Overview

- An alternative arrayforth toolchain
- Gesture recognition on GA144
- Chlorophyll modifications
- Sensortag application
- Demo
Another Arrayforth Toolchain

- compiler, simulator, bootstreamer, and disassembler
- supports work with Chlorophyll, also useful by itself
- linux interoperability

Before:
- Compile Chlorophyll source
- Convert to colorforth format
- Open colorforth IDE
- Compile
- Manually set the boot descriptors
- Load
- Repeat until madness

Now:
- single command to compile and load
Compiler

- No semantic color: `: add10 0xa +`
- Address literals: `&wordname`
- Automatic `nop` insertion
- `north, east, south, west => up, down, left, right`
- Resolves forward word references
- `word@node` for compiling calls to words in other nodes
- Arrayforth Emacs mode
- Currently no support for generalized host computations during compilation
Bootstreamer

- Generates bootstreams for async(node 708) or 2wire (node 300)
- Supports streaming programs into host chip through target chip
- Supports most boot descriptors `/p /a /b /io`
- Currently no support for `/stack`
GA144 Simulator

- Supports most features
- Debugging
  - Unlimited breakpoints
  - Break on instruction word, function name, io pin change
  - Display current state, disassemble memory
- Currently no support for
  - Phantom wakeup signals
  - Parallel bus
  - Serdes
- Multi-chip simulation, can ‘wire’ them together
- No GUI (yet?)
- Demo?
Gesture recognition on GA144

Use hidden markov models to classify gestures

Stream accelerometer values to hmm models

Model training done on PC

- For each accelerometer measurement:
  - Filter - idle state, directional equivalence
  - Quantizer
  - Step forward procedure
Chlorophyll Evolution - Adding ‘actors’

Problem:

- high density code around IO nodes
- too much communication

Solution:

- Separate functions that communicate via port executable ‘messages’
- Inspired by actor model of concurrent computation

actor read_accelerometer@(8~>7);
Sensortag

Implemented part of the sensortag application in Chlorophyll

Based on Greenarrays AN012
Chlorophyll vs. Arrayforth
Crystal control

```plaintext
void osc(int@1 k)
    for (i from 0 to 5000) {
        set_io(715, SOURCE);
        delay_unext(715, k);
        set_io(715, SINK);
        delay_unext(715, k);
    }
    set_io(715, IMPED);
    for (i from 0 to 30000) {
        if (digital_read(715, 0)) {
            while (1) {
                set_io(715, SOURCE);
                set_io(715, HIGH_IMPEDE, WAKEUP_LOW);
                digital_wakeup(715);
                set_io(715, SINK);
                set_io(715, HIGH_IMPEDE, WAKEUP_HIGH);
                digital_wakeup(715);
            }
        }
    }
}

void crystal_init() {
    int@1 period;
    period = 12900;
    while (1) {
        osc(period >>@1 1);
        period = period +@1 1;
    }
}
```

```
node 715
: -osc over
  io b! for
  0x30000 lb dup .. 2/ dup for unext
  0x20000 lb .. over 1 and .. + for unext next
dup or lb dup 30000 for
  drop @b - -while next ;
  then dup or pop drop ;
: clang
  12470 2000 for dup 5000 -osc while
  drop 1 . + next clang ; then
: prep
  0 0x20000 0x800 0x30800 0 0x20000 0x800 0x30800
dup up a! drop
: run !b !b @ drop run ;
: main south a! clang ;
```
https://github.com/mangpo/chlorophyll

https://github.com/mschuldt/ga144