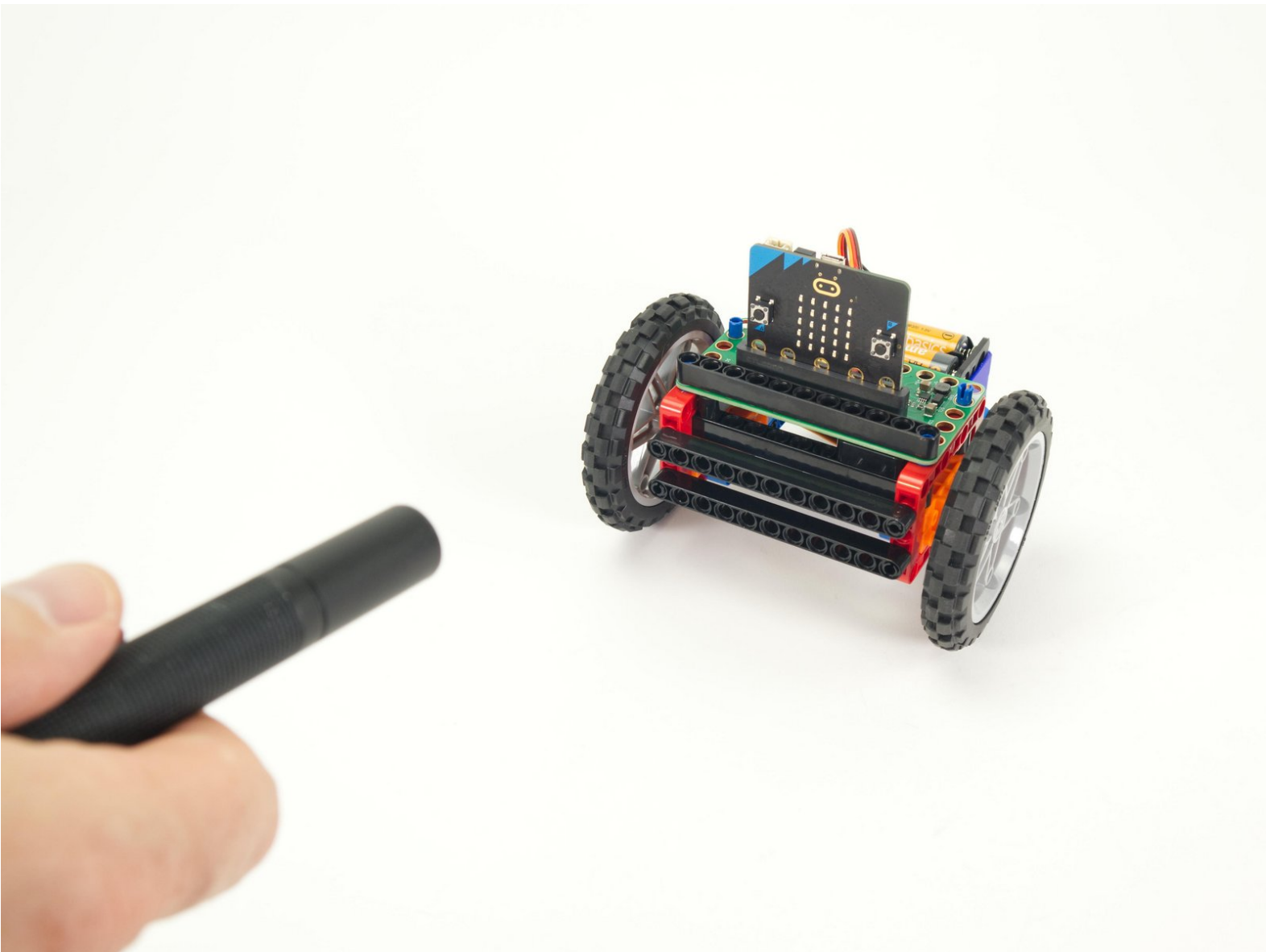




# Rover - Light Activated

Remotely control your Rover with a flashlight! Use the built-in light sensor on the micro:bit to make your Rover move around.

Written By: Pete Prodoehl



## INTRODUCTION

The Bit Board Rover can take advantage of the built-in sensors found on the micro:bit, and we can use the light sensor to "remotely control" the Rover with a flashlight.

We'll start with simple code to make the Rover roll away from you when you shine a flashlight at it, and then use more complex code to get the Rover turning as well as rolling away.



### TOOLS:

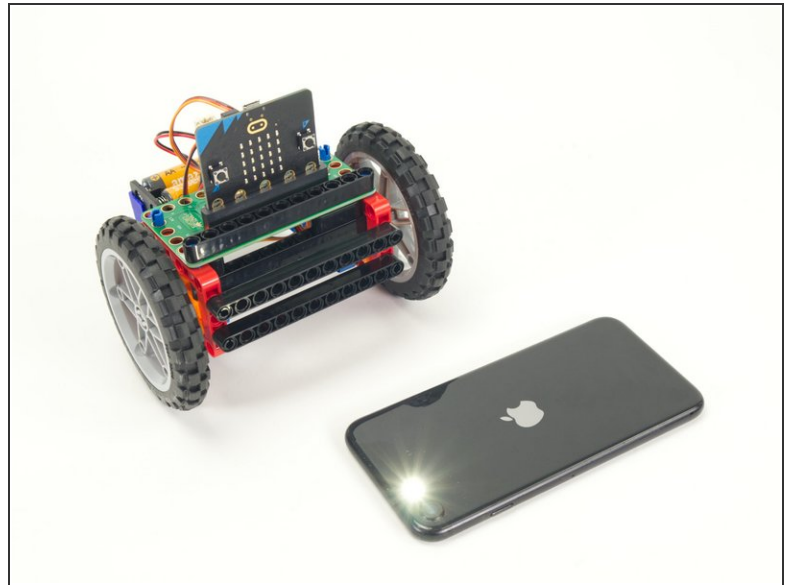
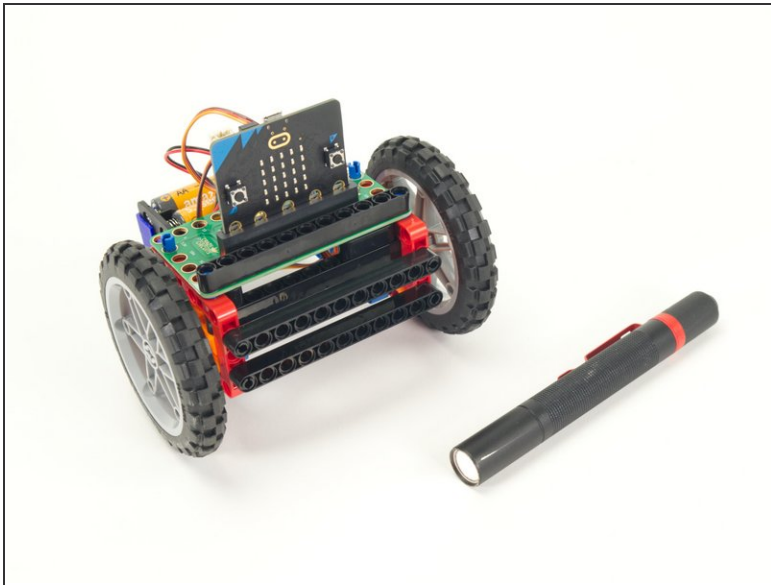
- [Computer](#) (1)



### PARTS:

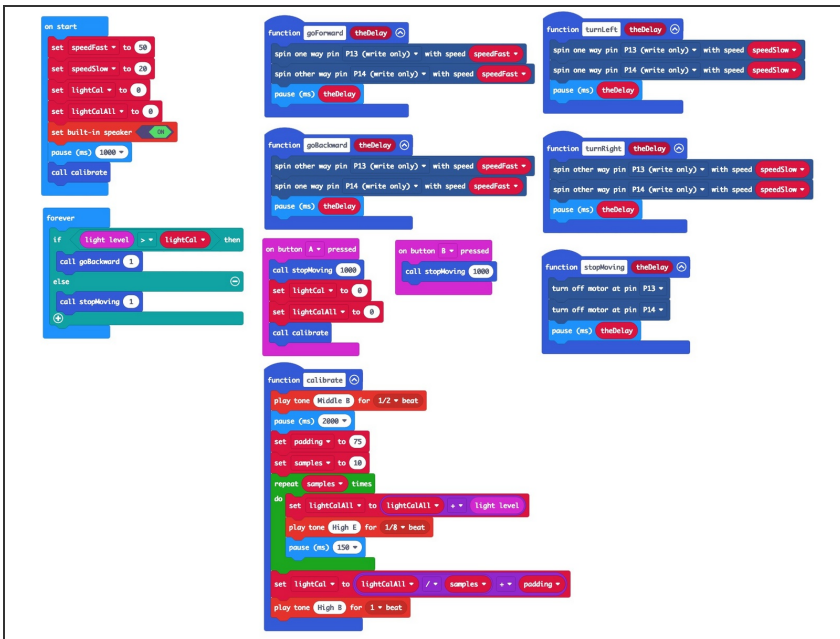
- [Bit Board Rover Kit](#) (1)
- [Flashlight](#) (1)

## Step 1 — Prepare Your Rover



- For this guide you'll need a completed [Rover Main Body](#).
  - You'll also need a flashlight you can shine at the Rover to make it move.
  - A nice powerful/bright flashlight is preferable, but you can use the "flashlight" feature of a phone in a pinch. You'll just need to get closer.
- ☒ Our [Sensor Showcase](#) covers using the sensors built in to the micro:bit. The light sensor is covered in [Step 10](#).

## Step 2 — Load the Code



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

- Note: This **Simple** version of the code is a great starting point for this project, but the **Turner** version in **Step 5** is more fun. This code is easier to understand and the Turner code is a bit more complex.
- We're going to load the following code for our **Rover Light Detecting Simple** program: [https://makecode.microbit.org/\\_Am6D3o5b1...](https://makecode.microbit.org/_Am6D3o5b1...)
- When you power on the Rover it will run a calibration routine to check the light levels in the room. (We'll cover the calibration in **Step 3**.)
- Once the calibration is done the Rover will sit there waiting for the light levels to go above the threshold that was set in the calibration routine...
- When you shine a flashlight on the face of the micro:bit it will detect the higher light level and the Rover will start rolling away from you!

## Step 3 — Calibration

```

function calibrate
  play tone Middle B for 1/2 beat
  pause (ms) 2000
  set padding to 50
  set samples to 10
  repeat samples times
    do
      set lightCalAll to lightCalAll + light level
      play tone High E for 1/8 beat
      pause (ms) 150
  set lightCal to lightCalAll / samples + padding
  play tone High B for 1 beat
  set runState to 1
  
```

```

on button A pressed
  call stopMoving 1000
  set lightCal to 0
  set lightCalAll to 0
  call calibrate

on button B pressed
  call stopMoving 1000
  
```

- We need to calibrate the light levels in the room before the micro:bit can tell if a light is shining on it.
- The light level can be anywhere between **0 and 255**. (0 is almost no light and 255 is very bright light.)
- Our code takes 10 readings (with a slight pause between each reading) adds them all together and then 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 light level we pad the number a bit to prevent false triggering. We used **75** 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.

## Step 4 — Test it Out!



- Power on the Rover, wait for the calibration to complete, and then shine a flashlight at the micro:bit on the Rover.
- The Rover should roll away from the light!
- You can make the Rover stop by moving the flashlight away so it is not shining on the micro:bit
- Alternately the Rover will stop rolling when it gets too far away and the light level drops due to the further distance of the flashlight.

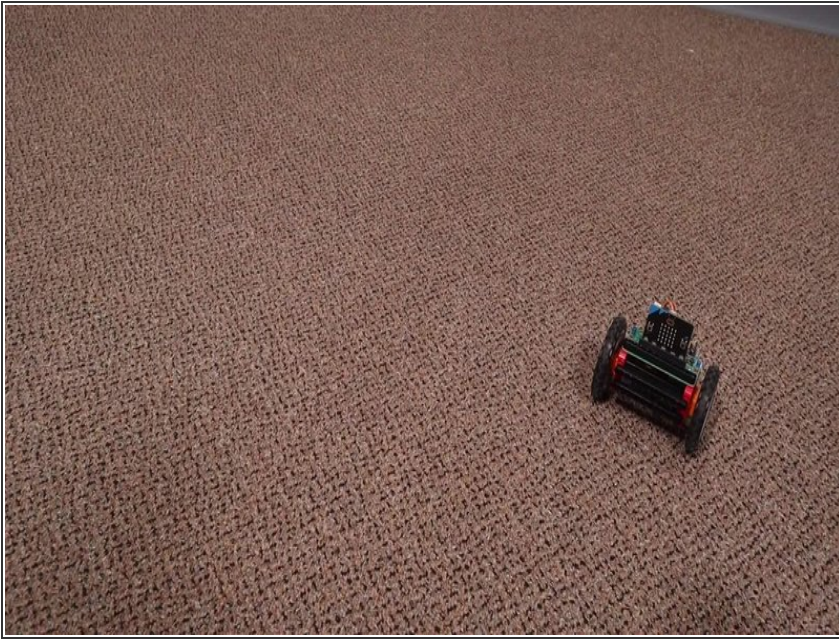


## Step 5 — Load New Code



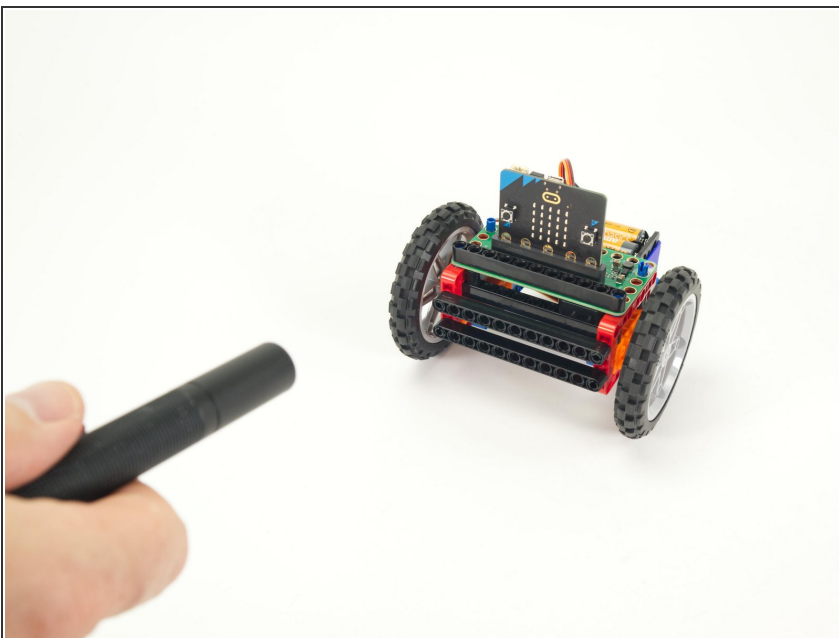
- This **Turner** version of the code adds a few features, the most important being that the Rover will no longer sit still.
- When the Rover does not detect the flashlight it will turn in a circle. You can wait for the Rover to turn a certain amount and then shine the flashlight on it to get it rolling again.
- Load the following code for our **Rover Light Detecting Turner** program:  
[https://makecode.microbit.org/\\_Dt22XJgFq...](https://makecode.microbit.org/_Dt22XJgFq...)
- We've added a few neat things to the code.
  - The Rover may turn clockwise or anti-clockwise when not rolling. There's a function in the code to randomly choose which way to turn.
  - We've also added a runState variable that allows you to start/stop the Rover from moving by using the **B** button on the micro:bit to toggle the state. One press starts the Rover, the next press stops it.

## Step 6 — Test it Out!



- Power on the Rover, wait for the calibration to complete, and then shine a flashlight at the micro:bit on the Rover.
- The Rover should start spinning around and then roll away from the light when you shine your flashlight on the micro:bit
- When you move your flashlight away from the Rover it will stop rolling away and start to spin again.

## Step 7 — Take it Further



- The code provided (both the Simple and Turner versions) can serve as a starting point for your own ideas



---

about how the Rover can react to light.

- 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 with a flashlight.
-