

Micro:bit Magic

Engaging Computer Science Activities for Grades 5-12
Sponsored by Washington University's Institute for School Partnership

Bill Siever
Principal Lecturer
Washington University in St. Louis

Outline

- Intros: Us, You, the micro:bit
- ★ Setup & Swag
- ★ “Hello, World!”: First Program
- ★ Programming: Logic & Action
- ★ Broadcast Basics & Firefly Fun
- ★ Motor Mayhem & Awesome Audio

Me: Background (degrees & career training) Computer Science & Computer Engineering. I’m a Lecturer here, so my main roll is teaching (but I don’t have a formal background in education nor do I have deep experience with K-12)

Lunch!

Outline

- Beyond Blocks: Bootstrapping Text-based languages
- ✦ Bluetooth Basics & Phone Phun
- Quick Tour: C++ & IoT Insanity
- Conclusions

- ## Intros: Us & You

- Us

- You

- **You: Stand If you work in a...**

- *Middle* school

- Teaching

- Other

- *High* school

- Teaching

- Other

- **You: Stand If You Teach...**

- AP C.S. Principles
- AP Computer Science A/B
- Arduino
- Raspberry Pi

- **You: Stand If You Teach/Use...**

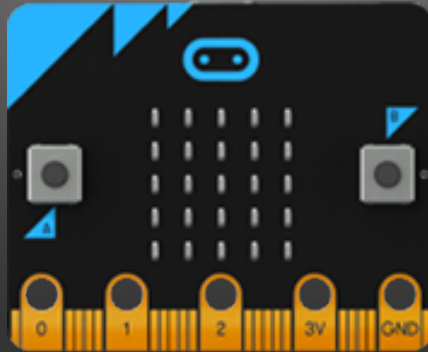
- Programming
 - Scratch
 - Java
 - Python
 - HTML+CSS & JavaScript

• Intros: You

- You: Intros
 - Name, School, Why here?
- Pair programming — pair up!

Intros: the micro:bit

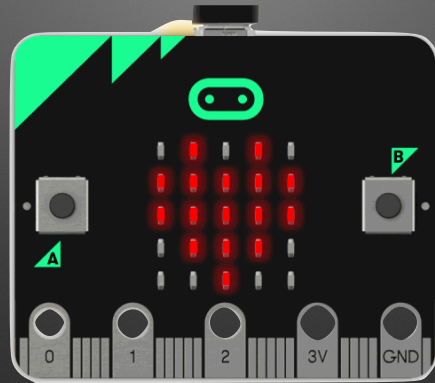
Small



5cm x 4cm

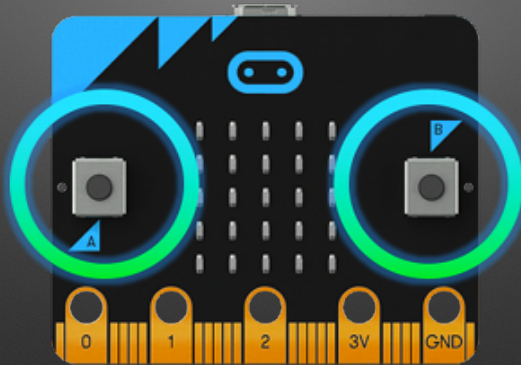
Artwork source: <http://microbit.org/images/microbit-features-temp.png>

LED Grid



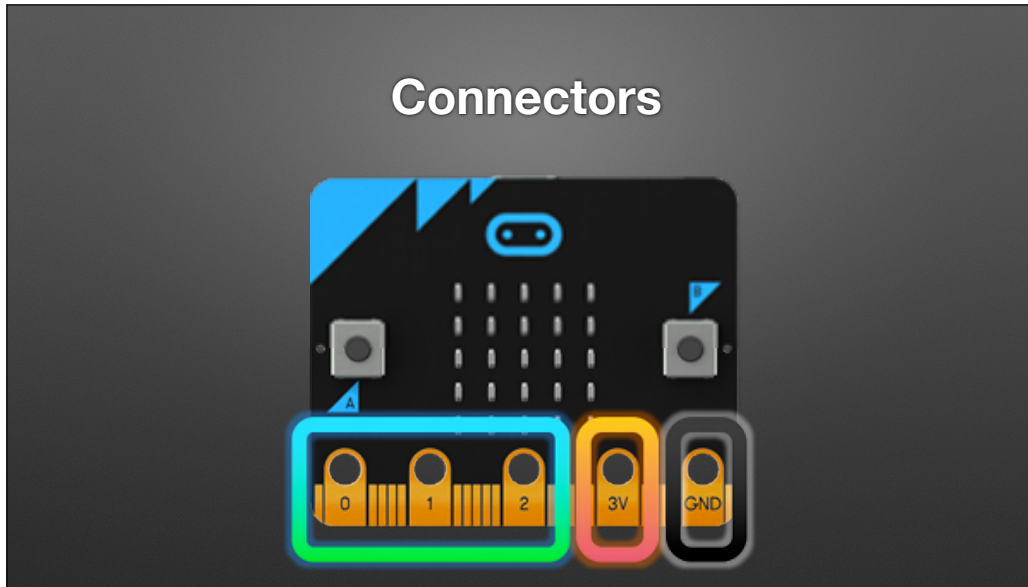
5xm x 4cm

Buttons



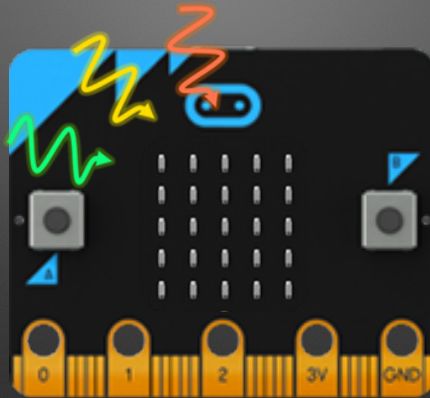
Artwork source: <http://microbit.org/images/microbit-features-buttons.png>

Connectors



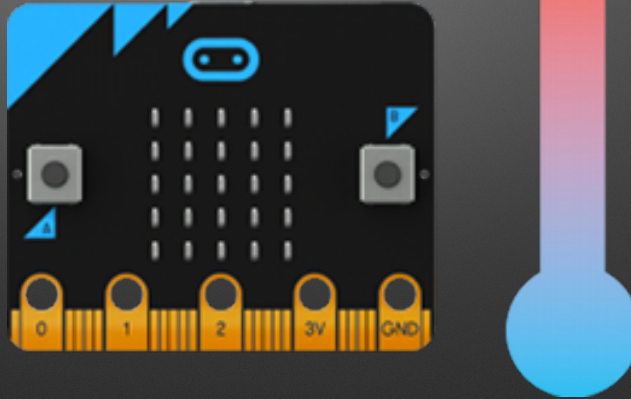
Artwork source: <http://microbit.org/images/microbit-features-pins.png>

Light Sensor



Artwork: <http://microbit.org/images/microbit-features-light.png>

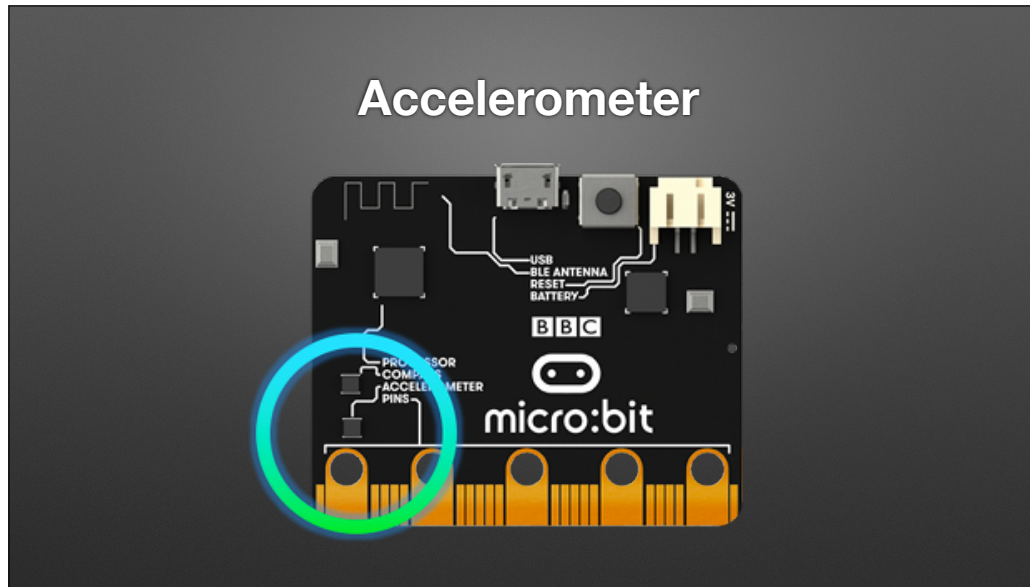
Temperature Sensor



Within about 2 degrees C (die temperature)/3.6 degrees F.

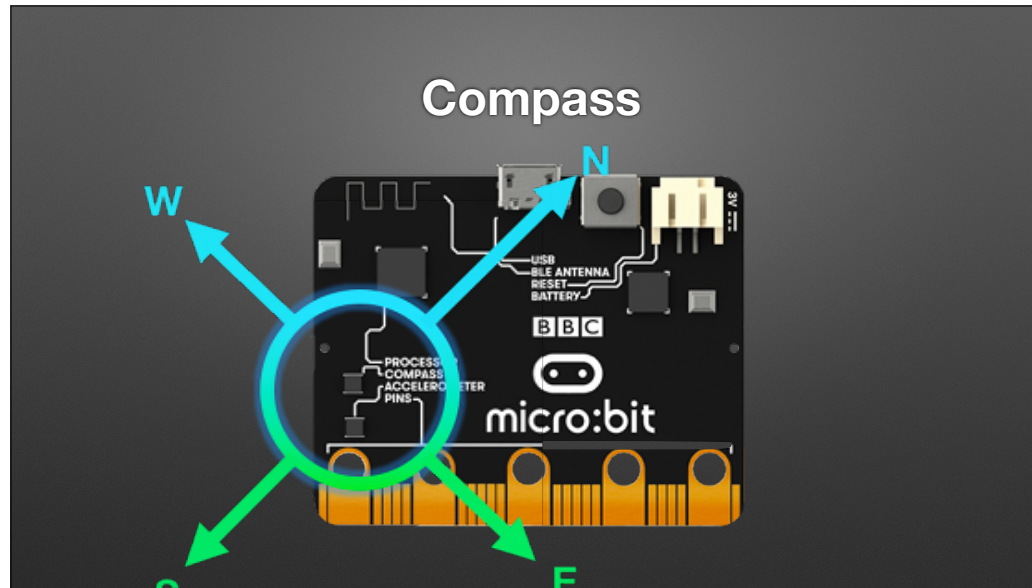
Artwork source: <http://microbit.org/images/microbit-features-temp.png>

Accelerometer



Detect/respond to tilt/tip/shake/etc.

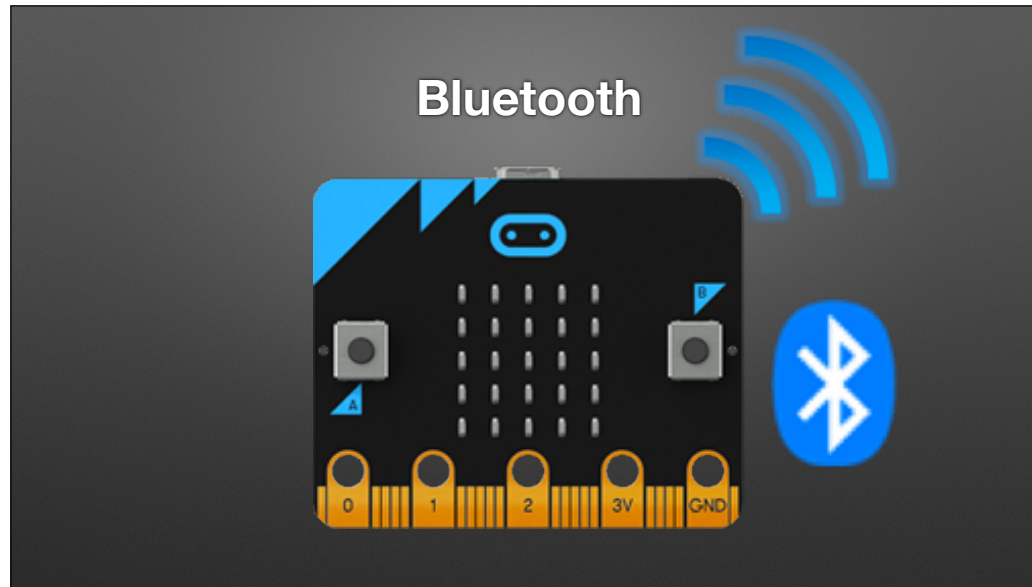
Artwork source: <http://microbit.org/images/microbit-features-accelerometer.png>



Artwork source: <http://microbit.org/images/microbit-features-compass.png>
(ugh. A little awkward to use)



“Broadcast”

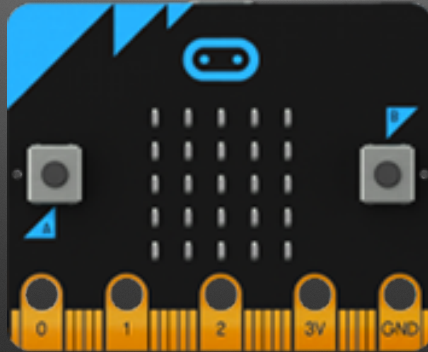


Bluetooth: It can talk to mobile devices!!!

(Mutually exclusive with broadcast)

Artwork source: <http://microbit.org/images/microbit-features-bluetooth.png>

Low Cost: ~\$13 US



Currently \$9 at micro:center here in town.

Artwork source: <http://microbit.org/images/microbit-features-temp.png>

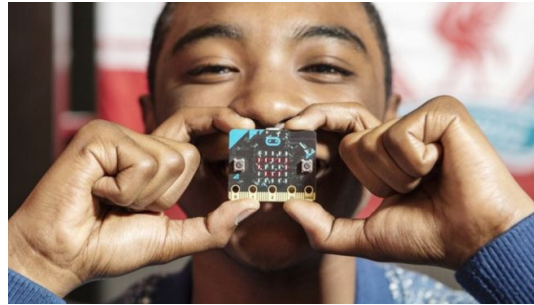
Background via Micro:bit Educational Foundation

and Hal Speed

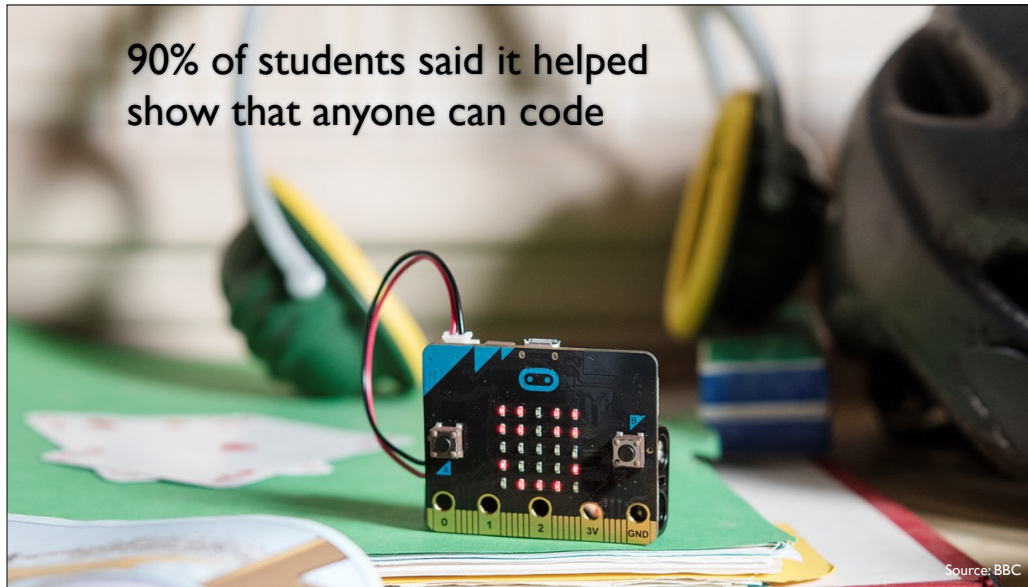
Thanks to The Micro:bit Educational Foundation and Hal Speed for the following 9 slides. (Hal is Chief of Global Engagement; Micro:bit foundation is a non-profit)

2015

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

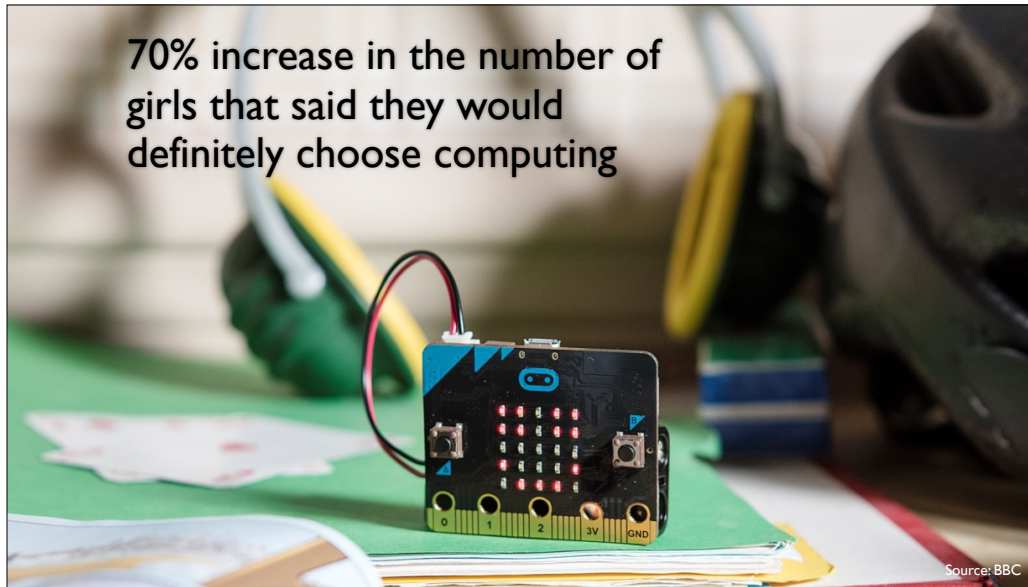


90% of students said it helped
show that anyone can code



Empowering students / enabling diversity in STEM (hopefully)

70% increase in the number of girls that said they would definitely choose computing



2016 Micro:bit Educational Foundation Formed

To empower children, parents and
teachers around the globe to learn
and innovate using the micro:bit

2017

micro:bit available in the U.S.



10 New & Innovative EdTech
Products Announced at ISTE 2017

27

© Micro:bit Educational Foundation 2018
@microbit_edu @HalSpeed


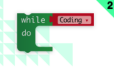





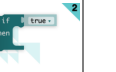

<http://www.gettingsmart.com/2017/06/10-innovative-new-products-announced-at-iste-2017/>







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

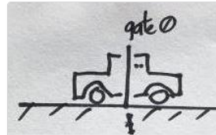


4 th Grade				5 th Grade			
Course E				Course F			
							
Lesson Course E - Loops 1 - Loops and Animations	Lesson Course E - Loops 2 - Nested Loops and Scoreboards	Lesson Course E - Functions 1 - A Simple Function for a Superhero	Lesson Course E - Functions 2 - Functions for a Digital Pet	Lesson Course 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
28	© Microbit Educational Foundation 2018			http://microbit.org/teach/code-org-fundamentals/			
@microbit_edu	@HalSpeed						

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

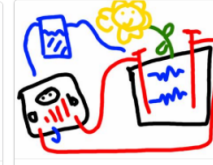
STEM Lessons



Timing gates



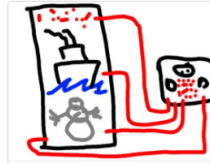
Soil Moisture



Plant Watering



Reaction Time



States of Matter

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

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

Disclaimer...

- The micro:bit is just a *platform*
 - I like it...
- The motivation and spirit behind it are compelling
- It's cheap and readily available now
- There are a lot of supporting resources

I was working with the processor used by the micro:bit well before the micro:bit existed. It's a prominent processor used in many Bluetooth gizmos, like some FitBits. I have a lot of experience developing bluetooth devices/software/apps.

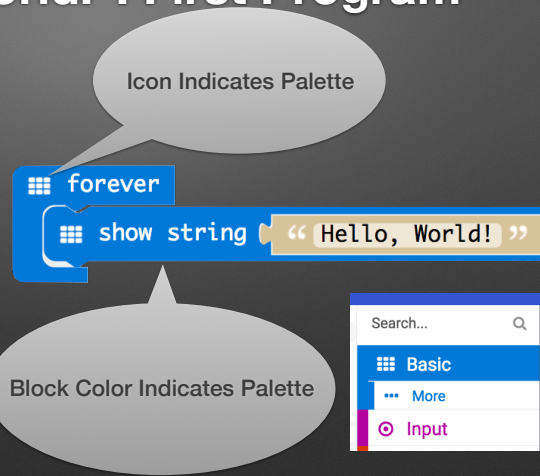
Campus Network

- SSID: wustl-guest-2.0



“Hello, World!”: First Program

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





Browser Setup

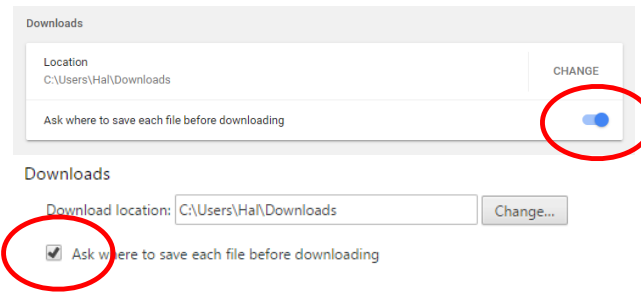
- Browser
 1. Open microbit.org
 2. Select “Let’s Code”
 3. Click “Let’s Code” button

★ Hardware Setup & Programming

- Hardware Handout — *Thanks to Wash U's Institute for School Partnership!*
 1. Pull out the micro:bit box / open
 2. Connect via USB cable
- Find browser Download Location
 - Micro:bit acts like a flash drive. It's programmed by dropping files on it

Chrome Setup

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



Personalization!

- Hello Bill / Hello

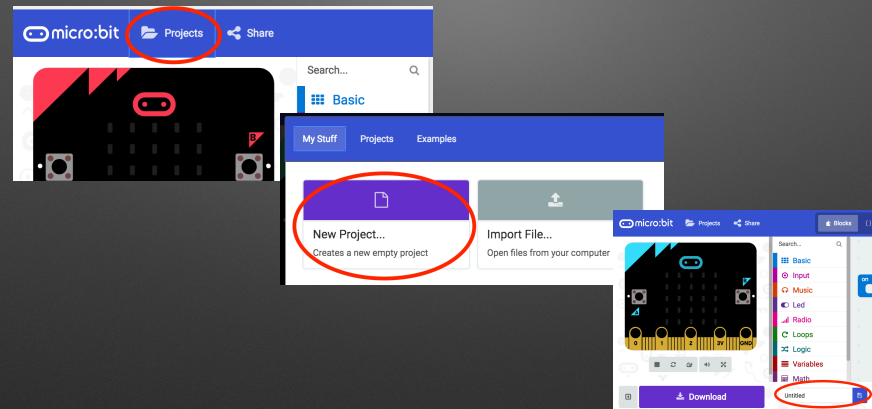
⌘ forever

⌘ show string “ Hello Friends! ”

Build a Marquee / Name Badge / Etc.!

- Re-program the micro:bit to scroll *your name*
 - Changing code in simulator *does not* change real micro:bit
 - Need to drag/drop each update
- Real micro:bit retains program until re-programmed
 - Try the battery pack!

New Project: Projects > New Project...



Name it “PickOne” (or something else)

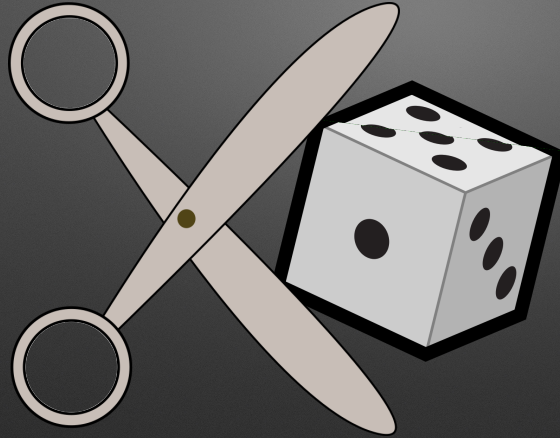


Programming: Logic & Action

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

CS...Int division; Mod; Etc.

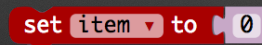
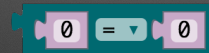
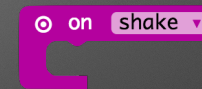
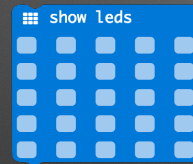
Obvious Solution...



<https://openclipart.org/detail/17370/a-die>

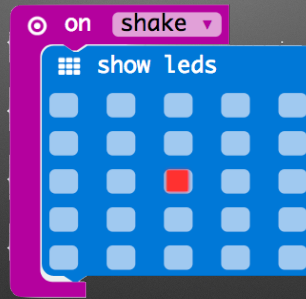
<https://openclipart.org/detail/19632/scissors>

Parts



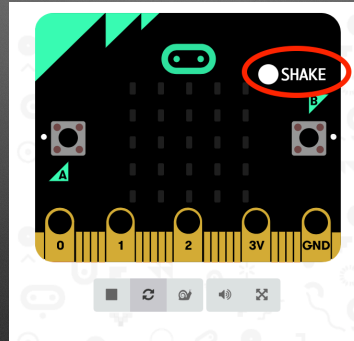
1. Color indicates Palette
2. Incremental Development:
Try parts in Simulator

Trying parts: Shaking things up



1. Test the “shake” construct (in the simulator)

Trying parts: Shaking things up



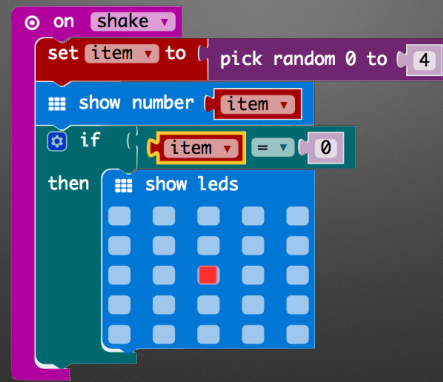
Notice the simulator now has a “shake” button to simulate shaking

Trying parts: Testing Random & Range



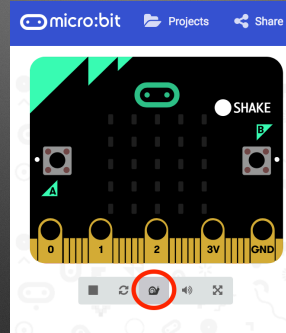
2. Try to understand variable / random number

Trying parts: Basic Logic / Selection



2. Try to understand variable / random number

Turtle Mode: Improving Execution Understanding

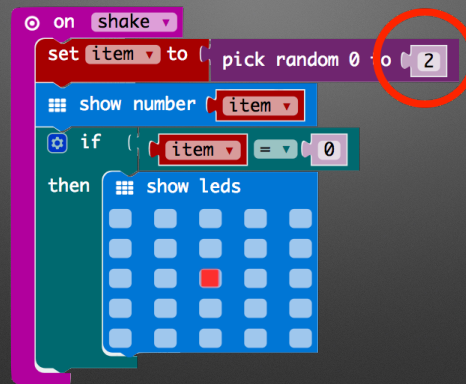


Use “Turtle mode” to better follow execution.

Notice how blocks are highlighted as they execute.

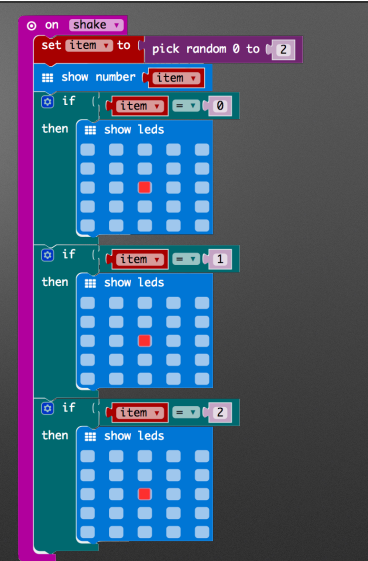
In particular, notice the behavior of the “body” of the if-statement (when/if the “show LEDs” block is highlighted)

Trying parts: Fixing Logic



2. Try to understand variable / random number

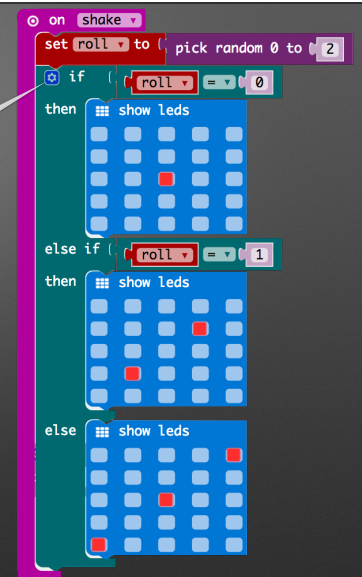
Completing Logic



2. Try to understand variable / random number

Solution

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



Currently



Blocks have squarer edges

If-Block options control

Next Major Update

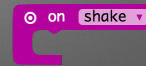


Data Types: Oval for numbers

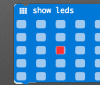
Data Types: Hexagons for boolean

Concepts

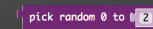
- Event driven programming



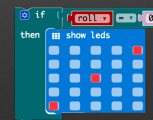
- Bitmapped Graphics



- Ranges & Representations



- Logic

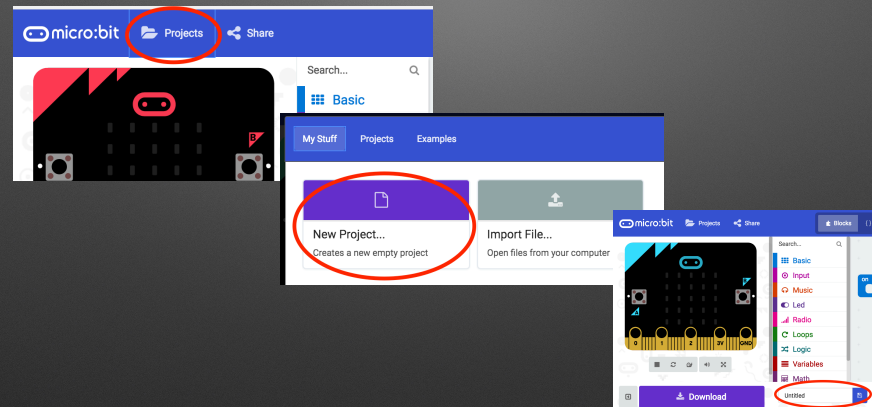


Pedagogy

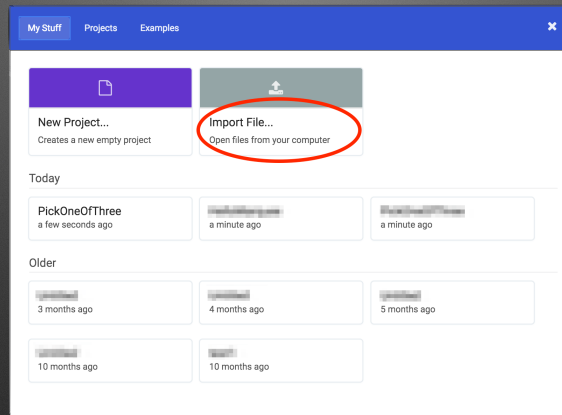
- Active Learning
- Discovery Based
- Constructionist

Great...but all concepts can be done with scratch.

New Project: Projects > New Project...



Resource Management



Notice the history. They are tied to the browser.

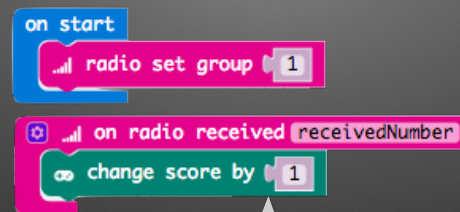
Drag & Drop Works Too!



Broadcast Basics

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

Receiver

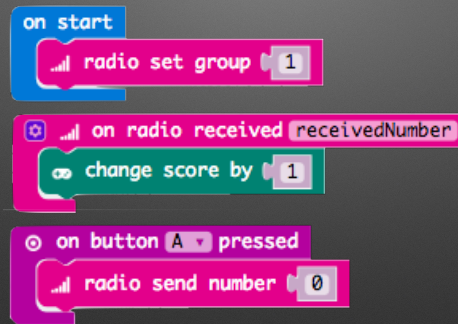


"Game" blocks in "Advanced"
Section of Palette

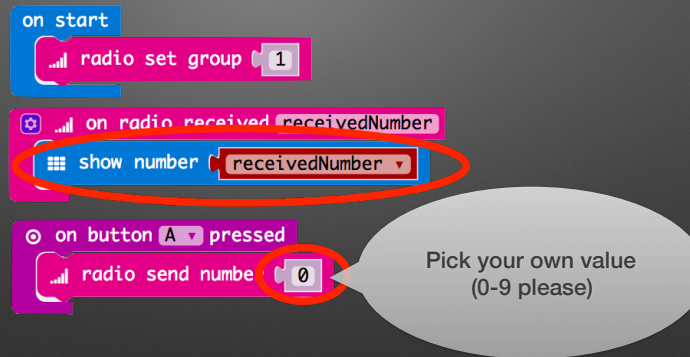
▼ Advanced

Program micro:bit

Full Boadcaster



Broadcast Data Content



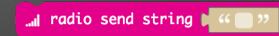
Private Texting!

1. Form a group &
Communicate with just your group!

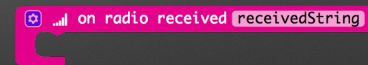


radio set group 1

2. Update to send/receive TEXT!

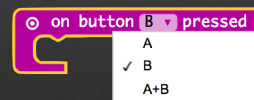


radio send string " " " "



on radio received receivedString

3. Use different buttons for different messages!

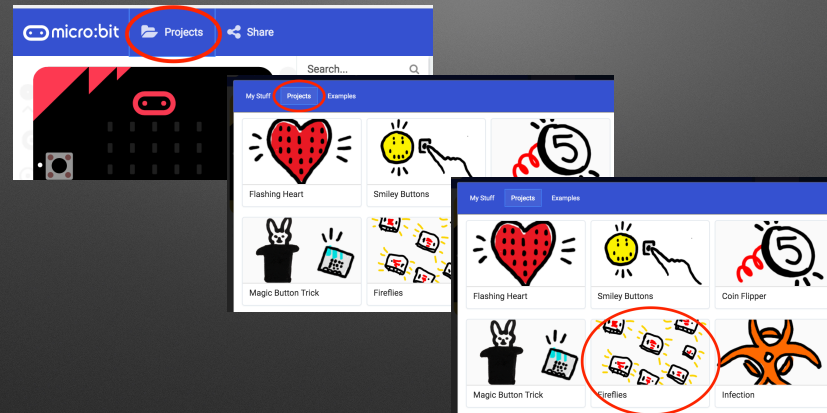


on button B pressed

- A
- ✓ B
- A+B



& Firefly Fun



Concepts

- Broadcasting

radio send number 0

- Network Addresses

radio set group 1

- Asynchronous clocks / Sync problems

on radio received receivedNumber

Broadcasting can be done with Scratch

Goody Bag: Hardware



Motor Mayhem

An Intro to Servos
(Unfortunately, very poor quality servos...)



Motor Mayhem

An Intro to Servos
(Unfortunately, very poor quality servos...)

on button **A** pressed

servo write pin **P0** to **120**

on button **B** pressed

servo write pin **P0** to **10**

Inchworm Insanity

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

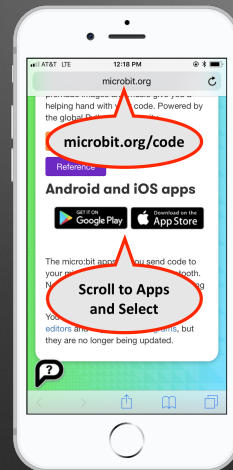


Awesome (?) Audio

Concepts

- I/O
- Basic Electric Circuits/Electronics

Lunch Break & App Install Android & iOS



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

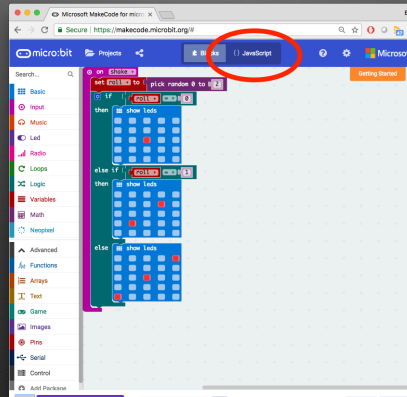
Artwork: <https://mockuphone.com> and microbit.org site

Prep the Pick One Application

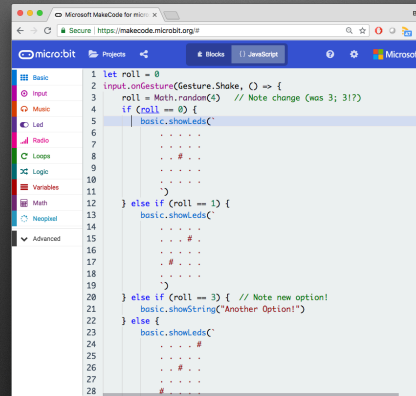
Beyond Blocks: Bootstrapping Text-based languages

- The editor translates all the code to static TypeScript
 - TypeScript is superset of JavaScript
 - Gets translated/compiled in the browser to machine code
- Easy to show the correlation from blocks to structured text!

Beyond Blocks: Bootstrapping Text-based languages



Beyond Blocks: Bootstrapping Text-based languages



The screenshot shows the Microsoft MakeCode for micro:bit IDE. The left sidebar contains a category list: Basic, Input, Music, Led, Radio, Loops, Logic, Variables, Math, Neopixel, and Advanced. The main editor area displays a JavaScript script for a random number generator. The script is as follows:

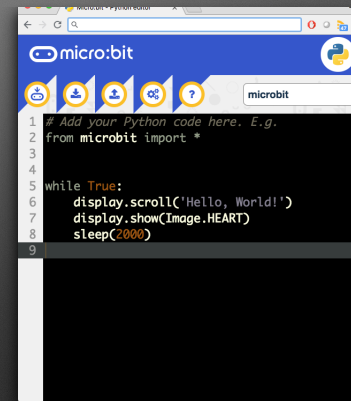
```
1 let roll = 0
2 input.onGesture(Gesture.Shake, () => {
3   roll = Math.random() * 3 // Note change (was 3; 31?)
4   if (roll == 0) {
5     basic.showLeds(`
6       . . . . .
7       . . . . .
8       . . . . .
9       . . . . .
10      . . . . .
11    `)
12   } else if (roll == 1) {
13     basic.showLeds(`
14       . . . . .
15       . . . . .
16       . . . . .
17       . . . . .
18       . . . . .
19    `)
20   } else if (roll == 3) { // Note new option!
21     basic.showString("Another Option!")
22   } else {
23     basic.showLeds(`
24       . . . . .
25       . . . . .
26       . . . . .
27       . . . . .
28       . . . . .
29    `)
```

Text-based languages: Beyond Blocks

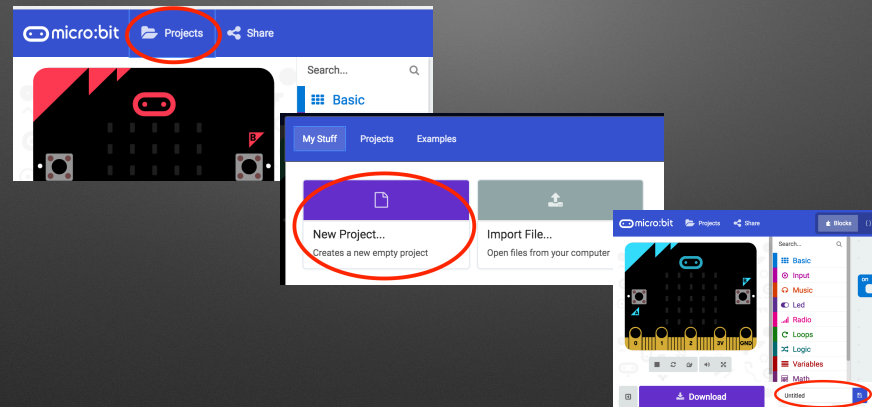
```
2 function pickInclusive(start: number, end: number) {  
3   return Math.random((end - start) + 1) + start  
4 }  
5  
6 input.onGesture(Gesture.Shake, () => {  
7   roll = pickInclusive(-2,3)
```

Text Based Languages: Micro Python


- <http://python.microbit.org/>



New Project: Projects > New Project...



Bluetooth Background

- Uses different protocol than  Radio
- Not a group broadcast

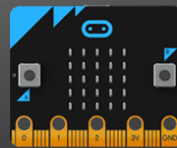
Bluetooth CAN do a type of broadcast, but that's not how the micro:bit uses bluetooth.

Bluetooth Background

Central



Peripheral



Bluetooth Background

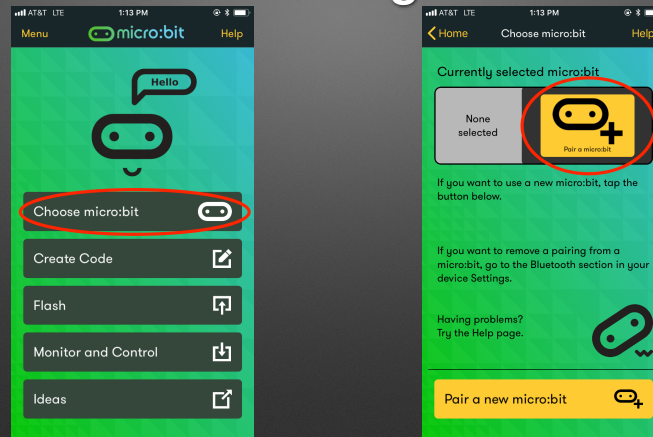


Bluetooth Background

- 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)

The data being exchanged is usually “low risk”

Pairing



Follow instructions to pair.

But you said we wouldn't use pairing...

- Pairing stores info. on both the micro:bit and the phone
 - This info. is created when they first pair and is unique
 - It must match for the devices to be able to connect securely
- Reprogramming the micro:bit erases its pairing information
 - But the Phone still retains its pairing information
 - The pairing details will be out-of-sync & they will be unable to connect

Micro:bit app (iOS)

- The only way for the app to identify a micro:bit by going through the pairing process
- The app retains the identify of the micro:bit even if the micro:bit no longer needs to be paired
- I.e., the “pairing” was just done to let the App know about the micro:bit

But you said we wouldn't use pairing...

- Every time you re-program the micro:bit it erases its pairing information but your phone does not forget the pair.
- Option A: Re-pair every time you re-program:
 1. Open bluetooth settings on phone and “Forget” the micro:bit
 2. Use the App to pair again

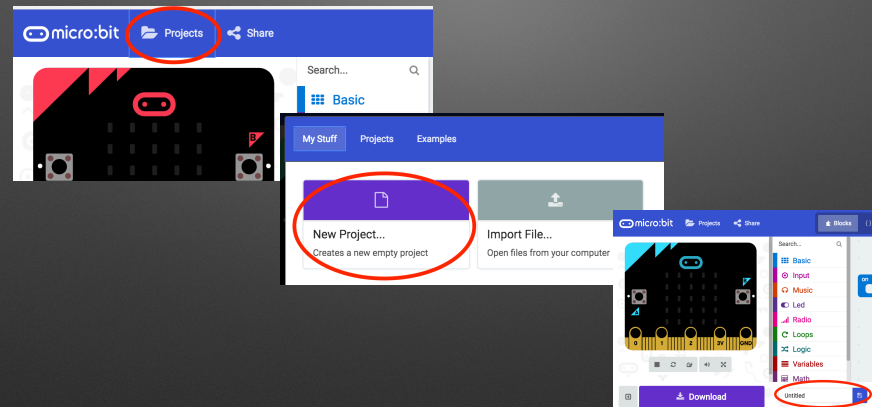
But you said we wouldn't use pairing...

- Every time you re-program the micro:bit it erases its pairing information but your phone does not forget the pair.
- Option B (today):
 1. Pair once so the app identifies the micro:bit
 2. Disable pairing on the micro:bit

Misc. on Pairing & Security

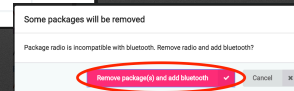
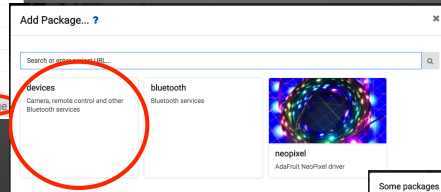
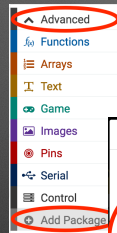
- Opportunity to discuss security issues & concepts
- The previous example was done on the official micro:bit app.
- Other apps may or may not require pairing
(many don't need to be "tricked" by pairing once)

New Project: Projects > New Project...

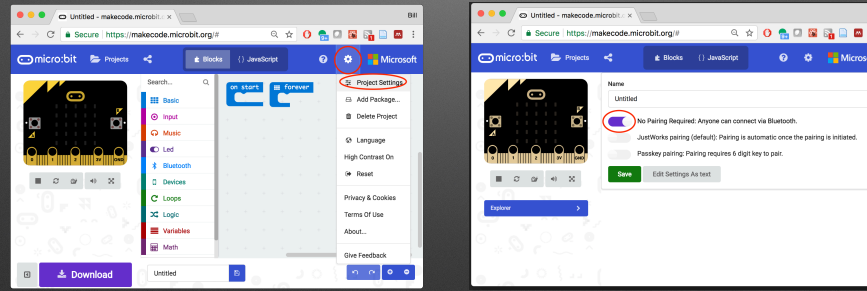


Add Devices

(& remove Radio)



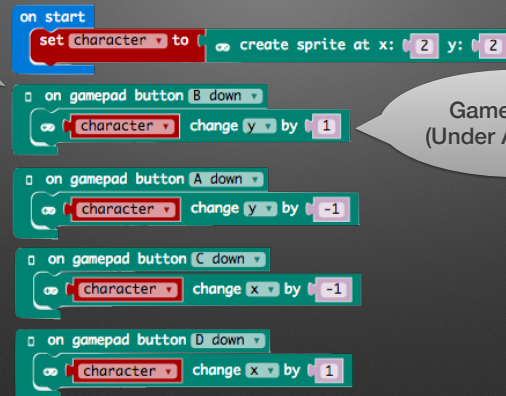
Project Settings: *Pairing Off*



Pairing only works prior to installing a bluetooth sketch.
May need to re-load a blank sketch and then start pairing process.

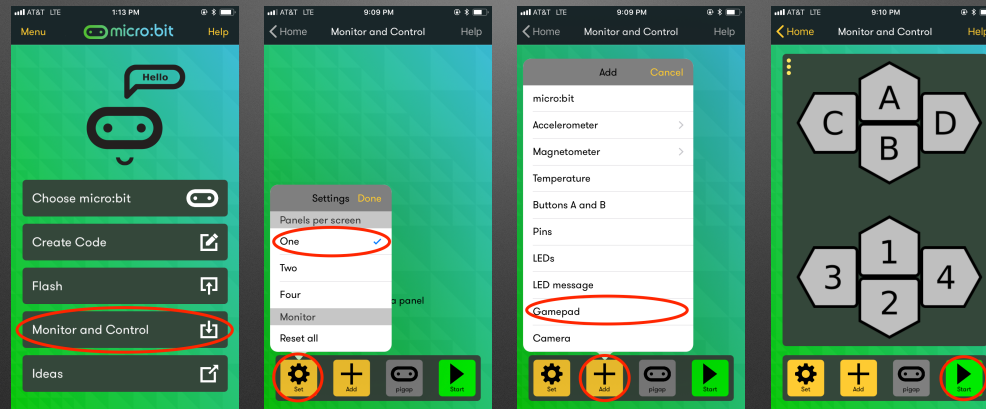
Program

Devices Palette



Game Palette
(Under Advanced)

App Configuration



Apps for Data Logging: bittydata logger

(This app doesn't require the "pairing once" trick)

Program

```
on start
  * bluetooth accelerometer service
  * bluetooth temperature service
```

Demo App / Collecting Data

Applications: Citizen Science (measuring vibration on a bridge...Frisbee being thrown...etc.

Concepts

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

Can also be used with Mobile Development

- Simple data logger program (on the micro:bit)
 - Provides a standard “service”
 - Very easy to write iOS and Android apps that access the data

Play Time!

- Peer-up
 - Form groups of 2-4 based on a common bond. Either:
 - Students are near the same ages
 - Teaching the same topics
 - Same level of comfort with everything today

Challenge: Lesson-plan-athon!

- Inspiration: Hackathons & Programming Competitions
- Review and try resources
- I'll walk around to help where/how I can
 - Time will be pseudo-structured / timed
- You'll (informally) report-out at the end to share your findings

Part 1: Finding and Prioritizing Potential Activities

**Part 2-3: Play! Try them out! (and/or Create)
Try 2+ (and then, optionally, brainstorm and
create new activity)**

Part 4: Reprioritize & *Report*

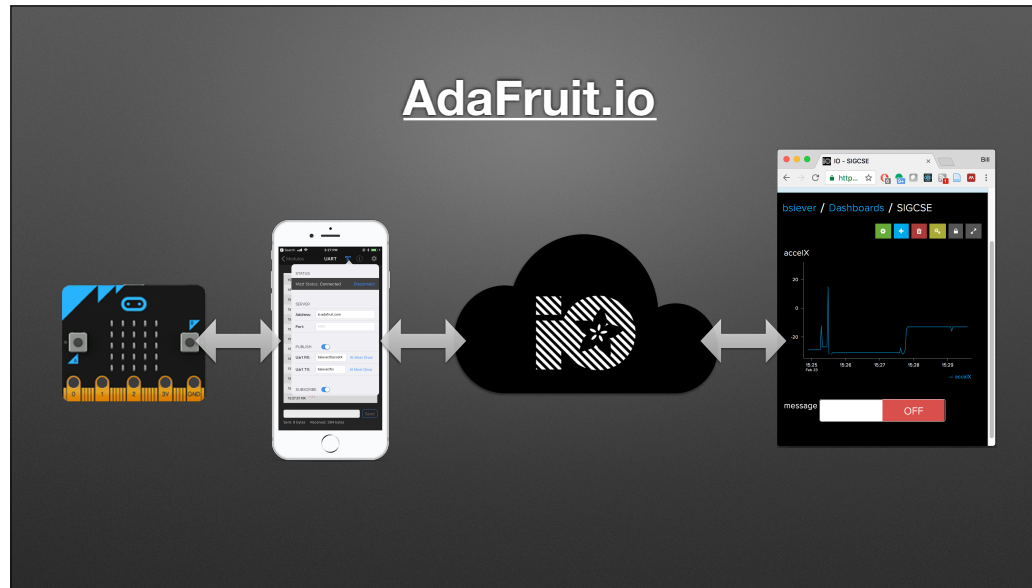
<http://bit.ly/microbitreport>

**<http://bit.ly/washumicrobit>
(microbit report)**

Report Out!

Question: What about more advanced topics?

IoT Example



<https://cdn-learn.adafruit.com/guides/images/000/001/691/medium800/Adafruit-IO-Logo.png>

Misc.

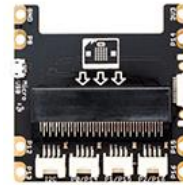
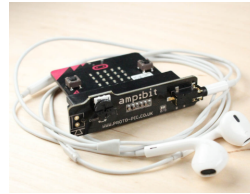
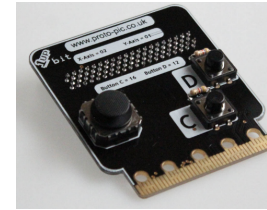
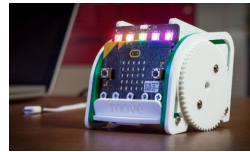
- C (C++) / Arduino
- Phone acts as border router
- [AdaFruit.io](https://adafruit.io) can tie into other services
 - Texts/Notifications (IFTTT), Webhooks, etc.

Demo

<https://io.adafruit.com/bsiever/dashboards/sigcse>

Other Hardware





U.S. Resellers



||| © Microbit Educational Foundation 2018
@microbit_edu @HiSpeed

<http://microbit.org/resellers/>



Available via DonorsChoose.org



- AKJ Education is an approved DonorsChoose.org vendor and micro:bit reseller
- Teachers enter projects and request classroom materials
- Individuals and companies can donate money towards the purchase of those materials

Odds & Ends

- **Address Safety!**

- Low voltage / low current vs. Mains power

Odds & Ends: Weirdness & Weaknesses

- Currently no floating point numbers
 - Great for practicing Integer concepts
 - Not a limitation of the board; Expected to be included in next major update
- Support for Functions is weak using blocks
 - Motivates JavaScript (or Python)

Odds & Ends: Micro:bit vs. ...

- Scratch: Probably a better choice for block-based intro to programming
 - But micro:bit can extend that to hardware & the physical world
- Raspberry Pi is a much more complex computer w/ lots of software choices; They are good for things like webcams
 - But micro:bit is good for some simple motors/motion/communication examples
- Lego: Expensive, but great at “mechanical” things

Questions / Discussion

Thanks 1: Thanks for attending!

Thanks 2: Thanks ISP!

- Special thanks to Washington University's Institute for School Partnership (<https://schoolpartnership.wustl.edu/>)
 - Jaime Gilligan & Chris Mohr
- All arrangements
- Supplies (Food! Swag bags!)

Bill's Blog Post
<http://bit.ly/washumicrobit>
(WashU Micro:bit)

Eval form (to compete certificate)

Remove Add Bluetooth

(& remove Radio)

