**Mestra**

Design Decisions

Michel Keijzers, © 2017

# History

Table : History

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

# Table of Contents

Contents

[History 1](#_Toc498126356)

[Table of Contents 2](#_Toc498126357)

[List of Tables 3](#_Toc498126358)

[List of Figures 3](#_Toc498126359)

[1 Introduction 4](#_Toc498126360)

[2 Environment 5](#_Toc498126361)

[2.1 IDE for Arduino 5](#_Toc498126362)

[2.2 IDE for STM32 5](#_Toc498126363)

[3 Design decisions 6](#_Toc498126364)

[3.1 Micro Controller selection 6](#_Toc498126365)

[3.1.1 Alt 1: Arduinos Only 6](#_Toc498126366)

[3.1.2 Alt 2: STM32 Mixed with Arduinos 6](#_Toc498126367)

[3.1.3 Analysis 6](#_Toc498126368)

[3.1.4 Pros/Cons list 6](#_Toc498126369)

[3.1.5 Selected Alternative 6](#_Toc498126370)

[3.2 Controller and Slave Split up 7](#_Toc498126371)

[3.2.1 Alt 1: Combined Controller / MIDI / DMX Slave 7](#_Toc498126372)

[3.2.2 Alt 2: Separate Controller 7](#_Toc498126373)

[3.2.3 Alt 3: Separate MIDI and DMX Slaves 7](#_Toc498126374)

[3.2.4 Pros/Cons list 7](#_Toc498126375)

[3.2.5 Selected Alternative 7](#_Toc498126376)

[3.3 Power 7](#_Toc498126377)

[3.3.1 Introduction 7](#_Toc498126378)

[3.3.2 Alt 1: 220V->12V adapter 7](#_Toc498126379)

[3.3.3 Alt 2: Battery 7](#_Toc498126380)

[3.3.4 Alt 3: MIDI power 7](#_Toc498126381)

[3.3.5 Alt 4: USB cable or USB powered adapter 7](#_Toc498126382)

[3.3.6 Pros/Cons list 8](#_Toc498126383)

[3.3.7 Selected Alternative 8](#_Toc498126384)

[4 Design 9](#_Toc498126385)

[4.1 Slave Loops 9](#_Toc498126386)

# List of Tables

[Table 1: History 1](#_Toc498126394)

[Table 2: Micro Controller Selection 6](#_Toc498126395)

[Table 3: Split up 7](#_Toc498126396)

[Table 4: Power Comparison 8](#_Toc498126397)

[Table 5: Power source location/type 8](#_Toc498126398)

# List of Figures

# Introduction

This document describes the design issues. Design alternatives which have been selected are worked out in other documents in more detail.

# Environment

## IDE for Arduino

The default IDE to be used for Arduino is the Arduino IDE. However, this IDE has the problem that when using more than approximately 10 files, the file names do not fit in the upper status bar. Therefore, another alternative will be chosen.

There are many options, like Eclipse, Microsoft Visual Studio plugins, and plugins for editors. Since I am used to Visual Studio, I selected the VisualMicro extension for Visual Studio.

## IDE for STM32

For STM32, the Eclipse IDE will be used, since it is free and user friendly.

Other alternatives are KEIL, AVR Studio.

# Design decisions

## Micro Controller selection

### Alt 1: Arduinos Only

The following Arduinos are good candidates:

* Arduino Micro
* Arduino Uno
* Arduino Mega

Arduino Due’s are out of focus because of the high price.

### Alt 2: STM32 Mixed with Arduinos

Use STM32s, the following STM32 are good candidates:

* STM32F103C8T6
* STM32F407VET6

### Analysis

The controller (including or even excluding MIDI slave) will need much more SRAM than an Arduino can deliver. Therefore, the Arduino’s are not possible for the controller. For other slave types Arduinos can be used.

### Pros/Cons list

Table : Micro Controller Selection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Alt 1**  **Arduino Uno** | **Alt 1**  **Arduino Mega** | **Alt 2**  **STM32 F103C8T6** | **Alt 2**  **STM32 F406VET6** |
| Price | E 3.50 | E 6.50 | E 1.50 | E 10.50 |
| Memory (SRAM, KB) | 2 | 8 | 20 | 192 |
| Voltage | 5V | 5V | 3.3V | 3.3V |
| UART/USART | 1 | 3 | 6 | 6 |
| CAN | 0 | 0 | 1 | 2 |
| USB (Slave) | 0 | 0 | 1 | 1 |
| Learning curve | +++ | +++ | --- | --- |
| Documentation available | +++ | +++ | --- | --- |
| Help Community | +++ | +++ | --- | --- |
| Space/dimensions | -- | --- | +++ | + |

### Selected Alternative

Use an Arduino whenever possible (regarding specifications), otherwise use STM32.

**Selected: Alternative 1 or 2, depending on situation**

## Controller and Slave Split up

### Alt 1: Combined Controller / MIDI / DMX Slave

Use one device which is both Controller, MIDI slave and DMX slave.

### Alt 2: Separate Controller

Use one device which is Controller, and one device which is both MIDI and DMX slave.

### Alt 3: Separate MIDI and DMX Slaves

Use three devices: a separate Controller, a MIDI slave and a DMX slave.

### Pros/Cons list

Table : Split up

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Property** | **Alt 1** | **Alt 2** | **Alt 3** | **Remarks** |
| Cost | ++ | ++ | - |  |
| Effort | ++ | ++ | - |  |
| Performance |  |  |  | Analyze |
| SRAM | -- | - | ++ | SRAM divided |
| CPU Power | -- | -- | ++ | CPU power divided |

### Selected Alternative

Although SRAM and CPU power can be divided (no sharing needed), for the first iteration a combined Controller/MIDI/DMX is selected.

**Selected: Alternative 3**

For this alternative, alternative 2 (STM32F407VET6) will be needed.

## Power

### Introduction

Regarding power, three options are taken into account to meet REQs GenHx.

### Alt 1: 220V->12V adapter

Generic adapters can be used to provide 12V, and up to 2A.

### Alt 2: Battery

A 9 V battery or 3 x 1.5 V AA batteries can be used for wireless devices.

### Alt 3: MIDI power

A MIDI cable can provide power too, which is 5V.

### Alt 4: USB cable or USB powered adapter

An USB cable (5V, 500 mA) or USB powered adapter can be used.

### Pros/Cons list

Table : Power Comparison

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item / Source:** | **Alt 1**  **220->12V Adapter** | **Alt 2**  **Battery** | **Alt 3**  **(MIDI Input power)** | **Alt 4**  **USB (adapter and/or cable)** |
| Minimum power | +++ | ++ | - | + |
| Cost of batteries | +++ | ---- | +++ | +++ |
| Continuity during usage | +++ | -- | - | +++ |
| Extra cable/plug needed | --- | + | +++ | -- |
| Wall outlet needed | --- | +++ | +++ | -/+ |
| Reliable power | +++ | + | - | +++ |
| Enclosure modification | ++ | --- | --- | +++ |

### Selected Alternative

The selected alternative depends on the device.

However, for the MIDI device, alt 3 will not be selected to keep consistent with other devices and because MIDI power is not reliable.

USB power is the default way of providing power while breadboarding. Depending on the slave type, USB power might be useful, either by cable (e.g. from a USB connection of a synth) or via a 5V USB adapter. 500 mA should suffice for devices. Also with one wall socket, 2 (or even 3 or 4) devices can be powered.

A 9V battery has the possibility of being empty during use, which is totally unacceptable. Of course inserting a 9V battery before each use is a solution, but is quite costly. This alternative is suitable for wireless devices.

Also, the power source depends on the location for the Controller or slave. See the following table.

Table : Power source location/type

|  |  |  |
| --- | --- | --- |
| **Device** | **Location** | **USB in neighbourhood** |
| Controller | Not relevant | No |
| MIDI | Near MIDI In/Out | Yes |
| USB MIDI | Near USB In/Out | Yes |
| DMX | Near DMX controller/light | No |
| Drum Pads | Near drums | No |
| Drums Trigger | Near drums | No |
| Pedal/Switches | Near pedals/switches on floor | No |
| Remote | Can be anywhere | No |
| Audio | Near guitar gear/floor | No |
| Microphones | Can be anywhere | No |
| Proximity | Can be anywhere, high likely near keyboard | No/Yes |
| GUI | On keyboard / in sight | Yes |
| Debug | - | Yes |

# Design

## Slave Loops

Each slave has the same loop:

* If slave has an incoming message (from the controller):
  + Read incoming messages
  + Convert to specific signal/message for device
  + Send signal/message to device specific output(s)
  + Show notification (LED)
* If device has a incoming (device specific) signal/message
  + Convert to generic message
  + Send to Controller
  + Show notification (LED)