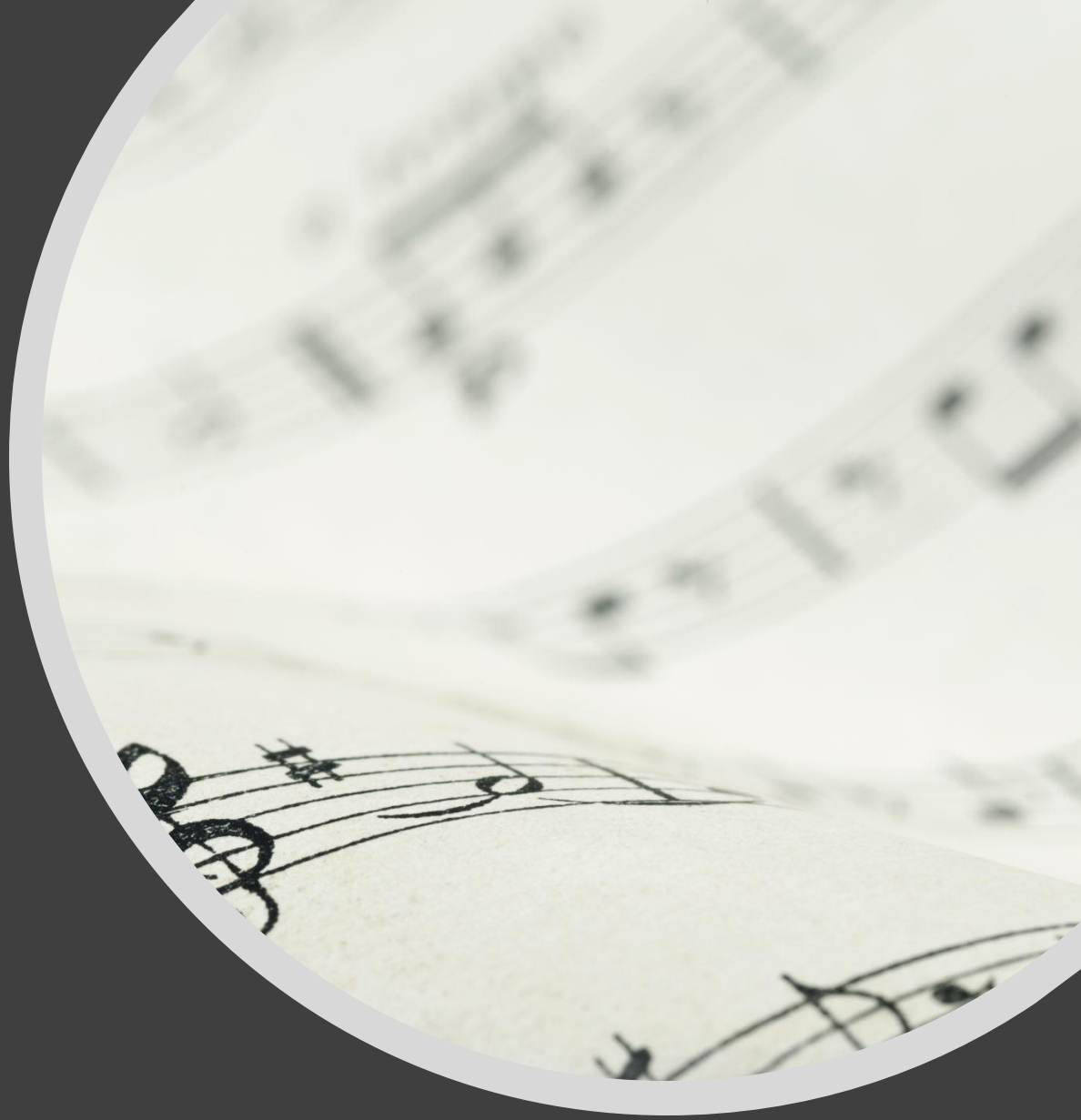
Computer Music Design Presentation MP615 A1

1080923

November 2021

Introduction

- Sometimes best musical ideas are inspired by accidental or random events. It can be difficult to make creative decisions without a starting point such as these.
- This project aims to reliably generate interesting musical ideas using software, which can be incorporated into existing projects or performed live as standalone parts.



Intentions of this project

- Use Max/MSP to build a generative “MIDI ideas machine”.
- Hardware/software crossover
- Convert fixed-architecture synth into a “semi-modular” generative synth. Increase variety of sounds. Generate creative ideas in an organic way.
- Customisation to suit various needs. (Scale lock, range limits for parameters, various levels of probability etc.)
- Simple user-friendly interface.

What prompted this idea?

- *Korg Minilogue,* produces exciting sounds but dynamic sound design is limited by in-built sequencer [1] . Only 1 free LFO, 1 filter envelope, 1 amp envelope
- Not much "*chance*". Sometimes stuck with robotic sound design and limited creativity.
- Not much option for routing modulators
- Modular synthesis expensive, takes a lot of space etc.

What hardware/software will I need to develop/research?

- Improve fluency in Max/MSP.
- Sending/receiving MIDI messages between MAX and an external MIDI synth
- Interface design (visual)

Examples of similar software and hardware

Andrew Norris' Minilogue Max device. Control parameters through DAW. Gives fairly static control [2]

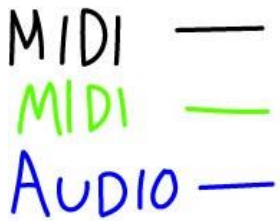
- Eurorack/VCVRack modules such as WALK2 [3]

- I want to build deeper and less predictable textures. For example, modulating these controls with the audio signal from *Minilogue* (FM).
- I want my software to also work with other synths in future.

Signal flow

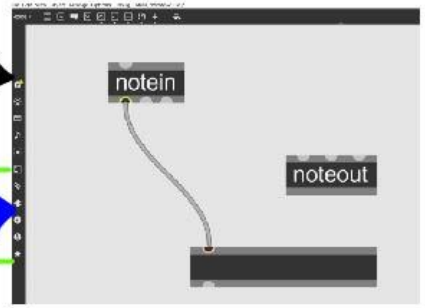
1- MIDI device controls Max sequencer settings

2- Max sequencer sends MIDI data to external synthesizer/DAW



MIDI Controller

Max Patch/Max4Live Device



3- DAW captures audio and MIDI output

DAW

MIDI/Audio Interface

Korg Minilogue

Development plan for this project (1):

1- Achieving basic control over the synthesizer

- Setup basic signal chain in MAX
- Read MIDI data transmitted to and from *Minilogue*
- and create a data conversion table
- Use this data in MAX to communicate simple MIDI CC commands to and from *Minilogue* (note and parameter controls)

2- Creating sequencer in MAX

- Build general layout
- Make signal control routes
- Read MIDI in from controller
- Map sequencer to soft-synths in Live and test

Development plan for this project (2):

3- Integration

- Connect *Minilogue* through MIDI
- Set ranges for synthesizer parameter controls (trial and error)
- Test various approaches toward parameter controls
- Create various templates for different sequence generation styles

4- Finalize

- Design a functional and attractive user interface
- Compile the patch as a Max4Live plugin
- Share!

Current stage of development

- Have purchased most necessary hardware already
- Currently learning to use the Max programming environment.
- In contact with *Andrew Norris* to learn more about his patch and to ask for advice for this project.
- General understanding of MIDI communications (Previously built a portable MIDI sequencer using ATmega 328).

Resolving issues this project may present:

High CPU/RAM Usage- Optimise patch as much as possible 4

Ties with

Glitchy audio (Configure audio sample rate, buffer size etc.) so it runs smoothly

Keeping organised- Separate sections into categories Making notes on each section of the project during development. Saving backup versions of the patch.

MIDI Feedback 5 (Mitigating this is something I will need to learn)

Managing time

In order, I expect the most time-intensive aspect of this projects will be:

1. Assembling the patch in Max.
2. Configuring parameter control ranges
3. Reading and converting MIDI data
4. Designing the user interface

I will focus my time so that my project will be *functional* before the final deadline, and unexpected problems can not cause the project to collapse entirely. (Contingency plans)

What will I achieve during this project?

1

Create functional, *new*, product that actually serves a function when I am creating music.

2

Provide others with something useful I wish somebody had already made!

3

Develop Max engineering skills which can apply to other projects in future

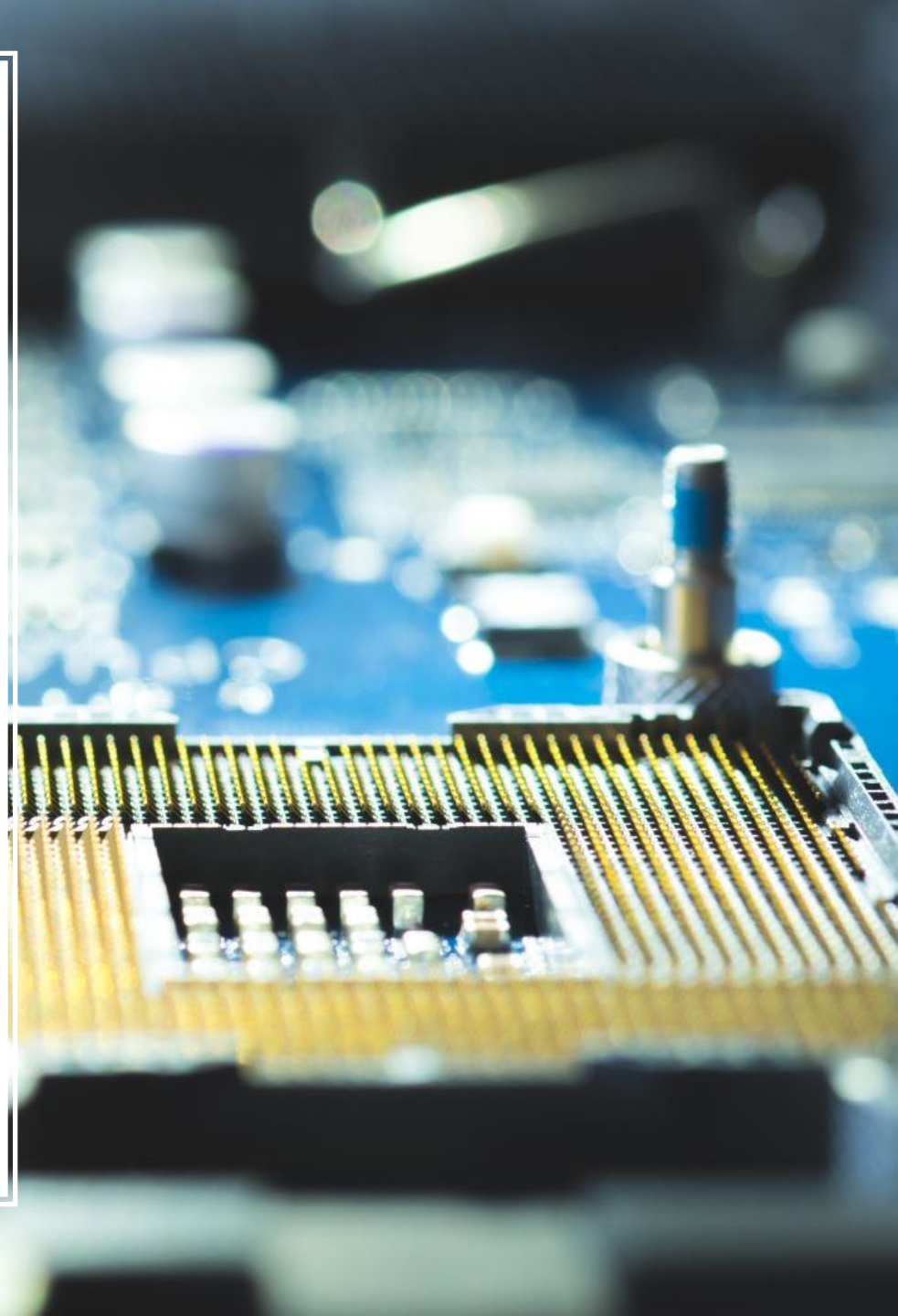
Extended Aims

In future, I aim to expand upon and develop these ideas further, further incorporating audio manipulation into the patch

Based on the success of this project I will review the possibility of building a portable version using PureData, and a Single-board computer such as Raspberry Pi.

Summary

- Lightweight, powerful software sequencer
- Control various parameters of external synths via MIDI CC
- Primarily used as a generative sequencer for the *Korg Minilogue*
- Modulation controls (Scale lock, probability, modulation range), variable level of sequence control
- Simple Interface with a degree of customisability
- Modularity to allow custom modifications and use with other hardware/software
- Combining technical synthesis and sequencing techniques with creative use of audio manipulation. This project comes in the form of a Max patch which can be run in Ableton Live



Thanks for listening

Please ask any questions you may have

1 https://www.korg.com/uk/products/synthesizers/minilogue/

2 https://maxforlive.com/library/device/3683/korg-minilogue-controller

3 https://library.vcvrack.com/Bogaudio/Bogaudio-Walk2

4 https://cycling74.com/forums/patch-optimisation-1

5 https://www.youtube.com/watch?v=FTkoIAeHEg4