

Code, Measure and Sonify: Science and Data in Action

Lesson summary

Target Audience: 10-16 years old. Beginners.

Duration: 50 minutes.

Learning goals:

Students will take part in an engaging hands-on activity where they will apply their programming skills to measure physical variables such as temperature and light intensity. They will collect real data using smart sensors and transform these data into sounds. The goal of this activity is to advance their understanding of sensor programming and physical measurement, explore automatic data collection through intelligent devices, and discover new ways of representing data through sound.

The objective is to help students develop computational thinking skills by:

- Develop programming skills to solve problems (algorithms and coding).
- Decomposing the problem into manageable parts (problem decomposition).
- Design and apply algorithms to process data (algorithmic design and process automation).
- Analyze and abstract relevant information from collected data (data abstraction and analysis).

Online or offline: online

Computational Thinking:

- **CT-foundations:**
 - Decomposition: Breaking down clues and steps to isolate password components
 - Abstraction: Ignoring irrelevant clues or distractions
 - Algorithmic Thinking: Systematically testing possibilities to uncover the correct password
 - Logical Reasoning: Deducting correct choices based on logical elimination

Materials

- Computer.
- Internet.



- Micro:bit electronics board
- Optional: external sensors like humidity and water level sensors, compatible with micro:bit.

Preparation

1. **Divide participants into small groups of 2 or 3 people.** If the groups are larger, it may be difficult for everyone to interact with the board and its components.
2. **Provide each group with a computer and a micro:bit board** and, if possible, a kit of external components. For example, this activity involves the use of external LEDs and a water level sensor.
3. **If external components** such as a water level sensor are available, provide each group with a glass of water. We can also change this setting depending on the external sensor we have. For example, we can have a small plant per group and a soil moisture level sensor.



Lesson Description - Code, Measure and Sonify

Introduction (5 minutes)

Welcome students and explain the challenge:

"Did you know that there's a whole world of data waiting to be discovered around you? In this activity, you'll become an explorer of your environment: you'll learn how to program a sensor board to capture information from the world around you, such as light, temperature, or sound, and to analyze that data in a fun and creative way.

But we won't stop at numbers, we'll also listen to the data. You'll discover how to turn information into sound and how technology can help us understand our surroundings in new and surprising ways. Get ready to experiment, program, and listen to the world in a new way!"

Core

Warm-up Activity (10 minutes)

- We encourage them to think about data or **variables associated with climate** and present them with a small sample of climate data (the experience will be better if we provide local climate data for our region). If not, we can use the example of the climate data table available in the appendix 1.

"Look at the data and think about the following questions:

- What data or climate-related variables are shown in this sample?
- How do you think these variables are measured?
- How could we represent the data?
- How could this data be encoded with sound?

- We **relate the activity to the computational thinking**. To do this, we explain how decomposition, abstraction, algorithms, and pattern recognition are applied in data analysis.

"We can apply what we know about **Computational Thinking** to solve this challenge. First, by using **abstraction**, we'll focus on the most important data from the table. Then, we can **decompose** the information and **look for patterns** that will help us represent the data through different sounds. For example, we could group the data (based on the variables of light and humidity) like this:

- Dark and rainy days → a loud sound
- Sunny days → a soft sound

Finally, to transform the data into sounds—in other words, to sonify the data—we can create an **algorithm**. We can do this by making sounds with objects we find in the classroom, or even with hand claps. This way, we'll turn weather data into a unique sound experience!"



Main Activity – Programming the sensors with micro:bit and sonify the data

Round 1 (20 minutes) *All steps are explained for the student in Appendix 2.*

- First, we show participants how to program the board to obtain the value of the **board's internal sensors**, such as the light sensor, and how to display that value on the LED matrix. This can be tested with other internal sensors such as the temperature sensor or the compass.
- Next, we show participants how to create **variables** and work with the data from a variable.
 - We ask them to create three variables: variable X, variable Xmax, and variable Xmin. This will allow us to record the minimum and maximum values of the sensor we are using. We set the variable X to take the values from the light sensor. The micro:bit's internal light sensor provides a value between 0 (dark) and 255 (very bright). We set the variable Xmin to the lowest possible measured light level value, 0, and the variable Xmax to the highest possible measured light level value, 255.
- Next, we show them how to **sonify** or how we can convert the sensor variations into sound. To do this, we write the following program:
 - “We play a 1-beat tone until it ends, with a frequency resulting from assigning the value X to the chosen frequency range (from 200 Hz to 2000 Hz)”.
- We encourage students to **experiment** with other internal sensors on the board. In each case, the data range provided by the sensor must be taken into account. For example, for the compass, the maximum value will be 360, and for the temperature sensor, we can use a range of 10° to 45°.
- Final **reflection** based on the following questions:
 - How can collecting this type of data through sensors be useful?
 - In what contexts could it be useful to emit audible alerts based on the sensors?
(*Example: audible alerts when there is seismic activity to warn the population*).

Round 2 (Optional or Advanced – 15 minutes)

- We can perform the same experiment with **external sensors**. For example, using a water level sensor connected to the micro:bit and a glass of water. To do this, we add the *IOT-Environment kit* extension to the micro:bit. A new category called OCTOPUS will appear, along with a variable called “value of water level (0-100) at pin.” In this case, the data range will be from 0 to 100. It is important to note which pin on the board we connect the external water level sensor to in order to enter it into the program.
- In the previous code, we only need to change the data range from 0 to 100 and change the sensor value. Where the light sensor was, we will now place the water level sensor variables.



- In this case, we observe how, when inserting the sensor into the glass of water and modifying the depth of the sensor, the sensor values change and, therefore, so does the sound.
- Green and red LEDs could also be connected and **light signals** added. We improved the code by connecting two LEDs, red and green.
 - We programmed it to fulfill a **condition**: when the water level value is greater than or equal to 50, the red LED (pin 2) turns on and the green LED (pin 3) turns off, and when it is less than 50, the red LED (pin 2) turns off and the green LED (pin 3) turns on.

Debrief and Evaluation (5 minutes)

- Ask reflection questions:
 - Have you thought about situations where it would be useful to program audible and visual alerts for water levels? For example, in a dam?
 - Would you now dare to connect other components to the board, such as a motor?
 - How would we program the motor to activate when the water level exceeds 75?
 - What programming concept do we need to implement to carry out this program? (*Answer: use of conditionals*).



Appendix 1

Example of a the climate data table

Date	Luminosity (lumens)	Relative Humidity (%)	Rainfall Intensity (mm/day)	Wind Speed (km/h)
05/01/2023	1534	76,1	3,26	13
20/01/2023	1609	75,4	0	12,5
03/02/2023	1879	77,1	2,17	11,8
19/02/2023	1983	76,6	3,73	11,5
12/03/2023	2488	70,7	3,88	19,3
22/03/2023	2448	71,2	1,19	11,2
01/04/2023	2514	61,4	0,03	14,8
12/04/2023	2423	65,6	0	11,9
02/05/2023	2235	54,4	0	10
13/05/2023	2105	54,4	2,62	11,6
05/06/2023	1849	43,3	3,48	13
21/06/2023	1574	46	0	11,7
03/07/2023	1564	51,9	4,18	12,6
17/07/2023	1276	54,8	2,47	12
05/08/2023	839	50,3	0	14,5
15/08/2023	829	51,6	3,18	13,6
02/09/2023	500	65,1	1,39	12,5
21/09/2023	500	62,5	0,18	7,6
12/10/2023	835	71,1	3,95	17,5
22/10/2023	614	69,4	3,49	9,7
07/11/2023	656	69,9	2,95	9,7
19/11/2023	703	74,2	0,49	12,8
08/12/2023	1135	72,3	0,85	11,3
27/12/2023	1492	79	0,6	6,7

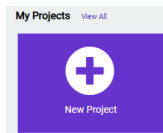


Appendix 2

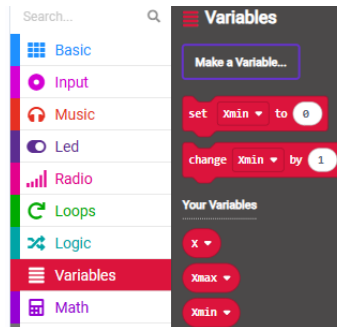
Instructional guide for students

Programming the sensors with micro:bit and sonify the data:

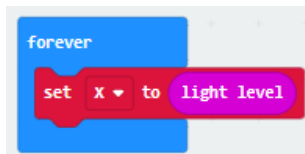
1. We go to <https://makecode.microbit.org/> and we create a new project.



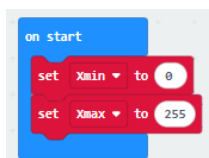
2. We create 3 new variables:
 - a. X
 - b. Xmax
 - c. Xmin



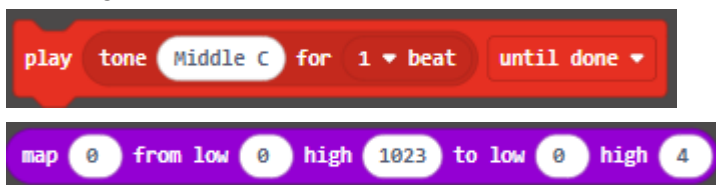
3. We establish that X will take the values from the light sensor.



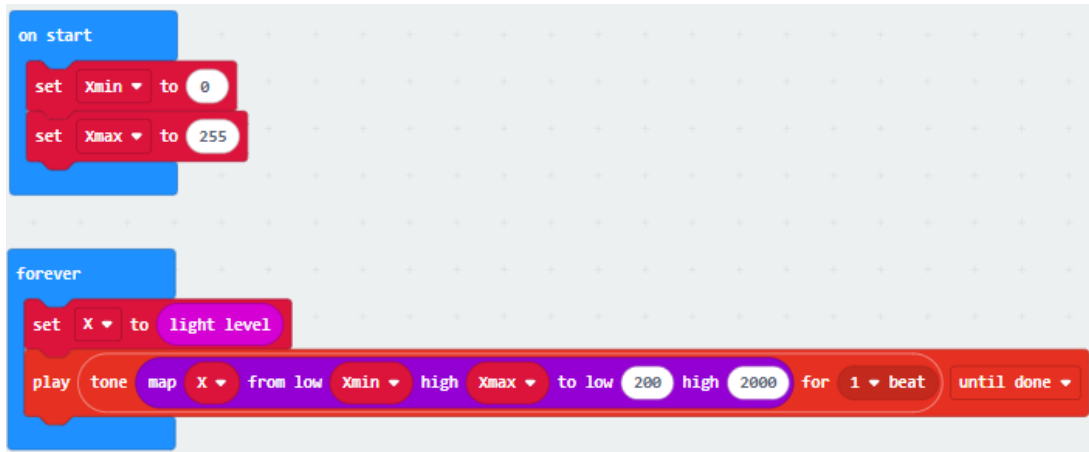
4. We establish that at the start of the program, Xmin is equal to 0 and Xmax is equal to 255. The minimum and maximum values of the light variable.



5. We add the following program to sonify the light sensor values: *“We play a 1-beat tone until it ends, with a frequency resulting from assigning the value X to the chosen frequency range (from 200 Hz to 2000 Hz)”*. To do this, you will need to use the following commands:



This is how the complete code would look:



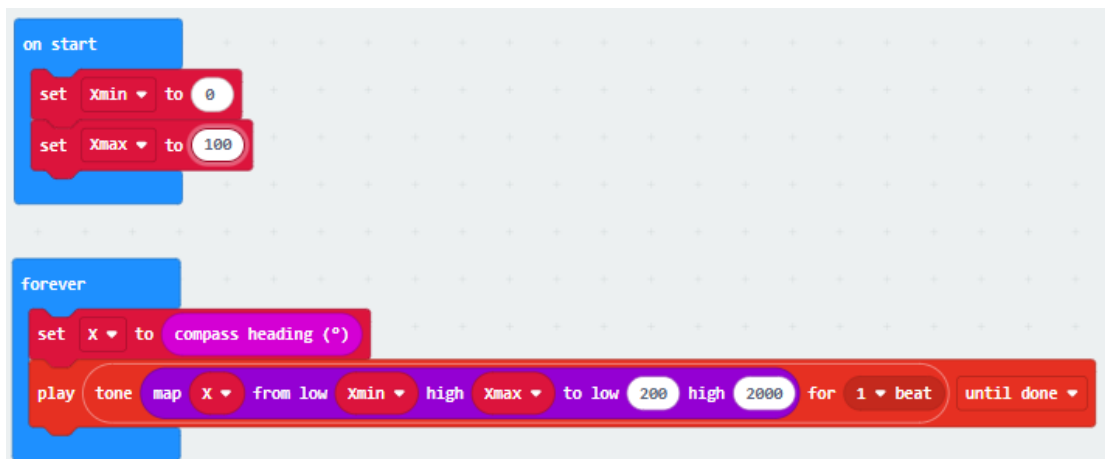
```
on start
  set Xmin to 0
  set Xmax to 255

forever
  set X to light level
  play tone map X from low Xmin high Xmax to low 200 high 2000 for 1 beat until done
```

As we change the light level in our environment, we will notice differences in sound.

- Next, we experiment with other sensors, such as the compass available into the micro:bit board. To do this, we have to change the value of X so that it corresponds to the value of the compass sensor and also the minimum and maximum values, which will be 0 to 360.

This is how the complete code would look:

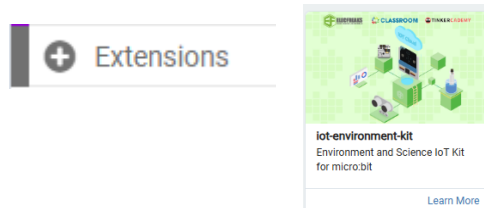


```
on start
  set Xmin to 0
  set Xmax to 100

forever
  set X to compass heading (°)
  play tone map X from low Xmin high Xmax to low 200 high 2000 for 1 beat until done
```

Advanced level (If we have external components)

- We can perform the same experiment with external sensors. For example, using a water level sensor connected to the micro:bit and a glass of water. To do this, we add the **IOT-Environment kit** extension to the micro:bit.

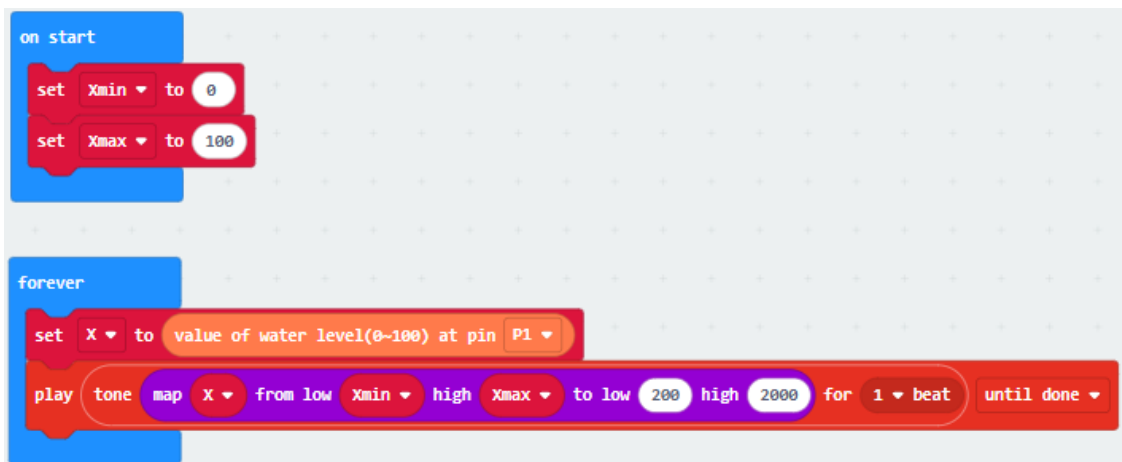


A new category called OCTOPUS will appear, along with a variable called “value of water level (0-100) at pin P1.



Pay attention to the pins! We connect the sensor to pin 1 and look at the color of the wires that match G with G. It is important to note which pin on the board we connect the external water level sensor to in order to enter it into the program.

8. In the previous code, we only need to change the data range from 0 to 100 and change the sensor value. Where the light sensor was, we will now place the water level sensor variables.



In this case, we observe how, when inserting the sensor into the glass of water and modifying the depth of the sensor, the sensor values change and, therefore, so does the sound.

9. Now we are going to connect green and red LEDs to add light signals. We programme it to fulfill a condition:
When the water level value is greater than or equal to 50, the red LED (pin 2) turns on and the green LED (pin 3) turns off, and when it is less than 50, the red LED (pin 2) turns off and the green LED (pin 3) turns on.

This is how the complete code would look:

```
on start
  set Xmin to 0
  set Xmax to 100

forever
  set X to value of water level(0-100) at pin P1
  if X >= 50 then
    LED P2 toggle to ON
    LED P3 toggle to OFF
  else
    LED P2 toggle to OFF
    LED P3 toggle to ON
  play tone map X from low Xmin high Xmax to low 200 high 2000 for 1 beat until done
```

