Micro:bit handle control

1. Learning objectives

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

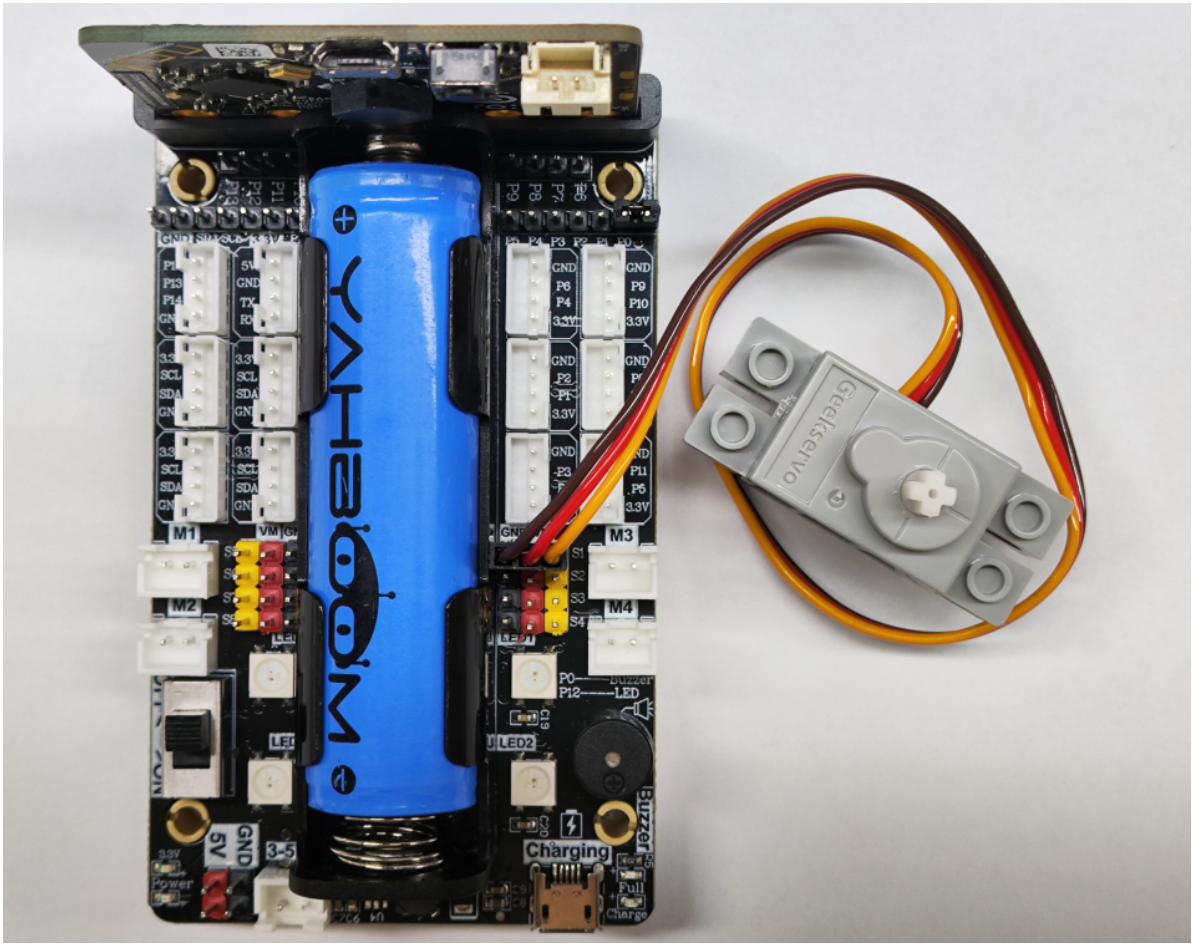
2. Building blocks

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

3. Motor wiring

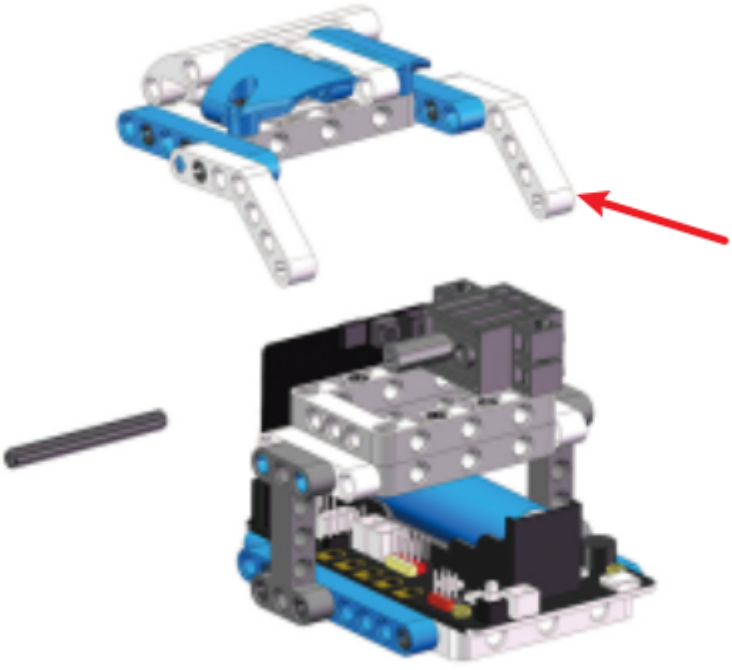
Insert the building blocks servo wiring into the S1 interface of the Super:bit expansion board, and the orange servo wiring into the yellow pin of S1.

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 white building blocks installed 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 Changing Face mask to the closed state, as shown in the figure below, and then install the building blocks. (If you have used the Changing Face and servo-related programs before, you can skip this step)



4. Programming

Method 1 Online Programming:

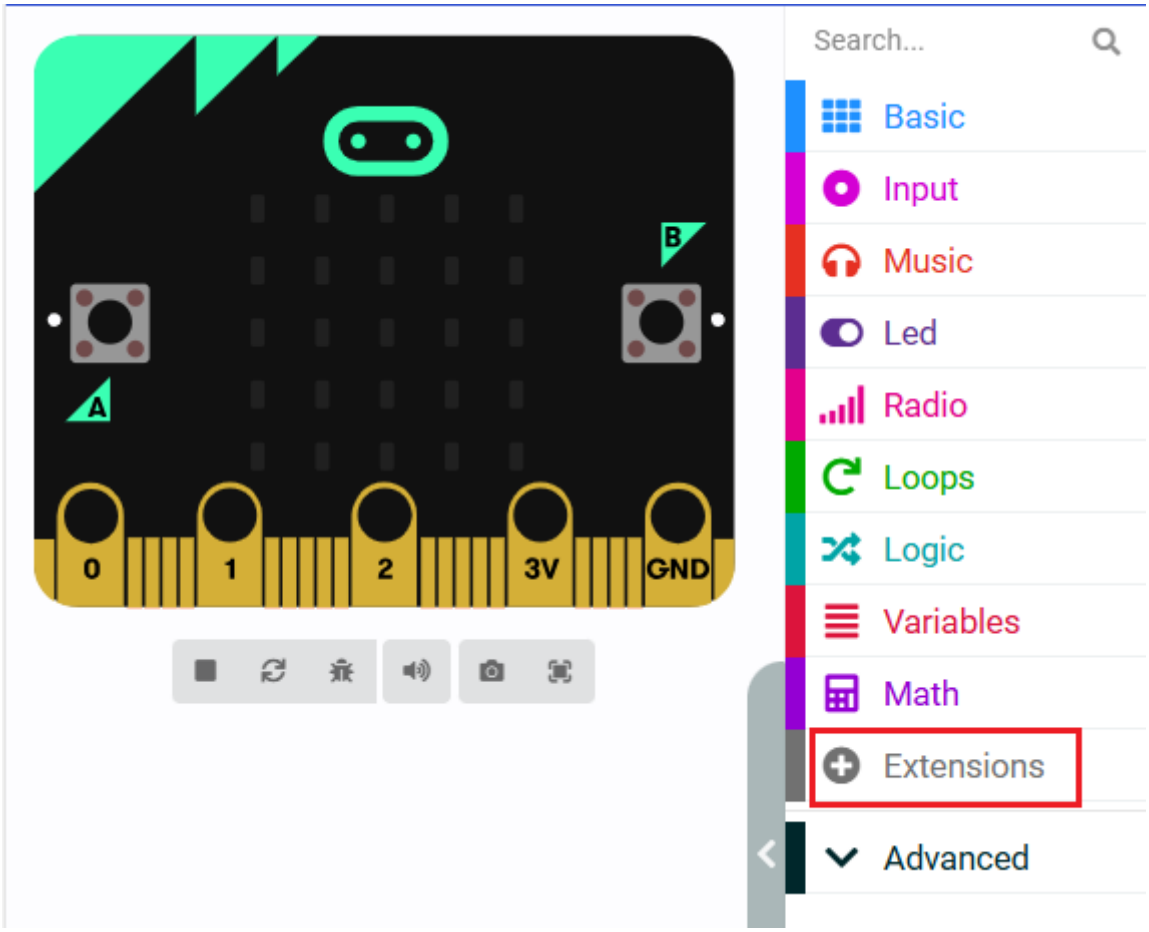
First, connect the micro:bit to the computer via a USB data cable, and a U disk will pop up on the computer. Click the URL in the U disk: https://makecode.microbit.org/ to enter the programming interface. Then, add the Yabo smart software package to program.

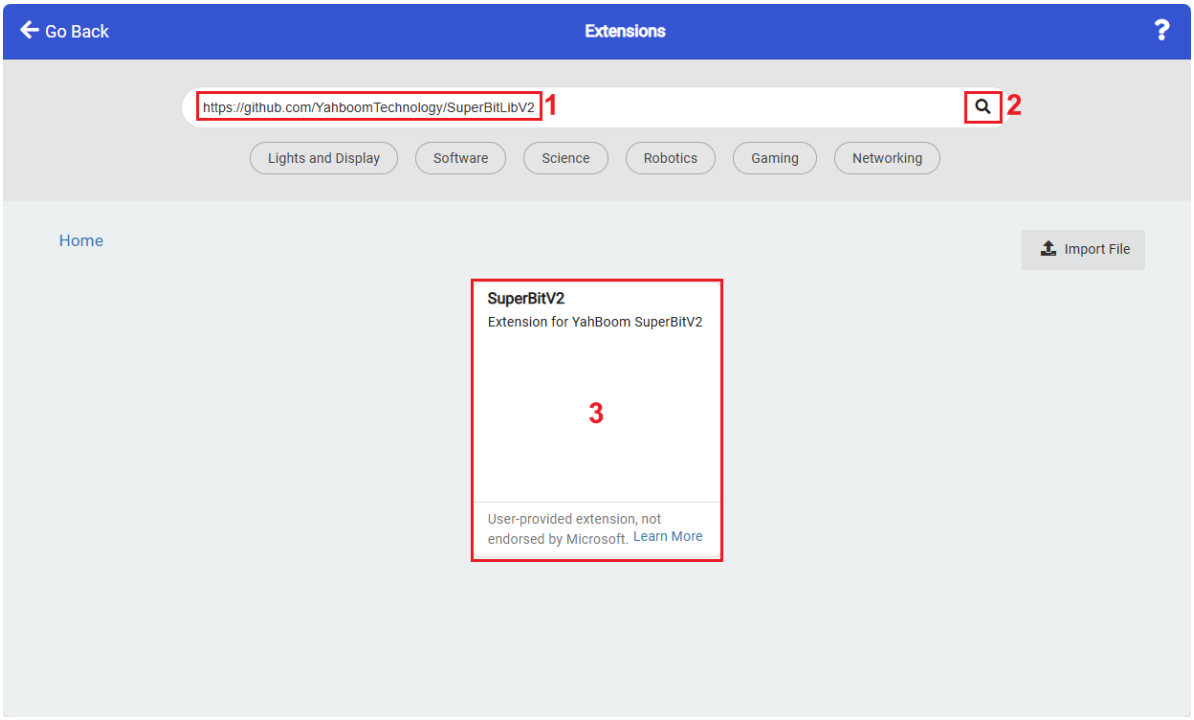
Method 2 Offline Programming:

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

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

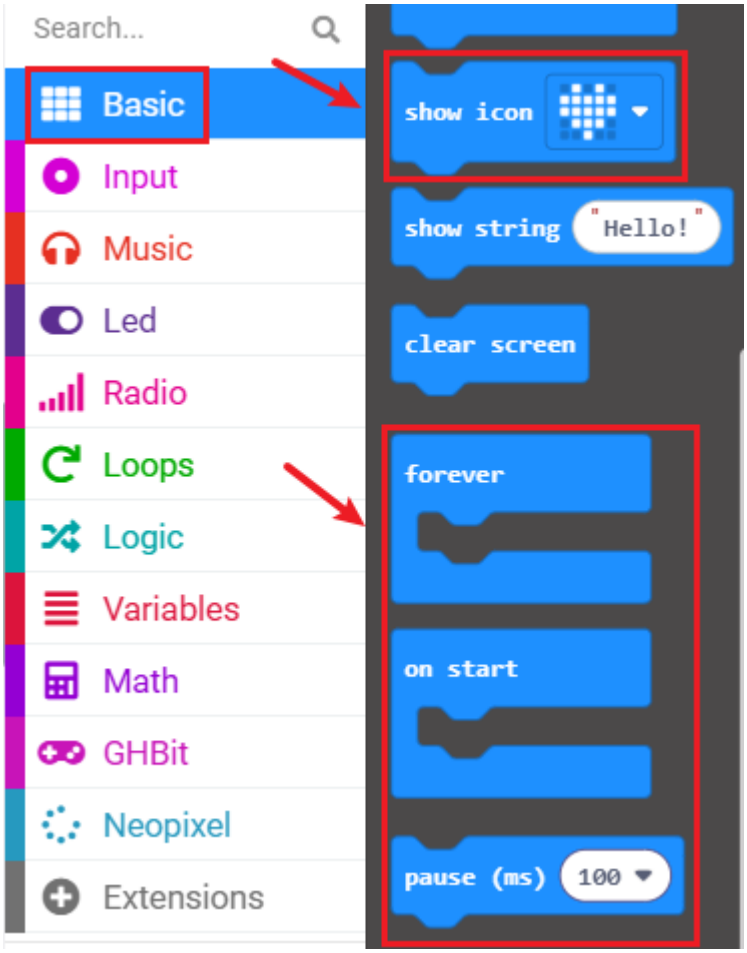
4.1 Add expansion package

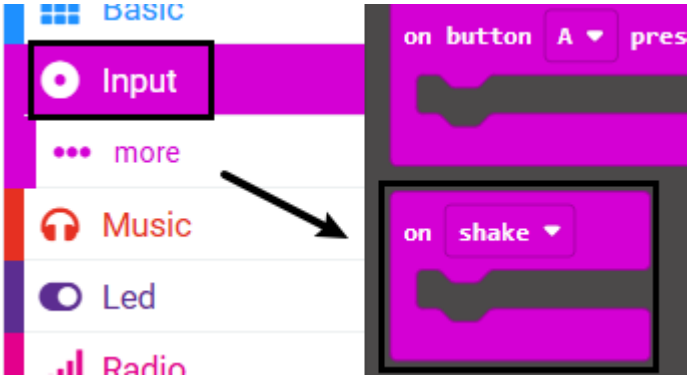


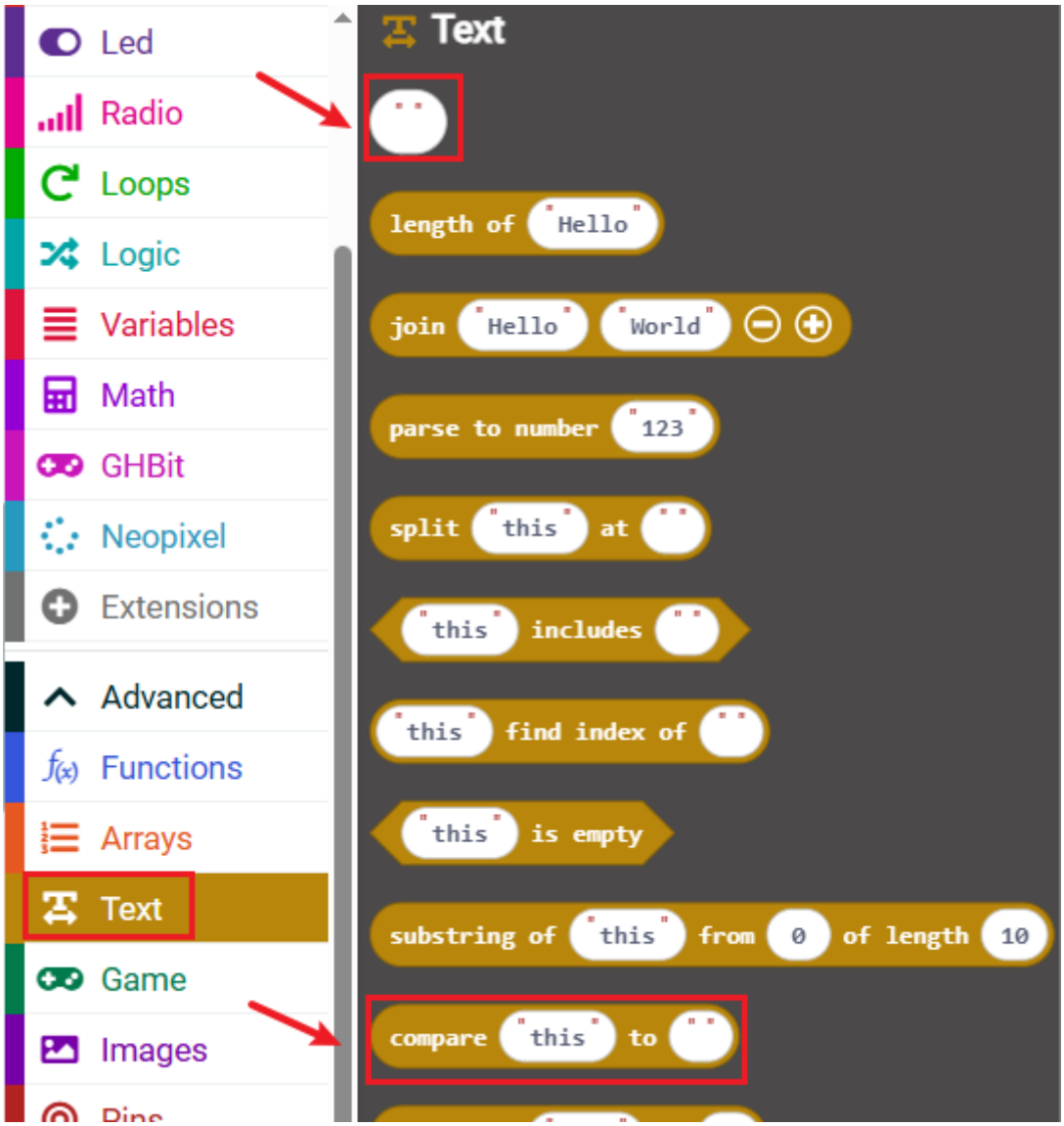


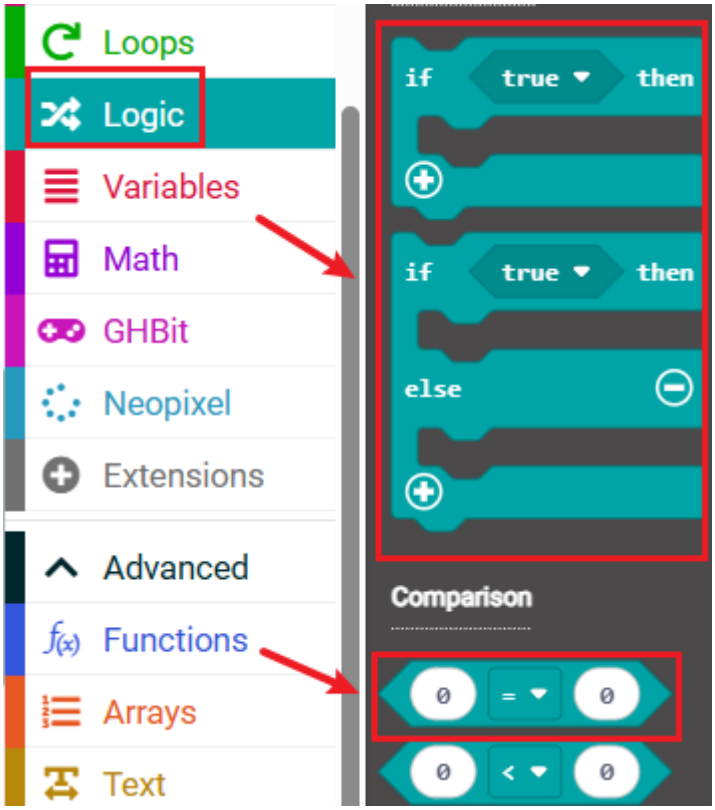
4.2 Building blocks used

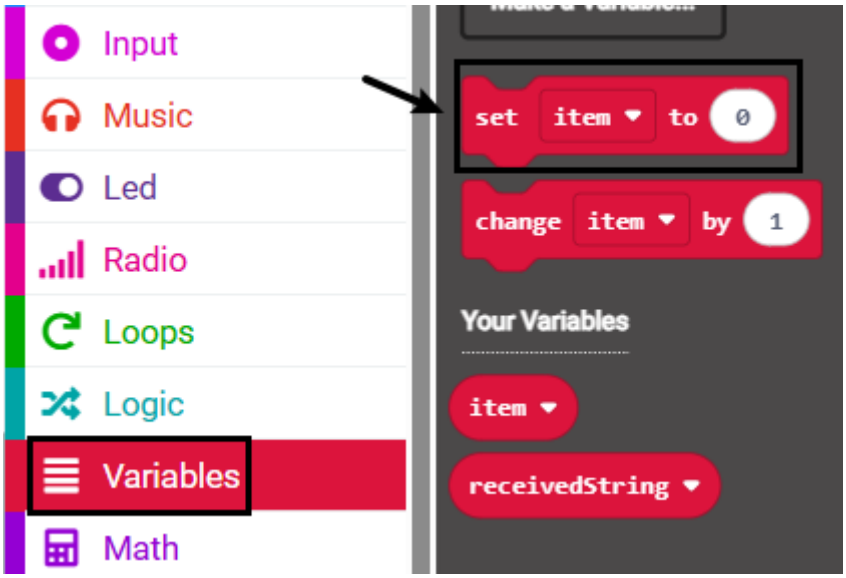
The location of the building blocks required for this programming is shown in the figure below.

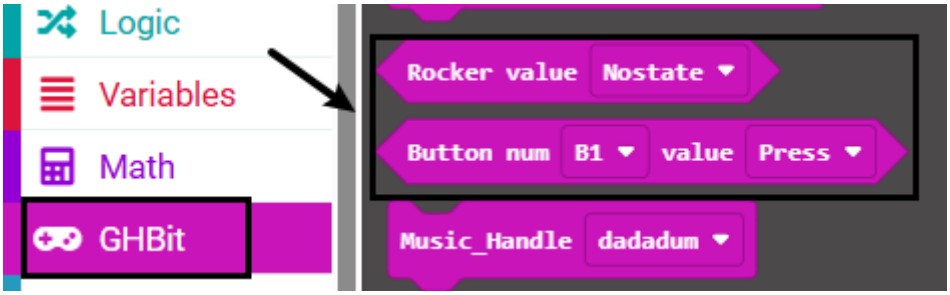



Basic
Input
more
Music
Led
Radio
on button A pres
on shake

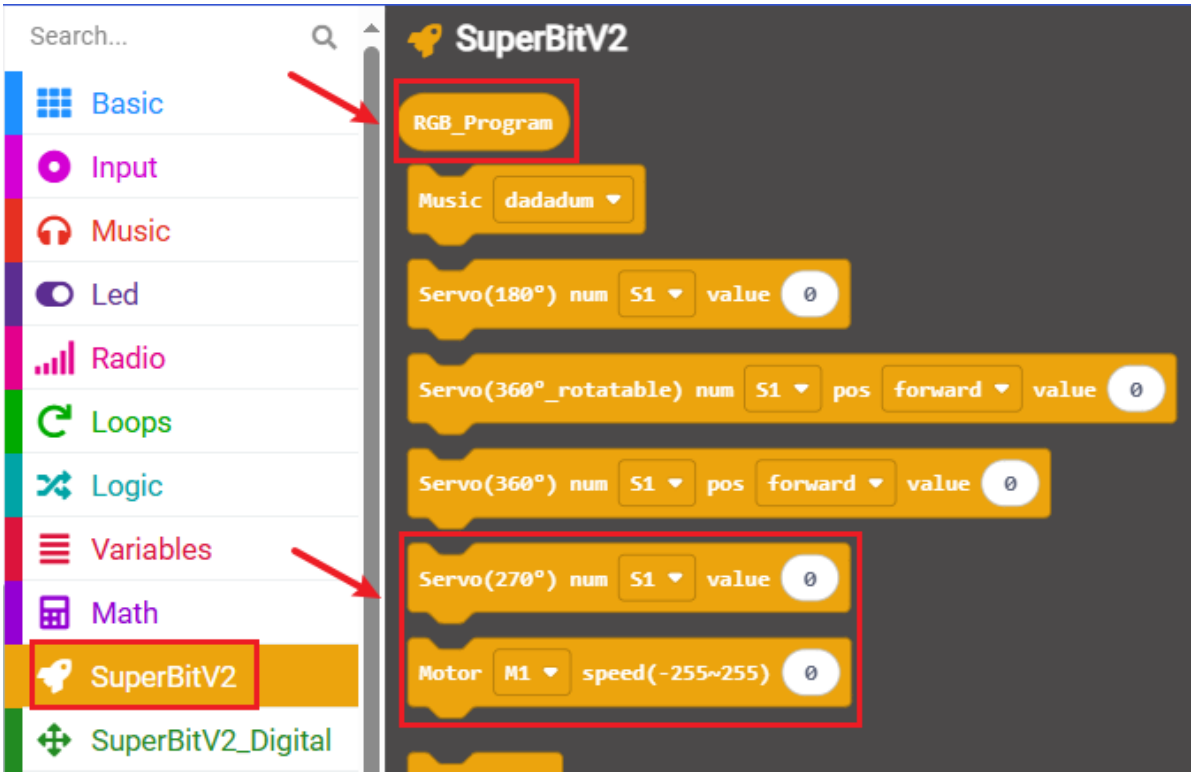

Led
Radio
Loops
Logic
Variables
Math
GHBit
Neopixel
Extensions
Advanced
Functions
Arrays
Text
Game
Images
Text
length of "Hello"
join "Hello" "World"
parse to number "123"
split "this" at ""
"this" includes ""
"this" find index of ""
"this" is empty
substring of "this" from 0 of length 10
compare "this"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


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

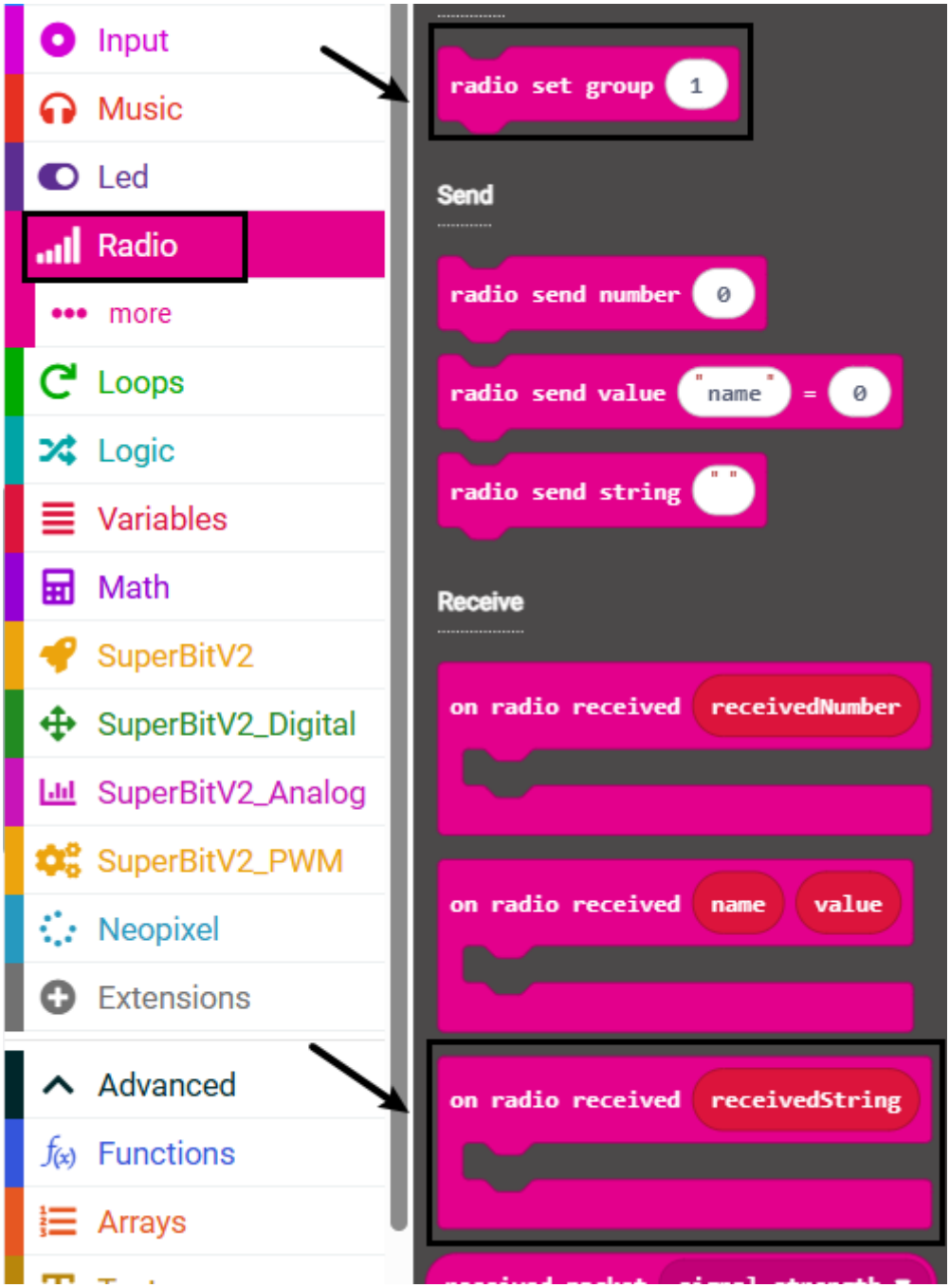

Input
Music
Led
Radio
Loops
Logic
Variables
Math
set item to 0
change item by 1
Your Variables
item
receivedString


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

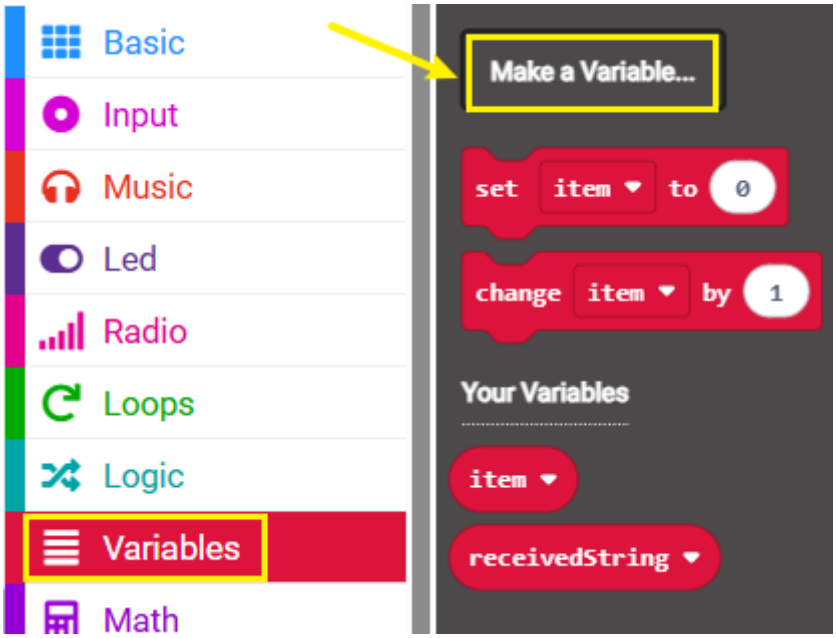

SuperBitV2
SuperBitV2_Digital
SuperBitV2_Analog
SuperBitV2_PWM
Neopixel
more
Extensions
strip show rainbow from 1 to 360
strip show color red
strip show bar graph of 0 up to 255
strip show
strip clear


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

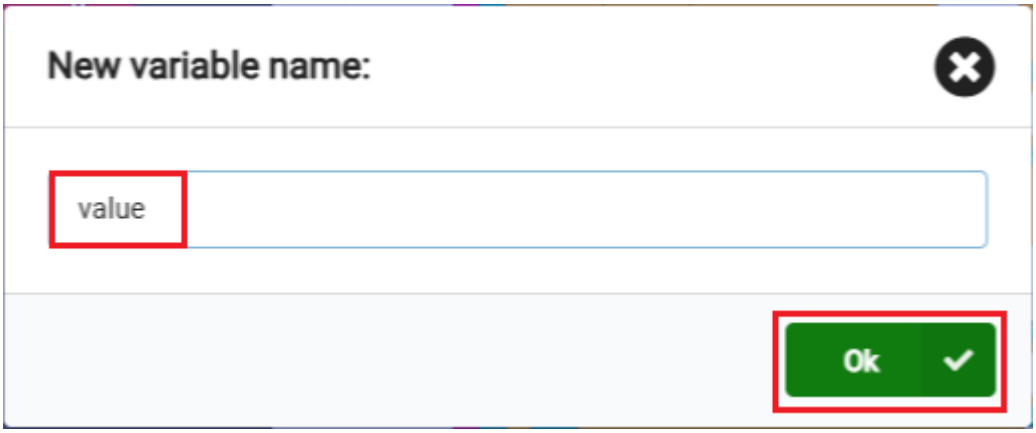

Search...
Basic
Input
Music
Led
Radio
more
more
radio set transmit power 7
radio set transmit serial number true
radio set frequency band 0
radio raise event
from source MICROBIT_ID_BUTTON_A

4.3 Add new variables

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

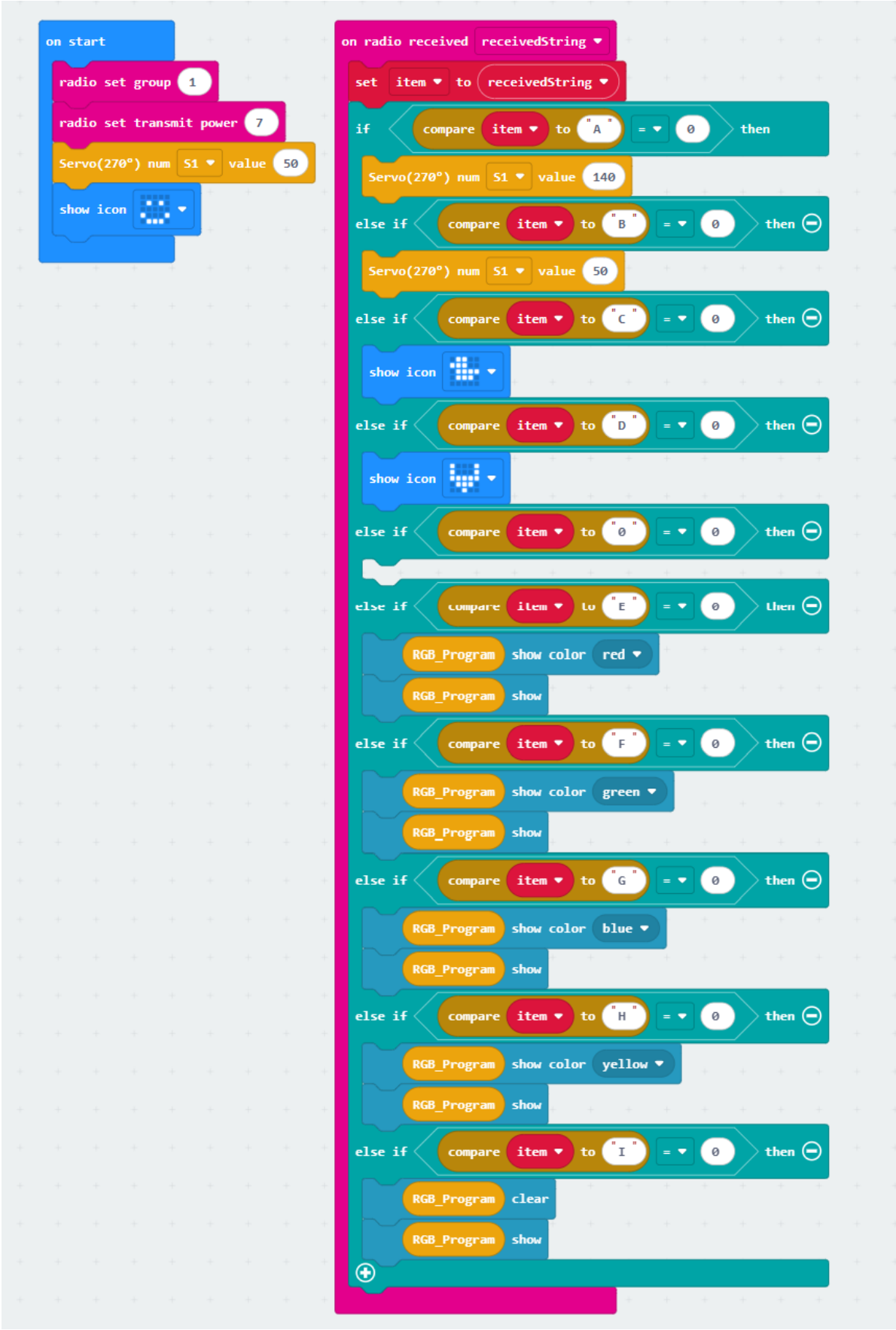


② Enter the variable name to complete the new variable.

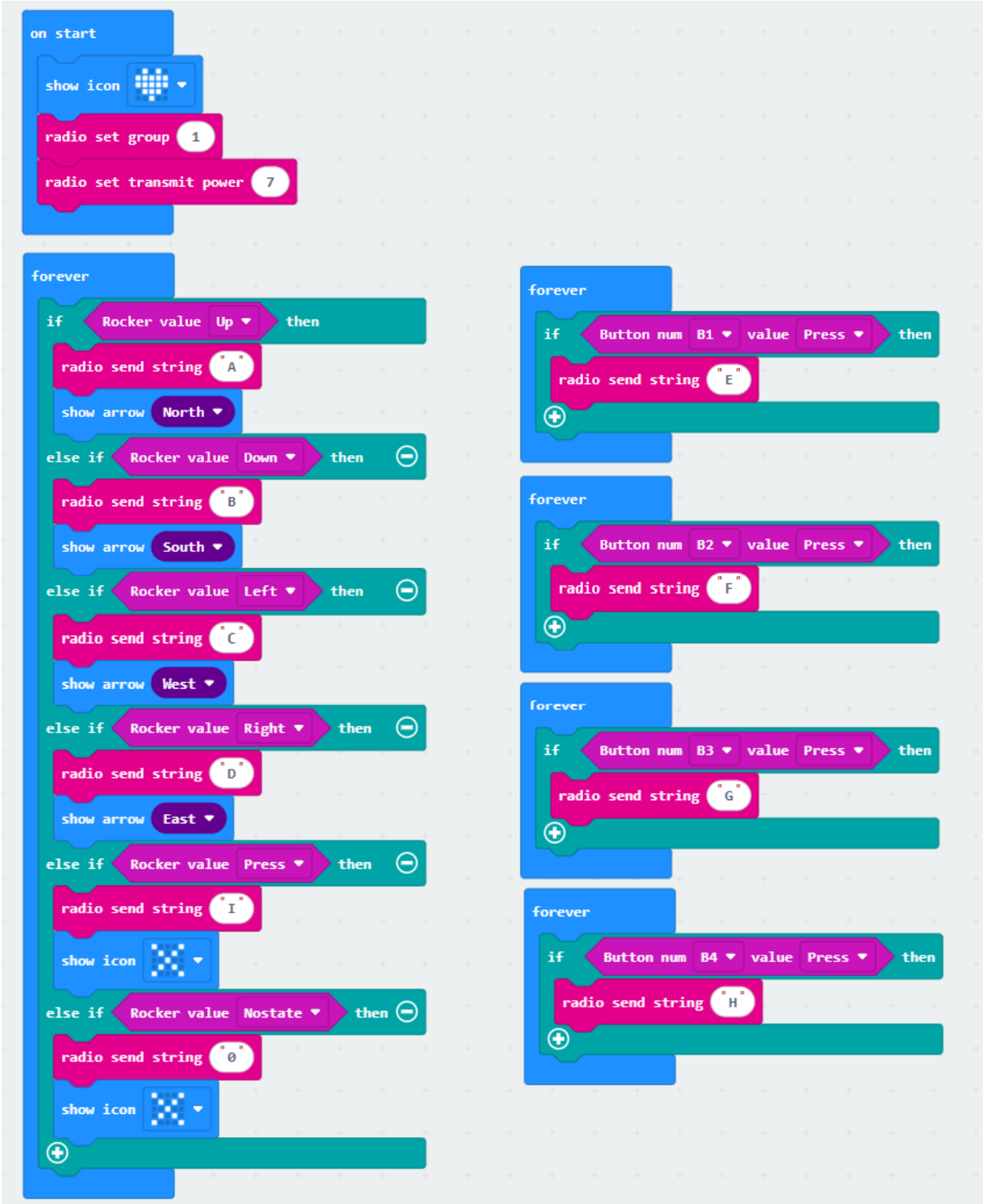


4.4 Combined blocks

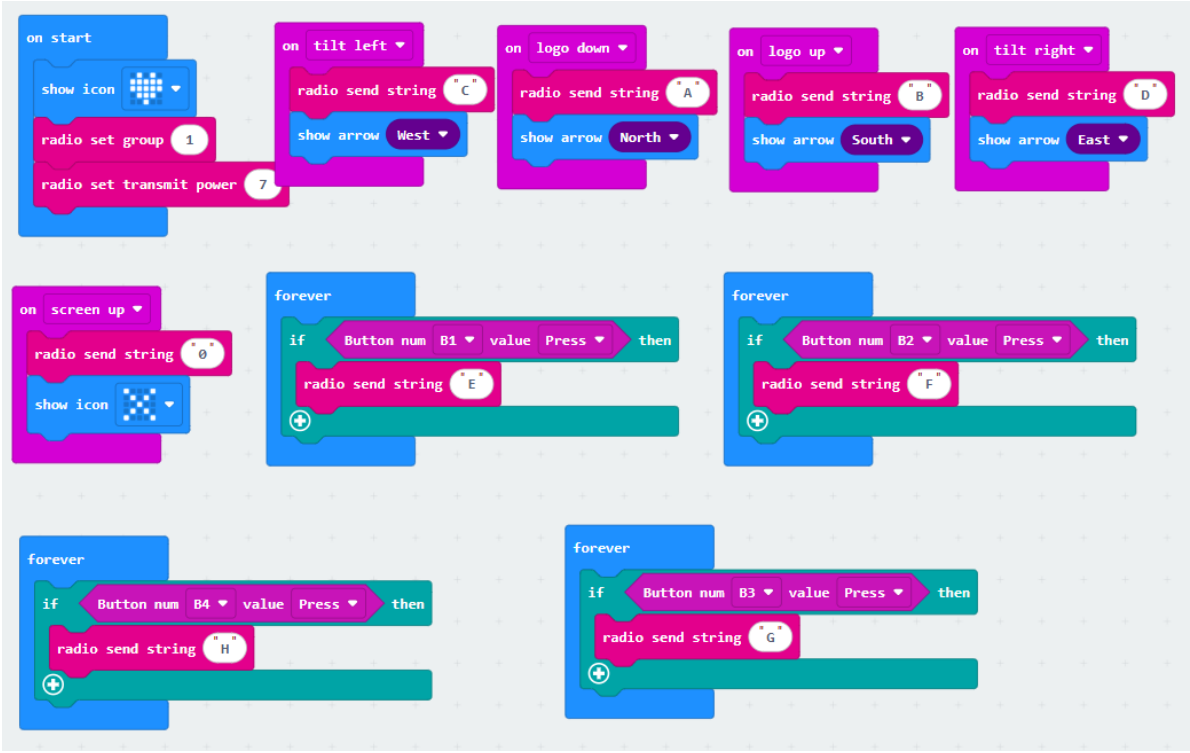
The summary program of microbit on **Changing Face** is shown in the figure below



The program of **Handle rocker control** is as follows



The program of **Handle gravity control** is as follows



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

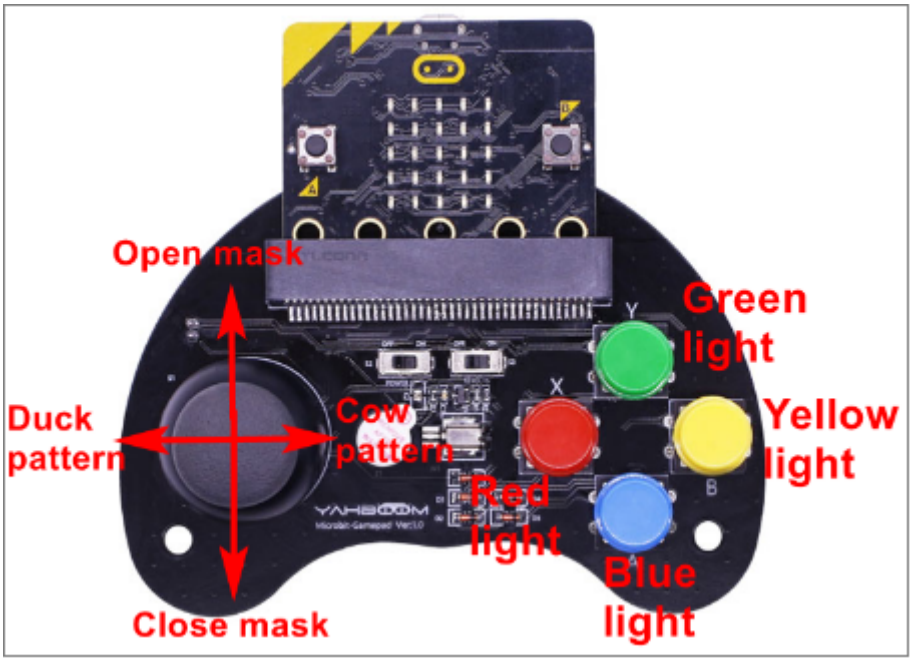
5. Experimental phenomenon

We need to download the Changing Face program to the micro:bit motherboard of Changing Face, turn on the power switch of Changing Face, and we can see a smiley face pattern displayed 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 the micro:bit dot matrix will initialize to display a heart pattern, and then display an "X" pattern, indicating that the handle is in the default state and no data is sent.

The two will automatically pair, and then we can start remote control of Changing Face.

The handle functions are as follows.


Open mask
Green light
Duck pattern
Cow pattern
Yellow light
Red light
Blue light
Close mask