

Add an Ultrasonic Sensor to your Train

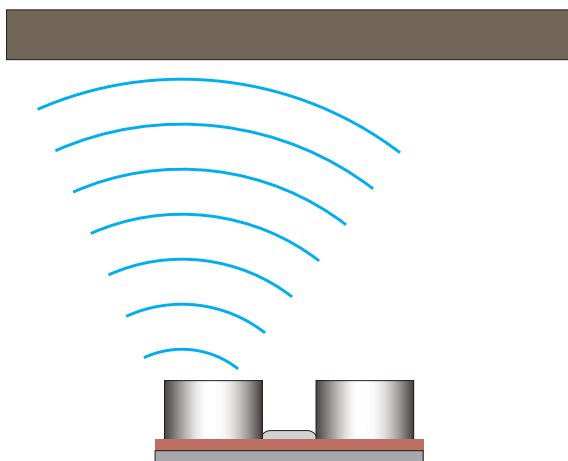
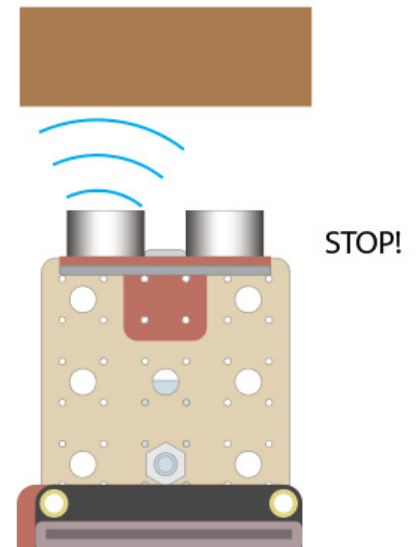
Project 2.04

In the previous worksheet, you added a crash sensor to your train so that it stopped when it crashed into a wall. But wouldn't it be better if the train could stop *before* it crashed?!

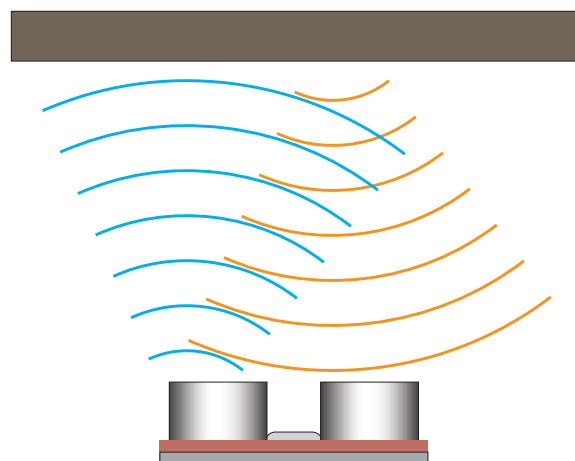
In this worksheet you will add a sensor to your train so that it detects when it is near a wall and stops before it crashes.

How it Works

You will attach an **ultrasonic sensor** to the front of your train. The ultrasonic sensor makes very high-pitched sounds that the human ear cannot detect. The sounds will bounce around the room and echo back to the train. 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 bat train!



What to do

- If you haven't already done so, build the train 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 train to slow down and stop

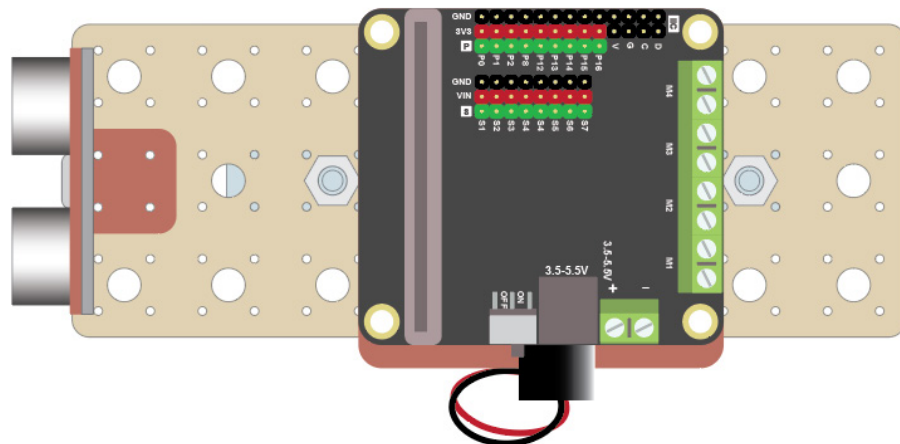
Add the Ultrasonic Sensor

Connect the ultrasonic sensor

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

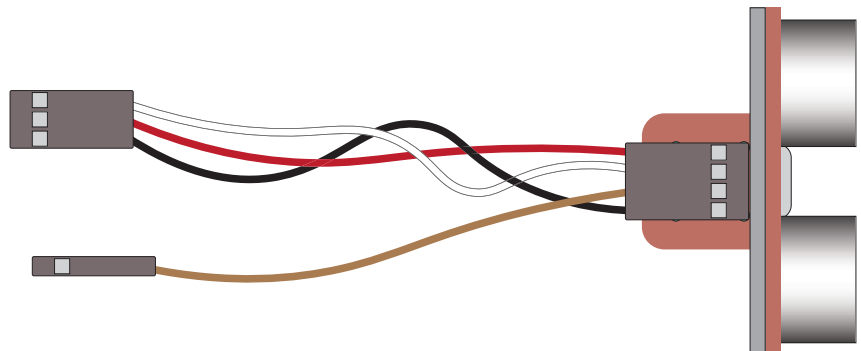
1

Connect the ultrasonic sensor, using a nut underneath to secure it



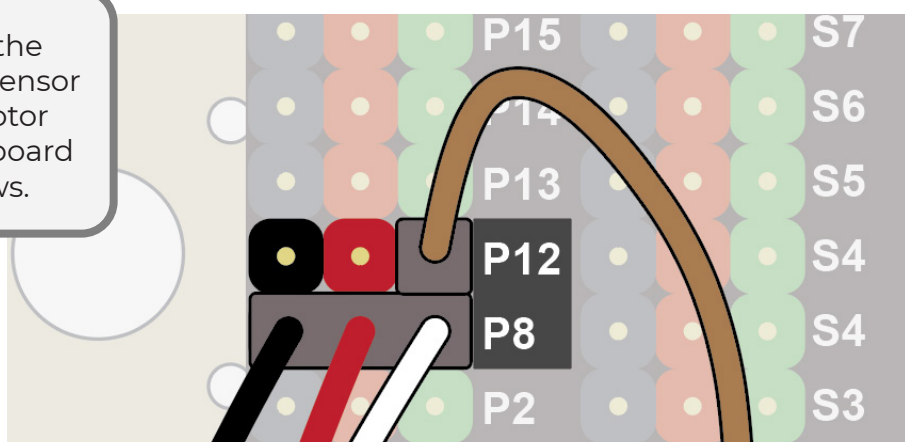
2

Connect the special ultrasonic cable to the sensor. The red wire should connect to the VCC pin



3

Wire up the ultrasonic sensor to the motor controller board as follows.



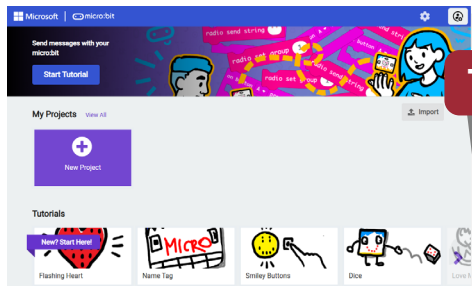
Ultrasonic	Microbit	Purpose
Black/red	Any GV pins	Power
Trig (white)	P8	Trig (make an ultrasound)
Echo (brown)	P12	Echo (listen for ultrasound)

Code the Ultrasonic Sensor 1

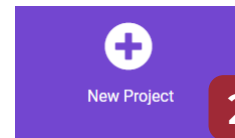
Create a Project

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.

A 'Create a Project' dialog box. It has a title bar with a close button. The main text says 'Give your project a name.' Below this is a text input field containing 'ultrasonic train'. There's a 'Code options' link below the input field. At the bottom right is a green 'Create' button with a checkmark.

3 Give the project a name (whatever name you like).

Add the Motor Driver and Sonar Extensions

The motor driver extension adds code blocks that allow you to control motors.

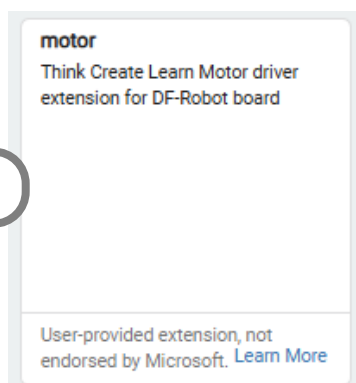
1 Select this block.



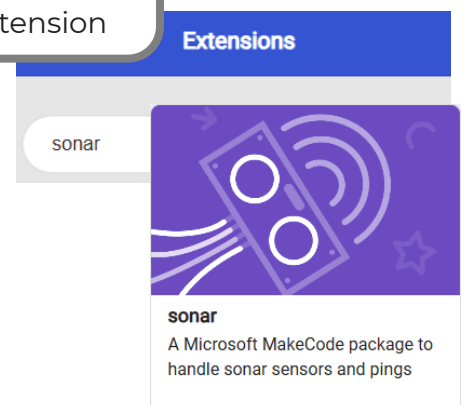
2 Enter the extension name

A search bar with the text 'github:lewfer/mb-motors' and a magnifying glass icon on the right. Above the search bar is a blue header with the word 'Extensions'.

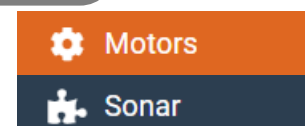
3 Add it



4 Search for sonar and add the extension



4 If all goes well you will see these new blocks appear.



Code the Ultrasonic Sensor 2

Read the Sensor Values

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

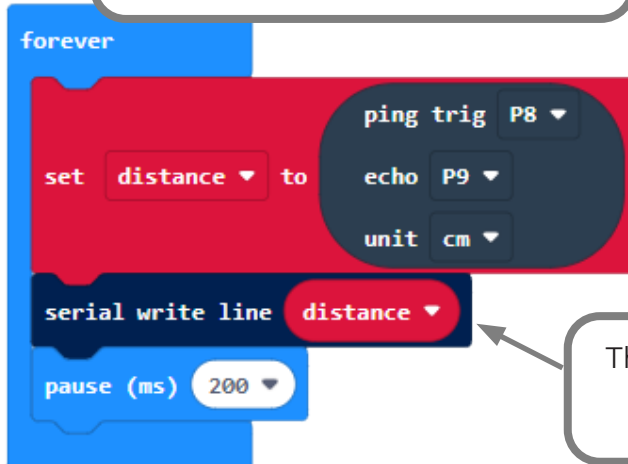
1

Add this code

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

Variables

Make a Variable...



This block reads the distance in cm from the sensor to the object

This block will send the reading back to your computer

Advanced

Serial

2

Download the code to the Microbit

Download

3

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

Show data Simulator

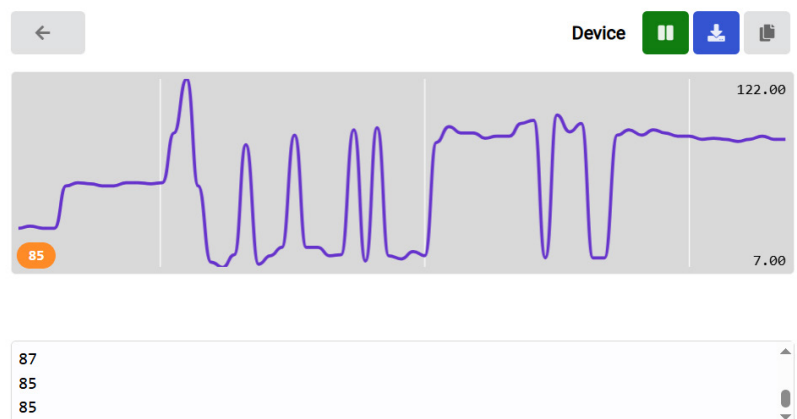
Show data Device

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

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.

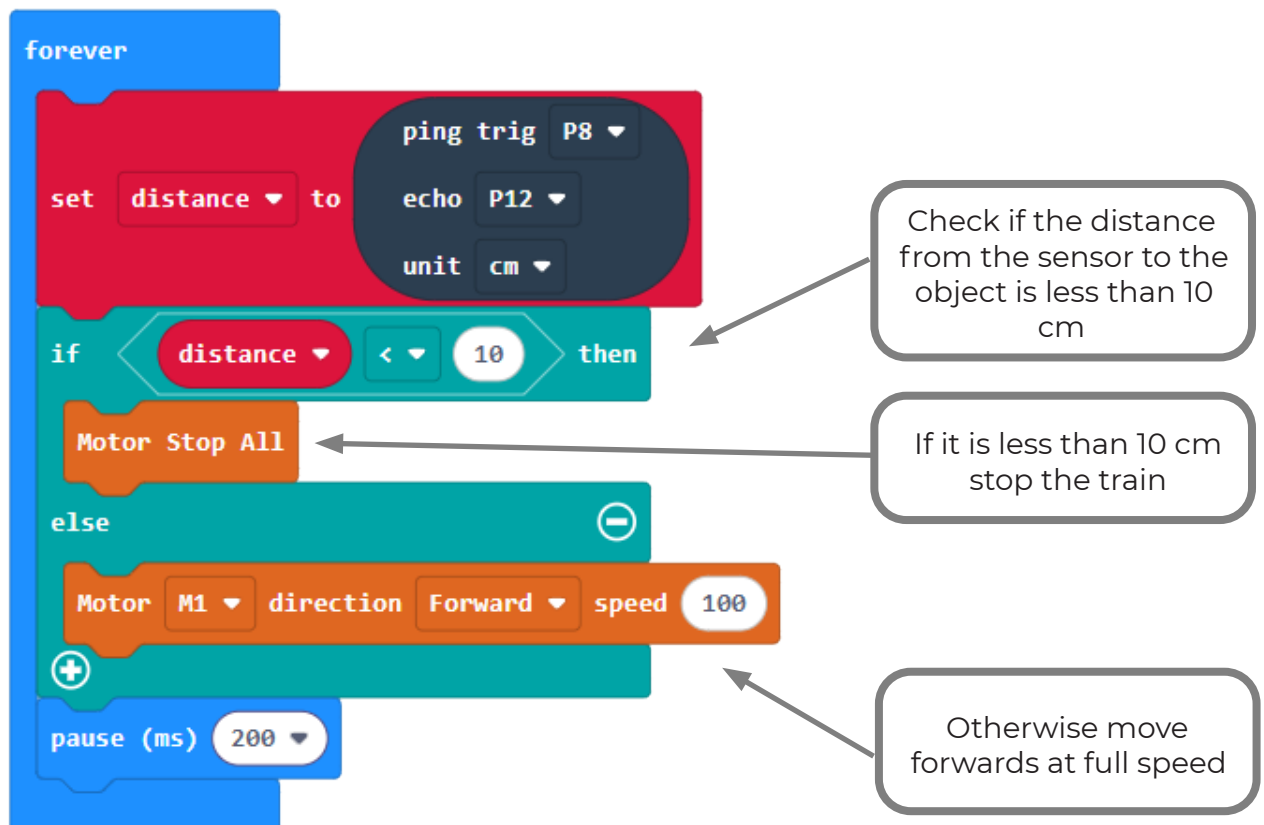


Code the Ultrasonic Sensor 3

Move forwards and stop

Now let's get our train 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.

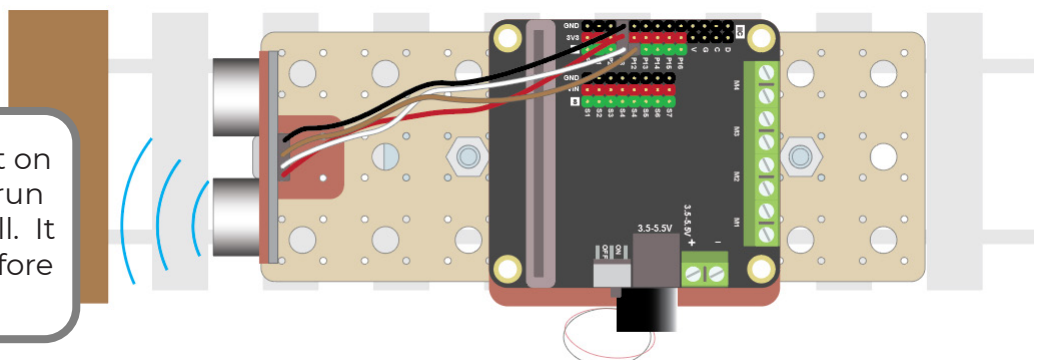
1 Change your code so it looks like this



2 Download the code to the Microbit.



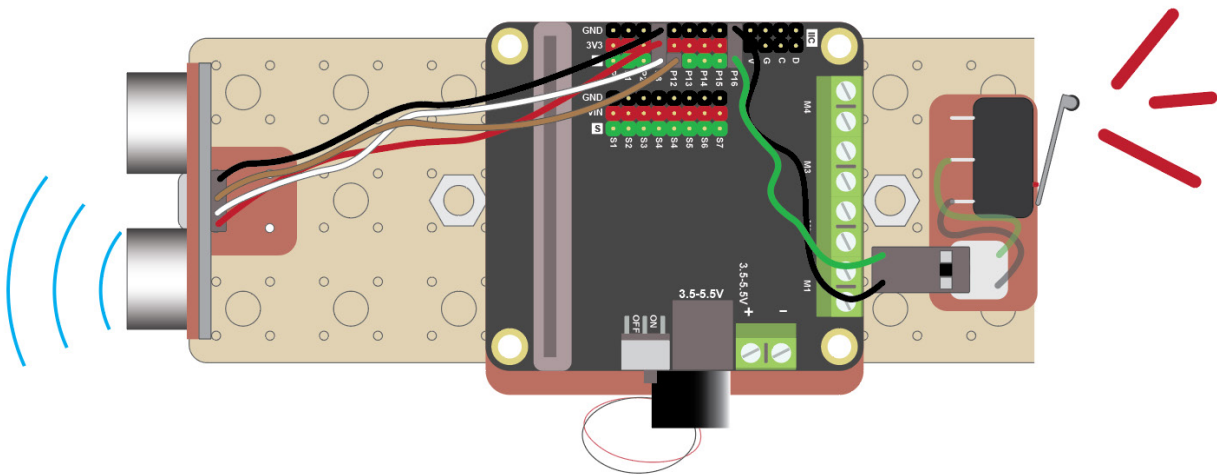
3 Place the robot on the track and run it towards a wall. It should stop before it hits it!



Your Challenges!

Now try these challenges:

- If you completed the code in this worksheet, your train should stop when it sees a wall. But it stops very suddenly. Any passengers would be thrown to the floor! Can you get the train to slow down as it approaches a wall and stop just before it?
- Can you make the train go backwards after it has stopped at a wall?
- Can you add a crash sensor on the back so that the train stops when it hits a wall when going backwards? So you will have an ultrasonic sensor on the front and a crash sensor on the back of the train:

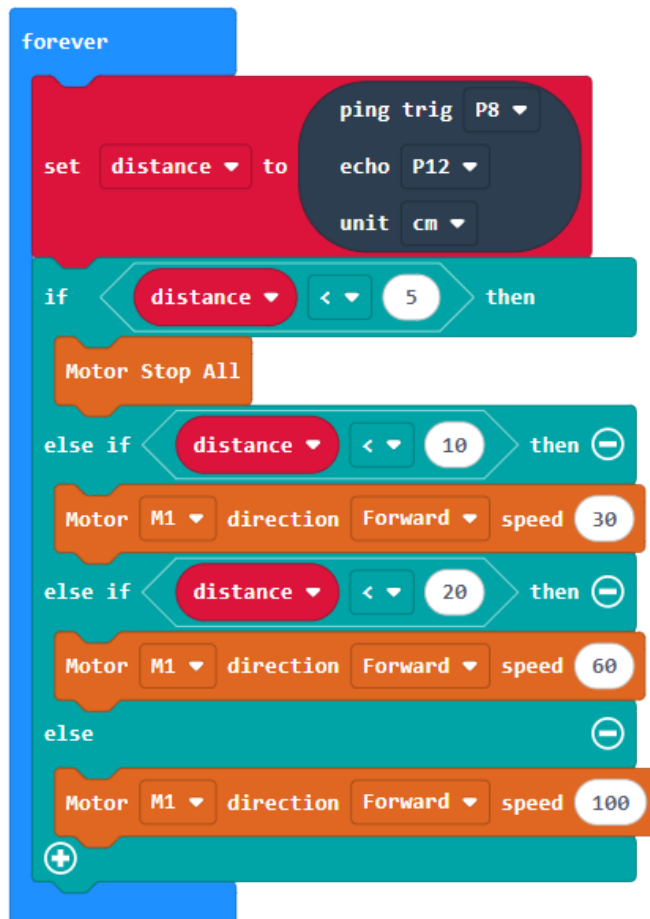


- Can you add a light and get it to light up with the train approaches a wall?

Solutions 1

Slow down and stop

This code uses the ultrasonic sensor to detect the distance from an object at the front of the train. Depending on the distance, the train will slow or stop.



If the distance is less than 5, stop.
If the distance is less than 10 slow to speed 30.
If the distance is less than 20 slow to speed 60.
Otherwise go forwards at full speed.

Solutions 2

Ultrasonic and crash sensor

This code uses an ultrasonic sensor to detect objects at the front of the train and a crash sensor to detect objects at the back of the train.

```
forever
  ping trig P8
  set distance to echo P12 unit cm

  if direction = "F" then
    if distance < 5 then
      Motor Stop All
      set direction to "R"
    else if distance < 10 then
      Motor M1 direction Forward speed 30
    else if distance < 20 then
      Motor M1 direction Forward speed 60
    else
      Motor M1 direction Forward speed 100
  else
    if digital read pin P16 = 0 then
      Motor Stop All
      set direction to "F"
    else
      Motor M1 direction Reverse speed 100
```

on start

set pull pin P16 to up

set direction to "F"

Handle the actions when the train is moving forwards:
If the distance is less than 5, stop and set the direction to reverse.

If the distance is less than 10 slow to speed 30.
If the distance is less than 20 slow to speed 60.
Otherwise go forwards at full speed.

Handle the actions when the train is moving backwards:

If the crash sensor is hit, stop the train and set the direction to forwards.
Otherwise go backwards at full speed.