Remote Sensing - Fireflies

Year level band: Years 5 - 6 *and/or* 7 - 8

Description: This lesson is focused on data collection and remote sensing using the radio blocks of the BBC Micro:bit. First, we will build up the student's radio block skills using an adaption of the fireflies activity. Then students will learn how to send numbers using radio blocks. Lastly, the student will use their soil moisture sensors, plus two built-in sensors on the Micro:bit, to remotely collect measurements on light, temperature and soil moisture. This activity is best run over two lessons and would take about two hours to complete. This lesson can follow on from a previous lesson in which students learned how to measure soil moisture and control a 'pump'.

This lesson can be used as a follow-on activity for learning how to use the BBC Micro:bit in a smart garden. Much of it can be done as a simulation on the screen. Once you have learned how to measure soil moisture and how to control a pump, the next stage is to use the radio blocks to send your measurements from the garden into the classroom. Then you can collect and analyse data and conduct experiments with your garden. You can also use radio blocks to communicate with a Micro:bit near the tap to turn the water on and off (a more advanced lesson for students to perhaps explore themselves later). This lesson uses the radio blocks of the BBC Micro:bit in a fun way to learn how to communicate from one Micro:bit to another.

Resources:

- BBC Micro:bits with batteries.

  ADDITIONAL ITEMS IF WISHING TO INCORPORATE IN A SMART GARDEN

- "Grove Inventor Kit" for micro:bit or parts purchased separately. https://goo.gl/F6xCxC
- Grove moisture sensor https://goo.gl/pMkfMd

  Alternatively, using the Micro:bits kit with:

- 2 x Crocodile clips with cables
- Two long nails or screws

Prior Student Learning: Some understanding of how a computer gives and receives instructions would be helpful but not necessary in the first instance. An introduction to visual language programming such as Scratch or Blockly would be helpful too. It would also be helpful for the students to have completed the earlier challenge of measuring soil moisture.

A review on how Fireflies seem to synchronise from the following website; https://ncase.me/fireflies/ would allow students to more easily define the challenge of this activity They might ask how remote sensing might assist in developing solutions for exploring a smart garden watering challenge. e.g. How could remote sensing trigger a tap to be turned on?

Ideas for this activity have been developed with support from Bob Elliott an educator with Curriculum Services at the DoE in Tasmania. His website is used here to assist teachers and students in their understanding of remote sensing. http://mbfireflies.strikingly.com/

Summary

This project involves students learning about user-interface design and the importance of considering the user when designing digital solutions, and that user-interface design is about considering the user's interaction with a solution and how to meet those needs. In doing so, students develop skills in designing a solution for a user with specific needs, and being able to

communicate their design intentions - with sketch designs, as well as verbally by sharing their designs with peers.

By reflecting on their own designs, as well as other designs, students develop skills in being able to evaluate designs and provide constructive feedback.

Year	**Content Descriptors**
5 - 6	Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input (ACTDIP020)
	Explain how student solutions and existing information systems are sustainable and meet current and future local community needs (ACTDIP021)
7-8	Define and decompose real-world problems taking into account functional requirements and economic, environmental, social, technical and usability constraints (ACTDIP027) *Note that this lesson can also be done using general-purpose programming language providing an extension to the relevant content descriptor.

Element	**Summary of tasks**
Learning hook	A good introduction for students on what the task they are going to work on would be the following video which demonstrates how fireflies interact: https://youtu.be/ZGvtnE1Wy6U Four questions to ask the students might be: 1. What is a firefly? 2. What patterns do they see in the video? 3. How do the fireflies coordinate in nature? 4. How might the Micro:bits in this video communicate/coordinate with one another? https://www.youtube.com/watch?v=kPW_lf0P7XU
Achievement Standards	Years 5 - 6 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. Years 5 - 6 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. Years 7 - 8 By the end of Year 8, students distinguish between different types of networks and defined purposes. They explain how text, image and audio data can be represented, secured and presented in digital systems.

Learning input	The teacher works with the students to talk about the problems associated with looking after a garden They might like to investigate how remote sensing between devices can assist in managing a garden. Identify the Problem: How do Fireflies seem to synchronise their flashing lights and how does this relate to a smart garden challenge? What are the functions I wish my smart garden to have? Introduce the design cycle: Empathise, Define, Ideate, Prototype and Test. (Repeat until solution found) *Empathy phase* How might the communications between the micro:bits assist an understanding with managing a smart garden. *Systems Thinking* Would remote sensing using radio connections assist me in managing my garden? *Refine the problem*: How do I manage a remote sensing program to water my garden? *Design Thinking;* Come up with ideas (Ideation) to solve the problem. *Computational Thinking*. ● Decomposition - look first at the Fireflies solution ● Abstraction - Ignore other factors for now ● Algorithms - Design sequences of instructions for the solution. ● Data Representation - What data is needed and how do we represent the data? Introduce the micro:bit and its capabilities. Have students work through tutorial https://makecode.micro:bit.org/, if new to this device. For the more advanced students who wish to move ahead, have them explore the possibilities of remote sensing
Learning construction	The class are introduced to the micro:bit and the two ways they might be programmed either in a visual (5 -6) or text-based language (years 7-10) Students research the problem as commenced in the learning input discussions. They review how in nature there are a number of systems that seem to be ordered, eg the flashing light of a firefly but in essence there is a scientific answer to these questions. The introductory sessions to this problem could take two to three lessons until students feel confident with the visual programming interface. This may be unnecessary for those students already familiar with an earlier micro:bit activity. If you are a good coder, you can go straight to the main website: https://makecode.microbit.org/projects/fireflies Otherwise, you can follow the step below to make a simpler version. This code has both the sender and receiver in the same program, and each student can make the code and put in on their own Micro:bit, but they will still all (hopefully- fingers crossed) trigger together. Follow instructions in micro:bit tutorial (below). http://mbfireflies.strikingly.com/ which contains the code for years 5-6. For

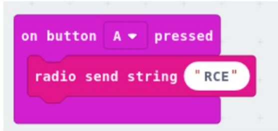
years 7-10, discuss the sequence of instructions using the visual code as template and then implement in text-based.

Getting Started with Remote Sensing: Step by step tutorial;

Who sent it?

Shouting in a crowded room

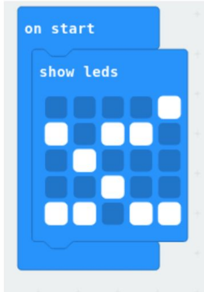
You might have noticed that letters were popping up, even when your partner was not pushing the button. This is because all the Micro:bits in the room are on the same channel. When one person presses a button, it gets sent to all the Micro:bits. To find out who is sending, put your initials into the radio-send-string (my initials are 'RCE', for instance).



Flashing Micro:bit

Create your firefly icon

Create a custom icon to display on your Micro:bit.



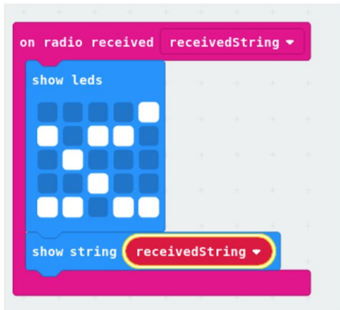
Flashing together

Trigger the fireflies with buttons

Replace the show-string block inside the on-radio-received with your custom firefly icon.

Now when someone pushes the A or B button, all the Micro:bits should flash together.

You can remove the show-string block, if you want.



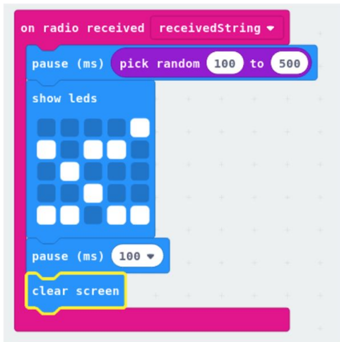
Flashing (almost) together

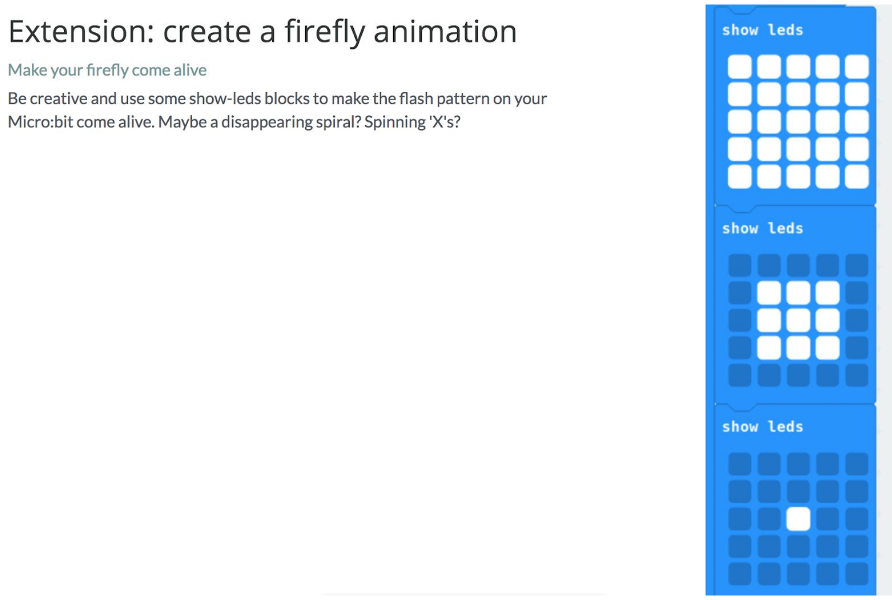
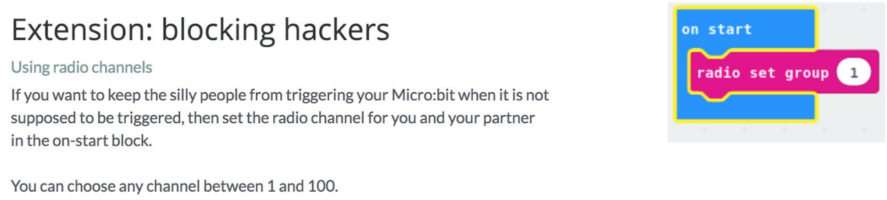
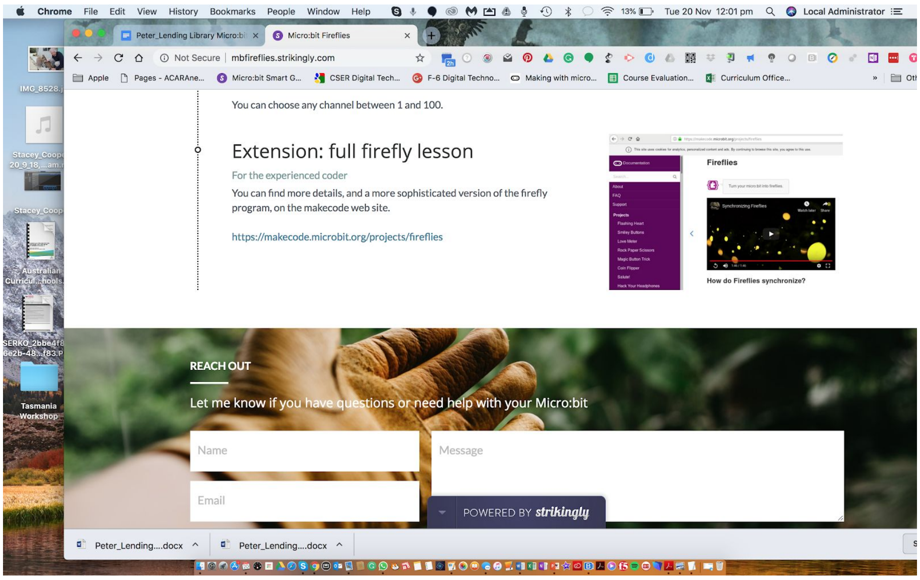
Random delay

Add a pick-random block (from the purple maths tab) to make the Micro:bits more like real fireflies. When the trigger signal is sent, each Micro:bit chooses it's own slightly different delay time.

100 = one tenth of a second
500 = one half of a second

You need the clear-screen block (under 'basic' 'more' tab) to reset the display.
You need the pause block to allow the LEDs time to be visible.



	   *with Permission of Bob Elliott* What next? After you do the firefly activities below, you can go onto the next lesson: remote sensing with Micro:bits
Learning demo	Students having reviewed the Firefly website https://ncase.me/fireflies/ discuss what they have learnt and how the solution might apply to the use of a digital technology to solve a problem.

	Students are encouraged to explore their own solutions using their earlier understanding of micro:bits and their capacity for connectivity via a wireless connection. Those who have mastered the capabilities of the micro:bit will be able to then plan how to incorporate the radio block on the makecode website. A follow-up challenge for the students would be to show them some commercial examples of remote sensing in agriculture.
Learning reflection	Students reflect on their initial understanding of the challenge and how their final solution using the example from the https://ncase.me/fireflies/ website has provided opportunities to further explore ways where remote sensing might assist in the managing of a smart garden.

Assessment:

In this lesson, teachers could **collect** evidence of learning and progression by the form of their artefacts, including **design** documents, **presentation** feedback, presentation recordings, photos of the final product and development stages.

	Quantity of knowledge			**Quality of understanding**	
Criteria	**Pre-structural**	**Uni-structural**	**Multi-structural**	**Relational**	**Extended abstract**
User-interface design	Standard controller - no consideration of the user.	A simple controller that explains general design considerations, but not necessarily unique to their user.	A controller with consideration made toward the user, as explained through a feature.	A controller that has considered the user through two or more design features supported by justification.	A controller that has addressed multiple user needs, with multiple features, and has a high level of complexity and justification for design features.
Design	No design used.	Basic design with no features identified.	Basic design with some features identified, but not linked to design justification.	Detailed design with numerous features identified and linked to design justifications.	Detailed design that brings in prior learning and/or independent learning beyond the task and possibly includes requirements, specifications, constraint factors.
Language	When describing their interface, no specific vocabulary is used.	The terms 'controller'' may be used as a general description.	The terms user-interface is used as a general description.	The terms user-interface is used confidently with specific reference to learner's work.	Specific vocabulary like 'requirements', 'specifications' and 'constraints' is used, going beyond the set language.

Teacher/Student Instructions:

Introduction to the micro:bit https://www.youtube.com/watch?v=Wuza5WXiMkc

https://makecode.micro:bit.org/

https://makecode.microbit.org/projects/fireflies

CSER Professional Learning:

This lesson plan corresponds to professional learning in the following CSER Digital Technologies MOOCs:

F-6 Digital Technologies: Foundations

- Unit 4: Digital Systems
- Unit 6: Algorithms and Visual Programming

See: http://csermoocs.adelaide.edu.au/moocs

Further Resources:

- Bob Elliott: tutorials on building smart gardens with the BBC micro:bit http://smartgarden.strikingly.com/
- Bob Elliott . Tutorial on remote sensing with micro:bits
- http://mbfireflies.strikingly.com/
- A useful web page on how fireflies syncronise to provide a background for teaching https://ncase.me/fireflies/
- micro:bit tutorials and projects
 https://makecode.micro:bit.org/

Author: Peter Lelong