

The Book of CLAMS

M. Edward (Ed) Borasky

2023-11-03

Table of contents

Command Line Algorithmic Music System (CLAMS)	3
Overview	3
How does it work?	3
Why Forth?	3
What about Forth standard (Forth 200x Committee 2012) compatibility?	4
What about portability?	4
1 Introduction	5
2 Summary	6
References	7

Command Line Algorithmic Music System (CLAMS)

“I’ve never seen a happy clam. In fact, most of them were really steamed.” ~ M. Edward (Ed) Borasky

Overview

CLAMS is a text-based interactive environment for composing and performing music and visuals on a [Pimoroni PicoVision](#). It can be made to work on other boards using the RP2040 microcontroller, but you will need to buy additional hardware and port some code.

How does it work?

CLAMS is a domain-specific language built on a Forth compiler / interpreter. The user connects to the board via a serial connection and enters **CLAMS** / Forth code interactively.

Why Forth?

“Premature optimization may be the root of all evil, but it is damned fun!” ~ M. Edward (Ed) Borasky

1. Forth (Brodie 2022) is an extensible interactive operating system. It supports editing, assembling, compiling, debugging and running real-time tasks from a terminal.
2. Forth is efficient. A well-designed Forth will usually run a task at no worse than half the speed of a hand-optimized assembly version. **CLAMS** will have several optimizations built in for the ultimate speed.
3. Forth is lean. There are very few concepts to learn, there is minimal run-time overhead in RAM, and the whole package takes up much less flash space than MicroPython or CircuitPython.

What about Forth standard (Forth 200x Committee 2012) compatibility?

CLAMS is an extended subset of the standard. It won't contain all of the standard's core word set, and it will contain some extensions to support the Raspberry Pi Pico C/C++ SDK, RP2040 assembly language programming, the PicoVision hardware, cooperative multitasking, and high-speed digital signal processing.

What about portability?

Within the RP2040 ecosystem, as long as the PicoVision and C/C++ SDK work, porting should be straightforward, though tedious. And you will undoubtedly need to buy more hardware.

Outside of the RP2040 ecosystem, there are a number of other micro-controller music boards, most notably the [Electro-Smith Daisy](#) and the [Rebel Technology OWL](#) platforms. But they have their own SDKs, so there's not much need to port CLAMS to them.

There are also a number of audio projects that use the [Teensy® USB Development Board](#), which has a [comprehensive audio library](#). Like the first two, it has its own SDK. And the Daisy, OWL and Teensy processors are all more powerful than the RP2040.

By contrast, there's not much music-specific development software for the Raspberry Pi Pico / RP2040. There are some simple demos, a few do-it-yourself hardware offerings, and there's the [Allen Synthesis EuroPi](#), a Eurorack module with an open source MicroPython software platform. CLAMS will be a different approach.

The overall concept is an interactive language for making music on Raspberry Pi Pico / RP2040. I'm aiming for [Chuck](#) (Salazar et al. 2014) semantics with Forth syntax - a single text-based language to implement both the definitions of synthesized instruments and the sequences of sounds they make, intended for [live coding](#) / [algorave](#) performances.

This is a Quarto book.

To learn more about Quarto books visit <https://quarto.org/docs/books>.

The Book of CLAMS by M. Edward (Ed) Borasky is marked with CC0 1.0 Universal

1 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

2 Summary

In summary, this book has no content whatsoever.

$1 + 1$

[1] 2

References

- Brodie, Leo. 2022. “Starting Forth.” FORTH, Inc. <https://www.forth.com/starting-forth/0-starting-forth/>.
- Forth 200x Committee. 2012. “Forth 2012 Standard.” Forth 200x Committee. <https://forth-standard.org/standard/words>.
- Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. <https://doi.org/10.1093/comjnl/27.2.97>.
- Salazar, S., A. Kapur, G. Wang, and P. Cook. 2014. *Programming for Musicians and Digital Artists: Creating Music with ChucK*. Manning.