to have basic code for just controlling the motorized pot with a standard pot? Touch response not needed. Just looking for a 1:1 linear relationship; pot turns up, fader moves up. Thank you.

^ | ˅ • Reply • Share ›

kdogus77 ➦ kdogus77 • 5 years ago

The motor is not running through the calibration. I get a very small movement just after uploading code, then nothing. any thoughts?

^ | ˅ • Reply • Share ›

codyhazelwood Mod ➦ kdogus77 • 5 years ago

You might have the positive and negative for the read circuit on the fader reversed (or on the motor), or you might not have enough current to drive the motor.

Try breaking everything down into pieces:
- Turn off the motor control and log the value of the fader. When the fader is up the value should be higher, and when the fader is down the value should be lower. - Then, try controlling just the motor. Make sure that when you tell it to go up that it goes up and not down. (Be careful, because if the motor is stuck and you continue applying power to it, it can fry it)
- Finally, once the other two pieces are working and you understand what is happening with each of them, try connecting it all together to have the fader control the motor.

^ | ˅ • Reply • Share ›

kdogus77 ➦ codyhazelwood • 5 years ago

I got it! thank you! My motor is slower than your was. I needed to increase the delay during the calibration. Min and max values were initially very small. The relationship is pretty good. is there any way to tighten up the control?

^ | ˅ • Reply • Share ›

codyhazelwood Mod ➜ kdogus77 • 5 years ago

https://github.com/codyhaze...

Then remove line 36, 52, 65, and 95-98.

^ | ⌄ • Reply • Share ›

Spray Life • 7 years ago

Hello! I need to contact you because I need just that your midi project for my daw Logic Pro 9. When loading the Midicontrol code Arduino gives me error loading because it tells me that is missing in the library captsense what should I do? Thanks Roberto

^ | ⌄ • Reply • Share ›

codyhazelwood Mod ➜ Spray Life • 7 years ago

Hi, you'll need to download the capsense library (http://playground.arduino.c..., add it to the Arduino IDE, and update the code to use the version you downloaded. I wrote this software a long time ago, so it doesn't work with the latest CapSense.

^ | ⌄ • Reply • Share ›

Spray Life ➜ codyhazelwood • 7 years ago

Ok fatto! Ma ora ottengo questo errore! Perdonatemi quindi i fili di canale MIDI da dove li collegano? Vi allego le foto e schermate !!!

see more

⌃ | ⌄ • Reply • Share ›

codyhazelwood Mod ➦ Spray Life
• 7 years ago

I would need to see all of your code to be sure, but it looks like you need to download and add the MIDI library (#include <midi.h>) (http://playground.arduino.c....

⌃ | ⌄ • Reply • Share ›

Spray Life ➦ codyhazelwood • 7 years ago

...sorry! This is code!

Cody Hazelwood

codyhazelwood

Software Engineer and Audio Enthusiast