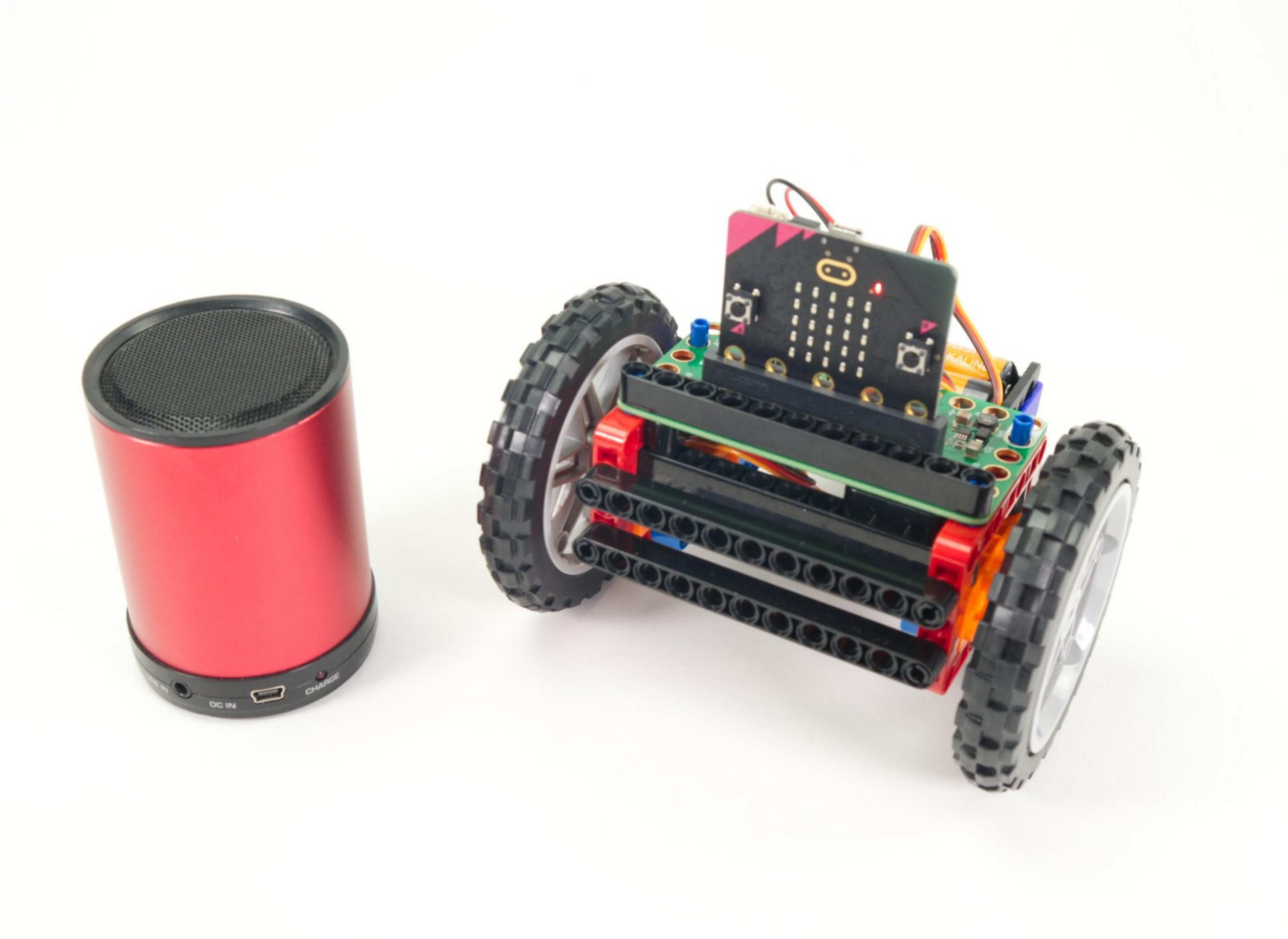
Rover - Sound Activated

Make your Rover move when it hears a sound! Get the whole class to applaud and watch the Rover roll away.

Written By: Pete Prodoehl



This document was generated on 2023-08-16 04:03:13 PM (MST).

INTRODUCTION

The Bit Board Rover can take advantage of the built-in sensors found on the micro:bit, and we can use microphone to "remotely control" the Rover so it moves when it detects sound.

You might also want to check out our Rover - Light Activated guide which lets you control your Rover with a flashlight.

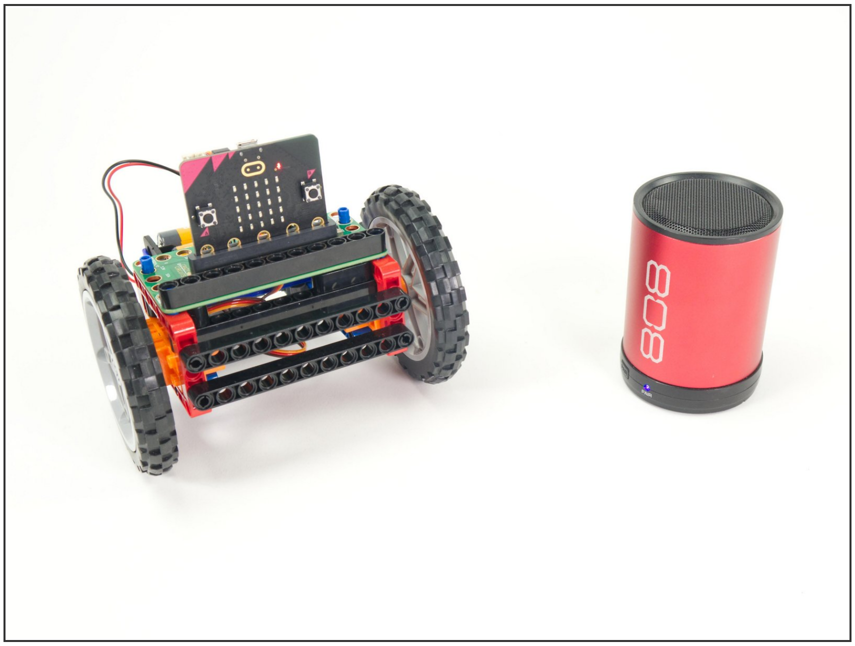
TOOLS:

- Computer (1)

PARTS:

- Bit Board Rover Kit (1)

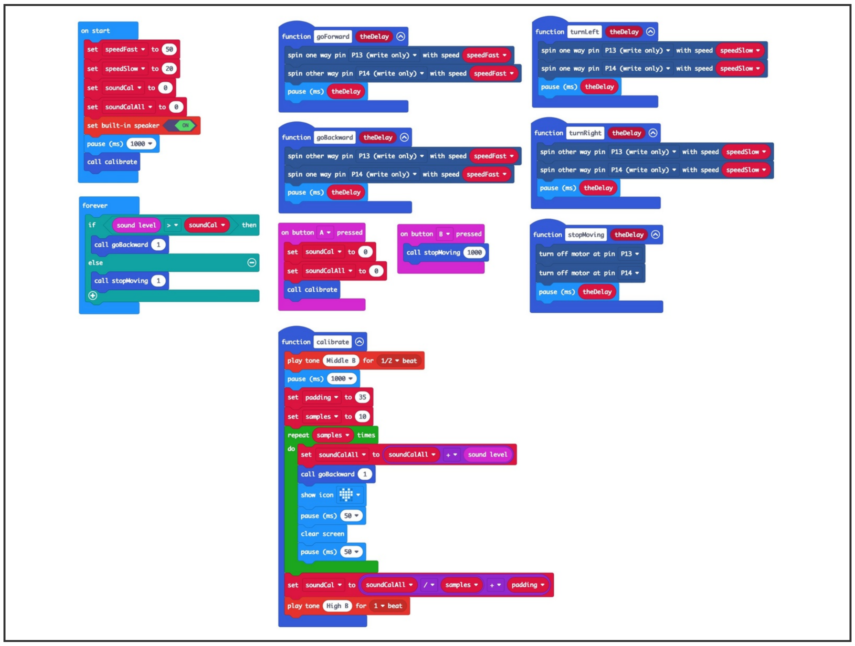
Step 1 — Prepare Your Rover



- For this guide you'll need a completed Rover Main Body.
- You'll also need something to make sound! It could be your own voice, or a classroom full of clapping students.
- You can also use a speaker playing a sound. (A Bluetooth speaker connected to a mobile phone or tablet works great.)

Our Sensor Showcase covers using the sensors built in to the micro:bit. The microphone is covered in Step 11.

Step 2 — Load the Code



```
on start
  set speedFast to 50
  set speedSlow to 20
  set soundCal to 0
  set soundCalAll to 0
  set built-in speaker ON
  pause (ms) 1000
  call calibrate

forever
  if sound level > soundCal then
    call goBackward 1
  else
    call stopMoving 1

function goForward theDelay
  spin one way pin P13 (write only) with speed speedFast
  spin other way pin P14 (write only) with speed speedFast
  pause (ms) theDelay

function goBackward theDelay
  spin other way pin P13 (write only) with speed speedFast
  spin one way pin P14 (write only) with speed speedFast
  pause (ms) theDelay

on button A pressed
  set soundCal to 0
  set soundCalAll to 0
  call calibrate

on button B pressed
  call stopMoving 1000

function calibrate
  play tone Middle B for 1/2 beat
  pause (ms) 1000
  set padding to 35
  set samples to 10
  repeat samples times
  do
    set soundCalAll to soundCalAll + sound level
    call goBackward 1
    show icon
    pause (ms) 50
    clear screen
    pause (ms) 50
  set soundCal to soundCalAll / samples + padding
  play tone High B for 1 beat

function turnLeft theDelay
  spin one way pin P13 (write only) with speed speedSlow
  spin one way pin P14 (write only) with speed speedSlow
  pause (ms) theDelay

function turnRight theDelay
  spin other way pin P13 (write only) with speed speedSlow
  spin other way pin P14 (write only) with speed speedSlow
  pause (ms) theDelay

function stopMoving theDelay
  turn off motor at pin P13
  turn off motor at pin P14
  pause (ms) theDelay
```

⚠ If you've never used a micro:bit before you'll want to check out this guide: Bit Board V2 Setup and Use

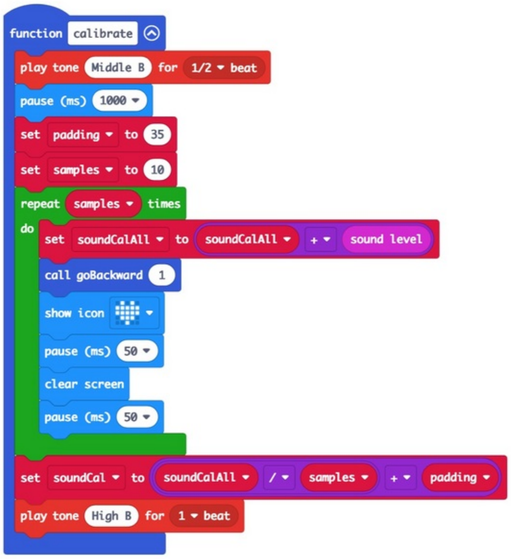
- We're going to load the following code for our **Rover Sound Detecting Simple** program: https://makecode.microbit.org/_6V8DirKRE...

- Note: This version of the code is a great starting point for this project. If you want to take it further examine the code for the Rover - Light Detecting Turner and consider using it for a starting point.

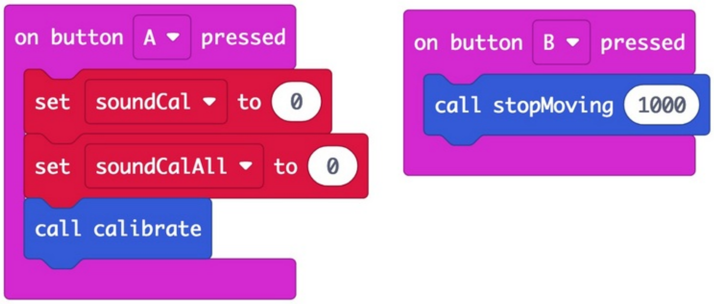
- When you power on the Rover **it will start moving** *while* it runs a calibration routine to check for sound levels. (We'll cover the calibration in **Step 3**.)

- Once the calibration is done the Rover will sit there waiting for the sound levels to go above the threshold that was set in the calibration routine...

Step 3 — Calibration



```
function calibrate
    play tone Middle B for 1/2 beat
    pause (ms) 1000
    set padding to 35
    set samples to 10
    repeat samples times
    do
        set soundCalAll to soundCalAll + sound level
        call goBackward 1
        show icon
        pause (ms) 50
        clear screen
        pause (ms) 50
    set soundCal to soundCalAll / samples + padding
    play tone High B for 1 beat
```



```
on button A pressed
    set soundCal to 0
    set soundCalAll to 0
    call calibrate

on button B pressed
    call stopMoving 1000
```

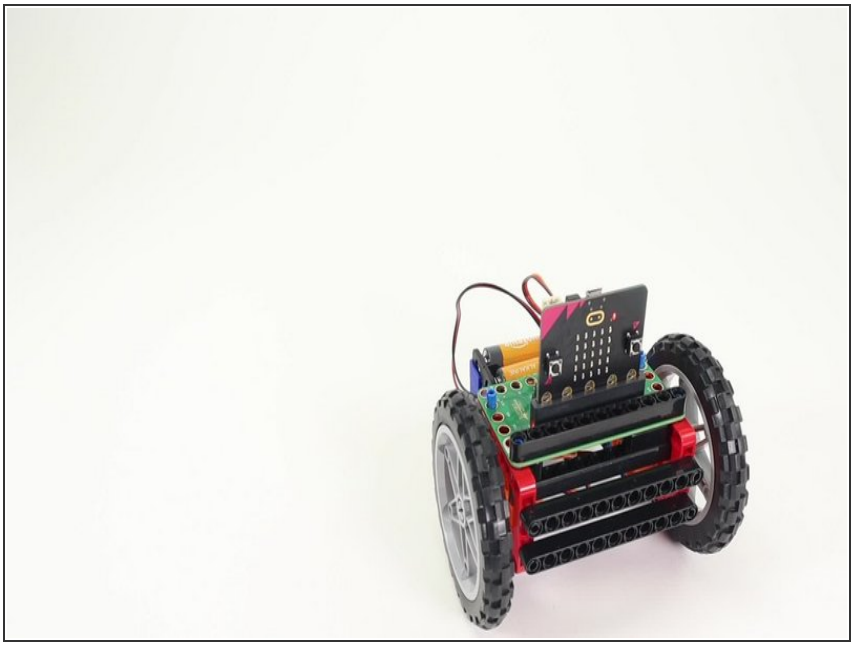
⚠ Make sure to see the **note** at the bottom of this step.

- We need to calibrate the sound levels in the room before the micro:bit can tell if there is a sound.
- The sound level can be anywhere between **0 and 255**. (0 is very quiet and 255 is very loud.)
- Our code takes 10 readings (with a slight pause between each reading) adds them all together and the divides by the number of samples (10 in this case) to get our final value.
    - Calibration routines often use this **sampling** technique where a number of values are captured and then the *average* of them is used.
- Once we've calculated our ambient sound level we pad the number a bit to prevent false triggering. We used **35** for our pad value in this example but you can experiment with lower (or higher) values.

The calibration runs automatically when you power on the Rover but you can also run it by pressing the **A** Button on the micro:bit if you need to recalibrate.

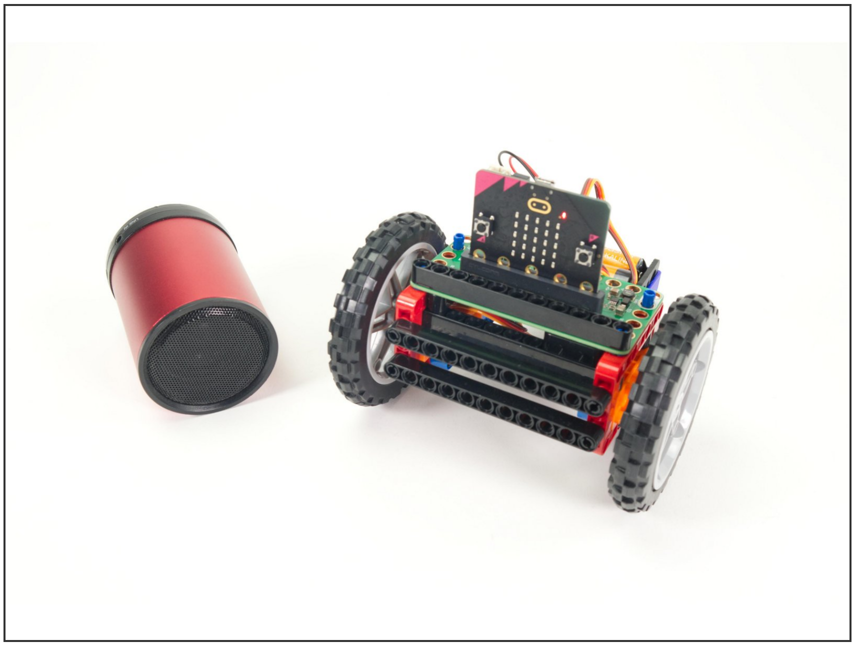
⚠ **Note:** The Rover needs to move (or at least spin the wheels) when it calibrates! This is because **the motors make noise**, and we need to take that sound into account when doing the calibration.

Step 4 — Test it Out!



- Power on the Rover, and either set it down so it can roll, or hold it in your hand so the wheels can spin freely.
- Do your best to be quiet and wait for the calibration to complete. Once the heart stops flashing on the micro:bit and you hear a second beep, the calibration should be done.
- The Rover should be still when the room is quiet, so... Make some noise! Clap, yell, laugh... See if you can get the Rover moving.
- You can make the Rover stop by being quiet. (Shhh!)

Step 5 — Take it Further



- The code provided should serve as a starting point for your own ideas about how the Rover can react to sound.
- You could program a sequence of movements, or with new code even add an accessory (like the Gripper, Lifter, or Sweeper) and then try to control them by making sounds.

This document was generated on 2023-08-16 04:03:13 PM (MST).

This document was generated on 2023-08-16 04:03:13 PM (MST).