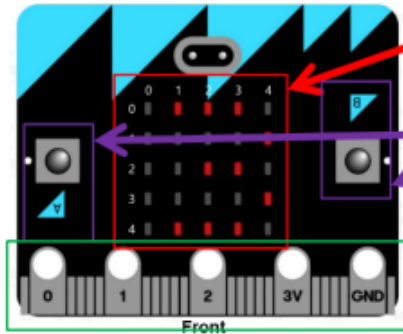


Year 7 Topic 5 – I am a Computational Scientist

The Micro:Bit

The BBC Micro:Bit is a pocket-sized codable computer. It is fully programmable...you can turn it into whatever device you want (within reason). Let's explore it!



- 25 Programmable LEDs**
These can be programmed to display text, numbers or the objects in simple games!
- Two Buttons**
These can be independently programmed to produce different actions when pressed. They can also be programmed to produce actions if pressed together.
- Input / Output (I/O) Connectors**
Different 'external' devices can be attached to the micro:bit (such as a motor or a speaker).

USB Connector
Enabling the Micro:Bit to connect to the PC in order to 'flash' your programs onto it.

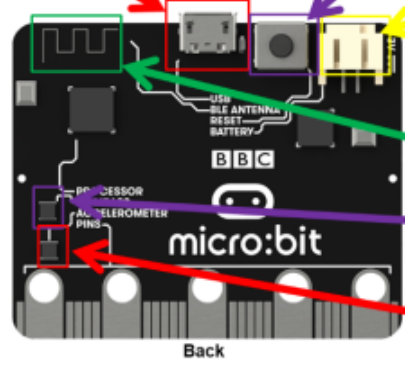
Reset Button
This is used to reset / restart your program, when it is in use.

Battery Connection
This can be used to connect the battery pack so you can use the Micro:Bit as a mobile device.

BlueTooth Antenna
Enabling the Micro:Bit to connect to devices wirelessly

A Compass
This can be used to create programs which requires data about which way the Micro:Bit is facing.

An Accelerometer
This can be used to sense movements in the Micro:Bit device.



Event Driven Programming

As the Micro:Bit is a portable device, we may choose to program it to respond to events, for example, the clicking of a button or perhaps movement.

When using the blocks-based language, there are some blocks which will help us run code, when these events occur.

On Button Press – Event

Here is an example of a script which will generate and display a random number (via the board's LEDs) when the 'A' button is pressed.

```

on button A pressed
  set item to pick random 0 to 10
  show number item
  
```

On Shake - Event

Similarly, here is an example of a script which will generate and display a random number (via the board's LEDs) when the Micro:Bit is shaken. This shake event will be sensed by the board's accelerometer, which will result in the execution of the code.

```

on shake
  set item to pick random 1 to 6
  show number item
  
```

Key Vocabulary

Key Word	Definition
Micro:Bit	A miniature computer which is fully programmable.
Compile	The process of translating our program code into machine code.
Flash	The process of transferring machine code onto a computer chip
Accelerometer	A component of the Micro:Bit which can sense movement.
Variables	A memory store in a program (think of it as a box which stores a piece of data).
Coordinates	A system which enables us to locate the position of an object using a horizontal (x) and vertical (y). For example, a pixel on a screen.
Syntax	The set of rules and key words that govern the structure of a programming language.

Text-Based Programming

Programming with 'Blocks' is great! But, there is a limit with what you can do. If you wish to get your Micro:Bit working with other components like motors and screens, we need to use a more advanced language. Luckily, the Micro:Bit can be coded in Python.

Programming in Python

To get started, choose 'Python' from 'Code options' on the 'create project' popup. You will be taken to the MicroPython IDE (programming environment). Here you can write your code, simulate the code in the Micro:Bit simulator, and download the compiled program, ready to flash it onto your device.



Example Python Program – Using the Temperature Sensor

The Micro:Bit has a built in thermometer and the following program makes use of it!

```

1 while True:
2     current_temp = input.temperature()
3     basic.show_number(current_temp)
  
```

This program begins with the code "while True". This line is effectively a forever loop. The code inside it (indented underneath) will run repeatedly.

The first line inside the loop will read the temperature of the thermometer and store it in a variable called 'current_temp'. Then, the second line displays the contents of the variable on the screen.

The Compass

Amazingly, the Micro:Bit contains an onboard compass which measures the direction that the board is facing, storing this measurement as a compass bearing.

The following code, when compiled and run on the Micro:Bit, can demonstrate this, by constantly displaying the board's compass direction via the LEDs.

```

forever
  show number compass heading (*)
  
```

And by using some IF statements (which allow programs to run different lines of code, depending on conditions), we can program the Micro:Bit to display the directions East, South, West and North, via text, depending on the compass bearings measured at that moment in time. Here is a glimpse of the top half of a code snippet to demonstrate this logic:

```

forever
  if compass heading (*) > 45 and compass heading (*) < 95 then
    show string 'E'
  else if compass heading (*) > 135 and compass heading (*) < 185 then
    show string 'S'
  else if compass heading (*) > 225 and compass heading (*) < 275 then
    show string 'W'
  else if compass heading (*) > 315 and compass heading (*) < 360 then
    show string 'N'
  
```

How do we program the Micro:Bit?

We program the Micro:Bit use the tools and facilities at <https://makecode.microbit.org/>. We can write our code in either a blocks-based language or text-based language.

The process of getting our Micro:Bits working (with software) is as follows:



- Code it**
First we code our program. We can check it works by using the online Micro:Bit emulator.
- Compile it**
Then we compile the code. This is where our code is converted into binary (zeros and ones) so that the Micro:bit's CPU can process the program.
- Flash it**
Finally, we transfer the compiled program onto the device. Flashing is where the code is stored on the Micro:bit's memory chip, ready to be executed (run)