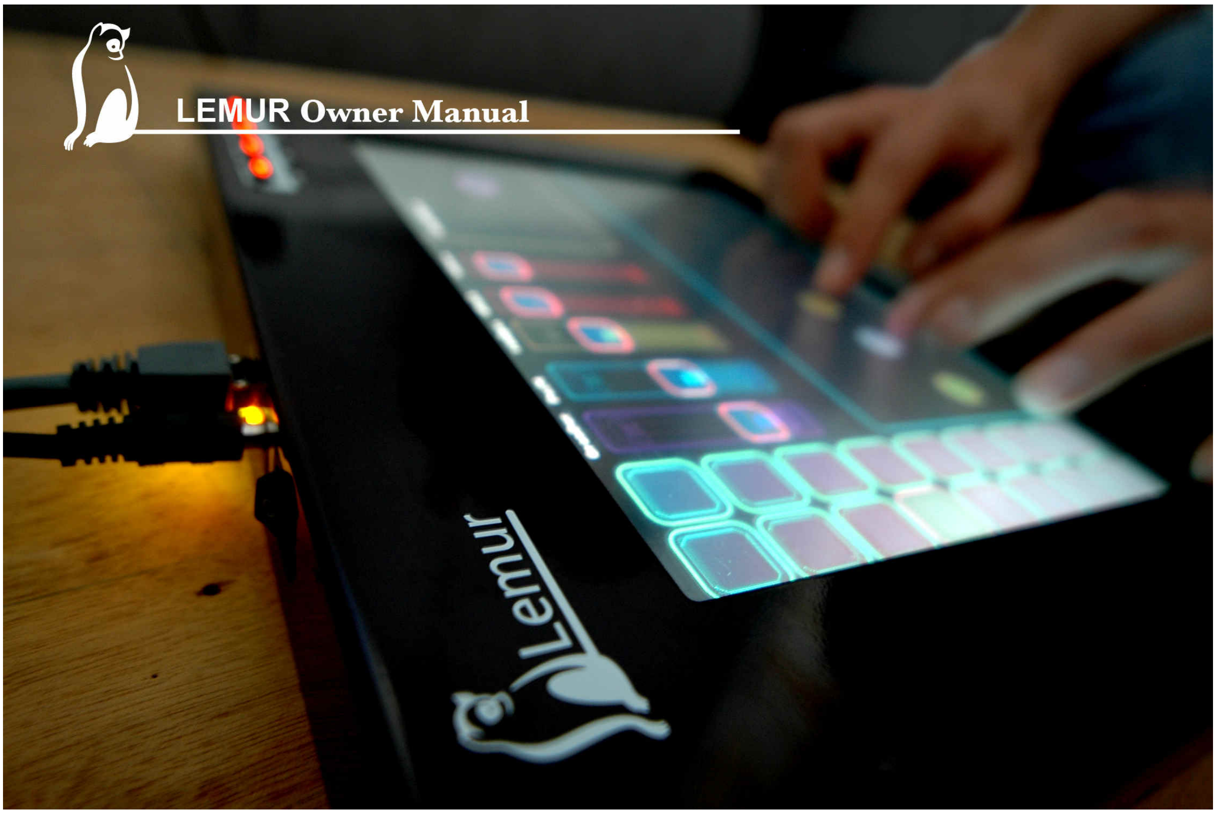
LEMUR
OWNER MANUAL

WWW.JAZZMUTANT.COM



Version 1.2

Thank you and congratulations on your choice of the JazzMutant LEMUR.

Before using this unit, carefully read the section entitled: "USING THE UNIT SAFELY" and "IMPORTANT NOTES". These sections provide important information concerning the proper operation of the unit. Furthermore, to fully grasp the features and possibilities of your new unit, we recommend that you read the Owner's Manual in its entirety and to keep it at hand as a reference.

TABLE OF CONTENTS

1 Using the unit safely

In order to prevent fire, electric shock, or injury to the user, always observe the following guidelines:
- Before using this unit, make sure to read the instructions below and the Owner's Manual.
- Do not open or modify in any way the unit or its AC adaptor.
- Do not attempt to repair the unit, or replace parts within it. Refer all servicing to your retailer or authorized JazzMutant distributor, as listed on the "Information" page.
- Never use or store the unit in places that are:
  · Subject to extreme temperature (e.g., direct sunlight, an enclosed vehicle, near a heating duct, on top of heat-generating equipment);
  · Damp (e.g., baths washroom, on wet floors);
  · Humid;
  · Exposed to rain;
  · Dusty;
  · Subject to high levels of vibration.
- Be sure to use only the AC adaptor supplied with the unit. Also, make sure the line voltage at the installation matches the input voltage specified on the AC adaptor's body. Other AC adaptors may use a different polarity or be designed for a different voltage, so their use could result in damage, malfunction, or electric shock.
- Do not excessively twist or bend the power cord, nor place heavy objects on it. Doing so can damage the cord, producing severed elements and short circuits. Damaged cords are fire and shock hazards!
- Do not allow any objects (e.g., flammable material, coins, pins); or liquid of any kind (water, soft drinks, etc.) to penetrate the unit.
- In the case of any of the following events, immediately turn the power off, remove the AC adaptor from the outlet, and request service by your retailer or authorized JazzMutant distributor:
  · The AC adaptor or the power-supply cord as been damaged;
  · Smoke or unusual odours are observed;
  · Objects have fallen into, or liquid has spilled onto the unit;
  · The unit has been exposed to rain (or otherwise has become wet);
  · The unit does not appear to operate normally or exhibits a marked change in performance.
- In households with small children, an adult should provide supervision until the child is capable of following all the rules essential for the safe operation of the unit.
- Protect the unit from strong impact (Do not drop it!)
- Do not force the unit's power-supply cord to share an outlet with an unreasonable number of other devices. Be especially careful when using extension cords--the total power used by all devices you have connected to the extension chord must never exceed the chord’s power rating (watts/amperes). Excessive loads can cause the insulation on the cord to heat up or even melt down.
- The unit and the AC adaptor should be located so their location or position does not interfere with the proper ventilation.
- Always grasp only the output plug or the body of the AC adaptor when plugging or unplugging the unit.
- Any accumulation of dust between the AC adaptor and the power outlet can result in poor insulation and possibly cause a fire. Periodically wipe away such dust with a dry cloth. Also, disconnect the power plug from the outlet whenever the unit is to remain unused for an extended period of time.
- Try to prevent cords and cables from becoming entangled. Also, all cords and cables should be placed so they are out of the reach of children.
- Never climb on top of, nor place heavy objects on the unit.

- Never handle the AC adaptor body, or its output plugs, with wet hands when plugging into, or unplugging from, an outlet or this unit.
- If you need to move the instrument, make sure to have a firm grip, to protect yourself from injury and the unit from damage. Also, don't forget to disconnect all cords and cables!
- Before cleaning the unit, turn off the power and unplug the AC adaptor from the outlet.
- Whenever you suspect the possibility of lightning in your area, disconnect the AC adaptor from the outlet.
- Should you remove security screws, make sure to put them in a safe place out of children's reach, so there is no chance of them being swallowed accidentally.

2 Important Notes

In addition to the items listed under "USING THE UNIT SAFELY", please read and observe the following:

Power Supply
- Do not use this unit on the same power circuit with any device that generates line noise (such as an electric motor or variable lightning system).
- The AC adaptor will generate heat after long hours of continuous use. This is normal and should not cause any alarm.
- Before connecting the unit to other devices, turn off the power to all units. This will help prevent malfunctions and/or damage to speakers or other devices.

Placement
- Using the unit near power amplifiers (or other equipment containing large power transformers) may induce “hum”. To alleviate this problem, change the orientation of this unit, or move it farther away from the source of interference.
- This device may interfere with radio and television reception. Do not use this device in the vicinity of such receivers.
- Noise may be produced if wireless communications devices, such as cell phones, are operated in the vicinity of this unit. Such noise could occur when receiving or initiating a call, or while conversing. Should you experience such problems, try relocating the wireless devices so that they are further from this LEMUR or switch them off.
- Do not expose the unit to direct sunlight, place it near devices that radiate heat, leave it inside an enclosed vehicle, or otherwise subject it to temperature extremes.
- When moved from one location to another where the temperature and/or humidity is very different, water droplets (condensation) may form inside the unit. Damage or malfunction may result if you attempt to use the unit in this condition. Therefore, before using the unit, you must leave it idle for several hours until the condensation has completely evaporated.

Maintenance
- For everyday cleaning wipe the unit with a soft, dry cloth or one that has been slightly dampened with water. To remove stubborn dirt, use a cloth with a mild, non-abrasive detergent. Afterwards, be sure to wipe the unit thoroughly with a soft, dry cloth.
- Never use benzene, thinners, alcohol or solvents of any kind; such chemical may cause discoloration and/or deformation.

Additional Precautions
- When working with the unit’s buttons, display, or other controls or when using its cords and cables try to be reasonably gentle. Rough handling can lead to malfunctions.

- Never strike or apply strong pressure to the display.
- When connecting/disconnecting all cables grasp the connector itself-never pull on the cable. This way you will avoid causing shorts, or damage to the cable's internal elements.
- When transporting the unit, please ensure it is properly packed to protect it from shock and damage, using, for example, a sturdy laptop case (15' or 17' should fit). Place an additional protective layer on the display.
- Use the LAN cable supplied with the unit. If using some other cable, please make sure it is a ‘normal’ LAN cable, not a 'cross-over' cable.

3 System Overview

3.1 Hardware Overview

Apart of the touch panel itself, Lemur features a set of four buttons

- The leftmost button open the Local Settings window
- The Interface List button displays a thumbnails gallery giving an overview of all the interfaces. Just tip one thumbnail with your finger to jump to the corresponding interface.
- The rightmost arrows buttons flip from one interface to the next / previous one.

Not surprisingly, the rear panel of the lemur features the power supply plug and switch as well as the CAT5/RJ45 Ethernet interface.

3.2 JazzEditor Overview

The LEMUR is a stand-alone peripheral that doesn't need any drivers. However, it does require proper configuration of your network interface. The LEMUR comes with an essential graphical interface builder: JazzEditor. To install, just drag and drop JazzEditor into the folder of your own choice on your computer's hard disk.

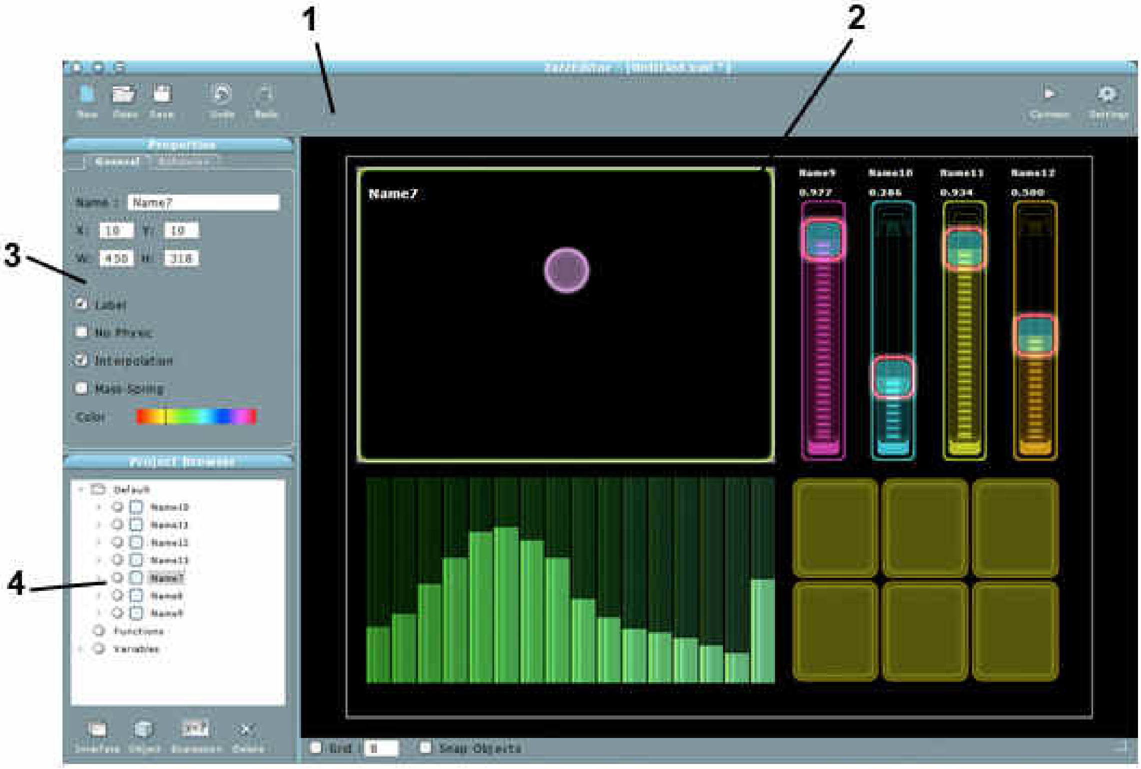
The interface editor software has been developed to simplify the creation and display of your control interfaces. With the exception of certain keyboard shortcuts, this software is completely multi-platform (Mac OSX, Windows). The conception of the JazzEditor software is quite simple to use. It consists of four elements:

The Toolbar **(1)** contains the standard menu items for File and Edit menus, such as New, Open, Save, Undo and Redo. The Toolbar also contains various other commands for changing your interfaces or the configuration of LEMUR.

The Browser window **(2)** gives you a hierarchical view of your current project (interfaces, objects, expressions) and access to editing functions (create or delete interfaces, objects, expressions).

The Edition **(3)** window represents the display of LEMUR, here you will construct your interface by dragging and dropping objects.

The Option window **(4)** allows you to configure the properties of selected interface objects.



Name : Name7

X: 10 Y: 10

W: 450 H: 318

☑ Label

☐ No Physic

☑ Interpolation

☐ Mass-Spring

Color

4 LEMUR configuration

The configuration of LEMUR is done from the settings window of the interface editor.

4.1 LEMUR IP address configuration

There are two configuration modes for the LEMUR IP address (Internet Protocol): static mode and dynamic mode (DHCP). Depending on your network setup, Lemur will automatically choose the proper one. Thus, when starting, the LEMUR automatically looks for a DHCP server in order to determine if it is connected to a local network or not. If it finds one, the DHCP server will assign dynamically an available address. In the case that the LEMUR doesn't find a DHCP server, it will automatically switch to static mode and assign itself a default IP address (i.e. 169.254.1.49).

4.1.1 Configuring your computer for Single connexion (P2P)

Unless you've configured your computer as a DHCP server, when connecting the LEMUR directly to it with the supplied crossover LAN cable, it will switch itself in static IP mode. In most case, you won't need to modify your computer network configuration. Just make sure your laptop IP address is different of Lemur's one.

4.1.2 Local network setup

If connecting several Lemurs to a local network administrated by a server, the DHCP server will assign a unique IP address to each LEMUR.

4.1.3 Configuring Targets and ports

As several LEMUR and/or computers can be connected on a local network, it is essential to define not only which LEMUR the newly created interface must be sent to, but also which computer has to be controlled by each LEMUR.

All Lemurs on the network will appear on the "Client" list of the Connexion window. From here you can select which one you want to edit.

Then, to select the target computer (i.e. the one that will receive control messages), open the Settings window.
The Settings window is comprised of three sub-windows. The upper part displays some useful information: the current JazzEditor version and above your computer IP address.
The second sub-window is a status monitor that tells you whether or not your laptop is connected to a LEMUR.
The lower part of the Setting Window features both the target IP and port number configuration. By default the target IP field is supposed to be fill by your computer IP address (the one that is displayed in the upper sub-window). If you'd like the LEMUR to control another computer, just enter its IP address manually in the target IP field.

Computers connected to the same network can exchange various types of data simultaneously (e.g., Internet browsing, instant messengers, file sharing, etc.). Consequently, each application with access to the network must be assigned one or several specific communications ports. In order to send messages to your target software, you must define the communication port through which your software will look for data sent to it. Refer to the documentation of your software in order to know which port it can “read” (some softwares can read only a certain port range, for example from 9000 to 9999).

Note: the 8001 port is reserved, it allows to different Lemurs on a same network to present themselves to the interface editor.

5 Projects, interfaces and objects

5.1 The XML project file

In short, a LEMUR project is an XML file (i.e., a text file which can be written and edited with any text editor) which describes an interface set. A project can also contain variables and functions, but we will talk about that later (see chapter 6: using mathematic expressions). Finally, it is inside the project where are save the configuration data. There are as many interfaces as there are display configurations (i.e. the way objects are organized together). The user will have these display configurations at his disposal once the contents of the XML file are uploaded onto the LEMUR. Then, it will be possible to switch from one display configuration to another by using the browsing buttons (+/-) or by selecting the right icon from the interface thumbnails.

In the editor software, the project Explorer window allows one to visualize and browse inside the XML project directory structure. The interfaces are symbolized by folders. Clicking on each folder shows the arrangement of interface items for that display configuration. The objects are represented by boxes whose content can also be visualized by clicking. Inside each box appear the object variables. We will talk about the notion of variable later.

5.2 Objects

5.2.1 Object properties

Each interactive object distinguishes itself from the others by its original name. The name is assigned to the object upon creation, and can be modified through the object's properties window.

Besides the name, other user-definable properties are available. Different objects have different adjustable properties; for instance, some properties concern the configuration of an object (e.g., the number of balls of a multiball area), others adjust the graphical appearance of objects, and some the physical behaviours.

Once created, the object appears on the upper left-hand corner of the Edition window, while an icon is displayed inside the Explorer window. Objects can be moved, resized, copied, cut and pasted from one display configuration to another. It's also possible to modify an object's properties. When clicking on an object in the Edition window or its icon in the Explorer window, its properties are displayed in the Option window, below the Explorer window. We will see later that most object properties (i.e., its physical behaviour) can also be determined by setting up values manually or modifying them in real-time by other objects.

The aim of the LEMUR is to send control information to software applications. This control information is the data describing the state of interface objects, passed from the LEMUR to the software at a high rate. Typically, the "x" variable of a fader corresponds to its cursor position, while the "x" and "y" variables of an Area reveal the Cartesian coordinates of the ball. When checking the box of a variable in the Explorer window, the variable is sent on the network as a OSC control message.

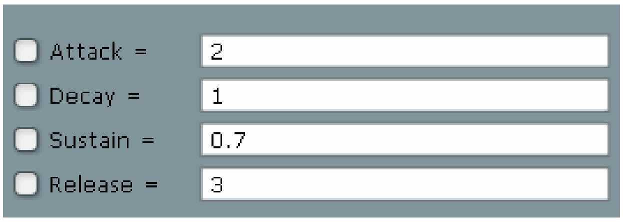
Those messages are the received by your target software for real-time control of some process.

5.2.2 *Object library*

Today’s user interface object gallery includes the following items.

Pads

Pads are basic keys which remain on so long as you touch them and turn off as soon as you release. For each pad set, you must configure the desired number of rows and columns. A pad set sends a single message comprised of an ordered list of values in range of 0 to 1 corresponding to the status of each key. The list is read from the top left cell, rows then columns. By default, this status is 1 when pressed and 0 when released. However, applying an ADSR envelope to a set of pads will interpolate its output value. In the example above, the ADSR is set so that the value goes from 0 to 1 in two seconds, then decreases down to 0.7 in one second and remains to this value until you release the key. Then, it will take 3 seconds to get down to 0.

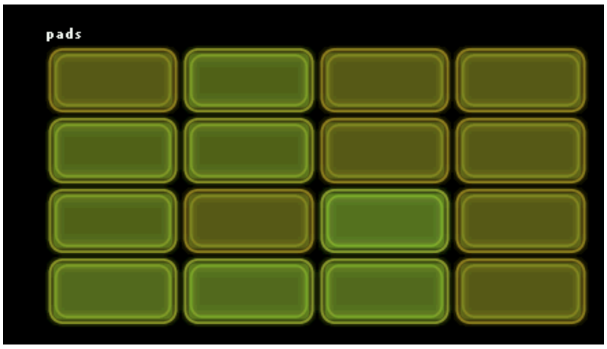


□ Attack = 2

□ Decay = 1

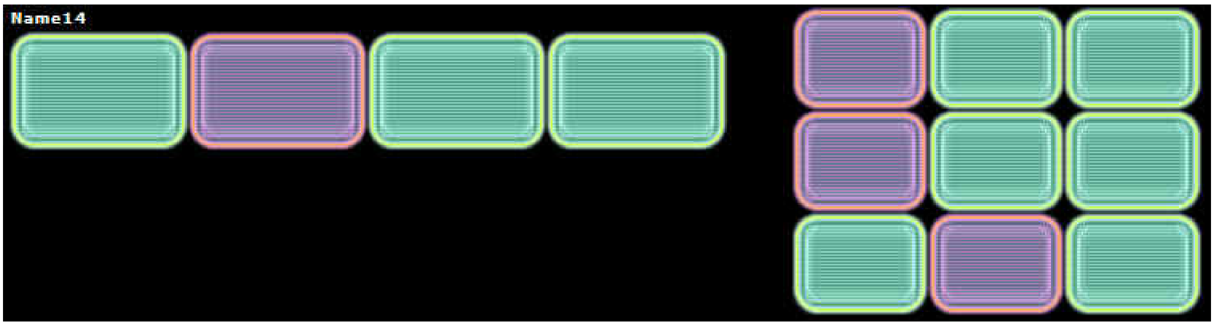
□ Sustain = 0.7

□ Release = 3



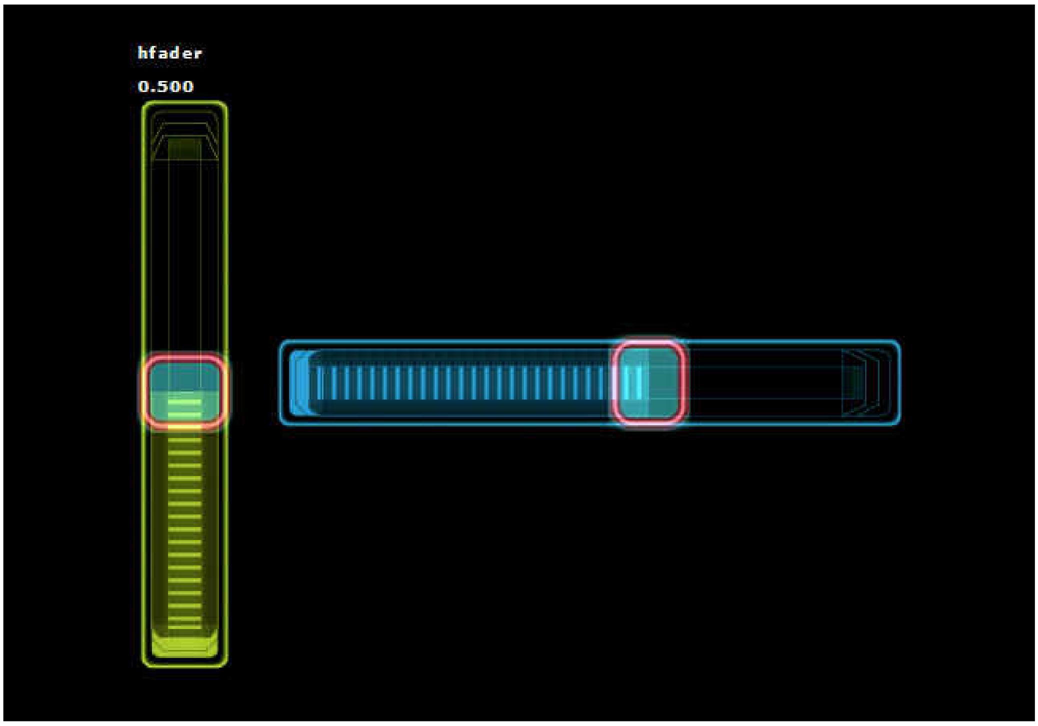
Switches

Switches are identical to the pad set except that they change their status each time you touch them. Two modes are available: the standard mode (one can turn on as much toggles as we want), and the “radio” mode (one can only turn on one toggle at a time).



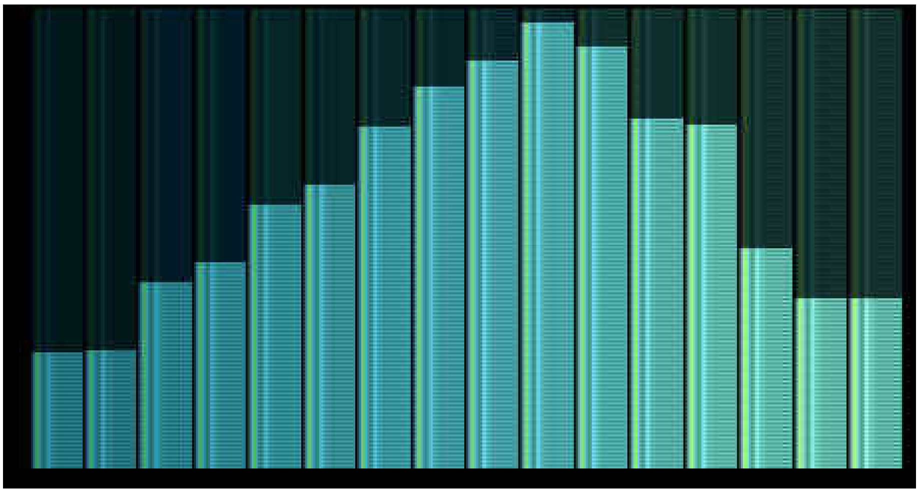
Faders

Simple linear faders with configurable physical behaviour. Their Options window allows one to choose vertical or horizontal faders. Appearance options include colour, label and arbitrary value display.



Multislider

A variable set of sliders (up to 256). When the "gravity" mode is checked, the multislider will obey the physical model of a string, whose tension and height parameters are adjustable.

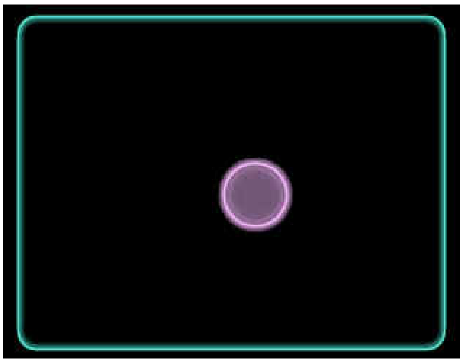


Monitor

This object displays the value of any other object or variable.

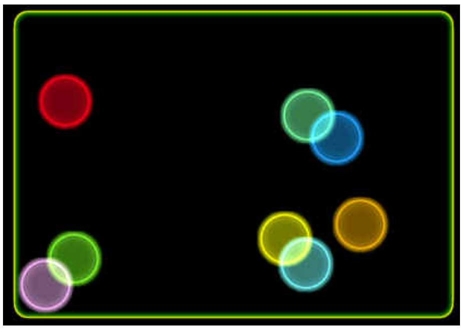
Area

A rectangular area in which a ball can be moved. The balls are attracted by the finger tip and react according to a mass-spring type of physical model. The physical parameters are the "friction" of the environment and the influence ("smooth") of the fingers on the ball. The default setting (almost zero friction and strong influence) allows the ball to follow the fingertips as closely as possible.



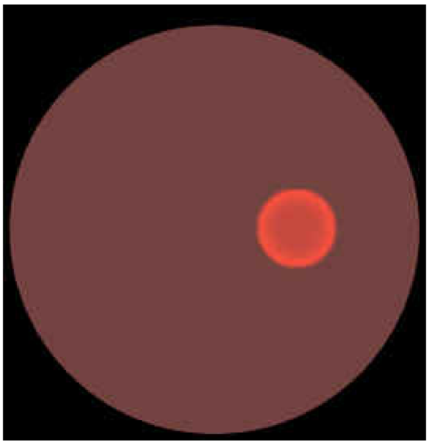
Multiball

A variation of the Area, but with a configurable number of balls (up to 10) and two different playing modes: continuous (by default) and ephemeral. In the continuous mode, the balls are created when initializing the object and remain on display. In the ephemeral mode, each finger will create a new ball when touching the display. The balls will progressively appear and disappear when loosing them in accordance with the ADSR envelope (see Pads).



Ring Area

Another variation of the Area is the ring area, with a circular form. The ball is always drawn to an attraction point by a spring. By default, this point is located to the center of the ring. However, this point can be moved by giving it new relative cartesian coordinates in a range of 0 to 1.



6 Using mathematical expressions

6.1 Definitions

The LEMUR includes a mathematical expressions parser which widens the control and interaction possibilities offered to the user. The expression parser is a simple and powerful tool to that allows, for example, easier calibration of the data that you want to send to your target software or to create complex interactions among different objects. It can also be very useful for configuring and adjusting in real time an object's behaviour, the physical parameters or at times the look of the objects.

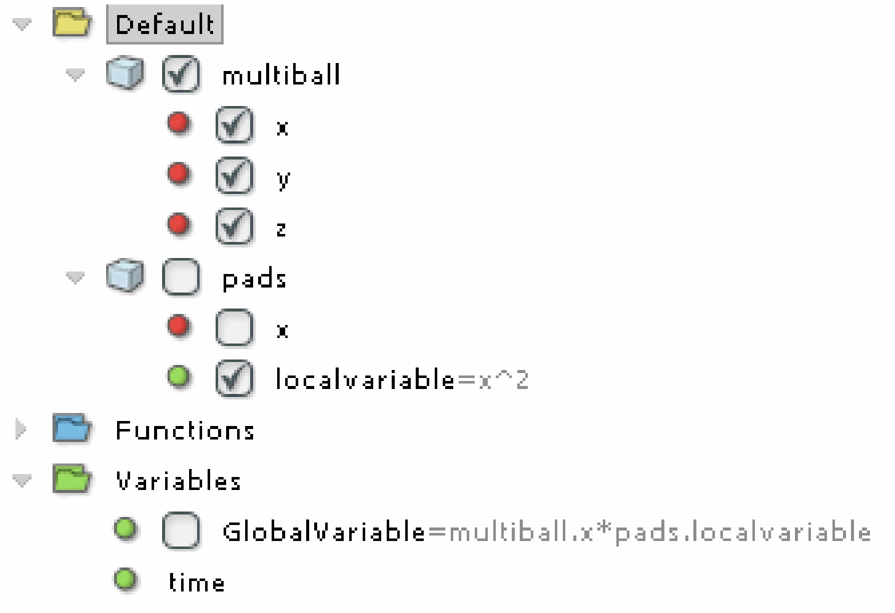
Before going into further detail regarding the functionalities of expressions, we first define two things: a vocabulary of the system and the infrastructure that allows inter-object communication with expressions.

We call Expression, a mathematical or logical operation that is defined by the user. For instance, (a+b)/2 and a>b are two basic expressions. In any software application, in order to get the result (output value) of an operation, the result must be stored in a specific space of the memory. Commonly, this space is sometimes referred to as "variable".
For example, the cursor position of a fader called "foo" will be stored in a variable called "foo.x". Likewise, when moving a ball inside the surface of an area called "bar", its Cartesian coordinates (x,y) will be stored in variables named "bar.x" and "bar.y" respectively.

In order to evaluate the expression (a+b)/2, one must first define the variable in which the result of the expression will be stored. Thus, one can create a variable called "my_variable", and assign to it a value of (a+b)/2.

6.2 Objects variables, local variables, global variables

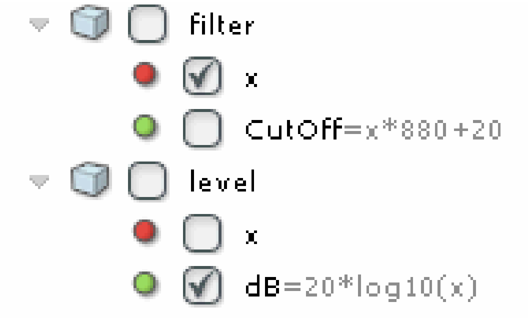
As mentioned above, each interface object has its own internal variables which are automatically created upon initialization. These variables are called "object variables" because they are internal and specific to each object: the "x" of the fader "foo" is different from the "x" of the area "bar". They cannot be deleted, unless by deleting the object itself. The object variables appear inside the object folder in the Exploration window.



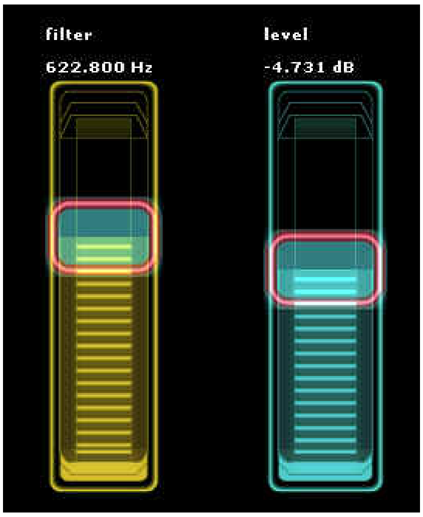
```
Default
  ☑ multiball
      ● ☑ x
      ● ☑ y
      ● ☑ z
  □ pads
      ● □ x
      ● ☑ localvariable=x^2
Functions
Variables
  ● □ GlobalVariable=multiball.x*pads.localvariable
  ● time
```

By default, the values of these variables are always a decimal number between 0 and 1. Nevertheless, in many cases this value range seems to be inadequate to control parameters such as the frequency of a filter, the amplitude of a signal, the running speed of a loop, the position in the space, etc. Thanks to the expression parser, it is possible to create variables that match exactly the range of the parameter you want to control.

For example, by creating a variable called ‘Cutoff" entering the expression "x*880+20", you will get a value range between 20 and 1000. In this case, the arithmetic scaling allows that are more appropriate for control, for instance, the cut-off frequency of a high-pass filter The same way, by entering the expression "20*log10(x)", you will get a logarithmic value segment between –inf and 0, which would be appropriate for controlling the decibel output level of an audio signal.



```
□ filter
    ● ☑ x
    ● □ CutOff=x*880+20
□ level
    ● □ x
    ● ☑ dB=20*log10(x)
```



Besides the local variables created for each object, it is also possible to create 'Global' variables, i.e. common to the entire project. Thus, we can think that a single object can control distinctive parameters having each one different value range.

To create a new expression, click on the "Expression" icon. Once created, the global variables appear inside the "Variable" folder in the Exploration window. As for local variables of an object, they appear inside its own object folder in the Exploration window.

6.3 Operators

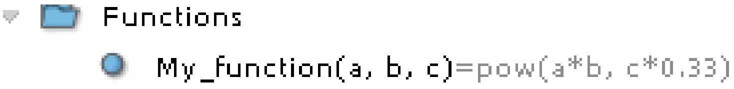
The available operators for creating and editing expressions are:

<table>
<tr><td>• +
• -
• *
• /
• pow</td><td>• a+b
• b-a
• a*b
• a/b
• pow(a,b)</td><td>Basic arithmetic operators</td></tr>
<tr><td>• sin
• cos
• tan
• log
• log10
• exp
• acos
• asin
• atan</td><td>• sin(a)
• cos(a)
• tan(a)
• log(a)
• log10(a)
• exp(a)
• acos(a)
• asin(a)
• atan(a)</td><td>Trigonometric operators</td></tr>
<tr><td>• >
• >=
• ==
• <
• <=
• !=</td><td>• a>b
• a>=b
• a==b
• a<b
• a<=b
• a!=b</td><td>Logical operators</td></tr>
<tr><td>• &&
• ||
• |
• ?</td><td>• a&&b
• a||b
• a|b
• a?b</td><td>Bitwise operators</td></tr>
<tr><td>• clamp
• range</td><td>• clamp (a, min, max)
• range (a, min, max)</td><td>Change input value range</td></tr>
</table>

6.4 User-defined Functions

The (+, -, cos, >, etc.) operators can be considered as "factory" functions. However, you can create your own functions, ie operation prototypes which doesn't give any result by themselves, but which can be called, once defined, by any other variable of your project. These customized functions prove to be quite useful when one reuses complex operations on multiple objects or display configurations.

To customize your functions, you just need to click on the "Expression" icon. In the "name" field, write for example: "my_function" followed by (a,b,c). This particular syntax allows you to declare a new mathematical function. The number of elements in brackets, separated by commas, defines the number of arguments necessary to evaluate the expression. In to our example, "my_function" requires three arguments called "a, b, c". In the "expression" field, you can now write your operation prototype, for example: pow(a*b, c*0.33). Once defined, the function appears in the "function" folder of the Exploration window.



```
Functions
    My_function(a, b, c)=pow(a*b, c*0.33)
```

From now on, the function "my_function" can be conveniently used in any expression inside a project just as a "factory" operator. For example, we could create two faders called "fader_1" and "fader_2" as well as a new variable "my_variable". In the "expression" field of "my_variable", you just need to write "my_function (fader_1.x, 0.5, fader_2.x)" in order to calculate to following operation: pow(fader_1.x*0.5, fader_2.x*0.33).

6.5 Vectors

The mathematical expression parser of the LEMUR allows one to calculate not only decimal values, but also vectors, i.e. lists of decimal values. For obvious gains in efficiency and simplicity, instead of manipulating a great deal of data one after the other, it is often preferable to work with lists of data. Thus, in the case of complex objects such as the multislider, which can include up to 256 parameters, a single object variable is created upon initialization: this variable contains a table where stores the value of each of the sliders in the multislider. Therefore, and expression that uses to the variable "x" of a multislider will have modify a list of numbers representing the slider values.

Now, let's consider the different types of operations that can be performed on vectors. Let's take as example the vector "foo" which contains 6 elements {0.1 0.2 0.4 0.5 0.6} and the vector "bar" which contains 4 elements {0.25 0.5 0.75 1.}.

- bar*2={0.5 1. 1.5 2.} // multiply all the elements of the list by 2
- foo+3= {3.1 3.2 3.3 3.4 3.5 3.6} // adds up to all elements from "foo"
- bar>0.5= {0. 0. 1. 1.} // return 1 as a result for all elements above 0.5, 0 for the others
- bar+foo= {0.35 0.7 1.05 1.4} // adds up all "foo" and "bar" elements together (the size of the output vector is determined by the smaller of the two)

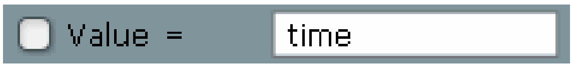
- foo[0]= 0.1 // return the 1st element of "foo"
- bar[2]= 0.75 // return the 3rd element of "bar"
- foo[0.5]= 0.15 // return the value interpolated between the 1st and 2nd element of "foo"
- bar[1.5]= 0.625 // return the value interpolated between the 2nd and 3rd element of "bar"
- {bar,foo}= {0.25 0.5 0.75 1. 0.1 0.2 0.3 0.4 0.5 0.6} // concatenates the vectors "foo" and "bar"
- {bar[0],foo[0],bar[1.5]}= {0.25 0.1 0.625} // creates a new list with the 1st element of bar, the 1st element of foo, and the average of the 2nd and 3rd elements of bar.

6.6 Using Time Variable:

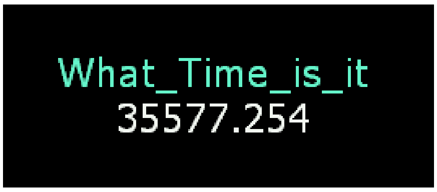
After exploring the mathematic expressions, you might reach a point where you'll love your Lemur to be capable of changing its behavior by itself.

For this purpose, the Lemur includes a very usefull 'factory defined' global variable that can be used by any expression : the Time variable. This variable is always displayed inside the browser window variable folder, but the user is not allowed to remove it nor customize its expression.

The well-called Time variable is a millisecond accurate internal counter that reports the elapsed time in second since the Lemur boot. Thus, after two minutes, five seconds and twenty-four milliseconds, Time will be '125.010'. After one hour, it'll be '3600.000'. Note that after one hour, Time will return '0.000' once again, since the counter loops over one hour. Obviously, you can display Time value by using a monitor object that takes it as its value argument.



□ Value = time



What_Time_is_it
35577.254

At first glance, you might wonder which benefit you could take of such a basic clock. Indeed, the Time variable is not really useful alone. However, in conjunction with basic mathematical and logical operators it will generate grateful time-dependant functions and messages such as metronomes, LFOs, vibratos, envelopes, sequencers, rhythmical textures of any sort, self-oscillating objects, and so on. Actually, this modest add-on dramatically enhances the field of Lemur's capabilities.

Certainly, the operators you will use most often with Time are multiply (*) , (/) and modulus (%).

Thus, multiplying or dividing time by any value is the common way to define a specific tempo: multiply Time by a number greater than one and the counter will accelerate. Not surprisingly, if you apply a division instead, the counter will slow down.

Modulus operator (%, also known as 'rest of division') is also pretty useful: depending on its argument, it will change the counter looping point. For instance, Time%1 will count from 0 to 1, while Time%60 will loop at 60.

By combining both % and * (or /) , you can obtain any time subdivisions and by this way creating counters and clocks of any sorts : seconds, minutes, hours, bars, ticks,, and so on.

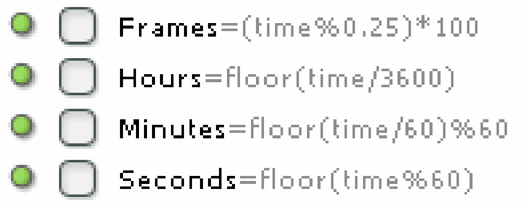
Some examples (TC, LFO, envelope, 16-step sequencer):

The two figures above show you a basic application of Time variable that consists in generating a SMPTE-like synchronisation Time Code.

To start with, create a set of four variables called *Frames, Seconds, Minutes, Hours* For each variable, define the expression described in the first figure.

You can display the resulting Time sub-divisions using several Monitor objects. You can also whether send the variable separately by checking each send box or encapsulate the whole Time Code into a single message called *Time Code* with the following expression:

TimeCode = {Hours, Minutes, Seconds, Frames]

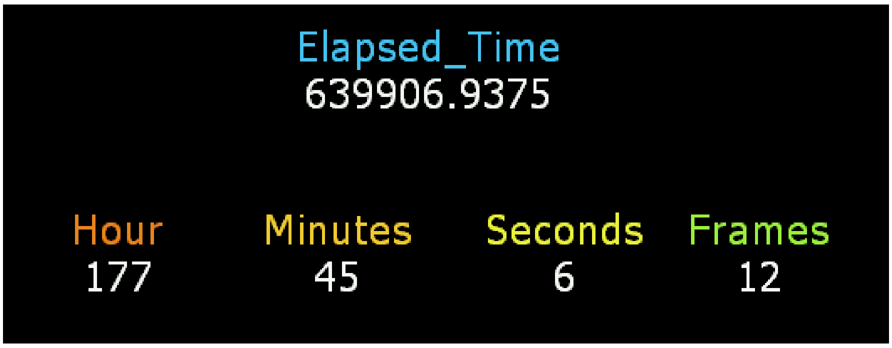


● □ Frames=(time%0.25)*100

● □ Hours=floor(time/3600)

● □ Minutes=floor(time/60)%60

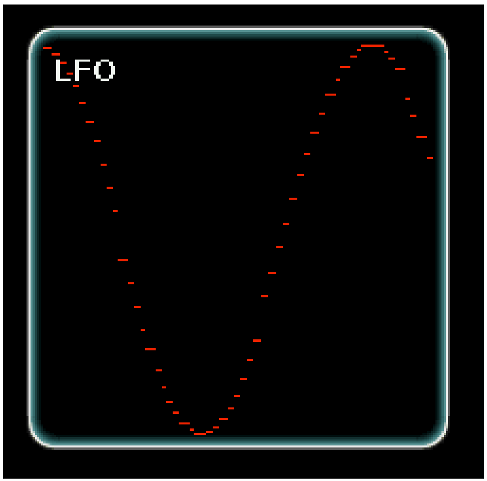
● □ Seconds=floor(time%60)

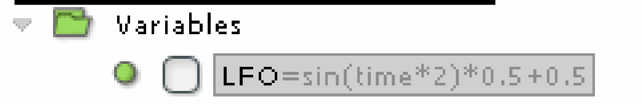


Elapsed_Time
639906.9375

Hour	Minutes	Seconds	Frames
177	45	6	12

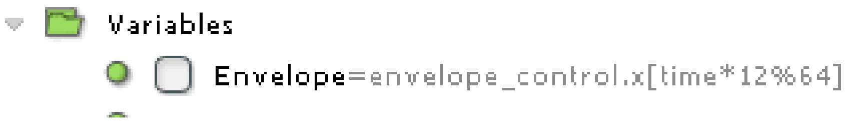
The following example demonstrates how to generate a sine wave low frequency oscillator. The resulting function can be displayed on Lemur thanks to Signalscope object.



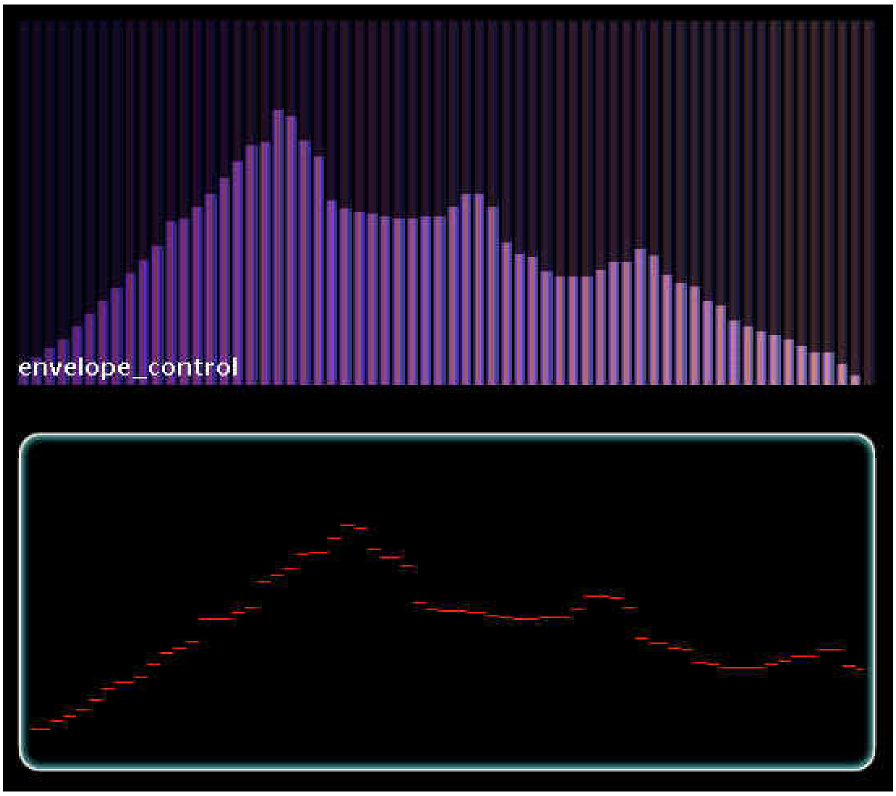


```
LFO=sin(time*2)*0.5+0.5
```

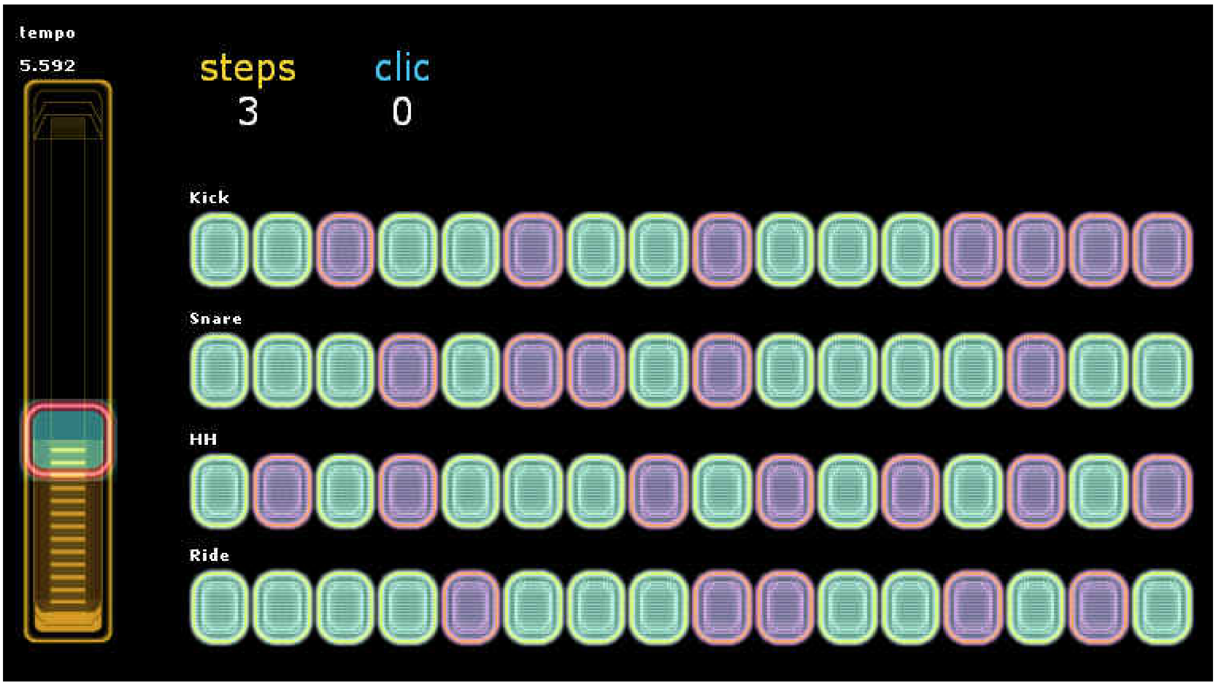
Another cool application for Time variable is the creation of customized envelopes. For that purpose, a multislider object will be helpful.



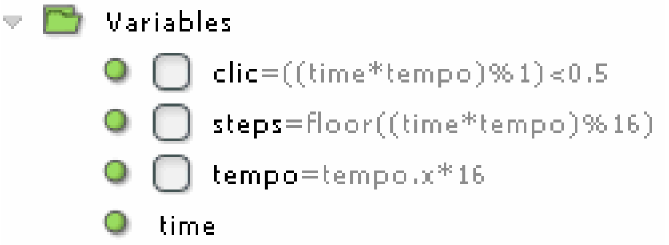
```
Envelope=envelope_control.x[time*12%64]
```



Finally, here is a way to turn your Lemur into a four voices 16-steps sequencer. To start with, create the four sequencing tracks by arranging as many sets of 16 switches as shown above. Adding a fader somewhere in your interface might help to adjust the tempo.



Once you have it done, create the following global variables:



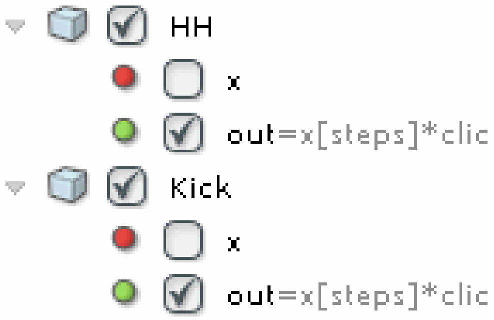
Variables

- □ clic=((time*tempo)%1)<0.5
- □ steps=floor((time*tempo)%16)
- □ tempo=tempo.x*16
- time

The *step* variable counts periodically in a range of 0 to 16. Its frequency depends on the *tempo* variable, which multiplies by 16 the output value of the fader. Thus, the step sequencer global frequency will be between 0 and 16 Hz, which corresponds to a tempo range of 0 (stop) to 960 BPM (low limit of the audio frequency bandwidth).

The *clic* variable features a basic sample-and-hold function to generate note-on and note-off messages. Thus, according to *tempo* variable, it will pick up the values of our four sequencing tracks periodically and sustain them over a period of half a tick.

Now, we can manage to read the 16 switches state of each track one after the other with the following expression:



☑ HH
- □ x
- ☑ out=x[steps]*clic

☑ Kick
- □ x
- ☑ out=x[steps]*clic

Send out the track object *out* variables and play.

7 Informations

Authorized JazzMutant distributor

Europe
JAZZMUTANT
2, allée du Doyen Georges Brus
33600 Pessac, France
Tel : +33 5 56 46 03 44
Fax : +33 5 56 46 17 60
URL : http://www.jazzmutant.com

USA
CYCLING'74 Inc.
379A Clementina Street
SAN FRANCISCO
California, CA 94103 USA
Tel : +1 415 974 1818
Fax : +1 415 974 1812
URL: http://www.cycling74.com

For EU Countries
CE This product complies with the requirements of European Directive 89/336/EEC.

For the USA
THE FCC REGULATION WARNING
This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communication. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Unauthorized changes or modification to this system can void the user's authority to operate this equipment.

For Canada
NOTICE: This Class B digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulation.
AVIS: Cet appareil numérique de la classe B respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.