Binary Bits

Year level band: 7-8

Description: The students will be exploring binary number systems by using their Micro:bit as a Transmogrifier before making their own BitWatch.

Resources:

- Micro:Bits, iPads, Computers
- Paper, pens (black)
- Binary Decoder Key (https://code.org/curriculum/course2/14/Activity14-BinaryBracelets.pdf)
- Transmogrifier Activity (https://makecode.microbit.org/courses/csintro/binary/activity)
- Transmogrifier (converter) from Microsoft MakeCode (https://makecode.microbit.org/v0/07576-23150-47531-98137)
- Number Systems Maths Fun (http://www.math-aids.com/Number_Systems/Decimal_Binary.html)
- Binary video links - Part 1 https://goo.gl/Xn2BF4 Part 2 https://goo.gl/ezihko
- 'BitWatch' - (https://github.com/petejbell/BitWatch)
  a Binary Watch programme written in MicroPython for the BBC Micro:bit by @petejbell and distributed under a MIT licence

Prior Student Learning:

MakeCode JavaScript Blocks programming tutorials.

Summary In this lesson students have the opportunity to develop their binary understandings and build their own Bit Watch.	
Year	**Content Descriptors**
7-8	Examine how whole numbers are used to represent all data in digital systems (ACTDIK015) Implement digital solutions as simple visual programs involving branching, iteration (repetition), and user input (ACTDIP020) Explain how student solutions and existing information systems are sustainable and meet current and future local community needs (ACTDIP021)

Element	**Summary of tasks**

Learning hook	[Explain Transmogrification](#)  by Calvin and Hobbes and Introduce students to binary with the following videos (or your own explanation). [Watch Understanding Binary - Part 1](#) Students from Courtney Gardens Primary School explain the Binary System with practical examples and [Converting Numbers to Binary](#) - Part 2 Students from Courtney Gardens Primary School explain the Binary System with practical examples. Have students code their initials for a [binary code bracelet.](#)
Achievement Standards	By the end of Year 8, students distinguish between different types of networks and defined purposes. They explain how text, image and audio data can be represented, secured and presented in digital systems. Students plan and manage digital projects to create interactive information. They define and decompose problems in terms of functional requirements and constraints. Students design user experiences and algorithms incorporating branching and iterations, and test, modify and implement digital solutions. They evaluate information systems and their solutions in terms of meeting needs, innovation and sustainability. They analyse and evaluate data from a range of sources to model and create solutions. They use appropriate protocols when communicating and collaborating online.
Learning Map (Sequence)	Students will: • use the Binary Decoder Key to show how a computer might represent capital letters and design a paper based alphabet game. • download the [Transmogrifier](#) (converter) from Microsoft MakeCode that converts between binary (base-2) and decimal (base-10) numbers. • translate the Transmogrifier code onto paper as a pseudocode that will represent the patterns needed to execute the program • identify patterns that will help them figure out, for any given decimal value, what the new decimal value should be if the user enters a 0, or if the user enters a 1 • complete these [number system tasks](#) (https://goo.gl/LBawG2) with the help of the transmogrifier. • [Make a Bit Watch](#)

Learning input	The teacher will step the class through binary decoding using the decoder key. Students will become more familiar with binary numbers by downloading the transmogrifier program from Microsoft MakeCode to their Micro:bit. For students to understand the binary patterns that make up this code, they will work backwards from the MakeCode to writing the program in pseudocode to identify the patterns. Students will use a transmogrifier to work with binary and decimal number systems. Students will use the Micro:bit to make a BitWatch. To make the watch students will need to follow these instructions from 'BitWatch' (https://github.com/petejbell/BitWatch/archive/master.zip) -a Binary Watch programme written in MicroPython for the BBC Micro:bit by @petejbell and distributed under a MIT licence. **Instructions:** Download Mu - a simple python for beginners - from here: https://github.com/ntoll/mu or work online from https://codewith.mu/ Copy and paste the BitWatch code (https://github.com/petejbell/BitWatch/archive/master.zip) to Mu, connect your Micro:bit to your computer and then flash the code to your Micro:bit The BitWatch will display 18:50 as the time for 10 seconds and will then show '18:51'. Use Button A to set the Hours and B to set the Minutes. Hold each one down and you will see the hours/minutes increment. Use Buttons A+B together to reset seconds to '0'. Column 0 shows the first digit in the hours (in 24hr clock) Column 1 shows the second digit. Column 2 shows the seconds flashing away. Column 3 shows the first digit in the minutes Column 4 shows the second digit.
Learning construction	1. Students use binary decoder keys 2. Students read Microsoft MakeCode to translate binary Base-2 to decimal Base 10 3. Students are able to download code to the Micro:bit 4. Students convert the blocks program to pseudocode to recognise the patterns needed to convert digital to binary. 5. Students will have fun with Number Systems (http://www.math-aids.com/Number_Systems/Decimal_Binary.html) using their transmogrifier. 6. To further explore binary numbers, students will complete the BitWatch Project (https://github.com/petejbell/BitWatch/archive/master.zip).
Learning demo	Have students convert 64 and 32-bit Base2 binary number to a base 10 digital and develop a pseudocode for the pattern.
Learning reflection	What is a Bit? Byte? Kilobyte? Megabyte? In the computer world in what context have they heard 32 bit and 64 bit used?

Assessment:

	Depth of knowledge		Depth of understanding		
Criteria	Pre-structural	Uni-structural	Multi-structural	Relational	Extended abstract
The nature of data and its representation	Knows that binary systems use '1s and 0s ' but doesn 't truly grasp what each symbol represents	Understands that the use of 1 and 0 in binary systems is a representation of two different states, and that computers store data in this way	Can explain why binary states are appropriate for representing data in a computer system given their physical characteristics	Articulates not only the reason binary data is used in computer systems, but also the ways that it can be structured to represent different types of data	Has a deep understanding of the nature of data and how its representation is fundamental not only to how computer systems operate, but to communication and interaction in society generally
Limitations and compromises of representing data	Knows that there are physical limits associated with the representation of data but has no true sense of the relative size of different measures (eg bit vs byte)	Understands that the storage of images and multimedia data requires more physical space than simple numeric and/or text data	Can describe how all data can be represented numerically regardless of its type, and that this is achieved using different storage techniques	Can explain why the storage of more complex data requires more physical space, and the challenges this creates for data transfer and communications	Has a deep understanding of how different algorithms balance the loss of quality or integrity of data against the costs of storage, and why this is relevant in a highly connected world

Assessment Activities

- Convert binary numbers to decimal and decimal to binary using their own Micro:bit Binary Transmogrifier.
- Present their pseudocode pattern identification tree for assessment
- Ability to download and modify the appropriate code to create a functioning binary watch.

CSER Professional Learning

This lesson plan corresponds to professional learning in the following CSER Digital Technologies MOOCs:

F-6 Digital Technologies: Foundations. See: http://csermoocs.adelaide.edu.au/moocs

● Unit 7: Algorithms and Programming

● Unit 8: Visual Programming

Further Resources

- Digital Technologies Hub: www.digitaltechnologieshub.edu.au
- CSER: https://csermoocs.adelaide.edu.au

Author: Robin McKean