TimeGridPlayer Manual

|  |  |
| --- | --- |
|  | Command Driven Sequencer |

Purposely left blank

Content

[Introduction 1](#_Toc153833585)

[Technical Details 1](#_Toc153833586)

[Scales 1](#_Toc153833587)

[Midi 2](#_Toc153833588)

[Connectors 2](#_Toc153833589)

[Dimensions 2](#_Toc153833590)

[Weight 2](#_Toc153833591)

[Hardware Version 2](#_Toc153833592)

[Contacts 2](#_Toc153833593)

[User Manual 2](#_Toc153833594)

# Introduction

The intent of the ChordsBOX is to help those less fluent on the domain of Chords when composing music. You may already own one or more synthesizers and still feel intimidated with the keyboard when it comes to be more elaborate with it. Or you are a One Finger player that despite knowing and understanding chords are unable to play them on time. If this is your case, fear not, because ChordsBOX is the truly One Finger player device.

Normally everyone is already familiarized with the Major scale and their Chords, after all it’s the first scale that people learn, this happens because the Major Scale is a scale composed of only white keys. All problems start when it’s time to work with other scales that more likely than not demand the consideration of those black keys. The ChordsBOX comes to help by working as a router from single keys to the elaborated multitude of keys that chords represent. It goes further with the Compounded Split, where with the usage of just white keys you can trigger all the notes concerning a specific Chord or Scale no matter how complex it may be. And on top of that, you can switch them with a press of a single black key!

This way, to make things easier, the concept of the ChordsBOX is Octave centered, not Note centered. This means that the programmed Chords or Scales are played accordingly to the Octave where the Trigger Keys are pressed, and only afterwards, are those key positions taken into consideration. For instance, a programmed C minor 7th on a given Trigger Key will always be a C minor 7th, what changes is only its Octave depending on which Octave that key is triggered. This applies too to the Compounded Split, where the individual triggered keys set the Octave of the triggered notes without changing the respective root notes.

# Commands

## Rulers

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Type** | **Parameters** | **Purpose** |
| actions | Selector | - | Filters all Rulers to Actions Rulers only |
| add | Incrementor | Ruler, - | Adds a given Ruler to the existing Rulers. Parameter can be left empty in which case adds an extra copy of existing Rulers from which this command is called for. |
| add\_lines | Incrementor | line, amount=1, id=None |  |
| arguments | Selector | - |  |
| beats | Selector | - |  |
| clone | Editor |  | Replicates per type of Ruler the data of the first respective Ruler |
| clone\_lines | Editor |  | Replicates per type of Ruler the lines data of the first respective Ruler |
| copy | Incrementor |  |  |
| disable | Editor |  |  |
| disabled | Selector |  |  |
| distribute | Positioner |  |  |
| drag | Editor |  |  |
| drop | Positioner |  |  |
| duplicate | Incrementor |  |  |
| empty | Selector |  |  |
| empty\_lines | Editor |  |  |
| enable | Editor |  |  |
| enabled | Selector |  |  |
| erase\_line | Editor | line, amount=1, id=None |  |
| erase\_lines | Editor | - |  |
| even | Selector | - |  |
| every | Selector | nth\_ruler, first=1 |  |
| exclude | Selector | index=1 |  |
| expand | Positioner | increments=4 |  |
| expand\_lines | Incrementor | amount=1 |  |
| filter | Selector | ids=[], type=None, links=[], positions=[], position\_range=[], lines=[], measures=[], beats=[], steps=[], enabled=None, on\_staff=None, player=None |  |
| first | Selector | - |  |
| float | Positioner | - |  |
| function\_lines | Commander |  |  |
| group | Positioner | - |  |
| head | Selector | elements=1 |  |
| ids | Selector | \*ids |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| single | Selector | index=1 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Midi

The Midi messages are processed by the ChordsBOX in the following way:

* All Channel type Midi Messages, like Note Off, Note On, Control Change, Pitch Bend, etc..., when sent to the configured Input Channel are always processed and sent to the configured Output Channel;
* All other Midi Messages regardless the type excluded from the previous point, are always sent straight away to the Midi Output if, and only if, the Channel Thru is enabled (Y);
* The USB ports and the Midi ports are used in unison as if they were a single one;
* The Midi over USB of the ChordsBOX works as Plug and Play on any PC.

## Connectors

* 2x 5 pin Female DIN
* 1x USB-B Female (Power)

## Dimensions

* 110x80x50mm / 4.33x3.15x1.97inches

## Weight

* 245 grams / 8.64oz

## Hardware Version

* 1.0 DEC/2021

# Contacts

Before proceed with a contact please check the Troubleshooting first, most of the time the misbehavior of the ChordsBOX is due to its wrong configuration and the Troubleshooting chapter was written considering all typical fail scenarios.

|  |  |
| --- | --- |
| **Contact** | **Address** |
| E-mail | [chordsbox@gmail.com](mailto:chordsbox@gmail.com) |
| Google Drive | <https://drive.google.com/drive/folders/1WswN5GRYlCtFYt_f1miNCH2Fd8jF7vl4?usp=sharing> |

# User Manual

* Version 1.0 – August/2022