# ELT-23056: MySynth

## Project Planning

Project planning refers to everything we do to set up our project for success. It’s the process we go through to establish the steps required to define our project objectives, clarify the scope of what needs to be done and develop the task list to do it. Project planning is the process of establishing the scope and defining the objectives and steps to obtain them.

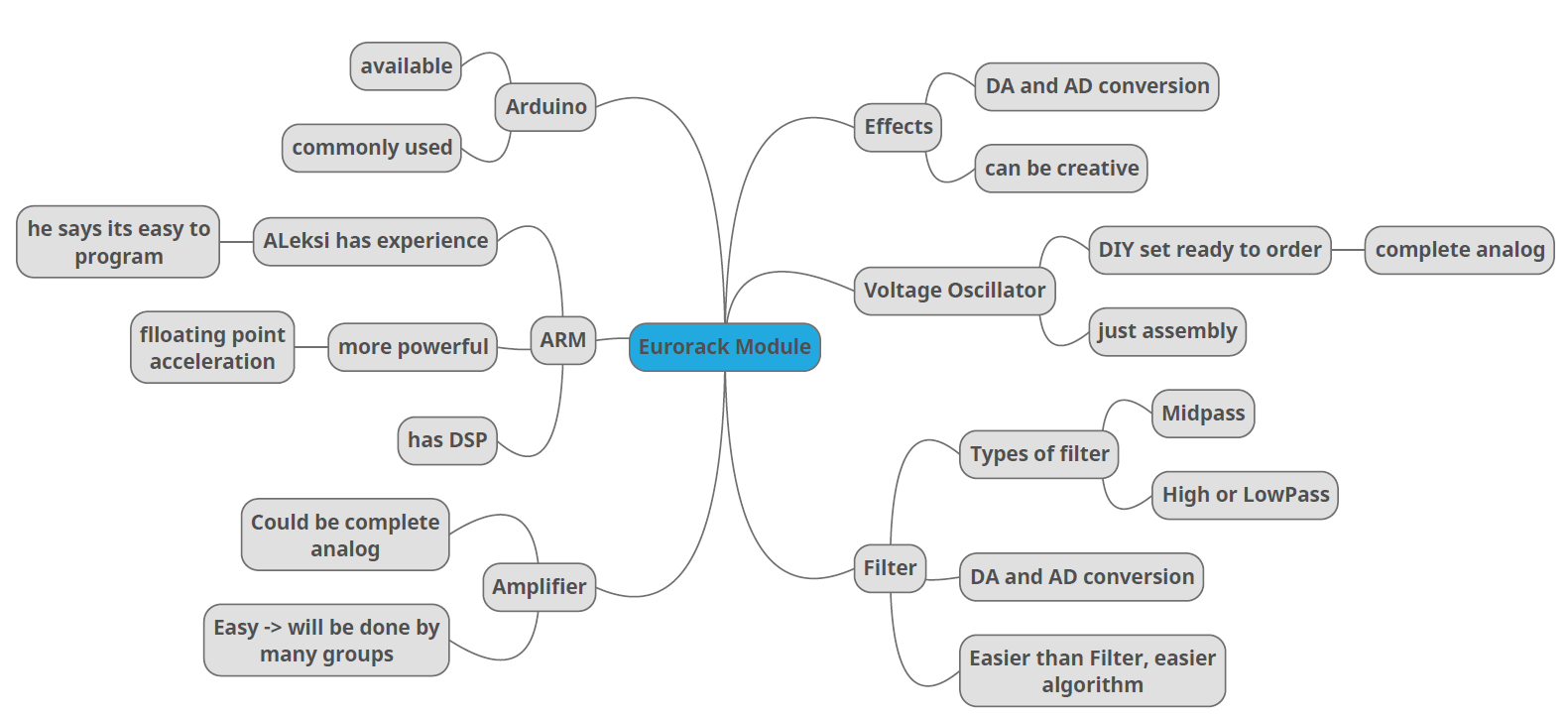
We started planning our project immediately after the first lecture. After detailed discussion on the ideas of each member, we agreed on making an effects module and implementing it as our project. After the initial decision of choosing the topic, we followed the general steps of project planning to come up with a concrete timeline for our effects module.

## Ideation Phase

Ideation is the process of forming ideas and concepts. It means creating new ideas to solve specific problems. Brainstorming is the generation of ideas in a face to face mode. Ideation phase is always when brainstorming takes place, and it is one of the most effective ways to explore new opportunities. The ideation phase is all about diverging and generating as many ideas as possible and to narrow things down and select the best approaches to take further and then to implement one good plan finally and to act and obtain the expected results.

When facilitated in a successful way, Ideation is an exciting process. The goal is to generate a large number of ideas — ideas that potentially inspire newer, better ideas — that the team can then cut down into the best, most practical and innovative ones.

While going though our ideation phase and brainstorming, we came up with the following mind map for our project:



## Microcontroller Selection

For construction and implementation of our effects module, we had two options of microcontroller – Arduino and ARM. After going through their data sheets, pros and cons we decided to go with Arduino. We will be using Raspberry Pi for our effects module.

## Features

Our Module to be created shall be fit-able inside a working Eurorack System and provide easy to use simple effects for a modular Synthesizer. The MySynth Effects Module will offer the following functions:

The module features includes as follows:

1. Audio Stream Record and Playback

* Description and Priority

When user doesn’t apply any sound effects, the system should be able to playback the sound stream from the audio stream recording interface.

Priority: High

* Stimulus/Response Sequences

The audio stream from the recording interfaces will be captured by the audio codec on the HW board. The sound stream will be digitalized by the audio codec and transmit to the SoC through I2S, the I2S controller will generate the interrupt to the driver. The driver will then start the DMA controller and read the data from the FIFO of the I2C controller to the system main memory. Then the driver will notice the upper layer software.

The upper layer software will talk to the driver to playback the data received. The driver will start the DMA controller to move the data to the FIFO of I2S controller. Once the FIFO of the I2S controller is full, the data will be transmitted to the audio codec. The audio code will turn the values into analog signal for the playback interfaces in the panel.

* Functional Requirements

REQ-1: The audio input jack in the panel. A standard 3.5-mm audio input jack is needed for the audio input signal from the eurorack.

REQ-2: The audio output jack in the panel.

REQ-3: Audio codec. The audio codec on the board will be responsible for capturing and playback the analog audio stream.

REQ-4: I2S controller. The I2S controller in the SoC is responsible for transmitting the digitalized values between SoC and audio codec.

REQ-5: I2S controller driver. The I2S driver will be responsible for managing the I2S controller

REQ-6: I2C controller. The I2C controller will be responsible for configuring the audio codec.

REQ-7: I2C controller driver. The I2C controller will be responsible for managing the I2C controller.

REQ-8: Upper layer software. The upper layer software will control and manage the audio data

1. Pitch Change

* Description and Priority

Provide the function of changing the pitch of sound to implement several funny sound effects for the users. Most funny sound effects are based on pitch change technology, like the Funny Tomcat.

Priority: high

* Stimulus/Response Sequences

Option 1: A tuner to adjust the pitch. The user will turn the tuner back and forth to change the pitch of the sound. When the user turns the tuner, the electrical level of the tuner will be changed. The electrical level of the tuner will be sampled by an ADC channel inside the SoC. The ADC channel will output the value of the electrical level. The user-space application will read the value and change the pitch according to it.

Option 2: Two buttons to adjust the pitch. One button will be responsible for higher the pitch and another button will be responsible for lower the pitch. When user presses a button, which is connected to a GPIO pin, an interrupt from the GPIO pin will be generated. The user-space application will read the events and adjust the pitch based on the type of the event. (Higher the pitch or lower the pitch)

* Functional Requirements

As the description above, the user input from the HW would be a tuner or two buttons.

SoC receives the external user input

REQ-9 The tuner. The tuner will output the electrical level. The output of the tuner will be connected an ADC channel of the SoC.

REQ-9 Two buttons. The two buttons will output the signal of “high” or “low”. The outputs of the two different pins will be connected to two different pins of the GPIO of the SoC.

Change the pitch based on the external user input

After the SoC received the input, the software will receive the input from the SoC and change the pitch according to it.

REQ-10 The kernel drivers of ADC and GPIO. The driver will receive the interrupts from ADC or GPIO controller and read the values from it, then expose the values to the upper layer software.

REQ-11 The upper layer software will read the values and the events and change the pitch based on the input from the kernel drivers.

1. Physical Modeling Synthesizer

* Description and Priority

Physical modelling synthesis refers to sound synthesis methods in which the waveform of the sound to be generated is computed using a mathematical model, a set of equations and algorithms to simulate a physical source of sound, usually a musical instrument.

Priority: Medium

* Stimulus/Response Sequences

When a button of the keyboard is pressed by the user. The key code will be scanned by the keyboard IC. The keyboard IC will generate the interrupt to the SoC. The keyboard driver will read the key code from the buffer of the keyboard IC through I2C bus. The key code will be reported to the upper layer software. The upper layer software will figure out the note by the mapping table between key code and notes. With the notes, the physical modelling synthesizer library will generate the wave of specific instruments. The upper layer software will playback the wave to the I2S controller driver, the I2C controller driver will start the DMA and move the wave data to the FIFO of I2S controller. The I2S controller will transform the wave data into analog signals and play them through the analog output.

* Functional Requirements

REQ-12: The keyboard with a scan matrix IC.

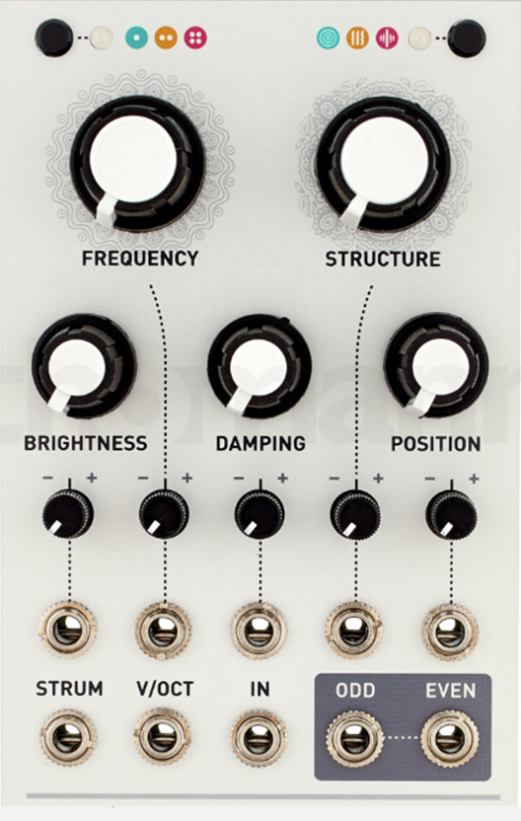
REQ-13: The physical modelling synthesizer.

## Time Planning

Powerpoint

## Project Implementation

## HW Design

**User Interfaces**

The User Interface shall consist of:

• 3 Knobs to control effects

• 3 Buttons to control effects

• LEDs to indicate the functionality of the module

The User Interface should be visually similar to the picture shown on the right.

**Hardware Interfaces**

The module shall have one analog Input:

• Audio Input

• Voltage Range: 0V -5V

The module shall have one analog Output:

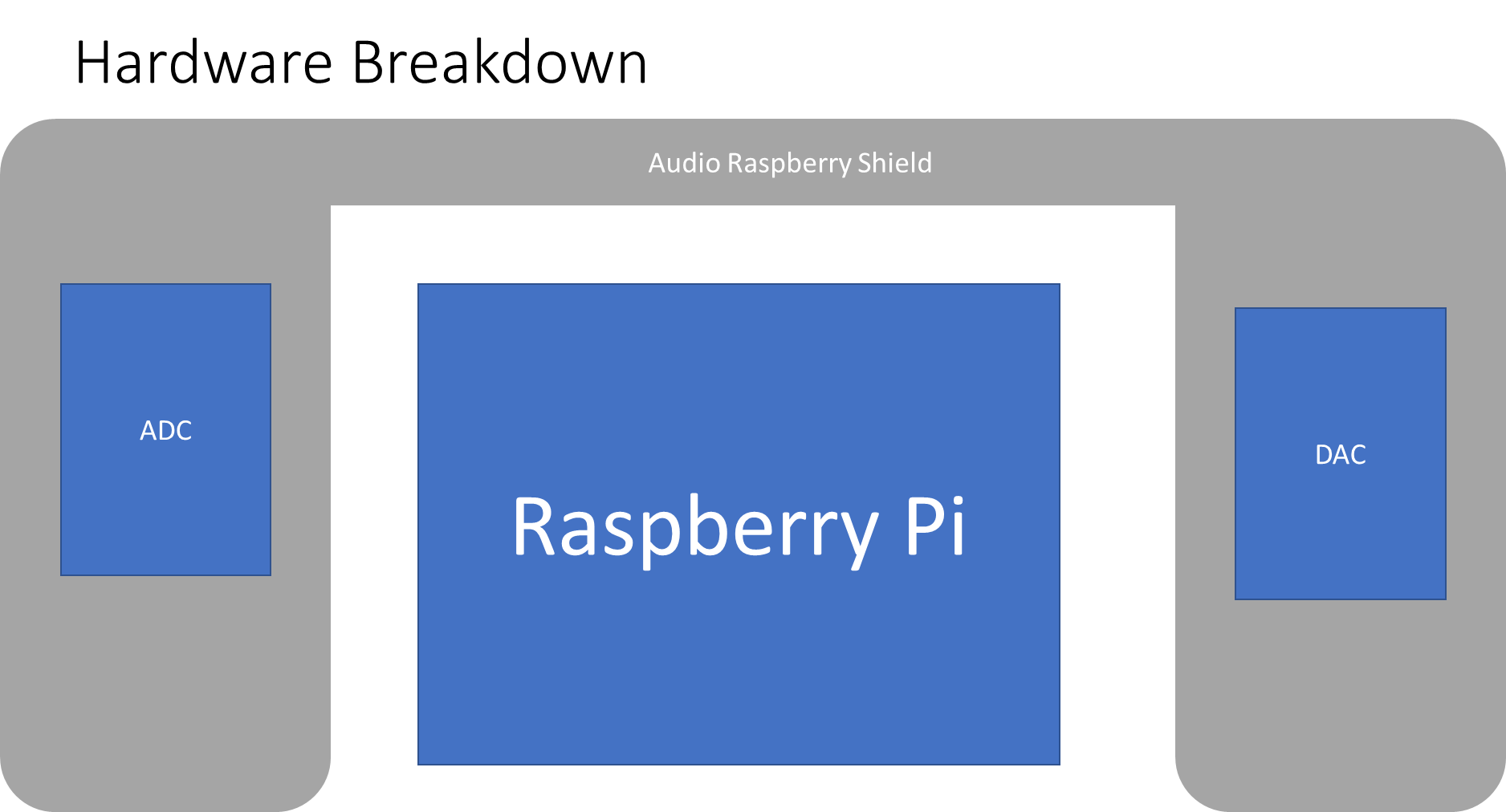
• Audio Output

• Voltage Range: 0V -5V

The Input voltage shall be transformed using a ADC to be used in Software.

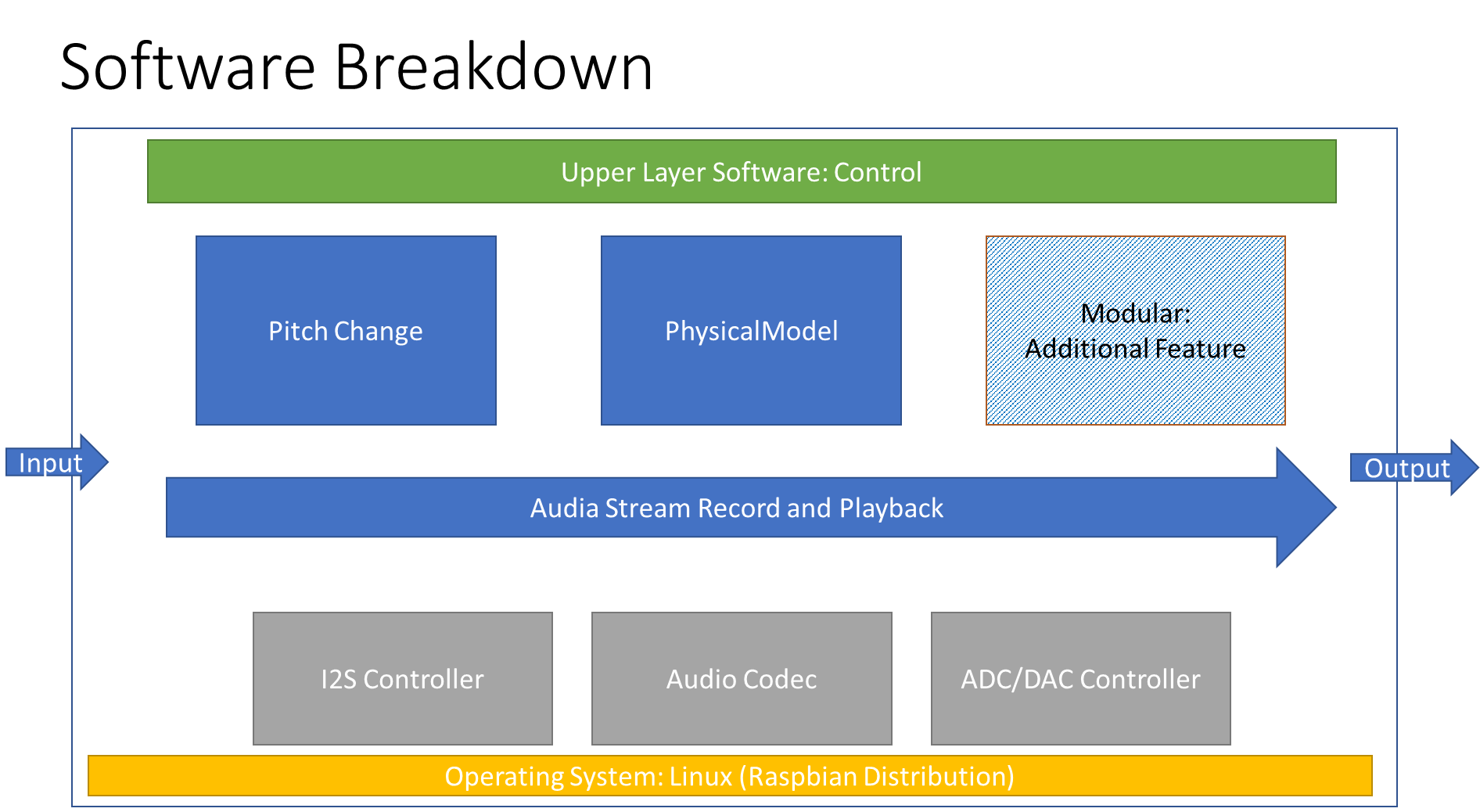
After the Effects described in Chapter 4 have been applied by Software, a DAC shall be used to create the analog Output.

A schematic of the HW design is as follows:



## SW Design

The Software and Hardware shall work within a Eurorack system as specified by Doepfer. The schematic is as follows:



## Project Monetization

## Competition

https://www.thomann.de/gb/mutable\_instruments\_rings.htm

## Testing

Add the Test Plan here

## Risk management

Add the risks and countermeasures here