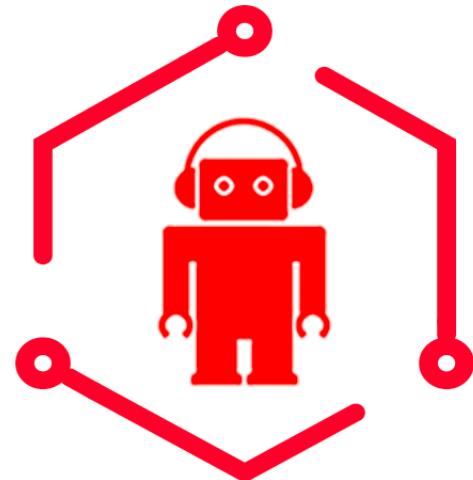


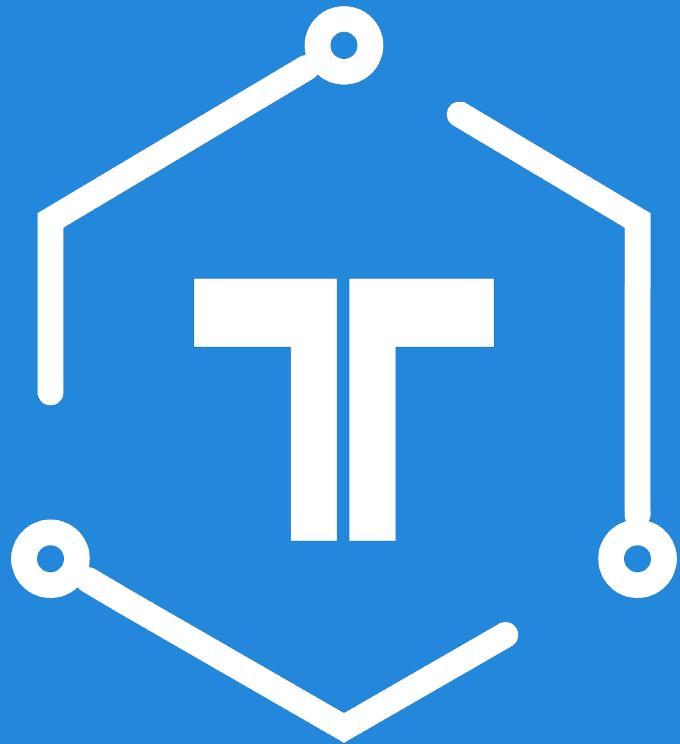
I work for Technical Machine, a startup in Berkeley focusing on making hardware development as accessible as software development is today. We manufacture Tessel, a business card-sized microcontroller that runs JavaScript, has Wifi capability, and can be programmed from the command line. We sell modules you can plug in like servos, climate/light/sound sensors, RFID card readers, and install code for them as easily as installing a Node package.

I'm overdue to give a technical talk about our implementation, but I'd love to present on any of these things:

- Forming a hardware startup with primarily CS knowledge
- How "hardware" may be the next platform after web and mobile
- How Tessel (and microcontrollers) work, in particular, my work on getting JavaScript on a board with limited memory
- How to buy a sensor or chip and talk to it in just 20 minutes
- How Open Source Hardware will catch up to Open Source Software

# GIANT ROBOTS CODED IN JAVASCRIPT





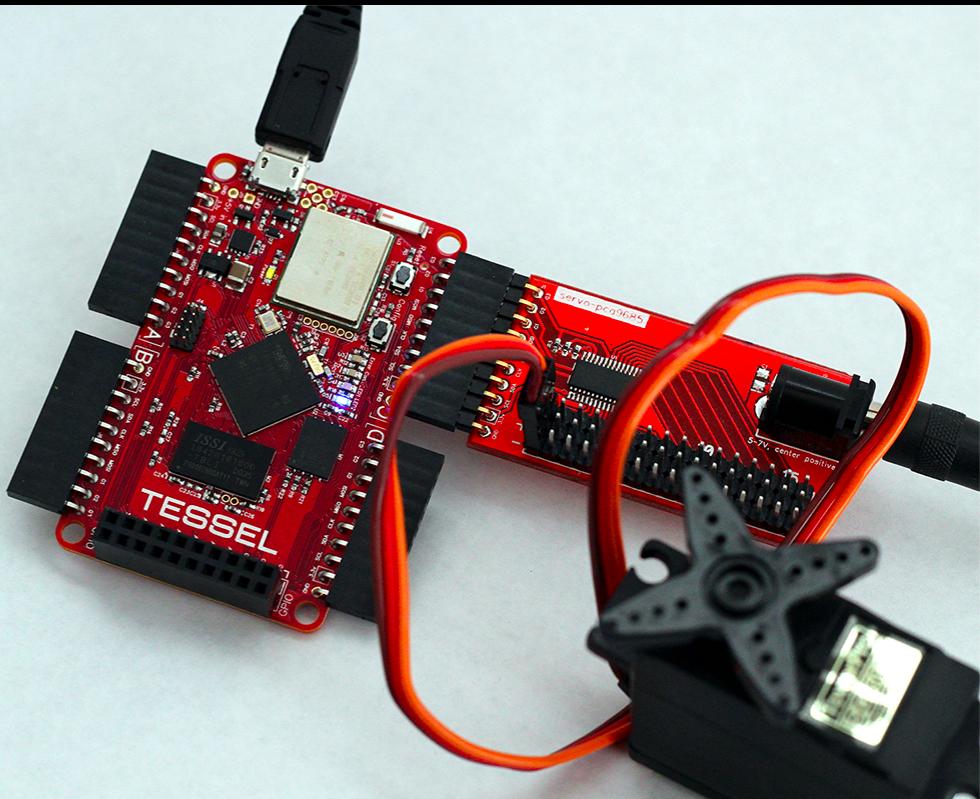
**Tessel** is a WiFi-enabled  
microcontroller that runs  
JavaScript.

Tim Ryan, Co-Founder / Software Developer

Technical Machine

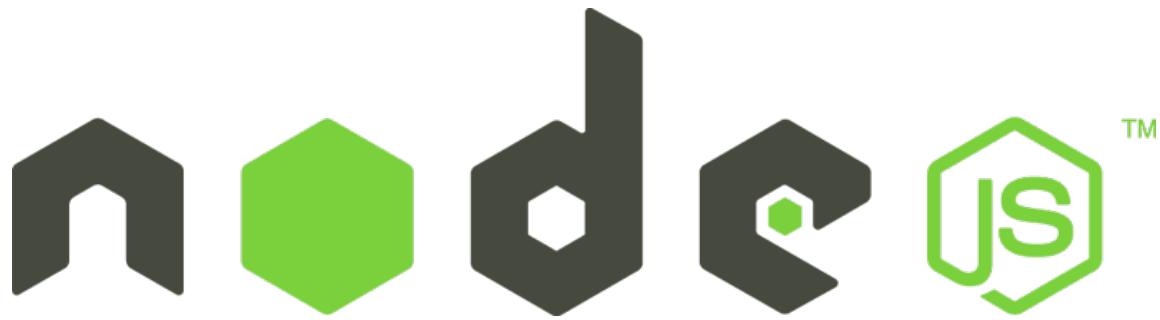
```
var tessel = require('tessel');
var servos = require('servo-pca9685')
  .connect(tessel.port('A'));

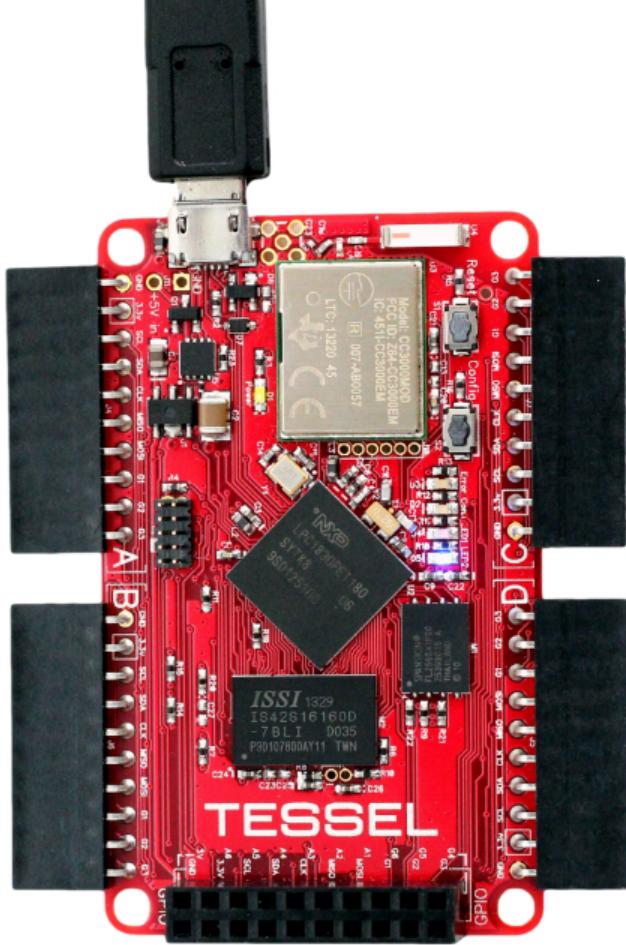
var degrees = 0;
setInterval(function () {
  servos.moveServo(1, degrees);
  degrees = degrees == 0 ? 180 : 0;
}, 500);
```



The skills you have in 2014 are all  
you need to make physical devices!

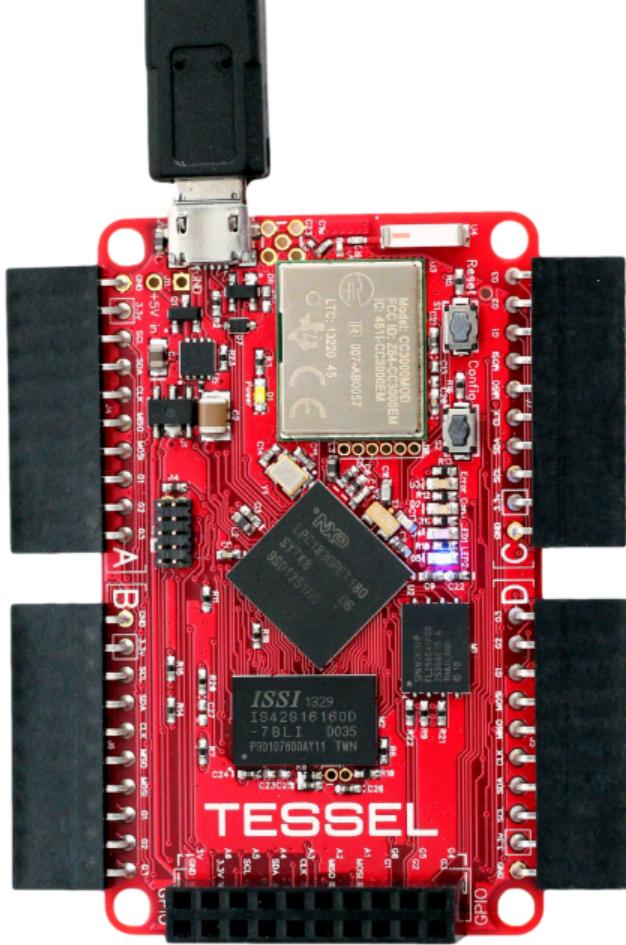
# I. Playing with Hardware





Tessel is a microcontroller that runs JavaScript.

- 4 ports for “modules”
- 1 port for breadboarding
- Built-in WiFi
- Program over USB or WiFi
- Power over USB, or  
Up to 12V battery
- 4 LEDs, one user button



```
$ npm install -g tessel
```

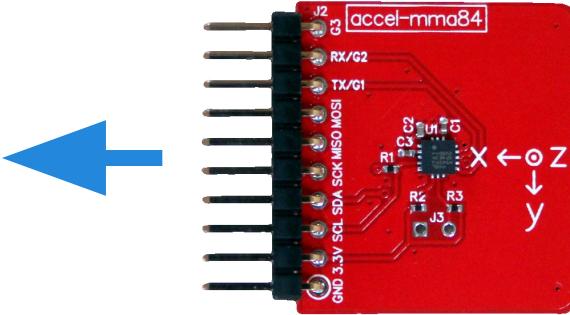
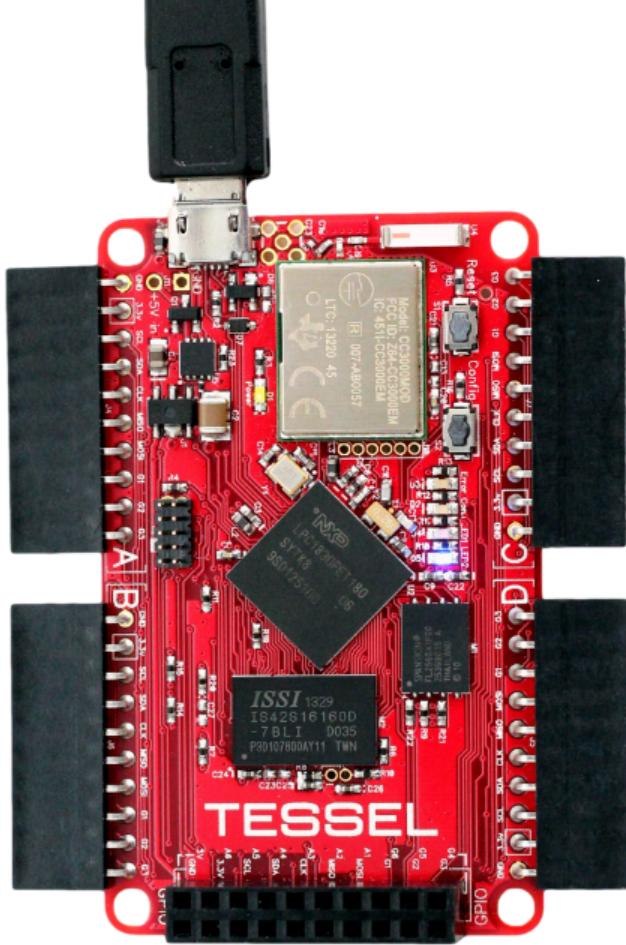
```
...
```

```
$ tessel blink
```

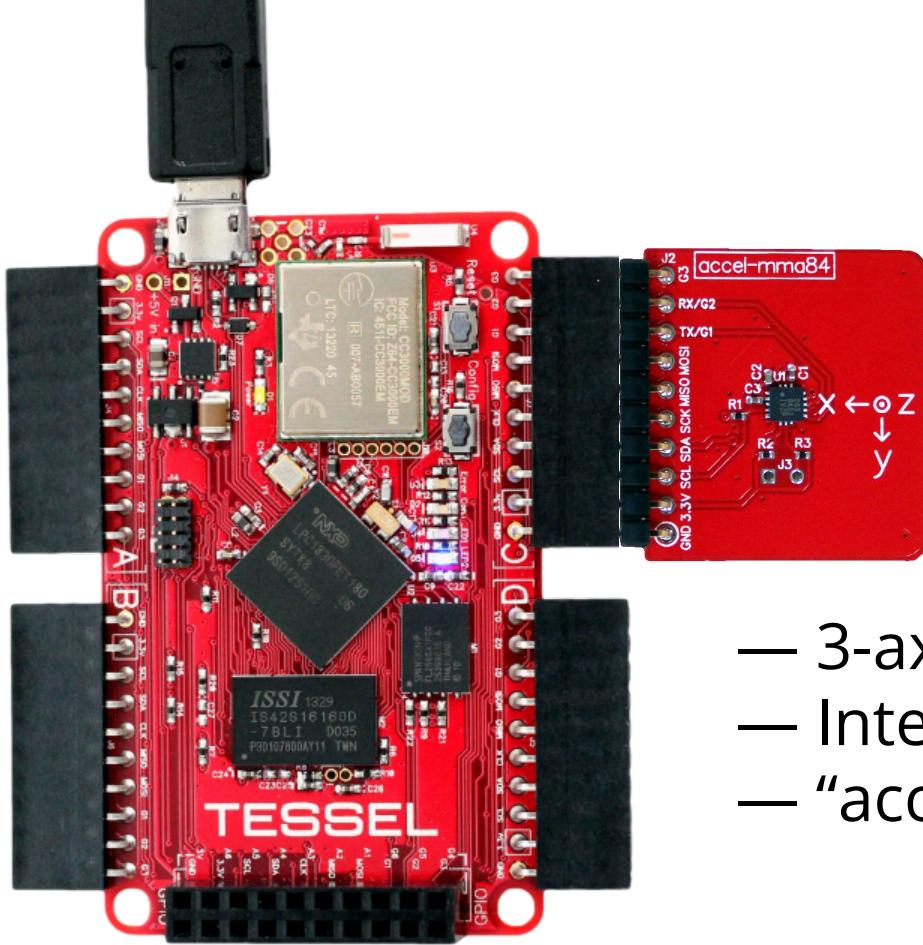
```
...
```

```
$ tessel repl
```

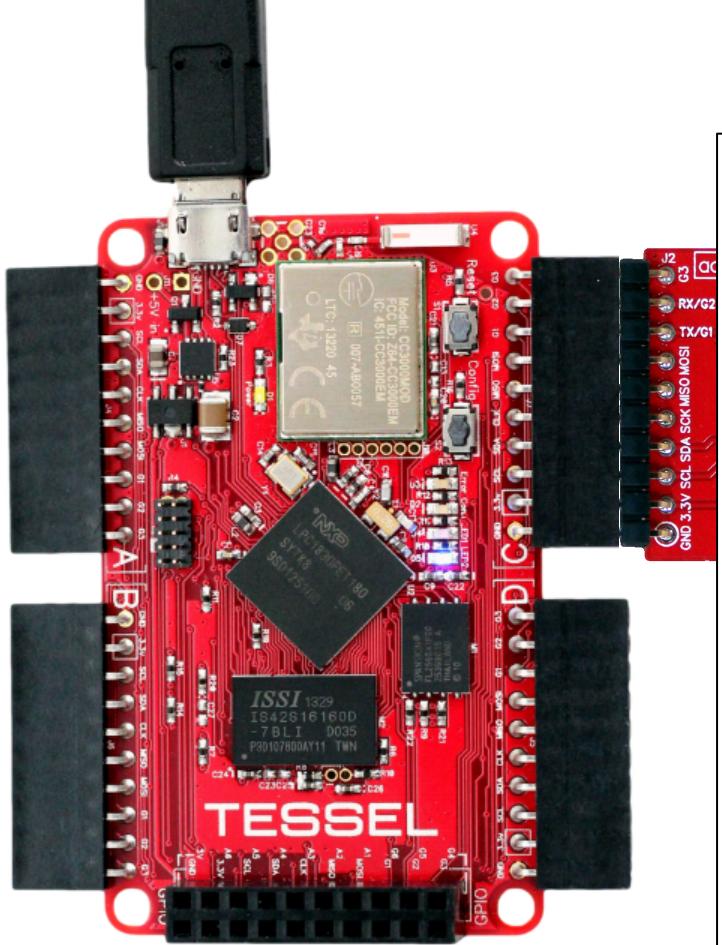
```
> tessel = require('tessel')
> tessel.led[0].low()
> tessel.led[0].high()
```



- 3-axis accelerometer
- Interrupt based
- “accel-mma84”



- 3-axis accelerometer
- Interrupt based
- “accel-mma84”



```
$ npm install accel-mma84
```

```
...
```

```
$ tessel run example.js
```

```
x: 0.01 y: 0.02 z: 1.01
```

```
x: 0.00 y: 0.02 z: 1.00
```

```
x: -0.02 y: -0.02 z: 1.00
```

```
x: 0.00 y: 0.03 z: 0.96
```

```
x: -0.02 y: -0.01 z: 1.00
```

```
...
```

- ```
var tessel = require('tessel');
var acellib = require('accel-mma84')

var accel = acellib.use(tessel.port['A']);

accel.on('data', function (xyz) {
  console.log('x:', xyz[0].toFixed(2),
              'y:', xyz[1].toFixed(2),
              'z:', xyz[2].toFixed(2));
});
```

```
var tessel = require('tessel');
● var accellib = require('accel-mma84')

var accel = accellib.use(tessel.port['A']);

accel.on('data', function (xyz) {
  console.log('x:', xyz[0].toFixed(2),
              'y:', xyz[1].toFixed(2),
              'z:', xyz[2].toFixed(2));
});
```

```
var tessel = require('tessel');
var accellib = require('accel-mma84')
```

- ```
var accel = accellib.use(tessel.port['A']);

accel.on('data', function (xyz) {
  console.log('x:', xyz[0].toFixed(2),
              'y:', xyz[1].toFixed(2),
              'z:', xyz[2].toFixed(2));
});
```

```
var tessel = require('tessel');
var accellib = require('accel-mma84')

var accel = accellib.use(tessel.port['A']);
```

- `accel.on('data', function (xyz) {  
 console.log('x:', xyz[0].toFixed(2),  
 'y:', xyz[1].toFixed(2),  
 'z:', xyz[2].toFixed(2));  
});`

*Demo*

## II. Talking to Hardware

# HTTP

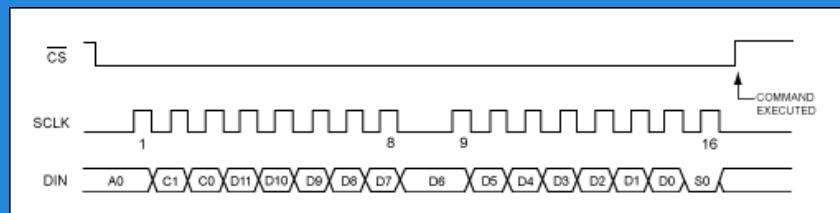
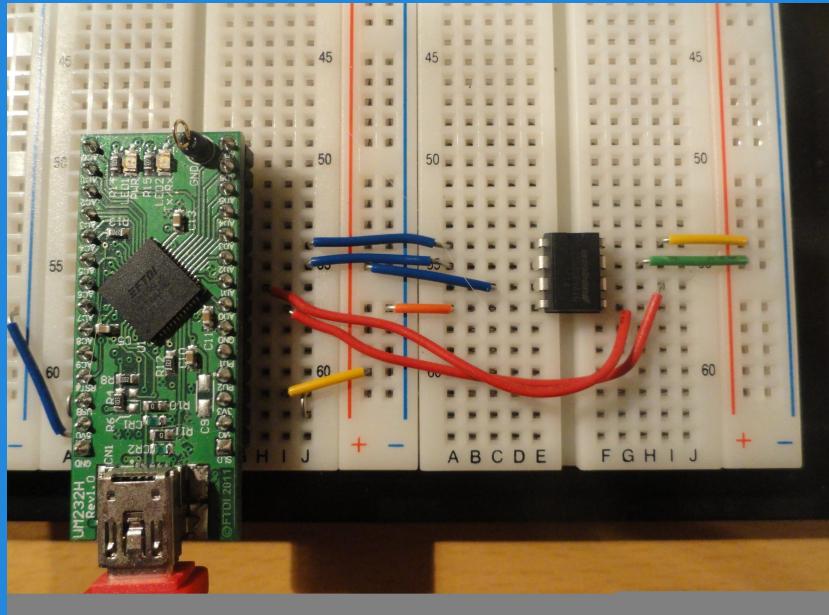


# HTTP

# SPI

# I2C

# UART



Triple Axis Accelerometer X

SparkFun Electronics [US] https://www.sparkfun.com/products/12756

\$ USD

sparkfun

Shop Learn Forum Data

0 items You are not logged in. log in / register

Start a Project Products Blog Tutorials Videos Wish Lists Distributors Support search...

HOME / PRODUCT CATEGORIES / 3-AXIS / SEN-12756

{ } {}

**Triple Axis Accelerometer Breakout - MMA8452Q**  
SEN-12756 RoHS ✓ \*







 SHARE

**Description:** This breakout board makes it easy to use the tiny MMA8452Q accelerometer in your project. The MMA8452Q is a smart low-power, three-axis, capacitive MEMS accelerometer with 12 bits of resolution. This accelerometer is packed with embedded functions with flexible user programmable options, configurable to two interrupt pins. Embedded interrupt functions allow for overall power savings relieving the host processor from continuously polling data.

The MMA8452Q has user selectable full scales of  $\pm 2g$ / $\pm 4g$ / $\pm 8g$  with high pass filtered data as well as non filtered

**\$9.95**

Add to Cart

1	quantity
<input checked="" type="radio"/>	213 in stock
\$9.95	1+ units
\$8.96	10+ units
\$7.96	100+ units

© images are CC BY-NC-SA

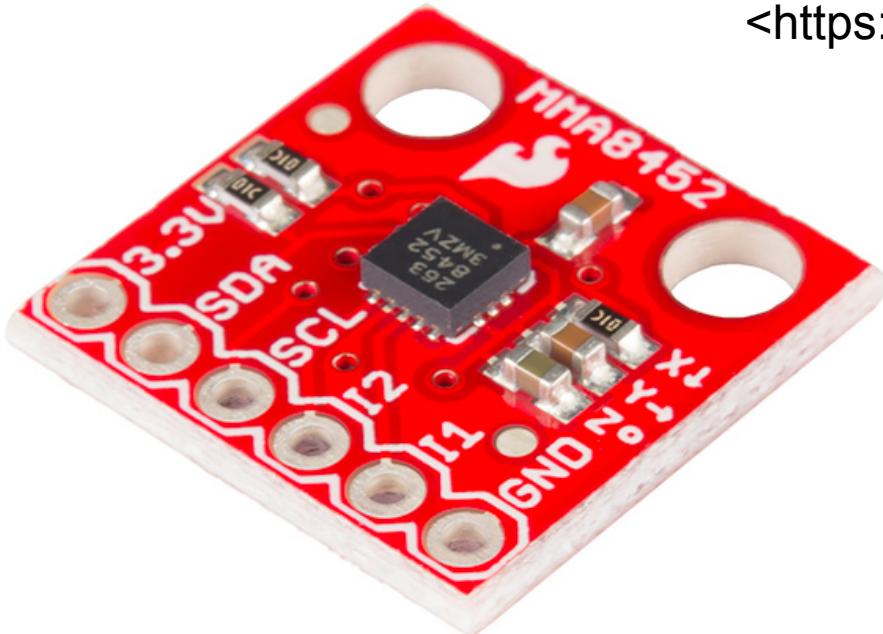
# Triple-Axis Accelerometer Breakout

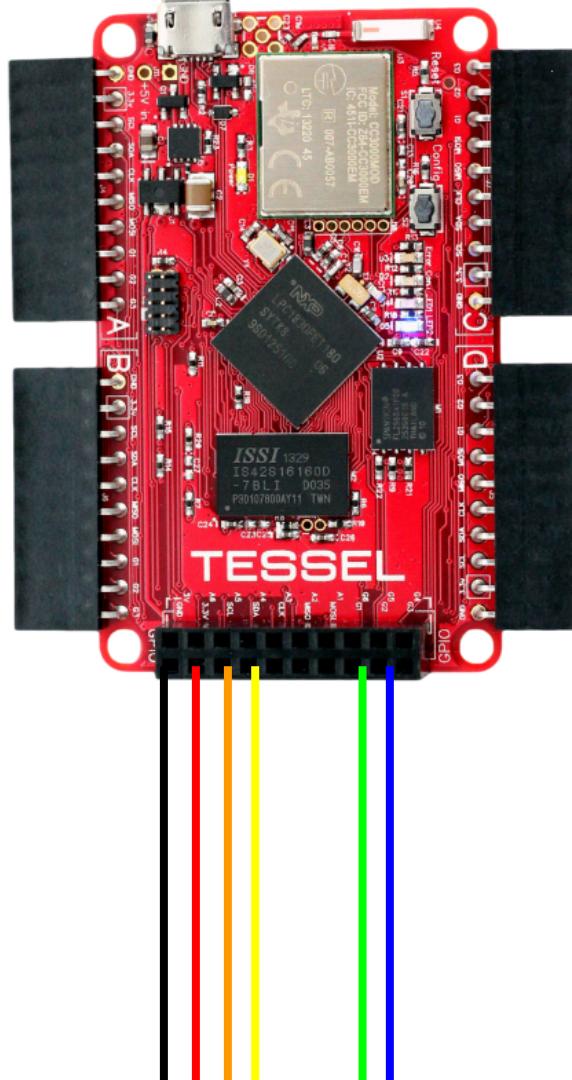
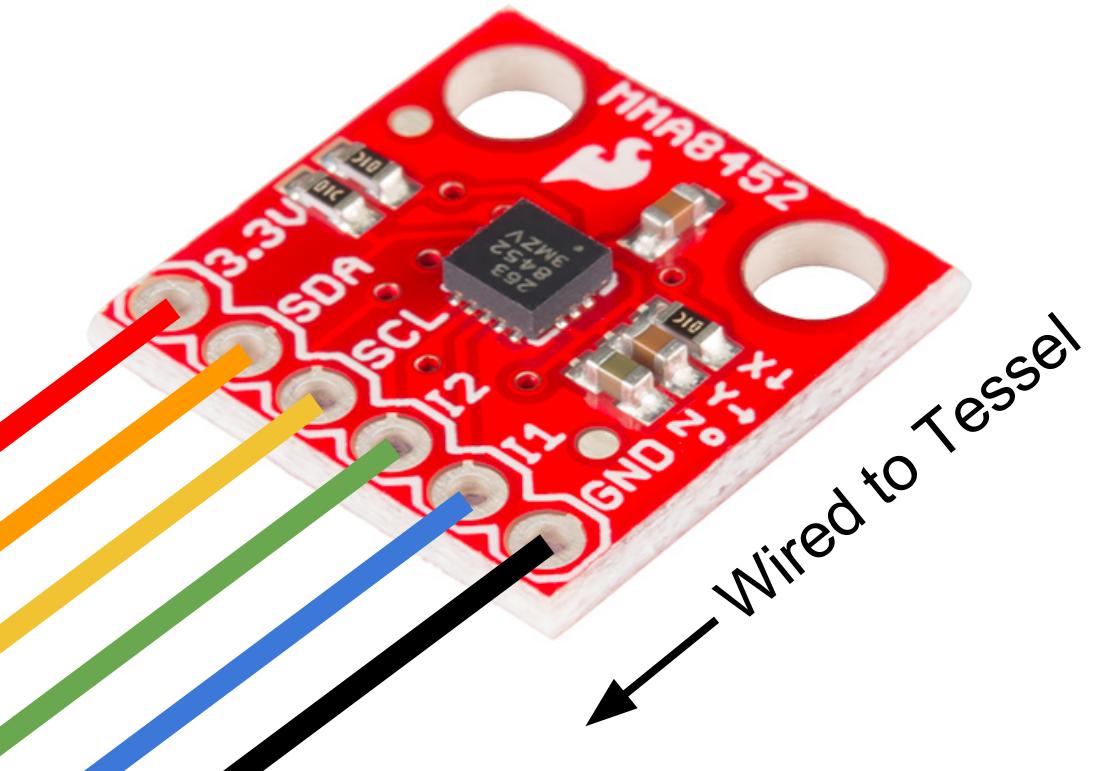
<<https://www.sparkfun.com/products/12756>>



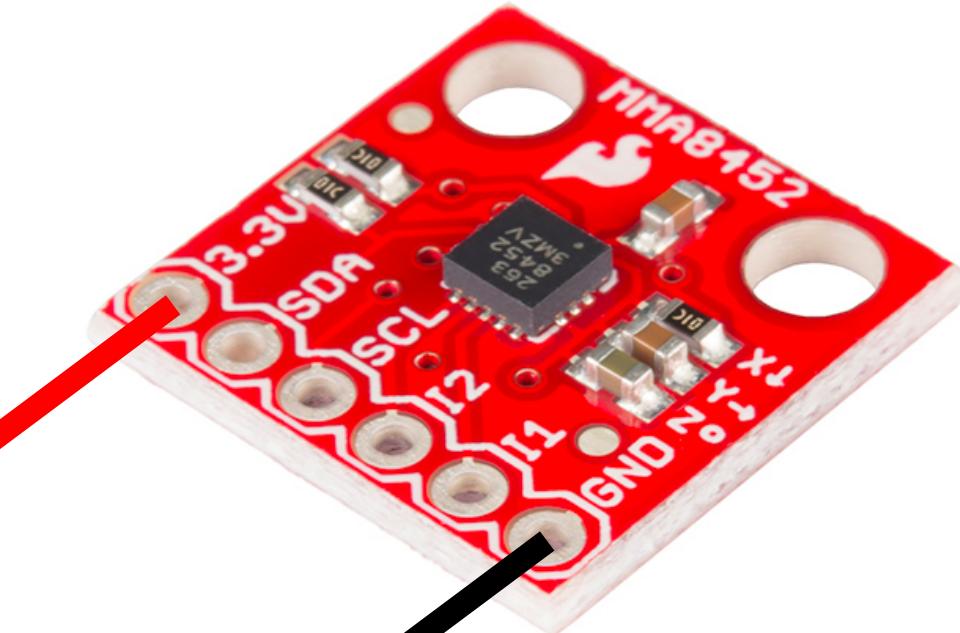
# Triple-Axis Accelerometer Breakout

<<https://www.sparkfun.com/products/12756>>





# 3.3V and GND



- Tessel runs off 3.3V
- Steps down 5V USB
- Can run off 3.3V directly
- ~2 AA batteries

# Interrupt lines

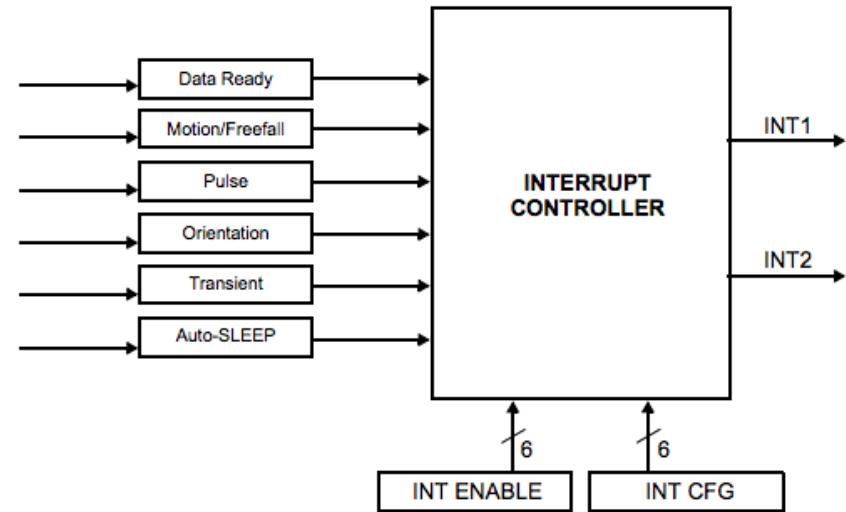
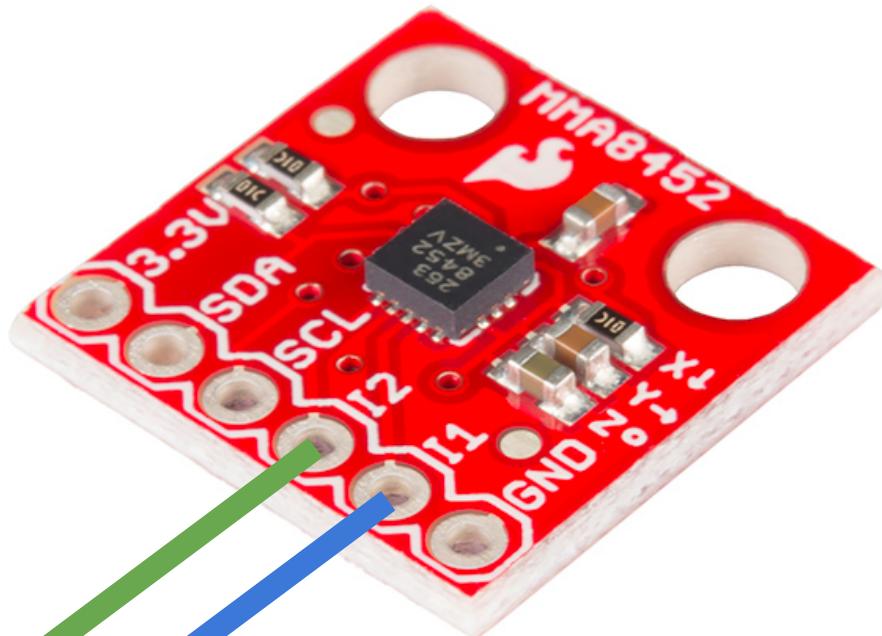


Figure 10. System Interrupt Generation Block Diagram

# Interrupt lines

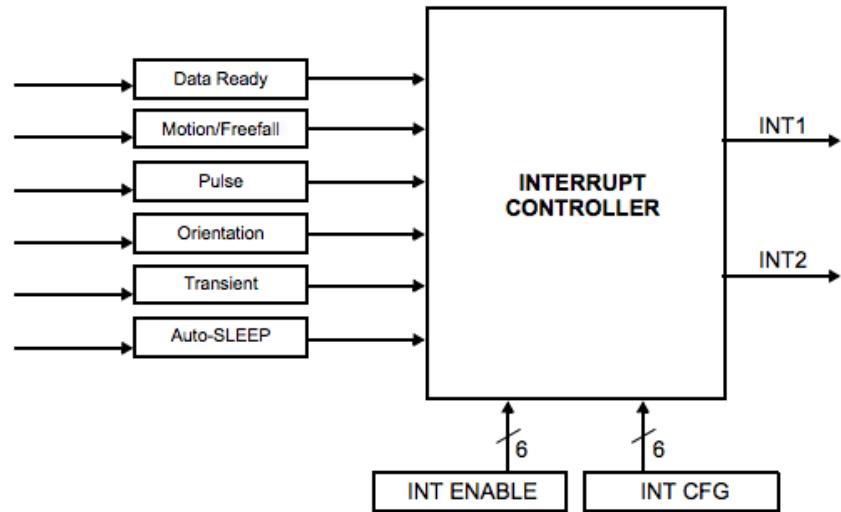
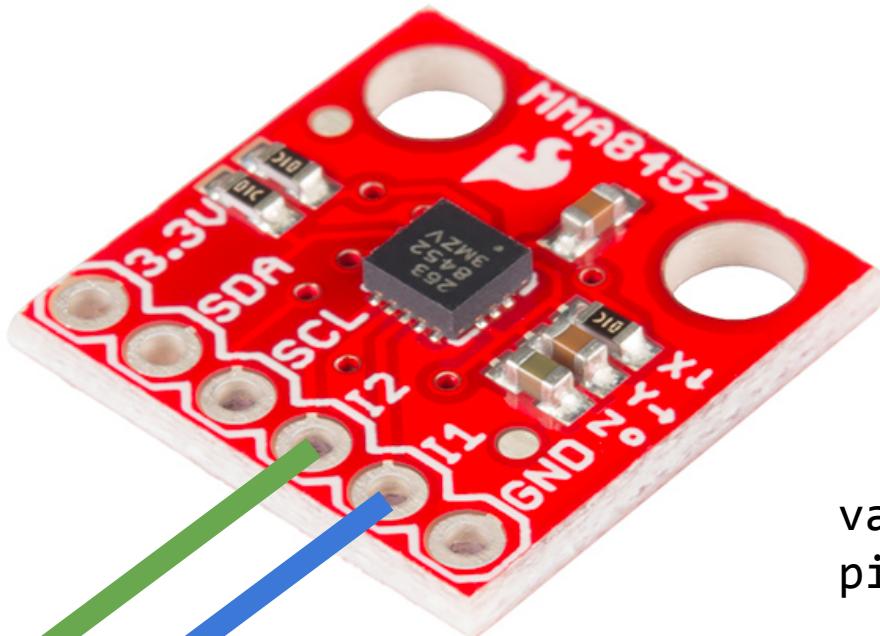
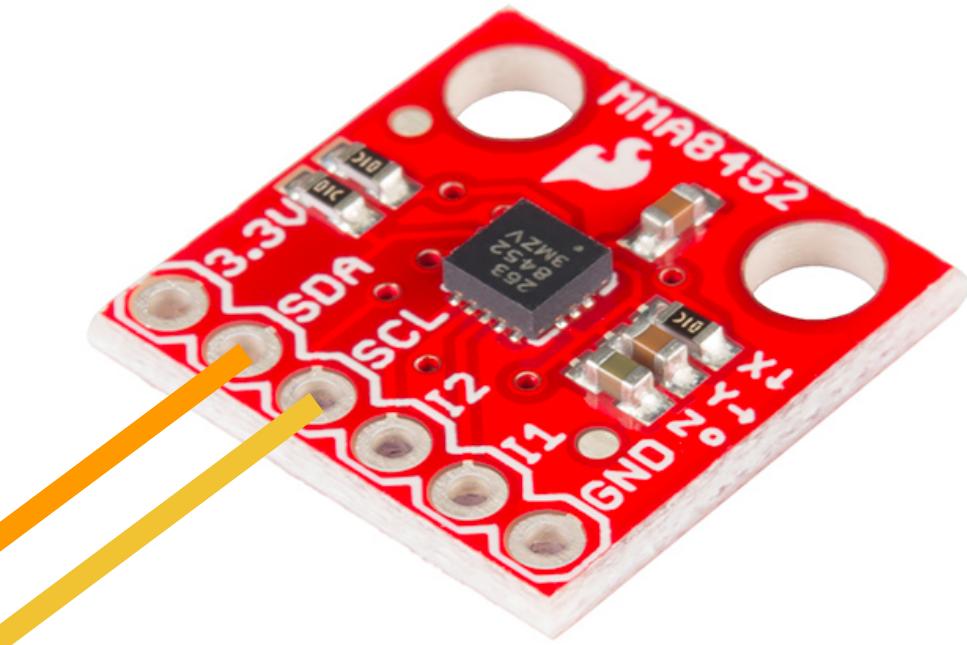


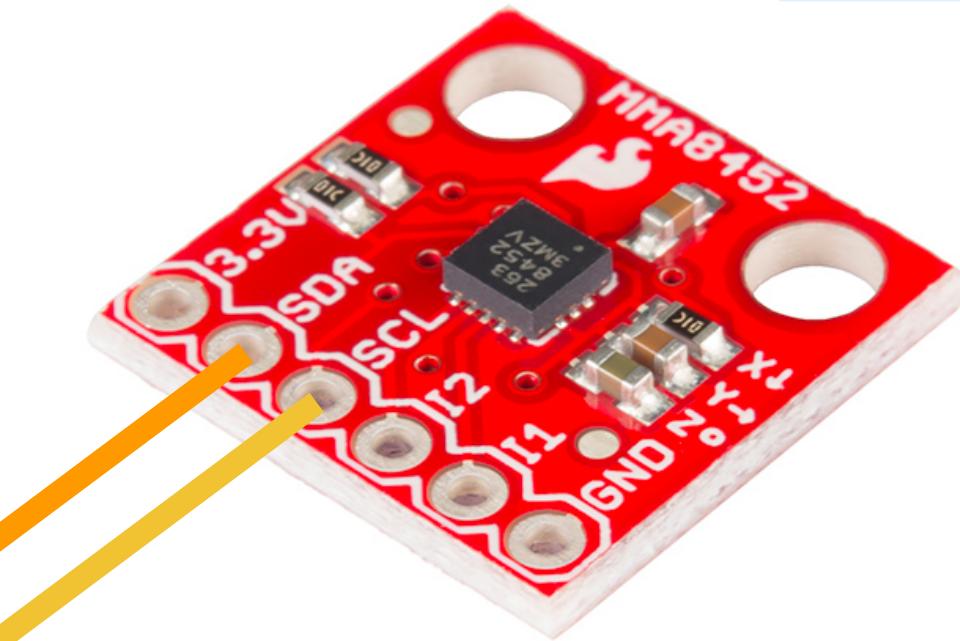
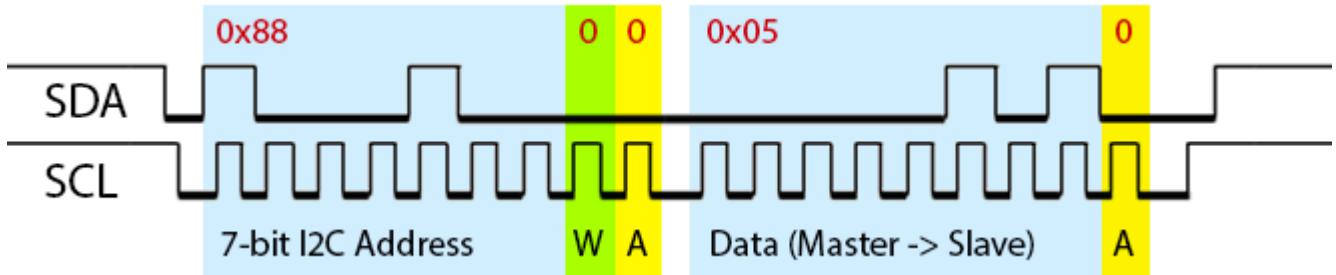
Figure 10. System Interrupt Generation Block Diagram

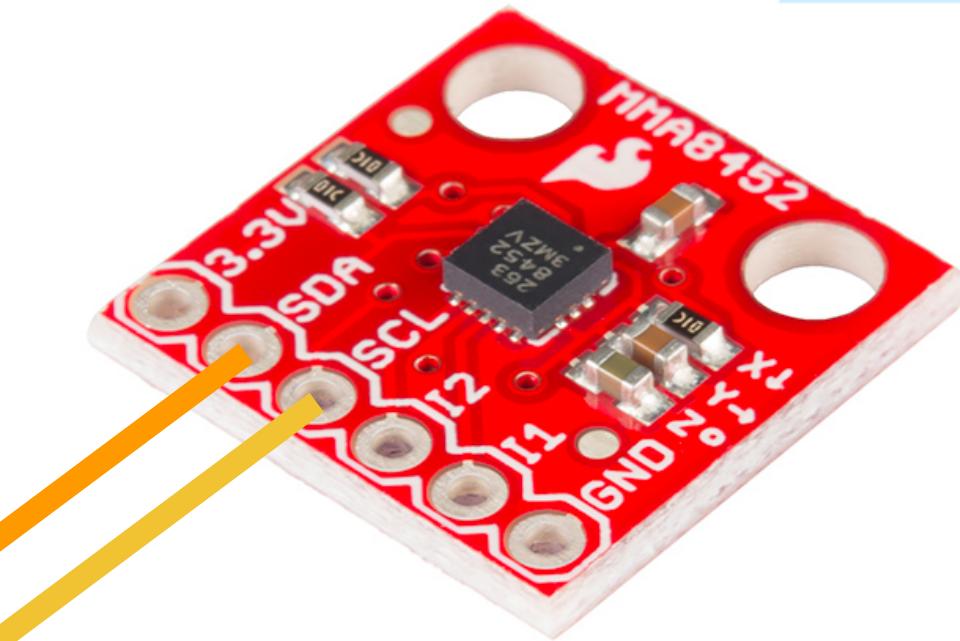
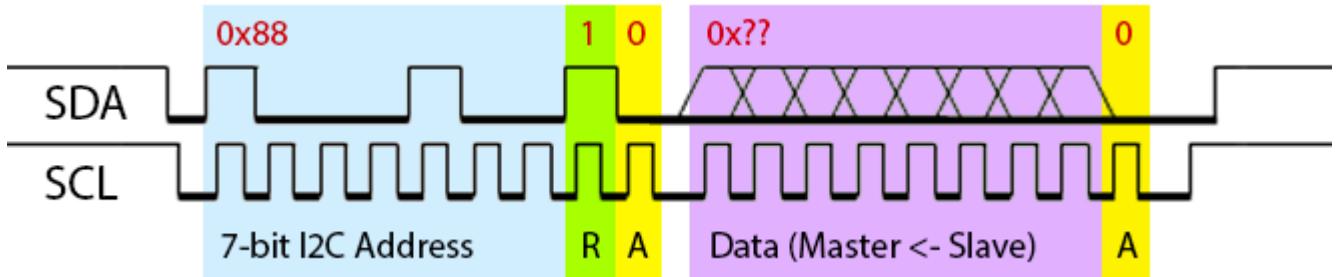
```
var pin = tessel.port['A'].digital[0]
pin.on('rise', function () {
  /* interrupt fired!
})
```

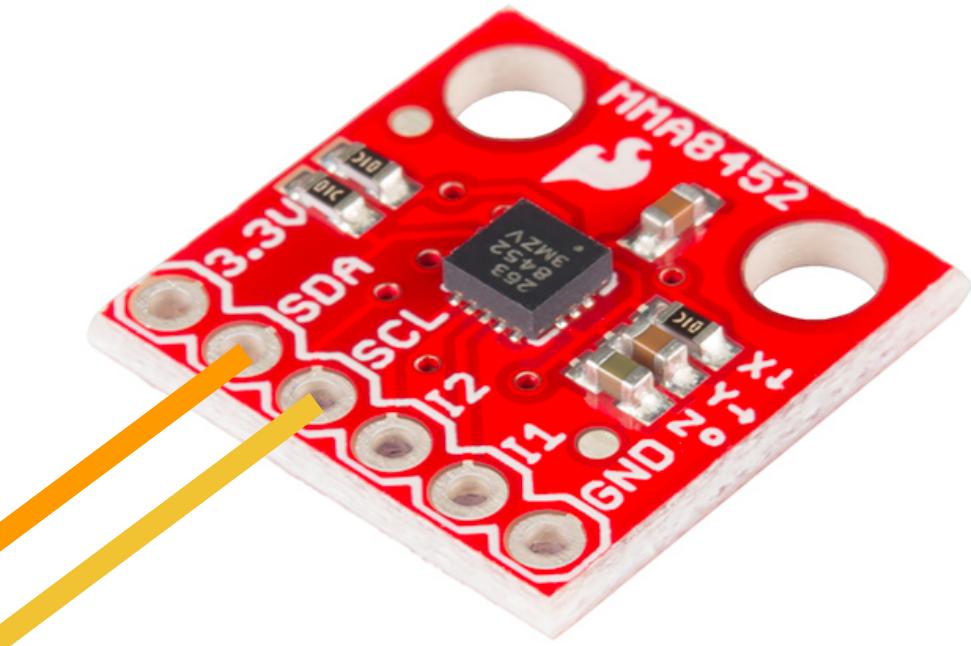
# I2C signal



- Many ICs talk one of I2C, SPI, or UART
- **Pro:** Connect many devices on two lines
- **Pro:** Anything can initiate a connection
- **Con:** Complex, also slower than SPI or UART





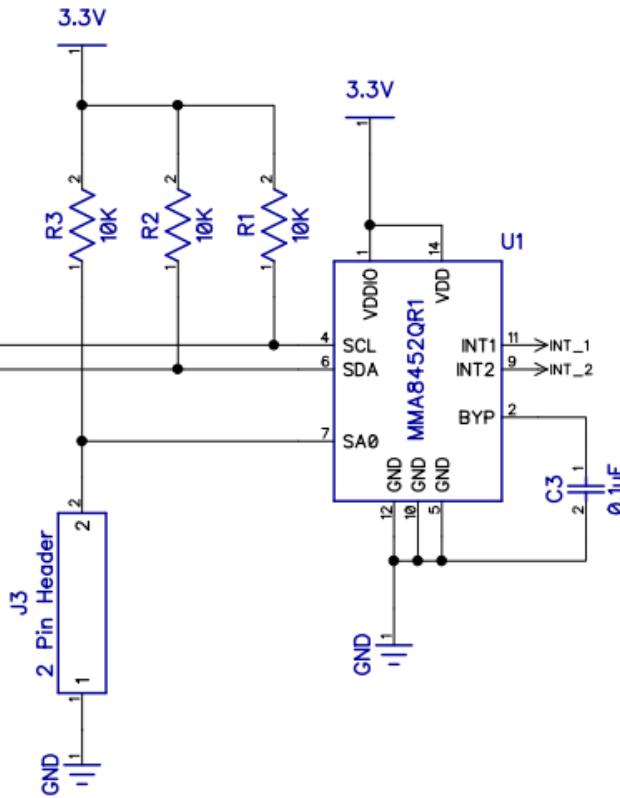
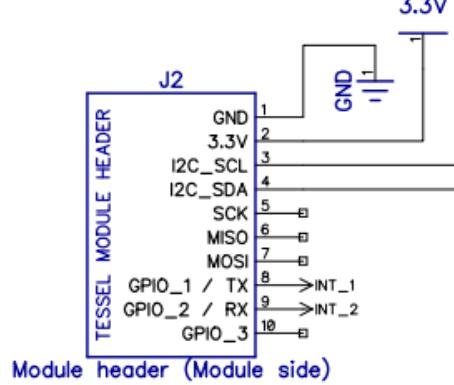
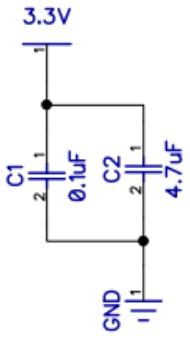


```
var port = tessel.port['A']
var i2c = new port.I2C(0x1D)
var reg = 0x01;
var out = new Buffer([reg]);

i2c.transfer(out, 6,
function (err, data) {
  console.log(xyz);
});

// prints XYZ data
```

# III. Making Hardware



Pcbnew (2011-nov-30)-testing /usr/share/doc/kicad/demos/kit-dev-coldfire-xilinx\_5213/kit-dev-coldfire-xilinx\_5213.brd [Read Only]

File Edit View Place Preferences Tools Design Rules Help

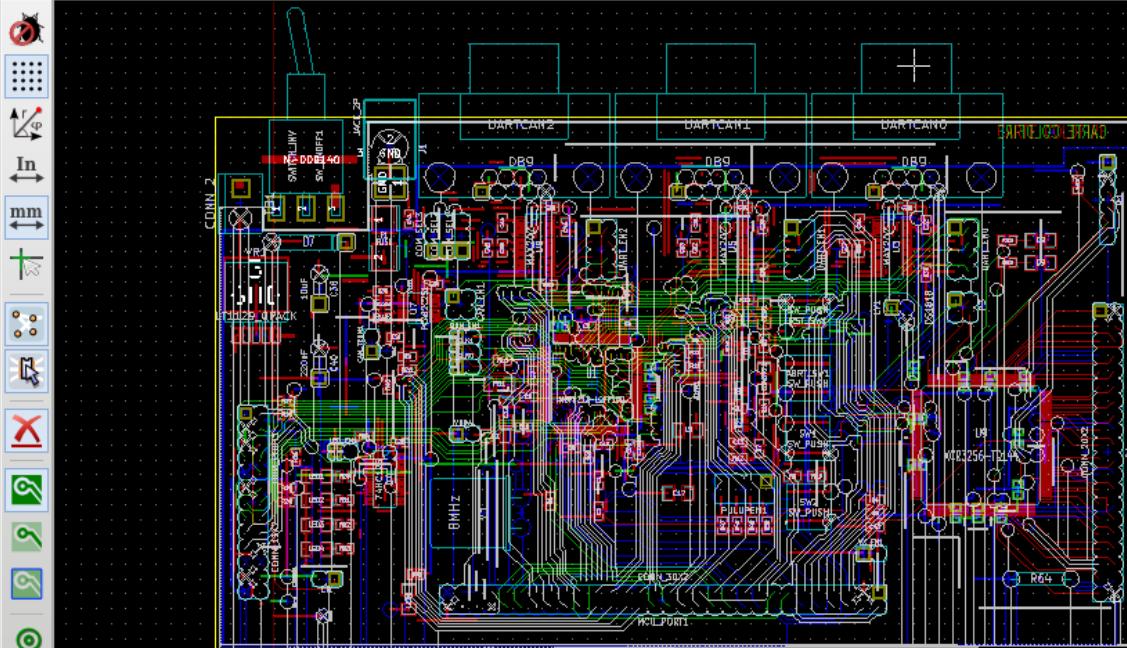


Track 0.198 mm \*

Via 0.635 mm \*

Grid 1.270

Auto



Visibles

Layer Render

- Composant
- GND\_layer
- 3.3V\_layer
- Cuivre
- Adhes\_Front
- Adhes\_Back
- SoldP\_Front
- SoldP\_Back
- SilkS\_Front
- SilkS\_Back
- Mask\_Front
- Mask\_Back
- Drawings
- Comments
- Eco1
- Eco2
- PCB\_Edges

Pads	821	Vias	253	trackSegm	3275	Nodes	743	Nets	210	Links	534	Connect	534	Unconnected	0
------	-----	------	-----	-----------	------	-------	-----	------	-----	-------	-----	---------	-----	-------------	---

Z 93.040 X 190.500 Y 46.990

dx 190.500 dy 46.990 d 196.210

mm

PCB routing for open source

dirtycircuits.com

## Dirty Circuits PCB Routing

Rough routing? No worries!

2 layer PCB routing in Eagle and KiCAD

Hand routing by the experienced team at Dangerous Prototypes

Order   Status   About   Contact   [Dirty Boards PCBs](#)   [Dirty Cables](#)

Submit your routing job

Submit a circuit drawing, KiCAD, or Eagle files

Submit a circuit drawing, KiCAD, or Eagle files. Boards returned in 2-7 days.

Project name	Files	License	Urgency	Hours	Cost
MyAccelerometer	<input type="button" value="Choose File"/> No file chosen	OSHW	Relaxed	3	\$105

CircuitHub

https://circuithub.com

CircuitHub

See example pricing    Login

# Scalable Electronics Manufacturing

Instantly quote and scale manufacturing.

The image illustrates the workflow for electronic manufacturing on CircuitHub. It starts with a screenshot of the EAGLE PCB editor showing a complex circuit board design. An arrow points from this to a screenshot of the CircuitHub.com quoting interface. This interface displays two schematic diagrams, a quantity selector set from '1 unit' to '10,000', and a time selector set from '1 day' to '3 mo'. A prominent blue 'FABRICATE' button is at the bottom. A second arrow points from the quoting interface to a photograph of a physical Arduino Uno R3 microcontroller board.

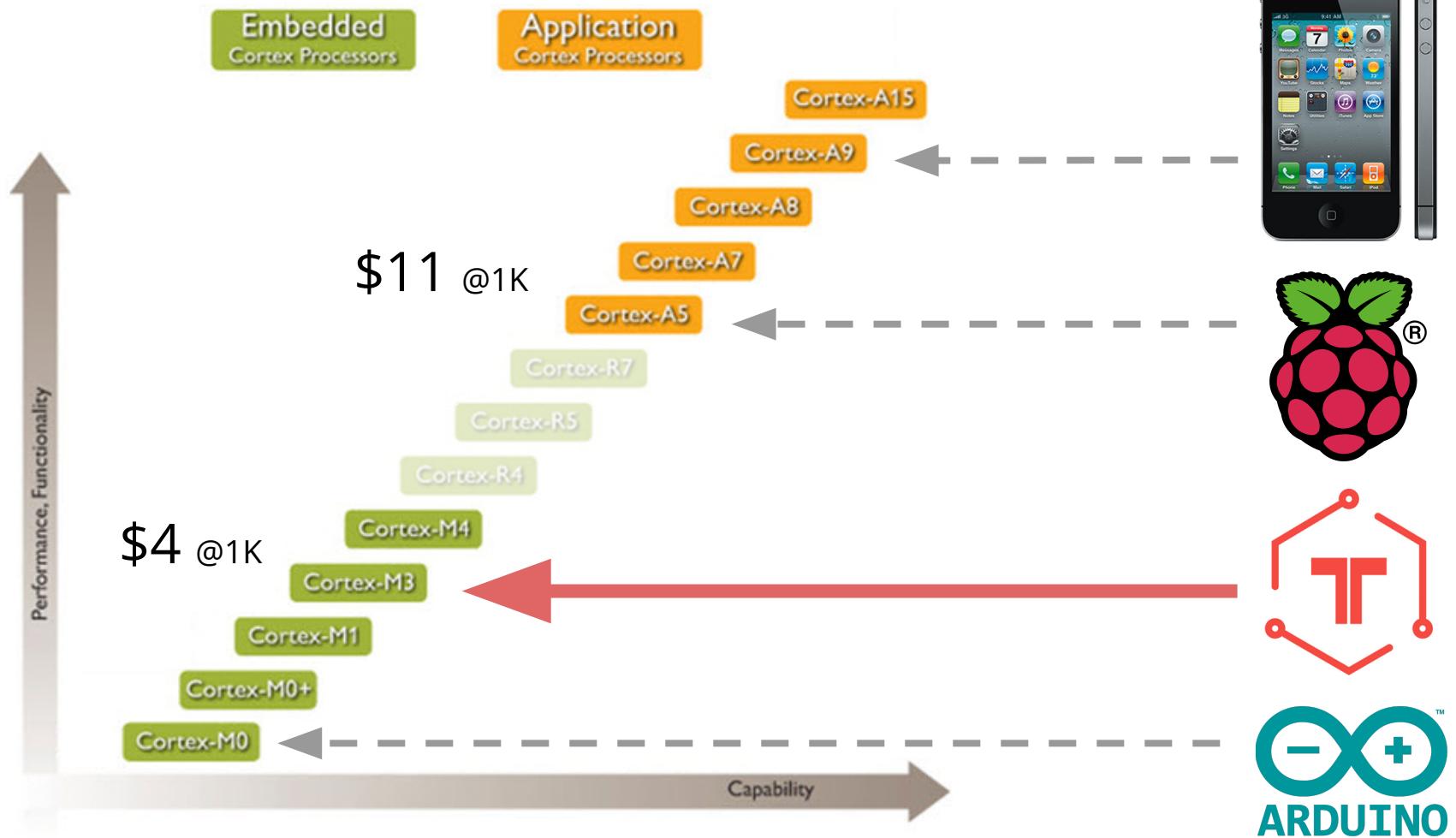
Lesson: Hardware is easy, and you  
can too!

Practical Lesson: Every VC pitch in  
five years will include a hardware  
gimmick, so get ready.

# Open Source Hardware

**SPEED & ENDURANCE**





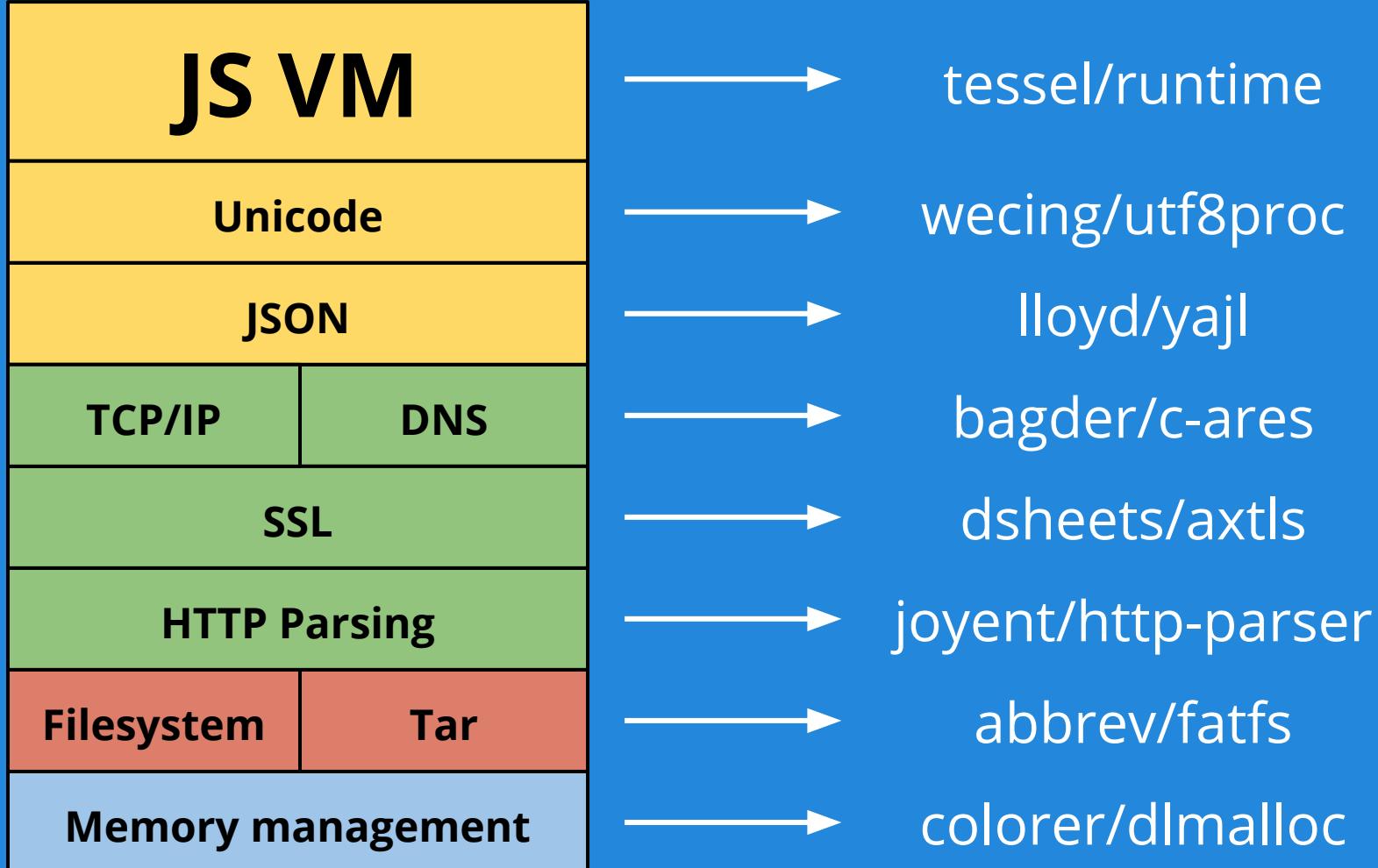
It's becoming feasible to embed in  
every product a microcontroller  
powerful enough to run a high level  
language.



- Chrome's JS engine
- C++
- ~10mb RAM for every process
- POSIX/Win32 environment



- embeddable language
- Written in C
- ~30kb memory
- Highly portable + embeddable



Q: “How do you make money?”

Q: “How do you make money?”

A: “By selling stuff.”

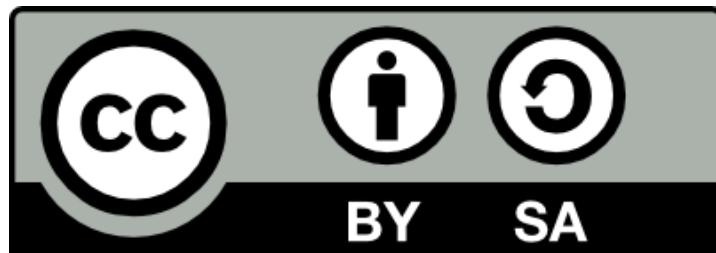
2003



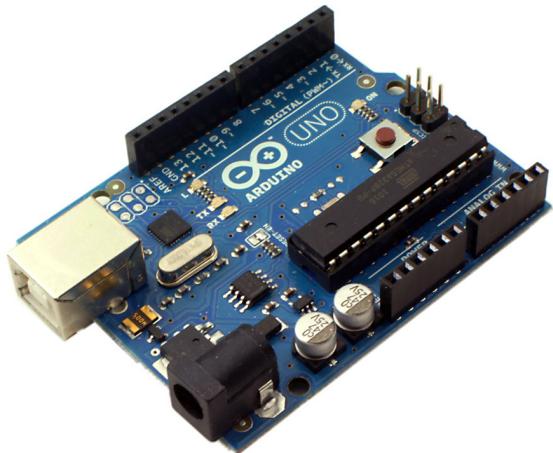
2005



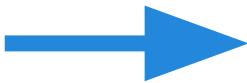
2005



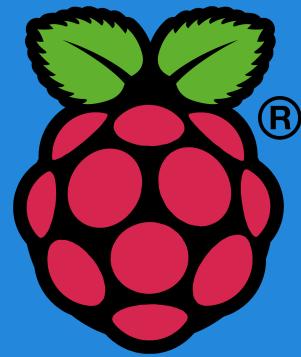
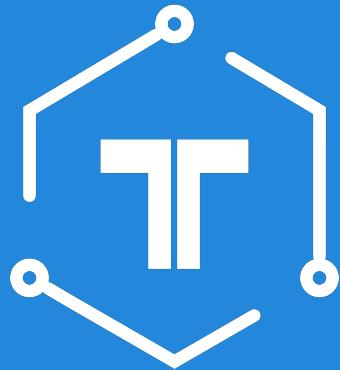




2005



2014



**1985**



**FREE SOFTWARE  
FOUNDATION**

**1998**



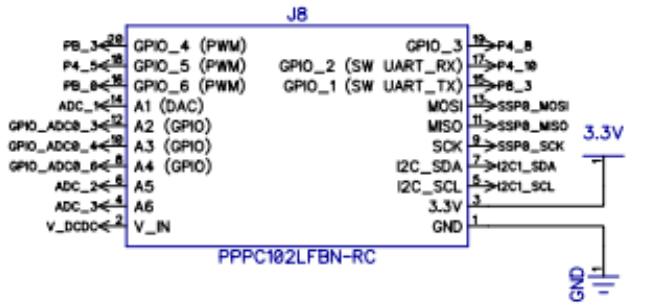
**open source  
initiative**

**2010**



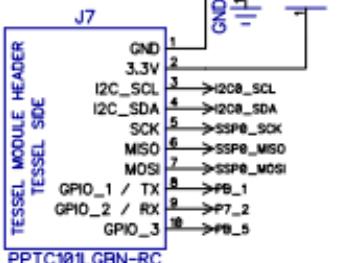
**open source  
hardware**

## GPIO BANK

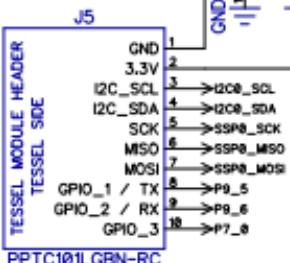


SOFTWARE UART POSSIBLE ON PORT C AND GPIO  
I2C BUSSES SHARED ON PORTS A/B/GPIO, C/D

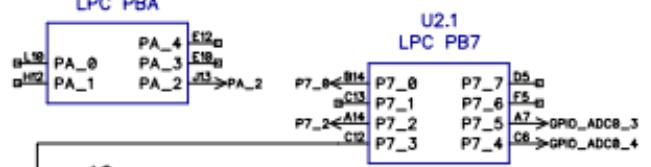
## UPPER LEFT (C)



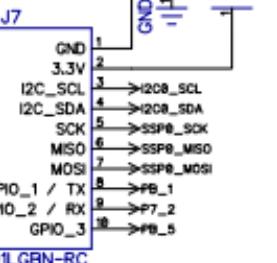
## UPPER RIGHT (D)



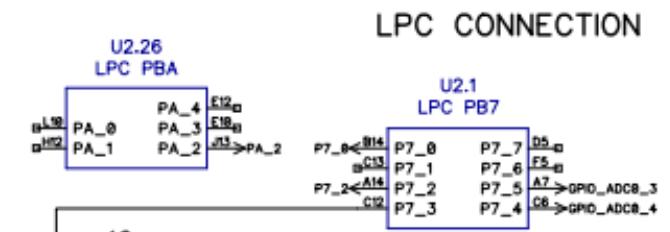
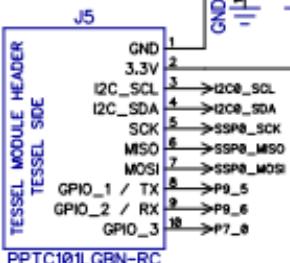
## LPC CONNECTION



## LOWER LEFT (A)



## LOWER RIGHT (B)



<http://github.com/tessel>

<http://github.com/tessel/firmware>

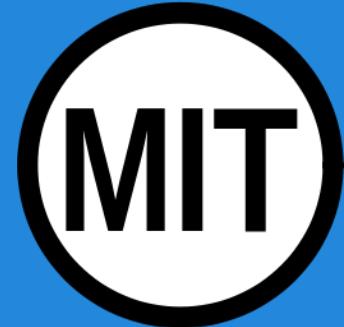
<http://github.com/tessel/runtime>

<http://github.com/tessel/cli>

<http://github.com/tessel/hardware>

<http://github.com/tessel/docs>

<http://github.com/tessel/logos>



# technicalmachine

Full Name      Technical Machine

Email            [team@technical.io](mailto:team@technical.io)

Github          [technicalmachine](#)

Twitter        [@technicalhumans](#)

Homepage      [www.technical.io](http://www.technical.io)

## Packages by technicalmachine

- [accel-mma84](#)
- [ambient-attx4](#)
- [audio-vs1053b](#)
- [bglib](#)
- [ble-ad-parser](#)
- [ble-ble113a](#)
- [bleadvertise](#)
- [camera-vc0706](#)
- [climate-si7005](#)
- [gprs-sim900](#)
- [gps-a2235h](#)
- [ir-attx4](#)
- [rf-nrf24](#)
- [rfid-pn532](#)
- [sdcard](#)
- [servo-pca9685](#)



Tessel Projects X

https://projects.tessel.io/projects

TESSEL beta

All Projects + New Login

Project Title

Filter by modules:

- Accelerometer
- Ambient
- Audio
- BluetoothLE
- Camera
- Climate
- GPS



**Tesselcam**  
Kelsey 15 views



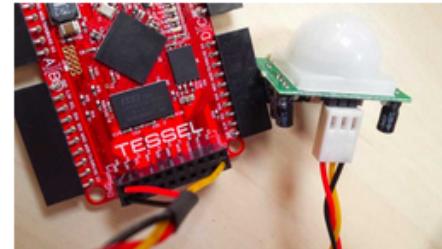
**NES controller - Tesselized**  
stefanvandockum 36 views



**Pulse Sensor**  
ifoundthemeaningoflife 59 views



**Wii Nunchuck - Tesselized**  
16 views



**PIR Motion Detector**  
16 views



**EnviroReport Web**  
Build your own environmental monitoring system

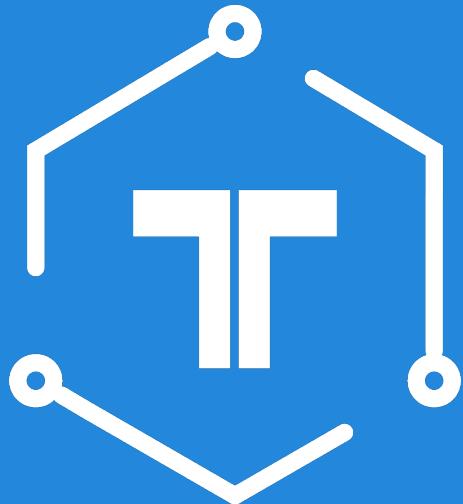
Report Issues



**Be bold!**  
**Be courageous!**  
**Be amazing!**



**Be bold!**  
**Be courageous!**  
**Be amazing!**  
**Build robots!**



tessel.io

tim@technical.io

@technicalhumans









how'd you start a company

2003

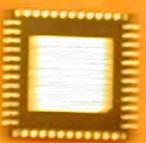


2005

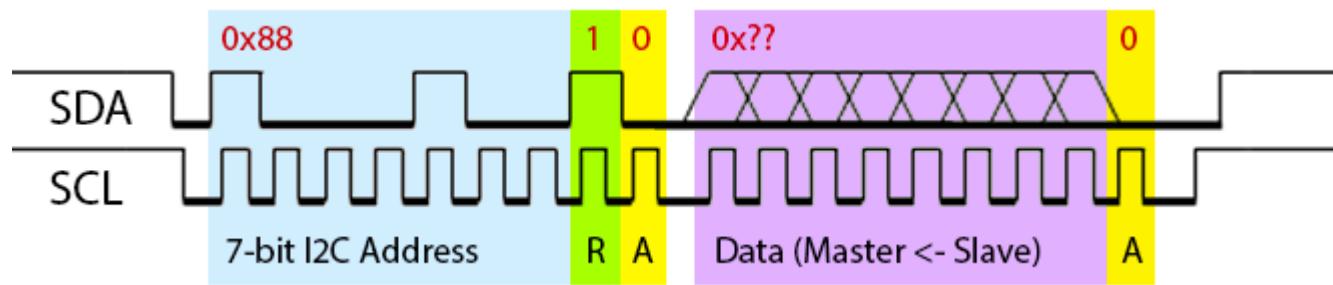
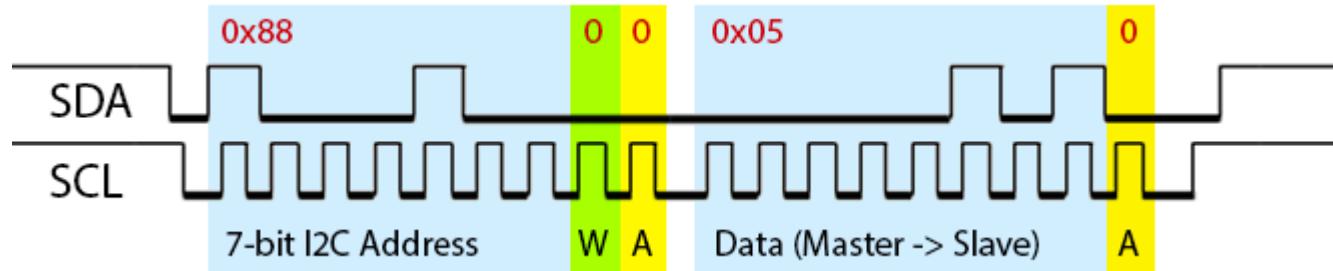


2005

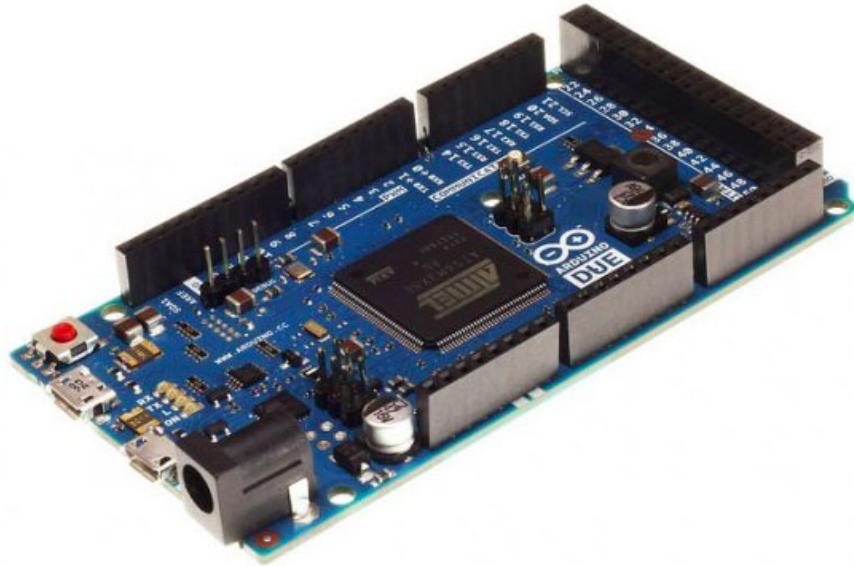
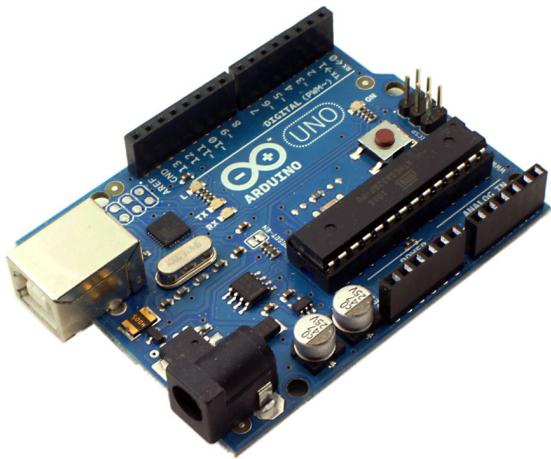




We can today  
**afford the processing power**  
to make developers' lives easier.



```
var i2c = new port.I2C(0x88, { speed: 5e6 })
i2c.read(2, function (err, data) {
  // got a register value
})
```



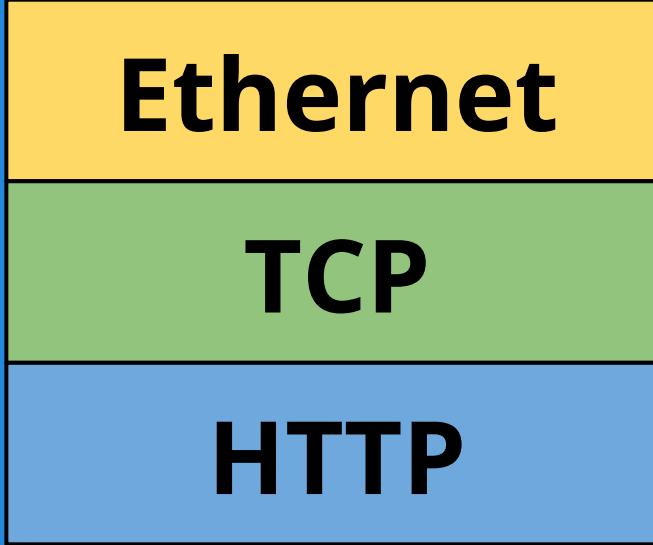
**8-bit**

**32-bit**

**Ethernet**

**TCP**

**HTTP**



**Ethernet**

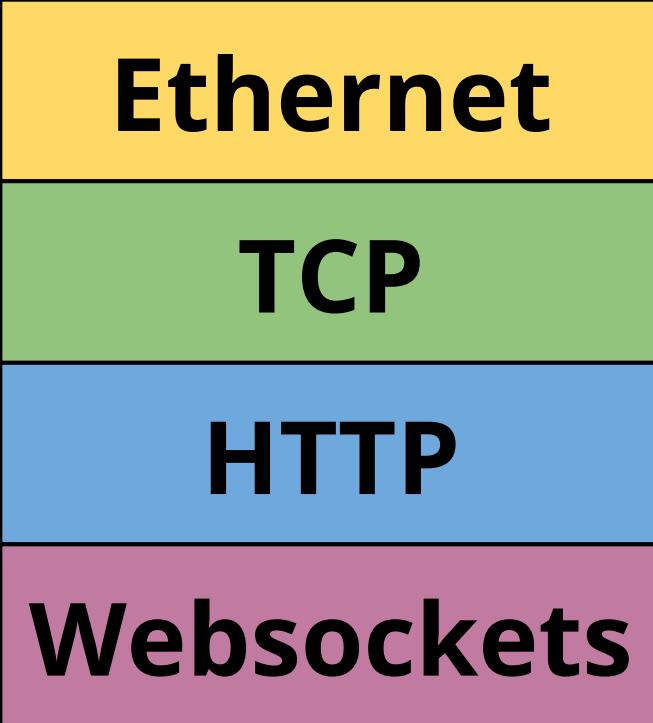
(Packet-based)

**TCP**

(Stream-based)

**HTTP**

(Packet-based)



**Ethernet**

(Packet-based)

**TCP**

(Stream-based)

**HTTP**

(Packet-based)

**Websockets**

(Streaming...)

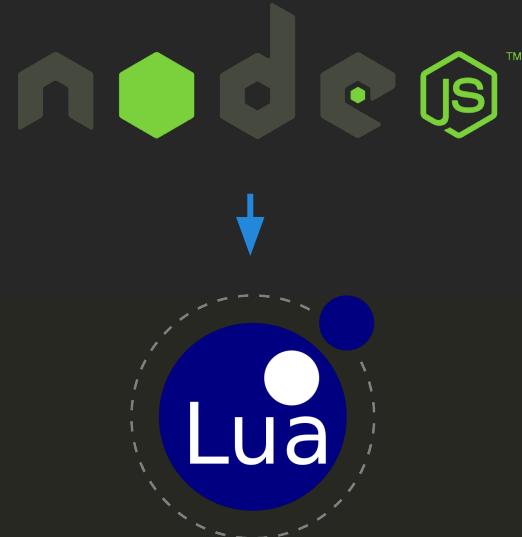
# Community

```
1 var tessel = require('tessel');
2
3 var led1 = tessel.led[1].writeSync('high')
4 var led2 = tessel.led[2].writeSync('low')
5
6 var i = 0;
7 setInterval(function () {
8   console.log('Blinked', i++, 'times');
9   led1.toggleSync();
10  led2.toggleSync();
11 }, 100);
```

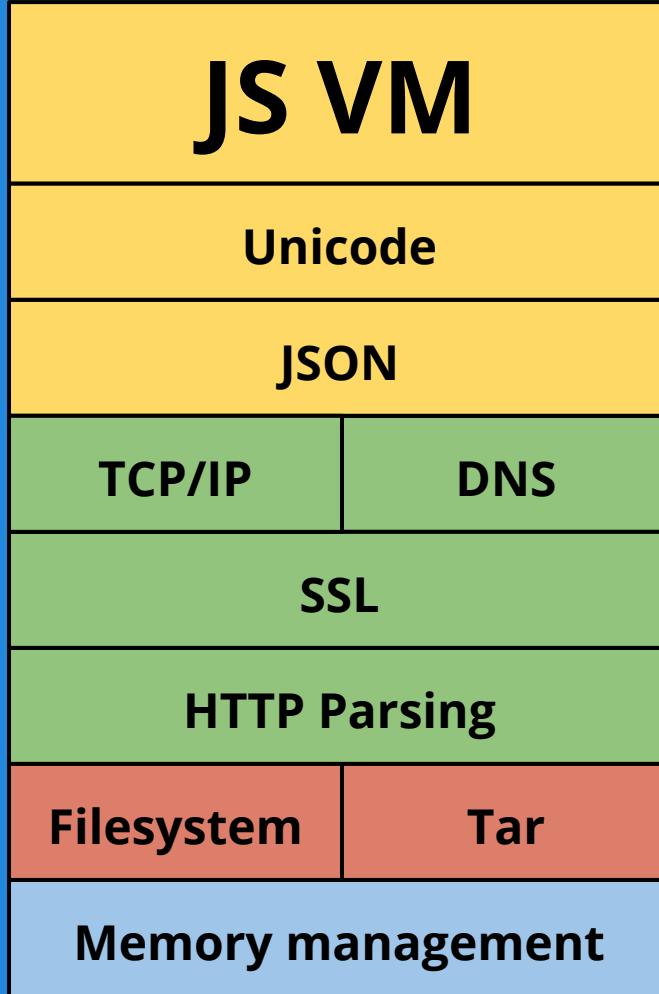


```
1 var tessel = require('tessel');
2
3 var led1 = tessel.led[1].writeSync('high')
4 var led2 = tessel.led[2].writeSync('low')
5
6 var i = 0;
7 setInterval(function () {
8   console.log('Blinked', i++, 'times');
9   led1.toggleSync();
10  led2.toggleSync();
11 }, 100);
```

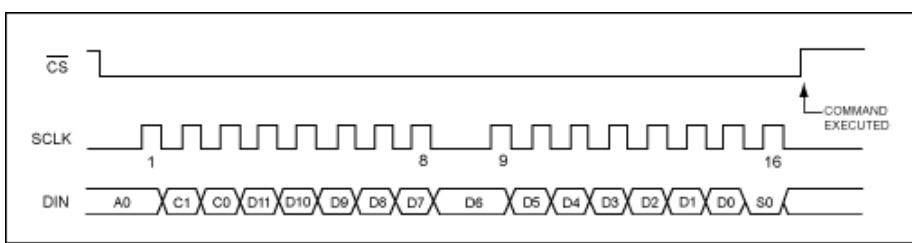
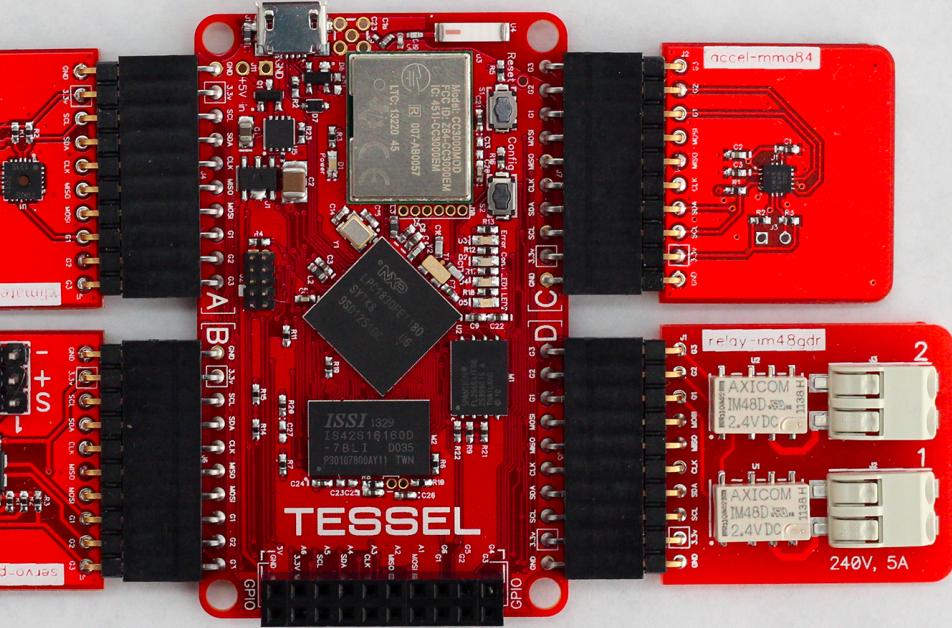
```
1 return function (_ENV, _module)
2 local exports, module = _module.exports, _module;
3
4 local tessel, led1, led2, i = tessel, led1, led2, i;
5 tessel = require(this, ("tessel"));
6 led1 = tessel:led((1)):output():high();
7 led2 = tessel:led((2)):output():low();
8 i = (0);
9 setInterval(this, (function (this)
10  console:log(("Blinked"), (function () local _r = i; i = _r + 1; return _r; end)(), ("times")));
11  led1:toggle();
12  led2:toggle();
13 end), (100));
14
15 return _module.exports;
16 end
```

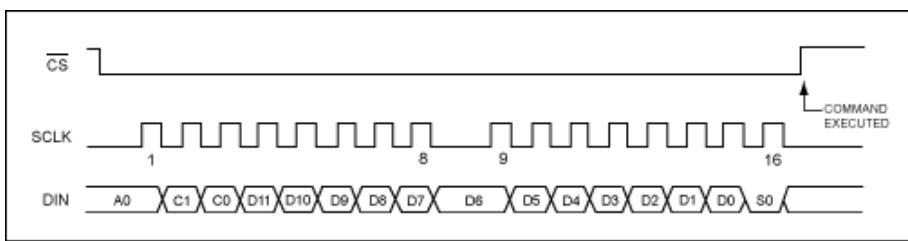
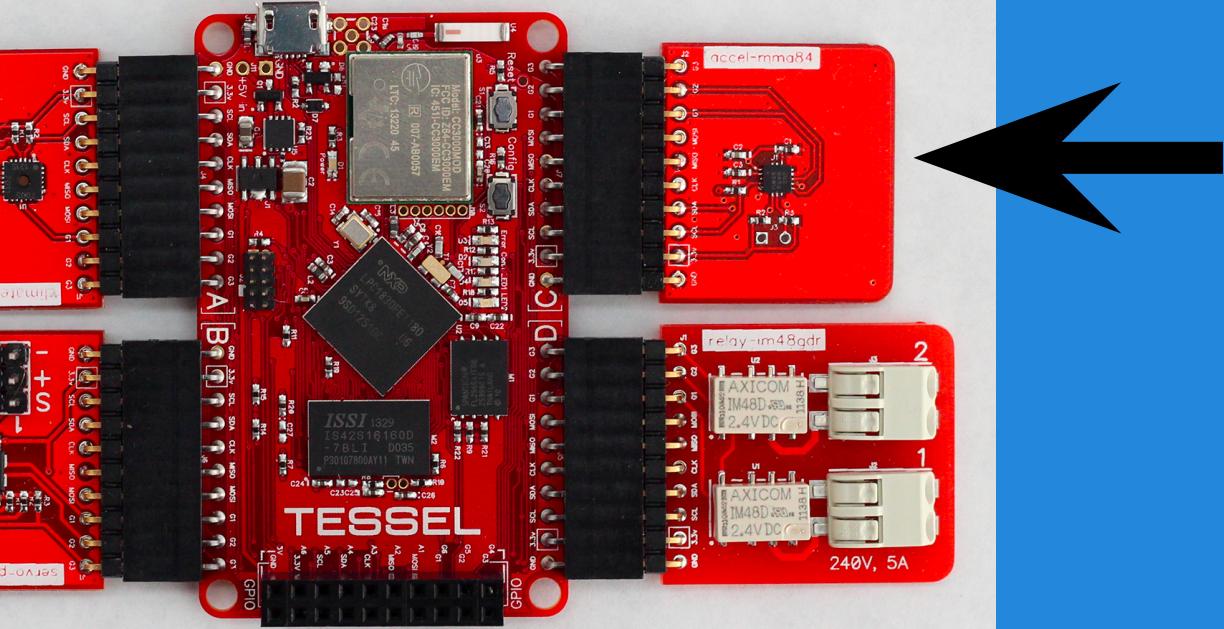


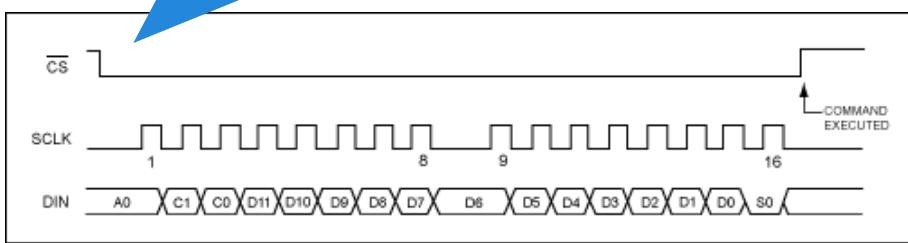
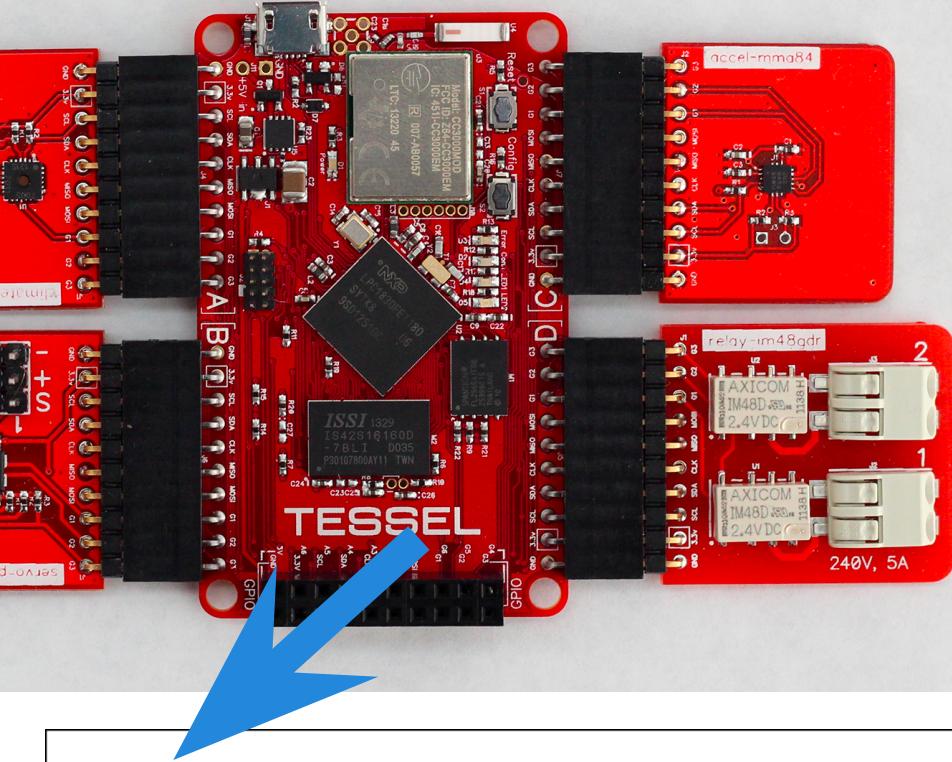
# Why JavaScript?

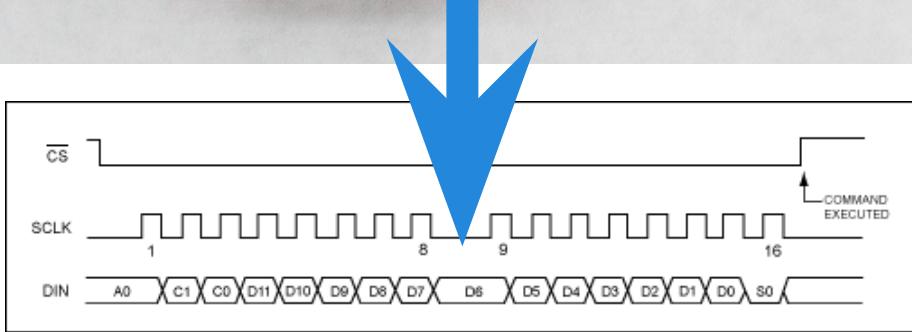
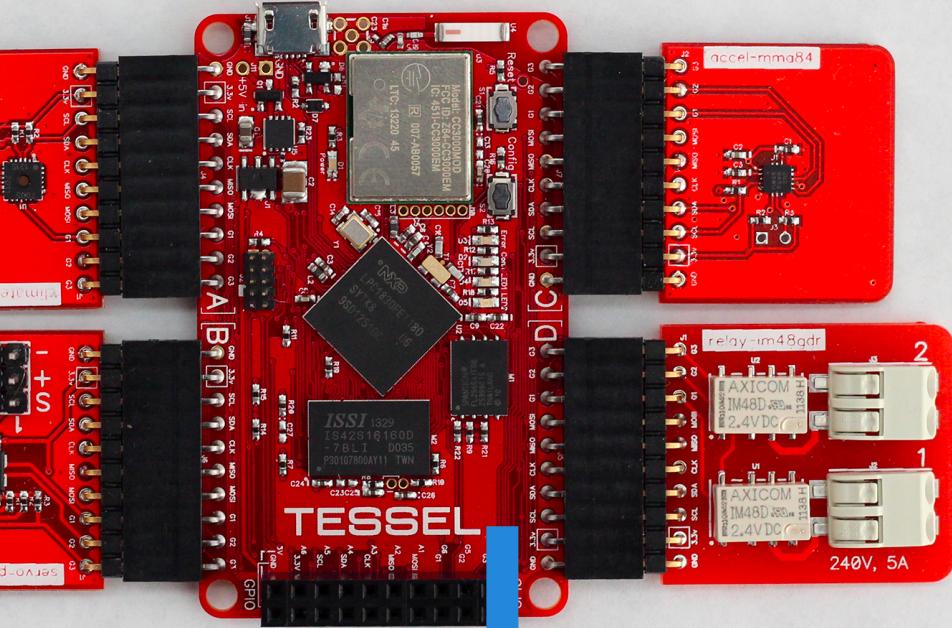


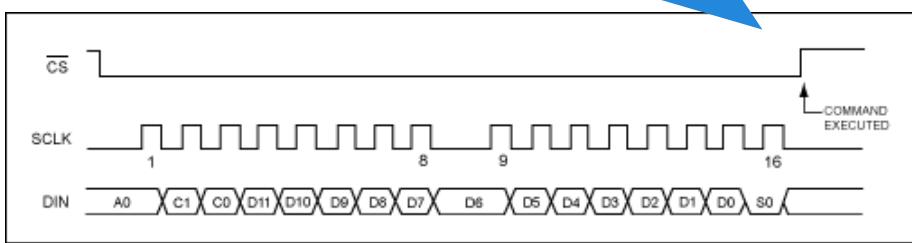
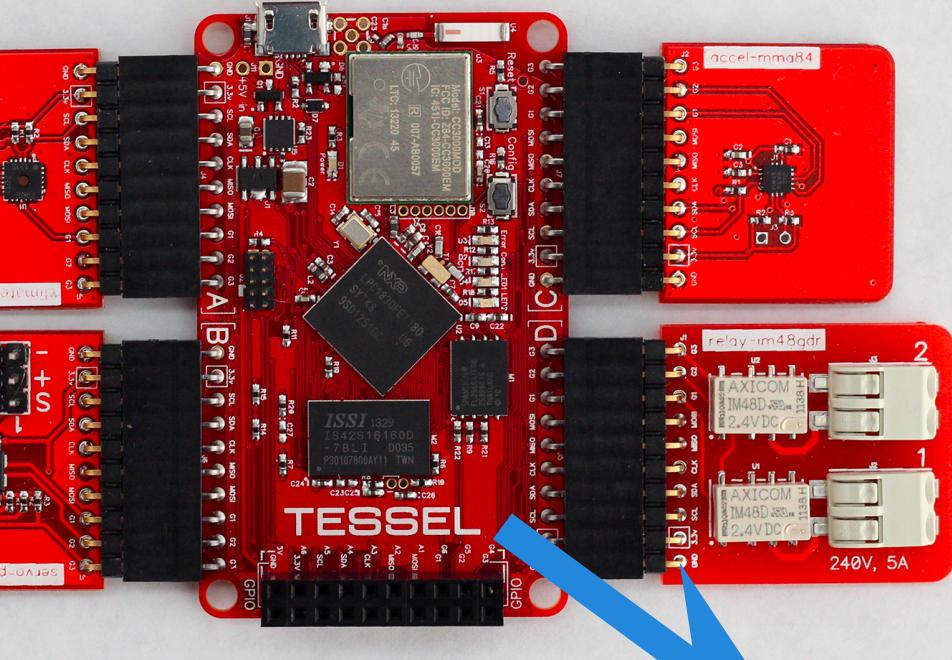
Lesson Learned: When reinventing  
everything, reuse as much as  
possible.

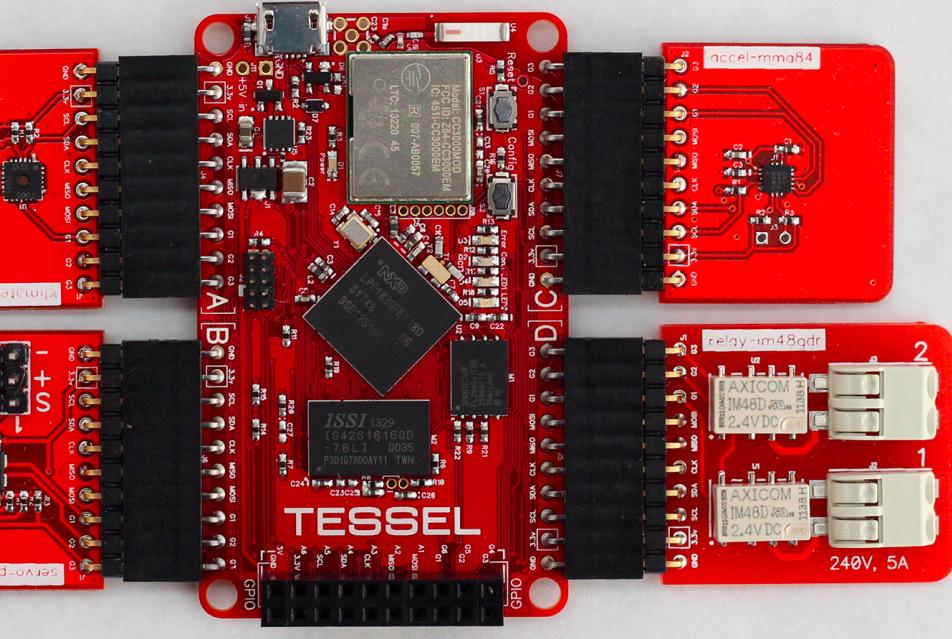










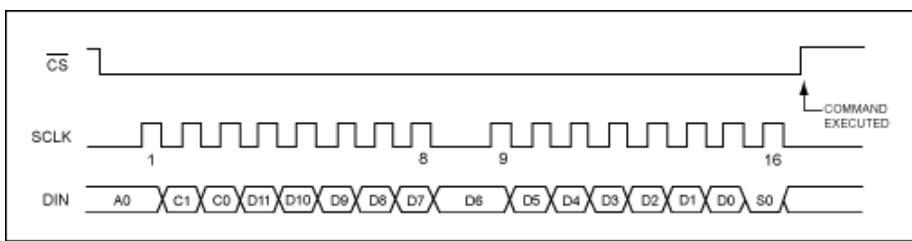


```

def receive_packet():
    buf = sock.recv()
    return buf

while True:
    print(receive_packet())

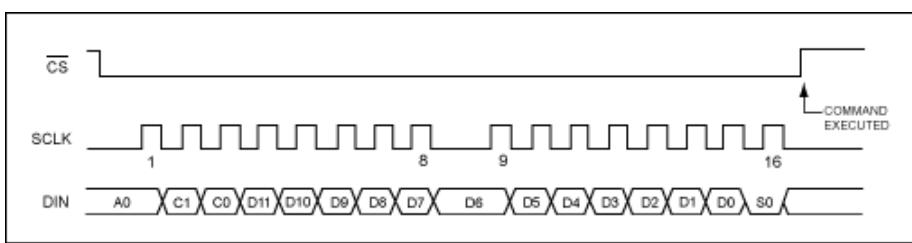
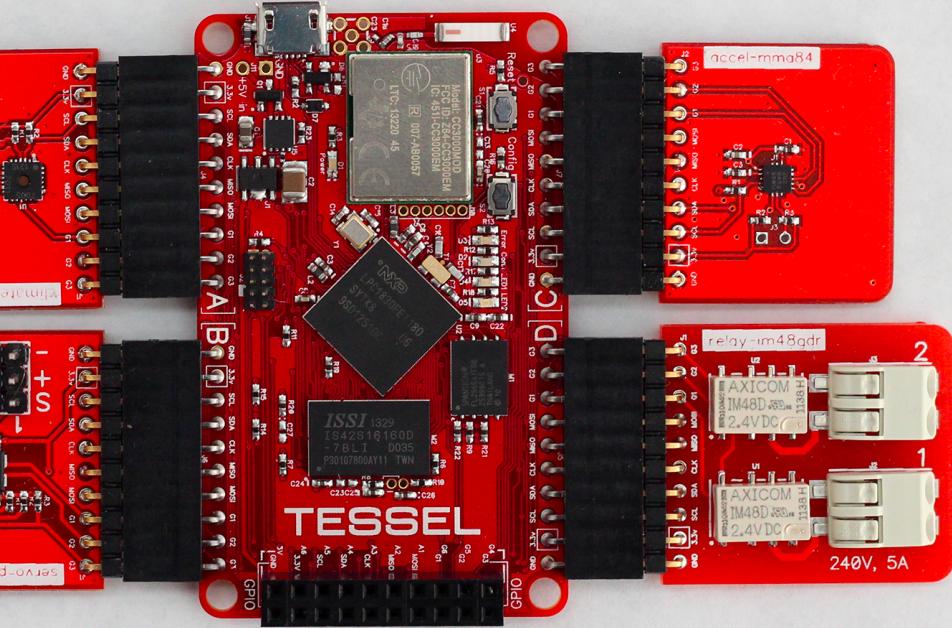
```



JavaScript (and asynchronous  
coding) is the perfect embedded  
language!

JavaScript (and asynchronous  
coding) is the perfect embedded  
language!

— *Alan Turing*



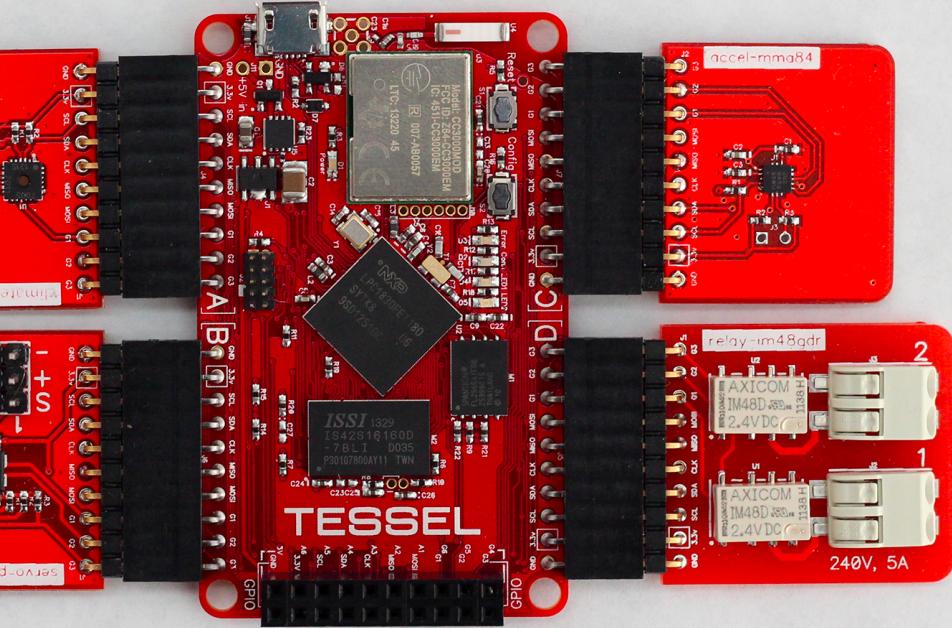
```

Future<byte[]> p =
pool.submit(new Callable<byte[]>() {
    public byte[] call() {
        return socket.recv();
    }
}

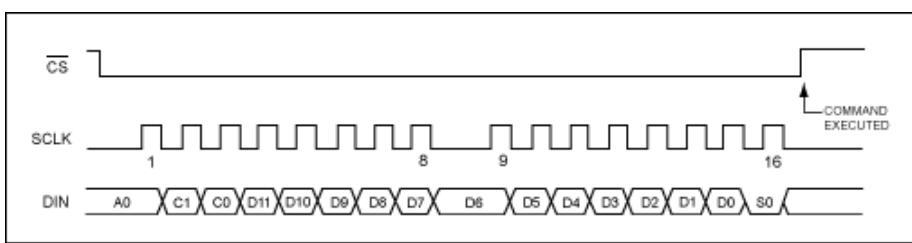
// ... while waiting ...

p.get();

```



```
socket.on('data',
  function (data) {
    // handle data buffer
  });
// ... other code ...
```



# Inventing the Universe



- Embeddable language
- Written in C
- ~30kb memory
- Highly portable
- 10x-100x slower