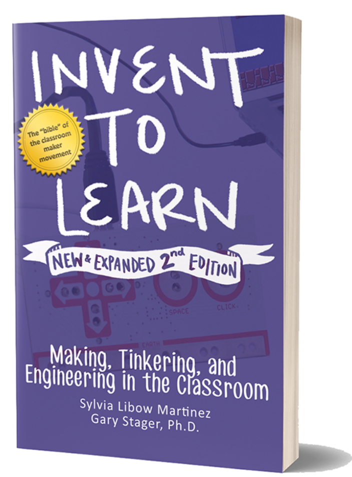
INVENT TO LEARN



Lots of great ideas at inventtolearn.com

Where educators invent the future!

Constructing Modern Knowledge

July 16-19, 2019 Manchester, NH USA

constructingmodernknowledge.com

Summer Institute for Creative Educators

Constructing Modern Knowledge, July 16-19, 2019 in Manchester, NH, USA is a life-changing educational experience for PK-12 educators and teacher educators. Participants enjoy 4 days immersed in remarkable project development with a mountain of materials and supported by a world-class faculty. Interaction with amazing guest speakers and a visit to the legendary MIT Media Lab round out this unique institute.

Send a team at discounted rates for maximum education impact!

Knowledge is a consequence of experience and CMK 2019 provides everything needed to help educators invent the future including:

- Well-stocked professional library
- Arts & craft supplies
- Microcontrollers including Arduino & micro:bit
- Hummingbird Robotics
- littleBits
- Programming software
- LEGO & EV3
- Raspberry Pi computers
- 3D printers
- Wearable computing and e-textiles
- Photographic, music, & video equipment

constructingmodernknowledge.com

Invent To Learn

Making, Tinkering, and Engineering in the Classroom

www.inventtolearn.com

What's in a Makerspace?

A well-equipped modern makerspace can feature flexible, computer-controlled manufacturing equipment for creating, cutting, and forming plastics, metal, plaster, and other common materials, including:

- 3D printers that are capable of producing three-dimensional objects (additive machines).
- Cutting machines that cut a variety of materials with precision. The cutting element can be a laser, water jet, knife, or other material (subtractive machines).
- Milling and routing machines that drill and shape complex parts.
- Joining machines that use computer control to sew, weld or bond in other ways.
- Traditional hand and power tools, including soldering irons.
- Decorative materials for painting, embroidery, and embellishing projects.
- Computers for programming, designing, and control.

What Else Do I Need?

Your makerspace – whether in your classroom or any other space – should include support materials and inspiration for your students. There should be room to move, build, and think alone and together. Well-stocked classroom libraries, supplies, gadgets, technology, tools, toys, recycled materials and other assorted stuff within an arm's reach of students are learning accelerants. Eleanor Duckworth reminds us, "If materials are slim, the only questions likely to be posed are the teacher's."

Resources to Explore

Invent To Learn Resources – Find all the resources from the book online, plus more!

Sparkfun Electronics – Specializes in hobbyist electronic components, tools, and kits. The website features a project blog, buying guides, and tutorials from soldering to using a breadboard. Ask for educator discounts. http://www.sparkfun.com

Adafruit – "Unique and Fun DIY Electronics and Kits." Check the Young Engineers section for selections that are whimsical and fun. Ask for educator discounts. http://www.adafruit.com

Maker Shed – Vetted by Make Magazine. Specializes in kits for learners of all ages http://www.makershed.com/

Electronics Goldmine – Extensive supply of cheap electronic parts. Look for deals and bulk purchases, for example, a container of 200 LEDs costs $5.00. http://www.goldmine-elec.com/

Jameco Electronics – Another popular source of electronics parts and kits. http://www.jameco.com

Invent To Learn

Making, Tinkering, and Engineering in the Classroom

Cardboard Construction

Ideas to Try

- Build a cardboard city to accompany the reading of a favorite book.
- How about a cardboard Alamo or puppet theatre?
- Create sets for movies and stop-motion animation projects
- Use cardboard creations to illustrate science principles
- Cardboard automata
- Build cars, castles, a haunted house, maze, or even an arcade
- Make a robot with lights, sensors, or moving parts
- Make giant cardboard robot or monster costumes to wear and make a movie.
- Play!

Resources to Explore

The Story of Caine's Arcade (Video) – Watch as an imaginative 9-year old uses his summer vacation to construct an elaborate cardboard arcade. No one comes to play until a local filmmaker stumbles on it and engages the neighborhood to surprise Caine. The video has become an inspirational YouTube sensation and spawned a global foundation for creativity. http://cainesarcade.com

Makedo – Reusable connectors and hinges for turning cardboard packaging materials into elaborate structures and play objects. http://mymakedo.com

Rolobox – Wheel sets for cardboard boxes. Great for use with Makedo and other materials. http://www.rolobox.com.au

Cardboard Automata – Machines made of cardboard http://tinkering.exploratorium.edu/cardboard-automata/

The Caberet Mechanical Theatre - Projects, blogs, and kits for building whimsical moving mechanical sculpture. http://www.cabaret.co.uk/

Cardboard Institute of Technology - From the San Francisco Exploratorium http://tinkering.exploratorium.edu/cit/

30 Ways to Use a Cardboard Box - From the Chicago Children's Museum http://www.chicagochildrensmuseum.org/CCM_30_Ways_to_Use_a_Cardboard%20Box.pdf

Invent To Learn

Making, Tinkering, and Engineering in the Classroom

Squishy Circuits

Dr. AnnMarie Thomas and her students at the University of St. Thomas School of Engineering developed conductive and non-conductive dough for fun and safe electronics exploration.

Dough will be the consistency of Play-Doh. Stored in an airtight container or plastic bag, it will last a week or two.

Conductive Dough - Cooked

1 cup (240 ml) Water
1 cup Flour (plus extra for kneading)
1/4 cup Salt
3 Tbsp. Cream of Tartar
1 Tbsp. Vegetable Oil
Food Coloring (optional)

*9 Tbsp. of Lemon Juice may be substituted

1. Mix water, 1cup of flour, salt, cream of tartar, vegetable oil, and food coloring in a medium sized pot.
2. Cook over medium heat and stir continuously.
3. The mixture will begin to boil and start to get chunky.
4. Keep stirring the mixture until it forms a ball in the center of the pot.
5. Once a ball forms, place the ball on a lightly floured surface.
6. Slowly knead the remaining flour into the ball until you've reached a desired consistency.

Insulating Dough - Not Cooked

1 cup Flour (plus extra for kneading)
1/2 cup Sugar
3 Tbsp. Vegetable Oil
Food Coloring (Different color than the conductive dough)
1/2 cup Deionized (or Distilled) Water

(Regular tap water can be used, but the resistance of the dough will be lower.)

1. Mix flour, sugar, and oil in a pot or large bowl.
2. Add water one spoonful at a time and stir until absorbed.
3. Add water until dough is crumbly. This should be almost all the water.
4. Turn the dough onto a lightly floured board and knead into a ball.
5. If it won't hold in a ball, knead in more water a little at a time.
6. If it is too sticky, knead in more flour.

Play With It!

- Attach a 6V or 9V battery to two pieces of conductive dough.
- Add LEDs, a small motor, or a buzzer in between your dough pieces to complete your circuit. *Warning: don't connect an LED directly to your battery leads - you will hear a pop and burn it out.*
- Make sure you have a complete circuit from one battery terminal to the other, but don't directly connect the two battery leads. That will create a "short circuit."
- Use the insulating dough to decorate and keep the conductive dough from touching and making shorts.

More Resources

Squishy Circuits – The home of all things related to Squishy Circuits.

http://courseweb.stthomas.edu/apthomas/SquishyCircuits/

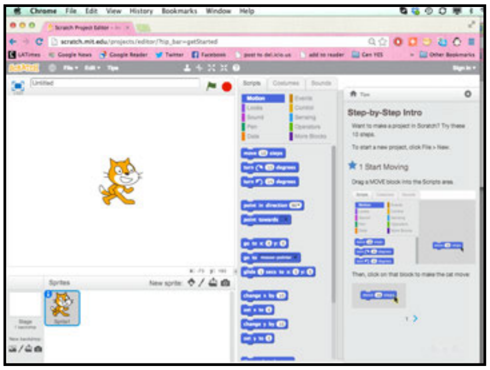
Sylvia's Super-Awesome Maker Show about Squishy Circuits – Super-Awesome Sylvia explains Squishy Circuits to kids of all ages. Shows photos of dough-making process.

http://makezine.com/2012/01/17/squishy-circuits-sylvias-mini-maker-show/

Invent To Learn

Making, Tinkering, and Engineering in the Classroom

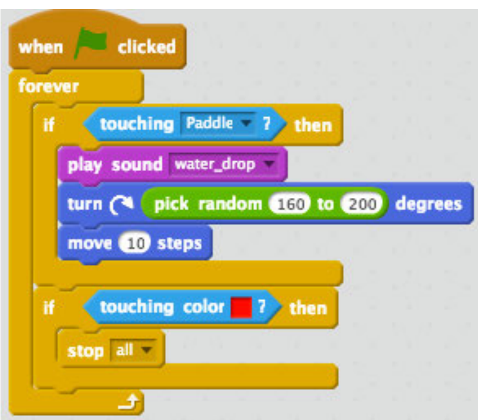
Scratch and Turtle Art



Scratch 2.0 in your browser

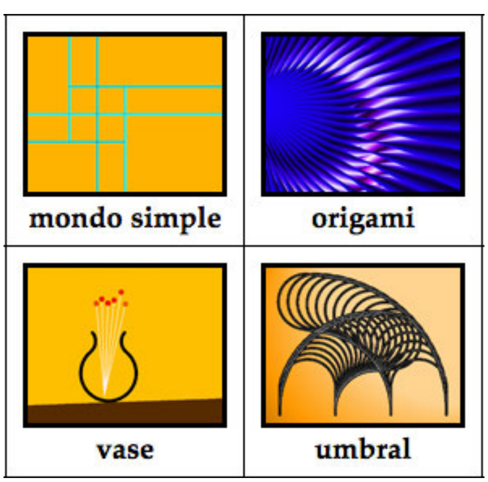
Scratch (Mac and Windows) was developed by the MIT Media Lab and is hands-down the most popular programming language ever created for kids. It's free, and the easy to use graphical interface features multiple turtles for turtle geometry, supports digital media integration, and projects may be published on the Web with the click of a mouse. Once published, other users may explore your creations, borrow some code or a cool sprite for use in their project, or remix the project. Scratch even keeps track of where each idea in a project came from so that authors are credited for their efforts. Scratch can be used with MaKey MaKey and LEGO WeDo robotics. See the *Invent To Learn* book for recommendations.

Now available - Scratch 2.0 works entirely in your browser and allows users to interact via gestures by using a laptop's camera or the Xbox Kinect. Scratch 2.0 projects can be created, run, and edited on any Flash-capable device. Today, you have to be connected to the Internet to use Scratch 2.0. An offline version is expected to release in late Summer 2013. Until then, use Scratch 1.4 if you need to have a version that works without being connected to the Internet. (**Note**: Scratch 2.0 programs cannot be run in Scratch 1.4) Robotics may not work in the Web-based version.



Part of a Pong Game

Turtle Art uses block programming like Scratch, but there is only one turtle, no multimedia, and projects are focused on the creation of beautiful art through the use of mathematics. The genius of Turtle Art is that you can drag and drop an image from the Turtle Art website into the software and the static picture includes the blocks that created it.



Turtle Art Gallery

Resources to Explore

- **Scratch website** – Download Scratch and explore projects http://scratch.mit.edu
- **Invent To Learn Resources for Programming** – More Scratch and Turtle Art resources including specialized versions for robotics and microprocessors http://www.inventtolearn.com/resources-programming/
- **Scratch educators** – Teachers of Scratch may find camaraderie, assistance, and project ideas at the ScratchEd website. http://scratched.media.mit.edu/
- **Turtle Art website** – http://turtleart.org

Turtle Art - Playing with Arithmetic

Problem 1
Create the following program:

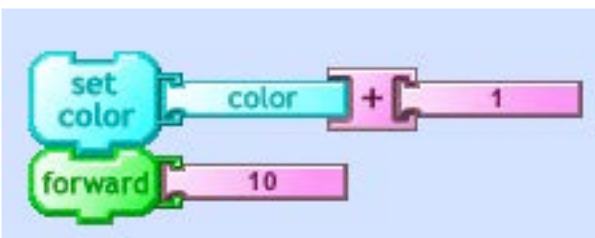
Can you predict what it will do before you run it?

What does it do?

What happens if you change the number 1 to another number?

What happens if you change the **X** to **+, - or /** ?

Problem 2
Create the following program:



Can you predict what it will do before you run this program?

How does it work?

What happens if you replace the 1 with a larger number, say 10?

When you increase the pen color by 1, does the color get lighter or darker?

What happens if you place a repeat block at the top of the program?

Problem 3
Here's a crazy idea!

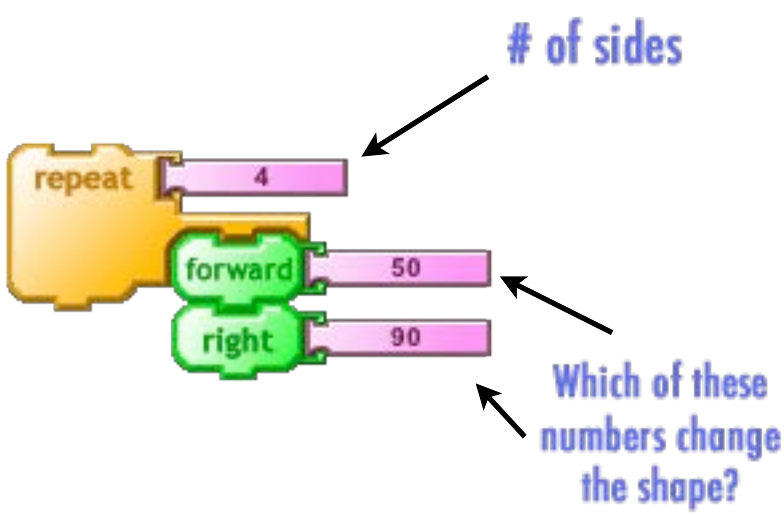
What do you predict will happen if you combine program 1 and program 2? Snap them together and fine out!

Making Polygons

Super Dooper Really Really Really Hard Challenge

Name	# of sides	amount of turn
Triangle	3	
Square	4	90
Pentagon	5	
	6	
	7	
Octagon	8	
	9	
	10	36
	11	
	12	

Change the number of sides and and amount of the turn to create the polygons.

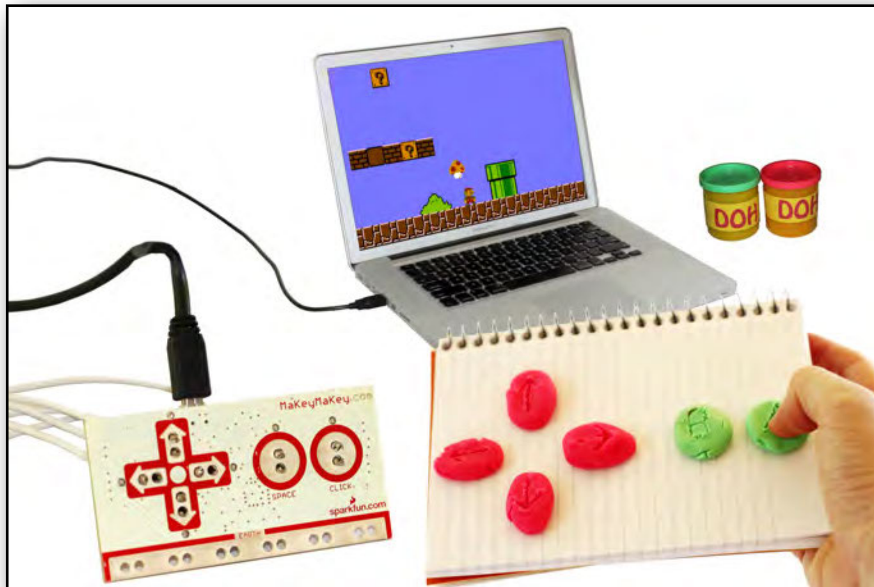


Invent To Learn

Making, Tinkering, and Engineering in the Classroom

MaKey MaKey

The MaKey MaKey, an "invention kit for everyone," should be a part of any tinkering classroom.



What is MaKey MaKey?

MaKey MaKey creates a simple alligator clip-based interface between the computer and everyday objects. It plugs into the USB port of any computer, even a Raspberry Pi, and turns household objects into a keyboard or joystick. Play a piano made of bananas, a drum solo on your best friend's head, a xylophone made of flowers, or a video game with a controller made of paper and Play-Doh. The MaKey MaKey simplifies the input so you don't have to worry about resistance or capacitance – just attach the alligator clips to any real world object.

The "neat phenomena" and playfulness of the MaKey MaKey makes it a great introduction to physical computing prior to tackling Raspberry Pi or Arduino.

Resources to Explore

- **MaKey MaKey website** – http://www.makeymakey.com/
- **Getting Started with MaKey Makey** – http://www.makeymakey.com/howto.php
- **Operation game using MaKey MaKey and Scratch** - see *Invent To Learn* pages 167-168
- **Digitally Interfaced Book: Paper, Graphite, Makey Makey, Scratch, and Imagination** – Instructions for completing an interactive book project http://digitalis.nwp.org/resource/4885

Getting Started with

MaKey MaKey & SCRATCH

Part I: Setting up Scratch

try it at
scratch.mit.edu

1. Build this program to play a sound when you press the space key

```
when [space ▾] key pressed
play sound [meow ▾]
```

2. Try some other keys and sounds

```
when [up arrow ▾] key pressed
play sound [Woohoo! ▾]
```

```
when [left arrow ▾] key pressed
play note (60 ▾) for (0.5) beats
```

```
when [right arrow ▾] key pressed
play drum (11 ▾) for (0.25) beats
```

3. Record your own sounds

click on the sounds tab

Click the mic to record a new sound

Press the record button

Try out some examples at

tinyurl.com/makeymakeyscratchmusic

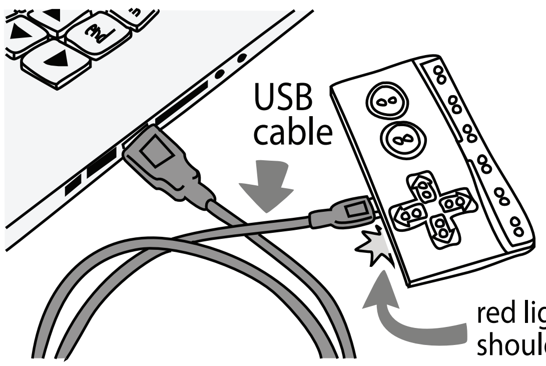
Getting Started with

MaKey MaKey & SCRATCH

Part II: Setting up MaKey MaKey

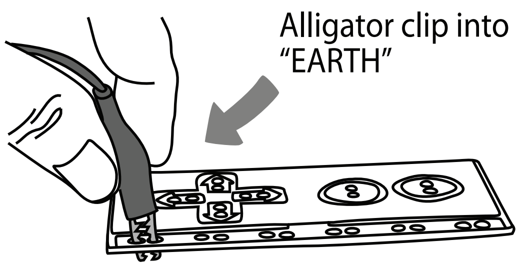
more info at
makeymakey.com

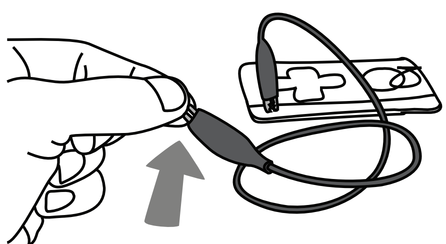
1. Plug board into computer

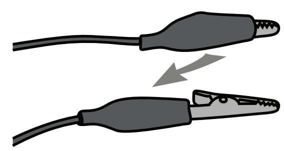


2. Close any pop-ups

3. Connect yourself to EARTH

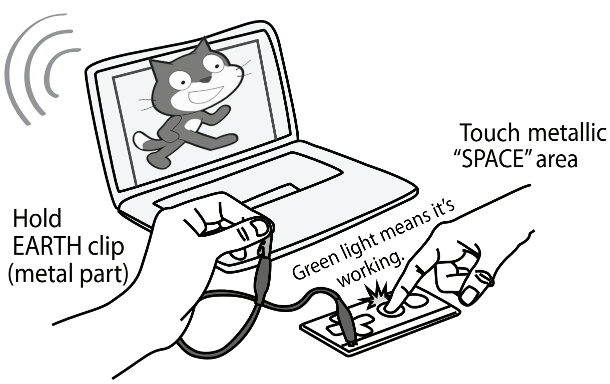






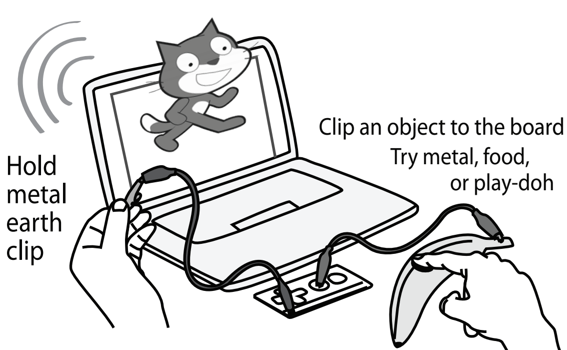
(Tip: Removing cover makes clip easier to hold)

4. Try it with Scratch



Your computer should think "space bar" was pressed.
If not, try pressing more firmly or licking your finger.

5. Connect everyday objects



Add craft materials and make a musical instrument out of **anything!**

Touch object to complete a circuit

Invent To Learn

Making, Tinkering, and Engineering in the Classroom

What is Wearable Computing?

Imagine making a sweatshirt with turn signals sewn into the back that flash while peddling your bike home from school, a backpack that detects intruders, or a T-shirt with LEDs that dance in a pattern that you've choreographed. What was science fiction a few years ago is now a classroom project. Known as wearable computing, e-textiles, or soft-circuits, these small computers can be incorporated into clothing or other projects where mobility and flexibility are desired.

Lilypad is a variation of Arduino, specially designed to be used in clothing and other textile objects. Designed by Leah Buechley at the MIT Media Lab, the board is machine washable and flexible. Sewing with conductive thread, rather than wire, creates circuits. Lilypad works with switches, sensors, lights, buzzers, battery holders, and other electronic components used with traditional Arduino microcontrollers. Smaller, flatter, sewable versions of these components are also available from the companies selling Lilypad. Some Lilypad boards do not have a built-in USB interface. In those cases, an FTDi connector is used to connect the Lilypad board to a USB cable from the computer.

Lilypad is programmed in the same way as Arduino, via the Arduino C IDE or alternative software like Modkit. Lilypad kits are a great way to get started with all of the bits and pieces you need.

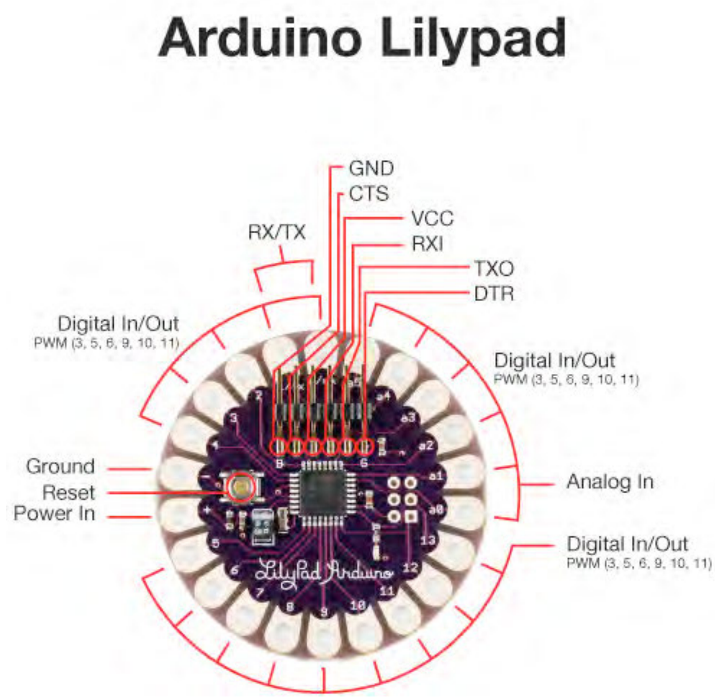


Diagram in *Invent To Learn*, page 124

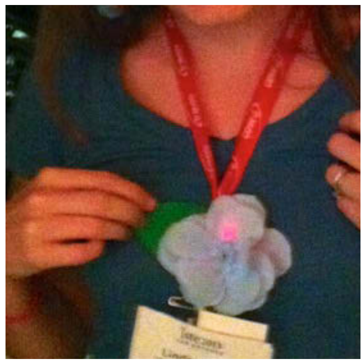
Participants at CMK12 Explore Lilypad Projects

Resources to Explore

- **Lilypad Arduino** – http://www.arduino.cc/en/Main/ArduinoBoardLilyPad
- **LilyPond** – Lilypad and other e-textile projects and workshop ideas from the inventors of the Lilypad. http://lilypond.media.mit.edu
- **Getting hands-on with soft circuits** – An e-textile workshop facilitators guide by Emily Lovell (e-book) http://web.media.mit.edu/~emme/guide.pdf
- **Soft Circuit Saturdays –** Tutorials and project ideas very suitable for kids from Angela, a maker in New England. http://softcircuitsaturdays.com/
- MODK.IT - A graphical block programming language for programming Lilypad, Arduino, and similar microcontrollers. http://modk.it

Sewing & Electronics Projects

- Wearables
- Stuffies
- Power Gloves
- Belts
- Headbands
- Finger Puppets
- Storyboards
- Ornaments



Components

Material: Felt, foam, fabric

Electrical connectors: Conductive thread, pipe cleaners, wire

LEDs - Light Emitting Diodes

Power: Coin cell batteries & sewable battery holders

Sewing supplies: Needles, thread, scissors, pliers, snaps, safety pins, velcro

3 tabs +

- no tabs

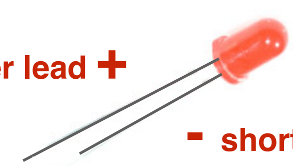
Sewable Battery Holder

top +

- bottom

Coin Cell Battery

longer lead +



- shorter lead

LED

Conductive Thread

LED SEWING TIPS

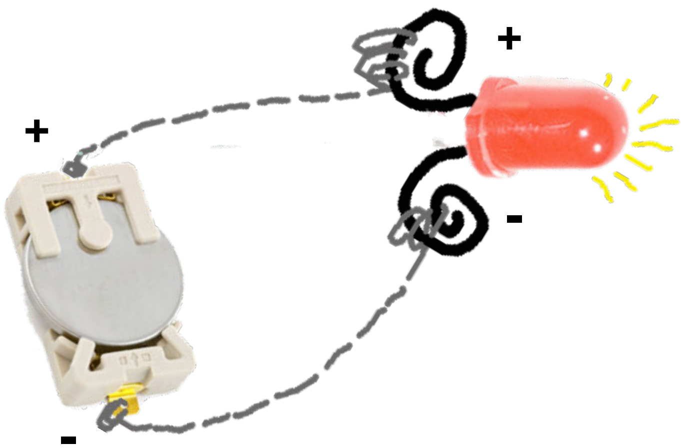
small (3mm) - medium (5mm) - large (10mm)

* You cannot shock yourself or burn out LEDs if you stick to coin cell batteries
* Prototype your circuit. Smaller LEDs use less power, bigger and flashers use more.
* Use the same color and size in one project
* Use a Sharpie to mark the positive (+) leads of LEDs and battery holder
* When the LED is in place, spiral the leads with pliers. This will hold it in place and give additional area to tightly connect the conducive thread to the lead.

component images from sparkfun.com

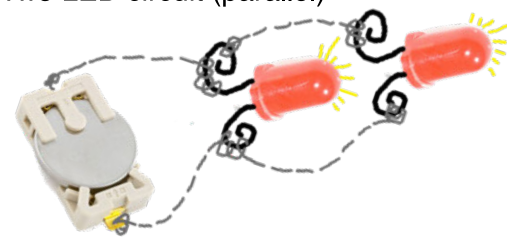
Simple LED Circuits

One LED circuit



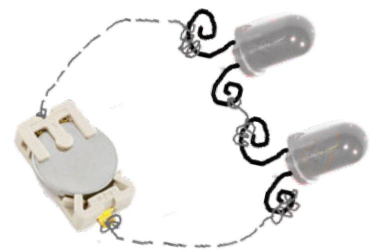
+ to + ALWAYS - to -

Two LED circuit (parallel)



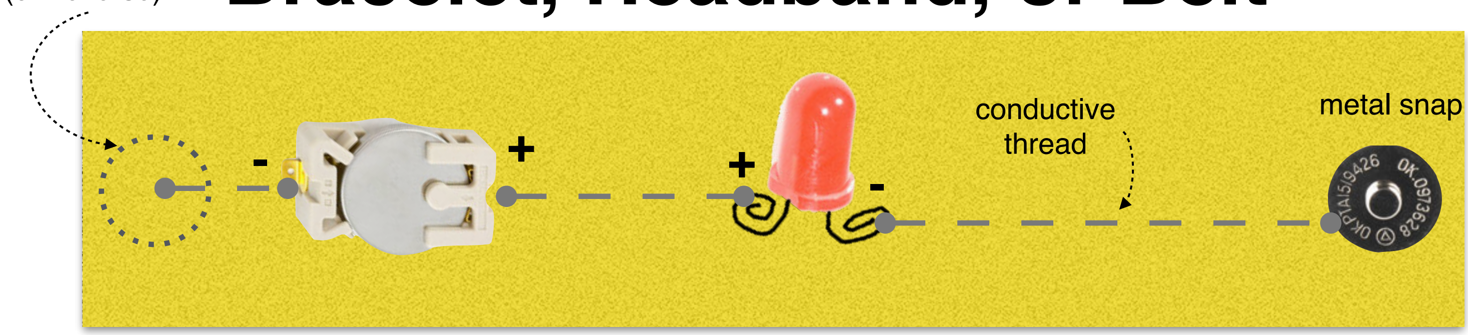
WILL WORK

Two LED circuit (series)



WON'T WORK
(with this size battery and LEDs)

Try It! - Wearable Bracelet, Headband, or Belt



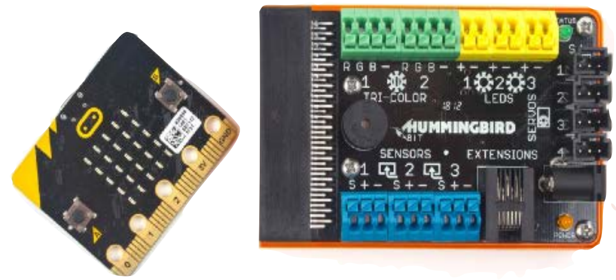
1. Cut a piece of felt that will wrap loosely around your wrist plus extra.
2. Sew a battery holder and LED onto felt with conductive thread. Connect the positive leads with conductive thread.
3. Use a metal snap at the ends of the bracelet as a switch. Make sure one snap is on the reverse side. Test to make sure it snaps correctly.
4. Connect negative leads to snaps with conductive thread.
5. Secure any loose parts with regular thread.
6. Test and decorate!

SEWING TIPS

* Use conductive thread only where necessary. Connect it tightly to components.
* Use regular thread for construction.
* Make sure conductive thread doesn't gap or "short" the circuit. The negative sides should never touch the positive sides.
* Spread the LED leads so they don't touch.
* Save your battery - a switch is anything that breaks (and reconnects) your circuit. Here the snap is a switch.

component images from sparkfun.com

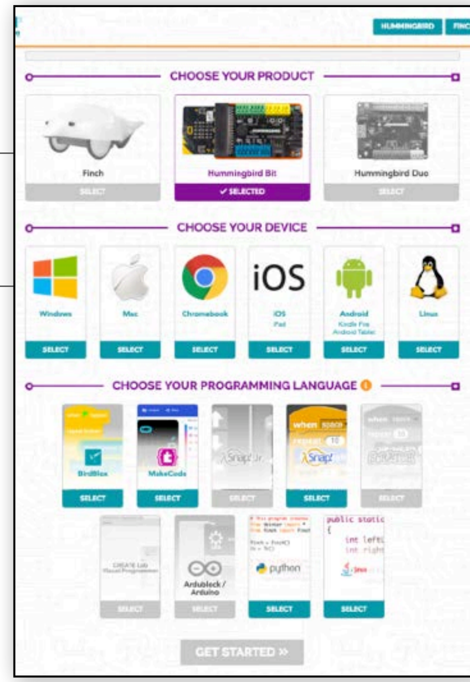
HummingBird Bit Robotics Kit Quickstart



The Hummingbird Kit contains everything you need to build super cool robots, complete with motors, lights, and sensors out of recycled materials. The brain of the Hummingbird is the popular new BBC micro:bit.The electronics just work when you plug components into the clearly labeled ports. There is no need to understand shields, resistors, or complex circuitry.

Best of all, Hummingbird robots can be programmed with Snap! (a dialect of Logo, similar to Scratch). Not only is this language designed for learning computer science and embraced by children, but the screen and outside world may now join forces.

For example, control your robot from the screen or build a game controller via the Hummingbird. No tricky libraries are required.The software you know and love just gains new blocks.



Step 1: The Birdbrain website

Birdbraintechnologies.com - click on **GET STARTED WITH OUR PRODUCTS**

Step 2: Select your configuration

The BirdBrain website walks you through the setup. Select the Hummingbird Bit, your device, and which programming language you want to use. Click **GET STARTED »**

We recommend you use **Snap!** via Bluetooth. Snap! is Scratch-like, has more programming options, and lets you use the screen in your program. Edits to Snap! blocks immediately start to work on the Hummingbird. Bluetooth lets you control your robots without a cable connected to the computer. A second choice would be MakeCode. MakeCode is also block-based, but has some limitations vs. Snap! and requires the code to be downloaded to the board every time you make an edit. Believe us, this is very annoying! *(Scratch will be available soon.)*

Step 3: Connect your Hummingbird Bit

Click the **Program** button for step-by-step instructions on connecting and programming your Hummingbird. Starting with this section makes sure that you complete essential steps that you will need later.

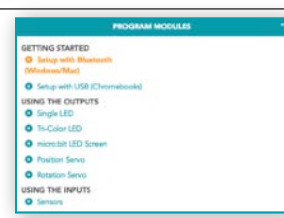
The Hummingbird connects to your computer over Bluetooth. Follow the on-screen installation instructions. You may need to adjust your security settings to allow the Connector app to run.

Step 4: Program, Build, or Both - Troubleshoot Along the Way

Read the User Guide!

The Hummingbird Robotics Kit allows you to create programmable actions and read sensors attached to a physical robot. Start with a simple action first. See if you can make an LED blink or change the micro:bit display. Add functions one at a time, testing as you go to make sure everything still works. Add sensors to trigger your robot actions.

As you build and program, test and troubleshoot along the way. Don't build everything at once and then try to figure out what's gone wrong.



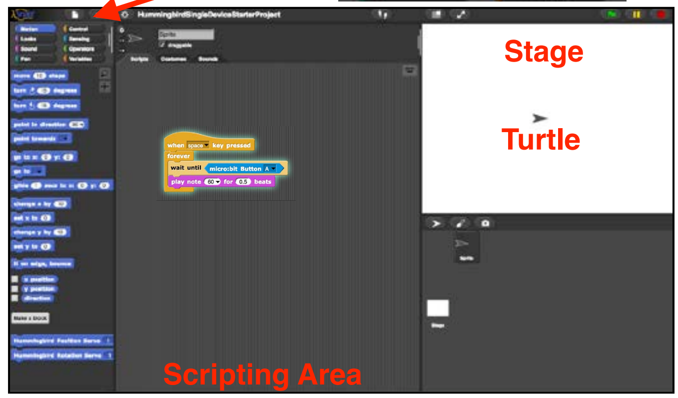
Click on the ***Program Modules*** dropdown to see directions for connecting LEDs, motors, and sensors.

Snap! Basics

Like Scratch, Snap! has palettes of different kinds of blocks. Click on the color coded palette buttons to see them.

- Blocks that say “Hummingbird” ONLY work with the Hummingbird.
- Blocks that say “micro:bit” ONLY work with the micro:bit
- Other blocks are used for programming or to control the actions of the on-screen Turtle.

Drag blocks out to the scripting area. Blocks will connect to form a stack that runs together. Use a control block to start your stack. The red light (top right) stops all scripts.



Connections - Inputs and Outputs

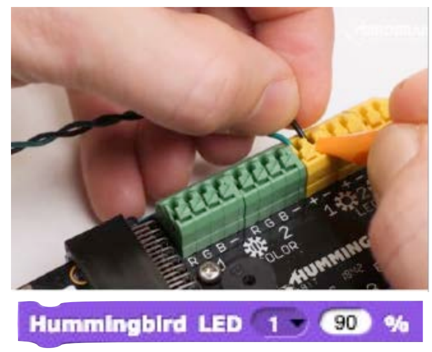
The Hummingbird board has easy-to-use connectors for the components that come in the kit. Use the orange terminal tool to press down on the connector tabs to insert wires into the correct holes. Each connector has a number printed in white on the board. The connector number is how you tell the code block which component you want to control.

Outputs

The program you write can send commands to the Hummingbird and the micro:bit to turn on LEDs, control the micro:bit display, make sounds, turn on/off motors, and more. There are two kinds of motors: **rotation** (turns until you tell it to stop) or **position** (turns to a fixed setting and then stops).

Inputs from Sensors

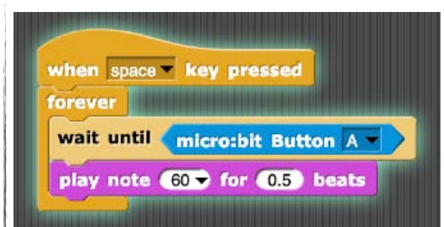
Your program can receive numbers from sensors on the micro:bit board or connected to the Hummingbird and use those values to make decisions. For example, the value of the micro:bit compass reading can be used as numerical input to a Snap! block. (Be sure to calibrate the compass.) The input from a light sensor connected to the Hummingbird is a number that represents darkness (a low number) or brightness (a high value).



This block tells the LED plugged into LED connector #1 to turn on at 90% brightness.

This block tells the motor (rotation servo) plugged into motor connector #1 to turn faster or slower as you turn the dial sensor plugged into sensor connector #3.

This stack tells the LED plugged into LED connector #1 to turn on only when the distance sensor detects an object closer than 100 cm. How would you change it to vary the brightness of the LED depending on the distance?



This stack plays a note when Button A on the micro:bit is pressed. If you don’t add the forever loop, the code will only run once. You can tell the code is running because it glows. Use a control block to start different stacks.

Troubleshooting

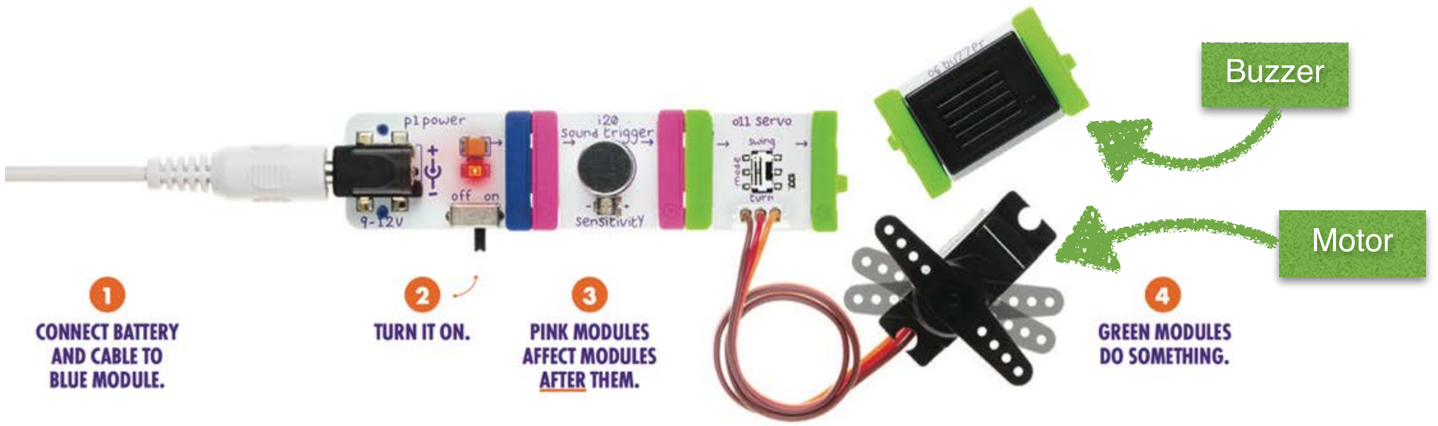
Check simple things first: is the Hummingbird powered (orange light is on)? Is the Connector app still running? Is the code block selecting the correct component in the same numbered connector? Is the component wired correctly and firmly?

Turtle to the Rescue!

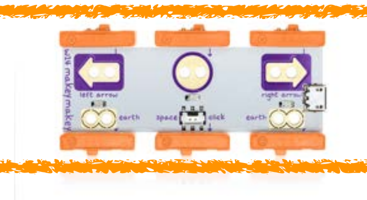
Use the “say” block to show sensor readings.

littleBits ELECTRONICS

littleBits electronic components are part toy, part electronic prototyping platform. They snap together magnetically and are color-coded for ease of use.



Orange modules offer logical functions, wires that extend distance, connections to computers and other devices (MIDI, wireless transmitters, MaKey MaKey, Arduino, and more).



Tips and Troubleshooting

- ☑ Always start with a blue power module. All littleBits need a power module, either the battery power module or a special USB power module. (They will not automatically get power form the USB port.)
- ☑ Make sure you have a flat surface to work on, otherwise the bits may come apart slightly and not work. Use the white mounting boards for added stability.
- ☑ Test complex combinations a section at a time.
- ☑ Try a new battery.
- ☑ Some bits have tiny switches for on/off or different modes. Some have tiny sensitivity adjustment screws.
- ☑ Order matters! Build circuits from left to right. Input modules only affect the modules that come "after" them (meaning to the right).

Project ideas

From the littleBits website littlebits.cc

- ✳ Logic Lesson littleBits.cc/browse-lessons/introduction-to-logic
- ✳ Investigating the Law of Reflection littleBits.cc/browse-lessons/investigating-the-law-of-reflection
- ✳ Creepy Crawly Cockroach littleBits.cc/projects/creepy-crawly-cockroach

Challenges

- ✳ Make a robot that makes art
- ✳ Make a nightlight
- ✳ Invent a security system

The littleBits site offers projects, resources and a 15% educator discount littlebits.cc

littleBits ELECTRONICS

Arduino Bit

littleBits solves two common problems encountered when using an Arduino microcontroller:

1. **To build Arduino-powered inventions, you need to understand electronics and sometimes delicate components.** Using the Arduino littleBit eliminates the complexity of prototyping using electronic components and breadboards. You can program the littleBits Arduino to control any of the littleBits components or read data from the littleBits sensors.

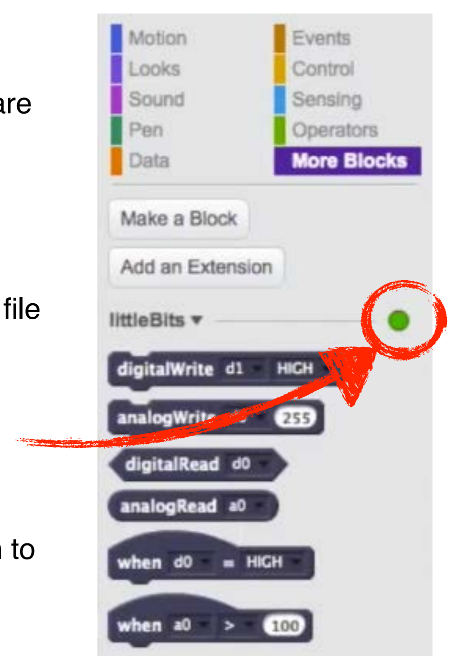
2. **The programming language (Arduino C) is difficult for beginner programmers.** You may use the Arduino littleBit with the standard Arduino IDE (Integrated Development Environment) and program it the same way as any other Arduino. However, to make programming simpler, there is currently a beta version that makes Scratch talk to the Arduino littleBit. Follow the directions below to try it out! *(Remember, this is beta, meaning it may be rough around the edges.)*

Arduino Bit Tips - Use the Arduino with standard Arduino C code (sketches)

- Go to littlebits.cc/bits/arduino and click on Forums link.
- This will take you to the “Getting Started with Arduino” page. **Be sure to follow all the directions.** You **must** install the IDE and get it talking to the Arduino or nothing will work.
- Find sample sketches (code) at littlebits.cc/arduino-sketches
- You may also use the files found under the Arduino IDE File:Examples menu.
- The links to download sketches are on the right side of these pages under “Additional Files”
- *The first “blink” sketch has a bug - you must change all the pin 13 references to pin 1, 5, or 9 depending on which output port you choose (the right side of the Arduino littleBit).*

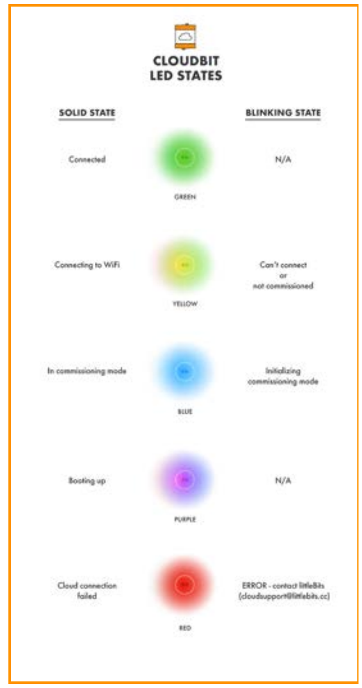
Arduino Scratch Setup Tips

- Open a browser window to Scratch (scratch.mit.edu) and click on **Create**
- Be sure the Arduino IDE is installed correctly, the correct board and port are selected, and you have successfully uploaded at least one sketch to the littleBits Arduino. (See steps above.)
- Now you are ready to connect Scratch to the Arduino littleBit. Navigate to bit.ly/littlebitsscratch and follow the steps.
- If your browser doesn’t download the littleBits Scratch extension (the .ino file in step 5), copy ALL the text on the page and paste it into a new sketch in your Arduino IDE. Save the sketch.
- When you see the green dot in Scratch panel “More blocks”, you can use the black littleBits blocks in any Scratch program, or use littleBits to control Scratch sprites.
- If the dot is yellow, try restarting the browser, turning the power off and on to the Arduino littleBit, restarting the Arduino IDE and reloading the Scratch extension sketch to the Arduino.



Cloud Bit

The littleBits cloudbit is a fast and easy way to create Internet-enabled inventions.



Start at littleBits.cc/cloudstart

1. Claim the cloudbit - create a login and give your cloudbit a name.
2. Follow the directions to setup and initialize the cloudbit. The cloud bit will create its own network - you need your computer to be able to select that network.
3. The setup process takes a few minutes; be patient with each step.
4. Follow the online tutorial with the simple button circuit and LED output.

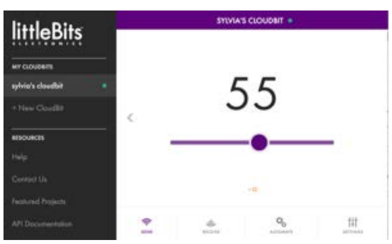
Next, change the cloudbit input from the button to a slider or switch. Test to see if the cloudbit browser dashboard is still working. When you change the circuit, it will take time to reset. Wait for the cloudbit LED to return to a steady green.

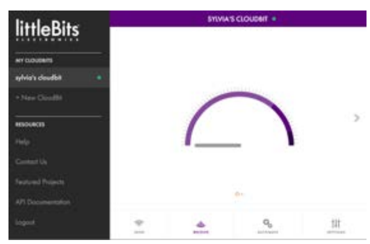
Classroom tips

- If your students don't have email addresses, setup the cloudbits ahead of time with generic email addresses. Store the cloudbits labeled with the email and password.
- Give the cloudbits distinct names, not "my cloudbit".

CloudBits Dashboard

When you log in to the Cloudbits account, you will see the Cloudbits Dashboard.





Send: Use a virtual button or slider to send values to the cloudbit. Any littleBits output module attached to the cloud bit will turn on, or respond to the slider.

Receive: Any littleBits input device, meaning buttons, sensors, dials, etc. attached to the left side of the cloudbit will send data to the dial and number display.

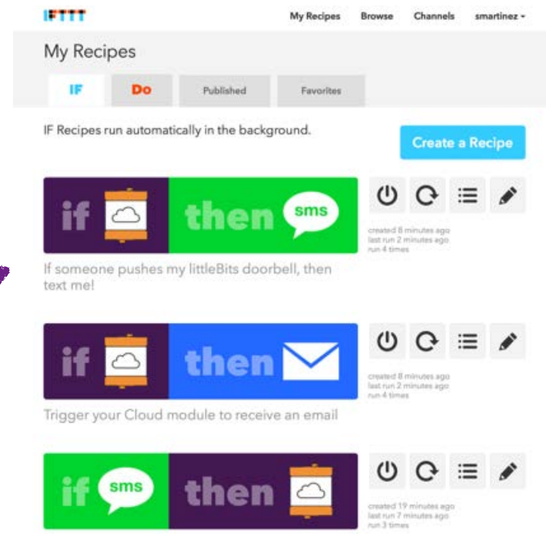
Automate: Automation requires an **If This Then That (IFTT)** account. IFTT is a website where you can create "recipes" that automate cloudbit inputs and outputs. See the next page for more information.

Tips

- Use the USB power LittleBit instead of the battery littleBit. Plug it into the wall outlet, not your computer.
- You can use your phone or tablet to set up and use the cloud bit (over wireless)
- MORE TIPS AND TRICKS at http://littlebits.cc/tips-tricks/tips-tricks-cloud-module

littleBits ELECTRONICS Cloud Bit

Automate your littleBits inventions through the Internet



If This Then That (IFTT.com) allows you to create recipes that connect services, like your email, text messages (SMS), Twitter, Google docs, and many others. You define a simple trigger (If this…) and decide what to do when that happens (…then that.) For example, the first recipe in this list represents:

"If the cloudbit gets an input, then send a text message."

You can name the recipe anything you like, "If someone pushes my littleBits doorbell, then text me!" However, IFTT doesn't really know whether you have put a button littleBit in the circuit or any other input. It's just waiting for the cloudbit to get ANY signal from ANY input (pink) littleBit.

Sign up for an IFTT account, then search for cloudbit recipes, or make your own.

IFTT & Multiple Cloudbit Tips

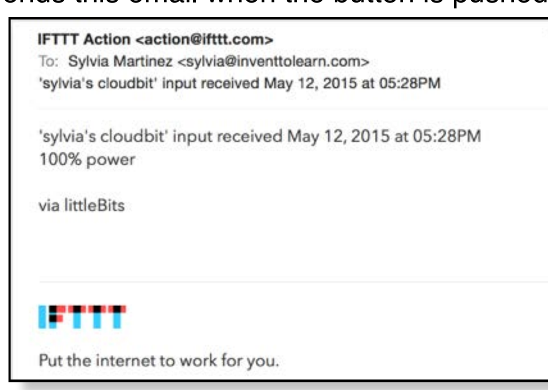
- The cloudbit and cloudbit Dashboard were not designed for multiple cloudbits and multiple computers. It does work if you are careful.
- Be sure to use the cloudbit Dashboard to access IFTT, this will force a reauthorization of the cloudbit when necessary.
- The cloudbit initialization process will break the connection to IFTT.
- If you reinitialize or change a cloudbit, edit the IFTT recipe to select the new cloudbit name, then click the update button to save the new recipe.
- If you are reinitializing a cloudbit, the setup process it may trigger IFTT recipes connected to the old cloudbit.
- The recipes in your IFTT account will remain active even when you put your bits away. Turn them off if there is a chance of them triggering from unintended inputs.

Example

This IFTT recipe…

Plus this littleBits circuit…

Sends this email when the button is pushed.



IFTTT Action <action@ifttt.com>
To: Sylvia Martinez <sylvia@inventtolearn.com>
'sylvia's cloudbit' input received May 12, 2015 at 05:28PM

'sylvia's cloudbit' input received May 12, 2015 at 05:28PM
100% power

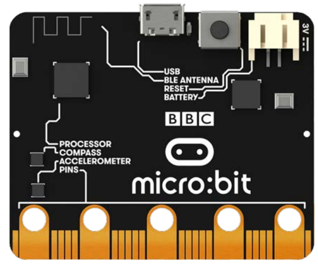
via littleBits

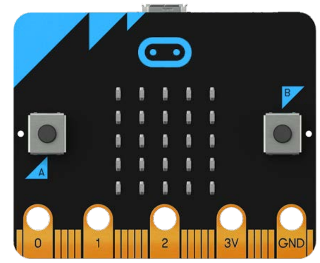
IFTTT

Put the internet to work for you.

BBC Micro:bit

The micro:bit is a credit card sized board containing a microcontroller and several built-in sensors and outputs. It is powered via USB connected to your computer or an external battery pack. Like an Arduino, additional sensors and peripherals can be attached to the "pins" at the bottom of the board.





Onboard sensors

- Light sensor
- Temperature sensor
- Accelerometer
- Magnetometer
- Compass
- 2 programmable buttons

Onboard display

- 25 LEDs arranged in a 5 by 5 array

Bluetooth

- Communicate wirelessly with your computer or other Bluetooth devices.

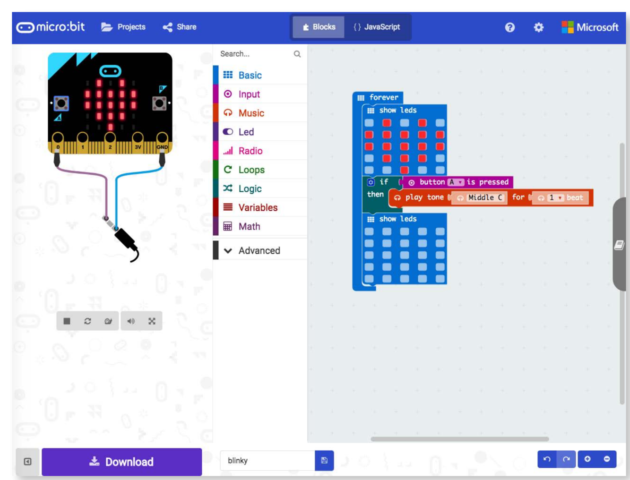
"Radio"

- Micro:bits can communicate with each other

Browser-based Block Programming at MakeCode.org

The micro:bit has several programming options, but the easiest one is directly in your browser.

- Go to MakeCode.org and click on the micro:bit. The "Getting Started" button (right under the Microsoft logo on the top right) will take you through a tutorial on the block language, and how to download programs to your micro:bit.



- *Code* – Drag code blocks from the palettes (mid screen) to the right and connect them to make programs.
- *Simulator* – On the left side of the screen, a live image of the micro:bit board simulates your code in real time. The simulator ONLY models the code, it does not communicate with the actual micro:bit.
- *Name your file* – Identify your work by naming it. Type a name into the box next to the download button at the lower left of your screen.
- *Download* – Connect your micro:bit to your computer via a USB cable and click the Download button at the bottom left of the screen. This will send a small file named "microbit-*yourfilename*.hex" to your computer.
- *Drag and drop* – The micro:bit looks to your computer like a memory card or external drive, allowing you to send your program to the micro:bit simply by dragging the file to the micro:bit "drive." Find your downloaded file (this is usually in your Download folder) and copy/paste or just drag it onto the external drive named MICROBIT.

Instant conversion to JavaScript
Use the toggle buttons at the top to switch back and forth. Changes made in Blocks will appear in JavaScript and vice versa.

Every time you change your program, remember to download and drag it to the micro:bit.

Scratch via Bluetooth is available. See scratch.mit.edu/microbit for instructions
Other programming options include Python and mobile apps. See microbit.org/code
More projects and resources inventtolearn.com/mb and makecode.microbit.org/projects

See our collection of resources at inventtolearn.com/resources-physical-computing for more online project ideas, micro:bit books and magazines, and micro:bit accessories.

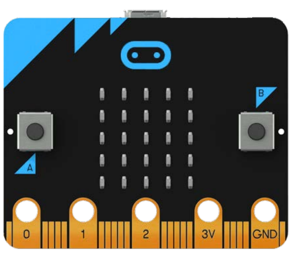
How to Connect to the Micro:bit…

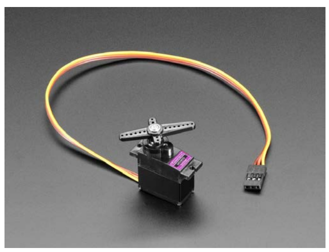
Connections use power – if you have things connected to the micro:bit, use the battery pack.

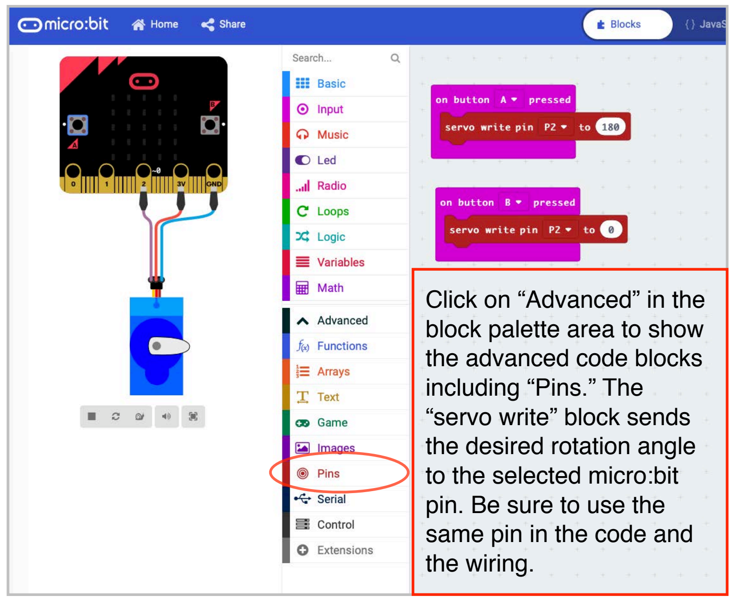
…to a Servo Motor

A servo motor turns to a specific position and stops. There are three connections from the micro:bit board to the servo motor: power, ground, and the control signal. Be sure to connect them correctly. The control signal can be connected to pins at the bottom of the micro:bit. Pins 0, 1, and 2 are large and useable with alligator clips. There are more pins in the edge connector that are accessible if you add a breakout board for the micro:bit. (Note: the colors of the servo wires on the MakeCode simulator may not be the same as your servo.)

Connect to..	Color Scheme 1	Color Scheme 2	Color Scheme 3
Ground (GND)	Black	Brown	Black
Power Supply (3V)	Red	Red	Red or Brown
Control Signal (Pin 0, 1, or 2)	White	Orange or Yellow	Yellow or White

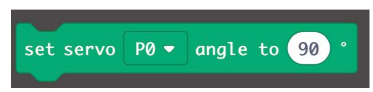






Click on “Advanced” in the block palette area to show the advanced code blocks including “Pins.” The “servo write” block sends the desired rotation angle to the selected micro:bit pin. Be sure to use the same pin in the code and the wiring.

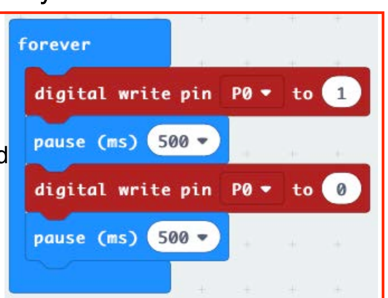
Servos can also be controlled by adding a “servo extension” palette. Click on “Extensions” at the bottom of the palette menu and select “Micro-servo library.” This will add specific servo blocks to MakeCode. It’s not any different than the “servo write” blocks under the Pins menu, but might make more sense if you are using a lot of servos in your projects.



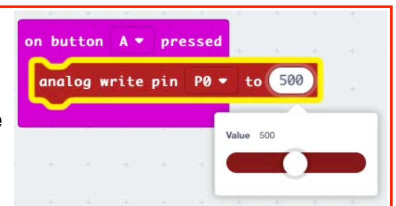
…to LEDs

An LED can be attached to the micro:bit and controlled by the code by connecting the + leg to a Pin and the - leg to GND. There are two ways to turn the LED on or off. (Note: use caution if you directly connect an LED to the micro:bit, it may overheat. We find that 10mm LEDs work the best if you don’t use a resistor.)

The ***digital write*** block sends ON (1) or OFF (0) signals to the LED attached to a Pin. The code to the right blinks the LED attached to Pin 0, with a 1/2 second pause between blinks.

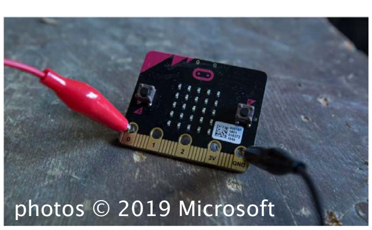


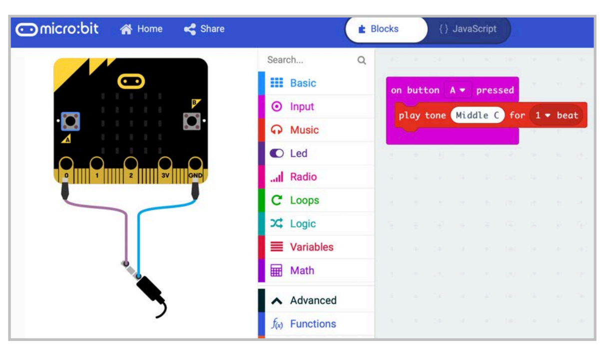
The ***analog write*** block sends a signal from 0 (off) to 1023 (full on) to the LED attached to a Pin. The code to the right will turn the the LED attached to Pin 0 to about half brightness when button A is pressed..



…to a speaker or headphones

The micro:bit will play tones if a speaker (or headphone jack) is connected to Pin 0 and GND.

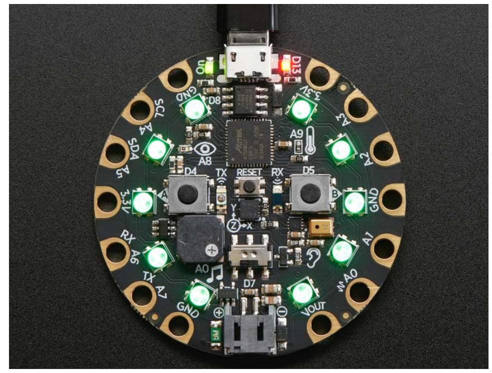


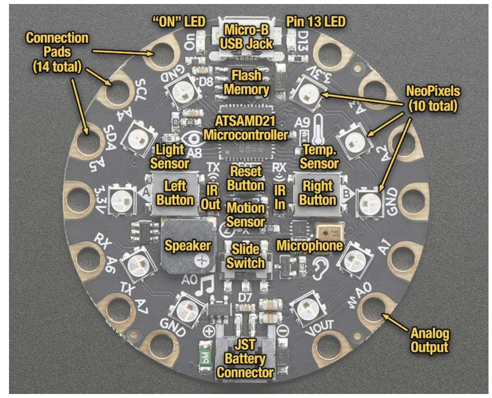


Circuit Playground Express

The Circuit Playground Express is a microcontroller board with additional sensors and buttons for quick prototyping of robotics, interactive circuits, and wearable electronics. It connects to your computer via USB, and can also run off a battery pack.

Connections to additional devices can be made with alligator clips, conductive thread, or soldering.





Onboard sensors

- Motion
- Temperature
- Light
- Sound

Inputs

- Capacitive touch pads
- 2 Buttons
- Slide switch

Onboard display

- 10 x mini NeoPixels, each one can display any color

IR communication

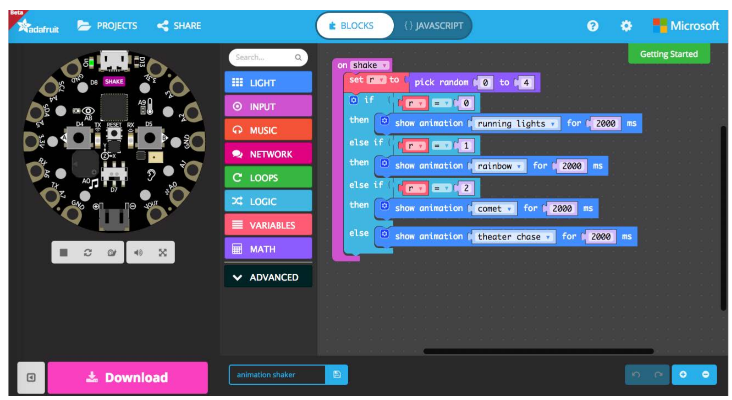
- Receive and transmit remote control codes
- Circuit Playground Express boards can send messages to each other

Sound

- Mini-speaker

Browser-based Block Programming at MakeCode.org

The CP Express has several programming options, but the easiest one is directly in your browser.



- Go to MakeCode.org and click on the Circuit Playground Express. The "Getting Started" button (right under the Microsoft logo on the top right) will take you through a tutorial on the block language, and how to download programs to your CP Express.
- Simulator – on the left side of the screen, a live image of the CP Express board simulates your code in real time.
- Name your file – Save your work by naming it. Type a name into the box next to the download button at the lower left of your screen.
- Download – When your code is working on the simulated CP Express, connect your CP Express to your computer via a USB cable and click the Download button at the bottom left of the screen. This will send a small file named "circuitplayground-filename.uf2" to your computer.
- Reset, drag, and drop – The CP Express looks to your computer like a memory card or external drive, allowing you to send your program to the CP Express simply by dragging the uf2 file to the CP Express "drive." Push the reset button on your CP Express. Find your downloaded file (this is usually in your Download folder) and copy or just drag it onto the external drive named CPLAYBOOT.
- Every time you change your program, remember to download, reset the CP Express, and drag the new file to the CP Express "drive." *This is a beta version. so be sure to read all the directions carefully as they may change.*

Instant conversion to JavaScript. Use the toggle buttons at the top to switch back and forth. Changes made in Blocks will appear in JavaScript and vice versa.

Other programming options include CircuitPython and the Arduino IDE. See learn.adafruit.com/adafruit-circuit-playground-express for complete instructions and example projects.

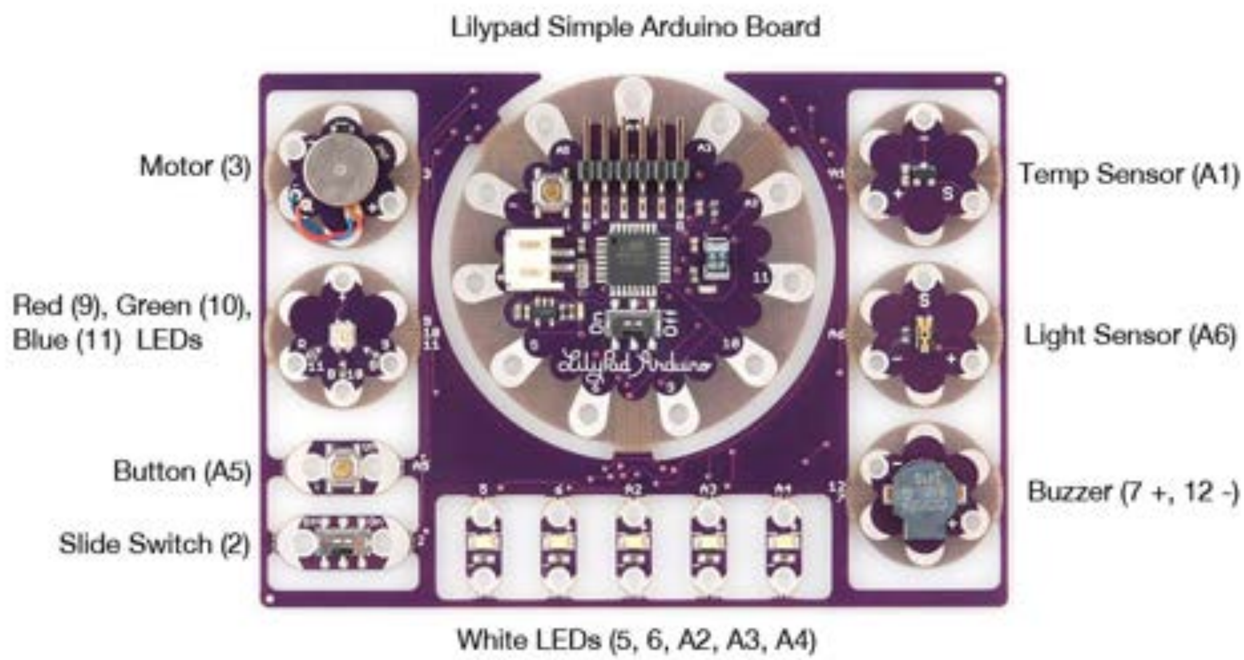
ProtoSnap - Lillypad Development Board Instructions

DO NOT BREAK APART!!

ProtoSnap = "Prototype" (now) + "Snap Apart" (later)

The ProtoSnap Lilypad Development Board provides an easy way to prototype (test) a Lilypad invention without a breadboard. A Lilypad, plus a useful set of peripherals, come out of the box on one plastic piece.

The sensors, buttons, LEDs, and more are ALREADY CONNECTED to the Lilypad - no additional wiring needed. This is the cool thing about the ProtoSnap board! If you get a project started with ProtoSnap and want to move on to the next step of sewing the components into fabric, see us for individual components.



Please DO prototype, please DON'T snap the parts off!
Thanks for understanding!

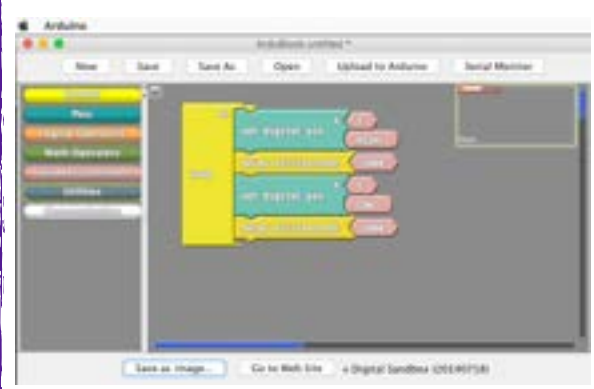
The Arduino pin numbers (in parenthesis) are used in the "sketch" (the program) to send or receive data to/from the sensors, LEDs. button, buzzer, or motor.

For more instructions on using and programming the Lilypad Protosnap see the Quickstart Guide on the Sparkfun website: https://www.sparkfun.com/tutorials/308

NOTE: You must follow the instructions to download the Arduino IDE and drivers in the tutorial for the Protosnap to work. There are sample code sketches on the workshop USB stick.

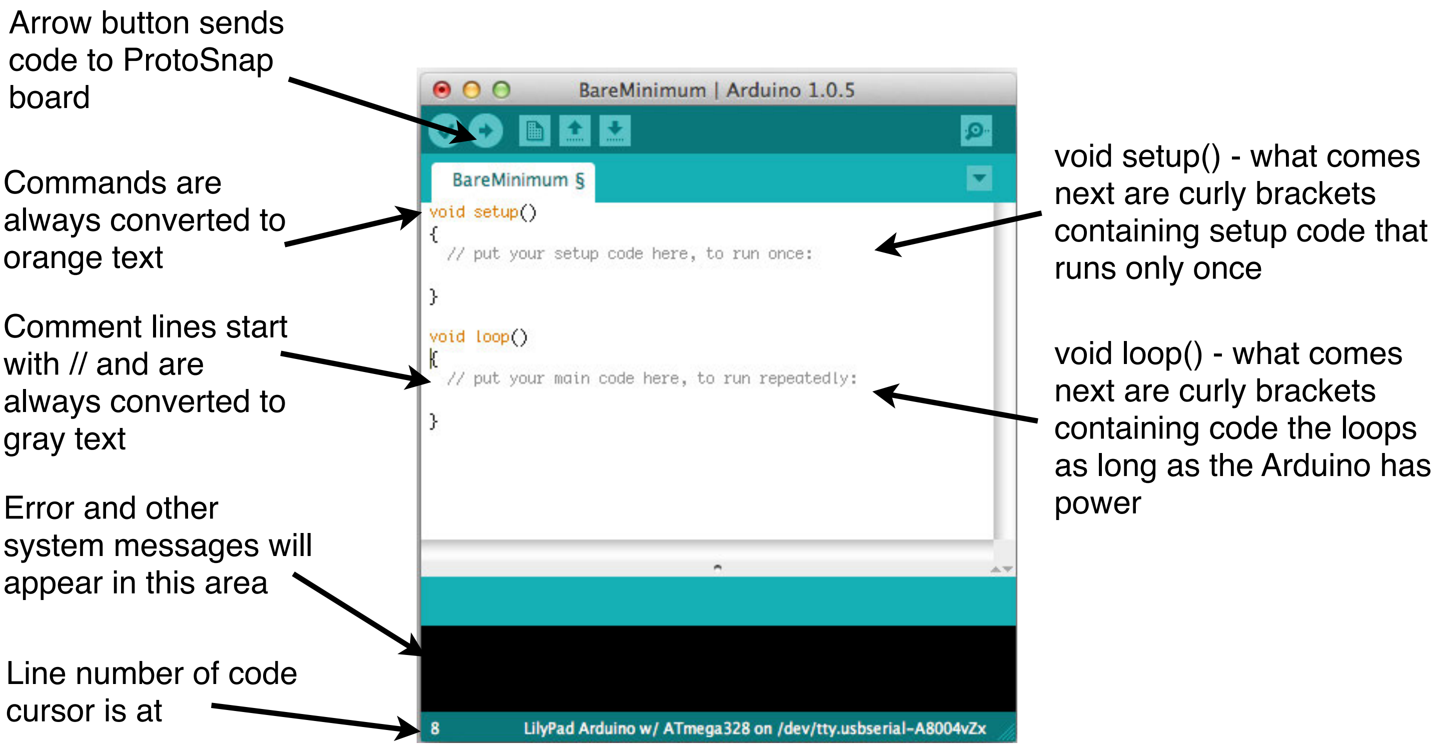
OR, try block programming!

Block programming with Ardublock
Snap together block programming - go to **http://bit.ly/martinez-ardublock.** Follow the directions for Mac or Windows.



Understanding the ProtoSnap LilyPad Example Code

All Arduino code starts with this basic template.



In the Arduino IDE, open the ProtoSnap LilyPad Example code file (lilypad_protosnap_all_device_example.ino). This can be found on the Invent To Learn workshop USB memory stick.

Click the arrow button in the IDE to send the code to the ProtoSnap Lilypad board. This code demonstrates all the capabilities of the ProtoSnap board. You can modify the code as you need for any project. When the code has been loaded to the ProtoSnap board, it will:

- Blink the white LEDs left to right
- Blink the RGB LED red, green, and blue
- Push the button and the buzzer will buzz (button is at the bottom left)
- Flip the switch and the vibrating motor will turn on (switch is at bottom left)
- White LEDs will turn on when the light sensor registers "dark"
- RGB LED will turn red when the temperature sensor registers "heat"
- Finally, the status of each sensor is streamed back to your computer (through the cable). To see what the values are, open up the serial monitor (the top right magnifying glass icon in the IDE) and make sure the baud rate is set to 9600 in the window that opens.

You can disconnect the USB cable and the code will still run (except for the serial monitor). The Lilypad battery must be plugged in and the on-board power switch must be set to on. This switch is ignored if the USB cable is plugged in.

The comments in the code will explain how all of this happens, and how to make small modifications to try out your own ideas!

Arduino/Lilypad Arduino ProtoSnap Setup Guide

You must follow these steps - no shortcuts!

1. Download the Arduino IDE software to your computer

Programs for the Arduino are written primarily using software called the Arduino Integrated Development Environment (IDE). Download this software to your computer from **http://arduino.cc/en/Main/Software** or ask the workshop leader for a USB drive.

Scroll down to find the heading ***Arduino IDE*** and select your operating system. The download should automatically begin. You may need to look in your DOWNLOADS folder to find the file. Double-click on the file to install the Arduino IDE. Do not download “beta” versions of the software or the source code.

2. Connect the Arduino or ProtoSnap board to your computer via a USB cable.

3. Install drivers

You may need to install drivers to allow the computer to send information out a port to the Arduino or Lilypad. It’s difficult to predict whether you actually need them or not, but it doesn’t hurt to install them just to be safe.

Follow directions for your operating system:

- Windows **http://arduino.cc/en/Guide/Windows**
- Mac **http://arduino.cc/en/Guide/MacOSX**

4. Open the Arduino IDE application

Arduino code files are called “sketches” and have the file extension .ino. Example sketches can be found under the File men of the IDEu. Protosnap example files are on the workshop USB stick. Additional Arduino Inventor Kit sketches are found at **http://sparkfun.com/sikcode**

5. Select your board

- Arduino Inventors Kit: Select (Menu) *Tools*: *Board*: *Arduino Uno*
- ProtoSnap: Select (Menu) *Tools*: *Board*: Lilypad *Arduino w/ ATmega328*

6. Select the serial port

Select (Menu) *Tools: Serial Port:*

- **Windows**: COM1 or COM2
- **Mac**: try the first listing with “usb” in the device name

7. Test the connection

Open the sketch “Blink” from the menu *File: Examples: 01.Basics: Blink.* Click the arrow button  in the top menu bar of the Arduino IDE. This button compiles the code and sends it to the Arduino. You should see two tiny red and green lights flash as the program is downloaded to the Arduino (if you look closely they are labeled TX and RX for transmit and receive). On the ProtoSnap, these lights are located on the small FTDI board that connects the USB cable and the Lilypad Arduino. The LED (pin 13) on the Arduino board should start blinking. Most Arduino boards have a built in LED to perform this troubleshooting function.

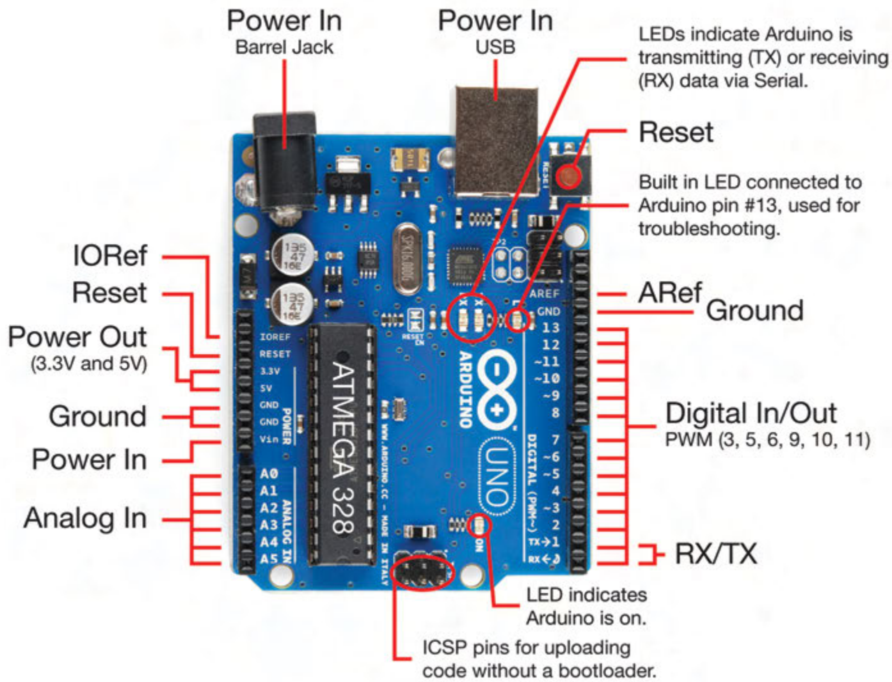
8. Try a simple change

The Arduino remembers the last program stored in it. Therefore, it may be running a program already uploaded. Try changing the delay speed of the blink sketch to make your LED flash differently and make sure you are in charge. Remember to re-upload your new sketch. Once you get all of this working, you’re home free and ready to invent!

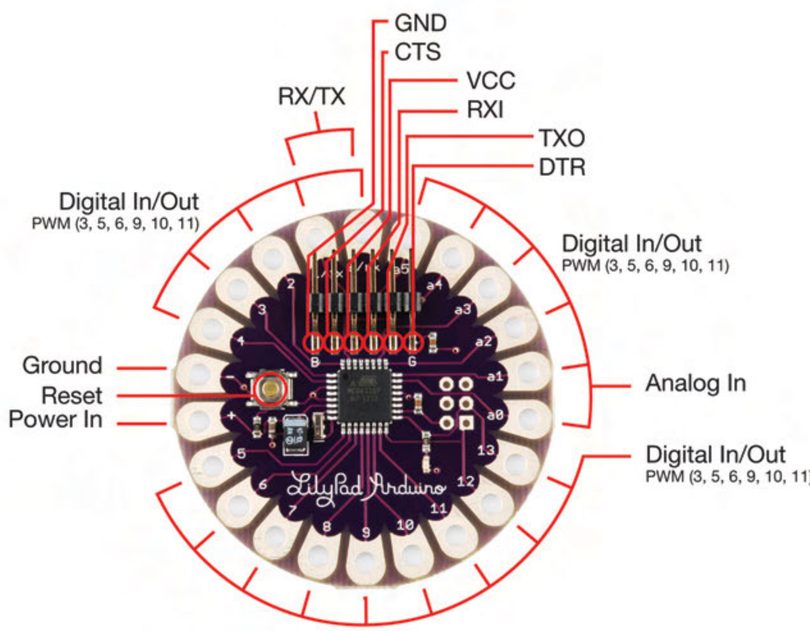
Troubleshooting: If you do not see the TX/RX lights blinking, it means the program was not downloaded to the Arduino. Check the IDE message area to see if there is an error message in orange. If a message says that the program cannot be uploaded, check the cable connections.

If the message says the serial port is not found, look at the list of serial ports, unplug the USB cable to the Arduino, then look at the list again. The missing port is the one to select when you plug the Arduino back in. You may need to try other USB ports.

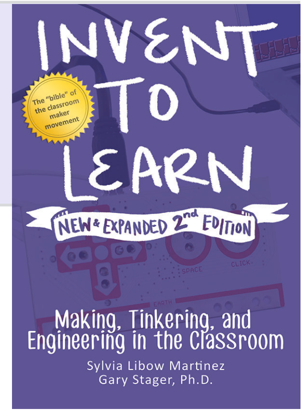
Arduino Uno



Arduino Lilypad



Books from CMK Press
www.cmkpress.com

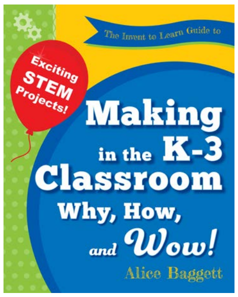


Invent To Learn: Making, Tinkering, and Engineering in the Classroom

(All New Second Edition)

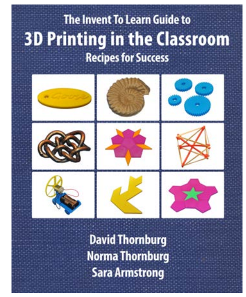
"The maker movement in schools now has a bible."
- Larry Magid, San Jose Mercury News

Invent to Learn Guides



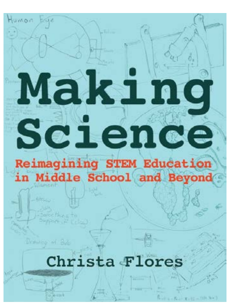
The Invent to Learn Guide to Making in the K-3 Classroom
Exciting STEM/STEAM project ideas from a veteran primary school teacher.

The Invent to Learn Guides to Fun & MORE Fun
Incredibly clever, whimsical, and fun project ideas for the classroom using Makey Makey, Scratch, Turtle Art, LEGO, and more!



The Invent to Learn Guide to 3D Printing in the Classroom
Practical project ideas for 3D printing across the curriculum.

The Art of Digital Fabrication: STEAM Projects for the Makerspace and Art Studio
Learning by making with the heart of a scientist and the critical eye of an artist.



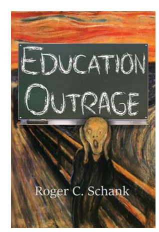
Making Science: Reimagining STEM Education in Middle School and Beyond
Anthropologist turned science teacher shares activities and lessons for authentic, inclusive science.

Sylvia's Super-Awesome Project Book: Super Simple Arduino
Kid maker and YouTube star shares simple and fun Arduino programming projects.

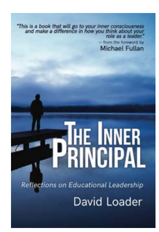
Meaningful Making (Vol. 1 & 2)
Columbia University's FabLearn Fellows, educators from around the world, share inspiration and projects for meaningful making in and out of school.

BUY NOW FROM amazon.com

Education Leadership



Education Outrage
A collection of can't-put-down essays by scientist, academic, entrepreneur, and artificial intelligence pioneer Roger Schank.



The Inner Principal: Reflections on Educational Leadership
Remarkably candid reflections by one of the most consequential school leaders of the past 50 years.

For special sales, out of US, and volume purchases - visit cmkpress.com

Invent the Future of Education Today

Imagine your school embracing learner-centered traditions, *supercharged by modern technology.*

Teacher Professional Development

The authors of ***Invent To Learn: Making, Tinkering, and Engineering in the Classroom*** are available to come to your school or district to speak or lead hands-on workshops that advocate for meaningful making, STEM/STEAM, and innovation in every classroom.

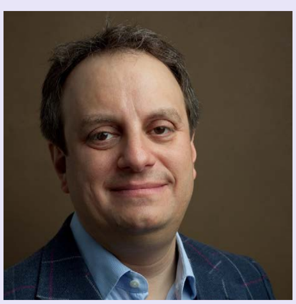
Inspiring Keynotes

Presentations should entertain, but also challenge the audience to think deeply and take action. Educators leave energized with a renewed sense of mission ***and*** practical ideas about what they can do immediately. We share an optimistic future where education can change children's lives, and educators are trusted to make this happen.

Student, Family, and Community Workshops

Presentations should entertain, but also challenge the audience to think deeply and take action.

Work with the Experts at Constructing Modern Knowledge



Gary Stager, PhD - A dynamic speaker and teacher with a global following, Gary has inspired a generation of educators to think outside the box. Creative, engaging learning is possible for all students.

Sylvia Martinez - Careers in aerospace engineering and video game design allow Sylvia to bring the real world into education. Create inclusive STEM/STEAM programs that combine the best of "maker" and PBL.

*Visit **CMKFutures.com** for more information*