My Relay FX looper with MidiBox - Flo http://www.midibox.org/dokuwiki/flo

# flo user page

... under construction ... last edit: 2009-07-26

# Guitar FX looper with MidiBox

WARNING: THIS PROJECT IS IN PROGRESS SO IT IS NOT FINISHED YET!
http://www.midibox.org/forum/index.php/topic,11705.0.html

# Requirements

## In General

### FX Bypass Loopers

Multiple true bypass guitar FX loopers in one enclosure.

From: http://www.voodoolab.com/manuals/gcx_manual.pdf
Chapters “Audio Loops” (page 7-9) and “Using Loops as Switches” (page 10-11), explains what can be done with a “regular” FX looper. I want my FX switcher to support these options so I guess I'll be modeling it to the VoodooLab GCX concerning the looper hardware but not its Midi implementation.

One looper can be used for several purposes:

#### FX bypass

A “looper” is basically a true bypass switch that can be used to bypass a guitar effect that is connected to it. So the effect can be switched in and out of the signal path.

- If the loop is OFF, the FX is OUT of the signal path.
- If the loop is ON, the FX is IN the signal path.

When putting multiple FXs in series, the whole FX series can be switched in and out of the loop. In this series, each FX can still be turned on and off individually with its own bypass switch.

#### A/B switching

- Select between two signal sources = A-in / B-in → Output
- Route a signal to one of two outputs = Input → A-out / B-out
- Muting Send. The loop can be used to mute the signal going to an FX or amp.

A/B switching can be handy to route the guitar either to the FXs & amp or to a tuner. Also nice is to be able to select between amps.

#### Sending a signal to multiple outputs

Use multiple loops to send a signal to multiple outputs where each one can be individually enabled/disabled.

#### Use a loop as a switch

A loop can act as a (foot-)switch that can be used to switch amp channels and bypassing FX processors.

These come in two types:

- Latched: Latching switch has two possible states, either open (tip and sleeve unconnected), or closed (tip and sleeve are connected together).
- Momentary: Some devices require momentary type switches. A momentary switch changes state (from ON to OFF or from OFF to ON) by closing for a short time and then opening again. These are most often used for bypass switches on digital effects.

Normally Open and Normally Closed:

A normally open switch is open when it's off. A normally closed switch is closed when it's off. In most cases, you will use normally open (plug into the OUT/N.O. jack). If the indicator does not correspond correctly to an FX state, use normally closed (plug into the SEND/N.C. jack).

## Optional: FX order switcher

Perhaps add a FX order switcher. This can be nice extra… have to think about it.
Using 3 loopers, FX order switching can be done but that's a lot of loopers for this task making it “expensive”.
A dedicated FX order switcher needs 3 relays internally and some jacks: IN - OUT - SEND1 - RETURN1 - SEND2 - RETURN2.

## Keep it simple

Try to avoid a user interface

# Audio Inputs and Outputs

For independent loopers, each looper must have:

- In, Out, Send, Return

For loopers in a series FX chain:

- Send and Return per looper & only 1 In at the beginning of the chain and 1 Out at the end of the chain.

# Midi Implementation

Control the FX loopers via a MIDI message.

## Midi Connections

Only one single Midi input is needed.
A Midi-thru can be nice for chaining multiple midi devices.

## Keep it simple

This implies that a single Midi message that is recieved must contain all information of the “FX patch”: Which FXs are bypassed or not.
The Midi channel will be “hardcoded” to channel 16.

## Midi Controller Change

Bypassing multiple FXs via a Midi Controller Change message:

- Midi Controller Change message contains 2x 7bits of info: CC Number and CC Value.
- CC Number controls which “bank” of 7 FXs is being controlled.
- CC Value controls the 7 FX of one bank = one FX “bank”.
- 2x 7 bits for controller number and value = 14 bits ⇒ 127 FX banks, each consisting of 7 FXs, can be controled.

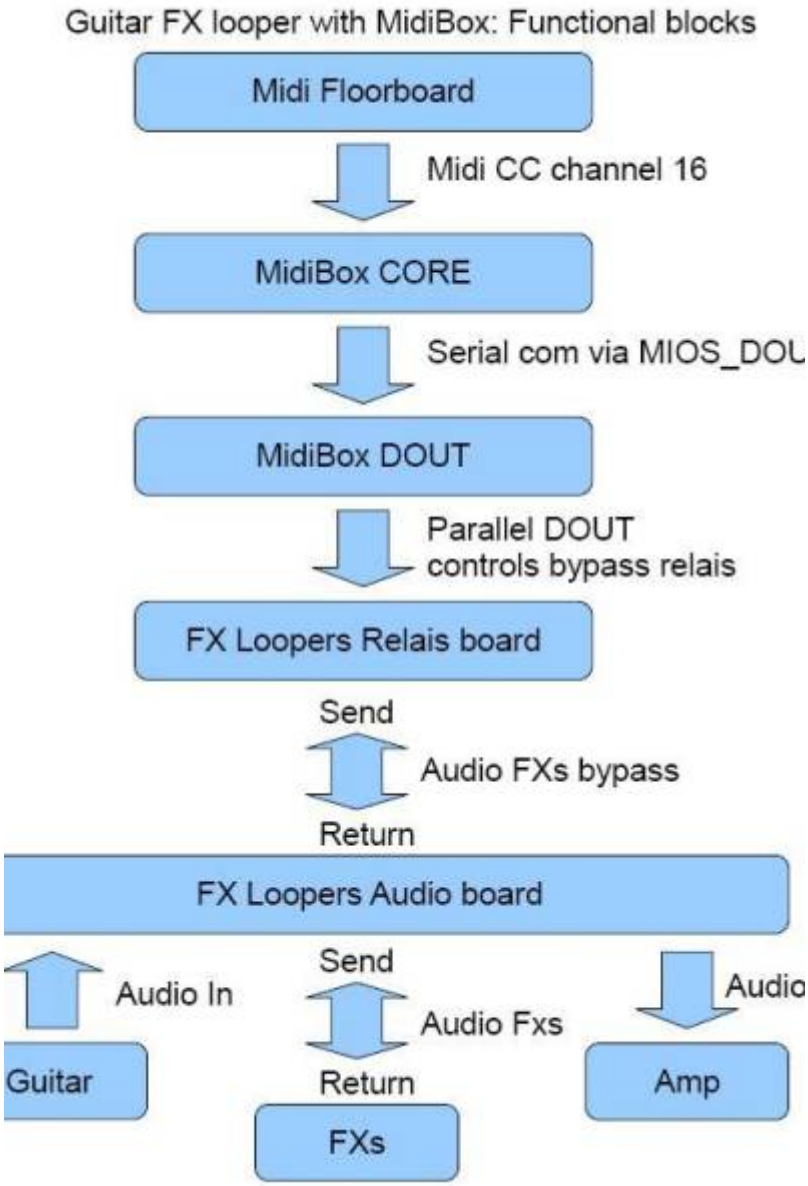
Specifying 7 FXs to be bypassed or not with one Midi Control Change message can be done “binary” using the CC value:

- Midi CC0 0 = 000 0000 = all FX OFF (= are bypassed)
- Midi CC0 1 = 000 0001 = first FX is ON, rest OFF
- Midi CC0 15 = 000 1111 = first 4 FXs are ON, last 3 FX are OFF
- Midi CC0 127 = 111 1111 = all FX are ON (= are not bypassed)

http://www.midi.org/about-midi/table1.shtml

```
1011nnnn      0ccccccc          Control Change.
              0vvvvvvv          This message is sent when a controller
                                value changes.  Controllers include devices
                                such as pedals and levers.
                                Controller numbers 120-127 are reserved
                                as "Channel Mode Messages" (below).
                                (ccccccc) is the controller number (0-119).
                                (vvvvvvv) is the controller value (0-127).
```

# Architecture



# Hardware

## External Controller Midi Floorboard

Use the Behringer FCB1010 floorboard that sends one Midi message when pressing down a knob-controller.




## MidiBox

Use the MidiBox as controller that recieves the Midi messages and controls the true bypass switches: www.ucapps.de

MidiBox kits: www.mikes-elektronikseite.de

I need a:

- CORE Module, that I can program to receive the midi from the footcontroller and translate midi program changes into on-off digital signal outputs.

http://www.ucapps.de/index.html?page=mios_c_set_dout.html

- DOUT Module, the digital outputs that will be set by the CORE module and will control the FX loopers.

NOTE: When using only 7 FX loopers, DOUT can be omitted. The FX looper board then needs to be connected directly to a CORE header. The application then needs to drive that header directly. This is not yet supported but should be relatively simple.
http://www.ucapps.de/index.html?page=mios_c_set_dout.html

- PIC18F4620 (with the bootloader)

www.mikes-elektronikseite.de

- OPTIONAL: LC Display 16×2 with backlight (for developing/debugging/testing purposes only, final version does not need a UserInterface)

http://www.ucapps.de/index.html?page=mios_c_set_dout.html

# User Interface: Keep it simple

Try to avoid or minimise a user interface so I would not need a DIN module nor a frontpanel with knobs and stuff.

### Indicator LEDs

It would be nice to add indicator LEDs for the FX loopers so you can see which FXs are bypassed (=LED OFF) and which ones are enabled (=LED ON). The indicator LEDs are also driven by the Relay driver chip.
LED ON ⇒ FX is not bypassed so it is in the FX chain.
LED OFF ⇒ FX is bypassed so it is not in the FX chain.

# Using DPDT Relays

Use a DPDT relay for bypass switching 1 FX:

DPDT relay switch board, controlled by the MidiBox DOUT module.

Each of these DPDT relays will bypass or enable one FX.

Drive the relay with a ULN2803, 8 relay driver, chip.

Perhaps add circuitry to remove clicks / pops induced by the relay into the audio path.

### Were to Buy the Relays

Dick Best, Mini relay, 5Vdc 2xwissel (DPDT)

€1.00

MINU64

http://www.dickbest.nl/webshop/index.php?act=viewProd&productId=986

Coil 195 ohm ⇒ 5V / 200 ohm = 25mA

Use a 220 ohm resistor for 12V supply voltage.

### How to Connect the Relays to a DOUT Module

http://www.ucapps.de/midio128/relay_example.pdf

G:\Electronica\Guitar Stompboxes\Bypass Loopers - Effect Bypassing - Effect Switching\Midi controlled FX looper or patchbay switch matrix\MidiBox - CORE and DOUT modules - ucapps.de

relay_example - MidiBox midio128.pdf

## ULN2803 Relay Driver Chip

http://www.datasheetcatalog.org/datasheets/90/366828_DS.pdf

G:\Electronica\DataSheets & Component info\Switching\Relais\Relais Drivers\ ULN2803 - 8 Relais Drives - Darlington Transistor Array.pdf

http://www.ucapps.de/midio128/uln2803.pdf

http://www.dickbest.nl/webshop/index.php?act=viewProd&productId=997

ULN953

€0.42

Used as output of the MidiBox DOUT module:

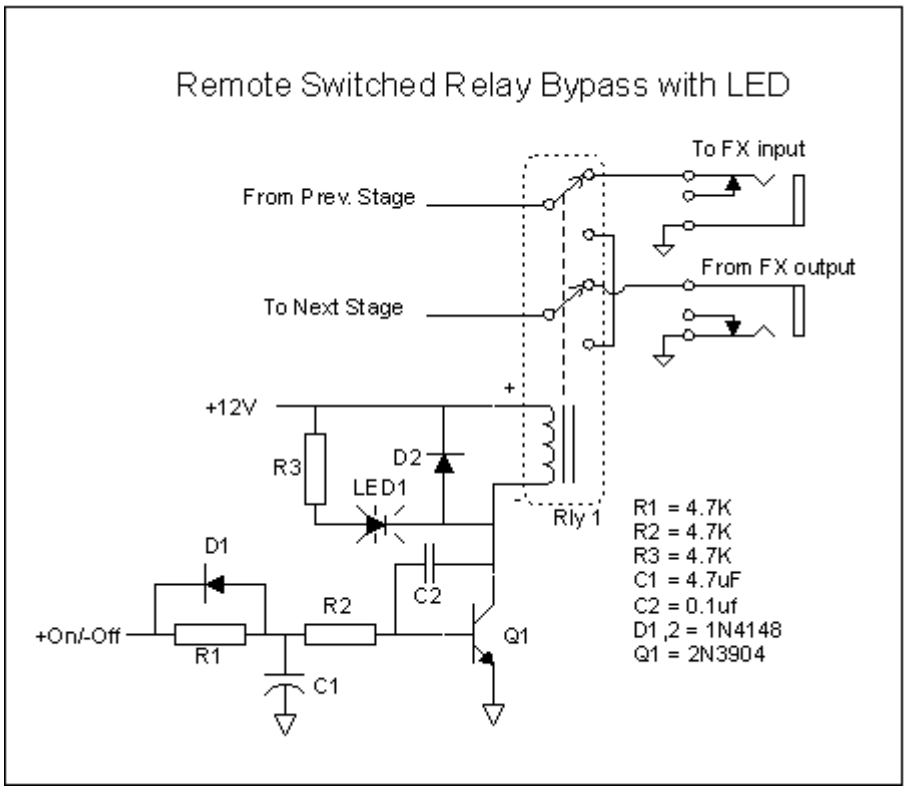
http://www.datasheetcatalog.org/datasheet/philips/74HC_HCT595_CNV_3.pdf

G:\Electronica\DataSheets & Component info\TTL and Logic\ 74HC595 - 8-Bit Serial-Input Serial or Parallel-Output Shift Register with Latched 3-State Outputs.pdf

## Circuitry to remove clicks / pops

From Geofex by R.G. Keen:

http://www.geofex.com/Article_Folders/rmtswtch/rmtsw.htm




“The neat way to defeat the clicks is to slow down the change in voltage across the inductor. The circuit at left does that. An on/off signal is applied to a resistor/diode/capacitor network on the base of a common 2N3904 transistor. The capacitor to ground slows the on/off transition down on the rising (turn on) edge of the input signal, and the diode lets it turn off fairly quickly.

The capacitor between the collector and base of the transistor causes the transistor to turn off slowly. Taken together, the speed with which the voltage across the coil inductance can change is slowed down. Tiny parasitic capacitors can't couple slow signals very well, and any transients are vastly reduced.”

## Connecting the Relays to the Output Jacks

### FX bypass

### A/B switching

### Sending a signal to multiple outputs

### Use a loop as a switch

- Latched
- Momentary

Normally Open and Normally Closed:

## Enclosure

### Front

Has the:

- State indicator LEDs for the FX loopers.
- On-off switch.

### Back

Has the:

- Midi in connector.
- Audio in/out/send/return jacks.
- Power in connector.

# Software

A list of available MIOS functions can be found here: http://www.ucapps.de/cmios_fun.html

## MPROC_NotifyReceivedEvent

Receiving and handling a Midi message:

http://www.ucapps.de/cmios_fun.html#USER_MPROC_NotifyReceivedEvent

USER_MPROC_NotifyReceivedEvent

C_DECLARATION void MPROC_NotifyReceivedEvent(unsigned char evnt0, unsigned char evnt1, unsigned char evnt2)

DESCRIPTION This function is called by MIOS when a complete MIDI event has been received

C_IN first MIDI event byte in <evnt0> second MIDI event byte in <evnt1> third MIDI event byte in <evnt2>

C_OUT -

ISR no

## MIOS_DOUT_SRSet

Controlling the DOUT module:

http://www.ucapps.de/cmios_fun.html#MIOS_DOUT_SRSet

MIOS_DOUT_SRSet

C_DECLARATION void MIOS_DOUT_SRSet(unsigned char sr, unsigned char sr_value)

DESCRIPTION sets value of DOUT shift register

C_IN number of shift register in <sr>

value in <sr_value>

C_OUT -

ISR no

## MIOS_SRIO_NumberSet

Initialising usage of the DOUT module:

http://www.ucapps.de/cmios_fun.html#MIOS_SRIO_NumberSet

MIOS_SRIO_NumberSet

C_DECLARATION void MIOS_SRIO_NumberSet(unsigned char number_sr)

DESCRIPTION sets number of available SR registers If number > 16, value will be forced to 16

C_IN number of SRs in <number_sr>

C_OUT -

## Controlling 7 Relays via MIDI CC Messages

Reworked from: Controlling 128 LEDs via MIDI http://www.ucapps.de/mios_c_set_dout.html

Enable/Disable 7 relays, which are connected to one DOUTX4 module, with Controller Change messages over MIDI Channel #16

Copy the SDCC skeleton into a new directory, open the main.c file and enhance the hooks like described below. Thereafter type “make” in the command shell, and upload the new project.hex file to the core.

Within the Init() function, you have to specify, how many shift registers are connected to the core:

```
/////////////////////////////////////////////////////////////////////////////
/////
```

```
// function: Init
// This function is called by MIOS after startup to initialize the
application.
/////////////////////////////////////////////////////////////////////////
/////
void Init( void ) __wparam
{
  // set shift register update frequency
  MIOS_SRIO_UpdateFrqSet( 1 ); // ms

  // Up to 4 DOUTX4 modules (=16 shift registers = 128 digital outputs) can
be connected
  // set the maximum number here - it doesn't really hurt!
  MIOS_SRIO_NumberSet( 16 );
}

/////////////////////////////////////////////////////////////////////////
/////
// function: MPROC_NotifyReceivedEvnt
//
// This function is called by MIOS when a complete MIDI event has been
received.
//
// Parameters:
//  unsigned char evnt0 [in] First MIDI event byte.
//    => Midi CC messages on channel 16 are supported. evnt0 = 0xBF.
//  unsigned char evnt1 [in] Second MIDI event byte in.
//    => Midi CC number: The "bank" of 7 FX loopers that is adressed. evnt1
= [0, 15].
//  unsigned char evnt2 [in] Third MIDI event byte in.
//    => Midi CC value: The bypass state of the 7 FX loopers
(enabled/disabled).
//       Each bit controls the state of one FX looper. Bits [0, 6] are used
for the 7 FX loopers.
/////////////////////////////////////////////////////////////////////////
/////
void MPROC_NotifyReceivedEvnt( unsigned char evnt0, unsigned char evnt1,
unsigned char evnt2 ) __wparam
{
  // evnt0 => Midi CC messages on channel 16 are supported. evnt0 = 0xBF.
  // Midi CC message (1011) on channel 16 (1111) => evnt0 = 1011 1111 = 0xBF
  if( evnt0 != 0xBF )
  {
    return;
  }

  // evnt1 = [0, 15].
  if( evnt1 > 15 )
  {
    return;
  }
```

```
  // evnt1 => Midi CC number: The "bank" of 7 FX loopers that is adressed.
  // evnt2 => Midi CC value: The bypass state of the 7 FX loopers
(enabled/disabled).
  // Each bit controls the state of one FX looper. Bits [0, 6] are used for
the 7 FX loopers.
  MIOS_DOUT_SRSet( evnt1, envt2 );
}
```

# Links to other related forum topics

From Midibox forum: DIY audio patchbay with digital routing....How hard? http://www.midibox.org/forum/index.php/topic,11262.0.html (Too complicated, too many features)

Program change to guitar fx gear controll http://www.midibox.org/forum/index.php/topic,7729.0.html

MT8816 based effects router http://www.midibox.org/forum/index.php/topic,10247.0.html (interesting)

looking to make a effect box http://www.midibox.org/forum/index.php/topic,10070.0.html

Midibox amp switcher http://www.midibox.org/forum/index.php/topic,3801.0.html Has this link but its dead: http://216.250.178.122/will/midibox/signalrouter/index.html

programable midi to relay unit http://www.midibox.org/forum/index.php/topic,8901.0.html

Midibox as Switcher advice sought (newbie alert 
http://www.midibox.org/forum/index.php?topic=4836.0

Building Pedal Box / Pedal Board http://www.midibox.org/dokuwiki/pedal_box

# Links to other related projects

There are of course other projects available on the web that try to accomplish the same goal with some variations. So for anybody that is interested I have also checked the following:

The Looper Mitch - Midi Switch http://www.jimkim.de/html/index.htm

MIDI Change-Over Relay Units http://tomscarff.tripod.com/relay_changeover/relay_changeover.htm

FX Switcher Project http://www.diystompboxes.com/smfforum/index.php?topic=56264.0

The Crossbar Project, a matrix effects switcher
http://www.diystompboxes.com/smfforum/index.php?topic=43104.0

Geofex FX switching:

http://www.geofex.com/Article_Folders/juggler/juggler.htm

http://www.geofex.com/Article_Folders/fxswitchr/fxswitchr.htm

http://www.geofex.com/Article_Folders/rmtswtch/rmtsw.htm

http://www.geofex.com/Article_Folders/ASMOP/ASMOP.htm

http://www.geofex.com/Article_Folders/ASMOP/asmop1d.pdf

http://www.geofex.com/Article_Folders/ASMOP/asmop1e.gif

From:
https://wiki.midibox.org/ - **MIDIbox**

Permanent link:
**https://wiki.midibox.org/doku.php?id=flo&rev=1248610712**

Last update: **2009/07/26 12:18**