# Yahboom Building:bit MicroPython API

Input this code to import Yahboom building:bit Micropython library：**import buildingbit**

1、**Car advance** **buildingbit.car_run(L_Speed, R_Speed, delay)**

Description of Parameter:

L_speed is left motor speed,range is 0-255.

R_speed is right motor speed,range is 0-255.

delay is delay time(ms)

Eg: buildingbit.car_run(255, 255, 1000) // The car will advance 1000ms with 255 speed

2、**Car back** **buildingbit.car_back(L_Speed, R_Speed, delay)**

Description of Parameter:

L_speed is left motor speed,range is 0-255.

R_speed is right motor speed,range is 0-255.

delay is delay time(ms)

Eg: buildingbit.car_back(255, 255, 1000) // The car will back 1000ms with 255 speed

3、**Car turn left** **buildingbit.car_left(Speed, delay)**

Description of Parameter:

speed is right motor speed,range is 0-255.

delay is delay time(ms)

!Note: car turn left is left wheel stop and right wheel advance,so we only input right speed.

Eg: buildingbit.car_left(255, 1000) // The car will turn left 1000ms with 255 speed

4、**Car turn right** **buildingbit.car_right(Speed, delay)**

Description of Parameter:

speed is left motor speed,range is 0-255.

delay is delay time(ms)

!Note: car turn right is right wheel stop and left wheel advance,so we only input left speed.

Eg: buildingbit.car_right(255, 1000) // The car will turn right 1000ms with 255 speed

5、**Car spin left** **buildingbit.car_spinleft(L_Speed, R_Speed, delay)**

Description of Parameter:

L_speed is left motor speed,range is 0-255.

R_speed is right motor speed,range is 0-255.

delay is delay time(ms)

Eg: buildingbit.car_spinleft(255, 255, 1000) // The car will spin left 1000ms with

255 speed

6、**Car spin right** **buildingbit.car_spinright(L_Speed, R_Speed, delay)**

Description of Parameter:

L_speed is left motor speed,range is 0-255.

R_speed is right motor speed,range is 0-255.

delay is delay time(ms)

Eg: buildingbit.car_spinright(255, 255, 1000) // The car will spin right 1000ms with 255 speed

7、**Car stop** **buildingbit.car_stop()**

8、**RGB search light** **buildingbit.car_HeadRGB(R_value, G_value, B_value)**

Description of Parameter:

R_value is red value,range is 0-255.

G_value is green value,range is 0-255.

B_value is blue value,range is 0-255.

Eg: buildingbit.car_HeadRGB(255, 255, 255) // The RGB search light will become white

9、**Ultrasonic module return distance** **buildingbit.ultrasonic()**

This function returns the current distance detected by the ultrasonic module(cm).

10、**Infrared obstacle avoidance** **buildingbit.avoid_sensor()**

Return value: if there are obstacles returns true, if there are no obstacles return false

! Note: When using this function, you must turn off the dot-matrix display, otherwise pin reuse will cause conflicts, and the method of turning off dot-matrix display: **microbit.display.off ()**

11、**Left tracking sensor** **buildingbit.traking_sensor_L()**

Return value: if black lines are detected returns true(Indicator light is off), if white lines are detected return false(Indicator light is on).

12、**Left tracking sensor** **buildingbit.traking_sensor_R()**

Return value: if black lines are detected returns true(Indicator light is off), if white lines are detected return false(Indicator light is on).

13、**Servo control** **buildingbit.servo(num, angle)**

Description of Parameter:

num is number of servo(1-3)

angle is angle of servo,range is 0-180.

Eg: buildingbit.servo(1, 180) // No.1 servo will rotate to 180°

14、**IR control** **buildingbit.init_IR(IR_pin)**

Description of Parameter:

IR_pin is Pins connected to the infrared receiver. **For building:bit, we need to select pin8**

Eg: buildingbit.init_IR(pin8)

15、**Return button value of IR controller** **buildingbit.get_IR(IR_pin)**

Description of Parameter:

IR_pin is Pins connected to the infrared receiver. **For building:bit, we need to select pin8**

Eg: buildingbit.get_IR(pin8)