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 Assemble building blocks
- 5. Experimental phenomenon

1. Learning objectives

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

2. Building blocks

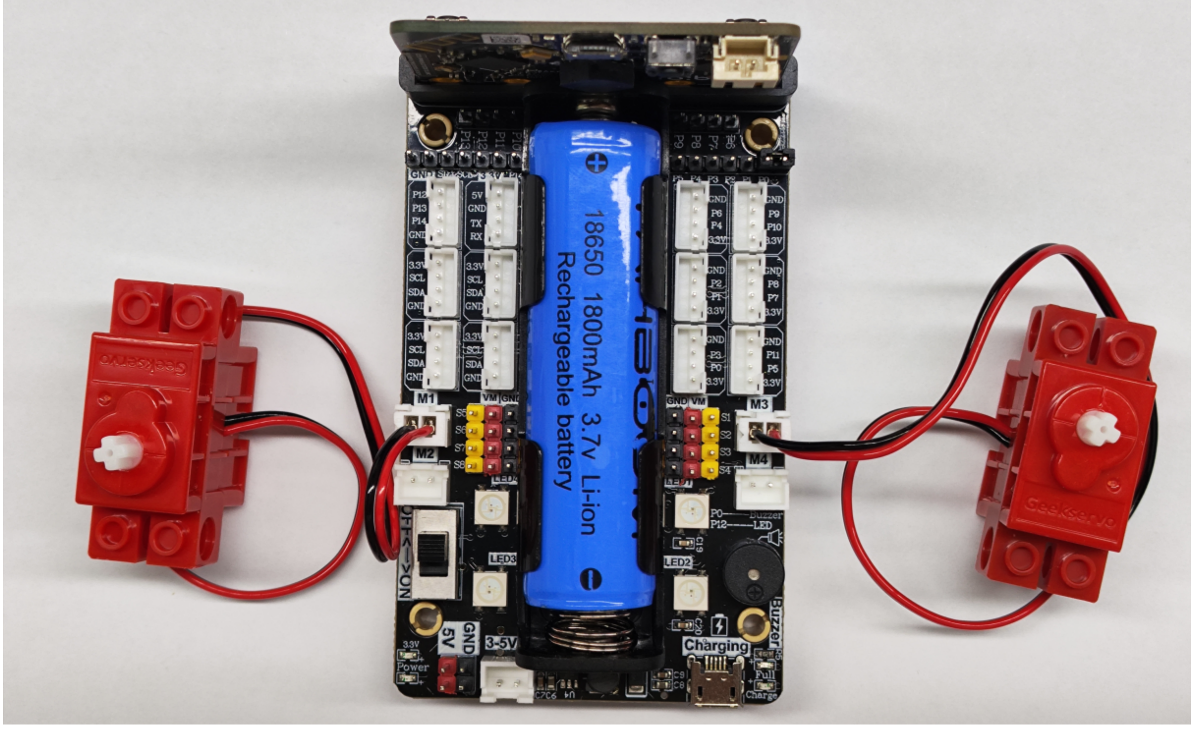
For the building block construction steps, please refer to the installation drawings of [Assembly Course]-[Pretty car] in the materials or the building block installation album.

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:

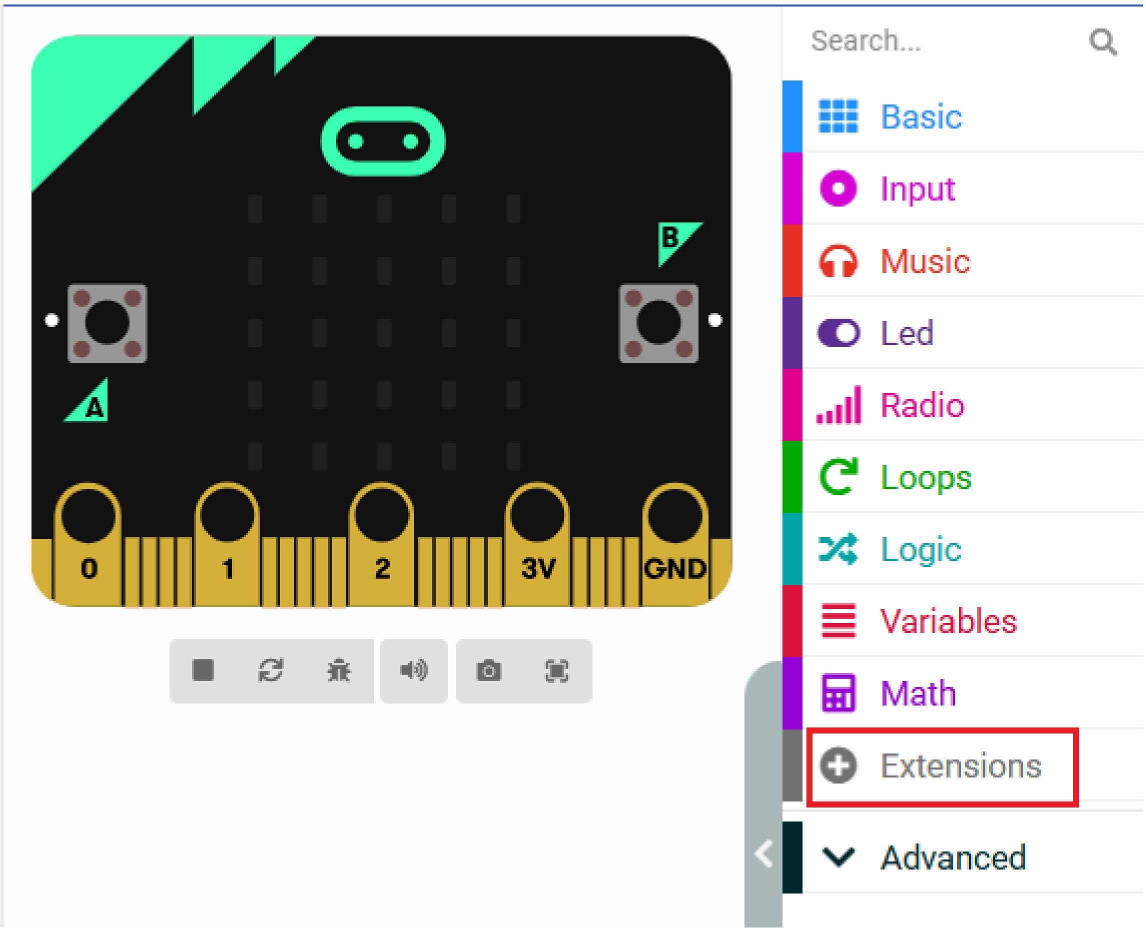
First, connect micro:bit to the computer via a USB data cable. The computer will pop up a U disk. Click the URL in the U disk: https://makecode.microbit.org/ to enter the programming interface. Then, add the Yabo smart software package to start programming.

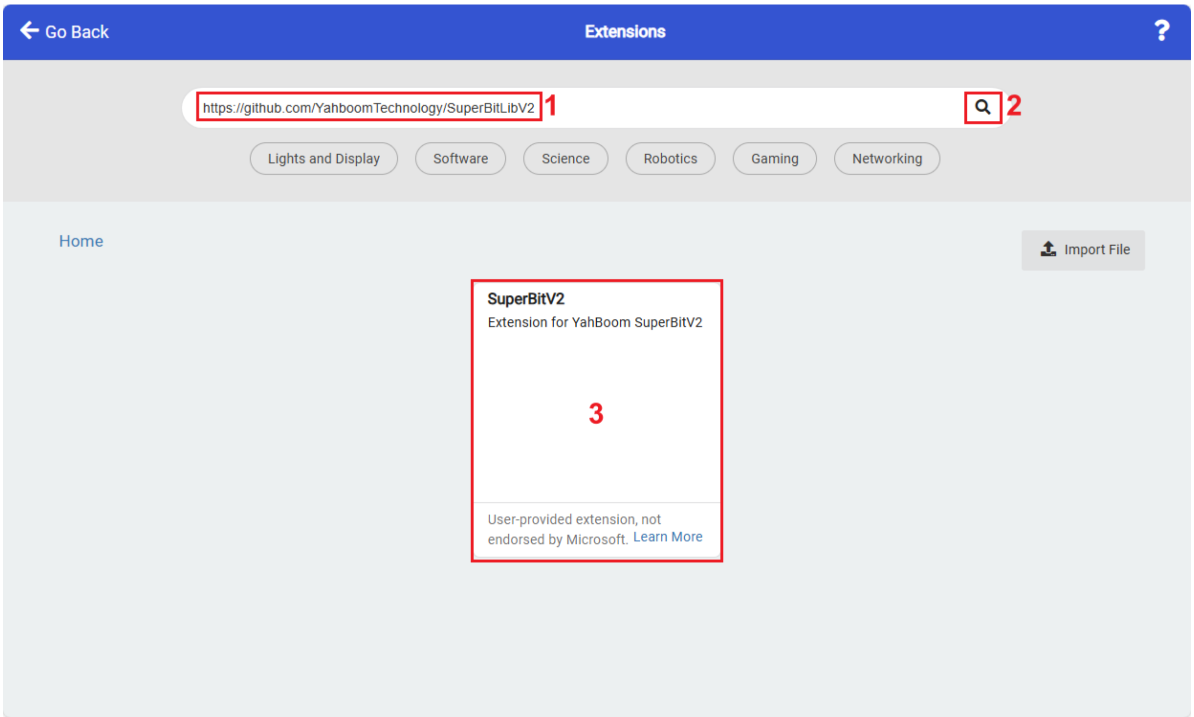
Method 2 Offline programming:

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

superbit kit expansion package: https://github.com/YahboomTechnology/SuperBitLibV2
handle expansion package: https://github.com/YahboomTechnology/GHBitLib

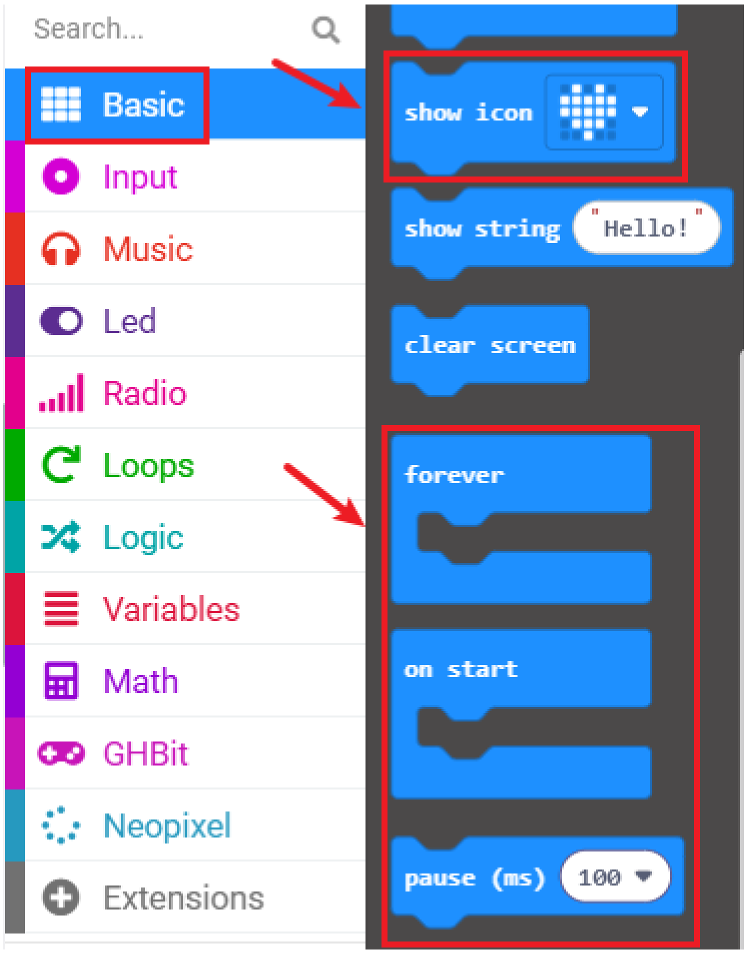
4.1 Add expansion package

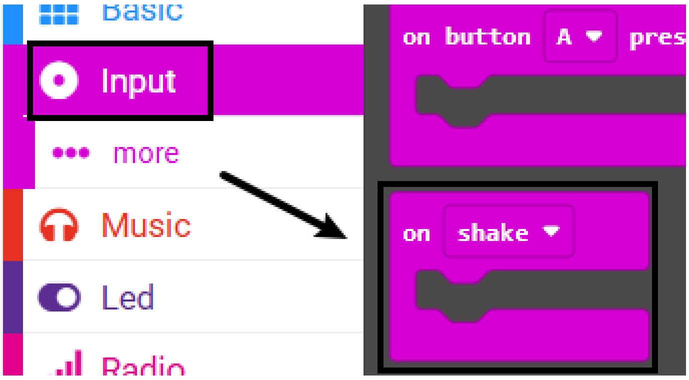


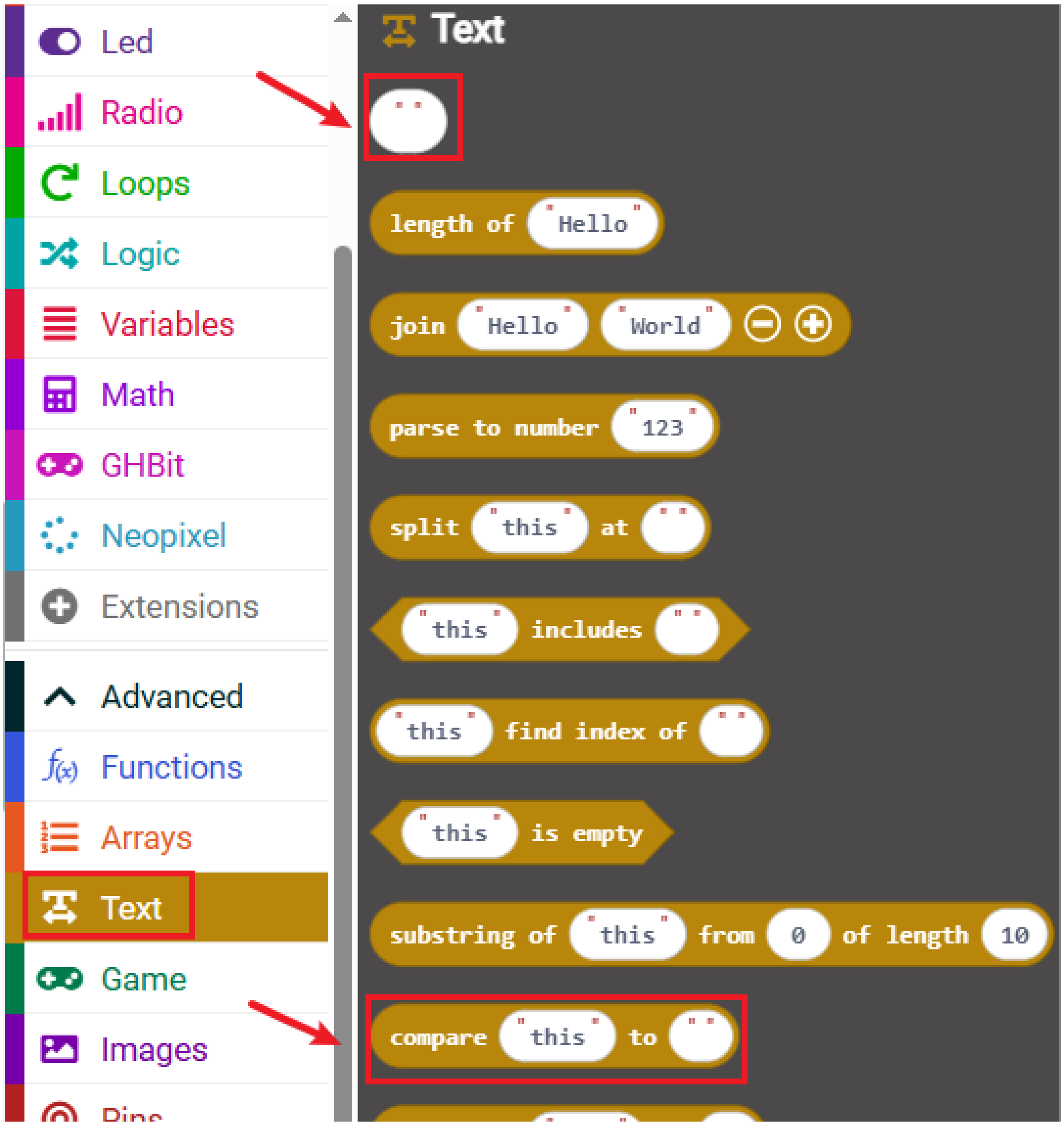

Go Back
Extensions
?
https://github.com/YahboomTechnology/SuperBitLibV2
1
2
Lights and Display
Software
Science
Robotics
Gaming
Networking
Home
Import File
SuperBitV2
Extension for YahBoom SuperBitV2
3
User-provided extension, not endorsed by Microsoft. Learn More

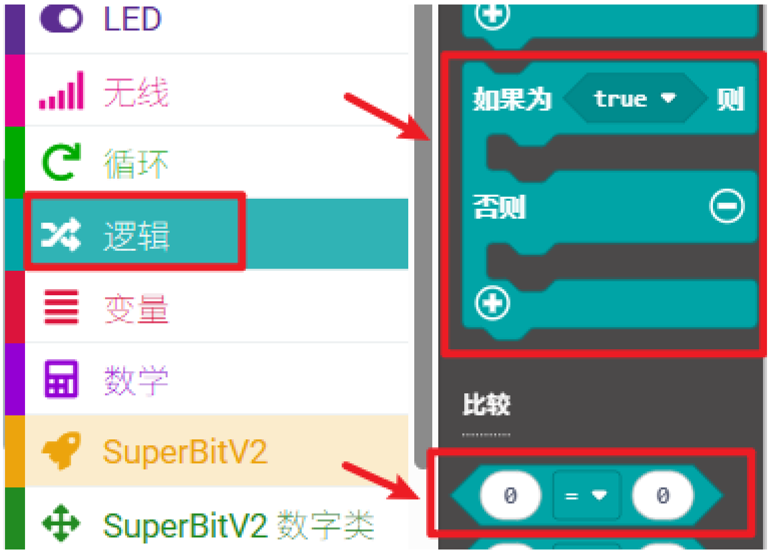
4.2 Blocks used

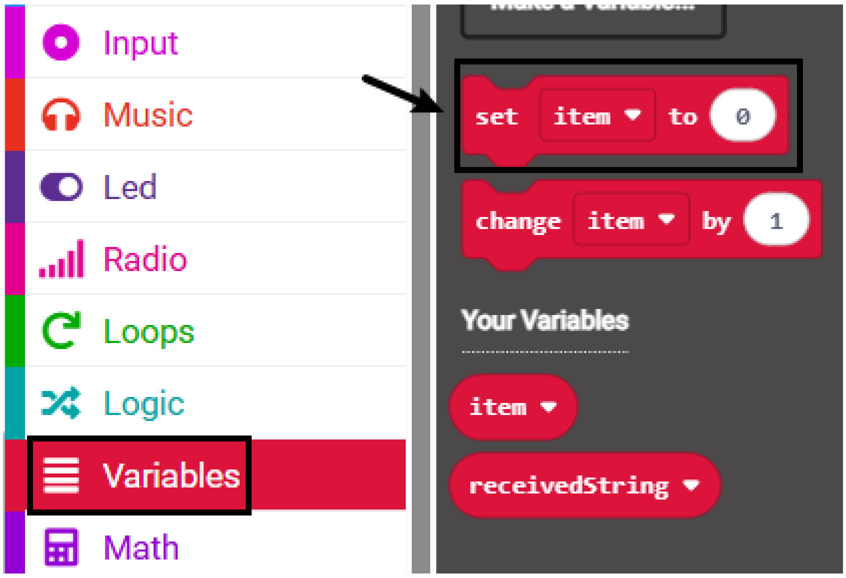
The location of the blocks required for this programming is shown in the figure below.

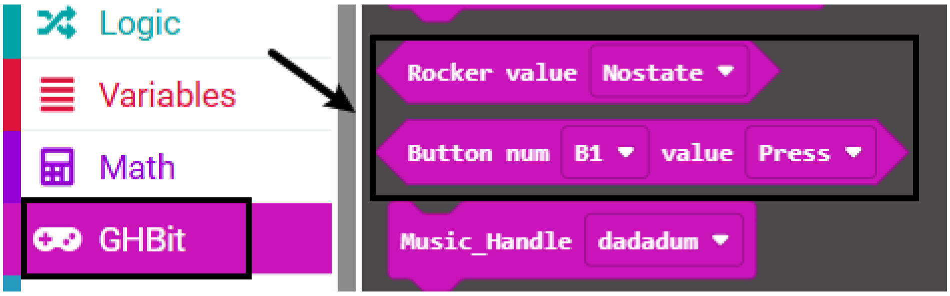

Search...
Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
GHBit
Neopixel
Extensions
show icon
show string "Hello!"
clear screen
forever
on start
pause (ms) 100

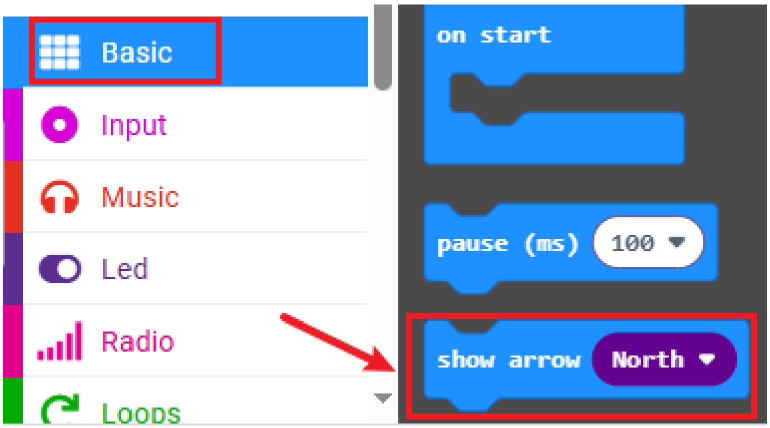

Basic
Input
more
Music
Led
Radio
on button A pres
on shake

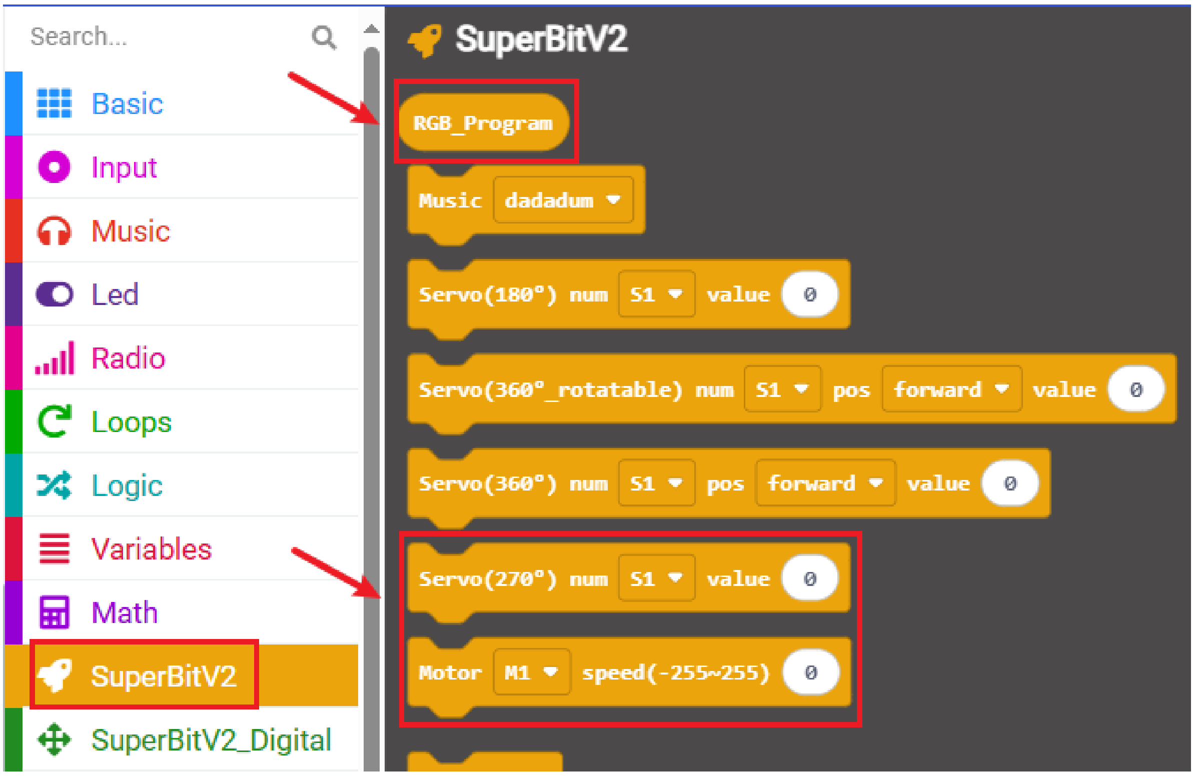

Led
Radio
Loops
Logic
Variables
Math
GHBit
Neopixel
Extensions
Advanced
Functions
Arrays
Text
Game
Images
Pins
Text
" "
length of "Hello"
join "Hello" "World"
parse to number "123"
split "this" at " "
"this" includes " "
"this" find index of " "
"this" is empty
substring of "this" from 0 of length 10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

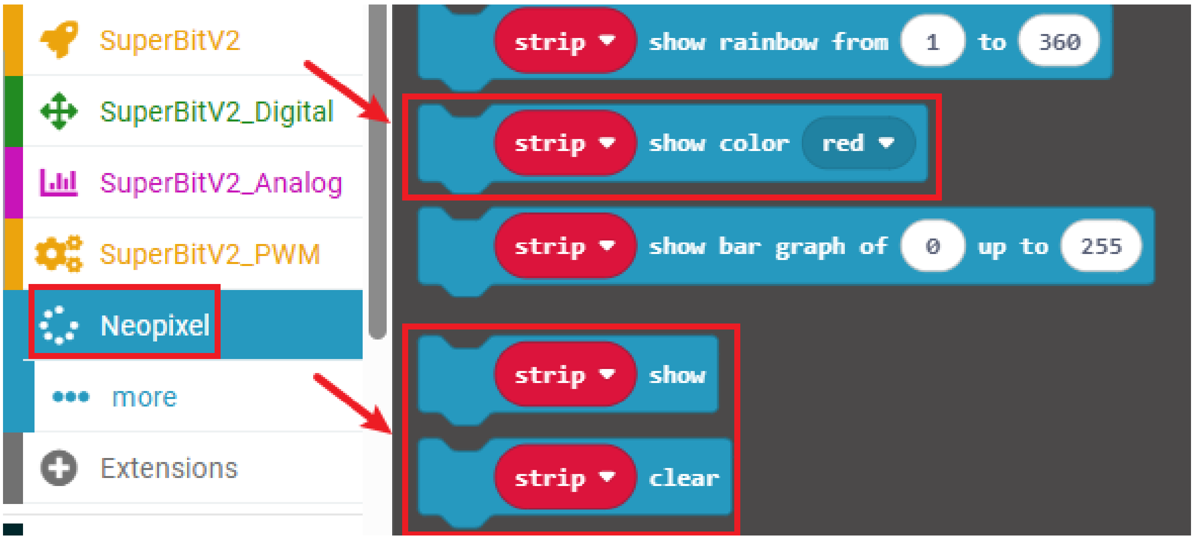

LED
无线
循环
逻辑
变量
数学
SuperBitV2
SuperBitV2 数字类
如果为 true 则
否则
比较
0 = 0

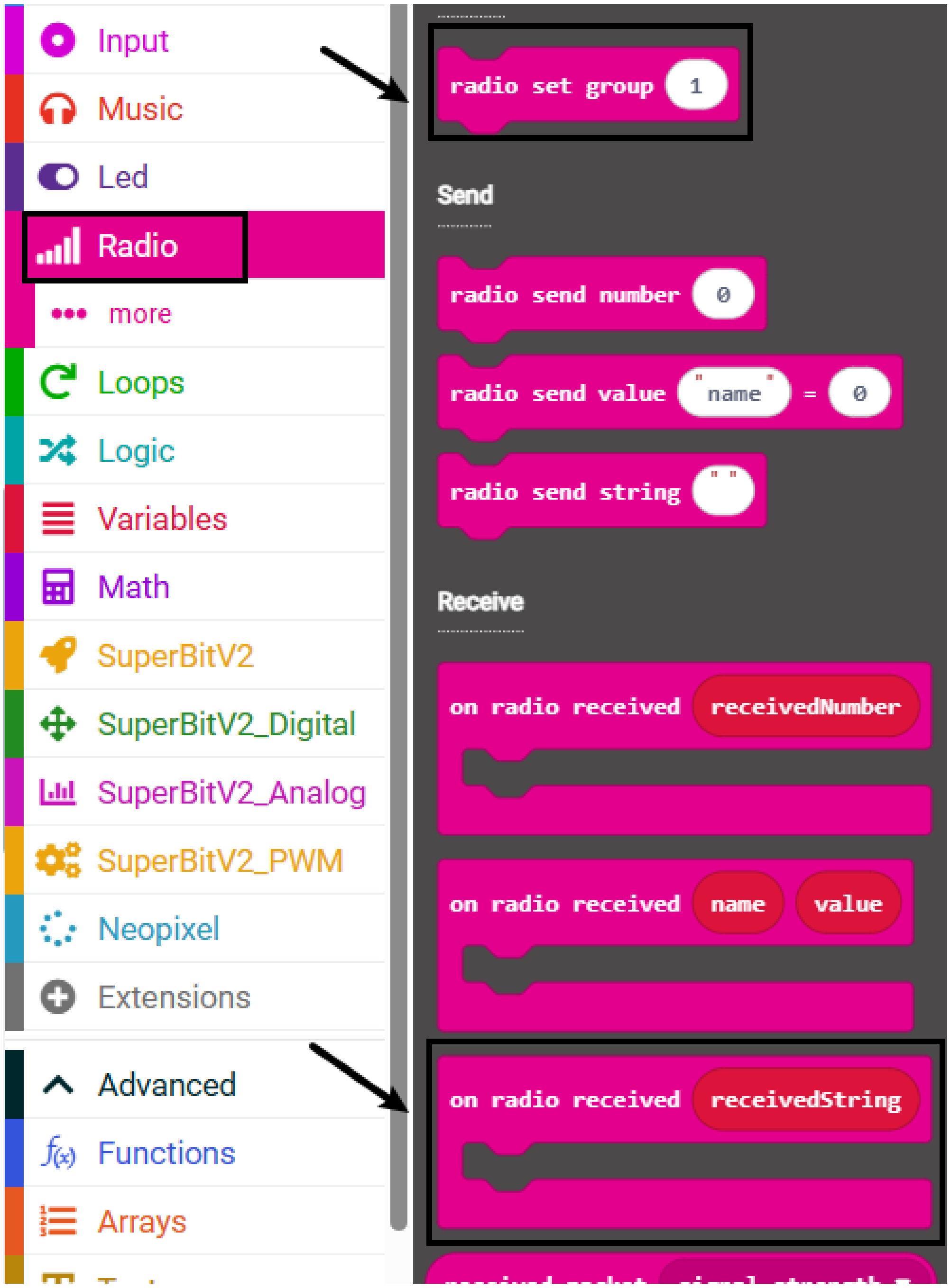

Input
Music
Led
Radio
Loops
Logic
Variables
Math
set item to 0
change item by 1
Your Variables
item
receivedString


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

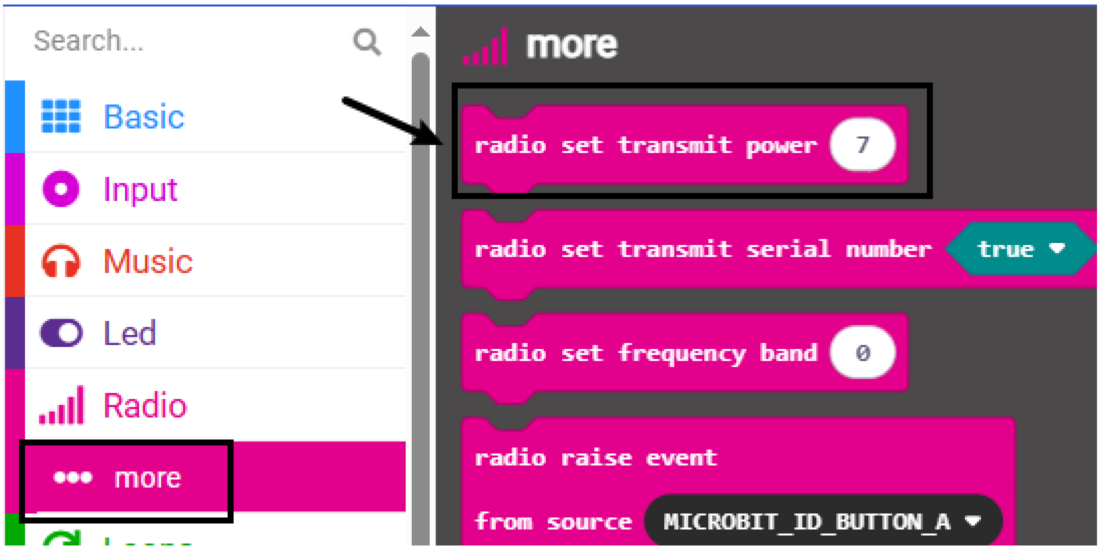

Basic
Input
Music
Led
Radio
Loops
on start
pause (ms) 100
show arrow North


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



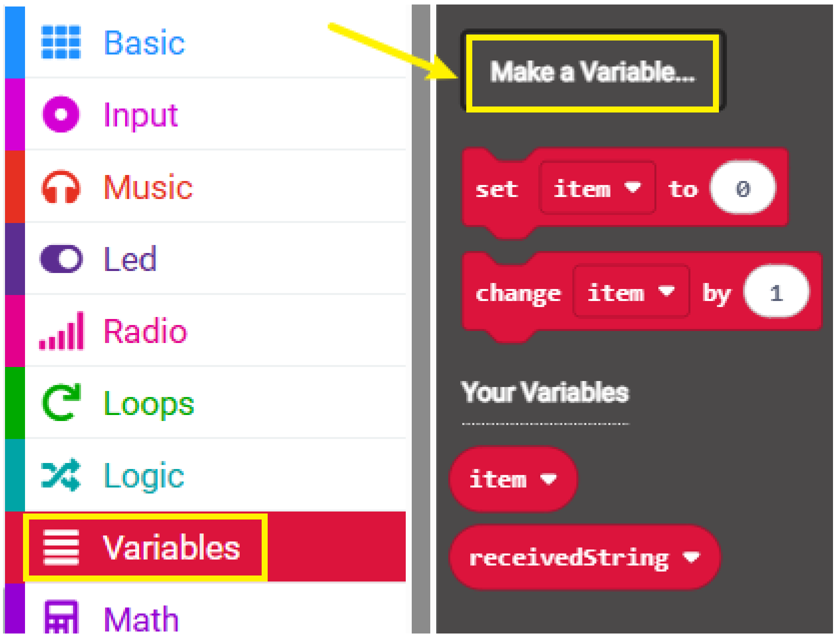
```
radio set group 1

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

on radio received receivedNumber
on radio received name value
on radio received receivedString
```



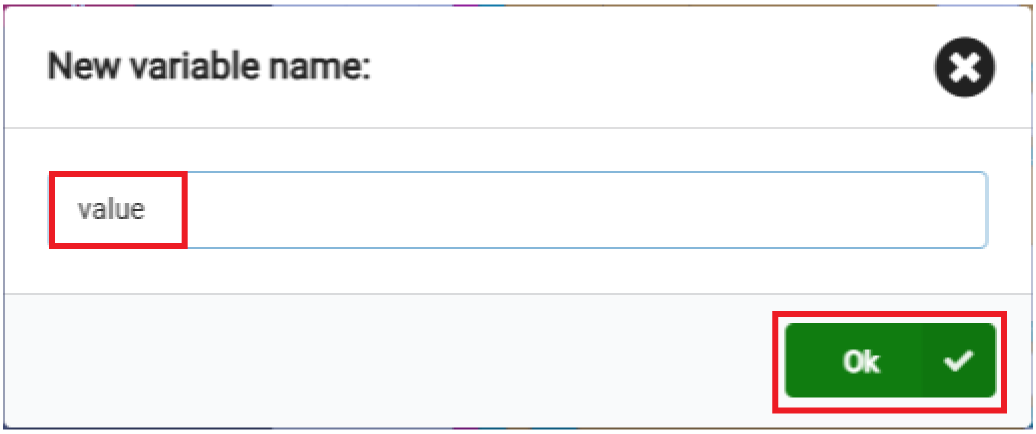
```
radio set transmit power 7
radio set transmit serial number true
radio set frequency band 0
radio raise event
from source MICROBIT_ID_BUTTON_A
```

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.

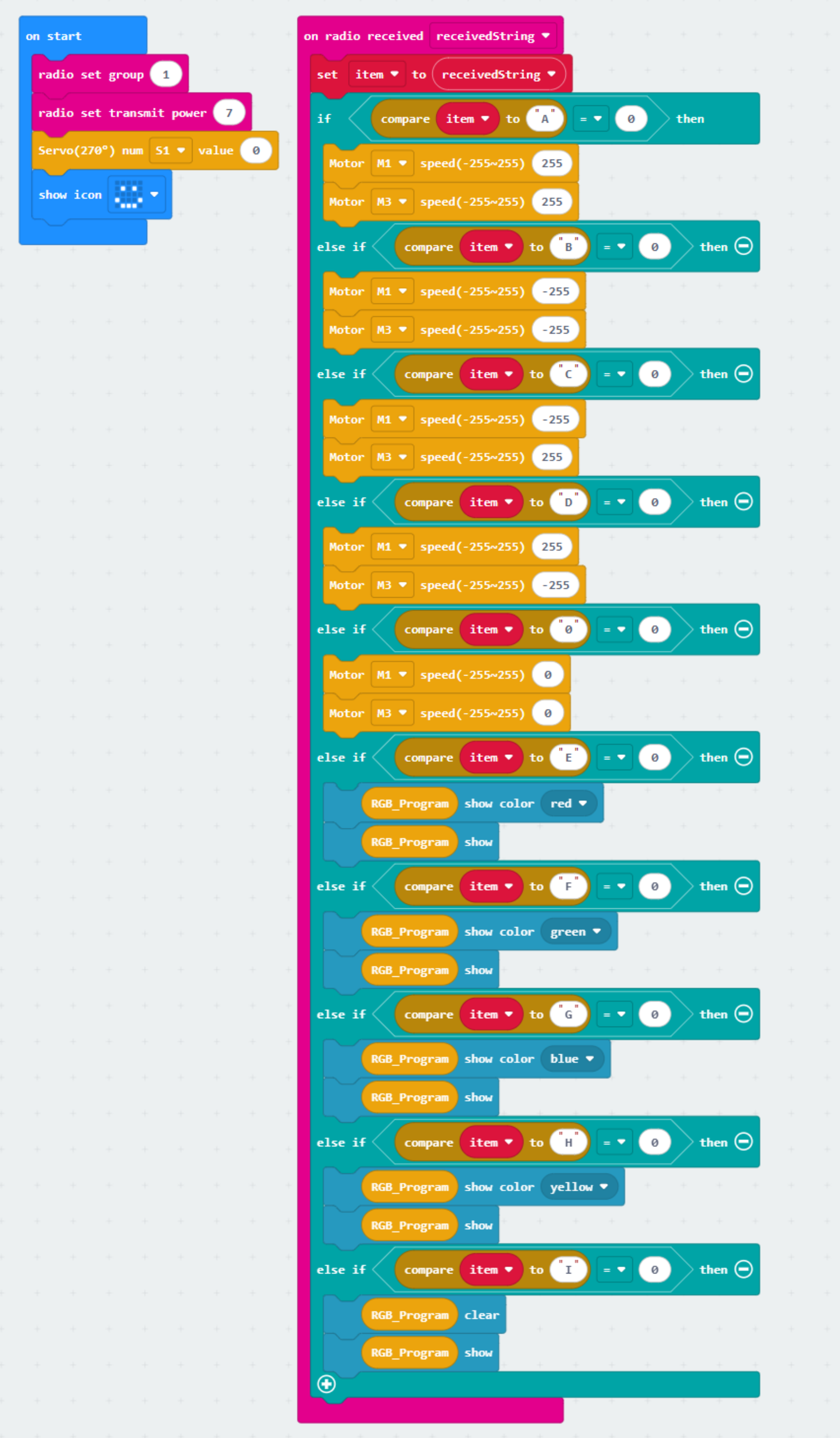


New variable name:

value

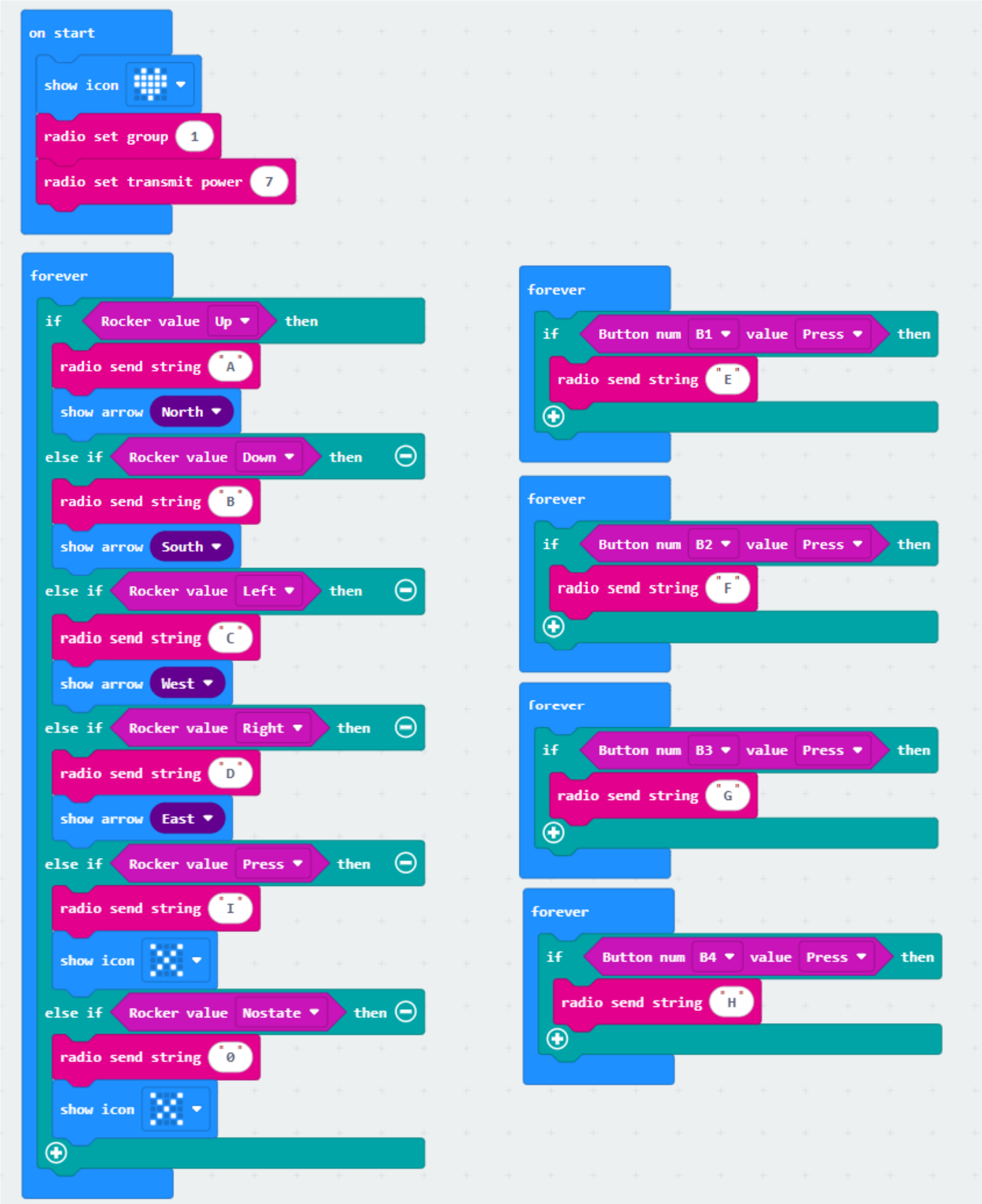
4.4 Assemble building blocks

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



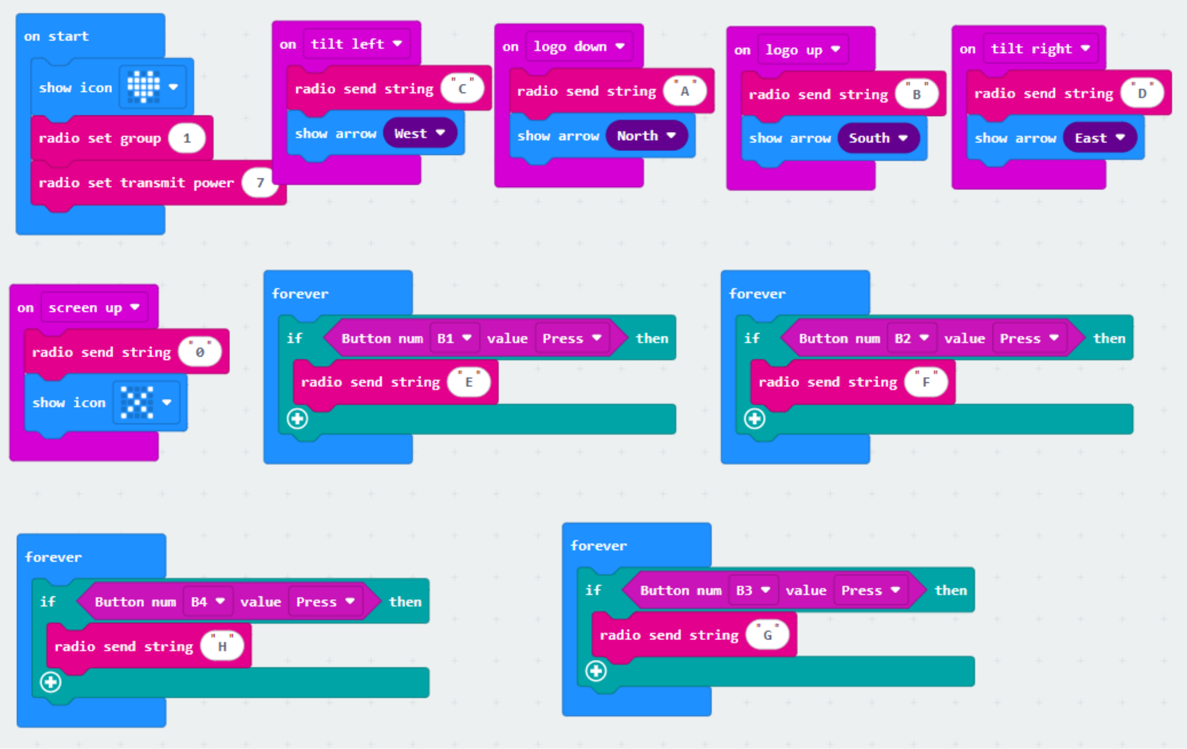
```
on start
    radio set group 1
    radio set transmit power 7
    Servo(270°) num S1 value 0
    show icon

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
        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
        RGB_Program show
    else if compare item to "H" = 0 then
        RGB_Program show color yellow
        RGB_Program show
    else if compare item to "I" = 0 then
        RGB_Program clear
        RGB_Program show
```

Handle rocker control program is as follows



```
on start
    show icon
    radio set group 1
    radio set transmit power 7

forever
    if Rocker value Up then
        radio send string "A"
        show arrow North
    else if Rocker value Down then
        radio send string "B"
        show arrow South
    else if Rocker value Left then
        radio send string "C"
        show arrow West
    else if Rocker value Right then
        radio send string "D"
        show arrow East
    else if Rocker value Press then
        radio send string "I"
        show icon
    else if Rocker value Nostate then
        radio send string "0"
        show icon

forever
    if Button num B1 value Press then
        radio send string "E"

forever
    if Button num B2 value Press then
        radio send string "F"

forever
    if Button num B3 value Press then
        radio send string "G"

forever
    if Button num B4 value Press then
        radio send string "H"
```

Handle gravity control program is as follows



```
on start
    show icon
    radio set group 1
    radio set transmit power 7

on tilt left
    radio send string "C"
    show arrow West

on logo down
    radio send string "A"
    show arrow North

on logo up
    radio send string "B"
    show arrow South

on tilt right
    radio send string "D"
    show arrow East

on screen up
    radio send string "0"
    show icon

forever
    if Button num B1 value Press then
        radio send string "E"

forever
    if Button num B2 value Press then
        radio send string "F"

forever
    if Button num B4 value Press then
        radio send string "H"

forever
    if Button num B3 value Press then
        radio send string "G"
```

You can also directly open the **microbit-Handle-control-pretty-car.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

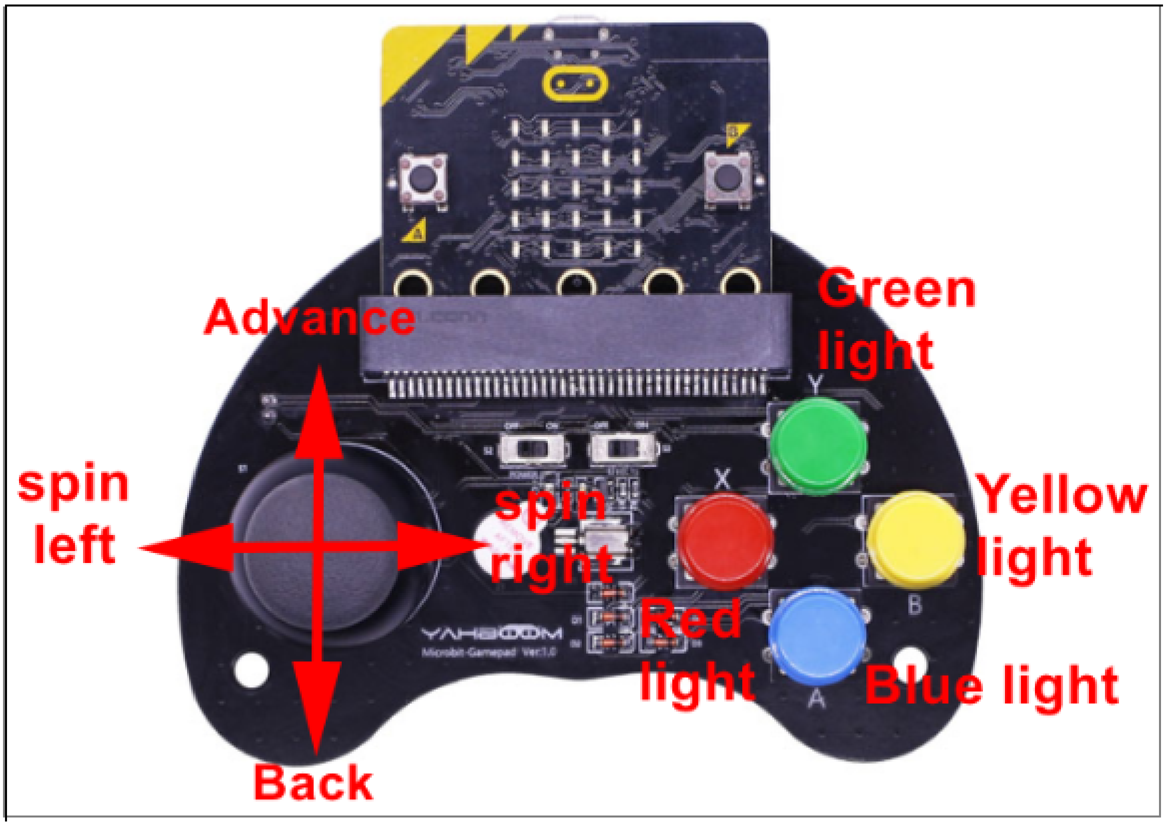
5. Experimental phenomenon

We need to download the Pretty car program to the micro:bit motherboard of Pretty car, open Pretty 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 display an "X" pattern, 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 remote control of Pretty car.

The controller functions are as follows.



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