

# 3.4 Object detection

---

## 3.4 Object detection

1. Learning objectives
2. Preparation before the class
3. Programming Methods
4. Blocks
5. Assortment blocks
6. Download code
7. Experimental phenomena

## 1. Learning objectives

---

In this lesson, we will realize how to let the K210 vision module recognize specific objects. There are 20 kinds of objects, namely: "airplane", "bicycle", "bird", "ship", "bottle", "bus", "car", "cat", "chair", "cow", "dining table", "dog", "horse", "motorcycle", "person", "potted plant", "sheep", "sofa", "Train", "TV screen".

Here we choose 5 kinds of objects as examples.

When a bird is recognized, music will be played;

When a car is recognized, the searchlights will light up in color order;

When the chair is recognized, the image of the chair will be displayed on the microbit board;

When a motorcycle is recognized, the car will move forward for one second, backward for one second, turn left for one second, and turn right for one second;

When a person is recognized and the car is within 20cm of the person, the car will move back for two seconds, then stop.

## 2. Preparation before the 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.



3. The computer recognises it and enters into the TF card.

K210	2023/6/28/周三 9:36
KPU	2023/4/13/周四 16:30
main.py	2060/1/1/周四 0:00

4. Go to the k210 folder, find the **2.5\_3.4\_object\_detect.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	8/24/2023 3:36 PM
 KPU	8/24/2023 3:36 PM
 2.5_3.4_object_detect.py	7/25/2023 9:29 AM
 main.py	8/24/2023 5:22 PM

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

Then, re-name the **2.5\_3.4\_object\_detect.py** file as the **main.py** file.

 K210	8/24/2023 3:36 PM
 KPU	8/24/2023 3:36 PM
 2.5_3.4_object_detect.py	7/25/2023 9:29 AM

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.org>. and enter the online programming interface.

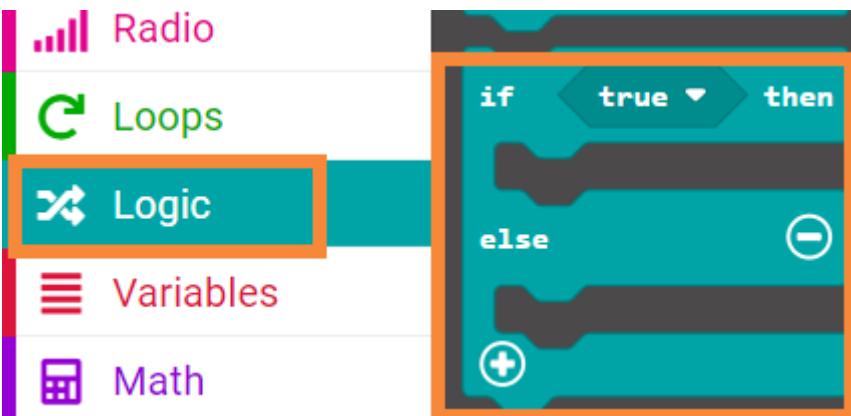
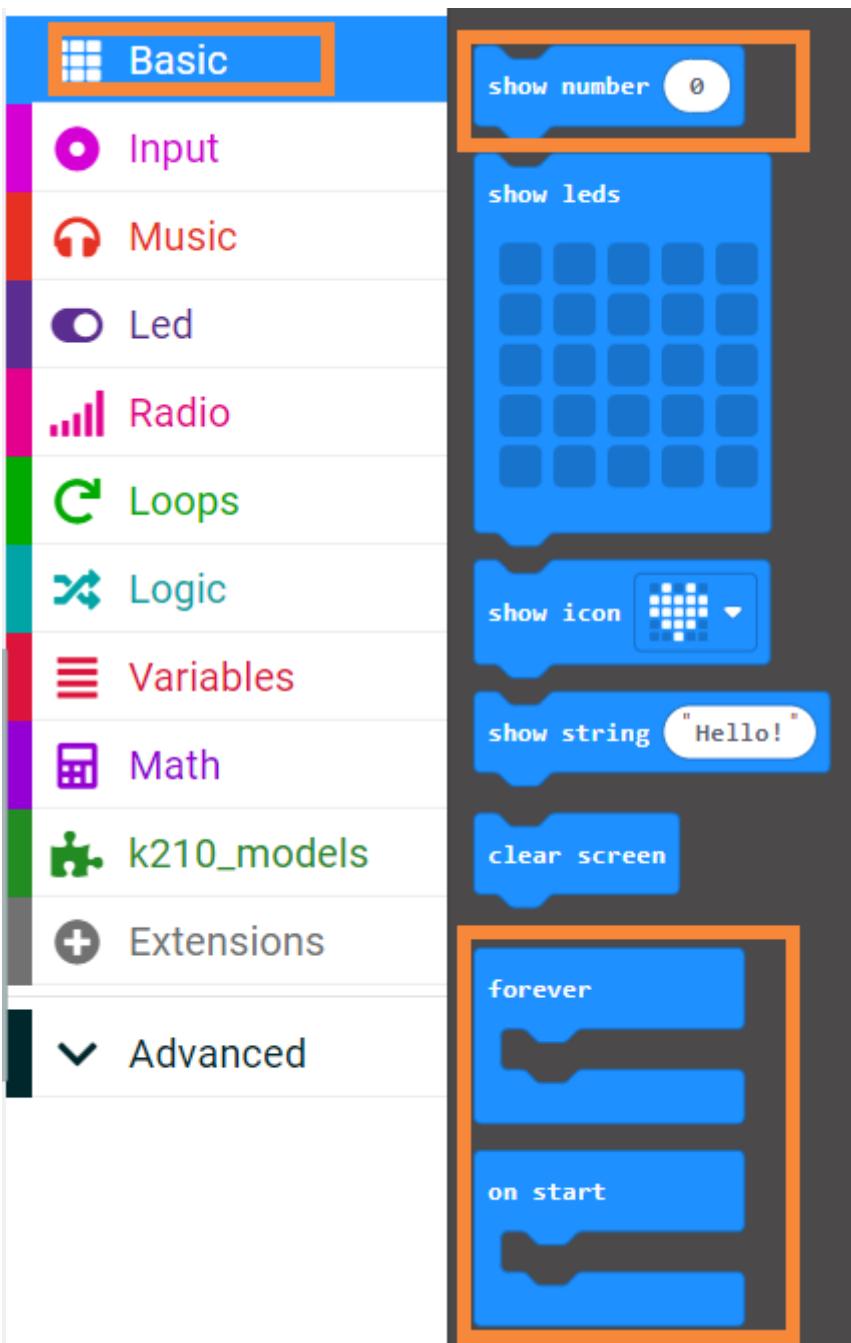
Click Extension, copy the package URL: <https://github.com/YahboomTechnology/K210-Module.git> to the input field, click Confirm to add package,

Click Extension again, copy the package URL: <https://github.com/YahboomTechnology/Tiny-bitLib> to the input field, click Confirm to add the package.

Finally you can use the K210 Vision Module package and Tinybit's building blocks.

## 4. Blocks

---



Search...



- Basic
- Input
- Music
- Led
- Radio
- Loops
- Logic
- Variables
- Math
- k210\_models
- Tinybit
- Neopixel
- Extensions
- Advanced
- Functions
- Arrays

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

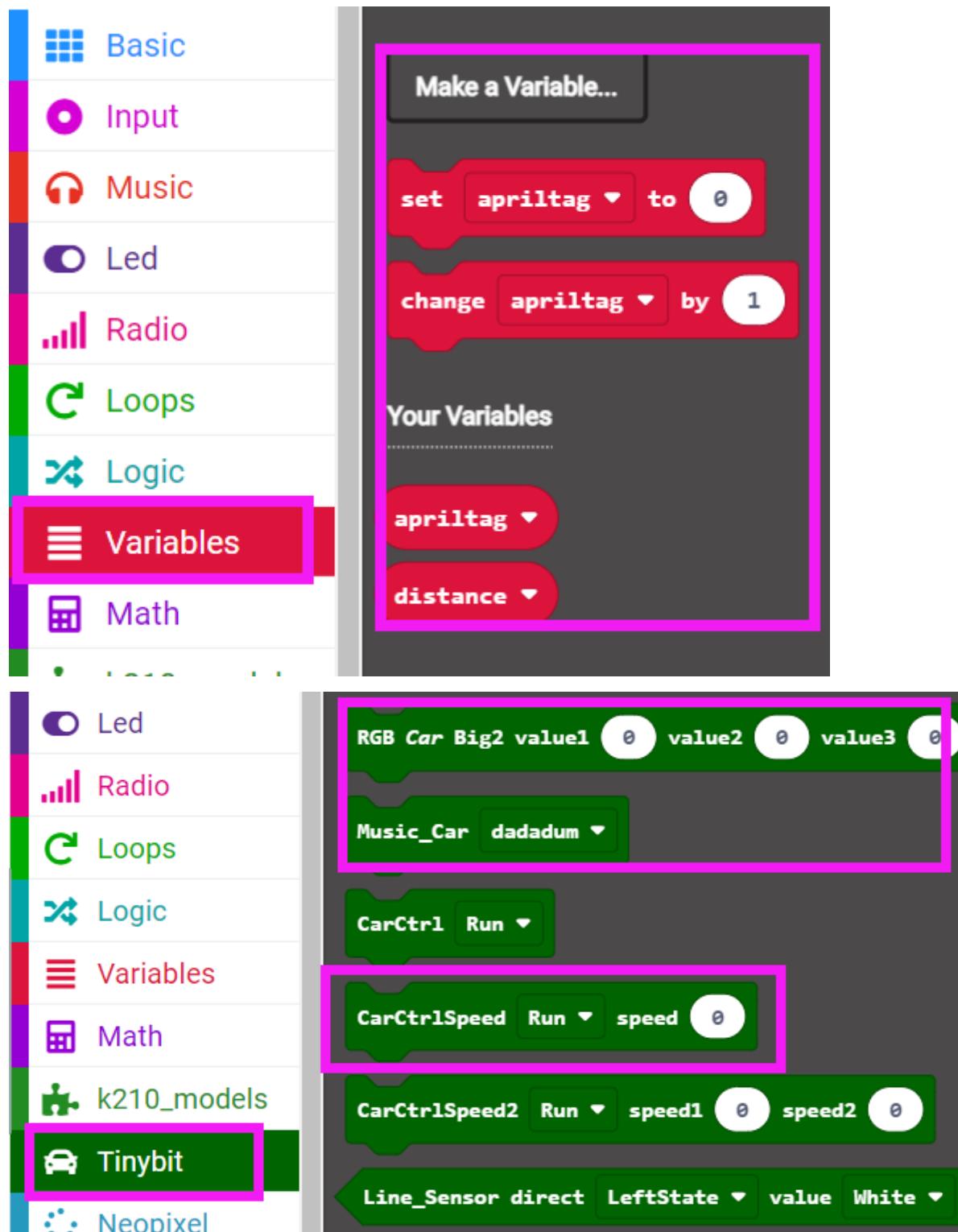
reg\_H return

reg\_W return

reg\_Y return

reg\_X return

init\_SerialPort



## 5. Assortment blocks

```

on start
  init_SerialPort
  show number 4
  pause (ms) 1000
  clear screen

forever
  if object_detct Scan return = "02" then
    clear screen
    Music_Car dadadum
  else if object_detct Scan return = "06" then
    clear screen
    RGB_Car Big2 value1 0 value2 255 value3 0
    pause (ms) 500
    RGB_Car Big2 value1 255 value2 0 value3 0
    pause (ms) 500
    RGB_Car Big2 value1 0 value2 0 value3 255
    pause (ms) 500
    RGB_Car Big2 value1 0 value2 0 value3 0
  else if object_detct Scan return = "08" then
    show leds
    [grid of 16 LEDs]
    pause (ms) 2000
    clear screen
  else if object_detct Scan return = "13" then
    clear screen
    CarCtrlSpeed Run speed 70
    pause (ms) 1000
    CarCtrlSpeed Back speed 70
    pause (ms) 1000
    CarCtrlSpeed Left speed 70
    pause (ms) 1000
    CarCtrlSpeed Right speed 70
    pause (ms) 1000
    CarCtrlSpeed Stop speed 0
  else if object_detct Scan return = "14" then
    if distance <= 20 then
      CarCtrlSpeed Back speed 60
      pause (ms) 2000
      CarCtrlSpeed Stop speed 0
      clear screen
    end
  end
  set distance to ultrasonic return distance(cm)

```

## 6. Download code

---

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

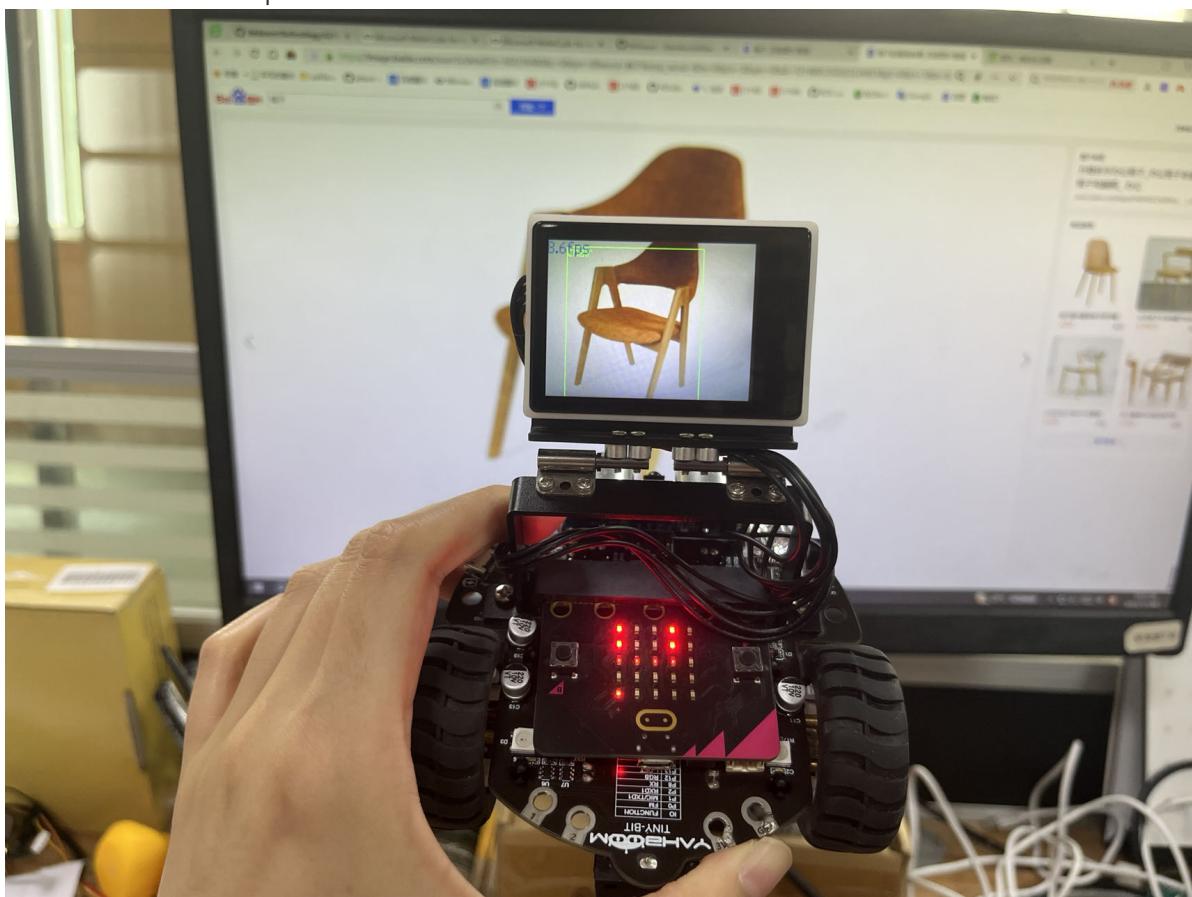
Then, select the **k210\_object\_detect.hex** code and right click to send it to the Micro:bit U disk.

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

## 7. Experimental phenomena

---

Turn on the switch of the trolley and wait for the screen to display the camera screen. After displaying the screen, point the camera at the object to be recognised (there are 20 recognisable objects, namely: "aeroplane", "bicycle", "bird", "boat", "bottle", "bus", "car", "cat", "chair", "cow", "dining table", "dog", "horse", "motorbike", "person", "potted plant", "sheep", "sofa", "train", "TV screen"). When the cart is recognised, the searchlight will light up in colour order; when the chair is recognised, the image of the chair will be displayed on the microbit motherboard; when the motorbike is recognised, the cart will move forward for one second, backward for one second, rotate to the left for one second, and rotate to the right for one second; and when a person is recognised and the cart is within 20cm of the person, the cart will move backward for two seconds and then stop.



Object detection pictures:





