

# Lesson 11 LED Flashing

## 1. Working Principle

The path to the source code of the program is 5. MaxArm Hardware Basic Learning/Arduino Development/Game Programs/LED/LED\_Blink/LED\_Blink.ino

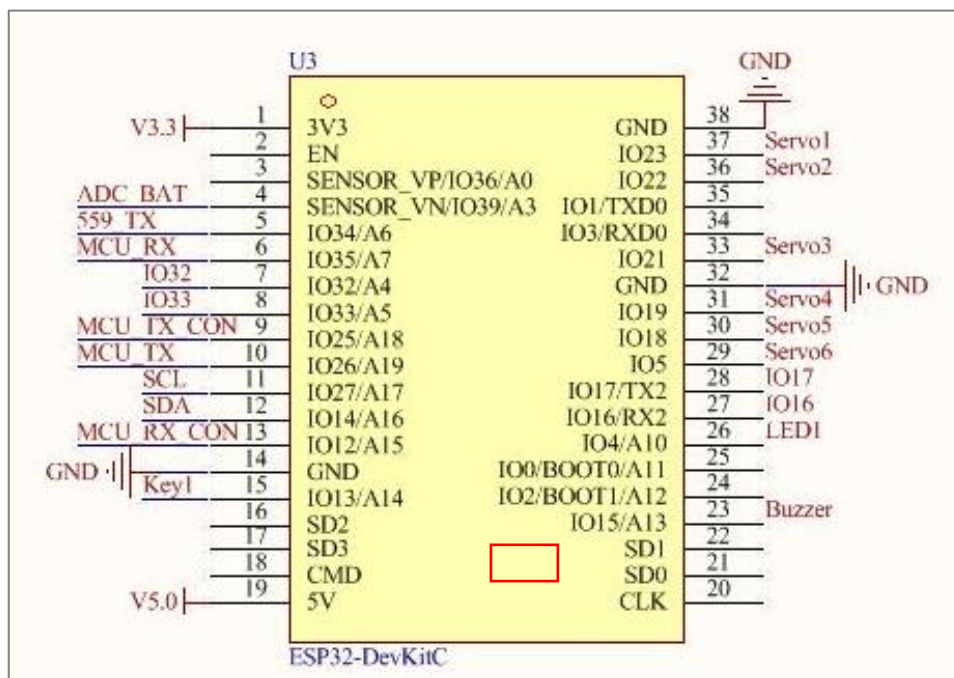
```

3 #define LED_BUILTIN 2 // Define LED control pin
4
5 void setup() {
6   // Initialize the pin LED_BUILTIN
7   pinMode(LED_BUILTIN, OUTPUT);
8 }
9
10 void loop() {
11   digitalWrite(LED_BUILTIN, HIGH); // Turn on LED (HIGH is the voltage level, which is set as high level)
12   delay(1000); // The delay of 1000ms
13   digitalWrite(LED_BUILTIN, LOW); // Turn off LED (LOW is the voltage level, which is set as low level)
14   delay(1000); // The delay of 1000ms
15 }

```

By defining the pin information of the LED, the digitalWrite() function is called to set the level signal of the pin. When the pin is high, the LED lights up, and when it is low, the LED is off.

The following image shows the pin information of ESP32 control chip, and the LED light is connected to IO2.



## 2. Preparation

### 2.1 Hardware

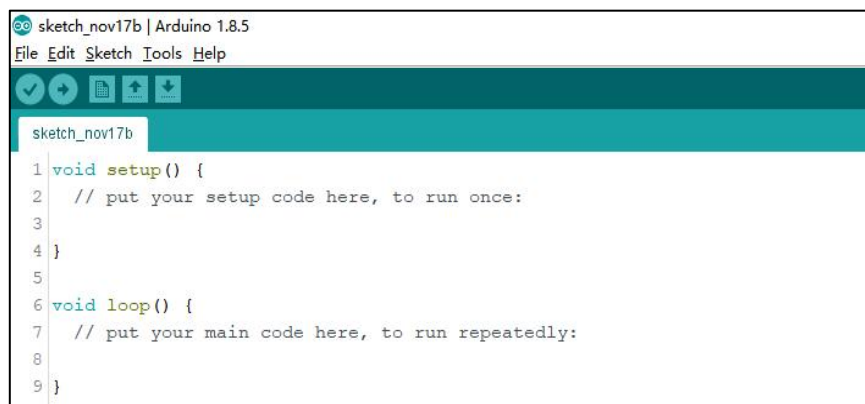
MaxArm robotic arm, power adapter, USB cable.

### 2.2 Software

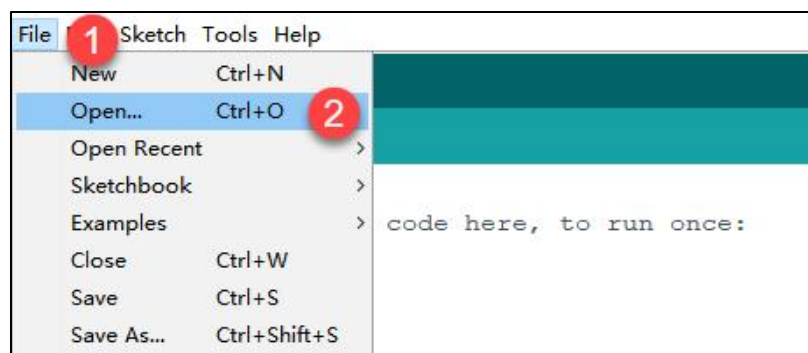
Please refer to the material in folder “4.MaxArm Underlying Program/Arduino Development/Lesson 1 Set Development Environment” to connect ESP32 controller to Arduino Editor.

## 3. Program Download

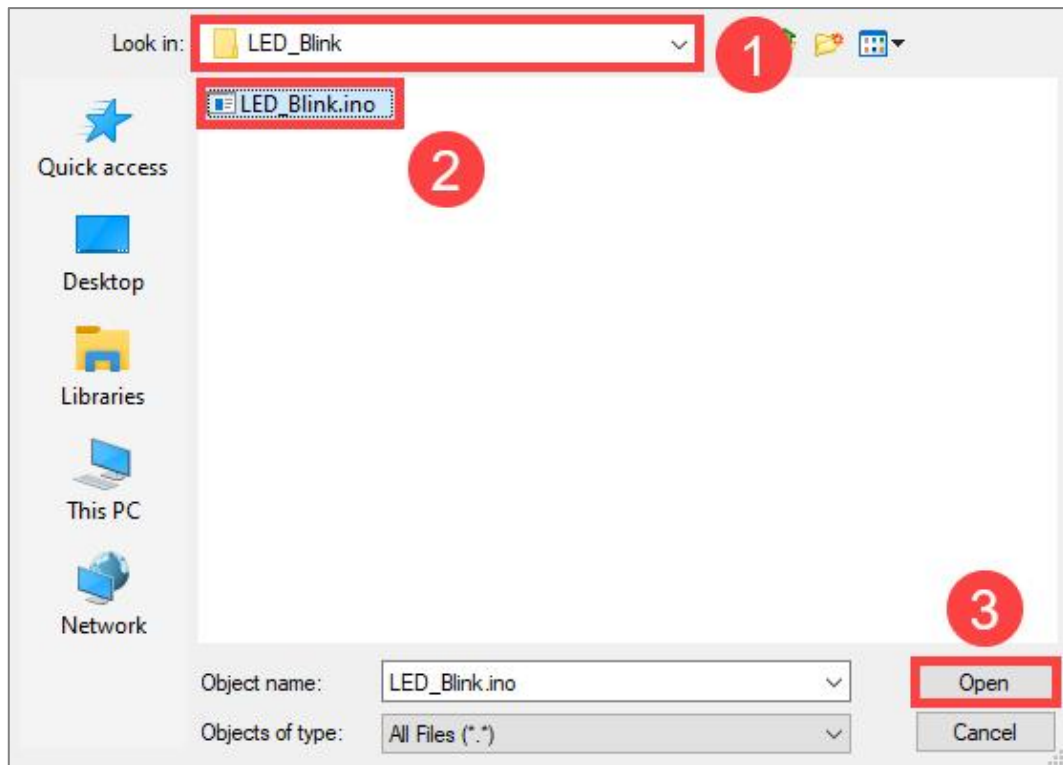
- 1) Double click on  icon to open Arduino IDE.



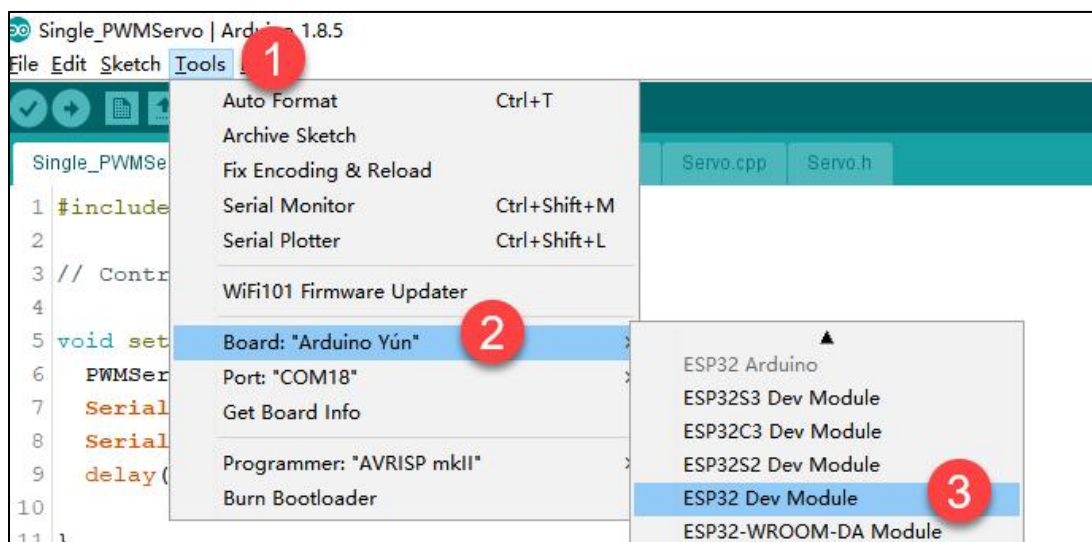
- 2) Click “File->Open” in turn.



- 3) Select the program “LED\_Blink.ino” in the folder “5.MaxArm Hardware Basic Learning/Arduino Development/Game Programs/LED/ LED\_Blink”.

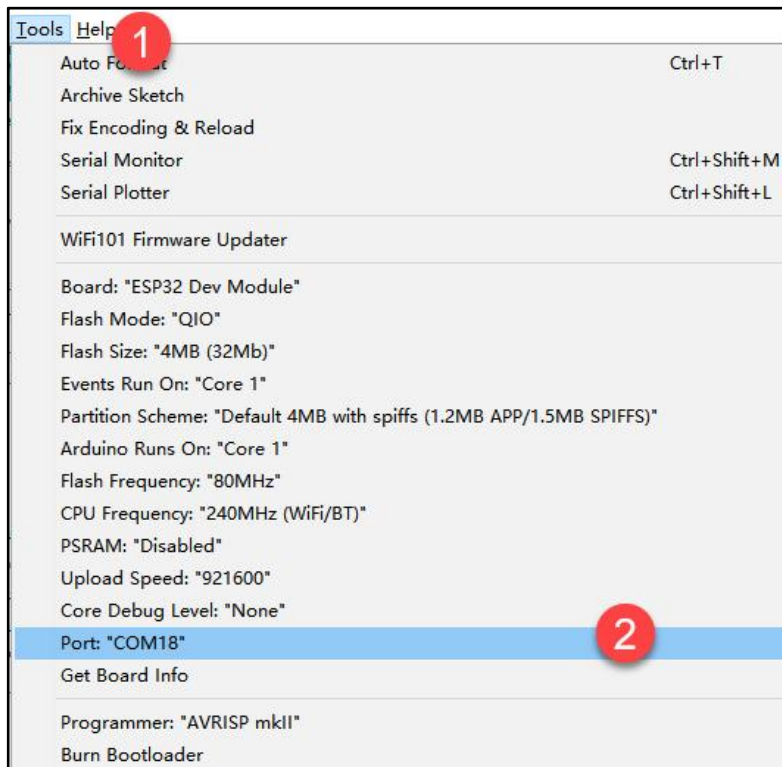


- 4) Check the board model. Click “Tools->Board” and select “ESP 32 Dev Module”. (If the model of development board has been configured when setting the development environment, you can skip this step.)

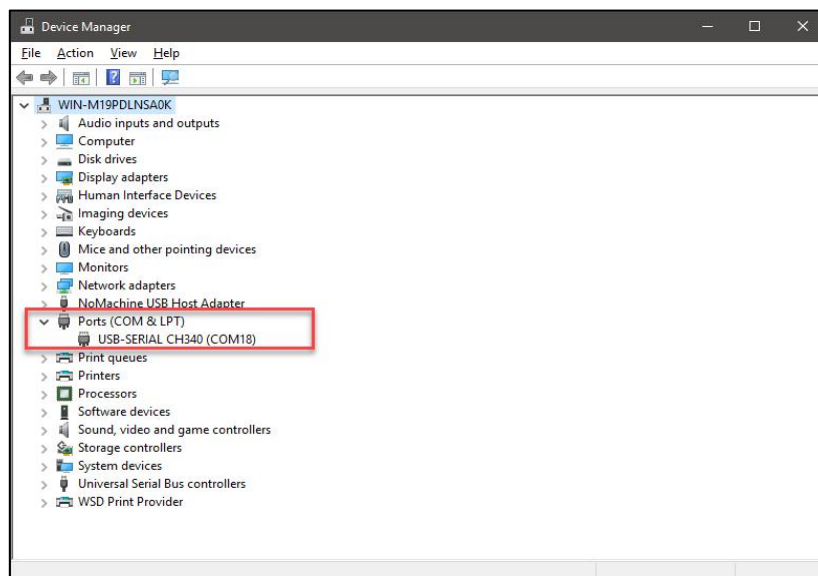


- 5) Select the corresponding port of ESP32 controller in “Tools->Port”. (Here take the port “COM5” as example. Please select the port based on your

computer. If COM1 appears, please do not select because it is the system communication port but not the actual port of the development port.)




- 6) If you're not sure about the port number, please open the "This PC" and click "Properties->Device Manager" in turns to check the corresponding port number (the device is with CH340).




- 7) After selecting, confirm the board "ESP32 Dev Module" in the lower right

corner and the port number “COM5” (it is an example here, please refer to the actual situation).

ESP32 Dev Module, Disabled, Default 4MB with spiiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None on COM18

- 8) Then click on  icon to verify the program. If no error, the status area will display “Compiling->Compile complete” in turn. After compiling, the information such as the current used bytes, and occupied program storage space will be displayed.

Done compiling.  
Sketch uses 247733 bytes (18%) of program storage space. Maximum is 1310720 bytes.  
Global variables use 16584 bytes (5%) of dynamic memory, leaving 311096 bytes for local variables. Maximum is 327680 bytes.

- 9) After compiling, click on  icon to upload the program to the development board. The status area will display “Compiling->Uploading->Complete” in turn. After uploading, the status area will stop printing the uploading information.

Done uploading.  
Leaving...  
Hard resetting via RTS pin... 175%

## 4. Project Outcome

The LED on ESP32 control chip flashes.

