Basics of coding with BBC Microbit

Tapio Hansson

Micro:bit is a tiny computer made by BBC for british pupils. Device got very popular and can be bought online separately or in large quantities. It costs only around 10 euros and thus it has only very few features. Clever coder and engineer can still make lots of things with it.

1 The Device

The processor of the micro:bit is 16 Hz ARM microcontroller. You certainly won't do any complex computing with it, but it is well enough to control few pins, input and output. There is 256 kB flash memory for the programs that are loaded to the device.
Similar devices have existed for a long time, but micro:bit has small user interface the more general devices usually lack. The UI consists of

- 5×5 led matrix, which can be used as a display.
- 2 fully programmable buttons labeled A and B.
- Accelerometer to detect if the device is shaken or dropped.
- A compass that can detect if the device is turned.
- 23 pin connectors to connect additional devices. Five of them are bigger so they can be used with crocodile clips.

2 Getting started

There are two different ways of coding with micro:bit. One can use either Python programming language with micro:bit libraries, or a graphical programming language created for the micro:bit. In this tutorial we use the graphical tool.

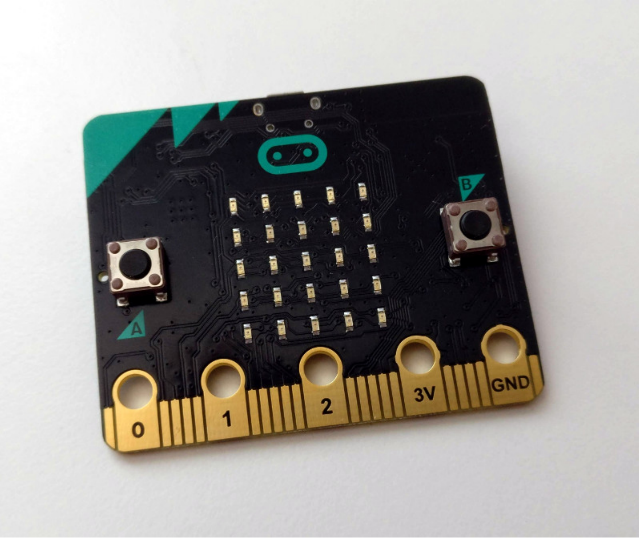


Figure 1: Micro:bit from the user interface. Led matrix is in the center and two programmable buttons labeled A and B are on it's sides.

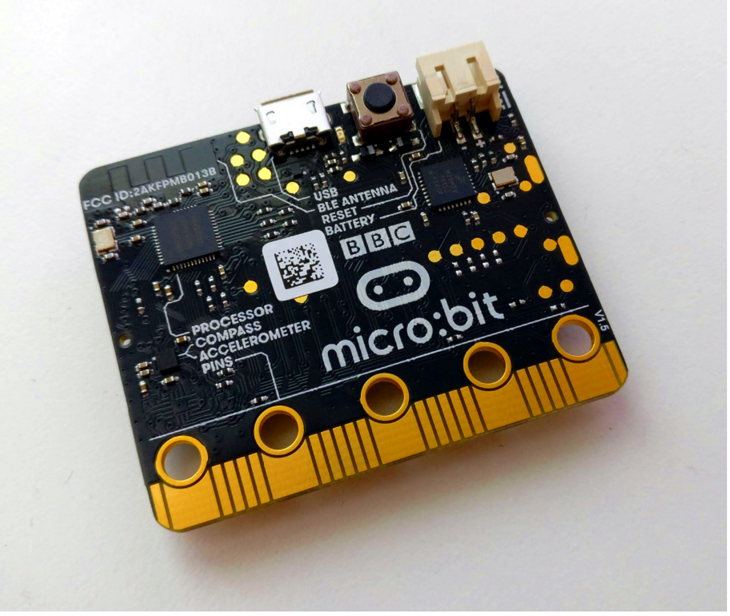
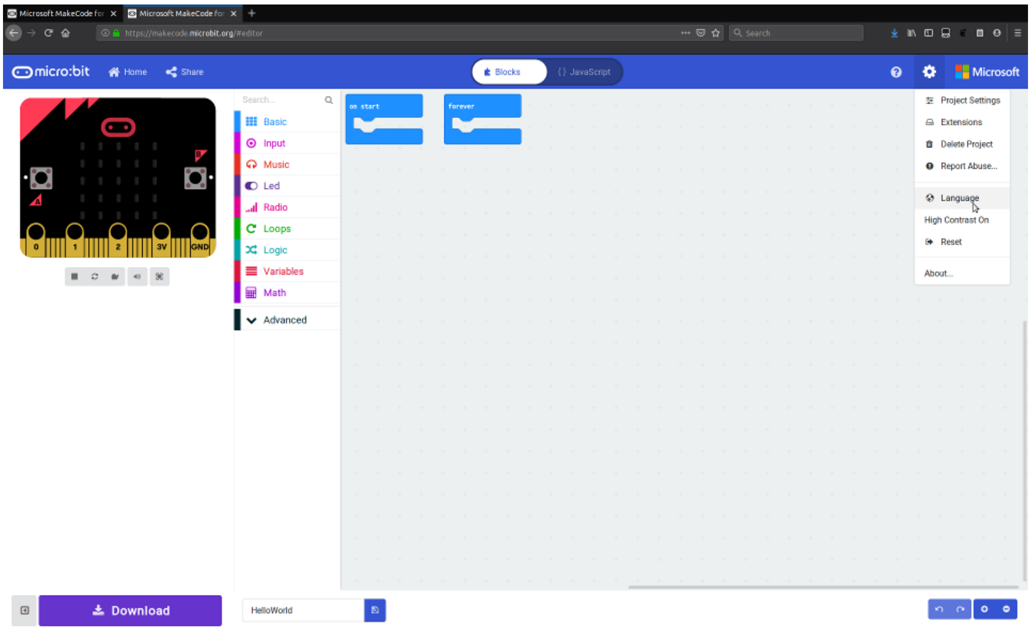


Figure 2: Microbit from behind. On the top are usb-connector, reset button and the power connector.

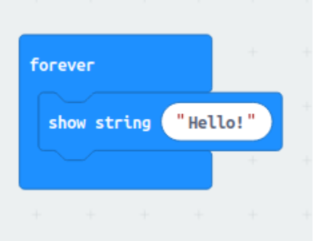
The online editor can be found from https://makecode.microbit.org/#editor. From the link you should find yourself in the micro:bit editor. The language can be chosen from the settings menu in the right hand corner.



The code consists of blocks that can be drawn to code area. In an empty project there are two blocks on the table labeled "on start" and "forever". All the code blocks can be used from the list on the left. There is also a micro:bit-emulator on the coder, so you can actually test your code without a physical device.

Let's create our first simple program.

Practice! 2.1. From the list choose Basic, and from there pull the block called "show string "hello!"" inside the forever-block on the table. Make sure the blocks connect properly.



After that, you should notice that the emulator should now be displaying the text. You can of course change the text to display something else. Name your project from the bottom and press the download button.

The code is transferred to the micro:bit via usb. The program is downloaded in a special hex-format which is precompiled for the micro:bit and works specifically on that device. After you have downloaded the program, connect the micro:bit to your computer. It should show up as an external storage device and the code can be transferred simply by copying it to the device. After copying .hex-file to the micro:bit you should see the yellow led in the device blinking for a while and after that the display should show ”Hello!” text repeatedly. As you may have noticed the micro:bit does not need batteries if the usb cable is attached, but if you want to show your nice text to the world you may also attach the batteries and the text keeps running even if you take off the usb cable.

3 Basics blocks of programming

There are not that many basic structures in programming. Especially in small and simple programs most of the code consists only on *variables*, *loops* and *conditional statements*. Loops are designed to repeat some code until some condition is fulfilled. Computers are especially powerful tools to do something boring many times, so loops are important tools. Conditional statements are only run if some condition is fulfilled, so you can create more complex programs with them.

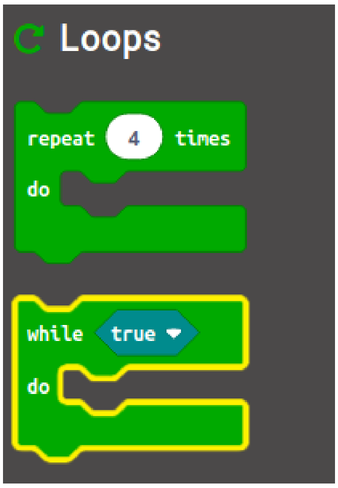
3.1 Microbit loops

When you open an empty microbit codesheet you can see two empty blocks on the screen. Other says ”On start” and the other ”forever”. These are the simple blocks on the microcontroller so everything you put inside these loops are run either once when you power up the microbit or looping forever while the power is on.

Practice! 3.1. Try it yourself! Move the ”show string” -block from your previous code to ”on start-block” Download the code to microbit and power it up.

3.2 For and while loops

Two most useful loops are for and while. You can find them from green ”loops” menu on microbit editor.

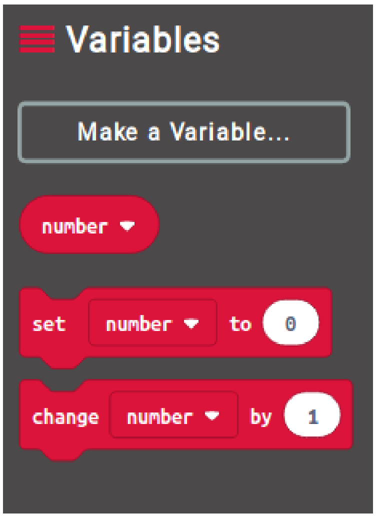


In microbit for-loop is named ”repeat”. It is somewhat self-explanatory since the point is to repeat something as many times as is defined in the block. While-loop is somewhat similar, but you simply repeat what is inside of the loop as long as the statement is true. You can find some logical statements from ”logic”-menu.

Practice! 3.2. Let’s change the hello world we created earlier. Pull the green ”repeat”-block inside ”on start” block. Define how many times you want to print the text and pull the "show string"-block inside the repeat-block. Download and run.

3.3 If-clause

Probably the most used block of code especially in the beginning of your programming career is the if clause. It simply implies that if some condition is true, then we do something. If-conditions can also be found from ”logic”-menu. To use if-conditions effectively we need to create a variable. Variables are a way to store some data. Variables can be created from ”Variables” menu. You have to give your variable a name first. We can use for example ”number” as a name, but you can use whatever name you want. As you can see after you create the variable three blocks appear.



There are blocks to insert the variable into the code and blocks to set and change the value of the variable. Let's try to use them.

Practice! 3.3.

- Open a new codesheet. Create a variable called number and set your variable to 8 on start.
- From Basic-menu pull a block called "show number" into the forever-block.
- From variables-menu pull your variable to place of the zero in show number-block.

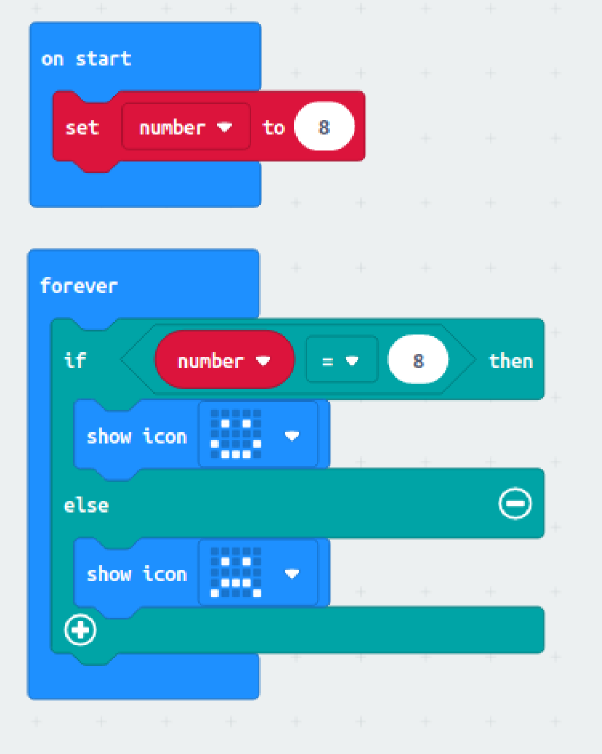
Now you can change the number you want to show by setting it to some other value on start. Let's build an if-condition with our variable.

Practice! 3.4.

- From logic menu pull an if statement into the forever-block.
- From logic menu pull a comparison to your if-block. Set the comparison to be "If number = 8".
- From Basic-menu pull a "show icon"-block into your if-block. Set it to be a smiley face for example.
- Add an else to your block, from a small plus-sign on the bottom of the if-block.

- Pull another show icon-block to your else statement, and set it to show some other icon.

Your code shoulde look like this one:



```
on start
    set number to 8

forever
    if number = 8 then
        show icon
    else
        show icon
```

When you run it, you should see only the smiley-face since we set our variable to be 8 which is also the condition for smiley. Let's add some functionality.

- From basic-menu pull a pause-block under both icons you have in the code. Set time in the pause-block to be 2000 ms. This will stop the code from running for 2 seconds.
- From variables-menu pull "set variable"-block under your smileyface and set your number to something else than 8, for example 3.
- Do the same thing in the else-block, but there set the number to be 8 again.

Now when you run the code you should see your microbit change icons every 2 seconds.

4 Input

Last thing we cover here for now is input mechanisms. As you know now, we mostly use the led display as output, though you could connect some other devices, such as motors or sensors, to the connector pins on the board. There are a few ways you can give information to microbit.

4.1 Buttons

There are two input buttons on the board, labeled A and B. Those can easily be programmed for various functions. You can find input mechanisms from ”input”-menu in the codesheet. Buttons can be used with block called ”on button A pressed”. You can choose which button you want to configure.

Practice! 4.1. Drag the ”on button pressed”-block to codesheet. Make the microbit display something, for example a smiley face when you press the button A. Drag another block to the sheet and make the other button clear the led screen.

4.2 Accelerometer and sensors

Microbit also has built-in sensors. Especially accelerometer is useful, since you can program microbit to do things when it is shaken or dropped. Accelerometer can be accessed similar to the buttons. From input-menu drag

5 Rock, paper, scissors

As an exercise create a microbit program which plays classic game rock, paper, scissors. Your microbit should choose randomly one of three images to display, which represent either rock, paper or scissors. The choosing should be done on shake.
Tips:

- Random numbers can be generated from menu called ”math”.
- You might want to assign a variable where you store a random number, and then display something depending on the value of the number.

5.1 Masterclass: Lan party

Microbit has also a radio which can be used to talk to other microbits. These functions can be found from menu ”Radio”. Now that your microbit can play rock, paper, scissors, create a microbit-program that can play the game and then compare results with another microbit to see which one wins. Then display the result of the game on the led display.
Tips:

- You should send the information of the random pick on radio after you have shaken the microbit.
- Your microbit should do a comparison of values when it recieves something from another microbit.