

Microbit handle control

Microbit handle control

1. Learning objectives
2. Building blocks
3. Motor wiring
4. Programming
 - 4.1 Add expansion package
 - 4.2 Blocks used
 - 4.3 Add new variables
 - 4.4 Combined blocks
5. Experimental phenomenon

1. Learning objectives

In this course, we mainly learn how to use MakeCode graphical programming to realize the control of the security automatic door by the microbit handle.

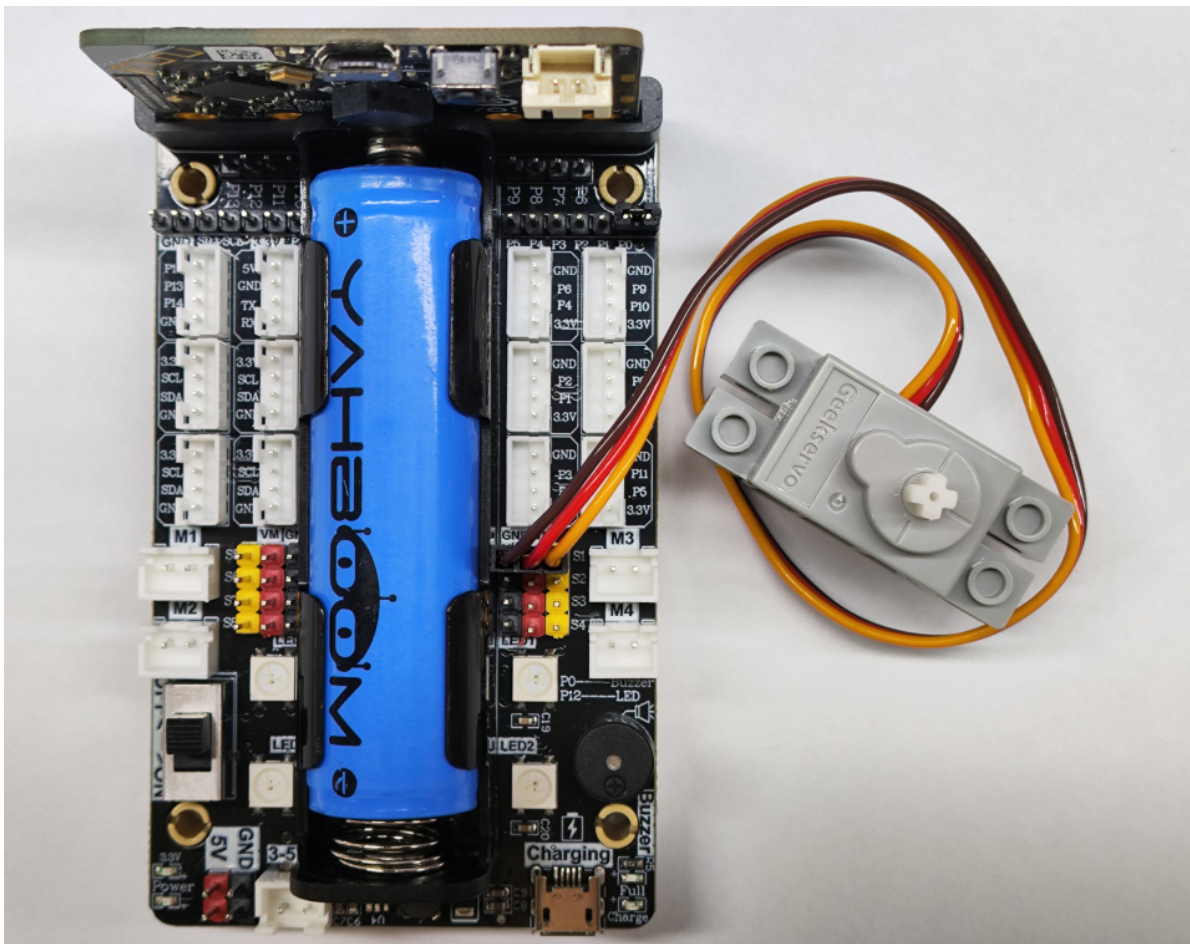
2. Building blocks

For the building blocks steps, please refer to the installation drawings of **[Assembly course]-[Lifting platform]** or the building blocks installation album in the materials.

3. Motor wiring

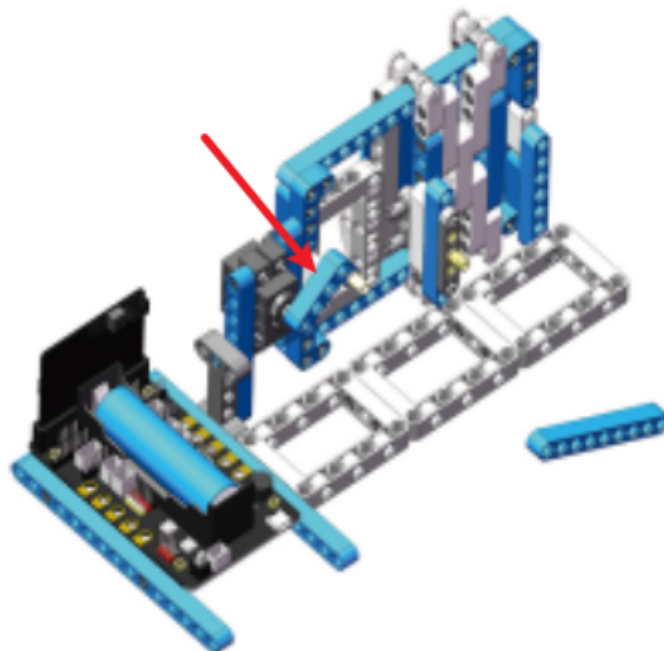
Insert the building blocks servo wiring into the Super:bit expansion board S1 interface, and the servo orange wiring into the yellow pin of S1.

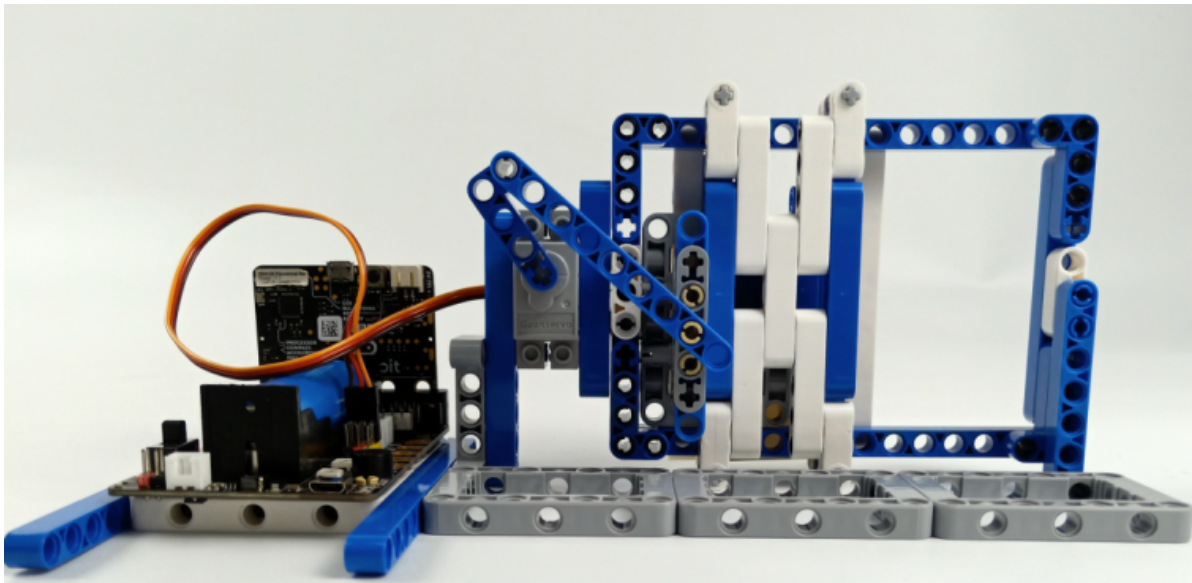
As shown in the figure below:



! Notes:

When taking a course related to the building block servo for the first time, we need to remove the blue building block installed on the servo and upload the program of this course to the micro:bit; then turn on the power switch of the Super:bit expansion board and wait for the building block servo to turn to the initial position; then, we can turn off the power, adjust the security automatic door to the open state, as shown in the figure below, and then install the blue building block. (If you have used the security automatic door and servo related programs before, you can skip this step)





4. Programming

Method 1 Online programming:

First connect micro:bit to the computer via a USB data cable, and a U disk will pop up on the computer. Click the URL in the U disk: <https://makecode.microbit.org/> to enter the programming interface. Then, add the Yahboom software package to start programming.

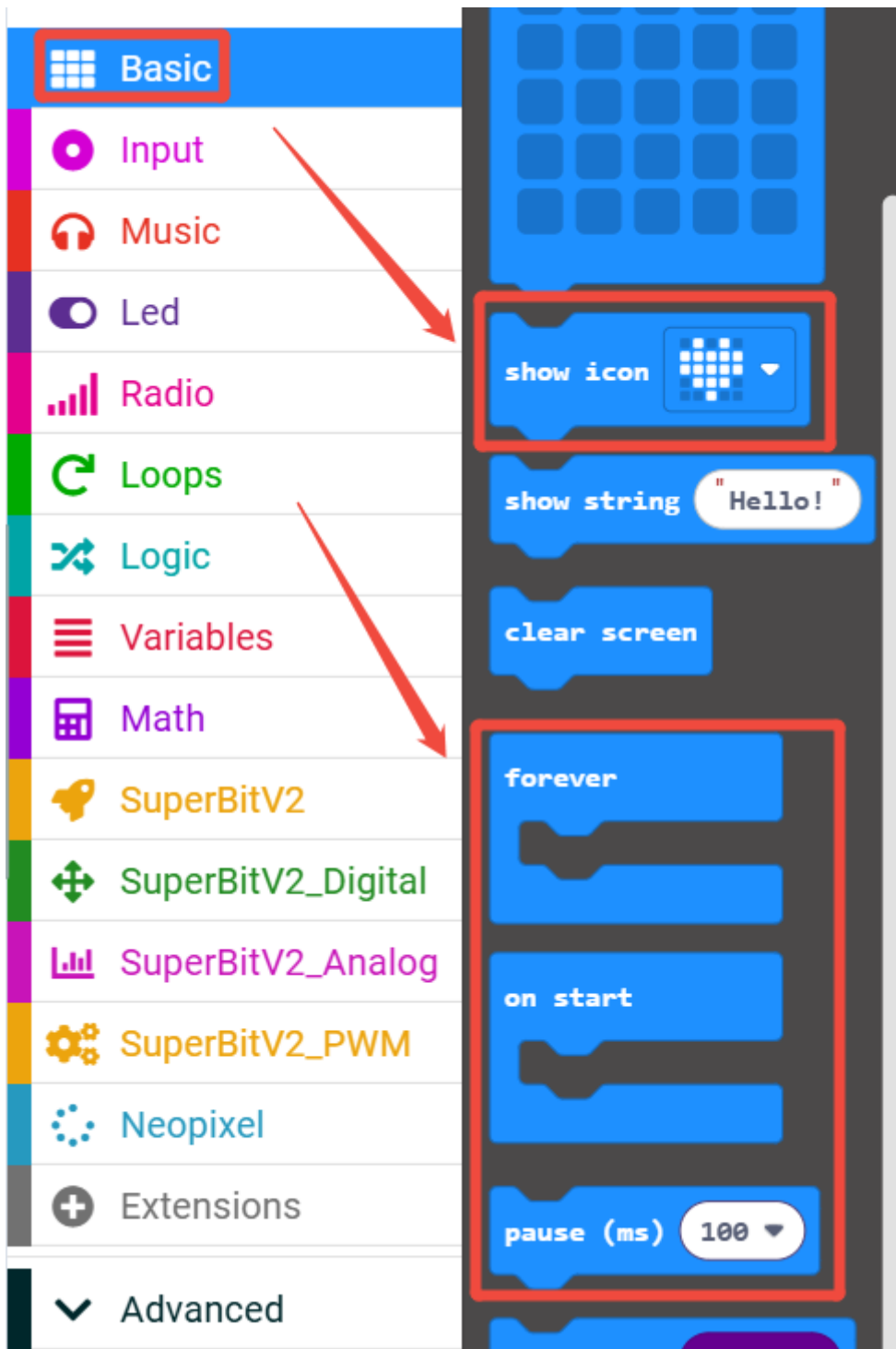
Method 2 Offline programming:

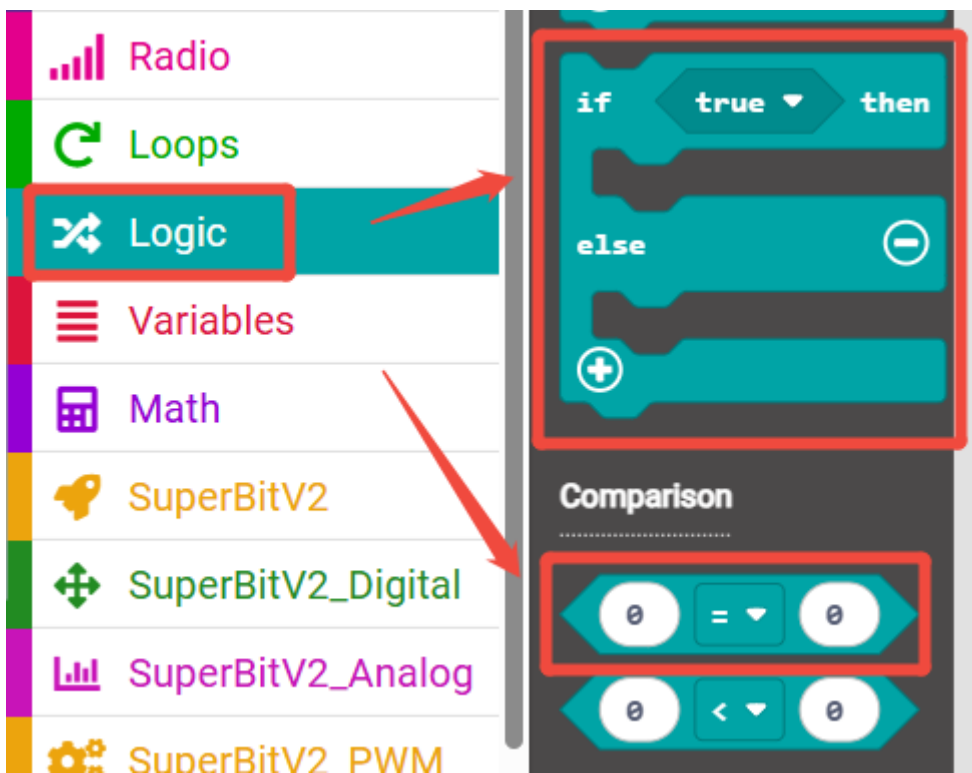
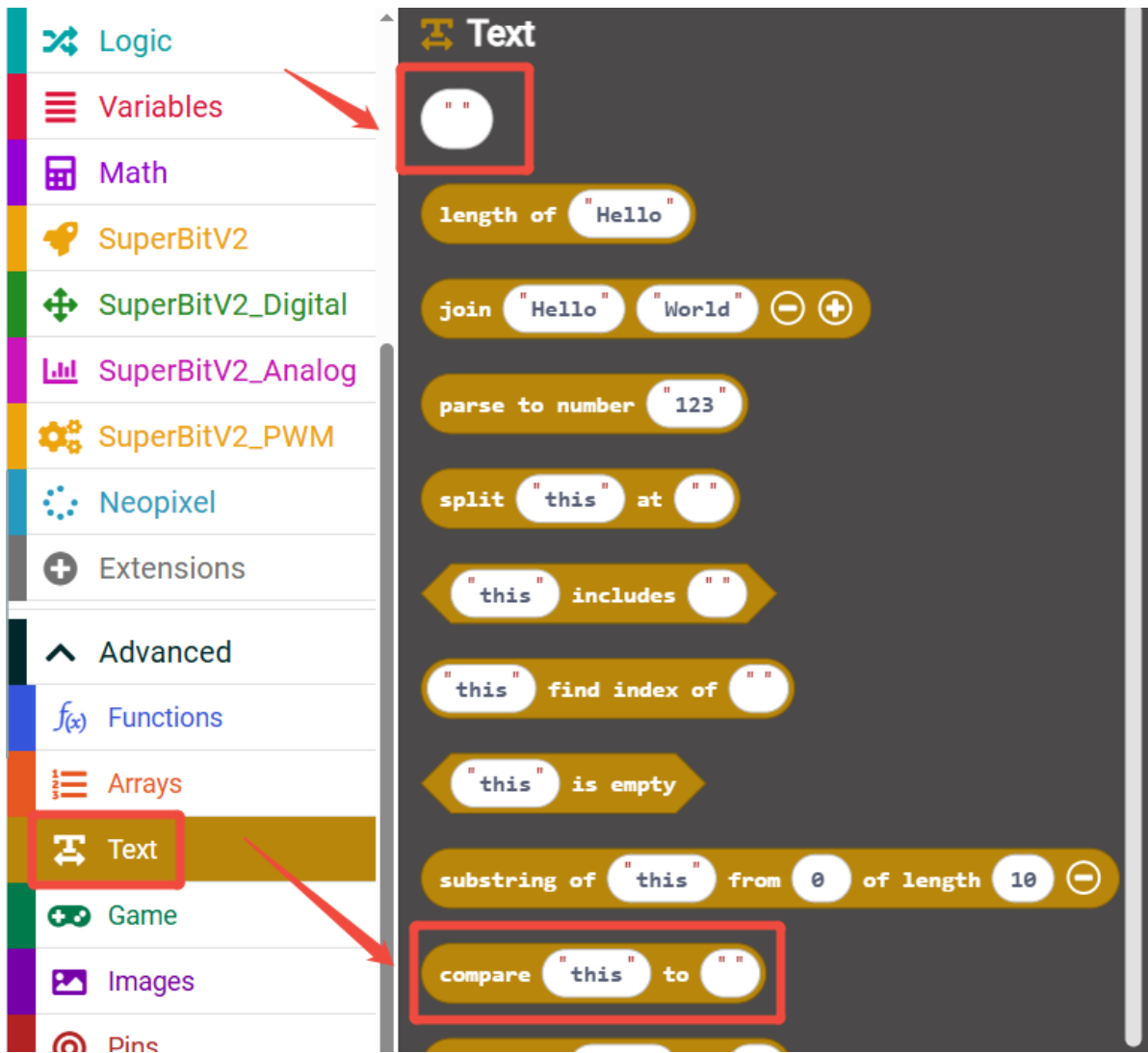
Open the offline programming software MakeCode and enter the programming interface. Click [New] and add the Yahboom software package to start programming.

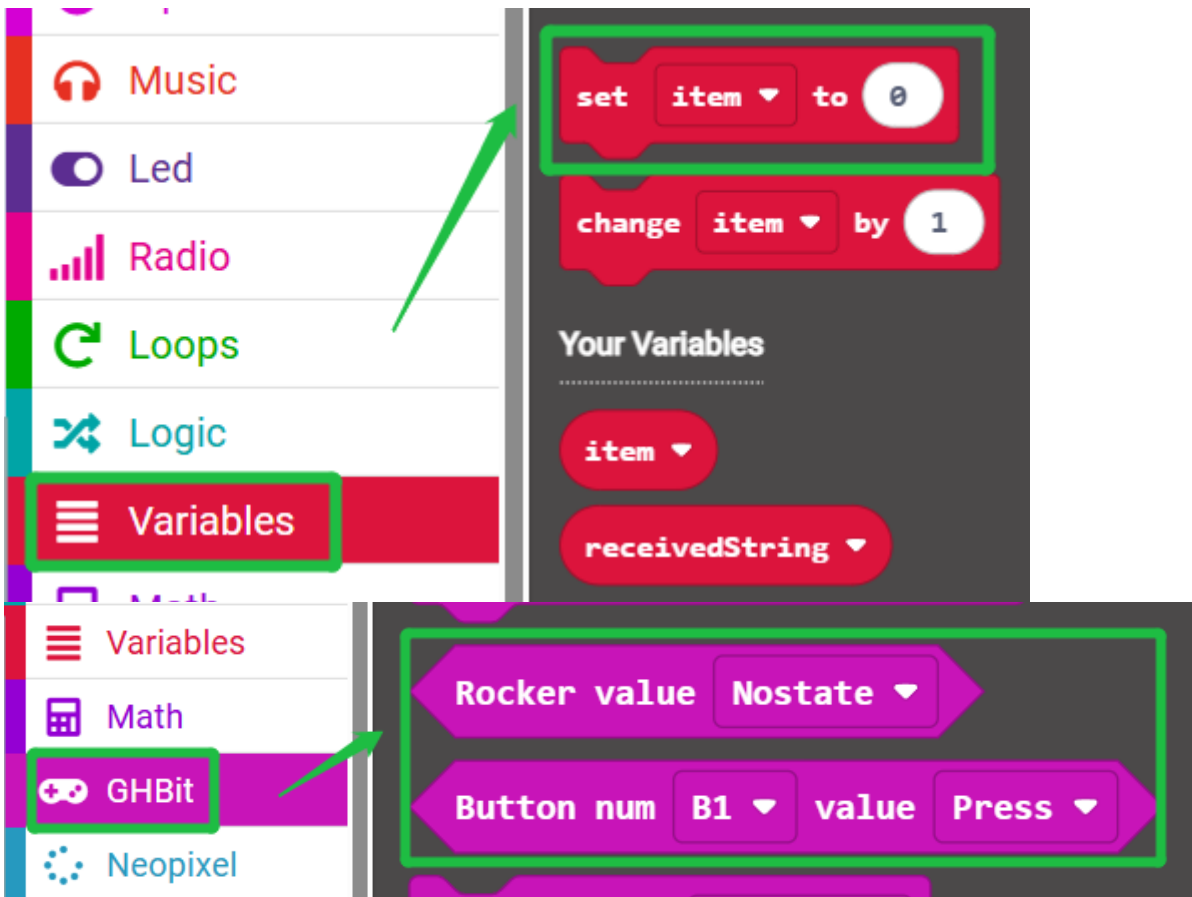
superbit kit expansion package: <https://github.com/YahboomTechnology/SuperBitLibV2>

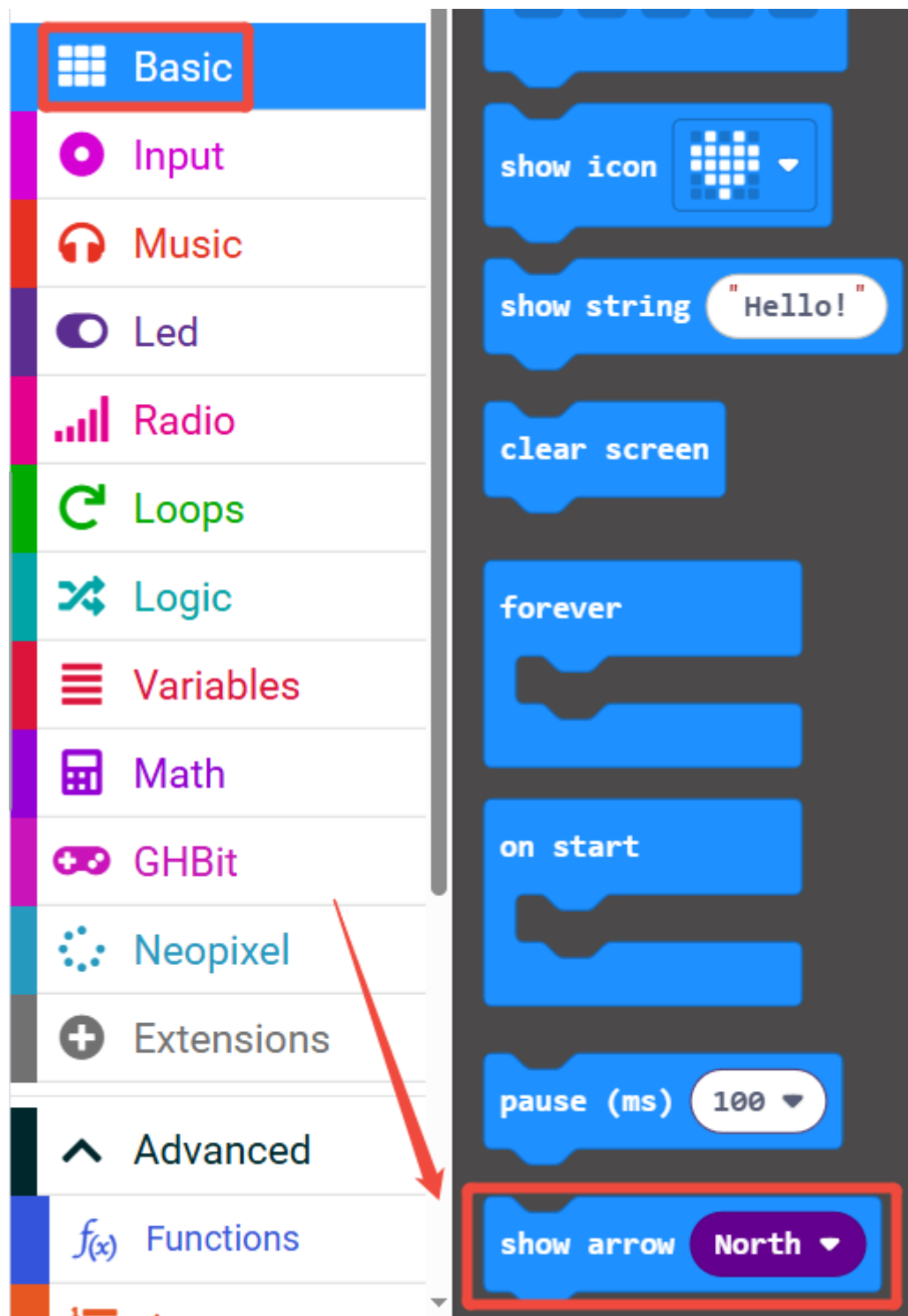
handle expansion package: <https://github.com/YahboomTechnology/GHBitLib>

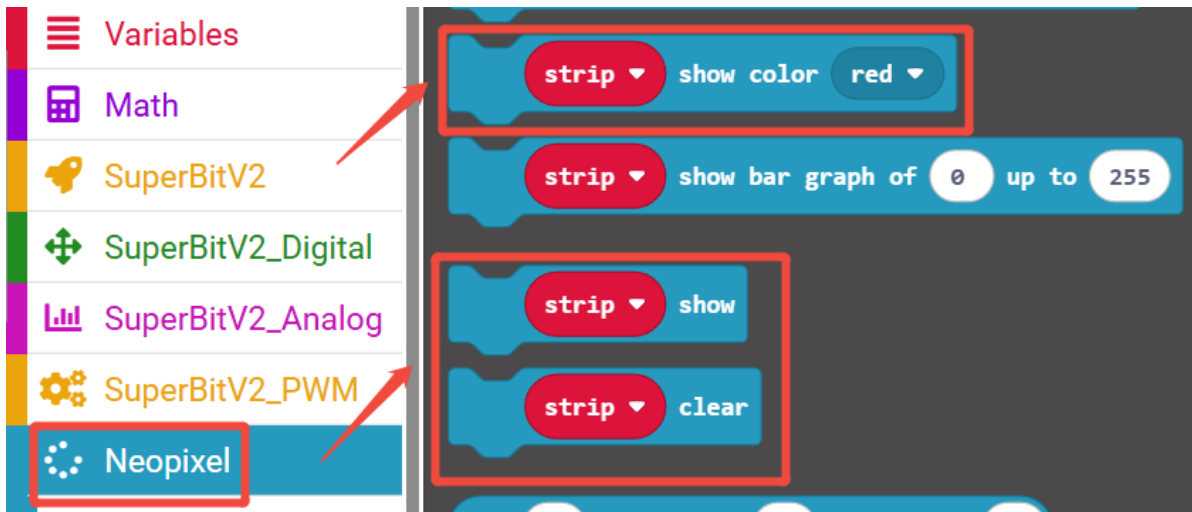
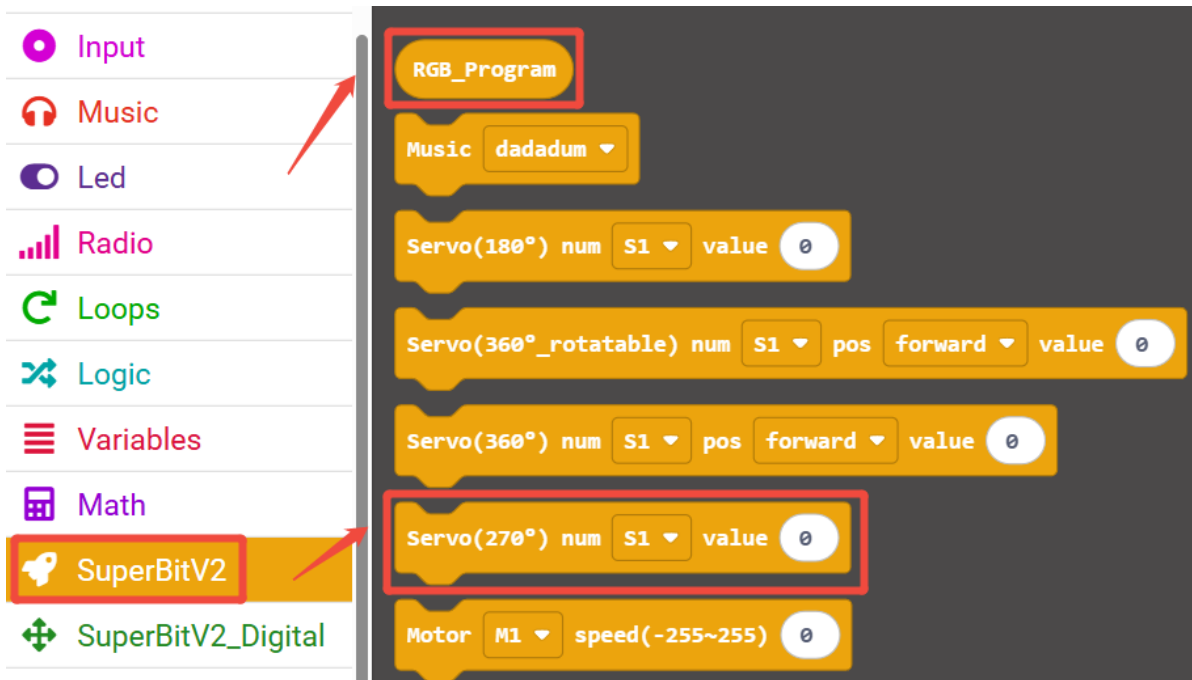
4.1 Add expansion package

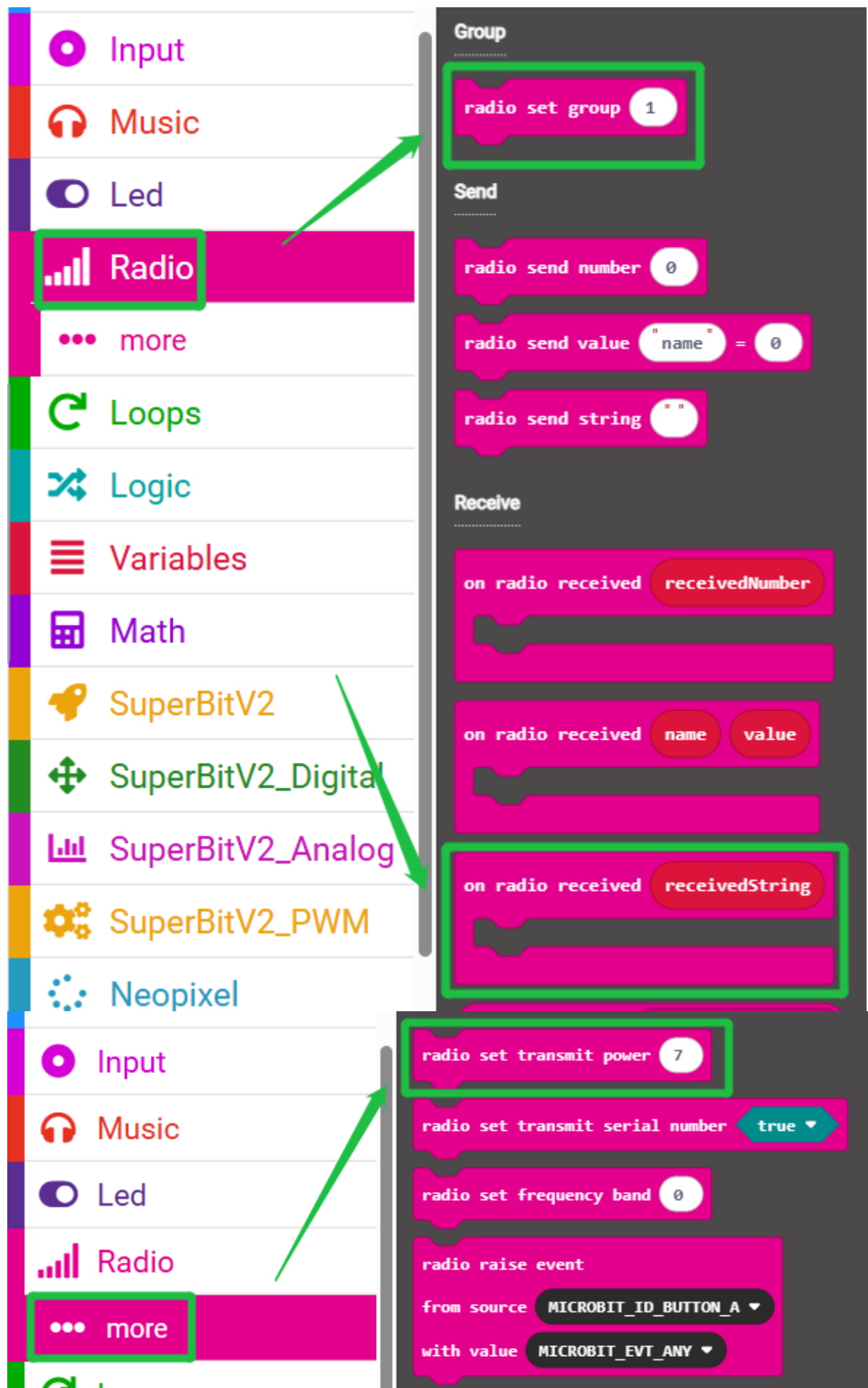












The image displays the Scratch Microbit interface, showing a library of blocks on the left and a workspace on the right. The 'Radio' block category is highlighted in pink. In the workspace, a 'Group' block contains a 'radio set group' block (value 1). Below it, a 'Send' section includes 'radio send number' (0), 'radio send value' (name = 0), and 'radio send string' ("). A 'Receive' section follows, with 'on radio received' events for 'receivedNumber', 'name' and 'value', and 'receivedString'. At the bottom, another 'radio set transmit power' block (7) is shown, followed by 'radio set transmit serial number' (true), 'radio set frequency band' (0), and a 'radio raise event' block (from source: MICROBIT_ID_BUTTON_A, with value: MICROBIT_EVT_ANY). Green arrows point from the 'Radio' block in the library to the 'radio set group' and 'radio set transmit power' blocks in the workspace.

Library:

- Input
- Music
- Led
- Radio**
- more
- Loops
- Logic
- Variables
- Math
- SuperBitV2
- SuperBitV2_Digital
- SuperBitV2_Analog
- SuperBitV2_PWM
- Neopixel

Workspace:

Group

- radio set group 1

Send

- radio send number 0
- radio send value "name" = 0
- radio send string "

Receive

- on radio received receivedNumber
- on radio received name value
- on radio received receivedString

radio set transmit power 7

radio set transmit serial number true

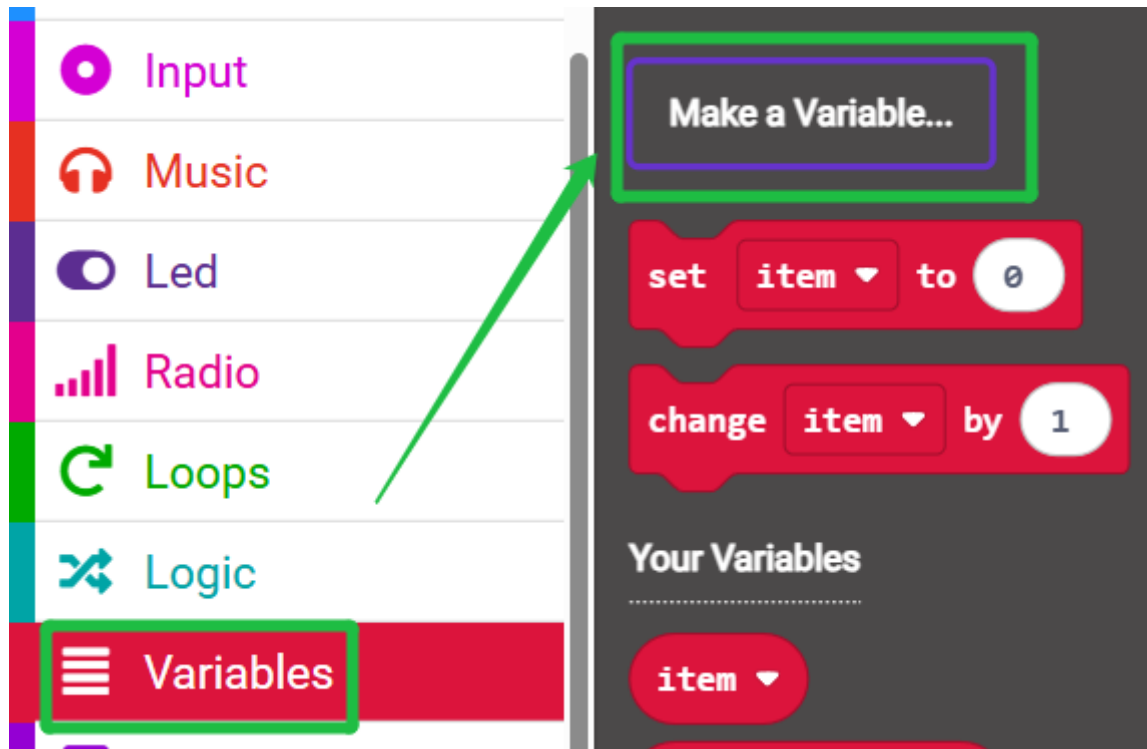
radio set frequency band 0

radio raise event

- from source MICROBIT_ID_BUTTON_A
- with value MICROBIT_EVT_ANY

4.3 Add new variables

① Find the [Variable] option in the building block bar ---- [Set variable].

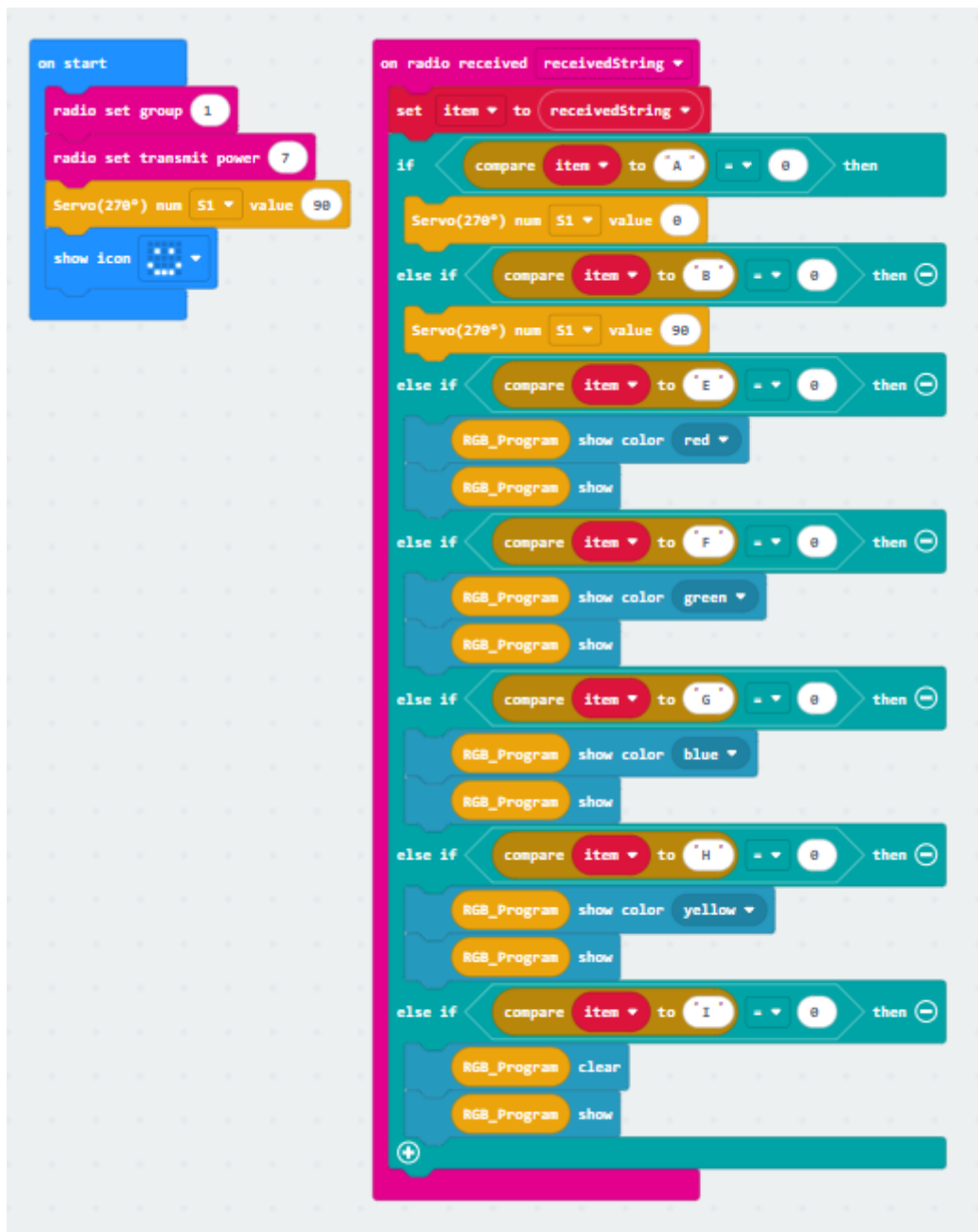


② Enter the variable name to complete the new variable.

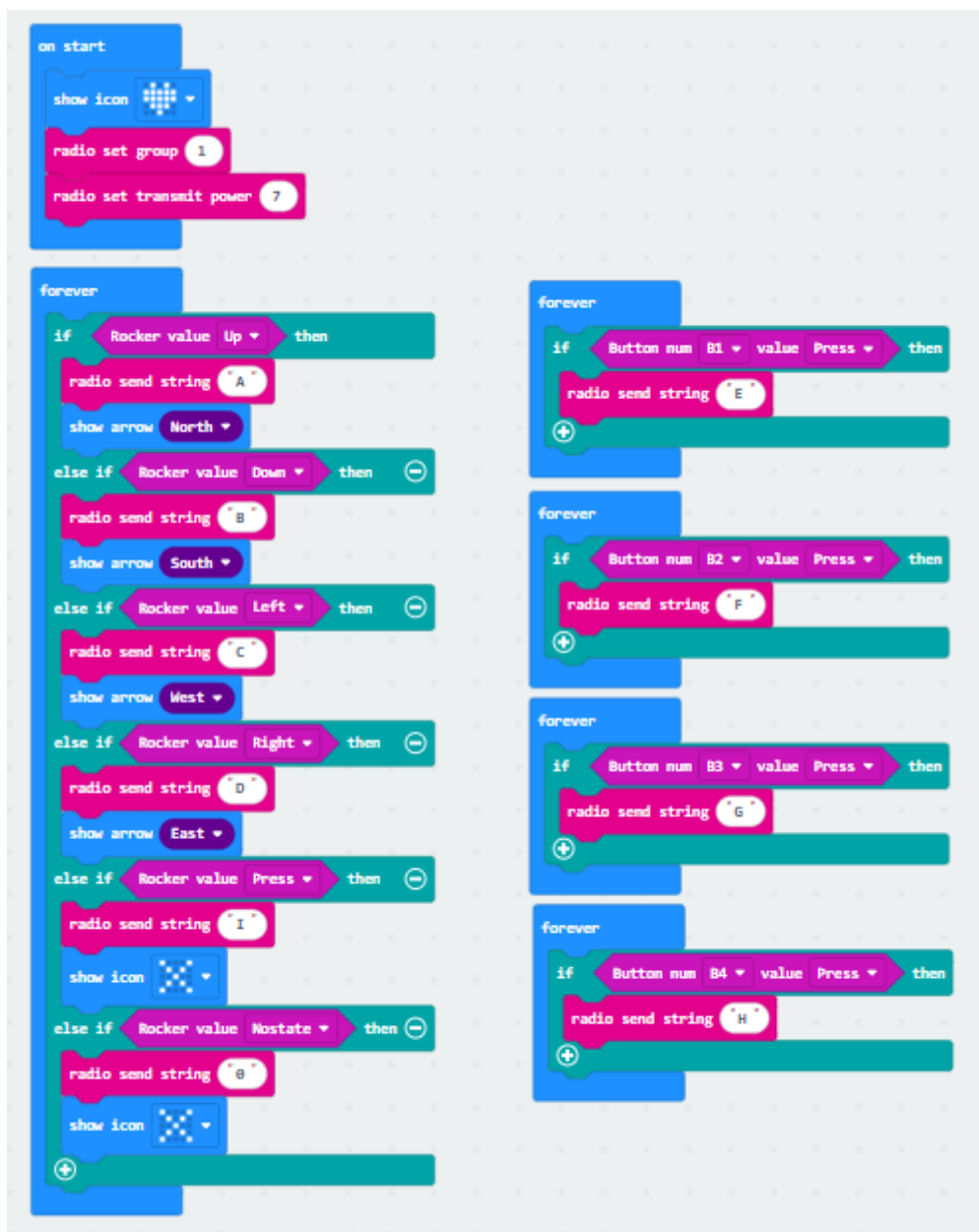
The image shows a 'New variable name' dialog box. It has a title bar with a close button (X). Below the title bar, there is a text input field containing the word 'value', which is highlighted with a red box. At the bottom right of the dialog box, there is a green 'Ok' button with a checkmark, also highlighted with a red box.

4.4 Combined blocks

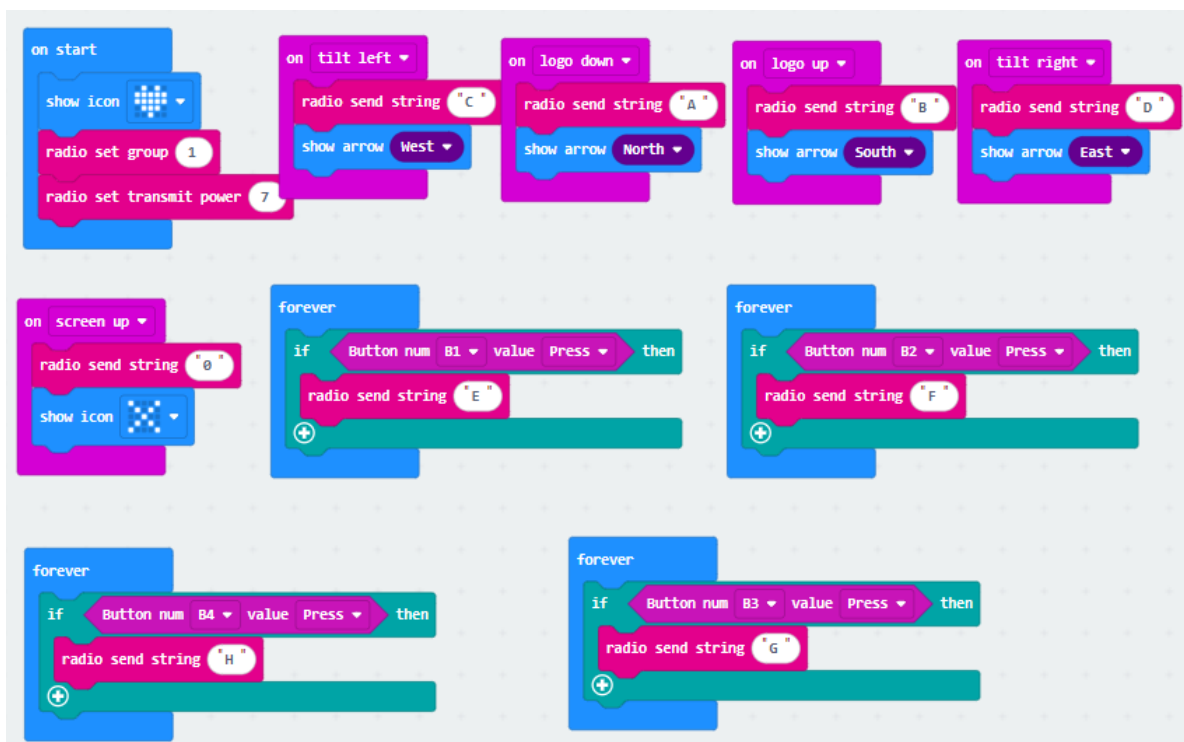
The microbit summary program on the **security automatic door** is shown in the figure below.



The **handle joystick control** program is as follows.



The **handle gravity control** program is as follows.



You can also directly open the **microbit-handle-control-auto-door-code.hex**, **microbit-Handle-rocker-control.hex**, **microbit-Handle-gravity-control.hex** files provided in this experiment and drag them into the browser that opens the URL, and the program diagram of this project source code will be automatically opened.

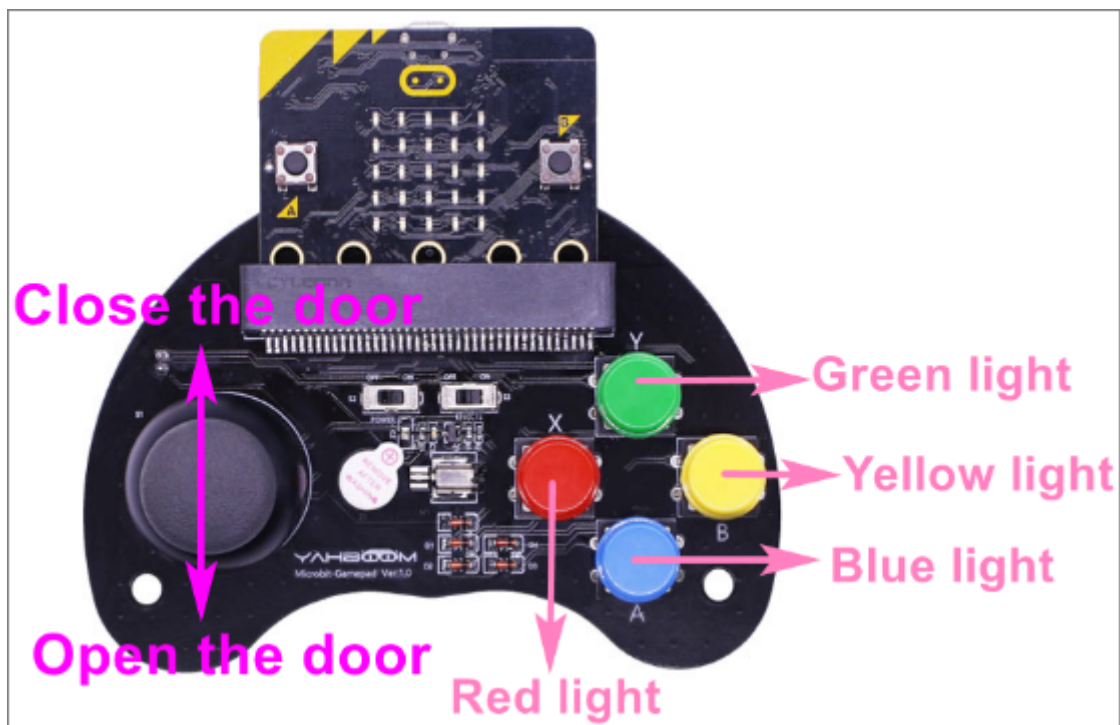
5. Experimental phenomenon

We need to download the security automatic door program to the micro:bit mainboard of the security automatic door, turn on the power switch of the security automatic door, and we can see a smiley face pattern displayed on the micro:bit dot matrix;

Download the handle remote control program to the micro:bit mainboard of the handle, turn on the power switch of the handle, and we can see that the micro:bit dot matrix will initialize to display a heart pattern, and then it will display an "X" pattern, indicating that the handle is in the default state and no data is sent.

The two will automatically complete the pairing, and then we can start to remotely control the security automatic door.

The functions of the handle are as follows.



!Note: When the handle is controlled by the joystick, pressing the joystick can turn off the RGB light. This function does not exist when the handle is controlled by gravity.