﻿Goal: get a kilobot simulator working in a web browser for which we can write normal c code (that would normally compile down into a hex) and have each robot in the swarm run that code.

Using a web browser suggests using javaScript.

We will use Emscripten which is a way to compile C code so that it can be called by JavaScript

Emscripten:

https://emscripten.org/docs/introducing\_emscripten/about\_emscripten.html

more info:

https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Emscripten

------------------------------------ Windows-------------------------------------

Installing Emscripten (on ubuntu):

https://emscripten.org/docs/getting\_started/downloads.html

--- install ubuntu prerequisites ----

it looks like we need to install the following prerequisites

(the first 2 were already installed for me):

x# Install Python (2.7 or later)

# Install git

https://git-scm.com/downleoads

--- install the Emscripten SDK ----

#Open command prompt

Windows + R --> type in "cmd" ---> Press Enter

#Go to the directory where you want to install this stuff, I did this in the directory containing this readme

cd [where you want]

# Get the emsdk repo

# Enter that directory

cd emsdk

# Fetch the latest version of the emsdk (not needed the first time you clone)

git pull

# Download and install the latest SDK tools.

emsdk install latest

# Make the "latest" SDK "active" for the current user. (writes ~/.emscripten file)

emsdk activate latest

# Activate PATH and other environment variables in the current terminal

Emsdk\_env.batcd

### NOTES git pull will fetch the current list of tags, but very recent ones may not yet be present there.

### You can run emsdk update-tags to update the list of tags directly.

###

### If you change the location of the SDK (e.g. take it to another computer on an USB),

### re-run the emsdk activate latest and emsdk\_env.bat commands.

---- updating the sdk (if some time has passed since the install ----

# Fetch the latest registry of available tools.

emsdk update

# Download and install the latest SDK tools.

emsdk install latest

# Set up the compiler configuration to point to the "latest" SDK.

emsdk activate latest

# Activate PATH and other environment variables in the current terminal

emsdk\_env.bat

#NOTE: If you get the error:

'emcc' is not recognized as an internal or external command, operable program or batch file.

Run the emsdk\_env.bat command from emsdk directory.

----- Hello World Example -------

# note, we'll need to run

emsdk\_env.bat

#from the emsdk directory every time we boot up, or add this to our bashrc script

# If you haven’t run Emscripten before, run it now with:

emcc -v

## put the following into hello\_world.c

## -------------------->

/\*

\* Copyright 2011 The Emscripten Authors. All rights reserved.

\* Emscripten is available under two separate licenses, the MIT license and the

\* University of Illinois/NCSA Open Source License. Both these licenses can be

\* found in the LICENSE file.

\*/

#include <stdio.h>

int main() {

printf("hello, world!\n");

return 0;

}

## <-------------------

## go to the directory where hello\_world.c it exits

cd [wherever]

# To build the JavaScript version of this code, simply specify the C/C++ file after emcc (use em++ to force compilation as C++):

emcc hello\_world.c

### You should see two files generated by that command: a.out.js and a.out.wasm. The second is a WebAssembly file containing the compiled code, and the first is a JavaScript file containing the runtime support to load and execute it. You can run them using node.js:

node a.out.js

# Generating HTML

# Emscripten can also generate HTML for testing embedded JavaScript. To generate HTML, use the -o (output) command and specify an html file as the target file:

emcc hello\_world.c -o hello.html

# start a local webserver in the directory (Python 3 does this as below, which worked for me):

python -m http.server

# depending on your configuration, you may need to use python2 or python3 instead of python for the command above.

# in a browser:

http://localhost:8000/hello.html

# ----------------------- quick start for working on code ----------------------------

# start python webserver (in one terminal, make sure to do it from the "code for kilobot simulator" directory):

python -m http.server 8080

# compile code (in another terminal, from the emsdk directory)

emsdk\_env.bat

Use source ./emsdk\_env.sh if using ubuntu

#start webpage (make sure to do it from the "code for kilobot simulator" directory where main.cpp is)v

emcc main.cpp -std=c++11 -s USE\_PTHREADS=1 -s WASM=1 -s USE\_SDL=2 -O3 -o index.js --embed-file light\_patterns

# in a google chrome browser:

http://localhost:8090/index.html