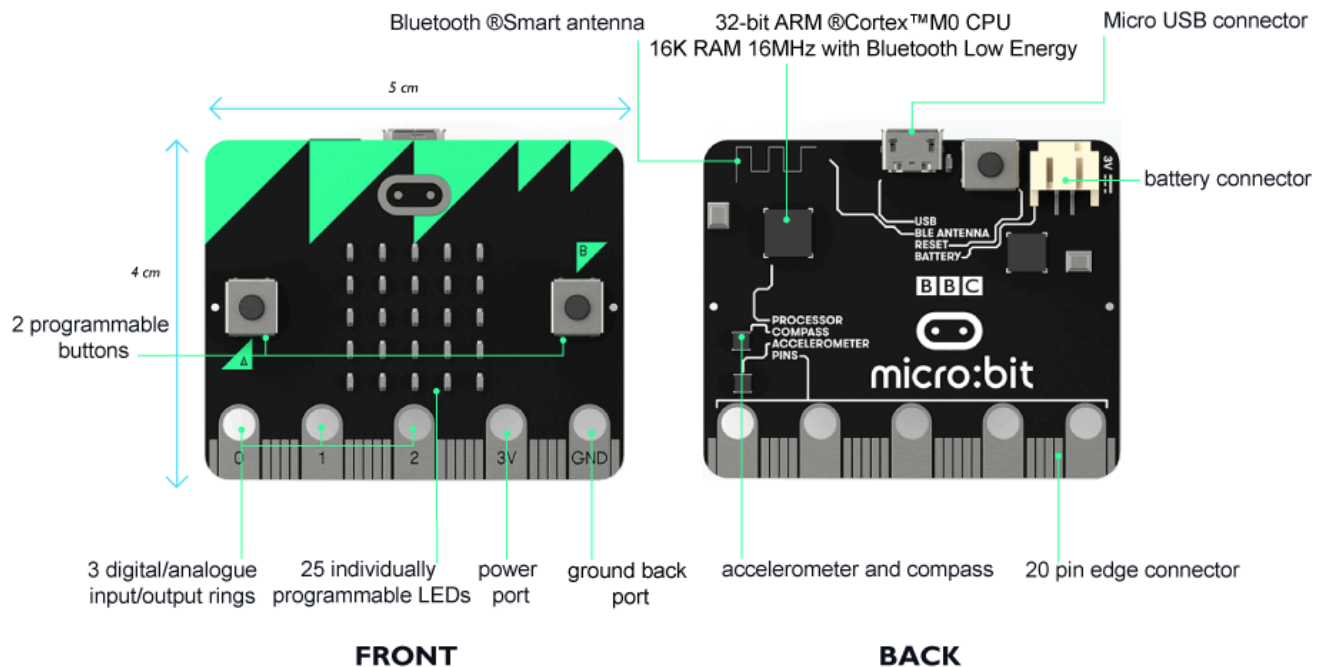




Meet the micro:bit



Overview:

Your micro:bit has the following physical features:

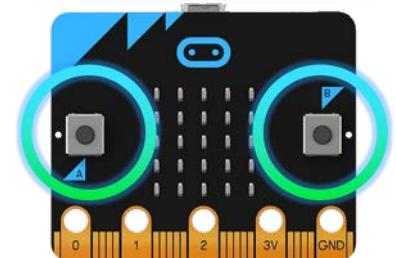
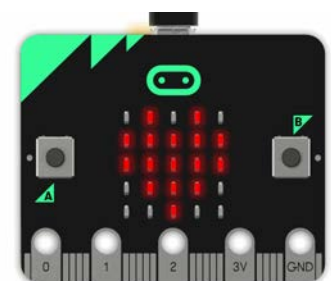
- 25 individually-programmable LEDs

LED stands for Light Emitting Diode. The micro:bit has 25 individually-programmable LEDs, allowing you to display text, numbers, and images.

→ [learn more about the LED screen](#)

- 2 programmable buttons

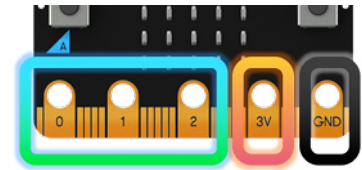
There are two buttons on the front of the micro:bit (labelled A and B). You can detect when these buttons are pressed, allowing you to trigger code on the device.





- Physical connection pins

There are 25 external connectors on the edge connector of the micro:bit, which we refer to as 'pins'. Program motors, LEDs, or other electrical components with the pins, or connect extra sensors to control your code!



→ [learn more about the physical connection pins](#)

- Light sensor

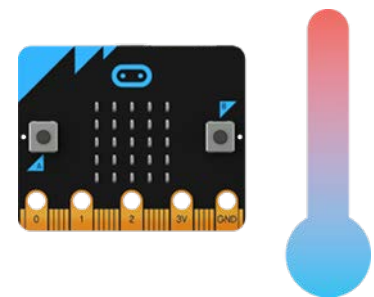
By reversing the LEDs of the screen to become an input, the LED screen works as a basic light sensor, allowing you to detect ambient light.



- Temperature sensor

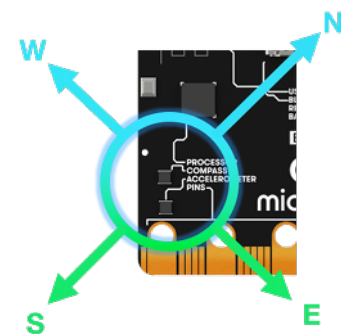
This sensor allows the micro:bit to detect the current ambient temperature, in degrees Celsius.

It does not have a dedicated temperature sensor. Instead, the temperature provided is actually the temperature of the silicon die on the main CPU. As the processor generally runs cold (it is a high efficiency ARM core), it's a good approximation of the ambient temperature. However, if you give the processor a lot of work to do that can affect the accuracy of your data.



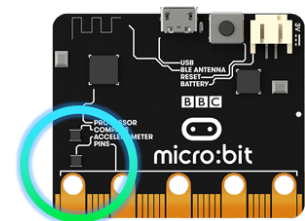
- Motion sensor – compass

The compass detects the earth's magnetic field, allowing you to detect which direction the micro:bit is facing. The compass has to be calibrated before it can be used.



- Motion sensor - accelerometer

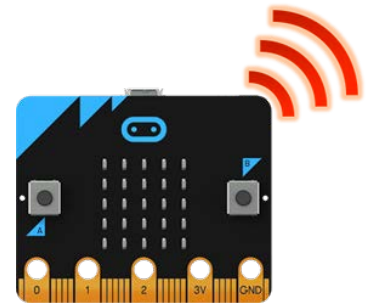
An accelerometer measures the acceleration of your micro:bit; this component senses when the micro:bit is moved. It can also detect other actions, e.g. shake, tilt, and free-fall.





- **Wireless Communication - via Radio**

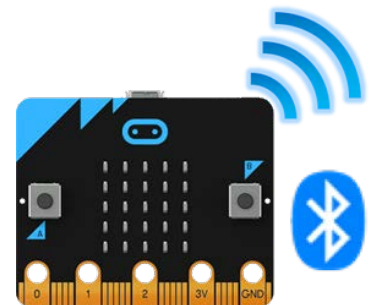
The radio feature allows you to communicate wirelessly between micro:bits. Use the radio to send messages to other micro:bits, build multiplayer games, and much more!



- **Wireless Communication - via Bluetooth**

A BLE (Bluetooth Low Energy) antenna allows the micro:bit to send and receive Bluetooth signals. This allows the micro:bit to wirelessly communicate with PCs, Phones, and Tablets, so you can control your phone from your micro:bit and send code wirelessly to your device from your phone!

Before using the Bluetooth Antenna you will need to pair your micro:bit with another device. Once paired, you can send programs wirelessly to your micro:bit.



- **USB interface**

The USB interface allows you to connect the micro:bit to your computer via a micro-USB cable, which will power the device and allow you to download programs onto the micro:bit.





Using the micro:bit

What's in the box?

- micro:bit enclosed in anti-static storage sleeve
- AAA battery pack that connects to the micro:bit
- 2 AAA batteries
- micro USB cable

You can connect your microbit to the battery pack or to your computer with the micro USB cable provided. The pre-programmed **SET UP** file will run and you will be able to get to know your micro:bit a little better before you begin coding.



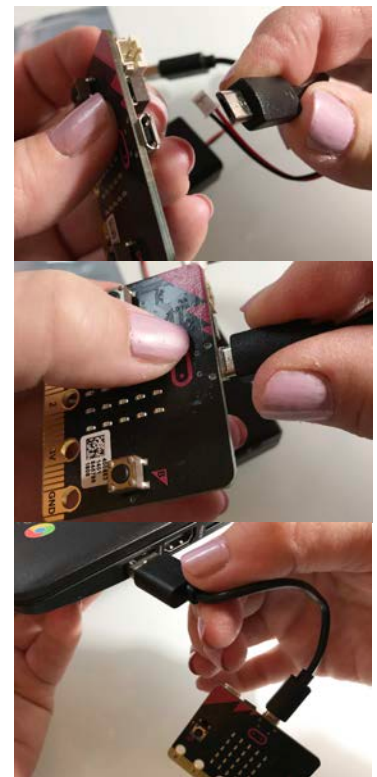
Step 1: Connect It

When you power it up for the very first time, press the buttons when your micro:bit asks you to and watch the lights display cool patterns and games. You can give it a shake and try to turn on the entire grid light. Play "Chase the Dot" and tilt your micro:bit to catch the flashing dot with the steady dot.

NOTE: This pre-programmed "SET UP" command file will be overwritten when you load your first program file.

Connect the micro:bit to your computer via the micro USB cable provided. Macs, PCs, Chromebooks and Linux systems (including Raspberry Pi) are all supported. Your micro:bit will show up on your computer as a drive called 'MICROBIT'.

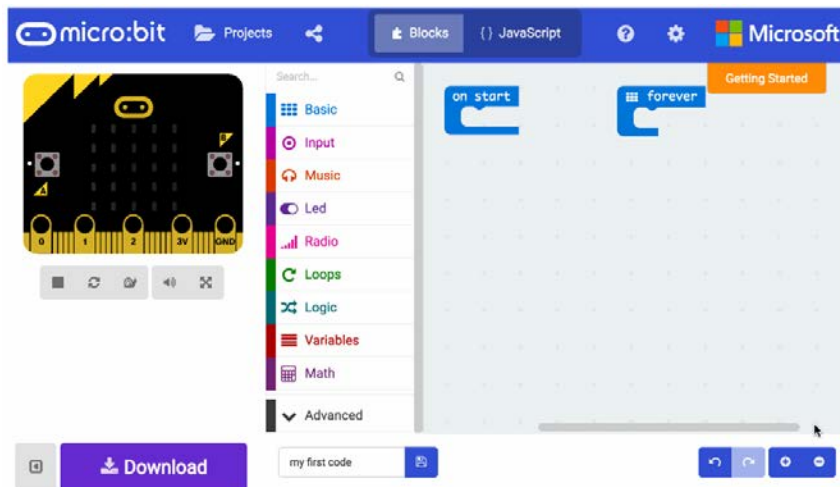
NOTE: You may substitute a longer micro USB as needed. The small end connects as shown. Be careful of too much force or twist as you put it in or remove it. Some computer networks may not allow USB drives, and some computer USB ports may be problematic. You can always save your file to a network drive or Google Drive, and then use another computer elsewhere to load your code into your micro:bit.





Step 2: Program It

Using MakeCode, a web-based visual block-based coding editor, write your first micro:bit code. Drag and drop some blocks and try your program or test your code on the Simulator in the Javascript Blocks Editor



Microsoft MakeCode files have a name you give them when you save and download a project and file extension (.hex) indicating is a hex code file that will run automatically once loaded into a powered micro:bit.

[Click here to try making your first code program in MakeCode](#)

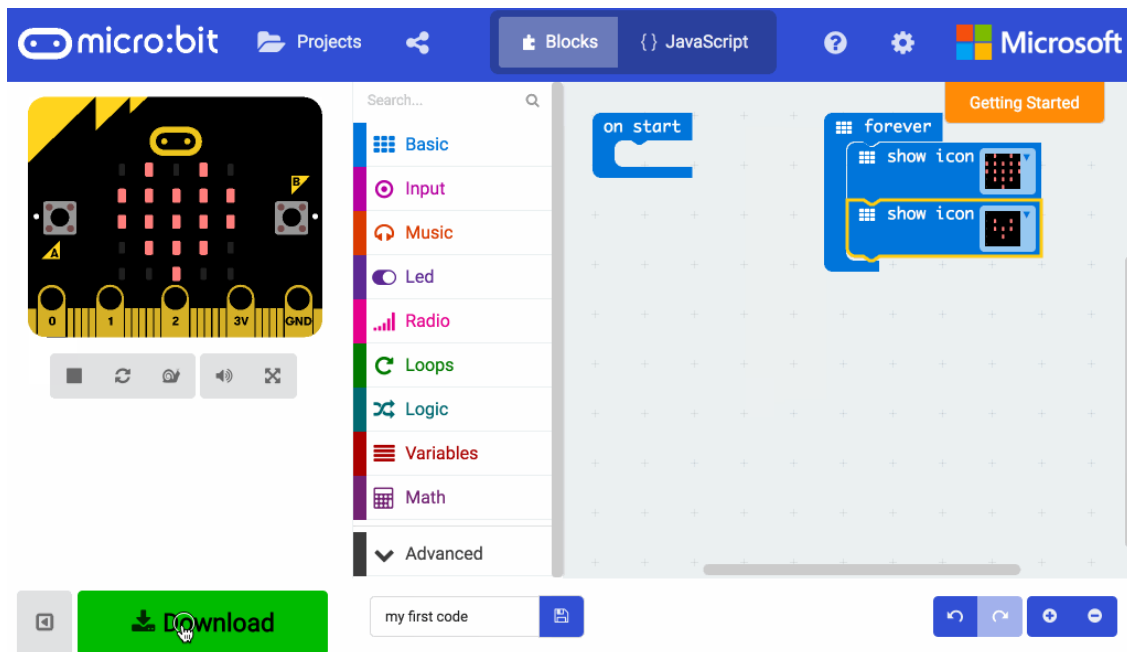
NOTE: Always keep your BBC micro:bit in the anti-static bag when not using it. It's good practice for students to ground themselves before handling it. Only handle the BBC micro:bit by its edges and avoid touching the components when the power is running.



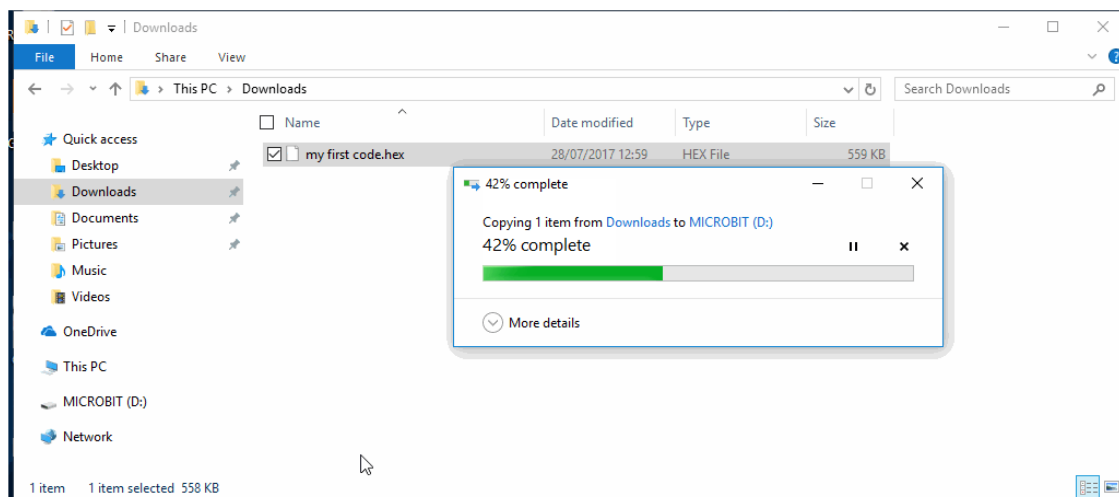


Step 3: Download It

Click the Download button in the MakeCode editor. This will download a 'hex' file, which is a compact format of your program that your micro:bit can read.



Once the hex file has downloaded, copy it to your micro:bit just like copying a file to a USB drive. Attach the micro USB large end of the cord into a computer, and save the file to the micro:bit as if it were a USB storage device. You can drag and drop the .hex file on a Mac. On Windows you can right click and choose "Send To→MICROBIT."



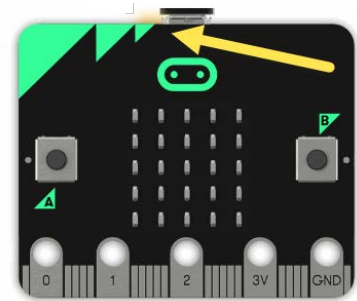


Step 4: Play It

The micro:bit will pause and the yellow LED on the back of the micro:bit will blink while your code is programmed. Once that's finished the code will run automatically!

The MICROBIT drive will automatically eject and come back each time you program it, but your hex file will be gone. The micro:bit can only receive hex files and won't store anything else!

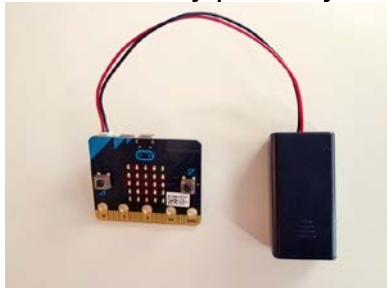
(You will not actually see your file listed once you transfer it, but you will observe the data transfer LED blinking, and your program will run automatically once copied.)



Step 5: Power It – micro:bit only

You can power your micro:bit with

a 3V AAA battery pack in your kit

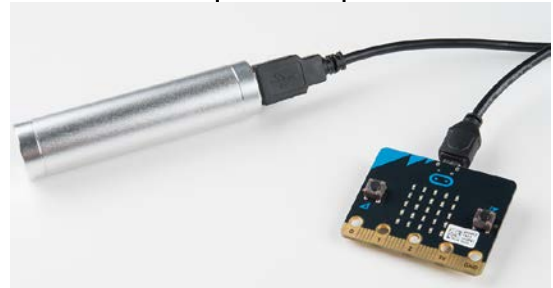


a 5V USB wall adapter



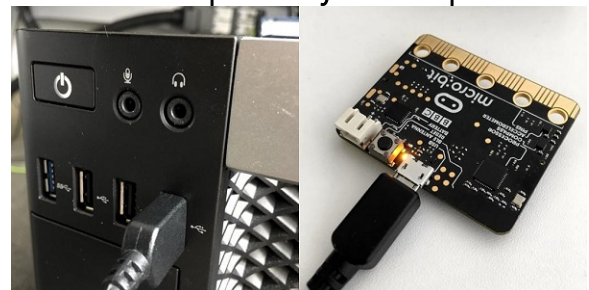
OR

a 5V USB portable power cell



OR

a USB port on your computer.



NOTE: The AAA 3V battery pack **WILL NOT** work for the COZIR Sensor – the COZIR sensor requires 3.5V to run efficiently – the battery pack does not supply enough power.

If you are using the COZIR sensor you **MUST** have the micro:bit plugged into either:

- a 5V portable power cell
- a 5V USB wall adapter
- a 9V battery attached to the barrel jack on the sensor board
- the USB port on your computer

