LESSON 7

Data Representation

80 minutes (2 periods) | Foundations of Computer Science | Outcome 2.13 | Outcome 2.14 | Outcome 2.17 | Outcome 2.18 | Outcome 1.9 | Outcome 3.11 | Outcome 3.13

WHAT THIS LESSON DOES

Explore how computers represent data using binary code. In this lesson, you'll learn the basics of converting text, images, and sound into 0s and 1s, with hands-on tasks to help you understand the process.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
2.13	describe the rationale for using the binary number system in digital computing and how to convert between binary, hexadecimal and decimal
2.14	describe the difference between digital and analogue input
2.17	use ASCII and Unicode character sets to encode/decode a message and consider the importance of having such standards
2.18	collect, store and sort both continuous and discrete data
1.9	use modelling and simulation in relevant situations
3.11	use and control digital inputs and outputs within an embedded system
3.13	develop a program that utilises digital and analogue inputs

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand how binary code represents various forms of data such as text, images, and sound in digital systems.
- Explain the principles of data digitisation, including ASCII for text, pixel encoding for images, and sampling for sound.
- Apply binary conversion techniques to encode and decode simple data representations.
- Demonstrate practical skills in creating and manipulating digital data through hands-on tasks.
- Recognise the importance of binary as the fundamental language of computers for data storage and processing.

BEFORE THE LESSON

- Test the online ASCII art drawing tool from a student machine, since the school filter may block it. Have the URL ready to paste into the class.
- Open the MakeCode editor yourself and confirm it loads and plays sound through the browser simulator. Physical micro:bits and cables are optional if the simulator works.
- Have the uppercase ASCII decimal and binary table on screen ready for the name conversion task.

Data Representation

HOW THE LESSON RUNS

- Introduction
- Binary Representation of Text
- Practical Task: Convert your Name
- Practical Task: Make Some ASCII Art
- Binary Representation of Images
- Practical task: Coding Image Representation
- Binary Representation of Sound
- Practical Task: Sound
- Wrap Up

TEACHING MOVES

- 2. Binary Representation of Text. Walk through 'A' equals 65 equals 01000001 on the board once, decimal to binary in full, before pointing students at the table. Show the halving or place-value method so they can check, not just look up.
- 3. Practical Task: Convert your Name. Insist they write the decimal values first, then convert each to 8-bit binary. The intermediate decimal step is where errors are caught. Ask a few to read a binary string back to a letter to prove decoding works both ways.
- 4. Practical Task: Make Some ASCII Art. Keep this short and light. The point is that the picture is just characters, each with its own code. Have one student paste their ASCII art as plain text to prove it is nothing but symbols.
- 5. Binary Representation of Images. Stress one bit per pixel only works for black and white. Ask what changes if you want grey or colour, and let them reach 'more bits per pixel' themselves before you name bit depth.
- 6. Practical task: Coding Image Representation. Do one row together, then one full symbol like the arrow, so they see the 5 groups of 5 map to rows top to bottom. Have pairs swap binary strings and draw each other's grid.
- 7. Binary Representation of Sound. Draw a smooth wave, then mark sample points on it. Make the analogue-to-digital jump concrete: the computer only keeps the dots, not the line. Introduce sampling rate and bit depth as how many dots and how precise each one is.
- 8. Practical Task: Sound. Let them change one parameter at a time and listen, so cause and effect stays clear. Connect what they hear back to sampling rate and bit depth affecting quality.

WHAT TO WATCH FOR

- Students treat the ASCII table as the definition of the letter, thinking 'A is 65' by decree, and cannot convert 65 to binary themselves. Make them derive at least one binary value by hand using place values or repeated halving, then check it against the table.
- Students think an image is stored as the picture itself, or that one pixel needs many bits by nature, missing that black-and-white needs only one bit each. Use the 5x5 grid: 25 pixels, 25 bits for black and white. Ask what extra you would need for colour to show bit count is a choice.
- Students believe digital sound is a perfect copy of the original wave rather than a set of samples. Return to the drawn wave with sample dots. Ask what happens to the sound between two samples to show information is lost, and why more samples means better quality.

DIFFERENTIATION

- Emerging: Give them a shorter word, two or three letters, for the name task and let them use the table directly rather than converting decimal to binary by hand. Pair them for the grid task so they read the binary string aloud while a partner colours squares.
- Developing: Have them explain in their own words why black-and-white needs one bit but colour needs more before they move to the sound section. Ask them to decode a classmate's binary name back into letters without the table in front of them.
- Proficient: Ask them to work out how many bytes their full name takes and extend the idea to file size, moving toward Higher Level treatment of storage. Have them design a colour scheme for the grid using two bits per pixel and state how many colours that allows, as evidence they could use in an Applied Learning Task.