

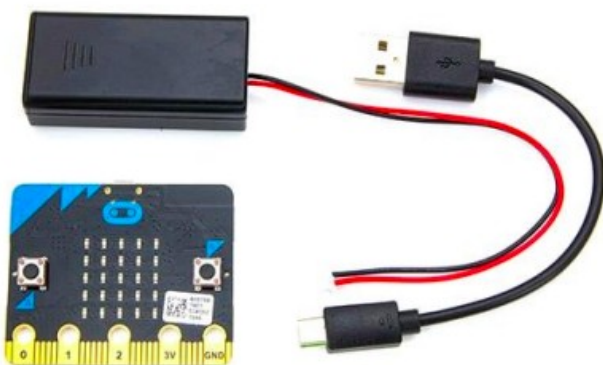
## Micro Bit - User Guide

### Course Introduction:

This is an introduction to coding and computer science by way of making and design, using the micro:bit microcontroller board, and block-based coding. It is project-based learning with a maker philosophy at its core; the idea is that by making physical objects, students will use a context for learning the coding and computer science concepts.

### Lesson structure and teaching plan for using the BBC micro:bit

You will need a device with a usb port, either Windows or Mac based, chromebooks can be used but may require more steps to programme and download to the Micro;Bit. iPads and Tablets will not be suitable for programming and downloading programmes.

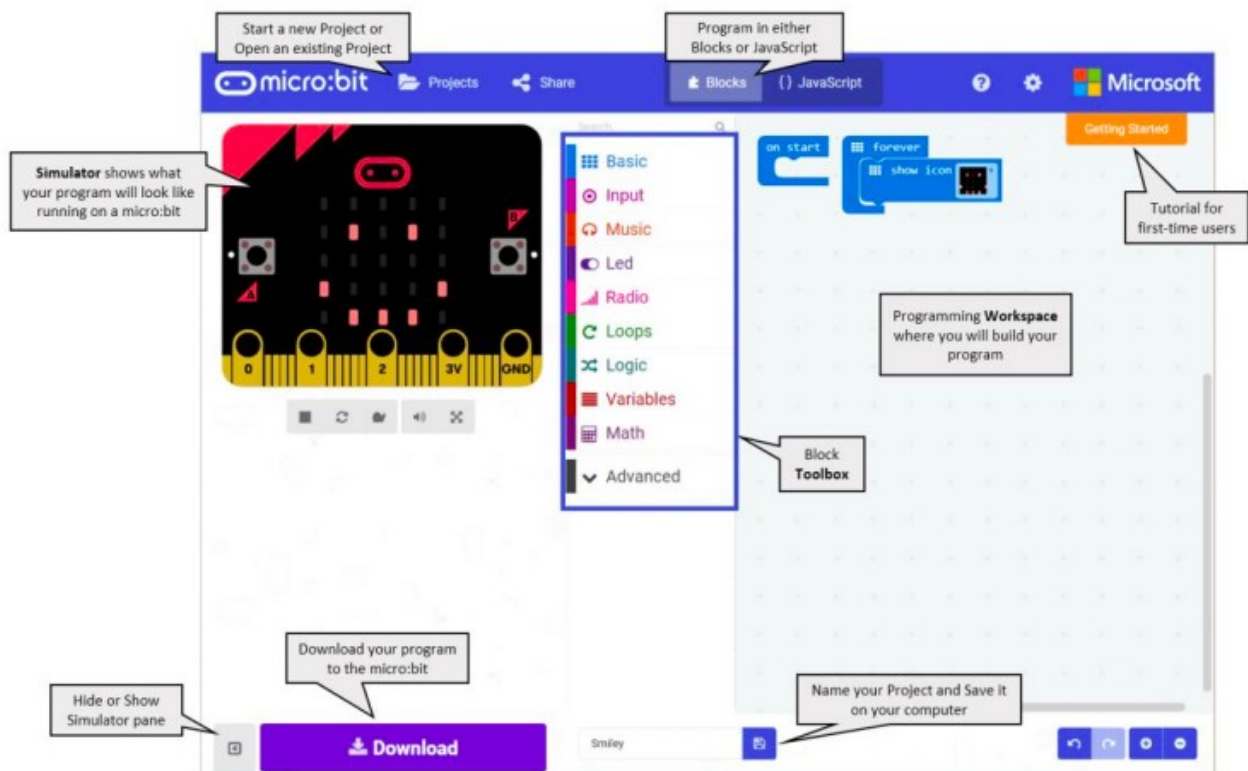


### Starting to use the Micro:Bit:

Students will create a simple program in MakeCode and download it to their Micro:bit using a USB cable. For this activity, students will each need a Micro:bit, a micro-USB cable, a computer, and a 3 volt battery pack.

First steps in using the micro:bit;  
Access the Makecode editor at <https://makecode.microbit.org/#>

You will see this screen which is where you will create code and download it to the Micro:bit You need to become familiar with this screen as it is a fundamental part of using a Micro:bit



### Step 1: Connect It

Connect the micro:bit to your computer via a micro USB cable. Macs, PCs, Chromebooks and Linux systems are all supported. Your micro:bit will show up on your computer as a drive called 'MICROBIT'.

If you are unsure you can use this link to show you how to connect

<https://support.microbit.org/support/solutions/articles/19000013986-how-do-i-transfer-my-code-onto-the-micro-bit-via-usb>

### Step 2: Program It

By using the block-based coding system (text based editors can also be used)

### Step 3: Download It.

Click the [Download](#) button in the editor. This will download a 'hex' file, which is a compact format of your program that your micro:bit can read. Once the hex file has downloaded, copy it to your micro:bit just like copying a file to a USB drive (the downloading process is also called 'flashing') On Windows you can click 'show in folder' and drag the downloaded programme to the Micro:Bit. On Mac, use the Drag and Drop function, drag the hex file onto the Micro:Bit drive

### Step 4: Play It

The micro:bit will pause and the yellow LED on the back of the micro:bit will blink while your code is programmed. Once that's finished the code will run automatically. Each time a new block of code is transferred to the micro:bit the existing code will be overwritten so it is important to name and save your programmes, this is done using the 'projects' and 'my stuff' buttons in the coding section.

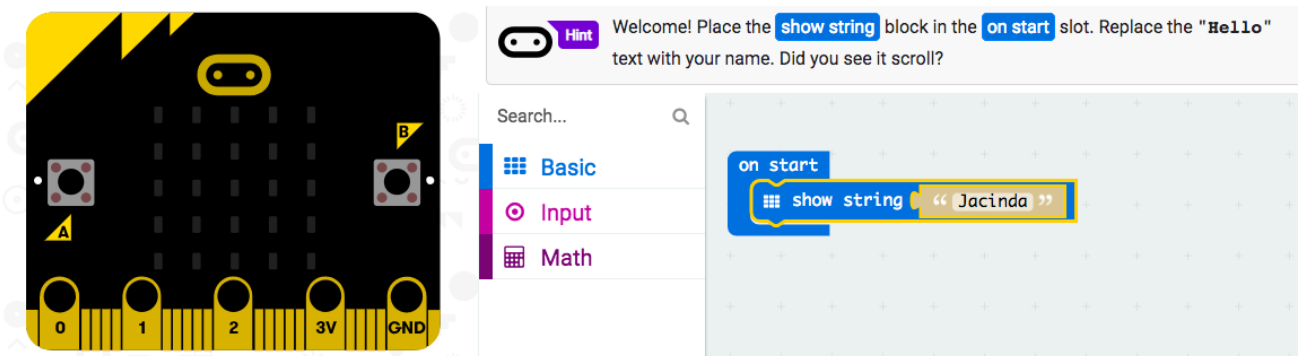
**To begin using the Micro:bit we will go through the four steps above:**

### Step 1: Connect It

Connect the micro:bit to your computer via a micro USB cable. Macs, PCs, Chromebooks and Linux systems are all supported. Your micro:bit will show up on your computer as a drive called 'MICROBIT'.

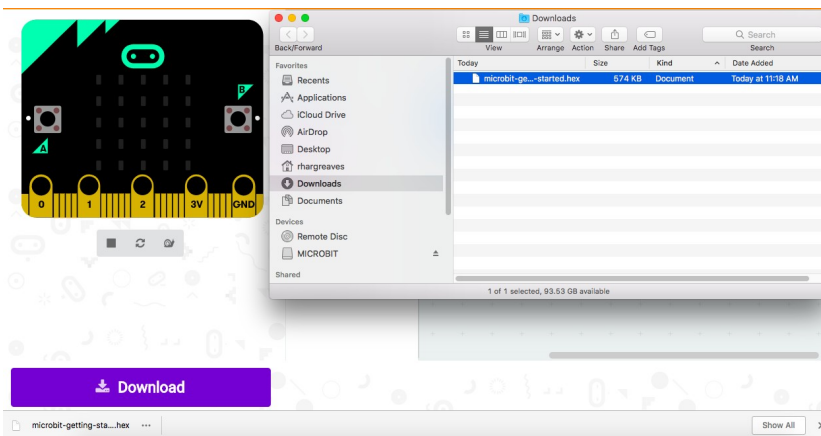
### Step 2: Program It

Go to <https://makecode.microbit.org/#> to access the Micro:bit coding editor, you will see the coding screen with the Micro:bit simulator on the left, this will show you what your code will actually do when flashed to the Micro:bit. Place the 'show string' block inside the 'start' block then replace the "hello" text with your name, you should see your name scroll on the led's (see the screenshot below)



### Step 3: Download It.

Click the Download button in the editor, you will see the programme being flashed and it will appear in the downloads ready to transfer to the Micro:bit. Once the hex file has downloaded, transfer it to your micro:bit. On Windows you can click 'show in folder' and drag the downloaded programme to the Micro:Bit. On Mac, use the Drag and Drop function, drag the hex file onto the Micro:Bit drive

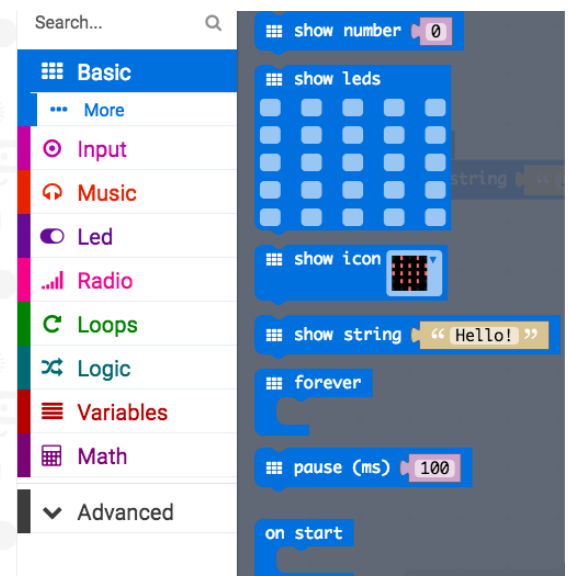
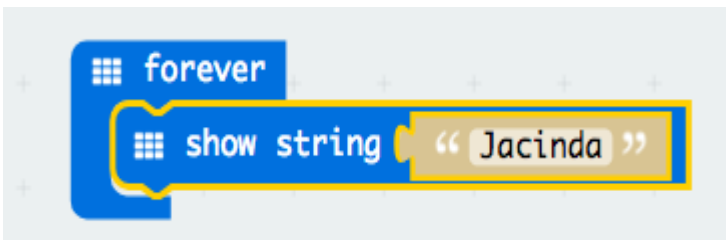


An easy way to see your programme and transfer it to the Micro:bit is to right click on the downloaded hex file, then click 'show in folder' the downloaded hex file will appear then transfer it onto the Micro;bit which is shown under the 'devices'.

The last downloaded programme will be highlighted in blue.

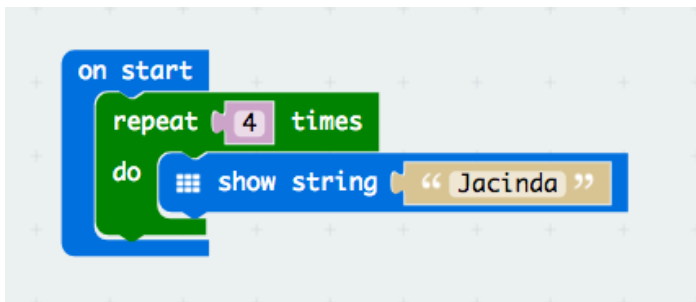
### Step 4: Play It

The micro:bit will pause and the yellow LED on the back of the micro:bit will blink while your code is being flashed. Once that's finished the code will run automatically, the Micro:bit led's will scroll your name but will only do this once, because the code block used was 'on start' To make your name scroll continually you need to use a different block of code so try this;



1. Click on the 'basic' toolbar icon
2. Drag the 'forever' block onto the screen
3. Drag and drop the 'show string' block beneath the 'forever' block
4. The text will continue to scroll across the led's in the simulator

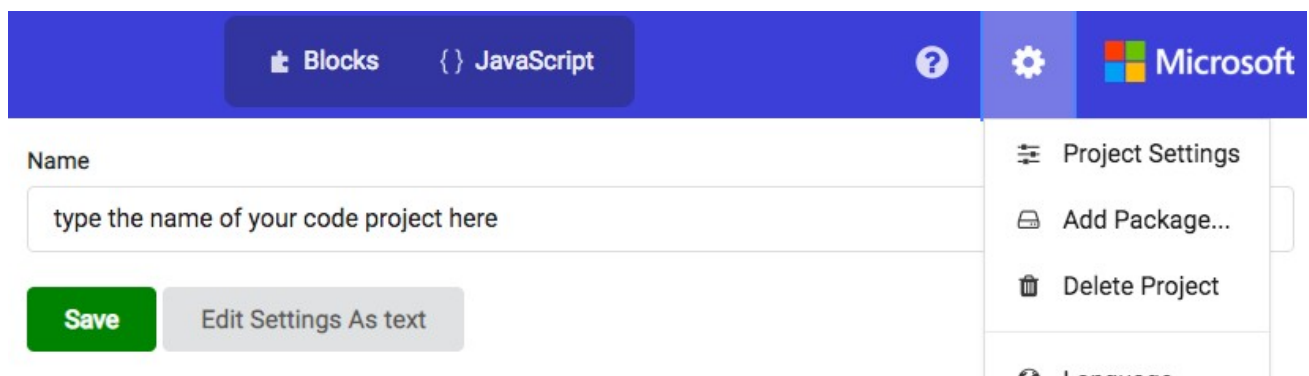
Note: you will notice that the makecode editor uses different colours for each of the toolboxes making it easy to combine blocks, you will use more of the toolboxes as you learn more coding. You can also chose the number of times to scroll the text by using a loop in your code. Select the 'repeat' block from the 'loops' toolbox and insert the 'show string' into it. You will also need the 'on start' block to tell the Micro:bit when to run the code. The repeat number shows how many times the text will be shown.



### Saving your programmes:

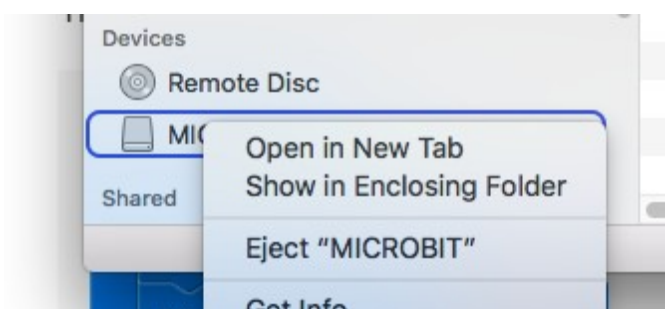
Each time a new block of code is transferred to the micro:bit the existing code will be overwritten so it is important to name and save your programmes, to save your programme follow these steps:

- On the coding screen click on the settings icon (the cog wheel to the right of ?)
  - Click on 'project settings' - the screen that is shown below will appear
  - Type the name of your project/code into the text box
  - Click on 'save' - the project will show in downloads with the name it has been given
- You can also make a new folder inside downloads and save these programmes inside it, then you will have all of your Micro:bit programmes/projects in one place



### Disconnecting the Micro:bit

When you have finished using the Micro:bit it needs to be disconnected from the computer, follow the steps below to do this safely



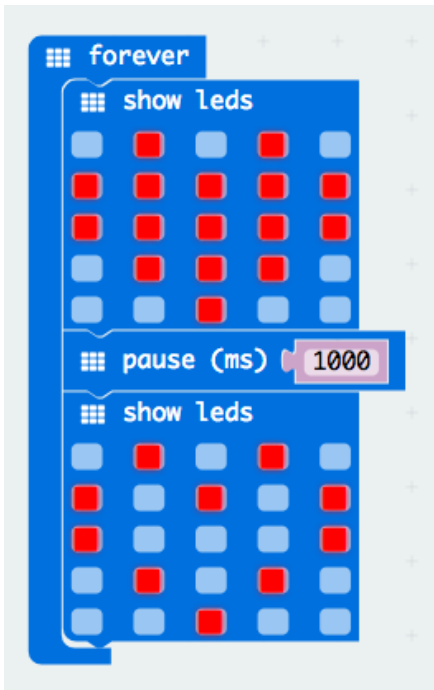
When you have finished working with the Micro:bit it must be disconnected from the computer by ejecting it. Right click on the Micro:bit location and select 'Eject MICROBIT' wait until the Micro:bit disappears from the screen, after this it can be safely removed from the usb drive

## Next coding projects:

### 1/ Flashing Hearts

After you have successfully coded the name text using the led's go on to the next project where you will learn to display different patterns by flashing the led's using code blocks

#### Step 1

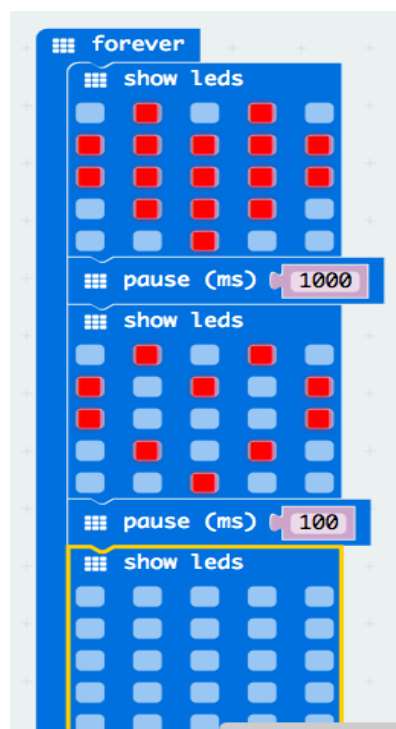


- Click on the led's you want to display, the selected led's will show as red
- Add a pause between the led displays
- Add a second led display
- The displays will flash alternatively

Download the code to the micro;bit

#### Step 2

- Add more led displays and time pauses to create an animation

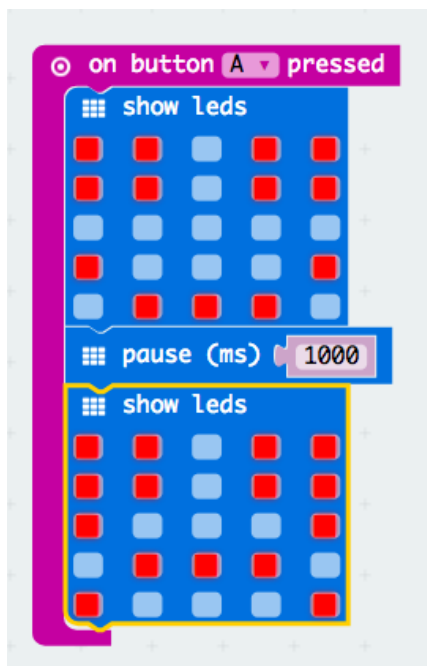


## 2/ Smiley Buttons

So far your Micro:bit has operated using only code which has been flashed from the computer, the next step is to use an input to make the Micro:bit operate.

The micro:bit has two push buttons (button A on the left side and B on the right side) which can be used as input switches, the switches are used to 'turn on' the Micro:bit, just like you press a button and turn on a microwave or any device, the next project will show you how to incorporate these buttons into your code.

Any blocks of code can be made to operate using input switches so after coding the smiley buttons try including text and led's together to show messages.

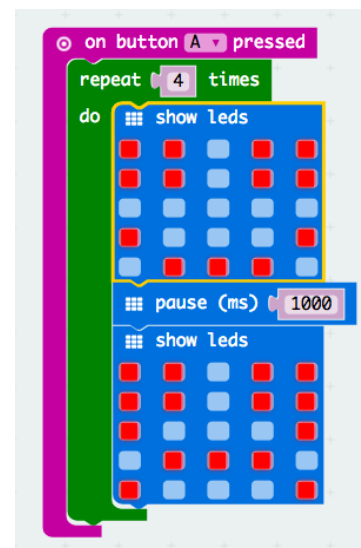


Step 1:

- Add a 'button pressed' command from the 'input' toolbox
- Select 'button A'
- Add the blocks of code for led's to display
- Make a smiley face
- Add a time pause
- Add a sad face

Step 2:

Add a 'repeat' command from the 'loops' toolbox to show the display a set number of times



## 3/ Radio transmission

Micro:bits can send and receive messages sent by radio. Messages can be either text or numbers. Just like a real radio network, each micro:bit within the network must use a similar 'group identification' and in the micro:bit's case this will be a number. A group can have any number of participants within it.

These are the steps to enable radio messages to be sent to different micro:bits within a group;

### Setting a channel number

If you only want share messages within a group of micro:bits then each micro:bit in the group must be configured to share the same channel number. The channel number must be a number between 0 and 100. You can do that like this:



- Select the 'radio set group' command from the 'radio' toolbox
- Each micro:bit that will be included in the group must have the same number so set the numbers in the 'radio set group' to the same

- number
- if all radio groups within the room use the same number then they will all be in the same group so chose a random number then if you pick up another groups messages change to a different number)

- It is important to do this if you are in a room with other people using their micro:bits because otherwise your micro:bit will overhear all the messages nearby and that is not what you want generally.

Finally, you should set the power level for the radio, by default, your micro:bit will be transmitting on power level 0 which means that your messages might not get transmitted very far. If you are transmitting a short distance of say 2 - 5 metres then level 0 will be OK so you do not need to adjust the power level (which can be a value between 0 and 7)

### **Sending and receiving a message**

Now you are ready to send and receive a message. A string is the name for a block of text, you can send a string which is up to 250 characters in length in the message. Here is an example of a radio send and receive block code.

```

on start
  radio set group 1
  radio set transmit power 7

  on radio received receivedString
    show string receivedString

  on button A pressed
    radio send string "type message here"
    show string "sending"
  
```

Explanation of the code:

- 'On start' means run the code when the Micro:bit starts up
- 'radio set group 1' sets the micro:bit to group number 1 (all senders and receivers must be set to the same number)
- Then set the transmit power
- 'Radio set transmit power' means set the power to 0 (between 0 and 7, the higher the number = greater distance - leave at 0)
- 'On radio received' means display the message that is received
- 'Show string' means show the text of the received message
- 'On button A pressed' means do something when button A is pressed
- 'Radio send string' means send a radio string of text
- 'Show string sending' means show that the message has been sent

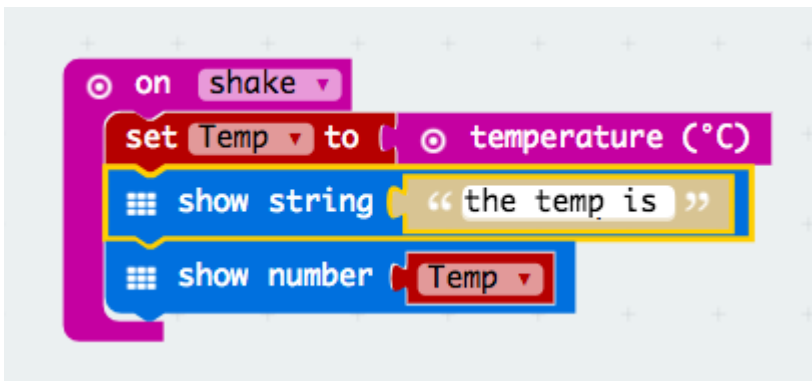
You can use one block of code like this to both send and receive messages.

Try sending and receiving messages using other button

You can continue sending and receiving messages by using these blocks of code

4/ Display the temperature

The Micro:bit has a sensor which detect and display the air temperature. So that the temperature display can be part of a multiple code programme an input can be used that when the micro:bit is moved then the temperature will be displayed, the code example below will show the current temperature when micro:bit is moved.



```
on shake
  set Temp to temperature (°C)
  show string "the temp is "
  show number Temp
```

The image shows a Scratch-style code editor with a light gray background and a grid of small plus signs. The code is contained within a purple 'on shake' block. Inside this block, there are three other blocks: a red 'set Temp to' block with a dropdown menu showing 'temperature (°C)', a blue 'show string' block with a text input field containing 'the temp is ', and another blue 'show number' block with a dropdown menu showing 'Temp'.

Display temperature code example