THE SYMBOLIC ELEMENT

Your group's symbolic element is **METAL**, which represents **strength**, **resilience**, and **adaptability**. In the context of the **United Nations Sustainable Development Goals (UN SDGs)**, metal can symbolise technological advancements and innovations aimed at addressing environmental challenges.

This worksheet addresses the following UN SDGs within the Transportation industry:

- **Goals 3** Ensure healthy lives and promote well-being for all at all ages.
- **Goals 9** Build resilient infrastructure, promote inclusive and sustainable industrialisation and foster innovation.
- **Goals 11** Make cities and human settlements inclusive, safe, resilient and sustainable.
- **Goals 17** Revitalize the global partnership for sustainable development.

THE TRANSPORTATION INDUSTRY

In the transportation industry, technologies such as Artificial Intelligence (AI), Robotics, Internet of Things (IoT) can be used to provide efficiency, safety, and sustainability in the transportation sector.

AI	Robotics	Internet of Things

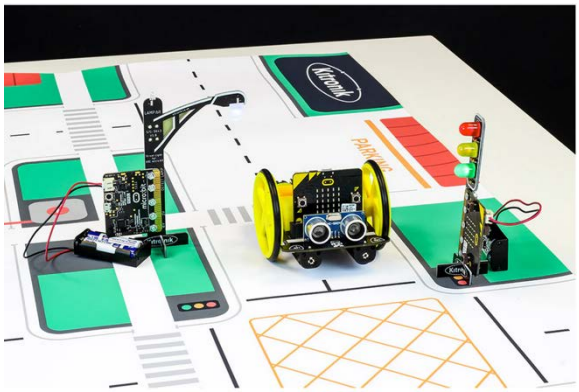
EXTENDED READING:
Artificial Intelligence (AI): AI optimises traffic light timings to reduce congestion. Example: Adaptive traffic signals in London improve traffic flow.

Robotics: Robots deliver packages efficiently. Example: Starship Technologies' delivery robots autonomously navigate sidewalks to deliver goods in cities.

Internet of Things (IoT): IoT devices track vehicle health and location. Example: UPS uses IoT sensors for real-time fleet management and maintenance alerts.

THE "INTERNET OF THINGS" TECHNOLOGY

- Internet of Things (IoT) is about things that have the ability to connect with other things and networks via the Internet.
- IoT helps with increasing automation, reducing costs, and improving end-customer experience.
- In the transportation industry, autonomous cars is usually connect to the Internet and are able to communicate with various devices that are also connected to the internet.
- The radio feature in the BBC micro:bit allows sending and receiving messages between two BBC micro:bits.



THE MISSION: CREATE SIMPLE AUTOMATION WITH THE ACCESS:bit BARRIER

Question: Where are road barriers used?
Please write down you answer below:

Answer:

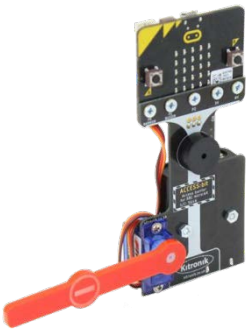
Question: What other features do barriers have?
Please write down you answer below:

Answer:

Question: What could be a reason for having an access barrier?
Please write down you answer below:

Answer:

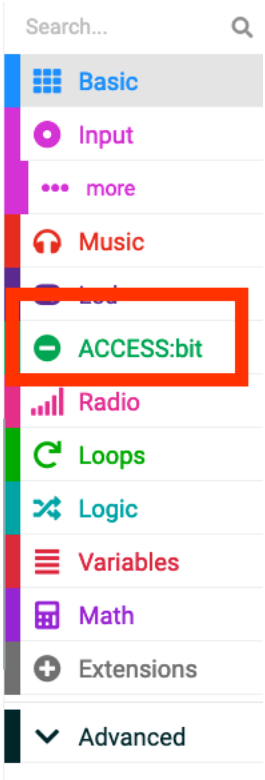
STAGE ONE: AUTOMATE THE BARRIER CONTROL



1. We will use the ACCESS:bit barrier and program the servo in the ACCESS:bit to move the barrier up and down. The ACCESS:bit can be controlled using simple blocks like **"Move Barrier Up".**
2. Let's program the ACCESS:bit to perform a simple up and down movement when you press the button A on the BBC micro:bit attached to the ACCESS:bit. To do that, please follow the steps below:

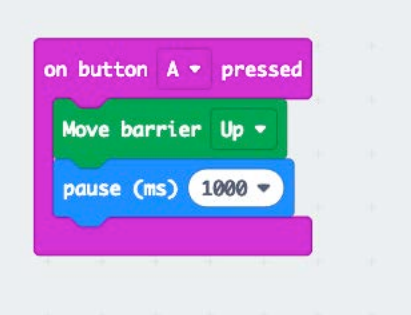
   Step 1: Go to https://makecode.microbit.org/S53638-86691-73956-61072

   Step 2: Click on the ACCESS:bit menu.

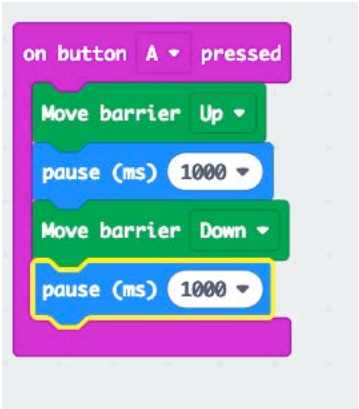


Step 3: Drag and drop the **Move barrier up** command inside the **on button A pressed**. See the screenshot below.

Step 4: Drag and drop **the pause (ms)** command inside the **on button A pressed** and below **the Move barrier up and** set it to **1000**. See the screenshot below.

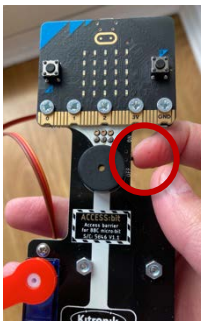


Step 5: Drag and drop the **Move barrier up** command inside the **on button A pressed,** below the **pause** command**,** and change the **up** to **down**. Then, drag and drop a **pause** command and set it to **1000**. See the screenshot below.



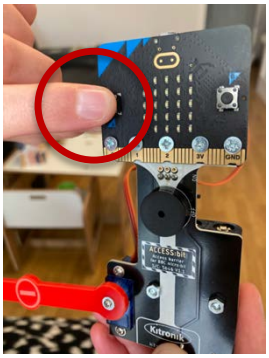
Step 6: Download your project and copy/install it into the BBC micro:bit attached to the ACCESS:bit.

Step 7: First turn on the ACCESS:bit using the button on the side, see the in the picture



Question: Press the A button. What does the ACCESS bit do when press the pressing the A button of the BBC micro:bit attached to the ACCESS:bit (see the picture)?
Please write down your answer below:

Answer:

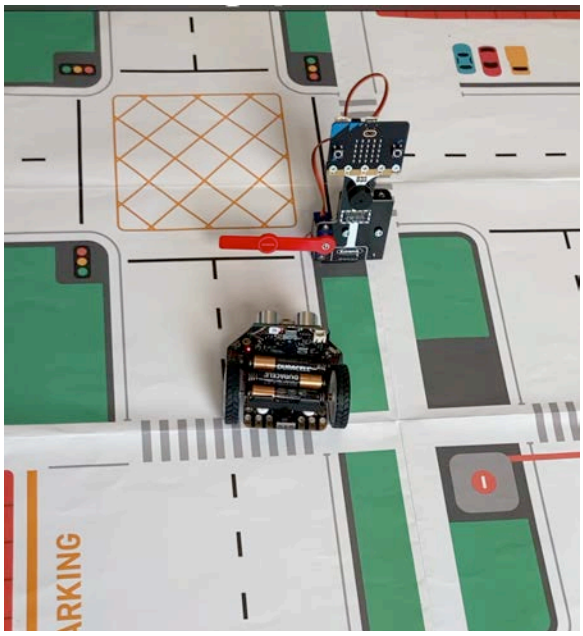


STAGE TWO: USING THE INTERNET OF THINGS TO ESTABLISH COMMUNICATION BETWEEN THE MAQUEEN ROBOT AND THE ACCESS:BIT

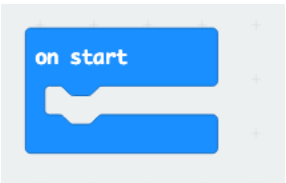
1. We will use the radio functionally inside the BBC micro:bits attached to the Maqueen car and the ACCESS:bit. Radio transmission is a way of sending and receiving messages: BBC micro:bits can use radio waves to communicate with each other. The BBC micro:bit attached to the Maqueen car will send a message to the BBC micro:bit attached to the ACCESS:bit, asking for the gate to be opened.

2. We need to improve on the program so that it detects the barrier in the Maqueen car and stops when it is close to the ACCESS:bit.

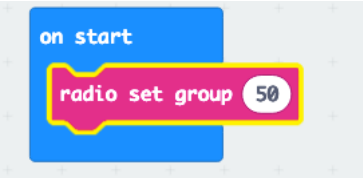
3. Firstly, **we will work on the Maqueen car**. The starting code can be found here: https://makecode.microbit.org/S30127-22369-21925-29652



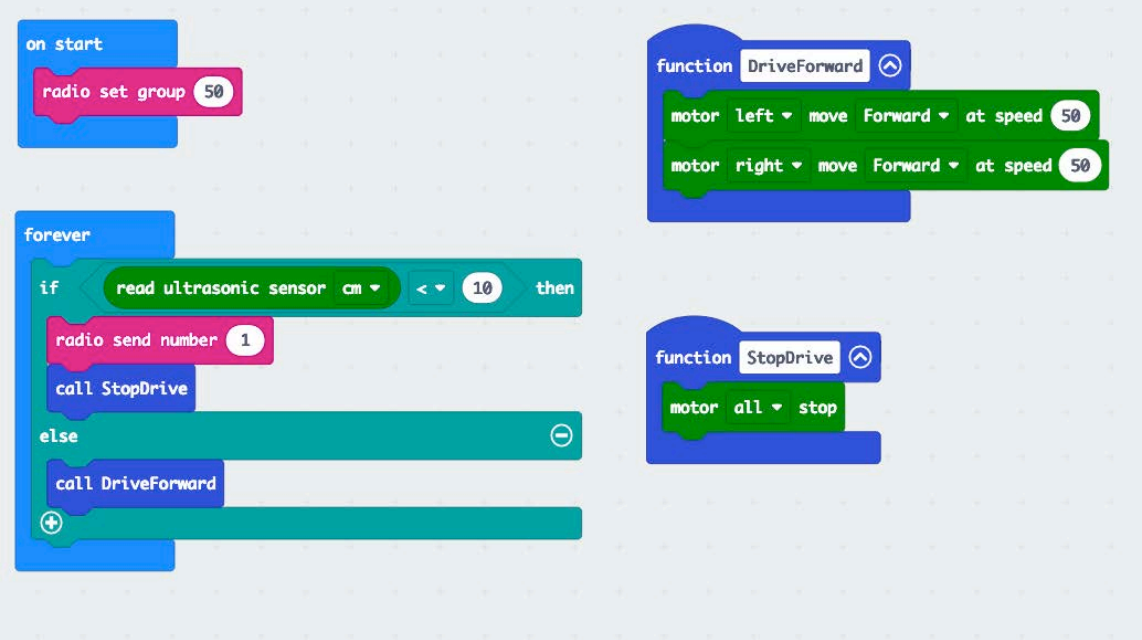
Step 1: To improve on this, we need to choose a “radio group” that allows the BBC micro:bit on the Maqueen to communicate to the BBC micro:bit on the ACCESS:bit. We can do that by dragging and dropping an **on start** command (see the screenshot below):



Step 2: From the radio menu, drag and drop **radio set group** and set the value to 50 (see the screenshot below):



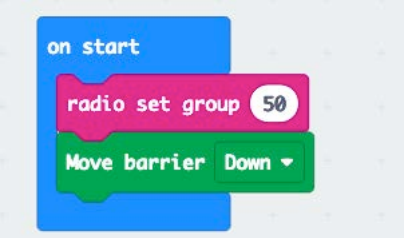
Step 3: Inside the **forever** loop, once the ultrasonic sensor in the Maqueen car detects the stop barrier as an obstacle <10 cm away, the BBC micro:bit in the Maqueen car sends a number 1 to everyone within the radio group 50. See the screenshot below.



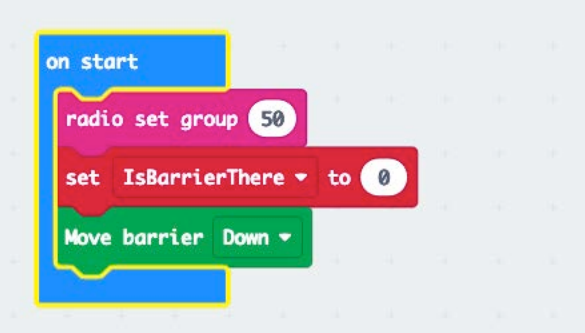
Step 4: Download your project and copy/install it into the BBC micro:bit attached to the Maqueen car.

4. Secondly, **we will work on the ACCESS:bit** and the start code can be found here: https://makecode.microbit.org/S19056-63379-28158-00754

   Step 1: We need to identify a radio group that allows the BBC micro:bit on the ACCESS:bit to receive a message from the Maqueen car and for that we need to drag and drop **radio set group** and set to 50 (see the screenshot below).

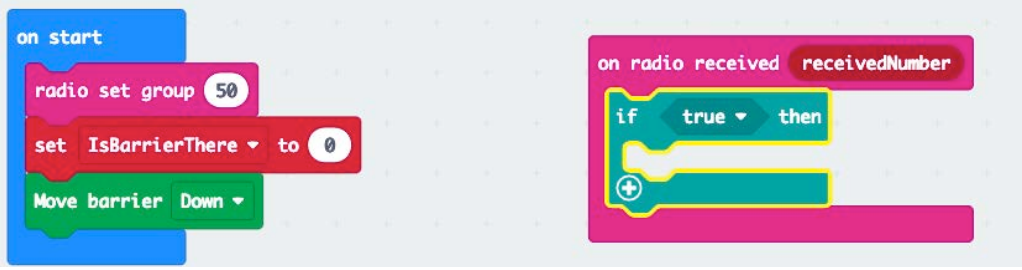


   Step 2: Define a variable called **IsBarrierThere**, and set **IsBarrierThere** to 0 (see the screenshot below).



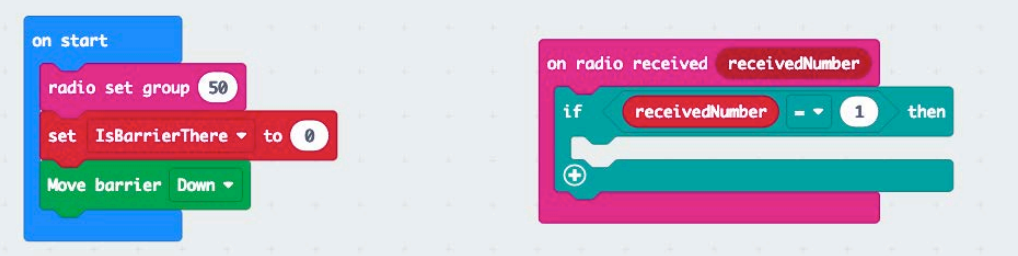
   Step 3: Now, to see if the BBC micro:bit on the ACCESS:bit is receiving a number from the BBC micro:bit on the Maqueen car, if the receivedNumber is 1, then set the "**IsBarrierThere"** variable to 1.Therefore, drag and drop from the radio menu the **on radio received** and set the value to **receivedNumber** (see the screenshot below).

Step 4: Drag and drop from the logic menu an **if** conditional statement (see the screenshot below).

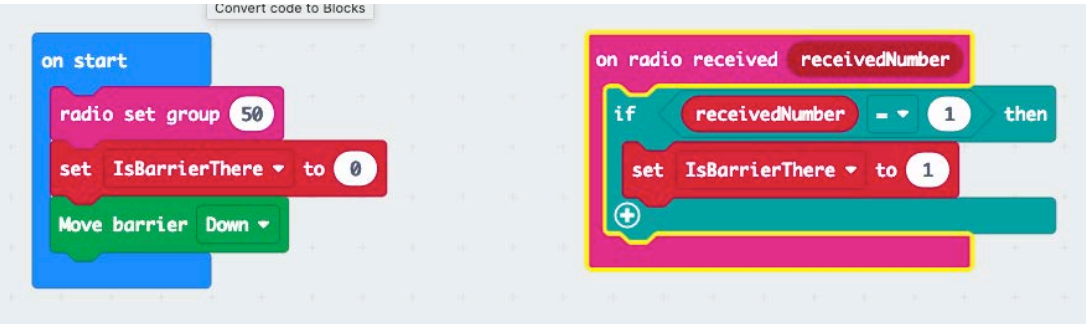


Step 5: Drag and drop from the logic menu a comparison (see the screenshot below).

Step 6: In the **if** statement, set **receivedNumber** to be equal to 1 (see the screenshot below).

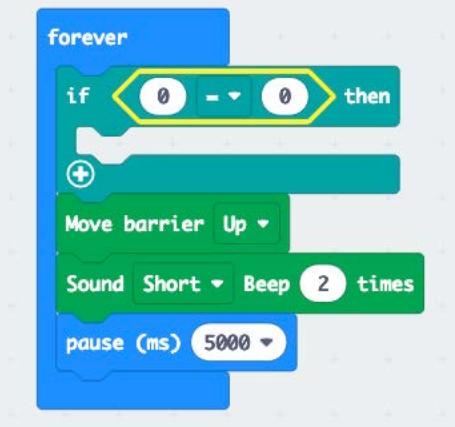


Step 7: Inside the if statement, set **IsBarrierThere** to be equal to 1 (see the screenshot below).



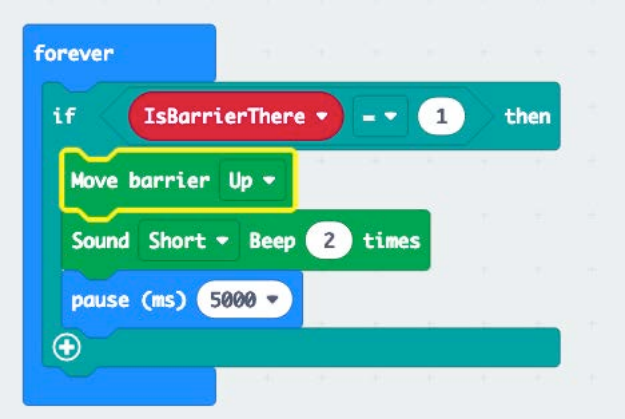
Step 8: Drag and drop from the logic menu an **if** conditional statement (see the screenshot below).

Step 9: Drag and drop from the logic menu a comparison (see the screenshot below).

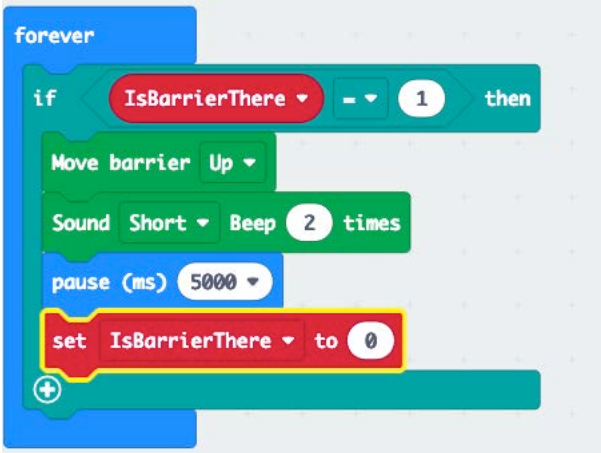


Step 10: Select the variable **IsBarrierThere** inside the comparison in the **if** statement, and set it to equal 1.

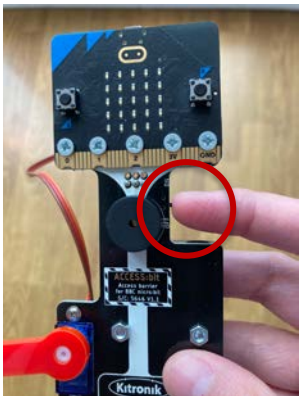
Step 11: Drag and drop **Move barrier up, sound short Beep 2 times, and pause (ms) 5000** inside the if statement.



Step 12: Select **IsBarrierThere** from the variable menu and set **IsBarrierThere** back to 0 (see the screenshot below).

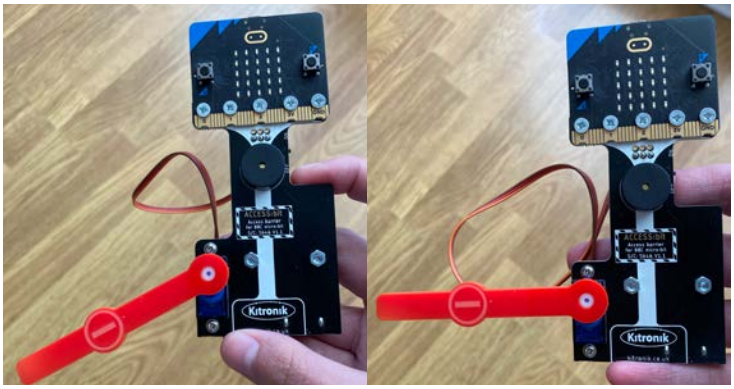


Step 13: Now, turn the ACCESS:bit off, if it's on.



Step 14: Then, download your project and copy/install it into the BBC micro:bit attached to the ACCESS:bit.

Step 16: Turn on the ACCESS:bit. The barrier should move down. Wait a second, and then (if needed) adjust the barrier by hand so that it looks like the picture on the right below.



Step 17: Place the ACCESS:bit and Maqueen car in the map given to you at the positions indicated in the screenshot below.