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 variables
    4.4 Assemble building blocks
5. Experimental phenomenon

1. Learning objectives

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

2. Building blocks

For the building blocks steps, please refer to the installation drawings of **[Assembly Course]--[Skip car]** in the materials or the building blocks installation book.

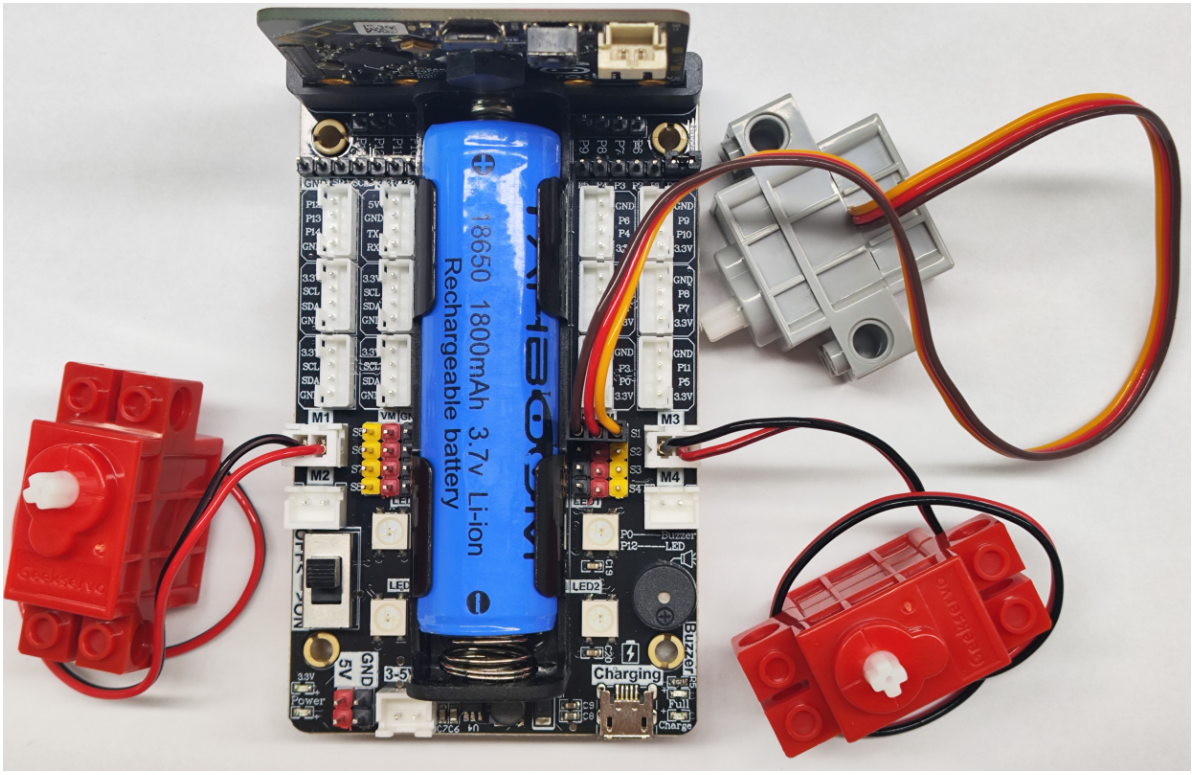
3. Motor wiring

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

Insert the motor wiring on the right side of the car into the M3 interface of the Super:bit expansion board, and the black wire is close to the battery side;

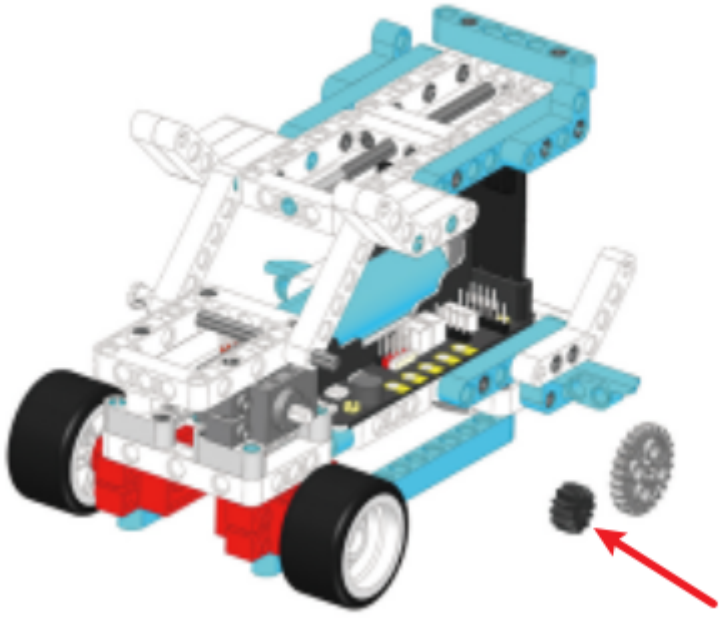
Insert the building blocks servo wiring into the S1 interface of the Super:bit expansion board, and the orange servo wiring is inserted into the yellow pin of S1.

As shown in the figure below:



! Notes:

When taking the course related to the building block servo for the first time, we need to remove the gear 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 angle of the car loading platform to be parallel to the ground, and then install the servo gear. (If you have used Skip car and servo-related programs before, you can skip this step)



4. Programming

Method 1 Online Programming:

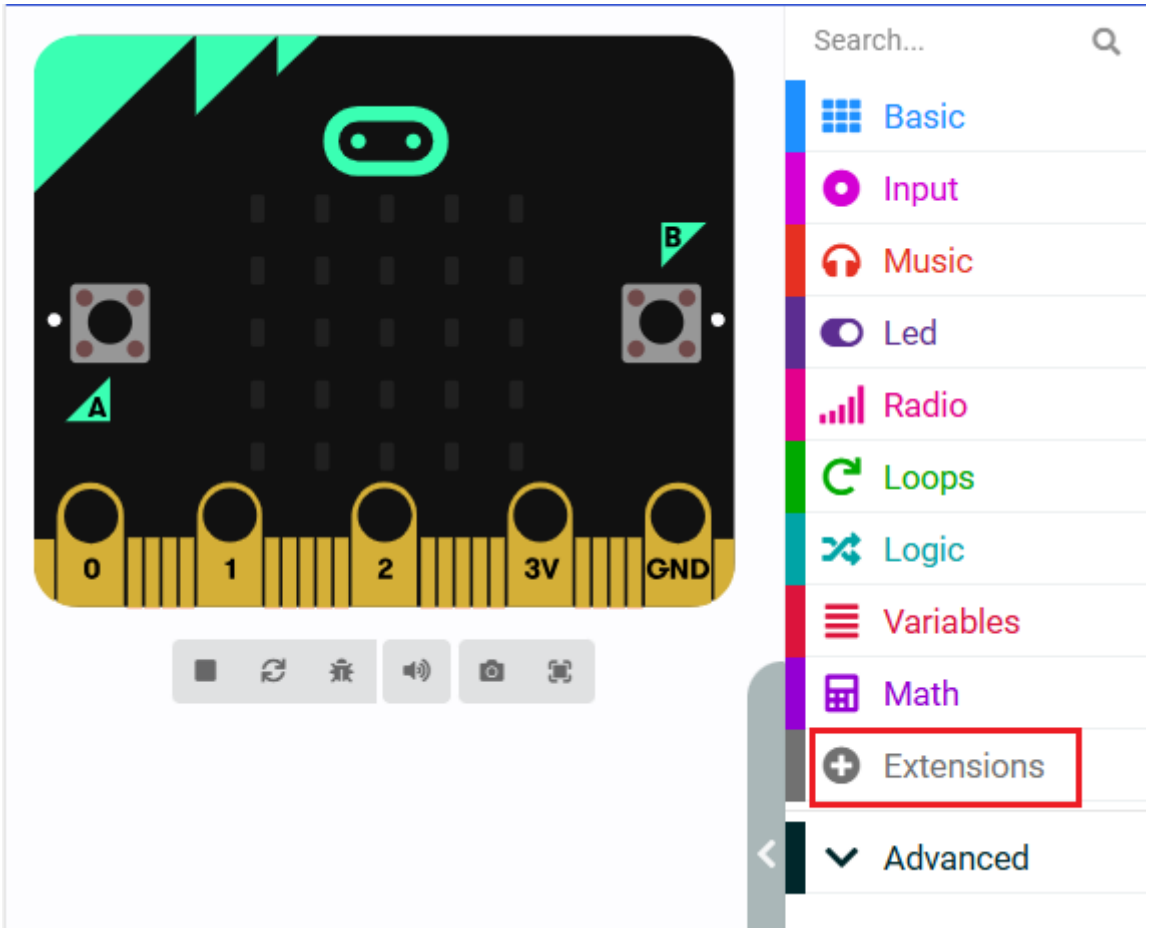
First, connect the micro:bit to the computer via a USB data cable, and a U disk will pop up on the computer. Click the URL in the U disk: https://makecode.microbit.org/ to enter the programming interface. Then, add the Yahboom software package to program.

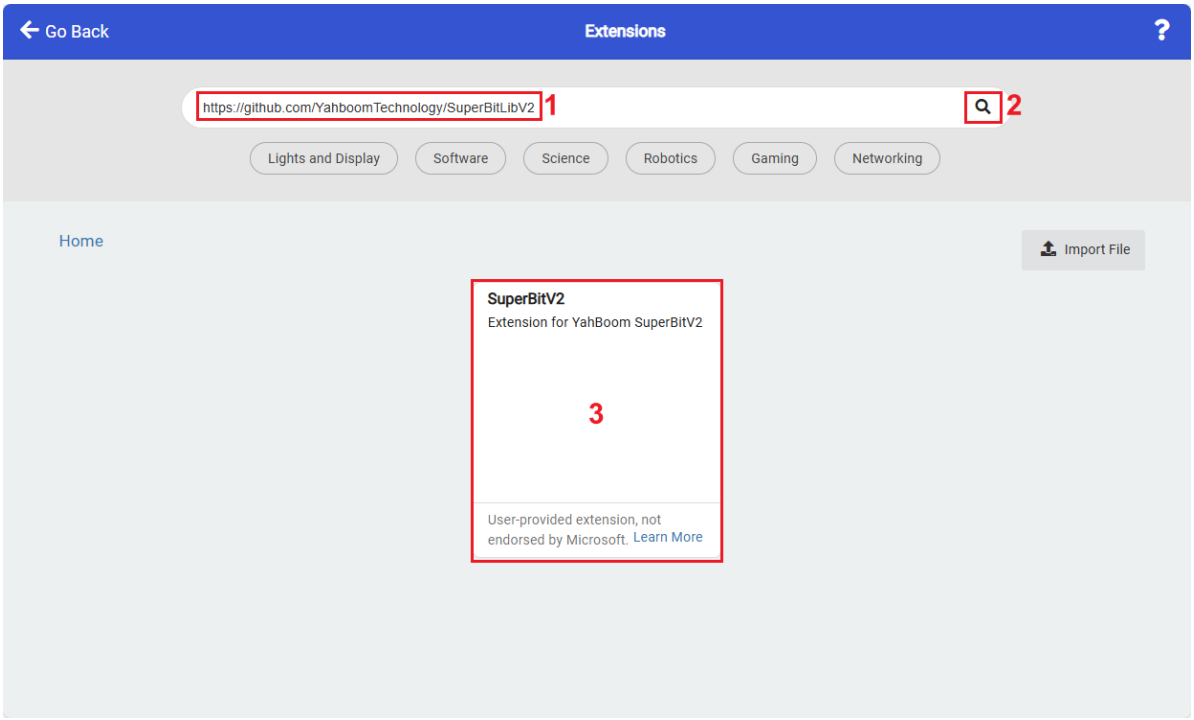
Method 2 Offline Programming:

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

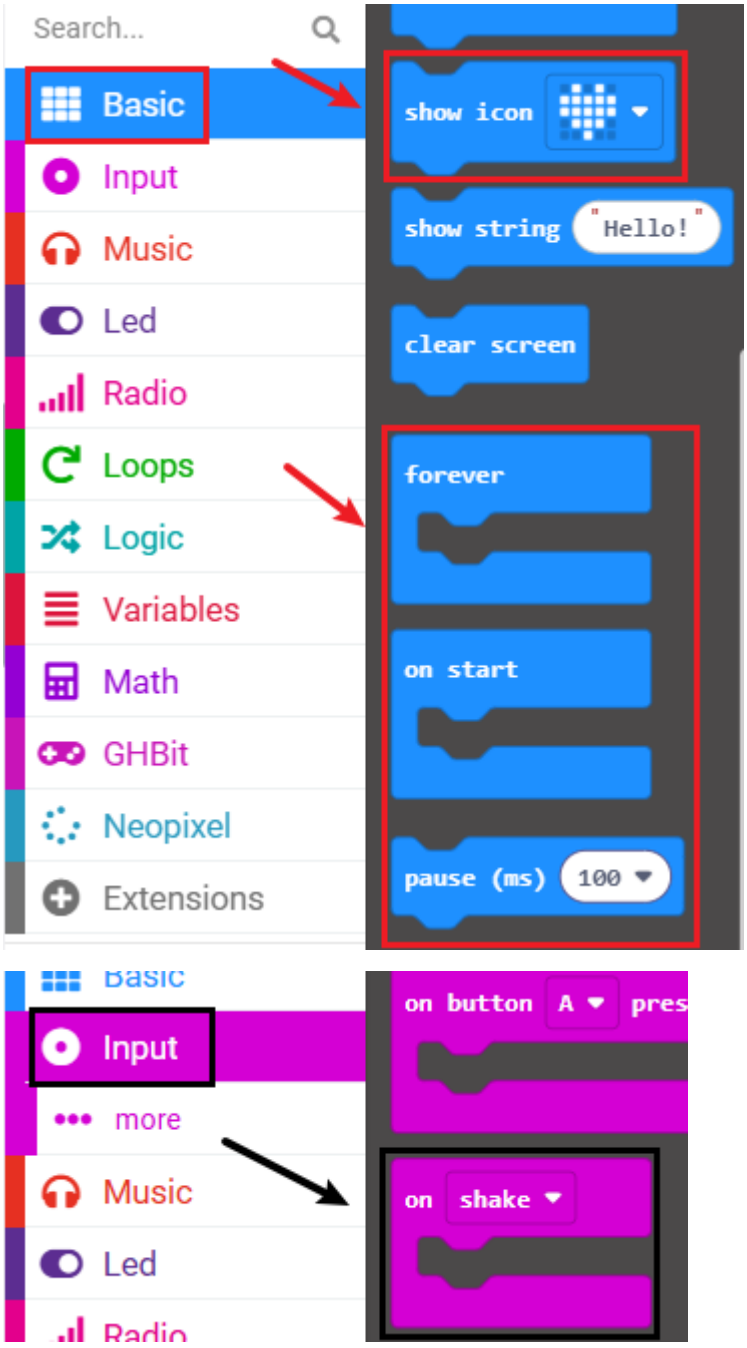
4.1 Add expansion package





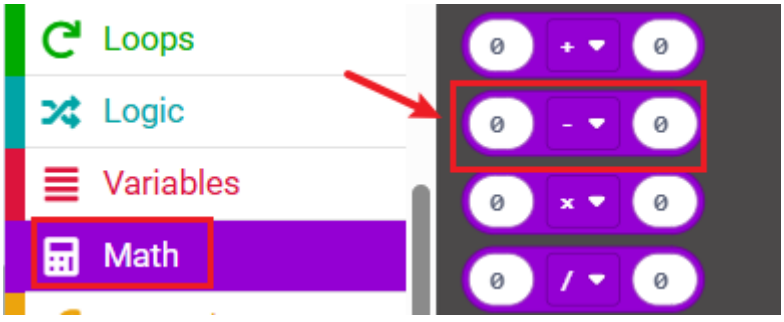
4.2 Building blocks used

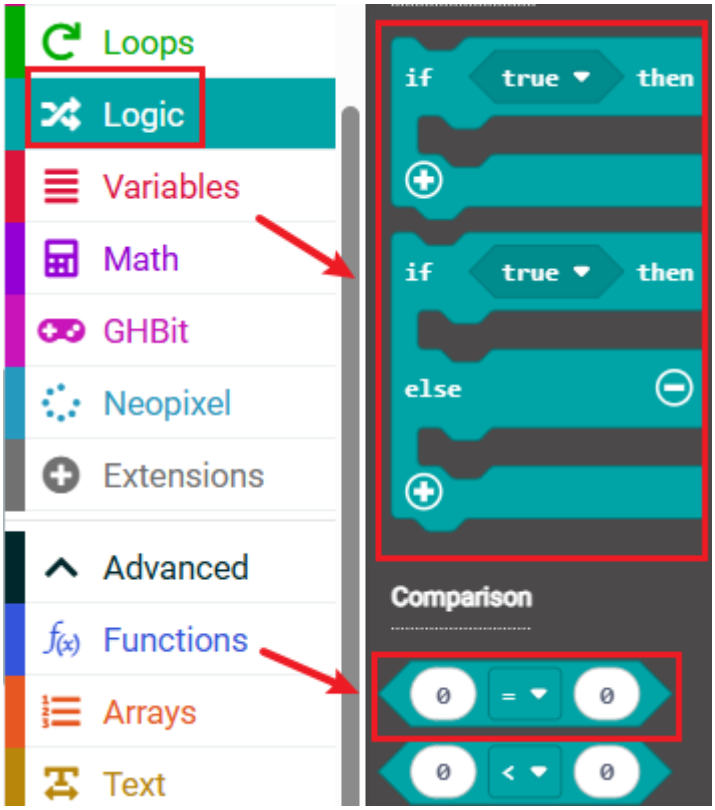
The location of the building blocks required for this programming is shown in the figure below.

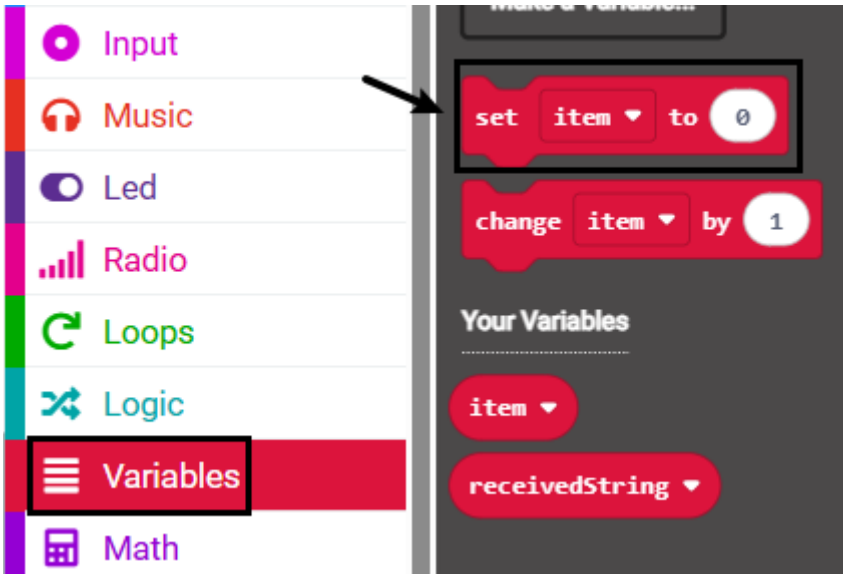


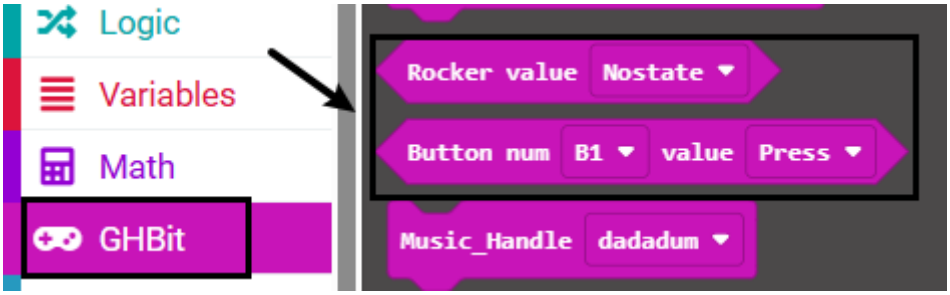
Led
Radio
Loops
Logic
Variables
Math
GHBit
Neopixel
Extensions
Advanced
Functions
Arrays
Text
Game
Images
Text
length of "Hello"
join "Hello" "World"
parse to number "123"
split "this" at ""
"this" includes ""
"this" find index of ""
"this" is empty
substring of "this" from 0 of length 10
compare "this"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

Bluetooth
Loops
Logic
Variables
Math
do
for index from 0 to 4
do


Loops
Logic
Variables
Math
0 + 0
0 - 0
0 x 0
0 / 0

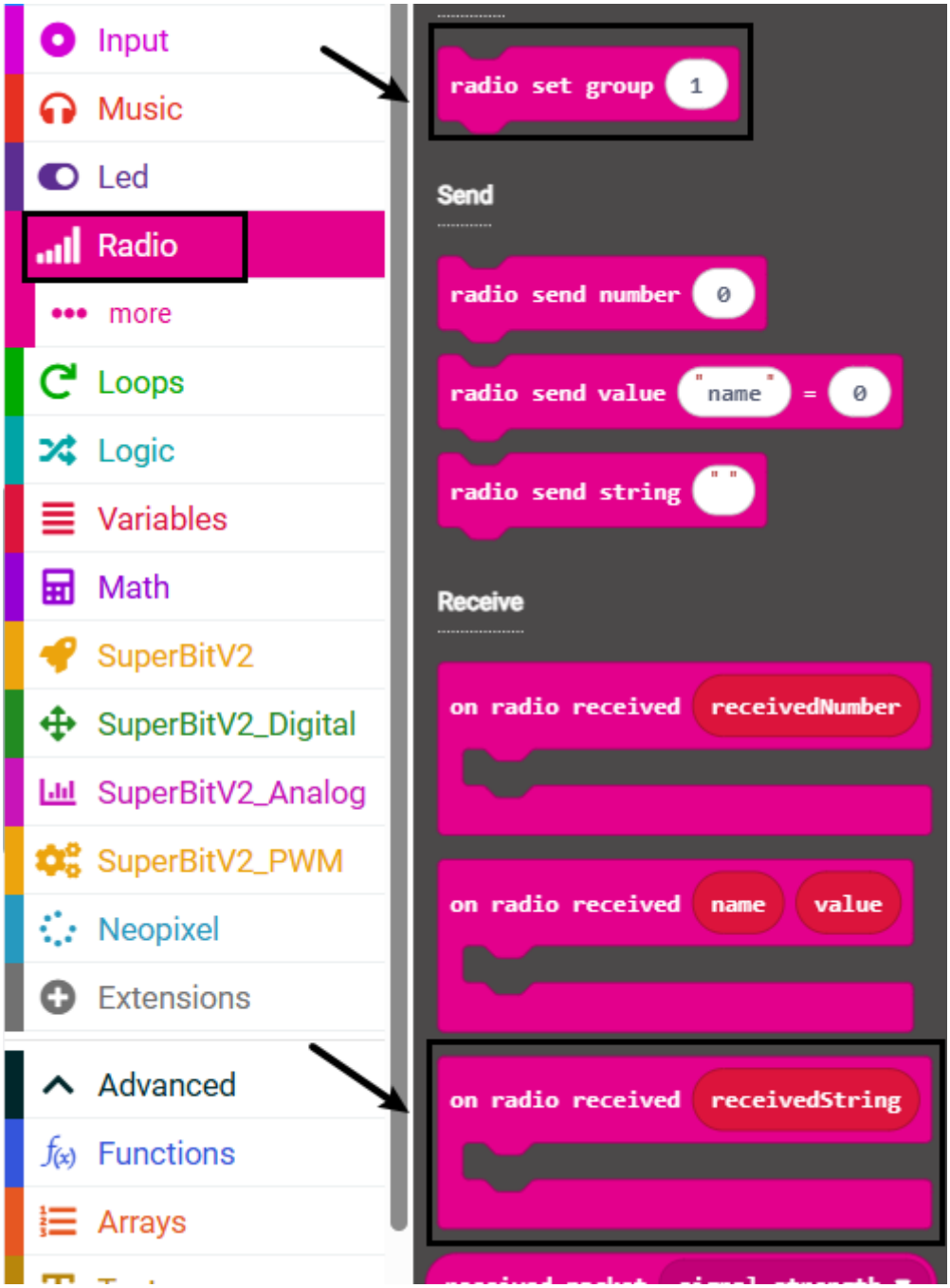

Loops
Logic
Variables
Math
GHBit
Neopixel
Extensions
Advanced
Functions
Arrays
Text
if true then
if true then
else
Comparison
0 = 0
0 < 0


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

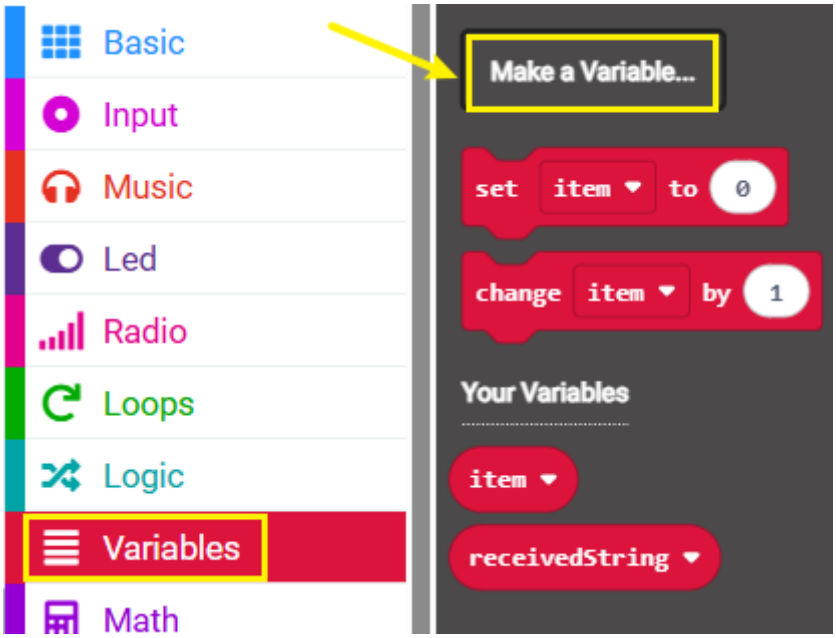
Search...
Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
SuperBitV2
SuperBitV2_Digital
SuperBitV2
SuperBitV2_Digital
SuperBitV2_Analog
SuperBitV2_PWM
Neopixel
more
Extensions
SuperBitV2
RGB_Program
Music dadadum
Servo(180°) num S1 value 0
Servo(360°_rotatable) num S1 pos forward value 0
Servo(360°)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

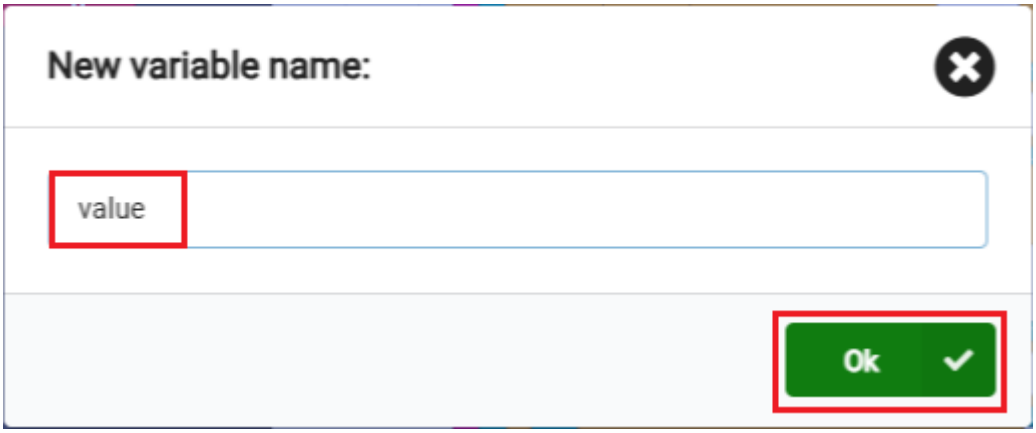
Search...
Basic
Input
Music
Led
Radio
more
more
radio set transmit power 7
radio set transmit serial number true
radio set frequency band 0
radio raise event
from source MICROBIT_ID_BUTTON_A

4.3 Add variables

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

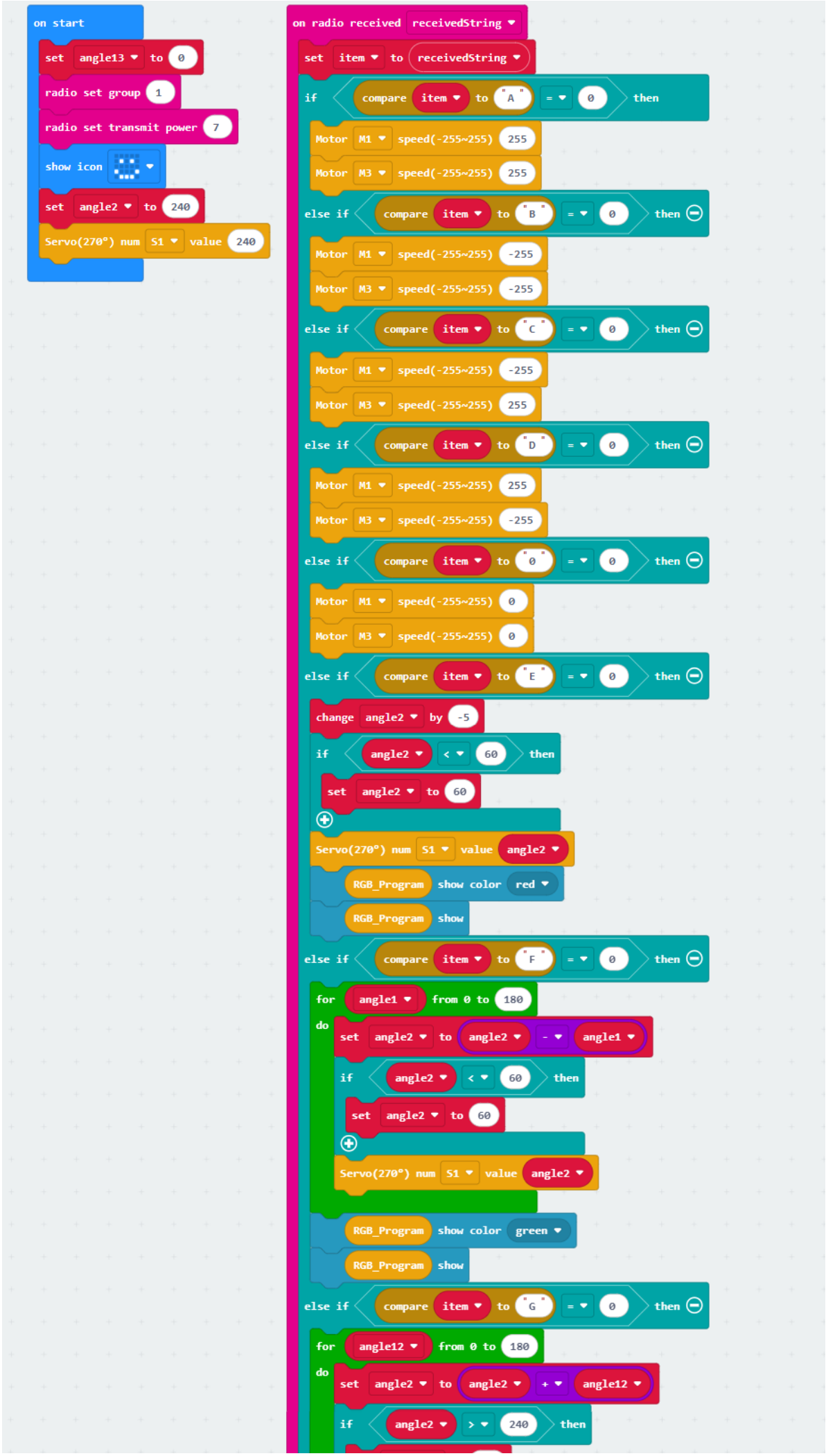


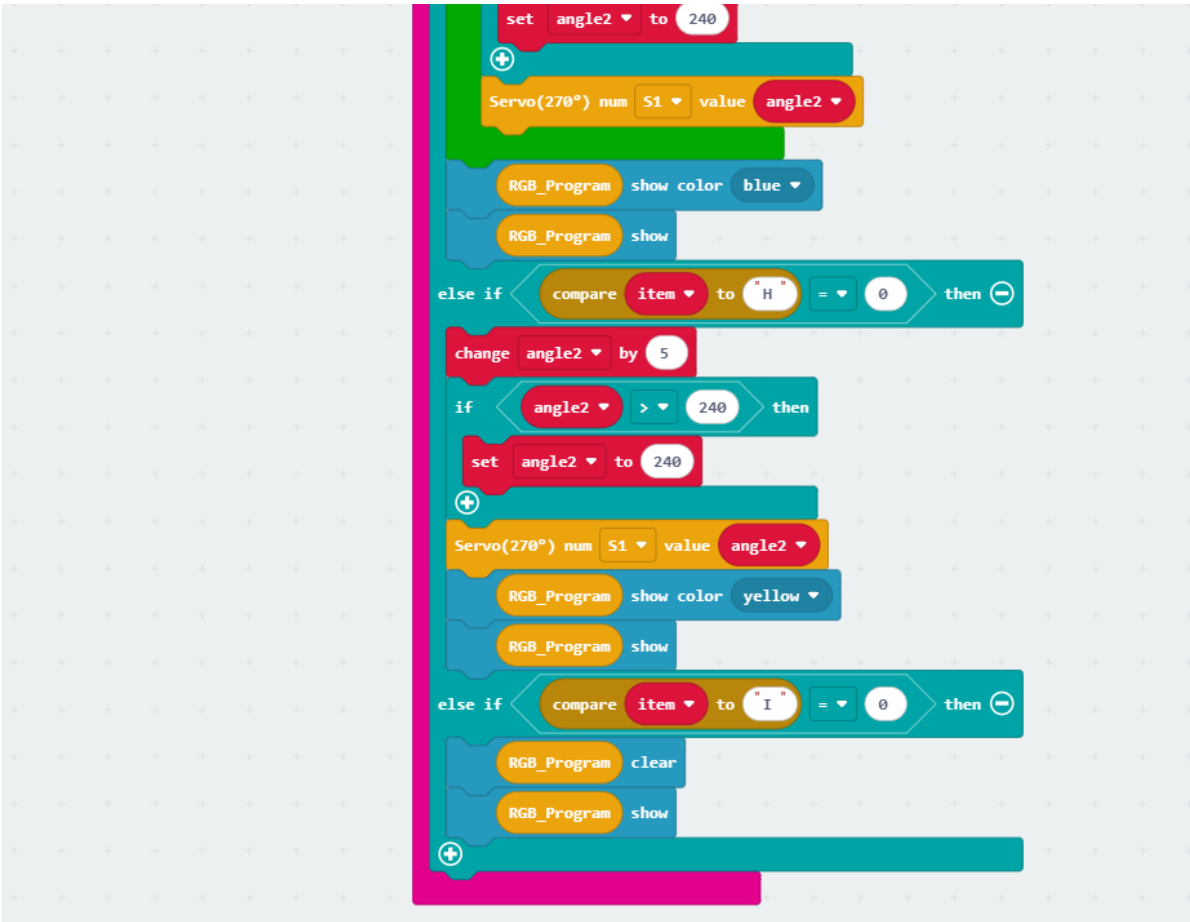
② Enter the variable name to complete the new variable.



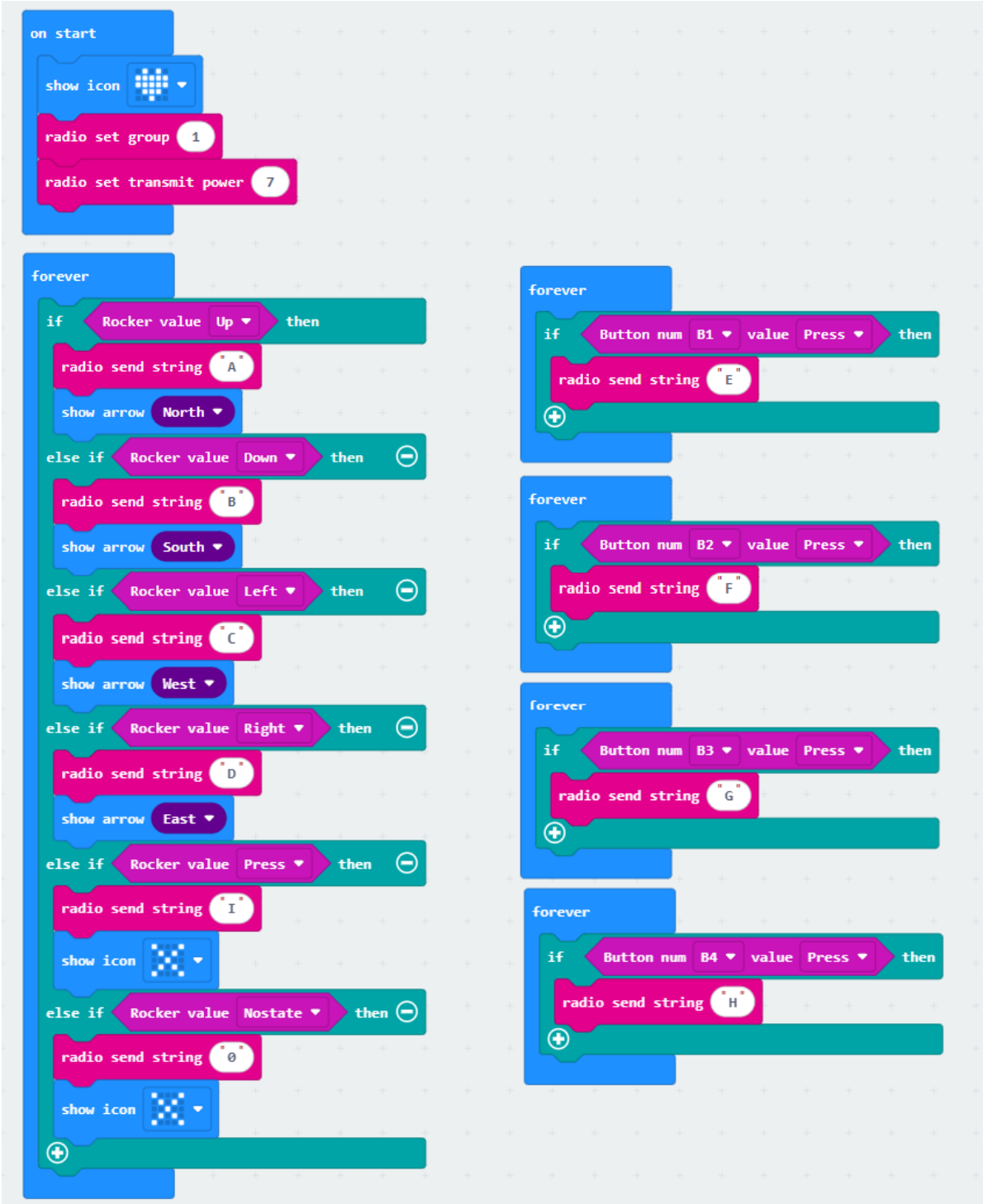
4.4 Assemble building blocks

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

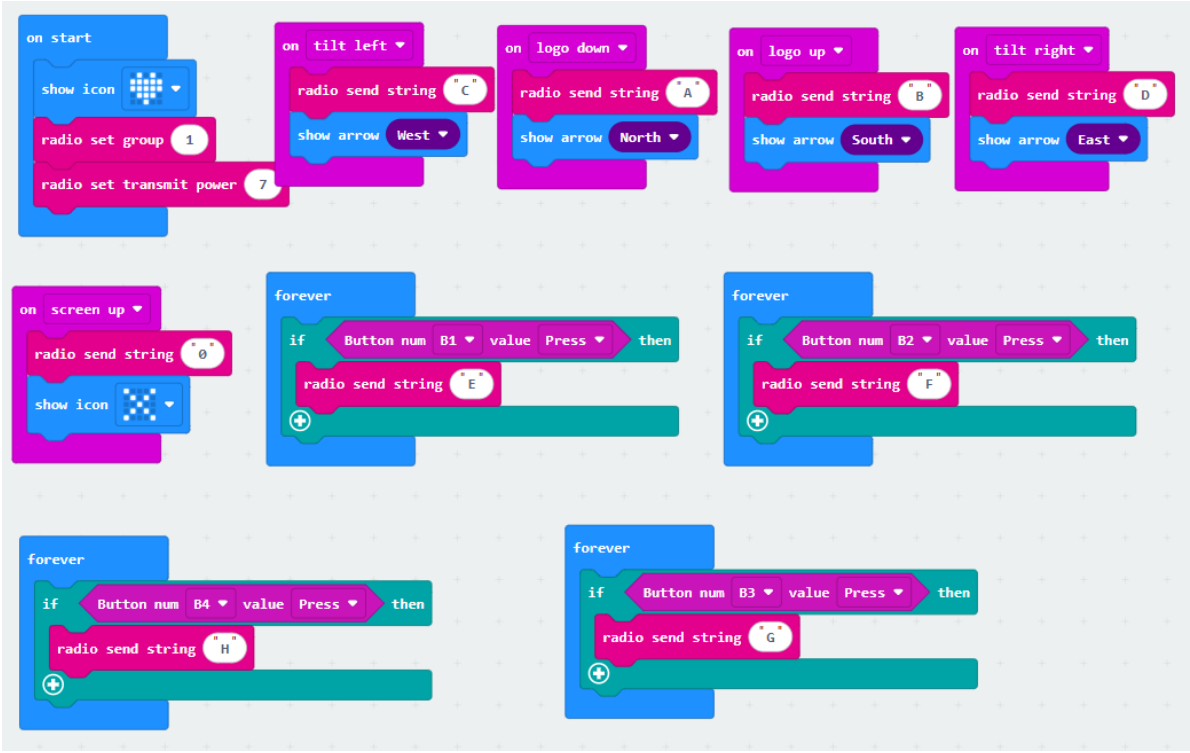

on start
set angle13 to 0
radio set group 1
radio set transmit power 7
show icon
set angle2 to 240
Servo(270°) num S1 value 240
on radio received receivedString
set item to receivedString
if compare item to "A" = 0 then
Motor M1 speed(-255~255) 255
Motor M3 speed(-255~255) 255
else if compare item to "B" = 0 then
Motor M1 speed(-255~255) -255
Motor M3 speed(-255~255) -255
else if compare item to "C" = 0 then
Motor M1 speed(-255~255) -255
Motor M3 speed(-255~255) 255
else if compare item to "D" = 0 then
Motor M1 speed(-255~255) 255
Motor M3 speed(-255~255) -255
else if compare item to "0" = 0 then
Motor M1 speed(-255~255) 0
Motor M3 speed(-255~255) 0
else if compare item to "E" = 0 then
change angle2 by -5
if angle2 < 60 then
set angle2 to 60
Servo(270°) num S1 value angle2
RGB_Program show color red
RGB_Program show
else if compare item to "F" = 0 then
for angle1 from 0 to 180
do set angle2 to angle2 - angle1
if angle2 < 60 then
set angle2 to 60
Servo(270°) num S1 value angle2
RGB_Program show color green
RGB_Program show
else if compare item to "G" = 0 then
for angle12 from 0 to 180
do set angle2 to angle2 + angle12
if angle2 > 240 then



Handle rocker control program is as follows



Handle gravity control program is as follows



You can also directly open the **microbit-handle remote control-Skip car.hex, microbit-handle joystick 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

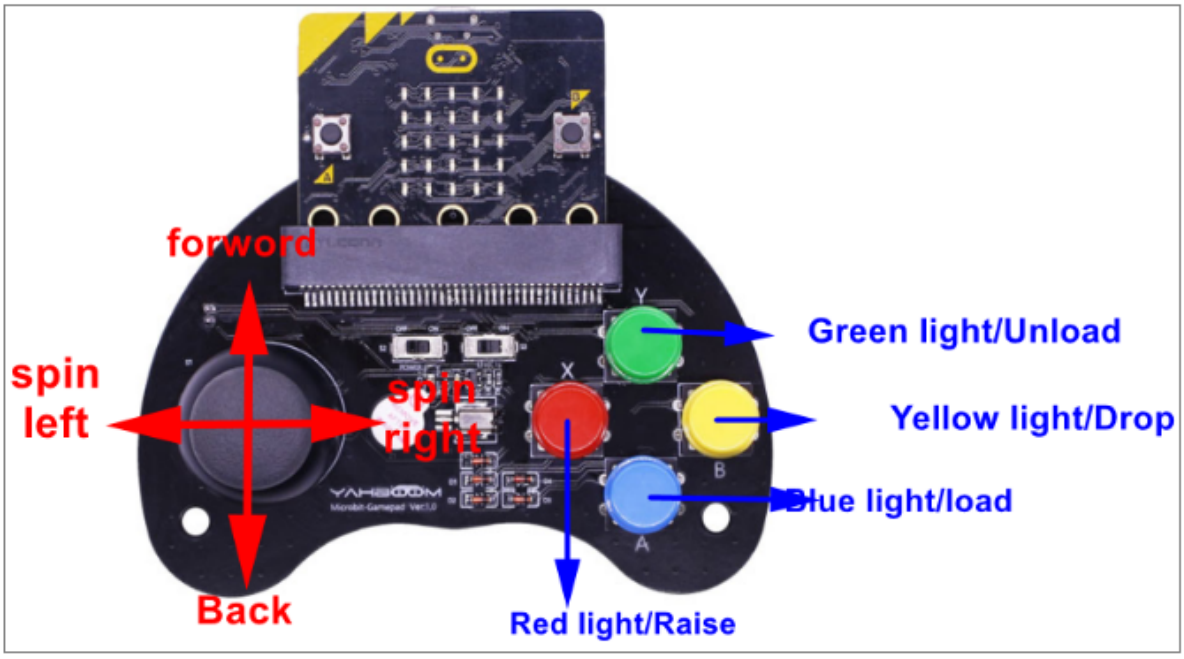
5. Experimental phenomenon

We need to download the Skip car program to the micro:bit motherboard of Skip car, open Skip car's power switch, we can see a smiley face pattern on the micro:bit dot matrix;

Download the controller remote control program to the micro:bit mainboard of the controller, turn on the power switch of the controller, we can see that the micro:bit dot matrix will initialize to display a heart pattern, and then an "X" pattern will be displayed, indicating that the controller 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 Skip car.

The controller functions are as follows.



!Note: In the case of controller joystick control, press the joystick to control the RGB light to go out, and this function does not exist in the case of gravity control of the controller.