

Musical micro:bit

**Lessons:** 5

**Programming languages:** MakeCode

**Target age:** 7-11 yrs

**Subjects & topics:**

* Computational thinking: Algorithms
* Programming: Iteration, Selection
* Computer systems: Input/output
* Music: Listening, Composition

# Unit of work summary

This series of five lessons is written for primary school students aged 9-10 years. Students compose musical phrases and write algorithms to play their phrases on pitched instruments (e.g. glockenspiels).

They then program the micro:bit to play their phrases when events are triggered and experiment with using the accelerometer.

Finally, they consider whether the micro:bit can be used as a music-making device, especially for those who might not have access to instruments.

## Overall key learning

* 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

## Additional skills

Creative thinking, collaboration, problem-solving, critical thinking, evaluation, researching, presenting

## Lesson 1: Musical algorithms

In this ‘unplugged’ lesson, pupils use a musical theme to extend their understanding of algorithms. They evaluate algorithms, create a short musical composition and write an algorithm for someone who cannot read music to follow.

**Key learning:**

* To read and interpret a range of algorithms
* To evaluate algorithms
* To write algorithms for a given audience

## Lesson 2: Programming & debugging music

Pupils explore using the BBC micro:bit to play musical phrases. They write programs using the MakeCode editor, extending their understanding of repetition and inputs from previous unit and connect speakers/headphones to their micro:bit to play their music.

**Key learning:**

* To use existing knowledge to improve programs
* To write and debug musical programs
* To experiment (tinker) with the micro:bit to make music

## Lesson 3: Musical gestures

Pupils extend their understanding of selection from previous units by following and modifying algorithms. They then consider how gestures are used by a conductor in an orchestra and write algorithms to instruct others which notes to play on a pitched instrument (i.e. glockenspiel) when certain gestures are made.

**Key learning:**

* To analyse and modify algorithms
* To identify patterns in algorithms
* To write algorithms using repetition and selection

## Lesson 4: Controlling music with inputs

Pupils develop their understanding of using the inputs and outputs on the BBC micro:bit and write, test and debug programs to experiment with using selection and the accelerometer to play different notes. They also consider how the micro:bit could be used to help people who find playing instruments difficult.

**Key learning:**

* To identify how inputs are used in programs
* To write programs that use inputs and selection
* To write and evaluate algorithms

## Lesson 5: Evaluating micro:bit music

Pupils recap and apply their learning from this unit by undertaking programming challenges. They then consider the information someone would need to make music using a BBC micro:bit, create slides for an explainer presentation and evaluate using the micro:bit for making music.

**Key learning:**

* To modify programs to meet given criteria
* To decompose learning from the unit
* To evaluate the micro:bit as a music-making device

# Curriculum links

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

## 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, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
* use sequence, selection and repetition 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 DT curriculum

* 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)

#### KS2 music curriculum

Students should be taught to:

* improvise and compose music for a range of purposes using the inter-related dimensions of music

[Read the full KS2 music curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239037/PRIMARY_national_curriculum_-_Music.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 understand how information is stored and how key components of computing technology connect and interact through networks. (TCH 2-14b)
* 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)

#### Expressive arts

* I can use my voice, musical instruments and music technology to discover and enjoy playing with sound, rhythm, pitch and dynamics.(EXA 1-17a)
* I can use my voice, musical instruments and music technology to experiment with sounds, pitch, melody, rhythm, timbre and dynamics. (EXA 2-17a)

[Read the full Curriculum for Excellence: expressive arts](https://education.gov.scot/Documents/expressive-arts-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

#### KS1 - music

* work creatively with sound by investigating, experimenting, selecting and combining sounds to express feelings, ideas, mood and atmosphere

#### KS2 - music

* work creatively with sound by creating musical stories, pictures, patterns, conversations, accompaniments and investigating ways of preserving the music they have created

[Read the full Northern Ireland curriculum - primary](https://ccea.org.uk/downloads/docs/ccea-asset/Curriculum/The%20Northern%20Ireland%20Curriculum%20-%20Primary.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 2 - design thinking and engineering offer technical and creative ways to meet society’s needs and wants:

* I can safely use a range of tools, materials and equipment to construct for a variety of reasons

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 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 identify repeating patterns and use loops to make my algorithms more concise
* 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/)

#### Expressive arts

Progression step 1 - exploring the expressive arts is essential to developing artistic skills and knowledge and it enables learners to become curious and creative individuals:

* I can explore and experiment with a variety of creative techniques, materials, processes, resources, tools and technologies

Progression step 2 - exploring the expressive arts is essential to developing artistic skills and knowledge and it enables learners to become curious and creative individuals:

* I can explore and experiment with and then select appropriate creative techniques, practices, materials, processes, resources, tools and technologies

[Read the full expressive arts curriculum](https://hwb.gov.wales/curriculum-for-wales/expressive-arts/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.
* I can identify repetitions or loops in a sequence, e.g. identify where to shorten a set of instructions by repeating steps, for instance when learning a new song.

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

* I can create and refine algorithms and flowcharts to solve problems, making use of features such as loops, Boolean values and formulae.
* 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

Courses E, F

Concepts included:

* algorithms & programs using conditionals
* 'for' loops

[Read the full Code.org 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-11 - Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.
* 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.