Arduino Programable 5 Pedal Switcher

by CarraN

Searching for a simple programable pedal switcher over the net I was
amazed about the overpriced gear around there... so I came out with DIY solution and found several great projects but found 'em too complicated (PCB etching...) or maybe too overpriced for my needs. All I needed at that time was the simplest possible programable true bypass relay switcher for 4 pedals but found a way to switch up to 5 pedals with the "nano" version of arduino UNO. With 2 Arduino nanos (thru serial communication) you can surely extend this as up to 8 switchers! (to see: second part of this instructable...). Posibility to add MIDI program change out on pin 1.

Addendum: connect a led in parallel of one of these little on the relay board (caution: check the polarity!). This manner you'll know which relay is on or off when programing. Tip: I 've used the hole of the screw near the inboard led to drill thru the chassis and put it the parallel one on. (see pictures on my second instructable)

After noob searching for the "code-part" of this project over internet and spent hours and hours of led debuging (hey, I'm NOT a coder, just a r'n'r guitar player!) I got out with an amazing library named "keypad.h". Initially intended to manage secret code doors within a keypad, we'll see how simple is to "fool" the code with a 3 way selector, so with only 5 momentary switches we'll can access up to 15 functions and store 5 presets on internal EEPROM! The hardware parts are current sees over Ebay / Google so please don't ask me "how and where can I buy this thing or this other..." ! Google is your friend and Ebay too!

-enjoy the art of rocking pedals with one footpress !!!

see more hardware pictures made by some followers and me

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...

USE:

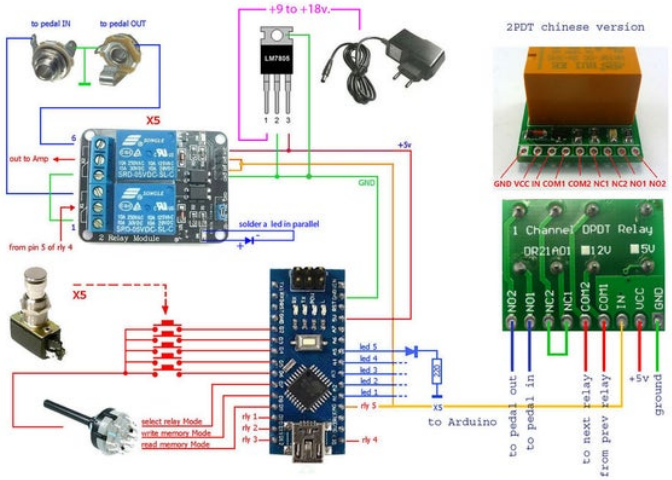
-position "A" (select switches) from the selector (cases 'a' 'b' 'c' 'd' 'e' on the sketch) -> select wich pedal to loop ON (1,2... 5). When finished, turn the selector to ->

-position "B" (preset memory mode -cases 'f' 'g' 'h' 'i' 'j') -> select the number of the preset (1 to 5) where you want to preserve the loops (the led of the preset you choose will lit 2 times. Caution: don't press the switch more than once because you'll reset the preset (all ' 0,0,0,0,0')!

-Rotate the selector back to position 'A', select another pedal combination and store it on next free preset on position "B". You can repeat this procedure 5 times. Once you have your 5 combination stored, rotate to->

-position "C" (cases 'k' 'l' 'm' 'n' 'o')-> read the preset (1 to 5) you memorized previously

1. a friend of mine punked it



Step 1: Arduino Programable 5 Pedal Switcher

So here are the parts:

-1 x arduino Nano with Atmega328 built-in (~4 to 5$)

-1 x 9v to 18v / 3 amp power supply (~10 $)

-1 x LM7805 or alike

-3 x ways rotary selector (~4$)

-5 x SPST Momentary Footswitches (~15$)

-5 x chinese double relay modules (~ 15$) or better, DPDT version (much less noisy)

-12 x chassis 6.3 female jack (~20$)

-10 x leds (half of a different color)

-5 x 220 to 560 ohms resistor (peanuts...)

-some spare cable (...)

~70$ if you don't have any of these as spare

EEPROM memory in case of debugging

You"l have first to download and install the keypad.h library here:

A big **THANK YOU** to Armando Colangelo aka 4Knobs Effects for this code . Note his project is based on a different approach that ommit the 3 positions switch and therefore adds 1 more selecting switch (6 instead of 5).

A video is available on youtube showing his awesome work.

(not included: metal chassis or woodbox or whatever...)

---

and here are the tools:

-plyers

-soldering station

-eventually some tools for wood/metal/plastic working

Here's the code part: (many thanks to Anton Efremoff for his feedback and suggestions) Tip: don't forget to add "//" on these two lines if you want to save your preset after booting it up!

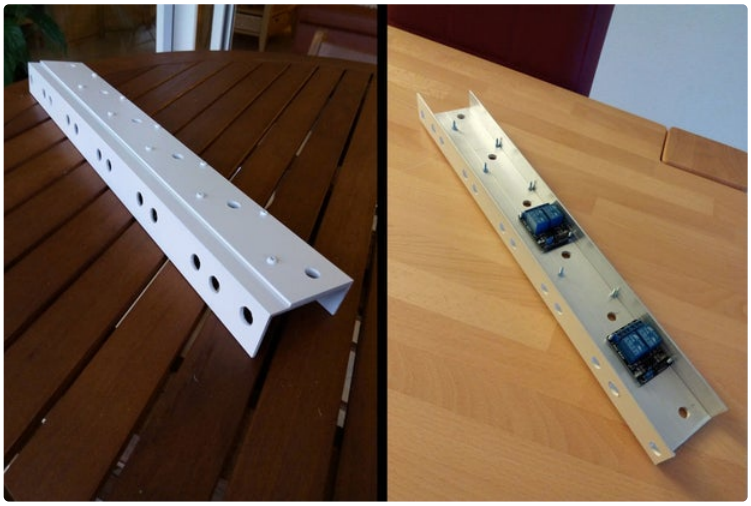
//for (int =0;i<512;i++)
//EEPROM.write(i,0); It's here just for initialize

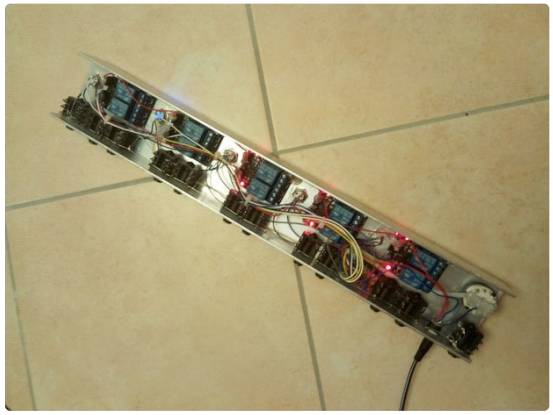
Another vid from follower Massam here showing working simulation

---

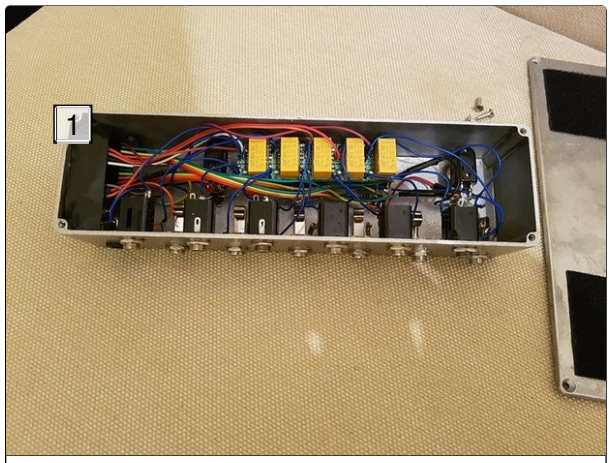
check out the outstanding band I play with:

ZeK

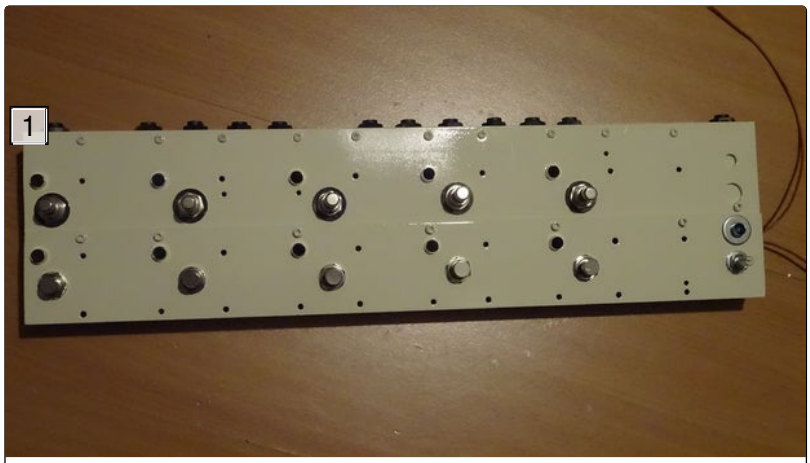




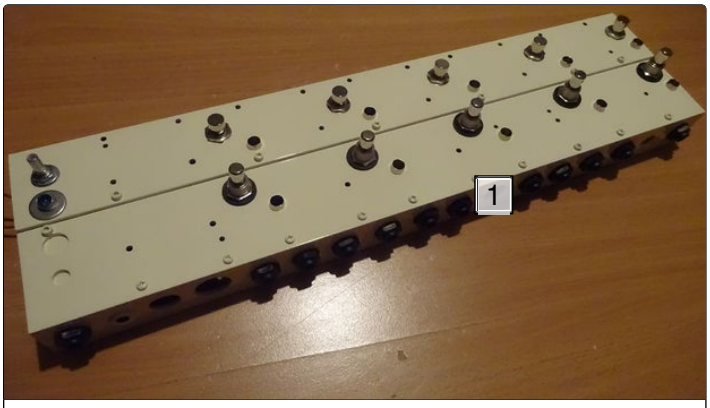
1. made by livewir3



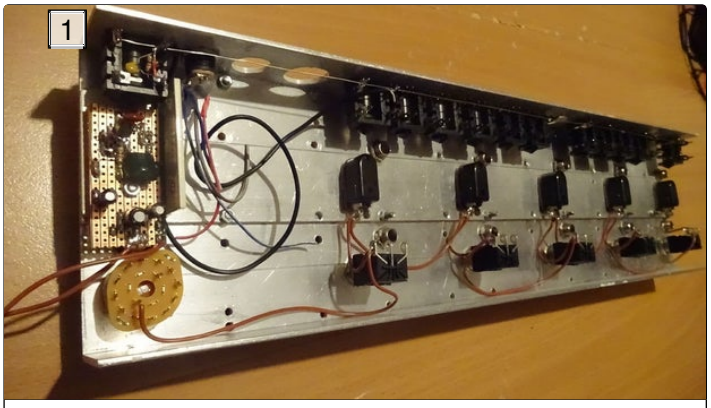
1. made by livewir3



1. 8 presets variant (mega 2560), see : https://www.instructables.com/id/Arduino-based-8-loops-pedal-switcher/



1. midi implementation



1. AMZ mosfet booster for improving noise ratio
http://www.muzique.com/schem/mosfet.htm

https://www.instructabl... Download

https://www.instructabl... Download

hi CarraN, i am not sure to find the right instruction in this here.

I want to create a 4 way looper for 1 instrument,

it means , at one side 1 input jacket, at the other side 1 output jacket.

in the middle my arduino mega board with different switches, different LED, in the best way to serv the switches with my foot.

is your work suitable for me ?

kindly regards

Hein66

This is my build of the 5 pedal switcher.

I couldn't find the right size of enclosure. This one was hard to fit everything in. but i got it all to fit anyhow.

I've added an extra footswitch / LED to add an extra feature, like a boost or bypass all. I just havn't hooked it up yet. But the switcher is done for now.

Did you use the code of five_pedal_looper.ino that comes at the end of the post ..?

it marks me a series of conflicts...

cool! yes it seem pretty little !
Do you observe some noise issues with your pedals on (mostly with overdrives)?
Did you use the Armando's code or mine?
cheers

I used your code for this project. This looper only has a little noise from the relays as you mentioned, but this goes away with a buffer before. I didn't use any decoupling capacitors either. one of my distortion pedals creates a very small going noise, but it's too quiet to warrant using

decoupling caps. Thanks for posting this project, it so much fun and very hand to have. On my pedal board.

i have problem with relay mode there is delay on switching. i need to double click the switch to engage the relay any solutions?

Hi

are you sure you
have choose the right buttons? momentary ones or "Toggle button" instead
of on/off buttons... that means when you apply pressure, switch is "on"
and when depressed switch is always "off".

cheers

yes sir. another question do you experience loud popping on amps because of relays?

neither popping, nor delay, just some noise without booster and some distortion pedals...

I am starting this project soon. However, I have a question. Why do I need the VOltage regulator LM7805?

In this project it states buying one separately. I am looking at using an Arduino Mega and that seems to have one built in. Will I still need a separate one?

Also will the power supply power everything including the Relays or does the relay need a separate power supply. I am thinking of getting the dpdt relays as some people seem to have experienced noise issues with the cheap ones.

Regards

Jim

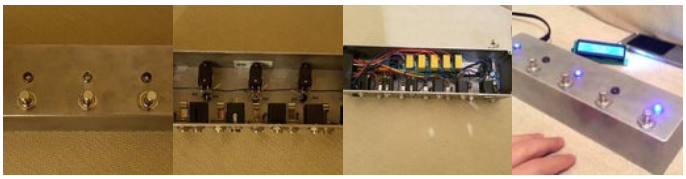
Yes the regulator included in the Mega doesn't have enough power for the relays in my opinion but It's your to test it and your welcome to share if it does. And yes I confirm the relays I've choosen in this instructable had some noise issues. Take the ones Liwewire3 installed in his project. Cheers!

hi all did a quick write up on the looper build and use. may do an update later for now here it is. including schematic and code. cheers

https://tt305.wordpress.com/

nice!

hey guys. here some pictures and the nearly finished product. 5 loops , 5 presets, 3 banks for now but could be expanded for more. went with different dpdt relays as the spdt took to much space and a lot of current.



NIIICE! good work mate! Can I include some of your pictures in this instructable?

I too have ordered the same dpdt and add some midi new features but all this soon..

cheers

yes sure feel free. i would like to hear more about midi. i need to get my head around it, would be interesting. any questions feel free. these relays work fine. you need to inverse the logic though to turn them on.

well thank you! So these dpdt need to be LOW to be ON? Mines are near from my house so I'll can test 'em soon. There are two ways MIDI works: the old one with the 5 pins DIN socket and easy to build from this same project, and the usb one I made in my other instructable

https://www.instructables.com/id/Arduino-Based-Bos...

Can be adapted pretty easy with this project too...

cheers

well the ones i got from ebay came with a pcb and couple component. so it uses a bjt npn transistor so requires a high to turn on. if you designing yourself i guess a pnp/ p channel fet etc can be used. i just went with what it was and set the logic in the code to put out a high.
im a bit consernerd about the relay noise coming from relays but still working on it. i will draw a schematic on how to connect the thing that may help people how is connected.

thanks for the link. EO

Hey! I wanted to click "made it" because I got something that sorta works, but I'm still working on it, but I couldn't get the upload to work As opposed to most builders here (and on the other tutorial), for the prototype I didn't use the off-the-shelf relay module, but designed one from scratch.

This, to my surprise, manifests no switching noise whatsoever. What I've noticed instead is that it picks hum like crazy and acts like a giant antenna. Me not putting things in an enclosure might be the culprit, so I'll try to fix that and post and update as well.

I'm also waiting for the ready-made relay modules to compare them to my own solution.

I'm pretty sure you're right: a lot of noise come from those chinese relay modules! Hey, they're aren't cheap for any reason...

I ordered other type, in fact dpdt ones, this way only half of the relays involved here are needed.
https://fr.aliexpress.com/item/DR21B01-2-2x-Ultra-...

cheers

ps: please send us some pictures of your work when finished

There we go, managed to upload it this time. I've drawn the board by hand for this one, so it's kinda wonky, but I couldn't wait until monday to print it at work, lol.

Just got back from a trip, will try to fit an enclosure this week and see if it helps.

Oh and for the DPDT ones, I've ordered them as well, but might buy a few locally and try to make a driver for them as well, to compare to the ready modules. The circuit in the Chinese modules looks perfectly fine, so if they're really that bad, I'm suspecting the build quality issues as well. Perhaps worth rechecking all the paths and components?

I put it in an enclosure and it seems to be working flawlessly. Still waiting for the parts for the big prototype.

Hi CarraN, im seeing some issues with the relay module i have and wondering if anyone seeing same things. im using the 8 relay module intead of 4 two relay modules you use for my 4 channel looper but im seeing a strange voltage drop if all 4 loops 8 relays are ON. the regulator im using is capable of 5V 2A output. im thinking is the relays in this 8 relay module or all behave this way if you have them all on. how re you powering your 5 or 8 loop switcher and have you see any of this behaviour?

cheers

Hi

how much current can handle your power transformer ? Try a 12v/3A
and maybe try another more powerfull regulator. Do you fix it a regulator radiator? You can fix your regulator to the metallic grounded chassis. By the way it seems an issue with power... See schematic again, it's all in there!

cheers

im using a l7805 2A to power the relay coils which is powered to a 9V 1A boss psa power suply. if i turn all 8 relays on and leave them on after a minute or so the relays go of as the voltage has droped to 3V from 5V regulator. im thinking that alot of power its been drawn to cause the voltage to drop. the connections are the same as you have but if you are not seeing any issues im thiking is these relays im using.

forget your boss power supply right now and go for a 9 or 12v, 3A ! Relays naturally eat a lot of power. And put your l7805 a eatsink or fix it to the chassis (if it's made of metal).

My relays take around 80mA at 5V when closed (and they're the bulky ones capable of hundreds of volts). 3A for 8 of them doesn't really sound right, unless the assembled relay boards are realyl that bad.

hi bananu7 that's more acceptable. i got these cheap from ebay to try them but not worth it. 80mA is still a lot but if it is no causing issues i would have gone with it. i think the 8 relay module i have is faulty and im going with DPDT relays. thanks for the msg

thanks for the reply CarraN. I wanted to know if anyone seen similar problems with these relay types. Throwing more power is not a solution for me. I'm powering my hole board with 5A supply and using 3A to power a switcher is not a good solution. These relay coils are expected to draw 35 mA each going by spec. So all 8 ON you would expect no more than 0.5 A current draw. Not that I believe the spec And you can't put more than 5V on them or the coil would go if they are speced at 5V. If no one here seen this issue I have a faulty relay module. thanks again for your help.

Great project! Is it possible to control the relays direct from the Arduino without the use of footswitches?

TLDR (and not entirely related to this project)

I'm building a switcher that controls the FX for every patch on my Boss MS-3 via an Arduino USB Host. Why you ask? Mainly because switching between patches causes a ~28ms audible delay, and while it has a stereo output, you can't select what effects go out of which output (ie FX left, clean right). Sounds superfluous, but delving into this project has unlocked so many more possibilities for my MS-3!

The MS-3 has 3 loop send/returns which is very cool, however, I'd also like to split the signal before the MS3, sending a clean signal to an acoustic simulator->DI (and ***not*** returning back to the signal chain). It's basically simulating the MOEN GEC9, except it's not handling any audio signals - yet! I'm setting presets like I do with the GEC9, except they toggle the FX on the MS-3's currently selected patch giving one MS-3 patch multiple presets within it (in my case it'll be 5, just like this project).

SO if Arduino can manage the relays directly this could open up more possibilities, namely expanding the MS3's loops from 3 to ...well, the sky is potentially the limit! ;)

Oh yeah! sounds interesting, I dream it and they build it! I didn't know that MS-3 until you told me about, thank you Alex!

So now your question: if I understood well, you'd want to "pilot" 5 external relays by the 4 presets footswitches of the MS-3? is that right? Obviously you'll must add an arduino between and (surely) It'll respond to midi messages thru usb... You're a lucky man because I've done another instructable that can handle midi messages between arduino and boss devices, it's here:

https://www.instructables.com/id/Arduino-Based-Bos...

It'll be interesting to melt this two projects in another new one! If you have more questions don't be affraid to ask.

see you

carraN

Almost right! It's a bit confusing I know. I want to pilot ***both*** MS3 and a relay (or multiple relays) from a separate pedal board (with Adruino) with 5 footswitches which are programmable subpatches (using buttons, like the GEC9) of the currently selected MS3 patch.

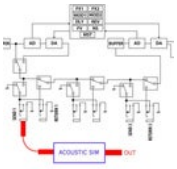
I am currently toggling 12 MS3 effects and save them to Arduino - *success*! - but I'm now thinking I could make a 13th to toggle a relay that essentially allows/disallows a clean signal to be sent to my acoustic simulator. So my question is: **will a digital out from Arduino be able to toggle the relay state?** *fingers crossed*

ok! so yes!

I'm onto it!

Hi !

regarding the block diagram of the ms3 you can already do that, see below!



This is true - this was one of the first things I tested when I realised the in-built acoustic sim wasn't able to be routed to a separate main output. So yes this approach works! But it means I sacrifice one precious loop. 3 is already not enough (yes I know, I'm greedy) ;) Being able to have a 4th would be great. There's even potential to expand to many more loops, where you could send from MS3 L1 into a new chain managed by arduino, and back to MS3 L1 -- but this is really going to crazy town... or is it?

Every time I have questions about some "new" hardware to aquire or whatever , I apply this simple rule: Do I want it or do I need it ? So the answer come like the rain in springtime!

cheers

hi CarraN,

thanks for the fun project and great instructions. I made some changes/updates to the code to use a click/hold button to substitute the 3way switch so the control can be done by foot (one click =

bank control, presshold = go through A,B,C stages you had) i will add the code below if someone wants to use it. i have a prototype built waiting in hardware for the final thing. cheers again.

code here

==========

```
/* programable stompbox looper. Original design and sketch by CarraN. Corrected, updated and upgraded by Pascal Paquay */
/* programable stompbox looper. Original design and sketch by CarraN. updated and upgraded by EltonO 09/10/17 */

#define LCDSCREEN 1 //#define LCDSCREEN 0 if not using lcd display

#include <Wire.h>
#include <EEPROM.h>
#include <Keypad.h>

#if LCDSCREEN == 1
#include <LiquidCrystal_I2C.h>
/*********************************************************/
LiquidCrystal_I2C lcd(0x3F, 16, 2); // set the LCD address to 0x3f for a 16 chars and 2 line display
/*********************************************************/
#endif

const int ctrlButton = A6; //Pin for controll Button
const byte rows = 5;
const byte cols = 1;
char keys [rows][cols] = {'0','1','2','3','4'};
byte rowPins[rows] = {2,3,4,5,6};
byte colPins[cols] = {7};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols);
int ledPin[5] = {A3,A2,A1,A0,13};
int relayPin[5] = {8,9,10,11,12};
byte midiChannel = 0;
int i;
int readOut;
int numberOfPedal = 5; /*adapt this number to your needs = number of loop pedals */
int bankVariable = 5;
int numberOfBanks = 5;
int bank = 0; //Starting Bank Number
int presetSelect = 3;
int preset = 0;
int loop1 = 0;
int loop2 = 0;
int loop3 = 0;
int loop4 = 0;
int loop5 = 0;
int bounceTime = 20;
int holdTime = 1000;
int doubleTime = 1000;
int lastReading = HIGH;
int hold = 0;
int single = 0;
```

```
long onTime = 0;
long lastSwitchTime = 0;
boolean currentRead = LOW;
boolean lastRead = LOW;

void setup() {
for(i=0; i<numberOfPedal; i++)
{
pinMode(ctrlButton, INPUT_PULLUP);
pinMode(relayPin[i], OUTPUT);
pinMode(ledPin[i], OUTPUT);
digitalWrite(relayPin[i],LOW); //pullup all relay outputs in case off low level relayboard
digitalWrite(ledPin[i],HIGH); //pullup all relay outputs in case off low level relayboard
delay(100);
digitalWrite(relayPin[i],HIGH); //pullup all relay outputs in case off low level relayboard
digitalWrite(ledPin[i],LOW); //pullup all relay outputs in case off low level relayboard

}

Serial.begin(31250); /* for midi communication - pin 1 TX */
//Serial.begin(9600); /* for serial communication - pin RX/ TX */

// *for (int i = 0; i < 512; i++) // erase eeprom (optional)
// EEPROM.write(i, 0); */

#if LCDSCREEN == 1
lcd.clear();
lcd.init(); //initialize the lcd
lcd.backlight(); //open the backlight
lcd.print("| DIY FX 5 |");
lcd.setCursor(0,1);
lcd.print("| LOOPER |");
delay(3000);
lcd.setCursor(0,0);
lcd.print(" | |: ");
lcd.setCursor(0,1);
lcd.print(" | |: ");
lcd.setCursor(5,0);
lcd.print("Pset");
lcd.setCursor(0,0);
lcd.print("Bank");
lcdprint ();
#endif
}

/*******************************************************/
boolean debounce(boolean last,int ctrlButton) {
boolean current = analogRead(ctrlButton);
if (last != current) {
delay(50);
current = analogRead(ctrlButton);
}
return current;
```

```
}
/*******************************************************/

void loop() {

currentRead = debounce(lastRead,ctrlButton);
// int currentRead = digitalRead(ctrlButton);

//first pressed
if (currentRead == LOW && lastRead == HIGH ) {
onTime = millis();
}
singlePress();

if (currentRead == LOW && lastRead == LOW ) {
if ((millis() - onTime) > holdTime) {
hold = 1;
}
}

if (currentRead == HIGH && lastRead == LOW ) {

if (((millis() - onTime) > bounceTime) && hold != 1) {

doublePress();
}

holdPress();
}

char key = keypad.getKey();

if((key)&&(preset == 1)) // Check for a valid key.
{

switch (key)
{
case '0': // a to x
writeOut(0); // relay
break;
case '1':
writeOut(1);
break;
case '2':
writeOut(2);
break;
case '3':
writeOut(3);
break;
case '4':
writeOut(4);
break;
```

```
}
}
/****************************** STORE PRESET MODE */
if(key && preset == 2) // Check for a valid key.
{

switch (key)
{
case '0':
memory(100+(bank*bankVariable),0); //addr, led
break;
case '1':
memory(200+(bank*bankVariable),1);
break;
case '2':
memory(300+(bank*bankVariable),2);
break;
case '3':
memory(400+(bank*bankVariable),3);
break;
case '4':
memory(500+(bank*bankVariable),4);
break;
}

}
/****************************** READ PRESET MODE */
if(key && preset == 3||preset == 0) // Check for a valid key.
{

switch (key)
{
case '0':
readPreset(100+(bank*bankVariable), 1, 0); // addr, pcNum, relay
break;
case '1':
readPreset(200+(bank*bankVariable), 2, 1);
break;
case '2':
readPreset(300+(bank*bankVariable), 3, 2);
break;
case '3':
readPreset(400+(bank*bankVariable), 4, 3);
break;
case '4':
readPreset(500+(bank*bankVariable), 5, 4);
break;
}
}

delay (100);
lastRead = currentRead;
```

```
}

void singlePress() {

if (single == 1 && (millis() - lastSwitchTime) > doubleTime) {

// Serial.println("single press");
bank++;

#if LCDSCREEN == 1
lcdprint ();
#endif


if (bank == 1) { ledtoggle(8); }
if (bank == 2) { ledtoggle(9); }
if (bank == 3) { ledtoggle(10); }
if (bank == 4) { ledtoggle(11); }
if (bank == 5||bank == 0) { ledtoggle(12); }

if (bank >= numberOfBanks) {
bank = 0;

}
single = 0;
}
}
void doublePress() {

if ((millis() - lastSwitchTime) >= doubleTime) {
single = 1;
lastSwitchTime = millis();
return;
}

if ((millis() - lastSwitchTime) < doubleTime) {
// Serial.println("double press");

digitalWrite(8, LOW);
digitalWrite(10, LOW);
digitalWrite(12, LOW);

#if LCDSCREEN == 1
lcdprint ();
#endif


single = 0;
lastSwitchTime = millis();
}

}
void holdPress() {
```

```
if (hold == 1) {
preset++;
if (preset == 1) {

ledtoggle(A3);
#if LCDSCREEN == 1
lcd.setCursor(5,0);
lcd.print("Set ");
#endif

}
if (preset == 2) {
ledtoggle(A2);
#if LCDSCREEN == 1
lcd.setCursor(5,0);
lcd.print("Prog");
#endif


}
if (preset == 3||preset == 0) {

ledtoggle(A1);

#if LCDSCREEN == 1
lcd.setCursor(5,0);
lcd.print("Pset");
#endif

}


if (preset >= presetSelect) {
preset = 0;

}
hold = 0;
}
}
/*********************************************************/
void midiProg(byte status, int data)
{
Serial.write(status);
Serial.write(data);
}

/*********************************************************/

void memory(int addr, int led)
{
for(i=0; i<numberOfPedal; i++)
{
```

```
EEPROM.write((addr) + i, digitalRead(relayPin[i]));
digitalWrite(ledPin[i], LOW);

}
ledtoggle(ledPin[led]);
#if LCDSCREEN == 1
lcd.setCursor(8,1);
lcd.print(led + 1);
#endif

}

/********************************************************/

void writeOut(int relay)
{
digitalWrite(relayPin[relay], !digitalRead(relayPin[relay]));
digitalWrite(ledPin[relay], !digitalRead(relayPin[relay]));

#if LCDSCREEN == 1
lcdprint ();
#endif


}

/********************************************************/

void readPreset(int addr, int pcNum, int led)
{
for(i=0; i<numberOfPedal; i++)
{
digitalWrite(relayPin[i], EEPROM.read((addr)+i));
digitalWrite(ledPin[i], LOW);
digitalWrite(ledPin[led], HIGH);
}

#if LCDSCREEN == 1
lcdprint ();
lcd.setCursor(8,1);
lcd.print(led + 1);
#endif


midiProg(0xC0, pcNum +1); /* send midi change program 1 */
}

void ledtoggle(int led)
{
digitalWrite(led, !digitalRead(led));
delay(100);
digitalWrite(led, !digitalRead(led));
delay(100);
```

```
digitalWrite(led, !digitalRead(led));
delay(100);
digitalWrite(led, !digitalRead(led));
delay(100);
digitalWrite(led, !digitalRead(led));
delay(100);
digitalWrite(led, !digitalRead(led));
delay(100);

}

void lcdprint() {
#if LCDSCREEN == 1
loop1 = !digitalRead (8);
loop2 = !digitalRead (9);
loop3 = !digitalRead (10);
loop4 = !digitalRead (11);
loop5 = !digitalRead (12);

lcd.setCursor(3,1);
lcd.print(bank);
lcd.setCursor(8,1);
lcd.print(">");
lcd.setCursor(11,0);
lcd.print("Loops");
lcd.setCursor(11,1);
lcd.print(loop1);
lcd.setCursor(12,1);
lcd.print(loop2);
lcd.setCursor(13,1);
lcd.print(loop3);
lcd.setCursor(14,1);
lcd.print(loop4);
lcd.setCursor(15,1);
lcd.print(loop5);
#endif


}
```

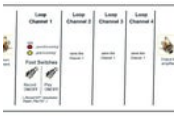
Armando did that yet (see the loopino code here) but thanks mate!

cheers

I did check armondos interestning work but i wanted something simple for what i needed to do and get to speed with arduino :)

that is my plan

hein66



OKAAYYY... you're not in the right place, I'm affraid! What you're searching for is a PHRASE LOOPER not a pedal switcher. This project is for creating presets among 5 external pedals like distortion -> chorus or distortion -> delay-> comp...

sorry

yes , i am afraid too.
thx for your comments. do you suppose there is a way to realize my plan with an ardiuno or an raspberry pi ?

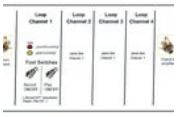
nope

cheers

Hi CarraN, Ok, that is a good idea,So I have made a short image. Additional
I will mention, that always must be a connect through between input jack
and output jack.

Is your
solution suitable for me ?

Hein66



Hi CarraN, thank you for your quickly response.

what i dont understand is, why do you use 5 pedals ?

For my 1 instrument (= 1 input jacket) , i should need 1 double relay, is'n it ?

I understand partially your sketch, but in the image, i cannot find out the relay connections of the relay and the mega board.

what can i do now ?

regards

hein66

5 pedals is the max that can manage Arduino uno. But if you want less -4 in your case- just ignore the fifth relay. That said you don't even need a mega for that purpose. So well, the input jack is connected on the pin 2 of the first relay board and the out jack is connected to the pin 5 of the last realy board like in the schematic. You don't need to cable the ground, just the live cable. The connections between the Uno and relay boards is as you wish, just pay attention to select the good digital pins in your sketch. Hope it'll help you.

Hi CarraN,
thank you for your patience with me. It is possible that we have a different wording and understanding of this issue ?
I am confused about the word “pedal” . In your schematic input jack is marked as “to pedal IN” , output jack as “to pedal OUT” .
What do you mean with this ? what is the function of your "pedal" ?

“to pedal IN” must be in my understanding “instrument IN” .
And “to pedal OUT” must be “to amplifier”

The only “pedal” I have are the foot switches to change the loop channel.
My plan is to have one input and one output jack.
Do I need for every couple of in/out jacks one relay
Or for every loop channel ?
What is correct ?
Sorry but I am musican, no specially technic freak.
Hein66

what do you want to do?

Hi CarraN,

Ok, that is a good idea,

So I have made a short image.

Additional
I will mention, that always must be a connect through between input jack and output jack.

Is your
solution suitable for me ?

Hein66