

# Command Line Algorithmic Music System (CLAMS)

M. Edward (Ed) Borasky

2023-02-25

# Section 1

## Me and Forth

# Me - retired scientific applications programmer

- I got paid to write
  - ▶ Assembly
  - ▶ Fortran
  - ▶ Awk / sed / grep / \*nix shell
  - ▶ Perl
  - ▶ R

# I learned for hobby projects

- Lisp
- Forth
- Ruby

# I never learned

- APL
- C / C++ / Java / C# / Objective C / D / Rust / Go
- PHP
- Python
- JavaScript

# Forth

- 1980s
  - ▶ learned Forth via HESForth on Commodore 64
- mid-late 1990s
  - ▶ primary Forth engine was HP100LX Palmtop PC
  - ▶ wrote some articles for FORTH Dimensions
  - ▶ wrote some trading system software
  - ▶ used mostly hforth and Tom Almy's Forth compiler

# Why I stopped writing Forth

- by 1999 I had faster machines that ran my code in Perl
- I wasn't using it at work
- I was learning Linux and R at work

## Section 2

(Two sunspot cycles pass ...)



## Section 3

He's baaack! And he brought ... CLAMS!

# CLAMS goal

- algorithmic music composition and performance ...
- live ...
- in real time ...
- on a Raspberry Pi Pico!

# Inspirations

- Hierarchical Music Specification Language (HMSL)
  - ▶ long history and experience with experimental composers
  - ▶ currently being actively enhanced!
  - ▶ <https://github.com/philburk/hmsl.git>
- FORMULA - FORth MUsic LAnguage
  - ▶ designed for improvisation in real time
  - ▶ contained a real-time operating system
  - ▶ David P. Anderson and Kuivila (1989), D. P. Anderson and Kuivila (1991)
  - ▶ Python successor: Numula  
<https://github.com/davidpanderson/Numula/wiki>
- *Live Coding: A User's Manual* (Blackwell et al. 2022)
- The TOPLAP Manifesto
  - ▶ “Show Us Your Screens”
  - ▶ <https://toplap.org/wiki/ManifestoDraft>

# CLAMS architecture

- a domain-specific language implemented in Forth
- conceptually, ChuckK (<https://chuck.cs.princeton.edu/>), (Salazar et al. 2014) semantics with Forth syntax
  - ▶ low-level words: digital synthesis and microcontroller audio
  - ▶ mid-level words: construct signal flow graph
  - ▶ high-level words: provide live performance interface

## Section 4

Forth base: zeptoforth

# Highly optimized!

- subroutine-threaded
- allows inline expansion of words
- many primitives in assembly

# Close to the metal

- words for nearly all the RP2040 hardware
- has an RP2040 assembler
- can compile to RAM or flash
- real-time operating system capabilities

# zeptoforth on GitHub

- repository: <https://github.com/tabemann/zeptoforth>
- wiki: <https://github.com/tabemann/zeptoforth/wiki>



## Section 5

### Road map

## Proof of concept (v0.2.5) - Software

- direct digital synthesis / sine wave of any frequency
- maybe two oscillators and frequency modulation
- maybe “triangle” / “sawtooth” / “pulse” waves
- maybe a filter and envelope generator and low-frequency oscillator
- target date: 2023-03-17

# Proof of concept (v0.2.5) - Hardware

- Pimoroni Pico Audio Pack
  - ▶ ships from the UK
  - ▶ one in hand
  - ▶ <https://shop.pimoroni.com/products/pico-audio-pack>

# First release (v0.5.0) - Software

- will re-scope project after proof of concept!
  - ▶ need to assess audio performance constraints
- all synthesis / audio I/O words
- goal is all synthesis algorithms in Csound 7
  - ▶ <https://flossmanual.csound.com/>
- target date: 2023-05-12

## First release (v0.5.0) - Hardware

- Waveshare Audio Expansion Module for Raspberry Pi Pico
  - ▶ ships from China
  - ▶ on order
  - ▶ <https://www.waveshare.com/pico-audio.htm>

## Second release (v0.7.5)

- signal flow graph interpreter
- target date: 2023-06-16

## Full release (v0.9.0+)

- live performance user interface
- possibly port to other audio microcontrollers
- target date: 2023-07-14

# CLAMS on the web

- GitHub: <https://github.com/AlgoCompSynth/CLAMS>
- blog: <https://algocompsynth.github.io/CLAMS-Blog/>
- this presentation: <https://github.com/AlgoCompSynth/CLAMS/blob/main/presentations/CLAMS-intro.pdf>



# References

- Anderson, D. P., and R. Kiuivila. 1991. "Formula: a programming language for expressive computer music." *IEEE Computer* 24 (7): 12–21. <https://doi.org/10.1109/2.84829>.
- Anderson, David P., and Ron Kuivila. 1989. "Continuous Abstractions for Discrete Event Languages." *Computer Music Journal* 13 (3): 11–23. <http://www.jstor.org/stable/3680007>.
- Blackwell, A. F., E. Cocker, G. Cox, A. McLean, and T. Magnusson. 2022. *Live Coding: A User's Manual*. Software Studies. MIT Press.
- Salazar, S., A. Kapur, G. Wang, and P. Cook. 2014. *Programming for Musicians and Digital Artists: Creating Music with ChuckK*. Manning.