

# Hack the Mic. IoT with Kinoma

Lizzie Prader  
[lizzie@kinoma.com](mailto:lizzie@kinoma.com)





# The Internet of Things (IoT)

# Egg Minder

THE SMART EGG TRAY



## Most Recent Customer Reviews

★★★★★ Essential in life

I used to walk around asking myself, "How many eggs are in the fridge?" I no longer have to worry, it is always at my fingertips.

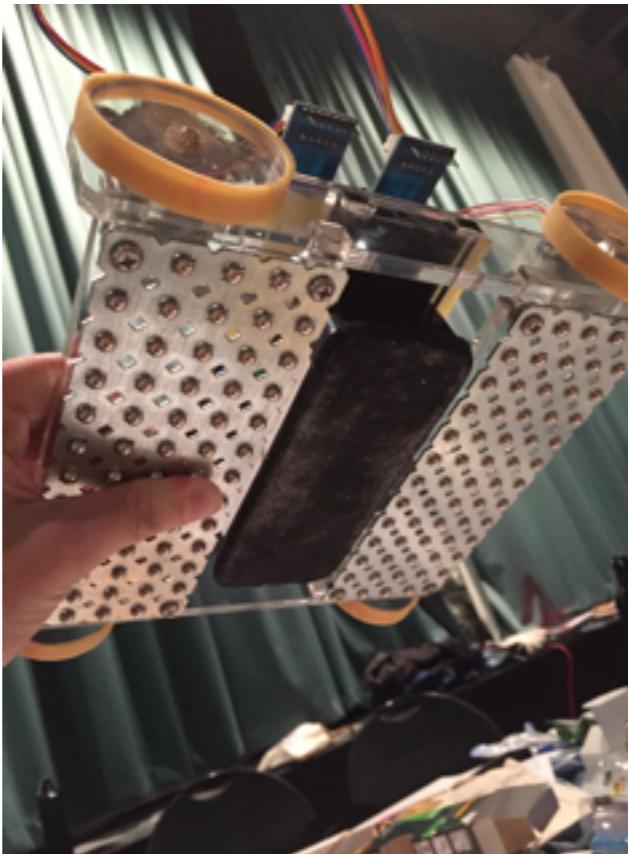
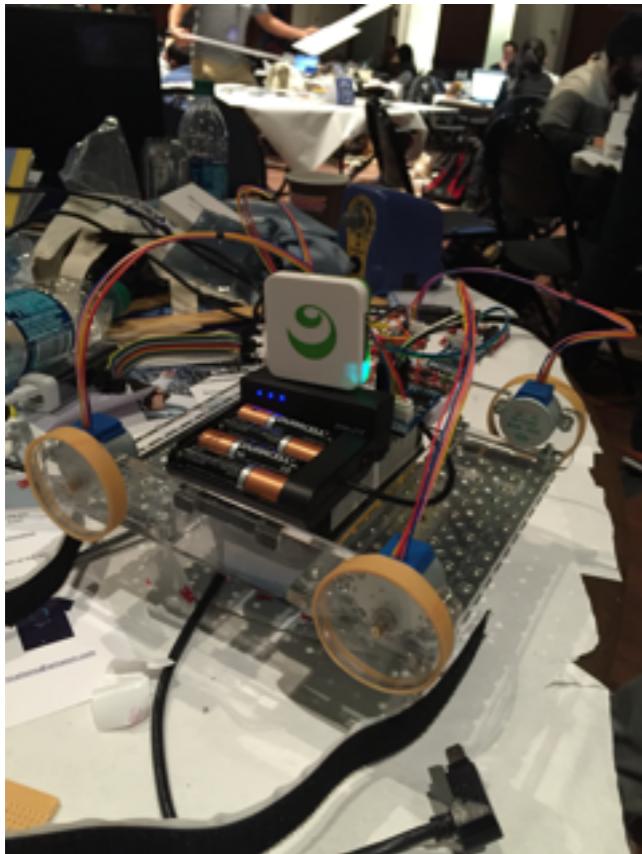
Published 4 days ago by JR



“Never lose your page again”

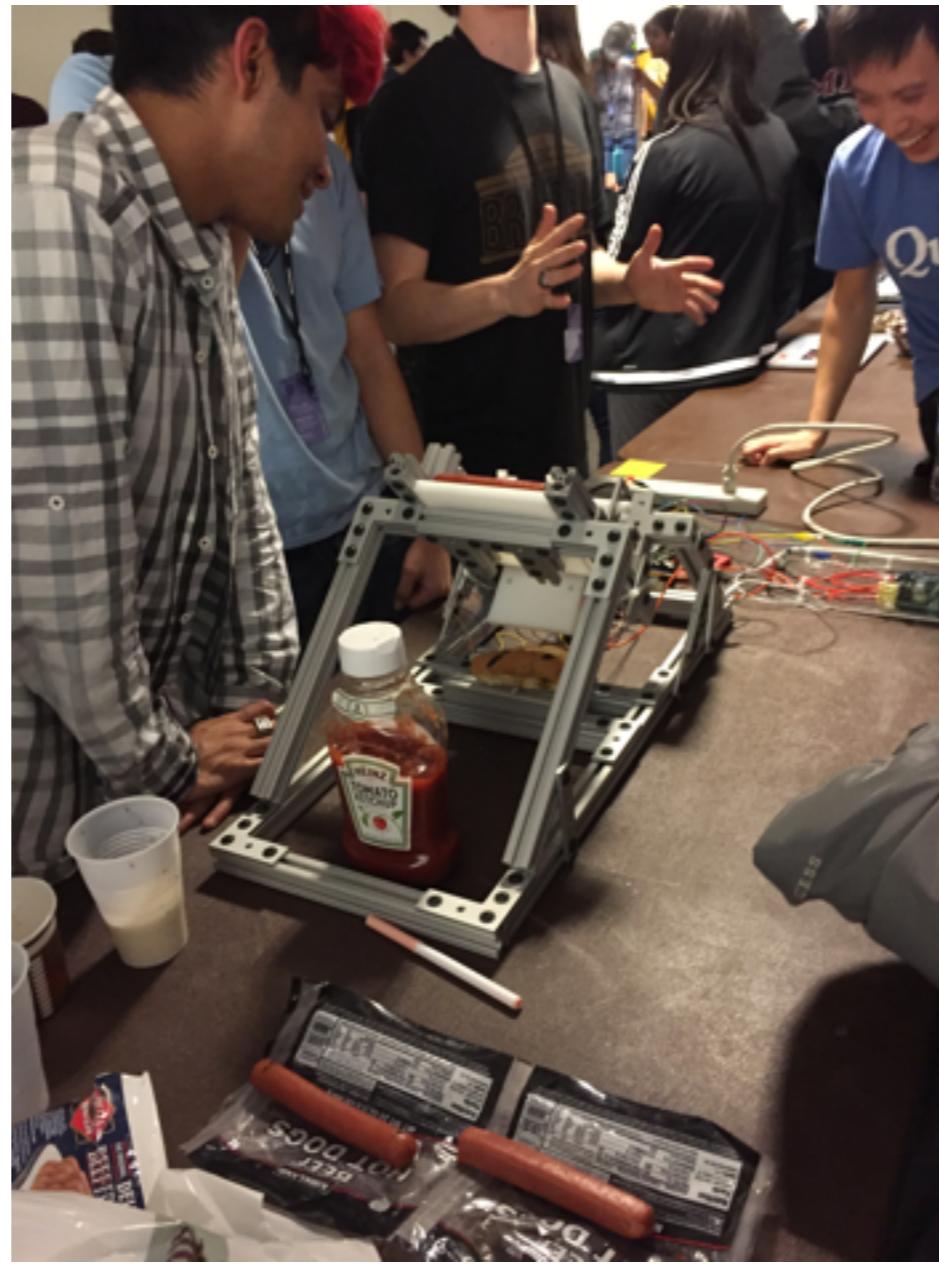
# MakeMIT

© Kinoma



# Making without a cause

© Kinoma



# The Maker Movement

- DIY meets tech
- Computers, robotics, 3-D printing, metalworking, woodworking, etc.

# Maker media

**Make:** Make Magazine



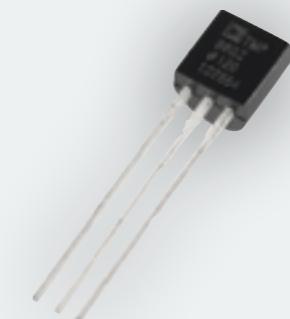
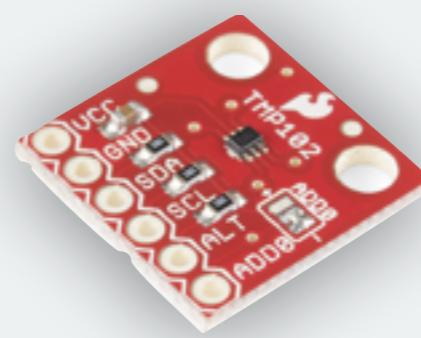
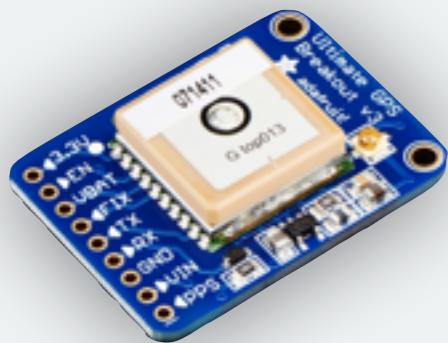
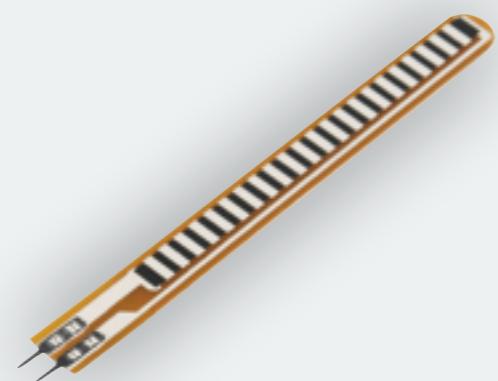
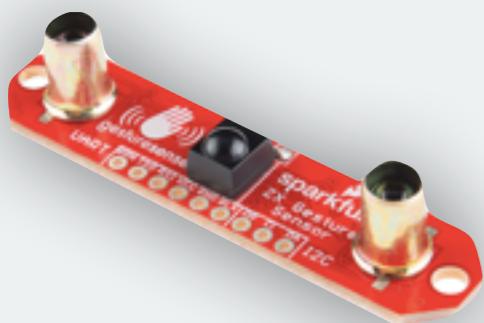
Maker Faires

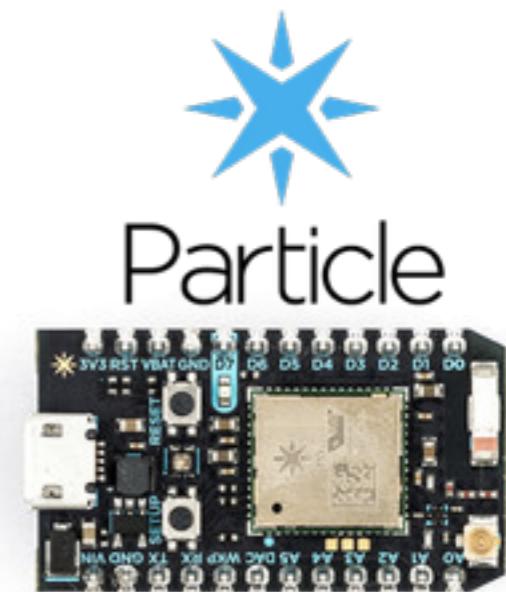
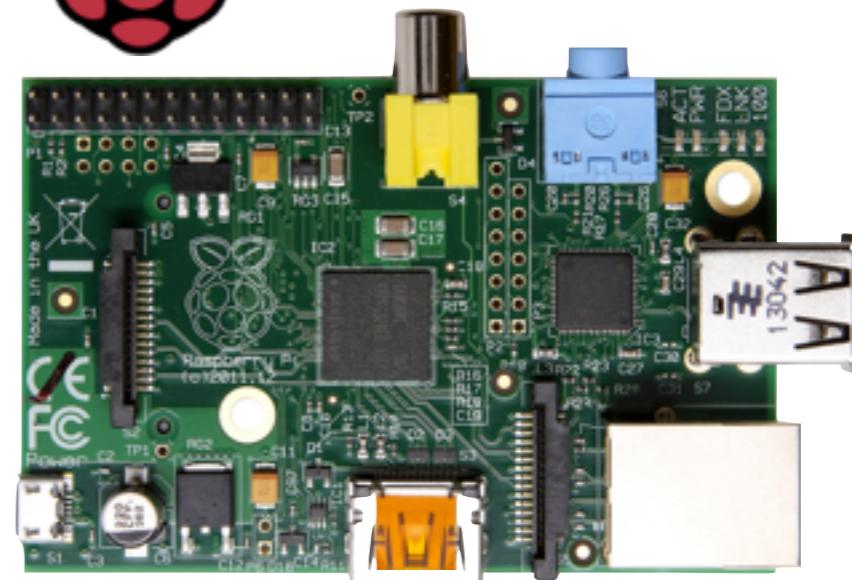


# Noisebridge Hackerspace

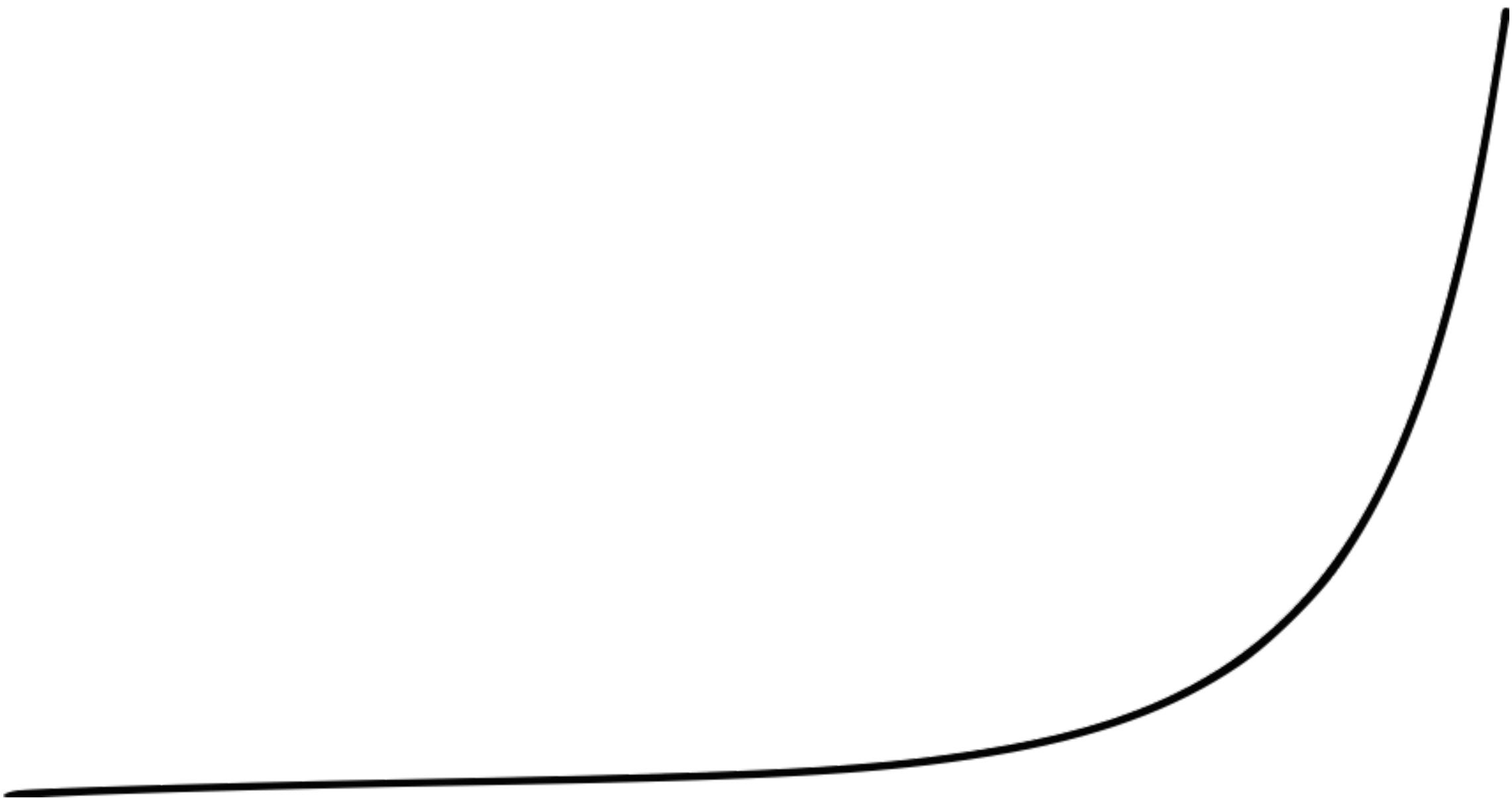
- Beginner friendly events with many materials provided
- [www.meetup.com/noisebridge](http://www.meetup.com/noisebridge)







# Getting started is hard





# Kinoma Element and Kinoma Create

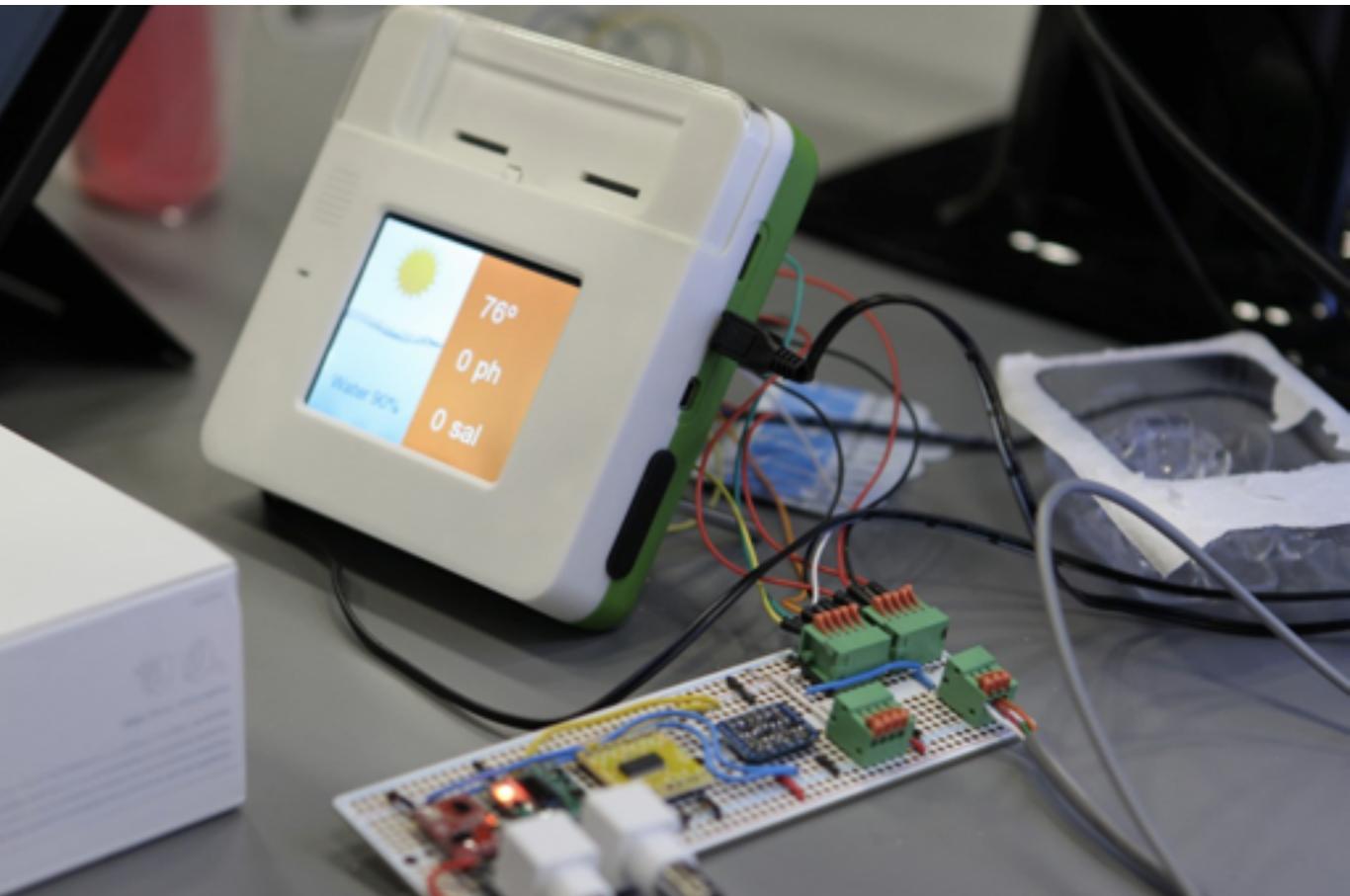


# Not your average prototyping boards



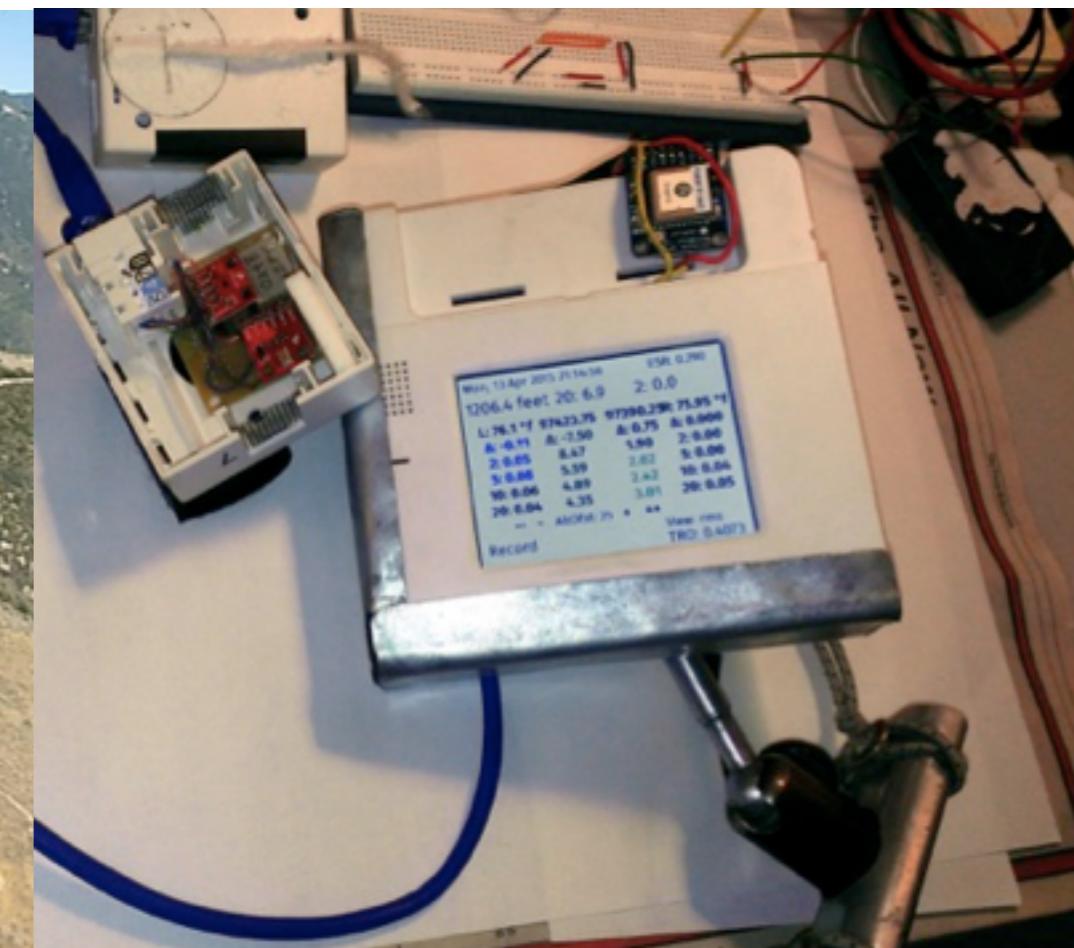
# Aquarium monitor

© Kinoma



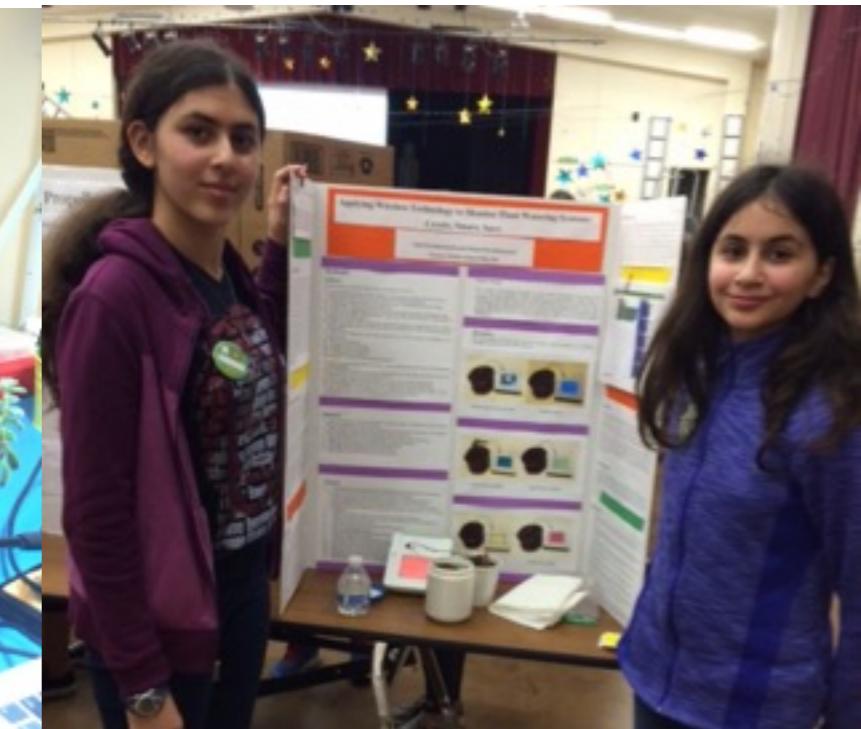
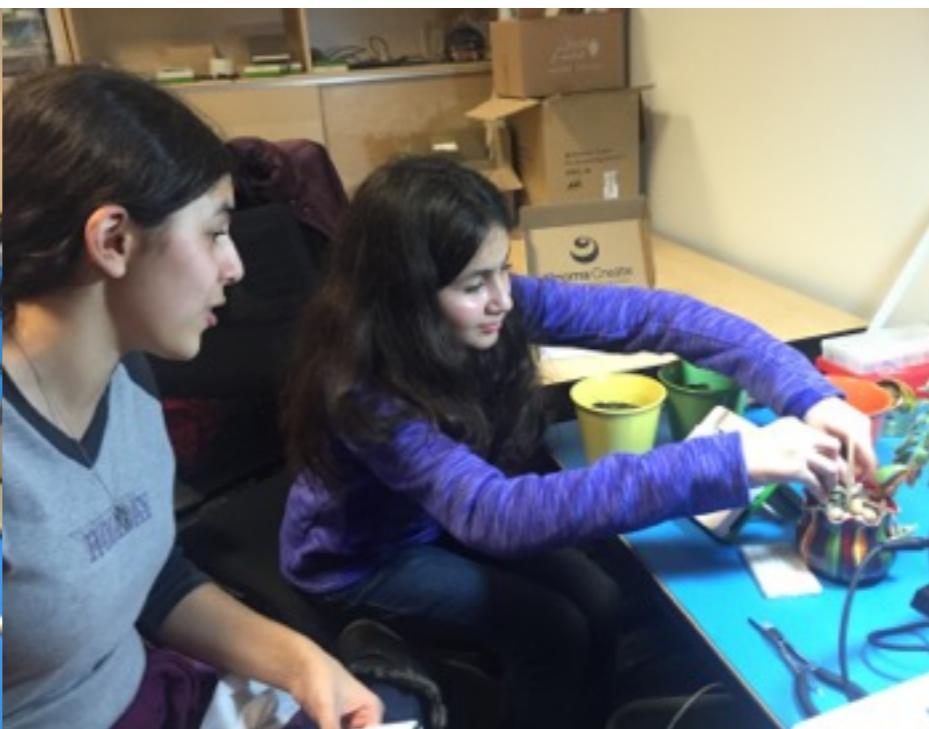
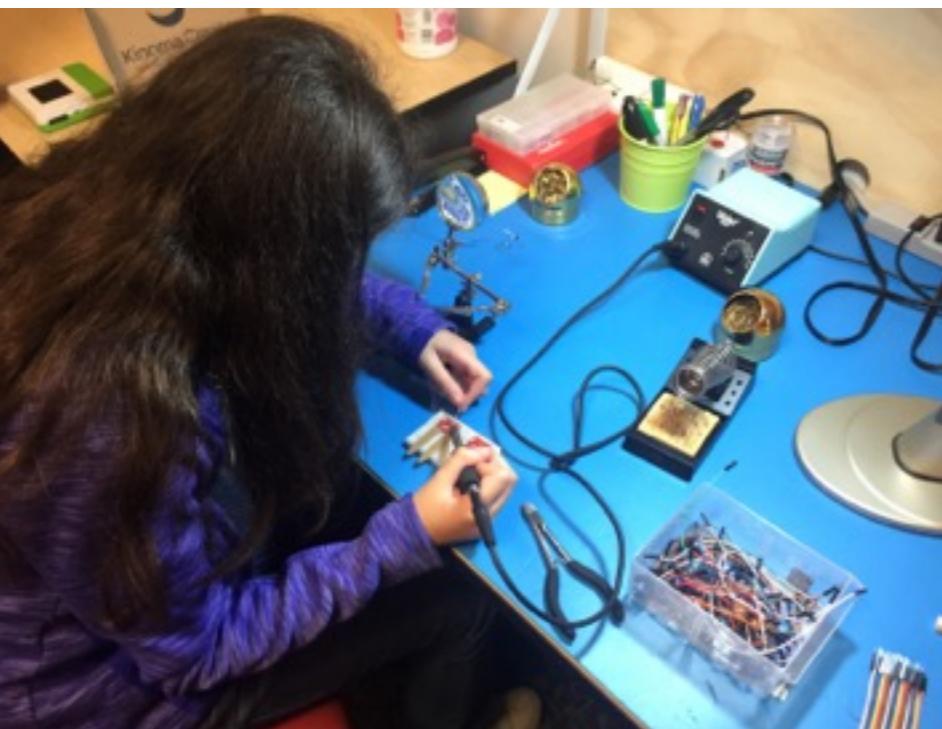
# Hang gliding helper

© Kinoma



# Plant monitor

© Kinoma





(video)



# Tri-color LED

# A note to new makers

- Never view projects at face value
- Prototyping kits let you decide what to make with them

# The Kinoma platform



- Prototyping devices
- Our own programming language
- Tools for developers to program them



# JavaScript has proven to be accessible to designers, students, and engineers

```
var redSkin = new Skin({ fill: 'red', });
var blueSkin = new Skin({ fill: 'blue', });

var labelStyle = new Style({ color: 'white', font: 'bold 36px', horizontal: 'center', vertical: 'middle', });

var MainContainer = Container.template($ => ({
    left: 0, right: 0, top: 0, bottom: 0,
    active: true, skin: blueSkin,
    contents: [
        Label($, { left: 0, right: 0, top: 0, bottom: 0, style: labelStyle, string: $.text })
    ],
    behavior: Behavior({
        onTouchBegan: function(container, id, x, y, ticks) {
            container.skin = redSkin;
        },
        onTouchEnded: function(container, id, x, y, ticks) {
            container.skin = blueSkin;
        }
    })
}));
```

```
application.add( new MainContainer({ text: "Hello, KPR" }));
```

# 4 hardware trends to watch in 2016

Early signals of what's to come in the hardware world.

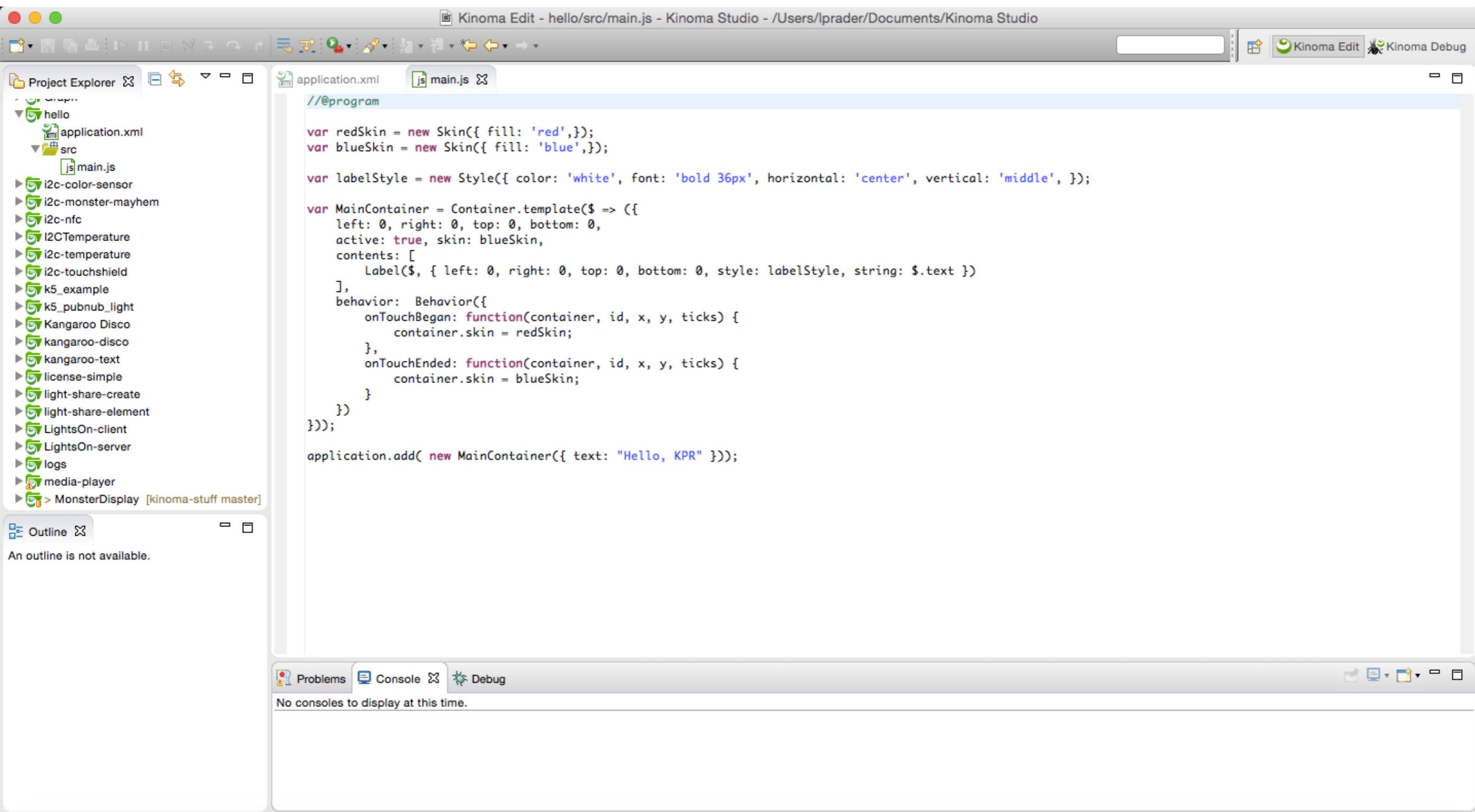


O'REILLY®

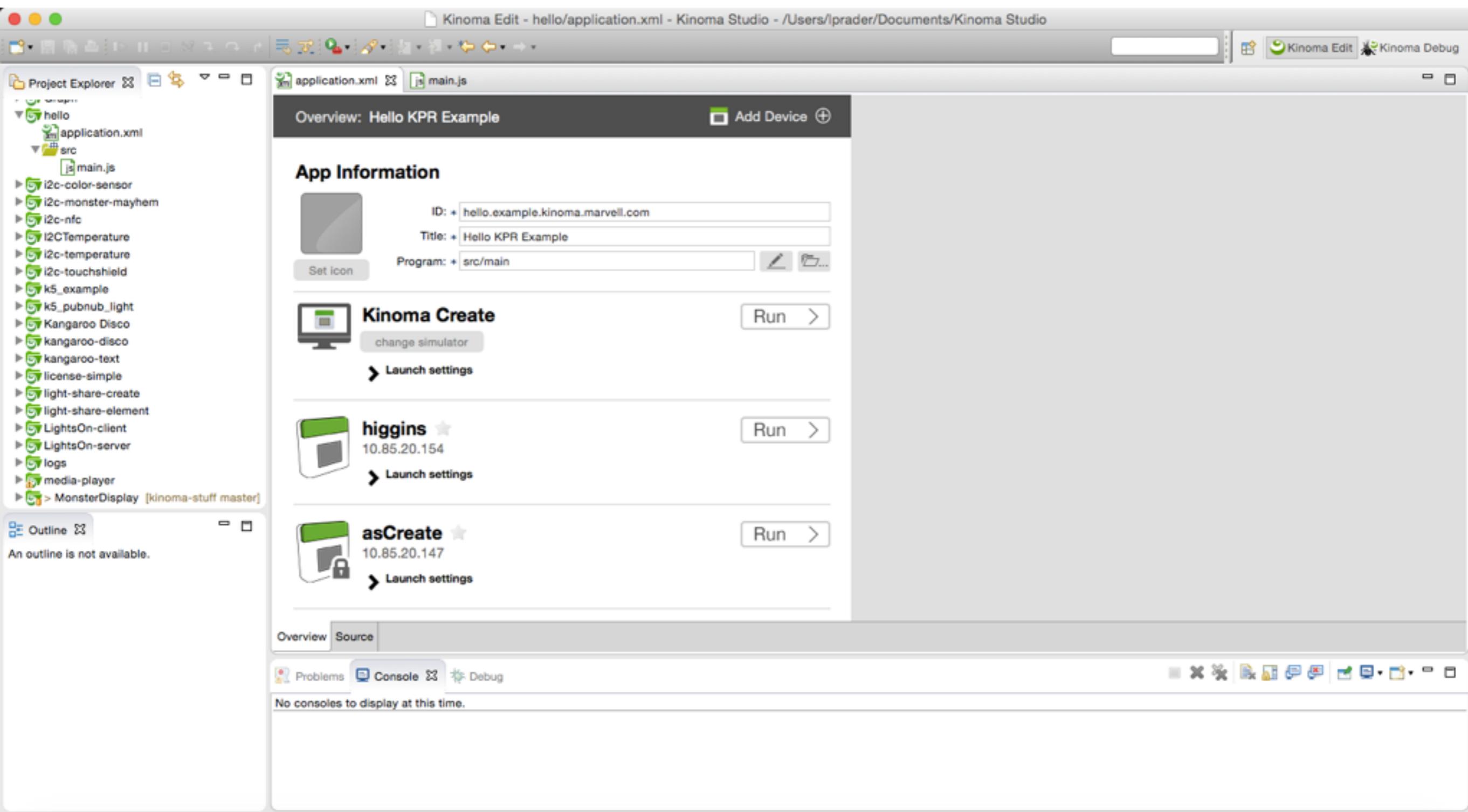
## **High level programming languages on embedded systems**

Relatedly, writing software to control drones, vending machines, and dishwashers has become as easy as spinning up a website. Fast, efficient processors ... are turning JavaScript into a popular embedded programming language—unthinkable less than a decade ago.

# Kinoma Studio



# Kinoma Studio



# KinomaJS Blocks

KinomaJS Blocks

View JavaScript Source    BlocklyApp    target IP    RUN >

Blockly

Logic  
Loops  
Math  
Text  
Lists  
Colour  
Variables  
Functions  
KinomaJS  
UI  
Pins  
Time  
Sound  
Advanced

screen [Screen1] default colour

onCreate

- label [title1] text ["Last Received Message: "] text colour black left 10 top 5
- label [receivedMessageLabel] text ["(nothing received)"] text colour black left 10 top 45
- rectangle [blueBackground] colour blue width 360 height 160 left 0 top 95
- label [title2] text ["Send: "] text colour black left 10 top 110
- button [hiButton] text ["Hi!"] colour green left 85 top 105

onTouch

offTouch sendMessage with: message "Hi!"

button [smileButton] text [" :) "] colour green left 140 top 105

onTouch

offTouch sendMessage with: message " :) "

button [bigSmileButton] text [":D"] colour green left 180 top 105

onTouch

offTouch sendMessage with: message ":D"

button [WOWbutton] text ["WOW"] colour green left 225 top 105

onTouch

offTouch sendMessage with: message "WOW"

server serveFunctions serve function receiveMessage

to receiveMessage with: message

- write "light" with 1
- set receivedMessageLabel text to message
- timeout timer on 1000 ms
- onTimeout write "light" with 0

pins

onCreate

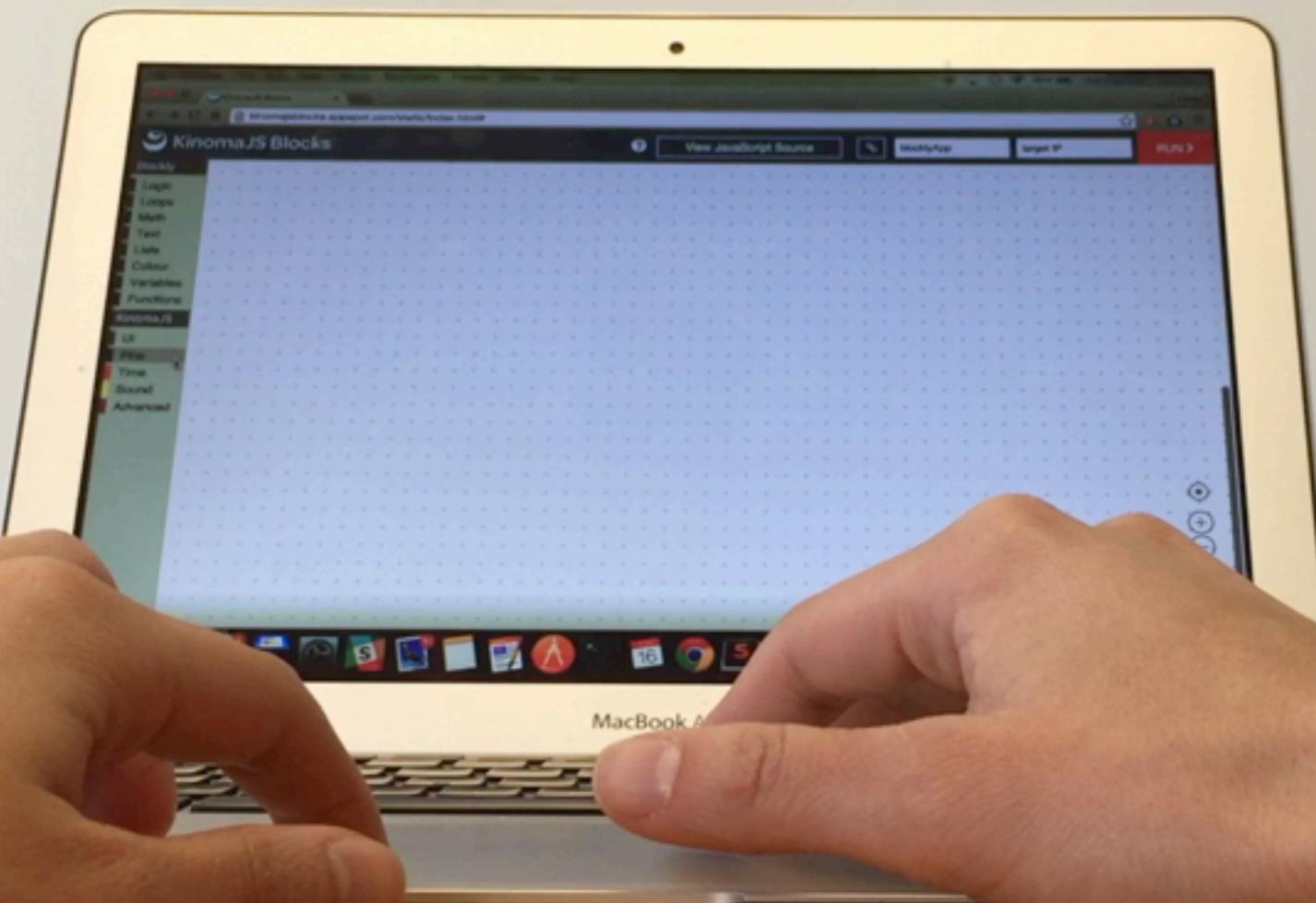
- make ground on pin 51 with name "ground"
- make digital out on pin 52 with name "light"

onReady

to sendMessage with: message

- set x to message
- call receiveMessage with: x at "PARTNER\_IP\_ADDRESS"

The screenshot shows the KinomaJS Blocks interface. The workspace contains several blocks: a screen block with a title1 label, a rectangle block for a blue background, three buttons labeled 'Hi!', ' :) ', and ':D', and a 'WOW' button. On the right, there's a server block for 'receiveMessage' and a pins block for 'sendMessage'. A large block on the right is a 'to receiveMessage with: message' loop containing a 'write' block, a 'set' block for the 'receivedMessageLabel', a 'timeout' block, and an 'onTimeout' block. Another block below it is a 'pins onCreate' block for setting up pins 51 and 52. At the bottom, there's a 'sendMessage' block with a 'call receiveMessage' sub-block.



(video)

# KinomaJS Blocks

© Kinoma

- As easy as it gets
- No downloading
- No wires



+



+



# Everyone can be a maker



# Thank you!



[kinoma.com](http://kinoma.com)



@kinoma



kinoma



kinomahq



kinoma



# Questions?

 kinoma.com

 @kinoma

 kinoma

 kinomahq

 kinoma

