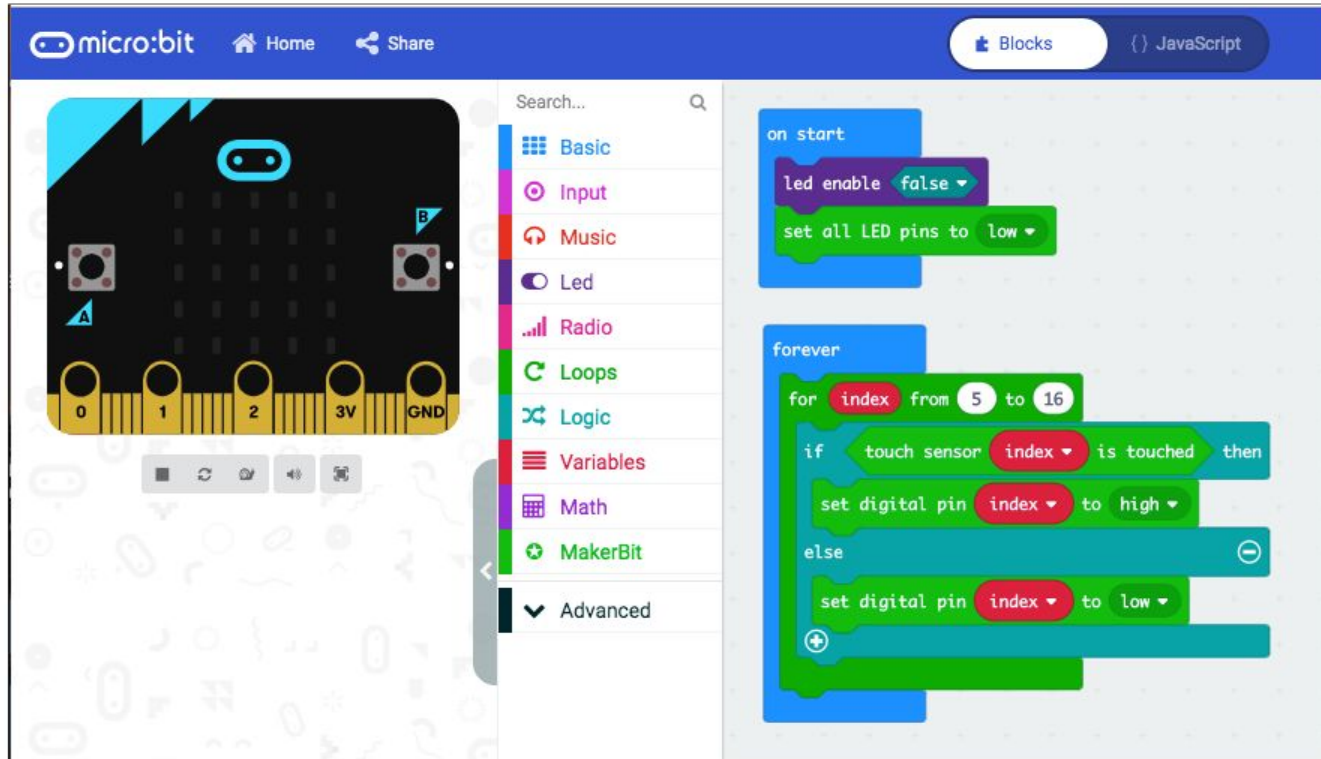


# Activity 4: Introduction to Coding with MakeCode & the MakerBit (20 minutes)

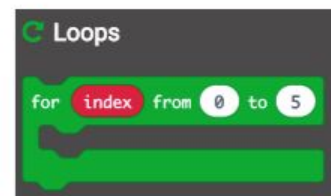
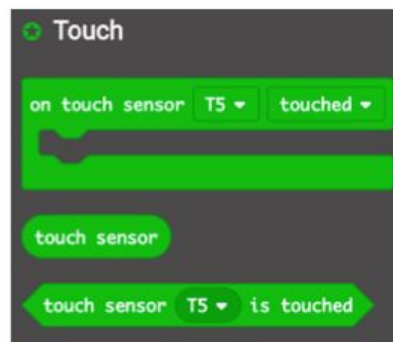
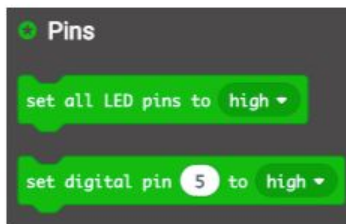


The Google Slides version of this is at: <http://bit.ly/2Nn9Duy>

# The MakerBit + MakeCode

With the MakerBit extension for MakeCode, you can use MakerBit projects as a springboard to coding. By understanding how physical devices in the everyday world are programmed to do what they do, you can add to the variety of physical interactions possible. In this activity, you'll see what the block programming looks like to turn on and off the LEDs when the touch sensors are touched.

Here are the MakerBit blocks for MakeCode that would be used for controlling LEDs and responding to touch inputs:



# Some Getting Started Tips

## Pin Numbers

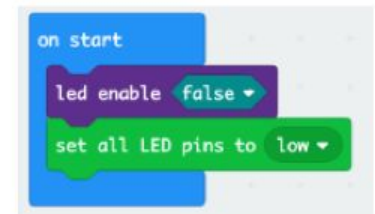
On the volcano model using the MakerBit, the LEDs are connected to pins 5, 6, and 7, and the touch sensors are T5, T6, T7, T8 and T9.

## Transferring Programs to the micro:bit (MakerBit)

To upload programs to the micro:bit, you first download a file to the Downloads folder on your computer, and then drag it from the Downloads folder on to what will appear to be a drive named "MICROBIT" (when the micro:bit is plugged into the USB port of your computer).

## The micro:bit LED and Buttons

The built-in LED display and buttons of the micro:bit use pins 5, 6, 7, 9 & 11 of which you will be using micro:bit pin numbers 5, 6 and 7 for the LEDs in the volcano. There is a programming block to turn off the micro:bit LED display that you should use at the beginning of any program that uses extra LEDs rather than the built-in LED display.



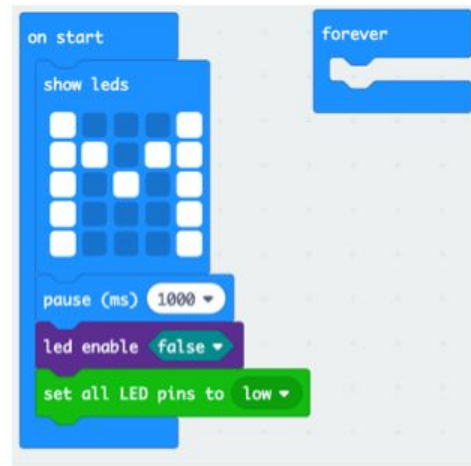
# Open MakeCode

- 1 Use this URL: <https://makecode makerbit.com> to open MakeCode on the computer attached to the MakerBit. (Close the MakerBit Media Linker if it is open.)

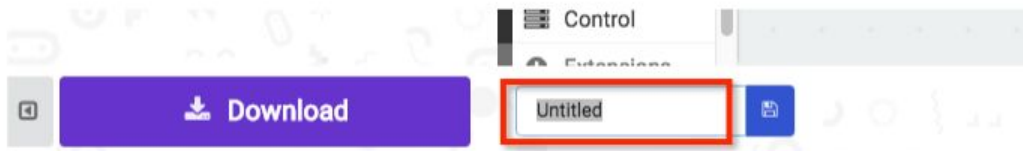
Using the above link, MakeCode will always open with the MakerBit extension already added, and with the "on start" block already set up with "led enable [false]" and "all LED pins [low]"

The on start block holds things that will only be done once in your program.

This tells the micro:bit to turn off the built-in LED display, and also to start with all external (5-16) LEDs off.



- 2 Click on "Untitled" to rename the program to "Blink", or any other name you prefer.



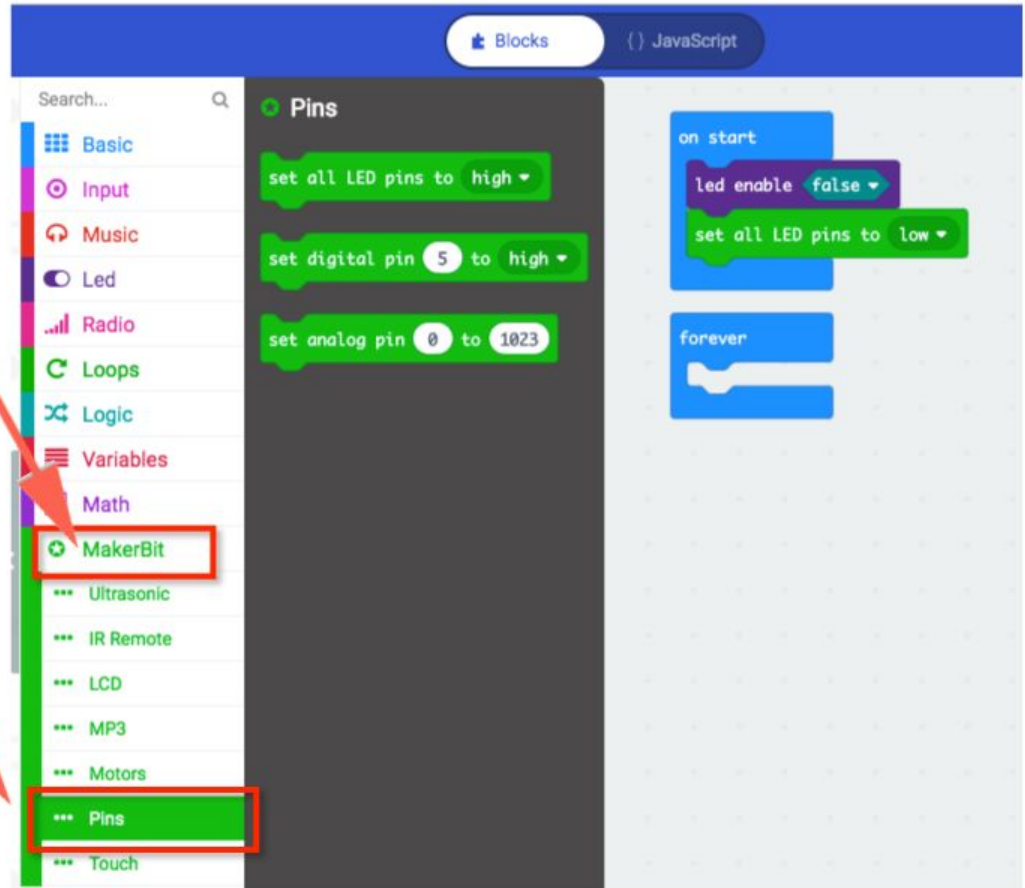
Opening MakeCode with the above link shows an "M" that will be displayed on the microbit LED, followed by a pause.

This is omitted on upcoming pages just to save space on the pages.

# Locate the MakerBit Pin Blocks

1

Click on "MakerBit" to display more blocks, and then click on "Pins".

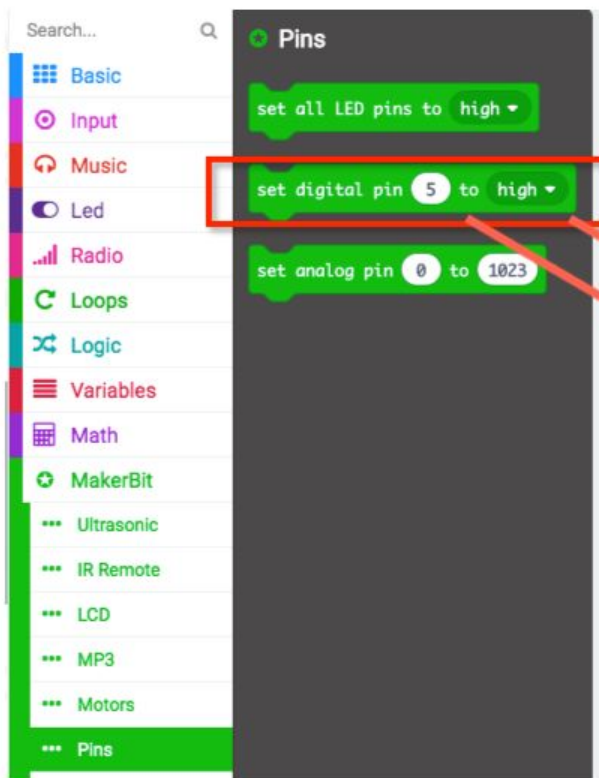


**Note:** If you add the wrong block, you can delete it by dragging it back into the column of categories at the left side of the work area.

# Add LED Control Blocks

- 2 Click on “set digital pin [5] to [high]” to place that block in the programming work area and drag the block into the forever block.

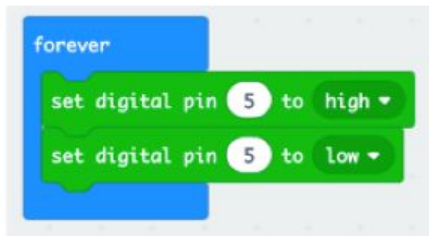
Repeat another time so that you have 2 copies of the block inside of the forever block.



The "forever" block is a loop that repeats over and over. This is where you put the things that you want to be continually happening with the micro:bit.

# Configure the blocks to turn LED 5 on and off

- 1 For the lower block, click on "high" and edit it to be "low".



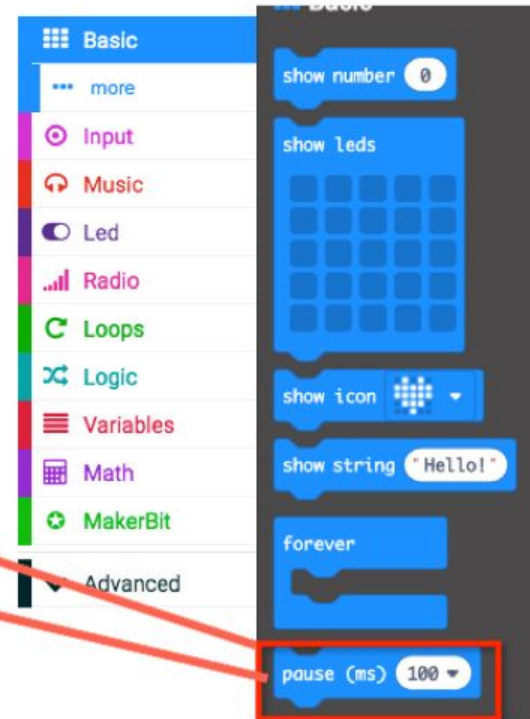
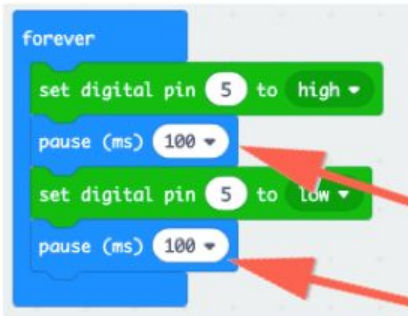
In digital circuits, "high" is "on", and "low" is "off". These can also be represented as "1" and "0".

Unique & special to the MakerBit: Only the MakerBit pin blocks let you edit the pin position to use a number or a variable.

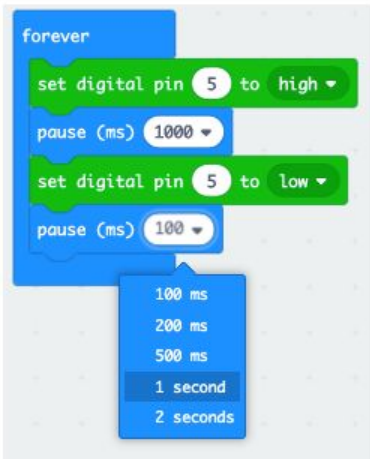
This isn't possible in MakeCode unless you've added the MakerBit extension.

# Add a Delay to Make a Blink

- 2 Click on the “Basic” category, and click on “pause” twice to add two blocks to your work area, and drag them into position after each of the “set digital pin” blocks.



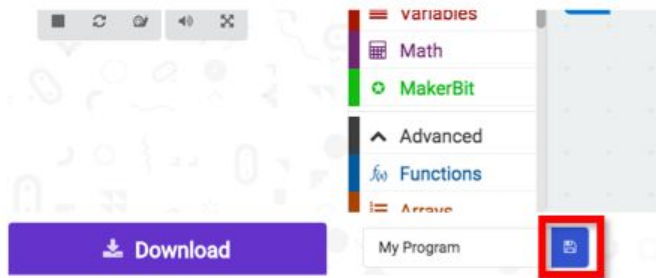
- 3 Click on "100" in each block to edit it to "1 second". 1 second will be a 1000 ms (millisecond) pause.



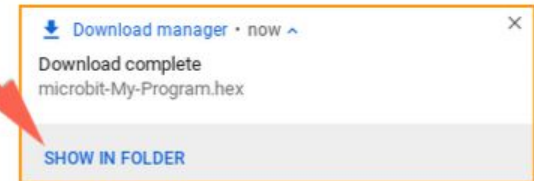
# Download the program

**1** Click on the "disk" (for downloading) button on the MakeCode editor screen.

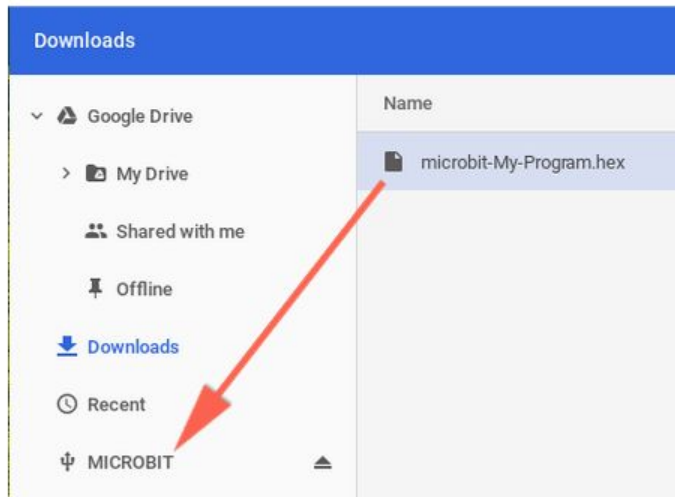
There will be a momentary pause, and then a ".hex" file of your program will be downloaded to your Downloads folder.



Then click here to see your downloaded file.



**2** With the USB cable and MakerBit connected to the computer, drag the "microbit-My-Program.hex" file to the "drive" named "MICROBIT".



On a Chromebook, you'll see the message above, and you can click on "SHOW IN FOLDER" to quickly get to the file.

On Mac and Windows, you'll need to navigate to your Downloads folder.

If you're having trouble finding the micro:bit device, use this link for more help: <http://bit.ly/2MKhnGs>

# Try It Out! Add More!

- 1 After dragging the program file (.hex) to the micro:bit, there will be a momentary pause, and then the LED of your model should start blinking!

## Programming Challenges

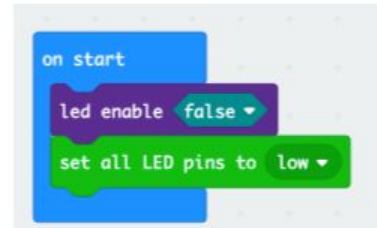
(adding touch response is on the next page)

Can you change the delay values to create a faster blink?

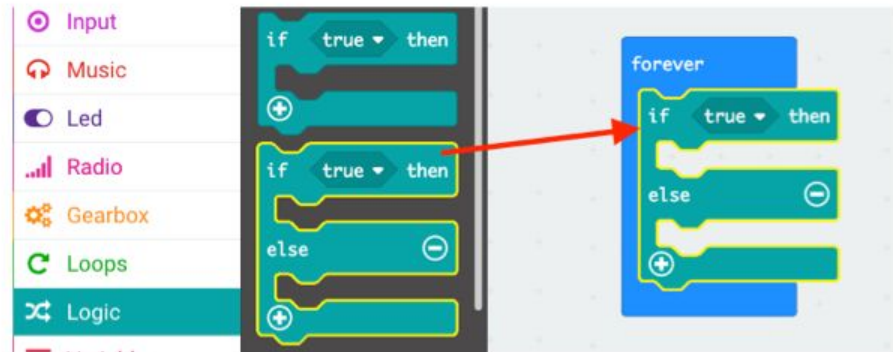
Can you add additional groups of blocks to also blink LEDs 6 and 7?

# Touch Interactivity

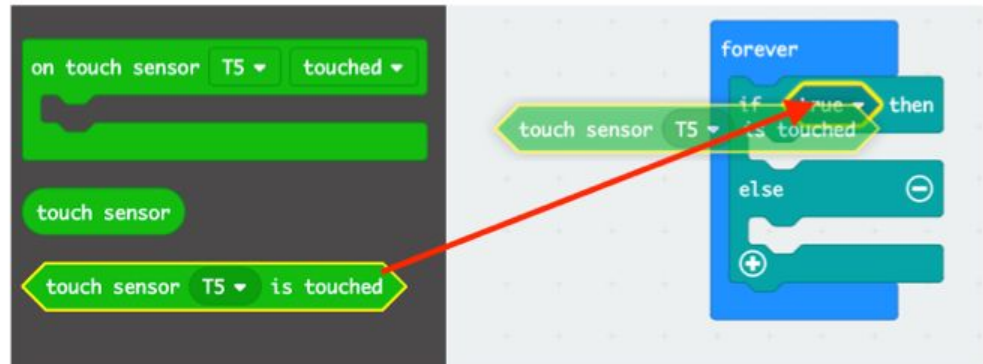
- 1 Start a New Project that is automatically MakerBit-ready by typing in this URL:  
<https://makecode.makerbit.com> to open MakeCode with MakerBit blocks already available.



- 2 In the "Logic" category is "If / then / else". Drag that into the forever block.



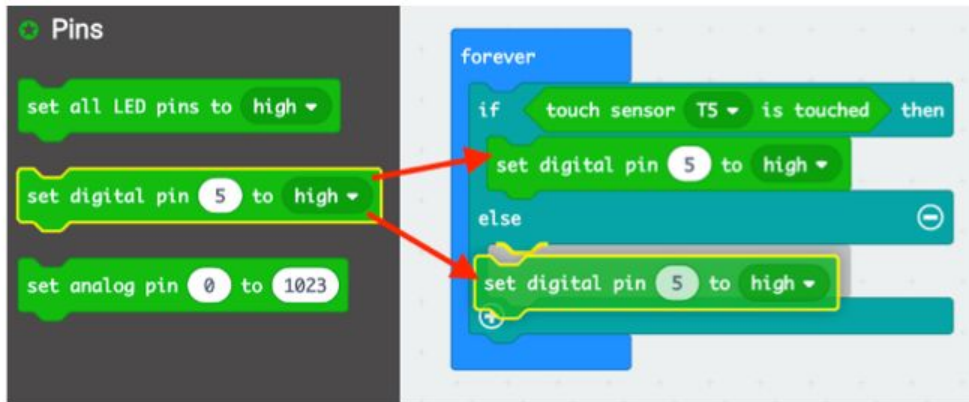
- 3 In the MakerBit touch blocks, find "touch sensor [T5] is touched", and drag that to <true> in the if-then-else block.



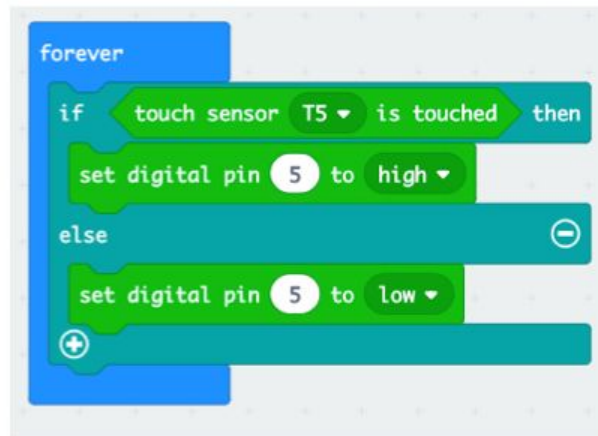
# Touch Interactivity

4

Drag the set digital pin (5) blocks into the spaces of the if-then-else block.



And edit the 2nd set digital pin (5) block to "low".



# Downloading & Testing

- 5 Use the disk icon to download and put your program on the micro:bit and test your model.

Touch sensor 5 ("main vent" on the volcano model) will turn on LED 5.

## Coding challenges:

Can you combine what you've learned earlier to make LED 5 blink while the sensor is touched?

Edit your existing program to create the one on the right.

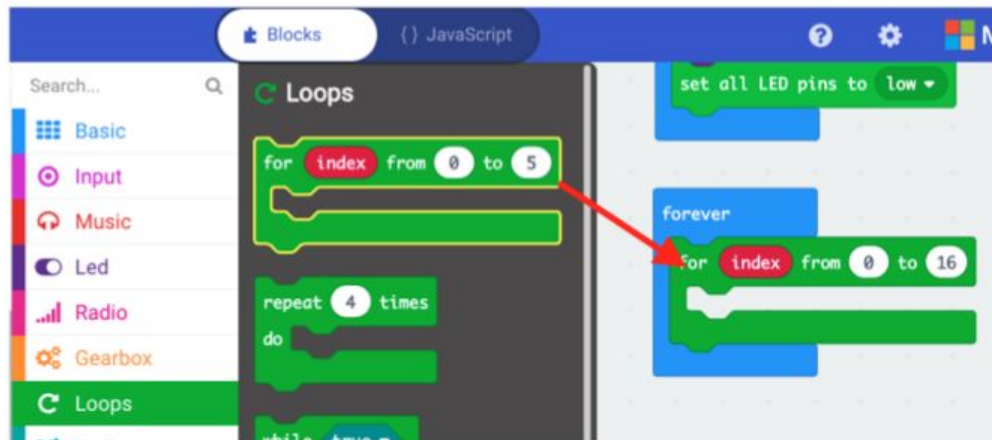
Can you add to the code so that the other 2 sensors, #6 and #7, ("secondary vent" and "the crater") on the volcano model turn on the corresponding LEDs?



```
forever
  if touch sensor T5 is touched then
    set digital pin 5 to high
    pause (ms) 500
    set digital pin 5 to low
    pause (ms) 500
  else
    set digital pin 5 to low
```

# Loops & Variables

- 1** To control all 12 LEDs with touch sensors, you could copy and paste lots of blocks, but there's an easier way: Tell the micro:bit (the actual controller of the MakerBit) to repeat the same process 12 times while using what's called a "variable" so that other parts of your program know which touch sensor and LED it is currently looking at.
- 2** Start a new program with Look in the Loops category and drag the top block that says "for index from 0 to 5" into the forever block.



Unique & special to the MakerBit: Only the MakerBit blocks have a for-from-to block that allows you to set both the beginning AND ending values of the loop.

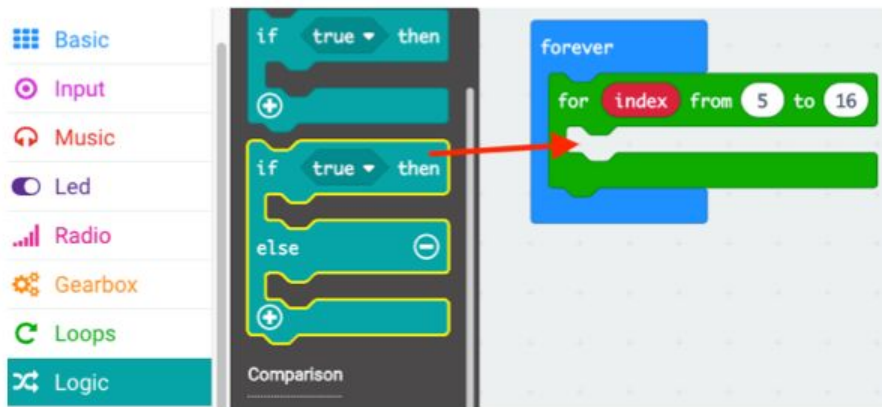
This isn't possible in MakeCode unless you've added the MakerBit extension.

# Loops & Variables

3 Edit the "0" to "5" in the for-index block to be "5" to "16"



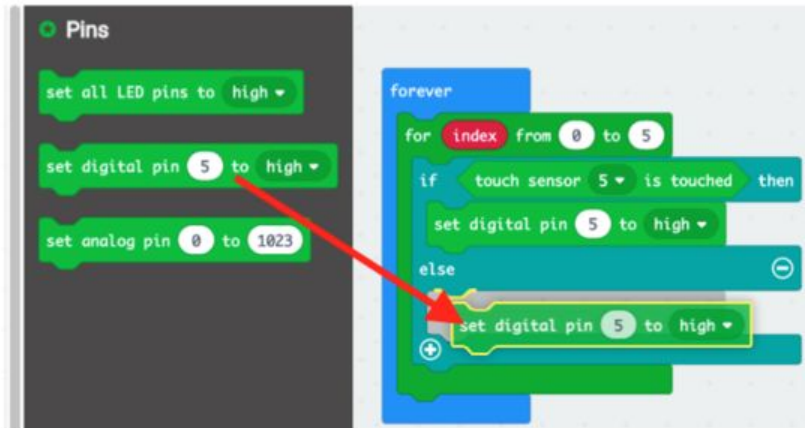
4 From the Logic category, drag the if-then-else block into the for-from-to block.



5 From the MakerBit touch category, drag the "touch sensor [T5] is touched"

# Loops & Variables

- 5 From the pins category of MakerBit, drag the "set digital pin" blocks into the two spaces of the for-from-to block.

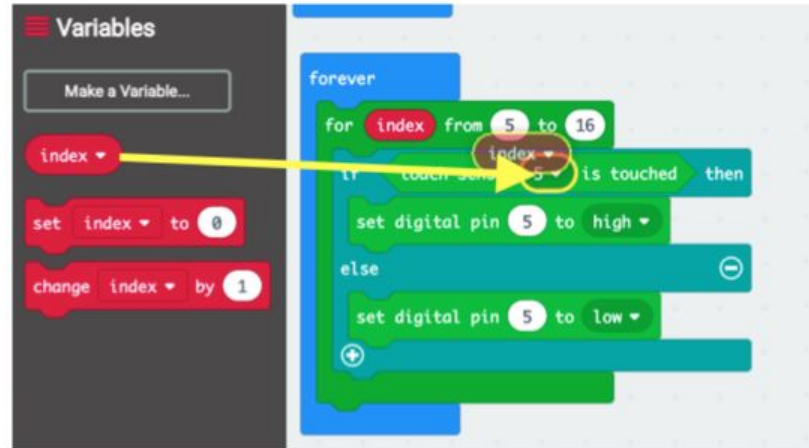


- 6 Edit the second "set digital pin block" from "high" to "low".



# Loops & Variables

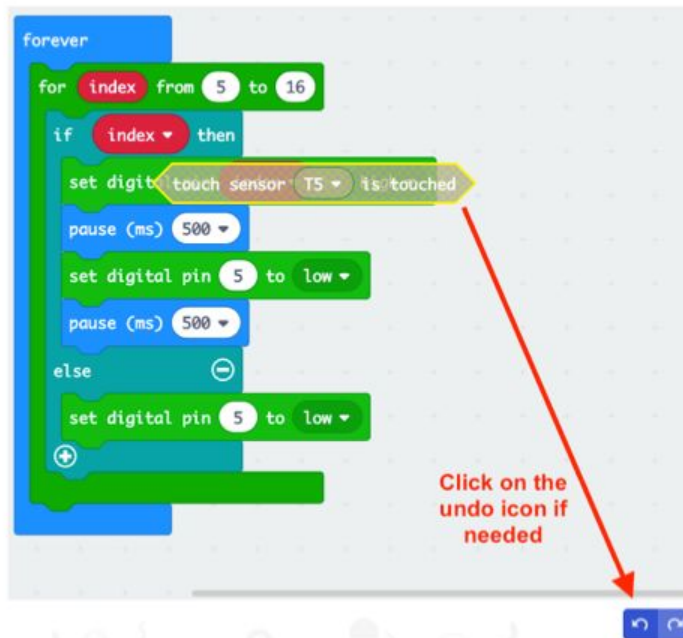
- 7 Click on the variables category, and drag the variable named "index" into each position where it has "5". Take care that only the "5" highlights in yellow when you drop the variable into position.



- 8 To replace "T5", place the right-hand side of the "index" variable oval over "T5" in the block.

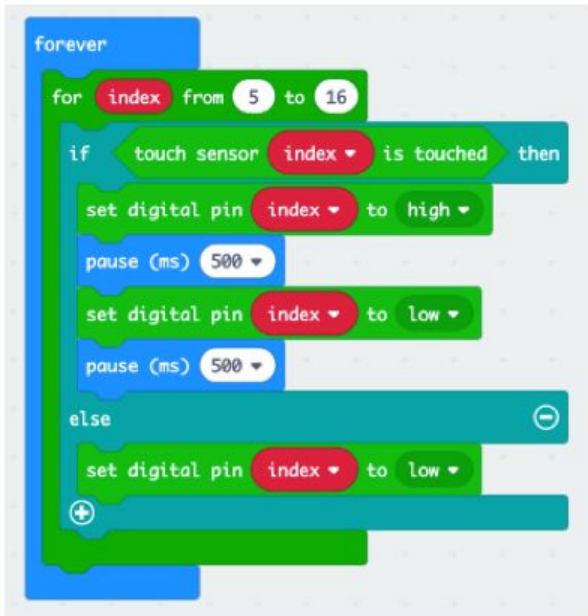
If it accidentally replaces a larger part of the block, click on the "undo" icon at the lower-right of the window.

<http://bit.ly/31Pf5c3>



# Loops & Variables

9 The completed program should look like this:



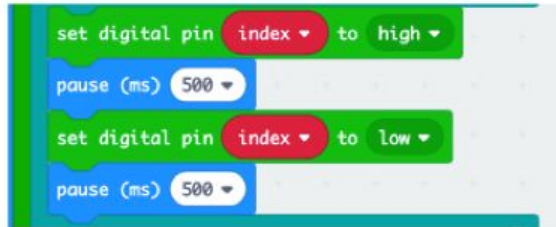
```
forever
  for index from 5 to 16
    if touch sensor index is touched then
      set digital pin index to high
      pause (ms) 500
      set digital pin index to low
      pause (ms) 500
    else
      set digital pin index to low
```

The image shows a Scratch-style code editor with a light gray background. The code is written in a block-based language. It starts with a blue 'forever' loop block. Inside the loop is a green 'for' loop block with 'index' as the variable, ranging from 5 to 16. Within the 'for' loop, there is a teal 'if' block: 'if touch sensor index is touched then'. The 'then' branch contains four blocks: a green 'set digital pin index to high' block, a blue 'pause (ms) 500' block, a green 'set digital pin index to low' block, and another blue 'pause (ms) 500' block. The 'else' branch contains a green 'set digital pin index to low' block. There are plus and minus icons on the 'if' and 'else' blocks, and a plus icon on the 'for' block.

10 Use the disk icon button to download and put your program onto the micro:bit and test your model.

# Programming Challenges

The total time for the blink cycle is the sum of the times for the two pause blocks.



In this example, the total is  $500 + 500$  milliseconds = 1000 milliseconds = 1 second.

If you change just one of the numbers, for example, the first "500" to "1000", the total time between blinks will now be 1500 ms. If you keep it so that BOTH pause times always add up to 1000ms, for example,  $700 + 300$ , then the blink rate will be the same, but the amount of time that the LED is on or off is changed.

Try times of 900ms right after the on ("high") block, and 100ms for "low". What do you observe. How about the opposite?

What are the smallest values that you can use where you can still observe a blinking effect? Can you make the on and off so fast that the LED is just "dim", but not visibly blinking?

# Going Further with Coding

## Computer Science Through Everyday Devices

**Congratulations! You have just coded some physical interactivity in the context of a curriculum-based project!**

**With what you have done in just this activity, and if more time has been allocated to group discussion and reflection, over two dozen Computer Science Standards as listed in this spreadsheet could have been covered: <http://bit.ly/2YLqPKy>**

**In addition to curriculum-centered uses of coding, there is the opportunity to further explore computer science objectives by creating working models of everyday devices that are computationally enabled, from traffic lights and thermostats to automatic door openers and self-driving cars. With the MakerBit and MakeCode, students can get a first-hand understanding of the everyday "smart" objects in their world and make computer science more relevant and less abstract.**

**This Google slides document shows some examples: <http://bit.ly/2WIMCY5>**

**You can find more MakerBit activities on Google's Workbench education platform: <http://bit.ly/MakerBitWorkbench>**