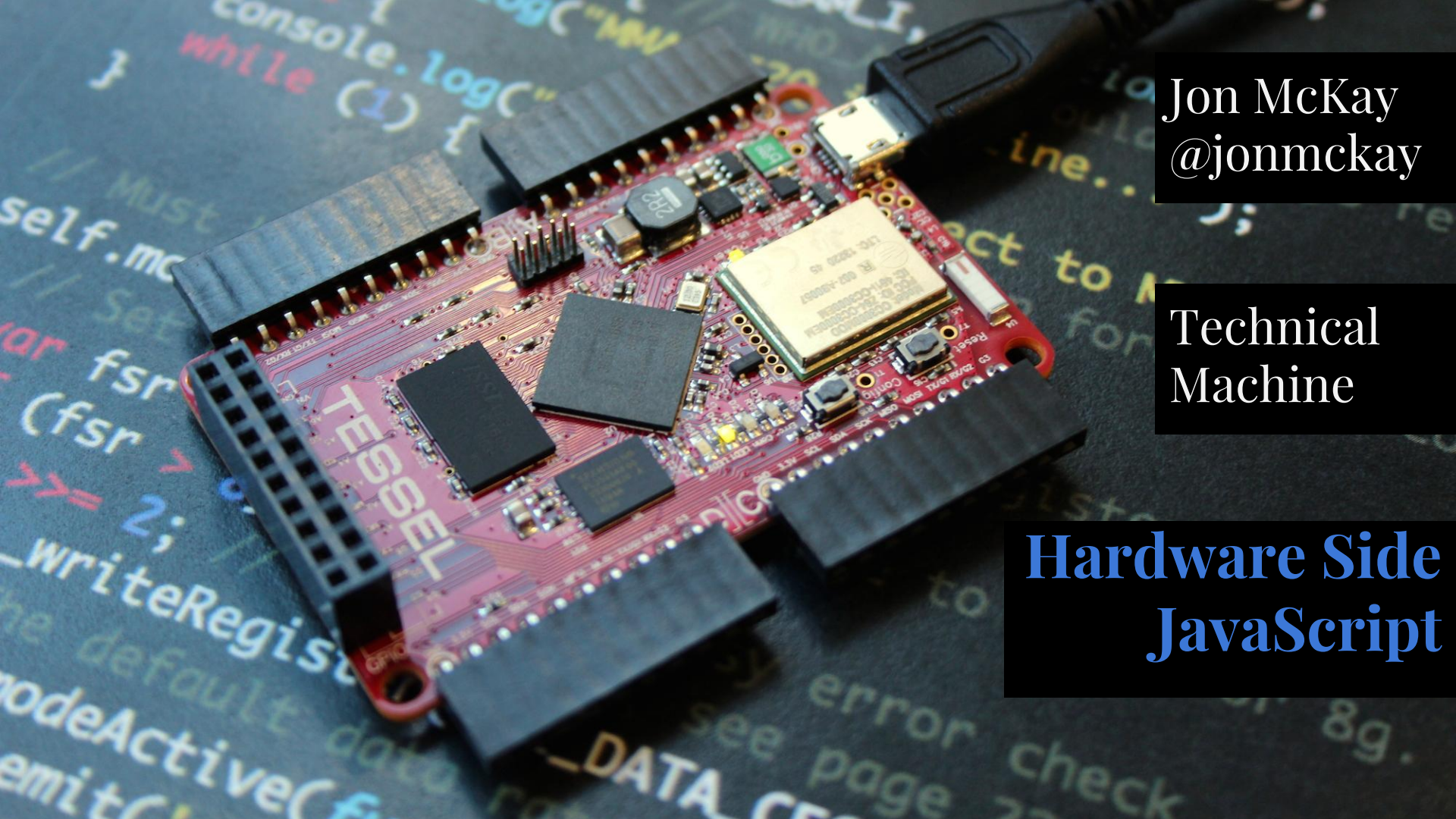


Jon McKay
@jonmckay

Technical
Machine

Hardware Side JavaScript

“Forgive me for not being hip but why try so hard to put JS in new places?”

- @farnsworth (HackerNews user)

Source: https://news.ycombinator.com/item?id=6214838

A: Flourishing Ecosystem

	Rank			# New Repos Created		
Language	2014	2013	2012	2014	2013	2012
JavaScript	1	1	2	383185	320534	277875
Java	2	3	3	283354	185530	240992
Ruby	3	2	1	259268	228145	310281
C	4	7	4	178891	79223	203992
CSS	5	12	25	175573	18869	3791
PHP	6	4	6	175476	139591	157185
Python	7	5	5	151669	126027	165655
C++	8	6	7	78878	104499	88615

Source: http://adambard.com/blog/top-github-languages-2014/

A: Prolific Community

Source: http://npmjs.org

A: Accessibility

```
// turn on an LED
scu_pinmux(g_APinDescription[ulPin].port,
           g_APinDescription[ulPin].pin,
           PIN_MODE | pin_modes[ulPin],
           g_APinDescription[ulPin].func);

GPIO_SetDir(g_APinDescription[ulPin].portNum,
      1 << (g_APinDescription[ulPin].bitNum),
      GPIO_INPUT);

GPIO_SetValue(g_APinDescription[ulPin].
portNum, 1 << g_APinDescription[ulPin].
bitNum);
```

```
// turn on an LED
var tessel = require('tessel');
tessel.led[0].high();
```

A: Event Based

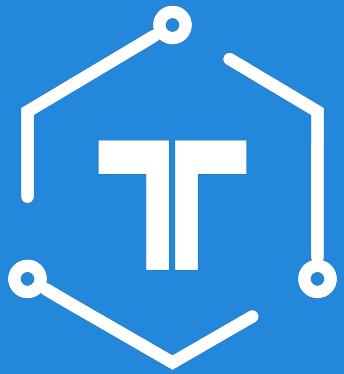
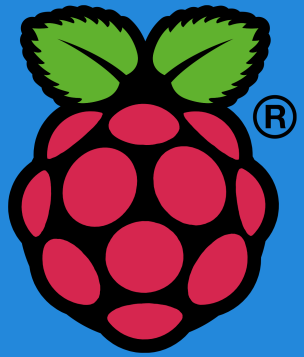
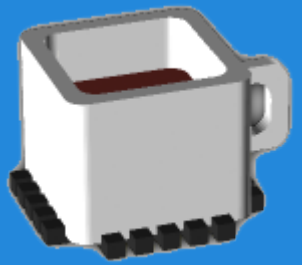
```
var tessel = require('tessel');
var led = tessel.led[0];
var button = tessel.button;

// Sleep until the button is pressed
button.on('press', led.toggle.bind(led));
```

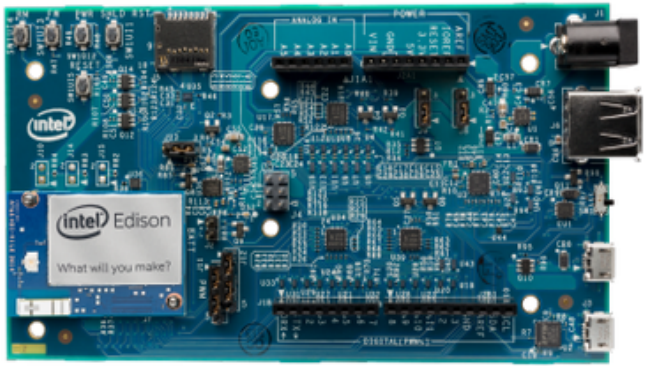
“In my view, the popularity of JavaScript comes down to two things: **Interactivity** and **Instant Gratification**”

- @naunga (HackerNews user)

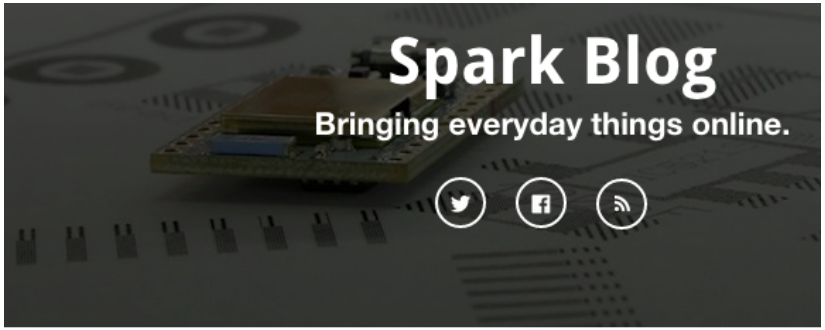
Source: https://news.ycombinator.com/item?id=6217993

JavaScript on hardware is ideal
for **prototyping new ideas.**



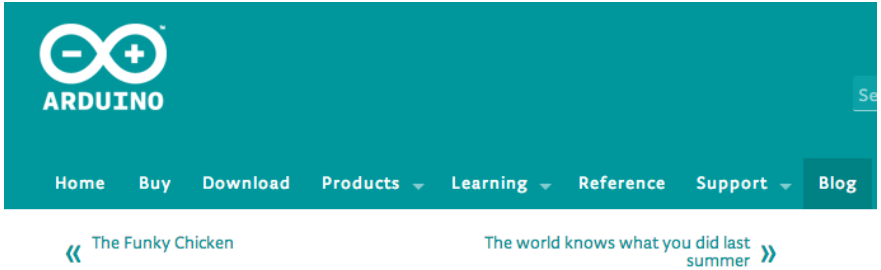
- The Intel Edison module will initially support development with Arduino* and C/C++, followed by **Node.JS**, Python, RTOS, and Visual Programming support in the near future.



Blink an LED with Javascript

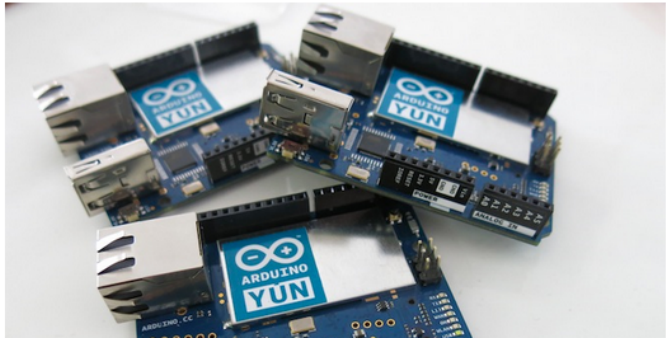
Use SparkJS to easily connect to your Spark Core.

WRITTEN BY CHRISTINE SUNU — SEPTEMBER 29, 2014



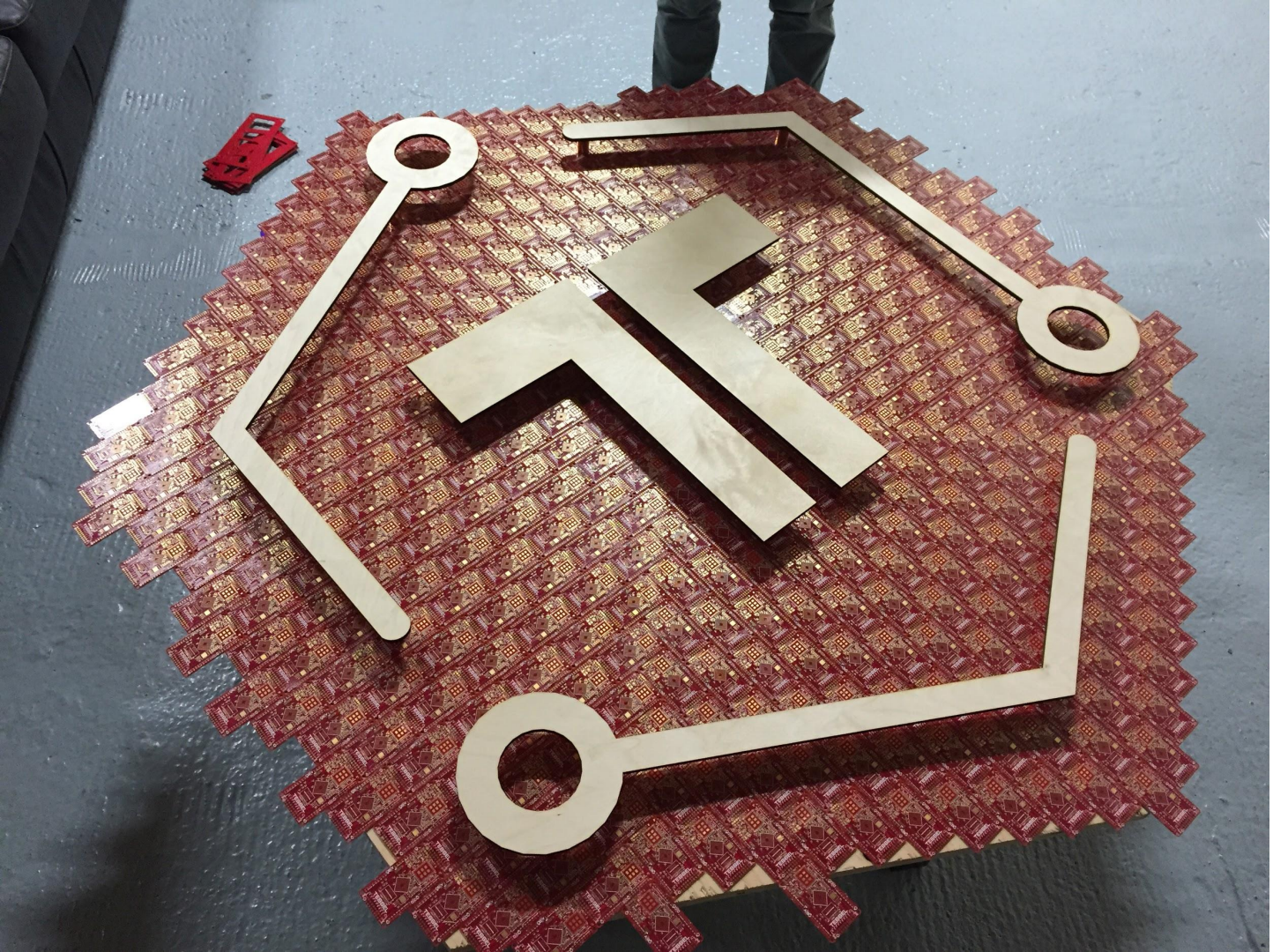
NODE.JS ON THE ARDUINO YÚN VIA THE BRIDGE LIBRARY

Zoe Romano — July 7th, 2014



Live Demo Time!

- Blinky
- HTTP
- Modules



Case Study:
Tessel Sign

Parts:

- Tessel PCBs
- Laser Cut Logo
- Neopixels
- Tessel


7:27 PM

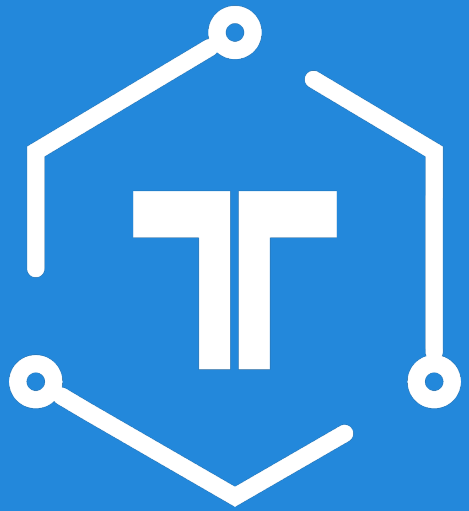
app.js

app.js

```
// Import a lightweight HTTP router
var router = require('tiny-router'),
  // Import the neopixel library
  Neopixels = require('neopixels'),
  // initialize the library
  neopixels = new Neopixels(),
  // set the number of leds
  numLEDs = 52,
  // set the port
  port = 8080;

// Set a route for the router to accept along with a parameter
router.get('/brightness/{intensity}', function(req, res) {
  // pull out the paramer
  var brightness = parseFloat(req.body.intensity);
  // multiply it by the max brightness
  var val = 0xff * brightness;
  // create a new buffer
  var buf = new Buffer(numLEDs * 3);
  // fill every rgb values with the brightness
  buf.fill(val);
  // send the animation to the LEDs
  neopixels.animate(numLEDs, buf);
  // send a response
  res.send("Brightness set to " + brightness);
});
// start listening on a port
router.listen(port);
console.log('listening on port', port)

```

Let’s build something.

Questions?



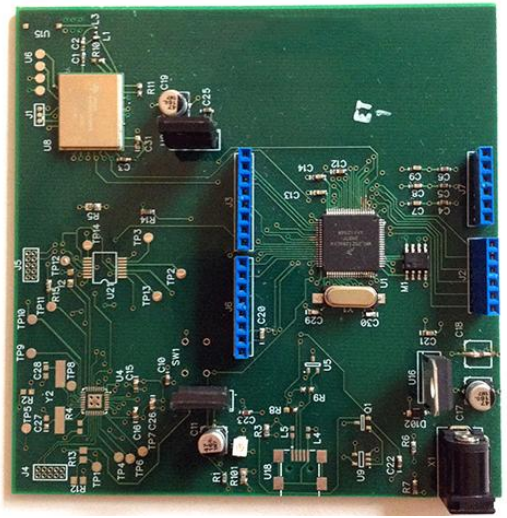
http://tessel.io

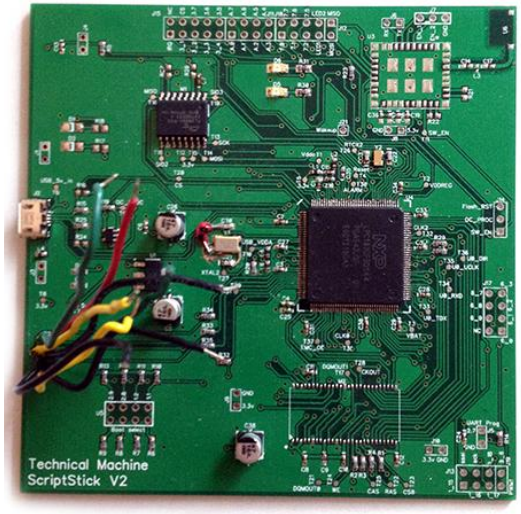
jon@technical.io

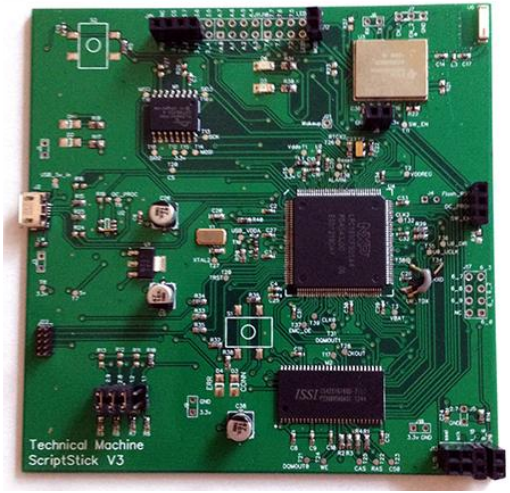
@technicalhumans

Appendix

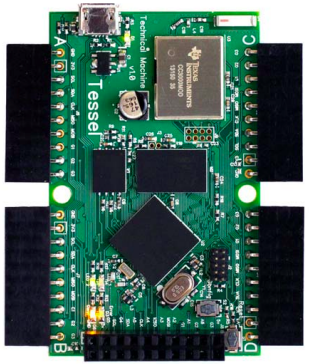
How does Tessel Work?

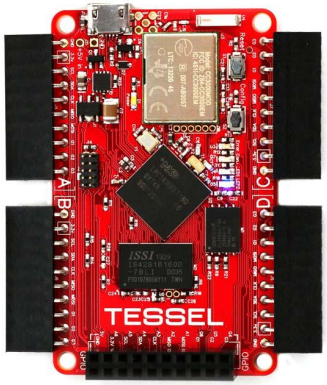


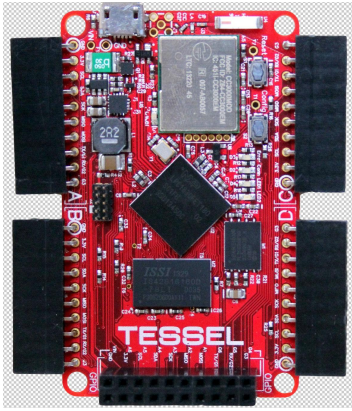

Technical Machine
ScriptStick V2

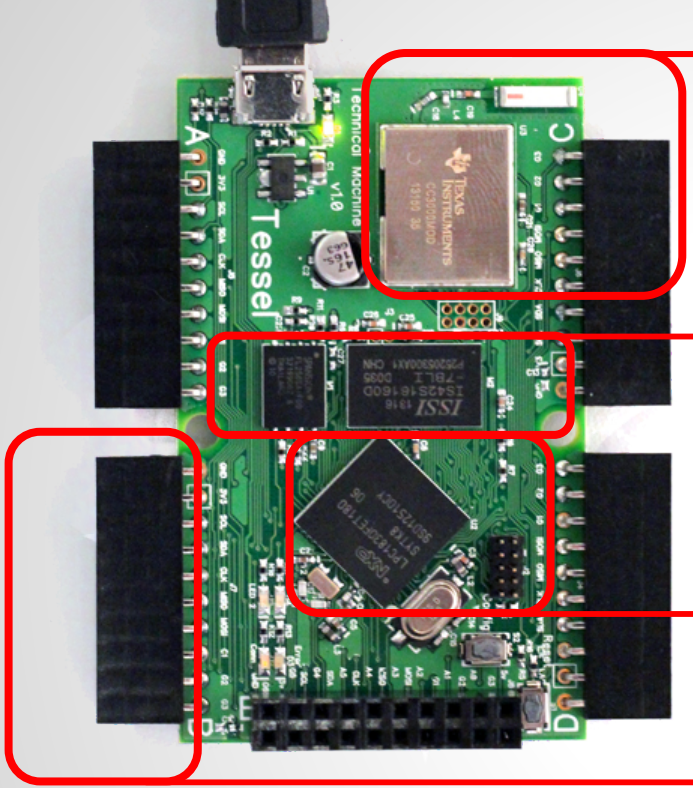

Technical Machine
ScriptStick V3


ScriptStick


Tessel
A
B
C
D


TESSEL
A
B
C
D


TESSEL
A
B
C
D



CC3000 WiFi Chip

Constant Connection
Remote Deployment
Mobile Friendly

32MB of Flash & RAM

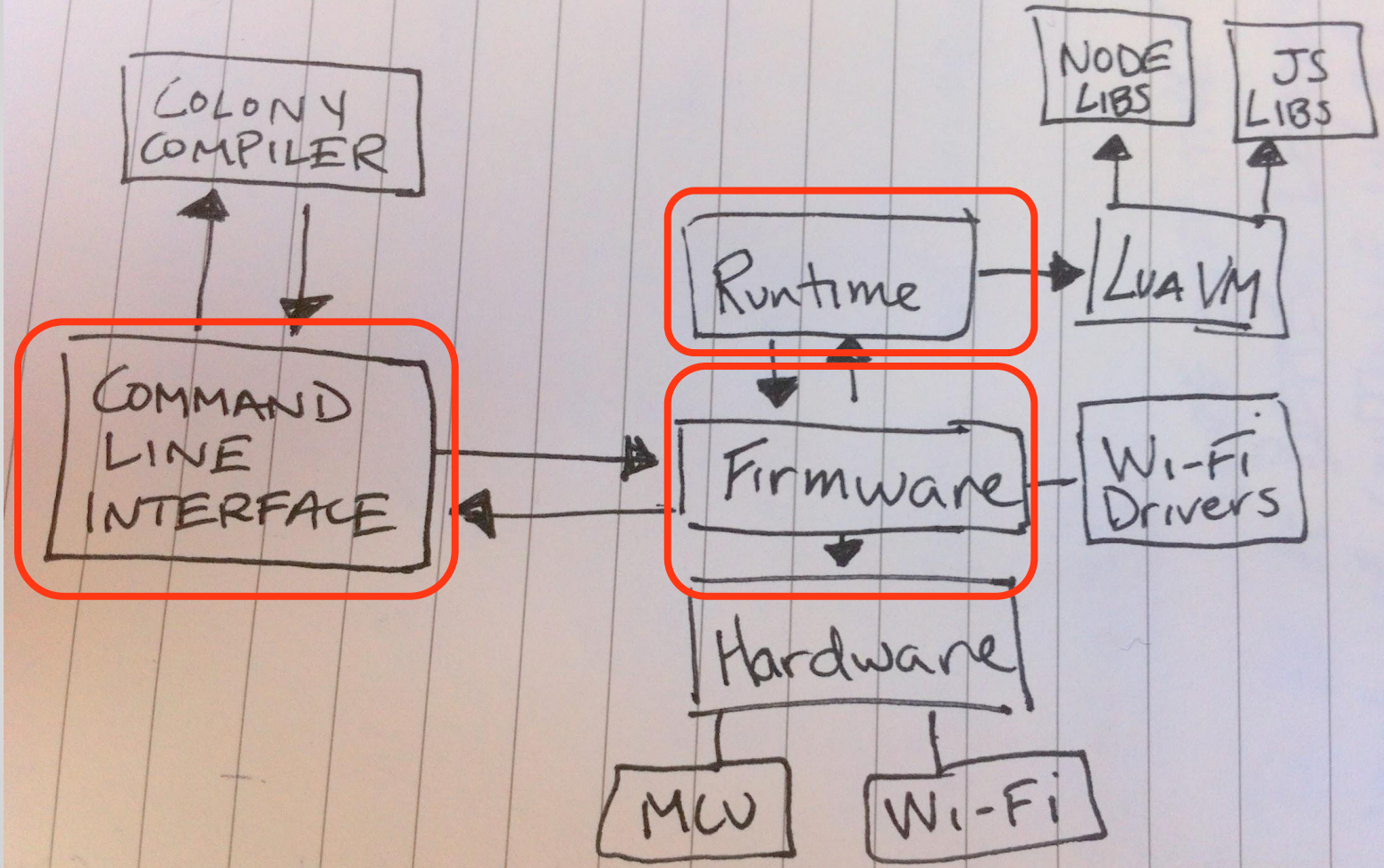
JavaScript-powered
Node.js compatible

Cortex M Processor

Computation

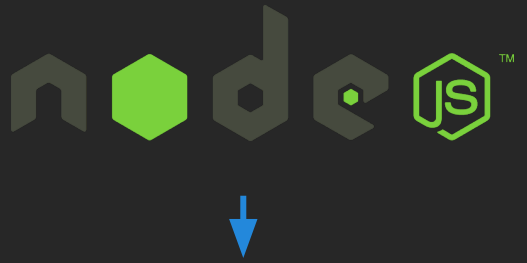
Horizontal Module Header

Plug n Play Capabilities

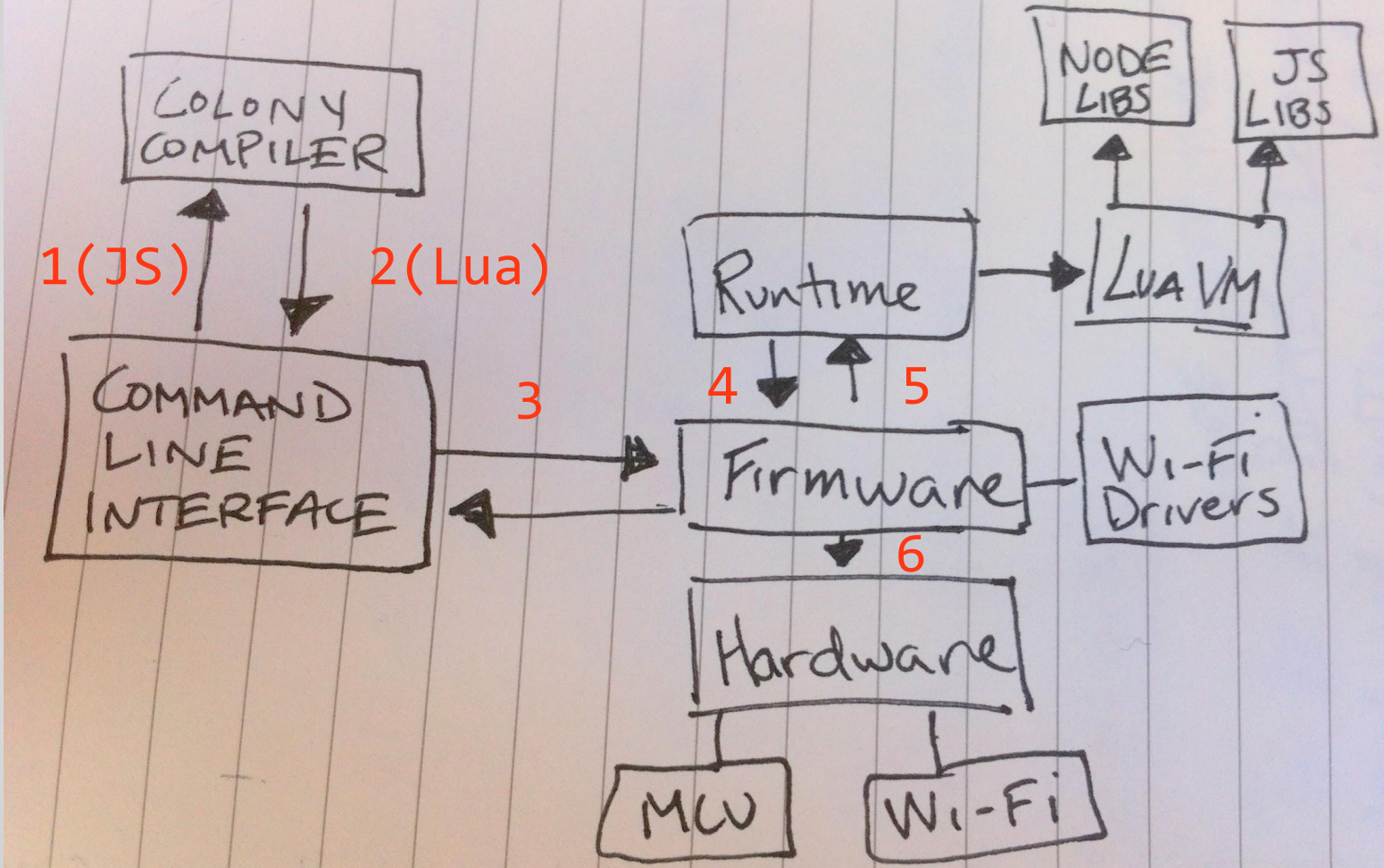

COLONY COMPILER
NODE LIBS
JS LIBS
Runtime
LUA VM
COMMAND LINE INTERFACE
Firmware
Wi-Fi Drivers
Hardware
MCU
Wi-Fi

```
var tessel = require('tessel');

var led1 = tessel.led[1].writeSync('high')
var led2 = tessel.led[2].writeSync('low')

var i = 0;
setInterval(function () {
  console.log('Blinked', i++, 'times');
  led1.toggleSync();
  led2.toggleSync();
}, 100);
```

```js
var tessel = require('tessel');

var led1 = tessel.led[1].writeSync('high')
var led2 = tessel.led[2].writeSync('low')

var i = 0;
setInterval(function () {
  console.log('Blinked', i++, 'times');
  led1.toggleSync();
  led2.toggleSync();
}, 100);
```




```lua
return function (_ENV, _module)
local exports, module = _module.exports, _module;

local tessel, led1, led2, i = tessel, led1, led2, i;
tessel = require(this, ("tessel"));
led1 = tessel:led((1)):output():high();
led2 = tessel:led((2)):output():low();
i = (0);
setInterval(this, (function (this)
console:log(("Blinked"), (function () local _r = i; i = _r + 1; return _r; end)(), ("times"));
led1:toggle();
led2:toggle();
end), (100));

return _module.exports;
end
```


COLONY COMPILER
1(JS)
2(Lua)
COMMAND LINE INTERFACE
3
Runtime
4
5
Firmware
6
Hardware
MCU
Wi-Fi
LUA VM
NODE LIBS
JS LIBS
Wi-Fi Drivers

```
var tessel = require('tessel');
tessel.leds[0].rawWrite(1);
```

User Land

```
Pin.prototype.rawWrite = function rawWrite(value) {
  hw.digital_write(this.pin, value ? hw.HIGH : hw.LOW)
  return this;
};
```

Driver

```
LUALIB_API int luaopen_hw(lua_State* L)
{
  luaL_reg regs[] = {

    { "digital_write", l_hw_digital_write },
```

Lua Binding declaration

```
LUALIB_API int luaopen_hw(lua_State* L)
{
  luaL_reg regs[] = {

    { "digital_write", l_hw_digital_write },
```

Lua Binding declaration

```
static int l_hw_digital_write(lua_State* L)
{
  uint32_t pin = (uint32_t)lua_tonumber(L, ARG1);
  uint32_t level = (uint32_t)lua_tonumber(L, ARG1 + 1);

  hw_digital_write(pin, level);

  return 0;
}
```

Lua-C bindings

```
void hw_digital_write (size_t ulPin, uint8_t ulVal)
{
  if (ulVal != HW_LOW) {
    GPIO_SetValue(g_APinDescription[ulPin].portNum, 1 << g_APinDescription[ulPin].bitNum);
  } else {
    GPIO_ClearValue(g_APinDescription[ulPin].portNum, 1 << (g_APinDescription[ulPin].bitNum)
  }
}
```

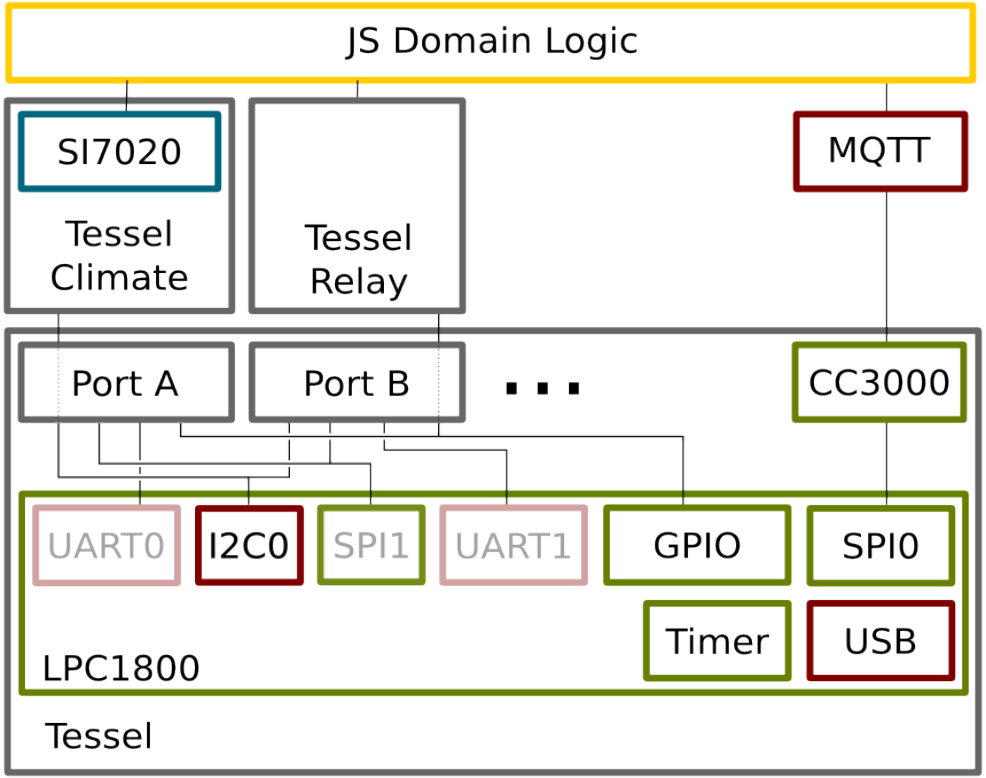
Firmware

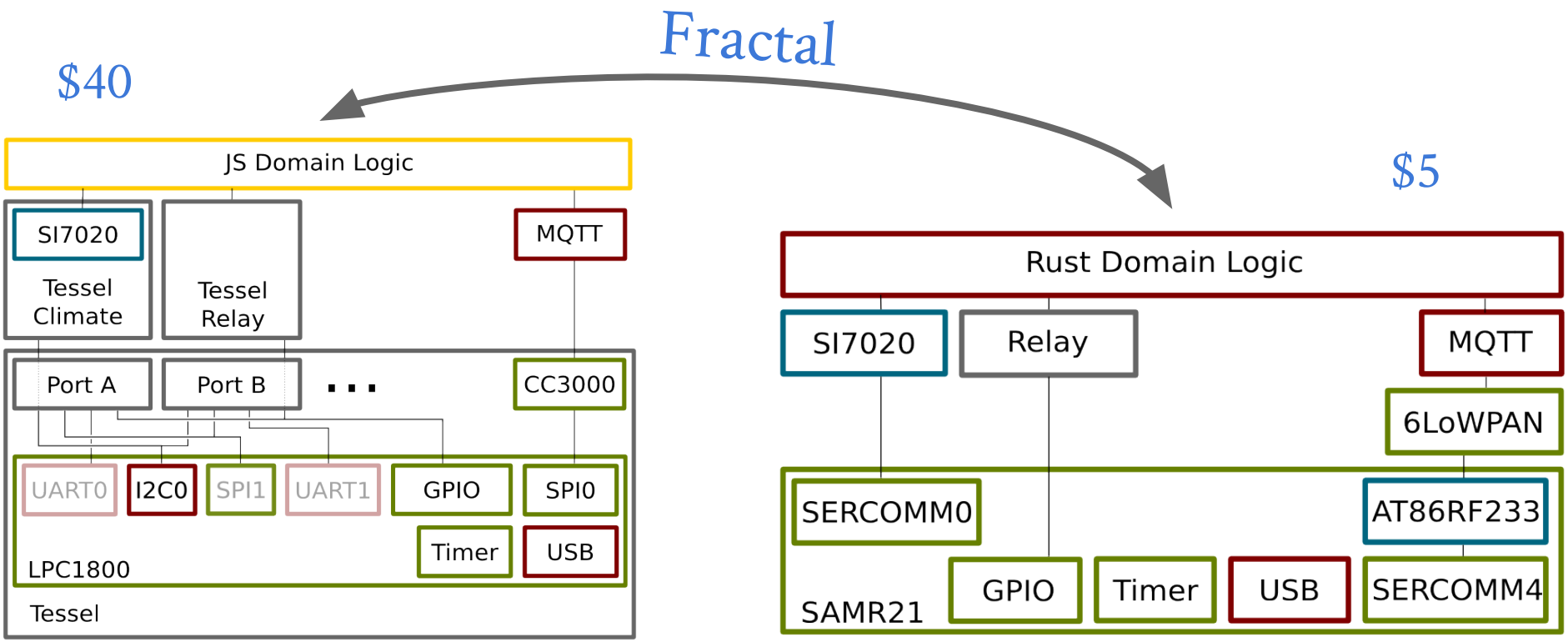
Why not use V8?

- Chrome’s JS engine
- C++
- ~10mb memory required for each instance
- POSIX/Win32 environment

- Embeddable language
- Written in C
- ~30kb memory
- Highly portable
- LuaJIT is 10x-100x faster than LuaVM

What about Post-Prototyping?

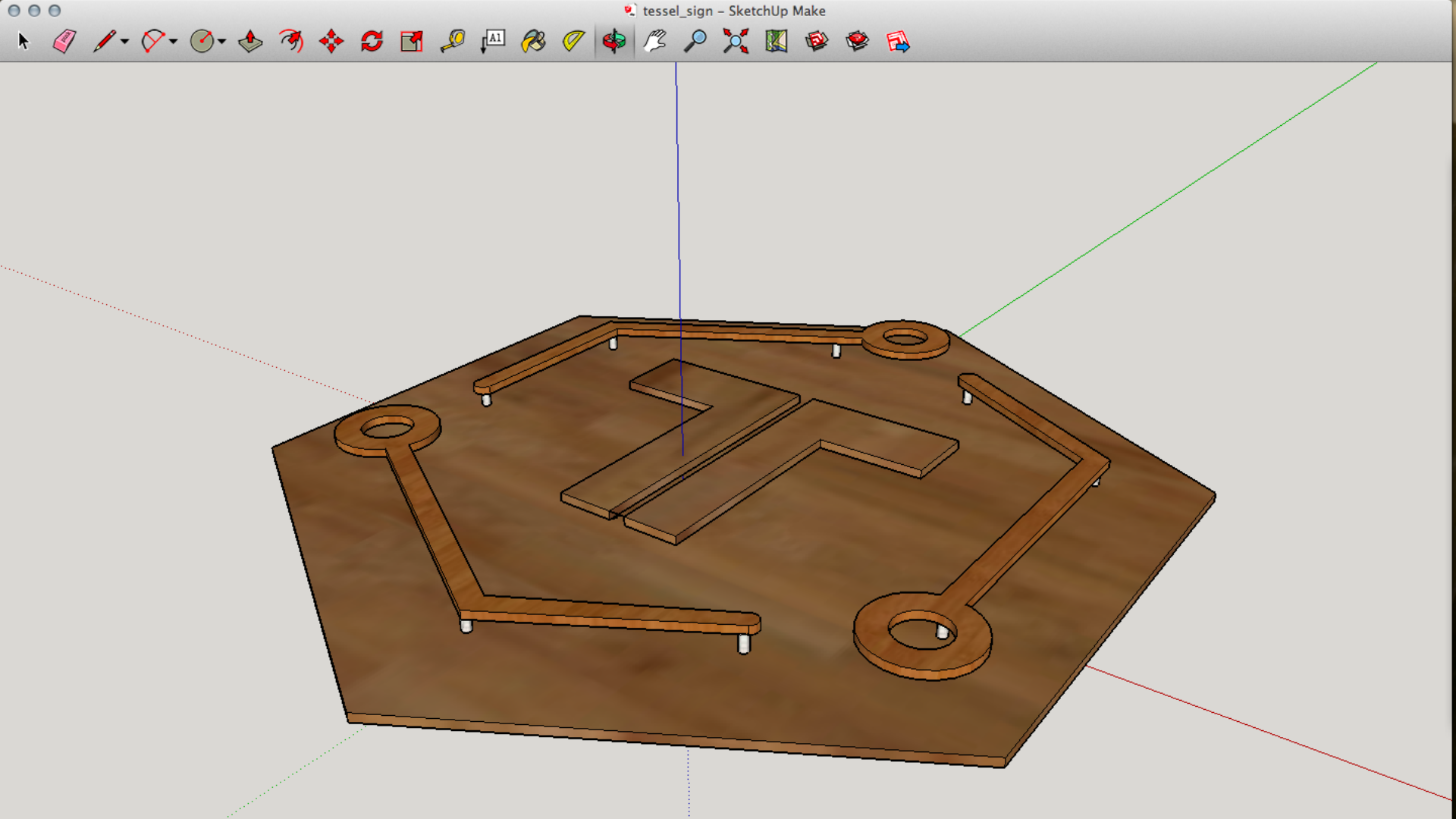

JS Domain Logic
SI7020
Tessel
Climate
Tessel
Relay
MQTT
Port A
Port B
. . .
CC3000
UART0
I2C0
SPI1
UART1
GPIO
SPI0
Timer
USB
LPC1800
Tessel

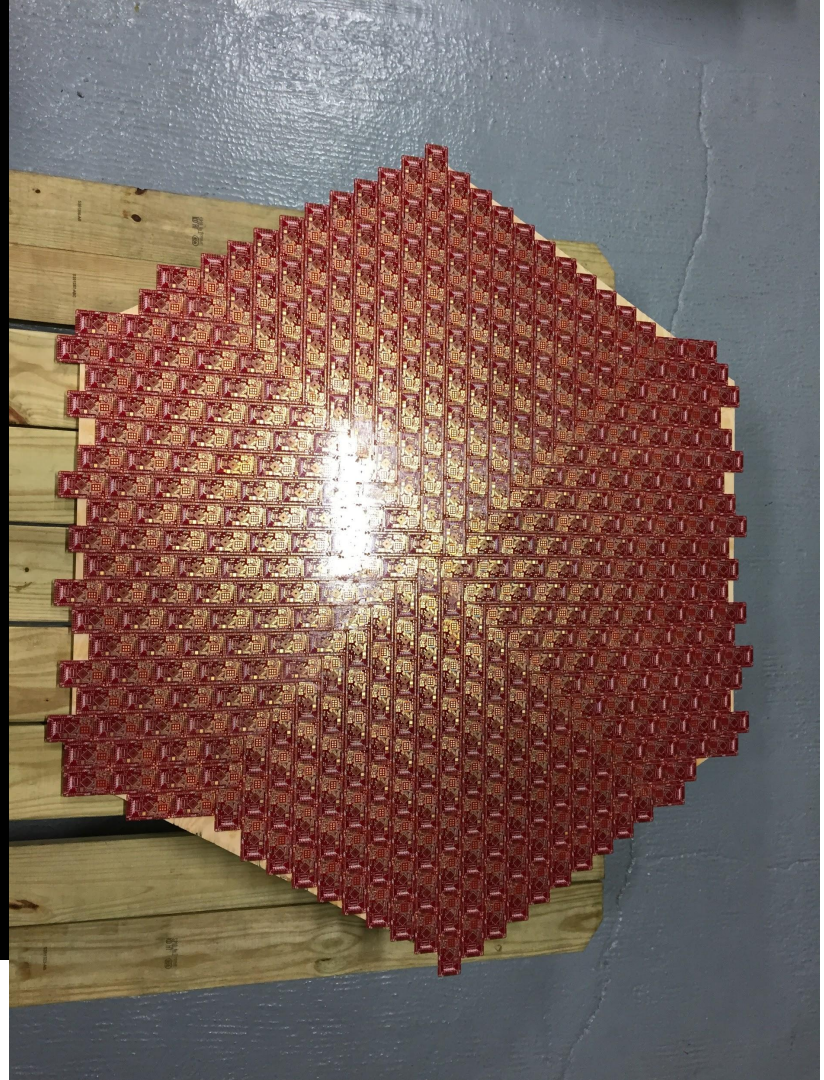

Fractal
$40
JS Domain Logic
SI7020
Tessel Climate
Tessel Relay
MQTT
Port A
Port B
. . .
CC3000
UART0
I2C0
SPI1
UART1
GPIO
SPI0
Timer
USB
LPC1800
Tessel
$5
Rust Domain Logic
SI7020
Relay
MQTT
6LoWPAN
SERCOMM0
AT86RF233
GPIO
Timer
USB
SERCOMM4
SAMR21

Read More About Fractal:

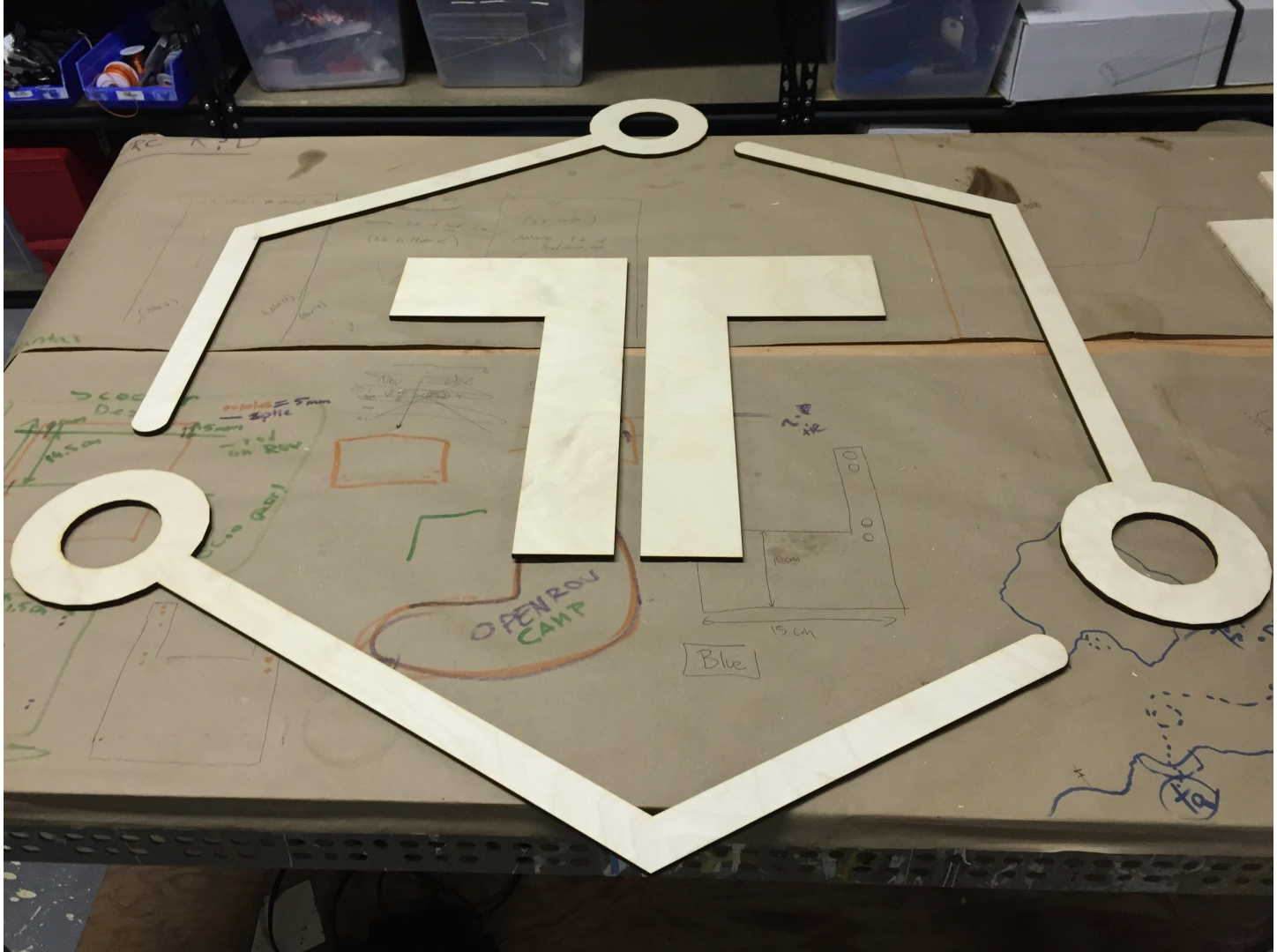
https://github.com/technicalmachine/fractal-docs

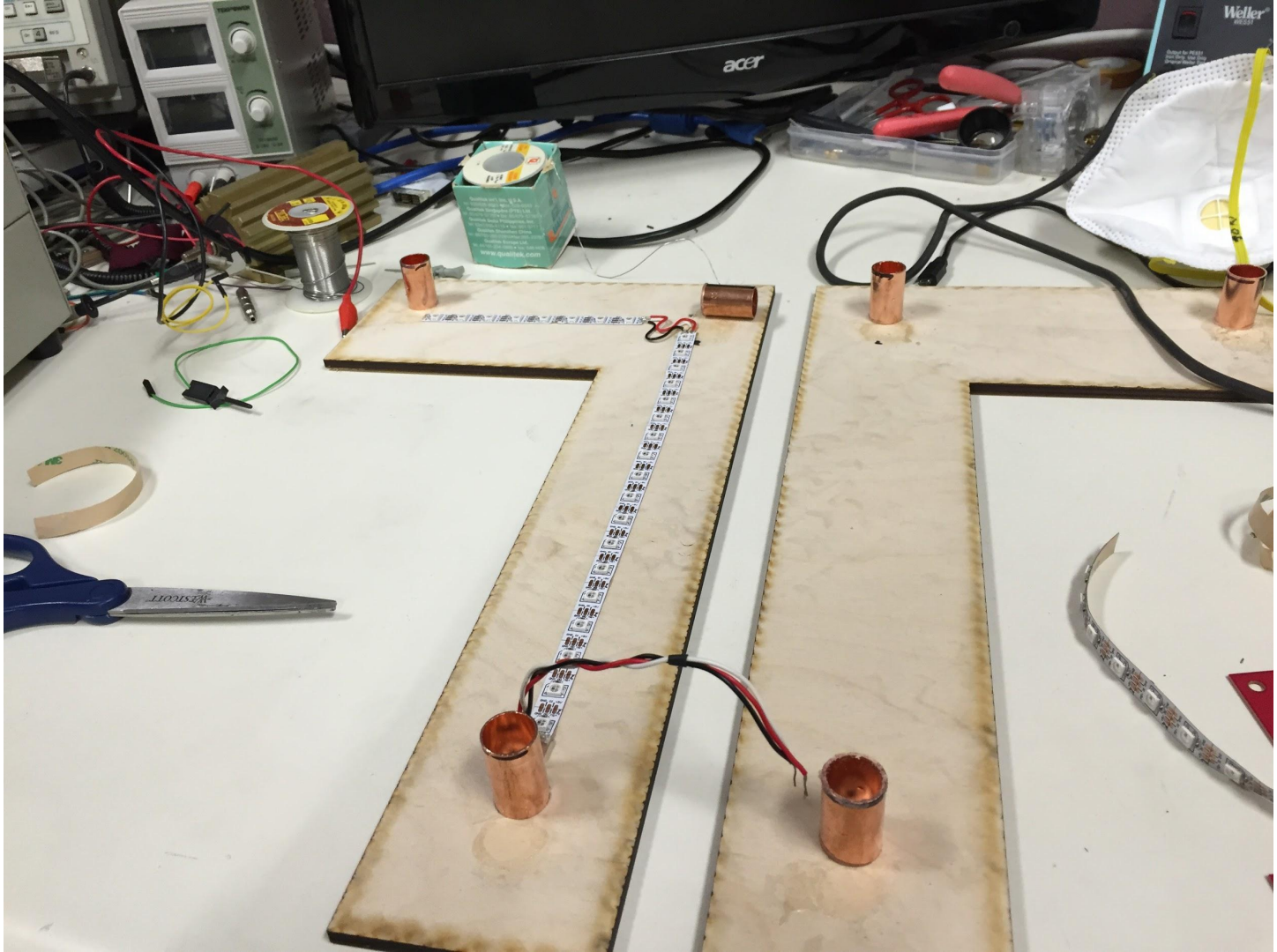
Creation of Tessel Sign


tessel_sign – SketchUp Make




DANGER – Laser radiation when open. AVOID
EYE OR SKIN EXPOSURE TO DIRECT OR
SCATTERED RADIATION.


OPEN ROW
CAMP
15 cm
Blue


acer
Weller