DTiF
Digital Technologies in focus
Initiative of and funded by the Australian Government Department of Education and Training

acara AUSTRALIAN CURRICULUM, ASSESSMENT AND REPORTING AUTHORITY

CLASSROOM IDEAS: YEARS 5–6

DIY micro:bit metal detector

The micro:bit, a small programmable microcontroller, has a magnetometer which senses Earth's magnetic force. It can be used to create a micro:bit compass and for mapping activities. As it is a programmable device with an LED display, the micro:bit can show text such as 'N' or 'S' (north, south), arrows that change to point constantly north or degrees from north as numbers; for example, 270, 45 or 180.

Metallic objects produce a magnetic field when electricity is near them. This means that the magnetometer in the micro:bit can be used to detect metal. Big pieces of metal are best. A metal pole is very good and if it is rusty, even better. A nail, screw or thumbtack is not so good. Using this information, students can make a metal detector with the micro:bit and then go hunting for gold (Figures 1a and b). This activity shows one way to incorporate Digital Technologies into a goldfields unit in an authentic way.

Figure 1: Students from Tea Gardens Public School, New South Wales, show their micro:bit metal detectors and 'gold nuggets'

The plan

The plan is to make a rock really metallic and detect it with the micro:bit. Once it works, make more, hide them in the long jump pit and get students to detect them using broom handles with micro:bits attached.

To save you time, it is useful to know that all the items shown in Figure 2 *do not work* with the micro:bit. There doesn't seem to be enough metal to create a magnetic field strong enough for the micro:bit to register anything except Earth's magnetism. So, what does work? How about magnets?

Figure 2 shows items that do not work with the micro:bit. L–R, clockwise: A metal coating paint, a rock with most of these materials on it (which the micro:bit cannot detect), copper powder, metal filings, paint with micaceous iron oxide in it and a handful of metal pieces.

Safety considerations: Always follow appropriate risk assessment procedures. If using magnets for metal detection, it is recommended that teachers discuss safe use of small and particularly strong magnets such as rare earth magnets with students. These could be harmful if swallowed by younger children. They should be handled with this in mind and packed away carefully after use. Copper powder, (pictured in Fig. 2) is dangerous if inhaled therefore is not a safe resource.

Materials: Resources used in this activity can be purchased from hardware stores and electronics suppliers.

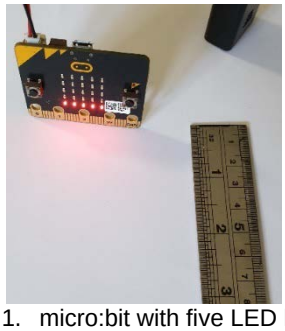
Using a magnet to trigger the micro:bit metal detector

(a) Sewable magnet

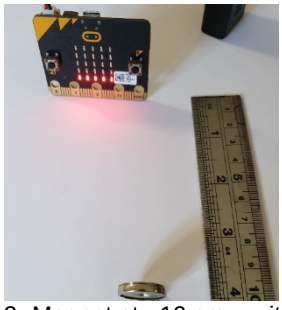
A sewable magnet (Figure 3) is great for creating wearable technology and also perfect for detection with the micro:bit magnetometer.

Ask students to use this magnet to do an investigation (collect data and make inferences) similar to the example images 1–5 in Figure 4.

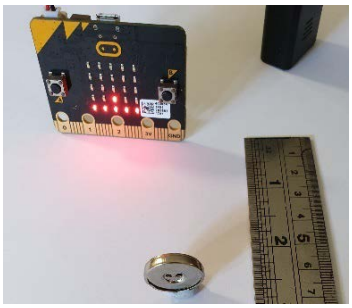
Figure 3: A sewable magnet



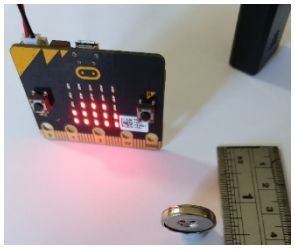
1. micro:bit with five LED lights indicates no magnet present – no detection



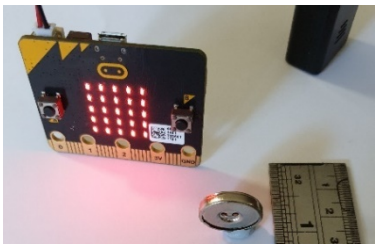
2. Magnet at ~10 cm – with five LED light indicates still no detection



3. Magnet at ~7 cm – Shows one extra LED light - more detection



4. Magnet at ~3 cm – eight extra LEDs



5. Magnet at ~2 cm – all 25 LEDs indicate magnet is fully detected

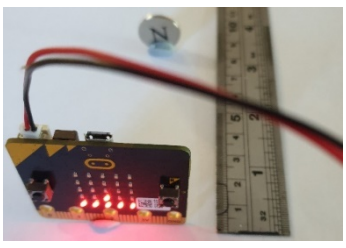
Figure 4: micro:bit detection of the sewable magnet at various distances

(b) Neodymium rare earth magnet

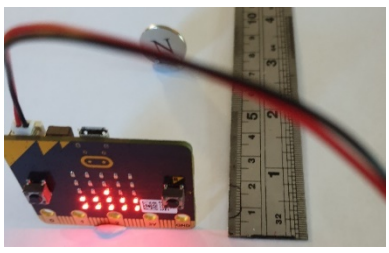
Figure 5 shows a neodymium rare earth magnet. A pack of five can be purchased from most large hardware stores for about $11. Students might benefit from experimenting with other magnets to determine what works best.

Since this magnet will be buried in a sandpit, it is probably a better choice because the micro:bit can detect it more easily. See the examples in Figure 6.

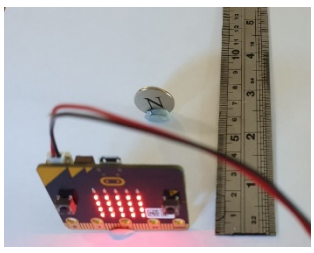
Figure 5: Neodymium rare earth magnet



1. At ~10 cm it is detected



2. ~8 cm shows more detection



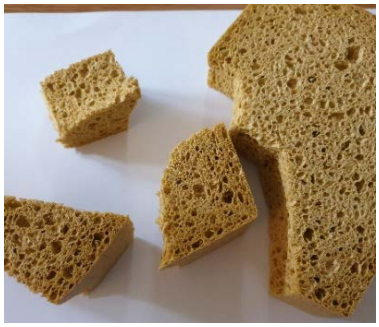
3. Strong detection at ~7 cm

Figure 6: micro:bit detection of the neodymium rare earth magnet at various distances

Making safe gold nuggets for detection

Hiding rocks with magnets glued to them in the school long jump pit could cause trouble so here is an easy way to make your own ‘safe gold nuggets’.

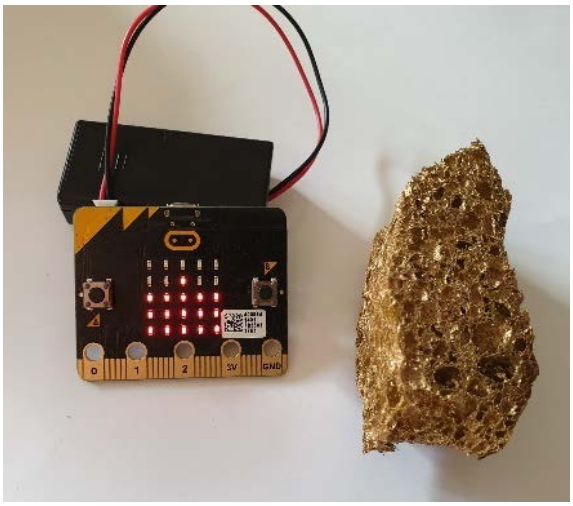
1. Find or buy a large, thick sponge – something that will look like a gold nugget.



2. Cut the sponge up into chunks (snip off corners to create shapes that better resemble rocks).

3. Cut a slit in the sponge to fit the magnet. Glue the magnet in place so it doesn’t fall out while the students dig for it.

4. Spray paint the sponge (with the magnet hidden within) using a gold (or other) colour to make it look like a precious metal.



5. A finished ‘gold nugget’ being detected by the micro:bit. This nugget has a weak magnet inside it, which is not as powerful as a neodymium rare earth magnet.

Now for the coding

Use MakeCode www.makecode.microbit.org, an online simulator/emulator where students can code and test a virtual micro:bit then download that code to the physical device.

To code the micro:bit there are a few possibilities:

a) The black box approach, where the students use a block of code (visual programming) that essentially hides a lot of the computational thinking.
b) A longer approach which includes much more computational thinking.

(a) Black box approach

A black box hides the thinking from the programmer. In this way, it is like a mobile phone: everyone knows how to use one but few people could identify the parts within it, let alone describe how they all work together to allow us to use GPS. The first example of the simplified (or black box) code is shown in Figure 7.

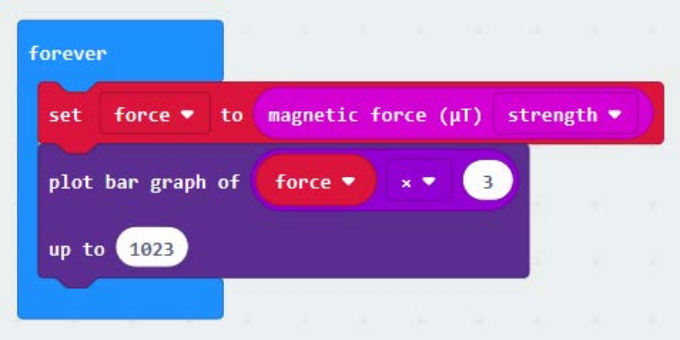


Figure 7

The variable 'force' (although not really needed) has been added in this example because getting new programmers used to using variables is good practice. As a simpler option, students could just use 'magnetic force (strength)' in the plot bar graph block directly (shown in Figure 7 in the third code block).

Note: When you download and run new code from https://makecode.microbit.org/ the micro:bit will first ask to be tilted.

This tilt instruction is just to calibrate the device so it will be ready to take measurements. To do this, tilt the micro:bit until every LED has lit up. A smiley face will then appear. Calibration need only occur each time the program is downloaded to the micro:bit (which may be only once).

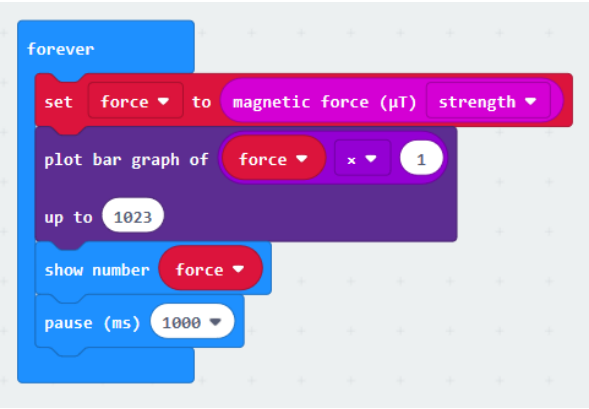


Figure 8

In the second example (Figure 8), the variable 'force' is multiplied by 1 rather than 3 (that is, left raw).

The x 3 value in the first example (Figure 7) is simply making the graph a little more sensitive to the magnetic field/presence of a large piece of metal. This is optional if you prefer a less complicated approach.

Notice that when the micro:bit is nowhere near a metallic or magnetic object it still reads a measurement of around 60. Asking the students for ideas as to what this reading is actually reporting would be a good discussion starter.

(b) Computational thinking approach

This approach adds in a little more computational thinking. Start with the code (Figure 9). This will give the students some numbers to work with.

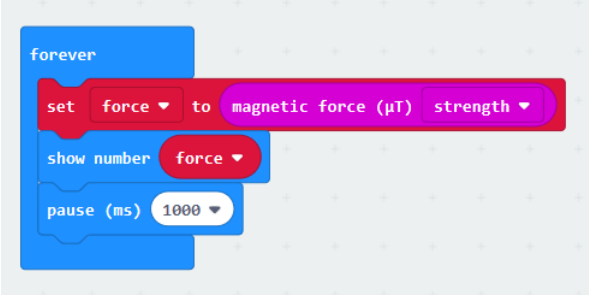


Figure 9

1. Place the micro:bit well away from any magnet or large piece of metal. Students should get some base readings of around 60. (It might be interesting to discuss that the base reading is due to magnetic north).
2. Ask the students to write down the number and start making a table.
3. Slowly introduce a magnet or large chunk of metal, making note of the numbers reported as the magnet and micro:bit get closer to each other.
4. You may also notice that the micro:bit starts reporting a much larger number even when the magnet is taken away. If this happens it will need to be recalibrated so a bit of patience may be required.

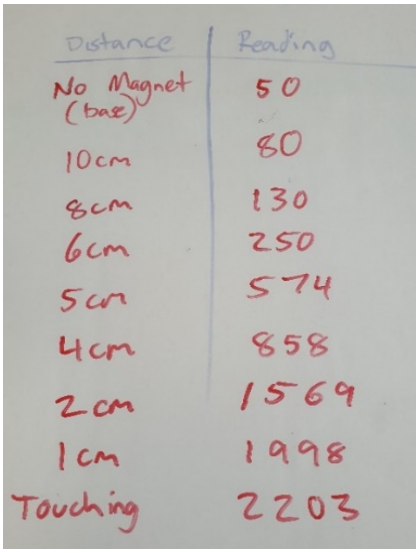
Figure 10

A simple part-solution is to include a recalibration function in the code (Figure 10). The calibrate compass block is found in the **…more** section of the input blocks. This way, if the micro:bit starts reporting strange readings it can be reset by pressing button A.

Always recalibrate the micro:bit far away from nearby magnets or large pieces of metal.

Students may come up with a table (Figure 11). It might be good practice to get three readings for each distance and find the average (applied mathematics). Once there are values for distances, we can then write the code.

What we want to happen is that as the micro:bit gets closer to the metallic source, the number of LEDs lighting up increases. The code in Figure 12 is one way to do this; there are, as in all programming, other ways to do it.



Distance	Reading
No Magnet (base)	50
10cm	80
8cm	130
6cm	250
5cm	574
4cm	858
2cm	1569
1cm	1998
Touching	2203

Figure 11

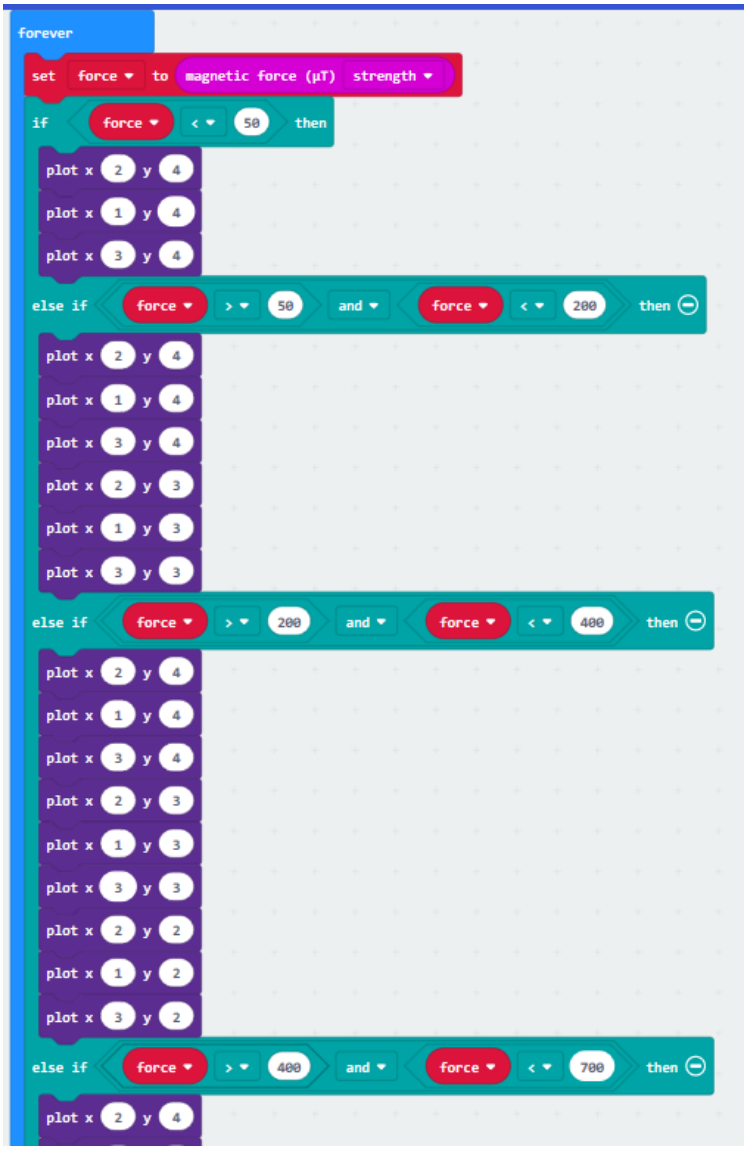


Figure 12 (continued over page)

We can use a conditional statement to capture the findings in a table.

The code, which lights more lines of LEDs as the signal becomes stronger, can look like Figure 12 (and continued on to the next page):

There is an opportunity to discuss the following when coding this with the students:

- X and Y coordinates
- computers start counting at zero
- comparisons (**force < 50**)
- comparisons include <, >, <=, >=, = and ≠
- logical statements like ‘and’ where both sides of the comparison need to be true
- the need to reset the LEDs to off after each loop. The blank **show LEDs** block does this.
- The final ‘else’ does not need a comparison – students could work out why.

All of the dot points above are part of computational thinking.

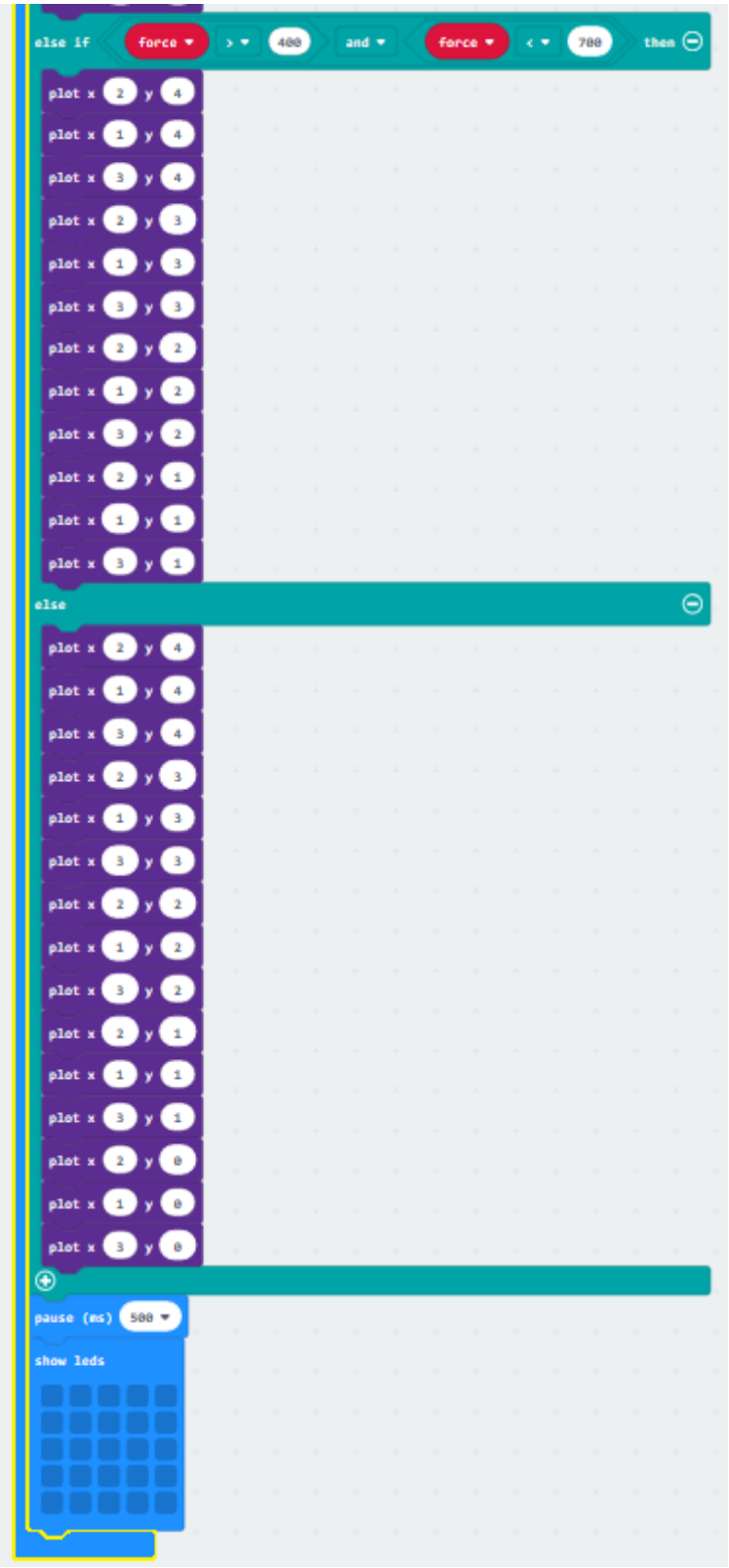


Figure 12 (continued)

(This code is continued from the previous page.)

Students will probably find it easier to plot X in the order 1, 2, 3.
In the pictured example (Figure 12), 'x' was plotted in a 2, 1, 3 order.

A pause instruction has been included at the end of the code sequence so that data overload doesn't occur. If the 'pause' block is a smaller number, then the transition between being closer and further away from a magnetic source will appear smoother.

Finally, note that there are better ways to plot multiple LEDs. This way would make a good extension activity.

Crafting a digital solution – extension activity

Once students have coded their micro:bits and tested the code, they could complete the extension activity (see next page).

Students will create their very own metal detector using two micro:bits (one to transmit and one to receive data). See Figure 13.

Make a do-it-yourself metal detector using the following steps:

1. Find something to use as a handle. A wooden broom handle, large length of dowel, metre ruler or wooden garden stake are all perfect for this purpose.
2. Using Blu Tack, attach two micro:bits to either end of the handle, noting which micro:bit is transmitting and which is receiving data.
3. Tape the battery packs to the handle.

Figure 13: A micro:bit metal detector made with two micro:bits, two battery packs and a wooden garden stake

Extension activity – getting the micro:bits to talk with one another

Having students crawl around the ground looking for gold nuggets can be a lot of fun. However, it can also be hard on their clothes, knees and hands. With a little systems, design and computational thinking we can create a micro:bit system that makes detecting the 'gold' much more civilised.

The plan is to have a micro:bit in charge of detecting the gold and transmitting these data. The micro:bit can be attached to the end of a stick or piece of dowel together with a battery. At the other end of the dowel another micro:bit can be attached, which is in charge of 'listening' for (receiving) detection and reporting the results. Following are two sample pieces of code that can get a system of this kind to work.

(a) Detecting (transmitting) micro:bit

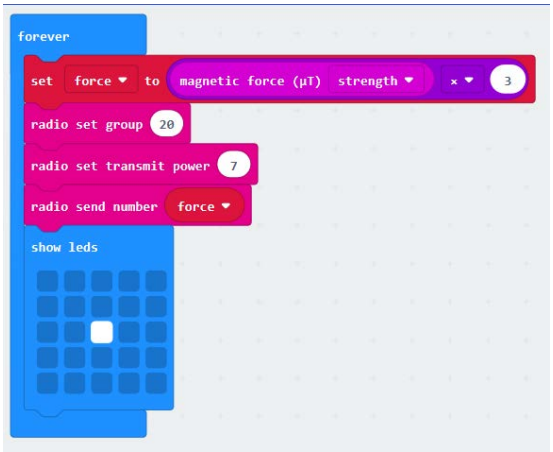


Figure 14

This sample code (Figure 14) uses a variable called **force** which is made by multiplying the detected magnetic force strength by 3 (any more and it becomes too sensitive to work properly).

The code chooses a radio channel to transmit on (20) as well as a transmission strength (7 – which is maximum). Note: If you have more than a few students, they will each need to be assigned different radio channels
(1–255).

The code also shows an LED just to let the user know it is working. The code will continually send the **force** value over channel 20 or whichever channel is assigned.

(b) Reporting (receiving) micro:bit

The reporting micro:bit needs to choose the same channel as the transmitting micro:bit (in Figure 15a this number is 20).

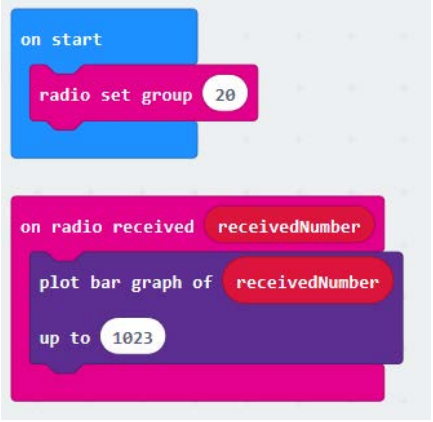


Figure 15a (top) and 15b (bottom)

It will receive the **force** from the transmitting micro:bit as a number it calls **receivedNumber**.

We can plot that **receivedNumber** in just the same way as we plotted **force** in the single micro:bit system we used earlier (Figure 15b).

Note: Due to the proximity of the micro:bits to each other on the metal detector, the radio set transmit power (see pink code block, fourth line in Figure 14) does not need to be set to 7 (the maximum). A setting of 1 or 2 should be more than adequate.

Further Digital Technologies learning

The 'getting the micro:bits to talk with one another' extension activity allows for much more to be done with learning about networks and digital systems.

- **Data collection and interpretation**

  Focus question: How do components of digital systems interact with each other to transmit data?
- **Digital systems**

  Focus question: How do the components of digital systems connect together to form networks?

Links to the Australian Curriculum

Table 1 gives teachers an opportunity to see related aspects of the Australian Curriculum: Digital Technologies which may be addressed depending upon the task. Teachers could discuss linked ideas in Mathematics (plotting, multiplication, comparison and logical operators), science (magnetism, metals and non-metals), Literacy, Numeracy and ICT Capability.

Table 1: Links to the Australian Curriculum: Digital Technologies 5–6

<table>
<tr><td>Digital Technologies
Achievement standard</td><td colspan="3">By the end of Year 6, students explain the fundamentals of digital system components (hardware, software and networks) and how digital systems are connected to form networks. They explain how digital systems use whole numbers as a basis for representing a variety of data types.

Students define problems in terms of data and functional requirements and design solutions by developing algorithms to address the problems. They incorporate decision-making, repetition and user interface design into their designs and implement their digital solutions, including a visual program. They explain how information systems and their solutions meet needs and consider sustainability. Students manage the creation and communication of ideas and information in collaborative digital projects using validated data and agreed protocols.</td></tr>
<tr><td>Strands</td><td colspan="3">Digital Technologies knowledge and understanding
• Digital systems
Digital Technologies processes and production skills
• Collecting, managing and analysing data
• Creating designed solutions by
– Investigating and defining
– Generating and designing
– Producing and implementing</td></tr>
<tr><td>Content descriptions</td><td colspan="3">• Examine the main components of common digital systems and how they may connect together to form networks to transmit data (ACTDIK014)
• Acquire, store and validate different types of data, and use a range of software to interpret and visualise data to create information (ACTDIP016)
• Define problems in terms of data and functional requirements drawing on previously solved problems (ACTDIP017)
• Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input (ACTDIP020)</td></tr>
<tr><td>Key concepts</td><td>• abstraction
• data collection
• data interpretation
• specification
• algorithms
• implementation
• digital systems</td><td>Key ideas</td><td>Thinking in Technologies
• computational thinking
• systems thinking
• design thinking</td></tr>
<tr><td>Cross-curriculum priorities</td><td></td><td>General capabilities</td><td>• Information and Communication Technology (ICT) Capability
• Literacy
• Numeracy</td></tr>
</table>

Useful links
Find out more about the micro:bit at www.microbit.org or code the micro:bit at www.makecode.org.

Disclaimer: ACARA does not endorse any product or make any representations as to the quality of such products. This resource is indicative only. Any product that uses material published on the ACARA website should not be taken to be affiliated with ACARA or have the sponsorship or approval of ACARA. It is up to each person to make their own assessment of the product, taking into account matters including the degree to which the materials align with the content descriptions and achievement standards of the Australian Curriculum. The Creative Commons licence BY 4.0 does not apply to any trademark-protected material.

All images in this resource are used with permission