

An IoTa of IoT: Micro:bit Magic & Photon Phun!

Bill Siever
Washington University in St. Louis

Michael Rogers
Northwest Missouri State University

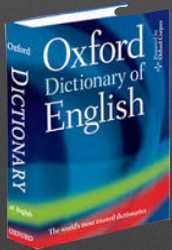
Outline

- Intros: Us, You
 - ★ Motor Mayhem or Awesome Audio
- Background, Motivations, Considerations, & Platforms
 - Quick Overview: IoT Insanity
- micro:bit Magic
 - ★ Setup & Dev. Environment
 - ★ “Hello, World!”: First Program
 - ★ Broadcast Basics & Firefly Fun
- Photon Phun
 - ★ Blinky & Remote Lights
 - ★ Buttons & Variables
 - ★ Apps & Real IoT

• Intros: Us & You

- Us
- You: Roll Call & Intros
 - Pair programming —pair up!

Background



- A network (internet) of sensor-, actuator- and software-equipped devices (things) that share information among themselves as they scheme to take over the world (we may need to work on this last bit) 🙄
- Home: Lights, Thermostat, Smoke Alarms, Security Systems, Internet of Things Doorbell 😊



Motivation

- Provides a unique outlet for creativity! Students can make something new.
- Fun, maker-like experience with tangible artifacts.
- Accessible multi-tier view of a complex, real-world systems.
- Jobs & industry Needs

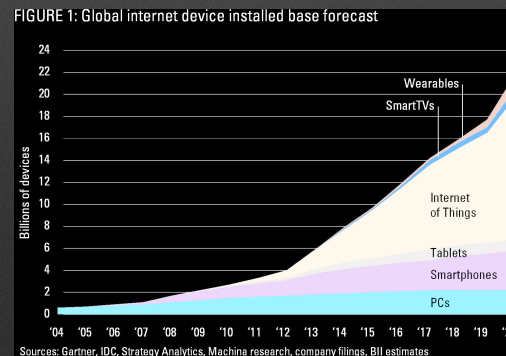
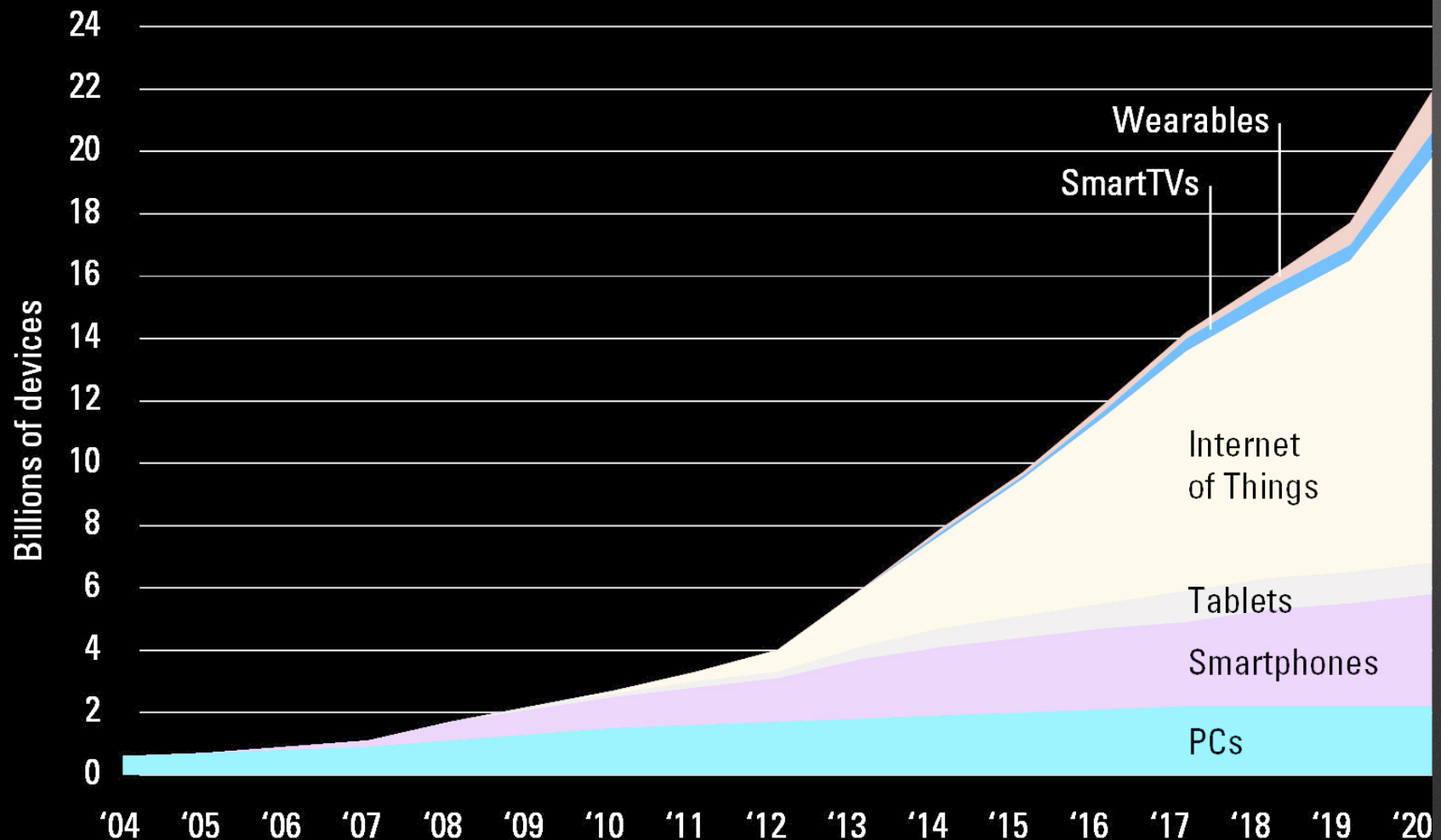


FIGURE 1: Global internet device installed base forecast

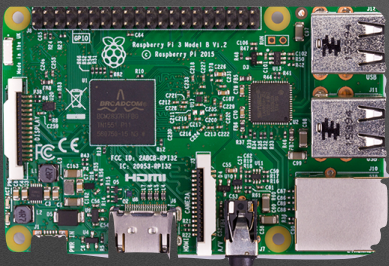


Sources: Gartner, IDC, Strategy Analytics, Machina research, company filings, BII estimates

• Considerations

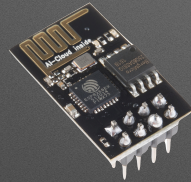
- What are your objectives?
- Who's your audience?
- Theoretical? Applied? A hybrid?
- Depth or breadth?

Plethora of Platforms



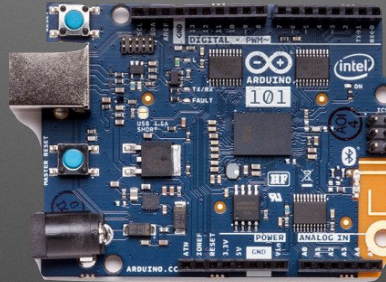
Raspberry Pi

\$16; Wi-Fi;Wiring



ESP8266

\$<16; Wi-Fi;Wiring



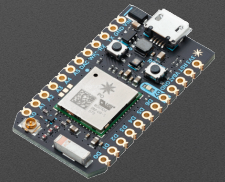
Arduino Variants

\$~8-80; Wi-Fi, BLE;Wiring



micro:bit

\$13; BLE

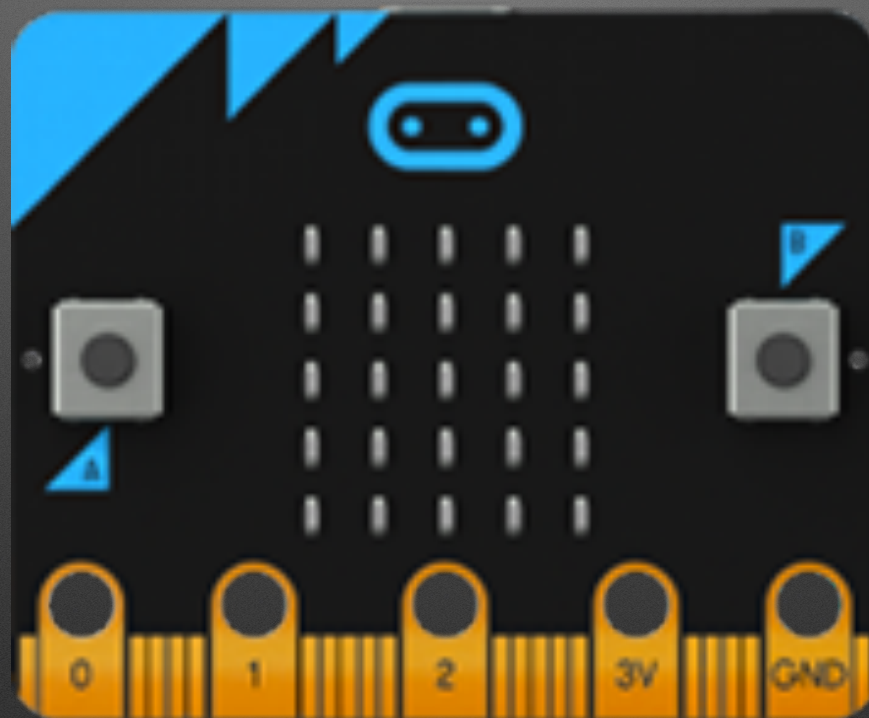


Photon

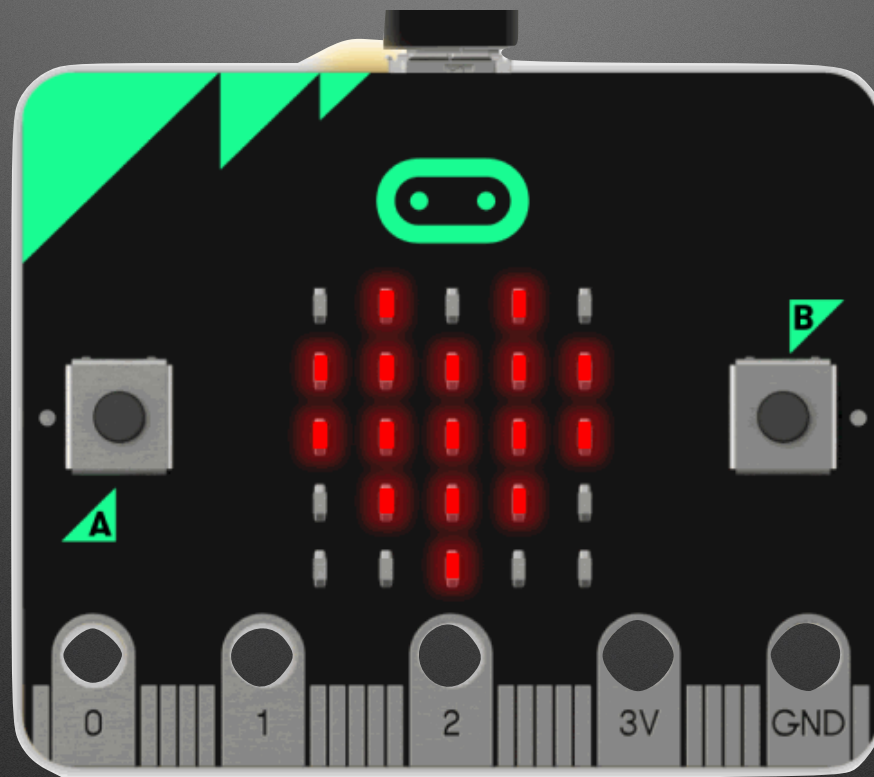
\$19; Wi-Fi;Wiring

Intros: the micro:bit

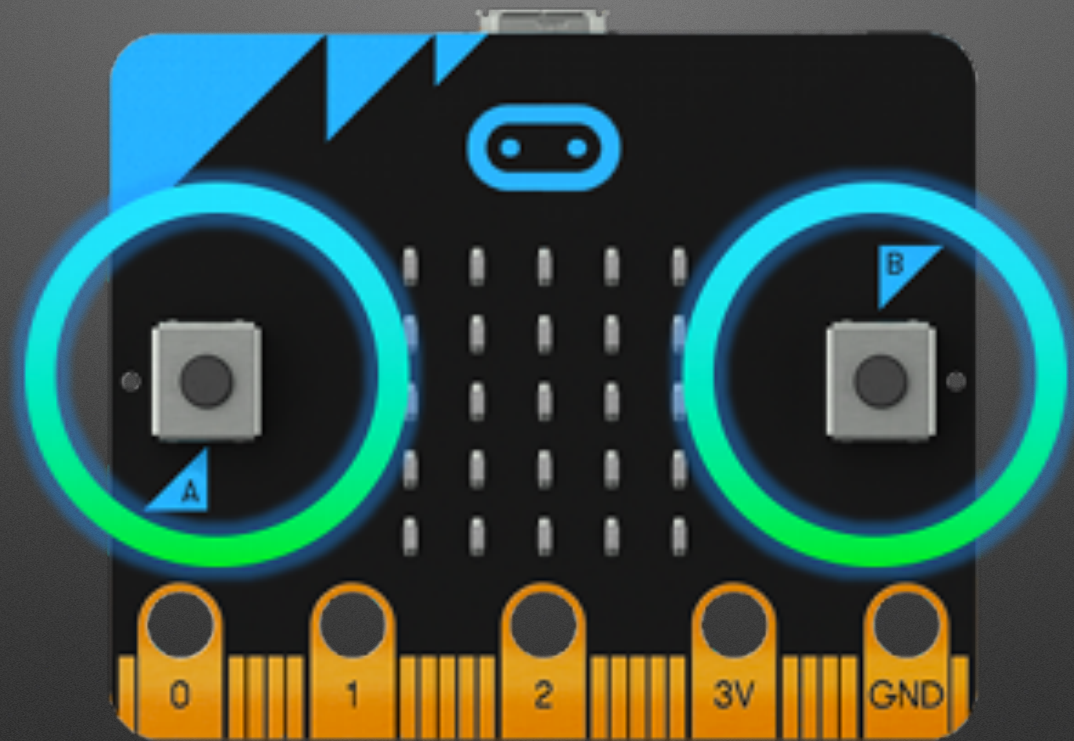
Small



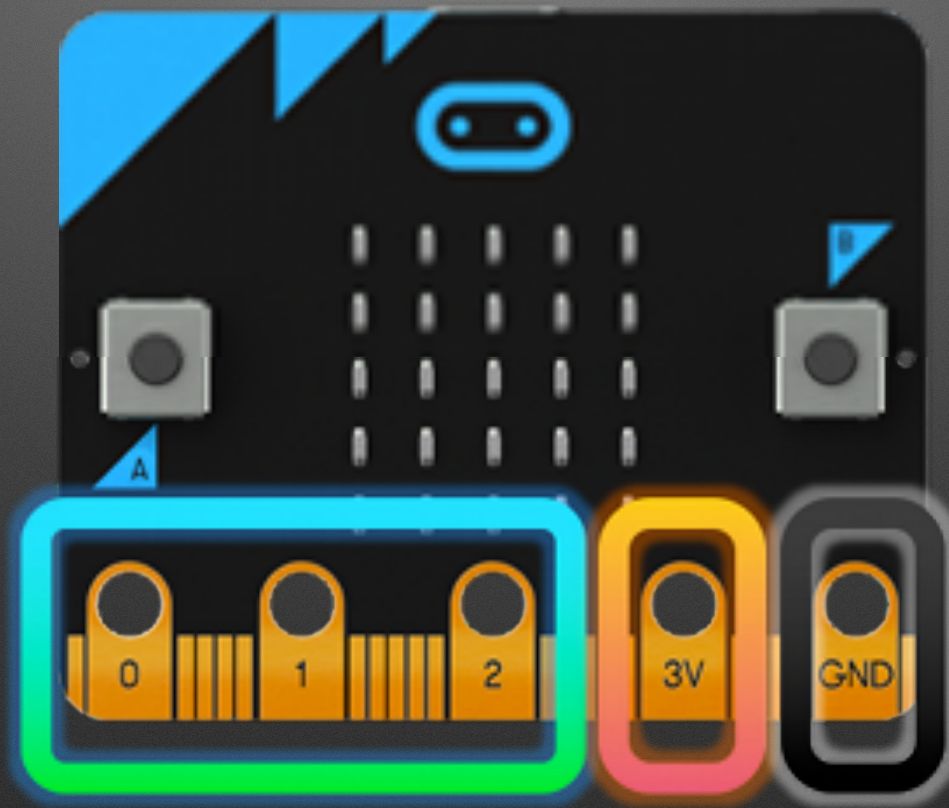
LED Grid



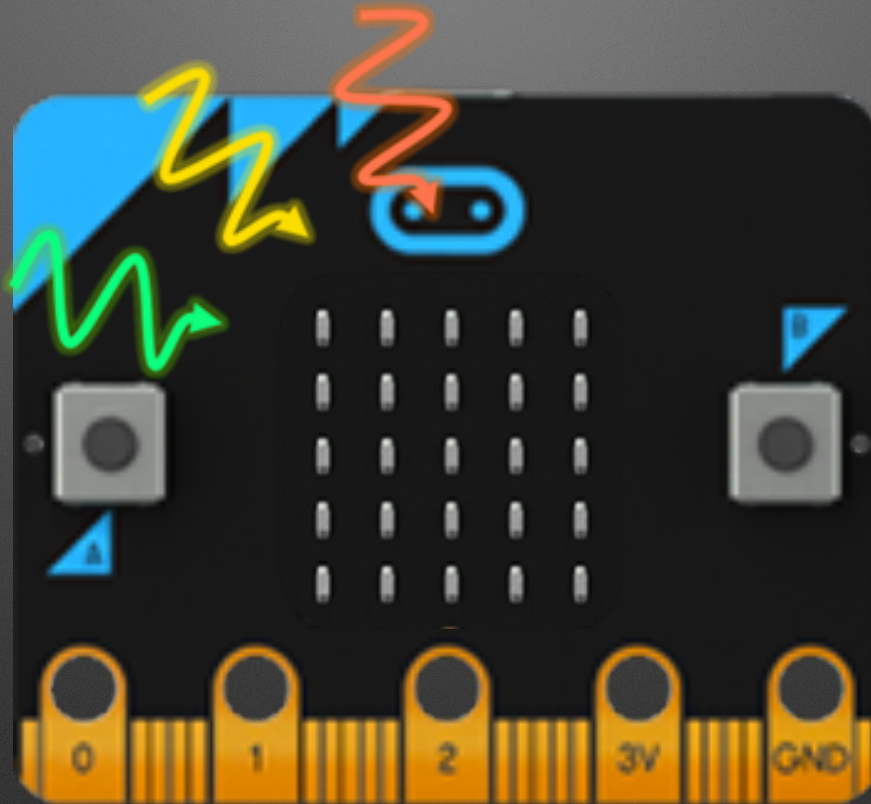
Buttons



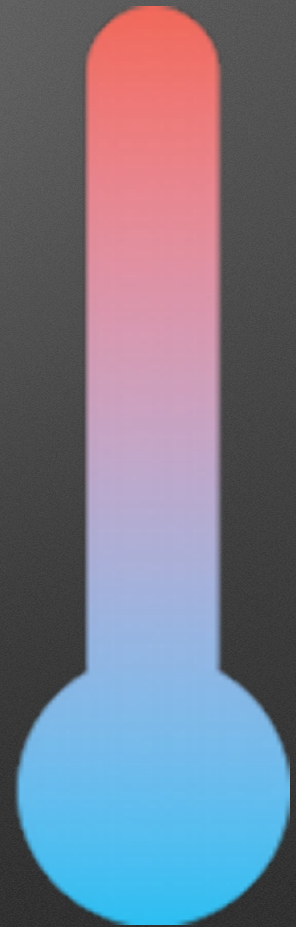
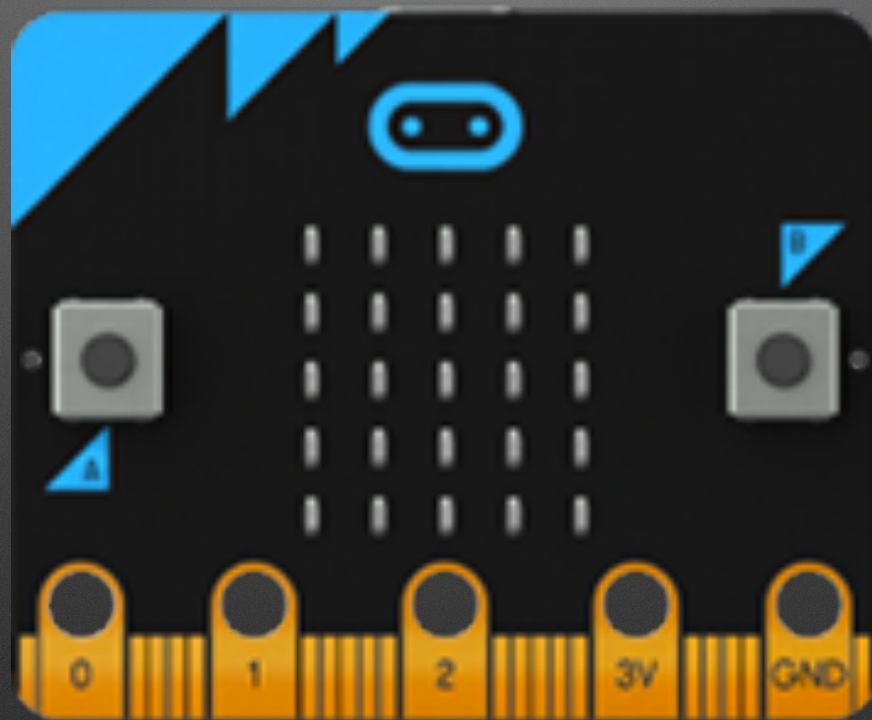
Connectors



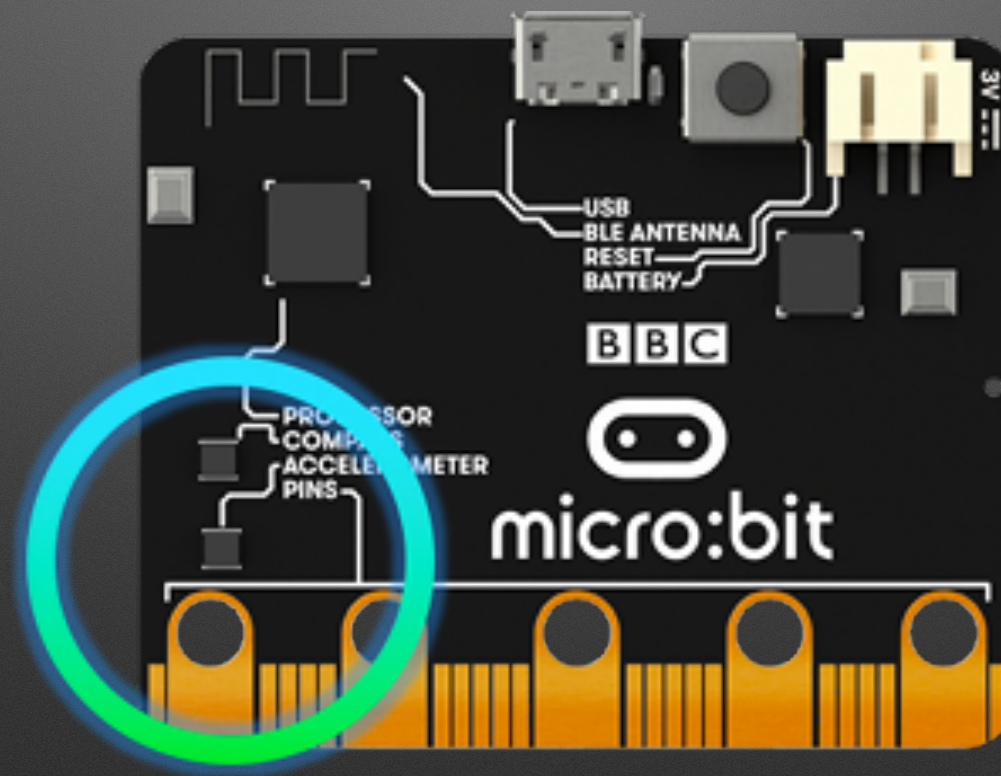
Light Sensor



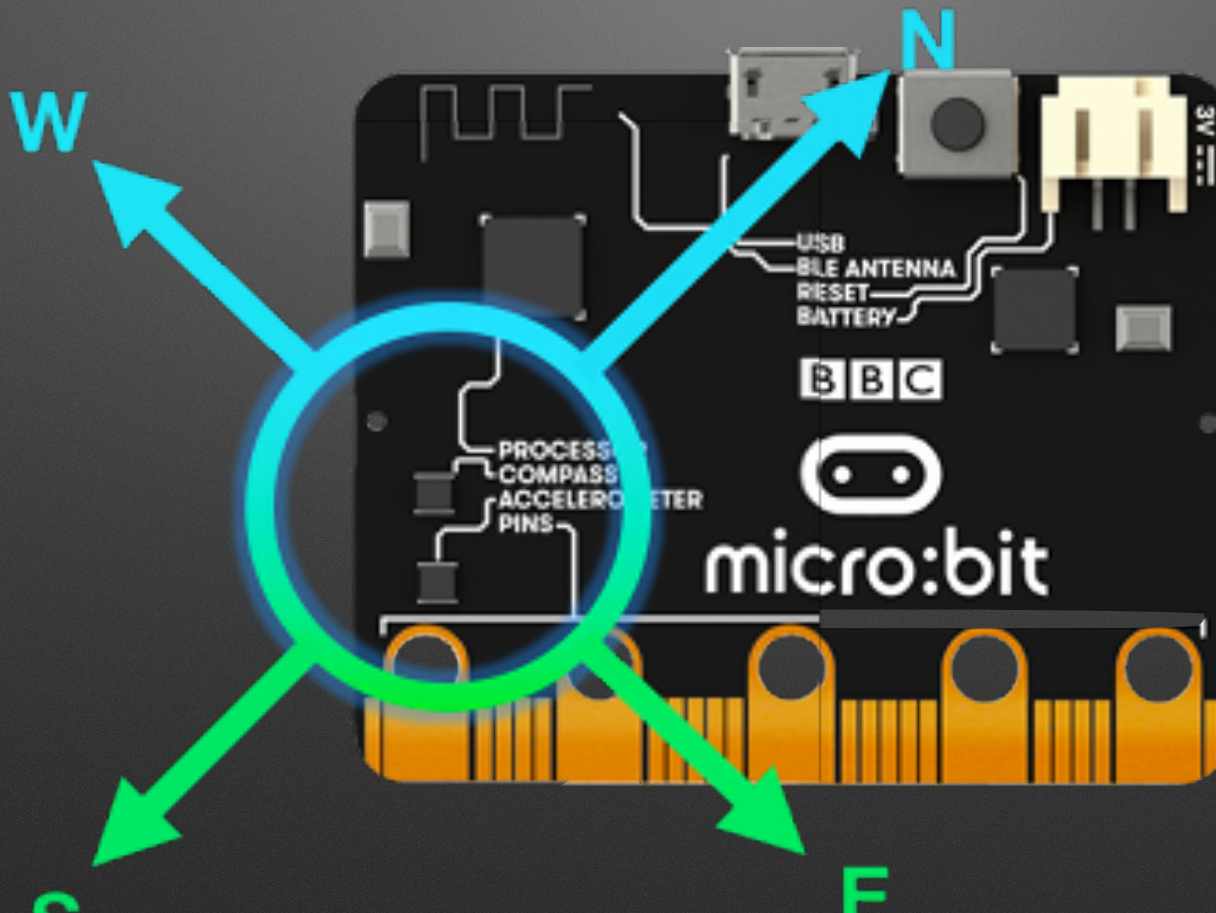
Temperature Sensor



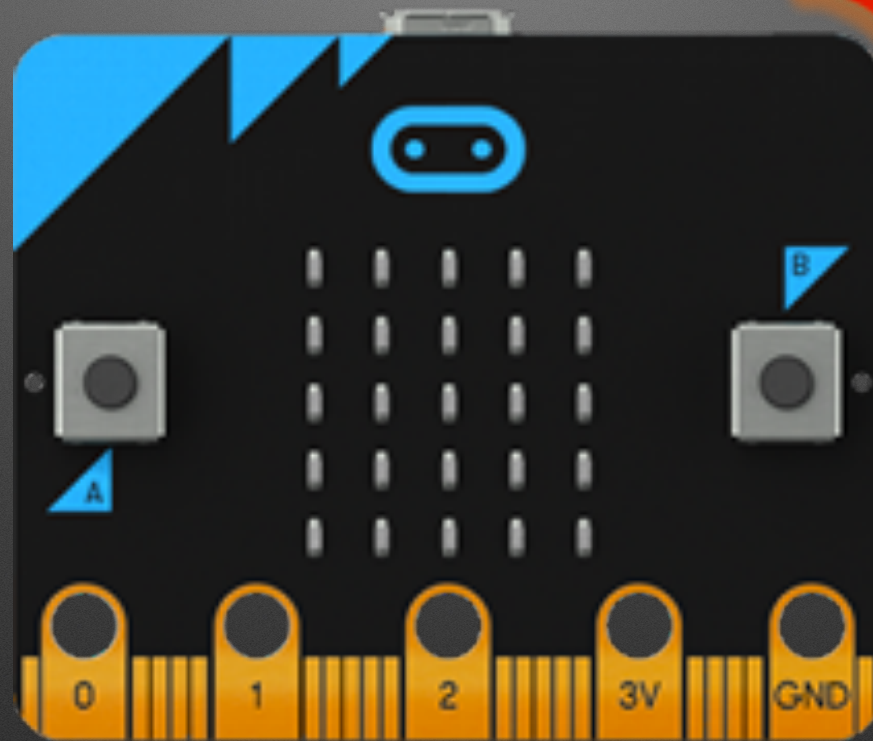
Accelerometer



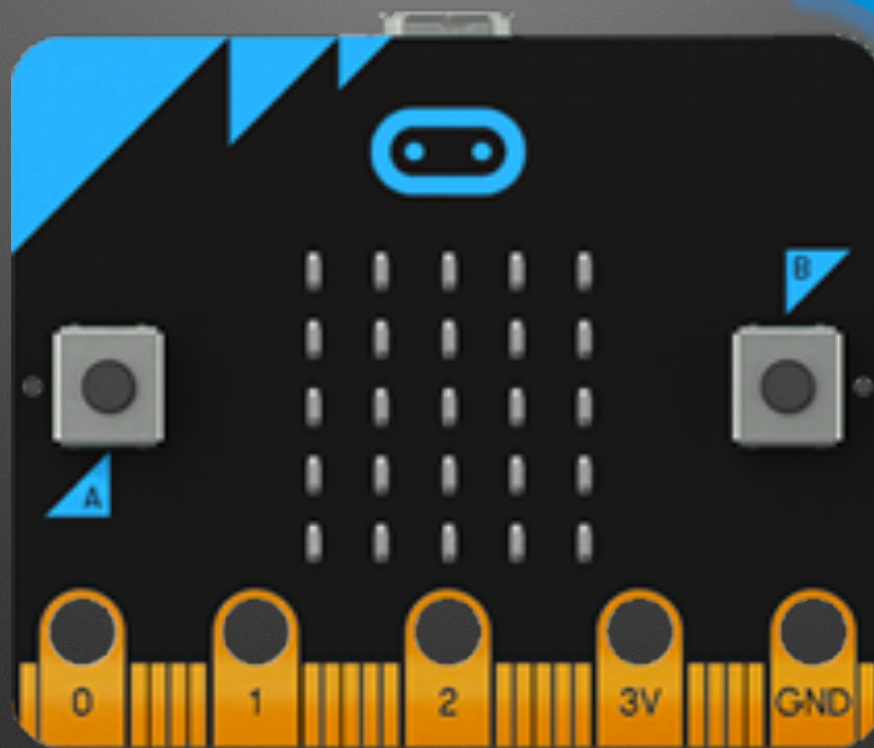
Compass



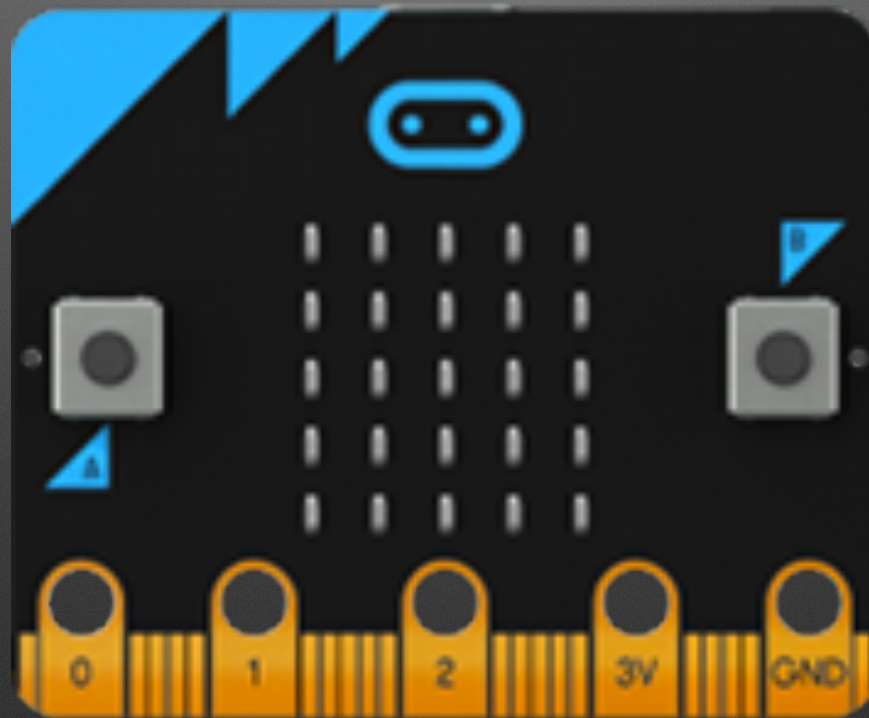
Radio



Bluetooth



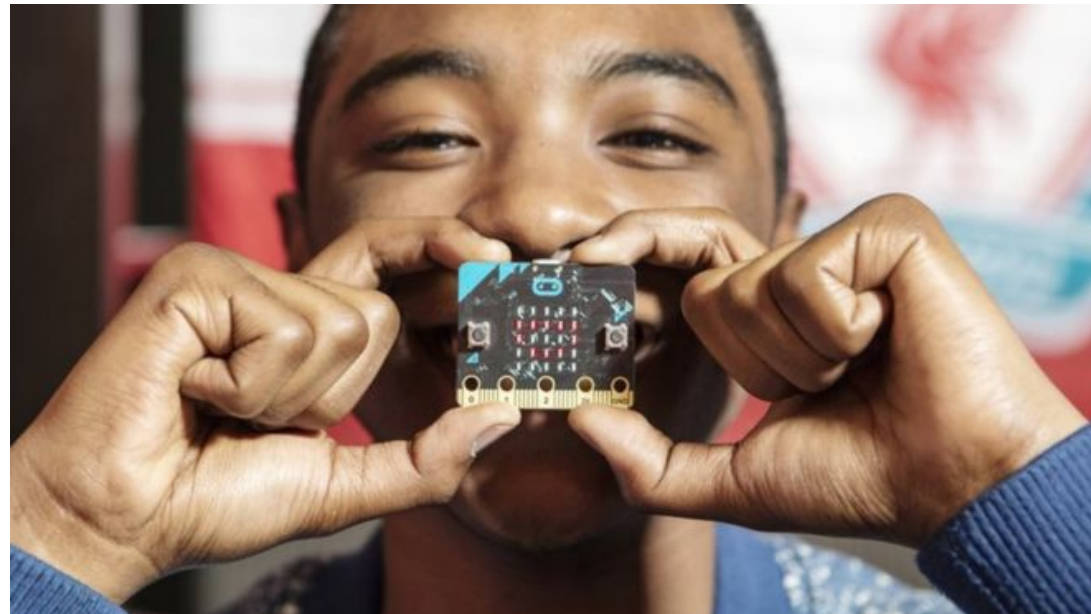
Low Cost: ~\$13 US



**Thanks: Micro:bit Educational
Foundation**
and Hal Speed

2015

- BBC Make It Digital
- 29 partners
- 1 million micro:bit devices
- 11-12 year olds
- Across the U.K.



Lessons Aligned to Code.org CS Fundamentals

- Lessons extend the concepts taught in the Code.org curriculum by using micro:bit and MakeCode
- Course E – Loop and Functions
- Course F – Variables and Conditionals



4th Grade

Course E

5th Grade

Course F



Lesson

**Course E - Loops 1 -
Loops and
Animations**



Lesson

**Course E - Loops 2 -
Nested Loops and
Scoreboards**



Lesson

**Course E -
Functions 1 - A
Simple Function for
a Superhero**



Lesson

**Course E -
Functions 2 -
Functions for a
Digital Pet**



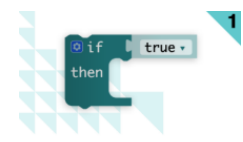
Lesson

**Course F -
Variables 1 -
Variables With A
Counter**



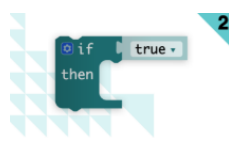
Lesson

**Course F -
Variables 2 -
Variables and
Emotions**



Lesson





**Course F -
Conditionals 1 -
Conditionals with
the Weather
Predictor**



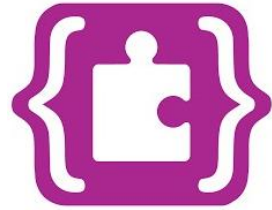
Lesson

**Course F -
Conditionals 2 -
Conditionals with
Rock Paper
Scissors Game**

Sample of Lessons

Lesson	Resources
 Answering Machine	<ul style="list-style-type: none"> • 5 Minute Lesson Plan (pdf) (pptx) • Teacher Notes (pdf) (docx) • Hex File (Tip: Save link for Mac, Save target for PC)
 Guess the Number	<ul style="list-style-type: none"> • 5 Minute Lesson Plan (pdf) (pptx) • Teacher Notes (pdf) (docx) • Hex File
 Temperature	<ul style="list-style-type: none"> • 5 Minute Lesson Plan (pdf) (pptx) • Teacher Notes (pdf) (docx) • Hex File
 Die Roll	<ul style="list-style-type: none"> • 5 Minute Lesson Plan (pdf) (pptx) • Teacher Notes (pdf) (docx) • Hex File

Third-Party Curricula



Microsoft MakeCode Intro to CS

<https://aka.ms/intro2cs>

- | | |
|---------------------------|-------------------------------|
| 1. Making | 8. Coordinate Grid System |
| 2. Algorithms | 9. Booleans |
| 3. Variables | 10. Music and Arrays |
| 4. Conditionals | 11. Bits, Bytes, and Binary |
| 5. Iteration | 12. Radio |
| 6. Review/Mini-Project | 13. Arrays |
| 7. Coordinate Grid System | 14. Independent Final Project |



PLTW Gateway:
Computer Science for
Innovators and Makers

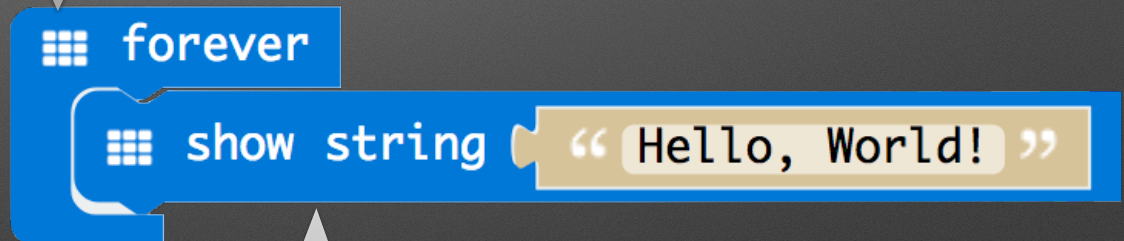
[https://www.pltw.org/our-program/
pltw-gateway-
curriculum#curriculum-4](https://www.pltw.org/our-program/pltw-gateway-curriculum#curriculum-4)



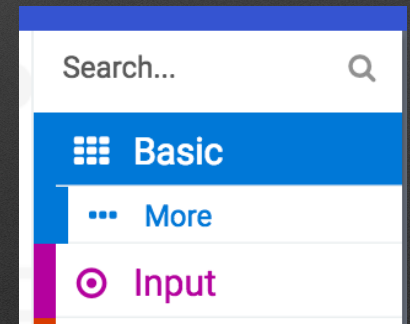
“Hello, World!”: First Program

- Block-based editor
- Built-in simulator
- Deployment to Micro:bit

Icon Indicates
Palette



Block Color Indicates Palette



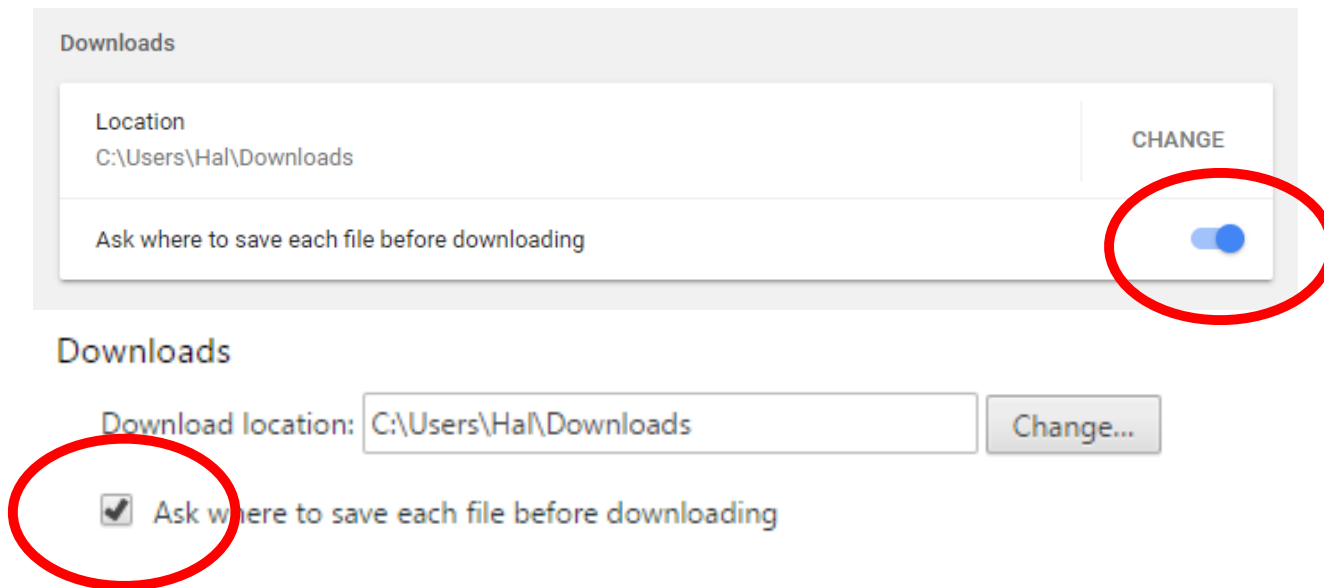


Setup

- Hardware Handout — *Thanks Microbit Foundation!*
 1. Pull out the micro:bit box / open
 2. Connect via USB cable
- Browser
 1. Open microbit.org
 2. Select “Let’s Code”
 3. Click “Let’s Code” button

Chrome Setup

- `chrome://settings/downloads` - OR - Show advanced settings...



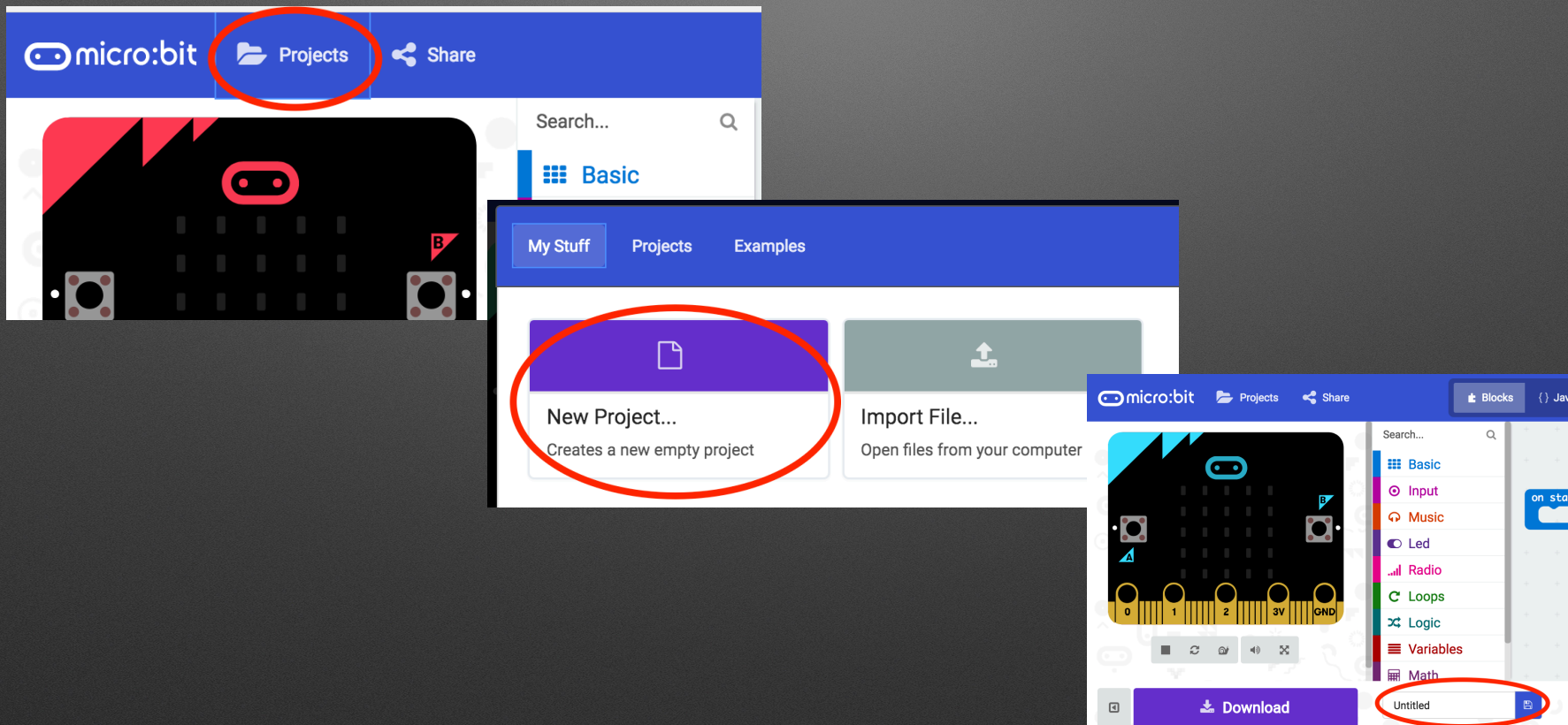
Personalization!

- Hello Bill / Hello Michael / Hello

```
⌘ forever
```

```
⌘ show string “ Hello SIGCSE! ”
```


New Project: Projects > New Project...

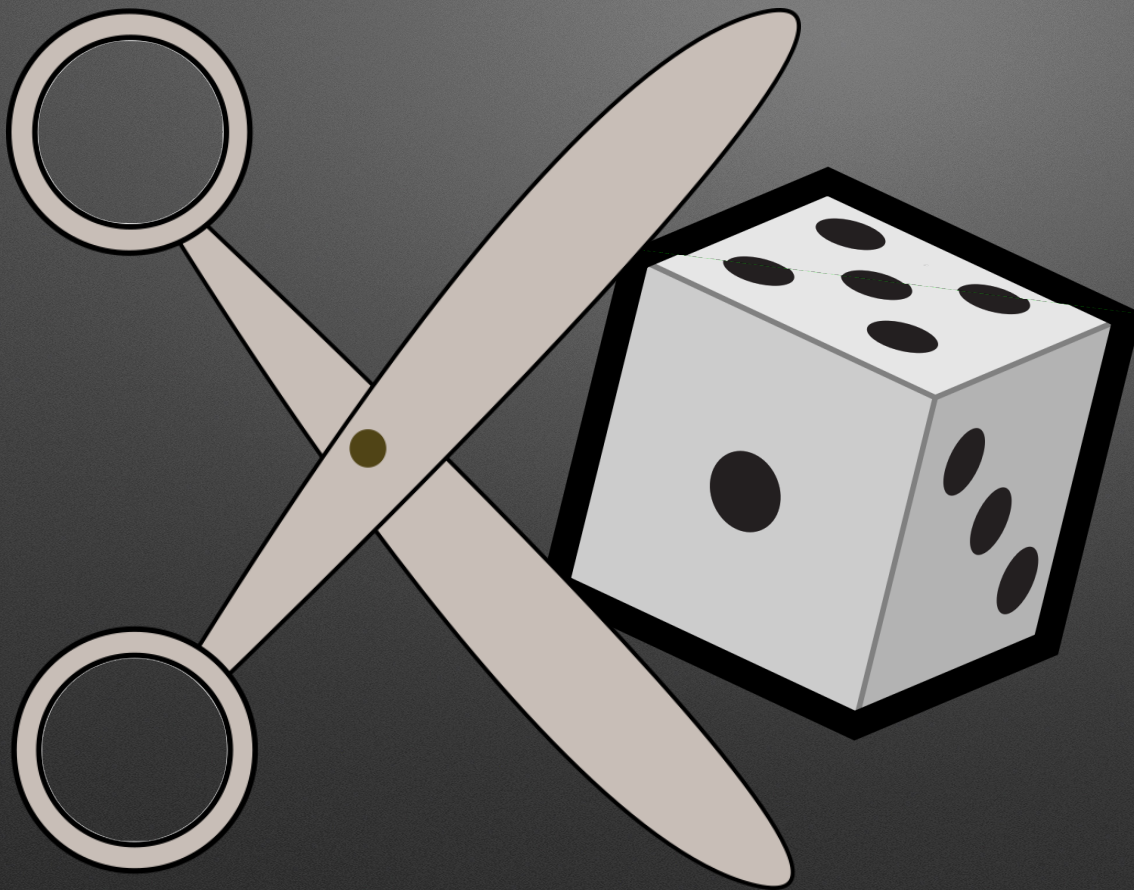




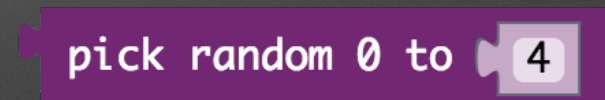
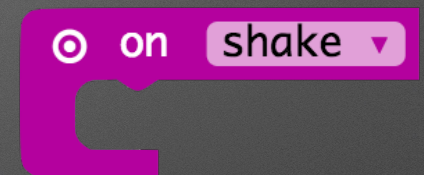
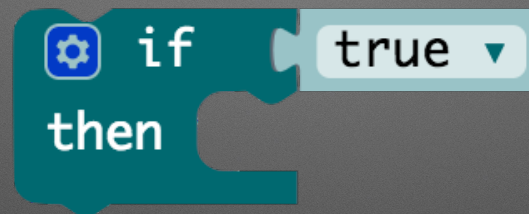
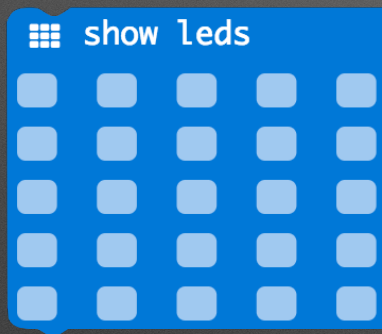
Programming: Logic & Action

- Picking between *three* tough choices
 - Cookie, Cake, Pie
 - Super Strength, Invisibility, Telekinesis
 - ...

Obvious Solution...



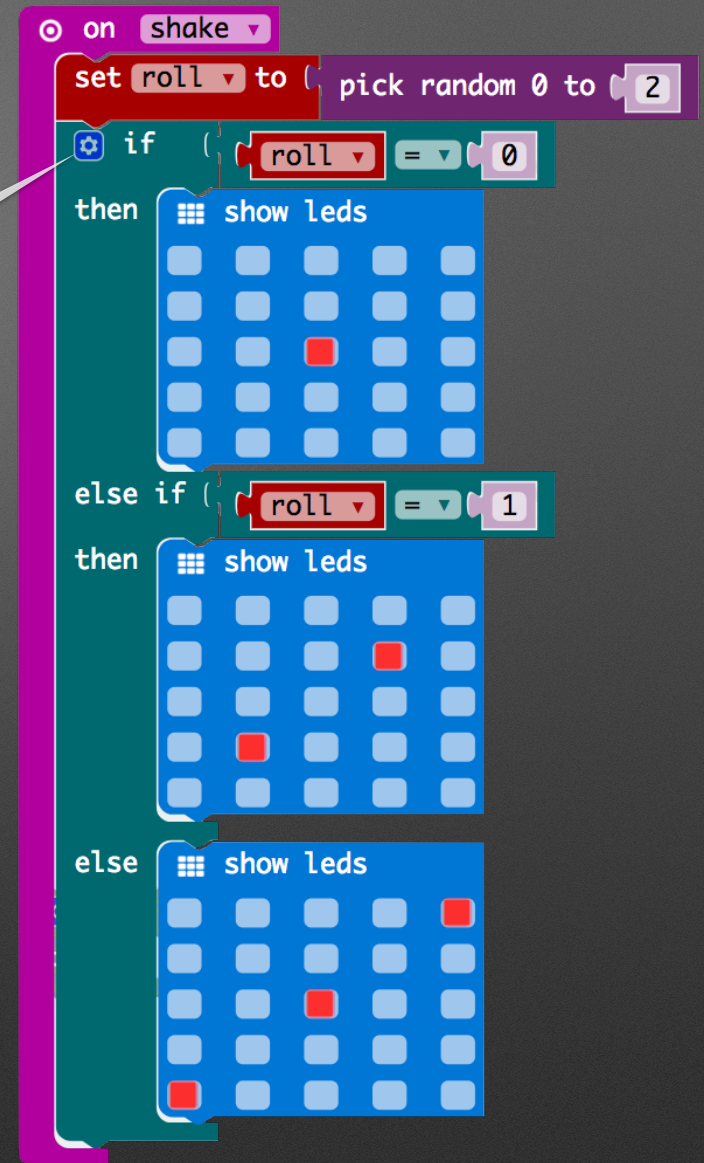
Parts



1. Color indicates Palette
2. Incremental Development:
Try parts in Simulator

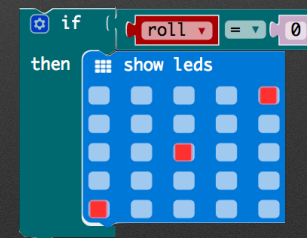
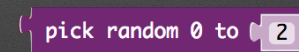
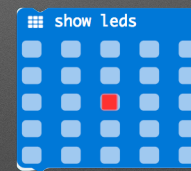
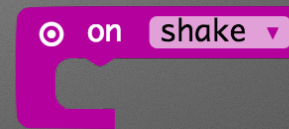
Solution

Pro Tip: Blocks with a button have additional features (else-if)



Concepts

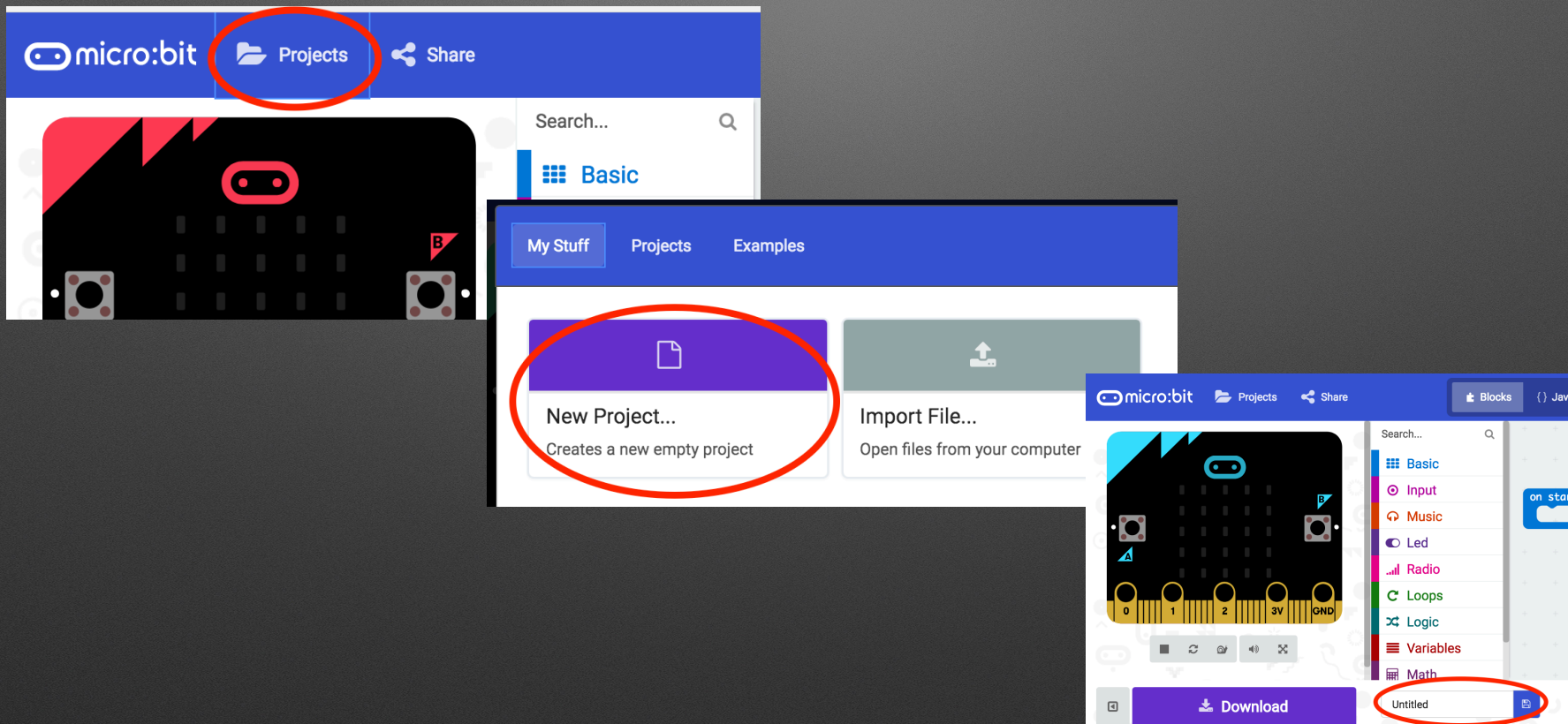
- Event driven programming
- Bitmapped Graphics
- Ranges & Representations
- Logic



Pedagogy

- Active Learning
- Discovery Based
- Constructionist

New Project: Projects > New Project...

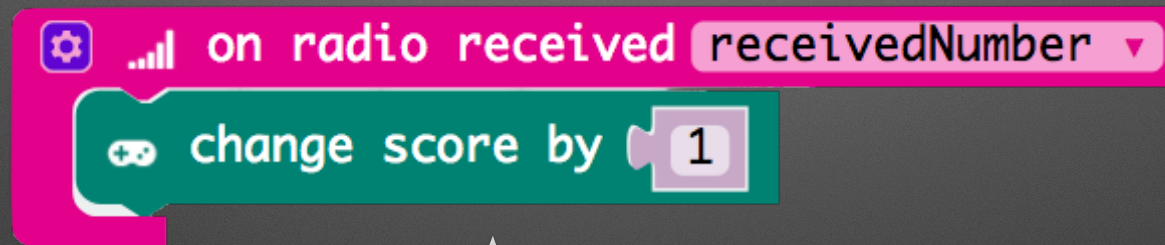




Broadcast Basics

- Radio Palette: Broadcast Based Radio Transmissions
 - String, Number, Key/Value Pairs, ...

Receiver



“Game” blocks in “Advanced”
Section of Palette

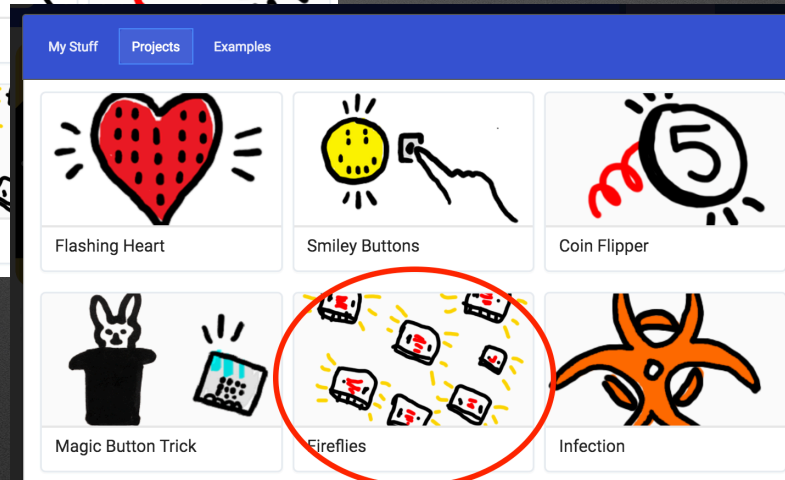
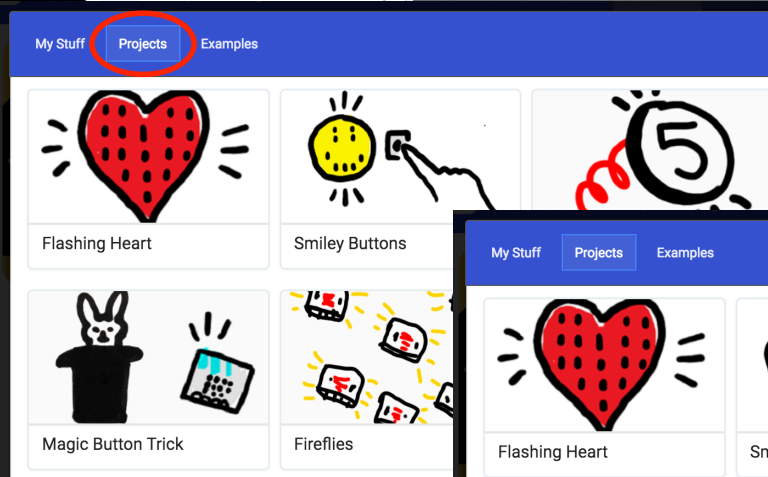
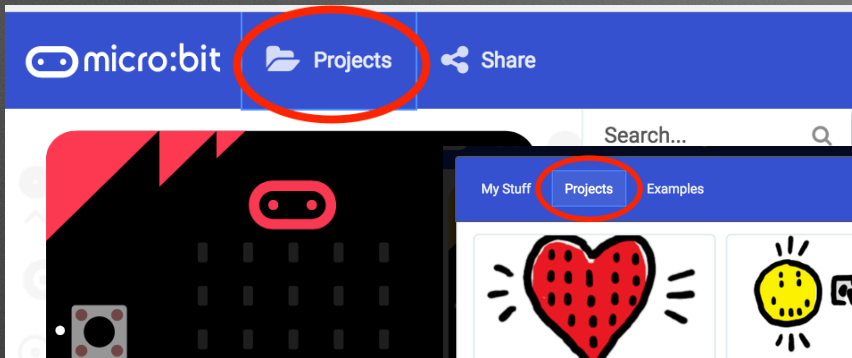
▼ Advanced

Full Broadcaster



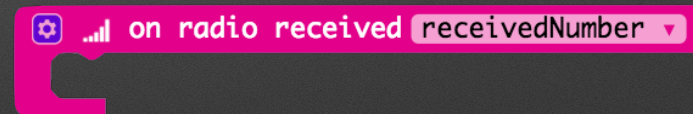
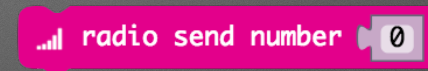


& Firefly Fun

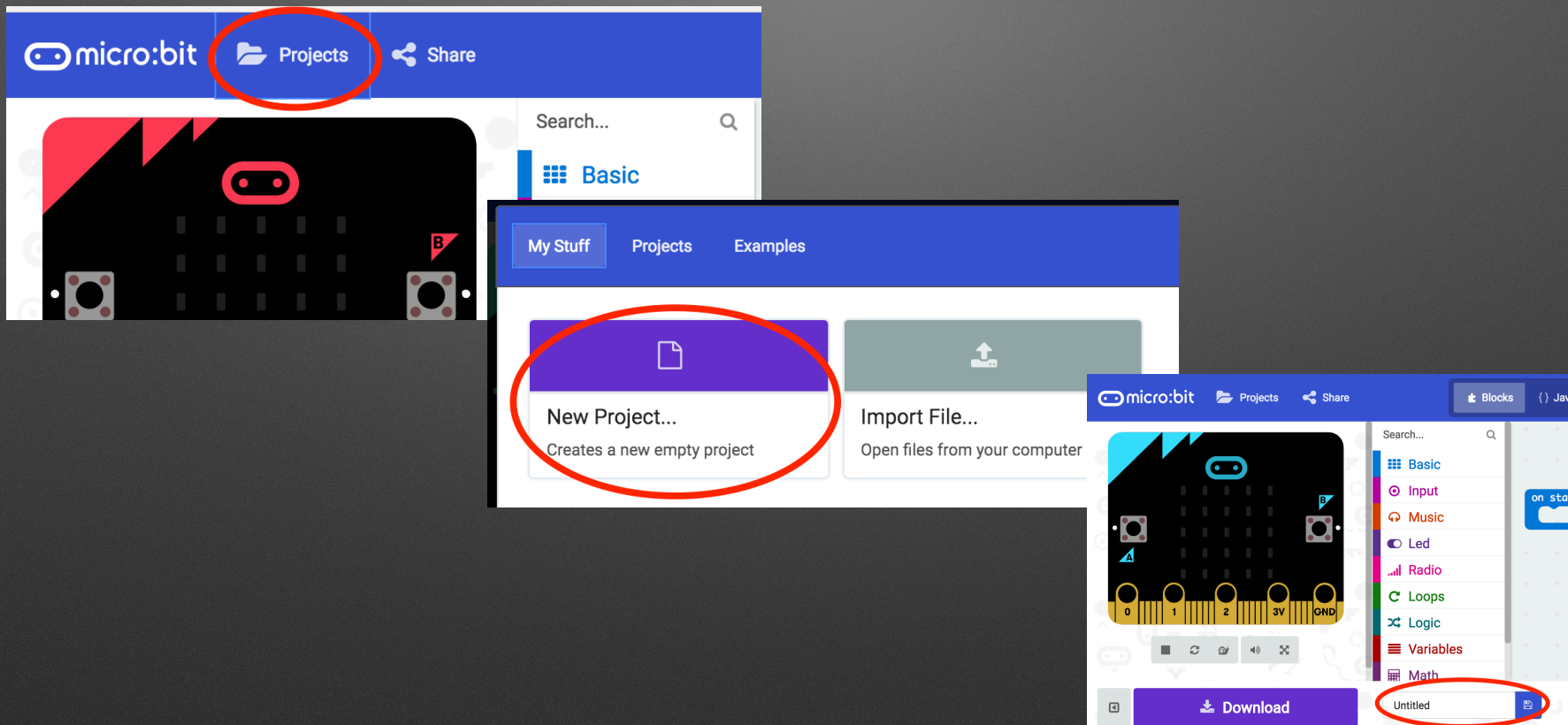


Concepts

- Broadcasting
- Network Addresses
- Asynchronous clocks / Sync problems



New Project: Projects > New Project...



Goody Bag: Hardware



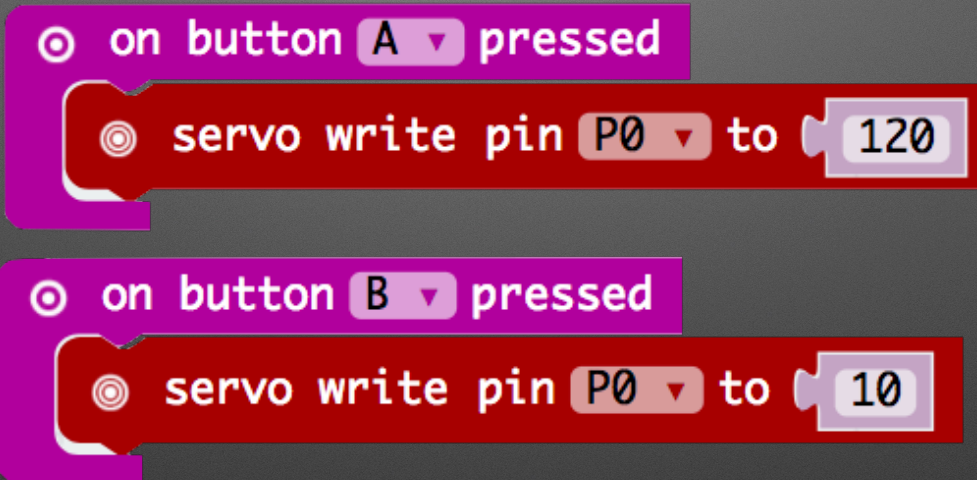
Motor Mayhem

An Intro to Servos



Motor Mayhem

An Intro to Servos



Inchworm Insanity

<https://makecode.microbit.org/projects/inchworm>



Awesome (?) Audio

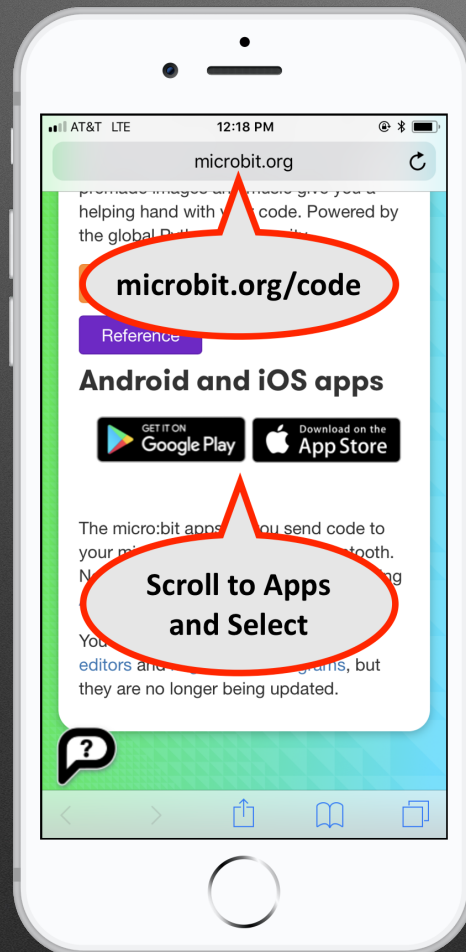
Time Permitting

Concepts

- I/O
- Basic Electric Circuits/Electronics

Break & App Install

Android & iOS



**Android
(optional)
Search for &
Install
Bitty
Controller
(\$1.99)**

Bluetooth Background

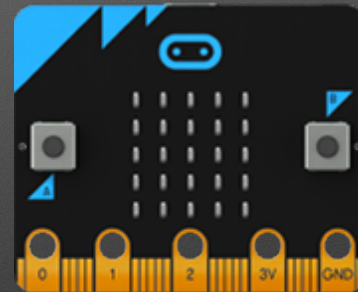
- Uses different protocol than  Radio
- Not a group broadcast

Bluetooth Background

Central



Peripheral



Bluetooth Background

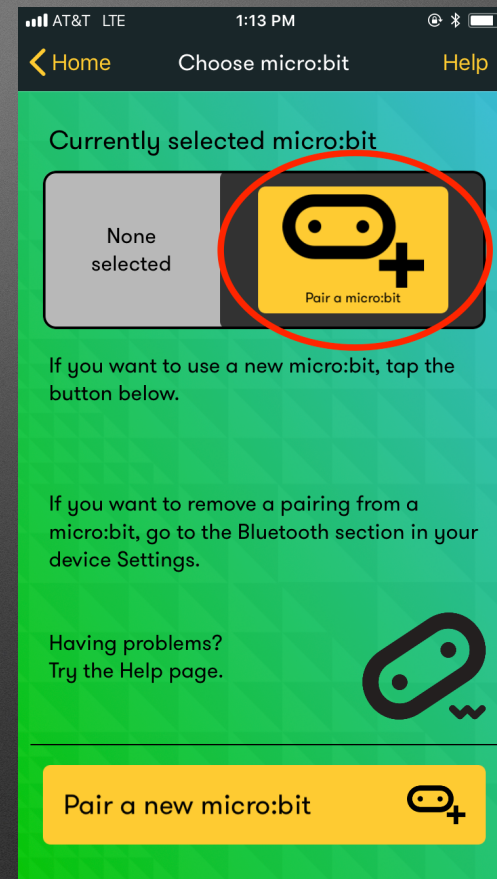
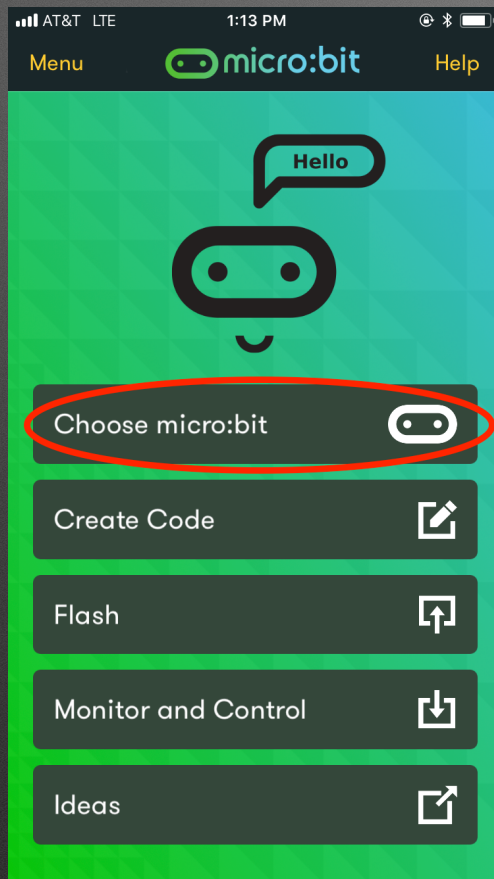
Central



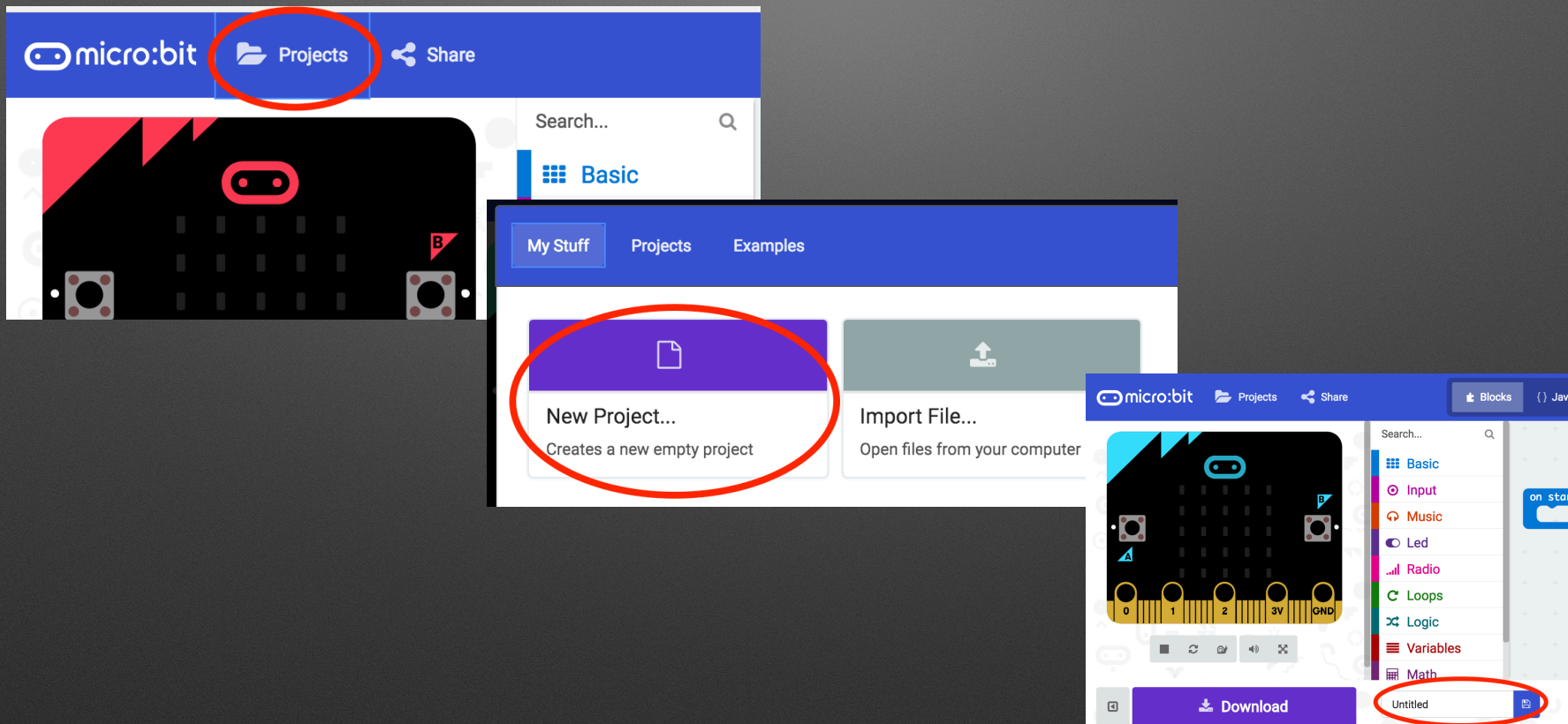
Bluetooth Background

- Bluetooth has various levels of security
 - “Pairing” — Forming a “permanent” bond
(Exchanging security info. once and storing it)
- Block editor supports three types
 - No pairing (“insecure” - we’ll use this)
 - Just Works (default; pretty safe)
 - Passkey Pairing (more secure)

Pairing

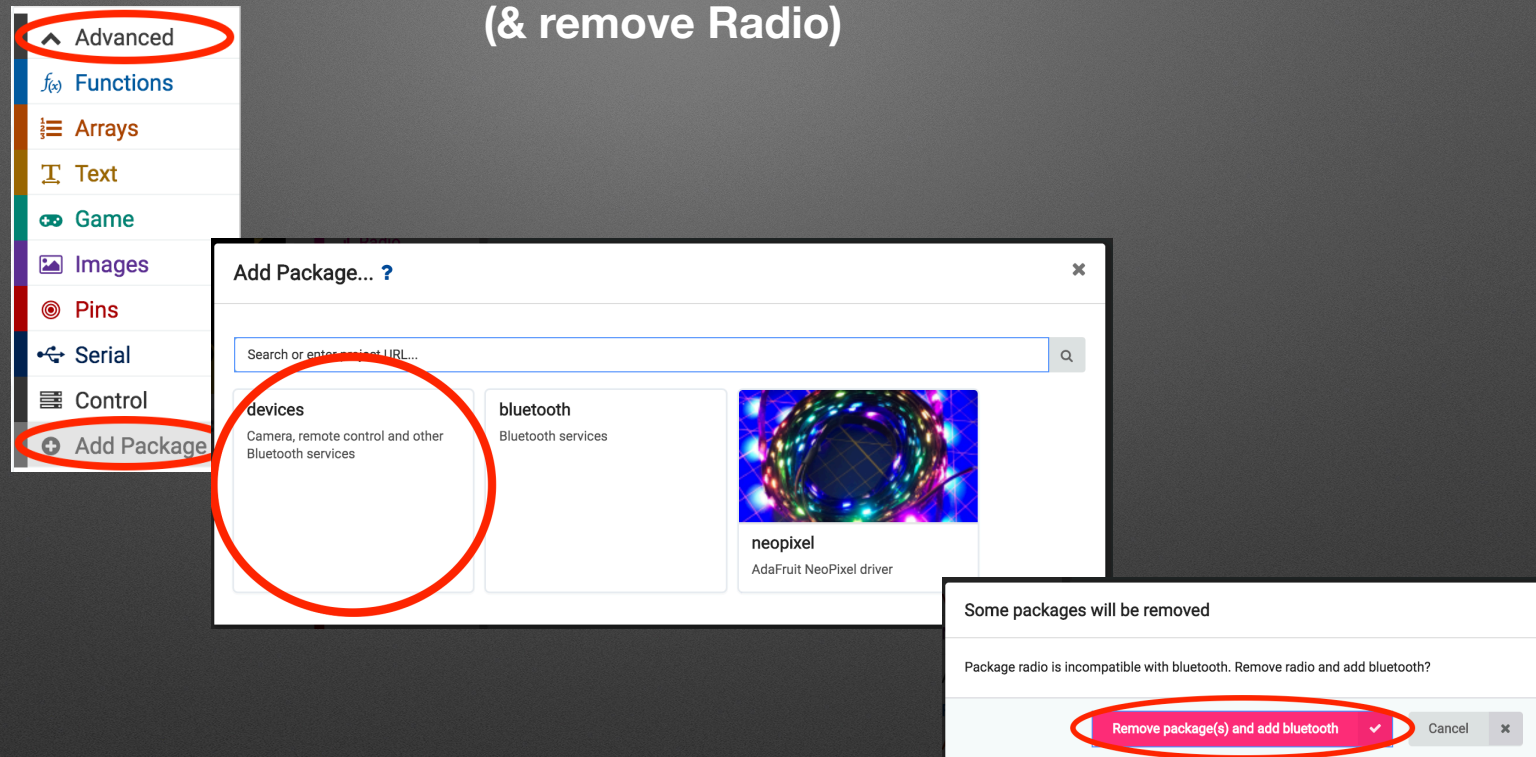


New Project: Projects > New Project...

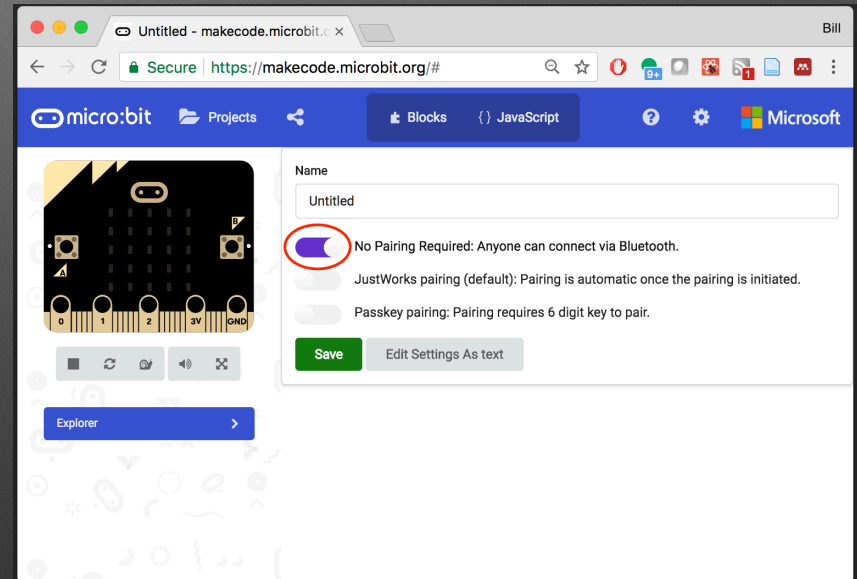
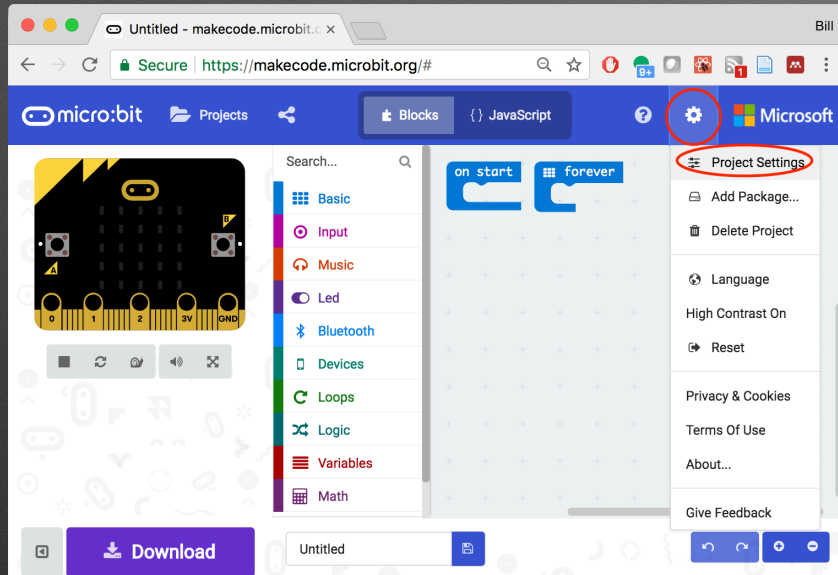


Add Devices

(& remove Radio)

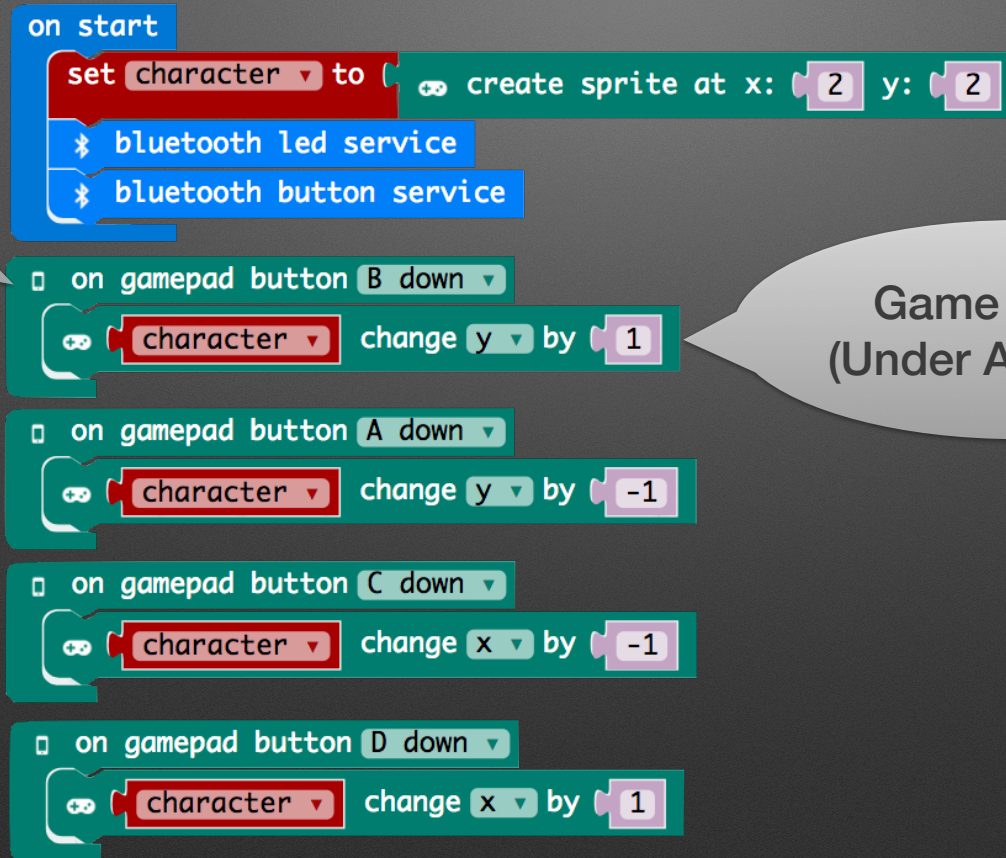


Project Settings



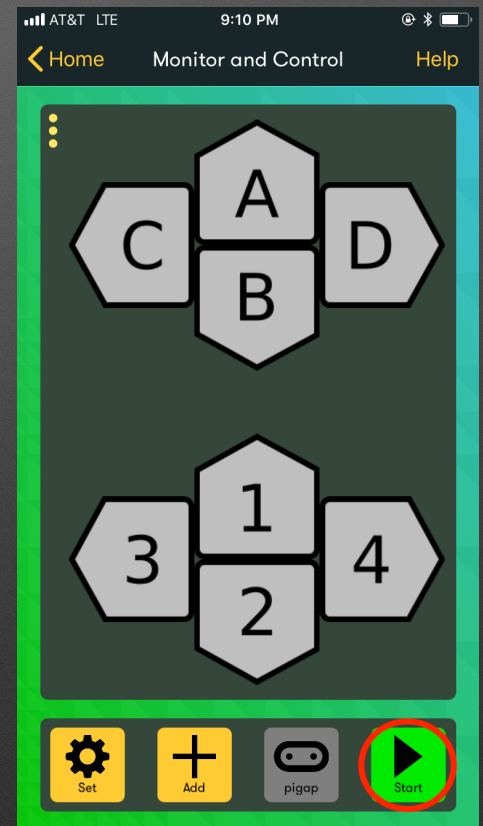
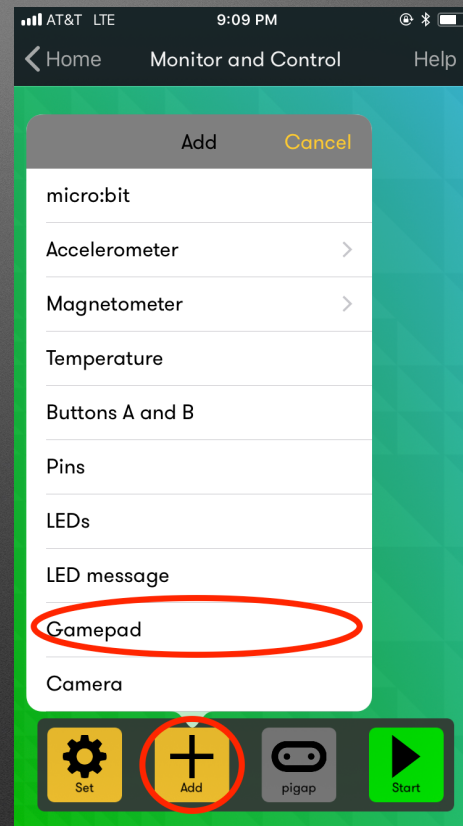
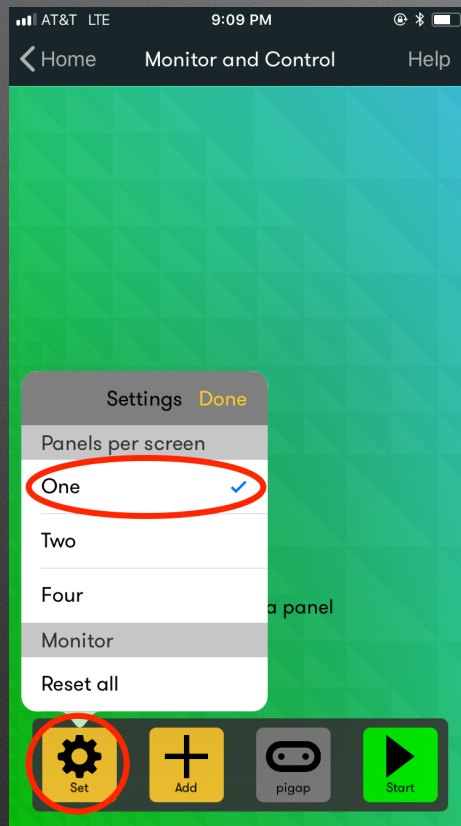
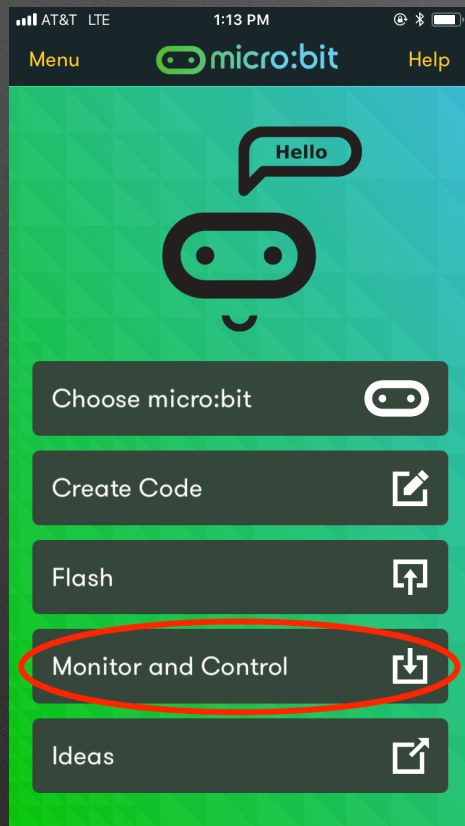
Program

Devices Palette



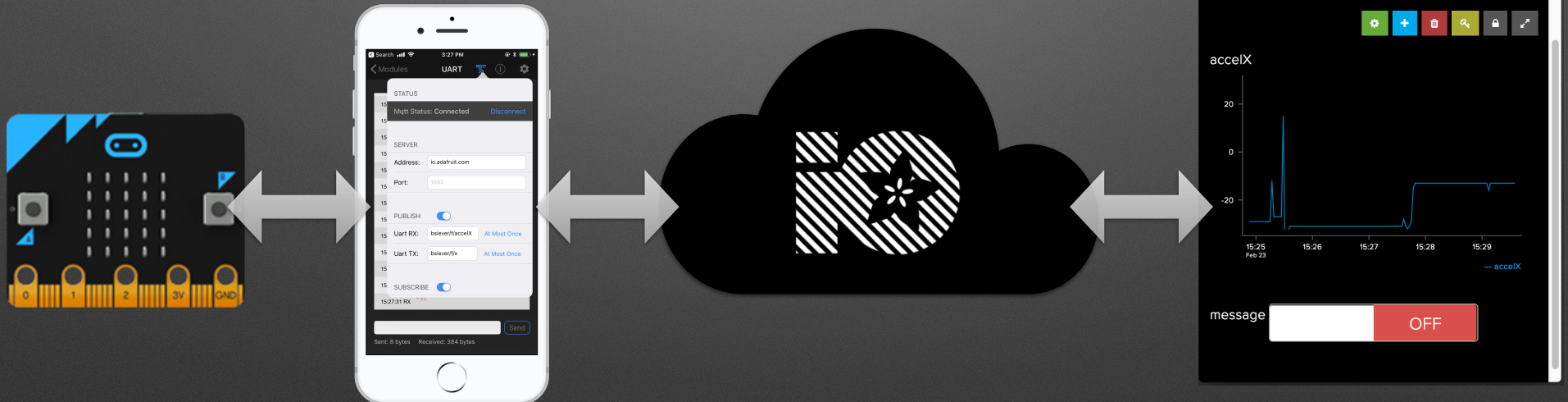
Game Palette
(Under Advanced)

App Configuration



IoT Example

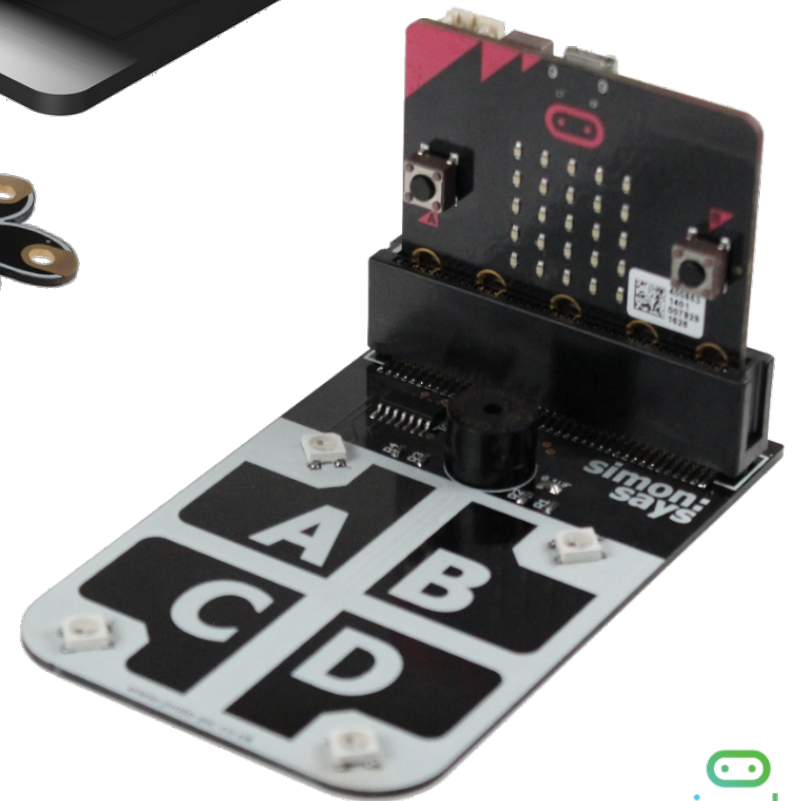
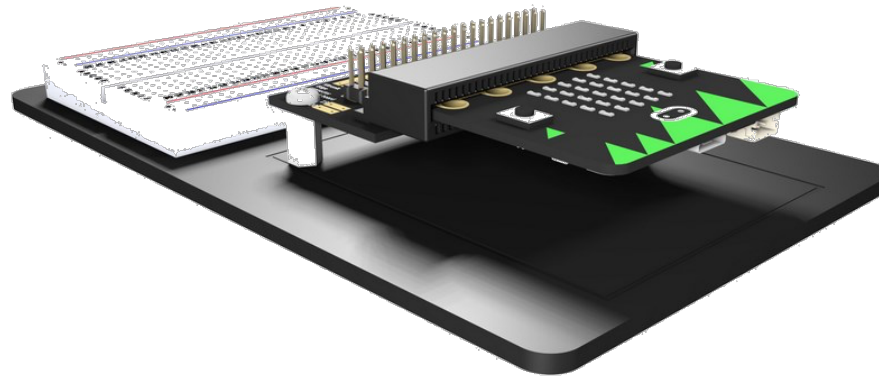
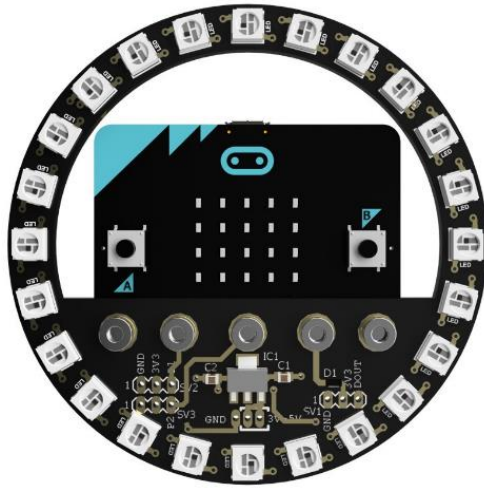
AdaFruit.io



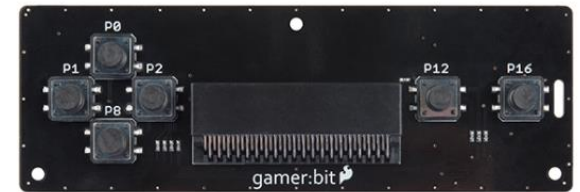
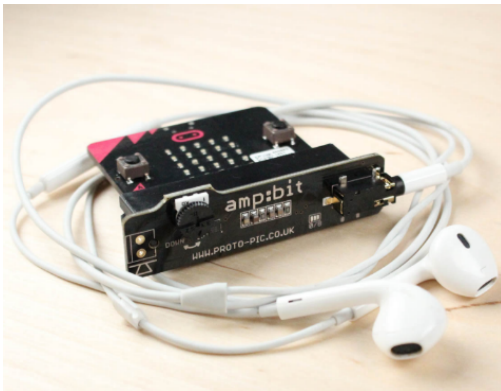
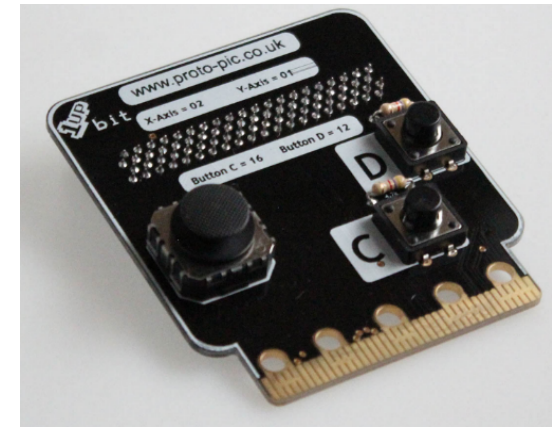
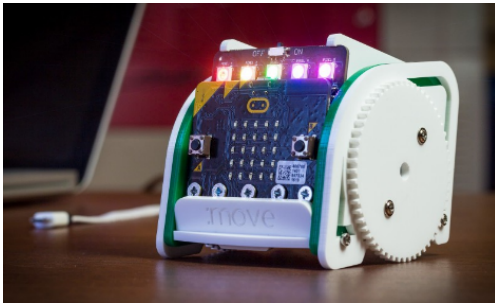
Misc.

- C (C++) / Arduino
- Phone acts as border router
- AdaFruit.io can tie into other services
 - Texts/Notifications (IFTTT), Webhooks, etc.

Additional Hardware



<http://microbit.org/assets/documents/microbit-accessories.pdf>

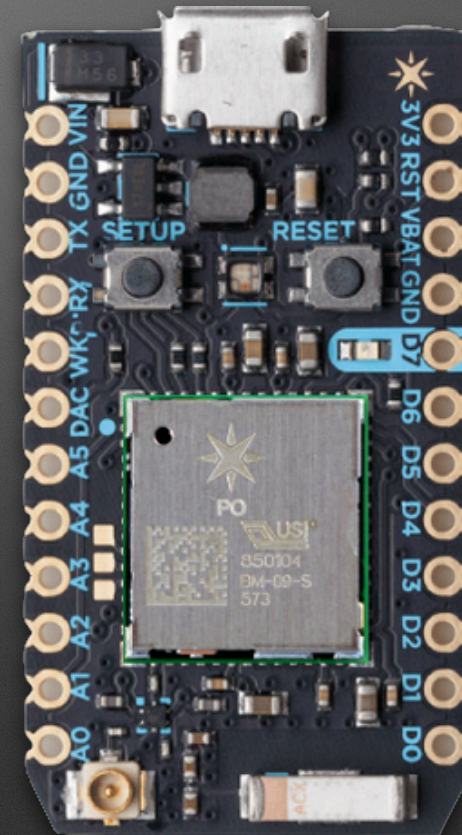


Break!



Photon Phun!

- Microcontroller
 - 120MHz ARM Cortex M3
 - 128kB RAM
 - 1MB Flash
 - Real-Time OS
 - 802.11 b/g/n
- Particle's API & Cloud Services



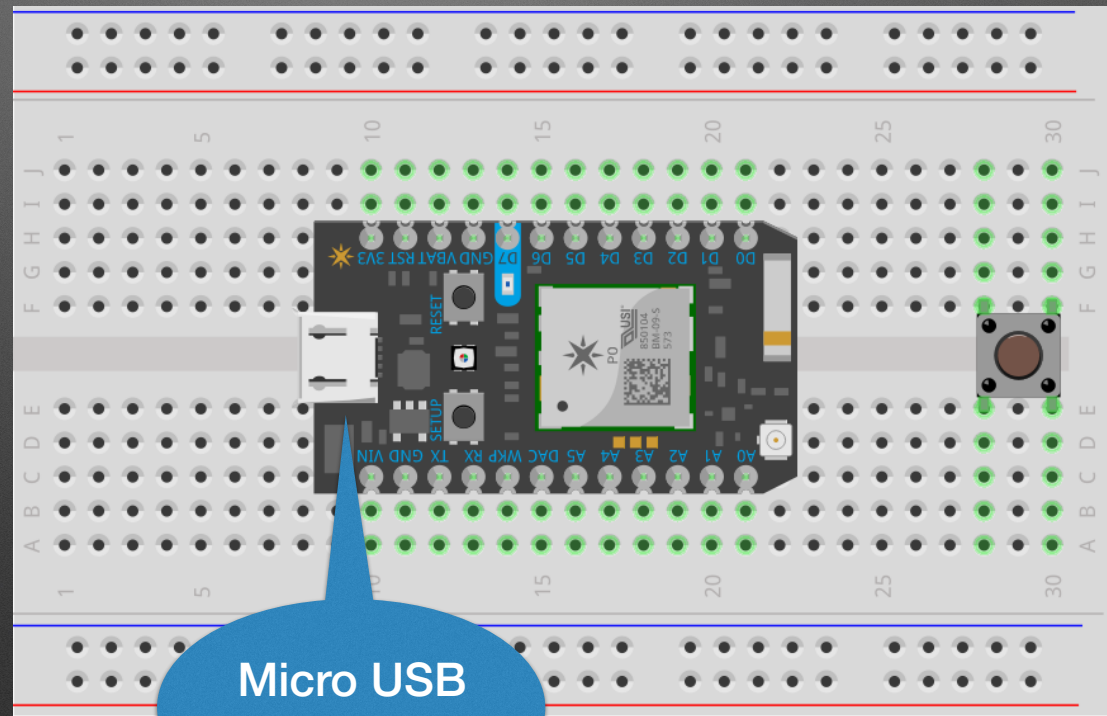
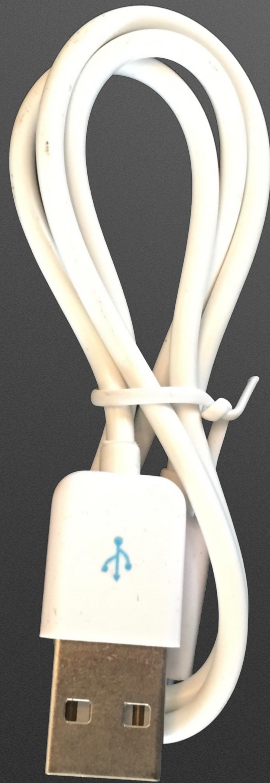
Project Overview

- Blink an LED
 - On-board
 - Circuits: An external LED
- Cloud-Controlled Blink
 - Console Control
- Button & Status

<http://particle.io/build>

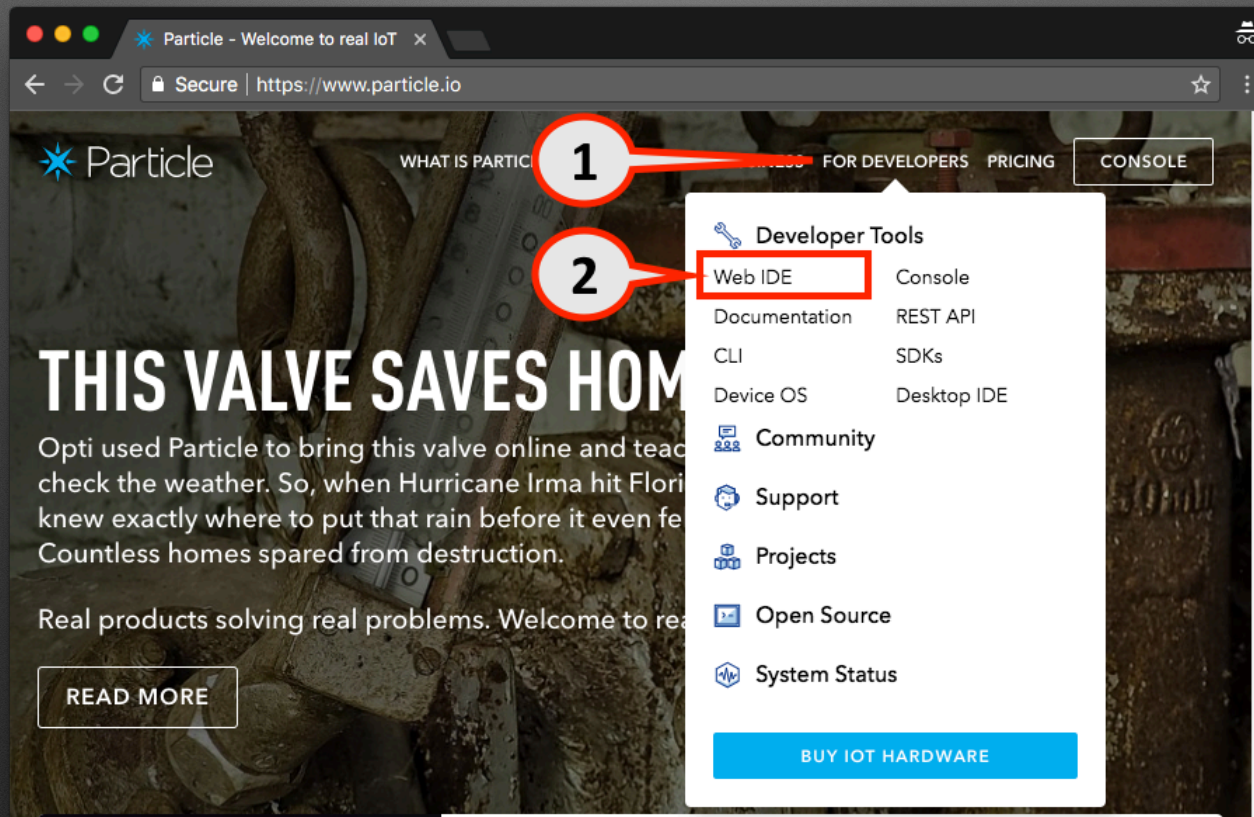
- Login
 - Username: ccsccp@siever.info
 - Password: ccsccp2018

Breadboard, Photon, & USB Power

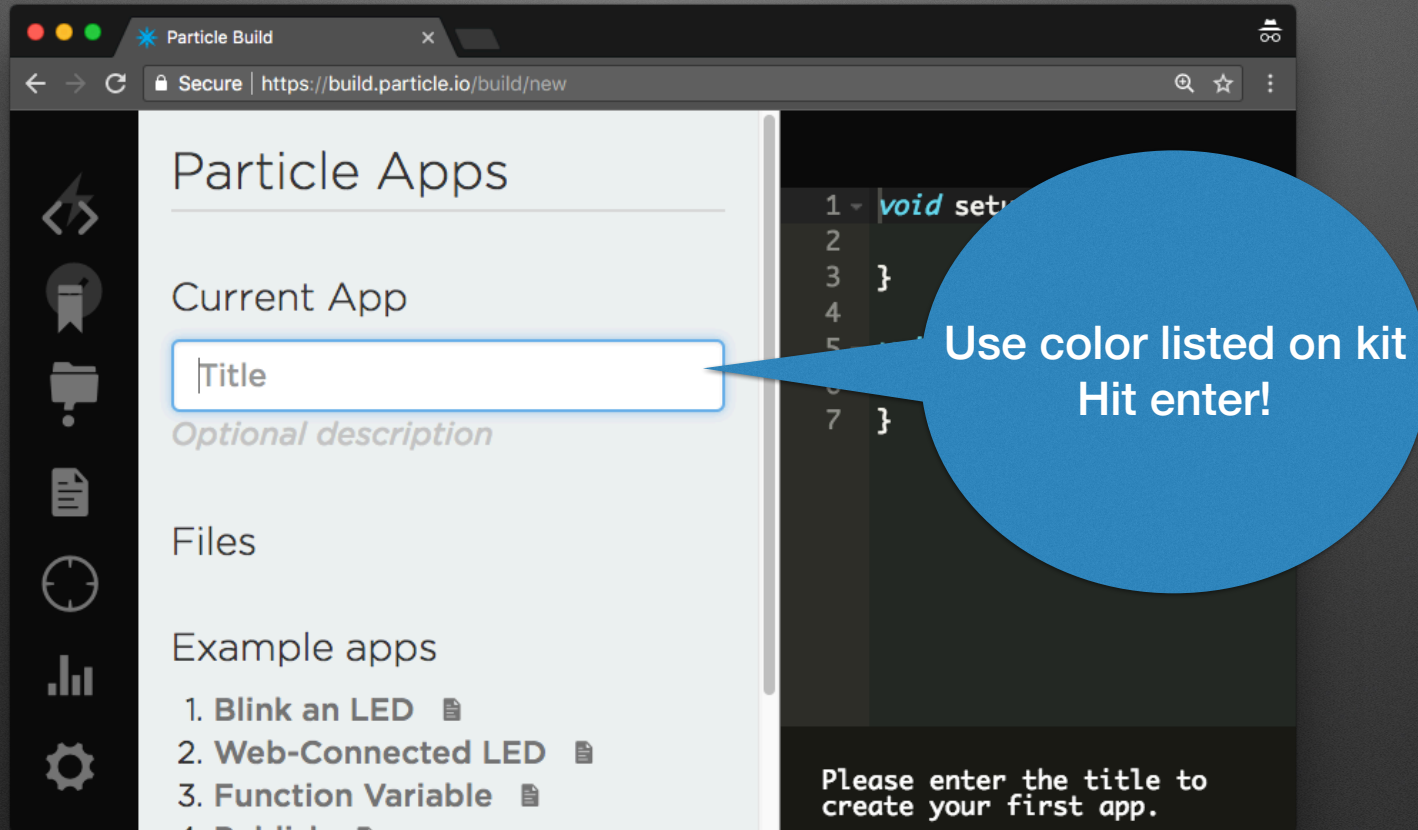


Micro USB
plug

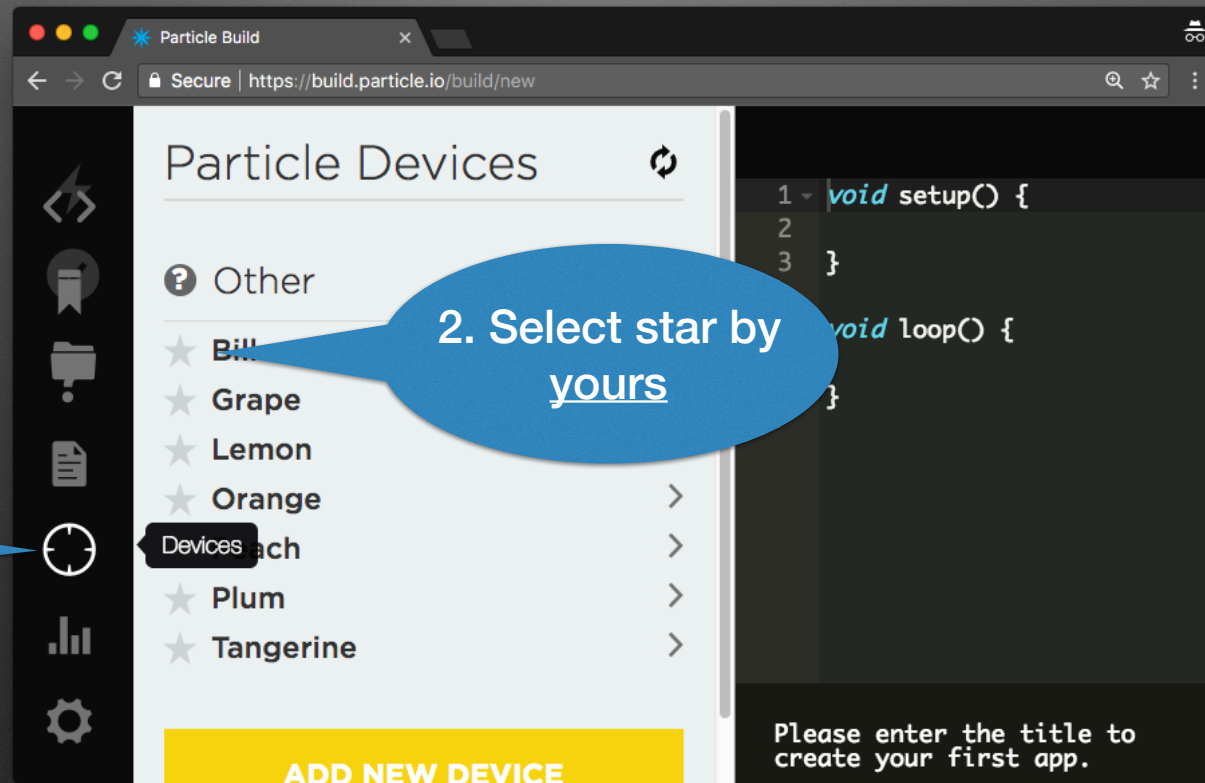
Web IDE



App Name



Select Your Device



Code



```
1 void setup() {  
2     pinMode(D7, OUTPUT);  
3 }  
4  
5 void loop() {  
6     digitalWrite(D7, HIGH);  
7     delay(500);  
8     digitalWrite(D7, LOW);  
9     delay(500);  
10 }
```

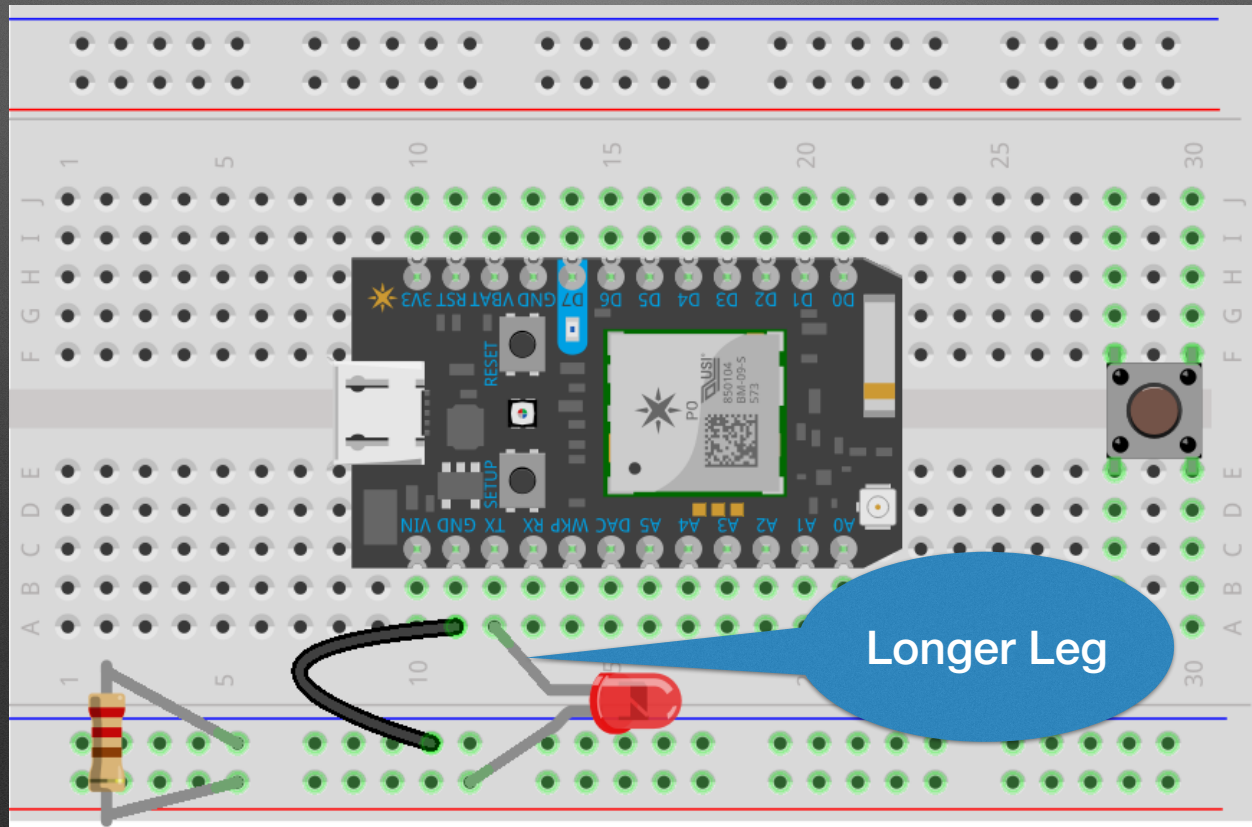
Program / Deploy

Hit "Flash" to
compile/deploy




```
1 void setup() {  
2     pinMode(D7, OUTPUT);  
3 }  
4  
5 void loop() {  
6     digitalWrite(D7, HIGH);  
7     delay(500);  
8     digitalWrite(D7, LOW);  
9     delay(500);  
10 }
```


Wire External LED




Update Code & Flash



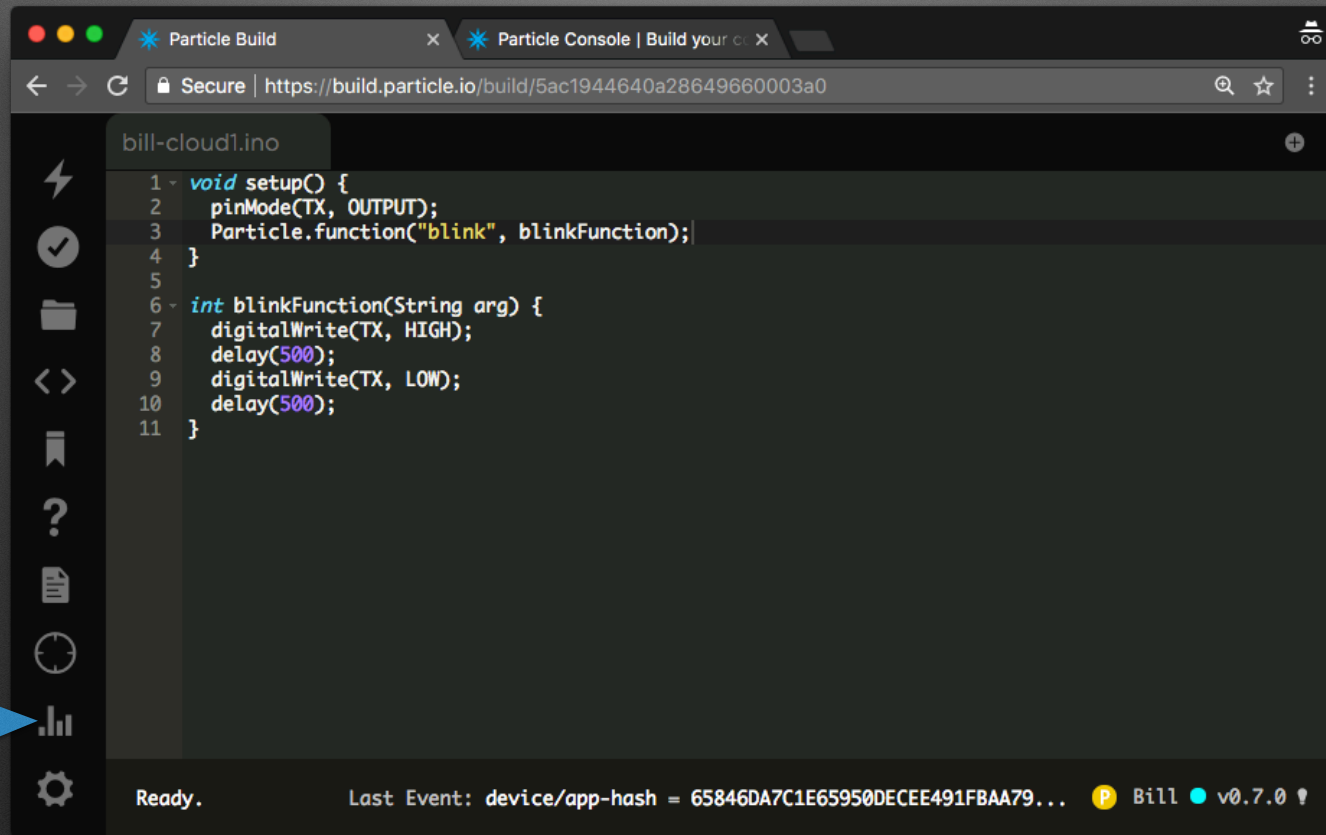
```
bill-blinky2.ino
1 void setup() {
2   pinMode(TX, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(TX, HIGH);
7   delay(500);
8   digitalWrite(TX, LOW);
9   delay(500);
10 }
```


Cloud Control



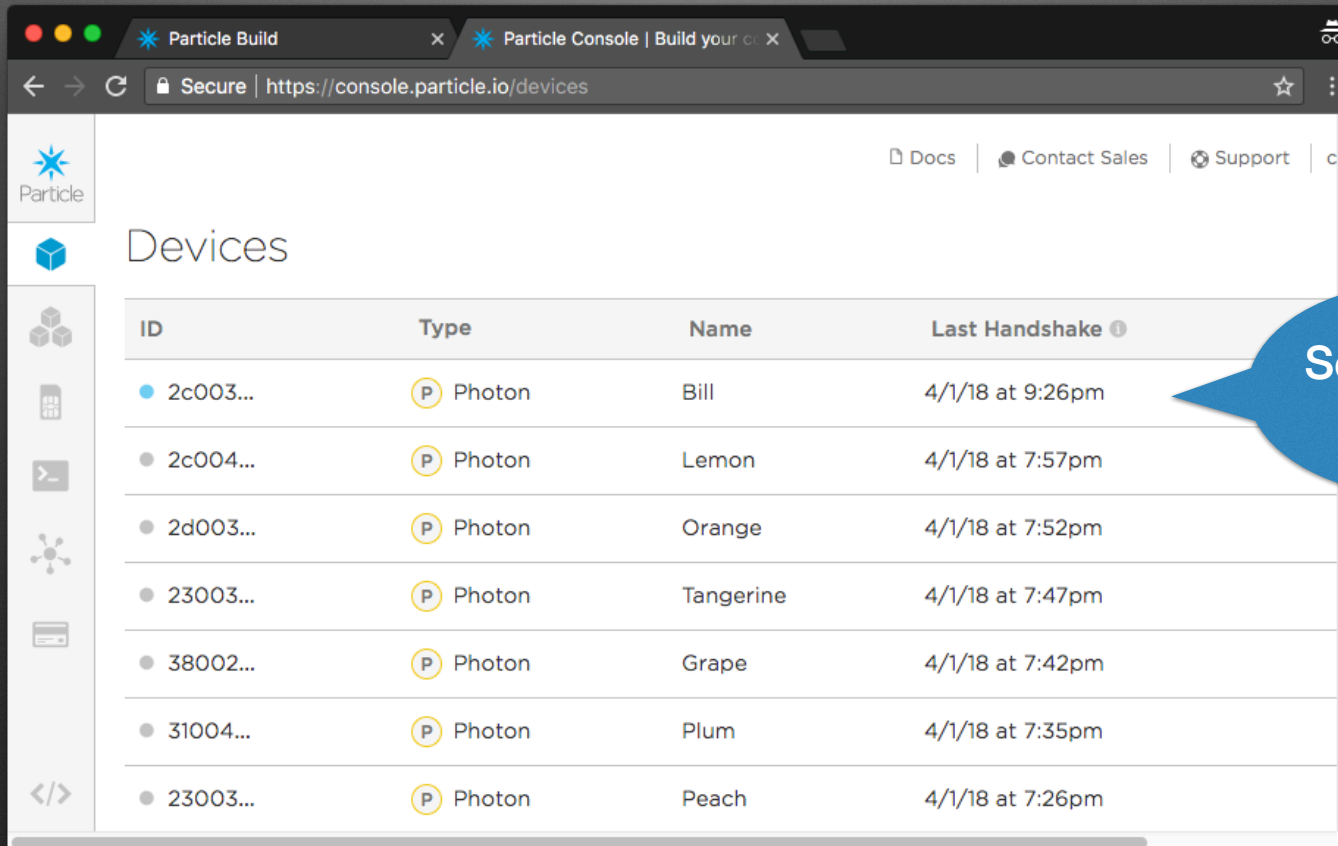
```
1 void setup() {  
2     pinMode(TX, OUTPUT);  
3     Particle.function("blink", blinkFunction);  
4 }  
5  
6 int blinkFunction(String arg) {  
7     digitalWrite(TX, HIGH);  
8     delay(500);  
9     digitalWrite(TX, LOW);  
10    delay(500);  
11    return 0;  
12 }
```

Cloud Console



Open Console

Select Device



The screenshot shows the Particle Console interface in a web browser. The browser tabs are 'Particle Build' and 'Particle Console | Build your cloud'. The address bar shows 'Secure | https://console.particle.io/devices'. The page title is 'Devices'. On the left is a sidebar with icons for 'Particle', 'Build', 'Devices', 'Flows', 'Logs', 'Settings', and 'Help'. The main content area displays a table of devices. The table has four columns: 'ID', 'Type', 'Name', and 'Last Handshake'. There are eight rows of Photon devices listed. A blue speech bubble points to the first row with the text 'Select your device'.

ID	Type	Name	Last Handshake ⓘ
● 2c003...	Ⓟ Photon	Bill	4/1/18 at 9:26pm
● 2c004...	Ⓟ Photon	Lemon	4/1/18 at 7:57pm
● 2d003...	Ⓟ Photon	Orange	4/1/18 at 7:52pm
● 23003...	Ⓟ Photon	Tangerine	4/1/18 at 7:47pm
● 38002...	Ⓟ Photon	Grape	4/1/18 at 7:42pm
● 31004...	Ⓟ Photon	Plum	4/1/18 at 7:35pm
● 23003...	Ⓟ Photon	Peach	4/1/18 at 7:26pm

Call Function

The screenshot displays the Particle Console web interface in a browser. The address bar shows the URL `https://console.particle.io/devices/2c0031000a47343432313031`. The interface is divided into several sections:


- Header:** Includes the Particle logo and navigation links for "Devices" and "View Device". Action buttons for "PING" and "EDIT" are located in the top right.
- Device Information:** A central panel displaying:
 - ID: `2c0031000a47343432313031`
 - Name: Bill
 - System Firmware: 0.7.0
 - Type: Photon
 - Last Handshake: Apr 1st 2018, 9:26 pm
- Notes:** A text area on the right with the instruction: "Click the edit button to keep notes on this device, like 'Deployed to customer site'".
- Event Logs:** A section titled "EVENT LOGS" with a graph area that says "WAITING FOR EVENTS ..". Below the graph are buttons for "PAUSE", "SEE IN TERMINAL", and "PUBLISH EVENT".
- Functions:** A section titled "FUNCTIONS" showing a function named "blink = -1" with an "Argument" input field and a "CALL" button.
- Variables:** A section titled "VARIABLES" with the message "No variables found on device."

Call Function

NOTES

Click the edit button to keep notes on this device, like 'Deployed to custom device'.

FUNCTIONS

f blink = -1 

Argument

CALL

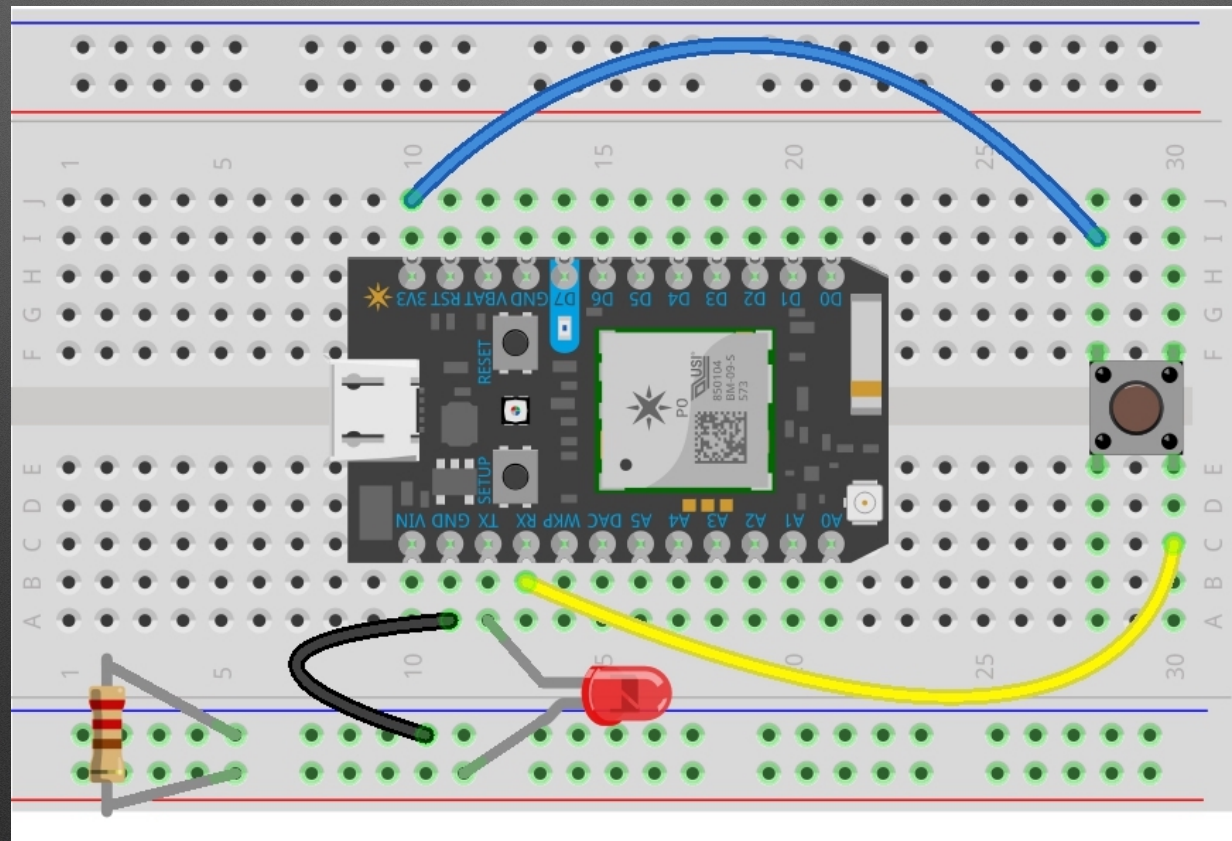
Hit "Call" to call it

VARIABLES

No variables found on device.

0
09:33:20 PM

Add a Button



Add a Cloud Variable

```
1  bool closed;  
2  
3  void setup() {  
4      pinMode(TX, OUTPUT);  
5      Particle.function("blink", blinkFunction);  
6  
7      pinMode(RX, INPUT_PULLDOWN);  
8      Particle.variable("closed", closed);  
9  }  
10  
11 void loop() {  
12     closed = digitalRead(RX);  
13 }  
14  
15 int blinkFunction(String arg) {  
16     digitalWrite(TX, HIGH);  
17     delay(500);
```

Reload Console

The screenshot displays the Particle Console interface for a specific device. The browser address bar shows the URL `https://console.particle.io/devices/2c0031000a47343432313031`. The interface includes a sidebar with navigation icons, a top navigation bar with links to Docs, Contact Sales, Support, and a user profile. The main content area is titled "Devices > View Device" and features a "PING" button and an "EDIT" button. The device details section shows the ID `2c0031000a47343432313031`, Name `Bill`, System Firmware `0.7.0`, Type `Photon`, and Last Handshake `Apr 1st 2018, 9:48 pm`. A "Notes" section contains a message about keeping notes on the device. The "EVENT LOGS" section is currently empty, displaying "WAITING FOR EVENTS." Below this are buttons for "PAUSE", "SEE IN TERMINAL", and "PUBLISH EVENT". A table with columns "EVENT NAME", "DATA", and "PUBLISHED AT" is shown, with a tooltip indicating how to get events using `Particle.publish()`. The right sidebar contains sections for "FUNCTIONS" (with a "blink" function and "CALL" button), "VARIABLES" (with a `closed (bool) = true` variable and "GET" button), and "ACTIONS" (with an "UNCLAIM" button).

Particle Console | Build your own IoT

Secure | `https://console.particle.io/devices/2c0031000a47343432313031`

Docs | Contact Sales | Support | ccscpp@siever.info

Devices > View Device

PING EDIT

ID: `2c0031000a47343432313031` Name: Bill

System Firmware: `0.7.0` Type: `P` Photon

Last Handshake: Apr 1st 2018, 9:48 pm

Notes

Click the edit button to keep notes on this device, like 'Deployed to customer site'.

EVENT LOGS

WAITING FOR EVENTS.

PAUSE SEE IN TERMINAL PUBLISH EVENT

EVENT NAME	DATA	PUBLISHED AT

Get events to appear here by using `Particle.publish()` in your firmware ([docs](#))

FUNCTIONS

blink

Argument CALL

VARIABLES

closed (bool) = true

GET

ACTIONS

UNCLAIM

Type: **P** Photon

Click the edit button to keep notes on this device, like 'Deployed to customer site'.

TING FOR EVENTS .

PUBLISH EVENT

PUBLISHED AT


nts to appear here by using
publish() in your firmware ([docs](#))

FUNCTIONS

 blink

Argument


VARIABLES

 closed (bool) = true



GET

ACTIONS

 UNCLAIM

New!
Hit Get while pushing/not
pushing button

An “App” for that: A quick tour

A “real” IoT application

Questions / Discussion



Apps for Data Logging: bittydatalogger

Program

on start

* bluetooth accelerometer service

* bluetooth temperature service

Concepts

- Data Formats (CSV vs. JSON)
- Data Analysis

Remove Add Bluetooth

(& remove Radio)

