



How can we use technology to monitor the wellbeing of dairy cows?

Context:

This activity builds on the idea of recycling slurry into mats for cows to sleep on. Learners consider how they can use the micro:bit to monitor whether the mats are improving the quality of the cow's sleep - making a fitbit for cows!

Learners will consider the different environmental conditions and activities that can be monitored in the dairy barn using technology.

Engineering focus:

Learners will be thinking the place of technology in improving engineering solutions to problems on a dairy farm. They will be writing computer programmes to automatically monitor what is happening in the dairy barn.

Learning time:

1.5 hours

Suggested age group:

9-11 years old

Keywords

automation
sensors
Micro:bit
program
code
debug
monitoring systems
sensor
variable
position sensor

Curriculum links:

Computing:

- Designing, writing and debugging programs that accomplish specific goals, including controlling or simulating physical systems;
- Using sequence, selection, and repetition in programs; work with variables and various forms of input

Resources:

- Computer with access to the internet for MakeCode
- Micro:bits or Micro:bit simulator at MakeCode

Note - this lesson uses micro:bits. However if you don't have micro:bits, you can still complete this lesson using the online simulator at <https://makecode.microbit.org/>

DISCLAIMER: These teacher notes were written by The University of Manchester's Science & Engineering Education Research and Innovation Hub. The Engineering Educates Challenge and the author(s) are not liable for the actions or activity of any persons who uses this resource or in any of the suggested further resources. We assume no liability with regard to injuries or damage to property that may occur as a result of using this information. These activities are designed to be carried out by children working with an adult. The adult is fully responsible for ensuring the activity is carried out safely. You can access further H&S advice from www.cleapss.org.uk.

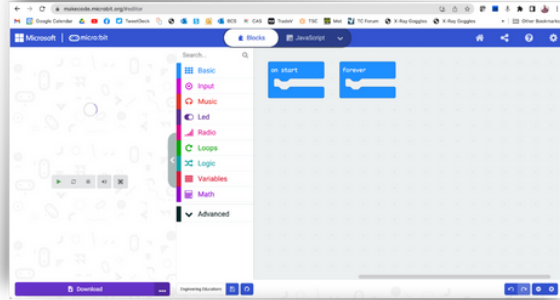
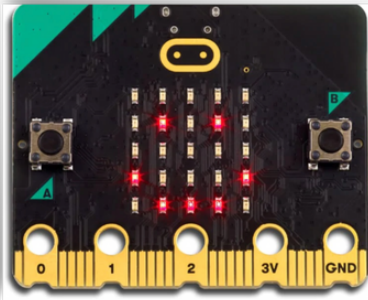


What is a Micro:bit and the MakeCode editor?

Display images of a micro:bit and the MakeCode editor, such as those shown below. Lead a discussion with pupils:

- Have pupils used the micro:bit before? If so, what did they create and how did they create this?
- Have pupils used the MakeCode editor before? What does the MakeCode editor remind pupils of? Why do they think this? (Make code looks and functions in a similar manner to Scratch - it is a graphical programming language)
- If pupils haven't used MakeCode before, but have used Scratch, lead a discussion on what they have previously created in Scratch. How did they create and run these programs?

1



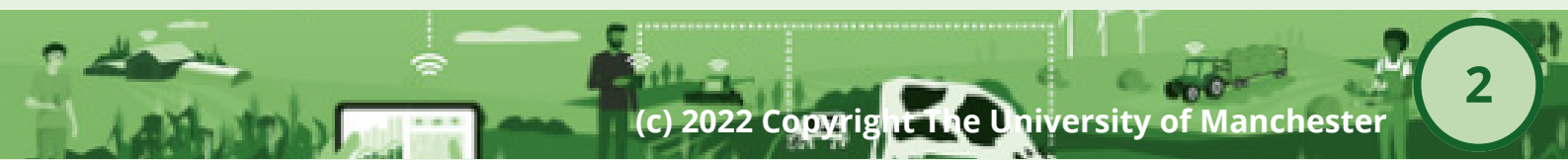
How could a Micro:bit help with the health and wellbeing of cows?

Display the picture to the right and remind pupils they've talked about the idea of recycling slurry into mats for cows to sleep on. Explain that in this project we will use the micro:bit to gauge whether the mats are improving the quality of cow's sleep. They will program them to be worn by the cow at night - like a cow fitbit!



Lead a discussion with pupils. How might we tell if the cows are sleeping better? Share a selection of pupils' thoughts and arrive at the idea that we could check to see how often the cows move when they're asleep. Cows that are tossing and turning less should be getting a better night's sleep. We want them to be moo-ving less!

2





How can we use code to count and store the number of times a cow moves?

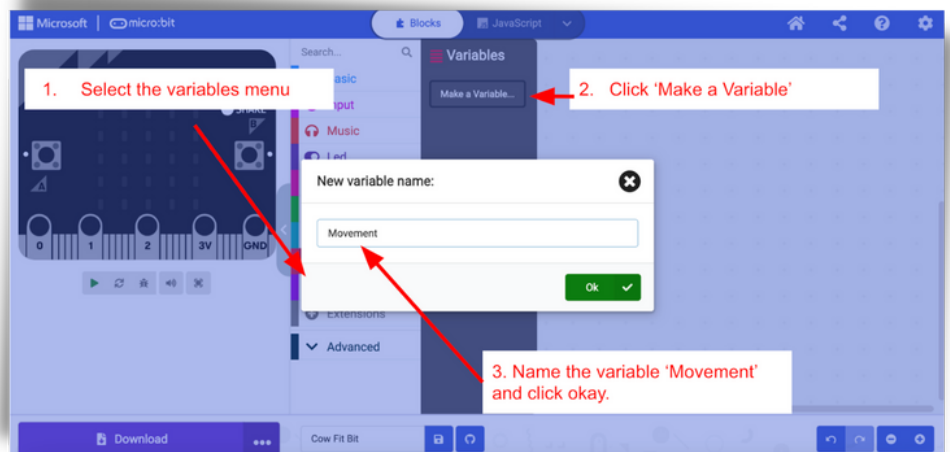
Based on the previous discussion, summarise the challenge we will be tackling with the micro:bits: **Program the micro:bit to act as a cow fitbit which automatically records how many times a cow moves when it is sleeping.**

Do pupils have any ideas on how they might tackle this challenge? Do they know of any programming concepts which might help here? Which features of the micro:bit might we use to help tackle this challenge? Share pupils' ideas to elicit prior understanding and lead into the following concept where possible.

Variables: Conclude with pupils a key feature of our code will be the need for counting and storing the number of times the cow moves. Explain that **variables** are used to store values in programs. Have pupils used variables before? A common use of variables are in computer games where they are used to store and display a player's score.

3

Explain that we will explore some code which uses a variable to better understand how they work. Display the following code to pupils. In this code a variable has been created called 'Movement'. In Make Code, commands relating to variables are red.



We can imagine our 'Movement' variable as a mini-whiteboard. We can write information on this mini-whiteboard to store it. We can also change the information stored on our whiteboard at any point as our program runs - we simply wipe it off and replace it with new information.

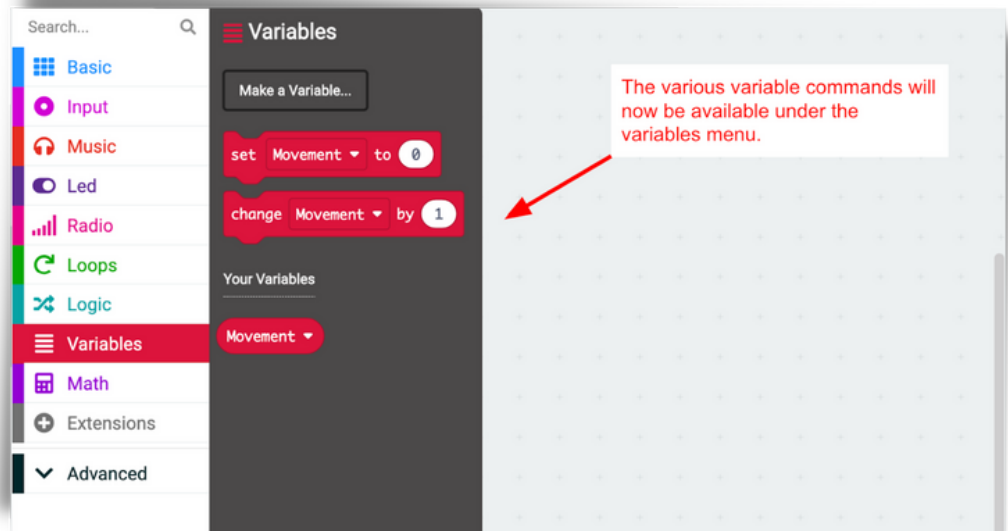


How can we use code to count and store the number of times a cow moves? (cont.)

Looking at the code. Can pupils predict:

- What the value of 'Movement' will be when the program is run (on start)?
- What happens to the value of 'Movement' when A is pressed?
- What happens to the value of 'Movement' when B is pressed?
- What happens when A+B is pressed?

Ask pupils to recreate the code in the Make Code editor, to run this and explore pushing the A and B buttons (and A+B) to see if their predictions are correct. To recreate this code, pupils will have to create the 'Movement' variable as shown below.



3

Answers to what happens when the code is run are as follows:

- When the program is run the value of the 'Movement' variable is set to 0.
- When the A button is pressed the value of the 'Movement' variable is increased by 1.
- When the B button is pressed the value of the 'Movement' variable is reduced by 1.
- When buttons A and B are pressed together, the value of the 'Movement' variable is displayed on the LED display.

Once pupils have had sufficient time to explore the code, lead a discussion on how and why a variable would be useful in our challenge? I.e. It could be used to store the number of times the cow moves.

4



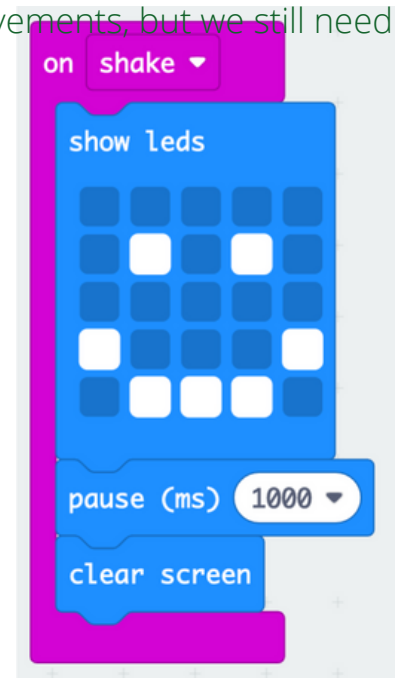
How can we use a Micro:bit to sense movement?

Remind pupils of our challenge: **Program the micro:bit to act as a cow fitbit which automatically records how many times a cow moves when it is sleeping.**

Recap with pupils that a variable, such as the 'Movement' variable we have just been exploring, could be used to record the cow's movements, but we still need a way of the micro:bit sensing these movements. Explain that we will now explore the 'On shake' command, as this might be able to help us.

Show pupils the following code. What do they predict it will do when run? Why?

Give pupils time to add this code to their micro:bit, to run it and explore what happens when the micro:bit is shaken. When shaken, the micro:bit will display a face for 1 second and then go blank. I.e. the micro:bit is detecting when it is moved and triggering something.



4

The commands and concepts pupils have explored in this resource up to this point should now be sufficient to solve our overall challenge. As such, give pupils time to discuss with a partner how they might combine the use of the 'On shake' command along with a variable to record the number of times the cow moves. Share a selection of pupils' ideas. Lead a discussion to arrive at the general idea:

If the micro:bit is attached to the cow, we could use the 'On shake' command to detect when the cow is moving. Each time a new movement is detected, this could be stored in a variable. We could call the variable 'Movement' so it is clear what it is recording.



Parson's Problem

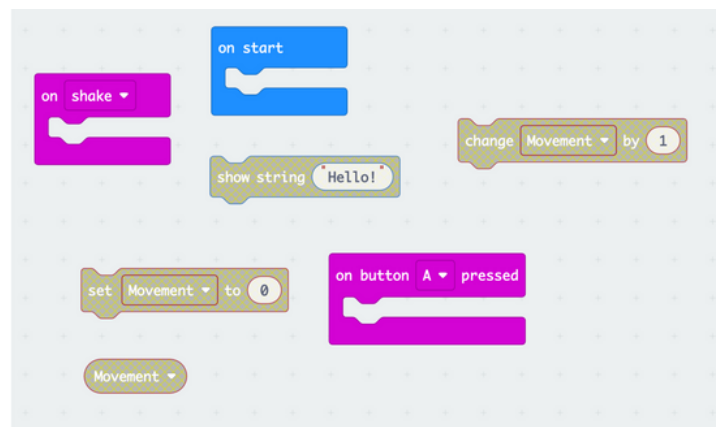
The code to now complete the overall challenge is contained in the Parson's Problem below. A Parson's problem scaffolds programming tasks by providing pupils all the code they need to complete a problem but doesn't show how the code should be combined.

Give pupils time to complete the Parson's Problem by adding the code to their Make Code project and combining the blocks of code to tackle the challenge. Pupils will have to create a 'Movement' variable for their Parson's Problem code as they did previously.

5

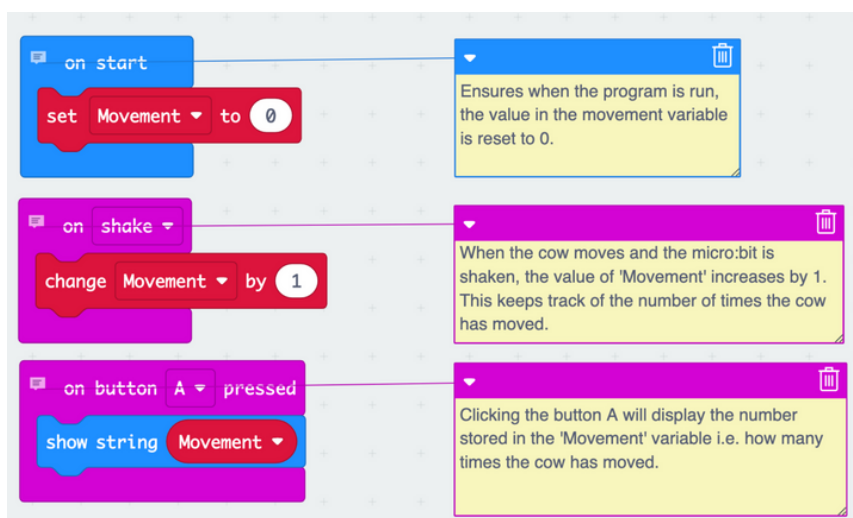
Pupils should test their programs - they could do this by strapping the micro:bits to themselves and pretending to be a sleeping cow with the occasional toss and turn!

Pupils should test their programs - they could do this by strapping the micro:bits to themselves and pretending to be a sleeping cow with the occasional toss and turn!



Code solution including comments:

Pupils should test their programs - they could do this by strapping the micro:bits to themselves and pretending to be a sleeping cow with the occasional toss and turn!





How effective is the programme at monitoring cow activity?

Lead a discussion with pupils to evaluate the success of their program. Based on their testing (as they took on the role of sleeping cows!) did their program:

- Automatically log each time the cow moved?
- Store the number of times the cow moved in the variable?
- Display the number of times the cow moved when the A button is pressed?

6

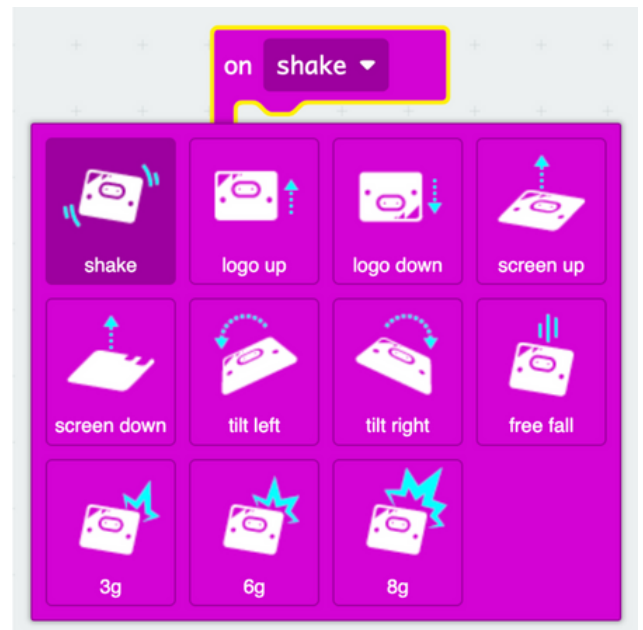
Discuss with pupils whether they encountered any bugs in their program - what were these and how did they remove them?

Whilst this activity has focused on the development of the micro:bit as a tool for sensing a cow's movement when sleeping, a discussion could now also be had on the scientific enquiry process of now testing if the mat is effective at improving a cow's sleep. What type of test would this be and why? How can we evaluate the effectiveness of the enquiry?

Take it further:

Pupils might find that the micro:bit doesn't log every movement, particularly more gentle movements as the micro:bit doesn't class the movement as a 'shake'.

Pupils could explore the variety of options within the 'On shake' command (shown right) to see if others provide a more accurate solution? Are there any potential draw backs of using other options? - such as the required orientation of the micro:bit.



Further links:

- Micro:bit Education Foundation <https://microbit.org/>
- Introducing inputs including shake <https://microbit.org/projects/make-it-code-it/get-silly/>