

# Chapter 20

## Programming Examples

The other chapters in this manual have described the K2600's features in detail. This tutorial chapter will take you step-by step through several programming operations.

Each of the following examples will begin from the same starting point: the default program with ID 199. This program is included specifically for the purpose of giving you a programming template. Most of its parameters have been set at values that don't affect the sound of the program.

You may want to adjust some of the parameters of Program 199, to create your own customized programming template. Even if you don't, it's a good idea to begin with Program 199 when you're building a new sound, so you'll know exactly what you have from the start.

Keep in mind that none of these examples provides you with a usable program. Instead, the examples are designed to give you tools and concepts which you can apply to your own sounds. Once you become familiar with the programming basics in this chapter, analyze a few of the factory presets by moving through the Program Editor, and observing how those presets were designed. This may help you learn more techniques for creating new sounds.

While in the Program Editor, there are several editing shortcuts you can use. To call up a control source, enter its number on the alphanumeric buttonpad, or hold **Enter** and strike a key on the keyboard (see Control Sources in Chapter 4 of the *Musician's Reference*). When a highlighted parameter has a Control Source as its value, press **Edit**, and you will go directly to that Control Source page. You can also use the **Previous Pg**, **Mark**, and **Jump** buttons (see page 5-7).

### Example 1

#### Trumpet with Delayed Vibrato and Velocity-triggered Falls

Vibrato is a regular oscillation in pitch that adds dimension to any sound. Brass players will often "fall off" from a note, punching it then letting the pitch roll down smoothly or in small fast steps.

To create these effects, we'll use an LFO to control the pitch, (this is the typical way to create vibrato), and delay it with an ASR. This way you'll hear the vibrato only on notes that you hold for a second or so. The stab will be done with a second ASR controlling pitch and amplitude. The stab's ASR will be triggered by a velocity trigger (VTRIG), so only those notes you play at fortissimo will stab.

Start by selecting Program 199 and pressing **Edit**. The ALG page will appear. The first task is to change the keymap. Press the **KEYMAP** soft button to select the KEYMAP page.

```

EditProg:KEYMAP <>Layer:1/1
KeyMap:17 Trumpet Stereo:Off
XPose :0ST TimbreShift :0ST
KeyTrk:100ct/key AltSwitch :OFF
VelTrk:0ct PlaybackMode:Normal
<more> ALG LAYER KEYMAP PITCH >more>
    
```

The KeyMap parameter is already selected, and as you can see, the Default program uses the Grand Piano keymap. Use any data entry method to change the keymap to **Trumpet**, which has ID 17. The KEYMAP page should look like the diagram above when you're done. Remember that you can play your MIDI controller's keyboard at any time while editing, so you can listen to each change as you make it.

Next set up the vibrato. Start by selecting the PITCH page (press the **PITCH** soft button). Use the cursor buttons to move the cursor to the Src2 parameter. Use any data entry method to select **LFO1** as its value by pressing **1, 1, 4, Enter** on the alphanumeric buttonpad (or hold **Enter** and strike B 5 on the keyboard). This assigns LFO1 to control the pitch of the trumpet sample.

The next step is to set the depth of the vibrato. Select the MaxDpt parameter and assign a value of **10 cents (1, 0, Enter)**. Since the default program is preset to have your controller's Mod Wheel control the depth of Src2, you can hear the vibrato by pushing the Mod Wheel fully up (LFO1 has nonzero default values in the default program, otherwise, you wouldn't hear the vibrato). If you're not sure you hear the vibrato, try setting the MaxDpt parameter to a larger value.

Next, select the DptCtl parameter and assign a value of **ASR2** by pressing **1, 1, 1**, then **Enter** (or hold **Enter** and strike G# 5 on the keyboard). This will cause ASR2 to control the depth of the vibrato. At this point, the default values for ASR2 will cause the vibrato to fade in and out.

There are two more steps to programming the delayed vibrato: adjusting the rate of LFO1 and setting up ASR2 to control the vibrato's delay. First, highlight Src2 and press **Edit**; this brings up the LFO page. The default value for LFO1's minimum rate (the MnRate parameter) is **2 seconds**. Select this parameter with the cursor buttons, and set its value to **.16 seconds (1, 6, Enter)**. Select the MxRate parameter, and set its value to **4.40 Hz (4, 4, 0, Enter)**. Select the RateCt parameter, and assign a value of **ASR2 (1, 1, 1, Enter)**. The vibrato will still fade in and out because of the default settings of ASR2.

The LFO page should now look like this:

```

EditProg:LFO <>Layer:1/1
MnRate:MxRate:RateCt:Shape: Phase:
LFO1: 0.16H 4.40H ASR2 Sine Odeg
LFO2: 2.00H 0.00H OFF Sine Odeg
<more> LFO ASR FUN UTRIG >more>
    
```

Now select the ASR page to adjust the settings for ASR2. Since the cursor is highlighting ASR2 as the value for the RateCt parameter, you can select the ASR page by pressing **Edit**.

To program a realistic delayed vibrato, you need to adjust the Mode, Delay, and Attack parameters. Select the Mode parameter and change its value to **Hold** (use the Alpha Wheel or **Plus/Minus** buttons). This will prevent the vibrato from fading as it did. (This fading was caused by the ASR repeating, which was the default setting.) Now select the Delay parameter and set its value to .4 seconds (**4, 0, Enter**). Select the Attack parameter and change its value to .48 seconds (**4, 8, Enter**). The vibrato should now begin to fade in gradually after a short delay, then remain constant at a rate of 4.40 cycles per second. The ASR page should look like the page below. We're finished with the delayed vibrato; next is the velocity stab.

```

EditProg:ASR <>Layer:1/1
ASR1: Trig: Mode: Delay: Attack:Releas:
      ON   Hold 1.00s 1.00s 1.00s
ASR2: ON   Hold 0.40s 0.48s 1.00s

<more> LFO ASR FUN UTRIG <more>

```

We want the stab to drop the trumpet's pitch *and* amplitude, but only when notes are played fortissimo or louder. This is done by using ASR1 as a control source on both the PITCH and AMP pages, then using a velocity trigger (VTRIG) to control ASR1.

First return to the PITCH page (if you're still on the ASR page, press the **more>** soft button three times, and the **PITCH** soft button will appear). Press **PITCH**, then select the Src1 parameter, and set its value to **ASR1** (press **1, 1, 0, Enter** or hold **Enter** and strike G 5). Then select the Depth parameter, and set its value to **-1200 cents** (+/-, **1, 2, 0, 0, Enter**). ASR1 has nonzero default values in Program 199, so you'll hear the pitch drop an octave if you strike and hold a note. The PITCH page should look like this:

```

Edit Prog:PITCH >Layer:1/1
Coarse:0S1 Src1 :ASR1
Fine :0ct Depth :-1200ct
FineHz: 0.00Hz Src2 :LFO1
KeyTrk:0ct/key DptCt1:ASR2
VelTrk:0ct MinDpt:0ct
MaxDpt:-10ct
<more> ALG LAYER KEYMAP PITCH <more>

```

The next step is to adjust the characteristics of the stab by programming ASR1. Select the Src1 parameter again, and press **Edit** to select the ASR page.

First, select the Trig parameter, and assign a value of **VTRIG1** (press **1, 0, 6, Enter** or hold **Enter** and strike **D# 5**). Select the Delay parameter and set its value to **.34** seconds (**3, 4, Enter**). Set the Attack parameter to a value of **.78** seconds. The ASR page should now look like this:

```

EditProg:ASR <>Layer:1/1
ASR1: Trig: Mode: Delay: Attack:Releas:
      VTRIG1 Hold 0.34s 0.78s 1.00s
ASR2: ON Hold 0.40s 0.48s 1.00s

<more> ALG LAYER KEYMAP PITCH <more>
  
```

Notice that the soft buttons are identical to those on the PITCH page; this is because you pressed **Edit** to call up the ASR page, instead of using the **<more>** soft buttons.

To make the stab sound realistic, we'll drop the amplitude at the same rate as the pitch. To do this, select the **F4 AMP** page (press the **more>** soft button once, then press the **F4 AMP** soft button). Select the Adjust parameter, and assign a value of **8 dB**. This will give the trumpet a bit more punch. Next, select the Src1 parameter, and set its value to **ASR1**. Then select the Depth parameter and set its value to **-68 dB**.

Here's an important point to remember: ASR1 is being used to control both the drop in pitch for the stab, and the drop in amplitude as well. All of the control sources can be similarly assigned for as many parameters as you like.

The final step in this example is to set the velocity threshold of the stab. Right now the stab is occurring on almost every note, but we want it to happen only when playing fortissimo or louder. To do this, press the **more>** soft button three times, then press the **VTRIG** soft button to select the VTRIG page. The VTrig1 Level parameter is already selected, so just turn the Alpha Wheel until the VTrig1 Level value is **ff**. Now you can play softly without triggering the stab.

That's it for Example 1. If you want to save your work, the easiest way is to press **Exit**. The K2600 will ask you if you want to save. You should probably press the **Rename** soft button, give your program a new name, then save it with a new ID. See page 5-3 if you need help with the Save dialog.

## Example 2

### Lowpass Filter, Envelopes

This example will show you how to assign a DSP function to an algorithm block (the 4-pole lowpass filter), and adjust its control parameters. You'll also set up an envelope to control the cutoff frequency of the filter.

Start with Program **199**, press **Edit**, and select the KEYMAP page (This is explained in a bit more detail in the previous example, if you haven't read it.) Change the value to **6 Ensemble Strings**. String sounds are especially responsive to lowpass filtering, because they have a great deal of activity in the higher frequencies. Lowpass filters, depending on their cutoff frequencies, attenuate high frequencies, so a bit of adjustment can alter a string sound considerably.

Next, press the **ALG** soft button to select the ALG page. The center DSP function block will already be selected. Set its value to **4POLE LOWPASS W/ SEP**. Notice how the sound changes.

Press the **more>** soft button, and the three soft buttons that select the control-input pages for the lowpass filter will appear. Press the **F1 FRQ** soft button.

The Coarse Adjust parameter will already be selected. Try a few different values for this parameter, to get a feel for its effect on the sound. Then set it to a value of **G#3 208 Hz**. Use the cursor buttons to select the Src1 parameter, and set its value to **MPress (3, 3, Enter)**. Cursor down once to the Depth parameter, and set it to a moderate value, like **3800 cents**. Now try playing and applying pressure to the keys. (Obviously, this will have no effect if your MIDI controller doesn't send mono pressure.)

Now we're going to use an envelope to control the sweep of the filter. The K2600 has one envelope dedicated to shaping the output of the sound; this is called AMPENV. However, two other envelopes (ENV 2 and ENV3) can be assigned to control other parameters. You can assign AMPENV as a Control Source, but using ENV2 and ENV3 lets you assign separate envelopes to do different things.

Cursor down to the Src2 parameter, and set its value to **ENV2 (1, 2, 1, Enter)**. Move the cursor down to the DptCtl parameter and set its value to **ON**. Cursor down to the MaxDpt parameter, and set its value to **5800 cents**. As soon as you set a depth, you'll hear the envelope sweep the cutoff frequency. We'll adjust it further in a minute. The F1 FRQ page should now look like this:

```

Edit Prog F1 FRQ(4P LUPHSS) >Layer:1/1
Coarse:G#3 208Hz      Src1 :MPress
Fine :0ct            Depth :3800ct
                        Src2 :ENV2
KeyTrk:0ct/key       DptCtl:ON
VelTrk:0ct           MinDpt:0ct
Pad :0dB             MaxDpt:5800ct
<more F1 FRQ F2 RES F3 SEP F4 AMP more>

```

Next, press the **F2 RES** soft button to select the page for adjusting the resonance of the lowpass filter. Select the Src1 parameter, and set its value to **Data (6, Enter)**. Select the Depth parameter and set its value to **30.0 dB**. This means that the partials at the cutoff frequency will be boosted 30 dB. The F2 RES page should look like the following diagram.

If your MIDI controller has a data slider (or a control that can be programmed to send MIDI 06), the data slider (or programmable control) will now affect the resonance of the filter. Be aware though that the change in resonance is subtle; resonance affects some waveforms more noticeably than others. You may want to set the Pad parameter on F1 FRQ page to a value of 18dB or so, then set the Gain parameter on the OUTPUT page to the same value; this reduces the potential of clipping due to resonance.

```

Edit Prog F2 RES(4P LUPHSS) >Layer:1/1
Adjust: 0.0dB        Src1 :Data
                        Depth :30.0dB
                        Src2 :OFF
KeyTrk:0.00dB/key   DptCtl:MWheel
VelTrk:0.00dB       MinDpt:0.00dB
                        MaxDpt:0.00dB
<more F1 FRQ F2 RES F3 SEP F4 AMP more>

```


Next press either of the **<more>** soft buttons until you see the **ENV2** soft button. Press it to select the ENV2 page. Here you'll program Envelope 2 to control the filter's cutoff frequency.

The Att1 time parameter will already be selected; set its value to **0.50 seconds**. Press the right cursor button once to select the Att2 time parameter and set its value to **1.06 seconds**. Cursor down once to select the Att2 level parameter and set its value to **-1%**. Cursor right once, then up once, selecting the Att3 time parameter. Set its value to **3.52 seconds**. The ENV2 page should look like the diagram below. Play around with the sound a bit. Hold a few notes through the full length of ENV2; the sound will continue to change for several seconds. Work your controller's Mod Wheel and data slider through several different positions to get an idea of the range of variation you can add by using the two control sources in tandem.

```

edit:Prog#ENV2 [1/1] <>Layer:1/1
Att1:Att2:Att3:Dec1:Rel1:Rel2:Rel3:Loop:
0.50 1.06 3.52 0s 0s 0s 0s Off
100% -1% 100% 0% 0% 0% 0% Inf

```



```

<more  AMPENV ENV2 ENV3 ENVCTL  more>

```

## Example 3

### Sample and Hold Using a FUN

This example will use one of the FUNs to create a sample and hold program. As usual, start with Program 199, and press **Edit**. While you're on the ALG page, select a value of **PARAMETRIC EQ** for the center DSP function block. Then select the **KEYMAP** page, and select the keymap **152 Dull Sawtooth**.

Now press the **more>** soft button, then the **F1 FRQ** soft button. This enables you to set the frequency for the parametric EQ, which will set the depth of the modulation for the sample and hold function. The Coarse parameter will already be selected; set its value to **D#4 311 Hz**. Cursor down to the KeyTrk parameter, and set its value to **100 cents per key**. Cursor over to the Src1 parameter, and set its value to **FUN1 (1, 1, 2, Enter)**. Cursor down once to the Depth parameter, and set its value to a fairly large negative number, like **-4000 cents**.

Next, press the **F3 AMP** soft button, to set the amplitude for the Parametric EQ (make sure you don't select **F4 FINAL AMP**). Set the value of the Adjust parameter to about **17 dB**.

Now press either **<more>** soft button until you see the **LFO** soft button. Press the **LFO** button to select the LFO page. We'll program LFO1 as a square wave, then use it as one of the FUN's inputs.

Select the MnRate parameter for LFO1 and set it to a value of **5.00 Hz**. Select the MxRate parameter and set its value to **15.00 Hz**. Select the RateCt parameter and set it to a value of **Data**. Select the Shape parameter and set its value to **Square**. That's it for the LFO page.

Next press the **FUN** soft button to select the FUN page. Select the Input a parameter for FUN1, and set its value to **LFO1 (1, 1, 4, Enter)**. Select the Input b parameter and set its value to **RandV1 (1, 0, 8, Enter)**. Finally, select the Function parameter, and set its value to **Sample B On A**. Now when you play and hold a note, you'll hear the sample and hold effect. Use your MIDI controller's data slider (or a programmable control set to MIDI 06) to control the rate of the effect.

Here's what's happening. The square wave of the LFO cycles from +1 to -1. Every time this occurs, the random signal generator (RandV1) randomly picks a value which changes the frequency of the parametric EQ. There are lots of ways to set up sample and hold effects, but the FUN is the basic element. In this case, every time the value of the FUN's Input a (LFO1) becomes greater than +.5, the value of Input b (RandV1) is sampled. That value becomes the FUN's output until the next time the value of Input a becomes greater than +.5 (see Chapter 17 for more). In this example, the FUN's output modulates the frequency of the EQ, and causes rapid changes in timbre.

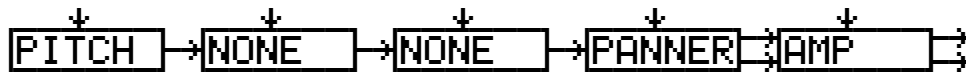
Now we'll apply the FUN to control the pitch of the sound as well. Press either of the <more> soft buttons until the **PITCH** soft button is visible, and press it to select the PITCH page. Select the Src1 parameter and set its value to **FUN1**. Select the Depth parameter and set its value to **400 cents**. Now the sound's pitch will fluctuate in sync with the EQ effect.

## Example 4

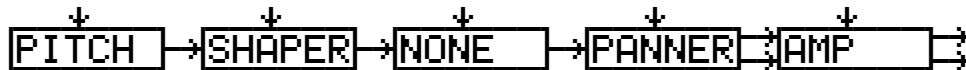
### SHAPER and PANNER

Our next example incorporates two of the DSP functions, and will give you a general overview of using the algorithms to build sounds.

Starting with Program 199, press **Edit**, and while you're on the ALG page, cursor up to the Algorithm parameter, and select **Algorithm 13**. Select values of **NONE** in the F1 and F2 blocks. The value of the F3 block is already set to **PANNER**, as shown below.



Next press the **KEYMAP** soft button to select the KEYMAP page, and select **163 Sine Wave** as the keymap. Play a few notes to accustom your ear to the sound of the unshaped sine wave. Then return to the ALG page, and select **SHAPER** for the algorithm's F1 block. You'll notice a bit of an effect right away. The SHAPER DSP function adds additional high frequency partials to the waveform in unpredictable ways, resulting in large changes in timbre. For more on the SHAPER, see page 16-46.



Press the <more> soft button, then press the **F1 AMT** soft button. This enables you to adjust the amount of shaping applied to the sine wave. Try different values for the Adjust parameter, then set it to its minimum—**0.100x**. Then cursor down to the VelTrk parameter, and set a value of **0.70**. Listen for the variation in the effect as you play with different attack velocities.

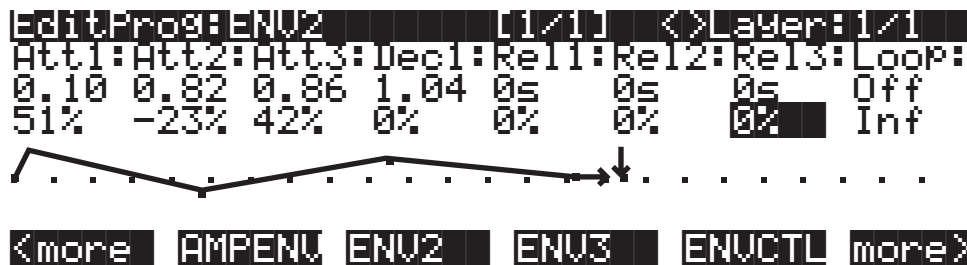
Now go back to the KEYMAP page, and select different keymaps so you can hear the SHAPER's effect on different sounds. When you're finished experimenting, set the Keymap parameter at **152 Dull Sawtooth**.

Notice how notes on the high end tend to break up. If you return to the F1 AMT page you can reduce this distortion and/or aliasing with the KStart and KeyTrk parameters.

Select the KStart parameter and set a value of **C 2 Unipolar**. This limits the amount of shaping applied to notes above or below C 2, depending on the value of the KeyTrk parameter. Next set the value for the KeyTrk parameter to **-0.018x per key**. Since we're using a negative value, the amount of shaping will decrease with higher notes.

Next we'll program an envelope to change the SHAPER in real time. While still on the F1 AMT page, select the Src2 parameter, and set it to a value of **ENV2 (1, 2, 1, Enter)**. Cursor down to the DptCtl parameter and set it to a value of **AttVel (1, 0, 0, Enter)**. Set the MinDpt parameter to **0.00x** and the MaxDpt parameter to a value of **1.70 x**. This will let you use attack velocity to control the amount of the envelope's effect on the SHAPER.

Next, use the **<more>** soft buttons to locate the **ENV2** soft button, then press it to select the ENV2 page. Set up the parameters as shown in the diagram on the following page.



This can still be a little harsh on the high end when you play with high attack velocities. One way to smooth it out would be to go back to the ALG page, select a lowpass filter for the F2 block, and adjust its cutoff frequency to about F# 6. This is done by pressing **Edit** when the F2 block is selected, then selecting the Adjust parameter and changing the value with any data entry method.



That's it for the SHAPER example; we'll continue with this program to describe the PANNER. You'll notice that in Algorithm 13, **PANNER** is the only value available for the F3 block. The PANNER function takes a single signal from the sound engine, and splits it into two. These are referred to as the upper and lower wires. The upper and lower wires pass independently into the final Amp block, and from there to the audio outputs.

The parameters on the control-input page for the PANNER let you distribute the signal between the upper and lower wires. You can send the signal all to the lower wire (an Adjust value of **-100%**), all to the upper wire (**100%**), or anywhere in between. This in itself won't necessarily change the pan position of the current layer. It works in tandem with the Pan parameter on the layer's OUTPUT page.

When a layer uses an algorithm that contains the PANNER function, you always have two wires going through the final Amp and to the audio outputs. Consequently, on the layer's OUTPUT page, there are parameters to assign the output pair and pan position of each wire. When you have one wire panned hard left, and the other hard right, changing the parameters on the PANNER control-input page will enable you to move the layer's pan position in real time. The closer a layer's output is to the center of the stereo field, the less effective the PANNER function will be.



The first step in our PANNER example, therefore, will be to select the OUTPUT page. Select the Pan parameter for the upper wire, and set it all the way to the right. Select the Pan parameter for the lower wire and set it all the way to the left.

Now you can select the F3 POS page, and program it to move the sound around. First try changing the value of the Adjust parameter. You should hear the sound move to the left when the value is negative, and to the right when it's positive. Set it back to 0%, then select the Src1 parameter. Select **LFO1** as the value, then select the Depth parameter and set it to 100%.

Now select the Src1 parameter again, and press **Edit** (or use the **more>** soft button to locate the **LFO** button, then press it). You will now see the LFO page. Set the MnRate parameter for LFO1 to a value of **0.1 Hz**. Set the MxRate parameter to a value of **2.00 Hz**, and the RateCt parameter to **Data**. Leave the Shape parameter set to **Sine**. You should hear the sound shift slowly from left to right as the LFO cycles. You can adjust the speed of the shift with your MIDI controller's data slider (or a programmable control set to MIDI 06).

## Example 5

### Building a Drum Program; Using the Keymap Editor

With our next example, you'll learn how to build a drum program using the Program and Keymap Editors. To keep the example as brief as possible, we'll include only a few timbres and DSP examples. This won't make for a terribly realistic drum program, but it will give you the basic ideas you need to build your own. In this example, you'll create a four-layer program, with a different percussion timbre in each layer, each timbre having a different set of DSP functions applied.

Start with the default program 199. Press **Edit**, then press the **KEYMAP** soft button. Select a value of **168 Silence**. This gives you a keymap with a single key range from C 0 to G 10. Select the KeyTrk parameter, and change the value to **0**. This will make the pitches of all the samples you assign the same on each key (you won't hear anything until you assign the samples). Next, press the **<more** soft button once, then press the **DupLyr** soft button. Layer 2 will be created. Repeat this twice, until you have four layers.

When you've created the four layers, you'll notice that the top line of the display tells you that you're looking at Layer 4 of a 4-layer program. Press the **Chan/Bank Up** button to return to Layer 1. Press the **LAYER** soft button to select the LAYER page. Set the LoKey and HiKey parameters to **C 4** and **D 4**. The easiest way to do this is to select the LoKey parameter, hold the **Enter** button, and strike C 4 on your MIDI controller. Do the same for the HiKey parameter, striking D 4. Press the **Chan/Bank Up** button to select Layer 2. Set its LoKey and HiKey to **D<sup>#</sup> 4** and **F 4**. Repeat this for Layer 3, setting its LoKey and HiKey parameters to **F<sup>#</sup> 4** and **G<sup>#</sup> 4**. Do the same for Layer 4, setting its LoKey and HiKey parameters to **A 4** and **B 4**. This might be a good time to save what you've done so far.

Next, return to Layer 1 (**Chan/Bank** button), press the **KEYMAP** soft button, select the Keymap parameter, and press **Edit** to enter the Keymap Editor. Select the Sample parameter, and select a value of **64 15in Dry Tom-C 4**. Press the **Save** soft button, and the K2600 will prompt you to save the keymap. Rename it as **Tom**, and save it to an unused ID (don't replace **168 Silence**). Press **Exit** to return to the Program Editor, and select Layer 2 with the **Chan/Bank** buttons. Press **Edit** to return to the Keymap Editor, select the Sample parameter, and assign a value of **47 Dry Kick 1 C 4**. Save the keymap, renaming it **Kick**. Press **Exit** to return to the Program Editor, and select Layer 3. Return to the Keymap Editor, select the Sample parameter, assign a value of **55 Dry Snare 2-C 4**, save the keymap—renaming it **Snare**. Return to the Program Editor, select Layer 4, select the Keymap parameter, and assign a value of **42 Closed Hihat-C 4**. Save the keymap, renaming it **HiHat**. You now have a four-layer program, each layer having its own keymap with a different sample assigned to each one.

This is the basic process for creating any keymap and incorporating it into a program. In this case, we don't want the layers to overlap, and we want each layer to use a distinct keymap with its own sample assignment. In other programs, you might want to create a keymap with different timbres in a single layer, and you might want the layers to overlap.

As an example of how to quickly set up a multi-sample keymap, we'll change the sample assignment in Layer 1. Return to the Program Editor, and select Layer 1. Return to the Keymap Editor (by pressing **Edit** while the Keymap parameter is highlighted on the EditProg\*KEYMAP page), then press the **NewRng** soft button. The K2600 will prompt you to strike a low and high key on your MIDI controller. Strike C 4 and C# 4. Notice that the value of the Key Range parameter changes to reflect the new range assignment. Now select the sample parameter, and turn the Alpha Wheel one click to the right, to select the sample **65 13in Amb Tom-C 4**. Save the keymap, replacing the earlier version. You can repeat this process to create as many new key ranges as you like (in this example, doing so would have no effect, since we've limited each layer to a narrow three-key span). Make sure the value of the KeyTrk parameter on the Keymap page is set to **100ct/key**.

If you wanted the layers to overlap, you would simply set each layer's LoKey and HiKey parameters (on the LAYER page in the Program Editor) to the same respective values. For example, you might set each LoKey parameter to C 2, and each HiKey parameter to C 7, causing all layers to play across five octaves.

Now we'll add some processing to some of the layers in our drum program example. The fact that each sound is on a different layer enables us to use a different algorithm for each layer, giving us enormous control over each sound.

Return to the Program Editor and select Layer 1. Press the **PITCH** soft button, and select the Coarse transpose parameter. Set its value to **-3 ST**. Select the KeyTrk parameter, and set its value to **400 cents per key**. This will give you a much different pitch on each key.

Next, select the PITCH page and select the Src2 parameter. Set its value to **ENV2 (1, 2, 1, Enter)**. Select the MaxDpt parameter and set its value to **300 cents**. Select the DptCtl parameter, and set its value to **VTRIG1 (1, 0, 6, Enter)**. Press **Edit** to select the VTRIG page, and set the VTrig1 Level parameter to a value of **fff**. Return to the PITCH page, select the Src2 parameter, and press **Edit** to select the ENV2 page. Set it up as shown in the ENV2 page below.

```

edit:Prog:ENV2 [1/1] <>Layer:1/1
Att1:Att2:Att3:Dec1:Rel1:Rel2:Rel3:Loop:
0.02 0.16 0s 0.38 0s 0s 0s Off
28% -100 0% -76% 0% 0% 0% Inf

```

↓

```

<more> AMPENV ENV2 ENV3 ENVCTL >more>

```

Next, select Layer 2 and select the F4 AMP page. Set the Adjust parameter to **8 dB** to give the kick a little more presence. Select the Src1 parameter and set it to a value of **AttVel (1, 0, 0, Enter)**. Select the Depth parameter and set a value of **10 dB**. The kick will get considerably louder as you strike the keys harder.

Select Layer 3, and press the **ALG** soft button. Select the center, 3-stage DSP function block and assign a value of **HIFREQ STIMULATOR**. Press the **more>** soft button once, then press the **F1 FRQ** soft button. Select the Coarse Adjust parameter, and assign a value of **G 10 25088 Hz**. Select the Src1 parameter and assign a value of **MWheel**. Select the Depth parameter and set it to

a value of **-10800 cents**. Move your MIDI controller's Mod Wheel to bring out the snare's high end.

## Example 6

### Editing a Setup for KB3 Control

1. Go to Setup mode, and select **97 ControlSetup**.
2. Press **Edit** to enter the Setup Editor.
3. Press the **more>** soft button, then press the **FOOTSW** soft button.
4. Use the cursor buttons to highlight **TapTempo** (it's the value of the Dest parameter for FtSw4).
5. On the alphanumeric button pad, press **6 8 Enter**. This changes the value to **LegatoSw**, which is MIDI 68 (for VAST programs, MIDI 68 will now control Legato Switch, but for KB3 programs, it will switch between slow and fast rotary speaker effect).
6. Press the **more>** soft button, then press the **RIBBON** soft button.
7. Highlight **AuxBend2** (it's the value of the Dest parameter for SmRib).
8. On the alphanumeric button pad, press **1 6 Enter**. This changes the value to **Ctl A**, which is MIDI 16.
9. Highlight **MPress** (it's the value of the Dest parameter for SmPrs).
10. On the alphanumeric button pad, press **1 7 Enter**. This changes the value to **Ctl B**, which is MIDI 17.
11. Highlight **AuxBend1** (it's the value of the Dest parameter for LgRib—or Sect1 if the large ribbon is configured in three sections).
12. On the alphanumeric button pad, press **1 8 Enter**. This changes the value to **Ctl C**, which is MIDI 18.
13. Press the **SWITCH** soft button.
14. Highlight **ArpSw** (it's the value of the Dest parameter for PSw1—Button 9 above the Pitch Wheel).
15. On the alphanumeric button pad, press **6 9 Enter**. This changes the value to **FrezPd**, which is MIDI 69.
16. Highlight **MIDI29** (it's the value of the Dest parameter for PSw2—Button 10 above the Mod Wheel).
17. On the alphanumeric button pad, press **7 0 Enter**. This changes the value to **MIDI70**.
18. Press **Exit**. The K2600 will ask you if you want to save the setup. Press the **Rename** soft button.
19. Press either **Chan/Bank** buttons until the KbdNaming parameter in the top line of the display shows a value of **Adv**. This lets you rename the setup by playing notes (in this

example, we're going to rename it as **KB3 Setup**). Of course, if you prefer, you can use soft buttons and data entry (Alpha Wheel, cursor buttons, or alphanumeric button pad) to rename the setup.

20. Hold down G 6, then strike D 4 (or do the equivalent on your MIDI source). This enters a K as the first character in the name.
21. Hold down G 6 and strike B 2 to enter a **B**.
22. Release G 6, then press A # 2 to enter a 3.
23. Strike G # 6 three times to delete three characters. Notice how the remaining characters shift to the left.
24. Strike G 2 to enter a space. You're finished renaming.
25. Press **OK**. The save dialog reappears. Change the setup's ID if you want, then press **Save**. You'll return to Setup mode.

## Example 7

### Using the KB3 Setup From the Accessory Disk

1. Insert the floppy disk called **K2600 Additional Programs and Factory RAM Banks** into the K2600's floppy drive.
2. Set the value of the CurrentDisk parameter to **Floppy**, if it isn't already there.
3. Use the cursor buttons to highlight the directory called **MOREPRGS**, then press **Open** or **OK**.
4. Highlight the file **KB3.K26**, if it isn't already highlighted. Press **OK**. The K2600 prompts you to select a memory bank.
5. Select a different bank if you wish, then press **OK**.
6. Press **Fill**, which determines how the K2600 assigns an ID to the file you're loading. This loads the file into the first available ID in the bank you selected—without affecting any objects already stored in the bank. When the file has loaded, the K2600 will return to the Disk-mode page.
7. Press the **MIDI** mode button, then press the **XMIT** soft button.
8. Highlight the value of the CtlSetup parameter, then press **2 0 0 Enter** on the alphanumeric buttonpad.