SuperKit Python library file API

! Note: The SuperKit library adds drivers for the superbit expansion board and the Cube World to the official microbit-microPython library.

For other micro:bit APIs, please visit the official website of microbit-microPython: https://microbit-micropython.readthedocs.io/en/latest/

A. Superbit library

1) import superbit

Import **superbit_micropython** library

2) superbit.motor_control(a, b, 0)

Function: Control the motor interface on the expansion board to send PWM

Parameters:

a: Select the corresponding motor port (superbit.M1~superbit.M4)

b: PWM duty cycle (-255~255, positive value means forward rotation, negative value means reverse rotation)

For example: superbit.motor_control(superbit.M1, 255, 0)

3) superbit.motor_control_dual(a, b, c, d, 0)

Function: Control the two motors on the expansion board to send PWM at the same time

Parameters:

a: Select the corresponding first motor port 0 (superbit.M1~superbit.M4)

b: Select the corresponding second motor port (superbit.M1~superbit.M4)

c: The duty cycle of the PWM of the first motor port (-255~255, positive value means forward rotation, negative value means reverse rotation)

d: The duty cycle of the PWM of the second motor port (-255~255, positive value means forward rotation, negative value means reverse rotation)

For example: superbit.motor_control(superbit.M1, superbit.M3, 255, 255, 0)

4) superbit.servo270_V2(a, b)

Function: Control the servo interface on the expansion board to control the 270° servo

Parameters:

a: Select the servo port number (superbit.S1~superbit.S8)

b: Set the control angle (0-270)

For example: superbit.servo270_V2(superbit.S1, 120)

5) superbit.servo180(a, b)

Function: Control the servo interface on the expansion board to control the 180° servo

Parameters:

a: Select the servo port number (superbit.S1~superbit.S8)

b: Set the control angle (0-180)

For example: superbit.servo180(superbit.S1, 90)

6) superbit.stepper_control(a, b)

Function: Control the stepper motor on the expansion board

Parameters:

a: Select the servo port number (superbit.B1~superbit.B2)

b: Set the control angle (0~360)

For example: superbit.stepper_control(superbit.B1, 90)

B. Handle library ghandle

1) import ghandle

Import the handle library file

2) ghandle.B1_is_pressed()

Function: Press button B1 (red), return True when pressed, return False when released

3) ghandle.B2_is_pressed()

Function: Press button B2 (green), return True when pressed, return False when released

4) ghandle.B3_is_pressed()

Function: Press button B3 (blue), return True when pressed, return False when released

5) ghandle.B4_is_pressed()

Function: Press button B4 (yellow), return True if pressed, return False if released

6) ghandle.rocker(state)

Function: Detect the direction of the joystick, return True if it is consistent with the parameter, return False if it is inconsistent

Parameter: state: ghandle.up (up), ghandle.down (down), ghandle.left (left), ghandle.right (right), ghandle.pressed (pressed), ghandle.noState (no operation)

7) ghandle.get_value_x()

Function: Return the analog value of the joystick X channel, the value will change when moving left and right, increase to the left, and decrease to the right

8) ghandle.get_value_y()

Function: Return the analog value of the joystick Y channel, the value will change when moving up and down, decrease up, and increase down

C. Cube World

import WOM_Sensor_Kit

Import **superbit_micropython** library

1) RGB module

WOM_Sensor_Kit.WOM_rgb(1023,1023,1023)

Parameter definition: WOM_rgb(red, green, blue) Color range 0-1023, fixed connection P14, P13, P12 pins, cannot be changed.

2) Button module

WOM_Sensor_Kit.WOM_button(pin0)

Parameter definition: WOM_button(pin) Press to return 1, otherwise return 0

3) Rocker module

WOM_Sensor_Kit.WOM_rocker(pin0,pin1,WOM_Sensor_Kit.WOM_up)

WOM_Sensor_Kit.WOM_rocker(pin0,pin1,WOM_Sensor_Kit.WOM_down)

WOM_Sensor_Kit.WOM_rocker(pin0,pin1,WOM_Sensor_Kit.WOM_left)

WOM_Sensor_Kit.WOM_rocker(pin0,pin1,WOM_Sensor_Kit.WOM_right)

Parameter definition: WOM_rocker(pin X, pin Y, status), returns 1 or 0, X and Y can be modified to other pins, pay attention to keep the wiring consistent with the program.

4) Photosensitive module

WOM_Sensor_Kit.WOM_light(pin0)

Parameter definition: WOM_light(pin X) returns analog value 0-1023. The stronger the light, the smaller the value. X can be changed to other pins. Please make sure that the wiring is consistent with the program.

5) Infrared module

WOM_Sensor_Kit.WOM_ir(pin0)

Parameter definition: WOM_ir(pin X). If there is an obstacle, it returns 0. If there is no obstacle, it returns 1. X can be changed to other pins. Please make sure that the wiring is consistent with the program.

6) Temperature and humidity module

WOM_Sensor_Kit.WOM_dht11(pin0, WOM_Sensor_Kit.WOM_temp_C)

WOM_Sensor_Kit.WOM_dht11(pin0, WOM_Sensor_Kit.WOM_temp_F)

WOM_Sensor_Kit.WOM_dht11(pin0, WOM_Sensor_Kit.WOM_humidity)

Parameter definition: WOM_dht11(pin, read data), X can be changed to other pins, pay attention to keep the wiring consistent with the program.

Return the corresponding read data, including WOM_temp_C, WOM_temp_F, WOM_humidity

The interval between each acquisition must be more than 1s.

7) Ultrasonic module

WOM_Sensor_Kit.WOM_ultrasonic(pin3,pin2)

Parameter definition: WOM_ultrasonic(pin Echo, pin Trig), returns ultrasonic data. Echo and Trig can be changed to other pins, pay attention to keep the wiring consistent with the program.

8) Digital tube module

WOM_Sensor_Kit.WOM_init_display(1)

Parameter definition: WOM_init_display(brightness level) Range 0-7, where 0 is the largest, 1 is the smallest, 1-7 is increasing

WOM_Sensor_Kit.WOM_display(100)

Parameter definition: WOM_display(digital) Display digital range 0-9999, fixed connection SCL and SDA pins

9) Human infrared sensor module

WOM_Sensor_Kit.WOM_pir(pin0)

Parameter definition: WOM_pir(pin X) Someone returns 1, X can be modified to other pins, pay attention to keep the wiring consistent with the program.

10) Color recognition module

WOM_Sensor_Kit.WOM_color(WOM_Sensor_Kit.WOM_red)

WOM_Sensor_Kit.WOM_color(WOM_Sensor_Kit.WOM_green)

WOM_Sensor_Kit.WOM_color(WOM_Sensor_Kit.WOM_blue)

Parameter definition: WOM_color(acquired color), returns the read color RGB value, the returned value range is [0,255], fixed connection SCL and SDA pins

11) Servo

WOM_Sensor_Kit.WOM_servo360(pin10,360)

Parameter definition: WOM_servo360(pin X, angle), drives the servo connected to pin P10 to rotate to 360 degrees, X can be changed to other pins, pay attention to keep the wiring consistent with the program. The angle range is 0-360.