TIMER EXAMPLE

```
#include <stdio.h>
#include <esp_types.h>
#include <freertos/FreeRTOS.h>
#include <freertos/task.h>
#include <freertos/queue.h>
#include <driver/peripheral_ctrl.h>
#include <driver/timer.h>


#define TIMER_DIVIDER 16
#define TIMER_SCALE (TIMER_BASE_CLK/TIMER_DIVIDER)
#define TIMER_INTERVAL0_SEC (3.4179)
#define TIMER_INTERVAL1_SEC (5.78)
#define TEST_WITHOUT_RELOAD 0
#define TEST_WITH_RELOAD 1


typedef struct {
  int type;
  int timer_group;
  int timer_idx;
  uint64_t timer_counter_value;
} timer_event_t;


xQueueHandle timer_queue;
```

```
static void inline print_timer_counter (uint64_t counter_value)
{
    printf("Counter: 0x%.08x%.08x\n", (uint32_t) (counter_value >> 32),
                                      (uint32_t) (counter_value));
    printf("Time: %.8fs\n", (double) counter_value / TIMER_SCALE);
}


void IRAM_ATTR timer0_group0_isr (void *para) {
    int timer_idx = (int) para;
    uint32_t intr_status = TIMERG0.int_st_timers.val;
    TIMERG0.hw_timer[timer_idx].update = 1;
    uint64_t timer_counter_value =
        ((uint64_t) TIMERG0.hw_timer[timer_idx].cnt_high << 32
        | TIMERG0.hw_timer[timer_idx].cnt_low;

    timer_event_t evt;
    evt.timer_group = 0;
    evt.timer_idx = timer_idx;
    evt.timer_counter_value = timer_counter_value;

    if ((intr_status & BIT(timer_idx)) && timer_idx == TIMER0) {
        evt.type = TEST_WITHOUT_RELOAD;
        TIMERG0.int_clr_timers.t0 = 1;
        timer_counter_value += (uint64_t) (TIMER_INTERVAL0_SEC * TIMER_SCALE);
        TIMERG0.hw_timer[timer_idx].alarm_high
                = (uint32_t) (timer_counter_value >> 32);
        TIMERG0.hw_timer[timer_idx].alarm_low
                = (uint32_t) timer_counter_value;
    }
```

```
    else if ((intr_status & BIT(timer_idx)) && timer_idx == TIMER_1) {
        evt.type = TEST_WITH_RELOAD;
        TIMERG0.int_clr_timers.t1 = 1;
    } else {
        evt.type = -1;
    }
    TIMERG0.hw_timer[timer_idx].config.alarm_en = TIMER_ALARM_EN;
    xQueueSendFromISR(timer_queue, &evt, NULL);
}

static void example_tg0_timer_init(int timer_idx,
                                   bool auto_reload, double timer_interval_sec) {
    timer_config_t config;
    config.divider = TIMER_DIVIDER;
    config.counter_dir = TIMER_COUNT_UP;
    config.counter_en = TIMER_PAUSE;
    config.alarm_en = TIMER_ALARM_EN;
    config.intr_type = TIMER_INTR_LEVEL;
    config.auto_reload = auto_reload;
    timer_init(TIMER_GROUP_0, timer_idx, &config);
    timer_set_counter_value(TIMER_GROUP_0, timer_idx, 0x00000000ULL);
    timer_set_alarm_value(TIMER_GROUP_0, timer_idx,
                          timer_interval_sec * TIMER_SCALE);
    timer_enable_intr(TIMER_GROUP_0, timer_idx);
    timer_isr_register(TIMER_GROUP_0, timer_idx,
                       timer_group0_isr, (void*) timer_idx, ESP_INTR_FLAG_IRAM, NULL);
    timer_start(TIMER_GROUP_0, timer_idx);
}
```

```
static void timer_example_evt_task (void *arg) {
  while (1) {
    timer_event_t evt;
    xQueueRecieve (timer_queue, &evt, portMAX_DELAY);

    if (evt.type == TEST_WITHOUT_RELOAD) {
      printf("\n Example timer without reload \n");
    } else if (evt.type == TEST_WITH_RELOAD) {
      printf("\n Example timer with auto reload \n");
    } else {
      printf("\n UNKNOWN EVENT TYPE\n");
    }
    printf("Group[%d], timer[%d] alarm event\n",
           evt.timer_group, evt.timer_idx);
    printf("---- EVENT TIME ----\n");
    print_timer_counter(evt.timer_counter_value);
    printf("---- TASK TIME ----\n");
    uint64_t task_counter_value;
    timer_get_counter_value(evt.timer_group, evt.timer_idx,
                            &task_counter_value);
    print_timer_counter(task_counter_value);
  }
}

void app_main() {
  timer_queue = xQueueCreate(10, sizeof(timer_event_t));
  example_tg0_timer_init(TIMER_0, TEST_WITHOUT_RELOAD,
                         TIMER_0_INTERVAL_SEC);
  example_tg0_timer_init(TIMER_1, TEST_WITH_RELOAD,
                         TIMER_INTERVAL_SEC);
  xTaskCreate(timer_example_evt_task, "timer_evt_task", 2048, NULL,
              5, NULL);
}
```