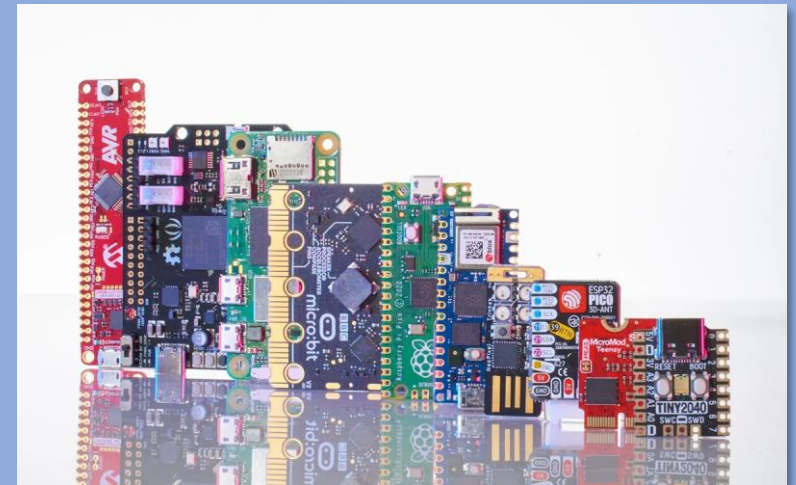


Year 9 Topic 4 – Physical computing

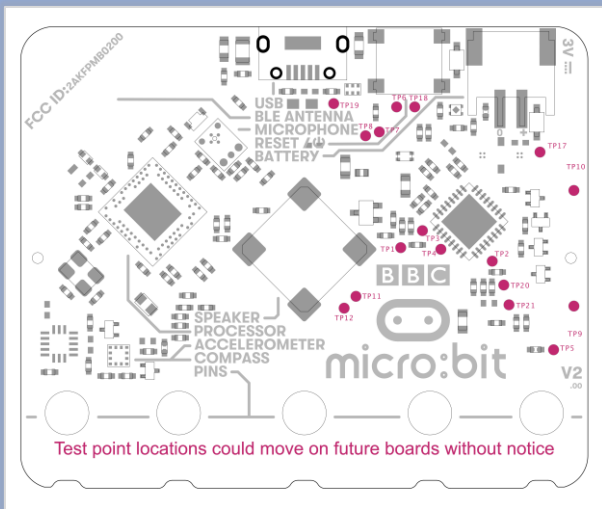
Lesson	Can you?
1 Hello physical world	Describe what the micro:bit is List the micro:bit's input and output devices Use a development environment to write, execute, and debug a Python program for the micro:bit
2 Bare bones	Write programs that use the micro:bit's built-in input and output devices
3 Connections	Write programs that use GPIO pins to generate output and receive input Write programs that communicate with other devices by sending and receiving messages wirelessly
4 Dream it up	Design a physical computing artifact purposefully, keeping in mind the problem at hand, the needs of the audience involved, and the available resources Decompose the functionality of a physical computing system into simpler features
5 Build it up	Implement a physical computing project, while following, revising, and refining the project plan
6 Wrap it up	Implement a physical computing project, while following, revising, and refining the project plan

Useful websites

- www.microbit.org
- www.python.microbit.org
- www.microbit-micropython.readthedocs.io/en/v1.0.1
- www.arm.com/resources/education/schools/content
- www.blog.teachcomputing.org/tag/pedagogy
- www.youtu.be/oNLF6aFYVoU



KNOWLEDGE ORGANISER
Key Stage 3 - COMPUTING



The micro:bit helps you understand how computers work. When you type on your laptop or touch the screen on your phone, you're using an **input** device. Inputs allow computers to sense things happening in the real world, so they can act on this and make something happen, usually on an **output** like a screen or headphones.

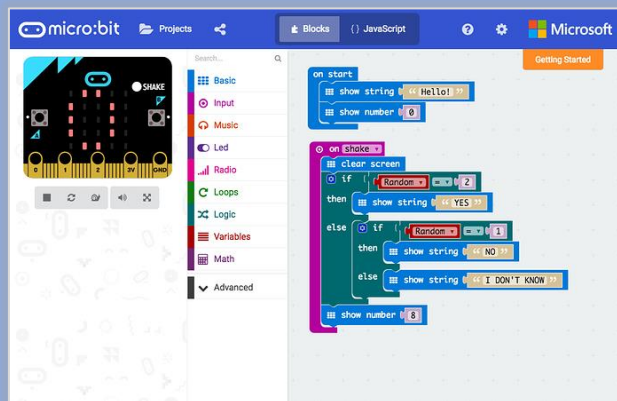
In between the input and the output, there is the **processor**. This takes information from inputs like buttons, and makes something happen on outputs, like playing a song in your headphones.

You tell computers like the micro:bit what to do by giving them instructions. Sets of instructions for computers are called **programs**. Programs are written in **code**, a language that both you and the computer can understand.

Transferring your program to your micro:bit is called **flashing** because it copies your program into the micro:bit's **flash memory**.

Your micro:bit will pause and the yellow LED on the back will blink while your program is being transferred. Once it's copied across, your program starts running on your micro:bit.

There are two ways to transfer your program from a computer:



A GPIO (general-purpose input/output) port **handles both incoming and outgoing digital signals**. As an input port, it can be used to communicate to the CPU the ON/OFF signals received from switches, or the digital readings received from sensors.

