Microbit handle control

1. Learning objectives

In this course, we mainly learn how to use MakeCode graphical programming to realize the microbit handle to control the creeping car.

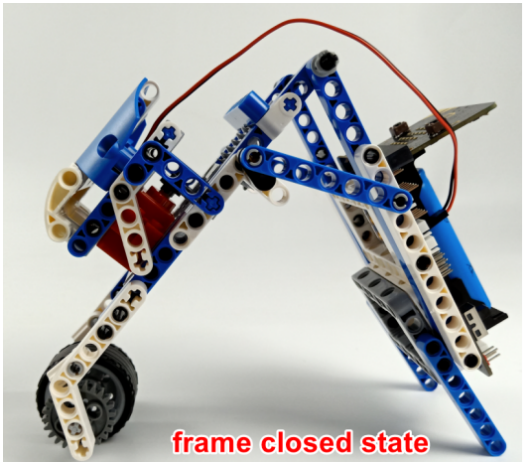
Principle of creeping car movement:

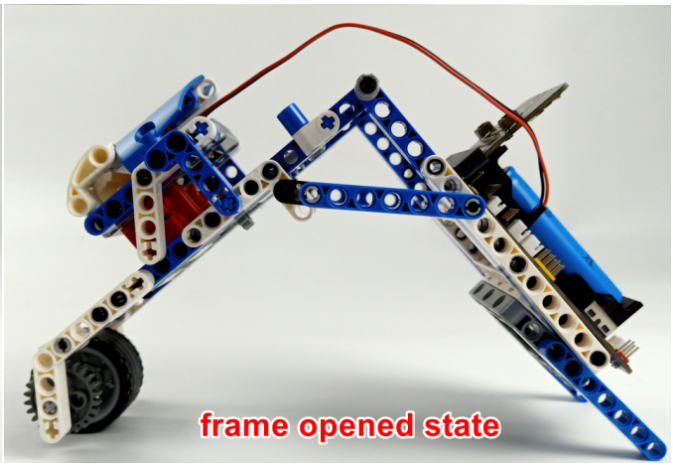
The friction of the front wheel is changed by adjusting the 1# bolt connection buckle ratchet to control the forward direction of the car. When the 1# bolt connection is located in front of the 24-tooth gear, the front wheel can only move forward, so the car creeps forward; when the 1# bolt is connected to the back of the 24-tooth gear, the front wheel can only move backward, and the car creeps backward.

2. Building blocks

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

After assembly, the frame of the creeping car needs to be adjusted to the closed state.

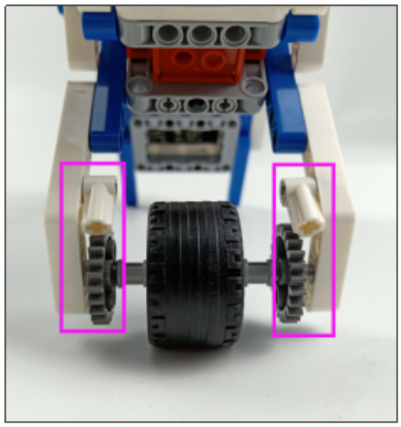




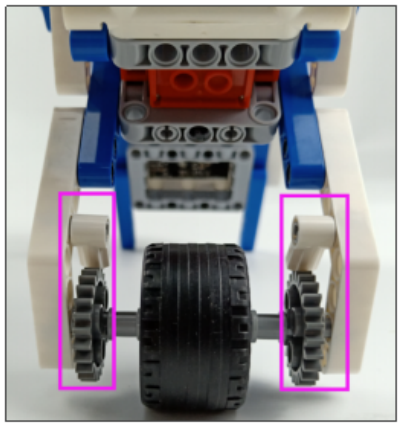
! Special Notes:

Place the two ratchets in front of the 24-tooth gear so that the crawler can move forward.

Place the two ratchets behind the 24-tooth gear so that the crawler can move backward.



【1# bolt connector are placed in front of the 24-tooth gear】

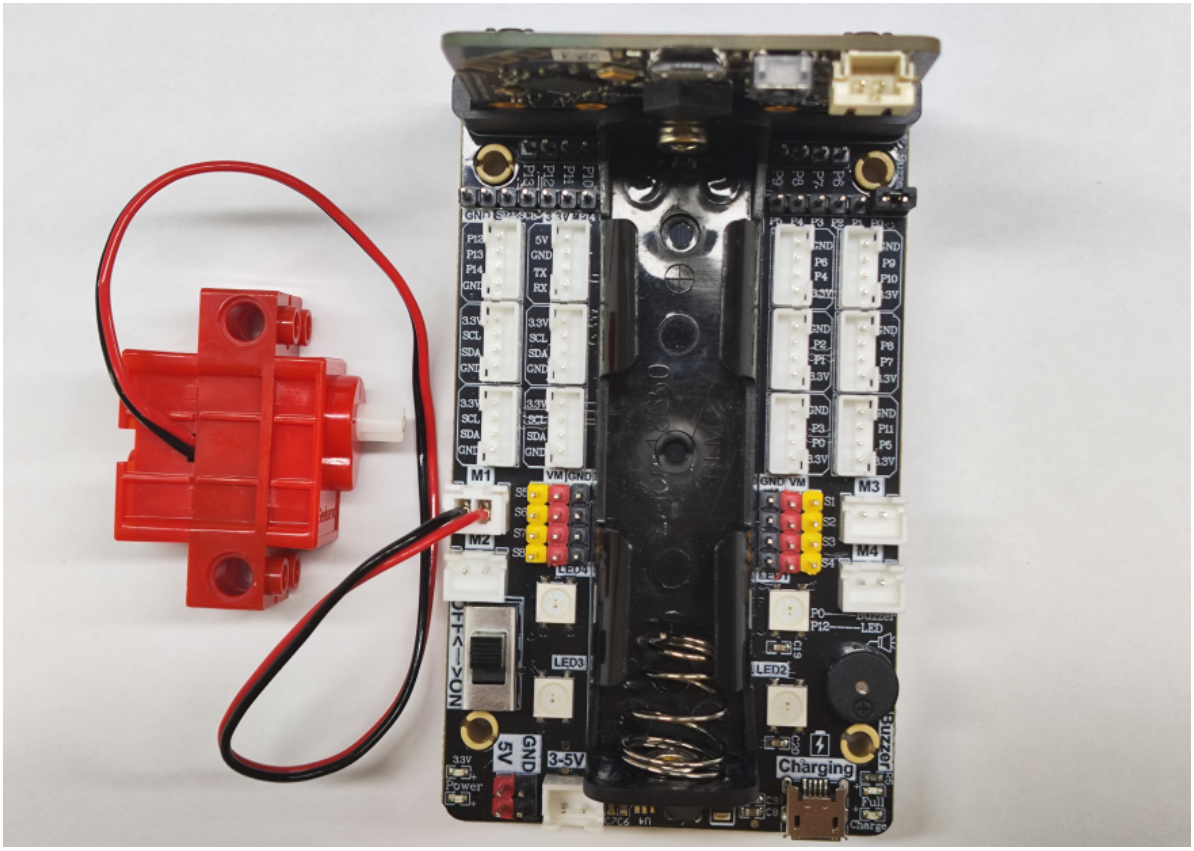


【1# bolt connector are placed behind the 24-tooth gear】

3. Motor wiring

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

As shown below:



4. Programming

Method 1 Online programming:

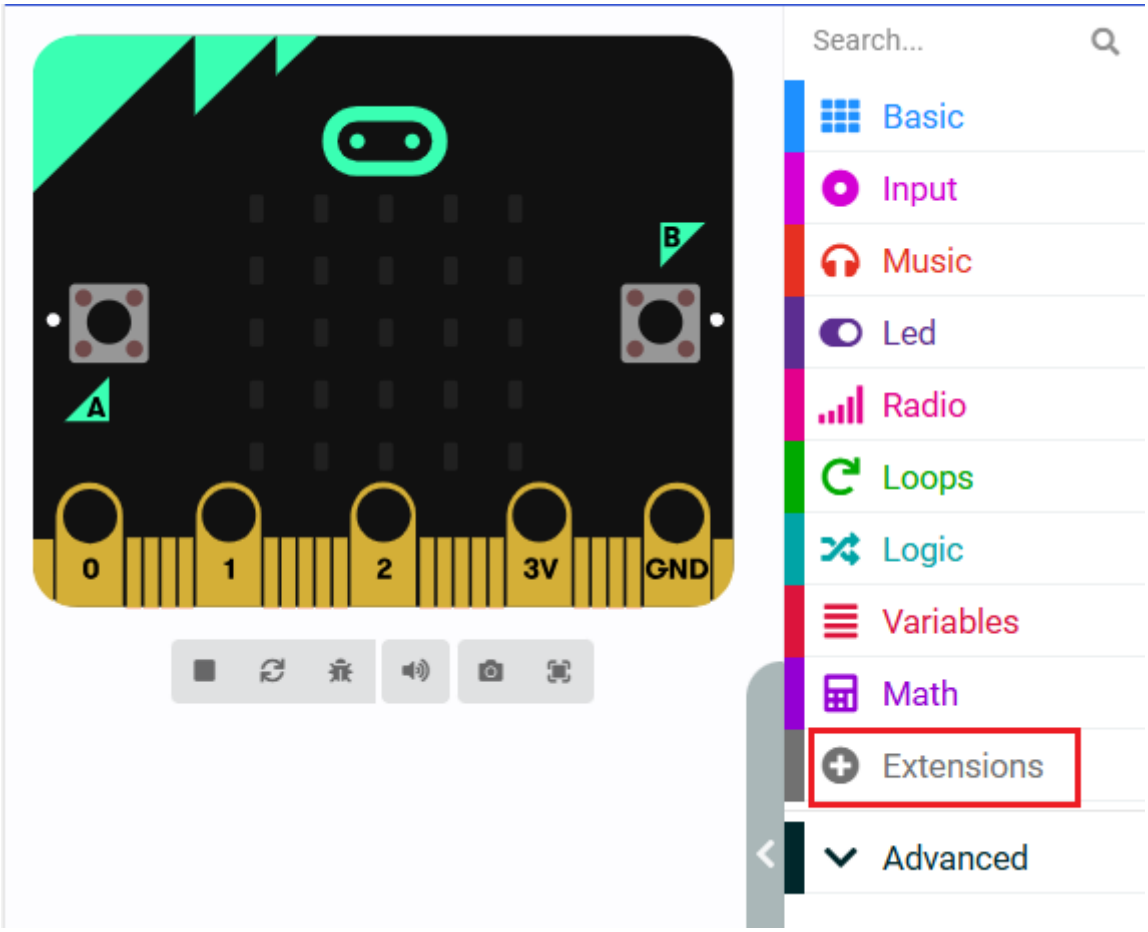
First, connect micro:bit to the computer via a USB data cable. The computer will pop up a U disk. Click the URL in the U disk: https://makecode.microbit.org/ to enter the programming interface. Then, add the Yahboom software package to start programming.

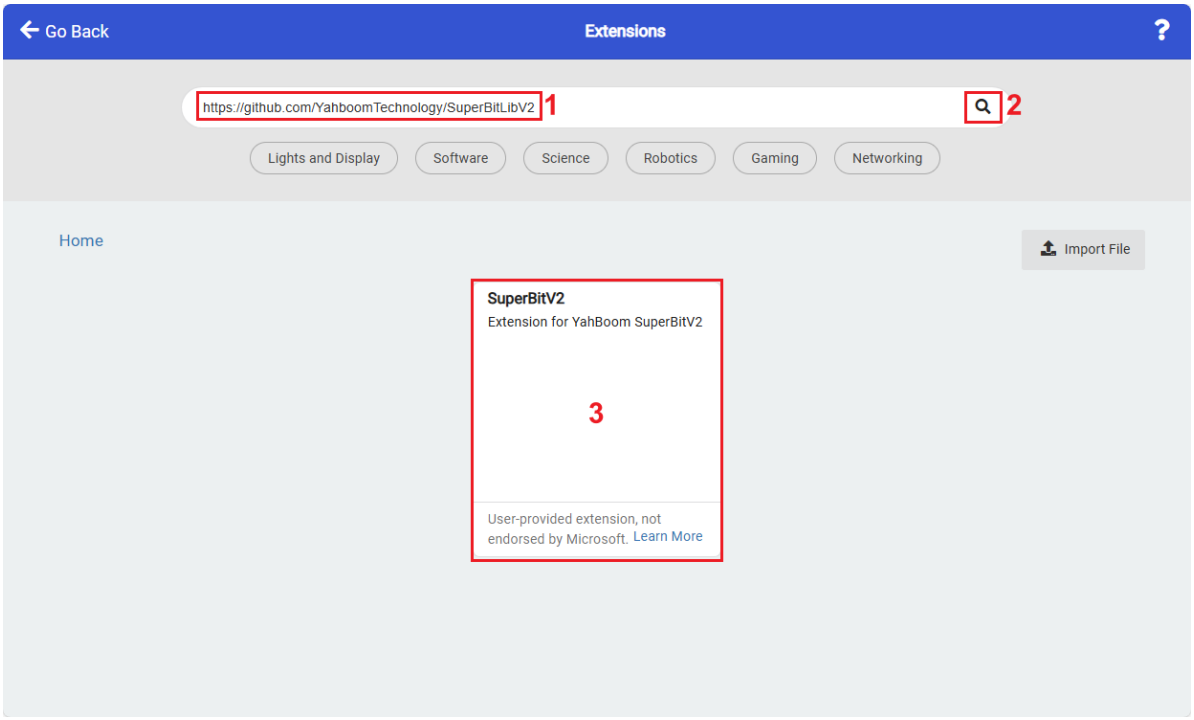
Method 2 Offline programming:

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

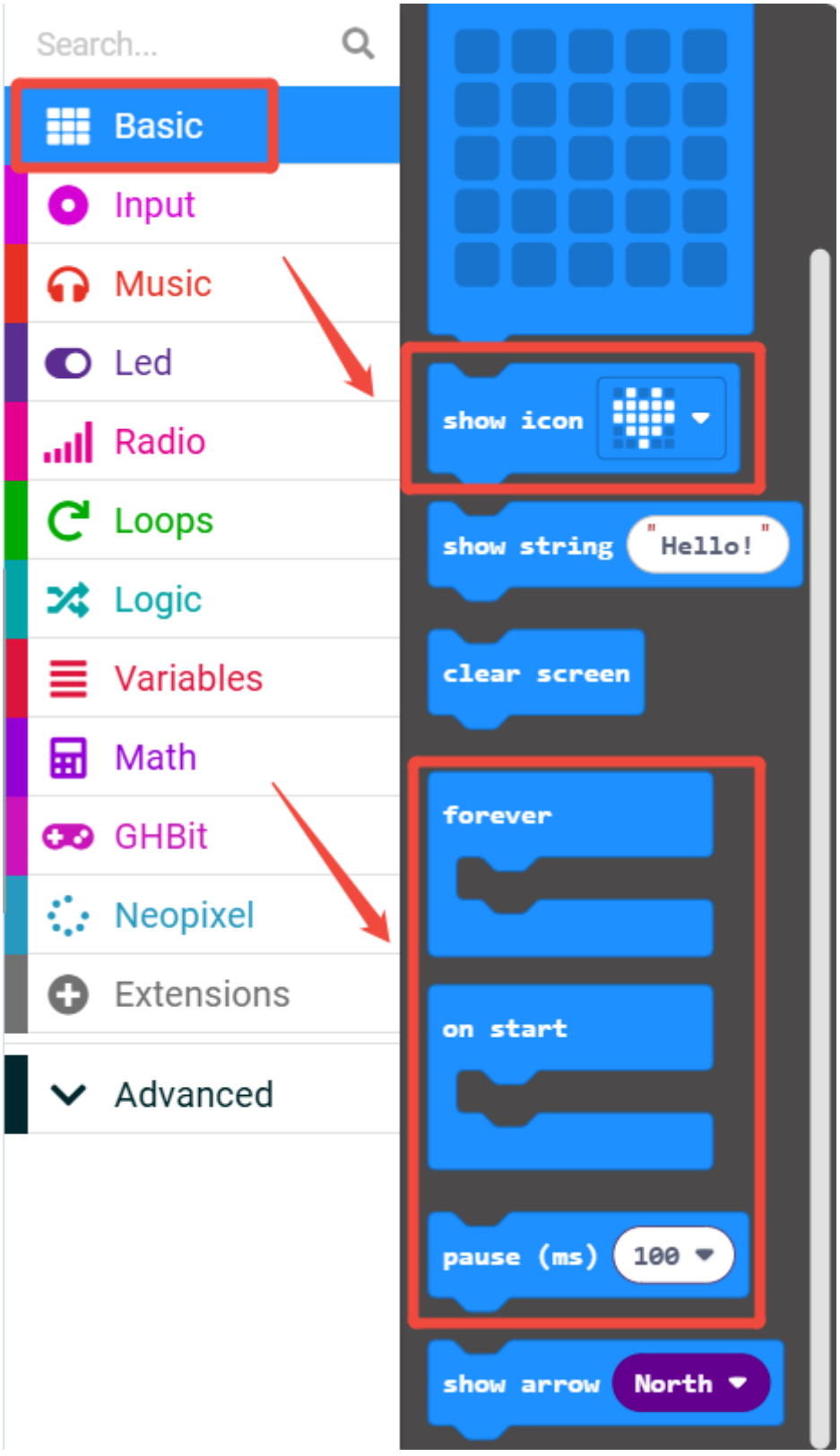
4.1 Add expansion package

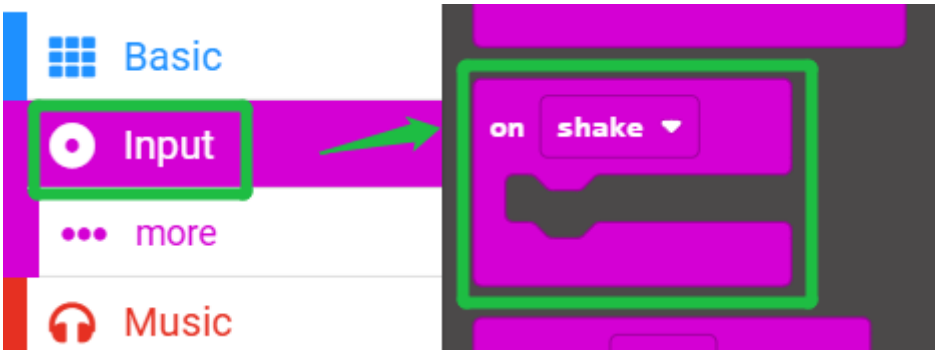




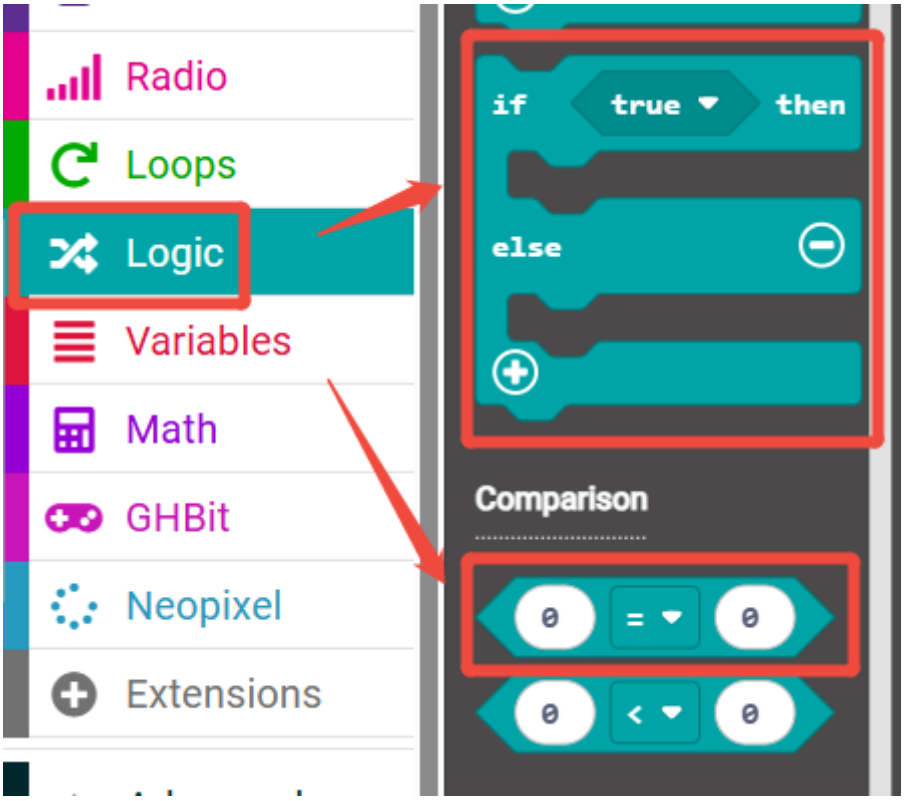
4.2 Blocks used

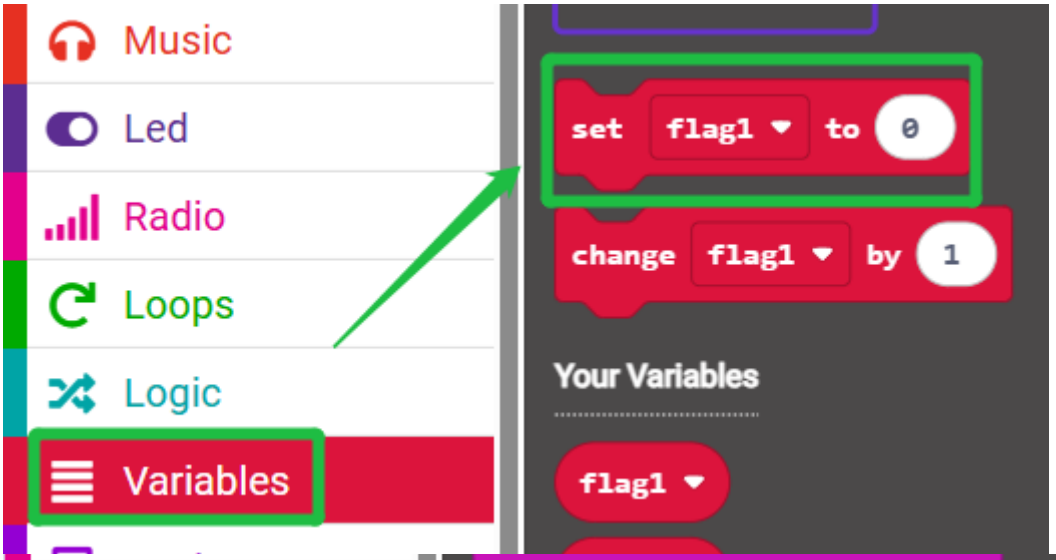
The location of the blocks required for this programming is shown in the figure below.

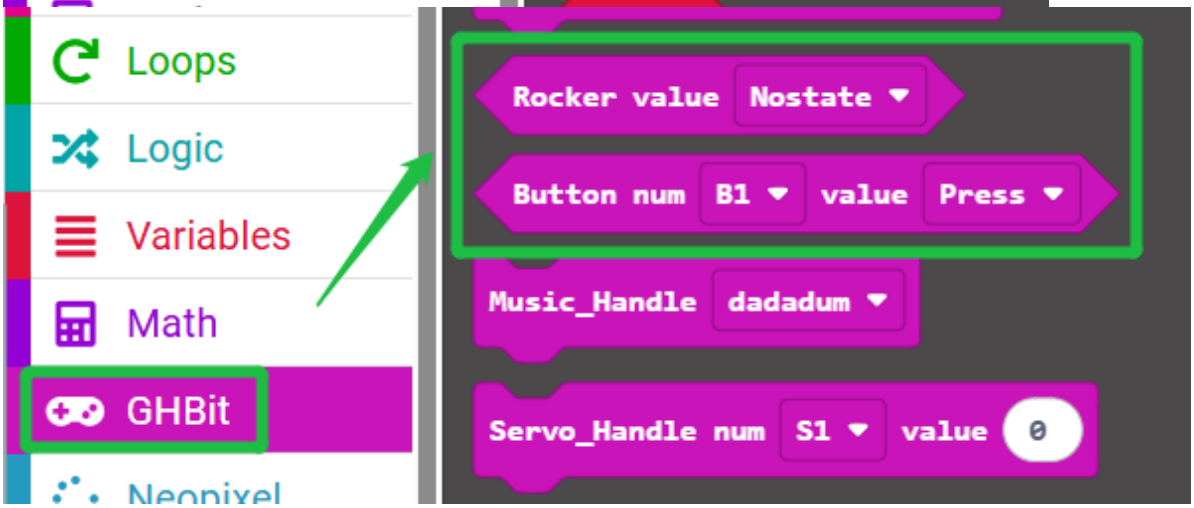

Search...
Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
GHBit
Neopixel
Extensions
Advanced
show icon
show string "Hello!"
clear screen
forever
on start
pause (ms) 100
show arrow North

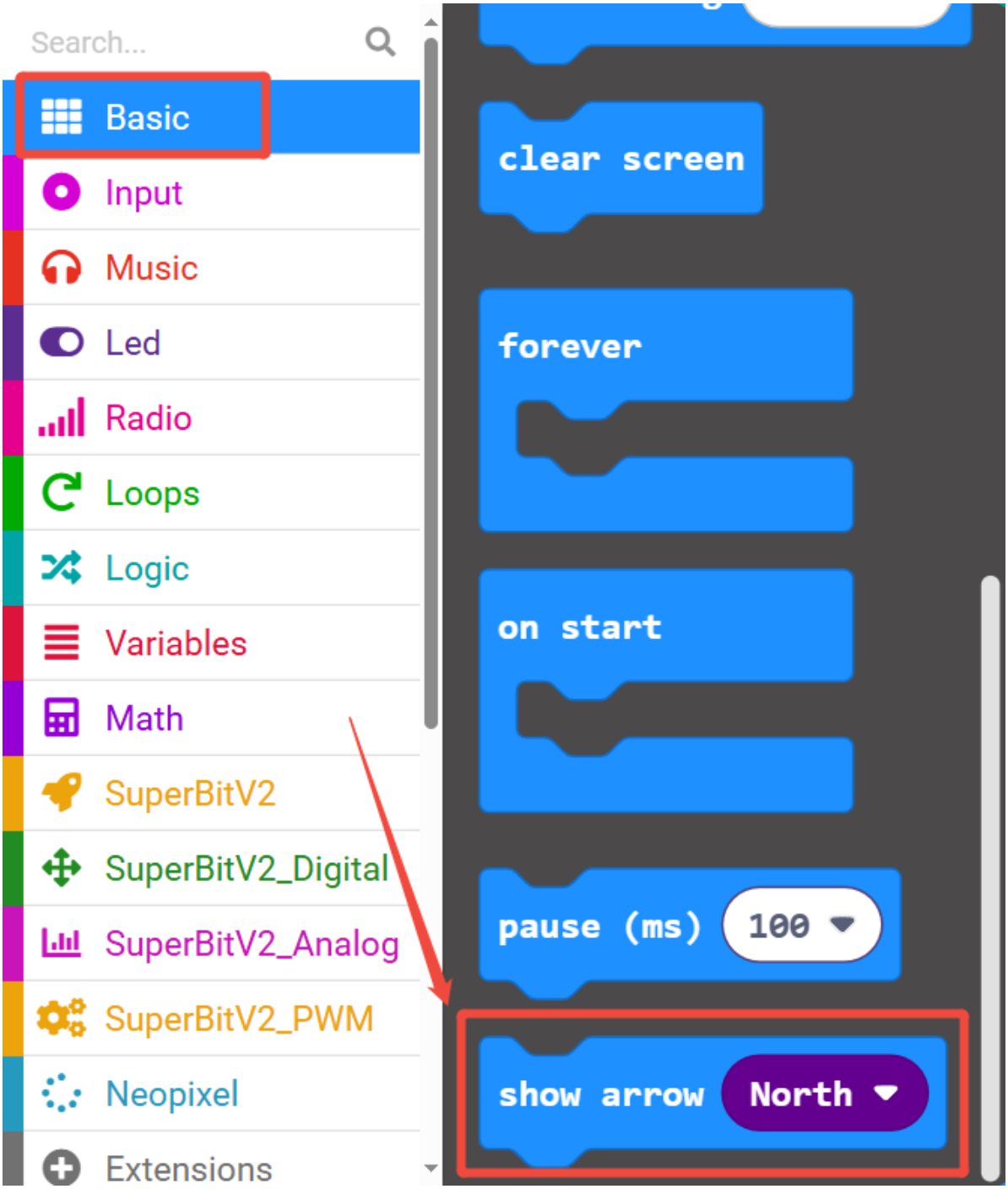

Basic
Input
more
Music
on shake

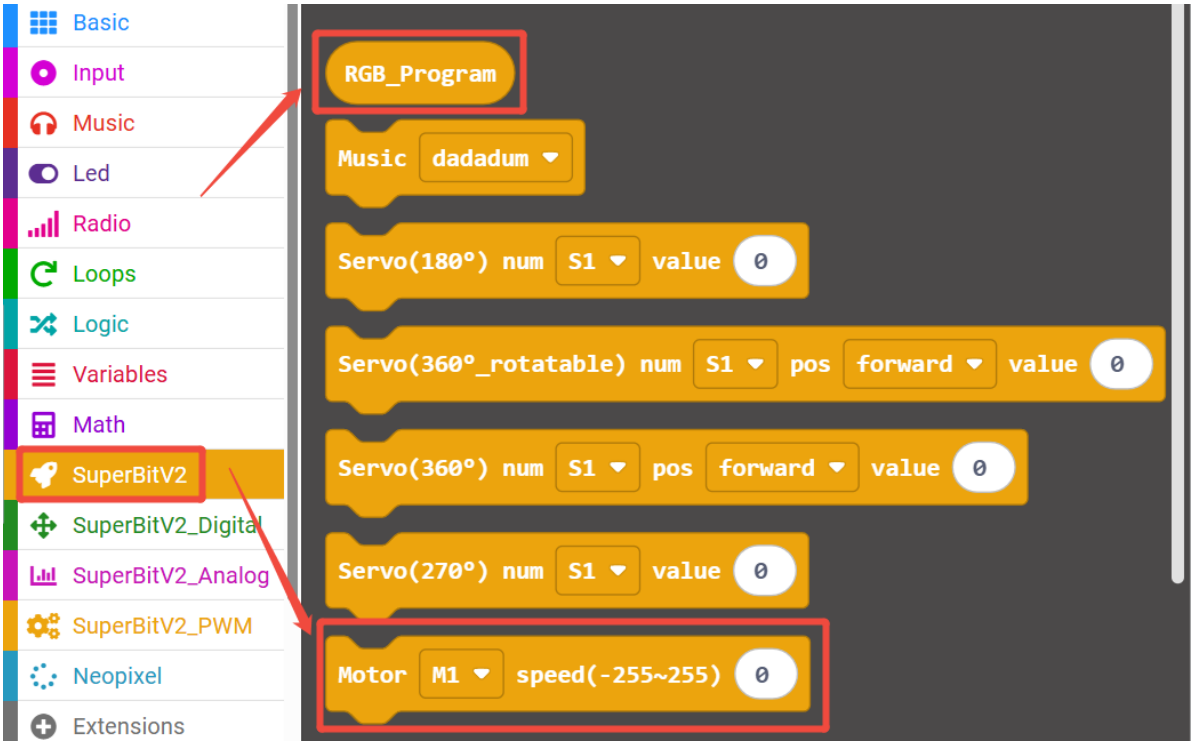

Search...
Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
GHBit
Neopixel
Extensions
Advanced
Functions
Arrays
Text
Game
Text
length of "Hello"
join "Hello" "World"
parse to number "123"
split "this" at " "
"this" includes " "
"this" find index of " "
"this" is empty
substring of "this" from 0 of length 10
compare "this" to " "
char from "this" at 0
char code from "this" at 0

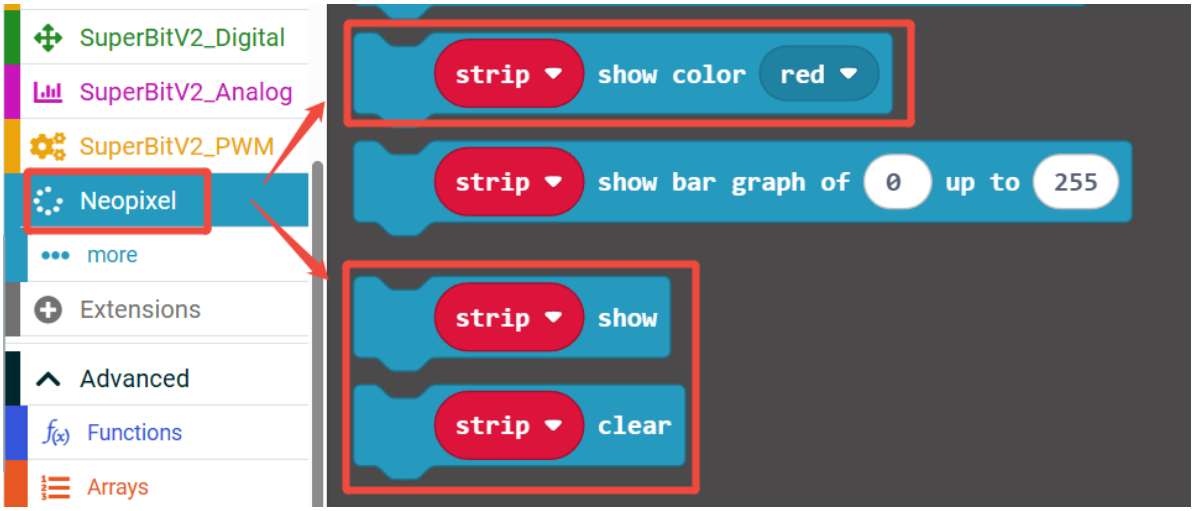

Radio
Loops
Logic
Variables
Math
GHBit
Neopixel
Extensions
if true then
else
Comparison
0 = 0
0 < 0

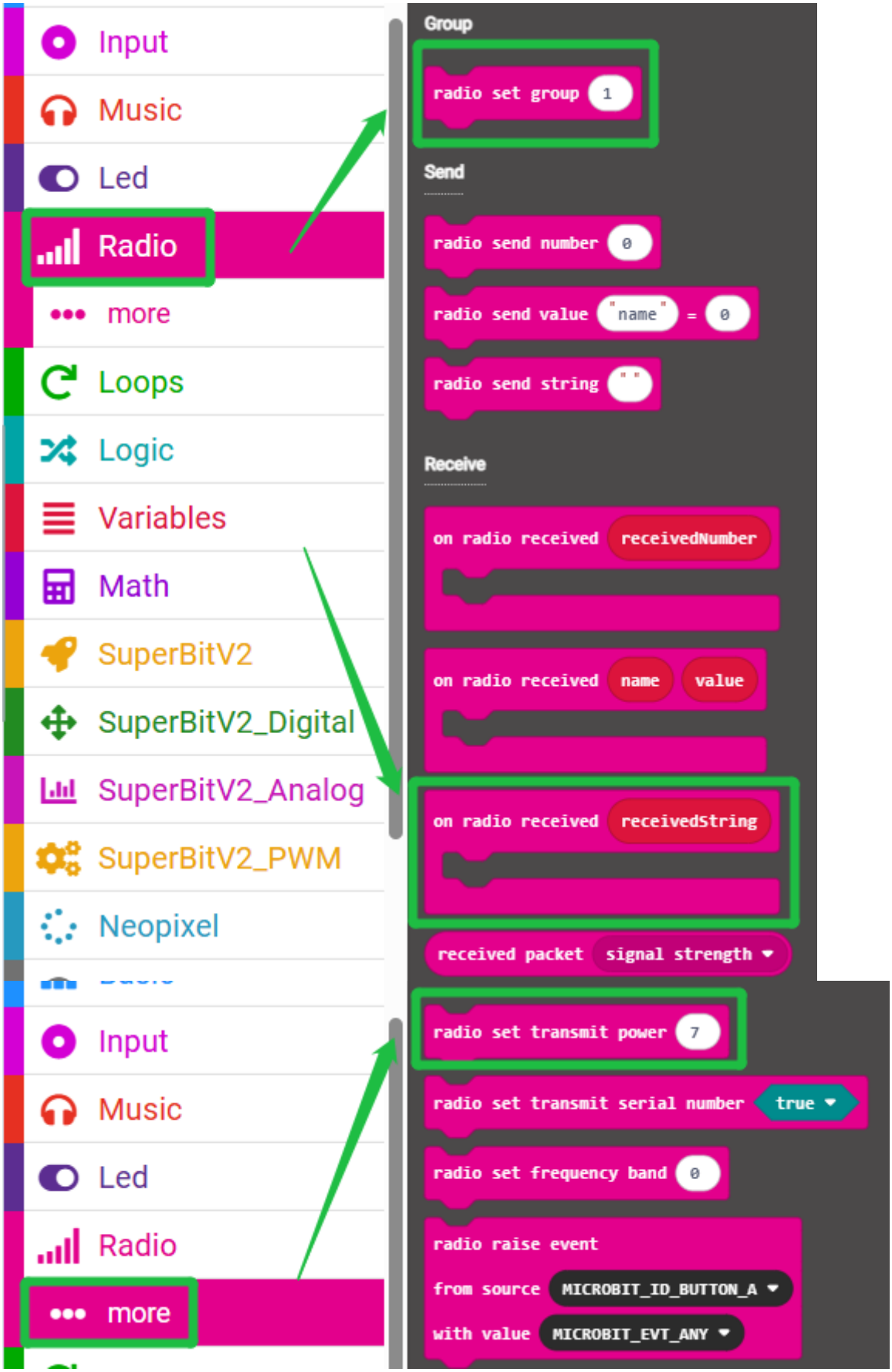

Music
Led
Radio
Loops
Logic
Variables
set flag1 to 0
change flag1 by 1
Your Variables
flag1


Loops
Logic
Variables
Math
GHBit
Neopixel
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
clear screen
forever
on start
pause (ms) 100
show arrow North

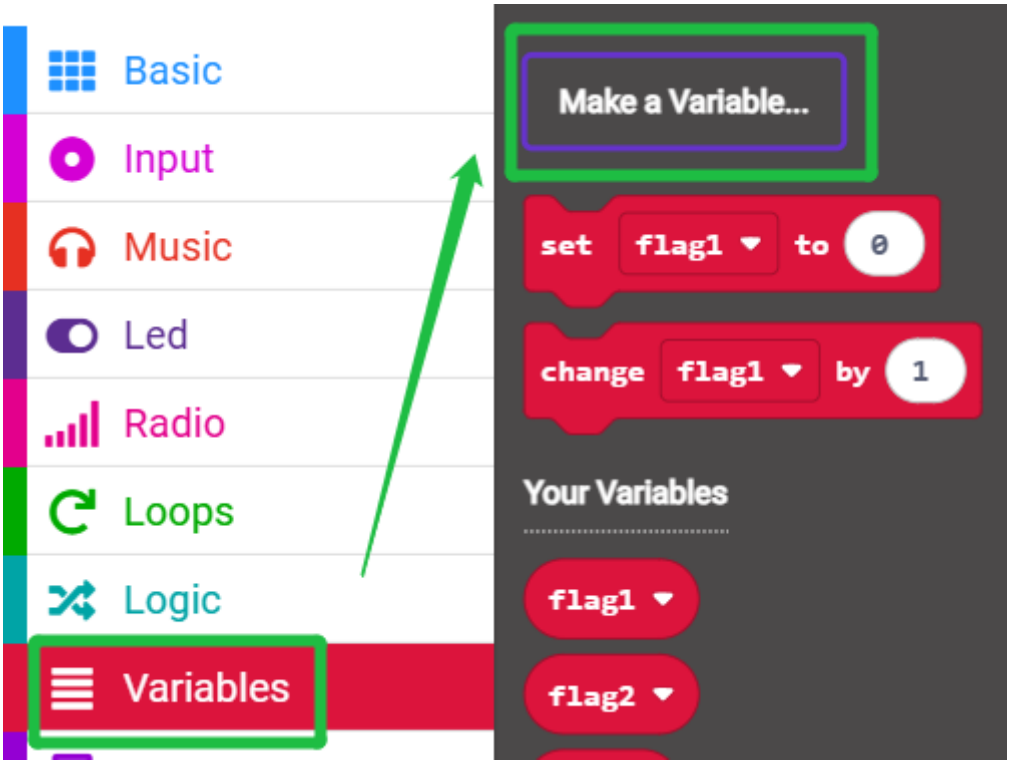

Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
SuperBitV2
SuperBitV2_Digital
SuperBitV2_Analog
SuperBitV2_PWM
Neopixel
Extensions
RGB_Program
Music dadadum
Servo(180°) num S1 value 0
Servo(360°_rotatable) num S1 pos forward value 0
Servo(360°) num S1 pos forward value 0
Servo(270°) num S1 value 0
Motor M1 speed(-255~255) 0


SuperBitV2_Digital
SuperBitV2_Analog
SuperBitV2_PWM
Neopixel
more
Extensions
Advanced
Functions
Arrays
strip show color red
strip show bar graph of 0 up to 255
strip show
strip clear

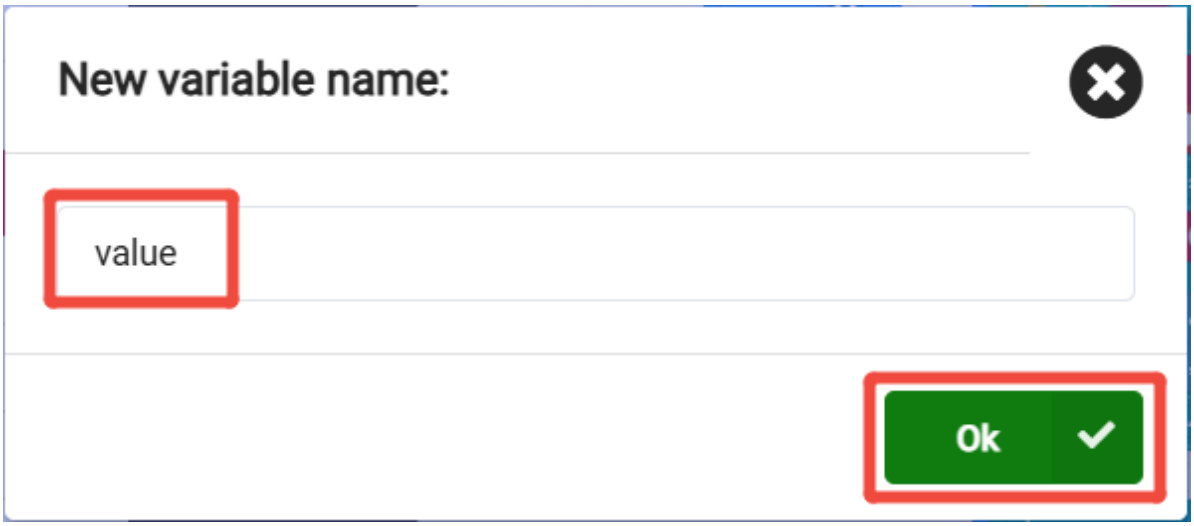


4.3 Add new variables

① Find the [Variable] option in the building block bar ---- **[Make a Variable]**

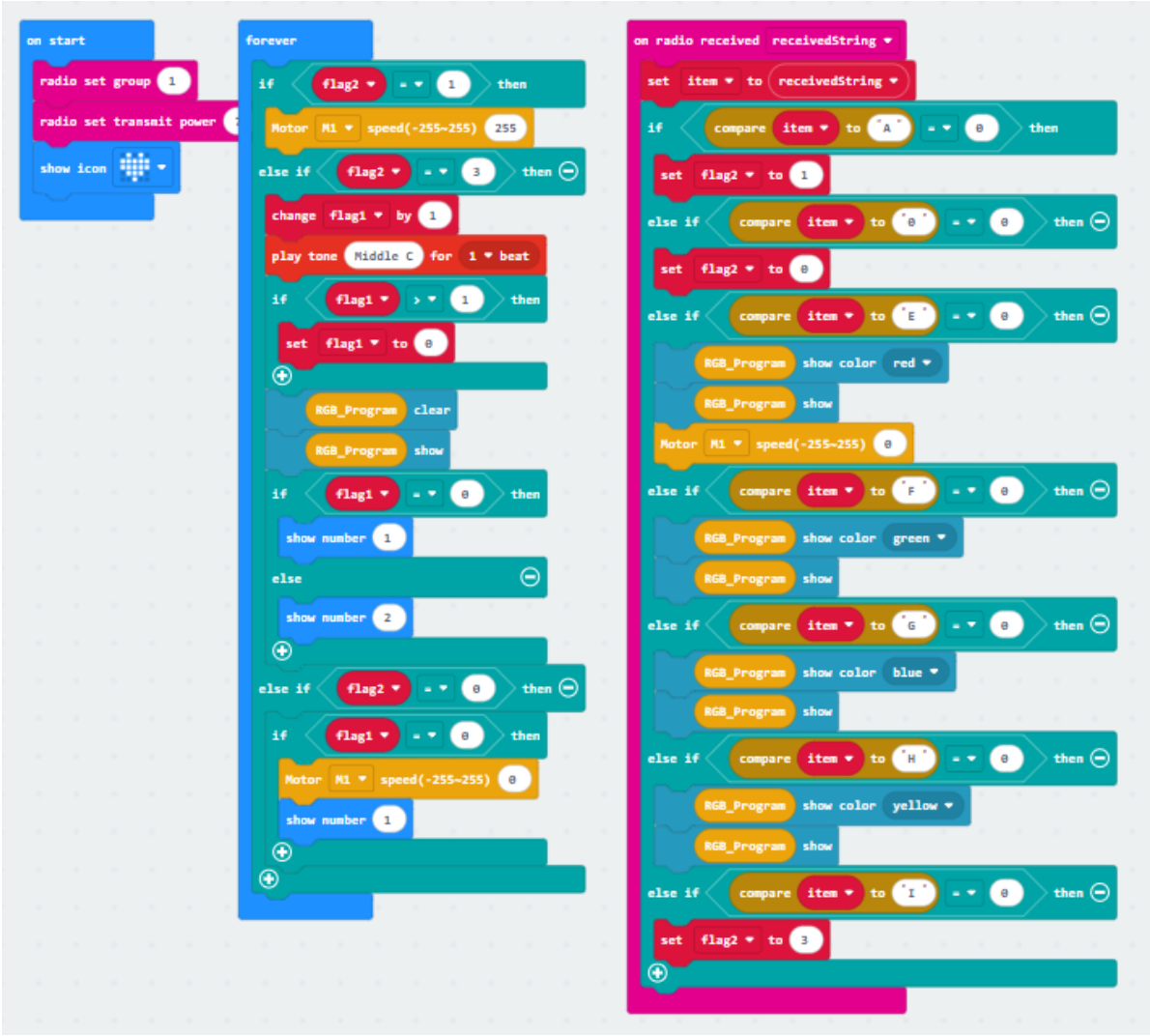


② Enter the variable name to complete the new variable.

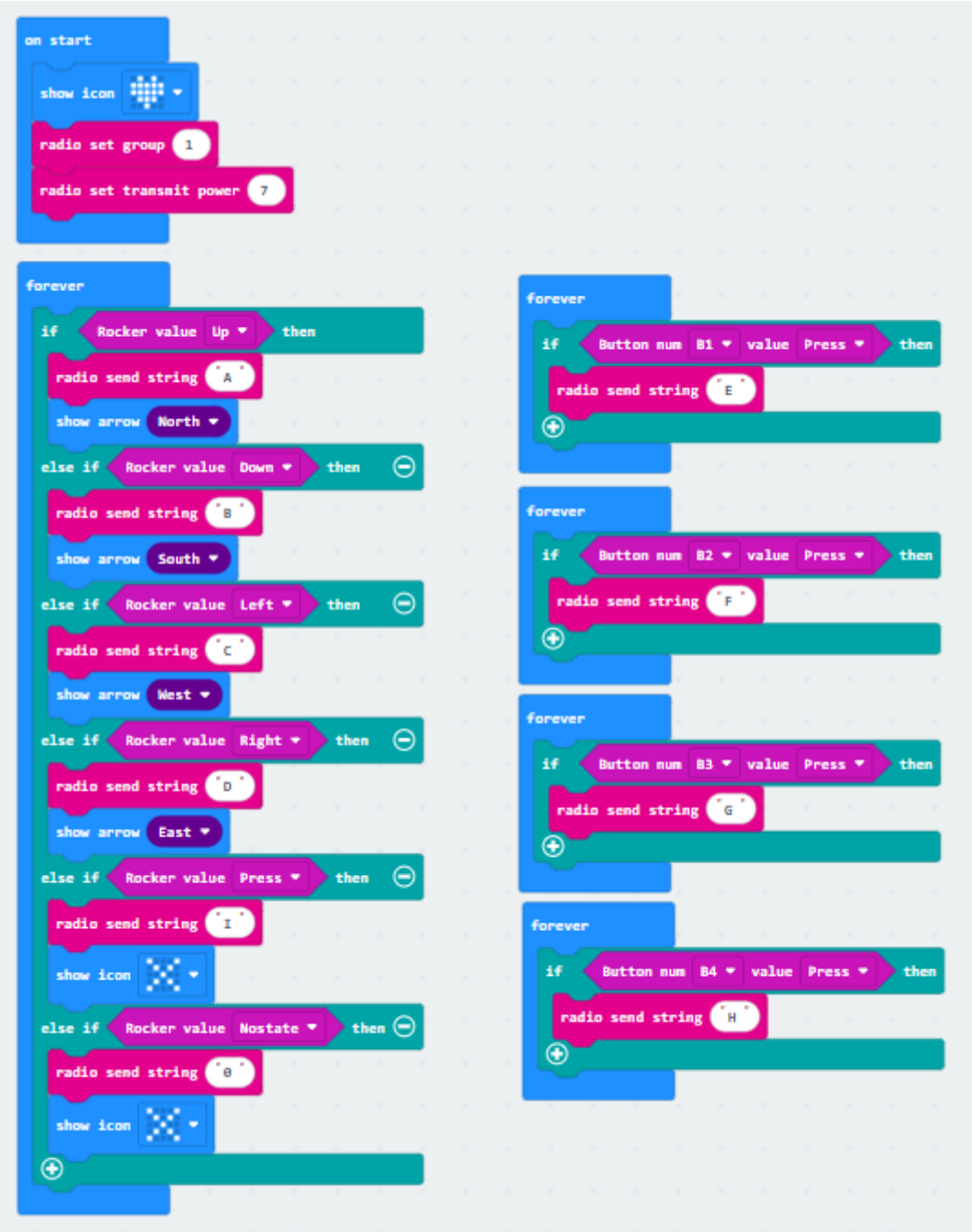


4.4 Combined blocks

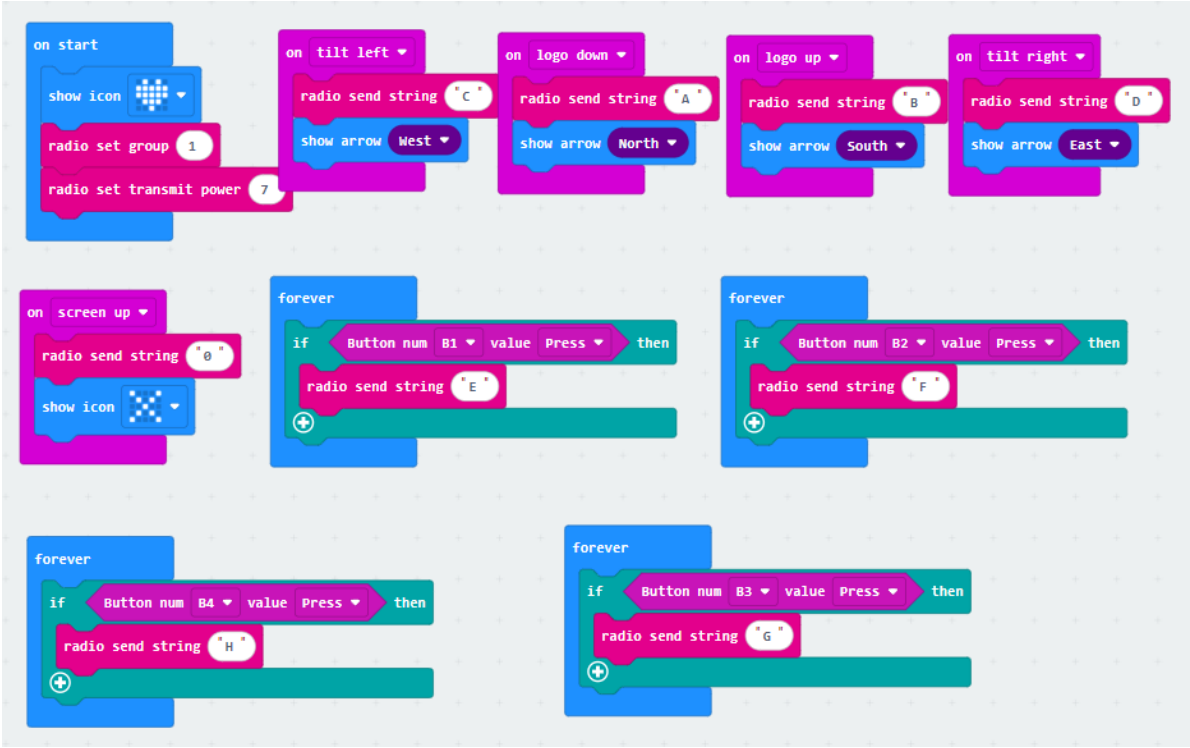
The microbit summary program on the **squirm car** is shown in the figure below



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-Unicycle-code.hex, microbit-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 program of the peristaltic car to the micro:bit motherboard of the peristaltic car, turn on the power switch of the peristaltic car, 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 motherboard 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 peristaltic car.

The functions of the handle are as follows.

Handle joystick control:

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

In mode 1:

- Push the joystick forward to control the movement of the creeper car, and it will 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 that the number 2 is displayed on the micro:bit dot matrix of the creeper car, indicating that it is in mode 2.

In mode 2:

- Push the joystick forward to control the movement of the peristaltic car, and it will keep rotating when you release your hand;
- Press the red button to light up the red RGB light and stop the moving peristaltic car;
- 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 peristaltic car are successfully paired, we can see that the number 1 is displayed on the micro:bit dot matrix of the peristaltic car.

- Tilt the handle forward to control the movement of the peristaltic car, 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)