Robot course7 --- Tracking

! Note: This experiment must be done indoors to reduce the interference of sunlight on the infrared sensor.

Learning goal：

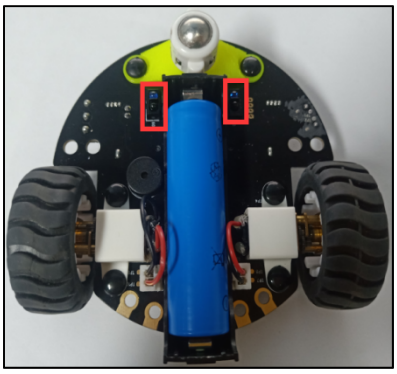
In this lesson, we will learn how to get infrared tracking sensor data, and use the detected data to go along the black line.

Principle of experimental:

The basic principle of the tracking sensor is to use the reflective nature of the object.

Our experiment is to tracking the black line. When the infrared light is emitted to the black line, it will be absorbed by the black line. When the infrared light is emitted to the other color line, it will reflected to the infrared receiver tube.

Position of tracking sensor as shown below:

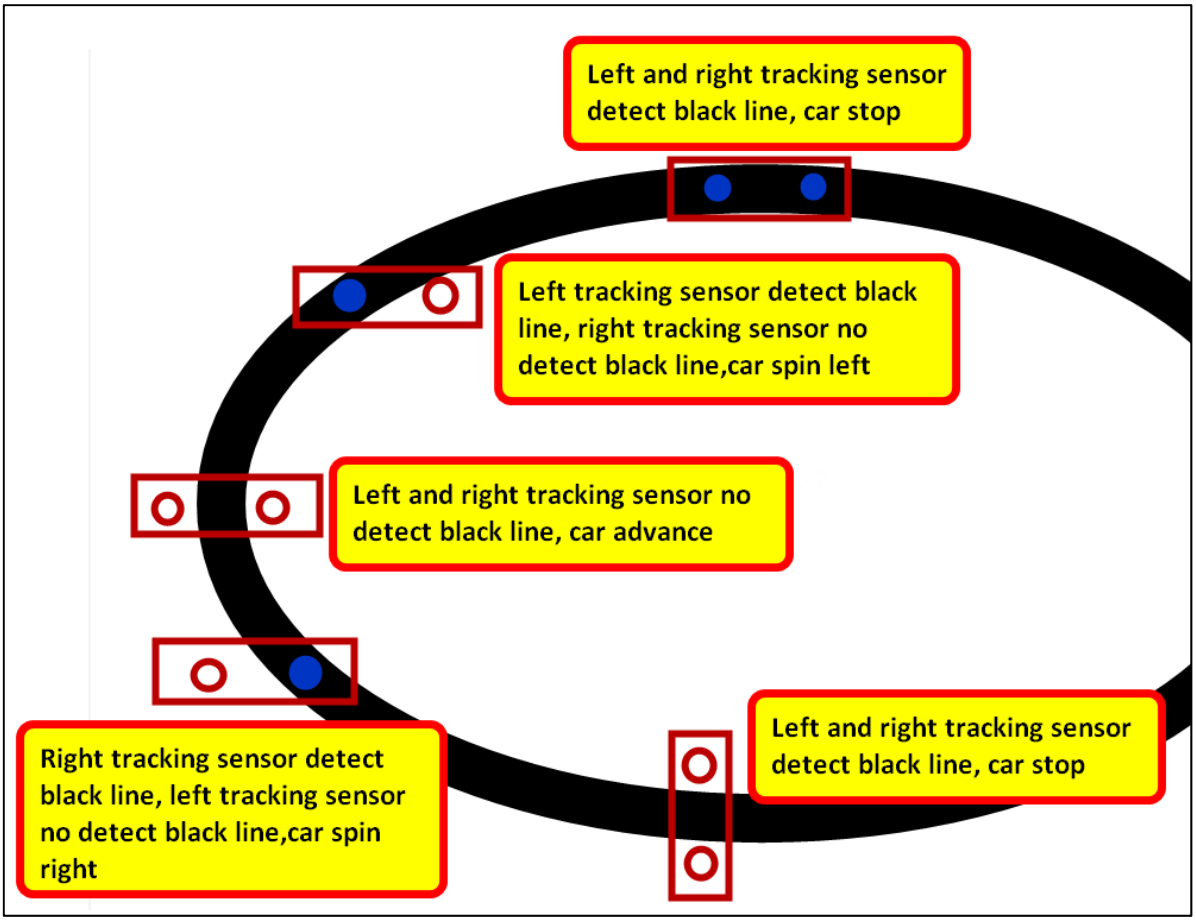


Code:

```
# -*- coding: utf-8-*-# Encoding cookie added by Mu Editor
from microbit import display, Image, pin12
import tinybit
import neopixel

np = neopixel.NeoPixel(pin12, 2)
np.clear()
tinybit.car_HeadRGB(0, 0, 0)
display.show(Image.HEART)


while True:
    left = tinybit.traking_sensor_L()
    right = tinybit.traking_sensor_R()

    if left is False and right is False:
        tinybit.car_run(90)
    elif left is True and right is False:
        tinybit.car_spinleft(80)
    elif left is False and right is True:
        tinybit.car_spinright(80)
    else:
        tinybit.car_stop()

```

1) Import the library needed for this lesson from micro:bit, display is used for dot matrix display, Image calls the built-in image, pin12 is the pin of the body colorful lights, neopixel drives the body colorful lights, and tinybit controls the car.
2) **display.show (Image.HAPPY)**: Display a smile on micro:bit dot matrix.
3) **np = neopixel.NeoPixel (pin12, 2)**: Initialize the body colorful lights. The first parameter is the pins connected to the lights, and the second parameter is the total number of RGB lights.
4) **np.clear ()**: Clear the body colorful lights.
5) **tinybit.car_HeadRGB (0, 0, 0)**: Clear the head RGB searchlight.
6) **display.show(Image.HEART)**：Display a heart on micro:bit dot matrix.
7) **left = tinybit.traking_sensor_L()**：Read the data from the left tracking sensor, return True when a black line is detected, return False when a white line is detected, and save the read result to the variable left.
8) **right = tinybit.traking_sensor_R()**：Read the data from the right tracking sensor, return True when a black line is detected, return False when a white line is detected, and save the read result to the variable right.

Analysis of car tracking status:



9) Determine the values of the variables left and right in the main loop, and analyze the movement of the car according to the figure above:

If both the left and right sensors do not detect the black line, the car moves forward;

If a black line is detected on the left and a black line is not detected on the right, the car spin left;

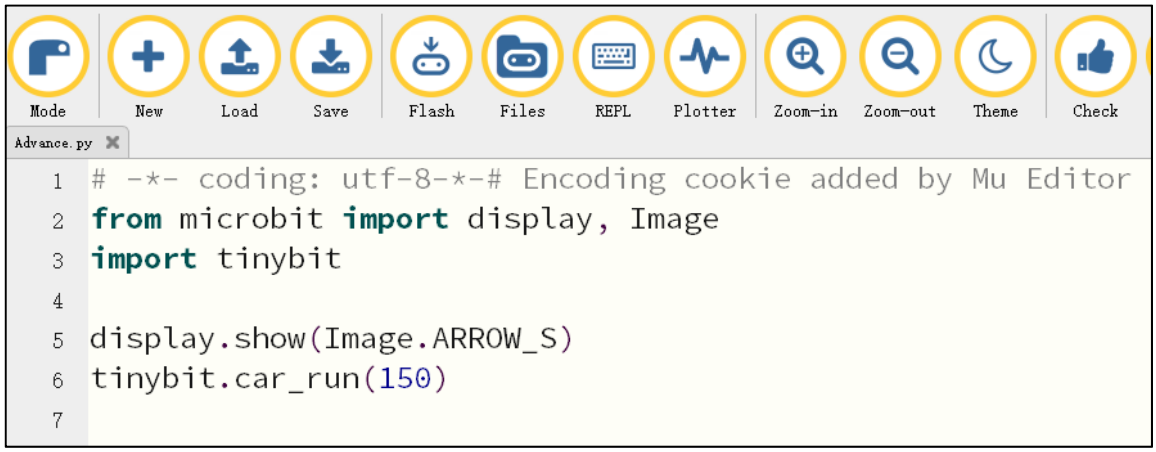
If no black line is not detected on the left and a black line is detected on the right, the car will spin right;

In other cases, if both sensors of the cart detect a black line, the car stop.

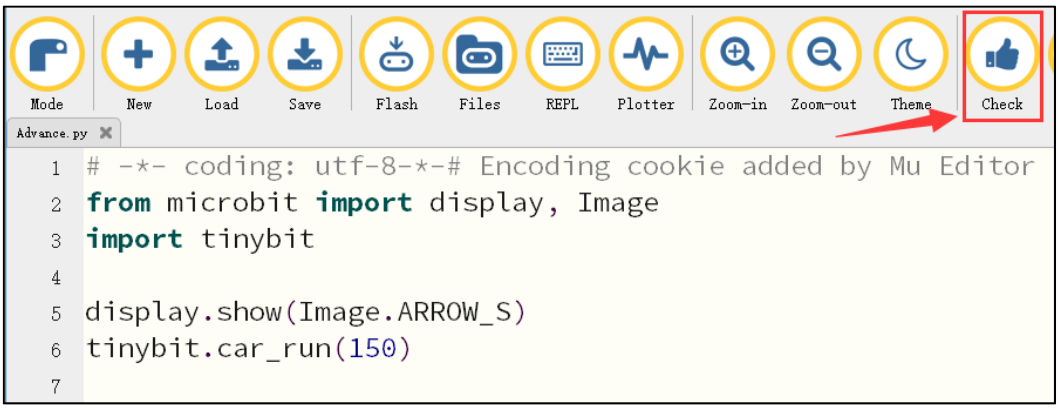
Programming and downloading：

1.You should open the Mu software, and enter the code in the edit window, , as shown below.

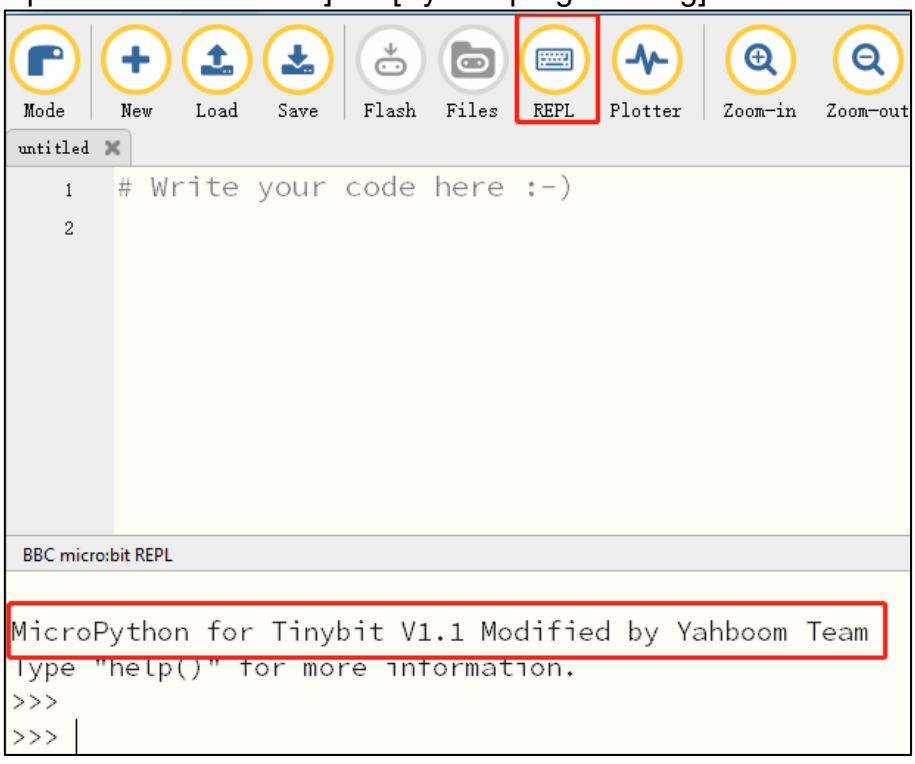
Note! All English and symbols should be entered in English, and the last line must be a space.



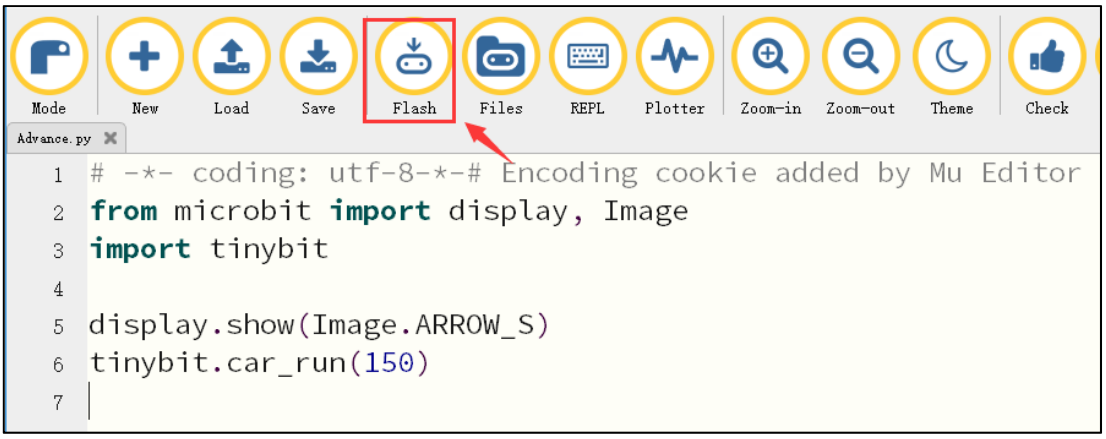
2.You can click the **"Check"** button to check if our code has an error. If a line appears with a cursor or an underscore, the program indicating this line is wrong.



3.Click **"REPL"** button,check whether the tinybit library has been downloaded. If not, please refer to the [preparation before class]---> [Python programming]



4.Click the **“Flash”** button to download the program to micro:bit board.



If the program is wrong or the experimental phenomenon is wrong after downloading, please confirm whether you have downloaded the Buildingbit libraryhex file we provided to the micro: bit board.

For the specific method of adding library files, please refer to 【1.Preparation before class】---【Python programming】

Experimental phenomena

After download is complete, open the power switch. Micro:bit will display a heart on dot matrix. Put Tiny:bit on the track with black line on white background (we can also stick a circle with black tape on white ground (or on white paper)), the car will follow the track of the track. If the car is picked up without detecting anything else, it will stop.