2.6 Learning and classification

2.6 Learning and classification

1. Learning Objectives
2. Preparation before class
3. Programming Methods
4. Blocks
5. Code
6. Download code
7. Experimental Phenomena

1. Learning Objectives

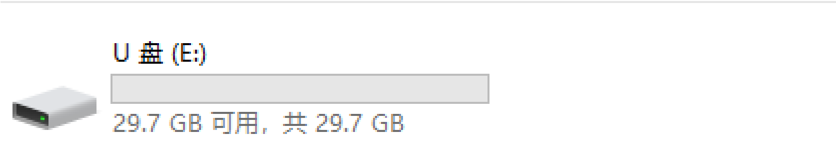
In this lesson, we learn how to make the k210 independent learning classification after the Micro:bit led dot matrix display classification serial number, independent learning can learn 3 different objects.

Note: It is recommended to use different color objects.

2. Preparation before class

1. Remove the TF card from the k210 vision module and insert it into the card reader.

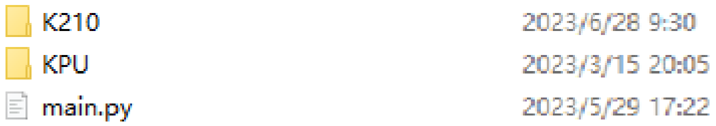
2. Plug the card reader into the computer, and wait for the computer to recognize the USB disk.



U 盘 (E:)

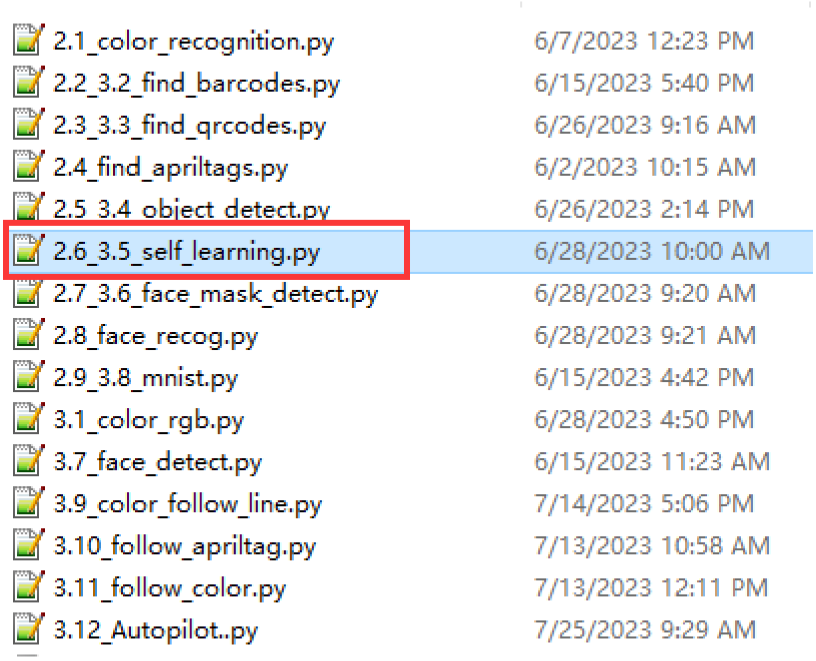
29.7 GB 可用，共 29.7 GB

3. Then, enter the TF card U disk. You will see following content.

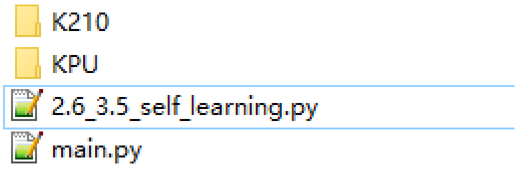


K210	2023/6/28 9:30
KPU	2023/3/15 20:05
main.py	2023/5/29 17:22

4. Go to the k210 folder, find the **2.6_3.5_self_learning.py** file from the folder and copy it to the root directory.



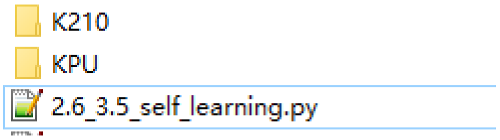
2.1_color_recognition.py	6/7/2023 12:23 PM
2.2_3.2_find_barcodes.py	6/15/2023 5:40 PM
2.3_3.3_find_qrcodes.py	6/26/2023 9:16 AM
2.4_find_apriltags.py	6/2/2023 10:15 AM
2.5_3.4_object_detect.py	6/26/2023 2:14 PM
2.6_3.5_self_learning.py	6/28/2023 10:00 AM
2.7_3.6_face_mask_detect.py	6/28/2023 9:20 AM
2.8_face_recog.py	6/28/2023 9:21 AM
2.9_3.8_mnist.py	6/15/2023 4:42 PM
3.1_color_rgb.py	6/28/2023 4:50 PM
3.7_face_detect.py	6/15/2023 11:23 AM
3.9_color_follow_line.py	7/14/2023 5:06 PM
3.10_follow_apriltag.py	7/13/2023 10:58 AM
3.11_follow_color.py	7/13/2023 12:11 PM
3.12_Autopilot..py	7/25/2023 9:29 AM



K210
KPU
2.6_3.5_self_learning.py
main.py

5. Delete the original **main.py** file.

   Then, re-name the **2.6_3.5_self_learning.py** file as the **main.py** file.



K210
KPU
2.6_3.5_self_learning.py

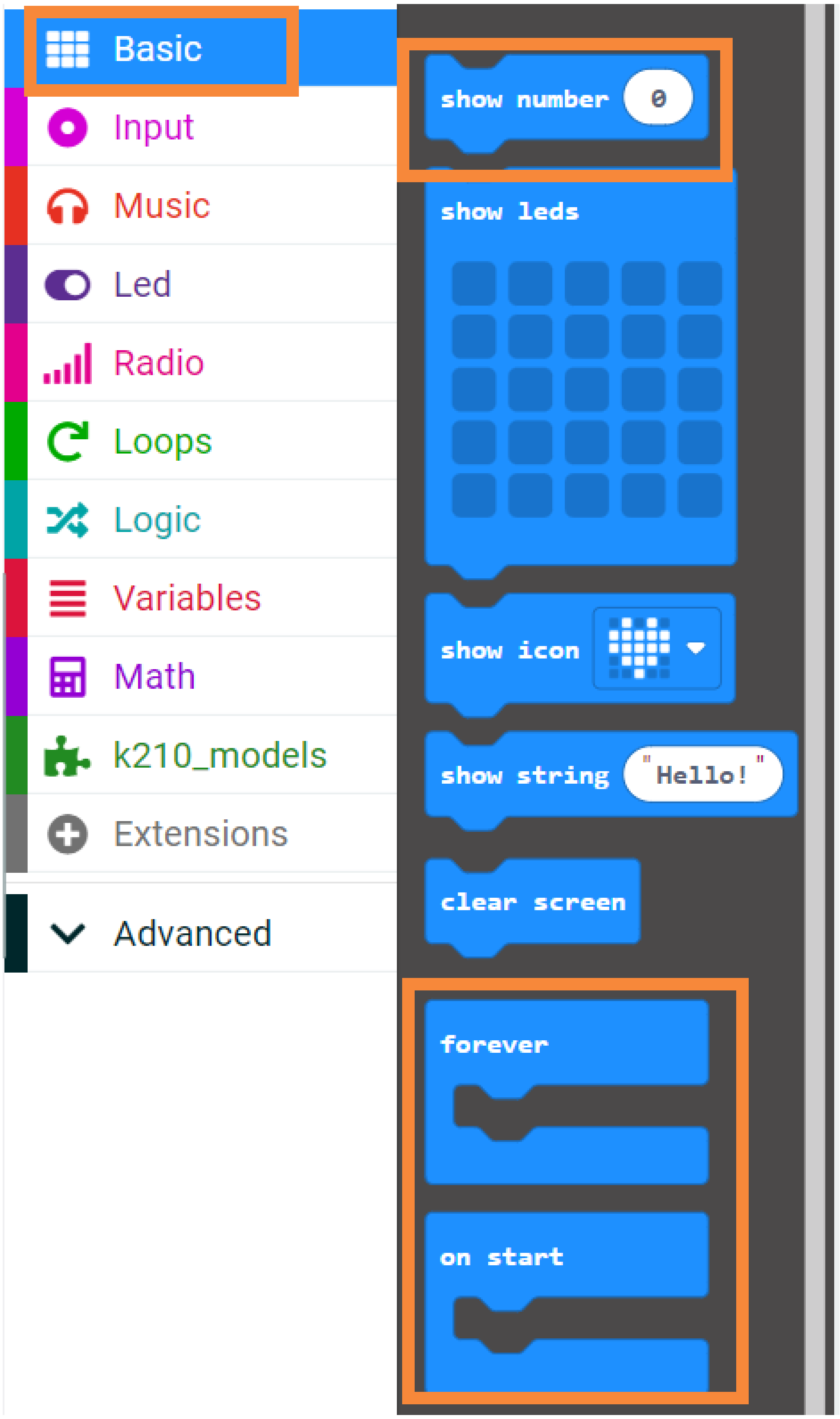
6. After re-name, pull out the card reader, remove the TF card and insert it back into the k210 vision module.

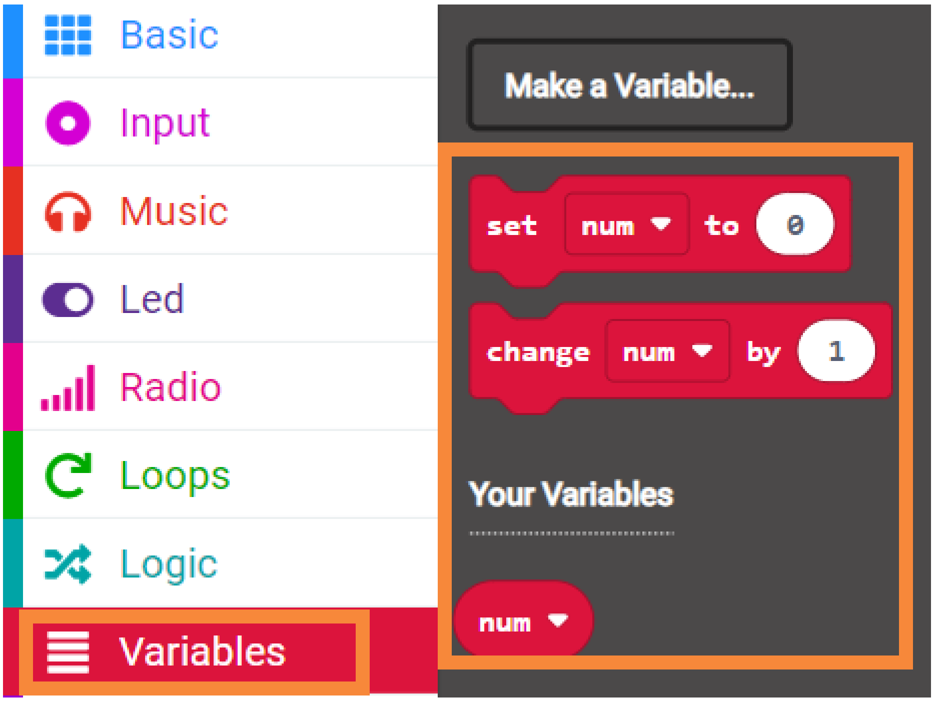
3. Programming Methods

Online programming: first copy this URL https://makecode.microbit. and enter the online programming interface.

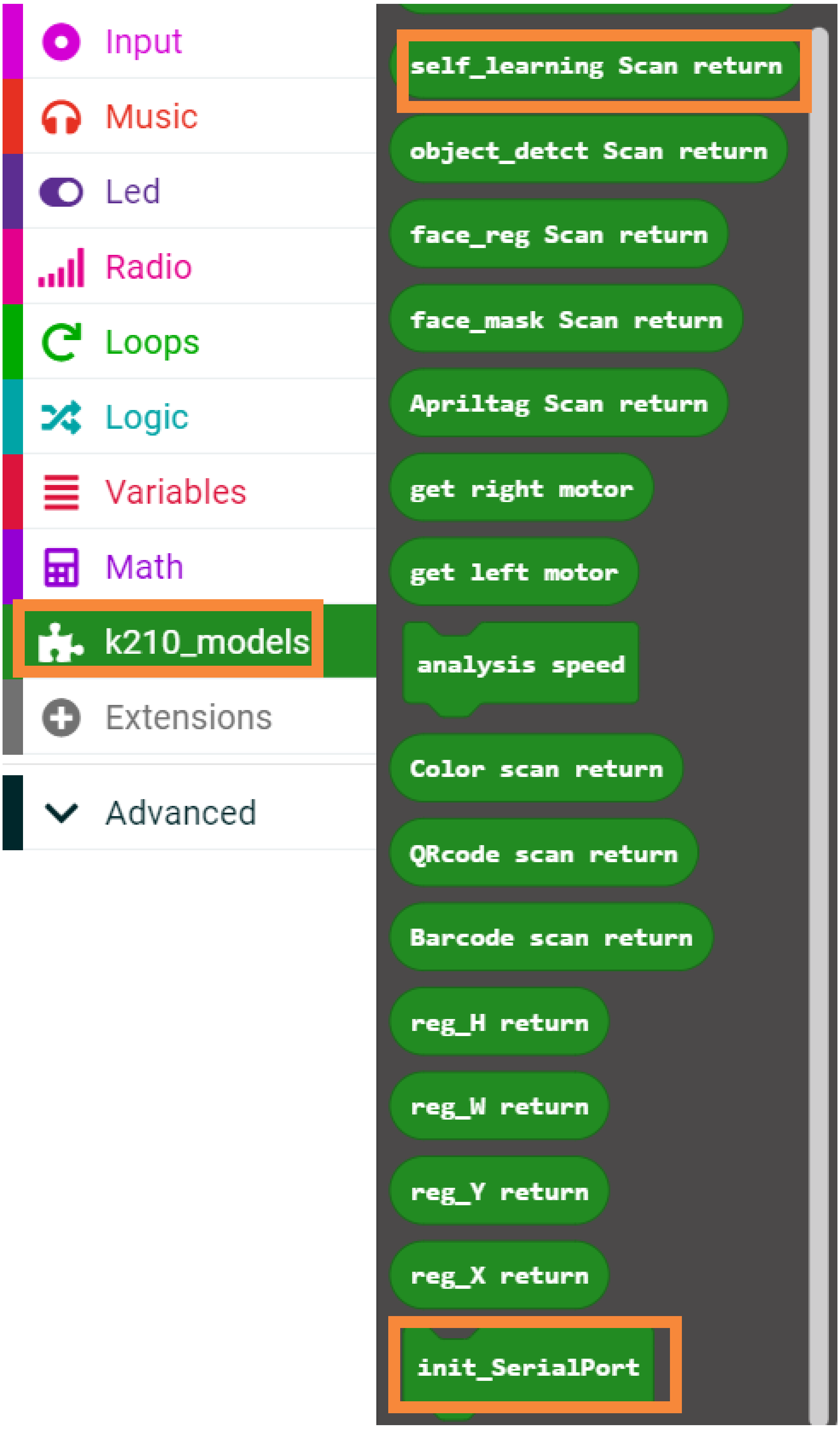
Copy the package URL: https://github.com/YahboomTechnology/K210-Module.git to the input field, click confirm to add the package, after that you can use the blocks of K210 vision module package.

4. Blocks

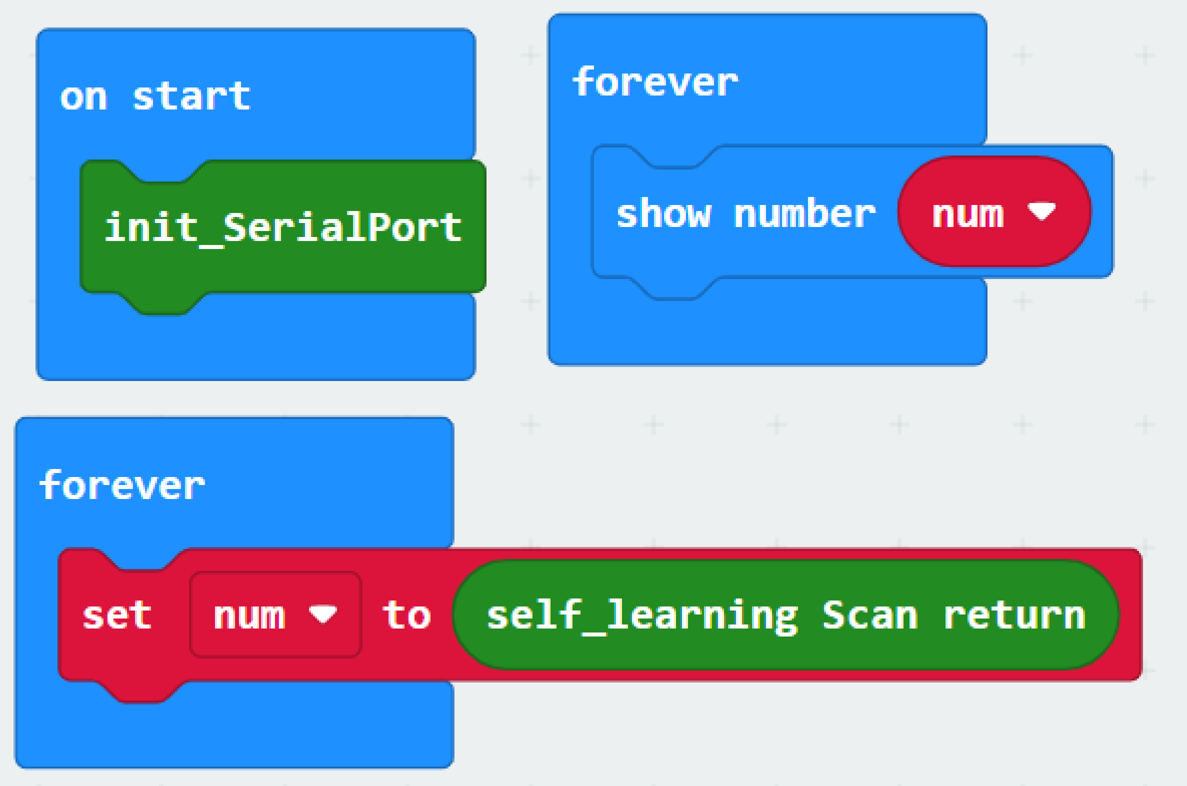

Basic
Input
Music
Led
Radio
Loops
Logic
Variables
Math
k210_models
Extensions
Advanced
show number 0
show leds
show icon
show string "Hello!"
clear screen
forever
on start



```
set num ▾ to 0
change num ▾ by 1
num ▾
```


Input
Music
Led
Radio
Loops
Logic
Variables
Math
k210_models
Extensions
Advanced
self_learning Scan return
object_detct Scan return
face_reg Scan return
face_mask Scan return
Apriltag Scan return
get right motor
get left motor
analysis speed
Color scan return
QRcode scan return
Barcode scan return
reg_H return
reg_W return
reg_Y return
reg_X return
init_SerialPort

5. Code



```
on start
    init_SerialPort

forever
    show number num

forever
    set num to self_learning Scan return
```

6. Download code

Connect the Micro:bit board to the computer via Micro USB cable, the computer will pop up a USB stick.

Then, select the **microbit-selflearn.hex** file and right click to send it to the Micro:bit U disk.

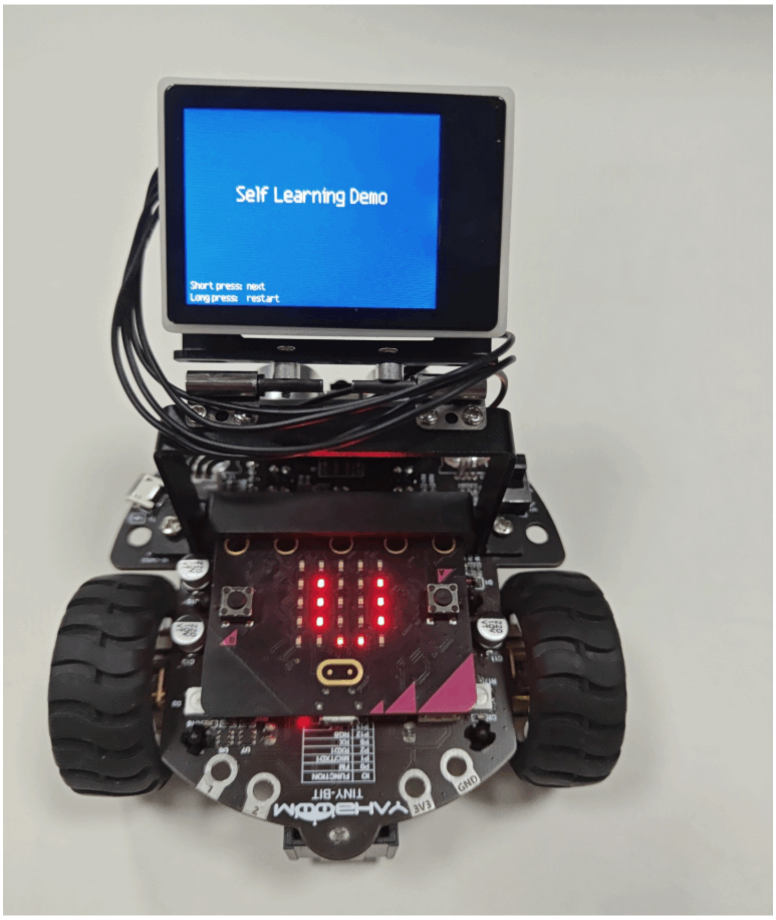
Wait until sending is complete and unplug the Micro:bit USB cable. Plug the Micro:bit board into the car.

7. Experimental Phenomena

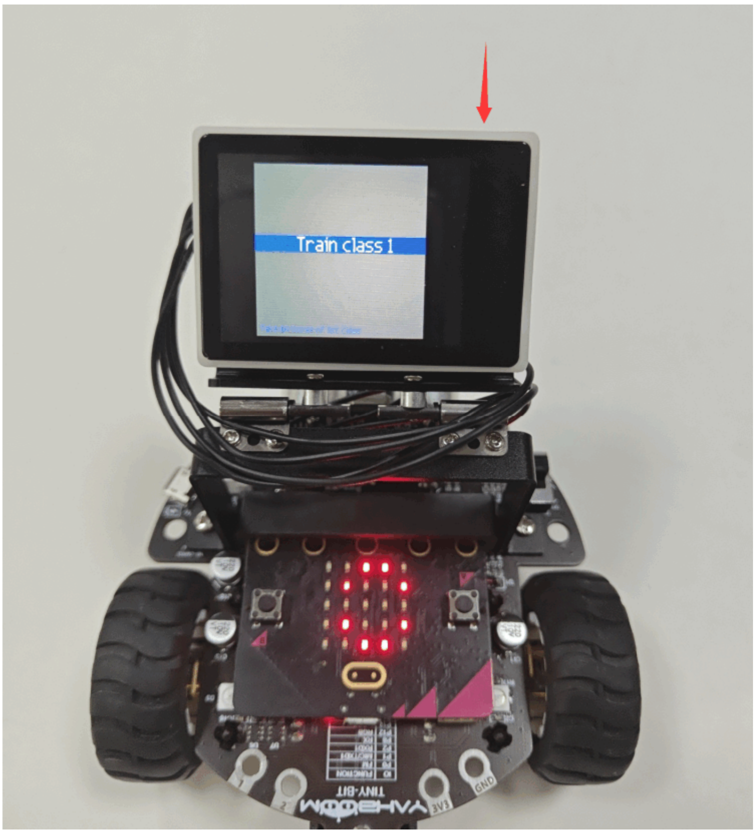
After starting the car, wait for the screen to display Self Learning Demo.

After the camera screen appears, find three different objects and train three different objects separately.

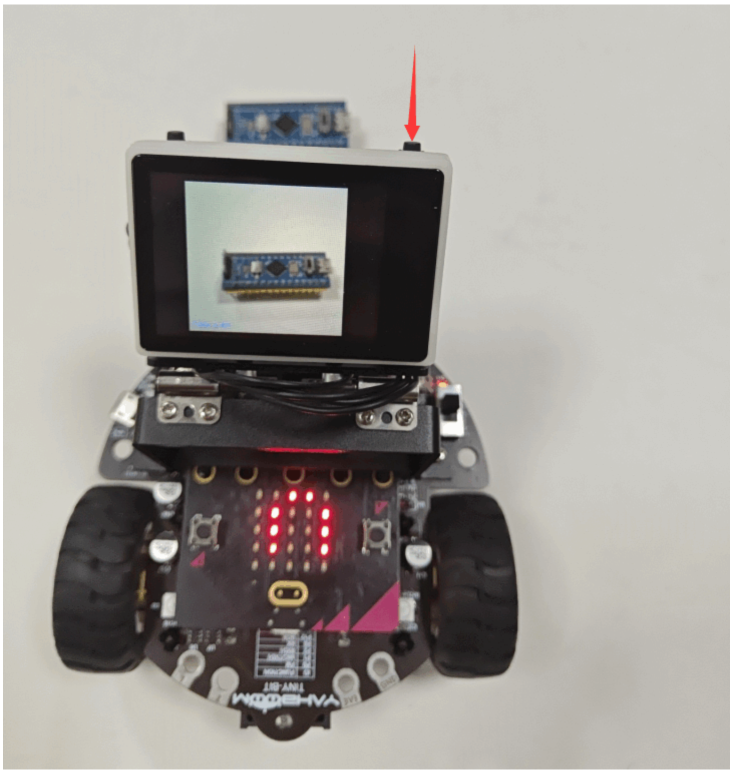
After the screen display prompts to press to enter the next state, long press to restart. At this time, press the K1 button in the upper right corner to enter the next state.



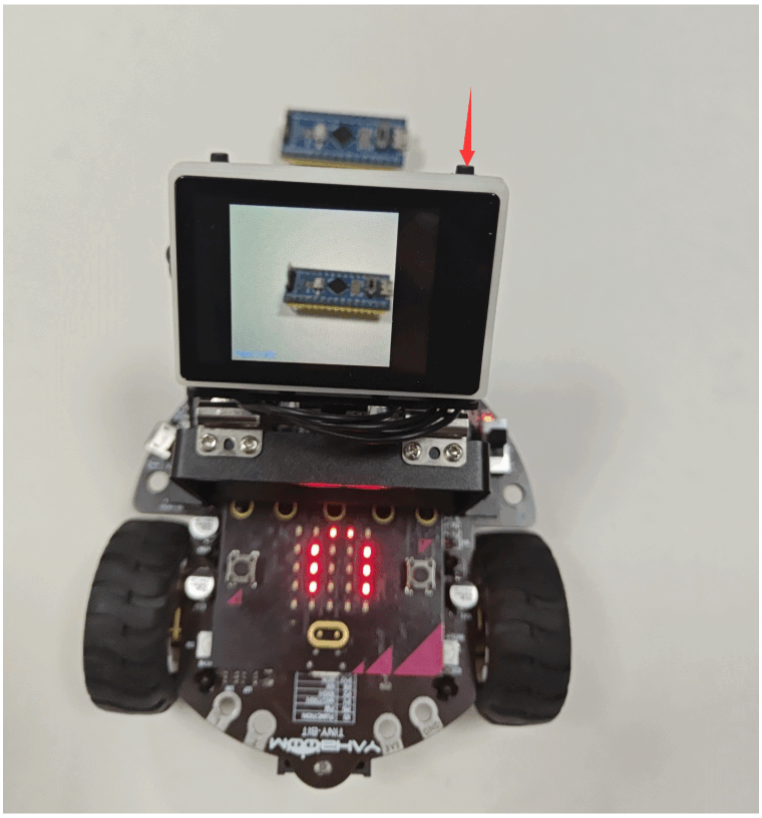
Start training the first object. Press the K1 button in the upper right corner to go to the next step and start capturing the object picture material.



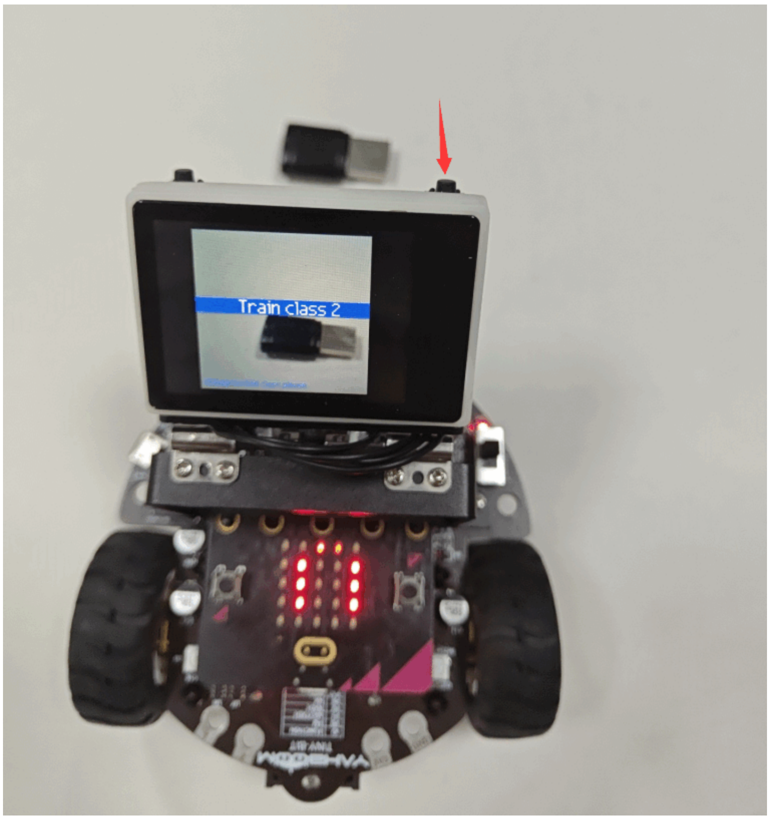
Put the first object to be recognized into the visible range of the camera, you can see that `Class 1: #P1` is displayed in the lower left corner of the screen, and if you press the K1 button, the first material picture of Class 1 will be recorded. Where Class denotes the category serial number and P denotes the picture serial number.



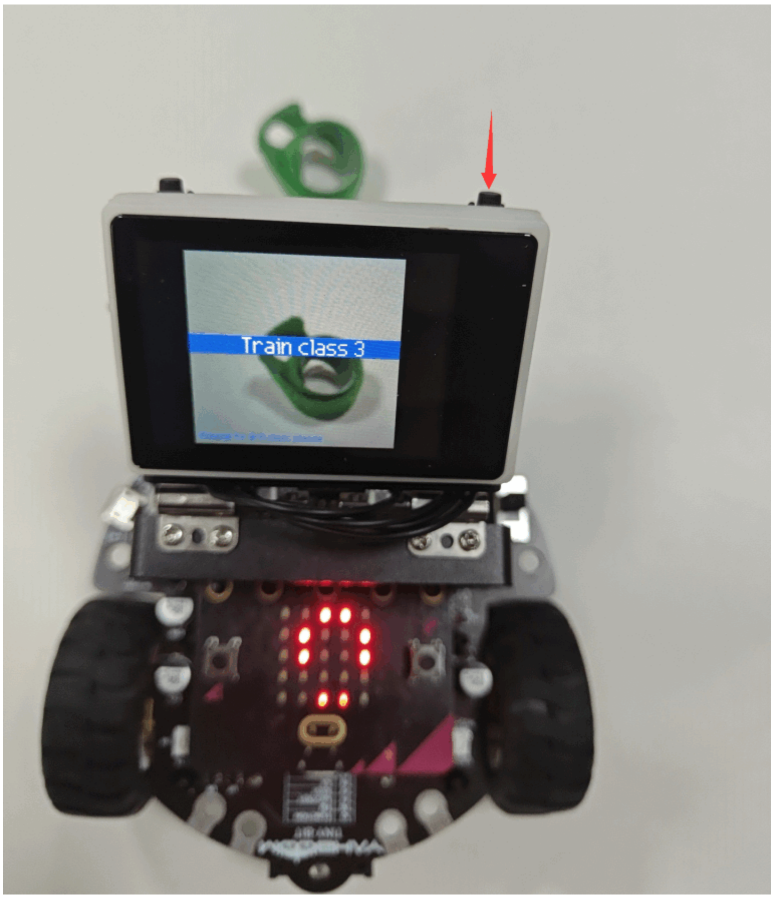
Shift the shooting angle slightly and press the BOOT key again to capture the second material picture. When five images have been captured, Category 1 capture is complete.



Next, enter the acquisition process of category 2, press the BOOT button to start the acquisition of category 2 material images, the steps are the same as the acquisition of category 1.

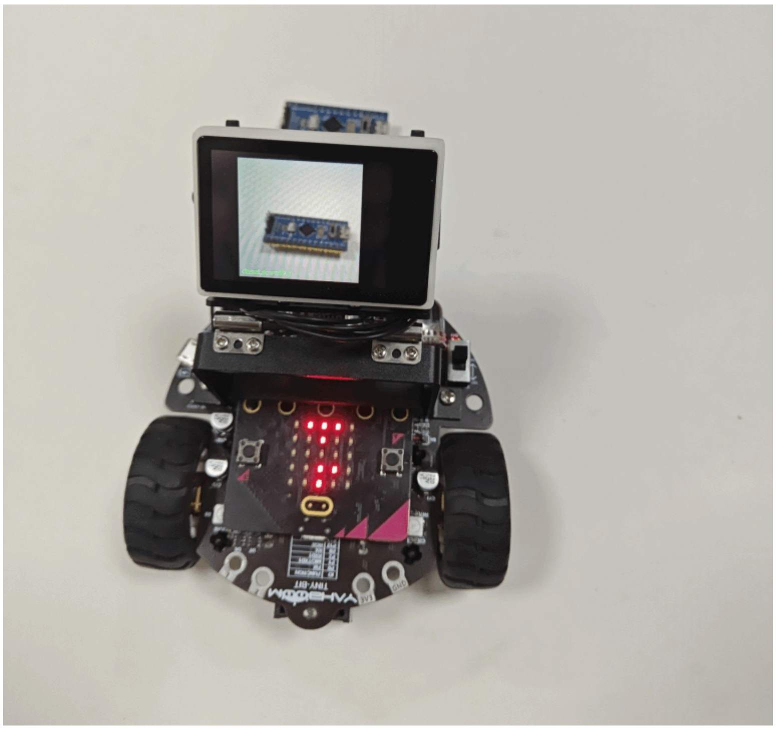


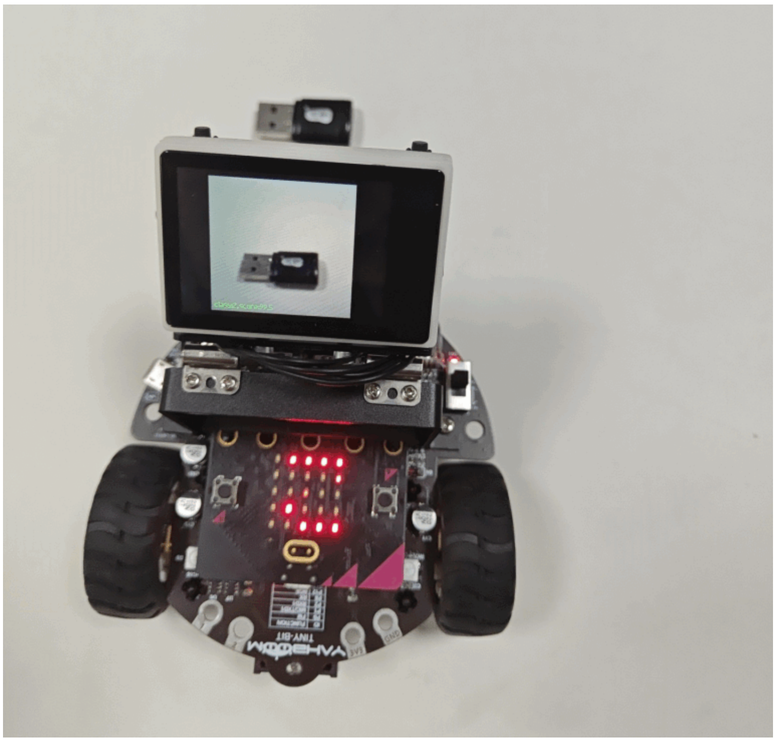
After category 2 has collected five material images, go to category 3 to collect five material images.

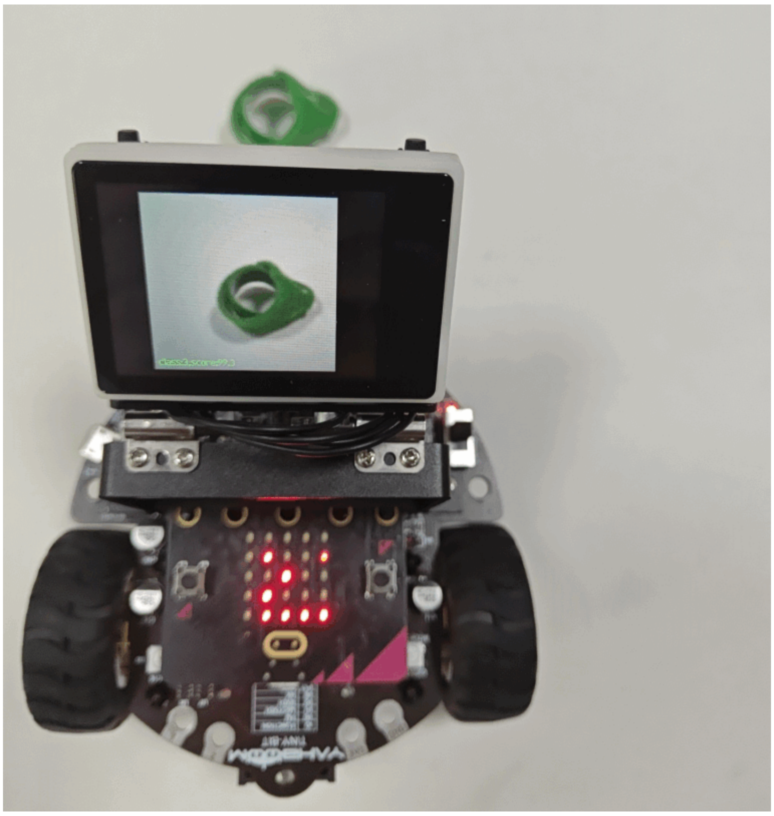


It automatically enters the classification state after finishing the collection. The display of `Classificatiion` in the middle of the screen indicates that it has entered the classification state, and the prompt disappears automatically by pressing the K1 button once more.

At this time, the camera will shoot one of the three categories just trained, the lower left corner of the screen will display the corresponding category serial number and score, at this time, the led dot matrix of the microbit board displays the classification serial number.







When no object is recognized, the led dot matrix will show -1, which means no object is recognized. If the recognition is not good, or the image acquisition is wrong, you can press reset, or long press the K1 button to start again.