# Programmable 8 Loops Pedal Switcher With Arduino Mega2560

by CarraN

*last update: february 17 2016*

This is the second part of my previous tutorial about how to do simple programable pedals switcher (or looper) now with a MEGA2560 Arduino to achieve 8 storable loops on 8 presets. I recommend to take it a view for the coding part explanation and how the things are tight together.

Note that is possible to improve the number of total loops. Just keep in mind that it add more lines to the code and more cost on hardware. That said, improving the possibilities in the mean time!

Number of max loops = total number of digital pins - 5 /3 (*minus 3 pins for selecting mode and the two Rx/Tx*)

In this tutorial we have 51 digital pins - 3 = 48 / 3 so 16 loops available. Just think you can add more interesting options like increasing the number of presets, add a display or read pot values for expression pedals by exemple, thus decreasing the number of available loops. But frankly: "who needs 16 pedals on a pedalboard?"

Sure, they are...

A**VOID using analog pins named A0 to A15, just use the digital ones! (2 to 53)** See the MEGA2560 pinout

***Note: this project DOESN'T allow to reverse order of the pedals***. This involves to matrix all that! That means: **number of relays = (quantity of pedals) X (quantity of pedals)**!!! and a lot of nightmares and disillusion! But actually you can consider 2 (or more) groups of loops, ie: one before the jack input of your amp and the other on the fx loop, if your amp is provided obviously! The only thing to do is to add a couple of chassis jack between two relays, the choice is yours. This is called the "breaking technique".

We can also use pin D1(Tx) for Midi communication.

So, in order to keep it as simple as wanted, see the schematic of this version of the project.

---

**Caution**: some noise issues where experienced by some
builders (me included). I suspect those cheap and chinese relays aren't made for audio signals. The ideal relays should be some called DPDT but for instance few advices can be followed: use shielded cables all along the signal path, ensure all the circuit in a metallic box use caps (usually 10u) for decoupling and/or add some high resistances to all in and outs jacks. Adding a booster pedal in front of the input may strongly reduce these noise...

_____

Hardware:
-1 chinese Arduino Mega 2560 (keep your original for test other projects)

-8 momentary switches

-1 LM7805 regulator (5v.)

-1 three ways selector

-8 5v. DPDT relay modules or

-1 sixteen relays module (9 / 12 volts) or *1 eight dpdt relay bord (it's a lot expensiver)*

-18 chassis 6.3 female jack (minimum) / 20 if you want to add the "*breaking technique*"

-1 9V-12v /3A power supply

-16 leds and 470 ohms resistors

-1 MIDI din + 1 220 ohms resistor (for optional MIDI out)

-spare cable or chinese Dupont wire (male to female)

(not included: the box)

---

Tools:

-plyers/ cutter

-soldering station/ iron

---

USE:

-position "A" from the selector (cases a,b,c,d,e,f,g,h on the sketch) -> select wich pedal to loop ON (1,2... 8)

-position "B" (cases i,j,k,l,m,n,o,p) -> select the number of the preset (1 to 8) where you want to preserve the loops (the led of the preset you choose will lit 2 or 3 times)

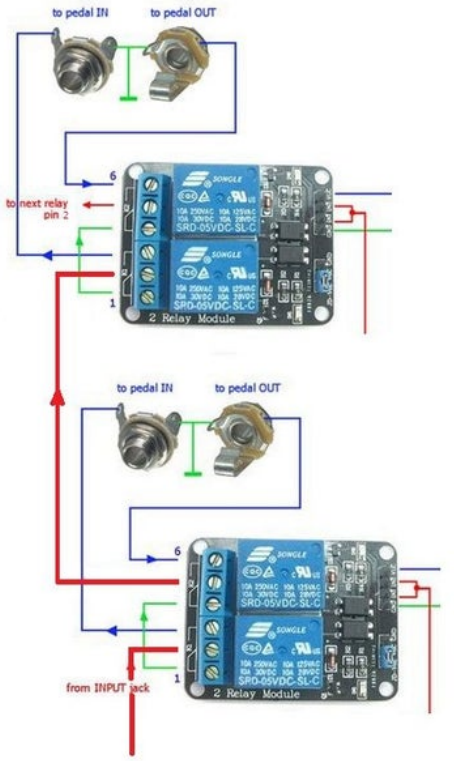
-position "C" (cases q,r,s,t,u,v,w,x)-> read the preset (1 to 8) you stored previously

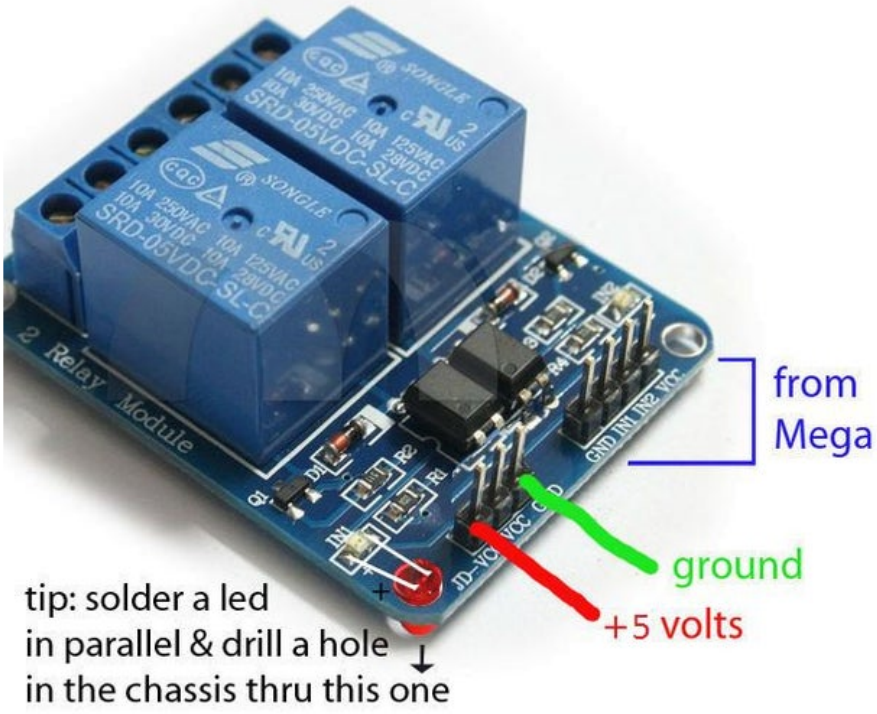
**Watch it on youtube**

---

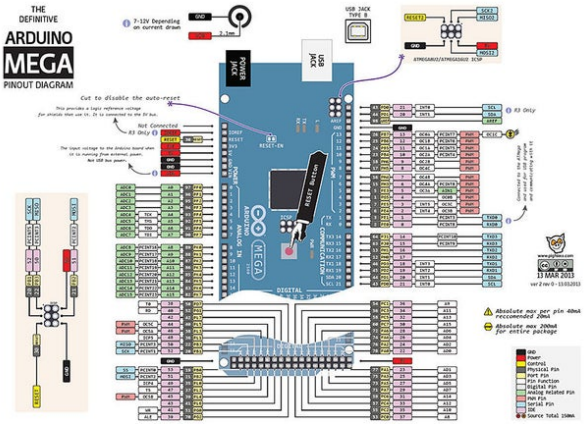
needed Arduino library:

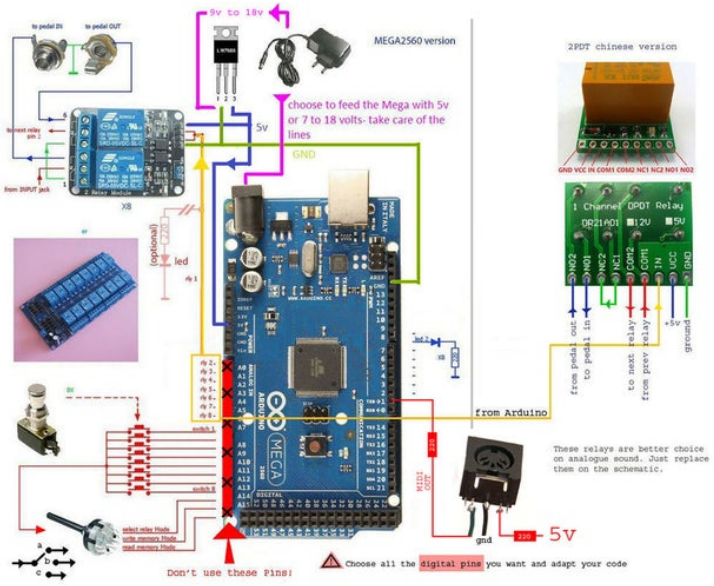
keypad.h


to pedal IN
to pedal OUT
to next relay pin 2
2 Relay Module
to pedal IN
to pedal OUT
from INPUT jack
2 Relay Module


2 Relay Module
from Mega
ground
+5 volts
tip: solder a led
in parallel & drill a hole
in the chassis thru this one


AMP CHANNEL SELECTOR TRICK
Amp control stereo jack
VCC
GND
to control 2 in Arduino
to control 1 in Arduino
2 Relay Module
* DO NOT CONNECT IT TO THE REST OF THE RELAYS


THE DEFINITIVE
ARDUINO
MEGA
PINOUT DIAGRAM


MEGA2560 version
9v to 18v
choose to feed the Mega with 5v or 7 to 18 volts- take care of the lines
5v
GND
led
from Arduino
2PDT chinese version
1 Channel DPDT Relay
These relays are better choice on analogue sound. Just replace them on the schematic.
5v
Don't use these Pins!
Choose all the digital pins you want and adapt your code

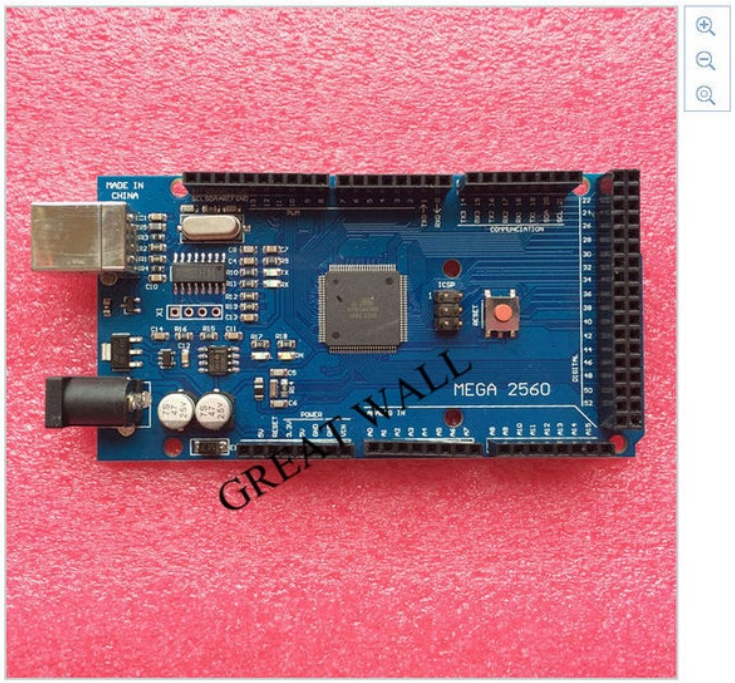
## Step 1: Programmable MEGA2560 Arduino Based 8 Pedals Looper

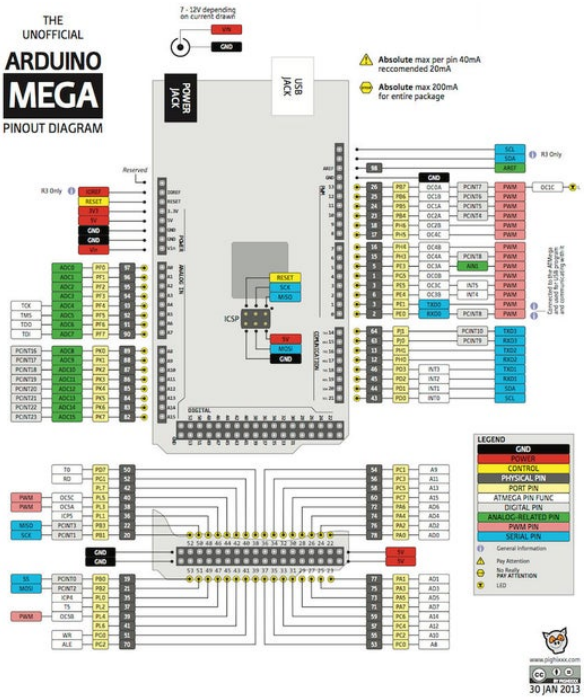
You can even add some pot reading for expression pedals: there's enough room for it! Or even 8 more leds that can indicate wich effect loop is engaged when in "read" mode (in parallel to relays). Some follower suggested me to implement the possibility to select in wich "pre" or "post preamp" mode each effect can be selected (google for "4 cables" technique). This is also possible but adding a 16 relays module that will take more space and weight on the box. I suggest choosing "16 x DPDT relay module" to limit space waste but I must admit that I didn't found any cheap of these (~100$/piece) !

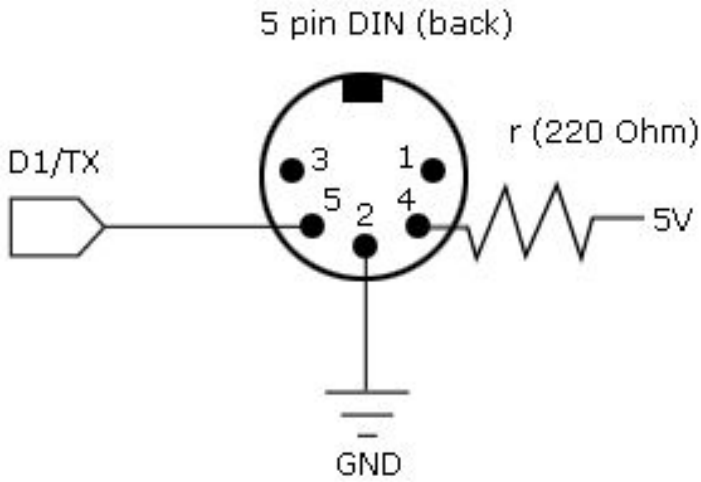
For this simple project, I suggest you to buy a cheap chinese copy of this board as you can find it for ~7$. You'll save your original board for arduino practicing.

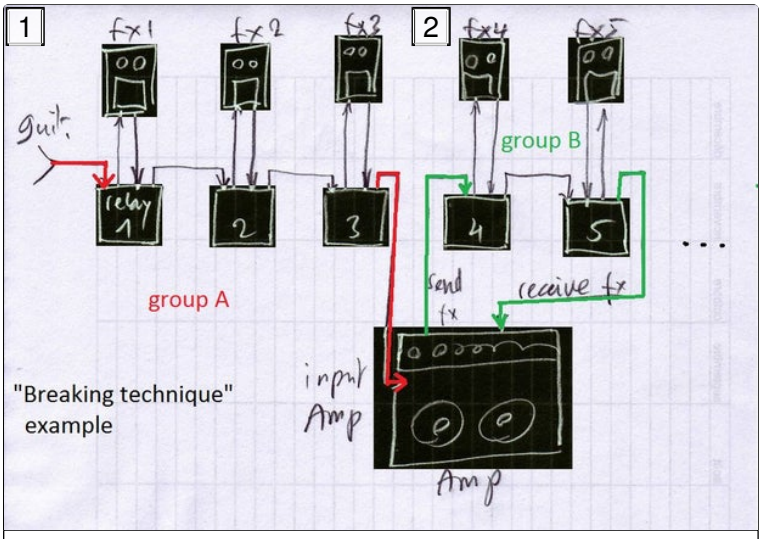
So, here is the code for this beast as an example (choose your own pins):

Many thanx to PascalP3 to have corrected and improved this sketch!

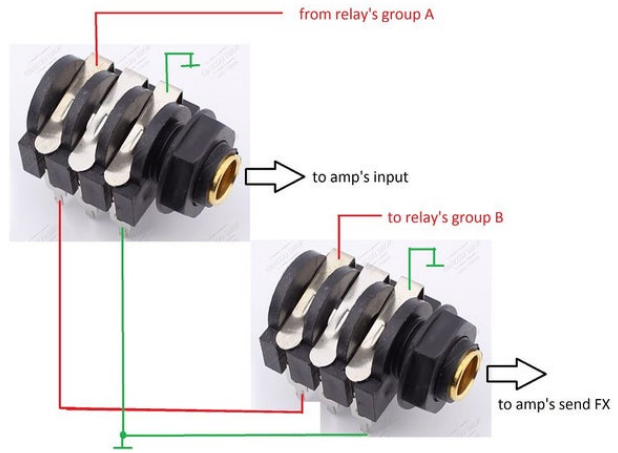





1. group A = fx1 to fx3 going to your input's amp
2. here the group B which connects to the send/return loop of your amp (if provided)




How to break the bridge between two groups of relays

https://www.instructabl...

Download

## Step 2: Hardware Building

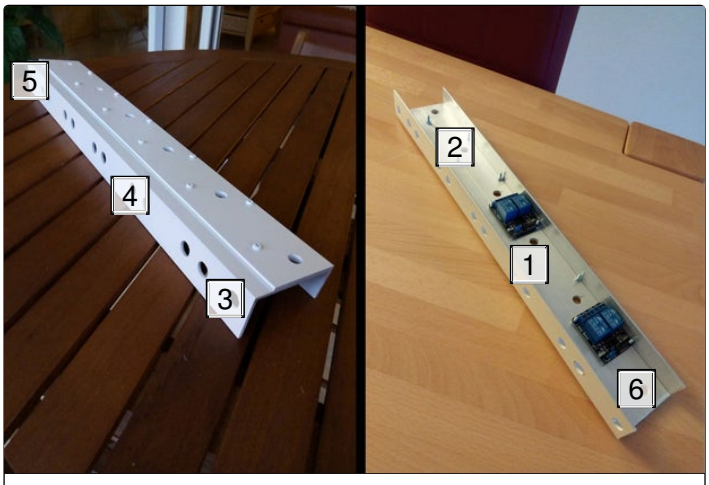
Here's an example with 5 switches and 1 function selector and another 8 loops project by David B1 who gently share pictures of his great job! Thanks to him.

---

check out some outstanding bands I play with

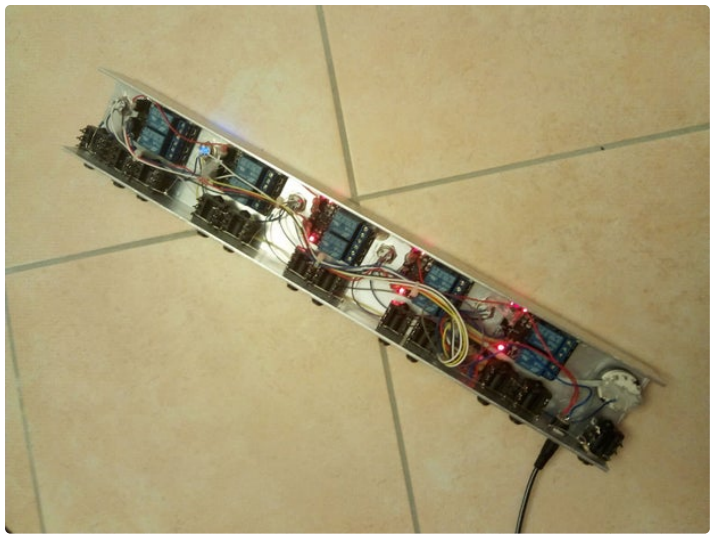
ZeK

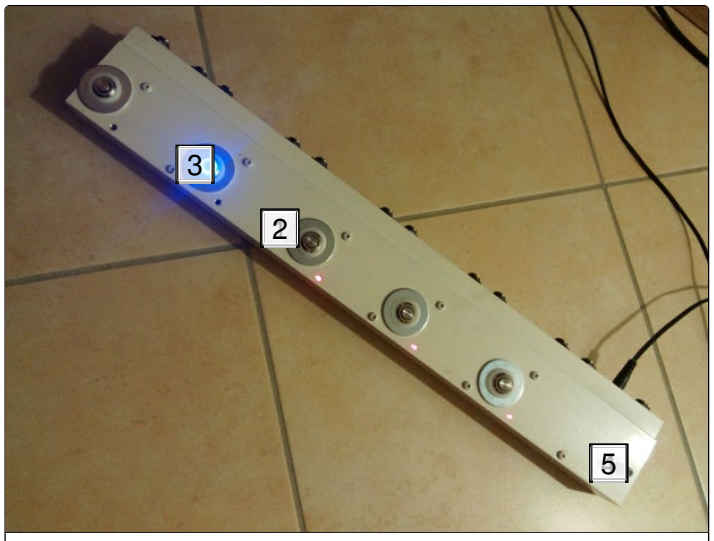

1. this aluminium profile is smaller than the other. Both have the same length.



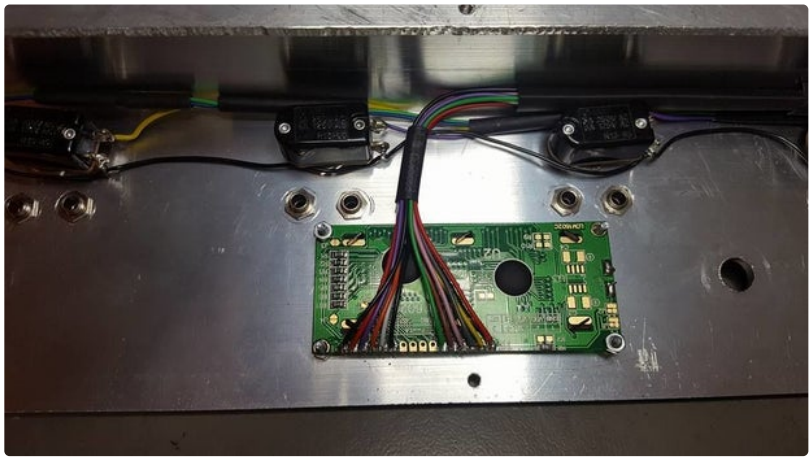

1. here come the 2 relays boards
2. and here for the pushbutton / led
3. the hole for the future jack in from guitar
4. the send and return loops
5. the output to amp

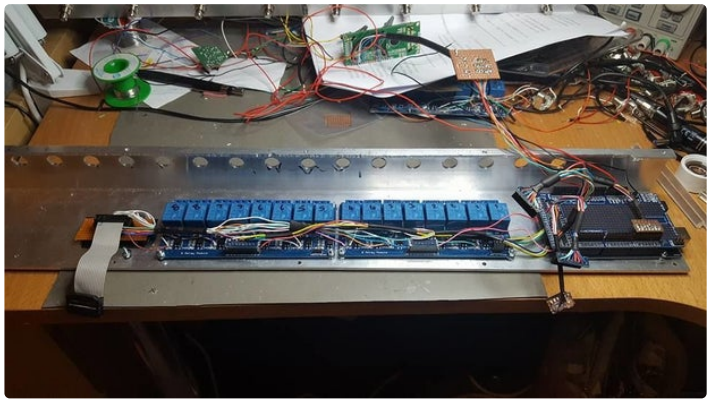
5. the output to amp
6. the function selector will come here

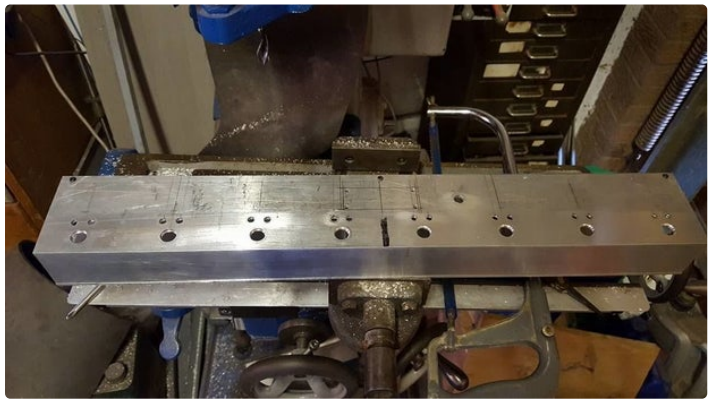


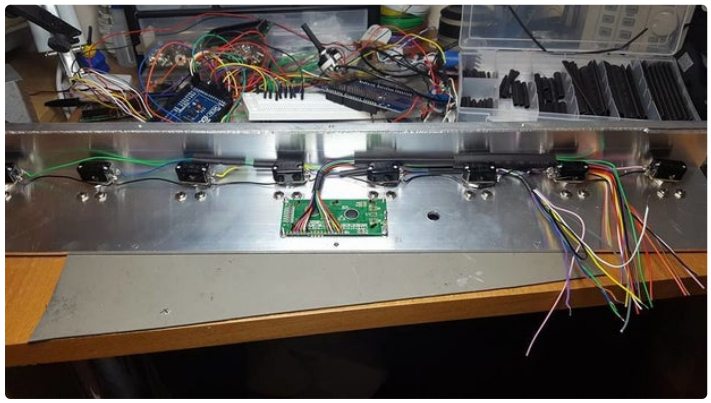


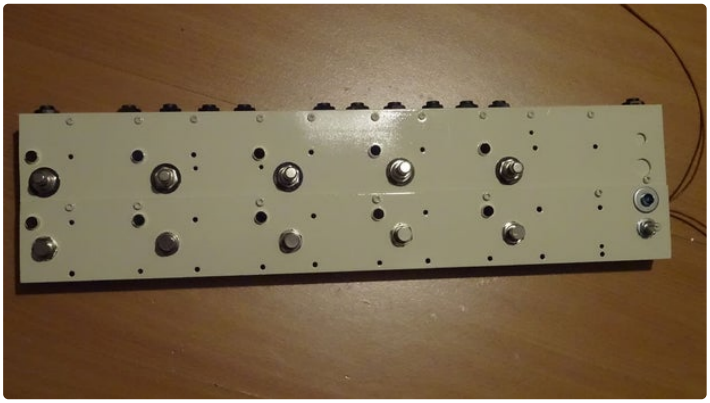

1. here the 3 last relays selected (red leds)
2. here the 3 last relays selected (red leds)
3. blue led preset
4. 3 ways selector
5. 3 ways selector

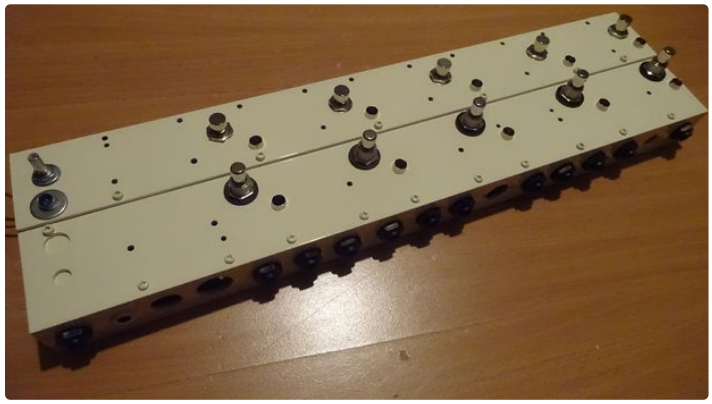


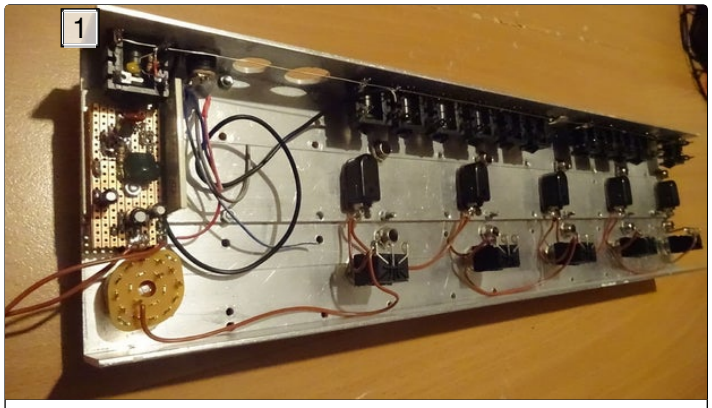











1. AMZ mosfet booster
http://www.muzique.com/schem/mosfet.htm

Can I create infinite patches for the 8 pedal loops? How would I do that? Thanks!

hey!

what do you mean about infinite patches?

I'm sorry, I mean "presets"! For example, in a gig I have 30 songs, each one with different pedal combinations, can I have 30 presets for each song?

Many thanks!

30 x 30 = 900 memory presets in a MEGA2560?

The eeprom has 4kB, so 4000 /8 = 500.
Sadly, it'll miss 400 places of presets the way memory is managed in this project...
One solution could be to write the eight relays states in a bit, by ex:
00010010, 11101100, etc... But I must admit that 's something that I've not achieved, though I KNOW it's possible. In this case you could have 4000 memory preset.
Other one is adding more memory to the Arduino. There are plenty of tutorials out there.

cheers

ps:If you find the solution you're welcome to share

I know it's 2 years late, but you asked for a shared solution ;-)
I made a version as you described with storing the preset as one byte. With bitRead you can iterate over all 8 bits of a byte and you will end up having all your relay states.

Something like:

```
void byteToRelayStates(byte b) {
for (byte i = 0; i < 8; i++) {
int bitValue = bitRead(b, 8-(i+1));
relay[i] = bitValue == 1;
}
}
```

Hey, why not to share the whole code ? Because is not clear how to implement these few lines into the rest of the sketch. If you don't want it's ok, it's up to you!

cheers

Hey, I definitely will do so. But I want to do some clean up as the current version is a little bit messy and needs some polish ;-)
I am a professional developer but new to Arduino and C and the IDE. I still want to raise the code quality to my personal desired level.

Sure I understand ! So can you please explain (to us) this line code I don't got it:

relay[i] = bitValue == 1;

The rest is just Arduino's standard code, so ... Thanks though !

cheers

Nice ! plenty more memory available in there !
I'll try your code these coming days though I dont really need that extra memory but for the sake of this instructable! Thank you man!

cheers

I made a programmable 19" rack version which can be controlled by an external MIDI device. I used the Sparkfun MIDI shield for the MIDI communication.
Up to 128 programs can be stored on SD card with a name. Fully programmable with 10 buttons on the front panel.

Finally I wrote the full code all by myself but used the basics for the wiring from this instruction. Thank you very much for this inspiration

Nice !

may ask you about the midi connection?

sure, what do you want to know?

do u have schematic or layout for the midi connection betwen arduino,relay,and midi socket, if want to use the midi connection for program change, how can i edit the midi program,as we know,there is midi section in the ino file..thank you so much i apreciated

please check the upgraded schem up there, I have added midi out connection on it. For more information just watch the usefull Amanda's instructable here->
https://www.instructables.com/id/Send-and-Receive-...
cheers

Nice...Thank you so much...

you're welcome!

Hello! first thanks so much for sharing this! greetings from Brazil!
I have a newbie doubt. In the list of materials were listed 9 / 12v relays. but in the images I am seeing the 5v connections.
I will buy DPDT relays! should i buy 5v or 12v?
thank you very much.

Hi!
yes you're right! These are 5v. relays. In fact it's not as important as this but the schematic sould be a bit different wit 12v. relays...
So following this instructable step by step, there are 5v. relays. I'm going to change this.

cheers

Hello, thank you for this ecxellent project.
I encounter a problem: when I switch to A mode to select my loops no led lights up except that of the last loaded bank. Why?
Greatings from France

salut!

en mode A (programmation des loops), c'est toi qui décide quel loop est engagé ou non. Par défaut, il y a la dernière combinaison que tu peux garder ou changer comme tu veux.

merci, j'ai debuger j'avais simplement une soudure défectueuse. Mais j'ai un autre problème en mode A, quand la loop 5 est sélectionnée, elle est contrairement aux autres impossible à deseiectionner

je dirais: continue à débugger t'es dans la bonne voie!

I have noticed a few people in the comments talking about having to give up on this because they could not solve the noise issue. I have mine working almost silent. Any noise you do hear is so low compared to the volume of the instrument that unless you were sitting there repeatedly hitting a switch without playing a note, you MIGHT hear it. I have also completely eliminated any interference and buzz or hum that comes along with it. Without decoupling caps, resistors, shielded cable or any of that stuff.

In my case the secret was simply connecting to an actual earth ground.

I have a 120v standard computer mains inlet in the back of my case, from there is a typical 12v transformer I stole from a wall wart power adapter. Connect that to an lm7805 and 7809 ( 5 to power relays, 9 to power the MEGA R3. While also keeping the MEGA behind its own 5v regulator for added safety.

But yeah this thing was a noisy piece of crap just using wall warts and regulators, once I brought a 3 prong mains cord into the case and attached all the "ground" wires to an actual earth ground and not just the negative pole on an AC/DC adapter, thing went dead silent and sounds brilliant. I am also using low signal relays. I just went to my local electronics surplus and bought a bunch of 5v relays that can handle a max of like 0.037A, made my own driver board for them and yep works like a charm, I would have no problem using this live now and even with my sovtek bigmuff the click is barely noticeable, you could have to know exactly what you are listening for to notice it. For reference the relays I am using: are CP Clare LSR 2C05.

MFR:CP Clare

MFR No.LSR2C05

STOCK No.JCFD-1011

SKU No.204184

Hope this helps someone out.

Thanks to share Kyle!

I'm convinced about how better are the low signal relays and the quality of the power! Nice job with a computer supply indeed!

cheers

carranen

I am confused with the additional voltage regulator that is needed between the arduino and the relays. Where does that go on the board? or is it just floating around whilst being grounded to the chasis which is metal?

Regards

Jim

fix it on the chassis, just take care that the other pins don't touch it (the chassis) or it will blow! The central leg is the same as the radiating part of the regulator so you can leave it off or ground it, it's yours. In the schematic it doesn' t appears but you can add one capacitor (100n/50v. typically) between input and ground and another (~10uF/25v) between output and ground. Good luck!

Thanks for that. I am using 8 relays, on one single board. Do I need 8 regulators?

one is enough.

Ok Many thanks for your quick replies. Will keep everyone updated. Some of the parts should be here today.

Programming is scarey though.

Another quick question. On the voltage regulator, there are 3 pins. Pin3 is connected to the 5volt of the arduino, pin 2 is ground, where do I connect pin 1.
Not sure from the diagram but it seems to connect to the main external power (9v to 18v) that comes into the Arduino from a power supply. Am I reading this correctly?

I have also seen on Amazon this board

https://www.amazon.co.uk/WINGONEER-L7805-LM7805-three-regulator/dp/B071VS43R5/ref=sr_1_1?ie=UTF8&qid=1518465787&sr=8-1&keywords=LM7805+regulator+%285v.%29&dpID=41OeDfGaP-L&preST=_SY300_QL70_&dpSrc=srch

Can I use that instead of just a voltage regulator?

Sure it's better.

will this project be capable of switching amp channels as well as different loops? I'm trying to find something that will for instance...go from dirty channel on amp, to clean with lets say a chorus and delay with one click, then with another button switch to dirty channel with delay.

hey

yes it can with a bit of mods!

cheers

Hi Pascal!

yes, I let it down for the same reason, though the noise appears only with distortion pedal, and I guess those chinese relays are not so friendly with audio. If I remember, putting a booster signal pedal before the switcher threw out this noise issue for a large part but not entirely... It will be interesting to see how many people achieved it without any sound issues. Some build are very impressive!

Hope you'll get it!

cheers

Hi CarraN,

It's been a while since i checked in. The last i worked on,was trying to get rid of the clicks of the relays in the audio line. Unfortunately i did not succeed and got quite discouraged. So the project ended up on a shelf in the attic. I'm wondering if anybody experience similar problems and if they managed to solve it...

I'm thinking about using audio relays which are much smaller too.

Also i've looking at the possibilities of using the CD4053 integrated circuit which should be noiseless (at least without relay clicks...)

Looking forward to some usefull input.

best regards,Pascal

Hello

Really impressive work, thanks to CarraN, also PascalP3 and now mikeganz.

Looking
to build this shortly (waiting for parts) and happily noticed
mikeganz's addition to the project. Bank support!! Awesome feature,
thanks for your work.

Everyone is welcome but pay attention that mikeganz have actually an issue with his code... (see commentaries)

cheers

carran

Hi there! I've been working through this and have come up with a few things.

1) Banks! I figured out some math that allows banks for easy storage with only one extra button/switch. I can share if you like.

2) I put together some code to allow for 2 buttons/switches and 2 relays to control read/program loops/program preset instead of using a rotary switch. Again, happy to share...

HOWEVER - I'm trying to get this to use 16 loops, and that's not working. With 10 loops, where "i" is always a single digit, it works great - but with 11 or more, where "i" is a double digit, it is running the loop constantly because it is reading each digit separately (e.g. 1 and 1, 1 and 2, 1 and 3 instead of 11, 12, 13, etc. — this is my theory anyway).

Any insight on how this might be fixed?

Hi!

please can you send us a brief schematic because I don't understand what's not working on your project. Is it the display or selector or what?

cheers

The issue isn't in the schematic... 10 or less loops, and your program works perfectly. From 11 or more loops, the program stops working properly. I think it's in the number of loops vs. keypad rows.

Your instruction said this could go up to 21 loops (or 0 to 20), but as soon as you get more than 10 (or 0 to 9), the whole program hangs up. Some debugging is showing me that it's possible the "char keys[rows][cols] = {" line is only using the first number in "byte rows" (in my case, 1 instead of 16) because it's not set for a 2 digit number, and I'm not sure how to make that work.

Running a serial line in Void Setup to see what "i" produces shows there is no issue there with 2 digit numbers, so I think that the keypad itself isn't setup to handle more than 10 rows (as it is set from 0 to 9).

This issue has the butterfly effect of having all these pins set for the keypad with nothing to use them for - thus the writeOut and memory functions hang up, and the readPreset mode doesn't work at all. It just keeps cycling through everything.

I may have a workaround for my specific issue, but it won't work for anyone with more than 10 loops. Does that make sense?

I don't think (but I'm not 100% sure though) the keypad library is limited to n caracters, except the by the max of ascii caracters available in keyboards, of course! check the keypad library.

So you're telling me that when you select the loop 11 by exemple, relay 11 doesn't work! is that right?

Hi Mike!

send us your code and we'll see ehat's going on.

cheers

carran

Hi CarraN! Code is below (doesn't include the 2 relays for programming yet. I also removed the midi stuff because it is not needed for what I'm doing, and I need every single pin I can get, hehe):

```
#include <LiquidCrystal.h>
#include <Wire.h>
#include <EEPROM.h>
#include <Keypad.h>
LiquidCrystal lcd(13, 12, 11, 10, 9, 8);
const int bankButton = 53; //Pin for Bank Button
const byte rows = 16;
const byte cols = 3;
char keys[rows][cols] = {
{'a','q','7'},
{'b','r','8'},
{'c','s','9'},
{'d','t','0'},
{'e','u','~'},
{'f','v','!'},
{'g','w','@'},
{'h','x','#'},
{'i','y','$'},
{'j','z','%'},
{'k','1','^'},
{'l','2','&'},
{'m','3','*'},
{'n','4','('},
{'o','5',')'},
{'p','6','+'}
};
byte rowPins[rows] = {30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45};
byte colPins[cols] = {46,47,48};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols);
int relayPin[16] = {14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29};
int ledPin[16] = {A0,A1,2,3,4,5,6,7,A8,A9,A10,A11,A12,A13,A14,A15};
int i;
int readOut;
int numberOfPedal = 16;
```

```
int bankVariable = 20;
int numberOfBanks = 5;
int bank = 0; //Starting Bank Number
boolean lastBankButton = LOW;
boolean currentBankButton = LOW;
/* SETUP */
void setup() {
lcd.begin (16,2);
lcd.print("Line 1");
lcd.setCursor(0,1);
lcd.print("Line 2");
for(i=0; i<numberOfPedal; i++) {
pinMode(relayPin[i], OUTPUT);
pinMode(ledPin[i], OUTPUT);
digitalWrite(relayPin[i],LOW); //turn ON all relays
digitalWrite(ledPin[i],LOW); //turn ON all preset LEDs
};
delay(500);
for(i=0; i<numberOfPedal; i++) {
pinMode(relayPin[i], OUTPUT);
pinMode(ledPin[i], OUTPUT);
digitalWrite(relayPin[i],HIGH); //turn OFF all relays
digitalWrite(ledPin[i],HIGH); //turn OFF all preset LEDs
};
pinMode(bankButton, INPUT);
Serial.begin(9600);
delay(1000);
lcd.clear();
lcd.print("Press any preset");
lcd.setCursor(0,1);
lcd.print("to start");
delay(1000);
}
/* Bank Button Debounce */
boolean bankdebounce(boolean last) {
boolean current = digitalRead(bankButton);
if (last != current) {
delay(5);
current = digitalRead(bankButton);
```

```
}
return current;
}
/* LOOP */
void loop() {
currentBankButton = bankdebounce(lastBankButton);
if (lastBankButton == LOW && currentBankButton == HIGH) {
bank ++;
lcd.clear();
lcd.print("Bank ");
lcd.print(bank);
lcd.setCursor(0,1);
lcd.print("Press Any Preset");
if (bank >= numberOfBanks) {
bank = 0;
}
}
lastBankButton = currentBankButton;
char key = keypad.getKey();
if(key) { // Check for a valid key.
switch (key) {
case 'a': writeOut(0);
break;
case 'b': writeOut(1);
break;
case 'c': writeOut(2);
break;
case 'd': writeOut(3);
break;
case 'e': writeOut(4);
break;
case 'f': writeOut(5);
break;
case 'g': writeOut(6);
break;
case 'h': writeOut(7);
break;
case 'i': writeOut(8);
break;
```

```
case 'j': writeOut(9);
break;
case 'k': writeOut(10);
break;
case 'l': writeOut(11);
break;
case 'm': writeOut(12);
break;
case 'n': writeOut(13);
break;
case 'o': writeOut(14);
break;
case 'p': writeOut(15);
break;
case 'q': memory(100+(bank*bankVariable),0);
break;
case 'r': memory(200+(bank*bankVariable),1);
break;
case 's': memory(300+(bank*bankVariable),2);
break;
case 't': memory(400+(bank*bankVariable),3);
break;
case 'u': memory(500+(bank*bankVariable),4);
break;
case 'v': memory(600+(bank*bankVariable),5);
break;
case 'w': memory(700+(bank*bankVariable),6);
break;
case 'x': memory(800+(bank*bankVariable),7);
break;
case 'y': memory(900+(bank*bankVariable),8);
break;
case 'z': memory(1000+(bank*bankVariable),9);
break;
case '1': memory(1100+(bank*bankVariable),10);
break;
case '2': memory(1200+(bank*bankVariable),11);
break;
case '3': memory(1300+(bank*bankVariable),12);
```

```
break;
case '4': memory(1400+(bank*bankVariable),13);
break;
case '5': memory(1500+(bank*bankVariable),14);
break;
case '6': memory(1600+(bank*bankVariable),15);
break;
case '7': readPreset(100+(bank*bankVariable),0);
break;
case '8': readPreset(200+(bank*bankVariable),1);
break;
case '9': readPreset(300+(bank*bankVariable),2);
break;
case '0': readPreset(400+(bank*bankVariable),3);
break;
case '~': readPreset(500+(bank*bankVariable),4);
break;
case '!': readPreset(600+(bank*bankVariable),5);
break;
case '@': readPreset(700+(bank*bankVariable),6);
break;
case '#': readPreset(800+(bank*bankVariable),7);
break;
case '$': readPreset(900+(bank*bankVariable),8);
break;
case '%': readPreset(1000+(bank*bankVariable),9);
break;
case '^': readPreset(1100+(bank*bankVariable),10);
break;
case '&': readPreset(1200+(bank*bankVariable),11);
break;
case '*': readPreset(1300+(bank*bankVariable),12);
break;
case '(': readPreset(1400+(bank*bankVariable),13);
break;
case ')': readPreset(1500+(bank*bankVariable),14);
break;
case '+': readPreset(1600+(bank*bankVariable),15);
break;
```

```
}
}
}
/* Use Presets */
void readPreset(int addr, int led) {
for(i=0; i<numberOfPedal; i++) {
digitalWrite(relayPin[i], EEPROM.read((addr)+i));
digitalWrite(ledPin[i], HIGH);
digitalWrite(ledPin[led], LOW);
}
lcd.clear();
lcd.print("Current Bank: ");
lcd.print(bank);
lcd.setCursor(0,1);
lcd.print("Preset: ");
lcd.print(led + 1);
}
/* Select Preset and Store */
void memory(int addr, int led) {
for(i=0; i<numberOfPedal; i++) {
EEPROM.write((addr) + i, digitalRead(relayPin[i]));
digitalWrite(ledPin[i], HIGH);
}
lcd.clear();
lcd.print("Program saved to");
lcd.setCursor(0,1);
lcd.print("Bank ");
lcd.print(bank);
lcd.print(", Preset ");
lcd.print(led + 1);
digitalWrite(ledPin[led], HIGH);
delay(100);
digitalWrite(ledPin[led], LOW);
delay(100);
digitalWrite(ledPin[led], HIGH);
delay(100);
digitalWrite(ledPin[led], LOW);
delay(100);
digitalWrite(ledPin[led], HIGH);
```

```
delay(100);
digitalWrite(ledPin[led], LOW);
delay(100);
digitalWrite(ledPin[led], HIGH);
lcd.clear();
}
/* Choose Loops/Relays */
void writeOut(int relay)
{
digitalWrite(relayPin[relay], !digitalRead(relayPin[relay]));
//digitalWrite(ledPin[relay], !digitalRead(relayPin[relay]));
lcd.clear();
lcd.print("Current Bank: ");
lcd.print(bank);
lcd.setCursor(0,1);
lcd.print("Choose Loops: ");
lcd.print(relay + 1);
}
```

Hi CarraN,

I'm a complete newbie on the arduino world, and I'm also a guitar player :)
I'd like to control 10 loops with thin project. Can you give me a hand with the code, please?
Also,
is it possible to add some kind of control over the text that is
displayed? Maybe costumize the name of the preset, change the midi PC
message...

If you could help me, I'd be very gratefull
Cheers!
Mike

Hola Miguel

all you need is in this instructable. The display stuff is a follower's addon of the initial project that I've included in the code but without any waranty from my part. So if you have a specific question you can still post here, maybe someone can will help you!
I suggest reading first some comments related to.

hope this will help

cheers

carran

Can this looper be use as footswitch? I mean changing loops AND amp channels?

yes it can! Just wire off those amp channel relays! Similar to the "breaking technique" explained on

this instructable.

cheers

I've checked YouTube videos and there is a big delay between button press and changing loop. It's unavoidable?

Hi!

no problem of any latency on my side! You'll do not have neither!

Youtube isn't lip synching accurate, maybe you 've experienced this...