

## Lesson 12 Timer

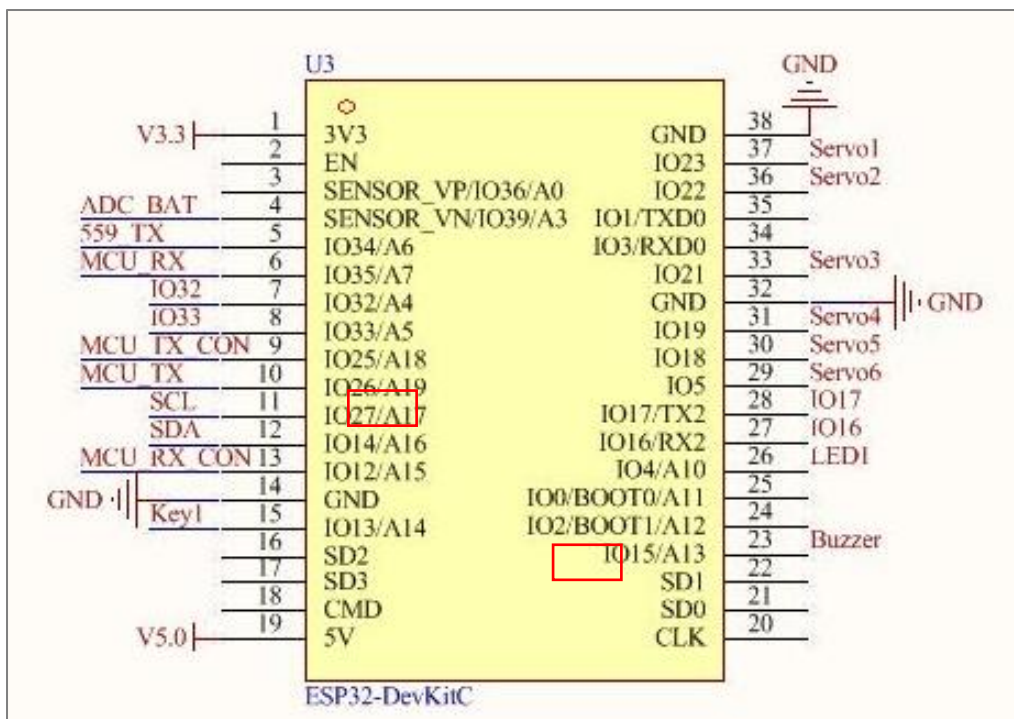
### 1. Working Principle

Timer as a hardware inside many microcontrollers is used to measure time, generally consisting of a time base generator and a counter. Time base is the basic unit of time. The time base generator generates a signal with a time base as the period, and the counter counts the number of signals generated by the time base generator.

The timer can make the microcontroller perform a specified operation at a specified interval or calculate the exact time interval between events.

For example, if the timebase is 1 second, the timebase generator will generate one signal per second and the counter will add 1 per second, when the counter value is equal to the set value, the microcontroller will perform the corresponding operation.

The following image shows the pin information of ESP32 expansion board.



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

```
4#define led1_pin 26 // Define the pin of LED1
5#define led2_pin 2 // Define the pin of LED2
6hw_timer_t * timer = NULL; //Declare a timer
7
8void IRAM_ATTR onTimer() {
9    digitalWrite(led1_pin, !digitalRead(led1_pin)); // Set LED1 as reverse status
10    digitalWrite(led2_pin, !digitalRead(led2_pin)); // Set LED2 as reverse status
11}
12
13void setup() {
14    Serial.begin(115200);
15    Serial.println("start...");
16    pinMode(led1_pin, OUTPUT);
17    pinMode(led2_pin, OUTPUT);
18    digitalWrite(led1_pin, LOW); // Set the pin of LED1 as low level
19    digitalWrite(led2_pin, LOW); // Set the pin of LED2 as low level
20    timer = timerBegin(0, 80, true); // Initialization. Use the first timer. Four timer count from 0. The prescaler is set as 80.
21    timerAttachInterrupt(timer, &onTimer, true); // Call the interruption function
22    timerAlarmWrite(timer, 1000000, true); // Set the trigger time (unit:us, 1000000us=1s )
23    timerAlarmEnable(timer); // Enable timer
24}
25
```

## 2. Preparation

### 2.1 Hardware


MaxArm robotic arm, power adapter, USB cable.

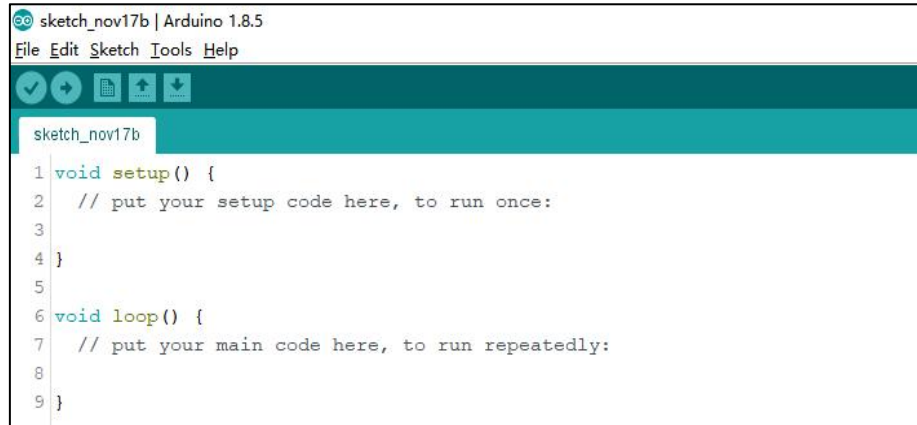
### 2.2 Software

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

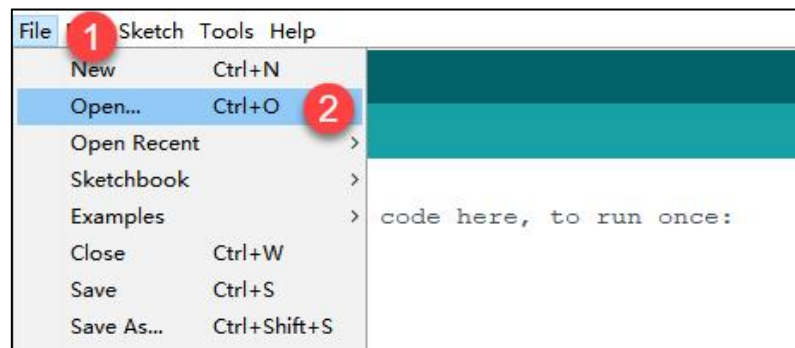
## 3. Program Download



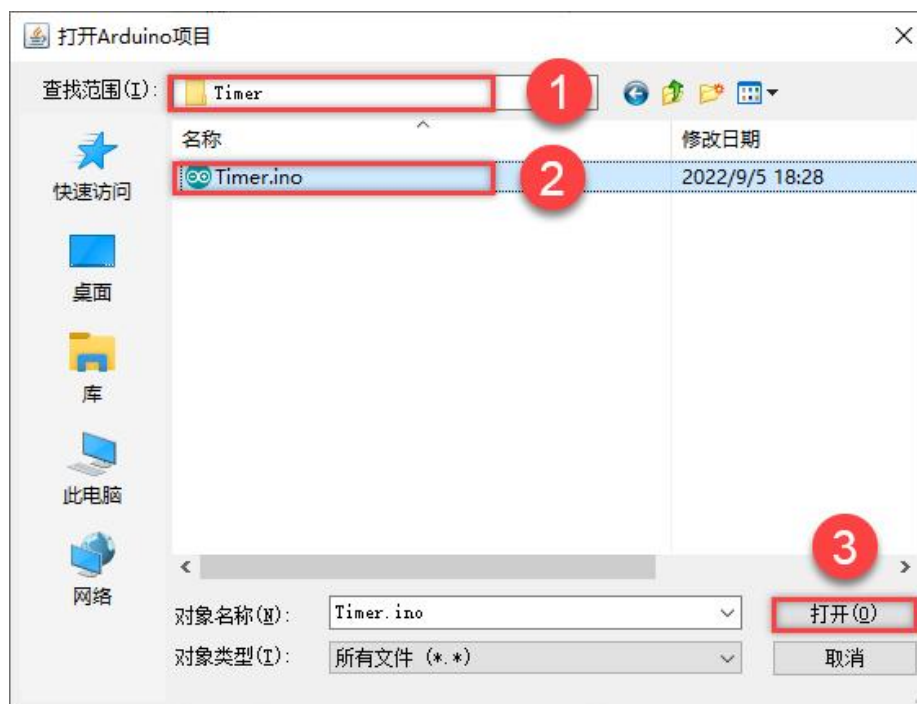
- 1) Double click on  icon to open Arduino IDE.



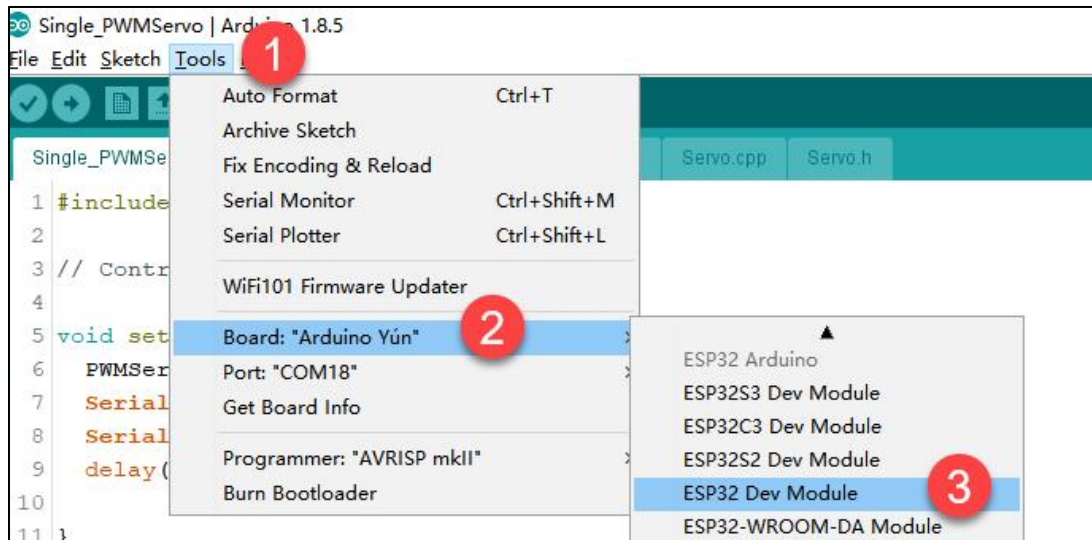
2) Click “File->Open” in turn.



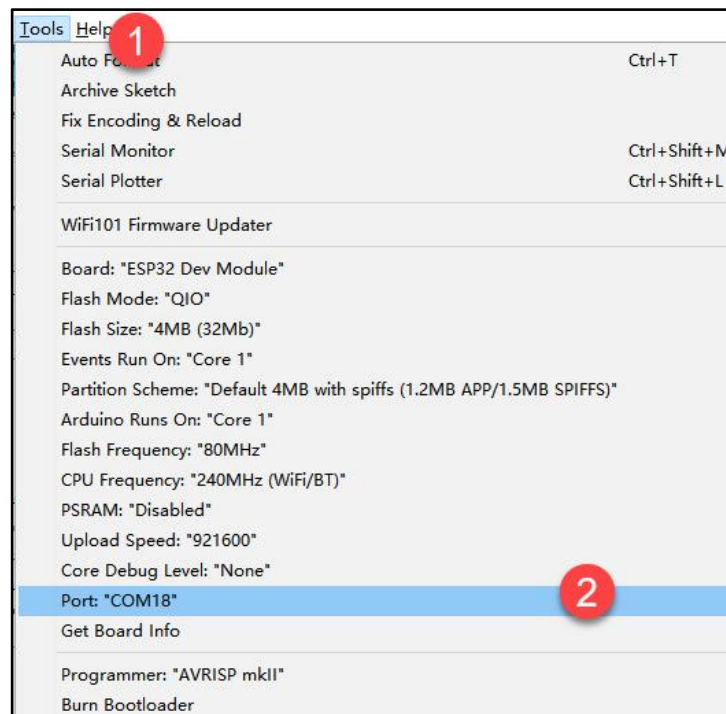
3) Select the program “Timer.ino” in the folder “5.MaxArm Hardware Basic Learning/Arduino Development/Game Programs/Timer”.



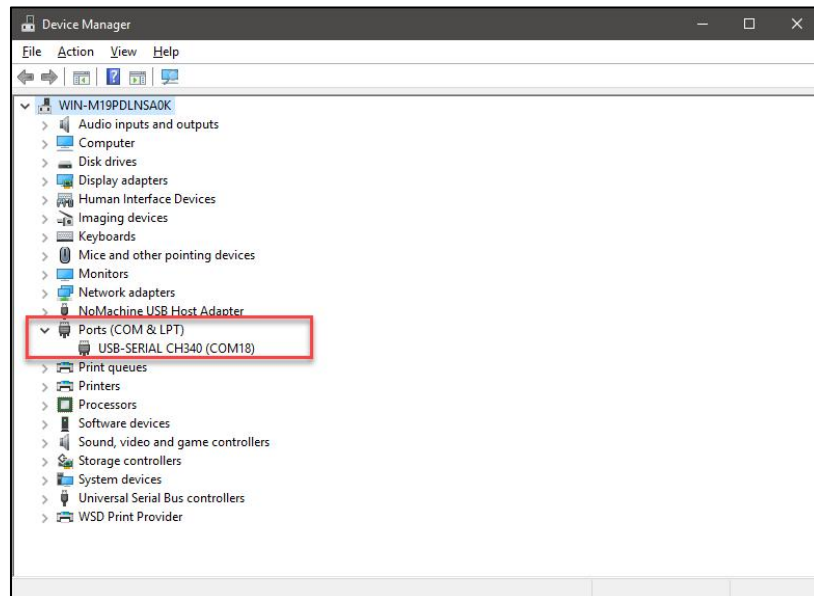
- 4) Check the board model. Click “Tools->Board” and select “ESP 32 Dev Module”. (If the model of development board has been configured when setting the development environment, you can skip this step.)



- 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.
```

## 4. Project Outcome

The LED lights on controller turn on and off alternately.