VIETNAM NATIONAL UNIVERSITY - HCM
Ho Chi Minh City University of Technology
Faculty of Computer Science and Engineering

EMBEDDED SYSTEM (CO3053)

REPORT LAB 4

FreeRTOS Software Timer

Name	Student's ID
Nguyễn Hữu Trung Nhân	1752392

Ho Chi Minh City, June 2020 - Semester 192

Contents

1 Goal

The objective is to create **2 software timers**. These 2 timers share **one timer callback function**

The first timer is used to print "ahihi" every 2 seconds and will stop after 10 times printing.

The second timer is used to print "ihaha" every 3 seconds and will stop after 5 times printing.

2 Implementation

After reading the lab material extremely carefully, I implemented my program based on some provided API function.

In order to create a software timer that call the callback function every 2 seconds, I did as below:

```
xTimerHandle timerHnd2Sec = xTimerCreate("timer2sec", pdMS_TO_TICKS(2000), pdTRUE,
     (void*)0, vTimerCallBackExpired);
xTimerStart(timerHnd2Sec, 0);

//pdMS_TO_TICKS(2000) means that this timer will call the callback function "
    vTimerCallBackExpired" (in my case, this is the name I choose for callback
    function)
```

In order to create a software timer that call the callback function every 3 seconds, I did as below:

```
timerHnd3Sec = xTimerCreate("timer3sec", pdMS_TO_TICKS(3000), pdTRUE, (void*)0,
    vTimerCallBackExpired);
xTimerStart(timerHnd3Sec, 0);
```

Because the requirement is **2 software timers** share "one timer callback function", I implement the callback function "vTimerCallBackExpired" as below:

```

int timesFirstTimer = 0;
int timesSecondTimer = 0;

static void vTimerCallBackExpired(xTimerHandle pxTimer){
    if (pxTimer == timerHnd2Sec){
        printf("ahihi\n");
        timesFirstTimer = timesFirstTimer + 1;
    }

    if (timesFirstTimer == 10){
        timesFirstTimer = 0;
        xTimerDelete(timerHnd2Sec, 0);
    }

    if (pxTimer == timerHnd3Sec){
        printf("ihahha\n");
        timesSecondTimer = timesSecondTimer + 1;
    }

    if (timesSecondTimer == 5){
        timesSecondTimer = 0;
        xTimerDelete(timerHnd3Sec, 0);
    }

}
```

Because the requirement are **the first timer will stop after 10 times printing** and **the second timer will stop after 5 times printing**, I created two variables "timesFirstTimer" and "timesSecondTimer". Whenever the first software timer calls this callback function, "timesFirstTimer" will increase 1 unit, when "timeFirstTimer" reaches 10, xTimerDelete() API function will be called to delete the first software timer. The same idea is applied for the second software timer.

3 Source Code

```
#include <stdio.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_system.h"
#include "esp_spi_flash.h"
#include "sdkconfig.h"
#include "driver/gpio.h"
#include "freertos/timers.h"

xTimerHandle timerHnd2Sec;
xTimerHandle timerHnd3Sec;

int timesFirstTimer = 0;
int timesSecondTimer = 0;

void task1(void *p){
    while(1){

    }
    vTaskDelete(NULL);
}

static void vTimerCallBackExpired(xTimerHandle pxTimer){
    if (pxTimer == timerHnd2Sec){
        printf("ahihi\n");
        timesFirstTimer = timesFirstTimer + 1;
    }

    if (timesFirstTimer == 10){
        timesFirstTimer = 0;
        xTimerDelete(timerHnd2Sec, 0);
    }

    if (pxTimer == timerHnd3Sec){
        printf("ihahha\n");
        timesSecondTimer = timesSecondTimer + 1;
    }

    if (timesSecondTimer == 5){
        timesSecondTimer = 0;
        xTimerDelete(timerHnd3Sec, 0);
    }

}

void app_main(void)
{
    xTaskCreate(&task1, "task1", 1024*1, (void*) 0, tskIDLE_PRIORITY + 0, NULL);

```

```
50      timerHnd2Sec = xTimerCreate("timer2sec", pdMS_TO_TICKS(2000), pdTRUE, (void*)
        0, vTimerCallBackExpired);
51      xTimerStart(timerHnd2Sec, 0);
52
53      timerHnd3Sec = xTimerCreate("timer3sec", pdMS_TO_TICKS(3000), pdTRUE, (void*)
        0, vTimerCallBackExpired);
54      xTimerStart(timerHnd3Sec, 0);
55  }
```

4 Picture of my workspace

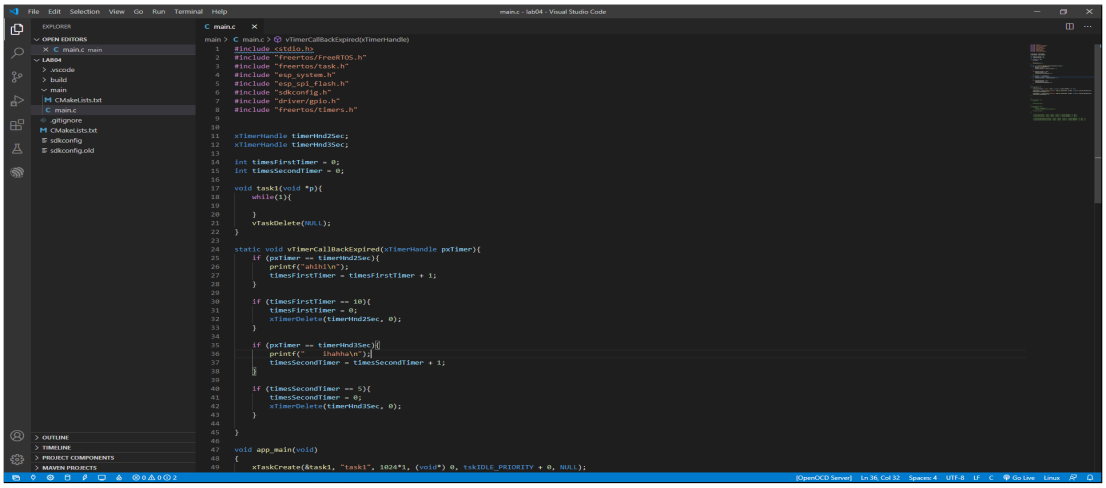

Figure 1

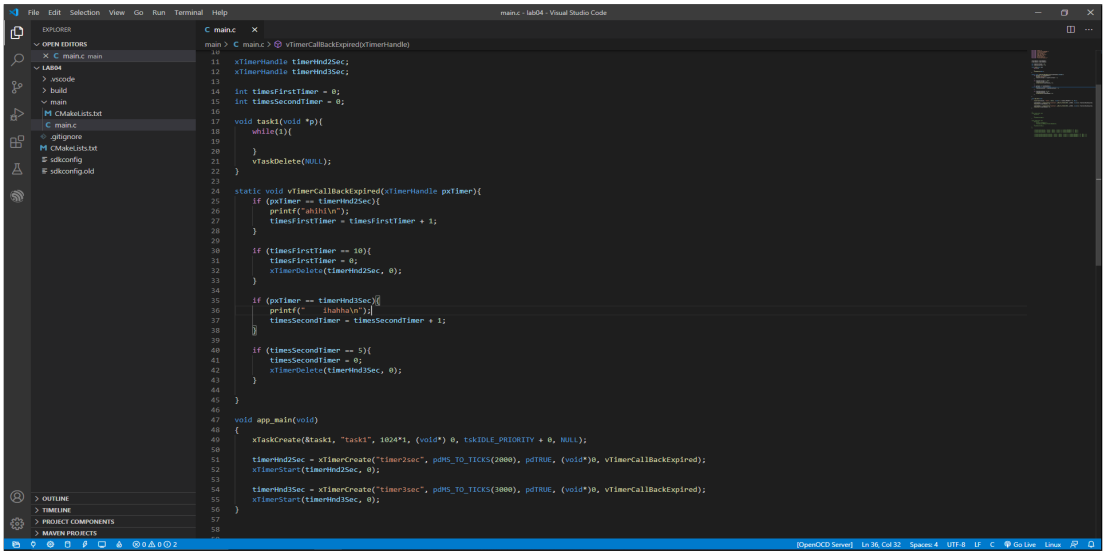

Figure 2

5 The Final Result

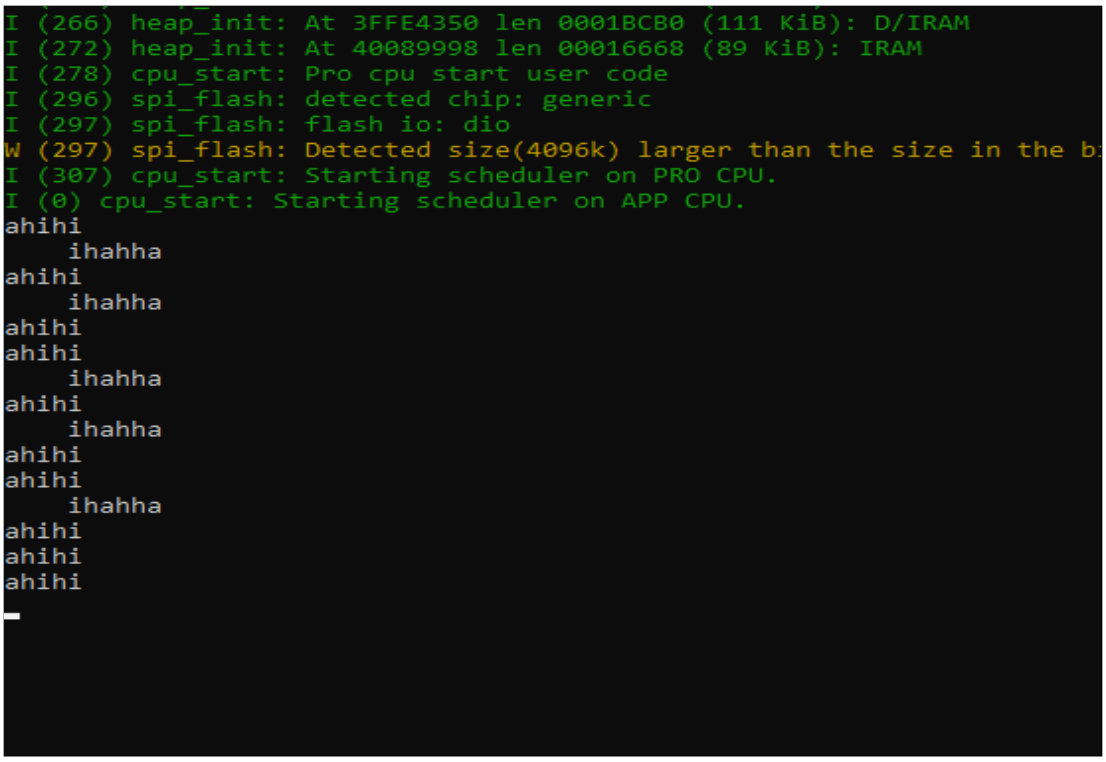

Figure 3