



UNIVERSIDAD
COMPLUTENSE
MADRID

ESP-IDF. Timers

IoT Node Architecture

- ❑ *Timer* is a programmable counter that we can configure to trigger an interrupt after a given time
 - We can program periodic *alarms*
 - It is **asynchronous**: we don't know the point of our code that will be executing when the timer alarm happens
- ❑ Any SoC includes several hardware *timers*, that we may use from our code

- ❑ We need to define a *callback* function that will be executed when the timer expires
 - Similar to an ISR
 - Indeed, the HW timer will raise an interrupt
 - We must avoid blocking calls in that *callback* (do not call *Delay()*...)
- ❑ There are ususally 2 modes
 - One-shot: we program the timer once. When it reaches 0, it triggers the alarm and timer stops
 - Auto-reload/continous: when the timer reaches 0, it triggers the alarm and it automatically arms the alarm again, so the next alarm is programmed
 - Best way to schedule periodic events

- ❑ ESP-IDF implements basic *software timers* básicos from FreeRTOS and their own *High Resolution Timer*
 - https://docs.espressif.com/projects/esp-idf/en/stable/api-reference/system/esp_timer.html
- ❑ It uses a 32 bits hardware timer
 - It creates a high priority task to execute the *callback*
 - Precision up to 50us
- ❑ Just like *any other time*
 - One-shot – AutoReload + *callback*
 - REMINDER: *callback* function should be as simple as possible and never get blocked

```
Void app_main() {  
    ...  
    const esp_timer_create_args_t periodic_timer_args = {  
        .callback = &periodic_timer_callback,  
        .name = "periodic" };  
    esp_timer_handle_t periodic_timer;  
    esp_timer_create(&periodic_timer_args, &periodic_timer);  
    ...  
    esp_timer_start_periodic(periodic_timer, 500000); // time in microseconds  
    ...  
    esp_timer_stop(periodic_timer);  
    ...  
    esp_timer_delete(periodic_timer);  
}  
  
static void periodic_timer_callback(void* arg) {  
    int64_t time_since_boot = esp_timer_get_time(); //returns time from boot in microseconds  
    printf("Periodic timer called, time since boot: %lld us",time_since_boot);  
}
```

https://github.com/espressif/esp-idf/tree/v4.1/examples/system/esp_timer