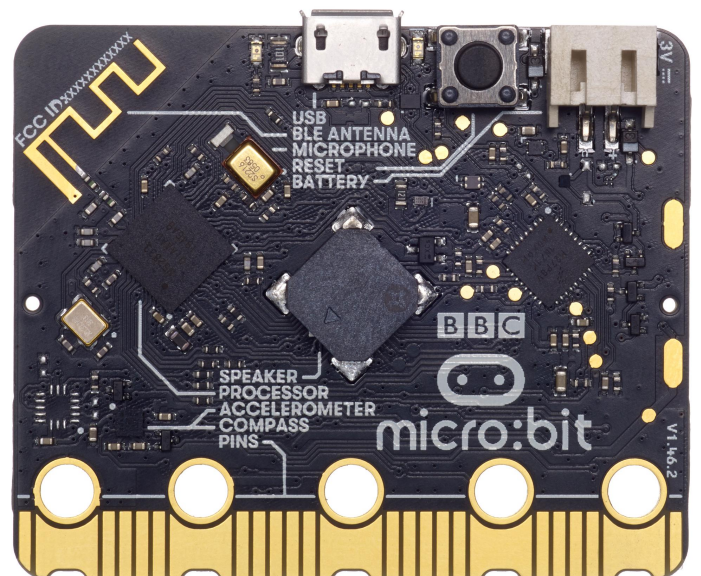
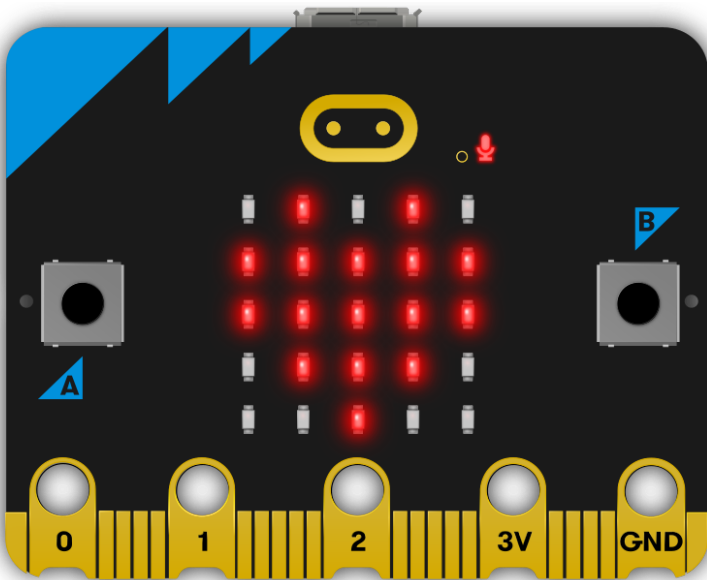
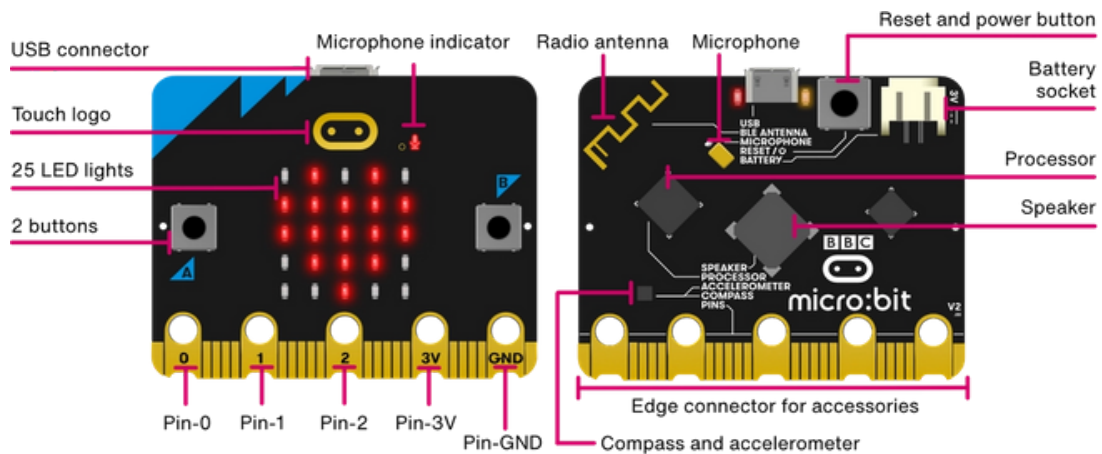


micro:bit kit

Instructions



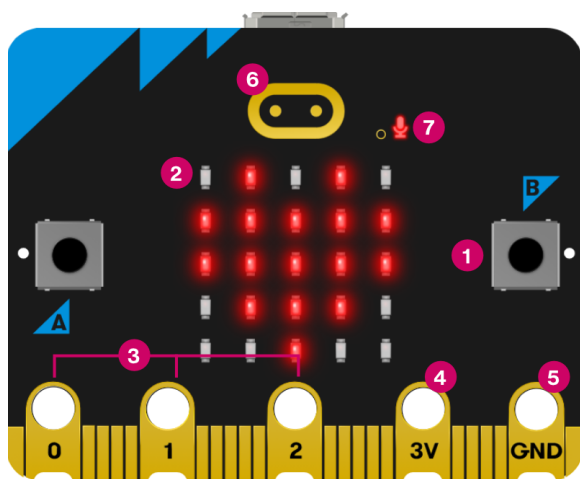
Parts of your micro:bit



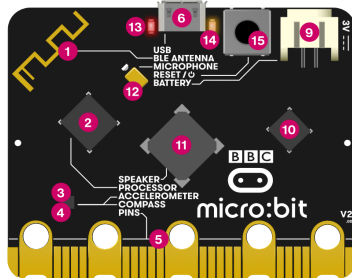
First Steps: <https://microbit.org/get-started/first-steps/introduction/>

User Guide: <https://microbit.org/get-started/user-guide/overview/>

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



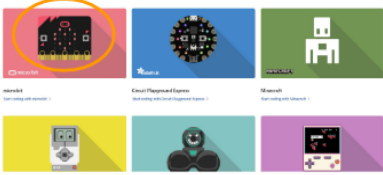
1. Buttons
2. LED display & light sensor
3. Pins - GPIO
4. Pin - 3 volt power
5. Pin - Ground
6. Touch logo - new
7. Microphone LED



- | | |
|--|---|
| <ol style="list-style-type: none"> 1. Radio & Bluetooth antenna 2. Processor & temperature sensor 3. Compass 4. Accelerometer 5. Pins 6. Micro USB socket 7. Single yellow LED 8. Reset button | <ol style="list-style-type: none"> 9. Battery socket 10. USB interface chip 11. Speaker 12. Microphone 13. Red power LED 14. Yellow USB LED 15. Reset & power button |
|--|---|

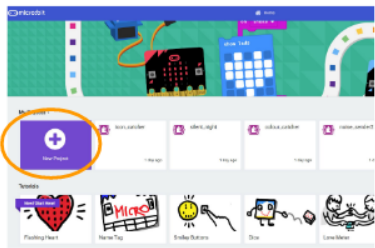
How to load code onto your micro:bit using a computer

1. Using your computer's web browser (Chrome, Edge or Firefox), go to makecode.com



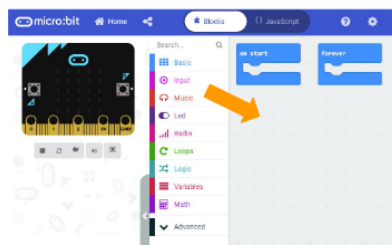
2. Pick the **micro:bit**.

3. Start a **new project** or **open your existing project** in the MakeCode editor. If you are starting a new project, make up a name for it.



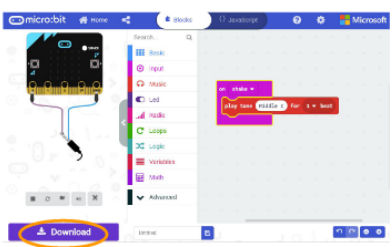
The MakeCode editor should now be shown on your computer, as in the picture in instruction #4 below.

4. Using the instructions for the project you are doing, **get the code written** on the gray part on the right of the screen (arrow in picture below) by dragging blocks there to make or adjust your program.



You can drag any blocks you don't need back over onto the middle menu (Basic, Input, Music, ...) and drop it on the trash can that appears.

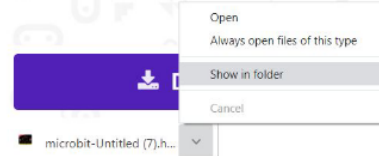
5. When you are satisfied with your code, press the **download** button on the bottom left to save the program to your computer.



You may be asked to choose which folder to save the .hex file in. You can either make a new folder to save your code, or

save it in an existing folder of your choice. Remember where you saved your code's .hex file

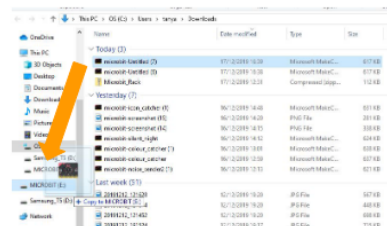
6. Using File Explorer (Windows) or Finder (Mac), open the folder where you saved the .hex file representing your program.



7. Plug the smaller end of the black USB cable into the top of the micro:bit. It can only go in one way. If it won't go in, flip the cable over. Plug the other end into a rectangular USB port on your computer.

A new storage device called MICROBIT should appear in the left column of your Explorer (Windows) or Finder (Mac) on your computer.

8. Load the program onto your micro:bit by **dragging the .hex file** you saved at step 5 over **onto the MICROBIT** storage device like you would to a USB stick.



The amber light on the back of the micro:bit will flash rapidly while the program is being copied over. When the amber light becomes solid, your program is loaded on your micro:bit and is running your code!

Congratulations!

If you are going to use a battery to power your micro:bit for this project, click "Eject" on the MICROBIT storage device your Explorer (Windows) or Finder (Mac) on your computer, then unplug the micro:bit from the computer, and plug in the battery.

How to load code onto your micro:bit using a phone or tablet

1. Install the micro:bit app



On your phone or tablet, install the MakeCode app. Using the Apple **App Store** or Android **Play Store**, search for **microbit**. The app has a green micro:bit logo on a white background. Install it.

There are also links to the apps at <https://microbit.org/get-started/user-guide/mobile/>

2. Pair your micro:bit with the app

Before you can use your device to push code onto your micro:bit, you must pair the micro:bit with your phone or tablet - this is called pairing.

Open the micro:bit app you just installed onto your phone or tablet.

If you are asked to grant permission to use Bluetooth, allow it.

Tap **Choose micro:bit**

Tap **Pair a new micro:bit**

Make sure your micro:bit is powered up by battery (or the USB cable). The red light on the back of the micro:bit will be glowing whenever it's powered up.

Follow the "Enter pairing mode" instructions on your screen. They show you to **Hold both the A and B buttons** on the front of your micro:bit, then **press the RESET button on the back for a moment**, and release the RESET button. Continue to hold both A and B **until the screen shows the "B" Bluetooth logo**, then release the A and B buttons. Your micro:bit is now in pairing mode. You will see a random shape on the micro:bit screen.

On your phone or tablet, press **Next**. Draw the shape from your micro:bit on the device and tap **Next** again.

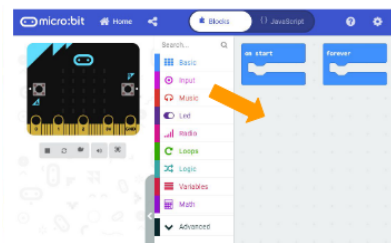
Tap **Next** again to start the app searching for a signal from the micro:bit. They will swap a code to make sure the right micro:bit is paired with your app. When asked if you want to pair, tap **Pair** or yes to allow it.

When the app shows that you are successfully paired, follow its instructions to **RESET** the micro:bit and press **Next** on the app to return to the Choose micro:bit screen. You should see your micro:bit listed with the shape you drew. This shows that your device and micro:bit are now paired and ready to be used together.

3. Send code to your micro:bit

Once paired, return to the app's front screen using the back button. Then tap "Create Code" to load the MakeCode editor onto your phone or tablet.

Using the instructions for the project you are doing, **get the code written** on the gray part on the right of the screen (arrow in picture below) by dragging blocks



there to make or adjust your program.

You can drag any blocks you don't need back over onto the vertical menu with the icons (Basic, Input, Music, ...) and drop it

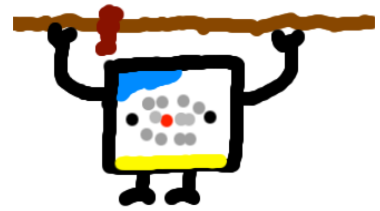
on the trash can that appears.

Tap the **Download** button with the down-arrow on the bottom left of the screen to send the code to your micro:bit.

If you are asked to Reset the micro:bit to Pairing mode, hold buttons A and B and press-release the reset button on the back, as before.

Congratulations! Your code is now running on your micro:bit!

Tug-Of-LED



The tug-of-**LED** is a virtual variation of the popular **tug of war** rope game. Instead of a rope, we'll use the **LED** screen by pulling the **LED** light through the center row.

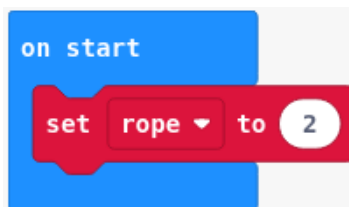
Start by loading <https://makecode.microbit.org/> on your computer, or load the micro:bit app on your tablet or phone. From the list of projects scroll down to **Games** then choose **Tug-of-LED**

There is a video of this project you could watch by choosing **Play Video Lesson**

Choose **Blocks: Start Tutorial** to load the code tutorial. Each of the following instructions will appear at the top of the screen. Follow each instruction, then tap **Next** to continue to the next one:

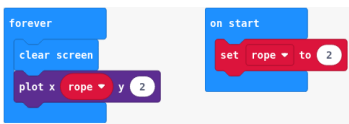
Code

Step 1: Create a new variable rope and put it in the on start. This will track the progress of the game. The rope variable will be used as the x coordinate of the LED to lit so we set it to 2 to start.



Step 2:

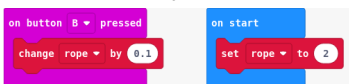
Add a forever loop that turns on the LED at the rope position.



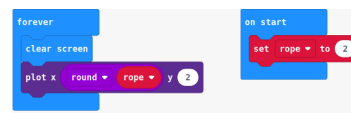
Step 3: Add an event on button A pressed to change the rope value by -0.1.



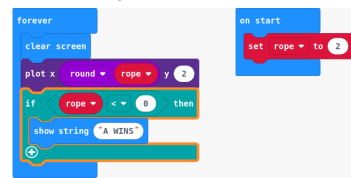
Step 4: Add an event on button B pressed to change the rope value by 0.1.



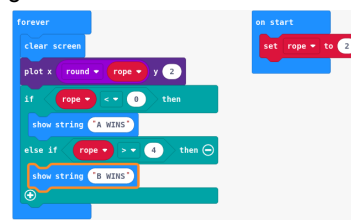
Step 5: Because a button press pulls the rope by 0.1 in either direction, plot the round value of rope to the nearest LED.



Step 6: Back in the forever, add code to test if the rope is negative then show A WINS on the screen.



Step 7: Add an else if condition to test if the rope is greater than 4 then show B WINS on the screen.



Step 8: Follow the printed instructions **How to load code onto your micro:bit** using a computer or phone/tablet to get your code onto the micro:bit

Step 9: Find a friend and start button smashing!

Rock Paper Scissors

Use the accelerometer and the screen to build a **Rock Paper Scissors** game that you can play with your friends!



Start by loading <https://makecode.microbit.org/> on your computer, or load the micro:bit app on your tablet or phone. From the list of projects scroll down to **Games** then choose **Rock Paper Scissors**.

There is a video of this project you could watch by choosing **Play Video Lesson**,

Or choose **Blocks: Start Tutorial** to load the code tutorial. Each of the following instructions will appear at the top of the screen. Follow each instruction, then tap **Next** to continue to the next one:

Step 1: Add an **on shake** block to run code when you shake the micro:bit.

Step 2: Add a hand variable and place the **set hand to** block in the on shake block.

Step 3: Add a **pick random** block to pick a random number from 1 to 3 and store it in the variable named hand. (In a later step each of the possible numbers 1, 2 and 3 is matched to its own picture. The picture is shown on the LEDs when its matching number is picked.)

Step 4: Place an **if** block under the **pick random** and check whether hand is equal to 1. Add a **show leds** that shows a piece of paper. The number 1 will mean paper.

Step 5: Click on the **SHAKE** button in the simulator. If you try enough times, you should see a picture of paper on the screen.

Step 6: Click the + button to add an **else** section.

Step 7: Add a **show leds** block inside the **else**. Make a picture of scissors in the LEDs.

Step 8: Click the + button again to add an **else if** section. Now, add a conditional block for **hand = 2** to the condition in **else if**. Since hand can only be 1, 2 or 3, your code is covering all possible cases!

Step 9: Get one more **show leds** block and put in the **else if**. Make a picture of a rock in the LEDs.

Step 10: Click on the **SHAKE** in the simulator and check to see that each image is showing up.

Step 11: Follow the printed instructions **How to load code onto your micro:bit** using a computer or phone/tablet to get your code onto the micro:bit

Your game is ready! Gather your friends and play rock paper scissors!

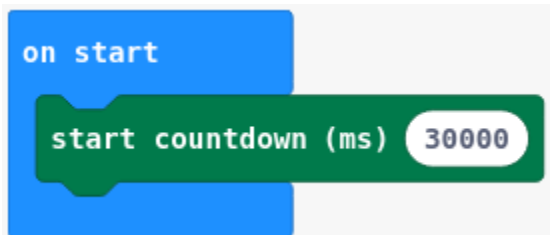
Heads Guess!

This is a simple remake of the famous **Heads Up!** game. The player holds the micro:bit on the forehead and has 30 seconds to guess words displayed on the screen. If the guess is correct, the player tilts the micro:bit forward; to pass, the player tilts it backwards.

Start by loading <https://makecode.microbit.org/> on your computer, or load the micro:bit app on your tablet or phone. From the list of projects scroll down to **Games** then choose **Head Guess!**

There is a video of this project you could watch by choosing **Play Video Lesson**, or you can choose **Blocks: Start Tutorial** to load the code tutorial. Each of the following instructions will appear at the top of the screen. Follow each instruction, then tap **Next** to continue to the next one.

Step 1: Put in code to start a countdown of 30 seconds.



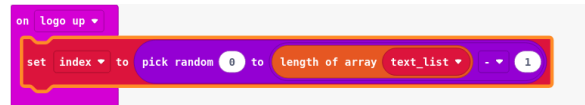
Step 2: Create a text list of words to guess. You will find **Arrays** under **Advanced**.



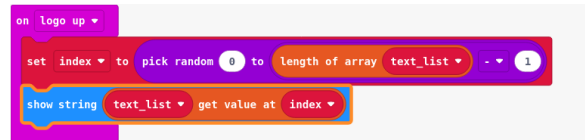
Step 3: Add an event to run code when the micro:bit logo is pointing up. This is the gesture to get a new word.



Step 4: The items in the text list are numbered 0 to length - 1. Add code to pick a random index.



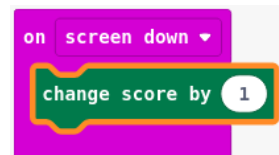
Step 5: Add code to show the value of the item stored at index in the text list.



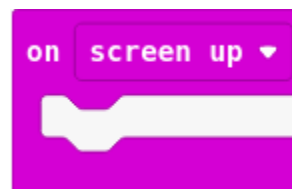
Step 6: Use an event to run code when the micro:bit screen is pointing down. This is the gesture for a correct guess.



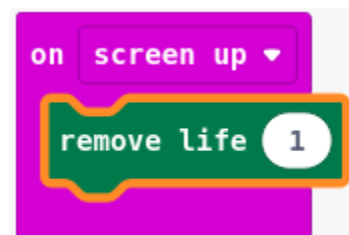
Step 7: Put in code to add points to the score.



Step 8: Add another event to run code when the micro:bit screen is pointing up. This is the gesture for a pass.



Step 9: For the pass gesture, add code to remove a life from the player.



Step 10: Follow the printed instructions **How to load code onto your micro:bit** using a computer or phone/tablet to get your code onto the micro:bit

Reaction Time Game

Start by loading <https://microbit.org/projects/make-it-code-it/reaction-game/> on your computer, tablet or phone.

What you need

- 1 micro:bit
- 4 crocodile clip leads
- Cardboard
- Tin foil
- Glue

Step 1: Make it

Make a reaction game with real physical switches you can bash as hard as you like!



How it works

- Make two physical input switches using cardboard and tin foil like in the picture
- Connect them to the [micro:bit pins](#) as in the picture – one tin foil pad on each switch goes to the micro:bit's GND pin, and the other is connected to pin 1 or pin 2 depending on whether you are player 1 or player 2.
- The program waits a random time between 1 and 5 seconds, then shows a heart on the LED display output.
- You can't hit your button before it lights because it uses Boolean logic to stop anyone cheating! **Boolean variables** can only have two values: True or False. For this it's whether the **game started** (or not) this variable prevents either player pressing their button too soon by only checking which

button is pressed **while** the game has started.

- An infinite loop keeps the game running so you can keep playing.

Step 2: Code it

```
forever
  set game started to false
  pause (ms) pick random 1000 to 5000
  set game started to true
  show icon [heart]
  while game started
    do
      if pin P1 is pressed then
        show string "A"
        set game started to false
      if pin P2 is pressed then
        show string "B"
        set game started to false
    pause (ms) 3000
  clear screen
```

Follow the printed instructions **How to load code onto your micro:bit** using a computer or phone/tablet to get your code onto the micro:bit

Step Counter

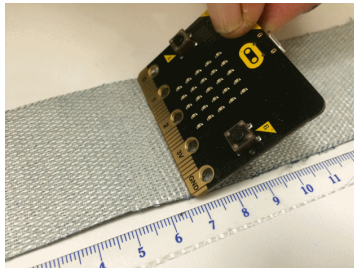
This project turns the micro:bit into a simple step counter. A step counter is also known as a pedometer. Each **shake** event increments a **counter** variable. The step count is displayed on the LEDs.



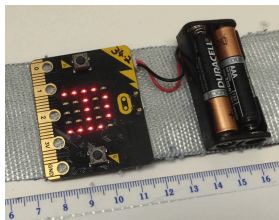
Materials

- Micro:bit
- Battery holder with 2 AAA batteries
- Fabric strip from T-shirt
- Fabric strip from Jeans with Double-sided tape already attached
- Velcro with sticky back

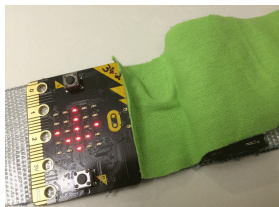
Step 1: Place the micro:bit pins on the tape. Then lower the micro:bit on the tape and press **gently**.



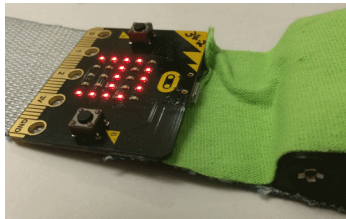
Step 2: Connect the battery holder and place it on the right of the micro:bit.



Step 3: Stick the T-shirt rectangle from the top of the micro:bit, over the battery cables.



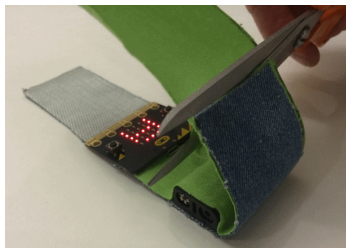
Step 4: Use your fingers to tuck the T-shirt cloth under the micro:bit to give access to the micro-USB plug.



Step 5: Place the T-shirt over the battery holder and stick it up to the end.

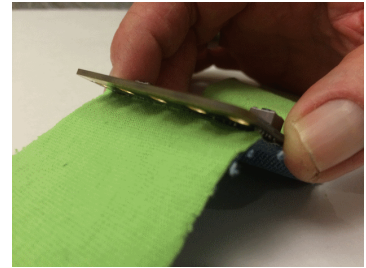


Step 6: Turn over the watch and cut the extra T-shirt material.

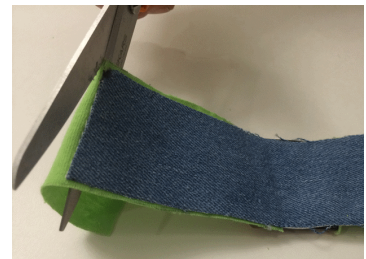


Step 7: Stick the T-shirt cloth on the other side of the watch. Lift the micro:bit pins to slide the cloth under and leave them free

from the tape.



Step 8: Turn over the watch and cut left over T-shirt cloth



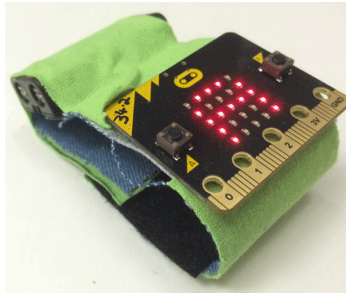
Step 9: Stick the Velcro hooks on the right side of the battery holder on the T-shirt cloth.



Step 10: Stick the Velcro loops on the other end of the jeans fabric.



Step 11: Your counter is done!



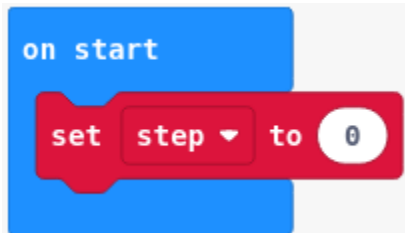
Step Counter Coding

Start by loading <https://makecode.microbit.org/> on your computer, or load the micro:bit app on your tablet or phone. From the list of projects scroll down to **Fashion** then choose **Step counter**.

Choose **Blocks: Start Tutorial** to load the code tutorial. Each of the following instructions will appear at the top of the screen. Follow each instruction, then tap **Next** to continue to the next one.

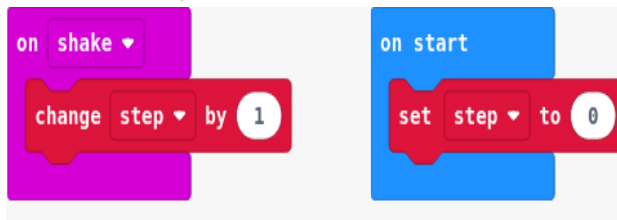
A counter

To build a counter, we'll need a variable **step** to store the number of steps. Choose **Variables** from the left column, and **Make a Variable**. Give it the name: **step**, and click OK. Drag **Set step to 0** into **on start**.



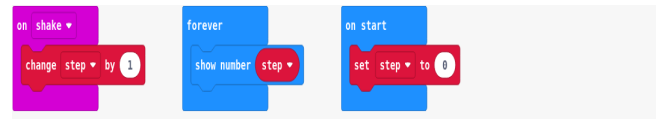
Detecting a step

Assuming you attach the micro:bit to your foot or ankle, it will get shaken when you take a step. We can use the on shake event to detect a step (it should notice a step most of the time). Let's add the code to increment step by 1 when the micro:bit is shaken.



How many steps so far?

We want to always see how many steps were counted. In a forever loop, we add a show number block to display the value of step.



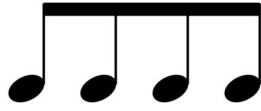
Did you notice there is a lag, or delay, in the display of steps? This is because the step value can change while the micro:bit is displaying a number. To remove the lag, add stop animation after changing the value of step.



Follow the printed instructions **How to load code onto your micro:bit** using a computer or phone/tablet to get your code onto the micro:bit

Strap the micro:bit on your leg and run around to see if it counts your steps!

Musical fruits



Build your own micro:bit piano using bananas!

<https://makecode.microbit.org/projects/banana-keyboard/make>

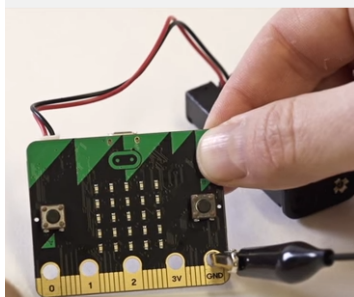
Materials

- Micro:bit, battery holder and 2 AAA batteries
- Banana and Orange
- 2 Crocodile clips

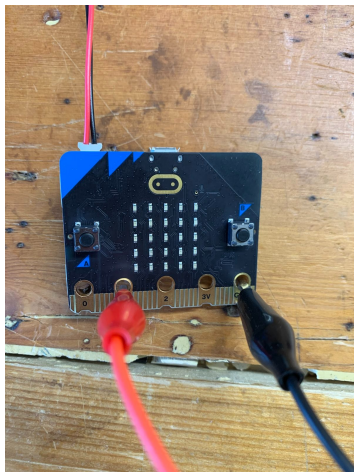
Start by loading <https://makecode.microbit.org/> on your computer, or load the micro:bit app on your tablet or phone. From the list of projects scroll down to **Music** then choose **Banana Keyboard**. Choose **Show Instructions** to load them.

Make

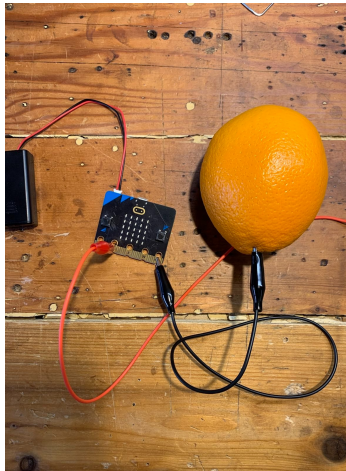
Step 1: Using the 1st crocodile clip, connect the end of the crocodile clip onto the GND pin of the micro:bit.



Step 2: Using the 2nd crocodile clip, connect the end of the crocodile clip onto pin 1 of the micro:bit.

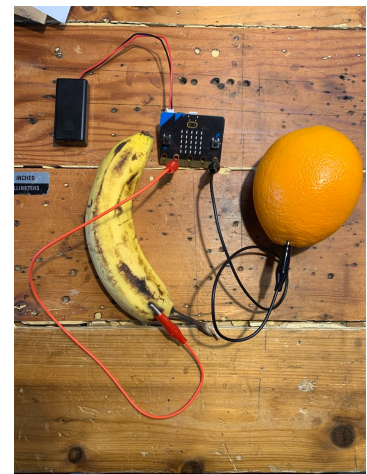


Step 3: Using the 1st crocodile clip, connect the unattached end of the crocodile clip onto the orange.

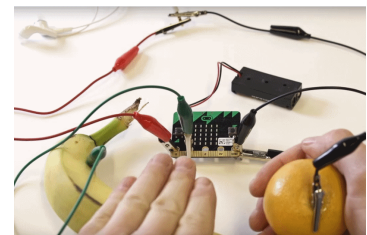


Step 4: Using the 2nd crocodile clip connect the unattached end of the crocodile clip onto the

banana.



Step 5: Touch both fruits and let go of one.



Musical Fruits Code

Have you ever tried making beatbox sounds? Let's try making a beatbox with code and, yes, a banana!

Code the banana key

Start by adding a variable to store a musical note. Rename the variable to sound. Set the value of the variable to the note block Middle A from the Music drawer.

```
on start
  set sound to Middle A
```

We want to play music when the fruit connected to a pin is pressed. So, we register an event handler that executes whenever pin 1 is pressed. Pin 1 is, of course, connected to the banana. Add an on pin pressed block from the Input drawer.

```
on pin P1 pressed
  set sound to Middle A
```

Now, let's create some notes to play when the banana is pressed. Click on the Loops drawer then insert a repeat loop into the on pin pressed block. Click on the Variables drawer and pull out a change item by block and put it into the loop. Rename the variable to sound. Change the value from 1 to 25. This will increase the variable sound from the note frequency of block Middle A to Middle A plus 25 and so on. Put a set to block for sound right after the loop. Set it to Middle A in order to reset the sound after a banana press.

```
on pin P1 pressed
  repeat 4 times
    change sound by 25
  set sound to Middle A
on start
  set sound to Middle A
```

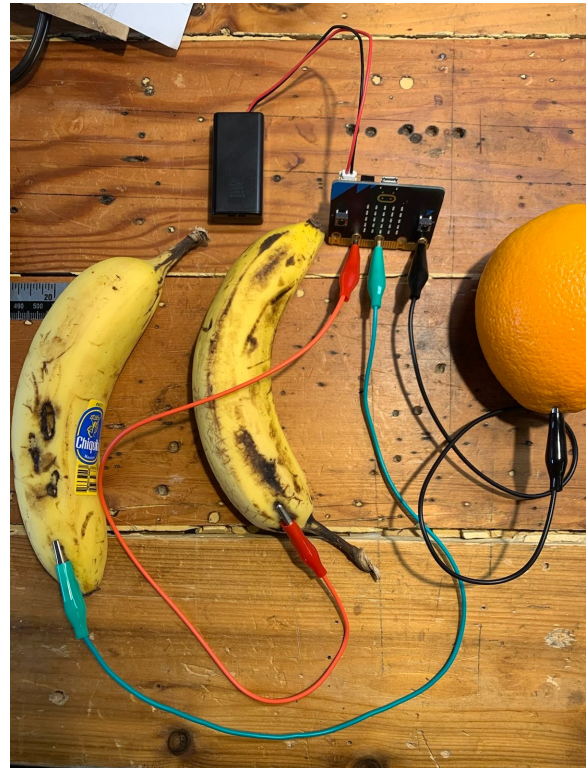
Finally, insert a play tone above the change by. Pull out the sound variable block and drop it in the note slot of play tone. Change the beat fraction from 1 to 1/4.

```
on pin P1 pressed
  repeat 4 times
    do
      play tone sound for 1/4 beat
    change sound by 25
  set sound to Middle A
on start
  set sound to Middle A
```

Click Download and try a banana press. Did you hear 4 notes play?

Add another banana key

Connect another banana but this time connect the crocodile clip to pin 2.



Duplicate the on pin pressed event handler to make a second one. For the new on pin pressed, change the pin name to P2. In the pin P2 event, let's have the frequency in the variable sound decrease by 5 instead of having it increase. Change the 25 in the change by block to - 5. OK, your code now looks like this:

```
on pin P1 pressed
  repeat 4 times
    do
      play tone sound for 1/4 beat
    change Sound by 25
  set Sound to Middle A
on pin P2 pressed
  repeat 4 times
    do
      play tone sound for 1/4 beat
    change Sound by -5
  set Sound to Middle F
on start
  set Sound to Middle A
```

Follow the printed instructions **How to load code onto your micro:bit** using a computer or phone/tablet to get your code onto the micro:bit.

Inchworm

Start by loading <https://makecode.microbit.org/> on your computer, or load the micro:bit app on your tablet or phone. From the list of projects scroll down to **Toys** then choose **Inchworm**. Choose **Show Instructions** to load them: <https://makecode.microbit.org/projects/inchworm>

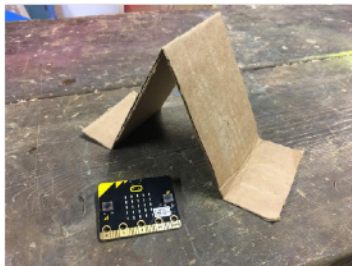
Materials

- Cardboard
- 1 paper clip
- Tape
- Alligator clip wires
- Googly eyes
- 1 Servo motor

Make

Step 1: Body arch

Make two smaller folds at the ends which lay flat on the ground. This is the base of the inchworm's body.



Step 2: Front teeth

Make a small fold in each corner on one of the bases. This is the front of the inchworm which will grip the ground.



Step 3: Back legs

Using scissors or a cutter (watch the fingers!) cut slits in the other end to make finger-like shapes. Fold one up, one down, and so on.



Step 4: Hold the servo upright and turn the arm as far counterclockwise as it will go. Then, carefully remove the arm, and push it back on at the 9 O'clock position.

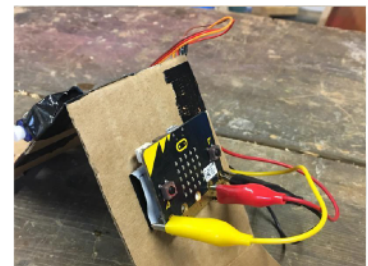


Step 5: Mounting the servo
Attach the servo (with the included double sided tape) on the edge of the cardboard.



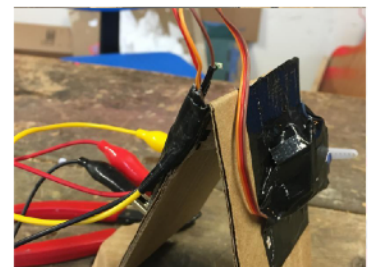
Step 6: Mounting the board

Using tape, mount the micro:bit on the front.



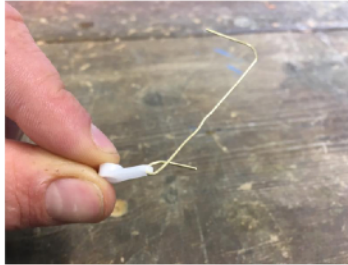
Step 7: Cable clean up

Use tape to route the cables nicely on the inchworm.



Step 8: Attaching the paper clip to servo

Unfold a paper clip and attach it through the hole at the end of the servo arm.



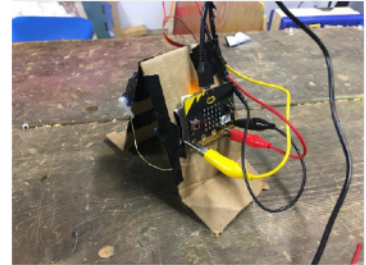
Step 9: Attaching paper clip to cardboard

Slide the folded part of the paper clip onto the cardboard and use tape to hold it there.

Step 10: Ohh pretty

Add some googly eyes, to **snazz** it up!

Step 11: It's ready!!!



Code

In order for the inchworm to move, the micro:bit needs to command the servo to move between 0 and 180 degrees at a certain pace. The code below starts the inchworm moving when the **A** button is pressed.

You might notice that the inchworm goes really slow or maybe just won't move at all. Try to improve the design of your legs and teeth until the inchworm can go as fast as possible. Also, trying it on the carpet gives it more grip and helps to avoid skidding.

```
on button A pressed
  set go to 1

on button B pressed
  set go to 0

forever
  if go = 1 then
    servo write pin P0 to 0
    pause (ms) 1000
    servo write pin P0 to 180
    pause (ms) 1000
```

Follow the printed instructions **How to load code onto your micro:bit** using a computer or phone/tablet to get your code onto the micro:bit

More Projects

There are so many **more fun micro:bit projects** to be found on the web!

Adafruit micro:bit projects

<https://learn.adafruit.com/category/micro-bit>

Hackaday micro:bit projects

<https://hackaday.com/tag/microbit/>

<https://hackaday.com/tag/bbc-microbit/>

Micro:bit MakeCode

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

Or just search the web for **microbit projects**, there are so many cool ones out there!

Buy more parts for your projects

If you want to purchase other parts for your micro:bit projects, check out

BC Robotics <https://bc-robotics.com/> (on Vancouver Island!)

PiShop.ca <https://www.pishop.ca/product-category/bbc-microbit/> (Canada)

Adafruit <https://www.adafruit.com/> (USA)

Sparkfun <https://www.sparkfun.com/> (USA)

Kitronik <https://kitronik.co.uk/> (UK)