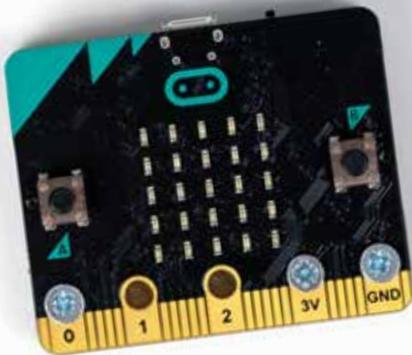


# PROJECT OVERVIEW

## SCRATCH + MICRO:BIT



In the past year, we explored a variety of tinkering activities using micro:bit. The microcontroller's compact size and programmability in Scratch made it an effective prototyping tool. Through our explorations, we played around with motion using servos, interactive animations created in Scratch, and custom controllers for games. We found toy creating and hacking to be a particularly rich activity and decided to go deeper with it. In this toy tinkering workshop, learners built or modified a toy and programmed its behavior in Scratch. We saw a variety of projects created with this framework and hosted a workshop at the Scratch conference at MIT in 2018.

## MATERIALS

### micro:bit

Micro:bit is an affordable and compact microcontroller that easily lends itself to tinkering. It packs a powerful punch with multiple features, but for this activity, we only used its two buttons and accelerometer.

### Power supplies

To run the micro:bit, we experimented with different types of batteries. Lithium polymer (LiPo) batteries are compact, AA battery holders allow for easy use of a common battery. We preferred using the *MI:power board*, an additional board that screws onto the micro:bit and houses a 3V coin cell battery. It's compact and wireless and didn't require using the micro:bit battery connector (which could easily break).

### Computer w/ Scratch 3.0

For our explorations, we used the newly-released Scratch 3.0 using Google Chrome. Chrome supports Web Bluetooth connections, which we used to connect our micro:bits along with an additional software called Scratch Link. For computers not running macOS 10.13+ or Windows 10+, we used an older device manager that supplied the connection for the micro:bit.

#### Device Manager Options

Version 1.0	<a href="https://khanning.github.io/scratch-gui/microbit-pro/">https://khanning.github.io/scratch-gui/microbit-pro/</a>	Uses Scratch Device Manager, found in the App Store
Version 2.0	<a href="https://scratch.mit.edu/microbit">https://scratch.mit.edu/microbit</a>	Uses Scratch Link, available for download from the website

the  
tinkering  
studio

## Tinkerable objects

The team created a wide variety of objects and mechanisms for the micro:bit to attach to based on our own interests around the activity. Our inspiration for displaying these materials for workshop participants were toy store display.

We categorized these pieces as:

- 1 Mechanisms: Things that crank, gyrate, tilt, spin, and wiggle
- 2 Toys: Plastic cars, stuffed animals, yo-yo's, snowglobes, Tamagotchi, Etch-A-Sketch
- 3 Puppets: Both handmade and repurposed

We also provided general craft supplies for participants to create what they would like to attach their micro:bit to, which included: cardboard, hot glue, felt, embroidery floss, scissors and other cutting implements, thin foam and colorful cardstock, string, velcro, and googly eyes.

## STARTING POINTS

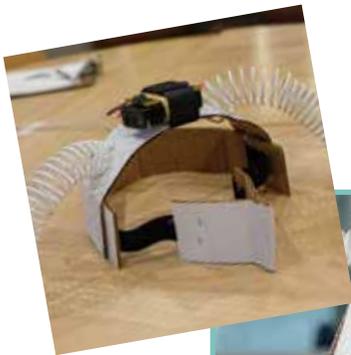
We curated starting points for visitors with micro:bits and Scratch. We categorized these different starting points as wearables, game controllers, and stuffed animals and sock puppets.

### Wearables

Some of our first micro:bit attachments were wearables. We wanted to make computational tinkering a full-body experience so we experimented with different ways to attach a micro:bit securely to a body. One core tenant of computational tinkering is that physical meets digital, and using your body to

make changes to your Scratch project is an excellent way to embody this.

We created two types of wearables: headgear and wristbands. They were made with cardboard and Velcro and were intuitive to use.



*Cardboard headgear*



*Velcro wristband*



## Game controllers



We explored other fully body experiences too, we built a tilt board that acted as a game controller. With a micro:bit mounted in the center, a sprite in Scratch could be moved left and right depending on the tilt of the board.



*A similar controller can be mounted on the head or other parts of the body.*

## Stuffed animals & sock puppets

We offered handmade and commercial puppets that had been hacked to use a micro:bit. Their movements were digitally represented on the screen and programmed to move and react to their physical counterpart.



These explorations often resulted in more coding since the connection between the physical puppet and its digital counterpart invited narrative explorations.

# FACILITATION

## Spark exploration

As a visitor to the Tinkering Studio, it may be difficult to discern the nature of the activity when only computers and micro:bits can be seen. This is why it's crucial that we have a quick project idea at a intro station.

Our favorite demo is a dinosaur plushie with a micro:bit attached. In Scratch, we animated a dinosaur on the stage and a few lines of code.



These two micro:bit commands are very consistent (they execute reliably) and are fun to play with. When visitors toss and shake the dinosaur plushie, they see the dinosaur on the screen jump and make noises in response. It tends to get an easy smile out of people of all ages.

Before visitors enter the space, we explained that the dinosaur can jump and make noise because of the micro:bit inside. We explained that they could create and program their own interactive toys.

## Sustain engagement

We strive to design problem spaces that support engagement with participants. One way we do this is through material selection and display. For micro:bit explorations, we invited participants to browse the materials and select what they would like to work with. We saw participants become deeply involved in constructing how to attach the micro:bit to their toy. Some start with making their toy, others will grab an existing hacked toy and start exploring the code.



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 in order to 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 enough to work 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 with the materials to use.

## Deepen understanding

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 a co-learner is a way 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.

## Scratch Conference

When we hosted a micro:bit and Scratch workshop at the 2018 Scratch conference, we offered an opportunity for participants to make their own toys

or objects to attach the micro:bit to. We saw many unique toys created, including a Magic 8 ball, an origami frog, and a car and racetrack.



The 90-minute session with adult educators afforded us to go deep with the activity in a way that's challenging with field trip groups visiting the Tinkering Studio. Scratch workshop participants created their own toys and code and worked in pairs. This sometimes resulted in participants choosing to focus entirely on the physical or digital aspect of the project. We heard some participants say that they "weren't good at making" or "weren't good at programming." While this organization may have been an effective use of time, it did not push at their preconceived notions of their own abilities to creatively explore technology. The workshop overall was incredibly positive and educators were amazed at how they could incorporate making into their computer programming classes.

# INSPIRING EXAMPLES

We see a great deal of possibility with micro:bit toys as an activity. Our team explored a variety of toy possibilities and ultimately created inspiring examples of micro:bit explorations.

## **Kaleidoscope**

A cardboard kaleidoscope with a micro:bit mounted on top. The when tilted command changed the direction of the colorful ball sprites.



## **Interactive Bunny**

A stuffed animal bunny was programmed to talk and respond to movement. It bounced, ate, and slept, and each command triggered a sound in Scratch.

## **Balance Board**

A wooden balance board with a micro:bit placed in the center. The board is used as a controller to interact with a Scratch animation using the tilt blocks.



# REFLECTIONS

## Projects that span the physical and digital world

We have explored activities connecting physical and digital worlds in the past, most recently with Digital Light Play. The task of tinkering in both worlds can be simultaneously stimulating and challenging. We saw participants tackle this task in a variety of ways: focusing on one realm exclusively, starting with one and bouncing back-and-forth, or building out one part entirely before moving onto the other.

We found the art of tinkering at the intersection of digital and physical incredibly rich and compelling. The variety and scope of projects completed by participants to be inspired and unique.

## Challenges in sharing physical coding projects

Sharing programming projects with the larger community is an important component of Scratch. For example, animations created in Scratch are shared and remixed within the community. However, we had an additional challenge when sharing our micro:bit projects with others because of the physical object component of the project. The screen animation requires the physical object to make a complete story, and the code alone shared on Scratch is not enough to get the full picture.

We shared via Twitter videos we took with the object and screen side-by-side. We are now exploring other ways of sharing projects that have both physical and digital components.

## Programming vs making

We found that not everyone wanted to do both programming and making. Many would focus on one aspect, or work in a partnership where roles were clearly defined. The project did pose an opportunity for folks to venture into a topic that they are potentially less familiar with through a building technique with which they are familiar. When we noticed this dynamic, we would encourage conversation between partners to engage them in what the other was doing. Sometimes this resulted in one person making contributions to what the other was working on.

## Servos and motion

One missing piece to this investigation is adding motion to our machines. Attaching servos directly to micro:bits proved to be unsuccessful, but we had success with adding expansion boards to micro:bits like the moto:bit board or using entirely different microcontroller. This fall, we had the opportunity to collaborate with Concord Consortium ([www.concord.org](http://www.concord.org)), a nonprofit educational research and development organization, on a project called Paper Mechatronics ([www.papermech.net](http://www.papermech.net)). We explored different design tools and techniques for mechanical movements combined with paper crafting and there are future possibilities to explore controllers like their LocoTap board (<https://flic.kr/p/2748MKP>) in a computational tinkering context.

# ACKNOWLEDGEMENTS



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



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