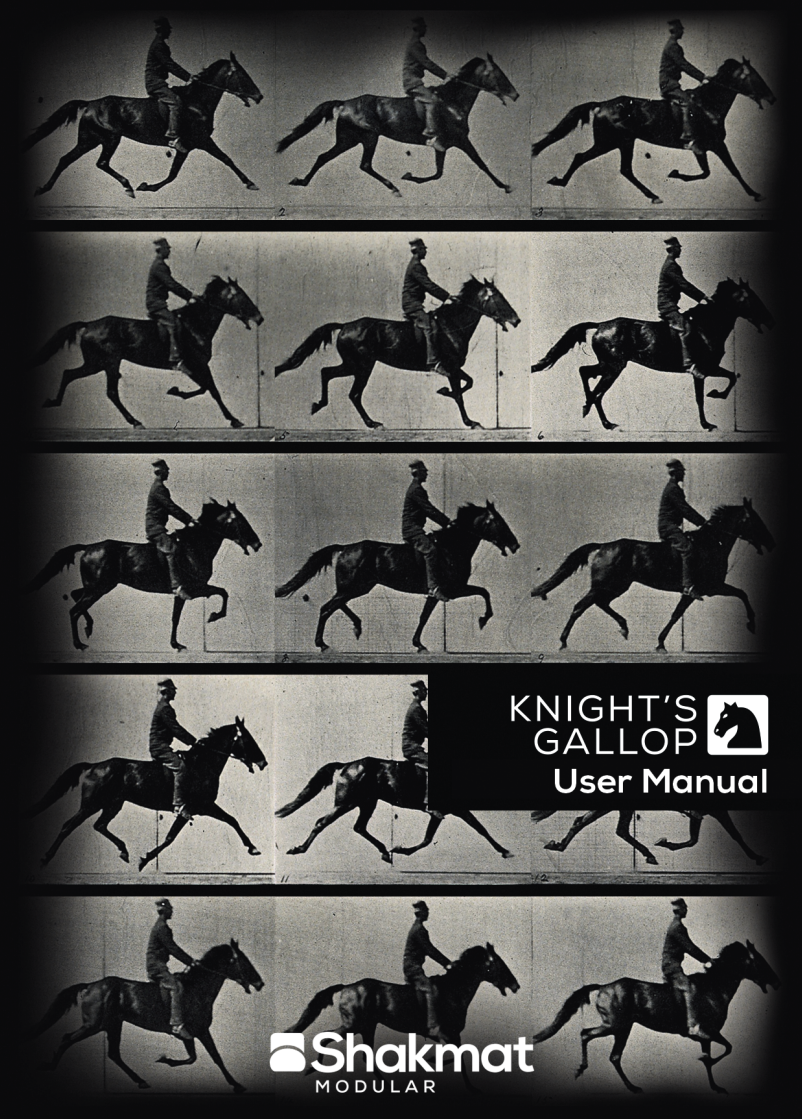


KNIGHT'S GALLOP

User Manual

Shakmat MODULAR

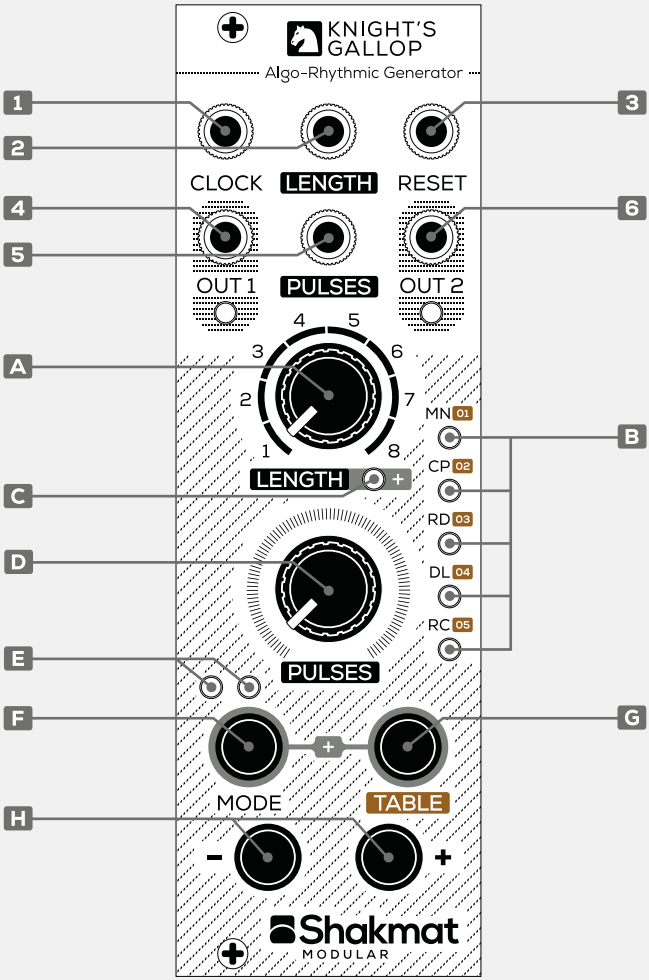
Introduction

Two outputs and five different rhythmic flavors multiply by a load of "algo-rhythmic" processing to result in a practically infinite number of poly-rhythms. That's the fundamental formula that produces the Knight's Gallop principle: a dual trigger generator providing sequences according to tables and modes.

The source tables contain patterns such as the Euclidean polyrhythms, and the modes allow the user manipulate and mangle those patterns according to different algorithms. As the module has two outputs, it is a very powerful instrument to create diversified and instantaneous polyrhythms.

1. Clock input
2. Length CV input
3. Reset input
4. Trigger Out 1
5. Pulses CV input
6. Trigger Out 2

A. Length potentiometer
B. Modes & Tables LEDs
C. Length+ LED
D. Pulses potentiometer
E. Sub-modes LEDs
F. Mode button
G. Table button
H. +/- Shift buttons


KNIGHT'S GALLOP
Algo-Rhythmic Generator
1
2
3
CLOCK
LENGTH
RESET
4
5
6
OUT 1
PULSES
OUT 2
A
4
5
3
6
2
7
1
8
MN 01
B
LENGTH
+
CP 02
C
RD 03
D
DL 04
RC 05
E
PULSES
F
G
+
MODE
TABLE
H
–
+
Shakmat
MODULAR

Basics

First of all, the module needs to be fed by a clock signal via its **Clock input 1.** The sequence generated by **Out 1 4** is determined by four parameters : **Table type B, Length A, Pulses D & Shift H .**

The **Length potentiometer A** adjusts the sequence length from 1 to 8 steps. By pressing the **Mode F & Table buttons G** together, the length range can be set from 1-8 to 9-16 steps, while in 9-16 range, the **Length+ LED C** is on.

The **Pulses potentiometer D** sets the number of hits in the sequence. Turned fully counter clockwise it mutes the outputs and as the potentiometer is turned clockwise, the number of hits distributed across the sequence increases.

Length & pulses parameters also have dedicated **CV inputs 2 5.**

The shift parameter can be adjusted with the **-/+ Shift buttons H.** Press **+** once to shift the sequence one step forward and **–** to shift it one step backward.

The module also has a **Reset input 3,** allowing to restart the sequence at its first step.

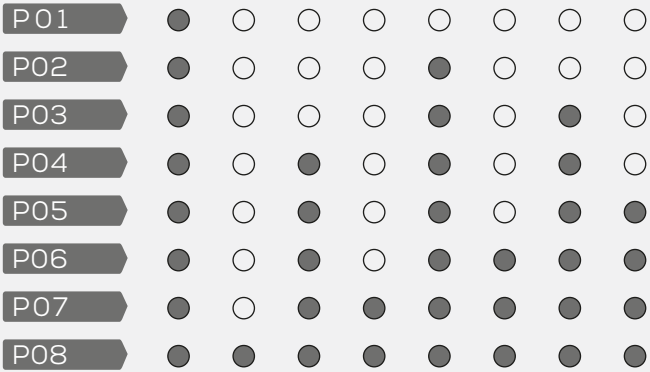
Tables

The module contains 5 tables with different feelings. When switched on, it starts on the third table, the Revised Euclidean one.

To know which set of table is used, press the **Table button** G **.** The **Mode & Table LED column** B indicates the selected table number by blinking. To navigate through the tables keep the **Table button** pressed and use the **+** & **- buttons** H **.**

01. Divider Sequences

This table is constituted of sequences made of different clock dividers tha lead to very straight, military patterns. For example, here are the 8 steps sequences :

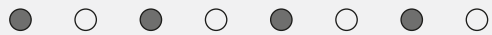


02. Classic Euclidean

The table provide the classic Euclidean feeling. The algorithm creating the sequences places pulses accross the sequence with the most constant distance between each pulse.

03. Revised Euclidean

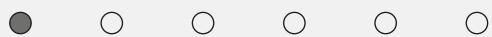
We found the classic Euclidean sequences groovy but not enough. A lot of those sequences got a syncopated feeling which is great for bouncy / tribal rythms. But some of them kill the groove as the 8 steps long one with 4 pulses in it :

E.G. 

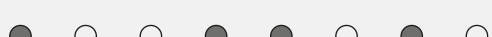
Another problem of the euclidean patterns is, for example, that 12 steps long sequence with 2 pulses in it :

E.G. 

is actually the repetition of two 6 steps long sequence with 1 pulse in it :

E.G. 

So we edited the tables, hit by hit, to avoid those little dead spots. For example, the L= 8 & P = 4 sequence is now :

E.G. 

and the L = 12 & P = 2 is now :

E.G. 

04. Anti Euclidean

Euclidean patterns lead to almost identical distances between pulses. Anti Euclidean patterns are the opposite : made to maximize the distance difference between pulses. Unlikely Euclidean patterns, they are more unpreddictable and not unifomely distributed across the sequence.

05. Split Sequences

The algorhythm creating this table split the sequence in two equally long parts. While the Pulse value is increasing, the alogorithm fills the first one according to the classic euclidean feeling until the first part is fully filled. Then, it fills up the second part with a different feeling (reversed and inverted).

Modes

As output one, the **second output** 4 is also generating patterns according to the table type and the Length, Pulse and Shift values but those patterns are modified by algorithms determined by the selected mode.

The **Mode LEDs** B show the current mode. To navigate through the modes press the **Mode** F and **+/- buttons** H**.**

Each mode contains several **Sub-modes,** to navigate through them, press the **Mode button** F.

A. Main Mode (MN)

The Main mode contain four utility modes:

SM 01 **Reset** **LEDs (E)** ○ ○

Out 2 gives a pulse every first step of the sequence.

SM 02 **No Shift** **LEDs (E)** ○ ☼

Out 2 is the same as Out 1 but not affected by the shift parameter.

SM 03 **Invert** **LEDs (E)** ☼ ○

At each step, if Out 1 is delivering a pulse, Out 2 won't . If out 1 is not delivering a pulse, out 2 will provide one.

SM 04 **Backward** **LEDs (E)** ☼ ☼

Out 2 is playing the same pattern as Out 1 but read backward.

B. Compute (CP)

As Out 1 is generating a L long sequece with P pulses in it, Out2 will provide another sequence withdrawed from the same table but with different lengths and pulses density.

Out 2 is reading the sequence with half the pulse and the same length as the one played by Out 1.
e.g. : Out 1 : P=4 & L=11 › Out 2 : P=2 & L = 11

SM 01 **L & P/2** **LEDs (E)** ○ ○

Same principle with different math.
e.g. : Out 1 : P=4 & L=13 › Out 2 : P=2 & L = 7

SM 02 **L/2 & P/2** **LEDs (E)** ○ ☼

Same principle with different math.
e.g. : Out 1 : P=9 & L=12 › Out 2 : P=6 & L = 8

SM 03 **2L/3 & 2P/3** **LEDs (E)** ☼ ○

Same principle with different math.
e.g. : Out 1 : P=5 & L=16 › Out 2 : P=3 & L = 11

SM 04 **L-P & P/2** **LEDs (E)** ☼ ☼

Note : In this mode, the sequence played by Out 2 is automatically reseted by the one played by Out 1.

C. Random (RD)

This mode adds randomness to the second output.

SM 01 **No Random** **LEDs (E)** ○ ○

Out 2 & Out 1 are the same, no randomization.

SM 02 **Soft Fill** **LEDs (E)** ○ ☼

Out 2 is the same as Out 1 but is randomly reading the L/2 & P/2 associated sequence.

SM 03 **Hard Fill** **LEDs (E)** 

Same principle as the previous Sub-mode but with a higher probability to read the associated sequence.

SM 04 **Full Random** **LEDs (E)** 

Out 2 is playing a randm sequence, the probability to get a hit at each step is controlled by the Pulses value.

D. Dual Mode (DL)

In this mode, Out 1 & Out 2 can be set independently.

SM 01 **Out 2** **LEDs (E)** 

Set the second output

SM 02 **Out 1** **LEDs (E)** 

Set the first output

Note : The potentiometers have to reach the previously set value to be effective, until that they are disabled to avoid value skips when switching from a Sub-Mode to another. This feature is shown causes the Length+ to flash. In this mode, CV inputs are disabled.

E. Record (RC)

This mode allows to record sequences with the + & - buttons. There is only two Sub-modes : Play and Rec.

SM 01 **Record** **LEDs (E)** 

The pattern played on the - button is assigned to Out 1 and the one played with the + button to Out 2. Both sequences are quantized by the module.

SM 02 **Play** **LEDs (E)** ○ ○

The recorded loops are played.

Note : The Pulses potentiometer have a specific behavior in this mode. Turned fully counter clockwise, it mutes the outputs as in the other modes but turned fully clockwise it causes the outputs to roll (delivering a trigger at each step). You can control this behavior with the Pulses CV too.

Installation

The Knight's Gallop requires a standard 2x5 pin eurorack connector. Make sure the red stripe on the ribbon cable is oriented on the -12V side of the board.

Technical Information

Size: 8 hp
Depth: 25 mm
Current Draw: 22 mA @ +12V / 0 mA @ -12V
Input Voltages: 0 - 5V
Output Voltages: 0 - 5V

Credits

Product design and engineering:
François Gaspard

Product and brand design:
Steve Hackx / MadeInside™

Beta Testers:
Hugo "Ucture" Ficher, Bj_gzp, Supercrysalis, Harold Osica, Mudd Corp & Nicolas Ripit.

Many thanks to **Erik Lukac** for the precious advices.

www.shakmatmodular.com