





## The MicroWave Programming Manual

<b>1. Introduction</b>	1-01
1. About this manual	1-03
2. The front panel	1-05
<b>2. Sound-architecture</b>	2-01
1. General description	2-03
2. Dynamic Spectral Wavetable synthesis	2-05
3. About modulators	2-08
4. Sound Init	2-14
<b>3. Audio modules</b>	3-01
1. Oscillators 1&2	3-04
2. Waves 1&2	3-10
3. Volume	3-21
4. Filter	3-29
5. Panning	3-36
<b>4. Modifier modules</b>	4-01
1. Volume Envelope	4-05
2. Filter Envelope	4-10
3. Wave Envelope	4-15
4. LFO 1	4-22
5. LFO 2	4-27
6. Glide	4-29
7. Temperment	4-31
8. Name	4-33
<b>5. Quick Edit</b>	5-01
1. Fast Access	5-05
2. Envelope Macros	5-16
3. Modulation Macros	5-20
<b>6. Storage</b>	6-01
1. Storing Sound-programs	6-04
2. Copying Sound-programs	6-06
3. Compare Sound-programs	6-07
4. Recall Sound-programs	6-09
5. Storing Multi-programs	6-10
6. Copying Multi-programs	6-12
7. Compare Multi-programs	6-13
8. Recall Multi-programs	6-15
9. Storing all edit-buffers	6-16
10. Back-up to Card	6-18
11. Load from Card	6-20
12. Back-up to MIDI	6-21
13. Load from MIDI	6-25
<b>7. Sound samples</b>	7-01
1. Sound-init Program	7-04
2. Analog Brass	7-10
3. Rolling Glass	7-19





1. Einführung

2. Sound-Aufbau

3. Audio-Module

4. Modifikations-Module

5. Quick Edit

6. Speichern

7. Sound-Samples





# Chapter 1

Introduction

**1.1 About this manual**

**1.2 The front panel**



# 1. Introduction





# 1. About this manual

## 1. About this manual

This manual is divided into two separate sections:

- This *Programming manual*
- The *Performance manual*

This Programming manual deals with all aspects of tweaking sound parameters or programming a sound from scratch. You will find anything that refers to the generation of a MicroWave Soundprogram within the pages of this manual, with the exclusion of programming any Velocity- or Tuning-tables or Program-change maps.

It does not, however, include any description of program selection, setting up the MIDI parameters, building Multi-programs for keyboard- or sequencer-use and generally using the sounds of the MicroWave. This is all covered in the MicroWave's *Performance manual*.

Before reading this Programming manual, you should at least be familiar with the front panel and basic MIDI functions of the MicroWave. You also should know how to set up and connect the MicroWave. You need not necessarily know how to create and edit Multi-programs, but it definitely wouldn't hurt.



# 1. Introduction

Throughout both manuals you will find some standards that will ease your way through the information:

▶ When the manual directs you to press a button it will be referred to in [brackets].

→ [parameter/value button], [select button], [alpha dial] etc.

▶ All labeling on the front panel will be referred to in <triangular parentheses>:

→ <MIDI mode> <Instrument Select> etc.

▶ All information on the MicroWave's display will be shown this way:



Sound Prog: 001 E  
Sound Init

▶ A 'How-to' description will use a black dot for each consecutive step:

● Read on!

▶ Any helpful explanation in cases where you might be stuck is introduced with an arrow:

→ If you can't read this, consult your optician.

All parameter-descriptions start with the parameter printed in **bold typeface**, followed, in parentheses, by the possible value-range in *italics*.

→ **Mastertune** (*-64.00..+63*)

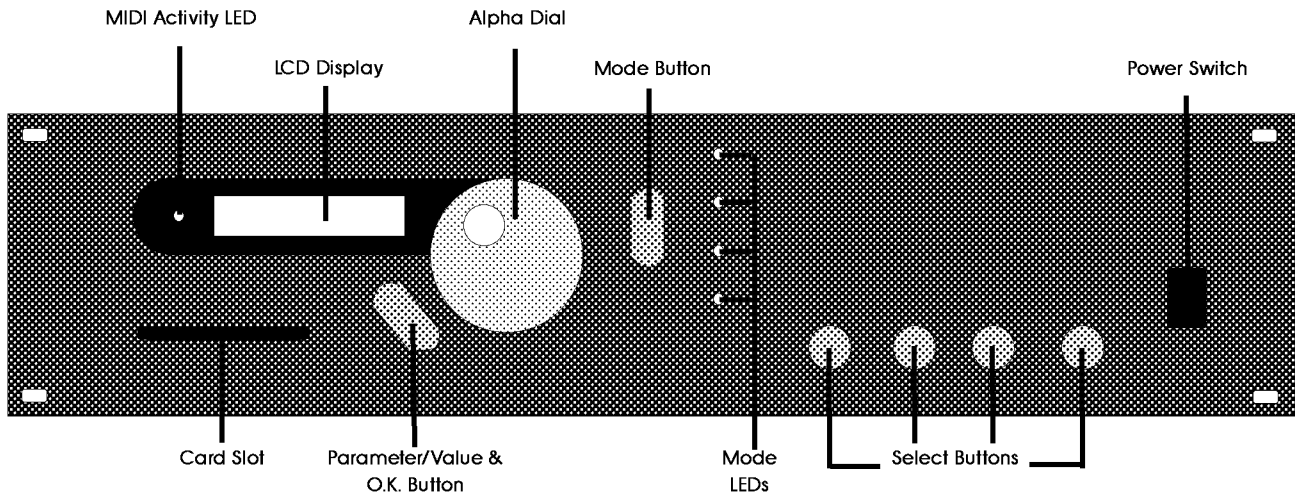
▶ In general any parameter value is referred to in *italics*:

→ *Gliss. O, 127*

## 1.2 The front panel

### 1.2. The front panel

After reading the Performance manual you already should be familiar with the front panel. It is laid out clearly and is easy to understand, helping you in mastering all parameters of the MicroWave. For those without battery-backup of their grey-matter (like me), here again the basic elements:



Let's start at the very left.

**MIDI activity LED:** indicates whenever a MIDI message is received by the MicroWave. When a note-on message is received, the LED will remain lit until the corresponding note-off is received, thus showing the actual duration of the note. The LED will display only a MIDI message meant for the MicroWave.

**Card slot:** here you can insert a MicroWave cartridge from Waldorf or a third party developer for storing and recalling programs. There are two types of cards:

**ROM:** This card will contain programs that you can read only.

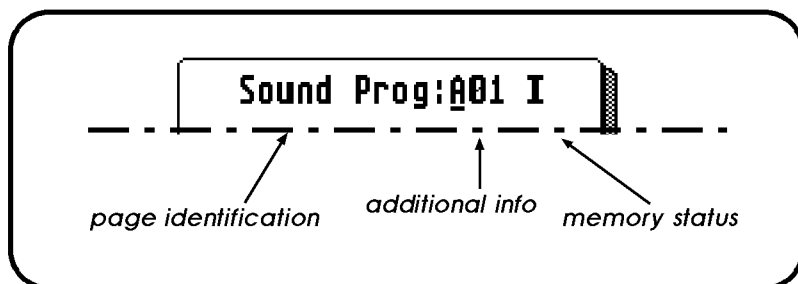
**RAM:** You can store programs onto this card yourself.

When inserting a card, the label should always be face up. You won't damage your MicroWave or the card by inserting it the wrong way; it simply won't work.

## 1.2 The front panel

**Display:** This is the window between you and the MicroWave. All necessary information is presented here. The display provides two lines of text that have specific functions.

The top line is the *status line*, which will indicate what mode or page you currently are in. For example, when selecting single Sound-programs, it will read the following:



*Sound Prog* is the *page identification*, telling you the mode and page you are currently in

*A01* is the *additional info*, here the number of the selected Sound-program.

*I* indicates the current *memory status*. In this example the program is in the Internal memory. The following memory status are available:

**I: Internal memory:** The sound is resident inside the MicroWave.

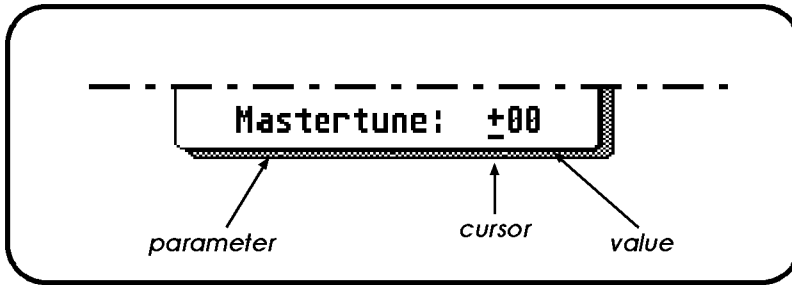
**C: card memory:** The sound is located on the card.

**E: Edit buffer:** The addressed sound resides in one of the eight available edit-buffers. This sound can be different from the original sound stored in memory. To save this program you must store it either to internal- or card-memory before switching off the MicroWave.

**S: Swapped sound:** You are listening to the original Sound-program from which the edited version is derived. This memory status can be achieved only by using the compare function after a program was loaded into an edit-buffer. It is meant to compare the edited version of a program with it's original version.

## 1.2 The front panel

The bottom line of the display gives you more detailed information about the page currently selected:



It will tell you the *parameter* that is currently selected and its *value*. For example, if you have selected the Global parameter mode, the display might read the following:



**Mastertune:** This is the currently selected *parameter*. Depending on the current mode or page there may be more parameters to choose from.

**+00:** This is the parameter's *value*. Most values may be positive or negative, depending upon the parameter.

**\_:** This is the *cursor*. Its location indicates the function of the [alpha dial]: if the cursor is placed under the value-field it will edit the value; if the cursor sits under the parameter-field, it will select different parameters.

Each of these fields may contain more entries that will be discussed at their respective chapters.

**[Alpha dial]:** this is the means of entering and changing information. It will either select parameters or enter values, according to where the cursor is positioned in the display. Turn the [alpha dial] clockwise to increment, counter-clockwise to decrement parameters or values.

**[parameter/value button]:** This button switches the position of the cursor between the parameter- and the value-field of the display. At the same time, of course, it switches the [alpha dial] between selecting parameters or changing values.

To set a value of a certain parameter at a certain page, first select the desired page (see below). See if the cursor sits under the parameter-field. Now use the [alpha dial] until you find the desired parameter. Press the

## 1.2 The front panel

[parameter/value button] once again. The cursor should now be positioned under the value-field. Now use the [alpha dial] to enter the desired value. You may repeat this process as often as you like until you have entered all desired values of all according parameters.

**[mode button]:** This will cycle through the four distinct programming-modes the MicroWave offers. Each of those modes is dedicated to certain related aspects of working with the MicroWave. Each will bear a number of pages that group the parameters of each mode into useful, easy to manage subsets. The four modes are:

*Play mode:* This is the first and most frequently used mode of the MicroWave. It allows the selecting of Sound-programs, switching between Single- and Multi mode as well as setting up Multi-programs. This will be your main editing environment for playing as opposed to programming the MicroWave.

*Global mode:* This gives access to parameters common to all Sound-programs. Most of those parameters will also be applicable to Multi-programs, while some are overridden by corresponding parameters of a Multi-program. Parameters include Mastervolume, Stereo Width, Mastertune, general MIDI parameters, Device (Sys-Ex) parameters and programming of tuning-, velocity- and program-change-tables. Usually you will use this mode to configure the MicroWave to fit into your personal MIDI system environment.

*Sound-edit:* This mode compiles all parameters that define a timbre. Here you'll find everything for programming oscillators, waves, filter, volume and all control modules like envelopes and LFO's etc. This mode allows you to edit a sound or create one from scratch.

*Quick Edit:* This mode is a truly unique mode available only for the MicroWave. It provides you with sound-edit shortcuts, thus speeding up the process of generating new timbres. There are two types of these shortcut-parameters available: Macros and FastAccess parameters. Macros are sets of preconfigured parameters that will affect envelopes or modulation routings.

FastAccess parameters control certain parameter-groups concurrently in an intelligent manner. You will find yourself creating new sounds in seconds using this mode, and then fine tuning them in Sound-edit mode.

## 1.2 The front panel

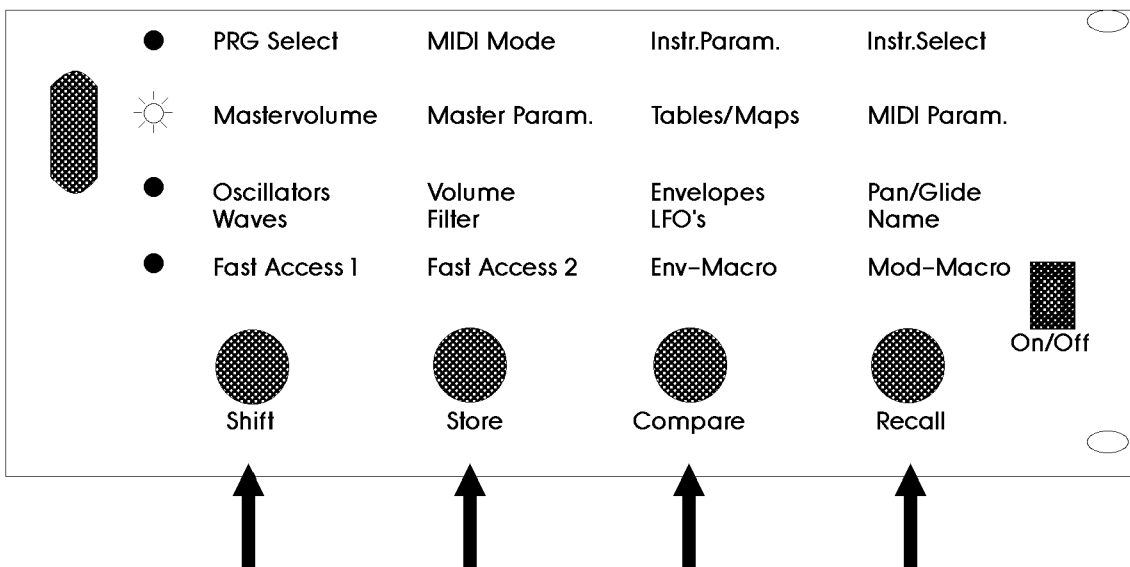
Play mode and Global mode are discussed in detail in the MicroWave's Performance manual.

These four modes are arranged in logical order from top to bottom. There is no further labeling to indicate the modes other than that of their according pages; however, within minutes you will have memorized their order.

Each mode is indicated by it's illuminated LED to the right of the [mode button]. This button will cycle through the four modes from top to bottom. Therefore, if you are in Global mode (second from above) and want to switch to Play mode (top row), you must push the [mode button] three times, cycling through the other modes to arrive at the top.

**Pages:** each mode contains a number of pages that split the parameters of that mode into manageable and useful subsets. Certain pages will contain numerous parameters, while others may contain only one. Also, the number of pages per mode varies according to the complexity of the mode. The name of a page is usually displayed in the *page-identifier-field* in the top-line of the display.

**[select button]:** These four buttons select the pages of each mode.



To arrive at a certain page, you must switch to the corresponding mode using the [mode button] and select the desired page by pressing its [select button]. To help

## 1.2 The front panel

you in finding the right pages, there is a page-matrix menu printed on the front panel that indicates what pages each [select button] will access in which mode.

There are four of those [select buttons] total, giving you quick access to any page. Depending on the mode, certain [select buttons] will cycle through a number of pages, while others address only a single function. This enables you to quickly select a frequently used function, while still getting to a more elaborate parameter with only a few button-presses.

### **Page-, parameter- and cursor-memory:**

As outlined in the Performance manual, the MicroWave's user interface is equipped with a thorough page-, parameter and cursor-memory that speeds up interactive parameter selection and comparison enormously.

At the back of this manual you will find a chart that depicts all parameters sorted by modes and pages.



# Chapter 2

Sound-architecture

**2.1 General description**

**2.2 Dynamic Spectral Wavetable  
synthesis**

**2.3 About modulators**

**2.4 Sound Init**



## 2. Sound architecture

## 2.1 General description

### 2.1 General description

This chapter outlines some general concepts of the theory behind the MicroWave's synthesis method. Read it carefully to fully visualize the power of programming sounds with the MicroWave.

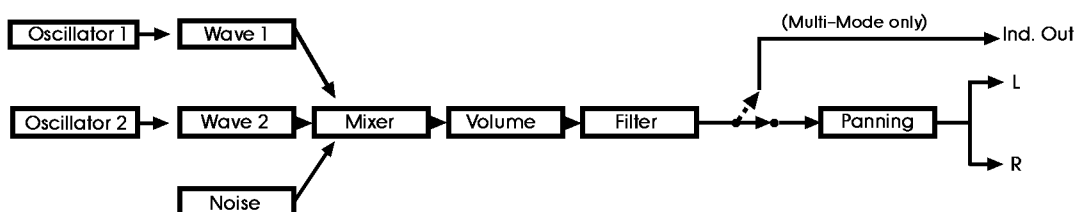
The MicroWave employs a number of different modules to achieve its sound. Each of those modules has a distinct function. There are two different groups of modules:

- *Audio modules*
- *Modifier modules*

All Audio modules are configured in a predefined manner. You cannot change their configuration at all.

Audio modules produce the actual sound you hear. However, they do it according to rules set up by the Modifier modules. Such a rule, for instance, might be "start the volume softly and gradually fade in to its maximum". This would be a typical rule set by the Volume-envelope and executed by the Volume-module.

The general signal flow of these audio modules is depicted in the figure below:



Two Oscillators with no sound of their own drive each a Wave-module that outputs the actual waveform. Both of these waveforms are fed into a Mixer, where they are joined by an additional Noise-source. The sum of that mixer is sent through a Volume-module, which shapes the sound's amplitude, followed by a Filter-module, which further processes the harmonic contents. Its output

## 2.1 General description

is connected to a Panning-module, which sends the signal to the Left and Right outputs.

This is the basic layout, in terms of programming the MicroWave.

There are a few exceptions to the above layout:

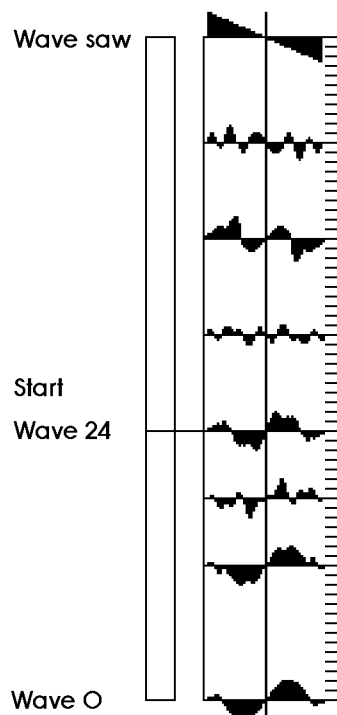
- If the *Stereo Width* parameter of Global mode (see the Performance manual) is set to mono, the panning-module will be inactive. Instead, the same mono-signal is sent to both outputs.
  
- If an Instrument is routed to an individual output, the panning-module will be inactive for that sound. The Sound-program will not appear at the Left and Right output but only at the individual output it has been assigned to.

Each of the modules has certain modulation-inputs. These modulation-inputs are fed by the Modifier modules. You will find both preconfigured as well as routeable Modifier modules.

As mentioned above, Modifier modules shape the sound produced by the Audio modules. An LFO, for instance, might shape the pitch to produce vibrato, while the filter-envelope might control the cutoff frequency of the filter.

### 2.2 Dynamic Spectral Wavetable Synthesis

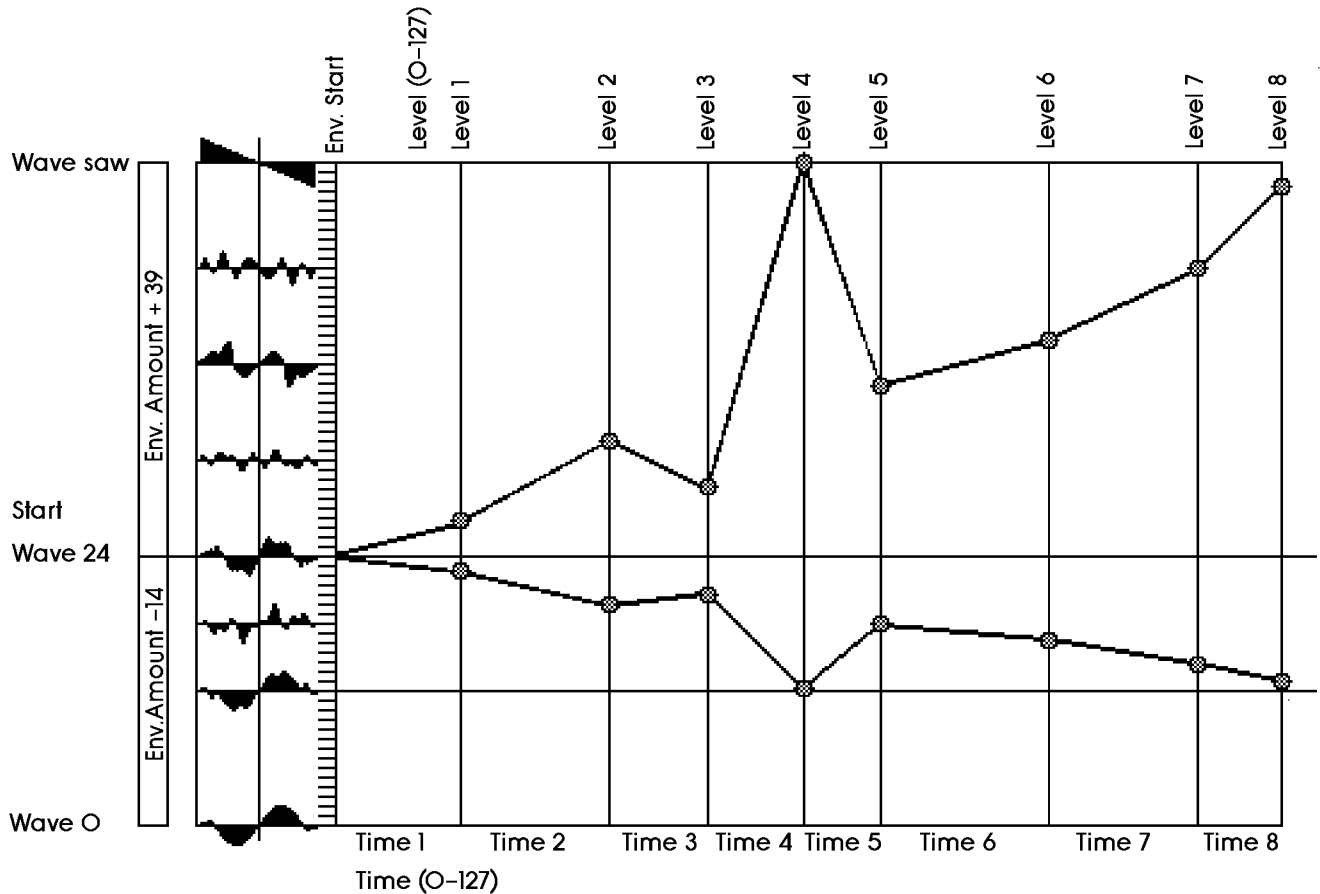
This synthesis method, besides the flexible modulation possibilities, is the key ingredient in producing the MicroWave's unique and precise timbres. It allows you to control the harmonic contents of a sound right at the source. Even more so, it gives you the capability of dynamically changing that timbre right at the source.



In Dynamic Spectral Wavetable Synthesis you start with a *Wavetable*. The MicroWave's wave-memory holds a total of 32 such *Wavetables*. Each *Wavetable* itself is a compilation of 64 *Waves*, each resembling a certain spectrum. The *Waves* of each *Wavetable* are usually compiled in a way that the spectres of each succeeding *Wave* bears some relation to the preceding *Wave*. Therefore, you might see each *Wave* as a very specific waveform with a unique spectrum that will yield a distinct timbre when processed by the other MicroWave modules. This would yield a huge array of interesting timbres; however, this alone would not be that revolutionary.

## 2.2 Dynamic Spectral Wavetable Synthesis

What really sets the MicroWave apart from all other synthesizers is its ability to *scan a Wavetable*, interpolating between the different Waves of that table, and thus dynamically changing the waveform itself.



The process of changing the waveform in realtime is done by MicroWave's unique *wavescan technology*. As stated above, a MicroWave Wavetable has 64 different positions, each encompassing a certain waveform and spectral identity. When a key is pressed, the Wavetable will output a spectrum according to the *StartWave* parameter. If no modulation is used for scanning the Wavetable, this spectrum will remain the same while a key is pressed.

## 2.2 Dynamic Spectral Wavetable Synthesis

If however, one or more modulation sources are enabled, the Wavetable will be scanned through different positions according to the defined modulations. Thus, if the multi-segment *wave-envelope* is used (as pictured above), it will change the position within the Wavetable over time as defined by its parameters. The spectrum of the Wave will change dynamically, allowing subtle or dramatic timbral evolutions as programmed into the modulators.

You can scan the Wavetable with up to four different modulation sources per Wave-module, two of which are routeable modulation inputs. This gives you tremendous power for creating rich, moving sounds that need not remain static. Consider all other means to process the Wave-modules, like Filter or Panning, and you begin to understand the vast capacity for programming unique, individual sounds.

## 2.3 About modulators

### 2.3 About modulators

As mentioned earlier, Modifier modules shape the sound, while Audio modules produce it. Please note that the MicroWave uses the principle of setting modulation-amounts at the destination exclusively. This allows you to use the same Modifier module at several destinations with different modulation amounts respectively.

Each Audio module (the Panning module being an exception) has a few *preconfigured modifier modules*. The oscillator, for instance, has pitch-bend as a standard modulation-input, while the filter has the filter envelope regularly assigned. These modifier modules can be programmed in their amplitude only; of course a value of 00 will effectively disconnect each modulation-input.

You will find the following preconfigured Modifier modules:

**Oscillators:** – Pitch bend range

**Waves:** – Wave envelope (amount and velocity)  
– Keytracking

**Volume:** – Volume envelope (amount and velocity)  
– Keytracking

**Filter:** – Filter envelope (amount and velocity)  
– Keytracking

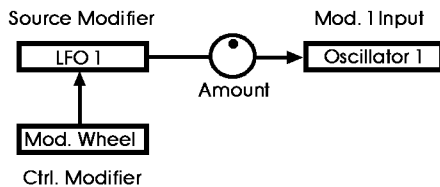
On top of this, most modules have *routeable modulation-inputs*; this is true for both the Audio as well as the Modifier modules themselves. With a routeable modulation-input, you can determine both the Modifier module used as the modulation source as well as its amount. There are two different kinds of such modulation-inputs:



## 2.3 About modulators

**Sidechain modulation-input:** These allow you to cross-modulate two Modifier modules. This way you can control the effect of the *source* Modifier module by using a different *control* Modifier module.

The figure below depicts how it works:

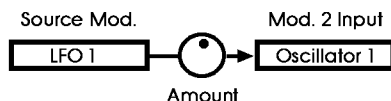


The *Source* is the Modifier module (here: LFO 1) that actually modulates the destination module (here: Oscillator 1) according to the *Amount* programmed. It bears a modulation-input of its own, the *Control* input, that allows the *Amount* to be scaled in real-time by another Modifier module (here: the Mod-Wheel).

Possible applications include:

- ▶ LFO modulations controlled by velocity, mod-wheel, after-touch or poly-pressure
- ▶ creating complex envelopes by scaling the envelope at the source-input with another envelope at the controller-input
- ▶ scaling envelope amount by velocity, keyboard tracking or a MIDI continuous controller

**Regular modulation-input:** These work as straightforward as you would expect:



The Modifier module programmed as the *Source* (here: LFO1) modulates the destination module (here: Oscillator 1) with an amplitude set in the *Amount*.

## 2.3 About modulators

Use regular modulation inputs for:

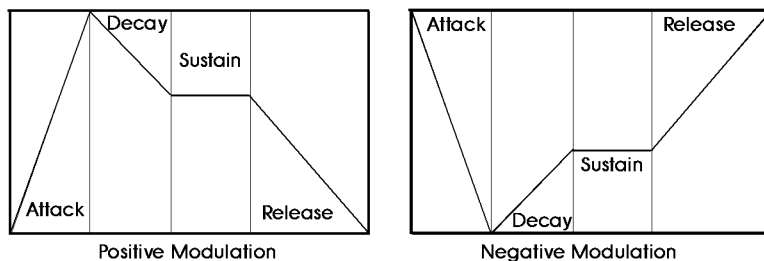
- ▶ constant LFO modulations
- ▶ regular envelope inputs
- ▶ MIDI continuous controller inputs

...and much more.

All routeable modulation-inputs share two basic functions:

**Modulation amount:** it can be set to both positive and negative values. The amount itself sets the peak of the modulation; for an ADSR envelope, it would be the level reached by the attack time.

A negative value will invert the output of the modifier module; an ADSR envelope, for instance, would be turned upside down, starting and ending at the highest level;



An LFO would be 180 degrees out of phase with the original signal produced by the Modifier module.

**Modifier modules table:** this table encompasses all modifier modules that can be used by the modulation source- and control-inputs:

*Internal MicroWave Modifier modules:*

- LFO 1
- LFO 2
- Volume envelope
- Filter envelope
- Wave envelope
- LFO envelope (the envelope of LFO 1)

## 2.3 About modulators

### *External MIDI Modifier modules:*

- Keytracking (the MIDI note number)
- Velocity (the regular attack velocity)
- Release velocity (the velocity sent upon release of a key)
  
- After-touch (monophonic)
- Poly pressure (polyphonic, individual per key)
- Pitch bend (the pitch-bend wheel or lever)
- Modulation wheel (MIDI continuous controller 1)
- Sustain pedal (MIDI continuous controller 64)
- Volume controller (MIDI continuous controller 7)
- Panning controller (MIDI continuous controller 10)
- Breath controller (MIDI continuous controller 2)
- Controller W (assigned in global mode or Multi-program)
- Controller X (assigned in global mode or Multi-program)
- Controller Y (assigned in global mode or Multi-program)
- Controller Z (assigned in global mode or Multi-program)

### *Fixed Modifier modules*

- max. (a fixed value of 127 = maximum amount)
- min. (a fixed value of 0 = minimum amount)

You can assign any of those Modifier modules to a modulation-input.

## 2.3 About modulators

### About MIDI Modifier modules:

**1.** The MIDI Modifier modules will be active only if the MIDI-controller can and will send them as programmed. If, for instance, your master keyboard does not support release velocity or poly pressure, it is useless to program these modifier modules since they will never be sent to the MicroWave from that keyboard.

**2.** The MIDI Modifier modules *Controller W* through *Z* support any MIDI continuous controller between values of 000 and 120. In order to use these user-definable MIDI controllers, you must program the actual MIDI controller number they shall represent in Global mode:

- Switch to Global mode.

- Press the [select button] labeled <MIDI Param.>. The page-identifier field should read "MIDI Param."

You now are in the Global MIDI parameter page.

- Use the [alpha dial] to select any one of the parameters Control.W to Control.Z.

All MIDI-controller values between 000 and 120 can be assigned to Controller W thru Z. Be aware that switch controllers (*# 64 and up*) will output only two values: *0* if they are not pressed and *127* if they are activated.

If you use Multi mode, each Multi-program can be assigned it's individual Controller W thru Z values. Consult the Performance manual about programming the respective Control W thru Z parameters.

**3.** Be aware that when assigning a MIDI Modifier module you should set it to be *active* at the MIDI-filters of Global mode or in the Multi-program. If the according MIDI controller is filtered, it is useless to assign it anywhere since all it's data will be ignored at the MIDI input of the MicroWave.

## 23 About modulators

### About the fixed value Modifier modules:

The *min.* Modifier module is especially handy when you want to disable a modulation input quickly. Just assign the *min.* Modifier module to set a fixed source value of 0.

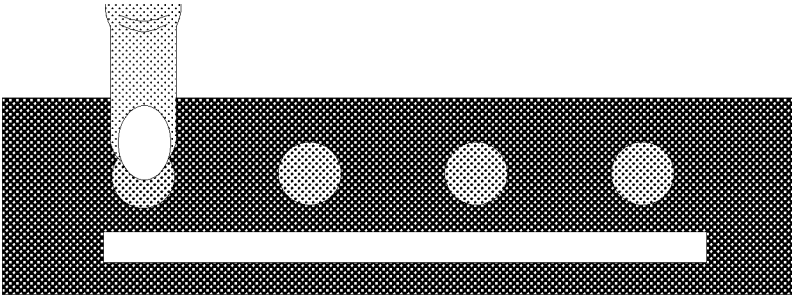
The *max.* Modifier module, on the other hand, comes in handy if you don't want to use the sidechain-capability of a modulation input. Assign the *max.* Modifier module to the *control-input* to disengage it. Also, if you need simply a fixed value somewhere, assign *max* to the source-input and set the desired fixed value using the amount value.

## 2.4 Sound – Init

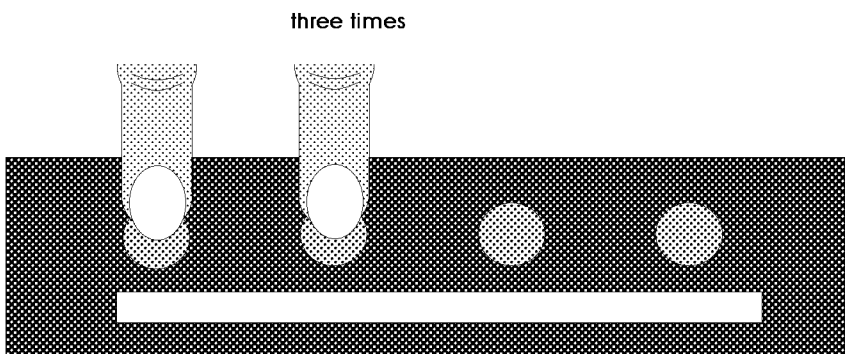
### 2.4 Sound-Init

Before programming a new sound, it is a good idea to recall the Sound-Init program so you can start from scratch. To recall the Sound-Init program, follow these steps:

- Press and hold the [Shift button].



- While holding the [Shift button], press the [Store button] *three times*.



Now release the [Shift button]. The display will read:

Datatransf. Mode  
Init Sound: A01?

- The Sound-Init program will automatically be placed in the currently selected location.
- To acknowledge the recall, press the [OK button], also known as the [parameter/value button].

## 2.4 Sound – Init

The display will momentarily read:



Datatransfer  
completed

You have now initiated the sound init program. For a listing of the default settings of all Sound-program parameters, refer to ch. 7 in this Programming manual.

An init-patch does not delete the sound-program currently selected, since it resides only in an edit-buffer. Therefore, if you want to keep the new sound, you must manually store it. On the other hand you can always recall the programm originally residing in that storage-location by doing a recall. Great if your 4-year-old turned that ringing pad to a singing pet.





# Chapter 3

Audio modules

**3.1 Oscillators 1 & 2**

**3.2 Waves 1 & 2**

**3.3 Volume**

**3.4 Filter**

**3.5 Panning**



## 3. Audio modules

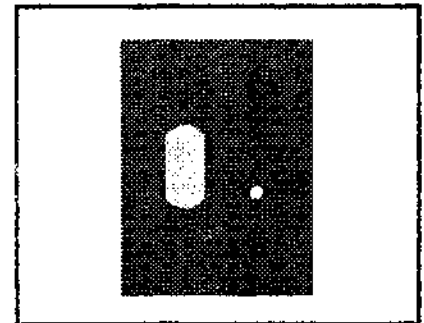
## 3.0 Audio modules

This chapter covers all aspects of programming the MicroWave's audio modules. By now, you should be familiar with the basic user-interface structure as well as the sound-architecture of the MicroWave.

Each of the audio modules is accessed on a separate MicroWave page.

To edit a Sound-program, you first must select Sound-edit mode.

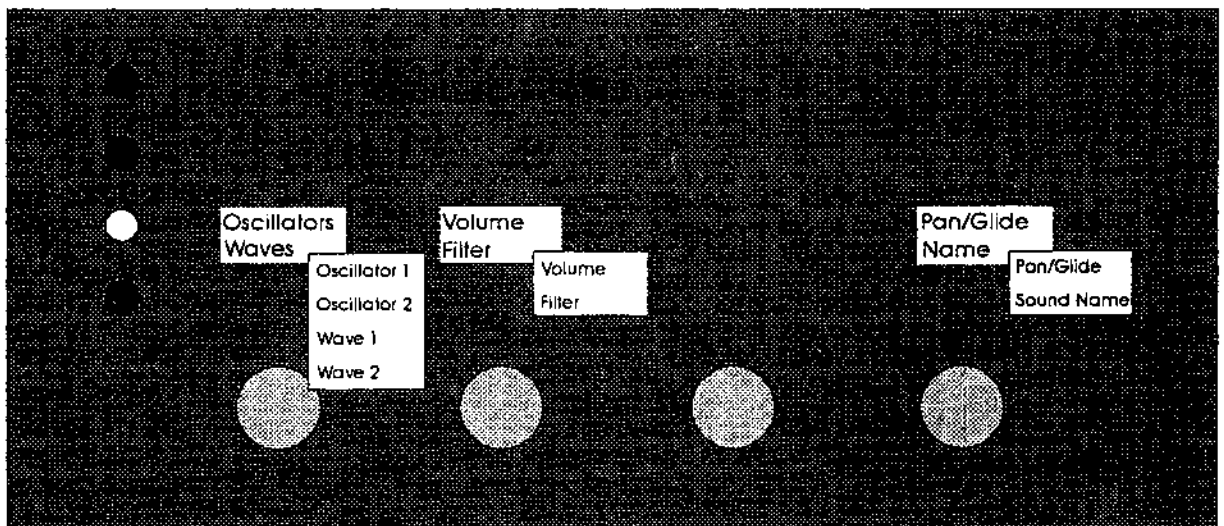
- Use the [mode button] to switch to Sound-edit mode.
- You will find all audio modules on pages to be selected with the three [select buttons] labeled <Oscillators/Waves>, <Volume/Filter> and <Pan/Glide/Name>



Audio modules

Listed below are the pages of the audio modules of *Sound-edit mode*:

- Oscillator 1
- Oscillator 2
- Wave 1
- Wave 2
- Volume
- Filter
- Pan/Glide



The above figure illustrates where to find the pages of the audio modules.

## 3.1 Oscillators 1 & 2

### 3.1 Oscillators 1 & 2

MicroWave Oscillators are merely a *pitch source*. They do not output any sound of their own, but drive the Waves, which in turn will produce the desired spectrum.

Both oscillators are completely identical in structure. They contain the same parameters except for one: Oscillator 2 has an additional **Link** parameter that disables it's own modulation parameters and replaces them with those of oscillator 1. See below for details.

To access oscillators 1&2:

- Select Sound-edit mode using the [mode button].
- Press the [select button] labeled <Oscillators/Waves> until the page-identification field of the display reads:

```
Oscillator 1  I
Octave:      +0
```

You have now selected the oscillator 1 page.

- To select oscillator 2, simply press the [select button] <Oscillators/Waves> once more. Now the display's page-identification field should read:

```
Oscillator 2  I
Octave:      +0
```

- To select oscillator 1 again, press the [select button] three times to cycle back to it's page.

### 3) Oscillators 1 & 2

**Octave** (-2..+2) sets the octave range in which the oscillator's pitch will be.

0 will yield 'equal' pitch. If the key C4 is struck, the oscillator's pitch will be C4 (that is, without any modulations or tuning-tables).

-2 will transpose the pitch downwards two octaves.

+2 will transpose the pitch upward by two octaves.

**Semitone** (0..12) lets you transpose the oscillator's pitch in semitones. You can transpose upwards only.

0 won't change the pitch according to the received MIDI note.

12 will output the oscillator's pitch an octave higher.

**Detune** (-64..+63) lets you fine tune each oscillator separately.

▣ Be aware that you can detune both oscillators independently. This might come in handy for creating lush chorus sounds; detune one oscillator upwards, the other downwards by the same amount. Thus the perceived pitch will not drift up or down but will be perceived as being in tune with other sounds.

-64 will detune the oscillator a semitone downwards.

+63 will detune the oscillator upwards by a semitone.

```
Oscillator 1  I
Octave:      +0
```

```
Oscillator 1  I
Semitone:     00
```

```
Oscillator 1  I
Detune:       +00
```

## 3.1 Oscillator: 1 & 2

**Bend Range** (*O..12*) sets the pitch bend range for this oscillator in semitones.

- ▣ Be aware that the bend range may be different for each oscillator, allowing you to achieve guitar-like bendings "of one string only" or other effects.
- ▣ If pitch-bend reception is disabled in the global MIDI- or Multi master-parameters this parameter will not show any effect. Consult the Performance manual for details.

*O* will disengage any pitch-bend reception.

*12* will bend the oscillator up to an octave in either direction depending of the pitch-bend usage.

**Pitchmode** (*normal/fixed*) defines if incoming MIDI note-messages will change the MicroWave's pitch or not.

*normal* will change the oscillator's pitch as defined by the incoming note-on messages, as you would usually expect.

*fixed* disconnects the MIDI note-ons from the oscillators, resulting in no pitch-change whatsoever no matter which note-on was received. The note-on message will still trigger the oscillator; only the incoming pitch will be ignored.

- ▣ Use *fixed Pitchmode* for percussion or effects sound. Use keytracking at the filter- and volume-modules to change timbre and loudness when you play different keys, but not the pitch. Set the desired pitch using the Octave/Semitone parameter. If both of them are set to *O*, all keys will play C4.

Next you will find the parameters for two routeable modulation inputs. Modulator 1 is a sidechain-modulator, while Modulator 2 is of the regular kind.

```
Oscillator 1  I
Bend Range:  02
```

```
Oscillator 1  I
Pitchmode: normal
```

## 3.1 Oscillators 1 & 2

**Modulator1 Source** (*modifier modules table*) defines the source-modifier-module of modulator 1 to alter the pitch. In the display-example, *LFO 1* would control the pitch of oscillator 1.

```
Oscillator 1 I
Mod1 Src.:LFO 1
```

**Modulator1 Control** (*modifier modules table*) defines the control-modifier-module to scale the source-modifier-module of modulator 1. In the display example, the controller keyboard's modulation wheel would control how much LFO 1 will affect the pitch of oscillator 1.

```
Oscillator 1 I
Mod1 Ctrl:ModWhl
```

**Modulator1 Amount** (*-64..+63*) sets the maximum possible amount of modulation for sidechain-modulator 1.

```
Oscillator 1 I
Mod1 Amnt: +20
```

-64 inverts the source's output signal and applies the full modulation amount.

+63 applies the source as it is at full amount.

□ In any case the actual amount of modulation will be determined by the control input. If the modifier-module connected to the control input is outputting its *full amount*, the source-modifier-module will modulate the pitch as set by the amount parameter. If the control-modifier-module outputs *nothing*, there will be no modulation of the pitch at all. The amount parameter only determines the *maximum possible value* of modulation; the actual value is set in realtime by the control-modifier-module.

Some possible applications:

- ▶ Connect an LFO to the source and the mod-wheel to the control input to adjust vibrato with the mod-wheel.
- ▶ Connect an envelope at the source and velocity to the control input for velocity-sensitive pitch envelopes.
- ▶ Connect an LFO at the source and an envelope to the control input to achieve envelope-controlled vibrato effects.

## 3.1 Oscillators 1 & 2

**Modulator2 Source** (*modifier modules table*) defines the source-modifier-module of modulator 2 to alter the pitch.

```
Oscillator 1 I
Mod2 Src.:LFO 1
```

**Modulator2 Amount** (-64..+63) sets the maximum possible amount of modulation for the regular modulator 2.

```
Oscillator 1 I
Mod2 Amt: +00
```

-64 inverts the source's output signal and applies the full modulation amount.

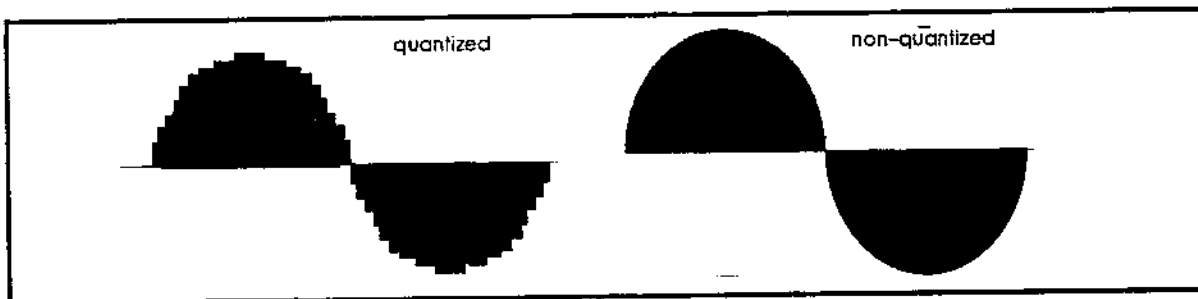
+63 applies the source as it is at full amount.

**Modulator2 Quantize** (*off..7*) gradually transforms continuous modulations like an LFO sine-wave into discreet steps.

```
Oscillator 1 I
Mod2 Qnt.: off
```

*off* does not introduce any quantization to the original signal. The modifier-module's output curve will remain unaltered.

*1..7* introduces different levels of quantization, thus changing the modifier-module's output from a continuous waveform to a quantized wave of discreet steps. See the figure below to understand the relation:



- ▶ use the quantization-value together with an LFO as source to achieve sample-and-hold effects.

Some possible applications for modulator 2:

- ▶ Use velocity as a source to alter the pitch according to how hard you strike a key; good for percussion-imitation.



## 3 | Oscillators 1 & 2

- ▶ Use keytracking on one oscillator with a slight amount only for pitch-dependent detunings. Use the detune parameter of that oscillator to shift the major detuning either towards the bass- or treble-region respectively.
- ▶ Use LFO 1 for vibrato. Program LFO 1's level modulation input to be scaled by the mod-wheel to control vibrato-depth with the mod-wheel.
- ▶ Use the pitch-bender as source and apply some quantizing to achieve guitar-fretboard like glissandi.

This last parameter is available for oscillator 2 only:

**Link Oscillator Modulation** (*off/on*) determines if both oscillators will use the exact same modulation-settings or if they will use their individually programmed settings



Oscillator 2 I  
Link Osc Mod:off

*off* will not link the modulators. Each oscillator will be modulated according to it's own parameter settings.

*on* will link the modulators of both oscillators. Each of the following parameters of oscillator 2 will be ignored and replaced by the same parameters of oscillator 1:

- ▶ Bend Range
- ▶ Modulator 1 Source
- ▶ Modulator 1 Control
- ▶ Modulator 1 Amount
- ▶ Modulator 2 Source
- ▶ Modulator 2 Amount
- ▶ Modulator 2 Quantize

Whenever you switch off the Link setting to separate the modulation parameters of both oscillators again, the last programmed parameter values of oscillator 2 will be valid again. Therefore, if you want to check if a global modulation setting works better than an individual one, simply toggle the Link setting on and off to find out what suits your needs best.

### 3.2 Waves 1 & 2

Waves produce the *basic spectrum* used to develop a timbre with the MicroWave. They do not possess any pitch-related parameters of their own, but rather are directly linked to their respective oscillators that define the Waves' pitch.

Both Waves are almost identical in design. They contain the same parameter-sets except for two:

- ▶ Only a single *Wavetable* can be used for each Sound-program; therefore only Wave 1 has a parameter to select the Wavetable.
- ▶ Wave 2 has an additional *Link* parameter that disables it's own modulation parameters and replaces them with those of Wave 1.

Remember that each Wave is driven by it's respective oscillator to output the waveform at the desired pitch. Therefore, if you want to change the pitch of a Wave you must do that at the according oscillator. Wave 1 is driven by oscillator 1, Wave 2 by oscillator 2.

To access Waves 1 & 2:

- Select Sound-edit mode using the [mode button]. If you've been following along with the manual, you should already be there.
- Press the [select button] labeled <Oscillators/Waves> until the page-identification field of the display reads:

```
Wave 1      I
Wavetable:  R01
```

You have now selected the Wave 1 page.

- To select Wave 2, simply press the [select button] <Oscillators/Waves> once more. Now the display's page-identification field should read:
- To select Wave 1 again, press the [select button] three times to cycle back to it's page.

```
Wave 2      I
Startwave:  00
```



## 32 Waves 1 & 2

**Wavetable** (*ROI..R32/I33..I34/C45..C56*) selects the Wavetable. It is valid for both Waves 1 and 2; though each Wave can address different Startwaves and modulations.

Wave 1	I
Wavetable:	R01

Since this parameter is used by both Waves, only Wave 1 will display it.

Each Wavetable is referred to by a number. If you happen to belong to the chosen few that have worked with the PPG 2.2 or 2.3 synthesizer, you will notice that the Wavetables are exactly the same. The numbering, however, is slightly different: A MicroWave Wavetable index succeeds the PPG index by 1. Also, there are 2 new wavetables in ROM: R31 and R32.

Each Wavetable has a characteristic of its own. Find out that characteristic by gradually listening to the Waveforms of each wavetable. Do it either manually or using a slow envelope.

*ROI..R32* are factory wavetables that reside in ROM and cannot be altered in any way by the user.

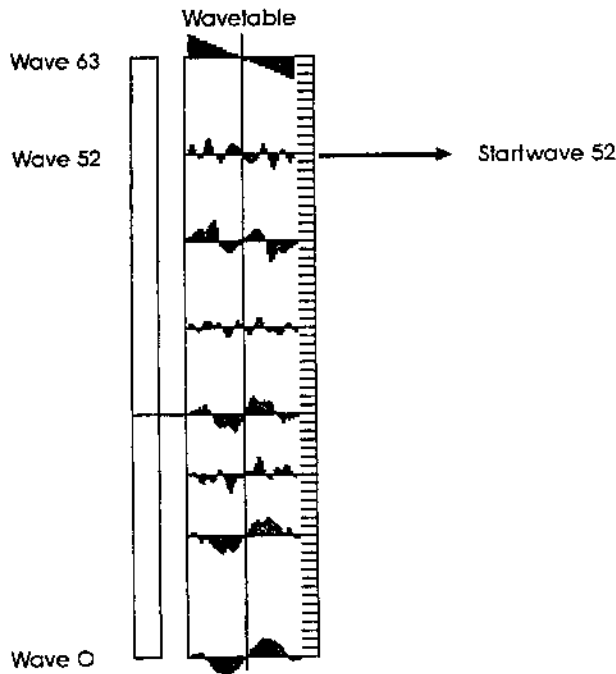
*I34..I44* are 12 possible user-wavetables that reside in the MicroWave's internal memory. These Wavetables are empty when the MicroWave leaves the factory and can only be accessed by special software that must run on a host computer.

*C45..C56* are 12 possible user-wavetables residing on an expansion card. Depending on the card, there may or may not be some wavetables present. However, just as with internal user-wavetables, you need additional software running on a host computer to create user-wavetables.

**Startwave** (*OO..60/tri/sqr/saw*) selects the first Waveform of the Wavetable to be played by the Wave-module. If you do not use any modulation at all to scan the Wavetable, this will be the waveform and spectrum the Wave-Module will produce.

Wave 1	I
Startwave:	00

The below figure should point out what a startwave is:



The Startwave defines the position within the Wavetable that the Wave-module will output first. This waveform may remain static, resulting in an uniform spectrum; or it may be altered using one or more of each Wave-modules' modulation-inputs, resulting in a dynamically varying spectrum.

Keep in mind that each Wave-module may use a different Startwave.

*OO.60* selects one of the unique waveforms of each Wavetable as a Startwave. These Startwaves will vary drastically, depending on the Wavetable selected.

*tri* selects a triangular waveform. It will be the same, no matter which Wavetable you have chosen. This way you may program a Wave to remain static using a regular synthesizer waveform for creating traditional analogue sounds.

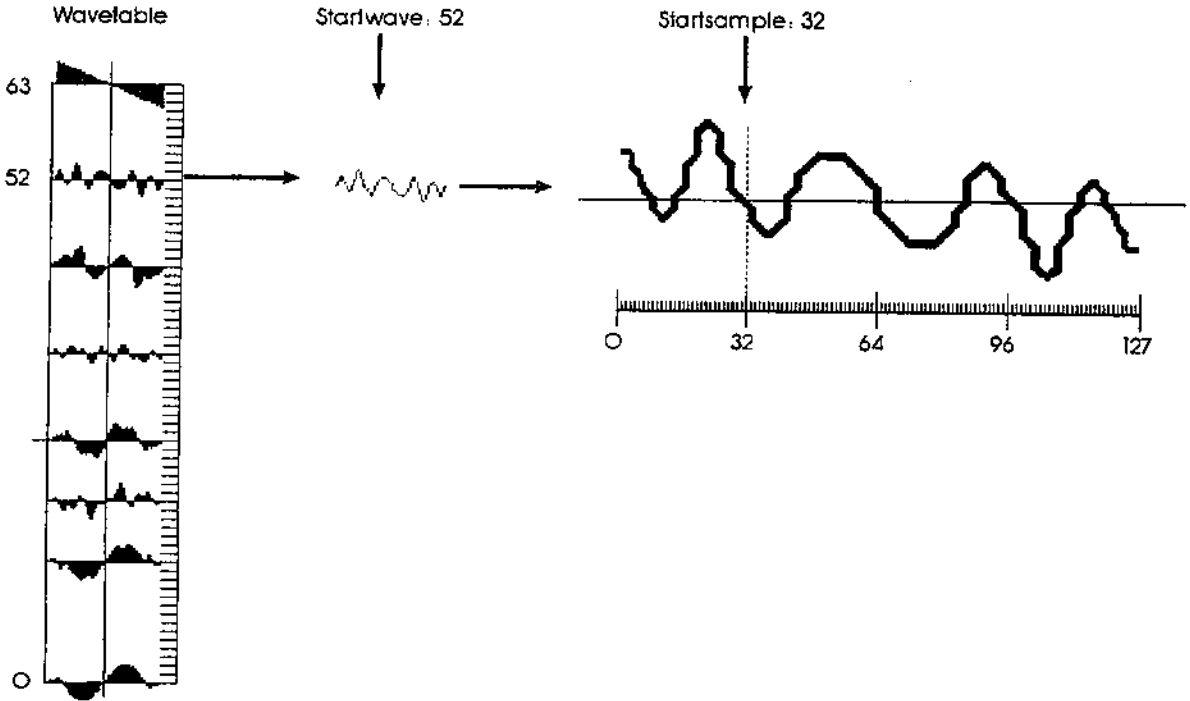
*square* selects a squarewave with 50% duty-cycle. It will be the same, no matter which Wavetable you have chosen.

*saw* selects a sawtooth waveform. It will be the same, no matter which Wavetable you have chosen.

# 32 Waves 1 & 2

**Startsample (free/1..127)** defines the exact sample within a Startwave at which the Wave-module will start playing. Look at the below figure to understand the relation of a Wavetable, a Startwave and the Startsample.

Wave 1 I  
Startsample:001



As you can see, with each parameter you further focus in on the exact location from where a waveform will begin.

Startsample is especially useful if both Waves are set to a value different than 'free'. Then you can determine a phase-relationship between the two Wave-modules that multiplies the actual available waveforms within a wavetable. By setting one Wave to a Startsample value of 1 and altering the other Wave's Startwave parameter, notches and peaks will be produced not inherent in one waveform alone. Try this by selecting the same Startwave and disabling all modulations for each Wavemodule.

*free* selects a Startsample value at random. Use this setting to slightly change the attack-sound for each successive keystroke. This works best if one Wave-module's Startsample parameter is set to a fixed value while the other is set to *free*.

Audio modules



1..63 selects a Startsample value of the positive half of a waveform-cycle. Usually you will get more peaks than notches if both Wave's are set to such a Startsample value.

64..127 selects a Startsample value of the negative half of a waveform-cycle. Usually this results in more notches if one Wave is set to a value in the positive, the other in the negative waveform-cycle. Be aware that, whenever you use similar Startwaves and the difference between the Startsample-values of both Waves is around 64, the sound might be very soft with the fundamental as well as certain harmonic components being close to zero due to phase-cancellation.

► Use the same Startwave for both Wave-modules with slightly different Startsamples. Don't use any modulation. Slightly detune one oscillator to achieve phasing effects that you can control by programming the Startsample parameter accordingly. Try a very slight LFO modulation instead of detuning one oscillator.

**Envelope Amount** (-64..+63) sets the basic amount of modulation for the Wave-envelope. This is a preconfigured modifier-module which you cannot change, but can only disable by setting it to 0. The value programmed here will always be applied, no matter which velocity is received. Envelope modulation results in a dynamic wavescan process.

Wave 1	I
Envel. Amnt:	+00

-64 inverts the Wave-envelope's output signal and applies the full modulation amount.

+63 applies the Wave-envelope's output as it is at full amount.

0 disables any constant (not velocity dependent) Wave-envelope modulation of the Wave module.

## 3.2 Waves 1 & 2

**Envelope Velocity** (-64..+63) sets the velocity-amount of the Wave-envelope modulating the Wave. This is a preconfigured modifier-module which you cannot change, but can only disable by setting it to 0. The value programmed here applies in direct proportion to the received velocity-value.

-64 inverts the Wave-envelope's output signal and applies the modulation proportional to the received velocity. A velocity-value of 127 will apply the full amount set here.

+63 applies the Wave-envelope's output as it is in direct proportion to the received velocity. A velocity-value of 127 will apply the full amount set here.

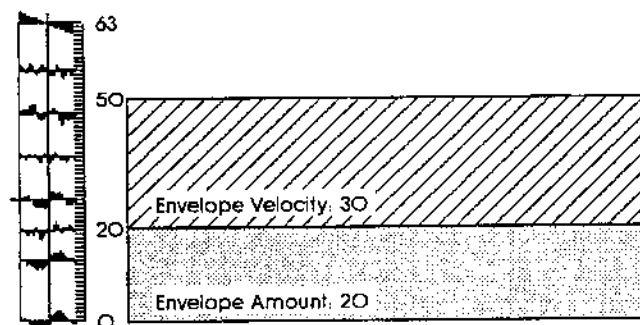
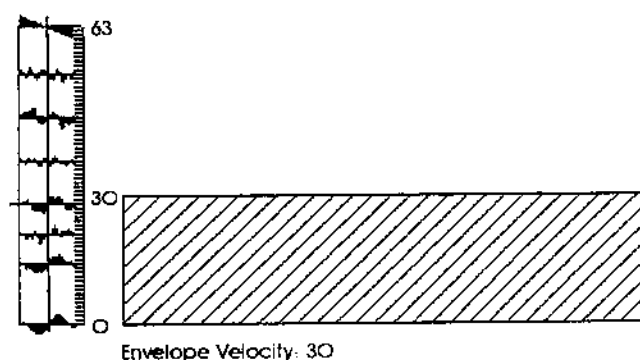
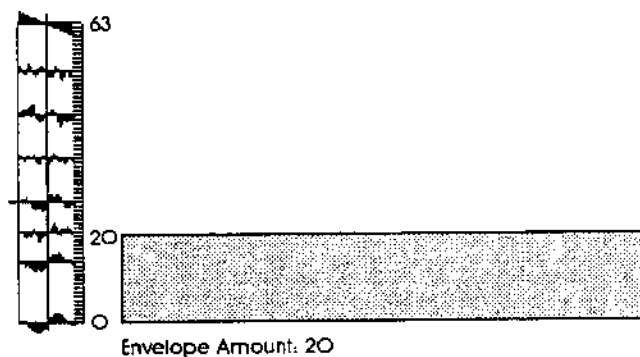
0 disables any velocity dependent Wave-envelope modulation of the Wave module.



Wave 1 I  
Envel.Vlcty: +00

## 3.2 Waves 1 & 2

- Look at the below graph to understand how the Envelope Amount and Envelope Velocity parameters do interact:



max. Envelope peak: Envelope Amount + Envelope Velocity = 50

The value set as *Envelope Amount* will always apply, no matter which velocity is received. The value set at *Envelope Velocity* will apply also, but will be scaled according to the velocity received. Both values are added before they are applied to the module. You might think of the two parameters to be similar to a compressor: At low velocities, the entire modulation of both Envelope parameters will be at least as strong as set by the *Amount* parameter. The higher the velocity is, the more the value set at the *Velocity* parameter determines the modulation.



However, the maximum sum of both parameters can never surpass +63 or -64, which also is the maximum range of each parameter alone. When both parameter values are added, the Amount-value will be prominent over the Velocity-value. Therefore, if you set the Amount-value to +63, no velocity whatsoever will raise that Amount, resulting in a non-velocity-sensing modulation. If however, you set the Velocity amount to a negative value, the sum of both parameters will decrease according to the incoming velocity, resulting in a reverse velocity-sensitivity where the modulation becomes less if you strike a key harder.

**Keytrack Amount** (-64,+63) sets the amount of modulation according to the keyboard position of the received MIDI note. This is a preconfigured modifier-module which you cannot change, but can only disable by setting it to 0. The greater this value is, the more the keyboard position will alter the Wave-module.

The center key is C4. No matter which value you set at Keyboard Amount, this key will remain unaltered.

By employing keytracking to the Wave-module, you will change the Startwave value according to the MIDI note and the Keytrack Amount.

-64 inverts the received note number and applies it at full value to the Wave-module. This results in a decreasing change of the Startwave for each successive pitch on the keyboard, or each received MIDI note number.

+63 applies the received key number as it is at full value to the Wave-module, resulting in an increasing change of the Startwave value for each successive pitch on the keyboard.

All other values will scale the incoming MIDI note numbers accordingly, resulting in fewer changes per octave.

0 will disable any keytracking modulation from the Wave-module.

Wave 1	I
Keytrk.Amnt:	+00

Next you will find the parameters for two routeable modulation inputs. Modulator 1 is a sidechain-modulator, while Modulator 2 is of the regular kind. All modulation will result in a dynamically changing waveform due to the wavescan technology.

**Modulator1 Source** (*modifier modules table*) defines the source-modifier-module of modulator 1 to alter the Wave.

```
Wave 1      I
Mod1 Src.:LFO 1
```

**Modulator1 Control** (*modifier modules table*) defines the control-modifier-module to scale the source-modifier-module of modulator 1.

```
Wave 1      I
Mod1 Ctrl:ModWhl
```

**Modulator1 Amount** (-64..+63) sets the maximum possible amount of modulation for sidechain-modulator 1.

```
Wave 1      I
Mod1 Amnt:  +00
```

-64 inverts the source's output signal and applies the full modulation amount.

+63 applies the source as it is at full amount.

❑ In any case the actual amount of modulation will be determined by the control input. If the modifier-module connected to the control input is outputting it's full amount, the source-modifier-module will modulate the Wave as set by the amount parameter. If the control-modifier-module outputs nothing, there will be no modulation of the Wave at all. The amount parameter only determines the *maximum possible* value of modulation; the *actual* value is set in realtime by the control-modifier-module.

Some possible applications:

- Connect an envelope at the source and the mod-wheel to the control input to adjust the envelope-amount with the mod-wheel.
- Connect an LFO to the source and velocity to the control input for velocity-sensitive LFO-changes.
- Connect an LFO at the source and an envelope to the control input to achieve envelope-controlled LFO-effects.

## 3.2 Waves 1 & 2

**Modulator2 Source** (*modifier modules table*) defines the source-modifier-module of modulator 2 to alter the Wave.

```
Wave 1      I
Mod2 Src.:LFO 1
```

**Modulator2 Amount** (*-64..+63*) sets the maximum possible amount of modulation for the regular modulator 2.

```
Wave 1      I
Mod2 Amnt:  +00
```

-64 inverts the source's output signal and applies the full modulation amount.

+63 applies the source as it is at full amount.

Some possible applications:

- ▶ Use velocity as a source to alter the Startwave according to how hard you strike a key. That's a good alternative to opening the filter using velocity.
- ▶ Use keytracking on only one Wave for constantly changing overtones of one basic waveform.
- ▶ Use LFO 1 with a slight amount for gentle phasing-like effects.

**Wave Mode** (*stepped/smooth*) determines the interpolation mode of the wavescan-process.

```
Wave 1      I
WaveMode:smooth
```

As long as you do not apply any dynamic wavescan techniques such as using envelopes, LFO's or continuous MIDI controllers, you won't notice any difference no matter how you set this parameter.

However, if you do use continuous changing modifiers to scan the Wavetable, this parameter is in effect.

*stepped* uses a hard interpolation algorithm that produces noticeable pops and clicks, whose energies depend on the Wavetable being used. If you strive for a raw, aggressive sound, this mode might be just what you're after.

## 3.2 Waves 1 & 2

*smooth* uses a soft interpolation algorithm that tries to change the waveform as smoothly as possible for a true gradual change in timbre. If using rapidly changing modulation sources with great amounts, you may nevertheless notice certain rough changes. These are due to very fast amplitude changes of complex harmonics, bearing physical reasons even the MicroWave must obey.

The last parameter is available for Wave 2 only:

**Link Wave Modulation** (*off/on*) determines if both Wave-modules will use the exact same modulation-settings or if they will use their individually programmed settings.

Wave 2	I
Link WaveMd:	off

*off* will not link the modulators. Each Wave-module will be modulated according to it's own parameter settings.

*on* will link the modulators of both Waves. Each of the following parameters of Wave 2 will be ignored and replaced by the same parameters of Wave 1:

- ▶ Modulator 1 Source
- ▶ Modulator 1 Control
- ▶ Modulator 1 Amount
- ▶ Modulator 2 Source
- ▶ Modulator 2 Amount

Whenever you switch off the Link setting to separate the modulation parameters of both Wave-modules again, the last programmed parameter values of Wave 2 will be valid again. Therefore, if you want to check if a global modulation setting works better than an individual one, simply toggle the Link setting on and off to find out what suits your needs best.

## 3.3 Volume

### 3.3 Volume

The Volume-module fulfills two functions: it sets the levels of each Wave, the noise-source and the Sound-program in it's mixer division. On the other hand, it puts all modulation inputs for changing a sound's loudness at your fingertips.


To get access to the Volume-module:

- ⊕ Select Sound-edit mode using the [mode button]. If you've been following along with the manual you should already be there.
- ⊕ Press the [select button] labeled <Volume/Filter> until the page-identification field of the display reads:

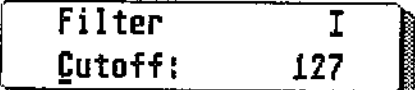
You have now selected the Volume page.

- ⊕ If instead the display's page-identification field reads:

...press the [select button] once more to cycle back to the Volume page.



Volume	I
Wave1 Vol.:	4



Filter	I
Cutoff:	127

## 3.3 Volume

### A: Mixer Functions

**Wave1 Volume** (0..7) adjusts the output-level of Wave 1.

0 mutes the Wave output.

7 outputs the Wave at full volume.

**Wave2 Volume**(0..7)adjusts the output-level of Wave 2.

0 mutes the Wave output.

7 outputs the Wave at full volume.

**Noise Volume** (0..7) adjusts the output-level of the noise-source.

0 mutes the noise-source's output.

7 outputs the noise at full volume.

To prevent distortion, the sum of all mixer-inputs should not exceed a value of 8 . Therefore, if you only use a single Wave, set it to 7 to enjoy full gain. If you combine both Waves at equal volume, set each one to a value of 4 to prevent distortion. This basic rule, of course, is very much dependent on the actual waveform produced by the Wavetable. Certain waveforms can be set to higher volumes than others.

On the other hand, you can achieve very interesting and powerful distortion-sounds by explicitly overloading the mixer. Set the output of one Wave to a value of 7, the other output between 5 and 7. If you strive for a power guitar and need only a limited range, adding some noise might produce an even more convincing result, depending on the range and the filter setting. Don't use detuning on any of the oscillators, since it will introduce level-changes that will alter the distortion, canceling it out now and then, resulting in a sound that switches from clean to distorted at the rate of the detuning-value.

Volume	I
Wave1 Vol.:	4

Volume	I
Wave2 Vol.:	4

Volume	I
Noise Vol.:	0

## 3.3 Volume

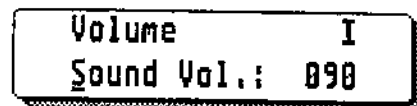
**Sound Volume** (*O..127*) sets the overall output-level of the entire sound. It represents the summing-fader of the mixer.

▣ Sound Volume will always be scaled by the Master Volume. Therefore, if Master Volume is set to 0, there will be no output whatsoever.

0 mutes the entire Sound-program.

*127* outputs the Sound-program at full volume. Still, this is dependent on many factors of a sound.

Use Sound Volume to adjust all your MicroWave Sound-programs to output at approximately the same level. Use Master Volume to adjust the entire output of the MicroWave.



### B. Loudness modulation

**Envelope Amount** (-64..+63) sets the basic amount of envelope-modulation for the loudness. This is a preconfigured modifier-module which you cannot change, but disable by setting it to 0. The value programmed here will always be applied, no matter which velocity is received. Envelope modulation will shape the basic loudness-curve of a sound.

Volume	I
Envel.Amnt:	+63

-64 inverts the Volume-envelope's output signal and applies the full modulation amount.

+63 applies the Volume-envelope's output as it is at full amount.

0 disables any constant (not velocity dependent) Volume-envelope modulation of the Volume module.

**Envelope Velocity** (-64..+63) sets the velocity-amount of the Volume-envelope modulating the loudness. This is a preconfigured modifier-module which you cannot change, but can only disable by setting it to 0. The value programmed here applies in direct proportion to the received velocity-value and will alter the loudness of a sound.

Volume	I
Envel.Vlcty:	+88

-64 inverts the Volume-envelope's output signal and applies the modulation proportional to the received velocity. A velocity-value of 127 will apply the full amount set here, resulting in a very soft volume since the envelope is inverted.

+63 applies the Volume-envelope's output as it is in direct proportion to the received velocity. A velocity-value of 127 will apply the full amount set here, resulting in maximum loudness.

0 disables any velocity dependent Volume-envelope modulation of the loudness.



## 3.3 Volume

As pointed out before, the value set as Envelope Amount will always apply, no matter which velocity is received. The value set at Envelope Velocity will apply also, but will be scaled according to the velocity received. Both values are added before they are applied to the module. You might think of the two parameters to be similar to a compressor: At low velocities, the entire modulation of both Envelope parameters will be at least as strong as set by the Amount parameter. The higher the velocity is, the more the value set at the Velocity parameter determines the modulation.

However, keep in mind that the maximum sum of both parameters can never surpass +63 or -64, which also is the maximum range of each parameter alone.

- You must set a value different from 0 at any one or both Envelope-parameters in order to hear a sound at all. If you want to use a different envelope than the Volume envelope, select it at one of the routeable modulation inputs. Set the sustain level of the Volume envelope to maximum and use an Envelope Amount of +63. Since the Volume Envelope Amount will actually be scaled according to all other modulation inputs, this procedure simply provides a basic setting for applying modulation at all. The actual loudness will be set by the other modulation inputs.

**Keytrack Amount** (-64..+63) sets the amount of modulation according to the keyboard position or received MIDI note. This is a preconfigured modifier-module which you cannot change, but can only disable by setting it to 0. The greater this value is set, the more the keyboard position will alter the loudness.

The center key is C4. No matter which value you set at Keyboard Amount, this key will remain unaltered.

By employing keytracking to the Volume-module, you will change the volume according to the received MIDI note and the Keytrack Amount.

-64 inverts the received key number and applies it at full value to the Volume-module. This results in decreasing volume for each successive pitch on the keyboard, or each received MIDI note number.



Volume I  
Keytrk Amnt: +88

## 3.3 Volume

+63 applies the received key number as it is at full value to the Volume-module, resulting in increasing the volume for each successive pitch on the keyboard.

All other values will scale the incoming MIDI note numbers accordingly, resulting in fewer changes per octave.

0 will disable any keytracking modulation from the Volume-module.

Next you will find the parameters for two routeable modulation inputs. Modulator 1 is a side-chain-modulator, while Modulator 2 is of the regular kind. All modulation will result in a dynamic change of the sound's loudness.

**Modulator1 Source** (*modifier modules table*) defines the source-modifier-module of modulator 1 to alter the Volume-module.

**Modulator1 Control** (*modifier modules table*) defines the control-modifier-module to scale the source-modifier-module of modulator 1.

**Modulator1 Amount** (-64..+63) sets the maximum possible amount of modulation for sidechain-modulator 1.

-64 inverts the source's output signal and applies it at full modulation amount.

+63 applies the source as it is at full amount.

■ In any case the actual amount of modulation will be determined by the control input. If the modifier-module connected to the control input is outputting it's full amount, the source-modifier-module will modulate the Volume as set by the amount parameter.

If the control-modifier-module outputs nothing, there will be no modulation of the Volume-module at all. The Amount parameter only determines the maximum possible value of modulation; the actual value is set in realtime by the control-modifier-module.

```
Volume      I
Modi Src.:LFO 1
```

```
Volume      I
Modi Ctrl:ModWh1
```

```
Volume      I
Modi Amnt:  +88
```

## 3.3 Volume

Some possible applications:

- ▶ Connect an LFO to the source and the mod-wheel to the control input to adjust tremolo with the mod-wheel.
- ▶ Connect the Wave-envelope to the source and velocity to the control input to create a complex velocity-sensitive loudness-contour.
- ▶ Connect an LFO at the source and an envelope to the control input to achieve envelope-controlled tremolo-effects.

**Modulator2 Source** (*modifier modules table*) defines the source-modifier-module of modulator 2 to alter the Volume-module.

**Modulator2 Amount** (*-64..+63*) sets the maximum possible amount of modulation for the regular modulator 2.

-64 inverts the source's output signal and applies it at the full modulation amount.

+63 applies the source as it is at full amount.

Some possible applications:

- ▶ Use keytracking as the source to scale the loudness according to the key-range. Create two sounds of which one uses positive, the other negative Amount-values and layer them in Multi mode to achieve positional crossfade sounds.
- ▶ Use the mod-wheel as source to fade sounds in or out using the mod-wheel. If using negative amounts, -32 will fade out the sound completely at the maximum mod-wheel position.



Volume I  
Mod2 Src. LFO 1



Volume I  
Mod2 Amnt: +08

### **C: Special MIDI-Implementation**

The Volume-module has a permanently assigned MIDI controller that is active without the need to program it as a modifier at a modulation-input:

**MIDI Volume Controller** (Controller #7) is permanently assigned to adjust the Sound-program's loudness.

If you want to disable this controller, you can do so in Global mode using the appropriate MIDI filter. If, however, you have assigned that controller anywhere else, it will be disabled there, too.

If using Multi mode, you may individually control the volume of each instrument as long as the MIDI channels differ; otherwise, all instruments that use the same MIDI channel will be altered simultaneously.


## 3.4 Filter

### 3.4 Filter

The Filter-module controls the overall harmonic content of a Sound-program. It is a 4-pole, low-pass analog filter that employs a 24dB/Octave slope, and can give your sound even more character. It's response varies with the Wavetable used and it's inherent waveforms. The less overtones a waveform has, the less the Filter can change the spectrum. Since the Filter is positioned after the summing-stage of the mixer, it applies to both Waves and the noise-source equally.

To get access to the Filter:

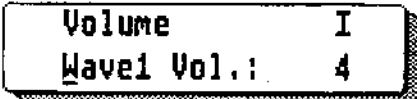
- Select Sound-edit mode using the [mode button]. If you've been following along with the manual, you should already be there.
- Press the [select button] labeled <Volume/Filter> until the page-identification field of the display reads:



```
Filter      I
Cutoff:    127
```

You have now selected the Filter page.

- If instead the display's page-identification field reads:



```
Volume     I
Wave1 Vol.: 4
```

...press the [select button] once more to cycle back to the Filter page.

## 3.4 Filter

**Cutoff** (O..127) sets the frequency at which the Filter will begin to work. All frequencies below the cutoff-point pass unaltered, while frequencies above the cutoff-point will be dampened by 24dB per octave.

Filter	I
Cutoff:	127

0 completely closes the Filter, practically shutting off all incoming audio.

127 opens up the Filter completely, allowing all frequencies up to 20 kHz to pass through.

**Resonance** (O..127) adjusts the Q-factor of the Filter at the Cutoff frequency. The higher you set Resonance, the louder the frequencies around the cutoff-point will be, resulting in a more 'nasal' quality of the audio signal. At high Resonance values the Filter will begin to oscillate, producing a sine-wave of the frequency set at the Cutoff-parameter.

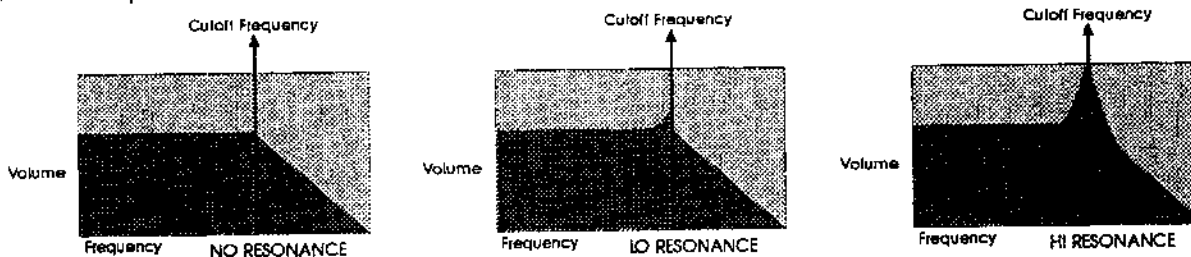
Filter	I
Resonance:	000

0 does not evoke any Resonance

127 evokes full resonance, self-oscillation begins around 80, relying very much on the Cutoff value and the pitch fed into the filter by the Waves.

If there is no input to the filter at all, self oscillation starts at 80. Use full Keytrack amount to play correct scales on the keyboard; use the Cutoff parameter to set the approximate pitch and fine tune with the Resonance parameter.

The graph below depicts the Cutoff and Resonance relationships:



As you can see, if Resonance is not used, the Filter's response is uniform up to the Cutoff-frequency. If a little Resonance is applied, there is a bandpass-like quality of the frequencies around the Cutoff. Finally, if Resonance is set to a high value, the Cutoff frequency is prominent over all other frequency, resulting in self-oscillation.



## 3.4 Filter

**Envelope Amount** (-64..+63) sets the basic amount of envelope-modulation for the Cutoff. This is a preconfigured modifier-module which you cannot change, but can only disable by setting it to 0. The value programmed here will always be applied, no matter which velocity is received. Envelope modulation will shape the Cutoff and thus the basic high-frequency content of a sound.

-64 inverts the Filter-envelope's output signal and applies it at the full modulation amount.

+63 applies the Filter-envelope's output as it is at full amount.

0 disables any constant (not velocity dependent) Filter-envelope modulation of the Cutoff.

**Envelope Velocity** (-64..+63) sets the velocity-amount of the Filter-envelope modulating the Cutoff. This is a preconfigured modifier-module which you cannot change, but can only disable by setting it to 0. The value programmed here applies in direct proportion to the received velocity-value and will alter the Cutoff, hence the high-frequency content, of a sound.

-64 inverts the Filter-envelope's output signal and applies the modulation proportional to the received velocity. A velocity-value of 127 will apply the full amount set here, resulting in sound becoming gradually dull since the envelope is inverted.

+63 applies the Filter-envelope's output as it is in direct proportion to the received velocity. A velocity-value of 127 will apply the full amount set here, resulting in maximum possible brightness.

0 disables any velocity dependent Filter-envelope modulation of the Cutoff.

The value set as Envelope Amount will always apply, no matter which velocity is received. The value set at Envelope Velocity will apply also, but will be scaled according to the velocity received. Both values are added before they are applied to the module. You might think of the two parameters to be similar to a compressor: At low velocities, the entire modulation of



```
Filter      I
Envel.Amnt: +00
```



```
Filter      I
Envel:Vlcty: +00
```

## 3.4 Filter

both Envelope parameters will be at least as strong as set by the Amount parameter. The higher the velocity is, the more the value set at the Velocity parameter determines the modulation.

However, the maximum sum of both parameters can never surpass +63 or -64, which also is the maximum range of each parameter alone. When both parameter values will be added, the Amount-value will be prominent over the Velocity-value. Therefore, if you set the Amount to +63, no velocity whatsoever will raise that Amount, resulting in a non velocity sensing modulation. If however, you set the Velocity amount to a negative value, the sum of both parameters will decrease according to the incoming velocity, resulting in a reverse velocity-sensitivity where the modulation becomes less if you strike a key harder.

**Keytrack Amount** (-64..+63) sets the amount of Cutoff-modulation according to the keyboard position or received MIDI note. This is a preconfigured modifier-module which you cannot change, but disable by setting it to 0. The greater this value is set, the more the keyboard position will alter the brightness.



Filter I  
Keytrk Amnt: +00

The center key is C4. No matter which value you set at Keyboard Amount, this key will remain unaltered.

By employing keytracking to the Filter, you will change the Cutoff, hence the brightness, according to the received MIDI note and the Keytrack Amount.

-64 inverts the received key number and applies it at full value to the Cutoff. This results in decreasing brightness for each successive pitch on the keyboard, or each received MIDI note number.

+63 applies the received key number as it is at full value to the Cutoff, resulting in an increasing brightness for each successive pitch on the keyboard.

All other values will scale the incoming MIDI note numbers accordingly, resulting in fewer changes per octave.

0 will disable any keytracking modulation from the Filter.



## 3.4 Filter

Next you will find the parameters for two routeable modulation inputs. Modulator 1 is a side-chain-modulator, while Modulator 2 is of the regular kind. All modulation will result in a dynamic change of the sound's brightness.

**Modulator1 Source** (*modifier modules table*) defines the source-modifier-module of modulator 1 to alter the Cutoff.

**Modulator1 Control** (*modifier modules table*) defines the control-modifier-module to scale the source-modifier-module of modulator 1.

**Modulator1 Amount** (*-64..+63*) sets the maximum possible amount of modulation for sidechain-modulator 1.

*-64* inverts the source's output signal and applies it at full modulation amount.


*+63* applies the source as it is at full amount.

▣ In any case the actual amount of modulation will be determined by the control input. If the modifier-module connected to the control input is outputting its *full amount*, the source-modifier-module will modulate the Cutoff as set by the amount parameter.

If the control-modifier-module outputs *nothing*, there will be no modulation of the Filter-module at all. The Amount parameter only determines the maximum possible value of modulation; the actual value is set in realtime by the control-modifier-module.

Some possible applications:

- ▶ Connect an LFO to the source and aftertouch to the control input to introduce a Filter-sweep with aftertouch.
- ▶ Connect the Wave-envelope to the source and velocity to the control input to create a complex velocity-sensitive Cutoff-contour.
- ▶ Connect an envelope at the source and the mod-wheel to the control input to achieve different Cutoff-envelopes according to the mod-wheel setting.



Filter I  
Mod1 Src.:LFO 1



Filter I  
Mod1 Ctrl:ModWh1



Filter I  
Mod1 Amnt: +00



## 3.4 Filter

**Modulator2 Source** (*modifier modules table*) defines the source-modifier-module of modulator 2 to alter the Cutoff.

**Modulator2 Amount** (-64..+63) sets the maximum possible amount of modulation for the regular modulator 2.

-64 inverts the source's output signal and applies it at the full modulation amount.

+63 applies the source as it is at full amount.

Some possible applications:

- ▶ Use keytracking as the source to scale the Cutoff-frequency according to the key-range. Set Resonance to a self-oscillating value and tune the Filter to an appropriate pitch.
- ▶ Use poly-pressure as source to alter the brightness of each note individually.

Yes, it can be done:

**Resonance Modulator Source** (*modifier modules table*) defines the source-modifier-module to modulate the filter's Resonance.

**Resonance Modulator Amount** (-64..+63) sets the maximum possible amount of modulation for the Resonance modulator.

-64 inverts the source's output signal and applies it at the full modulation amount.

+63 applies the source as it is at full amount.

```
Filter      I
Mod2 Src.: LFO 1
```

```
Filter      I
Mod2 Amt.:  +00
```

```
Filter      I
Rmod Src.: LFO 1
```

```
Filter      I
Rmod Amt.:  +00
```

## 3.4 Filter

Some possible Applications:

- ▶ Use the mod-wheel or poly-pressure to drive the Filter into self-oscillation to create guitar-like feedback effects. Don't use much envelope, otherwise the feedback's pitch will change.
- ▶ Use an envelope to change from a narrow, 'nasal' sound to a broader, warmer sound.
- ▶ If you use a high Resonance setting in conjunction with an envelope that modulates the Cutoff, apply that same envelope inverted to suppress that typical "high-Q-waow" produced by a sweeping band-pass.

### 3.5 Panning

The Panning-module defines the position of a sound in the stereo-base. Panning is, of course, applied individually to each voice, thus allowing stunning polyphonic panorama effects.

- If you have set the Stereo Width parameter of the Global parameters to mono, Panning will be disabled completely. Instead, a mono signal will appear of equal volume at both the Left and Right outputs.
- If you have programmed an individual output at the Instrument parameters of Multi mode, this Instrument will be withdrawn from the Left and Right output and will instead be outputted via the individual output assignment. Since the individual outputs are not stereophonic, the Panning-module will be inactive for this instrument if it is using individual outputs.

To access Panning:

- Select Sound-edit mode using the [mode button]. If you have been following along with the manual, you should already be there.
- Press the [select button] labeled <Pan/Glide/Name> until the page-identification field of the display reads:

```
Pan / Glide  I
Panning:    M00
```

You have now selected the Panning page.

- If instead the display's page-identification field reads:

```
Sound Name  I
Sound Init  +00
```

...press the [select button] once more to cycle back to the Panning page.

## 3.5 Panning

**Panning** (*L64..MOO..R63*) allows you to program the Sound-program's position in the stereo-field.

*L64* places the Sound-program all the way to the Left.

*MOO* places the Sound-program in the middle of the stereo-field.

*R63* places the Sound-program all the way to the Right.

Yes, this can be done, too:

**Panning Modulator Source** (*modifier modules table*) defines the source-modifier-module to modulate the Panning.

**Panning Modulator Amount** (*-64..+63*) sets the maximum possible amount of modulation for the Panning.

*-64* inverts the source's output signal and applies it at the full modulation amount.

*+63* applies the source as it is at full amount.

Some possible Applications:

- ▶ Use Keytracking to pan the Sound-program according to the keyboard-position. Set the Panning parameter to *MOO* and set a positive amount to the modulation-input.
- ▶ Use an LFO to achieve auto-panning effects. Use the Symmetry parameter of the LFO to achieve more adventurous panning effects.
- ▶ Use velocity or poly-pressure to individually pan each note you play to its own position according to how you play.

```
Pan / Glide  I
Panning:    M00
```

```
Pan / Glide  I
Pmod Src.:LFO 1
```

```
Pan / Glide  I
Pmod Amnt:  +00
```



### **Special MIDI-Implementation**

The Panning-module has a permanently assigned MIDI controller that is active without the need to program it as a modifier at a modulation-input:

**MIDI Panning Controller** (*Controller #10*) is permanently assigned to adjust the Sound-program's panning-position.

If you want to disable this controller, you can do so in Global mode using the appropriate MIDI filter. If, however, you have assigned that controller anywhere else, it will be disabled there, too.

If using Multi mode, you may individually control the panning of each Instrument as long as the MIDI channels differ; otherwise, all Instruments that bear the same MIDI channel will be altered simultaneously.

# Chapter 4

Modifier modules

**4.1 Volume Envelope**

**4.2 Filter Envelope**

**4.3 Wave Envelope**

**4.4 LFO 1**

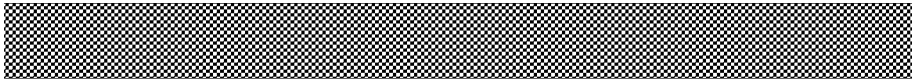
**4.5 LFO 2**

**4.6 Glide**

**4.7 Temperment**

**4.8 Name**







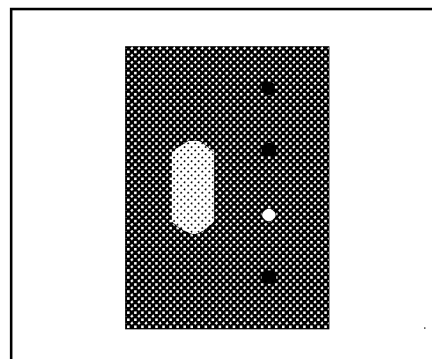
### Modifier Modules

This chapter explains all aspects of programming the MicroWave's modifier modules. By now, you should be familiar with the basic user-interface structure, the sound-architecture and the audio modules of the MicroWave.

As with the audio modules, each of the modifier modules is accessed on a separate page.

To edit a Sound-program, you must first select Sound-edit mode.

- Use the [mode button] to switch to Sound-edit mode.



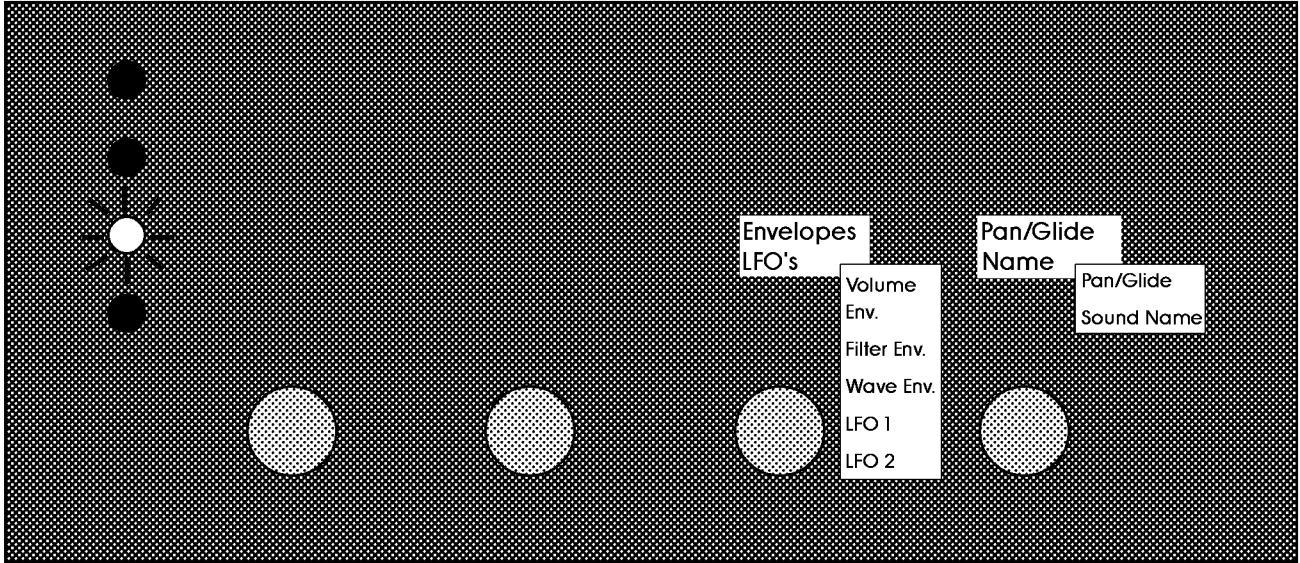
Modifier modules

- You will find all modifier modules on pages to be selected with the two [select buttons] labeled <Envelopes/LFOs> and <Pan/Glide/Name>.

Listed below are the pages of the modifier modules of Sound-edit mode:

- *Volume Envelope*
- *Filter Envelope*
- *Wave Envelope*
- *LFO 1*
- *LFO 2*
- *Pan/Glide*
- *Sound Name*

## 4.0 Modifier modules



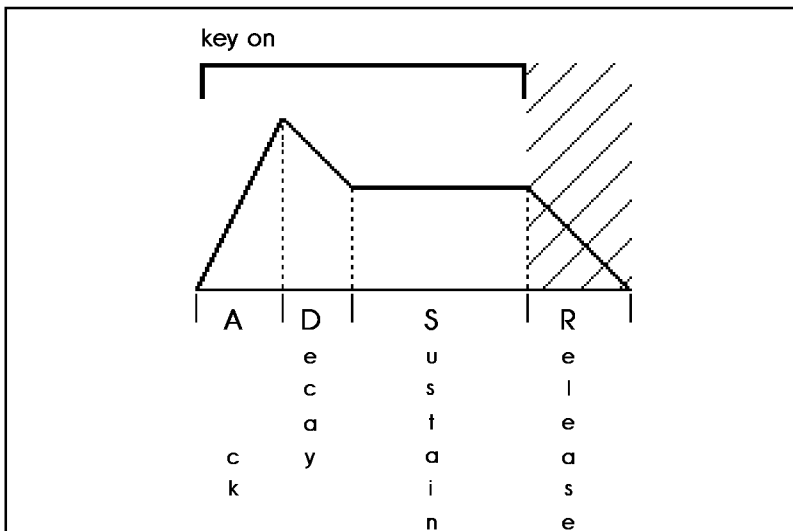
The above figure illustrates where to find the pages of the modifier modules.

# 4.1 Volume Envelope

## 4.1 Volume Envelope

The Volume envelope is a *standard ADSR-type envelope*. It is used as a pre-configured modifier-module at the Volume-module. You must use the Volume envelope at the Volume-module by setting its Envelope Amount or -Velocity parameters high enough; otherwise there will be no output at all. Therefore you must define the Volume envelope in a useful way, even if you intend to use another envelope to shape the sound's loudness.

The Volume envelope, as stated above, is a standard ADSR envelope. See the figure below for details about the associated parameters:



By pressing a key, the envelope starts to open at the time set under Attack. The attack-level is determined by the maximum amount of envelope-modulation, which might comprise both the fixed amount plus the velocity-portion, as programmed at the envelope's destination. After reaching the attack-level, the envelope will fall in the according time set at Decay to the level programmed as the Sustain parameter. After the key is released, the envelope will fade to zero in the time given by the Release parameter.

To access the Volume envelope:

- Select Sound-edit mode using the [mode button].
- Press the [select button] labeled <Envelopes/LFOs> until the page-identification field of the display reads:



You have now selected the Volume envelope page.

## 4.1 Volume Envelope

**Attack** (*O..127*) sets the attack-time of the Volume envelope.

*0* equals instant attack; the envelope opens right away to the level set as envelope amount. Due to the very fast attack there might be an audible click. Great for Hammond-revival sounds. To prevent the click, set Attack to *1*.

*127* is the longest attack-time; it takes about 8 minutes and 45 seconds (really!) until the attack-level is reached.

**Decay** (*O..127*) sets the decay-time of the Volume envelope.

*0* means instantaneous decay to the Sustain-level.

*127* is the longest decay-time; it takes about 5 minutes and 45 seconds until the Sustain-level will be reached.

**Sustain** (*O..127*) adjusts the level to which the Volume envelope will fall to in the decay-time.

*0* disables any Sustain, giving a percussive envelope that will fade away to zero in the decay- or release-time, depending on how long the key is held.

*127* leaves the envelope at the attack-level once it is reached, ignoring any Decay-parameter setting since attack- and sustain-level are equal.

**Release** (*O..127*) sets the time it takes the envelope to fade to zero after the key is released.

*0* shuts off the envelope immediately upon release of the key, or as soon as the MIDI note-off command is received.

*127* lets the envelope fade gradually to zero in about 5 minutes and 45 seconds.



VolumeEnvelope I  
Attack: 000



VolumeEnvelope I  
Decay: 000



VolumeEnvelope I  
Sustain: 127



VolumeEnvelope I  
Release: 016

## 4.1 Volume Envelope

- ❑ The Release parameter is very much dependent on the Sustain level. The higher the Sustain level is, the more the Release will be noticeable.
- ❑ If you set the Sustain to 0, the Release will start at the level present when the MIDI note-off command was received.

**Attack Modulator Source** (*modifier modules table*) defines the source-modifier-module to alter the Attack-time.

**Attack Modulator Amount** ( $-64..+63$ ) sets the maximum possible amount of modulation for the Attack modulator.

$-64$  inverts the source's output signal and applies the full modulation amount.

$+63$  applies the source as it is at full amount.

All modulation-values will scale the actual value set at the Attack parameter. Therefore, a negative value will shorten the Attack, while a positive value will prolong it.

- Use this modulator to alter the attack-time according to the sound's needs. Use velocity for changes due to playing style, keytracking for attack-times that vary with the pitch, etc.
- Experiment with other modifiers: an LFO, for instance, might produce Attacks that constantly differ from one another; a MIDI controller allows delicate phrasing by manually altering the Attack as needed. There are many more possibilities; be adventurous!

**Decay Modulator Source** (*modifier modules table*) defines the source-modifier-module to alter the Decay-time.

**Decay Modulator Amount** ( $-64..+63$ ) sets the maximum possible amount of modulation for the Decay modulator.

$-64$  inverts the source's output signal and applies the full modulation amount.

$+63$  applies the source as it is at full amount.

```
VolumeEnvelope I
AMod Src.: Velcty
```

```
VolumeEnvelope I
AMod Amnt: +00
```

```
VolumeEnvelope I
DMod Src.: Keytrk
```

```
VolumeEnvelope I
DMod Amnt: +00
```

## 4.1 Volume Envelope

All modulation-values will scale the actual value set at the Decay parameter. Therefore, a negative value will shorten the Decay, while a positive value will prolong it.

- ▶ Use the same modulation techniques described for the Attack.
- ▶ Use negative Keytracking to mimic the pitch-dependent decay-characteristic of most natural percussive instruments.

**Sustain Modulator Source** (*modifier modules table*) defines the source-modifier-module to alter the Sustain-level.

**Sustain Modulator Amount** (*-64..+63*) sets the maximum possible amount of modulation for the Sustain modulator.

*-64* inverts the source's output signal and applies the full modulation amount.

*+63* applies the source as it is at full amount.

All modulation-values will scale the actual value set at the Sustain parameter. Therefore, a negative value will decrease the Sustain-level, while a positive value will increase it.

- ▶ Use velocity to change between a more percussive and a more sustaining sound depending on how hard you strike a key.
- ▶ Use aftertouch to change the amplitude of a sound after you hit a key. By using it as a Sustain modifier rather than apply it directly at a destination you achieve a natural release that is dependent on how much aftertouch you applied. Otherwise a sudden level-jump might be introduced due to the difference of the aftertouch-value to the sustain-value.

```
VolumeEnvelope I
$Mod Src.:Afttch
```

```
VolumeEnvelope I
$Mod Amnt: +00
```

## 4.1 Volume Envelope

**Release Modulator Source** (*modifier modules table*) defines the source-modifier-module to alter the Release-time.

**Release Modulator Amount** ( $-64..+63$ ) sets the maximum possible amount of modulation for the Release modulator.

$-64$  inverts the source's output signal and applies the full modulation amount.

$+63$  applies the source as it is at full amount.

All modulation-values will scale the actual value set at the Release parameter. Therefore, a negative value will shorten the Release, while a positive value will prolong it.

- ▶ Use release-velocity to alter the Release according to how fast you let go off a key.
- ▶ Use an LFO to always alter the Release slightly to achieve a more varying overall sound.



VolumeEnvelope I  
RMod Scr:RelVel



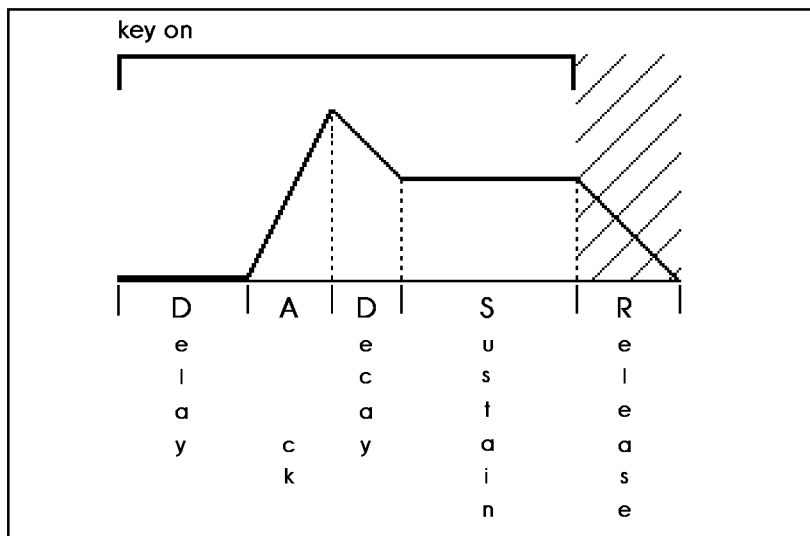
VolumeEnvelope I  
RMod Amnt: +00

## 4.2 Filter envelope

### 4.2 Filter envelope

The Filter envelope is a *standard ADSR-type envelope* with an added *Delay-segment* in the front. It is used as a pre-configured modifier-module at the Filter-module.

See the figure below for details about the associated parameters:



By pressing a key, the envelope is first delayed by the time set at the Delay parameter. It then starts to open at the time set under Attack. The attack-level is determined by the maximum amount of envelope-modulation, which might comprise both the fixed amount plus the velocity-portion, as programmed at the envelope's destination. After reaching the attack-level, the envelope will fall in the according time set at Decay to the level programmed as the Sustain parameter. After the key is released, the envelope will fade to zero in the time given by the Release parameter.

To get access to the Filter envelope:

- Select Sound-edit mode using the [mode button]. If you've been following along with the manual, you should already be there.
- Press the [select button] labeled <Envelopes/LFOs> until the page-identification field of the display reads:

FilterEnvelope I  
Delay: 000



## 4.2 Filter envelope

You have now selected the Filter envelope page.

**Delay** (*O..127*) introduces an initial delay before the Attack begins; it is totally independent from the value set as Attack.



*O* disables the Delay-segment completely, yielding a regular ADSR envelope.

*127* introduces the maximum delay before the Attack-segment Begins; it is about 36 seconds long.

**Attack** (*O..127*) sets the attack-time of the Filter envelope.



*O* equals instant attack; the envelope opens right away at the level set as envelope amount. Due to the very fast attack there might be an audible click. To prevent the click, set Attack to 1.

*127* is the longest attack-time; it takes about 8 minutes and 45 seconds until the attack-level is reached.

**Decay** (*O..127*) sets the decay-time of the Filter envelope.



*O* means instantaneous decay to the Sustain-level.

*127* is the longest decay-time; it takes about 5 minutes and 45 seconds until the Sustain-level will be reached.

**Sustain** (*O..127*) adjusts the level to which the Filter envelope will fall in the decay-time.



*O* disables any Sustain, giving a percussive envelope that will fade away to zero in the decay- or release-time, depending on how long the key is held.

*127* leaves the envelope at the attack-level once it is reached, ignoring any Decay-parameter setting since attack- and sustain-level are equal.

Modifier modules

## 4.2 Filter envelope

**Release** (*O..127*) sets the time it takes the Filter envelope to fade to zero after the key is released.

*O* shuts off the envelope immediately upon release of the key, as soon as the MIDI note-off command is received.

*127* lets the envelope fade gradually to zero in about 5 minutes and 45 seconds.

- ▣ The Release parameter is very much dependent on the Sustain level. The higher the Sustain level is, the more the Release will be noticeable.
- ▣ If you set the Sustain to *O*, the Release will start at the level present when the MIDI note-off command was received.

**Delay Modulator Source** (*modifier modules table*) defines the source-modifier-module to alter the Delay-time.

**Delay Modulator Amount** (*-64..+63*) sets the maximum possible amount of modulation for the Delay modulator.

*-64* inverts the source's output signal and applies the full modulation amount.

*+63* applies the source as it is at full amount.

All modulation-values will scale the actual value set at the Delay parameter. Therefore, a negative value will shorten the Delay, while a positive value will prolong it.

- ▶ Use this modulator to change the Delay time according to your phrasing.
- ▶ You can program great velocity dependant panorama-jumps if you route the Filter envelope to the Panning modulation-input and use some modulated Delay for it.



```
FilterEnvelope I
Release:      016
```



```
FilterEnvelope I
D1Md Src.:LFO 1
```



```
FilterEnvelope I
D1 Md Amnt:  +00
```

## 4.2 Filter envelope

**Attack Modulator Source** (*modifier modules table*) defines the source-modifier-module to alter the Attack-time.

**Attack Modulator Amount** ( $-64..+63$ ) sets the maximum possible amount of modulation for the Attack modulator.

$-64$  inverts the source's output signal and applies the full modulation amount.

$+63$  applies the source as it is at full amount.

All modulation-values will scale the actual value set at the Attack parameter. Therefore, a negative value will shorten the Attack, while a positive value will prolong it.

**Decay Modulator Source** (*modifier modules table*) defines the source-modifier-module to alter the Decay-time.

**Decay Modulator Amount** ( $-64..+63$ ) sets the maximum possible amount of modulation for the Decay modulator.

$-64$  inverts the source's output signal and applies the full modulation amount.

$+63$  applies the source as it is at full amount.

All modulation-values will scale the actual value set at the Decay parameter. Therefore, a negative value will shorten the Decay, while a positive value will prolong it.

```
FilterEnvelope I
AMod Src.:Velcty
```

```
FilterEnvelope I
AMod Amnt: +00
```

```
FilterEnvelope I
DcMd Src.:Velcty
```

```
FilterEnvelope I
Dc Md Amnt: +00
```

Modifier modules

## 4.2 Filter envelope

**Sustain Modulator Source** (*modifier modules table*) defines the source-modifier-module to alter the Sustain-level.

**Sustain Modulator Amount** ( $-64..+63$ ) sets the maximum possible amount of modulation for the Sustain modulator.

$-64$  inverts the source's output signal and applies the full modulation amount.

$+63$  applies the source as it is at full amount.

All modulation-values will scale the actual value set at the Sustain parameter. Therefore, a negative value will decrease the Sustain-level, while a positive value will increase it.

**Release Modulator Source** (*modifier modules table*) defines the source-modifier-module to alter the Release-time.

**Release Modulator Amount** ( $-64..+63$ ) sets the maximum possible amount of modulation for the Release modulator.

$-64$  inverts the source's output signal and applies the full modulation amount.

$+63$  applies the source as it is at full amount.

All modulation-values will scale the actual value set at the Release parameter. Therefore, a negative value will shorten the Release, while a positive value will prolong it.

```
FilterEnvelope I
$Mod Src.:Afttch
```

```
FilterEnvelope I
$Mod Amnt: +00
```

```
FilterEnvelope I
RMod Src.:RelVel
```

```
FilterEnvelope I
RMod Amnt: +00
```

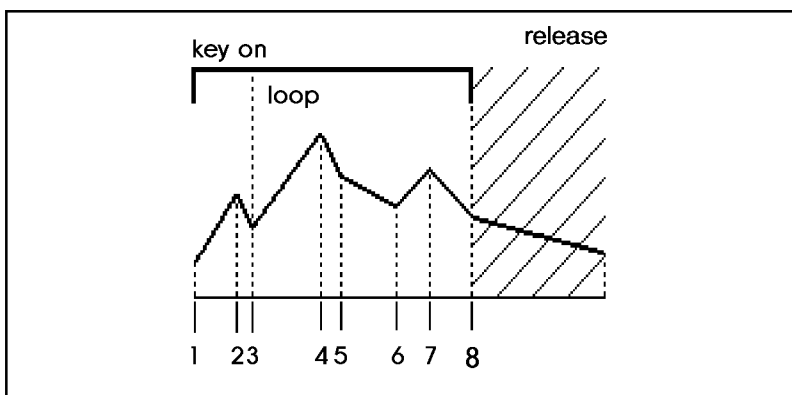
## 4.3 Wave Envelope

### 4.3 Wave envelope

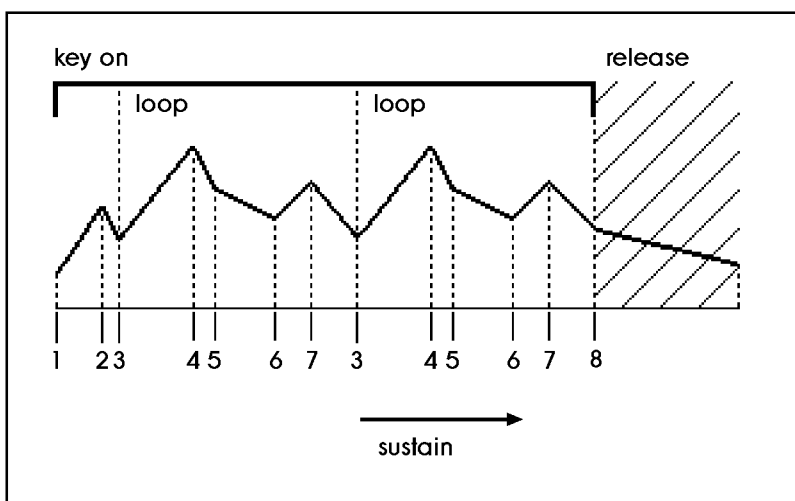
The Wave envelope is the pre-configured envelope for the Wave-modules.

It differs very much from the other two envelopes. It is a *multi-segment envelope* that can be split into a key-on and key-off (release) portion as you like. Additionally you may install a loop either during the key-on- or the release-portion of the envelope.

See the below figure for further details:

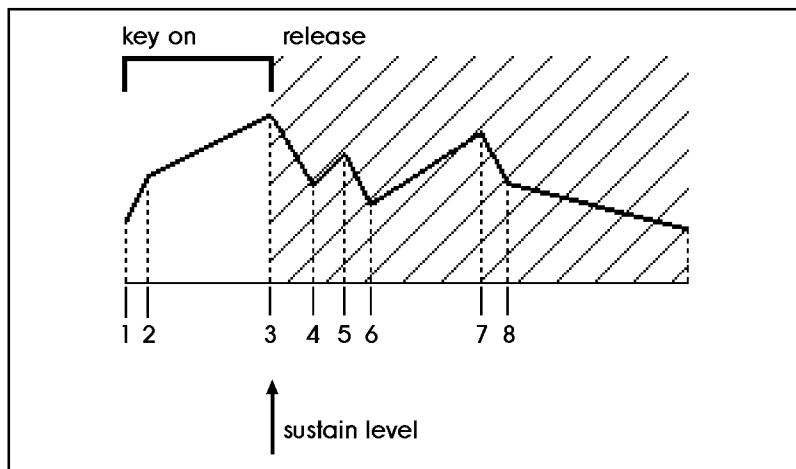


Above you see one application: 7 segments of the Wave-envelope are defined for the key-on period, while one segment governs the release phase. If the key is held longer than the combined segment-times last, the envelope will jump back to segment 3 and repeat segments 3 through 7 until the key is released, to finish as programmed in segment 8, which is the release segment:



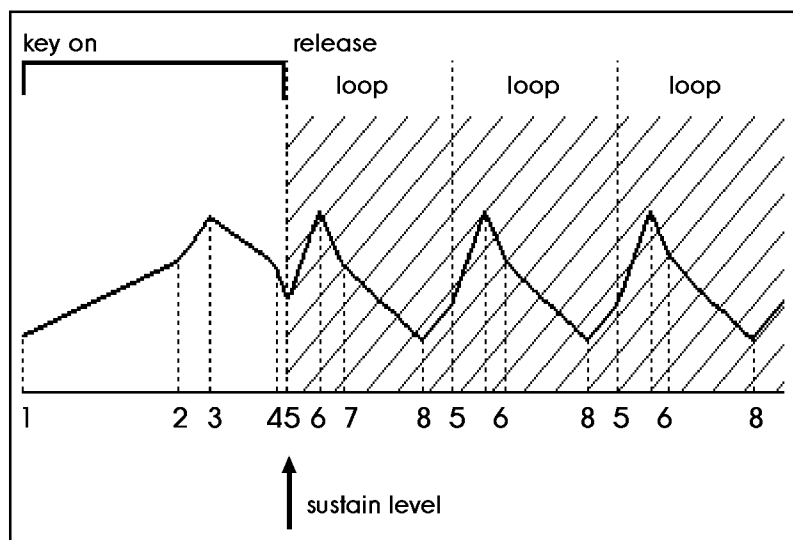
## 4.3 Wave Envelope

Next you find a Wave envelope with only 2 key-on-, but 6 release-segments defined, without any loop:



When a key is pressed, the first two segments are played until Level 2 is reached. The envelope then will remain at this level until the key is released, since there is no loop defined. Upon release of the key, the other 6 segments are run through.

Finally, you might program a loop to take place in the release phase of the envelope:



Here, the envelope will run through the 4 key-on segments, half at Level 4 until the key is released to play and repeat the 4 release segments over and over.

## 4.3 Wave Envelope

To access the Wave envelope:

- Select Sound-edit mode using the [mode button]. If you've been following along with the manual, you should already be there.
- Press the [select button] labeled <Envelopes/LFOs> until the page-identification field of the display reads:



```
Wave Envelope I
Time 1: 032
```

You have now selected the Wave envelope page.

**Time 1** (*O..127*) sets the time it takes for segment 1 of the Wave envelope to travel to Level 1.



```
Wave Envelope I
Time 1: 032
```

*O* jumps immediately to Level 1; the time-value is disabled.

*127* is the maximum time one segment can use to reach it's level.

**Level 1** (*O..127*) sets the level to be reached by Time 1. Each Time/Level pair makes up one segment of the multi-segment Wave envelope.



```
Wave Envelope I
Level 1: 127
```

*O* is the minimum level, evoking the base-value of the destination the Wave envelope is routed to.

*127* is the maximum value, equal to the maximum value comprising both the Amount and Velocity portions set at the envelope modulation.

**Time 2..8 / Level 2..8** define segments **2** through **8** respectively and are completely identical to the first segment-pair. The only difference is that each pair succeeds the previous one.

There can be no unused segment in the Wave envelope.

## 4.3 Wave Envelope

**Time Modulator Source** (*modifier modules table*) defines the source-modifier-module to alter Time-values concurrently.

```
Wave Envelope I
TMod Src.:Keytrk
```

**Time Modulator Amount** ( $-64..+63$ ) sets the maximum possible amount of modulation for the Time modulator.

```
Wave Envelope I
TMod Amnt: +00
```

$-64$  inverts the source's output signal and applies the full modulation amount.

$+63$  applies the source as it is at full amount.

All modulation-values will scale the actual values set at the Time parameters. Therefore, a negative value will shorten the Time-values, while a positive value will prolong them.

**Level Modulator Source** (*modifier modules table*) defines the source-modifier-module to alter all Levels simultaneously.

```
Wave Envelope I
LMod Src.:Afttch
```

**Level Modulator Amount** ( $-64..+63$ ) sets the maximum possible amount of modulation for the Level modulator.

```
Wave Envelope I
LMod Amnt: +00
```

$-64$  inverts the source's output signal and applies the full modulation amount.

$+63$  applies the source as it is at full amount.

All modulation-values will scale the actual values set at the Level parameters. Therefore, a negative value will decrease all Levels, while a positive value will increase them.



## 4.3 Wave Envelope

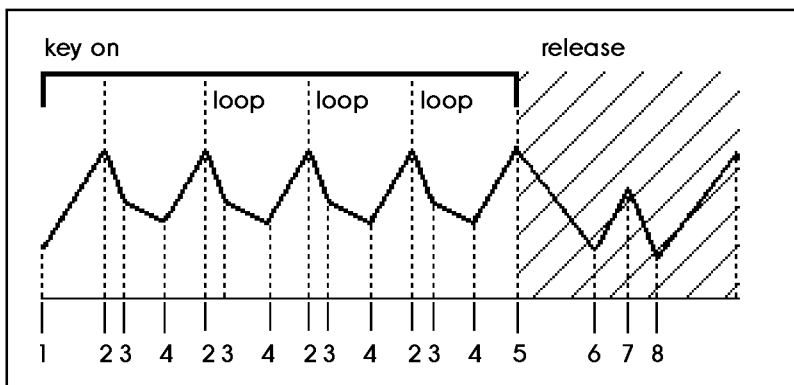
- ▶ Use the modulation-inputs to change the Wave envelope's characteristic according to your playing by using velocity, aftertouch, poly pressure or a MIDI controller to change the Times or Levels in realtime.
- ▶ Use the *max* fixed-modifier-module as the source to scale all Time- or Level-values equally. This speeds up the process of increasing or decreasing the Wave-envelope's values enormously.

**Key Off Point** (1..8) defines the border between the key-on and the release portion of the Wave envelope. The Key Off Point is the last segment of the key-on portion. If there is no loop defined, the Level of the Key Off Point is the sustain-level at which the envelope will remain until the key is released (the MIDI note-off command is received) and the release-portion of the Wave envelope begins.

**Loop Start Point** (1..8) defines the segment at which the loop will start if Loop Mode is set to on.

The loop will always run between the Loop Start Point and the Key Off Point. There exists a total of three different algorithms that might apply when using a loop:

1. *Loop Start Point is in front of Key Off Point:* The Wave envelope will run through all key-on segments once. After reaching the Key Off Level, it jumps back to the Loop Start segment and plays again to the Key Off level, until the MIDI note-off command is received. Look at the figure below:

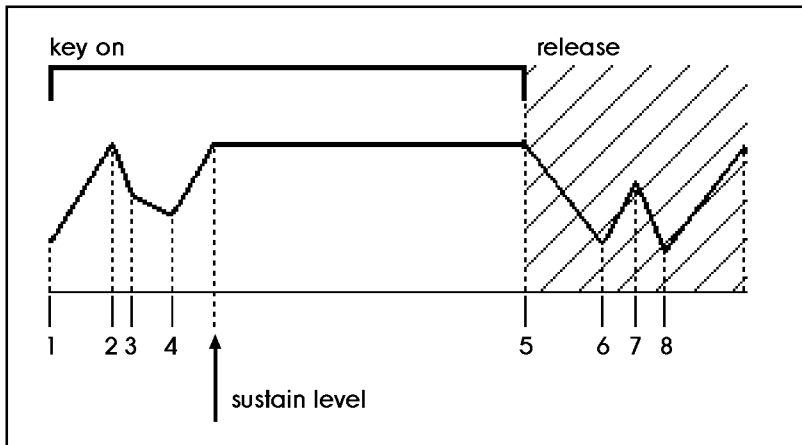


Wave Envelope I  
Key Off Point: 2

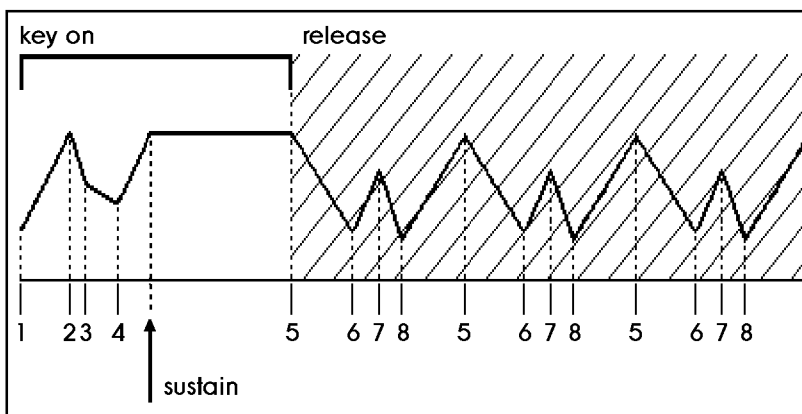
Wave Envelope I  
Loop Start Pt: 1

## 4.3 Wave Envelope

2. *Loop Start Point is identical to Key Off Point*: In this case, a loop is created that only loops the Key Off Level. No loop will be audible; this is the same as turning Loop Mode off.



3. *Key Off Point is in front of Loop Start Point*: This way you can create a loop in the release portion of the Wave envelope. The envelope will run through all its key-on segments, remain at the Key Off Level until the MIDI note-off command is received, play all the release-segments up to the Loop Start Point, jump back to the Key Off Point and repeat the loop.



**Loop Mode** (*on/off*) defines if the loop will be active or not.

Wave Envelope I  
Loop Mode: off

*on* will enable looping.

*off* disables looping. The Wave envelope will pass all key-on segments, remain at the Key Off Point Level until the MIDI note-off is received, to finally run through the release portion. It will keep the Level of the last segment, and if that is different from 0, at least a Level will remain of the Wave envelope.

## 4.3 Wave Envelope

Here are a few examples of how to use the Loop feature:

- ▶ Use a release loop to create unique echo effects. Route the Wave-envelope to Volume and, if you like, Panning also. Set a long Release to both the Filter and Volume envelope and play around with the Wave envelope's release segments.
- ▶ Use only segments 1 through 3 without a loop to recreate a standard ADSR envelope. Set Level 1 to 127. Time 1 will be the attack-time, travelling to the attack-level (Level 1) which is set using the envelope amount and velocity parameters. It then falls at the decay-time (Time 2) to the sustain-level (Level 2). As soon as the key is released, the envelope will fall at the release-rate (Time 3) to zero (set Level 3 and all succeeding Levels to 0).

## 4.4 LFO 1

### 4.4 LFO 1

This Low Frequency Oscillator is rather elaborate in nature, providing some effects that are unique to the MicroWave. LFO 1 is exclusively found in the modifier-modules table. If you want to use it, you must assign it to a routeable modulation-input.

In general, you may set up a certain waveform at a certain rate, modulate both rate and level and finally apply an AD/AR envelope to control the output.

To access LFO 1:

- Select Sound-edit mode using the [mode button]. If you've been following along with the manual, you should already be there.
- Press the [select button] labeled <Envelopes/LFOs> until the page-identification field of the display reads:

LFO 1	I
Rate:	050

You have now selected the LFO 1 page.

## 4.4 LFO 1

**Rate** (1..127) sets the speed of LFO 1.

1 sets the slowest sweeping speed.

127 sets the highest possible rate. It is close to the low audio range, useful for "flutter-tongue" effects.

**Shape** (*sin/tri/pulse/random*) lets you choose one of four waveforms.

*sin* selects a sinewave. Use negative amount at the destination to change the phase 180 degrees.

*tri* selects a triangle-wave. However, it's true waveform is set by the Symmetry parameter.

at 00 it is a *triangle-wave*,

at +63 it is a *positive ramp sawtooth*,

at -64 it is a *negative ramp sawtooth*.

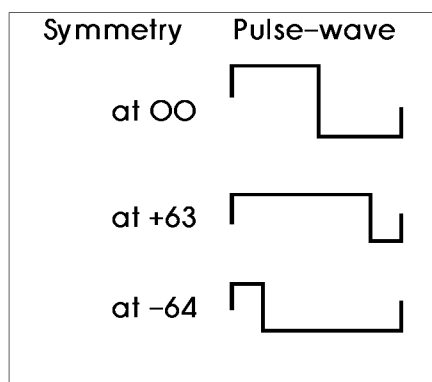
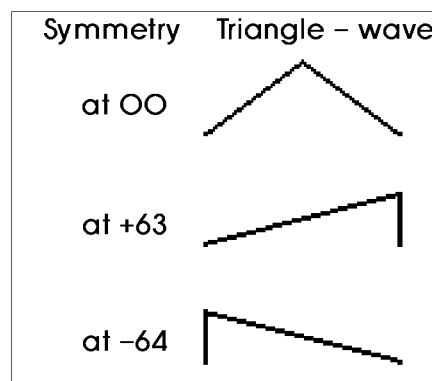
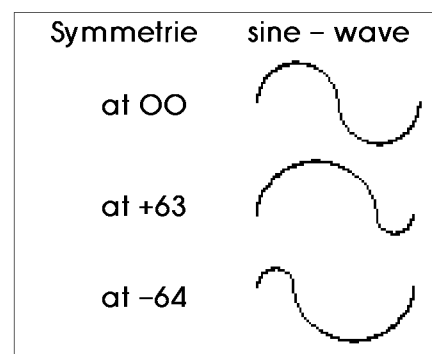
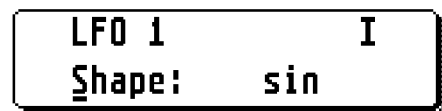
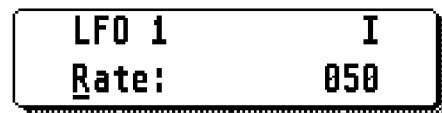
*pulse* is a pulsewave, whose pulse-width is set by the Symmetry function.

At 00 it is a *square wave* with 50% duty-cycle,

at +63 it is a pulse width about 95% *duty cycle*,

at -64 it is a pulse width about 5% *duty cycle*.

*random* is a random waveform. Use Rate to adjust it's intensity. Symmetrie does not influence the random waveshape.



**Symmetry** (-64..+63) further edits the waveform of the LFO.

-64 will change the symmetry-axis of the waveform to the beginning. See "Shape" for details.

00 leaves the symmetry-axis of a waveform in the middle. This yields a perfect sine-wave, for instance.

+ 64 changes the symmetry-axis of the waveform towards the end. Again, see "Shape" for details.

LFO 1	I
Symmetry:	+00

**Humanize** (off..7) imposes random variations upon the Rate of the LFO.

off keeps the Rate absolutely constant at the value set.

7 continuously imposes rather large random variations on the Rate.

LFO 1	I
Humanize:	off

**Rate Modulator Source** (modifier modules table) defines the source-modifier-module to alter the LFO 1 Rate.

LFO 1	I
RMod Src.:	Velcty

**Rate Modulator Amount** (-64..+63) sets the maximum possible amount of modulation for the Rate modulator.

LFO 1	I
RMod Amnt:	+00

-64 inverts the source's output signal and applies the full modulation amount.

+63 applies the source as it is at full amount.

All modulation-values will be added to the actual value set at the Rate parameter. Therefore, a negative value will slow down the Rate, while a positive value will speed it up.

- ▶ Use the mod-wheel to increase the Rate while applying vibrato, enhancing it's intensity.
- ▶ Use LFO 1 itself to modulate the Rate to achieve even faster speeds.

**Level Modulator Source** (*modifier modules table*) defines the source-modifier-module to alter the LFO Level.

Since the actual level of LFO intensity is set at the destination, there is no further amount value needed here. Instead, full, positive modulation is always applied whenever a source is selected.

If you do not want to use any dynamic modulation at all, select the max fixed-modifier-module as a source.

- ▶ Use a mod-wheel or other MIDI controller to adjust the LFO level in realtime.
- ▶ Use LFO 2 to even further alter the waveform of LFO 1.

**Sync** (*off/on*) synchronizes the LFO for all individual voices.

*off* leaves the LFO independent for each triggered voice. This way you will achieve a very natural sounding LFO modulation, since all LFOs will start at a different point in time. Use "Humanize" to even further enhance this behavior.

*on* synchronizes the LFOs of all voices. This is great to achieve typical analogue-synthesizer effects of those days when there was only one LFO to modulate all voices concurrently. Use a modest "Humanize" value to imitate bad circuit design...

**Delay** (*off/retrigger/1..127*) delays the start of LFO 1. If Delay is set to anything except 'off', the "**Sync**" parameter will be disabled automatically.

*off* does not produce any delay at all.

*retrigger* does not delay LFO 1 but starts the waveform new whenever a key is pressed. This allows you to synchronize modulations to each keypress. This is great for achieving reliable phasing and chorus effects.

1..127 delays the time in which the LFO waveform 'kicks in'. This should not be confused with the attack-time of the LFO. Use the Attack parameter to determine the post-delay attack-slope of the LFO.

```
LFO 1      I
LMod Src.: max.
```

```
LFO 1      I
Sync:      off
```

```
LFO 1      I
Delay:      off
```

## 4.4 LFO 1

Next you'll find a useful, though small, AD/AR envelope to control the LFO 1 level. Notice that you may use this envelope elsewhere by assigning it to a routeable modulation-input. However, keep in mind that the LFO envelope will always be applied to the LFO 1 also.

**Attack** (*O..127*) sets the attack-time for the LFO envelope.

*O* gives instantaneous attack after the delay-time has passed.

*127* is the longest attack-time, requiring several minutes to fade in the LFO completely.

**Decay** (*off/1..127*) adjusts the decay-time of the LFO envelope.

*off* does not introduce any decay at all. Rather, the LFO will fade in according to the Attack parameter and stay constant throughout the keypress. To guarantee a smooth LFO-modulation, upon release of the key the LFO will fade out at the release-rate of the Volume envelope. In this configuration, the LFO envelope is turned into an AR (attack/release) envelope.

*1..127* adjust the decay-time of the LFO envelope, which now is an AD (attack/decay)-type envelope. *1* is the shortest, *127* the longest decay-time.

LFO 1	I
Attack:	000

LFO 1	I
Decay:	off



## 4.5 LFO 2

LFO 2 is very similar to LFO 1. The difference is that LFO 2 is not as elaborate as LFO 1 and features only the most common parameters. However, by using a sidechain-modulator at the destination you may achieve similar results as you would using LFO 1.

LFO 2 is exclusively found in the modifier-modules table. If you want to use it, you must assign it to a routeable modulation-input.

In general, you may set up a certain waveform at a certain rate.

To access LFO 2:

- Select Sound-edit mode using the [mode button]. If you've been following along with the manual, you should already be there.
- Press the [select button] labeled <Envelopes/LFOs> until the page-identification field of the display reads:

You have now selected the LFO 2 page.

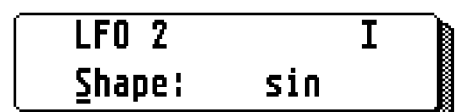
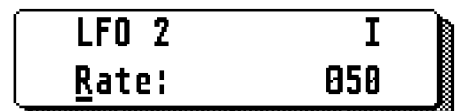
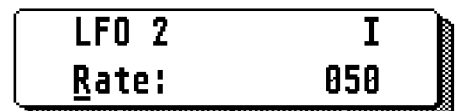
**Rate** (*1..127*) sets the speed of LFO 2.

*1* sets the slowest sweeping speed.

*127* sets the highest possible rate. It is close to the low audio range, useful for "flutter-tongue" effects.

**Shape** (*sin/tri/pulse/random*) lets you choose one of four waveforms.

*sin* selects a sinewave. Use negative amount at the destination to change the phase 180 degrees.



## 4.5 LFO 2

*tri* selects a triangle-wave. However, it's true waveform is set by the Symmetry parameter.

At 00 it is a *triangle-wave*,  
at +63 it is a *positive ramp sawtooth*,  
at -64 it is a *negative ramp sawtooth*.

*pulse* is a pulsewave, whose pulse-width is set by the Symmetry function.

At 00 it is a *square wave* with 50% duty-cycle,  
at +63 it is a pulse width about 95% *duty cycle*,  
at -64 it is a pulse width about 5% *duty cycle*.

*random* is a random waveform. Use Rate to adjust it's intensity.

**Symmetry** (-64..+63) further edits the waveform of LFO 2.

LFO 2	I
Symmetry:	+00

-64 will change the symmetry-axis of the waveform to the beginning.

00 leaves the symmetry-axis of a waveform in the middle. This yields a perfect sine-wave, for instance.

+ 64 changes the symmetry-axis of the waveform towards the end.

**Humanize** (off..7) imposes random variations upon the Rate of LFO 2.

LFO 2	I
Humanize:	off

*off* keeps the Rate absolutely constant at the value set.

7 continuously imposes rather large random variations on the Rate.

## 4.6 Glide

### 4.6 Glide

Glide gives you access to both portamento and glissando effects. Rather than immediately playing the corresponding pitch of a struck key, Glide gradually sweeps from the previous pitch to the new one. It works such that the highest pitch of the previously struck chord will sweep again to the highest, the lowest to the lowest pitch of the newly struck chord and so on.

To access Glide:

- Select Sound-edit mode using the [mode button]. If you've been following along with the manual, you should already be there.
- Press the [select button] labeled <Pan/Glide/Name> until the page-identification field of the display reads:

If instead it reads

...press the [select button] once more to switch from the Sound Name page to the Pan/Glide page.

You have now selected the Glide page.

The first three parameters of this page are devoted to Panning; the next three parameters let you program Glide.

**Glide** (*off/Gliss/Porta/MGliss/MPorta*) selects the basic type of Glide.

*off* disables Glide completely, regardless of the other parameters.

*Gliss* selects a glissando Glide effect. Rather than continuously moving from one pitch to the next, Gliss uses semitone-steps to glide between the two pitches.

*Porta* selects the portamento-effects, resulting in a continuous transition between the last and new pitch.

Pan / Glide	I
<u>G</u> lide:	off

Sound Name	I
<u>S</u> ound Init	

Modifier modules

Pan / Glide	I
<u>G</u> lide:	off

## 4.6 Glide

*MGliss* is the same as *Gliss*, with the difference being that the glissando effect will only be applied if a MIDI portamento-command (MIDI controller #65) is received.

*MPorta* is the same as *Porta*, with the difference being that the portamento effect will only be applied if a MIDI portamento-command (MIDI controller #65) is received.

**Glide Rate** (*O..127*) sets the speed of the Glide-effect.

*O* disables any Glide, since the rate is set to be instantaneous, having the same effect as if Glide were set to off.

*127* selects the longest possible Glide Rate.


**Glide Mode** (*Time/Dist*) lets you choose between two different ways of applying Glide: equal Time and equal Distance.

*Time* selects the equal time option. It means that a glide always takes the same time to reach it's destination pitch regardless of the distance it must travel. In other words, if the interval the pitch must sweep is a minor third, the actual glide-speed will be slower than of a glide over three octaves. However, the time it takes to reach the destination pitch will be the same for both intervals. This is useful if you play chords in open position that must hit at the exact same time after a glide. The *Glide Rate* parameter sets the time it takes to reach the destination pitch.

*Dist* selects the equal distance option. It means that a glide takes place always at the same speed for the same distance regardless of the time it may take to reach the destination pitch. Therefore, a glide over two octaves takes 8-times the time of a glide over a minor third. This is good, for instance, if you use *Gliss* and want the semitone-movement to stay in rhythm. The *Glide Rate* parameter sets the actual speed of the glide.



```
Pan / Glide I
Glide Rate: 032
```



```
Pan / Glide I
Glide Mode: Dist
```

## 4.7 Temperment

### 4.7 Temperment

Temperment lets you select the intonation used to play the Sound-program.

In Multi mode, the according Instrument parameter "Temperment" will over-ride this parameter of the Sound-program.

To access Temperment:

- Select Sound-edit mode using the [mode button]. If you've been following along with the manual, you should already be there.
- Press the [select button] labeled <Pan/Glide/Name> until the page-identification field of the display reads:
- If instead it reads

...press the [select button] once more to switch from the Sound Name page to the Pan/Glide page.

You have now selected the Pan/Glide page, in which you also will find the Temperment parameter.

The first three parameters of this page are devoted to Panning, the next three parameters let you program Glide. The last parameter, finally, allows adjusting of the Temperment parameter.

Pan / Glide I  
Temperment: ln+

Sound Name I  
Sound Init

Modifier modules

## 4.7 Temperment

**Temperment** (*Ln+ / Ln- / rn1 / rn2 / TT1 / TT2*) is the parameter to select the desired intonation the Sound-program shall be played with. Temperment always affects the entire keyboard range; it is not simply a recursive detuning of the intervals within one octave.

Refer to the section "Tables" of the Performance manual for more detail.

*Ln+* is the regular equal-tempered intonation you find all around you.

*Ln-* is also an equal-tempered intonation; however, it is exactly inverted to the incoming MIDI note-number. An incoming notenumber 1 will produce a pitch equal to note-number 127, notenumber 30 will be pitched like number 97, etc.

*rn1* will use the incoming MIDI notes as they are but will randomly detune them very slightly to achieve a more 'acoustic' effect.

*rn2* does essentially the same as *rn1*, the only difference being in the greater amount of detuning it exerts on the notes. Use this on one Instrument on a layered sound in Multi-mode to create a beautiful dynamically changing chorus effect.

*TT1* is the user-definable tuning-table 1. You can map any incoming note to any note of your choice and, on top of that, detune it by up to a semitone in each direction. Refer to section "Tables" of the Performance manual for more details.

*TT2* is the user-definable tuning-table 2. You can map any incoming note to any note of your choice and, on top of that, detune it by up to a semitone in each direction. Refer to section "Tables" of the Performance manual for more details.

*TT3 & 4* selects either of the two Tuning-tables on an expansion card. Since you cannot edit a card's table, you must first program all tables in the MicroWave's internal memory and then transfer them to the card using the datatransfer mode of the store-function.



Pan / Glide I  
Temperment: ln+

## 4.8 Sound Name

### 4.8 Sound Name

Using this parameter you can give your Sound-program a name. This name appears upon Sound-program-selection in Single mode. It also appears at the Instrument-parameter "Sound Program" in Multi mode; however, there only the first 9 characters of the name will be visible.

To access Sound Name:

- Select Sound-edit mode using the [mode button]. If you've been following along with the manual, you should already be there.
- Press the [select button] labeled <Pan/Glide/Name> until the page-identification field of the display reads:
- If instead it reads

...press the [select button] once more to switch from the Pan/Glide page to the Sound Name page. There is only one parameter on this page: the Sound Name.



```
Sound Name  I
Sound Init  I
```



```
Pan / Glide  I
Imperment:  In+
```

Modifier modules

## 4.8 Sound Name

**Sound Name** lets you name the Sound-program you currently edit. You can use up to 16 characters to define a name.



Sound Name I  
Sound Init

- After selecting the page Sound Name press the [parameter/value button] once.
- Now use the [alpha dial] to set the first character of the new name.
- Press the [parameter/value button] again to advance to the next character-position.
- Select it's character with the [alpha dial] again as described above.
- Repeat this process until you reach the last character. You must input all 16 possible characters; use blanks wherever you wish to place an empty space.
- After inputting the 16th character, press the [parameter/value button] once more. You are now back in the parameter-select level. Use the [parameter/value button] and [alpha dial] as usual.



# Chapter 5

Quick Edit

**5.1 Fast Access**

**5.2 Envelope Macros**

**5.3 Modulation Macros**





### Quick Edit

This chapter tells you everything about those quick changes you always wished you could do on other synthesizers. The MicroWave development team devoted an entire mode to those quick changes. Welcome to Quick Edit!

There are two different Quick Edit types: FastAccess and Macros.

FastAccess gives you instantaneous control over the most vital aspects of a sound. It allows you to control several modules of the MicroWave simultaneously with a single turn of the [alpha dial] to achieve an overall sound edit.

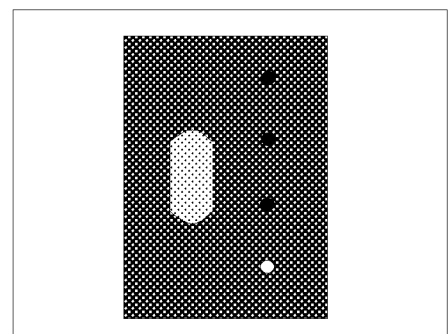
Macros recall certain preconfigured parameter settings that give you instant envelope-shapes or modulation-effects.

Using Macros and FastAccess will open up brand new sound possibilities in seconds. If you like, you can always switch to Sound-edit mode for tailoring a Sound-program exactly to your need.

Especially if you are in the studio with a producer on your back that wants "more percussive bite" or "less touch-sensitivity" for a sound, you will appreciate the FastAccess parameters.

To use Macros and FastAccess, you must switch to the Quick Edit mode:

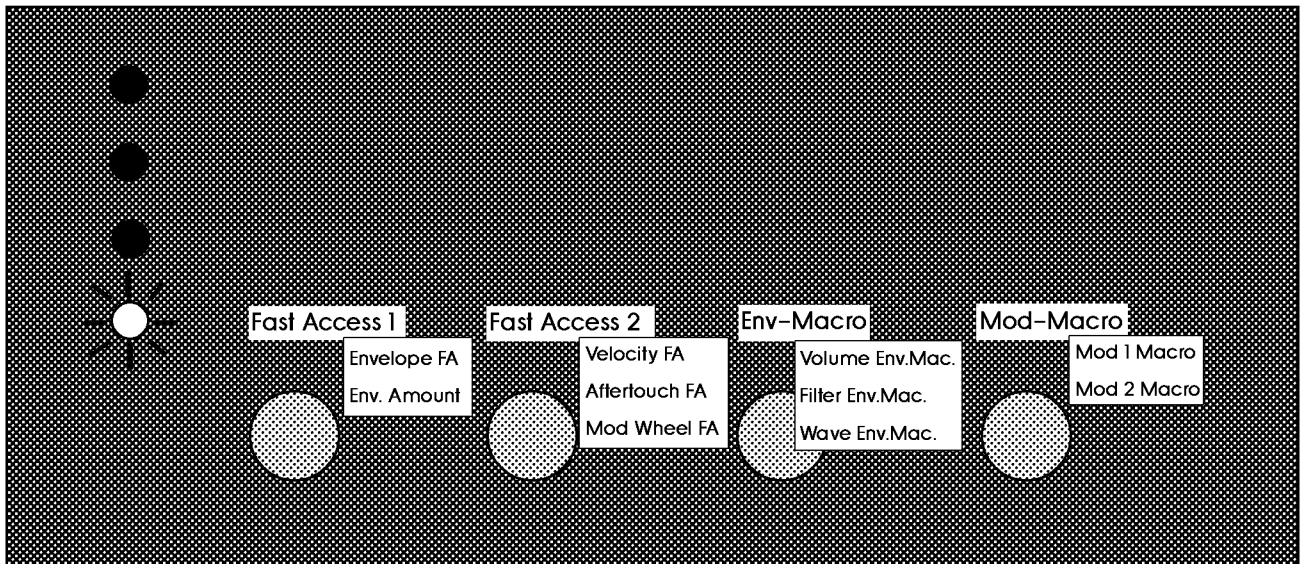
- Use the [mode button] to switch to Quick Edit mode.
- Use the two left [select buttons] to access the FastAccess pages and the other two to select Macro-functions.



## 5.0 Quick Edit

Quick edit mode is comprised of the following pages:

- *Envelope FA*
- *Env. Amount FA*
- *Velocity FA*
- *Aftertouch FA*
- *Mod Wheel FA*
- *Volume Env. Mac*
- *Filter Env. Mac*
- *Wave Env. Mac*
- *Mod1 Mac*
- *Mod2 Mac*



The above figure illustrates where to find the Quick Edit pages.

## 5.1 FastAccess

### 5.1 FastAccess pages

Since FastAccess is slightly different from the rest of the MicroWave's user-interface, here is a general explanation about using it.

All FastAccess-parameters of one page are displayed **simultaneously**.

As elsewhere in the MicroWave user-interface, you can edit the parameter that the cursor is currently residing under:

In the above shown example that would be the *Attack* of the Envelope FA page, since the cursor rests under the A.

To change to another parameter, select it using the [parameter/value button]. With each press on the button the cursor will advance to the next parameter. A single press on the [parameter/value button] would change the above display to:


Contrary to the regular parameter/value selection, you need not press the [parameter/value button] to switch from parameter-selection to value-input. Each parameter is in constant value-input mode, activated whenever the cursor resides under the desired parameter.

There are three symbols a FastAccess parameter can display:

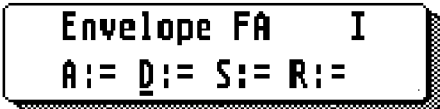
= indicates that all parameters are in a definite condition resembling the currently stored Sound-program-parameters.

+ shows that the Sound-program-parameter values have been increased.

- shows that the Sound-program-parameter values have been decreased.



```
Envelope FA   I
A:= D:= S:= R:=
```



```
Envelope FA   I
A:= D:= S:= R:=
```

Quick Edit

## 5.1 FastAccess

As long as you edit only a single FastAccess parameter, the MicroWave still knows the original values and calculates any parameter changes in relation to these original values. However, as soon as you jump to another FastAccess parameter all previous changes will be permanent, indicated by a " = " symbol next to the parameter. If you now select that parameter again, all changes will be based upon of the newly stored parameters. Of course, the Sound-program is still only in an edit-buffer. Therefore, the original Sound-program can be recalled at any time using the 'Recall' command.

All of this might sound more difficult than it really is. Just try it out with a couple of sounds, and you will love it immediately.

To summarize the user-interface functions:

- Select the correct FastAccess page you want to work on; in this example it would be the Envelope FA page.
- Place the cursor under the parameter you want to edit; in this example it would be the Decay-value.
- Use the [alpha dial] to edit this parameter directly; in the example the decays of all envelopes would be increased.
- Press the [parameter/value button] to advance to the next parameter of that FastAccess page you want to edit.
- Use the [alpha dial] again to edit this parameter directly; in the example to the right the release time of all envelopes would be shortened.

```
Envelope FA   I  
A:= D:= S:= R:=
```

```
Envelope FA   I  
A:= D:= S:= R:=
```

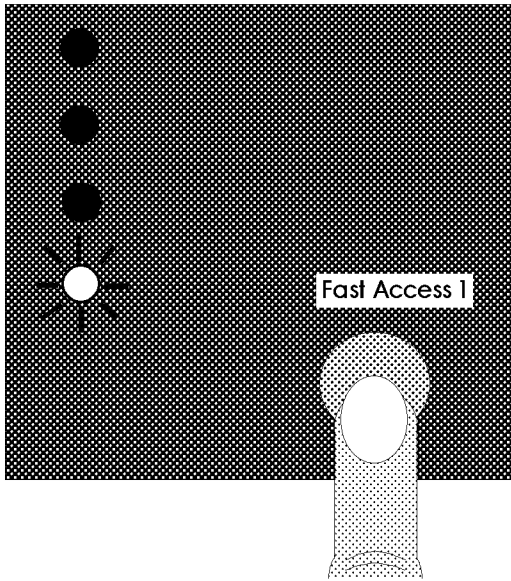
```
Envelope FA   I  
A:= D:=+ S:= R:=
```

```
Envelope FA   I  
A:= D:= S:= R:=
```

```
Envelope FA   I  
A:= D:= S:= R:-
```

## 5.1 FastAccess

- Press the [select button] to toggle between pages or to access a different FastAccess parameter.



## 5.1.1 Envelope FastAccess

### 5.1.1 Envelope FastAccess

Envelope FastAccess allows you to quickly change the shape of all envelopes simultaneously. This is done in an intelligent way: parameters are recalculated on a percentage-basis, so that their relationship between each other will remain intact.

To access the **Envelope FA** page:

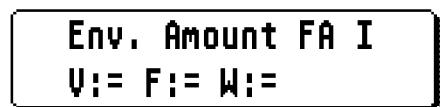
- Select the Quick Edit mode using the [mode button].

- Press the [select button] labeled <FastAccess 1> once. The display's page identification field should read:



```
Envelope FA I
A:= D:= S:= R:=
```

- If instead it reads:



```
Env. Amount FA I
V:= F:= W:=
```

...toggle the page by pressing the [select button] once more.

All four envelope-parameters are displayed simultaneously. As explained above, you can change the parameter that is shown with the cursor underneath.



```
Envelope FA I
A:= D:= S:= R:=
```

**A:** adjusts the *Attack rates* of the Volume- and Filter-envelopes and Time 1 of the Wave-envelope. Increasing the Attack will prompt a '+' instead of the '='-symbol, while shortening it displays a '-' sign.

By decreasing the Attack parameter heavily, the MicroWave assumes that you want to achieve a more percussive sound. Therefore if the attack is already close to or is instantaneous, it will begin to decrease the Decay parameter and even lower the Sustain level to make the Sound-program more and more percussive.



## 5.1.1 Envelope FastAccess

Increasing the Attack, on the other hand, will prolong the Decay time above a certain value and even raise the Sustain to achieve a more sustaining envelope.

**D:** alters the *Decay times* of the Volume- and Filter-envelope and Time 2 of the Wave-envelope if the Key Off Point is set equal to or greater than to 2. Increasing the Decay will prompt a '+' instead of the '=' symbol, while shortening it displays a '-' sign.

Decreasing the Decay below a certain value will simultaneously decrease the Attack, and even the Sustain level, since the MicroWave assumes you are striving for a more percussive sound.

Increasing the Decay beyond a certain point will also increase the Sustain in order to produce a more sustaining envelope.

**S:** changes the *Sustain levels* of the Volume- and Filter-envelopes and the Key Off Point level of the Wave-envelope. Increasing the Sustain level will prompt a '+' instead of the '=' symbol, while lowering it displays a '-' sign.

Decreasing the Sustain below a certain value will simultaneously decrease the Decay, since the MicroWave assumes you strive for a more percussive sound.

Increasing the Sustain will only alter the respective Sustain levels.

**R:** adjusts the *Release rates* of all envelopes; for the Wave-envelope all rates beyond the Key Off Point will be shortened simultaneously. Increasing the Release times will prompt a '+' instead of the '=' symbol, while shortening them displays a '-' sign.

Only the respective Release values are altered by the *R:* parameter.

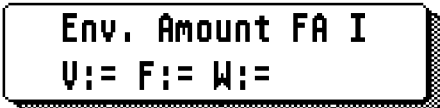
## 5.1.2 Envelope Amount FastAccess

### 5.1.2 Envelope Amount FastAccess

Envelope Amount FastAccess allows you to quickly change the Amount-value of all preconfigured envelopes. This is simply a shortcut for setting the amounts. It comes in handy after altering the envelopes to adjust their relative Amounts. Also, if you use the Velocity FastAccess function to adjust the velocity-sensitivity of all envelopes, you might want to alter just the Envelope amounts again, since the FastAccess Velocity parameter recalculates both the velocity- and the amount-portion of the envelope modulation.

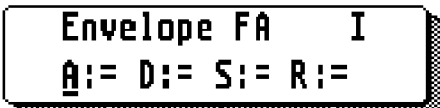
To access the **Env Amount FA** page:

- Select the Quick Edit mode using the [mode button].
- Press the [select button] labeled <FastAccess 1>. The display's page identification field should read:



Env. Amount FA I  
V:= F:= W:=

- If instead it reads:



Envelope FA I  
A:= D:= S:= R:=

...toggle the page by pressing the [select button] once more.

## 5.1.2 Envelope Amount FastAccess

**V:** changes the amount of the preconfigured input for the *Volume-envelope* at the *Volume-module*. As always, increasing the V:-value will prompt a '+' instead of the '=' symbol, while decreasing it displays a '-' sign.

No other parameter is affected.

**F:** changes the amount of the preconfigured input for the *Filter-envelope* at the *Filter-module*. Again, increasing the F:-value will prompt a + instead of the = symbol, while decreasing it displays a - sign.

No other parameter is affected.

**W:** changes the amounts of the preconfigured inputs for the *Wave-envelopes* at their respective *Wave-modules*. Both modules are affected simultaneously on a percentage basis to preserve their original relationship. Again, increasing the W:-value will prompt a '+' instead of the '=' symbol, while decreasing it displays a '-' sign.

No other parameter is affected.


## 5.1.3 Velocity FastAccess

### 5.1.3 Velocity FastAccess

Velocity FastAccess lets you quickly alter the velocity-sensitivity of all modules. Use it to quickly adjust a program to your playing-style or particular application.


To access the **Velocity FA** page:

- Select the Quick Edit mode using the [mode button].
- Press the [select button] labeled <FastAccess 2>. The display's page identification field should read:
- If instead it reads anything different, cycle through the pages by simply pressing the [select button] until the correct page appears.



```
Velocity FA  I
EV:= MC:= MS:=
```

**EV:** (Envelope Velocity) adjusts the *Envelope Velocity* parameters of all preconfigured envelope-modulation-inputs. These inputs are:



```
Velocity FA  I
EV:= MC:= MS:=
```

- ▶ Volume Envelope velocity
- ▶ Filter Envelope velocity
- ▶ Wave1 Envelope velocity
- ▶ Wave2 Envelope velocity

This parameter will actually always adjust the difference between the Envelope-amount and -velocity parameters. If no velocity is originally set, increasing the velocity a bit will result in an almost maximum setting for the corresponding Envelope-amount parameter, since the *EV:-value* always produces maximum envelope modulation; this modulation is split between the velocity- and amount-portion by the *EV:* parameter. If you want a smaller Amount-portion, use Envelope Amount FastAccess to diminish the desired amount-value.

## 5.1.2 Envelope Amount FastAccess

**MC:** (Modulation Control-inputs) adjusts the Amount-parameters of all *sidechain-modulation-inputs* that have *Velocity* routed to their respective *Control-inputs*. If no sidechain-modulator uses *Velocity* as a *Control-input* parameter, this FastAccess parameter is rendered useless.

All amounts affected will be scaled on a percentage basis to preserve the parameter's original relationship to each other.

**MS:** (Modulation Source-inputs) changes the Amount-parameters of all *routeable modulation-inputs* that use *Velocity* as the *Source-input*. This applies both to sidechain- and regular-modulators.

All amounts affected will be scaled on a percentage basis to preserve the parameter's original relationship to each other.


## 5.1.4 Aftertouch FastAccess

### 5.1.4. Aftertouch FastAccess

Aftertouch FastAccess allows you to quickly alter the aftertouch-sensitivity of all modules. Use it to quickly adjust a program to your playing-style or particular application.

To access the **Aftertouch FA** page:


- Select the Quick Edit mode using the [mode button].
- Press the [select button] labeled <FastAccess 2>. The display's page identification field should read:



```
Aftertouch FA I  
MC:= MS:=
```

- If instead it reads anything different, cycle through the pages by simply pressing the [select button] until the correct page appears.

**MC:** (Modulation Control-inputs) adjusts the Amount-parameters of all *sidechain-modulation-inputs* that have *Aftertouch* assigned to their respective *Control-inputs*. If no sidechain-modulator uses Aftertouch as a Control-input parameter, this FastAccess parameter is rendered useless.



```
Aftertouch FA I  
MC:= MS:=
```

All amounts affected will be scaled on a percentage basis to preserve the parameter's original relationship to each other.

**MS:** (Modulation Source-inputs) changes the Amount-parameters of all *routeable modulation-inputs* that use *Aftertouch* at the *Source-input*. This applies both to sidechain- and regular-modulators.

All amounts affected will be scaled on a percentage basis to preserve the parameter's original relationship to each other.


## 5.1.5 Mod Wheel FA

### 5.1.5 Mod Wheel FA

Mod Wheel FastAccess allows you to quickly alter the sensitivity to the modulation-wheel (or similar device that sends MIDI-controller #1 messages) of all modules concerned. Use it to quickly adjust a program to your playing-style or particular application.

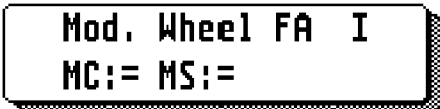
To access the **Mod Wheel FA** page:

- Select the Quick Edit mode using the [mode button].
- Press the [select button] labeled <FastAccess 2>. The display's page identification field should read:
- If instead it reads anything different, cycle through the pages by simply pressing the [select button] until the correct page appears.



```
Mod. Wheel FA I
MC:= MS:=
```

**MC:** (Modulation Control-inputs) adjusts the Amount-parameters of all sidechain-modulation-inputs that have *Mod Wheel* assigned to their respective *Control-inputs*. If no sidechain-modulator uses Mod Wheel as a Control-input parameter, this FastAccess parameter is rendered useless.



```
Mod. Wheel FA I
MC:= MS:=
```

All amounts affected will be scaled on a percentage basis to preserve the parameter's original relationship to each other.

**MS:** (Modulation Source-inputs) changes the Amount-parameters of all *routeable modulation-inputs* that use *Mod Wheel* at the *Source-input*. This applies both to sidechain- and regular-modulators.

All amounts affected will be scaled on a percentage basis to preserve the parameter's original relationship to each other.

Quick Edit

## 5.2 Envelope Macros

### 5.2 Envelope Macros

Envelope Macros copy predefined envelope shapes stored in ROM to the respective envelopes of the MicroWave. This offers a convenient way of setting up basic envelopes for a typical sound-type. Use the Envelope FA page or the respective individual parameters in Sound-edit mode to fine-tune the Macro envelopes to your liking.

There are 3 pages reserved for Envelope Macros:

- ▶ *Volume Env. Mac*
- ▶ *Filter Env. Mac*
- ▶ *Wave Env. Mac*

Each page copies the corresponding ROM-templates to their respective envelopes.

The basic user-interface functions are the same for all Envelope Macros:

- Select the Quick Edit mode using the [mode button].
- Press the [select button] labeled <ENV Macro> until the desired page is indicated in the page-identifier-field.
- Use the [alpha dial] to select the envelope-template of your liking.
- Press the [parameter/value button] to acknowledge and copy the template to the respective envelope.

Of course, only a single template can be used per envelope. Sorry, no mixing, merging or hidden hardware will alter that ever. If you need two envelopes at a single destination-module, use the routeable modulators.

Envelope templates will set all parameters of an envelope including the modulation-inputs.



## 5.2.1 Volume Envelope Macros

### 5.2.1 Volume Envelope Macros

These Macros use specific templates for the Volume-envelope. Of course, the destination-module of the Volume-envelope must not necessarily be the Volume-module; this depends on where the envelope is routed and applied. You can always connect it to the Volume-module by setting the velocity- or amount-parameters at the *Velocity FA* respectively in the *Envelope Amount FA* pages.

To access the **Volume Envelope Macro** page:

- Select the Quick Edit mode using the [mode button].
- Press the [select button] labeled <ENV Macro>. The display's page identification field should read:
- If instead it reads anything different, cycle through the pages by simply pressing the [select button] until the correct page appears.



VolumeEnv.Mac. I  
Click Organ

You can choose between the following Volume-envelope Macros using the [alpha dial]:

*Click Organ*

*Pipe Organ*

*Strings*

*Woodwind*

*Orch. Brass*

*Pop Brass*

*Piano*

*Pluck*

*Long Perc.*

*Medium Perc.*

*Short Perc.*

- Press the [parameter/value button] to copy the template to the Volume-envelope. All parameters previously set for the Volume-envelope will be lost.

## 5.2.2 Filter Envelope Macros

### 5.2.2 Filter Envelope Macros

These Macros use specific templates for the Filter-envelope. Of course, as with the Volume-envelope, the destination-module of the Filter-envelope doesn't necessarily need to be the Filter-module.

To access the **Filter Envelope Macro** page:

- Select the Quick Edit mode using the [mode button].
- Press the [select button] labeled <ENV Macro>. The display's page identification field should read:
- If instead it reads anything different, cycle through the pages by simply pressing the [select button] until the correct page appears.



FilterEnv.Mac. I  
Click Organ

You can choose between the following Filter-envelope Macros using the [alpha dial]:

*Click Organ*

*Pipe Organ*

*Strings*

*Woodwind*

*Orch. Brass*

*Pop Brass*

*Piano*

*Pluck*

*Long Perc.*

*Medium Perc.*

*Short Perc.*

*Delay Perc.*

- Press the [parameter/value button] to copy the template to the Filter-envelope. All parameters previously set for the Filter-envelope will be lost.

## 5.2.3 Wave Envelope Macros

### 5.2.3 Wave Envelope Macros

These Macros use specific templates for the Wave-envelope. Of course, as with the Volume-envelope, the destination-module of the Wave-envelope doesn't necessarily need to be a Wave-module.

To access the **Wave Envelope Macro** page:

- Select the Quick Edit mode using the [mode button].
- Press the [select button] labeled <ENV Macro>. The display's page identification field should read:
- If instead it reads anything different, cycle through the pages by simply pressing the [select button] until the correct page appears.



Wave Env. Mac. I  
Slow Attack

You can choose between the following Wave-envelope Macros using the [alpha dial]:

*Slow Attack*

*Slow Decay*

*ADSR Envelope*

*Inverse ADSR Envelope*

*Spit Valve*

*Slap Back*

*Wah-Wah*

*Single Echo*

*Repeat Echo*

*Long Loop*

- Press the [parameter/value button] to copy the template to the Wave-envelope. All parameters previously set for the Wave-envelope will be lost.

Quick Edit

## 5.3 Modulation Macros

### 5.3 Modulation Macros

Modulation Macros copy predefined modulation-settings stored in ROM to the respective modulation-inputs of the MicroWave. This offers a convenient way of setting up basic modulation effects.

There are 2 pages reserved for Modulation Macros:

- ▶ *Mod1 Mac*
- ▶ *Mod2 Mac*

Each page copies the corresponding ROM-templates to their respective routeable modulation-inputs. The templates of the Mod1 page will be copied to the respective parameters of the corresponding sidechain-modulators, while the templates of the Mod2 page will alter the regular modulation-inputs.

Depending on the modulation-effect, any one or all modulation-inputs might be used; any used modulation-input will completely erase the parameters set at this input.

The basic user-interface functions are similar to those of the Envelope Macros:

- Select the Quick Edit mode using the [mode button].
- Press the [select button] labeled <MOD Macro> until the desired page is indicated in the page-identifier-field.
- Use the [alpha dial] to select the modulation-template of your liking.
- Press the [parameter/value button] to acknowledge and copy the template to the respective envelope.

Of course, only a single template can be used per modulation page.

## 5.3.1 Modulation 1 Macros

### 5.3.1 Modulation 1 Macros

Modulation 1 Macros put certain predefined effect-types at your disposal. They will alter a number of audio- and modifier-modules depending on the effect.

To access the **Modulation 1 Macro** page:

- Select the Quick Edit mode using the [mode button].
- Press the [select button] labeled <MOD Macro>. The display's page identification field should read:



Mod 1 Macro I  
Mod easy Vib

- If instead it reads:



Mod 2 Macro I  
Mod easy Vib

...toggle the pages by simply pressing the [select button].

You can choose between the following Modulation 1 Macros using the [alpha dial]:

*Mod easy Vib* LFO 2 to Oscillator 1&2 only; use Modwheel

*Aft easy Vib* LFO 2 to Oscillator 1&2 only; use Aftertouch

*Mod acoust. Vib* LFO 2 to Oscillators, Volume, Filter; use Modwheel

*Aft acoust. Vib* LFO 2 to Oscillators, Volume, Filter; use Aftertouch

*Stereo Chorus* LFO 2 to Oscillator 2, Panning

*Vel Chorus* same as Stereo Chorus, but Velocity on Control inputs

*Vel Timbre Vib* LFO 2 to Wave1&2, Filter; use Velocity

- Select the desired effect with the [alpha dial].
- Press the [parameter/value button] to acknowledge.

Remember, you can use only one effect for Modulation 1 Macro at a time.

## 5.3.2 Modulation 2 Macros

### 5.3.2 Modulation 2 Macros

Modulation 2 Macros put certain predefined effect-types at your disposal. They will alter a number of audio- and modifier-modules depending on the effect.

To access the **Modulation 2 Macro** page:

- Select the Quick Edit mode using the [mode button].
- Press the [select button] labeled <MOD Macro>. The display's page identification field should read:



Mod 2 Macro I  
Mod easy Vib

- If instead it read:



Mod 1 Macro I  
Mod easy Vib

...toggle the pages by simply pressing the [select button].

## 5.3.2 Modulation 2 Macros

You can choose between the following Modulation 2 Macros using the [alpha dial]:

*Mod easy Vib* LFO 1 to Oscillator 1&2 only; use Modwheel

*Aft easy Vib* LFO 1 to Oscillator 1&2 only; use Aftertouch

*Delay easy Vib* LFO 1 to Oscillator 1&2 only, AR Env. used on LFO1

*Mod acoust. Vib* LFO 1 to Oscillators, Volume, Filter; use Modwheel

*Aft acoust. Vib* LFO 1 to Oscillators, Volume, Filter; use Aftertouch

*Del acoust. Vib* LFO 1 to Oscillators, Volume, Filter, AR Env. used on LFO1

*Tremolo* LFO 1 to Volume

*Pseudo Leslie* LFO 1 to Oscillator 1&2, Filter, Volume, Pan; use Mod Wheel for speed

*Auto Wah-Wah* LFO 1 to Filter Cut and Resonance

*Auto Panning* LFO 1 to Panning

*Vel Auto Pan* LFO 1 to Panning, LFO Level Mod Source: Velocity

*Echo* LFO 1 to Volume; shape: neg. saw; full Mod amount at Volume Module, slow speed, AD envelope; Vol-env-release 60, full Sustain

*Stereo Echo* as above, additionally LFO Env to Panning

- Select the desired effect with the [alpha dial].
- Press the [parameter/value button] to acknowledge.

Remember, you can use only one effect for Modulation 2 Macro at a time.





# Chapter 6

## Storage

- 6.1 Storing Sound-programs**
- 6.2 Copying Sound-programs**
- 6.3 Compare Sound -programs**
- 6.4 Recall Sound-programs**
- 6.5 Storing Multi-programs**
- 6.6 Copying Multi-programs**
- 6.7 Compare Multi-programs**
- 6.8 Recall Multi-programs**
- 6.9 Storing all edit-buffers**
- 6.10 Back-up to Card**
- 6.11 Load from Card**
- 6.12 Back-up to MIDI**
- 6.13 Load from MIDI**



## 6. Storage

## 6. Storage

Storing your programs is a vital part of programming your MicroWave. Anything that has not been stored, be it Sound-program or Multi-program, will be lost forever upon switching power off.

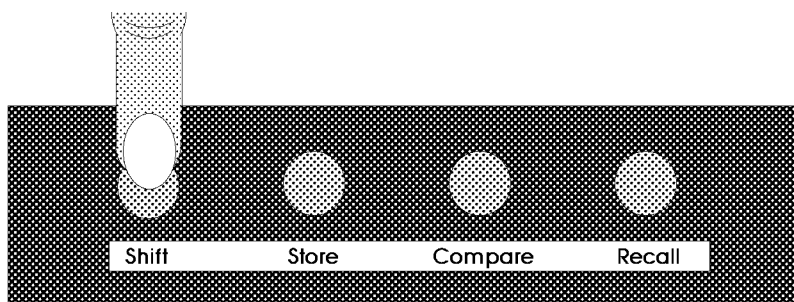
There are four levels to the store section. *Level one* deals with Sound- and Multi- programs; *level two* allows all eight edit-buffers to be stored to the respective locations simultaneously; *level three* deals with data transfer to and from a RAM Card; and *level four* allows data to be sent as MIDI System/Exclusive dump messages.

## 6.1 Storing Sound-program

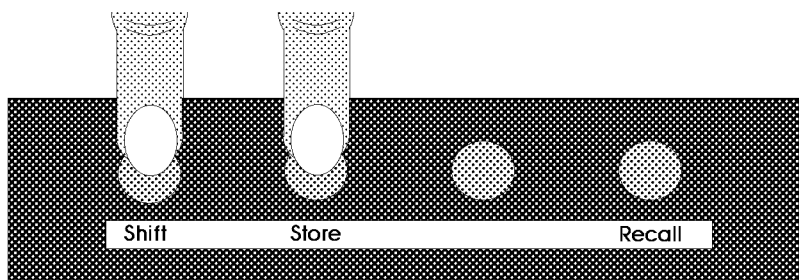
### 6.1 Storing Sound-programs

While in Single mode you can always store a Sound-program that still resides in an edit-buffer. It doesn't matter which page you are in. You can also store a Sound-program while in Multi mode if you have been editing one or more Sounds in Sound-edit mode or Quick Edit mode. However, you cannot store anything while in Global mode.

- Select the Sound-program you wish to store. Only the *selected Sound-program* will be stored.



- Press the [Shift button] and hold it.
  - Don't release the [Shift button] yet since otherwise it will act as a regular [select button] by itself and select a page. However, nothing will be lost if that happens, just begin again by holding the [Shift button].



- Press the [Store button] once. Now release the [Shift button].

## 6.1 Storing Sound-program

- Use the [alpha dial] to select the location you want to store this Sound-program into. If you're not sure where there is an empty location, you can still cancel the store command and check. You can use any internal or card location as a destination.
- To execute: press the [OK button] (also called the [parameter/value button]). The Sound-program is now stored.

The store command will be acknowledged by the following display:



Store Sound:A01  
Destination?A01

- After acknowledging the storage, the memory-status of the display will change to "I" or "C", depending on where the program has been stored.
- To cancel: press any other button besides [OK] to cancel the store command. The display will momentarily read:



Store cancelled.

## 6.2 Copying Sound-programs

### 6.2 Copying Sound-programs

This is essentially the same as storing a Sound-program. The only difference is that you are not necessarily storing an edited Sound-program, but moving one from internal- or card-memory to another location.

You may copy (and store) freely between internal- and card-memory. Follow the steps explained in topic 1.

## 6.3 Compare Sound-programs

### 6.3 Compare Sound-programs

With the Compare function you can compare the edited version of a Sound-program (whose memory status will be displayed as "E") with the original sound in memory. This function only lets you compare the currently edited Sound-program with its original.

Comparing will actually swap the two programs temporarily. Therefore, so as not to destroy any vital information, you will not be able to edit while comparing; you have to swap again to gain access to the program which is in the Edit-buffer, since this is the only place you can edit a program.

However, you can Recall a swapped program (see below). In this case the edited version residing in the Edit-buffer will be lost. You can always know whether you are listening to the original or edited version of a sound by watching the memory-status field in the upper-right corner of the display:

■ "E" represents the program residing in the Edit-buffer. This indicates that the program has been edited and is different than what is stored in its memory location.

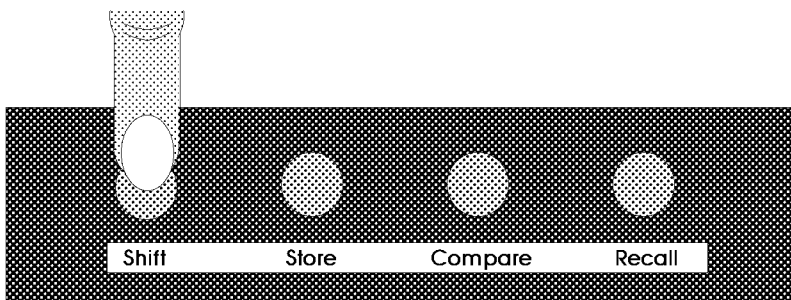


Sound Prog:A01 E  
Sound Init

■ "S" indicates a swap (Compare). This means that you are now hearing the original version stored in memory. Currently you cannot edit this sound.

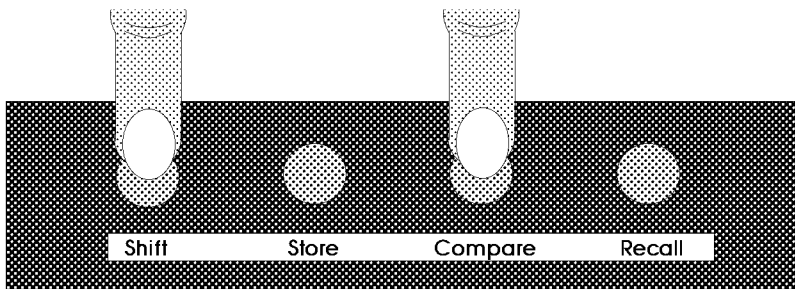


Sound Prog:A01 S  
Sound Init



- Press the [Shift button] and hold it.
  - Don't release the [Shift button] since otherwise it will act as a regular [select button] and will select a page.

## 6.3 Compare Sound-programs



- Press the [Compare button] once. Release the [Shift button].
- You now have swapped the program with it's original version. The memory-status field of the display will read "S".
- To swap back to the edited version, simply repeat the same procedure. The memory-status field will display an "E", indicating that you are listening to the edited version.

Sound Prog:A01 S  
Sound Init

Sound Prog:A01 E  
Sound Init



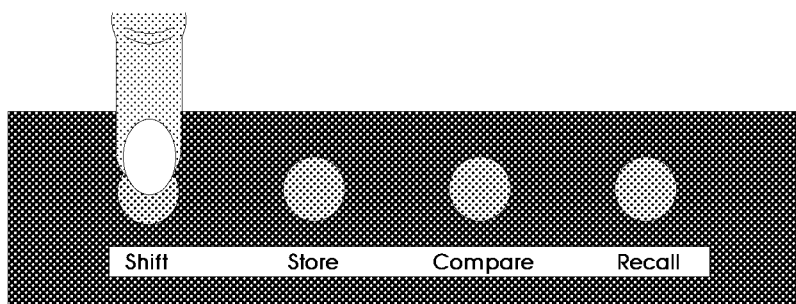
## 6.4 Recall Sound-programs

### 6.4 Recall Sound-programs

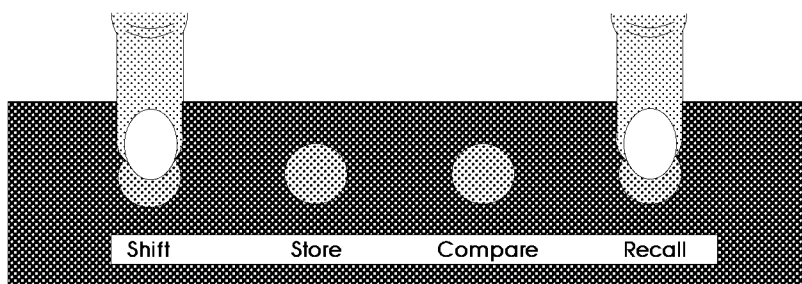
Recalling a Sound-program will revert an edited version to the original version stored in memory. You can only recall a Sound-program residing in one of the eight available Edit-buffers. You may recall a program regardless of it's compare-status (ie even while you are comparing an edited sound to its original).

Recalling a Sound-program will erase the edited version. So be careful when recalling a program, especially since no warning will appear on the display.

On the other hand, you should always recall a Sound-program after you have stored it. Otherwise it will remain in the Edit-buffer and might prevent you from editing another program in case the Edit-buffer is full. Remember, there are eight Edit-buffers in all, dedicated to Sound-programs; if these are all occupied, you will not be able to edit another Sound-program.



- Press the [Shift button] and hold it.  
→ Don't let go of the [Shift button] since otherwise it will act as a regular [select button] and will select a page.



- Press the [Recall button] once. Release the [Shift button].
- You now have recalled the Sound-program's original version. The memory-status field of the display will read "I" or "C", according to where the program originates.

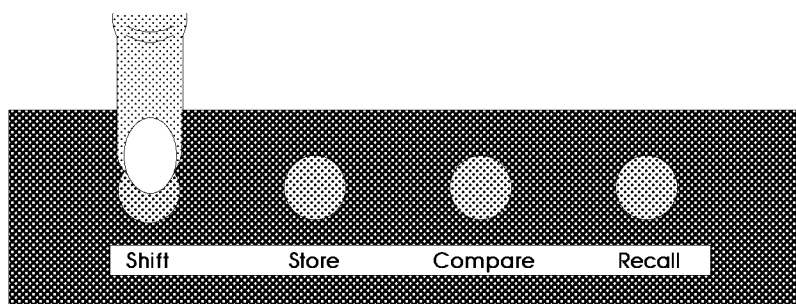
You cannot undo a recall.

## 6.5 Storing Multi-programs

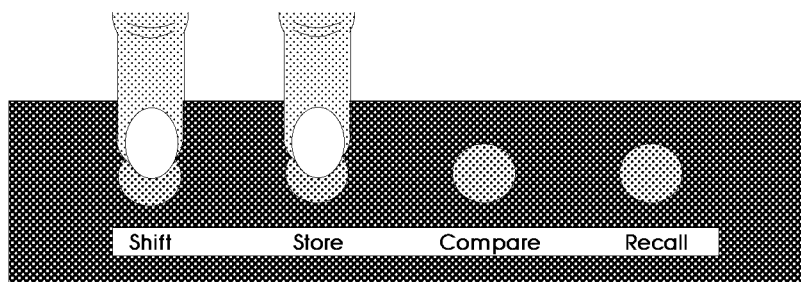
### 6.5 Storing Multi-programs

While you are in Multi mode you can always store a Multi-program from within any page in Play mode. Storing a Multi-program is very similar to storing a Sound-program.

- Select the Multi-program you wish to store. Only the *selected Multi-program* will be stored, no matter if it is in an Edit-buffer or not.



- Press the [Shift button] and hold it.
  - Don't release the [Shift button] since otherwise it will act as a regular [select button] and select a page. However, nothing will get lost if that happens, just repeat the process again by holding the [Shift button].



- Press the [Store button] once. Now release the [Shift button].
- Use the [alpha dial] to set the destination you want to store this Sound-program. If you're not sure where there is an empty location, you can still cancel the storage and check. Use any internal or card location as destination.

Store Multi:A01  
Destination?A01

## 6.5 Storing Multi-programs

To execute: press the [OK button] (which is the same as the [parameter/value button]).The Sound-program is now stored.

The store command will be acknowledged by the following display:



Store complete.\_

- After acknowledging the storage, the memory-status of the display will change to "1" or "C", depending where the program has been stored.
- To cancel: press any other but the [OK] button to cancel storing the Multi-program. The display will momentarily read:



Store cancelled.

## 6.6 Copying Multi-programs

### 6.6 Copying Multi-programs

This is the same procedure as storing a Multi-program. The only difference, as in copying a Sound-program, is that you select a Multi-program from internal- or card-memory rather than an edited one, and store it in another location.

## 6.7 Compare Multi-programs

### 6.7 Compare Multi-programs

With the Compare function you can compare the edited version of a Multi-program (whose memory status will be displayed as "E") with its original stored version in memory. You can only compare the currently edited Multi-program with its original.

Comparing will actually swap the two programs. Therefore, so as not to destroy any vital information, you will not be able to edit while comparing; you have to swap again to gain access to the program which is in the Edit-buffer, since this is the only place you can edit a program.

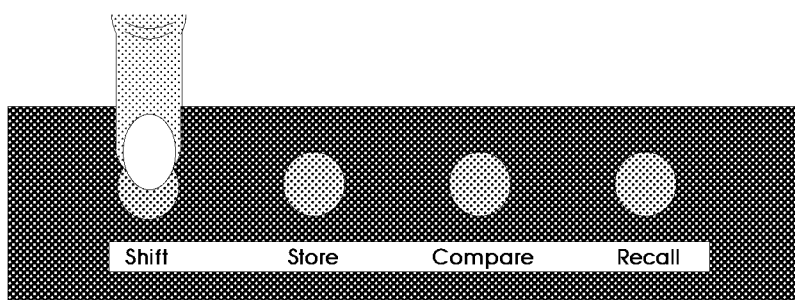
You can always know if you are listening to the original or edited version by watching the memory-status field in the upper-right corner of the display.

▣ "E" stands for the program residing in the Edit-buffer. That is the edited version which you may edit further.

```
Multi Prog:A01 E
Multi Init
```

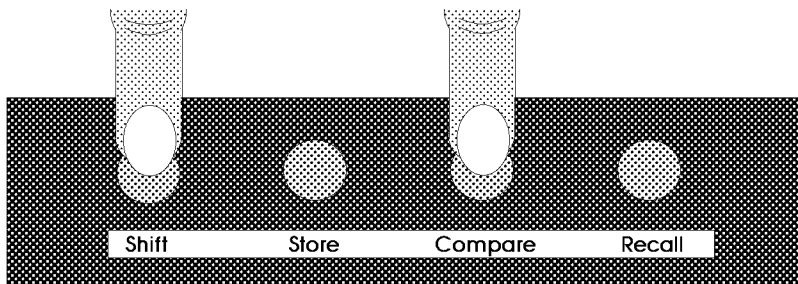
▣ "S" indicates a swap. This means, you are now listening to the original version stored in memory. Currently you cannot edit the Multi-program.

```
Multi Prog:A01 S
Multi Init
```



- Press the [Shift button] and hold it.
  - Don't let go of the [Shift button] since otherwise it will act as a regular [select button] and select a page.

## 6.7 Compare Multi-programs



- Press the [Compare button] and let go. Release the [Shift button] also.
- You now have swapped the edited program with its original version. The memory-status field of the display will read "S".
- To swap back to the edited version, simply repeat the same procedure. The memory-status field will display an "E", assuring that you are now listening to the edited version.

```
Multi Prog:A01 S  
Multi Init
```

```
Multi Prog:A01 E  
Multi Init
```

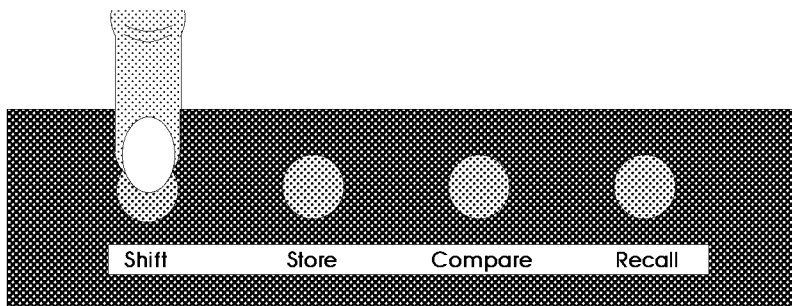
## 6.8 Recall Multi-programs

### 6.8 Recall Multi-programs

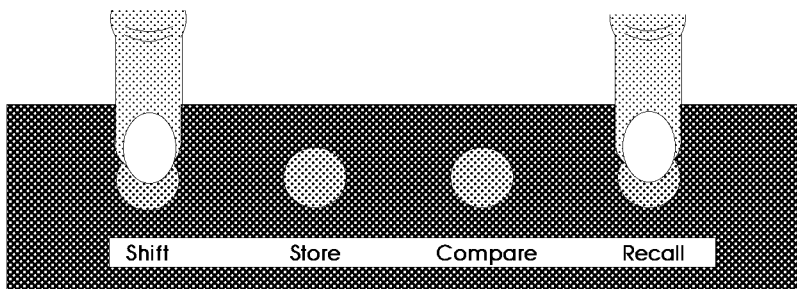
Recalling a Multi-program will revert an edited version to the original version stored in memory. You can only recall the current Multi-program residing in the Multi-program Edit-buffer. You may recall a program regardless of its compare-status (ie even if you are currently comparing the edited sound to the original).

Recalling will erase the edited version permanently. So be careful when recalling a program, especially since no warning will appear upon recalling a program.

On the other hand, you should always recall a program after you have stored it. Otherwise it will remain in the Edit-buffer and might prevent you from editing another program in case the Edit-buffer is full. Remember, there is only a single Edit-buffer for Multi-programs.



- Press the [Shift button] and hold it.
  - Don't let go of the [Shift button] since otherwise it will act as a regular [select button] and select a page.



- Press the [Recall button] and let go. Release the [Shift button] also.
- You now have recalled the program's original version. The memory-status field of the display will read "1" or "C", according to where the original version is stored.

You cannot undo a recall.

## 6.9 Storing all edit-buffers

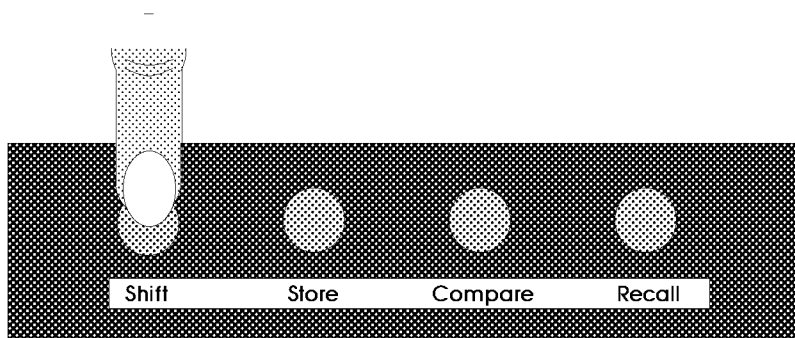
### 6.9 Storing all edit-buffers

Once you start editing sounds, you will be able to leave an edited sound temporarily to edit another sound, and another. This is possible because of the multiple edit-buffers, which store your edits for up to eight Sound-programs and one Multi-program until you store them into memory. This is convenient when working on a Multi-program because you can leave any instrument in a temporary edit state while fine-tuning the other sounds. Once you have used up all eight edit-buffers, however, the display will read the following:

**Sorry,  
Editorbuffer full!**

You will not be able to edit any other sounds until you have stored your edits.

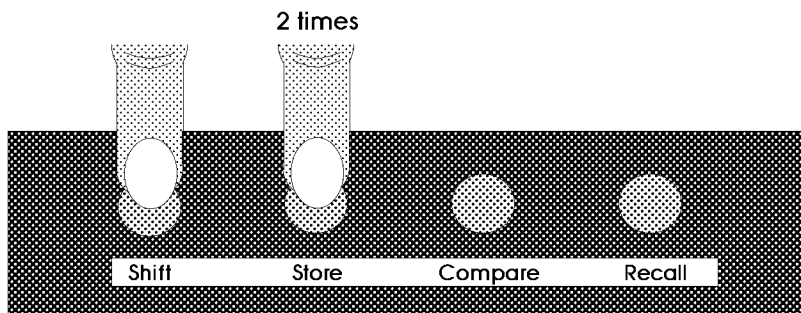
Global Store mode allows you to store all eight Sound-program edit-buffers and the one Multi-program edit-buffer to their respective destinations simultaneously. You can access this mode from within any page. Be careful when using this function, for it will store all programs to their current locations, and will overwrite the original versions of any sounds you have edited thus far. Therefore, if you wish to keep the original programs, you must store each edited program individually so as to place them in different locations.



- Press the [Shift button] and hold it.
  - Don't let go of the [Shift button] since otherwise it will act as a regular [select button] and select a page.



## 6.9 Storing all edit-buffers



- Press the [Store button] *twice* and let go. Release the [Shift button] also. the display will read the following:

```
Global Store Md.  
Store all edits?
```

You are now in the *second store level*.

- To execute the store, press the [OK button] once. The display will read:

```
Global Store Md.  
updating... ok
```

- You now have stored all eight Sound-program edit-buffers. The memory-status field of the display will read "I" or "C" for each of the edited Sound-programs, according to where their original versions were stored. All edit-buffers will now be empty, allowing to immediately edit any Sound-program.

- To cancel the store function, press any other button on the front panel except the [OK button]. The display will read:

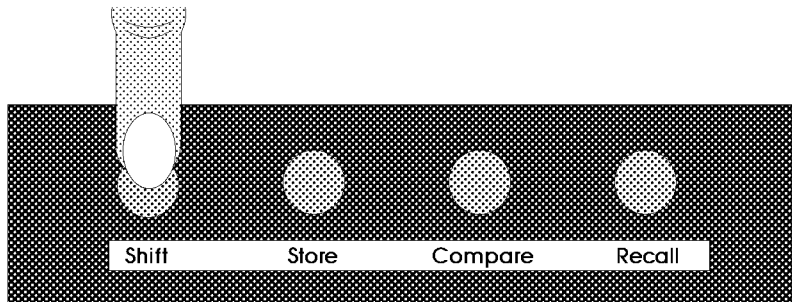
```
Datatransf. Mode  
Init Sound: A01?
```

- After storing all edit-buffers using this function, you need not recall them again, since this automatically happens after the store command has been executed.

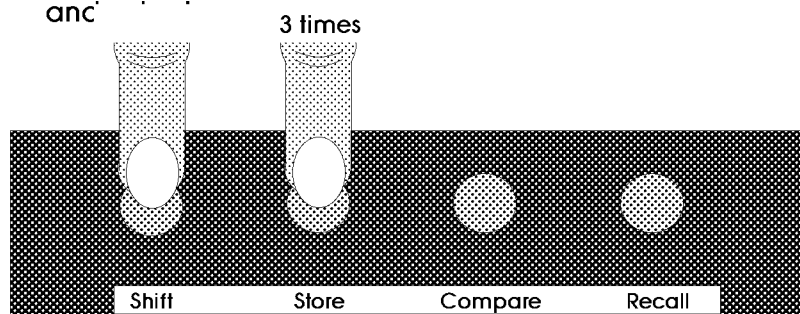
## 6.10 Back-up to Card

### 6.10 Back-up to Card

You can copy individual programs to and from a card at any time. To back-up the entire memory contents to the card, use the following procedure:



- Press the [Shift button] and hold it.  
→ Don't let go of the [Shift button] since otherwise it will act as a regular [select button] and



- Press the [Store button] *three times* and let go. Release the [Shift button] also. the display will read the following:

Datatransf. Mode  
Init Sound: A01?

You are now in the *third store level*.

- You now can select one of the following storage-alternatives using the [alpha dial]:

□ **Init Sound: A01?** will initialize the currently selected Sound-program, and will reset all its parameters to their default settings. Use this function when you wish to program a new sound from scratch.

□ **All to Card?** will dump the entire memory-contents to the card. Any programs currently residing on that card will be overwritten and lost.

## 6.10 Back-up to Card

- **Sounds to Card?** will dump all Sound-programs to the card. Any Sound-programs currently residing on that card will be overwritten and lost.
  
- **Multis to Card?** will dump all Multi-programs to the card. Any Sound-programs currently residing on that card will be overwritten and lost.
  
- **Tables to Card?** will dump all Tables and Maps to the card. Any Tables and Maps currently residing on that card will be lost.
  
- To execute: press the [OK button] (which corresponds to the [parameter/value button]) and the transfer will begin.
  
  
- To cancel: press any other but the [OK] button to cancel the transfer.

## 6.11 Load from Card

### 6.11 Load from Card

From within the same store level as above, the following options are also available by turning the [alpha dial]:

- ❑ **Card to Int.?** will dump the entire memory—contents of the card into internal memory. Any programs currently residing in the MicroWave will be overwritten and lost.
- ❑ **Sounds to Int.?** will dump all the Sound—programs of the card into internal memory. Any Sound—programs currently residing in the MicroWave will be overwritten and lost.
- ❑ **Multis to Int.?** will dump all the Multi—programs of the card into internal memory. Any Multi—programs currently residing in the MicroWave will be overwritten and lost.
- ❑ **Tables to Int.?** will dump all Tables and Maps of the card into internal memory. Any Sound—programs currently residing in the MicroWave will be overwritten and lost.
- To execute: press the [OK button] (which corresponds to the [parameter/value button]) and the transfer will begin.
  
- To cancel: press any other but the [OK] button to cancel the transfer.

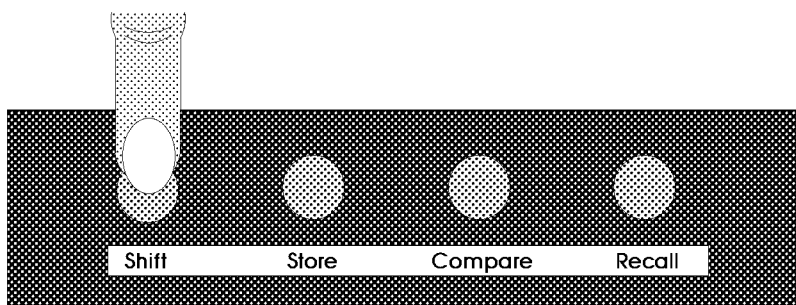
## 6.12 Back-up to MIDI

### 6.12 Back-up to MIDI

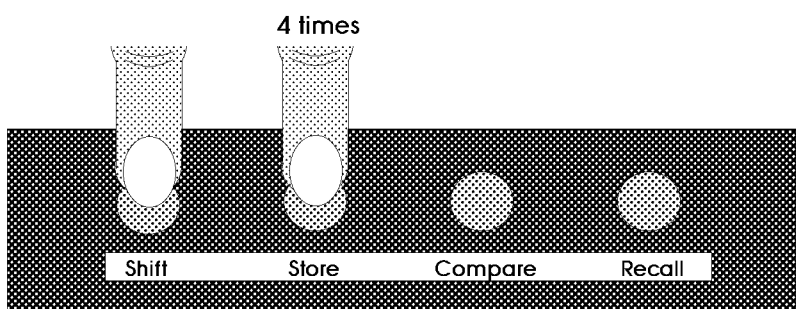
Instead of backing up your sounds to the card, you may alternatively back them up to MIDI. To do that, you must connect a MIDI system-exclusive librarian to the MIDI-Out port of the MicroWave that understands the MicroWave dump-format. Most generic librarians should be able to do that. If in doubt, try it out.

If you must allocate a certain memory-space at the receiver, you should use a generous 40 kByte buffer.

You can copy individual programs to and from any MIDI data storage device at any time. To back-up the entire memory contents to a MIDI storage device, use the following procedure:



- Press the [Shift button] and hold it.  
→ Don't let go of the [Shift button] since otherwise it will act as a regular [select button] and select a page.



- Press the [Store button] *four times* and let go. Release the [Shift button] also. the display will read the following:


```
SysEx Transfer
Dump Sound: A01?
```

You are now in the *fourth store level*.

## 6.12 Back-up to MIDI

- You now can select one of the following storage-alternatives using the [alpha dial]:
  - **DumpSound:** AOI? will dump the currently selected Sound-program to MIDI.
  - **Dump all Int.?** will dump the entire memory-contents to MIDI.
  - **Dump Sounds?** will dump all Sound-programs to MIDI.
  - **Dump Multis?** will dump all Multi-programs to MIDI.
  - **Dump Tables?** will dump all Tables and Maps to MIDI.
- To execute: press the [OK button] (which corresponds with the [parameter/value button]) and start the transfer.

While the data is being sent, the display will momentarily read:



transmitting,  
please wait...

When the MicroWave is finished sending, the display will return to the page you were currently in before initiating the MIDI transfer.

- To cancel: press any other but the [OK] button to cancel the transfer.

## 6.12 Back-up to MIDI

Another way to execute the dump-to-MIDI command than from the MicroWave's front panel, is by sending a MIDI dump-request message to the MicroWave. Here is a list of request messages that the MicroWave will respond to, and their dump-request message formats:

(Note: all requests are in Hex!)

### □ **Version Number Request**

(requests software version number and date)

F0,3E,00,device ID\*,00,00,F7

### □ **Device Status Request**

(requests device parameters)

F0,3E,00,device ID\*,01,00,F7

### □ **Sound Dump Request**

(requests current Sound-program; in Multi-mode, the Sound-program of the currently selected instrument is sent)

F0,3E,00,device ID\*,02,00,F7

### □ **Multi-Program Dump Request**

(requests current Multi-program; however, corresponding Sound-programs are not sent)

F0,3E,00,device ID\*,03,00,F7

### □ **Tuning Table Dump Request**

(requests a user-tuning table; specify table number in request)

F0,3E,00,device ID\*,06,00,F7

### □ **Velocity Table Dump Request**

(requests a user-velocity table; specify table number in request)

F0,3E,00,device ID\*,07,00,F7

### □ **Sound PC Map Dump Request**

(requests the Sound-program change map)

F0,3E,00,device ID\*,08,00,F7

### □ **Multi PC Map Dump Request**

(requests the Multi-program change map)

F0,3E,00,device ID\*,09,00,F7



## 6.12 Back-up to MIDI

- |  |                              |
|--|------------------------------|
| <b>□ Sound-program Bank Dump Request</b><br>(requests entire Sound-program bank)                                     | F0,3E,00,device ID*,10,00,F7 |
| <b>□ Multi-program Bank Dump Request</b><br>(requests entire Multi-program bank)                                     | F0,3E,00,device ID*,11,00,F7 |
| <b>□ Table Dump Request</b><br>(requests all tables and maps)  | F0,3E,00,device ID*,12,00,F7 |
| <b>□ Card Dump Request</b><br>(requests all data resident in card memory)  | F0,3E,00,device ID*,14,00,F7 |
| <b>□ Arrangement Dump Request</b><br>(requests the current Multi-program<br>INCLUDING all associated Sound-programs) | F0,3E,00,device ID*,15,00,F7 |

**□ device ID\*:** Here you must set the Device number to be identical to what is set in your MicroWave. However, be aware that the MicroWave displays Device-numbers in decimal, while a dump-request will need this information in Hex. In most cases, though, a Device number of 000 in decimal does the job, which correspond to 00 in Hex. If in doubt, ask a friend who has computer-knowledge about Hex-numbers.



### 6.13. Load from MIDI

You may send any dump to the MicroWave via MIDI. If the device-number is correct, the MicroWave will receive the incoming system-exclusive data and store the data according to it's type.

If you want to transfer data from one MicroWave to another, set their Device numbers to the same value.

If you send a MIDI dump from a MicroWave with a different Device number set, be sure to set the correct Device number at the librarian before sending it to the MicroWave. If you only own a generic librarian that does not support the MicroWave particularly, you must change the Device number at the MicroWave until it receives the data. Although there are theoretically 127 possible chances, most likely a Device number 000 will do the job.

You can also initiate the same dumps from the MicroWave front panel by sending a 'request'. This comes in handy when you want to exchange data between two MicroWaves. Simply connect the MIDI-Out of the receiver to the MIDI-in of the transmitter and vice-versa.

From within the same store level as above, the following options are also available by turning the [alpha dial]:

□ **Request Sound?** will send a request message to a connected MicroWave with the same device number telling it to send the currently selected Sound-program or Multi-program depending on the mode the MicroWave is currently set to.

- A received **Single Sound-program Dump** will be put into a free Edit-buffer. If there are no free Edit-buffers left, it will erase one of them. If you want to keep the received Sound-program, you must Store it manually.
- A received **Single Multi-program Dump** will be put into the Multi-program Edit-buffer. Any program residing in that buffer will be deleted. If you want to keep the received Multi-program, you must store it manually. Remember, though, that no Sound-programs are stored with a Multi-program, only the Sound-program locations, so the resulting sound might be surprising.

## 6.13 Load from MIDI

If you want practicable result, request an Arrangement dump via transfer the associated sound-programs together with the actual Multi-program.

□ **Request All?** will send a request message to a connected MicroWave with the same device number telling it to send its entire memory contents. The received data will replace the entire memory contents of the receiving MicroWave.

→ When a MIDI dump of **all memory** is received it will replace the entire memory-contents of the MicroWave. Once all data has been received, the display will momentarily read:



WELCOME TO THE  
MICROWAVE

...and will then display the first Sound-program location:



Sound Prog:001 I  
Sound Init

Thus you know that you have just blown your killer Alphorn patch forever.

□ **Request Sounds?** will send a request message to a connected MicroWave with the same device number telling it to send all of its Sound-programs. The received data will replace all the Sound-programs of the receiving MicroWave.

→ A received **Sound-bank** from a MIDI Dump will replace all Sound-programs of the MicroWave. The display will not indicate when a Sound-bank has been received.

□ **Request Multis?** will send a request message to a connected MicroWave with the same device number telling it to send all of its Multi-programs. The received data will replace all the Multi-programs of the receiving MicroWave.

→ A received Multi-bank from a MIDI Dump will replace all Multi-programs of the MicroWave. The display will not indicate when a Multi-bank has been received.

## 6.13 Load from MIDI

□ **Request Tables?** will send a request message to a connected MicroWave with the same device number telling it to send all of its Tables and Maps. The received data will replace all of the Tables and Maps of the receiving MicroWave.

→ Tables and Maps from a MIDI Dump will replace all Tables and Maps of the MicroWave. The Display will not indicate when Tables and Maps have been received.

It does not matter which page you are in when sending a MIDI dump to the MicroWave. The data is simply received.



# Chapter 7

Sound Samples

**7.1 The Sound - init Program**

**7.2 Analog Brass**

**7.3 Rolling Glass**



## 7 Sound Samples



## 7 Sound Samples

### Sample Programs

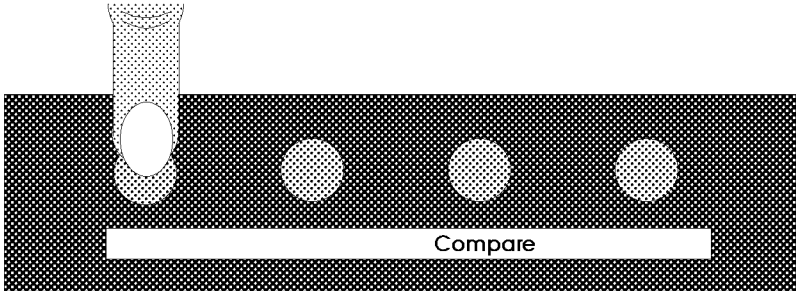
This chapter lists some Sample programs to give you an idea how you might program the MicroWave. This, of course, is just a very rough introduction; with a little practice you will be able to program sounds of truly outstanding quality.



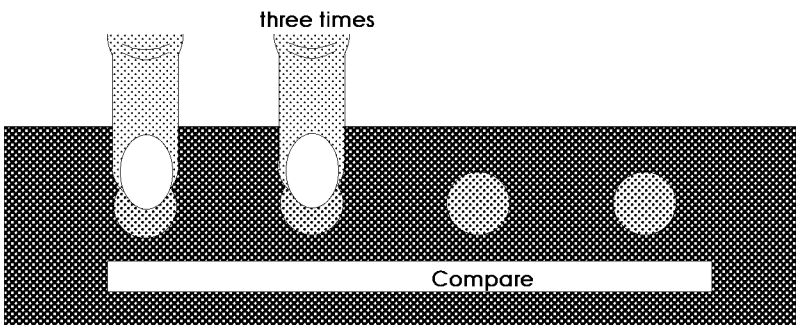
# 7.1 The Sound-init Program

## 7.1 The Sound-init Program

This program comes up whenever you do a sound-init to initialize all parameters. To recall the sound-init, follow these steps:



- Press and hold the [Shift button].



- While holding the [Shift button], press the [Store button] *three times*. Now release the [Shift button]. The display will read:

Datatrnsf. Mode  
Init Sound: A01?

- The Sound-Init program will automatically be placed in the currently selected location.

- To acknowledge the recall, press the [OK button], also known as the [parameter/value button]. The display will momentarily read:

Datatrnsfer  
completed

You have now initiated the sound init program. These are the default settings for the following parameters:



# 71 The Sound-init Program

## Oscillator 1 Parameters set to value

---

Octave	(-2..0..+2)	+0
Semitone	(0..12)	00
Detune	(-64..0..+63)	+00
Bend Range	(0..12 semitones)	02
Pitchmode	(normal/fixed)	<i>normal</i>
Mod1 Source	(table)	<i>LFO 1</i>
Mod1 Control	(table)	<i>Mod Wheel</i>
Mod1 Amount	(-64..0..+63)	+20
Mod2 Source	(table)	<i>LFO 1</i>
Mod2 Amount	(-64..0..+63)	+00
Mod2 Quantize	(Off..7)	<i>Off</i>

## Oscillator 2 Parameters

---

Octave	(2..0..+2)	+0
Semitone	(0..12)	00
Detune	(-64..0..+63)	+00
Bend Range	(0..12 semitones)	02
Pitchmode	(normal/fixed)	<i>normal</i>
Mod1 Source	(table)	<i>LFO 1</i>
Mod1 Control	(table)	<i>Mod Wheel</i>
Mod1 Amount	(-64..0..+63)	+20
Mod2 Source	(table)	<i>LFO 1</i>
Mod2 Amount	(-64..0..+63)	+00
Mod2 Quantize	(Off..7)	<i>Off</i>
LinkOscMod	(off/on)	<i>Off</i>

## Wave 1 Parameters

---

Wavetable	(R1..32/I33..44/R45..56)	<i>ROI</i>
Startwave	(0..60/Tri/Squ/Saw)	00
Startsample	(free..1..127)	001
Envelope Amount	(-64..0..+63)	+00
Envelope Velocity	(-64..0..+63)	+00
Keytrack Amount	(-64..0..+63)	+00
Mod1 Source	(table)	<i>LFO 1</i>
Mod1 Control	(table)	<i>Mod Wheel</i>
Mod1 Amount	(-64..0..+63)	+00
Mod2 Source	(table)	<i>LFO 1</i>
Mod2 Amount	(-64..0..+63)	+00
Mode	(stepped/smooth)	<i>smooth</i>



# 71 The Sound-init Program

## Wave 2 Parameters

---

Startwave	(0..60/tri/squ/saw)	00
Startsample	(free..1..127)	001
Envelope Amount	(-64..0..+63)	+00
Envelope Velocity	(-64..0..+63)	+00
Keytrack Amount	(-64..0..+63)	+00
Mod1 Source	(table)	LFO 1
Mod1 Control	(table)	Mod Wheel
Mod1 Amount	(-64..0..+63)	+000
Mod2 Source	(table)	LFO 1
Mod2 Amount	(-64..0..+63)	+000
Mode	(stepped/smooth)	smooth
LinkWaveMod	(off/on)	off

## Volume Parameters

---

Wave1 Vol	(0..7)	4
Wave2 Vol	(0..7)	4
Noise Vol	(0..7)	0
Sound Volume	(1..127)	090
Envelope Amount	(-64..0..+63)	+63
Envelope Velocity	(-64..0..+63)	+00
Keytrack Amount	(-64..0..+63)	+00
Mod1 Source	(table)	LFO 1
Mod1 Control	(table)	Mod Wheel
Mod1 Amount	(-64..0..+63)	+00
Mod2 Source	(table)	LFO 1
Mod2 Amount	(-64..0..+63)	+00

## Filter Parameters

---

Cutoff	(0..127)	127
Resonance	(0..127)	000
Envelope Amount	(-64..0..+63)	+00
Envelope Velocity	(-64..0..+63)	+00
Keytrack Amount	(-64..0..+63)	+00
Mod1 Source	(table)	LFO 1
Mod1 Control	(table)	Mod Wheel
Mod1 Amount	(-64..0..+63)	+00



# 71 The Sound-init Program

Mod2 Source (table) *LFO 1*  
 Mod2 Amount (-64..0..+63) +00

RMod Source (table) *LFO 1*  
 RMod Amount (-64..0..+63) +00

## Volume Envelope Parameters

---

Attack (0..127) 000  
 Decay (0..127) 032  
 Sustain (0..127) 127  
 Release (0..127) 016

Attack Mod Source (table) *Velocity*  
 Attack Mod Amount (-64..0..+63) +00

Decay Mod Source (table) *Keytracking*  
 Decay Mod Amount (-64..0..+63) +00

Sustain Mod Source (table) *Affertouch*  
 Sustain Mod Amount (-64..0..+63) +00

Release Mod Source (table) *Rel*  
*Velocity*  
 Release Mod Amount (-64..0..+63) +00

## Filter Envelope Parameters

---

Delay (0..127) 000  
 Attack (0..127) 000  
 Decay (0..127) 032  
 Sustain (0..127) 127  
 Release (0..127) 016

Delay Mod Source (table) *LFO 1*  
 Delay Mod Amount (-64..0..+63) +00

Attack Mod Source (table) *Velocity*  
 Attack Mod Amount (-64..0..+63) +00

Decay Mod Source (table) *Keytracking*  
 Decay Mod Amount (-64..0..+63) +00

Sustain Mod Source (table) *Affertouch*  
 Sustain Mod Amount (-64..0..+63) +00

Release Mod Source (table) *Rel*  
*Velocity*  
 Release Mod Amount (-64..0..+63) +00



# 71 The Sound-init Program

## Wave Envelope Parameters

Time1	(0..127)	032
Level1	(0..127)	127
Time2		032
Level2		064
Time3		016
Level3		000
Time4		000
Level4		000
Time5		000
Level5		000
Time6		000
Level6		000
Time7		000
Level7		000
Time8		000
Level8		000
Time Mod Source	(table)	<i>Keytracking</i>
Time Mod Amount	(-64..0..+63)	+00
Level Mod Source	(table)	<i>Aftertouch</i>
Level Mod Amount	(-64..0..+63)	+00
Key Off Point	(1..8)	2
Loop Start Point	(1..8)	1
Loop Mode	(off/on)	<i>off</i>

## LFO 1 Parameters

Rate (1..127)	050	
Shape	(sin/saw/ pulse/random)	<i>sin</i>
Symmetry	(-64..0..+63)	+00
Humanize	(off..7)	<i>off</i>
Rate Mod Source	(table)	<i>Velocity</i>
Rate Mod Amount	(-64..0..+63)	+00
Level Mod Source	(table)	<i>max.</i>
Sync (off/on)	<i>off</i>	
Delay	(off/retrigger/ 1..126)	<i>off</i>
Attack	(0..127)	000
Decay	(Off/1..127)	<i>off</i>

## LFO 2 Parameters

Rate	(1..127)	050
------	----------	-----



# 71 The Sound-init Program

Shape	(sin/saw/ pulse/random)	<i>sin</i>
Symmetry	(-64..0..+63)	<i>+00</i>
Humanize	(off..7)	<i>off</i>

## Panning/Glide Parameters

---

Panning	(L63..M..R64)	<i>MOO</i>
Pan Mod Source	(table)	<i>LFO 1</i>
Pan Mod Amount	(-64..0..+63)	<i>+00</i>

Glide	(off/Gliss/Porta MidiPorta/MIDIgliss)	<i>off</i>
Glide Rate	(0..127)	<i>032</i>
Glide Mode	(Time/Distance)	<i>Distance</i>

Temperment	(In+/In-/rn1/rn2/TT1..4)	<i>In+</i>
------------	--------------------------	------------

## Sound Name

---

Sound Name	(16 Digits)	<i>Sound Init</i>
------------	-------------	-------------------

## Modulation-modifier-modules table

---

LFO1  
LFO2  
Vol Envelope  
Filt Envelope  
Wave Envelope  
LFO Envelope

Keytrack  
Velocity  
ReleaseVelocity  
Aftertouch  
PolyPressure  
PitchBend  
ModWheel  
SustainPedal  
VolumeControl  
PanControl  
BreathControl  
ControllerW  
ControllerX  
ControllerY  
ControllerZ

max.  
min.



## 7.2 Analog Brass

There are a lot of parameters (146!), and it's likely you will want to alter most of them. However, these settings should represent a good starting point for developing your own Sound-programs.

### 7.2 Analog Brass

The MicroWave is very well suited to do fantastic analog sounds. Here is a sample-program including some comments of why certain parameters are set a certain way.

Start off by recalling a sound-init program as described before.

Then work your way through by assigning each parameter the following values:

Oscillator 1 Parameters		set to value
Octave	(-2..0..+2)	+0
Semitone	(0..12)	00
Detune	(-64..0..+63)	+00

This sets the basic octave range. Change the Octave parameter if you need to change ranges on your keyboard.

Bend Range	(0..12 semitones)	02
------------	-------------------	----

This allows a pitchbend of a whole step; if you like a different value, just program whatever you need.

Pitchmode	(normal/fixed)	<i>normal</i>
-----------	----------------	---------------

Normal allows regular playing on the keyboard; fixed would be useful for an percussion- or effects-sound.

Mod1 Source	(table)	<i>LFO 2</i>
Mod1 Control	(table)	<i>Mod Wheel</i>
Mod1 Amount	(-64..0..+63)	+20

This is a useful setting for basic Vibrato. If you like more dramatic effects, increase the Amount. If you dislike the Mod Wheel to control Vibrato, either use another MIDI-controller or set the Control input to max for a constant vibrato. You may want to use the Filter-envelope to control LFO 2.



## 7.2 Analog Brass

Why use LFO 2 instead of LFO 1? Because we will use the AD envelope of LFO 1 for a pitch-warp effect at oscillator 2, which excludes LFO 1 from regular modulation tasks.

Mod2 Source	(table)	<i>LFO 1</i>
Mod2 Amount	(-64..0..+63)	<i>+00</i>
Mod2 Quantize	(Off..7)	<i>Off</i>

Modulation-input 2 is not used.

### Oscillator 2 Parameters

---

Octave	(-2..0..+2)	<i>+0</i>
Semitone	(0..12)	<i>00</i>
Detune	(-64..0..+63)	<i>+06</i>

A slight Detune-value will fatten the sound. If you like instant lush-jazz chords, set Semitone to 7 and play some major and minor chords. You will get an interesting result.

Bend Range	(0..12 semitones)	<i>02</i>
Pitchmode	(normal/fixed)	<i>normal</i>

Mod1 Source	(table)	<i>LFO 2</i>
Mod1 Control	(table)	<i>Mod Wheel</i>
Mod1 Amount	(-64..0..+63)	<i>+14</i>

This is the same mod-wheel setting as used for <sup>Envelope</sup>Oscillator 1. To achieve a better effect, Mod1 Amount is set slightly different from that of Oscillator 1.

Mod2 Source	(table)	<i>LFO-</i>
Mod2 Amount	(-64..0..+63)	<i>-16</i>
Mod2 Quantize	(Off..7)	<i>Off</i>

Here the typical brassy pitch-warp is used. It originates from the LFO envelope, which is set at LFO 1. Negative amount is used to achieve a downward dip at the beginning of each note.

If you have entered the entire Sound-program, set this parameter to 0 and you'll see how important this little detail is.

LinkOscMod	(off/on)	<i>Off</i>
------------	----------	------------

## 7.2 Analog Brass

No Link is used since Oscillator 2 is modulated differently from Oscillator 1, thus requiring it's own parameter settings.

### Wave 1 Parameters

---

Wavetable	(R1..32/I33..44/R45..56)	<i>ROI</i>
Startwave	(O..6O/Tri/Squ/Saw)	<i>saw</i>
Startsample	(free..1..127)	<i>free</i>

This results in a basic sawtooth patch. To achieve a bit of analog inaccuracy, this Startsample parameter is set to free.

Envelope Amount	(-64..O..+63)	<i>+OO</i>
Envelope Velocity	(-64..O..+63)	<i>+OO</i>
Keytrack Amount	(-64..O..+63)	<i>+OO</i>

Mod1 Source	(table)	<i>LFO 1</i>
Mod1 Control	(table)	<i>Mod Wheel</i>
Mod1 Amount	(-64..O..+63)	<i>+OO</i>

Mod2 Source	(table)	<i>LFO 1</i>
Mod2 Amount	(-64..O..+63)	<i>+OO</i>

Mode	(stepped/smooth)	<i>smooth</i>
------	------------------	---------------

No Modulation whatsoever is used, since otherwise the waveform would drastically change, yielding anything but analog-brass.

### Wave 2 Parameters

---

Startwave	(O..6O/tri/squ/saw)	<i>saw</i>
Startsample	(free..1..127)	<i>O54</i>

Again, a sawtooth wave makes for a good analog sound. A Startsample-value of 54 is used to make the overall sound a little brighter and punchier, since it introduces certain peaks in combination with Wave 1. However, they are not always the same, since Wave 1's Startsample is set to free and thus will change with each attack. This yields a nice analog-effect.

Envelope Amount	(-64..O..+63)	<i>+OO</i>
Envelope Velocity	(-64..O..+63)	<i>+OO</i>
Keytrack Amount	(-64..O..+63)	<i>+OO</i>



## 7.2 Analog Brass

Mod1 Source	(table)	<i>LFO 1</i>
Mod1 Control <i>Wheel</i>	(table)	<i>Mod</i>
Mod1 Amount	(-64..0..+63)	+00
Mod2 Source	(table)	<i>LFO 1</i>
Mod2 Amount	(-64..0..+63)	+00
Mode	(stepped/smooth)	<i>smooth</i>
LinkWaveMod	(off/on)	<i>off</i>

Again, no modulation allowed.

### Volume Parameters

---

Wave1 Vol	(0..7)	6
Wave2 Vol	(0..7)	4
Noise Vol	(0..7)	0
Sound Volume	(1..127)	127

Wave 1 is set to be approximately twice as loud as Wave 2. This is the best setting for the pitch-warp and detune settings, so they won't make a string-sound out of your brass blowers.

The total input volume is 10. Mistake? No! This yields an ever so slightly saturation effect typical for analog-synthesis without distortion. This works because of the spectrum of the saw-waveform; it doesn't necessarily apply to other waveforms.

Envelope Amount	(-64..0..+63)	+20
Envelope Velocity	(-64..0..+63)	+45

This gives a nice, not-so-dominant velocity sensitivity.

Keytrack Amount	(-64..0..+63)	+00
Mod1 Source	(table)	<i>LFO 1</i>
Mod1 Control <i>Wheel</i>	(table)	<i>Mod</i>
Mod1 Amount	(-64..0..+63)	+00
Mod2 Source	(table)	<i>LFO 1</i>
Mod2 Amount	(-64..0..+63)	+00

Once more, no further modulation of the Volume.

## 7.2 Analog Brass

### Filter Parameters

---

Cutoff	(0..127)	44
Resonance	(0..127)	16

The crucial parameters for brass sounds. The Cutoff interacts with the Filter envelope. By the way, so does the Resonance; see below for details.

The Resonance will yield this typical narrow sound important for brass to cut through.

Envelope Amount	(-64..0..+63)	+10
Envelope Velocity	(-64..0..+63)	+62

This uses the full bandwidth of the Filter envelope at the highest velocity. Due to the Amount parameter there will always be a typical brass-envelope for the Filter, yet the sound gets brighter with higher velocity, just like a brass instrument.

Keytrack Amount	(-64..0..+63)	+50
-----------------	---------------	-----

Allows more brightness for the treble-range than the bass-range.

Mod1 Source	(table)	<i>LFO 1 Envelope</i>
Mod1 Control	(table)	<i>Mod Wheel</i>
Mod1 Amount	(-64..0..+63)	+00
Mod2 Source	(table)	<i>LFO 1</i>
Mod2 Amount	(-64..0..+63)	+00

No further Cutoff modulation.

RMod Source	(table)	
<i>Filter-envelope</i>		
RMod Amount	(-64..0..+63)	-20

This is a neat effect you'll have trouble realizing on any other synthesizer. The Filter-envelope will reduce the Resonance when at the same time opening the Filter. This allows a well-sized basic Resonance without the typical "weow" when the Filter opens and closes according to the Attack- and Decay-times. Set Resonance Modulation to 0 to hear the difference.

## 7.2 Analog Brass

### Volume Envelope Parameters

---

Attack	(0..127)	020
Decay	(0..127)	025
Sustain	(0..127)	090
Release	(0..127)	037

A typical brassy envelope.

Attack Mod Source	(table)	<i>Velocity</i>
Attack Mod Amount	(-64..0..+63)	-22

Shortens the Attack when you play louder.

Decay Mod Source	(table)	
<i>Keytracking</i>		
Decay Mod Amount	(-64..0..+63)	+17

Prolongs the Decay-time at higher pitches. Together with the Keytracking-parameter of the Filter this basically produces a more tuba-like sound in the bass and a more trumpet-like sound at the treble-register. *Velocity*

Sustain Mod Source	(table)	<i>Aftertouch</i>
Sustain Mod Amount	(-64..0..+63)	+63

Opens the Filter with aftertouch. Note that aftertouch is not used directly at the Filter but rather via the envelope, since this yields better results when releasing the keys.

Release Mod Source	(table)	<i>Rel</i>
Release Mod Amount	(-64..0..+63)	+00

### Filter Envelope Parameters

---

Delay	(0..127)	000
Attack	(0..127)	023
Decay	(0..127)	020
Sustain	(0..127)	080
Release	(0..127)	009

A nice brassy Filter envelope.

Delay Mod Source	(table)	<i>LFO 1</i>
Delay Mod Amount	(-64..0..+63)	+00
Attack Mod Source	(table)	<i>Velocity</i>
Attack Mod Amount	(-64..0..+63)	-33

## 7.2 Analog Brass

Shortens the Attack when you play louder.

Decay Mod Source	(table)	
<i>Keytracking</i>		
Decay Mod Amount	(-64..0..+63)	+22

Prolongs the Decay when you play higher, giving that cute tuba-sound in the bass.

Sustain Mod Source	(table)	<i>Aftertouch</i>
Sustain Mod Amount	(-64..0..+63)	+63

Wow! It not only gets louder, but even brighter when you apply pressure to the keys!

Release Mod Source	(table)	<i>Rel Velocity</i>
Release Mod Amount	(-64..0..+63)	+00

### Wave Envelope Parameters

---

Time1	(0..127)	031
Level1	(0..127)	127
Time2		032
Level2		064
Time3		016
Level3		000
Time4		000
Level4		000
Time5		000
Level5		000
Time15		000
Level6		000
Time7		000
Level7		000
Time8		000
Level8		000

Time Mod Source	(table)	<i>Keytracking</i>
Time Mod Amount	(-64..0..+63)	+00
Level Mod Source	(table)	<i>Aftertouch</i>
Level Mod Amount	(-64..0..+63)	+00

Key Off Point	(1..8)	2
Loop Start Point	(1..8)	1
Loop Mode	(off/on)	off

Sorry. No Wave-envelope for this sound.

### LFO 1 Parameters

---

Rate (1..127)	050
---------------	-----



## 7.2 Analog Brass

Shape	(sin/saw/ pulse/random)	<i>sin</i>
Symmetry	(-64..0..+63)	<i>+00</i>
Humanize	(off..7)	<i>off</i>
Rate Mod Source	(table)	<i>LFO 1</i>
Rate Mod Amount	(-64..0..+63)	<i>+00</i>
Level Mod Source	(table)	<i>max.</i>
Sync	(off/on)	<i>off</i>
Delay	(off/retrigger/ 1..126)	<i>off</i>

The actual LFO isn't used, so leave everything as it is set by `sound-init`.

Attack	(0..127)	<i>009</i>
Decay	(Off/1..127)	<i>018</i>

These are the times for the pitch-warp of Oscillator 2. If you like more "detuning" at the attack of a note, set them slightly longer. If you hate this much pitch-warp - you know what to do, don't ya...right, turn these values down.

### LFO 2 Parameters

---

Rate	(1..127)	<i>060</i>
------	----------	------------

A nice smear-rate for brass.

Shape	(sin/saw/ pulse/random)	<i>sin</i>
Symmetry	(-64..0..+63)	<i>+15</i>
Humanize	(off..7)	<i>004</i>

Sets a little asymmetrical smear, whose rate is not constant. Sounds like humans who would blow a funky MicroHorn.

### Panning/Glide Parameters

---

Panning	(L63..M..R64)	<i>L10</i>
Pan Mod Source	(table)	<i>Keytrack</i>
Pan Mod Amount	(-64..0..+63)	<i>+63</i>

## 72 Analog Brass

An arbitrary Panning-effect that changes with pitch. Use LFO 2 to give the effect of a wild bunch of session players on drugs. Or use the LFO-envelope to give the impression of them moving at the same speed...

Glide	(off/Gliss/Porta MidiPorta/MIDI Gliss)	<i>off</i>
Glide Rate	(0..127)	<i>032</i>
Glide Mode	(Time/Distance)	<i>Distance</i>

No glide. It's better for colder regions.

Temperment	(In+/In-/rn1/rn2/TT1..4)	<i>In+</i>
------------	--------------------------	------------

Use any temperment you like. A nice slightly stretched tuning might sound good.

Sound Name

---

Sound Name	(16 Digits)	<i>Analog Brass</i>
------------	-------------	---------------------



## 7.3 Rolling Glass

### 7.3 Rolling Glass

Now here comes a sound which shows the unique characteristics of a MicroWave wavescan-sound. Use it to really let those pads shine in your next ballad.

Oscillator 1 Parameters

set to value







*Envelope*



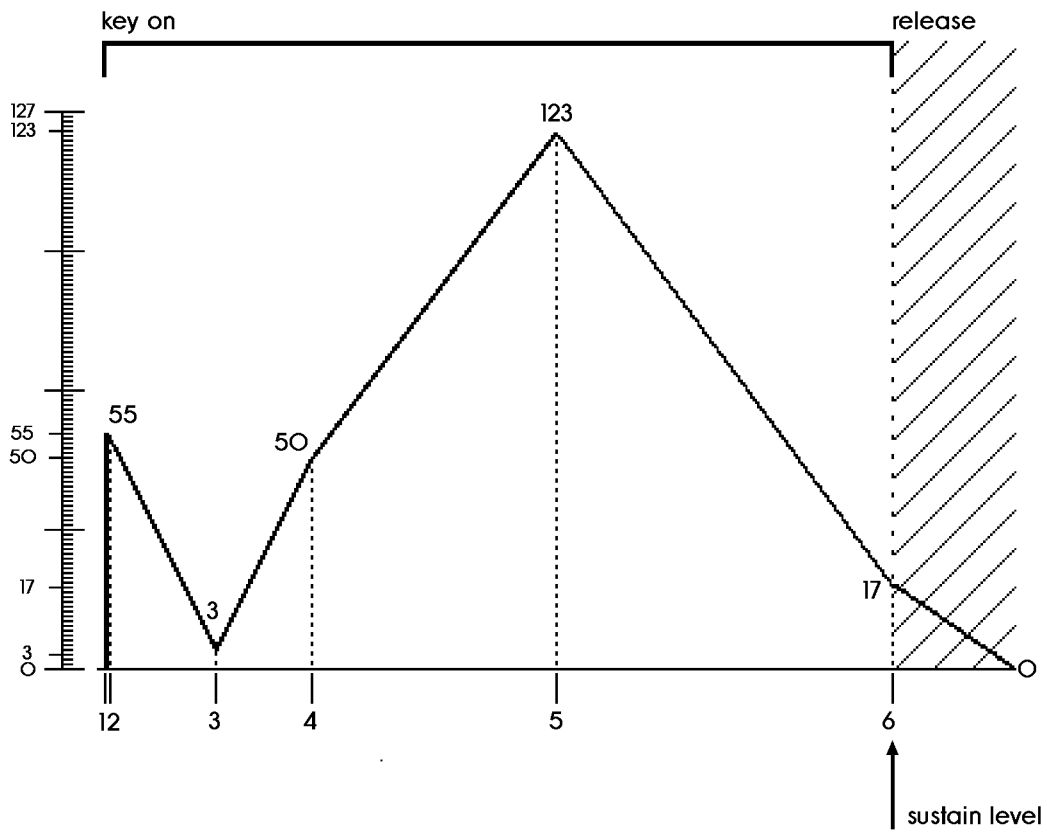


*Velocity*



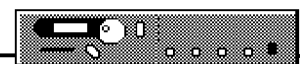


# 73 Rolling Glass





## 73 Rolling Glass

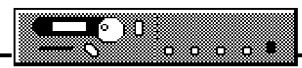






# MicroWave Parameter List

PRG Select	MIDI Mode	Instr Param	Instr Select
Sound Prog. No. or Multi Prog. No.	Sound Prog. Multi Prog.	<p><b>Instr. Param.:</b>                      Master Vol.                      Control. W                      Control. X                      Control. Y                      Control. Z                      Prog Ch. Mode                      No. of Inst.                      Name</p> <p><b>Multi Param.:</b>                      Enable Inst.                      Midi Channel                      Sound                      Key Limit L                      Key Limit H                      Vel Limit L                      Vel Limit H                      Vel Curve                      Transpose                      Detune                      Temperment                      Volume                      Panning                      Panning Mod.                      Routing                      Prog Change                      Pitch Wheel                      Modulation                      Aftertouch                      Poly Press.                      Volume Ctrl.                      Panning Ctrl.                      Sustain Ped.</p>	Multi Instr. 1-8



# MicroWave Parameter List

Volume/Tune	MIDI Param	Tables/Maps	Device Param
<p>Master Vol Stereo With Mastertune</p>	<p>Midi Chan. Control. W Control. X Control. Y Control. Z Prog Change Pitch Wheel Modulation Aftertouch Poly Press. Volume Ctrl. Panning Ctrl. Sustain Ped. PC maps PC Shutdown</p>	<p><b>User Tuning T1:</b> Key Pitch Tune</p> <p><b>User Tuning T2:</b> Key Pitch Tune</p> <p><b>User Veloc. T1:</b> I Vel. O Vel.</p> <p><b>User Veloc. T2:</b> I Vel. O Vel.</p> <p><b>Sound PC Map:</b> PC Sound</p> <p><b>Multi PC Map:</b> PC Multi</p>	<p>Midi Overfl. Midi Out=Thru Dev. Number</p>



# MicroWave Parameter List

Oscillators Waves	Volume Filter	Envelopes LFO's	Pan/Glide Name
<p><b>Oscillator 1:</b>            Octave            Semitone            Detune            Bend Range            Pitchmode            Mod 1 Src.            Mod 1 Ctrl.            Mod 1 Amnt.            Mod 2 Src.            Mod 2 Amnt.            Mod 2 Qnt.</p> <p><b>Oscillator 2:</b>            Octave            Semitone            Detune            Bend Range            Pitchmode            Mod 1 Src.            Mod 1 Ctrl.            Mod 1 Amnt.            Mod 2 Src.            Mod 2 Amnt.            Mod 2 Qnt.            Link Osc. Mod.</p> <p><b>Wave 1:</b>            Wavetable            Startwave            Startsample            Envel. Amnt.            Envel. Vlcty.            Keytrk. Amnt.            Mod 1 Src.            Mod 1 Ctrl.            Mod 1 Amnt.            Mod 2 Src.            Mod 2 Amnt.            WaveMode</p> <p><b>Wave 2:</b>            Startwave            Startsample            Envel. Amnt.            Envel. Vlcty.            Keytrk. Amnt.            Mod 1 Src.            Mod 1 Ctrl.            Mod 1 Amnt.            Mod 2 Src.            Mod 2 Amnt.            WaveMode            Link Wave Md</p>	<p><b>Volume:</b>            Wave 1 Vol.            Wave 2 Vol.            Noise Vol.            Sound Vol.            Envel.Amnt.            Envel. Vlcty.            Keytrk. Amnt.            Mod 1 Src.            Mod 1 Ctrl.            Mod 1 Amnt.            Mod 2 Src.            Mod 2 Amnt.</p> <p><b>Filter:</b>            Cutoff            Resonance            Envel.Amnt.            Envel.Vlcty.            Keytrk Amnt.            Mod 1 Src.            Mod 1 Ctrl.            Mod 1 Amnt.            Mod 2 Scr.            Mod 2 Amnt.            R Mod Src.            R Mod Amnt.            Velocity FA            Aftertouch FA            Mod1 Wheel FA</p>	<p><b>Volume Envelope:</b>            Attack            Decay            Sustain            Release            AMod Src.            AMod Amnt            DMod Src.            DMod Amnt            SMod Src.            SMod Amnt            RMod Src.            RMod Amnt</p> <p><b>Filter Envelope:</b>            Delay            Attack            Decay            Sustain            Release            DMod Src.            DMod Amnt            AMod Src.            AMod Amnt            DcMod Src.            DcMod Amnt            SMod Src.            SMod Amnt            RMod Src.            RMod Amnt</p> <p><b>Wave Env.</b>            Time 1 /Level 1            Time 2 /Level 2            Time 3 /Level 3            Time 4 /Level 4            Time 5 /Level 5            Time 6 /Level 6            Time 7 /Level 7            Time 8 /Level 8</p> <p><b>LFO 1</b>            Rate            Shape            Symmetry            Humanize            R Mod Src.            R Mod Amnt            L Mod Src.            L Mod Amnt            Sync            Delay            Attack            Decay</p> <p><b>LFO 2</b>            Rate            Shape            Symmetry            Humanize</p>	<p>Panning            PMod Src.            PMod Amnt.            Glide            Glide Rate            Glide Mode            Temperment</p> <p>Sound Name</p>



# MicroWave Parameter List

Fast Access 1	Fast Access 2	Env-Macro	Mod-Macro
<p><b>Envelope FA:</b> Attack Decay Sustain Release</p> <p><b>Env. Amount FA:</b> Volume Filter Wave</p>	<p><b>Velocity FA:</b> Envelope Velocity Mod Ctrl-Inputs Mod Source-Inputs</p> <p><b>Aftertouch FA:</b> Mod Ctrl-Inputs Mod Source-Inputs</p> <p><b>Mod Wheel FA:</b> Mod Ctrl-Inputs Mod Source-Inputs</p>	<p><b>Volume Env. Mac.:</b> Click Organ Pipe Organ Strings Woodwind Orch. Brass Pop Brass Piano Pluck Long Perc. Medium Perc. Short Perc.</p> <p><b>Filter Env. Mac.:</b> Click Organ Pipe Organ Strings Woodwind Orch. Brass Pop Brass Piano Pluck Long Perc. Medium Perc. Short Perc. Delay Perc.</p> <p><b>Wave Env. Mac.:</b> Slow Attack Slow Decay ADSR Envelope Inverse ADSR Env. Spit Valve Slap Back Wah Wah Single Echo Repeat Echo Long Loop</p>	<p><b>Mod-Macro 1:</b> Mod easy Vib Aft easy Vib Mod acoust. Vib Aft acoust. Vib Stereo Chorus Vel Chorus Vel Timbre Vib</p> <p><b>Mod-Macro 2:</b> Mod easy Vib Aft easy Vib Delay easy Vib Mod acoust. Vib Aft acoust. Vib Del acoust. Vib Tremolo Pseudo Leslie Auto Wah-Wah Auto Panning Vel Auto Pan Echo Stereo Echo</p>



MIDI Implementation Chart

MODEL: MicroWave

Date: Feb. 20, 1990  
Version: 1.10

Function		Transmitted	Received	Remarks
Basic Channel	Default Changed	-	1 1..16	
Mode	Default Messages Altered	- -	1 Omni/Poly/ Multi >All notes off	Modes can't be switched via MIDI
Note Number	True Voice	- -	0..127 0..108	
Velocity	Note ON Note OFF	x x	o o	
After Touch	Key's Ch's	x x	o o	
Pitch Bender		x	o	
Control Change		x	o	<i>fixed:</i> Ctrl. 1 (Mod) 3 (Breath) 7 (Vol) 10 (Pan) 64 (Sustain) 65 (Porta) <i>free:</i> 4 assignable Controller, 0..120
Program Change	True #	x	o, 0..127 0..127	0...63:internal 64..127:Card (x2, single/Multi)
System Exclusive		o	o	
System Common	Song Pos Song Sel Tune	x x x	x x x	
System Real Time	Clock Commands	x x	x x	
Aux Messages	Local on/off All Notes Off Active Sense Reset	x x x x	x o o o	

Notes | Program Changes may be redirected via 2 maps

Mode 1: OMNI ON, POLY    Mode 2: OMNI ON, MONO    o: yes  
Mode 3: OMNI OFF, POLY    Mode 4: OMNI OFF, MONO    x: no

