

4.7 Button control buzzer

1. Learning goal:

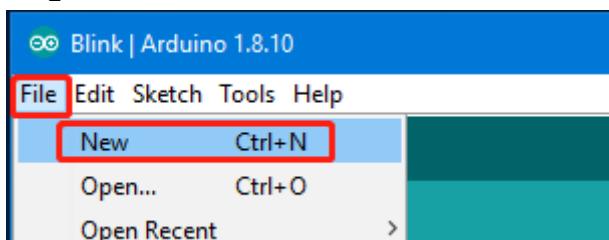
Learn about how to control buzzer by key.

2. Experimental phenomena:

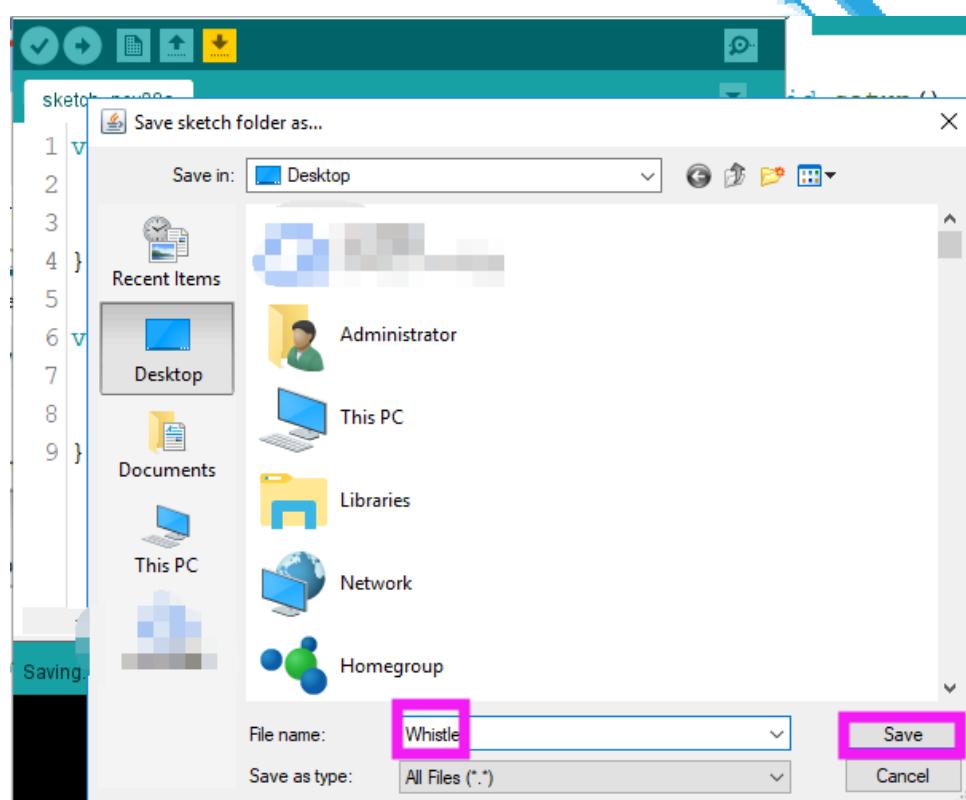
When button is pressed, buzzer will sound.

3. Create new project

3.1 Click 【File】 --> 【New】 .



3.2 Press **Ctrl+S** to save and rename Whistle. As shown below.



3.3 We can see that there is a Serial folder with **Whistle.ino** on the computer desktop.

3.4 We will Whistle.ino as shown below.

```

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
  
```

The setup() function only runs once when the car is turned on or when the reset button is pressed, and the program for initializing the relevant content can be written;

The loop() function is the main loop function of the car and most of the data processing and logic processing are done in this function.

4. Programming

4.1 From the hardware manual, we can know that the button K1 connect to Pin10 of UNO.

```
#define BUZZER 10      //Define buzzer pin
#define KEY_PIN 8       //Define key pin
#define LED_PIN 5       //Define LED(D9) pin
```

4.2 New create whistle function, Since Omniduino car adopts a passive buzzer, it requires a certain high and low frequency to drive it.

```
void whistle()
{
    for (int i = 0; i < 100; i++)
    {
        digitalWrite(BUZZER, HIGH); //sound
        delay(3);
        digitalWrite(BUZZER, LOW); //no sound
        delay(1);
    }
}
```

4.3 Initialize the pin mode in the setup() function, the button is in pull-up input mode, and the LED is in output mode.

Because from the hardware manual, we can see that the function button is pressed low, the release is high, that is, the low level is valid, so the pin voltage is pulled high by default to work properly.

```
void setup() {
    // put your setup code here, to run once:
    pinMode(KEY_PIN, INPUT_PULLUP); //Set the button pin to pull-up input mode
    pinMode(LED_PIN, OUTPUT);      //Set the LED pin to output mode
    pinMode(BUZZER, OUTPUT);       //Set the buzzer pin to output mode
}
```

4.4 In the loop() main loop function to detect whether the button is pressed, if button is pressed, the buzzer sound.

```
void loop() {
    // put your main code here, to run repeatedly:
    // key scan
    keyscan();
    if (button_press)
    {
        whistle();
        button_press = false; //set key status to false
    }
}
```

5. Compiling and downloading code

5.1 After the code is written, press Ctrl+S to save, then click the “√” button to compile. If there

is no problem, click “→” to upload (the car must be connected to the computer via the USB cable).

```

File Edit Sketch Tools Help
LED
10 /*
11 //Define LED light(D9)pin
12 #define LED_PIN 5
13
14 void setup() {
15 // put your setup code here, to run once:
16 // set LED pin to output mode
17 pinMode(LED_PIN, OUTPUT);
18 }
19
20 void loop() {
21 // put your main code here, to run repeatedly:
22 digitalWrite(LED_PIN, LOW); //LED is on
23 delay(500);
24 digitalWrite(LED_PIN, HIGH); //LED is off
25 delay(500);
26 }

```



5.2 If the compilation passes normally, but the following error occurs during uploading, the reason may be that the wrong serial port or the serial port is occupied.

```

An error occurred while uploading the sketch
Sketch uses 924 bytes (2%) of program storage space. Maximum is 32
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039
An error occurred while uploading the sketch
avrduude: ser_open(): can't open device "\.\COM32": The system can

```

Solution: Open the device manager to see if there is a serial port with CH340 tag. If not, please restart the Omniduino car, then, re-plug the USB cable or replace a USB cable; If there is a serial port number, we need to close the other serial port or assistant software, avoid serial port occupation, and then re-select the serial port to ArduinoIDE [Tool]-->[Port].