Creating a Step Counter with the micro:bit
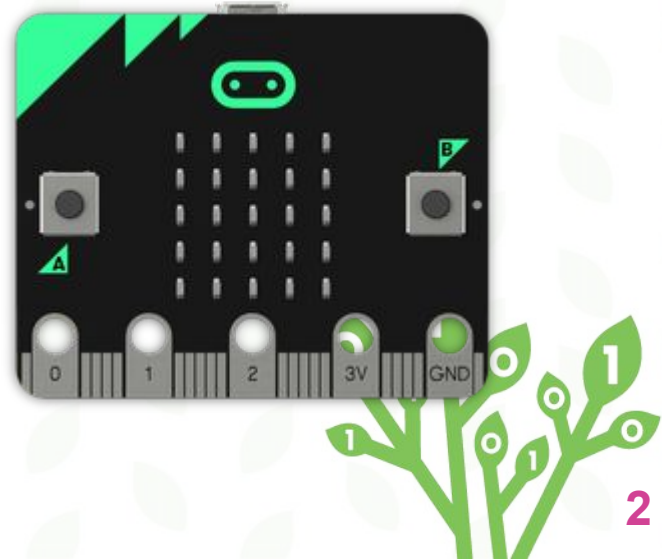
with

lynxcoding.club

DESCRIPTION
Creating an Interactive Step Counter

Students (grades 4-8) will create a Step Counter using the micro:bit.

Students will code this app using Makecode at: makecode.microbit.org/

Students will create a Step Counter that will keep track of their steps and compare the result to the steps taken by Terry Fox during his Marathon of Hope.
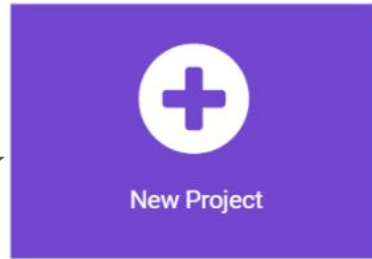
Start Coding the Step Counter!

1. Go to
makecode.microbit.org/

2. Select *New Project*.

New Project

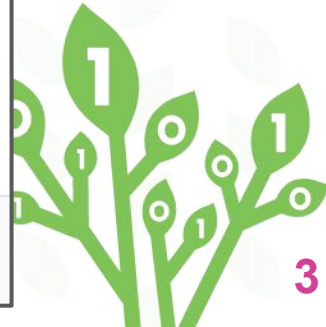**3. Name your project and click on *Create*.**

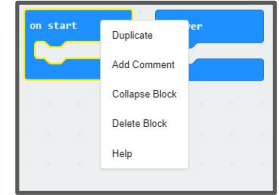Create a Project 🤩🤩🤩

Give your project a name.

stepscounter

> Code options

Create ✔

Adding the *on shake* block...

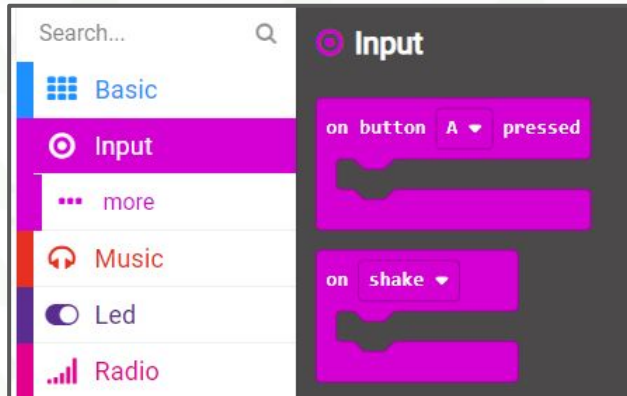**1. Remove the on start and forever blocks by right clicking on each one and selecting delete, or by clicking and dragging them to the left of the screen into the trash can that appears.**

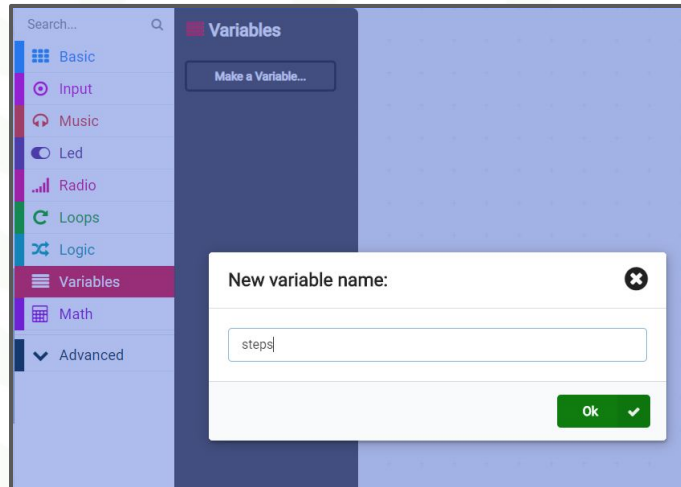**2. We will need to add an Input block so that the micro:bit knows when you shake it (i.e., when you've taken a step). Go to Input, select the on shake block and drag it to the right into your work area.**

Creating a Variable...

1. We need to create a variable that will keep track of the number of steps you take.

2. Click on *Variables* then *Make a Variable* and name the variable *steps*. Click on *Ok*.
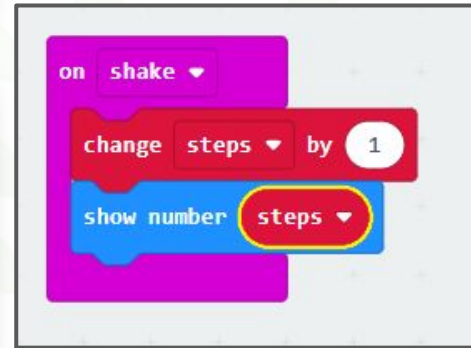
Increasing the *steps* variable with each step...

1. On the left side of your screen, go back to *Variables*, select the *change steps by 1* block, drag it over and snap it inside the *on shake* block as shown.

2. To display the number of steps, go to *Basic*, select *show number* and snap it inside of the *on shake* block as shown.

3. Rather than 0, we want it to display the number of steps you take. Go back to *Variables* and select the *steps* block and drag it inside of the *show number* block replacing the 0.

Show an encouraging message when button A is pressed…

1. *Go to* *Input**, select the* *on button A pressed* *block and drag it to your work area.*

2. Go to *Basic**, select the* *show string* *block and drag it inside of the* *on button A pressed* *block.*

3. Replace the word "*Hello!*" *in the* *show string* *block with an encouraging message, for example, keep going!*

Displaying steps and clearing the screen...

1. *Go to* *Input*, *select the* *on button A pressed* *block, drag it into your work area. Use the pull-down menu beside the letter* *A* *and change it to* *B.*

2. *Go to* *Basic*, *select the* *show number* *block and place it inside of the* *on button B pressed* *block.*
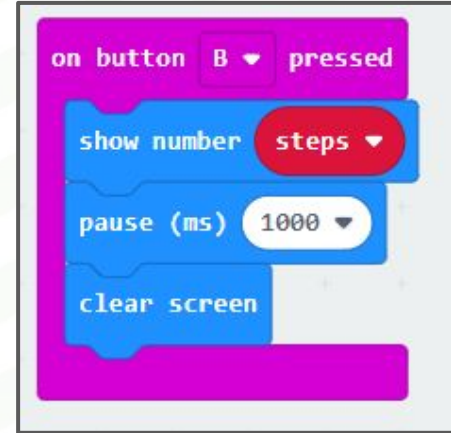
3. *Just as we did before, rather than 0, we want it to show the number of steps. Go to* *Variables* *and select the* *steps* *block and move it inside of the* *show number* *block.*

Displaying steps and clearing the screen...

4. To clear the screen after 1 second, we will add a *pause* block as well as a clear screen block. Go to *Basic*, scroll down and select the *pause* block and, in the *button B* block, place it beneath the *show number* block. In the pull down menu, beside the *100,* choose *1 second* which is *1000 ms*. Finally, go to *Basic,* select the *clear screen* block and add it to your code as shown on the right.

When button B is pressed, the current number of steps will be displayed and the micro:bit LED screen will clear to save the battery. When the user begins taking steps again, the number of steps will continue to display.

Resetting the micro:bit...

1. *Go to Input, select the on button A pressed, drag it onto your work area, and in the pull-down menu, change it to A+B.*

2. *Go to Variables and select the set steps to 0 block and drag it into the on button A+B pressed block.*

3. *From Basic, choose show number and drag it below set steps. Next, from Variables add the steps variable as shown on the right. This will display the current steps, which will be zero, since it was reset.*
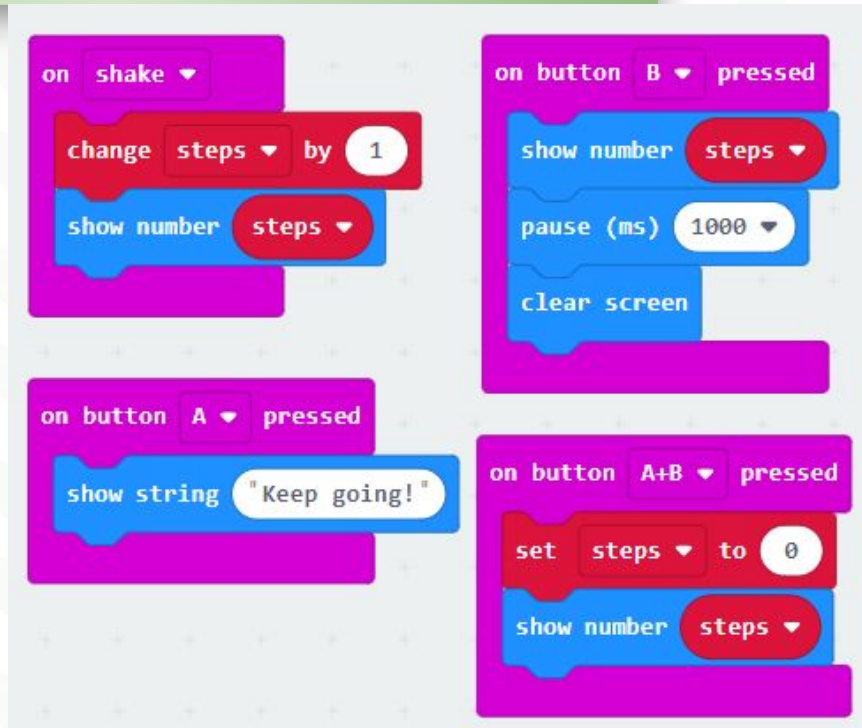

To reset the micro:bit at the end of the day, a student will need to press BOTH buttons A and B. This avoids the micro:bit from resetting if the user presses button A or B by accident.

Double-checking your code...

Make sure your code is complete and accurate! It should look like the blocks below.

Saving your code and transferring it onto the micro:bit

1. **Save your code onto your computer then select *Download.***

🔻 Download   ...

Note: *Your micro:bit's firmware may need to be updated if you are going to use direct flashing to get the code onto the micro:bit.. If it doesn't pair properly, go to the link below to learn how to update the micro:bit's firmware.*
Link to download the latest firmware:
https://microbit.org/get-started/user-guide/firmware/

2. *Usually the micro:bit HEX file will be put in your Downloads folder but you can select another folder to save it in.*

3. *Finally, click and drag this downloaded HEX file onto your micro:bit. If you get stuck, see this link for additional information (see Drag and Drop):*
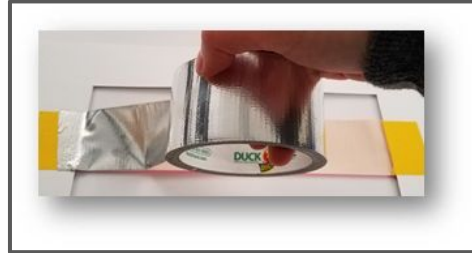*https://microbit.org/get-started/first-steps/set-up/*
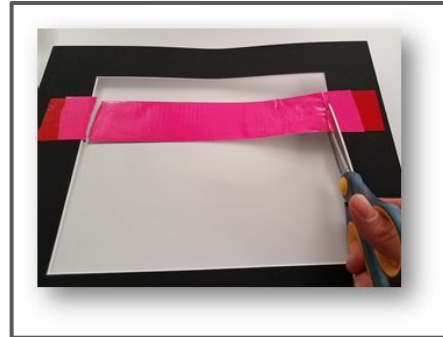*You will also see the Direct Flashing option here, which may require a firmware update.*

Making a Wrist Cuff for the micro:bit...

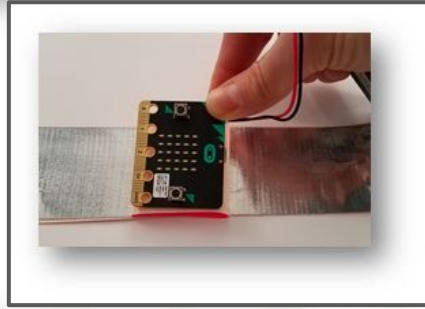1. *Using a frame, cut two strips of duct tape and stick them back-to-back as shown.*
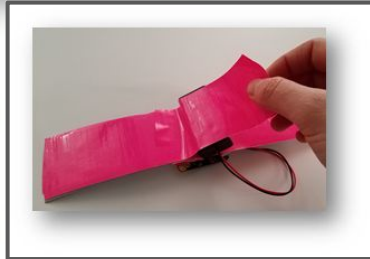
2. *Cut the tape off of the frame.*
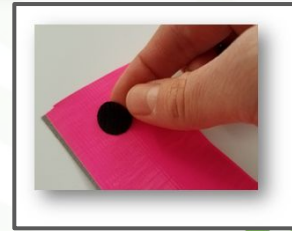
Making a Wrist Cuff for the micro:bit...

3. *Loop a piece of duct tape to stick on your micro:bit.*

5. *Loop a piece of duct tape to stick on your micro:bit.*

4. *Tape the battery pack on the opposite side.*

CODE to LEARN

Credits

Principal Writer........... *Lisa Anne Floyd* **Contributors**.............. **Michael Quinn** **Elena Yakovleva**	Create an Interactive Steps Converter by Code To Learn is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

Partners

LCSI   fair chance learning *what every student deserves*   CISCO   21   Deloitte.

A program of
TakingITGlobal
INSPIRE INFORM INVOLVE

Connected North

With funding from
Canada