**Flashing wheel light**

**Introduction**

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

**Time:** 60 minutes+

**Learning objectives**

* 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.

**Materials needed:** lesson slides, large sheets of paper, MakeCode editor, micro:bits, Flashing light MakeCode sample code.

**Road safety for wheelchair users (10 minutes)**

* Give groups or pairs of students large sheets of paper and ask them to consider the main problems around road safety for wheelchair users, especially at night **(slide 2).**
* Discuss students’ ideas then ask them to come up with ways they think travelling at night could be made safer for people who are wheelchair users.

**Flashing wheel lights (10 minutes)**

* Introduce the Flashing wheel lights project **(slide 3).**
* If appropriate, make the classroom dark and show students an example Flashing wheel light (a micro:bit with the Flashing wheel code running on it).
* Ask students to discuss different ways it could work (e.g. light always on, blinking rapidly, flashing more slowly etc).
* Go into appropriate depth about how it works for your students (e.g. how do they think the light level is detected, how does a user turn the light on and off, what makes it flash, how long does the flashing word etc).

**Algorithm writing (10 minutes+)**

* Ask students individually or in pairs to write an algorithm for a Flashing wheel light on another large sheet of paper **(slide 4).**
* Depending on your student’s level you may need to introduce or recap iteration, selection and variables.

**Programming a Flashing wheel light (15 minutes+)**

* Ask students to use their algorithm to program their Flashing wheel light **(slide 5).**
* You may need to talk students through the MakeCode editor and using iteration, selection and variables, depending on their experience. Example programs are given as files and on **slide 6.**
* If working in pairs, encourage them to use paired programming, work through problems systematically together and test and debug their work regularly.

**Evaluating and presenting (15 mins+)**

* Ask students to evaluate their work in simple ways (e.g. What went well/Even better if) **(slide 7).**
* Invite them to present their Flashing wheel light (if you can find wheels for students to attach their lights to, all the better) and what they have learnt from this project.

**Differentiation**

**Support**

* You could give out instructions to sequence to help students create their algorithm and printed versions of the blocks to sequence before coding.

**Stretch & challenge**

* Students can be challenged to consider the most efficient way of writing their program and to explain why with comments. They could also use one of the other editors (e.g. python).

**Opportunities for assessment**

* Informal observation of students during algorithm writing and programming and more formal assessment of their final program and Flashing wheel light prototype if wished.