Holy City Audio Forum

Former Home of SpinCAD Designer
Skip to content

Search... Search
Advanced search
[Moderator Control Panel]

Chorus/Flanger block

Post a reply

Search this topic... Search

13 posts • Page **1** of **1**

- Edit post (./posting.php?mode=edit&f=42&p=2139)
- Delete post (./posting.php?mode=delete&f=42&p=2139)
- Report this post (./report.php?f=42&p=2139)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=2139&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2139)

Chorus/Flanger block (#p2139)

by **Digital Larry** » Tue Feb 17, 2015 8:27 pm

Anyone who's been around here for awhile probably realizes that I haven't made any blocks that use the chorus-specific between-sample interpolation capabilities of the FV-1. And I think it's about time we changed that. Let's just consider what a given block might contain.

#1 You allocate a buffer.
#2 Since using the CHO RDA instructions in this manner for chorusing requires the use of the direct use of the SIN/COS (for sine wave modulation) or Ramp (for servoing to arbitrary input waveforms), I have to have user adjustable parameters to control the sweep width within the buffer.

Sine modulated delays
Suppose that I allocate a 4096 word buffer to do some sine modulated delay. I suppose in general it's going to be smaller than that but let's just start with that. If I put the modulation in the center of the buffer, then I could get +2048/-2047 or symmetrically speaking +/- 2047. Or I could offset the modulated delay within that buffer in addition to making the width smaller (since sweep needs to be symmetrical). So If I centered the delay at 1024 or (4096 -1024) = 3072, then I could have a sweep of +/- 1024.

So then let's think of a modulated delay as having a buffer length, and some number of modulated taps so to speak. Maybe 4 maximum with checkboxes to enable 2-4. We'll start with just one though.

Each tap has the following attributes:
Offset within the buffer.
Sin LFO 1 or 2
Sin/Cos
LFO speed
LFO width (has to be less than min(0ffset, max - offset)) - I could have the parameter be a %, so it would just be whatever percentage of the maximum possible. This way it wouldn't need to change as you changed the max or the offset.

As far as I know the same LFO could be used for multiple taps so long as they all have the same width. Cos and Sin from the same LFO also have the same width.

Controls come into the block to scale down LFO width and speed from their maximum settings.

Here's the example from AN-0001:

```
; Program AN0001-1.spn
;
; Memory declarations
delayl mem 8193 ; Left delay
delayr mem 8193 ; Right delay
; Initialization, only run on first execution of code
; Skip to the START label if NOT the first time
skp run,START
; Load up a sin LFO, this is about 0.2Hz
; and +/-4096 samples for a total delay requirement of 8193
wlds SIN0,5,16384
; End of skip/initiaization
; Main program code
; First, read in the sample and write to the start of the delay
; Read in the current left sample, ADCL -> ACC
START: ldax ADCL
; Write it to left delay and clear ACC
wra delayl,0
; Read in the current sample, ADCR -> ACC
ldax ADCR
; Write it to right delay and clear ACC
```

```
wra delayr,0
; We use the middle of the sample memory block as the
; address since we are using a signed sine wave
; that ranges -1.0 to 1.0 (-4096 to +4095 in this case)
; The following instruction will read LFO 0, "REG" it in a temp register
; in the LFOblock (we don't want it to change while we are doing calculations)
; add the "integer" portion to the address of the middle of the delay
; and down counter and finally multiply the sample from the delay
; memory by (1-K). That's a lot for one instruction! The "rda"
; tells the ALU to add the result of the multiplication to the ACC,
; similar to a normal rda instruction. You can combine rda, wra, sof
; or rdal with a chorus instruction.
; (1-k)*sample[addr]
cho rda,SIN0,SIN|REG|COMPC,delayl^
; Now we have sample[N] * (1-interpolation coeff) in the accumulator,
; now need to generate sample[N+1] * (interpolation coefficient) and
; add it to the value in the accumulator. Note that the address used
; in the instruction is 1 greater than the preceding instruction.
;k*sample[addr+1] + ACC
cho rda,SIN0,SIN, delayl^+1
; Interpolated sample in ACC, write it to DACL and clear ACC
wrax DACL,0
; Repeat for right channel. Since we are using the same LFO and output
; we do not need to REG the LFO again.
cho rda,SIN0,SIN|COMPC,delayr^
cho rda,SIN0,SIN,delayr^+1
wrax DACR,0
; That's it!
```

Now I'm going to trim that down to just one channel. I'm also going to substitute some variables from the control panel.

```
;
; Memory declarations
delayl mem 8193 ; Left delay
skp run,START
wlds SIN0, speed,16384
START: ldax input
; Write it to left delay and clear ACC
wra delayl,0
cho rda,SIN0,SIN|REG|COMPC,delayl+offset
cho rda,SIN0,SIN, delayl+1+offset
; Interpolated sample in ACC, write it to DACL and clear ACC
wrax output,0
; That's it!
```

That's a little weird, LFO width = 16384 but buffers are only 8193? I'll need to go figure that one out.

Well, before I worry about that, I have to deal with the delay memory offset just like I had to with the spincad based delay blocks.

This uses this macro sequence:

```
@getDelayScaleControl tap1Ratio delayLength delayOffset
wrax ADDR_PTR, 0
rmpa 1.0
```

@getDelayScaleControl gets the absolute delay RAM address based on the offset into RAM of the base (since there are possibly other blocks using RAM), the delay length, and the % of the delay length from the base we wish to point at. Oh and it puts this value into ACC for use by RMPA.

Now I need a way to create a value that can be used in the CHO RDA instruction. It's so confusing to me now because obviously I never used the CHO RDA instruction, at least not for the purpose I'm currently trying to accomplish.

cho rda example:
cho rda,SIN0,SIN|REG|COMPC,delay1^

SpinCAD Builder parsing syntax:
ChorusReadDelay: 'cho rda' ',' arg1 = SPINREGISTER ',' arg2 = SPINCHOREGFLAGS ',' arg3 = SPINMEM;

```
SPINMEM:
buffer = SPINBUF (value = INT | offset=[Offset])?
;

SPINBUF: ID('#'|'^')?('+'|'-')?
;
```

Unfortunately I have to take a few minutes to remember what this means! All right. I think I got it. The following expressions are OK for the third argument to **cho rda**.

```
cho rda, rmp0, reg|compc, fdell
cho rda, rmp0, reg|compc, fdell+
cho rda, rmp0, reg|compc, fdell-
cho rda, rmp0, reg|compc, fdell+12
cho rda, rmp0, reg|compc, fdell#+12
cho rda, rmp0, reg|compc, fdell^+12
cho rda, rmp0, reg|compc, fdell^-12
cho rda, rmp0, reg|compc, 300
```

I need to be able to generate one of those forms somehow~!

In spincad.xtext:

```
GetDelayScaleControl: '@getDelayScaleControl' ratio = ID length = ID offset = ID (control = ID)?;
```

In spinCADCodeGenerator.xtend:

```
// scale = (95/100) * tap1 * (bufferLength/32768.0);
// offset = (bufferBase + tap1 * (5/100.0) * bufferLength)/32768.0;

def genGetDelayScaleControl(GetDelayScaleControl g) {
'''
«IF g.control != null»
sfxb.mulx(«g.control»);
«ENDIF»
sfxb.scaleOffset((0.95 * «g.ratio» * «g.length»)/32768.0, («g.offset» + (0.05 * «g.ratio» * «g.length»))/32768.0);
'''
}
```

How it's used in a variety of multi-tap delays:

```
@isPinConnected Delay_Time_1
equ max reg1

clr
or $7FFF00
mulx cIn1
// max now holds the value of the POT scaled by max ADDR_PTR
wrax max,1.0

@getDelayScaleControl tap1Ratio delayLength delayOffset
wrax ADDR_PTR, 0
rmpa tap1Gain
wrax mix, 0.0
```

This is all customized for the RMPA instruction, so the value that gets scaled when we first come in is the maximum value (7FFF00) multiplied by a control pot if it exists.

@getDelayScaleControl generates a SOF instruction to place a value into ACC that can be used by RMPA to point to a specific place in delay RAM.

What we need for the chorusing instruction is something like that, but instead of putting a value into ACC for RMPA, it should create a Java variable, which renders ultimately as a constant between 0 and 32767 for the **cho rda** instruction.

Top
- Edit post (./posting.php?mode=edit&f=42&p=2140)
- Delete post (./posting.php?mode=delete&f=42&p=2140)
- Report this post (./report.php?f=42&p=2140)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=2140&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2140)

Re: Chorus/Flanger block (#p2140)

by **Digital Larry** » Thu Feb 19, 2015 3:40 pm

K I think the best way is to have a compact representation in the spincad file and break out to move ElmGen Java during during code generation.

For example:
In spincad.xtext:
GetChorusScaleControl: '@getChorusScaleControl' ratio = ID length = ID offset = ID;

In spinCADCodeGenerator.xtend:
tap = ratio
length = bufferLength
offset = bufferBase

For the delay line function, the actual tap is at 5% when the slider is zero, up to 100% when the slider is all the way up. Since the chorus tap point has to have + and - swings, it can't be right at the end. I think for now I'll limit it to between 25% and 75% of the delay's length. If the mod delay center is at 25% or 75% then the maximum width is 0.5. If the center is at 50% then the max width is 1.0 (+/- 0.5). Still not exactly sure how to represent it.

If tap = 0.0 to 1.00 (scaled slider value),

offset = bufferBase + tap * bufferLength;

We want to limit tap position to being between 0.25 and 0.75 of bufferLength.
We also want to specify LFO 0 or 1 and Cos or Sin waveform.

```
int LFO;
int WvForm;

def genGetChorusScaleControl(GetChorusScaleControl g) {
'''
int offset = (0.5 * «g.tap» * «g.bufferLength») + «g.offset» + (0.25 * «g.bufferLength»); // offset needs to be calculated in Xtend, not generated Java
sfxb.FXchorusReadDelay(SIN1, 0x06, <<offset>>);
sfxb.FXchorusReadDelay(SIN1, 0x00, <<offset+1>>);

}
```

Top
- Edit post (./posting.php?mode=edit&f=42&p=2141)
- Delete post (./posting.php?mode=delete&f=42&p=2141)
- Report this post (./report.php?f=42&p=2141)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=2141&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2141)

Re: Chorus/Flanger block (#p2141)

by **Digital Larry** » Thu Feb 19, 2015 7:03 pm

I hope nobody is reading this, because it is really only for me to get my thoughts out so that they fall into some sort of order.

And here is the order they seem to be falling.

#1 A chorus block has a delay RAM allocation. (delayLength) This slider could go from 128 to 4096 for example. Subject to change once I get it going.
#2 A chorus block has up to 4 taps, each of which is positioned within the delay RAM. The position is taken from a delay position slider that goes from 25% to 75% of the delay length.
#3 Each chorus tap has a % offset within the buffer, LFO (0 or 1) and (Cos or Sin) LFO phase select.
#4 I should make a @readChorusTap macro, which takes as parameters LFO (0,1) SinCos (0,1) and offset (0.25 - 0.75) into a buffer (bufferLength).

The @readChorusTap macro will create the two adjacent **cho rda** instructions and will return the interpolated value in the ACC.

It will be the obligation (somehow) of the code to manage the LFO width to make sure it does not go outside the boundaries of the buffer.

An example @readChorusTap % value of some JSlider in the chorus block's Control Panel. I guess I can map from 0 - 1.0 to 0.25 - 0.75 in the block Java code.

Top
- Edit post (./posting.php?mode=edit&f=42&p=2142)
- Delete post (./posting.php?mode=delete&f=42&p=2142)
- Report this post (./report.php?f=42&p=2142)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=2142&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2142)

Re: Chorus/Flanger block (#p2142)

by **Digital Larry** » Fri Feb 20, 2015 7:29 am

Getting a little somewhere. No pot adjustments yet, and had to override some of the block generated values. Just gotta keep chipping away at it.

This is just a fixed sweep chorus, finally using the CHO RDA instructions.

```
;  Program: Render Block exported from SpinCAD Designer
;------ Input
;------ Chorus
SKP RUN ,1
WLDS 0,10,127
LDAX 20
WRA 0,0.0
CHO RDA,0,REG | COMPC,512
CHO RDA,0,0,513
WRAX REG0,0.0000000000
;------ Mixer 2-1
RDAX REG0,0.5000000000
WRAX REG1,0.0000000000
RDAX ADCL,0.5000000000
RDAX REG1,1.0000000000
WRAX REG1,0.0000000000
;------ Output
RDAX REG1,1.0000000000
WRAX DACL,0.0000000000
RDAX REG1,1.0000000000
WRAX DACR,0.0000000000
```

Top
- Edit post (./posting.php?mode=edit&f=42&p=2143)
- Delete post (./posting.php?mode=delete&f=42&p=2143)
- Report this post (./report.php?f=42&p=2143)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=2143&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2143)

Re: Chorus/Flanger block (#p2143)

by **Digital Larry** » Sat Feb 21, 2015 5:53 pm

OK I've got adjustments for total buffer delay time and center position of the chorus tap within that delay from 25% to 75%. This means width can be 50% of the buffer length. Now I've got to create a function that will translate a slider position value to an LFO coefficient and or perhaps vice versa.

Here's the code extracted from the Java implementation of SinCosLFOCADBlock:

```
   private void updateLfoRateLabel() {
      lfoRateLabel.setText(String.format("%2.1f Hz", (ElmProgram.getSamplerate() * pC.getLFORate()) / (2 *
Math.PI * Math.pow(2.0 ,17))));
   }
```

Top
- Edit post (./posting.php?mode=edit&f=42&p=2144)
- Delete post (./posting.php?mode=delete&f=42&p=2144)
- Report this post (./report.php?f=42&p=2144)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=2144&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2144)

Re: Chorus/Flanger block (#p2144)

by **Digital Larry** » Sat Feb 21, 2015 10:24 pm

Here's the next development in this super exciting saga.

I have to figure out how to present the block so that it's fairly easy to use. Sometimes this means sacrificing some flexibility.

Right now I have:
- Delay length
- Position of tap within delay (25% to 75%)
- LFO Speed (it's the max speed if you connect a control input) - the LFO speed goes up to 20.3 Hz which is pretty insane. I like my LFOs down about 2 Hz max!
- LFO Width (it's the max width if you connect a control input) - the full sweep width possible sounds pretty seasick. I may limit it to something more reasonable (some seasisckness possible).

Need to have an LFO selector.

I was also considering having a second tap which would default to the COS phase LFO.

Also, perhaps a second output which if connected, would be the second tap using COS LFO. If not connected, either you'd only get the first tap, or you could have them mix together to one signal.

Also considering adding an output tap for the delay center, so that if your tap is also centered, or sweeps over the center, you could get those through-zero sounds.

Had me a look at the Spin Knowledge base: http://www.spinsemi.com/knowledge_base/ ... tml#Chorus (http://www.spinsemi.com/knowledge_base/coding_examples.html#Chorus)

I need to at least try to accomplish this:

```
For a wild set of chorus outputs, you can set up a single sine LFO and get 4 outputs, that we'll call c1
through c4:

cho   rda,sin0,sin|reg|compc,cdel+800
cho   rda,sin0,sin,cdel+801
wrax   c1,0
cho   rda,sin0,sin|reg|compa,cdel+400
cho   rda,sin0,sin|compa,cdel+401
wrax   c2,0
cho   rda,sin0,cos|reg|compc,cdel+1100
cho   rda,sin0,cos,cdel+1101
wrax   c3,0
cho   rda,sin0,cos|reg|compa,cdel+1400
cho   rda,sin0,cos|compc,cdel+1401
wrax   c4,0
```

Top
- Edit post (./posting.php?mode=edit&f=42&p=2146)
- Delete post (./posting.php?mode=delete&f=42&p=2146)
- Report this post (./report.php?f=42&p=2146)
- Warn user (./mcp.php?i=warn&mode=warn_post&f=42&p=2146&sid=90615c931909e0633cbcd2a3573379b1)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=2146&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2146)

Re: Chorus/Flanger block (#p2146)

by **Jacko** » Mon Feb 23, 2015 3:30 pm

> ***Digital Larry wrote:***
> I hope nobody is reading this, because it is really only for me to get my thoughts out so that they fall into some sort of order.

No such luck... but I like where you are headed with it!

Best regards, Jacko

Top
- Edit post (./posting.php?mode=edit&f=42&p=2147)
- Delete post (./posting.php?mode=delete&f=42&p=2147)
- Report this post (./report.php?f=42&p=2147)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=2147&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2147)

Re: Chorus/Flanger block (#p2147)

by **Digital Larry** » Mon Feb 23, 2015 9:40 pm

Hey Jacko,

Glad you are enjoying it! That makes one of us. In the case of software development, the destination is way better than the journey!

Now I need to add an LFO selector. Here's how I select between 2 and 4 poles on the low pass filter.

```
   private JComboBox<Object> nPoles;

   private String listOptions[] = {
         " 2 poles ",
         " 4 poles "
   };

   public LPF4PControlPanel(LPF4PCADBlock lpf1pcadBlock) {
      this.LPF = lpf1pcadBlock;
      nPoles = new JComboBox<Object>(listOptions);
      nPoles.addActionListener(this);
```

```
      createAndShowUI();
   }

   private void createAndShowUI() {
      SwingUtilities.invokeLater(new Runnable() {
         public void run() {
            if(LPF.getIs4Pole()) {
               setTitle("Low pass 4 pole");
               nPoles.setSelectedIndex(1);
            } else {
               setTitle("Low pass 2 pole");
               nPoles.setSelectedIndex(0);
            }
            setLayout(new BoxLayout(getContentPane(), BoxLayout.Y_AXIS));

            getContentPane().add(Box.createRigidArea(new Dimension(250,7)));
            getContentPane().add(nPoles);
            getContentPane().add(Box.createRigidArea(new Dimension(250,4)));

            setAlwaysOnTop(true);
            setVisible(true);
            setLocation(new Point(LPF.getX() + 200, LPF.getY() + 150));
            pack();
            setResizable(true);
         }
      });
   }

   @Override
   public void actionPerformed(ActionEvent arg0) {
      if (arg0.getSource() == nPoles) {
           JComboBox cb = (JComboBox)arg0.getSource();
           String range = (String)cb.getSelectedItem();
           if (range == listOptions[0]) {
              LPF.setIs4Pole(false);
           } else if (range == listOptions[1]) {
              LPF.setIs4Pole(true);
           }
      }
   }
```

And over in the block:

```
   boolean is4Pole = false;

   public boolean getIs4Pole() {
      return is4Pole;
   }

   public void setIs4Pole(boolean r) {
      is4Pole = r;
   }
```

So, I'm thinking I need to create a SpinCAD parser rule in Xtext something like:

@comboBox nPoles "2 poles" "4 poles" "6 poles"

and then some way to figure out which one is selected, like

@ifComboBoxSelectionEquals nPoles "2 poles"

Hmmmm. Hmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm. <smoke emitting from ears>

Top

- Edit post (./posting.php?mode=edit&f=42&p=2152)
- Delete post (./posting.php?mode=delete&f=42&p=2152)
- Report this post (./report.php?f=42&p=2152)

- Information (./mcp.php?i=main&mode=post_details&f=42&p=2152&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2152)

Re: Chorus/Flanger block (#p2152)

by **Digital Larry** » Sat Feb 28, 2015 9:14 am

Just in case anyone was wondering, I'm making some progress but boy this is turning out to be a lot of work! However I am confident that I will get it working eventually. Maybe another week or two? At that point I will be able to add LFO selectors to all SpinCAD Builder based blocks which use either a Ramp or Sin LFO.

Top

- Edit post (./posting.php?mode=edit&f=42&p=2153)
- Delete post (./posting.php?mode=delete&f=42&p=2153)
- Report this post (./report.php?f=42&p=2153)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=2153&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2153)

Re: Chorus/Flanger block (#p2153)

by **Digital Larry** » Sun Mar 01, 2015 5:09 pm

Getting closer. Here's a dual chorus.

Left in/left out mono.

Pot 0 = LFO speed - there are two chorus LFOs, and they speed up and slow down in opposite directions over different ranges.
Pot 1 = LFO width

```
;  Program: Render Block exported from SpinCAD Designer
;------ Input
;------ Pot 1
;------ Pot 0
;------ Scale/Offset
RDAX POT0,1.0000000000
SOF -0.72000000,0.7700000000
WRAX REG0,0.0000000000
;------ Chorus
SKP RUN ,1
WLDS 0,50,64
RDAX POT1,0.0027319336
WRAX SIN0_RANGE,0.0000000000
RDAX REG0,0.3381017613
WRAX SIN0_RATE,0.0000000000
LDAX 20
WRA 0,0.0
CHO RDA,0,REG | COMPC,289
CHO RDA,0,0,290
WRAX REG1,0.0000000000
;------ Scale/Offset
RDAX POT0,1.0000000000
SOF 0.9600000000,0.0400000000
WRAX REG2,0.0000000000
;------ Chorus
SKP RUN ,1
WLDS 1,50,64
RDAX POT1,0.0020343018
WRAX SIN1_RANGE,0.0000000000
RDAX REG2,0.5095303327
WRAX SIN1_RATE,0.0000000000
LDAX 20
WRA 620,0.0
CHO RDA,1,REG | COMPC,1153
CHO RDA,1,0,1154
WRAX REG3,0.0000000000
;------ Mixer 3-1
RDAX REG1,0.530
WRAX REG4,0.0000000000
RDAX ADCL,0.53333333333
RDAX REG4,1.0000000000
WRAX REG4,0.0000000000
RDAX REG3,0.53333333333
RDAX REG4,1.0000000000
```

```
WRAX REG4,0.0000000000
;------ Output
RDAX REG4,1.0000000000
WRAX DACL,0.0000000000
RDAX REG4,1.0000000000
WRAX DACR,0.0000000000
```

Top
- Edit post (./posting.php?mode=edit&f=42&p=2156)
- Delete post (./posting.php?mode=delete&f=42&p=2156)
- Report this post (./report.php?f=42&p=2156)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=2156&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2156)

Re: Chorus/Flanger block (#p2156)

by **Digital Larry** » Mon Mar 02, 2015 7:38 am

It's all working pretty well EXCEPT that I need the drop down to set itself to the chosen LFO when the control panel opens. Right now it's always going back to LFO 0.

The output range selector in the Sin/Cos LFO also uses a JComboBox ActionListener, like so:

```
   @Override
   public void actionPerformed(ActionEvent arg0) {
      if (arg0.getSource() == outputRange) {
           JComboBox cb = (JComboBox)arg0.getSource();
           String range = (String)cb.getSelectedItem();
           if (range == listOptions[0]) {
              pC.setRange(0);
           } else if (range == listOptions[1]) {
              pC.setRange(1);
           }
      }
   }
```

whereas the one I have going on for the Chorus block looks like this:

```
        @Override
        public void actionPerformed(ActionEvent arg0) {
        if(arg0.getSource() == lfoSelComboBox) {
           gCB.setlfoSel((lfoSelComboBox.getSelectedIndex()));
        }
```

One big difference is that I'm not populating the ComboBox from a String list... suppose I could do that.

The other thing I have a hard time grasping sometimes is all the possible things that trigger the ActionEvent for the ComboBox.

Top
- Edit post (./posting.php?mode=edit&f=42&p=2158)
- Delete post (./posting.php?mode=delete&f=42&p=2158)
- Report this post (./report.php?f=42&p=2158)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=2158&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2158)

Re: Chorus/Flanger block (#p2158)

by **Digital Larry** » Tue Mar 03, 2015 6:27 am

Oh man I'm getting close folks! The problem with the LFO Combo Box always showing "LFO 1" was a bug where I had set the ActionListener prior to adding the items to the combo box, so the ActionListener was getting triggered before the thing was even set up. I had copied the ComboBox code from some CheckBox code I'd written, however I'd never actually debugged that either! Well, let that be a lesson to you.

Next thing I need to do is to correctly calculate the Chorus width based on the delay buffer length and the % width setting in the control panel.

// From AN-001:
// The equation to calculate the coefficient for the amplitude for a given delay length for use in a WLDS instruction is:
// Ka = (N * 32767)/16385
Where:
N: Delay length in samples
Valid values are in the range of 0 to 32767 for Ka.
// So, assuming that we want N to be delayLength/2, then Ka should be:
// Ka = delayLength * (0.5 * 32767/16385)

// or
// Ka = delayLength * 0.9999084528532194 * width (0.05 to 1.0) derived from control panel width slider.

I want the modulated delay center point to be adjustable from 25% to 75% of the buffer length, so that would make the maximum width 50% (which is where the 0.5 comes from above). I could make another chorus block where the center point is just in the center of the buffer so you can make the width the full length of the buffer. I'm also putting together a quad chorus where you can set the center point for each tap between 25% and 75%, and each output tap would be 0, 90. 180, 270 degrees phase shifted on the LFO. It would still use just one LFO, so it's not quite at the same level of complexity as the "rich chorus" snippet at the Spin web site.

Since the LFO selector already works I put one for the Ramp LFO selection in the Sample/Hold block, it works!

Top
- Edit post (./posting.php?mode=edit&f=42&p=2160)
- Delete post (./posting.php?mode=delete&f=42&p=2160)
- Report this post (./report.php?f=42&p=2160)
- Information (./mcp.php?i=main&mode=post_details&f=42&p=2160&sid=90615c931909e0633cbcd2a3573379b1)
- Reply with quote (./posting.php?mode=quote&f=42&p=2160)

Re: Chorus/Flanger block (#p2160)

by **Digital Larry** » Wed Mar 04, 2015 4:43 pm

The "final" SpinCAD Builder code for the Chorus block in build 893.

I decided, with only one tap, it does not matter to have the center point be variable within the length of the buffer. It's no different than simply choosing a slightly different delay time. I do have another block up my sleeve which is the Quad Chorus, that will have adjustable-position taps like I'd been planning, and the LFO for each of the 4 taps will also be 90 degrees lagged in phase. This gives "some bending up, some bending down" from a single Sin/Cos LFO. Haven't tried it yet, looking forward to it.

```
@name Chorus
@audioInput input Input
@audioOutput output1 Output
@controlInput rateIn LFO_Rate
@controlInput widthIn LFO_Width

equ rateMax 511
equ number6554000 6554000.0
equ output1 reg0

// total allocated memory buffer for this delay
// create a Control Panel with a Slider Label
equ delayLength 512
@sliderLabel delayLength Chorus_Time 128 2048 128 1 0 lengthToTime

equ tap1Center 0.5
; @sliderLabel tap1Center Tap_1_Center 0.25 0.75 0.5 1000.0 2

;equ tap2Center 0.25
;@sliderLabel tap2Center Tap_2_Center 0.0 1.0 0.5 1000.0 2

equ rate 20
@sliderLabel rate LFO_Rate 0.0 511.0 20.0 100.0 2 SINLFOFREQ

// max width (in samples) of LFO = delayLength/2.
// this allows the center to be placed from 25% to 75%
// From AN-001:
// The equation to calculate the coefficient for the amplitude for a given delay length for use in a WLDS
instruction is:
// Ka = (N * 32767)/16385

equ width 30   // double value of width %
@sliderLabel width LFO_Width 5.0 100.0 30.0 100.0 1

// ComboBox needs to show LFO 0 or LFO 1
// in block code, need to select between SIN0 or SIN1 parameter
// setter/getter passes index of combobox item, for now
// let's see if we can make it work.

equ lfoSel 0
@comboBox lfoSel 'LFO 0' 'LFO 1'
```

```
@isPinConnected Input
;
; Memory declarations
//equ delayOffset -1
@getBaseAddress
mem delay1 delayLength ; delay should actually be delay * 2 as panel setting is sweep center

@isEqualTo lfoSel 0
skp run,START
wlds SIN0, 50, 64
START:
@else
skp run,START1
wlds SIN1, 50, 64
START1:
@endif

// want to try to get to this:
// From AN-001:
// The equation to calculate the coefficient for the amplitude for a given delay length for use in a WLDS
instruction is:
// Ka = (N * 32767)/16385
// Where:
// N: Delay length in samples, 16385 max( or length of delay buffer).
// back to SpinCAD ============================
// Then there is the width factor, 5% to 100%
// Valid values are in the range of 0 to 32767 for Ka.
// Ka = (scale * ( N * 32767))/16385
// Finally if needed it is all scaled by the control input
// double x1 = delayLength * 32767
// double x2 = x1 * scale
// double x3 = x2 / 16385
// I totally fudged this!  Whatever works.

@isPinConnected LFO_Width

@multiplyDouble x1 delayLength width
@divideDouble x3 x1 number6554000

rdax   widthIn, x3
@isEqualTo lfoSel 0
wrax SIN0_RANGE, 0
@else
wrax SIN1_RANGE, 0
@endif
@endif

// Here's the expression for scaling the LFO rate by the control input.
// we read the control and scale it by the slider setting/511 which is max.
// sfxb.readRegister(speedIn, lfoRate/511.0);

@isPinConnected LFO_Rate
@divideDouble temp1 rate rateMax
rdax   rateIn, temp1
@isEqualTo lfoSel 0
wrax SIN0_RATE, 0
@else
wrax SIN1_RATE, 0
@endif
@endif

ldax input
; Write it to delay and clear ACC
wra delay1,0

@isEqualTo lfoSel 0
@readChorusTap 0 0 tap1Center delayLength delayOffset
```

```
@else
@readChorusTap 1 0 tap1Center delayLength delayOffset
@endif

; Interpolated sample in ACC, write it to DACL and clear ACC
wrax output1,0
; That's it!
@setOutputPin Output output1
@endif
```

Top
Display posts from previous: All posts Sort by Post time Ascending Go
Post a reply
13 posts • Page **1** of **1**
Return to Developer's Corner
Jump to: Developer's Corner Go
Quick-mod tools: Lock topic Go

Who is online

Users browsing this forum: Digital Larry and 0 guests
Powered by phpBB® Forum Software © phpBB Group
Administration Control Panel