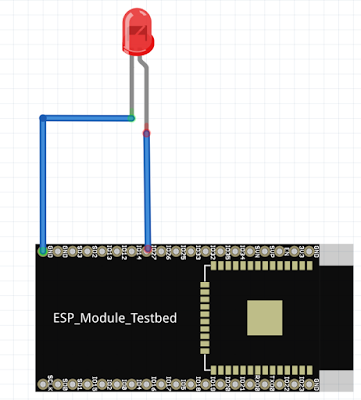
# **[Demo 22: How to use Timer interrupt in Arduino ESP32](http://www.iotsharing.com/2017/06/how-to-use-interrupt-timer-in-arduino-esp32.html)**

**1. Introduction**  
- In[blinky demo](http://www.iotsharing.com/2017/05/blinky-hello-world-on-arduino-esp32.html" \t "http://www.iotsharing.com/2017/06/_blank) we use delay() function to make the LED blink. In this demo we will replcae delay() function by using Timer (this is hardware timer). Using Timer we can schedule when a task need to be started and repeat or not. Applying to this demo, when the Timer is timeout we will change the current state of LED (ON to OFF to ON) every 1 second.  
**2. Hardware**

[](https://2.bp.blogspot.com/-Edc2Mnm_Rh8/WRjx2wVCVhI/AAAAAAAADzc/NK8uFvWzGC8dW6XMTuiXrDEsX9UCDajzQCPcBGAYYCw/s1600/led2.png)

**Figure: ESP32 connect to LED**

Connection:  
[ESP32 IO14 and ESP32 GND to LED]  
**3. Software**  
In order to use Timer we will use the functions:  
**"hw\_timer\_t \* timerBegin(uint8\_t num, uint16\_t divider, bool countUp)"**  
+ num: is order of Timer. We only have 4 timers so the order can be 0,1,2,3.  
+ divider: it is a prescaler. To make 1 second scheduler, we will use divider value is 80. ESP32 main clock is 80MHZ so every tick will take T = 1/(80MHZ/80) = 1 microsecond. We need 1000000 ticks for 1 second.  
+ countUp: if it is true the timer will count up and vice versa.  
**"void timerAttachInterrupt(hw\_timer\_t \*timer, void (\*fn)(void), bool edge)"**  
+ fn: is the callback function that will be invoked when timer timeout and timer interrupt will be invoked.  
+ edge: if it is true, an alarm will generate an edge type interrupt.  
**"void timerAlarmWrite(hw\_timer\_t \*timer, uint64\_t alarm\_value, bool autoreload)"**  
+ alarm\_value: we set it to 1000000 as calculated above  
+ autoreload: if it is true, timer will repeat.  
**"void timerAlarmEnable(hw\_timer\_t \*timer)"**: enable the Timer.

|  |
| --- |
| /\* create a hardware timer \*/  hw\_timer\_t \* timer = NULL;  /\* LED pin \*/  int led = 14;  /\* LED state \*/  volatile byte state = LOW;  void IRAM\_ATTR onTimer(){  state = !state;  digitalWrite(led, state);  }  void setup() {  Serial.begin(115200);  pinMode(led, OUTPUT);  /\* Use 1st timer of 4 \*/  /\* 1 tick take 1/(80MHZ/80) = 1us so we set divider 80 and count up \*/  timer = timerBegin(0, 80, true);  /\* Attach onTimer function to our timer \*/  timerAttachInterrupt(timer, &onTimer, true);  /\* Set alarm to call onTimer function every second 1 tick is 1us  => 1 second is 1000000us \*/  /\* Repeat the alarm (third parameter) \*/  timerAlarmWrite(timer, 1000000, true);  /\* Start an alarm \*/  timerAlarmEnable(timer);  Serial.println("start timer");  }  void loop() {  } |