# Wireless Telemetry with micro:bit

You will need:

- A laptop or PC with internet access to write and test programs (two are useful)

- Two micro:bit devices (*one to act as the transmitter and one as a receiver*)

- Micro USB cable/s to download the programs to the micro:bits

- Battery pack/s for using the micro:bit without a connection to a computer

ACTIVITY OVERVIEW

In this activity we will explore the **accelerometer**, **LED plotting** and **radio communication** features of the micro:bit.

One micro:bit will be programmed to continually measure and transmit movement data captured using the in-built accelerometer sensor. The other micro:bit will receive this information by radio and draw a graph based on it in real-time on its screen. Using this we will be able to detect if and by how much the first micro:bit is moving from the second without them being connected by wires.
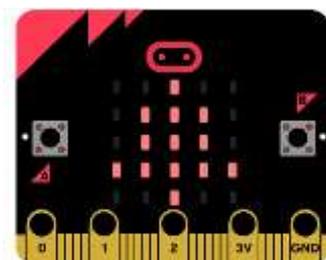
This technique is called **wireless telemetry.**

Radio Communication
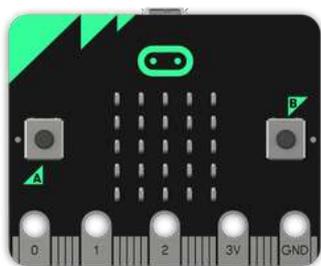
Up to 70m / 200 feet

Transmitter

Receiver

Micro:bit 1: Setting up the transmitter

We first need to set a **radio group.** This allows micro:bits that are in the same group to share data with each other. To measure rapid movement changes we can set the **accelerometer range** to be 8g (*the maximum*)

We then need to set a **forever loop** to continually transmit the value of the **accelerometer** by **radio**.
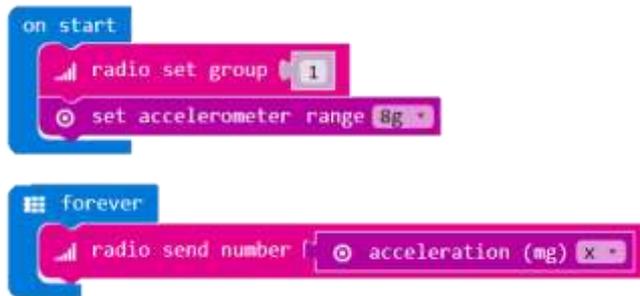
In this program the radio group is 1 and acceleration in the X dimension is being measured.

If you want to measure G forces in the Y or Z dimension, you can use this diagram to help identify the different micro:bit axes.
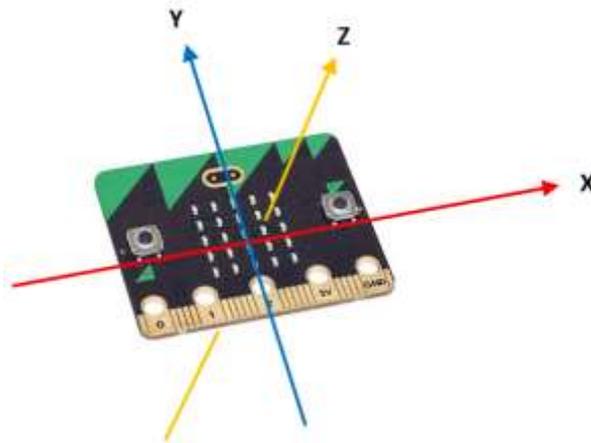
That's it!

Name your program so that you know that it is 'transmitter code' and download it to the first micro:bit. It might not seem to be doing much but it is already busy transmitting data by radio.

Now for the other micro:bit…..
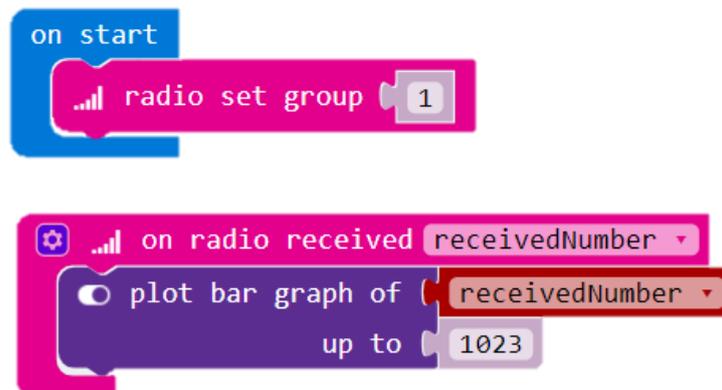
Micro:bit 2: Setting up the receiver

As before we need to set the **radio group** to be the same as the transmitter.

A **on radio received** block will activate whenever a number is received by wireless.

The **plot bar graph block** attached to it then displays a bar graph of the data based on the last number received.

*Note: You can find the **receivedNumber variable** in the variables section.*

Note: The 'up to' part of the plot bar graph sets the range. If the plot fills up too quickly you can increase this. (Good if doing high g tests)

```
on start
    radio set group  1
```

```
    on radio received  receivedNumber
        plot bar graph of  receivedNumber
                     up to  1023
```

Give it a name that reminds you it is receiver code and download it!

If both micro:bits are powered up it should be working.

Why does the receiver show a plot even when the transmitter is not moving?

A very good question!

What the micro:bit is detecting at rest is the acceleration due to gravity. The gravitational pull of the Earth accelerates objects toward it at 1G.

If you were on The Moon the micro:bit would only measure about 1/6 as much!

You need to be aware of this and know in which dimension gravity is pulling when doing Rocket Car experiments with accelerometers.

Dendrite
connecting learning

micro:bit

The Learning
Partnership
.com

Improving your program

Idea 1: Show that the transmitter is working.
There is not much to see on the transmitter. Add some code so that it indicates to users that it is transmitting radio data. This could be an animation or scrolling words.

Idea 2: Make the receiver 'ask' for the acceleration value as a number.
It might be useful to be able to get an instant reading of acceleration remotely.   Add blocks to both bits of code that allows a button to be pressed on the receiver that makes the transmitter collect the current acceleration value and transmit it back. Have the receiver then show this as a number rather than as a plot so it can be written down or recorded.

Idea 3: Have the transmitter record and transmit acceleration data from all three dimensions or the compass direction.
This is an interesting challenge. How can you display all the information on the receiver? Could you use different buttons (A, B & A+B) to request different data?

Idea 4: Make a radio relay station.
The range of the micro:bit wireless radio in the open and in the outdoors is around 200 feet or 70 metres. Indoors it is much less.
Why not have a third microbit programmed to act as a relay, retransmitting the data it collects to another receiver or another radio group.

Idea 5: Make a simple text message service.
Use buttons to select messages that you send to your radio group.
Or use buttons to select different radio groups so you can choose which micro:bits to send messages to.