Micro:bit handle control

1. Learning objectives

In this course, we mainly learn how to use MakeCode graphical programming to control Spider with the microbit handle.

2. Building blocks

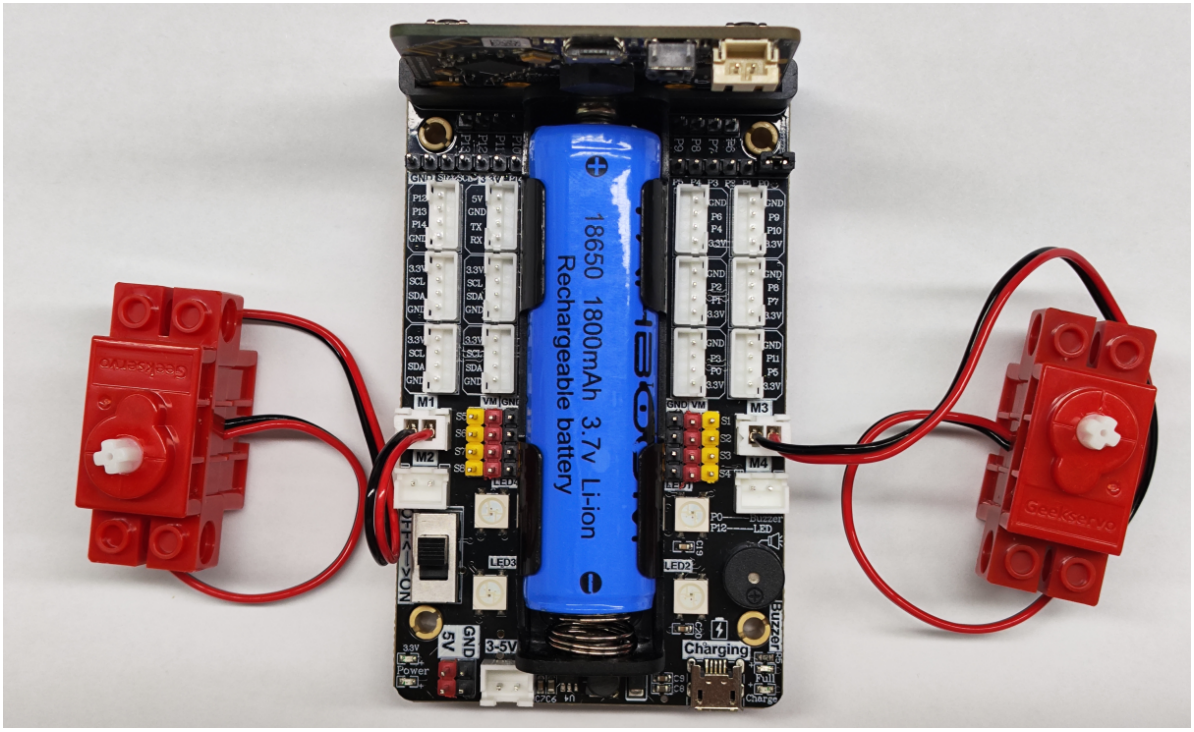
For detailed steps of building blocks, please refer to the installation drawings of **[Assembly Course]--[Spider]** in the materials or the building block installation album.

3. Motor wiring

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

The motor wiring on the right side of the car is inserted into the M3 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:

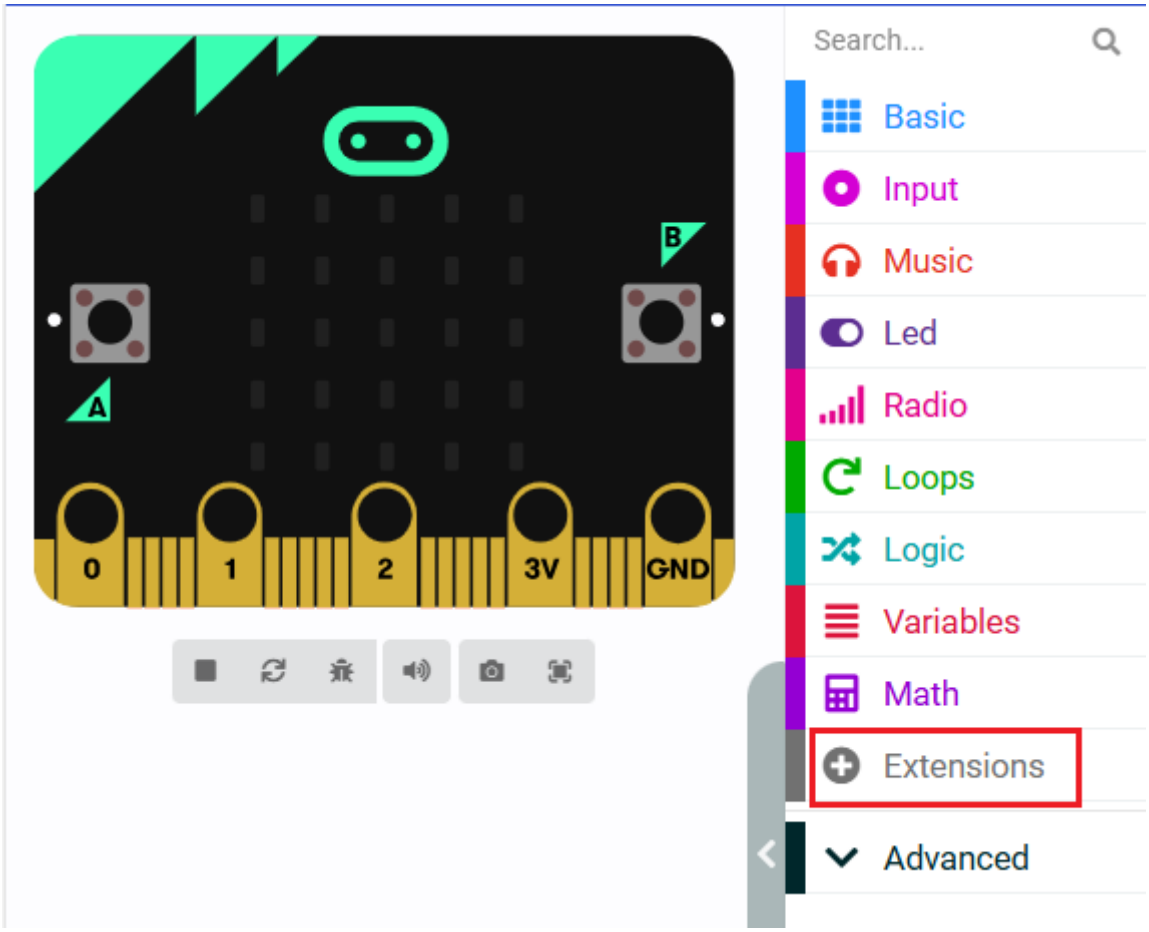
First, connect micro:bit to the computer via a USB data cable. The computer will pop up a U disk. Click the URL in the U disk: https://makecode.microbit.org/ to enter the programming interface. Then, add the Yahboom software package to start programming.

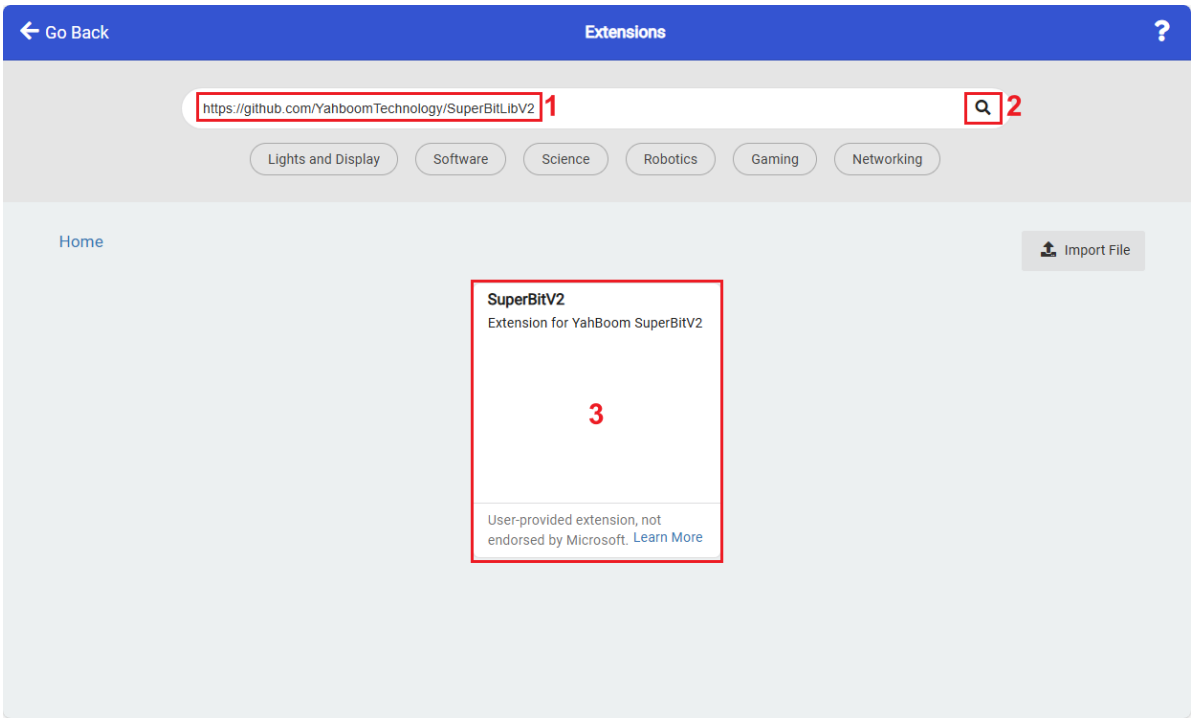
Method 2 Offline programming:

Open the offline programming software MakeCode and enter the programming interface. Click [New] and then 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

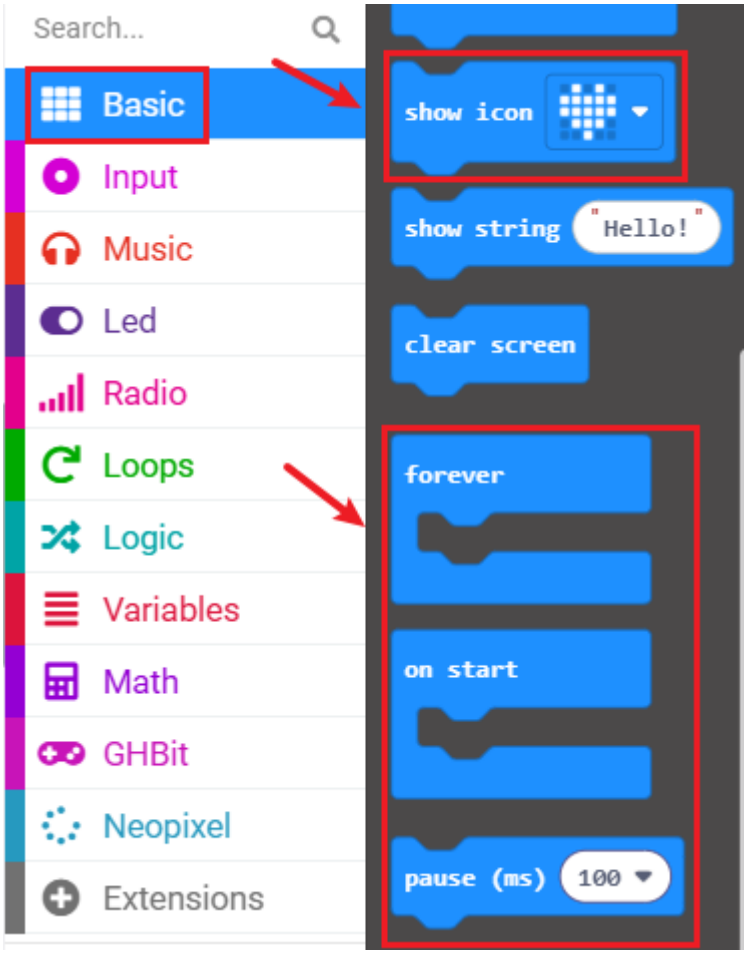
4.1 Add expansion package





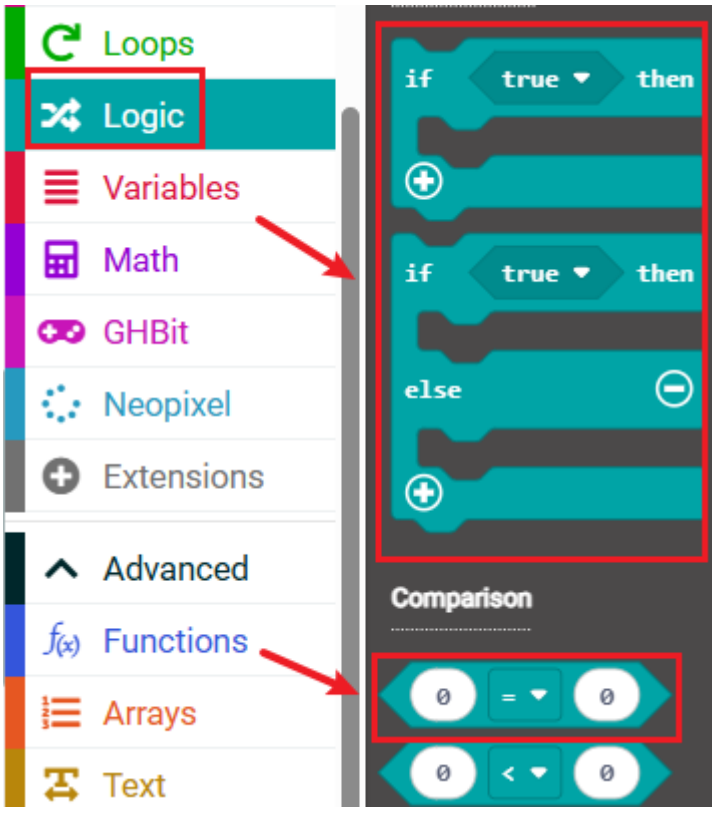
4.2 Blocks used

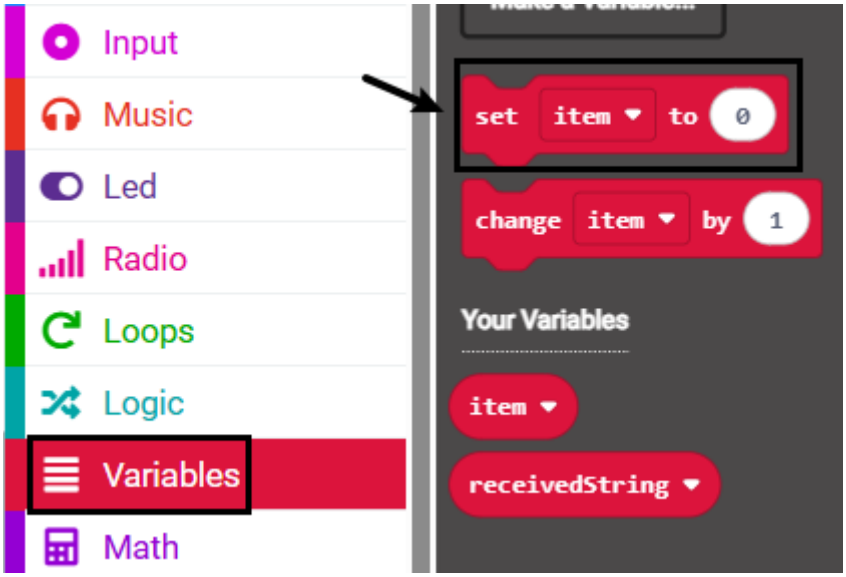
The location of the blocks required for this programming is shown in the figure below.

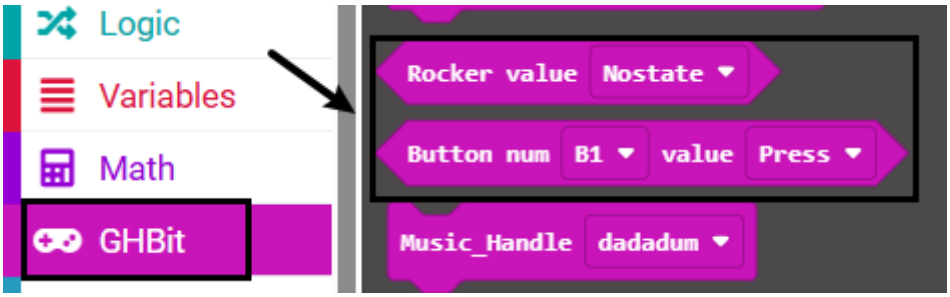

Search...
Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
GHBit
Neopixel
Extensions
show icon
show string
"Hello!"
clear screen
forever
on start
pause (ms)
100


Basic
Input
more
Music
Led
Radio
on button
A
pres
on
shake

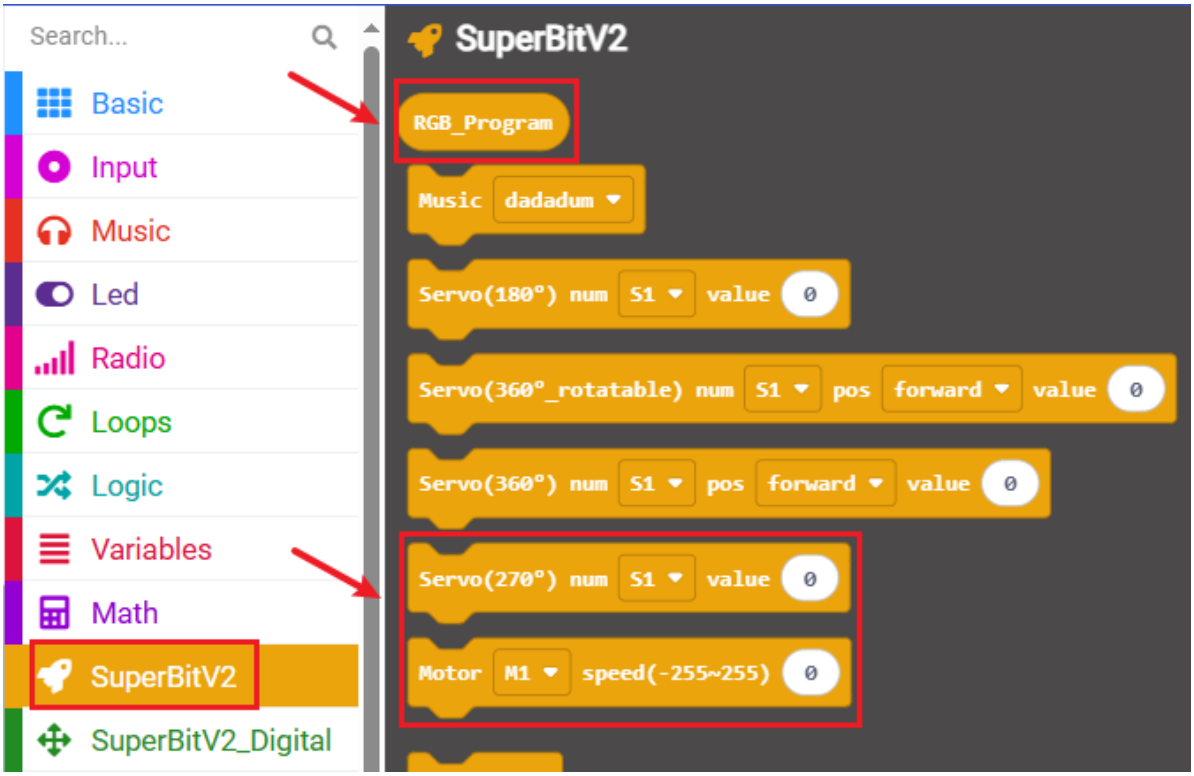

Led
Radio
Loops
Logic
Variables
Math
GHBit
Neopixel
Extensions
Advanced
Functions
Arrays
Text
Game
Images
Text
length of "Hello"
join "Hello" "World"
parse to number "123"
split "this" at " "
"this" includes " "
"this" find index of " "
"this" is empty
substring of "this" from 0 of length 10
compare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


Loops
Logic
Variables
Math
GHBit
Neopixel
Extensions
Advanced
Functions
Arrays
Text
if true then
if true then
else
Comparison
0 = 0
0 < 0

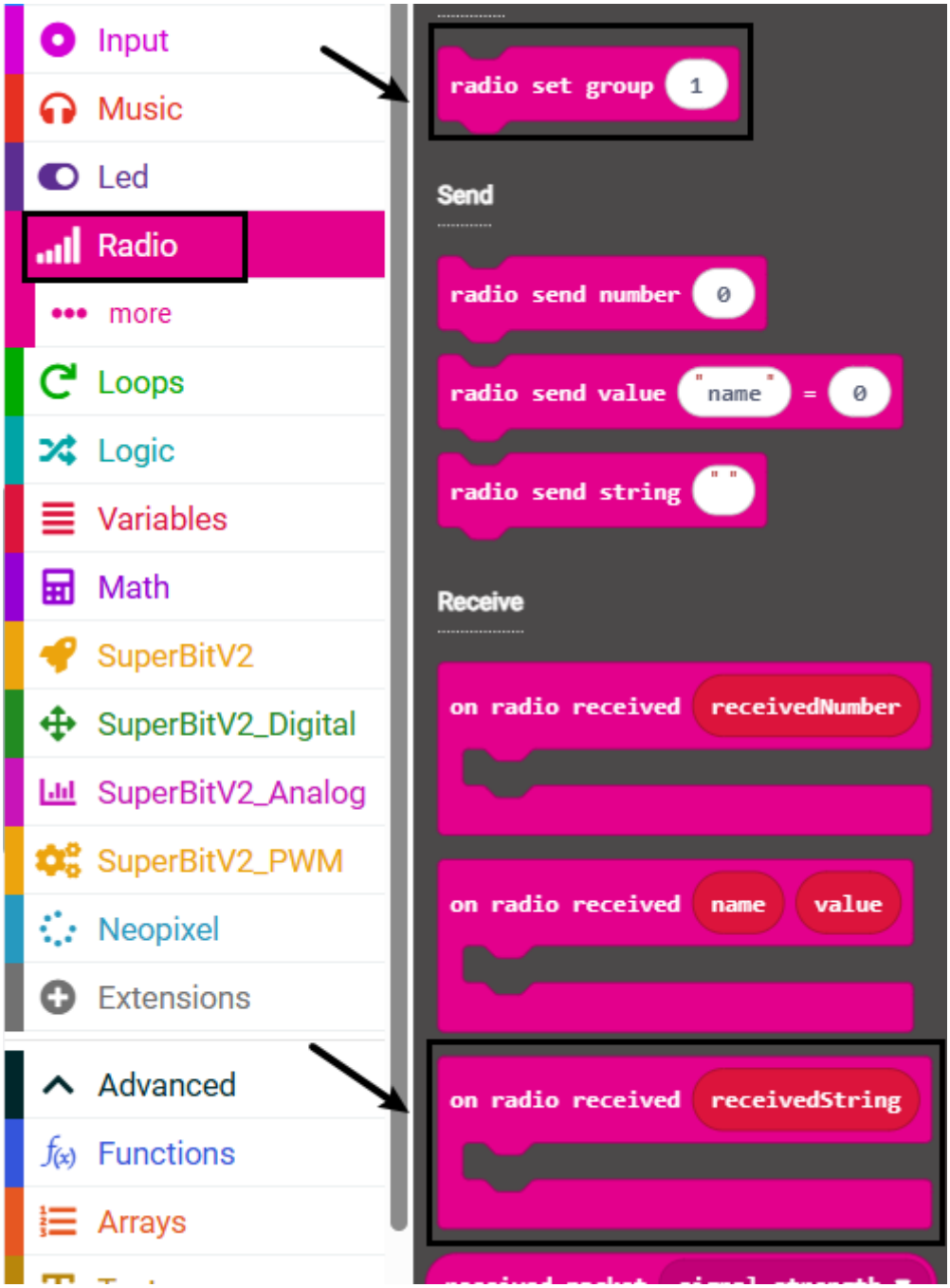

Input
Music
Led
Radio
Loops
Logic
Variables
Math
set item to 0
change item by 1
Your Variables
item
receivedString


Logic
Variables
Math
GHBit
Rocker value Nostate
Button num B1 value Press
Music_Handle dadadum


Basic
Input
Music
Led
Radio
Loops
on start
pause (ms) 100
show arrow North


Search...
Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
SuperBitV2
SuperBitV2_Digital
SuperBitV2
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

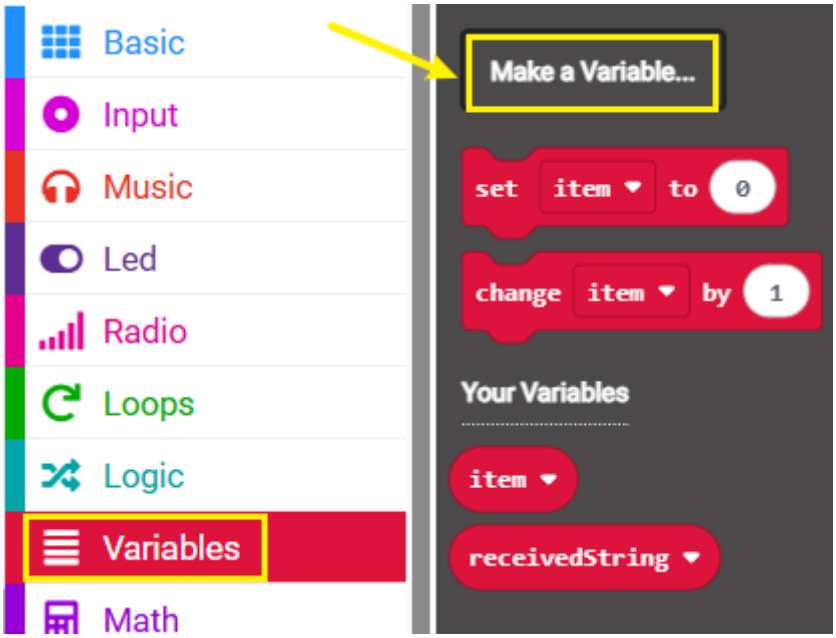

SuperBitV2
SuperBitV2_Digital
SuperBitV2_Analog
SuperBitV2_PWM
Neopixel
more
Extensions
strip show rainbow from 1 to 360
strip show color red
strip show bar graph of 0 up to 255
strip show
strip clear


Input
Music
Led
Radio
more
Loops
Logic
Variables
Math
SuperBitV2
SuperBitV2_Digital
SuperBitV2_Analog
SuperBitV2_PWM
Neopixel
Extensions
Advanced
Functions
Arrays
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

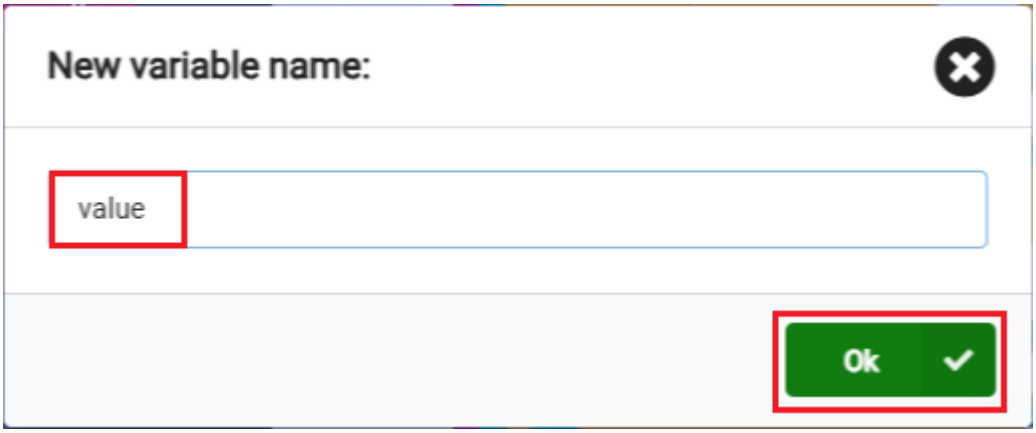

Search...
Basic
Input
Music
Led
Radio
more
more
radio set transmit power 7
radio set transmit serial number true
radio set frequency band 0
radio raise event
from source MICROBIT_ID_BUTTON_A

4.3 Add new variables

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

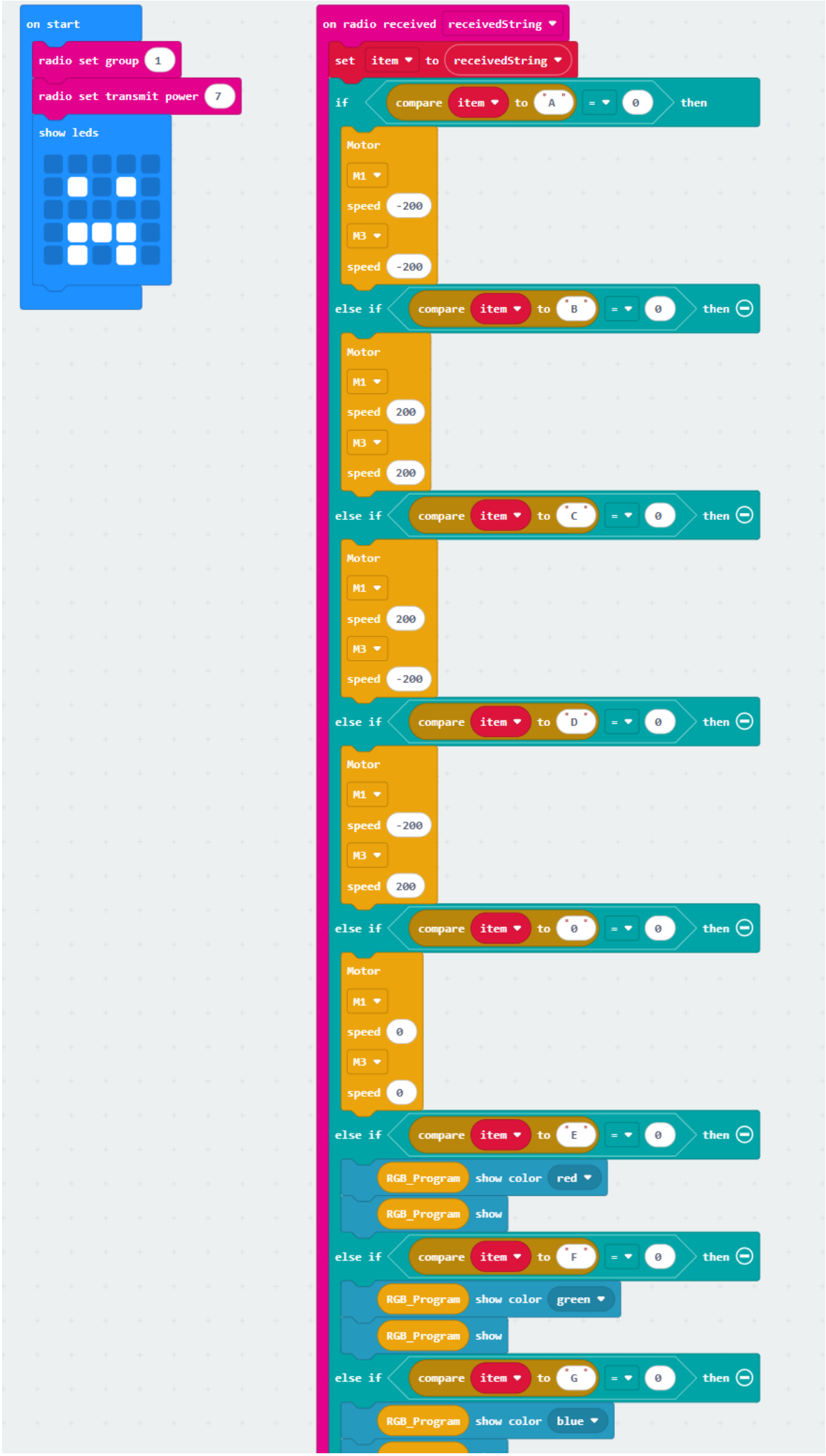


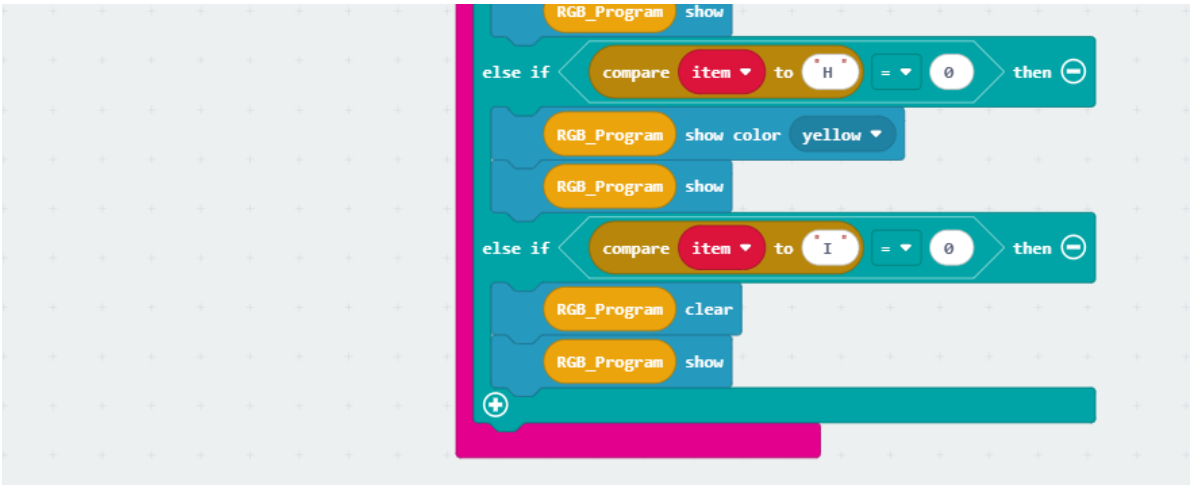
② Enter the variable name to complete the new variable.



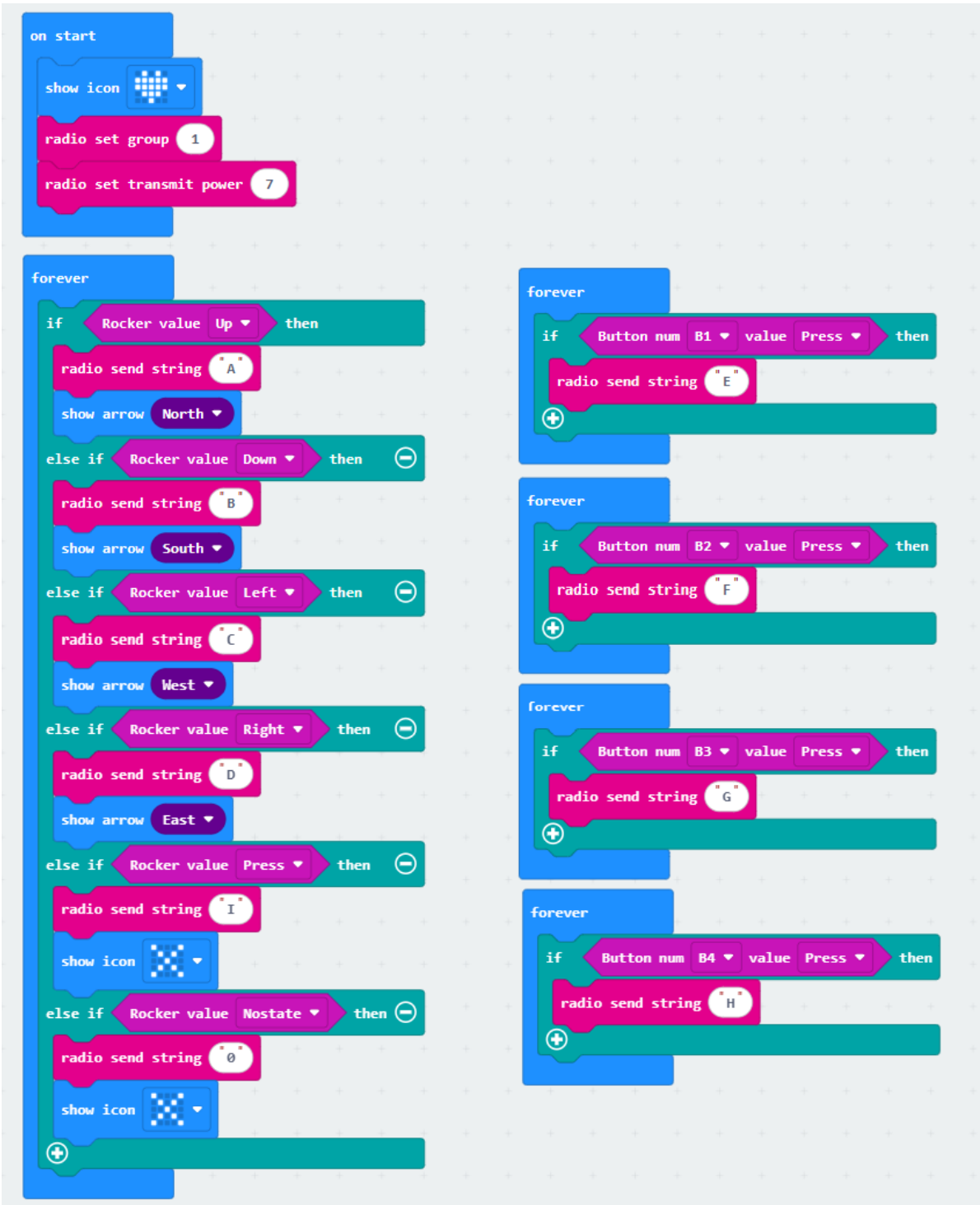
4.4 Combined blocks

The summary program of microbit on **Spider** is shown in the figure below

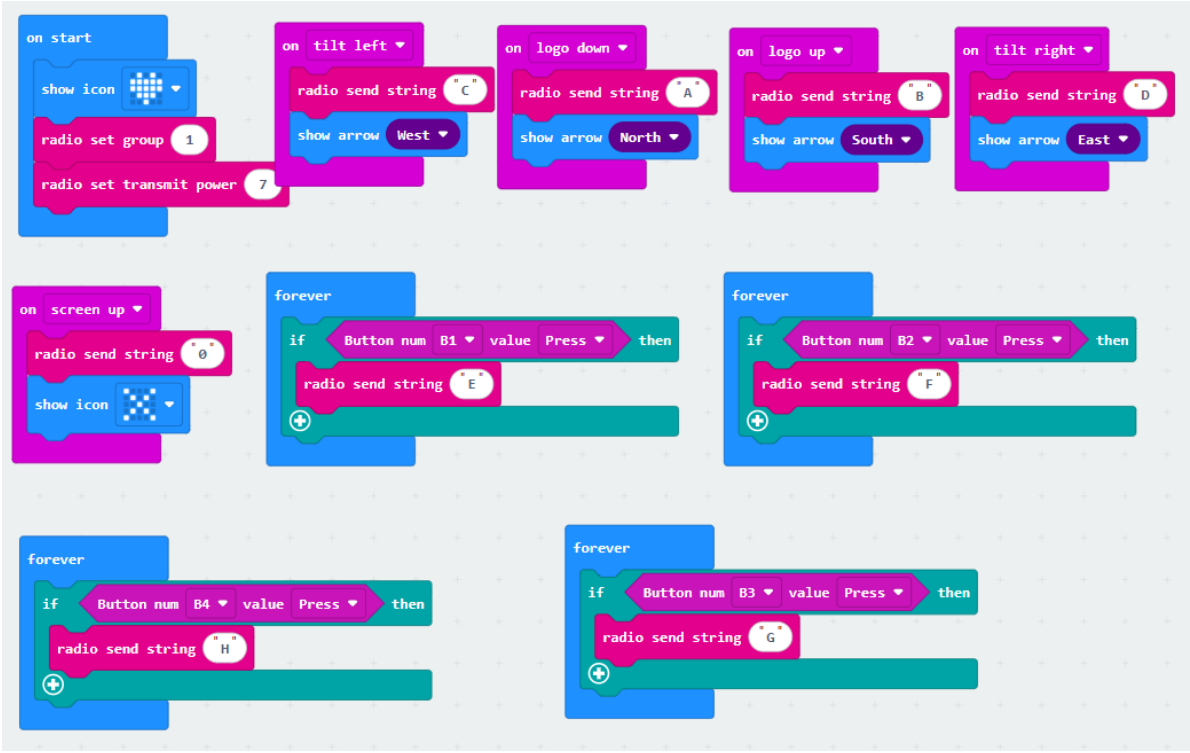

on start
radio set group 1
radio set transmit power 7
show leds
on radio received receivedString
set item to receivedString
if compare item to "A" = 0 then
Motor
M1
speed -200
M3
speed -200
else if compare item to "B" = 0 then
Motor
M1
speed 200
M3
speed 200
else if compare item to "C" = 0 then
Motor
M1
speed 200
M3
speed -200
else if compare item to "D" = 0 then
Motor
M1
speed -200
M3
speed 200
else if compare item to "0" = 0 then
Motor
M1
speed 0
M3
speed 0
else if compare item to "E" = 0 then
RGB_Program show color red
RGB_Program show
else if compare item to "F" = 0 then
RGB_Program show color green
RGB_Program show
else if compare item to "G" = 0 then
RGB_Program show color blue



Handle rocker control program is as follows



Handle gravity control program is as follows



You can also directly open the **microbit-handle-control-Spider.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

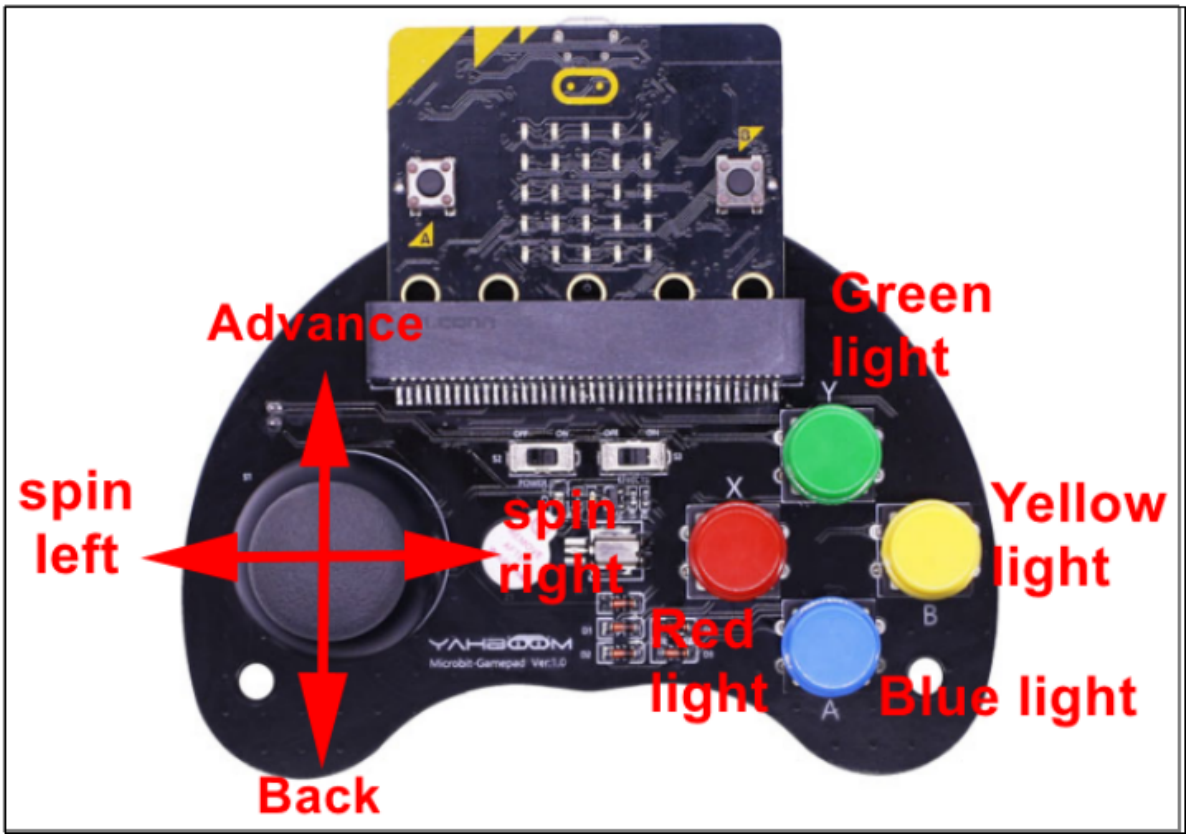
5. Experimental phenomenon

We need to download the Spider program to the Spider's micro:bit motherboard, turn on the Spider's power switch, and we can see a weird face pattern on the micro:bit dot matrix;

Download the handle remote control program to the handle's micro:bit motherboard, turn on the handle's power switch, and we can see that the micro:bit dot matrix will be initialized to display a heart pattern, 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 Spider.

The handle functions are as follows.



! Note: In the case of Handle-rocker-control, press the joystick to control the RGB light to go out. This function does not exist in the case of Handle-gravity-control.