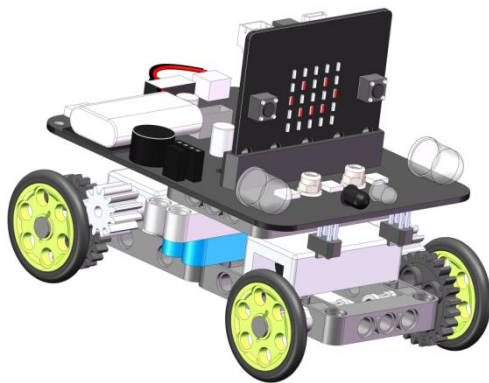
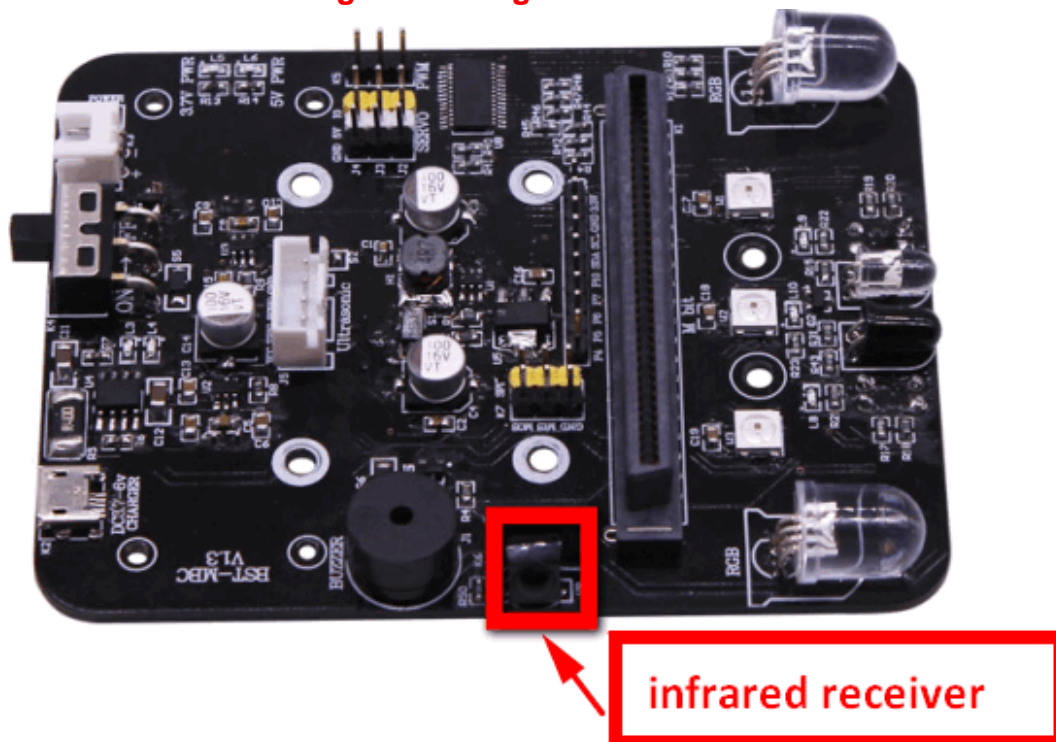


Lesson2 of Building:bit Mini car---“Infrared remote control”



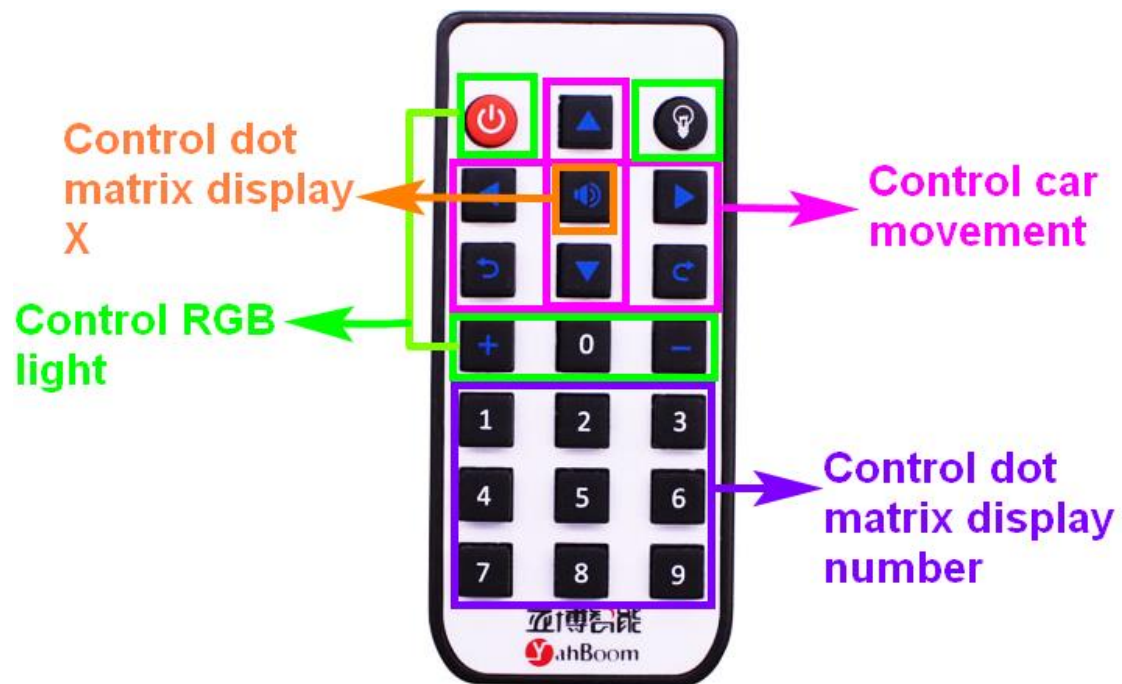
Note:

1. When performing infrared remote control, the remote controller should face the infrared receiver on the expansion board.
2. There is a plastic piece on the bottom of the infrared remote controller that needs to be taken down for normal use.
3. The infrared light emitted by the infrared remote controller and the infrared receiver is invisible to the human eye. It can be seen under the camera without filtering infrared light.



1.Experimental phenomena

The corresponding functions of the infrared remote control are shown below.



2.Preparation before class

We need to be ready:

Building Block Mini car*1

Infrared remote controller*1

USB data cable*1

2-1.Two programming methods:

Online programming:

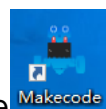
First, we need to connect the micro:bit to the computer by USB data cable, the computer will pop up a USB flash drive. Then, click on the URL in the USB flash drive: <http://microbit.org/> to enter the edit process interface, click to

【Extensions】, and copy the package URL:

https://github.com/lzty634158/yahboom_mbit_en and

https://github.com/lzty634158/YB_IR to the input field, and you can use the building blocks of the Yahboom software package.

Offline programming:

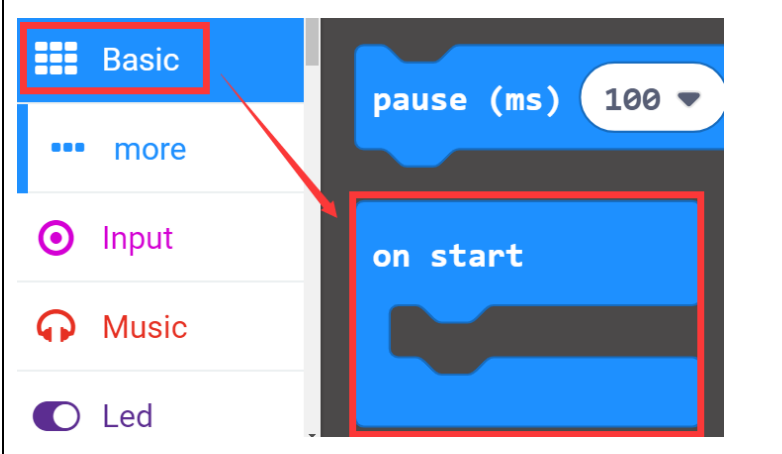
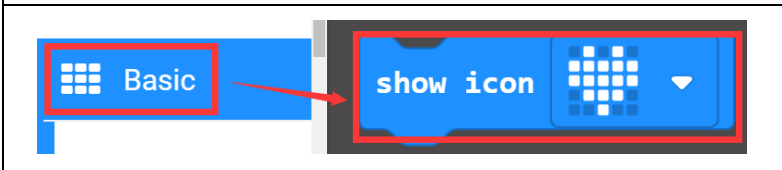
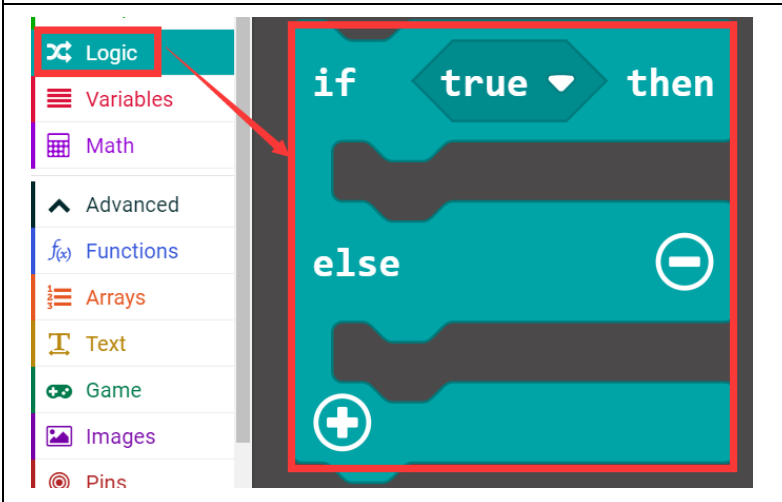
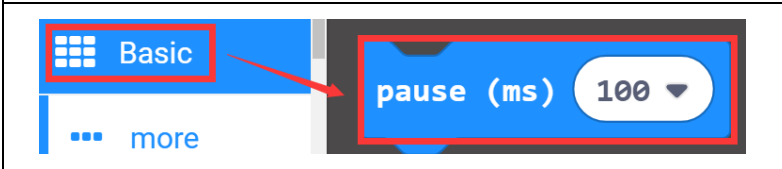



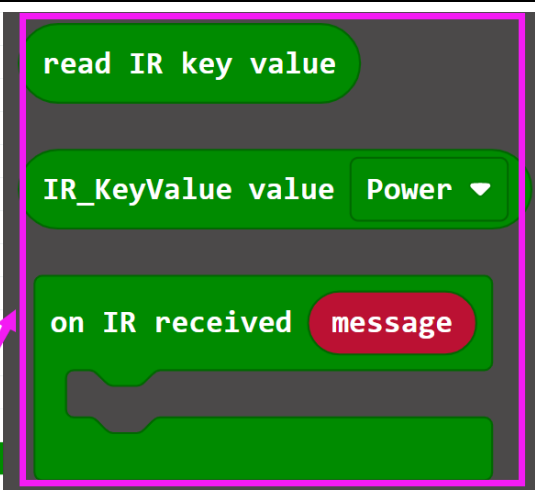
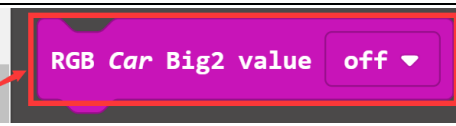

Open the offline programming software [Makecode](#), click to 【Extension】 and copy the package URL: https://github.com/lzty634158/yahboom_mbit_en and https://github.com/lzty634158/YB_IR to the input field, and you can use the building blocks of the Yahboom software package.

For detailed programming, please read the documentation before class

【1. Preparation before class】----【Introduction of programming method】. We use micro:bit official website for online programming in here.

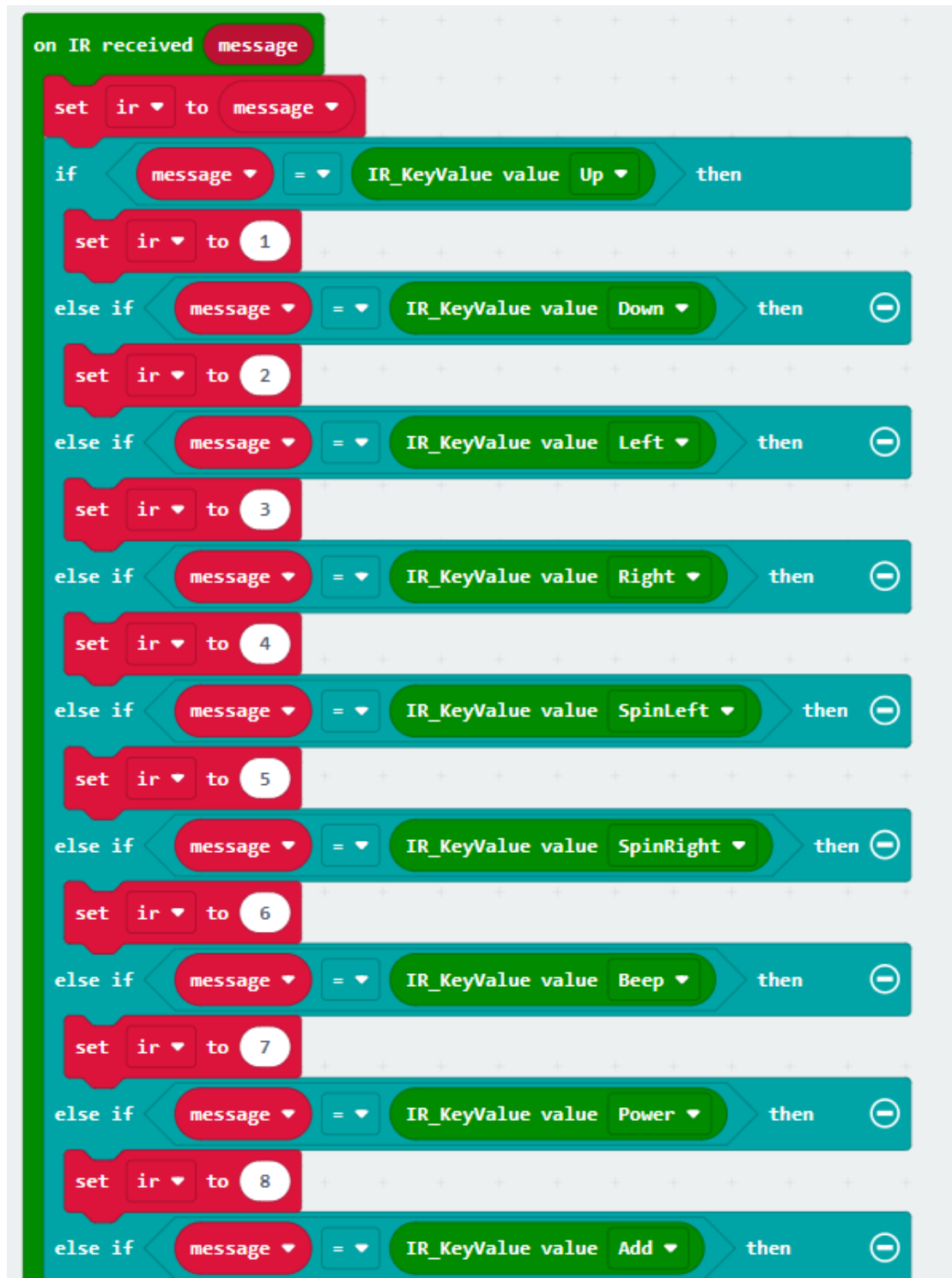
3.Studying blocks

Blocks	Instruction
	<p>Executed at boot time, the code is only executed once.</p>
	<p>Display image on the lattice of micro:bit.</p>
	<p>If true then execute. If it is false, it will not be executed.</p>
	<p>The program pauses for 100 milliseconds and the time can be modified by yourself.</p>
	<p>The car's motion state selection. You can select forward, back, turn left, turn right, rotate left, rotate right, and stop.</p>

<ul style="list-style-type: none"> Led Radio Loops Logic Variables Math Mbit_Display Mbit_Sensor Mbit_Input Mbit_Music Mbit_Motor Mbit_Robot Mbit_IR_V1.5 Mbit_IR_V2 		<p>When the power button on the remote controller is pressed, the code inside will be executed, and the button can be customized.</p> <p>Set the infrared remote control receiving pin. In this experiment, the receiving pin is P8, so you must select P8, otherwise you will not receive the signal.</p>
<ul style="list-style-type: none"> Mbit_Motor Mbit_Robot 		<p>Select the color of the lights.</p>
<ul style="list-style-type: none"> Music 		<p>Can play different tones.</p>

4.Programming

Next, we started to write the program for the infrared remote control of the building block Mini car, as shown below:



The image shows a Scratch script designed to map specific IR key values to a sequence of IR numbers (9 through 18). The script is composed of a series of 'set' and 'else if' blocks. Each 'set' block is followed by an 'else if' block that checks if the 'message' variable is equal to a specific 'IR_KeyValue' value. If the condition is met, the script proceeds to the next 'set' block. The 'IR_KeyValue' values are Zero, Minus, One, Two, Three, Four, Five, Six, and Seven, which correspond to IR numbers 9 through 18 respectively. The script is visually organized with red 'set' blocks and teal 'else if' blocks, each containing a dropdown menu for the 'message' variable and a dropdown menu for the 'IR_KeyValue' value. The 'set' blocks are labeled with the IR number they are setting (9 to 18). The 'else if' blocks are labeled with the 'IR_KeyValue' value they are checking for. The script is set against a light gray background with a green vertical bar on the left side.

```
set ir to 9
else if message = IR_KeyValue value Zero then
set ir to 10
else if message = IR_KeyValue value Minus then
set ir to 11
else if message = IR_KeyValue value One then
set ir to 12
else if message = IR_KeyValue value Two then
set ir to 13
else if message = IR_KeyValue value Three then
set ir to 14
else if message = IR_KeyValue value Four then
set ir to 15
else if message = IR_KeyValue value Five then
set ir to 16
else if message = IR_KeyValue value Six then
set ir to 17
else if message = IR_KeyValue value Seven then
set ir to 18
```

```

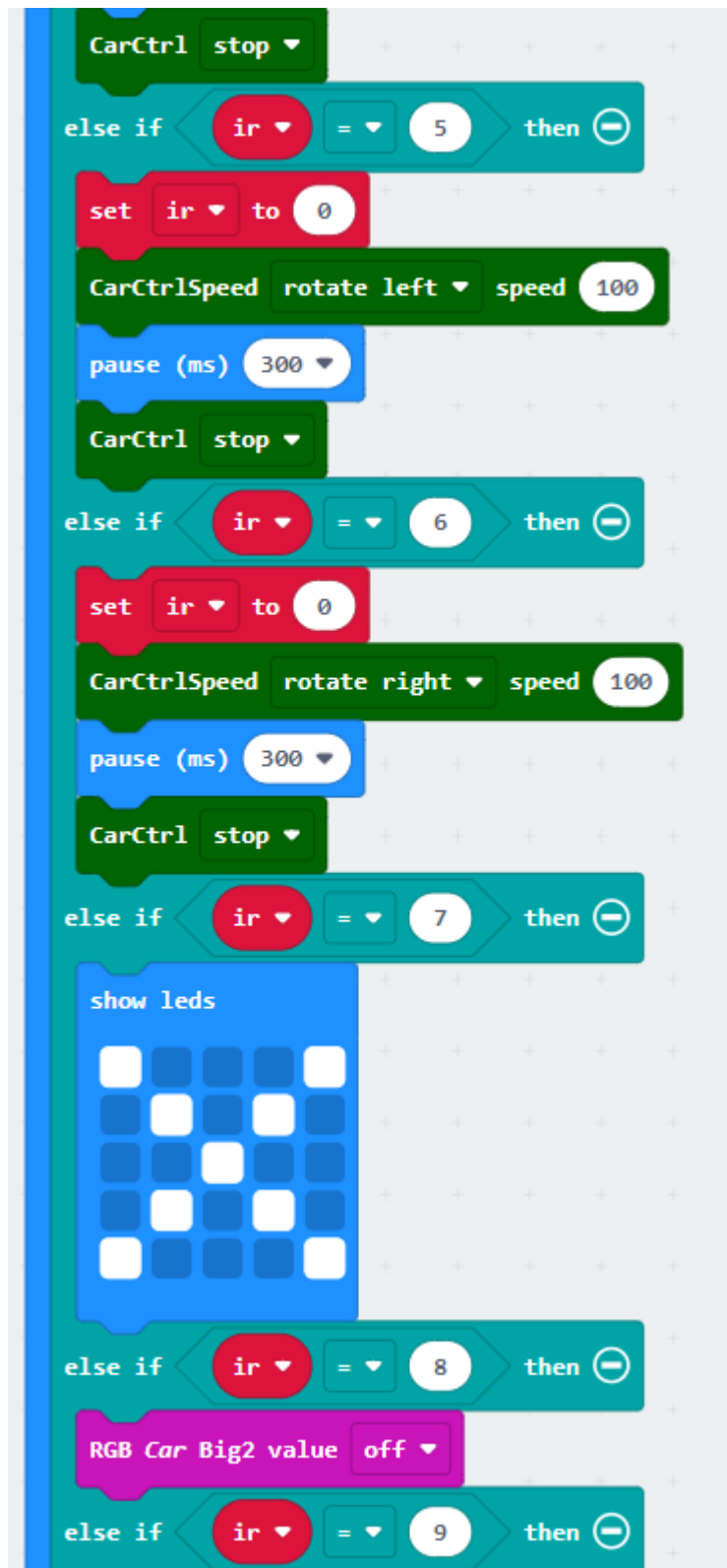
else if <message v = IR_KeyValue value Eight v> then -
  set ir v to 19
else if <message v = IR_KeyValue value Nine v> then -
  set ir v to 20
else if <message v = IR_KeyValue value Light v> then -
  set ir v to 21
+

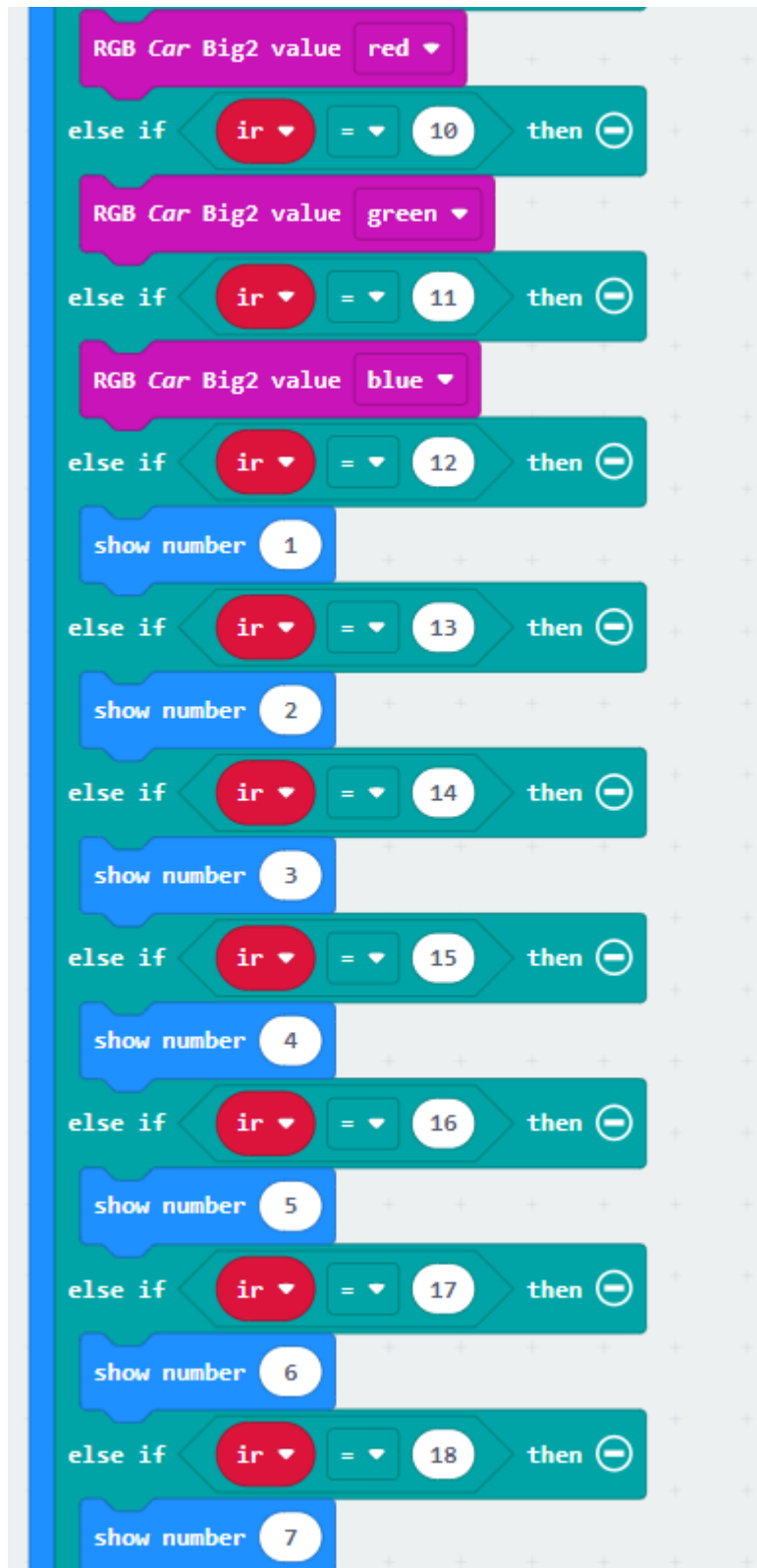
```

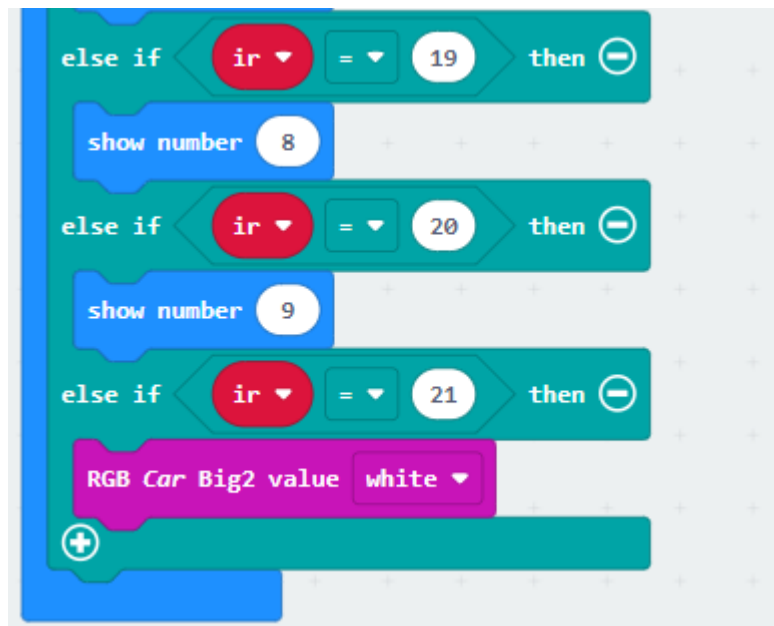
```

forever
  if <ir v = 2> then
    set ir v to 0
    CarCtrlSpeed forward v speed 100
    pause (ms) 300
    CarCtrl stop v
  else if <ir v = 2> then -
    set ir v to 0
    CarCtrlSpeed back v speed 100
    pause (ms) 300
    CarCtrl stop v
  else if <ir v = 3> then -
    set ir v to 0
    CarCtrlSpeed turn left v speed 100
    pause (ms) 300
    CarCtrl stop v
  else if <ir v = 4> then -
    set ir v to 0
    CarCtrlSpeed turn right v speed 100
    pause (ms) 300

```







The above is the program for this Mini car. After writing, we need to download it to the micro:bit board.