

TimeFrog 1.3 Owner Manual rev1

Index

1	Introduction.....	2
1.1	What is not the timeFrog ?.....	2
1.2	What is the timeFrog ?.....	2
1.3	What is it for ?.....	2
1.4	What is new.....	2
1.5	What are the differences between tfI, tfII and tfIII.....	2
1.6	Why is the firmware version so important ?.....	3
1.7	Is 1.3 firmware back-compatible to tfII ?.....	3
1.8	How to upgrade firmwares ?.....	3
2	TimeFrog III.....	4
3	Getting started.....	5
3.1	Perform.....	5
3.2	Sequencer.....	5
4	Firmware 1.3.....	6
4.1	Introduction.....	6
4.2	Starting	6
4.3	Slots.....	7
4.4	Perform.....	7
4.5	Sequencer.....	9
4.6	Synoptic.....	10



1 Introduction

Here is the manual for the firmware 1.3 of the timeFrog.
We first begin with a light FAQ in order to delimit field, then is the hardware version III described, and finally is spoken about the functions, from high level to low one.

1.1 What is not the timeFrog ?

- The timeFrog is not a sound synthesizer or other hardware generating audio signal. It only outputs digital messages to control them.
- Despite the appearances, the timeFrog is not a MIDI controller. It is placed as a digital instrument.

1.2 What is the timeFrog ?

The timeFrog is a device for MIDI messages handling.
Handling means there receiving, generating, processing and sending.

In one hand, it relies on an electronic & mechanic design, in the other hand is finality is mainly determined by the software piece embedded, which name is Firmware.

1.3 What is it for ?

The timeFrog is a versatile MIDI toolbox, designed for triggering and controlling software and hardware instruments.

Since MIDI is designed for computer music, targets are mainly

- composing / sequencing / sampling / mixing software
- virtual instruments
- hardware instruments

Main purposes are :

- Studio use, sound design, such has song composing or rhythm research.
- Live use, concentrating several types of actuators in one hand and providing advanced automation and sequencing modes in the other.

1.4 What is new

- The large metal wheel allow to play with different feeling has usual, by the sensitivity and the inertia of the actuator.
- The software piece explore brand new kinds of data processing, leading to inedit ways to compose.

1.5 What are the differences between tfI, tfII and tfIII

timeFrogI is the very earlier model, a simple wheel and automation MIDI tool.

timeFrogII is the first serial batch, embedding 16 actuators, and advanced software for note triggering and sequencing.

timeFrogIII is the second serial batch, based on the same hardware it includes extensions ability for better integration with oldschool hardware such as analog synthesizer.

1.6 Why is the firmware version so important ?

From the beginning, we distinguish hardware from software. As the hardware constitute a physical structure, the software (or firmware) is really the heart of the device.

We are there in a fast software evolving process : nothing is perfect, and very often, rebuild software from scratch is a concrete way to innovate and progress.

For the timeFrog, we have until now 4 firmware versions, each offering different way to exploit the same hardware.

This allow the user to choose the way adapted to its use. The basic structure for the firmware is also available for customized uses.

Instruments should not die unless good reasons, and obsolescence is not a good one.

1.7 Is 1.3 firmware back-compatible to tfII ?

Yes, the basics of firmware is back compatible. Note that enhanced firmware version exists and are designed for handling tfIII extensions plugins. Those versions are still compatible but should hangs on plugins features.

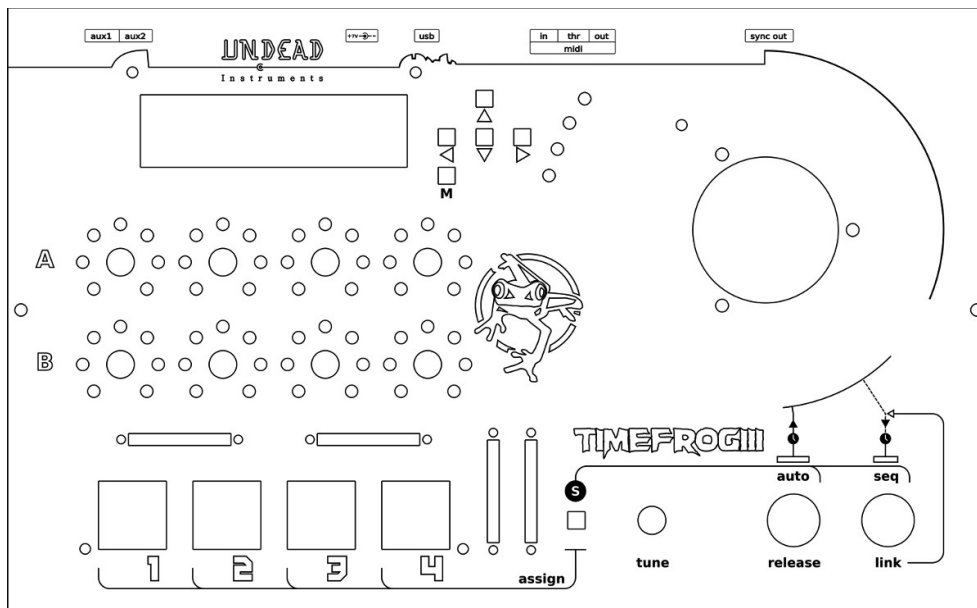
1.8 How to upgrade firmwares ?

The full procedure is explained on www.undeadinst.com, using public tools.

The user can manage his firmware library and upgrade or downgrade when needed.

2 TimeFrog III

Layout



Connections

sync out: pulse output for analog sync'ed hardware, TTL level

MIDI out, through, in:

USB:

Power supply: although the timefrog is bus-powered, it can be externally powered with a DC adapter. Tension range is 6V – 9V

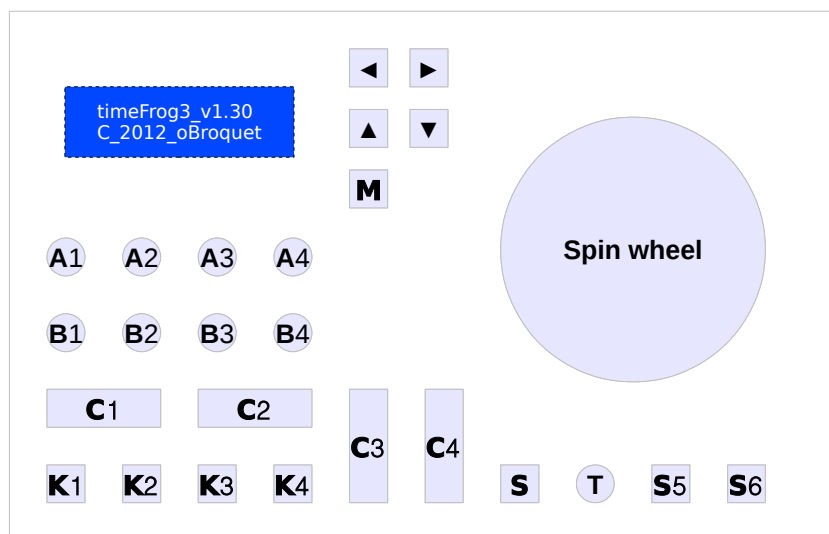
aux1: internal pinout report (10p). Not implemented yet, reserved for integration of external boards.

Aux2: internal pinout report (16p). Not implemented yet, reserved for integration of external boards.

Naming conventions

T refers to a coder

T refers to a button



3 Getting started

Short tour of timeFrogII features:

First, plug the device, make sure it is recognized by your system if you want to use it with USB.

Build your midi chain using hardware or software

Press **Kx** to send basic midi notes.

3.1 Perform

- Configure your slots: midi channel, note and velocity.
 - T** Navigate in parameters. Refer to §4.4 for more details.
 - ◀▶** Navigate in parameters. Refer to §4.4 for more details.
 - ◀▶** Navigate in 32 slots 4 by 4.
 - A** Change top line parameter.
 - B** Change bottom line parameter.
- Test your settings by manually triggering using the keys **Kx**
- Assign a slot to the disc using **S+Kx**
- Trig the slot using the disc
- Start/Stop the automate **S+S5**
- Change the automate speed (SPB) **S+T**
- Change the increment & velocity parameter
- Change the mode parameter to enable transposition

3.2 Sequencer

- Go to sequencer view **M+▶**
- Go to slot view **A1**
- enable steps using **Kx**, select the slot to using **B**
- navigate in steps with **A2**, in voices with **A3**
- start/stop the sequencer **S+S6**
- change the sequencer speed (BPM) **T**
- fine tune velocity and length within appropriate view **A1**

4 Firmware 1.3

4.1 Introduction

The firmware 1.3 is based upon a software concept the slot, here explained.

The **slot** is the main internal concept in the software: this is the entity that is triggered by direct user action or by planned action (automation or sequencing).

A slot is constituted by two kinds of parameters:

- the MIDI config part, where are stored:
 - the channel
 - the kind of slot: control/change or note
 - the id of the slot: for notes, the id is the pitch, for control change, the id is the CC number
 - the slot value: for notes, it is the velocity, for control changes, it is the value of the controller
- the control part, where are stored parameters controlling the variations each time a slot is triggered
 - the increment is the value added before each time the slot is triggered
 - min and max define the range of variation of the value
 - limit define the way of looping for values
 - the octave represent the gap from the pitch defined for notes slot.
 - The mode defines the transpose variation rules
 - Pa1 & Pa2 are extended parameters for transpose variation

Philosophically, the slot introduce an indirection level for configuring messages emitted.

Basically, this concept permit to select specific note pitches to form accords, and make them evolve without dissonance.

The **Perform view** allows the user:

- to **direct trig slots**, using keys and spin wheel
- to access and **tune slots parameters**

In background, there are several ways of **indirect slot triggering**:

- the slot **sequencer** lets the user to compose measures on 16 steps/4 voices. Each step
- the **spin** automation simulates the wheel rotation and by this way the triggering of spin-assigned slots.

4.2 Starting

After the splash screen, the timeFrog displays the perform view, use

M+ or **M**+ to change view.

Here is the view order:

Slots <> **Perform** <> **Sequencer** <> **Service**

4.3 Slots

This view allows to generate the slots configuration
Parameters in *italic* have no effect versions before 1.32.

Main view		Ga Me Rdm MCC NSt Cha Rdn Vel		
Ax	gamme	Merge (mode)	<i>NoiseM</i>	<i>ModeCC</i>
Bx	Start note	channel	<i>NoiseN</i>	velocity
Kx	Fill slots, reset evolving parameters	Fill slots	Fill Channels	Fill velocity

4.4 Perform

In perform mode, the view is split into 4 columns, you can edit the parameters for 4 slots.

View order:

inc,velo <> min,max <> octave,mode <> pa1,pa2 <> channel,pitch/id control

T coder is used for change parameter view.

◀ ▶ navigate in 32 slots 4 by 4.

A row of coders allow to change top line parameter.

B row of coders allow to change bottom line parameter.

M displays current slots, and current parameters.

Increment and velocity view		001 001 001 001 126 110 120 065		
Ax	Increment for slot 0	Increment for slot 1	<i>Increment for slot 2</i>	<i>Increment for slot 3</i>
Bx	Velocity for slot 0	Velocity for slot 1	<i>Velocity for slot 2</i>	<i>Velocity for slot 3</i>
Kx	Trig manually the slot	Trig manually the slot	Trig manually the slot	Trig manually the slot
S + Kx	Assign the slot to the wheel	Assign the slot to the wheel	Assign the slot to the wheel	Assign the slot to the wheel
M		Slots 00 - 03 Incr Velo		

Slots parameters :

- channel: MIDI channel for the slot
- note: Midi pitch for the note
- CC: controller number
- increment: gap added or subtracted to velocity before triggering. 0 disable velocity changes.
- velocity: value used for the velocity of the note or the value of a controller
- min,max: range for velocity
- octave : effective note transposition in octave
- mode : octave variation control : T : transpose map, R : random map, L : limit variation amplitude
- pa1 : map generation control : 0,1 : binary, 2 ternary, <14 : random map, 15 : large range note randomizer, 14: seq start stop, octave used as multiplier for BPM
- pa2 : map index

Slots indicators:

The status of a slot is displayed in the perform view:

- top line indicator: note status (on/off)
- bottom line indicator: slot assigned (o), slot locked (+), slot assigned & locked (*)

Commands :

T change edited parameter:
inc,velo> min,max > octave,mode > pa1,pa2 > channel,pitch/id control

T in view channel,pitch/id control: switch slot type: CC or note
in other views: enable value looping around the range (min,max).

Kx trig slot

S+**Kx** assign slot to spin wheel

Ax strict limit velocity to min-max instead of looping values

Ax in channel/pitch view : toggle note/control change

◀ change edited slot range : 00-03 > 28-31

▶

S5 release notes and sounds on all channels

M+**S5** automation on/off

S6 scrub on/off

M+**S6** seq on/off

4.5 Sequencer

- A1** change seq view : main > slot > velo > length > fill
- A2** change edited seq range : [0-3] > [12-15]
- A3** change edited voice : 0 > 3
- A4** change seq current position: 0 > 15
- Bx** change parameter (slot / velo / length)
- Kx** enable step

Main view		Sin_SOut_In_Out NoneNone_00_15_		
Ax	Change view	Not used	Not used	Not used
Bx	Synchro In	Synchro Out	Start point	Stop point
Kx	Synchro In	Synchro Out	Start point	Stop point
Slot choice view		Slot_[00_03]3_01 _01_____01_01		
Ax	Change view	Step range	Voice	Seq position
	Slot number	Slot number	Slot number	Slot number
Kx	Step enable	Step enable	Step enable	Step enable
Velocity tune view		Velo_[00_03]3_01 _80_____121_101		
Ax	Change view	Step range	Voice	Seq position
Bx	Velocity	Velocity	Velocity	Velocity
Kx	Step enable	Step enable	Step enable	Step enable
Len tune view		Len_[00_03]3_01 _001_____10_020		
Ax	Change view	Step range	Voice	Seq position
Bx	Len (24/BPM)	Len (24/BPM)	Len (24/BPM)	Len (24/BPM)
Kx	Step enable	Step enable	Step enable	Step enable
Fill view		Fill_[00_03]3_01 250_862_401_585_		
Ax	Change view	Step range	Voice	Seq position
Bx	Randomize slot enable	Increase slot number	Increase velocity	Increase length
Kx	Reset	Randomize slot number	Randomize velocity	Randomize length

Fill view:

values displayed in are just filling indicators (mean of pitches + mean of velocities)

4.6 Synoptic

