Grade 5/6

Micro:Bits

Any questions?
Reach out
svcamp@engr.uvic.ca

Micro:Bits Intro

Introduction to Micro:bits programming. Familiarize students with the MakeCode editor and block coding concepts. Practice the update cycle of coding, downloading code to Micro:bit, testing, and debugging.

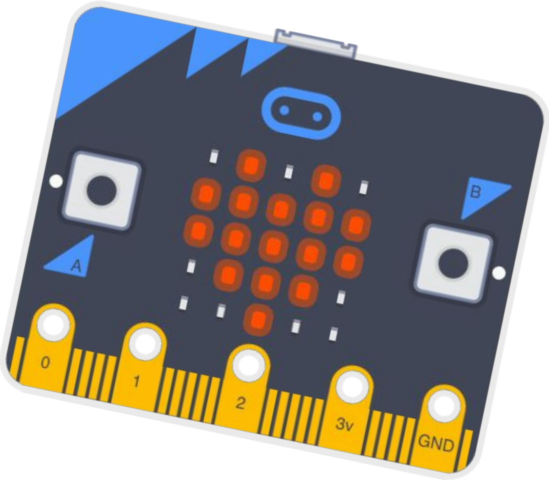
What is a Micro:Bit?

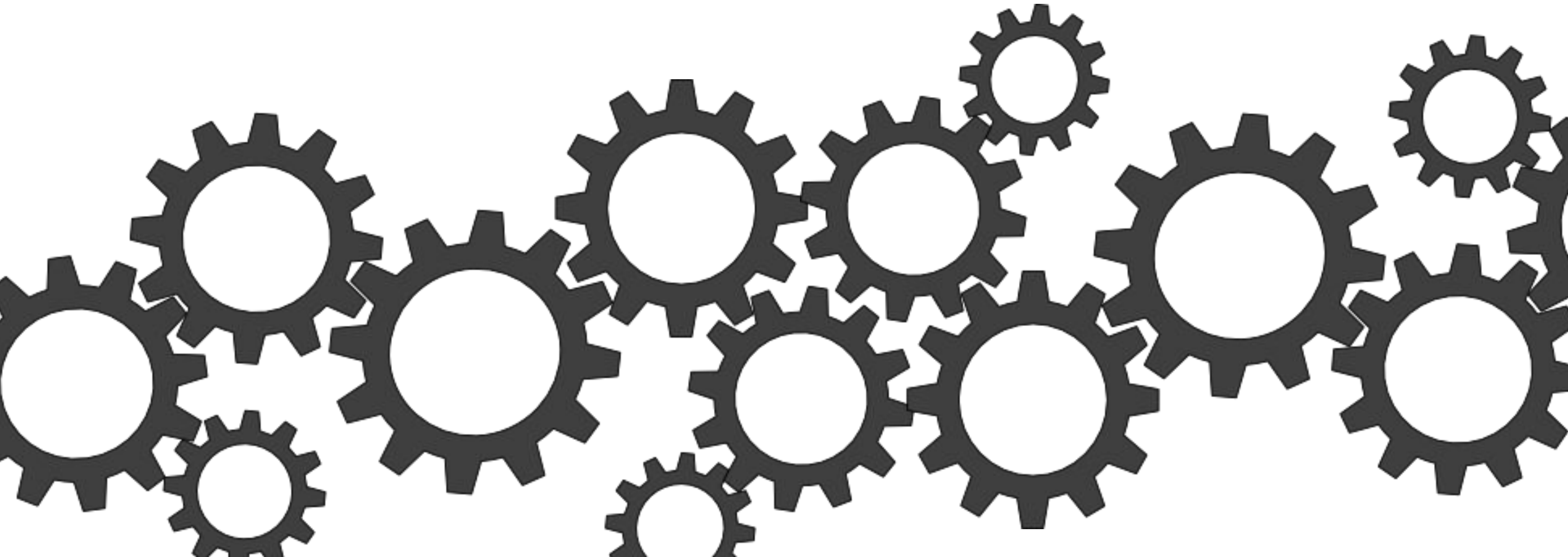
A Micro:bit is a miniature computer with a 5x5 LED display grid. We can use the MakeCode editor to code a program to execute on our Micro:bit. A program is simply a set of instructions for our Micro:bit to complete.

Materials

- Micro:bit kit
  - 1 Micro:bit
  - 1 Battery pack
  - 1 Micro USB cable

- Optional: 1 Dongle adapter if using Mac





Micro:Bits Intro

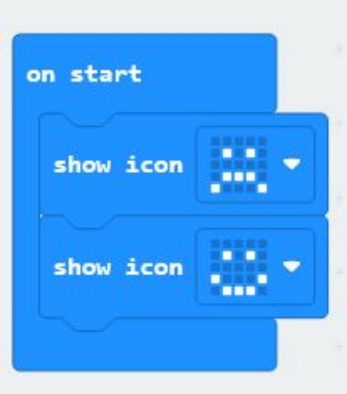
Procedure

Using MakeCode Editor and Uploading a Program

Step 1: Open MakeCode for Micro:Bit https://makecode.microbit.org/

Step 2: Create a new program and name your program

Step 3: Drag "show icon" or "show LEDs" blocks into the "on start" frame. Your block code should look something like this:



```
on start
    show icon
    show icon
```

Step 3: Detailed instructions on how to flash (or upload) the code to the micro:bit for different computers are given here: https://support.microbit.org/support/solutions/articles/19000013986-how-do-i-transfer-my-code-onto-the-micro-bit-via-usb

Basic steps to flash code to your Micro:Bit:

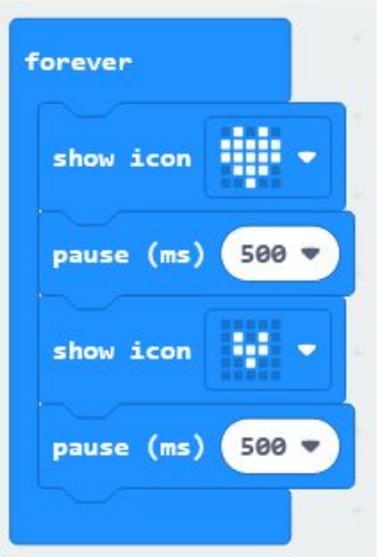
(i) Plug the micro:bit into your computer with the usb cord
(ii) Download the .hex file from the Micro:bit website
(iii) Drag the .hex file over the Micro:bit drive

Step 4: Your program should automatically start running on your Micro:bit once you have flashed the program to your device.

Step 5: Try out different variations of your program by changing what is displayed on your Micro:Bit's LED screen. Remember, any changes to your program will require you to re-flash it to your device. Practice this process!

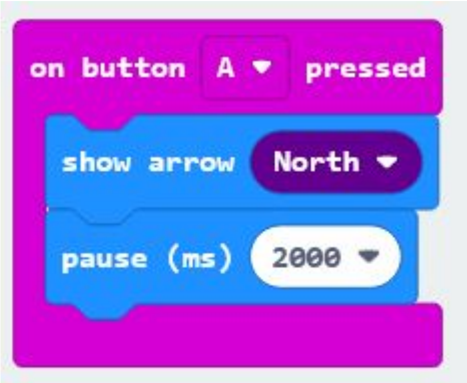
Procedure Continued

Step 6: What happens if we put some display blocks into the "forever" frame instead of the "on start" frame? Try this, and practice flashing the new code onto your Micro:bit again. Add pause blocks as well to control the speed of the animation, your code should look something like this:



```
forever
    show icon ▾
    pause (ms) 500 ▾
    show icon ▾
    pause (ms) 500 ▾
```

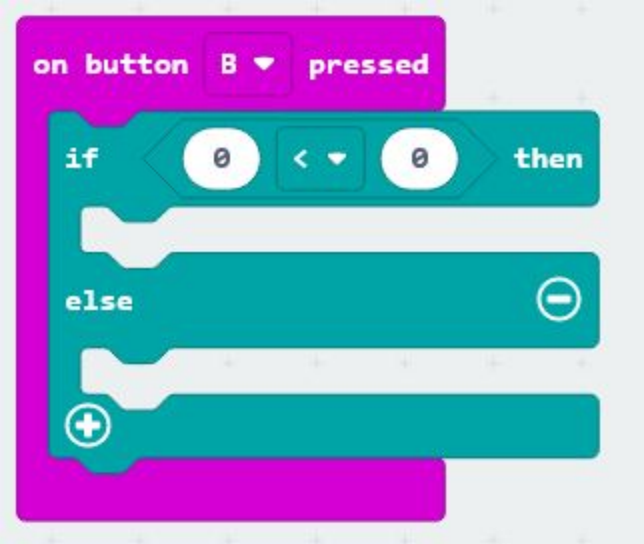
Step 7: We can also make different things happen on the micro:bit by pressing buttons. Go to the Input tab and bring out an "on button A pressed" frame. You can change whether this frame is triggered by pressing A, B, or both at the same time. At the bottom of the Basic tab, bring out the "show arrow North" block and add it to your "on button A pressed" frame. The micro:bit has a built in compass.



```
on button A ▾ pressed
    show arrow North ▾
    pause (ms) 2000 ▾
```

Procedure Continued

Step 8: Duplicate the "on button A pressed" block and use the drop-down arrow to switch it to a B. We'll try to use the temperature sensor to display a different image to show us if it is hot or cold. Since our program needs to make a decision, we will need to use an if-then-else block. These can be found under the logic tab. Drag out a "0<0" block, also from the logic tab, and use it to replace the "True" block.

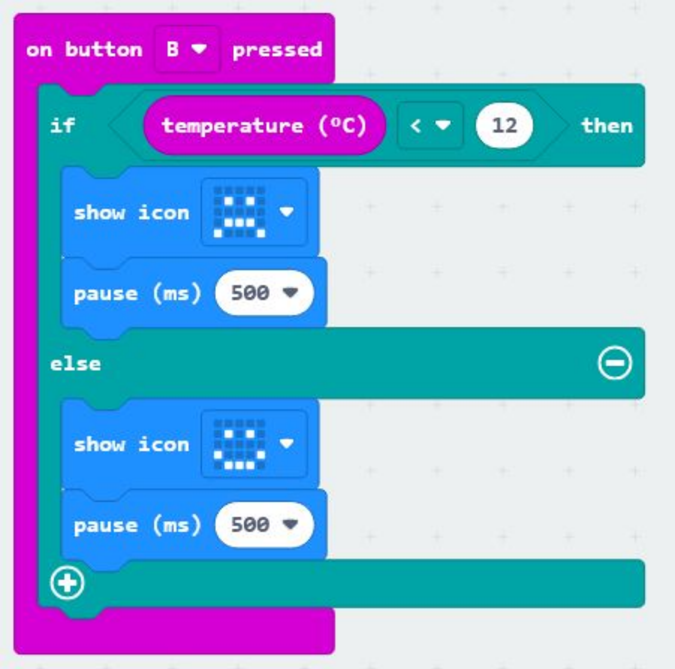


```
on button B ▾ pressed
  if 0 < ▾ 0 then

  else ⊖

  ⊕
```

Step 9: Under the input tab, find the "Temperature" block, and replace the left 0 with it. Decide what temperature will be your cut off for a "cold" temperature. I chose 12 degrees celsius but the choice is yours! When the B button is pressed our new code should check to see if the temperature is less than 12. If it is, then we display the sad face and pause. If not, we display a happy face. (see next page for example)

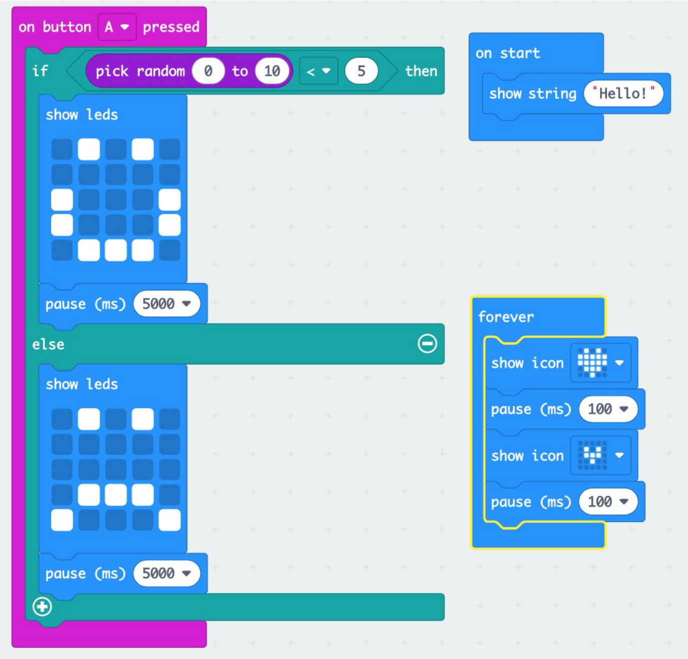
Procedure Continued

(Step 9 example)



```
on button B ▾ pressed
    if temperature (°C) < ▾ 12 then
        show icon ▾
        pause (ms) 500 ▾
    else
        show icon ▾
        pause (ms) 500 ▾
```

Step 10: Instead of checking the temperature, we could compare a random number to our value instead. The random number chooser can be found in the Math block.



```
on button A ▾ pressed
    if pick random 0 to 10 < ▾ 5 then
        show leds
            . # . # .
            . . . . .
            # . . . #
            # . . . #
            . # # # .
        pause (ms) 5000 ▾
    else
        show leds
            . # . # .
            . . . . .
            . . . . .
            . # # # .
            # . . . #
        pause (ms) 5000 ▾

on start
    show string "Hello!"

forever
    show icon ▾
    pause (ms) 100 ▾
    show icon ▾
    pause (ms) 100 ▾
```

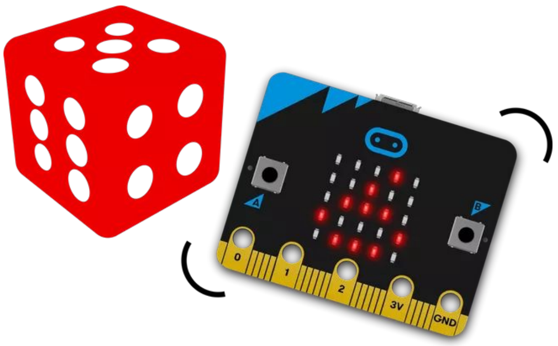
Micro:Bits Intro

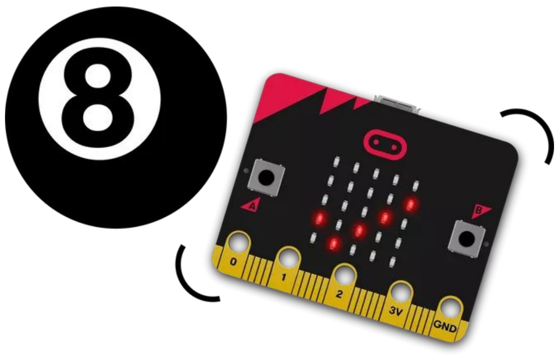
Additional Activities

Here are two activities that use the skills we just practiced in this lesson while adding a twist, motion! The Micro:Bit has a built in accelerometer that allows the device to detect movement. The following projects will incorporate some basic programming skills alongside this motion feature.

Dice: https://microbit.org/projects/make-it-code-it/dice/

Magic 8 Ball: https://microbit.org/projects/make-it-code-it/magic-8ball/





The first activity walks you through how to program your Micro:bit to act as a digital dice. This is the more simple of the two activities, we recommend you start with this! Follow the steps and remember the skills we just learned in this lesson to help you through the programming and uploading of the file.

The second activity will walk you through the creation of a program that will transform your Micro:Bit into a magic 8 ball. This is the more complex of the two activities and will use some programming functions of the Micro:BIt that we have not explored yet. We encourage you to get creative and explore the possibilities of your Micro:Bit on your own.