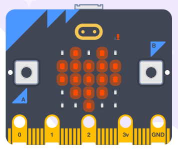
SCRATCH LESSON PLAN



Physical Computing with Scratch and micro:bit

The micro:bit is a tiny circuit board designed to help learners code and create with technology. It has many features including an LED display, buttons, and a motion sensor. You can connect it to Scratch and build creative projects that combine the magic of the digital and physical worlds.

Audience:

Classroom Teachers, Instructional Technology Specialists, Library Media Specialists, Informal Learning Environments

Time: Approx 1.5 hours total

- Part 1: Setup and Test - 15 min
- Part 2: Create a micro:bit Project - 30-60 min
- Part 3: Reflect and Share - 30 min

Objectives (Learners Will):

- Identify ways to connect the physical to the digital world
- Create projects that utilize the micro:bit blocks
- Remix and/or adapt projects that used keyboard keys, a mouse, etc., as inputs to use a micro:bit as the controller
- Evaluate problems/identify bugs and test solutions
- Reflect on the design process
- Communicate and share their projects with their learning community

See page 13 for aligned standards.

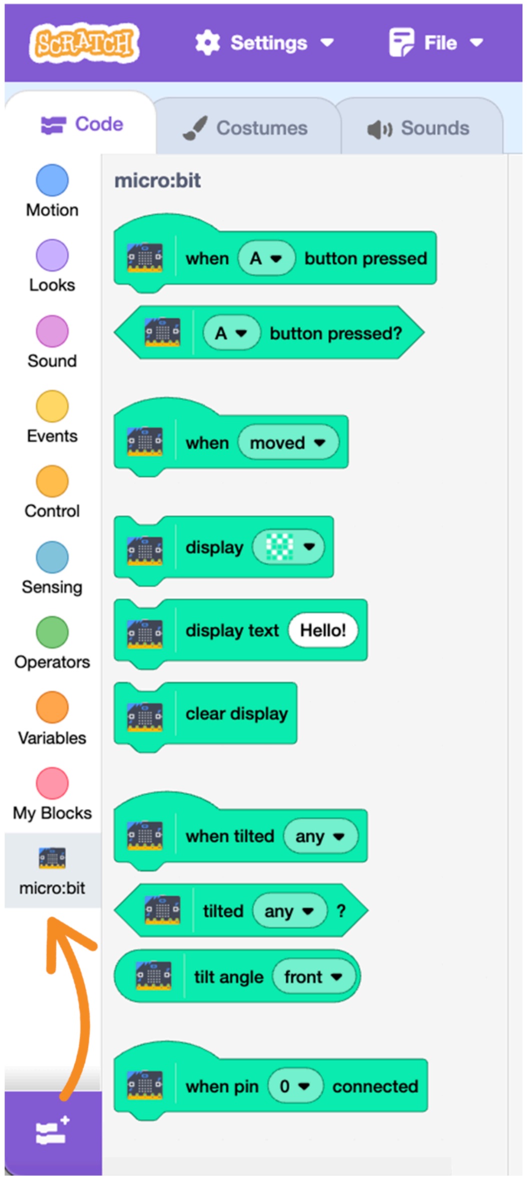
Resources for Learners:

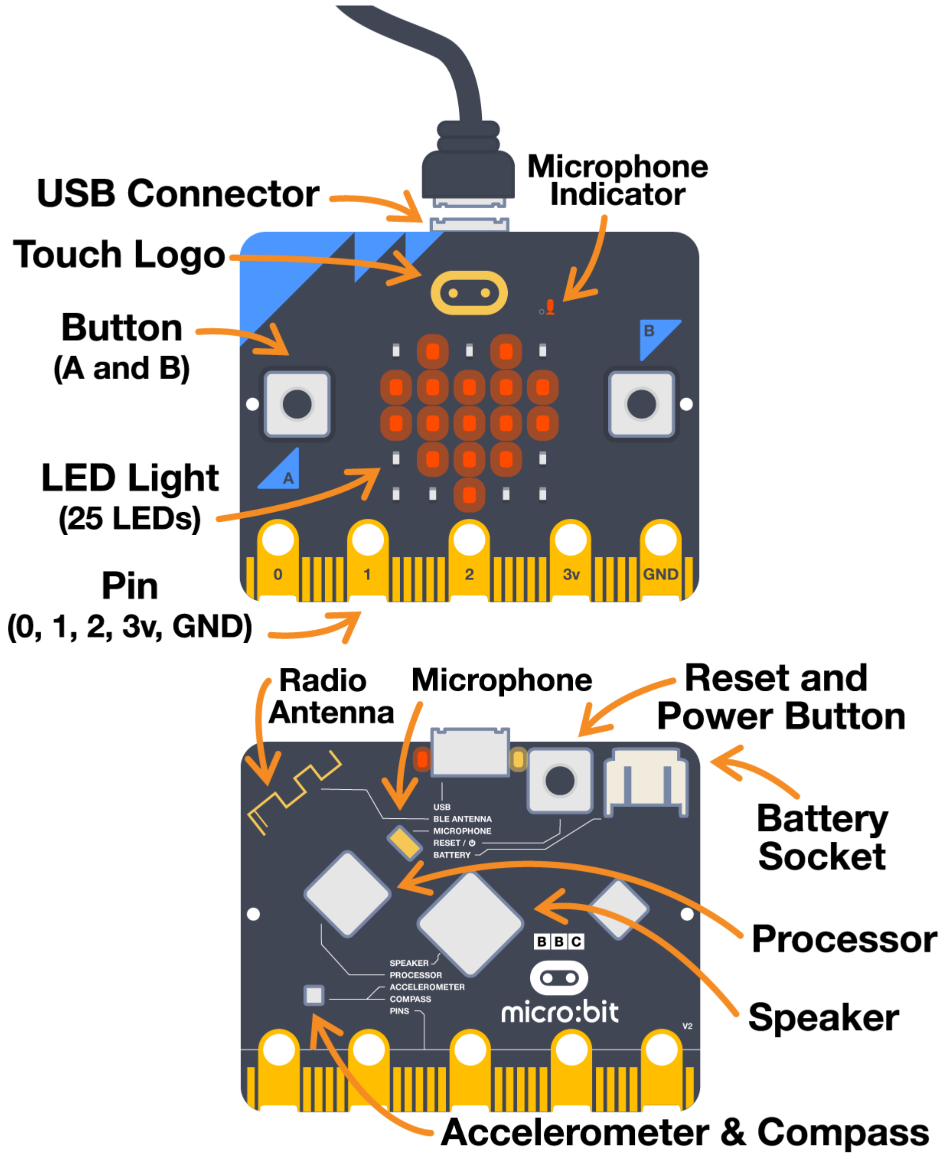
- micro:bit Coding Cards that align with this lesson, and the original micro:bit Cards (original) set (Student-Facing) - printable cards students can use to follow along with the lesson
- Scratch All Blocks Posters 18x24 including micro:bit hardware diagrams and blocks
- Extension Page (Website)
- Heart Beat (Example Project)
- Tilt Guitar (Example Project)
- Ocean Adventure (Example Project)
- Scratch Design Journal (Worksheet)
- Sprite Creation Cards (Student-Facing)
- Sound and Music Cards and Video Tutorial (Student-Facing)
- Conditional Statements Cards and Video Tutorial Part 1 and Part 2 (Student-Facing)
- Variables and Lists Cards and Video Tutorial Part 1 and Part 2 (Student-Facing)

Additional resources provided throughout the guide.

Where to Purchase the micro:bit (Website)

Get to Know the micro:bit Hardware and Blocks





Scratch works with v1 and v2 (shown) versions of the micro:bit. To learn more about different micro:bit versions and available features, see this page from the micro:bit site. And a poster version of the micro:bit hardware diagrams and blocks is available in the Scratch All Blocks Posters 18x24 set.

Part 1: Setup and Test

Setup (10 minutes)

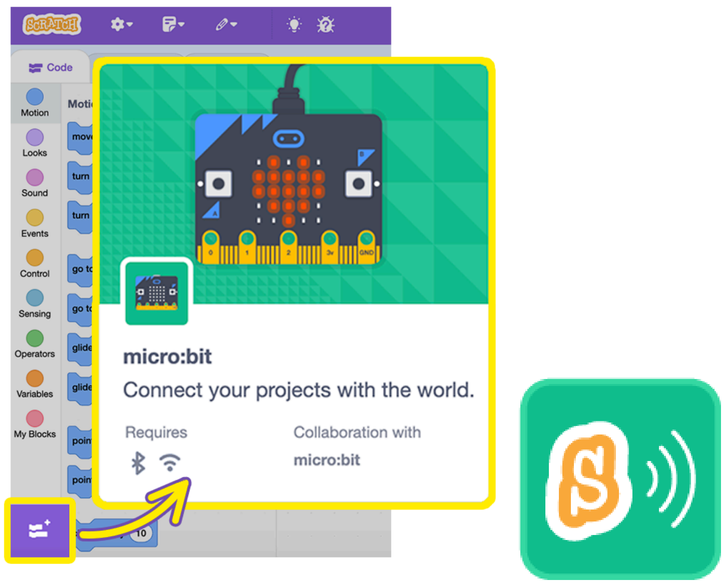
To add the micro:bit extension, click on the extension menu in the lower-left of the Scratch editor and choose "micro:bit."

Step 1: The first time you connect your micro:bit for use with Scratch, you'll need to download and install "Scratch Link," available on our extension page here.

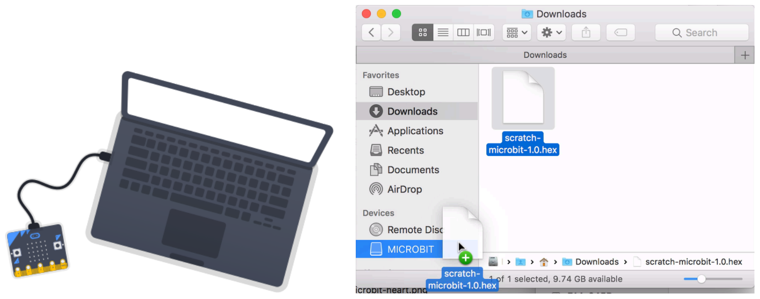
Each time you want to use the micro:bit with Scratch, you'll need to start Scratch Link from your Applications folder and make sure it is running. It should appear in your menu bar when running. *(Note: Scratch Link is compatible with most browsers, but there are minimum operating systems required. See our troubleshooting tips on our extension page here.)*

Step 2: **Connect a micro:bit to your computer** with a USB cable. *(Note: only one computer can be connected to a micro:bit at a time.)*

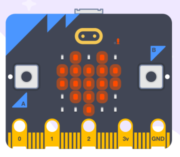
Step 3: When connecting your micro:bit for use with Scratch, you'll need to download (unzip, if zipped) and **drag and drop the Scratch HEX file (available on our extension page here) onto your micro:bit.** *(Note: When the file is successfully loaded, you will see the micro:bit's letter ID scroll across the LED display. You should not need to repeat this step unless you overwrite your micro:bit with another HEX file.)*



Scratch Extension Menu micro:bit Extension and Scratch Link



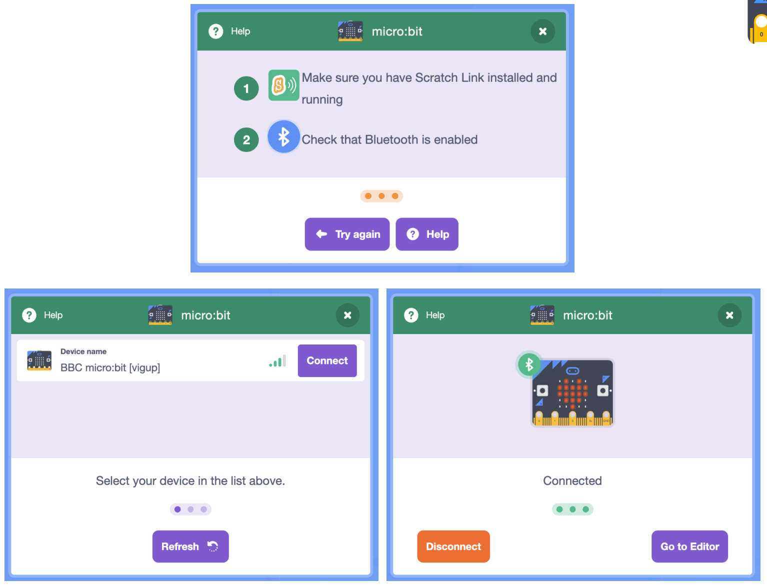
Drag and drop the HEX file onto your connected micro:bit.



Step 4: Ensure that Bluetooth is switched on on your computer.

Step 5: Head to the Scratch project editor to finish connecting your micro:bit. In the blocks palette under the micro:bit category, you should be prompted to connect to your micro:bit. An orange circle with an exclamation mark at the top of the category signals no connection. Click the orange circle to pull up the connection menu if it does not automatically appear and reconnect. A green check will appear in the blocks palette when the micro:bit is connected.

Each micro:bit has a unique five-character name assigned to it. If there are multiple micro:bits in the list of available devices, check the device name to find the correct one. Hint: when you drag the HEX file onto the micro:bit (aka "flash the micro:bit"), you will see the five-character name of the micro:bit scroll across the screen. Flash the micro:bit again if you missed the name the first time. You may want to add a sticker label with the ID name directly onto the micro:bit for future reference.



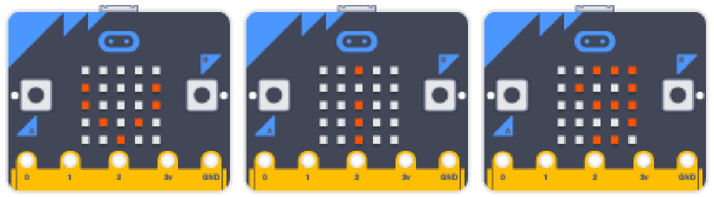
Make sure you have Scratch Link installed and running

Check that Bluetooth is enabled

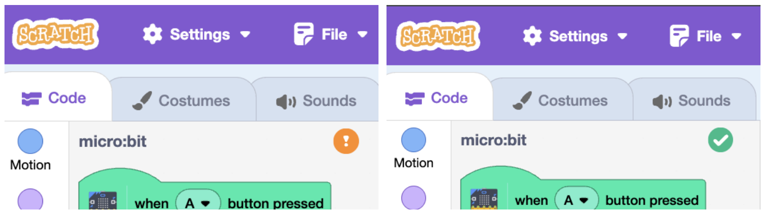
Select your device in the list above.

Connected

Steps to connect your micro:bit.



Flash the micro:bit (add the HEX file) to see the unique ID.



An orange circle with an exclamation mark means no connection. A green circle with a check means a connection is established.

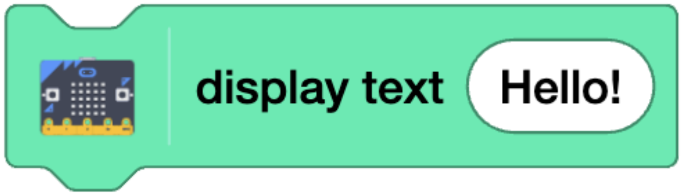
Resources:

- micro:bit Coding Cards (Student-Facing)
- Extension Page (Website)
- micro:bit Support article (Website)

Test the Connection (15 minutes)

Find the “display hello” block, drag it to the script area, and click on it. You should see “H E L L O” scroll across the micro:bit LED display.

Now, you can test out other blocks in the micro:bit category to see what they do. Pair them with Scratch blocks from other categories to see how you might use the micro:bit as a controller for action on the stage.



Scratch micro:bit block to try: display text.

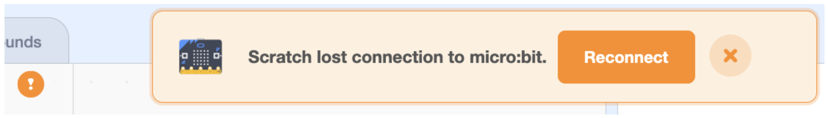
Try Battery Power (5 minutes)

Unplug your micro:bit from the computer and plug it into the provided battery pack.

You will be prompted to reconnect to your micro:bit.

Once connected, try running your script again.

Now, you can use your micro:bit as an unplugged controller in games, animations, informational projects, and more!


Scratch lost connection to micro:bit.

Reconnect notice.

Part 2: Create a mico:bit Project

Option 1: Try a Coding Card Project

See our original micro:bit cards for simple project ideas like:

- Squeak
- Move Around
- Press a Button
- Jump
- Move Back and Forth
- Create an Emoji

Option 2: Heart Beat

Try using the buttons on the micro:bit to animate a heart, or any other sprite you choose.

Step 1: Select one of the heart sprites from the Sprite Library, or create your own.

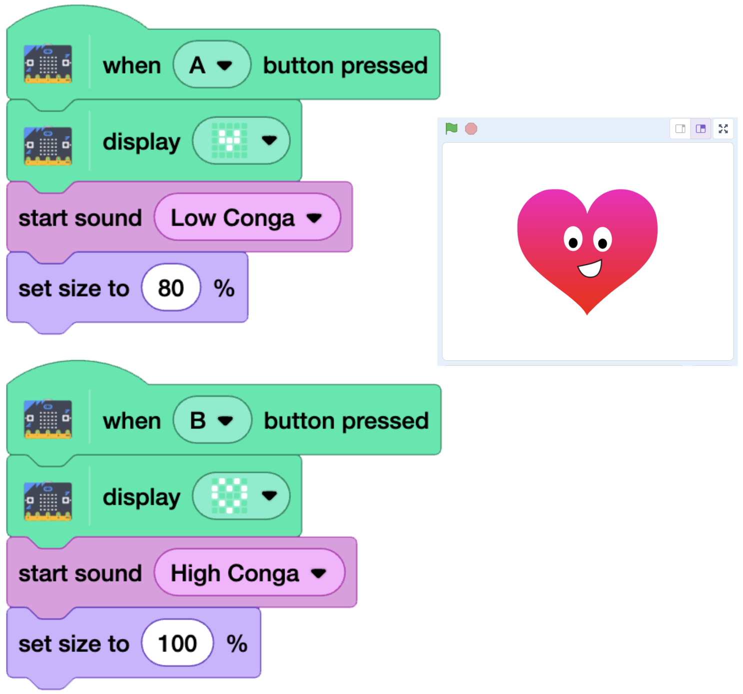
Step 2: Drag a “when _ button pressed” block from the micro:bit category onto the script area. Select the button (A, B, or any) that will trigger the script.

Step 3: Choose blocks from other categories (like Motion, Looks, and Sound) to create an animation that the button(s) will trigger.

Step 4: What else can you add to customize your project?

Resources:

- Heart Beat (Example Project)
- micro:bit Coding Cards (Student-Facing)
- Sprite Creation Cards (Student-Facing)



```
when A button pressed
display
start sound Low Conga
set size to 80 %

when B button pressed
display
start sound High Conga
set size to 100 %
```

Example “Heart Beat” scripts.

Option 3: Tilt Guitar

Try making music by tilting your micro:bit.

Step 1: Select one of the guitar sprites (or another instrument) from the Sprite Library or create your own.

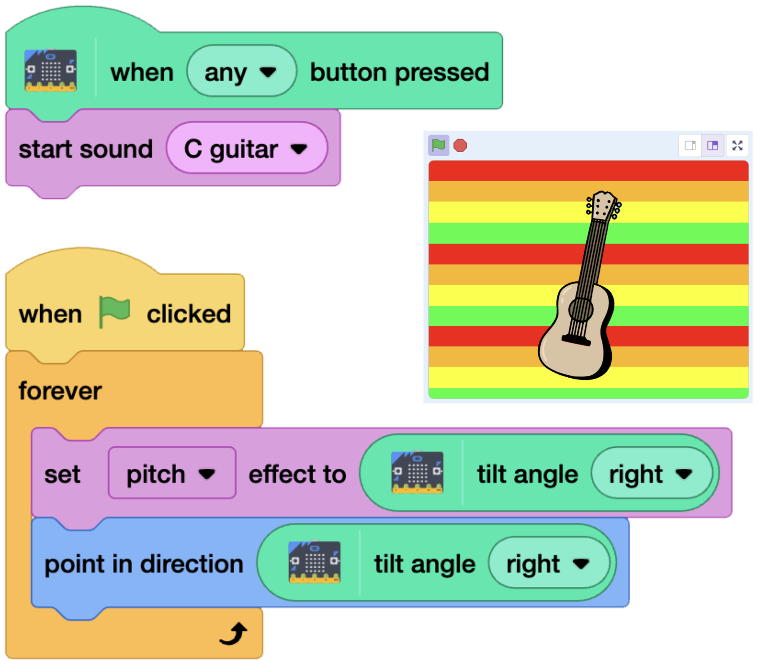
Step 2: Drag a “when _ button pressed” block from the micro:bit category onto the script area. Select the button (A, B, or any) that will trigger a sound to play.

Step 3: Add a “start sound” or “play sound until done” block. Choose a sound to play from the Sound Library that matches your instrument.

Step 4: To animate your instrument, you can add the “tilt angle” reporter block inside a “point in direction” block. Select a tilt angle (front, back, left, or right). Place this block inside a forever loop to have the program constantly check for the micro:bit’s tilt angle and adjust the sprite’s direction. Try different tile angles and experiment to see the differences.

Step 5: What else can you add to customize your project? Try out other blocks like “next costume” or “set pitch effect to (tilt angle)” to add additional effects.

Add a backdrop and animate it with the buttons.



```
when [any ▾] button pressed
start sound [C guitar ▾]

when ⚑ clicked
forever
    set [pitch ▾] effect to (tilt angle [right ▾])
    point in direction (tilt angle [right ▾])
```

Example “Tilt Guitar” scripts.

Resources:

- Tilt Guitar (Example Project)
- micro:bit Coding Cards (Student-Facing)
- Sprite Creation Cards (Student-Facing)

Option 4: Ocean Adventure

Build a game that uses the micro:bit as a controller.

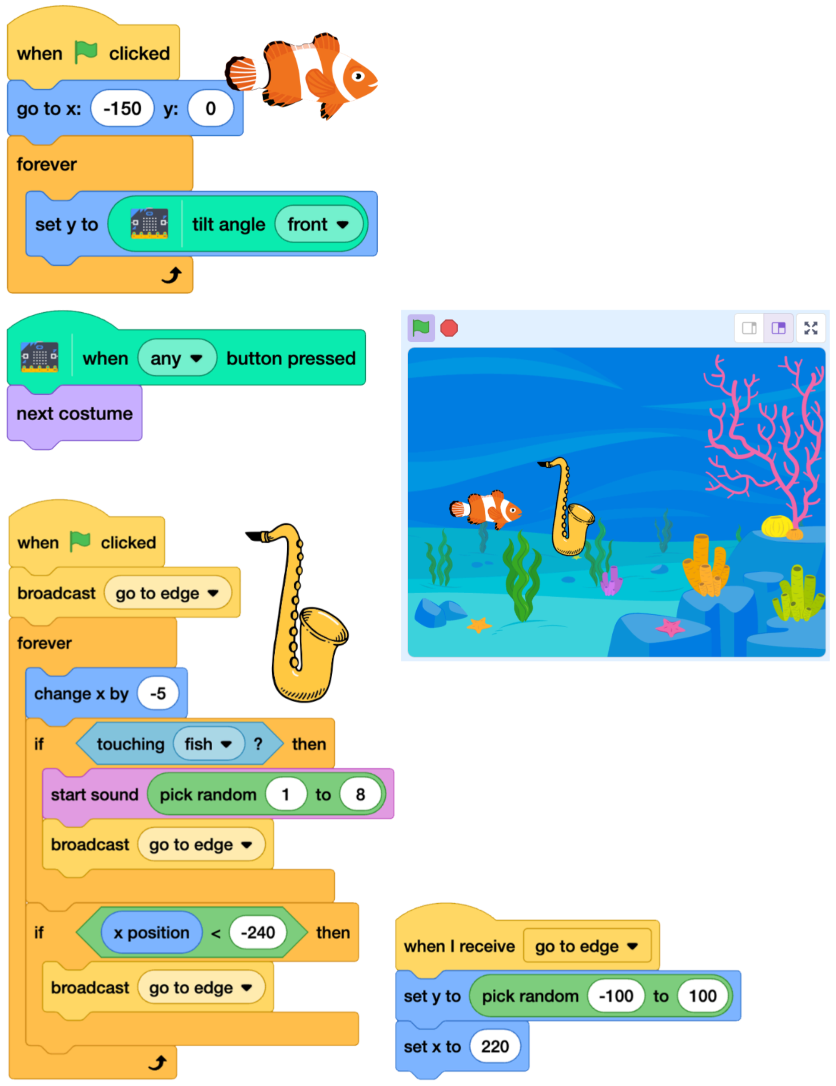
Step 1: Select two sprites from the Sprite Library or create your own. You can also add a backdrop.

Step 2: Add the “tilt angle” reporter block inside a “set y to” or “set x to” block. Select a tilt angle (front, back, left, or right). Place this block inside a forever loop to have the program constantly check for the micro:bit’s tilt angle and move the sprite by the amount of the tilt.

Step 3: Program the second sprite to constantly move across or down the stage. Use a conditional statement to make something happen if they touch.

Step 4: Use your micro:bit to move the first sprite and collide!

Step 5: What else can you add to your project? Try switching a sprite’s costume with a button press. Play a sound or use a “change effect” block when the sprites touch. Add a positive or negative score for each collision.



```
when [flag] clicked
go to x: -150 y: 0
forever
    set y to (tilt angle front)

when any button pressed
next costume

when [flag] clicked
broadcast go to edge
forever
    change x by -5
    if touching fish ? then
        start sound pick random 1 to 8
        broadcast go to edge
    if x position < -240 then
        broadcast go to edge

when I receive go to edge
set y to pick random -100 to 100
set x to 220
```

Example “Ocean Adventure” scripts.

Resources:

- Ocean Adventure (Example Project)
- micro:bit Coding Cards (Student-Facing)
- Sprite Creation Cards (Student-Facing)
- Conditional Statements Part 1 and Part 2 (Video Tutorial)
- Conditional Statements Written Guide and Coding Cards
- Variables and Lists Part 1 and Part 2 (Video Tutorial)
- Variables and Lists Written Guide and Coding Cards

Option 5: Day and Night

Close the circuit of the micro:bit pins to trigger an animation.

Step 1: Select a backdrop from the Backdrop Library, or create your own.

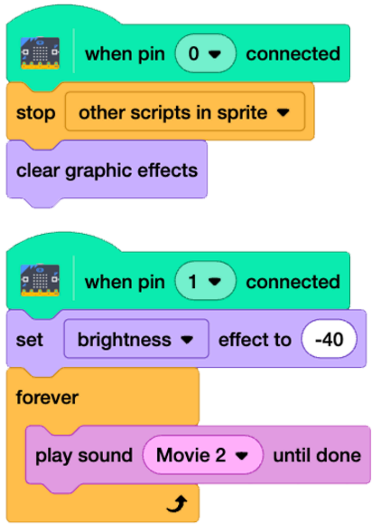
Step 2: While on the Backdrop, select two “when pin connected” hat blocks.

Step 3: Under the Looks category, select a “clear graphic effects” and “set brightness effect” block and add them under the hat blocks. Use a negative number when setting the brightness effect to darken the backdrop. Clear the effect or use a block to set the brightness to a positive number to reset or brighten the backdrop.

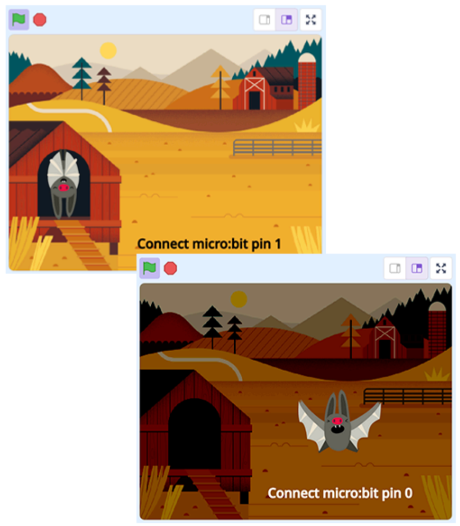
Step 4: Connect alligator clips to two pins and GND on the micro:bit. Then, trigger the code by closing the circuit, holding the end of the clips attached to GND and a pin (or attach the clips to conductive materials).

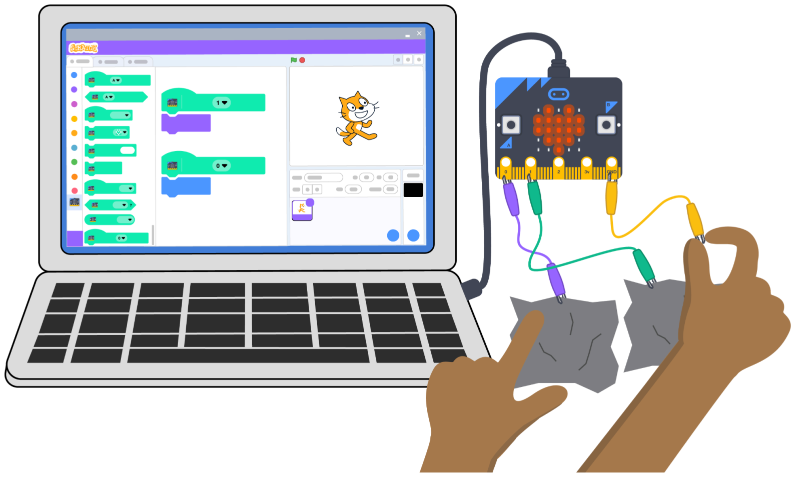
Step 5: What else can you add to customize your project? What if you wanted the backdrop to slowly brighten and darken? How might you use a loop to adjust the brightness slowly?

What if you wanted to add a sprite, like a bat, that would respond to the change in brightness?



```
when pin (0 ▾) connected
stop (other scripts in sprite ▾)
clear graphic effects

when pin (1 ▾) connected
set (brightness ▾) effect to (-40)
forever
    play sound (Movie 2 ▾) until done
```





Example “Day and Night” scripts and closing the circuit to register the pin connection.

Resources:

- micro:bit Coding Cards (Student-Facing)
- micro:bit Day and Night (Example Project)

Option 6: Create Your Own Unique Project

Try creating your own unique project or remix projects to use the micro:bit (versus keyboard keys or the mouse, etc.) to control sprites.

See our Starter Projects page for projects you might remix. Such as:

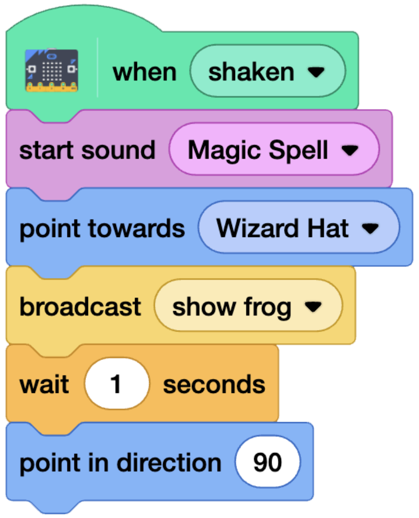
- Make It Fly
- Maze Starter
- Pong Starter
- Spin Art

Or try algorithmar's "Vietnamese Boat Float" project that could use Face Sensing blocks, loudness, keyboard keys, or a micro:bit to control the boat and stop it from hitting the shore.

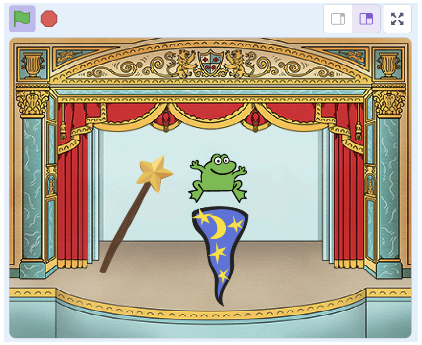
How could you use tilt or button presses to control a sprite's direction or movement or to change variable values?

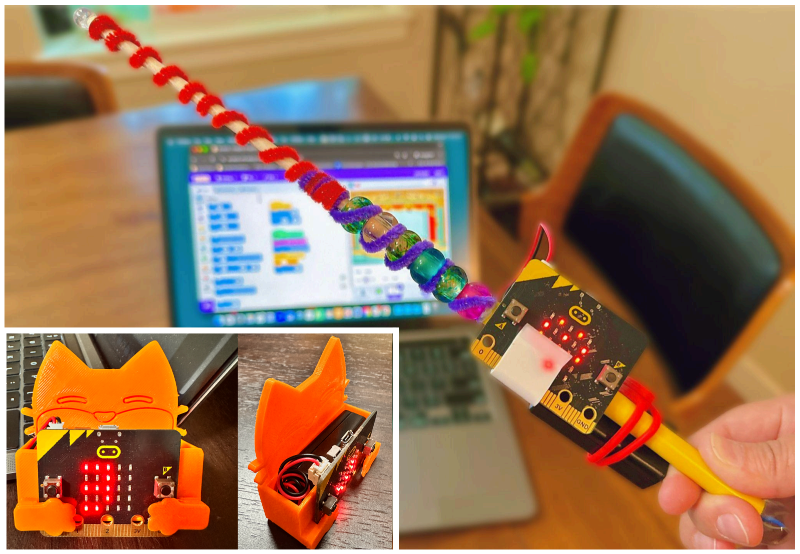
Or create a physical object you can attach your micro:bit to, in order to blend crafting and code. For instance, you could try following ceebee's "Elder Wand [DIY]" tutorial project to create a wand. Then, use a rubber band, pipe cleaners, hair bands, zip ties, adhesive tape, etc., to attach a micro:bit to it (ensure the area where you'll attach is clean and dry). Code animation when the wand with your micro:bit is moved, shaken, or jumped.

Or use a 3D program like Tinkercad to design a holder for your micro:bit and battery pack. Scratch Learning Resource Designer Maren has shared her Scratchy design here.

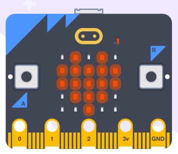


```
when shaken ▾
start sound Magic Spell ▾
point towards Wizard Hat ▾
broadcast show frog ▾
wait 1 seconds
point in direction 90
```





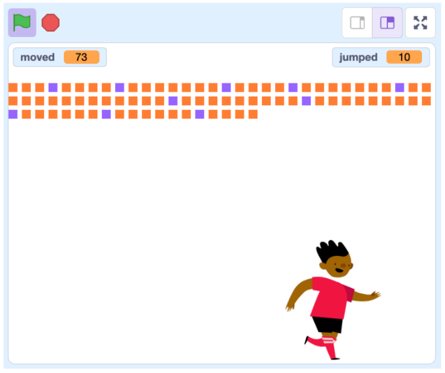
Crafted wand made of found materials with a micro:bit attached, triggering the "micro:bit Magic Wand" example project above. Inset is the 3D printed micro:bit holder designed by Maren.



Interested in capturing some of the data the micro:bit is collecting? Try creating a fitness tracker project that uses variables to store readings such as movement and jumping. Then, you could customize:

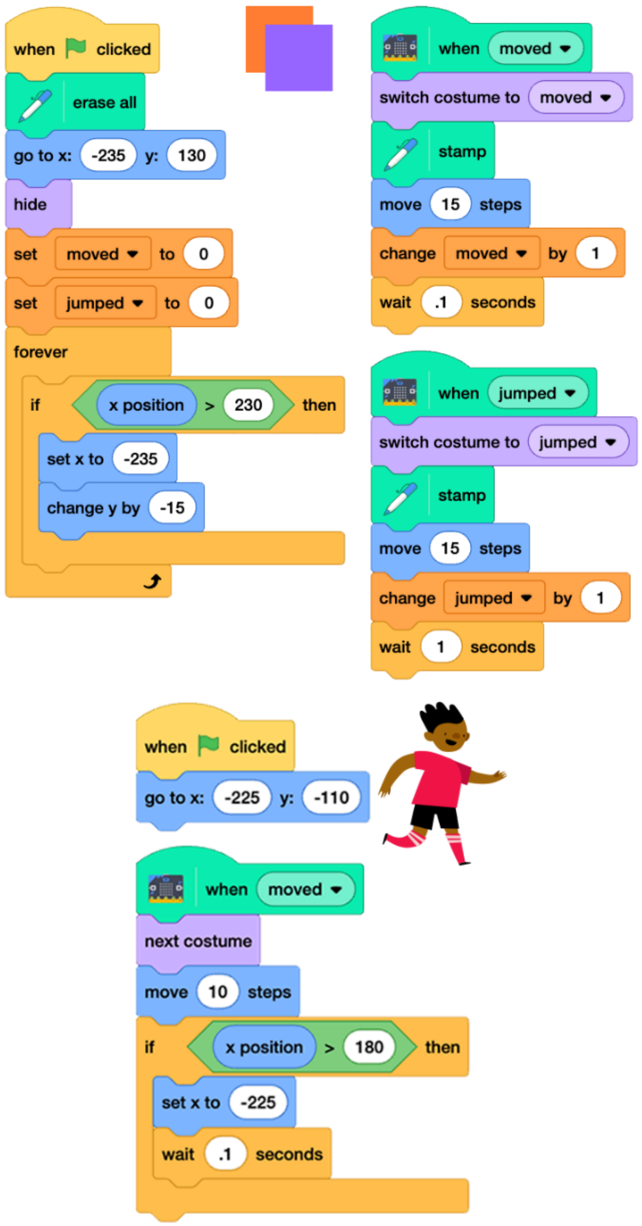
- How could you graphically represent when the micro:bit is moved and jumped? This example to the right, uses the Pen extension to stamp colored squares on the stage each time "moved" or "jumped" is recorded.
- Add a sprite that moves on the stage representing the actions of the player. In this example to the right, movement causes a costume change.

Strap the micro:bit to your shoe or ankle and move your feet or jump up and down and see the results!



Resources:

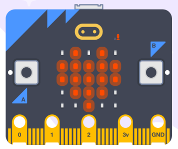
- Starter Projects (Website) - remix and explore
- Scratch Coding Cards (Student-Facing) - individual coding cards are also available
- micro:bit Magic Wand (Example Project)
- micro:bit Fitness Tracker (Example Project)



```
when [flag] clicked
erase all
go to x: -235 y: 130
hide
set moved to 0
set jumped to 0
forever
    if x position > 230 then
        set x to -235
        change y by -15

when moved
switch costume to moved
stamp
move 15 steps
change moved by 1
wait .1 seconds

when jumped
switch costume to jumped
stamp
move 15 steps
change jumped by 1
wait 1 seconds

when [flag] clicked
go to x: -225 y: -110

when moved
next costume
move 10 steps
if x position > 180 then
    set x to -225
    wait .1 seconds
```

Example "micro:bit Fitness Tracker" project, with squares graphically representing moves and jumps, variables showing counts, and for fun a sprite representing movement visually.

Part 3: Reflect and Share



Reflect (15 minutes)

Learners can reflect on their project creation and process as they complete the Show-and-Tell Sharing Sheet. Next, their peers are encouraged to leave feedback or comments on the sheet for the creator as they view the projects in a studio or participate in the gallery walk.

Resources:

- Show-and-Tell Sharing Sheet or Project Gallery Walk Self-Reflection and Peer Feedback Sheet (Worksheet)

Share Option #1: Create a Class Studio to Gather Shared Projects

Studios are a space on Scratch where users can come together to make, share, and collect projects related to a particular theme, idea, or prompt. Set up a class studio* for your learners and add their original asset projects. Learners are encouraged to take time to look at projects and read/listen/interact with them to learn more about their peers.

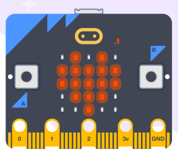
Resources:

- Teacher Account Guide (Written Guide) - This resource contains information on setting up teacher accounts and student accounts, managing classes, and class studios.
- Scratch Studios Guide (Written Guide) - General information on setting up and managing.

**Note: Learners need a Scratch account and access to the online editor to participate in this option.*

Share Option #2: Gallery Walk

Have each participant's project open on their computer or other device. Participants can walk around a room, or take turns sharing their screen in a virtual space, to experience each other’s creations. Or display one project at a time on a large screen. Learners are encouraged to take time to look at projects and read/listen/interact with them to learn more about their peers.



More Things to Try

- Projects with Scratch from micro:bit (Website)
- Scratch Design Journal (Worksheet) - imagine, plan, iterate, and reflect throughout all of the phases of your project's development
- Debugging Strategies Posters (Printable Posters)

Standards Aligned

CSTA Standards	ISTE Standards	CASEL Framework	RITEC Indicators
Link to full standards	Link to full standards	Link to full standards	Link to full standards
● 1B-AP-08 Compare & refine algorithms	● 1.1a Learning Goals	● Self-awareness	● Autonomy
● 1B-AP-10 Create programs	● 1.1d Technology Fundamentals	● Self-management	● Competence
● 1B-AP-11 Decompose problems	● 1.5.b Data Sets		● Creativity
● 1B-AP-12 Modify, remix, or incorporate	● 1.5.c Decompose Problems		● Diversity, equity and inclusion
● 1B-AP-15 Test and debug	● 1.5.d Algorithmic Thinking		
● 1B-AP-17 Describe choices made	● 1.6.b Creative Communicator		
● 1B-CS-02 Model how computer hardware and software work together	● 1.6.c Communicate Complex Ideas		
● 2-CS-02 Design projects that combine hardware and software			

This lesson also fulfills all three of the ISB Indicators of Playful Learning (Choice, Delight, Wonder), developed by the Pedagogy of Play (PoP) research project at Harvard University.

Tip: If you'd like to translate this guide, **click here to make a copy** of this Google doc.