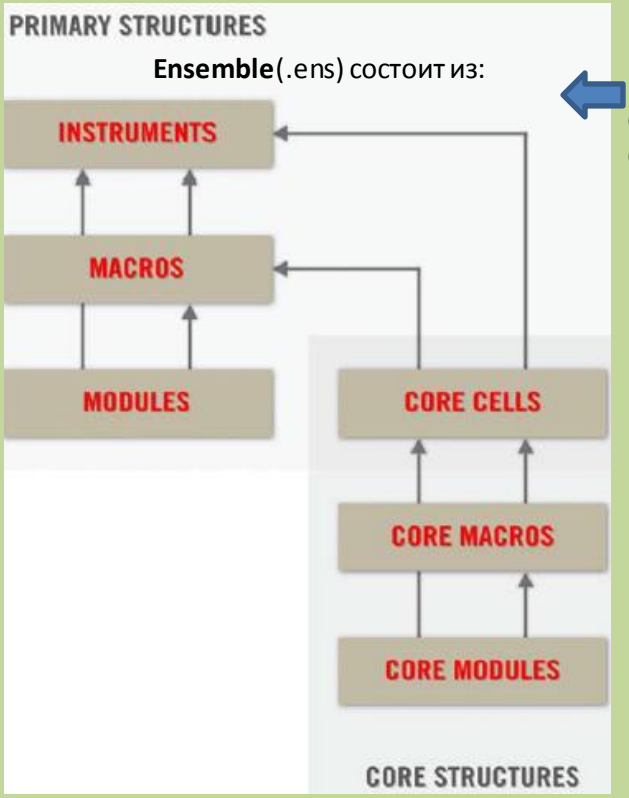
Примеры, описания, схемы.

REAKTOR – это отдельная программа, и виртуальный инструмент, работающий по протоколу VSTi. Это даже не программа, а среда для программиста (сразу отметим, что среда визуальная, а значит, знания языков программирования не требует) для создания новых звуков, синтезаторов, сэпмлеров и т.п. Благодаря модульной структуре, REAKTOR позволяет строить практически любое цифровое устройство, которое только можно вообразить: от простого аналогового синтезатора до сложных систем, использующих несколько видов синтеза и различных обработок.
И самое главное – изучение этой программы позволит вам понять основные принципы создания электронных музыкальных звуков, принципы и логику работы синтезаторов, изучить структуры, формирующие и модифицирующие звуковую информацию. В случае если возможности, заложенные в стандартную библиотеку, вас не устраивают – модульная структура и любые элементы управления доступны для модификации.

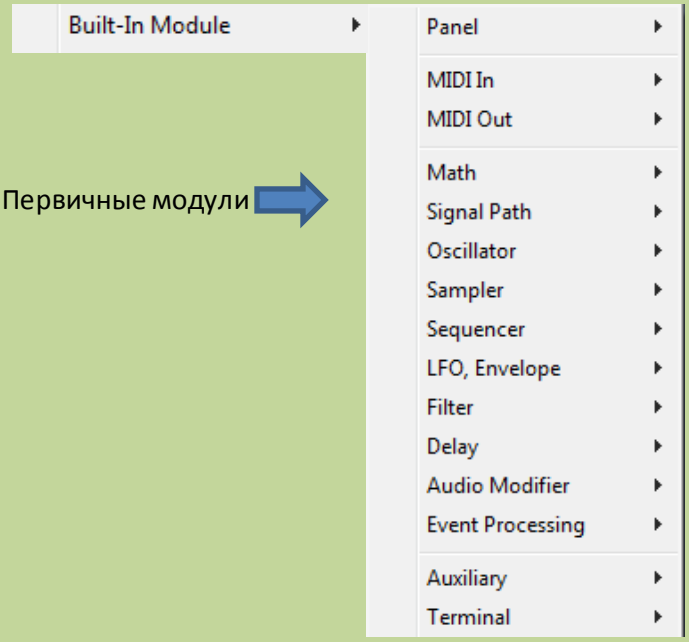
Event Iter

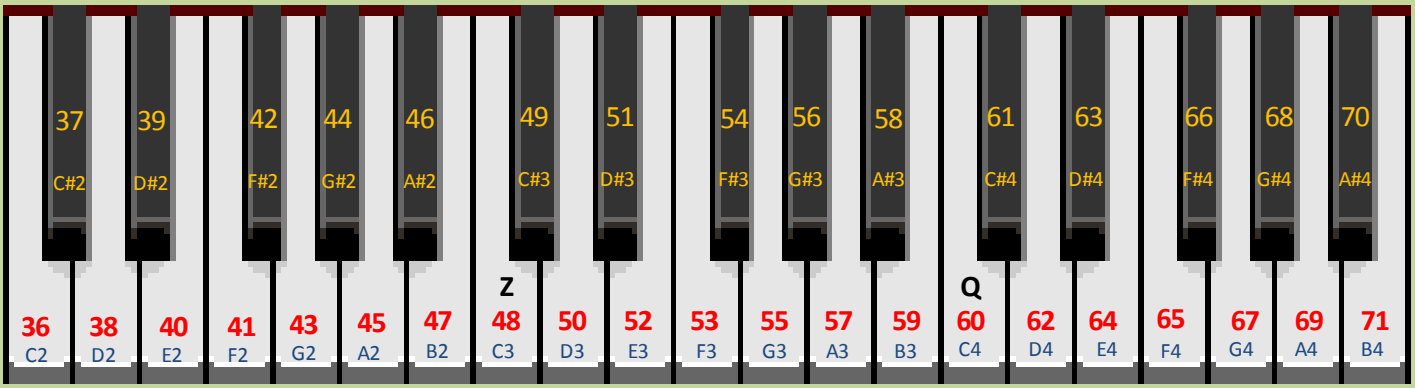
Небольшой обзор по модулям Reaktor



Иерархия :
Instruments(.ism) состоят из macros(.mdl),modules,core cells(.rcc)
Macros(.mdl) состоят из modules,core cells(.rcc)
Core cells(.rcc) состоят из core macros(.rcm),core modules
Core macros(.rcm) состоят из core modules
Modules и core modules это встроенные первичные модули.

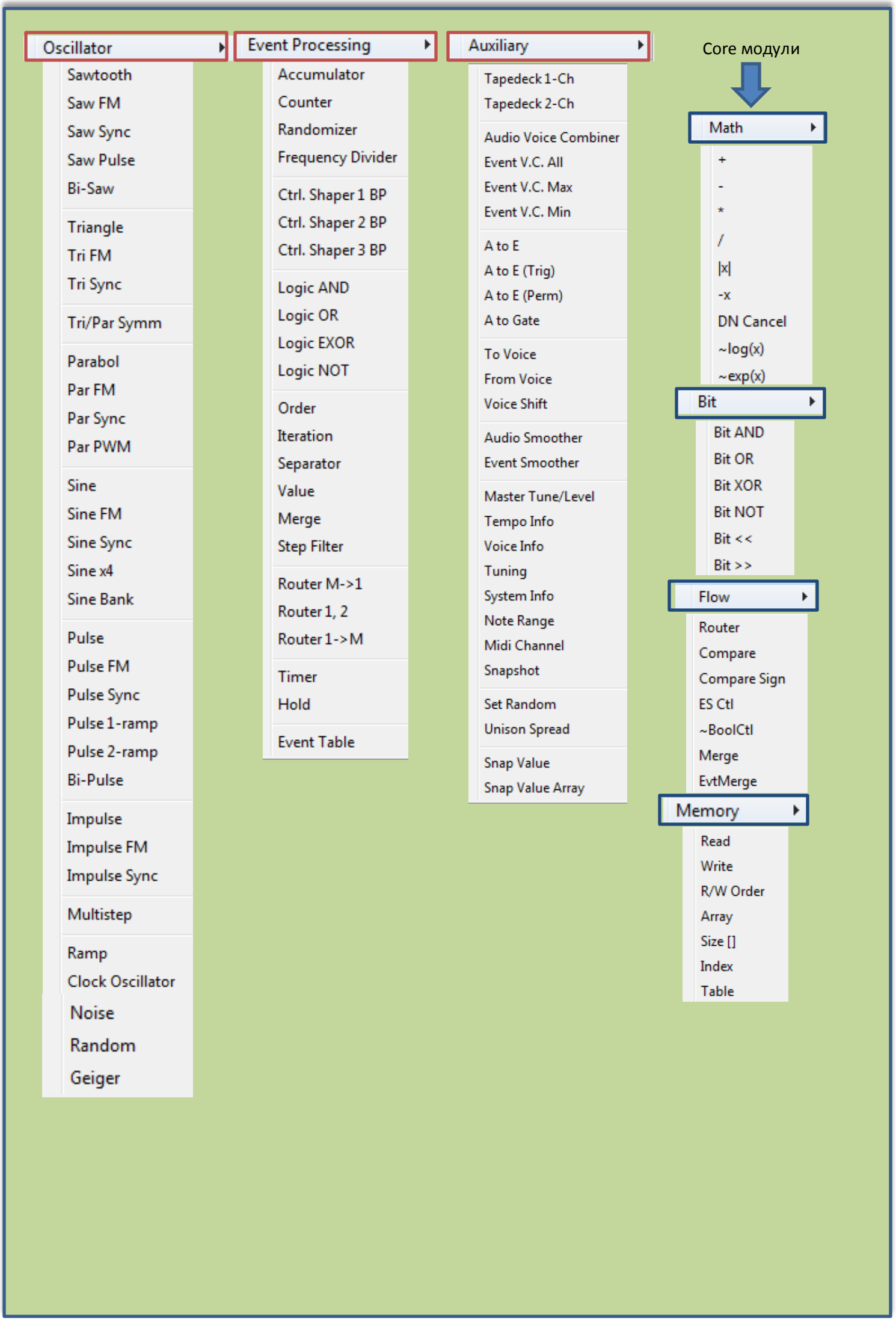
Первичные модули



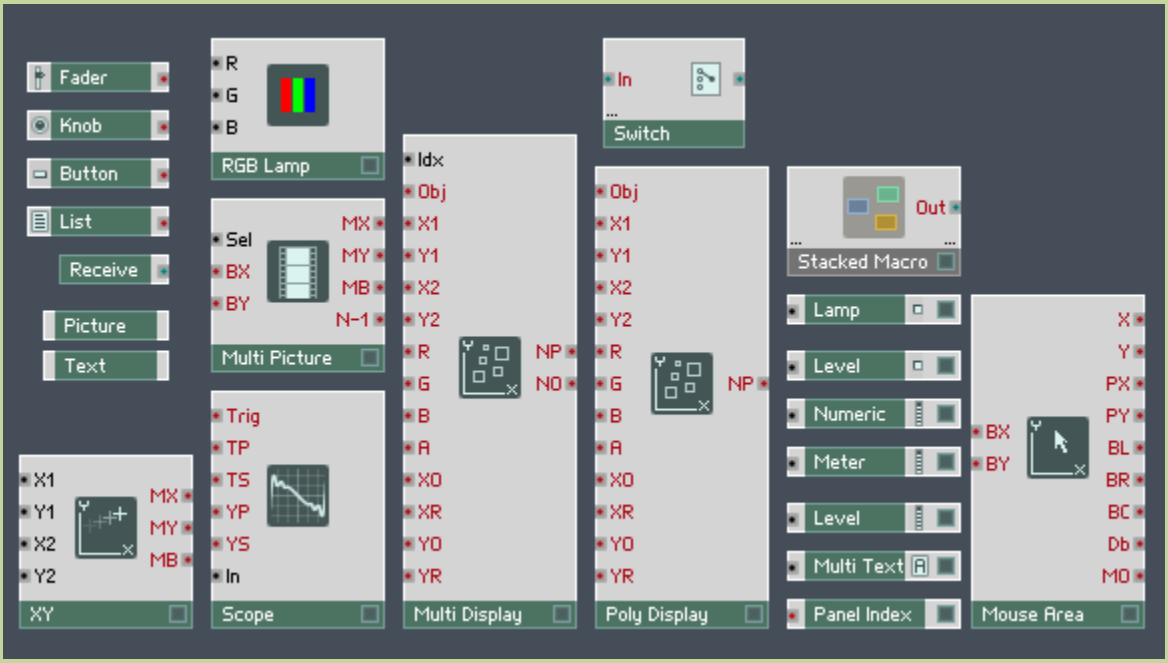


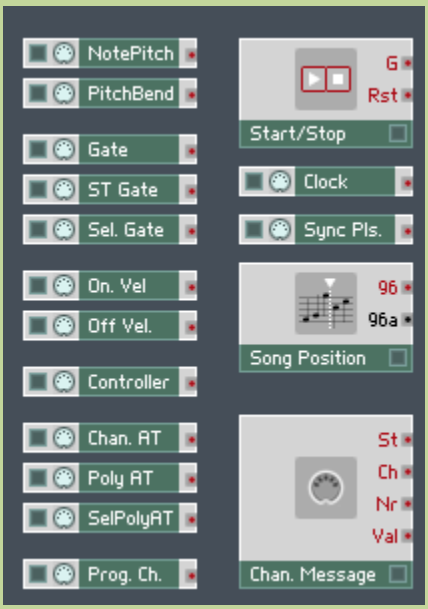
Находим частоту по ноте : F=2^(p/12+3.031) или F=(2^((p-69)/12))*440 или F=(1,05946^p)*8,17742
Значения трёх формул для ноты До первой октавы(C4) : 261.560, 261.625, 261.631
На компьютерной клавиатуре : Z – малая октава, Q – первая октава, I – вторая октава.

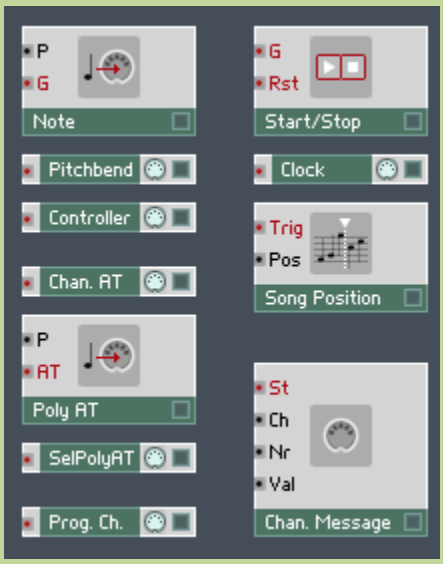

Panel
Fader
Knob
Button
List
Switch
Receive
Lamp
Level Lamp
RGB Lamp
Meter
Level Meter
Numeric Readout
Picture
Multi Picture
Text
Multi Text
XY
Scope
Multi Display
Poly Display
Mouse Area
Stacked Macro
Panel Index
LFO, Envelope
LFO
Slow Random
H
HR
D
DR
DSR
DBDR
DBDSR
AD
AR
ADR
ADSR
ADBDR
ADBDSR
AHDSR
AHDBDR
4-Ramp
5-Ramp
6-Ramp
MIDI In
Note Pitch
Pitchbend
Gate
Single Trig. Gate
Sel. Note Gate
On Velocity
Off Velocity
Controller
Ch. Aftertouch
Poly Aftertouch
Sel. Poly AT
Program Change
Start/Stop
1/96 Clock
Sync Clock
Song Position
Channel Message
Signal Path
Selector/Scanner
Relay 1, 2
Crossfade
Distributor/Panner
Stereo Pan
Amp/Mixer
Stereo Amp/Mixer
Filter
HP/LP 1-Pole
HP/LP 1-Pole FM
Allpass 1-Pole
Multi 2-Pole
Multi 2-Pole FM
Multi/Notch 2-Pole
Multi/Notch 2-Pole FM
Multi/LP 4-Pole
Multi/LP 4-Pole FM
Multi/HP 4-Pole
Multi/HP 4-Pole FM
Pro-52
Ladder
Ladder FM
Modal Bank
Peak EQ
Peak EQ FM
High Shelf EQ
High Shelf EQ FM
Low Shelf EQ
Low Shelf EQ FM
Differentiator
Integrator
MIDI Out
Note Pitch/Gate
Pitchbend
Controller
Ch. Aftertouch
Poly Aftertouch
Sel. Poly AT
Program Change
Start/Stop
1/96 Clock
Song Position
Channel Message
Sampler
Sampler
Sampler FM
Sampler Loop
Grain Resynth
Grain Pitchformer
Grain Cloud
Beat Loop
Sample Lookup
Sequencer
6-Step
8-Step
12-Step
16-Step
Multiplex 16
Delay
Single Delay
Multi-Tap Delay
Diffuser Delay
Grain Delay
Grain Cloud Delay
Unit Delay
Terminal
In Port
Out Port
Send
Receive
IC Send
IC Receive
OSC Send
OSC Receive
OSC Send Array
OSC Receive Array
Math
Constant
Add
Subtract
Invert, -x
Multiply
Mult/Add, a*b+c
Reciprocal, 1/x
Divide, x/y
Modulo, x%y
Rectify, |x|
Rectify/Sign
Compare
Compare/Equal
Quantize
Expon. (A)
Expon. (F)
Log. (A)
Log. (F)
Power, x^y
Square Root
1 / Square Root
Sine
Sine/Cosine
ArcSin
ArcCos
ArcTan
Audio Modifier
Saturator
Saturator 2
Clipper
Mod. Clipper
Mirror 1 Level
Mirror 2 Levels
Chopper
Shaper 1 BP
Shaper 2 BP
Shaper 3 BP
Shaper Parabolic
Shaper Cubic
Slew Limiter
Peak Detector
Sample + Hold
Frequency Divider
Audio Table

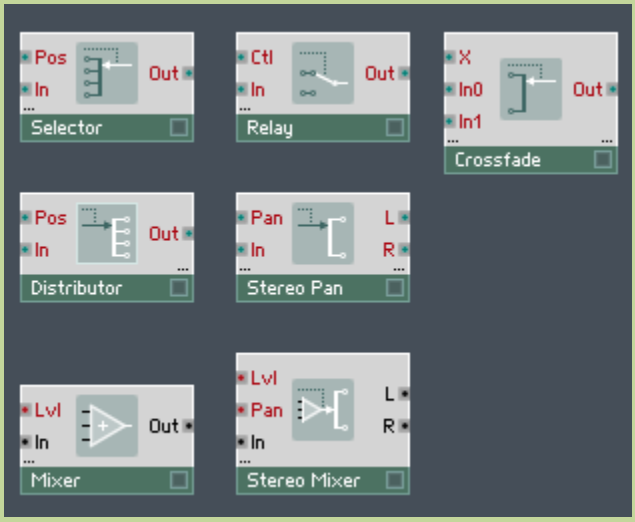

Oscillator
Sawtooth
Saw FM
Saw Sync
Saw Pulse
Bi-Saw
Triangle
Tri FM
Tri Sync
Tri/Par Symm
Parabol
Par FM
Par Sync
Par PWM
Sine
Sine FM
Sine Sync
Sine x4
Sine Bank
Pulse
Pulse FM
Pulse Sync
Pulse 1-ramp
Pulse 2-ramp
Bi-Pulse
Impulse
Impulse FM
Impulse Sync
Multistep
Ramp
Clock Oscillator
Noise
Random
Geiger
Event Processing
Accumulator
Counter
Randomizer
Frequency Divider
Ctrl. Shaper 1 BP
Ctrl. Shaper 2 BP
Ctrl. Shaper 3 BP
Logic AND
Logic OR
Logic EXOR
Logic NOT
Order
Iteration
Separator
Value
Merge
Step Filter
Router M->1
Router 1, 2
Router 1->M
Timer
Hold
Event Table
Auxiliary
Tapedeck 1-Ch
Tapedeck 2-Ch
Audio Voice Combiner
Event V.C. All
Event V.C. Max
Event V.C. Min
A to E
A to E (Trig)
A to E (Perm)
A to Gate
To Voice
From Voice
Voice Shift
Audio Smoother
Event Smoother
Master Tune/Level
Tempo Info
Voice Info
Tuning
System Info
Note Range
Midi Channel
Snapshot
Set Random
Unison Spread
Snap Value
Snap Value Array
Core модули
Math
+
-
*
/
|x|
-x
DN Cancel
~log(x)
~exp(x)
Bit
Bit AND
Bit OR
Bit XOR
Bit NOT
Bit <<
Bit >>
Flow
Router
Compare
Compare Sign
ES Ctl
~BoolCtl
Merge
EvtMerge
Memory
Read
Write
R/W Order
Array
Size []
Index
Table

Первичные модули





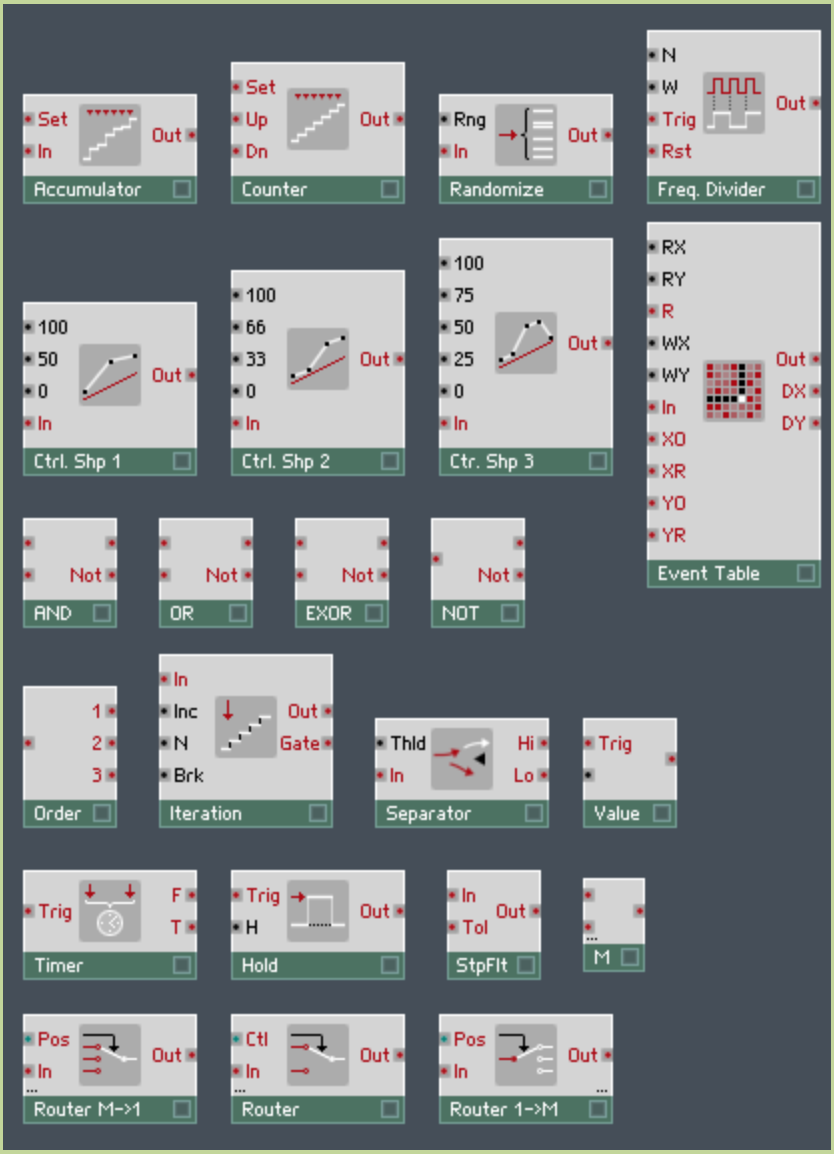





P
A
Out
Sawtooth
Triangle
Parabol
F
Snc
Ph
Sine Sync
W
Pulse FM
Impulse
Saw FM
Tri FM
Par FM
Impulse FM
Saw Sync
Tri Sync
Par Sync
Impulse Sync
F1
A1
F2
A2
F3
A3
F4
A4
S1
S2
S3
S4
Sine x4
Pulse Sync
Slp
Saw Pulse
Tri
Par
Tri/Par Symm
Par PWM
Pulse 1-Ramp
S5
5-Ramp
Bi-Saw
Ramp Osc
Clock Osc
Num
Idx
Rtio
AL
AR
App
L
R
Max
Sine Bank
Up
Dn
Pulse 2-Ramp
Random
Noise
Rnd
Clk
Geiger
Sine
Bi-Pulse
5-Step
Pulse
Sine FM
Trig
Sampler
St
Len
Sampler FM
G
Sel
LS
LL
Sampler Loop
Lookup
D/F
PJ
PS
Pos
PsJ
Len
LnJ
Att
Dec
Dist
DisJ
Pan
PnJ
GTr
GLn
Grain Cloud
Sp
Gr
SO
Sm
Resynth
FS
Pitch Former
L16
P16
Ct16
Beat Loop
Rst
GA
Sequencer
Bar
Multiplex 16


F
A
W
Snc
Ph
Sin
Tri
Pls
LFO
Trig
A
H
Out
H - Env
Trig
A
D
Out
D - Env
Trig
A
H
R
Out
HR - Env
G
D1
B
D2
R
Out
DBDR - Env
G
A
D
S
R
Out
ADSR - Env
G
A
D1
B
D2
R
Out
ADBDR - Env
G
D
R
Out
DR - Env
Trig
A
A
D
Out
AD - Env
G
D
S
R
Out
DSR - Env
G
A
D
R
Out
ADR - Env
G
D1
B
D2
S
R
Out
DBDSR - Env
F
A
Out
Slow Random
G
A
R
Out
AR - Env
G
T1
L1
T2
L2
T3
L3
T4
L5
TR
Out
St
5-Ramp - Env
G
T1
L1
T2
L2
T3
L4
T5
L5
TR
Out
St
6-Ramp - Env
G
A
H
D
S
R
Out
AHDSR - Env
G
A
H
D1
B
D2
R
Out
AHDBDR - Env
G
T1
L1
T2
L2
T3
LS
TR
Out
St
4-Ramp - Env
G
A
D1
B
D2
S
R
Out
ADBDSR-Env
1
In
Out
1
Receive
Send
Out
...
OSC Receive
In
...
OSC Send
IC Receive
IC Send
In
Dly1
Dly2
Dly3
Dly4
Dly5
Dly6
Dly7
Dly8
1
2
3
4
5
6
7
8
Multi-Tap Delay
Dly
Dffs
In
Out
Diffuser Delay
Dly
In
Out
Single Delay
P
Dly
Gr
Sm
Pan
A
In
L
R
Dly
Grain Delay
T
G
Frz
P
D/F
PJ
PS
Dly
DlJ
Len
LnJ
Att
Dec
Dist
DisJ
Pan
PnJ
A
In
L
R
Dly
GTr
GLn
Cloud Delay
In
Out
Saturator
LA
KH
KHA
Offs
In
Out
Saturator 2
Max
Min
In
Out
Clipper
M
In
Out
Mod. Clipper
Max
In
Out
Mirror 1
Max
Min
In
Out
Mirror 2
M
X
In
Out
Chopper
Bp
Sl
In
Out
Shaper 1 BP
Bp1
Sl1
Bp2
Sl2
In
Out
Shaper 2 BP
Bp1
Sl1
Bp2
SL2
Sl+
Sl-
In
Out
Shaper 3 BP
Lin
Par
In
Out
Parabol Shp
Lin
Par
Cub
In
Out
Cubic Shp
C+
C-
A
In
Out
Freq. Divider
RX
RY
WX
WY
In
W
XO
XR
YO
YR
Out
DX
DY
Audio Table
Up
Dn
In
Out
Slew Limiter
Rel
In
Out
Peak Detect.
Trig
In
Out
Sample & H.


P
In
HP
LP
1-Pole Filter
F
Out
1-Pole Allpass
Res
BP
N
2-Pole Notch
2-Pole Filter
HP2
BP2
LP2
HP4
BH4
BP4
4-Pole HP
4-Pole Filter
Nurr
Idx
Rtio
Gln
AL
AR
Dec
Dmp
App
InC
L
R
Max
Modal Bank
LP1
LP3
LP4
Pro-52 Filter
Ladder Filter
B
Hi Shelf EQ
Lo Shelf EQ
Peak EQ
Set
Differentiator
Integrator
Tempo Info
A
A to E Perm
G
A to Gate
V
To V
Fr. V
Wrp
Shft
Voice Shift
S
E }
}
A/E
Rec
Play
Lp
In L
In R
Pse
Pos
Spd
Len
Tapedeck 2-Ch
Tun
Lvl
Master
Trig
A to E Trig
Select Max
Select Min
Min
Max
Vcs
Voice Info
Snap Value
Spr
Tuning
ICh
OCh
Midi Channel
Set Random
Upr
Lwr
Note Range
Spread
Snp
Bnk
Recl
St
Mph
Sw
MT
Rnd
Amt
Mrg
Snapshot
W
Snap Value []
SR
CR
DClk
CPU
System Info
Tapedeck 1-Ch



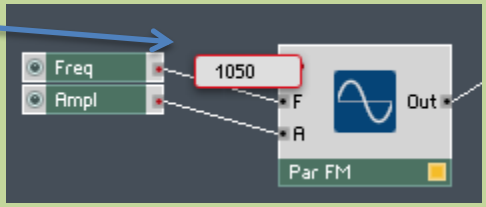
Как видно по модулям порты имеют три вида :

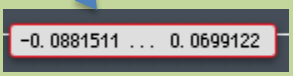
Аудио порт – чёрная точка

Евент порт – красная точка

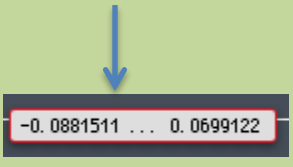
Гибридный порт – зелёная точка (можно комбинировать аудио и евент сигналы, умножать и т.д)

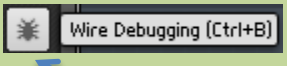
Обычно аудио порты для управления (F,A) имеют тип
но, на них в некоторых случаях можно подавать
непосредственно сам аудио-сигнал





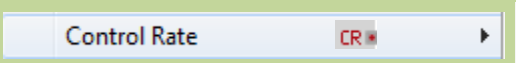
Данные : **Аудио**, **Евент**, **Миди**



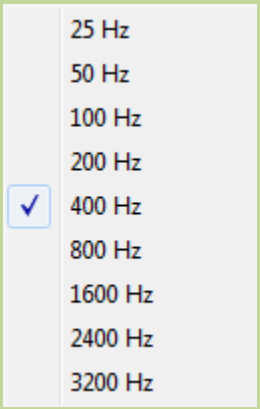


Можно увидеть если нажать и навести на провода в схеме

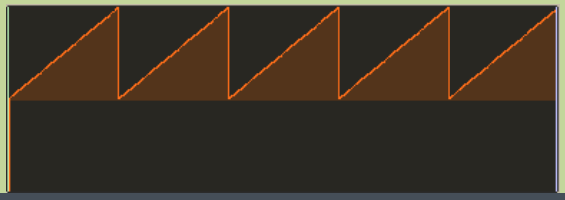
Частота сэмплирования - количество сэмплов в секунду при записи звука, чем выше, тем точней звук. 44100 сэмплов в секунду - стандартное значение. Длительность : 10 сэмплов, 10/44100=0.0002секунд=0.2миллисекунд. Примерный размер файла при записи (частота 44100 Hz) : Sec * 172 в килобайтах (Sec-длительность записи). 1024Kb=1Mb, если ответ свыше тысячи то разделите на 1024 и получите ответ в мегабайтах.
Подробнее :
Since for internal processing the audio information contained in the sample files is converted to 32-bit, each sample uses up 4 bytes of RAM. Hence, the amount of RAM that a one second long sample file uses up at a sampling rate of 44'100 Hertz is 44'100 * 4 = 176'400 bytes, which is a little bit more than 172 Kb (a kilobyte has 1024 bytes). It follows, that for a rough estimate of the total RAM usage of your sample at 44'100 Hertz Sample Rate, you should multiply the length of your sample in seconds by 172 Kb / sec.

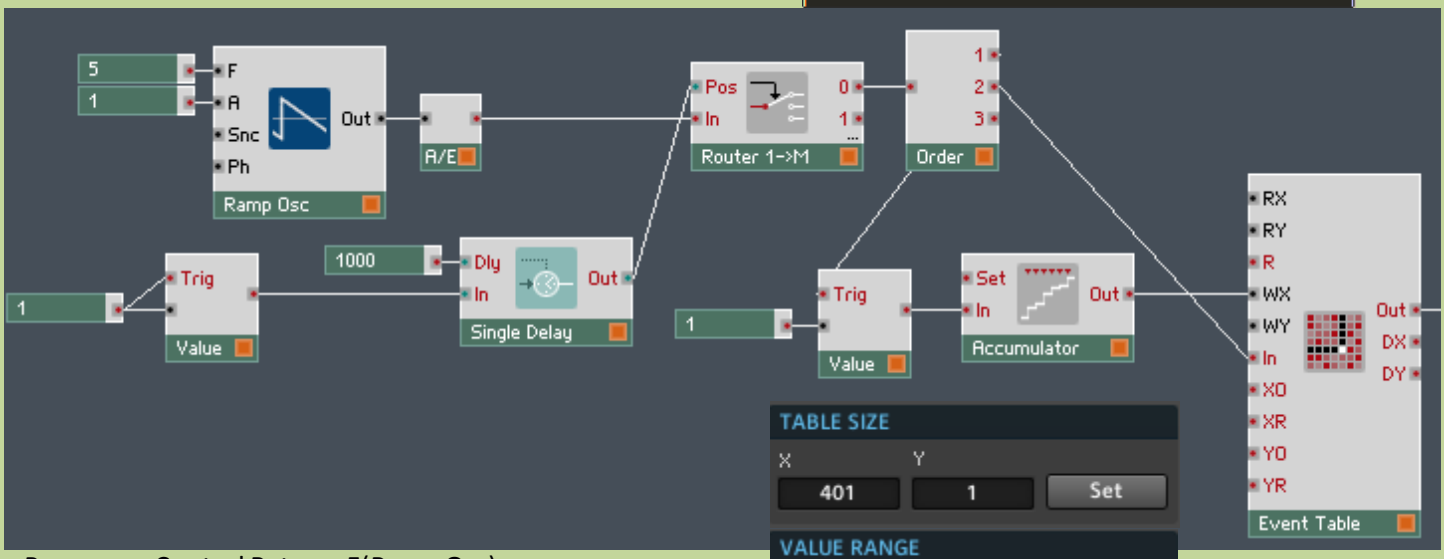


Частота евент процесса в секунду

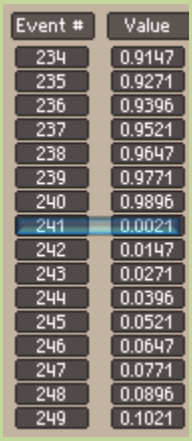


Пример : Нажмём 2 раза  и получим на Event Table





Разделим Control Rate на F(Ramp Osc)
CR/F=400/5=80 – пять периодов по 80 event значений
Начало любого периода n*80+1
3*80+1=241 – начало третьего периода



Event #	Value
234	0.9147
235	0.9271
236	0.9396
237	0.9521
238	0.9647
239	0.9771
240	0.9896
241	0.0021
242	0.0147
243	0.0271
244	0.0396
245	0.0521
246	0.0647
247	0.0771
248	0.0896
249	0.1021

Если сменить Control Rate то значения и расчёты будут другими так как частоту сменили.

Конвертация и нахождение некоторых значений :

ms→Samples (SR*T)
Пример : сколько нужно сэмплов чтобы записать 20 секундный фрагмент (частота по умолчанию=44100Гц)
44100*20=882000

bmp→dur (beat*60/bmp)

beat 

Label	Value
1	4
1 t	3
2	2
2 t	1.5
4	1
4 t	0.75
8	0.5
8 t	0.375
16	0.25
16 t	0.1875
32	0.125
32 t	0.09375
64	0.0625
64t	0.046875
128	0.03125

Пример : какова будет длительность четвертных нот в миллисекундах при bmp=87
1*60/87=0.689 секунд =689 миллисекунд

Bmp→bps (bmp/60)
Пример : сколько ударов в секунду будет при bmp=120
120/60=2 удара в секунду

Frequency : F=2^(p/12+3.031) или F=(2^((p-69)/12))*440 или F=(1,05946^p)*8,17742

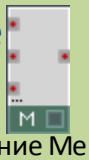
Pitch : 69+12*((log(f/440))/log2)

Числа имеют запись : 0,1,10,100,1000,10000,100000,**1E+006**,**1E+007**. Также и для отрицательных.

Единственный минус Реактора это то, что с выходом новой версии некоторые старые модули могут неправильно работать, но это бывает редко. Нет также десятичного и натурального логарифмом и логарифмов с базой среди первичных модулей , но их можно создать самому (хотя в core-cell есть логарифм, но у него нельзя менять базу в реалтайме).

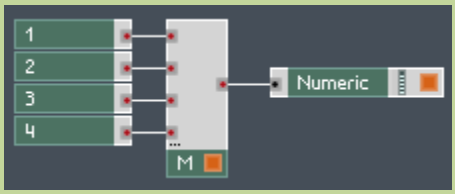
Merge, Value, Order, Iteration, Idx.

Merge



Назначение Merge – транспортировать на выход Event события в порядке их поступления на вход. Входов может быть несколько, а выход один. Допустим, что на первый вход пришло значение 32 от 1 источника, через 1 секунду на 3-ий вход пришло значение 64 от второго источника, а через 10 миллисекунд после второго значения на вход 2 пришло значение 128 от третьего источника, тогда выход будет : 32, 64 через 1секунду, 128 через 10 миллисекунд после значения 64.

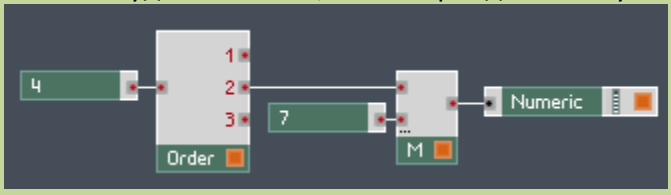
Merge имеет одну важную особенность – если создать такую схему, сохранить, выйти из программы, и снова открыть, то можно заметить, что при открытии файла Merge всегда посылает значение с нижнего входа. Когда вы откроете файл, то на выходе будет только значение 4.



Но это происходит только тогда, когда источник Event значений не является активным (в данном случае это константы), активным в том смысле , что не произошло никаких Event манипуляций или вы ничего не нажали. Здесь важно знать такую таблицу :

Modules that always send events in response to a global reset.	*Modules that may send an event during a global reset if they are wired to receive an event. Check with EventWatcher if needed!*		*Modules that never send events in response to a global reset*
Buttons	1/Square Root	Randomize	1/96 Clock
Channel Aftertouch	Accumulator	Reciprocal	4 Ramp Envelope
Channel Message	Add	Rectify	5 Ramp Envelope
Constant	ArcCos	Rectify/Sign	6 Ramp Envelope
Controller	ArcSin	Relay 1,2	6 Step Sequencer
Event Smoother	ArcTan	Routers	8 Step Sequencer
Faders	Compare	Select Max	12 Step Sequencer
Gate	Compare/Equal	Select Min	16 Step Sequencer
Knobs	Control Shaper 1 BP	Selector/Scanner	A to E Perm
Lists	Control Shaper 2 BP	Separator	A to E Trig
MIDI Channel	Control Shaper 3 BP	Sine	Audio Table
Modal Bank	Counter	Sine/Cosine	Beat Loop
Multi Display	Crossfade	Single Delay	Clock Oscillator
Note Pitch	Distributor/Panner	Square Root	Event Table
Note Range	Divide	Step Filter	Geiger
Off Velocity	Event Voice Combiner	Stereo Pan	Grain Cloud
On Velocity	Expon (A)	Subtract	Grain Cloud Delay
Pitchbend	Expon (F)	To Voice	Grain Delay
Poly Aftertouch	Frequency Divider	Timer	Grain Pitchformer
Poly Display	From Voice	Value	Grain Resynth
Sample Lookup (if a sample is loaded)	Invert		Hold Envelope
Sampler FM (if a sample is loaded)	Iteration		LFO
Sampler Loop (if a sample is loaded)	Log (A)		Mouse Area
Sel. Note Gate	Log (F)		Program Change
Sel. Poly Aftertouch	Logic AND		Slow Random
Sine Bank	Logic EXOR		Song Position
Single Trigger Gate	Logic NOT		Sync Clock
Snap Value	Logic OR		
Snap Value Array (reads out the whole array)	Master Tune/Level		
Snapshot (Snp, Bnk, and Amt outputs only)	Merge		
Start/Stop (G output only)	Modulo		
System Info	Multiplex 16		
Tapedeck 1+2 Channels (Len output only)	Multiply		
Tempo Info	Mult/Add		
Tuning	Order		
Voice Info	Power x^y		
XY module (X and Y outputs only)	Quantize		

Если один из сигналов будет активным, то есть пройдёт манипуляцию , то при открытие файла выход с Merge будет другим



Выход : 4,7

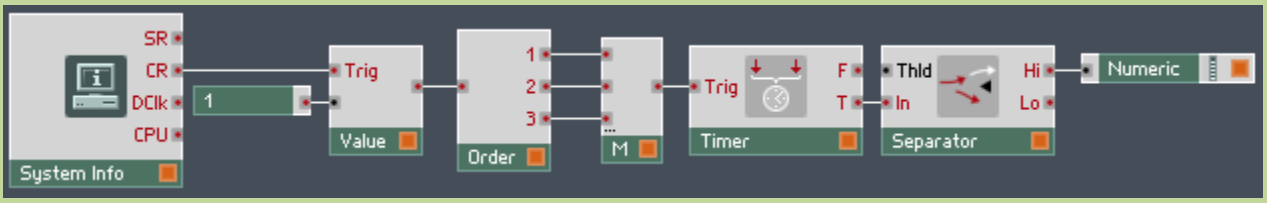
Value 

Value работает как револьвер – нижний вход как барабан, куда вставляется патрон, а верхний вход как спусковой крючок. Пока мы не нажмём Trig, значение с нижнего входа не выйдет через выход. Если на нижнем входе ничего не подключено, то нажимая Trig выход будет = 0

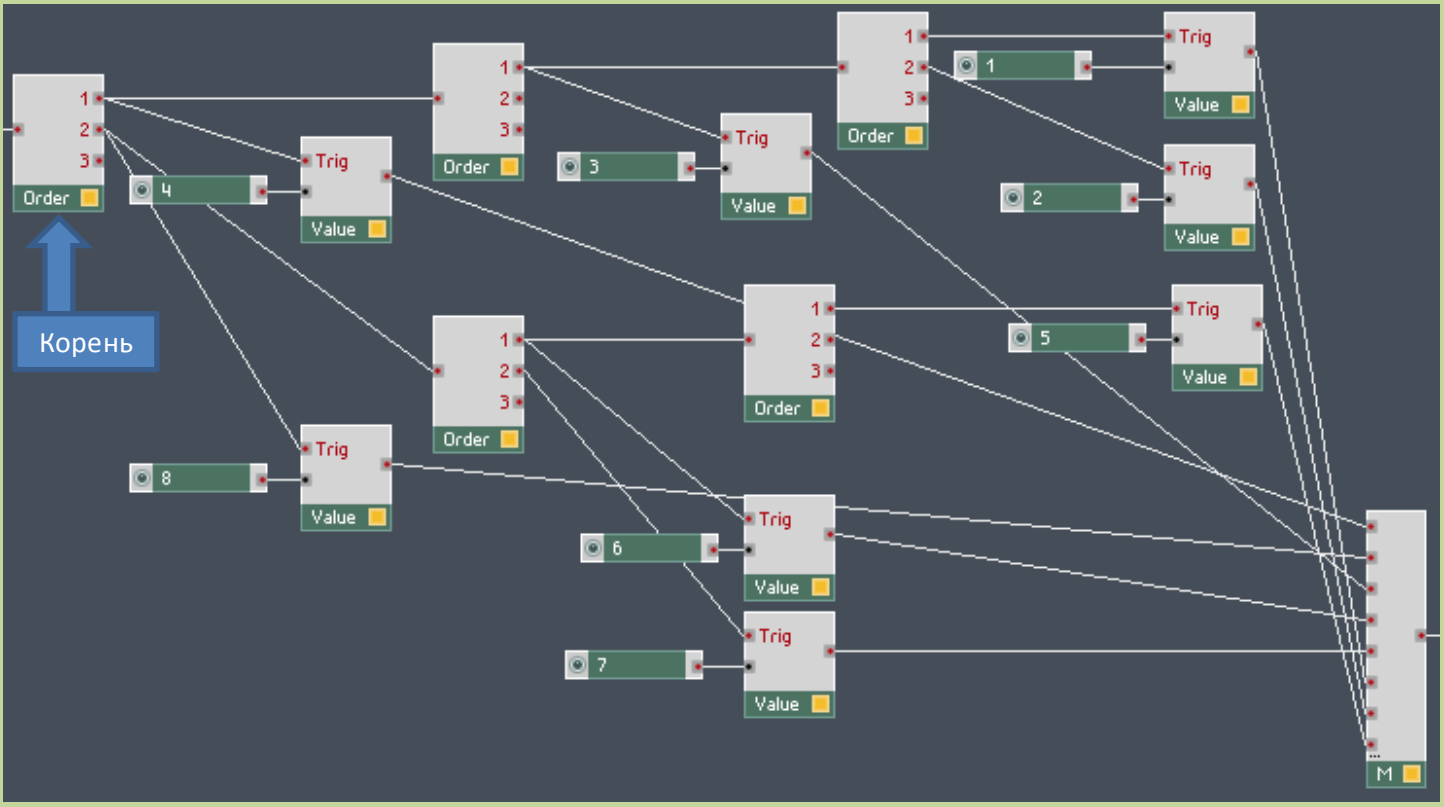
Order 

Назначение Order – распределять сигнал в нужном порядке. Если сигнал прийдёт на вход, то выход будет такой : сначала он выйдет с выхода 1, потом с 2, потом с 3. Время между выходами будет = 2.5 миллисекундам, при Current Rate=400 Hz

Проверка :



Order в множестве :

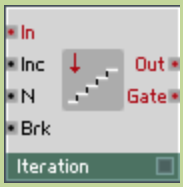


Если мы посмотрим на эту схему, то возникнет вопрос, что будет на выходе Merge первым, вторым и т.д ?
Здесь такое правило : сначала находится корневой Order (на него ничего не приходит с других Order, но он сам посылает сигнал на другие Order). Далее смотрится куда он посылает с 1 выхода, если на другой Order , то смотрится куда уже второй Order посылает с выхода 1, пока не дойдём до последнего. Этот последний и будет самым первым выходом.
Тогда согласно схеме получим :

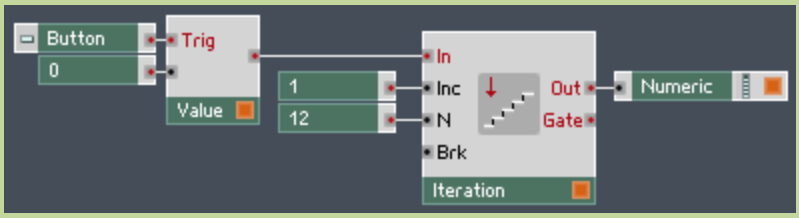
1. 1.1.1
2. 1.1.2
3. 1.1
4. 1
5. 2.1.1
6. 2.1
7. 2.2
8. 2

Синим выделен корневой Order

Iteration



Iteration является повторителем, входящий сигнал повторяется некоторое количество раз согласно входу N и приращению Inc к каждому шагу. Время между итерациями = 2.5 миллисекундам при Current Rate=400 Hz.



Создайте Button и в опциях выберите режим Trigger. Когда мы нажмём на Button то выход с Out модуля Iteration будет :

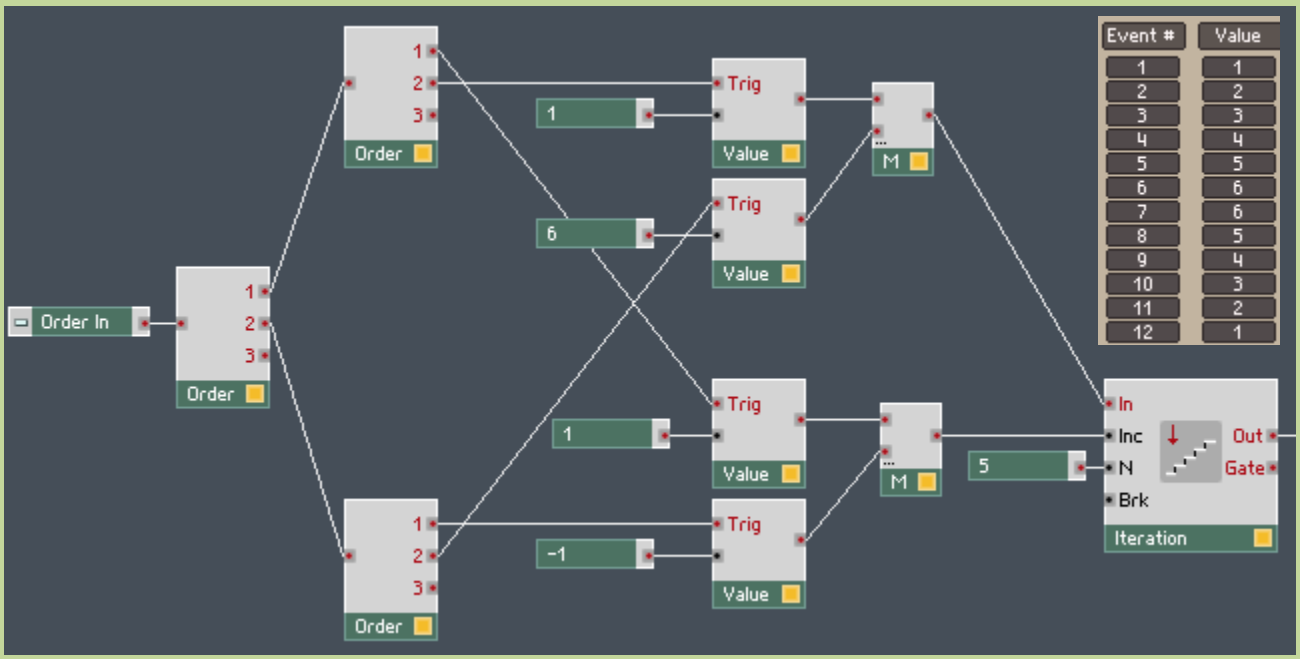
Event #	Value
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10
12	11
13	12

Как видно первым идёт 0, который мы послали нажатием Button через Value, дальше уже идёт итерация. К 0 прибавляется 1, так как у Inc=1. Итак 12 шагов с приращением 1. Итого 13 событий.

Если выбрать Inc=2 то будет :

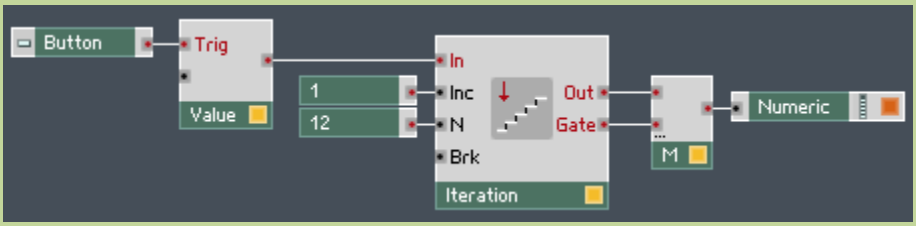
Event #	Value
1	0
2	2
3	4
4	6
5	8
6	10
7	12
8	14
9	16
10	18
11	20
12	22
13	24

Создание пирамидальной итерации. Как видно вход Inc должен измениться первым, а затем вход In. (Если входы Inc, N изменяются , то они должны изменится раньше чем вход In)



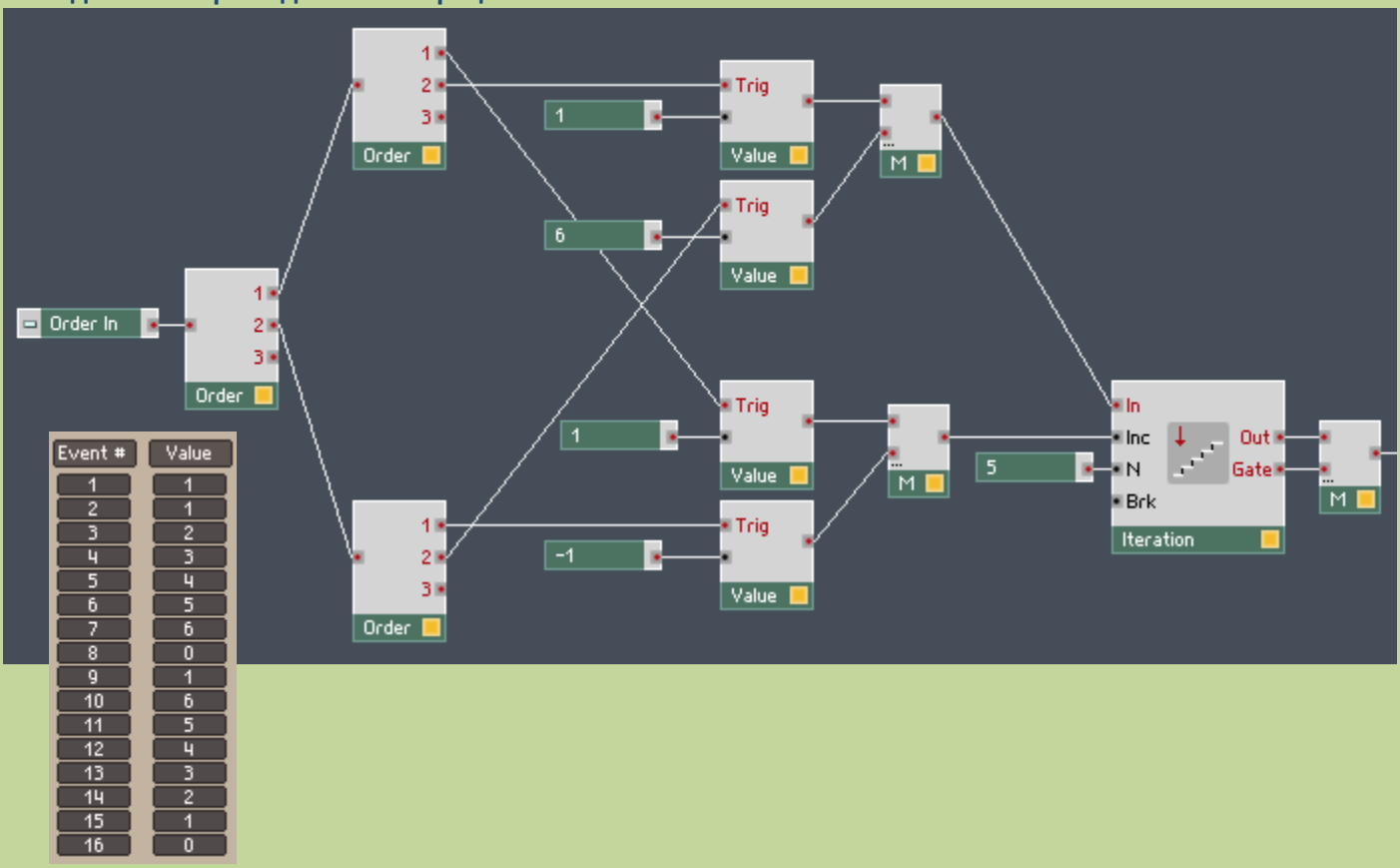
Event #	Value
1	1
2	2
3	3
4	4
5	5
6	6
7	6
8	5
9	4
10	3
11	2
12	1

Gate у Iteration посылает 1 раньше чем начнётся выход с Out, когда выход с Out закончится то Gate пошлёт 0.



Event #	Value
1	1
2	0
3	1
4	2
5	3
6	4
7	5
8	6
9	7
10	8
11	9
12	10
13	11
14	12
15	0

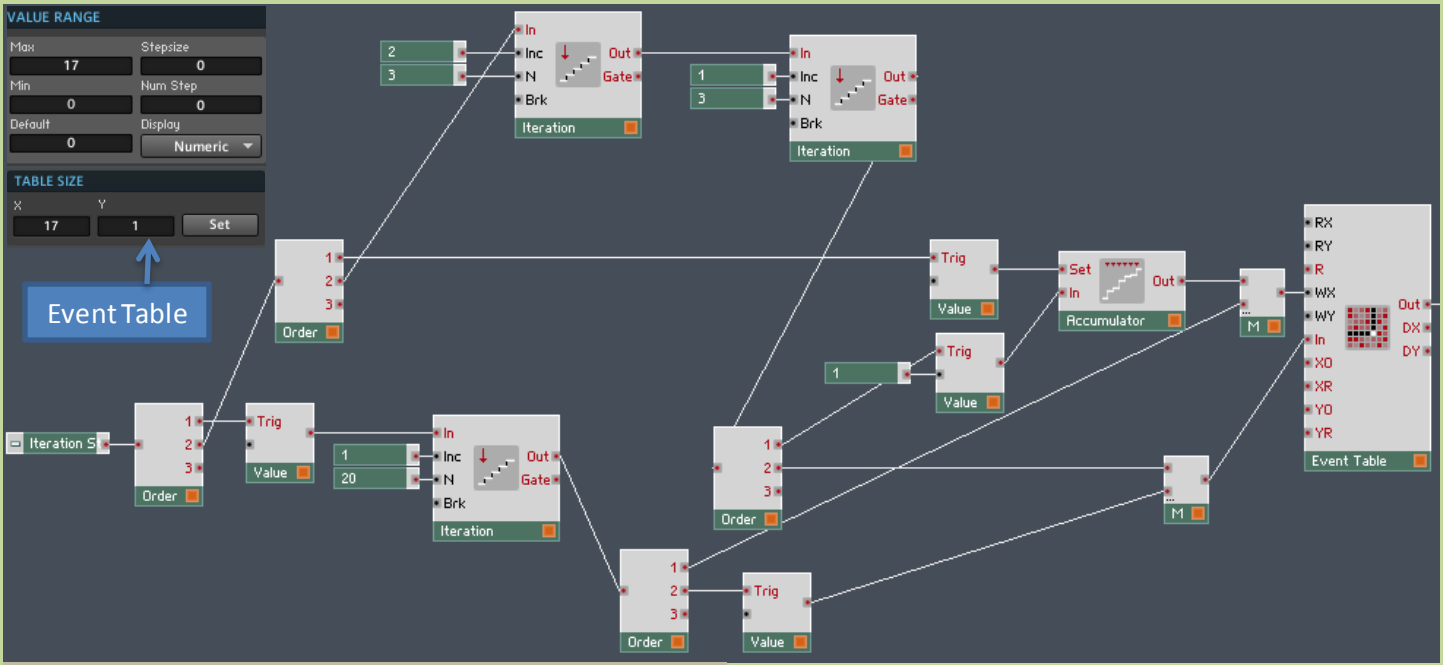
Выход Gate и пирамидальная итерация.



Event #	Value
1	1
2	1
3	2
4	3
5	4
6	5
7	6
8	0
9	1
10	6
11	5
12	4
13	3
14	2
15	1
16	0

Brk – при положительном значении на входе служит для остановки итерации, с выхода будет поступать только первое значение, а итерации не будет.

Итерация проходит через итерацию – рекурсия.(рекурсия-процесс повторения чего либо)
Сначала в этой схеме происходит очистка, затем запись.



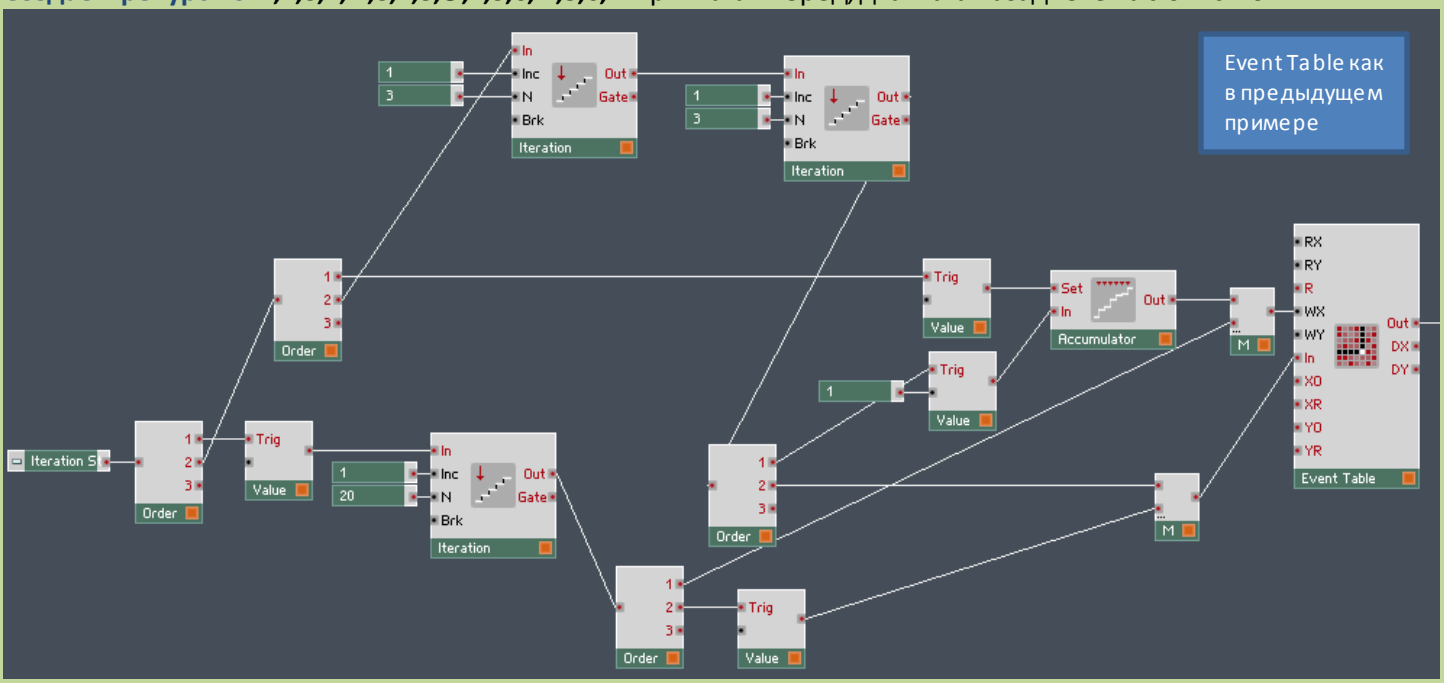
Как видно у нас получилась рекурсия :

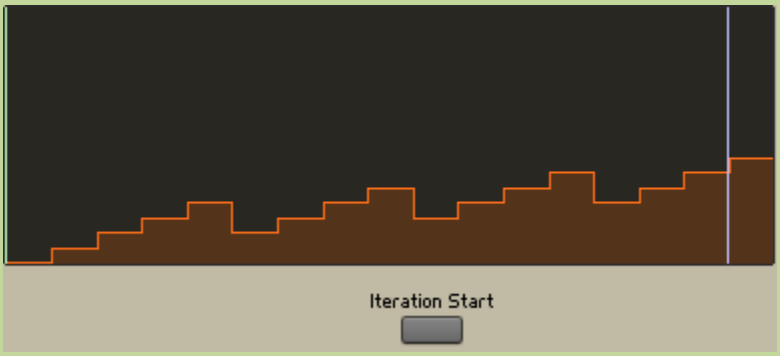
1,2,3,4,3,4,5,6,5,6,7,8,7,8,9,10

Чёрным выделено значения первого модуля итерации.Если мы пустим Counter-счётчик от 1 до 16 по RX,R модуля Event Table, то будем иметь на выходе эти значения рекурсии.

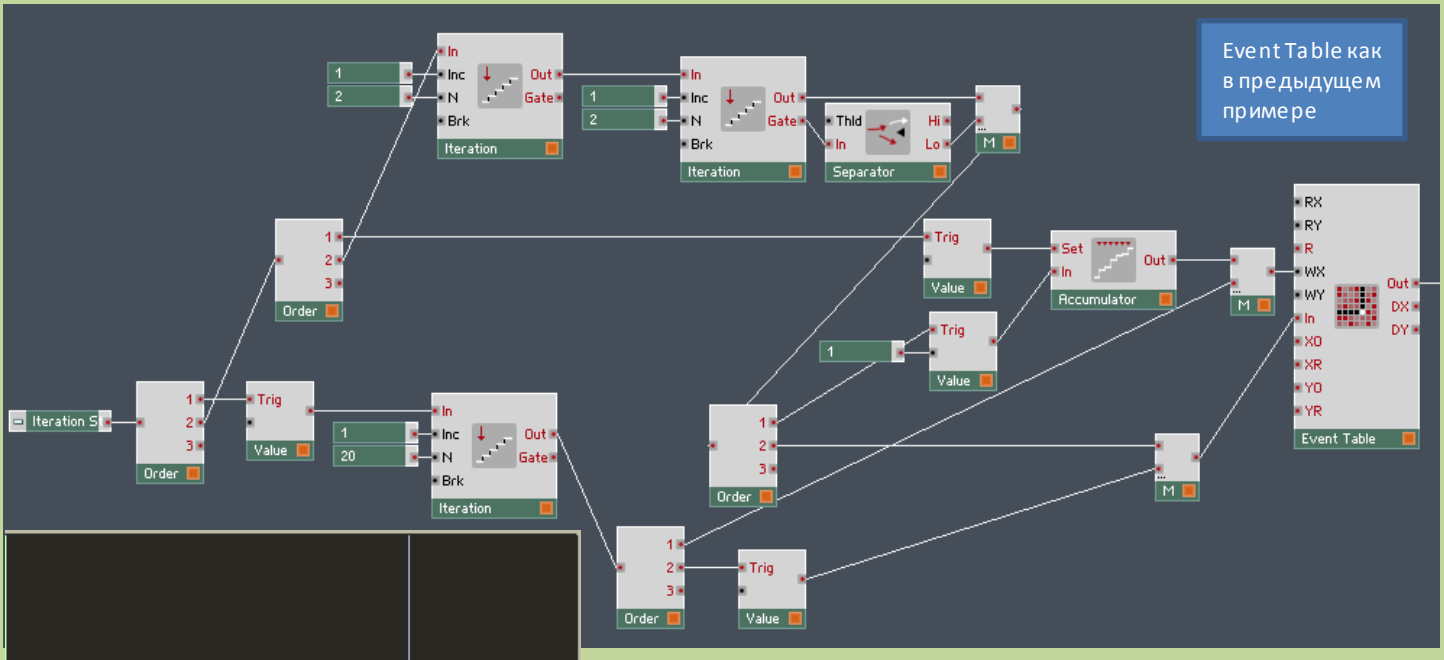
Эту схему можно выразить словами : три шага вперёд, один шаг назад, 1-стартовая точка.

Создаём рекурсию 1,2,3,4,2,3,4,5,3,4,5,6,4,5,6,7. Три шага вперёд, два шага назад. Схема с очисткой.



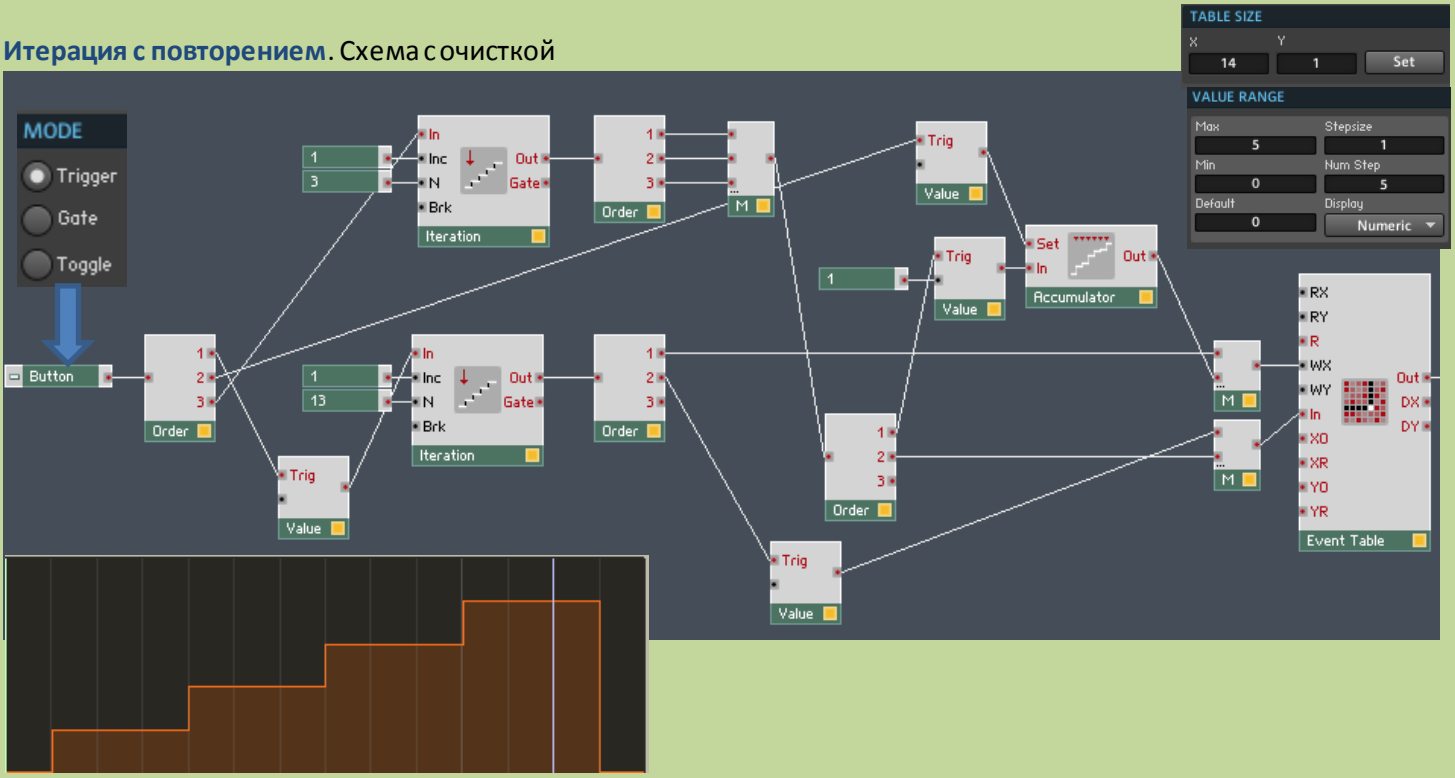


Рекурсия с вырезами 1,2,3,0,2,3,4,0,3,4,5,0. Схема с очисткой

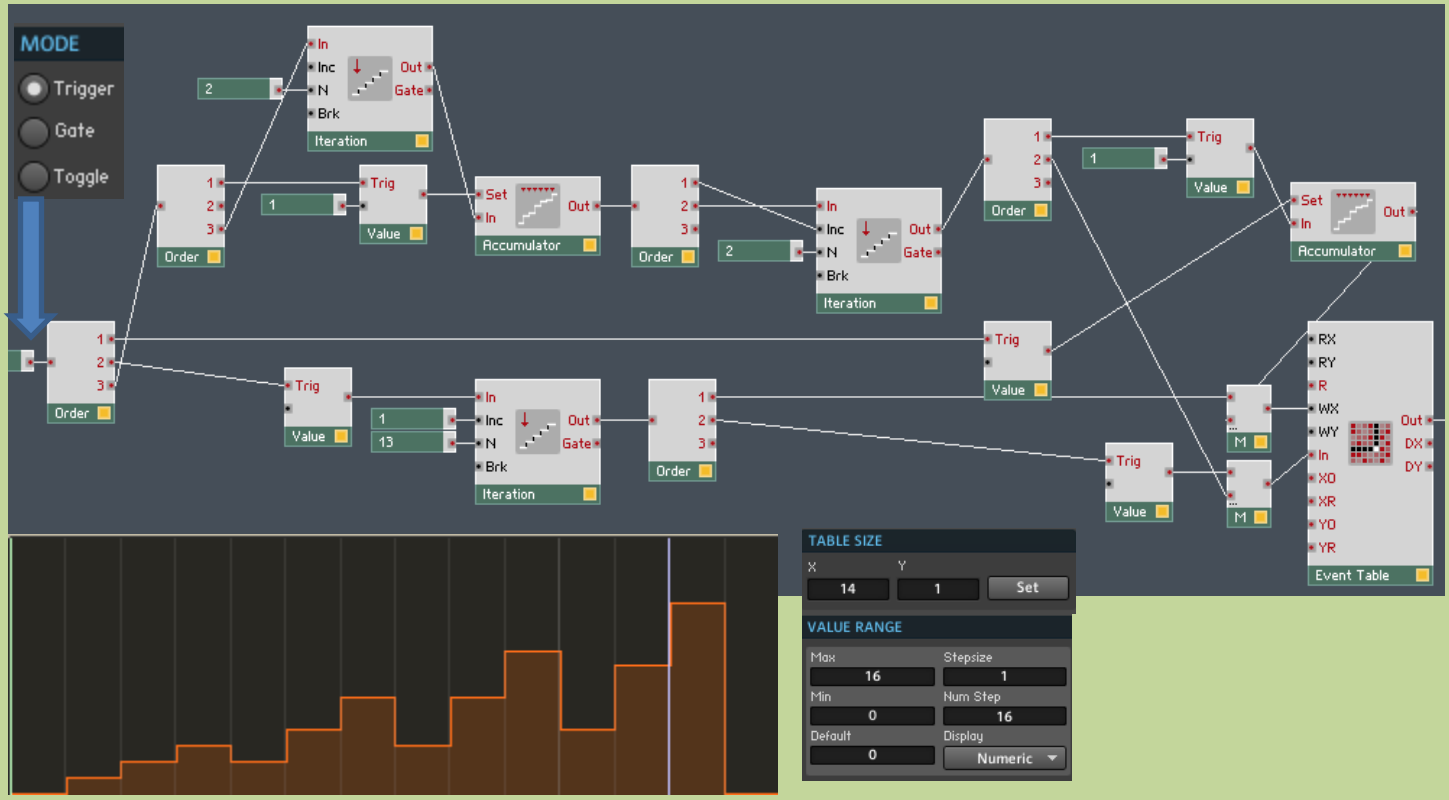


Здесь Separator не пропускает 1 с выхода Gate модуля Iteration

Итерация с повторением. Схема с очисткой



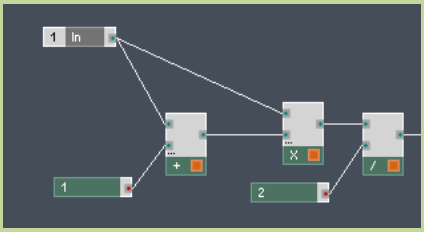
Рекурсия с наращиванием : **1**,2,3,**2**,4,6,**3**,6,9,**4**,8,12. Схема с очисткой



Не забывайте что на Numeric Readout вы увидите только самое последнее значение итерации. Так как изменение между значениями итераций = 2.5 миллисекундам и глаза не успеют увидеть эти изменения на Numeric Readout.

Суммирование всех чисел по заданное число, Knobs->N=4 -> 1+2+3+4=10

Такой последовательностью обладают треугольные числа

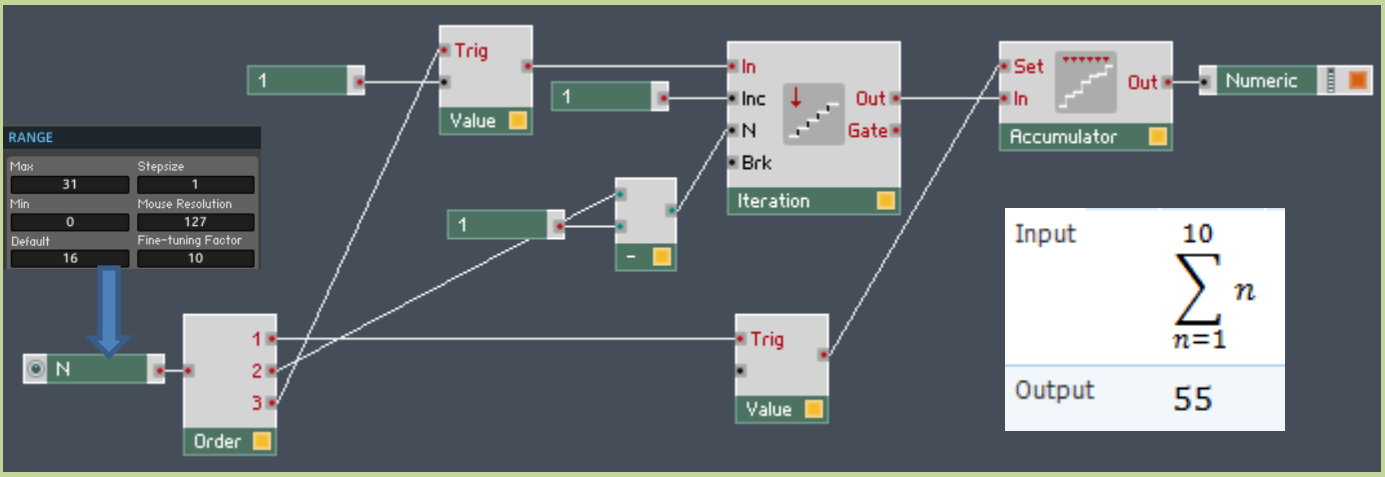


$n := 8$

Input $\frac{n\,(n+1)}{2}$

Output 36

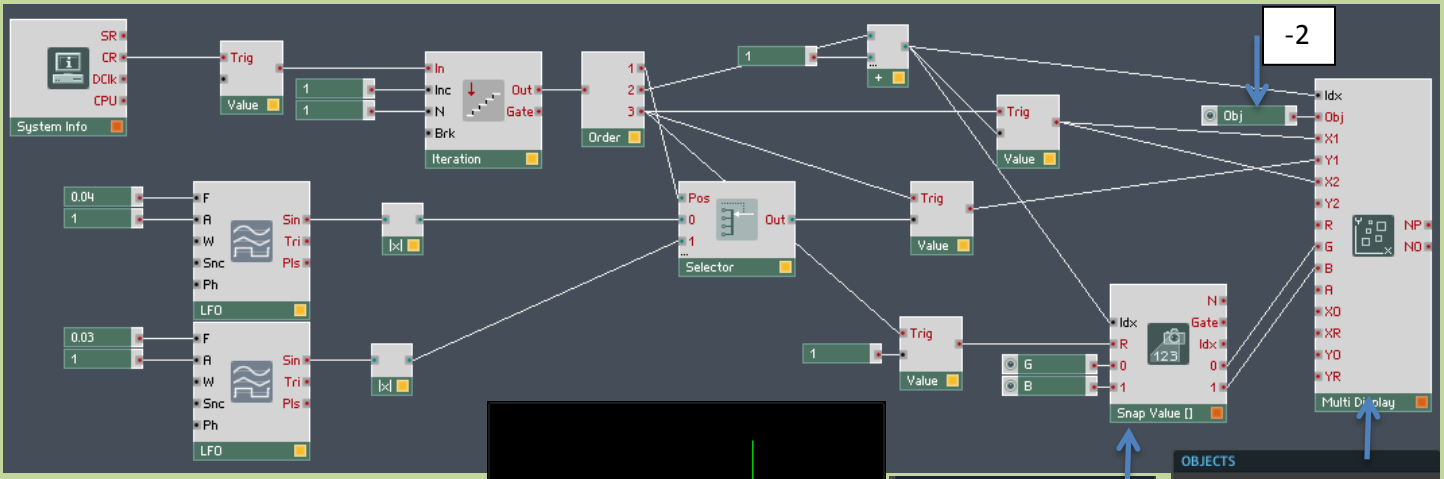
$$\sum_{n=1}^{N} n$$



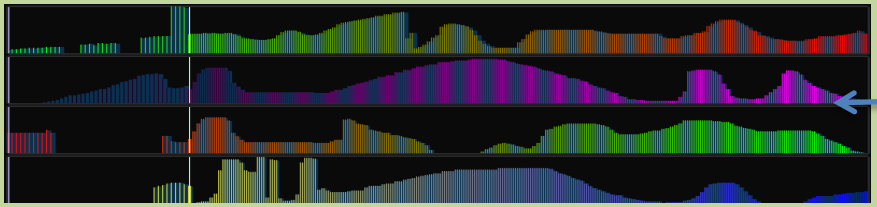
Как видно из схемы, когда мы крутим Knobs с целыми числами , допустим N=4 , то первым сигнал у Order идёт на Value с нулём, ноль приходит на Set модуля Accumulator и настраивает аккамулятор на ноль. Вторым сигнал с Order идёт сперва на вычитание единицы, а затем на вход N модуля Iteration – сколько будет итераций. Третим сигнал с Order идёт на Value с константой=1, так как для модуля Iteration самый последний сигнал должен прийти на вход In (Если входы Inc, N изменяются , то они должны измениться раньше чем вход In). Дальше Iteration выдаёт последовательность 1,2,3,4 при N=4, то аккамулятор суммирует эти значения и выдает на Out значение=10

Idx. Индексация значений

Для чего нужна индексация значений ? Допустим у вас два модуля LFO и нужно отобразить графически на одном табло (Multi Display) что они выдают. Некоторые Idx модули работают начиная с единицы (1 индекс, 2 индекс и т.д)



Модуль Snap Value [] служит для индексации цветов. Вставьте его и выберите в его опциях Append, нажмите Append 2 раза, в строке Label один назовите 0, а другой 1. Далее у Idx создайте крутилку-Knob, этой крутилкой выберите 1 и затем G(зелёный) поставьте в 0, а синий в 1. Потом крутилку-Idx поставьте на 2, а цвета наоборот, G=1, B=0. Тем самым первый индекс будет синим, а второй зелёным. Далее подключите выход где стоит +1 к входу Idx у Snap Value[].
Если у модуля Multi Display не убрать галочку с RGB то цвета будут одинаковыми.

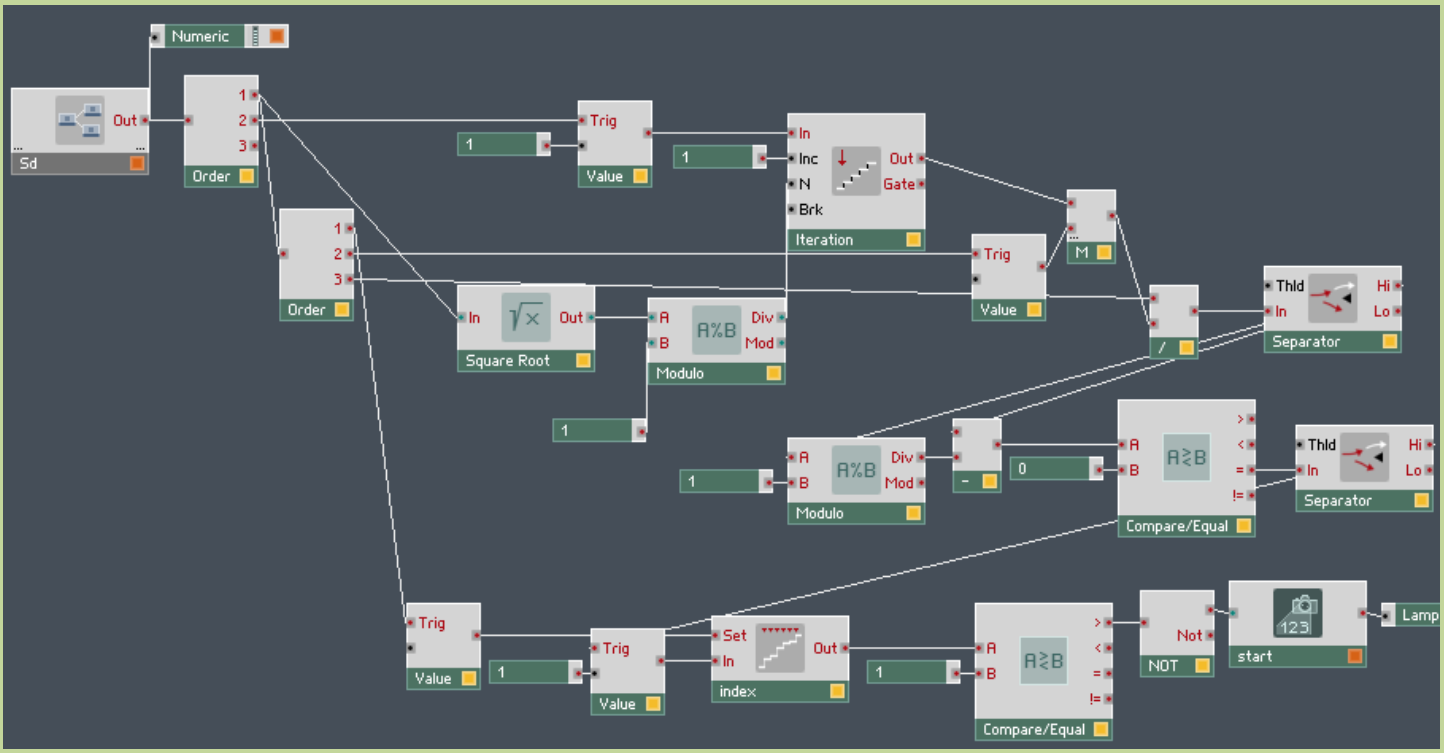


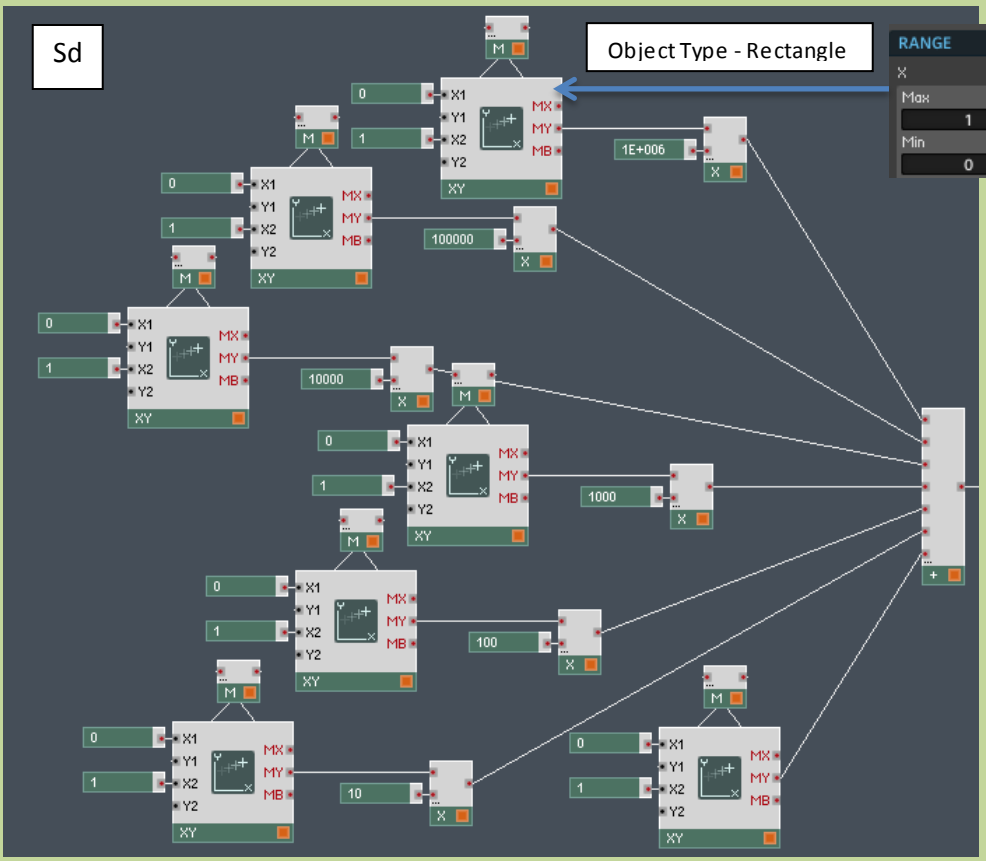
С помощью индексации можно создавать удобную визуализацию, например CC-контролер, он посылает миди сообщения на другие Knobs и управляет ими.

Простые числа до 9 миллионов

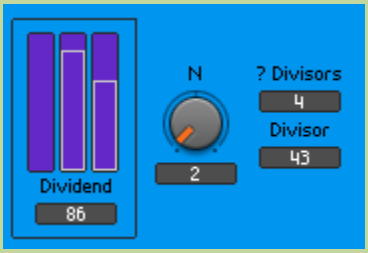
Здесь выбераем числа, справа налево разряды единиц, десятков, сотен и т.д

Если число простое то загарается лампочка



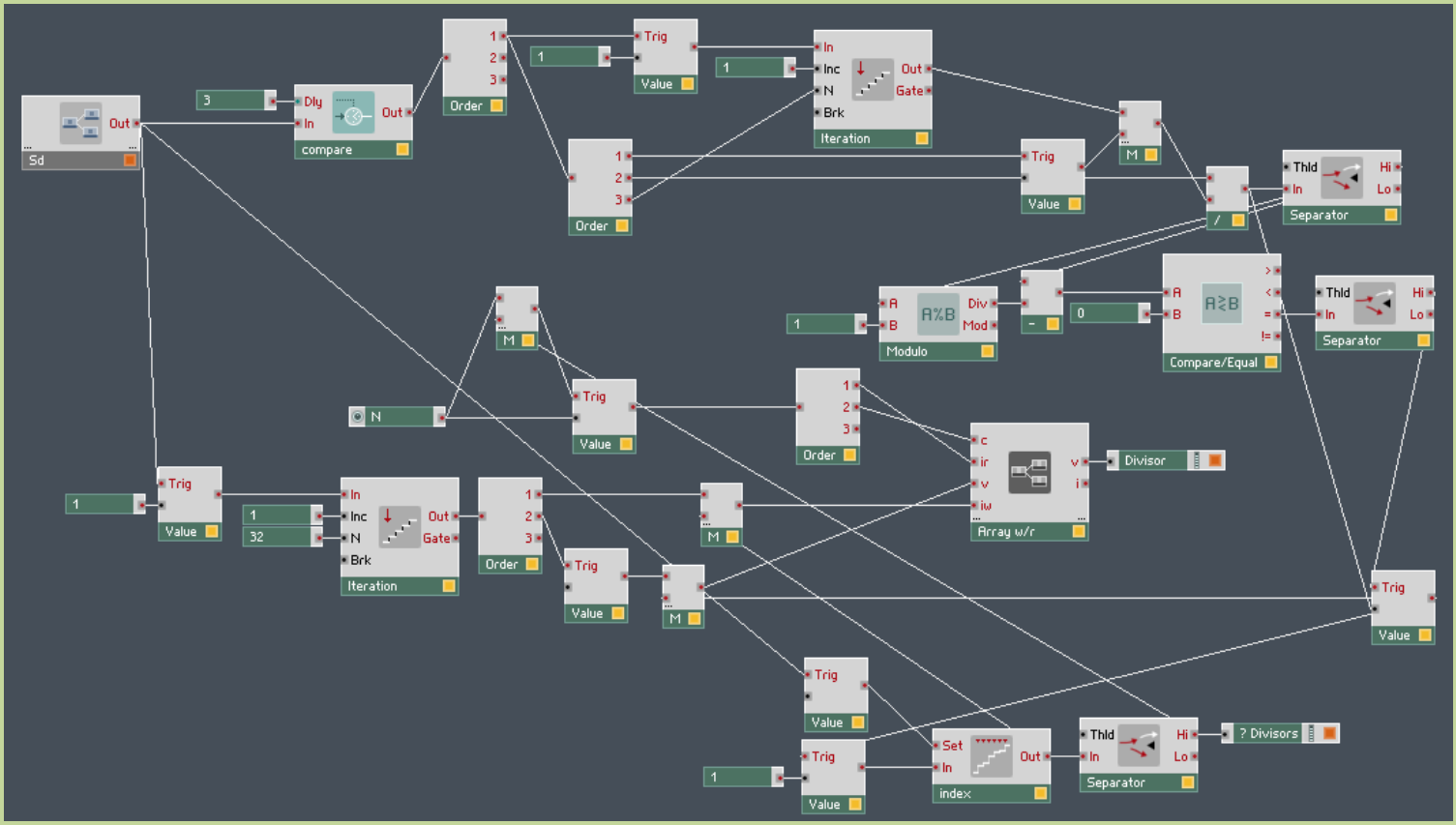


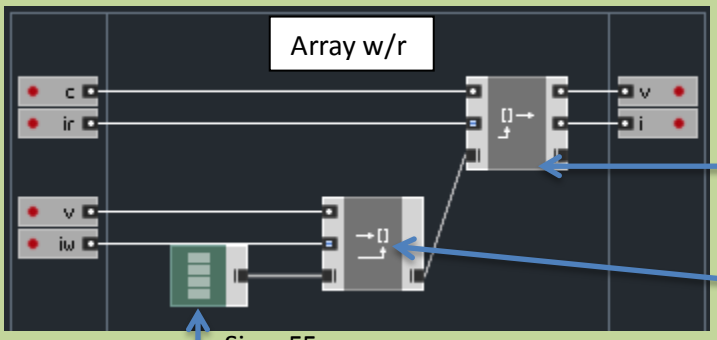
Делимое и делители

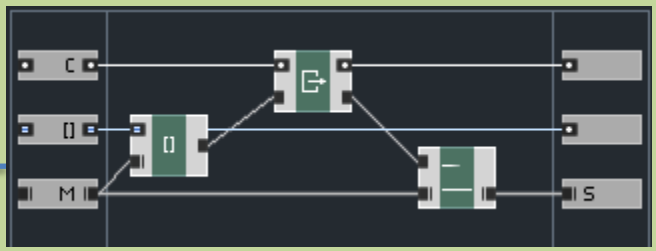


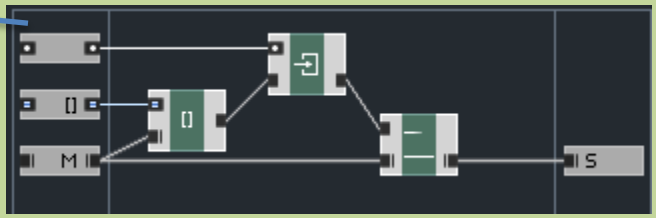
Dividend - делимое
?Divisors – показывает сколько делителей у делимого
N – выбор делителя
Divisor – значение делителя

Пример: выбрали число 86, ?Divisors показывает что у числа 86 четыре делителя, с помощью N выбераем делитель, второй делитель равен 43.

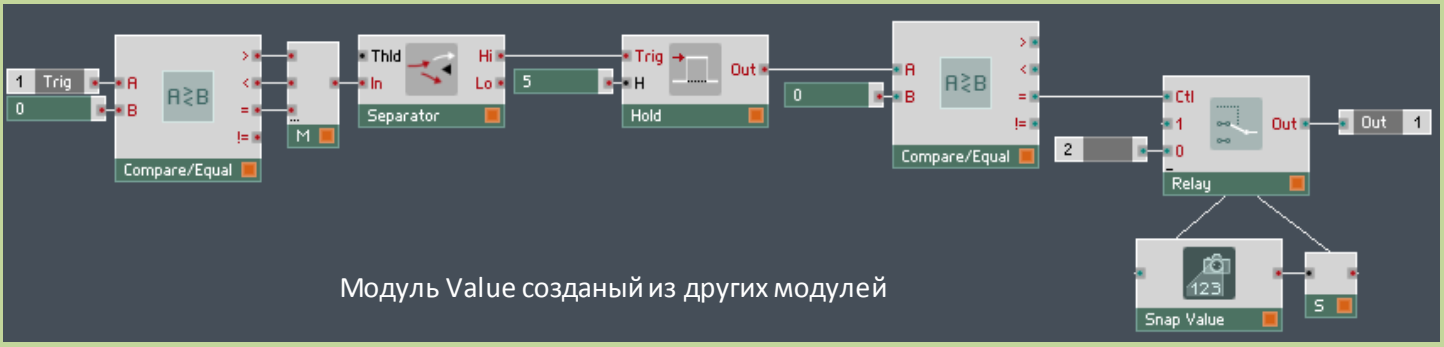






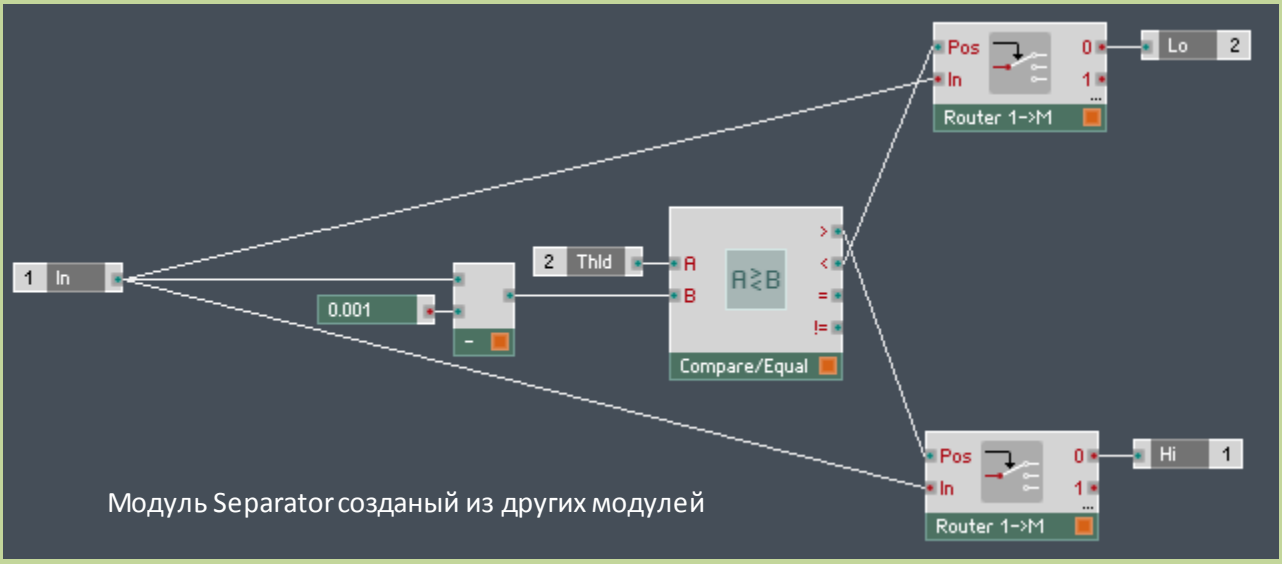


Модуль Sd как в предыдущем примере
Если ?Divisors равен 2, то делимое это простое число.



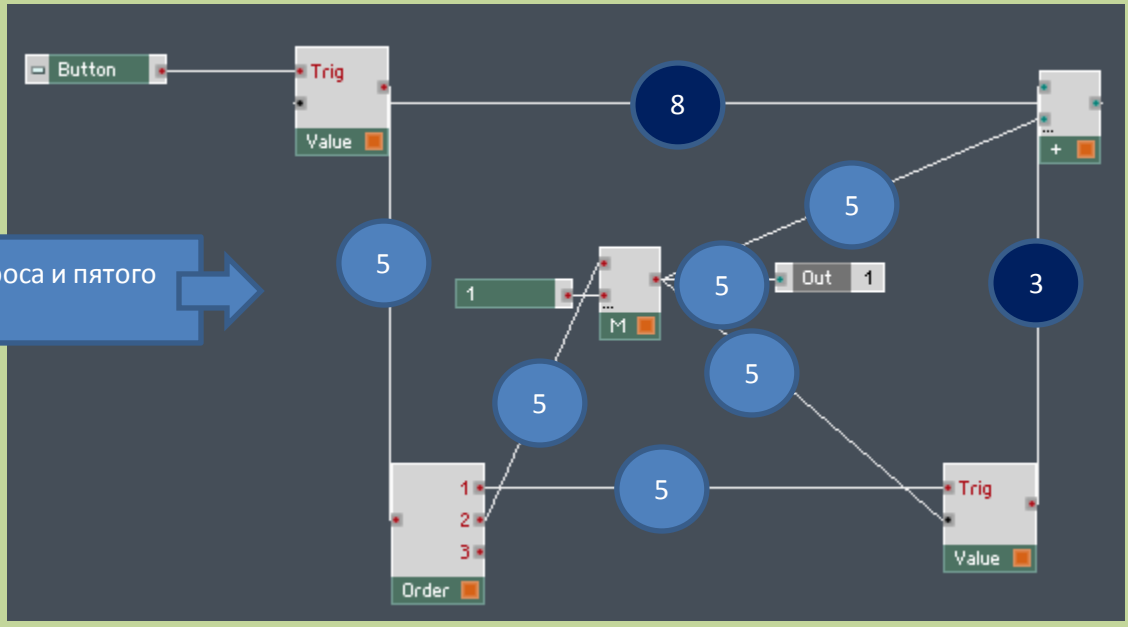
Separator

Сепаратор – это решето, которое пропускает или не пропускает входящие значения согласно уставке на входе Thld(Threshold). Если на входе Thld ничего нет, то значение входа Thld принемается за ноль, тогда выход с Hi будет от 1 и выше, а с Lo до 0 и ниже.



При такой схеме идёт пропускание от 3 до 10

Числа Фибоначи



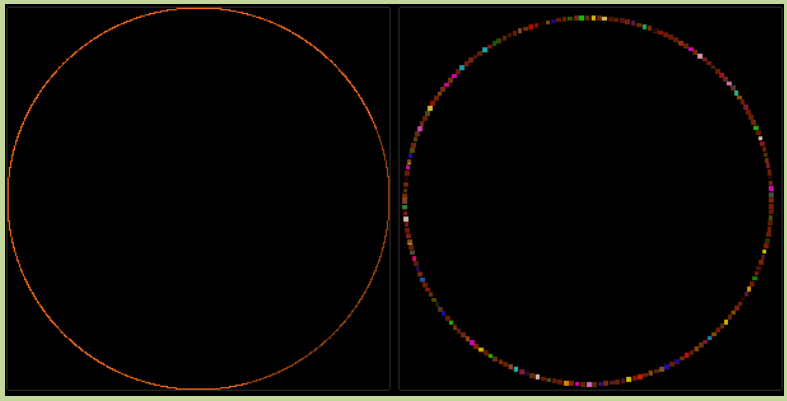
Значения после сброса и пятого нажатия Button

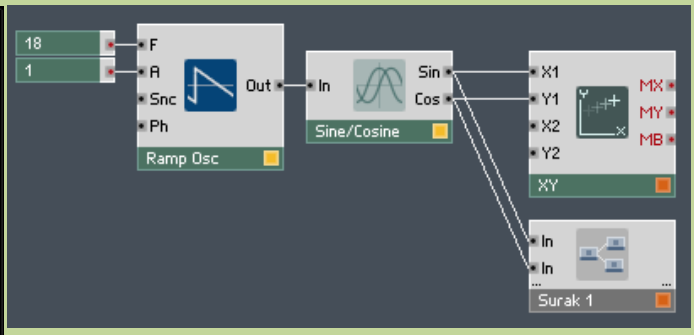
Преобразование декартовых координат модуля XY (режим Scope) в Multi Display для кривых линий.

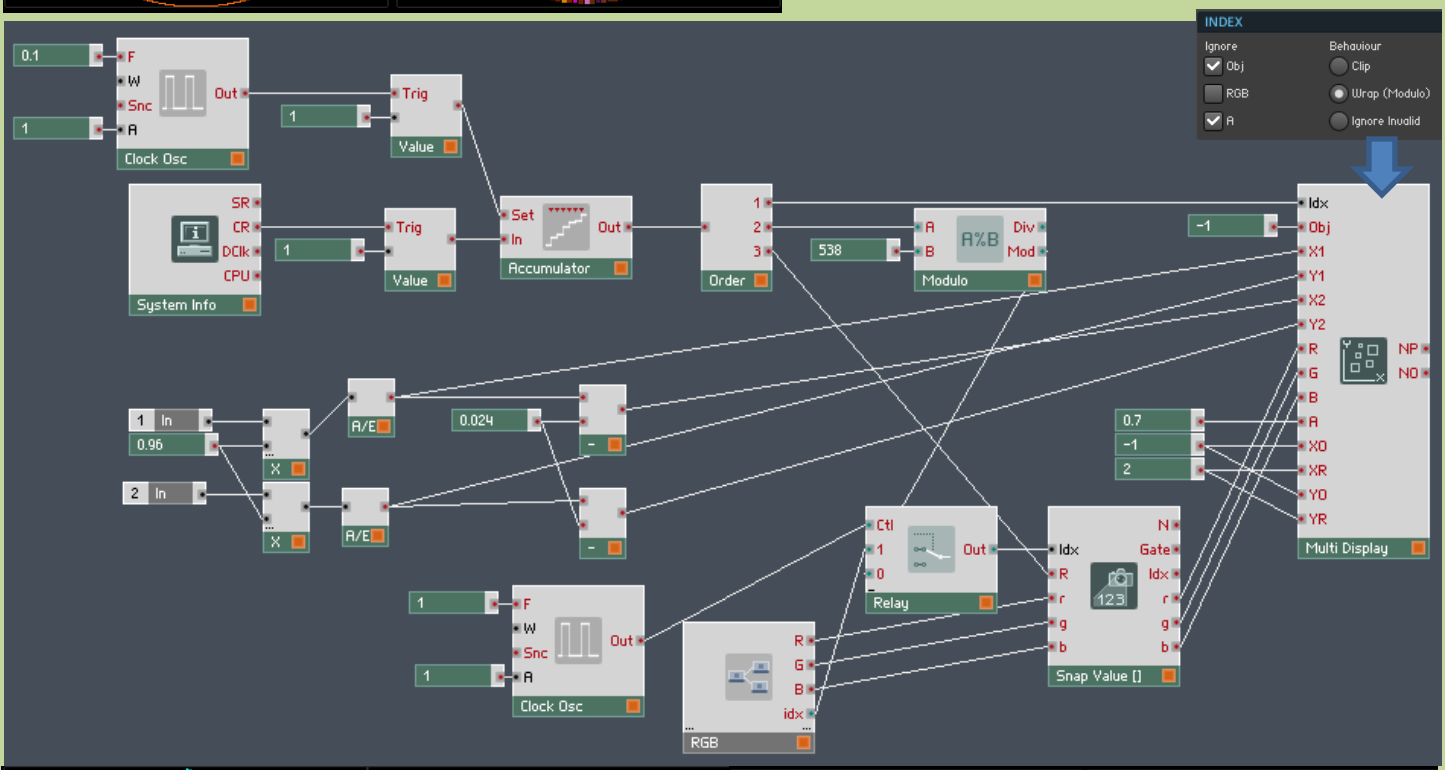
Так как настройка цвета для линий в Реакторе одна и та же то если использовать несколько модулей XY с режимом Scope то у всех модулей будет один и тот же цвет.

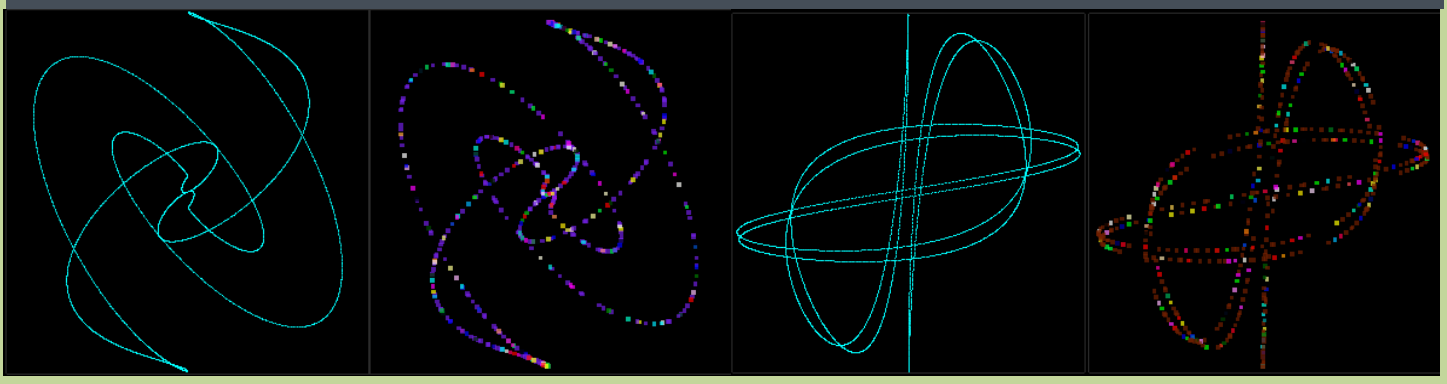
Item	R	G	B
Panel	193	186	165
Indicator	255	110	25
Graph Line	255	110	25
Graph Fill	83	52	27
Graph BG	40	39	34

Чтобы внести разнообразие можно воспользоваться таким преобразованием – кривые линии станут разноцветными квадратами.





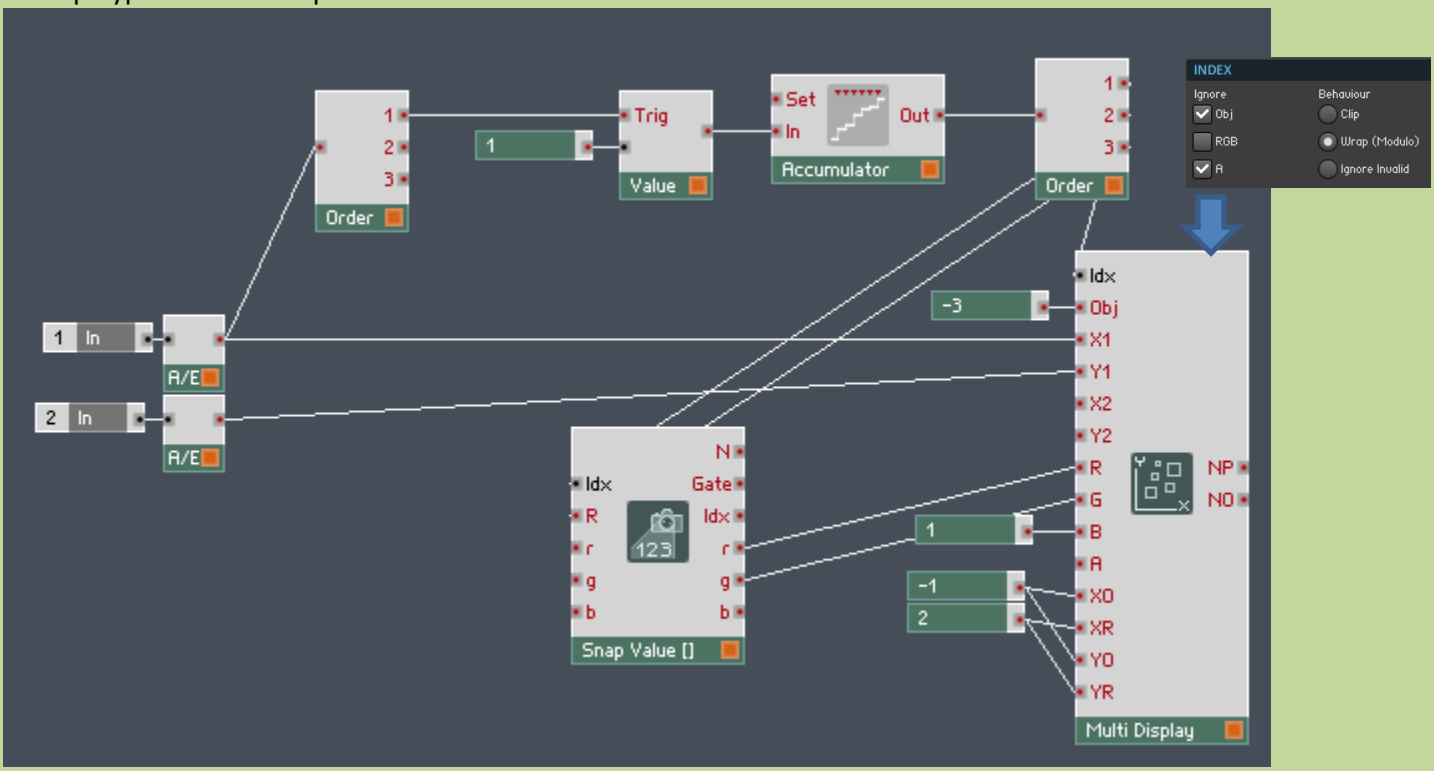




XY Multi XY Multi

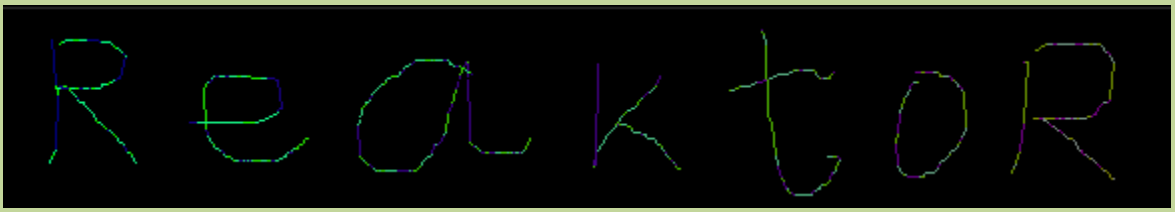
Преобразование декартовых координат модуля XY (режим Scope) в Multi Display для прямых линий.

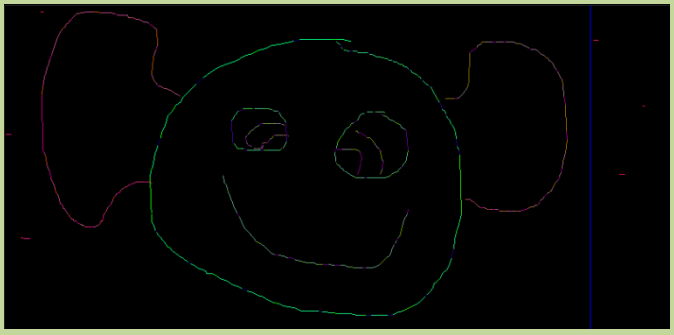
Если фигура состоит из прямых линий то можно воспользоваться такой схемой :



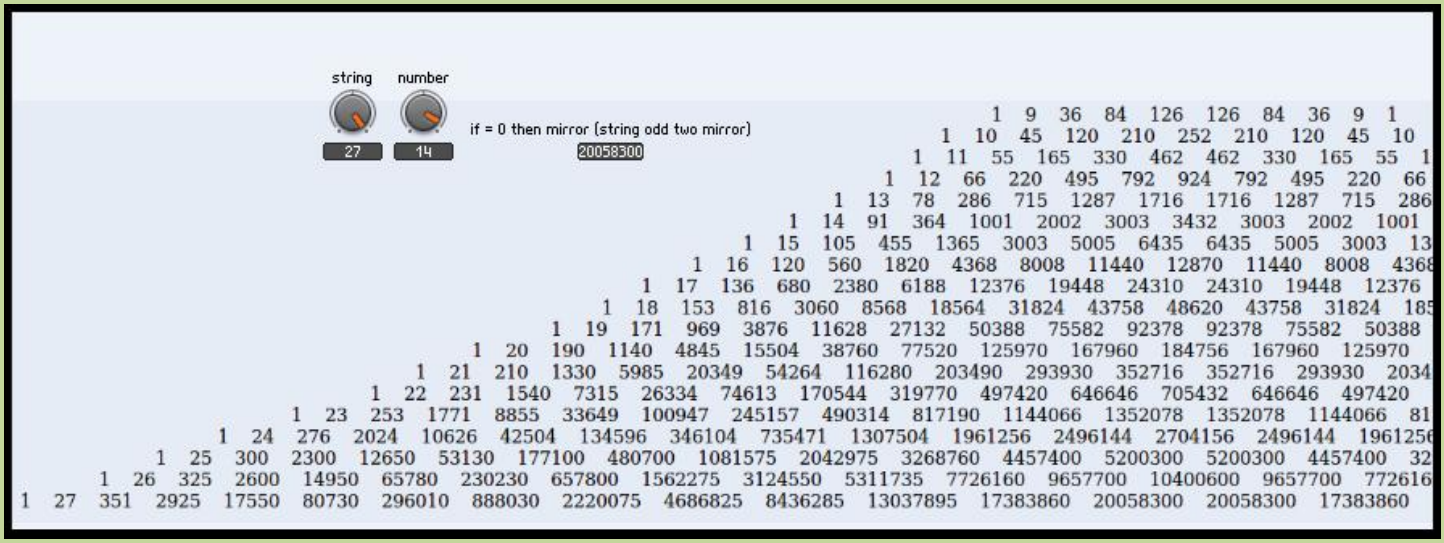
фигуры фиолетового цвета это XY модуль
Фигуры красного цвета это Multi Display

С помощью Mouse Area и Multi Dispay можно создавать текст или рисовать, также путём считывания можно это озвучить – будет чем-то похоже на систему UPIC.

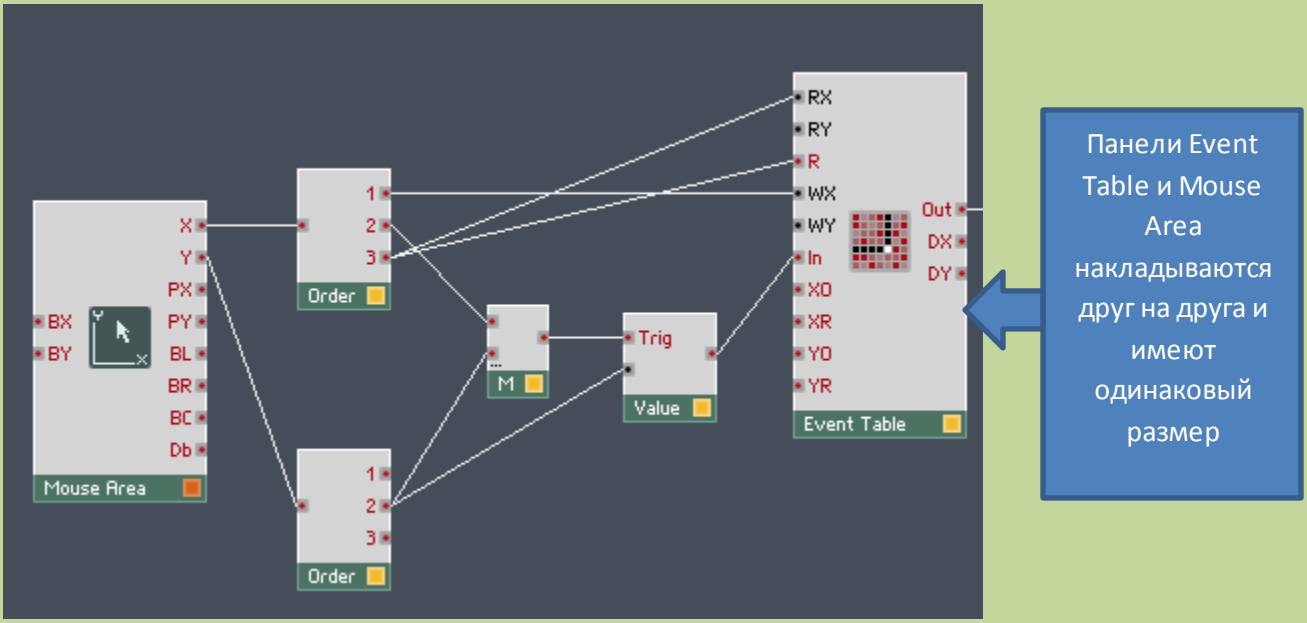




С помощью итерации можно получать разные математические модули.
Пример : 27-ая строчка треугольника Паскаля, 14 число – 20 миллионов 58 тысяч 300 триста

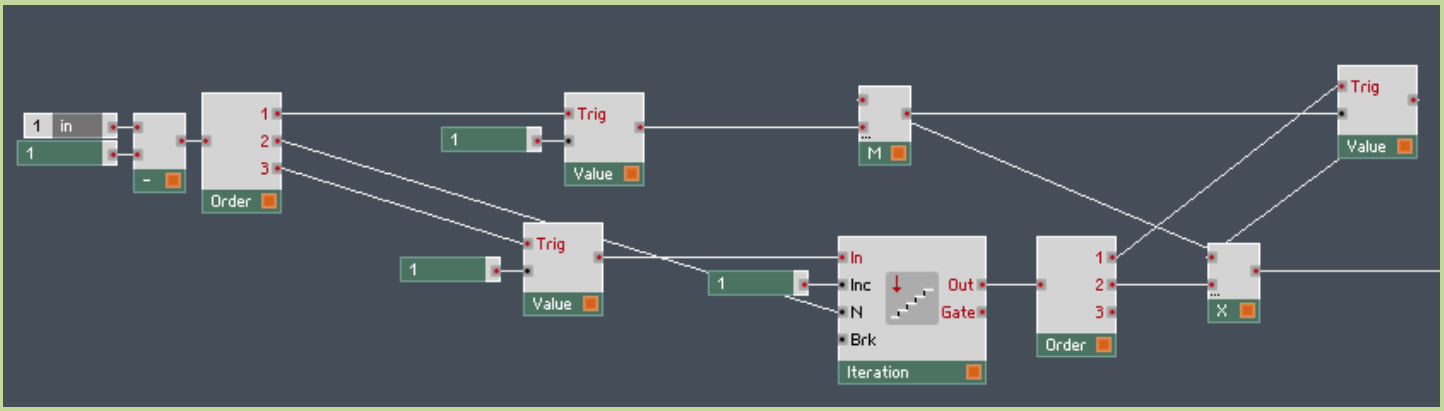


В Event Table если нажать правой клавишей мыши то можно выбрать режим Table Draw Mode, в этом режиме можно рисовать Y значения в каждой X ячейки. Это также можно сделать с помощью Mouse Area

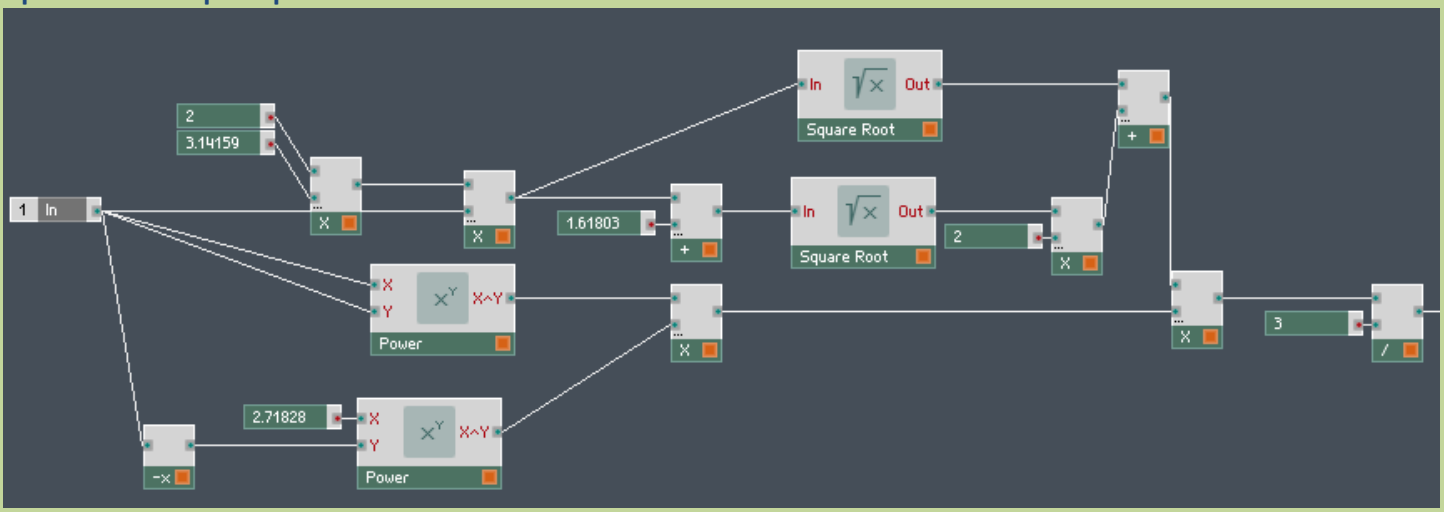


Панели Event Table и Mouse Area накладываются друг на друга и имеют одинаковый размер

Факториал n!

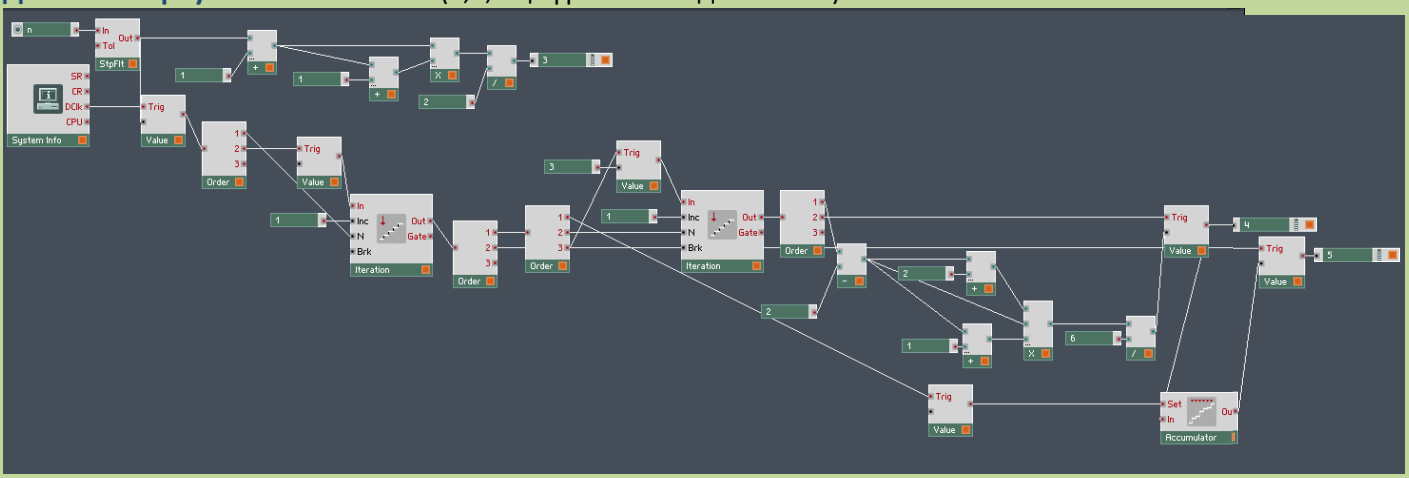


Приближёный факториал ~n!

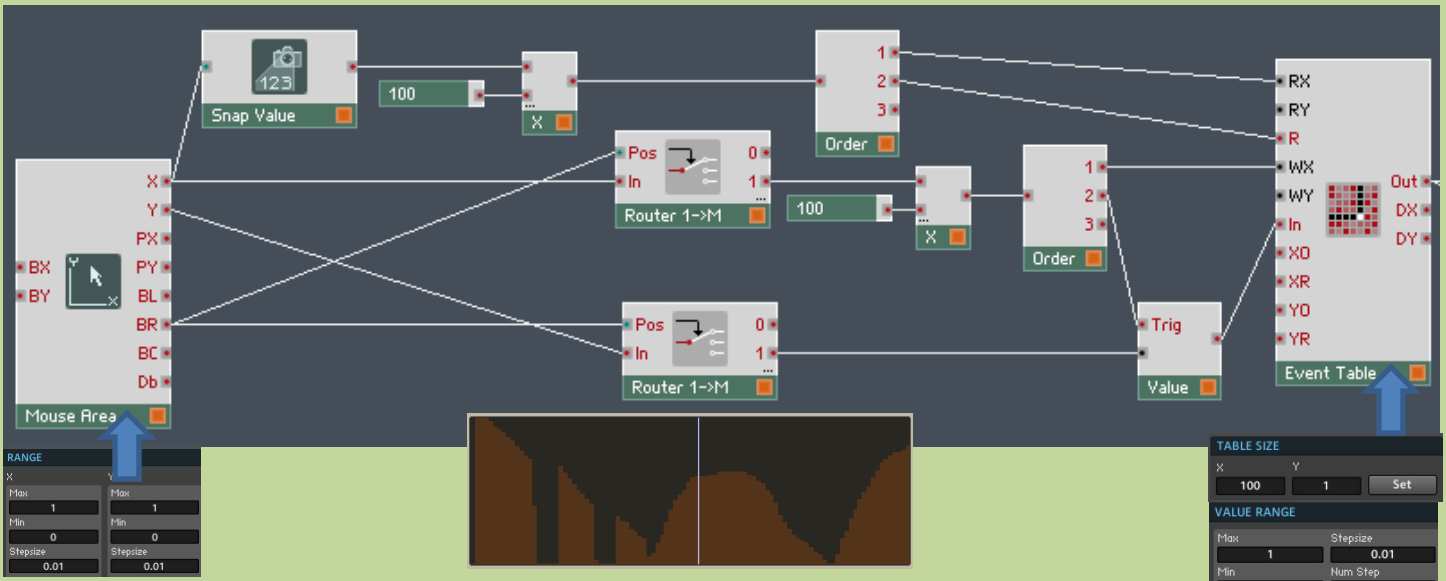


$$\sim n! = \frac{\left(\sqrt{n2\pi} + \sqrt{n2\pi + \varphi}\,2\right)\left(n^n\left(e^{-n}\right)\right)}{3}$$

Диагонали треугольника Паскаля (3,4,5 цифры любой диагонали)

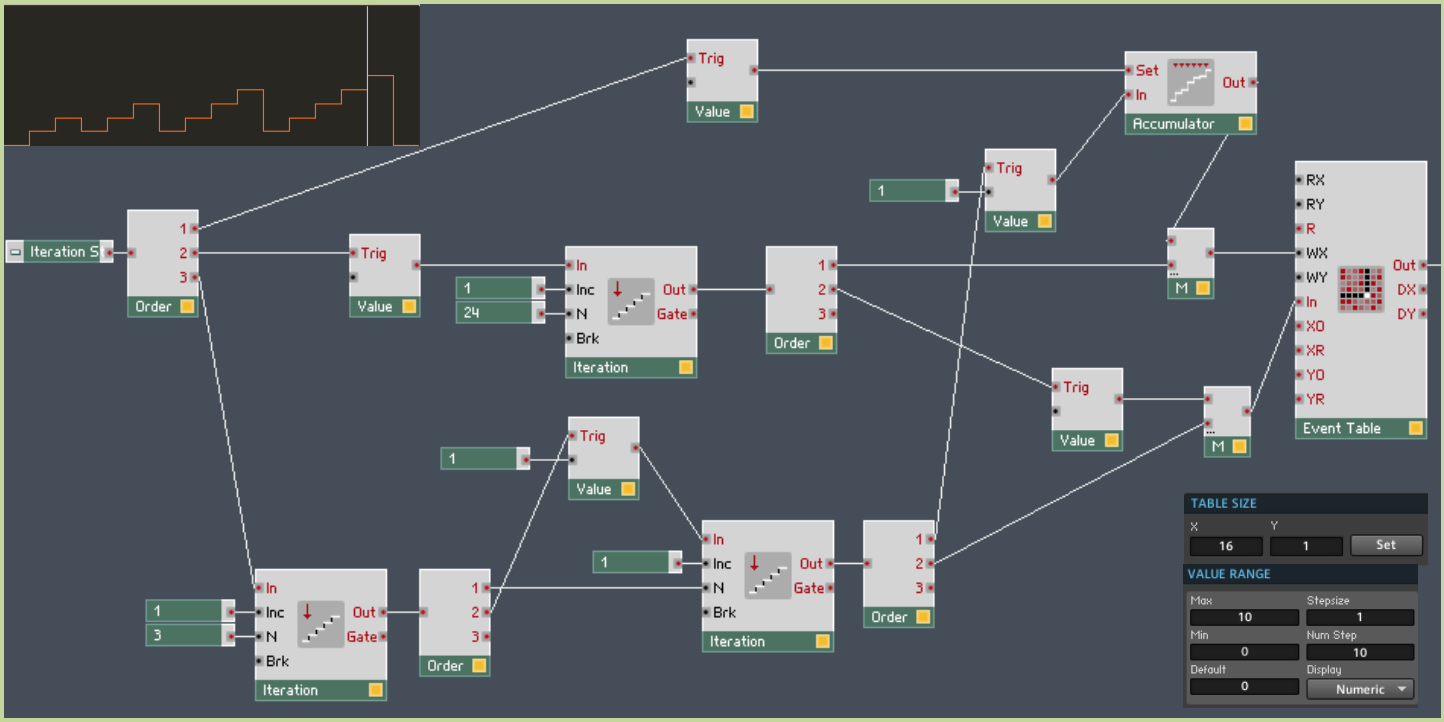


Рисуем и считываем значения в качестве Knob



Левая клавиша мыши считывает, правая записывает. Выход можно подать на вход умножения с другим модулем и управлять громкостью или на IC Send и управлять чем-либо согласно нарисовному графику. Панели Event Table и Mouse Area накладываются друг на друга и имеют одинаковый размер.

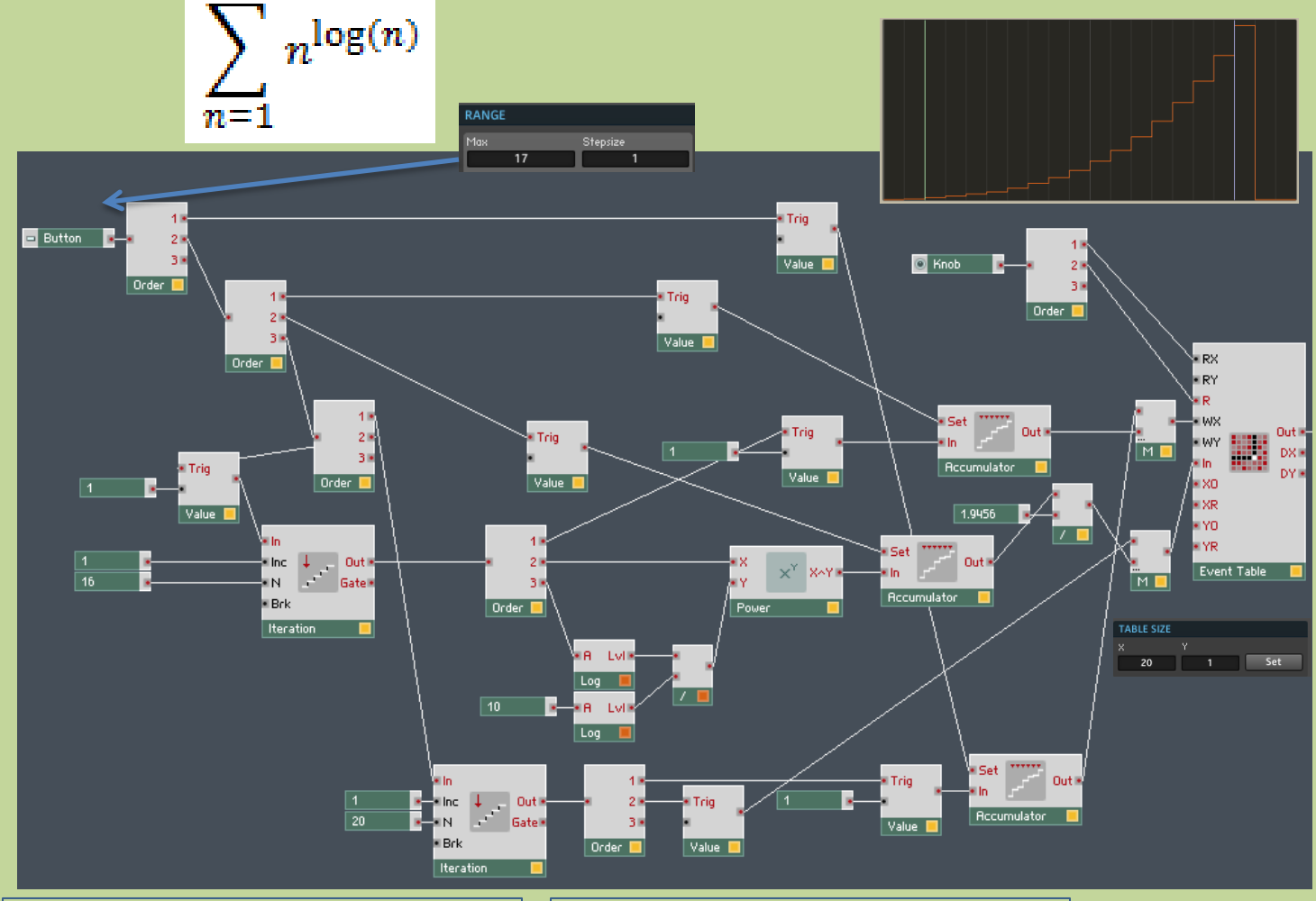
Создаём рекурсию : 1,2,1,2,3,1,2,3,4,1,2,3,4,5



Преобразование формулы в схему

Дана формула

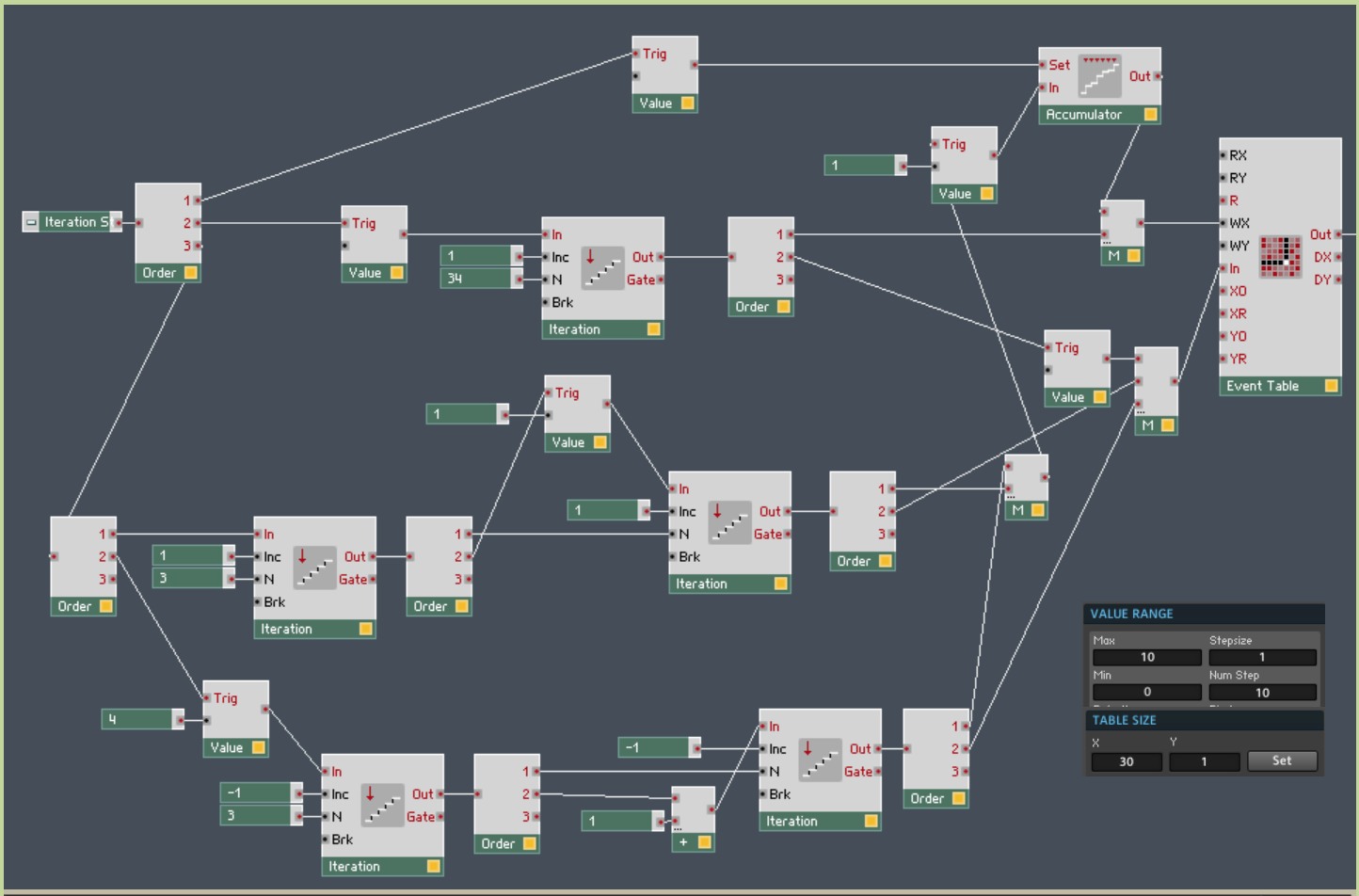
$$\sum_{n=1}^{17} n^{\log(n)}$$

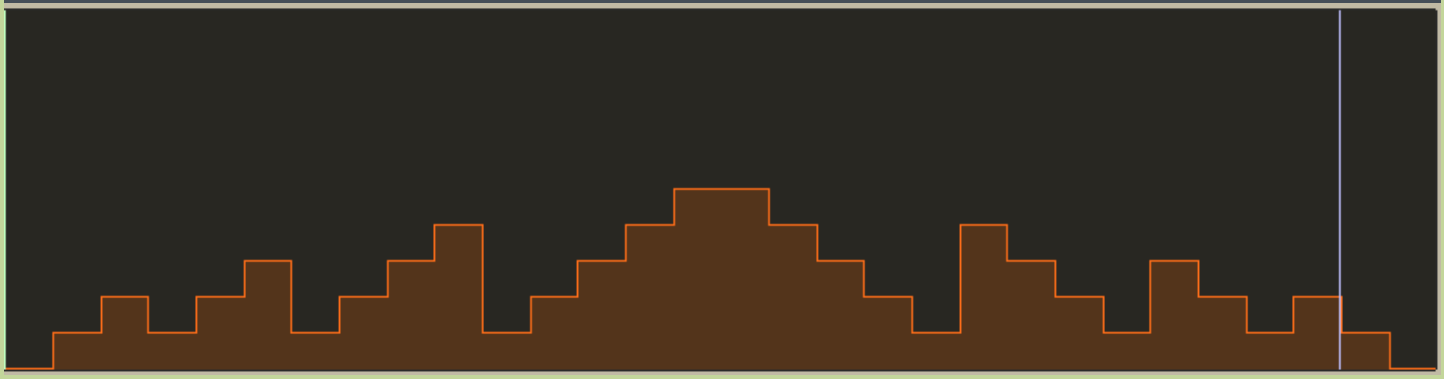


$$\sum_{n=1}^{2} n^{\log(n)} = 2.232023688689$$ Numeric 2.1752

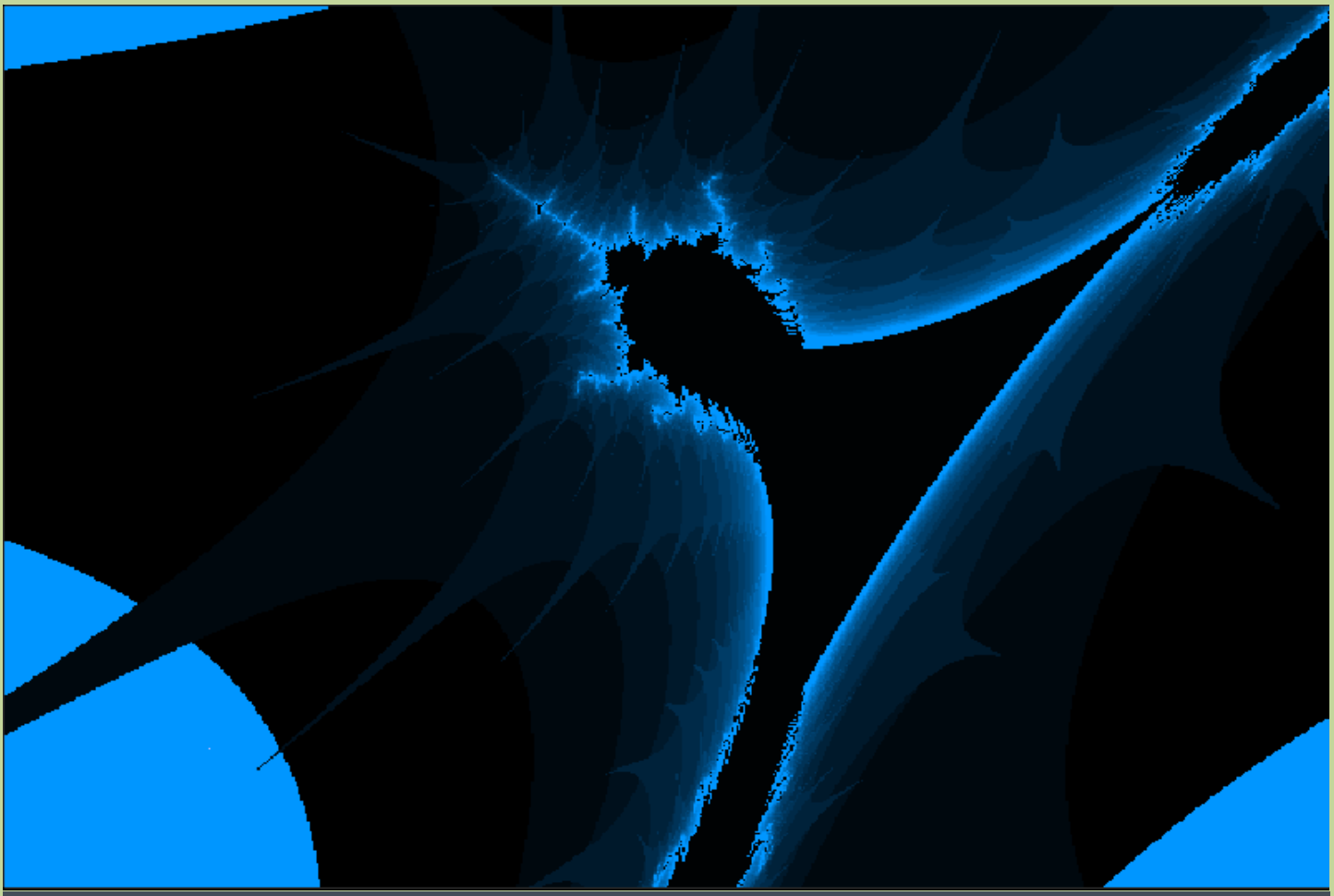
$$\sum_{n=1}^{17} n^{\log(n)} = 192.954338512415$$ Numeric 189.5752

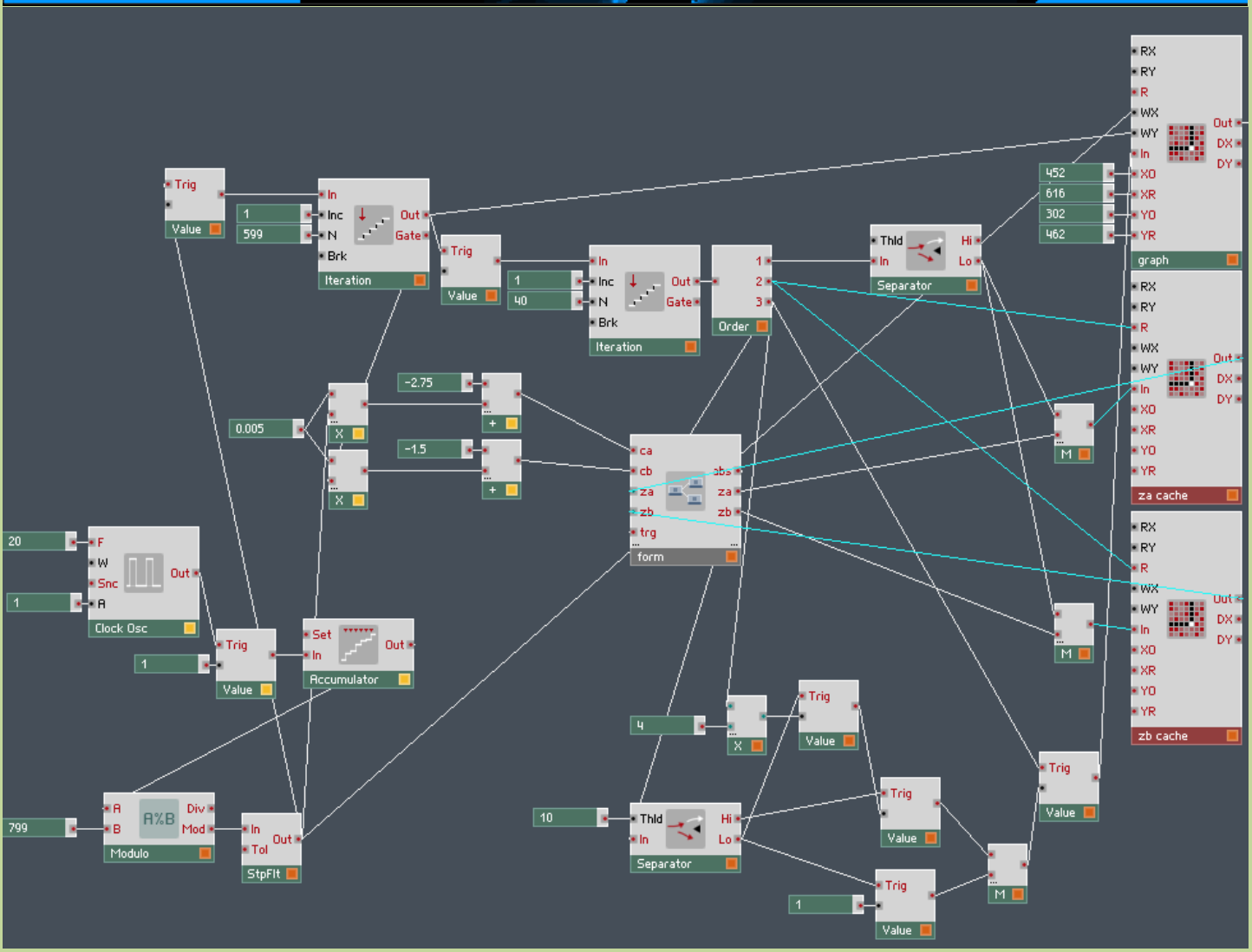
Создаём пирамидальную рекурсию 1,2,1,2,3,1,2,3,4,1,2,3,4,5,5,4,3,2,1,4,3,2,1,3,2,1,2,1





Создаём фрактал





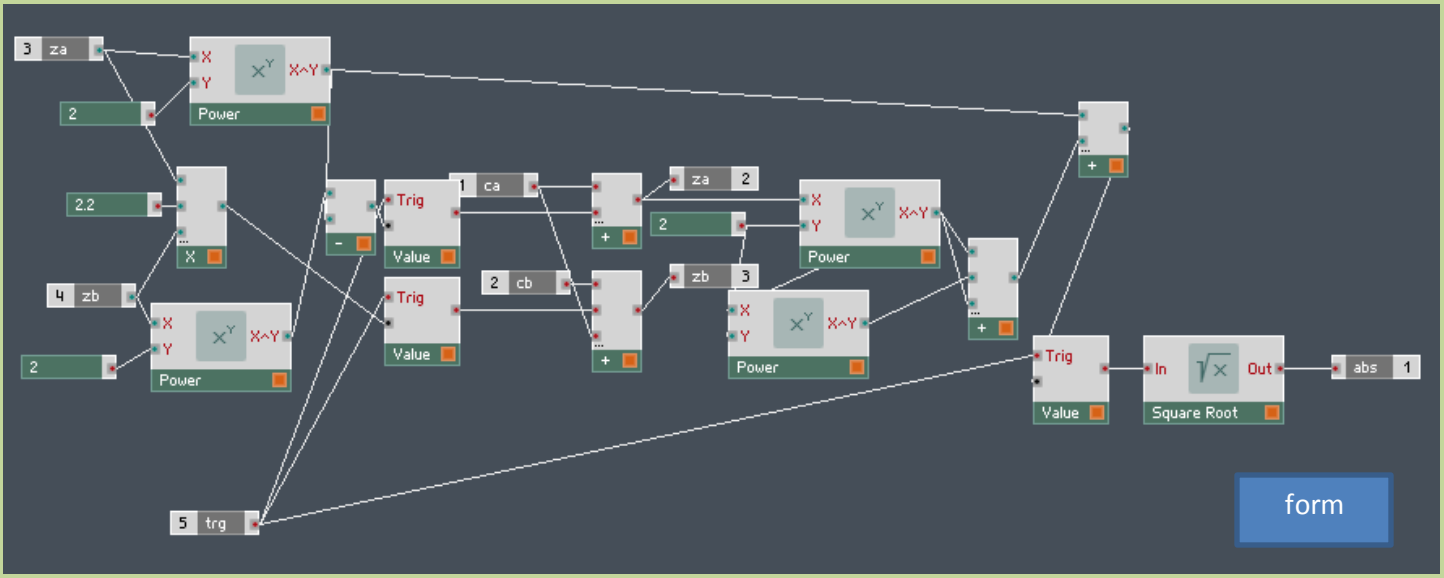


TABLE SIZE

X	Y	
800	600	Set

VALUE RANGE

Max	Stepsize
70	0
Min	Num Step
0	0
Default	Display
0	Numeric

MODE

Interpolation	Clip / Wrap XY
None	Wrap

Graph: 2D Color

graph

TABLE SIZE

X	Y	
1	1	Set

VALUE RANGE

Max	Stepsize
50	0
Min	Num Step
-50	0
Default	Display
0	Numeric

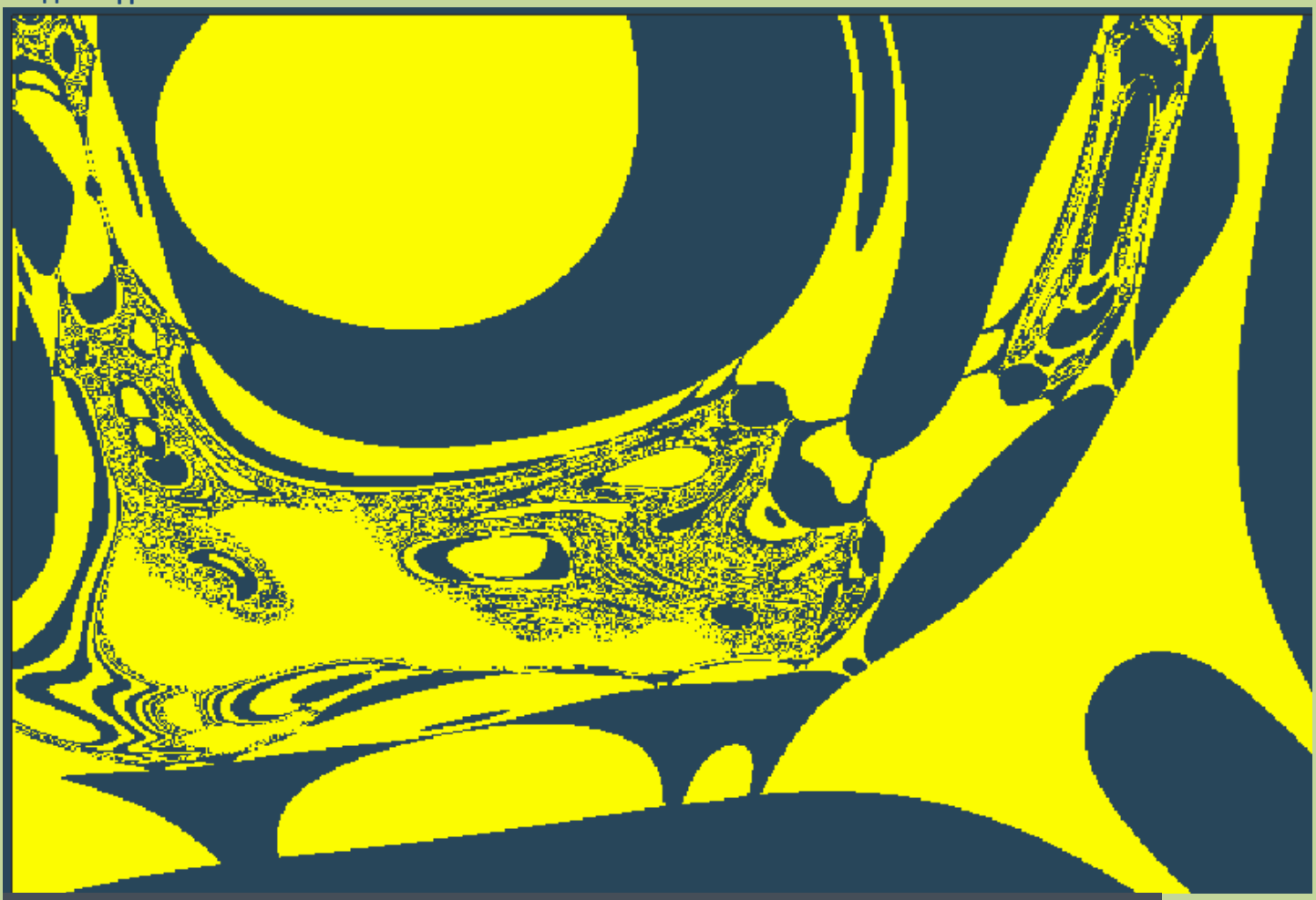
MODE

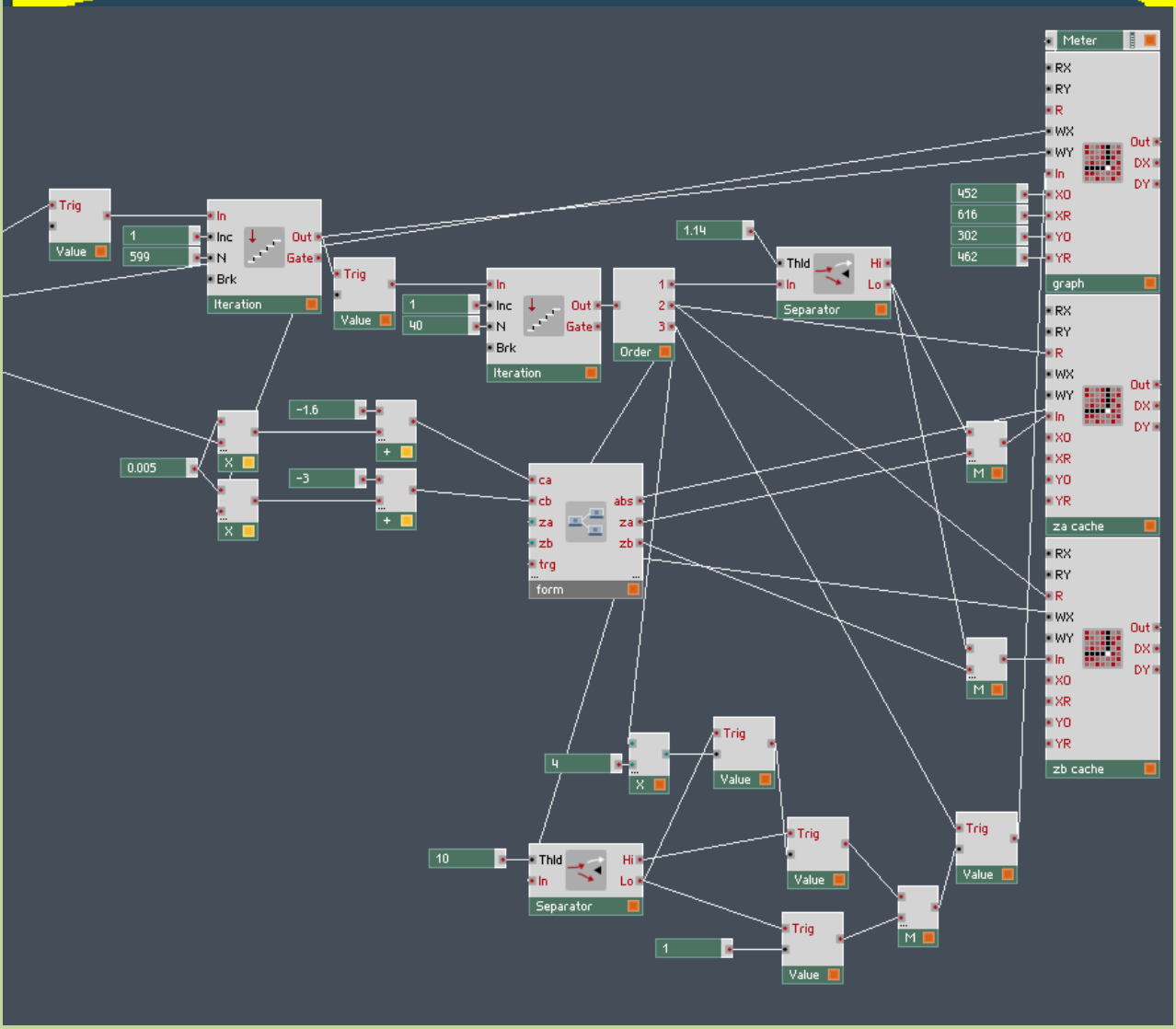
Interpolation	Clip / Wrap XY
None	Clip

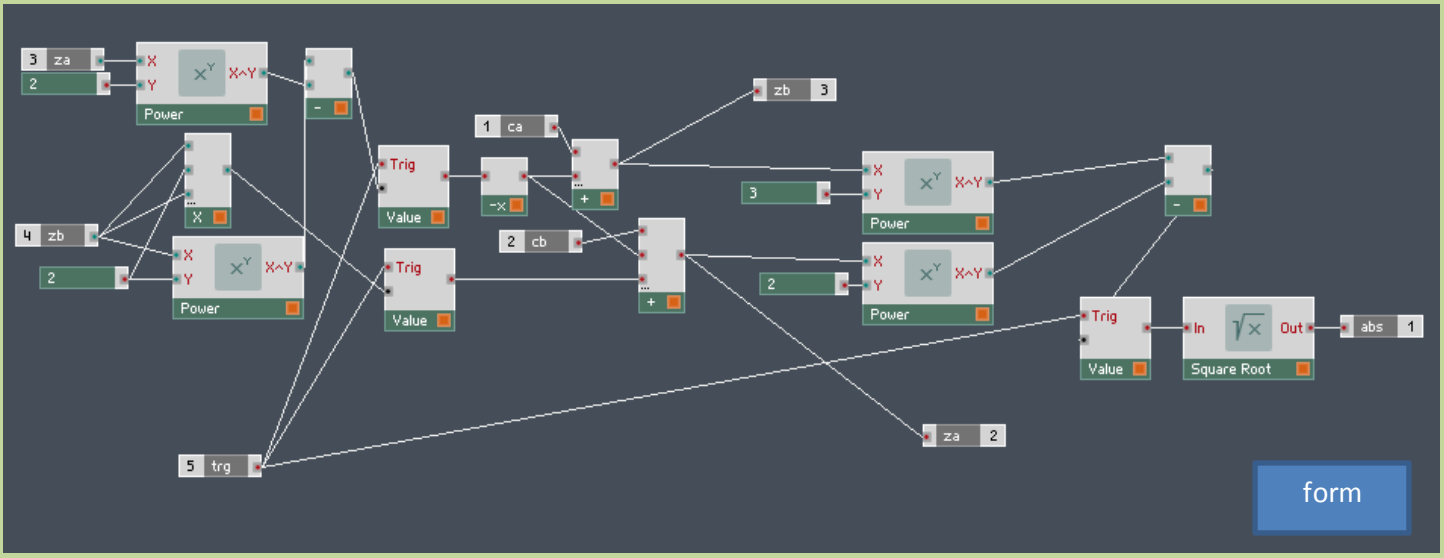
Graph: Line

za,zb cache

Создаём фрактал

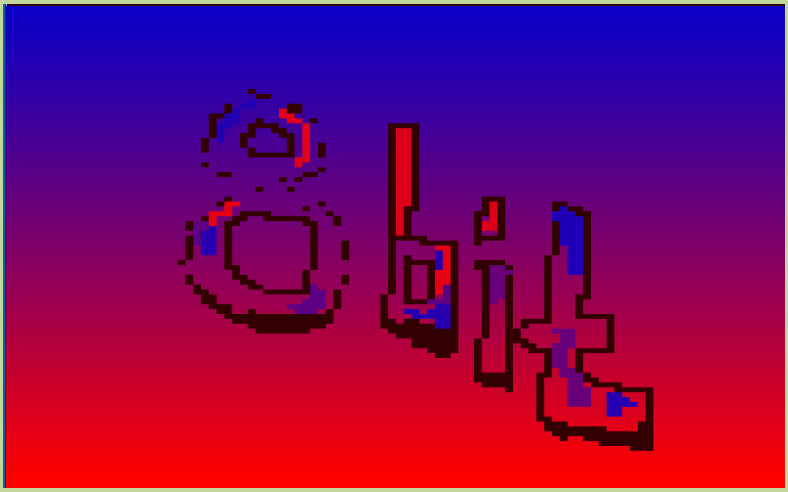


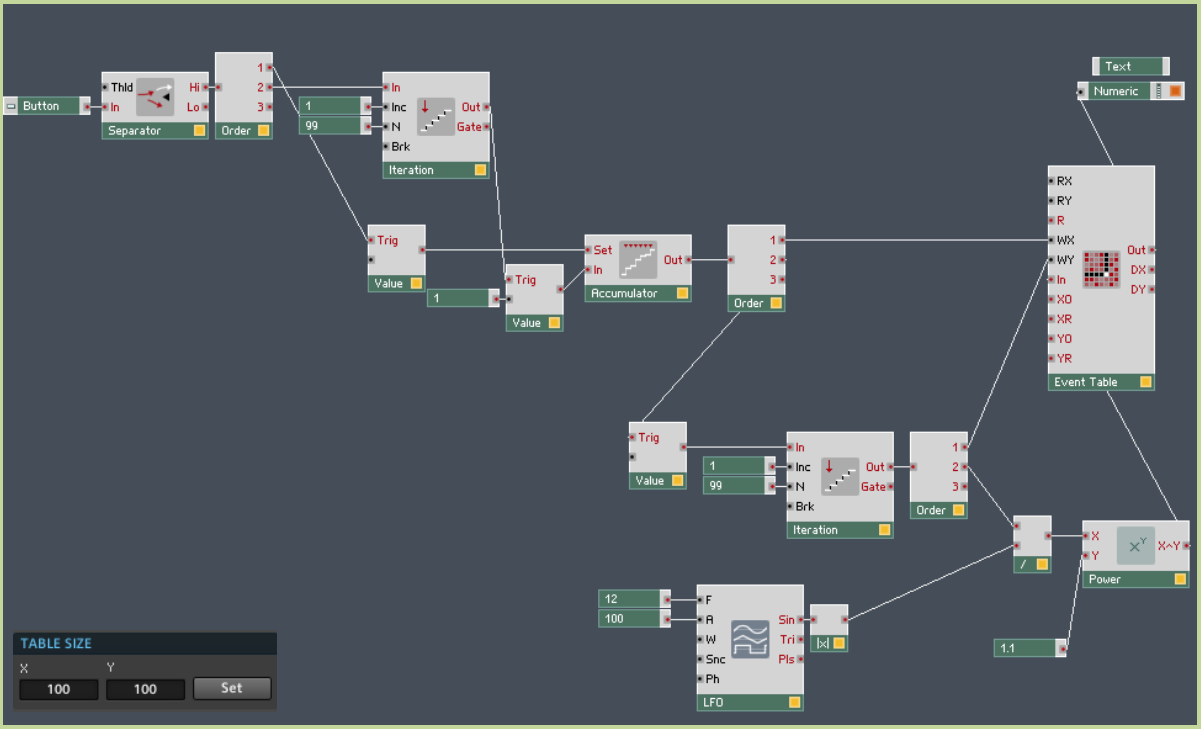




Остальные настройки как в предыдущем примере.

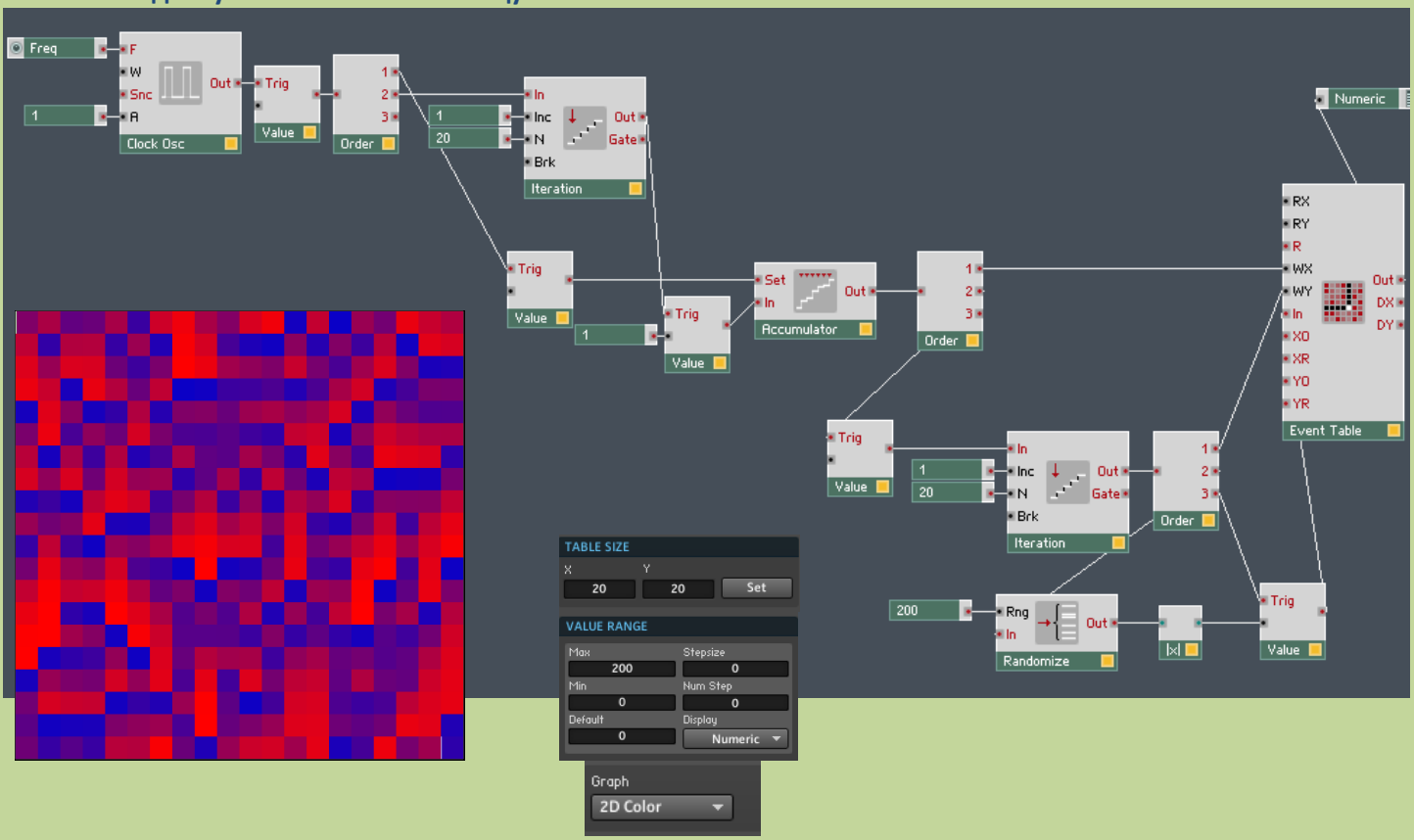
Создаём панель для рисования





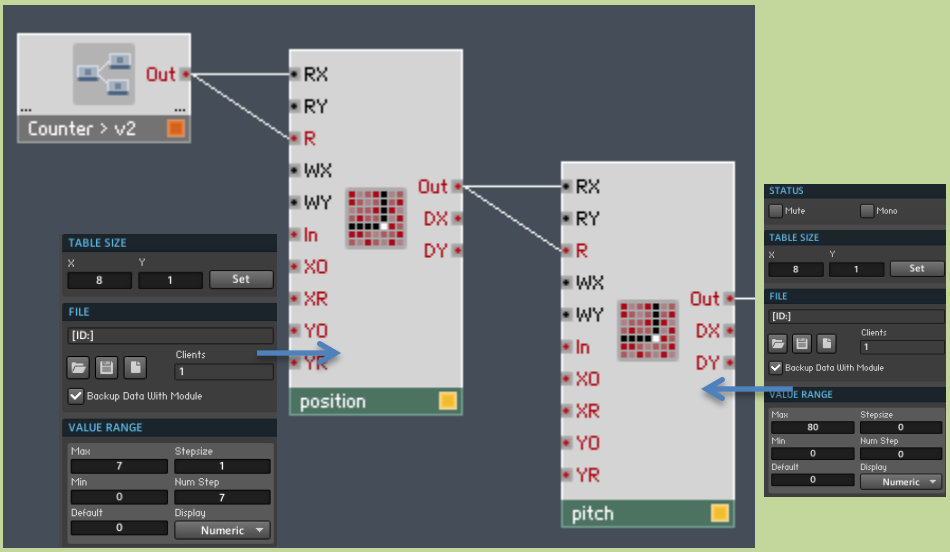
Для рисования нужно нажать правой клавишей мыши по Event Table и выбрать Table Draw Mode. Чтобы поменять цвет нужно удерживать на клавиатуре клавишу Ctrl и нажать левой клавишей мыши на панели рисования.

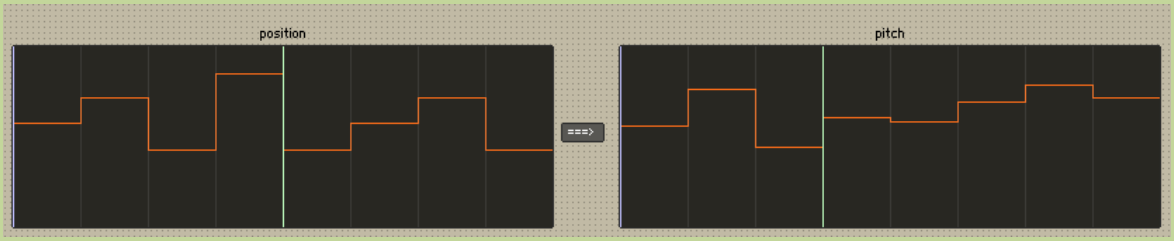
Запись псевдослучайных чисел в таблицу 20x20.



Как двигаться по таблице Event Table по определёной последовательности.

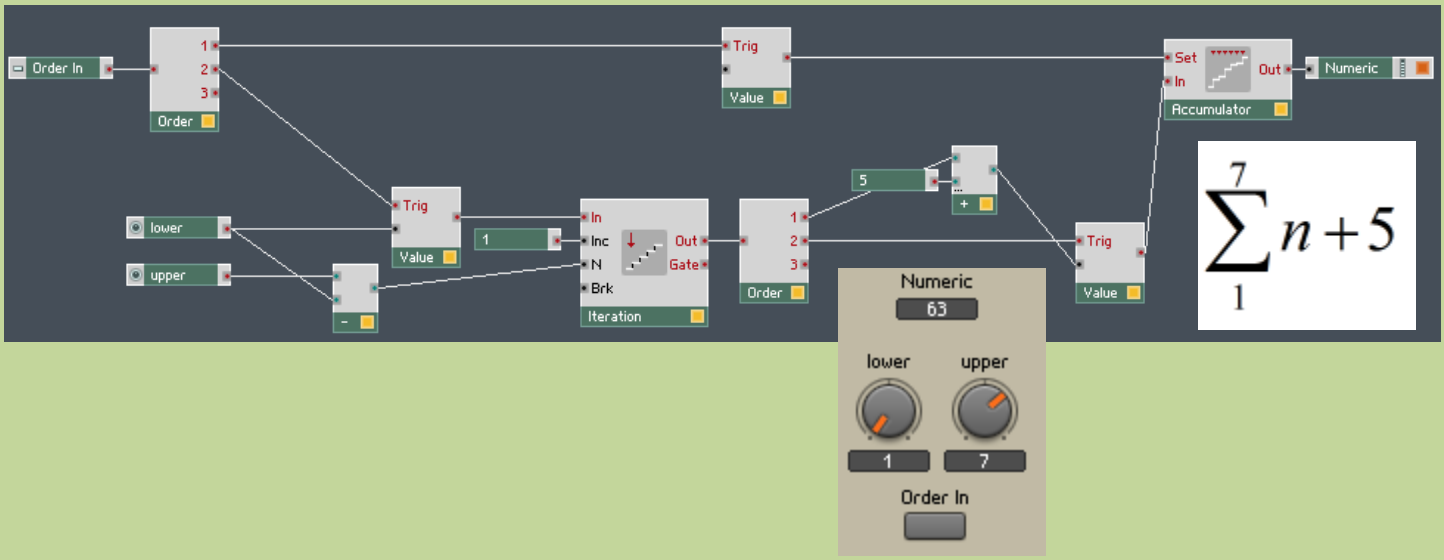
Допустим мы двигаемся по таблице по x=8, y=1. Value Range = 80. Такая таблица подходит для высоты звуков. Обычный счётчик считает от 0...7, итого 8. Выберем у таблицы Table Draw Mode и нарисуем ноты. Допустим ноты будут 60,72,65,80 и т.д .Вопрос : как начать двигаться допустим 72,60,80,72 и т.д. Можно конечно перерисовать таблицу, но есть более простой вариант позволяющий двигаться по таблице в любом направлении.





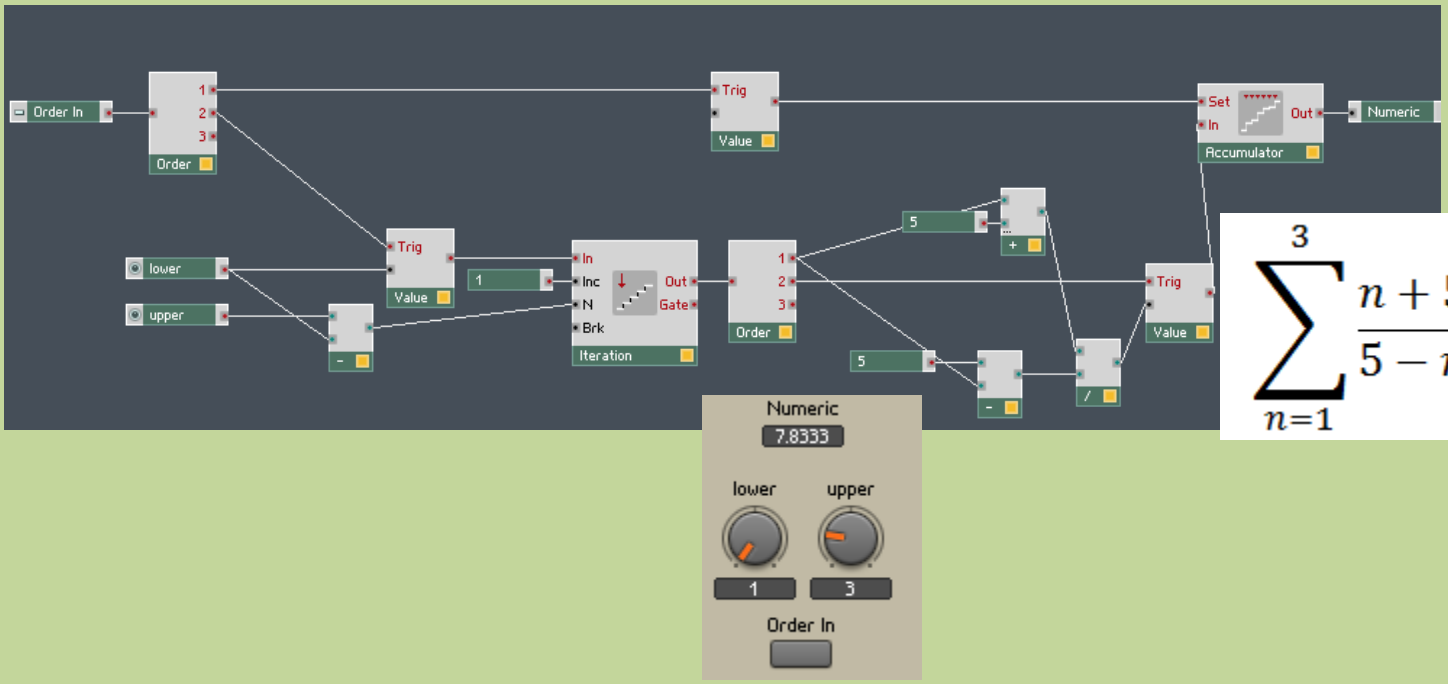
Рисуем в position путь для движения в pitch

Модуль Итерации и Сигма



$$\sum_{1}^{7} n+5$$

Если в качестве n идёт итерация и происходит мат.действия, то n надо использовать у Order – выход 2 для получения правильного ответа.



$$\sum_{n=1}^{3} \frac{n+5}{5-n}$$

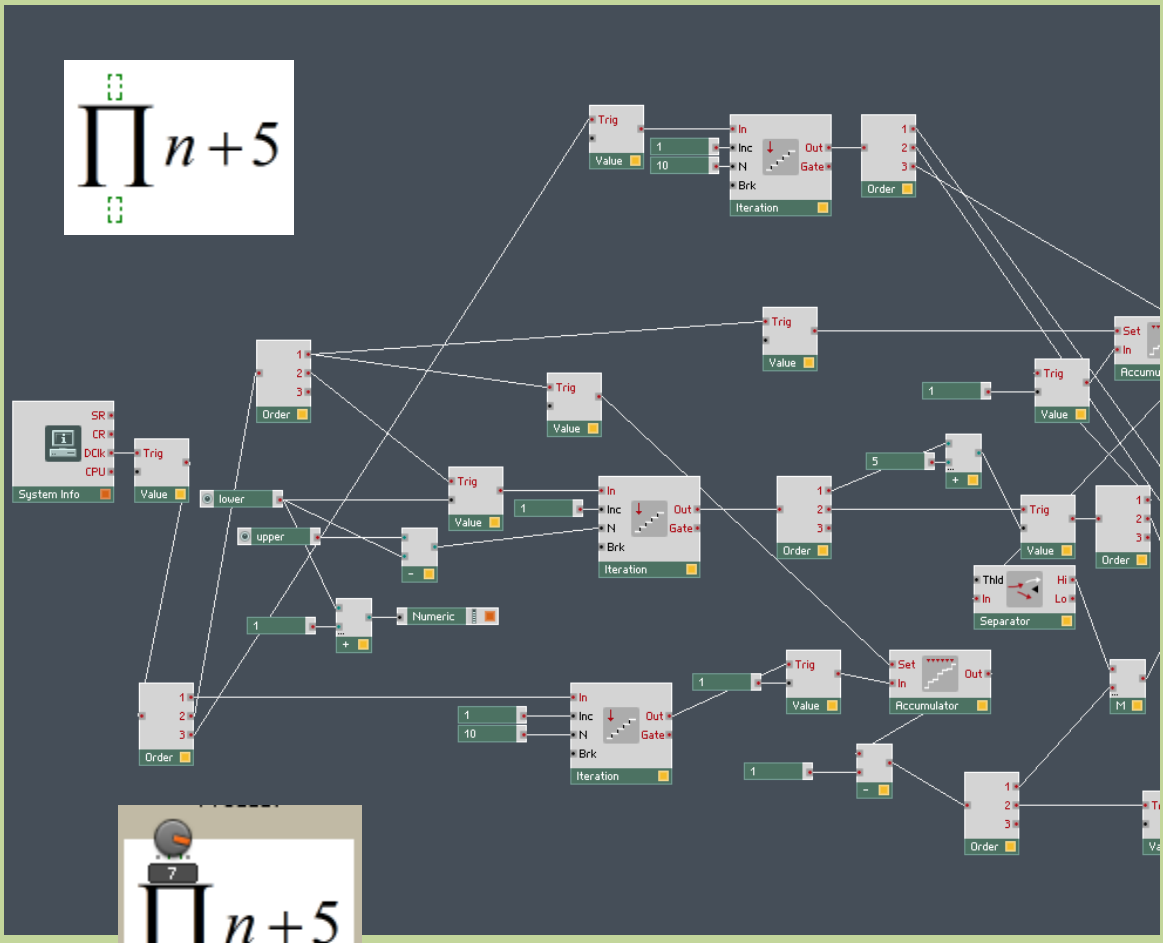
Замечание

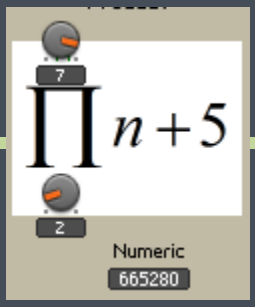
Если использовать нижестоящую формулу, то в реакторе будет ответ 16.833, хотя на самом деле это ошибочное выражение, так как в выражении встречается 10/0, а на ноль делить нельзя. Поэтому в реакторе надо обращать внимание на такие моменты и проверять расчёты в сторонем приложении.

$$\sum_{n=1}^{5} \frac{n+5}{5-n}$$

Indeterminate

Product series





Таким образом можно получать мат.выражения

Input

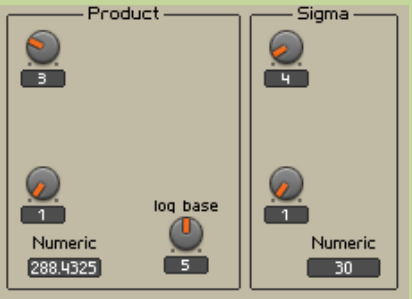
$$\sum_{n=1}^{4} n^2 \sqrt{\prod_{n=1}^{3} n + 5 - \log_5(n)}$$

Output

$$\sqrt[30]{6\,(-\log_5 2 + 7)\,(-\log_5 3 + 8)}$$

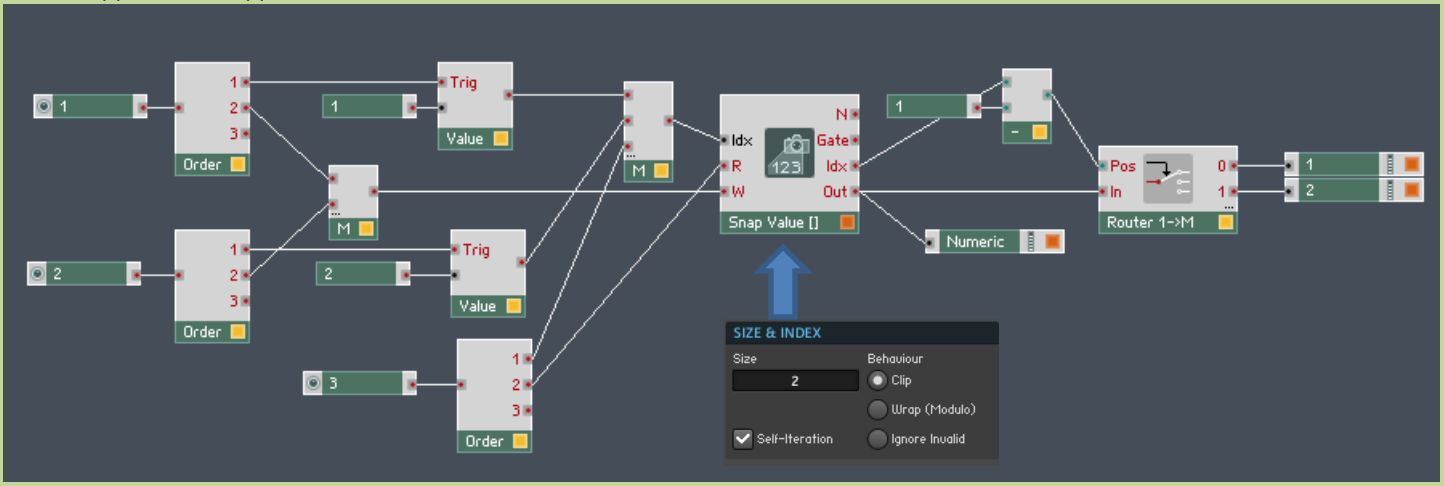
Decimal Output

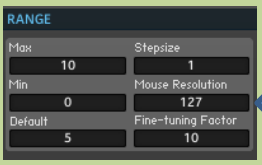
1.2078164608107



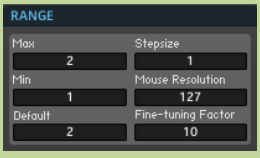
Snap Value Array

Массив для записи данных





Knobs 1,2



Knob 3

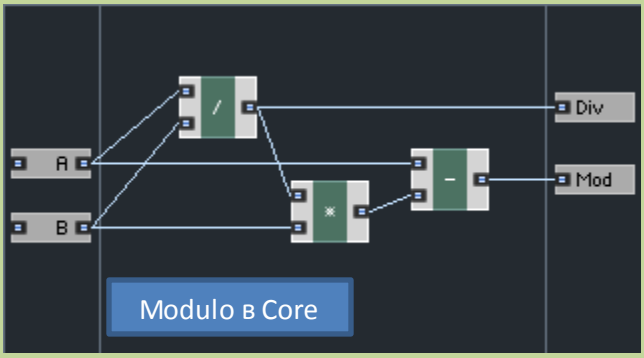
Крутя 1 и 2 создаём индексы Idx (Order 1) и одновременно записываем W (Order 2). Крутя 3 выбераем индексы и считываем R.

Modulo

Математический модуль . Два гибридных входа A и B и два гибридных выхода Mod и Div. Читается как - A по модулю B. Пишется A mod B. Пример : 6 mod 5.

Div : деление A на B без остатка. Пример : 27 div 6 = 4, так как 27/6=4,5=4

Mod : A – B*Div . Пример : 27 mod 11 = 5, так как Div=27/11=2, то 27-11*2=5



Формула Modulo

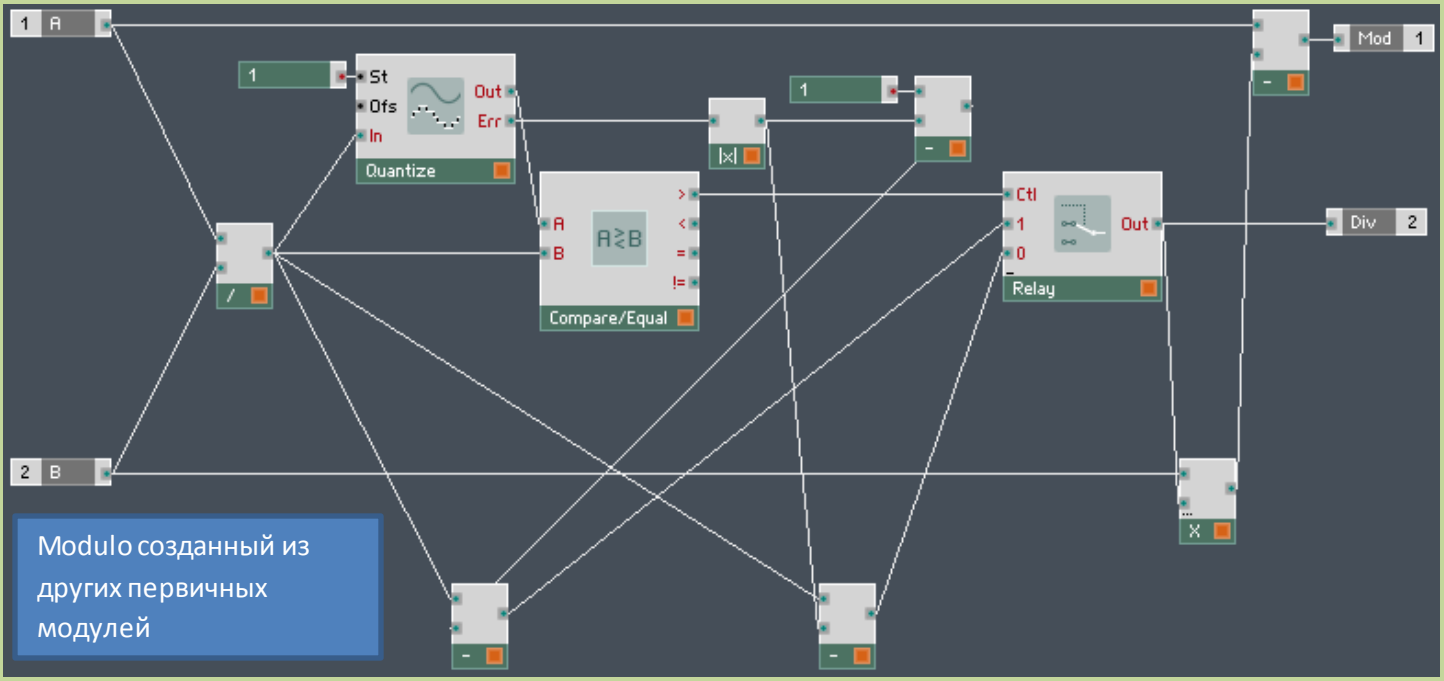
If, Quatize A/B≤A/B, Then Mod=

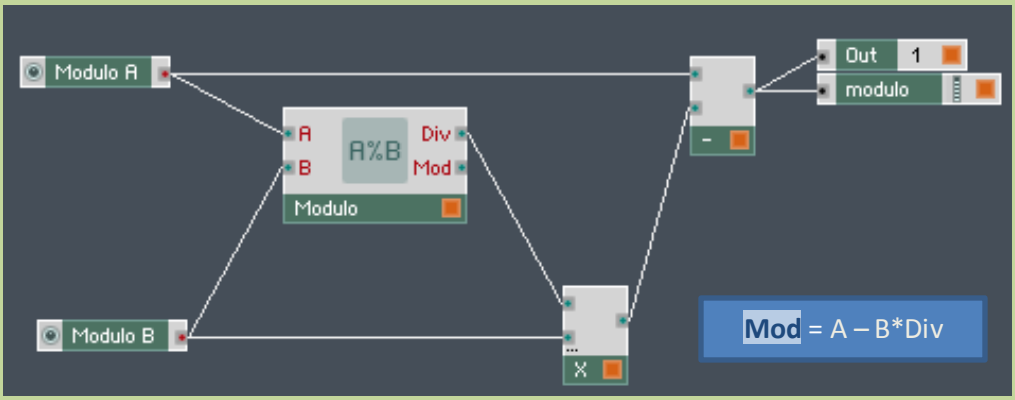
A-B*(A/B-|Err|)

If, Quantize A/B>A/B, Then Mod=

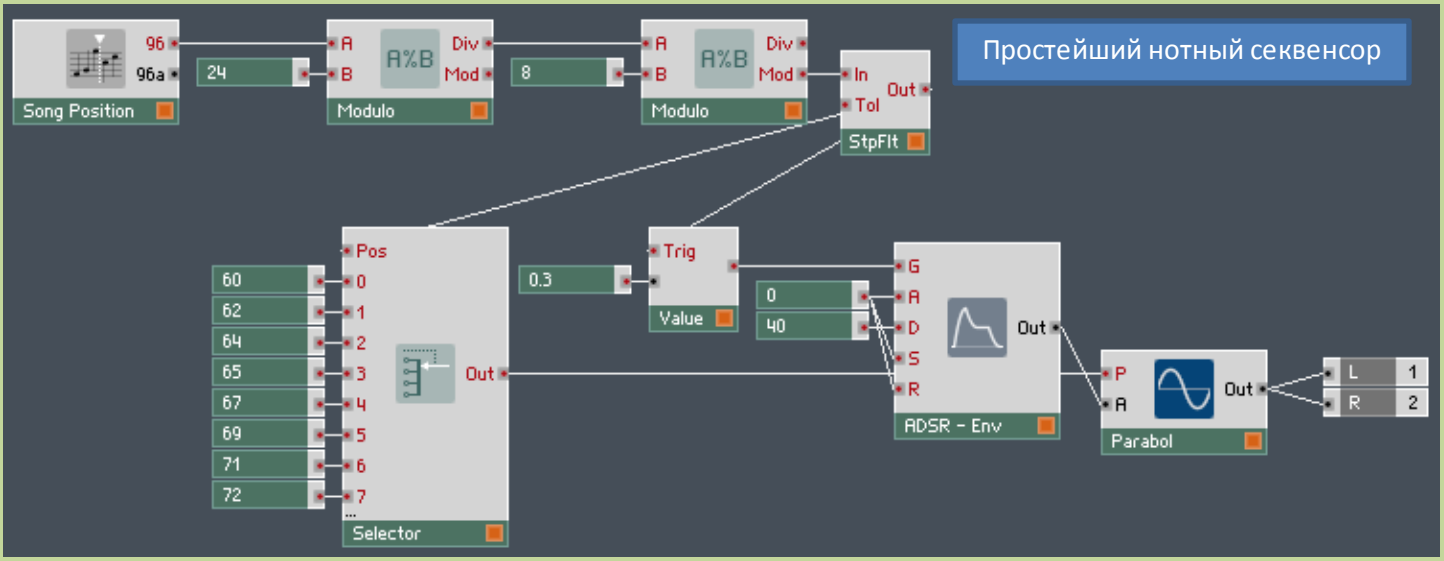
A-B*(A/B-(1-|Err|))

Where, Err=Quantization Error Signal A/B

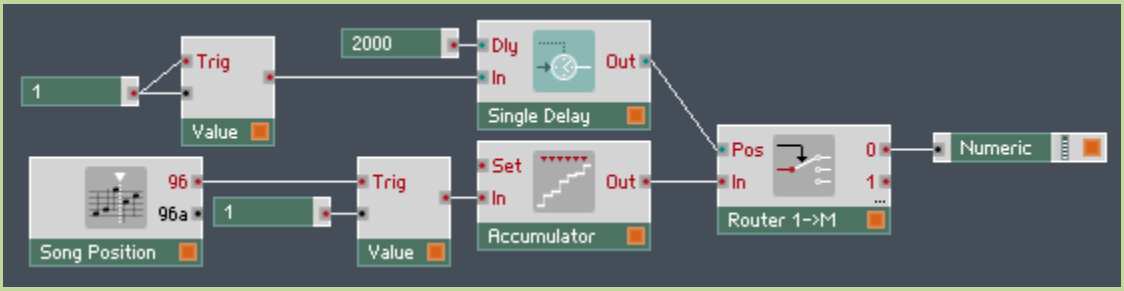




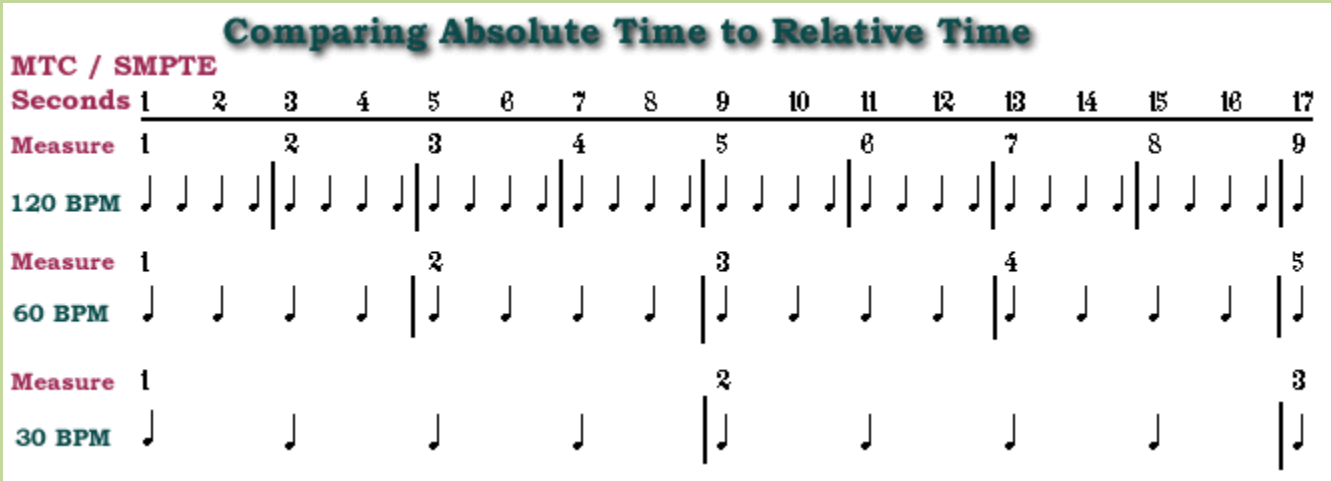
Применение Modulo в секвенсорах.



Song Position является тактовым генератором Event значений. При нажатие ▶ и значении 120.0 BPM , Song Position генерирует 96 событий каждые 2 секунды. Это можно проверить такой схемой, нажав два раза ⏻



96 значений за 2 секунды нужны для того, чтобы получать длительности нот (целые, половинные, четвертные и т.д). Это соответствует стандарту MIDI и выраженно в следующей таблице :



Чтобы получить эти длительности и применяется модуль Modulo с выходом Div. Четвертные ноты : после модуля Song Position ставится модуль Modulo (Вход A подключается к Song Position, а у входа B ставится константа 24) и сигнал берётся с выхода Div. Как вы помните Div это деление A на B без остатка, так как на вход A поступает 96 значений в порядке возрастания в течении 2 секунд, то выход Div будет иметь следующую таблицу :

96 events по возрастанию и по div 24	Значение Div
№ 1-24 , value 0-23 , /24	0 –> 24 раза в течении 500 миллисекунд
№ 25-48 , value 24-47 , /24	1 –> 24 раза в течении 500 миллисекунд
№ 49-72 , value 48-71 , /24	2 –> 24 раза в течении 500 миллисекунд
№ 73-96 , value 72-95 , /24	3 –> 24 раза в течении 500 миллисекунд

Как мы видем, у нас есть 4 значения : 0,1,2,3 каждый из них появляется через 500 мил лисекунд, но в течении этих 500 миллисекунд каждый повторяется 24 раза. Чтобы этого избежать , после выхода Div ставят модуль Step Filter, который не пропускает повторяющиеся значения. Так получается длительность четвертных нот. 0, через 500мс 1, через 500мс 2, и т.д . Все эти четвертные events подаются на триггер модуля Value, а потом на Gate. Song Position

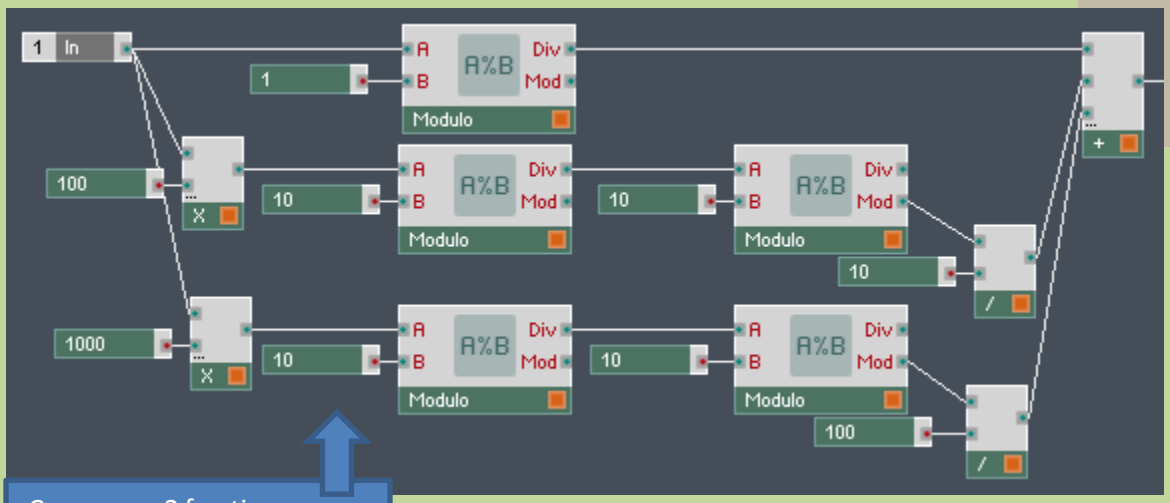
продолжает бесконечно генерировать числа, но всегда по 96 за 2 секунды при BMP=120. Если вы нажмёте  то произойдёт остановка, если ещё раз нажать, то продолжится с того же самого места, так как Song Position сохраняет текущее значение генерируемых чисел при остановке.

#	Label	Value
1	1/4	24
2	1/6	16
3	1/8	12
4	1/12	8
5	1/16	6
6	1/24	4
7	1/32	3

Несколько длительностей и их константы для входа B у Modulo

С помощью Modulo можно получать разнообразные схемы.

Пример : **Устранение дробной части**.



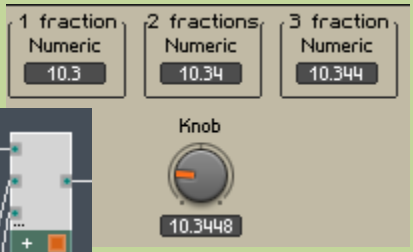
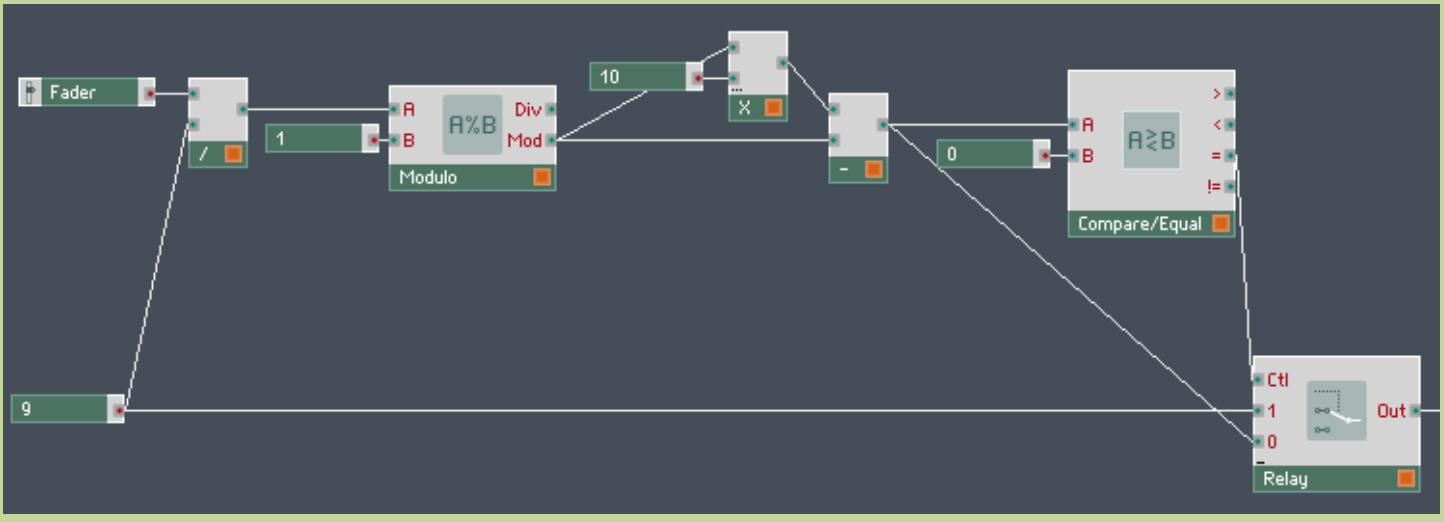
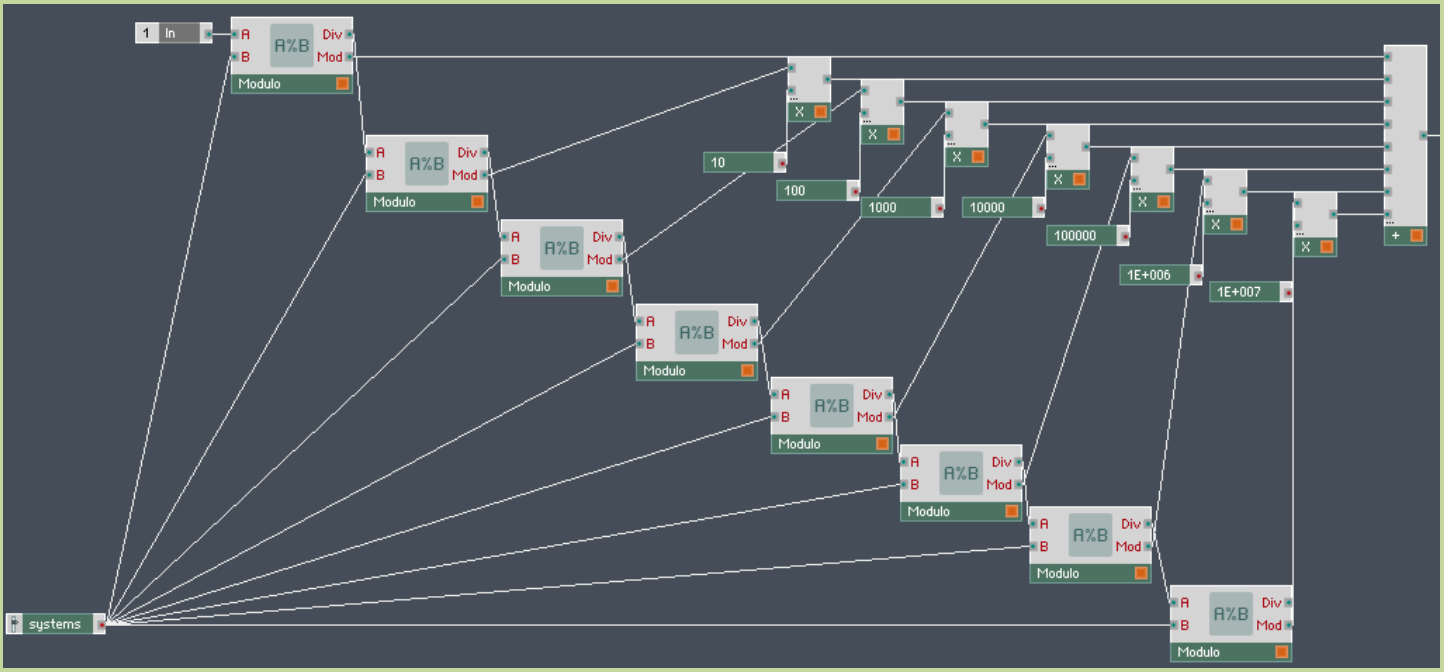


Схема для 2 fractions, допустим на вход In пришло значение 10.3448, тогда на выходе будет значение 10.34, последние две цифры дробной части будут откинуты

Пример : **Цифровой корень**. Цифровой корень это когда все цифры данного числа складываются, пока не останется одна цифра. Пример : 148=1+4+8=13, 1+3=4. Fader stepsize=1.

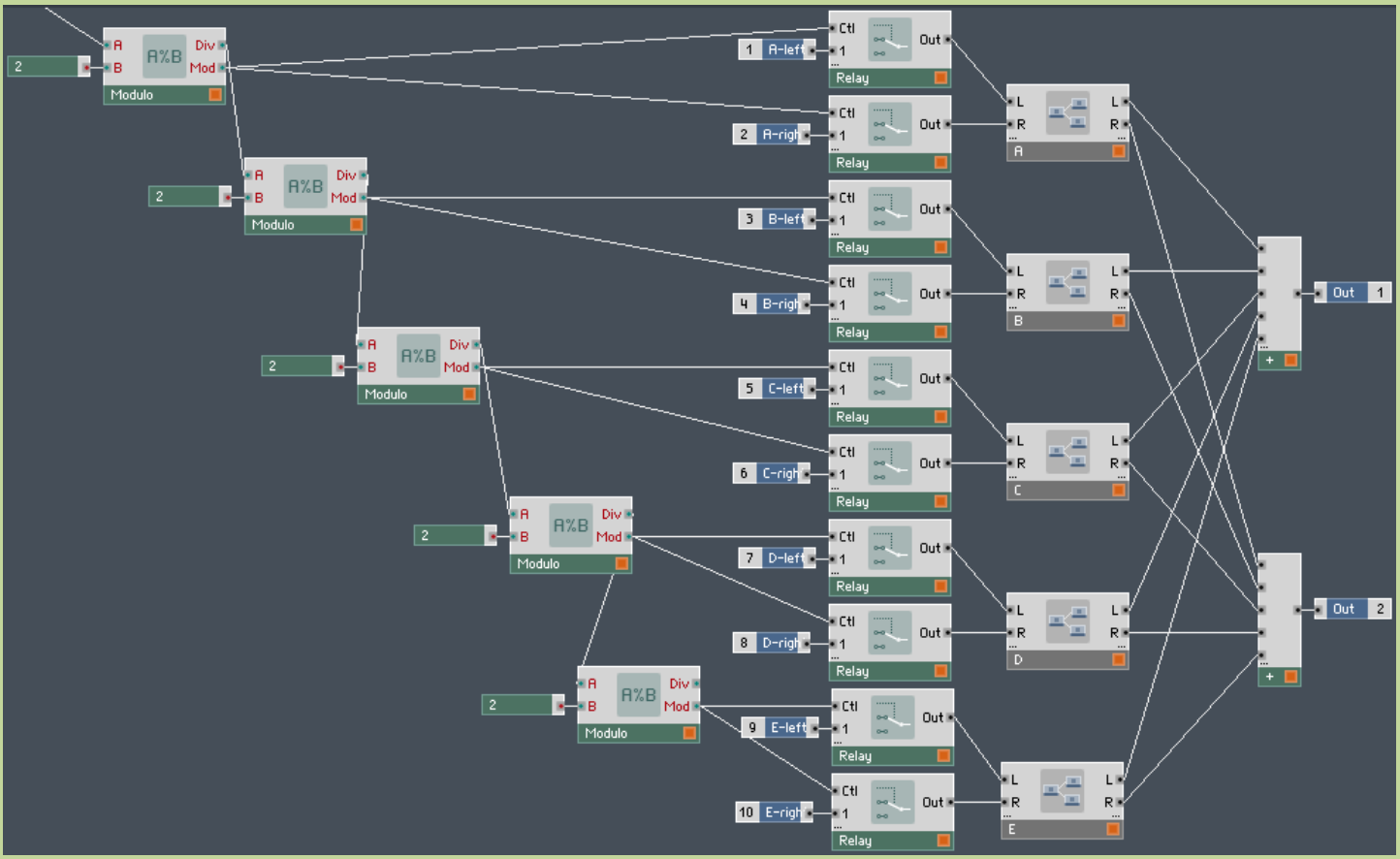


Пример : **Системы счисления**.

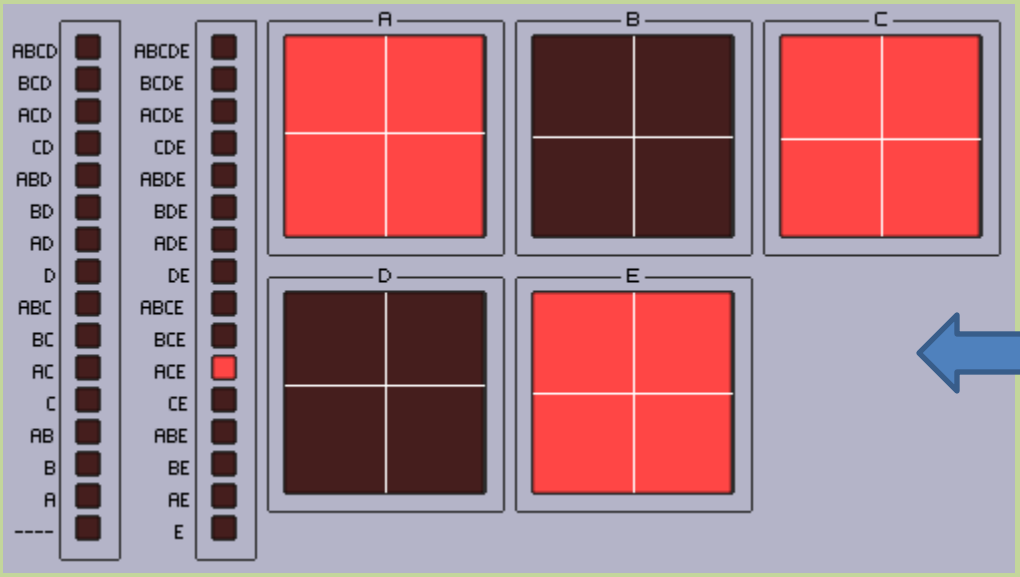


Если на вход In подать цифру 5, а у Knob-Systems выбрать 2, то на выходе будет 101 – число 5 в двоичной системе счисления это 101. Меняя положения Knob-Systems (от 2 до 10) мы выбираем системы счисления – двоичная, троичная и т.д вплоть до десятичной. На In подадим число 32, а System выберем 7, выход будет 44, так как в семеричной системе счисления 32 это 44.

Пример : **аудио-миксер с 5 стерео входами и 32 пермутации каналов**.



Так как с помощью Modulo можно число из десятичной системы перевести в двоичную, то можно это использовать в комбинациях Relay –реле, так как двоичная система и сигналы реле базируются на 0 и 1. У нас есть 5 каналов A,B,C,D,E. Если мы переберём их варианты перестановки , то получим 32 варианта и следующую схему :



Так как управление-Ctl реле работает на 0 и 1, то можно с помощью этого включать (1) и отключать (0) каналы следующим образом. Так как у нас есть 32 варианта перестановки, то подадим для примера 15 на верхний модуло, он выдаст сигнал на остальные модуло пока не получим на всех модуло – 01111 – это 15 в бинарной форме. Таким образом можно все комбинации от 0-31 итого 32, преобразовывать в 0 и 1 и переключать каналы 32 различными способами.

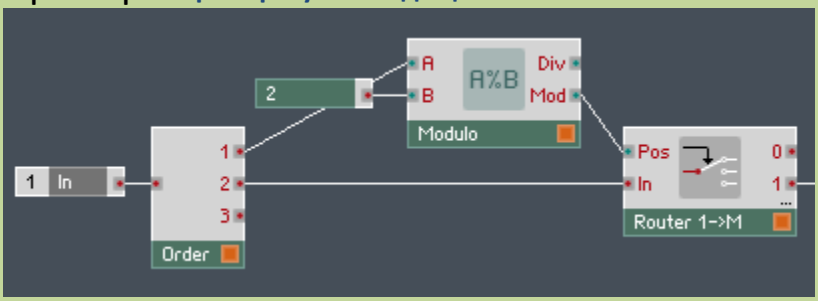
Проверка формулы
Дано: 17mod2
Quantize 17/2=9 > 17/2=8.5
17-2*(17/2-(1-|0.5|))=1
17mod2=1

Формула Modulo
If, Quatize A/B≤A/B, Then Mod=
A-B*(A/B-|Err|)
If, Quantize A/B>A/B, Then Mod=
A-B*(A/B-(1-|Err|))
Where, Err=Quantization Error Signal A/B

Дано: 23mod3
Quantize 23/3=8 > 23/3=7.66
23-3*(23/3-(1-|0.34|))=2
23mod3=2

Хотя конечно для нахождения mod легче использовать A – B*Div, но с помощью такой формулы можно создать Modulo из других модулей.

Пример : **преобразуем входящие значения в нечётные или чётные**.

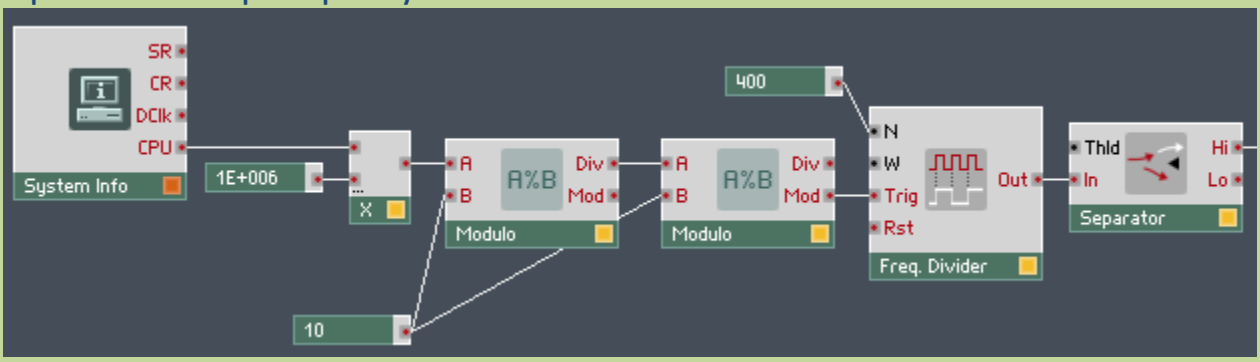


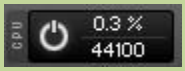
Выход 1 с Router пропускает нечётные, а выход 0 чётные значения.

Пример : **округляем в меньшую сторону**

AdivB,(B=1)<A

Простой генератор случайных чисел.

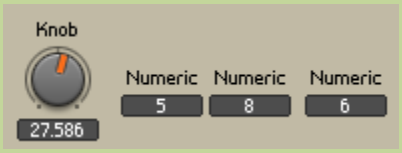


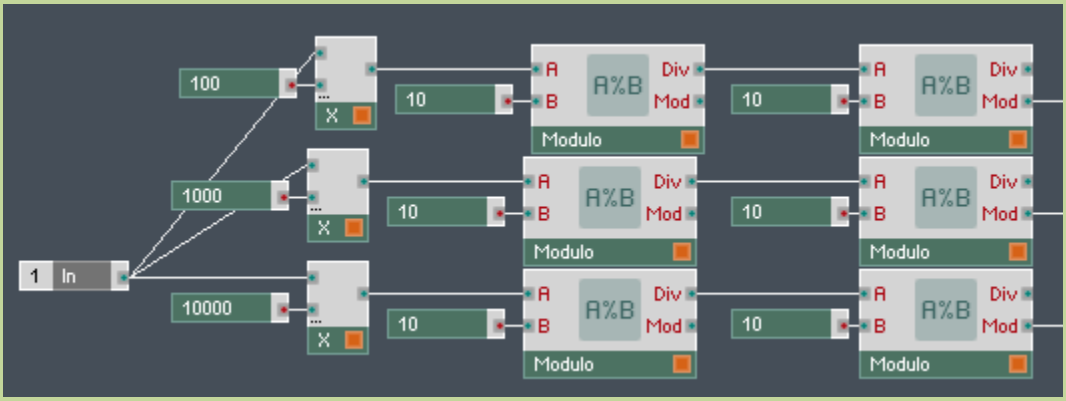
Хотя CPU и показывает нагрузку на процессоре в 0.3 % 

, но на самом деле за 1 секунду сигнал на

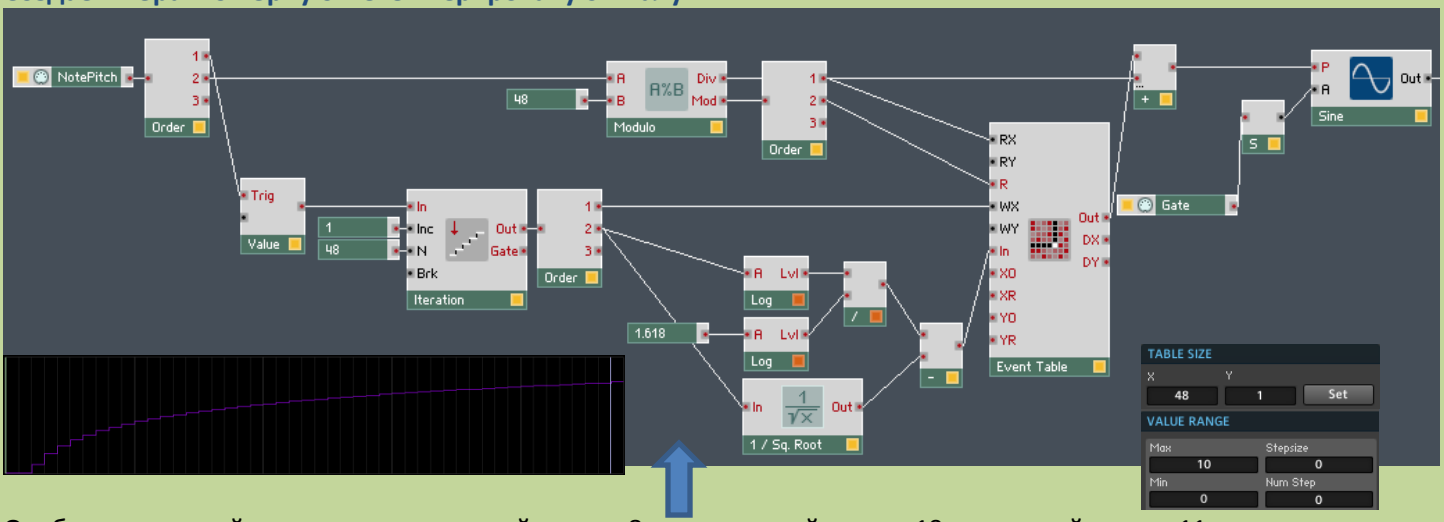
CPU меняется 400 раз и представляет собой не 0.3 %, а допустим 0.351574. Эта схема откидывает все цифры кроме последней. Если сохранить и открыть то вы увидите примерно такую последовательность 1,4,8 т.д , если снова открыть , то последовательность будет другой.

Вытаскиваем числа после запятой

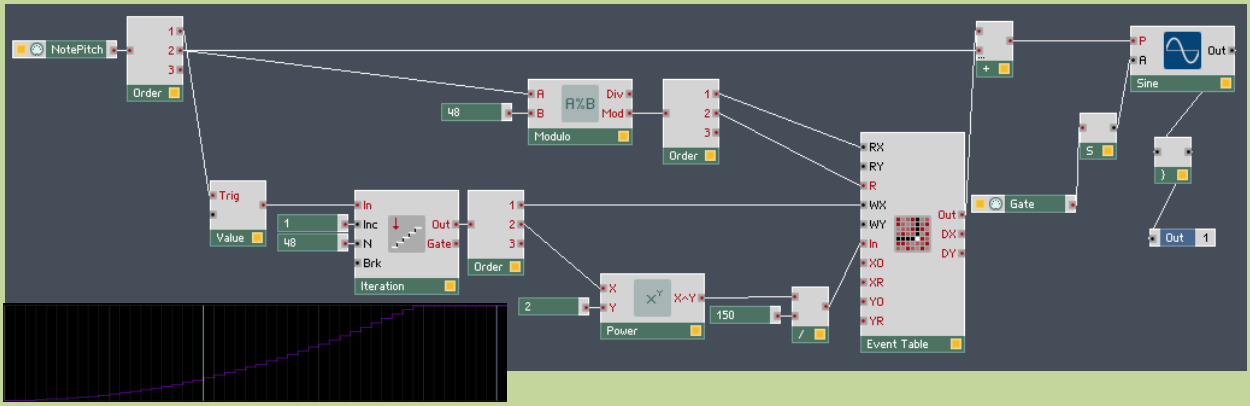


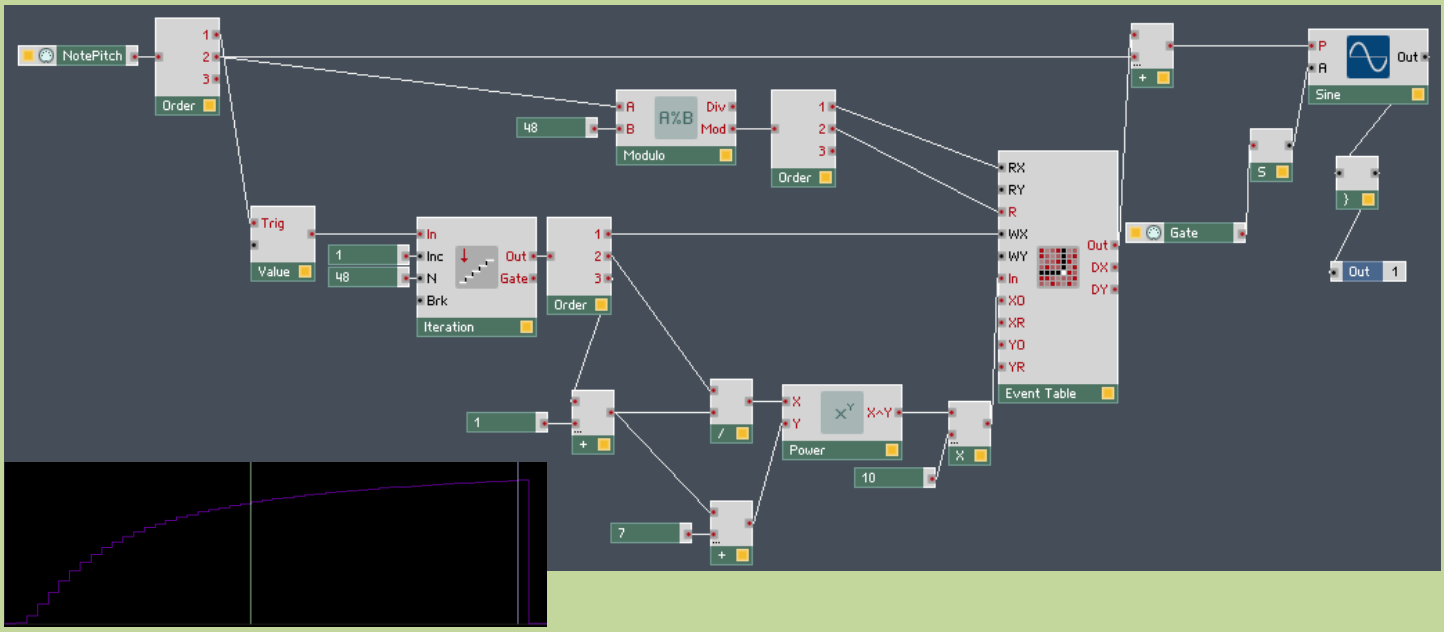


Создаём неравномерную-нетемперированую шкалу



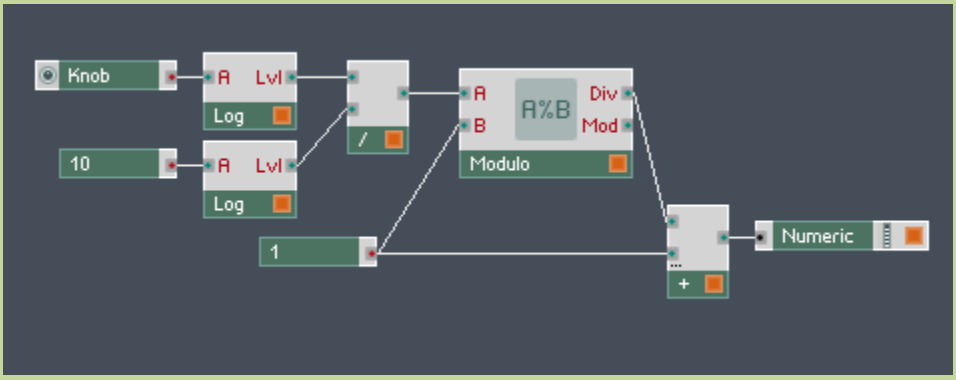
Особенность такой темперации – в малой октаве 8 нот, в первой октаве 10, во второй октаве 11 нот.

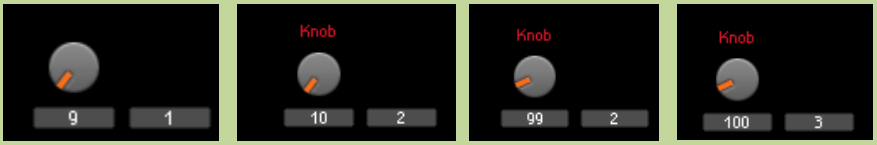




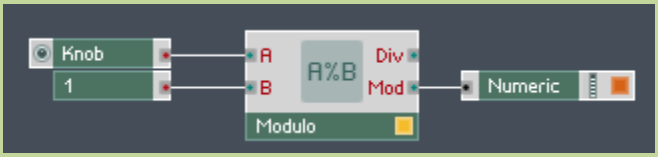
Если убрать у Event Table входы WX и In, далее нажать правой клавишей мыши по Event Table и выбрать Table Draw Mode то можно нарисовать свою собственную шкалу.

Сколько цифр в числе





Значения после запятой

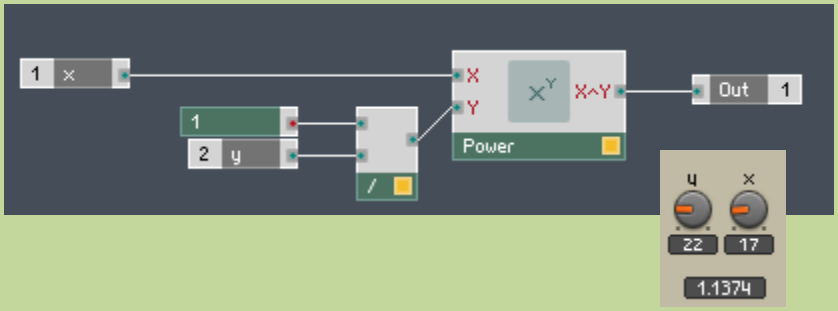


Если входящее значение числа имеет дробную часть, то дробную часть можно использовать отдельно.

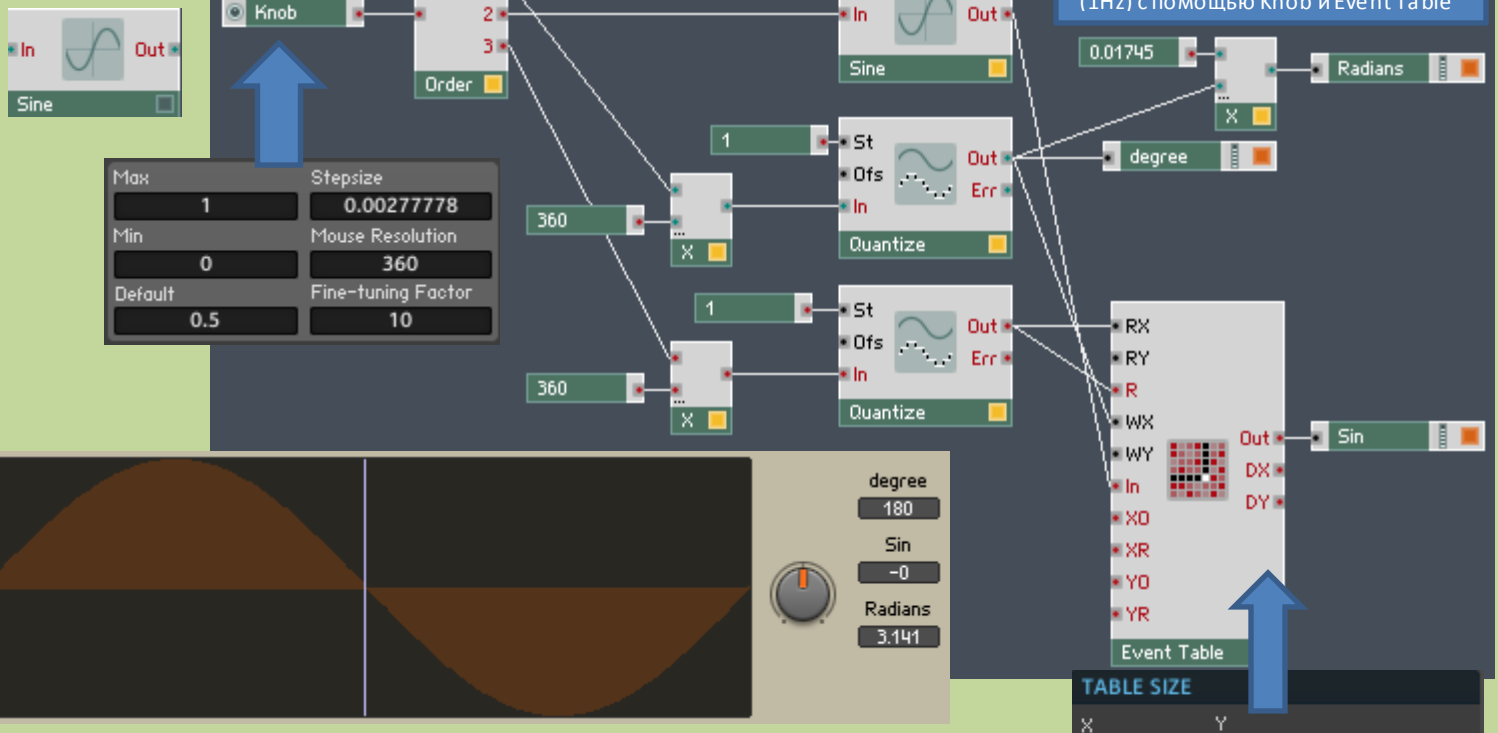
Корень числа

$\sqrt[y]{x}$

$\sqrt[22]{17}$
$\sqrt[22]{17}$
1.1374426178562



Sine



Рисуем полный период синусойды (1Hz) с помощью Knob и Event Table

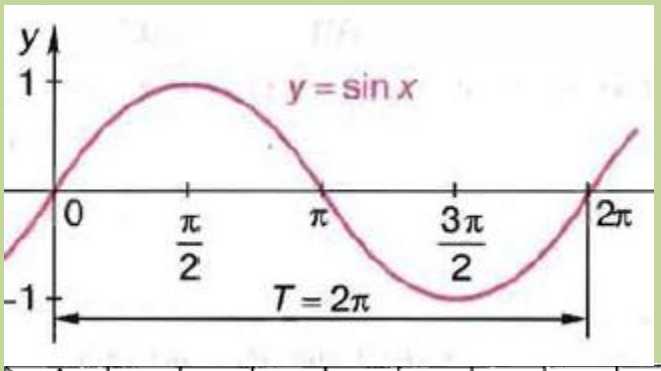


TABLE SIZE

X	Y	
361	1	Set

MODE

Interpolation	Clip / Wrap XY
None	Clip

VALUE RANGE

Max	Stepsize
1	0
Min	Num Step
-1	0
Default	Display
0	Numeric

Аргумент α / Функция	0	π/6	π/4	π/3	π/2	2π/3	3π/4	5π/6	π
sin α	0	1/2	√2/2	√3/2	1	√3/2	√2/2	1/2	0
cos α	1	√3/2	√2/2	1/2	0	−1/2	−√2/2	−√3/2	−1

Аргумент α / Функция	π	7π/6	5π/4	4π/3	3π/2	5π/3	7π/4	11π/6	2π
sin α	0	−1/2	−√2/2	−√3/2	−1	−√3/2	−√2/2	−1/2	0
cos α	−1	−√3/2	−√2/2	−1/2	0	1/2	√2/2	√3/2	1

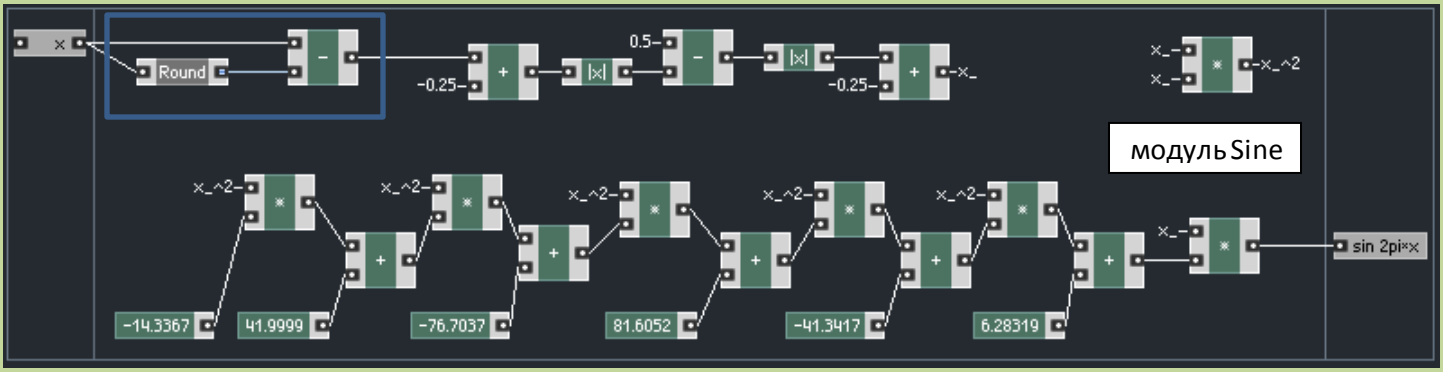
Углы в градусах	Углы в радианах	Углы в градусах	Углы в радианах
0°	0	180°	π
30°	π/6	210°	7π/6
45°	π/4	225°	5π/4
60°	π/3	240°	4π/3
90°	π/2	270°	3π/2
120°	2π/3	300°	5π/3
135°	3π/4	315°	7π/4
150°	5π/6	330°	11π/6
		360°	2π

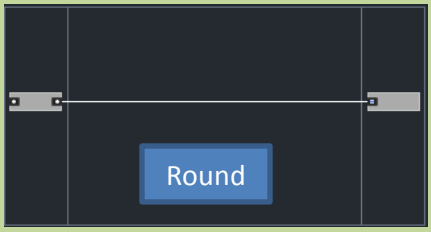
α° – градусная мера,
α – радианная мера
$\alpha^\circ : \alpha = 180^\circ : \pi$

$$\alpha^\circ = \frac{180^\circ \times \alpha}{\pi}$$

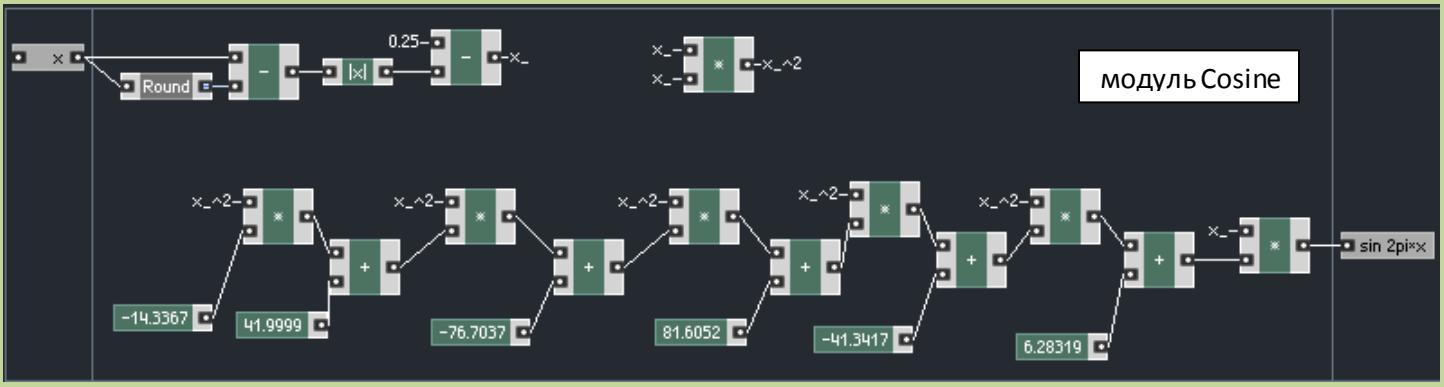
$$\alpha = \frac{\pi \alpha^\circ}{180^\circ}$$

Так как полный период равен 2*пи, а в градусной мере равен 360 градусам, то создадим Knob (чтобы получить Stepsize нужно 1/360=0,002777, тогда получим 360 шагов). Event Table – 1.сначало происходит выбор ячейки – куда будет происходить запись - WX (write to X – запись в ячейку по X), 2.затем происходит запись значения в эту ячейку , 3. затем происходит считывание с ячейки - RX (read to X – прочитать ячейку по X), а R является триггером. Degree (Градусы) и Radians (Радианы) – это Numeric Readout . Sin – Numeric Readout – sinα.
На модуль Sine подаётся значения от 0 до 1, он преобразует эти значения в значения функции y=sinα. Если подавать значения выше 1, то ничего не изменится – так как модуль Sine обрабатывает входящие значения по принципу Wrap :

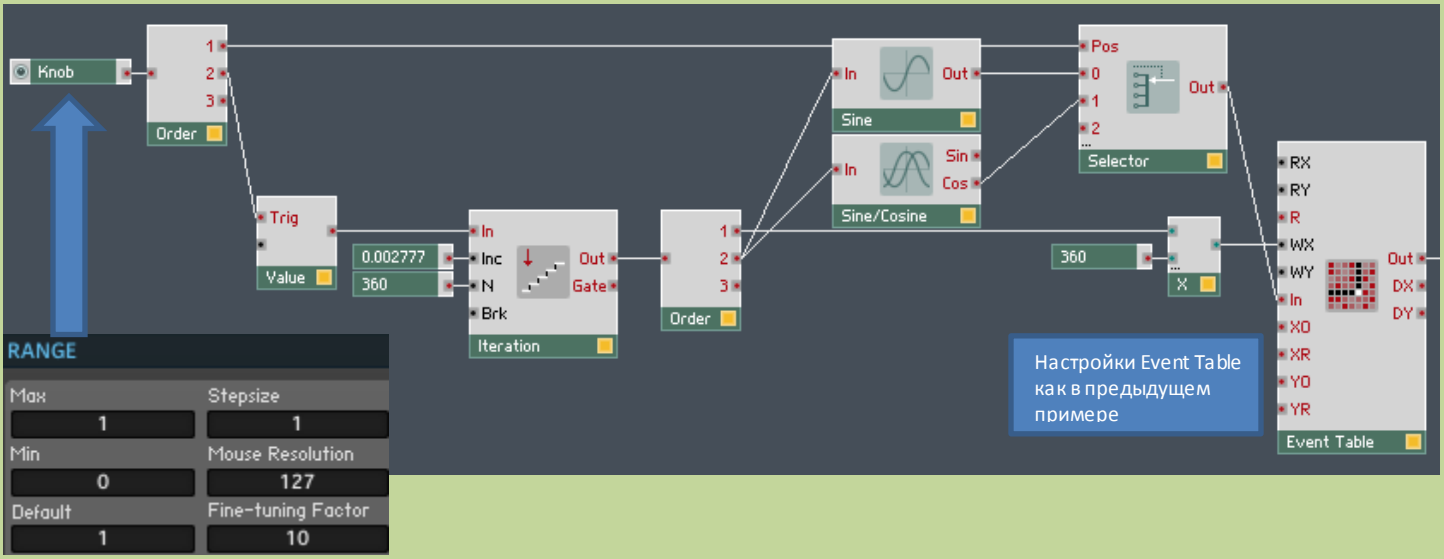




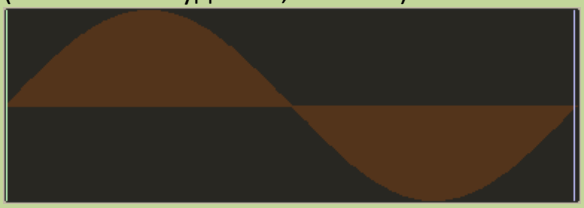
Синим выделен Wrap – он преобразует входящие значения в значения от -0.5 до 0.5. Поэтому важно помнить – если подавать значения на вход с шагом Stepsize=1, то вы будете с каждым новым значением получать один и тот же результат. Пример : 0.4=0.5877, 1.4=0.5877, 2.4=0.5877. Поэтому подавайте для модуля Sine/Cosine дробные значения с шагом 0.1 например.



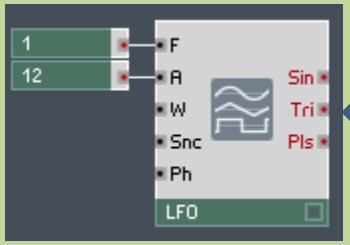
Получение мгновенного снимка полного периода (1Hz) синуса и косинуса



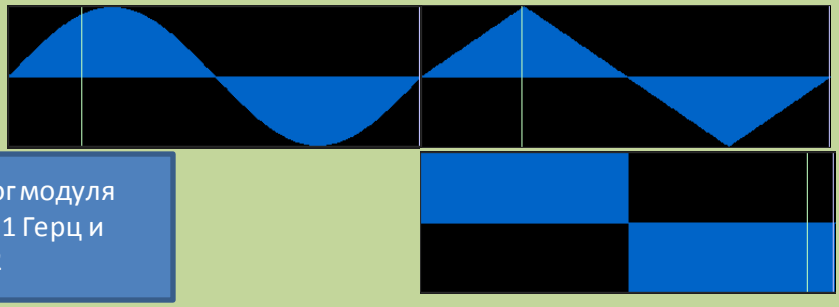
Меняя значения Knob с единицы на ноль и наоборот, мы сперва согласно модулю Order переключаем вход модуля Selector, дальше с выхода №2 модуля Order идёт на Value с нулём для того чтобы дальнейшая итерация начиналась с нуля. Дальше идёт итерация из 360 шагов от нуля до 360 с приращением 0.002777 к каждому шагу (1/360). Дальше итерация идёт через Order для того чтобы на WX у Event Table пришли значения раньше чем на In (Система – 1.Куда?WX, 2.Что?In). Меняя Knob получаем моментальные снимки :

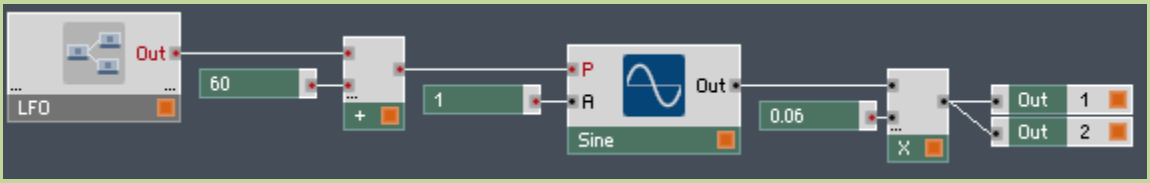


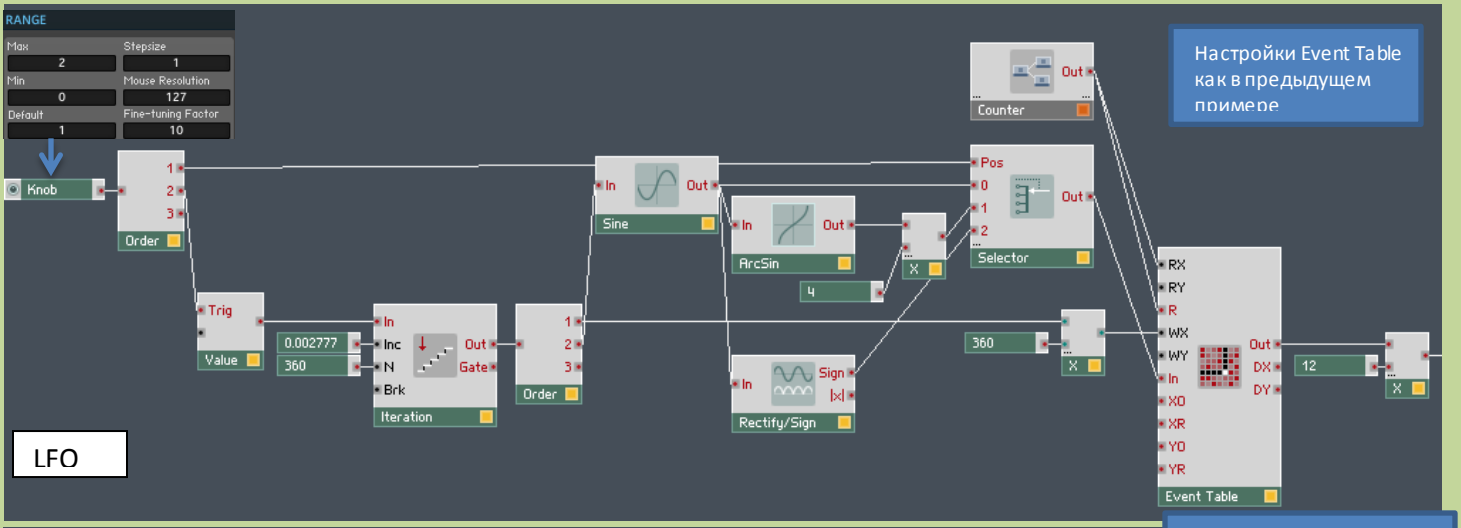
Создаём LFO



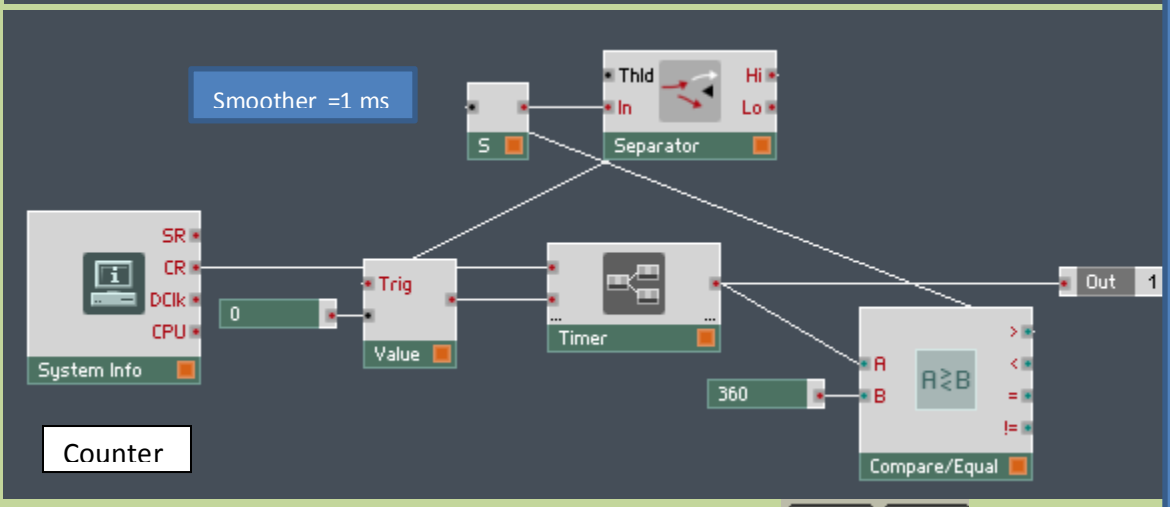
Создаём аналог модуля LFO с частотой 1 Герц и амплитудой 12



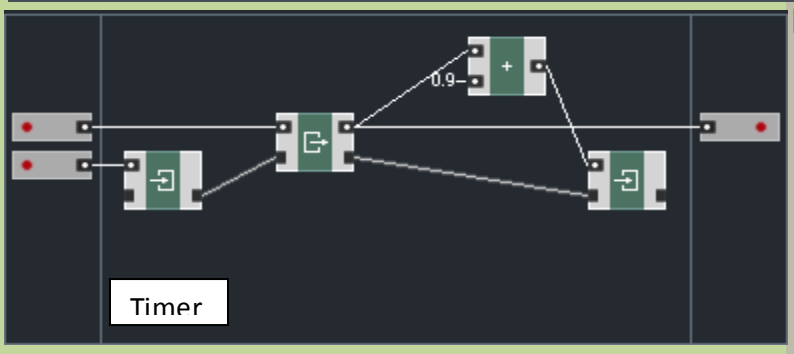




Настройки Event Table как в предыдущем примере



Меняя положение Knob мы сперва переключаем вход селектора(3 вида – синусойда-Sine, треугольная форма-Sine to ArcSin*4, пульс-Sine to Sign). Дальше идёт итерация с записью в таблицу. Считывание проводит Counter-Счётчик и подаёт считаные значения на выход Out

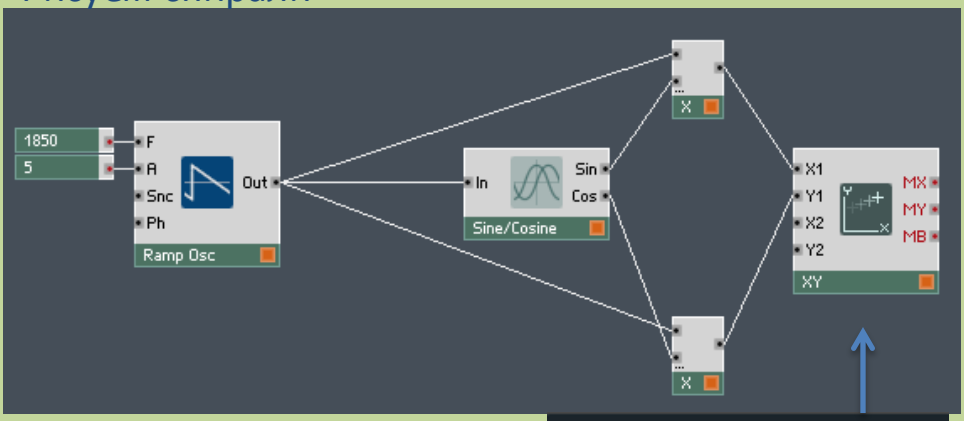


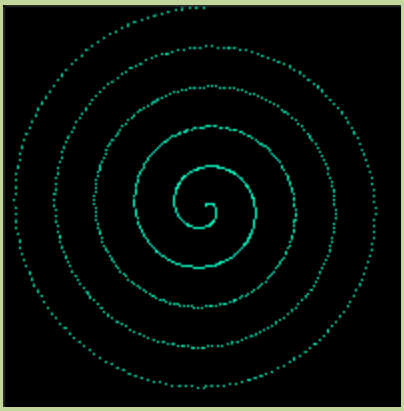
Event #	Value
385	345.5987
386	346.4987
387	347.3987
388	348.2987
389	349.1987
390	350.0987
391	350.9987
392	351.8987
393	352.7986
394	353.6986
395	354.5986
396	355.4986
397	356.3986
398	357.2986
399	358.1986
400	359.0986

Здесь показано как за 1 секунду Counter считает до 359

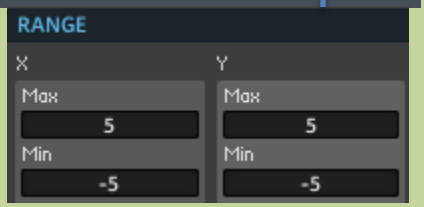
Чтобы получить частоту 2 Гц, нужно в Timer вместо 0.9 поставить 1.8. При значении 1.8 мы пройдём дважды полный период, что и есть 2 Гц.

Рисуем спирали



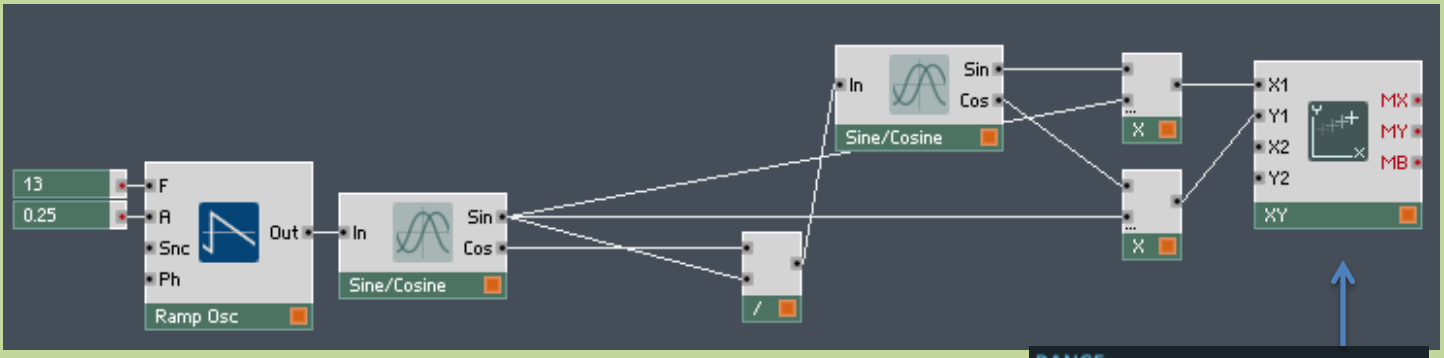


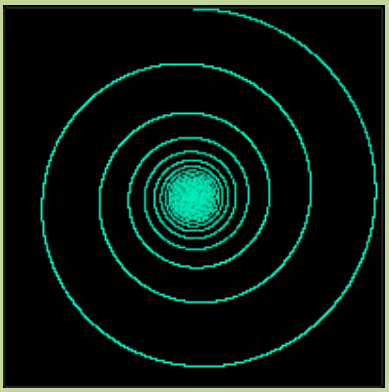
XY модуль в режиме Scope



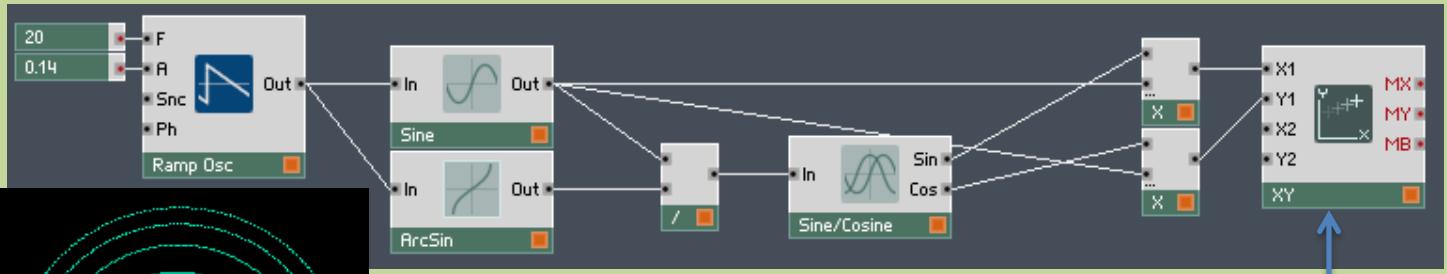
Архимедова спираль

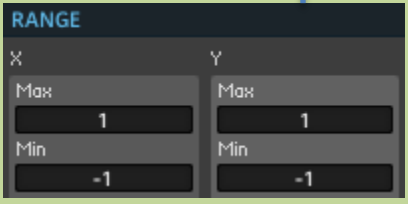
Так как Ramp выдаёт 400 значений в порядке возрастания от 0 до 1 за секунду, то он идеально подходит для подачи на тригонометрические модули.

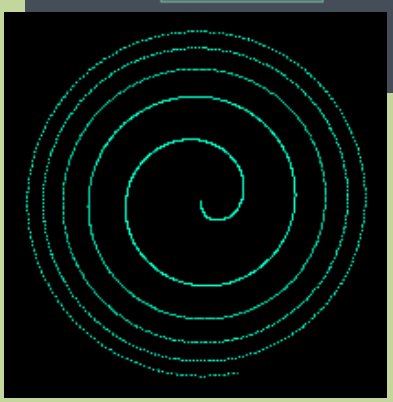




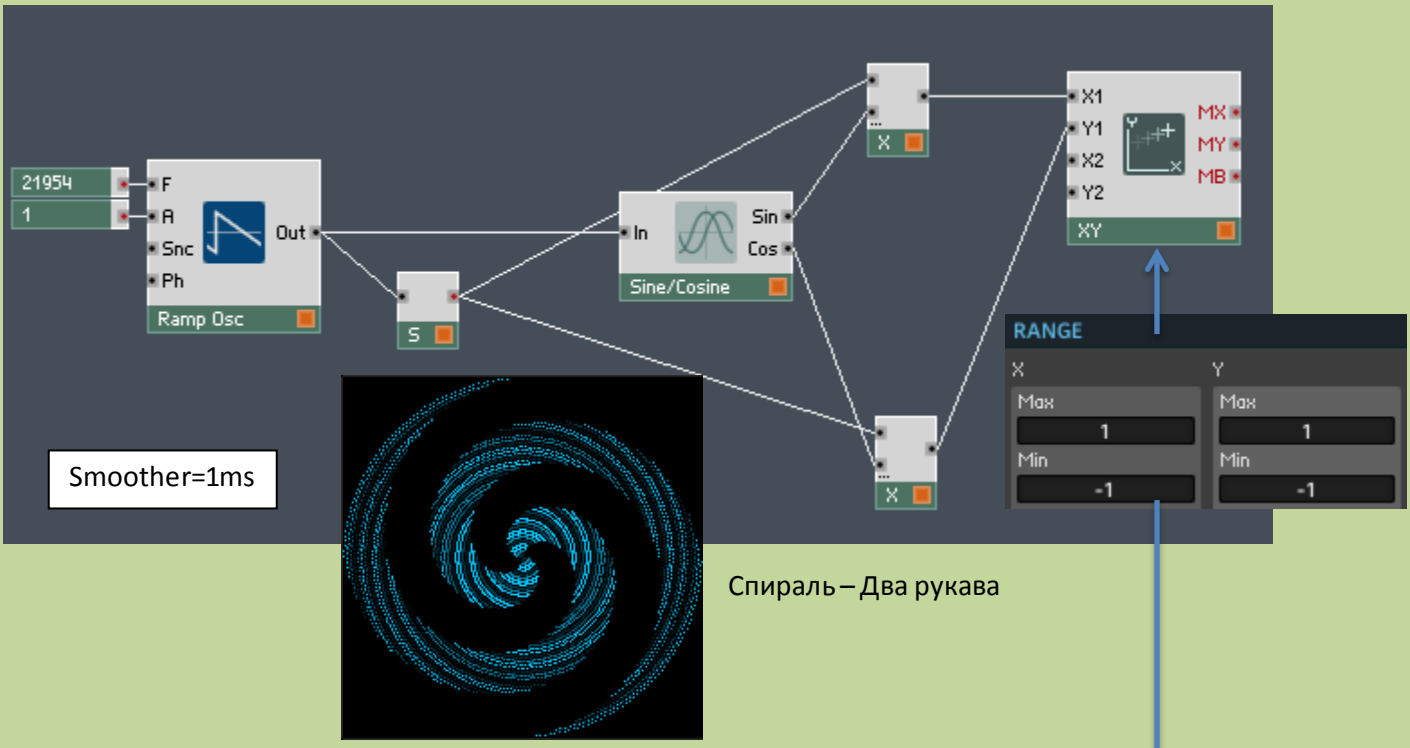
Логарифмическая спираль



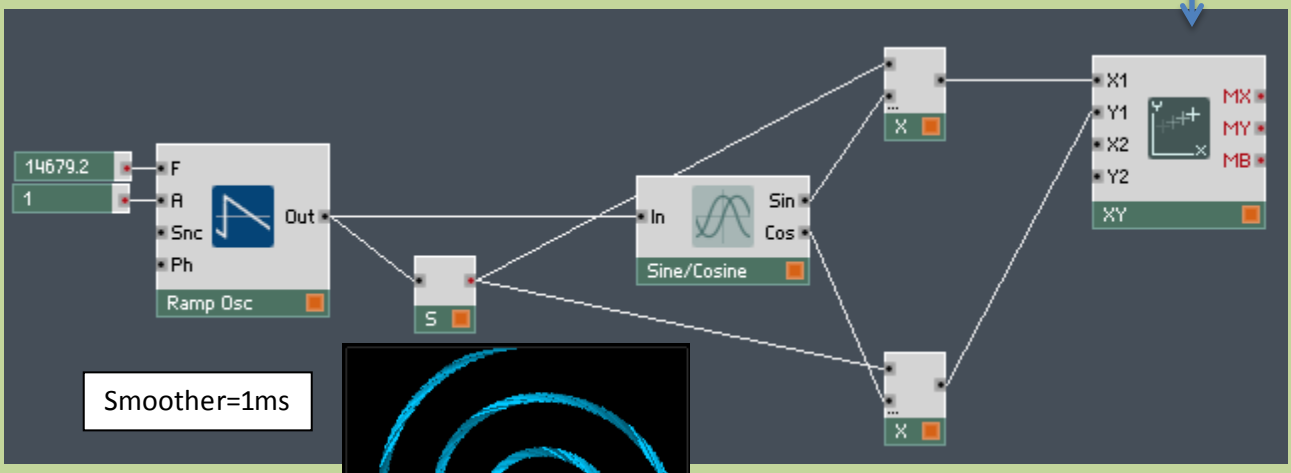




Ферма спираль

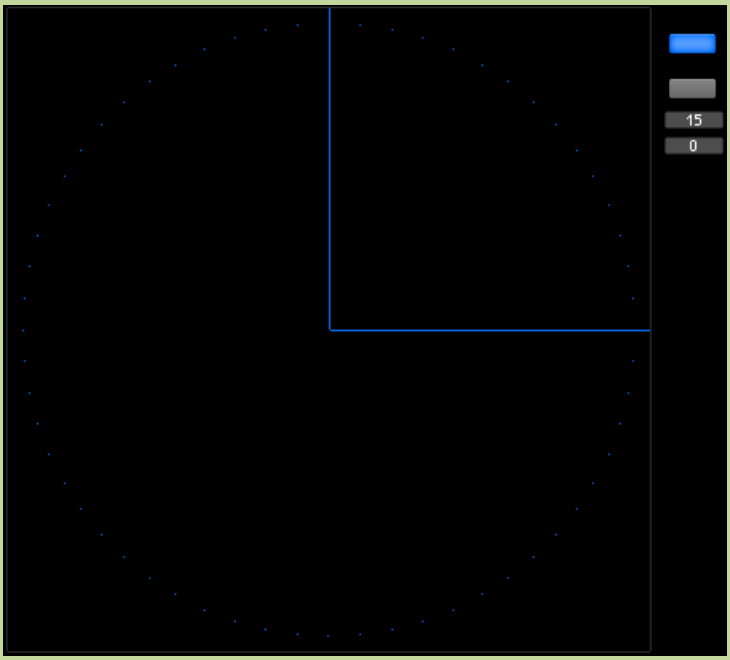


Спираль – Два рукава

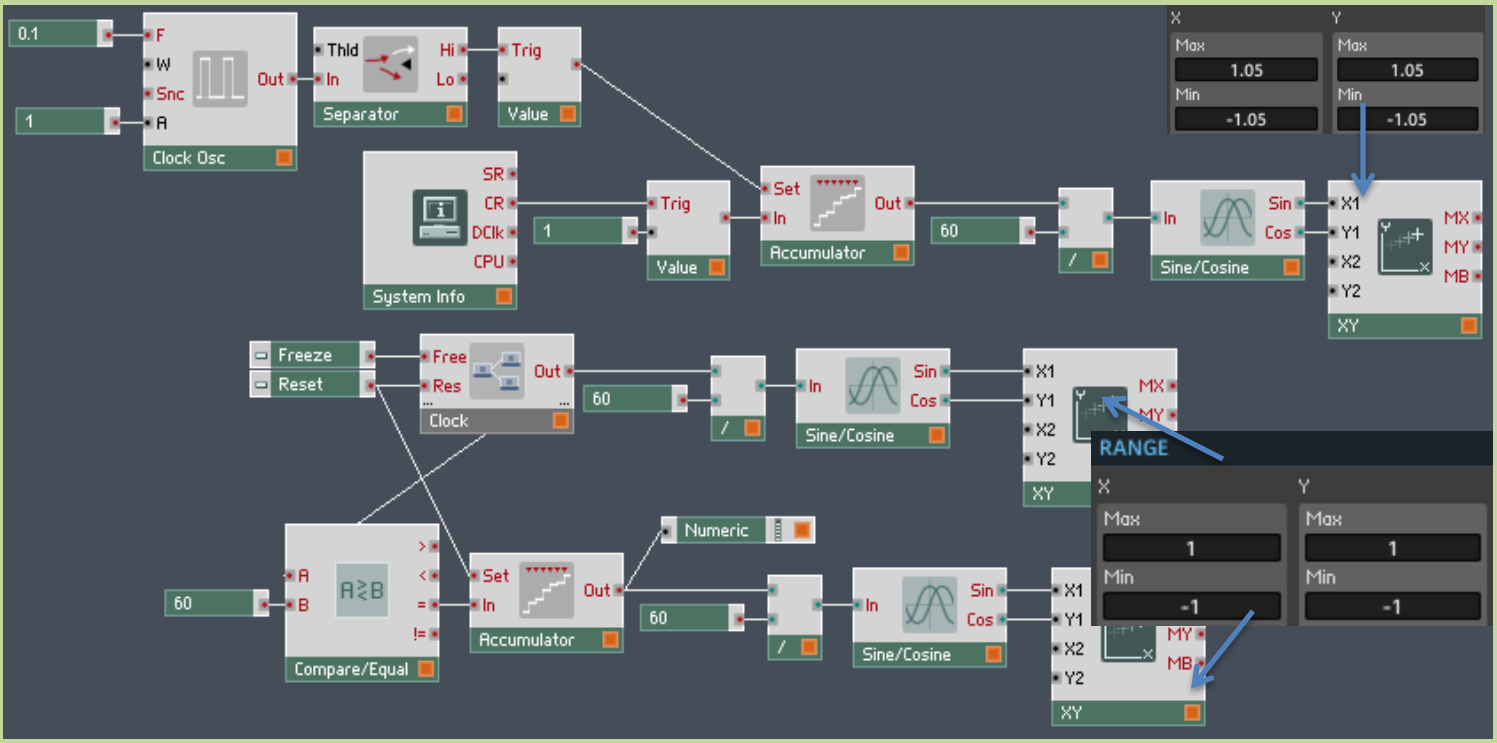


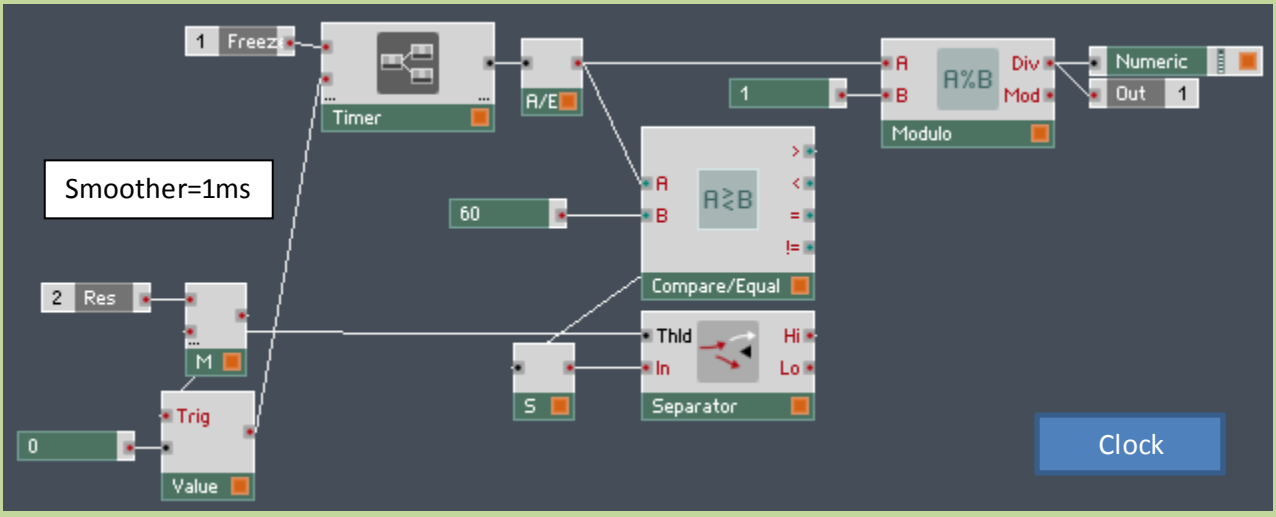
Спираль – Три рукава

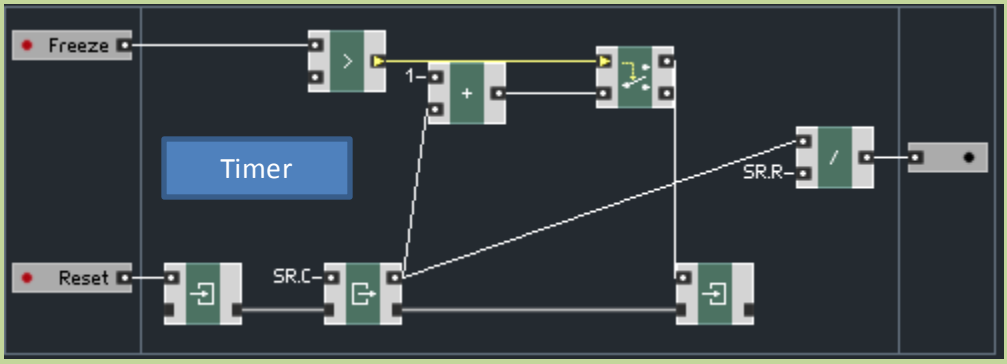
Создаём часы.



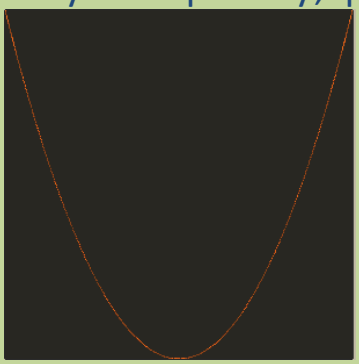
Часы содержат табло с 60 секундным делением, минутную и секундную стрелку, кнопку сброса, кнопку остановки и продолжения. Верхняя кнопка Freeze работает в режиме Toggle, нижняя кнопка Reset работает в режиме Trigger (в настройках 0=On Value)

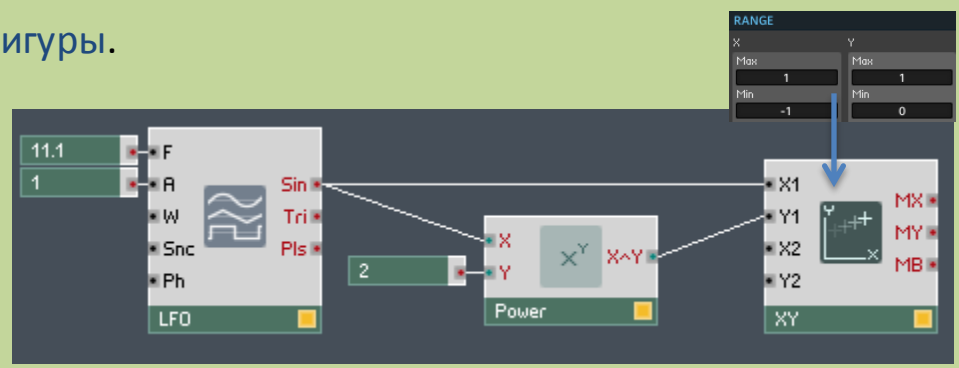


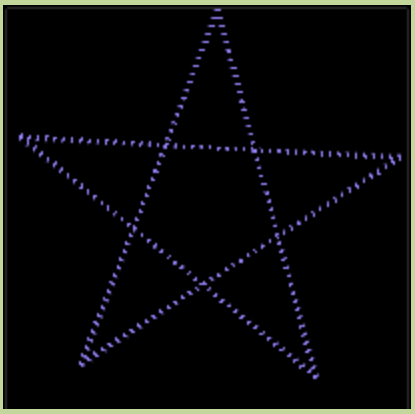


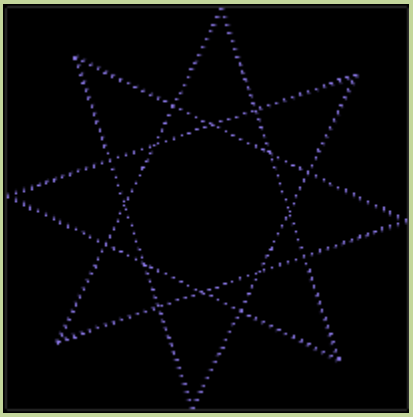


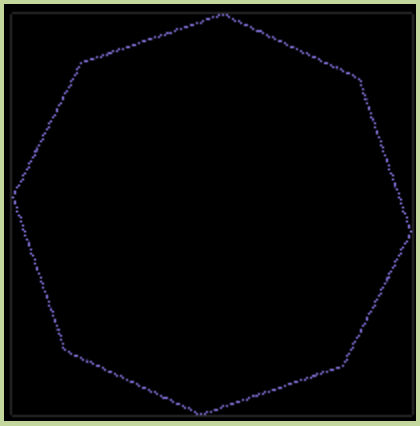
Рисуем параболу, фигуры.

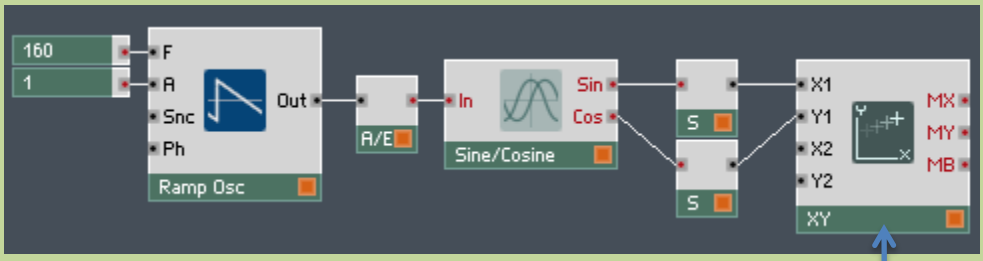




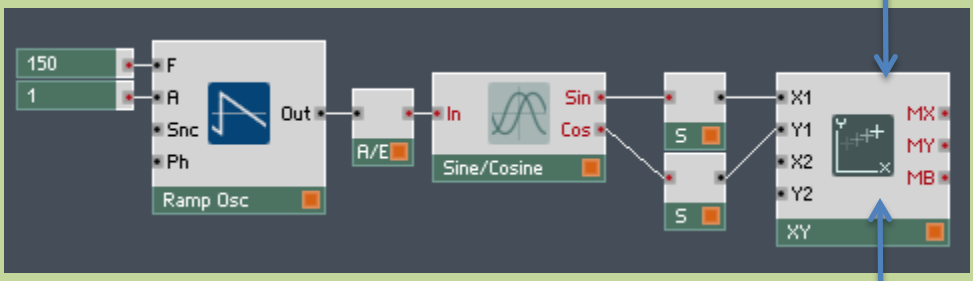


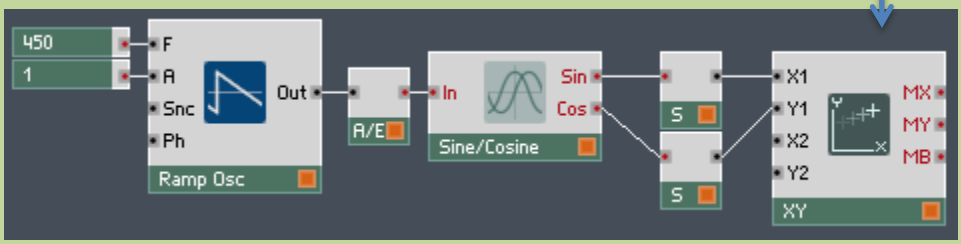




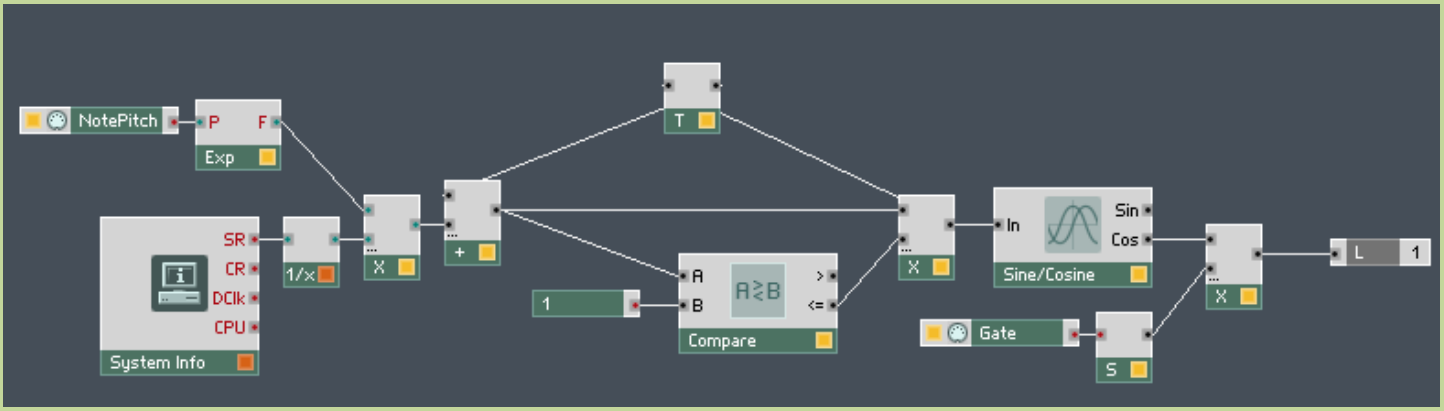


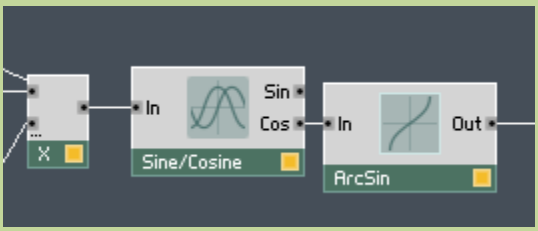
Smoother=1ms





Создаём Sine, Triangle, Saw осцилятор



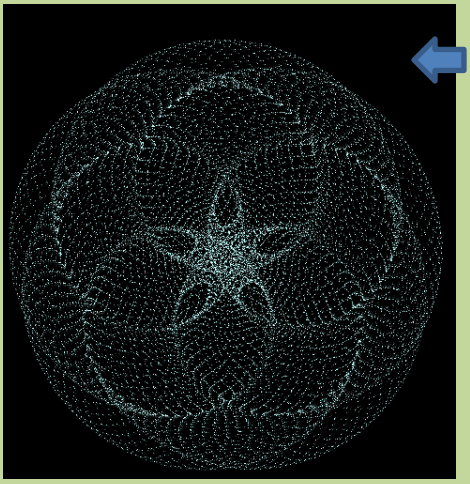


Triangle

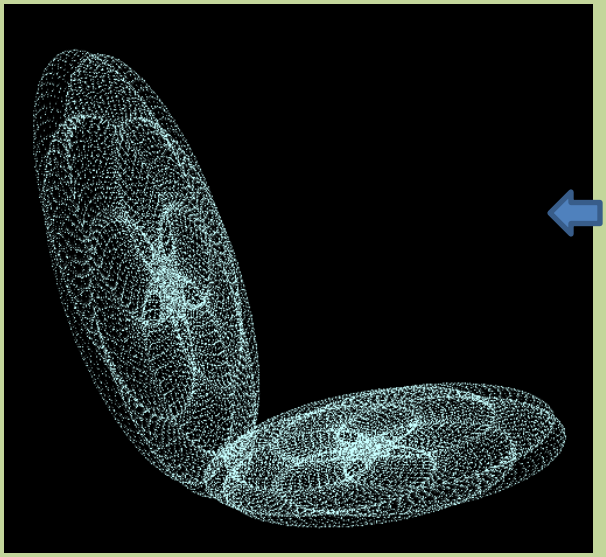
Saw

Конвертируем XY значения для модуля XY (режим Scope)

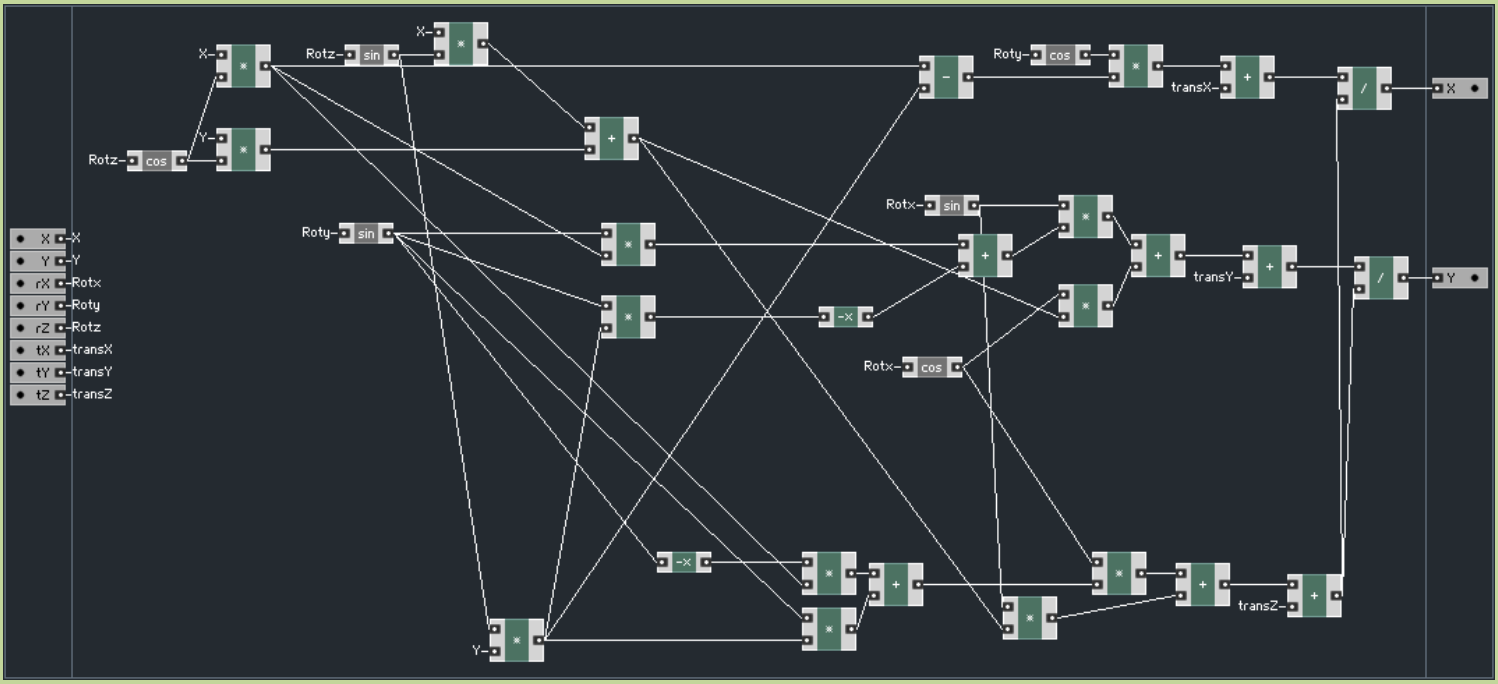
С помощью конвертации можно поворачивать, увеличивать, наклонять значения



Исходный рисунок

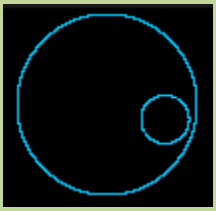


С конвертацией



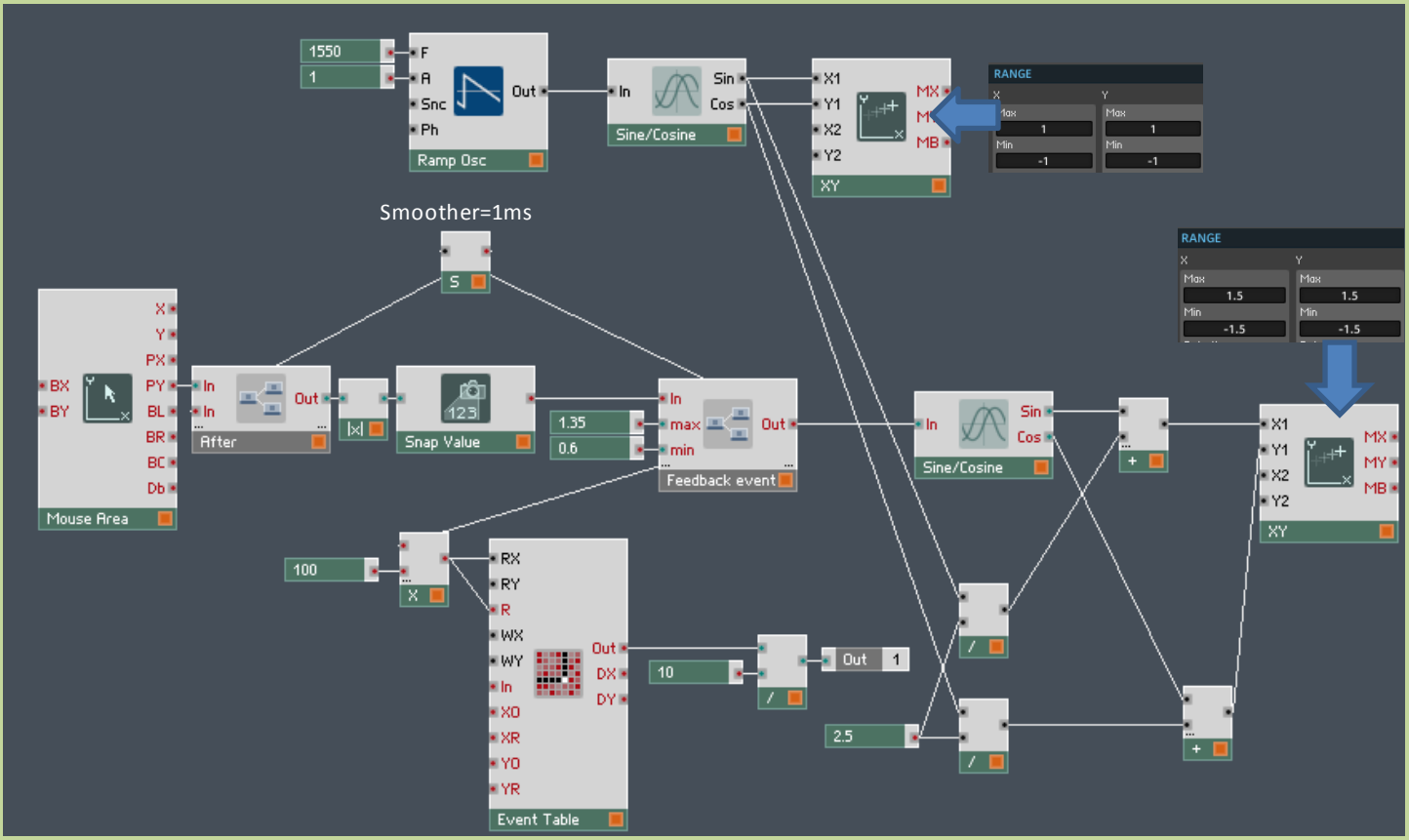
Входы rX,rY,rZ,tX,tY,tZ – регулируются knobs и лежат в пределах +3...-3

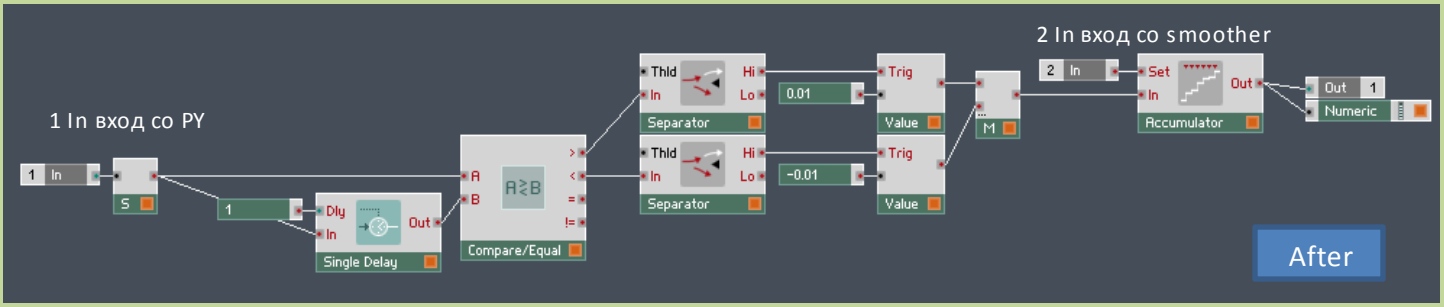
Создаём Knob средствами самого Reaktor

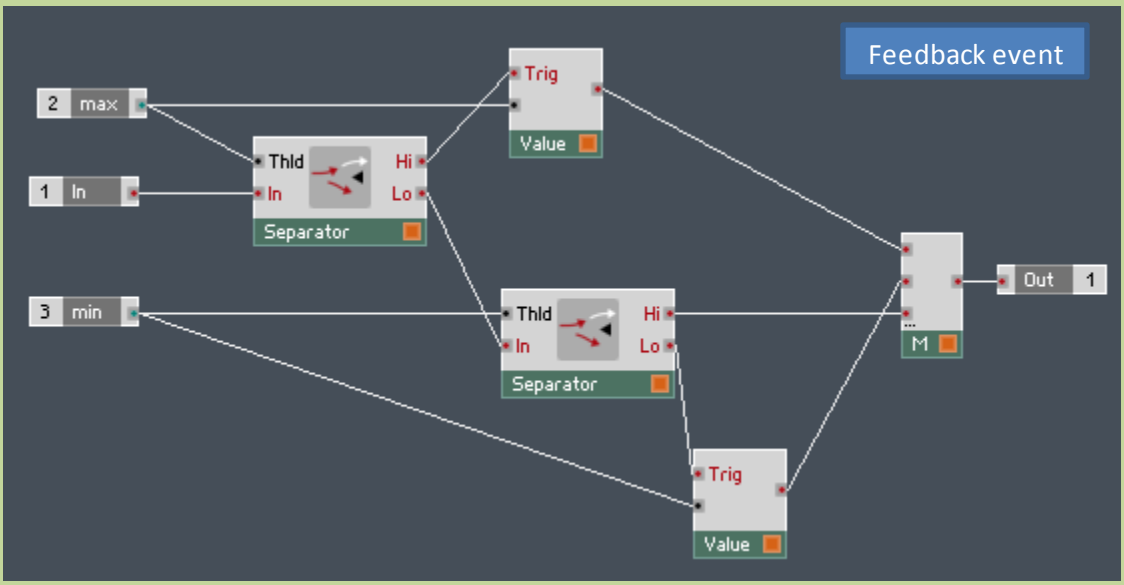


Стандартный вид кнопки

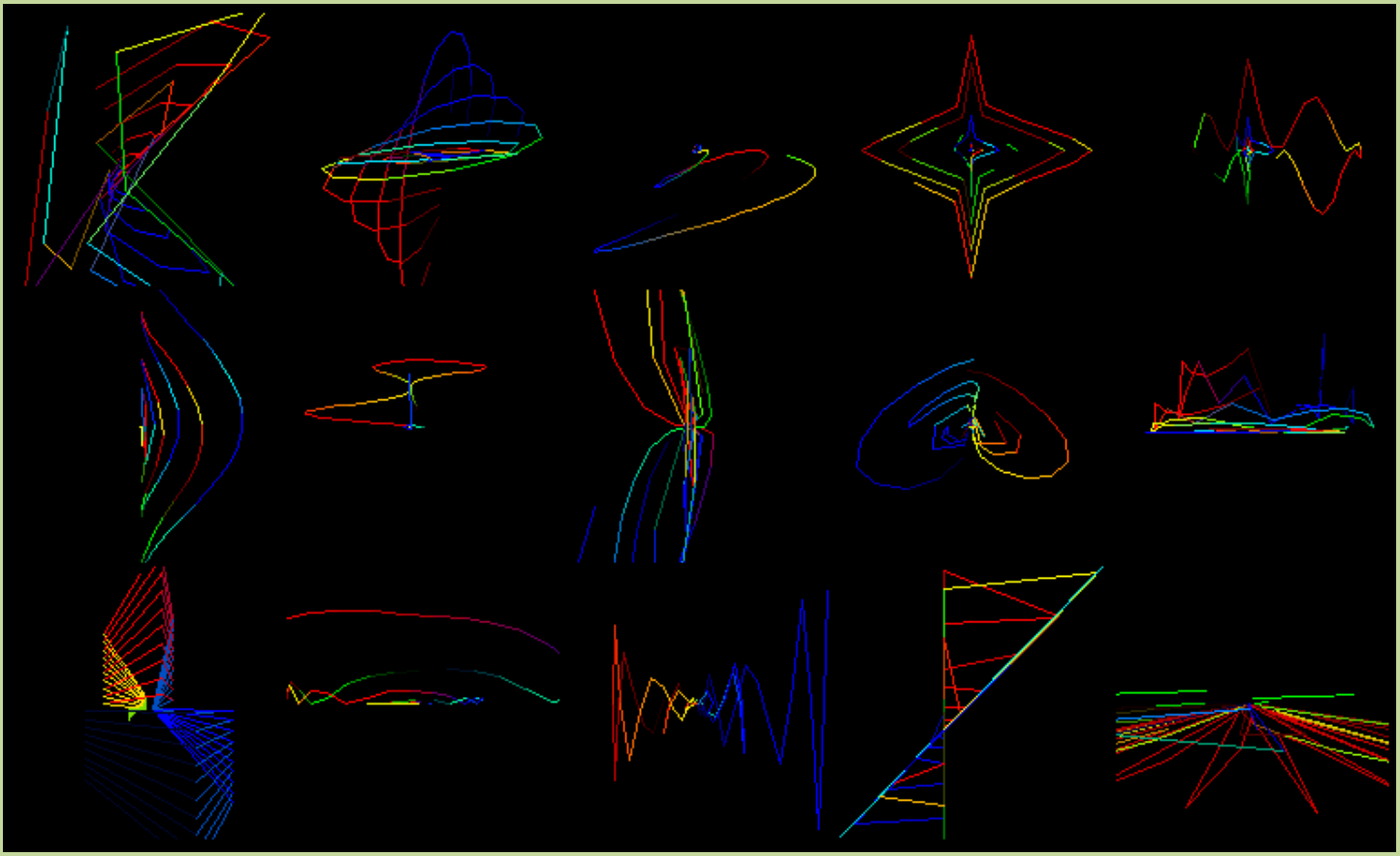
Разные виды Knob


1550
1
F
A
Snc
Ph
Out
Ramp Osc
In
Sin
Cos
Sine/Cosine
X1
Y1
X2
Y2
MX
MY
MB
XY
RANGE
X
Y
Max
Min
1
-1
Smoother=1ms
S
X
Y
PX
PY
BL
BR
BC
Db
BX
BY
Mouse Area
In
Out
After
Snap Value
123
1.35
0.6
max
min
Feedback event
RANGE
Max
Min
1.5
-1.5
100
RX
RY
R
WX
WY
XO
XR
YO
YR
DX
DY
Event Table
10
/
Out 1
2.5
+

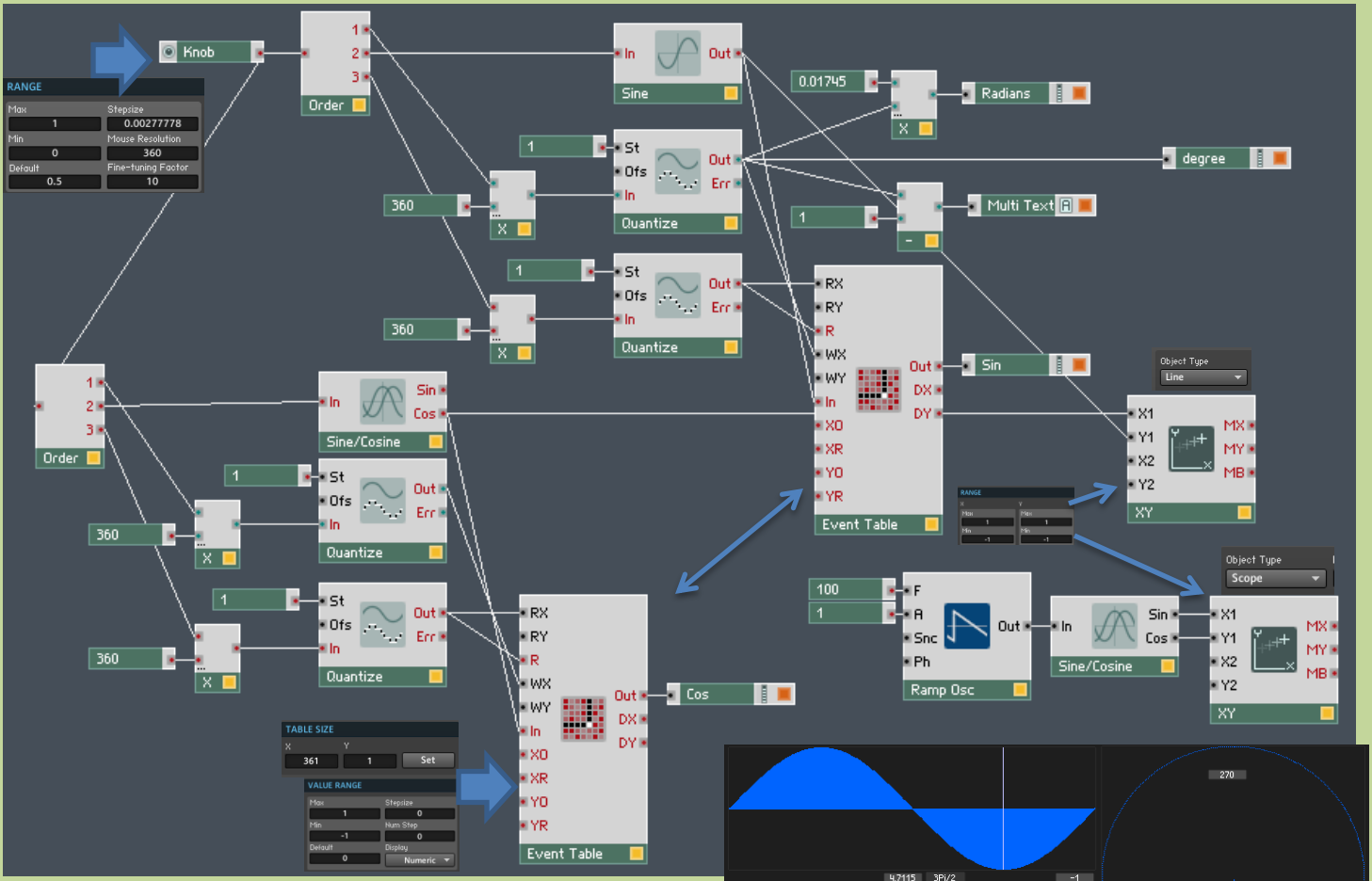

1 In вход со PY
2 In вход со smoother
1 In
S
1
Dly
In
Out
Single Delay
A
B
A≷B
Compare/Equal
Thld
In
Hi
Lo
Separator
0.01
-0.01
Trig
Value
M
2 In
Set
In
Out
Accumulator
Out 1
Numeric
After

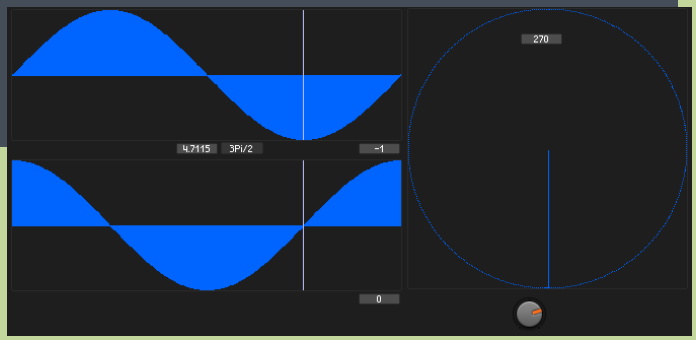

Feedback event
2 max
1 In
3 min
Thld
In
Hi
Lo
Separator
Trig
Value
M
Out 1

Необычные Knobs сделанные в самом Reaktor



Создаём визуализацию градусной меры для синуса и косинуса



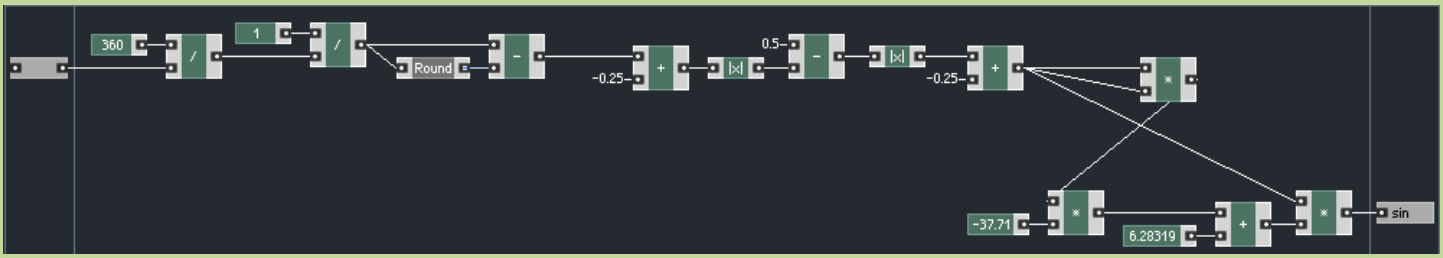


Синус 0.75=-1 (y)
Косинус 0.75=0 (x)

Как найти синус любого числа (точность до сотых)

Наберите на калькуляторе 35 (калькулятор в режиме Градусов) , а затем нажмите Sin. Ответ будет 0,57357

Как в Реакторе сделать тоже самое



1. $\zeta := \frac{1}{\frac{360}{35}}$
2. $\xi := \left|0.5 - \left|\zeta - \text{round}(\zeta) + (-0.25)\right|\right| + (-0.25)$
3. $(\xi^2\ (-37.71) + 6.28319)\ \xi = 0.576211711034$

Допустим Sin 461 = 0,9816

$\zeta := \frac{1}{\frac{360}{461}}$

$\xi := \left|0.5 - \left|\zeta - \text{round}(\zeta) + (-0.25)\right|\right| + (-0.25)$

$(\xi^2\ (-37.71) + 6.28319)\ \xi = 0.9803093236883$

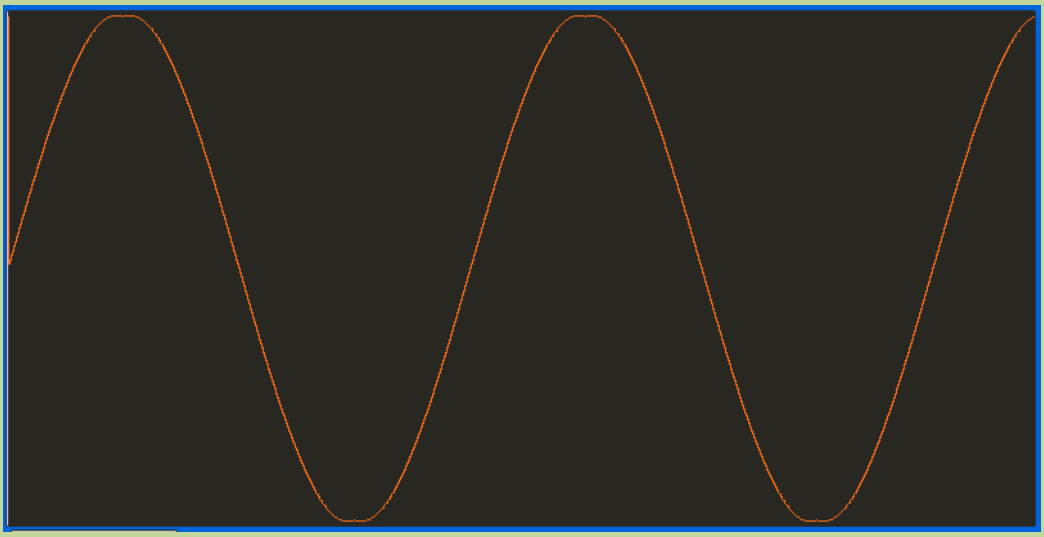
Допустим Sin 682 = -0,6156

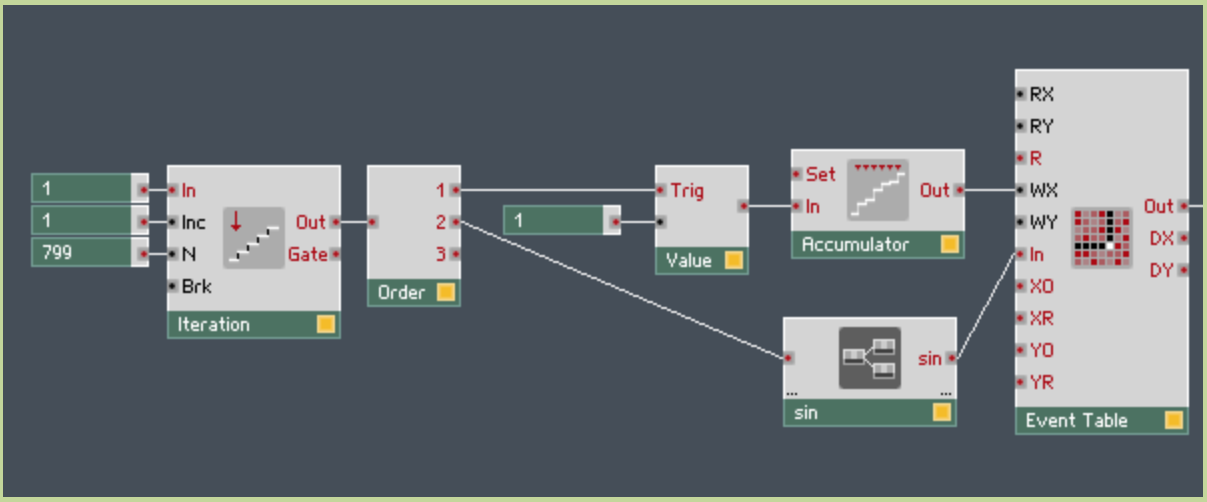
$\zeta := \frac{1}{\frac{360}{682}}$

$\xi := \left|0.5 - \left|\zeta - \text{round}(\zeta) + (-0.25)\right|\right| + (-0.25)$

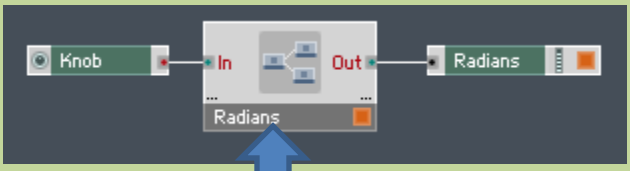
$(\xi^2\ (-37.71) + 6.28319)\ \xi = -0.6188749783951$

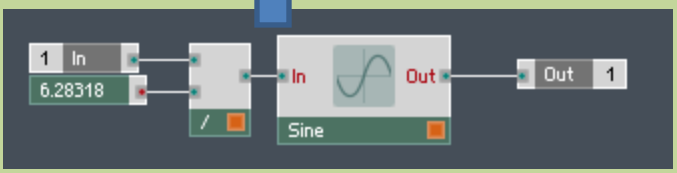
График для первых 800 чисел.





Радианы и градусы



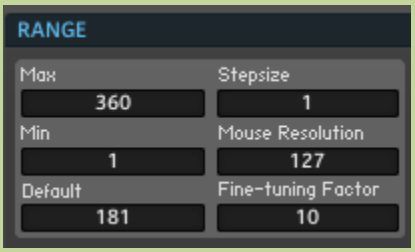


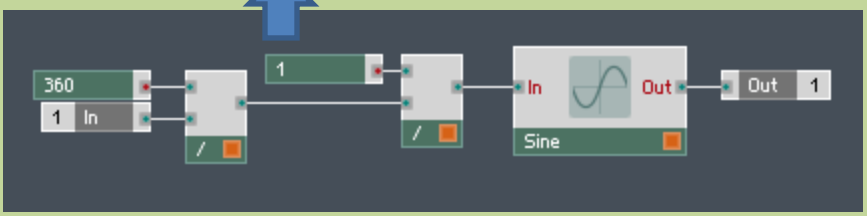
$$\sin(138)$$

$$-0.2280522595009$$

$$\sin(49)$$

$$-0.9537526527595$$

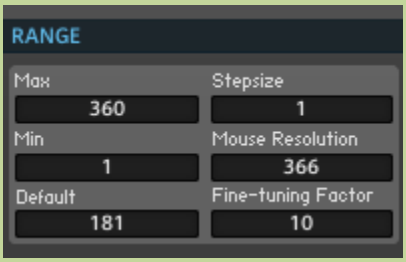




$$\sin(45)$$

$$\frac{\sqrt{2}}{2}$$

$$0.7071067811865$$

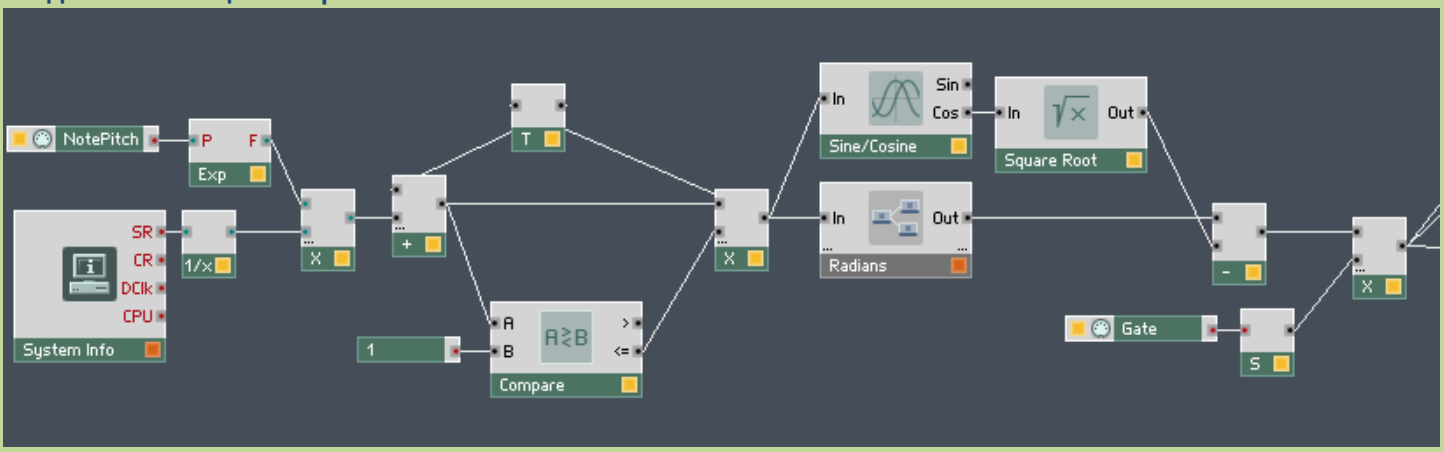


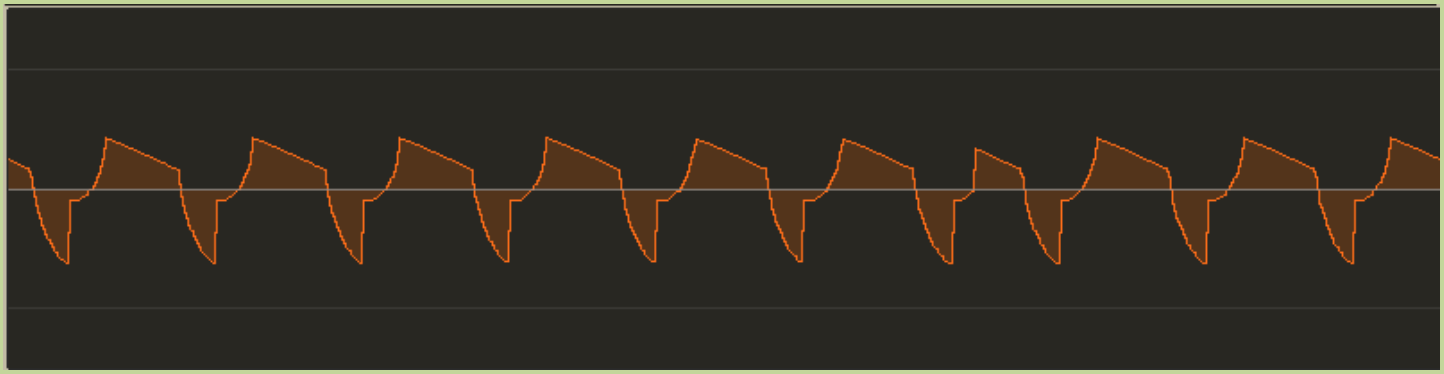
$$\sin(201)$$

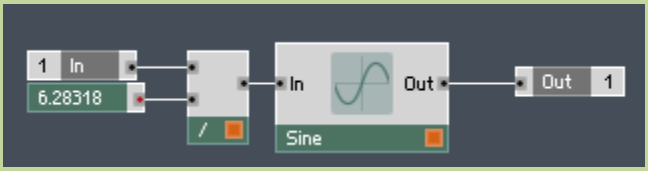
$$-\sin(21)$$

$$-0.3583679495453$$

Создаём свой осциллятор



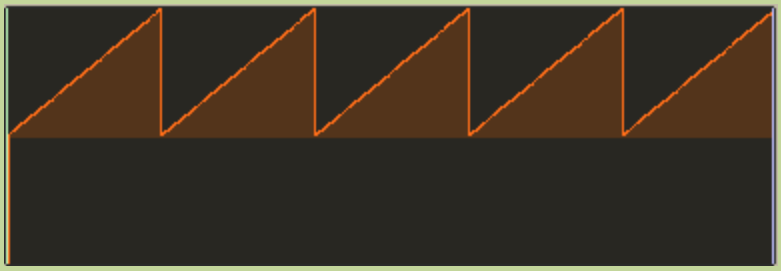




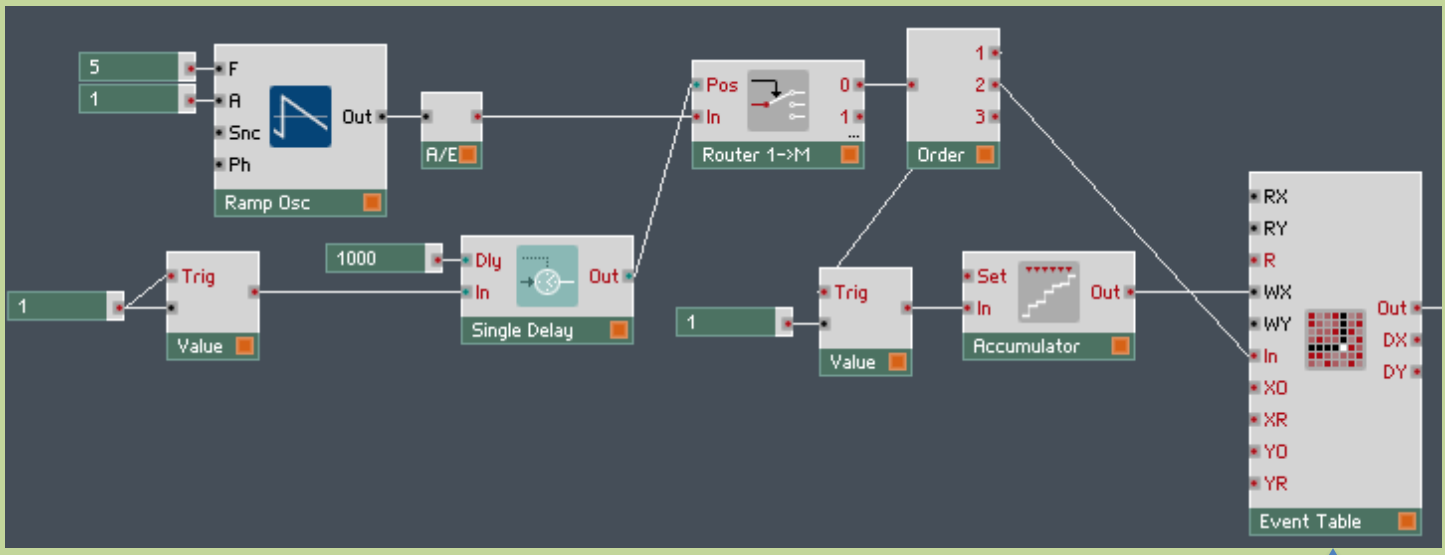
Макрос Radians

Ramp 5Hz

Нажмём 2 раза  и получим на Event Table

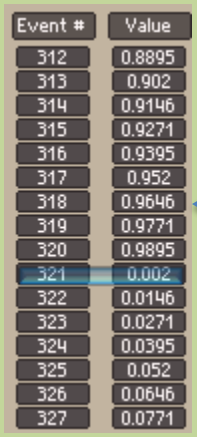


Как видно Ramp 5 Гц это пять полных периодов от 0 до 1 проходящих за 1 секунду.

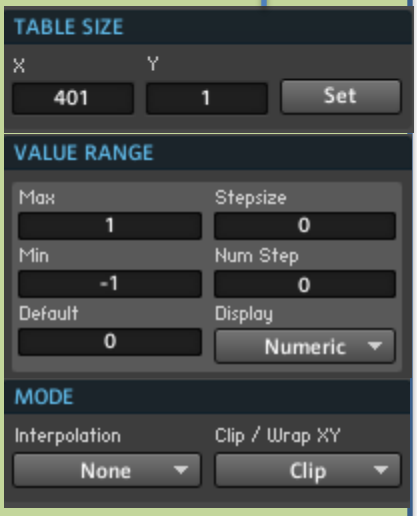


Разделим 400/5=80.Через каждые 80 events будет заканчиваться период.

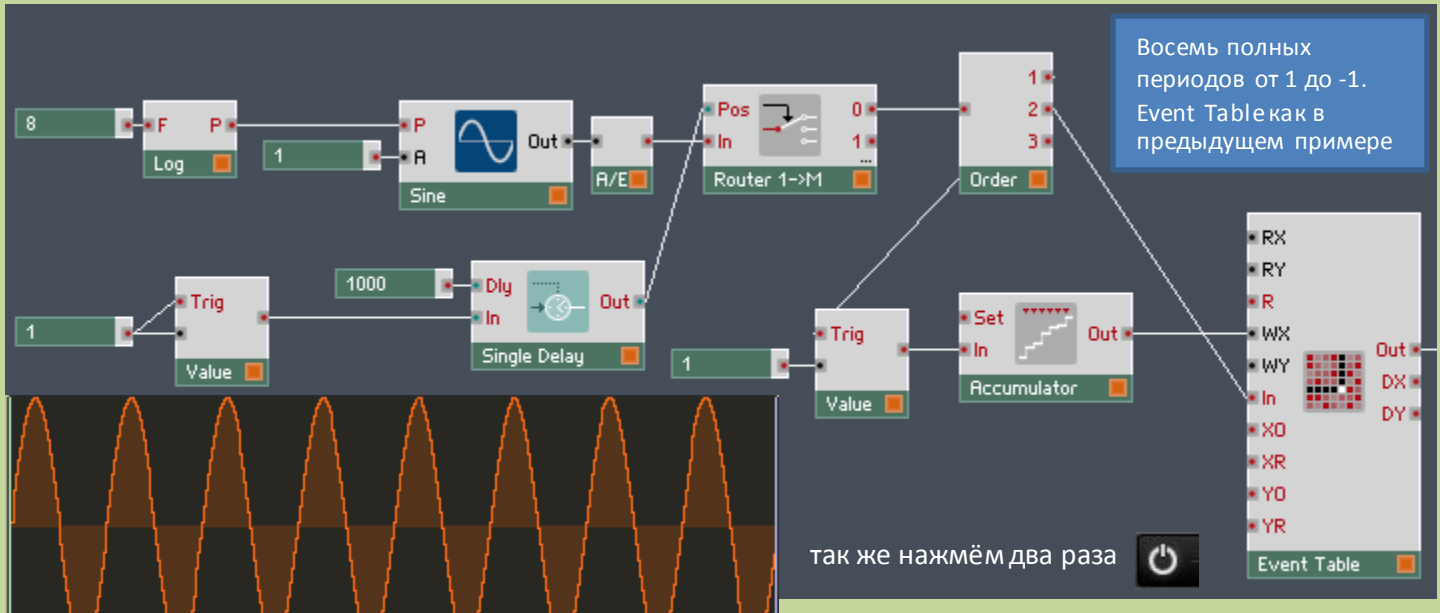
Синим выделено – начало 3 периода. 3*80+1=241



Синим выделено – начало 4 периода. 4*80+1=321



Sine 8Hz



Восемь полных периодов от 1 до -1. Event Table как в предыдущем примере

так же нажмём два раза

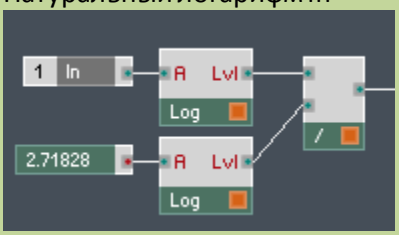
Разделим 400/8=50.Через каждые 50 events будет заканчиваться период.

Event #	Value
243	-0.8309
244	-0.7541
245	-0.666
246	-0.5676
247	-0.4602
248	-0.3445
249	-0.2244
250	-0.1008
251	0.0243
252	0.1503
253	0.2727
254	0.3909
255	0.5029
256	0.6079
257	0.7025
258	0.786

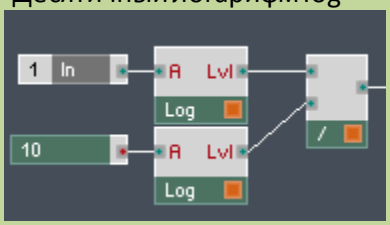
Синим выделено – начало 5 периода. 5*50+1=251

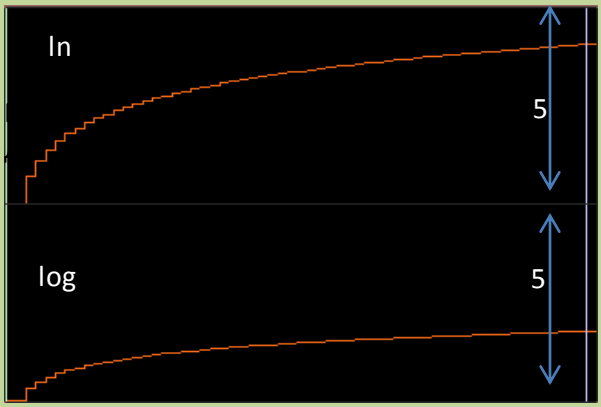
Логарифмы

Натуральный логарифм ln



Десятичный логарифм log

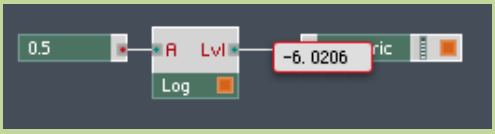




Графики логарифмов при подачи натуральных целых чисел от 1 до 60. При Входе=60, ln=4.094, а log=1.778

Как видно логарифмы преобразуют большие значения в меньшие

Частота 440Гц с помощью специального логарифма преобразуется в значение ноты 69 (Ля первой октавы)



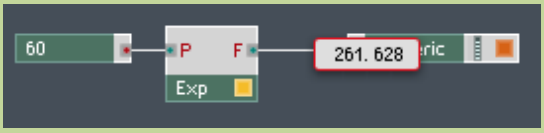
Амплитуда 0.5 с помощью специального логарифма преобразуется в – 6.0206 dB

Экспонента

Экспонента преобразует меньшие значения в большие .Экспонента есть показательная функция y=a^x

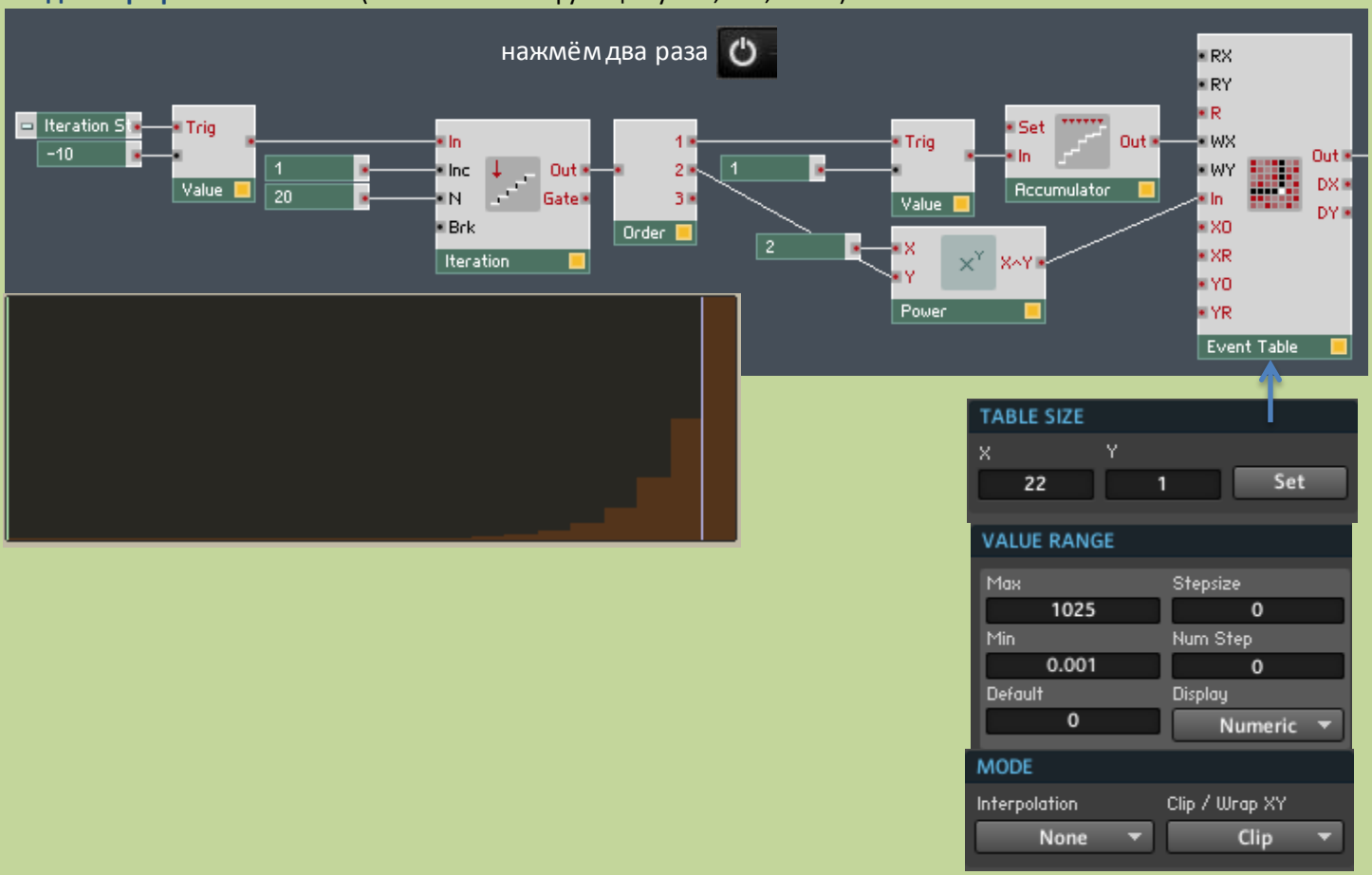
-1.5 dB с помощью экспоненты преобразуется в 0.841 амплитуды

Важное замечание : При конвертировании Lvl в A, значение Lvl должно варьироваться от -50 до 1.1 dB, так как если подать lvl=1.2, то A будет = 1.1482, что есть нарушение функции экспоненты (выход стал меньше чем вход), это приводит к завышению громкости звука, но иногда это необходимо.

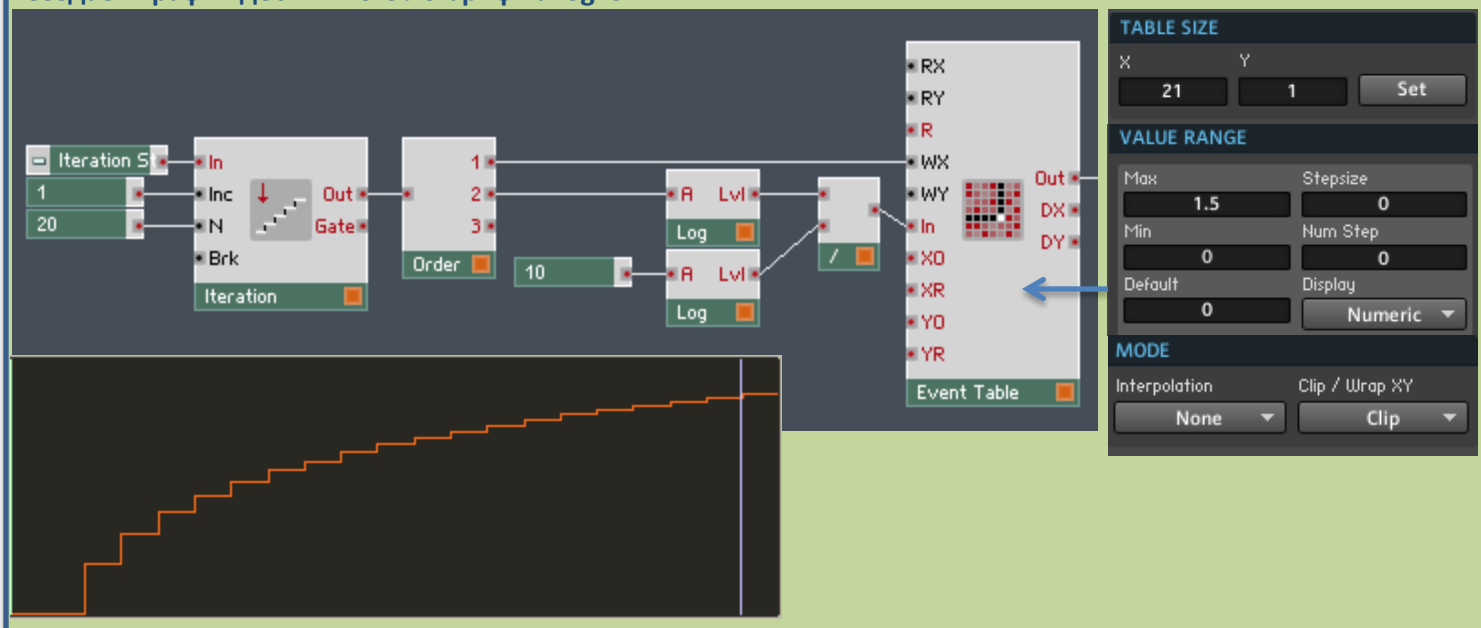


Значение ноты =60 с помощью экспоненты преобразуется в частоту 261.628Гц

Создаём график экспоненты (показательная функция y=a^x, a=2, x∈ N)

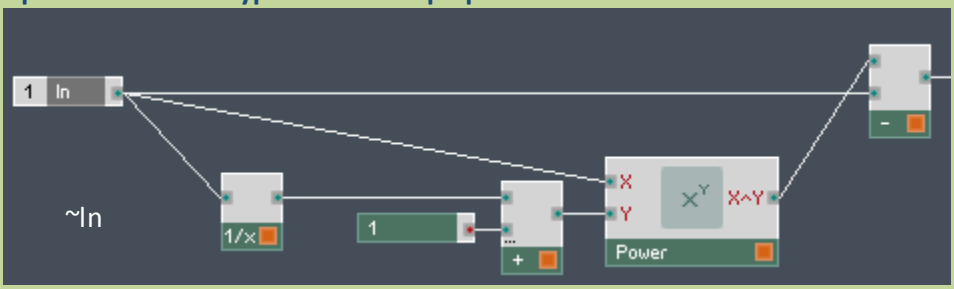


Создаём график десятичного логарифма log20



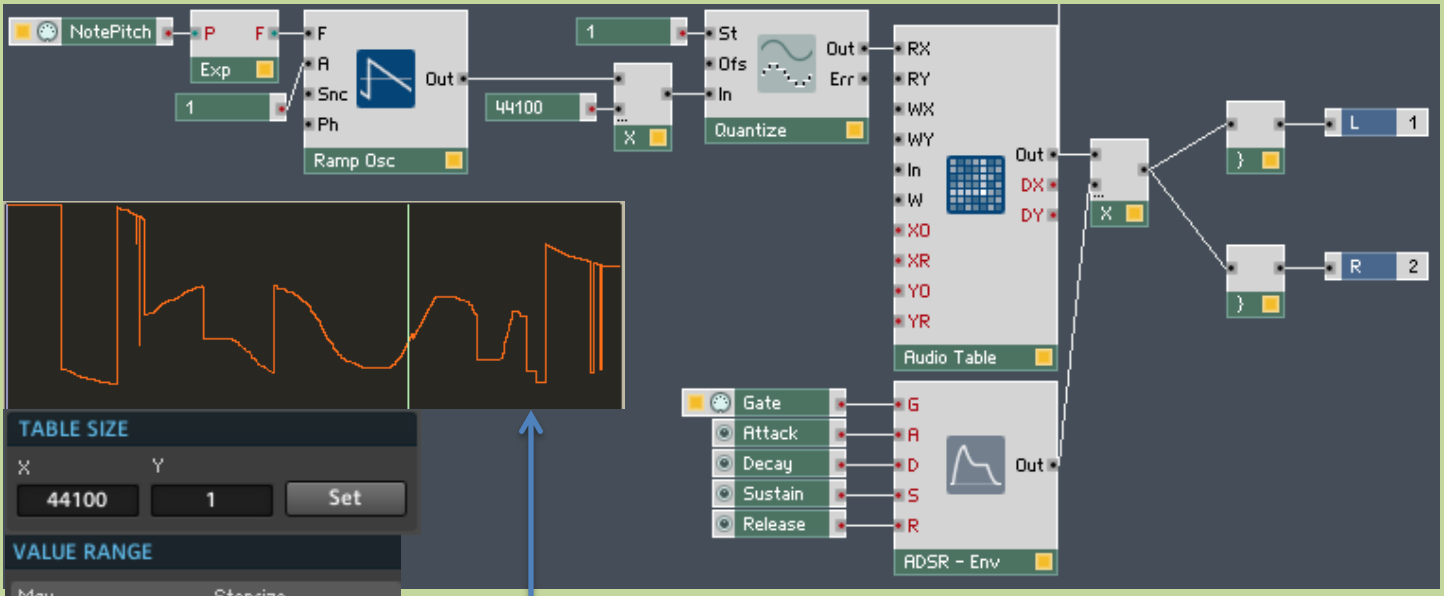
log20=1,301029

Приближёный натуральный логарифм



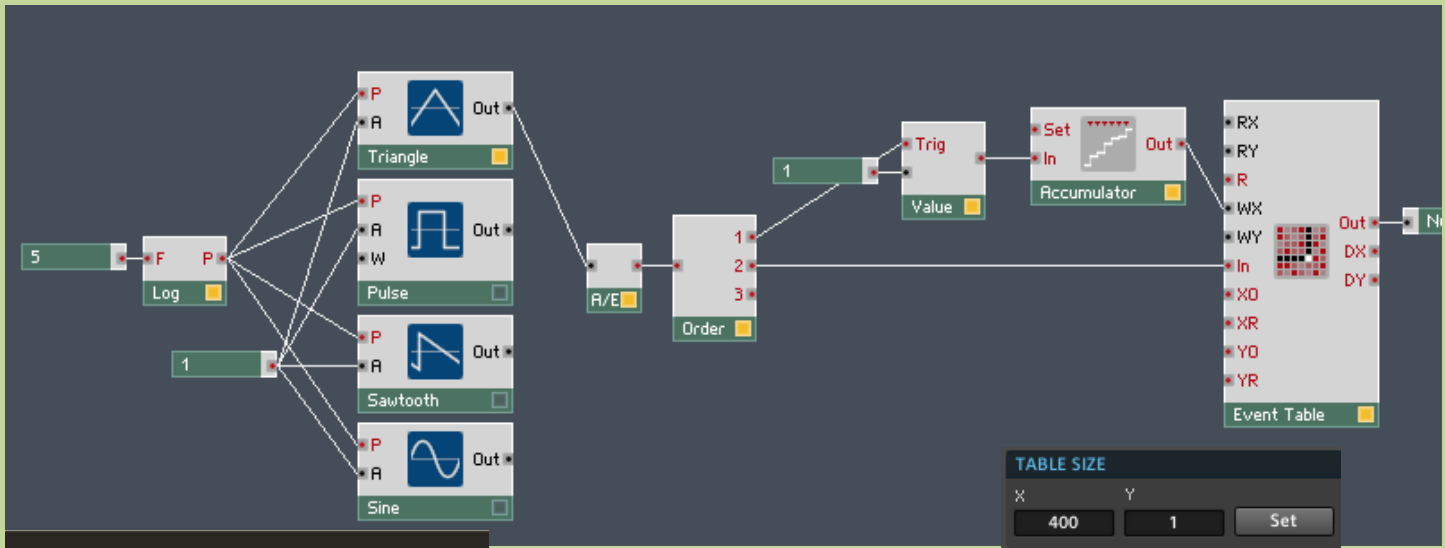
ln33=3.4965 ~ln33=3.6885
ln450=6.1093 ~ln450=6.1508
ln1283=7.157 ~ln1283=7.176
ln3083=8.033 ~ln3083=8.044
ln5000=8.517 ~ln5000=8.525

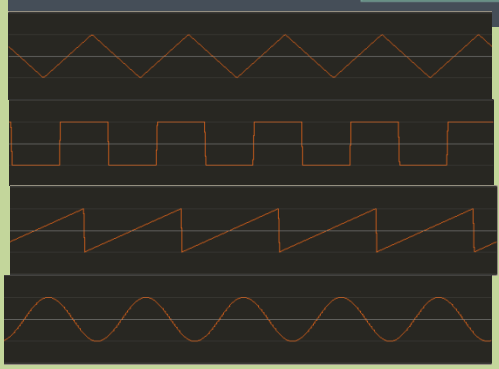
Рисуем, считываем, проигрываем значения с Audio Table с помощью Ramp



Чтобы рисовать нужно нажать правой клавишей мыши по таблице и выбрать Table Draw Mode

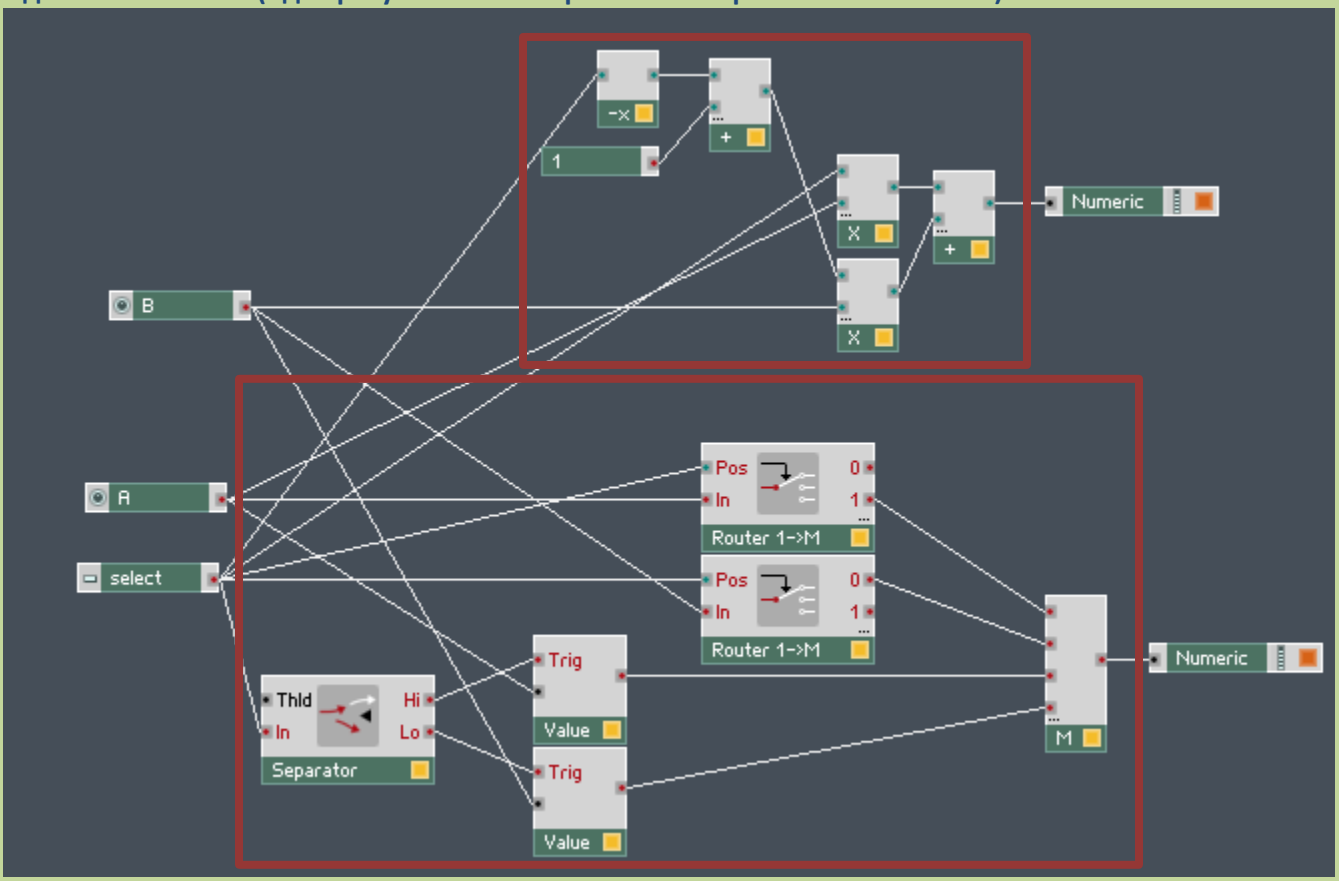
Event Table как Scope





Wrap это то же самое что modulo (где b=400), значения идут по кругу, Interpolation это сглаживание

Идентичность схем (один результат можно реализовать разными способами)



Записываем и считываем сигнал синусойды в Audio Table.

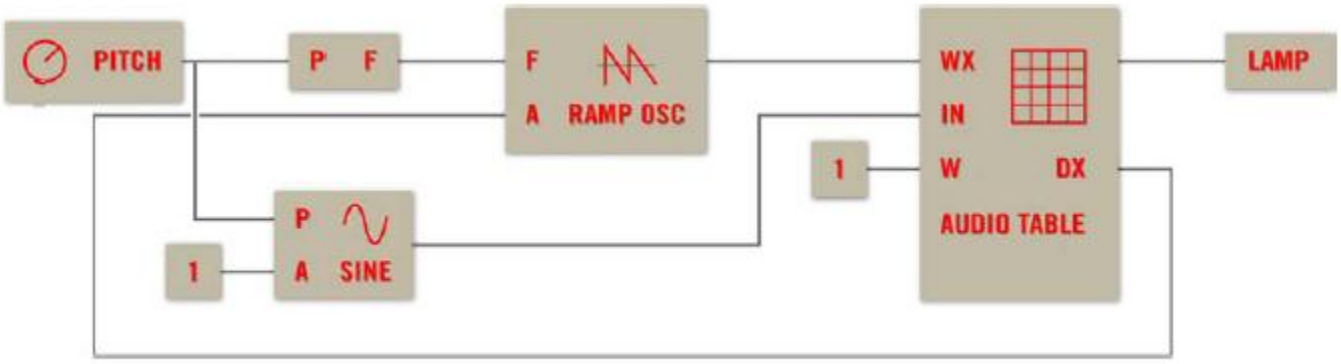
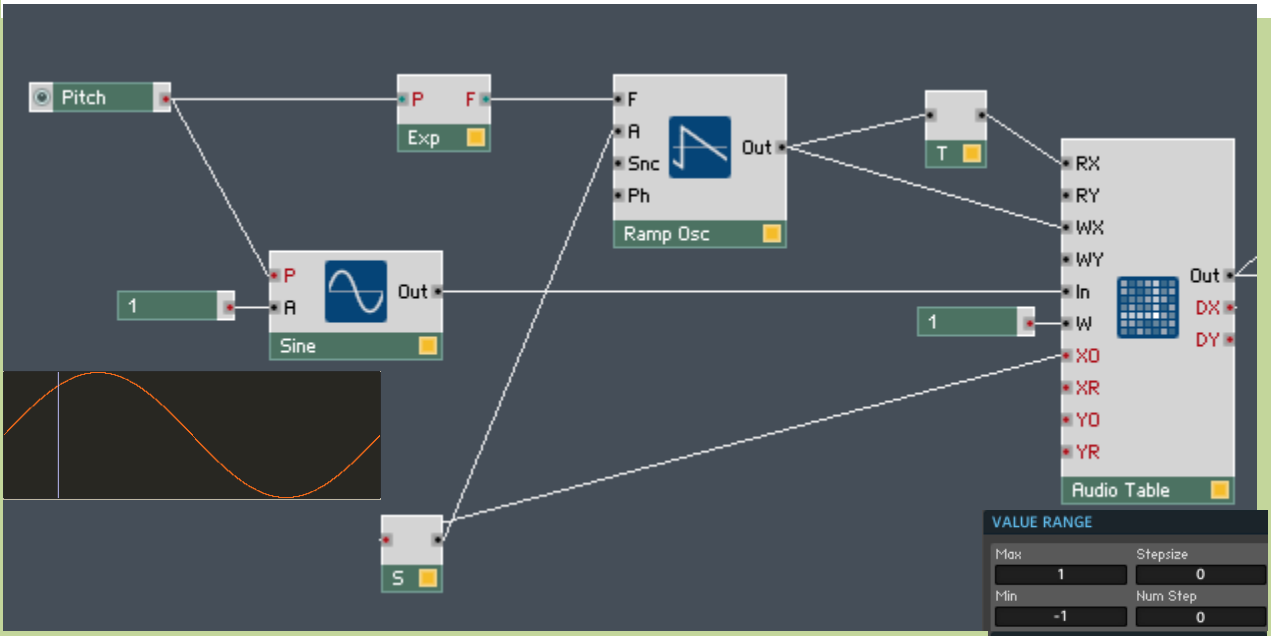
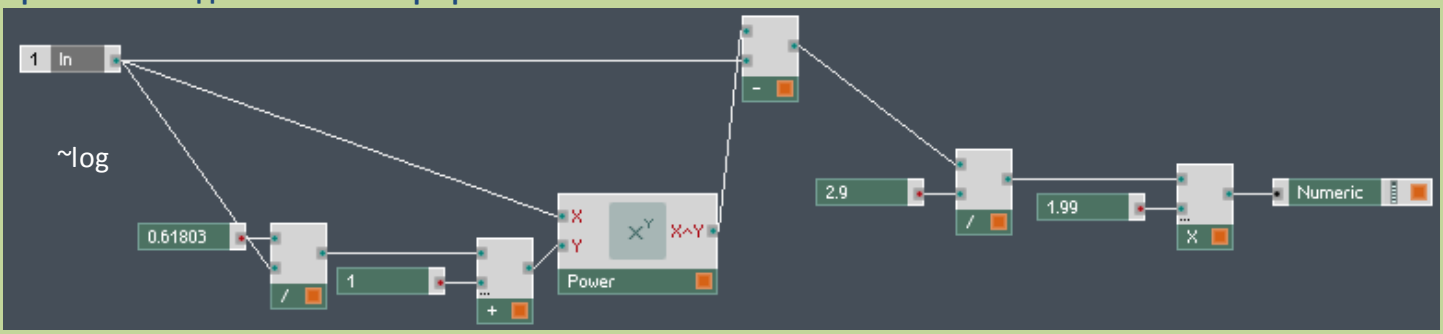


Fig. 11.3 A schematic of the Structure that uses an Audio Table Module to record a signal from a Sine Module.



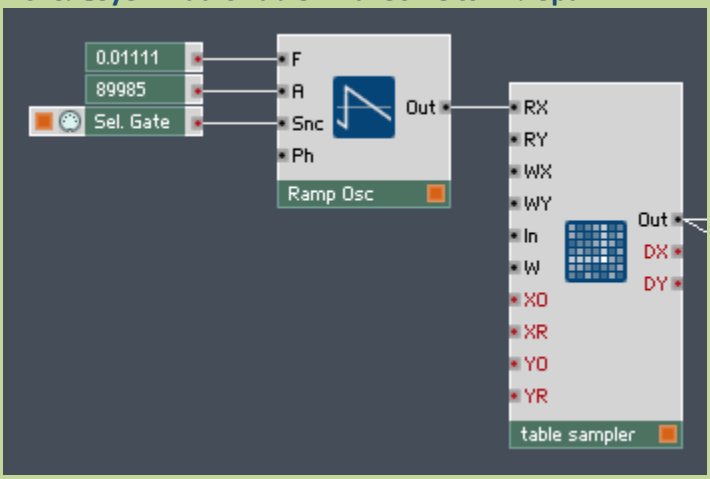
Смуфер=1мс, если W=0, то будет происходить только считывание.

Приблежёный десятичный логарифм

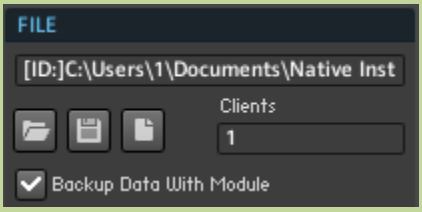


log33=1,5185 ~log33=1,5325
log208=2,318 ~log208=2,281
log983=2,992 ~log983=2,928
log5000=3,698 ~log5000=3,614

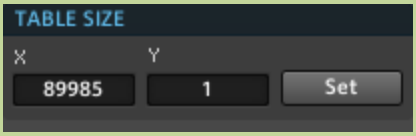
Используем Audio Table в качестве сэмплера.



Чтобы добавить аудио-файл выберите



Когда вы добавите файл то увидите что значения Table Size изменились, используйте это значение для входа A



Чтобы расчитать F разделите 1000/Table Size. 1000/89985=0.01111

Формула для логарифмов с любой базой. (точность до сотых, тысячных)

$$\log_b t = \frac{\ln t}{\ln b}$$

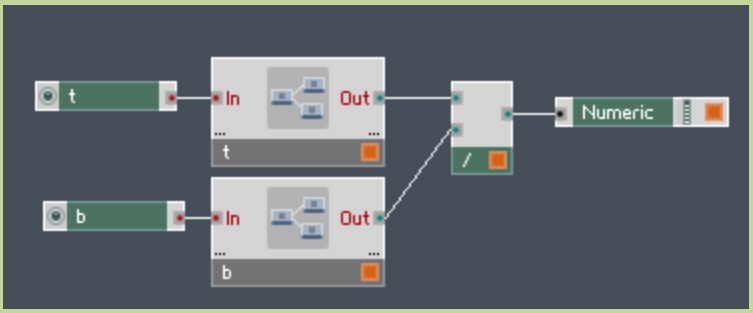
$$\ln = n10^7 \cdot n^{\frac{1}{n10^7}} - n10^7$$

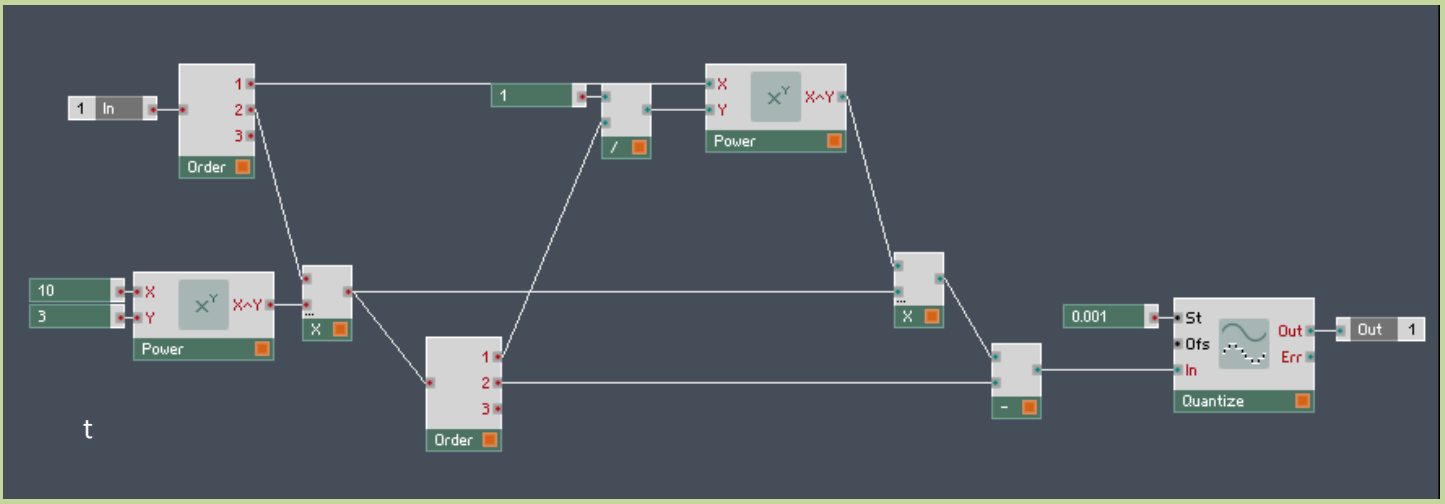
$\log_2 47$

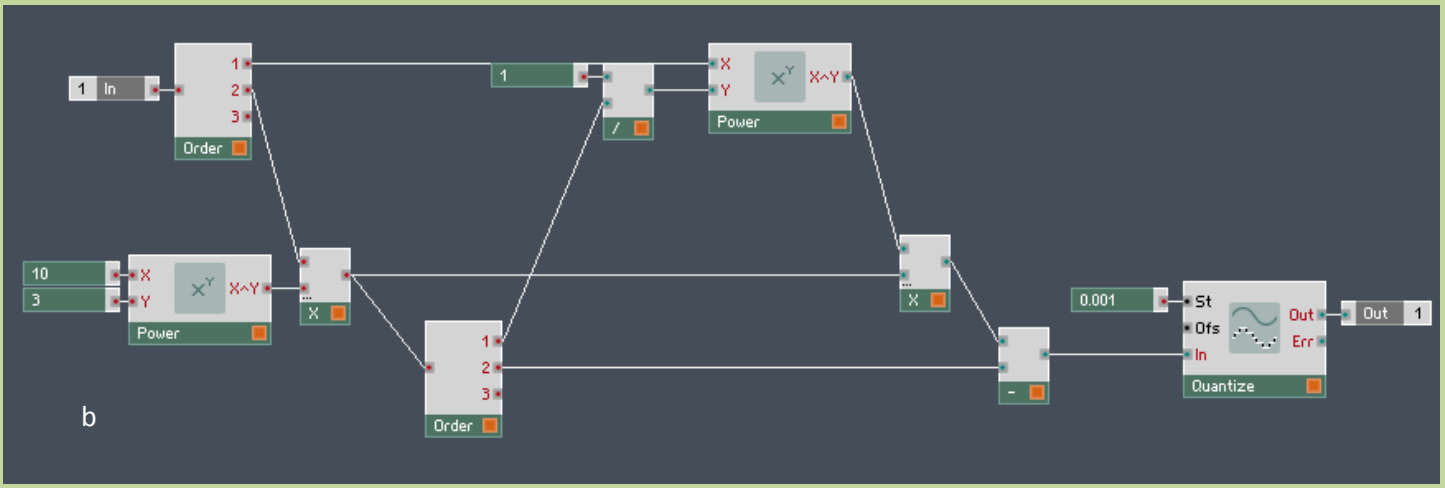
5.5545888516776

$\Rightarrow \frac{\ln(47)}{\ln(2)} \Rightarrow \ln = n10^7 \cdot n^{\frac{1}{n10^7}} - n10^7 \Rightarrow \frac{47 \cdot 10^7 \cdot 47^{\frac{1}{47 \cdot 10^7}} - 47 \cdot 10^7}{2 \cdot 10^7 \cdot 2^{\frac{1}{2 \cdot 10^7}} - 2 \cdot 10^7}$

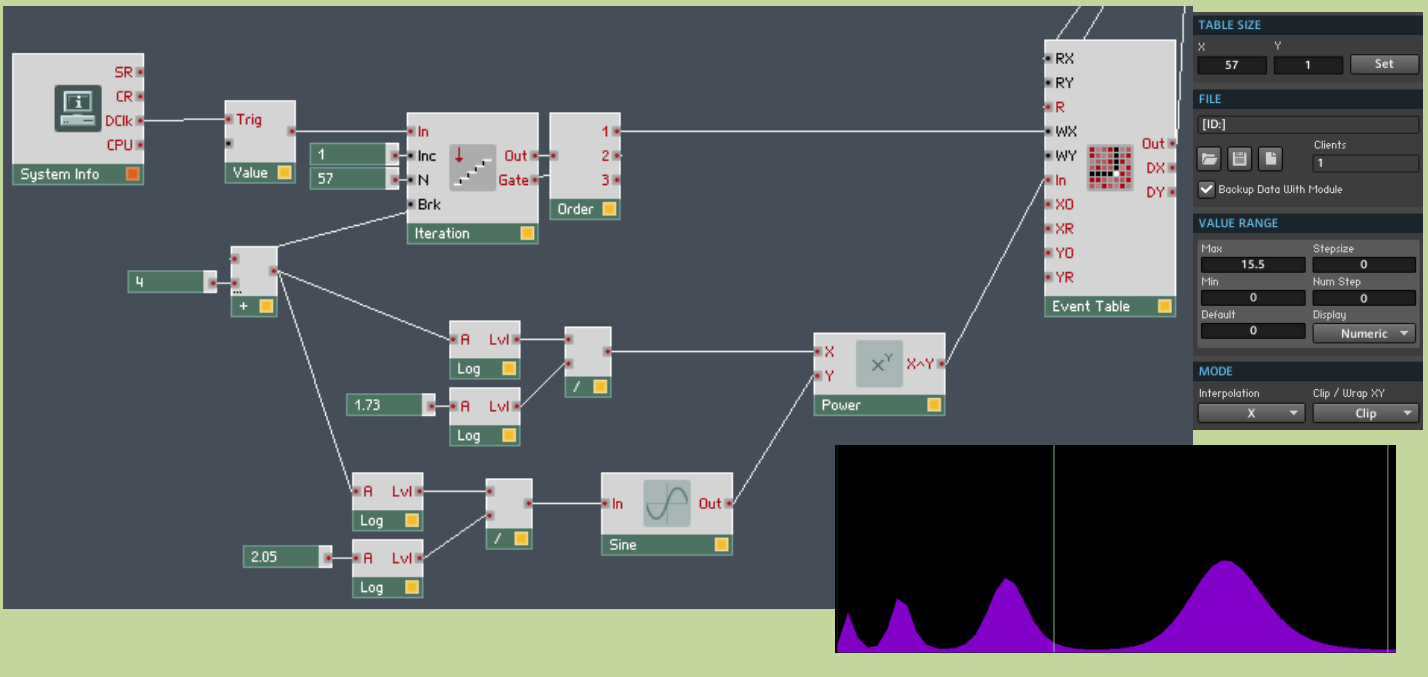
$\Rightarrow$ 5.554588751519





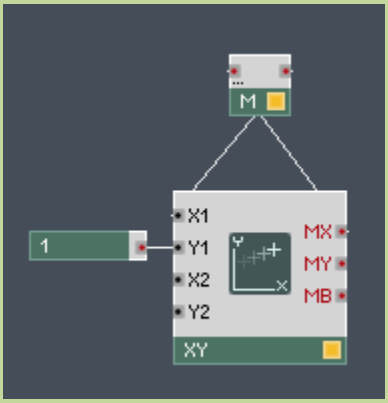


Логарифмический синус



Создаём биполярную горизонтальную крутилку – knob

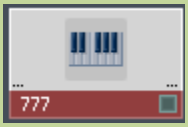
Биполярная означает, что крутилку можно крутить от 0 до +-n. То есть её можно крутить в обе стороны.

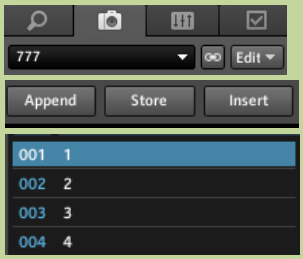


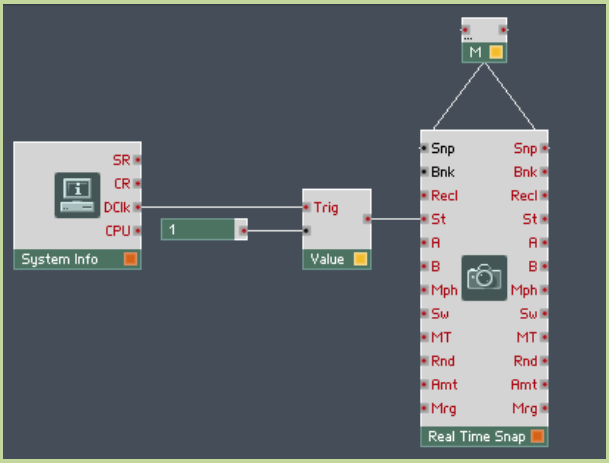
Эта крутилка лучше чем обычная, так как помимо кручения можно прыгать по значениям (нажимать в разных местах, в то время когда обычную крутилку можно только крутить)

Создаём Real Time Snap – данная схема в реалтайме записывает все изменения в пресетах.

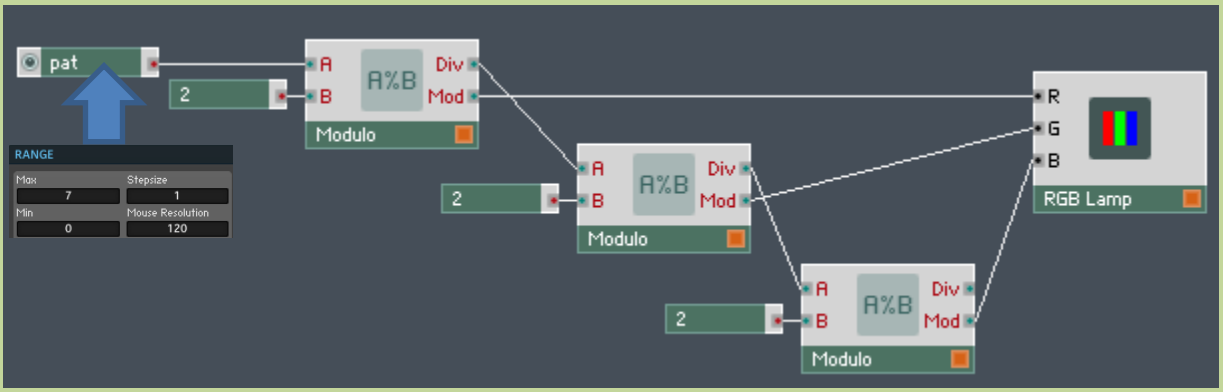
Для начала надо создать инструмент, далее создать несколько пустых пресетов. Затем добавить следущую схему. В дальнейшем будет невозможно добавить пресеты нажимая Append, поэтому до добавления схемы создайте столько пресетов сколько вам необходимо.

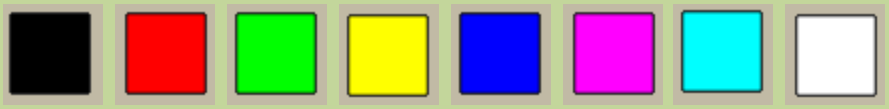




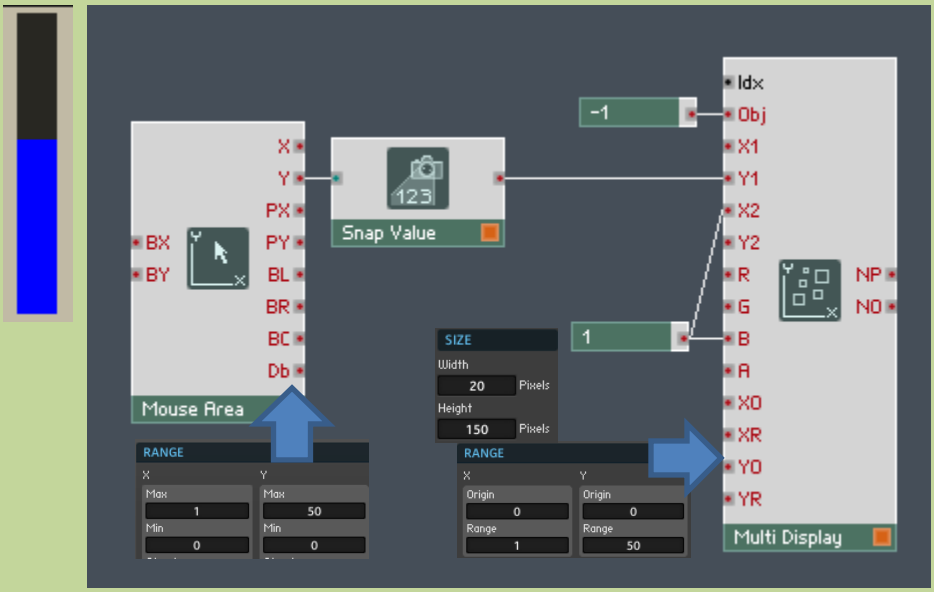


Одна крутилка – 8 цветов, RGB Lamp.



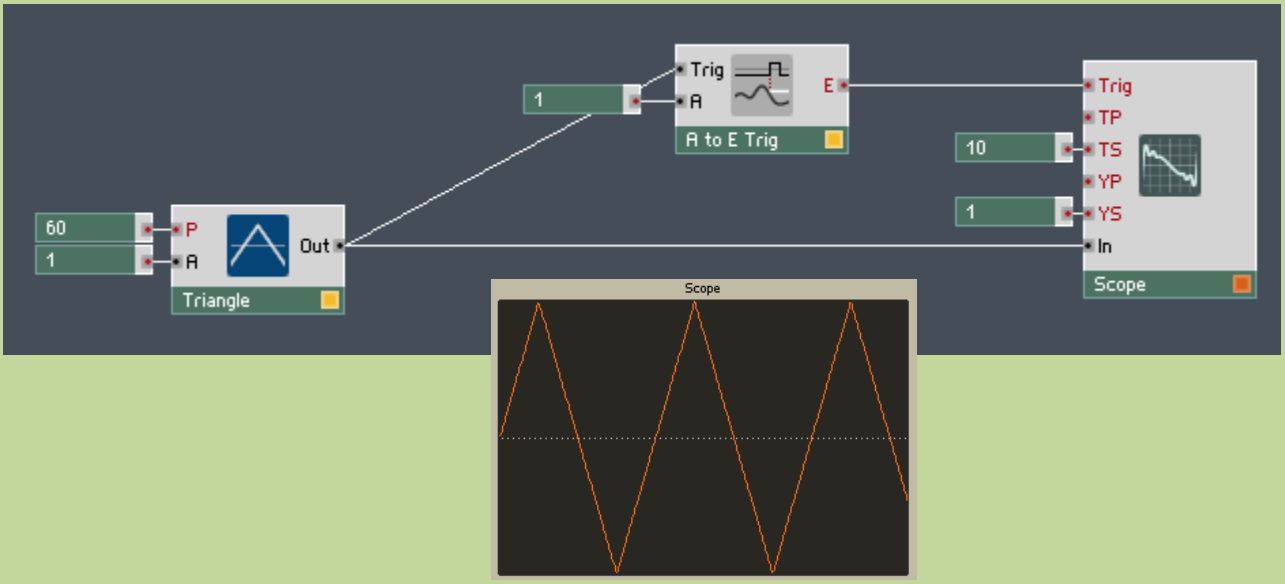


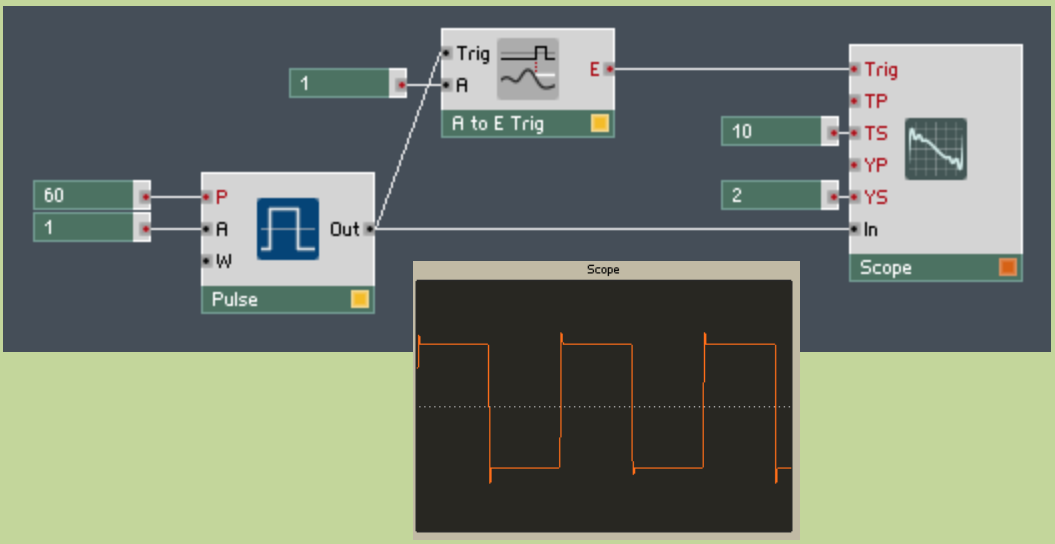
Создаём вертикальную крутилку с помощью Mouse Area и Multi Display



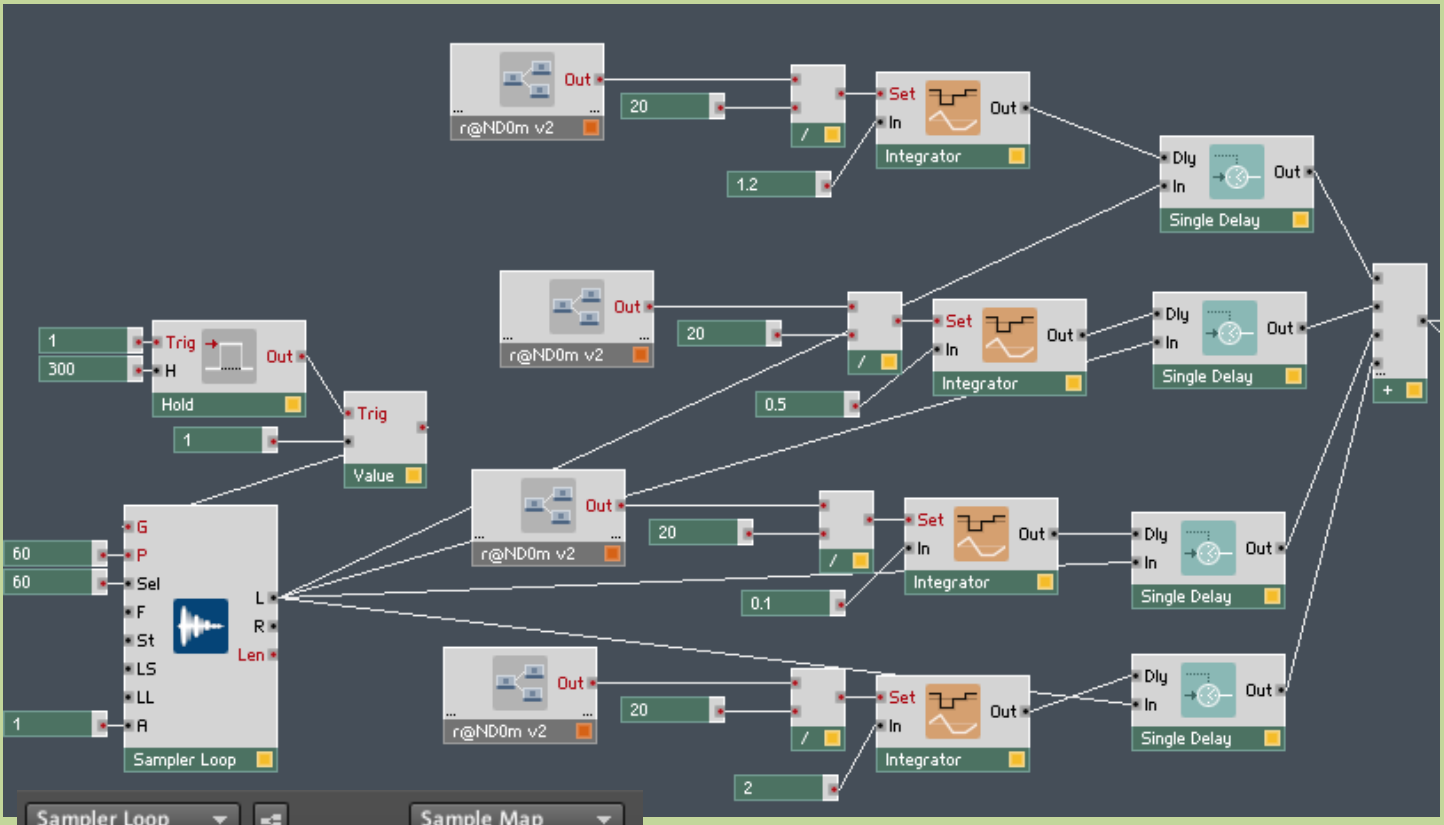
Модули имеют одинаковый размер по Width, Height и накладываются друг на друга на панели. Эта крутилка лучше чем обычная, так как помимо кручения можно прыгать по значениям (нажимать в разных местах, в то время когда обычную крутилку можно только крутить). Если Snap Value не будет в схеме , то при открытии файла значение с Y будет 0.

Используем Scope

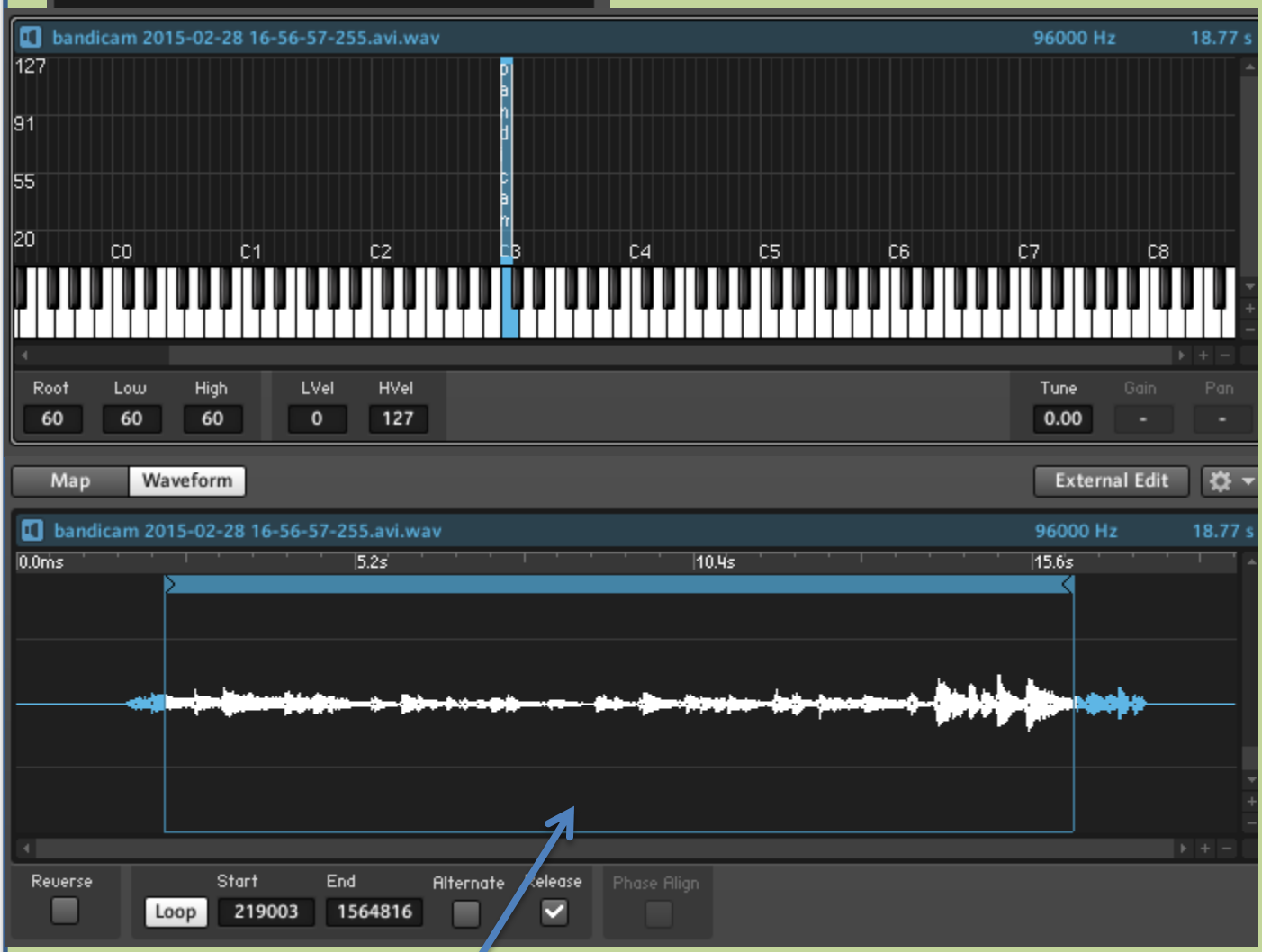




Sampler Loop Glitch

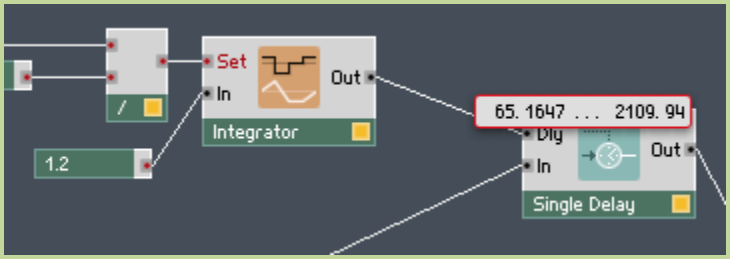


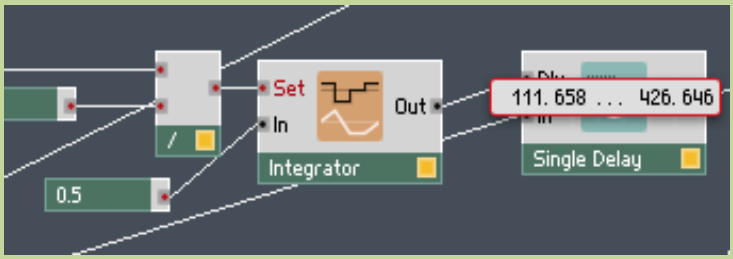
Посыл на G - для активации при открытии ensemble



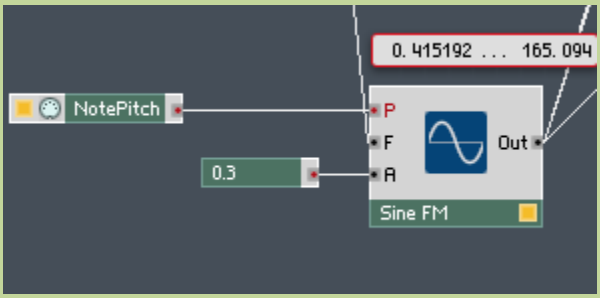
Правой клавишей мыши рисуем рамки для Loop

Можно использовать верхний вход Dly у Single Delay следующим способом, причем эффект будет мини-буфер : так как Integrator накапливает значения, то эти значения-мини-буфер можно подать на Dly, причем сигнал будет иметь вид : например от 100.321 .... до 2032.547 ms в миллисекундах, если пришёл новый сигнал на Set Integrator то начнётся новое накопление

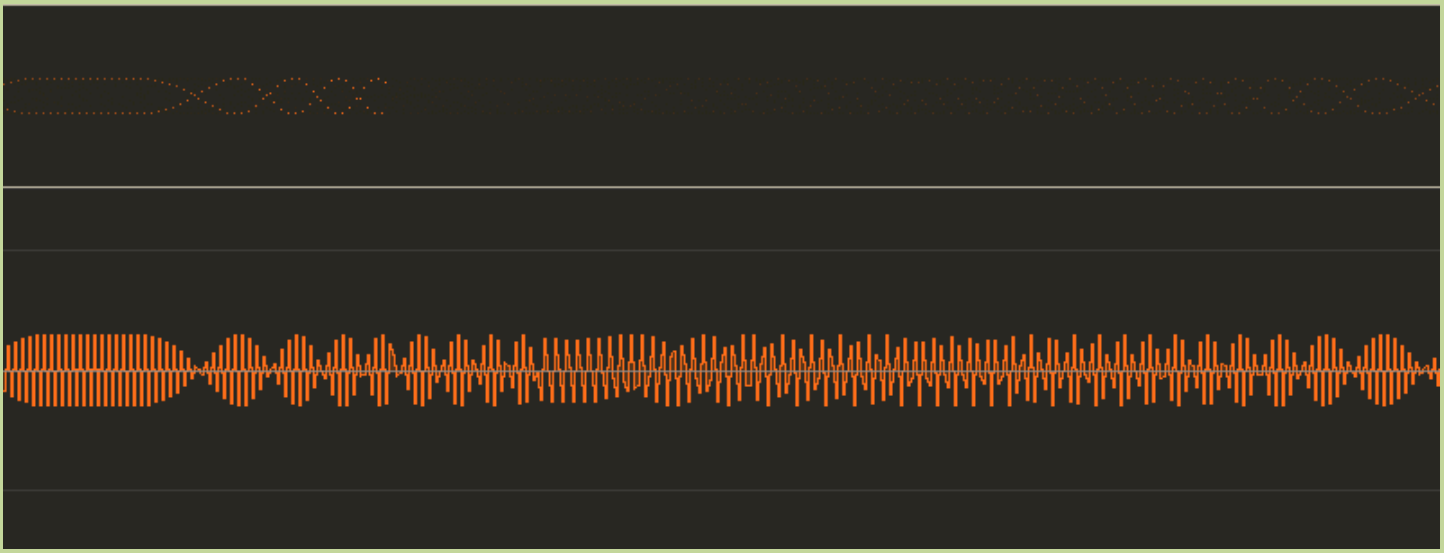




Integrator и F

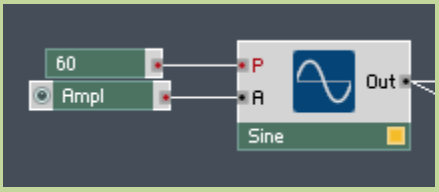


На F приходит значение с Integrator, при этом волна будет иметь вид разгона и замедления частоты



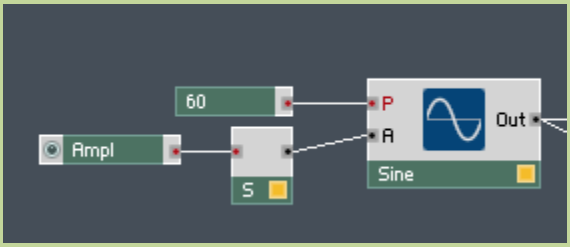
Audio, Event Smoother

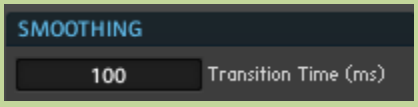
Создадим такую схему :



Если крутить Knob-Amp , то можно услышать потрескивания, чтобы это избежать - можно использовать :

Audio Smoother





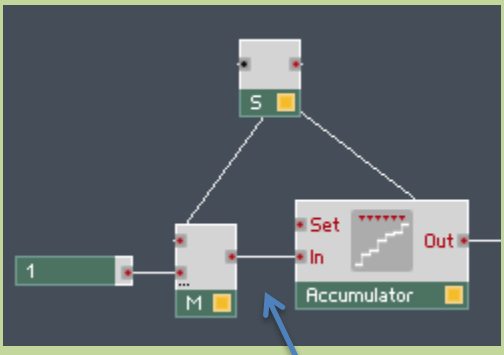
Значения будут сглаживаться это можно увидеть

Допустим Knob имеет значения
Перемещая от 0 к 1 с шагом=1
мы получим сигнал со Smoother

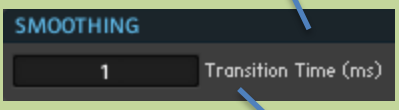
Event #	Value
1	0.0066
2	0.1066
3	0.2066
4	0.3066
5	0.4066
6	0.5066
7	0.6066
8	0.7066
9	0.8066
10	0.9065
11	1

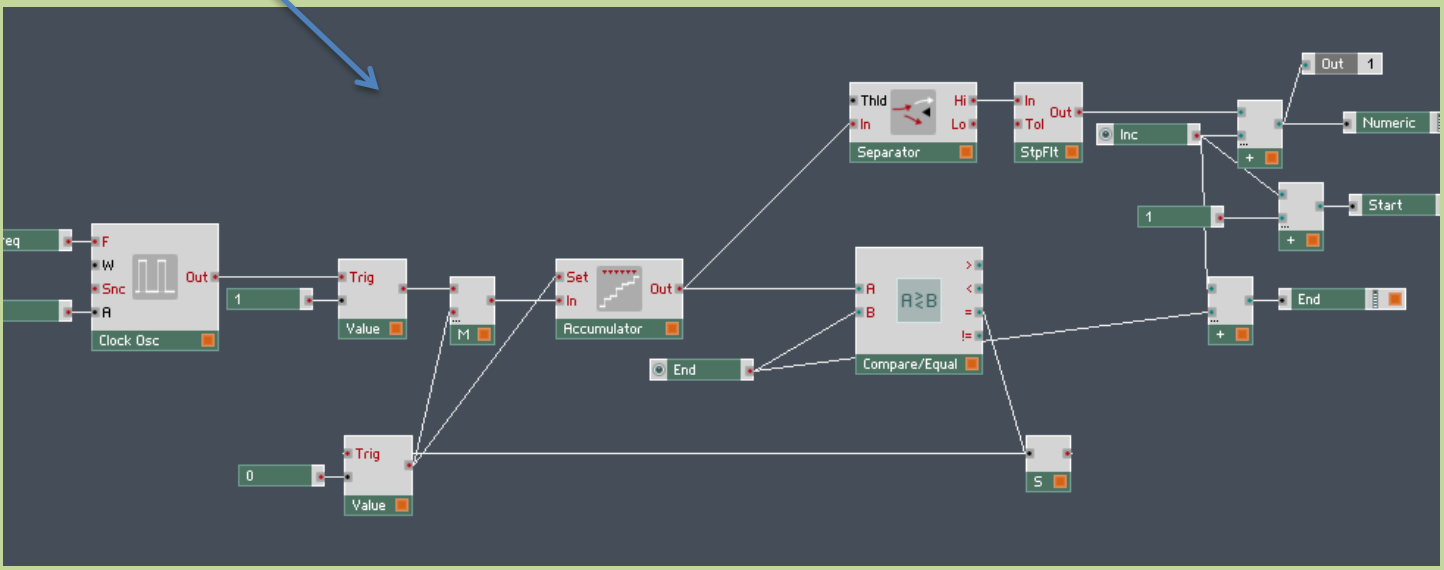
Сигнал от 0 до 1 дробится, сигнал получается не резкий, а плавный.

Event Smoother можно использовать в цепях с обратной связью

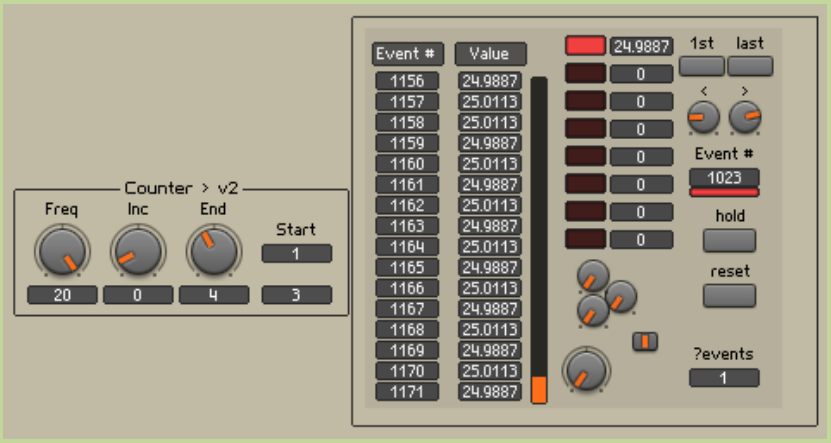


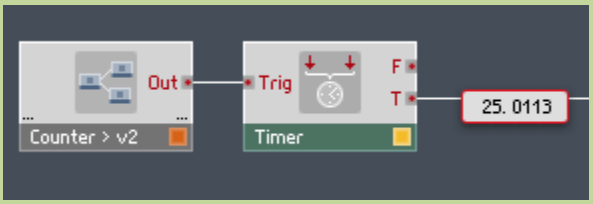
Event #	Value
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096
13	8192
14	16384
15	32768
16	65536



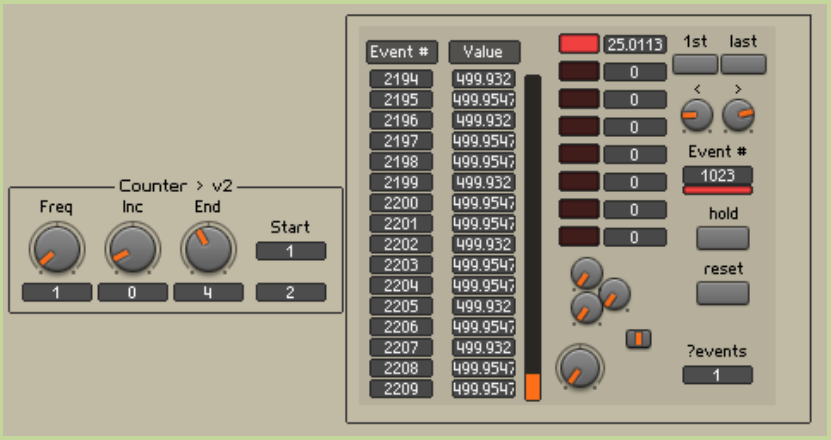


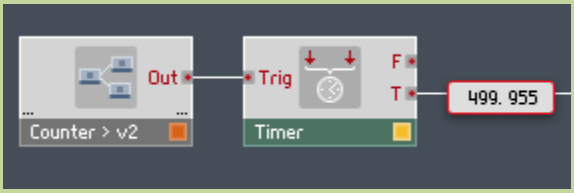
Почему работает такая схема : Clock Osc генерирует единицу допустим F=1 Hz, так как он генерирует сигнал 0...1 , то время между 0...1 будет 500 миллисекунд , а обратная связь через смуфер происходит за 1 ms. Даже если будет F=20 Hz , то растояние во времени между 0 и 1 будет (Hz/F/2 – 1000/20/20=25ms).
25 миллисекунд при частоте 20 Hz , а смуфер = 1 миллисекунде, то есть не успеет ещё прийти новый сигнал через 25 ms , а цепь обратной связи сработает .





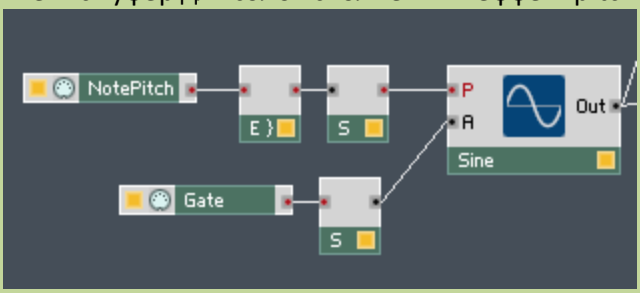
25 ms при 20 Hz





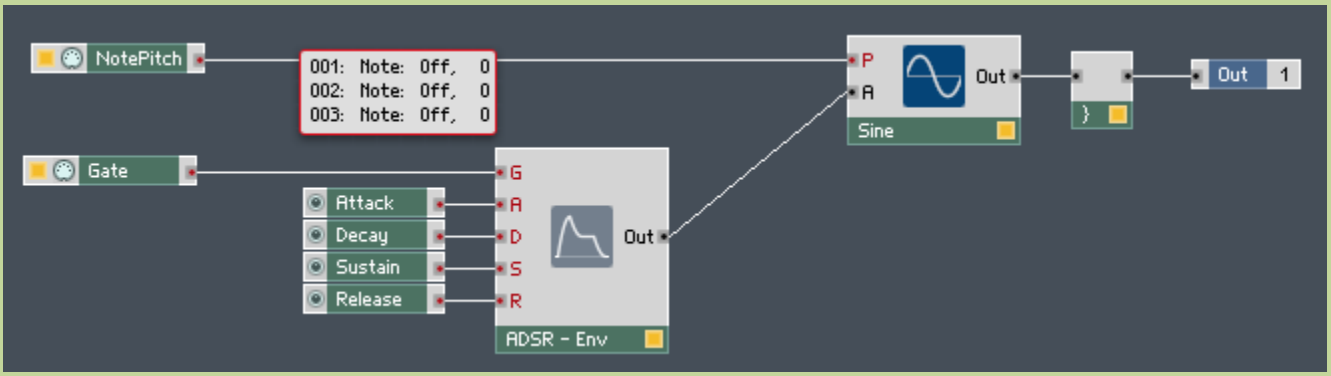
500 ms при 1 Hz

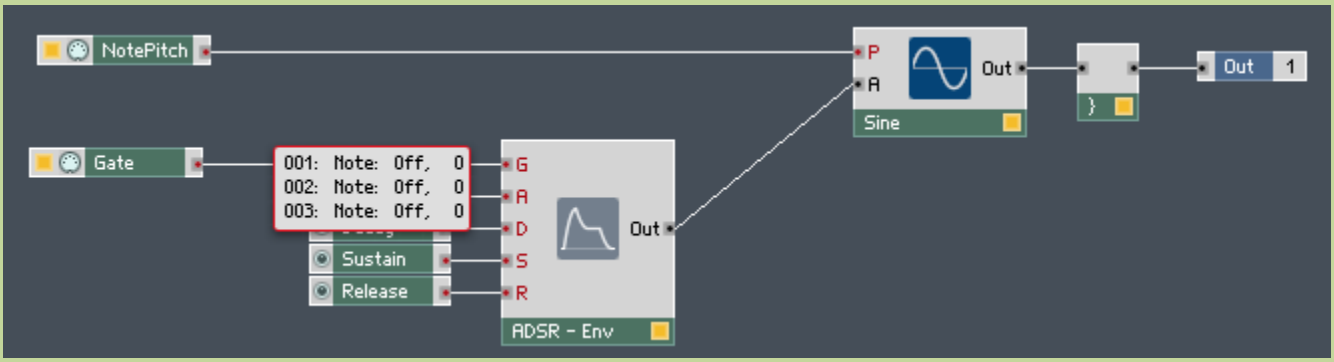
Евент смуфер для соло исполнения – ефект pitch bend



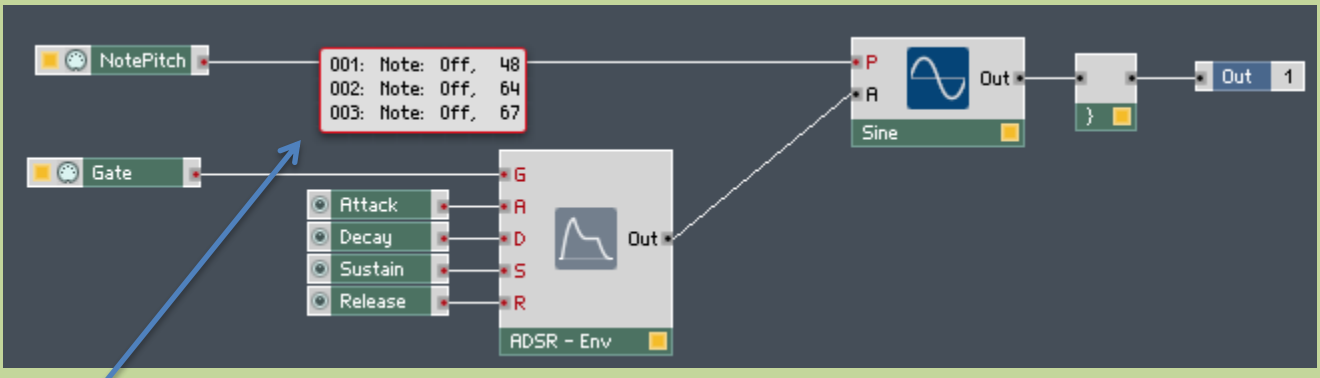
MIDI

Полифония – Poly - это когда одновременно звучат несколько нот, то есть аккорд. В настройках инструмента - instrument задаётся количество голосов. В данном примере их три. То есть максимальный аккорд будет – трезвучие.

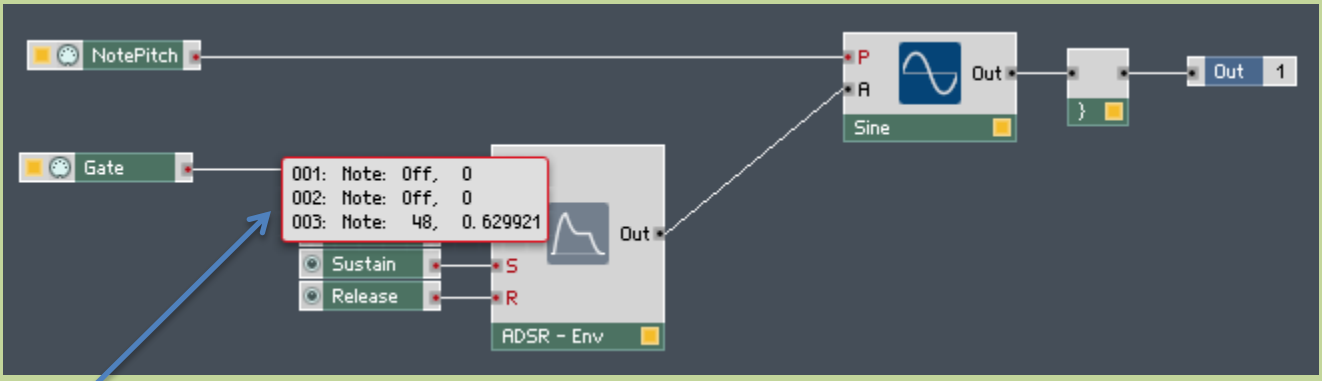




Изначальные миди-события(Note Pitch, Gate – Высота ноты, Клавиша нажата) – всё по нулям.
001,002,003 – голоса. Note Off – Какая клавиша нажата и удерживается - если Off – клавиша отжата, последний столбец – значения после того или иного модуля, например после NotePitch – значения нот.

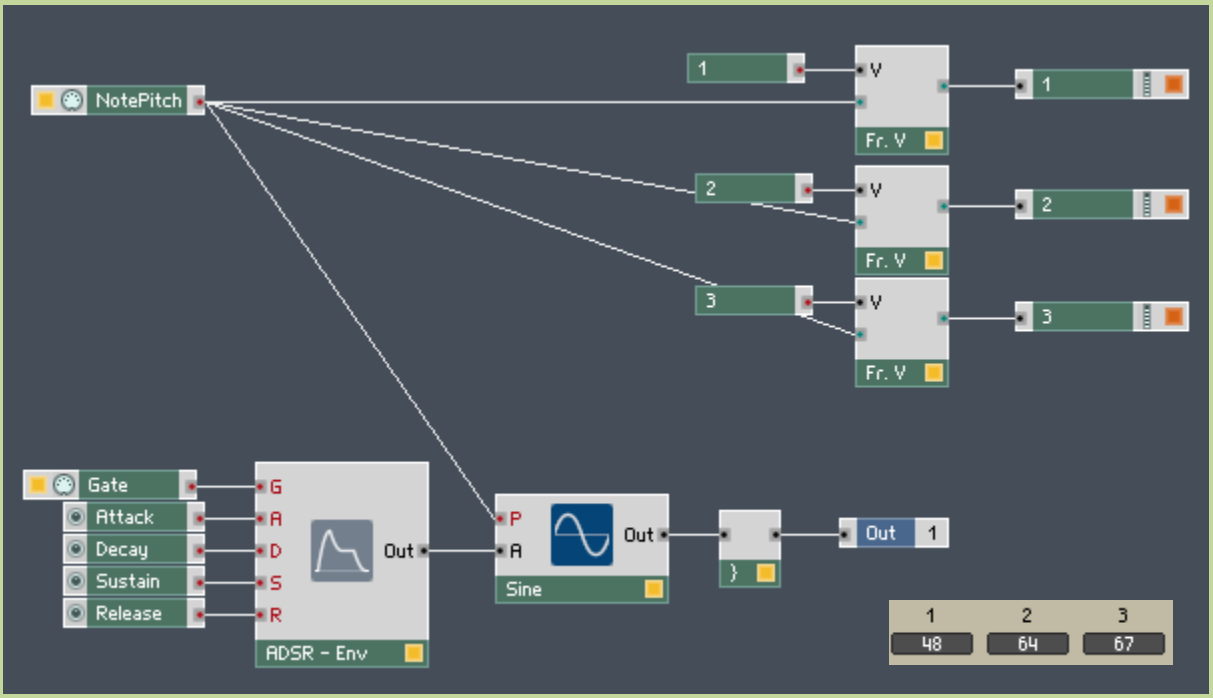


Нажали на три клавиши и отпустили(Off), последния столбец 48,64,67 – значение NotePitch

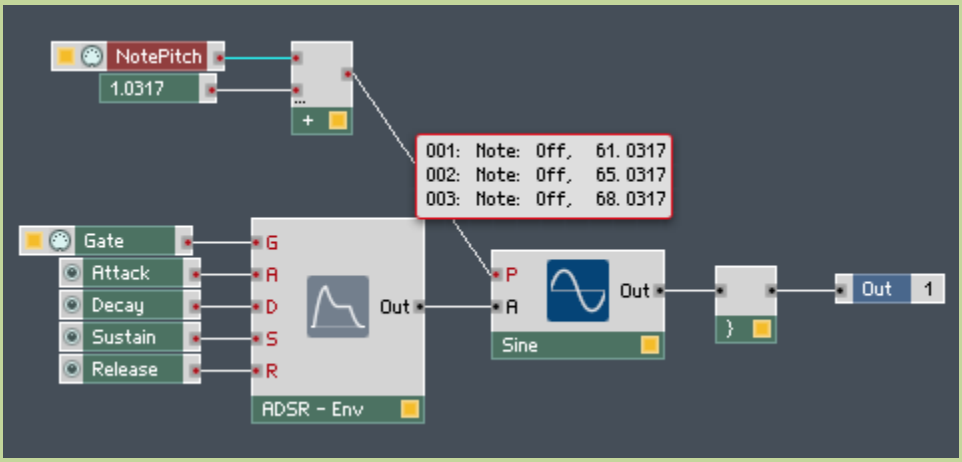


Нажали одну клавишу и удерживаем, последния столбец 0.62991 – значение Gate

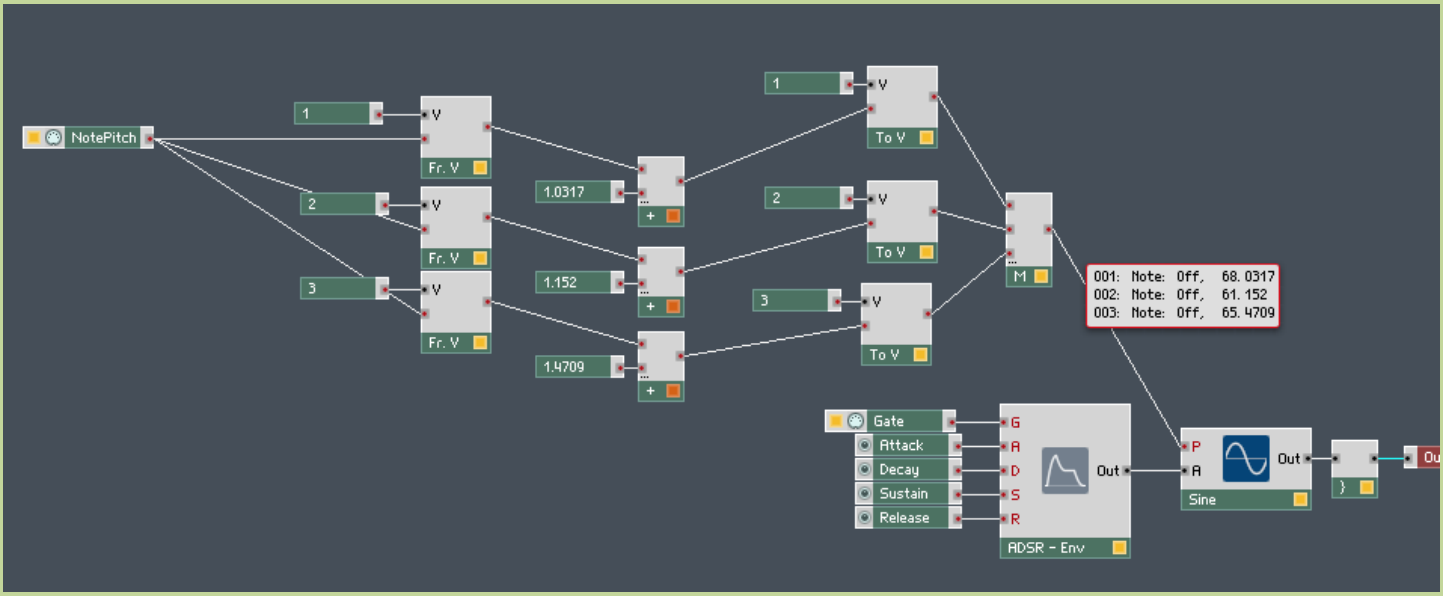
Используем Fr.V, To.V



Fr.V на вход поступает Poly-сигнал, а выход Mono-сигнал. В данном случае мы видим на Numeric Readout какие ноты нажимаются.

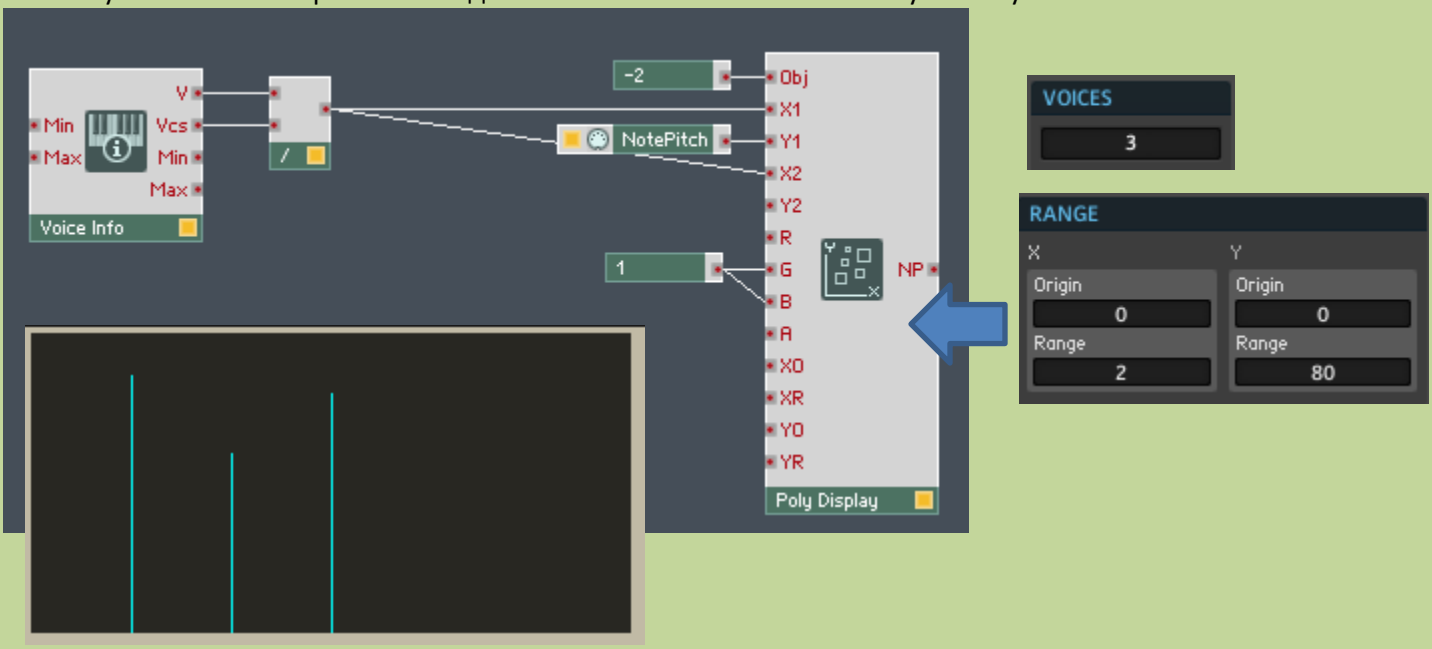


Если будем играть аккорд-трезвучие , то ко всем нотам будет прибавляться одна и та же константа.



To.V – на вход поступает Mono-сигнал, а выход Poly. Таким образом можно получать аккорды с произвольной подстройкой на ноты.

Чтобы лучше понять как работает миди-сигнал можно использовать такую схему



Нажимая аккорд-трезвучие можно заметить что звучание будет одно и то же, а сами миди-сигналы их появление будет разное. Например

```
001:  Note:  Off,  64
002:  Note:  Off,  48
003:  Note:  Off,  67
```

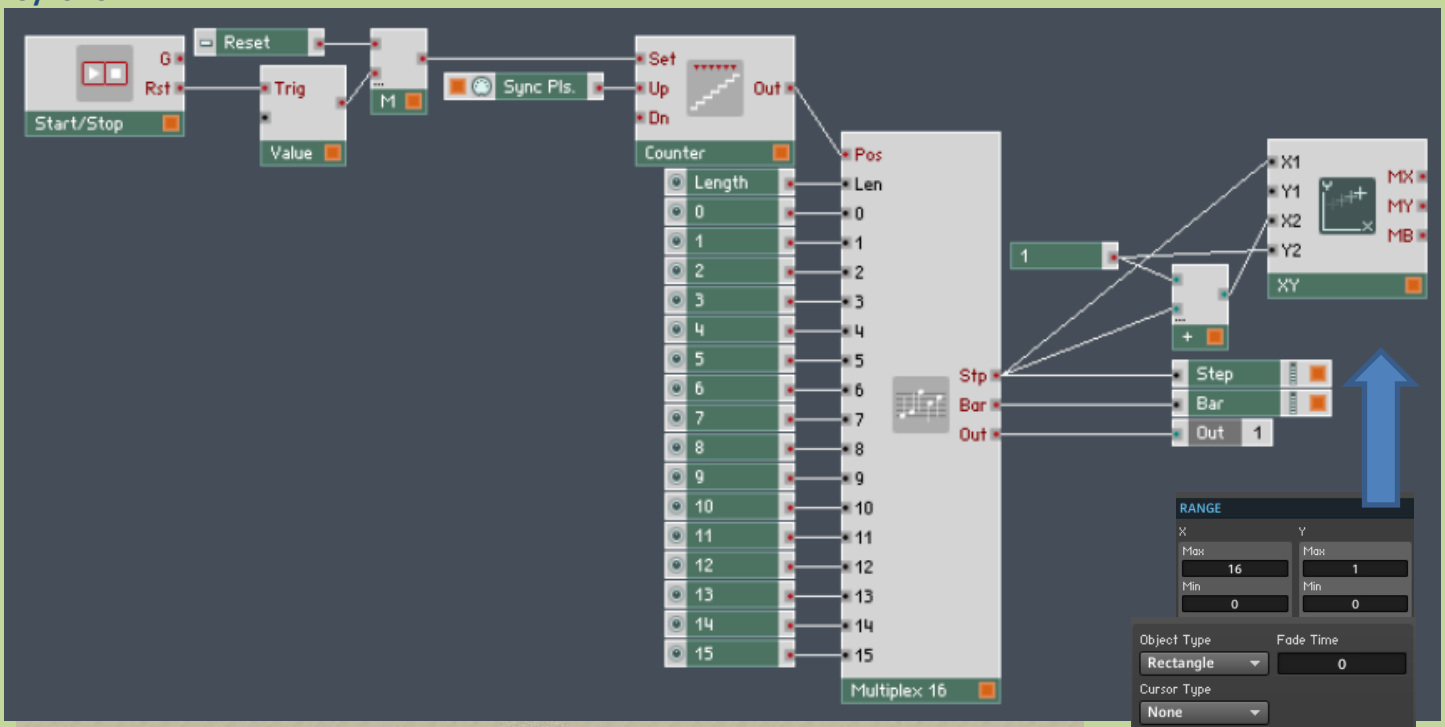
```
001:  Note:  Off,  67
002:  Note:  Off,  64
003:  Note:  Off,  48
```

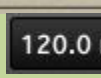
так как к каким-то клавишам пальцы прикасаются раньше а другие позже, также **появление и отображение миди-сигналов идёт по кругу.** Это значит что нажимая одну и туже клавишу можно увидеть

```
001:  Note:  Off,  67
002:  Note:  67,  67
003:  Note:  Off,  67
```

```
001:  Note:  67,  67
002:  Note:  Off,  67
003:  Note:  Off,  67
```

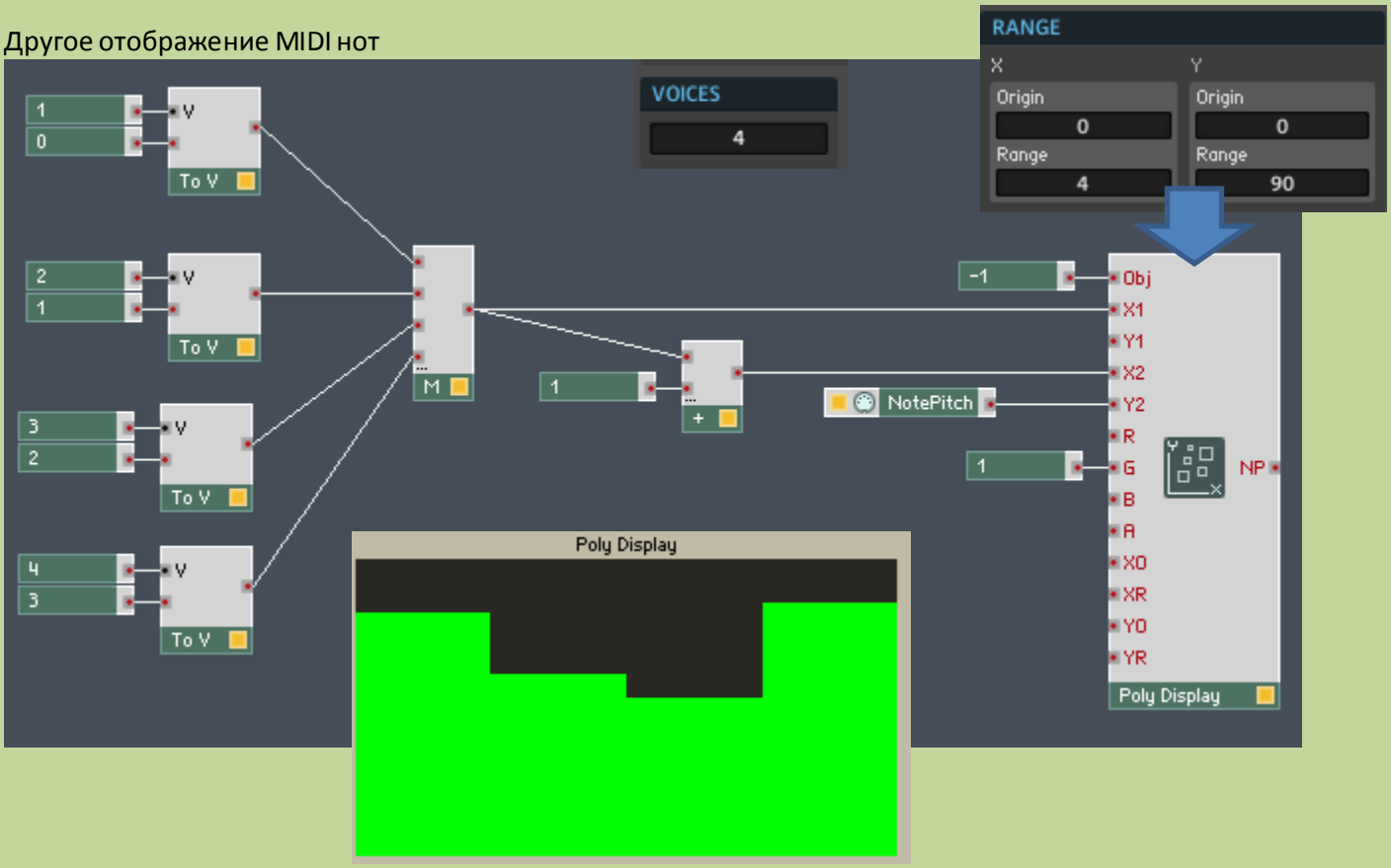
SyncPls



Скорость зависит от 

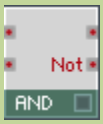
поэтому Sync Pls – синхроная пульсация (синхроная-зависит от BMP)

Другое отображение MIDI нот



Логические элементы

AND



AND.Когда на обоих входах положительные сигналы, то верхний выход = 1.

Оба значим два входа как A и B

A	0	0	1	1
B	0	1	0	1
Out	0	0	0	1

0 и 1 у A и B это отрицательный или положительный вход (даже малые значения учитываются, 0.001 это положительный сигнал), а выход Out это исключительно выход 0 или 1

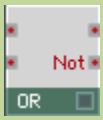
NAND

NAND это тот же модуль AND только выход у него нижний Not(отрицание).

A	0	0	1	1
B	0	1	0	1
Out-Not	1	1	1	0

Как видно, логика AND и NAND прямопротивоположна.

OR



OR.Когда на любом из двух или на обоих входах положительный сигнал, то верхний выход = 1

A	0	0	1	1
B	0	1	0	1
Out	0	1	1	1

NOR

NOR это тот же модуль OR только выход у него нижний Not.

A	0	0	1	1
B	0	1	0	1
Out-Not	1	0	0	0

EXOR

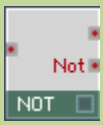
EXOR работает как и OR, но когда два положительных сигнала на входе, то выход будет = 0. Поэтому о логике EXOR говорят «исключающий ИЛИ»

A	0	0	1	1
B	0	1	0	1
Out	0	1	1	0

Exor с выходом Not, в электронике встречается редко

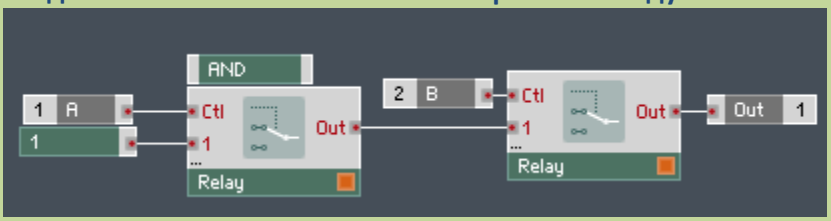
A	0	0	1	1
B	0	1	0	1
Out-Not	1	0	0	1

NOT

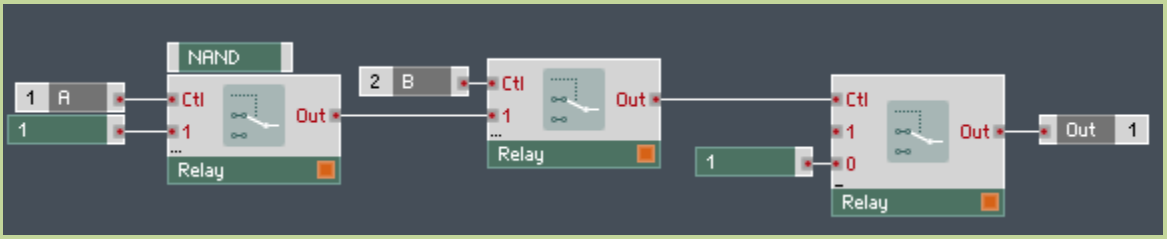


Как видно у Not только один вход. Верхний выход Not является отрицанием любого сигнала, если вход положительный, то выход=0, если вход отрицательный, то выход=1. Нижний выход является потверждением любого сигнала (плюс это 1, минус это 0)

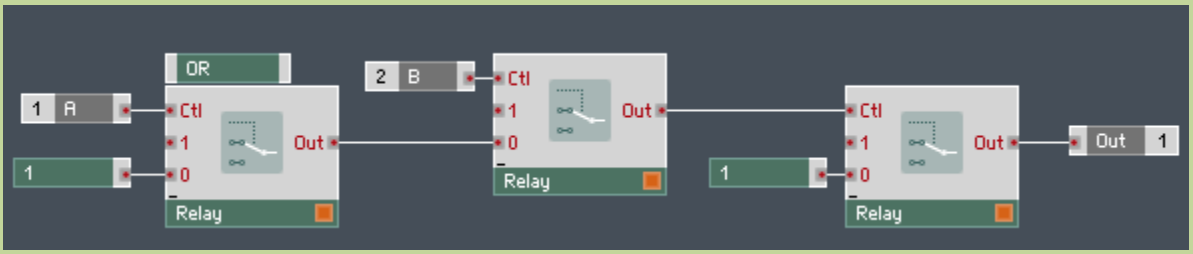
Создание логических элементов из первичных модулей :



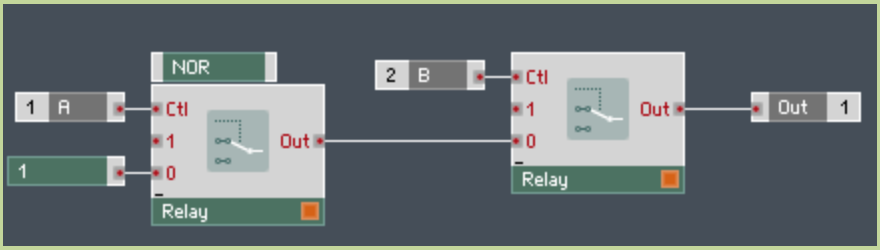
AND



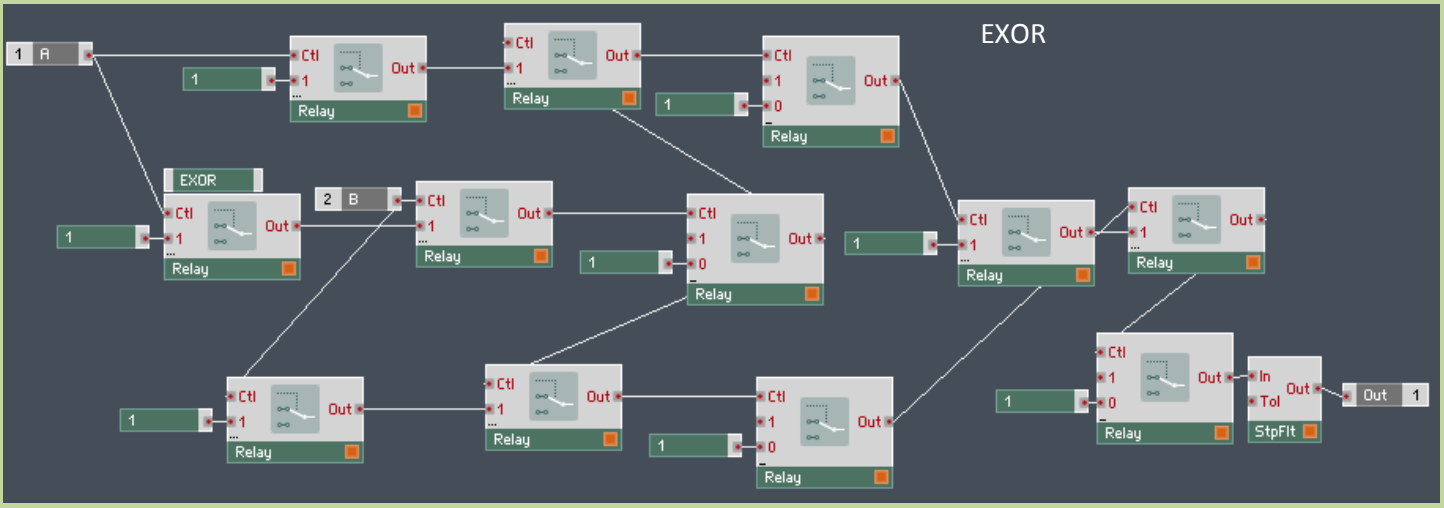
NAND



OR

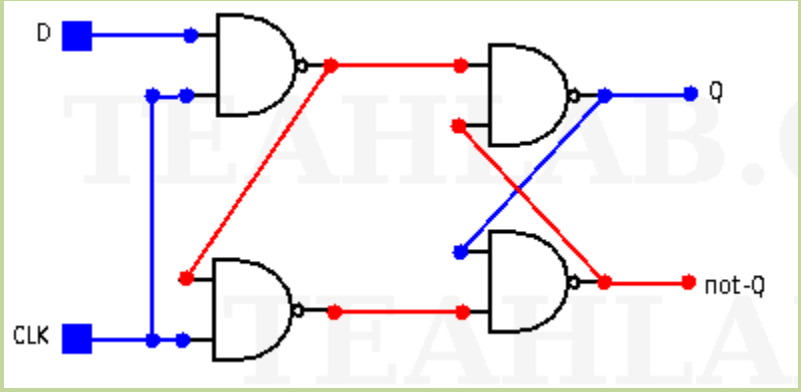


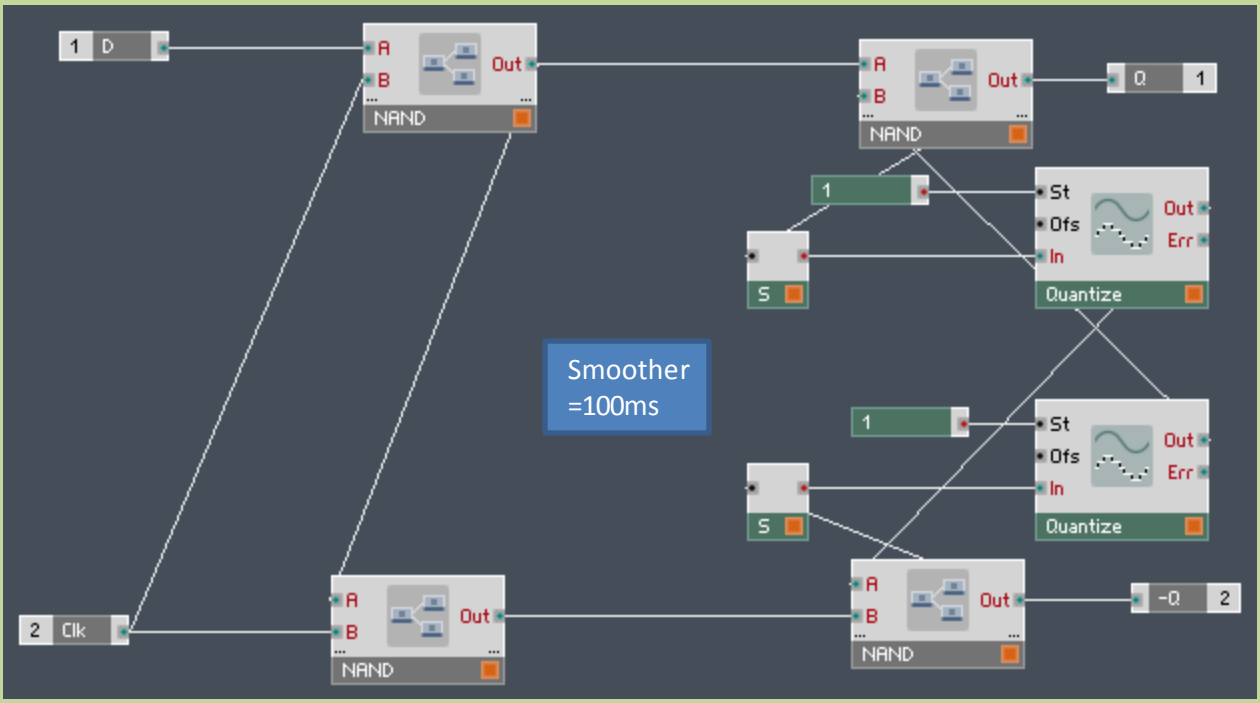
NOR



Из логических элементов собраных из реле можно создавать другие разнообразные модули.

например Gated D latch

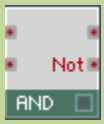




Gated D latch является заглушкой для 1 бита. Если мы нажмём Clk, затем D, затем Clk отключим, то далее нажимая D, выход Q не будет меняться – схема заперла значение с D. Чтобы очистить 1 бит, надо D выключить, а Clk включить

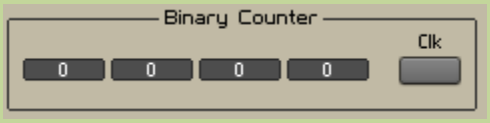
Buttons D и Clk в режиме Toggle

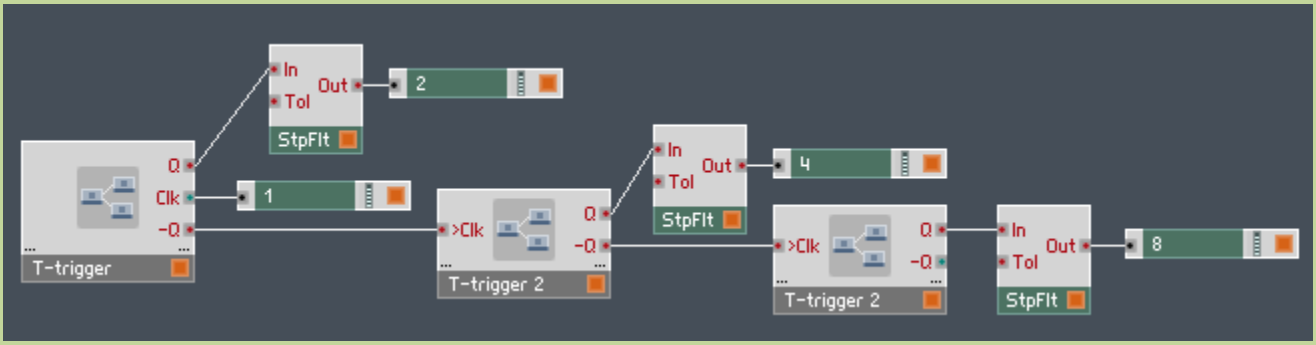
Важное замечание – если использовать встроеные логические элементы, то у вас не получится создать выше описаную схему.

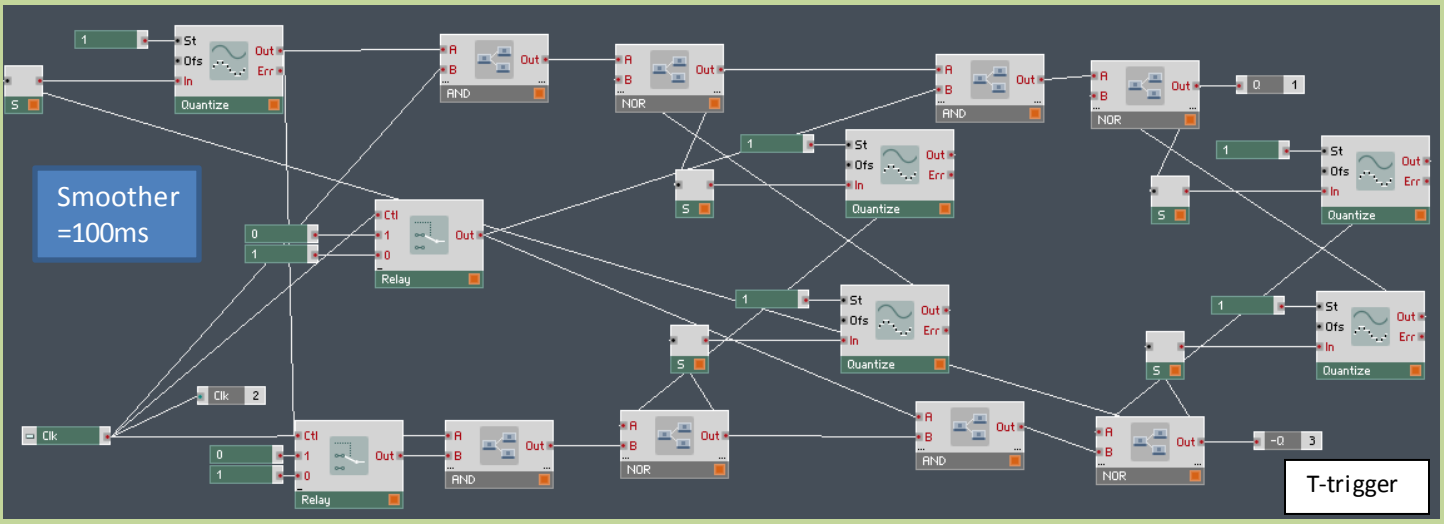


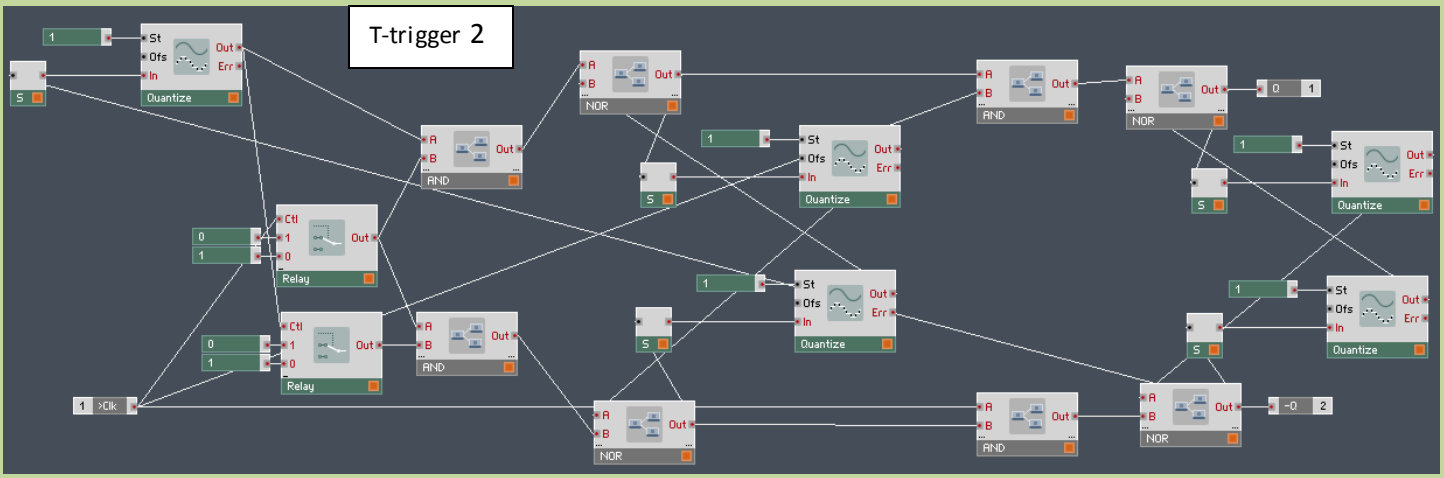
Так можно создать RS, JK, T triggers и т.д

Бинарный счётчик









Нажимая Clk мы получим следующую последовательность :

00-0000
01-0001
02-0010
03-0011
04-0100
05-0101
06-0110
07-0111
08-1000
09-1001
10-1010
11-1011
12-1100
13-1101
14-1110
15-1111

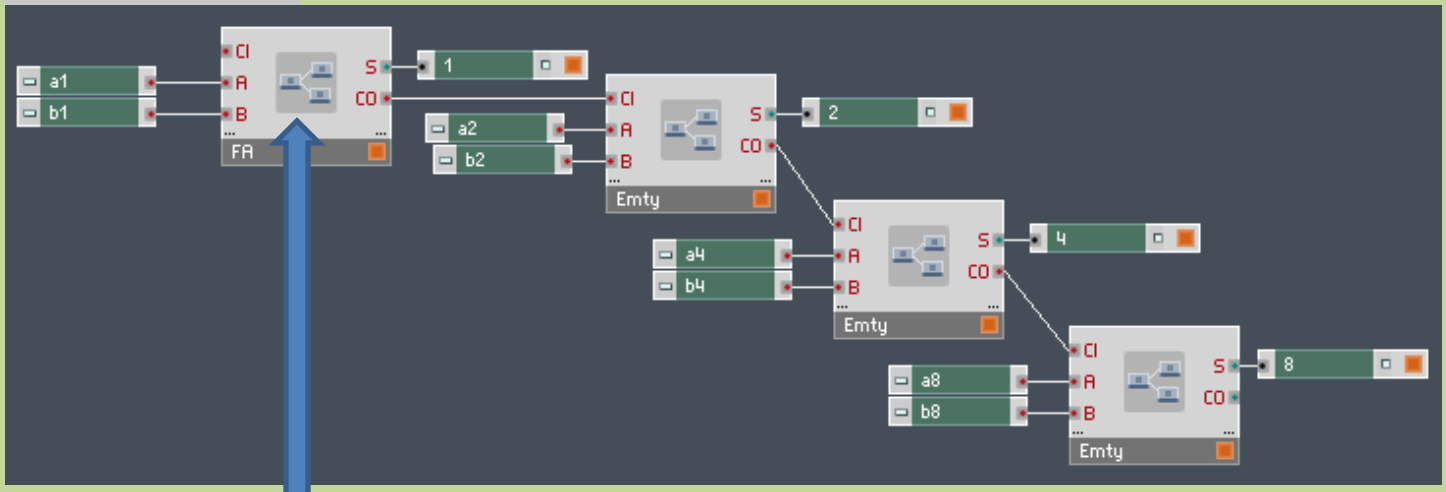
Красным выделен выход 1, синим выход 2, серым выход 4, чёрным выход 8. Как видно T-trigger в бинарном счётчике является частотным делителем - выход 1 это последовательность 0,1, выход 2 это 0,0,1,1, выход 4 это 0,0,0,0,1,1,1,1, выход 8 это 0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1.

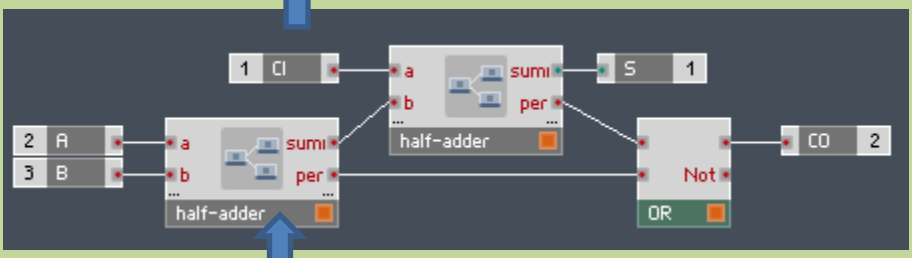
В десятичной системе счисления эта последовательность будет от 0 до 15 итого 16 вместе с 0.

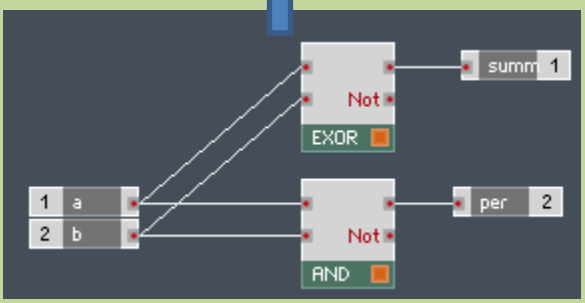
Бинарный сумматор-калькулятор считающий до 15

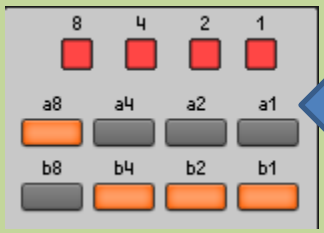
Бинарные числа
1 0 1 0 – число в бинарной системе, (1-потверждает нижестоящее число, 0-отвергает)
8 4 2 1 – из каких десятчных чисел состоит бинарное число
1010=8+2=10(десятичная система)

Система из букв “a” это первое слагаемое число, на панеле слева из букв “a” набрано бинарное число 101, что в десятичной системе означает 5, из букв “b” набрано второе слагаемое чисо 001, что в десятичной системе означает 1. 5+1=6 это отображается на лампочках









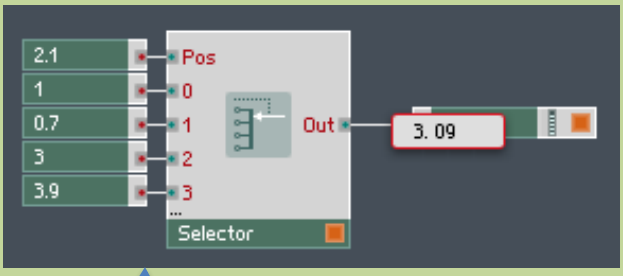
1000+0111=1111(бинарная система), 8+7=15(десятичная система)

Selector

Находим значение линейной интерполяции.

Когда на вход Pos подаются целые значения (0,1,2 и т.д), то выход будет равен тому, что подаётся на вход. Допустим Pos=1, а на вход 1 подаётся 3,7, то выход=3,7.

Совсем иначе дело обстоит, когда на вход Pos подаются дробные значения. У Selector стоит по умолчанию Curve-Linear, это и есть линейная интерполяция между двумя значениями входов.



Как видно Pos=2.1 и это значение находится между входами 2 и 3, тогда возникает вопрос: если вход 2=3, а вход 3=3.9, то что будет на выходе когда Pos=2.1 ?

Для этого можно воспользоваться таким расчётом

Out=N+L,

L=((N+1)-N)*Q,

Q=Pos mod 1, if Q=0 then Q=1

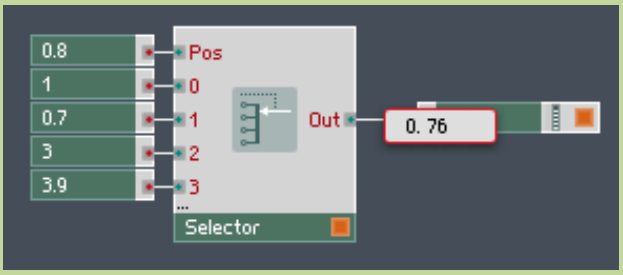
Проверим расчёт

Q=2.1 mod 1=0.1

L=(3.9-3)*0.1=0.09

Out=3+0.09=3.09

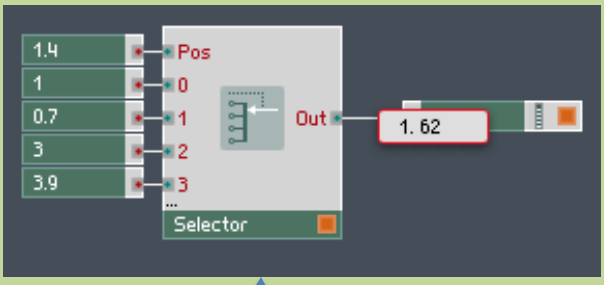
Когда Pos имеет дробное значение, он находится между двумя входами и эти два входа всегда будут активны, если каждый из них изменять ,то и выход будет меняться



Q=0.8 mod 1=0.8

L=(0.7-1)*0.8=-0.24

Out=1+(-0.24)=0.76



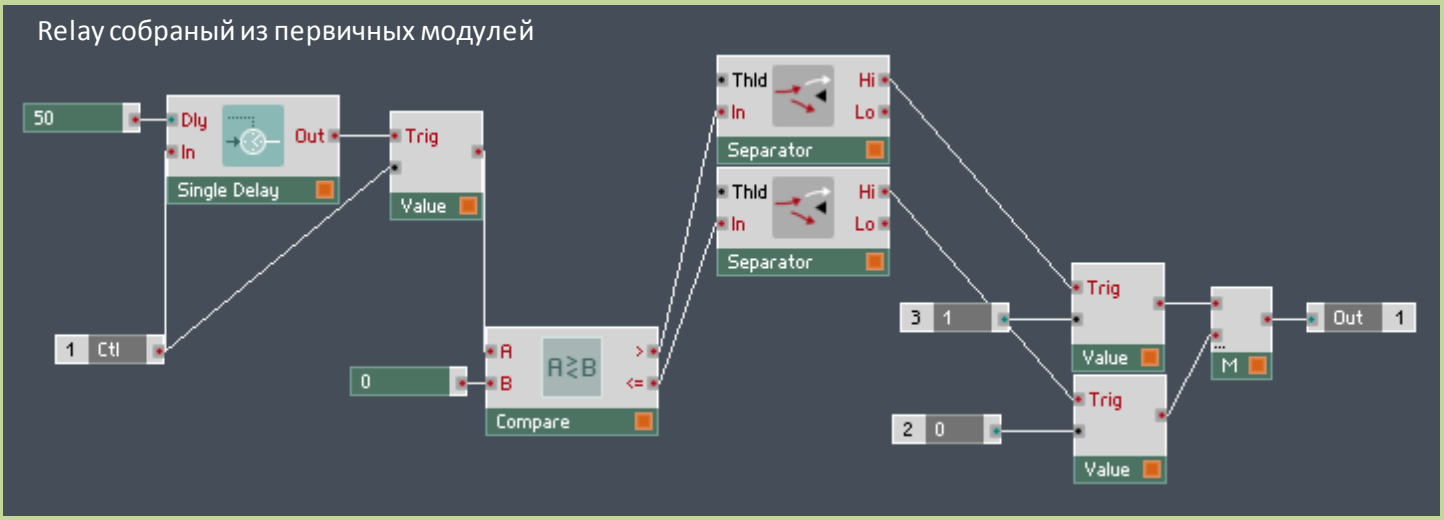
Q=1.4 mod 1=0.4

L=(3-0.7)*0.4=0.92

Out=0.7+0.92=1.62

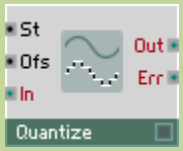
Relay

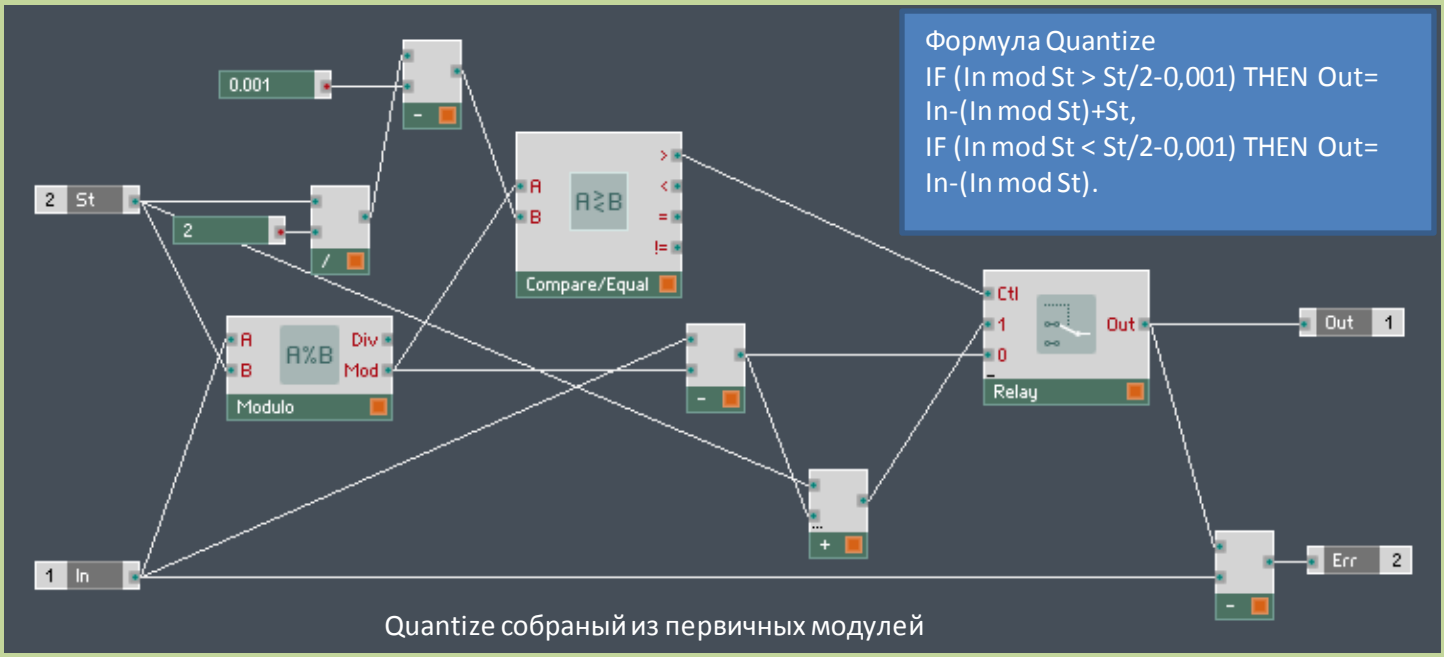
Relay собраный из первичных модулей



Реле имеет 3 входа и один выход, если значение на входе Ctl положительное,то выход будет браться со входа 1, если значение на входе отрицательное или равно 0, то выход будет браться со входа 0.

Quantize





Quantize собраный из первичных модулей

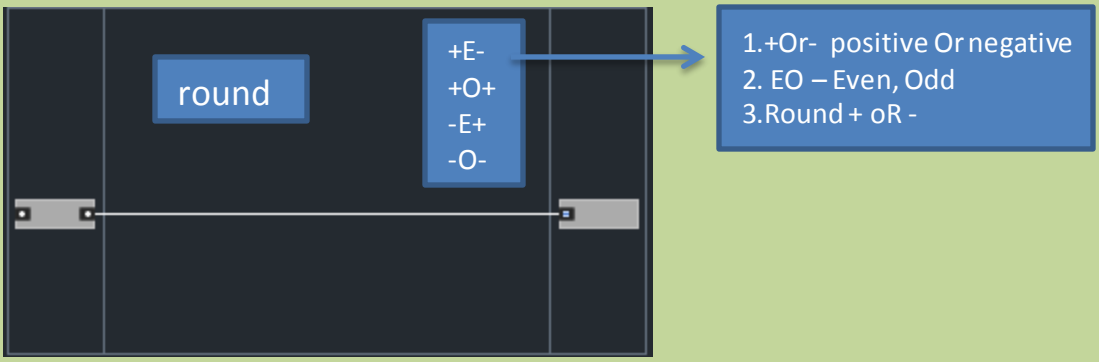
Quantize округляет входящий сигнал согласно входу St(шаг округления). Допустим на входе 25.5 , то при St=1 Out=26. Когда St=1, это есть стандартное округление, если дробная часть = .499 или меньше, то число округляется в меньшую сторону, если дробная часть больше .499 или равно .5 то число округляется в большую сторону.

Если шаг округления 1.4, а вход 25.5, то воспользуемя формулой :
25.5mod1.4=0.3 < 1.4/2-0.001=0.699
25.5-0.3=25.2

Выход Err является разницей между Out и In, Out-In=Err

Пример : Найти значение Err если In=-17.8, St=7.3
-17.8mod7.3=-3.2 < 7.3/2-0.001=3.649
-17.8-(-3.2)=-14.6
Err=-14.6-(-17.8)=3.2

Виды округления

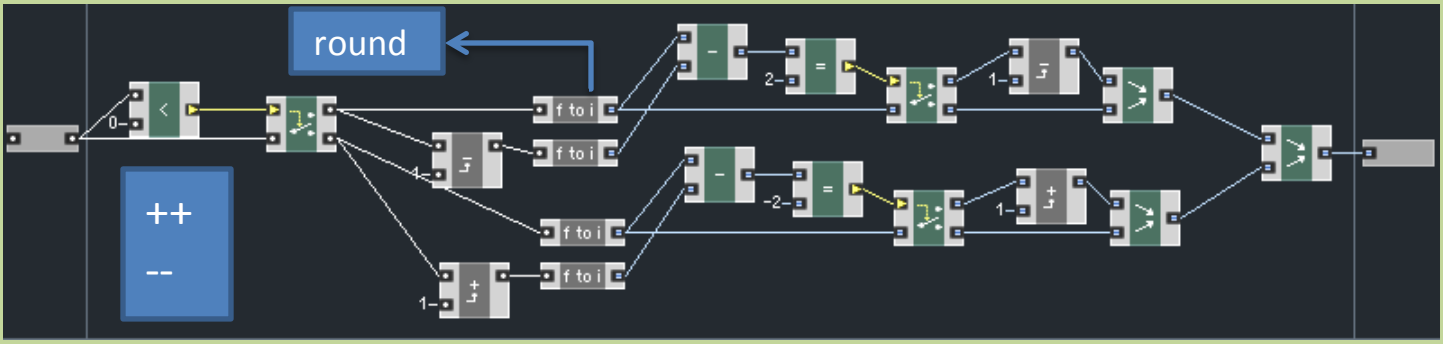


0.5 to 0
3.5 to 4

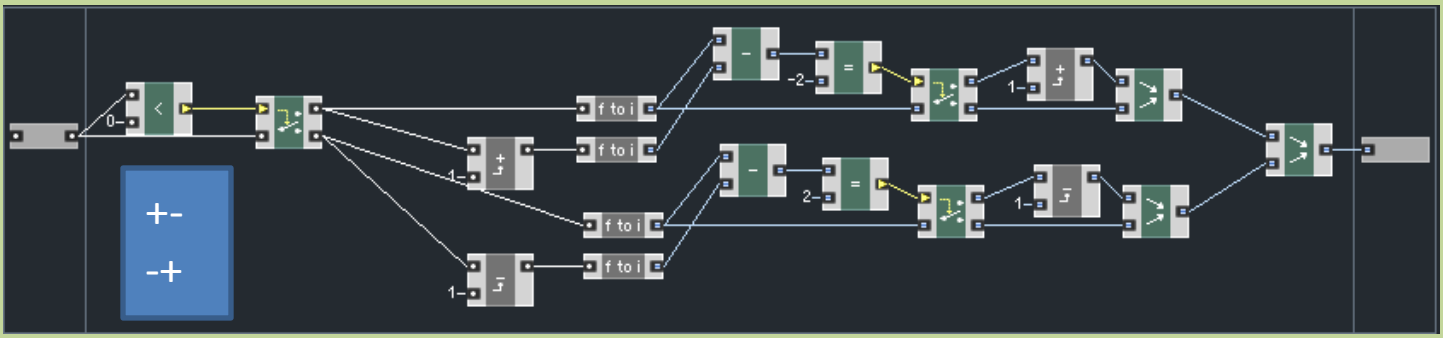
Округление в Core по умолчанию : для положительных значений - если целое чётное, то округление в меньшую сторону, если целое нечётное, то округление в большую сторону.

-1.5 to -2
-4.5 to -4

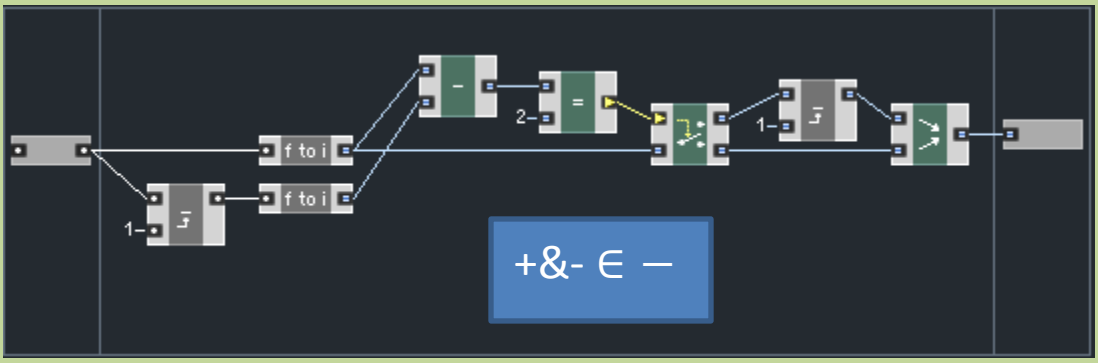
С отрицательными значениями всё наоборот.



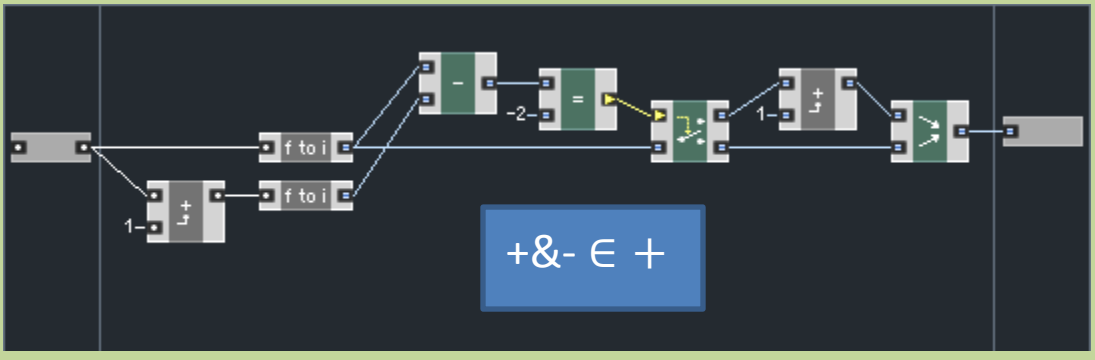
0.5 to 1 Положительные округляются в большую сторону, отрицательные в меньшую.
3.5 to 4
-4.5 to -5



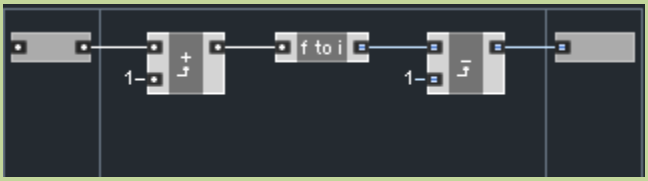
0.5 to 0 Положительные округляются в меньшую сторону, отрицательные в большую.
3.5 to 3
-4.5 to -4



0.5 to 0 Положительные и отрицательные округляются в меньшую сторону
3.5 to 3
-4.5 to -5



0.5 to 1 Положительные и отрицательные округляются в большую сторону
3.5 to 4
-4.5 to -4



+E+
+O-
-E-
-O+

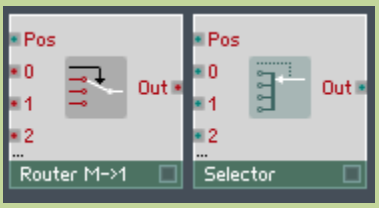
1.+Or- positive Or negative
2. EO – Even, Odd
3.Round + oR -

0.5 to 1 Положительные чётные в большую сторону, положительные нечётные в меньшую.
3.5 to 3 Отрицательные наоборот.
-4.5 to -5
-3.5 to -3

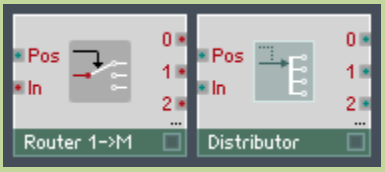
Routers

Router 1,2 – отличается от Relay тем, что если сигнал на каком-либо из входов не активный, то переключая Ctl выход будет показываться с последнего активного входа.

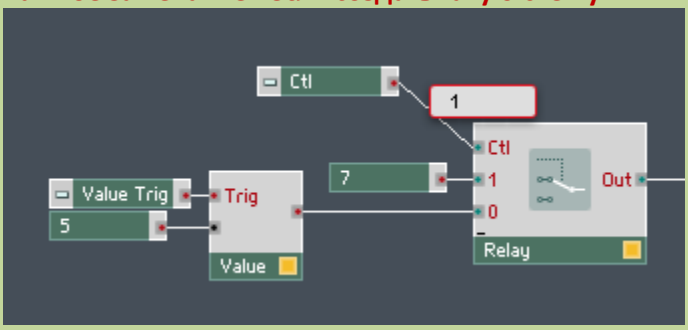
Router M->1 – отличается от Selector также как в предыдущеи примере



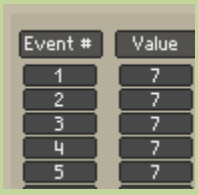
Router 1->M – отличается от Distributor тем, что переключая Pos выходы у Router 1->M остаются, а у Distributor становятся нулями.

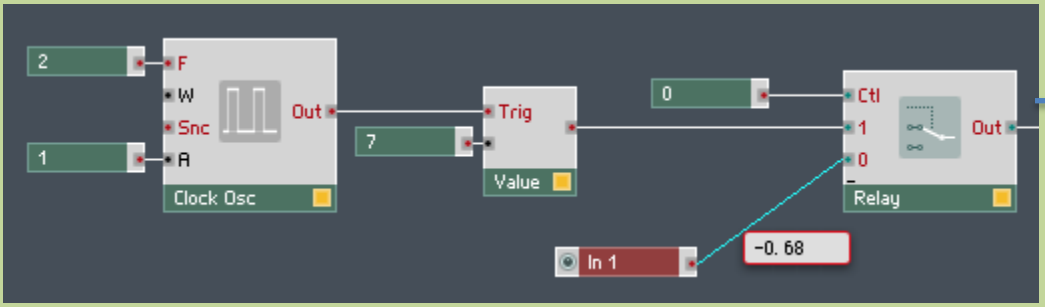


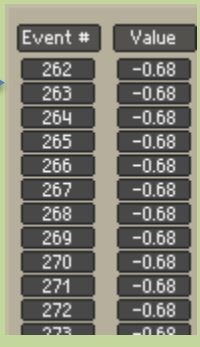
Важное замечание : если создать такую схему



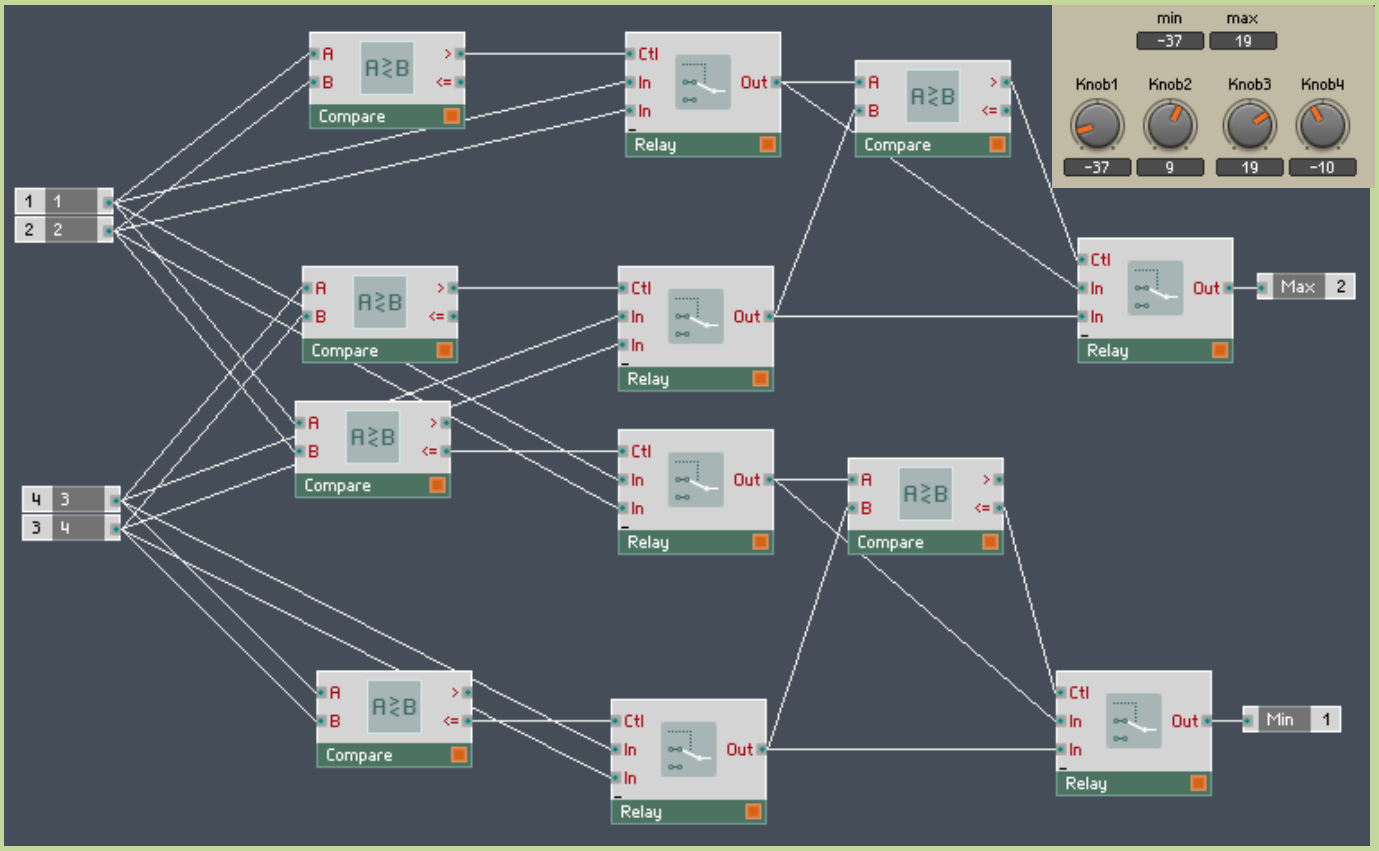
Ctl = 1
нажимая ValueTrig мы на выходе Relay будем получать
хотя сама 7 не активна. Тоже самое и для Selector.



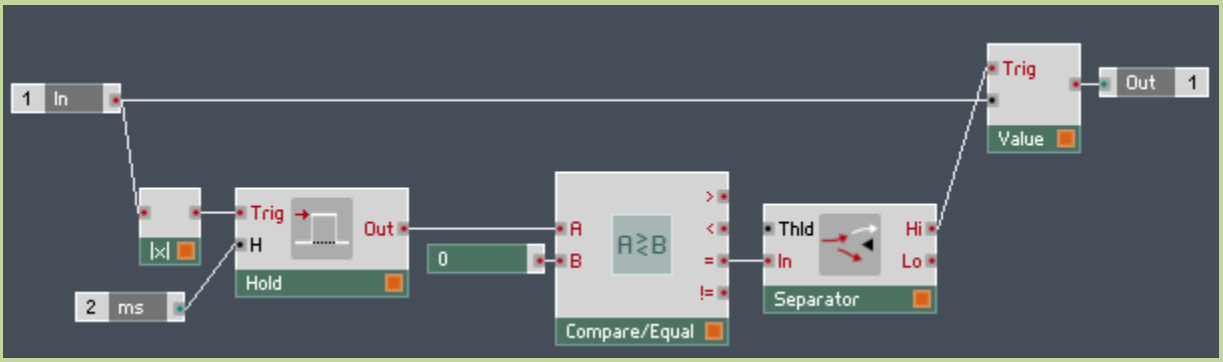




Max/Min – показывет макс.мин из четырёх значений

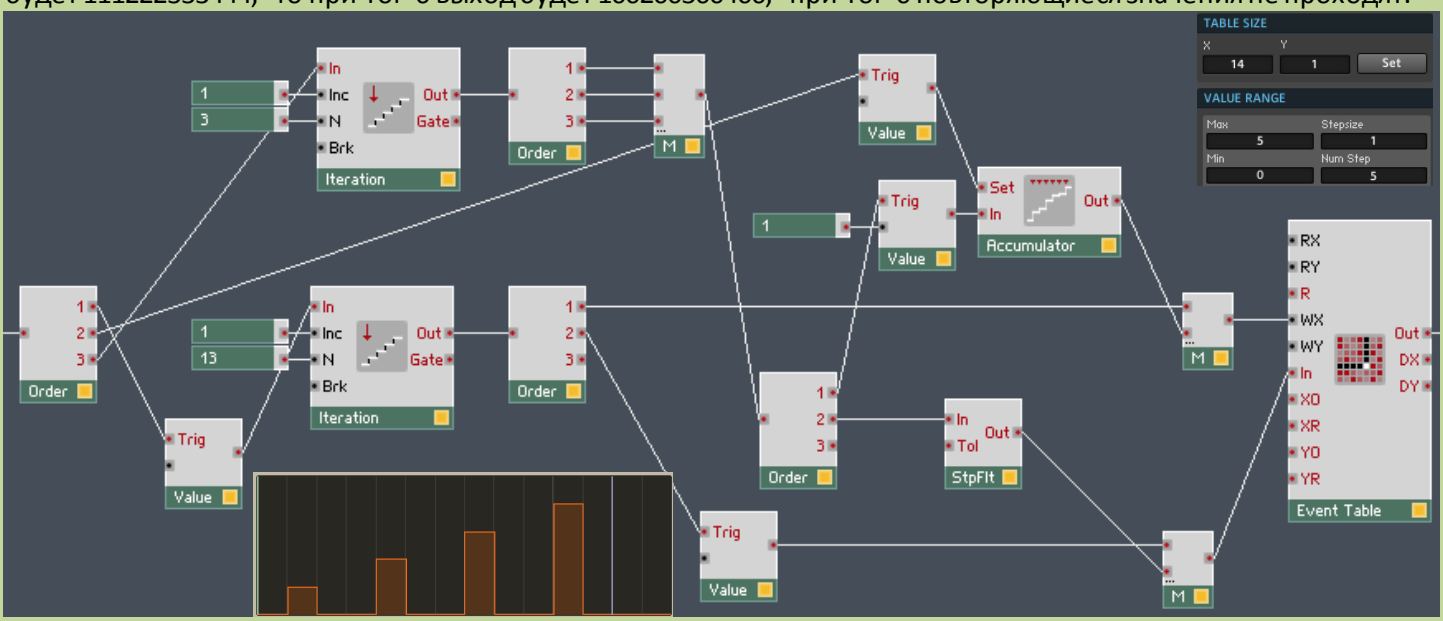


Event Delay – работает как модуль Single Delay

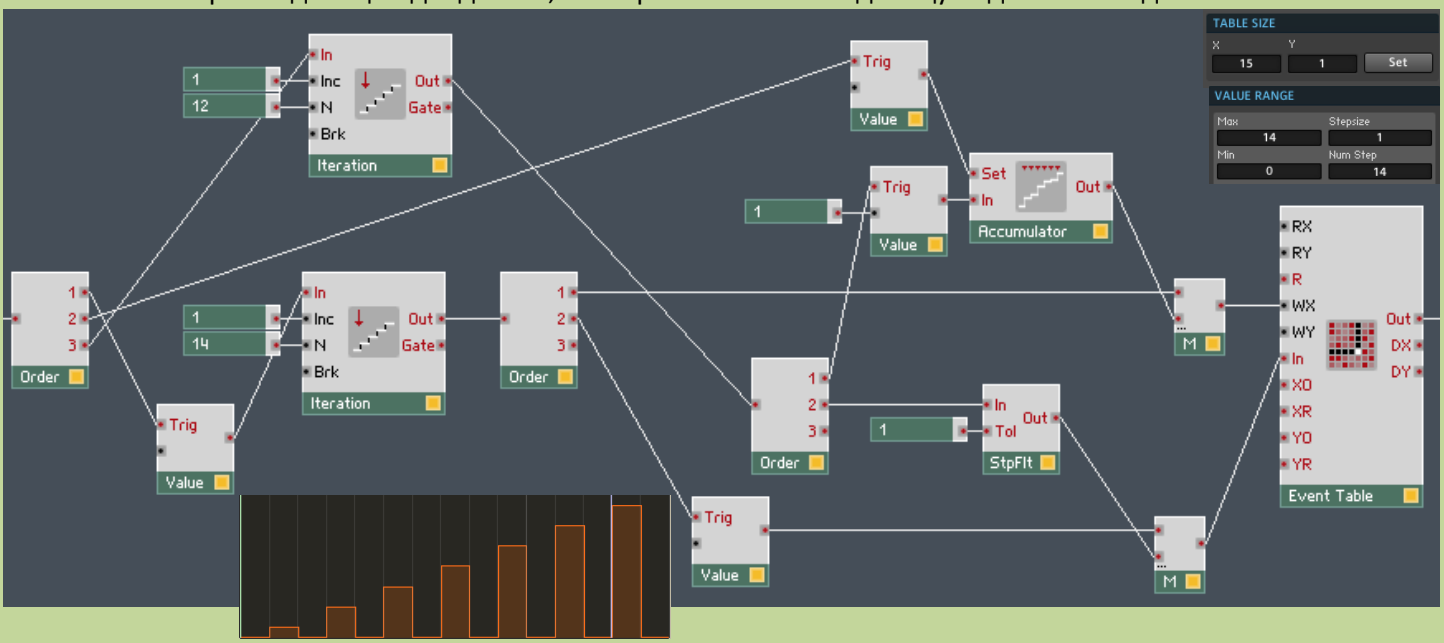


Step Filter

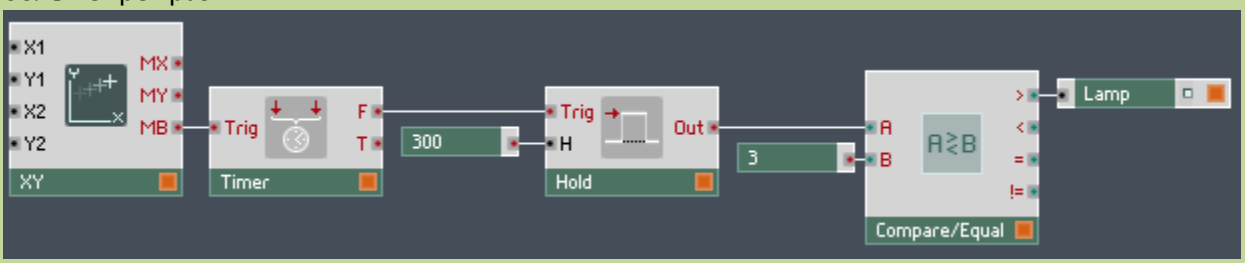
Назначение Step Filter – не пропускать значения выше или ниже заданого порога Tol. Например если на входе будет 111222333444, то при Tol=0 выход будет 100200300400, при Tol=0 повторяющиеся значения не проходят.



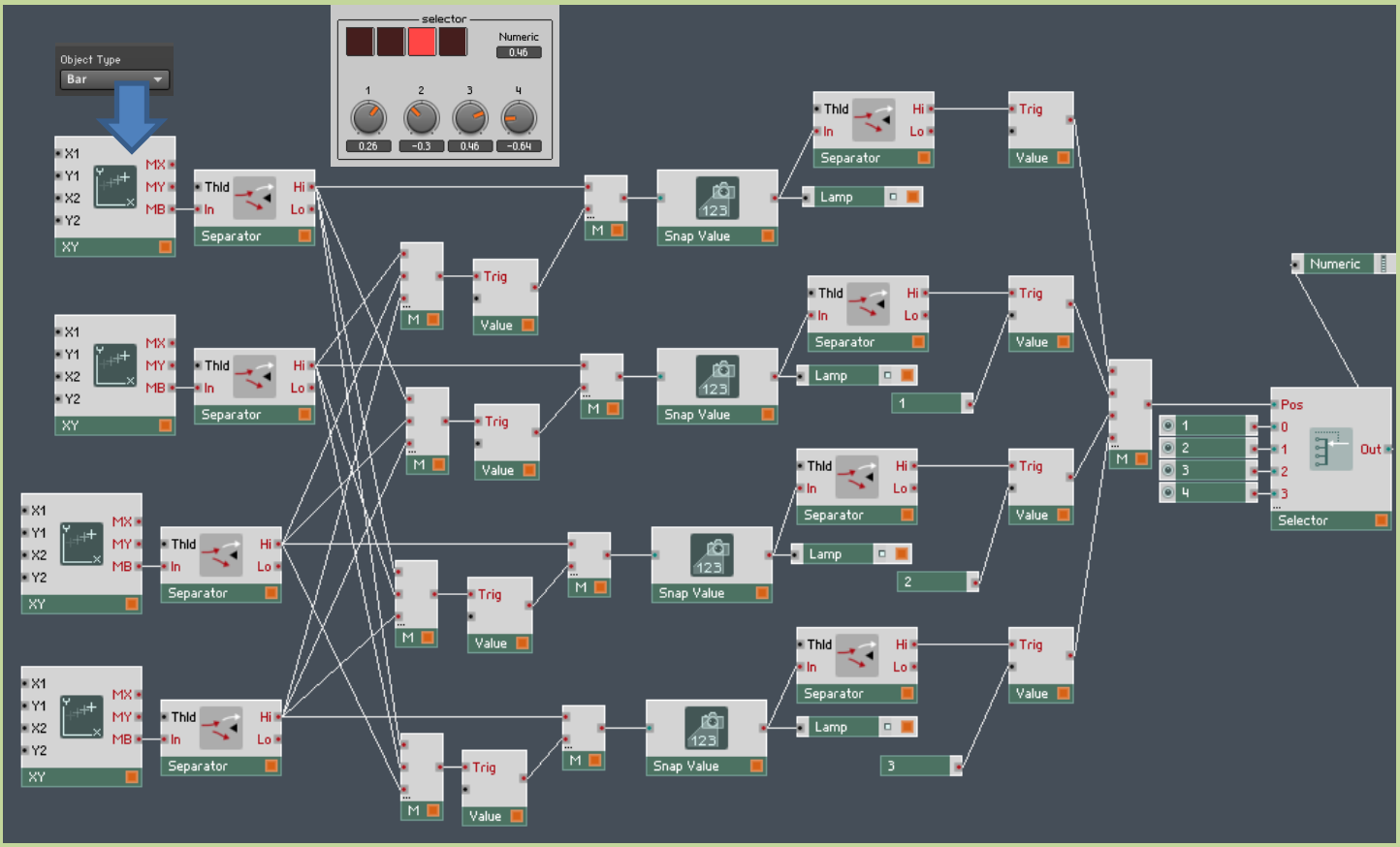
Если подадим такую последовательность 123456789,10,11,12,13 а Tol=1, то выход будет 1,0,3,0,5,0,7,0,9,0,11,0,13 так как после первой единицы идёт двойка, а они различаются на единицу то двойка откидывается.



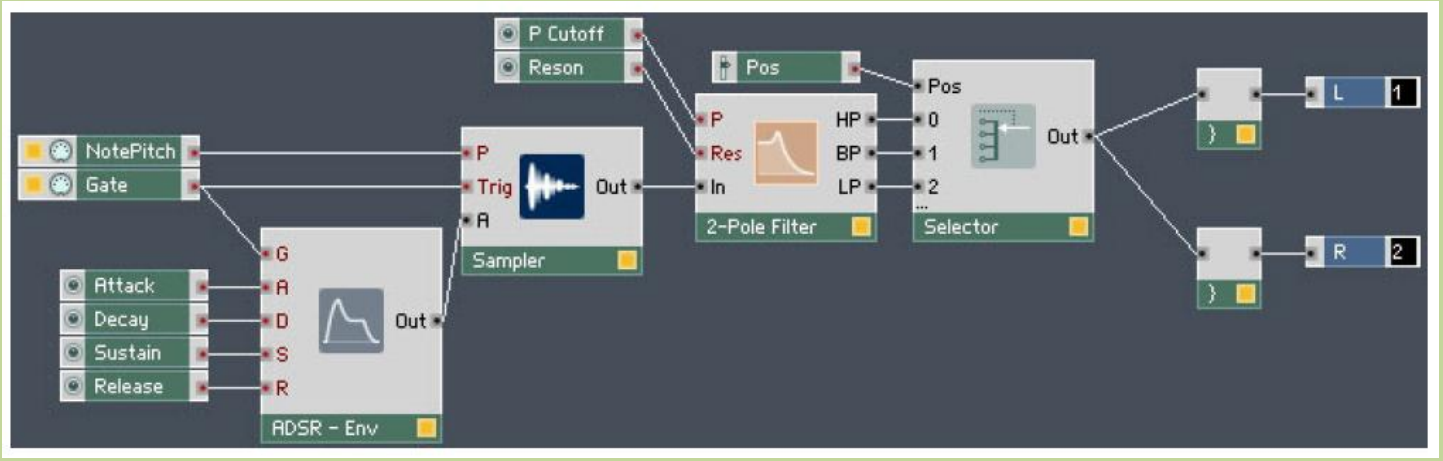
Подсчёт кликов мыши за секунду(чтобы загорелась лампочка нужно за 1 секунду нажать левую клавишу мыши больше трёх раз.



Графический интерфейс для селектора



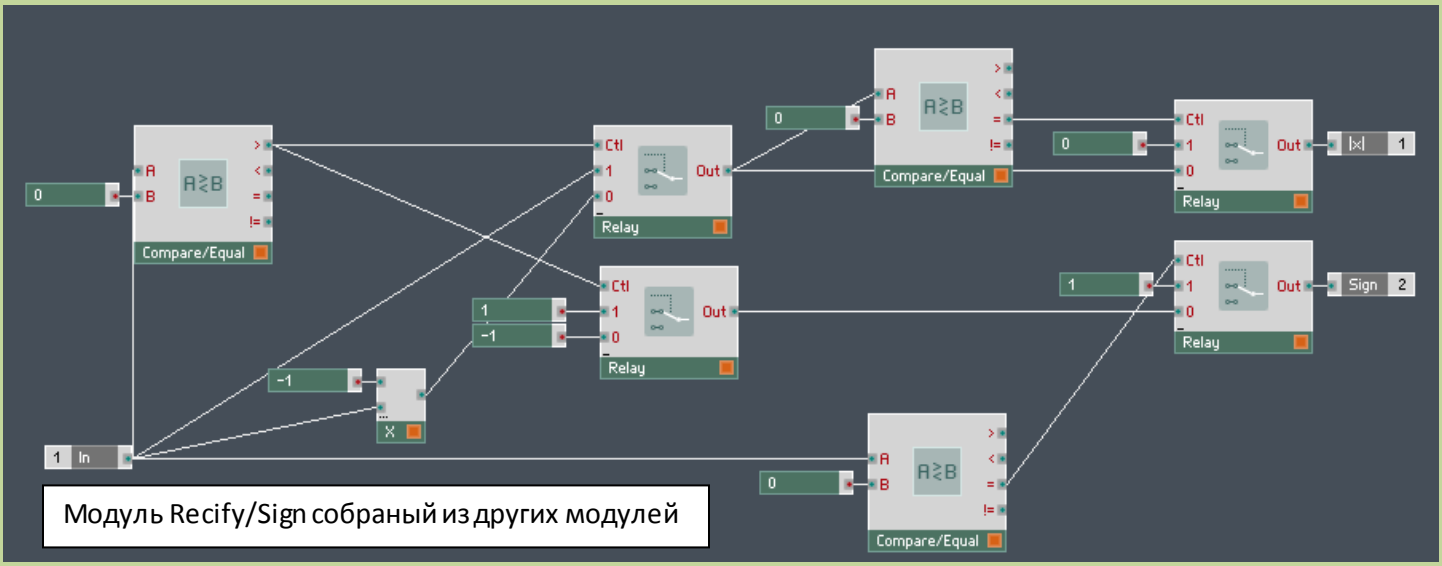
Модули XY и Lamp имеют одинаковый размер по Width, Height и накладываются друг на друга на панели.



Эффект как у кроссфейдар, переключая pos мы плавно идём от фильтра к другому фильтру HP,BP,LP

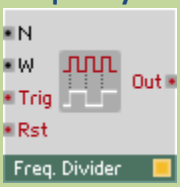
Recify/Sign

Sign – если входящее значение >=0 то выход будет 1, если вход <0, то выход будет -1.
|x| Recify – преобразует любые входящие значение в положительные

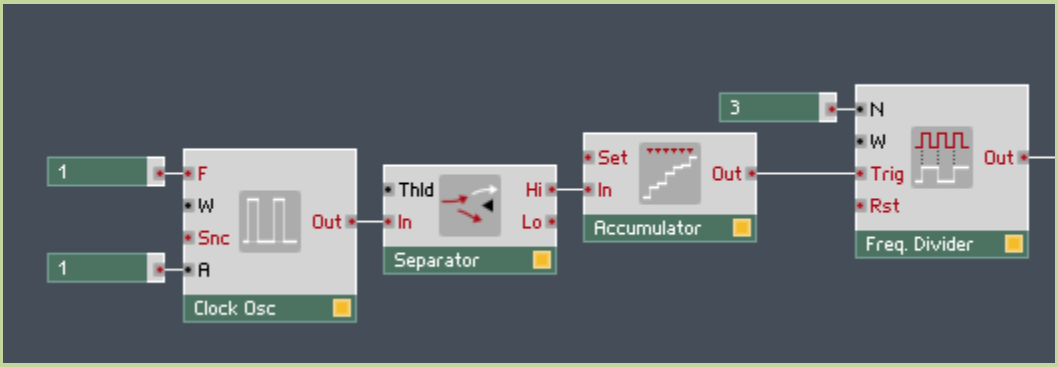


Модуль Recify/Sign собраный из других модулей

Frequency Divider – частотный делитель.

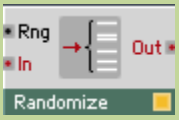


Trig пропускает только положительные входящие сигналы.

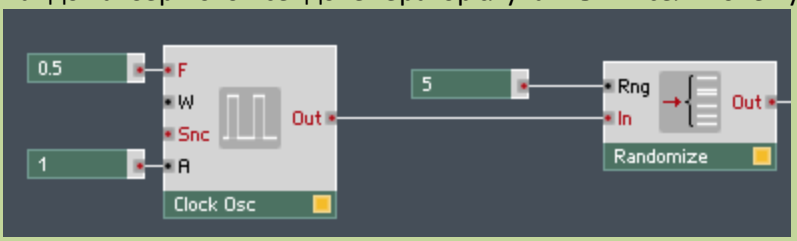


Выход с аккамулятора будет таким : 1,2,3,4,5,6,7,8,9,10, и т.д
Выход с Freq.Divider будет 0,0,**3**,0,0,**6**,0,0,**9**,0 , то есть будет пропускать каждый третий сигнал, согласно входу N, остальные числа будут нулями.

Randomize – Рандомайзер



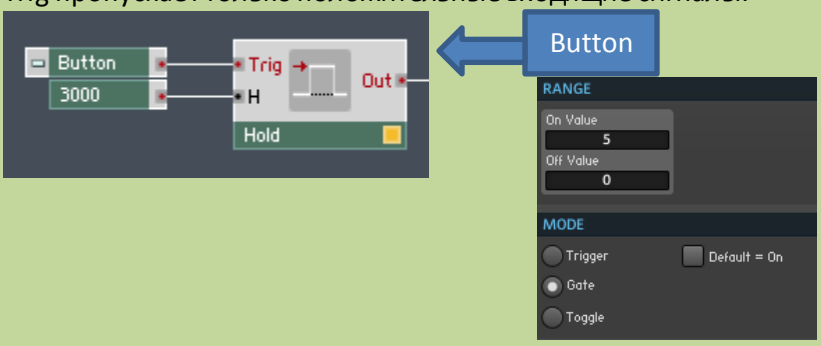
Рандомайзер – это псевдо генератор случайных чисел. Почему псевдо ? Если создать такую схему ,



сохранить, выйти и снова открыть, то каждый раз открывая файл вы увидите одинаковую последовательность, например 0,234 – 1,345 – 2,037 и т.д. Randomize хорош как генератор случайных чисел когда вы сами что-то нажимаете, ведь никто не знает когда вы нажмёте клавишу, и поэтому это будет звучать как случайные числа, но на самом деле это псевдо генератор как показывалось выше. Вход Rng – Range – рамки, допустим = 5, то генерация будет от -5 до 5.

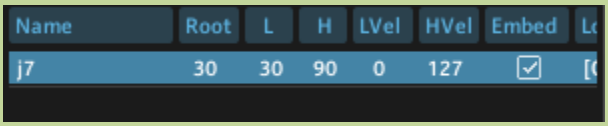
Hold – задержка сигнала

Trig пропускает только положительные входящие сигналы.

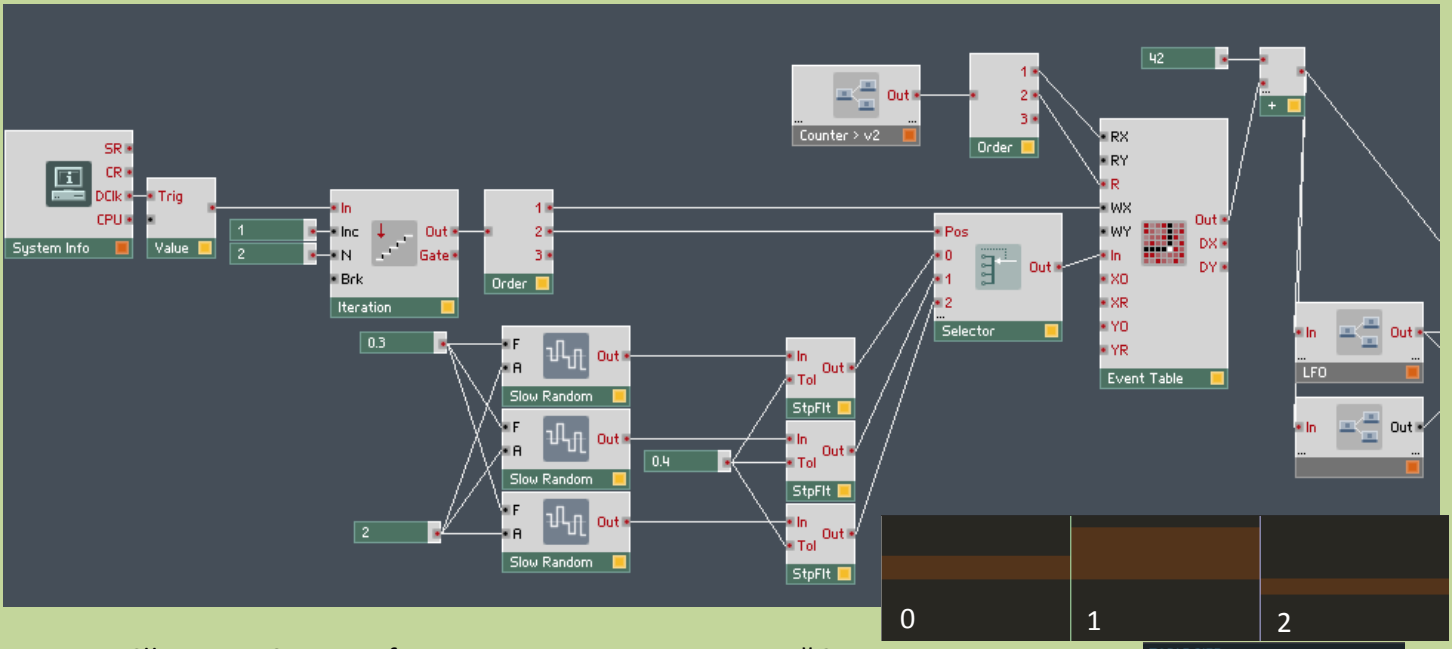


Если нажать Button то Hold будет держать сигнал=5 три секунды, по истечению трёх секунд Hold выдаст 0.
Вход H задётся в миллисекундах.

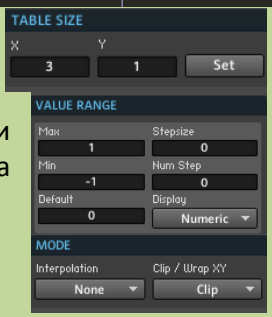
Чтобы сэмплы сохранялись в ensemble нужно напротив сэмпла ставить галочку Embed



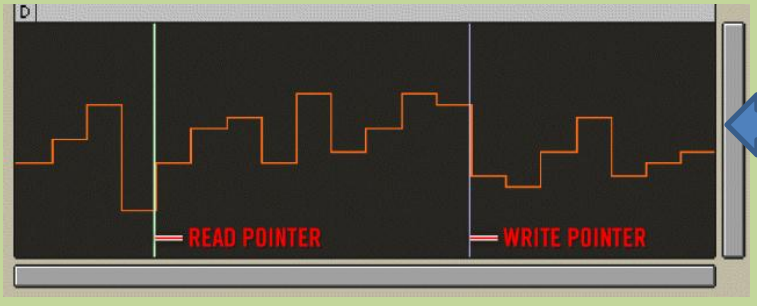
Записываем и считываем значения для Event Table



Выход с DClk модуля System Info генерирует еденицу с частотой 25 Герц.
Далее сигнал поступает на Value с нулём чтобы итерация начиналась с нуля.
Итерация генерирует 0,1,2 и состоит из двух шагов с приращением еденицы для того чтобы сначала подать на WX модуля Event Table, затем обсчитать Selector с тремя входами и подать на вход In модуля Event Table. (У модуля Event Table сначала выберается - 1. куда подать – вход WX. 2. Что подать – вход In). Таблица по WX начинается с нуля (1-нулевая ячейка.2-первая ячейка и т.д).



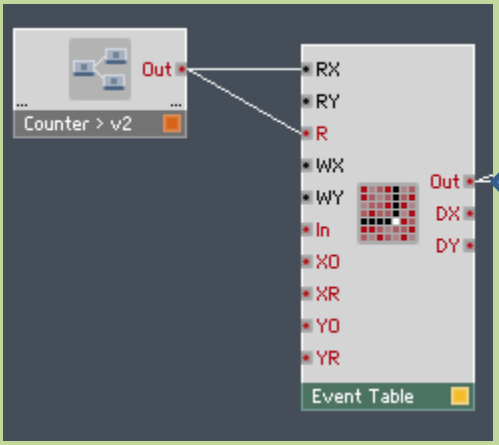
Режим MODE у Event Table имеет два значения : Clip и Wrap. Clip – означает, что если таблица в настройках Table Size имеет например X=3 , а на вход WX допустим пришло 4, то считывание будет производиться всёравно из 2-ой ячейки. Wrap - означает, что если таблица в настройках Table Size имеет например X=3 , а на вход WX допустим пришло 4, то считывание будет производиться из 1-ой ячейки, так как мыслено Wrap это то же самое что Modulo у которого вход B это X-Table Size.
Настройка Graph=Line&Fill
В итоге получается что мы с частотой 25 Hz , то есть 25 раз в секунду записываем в Event Table. Спрашивается зачем так часто записывать ? Если посмотреть на схему, то увидем что Slow Random генерирует числа с частотой 0,3 Гц, поэтому обсчитывание надо производить с такой же или большей частотой. Например чтобы производить обсчитывание модуля LFO который работает на частоте Control Rate 400 Гц, выход у System Info надо брать CR который работает на этой же частоте.
Чтобы был выход с Out Event Table надо считывать с RX, R с помощью какого-нибудь счётчика RX – это выбор ячейки, а R – это триггер.



Зелёная вертикальная линия – это RX считывание, серая вертикальная линия – это WX записывание.

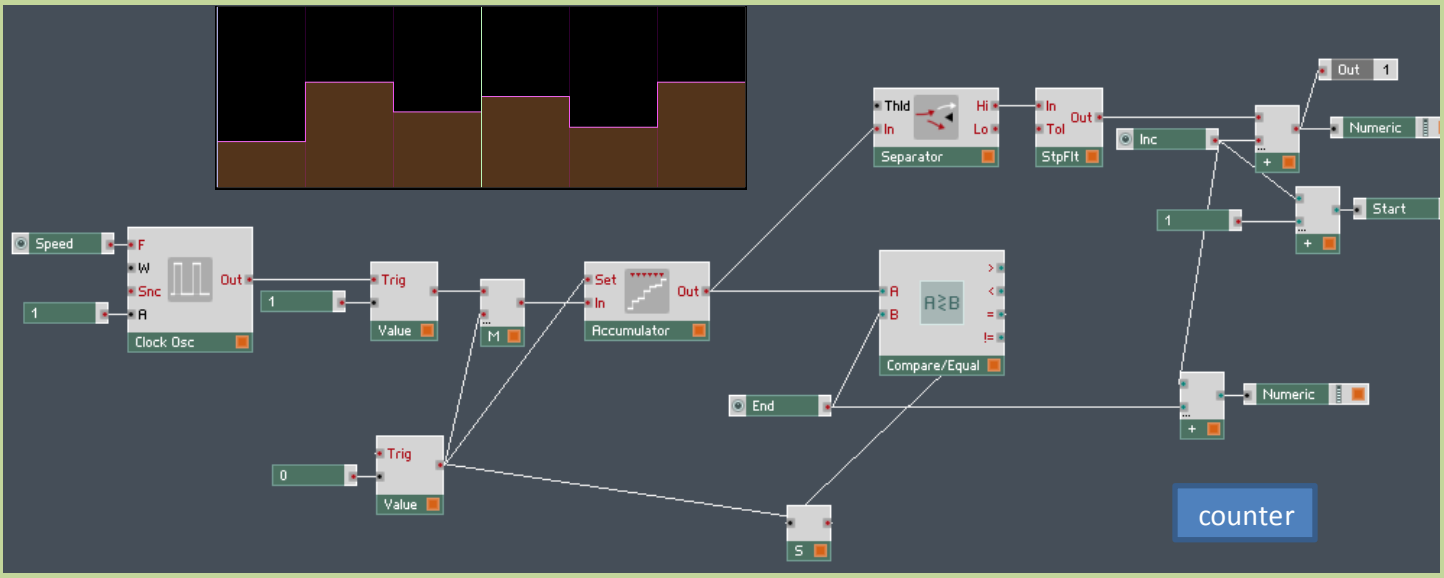
Используем RY и YO у Event Table

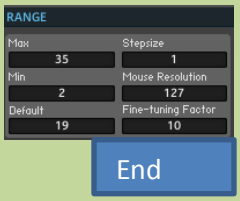
Допустим у нас есть счётчик-counter который считывает по RX, допустим он считает до 0 до 5. Следовательно у Event Table Table Size по X должен быть 6, Table Table Size по Y=1.

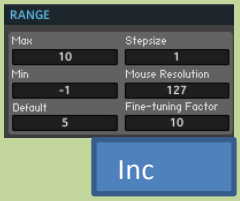


Нажмём по Event Table правой клавишей мыши и выберем Table Draw Mode, тем самым сможем рисовать значения по Y-Value Range.

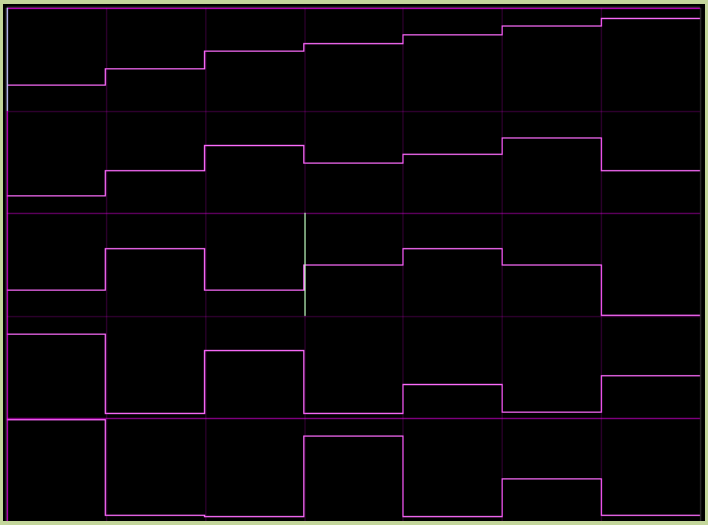
Graph= Line&Fill



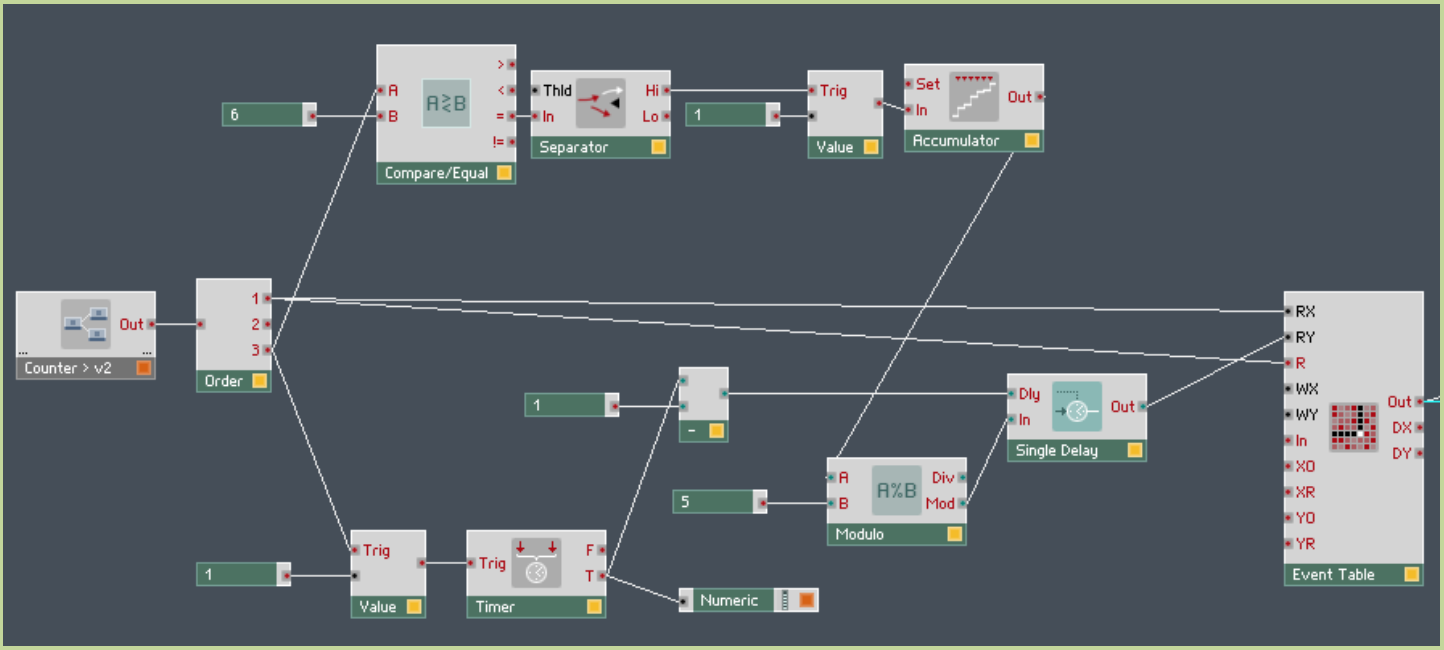




Спрашивается : если взять Table Size с Y=5, X=6 то как автоматически последовательно обсчитывать такую таблицу с несколькими строками по горизонтали и вертикале ?



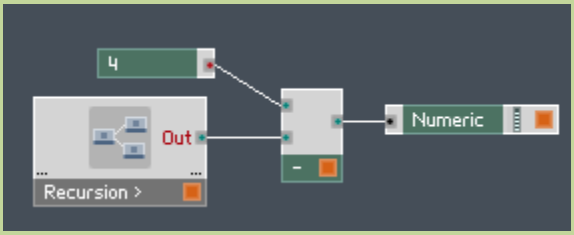
Во-первых в опциях надо выбрать Graph= Multi Line

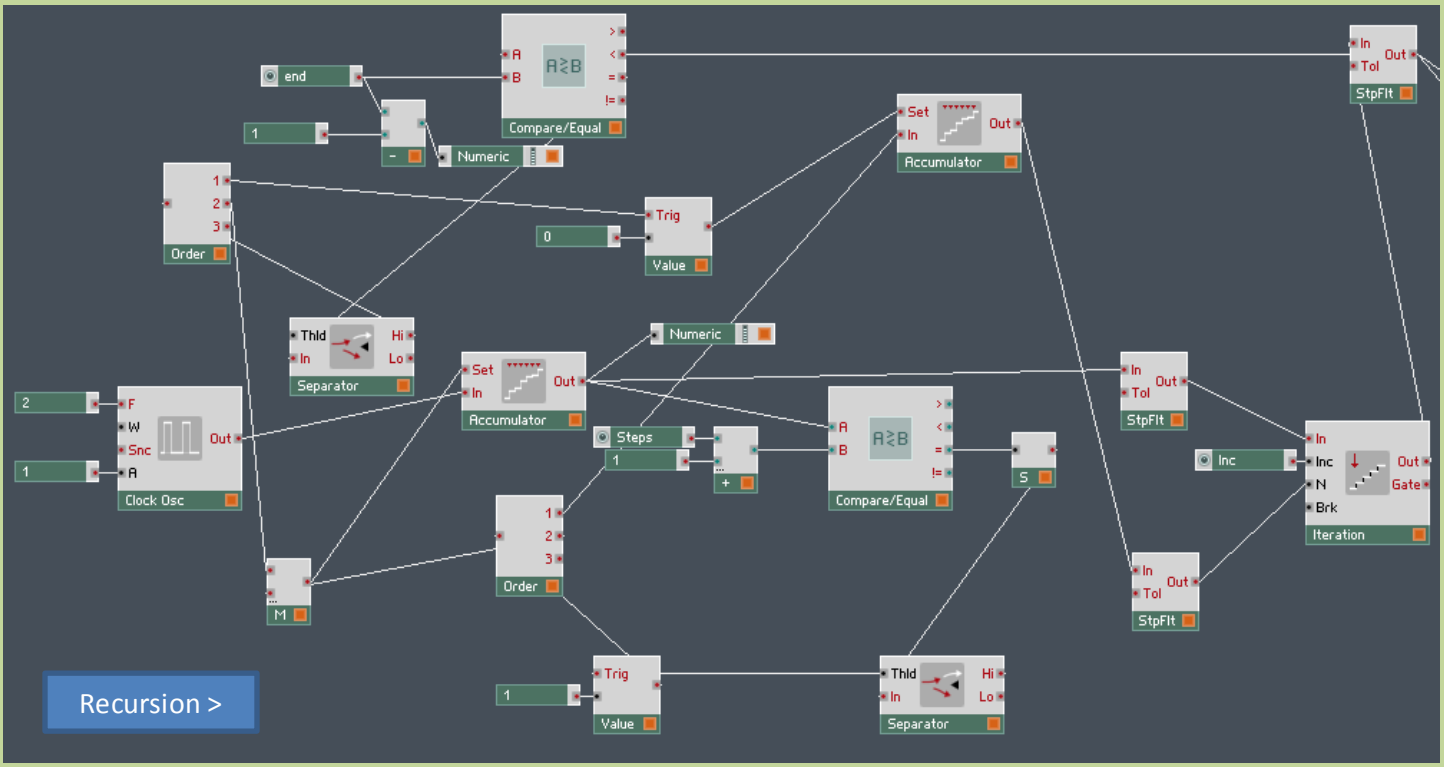


Timer здесь нужен для того чтобы определить когда нужно передвинуть каретку–перейти на следующую строку, это происходит буквально за 1 миллисекунду до того, когда начнётся следующее считывание по RX в новой строке.

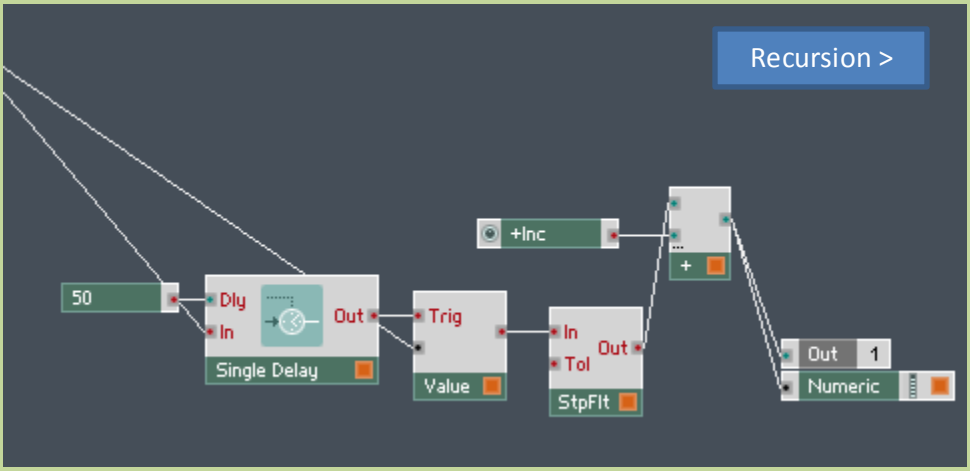
Если мы сейчас подключим выход от Single Delay к YO, то увидим что ничего не изменилось. Чтобы RY и YO работали вместе надо выбрать в настройках Graph= Line&Fill – будет отображаться одна строка, затем выбрать в настройках Table Table Size по Y=8 допустим. Получается что мы переключая одновременно RY YO будем видеть не несколько строк по Y , а одну строку, но она будет меняться при переключении. Это хорошо при ручном переключении и можно использовать как Банк данных.

Рекурсионый счётчик с обратной последовательностью 4,3,2,3,2,1,2,1,0.



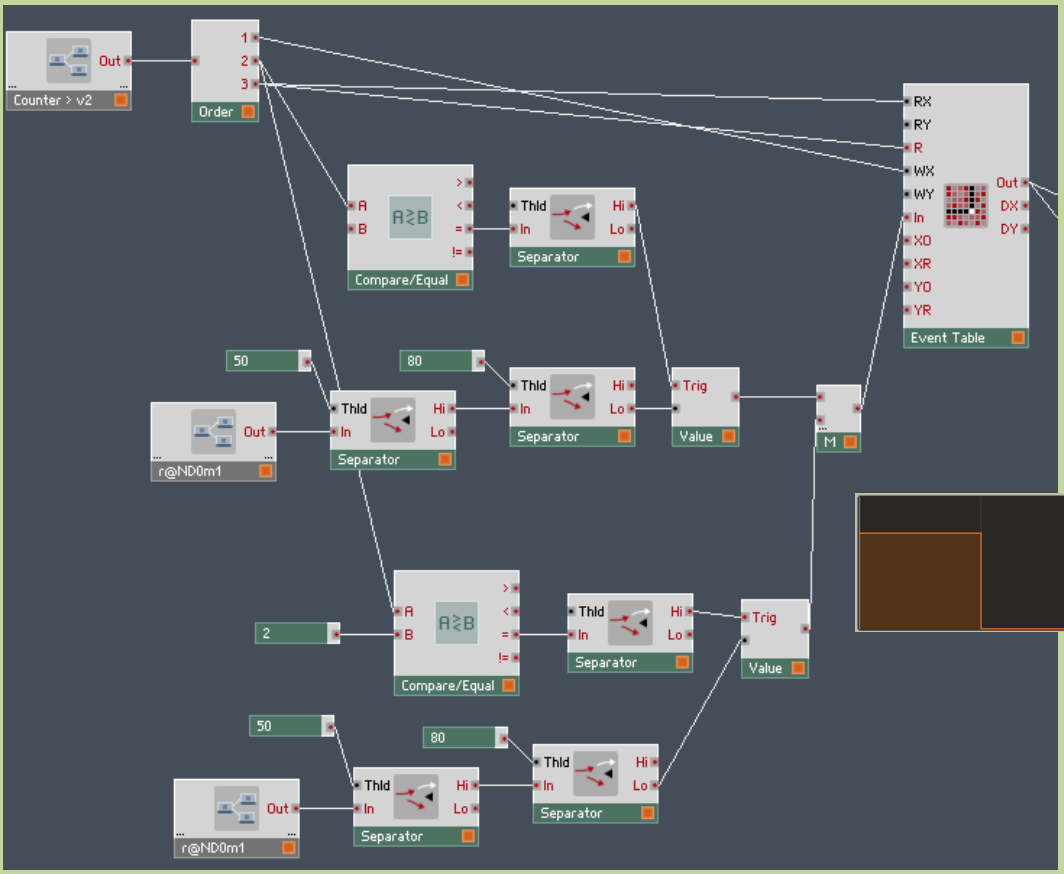


(продолжение схемы) выход со Step Filter



Многие схемы довольно трудно обьяснить, так как долго описывать их работу-логику.

Запись случайных чисел в определёные ячейки Event Table

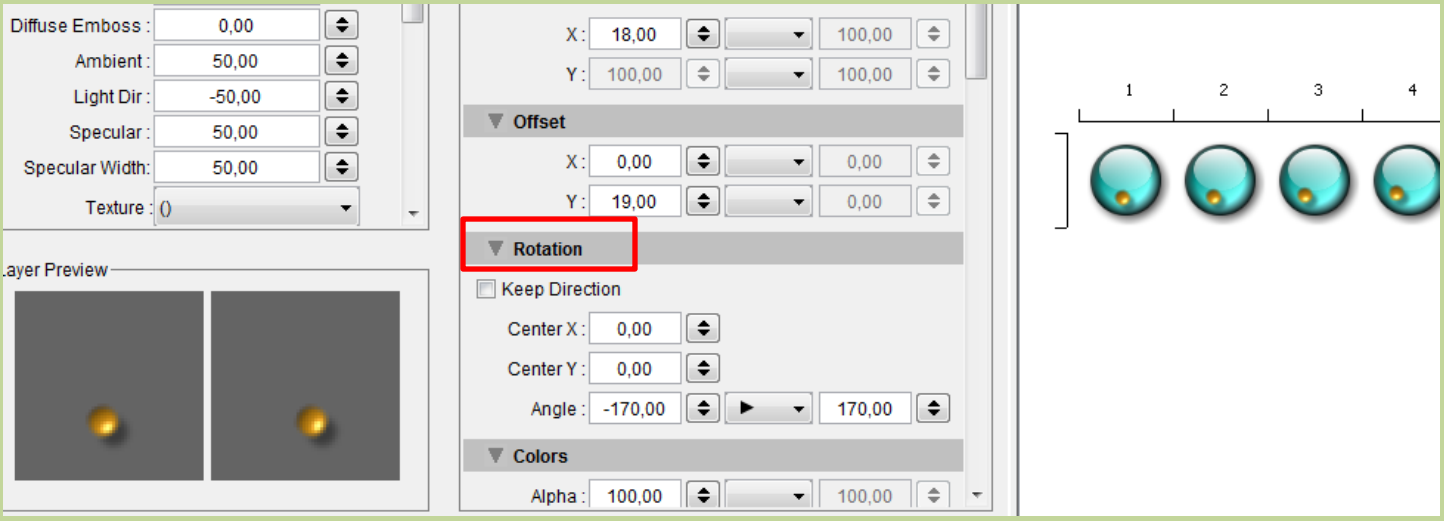


Запись будет вестись в нулевую и вторую ячейку

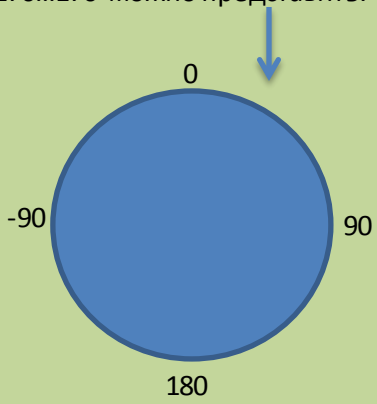
Создаём Knobs, Meters.

С помощью программы JknobMan можно создавать разные knobs, meters, и т.д

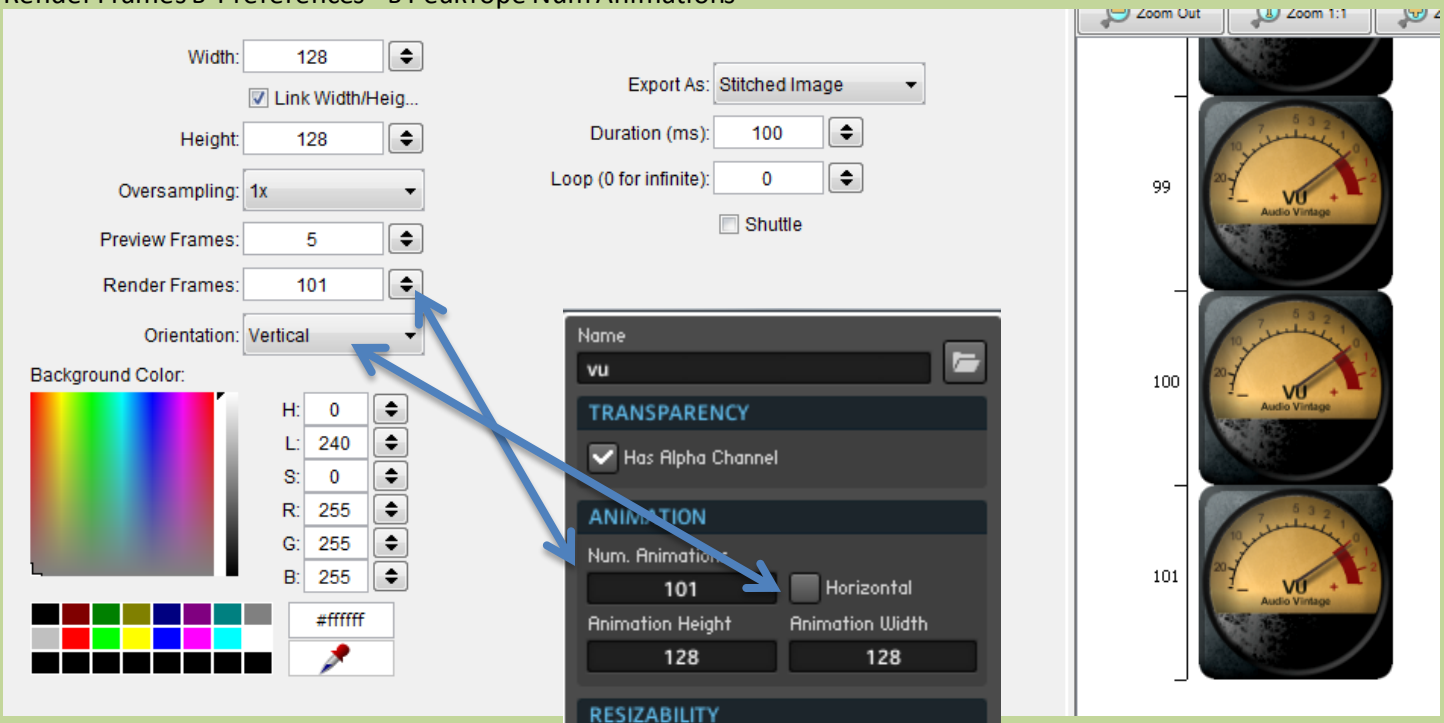
Настройка кручения задаётся – Rotation (например от -170... до 170)

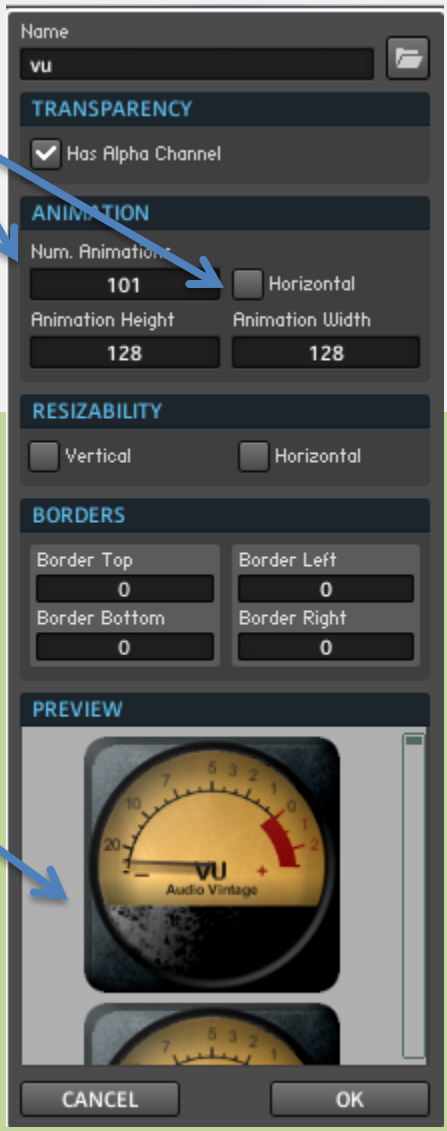


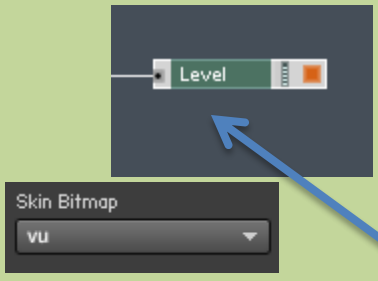
-170...170 можно представить.



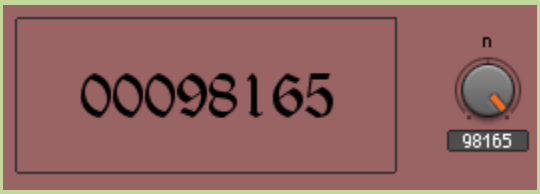
Render Frames в Preferences – в Реакторе Num Animations



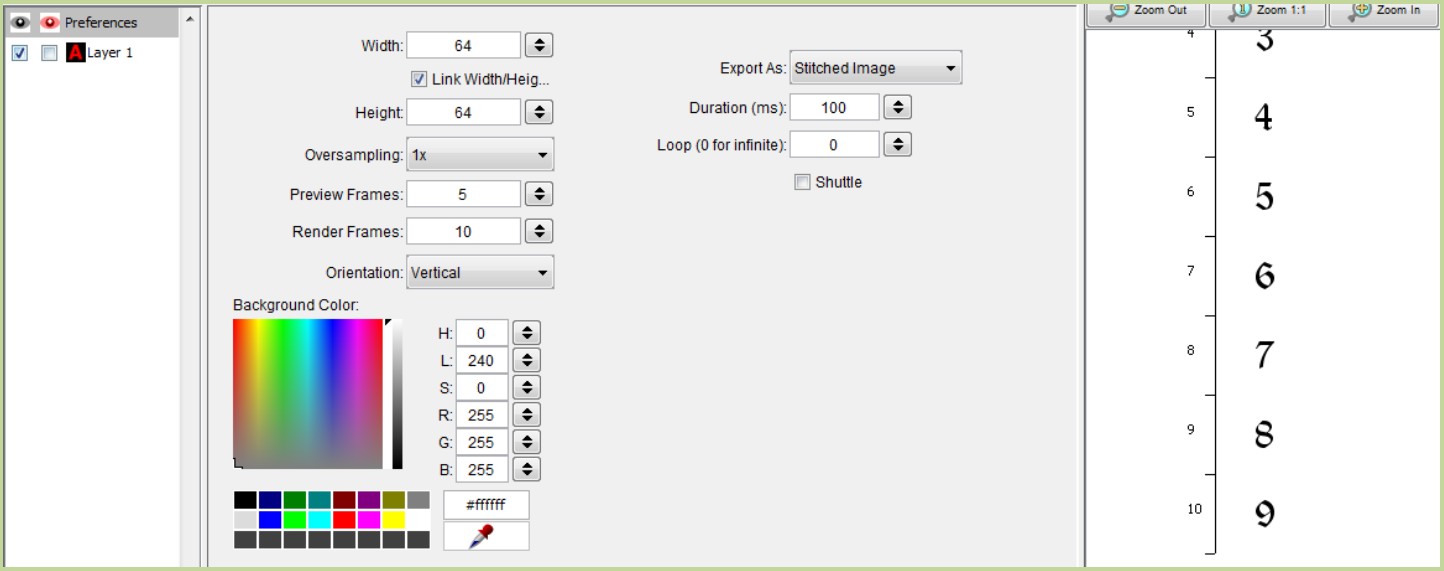




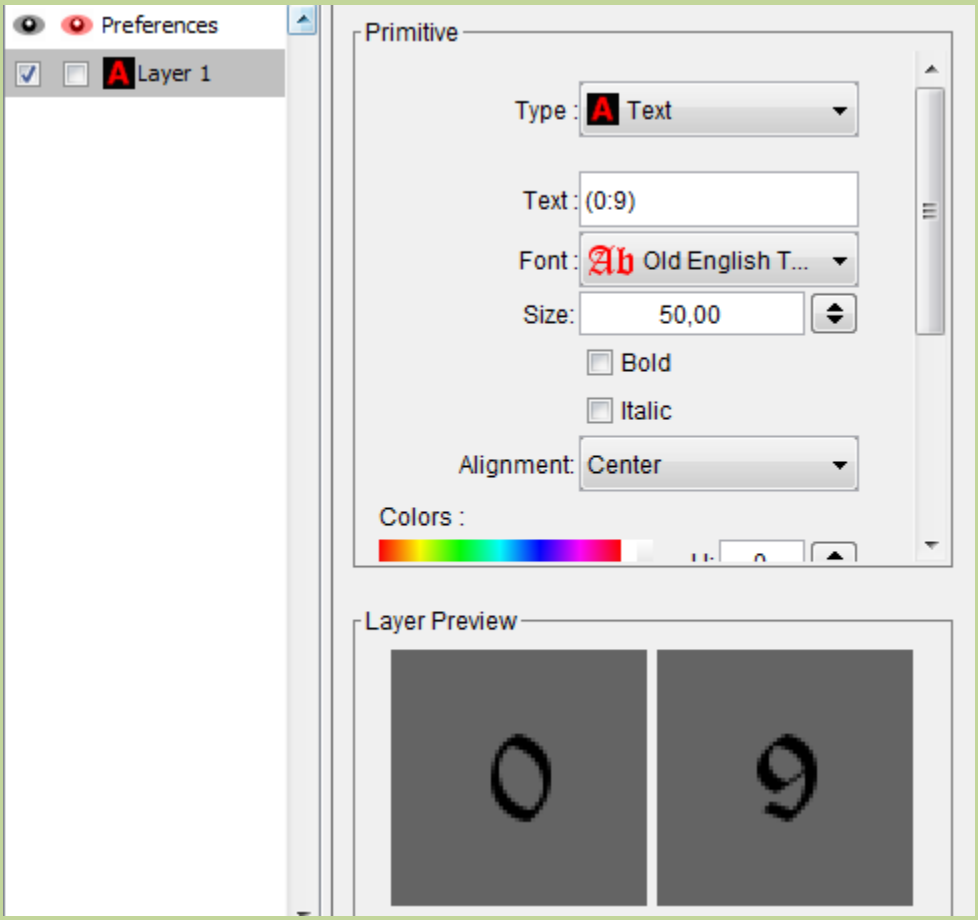
Создаём Numeric Readout с помощью JknobMan

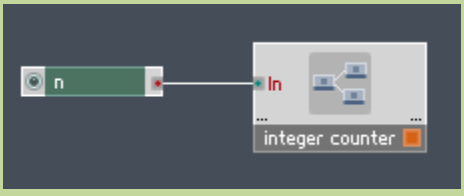


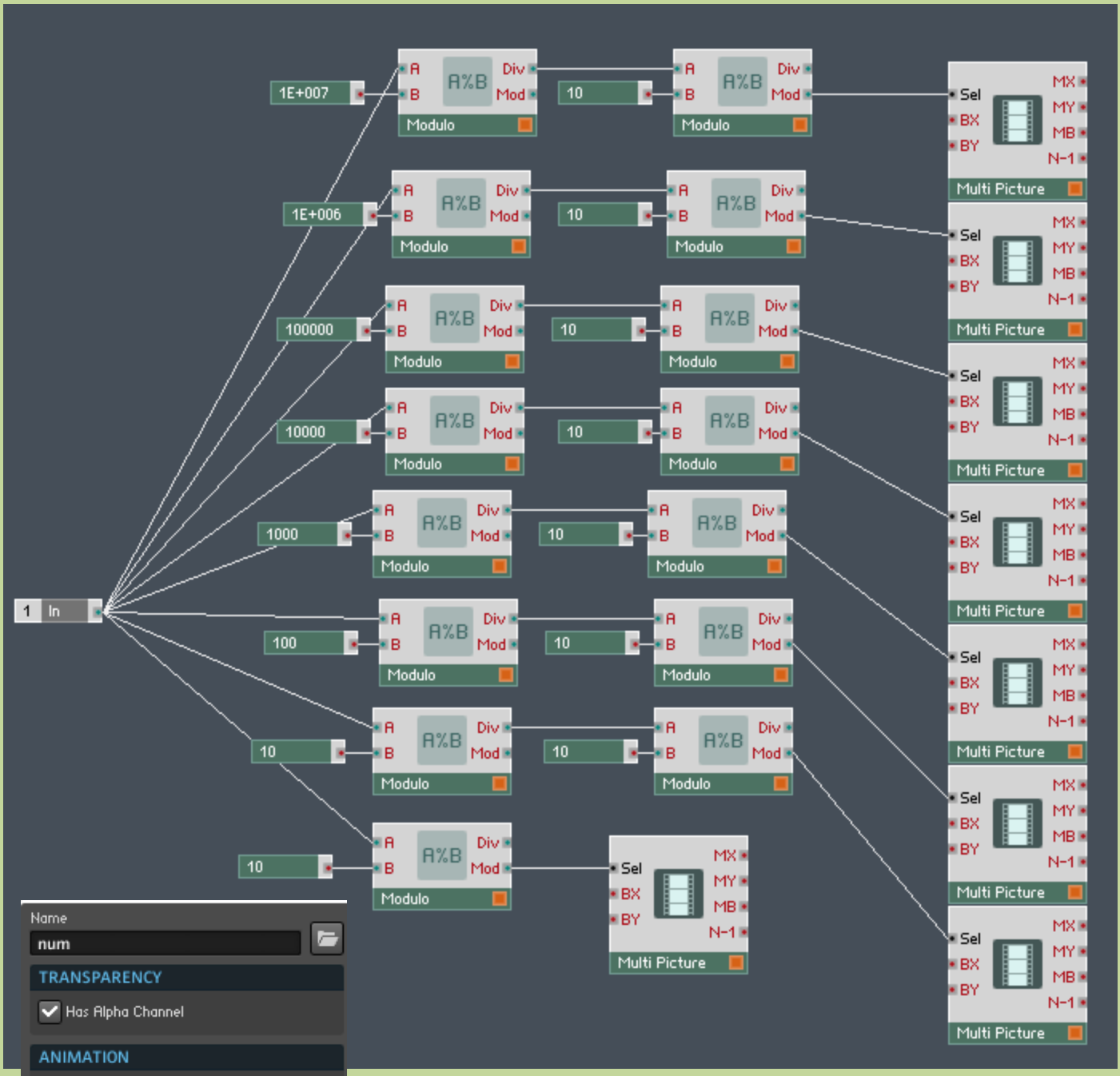
1.



2.







Name

num

TRANSPARENCY

Has Alpha Channel

ANIMATION

Num. Animations

10

Horizontal

Animation Height

64

Animation Width

64

RESIZABILITY

Vertical

Horizontal

BORDERS

Border Top

0

Border Left

0

Border Bottom

0

Border Right

0

PREVIEW

0 1

2 3

PICTURE

Select Picture

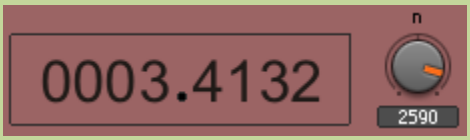
dig

Max Index

9

Multi PictureProperties

Numeric с десятичной частью – до десятитысячной



Input	log(2590)
Output	log(2590)
Decimal Output	3.4132997640813

Lamp служит точкой. Красным выделены цифры после точки

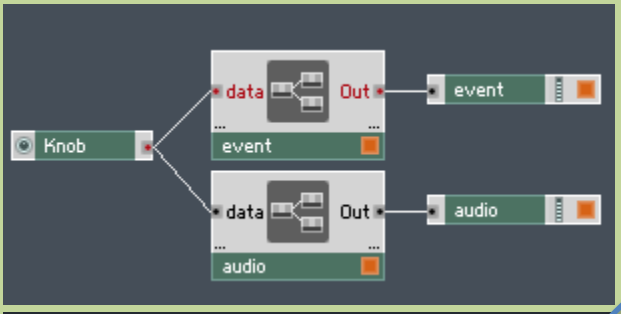
Core

Flavor	Înputs	Outputs	Clock Src
Event	Event	Event	Disabled
Audio	Event/Audio	Audio	Enabled

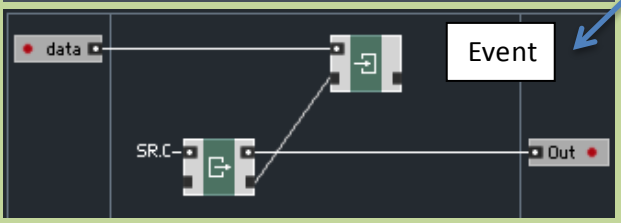
Core ячейки делятся на два типа : event и audio. В этой таблице показаны их характеристики.

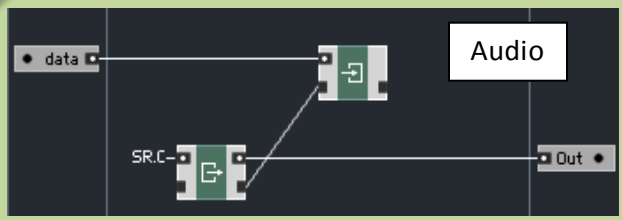
Иерархия Core : Cells -> Macro -> Macro. Внутри них находятся modules.

В Event ячейках-макросах не работает Clock Src. Clock Src это Control Rate по умолчанию = 400 Hz.

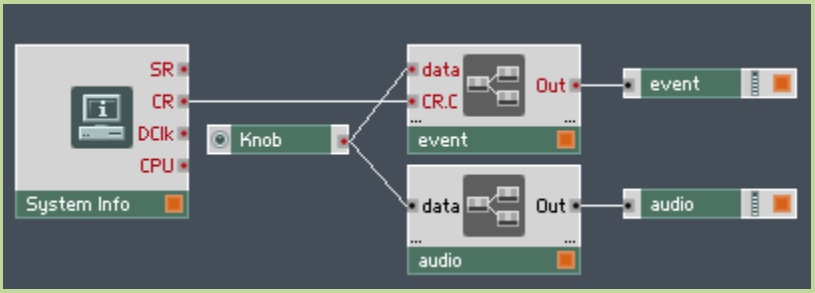


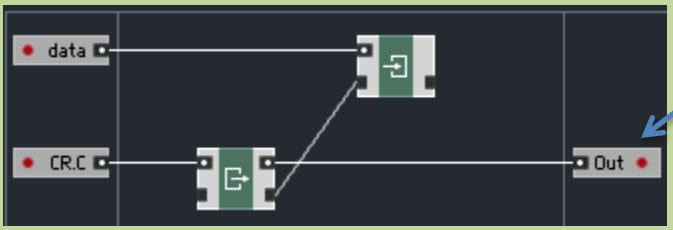
Как видно event не работает





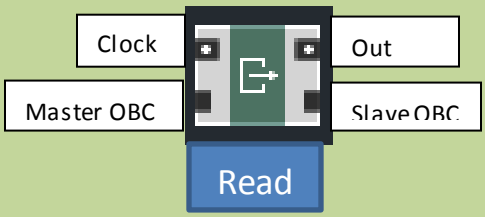
Чтобы Event ячейки-макросы работали с CR.C надо подать его с первичного уровня :

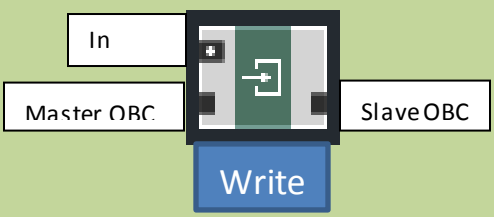




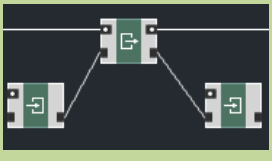
SR.R в Core это SR на первичном уровне, является константой по умолчанию равной 44100 Гц.

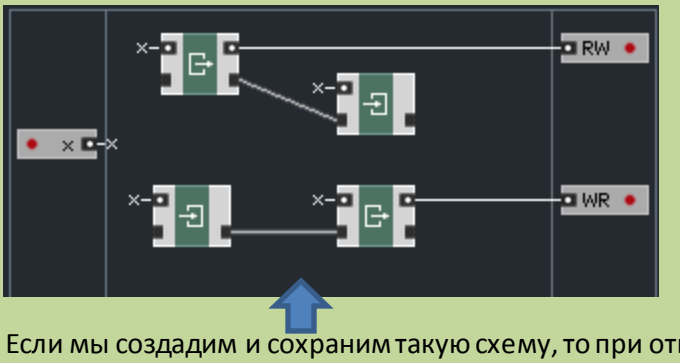
Модули Read и Write.





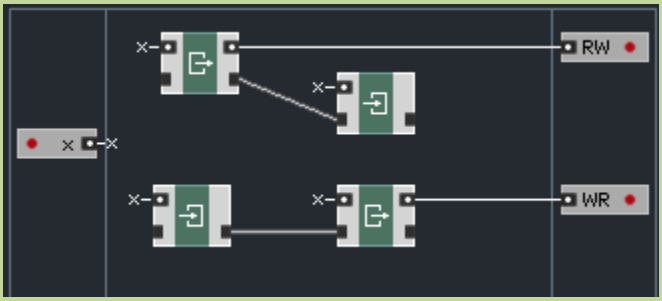
Для чего нужны входы-выходы OBC ?
OBC связь это разделёная память между Write и Read модулями .Если у Read оба OBC входа подключены, то при старте первым считывается Write-Slave.





Если мы создадим и сохраним такую схему, то при открытии мы увидим что хоть Button и послал единицу в качестве синхросигнала (Clock) и самого значения, но на выходе WR есть единица, а на выходе RW ничего нет. Почему так ?

Здесь главное запомнить, что при инициализации память ОВС находится в нулевом состоянии, и так как у WR Write-Slave то он обработался первым и послал значение на Master, то у RW Write-Master пошлёт сигнал только после второго клок сигнала.



после второго Clock

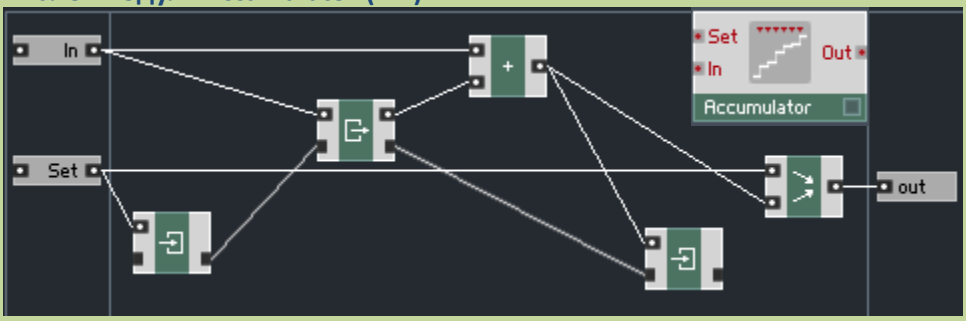
после первого Clock

Если после открытия нажимать Button то получим следующюю схему:

	Открытие	Clock 1	Clock2
WR	1	2	1
RW	0	1	2

Нажимая Button-Clock можно заметить что RW запаздывает на один Clock. Такой принцип называется Z-1. Его можно использовать в схемах с обратной связью(когда выход подаётся обратно на вход) :

Аналог модуля Accamulator (Z-1)



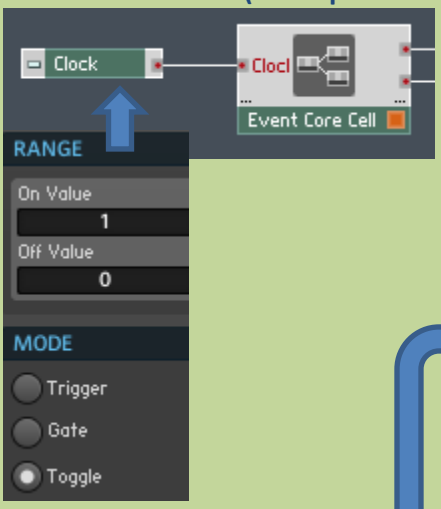
Верхний вход аккамулятора Set на первичном уровне имеет приоретет при инициализации(старте), если при старте на Set пришла 7,а на In 3, то Out будет их суммой = 10, поэтому в коре используется такая схема с предустановкой (Write-Slave это Set на первичном уровне, при старте происходит суммирование предустановки с In входом) ,merge – при старте merge посылает сигнал с самого нижнего выхода.

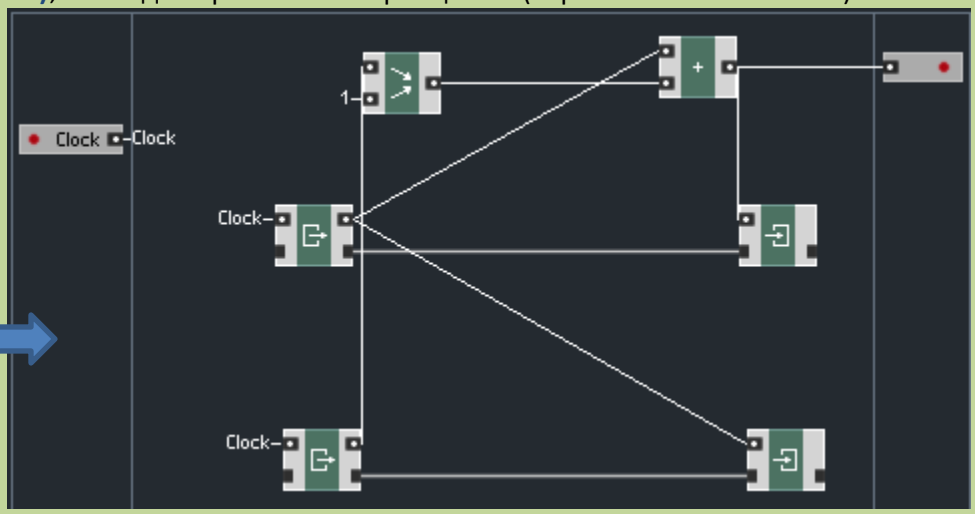
Уместно заранее сказать что clock сигнал можно разделить на 2 типа : пошаговый (например когда Button выступает в качестве clock или сам clock не изменяется быстро), мгновеный (CR.C- 400 значений в секунду) когда трудно увидеть изменения при инициализации-старте.

Если же мы попытаемся подключить как в варианте WR, то получим ошибку :

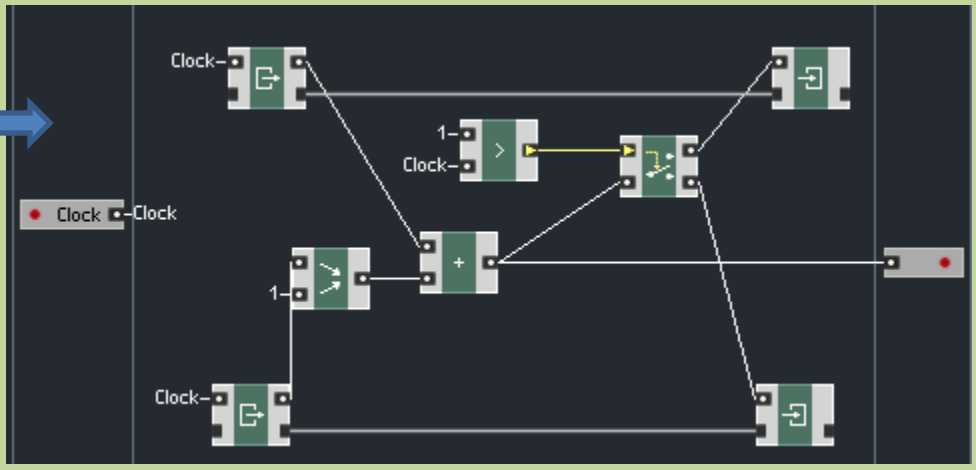
Буква Z показывает что происходит обратная связь, но без задержки в один клок или сэмпл.

Числа Фибоначи – (без сброса в ноль), как видно применяется принцип Z-1(вариант RW – Read-Slave)

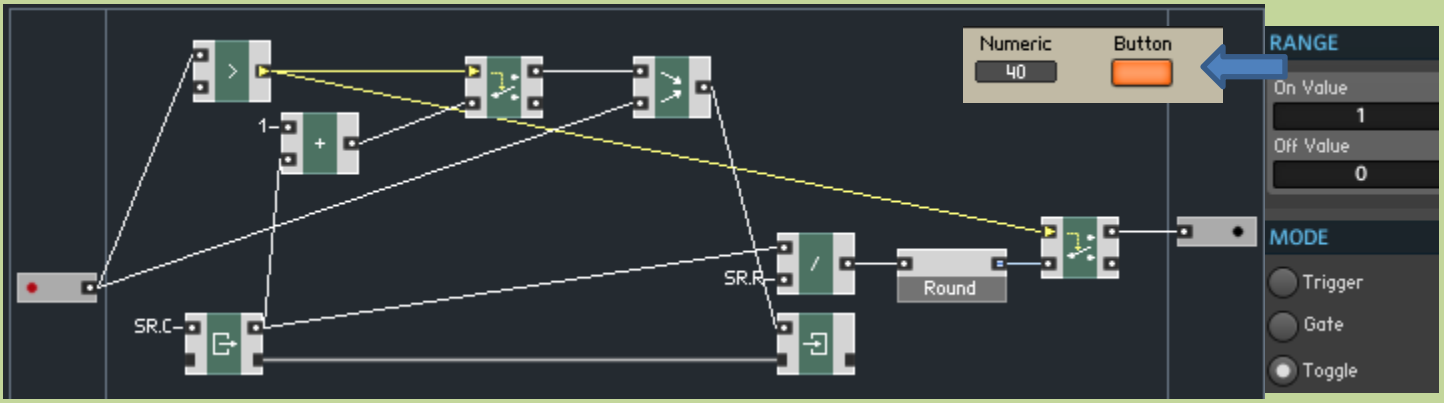




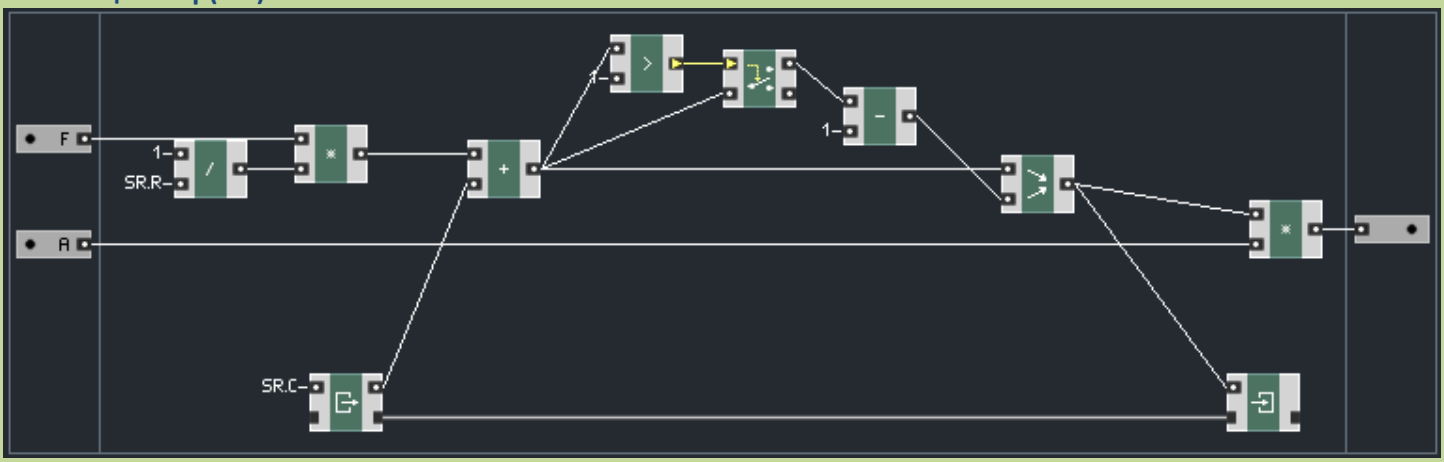
Числа Фибоначе – каждое число есть сумма двух предыдущих : 1,1,2,3,5,8,13,21,34,55 и т.д, если делить следующее на предыдущее то будет получаться всё более и более точная золотая пропорция 1,61803



Timer(Z-1)

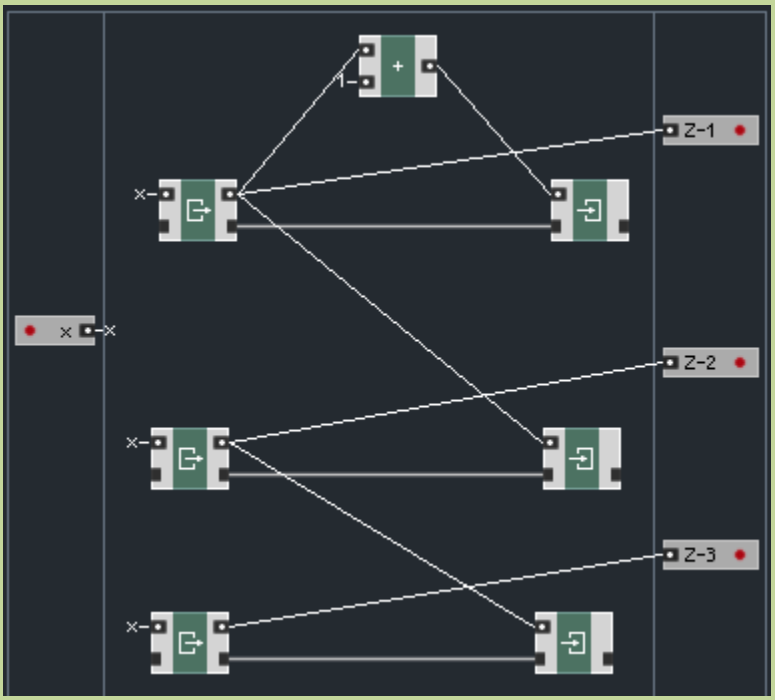


Saw осцилятор(Z-1)

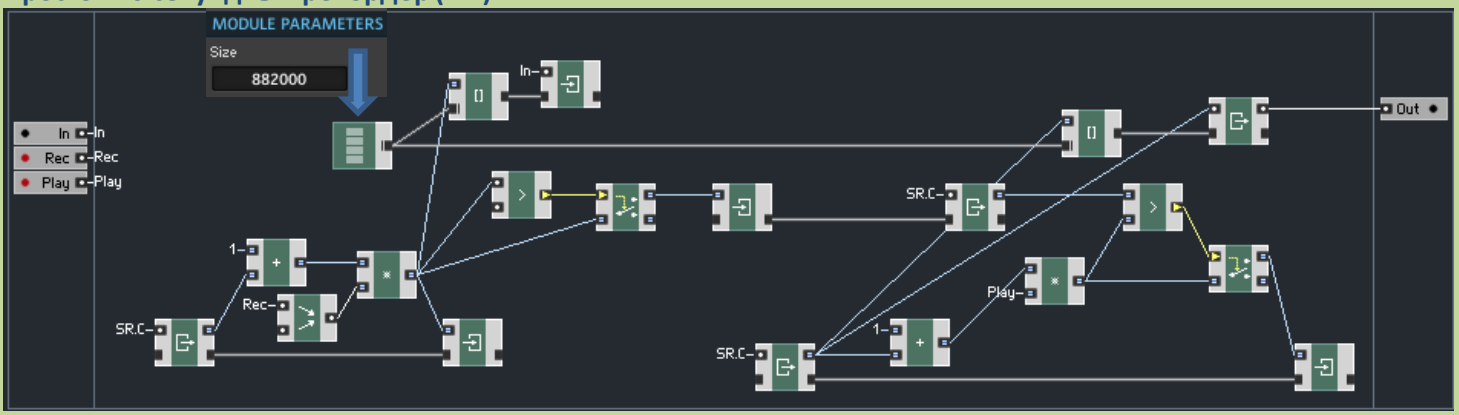


Также встречается макрос Latch (защёлка)  – этот макрос напоминает модуль Value – верхний вход для сигнала, а нижний для Clock, пока Clock не нажмётся – сигнал не появится на выходе, то есть Clock защёлкивает сигнал. Он работает по принципу WR – Read(Master).

Цепочка Z-1, Z-2, Z-3

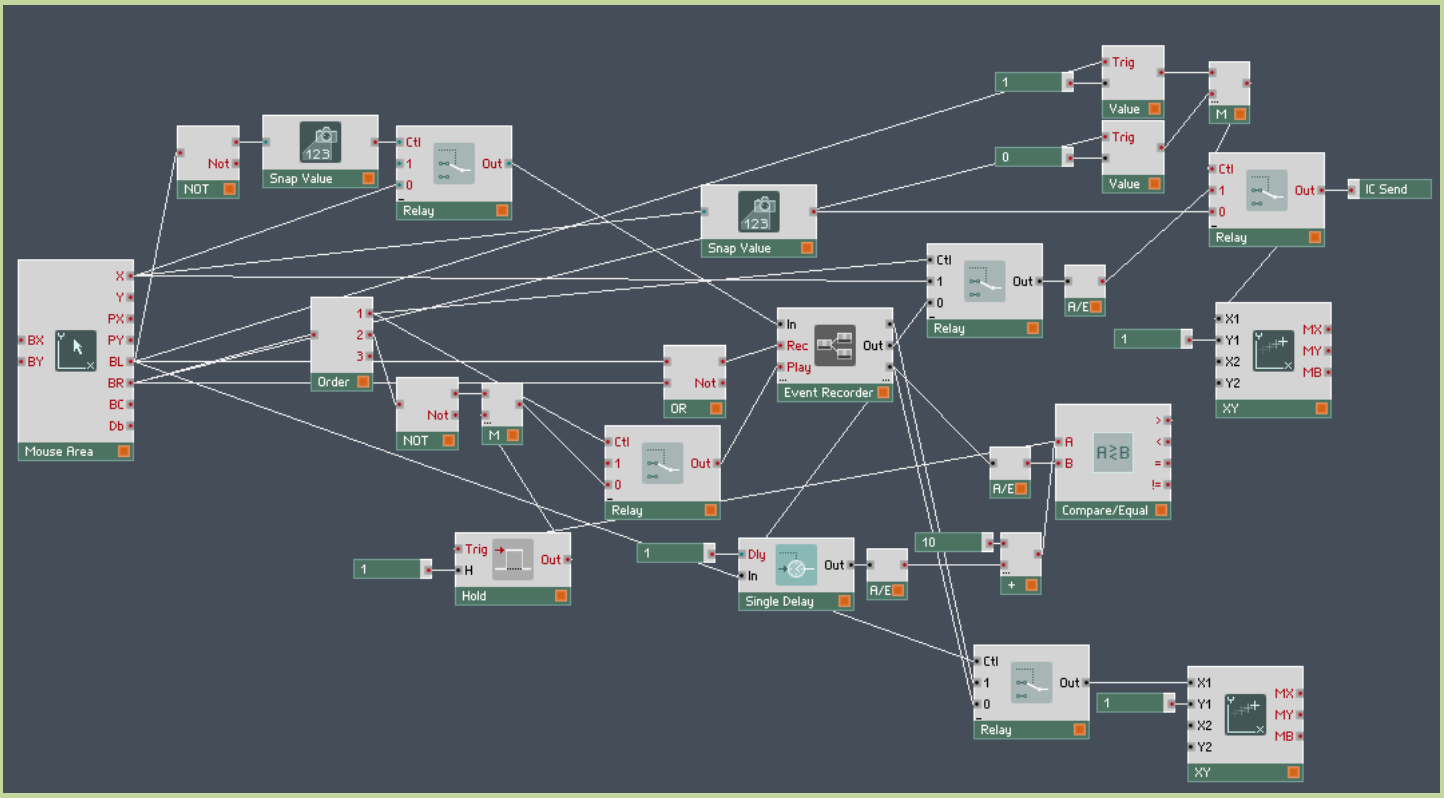


Простой 20 секундный рекордер (Z-1)



Rec, Play

Более сложная схема рекордера - контроллера



Числа Фибоначе (со сбросом в 0)

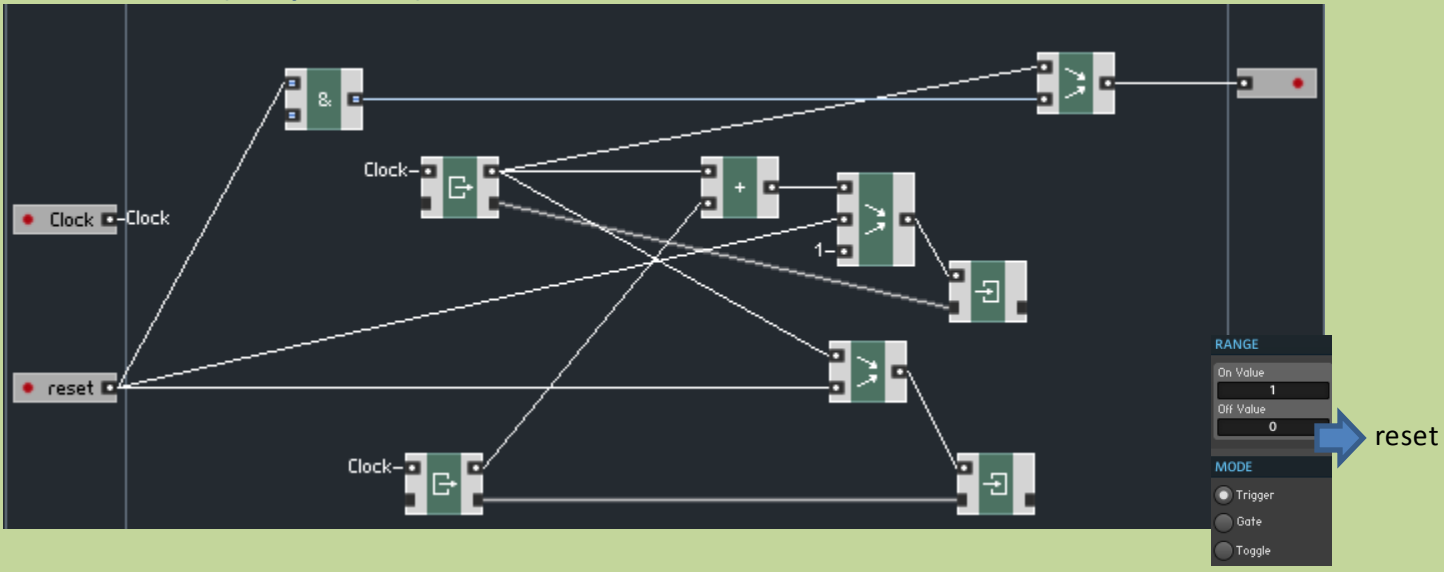
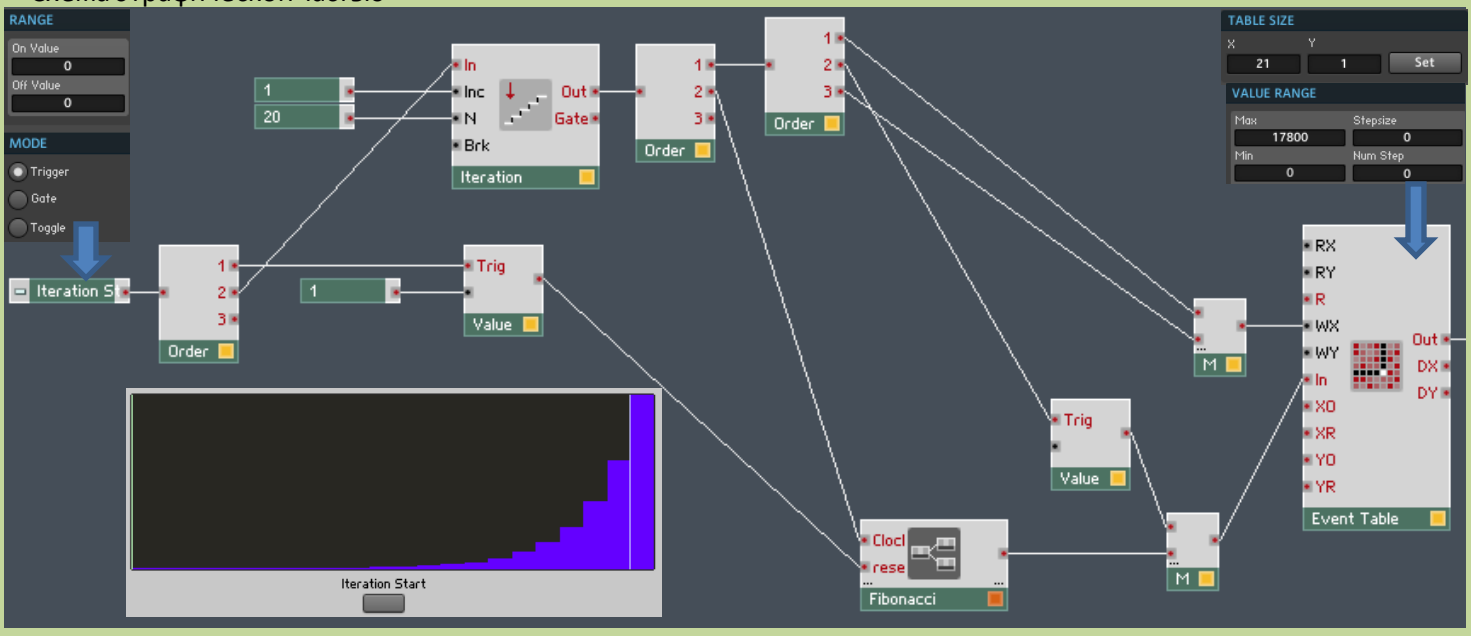
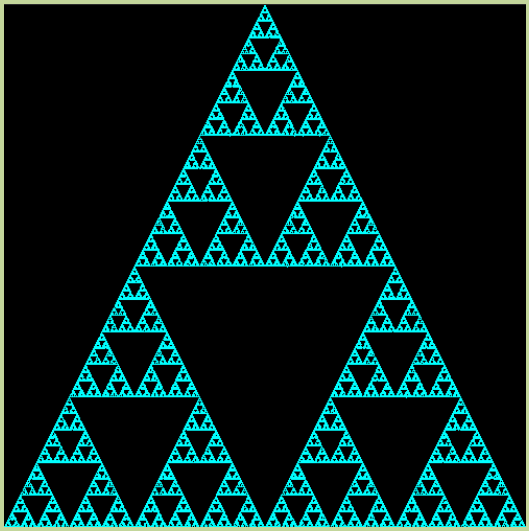
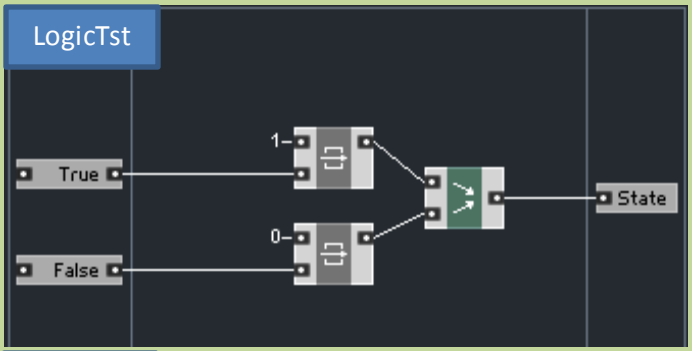


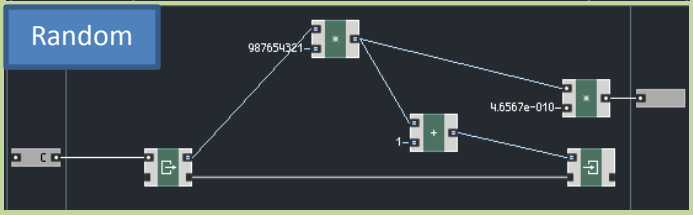
Схема с графической частью

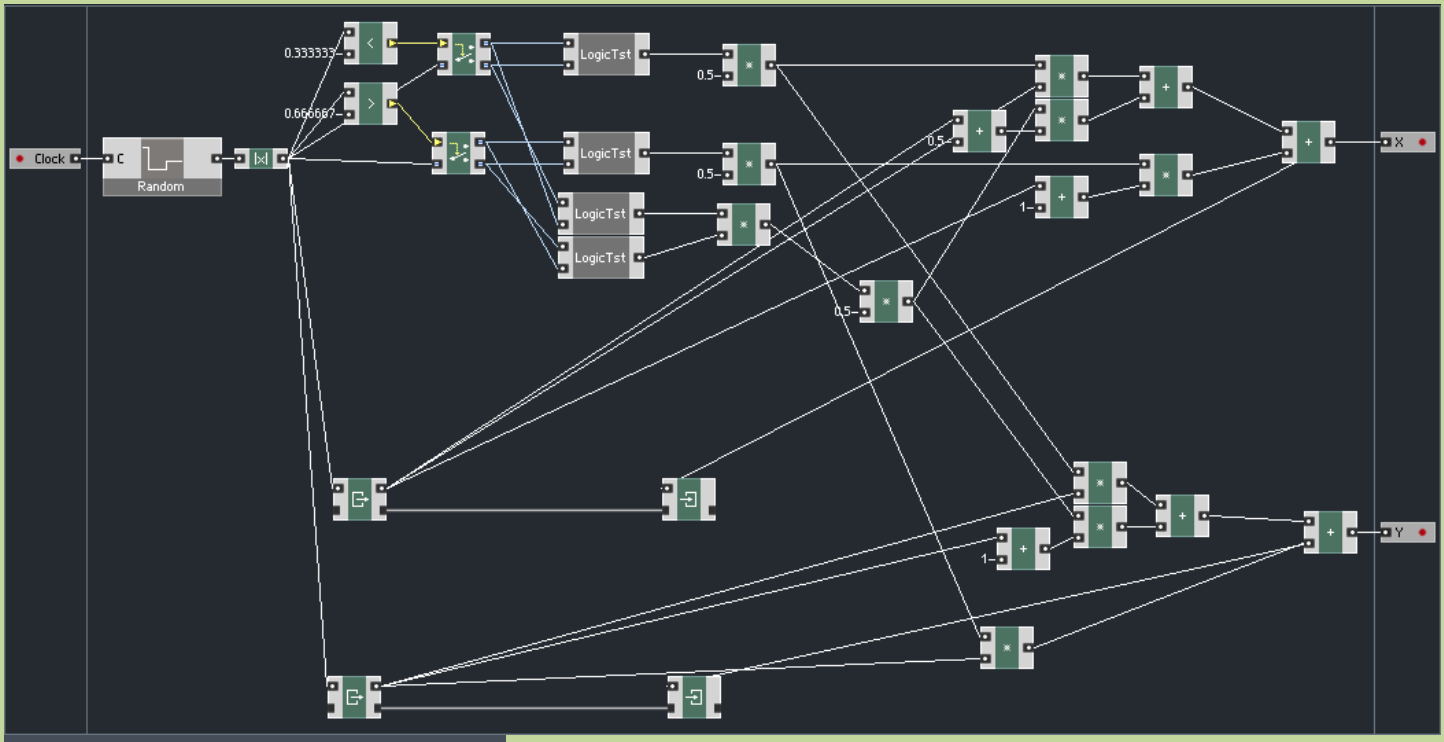


Фрактал – Треугольник Паскаля, Серпинского (Z-1)

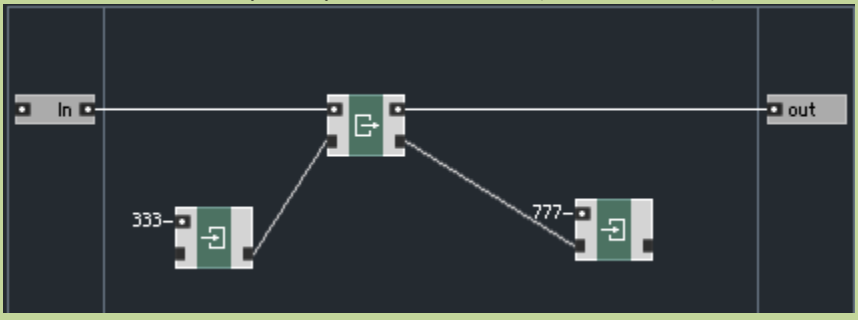






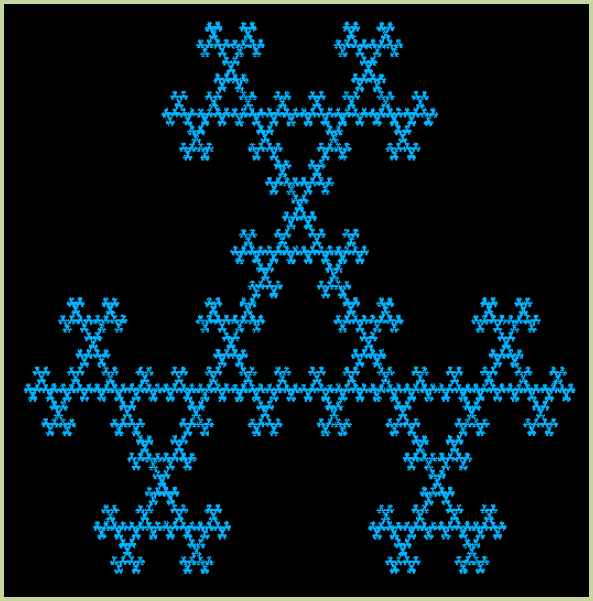


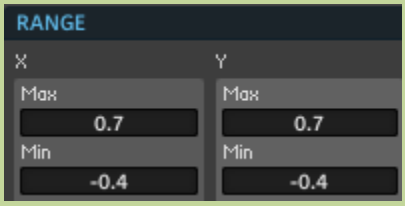
Что выдаст выход при старте такой схемы (event ячейка) ?



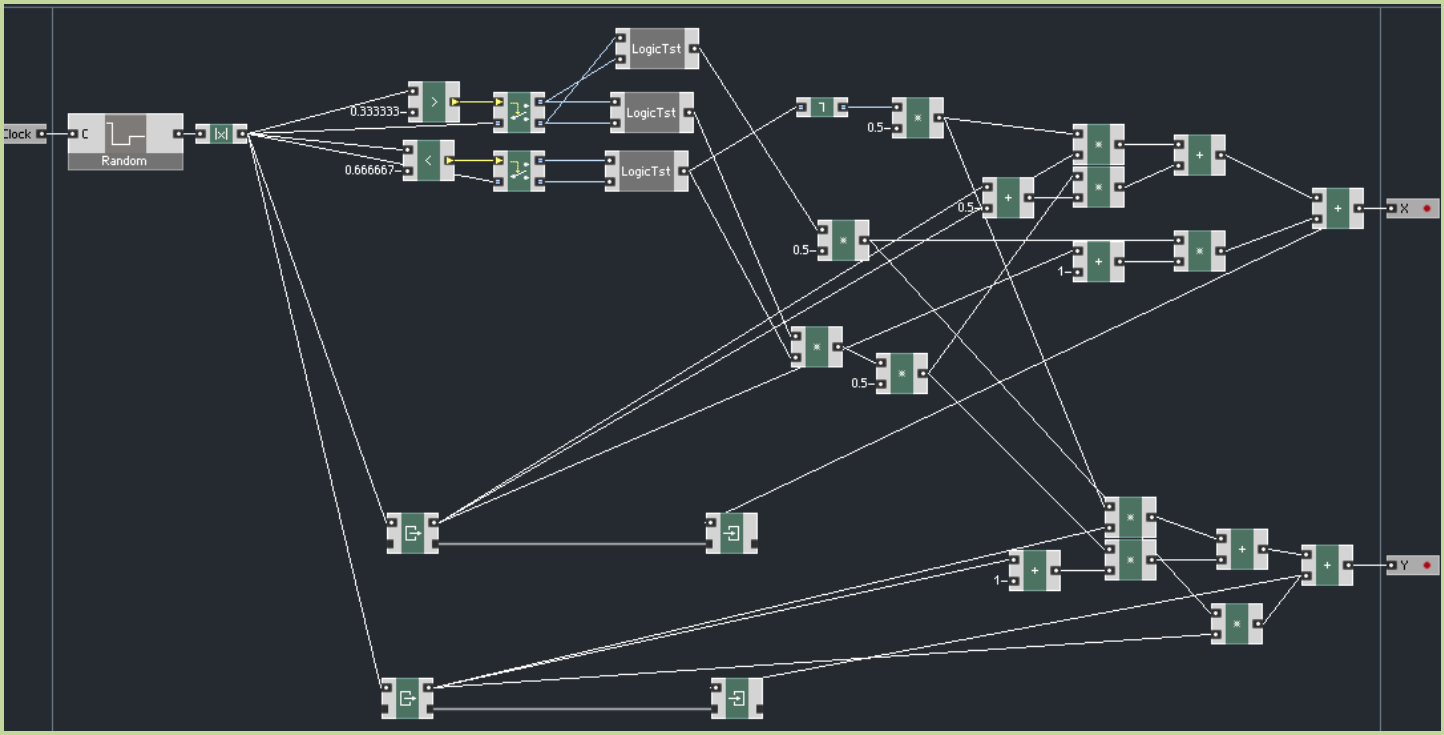
выход при старте выдаст 333, если дальше подавать сигнал на In то выход будет всегда идти с 777, пока не изменится 333. это можно использовать в качестве предустановки.

Самоподобный фрактал



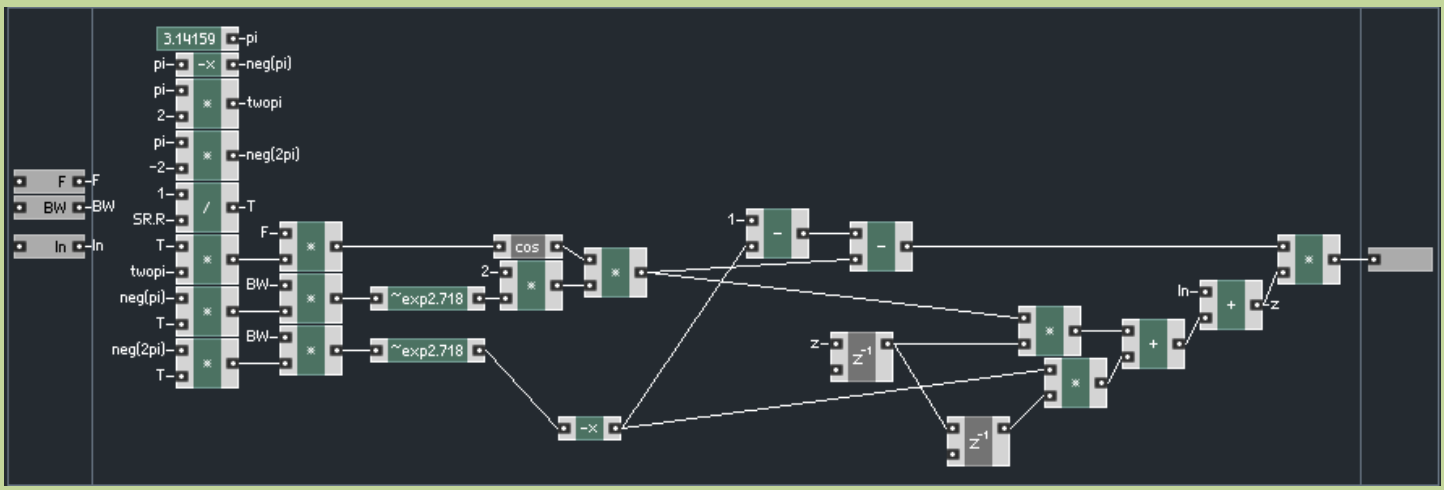


XY



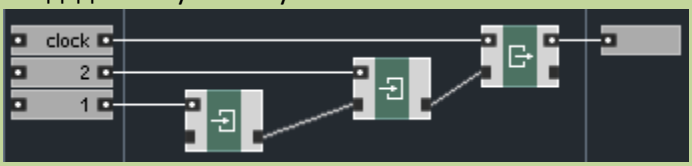
Не стоит также забывать что когда вы открываете core ячейку(заглядываете во внутрь), то данные в ней сбрасываются, для этого лучше использовать Horizontal Split – это когда экран поделён на две части. В нижней например коре-схема, а в вверхней части панель управления. И с помощью Wire Debugging вы сможете посмотреть в нижней части что происходит в схеме.

Резонансный аудио фильтр (Z-1), (F=0..5000, BW=0...7)



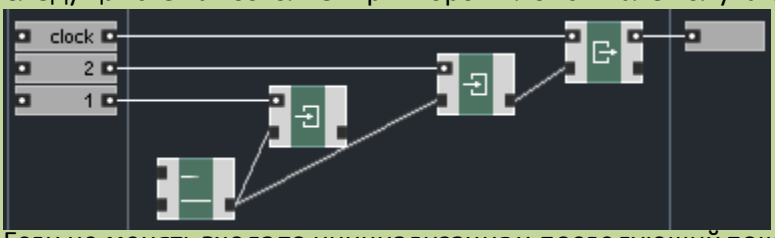
R/W Order, Array, Index

Создадим такую схему :

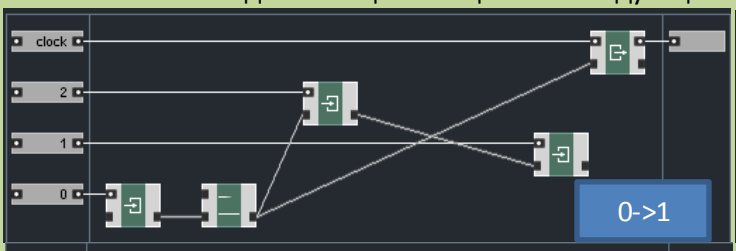


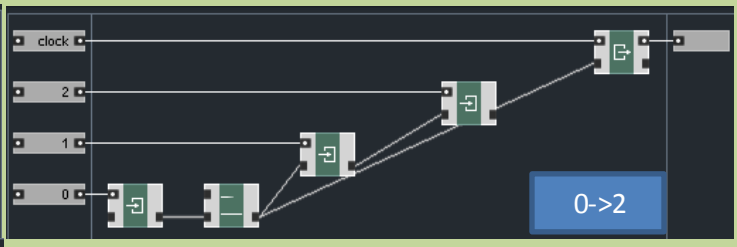
При старте выход покажет сигнал со второго входа, если не менять значения на 1 и 2 входах, то нажамая клок выход будет идти со второго входа.

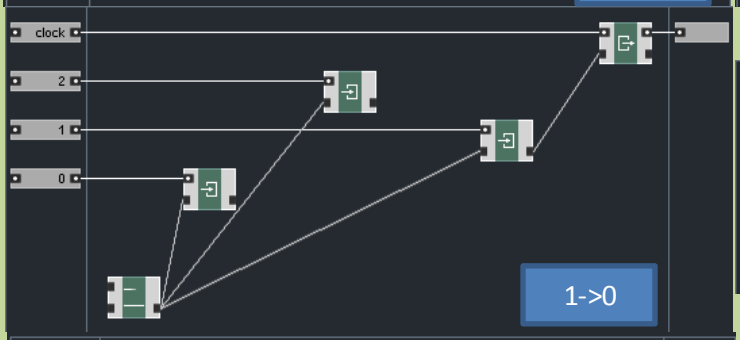
Следущая схема позволяет при втором клок сигнале получать выход с первого входа

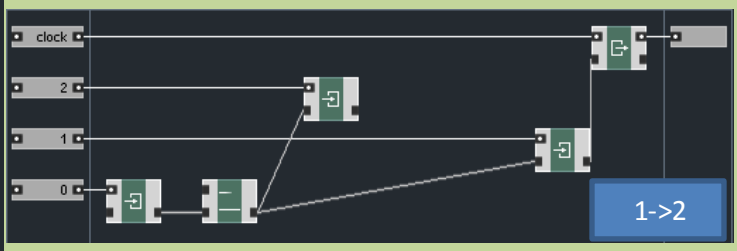


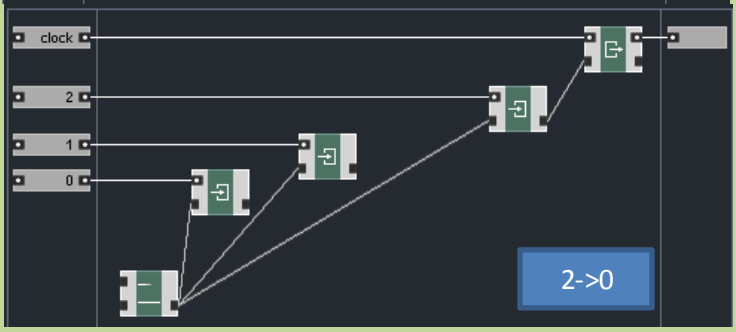
Если не менять входа то инициализация и последующий пошаговый клокинг покажет :

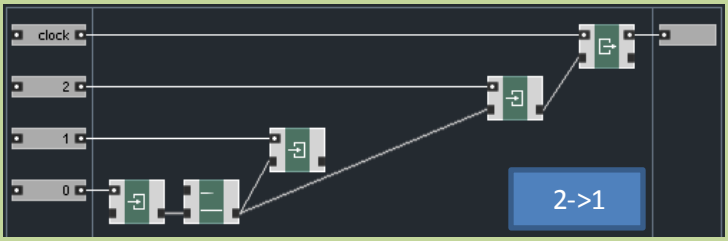






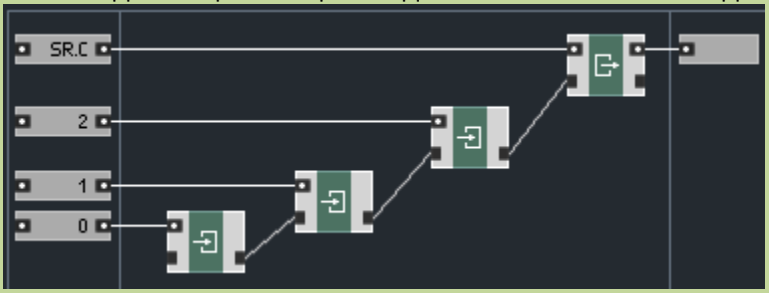






Если в качестве clock выбрать внешний CR.C то на дисплее при старте будут видны только данные с второго clock, так как CR.C это мгновеный clock, то изменения при старте вы не заметите.

Такие виды инициализаций входов можно использовать для старта модуля Merge :

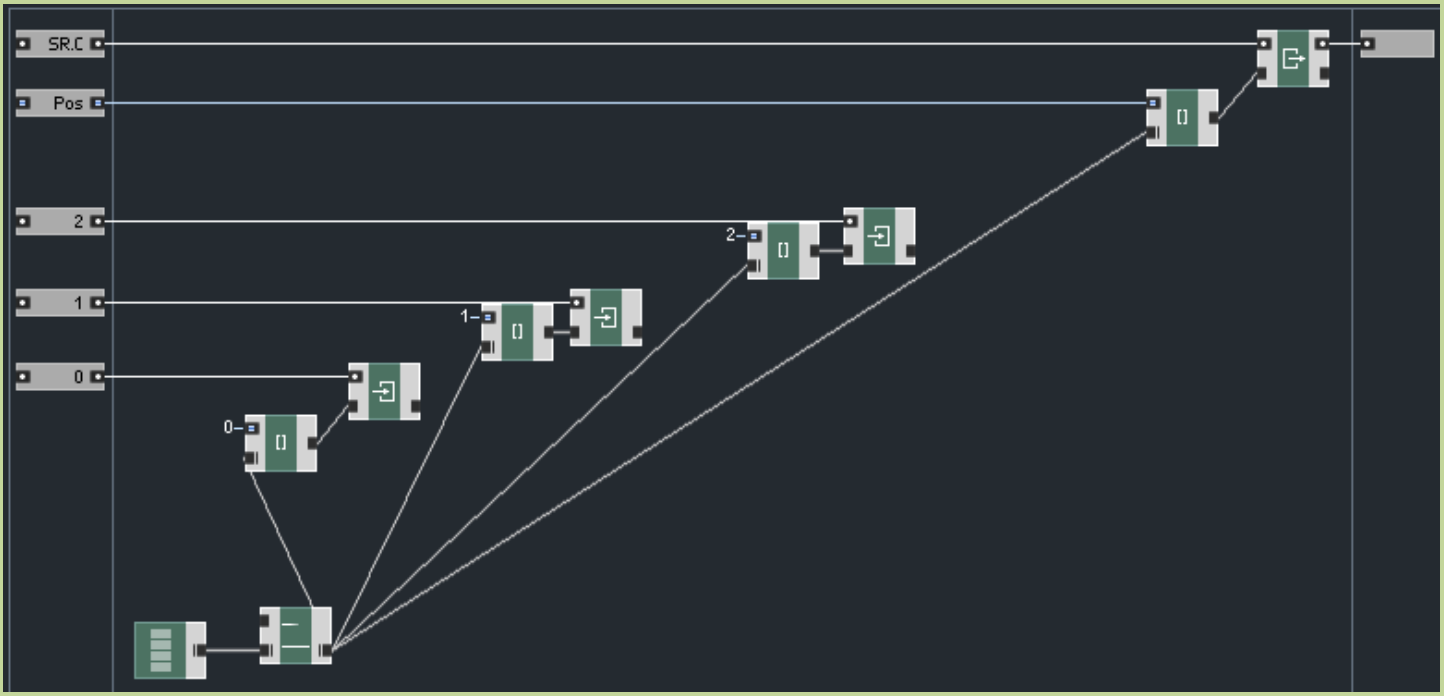


старт со вторым входом

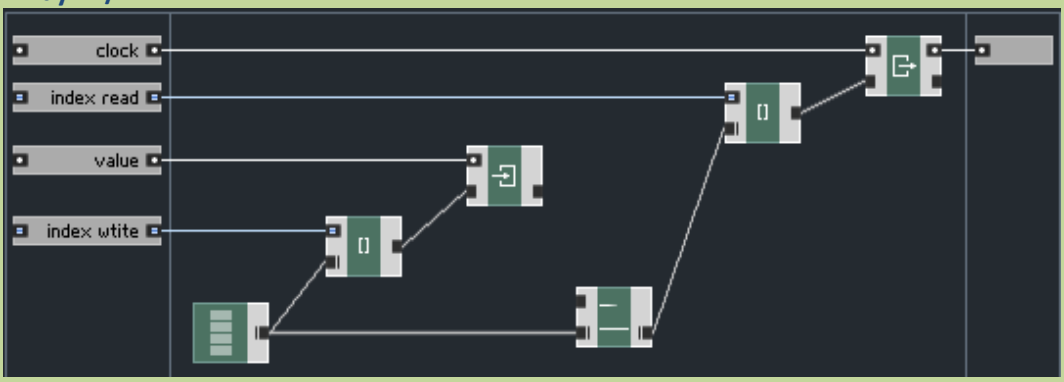
В отличии от первичного Merge где при старте (если входы не активны) выход всегда подаётся с самого нижнего входа, в core можно подать с любого входа при старте, с помощью Order W/R в том числе.

Array – это массив в котором хранятся значения. Массив можно расматривать как 2D структуру – XY, X это адрес который выберается с помощью индекс, Y – записываемое значение в этот адрес. Работа Core Array чем-то похожа на первичный модуль Event Table – в нём сначала выберается WX (адрес), а запись в нём это In. Индекс всегда должен иметь целочисленое (integer) значение. В отличии от первичных модулей кроме Event Table, в Core Array в качестве первого индекса можно использовать ноль.

Selector



Array W/R

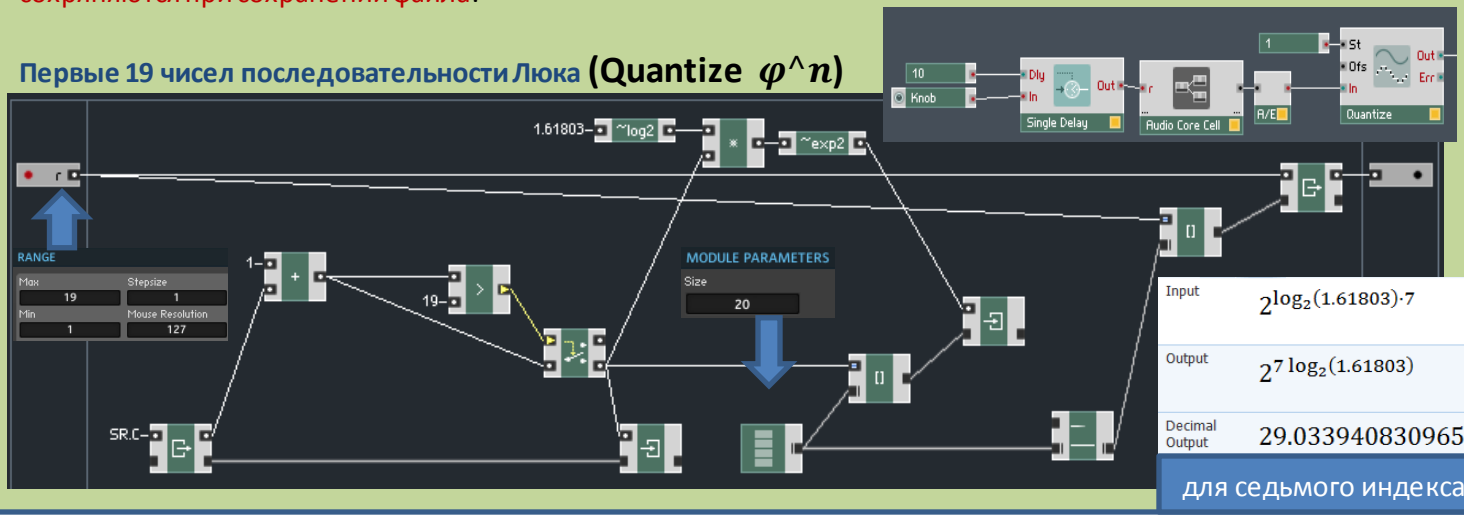


1. Выбераем индекс – крутим iw (вход index write)
2. Записываем значение в выбраную ячейку - крутим Knob (вход value)
3. Считываем значение с ячейки – крутим ir (вход index read)

Clock здесь мгновенный CR.C

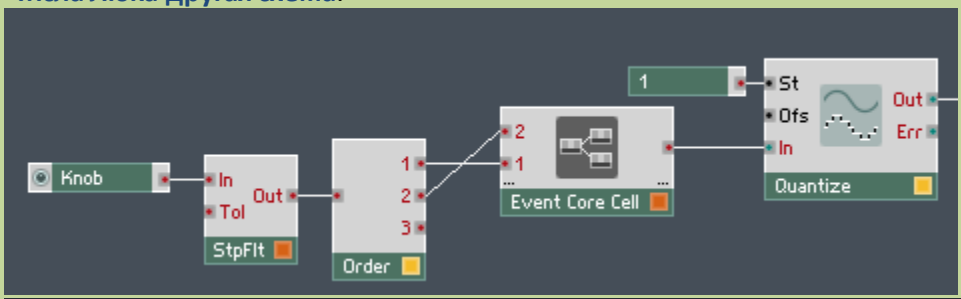
В отличии от Event Table – в Core Array значения пропадают после нового открытия файла, значения в Array не сохраняются при сохранении файла.

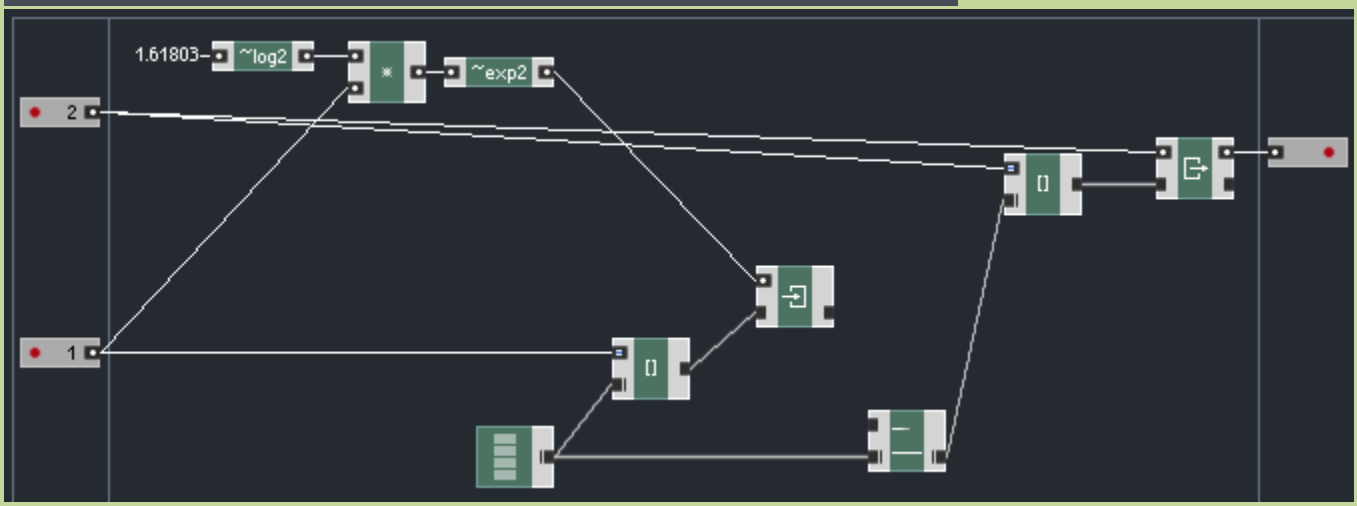
Первые 19 чисел последовательности Люка (Quantize φ^n)



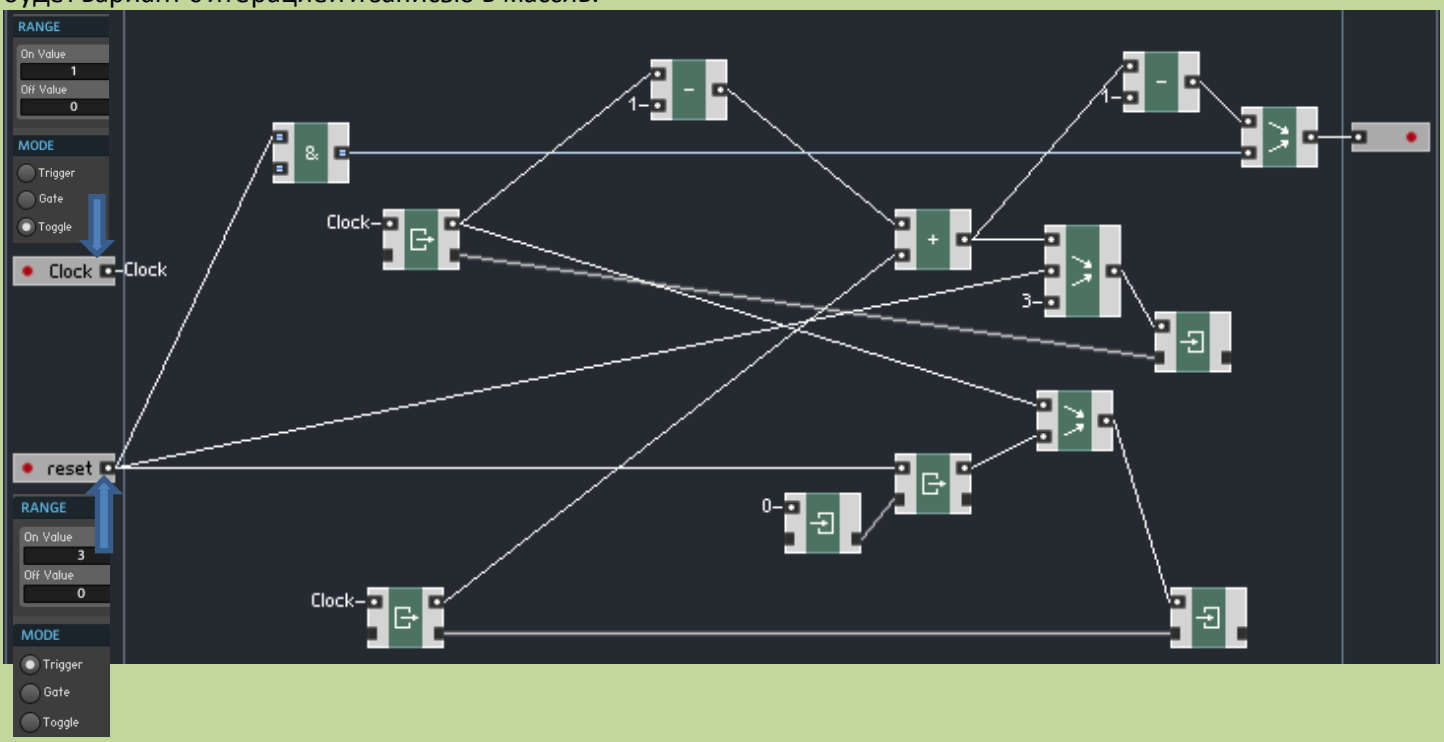
Числа Люка это золотое сечение в какой-либо степени с округлением.
Начиная с 20-го члена начинается ошибка в округлении, так как константу 1.61803 можно записать только до стотысячных 0.00001.

Числа Люка другая схема.

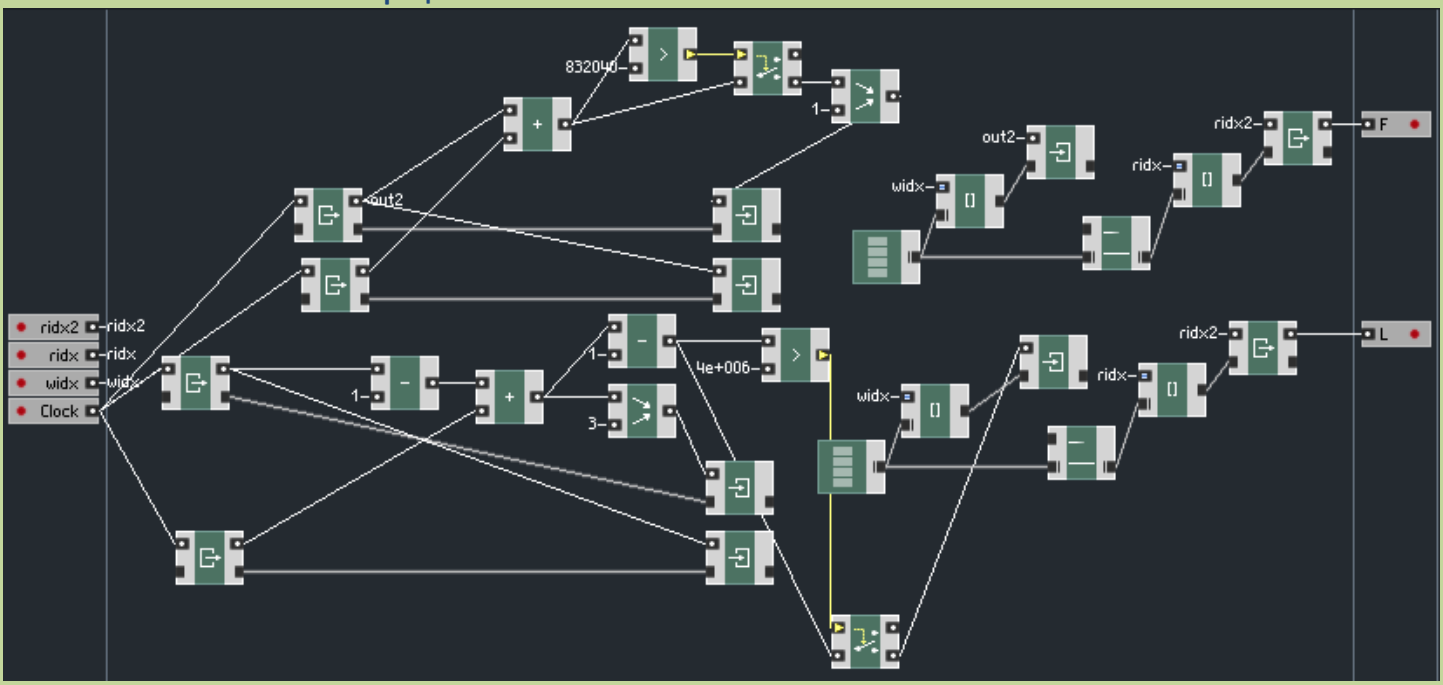


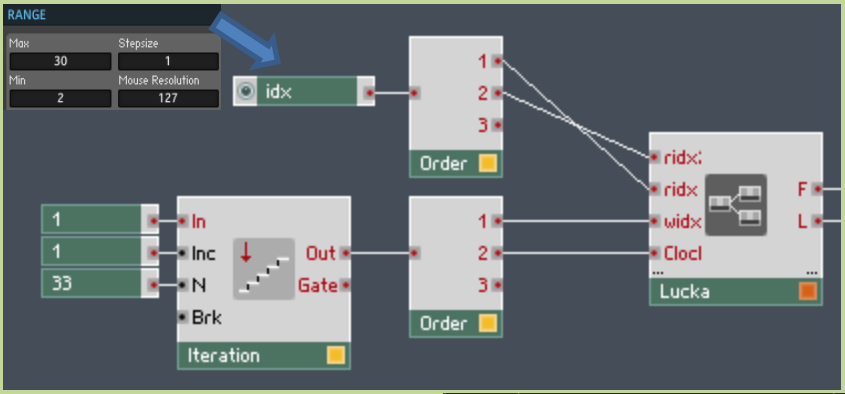


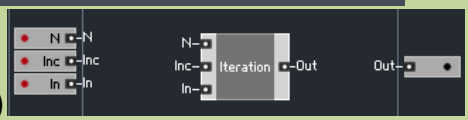
Числа Люка с обратной связью, так как здесь идёт суммирование то выход может идти до бесконечности без ошибок. Но чтобы добраться до n-члена последовательности нужно долго нажимать клок, в последующей схеме будет вариант с итерацией и записью в массив.



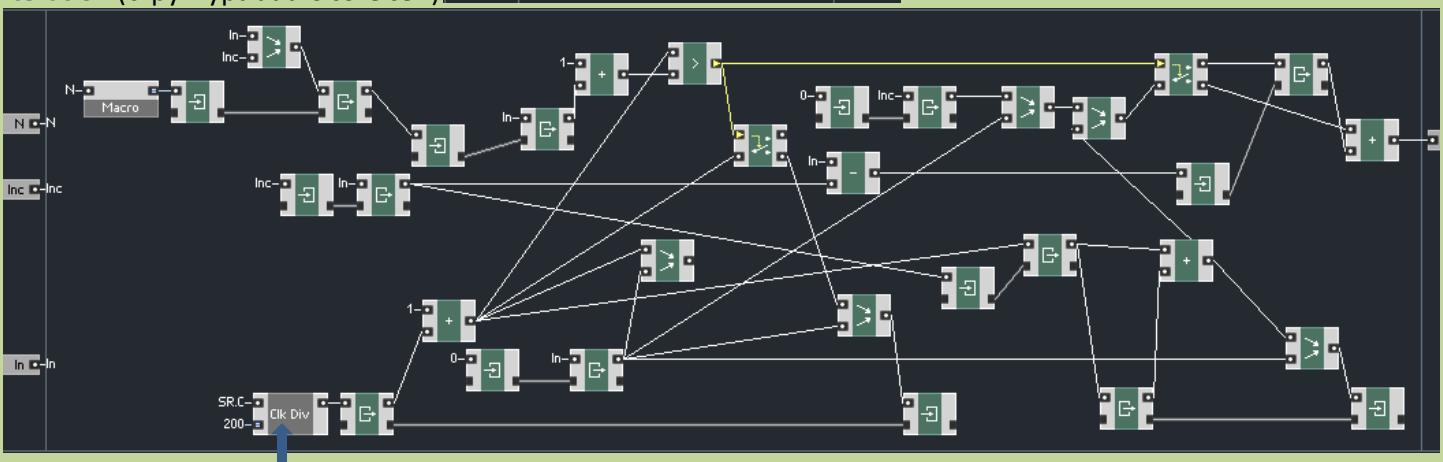
Числа Люка и Фибоначи с итерацией и записью в массив

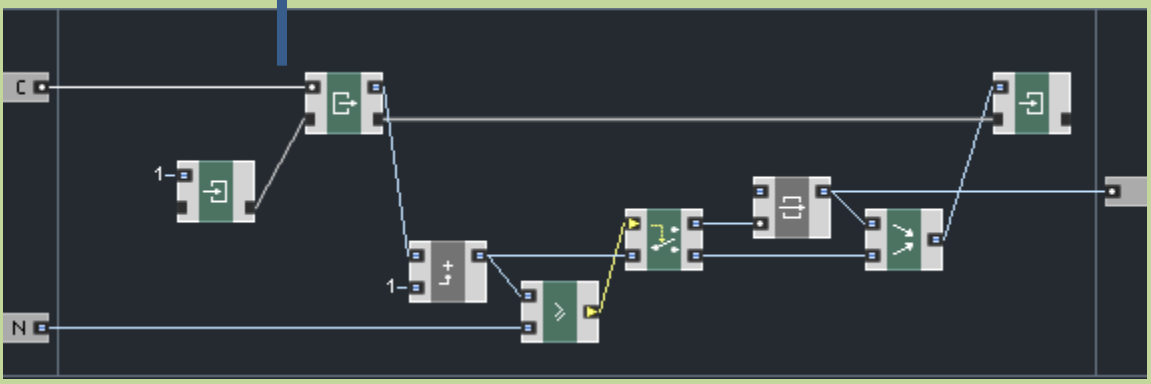




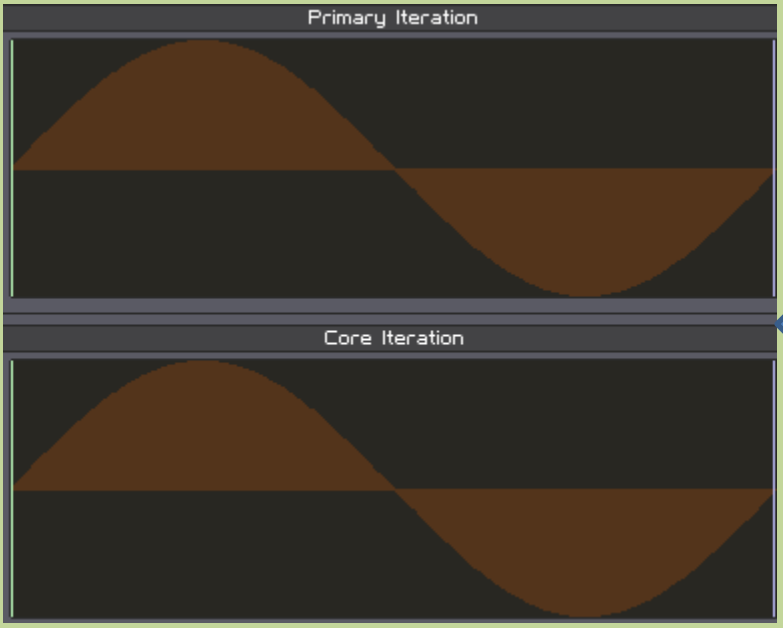


Iteration (структура audio core cell)



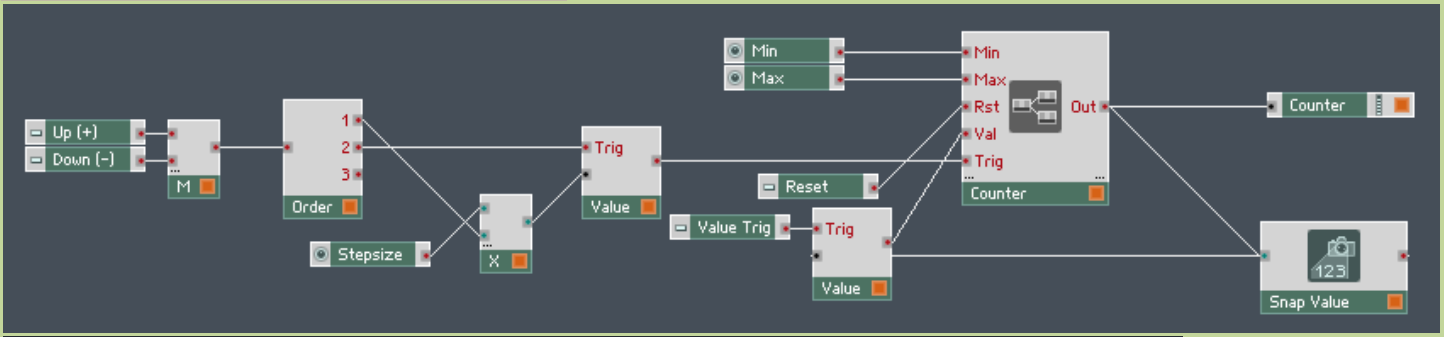


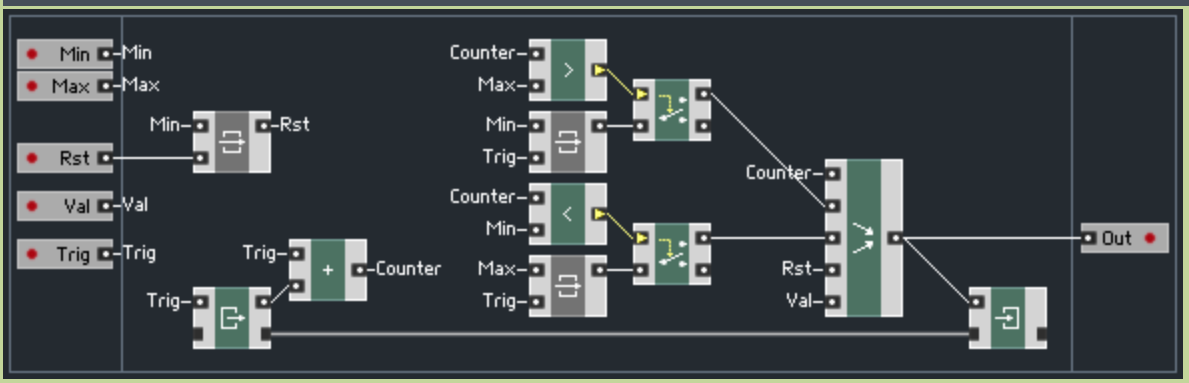
Этот итератор работает на частоте 200 Гц. Его работа такова – первое что он делает это создаёт индексы, индексы это сумма N+1. Допустим у нас N=4, значит количество индексов будет 5. Чтобы индексы создавались нужно получить цикл-сумматор с авто-остановкой. Второе что он делает это начинает суммировать предустановку In (первый индекс) с последующими N индексами где каждый индекс это + Inc.
Прорисовка полного периода синусойды коре итератором в евент тэйбл



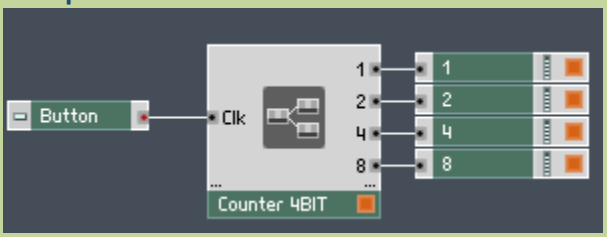
Здесь евент тэйбл содержит 360 WX индексов, так как коре итератор работает на частоте 200 Гц (200 индексов в секунду), то он будет немного запаздывать при прорисовке

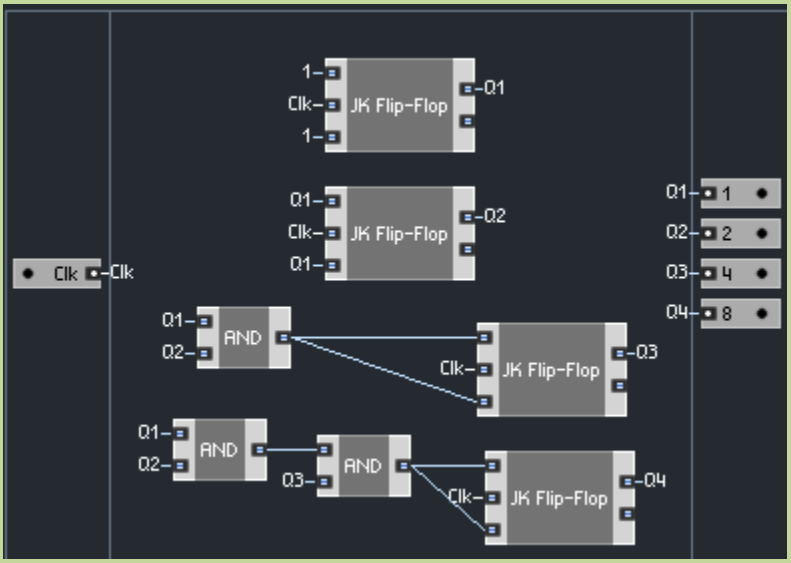
Счётчик

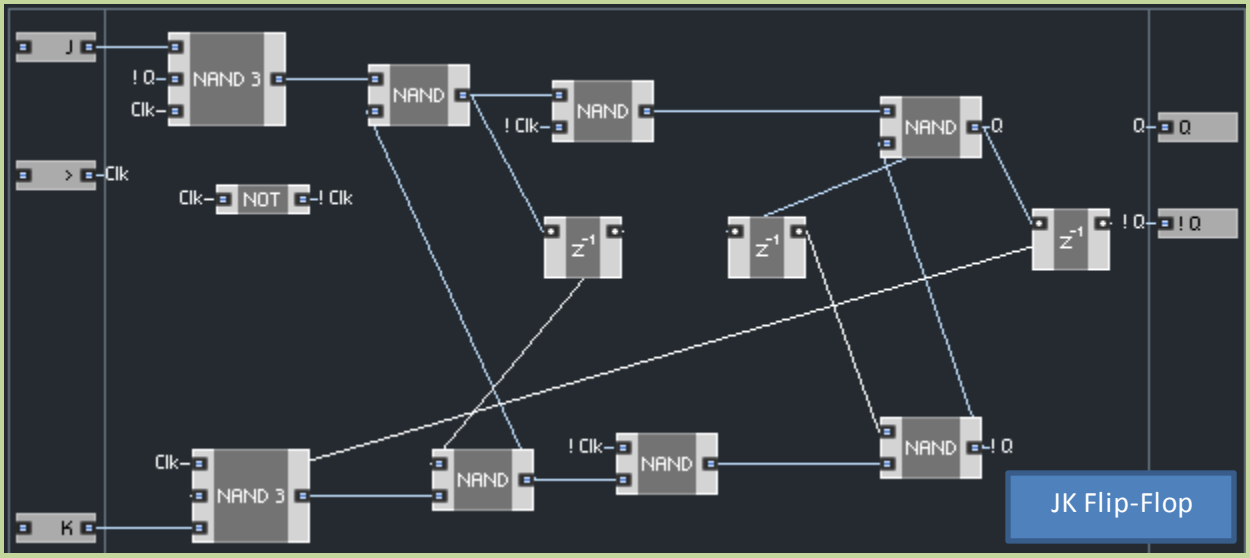




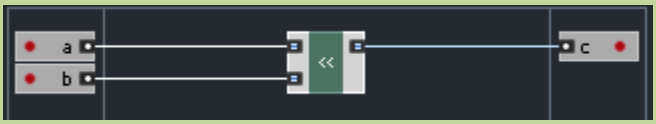
Бинарный счётчик



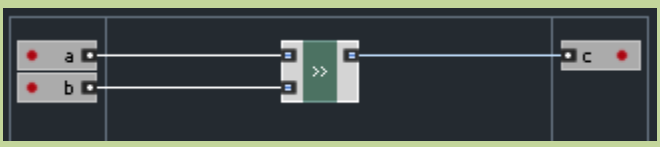




Битовый сдвиг

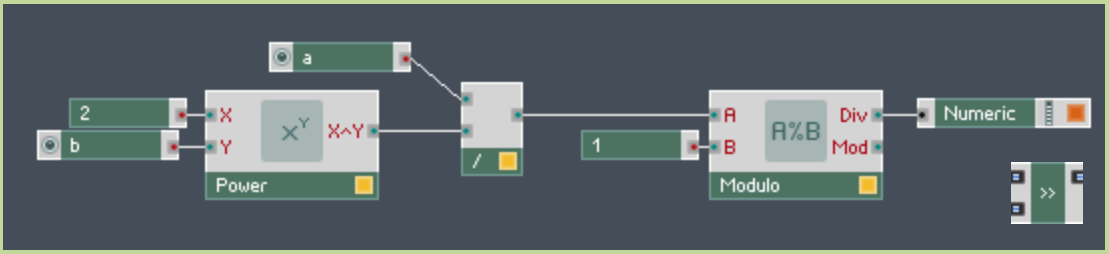


Пример 1 :
a=4, b=3
4 в двоичном виде 100
число 3 показывает сколько нулей надо ДОБАВИТЬ с правой стороны двоичного числа, поэтому
c=100000, что в десятичном виде = 32. Выход c=32
Пример 2 :
a=3, b=1
a=3 в двоичном виде 11
b=1 показывет сколько нулей надо добавить с правой стороны двоичного числа, поэтому
c=110, что в десятичном виде = 6. Выход c=6. (По сути, **c=a*2^b**)

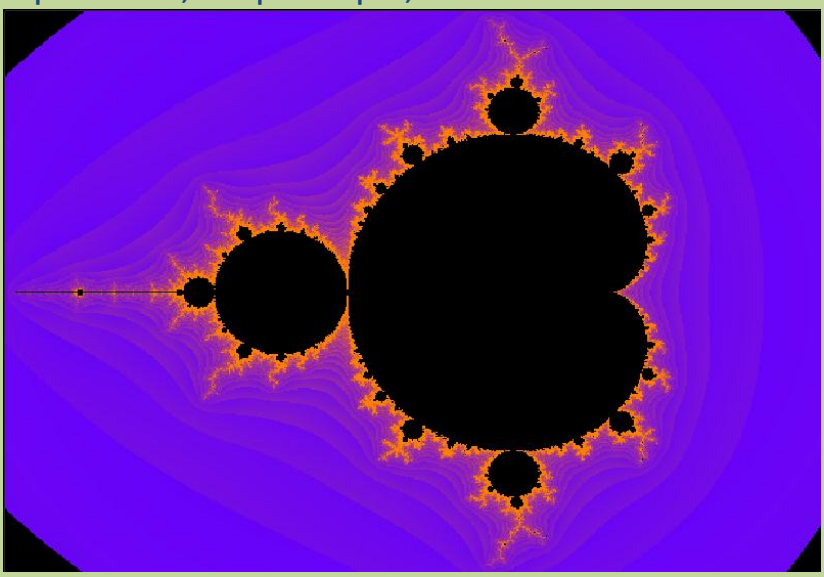


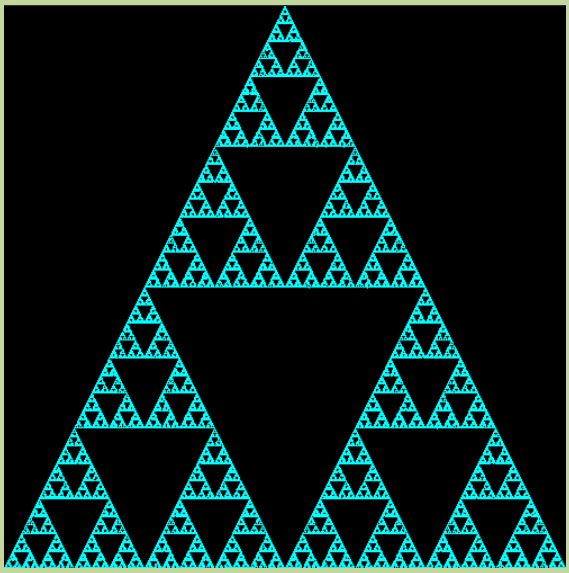
В таком сдвиге мы УБЕРАЕМ цифры с правой стороны, а не добавляем. Сколько цифр надо убрать, показывает число b.
Пример :
a=77, b=3 . a=1001101, убераем три числа справа, получаем 1001, что в десятичной системе = 9. Выход c=9

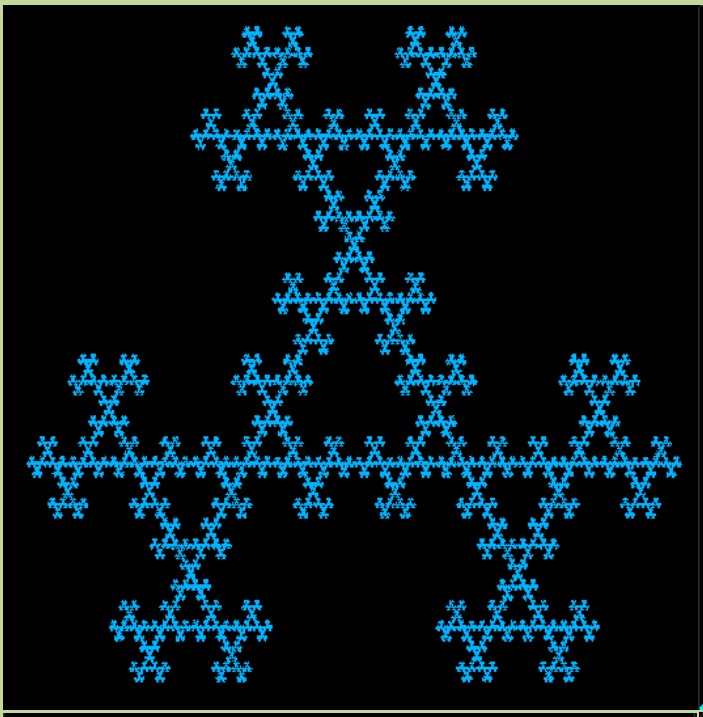
По сути, **c=a/2^b** с округлением в меньшую сторону

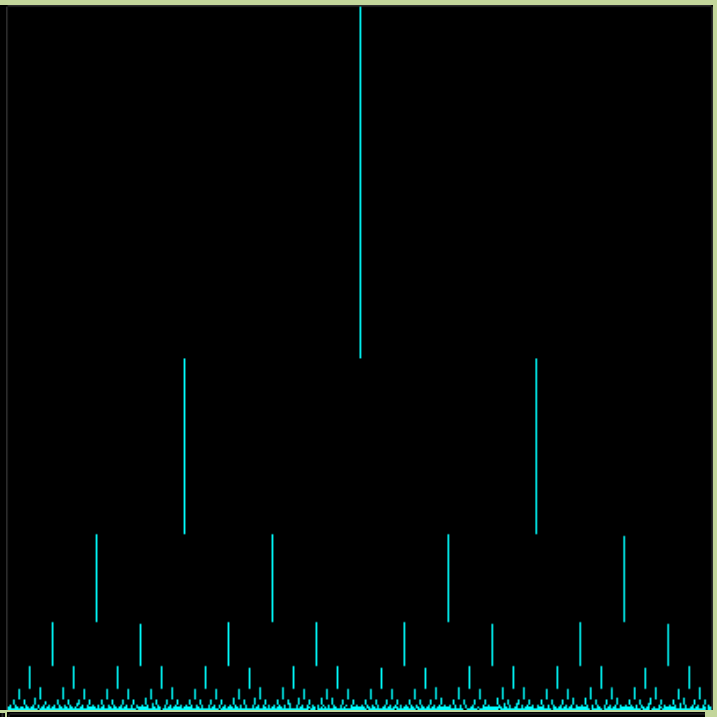


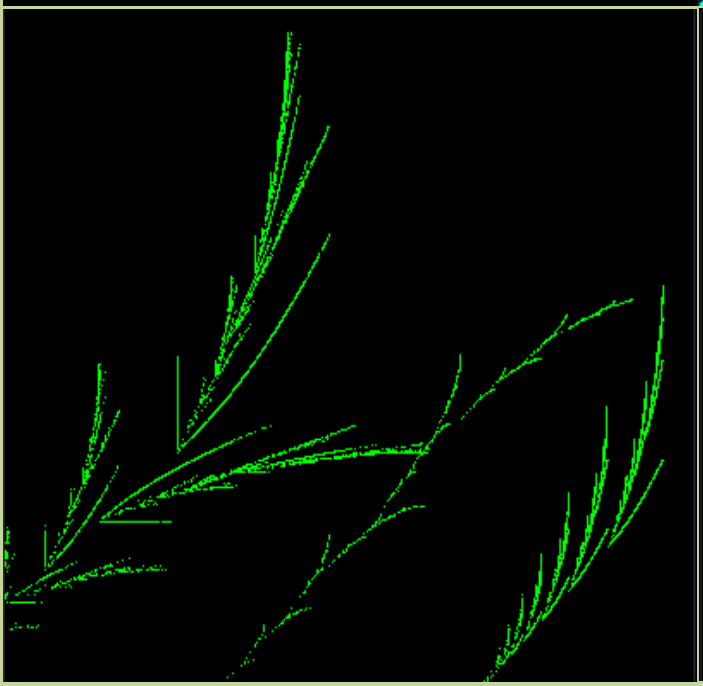
Фракталы, аттракторы, хаос.

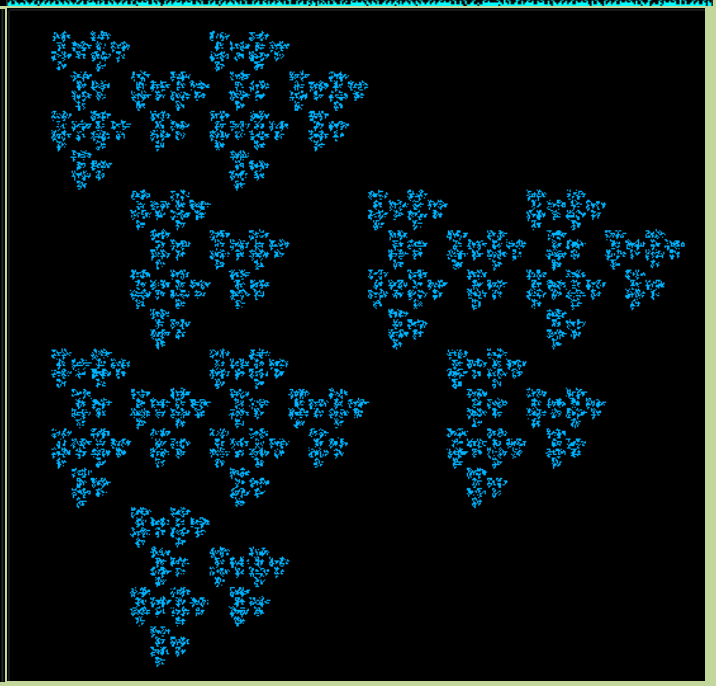




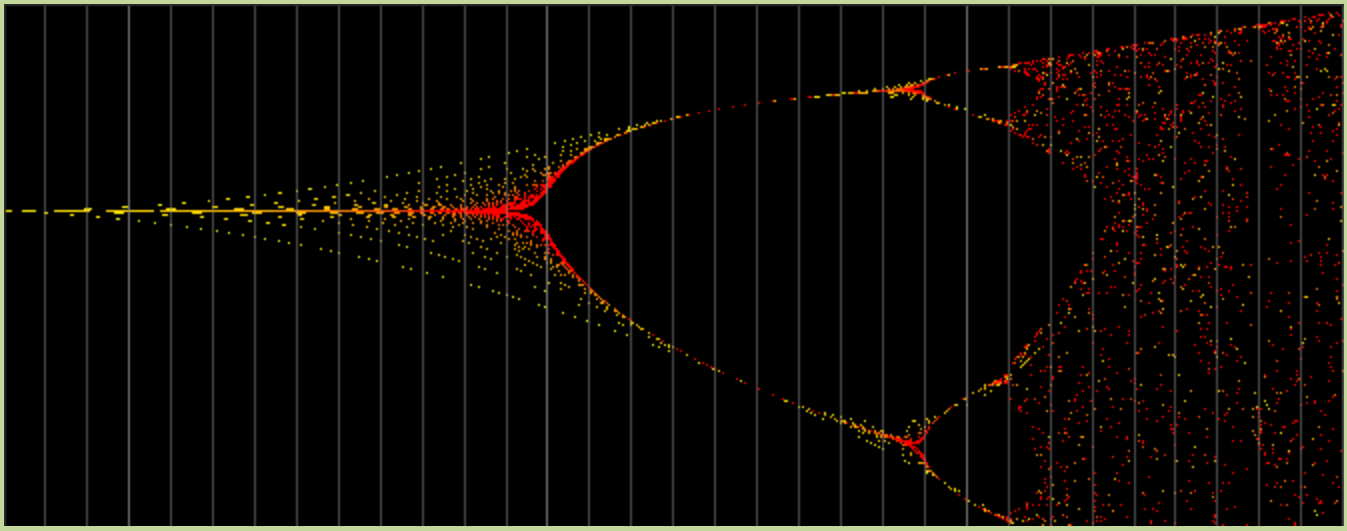




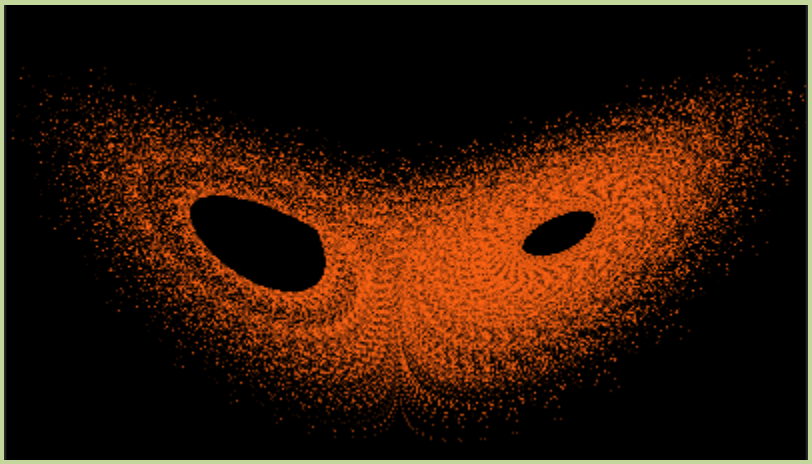


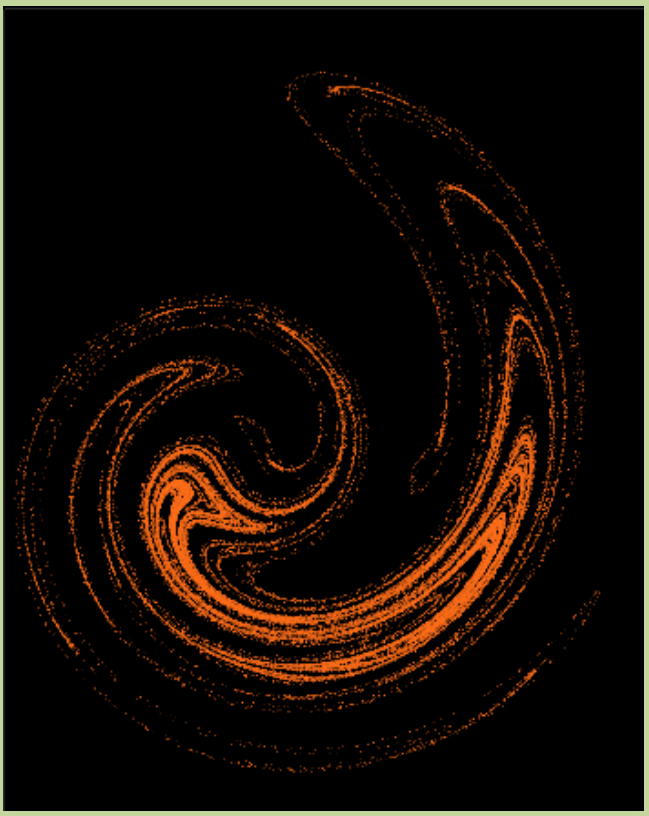


1.Мандельброта фрактал.2.Треугольник Паскаля - Серпинского.3.Снежинка.4.Пыль Кантора.
5.L-фракталы.6.Самоподобный фрактал.

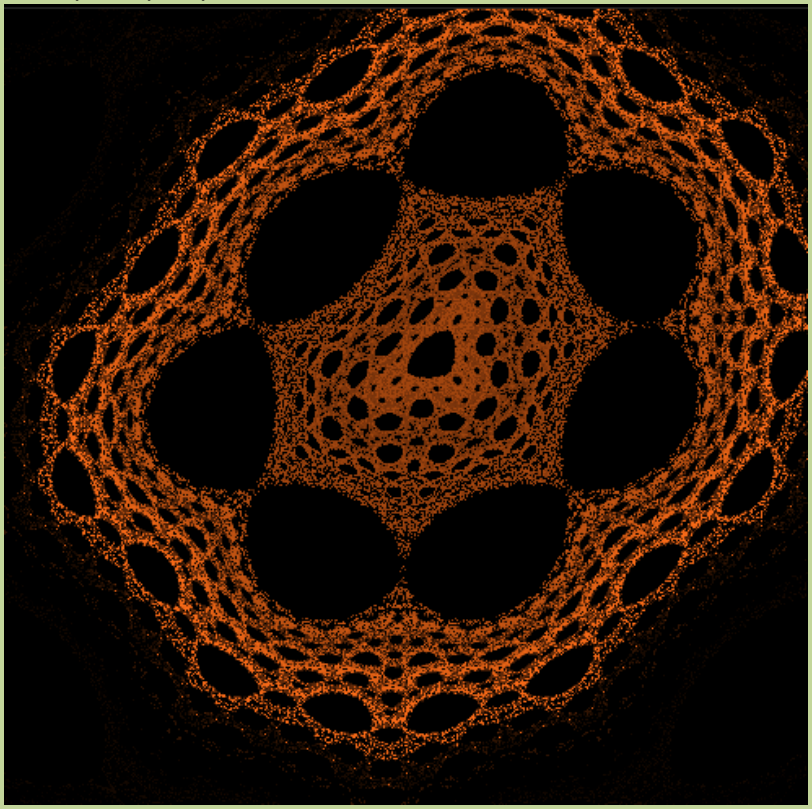


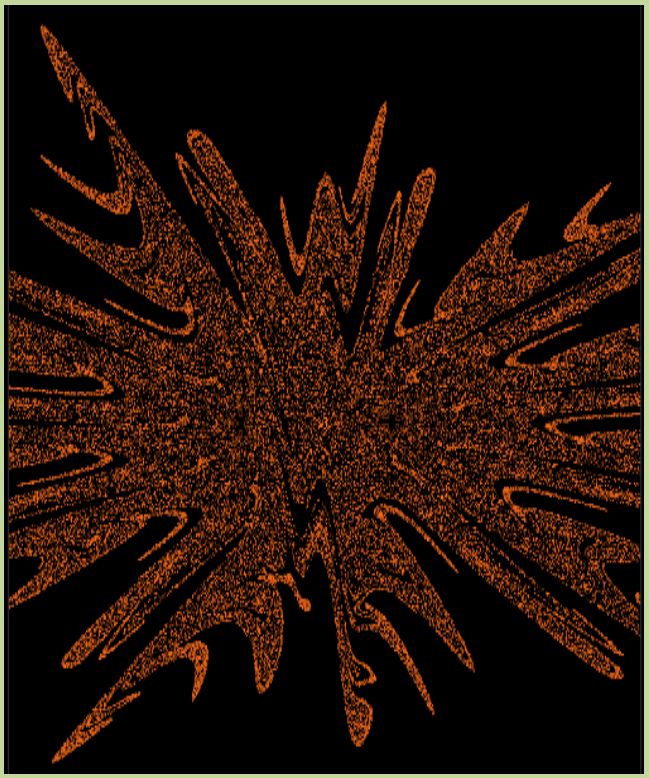
Бифуркация Фейгенбаума

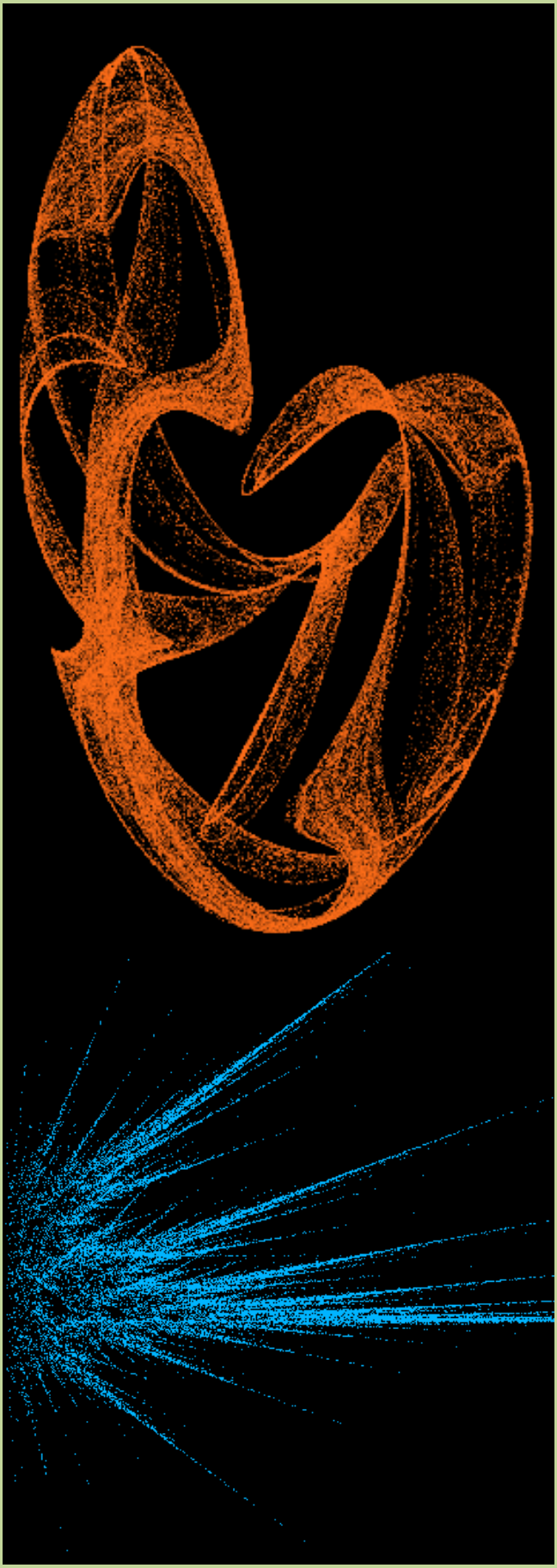


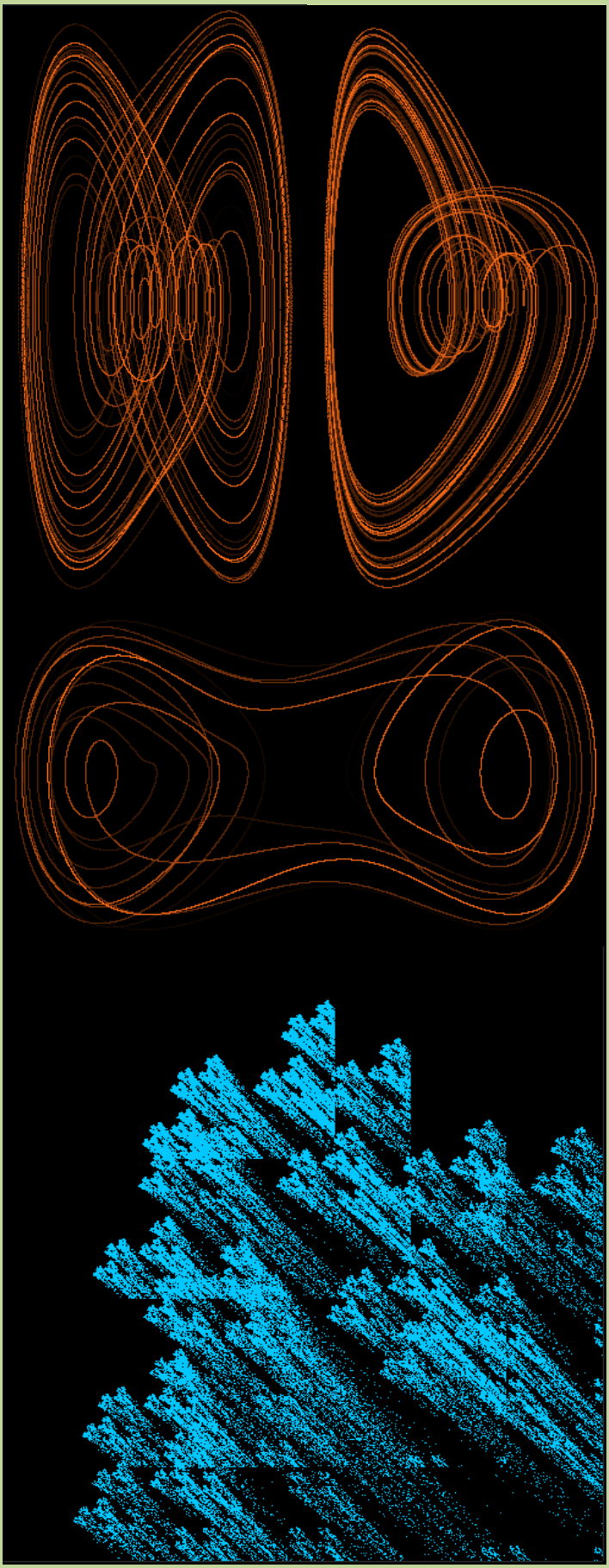


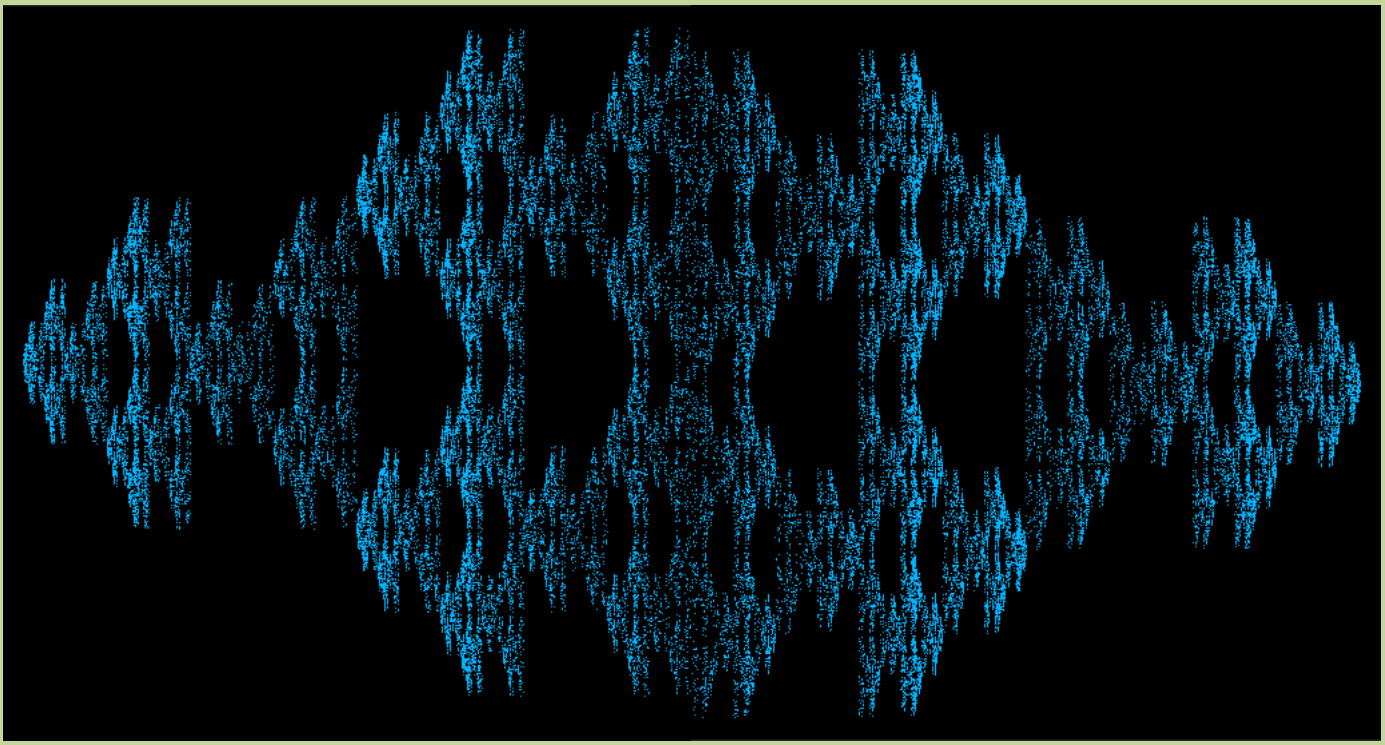
Аттрактор Лоренца

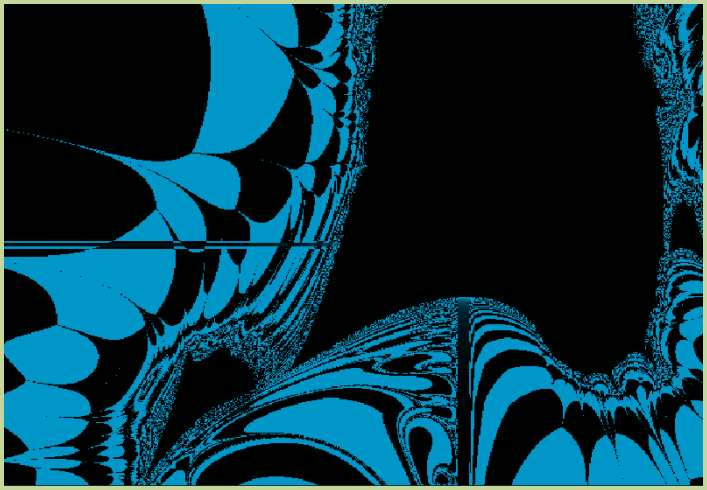


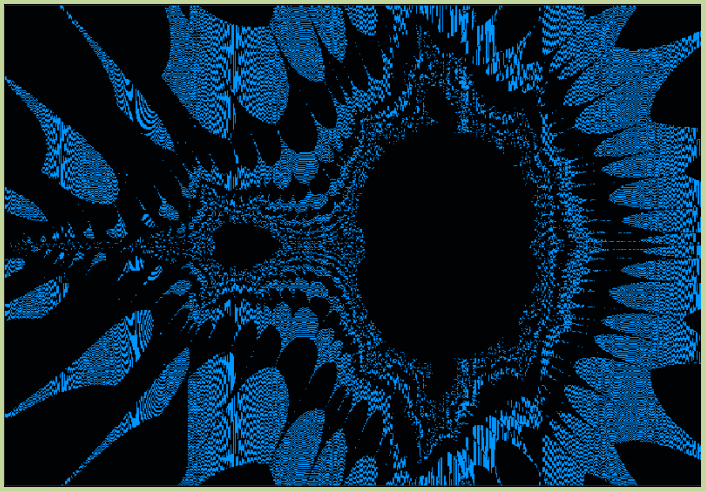


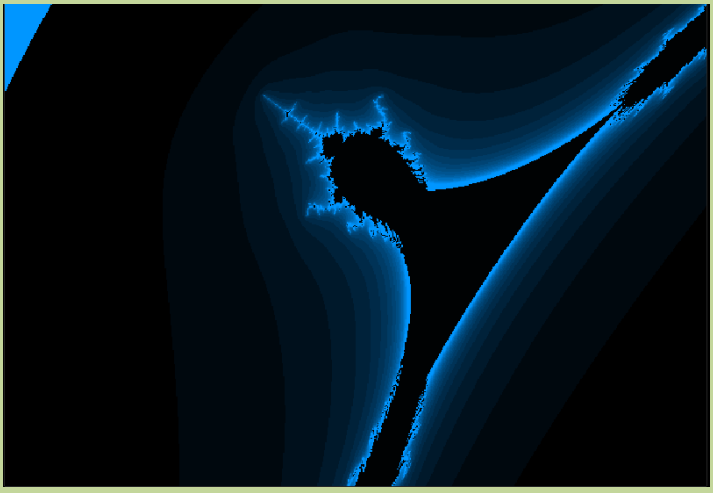


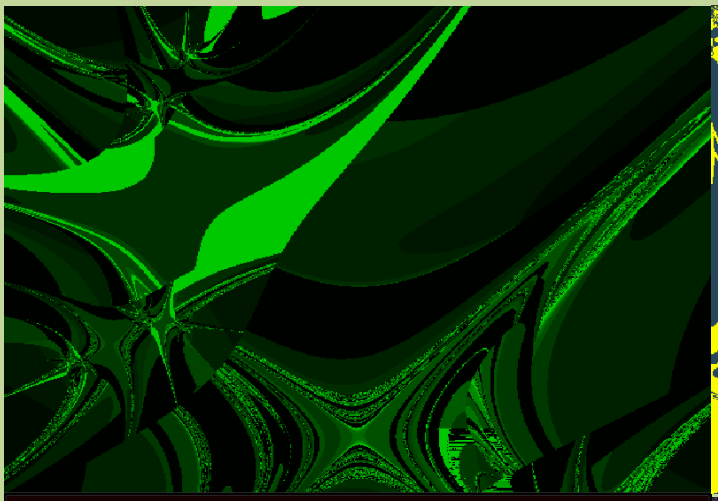
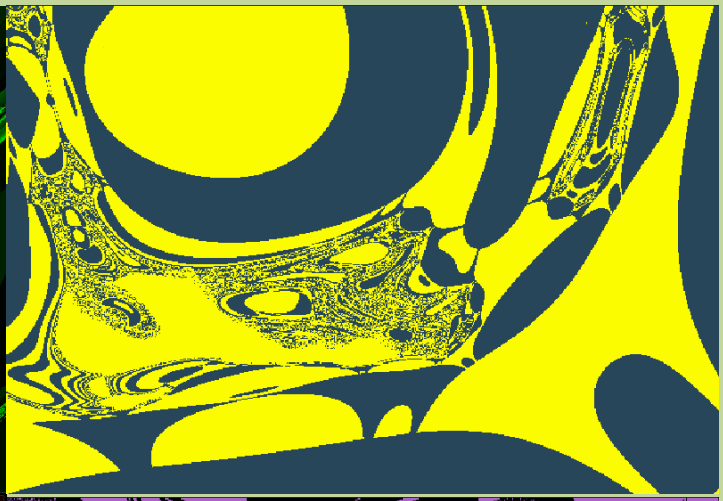
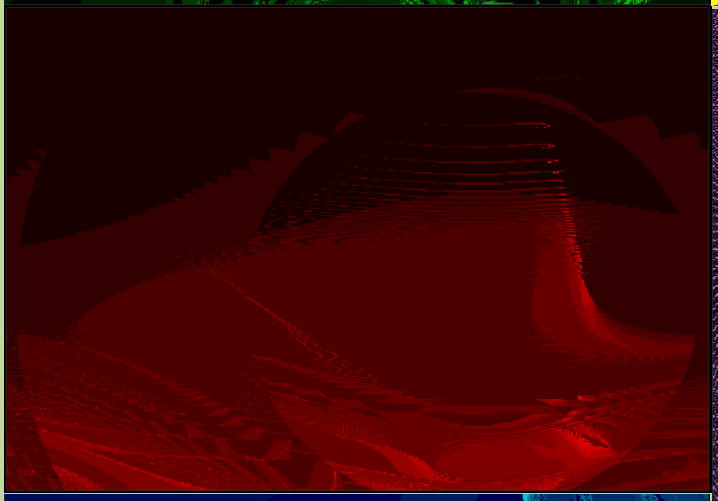


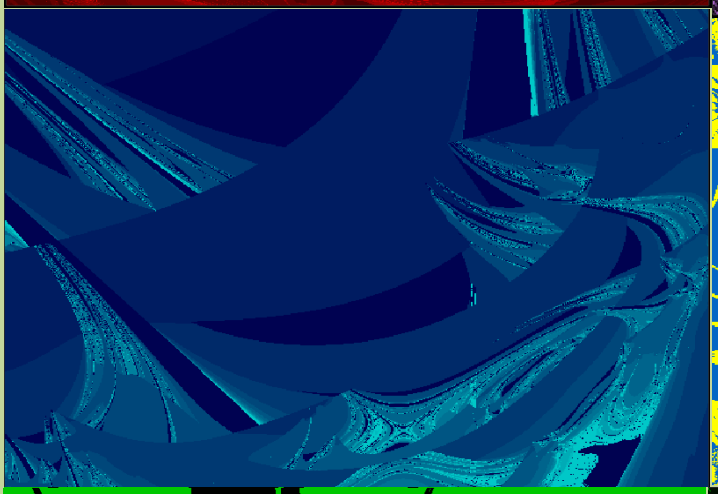
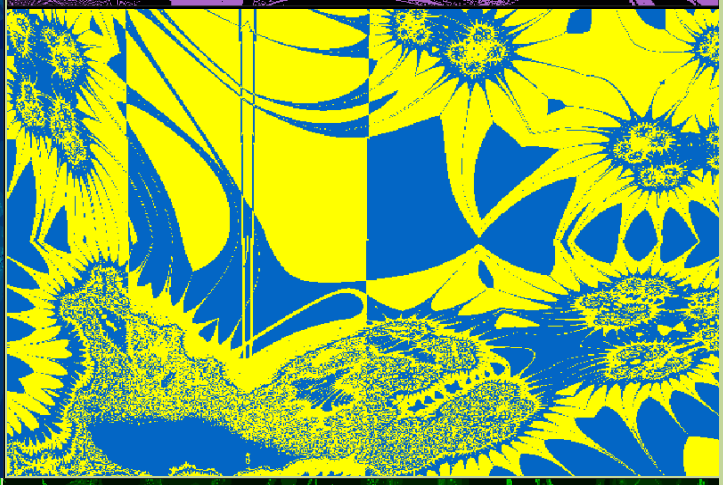
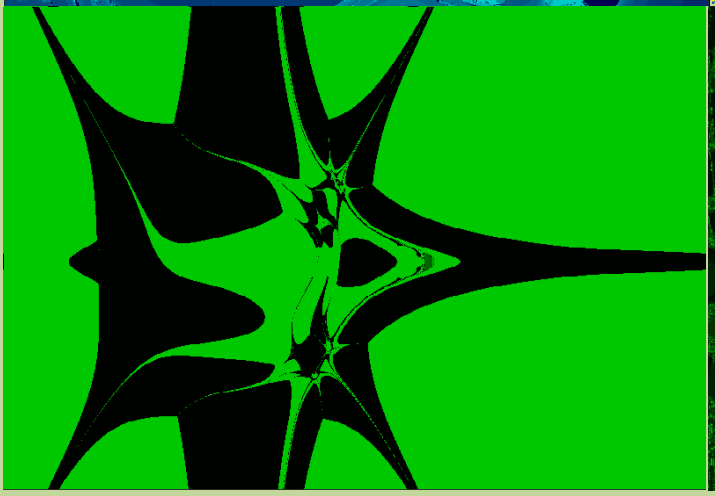
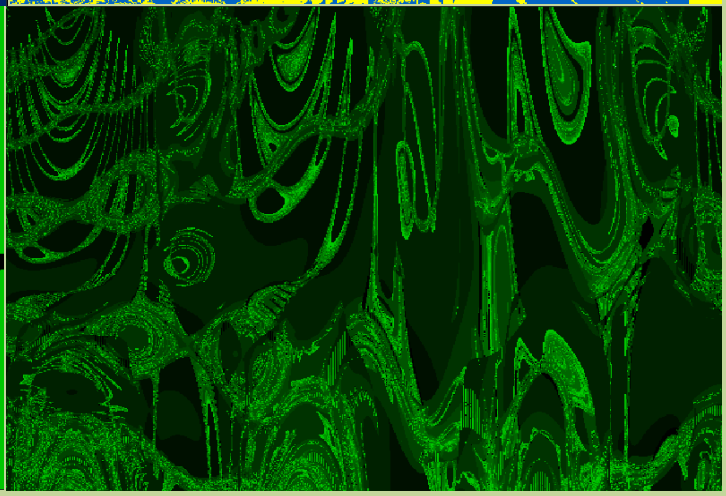


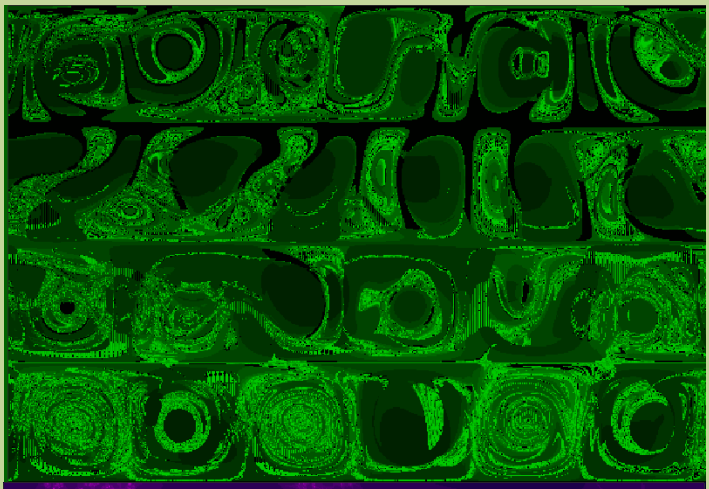


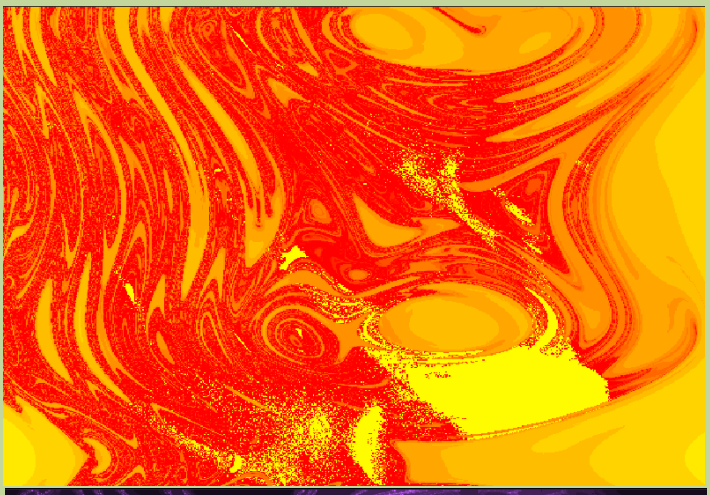


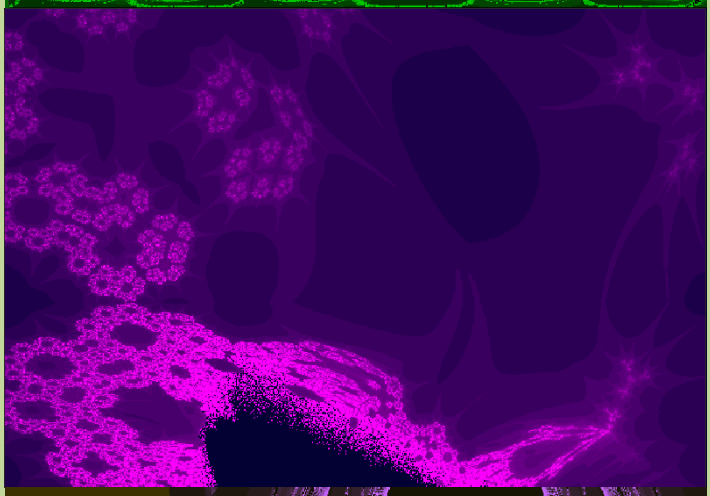


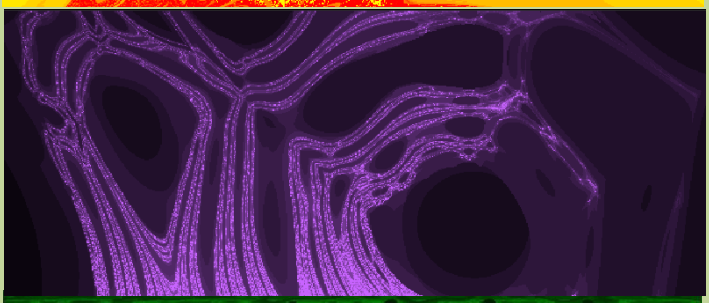




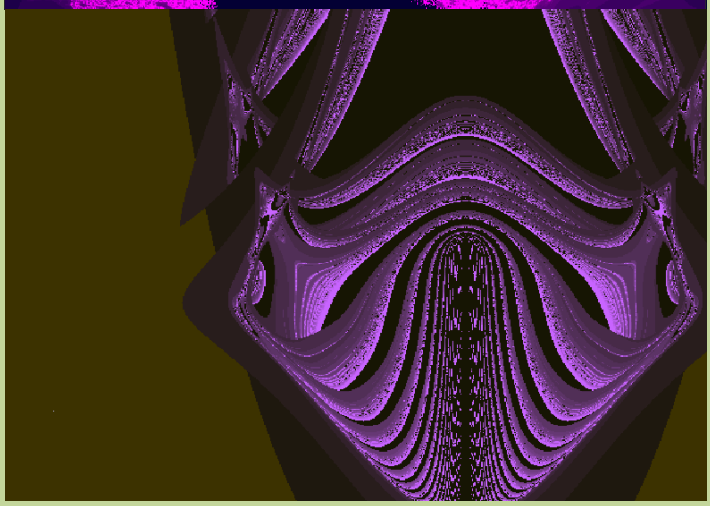





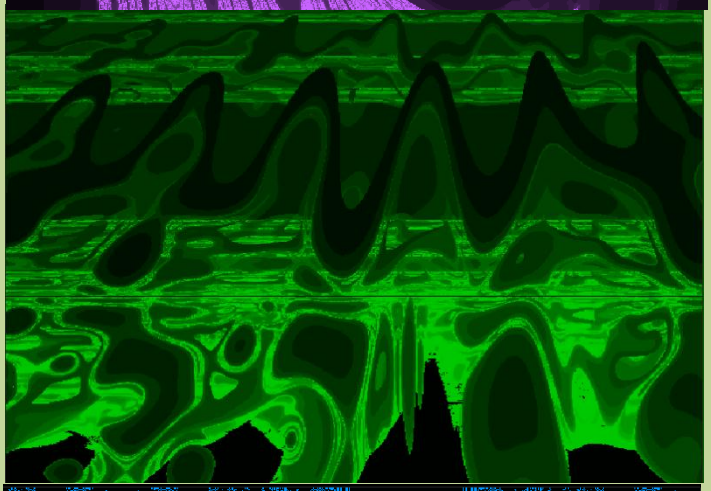


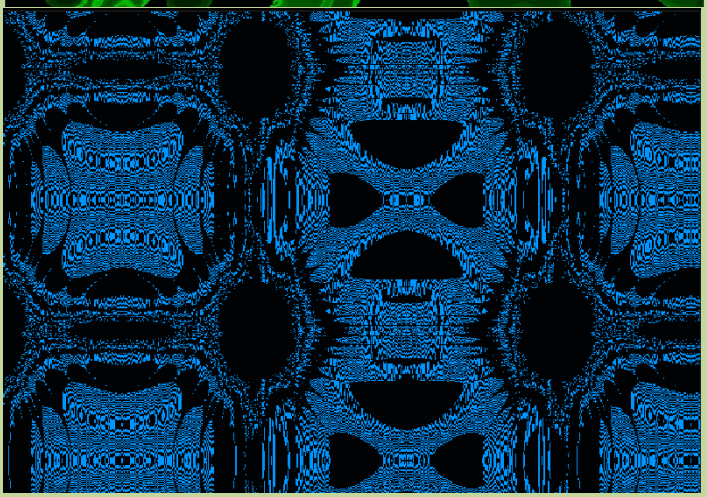


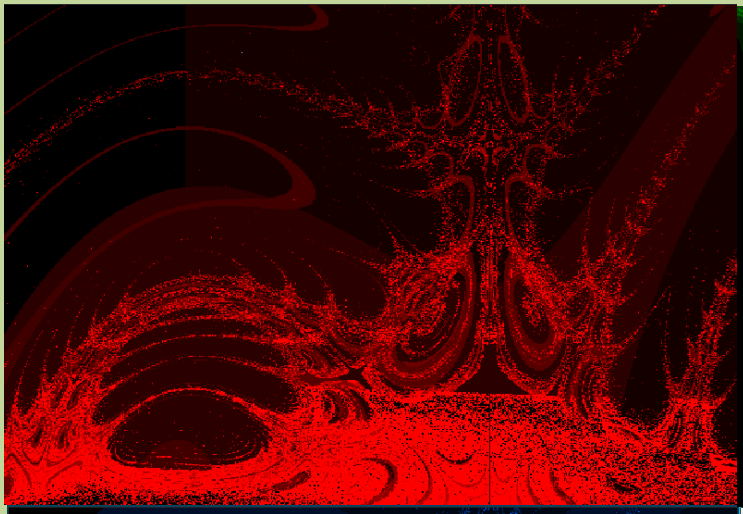


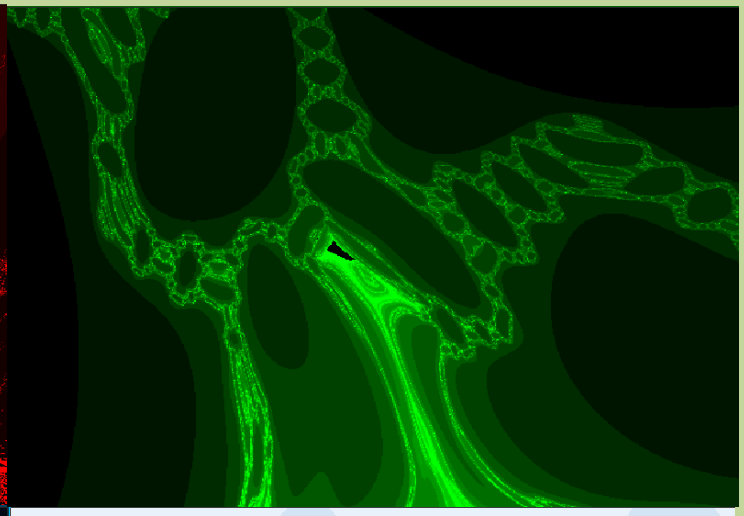


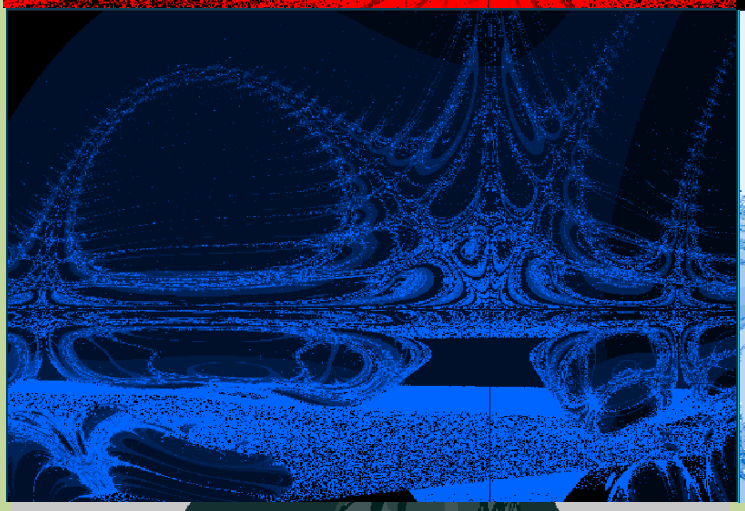


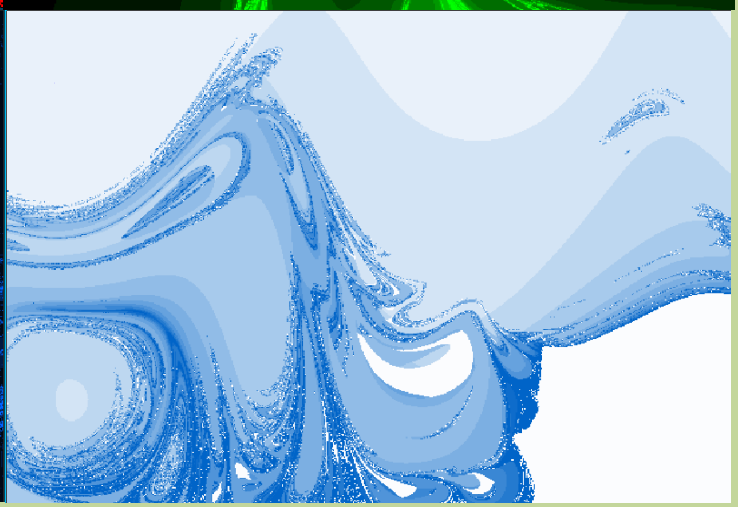


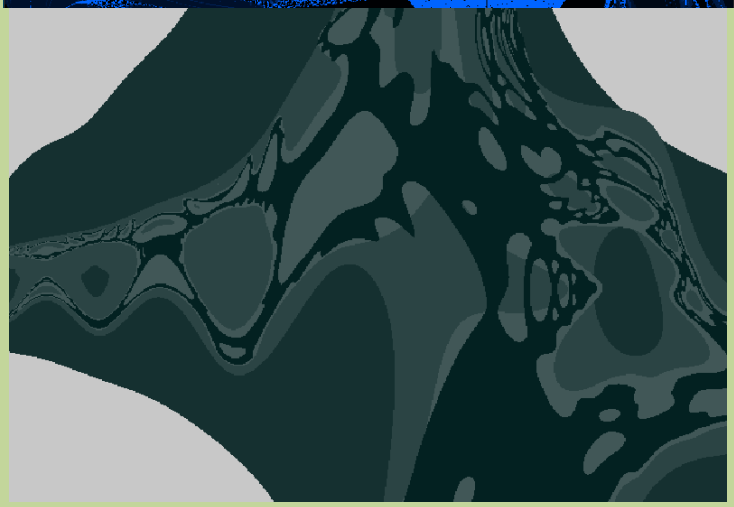












Эффекты делея и эхо

1. Простейшая задержка

Рис. 1.1. Вид панели управления устройства

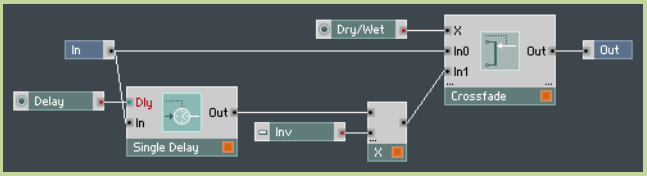


Рис. 1.2. Общая структура устройства

Самое простое, что может быть – это единичная задержка, которая микшируется с исходным сигналом. Устройство с панелью на рис. 1.1 и со схемой структуры 1.2 реализует это.

Модуль Single Delay и обеспечивает эту задержку. Регулятор Delay определяет время задержки, а кнопка Inv – инвертируем сигнал при надобности (далее….). Исходный сигнал и задержанный микшируются модулем Crossfade.

2. Простейшая задержка с обратной связью

Рис. 2.1. Вид панели управления устройства

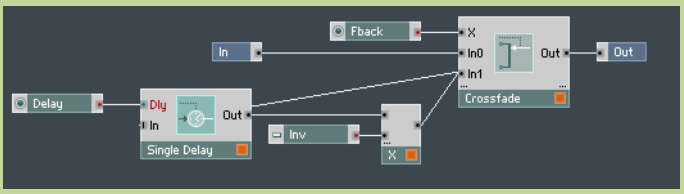


Рис. 2.2. Общая структура устройства

Рассмотрим устройство, реализующее задержку с обратной связью (см. рис. 2.1 и 2.2). Непосредственно исходный сигнал не идет на линию задержки. На модуль идет сигнал, взятый выходного порта модуля Crossfade. При таком построении возникает эффект как бы многократной задержки: сначала чистый сигнал поступает на модуль задержки, затем сигнал, содержащий исходный и задержанный – продолжает поступать на модуль задержки, формируя уже задержку этого нового сигнала, и так далее. Регулятором Fback можно регулировать уровень «чистоты» сигнала, поступающего (в очередной раз) на задержку и в выходной порт устройства.

3. Простейшая задержка с обратной связью и фильтром

Рис. 3.1. Вид панели управления устройства

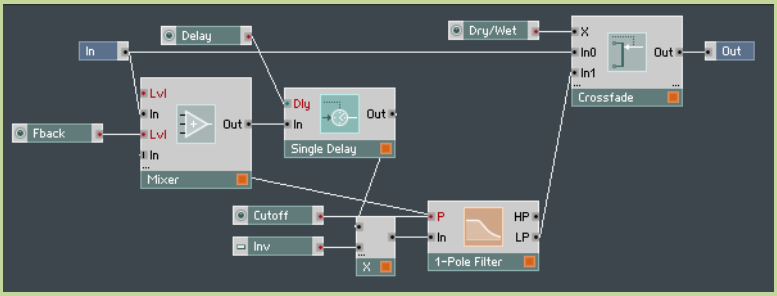


Рис. 3.2. Общая структура устройства

Рассмотрим версию задержки с фильтром. В цепь обратной связи будет входить непосредственно задержка и фильтр, а также отдельный микшер, осуществляющей контроллирование уровня этой обратной связи (рис. 3.1 и 3.2).

Итак, входной сигнал идет на выводящий смеситель Crossfade и на миксер цепи задержки цепи. Второй сигнал на Crossfade берется с фильтра. Тот же самый сигнал с фильтра берется в качестве обратной связи и поступает на микшер обратной связи. Его уровень регулируется Fback. Для фильтра задается частота среза регулятором Cutoff (логарифмический контроль). В остальном схема повторяет структуру схемы в первом примере.

4. Задержка со смещением тона

Рис. 4.1. Вид панели управления устройства

Рис. 4.2. Общая структура устройства

(разработка)

5. Грейн-делей с фильтром

Рис. 5.1. Вид панели управления устройства

Рис. 5.2. Общая структура устройства

(разработка)

6. Стереоделей с фильтром и регенерацией

Рис. 6.1. Вид панели управления устройства

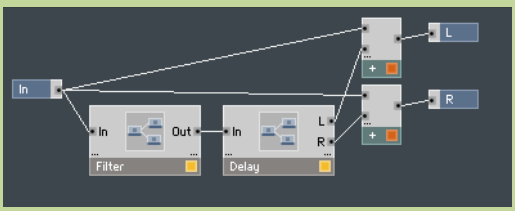


Рис. 6.2. Общая структура устройства

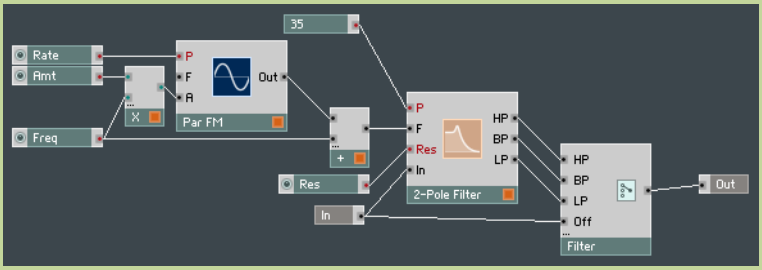


Рис. 6.3. Структура макроса Filter

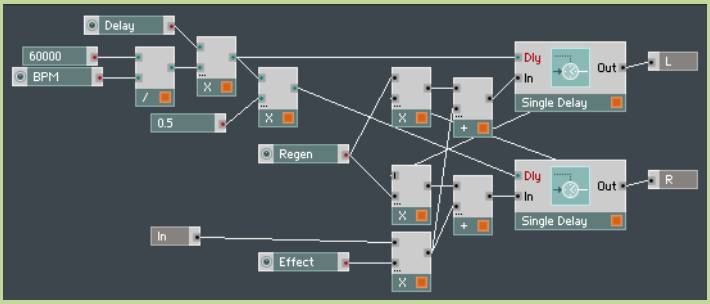


Рис. 6.4. Структура макроса Delay

Панель устройства (рис. 6.1) позволяет контролировать параметры фильтра, а также тип фильтрации. Общая структура состоит всего лишь из двух макросов и сумматоров обработанного сигнала с исходным (рис. 6.2).

Рассмотрим макрос фильтра (рис. 6.3). Здесь реализован фильтр второго порядка. Модуляция фильтра двойная: логарифмическая – задается константой 35 (это номер ноты), линейная более хитрым способом, с помощью параболического осциллятора и регулятора Freq. Высота тона осциллятора определяется Rate, а амплитуда зависит от регулятора Freq. Эта амплитуда регулируется Amt. (??)

Макрос Delay (рис. 6.4) состоит из двух параллельных линий простой задержки с использованием кросс-обратной связи (то есть сигнал с одной задержки подается в качестве обратной связи на другую линию задержки). Уровень входящего сигнала регулируется Effect, после чего сигнал поступает на два сумматора. На каждый сумматор также поступает сигнал обратной связи от делея (содержащегося в противоположной линии задержки), предварительно отрегулированный Regen. Время задержки для этих модулей делеев устанавливается такой схемой: (60000/BPM)*Delay для одной линии, а для второй время задержки уменьшается в два раза (в формуле BPM и Delay – это значения соответствующих регуляторов).

С этих двух делеев снимается в общей сумме два сигнала, представляющие компоненты стереосигнала, которые устройством и выводятся.

7. Стереоделей с обратной связью и резонансом

Рис. 7.1. Вид панели управления устройства

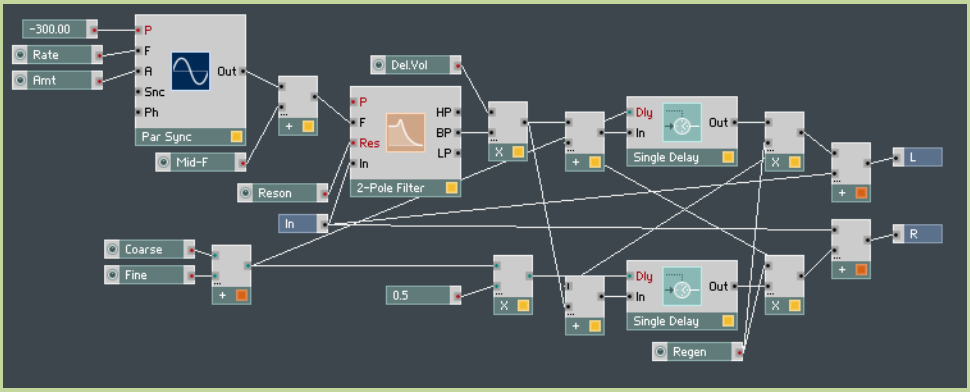


Рис. 7.2. Общая структура устройства

Рассмотрим еще один вариант стереоделея с применением фильтра (рис. 7.1 и 7.2). Идея этого устройства состоит в том, что здесь в отличие от предыдущего примера входной сигнал передается сразу на три порта входа: на фильтр, и на два сумматора в цепях задержки. Смешивание исходного сигнала и задержанного происходит в этих двух оконечных сумматорах.

В линию задержки включены: фильтр второго порядка с активным портом BP, два модуля задержки с кроссобратной связью. В цепи задержки регулятор Regen стоит после модулей задержки (а не перед ними, как в прошлом примере), и осуществляет регуляцию сигнала, идущего как на обратную связь, так и на оконечный сумматор. Также, в отличие от предыдущего примера, где уже к фильтрованному сигналу применялась задержка, в этой схеме реализуется другой принцип: в каждую обратную связь добавляется непосредственно компонента входного сигнала, прошедшего через фильтр. Это и есть главное различие приведенных двух схем. Количество этого дополнительного сигнала с фильтра регулируется Del.Vol. Частота среза фильтра (хотя здесь правильнее говорить о центральной частоте) управляется линейно посредством параболического осциллятора (плюс смещение регулятора Mid-F). В свою очередь, этот осциллятор управляется двухкомпонентно: логарифмически (-300) и линейно модулируется регулятором Rate (вернее, его значением). Амплитуда осциллятора задается регулятором Amt.

Время задержки устанавливается двухкомпонентно (Coarse+Fine), причем, на второй модуль задержки поступает значение в два раза меньшее.

8. Стереоделей с кросс-обратным фильтром (время задержки контролируется секвенсором)

Рис. 8.1. Вид панели управления устройства

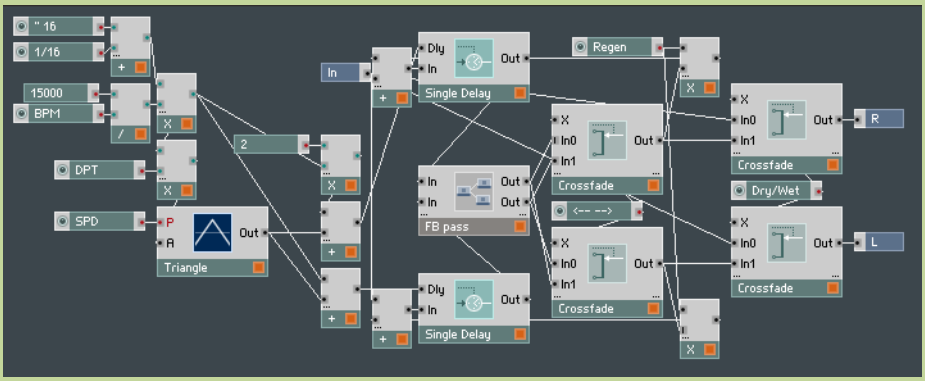


Рис. 8.2. Общая структура устройства

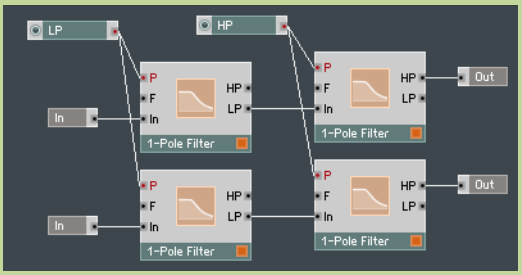


Рис. 8.3. Структура макроса FB pass

Рассмотрим стереоделей со смешенной (весовой) кроссобратной связью через блок фильтров. Панель и общая структура устройства изображена на рис. 8.1 и 8.2.

Левая часть схемы посвящена вычислению времени задержки делея. Оно складывается из двух компонент: осциллятора треугольной волны и выражения (`16+1/16)*(15000/BPM). Для одной из этих двух задержек (верхней) берется время удвоенного выражения. Это же выражение (правда, отрегулированное DPT, от сокращенного «глубина») посылается в качестве амплитуды осциллятора.

Цепь обратной связи состоит из двух делеев, макроса фильтров FB pass и двух Crossfade (последние два Crossfade служат для поканального смешивания исходного сигнала с задержанным). Исходный сигнал поступает на сумматор с сигналом обратной связи. После прохождения модуля задержки сигнал подается на макрос фильтров (рис. 8.3). Здесь сигнал поканально дважды фильтруется: сначала LP, потом HP (соответствующие регуляторы определяют частоту среза фильтров).

После фильтрации сигнал попадает на два Crossfade в двух анологичных копиях. На иллюстрации плохо видно как соединен макрос FB pass с модулями Crossfade. А он соединен симметрично: верхний out макроса соединен с In0 для верхнего модуля Crossfade, и с In1 для нижнего модуля, а нижний – наоборот. Регулятор ← → осуществляет вариацию смешивания. Если значение его равно 0, то верхний Crossfade пропускает сигнал только из In0, а нижний Crossfade (в силу симметричности) – только из In1. При значении регулятора 1 – все меняется с точностью до наоборот. И, наконец, если значение регулятора лежит внутри отрезка 0..1, то на выход подается соответственно смешанный сигнал (см. описание модуля Crossfade).

С этих двух Crossfade сигнал также снимается дважды: один идет в качестве сигнала обратной связи через регулятор Regen обратно на сумматор,

другой – уже на оконечные Crossfade, в которых он смешивается с исходным и подается на выходные порты устройства.

9. Статический стереоделей

Рис. 9.1. Вид панели управления устройства

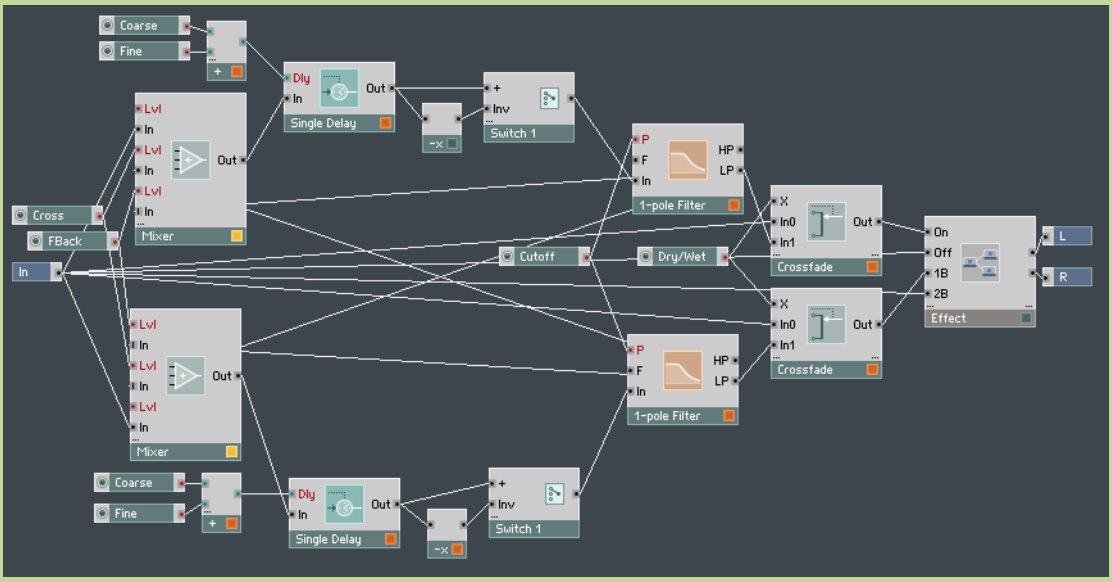


Рис. 9.2. Общая структура устройства

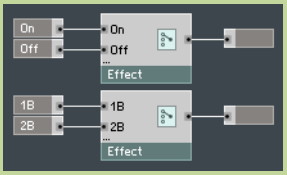


Рис. 9.3. Структура макроса Effect

Рассмотрим стереоделей с независимыми поканальными временами задержки (то есть время задержки устанавливается для каждого канала раздельно) и двойной обратной связью через фильтры (рис. 9.1 и 9.2).

На рис. 9.3 изображен макрос осуществляющий включение и выключение применения этого эффекта, расположенный перед выходными портами устройства.

Время задержки устанавливается независимо для каждого модуля задержки соответствующим двухкомпонентным выражением Coarse+Fine.

Каждый канал линии задержки имеет трехкомпонентный микшер, который обеспечивает смешение трех сигналов: исходного сигнала, сигнала обратной связи от фильтра (регулируется FBack) и кросс-сигнала обратной связи из другой линии задержки (регулируется Cross). Сигнал с микшера проходит через модуль задержки. После этого есть возможность его

инвертировать. Для этого создана простая конструкция из переключателя Switch 1 и модуля инвертирования сигнала. Далее сигнал попадает на фильтр (с общим для обоих фильтров логарифмическим регулятором Cutoff). С порта LP каждого фильтра сигнал идет в трех направлениях: в качестве обратной связи в микшер линии задержки, в качестве кросс-обратной связи в микшер противоположной линии задержки и в финальный Crossfade, где он смешивается с исходным сигналом и далее следует в соответствующий порт выхода устройства.

10. Простое эхо (из двух делеев, параллельное)

Рис. 10.1. Вид панели управления устройства

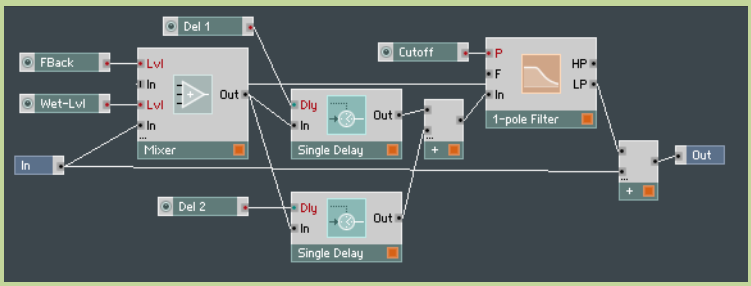


Рис. 10.2. Общая структура устройства

Эхо – это многократное повторение задержанного сигнала, следовательно без обратных связей эффект эха построить невозможно. Рассмотрим простой эффект эха (см. рис. 10.1 и 10.2). Устройство состоит из модулей двух простых задержек, включенных параллельно, фильтра и микшера.

Время задержки для каждого модуля Simple Delay определяется независимо регуляторами Del1 и Del2.

В микшер поступает два сигнала: исходный (регулируется Wet-lvl) и сигнал обратной связи (регулируется FBack). Сигналы двух задержек суммируются и проходят фильтр первого порядка с логарифмическим управлением частотой среза Cutoff. С фильтра сигнал поступает в качестве сигнала обратной связи на микшер, а также на оконечный сумматор, и, складываясь с исходным сигналом, выводится в выходной порт устройства.

11. Простое эхо (из двух делеев, последовательное двухступенчатое)

Рис. 11.1. Вид панели управления устройства

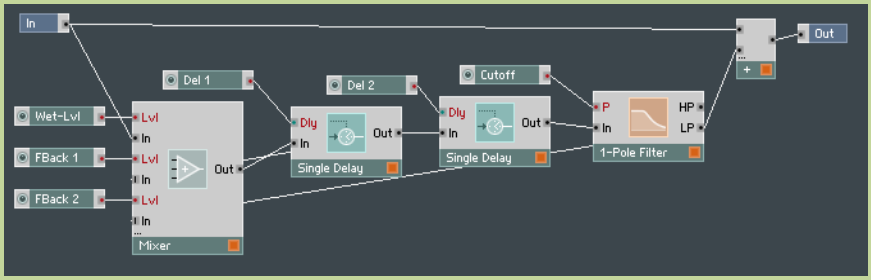


Рис. 11.2. Общая структура устройства

Модифицируем предыдущий пример, выстроив из модулей задержек и фильтра последовательную цепь (рис. 11.2). Панель устройства изображена на рис. 11.1.

Здесь микшер имеет уже три входных порта: порт исходного сигнала, порт обратной связи первого делея и порт сигнала обратной связи, идущего с фильтра. И регуляторов тоже три: Wet-lvl, FBack 1, Fback 2. Таким образом, в этом устройстве мы имеем две регулируемые цепи обратной связи.

12. Стереоэхо (с фильтром)

Рис. 12.1. Вид панели управления устройства

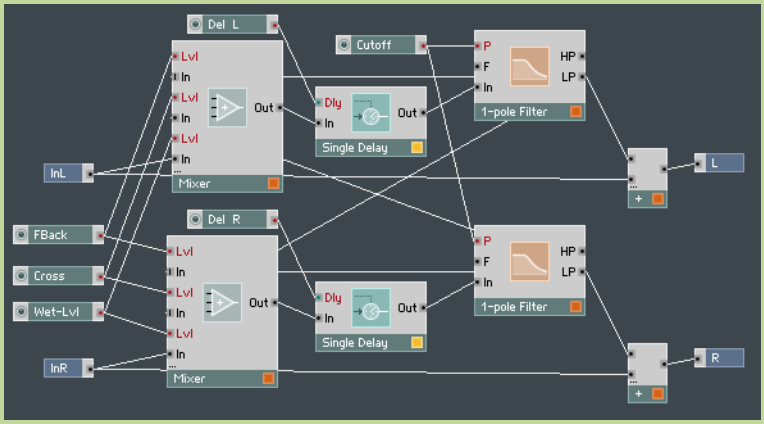


Рис. 12.2. Общая структура устройства

На рис. 12.2 приведена схема устройства стереоэха с обратной связью и поканальной фильтрацией. Его панель изображена на рис. 12.1.

Устройство очень напоминает пример 9: та же обратная связь через фильтр и обратная кросс-связь с фильтра противоположной сигнальной линии.

Отличие заключается в том, что изначально имеем двухканальный стереосигнал InL и InR. Каждый канал имеет независимую обратную связь, за исключением кросс-связи от противоположного сигнала, регулируемую в микшере регулятором Cross. (и Del L и Del R малы – для эха??? Проверить!). В остальном структура больше не требует пояснений.

13. Стереоэхо (4 делея, по 2 параллельных)

Рис. 13.1. Вид панели управления устройства

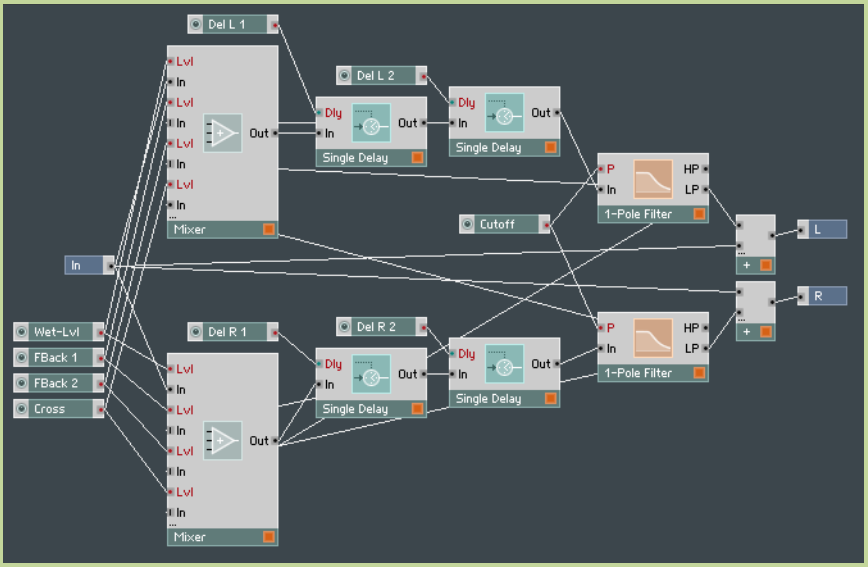


Рис. 13.2. Общая структура устройства

Данное устройство (см. рис. 13.1 и 13.2) является стереомодификацией устройства из примера 11, куда дополнительно добавлена еще и кросс-обратная связь между каналами. Итого, микшер каждой цепи обратной связи имеет уже по 4 входа (и, следовательно, по 4 регулятора; регуляторы, как видно, общие для обоих микшеров): порт сигнала обратной связи с первого делея, порт сигнала обратной связи с фильтра, порт сигнала кросс-обратной связи с фильтра противоположной цепи и самого исходного сигнала.

14. Четырехступенчатое стерео эхо

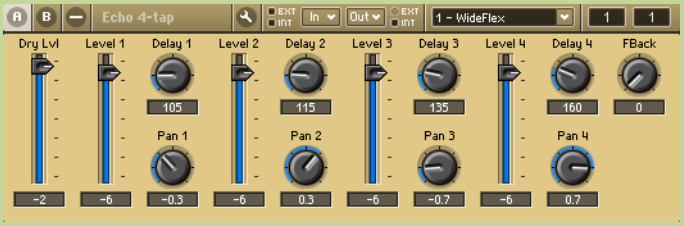


Рис. 14.1. Вид панели управления устройства

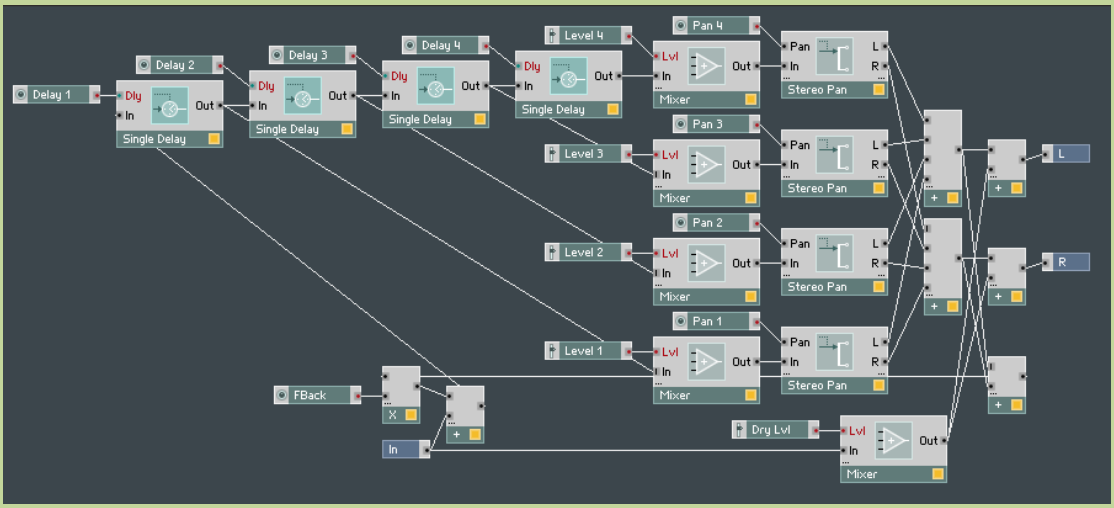


Рис. 14.2. Общая структура устройства

Рассмотрим еще одно устройство создания стереоэха на основе четырех простых делеев, построенное с применением модулей панаромирования Pan. Его панель управления изображена на рис. 14.1, а структура на рис. 14.2.

Устройство состоит из четерехступенчатой линии задержки и четырех одинаковых блоков микширования/панаромирования. У каждого такого блока есть пара соответсвующих регуляторов Level и Pan.

Исходный сигнал поступает на специальный микшер, которым регулируется его уровень в общем миксе, подаваемом на стерео выход. Также исходный сигнал, сложенный с сигналом обратной связи (который предварительно отрегулирован FBack) поступает на линию задержки. Из схемы видно как работает линия задержки. Сигнал от каждого модуля задержки, кроме последнего, поступает в двух направлениях: на соответствующий блок микширования/панаромирования а также на следующий модуль задержки. Время задержки каждого модуля устанавливается соответствующим ему регулятором Delay.

После панаромирования мы имеем уже по 8 сигналов. Теперь все сигналы складываются поканально, то есть вместе 4 правых сигнала, и 4 левых. В итоге получаются два сигнала, каждый из которых идет в двух направлениях: складывается с исходным сигналом и выводится в соответствующий выходной порт (правый или левый) и в общий суммированный сигнал, который используется в качестве сигнала обратной связи.

Эффекты вибрато (хорус, фланжер, фазер)

1. Простой хорус на основе делея и LFO

Рис. 1.1. Вид панели управления устройства

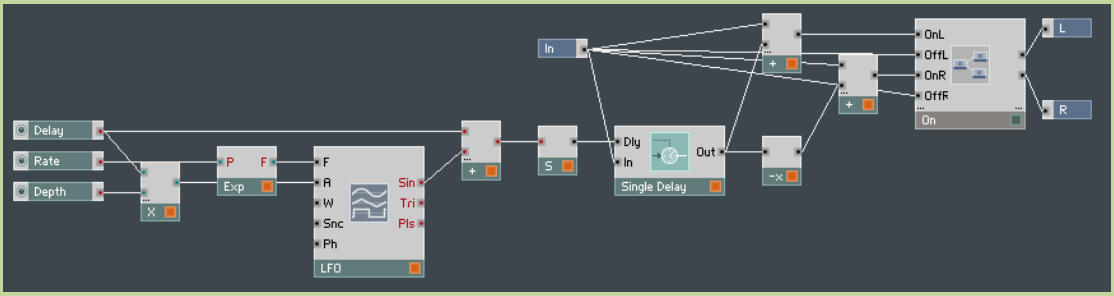


Рис. 1.2. Общая структура устройства

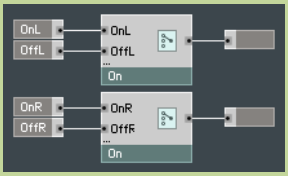


Рис. 1.3. Структура макроса On

Рассмотрим простой эффект хоруса (рис. 1.1 и 1.2). На рис. 1.3 изображен макрос, осуществляющий включение или обход эффекта.

(модуль S это простая задержка на 1 семпл, далее….)

Исходный сигнал складывается с задержанной его копией формируя два канала (сигнал задержки складываемый с одним из каналов R инвертируется), причем время задержки регулируется двумя составляющими – регулятором Delay и модулем LFO. Частота LFO контролируется экспоненциально, а амплитуда – зависит от отдельного регулятора Depth, на который влияет время задержки Delay: чем задержка длиннее, тем сильнее ее влияние на амплитуду.

2. Простой хорус с двумя делеями и LFO

Рис. 2.1. Вид панели управления устройства

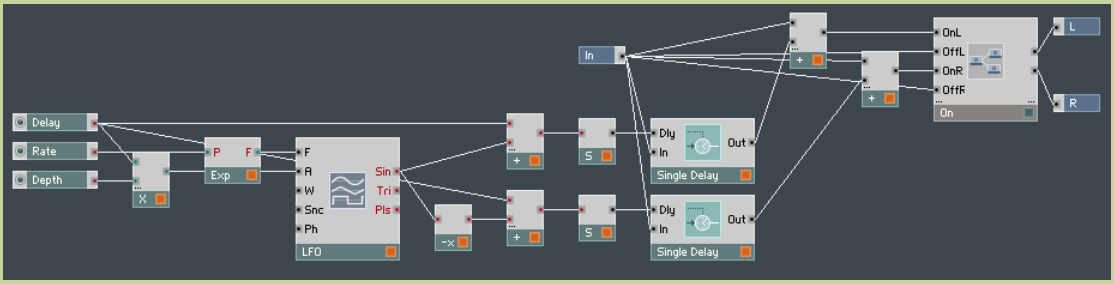


Рис. 2.2. Общая структура устройства

В отличие от предыдущего примера здесь используется два модуля простой задержки. Видно, что на одну задержку сигнал от LFO подается в противофазе к другому. В остальном схема повторяет структуру устройства из предыдущего примера.

3. Хорус с фильтром и обратной связью

Рис. 3.1. Вид панели управления устройства

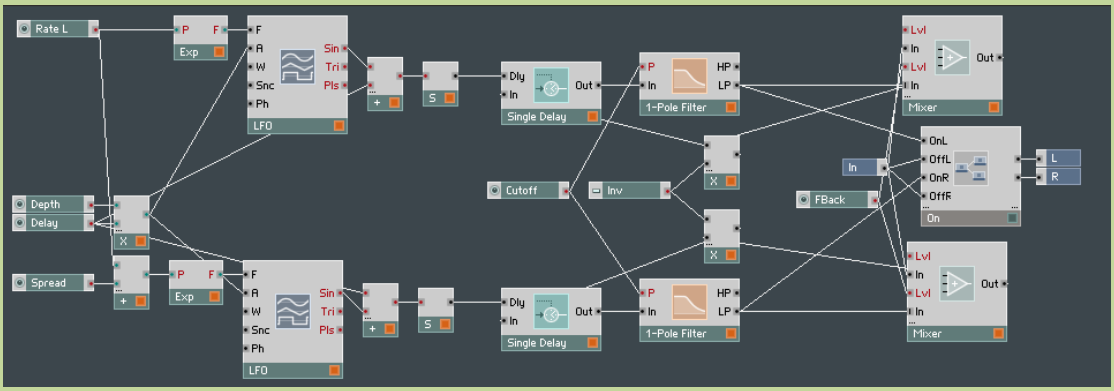


Рис. 3.2. Общая структура устройства

На рис. 3.1 изображена панель, а на рис. 3.2 – структура инструмента, построенного уже на двух LFO. Здесь уже используется поканальная обратная связь.

Вначале рассмотрим схему управления двумя LFO. Видно, что верхний LFO управляется Rate L через приведение к линейному выражению сигнала. Частота второго LFO дополнительно смещена на Spread. Амплитуда для обоих LFO задается регулятором Depth, причем присутствует дополнительная регуляция ее регулятором Delay, осуществляющим регуляцию времени задержки. Впоследствии это время задержки складывается поканально с значением модулей LFO и идет на входной порт в качестве окончательного времени задержки сигнала.

Теперь рассмотрим цепь обратной связи. Для каждого канала она построена на микшере, простом делее и фильтре первого порядка, причем регуляция частоты среза для фильтров (регулятор Cutoff), регуляция уровня обратной связи (FBack) и инвертирования сигнала (регулятор Inv; осуществляет умножение сигнала на значение–1 или 1 в качестве переключателя) осуществляется сразу для обоих каналов.

Итак, сигнал их входного порта идет в двух направлениях: на макрос On, осуществляющий обход сигнала и на два микшера обратной цепи. В микшере сигнал складывается с задержанным сигналом из обратной цепи и пройдя поканально умножение на Inv поступает на модули задержки. Задержанный сигнал фильтруется фильтром низких частот и поступает в два направления: на микшер в качестве сигнала обратной связи и в макрос On в качестве обработанного сигнала.

4. Поканальный стереохорус

Рис. 4.1. Вид панели управления устройства

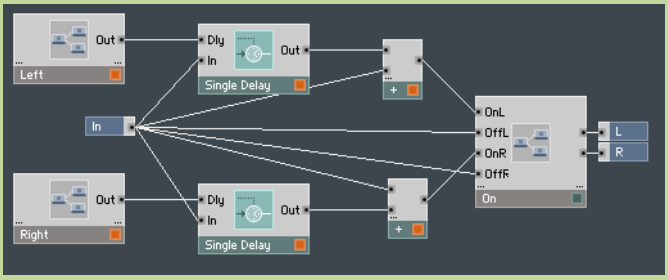


Рис. 4.2. Общая структура устройства

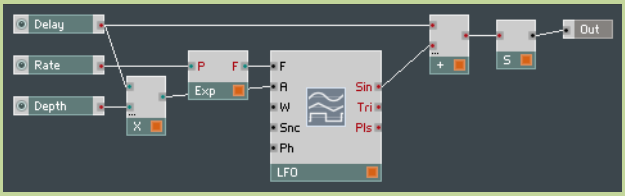


Рис. 4.3. Структура макросов Right и Left

Из схемы 4.2 видно, что сигнал формируется посредством сложения входного сигнала, а также задержанного. Время задержки определяется двумя отдельными макросами имеющими тождественную структуру.

Рассмотрим такой макрос (рис. 4.3). Он состоит из модуля LFO и схемы управления им. Эта структура управления полностью повторяет управление верхним макросом LFO из примера 2.

5. Хорус с 4LFO и делеями

Рис. 5.1. Вид панели управления устройства

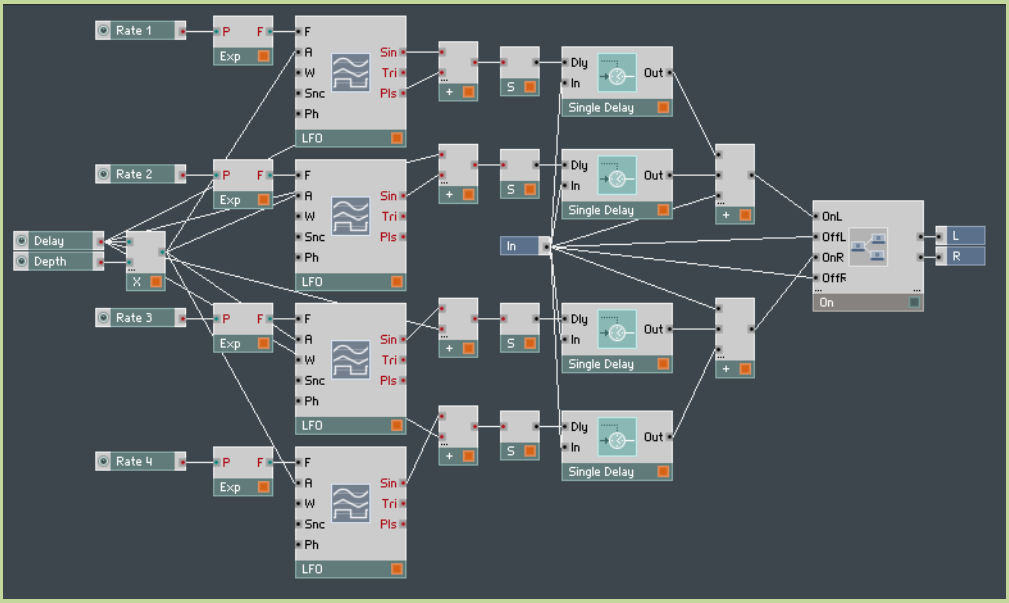


Рис. 5.2. Общая структура устройства

Рассмотрим устройство на основе 4 LFO (рис. 5.1 и 5.2). Идея его схожа с примером 2, только здесь уже не 2 канала а 4. В конечно итоге формируется все же два канала: путем сложения первых двух с исходным сигналом, а также последних двух с ним же. Из особенностей отметим, что частота каждого LFO регулируется отдельным Rate 1..4, а амплитуда для всех LFO общая, и задается такой же схемой управления как и в примере 2.

6. Фазер 1

Рис. 6.1. Вид панели управления устройства

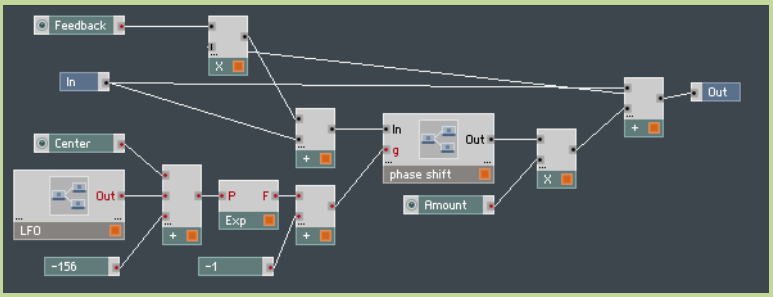


Рис. 6.2. Общая структура устройства

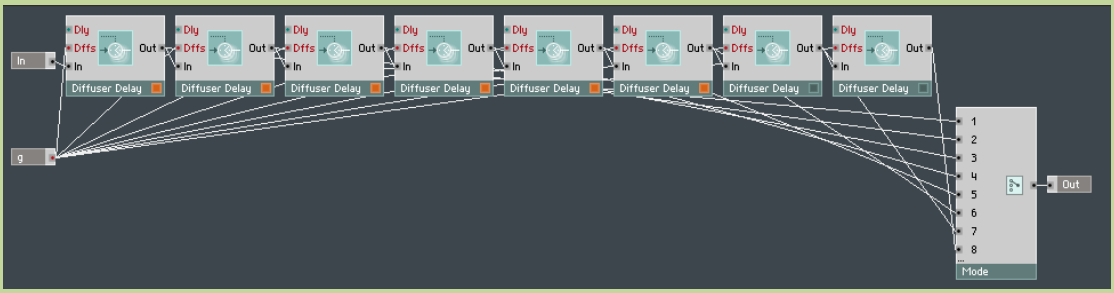


Рис. 6.3. Структура макроса Phase shift

Фазер на основе делеев Diffuser Delay. Панель устройства изображена на рис. 6.1. В общую схему структуры (рис. 6.2) входят два макроса: phase shift (рис. 6.3)и LFO (рис. 6.4 рисунка нет!).

Макрос LFO состоит из модуля LFO и двух регуляторов частоты и амплитуды (LFO Rate и Depth) а также Смушера (!) и преобразователя аудиосигнала в сигнал событий. Макрос входит в схему управления портом g макроса Phase shift. Этот порт регулирует уровень диффузии каждого модуля задержки. Сигнал от макроса LFO складывается с константой –156 и регулятором Center [5,132], преобразуется к линейной и сдвигается дополнительно на –1. Только после этого сигнал поступает на порт g макроса Phase shift.

Рассмотрим макрос Phase shift. Здесь представлено последовательное соединение восьми модулей Diffuse Delay и одного переключателя. На входные порты переключателя идет сигнал с модулей задержек таким образом: на первый порт – только от одного модуля, на второй – от второго и т.д. Видно, что чем более высок номер порта переключателя, то тем более глубже проходит обработку исходный сигнал.

Рассмотрим общую структуру (рис. 6.2). В ней присутствует сигнал обратной связи. Сигнал на входе устройства идет в двух направлениях: на оконечный сумматор (с которого и берется сигнал обратной связи, ее уровень регулируется FeedBack) и складываясь с сигналом обратной связи – на обработку макросом Phase shift. Уровень обработанного сигнала после макроса также регулируется отдельно при помощи Amount [-1,1] и поступает на оконечный сумматор, с которого и выводится из устройства.

7. Фазер 2

Рис. 7.1. Вид панели управления устройства

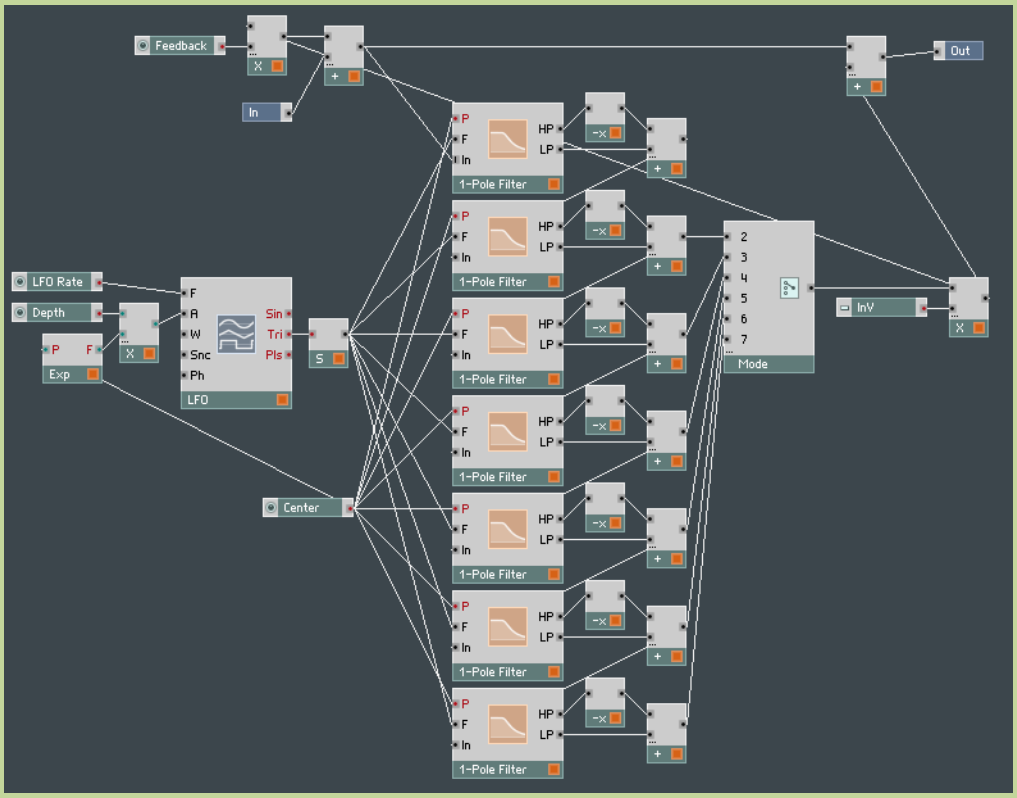


Рис. 7.2. Общая структура устройства

Рассмотрим вариант фазера, собранного на модулях фильтра первого порядка с двойной регуляцией (рис. 7.1 и 7.2). Рисунок в силу громоздкости плохо отражает принцип работы устройства, но он практически полностью повторяет предыдущий пример. Фильтры расположены параллельно один за другим. Сигналы от портов фильтра HP и LP складываются, причем сигнал порта HP дополнительно инвертируется, и общий сигнал переходит уже на входной порт следующего фильтра и т.д. На переключатель Mode идет сигнал от всех таких сумм, кроме суммы после прохождения первого фильтра. Далее сигнал поступает через возможность инвертирования на оконечный сумматор, а также в качестве сигнала обратной связи с регулятором Feedback. В оконечном сумматоре сигнал складывается с исходным и выводится в выходной порт устройства.

Осталось рассмотреть управление фильтрами. Оно производится двухкомпонентно и обще для всех этих фильтров: регулятором Center [40,115] для логарифмического контроля и схемой линейного контроля на основе модуля LFO. Частота LFO регулируется LFO Rate [0.1, 5.1], а амплитуда регулятором Depth [0, 1] значение которого дополнительно умножается на значение того же Center, предварительно конвертированного в линейный сигнал. Сигнал от модуля LFO сглаживается и поступает на порт f всех фильтров.

8. Фленджер с S+H

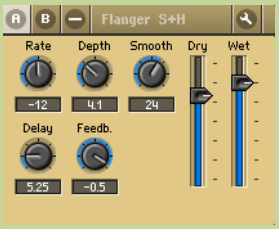


Рис. 8.1. Вид панели управления устройства

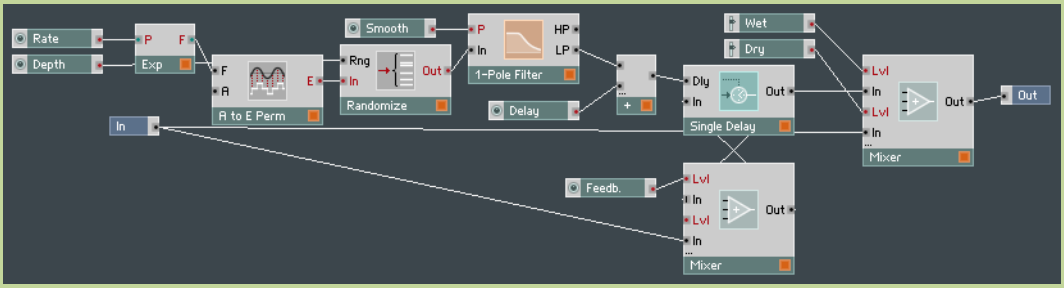


Рис. 8.2. Общая структура устройства

Рассмотрим схему управления временем задержки делея. Сигнал регулятора Rate [-43.5,20] переводится в линейный и служит в качестве источника частоты семплирования модуля A to E Perm (порт A не задействован – значит с нулевой амплитудой). Далее значение рандомизируется в модуле Randomize в диапазоне Depth [0, 12.7]. В итоге получаем, что сигнал располагается в некоторой окрестности нуля. Далее этот сигнал фильтруется с частотой среза Smooth [0, 63.5], складывается с Delay [0, 31.75] и поступает в качестве времени задержки на порт модуля задержки.

Обратная связь реализована на основе этого модуля задержки и микшера. Входной сигнал поступает в двух направлениях: на оконечный микшер и на микшер обратной связи. В микшере обратной связи смешиваются два сигнала – исходный и сигнал обратной связи (с дополнительной регуляцией Feedb [0, -12.7]) и поступает на модуль задержки в качестве входного сигнала.

Оконечный микшер с двумя регуляторами Wet и Dry [-60, 3.5] микширует соответственно сигнал с делея и исходный сигнал, после чего сигнал выводится из устройства.

9. Фленджер с кросс-бэком и эффектом хоруса

Рис. 9.1. Вид панели управления устройства

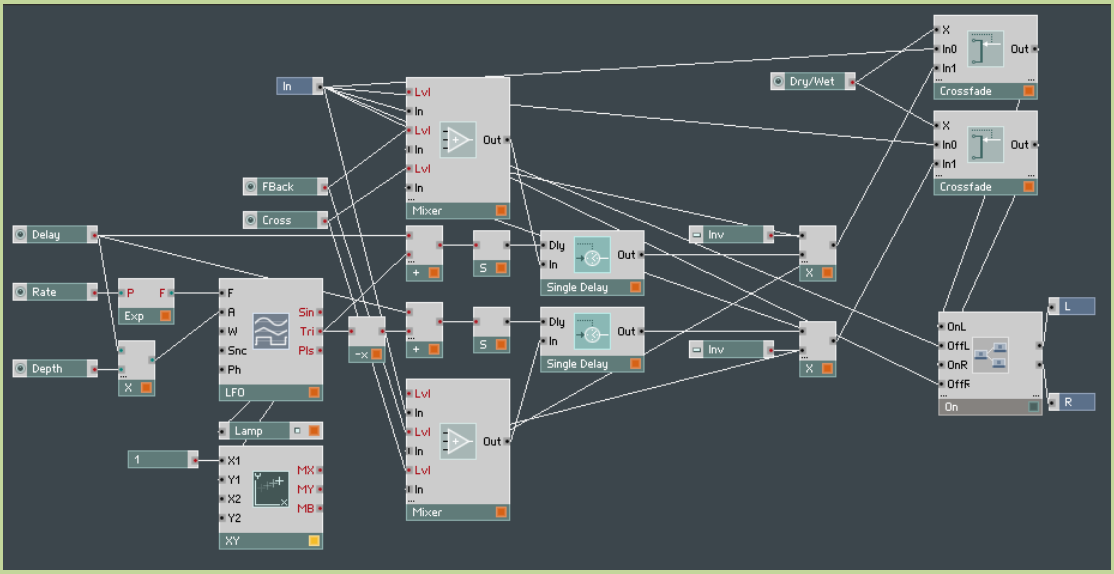


Рис. 9.2. Общая структура устройства

Схема содержит две линии задержки с линиями обратной связи и кроссобратной связи. Схема управления временем задержки повторяет схему из примера 2 (см. рис. 2.2), только здесь используется треугольная волна модуля LFO в отличие от синуса во втором примере. Здесь для визуализации добавлены два модуля – это лампа Lamp и XY. (далее…..)

Исходный сигнал поступает в шести направлениях: два – на обход эффекта в макросе On, два на оконечные Crossfade для смешивания с обработанным сигналом (регулятор Dry/Wet [0, 1]), а также на два микшера линий задержки.

Рассмотрим линию задержки и ее обратные связи. Сигнал с модуля задержки поступает на переключатель-инвертор Inv, где его можно инвертировать; далее сигнал поступает в трех направлениях: на соответствующий Crossfade, далее в качестве сигнала обратной связи на микшер (уровень регулируется Fback [-3, -24]) и в качестве кроссобратной связи на микшер другой линии задержки (регулятор Cross [-3, -24]).

10. Стереофленджер

Рис. 10.1. Вид панели управления устройства

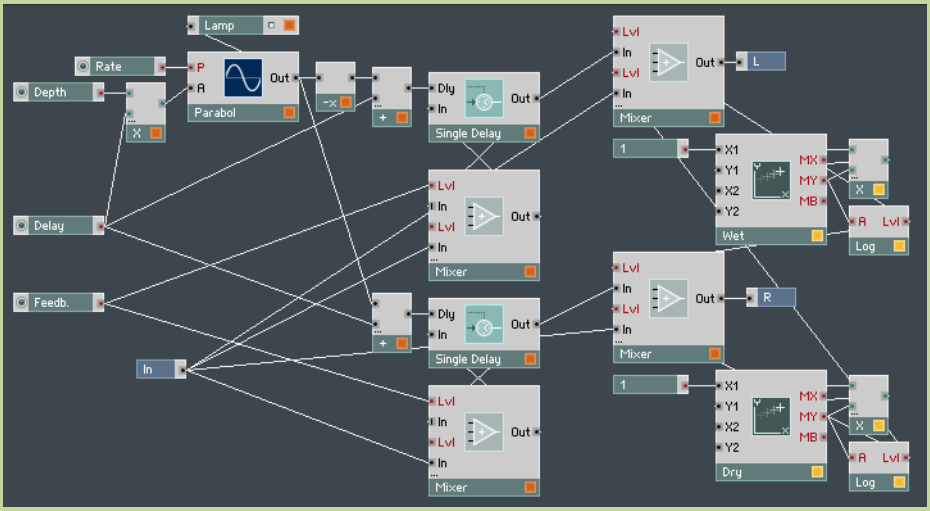


Рис. 10.2. Общая структура устройства

Итак, здесь мы видим структуру, состоящую из двух линий задержки с обратной связью но без кроссобратной, таким образом сигналы двух каналов независимы.

Рассмотрим схему формирования времени задержки для модулей Single Delay. Она построена на основе параболического осциллятора. Регулятор Rate [0, -80] определяет высоту тона осциллятора. Амплитуда задается на основе Depth [0, 0.99] и Delay [0, 31.75], который задает базовое время задержки. Сигнал с осциллятора попадает на лампу для индикации, а также складываясь с тем же Delay попадает в качестве времени задержки на модули задержки (видно, что перед сложением сигнала идущего к верхнему модулю задержки сигнал дополнительно инвертируется).

Входной сигнал идет в четырех направлениях: на два оконечных микшера, и да микшеры цепей обратной связи. С модуля задержки сигнал снимается дважды – на оконечный микшер и на микшер линии задержки в качестве сигнала обратной связи. Этот уровень обратной связи регулируется Feedb. [0, -12] для обоих линий одновременно.

Регуляция Dry/Wet оконечных микшеров построена весьма оригинально: для этого использовались модули XY. Уровень микширования задается координатой Y, приведенной к логарифмической шкале, одновременно для двух микшеров. Микширование обработанного и исходного сигналов осуществляется независимо.

11. Фланджер с фильтром

Рис. 11.1. Вид панели управления устройства

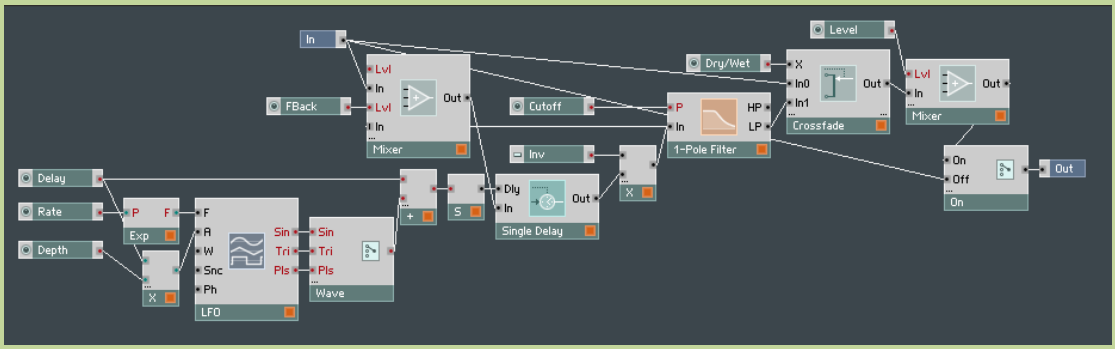


Рис. 11.2. Общая структура устройства

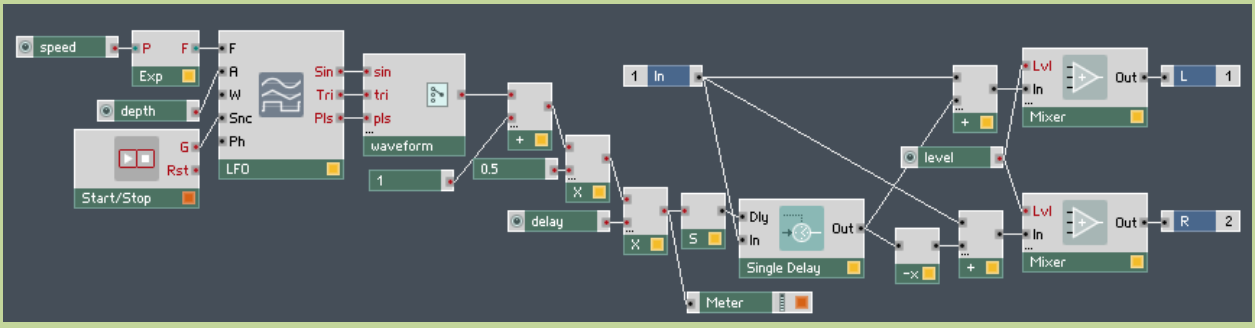
Рассмотрим устройство реализующее фланджер с обратной связью через фильтр. На рис. 11.1 изображена панель управления устройства. Видно что здесь также реализована возможность выбора типа волны LFO. На рис. 11.2 показана структура устройства.

Схема вычисления времени задержки сходна с рассмотренными ранее, за исключанием того, что здесь дополнительно использован переключатель типа волны Wave для модуля LFO.

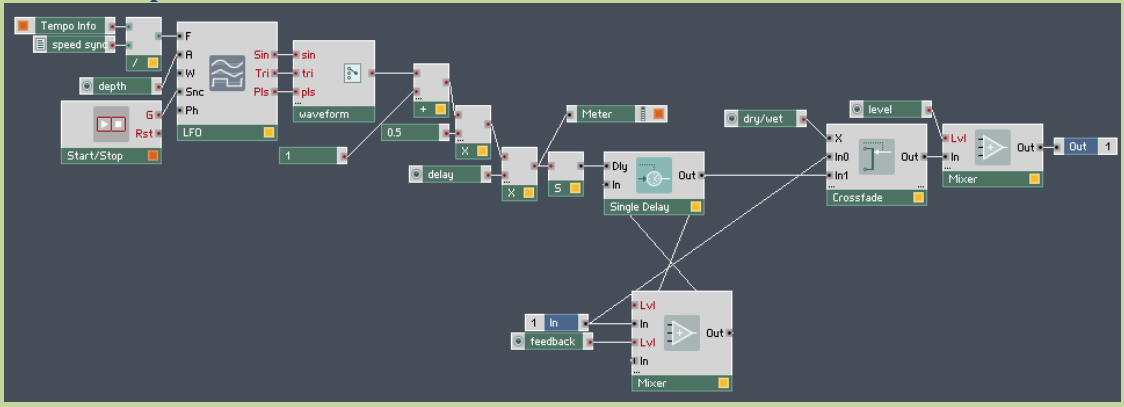
Цепь обратной связи состоит из микшера, модуля задержки и фильтра первого порядка с активным портом LP (фильтр низких частот). Уровень сигнала обратной связи, снимаемого с фильтра, регулируется FBack [0.0001, -24]. Уровень частоты среза фильтра регулируется логарифмически Cutoff [20, 130]. После модуля задержки есть возможность инвертировать сигнал при помощи умножения на Inv.

Входной сигнал на устройство поступает в трех направлениях: в цепь задержки с обратной связью, на оконечный Crossfade а также в селектор On на полный обход эффекта. Crossfade пересекает исходный сигнал и обработанный, снятый с фильтра регулятором Dry/Wet [0, 1]. После Crossfade стоит дополнительный микшер регулирующий уровень сигнала в пределах Level [-40, 20].

Хорус



Фланджер



Аналоговые синтезаторы

В этом разделе мы рассмотрим построение нескольких простых аналоговых синтезаторов.

1. Простейший saw-синтезатор с фильтром и огибающей (раздельными, на каждую ноту) и LFO для вибрато

Рис. 1.1. Панель управления синтезатора

Рассмотрим общую структуру синтезатора, построенного на одном осцилляторе скошенной треугольной волны Sawtooth. Здесь все логически целостные его блоки оформлены в макросы, за исключением модуля самого осциллятора и микшера. Эта структура изображена на рис. 1.2.

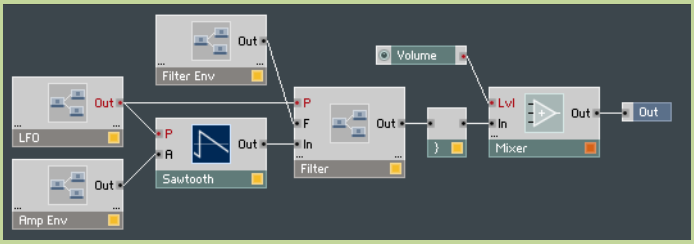


Рис. 1.2. Общая структура синтезатора

Сигнал, генерируемый осциллятором, фильтруется в макросе фильтра, комбинируется и направляется в порт выхода инструмента проходя через микшер.

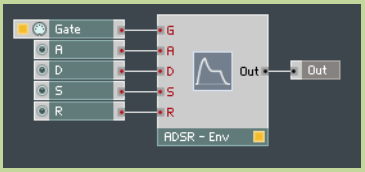


Рис. 1.3. Структура макросов Amp Env и Filter Env

Макросы Amp Env и Filter Env здесь абсолютно идентичны (рис. 1.3). Именно в этих макросах содержится модуль Gate, который осуществляет генерацию импульсов в соответствии с принятыми MIDI-командами Note on и Note off. Точнее, он выдает сигнал, соответствующий силе нажатия на клавишу velocity, то есть амплитуду сигнала в диапазоне 0..1. Но в этих макросах нет модуля, задающего высоту тона (MIDI-ноту).

Элементы управления A, D, R (0..80), S (0..1) соответствуют диапазону значений соответствующих портов модуля ADSR-огибающей, даже больше.

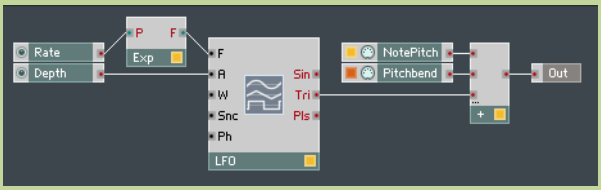


Рис. 1.4. Структура макроса LFO

На рис. 1.4. изображена структура макроса LFO. Здесь происходит следующее: сигнал от модуля LFO складывается с высотой тона ноты, также со смещением колеса pitch.

Амплитуда LFO управляется регулятором Depth ((0..1), глубина), сигнал логарифмического регулятор Rate (-60..3,5) проходя через преобразователь к линейному дает частоту приблизительно (0,25…10). Зачем нужен логарифмический регулятор? Просто потому, что осуществлять точную подстройку им удобнее.

Далее, как мы видим, этот сигнал подается в двух направлениях в качестве сигнала, управляющего высотой тона – на осциллятор и на фильтр.

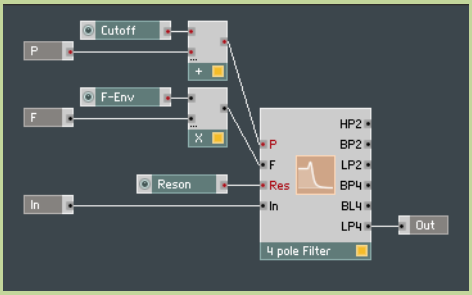


Рис. 1.5. Структура макроса Filter

Главным модулем Макроса фильтра Filter является фильтр-модуль, реализующий фильтрацию 4-порядка (рис. 1.5). Заметим, что здесь используются оба метода контроля частоты среза фильтра – как логарифмического так и линейного. Линейный контроль идет напрямую с макроса Filter Env и реализует динамику обработки сигнала во времени (она придает звуку, проходящему через фильтр окраску, как бы «скольжение»). Это управление от высоты тона ноты не зависит. Сигнал проходит через управляющую цепь – умножитель, с помощью которого задается количество модуляции частоты среза.

Логарифмический контроль. На него подается сигнал с макроса LFO, как мы знаем, содержащий три компонента (высоту ноты, смещение колеса pitch и непосредственно низкочастотную модуляцию). Этот сигнал – также динамичен во времени периодически, за счет LFO.

Итак, фильтр на панели имеет два органа управления частотой среза: постоянную Cutoff (-10..80, смещение значения среза фильтра определяется в полутонах от взятой ноты) и динамическую F-Env (0..10000, уровень «влияния» макроса Filter Env на управление фильтром, при 0 – это управление фактически отсутствует). Первая как бы служит точкой базового отсчета относительно взятой ноты (плюс периодические изменения LFO), а вторая выражает динамику во времени.

Уровень резонанса фильтра Reson (0..0,98) управляется независимо.

2. Одноосцилляторный синтезатор с 2 фильтрами

Рис. 2.1. Вид панели управления синтезатора

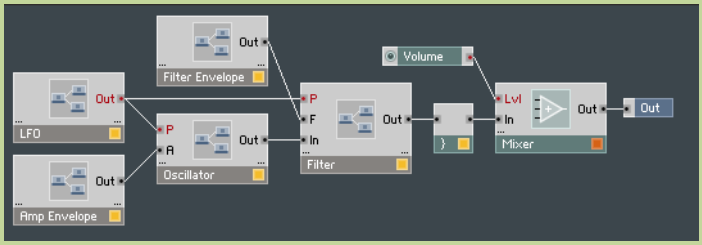


Рис. 2.2. Общая структура синтезатора

Рассмотрим более продвинутую версию синтезатора, созданного на основе схемы предыдущего примера. Его общая структура изображена на рис. 2.2. Здесь мы реализуем такие дополнительные возможности: выбор типа волны осциллятора из трех вариантов, смещение ноты на октаву вверх и вниз, естественный овердрайв, а также сдвоенный фильтр.

Макросы огибающих и LFO остаются без изменений. Основные изменения произошли в макросе Filter а также добавлен новый макрос Oscillator. Его вначале и рассмотрим (рис 2.3).

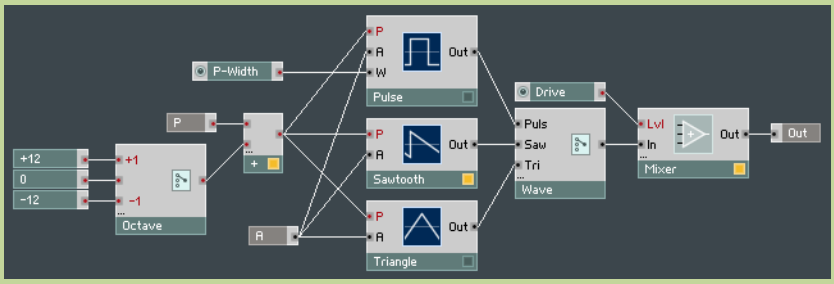


Рис. 2.3. Структура макроса Oscillator

Основу макроса составляют три модуля различного типа волны (Pulse, Sawtooth, Triangle), включенные параллельно. Причем одновременно может быть активен только единственный модуль. Это обеспечивается кнопочным переключателем Wave. Аналогичный переключатель Octave служит для выбора транспонирования ноты на октаву вверх и вниз. Это смещение – и значение выходного порта переключателя Octave складывается со значением высоты тона ноты P, и в дальнейшем посылается сразу на все три осцилляторных модуля. Амплитуда (порт A) также передается всем этим модулям на соответствующий порт без каких-либо преобразований. Для модуля Pulse доступен также регулятор ширины импульса P-Width (0..0,98). И последний модуль – это микшер. Регулятор уровня усиления назван Drive. Почему? Посмотрим на

диапазон его значений. Они от –40 до 20! Именно завышенный уровень микшера и осуществляет здесь эффект перегруза.

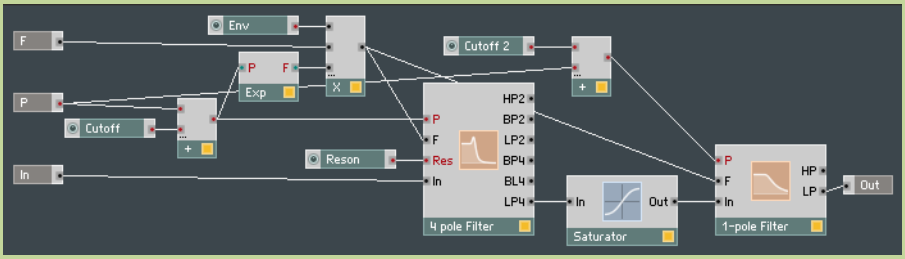


Рис. 2.4. Структура макроса Filter

На рис. 2.4. изображена структура макроса Filter. Здесь добавилось два основных модуля – еще один фильтр низких частот первого порядка и Saturator (сглаживатель).

Рассмотрим механизмы контроля частотами срезов фильтров. Линейный контроль для обоих фильтров одинаков, но в отличие от предыдущего примера расширен. Теперь линейный контроль управляется не только регулятором Env, но частично и высотой тона, передающегося портом P. При этом высота тона смещается регулятором Cutoff (-20..60) и линеализируется.

Логарифмический контроль обоих фильтров осуществляется также совместно, но при этом для них задаются два различных смещения в полутонах: уже известное нам Cutoff для первого фильтра четвертого порядка и Cutoff2 (0..80) для второго фильтра.

Сигнал после первого фильтра проходит через модуль Saturate на второй фильтр и далее – в выходной порт. Этот промежуточный модуль производит мягкий овердрайв ограничивая выход амплитуды в +-2 цифровых отсчета для каждой входной амплитуды большей +-4. Это немного смягчает «ударность» звука овердрайва, который получен на выходе макроса осцилляторов.

Так как амплитуда сигнала, особенно при использовании перегруза, часто выходит за границы (-1..1), то для того, чтобы не создавать уже паразитные перегрузки при подаче сигнала на звуковую карту компьютера, звуковым микшером, установленным перед выходным портом синтезатора регулируется уровень сигнала (чаще на понижение).

3. Простой аналоговый синтезатор классической схемы (2 осциллятора, один фильтр и усилитель)

Рис. 3.1. Вид панели управления синтезатора

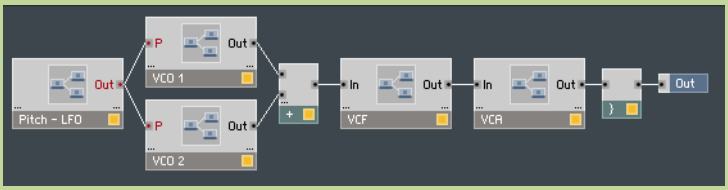


Рис. 3.2. Общая структура синтезатора

Теперь рассмотрим классическую двухосцилляторную структуру, которая часто характеризует принцип построения настоящего железного устройства (имеется в виду идеология модульности), управляющая панель которой изображена на рис. 3.1. В самой структуре (рис. 3.2) присутствуют: блок Pitch-LFO, два осциллятора, включенные параллельно, сумматор, VCF – фильтр, VCA – усилитель. Прямо перед выходным портом включен сумматор.

Макрос Pitch-LFO (рис. 3.3) и здесь не претерпел никаких изменений, и полностью повторяет структуру макроса LFO из предыдущих примеров.

Рис. 3.3. Структура макроса LFO

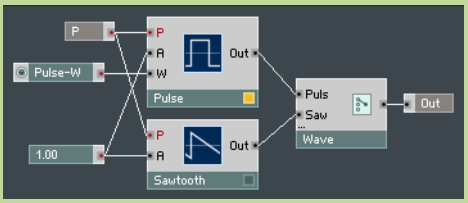


Рис. 3.4. Структура макроса VCO 1

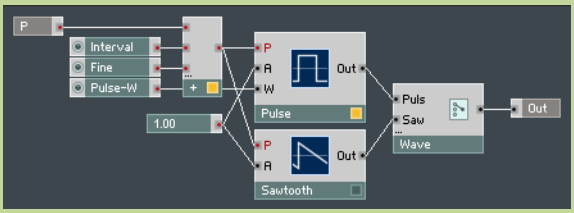


Рис. 3.5. Структура макроса VCO 2

На рисунках 3.4 и 3.5 изображены схемы структур двух осцилляторов. Видно, что они различаются весьма незначительно: во втором осцилляторе всего лишь расширено управление высотой тона. Дополнительные регуляторы Interval и Fine обеспечивают смещение высоты тона второго осциллятора относительно высоты тона первого. Заметим, что в отличие от предыдущих примеров амплитуда сигнала для осцилляторов задается константой, то есть от силы нажатия на MIDI-клавишу не зависит. Для модуля Pulse доступен регулятор ширины импульса Pulse-W(0..0,98). Переключатель Wave здесь осуществляет переключение сигнала от одной формы волны к другой.

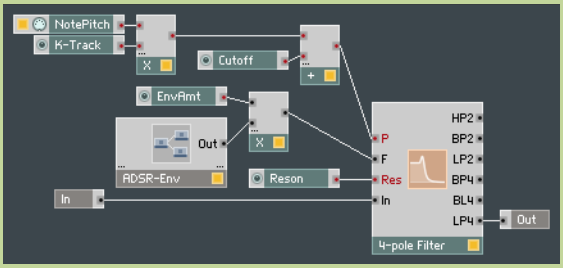


Рис. 3.6. Структура макроса VCF

После этого сигналы выходных портов модулей VCO1 и VCO2 складываются и поступают на фильтр. Структура макроса фильтра (рис. 3.6) напоминает структуру фильтра первого примера, но, во-первых внутрь этого макроса внедрен подмакрос огибающей ADSR-Env (полностью аналогичный ранее рассмотренному) и здесь же еще раз используется модуль Note Pitch. (многократное использование таких передающих модулей – не возбраняется: любая копия передает полностью соответствующие сигналы).

Линейный контроль частотой среза фильтра осуществляется посредством огибающей – и вот уже здесь, на фильтр – используется влияние MIDI-контроллера Velocity посредством уровня сигнала генерируемого Gate, который находится в ADSR-Env. EnvAmt здесь – контролирует уровень этого влияния огибающей на частоту среза фильтра.

Логарифмический контроль фильтра – его уровень контролирует регулятор K-Track – может быть смещен относительно высоты тона нажатой ноты регулятором Cutoff.

Некоторые сообразительные читатели, наверное, уже догадались, что, так как амплитуда осцилляторов уже установлена константой и совершенно не зависит от сигнала Gate, то синтезатор должен издавать звуки непрерывно вне зависимости от нажатия и отпускания клавиши. Это так и есть. Но почему же не слышно звуков? Секрет скрыт в последнем макросе VCA.

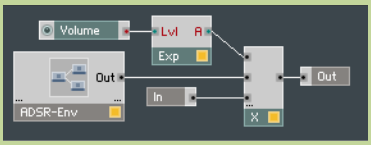


Рис. 3.7. Структура макроса VCA

Рассмотрим его повнимательнее (рис. 3.7). Оказывается, поступающий с фильтра сигнал не только регулируется Volume, но в этом макросе также расположен уже известный нам подмакрос ADSR-Env. И именно здесь, на этой последней стадии перед выводом сигнала вы порт выхода – регулируется амплитуда выходного сигнала. Для этого все эти три сигнала перемножаются. Таким образом осуществляется модуляция конечного сигнала огибающей (заметим, уже фильтрованного!) и, по совместительству, именно наличие нуля в этом умножении “гасит” сигнал в нулевую амплитуду при отсутствии нажатий на MIDI-клавиши. Таким образом, эти две схемы построения синтезаторов 1-2 примера и рассматриваемого сейчас имеют принципиальные различия в подходе построения.

4. Двухосцилляторный импульсный синтезатор

Рис. 4.1. Вид панели управления синтезатора

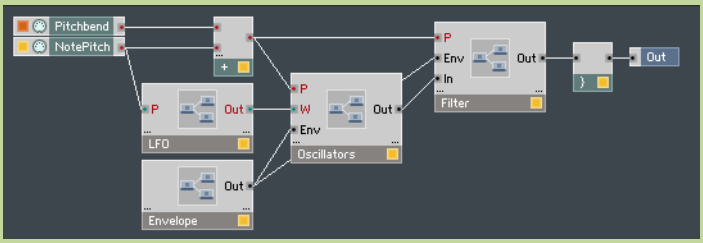


Рис. 4.2. Общая структура синтезатора

Сейчас рассмотрим импульсный двухволновый синтезатор. Его панель приведена на рис. 4.1, а структура на рис. 4.2. Синтезатор хорошо синтезирует звуки типа Pad.

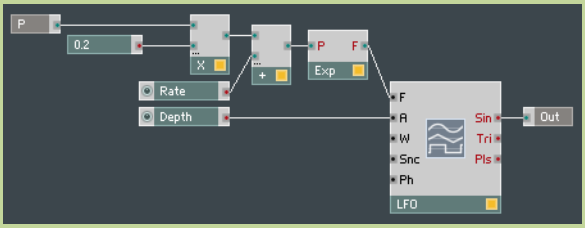


Рис. 4.3. Структура макроса LFO

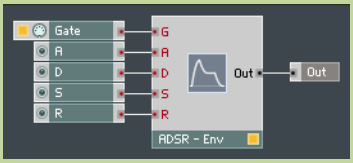


Рис. 4.4. Структура макроса Envelope

Особенность этого синтезатора – это наличие всего лишь одной общей для осцилляторов и фильтра огибающей. Ее макрос – тот же как и в предыдущих примерах (рис. 4.4). Но структура макроса LFO немного изменена (рис. 4.3): туда добавлена регулирующая константа со значением 0,2, ослабляющая влияние на частоту, подаваемую на входной порт модуля LFO от высоты тона взятой MIDI-клавиши. Это сделано потому что в таких звуках как пады (их еще называют «подкладки», потому что их обычно используют как бы в виде гармонического фона) резкие и быстрые перепады не нужны, а нужна плавность и медленное течение звука. Таким образом если регулятор Rate, обеспечивающий увеличение и уменьшение частоты будет установлен в ноль, то период колебаний LFO будет равен эквиваленту линеаризованной частоты «номер ноты»/5. Например, если взятая нота будет 50, то частота будет равна... (привести пример). Амплитуда задается независимым регулятором Depth.

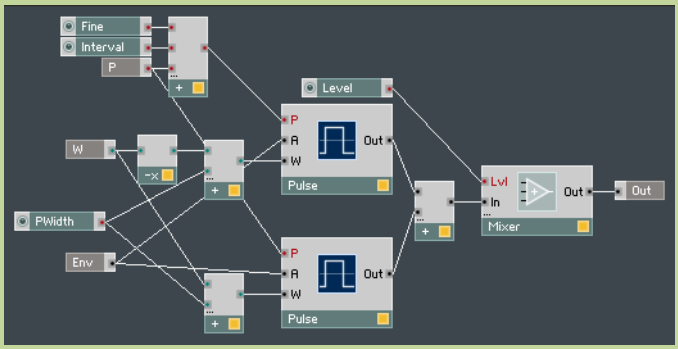


Рис. 4.5. Структура макроса Oscillators

Рассмотрим макрос Oscillators (рис. 4.5). Основу его составляют два импульсных осциллятора включенные параллельно и микшер. Амплитуда на осцилляторы передается из макроса Envelope. Рассмотрим логарифмический контроль высоты тона осцилляторов. Он практически эквивалентен методу, использованному в примере 4: на один из осцилляторов подается сигнал (номер MIDI-ноты) непосредственно с порта P (что есть ничто иное как номер ноты + значение контроллера колеса сдвига тона), на второй же осциллятор – сигнал смещенный регуляторами Interval и Fine.

Модулирующий сигнал от LFO поступает на входной порт W макроса Oscillators, но используется для модуляции высоты тона и амплитуды. Что же модулируется? Оказывается – ширина импульса. Причем таким образом: регулятор PWidth задает базовое значение ширины импульса, а сигнал от LFO осуществляет плавное движение в некоторой окрестности этого базового значения. Причем, на один осциллятор сигнал идет напрямую, а на второй – инвертируется (то есть в противофазе).

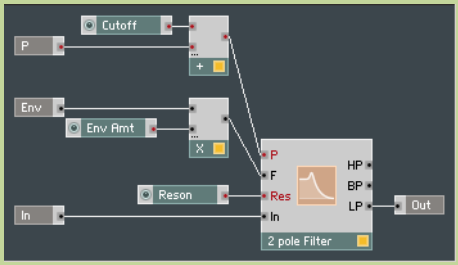


Рис. 4.6. Структура макроса Filter

Теперь рассмотрим фильтр. Его структура (рис. 4.6) полностью повторяет структуру фильтра из примера 1, за исключением того, что здесь используется фильтр второго порядка (в режиме LP, то есть фильтра низких частот).

И, наконец, сигнал проходя через смеситель выводится в выходной порт синтезатора. Заметим, что регулятор уровня сигнала после смесителя отсутствует. Это связано с тем, что уровень сигнала для приведенного синтезатора хорошо регулируется еще в макросе Oscillators и практически не требует дальнейшего микширования.

5. Двухосцилляторный синтезатор с мультимодовым фильтром и двумя огибающими

В этом примере мы рассмотрим двухосцилляторный синтезатор с мультимоновым фильтром – то есть режим фильтрации сигнала будет выбираться пользователем. Другая особенность этого синтезатора – балансовый микшер двух осцилляторов.

Рис. 5.1. Вид панели управления синтезатора

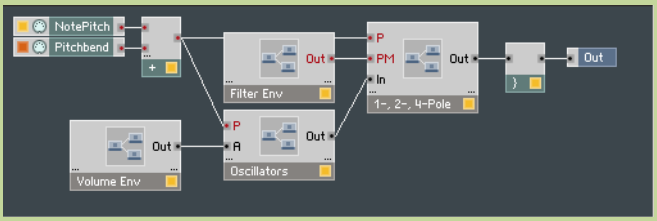


Рис. 5.2. Общая структура синтезатора

Рассмотрим панель управления (рис. 5.1) и общую структуру синтезатора (рис. 5.2). Сразу бросается в глаза переключатель режима работы фильтра. Он выполнен в виде ряда кнопок переключения.

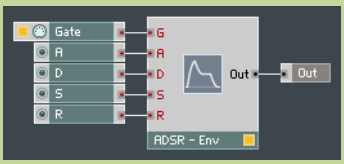


Рис. 5.3. Структура макроса Envelope

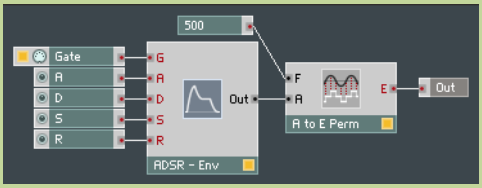


Рис. 5.4. Структура макроса Filter Envelope

Общая структура следующая: высота тона ноты суммируется с показанием MIDI-контроллера колеса Pitch и посылается на осцилляторы и фильтр. В структуре также присутствуют две раздельные огибающие – для макроса осцилляторов (Volume Env) и для фильтра (Filter Env). Макрос огибающей осцилляторов Volume Env (рис. 5.3) полностью идентичен ранее рассмотренным. Макрос Filter Env (рис. 5.4) имеет отличие: перед выходом на порт выхода макроса сигнал поступает на A to E perm (---).

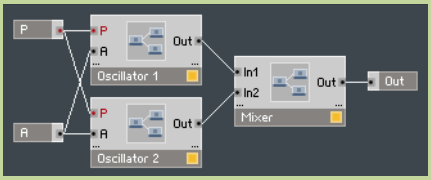


Рис. 5.5. Структура макроса Oscillators

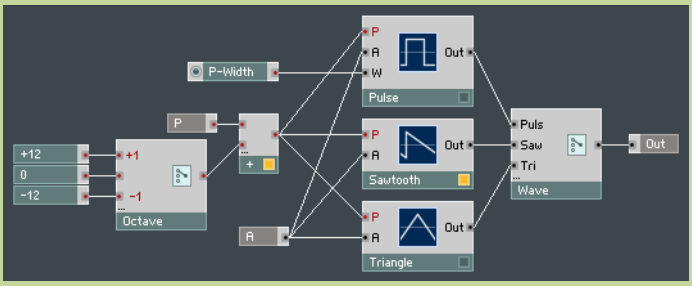


Рис. 5.6. Структура подмакроса Oscillator 1 в макросе Oscillators

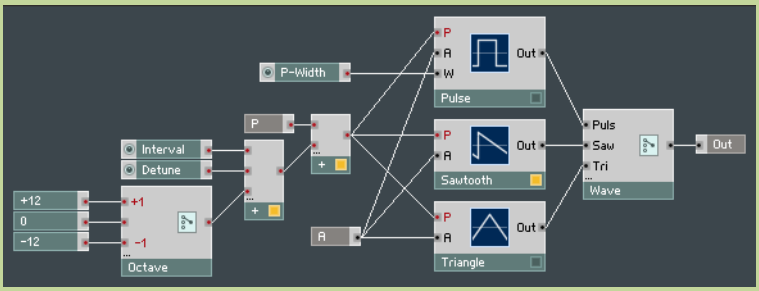


Рис. 5.7. Структура подмакроса Oscillator 2 в макросе Oscillators

Макрос осцилляторов изображен на рис. 5.5. Он служит только контейнером для дальнейшей структуризации – в макросах Oscillator 1, Oscillator 2 и Mixer. Структура первых двух подмакросов изображена на рис. 5.6 – 5.7. Можно заметить, что первый макрос очень похож на рассмотренный нами ранее в примере номер 2 (см. рис. 2.3). Действительно, его структура аналогична, за исключением того, что сейчас в нем нет микшера. В макросе Oscillator 2 всего лишь добавлена возможность смещения тона (которое, как видно, складывается со смещением тона переключателя Octave). Такой механизм также был рассмотрен в предыдущем примере номер 3.

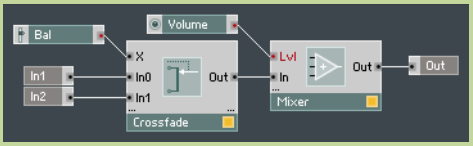


Рис. 5.8. Структура подмакроса Mixer в макросе Oscillators

Подмакрос Mixer содержит модуль Crossfade, а также обычный микшер, регулирующий уровень сигнала. Регулятор Bal модуля Crossfade делает вот что: он определяет в какой пропорции сигналы этих двух осцилляторов будут смешиваться. Например, если значение равно 0, то модуль Crossfade пропустит только сигнал первого осциллятора и не пропустит сигнал второго, и наоборот – если 1. В промежуточном положении модуль Crossfade микширует сигнал в соответствии с весами: x для первого осциллятора и 1-x для второго.

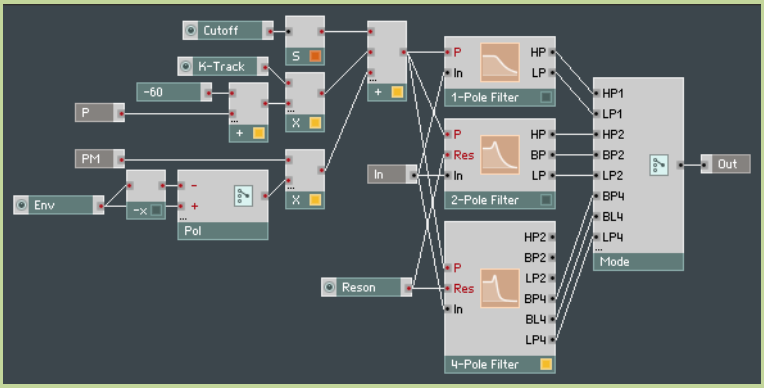


Рис. 5.9. Структура макроса Filter

После макроса Oscillators сигнал идет на фильтрацию в макрос Filter (рис. 5.9). В структуру макроса входят три фильтра. Это фильтры первого, второго и четвертого порядка. Здесь, в отличие от предыдущих примеров, используются варианты модулей фильтров только с логарифмической модуляцией частоты среза. (м.б. это уменьшает нагрузку на процессор???) Рассмотрим это управление. Оно складывается из трех составляющих: сигнала порта P (номер ноты), сигнала приходящего с огибающей фильтра Filter Env а также независимого регулятора Cutoff, задающего постоянный компонент. Из сигнала с порта P дополнительно вычитается 60. Влияние огибающей задается регулятором Env, причем его значение может быть инвертировано путем переключения модулем Pol (полярность).

Сигналы от выходных портов фильтров переключаются модулем Mode, отвечающим за выбор текущего режима фильтрации.

6. Три осциллятора с мультимодовым фильтром при использовании микшера

Данный синтезатор построен по принципу структуры из третьего примера. Здесь сигналы от трех осцилляторов микшируются в дополнительном макросе Mixer. Общий сигнал фильтруется мультимодовым фильтром и проходит через макрос усилителя Amp, аналогичный макросу на рис. 3.7.

Рис. 6.1. Вид панели управления синтезатора

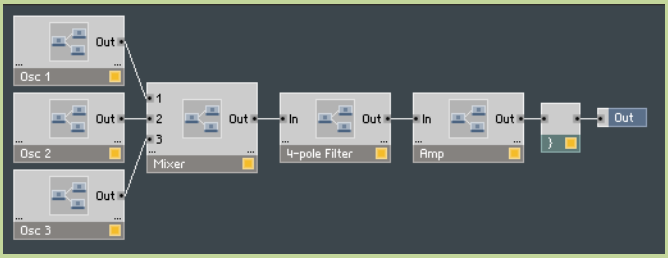


Рис. 6.2. Общая структура синтезатора

Панель управления синтезатора изображена на рис. 6.1, а общая структура – на рис. 6.2.

В качестве осцилляторов использованы три одинаковых структуры, дающие возможность выбрать два типа волны: Pulse или Sawtooth (рис. 6.3). Также в осцилляторе присутствует подстройка уровня Pitch и Fine.

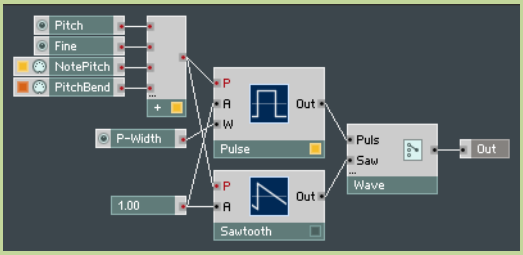


Рис. 6.3. Структура макросов Ocs 1, Osc 2 и Osc 3

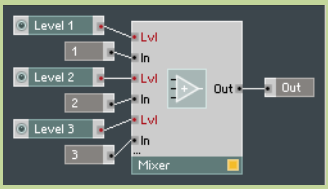


Рис. 6.4. Структура макроса Mixer

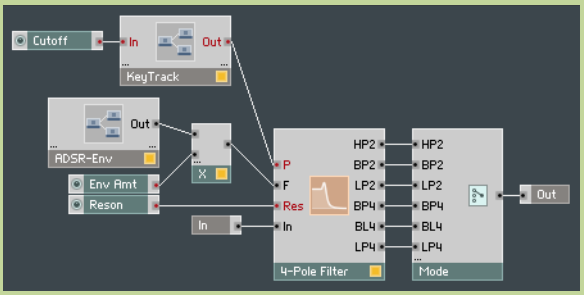


Рис. 6.5. Структура макроса 4-pole Filter

Макрос Mixer представляет собой трехканальный одноименный модуль с тремя регуляторами уровня входящих сигналов (рис. 6.4).

Рассмотрим макрос, содержащий фильтр (рис. 6.5). Здесь использован модуль фильтра четвертого порядка с двумя механизмами контроля частоты среза (линейным и логарифмическим).

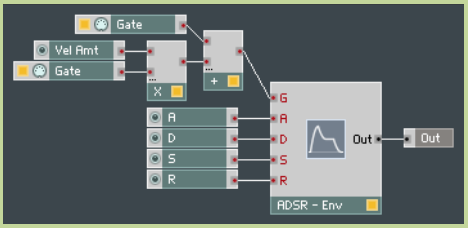


Рис. 6.6. Структура подмакроса ADSR-Env в макросе 4-pole Filter

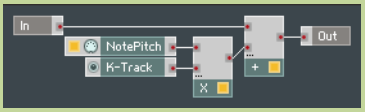


Рис. 6.7. Структура подмакроса KeyTrack в макросе 4-pole Filter

Линейный контроль осуществляется посредством получения сигнала из подмакроса ADSR-Env (рис. 6.6). В этом подмакросе два модуля Gate, причем значения выходных портов их складываются. Уровень одного из значений контролируется регулятором Vel Amt.

Подмакрос KeyTrack, при помощи которого осуществляется контроль логарифмическим портом частоты среза, ни что иное как кусок структуры фильтра из примера 3 (см. рис. 3.6), оформленный в отдельный макрос для наглядности.

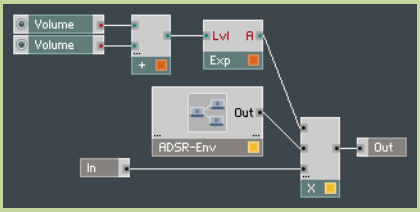


Рис. 6.8. Структура макроса Amp

Переключатель Mode позволяет выбрать необходимый сигнал и переслать его в макрос усилителя Amp.

7. Трехволновый синтезатор со сдвоенным фильтром и 2 огибающими

Рассмотрим еще одну модификацию трехосцилляторного синтезатора. В этом синтезаторе два фильтра со смещением.

Рис. 7.1. Вид панели управления синтезатора

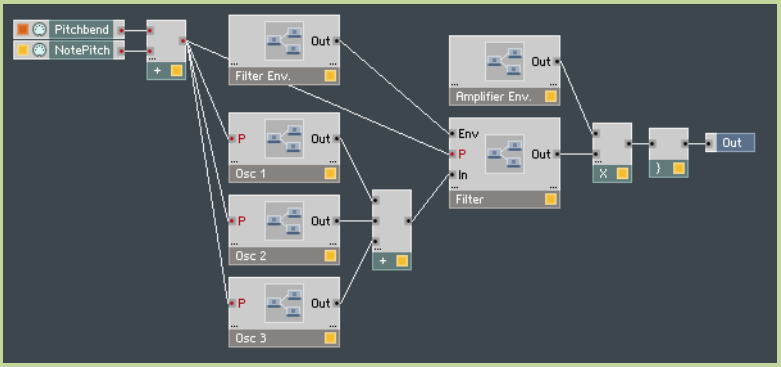


Рис. 7.2. Общая структура синтезатора

Панель управления синтезатором изображена на рис. 7.1. В общую структуру (рис. 7.2) входят помимо трех одинаковых осцилляторов и фильтра два одинаковых макроса огибающих Amplifier Env и Filter Env (см., например, рис. 1.3).

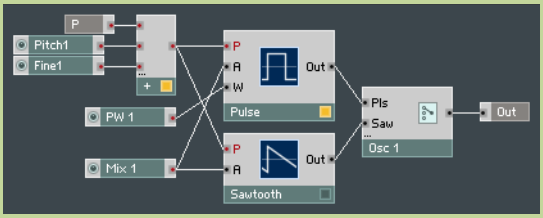


Рис. 7.3. Структура макросов Ocs 1, Osc 2 и Osc 3

Макрос осциллятора изображен на рис. 7.3. Его структура подобна структуре осциллятора из предыдущего примера, только амплитуда задается не константой, а регулятором (они названы Mix1, Mix2, Mix3 соответственно для первого, второго и третьего осциллятора). По этой причине в синтезаторе нет какого-либо дополнительного микшера. Также модули NotePitch и Pitchband вынесены в основную структуру, тогда как в предыдущем примере они находились в каждом осцилляторе.

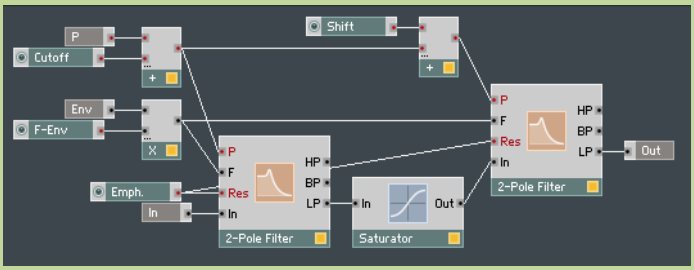


Рис. 7.4. Структура макроса Filter

После суммирования трех волн, сигнал фильтруется (рис. 7.4). Макрос фильтра состоит из двух фильтров второго порядка (с двойным контролем частоты среза) и сатуратором, расположенным между ними. Линейный контроль осуществляется при помощи огибающей фильтра Filter Env, согнал которой дозируется регулятором F-Env. Этот контроль един для обоих фильтров. Логарифмический контроль осуществляется по схеме: высота тона (номер ноты) складывается с регулятором Cutoff и идет на порт P первого фильтра, а также этот сложенный сигнал еще раз смещается другим регулятором Shift и проходит уже на второй фильтр. Фактически, на входной порт P второго фильтра сигнал приходит дважды смещенным: первый раз относительно значения ноты и второй раз – относительно смещения первого фильтра.

8. Синтезатор на основе модулей step

Рассмотрим весьма специфический синтезатор, построенный на основе модулей Step (далее рассказать об этих модулях).

Макрос огибающей Envelope здесь такой же стандартный как, например, на рис. 1.3.

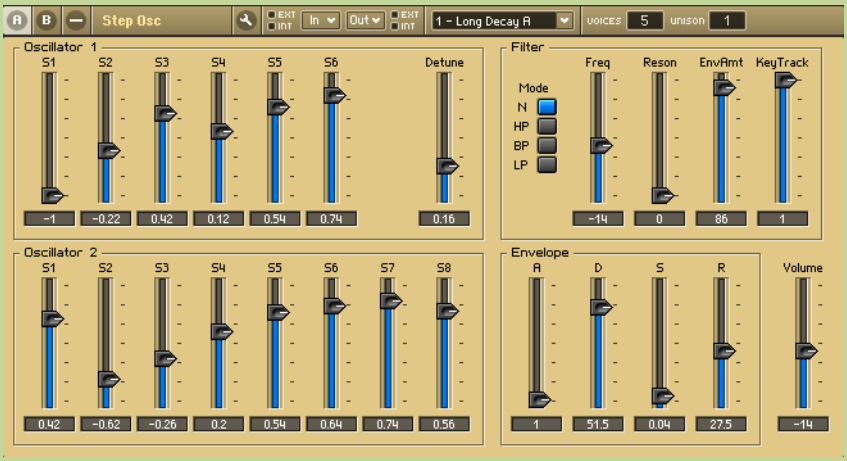


Рис. 8.1. Вид панели управления синтезатора

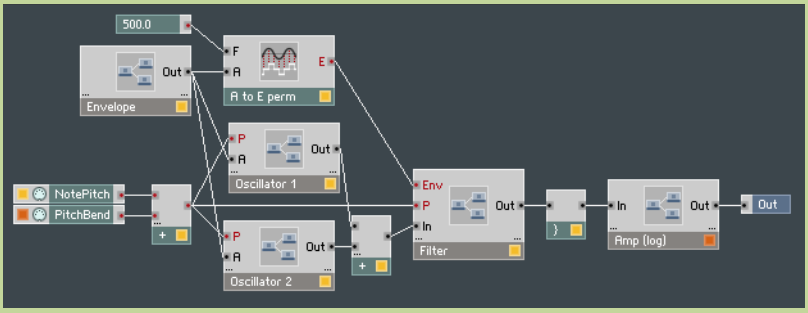


Рис. 8.2. Общая структура синтезатора

В построении общей структуры синтезатора (рис. 8.2) ничего особенного не наблюдается. Его панель изображена на рис. 8.1. Основными объектами рассмотрения будут являться макросы осцилляторов (рис. 8.3 и 8.4) и макрос фильтра (рис. 8.5).

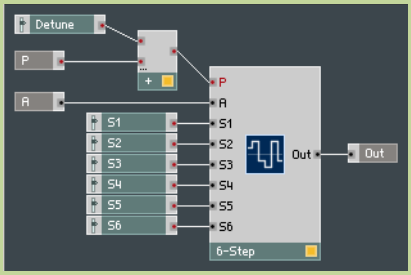


Рис. 8.3. Структура макроса Oscillator 1

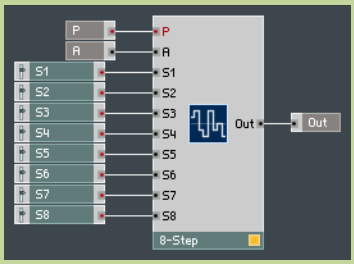


Рис. 8.4. Структура макроса Oscillator 2

Макросы осцилляторов состоят практически из одного модуля – это 6-step и 8-step, модулей-регуляторов. Также в Oscillator 1 предусмотрено управление смещением высоты тона ноты (регулятор Detune).

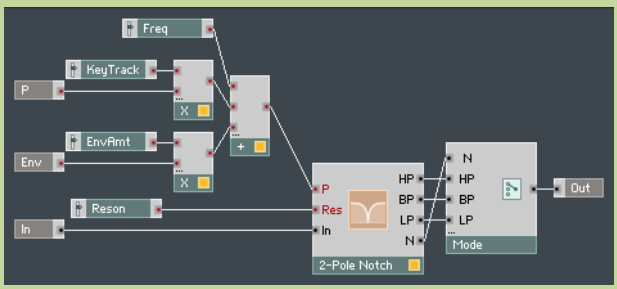


Рис. 8.5. Структура макроса Filter

Фильтр построен на основе модуля 2-pole Notch (вырезает частоты). Он имеет только логарифмическое управление уже известного нам

трехкомпонентного метода: фиксированный Freq, сигнал с порта P контролируется KeyTrack, а сигнал с огибающей – регулятором EnvAmt.

Также присутствует переключатель Mode, который выводит один выбранный сигнал с фильтра.

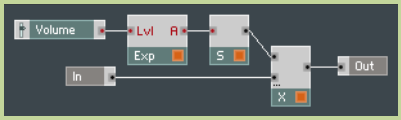


Рис. 8.6. Структура макроса Amp

И, наконец, макрос Amp (рис. 8.6) осуществляет логарифмическое регулирование выходного сигнала синтезатора.

FM-синтезаторы

1. Простейший FM-синтез с двумя операторами FM 2 ops

Рассмотрим простой синтезатор, реализующий идею частотной модуляции. Оператором в таком синтезе обычно называется элемент, содержащий в себе огибающую и осциллятор

.

Рис. 1.1. Вид панели управления синтезатора

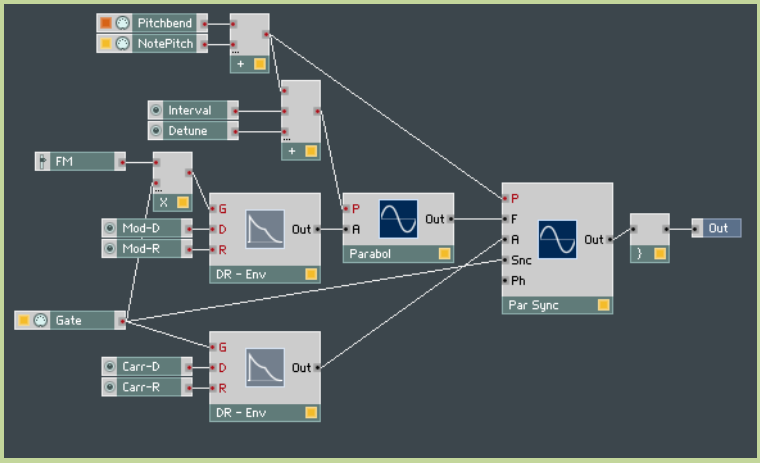


Рис. 1.2. Общая структура синтезатора

Панель управления синтезатора можно видеть на рис. 1.1.

Общая структура (рис. 1.2) выполнена без применения структуризации с помощью макросов. Итак, мы реализуем двухоператорный синтез, следовательно видим перед собой две огибающих, два осциллятора а также элементы схемы управления ими. Для того чтобы осуществить частотную

модуляцию, один из осцилляторов должен иметь порт, на который подается линейный сигнал от второго. Так и есть. При этом оператор, на который подается модулирующий сигнал называется “несущей” (волной), а второй оператор – “модулятором”.

Рассмотрим схему управления осцилляторами. На рисунке 1.2 видно, что на несущую подается исходный NotePitch+Pitchband сигнал, тогда как на модулятор к сигналу применяется смещение регуляторов Interval и Detune. Сигнал Gate подается в три направления: двум огибающим типа DR-Env (имеющих по два регулятора – двух параметров Dacay и Release) и на порт синхронизации несущей (далее…..). На огибающую модулятора сигнал от Gate подается дозировано, проходя через регулятор FM (этот регулятор и осуществляет регуляцию уровня FM-модуляции).

Перед выходом сигнал комбинируется в моно.

2. Трехоператорный синтез (два варианта) FM 3 ops ab

Рис. 2.1. Вид панели управления синтезатора с двумя модуляторами и несущей

Рис. 2.2. Вид панели управления синтезатора с одним модулятором и двумя несущими

Рассмотрим синтезатор на основе трех операторов. Ясно, что такой синтезатор можно построить двумя разными способами: сигнал, составленный из двух операторов модулирует третий оператор и второй случай – когда все три оператора соединены в последовательную цепь, при этом третий оператор модулируется уже составной волной, произошедшей от модуляции второго

оператора первым. Панели синтезаторов на рис. 2.1 и 2.2 ничем не отличаются, кроме названий операторов.

Структуры этих двух типов приведены на рис. 2.3 и 2.4.

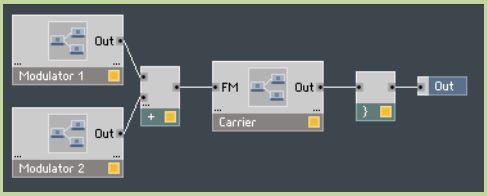


Рис. 2.3. Общая структура синтезатора с двумя модуляторами и несущей

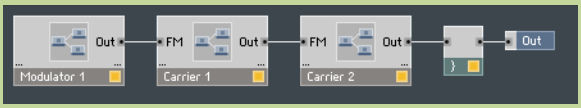


Рис. 2.4. Общая структура синтезатора с одним модулятором и двумя несущими

Нам остается только рассмотреть структуру оператора (рис. 2.5, здесь изображен модулятор).

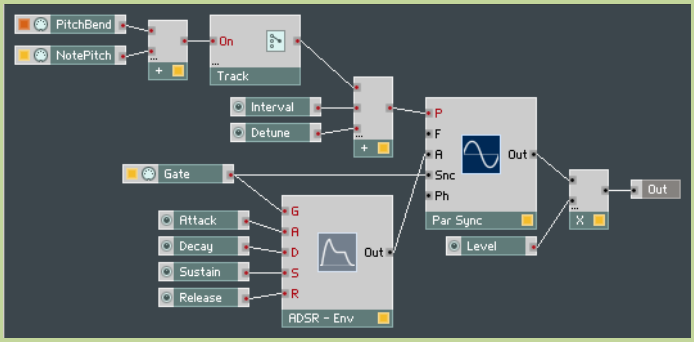


Рис. 2.5. Структура макроса оператора Modulator

Итак, мы видим огибающую ADSR-Env со средствами управления, осциллятор с возможностью линейной модуляции, переключатель Track, осуществляющий, по существу, включение и выключение сигнала оператора, регуляторы смещения Interval и Detune, а также дозатор уровня сигнала Level. На рисунке изображен оператор типа «модулятор». Оператор типа «несущая» отличается лишь дополнительным входным портом, который соединяется на порт F осциллятора, как раз для осуществления частотной модуляции.

3. Трехоператорный синтез с фильтром

Рис. 3.1. Вид панели управления синтезатора

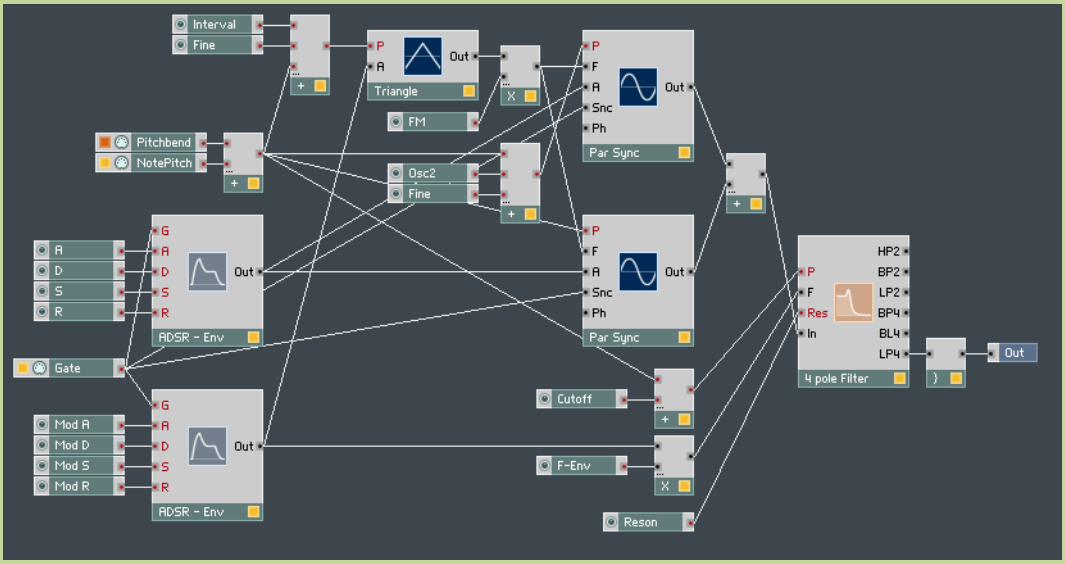


Рис. 3.2. Общая структура синтезатора

Теперь рассмотрим трехоператорный синтез с фильтрацией (панель изображена на рис. 3.1). Идея такова: два параболических осциллятора с общей огибающей модулируются осциллятором волны треугольного типа, затем суммируются и фильтруются. Причем фильтр и этот модулятор имеют отдельную общую огибающую. На схеме структуры это видно (рис. 3.2).

Смещение модулятора относительно взятой ноты (точнее относительно NotePitch+Pitchband) осуществляется регуляторами Interval и Fine, уровень сигнала модуляции регулируется значением FM. Для одной из несущих сигнал высоты ноты передается без смещения, для второй смещается на значение Osc2+Fine (не спутайте этот регулятор с одноименным у модулятора, они разные!). Фильтр смещается на значение Cutoff (что дает постоянную компоненту частоты среза фильтра в пределах одной ноты). Уровень сигнала от огибающей дозируется регулятором F-Env, а резонанс – Reson. Используется выходной порт фильтра, осуществляющий фильтрацию нижних частот 4 порядка.

Перед выводом сигнал комбинируется в моно.

4. Двухоператорный с чёперным (вращающимся) фильтром (+статический фильтр) и импульсной модуляцией

Рис. 4.1. Вид панели управления синтезатора

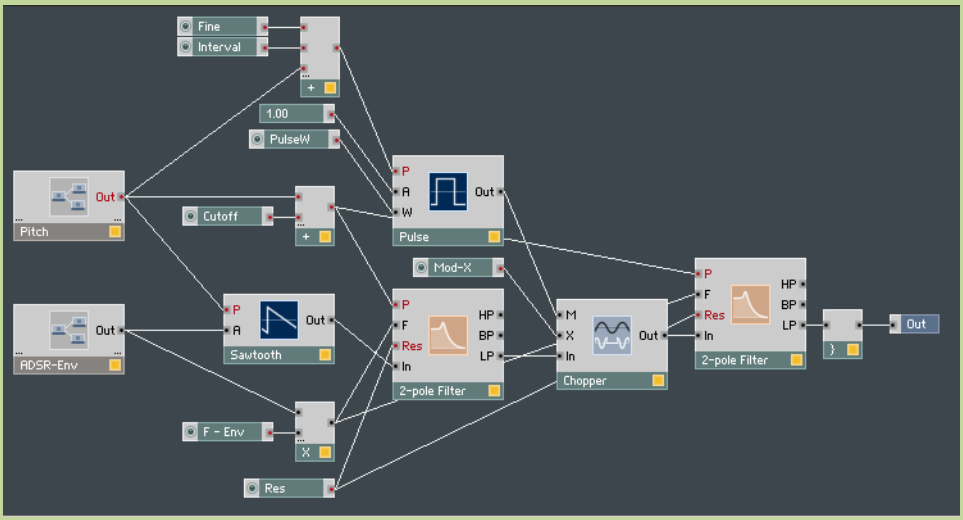


Рис. 4.2. Общая структура синтезатора

Здесь мы рассмотрим синтезатор с “вращающимся” фильтром, построенный на двух осцилляторах (непосредственной модуляции нет). Панель этого синтезатора изображена на рис. 4.1. Общая структура синтезатора (рис. 4.2) выражает такую идею: сигнал с осциллятора Sawtooth фильтруется первым фильтром, проходит на Chopper (далее…….), который, в свою очередь, модулируется сигналом импульсного осциллятора единичной амплитуды. Затем сигнал вторично фильтруется и подается через комбинатор на выходной порт синтезатора. Макрос Pitch (рис. 4.3) введен просто для удобства. Макрос огибающей стандартен (рис. 4.4).

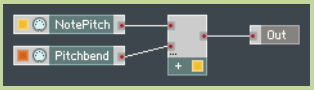


Рис. 4.3. Структура макроса Pitch

Рис. 4.4. Структура макроса ADSR-Env

Управление двумя фильтрами осуществляется одной цепью управления синхронно как по линейному контролю (с дозатором F-Env), так и по логарифмическому (со смещением Cutoff). Для импульсного осциллятора введены регуляторы подстройки (смещения) Fine и Interval.

5. SAW-модуляция трех несущих треугольных волн с фильтром (Модуляция несущей такая же как в предыдущем примере!) см.

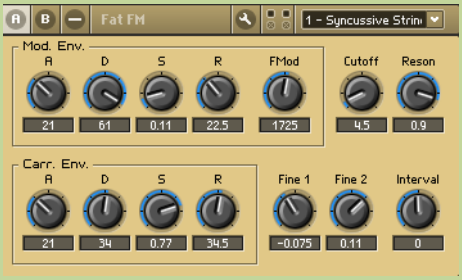


Рис. 5.1. Вид панели управления синтезатора

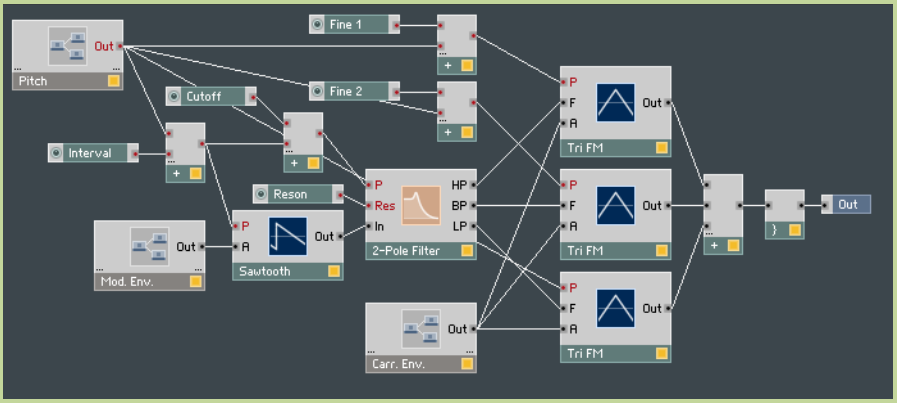


Рис. 5.2. Общая структура синтезатора

Рассмотрим модуляцию волной типа Sawtooth блока, состоящего из трех треугольных осцилляторов через фильтр.

Вид панели управления синтезатора изображен на рис 5.1. По структуре (рис. 5.2) видно, что присутствуют две огибающие. Огибающая для модулятора Mod.Env. (рис. 5.3) отличается от огибающей Carr.Env. (см. ранее рис. 4.4) всего лишь наличием дополнительного регулятора Fmod, который задает уровень сигнала от Gate (этот-то сигнал и пойдет на модуляцию через фильтр).

Итак, рассмотрим механизм управления. Сигнал NotePitch+Pitchband из макроса Pitch (см. ранее рис. 4.3) поступает на 5 модулей: на три осциллятора треугольных волн (несущие) – на один без смещения, на два других с различными смещениями Fine1 и Fine2; на модулятор Sawtooth со смещением Interval, а также на фильтр с дополнительным (к Interval) смещением Cutoff.

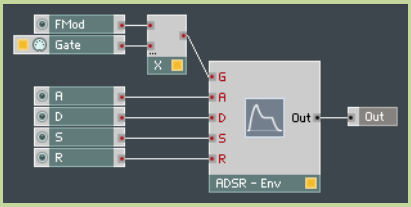


Рис. 5.3. Структура макроса Mod.Env.

Сигнал от модулятора проходит через фильтр, который выводит три сигнала (три разных типа фильтрации), и каждый из этих сигналов модулирует только одну треугольную волну. Затем сигналы складываются.

Перед выводом сигнал комбинируется в моно.

Органы (аддитивный синтез) на основе сложения волн

1. 8-волновый орган

Рис. 1.1. Вид панели управления синтезатора

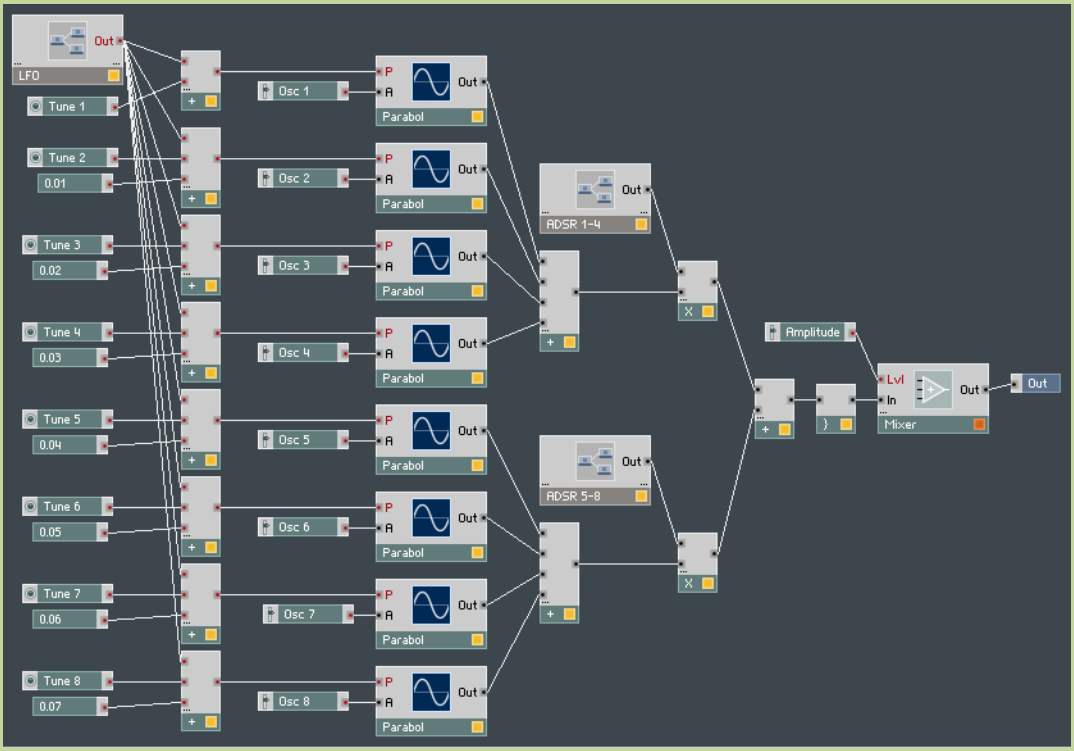


Рис. 1.2. Общая структура синтезатора

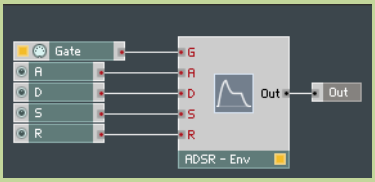


Рис. 1.4. Структура макросов ADSR 1-4 и ADSR 5-8

Рассмотрим синтезатор на основе восьми параболических осцилляторов, реализующий аддитивный синтез, то есть звук формируется большим числом осцилляторов, сигналы которых складываются. Звуки синтезаторов такого типа хорошо имитируют органы.

На рисунке 1.1 изображена панель управления синтезатором, а на рис. 1.2 его структура. Читателям, которые изучили предыдущие примеры, будет предельно ясно и построение этого синтезатора: 8 осцилляторов разбиты на два блока, каждый блок управляется отдельной огибающей (см. рис. 1.4). Каждый осциллятор имеет точную подстройку высоты тона и амплитуды сигнала.

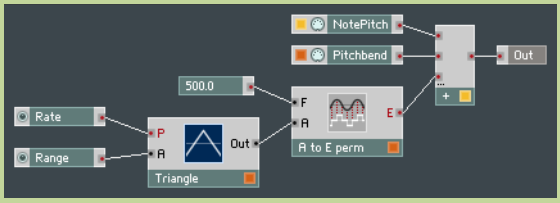


Рис. 1.3. Структура макроса LFO

Отдельно рассмотрим макрос LFO (рис. 1.3). Он построен не на модуле LFO, а на простом осцилляторе треугольной волны. Это сделано для того, чтобы получить белее высокие частоты модуляции (далее….).

Перед выводом сигнал комбинируется и микшируется его уровень (так как микшер расположен за комбинатором, то выстраивается не звук каждой отдельной ноты, а всего звукового потока в целом).

2. Импульсный 6-волновый орган

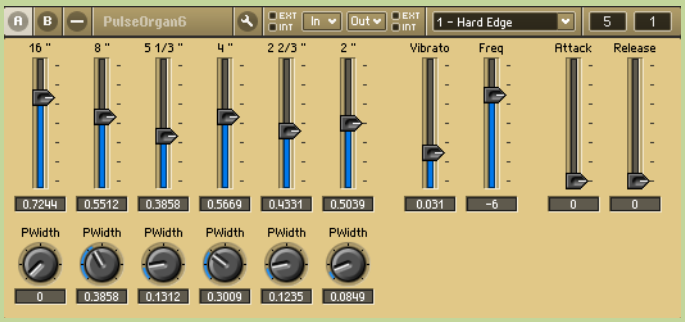


Рис. 2.1. Вид панели управления синтезатора

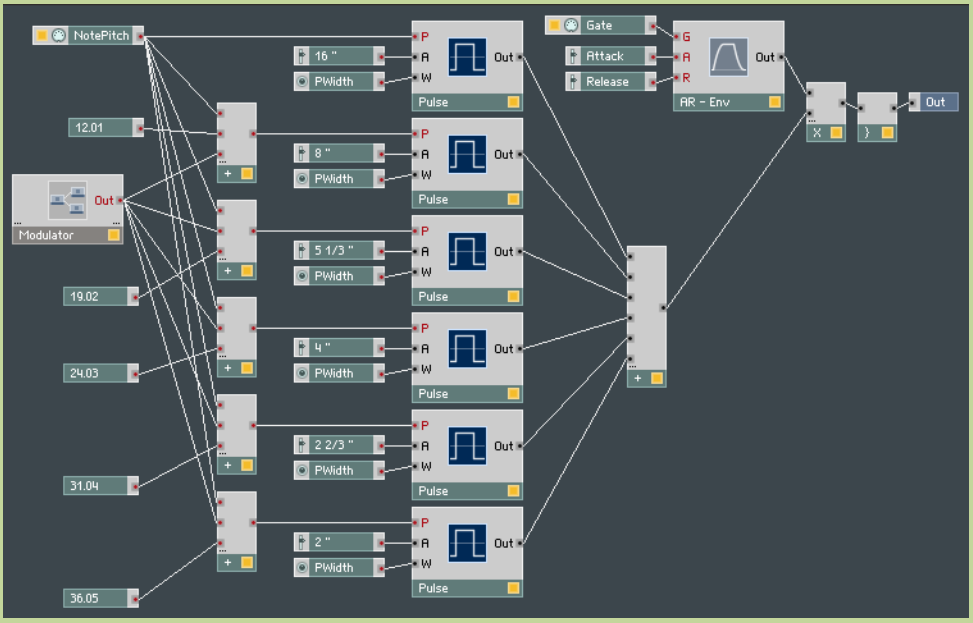


Рис. 2.2. Общая структура синтезатора

Теперь рассмотрим аддитивный синтезатор на основе импульсных осцилляторов. Его панель управления изображена на рис. 2.1. Структура его (рис. 2.2) весьма схожа со структурой предыдущего примера, только здесь осцилляторов всего 6, и огибающая только одна, и к тому же типа AR-Env.

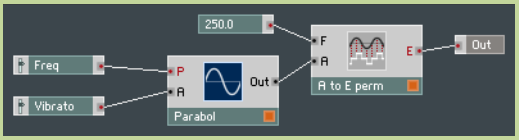


Рис. 2.3. Структура макроса Modulator

Константы смещений высоты тона ноты подобраны так, чтобы звучали гармонично, и дополнительной настройки сдвига тона с панели синтезатора не предусмотрено. Макрос Modulator (рис. 2.3) построен на основе параболического осциллятора (далее…….).

3. Shape org (не аддитивный)

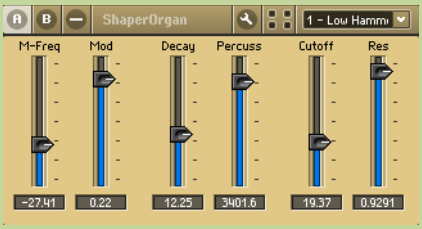


Рис. 3.1. Вид панели управления синтезатора

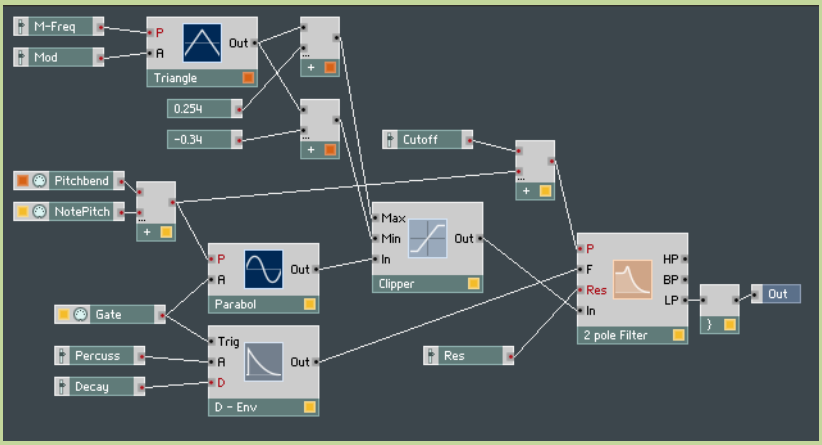


Рис. 3.2. Общая структура синтезатора

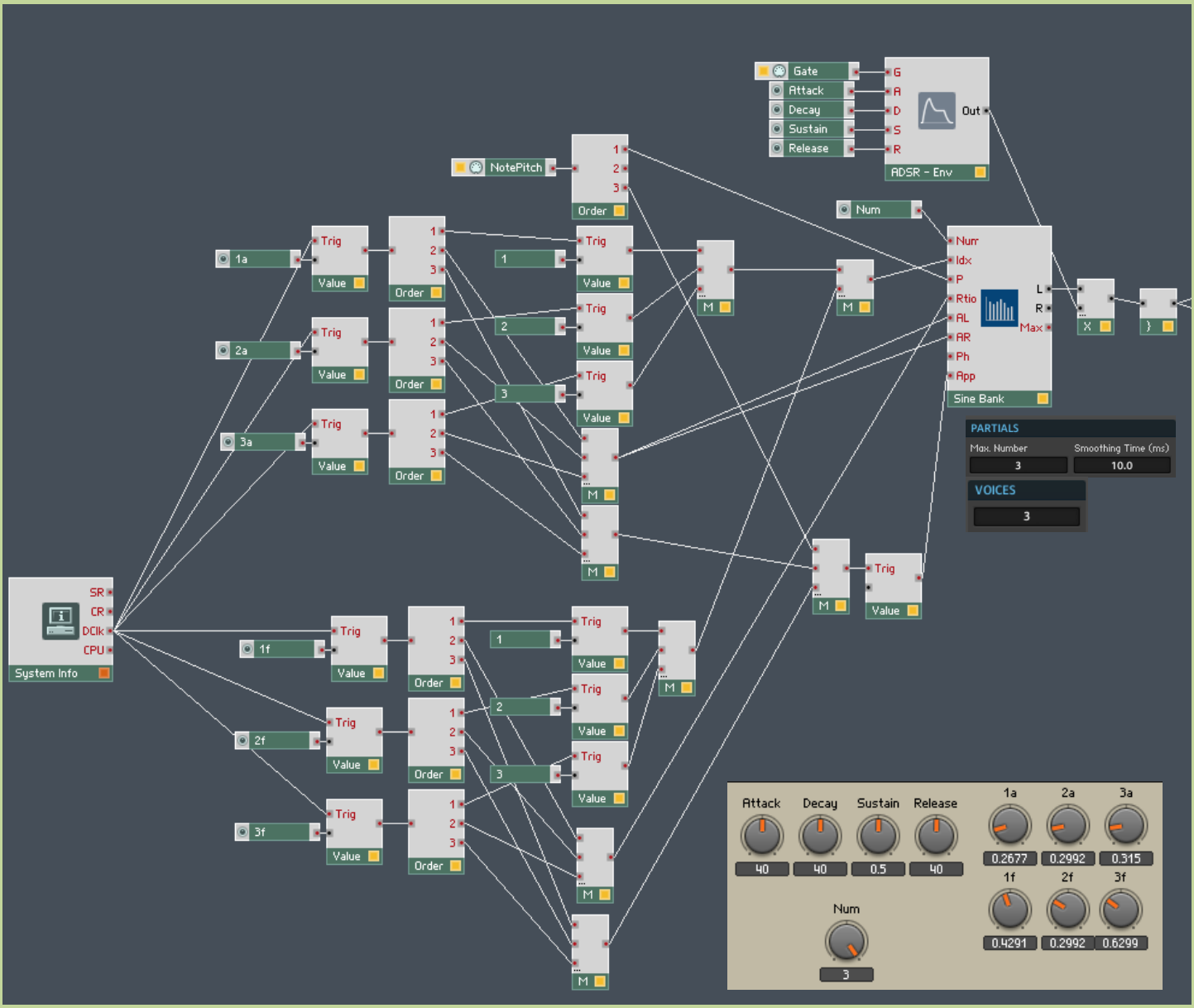
Здесь рассмотрим еще один вариант синтезатора, генерирующего органоподобные звуки, но без использования аддитивного синтеза. Идея состоит в том, чтобы создавать звуковую волну не суммированием, а при использовании клиппирования (ограничения сигнала). Это клиппирование сигнала также модулируется осциллятором. Рассмотрим синтезатор более подробно. На панели (рис. 3.1) присутствует немного управляющих элементов.

Структура синтезатора изображена на рис. 3.2. Как видно, она построена без макросов. В структуре используется основной параболический осциллятор, осциллятор волны треугольного типа с ограничителем Clipper, а также фильтр в связке с простой огибающей типа D-Env.

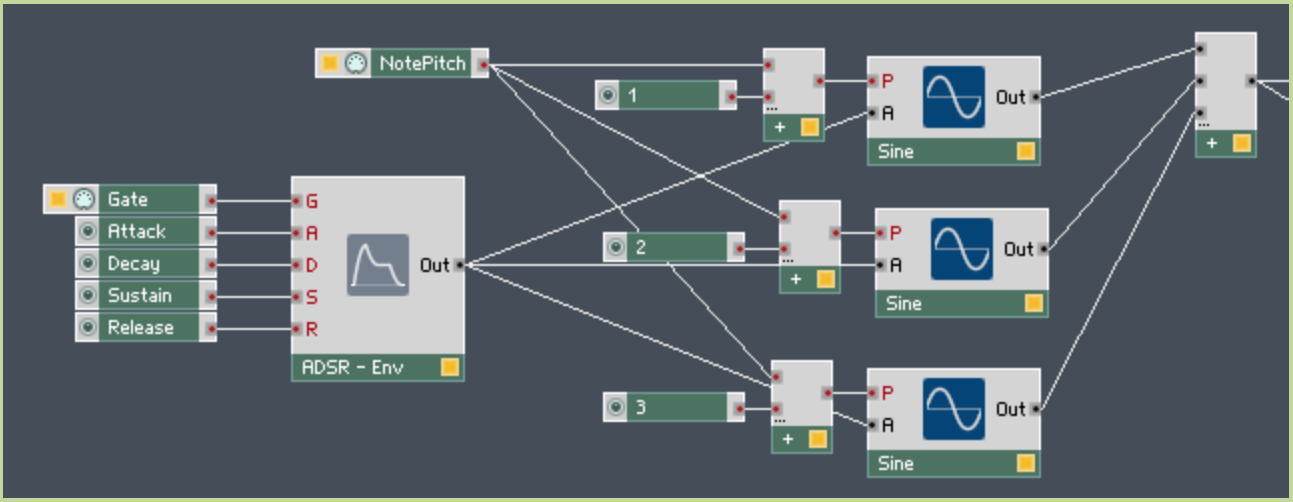
Из схемы видно, что диапазон клиппирования в каждый момент времени меняется посредством осциллятора треугольной волны (причем эта волна еще смещена двумя константами, а регуляторы этой волны не зависят от нажатых MIDI-клавиш). Сигнал NotePitch+Pitchband посылается только на осциллятор основной волны и на фильтр. Для фильтра также применено смещение высоты тона на Cutoff по линии логарифмического управления. Линейный контроль фильтра идет непосредственно с огибающей D-Env.

Как обычно, сигнал перед выводом в выходной порт синтезатора комбинируется.

Sine Bank



Sine Bank осциллятор отличается от других осцилляторов тем, что в нём можно задать произвольное отношение между гармониками-partials. При этом звук будет необычным, так называемый дисгармоничный, в котором довольно трудно определить тонику. Хотя такое можно сделать суммой Sine осциляторов.

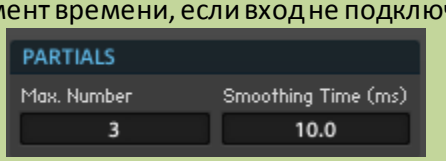


Knobs 1,2,3 – рациональная подстройка – дробный Stepsize. В отличии от Sine Bank здесь громкость для каждой гармоники одинаковая.

Sine Bank более удобен при построение сложного звука состоящего например из 128 гармоник, если бы мы использовали просто модули Sine то их довольно долго надо было бы вставлять – 128 штук, а в Sine Bank гармоники это idx индексы, с помощью Iteration и формул можно получить простую схему со сложным звуком.

Sine Bank

Num – сколько гармоник-partials звучит в данный момент времени, если вход не подключён , то будут работать все гармоники-partials согласно настройке Sine Bank



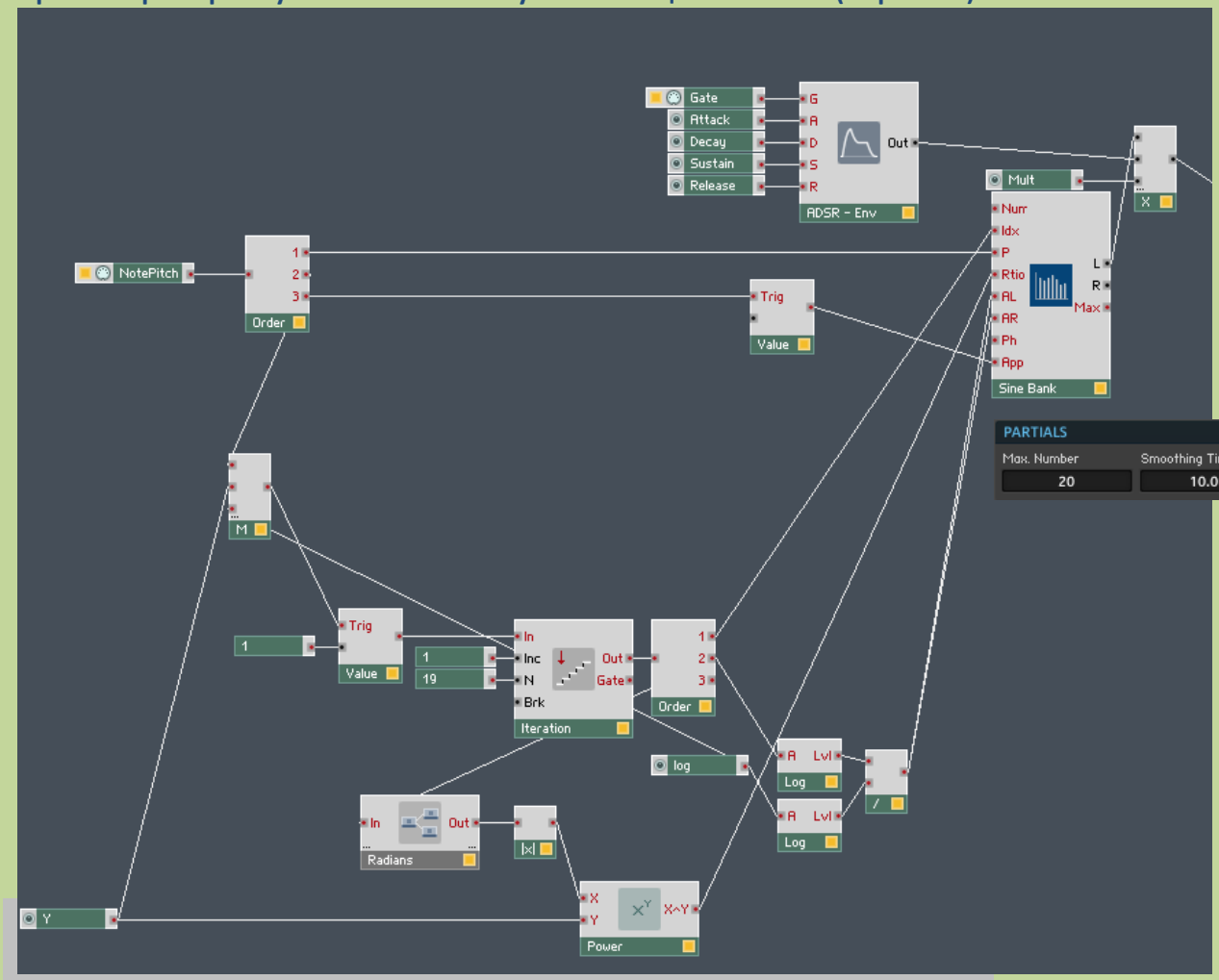
Idx – номер гармоники, начинается с 1.

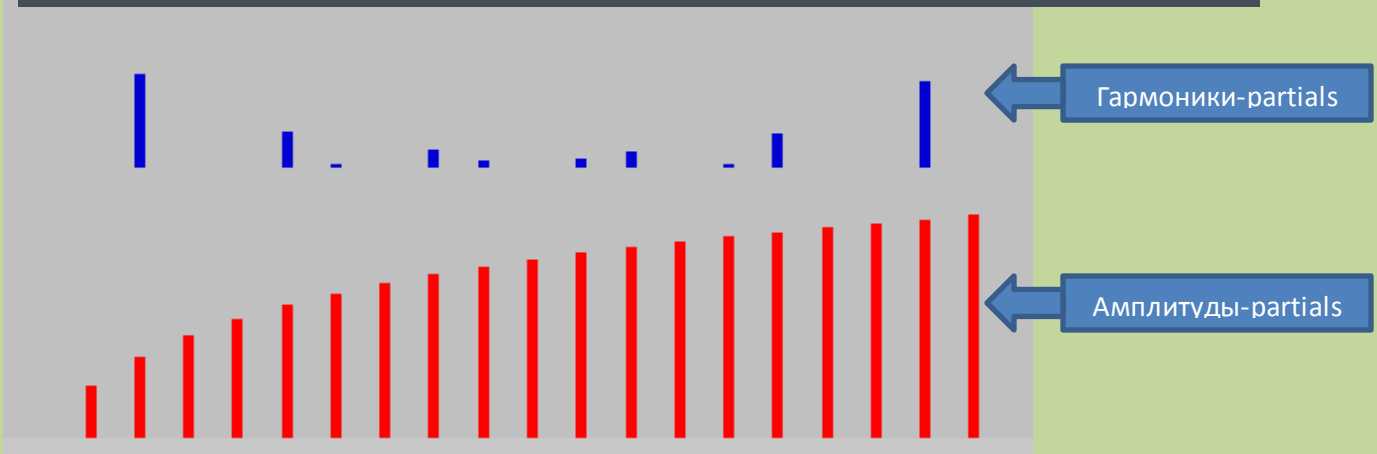
Rtio – отношение между гармониками. Если гармоника=1, то это гармоника=NotePitch.Если вторая гармоника в двое больше чем другая, то отношение по высоте это октава.

Al,AR – амплитуда-громкость гармоник

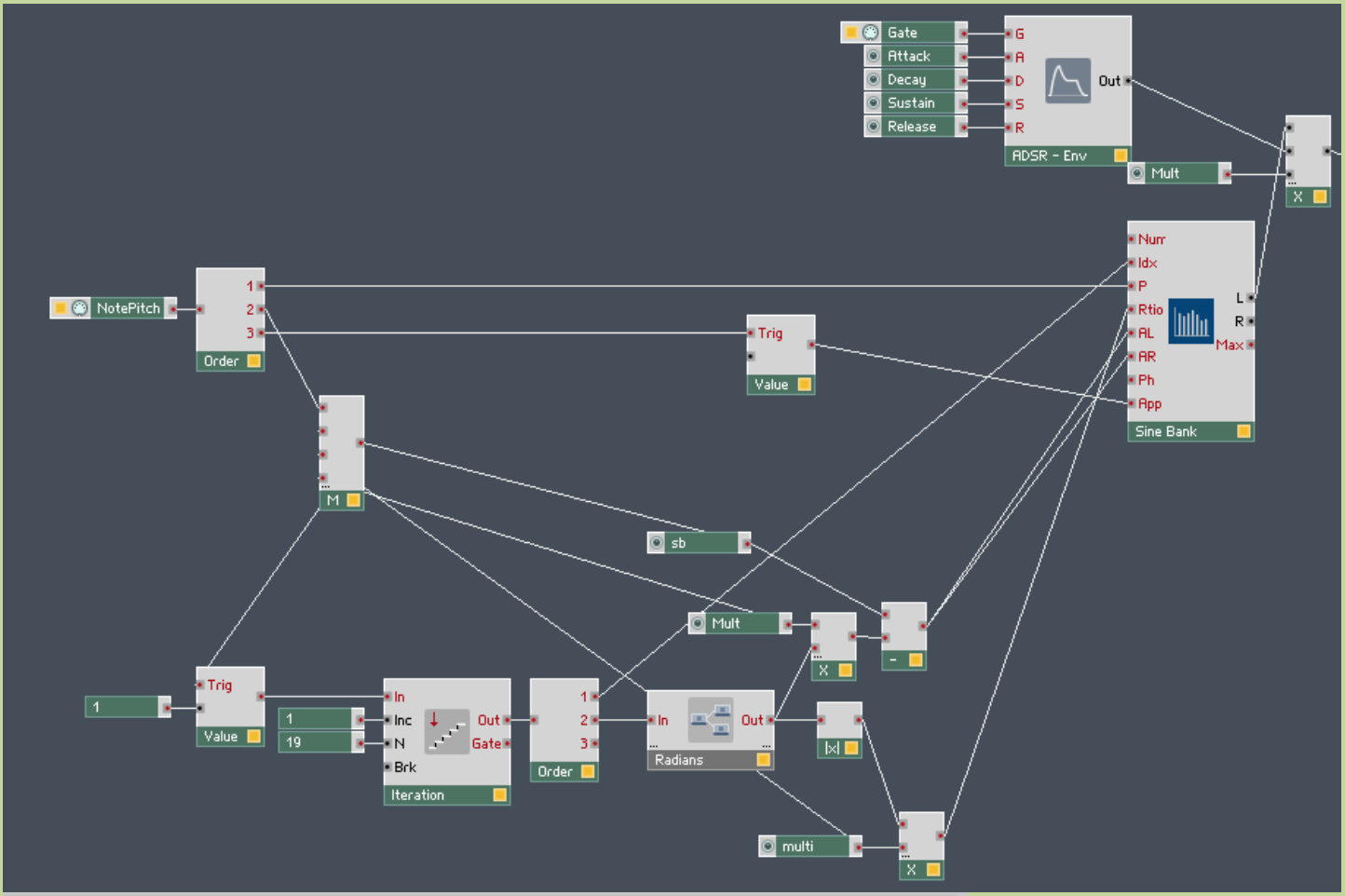
App – сокращённое от Applied – применить. Если вход **Ph-Phase** не применяется в схеме, то на вход App посылается ноль. В схеме этот вход App должен измениться самым последним, то есть сначала выбрались idx, затем AL-AR и Rtio, и только потом применить App.

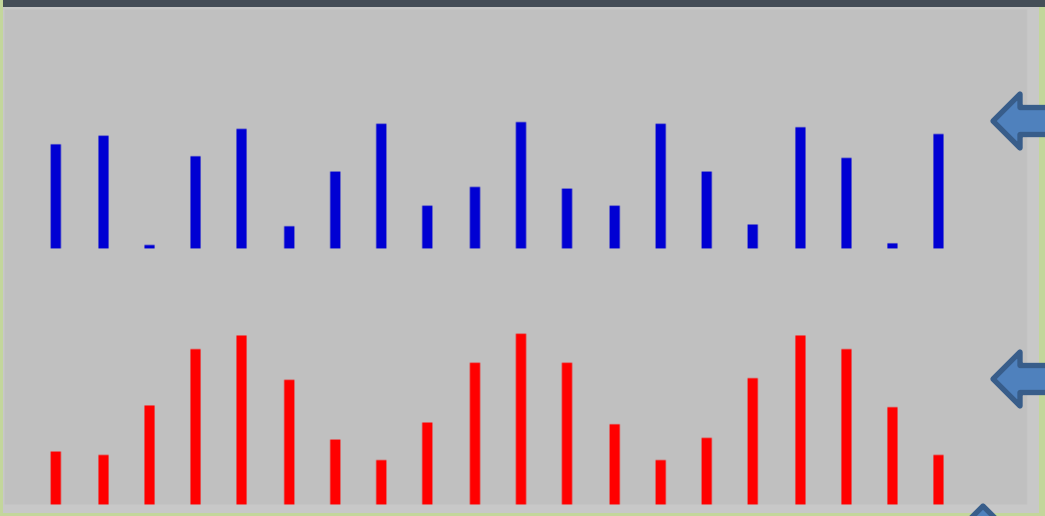
Простой пример получения сложного звука с помощью Sine Bank (20 partials)





громкость для гармоник логарифмическая

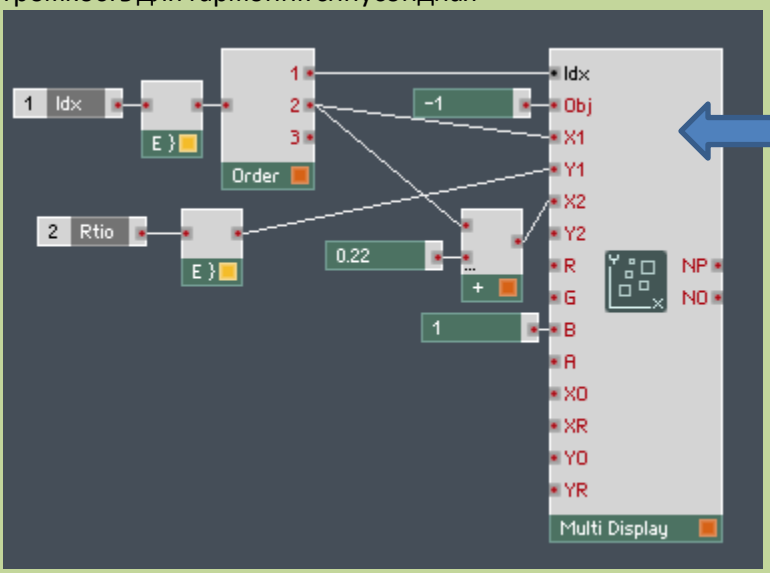




Гармоники-partials

Амплитуды-partials

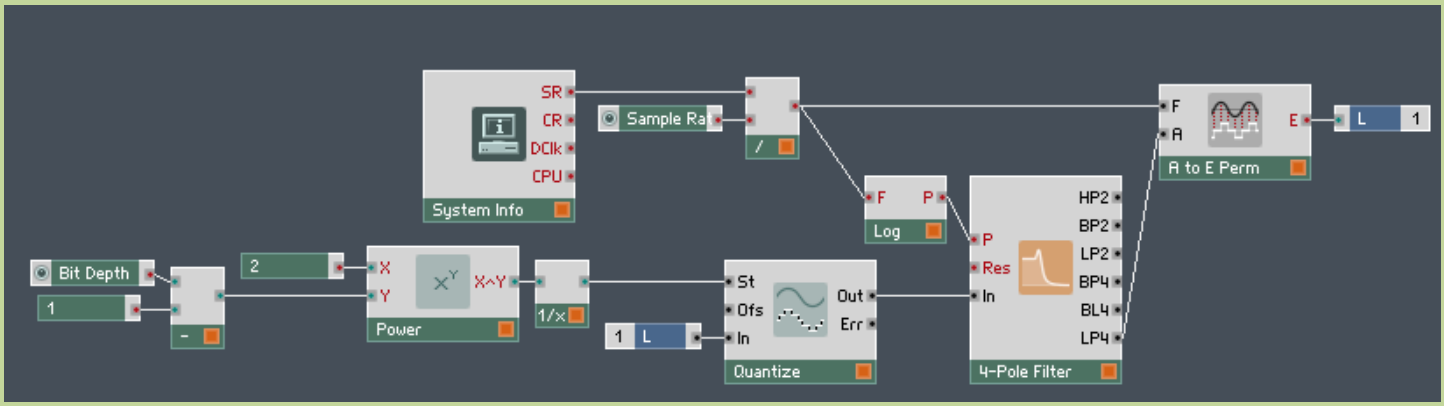
громкость для гармоник синусойдная



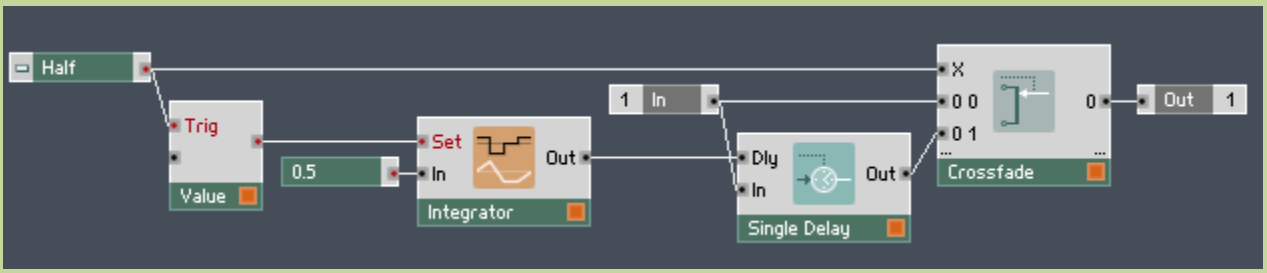
Макрос для отображения Rtio(синий цвет)

В макросе вход Idx подключается к выходу Order (1), а Rtio к выходу который посылает Rtio на SineBank.

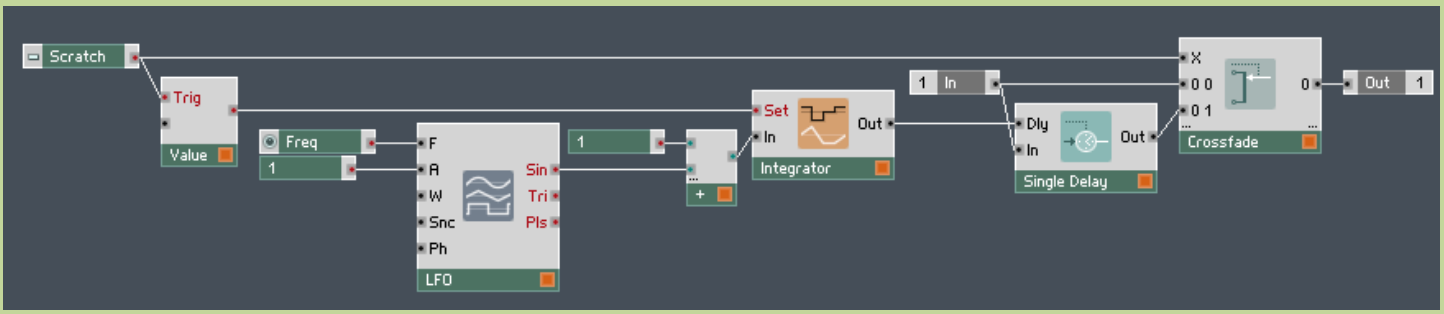
Bit Crusher



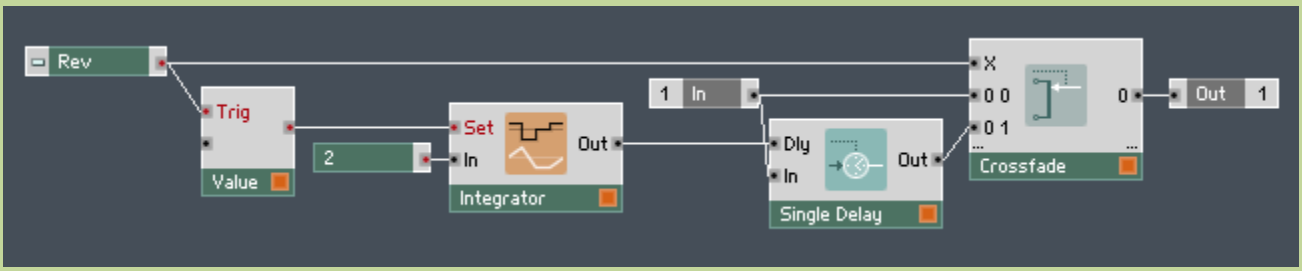
Half Speed



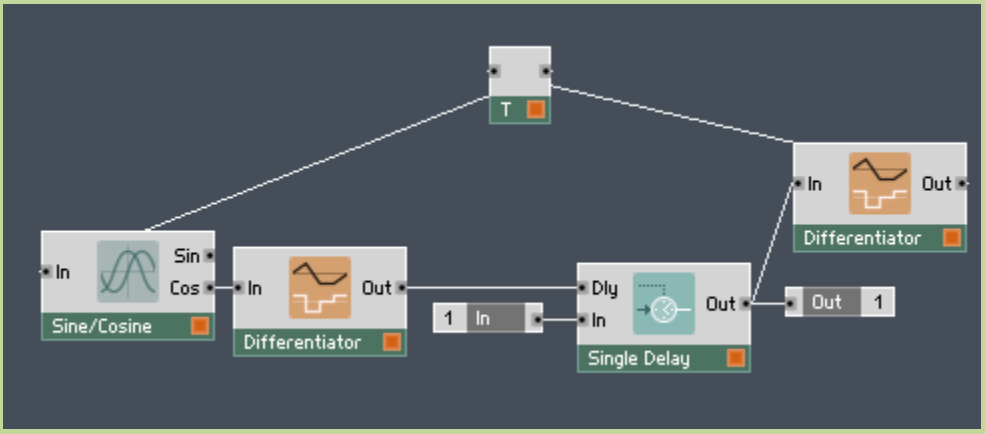
Scratch



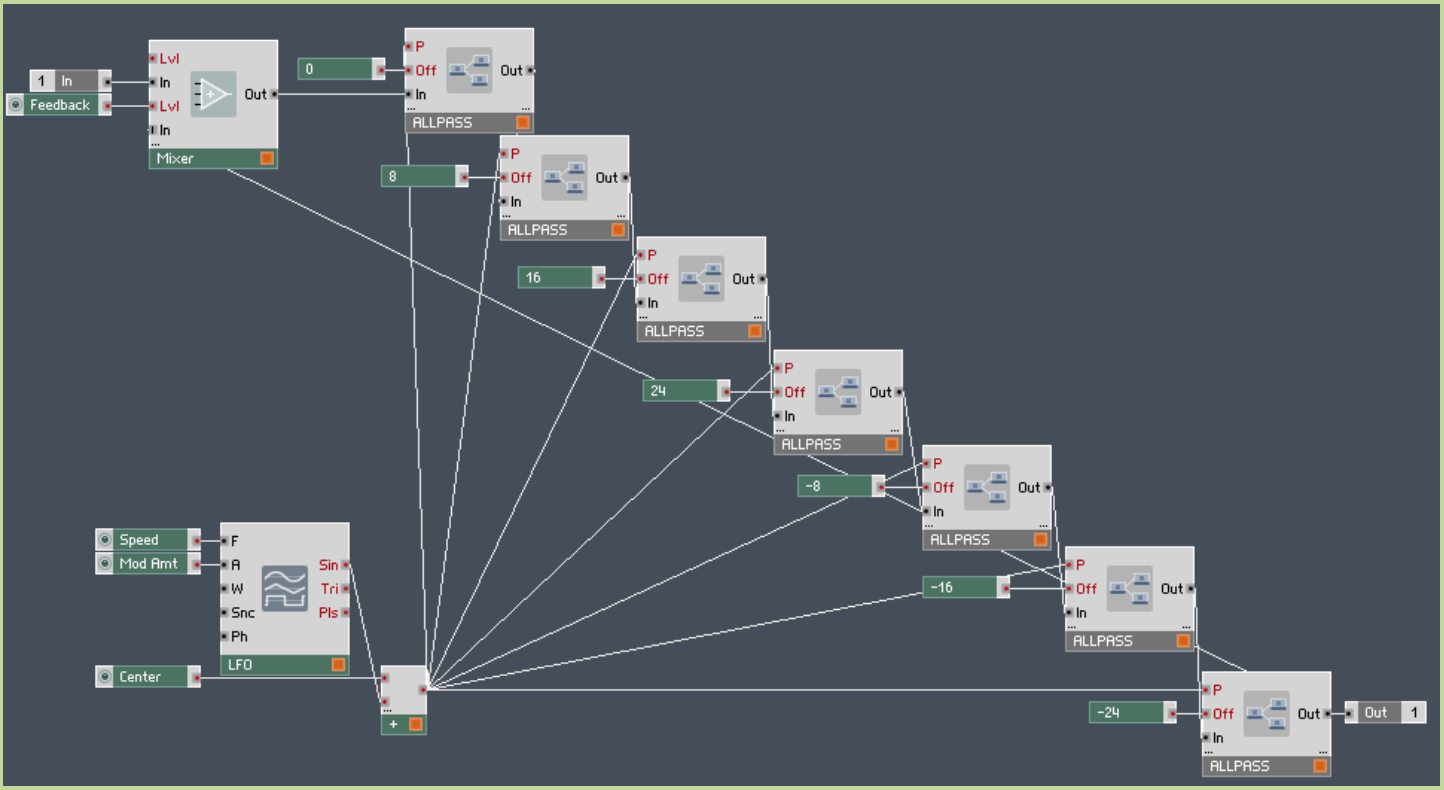
Reverse

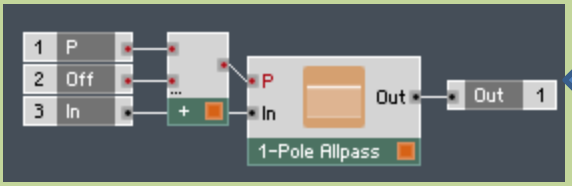


Noise



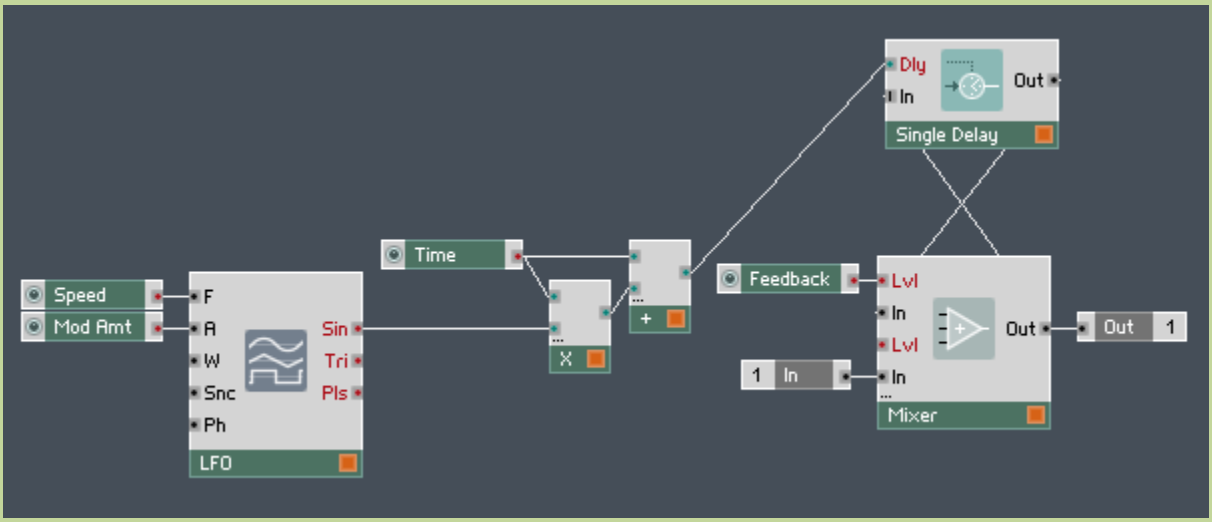
Phaser



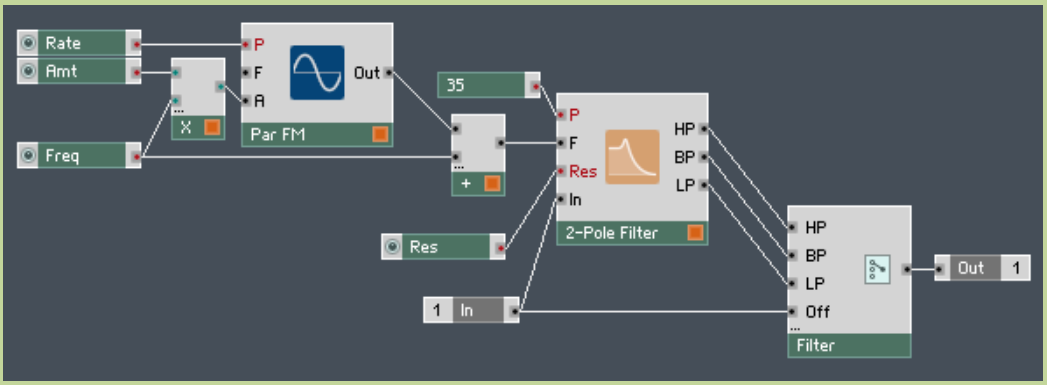


ALLPASS

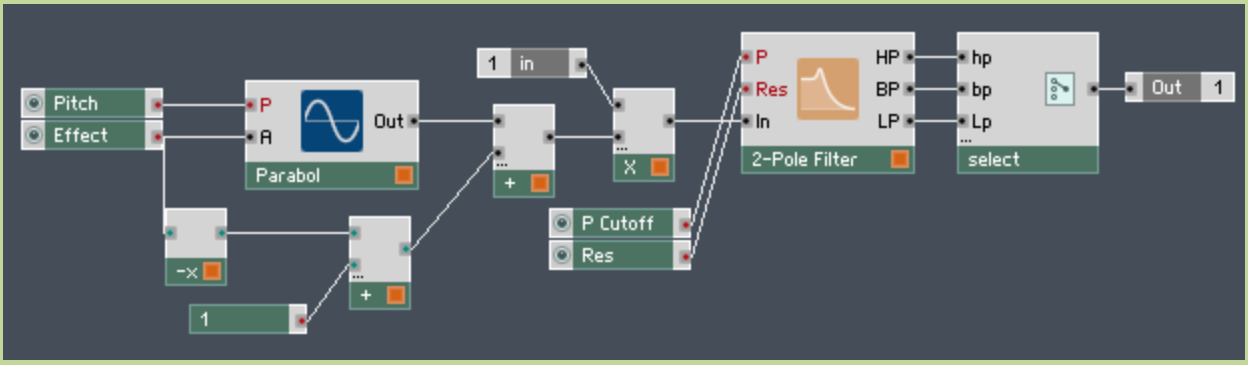
Flanger



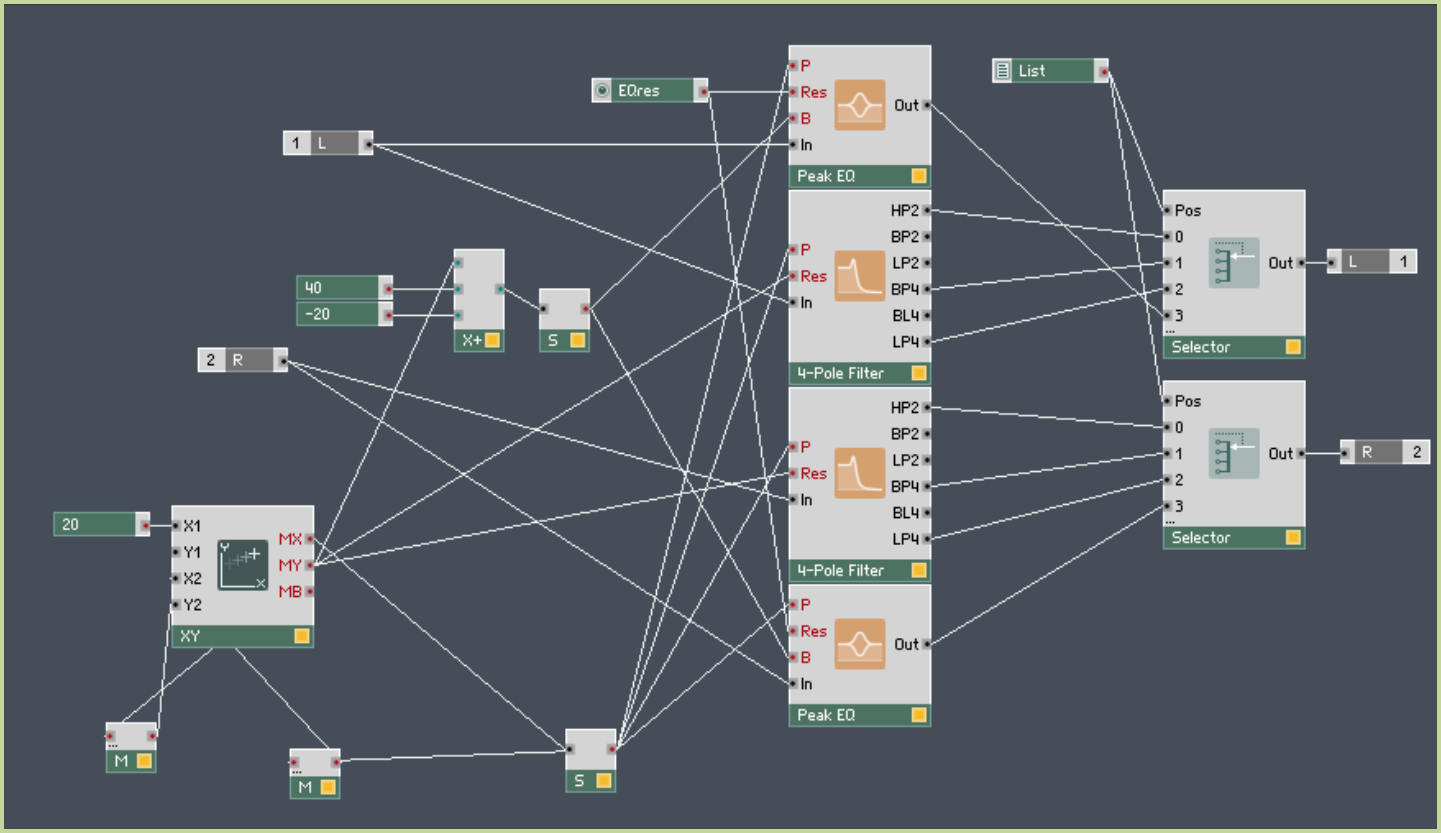
Filter



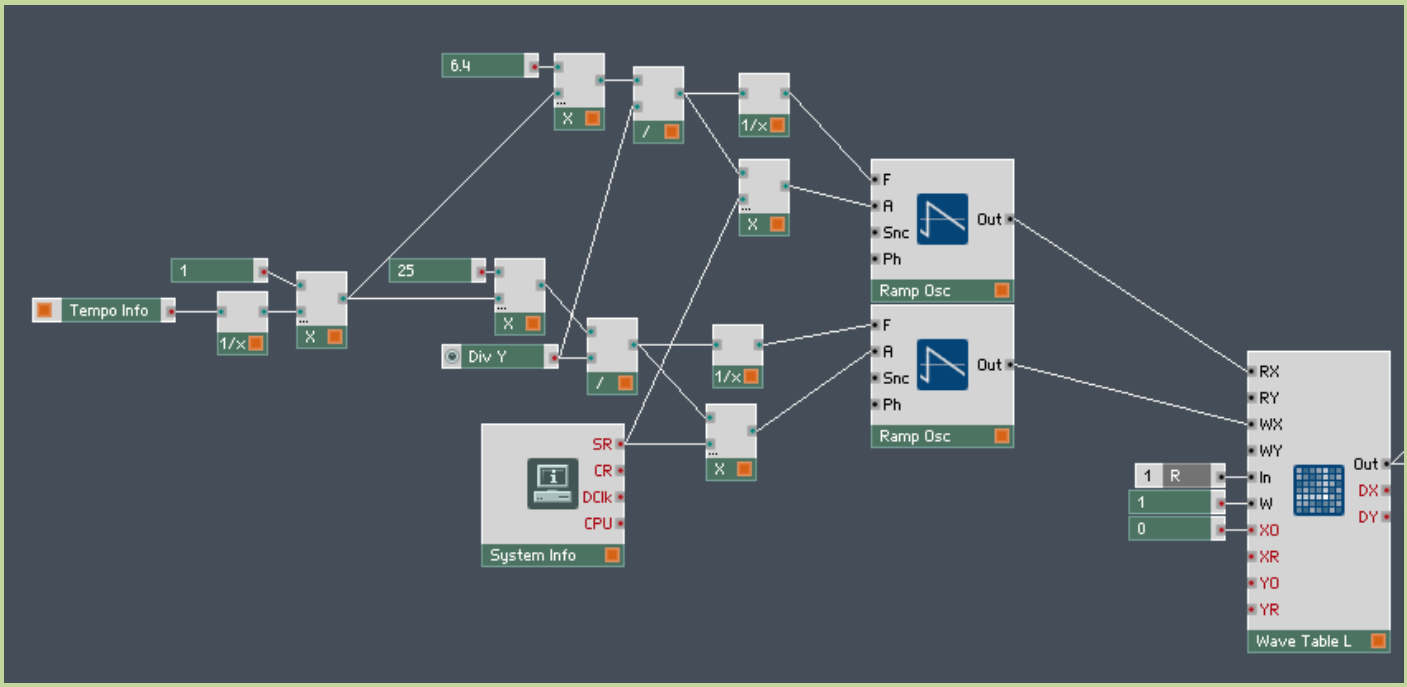
Ringmod



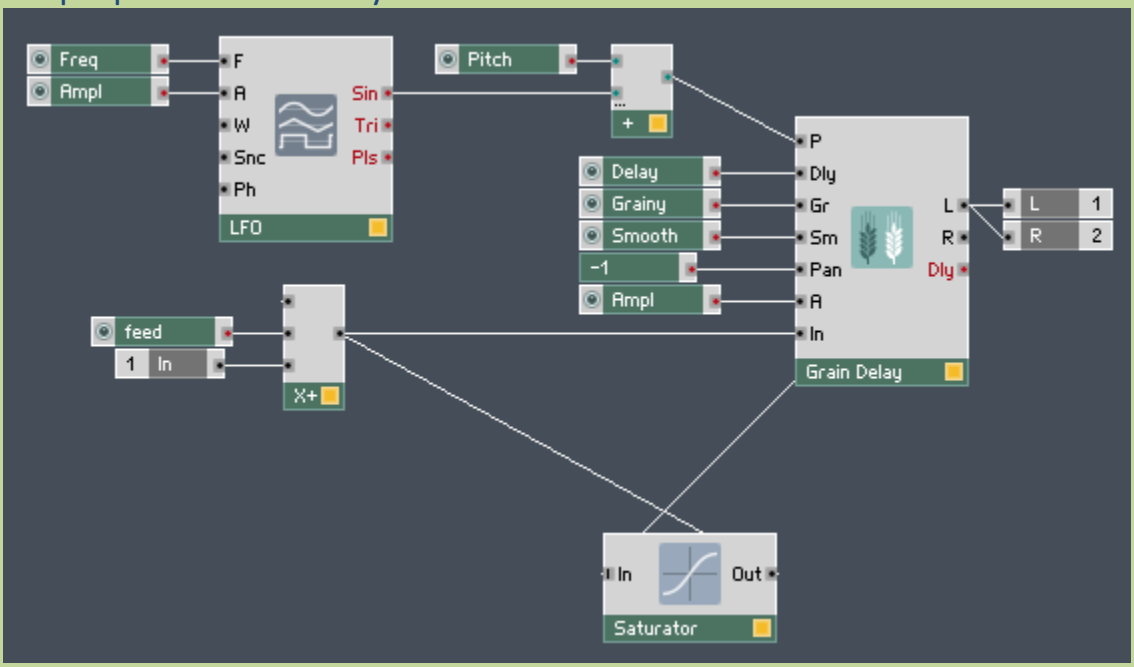
Filter



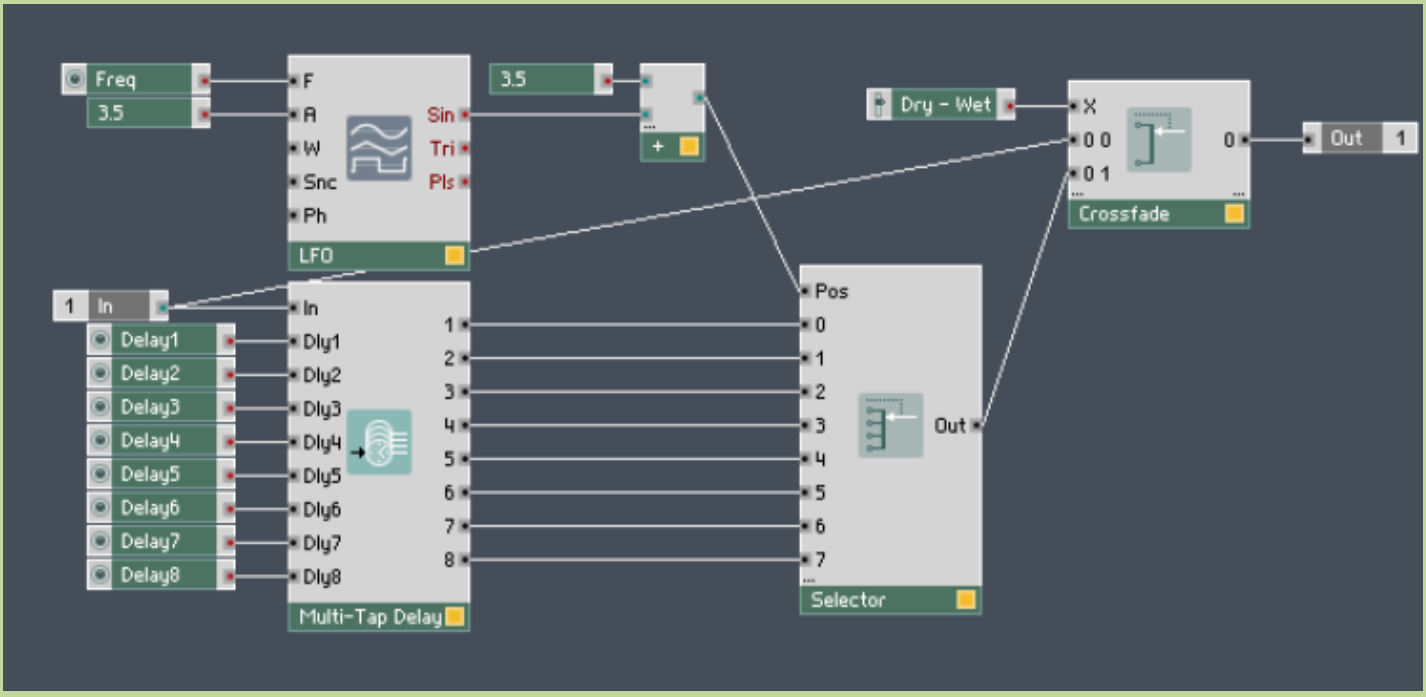
Buffers



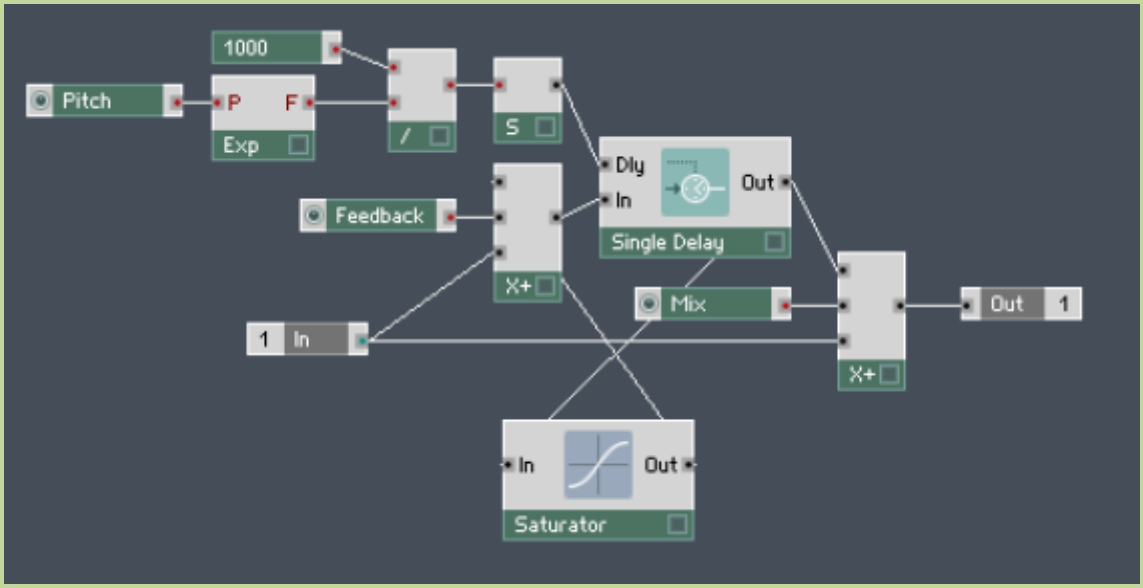
Simple pitch shifter delay effect



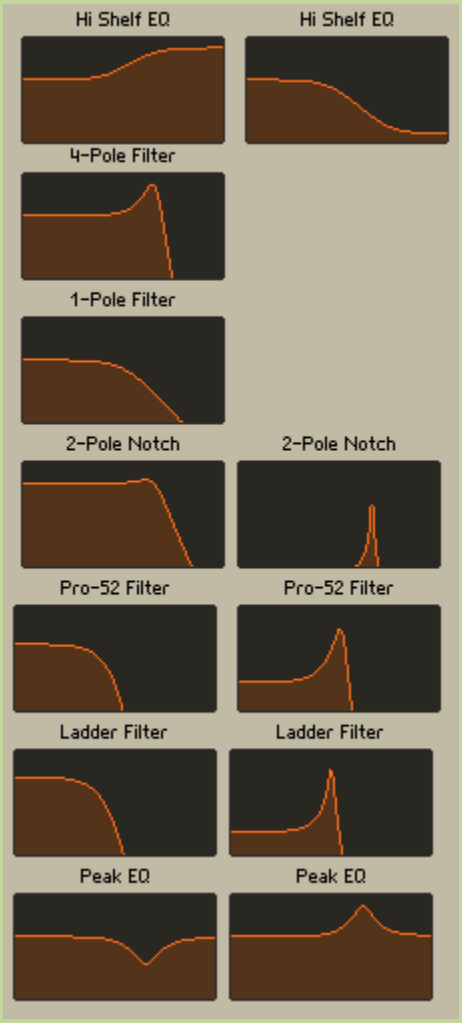
Scanning delay effect



Comb Filter



Фильтры

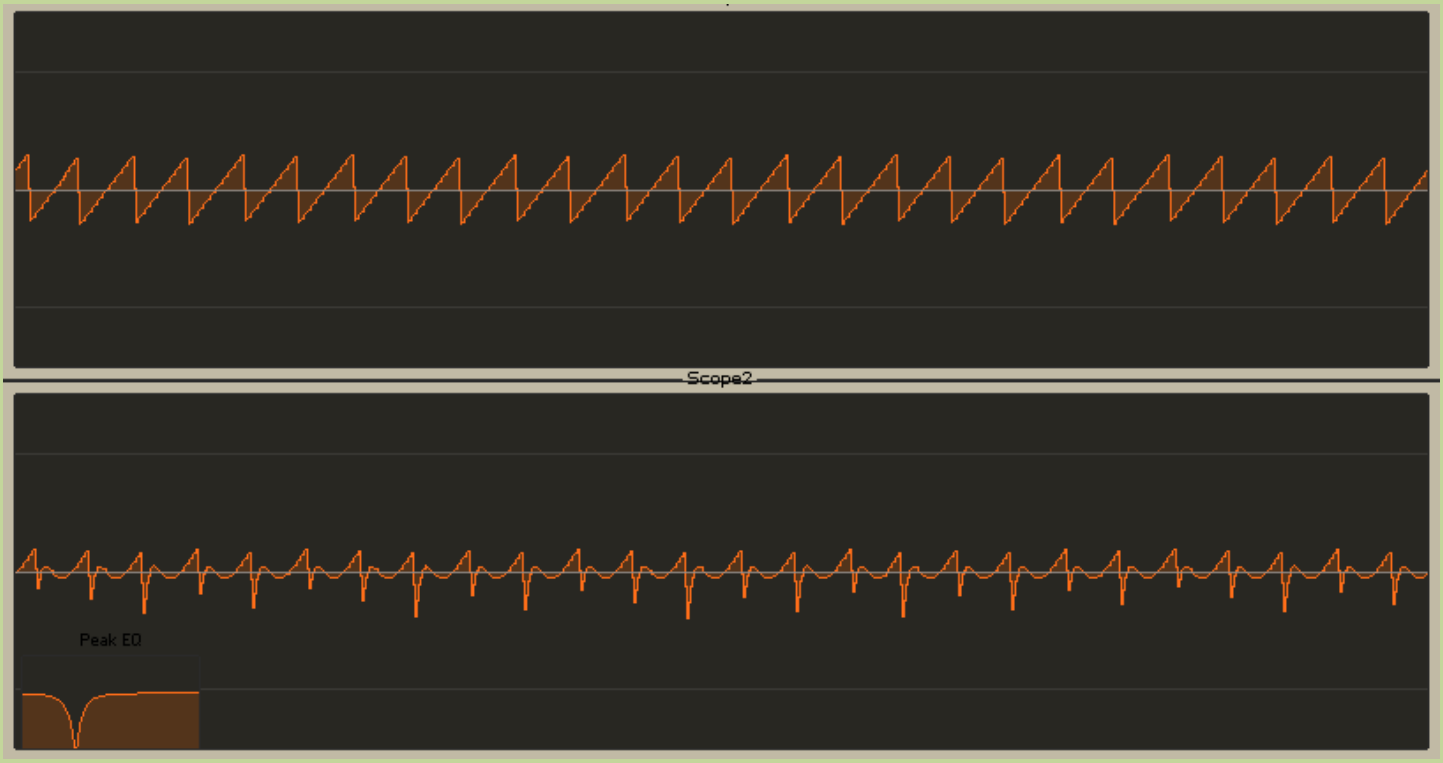


Основные параметры :

Res - резонанс

B – boost - усиление

P – частота среза



Наверху простой чистый сигнал Sawtooth , внизу этот же сигнал после фильтра Peak EQ