**Mestra**

**Combined Controller**

Design

Michel Keijzers, © 2017

# History

Table : History

|  |  |
| --- | --- |
| **Date/period** | **Actions** |
| Nov 8, 2017 | Initial version |

# Table of Contents

Contents

[History 1](#_Toc498126522)

[Table of Contents 2](#_Toc498126523)

[List of Tables 2](#_Toc498126524)

[List of Figures 2](#_Toc498126525)

[1 Introduction 3](#_Toc498126526)

[2 Folder Structure 4](#_Toc498126527)

# List of Tables

[Table 1: History 1](#_Toc498126528)

# List of Figures

# Introduction

## General

This document shows the design of the combined Controller and MIDI/DMX slaves.

## Typical Usage

Kronos MIDI OUT -> Mestra MIDI IN 1

Studiologic MIDI OUT -> Mestra MIDI IN 2

Behringer MIDI OUT -> Mestra MIDI IN 3

(KeyTar -> Mestra MIDI IN 4)

Mestra MIDI OUT 1 -> Kronos MIDI IN

Mestra DMX OUT 5 -> DMX Chain

OR

Kronos MIDI OUT -> Mestra MIDI IN 1

Studiologic MIDI OUT -> Behringer MIDI IN

Behringer MIDI OUT -> Mestra MIDI IN 2 (merged with Studiologic)

Mestra MIDI OUT 1 -> Kronos MIDI IN

Mestra DMX OUT 3 -> DMX Chain

# Requirements

# Design Decisions

# Software

## Trigger Handling

### MIDI

#### Translation to MC

Always one MC

If note range = ALL or ~: create trigger in ALL-notes trigger table

Else if note range: create trigger in OCTAVE\_notes trigger table and specific NOTE trigger table \*

Else if single note: create trigger in single NOTE trigger table

\* example Note range F4~G7 results in 7 single-note triggers for F4 to B4 + 2 octave triggers for octave 5 and 6 and 8 single note triggers for C7-G7, total: 7 + 2 + 8 + 17.

The trigger table is defined by a hash key depending on MC, Note. Note can be:

Single note triggers: 0-127

Octave note triggers: 128-137 (octave 1..10)

All note trigger: 255

#### When a note is received

3 hash keys are calculated:

* All notes trigger table (using also MC)
* Octave trigger table (using also MC, note (octave))
* Single note trigger table (using also MC)

For all three tables, all commands are executed (and checked if the trigger is enabled and condition is met since other triggers can end up in the same trigger table).

## Memory Usage

## Performance

### MIDI

To loop through 100 triggers within 3 tables, every trigger taking 10 instructions to check, taking 50 clock cycles. These are 50,00 clock cycle. Assuming 168 MHz this will cost 5,000/168,000,000 = 0.029 ms, thus very less.

The execution of the programs (assume 5 on average high), cost 500 instructions of 4 clock cycli each, resulting in 10,000 clock cycli, which results in 0.059 ms (assuming 168 MHz), also vey less.

**Memory usage**

Assuming there are 256 hash keys / trigger tables, and per entry the following information is stored:

* Type (MIDI/PedSw): 3 bits
* Enabled 1 bit
* For e.g. note on: MC 4 bits
* For e.g. note on: Note 8 bits (for oct)
* Program index 16 bits

Total: 32 bits -> 4 bytes

Thus total storage:

* Table start offsets: 256 \* 2 bytes (start) = 512 bytes
* Tables itself: 1,000 (entries) \* 4 bytes = 4,000 bytes