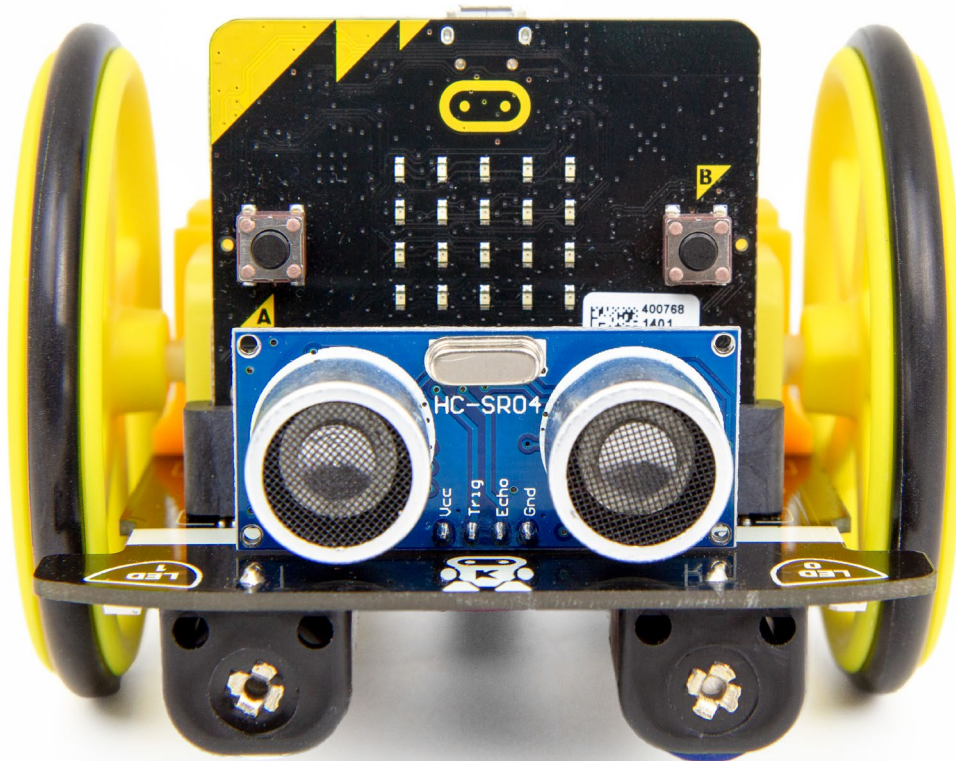

:MOVE MOTOR ADVANCED MOTOR ADJUSTMENT TUTORIAL

KITRONIK RESOURCES



INTRODUCTION

Motors are great, but even when they are supposed to be identical, they don't always run at the same speed. In this tutorial, we're going to learn how to adjust the individual motor outputs on :MOVE Motor to keep it driving in a straight line.

SETTING UP

EQUIPMENT REQUIRED:

- 1 x BBC micro:bit (www.kitronik.co.uk/5613),
- 1 x :MOVE Motor Buggy (www.kitronik.co.uk/5683)

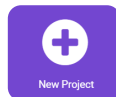
ADDING IN CUSTOM MAKECODE BLOCKS:

We have made custom coding blocks especially for the :MOVE Motor, which helps to make coding super simple within Microsoft MakeCode.

To add these blocks, follow the steps below:

STEP 1: Bring up the MakeCode Block Editor - (makecode.microbit.org).

STEP 2: Click 'New Project'.

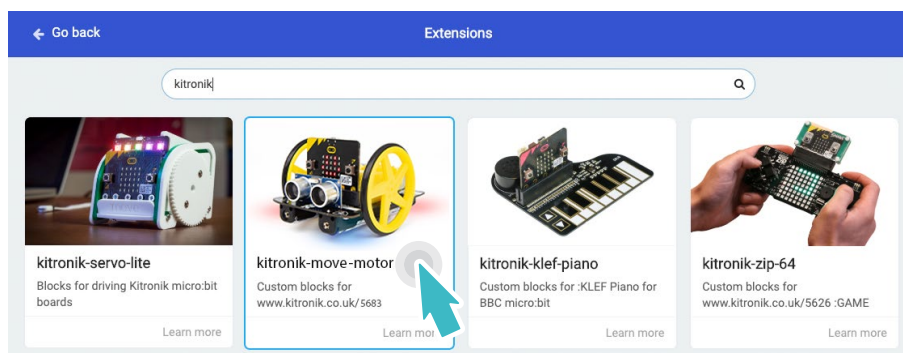
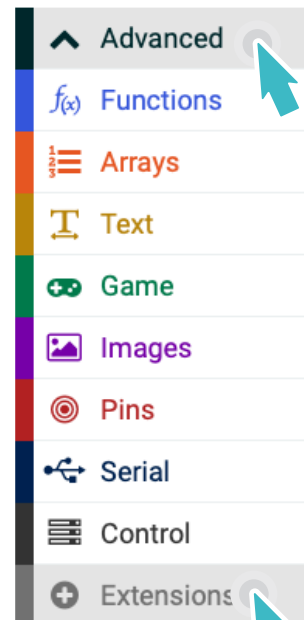


STEP 3: In the toolbox towards the left of the screen, select the 'Advanced' section. Additional block categories will appear below.

STEP 4: Select 'Extensions'.

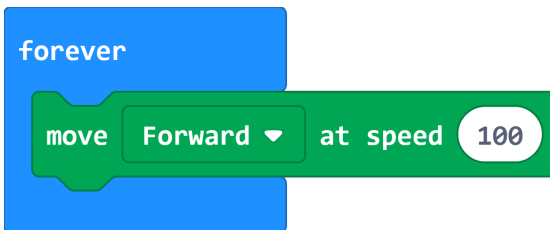
STEP 5: In the pop up's search bar type 'Kitronik'.

STEP 6: Locate & select the 'kitronik-move-motor' box.



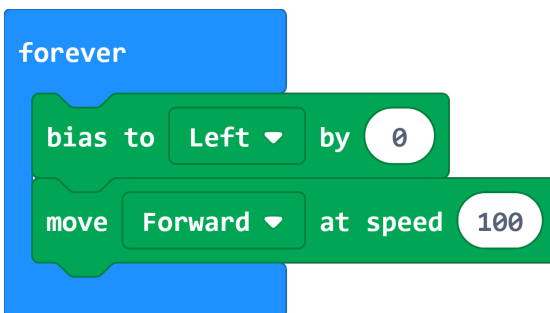
THE TUTORIAL

STEP 1: Before we start making adjustments, we need to check which way :MOVE Motor is tending to turn when it's supposed to be driving in a straight line. In the "forever" loop, add a "move Forward at speed 100" block.

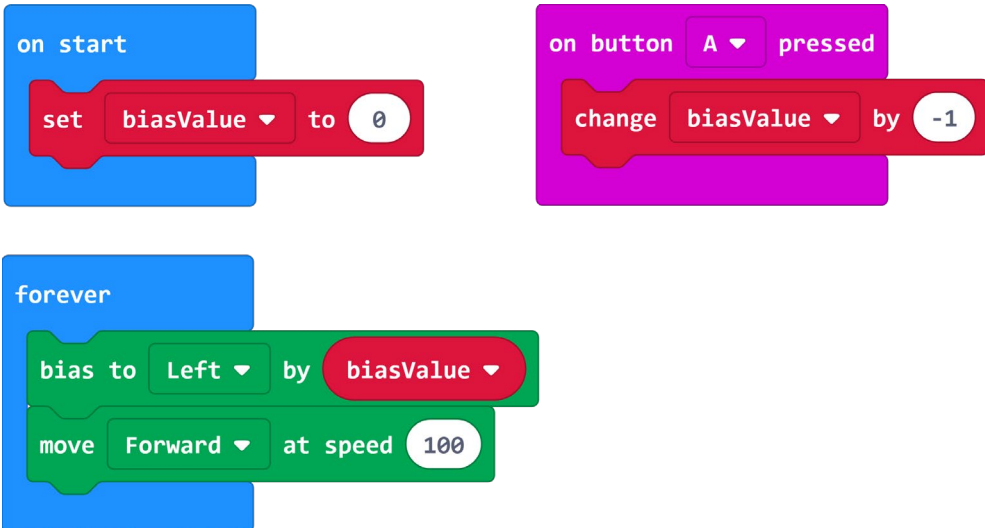


STEP 2: If you have a micro:bit connected, click "Download" to transfer your code. Make sure there is a good, open area for :MOVE Motor to drive on, and then switch it on. Watch carefully and see whether it tends to go towards the left or the right. (It might drive in a perfect line, in which case, it doesn't need to be adjusted. However, the motors are also affected by the battery voltage, so it might not stay straight forever!)

STEP 3: To actually adjust the motors, you'll need to pull in the "bias to Left by 0" block. Put it in the "forever" loop, above the "move Forward" block. If :MOVE Motor turned more left in your test, set the drop-down to "Right", otherwise leave it as "Left".



STEP 4: Rather than having to keep reprogramming the micro:bit each time the bias number needs changing, we're going to use the buttons instead. Create a variable called "biasValue" and in the "on start" block, "set biasValue to 0". Add an "on button A pressed" block and then put a "change biasValue by -1" block inside. Finally, put "biasValue" in the "bias to Left/Right" block.



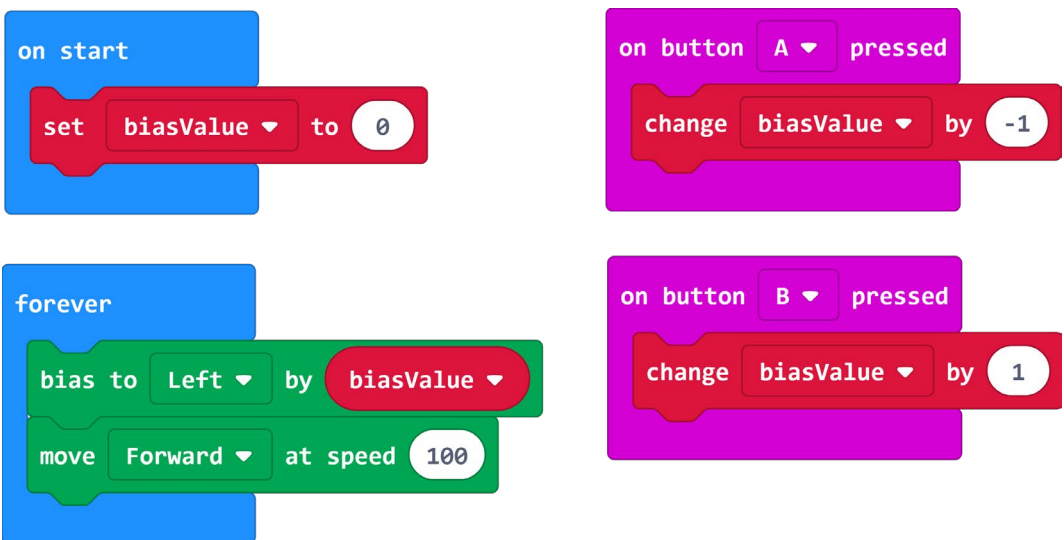
```
on start
  set biasValue to 0

on button A pressed
  change biasValue by -1

forever
  bias to Left by biasValue
  move Forward at speed 100
```

STEP 5: Add an "on button B pressed" block and inside "change biasValue by 1". If you have a micro:bit connected, click "Download" to transfer your code.

Set :MOVE Motor going again, trying different bias values by pressing "button A" or "button B". (Note: The bias can be in the range 0 to 10).



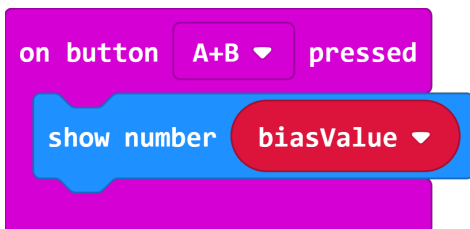
```
on start
  set biasValue to 0

on button A pressed
  change biasValue by -1

on button B pressed
  change biasValue by 1

forever
  bias to Left by biasValue
  move Forward at speed 100
```

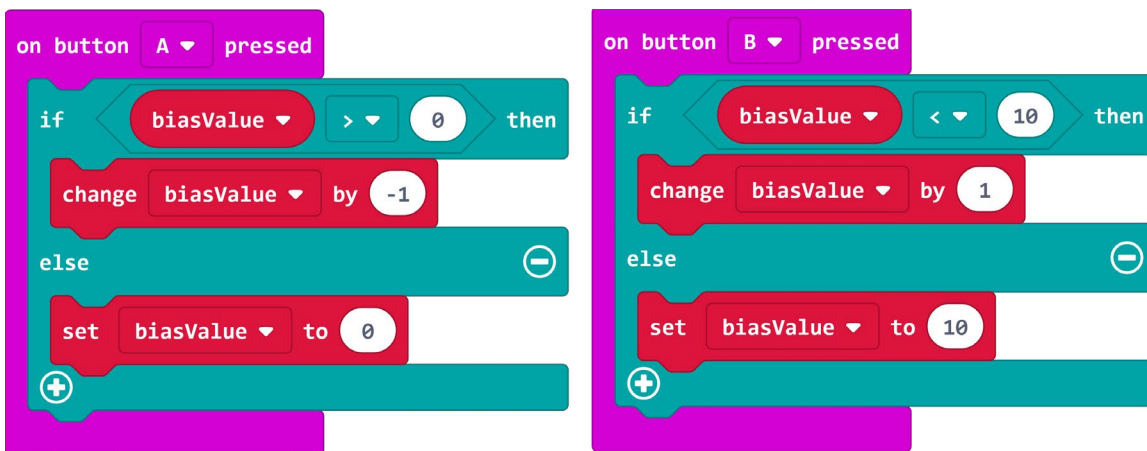
STEP 6: Hopefully the bias block with the correct setting now means that :MOVE Motor drives in a straight line. We're just going to make a few more changes to the code to make it more useful. To start with, it would be helpful to know what the current "biasValue" is. Add an "on button A+B pressed" block, and then inside, "show number biasValue".



```
on button A+B pressed
  show number biasValue
```

STEP 7: Finally, we want to make sure we don't try to set "biasValue" to something outside the 0-10 range.

In the "button A" block, add an "if else" statement, putting the "change biasValue to -1" inside the "if" section. The test condition needs to be "if biasValue > 0". In the "else" section, "set biasValue to 0". Do the same for the "on button B pressed" block, but the test condition should be "if biasValue < 10", and "set biasValue to 10" in the "else" section.



```
on button A pressed
  if biasValue > 0 then
    change biasValue by -1
  else
    set biasValue to 0

on button B pressed
  if biasValue < 10 then
    change biasValue by 1
  else
    set biasValue to 10
```

STEP 8: CODING COMPLETE! If you have a micro:bit connected, click "Download" to transfer your code. You can now adjust the motor bias as before, but now you can't set it outside the limits and you can see what value it is by pressing "buttons A+B".

For any further queries or support, please visit the Kitronik website: www.kitronik.co.uk/5683

Or get in touch:

Telephone +44 (0) 115 970 4243

Sales email: sales@kitronik.co.uk

Tech Support email: support@kitronik.co.uk

Web: www.kitronik.co.uk



[kitronik.co.uk/twitter](https://twitter.com/kitronik)



[kitronik.co.uk/facebook](https://www.facebook.com/kitronik)



[kitronik.co.uk/youtube](https://www.youtube.com/kitronik)



[kitronik.co.uk/instagram](https://www.instagram.com/kitronik)



Designed & manufactured
in the UK by 

