

# Microbit handle control

---

## Microbit handle control

1. Learning objectives
2. Building blocks
3. Motor wiring
4. Programming
  - 4.1 Add expansion package
  - 4.2 Building 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 microbit handle control of the Dragon Knight.

## 2. Building blocks

---

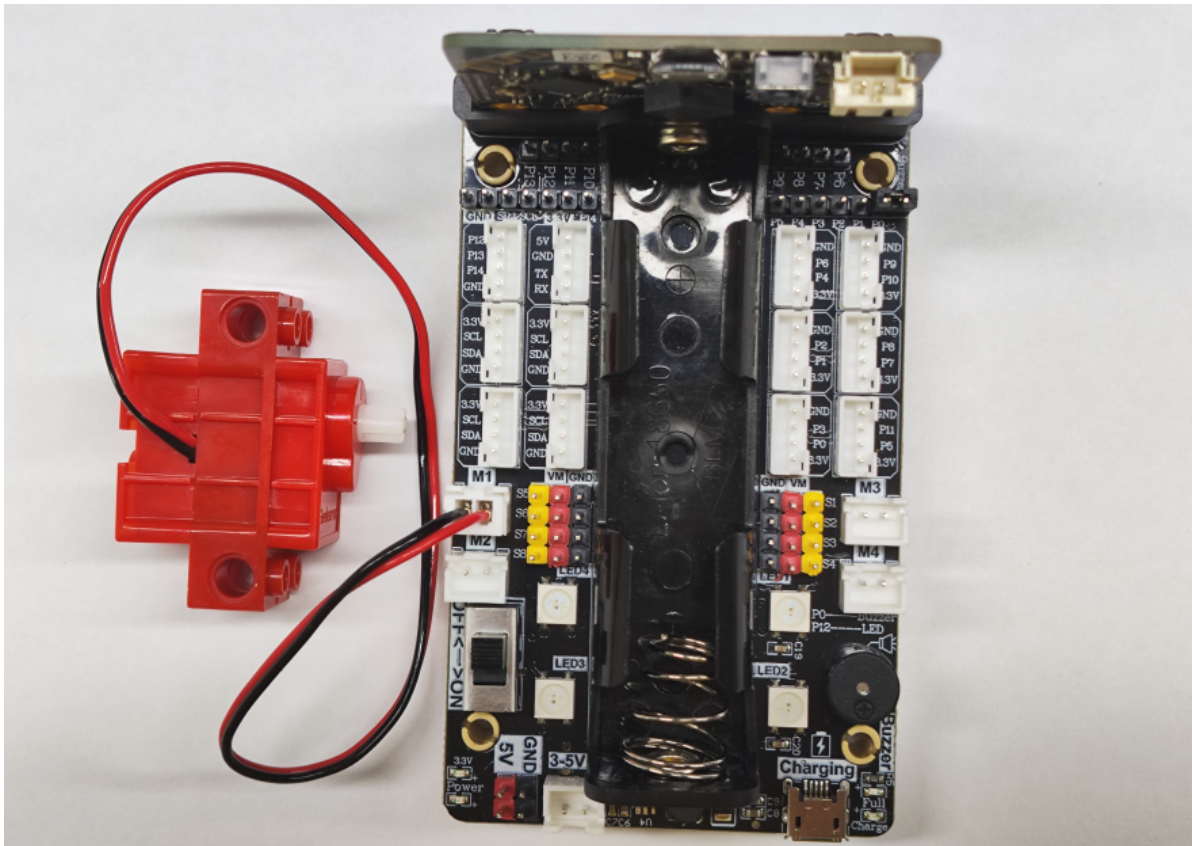
For the building block steps, please refer to the installation drawings of **[Assembly Course]-- [Dragon knight]** in the materials or the building block installation album.

## 3. Motor wiring

---

Insert the motor wiring on the left side of the car into the M1 interface of the Super:bit expansion board, and the black line is close to the battery side;

As shown below:



## 4. Programming

---

### Method 1 Online programming:

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

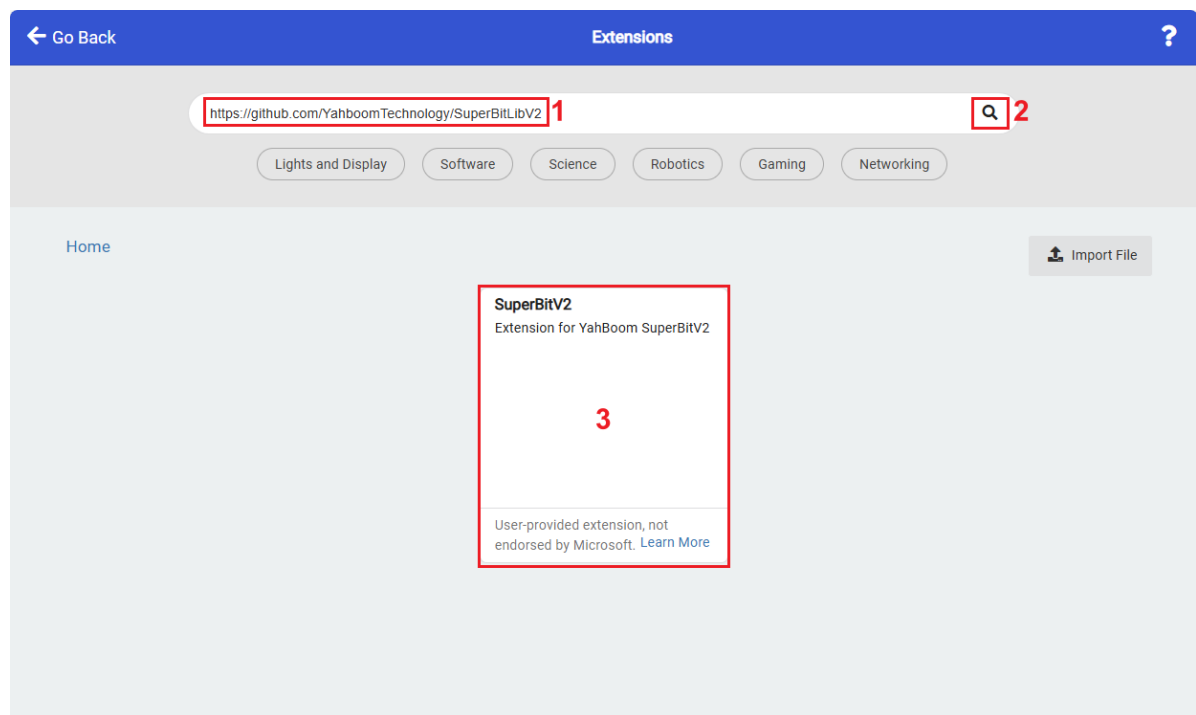
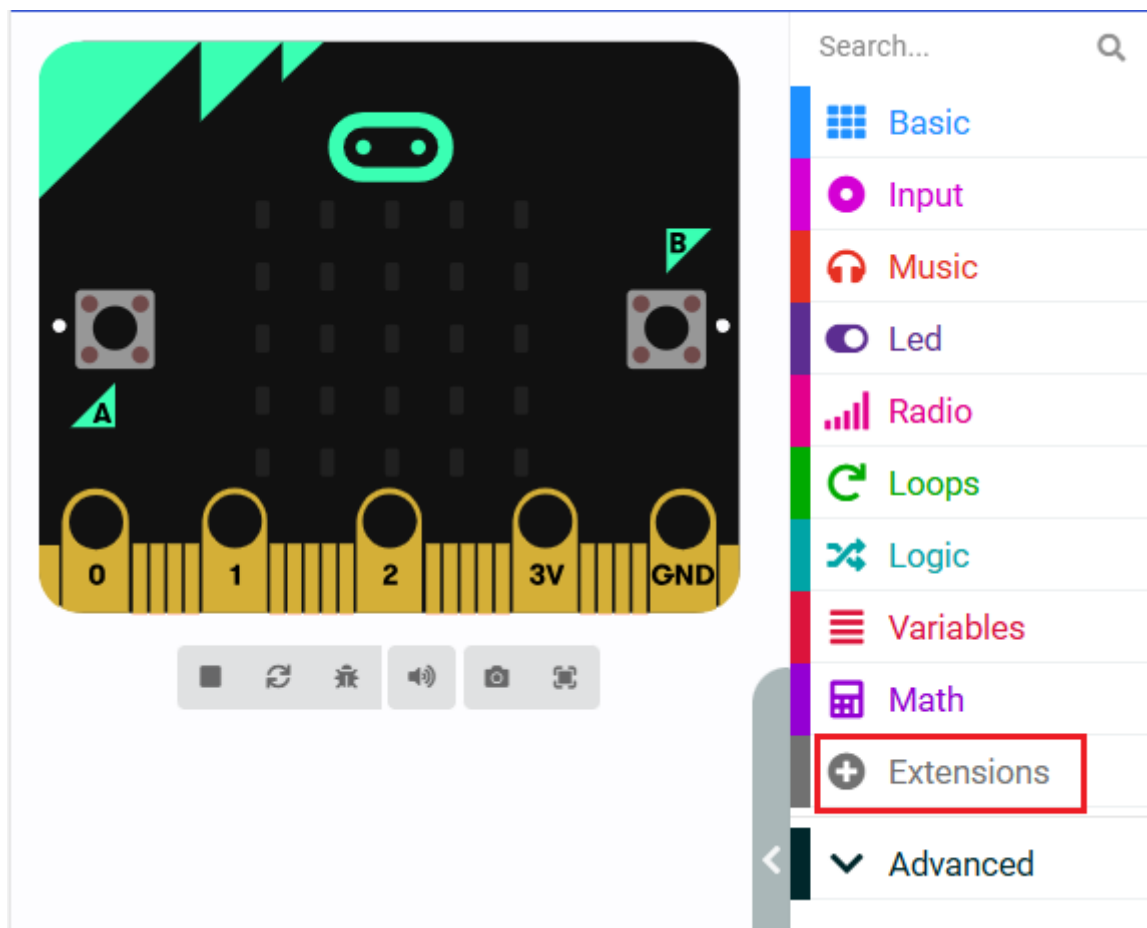
### Method 2 Offline Programming:

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>

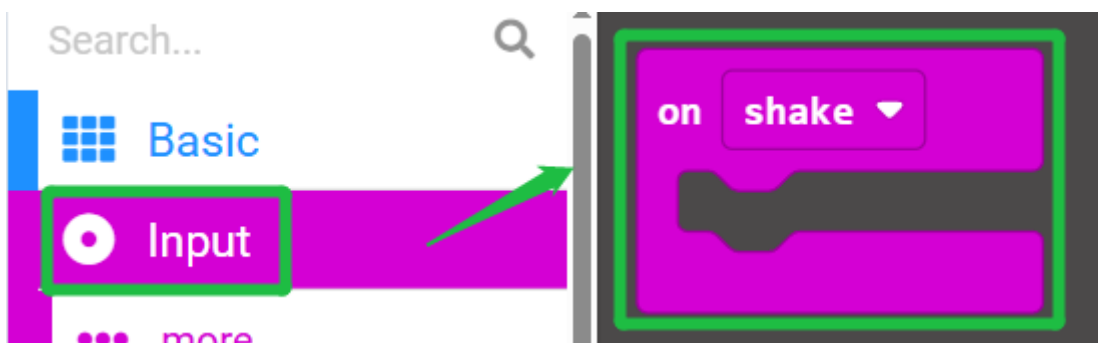
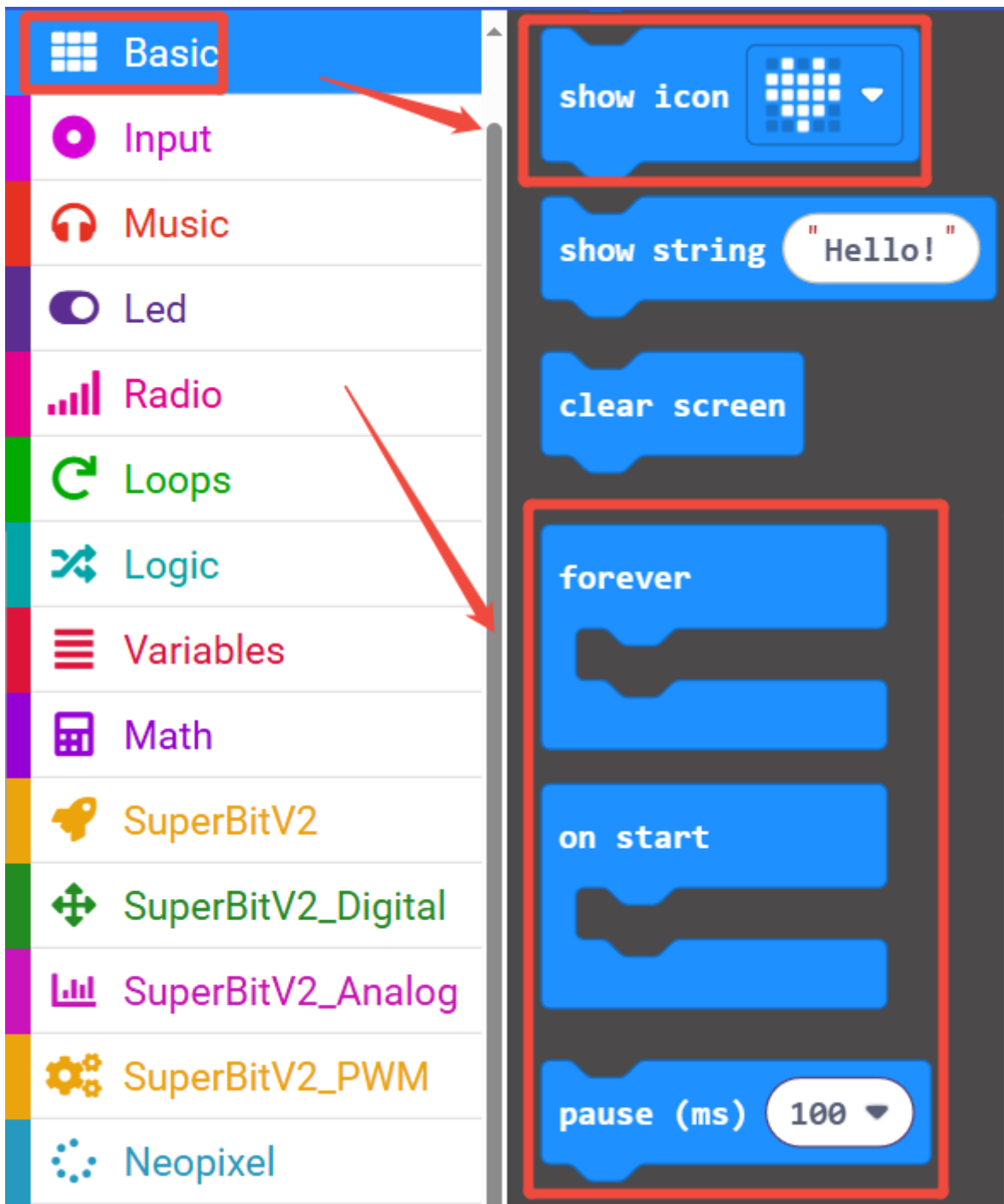
**Handle expansion package:** <https://github.com/YahboomTechnology/GHBitLib>

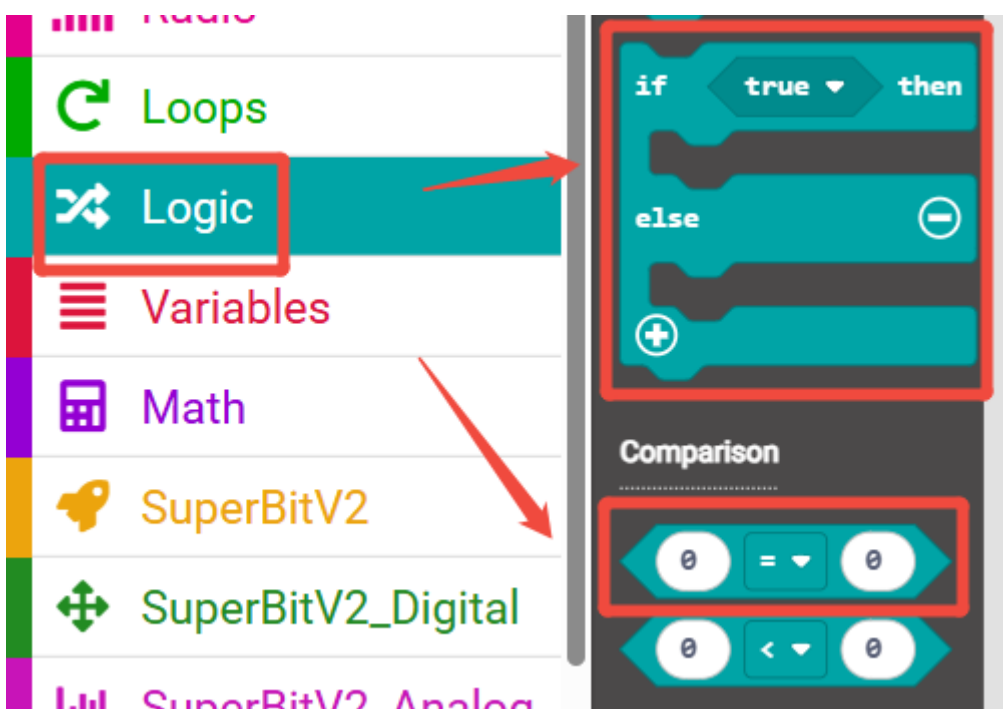
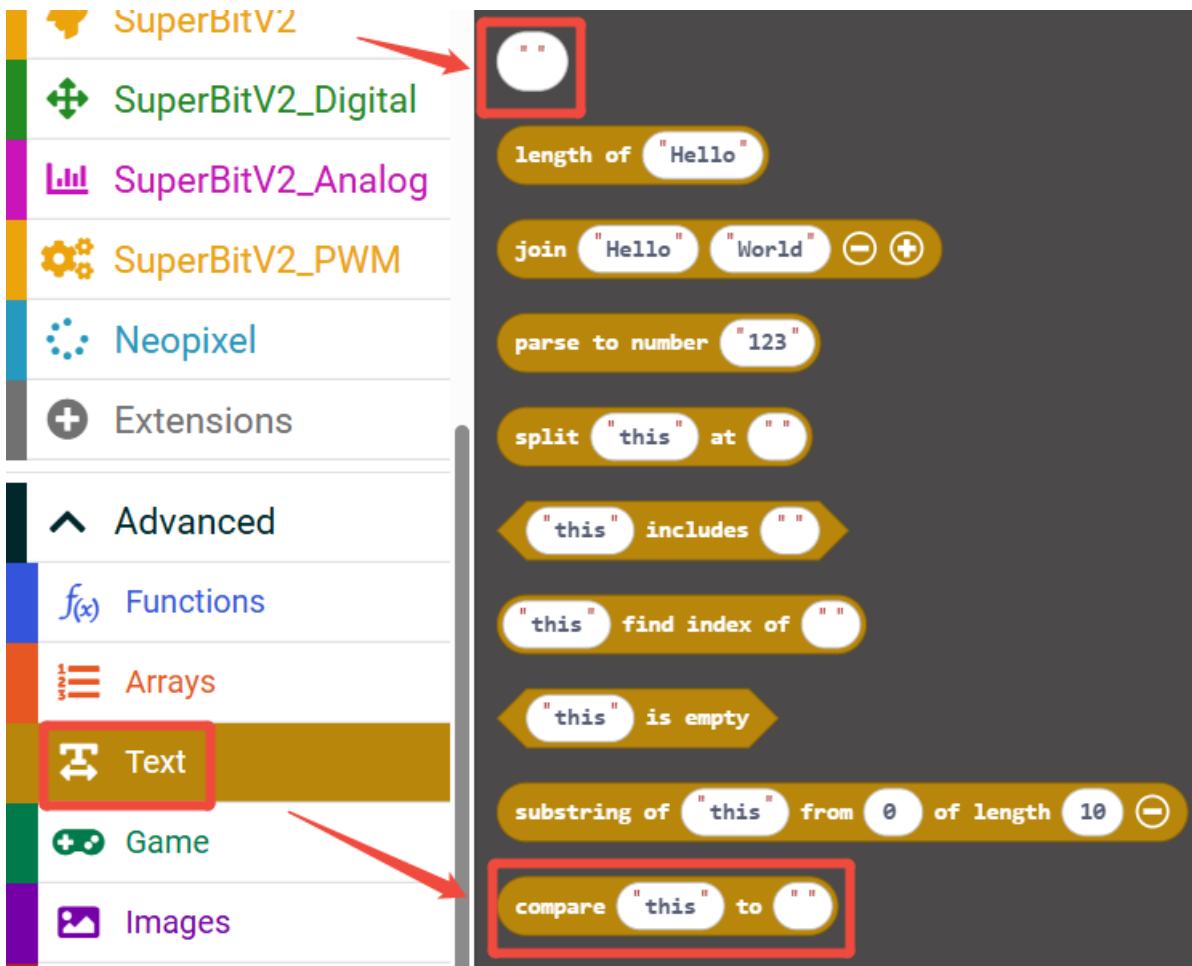
### 4.1 Add expansion package



## 4.2 Building blocks used

The location of the building blocks required for this programming is shown in the figure below.





The image displays the Scratch code editor interface. On the left, the 'Variables' category is selected in the palette, and the 'Variables' block is highlighted. A green arrow points from this block to the 'set flag1 to 0' block in the script area. Another green arrow points from the 'Variables' category to the 'Rocker value' block in the logic area.

**Script Area:**

- set flag1 ▼ to 0
- change flag1 ▼ by 1

**Your Variables**

- flag1 ▼

**Logic Area:**

- Rocker value Nostate ▼
- Button num B1 ▼ value Press ▼
- Music\_Handle dadadum ▼
- Servo\_Handle num S1 ▼ value 0

Search...

**Basic**

Input

Music

Led

Radio

Loops

Logic

Variables

Math

SuperBitV2

SuperBitV2\_Digital

SuperBitV2\_Analog

SuperBitV2\_PWM

Neopixel

Extensions

show icon

show string "Hello!"

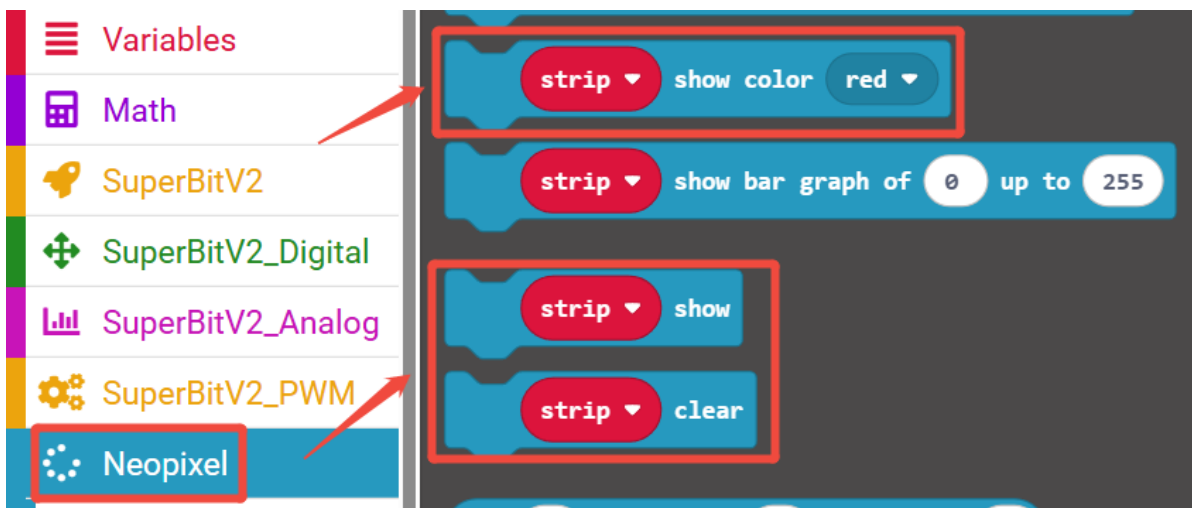
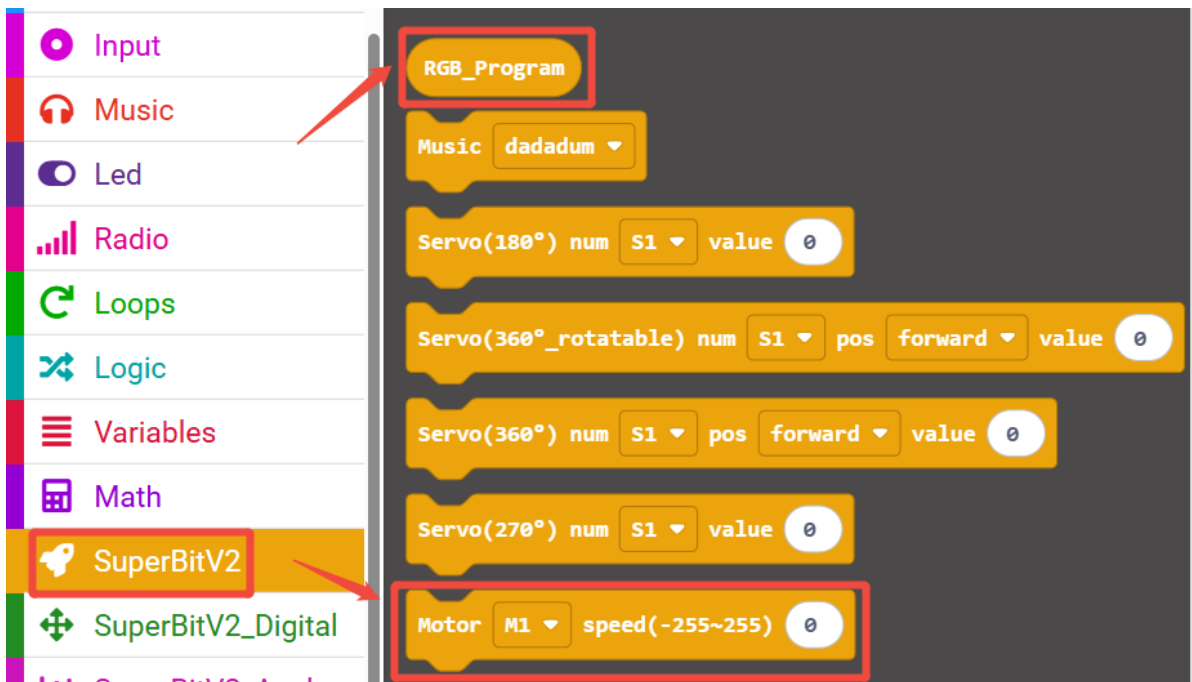
clear screen

forever

on start

pause (ms) 100

show arrow North





The image shows the MIT Scratch Radio extension interface. The left sidebar contains a list of categories: Basic, Input, Music, Led, Radio, more, Loops, Logic, Variables, Math, SuperBitV2, SuperBitV2\_Digital, SuperBitV2\_Analog, SuperBitV2\_PWM, and Neonixel. The 'Radio' category is highlighted with a green box. The main script area on the right is divided into 'Group', 'Send', and 'Receive' sections. The 'Group' section contains a 'radio set group' block with the value '1'. The 'Send' section contains three blocks: 'radio send number' with value '0', 'radio send value' with 'name' and value '0', and 'radio send string' with an empty string. The 'Receive' section contains three 'on radio received' blocks: the first with 'receivedNumber', the second with 'name' and 'value', and the third with 'receivedString'. The third 'on radio received' block is highlighted with a green box. Below these is a 'received packet' block with 'signal strength' selected. At the bottom, there are three more blocks: 'radio set transmit power' with value '7', 'radio set transmit serial number' with 'true', and 'radio set frequency band' with value '0'. Finally, a 'radio raise event' block is shown with 'from source' set to 'MICROBIT\_ID\_BUTTON\_A' and 'with value' set to 'MICROBIT\_EVT\_ANY'.

**Left Sidebar:**

- Basic
- Input
- Music
- Led
- Radio**
- more
- Loops
- Logic
- Variables
- Math
- SuperBitV2
- SuperBitV2\_Digital
- SuperBitV2\_Analog
- SuperBitV2\_PWM
- Neonixel

**Main Script Area:**

**Group**

- radio set group 1

**Send**

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

**Receive**

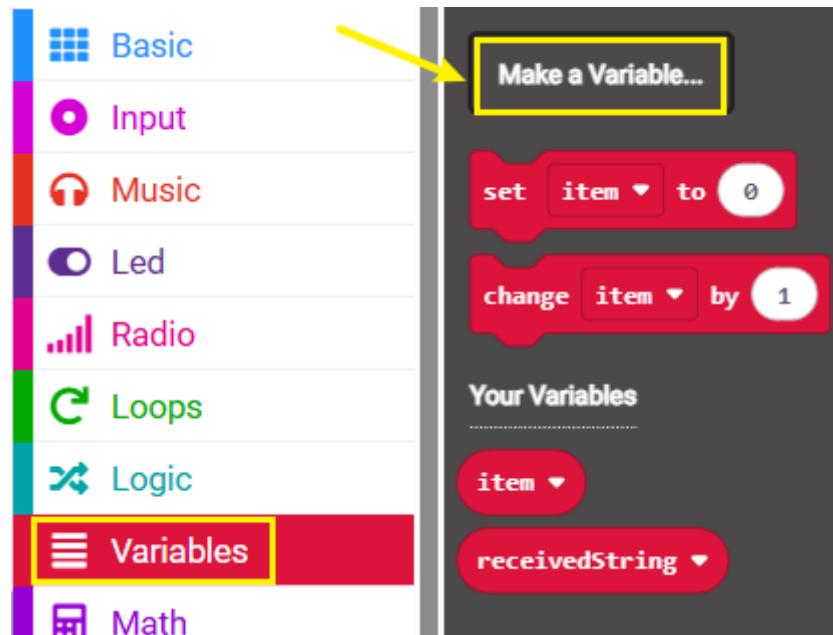
- on radio received receivedNumber
- on radio received name value
- on radio received receivedString**
- received packet signal strength

**Bottom Section:**

- 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 ---- **[Make a Variable]**.

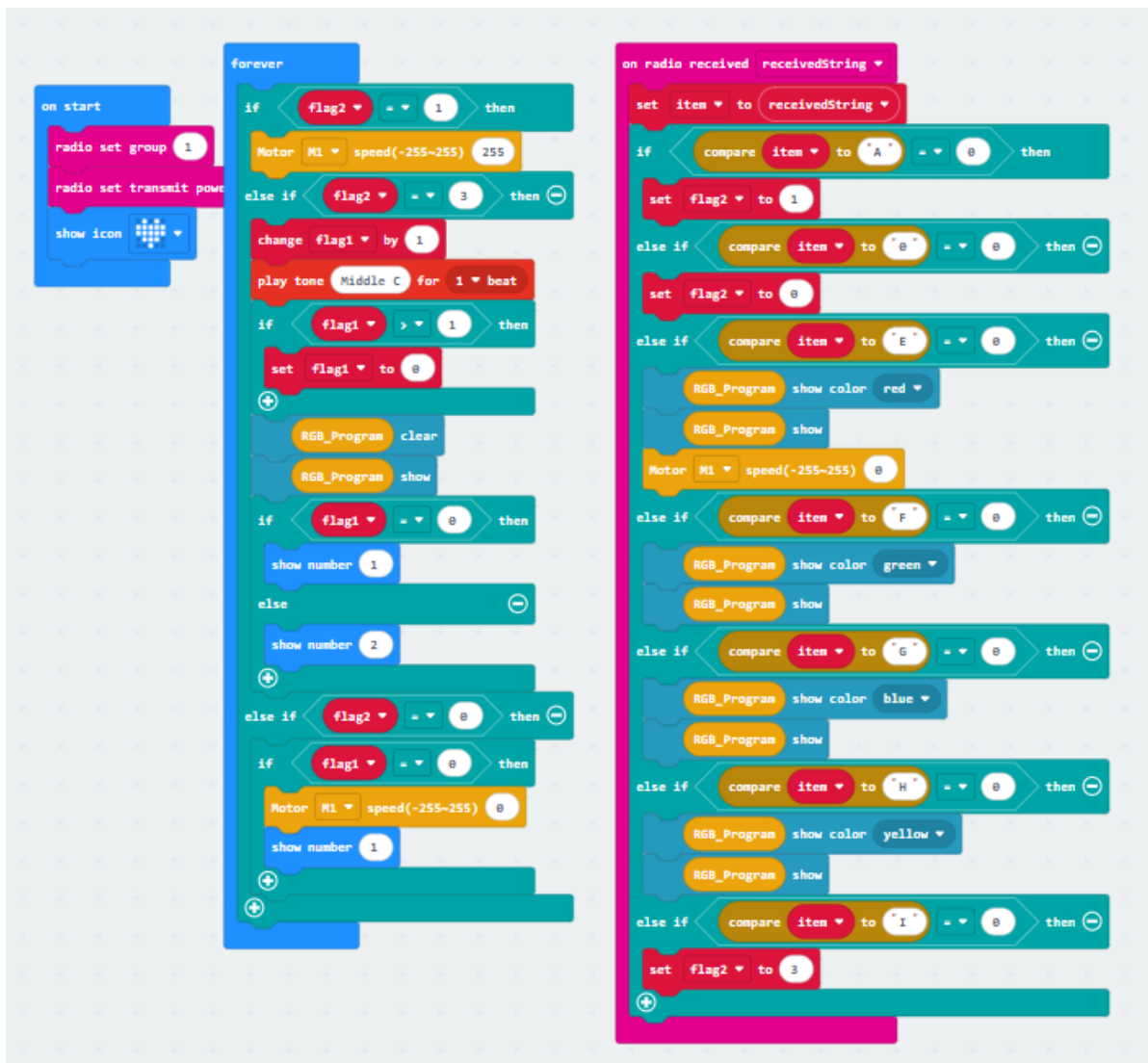


② Enter the variable name to complete the new variable.

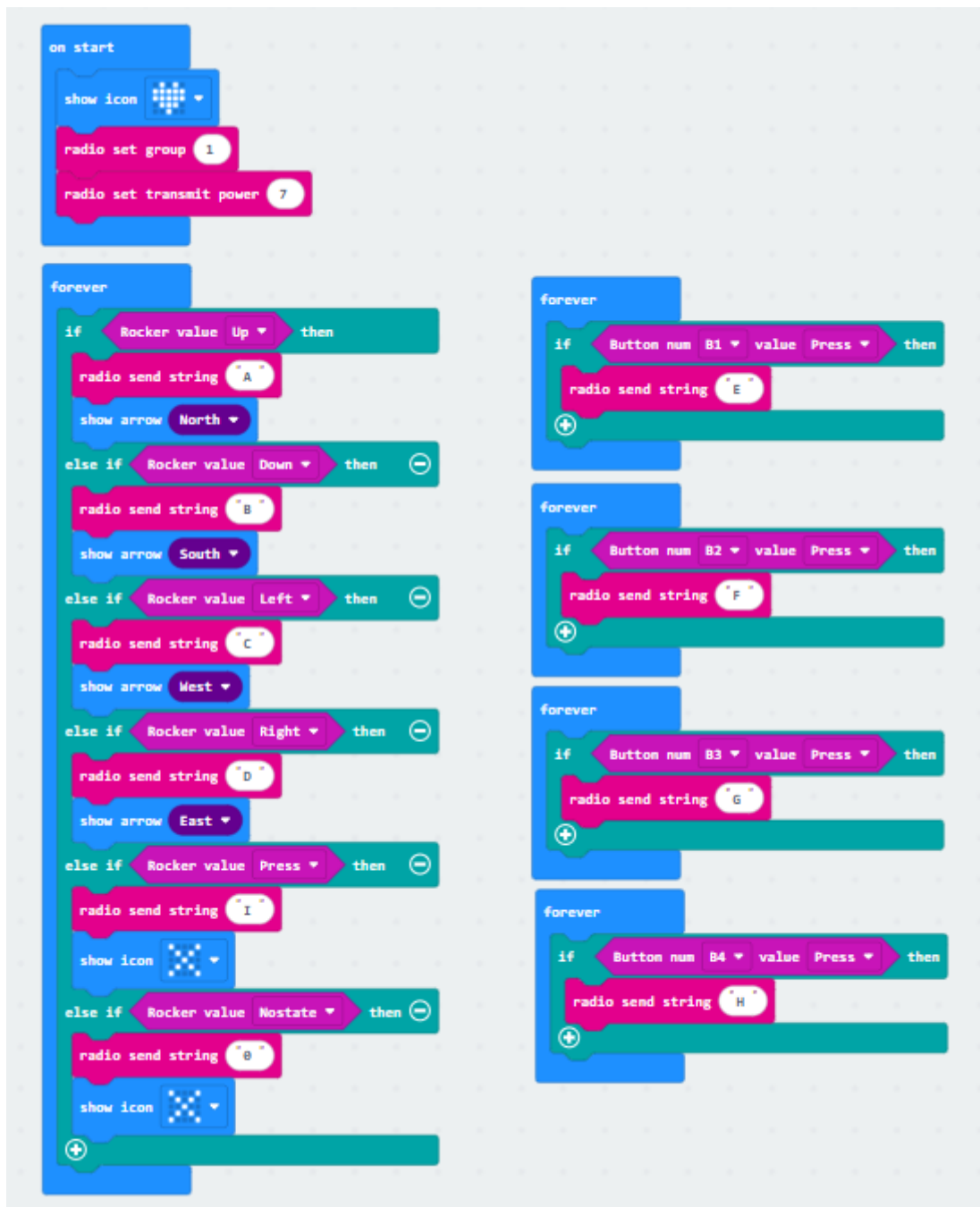
A screenshot of a dialog box titled 'New variable name:'. It has a close button (X) in the top right corner. Below the title is a text input field containing the word 'value'. At the bottom right of the dialog is a green button labeled 'Ok' with a checkmark icon. Red boxes highlight the input field and the 'Ok' button.

## 4.4 Combined blocks

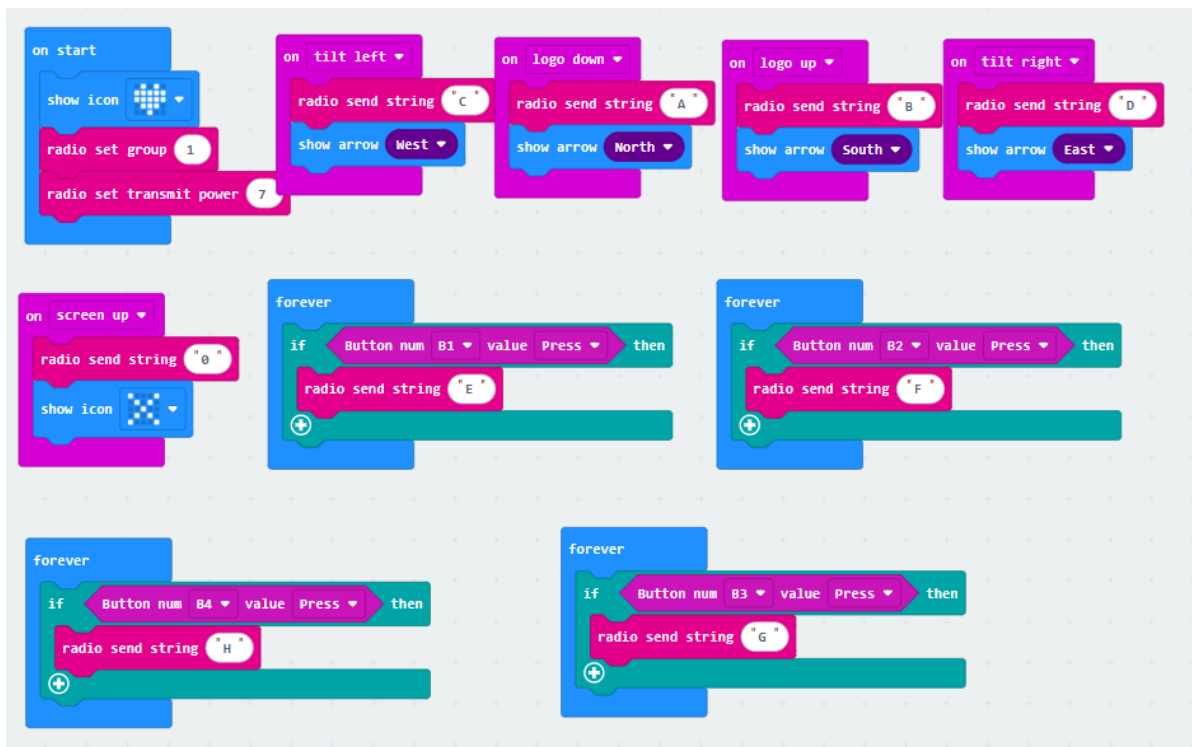
The microbit summary program on **Dragon Knight** is shown in the figure below.



Handle joystick control program is as follows.



Handle gravity control program is as follows.



You can also directly open the **microbit-Handle-control-Dragon-knight-code.hex**, **microbit-Handle-rocker-control.hex**, **microbit-Handle-gravity-control.hex**. Drag the file 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 Dragon Knight program to the micro:bit mainboard of the Dragon Knight, turn on the power switch of the Dragon Knight, and we can see that a heart pattern will be initialized 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 a heart pattern will be initialized on the micro:bit dot matrix, and then an "X" pattern will be displayed, 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 remote control of the Dragon Knight.

The handle functions are as follows.



**Joystick control:**

After the joystick and Dragon Knight are paired successfully, we can see the number 1 displayed on the micro:bit dot matrix of Dragon Knight, indicating that it is in mode 1 at this time.

In mode 1:

- Push the joystick forward to control the wings of Dragon Knight to flap, and stop when you release your hand;
- Press the red button to light up the red RGB light;
- Press the green button to light up the green RGB light;
- Press the yellow button to light up the yellow RGB light;
- Press the blue button to light up the blue RGB light.

We can press the joystick to switch to mode 2. At this time, we can see the number 2 displayed on the micro:bit dot matrix of Dragon Knight, indicating that it is in mode 2.

In mode 2:

- Push the joystick forward to control the dragon knight's wings to flap, and the wings will still flap when you release your hand;
- Press the red button to light up the red RGB light and stop the moving dragon knight;
- Press the green button to light up the green RGB light;
- Press the yellow button to light up the yellow RGB light;
- Press the blue button to light up the blue RGB light.

Each time you press the joystick, it will switch back and forth between mode 1 and mode 2, and the RGB light will go out.

### **Handle gravity control:**

After the handle and the dragon knight are successfully paired, we can see the number 1 displayed on the micro:bit dot matrix of the dragon knight.

- Tilt the handle forward to control the dragon knight's wings to flap, and stop the movement when placed horizontally;
- Press the red button to light up the red RGB light;
- Press the green button to light up the green RGB light;
- Press the yellow button to light up the yellow RGB light;
- Press the blue button to light up the blue RGB light.

**(Note: There is only one mode for handle gravity control, and mode 2 cannot be switched)**