Lesson4 Building:bit Caterpillar tripod ---"micro:bit handle remote control"

1.Experimental phenomena

After downloading the program, open the power switch of the Caterpillar tripod and the power switch of the micro:bit handle, they will be automatically paired. When you press the micro:bit handle remote control button, the Caterpillar tripod will have the corresponding action. The button on the right side of the remote control is used to control the color of the lights, and pressing the joystick down to turn off the light. Pushing the rocker forward, backward, left, right is to control the advance, back, turn left and turn right of the Caterpillar tripod.

2.Preparation before class

We needs to be ready:
Building Block Caterpillar tripod *1
micro:bit handle *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 to the input field, and you can use the building blocks of the Yahboom software package.

Offilne programming:
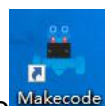
Open the offline programming software Makecode, click to 【Extension】 and copy the package URL: https://github.com/lzty634158/yahboom_mbit_en to the input field, and you can use the building blocks of the Yahboom software package.
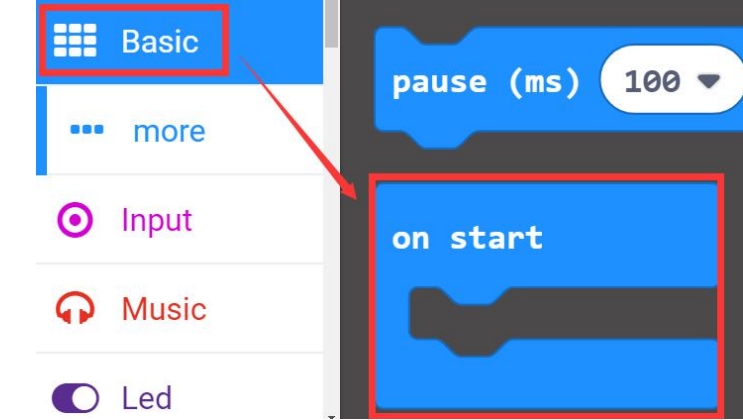
For detailed programming, please read the documentation before class 【1. Preparation before class】----【Introduction of programming method】. We use micro:bit official website for online programming in here.

3.Studying blocks

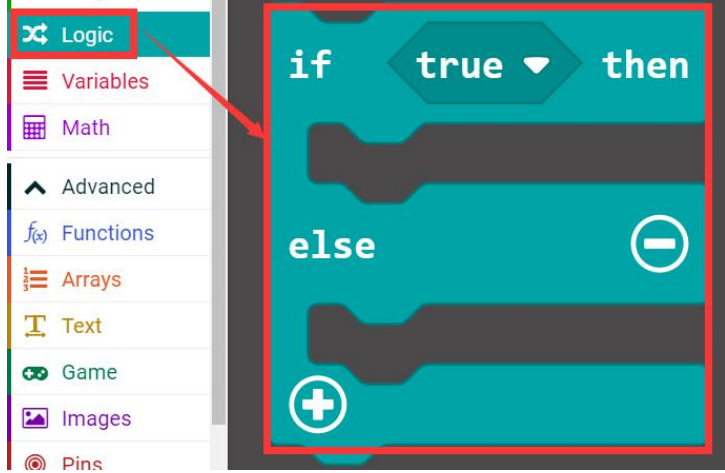| Blocks	Instruction
Basic → show icon	Display image on the lattice of micro:bit.
Led, Radio, more → radio set transmit power 7 / radio set transmit serial numbe	Set radio transmit power
Basic, more, Input, Music, Led → pause (ms) 100 / on start	Executed at boot time, the code is only executed once.
Radio, more, Loops → received packet signa / radio set group 1	Set the radio channel group, the receiving and sending must be in the same group.
Led, Radio, more → on radio received receivedNumber	The code inside is executed when radio receives data.
Logic, Variables → Make a Variable...	Set a variable we need.

New variable name: item Ok ✔ Cancel ✖	New variable name.
🔀 Logic ☰ Variables Math item ▾ set item ▾ to 0	Assign the received data to the variable item.
🔀 Logic ☰ Variables ▦ Math ∧ Advanced f(x) Functions ▤ Arrays T Text 🎮 Game 🖼 Images ◎ Pins if true ▾ then else ⊖ ⊕	If true then execute. If it is false, it will not be executed.
▦ Basic ••• more pause (ms) 100 ▾	The program pauses for 100 milliseconds and the time can be modified by yourself.

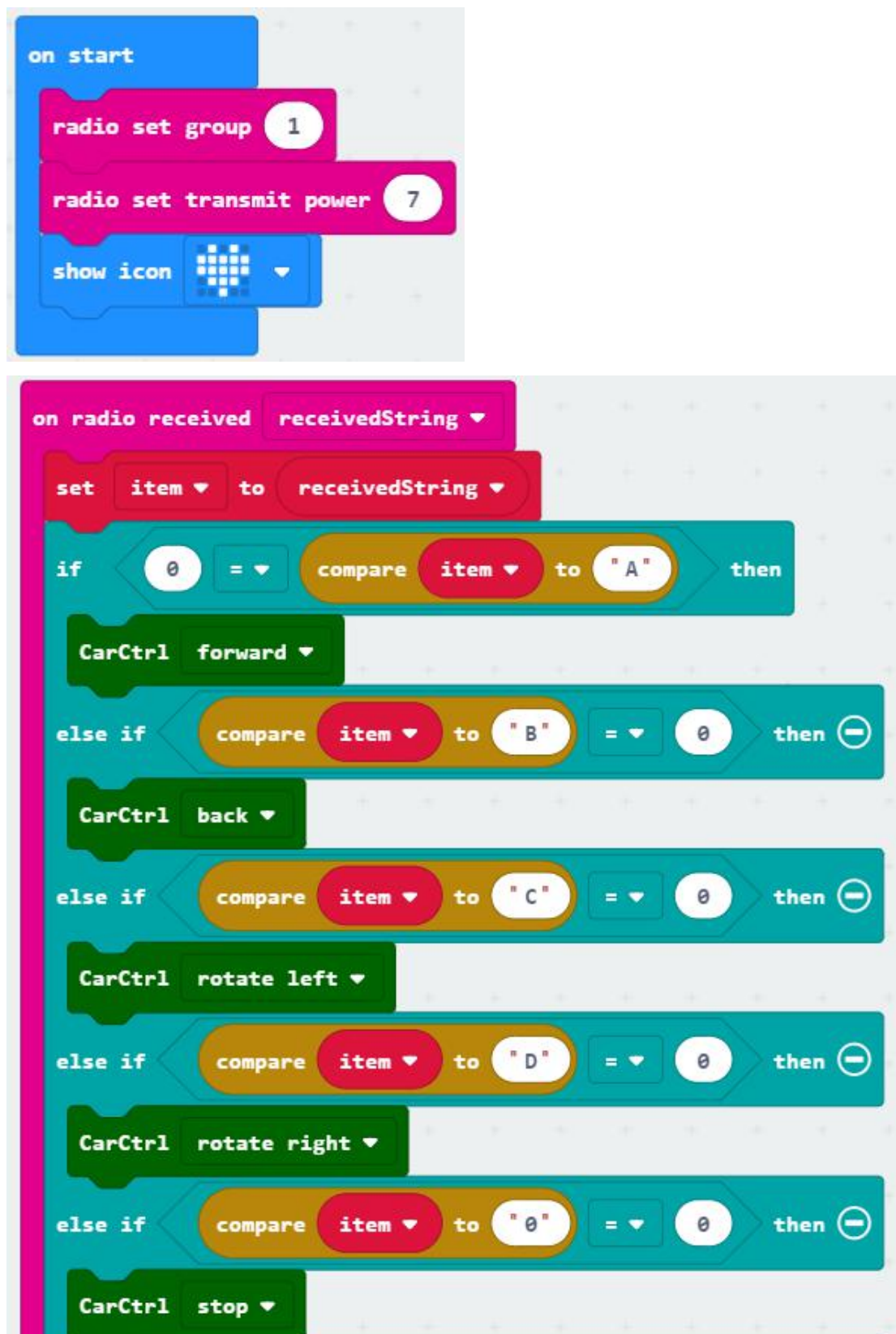|		
		Compare the two, if they are the same, they return 0, and if they are different, they return 1.
	Line_Sensor direct CarCtrl forward ▼	The car's motion state selection.You can select forward,back,turn left, turn right,rotate left, rotate right, and stop.
Mbit_Robot Mbit_IR	RGB *Car* Big value1 0 value2 0 value3 0	Select the color of the lights. 255 is the brightest.

4.Programming

Caterpillar tripod program:

Next, we started to write the program for the building block Caterpillar tripod .
After the writing, we need to download the program to the micro:bit board of
the building block Caterpillar tripod .

The program is shown below:

```
on start
    radio set group (1)
    radio set transmit power (7)
    show icon [heart]
```

```
on radio received (receivedString)
    set (item) to (receivedString)
    if (0) = (compare (item) to ("A")) then
        CarCtrl (forward)
    else if (compare (item) to ("B")) = (0) then
        CarCtrl (back)
    else if (compare (item) to ("C")) = (0) then
        CarCtrl (rotate left)
    else if (compare (item) to ("D")) = (0) then
        CarCtrl (rotate right)
    else if (compare (item) to ("0")) = (0) then
        CarCtrl (stop)
```

else if compare item ▼ to "E" = ▼ 0 then ⊖

RGB Car Big value1 255 value2 0 value3 0

else if compare item ▼ to "F" = ▼ 0 then ⊖

RGB Car Big value1 0 value2 255 value3 0

else if compare item ▼ to "G" = ▼ 0 then ⊖

RGB Car Big value1 0 value2 0 value3 255

else if compare item ▼ to "H" = ▼ 0 then ⊖

RGB Car Big value1 255 value2 255 value3 0

else if compare item ▼ to "I" = ▼ 0 then ⊖

RGB Car Big value1 0 value2 0 value3 0

⊕

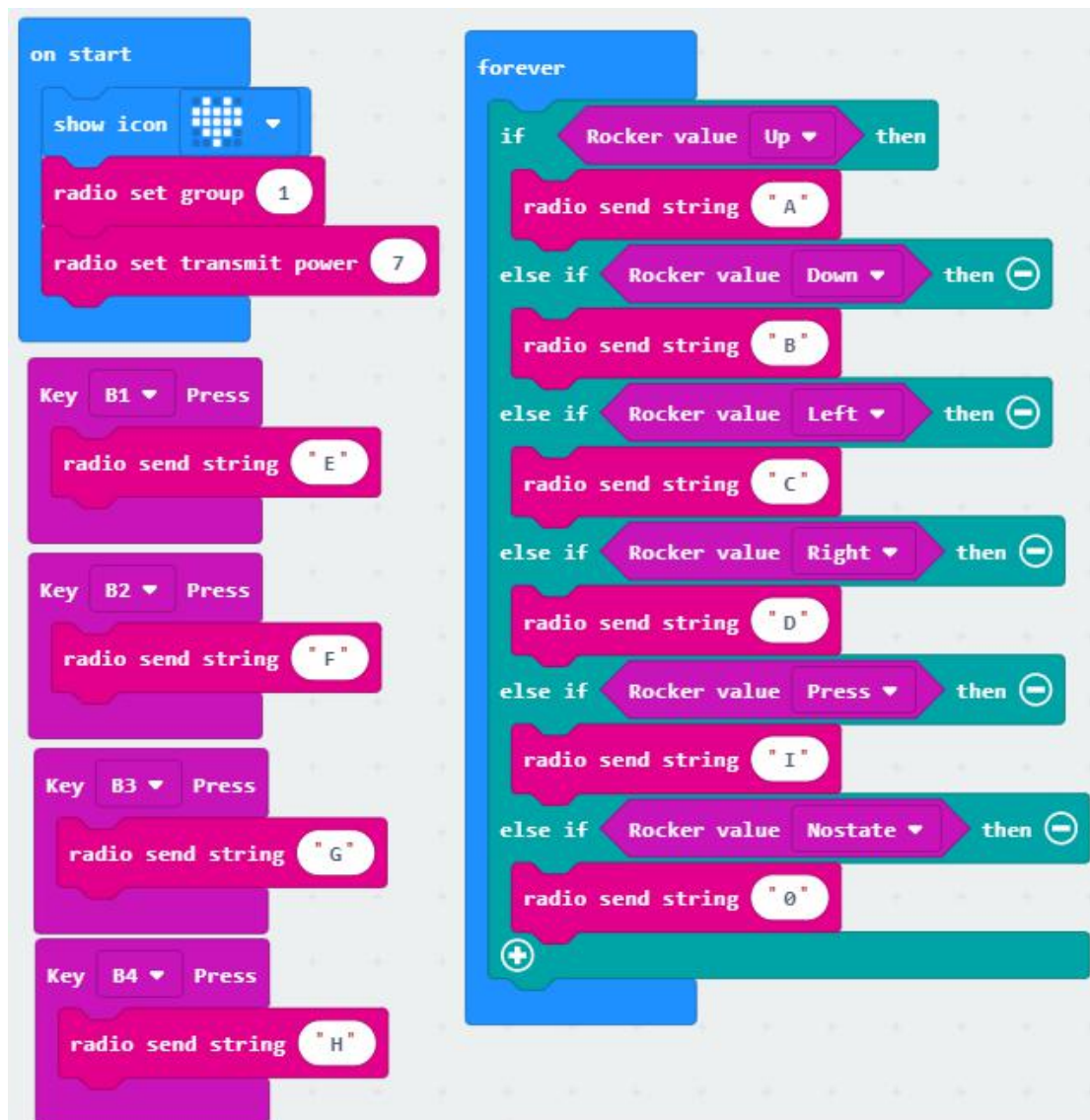## Handle joystick control program:

After adding the https://github.com/lzty634158/GHBit package, we can see the Yahboom handle blocks in the left side of the building block, as shown below:
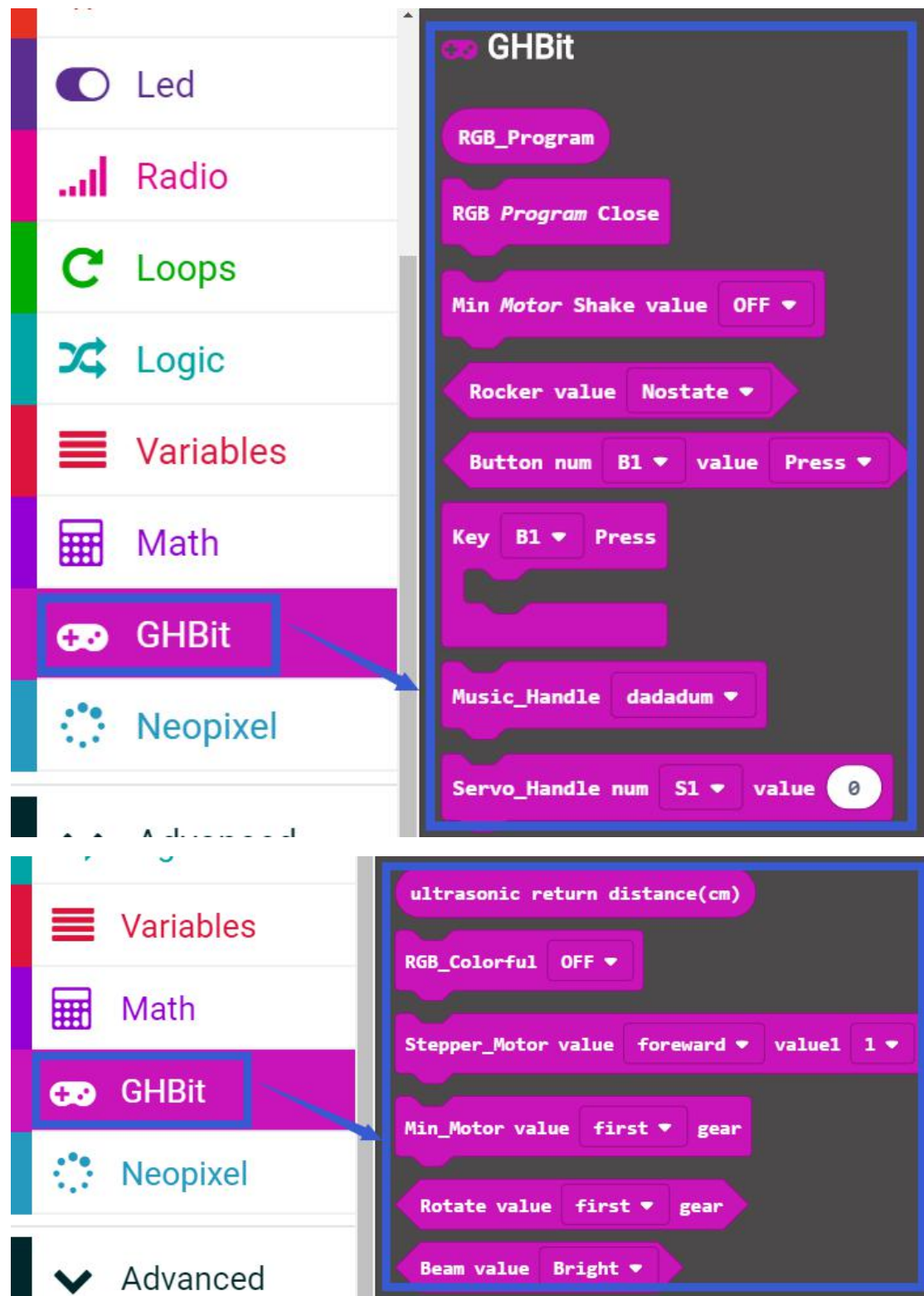
Next, we started to write the joystick control program. After writing, we need to download the program to the micro:bit board of the handle.
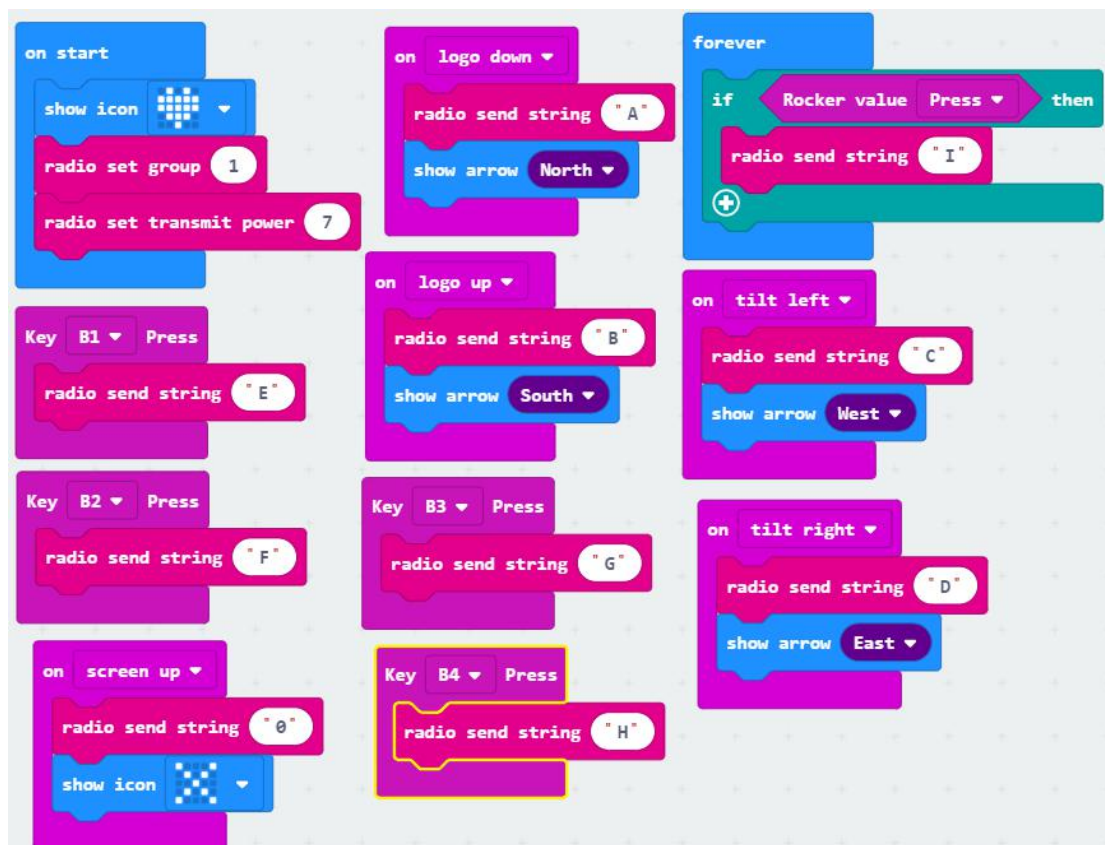

The program is shown below:

We also offer another way to control the Caterpillar tripod, that is Handle gravity sensing control.

Handle gravity sensing control program is shown below:

The above three programs, which we have provided. They are in the course directory.
If you want to use the program we provide directly, Please read the following carefully:

1. You should download the Caterpillar-tripod-program.hex in the Caterpillar tripod code folder to the micro:bit board of the Caterpillar tripod.

2. You should download the Handle-rocker-program.hex in the bit handle code folder to the micro:bit board of the handle.

3. If you want to use the handle gravity sensing to control the Caterpillar tripod, you should download the Handle-gravity-sensing-program.hex in the bit handle code folder to the micro:bit board of the handle.

After the program is downloaded, the handle and the car are powered up normally, they will be automatically paired, the two micro:bit board LED dots will show a heart on it, you can start to control.