

RGB_breathing_light

1.Learning goals

In this lesson, we mainly learn how to control the color of RGB by micro:bit and Super:bit expansion board and achieve the effect of breathing lights.

2.Programming method

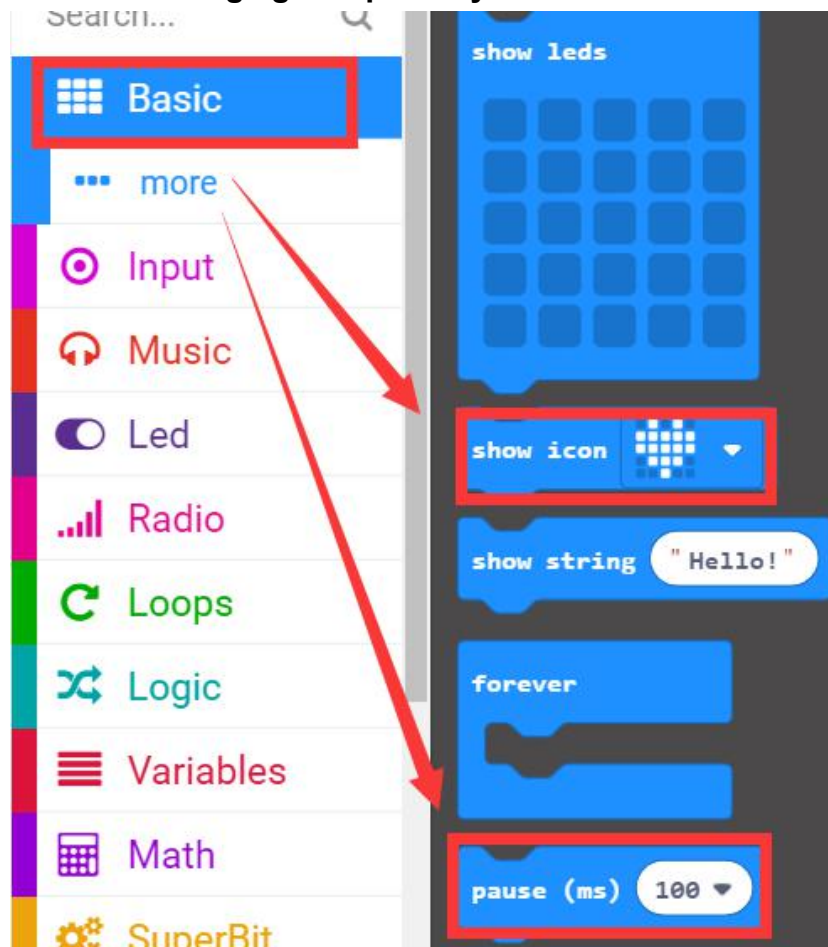
Mode 1 online programming: First, we need to connect the micro:bit to the computer by USB cable. The computer will pop up a USB flash drive and click on the URL in the USB flash drive: <http://microbit.org/> to enter the programming interface. Add the Yahboom package <https://github.com/lzty634158/SuperBit> to program.

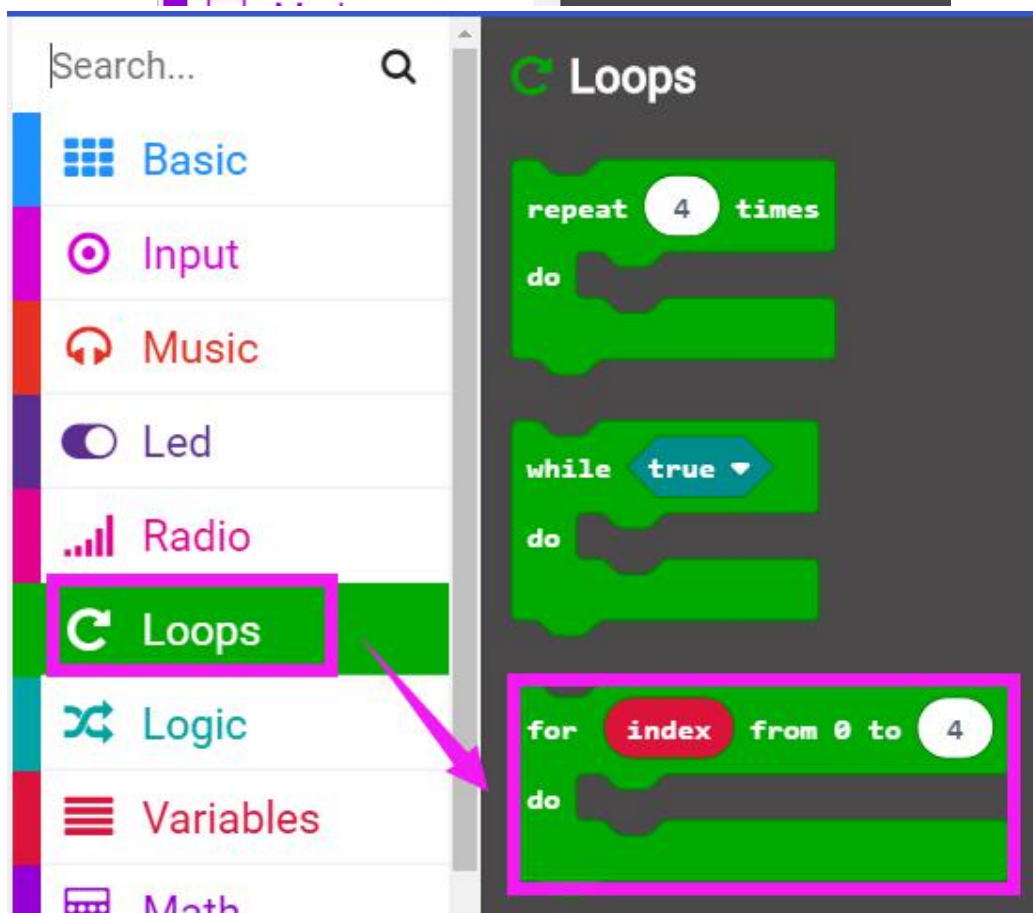
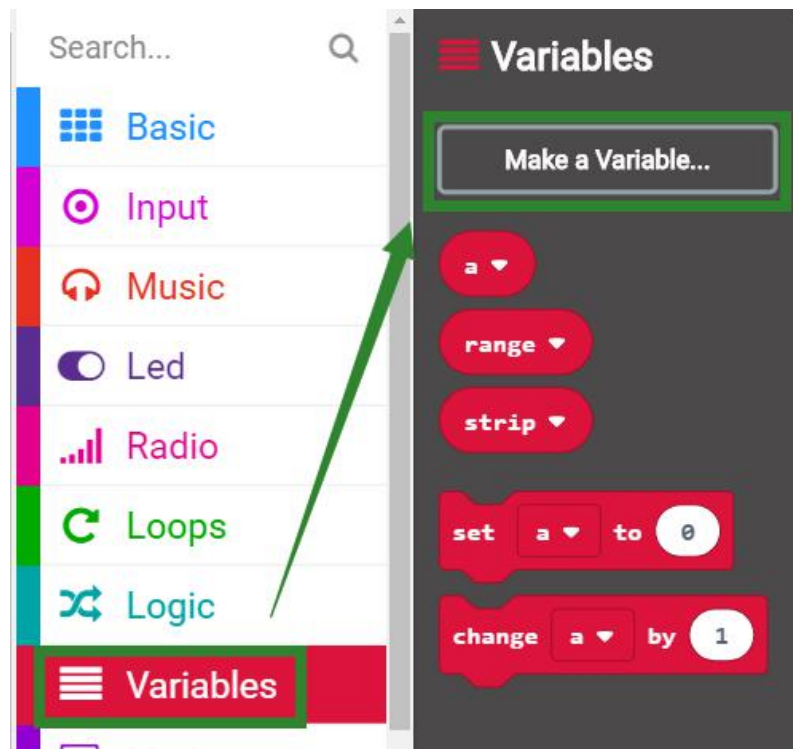
Mode 2 offline programming: We need to open the offline programming software. After the installation is complete, enter the programming interface, click **【New Project】**, add Yahboom package: <https://github.com/lzty634158/SuperBit>, you can program.

3.Looking for blocks

The following is the location of the building blocks required for this programming.

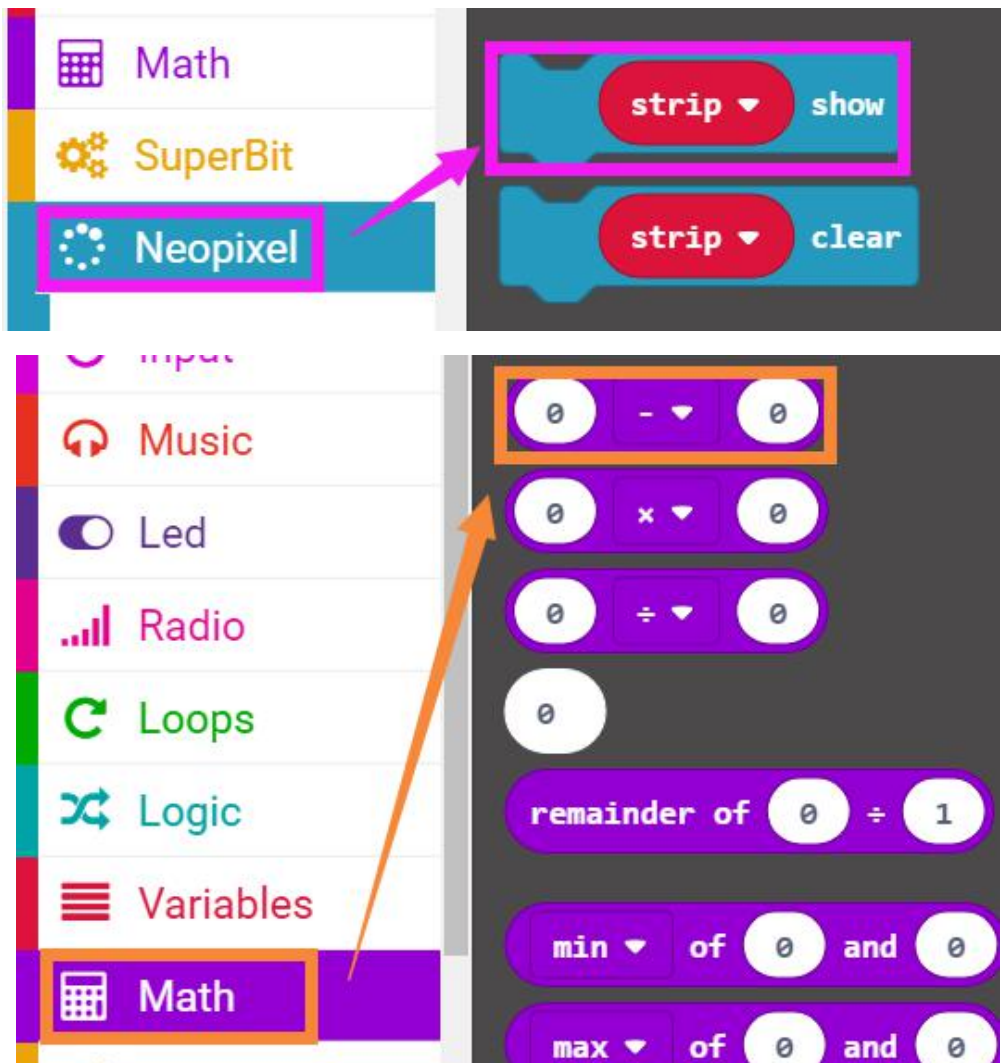
Control of one breathing light separately:



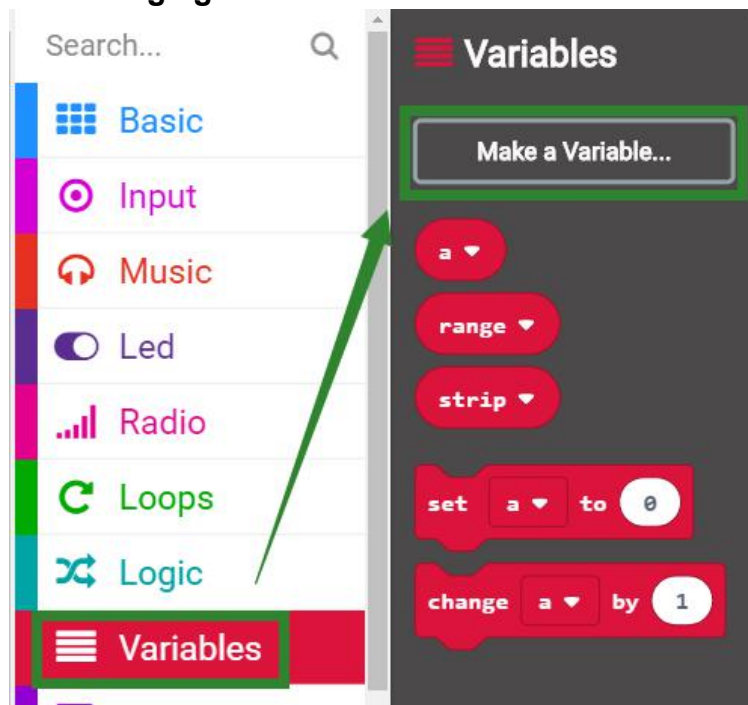


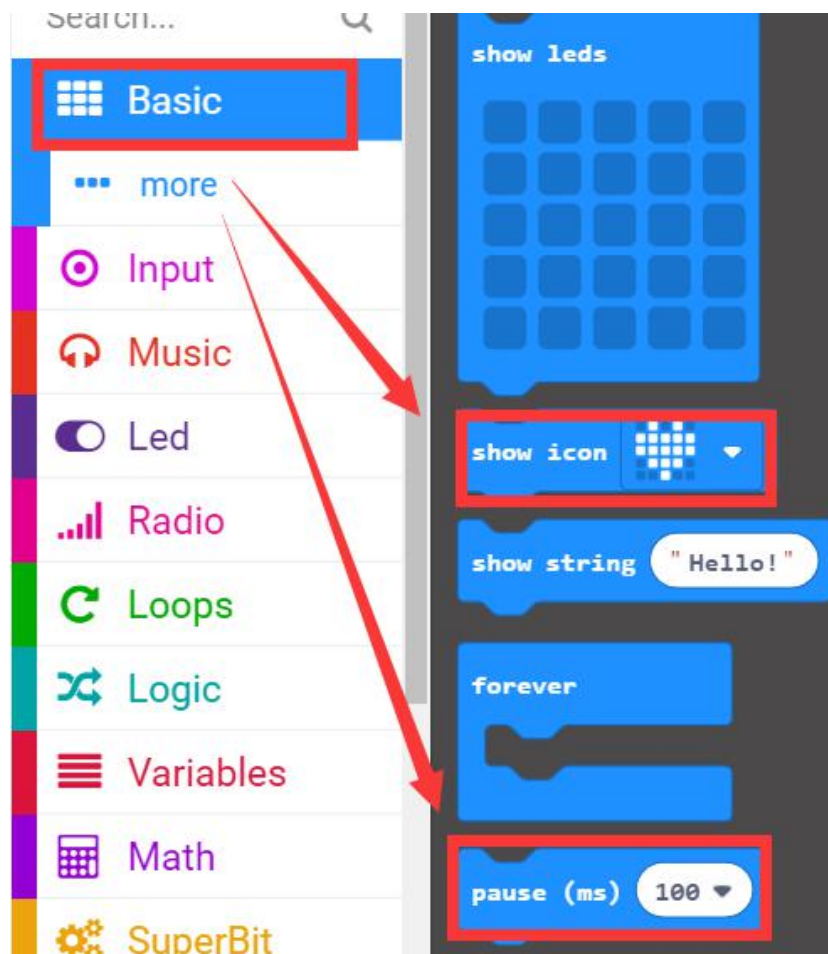
The screenshot shows the SuperBit interface. On the left, a sidebar contains a search bar and a list of categories: Basic, Input, Music, Led, Radio, Loops, Logic, Variables, Math, and SuperBit. The 'SuperBit' category is highlighted with a pink box. A pink arrow points from this box to the 'RGB_Program' block in the main workspace. The 'RGB_Program' block is also highlighted with a pink box. Below it, several other blocks are visible: 'Music' (dadadum), 'Servo(180°)' (num S1, value 0), 'Servo(270°)' (num S1, value 0), 'Servo(360°)' (num S1, pos forward), and 'Motor' (M1, speed(-255~255) 0).

The screenshot shows the SuperBit interface. On the left, a sidebar contains a search bar and a list of categories: Basic, Input, Music, Led, Radio, Loops, Logic, Variables, Math, SuperBit, and Neopixel. The 'Neopixel' category is highlighted with a pink box. A pink arrow points from this box to the 'strip' blocks in the main workspace. The 'strip' blocks are highlighted with pink boxes. The blocks in the workspace are: 'strip' (set pixel color at 0 to red), 'strip' (set pixel white LED at 0 to 0), 'strip' (length), 'strip' (set brightness 255), 'strip' (ease brightness), 'strip' (power (mA)), 'strip' (set matrix width 0 rotation 0 ch), and 'strip' (set matrix color at x 0 y 0 to).



Control of 4 breathing lights:

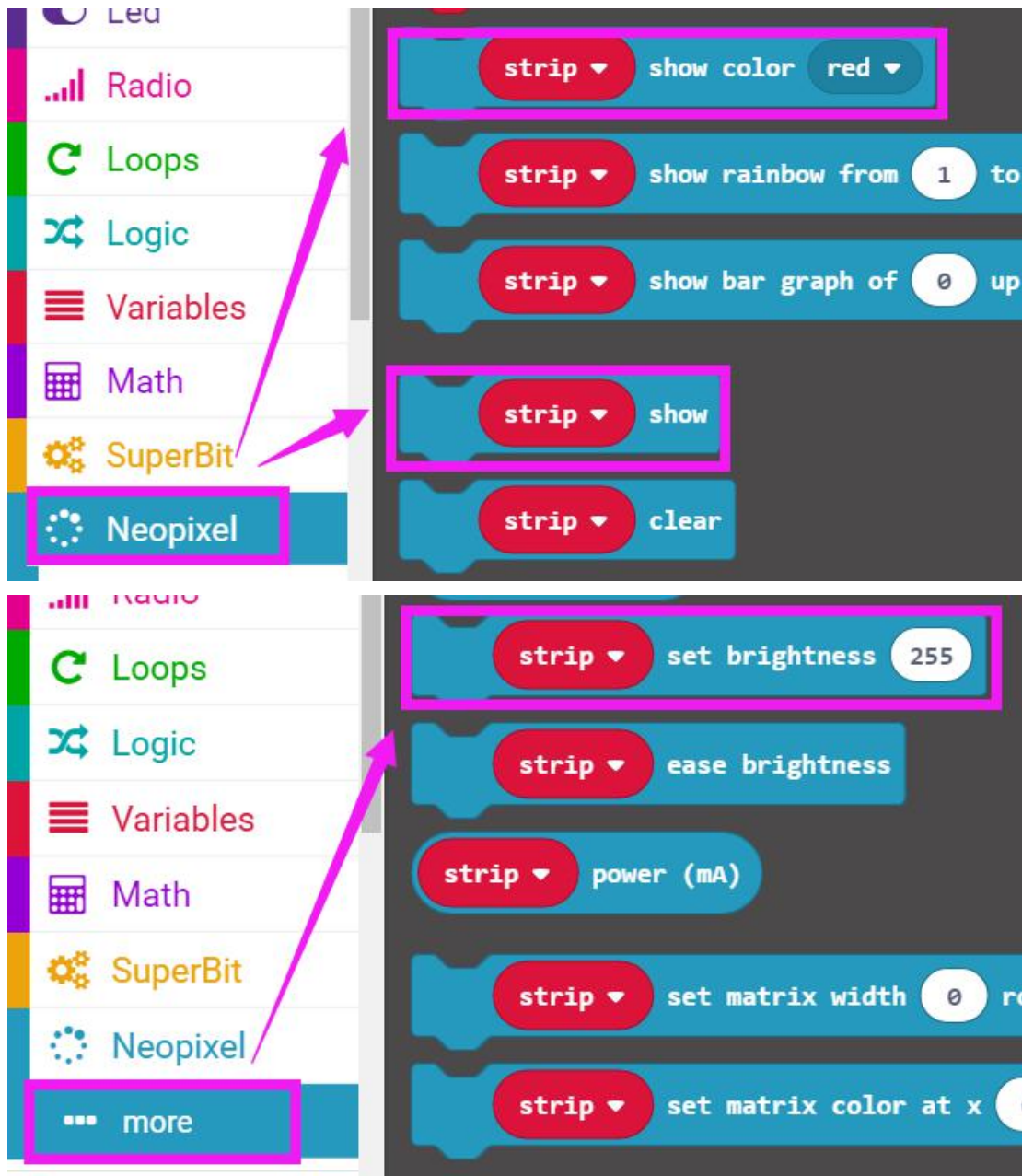




The image displays two screenshots of the SuperBit block-based programming interface.

Top Screenshot: The interface shows a search bar and a category list on the left. The 'SuperBit' category is selected. The 'RGB_Program' block is highlighted with a pink box. A pink arrow points from the 'RGB_Program' block to the 'Music' block. The workspace contains several blocks: 'Music' (dadadum), 'Servo(180°)' (num S1, value 0), 'Servo(270°)' (num S1, value 0), 'Servo(360°)' (num S1, pos forward), and 'Motor' (M1, speed(-255~255) 0).

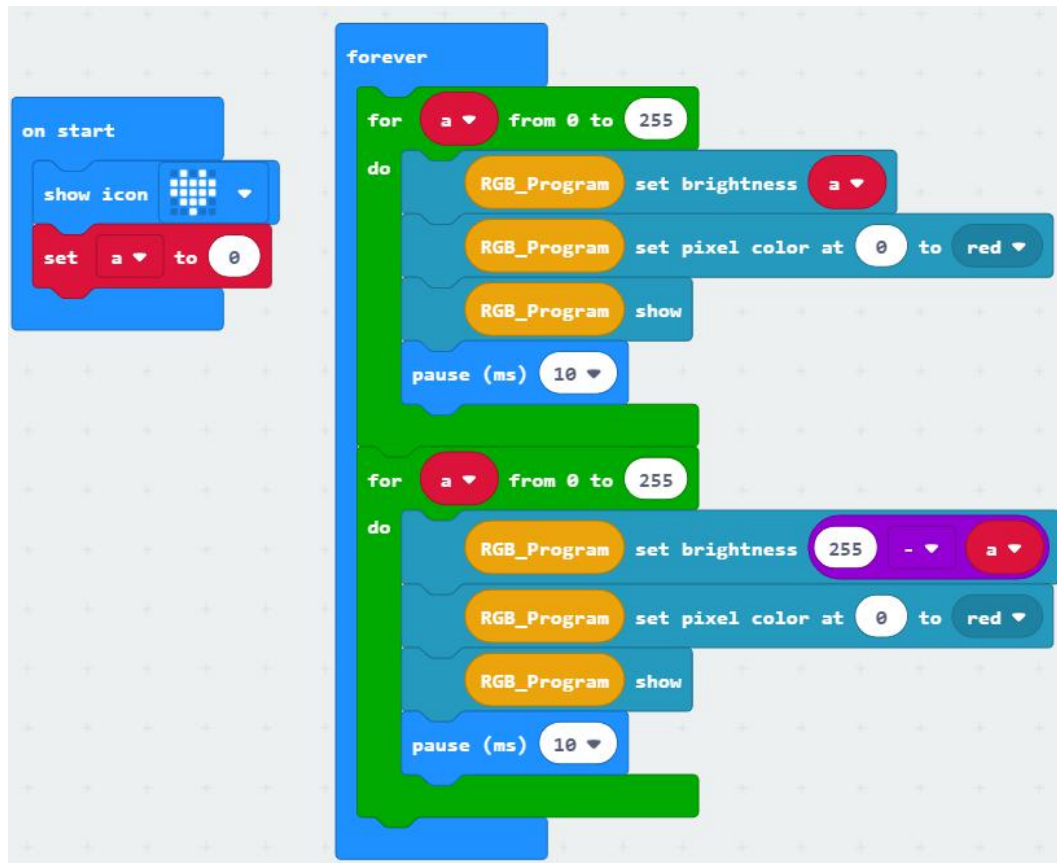
Bottom Screenshot: The interface shows the 'Math' category selected. The 'Math' block is highlighted with an orange box. An orange arrow points from the 'Math' block to the 'remainder of' block. The workspace contains several math blocks: a subtraction block (0 - 0), a multiplication block (0 x 0), a division block (0 ÷ 0), a remainder block (remainder of 0 ÷ 1), a min block (min of 0 and 0), and a max block (max of 0 and 0).



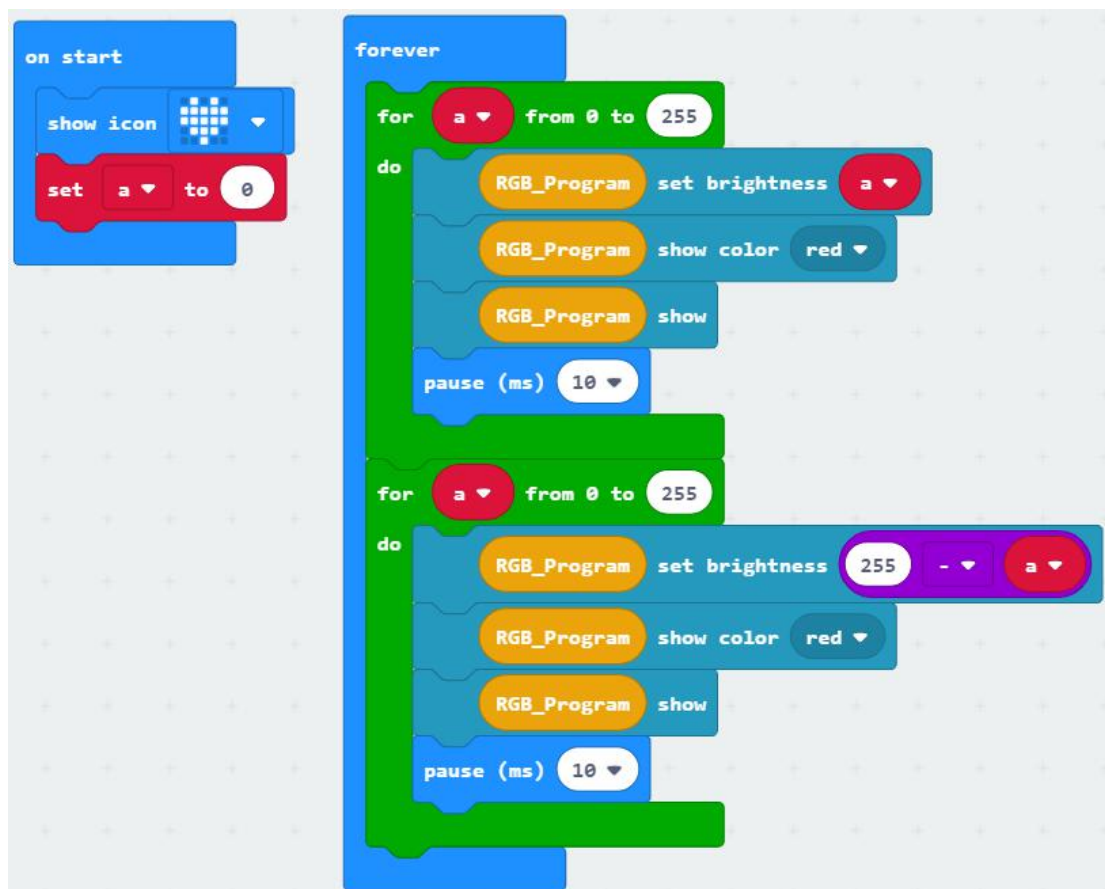
4.Combine building block

The summary program is shown below:

Control of one breathing light separately:



Control of 4 breathing lights:



5. Experimental phenomena

After the program is successfully downloaded, the micro:bit dot matrix will display the heart pattern .

Control of one breathing RGB light separately:

The color of the 0th RGB lamp will become breathing light.

Control of 4 breathing RGB lights:

The color of all RGB lamp will become breathing lights.

If you need to start over, press the reset button on the back of the micro:bit board.