

## Robot course2 --- CarRun

### Learning goal:

This lesson learns how to control car advance, back, left and right by Python programming.

### Code:

```
1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image, sleep
3 import tinybit
4
5
6 while True:
7     tinybit.car_run(150)
8     display.show(Image.ARROW_S)
9     sleep(1000)
10    tinybit.car_back(150)
11    display.show(Image.ARROW_N)
12    sleep(1000)
13    tinybit.car_left(150)
14    display.show(Image.ARROW_E)
15    sleep(1000)
16    tinybit.car_right(150)
17    display.show(Image.ARROW_W)
18    sleep(1000)
19    tinybit.car_spinleft(150)
20    display.show(Image.ARROW_E)
21    sleep(1000)
22    tinybit.car_spinright(150)
23    display.show(Image.ARROW_W)
24    sleep(1000)
25    tinybit.car_stop()
26    display.clear()
27    sleep(1000)
```

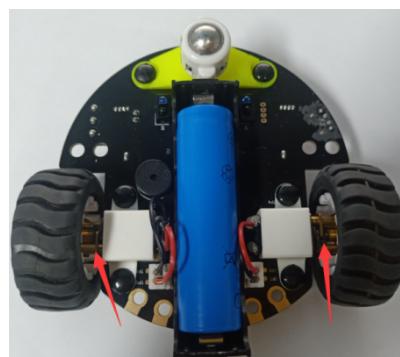
- 1) Import the library needed for this lesson from microbit: display for dot matrix display, Image calls the built-in image, tinybit controls the car;
- 2) **display.show(Image.ARROW\_S):** Display forward arrow on the micro:bit dot matrix.
- 3) **tinybit.car\_run(150):** Make car advance with 150 speed, Range of speed: 0~255.
- 4) **display.show(Image.ARROW\_N):** Display backward arrows on the micro:bit dot matrix;
- 5) **tinybit.car\_back(150):** Make car back with 150 speed, Range of speed: 0~255.
- 6) **tinybit.car\_left(150):** Make car turn left with 150 speed, left wheel stop, right wheel advance.
- 7) **tinybit.car\_right(150):** Make car turn right with 150 speed, right wheel stop, left wheel advance.

- 8) **tinybit.car\_spinleft(150)**: Make car spin left with 150 speed, right wheel advance, left wheel stop.
- 9) **tinybit.car\_spinright(150)**: Make car spin right with 150 speed, left wheel advance, right wheel stop.
- 10) **tinybit.car\_stop()**: Make car stop.
- 11) **display.clear()**: Clear dot matrix.

From the hardware interface manual, we can know that the tracking sensor is directly driven by the micro:bit IIC pin.

Category	Function	Number	Drive	The number of Drive pin	The number of connected to the controller	micro:bit
Buzzer	Buzzer	FM			P0	
Voice sensor	Voice sensor	MIC			P1	
LED light	Water light	LED-RGB			P12	
Tracking sensor	Left tracking	L-DET			P13	
	Right tracking	R-DET			P14	
Ultrasonic module	Echo pin	ECHO			P15	
	Trigger pin	TRIG			P16	
Infrared receiver	Infrared remote control	RX			P8	
I2C interface	I2C interface	SCL			P19	
		SDA			P20	
Motor	Left motor Forward	L-INA	STM8S	P06/TIM1_CH1		
	Left motor Reverse	L-INB		P07/TIM1_CH2		
	Right motor Forward	R-INA		P03/TIM1_CH3		
	Right motor Reverse	R-INB		P04/TIM1_CH4		
RGB Searching light	Red	LED-R		PC5/TIM2_CH1	SCL, SDA	P19, P20
	Green	LED-G		PD3/TIM2_CH2		
	Blue	LED-B		PD2/TIM2_CH3		

Position of motor as shown below:



## 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.**

```

Mode New Load Save Flash Files REPL Plotter Zoom-in Zoom-out Theme Check
Advance.py
1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image
3 import tinybit
4
5 display.show(Image.ARROW_S)
6 tinybit.car_run(150)
7

```

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.

```

1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image
3 import tinybit
4
5 display.show(Image.ARROW_S)
6 tinybit.car_run(150)
7

```

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

```

# Write your code here :-

```

BBC micro:bit REPL

MicroPython for Tinybit V1.1 Modified by Yahboom Team  
Type "help()" for more information.  
>>>  
>>> |

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

```

1 # -*- coding: utf-8-*# Encoding cookie added by Mu Editor
2 from microbit import display, Image
3 import tinybit
4
5 display.show(Image.ARROW_S)
6 tinybit.car_run(150)
7

```

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. The car will go forward for one second, and the micro:bit dot matrix will display a forward arrow --> the car will go backward for one second, the dot matrix will display a backward arrow ---> the car will turn left for one second, the dot matrix displays the left arrow ---> the car turn right for one second, the dot matrix displays the right arrow ---> the car will spin left for one second, the dot matrix displays the left arrow ---> the car will spin right for one second, the dot matrix displays the right arrow ---> the car will stops for one second, the dot matrix clear display. And keep looping like this status.