

Electrical conductors

**Lessons:** 5

**Programming languages:** MakeCode

**Target age:** 7-11 yrs

**Subjects & topics:**

* Computational thinking: Flowcharts, Algorithms
* Sciences: Electricity
* Computer systems: Input/output
* Programming: Selection
* Design & technology: Electronics

# Unit of work summary

In this series of five lessons aimed at students aged 8-9 years, students develop their understanding of flowchart algorithms, selection and inputs and outputs by using electrical circuits and the BBC micro:bit to test the conductivity of different materials.

## Overall key learning

* can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
* have repeated practical experience of writing computer programs in order to solve problems
* can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
* are responsible, competent, confident and creative users of information and communication technology

## Additional skills

Creative thinking, collaboration, problem-solving, testing

## Lesson 1: Selection & conductivity investigation

In this ‘unplugged’ lesson, pupils construct electrical circuits and use selection to explain what happens to the output when conductors and insulators are added.

**Key learning:**

* To identify the output in an electrical circuit
* To understand the term selection
* To use selection when describing the output of an electrical circuit

## Lesson 2: Decision boxes

In this ‘unplugged’ lesson, pupils further develop their understanding of selection using decision boxes and creating flowchart algorithms.

**Key learning:**

* To understand how selection is represented in flowcharts
* To understand and use decision boxes
* To create flowcharts algorithms

## Lesson 3: Inputs

Pupils experiment with the inputs on the BBC micro:bit and use their knowledge of selection to record their findings in decision boxes.

**Key learning:**

* To review outputs
* To understand what inputs are
* To use tinkering to find inputs on the BBC micro:bit
* To represent selection with inputs using decision boxes

## Lesson 4: Making a conductivity tester

Pupils learn how to use the BBC micro:bit’s pins as inputs. They plan, write, test and debug MakeCode programs to use micro:bits to test the electrical conductivity of materials.

**Key learning:**

* To plan, write, test and debug programs
* To write programs that use selection
* To write programs that use inputs and output

## Lesson 5: Review & reflection

In this ‘unplugged’ lesson, pupils reflect on their learning in this unit and write an algorithm to test the electrical conductivity of materials using the BBC micro:bit.

**Key learning:**

* To decompose a problem into smaller steps
* To write a flowchart algorithm that uses selection
* To know and identify inputs and outputs

# Curriculum links

These lessons are mapped to the following learning objectives and standards for computing and science:

## England National Curriculum

#### KS2 computing curriculum

Curriculum aims:

* can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
* can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
* are responsible, competent, confident and creative users of information and communication technology

Students should be taught to:

* design, write and debug programs that accomplish specific goals
* solve problems by decomposing them into smaller parts
* use sequence, selection in programs; work with various forms of input and output
* use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs

[Read the full KS2 computing curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239033/PRIMARY_national_curriculum_-_Computing.pdf).

#### KS2 science curriculum

Electricity (Year 4 programme of study)

Students should be taught to:

* Recognise some common conductors and insulators

[Read the full KS2 science curriculum](https://www.gov.uk/government/publications/national-curriculum-in-england-science-programmes-of-study/national-curriculum-in-england-science-programmes-of-study).

#### KS2 DT curriculum

Students should be taught to:

* understand and use electrical systems in their products [for example, series circuits incorporating switches, bulbs, buzzers and motors
* apply their understanding of computing to program, monitor and control their products

[Read the full KS2 DT curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239041/PRIMARY_national_curriculum_-_Design_and_technology.pdf)

## Scotland Curriculum for Excellence

#### Technologies

* I understand the operation of a process and its outcome. I can structure related items of information (TCH 2-13a)
* I can explain core programming language concepts in appropriate technical language (TCH 2-14a)
* I can create, develop and evaluate computing solutions in response to a design challenge (TCH 2-15a)

[Read the full Curriculum for Excellence: technologies](https://education.gov.scot/Documents/Technologies-es-os.pdf)

#### Sciences

* I can describe an electrical circuit as a continuous loop of conducting materials. I can combine simple components in a series circuit to make a game or model (SCN 1-09a)
* I have used a range of electrical components to help to make a variety of circuits for differing purposes. I can represent my circuit using symbols and describe the transfer of energy around the circuit (SCN 2-09a)

[Read the full Curriculum for Excellence: sciences](https://www.education.gov.scot/Documents/sciences-eo.pdf)

## Northern Ireland Curriculum - Primary

#### Using ICT across the curriculum

Pupils should be taught to:

* explore - investigate, make predictions and solve problems through interaction with digital tools
* express - create information and multimedia products using a range of assets
* evaluate - talk about, review and make improvements to work, reflecting on the process and outcome

#### KS1 - suggested curriculum ideas for the world around us

* design and make simple models

#### KS2 - suggested curriculum ideas for the world around us

* how knowledge in science supports technological inventions
* the effect of adding components to circuits

#### KS1 - suggested curriculum ideas for science

* the use of electricity as an energy source and the importance of using it safely

#### KS2 - suggested curriculum ideas for science

* why materials are chosen for their use

[Read the full Northern Ireland curriculum - primary](https://ccea.org.uk/downloads/docs/ccea-asset/Curriculum/The%20Northern%20Ireland%20Curriculum%20-%20Primary.pdf)

#### KS1 & 2 - requirements for using ICT

* explore - investigate, make predictions and solve problems through interaction with digital tools

[Read the full KS1 & KS2 requirements for using ICT](https://ccea.org.uk/downloads/docs/ccea-asset/Curriculum/Curriculum%20Requirements%20for%20Using%20ICT.pdf)

#### Primary using ICT - desirable features - computational thinking and coding

**Level 4**

Pupils should:

* create a more sophisticated coding project using a broad range of commands; and/or
* solve a given problem using commands in a programming environment.

**Programmable devices (such as Parrot Drone, micro:bit or Sphere)**

* look at and talk about examples of coding projects, including the use of motion, looks, lights or sounds, sensors, control and events such as ‘if...then’ and ‘loop until’ (or equivalent) that make the code more efficient;
* recognise that these projects are composed of different components and break the task into smaller manageable tasks (decomposition);
* in small groups, plan and storyboard their own coding project, working out what different parts of the program must do, using logical reasoning to discuss and compare the commands that are required for their algorithm;
* use a range of commands to create a project including triggering commands such as ‘if...then’ and ‘loop until’ to facilitate a more efficient method of interaction;
* test and debug at regular intervals and collaborate with others to solve problems as they arise;

**Finally**

* share their work (possibly using digital tools), respond to feedback and comment on others’ work; and
* organise files and export work in an appropriate format so that others may view it.

[Read all Primary using ICT desirable features](https://ccea.org.uk/downloads/docs/ccea-asset/Curriculum/Primary%20Using%20ICT%20Desirable%20Features%20Update%202019.pdf)

## Curriculum for Wales

#### Science and technology

Progression step 3 - forces and energy provide a foundation for understanding our universe:

* I can describe the factors that affect electrical circuits and this will enable me to change variables and predict what will happen

Progression step 2 - computation is the foundation for our digital world:

* I can safely use a range of tools, materials and equipment to construct for a variety of reasons
* I can use computational thinking techniques, through unplugged or offline activities
* I can create simple algorithms and am beginning to explain errors
* I can follow algorithms to determine their purpose and predict outcomes
* I can follow instructions to build and control a physical device

Progression step 3 - computation is the foundation for our digital world:

* I can use conditional statements to add control and decision-making to algorithms
* I can explain and debug algorithms

[Read the full science and technology curriculum](https://hwb.gov.wales/curriculum-for-wales/science-and-technology/descriptions-of-learning/)

#### Digital competence framework

Progression step 1 - data and computational thinking - problem-solving and modelling:

* I can recognise and follow instructions in the appropriate order to perform a task.
* I can organise, select and use simple language to give instructions to others.
* I can control devices giving instructions.
* I can identify errors in simple sets of instructions (algorithm).

Progression step 2 - data and computational thinking - problem-solving and modelling:

* I can detect and correct mistakes which cause instructions (a solution) to fail (debug).
* I can create and record verbal, written and symbolic instructions to test ideas, e.g. the order of waking up through a diagram or flowchart.
* I can change instructions to achieve a different outcome.

Progression step 3 - data and computational thinking - problem-solving and modelling:

* I can understand the importance of the order of statements within algorithms.

Progression step 1 - producing - evaluating and improving digital content:

* I can comment on work in relation to a single success criterion.

Progression step 2 - producing - evaluating and improving digital content:

* I can give an opinion about my own work and suggest improvements based on the success criteria.

[Read the digital competence framework](https://hwb.gov.wales/curriculum-for-wales/cross-curricular-skills-frameworks/digital-competence-framework)

## USA Code.org

#### CS Fundamentals

Course C

Concepts included:

* flowchart algorithms
* sequencing
* events
* conditionals
* inputs/outputs.

[Read the full CS Fundamentals curriculum](https://code.org/educate/curriculum/elementary-school).

## USA CSTA Standards

#### Grades 3-5

1B-CS-01 - Describe how internal and external parts of computing devices function to form a system.

1B-CS-02 - Model how computer hardware and software work together as a system to accomplish tasks

1B-CS-03 - Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.

1B-AP-08 - Compare and refine multiple algorithms for the same task and determine which is the most appropriate

1B-AP-10 - Create programs that include sequences, events, loops, and conditionals.

1B-AP-12 - Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.

[Read the CSTA Standards in full](https://csteachers.org/k12standards/ ).

This content is published under a [Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)](https://creativecommons.org/licenses/by-sa/4.0/) licence.