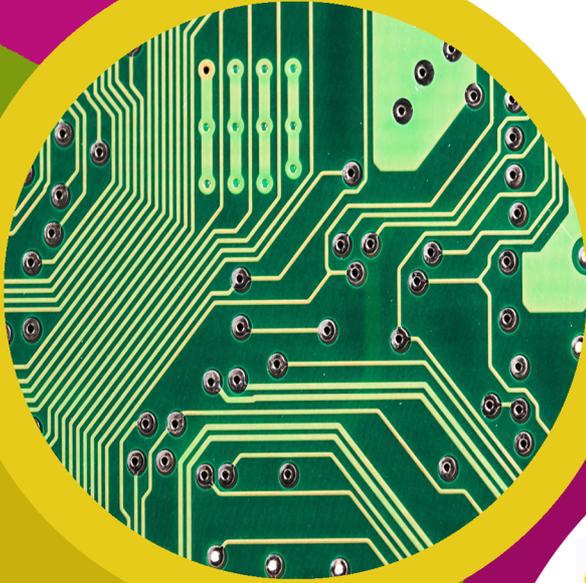
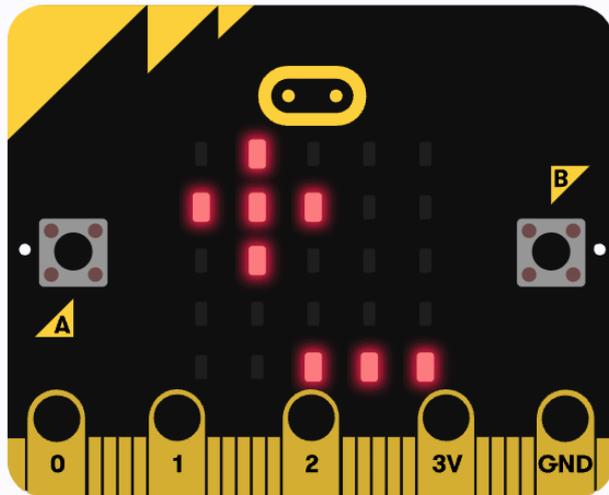


# technocamps

## Electrifying micro:bits Teacher Guidance



01010100  
01100101011  
0001101101000  
01101110011011  
1011000110110  
000101101101  
0111000001  
110011



## Links to Science and Technology AoLE

### **Computation:**

**(PS2)** I can create simple algorithms and am beginning to explain errors.

**(PS3)** I can use conditional statements to add control and decision-making to algorithms.

**(PS3)** I can use sensors and actuators in systems that gather and process data about the systems' environment.

### **Forces and Energy :**

**(PS2)** I can explore and describe the properties of materials and justify their uses.

**(PS2)** I can investigate different forms of energy and how it can be transferred

## The Four Purposes and Cross-Curricular Skills

This resource provides **Critical Thinking and Problem-Solving** opportunities. Students are required to understand and design an algorithm using block-based programming. They are able to analyse errors in the code, identify solutions, and deduce the next steps in the code.

Learners will also use **Creativity and Innovation** as they are encouraged to discuss and implement strategies to improve their program. They are taught about electrical currents and electron transfer, and are given the freedom to explore which materials do and do not conduct electricity.

The **Data and Computational Thinking** section of the **DCF** applies to this resource. Students will learn to code using loops, events, and conditionals to create an algorithm that takes input and displays an appropriate output depending on the electrical conductivity of the input.

## Why Is Learning This Important?

This resource provides learners with the opportunity to create simple algorithms with a demonstrable application, using a block-based programming language. It introduces concepts such as conditionals, loops, and event-based programming which are critical to most common programming languages.

This resource also explores interactions between materials at a molecular level, giving a basic introduction to atoms and electrons.

This resource can be expanded to introduce students to text-based programming such as Python.

## Suggested Approaches Key

In this suggested approach we use the following colours to differentiate the types of activities:

- **Yellow - Explain.** Teachers should explain the slide/example to the class.
- **Green - Discuss.** Teachers should start an open discussion with the class to get them to feedback some answers/ideas.
- **Purple - Activity.** Students are expected to complete an activity whether it be in their workbooks or on the computer, followed by a discussion of their solutions.
- **Green - Introduction/Conclusion.** The introduction/conclusion is also colour coded green. Teachers should hand out materials in the introduction and conclude the session and collect materials at the end.

### Introduction

Begin with introductions, and a brief explanation of the Technocamps programme, before handing out any resources required by learners and any additional aids for learners with additional learning needs.

### Explain: Topics Covered Today

We will be learning about electrical currents and conductivity, and how we can use micro:bits and programming to create a conductivity checker.

## Discuss: Micro:bit

Provide each student with a micro:bit and ask them what they know about it. Have they used them before? Can they tell you what some of the components are (e.g. buttons, LED lights, USB connector)?

## Explain: What is micro:bit?

The micro:bit is a very small computer that is used to teach how hardware and software work together.

It has several components: 25 Led lights that can be used to display images, sensors that can detect light/temperature/movement, buttons, and radio and bluetooth antenna.

We can program the micro:bit to take input, display output, process information, communicate with other micro:bits and many more things.

## Explain: What is programming?

Programming is telling a computer what to do using a set of ordered instructions. The set of ordered instructions is called an **algorithm**. The language used to tell the computer what to do is called a **programming language**.

Introduce the students to the MakeCode editor and explain how to connect their micro:bit devices.

## Explain: Downloading Programs

Each time you make changes to your program you need to click on the download button before the micro:bit can run the program.

## Activity: Hearty Start

Program the micro:bit to display a heart icon. Drag the **show icon** block into the **forever** loop. The heart is the default icon.

## Explain: Loops

Loops can be used to repeat commands in a program without typing out each action every time. They can be repeated forever, for a certain number of times, or for a given condition.

In MakeCode, these are found in the **Loops** section.

For a loop to have a purpose, an action command needs to be placed inside it.

All MakeCode programs start with a default **forever** loop. This loop will run a set of commands until the micro:bit is unplugged or reset. You can only have one **forever** loop in a micro:bit code.

## Discuss: Forever vs. On Start

Explain the difference between the **on start** block and the **forever** loop. Ask the students what they think would happen if they put the code from the animation activity in the **on start** block instead of the **forever** block.

Answer: The code would run once and then stop without repeating.

## Activity: Heartbeat

Ask the students to create a program that displays two alternating images - the heart icon and the miniature heart icon. They can do this in the default **forever** loop.

Suggest that they add a **pause** command between the two icons.

## Explain: Events

Each micro:bit has two buttons: A on the left and B on the right. These buttons allow us to choose which action to take without reprogramming the micro:bit each time.

For example, we can show a happy face when we press button A and a sad face when we press button B.

These commands are found in the **Input** section.

## Activity: An Eventful Button Press

Create a program that displays a different face when each button is pressed.

## Explain: micro:bit Sensors

The micro:bit has several sensors in it. It can sense the light level, temperature, sound, and movement. We can program the micro:bit to plot a graph of the current light level in the room.

## Activity: Light Sensor

Ask the students to create a light level graph with the micro:bit. Use the LED block **plot bar graph of** and the input block **light level**.

Test the code and see what happens.

## Discuss: Light Sensor

Discuss with the students how the micro:bit can measure light intensity.

## Explain: micro:bit Electrical Current

The sensor in the micro:bit generates an electrical current from light, like a solar panel does. The micro:bits can measure the strength of this current and give us a visual representation (like the light level sensor we already made).

## Explain: Electricity and Conductivity

Use the diagrams on the slides to give a simple explanation of electricity and electrons.

Electricity is the movement of electrons.

Many metals are excellent electrical conductors. Metal atoms have a free electron that can jump from atom to atom. By sharing electrons in this way, metal atoms allow for the movement of electrons. Applying a power source causes free electrons to all move in the same direction (current).

Some non-metals, such as water, are also very good conductors. Ask the students why they think this is.

Ions are special forms of atoms that have an electrical charge. They have either gained or lost an electron making them either negative or positive, respectively.

Water has salts (eg. NaCl) dissolved in it. When dissolved, the ions separate into  $\text{Na}^+$  and  $\text{Cl}^-$ . These ions are attracted to the cathode or anode, depending on their charge, causing them to move through the water.

Movement of ions = movement of charge = movement of electrons = electricity.

Ask the students about distilled/deionised water. Will this conduct electricity? (No).

## Explain: Logic and If-statements

Conditions help us perform different actions based on different conditions. These conditions should always result in Yes/No or True/False.

For example:

**If** your age is  $> 17$ , **then** you are eligible to drive.

**Else**, you have to wait until you are eligible.

Ask the students for more examples.

In MakeCode, the if-statements are found in the **Logic** section.

## Explain: Operators

We can create conditions using expressions. These expressions use mathematical operators. Talk through the operators that are on the slides with the students. ( $>$ ,  $<$ ,  $\leq$ ,  $\geq$ ,  $=$ ,  $\neq$ ).

## Activity: Conductivity Checker

Ask the students to create a program that checks whether an object or material conducts electricity.

Step-by step instructions are provided on the slides. As you go through the slides, ask the students if they can work out what the next step in the code will be.

1. Start by creating the conditional expressions. Drag the **if-else** block from the **logic** section into the **forever** block.
2. Drag the **pin P0 is pressed** block into the **if-else** block. Make sure P0 is selected, not P1, P2, etc.
3. Set the sound. Drag the **ring tone(Hz)** block from the **music** section into the if-statement. Choose a note from the **ring tone** block.
4. Drag the **stop all sounds** block into the **else**.
5. Set the icon. Drag two **show icon** blocks to your code - one under the **ring tone** block and one under the **stop all sounds**. Choose a different icon for each.
6. Attach crocodile clips to **Pin 0** and **GND** on the micro:bit and download the program.

Ask the students to test the conductivity checker. Touch the loose ends of the crocodile clips together. Test some other things to see if they are conducting (what about yourself?).

## Explain: micro:bit Radio

The micro:bit has a Bluetooth feature which allows them to communicate with each other. The commands for this are found in the **Radio** section in MakeCode.

The sender uses a radio group (eg. 10).

The sender sends the message using that radio number.

The message is transmitted to all micro:bits.

The receiver who is listening to the same radio group number receives that message.

There are two different commands to send messages: **radio send number** to send numbers and **radio send string** to send text messages.

## Extension Activity: Radio Communication

Show the code on the slides to the students and talk through all the blocks. This code allows the micro:bits to send messages to each other. The following slide shows how to code the receiving micro:bit.

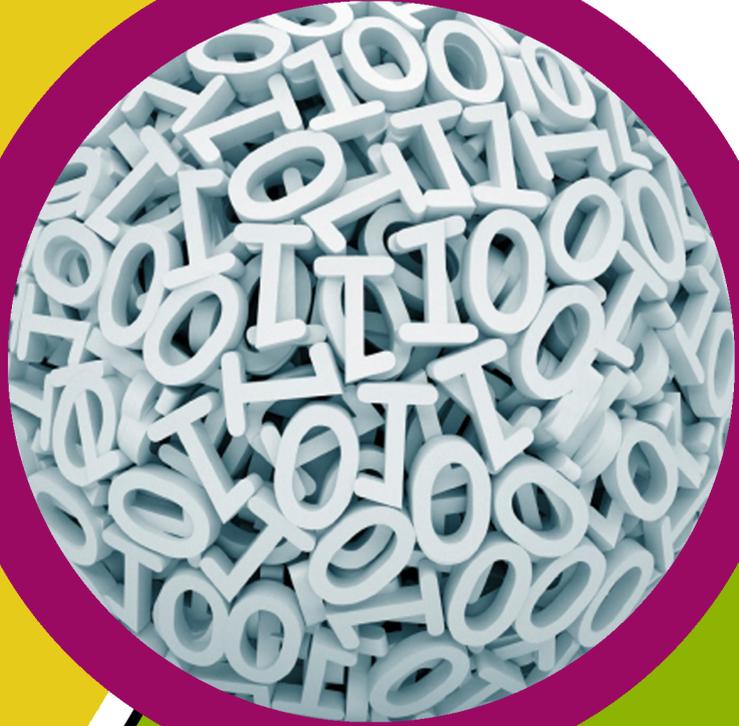
Have the students experiment with different commands and messages.

## Differentiating for Learners

- Extension tasks have been provided to challenge students to improve their code using radio communication. There is also extra information about conductivity, electron exchange, and ions that can be used for some classes.
- Some learner may need more guidance in assembling the code than others. Since this workshop is aimed at ages 9-11, there might be several learners with little coding experience so will require step-by-step instructions in assembling the blocks. Other learners can use their previous coding experience to write the algorithm without any code provision.
- The micro:bit website and editor allow for navigation using accessibility features such as a screenreader, or speech recognition software.

## Where To Go Next

- This workshop introduced light sensors. This can be expanded to explore the other micro:bit sensors such as temperature, movement, and sound.
- Micro:bit also has a user-friendly Python editor. This workshop can be adapted to introduce learners to a text-based programming language.
- The students can continue exploring radio communication which is introduced in the extension activity and try to incorporate this into other programs.



**technocamps**



@Technocamps



Find us on  
**Facebook**