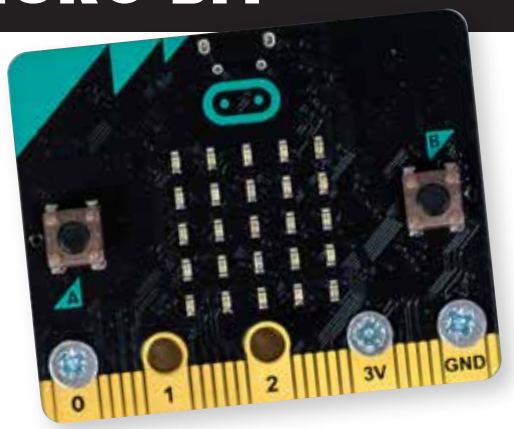
TOY TINKERING WITH MICRO:BIT



Make or hack your own toy, then program it to respond to your touch and movement! Micro:bit is an affordable, compact microcontroller that can interact with the Scratch programming environment. With it you can make an interactive puppet, game controller, or playful object that interfaces with the digital world.

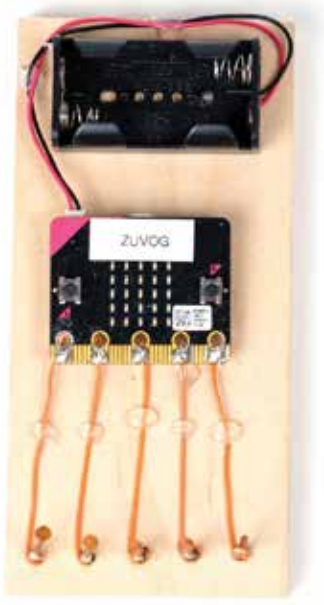
TRY IT!

Essential materials

Micro:bit has a variety of power options. The prototype board we build (on the right) is powered by two AA batteries and uses copper nails for easy attachment of alligator clips. The MI:power board is the most portable option and is powered by a 3V coin cell battery. The lithium polymer (or LiPo) battery is another common battery that can be recharged.

LiPo battery

MI:power board with
3V coin cell battery (our favorite)



AA batteries with pack

Computer with Google Chrome + and Scratch (www.scratch.mit.edu)
See https://scratch.mit.edu/microbit for all system requirements

the tinkering studio

© 2019 Exploratorium. www.exploratorium.edu
The Exploratorium grants reprint permission of this material for noncommercial, educational use only. Copyright notice must be included on all reprints. Requests for electronic or other uses should be directed to permissions@exploratorium.edu

exploratorium®

Collect a variety of objects to modify or hack

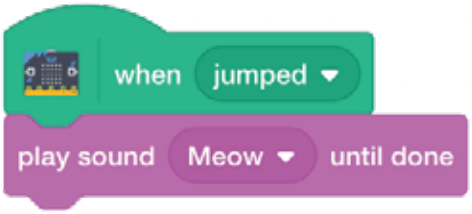
Tools and craft materials

GETTING STARTED

Connect a computer and micro:bit Go to https://scratch.mit.edu/microbit and follow the steps to set up a micro:bit and connecting a micro:bit to Scratch.

Choose a toy to hack or create The possibilities of toys to design or modify for this activity are limitless. We use pre-made toys as starting points and have materials readily available to make a custom toy out of it. Some starting points we use are wearables (headgear, wristlet), hand puppets, stuffed animals, and game controllers.

Program the micro:bit in Scratch Sample code can be very helpful to get someone started on their project. For example, we frequently use this code stack

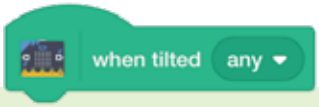


This is an immediate way to see how a physical action (tossing a micro:bit in the air) triggers a sound. Other possible commands are:

A micro:bit embedded in the stuffed animal is tested by tossing it in the air.

Test and explore Scratch provides a number of built in triggers for various actions. Play around with them to see how your creation responds to your motions. If the micro:bit is too sensitive and triggers too quickly, add a "wait" block to allow more time to elapse.

PROJECT EXAMPLES



Hand puppet A pre-made or homemade puppet "comes to life" with a micro:bit. When a micro:bit is placed inside a puppet's head, it will tilt back and forth with the mouth movement powered by a hand. The *when tilted* command can be utilized in Scratch to trigger animations and sounds based on this motion. The buttons on the micro:bit can also be utilized by pressing them when a hand is inside the puppet.

Hot potato A digital version of the pass-the-object game. Players arrange themselves in a circle and toss the potato, or any object, to each other while the music plays from the computer. When the music stops, the player holding the potato is out and the game continues until only one person remains. In this version, a micro:bit is placed inside a potato made of felt and cotton batting and is programmed with the game rules. A countdown timer controls the pace of the game and randomizes the amount of play time. The *when jumped* block tracks the tosses between players. If this block has not been triggered for a few seconds, seconds is removed from the play time to discourage running down the clock.

Tamagotchi A customized handheld digital pet made with code and everyday materials. The original Japanese version uses buttons to feed and take care of a digital pet on a screen. Here, we utilized the two buttons on the micro:bit to facilitate the interactions with a sprite in a Scratch animation. Code a digital dog that is fed by pressing one of the buttons on the micro:bit. A egg-shaped holder for the micro:bit gives the Tamagotchi an extra touch.

EDUCATOR ADDENDUM

A note on our philosophy:

The Tinkering Studio is based on a constructivist theory of learning, which asserts that knowledge is not simply transmitted from teacher to learner, but actively constructed by the mind of the learner. Constructionism suggests that learners are more likely to make new ideas while actively engaged in making an external artifact. The Tinkering Studio supports the construction of knowledge within the context of building personally meaningful artifacts. We design opportunities for people to "think with their hands" in order to construct meaning and understanding.

Environment

When setting up an environment for micro:bit explorations, we create three distinct areas for materials: the "toy store" with starter objects, an area for general craft supplies, and a hot glue gun station. We find that separating the materials from the main workstation keeps the table from being over crowded. Also, keeping the hot glue guns away from the work tables is important for safety.

We have laptops with micro:bits paired and ready to go on the work table. Since pairing micro:bits can take time, we like to have this ready ahead of time for participants.

When introducing the activity, it's helpful to have an example already set up. In the Tinkering Studio this is a computer running a program with a hacked toy already paired to it. We ask participants to interact with an example to get a feel for how the micro:bit behaves and how it controls the on-screen animation.

Facilitation

Before facilitating this activity for the first time, try syncing micro:bits with computers and writing a program in Scratch for a toy that you are augmenting. Messing around the technology ahead of time can help with on-the-fly troubleshooting problems that will likely arise. Also, taking the time to try the activity yourself can help you anticipate what moments might be challenging for learners.

This activity can be intimidating to jump into without some prior experience with Scratch or programming. It's important for the facilitators to know simple starting blocks of code to work with visitors on, and make suggestions for blocks based on the project that participants have in mind. It's also important for the learner to have an idea that they are excited about, especially when working through the challenges of a coding project. In this case, the role of the facilitator is to help them find a project idea they like, drawing on their personal interests, or interesting objects and materials to use.

When challenges inevitably arise, the facilitator can act as a brainstorming partner to talk through what is not working with the code. Sometimes the facilitator knows what can be changed, and other times they are also trying to figure out what's happening. Being co-learners is a way to collaboratively work through a challenging problem. This process may also bring about further understanding from the participant in how to structure a Scratch program and what the blocks mean.

RELATED TINKERING ACTIVITIES

Activity Connections

Try these related activities to develop your own repertoire of tinkering experiences.

Toy Take-Apart: Collect discarded mechanical stuffed toys and dissect them to find battery packs, switches, sensors, and motor-driven mechanical elements. You can test the things that you find inside, repair broken toys, or repurpose them using your imagination and a few tools to create new and original playthings.
https://tinkering.exploratorium.edu/toy-take-apart

Light Play: Use common materials in unusual ways to create kinetic light and shadow vignettes. Individual vignettes, carefully constructed using point source lights and slow moving motors, are eventually combined into one large light and shadow play wall. Incorporate a homemade switch into your vignette to control the motor speed and direction, or lights turning on and off in interesting ways.
http://tinkering.exploratorium.edu/light-play

ARTIST CONNECTIONS

Natalie Rusk

Natalie, a Research Specialist at the MIT Media Lab, focuses on the development of programs that build on young people's interests. She is a founder of the Computer Clubhouse program and a developer of the Scratch programming language. Her doctoral degree at Tufts University focused on bridging research on emotion, motivation, and informal learning environments. She continues to appreciate constructionist learning philosophy, finding that the most meaningful way to learn about ideas is by creating something, iteratively and collaboratively.

Asia Ward

Asia is a Minnesota based artist who creates public art projects about the environment, water systems, and electric power production. Since 2006, Asia has been working for the Science Museum of Minnesota and in 2012 started working for the KidWind Project developing products, writing curriculum, and training teachers about renewable energy education. Asia has participated in several Artist Residencies around the US and has received grants from the Minnesota State Arts board, the Jerome Foundation, Northern Lights, Public Art Saint Paul, and the Knight Foundation.

ACKNOWLEDGEMENTS

SCIENCE Simons Foundation SANDBOX This work was supported by a grant from Science Sandbox, an initiative of the Simons Foundation

The LEGO Foundation This project was made possible through the generous support from the LEGO Foundation