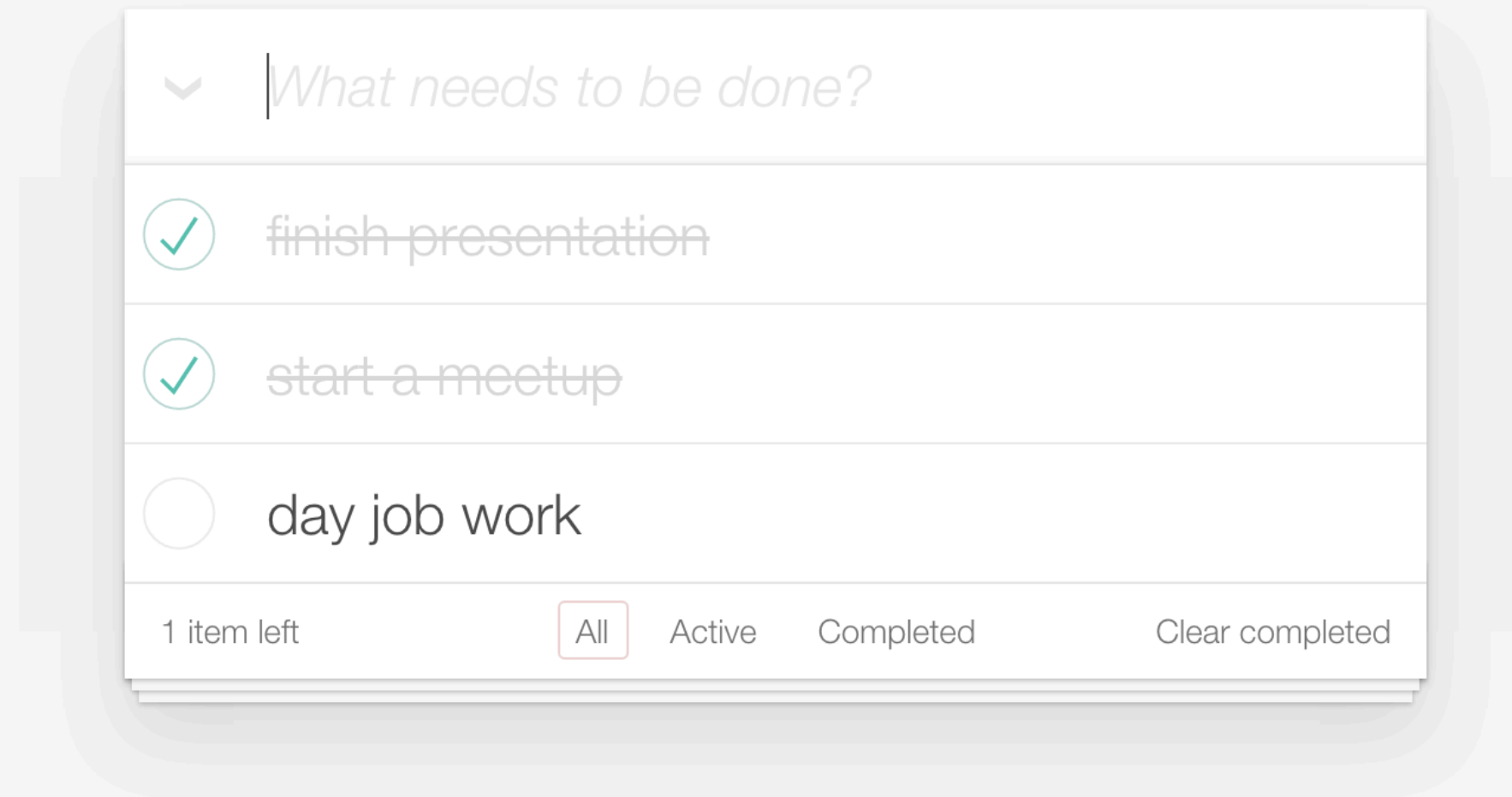


# Missile Commands

MILK  
COMMA

photo of the  
launcher

# todos



Double-click to edit a todo

Created by petehunt

Part of TodoMVC

# Hi I'm Francis



NO FOOD  
TODAY !! ☹️

*Adventure Time's*  
**BACON PANCAKES**

*Warframe's*  
**GREEDY MILK**

FOOD SERVED:  
7-3PM FRIDAY & SAT

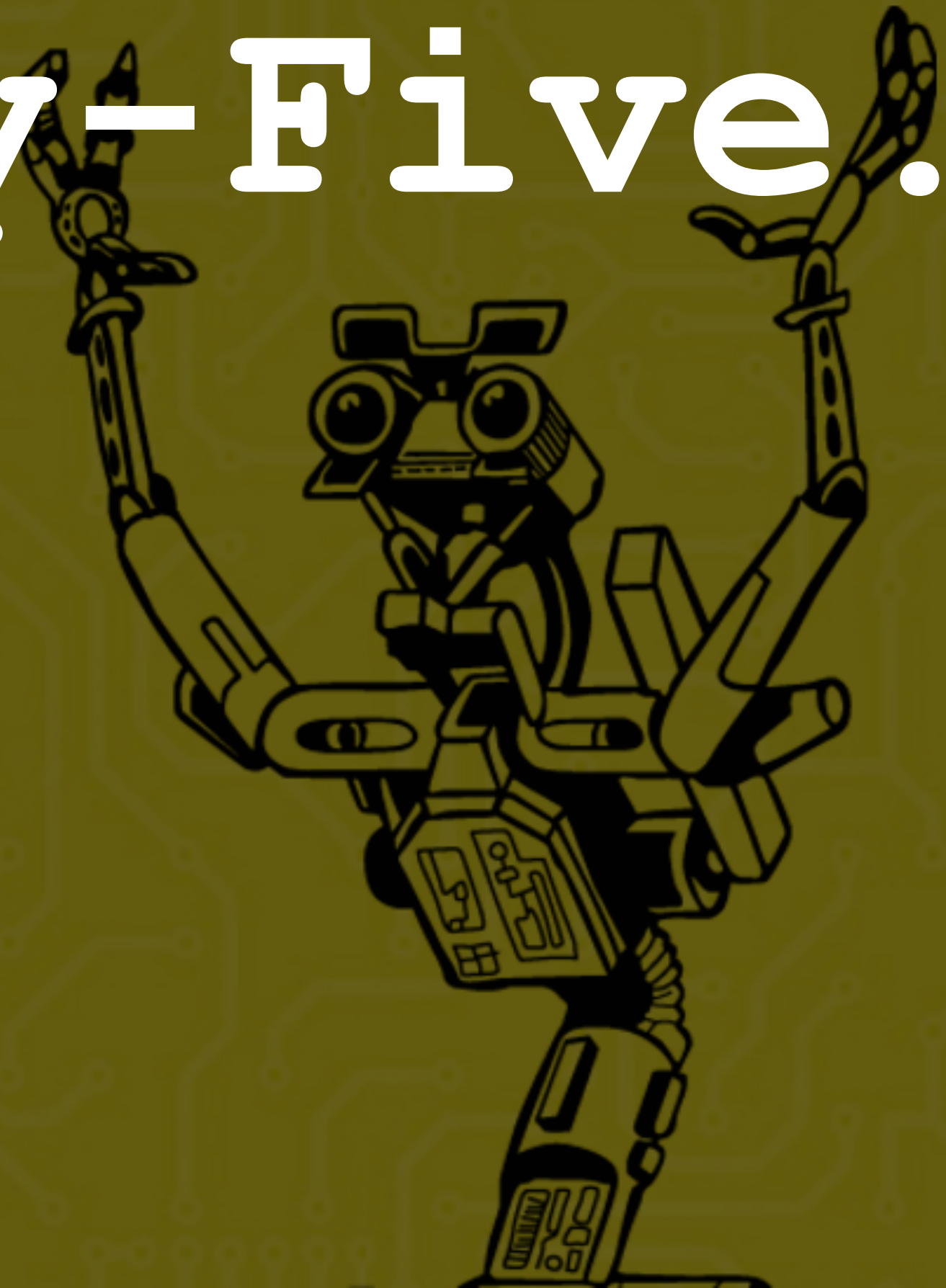
@reconbot





boccoup

Johnny-Five.io





*J5 makes hardware  
development as easy  
as web development*

– Probably Rick Waldron



**Paul Ford**  
@ftrain



Following

after 20 years of being told that web programmers aren't real programmers it's weird to learn that code in cars is garbage

RETWEETS

527

LIKES

597



1:19 PM - 30 Nov 2015

Brooklyn, NY



Reply to @ftrain



**Paul Ford** @ftrain · Nov 30

i just assumed it was all formally-verified Ada done to the highest standards with unbelievably high testing standards



61



122



[View other replies](#)



**Paul Ford** @ftrain · Nov 30

i'm sitting here like an ass using immutable data structures and reading about TLA+ and people at Toyota are like #define MAKE\_CAR\_GO 1



353



516



[View other replies](#)



**Francis Gulotta** @reconbot · Nov 30

@ftrain this is pretty much why I started @bocoup's web connected devices



3





**Paul Ford**

@ftrain



Following

i'm sitting here like an ass using immutable data structures and reading about TLA+ and people at Toyota are like [#define MAKE\\_CAR\\_GO](#) 1

RETWEETS

353

LIKES

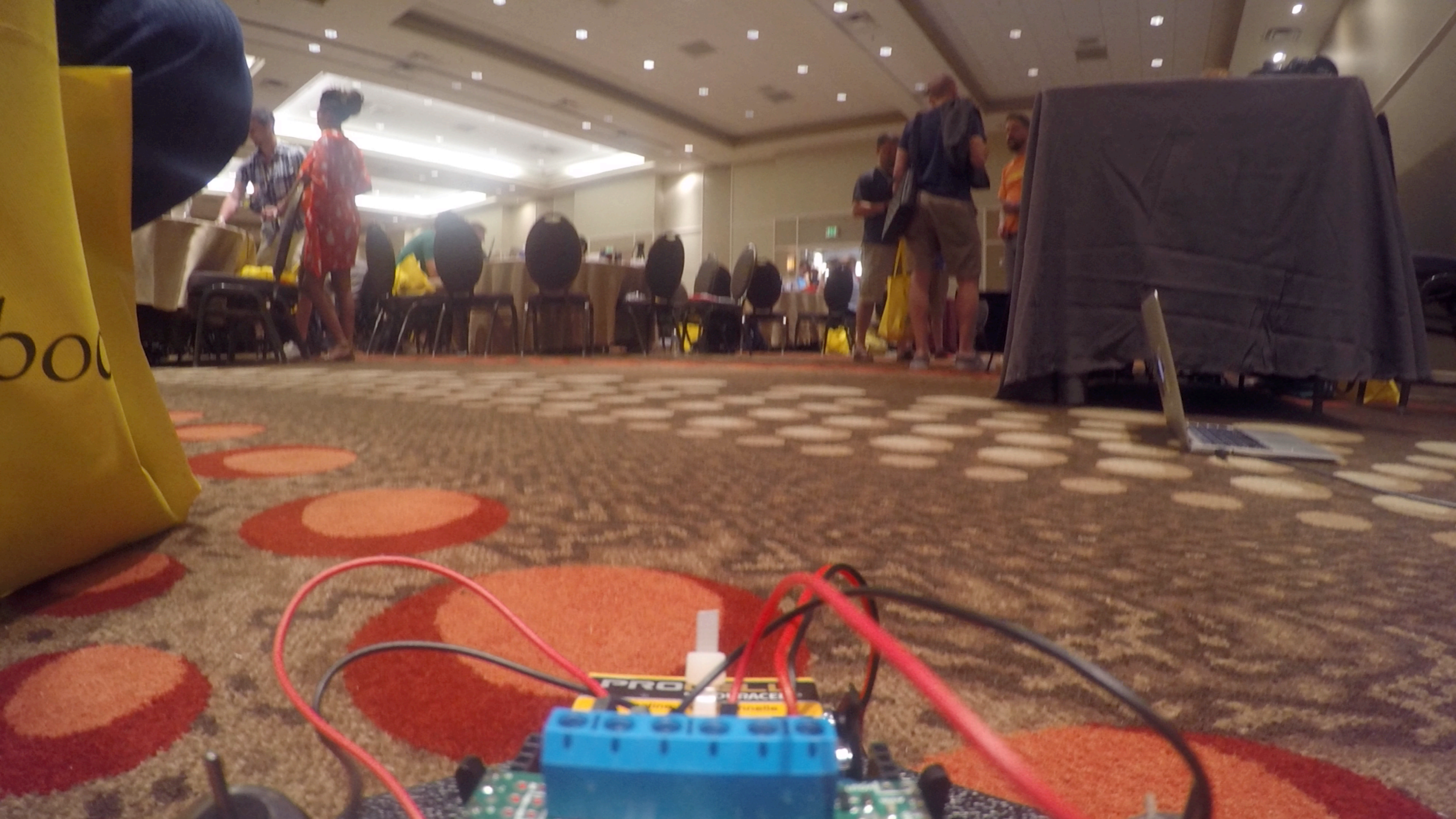
516



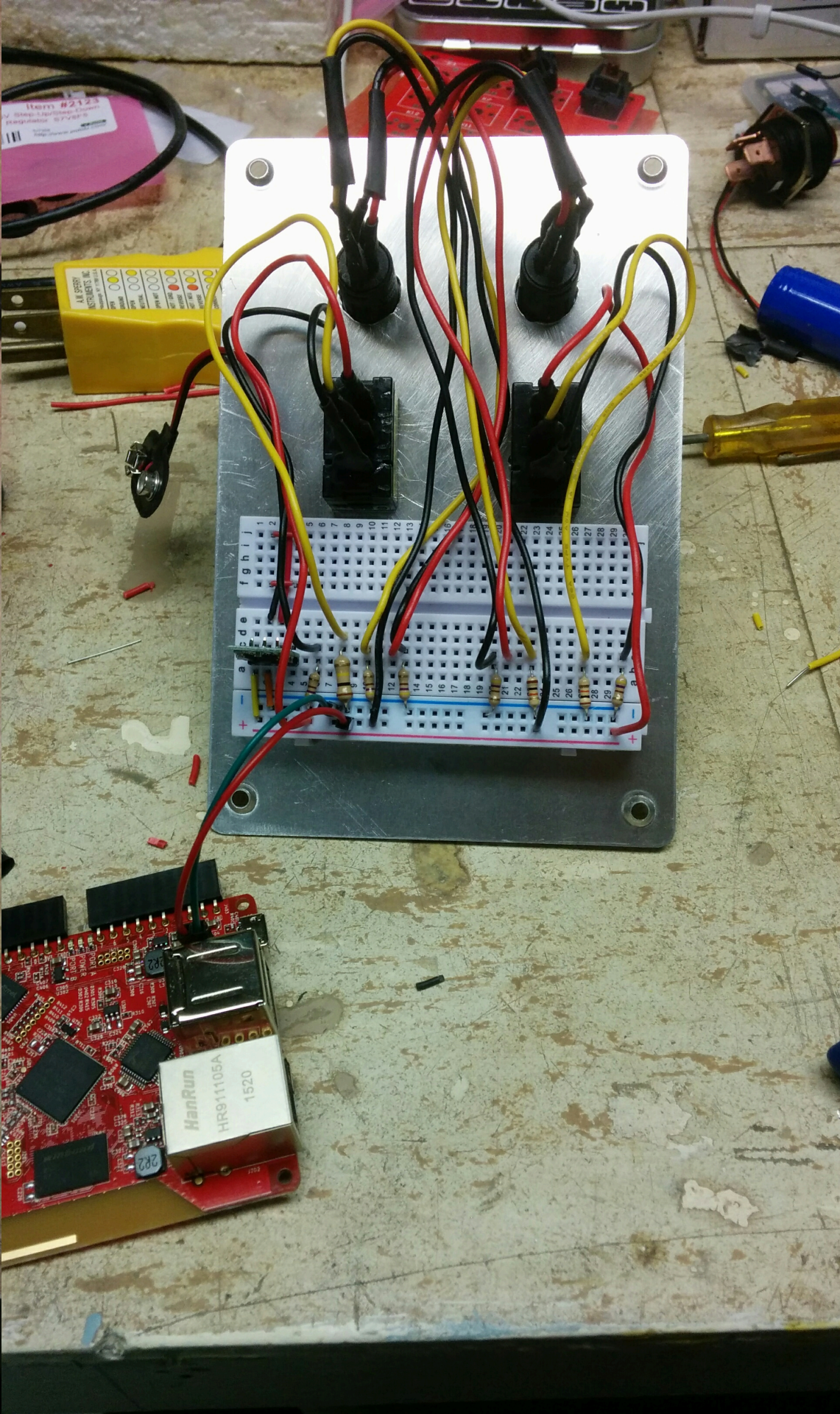
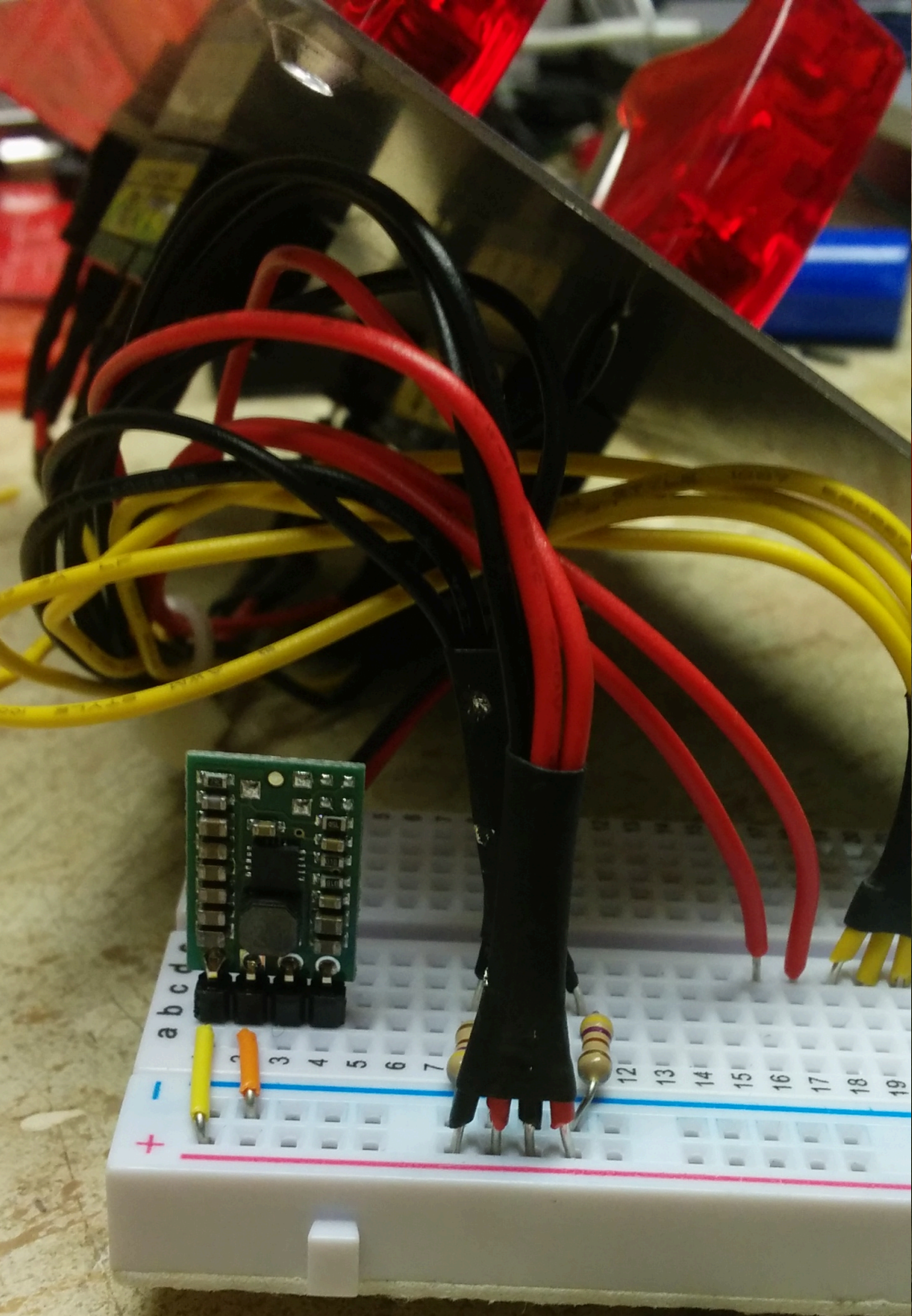
1:23 PM - 30 Nov 2015

📍 Brooklyn, NY





# Hardware Preview



Redux

# Redux

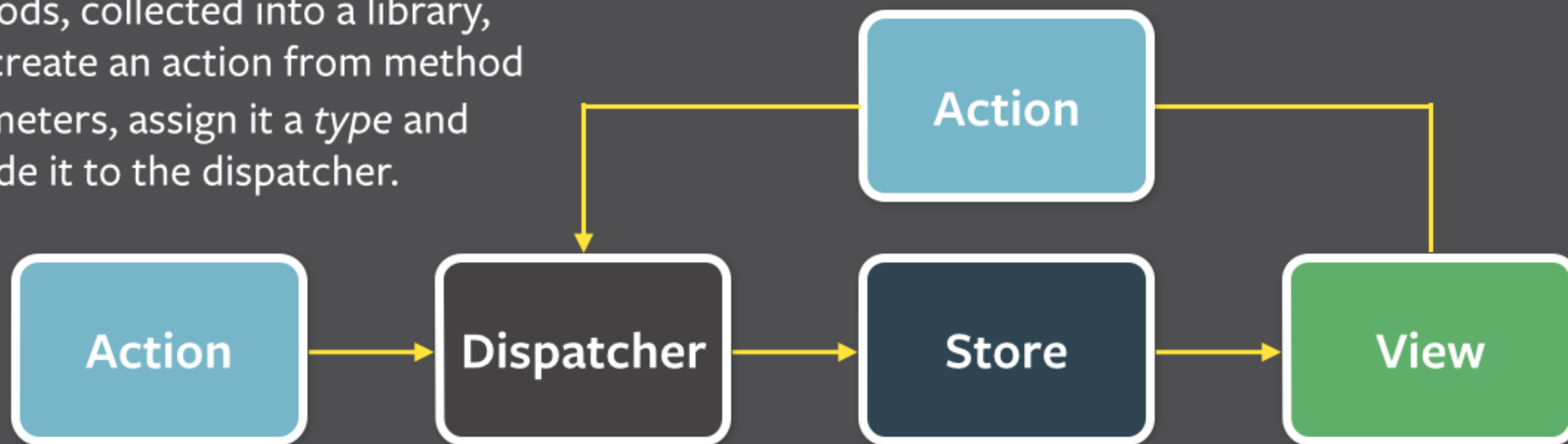
The predictable state container for  
JavaScript apps



# Redux

The *immutable* state container for  
JavaScript apps

*Action creators* are helper methods, collected into a library, that create an action from method parameters, assign it a *type* and provide it to the dispatcher.

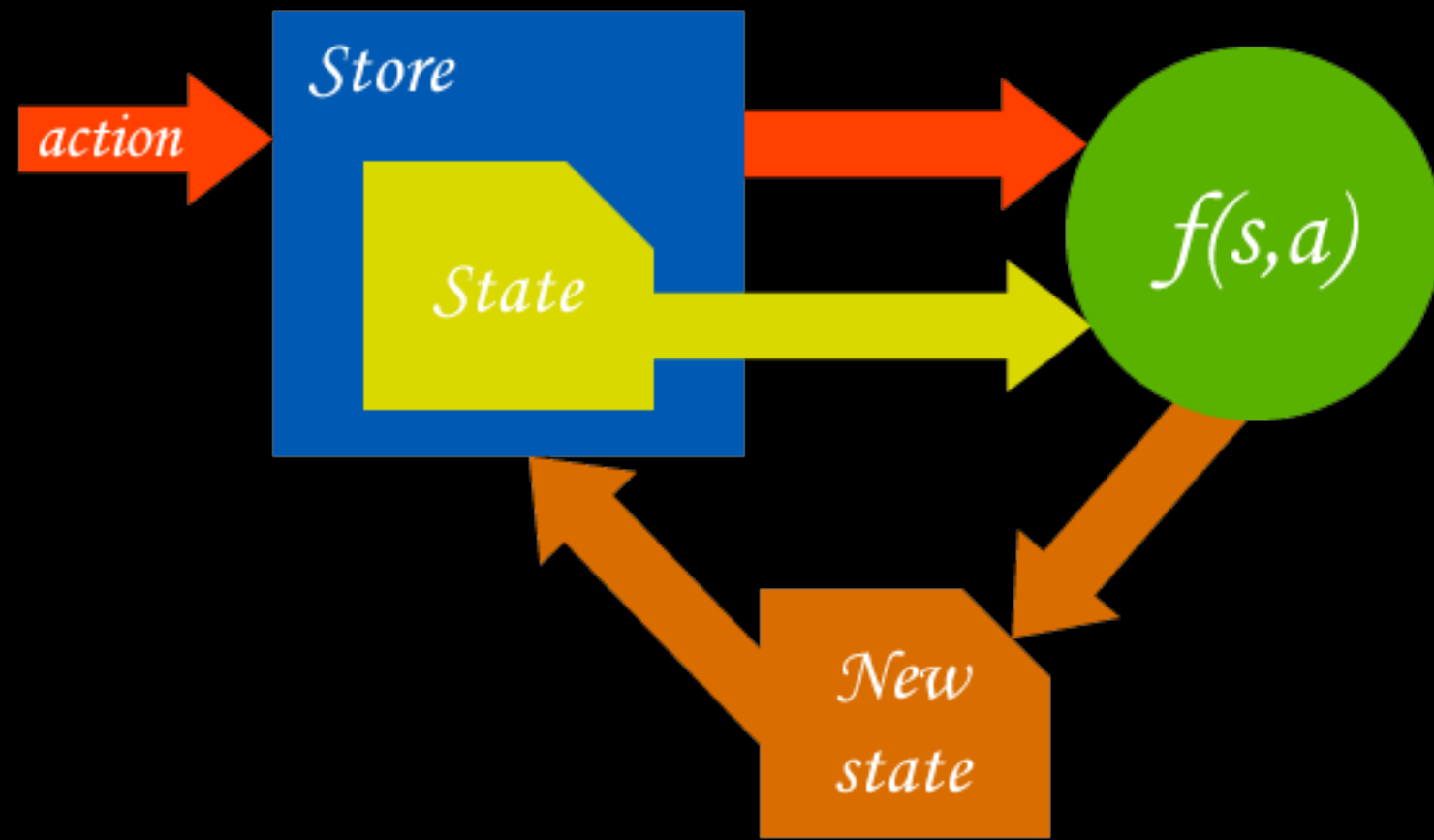


Every action is sent to all stores via the *callbacks* the stores register with the dispatcher.

After stores update themselves in response to an action, they emit a *change* event.

Special views called *controller-views*, listen for *change* events, retrieve the new data from the stores and provide the new data to the entire tree of their child views.

Whatever



# Redux

```
let state = { mode: 'resting' };  
let action = { type: 'launch the missiles!' };  
let newState = reduxMagic(state, action);  
// { mode: 'launching missiles' }
```

```
// React Redux Example
import React, { Component, PropTypes } from 'react'
import { connect } from 'react-redux'
import * as actionCreators from '../actions'
import Armer from './armer';

const Launcher = React.createClass({
  render() {
    const { launch, arm, disarm, status } = this.props
    return (<div>
      <h1>Launcher is {status}</h1>
      <Armer {... { status, arm, disarm }}/>
      <div>
        <button onClick={ launch }>Launch</button>
      </div>
    </div>);
  }
});

export default connect(state => state, actionCreators)(Launcher);
```

# Redux Demo

# Johnny-Five

JavaScript Robotics and IoT programming framework,  
developed at Bocoup.



# Johnny-Five

More than just a framework

# Johnny-Five

The Platform with APIs, JS Libraries, Philosophy,  
Hardware, community for robots and the IoT

# Johnny-Five

```
let button = new five.Button(2);
```

```
button.on("press", () => console.log("Button pressed"));  
button.on("release", () => console.log("Button released"));  
button.on("hold", () => console.log("Button held"));
```

Hold event fires after  
1/2 Second or 500 Milliseconds

50 centiseconds

# Johnny-Five

```
let flipSwitch = new five.Switch(8);
```

```
let led = new five.Led(13);
```

```
flipSwitch.on("open", () => led.off());
```

```
flipSwitch.on("close", () => led.on());
```

# Johnny-Five-Redux

```
let actions = [  
  { type: 'button:press' },  
  { type: 'button:release' },  
  { type: 'button:hold' }  
];
```

# Johnny-Five Callbacks

```
var five = require('johnny-five');
var board = new five.Board();
board.on('ready', function(){
  var armer = new five.Switch(5);
  var launch = new five.Button(6);
  var launchLight = new five.Led(7);
  var launched = true;
  armer.on('close', function(){ launchLight.on(); });
  armer.on('open', function(){
    launchLight.off()
    launched = false;
  });
  launch.on('press', function(){
    if (armer.isClosed && !launched){
      launched = true;
      launchLight.off();
    }
  });
});
```



```
// dispatchEvent = ()=>{};

board.on('ready', ()=>{
  var button = new five.Button(4);
  button.on('press', () => {
    dispatchEvent({
      type: 'button:press'
    });
  })
});
```

.

```
// actionify.js
export default function(device, event, name, callback) {
  device.on(event, (data) => {
    callback({
      type: `${name}:${event}`,
      payload: data
    });
  })
}
```

```
import actionify from 'actionify';

// dispatchEvent = ()=>{};
var button = new five.Button(4);
actionify(button, 'press', 'launcher', dispatchEvent);

// {
//   type: 'launcher:press'
// }
```

```
import { combineReducers } from 'redux'
export default function status(state = 'rest', action){
  switch(action.type){
    case 'armer:closed':
      return (state === 'rest') ? 'armed' : state;
    case 'armer:open':
      return 'rest';
    case 'launcher:press':
      return (state === 'armed') ? 'launched' : state;
    default:
      return state;
  }
}
```

```
// j5-render.js
export default function render(state) {
  switch(state) {
    case 'rest':
      LaunchLight.off(); break;
    case 'armed':
      LaunchLight.on(); break;
    case 'launched':
      launch(); break;
  }
}
```

.

**DEMO TIME**



# Tessel 2 in January

NodeBots .NYC in  
January



Thanks to Polygon for the gifs

<http://www.polygon.com/features/2013/8/15/4528228/missile-command-dave-theurer>

Thanks to Vernimark for the inside missile command images <http://www.vernimark.com/?p=1183>

Thanks to Paul Ford for being angry about cars  
<https://twitter.com/ftrain/status/671393150628438016>

Thanks to Pololu for being awesome <https://www.pololu.com/product/2123>