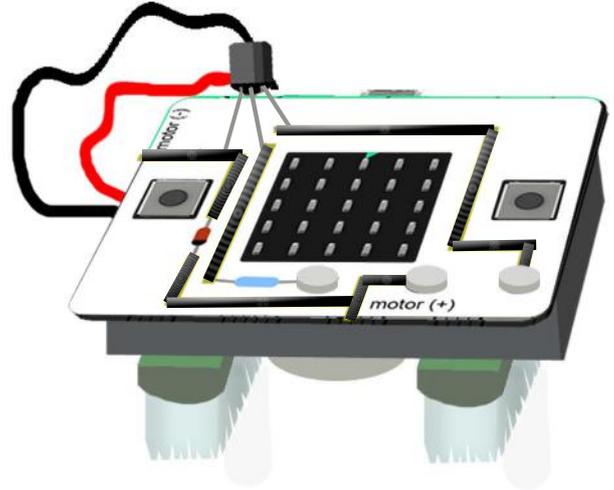


## Make a micro:bot

### Step 1: Gather the materials

#### What you'll need:

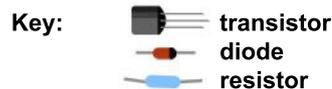
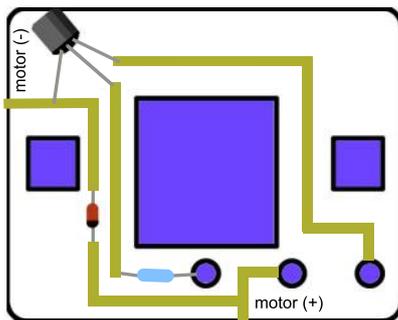
- 1 x 1N4148 diode
- 1 x 1.5 - 3V vibration motor
- 1 x BC547C low power NPN transistor
- 1 x 1k $\Omega$  resistor
- 2 x toothbrushes – heads removed with scissors or pliers
- 1 x BBC micro:bit
- 1 x 2AAA battery pack (for the micro:bit)
- 2 x AAA 1.5V batteries
- 3 x M3 machine screws and nuts
- Single core insulated wire
- Crocodile clips
- Velcro strips or tack



### Step 2: Assemble the circuit and micro:bot

First, print out this worksheet on a piece of paper or card and cut out the diagram below.

You're going to mount this on your micro:bit, so do be careful while cutting around the edges. Cut out the bits marked in **bright purple** - you won't need these.



Place the printed diagram onto the micro:bit, lining up the slots for the LEDs, buttons and the holes for I/O pin 2, 3V and GND. This will form a guide to help you create your circuit.

Carefully place the transistor, diode and resistor in place according to the diagram. Make sure your diode is facing the right way and the transistor is flat side up.

Connect up the circuit as per the diagram. The gold lines indicate where connections between the components and the micro:bit have to be made. Use the insulated wire to form rails and make the connections to the other components using crocodile clips.

Take the positive wire from the motor and slot it into the I/O pin hole that says motor (+). Take the negative wire from the motor and connect it to the leg of the transistor that says motor (-), in parallel with the diode.

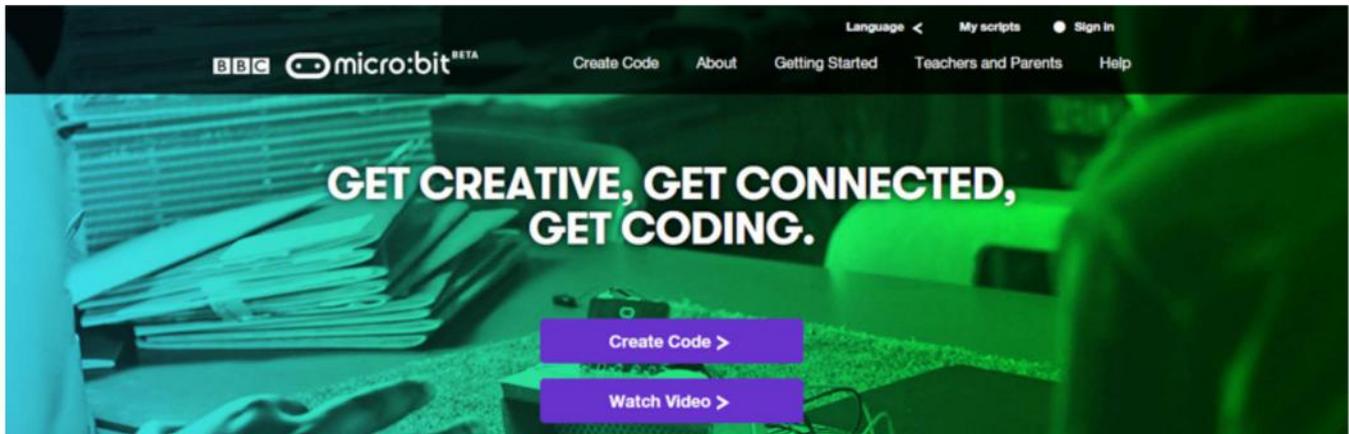
Where the rails lead to the I/O pins, you can thread the wires and components into the holes of the I/O pins and secure them with the screws and nuts to help form the connections and fix the components to your micro:bit.

Using tack, place the motor and toothbrush heads on the bottom of your already attached battery pack. Your circuit should now be complete, and your micro:bot is fully built.

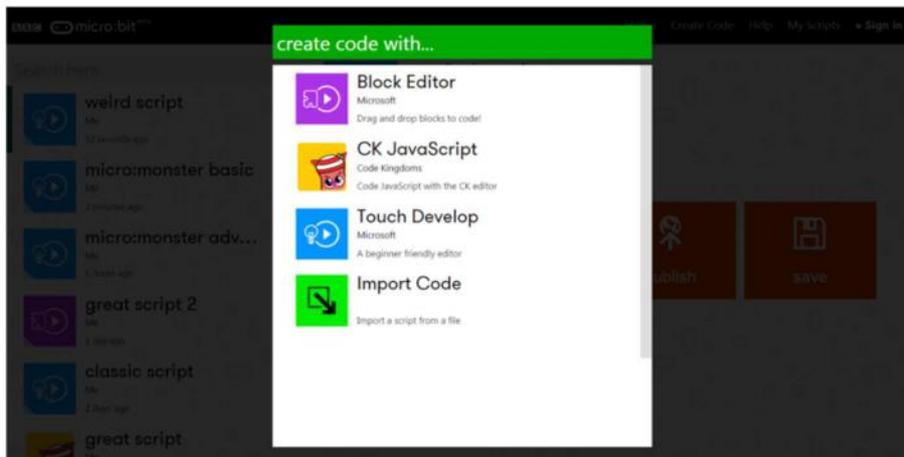
## Step 3: Import the code

Download the hex file from our Live Lessons website by clicking on the **micro:bot hex file** link.

Firstly, select 'My scripts' on the top navigation on the micro:bit website ([www.microbit.co.uk](http://www.microbit.co.uk)), and choose 'Create code'.



Choose 'Import Code' and upload the hex file that you've downloaded from the Live Lessons website.



The code for your micro:bot should now appear in your code window. Plug in your micro:bit, hit 'compile' and drag your hex file onto your micro:bit.

## Step 4: Your robot is ready to go

Unplug your micro:bit from the USB, plug in the battery pack, and **press button A** to watch your micro:bot move. You can **press button B** to get it to stop.

Remember to place it on a smooth surface for best results.

If you're happy with the way your micro:bot is going, you can stop there. However, there are lots of ways you can adapt your micro:bot if you want to make it your own, including changing the pattern on the LEDs and what makes it stop or go.

Turn over the page to learn more about the code behind the micro:bot and what you can change.

## Step 5: Understanding the code

```
forever
  if robotgo == false
    do
      show leds
      0 1 2 3 4
      0
      1 ✓
      2
      3 ✓ ✓ ✓ ✓
      4
      pause (ms) 1000
    else
      digital write (0,1) 1 to pin P2
      pause (ms) 500
      digital write (0,1) 0 to pin P2
      pause (ms) 500
```

### Setting up the loop

Here is where we start the program for the micro:bot.

We've introduced a **loop** that goes on **forever**, and a Boolean variable **robotgo**.

Here we've also brought in a **conditional statement**: IF the variable **robotgo** is **false**, then your micro:bot's LEDs show a neutral face.

Otherwise (ELSE), we've sent a digital signal to **pin2** to turn the vibration motor on and off in short pulses, allowing it to move. Writing **1** to the pin turns it on, and writing **0** turns it off.

```
on button B pressed
  do
    set robotgo to false
    digital write (0,1) 0 to pin P2
```

forever

### Switching the micro:bot on and off

So how do we turn on the micro:bot?

In this block of code, we've specified that when **button A** is pressed, the variable **robotgo** is set to **true**, and the neutral face on the micro:bot becomes a smiley face.

Setting **robotgo** to true sets off the pulsing vibration motor, as we specified earlier in our loop.

In the next block, we've said that when **button B** is pressed, the variable **robotgo** is set to **false** again. It then writes **0** to **pin2**, so the vibrating motor is switched off.

```
on button A pressed
  do
    set robotgo to true
    show leds
    0 1 2 3 4
    0
    1 ✓
    2
    3 ✓
    4 ✓ ✓ ✓ ✓
```

```
on button B pressed
  do
    set robotgo to false
    digital write (0,1) 0 to pin P2
```

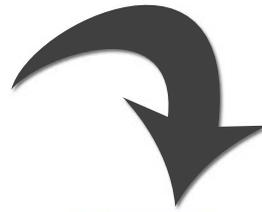
## Step 6: Customise your micro:bot

So what can you change to make your micro:bot your own?

### 1 Change what switches it on

You can change how your micro:bot is turned on and off - why not change the buttons around or make it respond to shaking?

You could even program it to respond to a sensor by attaching it to one of the pins. You'll find all the blocks you need in the 'input' section.



```
on button A pressed
do
  set robotgo to true
  show leds
    0 1 2 3 4
    0
    1
    2
    3
    4
```

```
on button B pressed
do
  set robotgo to false
  digital write (0,1) 0 to pin P2
```

### 2 Design a face for your micro:bot

If you want to, you can design different 'neutral' and 'smiley' faces for your micro:bot.

Simply change the LEDs to anything you like - this could be as simple as a tick for go and a cross for stop.

### 3 Put your art skills into practice

If you don't want to change the code, you can simply modify your micro:bot by adding eyes, tails, whiskers or any sort of extra features. Be creative and show off what you've done to your friends.