

4.2 LED Breathing Light

1. Learning goal:

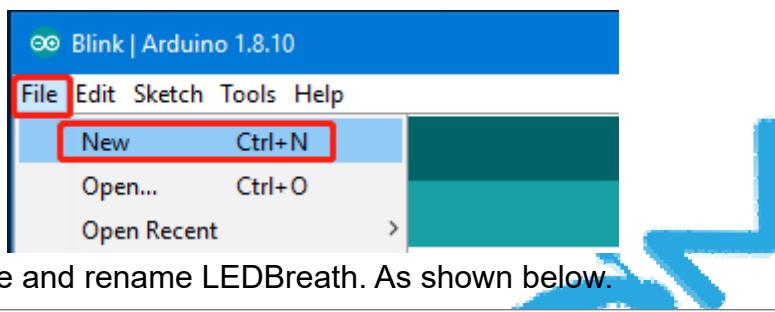
Configure the IO port analog value output, LED light breathing effect.

2. Experimental phenomena:

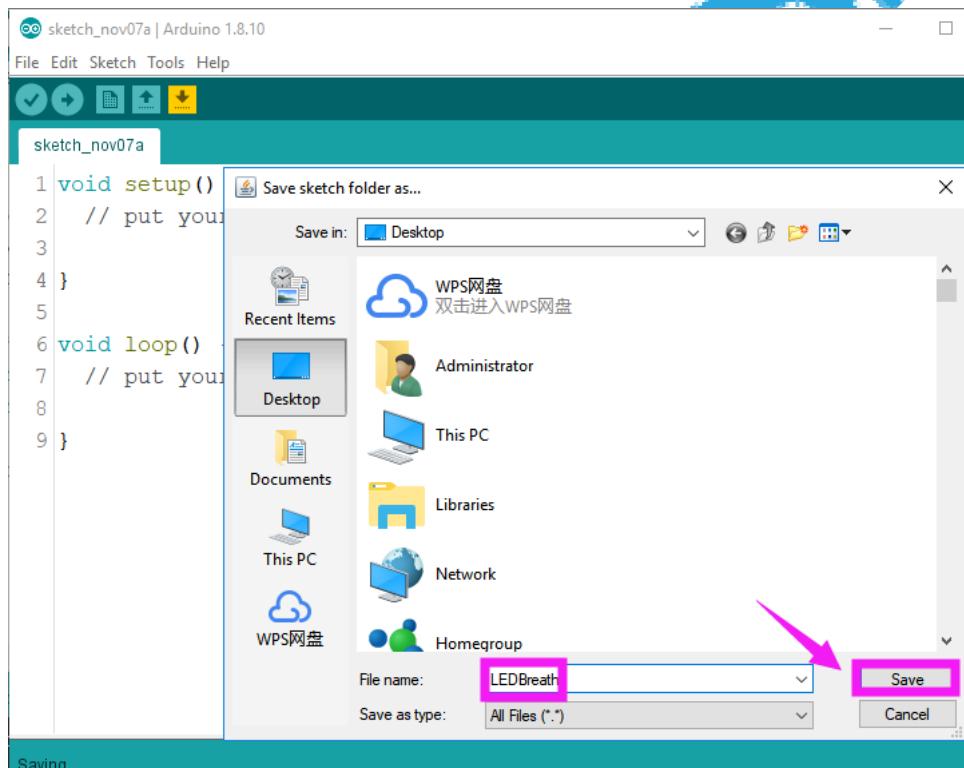
After the power is turned on, LED D9 will become breathing light.

3. Create new project

3.1 Click 【File】 --> 【New】 .



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



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

3.4 We will **LEDBreath.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 Define LED D9 pin

From the hardware manual, we can see that the pin corresponding to LED D9 is pin 5, so we create a new create definition and define LED D9 pin as pin 5

```
//Define LED light(D9)pin
#define LED_PIN 5
```

4.2 Define global variables, save LED brightness and change parameters

```
/* Define global variables */
int brightness = 0; //LED brightness value
int fadeAmount = 5; //Increase or decrease the brightness value each time
```

4.3 Initialize the LED D9 pin in the setup() function and set the pin mode to output mode.

```
void setup() {
    //put your setup code here, to run once:
    //set LED pin to output mode
    pinMode(LED_PIN, OUTPUT);
}
```

4.4 Writing the main loop function loop() code

We need to write the analog value to LED_PIN to modify the brightness of the LED. After writing the analog value, the LED will light different brightness according to different values. The higher the value, the higher the brightness.

```
//Write analog values to the LED_PIN pin
analogWrite(LED_PIN, brightness);
```

The value of the next loop plus the fadeAmount, the value of fadeAmount can be positive or negative, so the brightness may be increased or reduced.

```
//Modify the brightness of the next display
brightness = brightness + fadeAmount;

//Flip the fadeAmount value
//If brightness >= 255, set fadeAmount to a r
if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
}

delay(30);
```

Delay is 30 milliseconds.

5. Compiling and downloading code

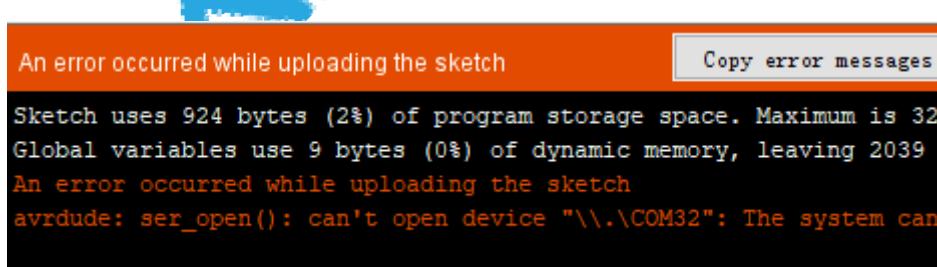
5.1 After the code is written, press Ctrl+S to save, then click the “√” button to compile. If there is no problem, click “→” to upload (the car must be connected to the computer via the USB cable).

```

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.



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].