

Night safety

**Activities:** 3

**Programming languages:** MakeCode

**Target age:** 7-11 yrs, 11-14 yrs

**Subjects & topics:**

* Computational thinking: Algorithms
* Design & technology: Product design
* Computer systems: Sensors, Input/output
* Programming: Sequence
* Global Goals: 3 Health

# Design challenge summary

In the **Night sensor** activity students create a wearable device to give a visual and audio reminder when it is time to “Be Safe: Be Seen!’ at nightfall.

In the **Flashing wheels** project students design a prototype of a flashing wheel light to help improve road safety for a wheelchair user.

In the **Bag for Juliane** project students learn about Juliane, a girl from Zimbabwe, and create a light-up bag for her journey to school.

## Overall key learning

* to understand the issues around road safety for children at night
* to consider how technology can help children to ‘Be Safe: Be Seen!’
* to design, program, test and present working prototypes using the BBC micro:bit to help children 'Be Safe: Be Seen!'
* 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, prototyping, testing, presenting

## Activity 1: Night sensor

Students develop their understanding around road safety for children at night and explore potential solutions before planning, creating and testing a Night sensor using the BBC micro:bit.

**Key learning:**

* To understand the problem of road safety for children
* To explore ways technology can help children stay safe at night
* To plan, create and test a ‘Night sensor’ using the BBC micro:bit to remind children to ‘Be Safe: Be Seen!

## Activity 2: Flashing wheels

Students design, create and test a prototype of a flashing wheel light to help improve road safety at night for wheelchair users.

**Key learning:**

* To develop understanding of issues around road safety for wheelchair users, especially at night.
* To design and create a prototype of a flashing wheel light using micro:bit to help wheelchair users 'Be Safe: Be Seen!' at night.
* To test, debug, evaluate and present a Flashing wheel light prototype.

## Activity 3: A bag for Juliane

In this activity, students learn about Juliane, a girl from Zimbabwe who has come to England as a refugee and create a light up bag for her journey to school.

**Key learning:**

* To develop empathy and understanding for child refugees
* To design a bag to help Juliane, a child refugee, feel safer on her school journey
* To design and code a light for Juliane’s bag using micro:bit
* To create a bag featuring a micro:bit light and other features to help Juliane (if you have materials)

# Curriculum links

## 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 variables and 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
* select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information

[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).

#### KS3 computing curriculum

* design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
* understand several key algorithms that reflect computational thinking (for example, ones for sorting and searching); use logical reasoning to compare the utility of alternative algorithms for the same problem
* understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems

[Read the full KS3 Computing curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239067/SECONDARY_national_curriculum_-_Computing.pdf)

#### KS2 DT curriculum

* evaluate - evaluate their ideas and products against their own design criteria and consider the views of others to improve their work
* 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)

#### KS3 DT curriculum

* evaluate - investigate new and emerging technologies
* test, evaluate and refine their ideas and products against a specification, taking into account the views of intended users and other interested groups

[Read the full KS3 DT curriculum](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239089/SECONDARY_national_curriculum_-_Design_and_technology.pdf)

## Scotland Curriculum for Excellence

#### Technologies

* I can explore and experiment with digital technologies and can use what I learn to support and enhance my learning in different contexts (TCH 1-01a)
* I can explore and comment on processes in the world around me making use of core computational thinking concepts and can organise information in a logical way (TCH 1-13a)
* I understand the instructions of a visual programming language and can predict the outcome of a program written using the language (TCH 1-14a)
* I can demonstrate a range of basic problem solving skills by building simple programs to carry out a given task, using an appropriate language (TCH 1-15a)
* I can extend and enhance my knowledge of digital technologies to collect, analyse ideas, relevant information and organise these in an appropriate way (TCH 2-01a)
* 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)
* I can select appropriate development tools to design, build, evaluate and refine computing solutions based on requirements. (TCH 3-15a )

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

#### Health and well-being

* I know and can demonstrate how to travel safely (HWB 1-18a)
* I know and can demonstrate how to travel safely (HWB 2-18a)

[Read the full health and well-being curriculum](https://education.gov.scot/Documents/health-and-wellbeing-eo.pdf)

## Northern Ireland Curriculum

#### Primary

#### Using ICT across the curriculum

* explore - investigate, make predictions and solve problems through interaction with digital tools
* 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
* the importance of light in our everyday lives
* the life of a child in a contrasting location, including similarities and differences, such as, homes, schools, events and celebrations

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

* design and make simple models
* travelling to school at different times of the year and in different types of weather

[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
* evaluate - talk about, review and make improvements to work, reflecting on the process and outcome and consider the sources and resources used, including safety, reliability and acceptability

[Read the full KS1 & 2 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.

**Level 5**

Pupils should:

* create more sophisticated coding projects using a broad range of commands and more than one platform; and
* solve a more complex problem using commands in a programming environment.

**Programmable devices (such as Parrot Drone, MicroBit or Sphero)**

* as a class look at and talk about examples of coding projects, including using multiple ‘if...then’ and ‘if...then...else’ commands, variables, sensors, events, operators and comparators;
* recognise how they can decompose these projects;
* in small groups, plan their own coding project, demonstrating a clear sense of purpose and audience, showing understanding of abstraction by deciding what details they need to include and what they can leave out, working out what different parts of the program must do and using logical reasoning to discuss and compare the commands that are required for their algorithm and predicting the outcome;
* use a range of commands to create a project, including variables, operators and control statements such as ‘if... then...’ alongside the use of ’if...then...else’ and comparators;
* 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 the work of others evaluating process and outcome; and
* organise files and publish work online (if available) so that others can 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)

#### Secondary

#### Science and technology - technology and design - statutory requirements, KS3

* design – identifying problems; investigating, generating, developing, modelling and evaluating design proposals; giving consideration to form, function and safety;
* control – incorporate control systems, such as mechanical, electronic or computer-based, in products and understand how these can be employed to achieve desired effects
* explore technical inventions and designs that have met a social need cost-effectively.
* design cost effective and appropriate solutions to meet the specific needs of diverse local and global groups. Citizenship
* show deeper understanding by thinking critically and flexibly, solving problems and making informed decisions, using Mathematics and ICT where appropriate;

[Read the full technology and design statutory requirements](https://ccea.org.uk/downloads/docs/ccea-asset/General/Statutory%20Requirements%20for%20Technology%20and%20Design%20at%20Key%20Stage%203.pdf)

#### Digital skills curriculum

#### Become a digital maker at KS3

* Design a digital solution for a problem using an appropriate method;
* Build a solution based on their design using appropriate tools and techniques;
* Review or test the solution against their original plan;

#### Becoming a digital worker at KS3

* Use applications to create products with thought given to both the audience and the purpose through the use of digital design;
* Troubleshoot basic problems with their digital technology.

[Read the full digital skills curriculum](https://ccea.org.uk/learning-resources/digital-skills-hub/key-stage-3-digital-skills-curriculum)

## Curriculum for Wales

#### Science and technology

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.
* I can use sensors and actuators in systems that gather and process data about the systems’ environment.

Progression step 4 - computation is the foundation for our digital world:

* I can decompose given problems and select appropriate constructs to express solutions in a variety of environments.
* I can plan and implement test strategies to identify errors in programs.
* I can select and use multiple sensors and actuators that allow computer systems to interact with the world around them.

[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.
* 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 4 - data and computational thinking - problem-solving and modelling:

* I can create a simple model or self-contained algorithm.
* I can identify the different parts of an algorithm to determine their purpose.
* I can detect and correct errors in 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 C, D and E

Concepts included:

* programming in a team
* selection
* variables
* writing algorithms
* events
* loops
* sequencing

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

**CS Discoveries**

Units 1 and 6

Concepts included:

* problem solving
* inputs and outputs
* hardware
* sensors

[Read the full Code.org CS Discoveries curriculum](https://studio.code.org/courses/csd-2022/?redirect_warning=true)

## 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.
* 1B-AP-13 - Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.
* 1B-AP-14 - Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.
* 1B-AP-15 - Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.
* 1B-AP-17 - Describe choices made during program development using code comments, presentations, and demonstrations.
* 1B-IC-18 - Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.

#### Grades 6-8

* 2-CS-03 - Systematically identify and fix problems with computing devices and their components.
* 2-AP-10 - Use flowcharts and/or pseudocode to address complex problems as algorithms.
* 2-AP-17 - Systematically test and refine programs using a range of test cases.

[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.