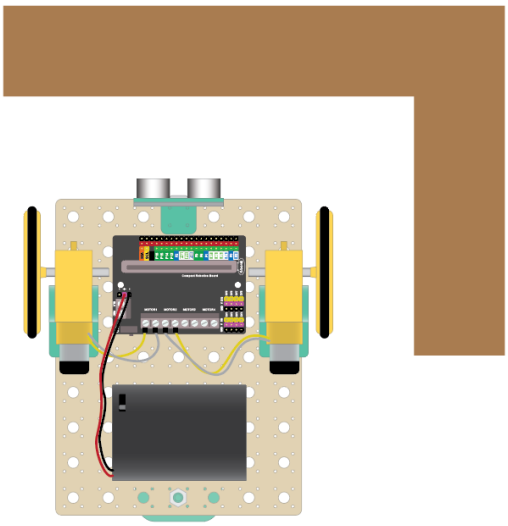
Build an Obstacle Avoiding Robot

Project 1.07

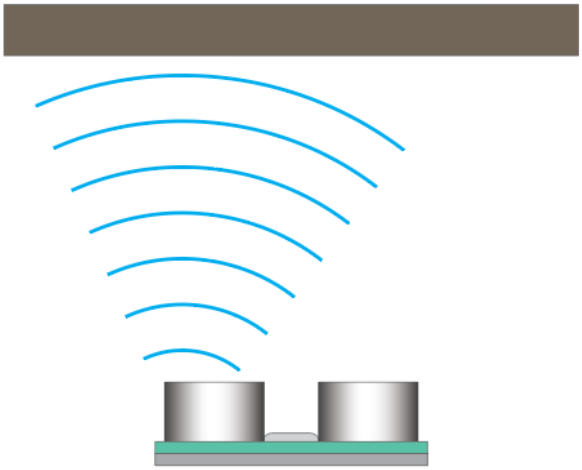
In this workshop you will make a robot that moves randomly around the room avoiding crashing into walls and other obstacles in its path.



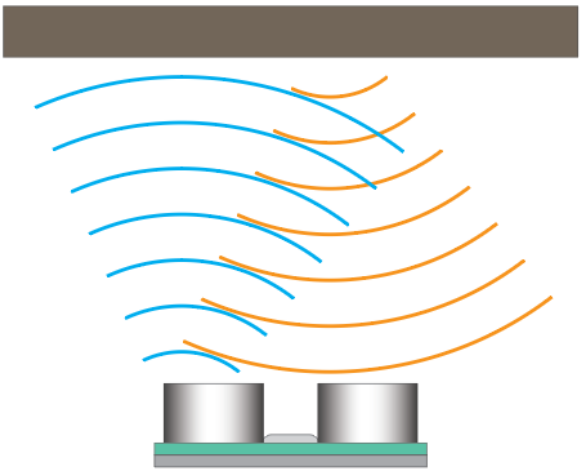
Can't go that way!

How it Works

You will attach an **ultrasonic sensor** to the front of your robot. The ultrasonic sensor sensor makes very high-pitched sounds that the human ear cannot detect. The sounds will bounce around the room and echo back to to the robot. These echos are then detected by the same sensor. By timing how long it takes for the sound to come back to the sensor, we can work out how far away the nearest object is.



Ultrasonic sounds are made by the sensor



Echos are detected by the sensor

Can you think of an animal that uses a technique similar to this? Yes! A bat can "see" using a technique called echolocation. This works exactly the same way. We are going to build a robot bat!

What to do

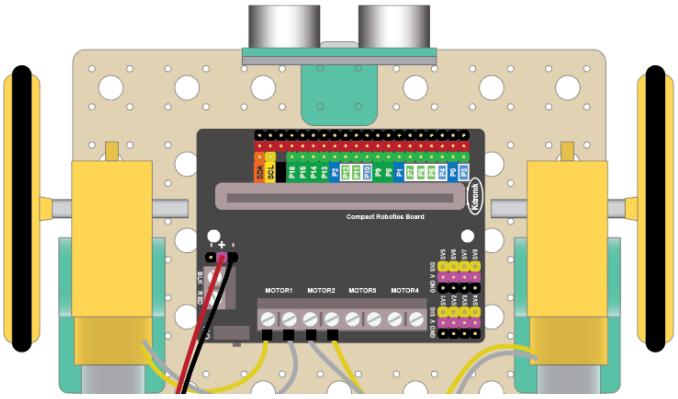
- If you haven't already done so, build the robot by referring to the previous worksheet (just build it, don't code it).
- Then follow this worksheet to add an ultrasonic sensor and start measuring distances to objects
- Finally, attempt the coding challenges to get your robot to move around the room avoiding obstacles

Add the Ultrasonic Sensor

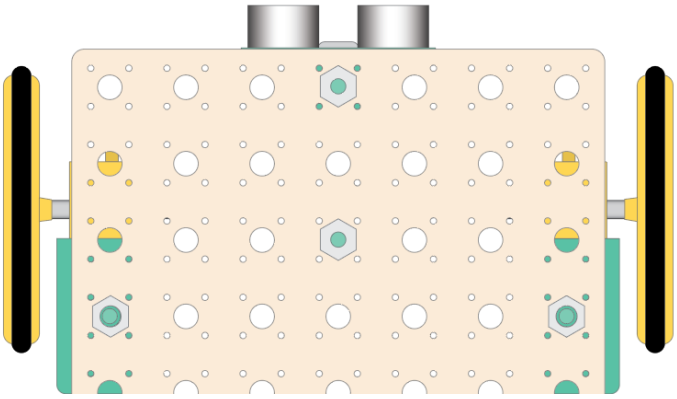
Connect the ultrasonic sensor

Connect the ultrasonic sensor to the top of the robot. It's best to connect it in the centre, but it will still work if it is offset a bit.

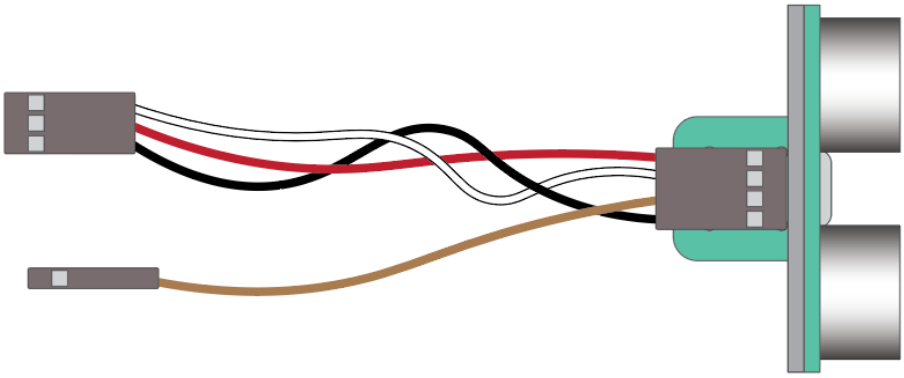
1 On top: Connect the sensor.



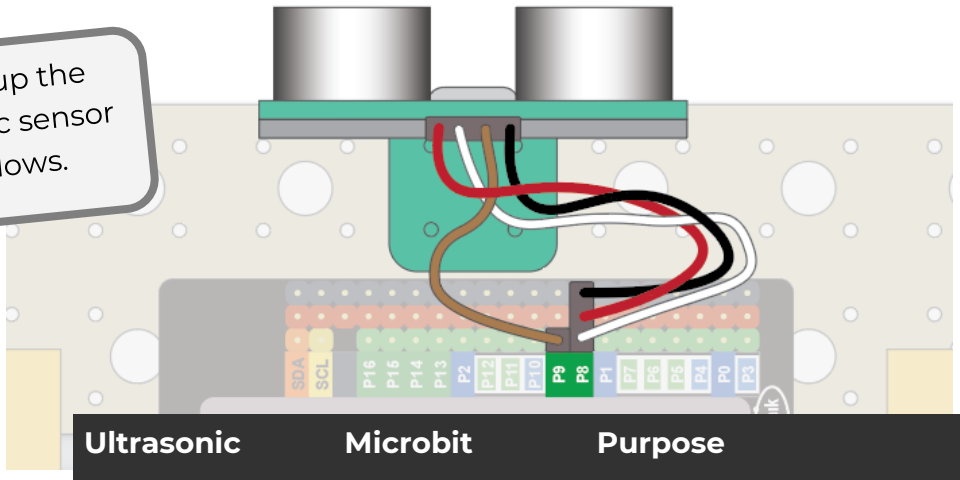
2 Under the robot: Secure with a nut.



3 Connect the special ultrasonic cable to the sensor.



4 Wire up the ultrasonic sensor as follows.



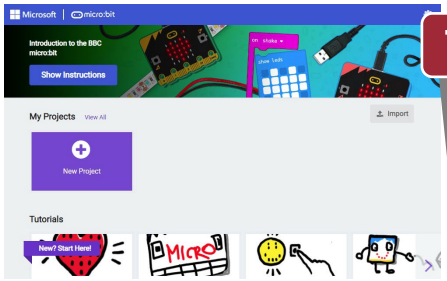
Ultrasonic	Microbit	Purpose
Black/red	Any GV pins	Power
Trig (white)	P8	Trig (make a sound)
Echo (brown)	P9	Echo (listen for sound)

Code the Ultrasonic Sensor 1

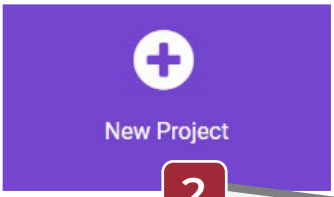
Create a Project

Create a new Makecode project so you can start coding.

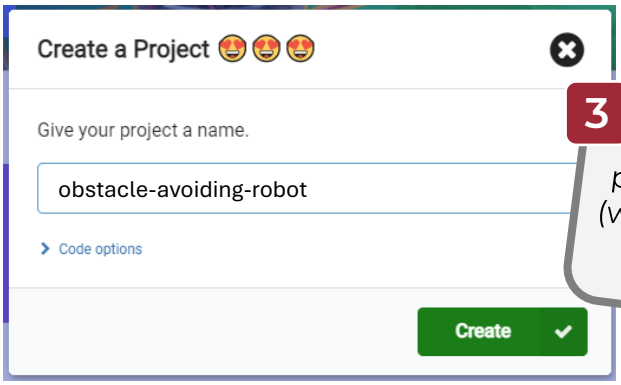
https://makecode.microbit.org/



1 Go to the Makecode website.



2 Click on New Project.

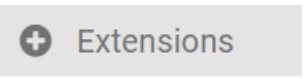


3 Give the project a name (whatever name you want!).

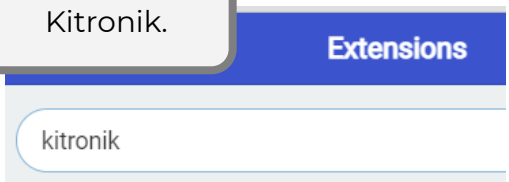
Add the Motor Driver and Sonar Extensions

The motor driver extension gives you the ability to control motors. The sonar extension will allow you to use the ultrasonic sensor.

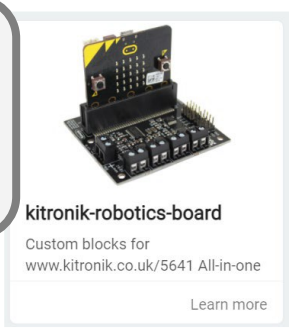
1 Select this block.



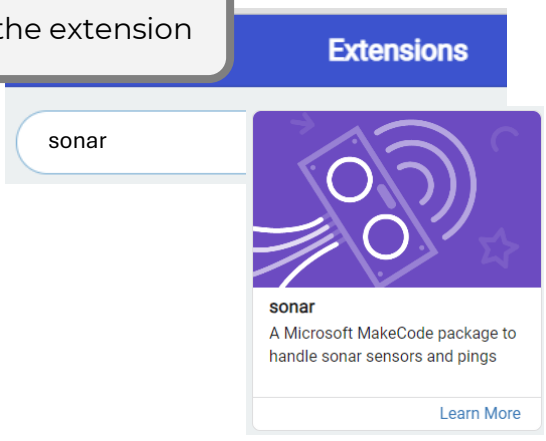
2 Search for Kitronik.



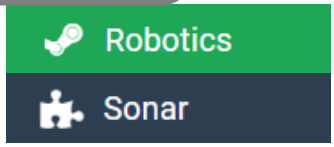
3 Make sure you select the right Kitronik extension. There are a few of them!



4 Search for sonar and select the extension



5 If all goes well you will see these new menus appear.



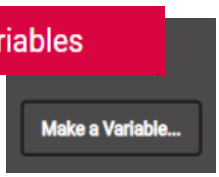
Code the Ultrasonic Sensor 2

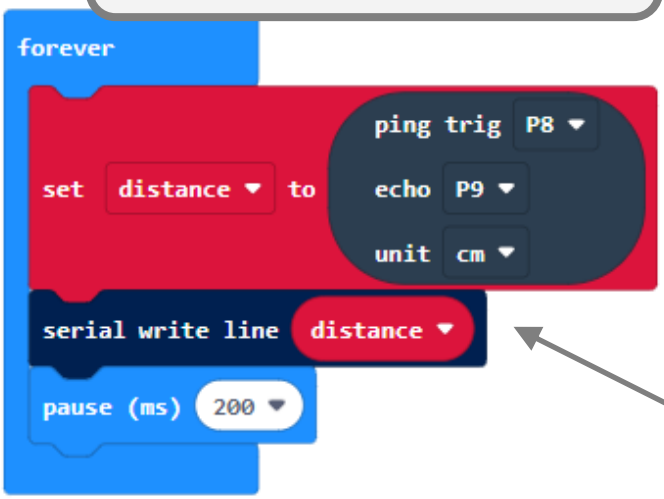
Read the Sensor Values

First we will take readings from sensor to see if we can measure distances.

1 Add this code.

You will first need to create the **distance** variable

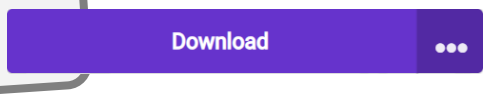




This block reads the distance in cm from the sensor

This block will send the reading back to your computer.

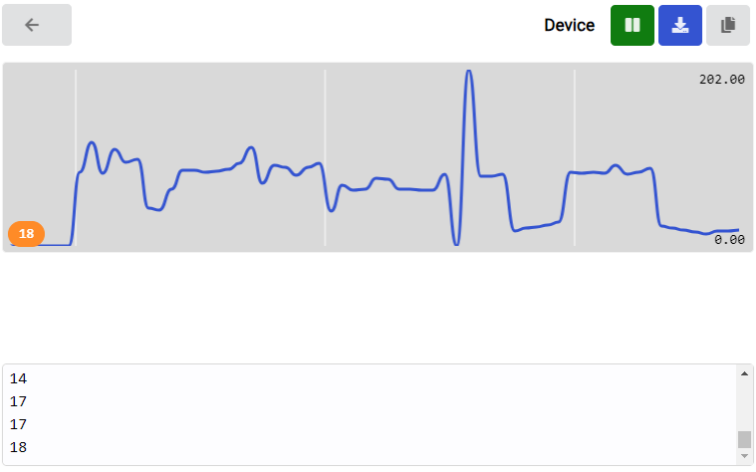
2 Download the code to the Microbit.



3 To see the data, click on the **Show data Device** button on the left.

Ensure your Microbit is still connected and paired to your computer or you won't see

4 You should see some numbers and a graph appear. These show the distance readings in cm.



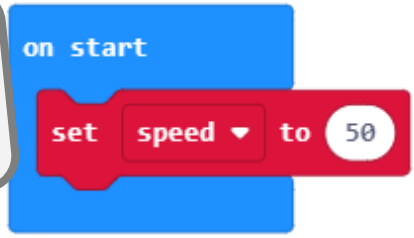
Place your hand in front of the sensor and move it back and forth. The readings should show how far your hand is from the sensor.

Code the Ultrasonic Sensor 3

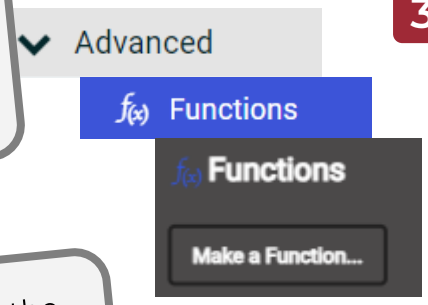
Move forwards and stop

Now let's get our robot to move forwards and then stop when it sees a wall. To do this we will create some **functions**. Functions are blocks of code that you can run whenever you want in your program, just by calling its name. This saves you repeating the same code over and over.

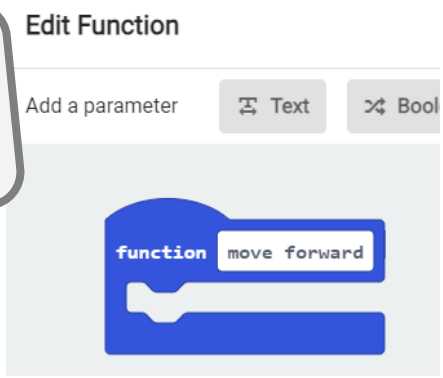
1 Create a variable called **speed** and set it to 50.



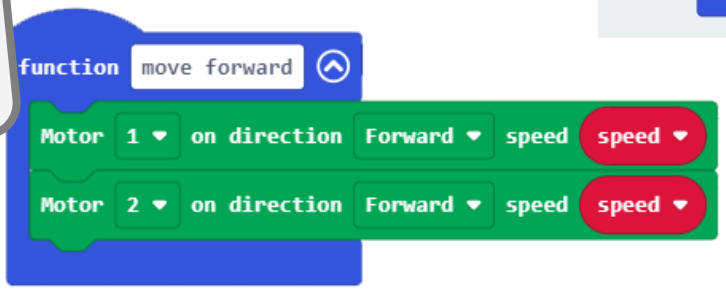
2 Create a new function.



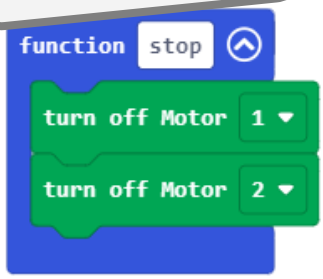
3 Name the function **move forward.**



4 Add code to the function to move the robot forward.

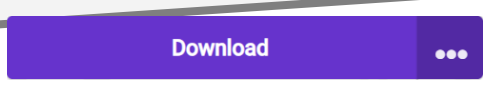


5 Create another function to **stop** the robot.



6 Change the forever code to call the functions.

7 Download the code to the Microbit. The robot should move forward but stop if it comes within 10cm of a wall.



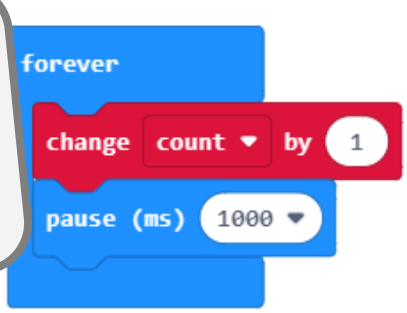
Code the Ultrasonic Sensor 4

Move randomly and stop

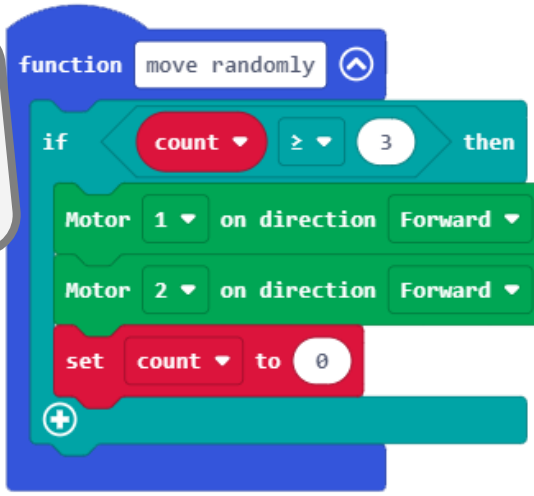
Let's make the robot movement a bit more interesting. Instead of moving forwards, we will make it move at different speeds and directions. To do this we will create a function that moves the robot at random speeds and directions and changes the speed and direction every 3 seconds.

1 Create a variable called **count** and set it to 0.

2 Create a new forever block and add this code to increase count by 1 every second.



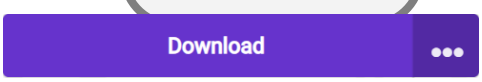
3 Create a function called move randomly.



Every 3 seconds this code changes the speed and resets the counter.

4 Change the forever code so it calls this new function, **move randomly**, instead of **move forward.**

5 Download the code to the Microbit. The robot should move randomly but stop if it comes within 10cm of a wall.



Challenges

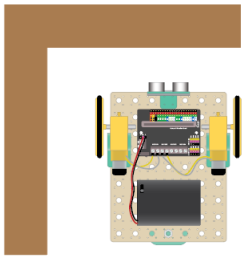
Your challenge!

When the robot sees an obstacle it just stops. Can you get it to turn around and continue moving instead?

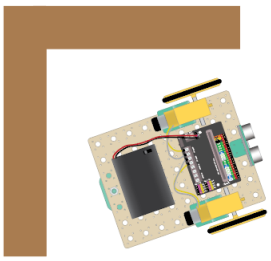
Hint: You will need a combination of movements. Create a function called **reverse** that contains all these movements and replace the **call stop** with **call reverse**.

Super challenge!

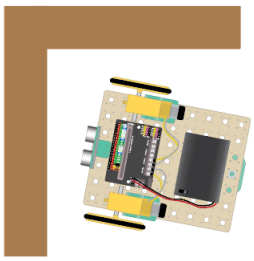
Sometimes you will find that the robot gets stuck in a corner. Change your robot's code so that it tries to find the best way out. Get it to look right then left and choose the direction that looks like it has most space to move into.



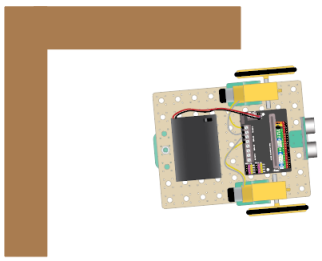
A wall!



Look right



Look left

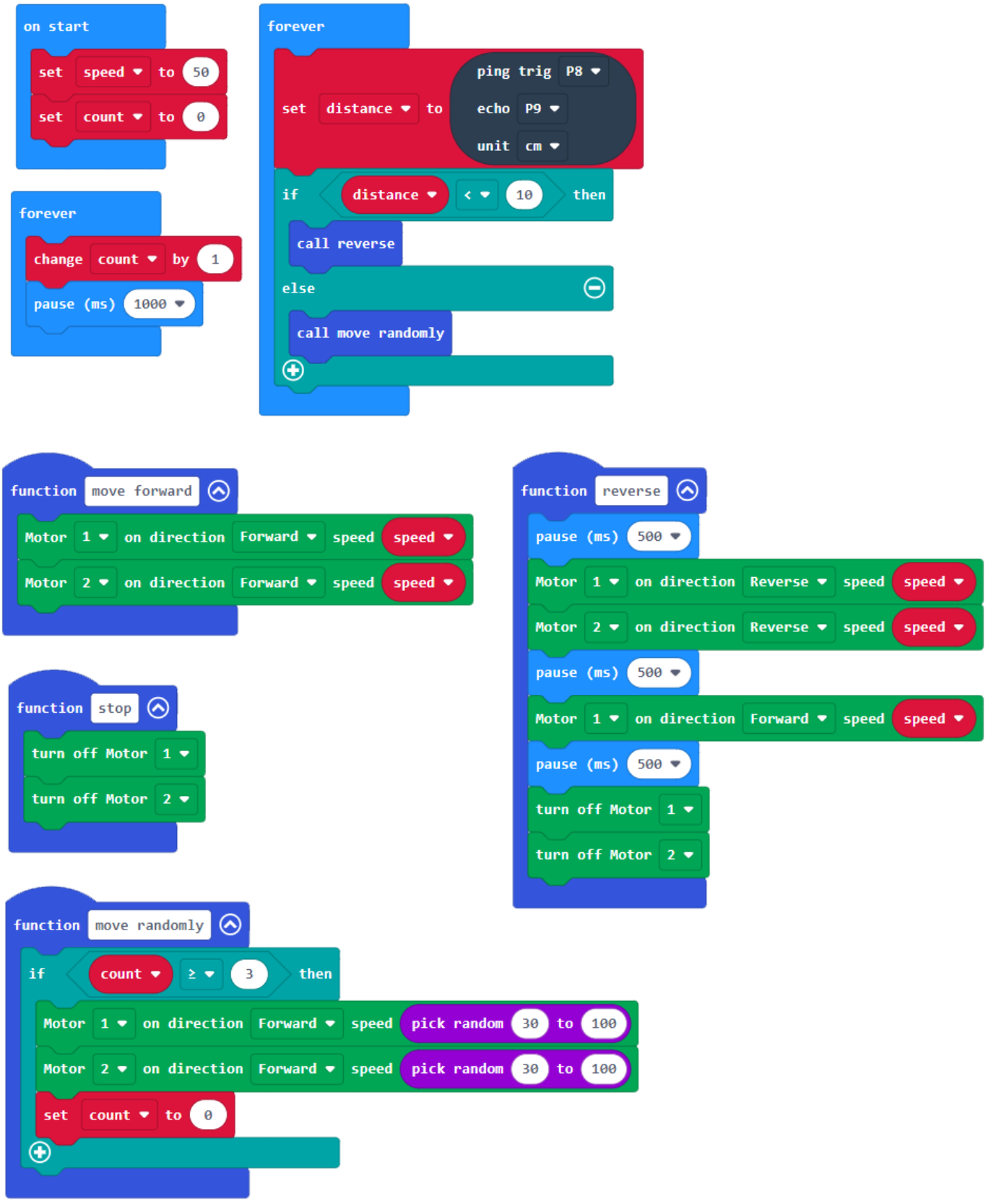


Right was better,
I'll head that way!

Solutions

Move randomly and reverse solution

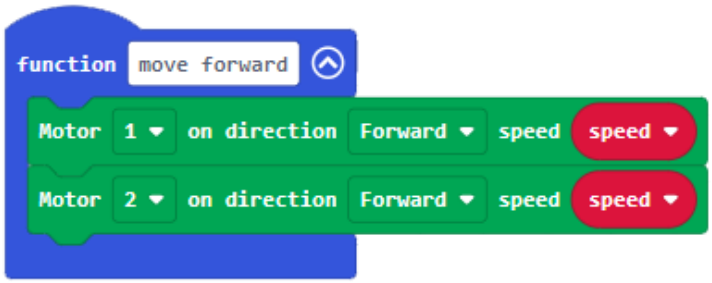
This code should make the robot reverse straight, then turn to the right and move off in the new direction. Here is the complete code.

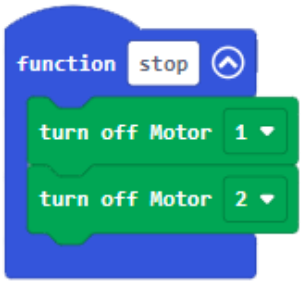


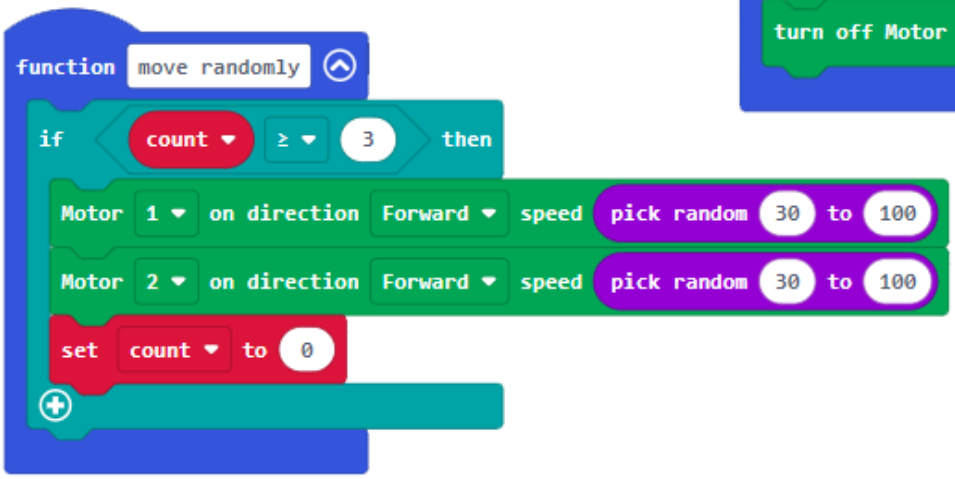
Super challenge solution part 1

When the robot sees an obstacle, this code will make the robot look to the left and right and see which is the best escape route. This is the complete code.

The move forward, move randomly and stop functions remains unchanged from previous code.



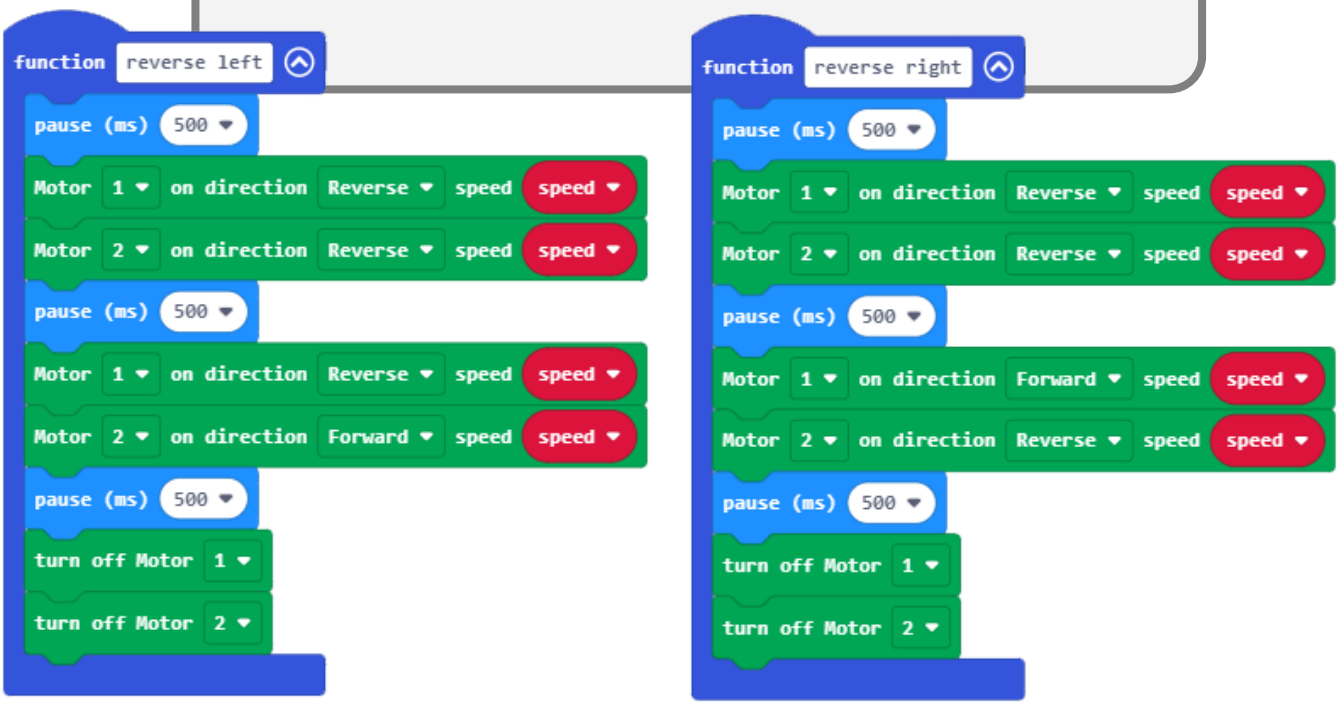




The look left and look right functions get the robot to turn to the left and then right and take a distance reading.

Super challenge solution part 2

Depending on whether we decided to move off to the left or right, we can call one of these functions.



Now the main forever block can make the robot look left and right and move off in the direction where there is most “distance” in front of the robot.