1. What is the BBC micro:bit, and how can we program it?

The BBC micro:bit is a small microcontroller which contains a range of built-in sensors, buttons and LEDs. The micro:bit is programmed via the computer's USB port and a standard web browser. The software used to develop programs for the micro:bit is developed by the micro:bit foundation, in cooperation with selected partners in the IT industry. Using these tools the micro:bit can be programmed via visual drag-and-drop tools, but also using either the text-based languages python or JavaScript. In this booklet we will be using the Microsoft PXT editor. This editor is currently being heavily developed, so some of the figures in this booklet might be out of date. The online PDF is kept updated to reflect any changes.

The micro:bit is a small circuit board that we can program. Using the built-in sensors and LEDs its possible to create fun and engaging mini-games using only a USB cable and a computer. The micro:bit is made so that no additional software needs to be installed on the computer, other than a web browser. On the front of the micro:bit we will find 25 programmable LEDs, and two tactile push buttons.

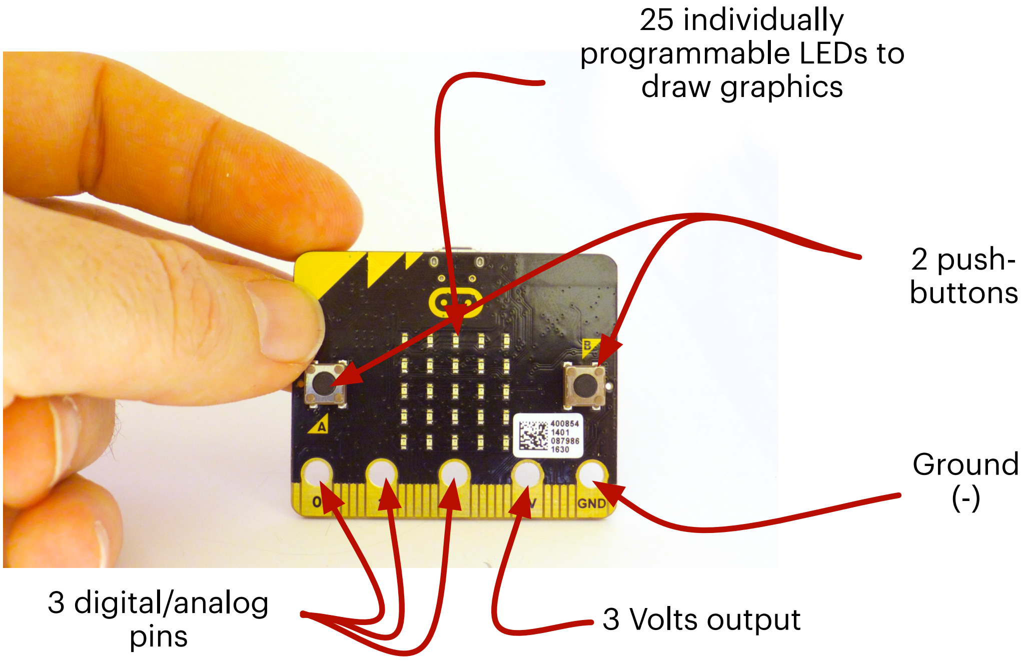


Figure 1 – The front of the BBC micro:bit

On the back of the micro:bit we will find the processor, the Bluetooth antennae, and the USB and battery ports. Along the bottom edge we will find the 20-pin edge connector which we can use to connect the micro:bit to electronic components and circuits.

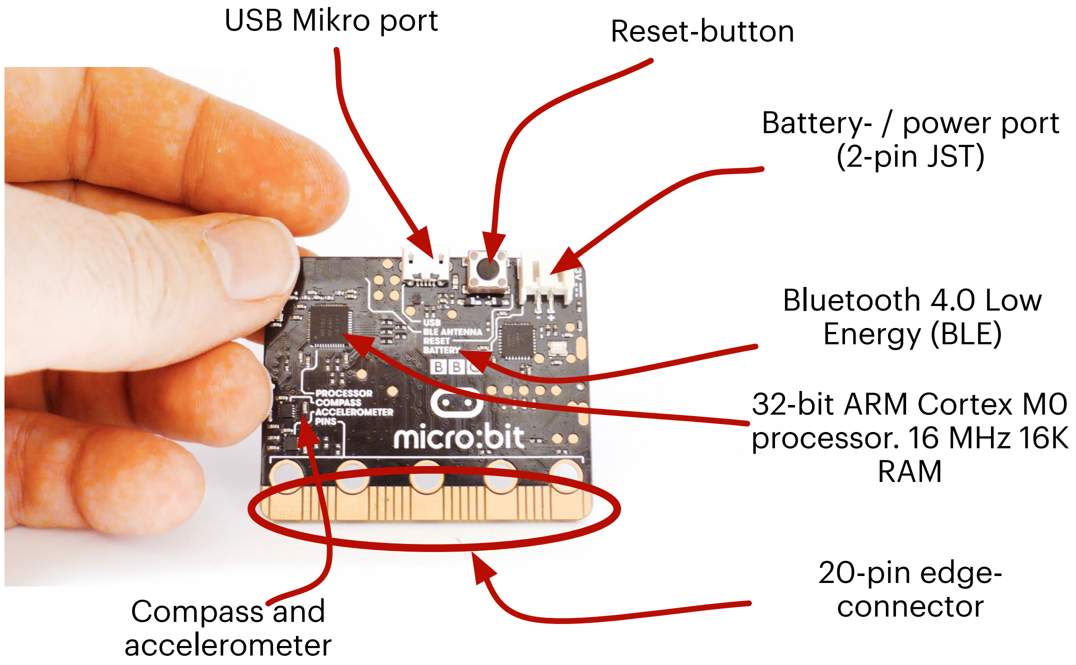

USB Mikro port
Reset-button
Battery- / power port (2-pin JST)
Bluetooth 4.0 Low Energy (BLE)
32-bit ARM Cortex M0 processor. 16 MHz 16K RAM
20-pin edge-connector
Compass and accelerometer
micro:bit

Figure 2 – The back of the BBC micro:bit

Along with the micro:bit we will be using the PXT editor in order to create a game that we can program the micro:bit with.

1.1. How is the micro:bit programmed?

When the micro:bit is connected to a computer it becomes available as an ordinary USB drive. On this drive we will find two files, a DETAILS.TXT file containig some basic information regarding the micro:bit, as well as a MICROBIT.HTML file that we can open in the browser.

Because the micro:bit operates as a microcontroller, we can upload compiled programs to it. The micro:bit will then overwrite any existing programs and start running the new program. The micro:bit is programmed by dragging a pre-compiled .HEX file into the micro:bit USB drive. Luckily, the online tools will create this .HEX file for us. Whenever we want to upload the program we have written to the micro:bit we can simply download and store the .HEX file from the PXT editor. To get started lets navigate to microbit.org and open up the PXT editor.

Once the PXT editor is loaded we are presented with the programming environment. This is the block-based drag-and-drop programming tool which we will be using throughout this booklet. The editor consists of five main areas, where the micro:bit simulator, the block palette and the scripting area take up most of the website.

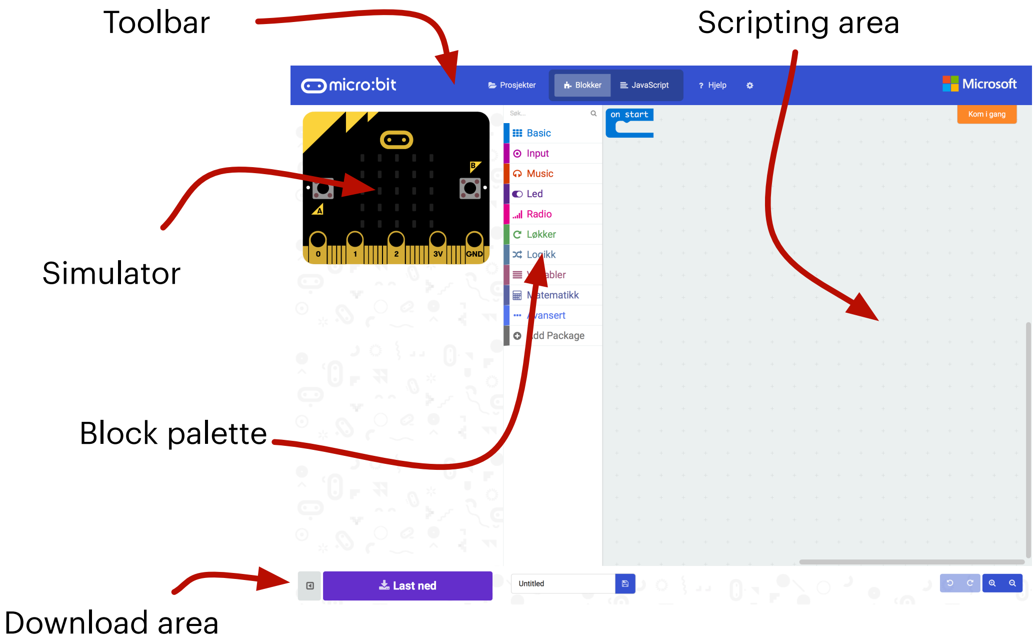


Figure 3- The Microsoft PXT editor

- **Toolbar** – This area is located at the very top of the editor. We can use the toolbar to create new or import existing projects, as well as choosing between the visual drag-and-drop editor and the text-based JavaSript mode. Using the toolbar we can also bring up the help-menu and change the settings
- **Block palette** – In this area we will find all the blocks that we can use in order to program the micro:bit. The blocks are divided into color-coded categories in order to make it easier to find the blocks we want to use
- **Scripting area** – We will drag blocks form the block palette to the scripting area to program the micro:bit.
- **Simulator** – In this area the PXT editor will attempt to show what we can expect to see and happen on the micro:bit. The simulator is updated whenever we add, delete of move a block in the scripting area
- **Download area** – At the bottom of the website we will find the download area. Here we can give our project a name, or choose to download the .HEX file we need to program the micro:bit with

Now that we have learned a little bit about what the BBC micro:bit is, while also introducing the Microsoft PXT editor, it is time to create our very first program for the micro:bit - Launching the spaceship.

2. Launching the spaceship

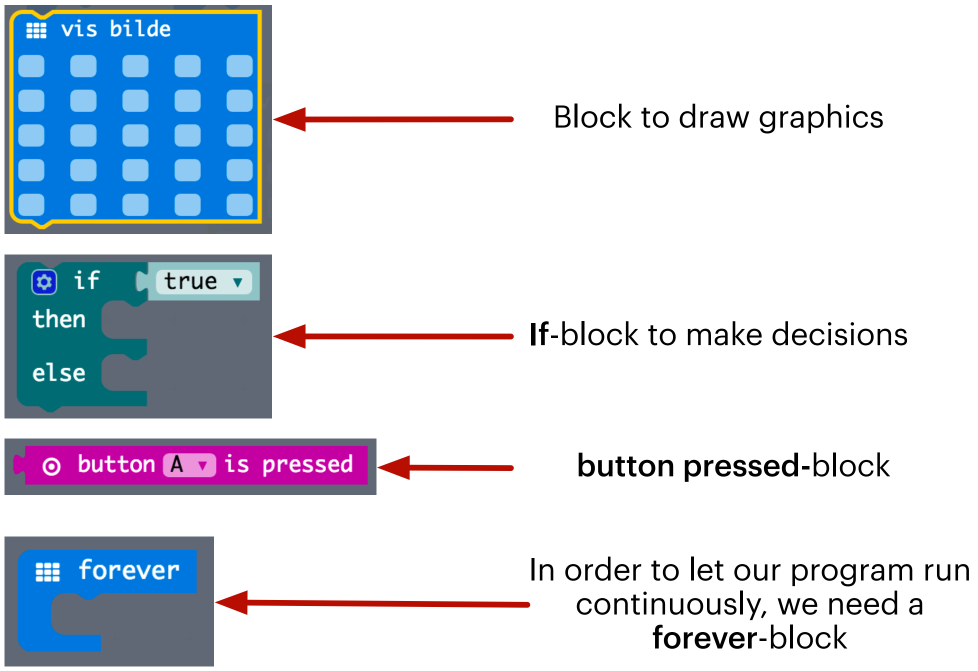
We will get started with programming the micro:bit by making the micro:bit draw simple graphics on its built-in LED screen. The word "graphics" might be a word you haven't heard before, so you might be wondering what it means. In essence the word "graphics" means "anything that is drawn or written". In the context of the micro:bit we mean what we can "draw" using the micro:bit display - utilizing the 5 rows and 5 columns of LEDs on the front of the micro:bit.

In this chapter we will learn to:

- Draw a small spaceship to the micro:bit display using the blocks available in PXT
- Expand out program so that the spaceship can lift off and launch upwards
- Expand the program so that we only start the lift-off process once we click on the A-button to the left of the micro:bit display

2.1. Draw the spaceship on the micro:bit display

In order to complete this chapter, we will use the following blocks:



Block to draw graphics

If-block to make decisions

button pressed-block

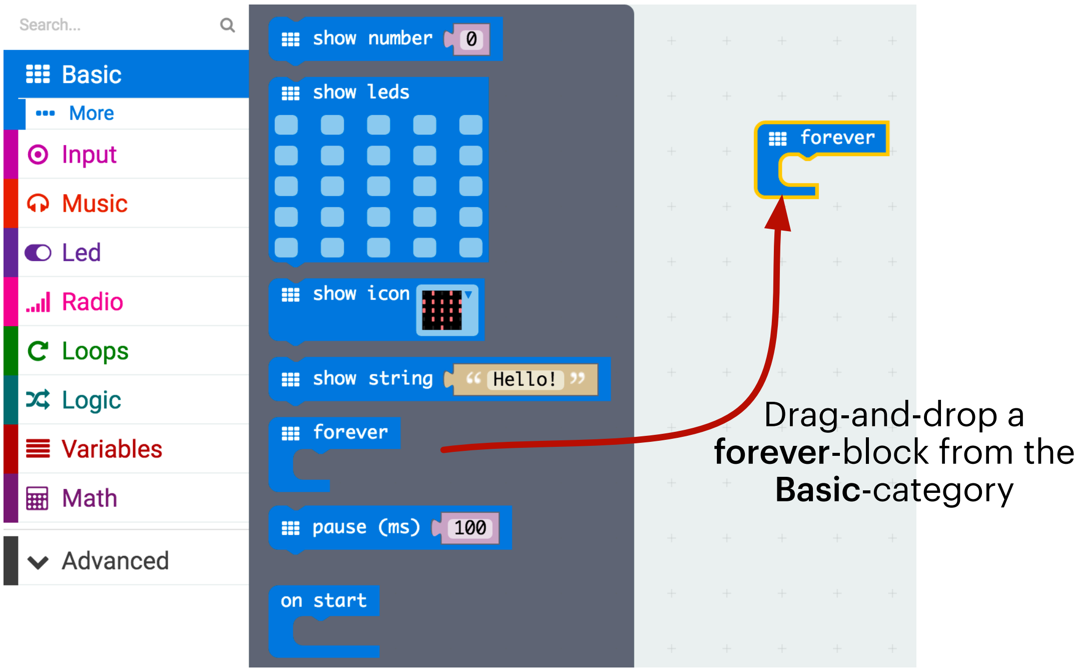
In order to let our program run continuously, we need a **forever**-block

Figure 4 - We will need these blocks to complete chapter 2

Due to the size of the micro:bit display, we have a fairly large limitation as we only have 5x5 LEDs to work with. Therefore, in order to draw something that looks like a spaceship, we need to find our creative side and think in simple terms. One way to draw the spaceship would be to use a single LED, but in order to draw something that looks somewhat like a spaceship, we are going to draw using 6 LEDs.

But before we can draw on the screen, we need to decide if we want to draw our graphics once, or *forever*.

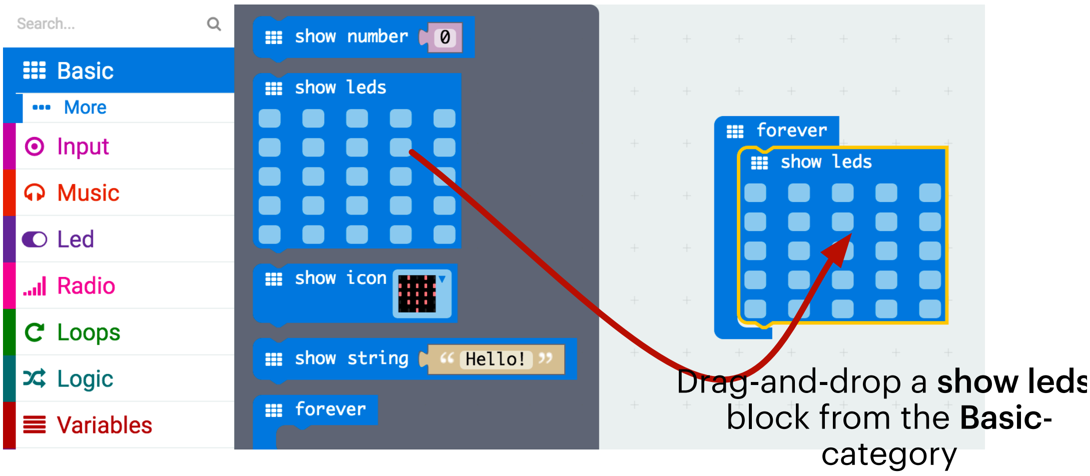
As we would like to later expand our game and make the spaceship move, we will start by dragging a *forever*-block from the block palette to the scripting area. The PXT editor might already have placed a *forever*-block in the scripting area, in which case, we can skip the next step.



Drag-and-drop a **forever**-block from the **Basic**-category

Figure 5 – We will start by dragging a forever-block to the scripting area

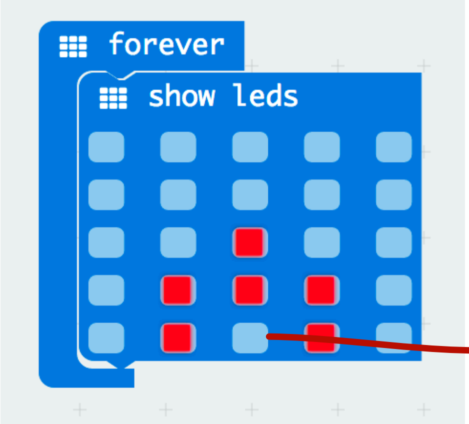
Once we have a `forever`–block in the scripting area, we will drag out a `show leds`-block and drop it inside the `forever`-block. This will redraw the graphics forever on the micro:bit display.



Drag-and-drop a **show leds** block from the **Basic**-category

Figure 6 – Drag out a show leds-block to the forever-block

Once that is complete, we can draw our spaceship. As we mentioned, we can draw the spaceship using only 6 LEDs. We can choose which LEDs to light up by simply clicking on the light-blue squares inside the *show leds*-block. This will toggle the LEDs on and off, turning the LEDs that are on red.



Draw a spaceship by marking these 6 LEDs in the **show leds**-block

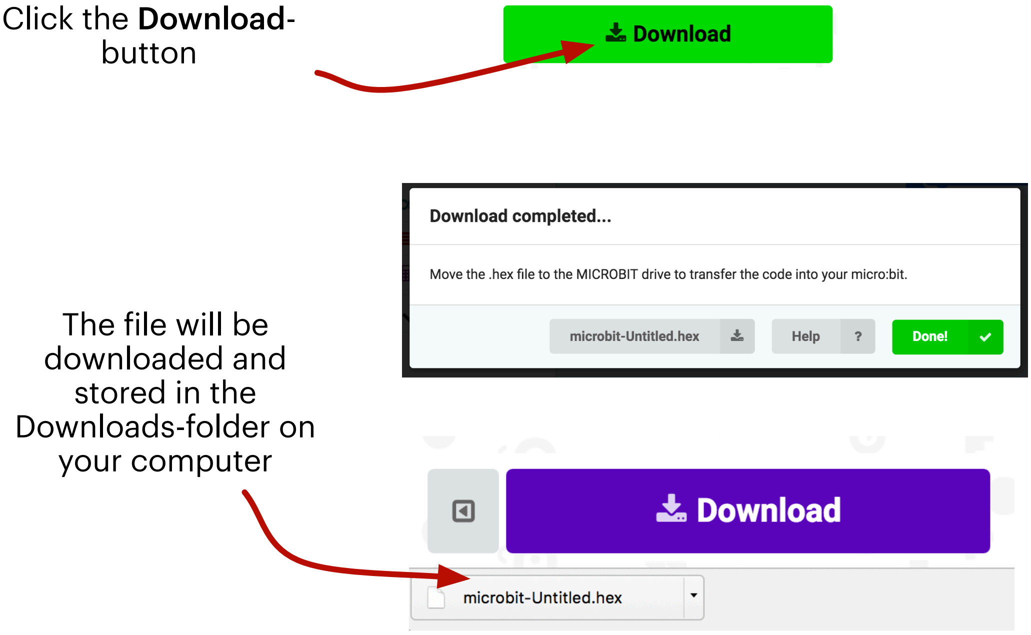
Figure 7 – Draw a spaceship on the display using the show leds-*block*

Now that we have the blocks we need to draw the spaceship, we need to upload our program to the micro:bit. In order to achieve this we need to:

1. Click on the **Download**-button in order to make the website create the HEX file and download it to your computer
2. Copy this HEX file to the micro:bit using the file system on your computer
    1. If your web browser have a "save as..." button, this can be used to save the file directly to the micro:bit.
    2. your web browser can also be configured to always ask you where you want to download files to

Depending on which browser you are using and your operating system (Windows, Mac or Linux, Internet Explorer, Chrome, Safari or Firefox), the process of uploading the HEX file to the micro:bit will be a little bit different.

The first step, however, is the same. We will get started by clicking on the **Download**-button. This will downlod a file which - on most browsers and operating systems - will have a .hex file ending.



Click the **Download**-button

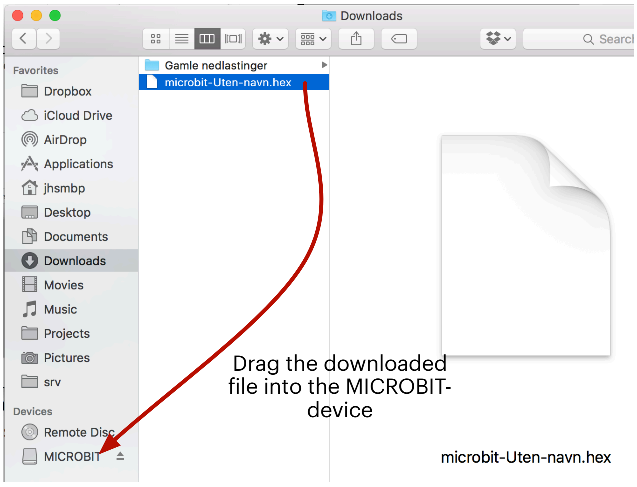
The file will be downloaded and stored in the Downloads-folder on your computer

Download completed...

Move the .hex file to the MICROBIT drive to transfer the code into your micro:bit.

Figure 8 – Download the .hex file that the micro:bit needs

This will download any file to our computer. The name of the file is not important, and regardless of its name, this is the file we need to upload to the micro:bit. We can upload it by navigating to the Downloads-folder on your computer and simply drag it over to the MICROBIT USB device that should be mounted as a new drive on your computer.



Drag the downloaded file into the MICROBIT-device

Figurw 9 – Drag-and-drop the HEX file over to the MICROBIT device

This will start the copy-process. Once the file is copied over to the MICROBIT device a couple of things are going to happen on the micro:bit.

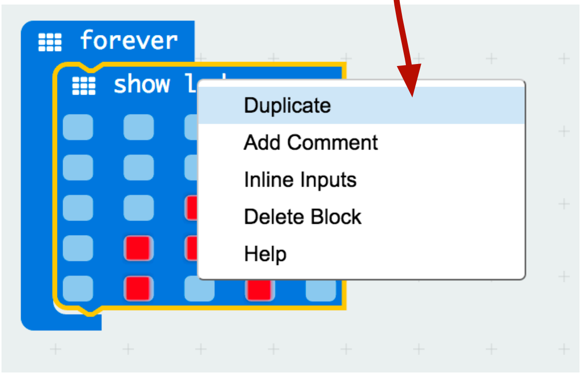
1. The micro:bit will move the program into inte internal memory (RAM). This will stop anything that is currently happening on the micro:bit and the display will be off until this process is complete
2. The micro:bit will then restart. At this point the MICROBIT-device will reappear. Some operating systems will at this point complain that the USB device was not ejected properly. You can safely disregard this messge
3. The micro:bit will now run your new program

Once the file is copied over and the micro:bit have had time to restart, we can see our spaceship appear on the display. Congratulations on your very first micro:bit program!

Next up, we will make the spaceship lift off and fly upwards on the display.

2.2. Make the spaceship launch

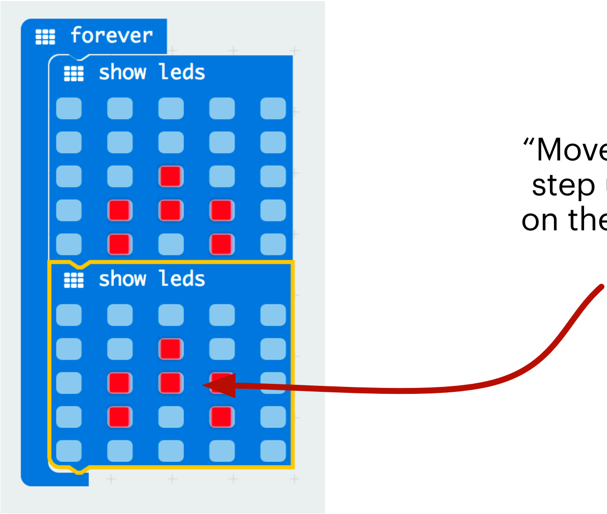
Now that we are going to get the spaceship to launch off the ground, we need to add a few more blocks to our program. We can achieve this by copying the `show leds`-block five times. We can then reposition the spaceship to make it lift off and fly upwards above the display. To copy the block we can right-click on it and choose `Duplicate`.



Copy the **show leds**-block by right-clicking and choosing **Duplicate**

Figure 10 – Duplicate the show leds-block

The result is a new copy of the block we duplicated. To attach it to the previous block, we simply drag-and-drop it into place. In this new block, we will manually move the spaceship one pixel up by deselecting and selecting which LEDs that are lit.

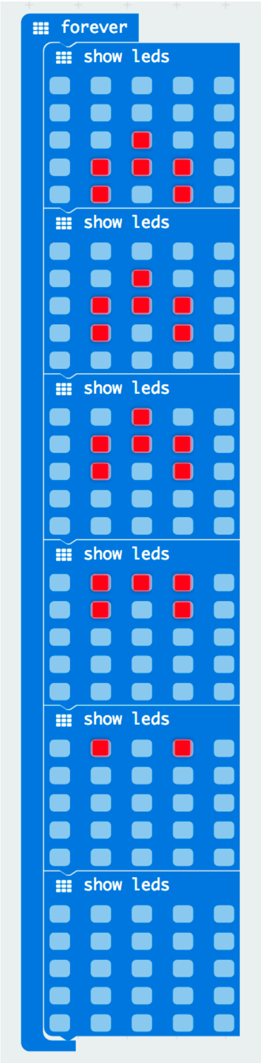


"Move" the spaceship one step upwards by clicking on the blue or red squares

Figure 11 – Move the duplicated block and connect it to the previous block

We can now view the result in the PXT simulator before uploading the new HEX file to the micro:bit. The result is a short animation with the spaceship moving upwards one pixel before moving back down.

We are on our way! Let's continue to copy and move the spaceship until it disappears above the screen.



Keep duplicating the blocks and move the spaceship upwards until you end up with a blank display

Figure 13 – Continue to copy the spaceship and move it upwards one pixel at a time

Lets upload and copy the new HEX file to the micro:bit and see that the spaceship lifts off and disappear above the screen. So far we have created an animation, but in the next step we will alter the program slightly so that the spaceship wont lift off before we click the A-button to the left of the display.

2.3. 3, 2, 1... Lift-off

In order to expand our program to listen to the A-button being pressed we will also, at the same time, change the flow of the program a little. So far, we have not yet used either of the buttons or the sensors on the micro:bit in order to