Embedded system lab 1

Introduction to ESP32 and ESP-IDF and ESP32 GPIO and FreeRTOS task

Name: Nguyễn Trần Anh Quân

ID: 1952418

In-class Requirements:

ESP-L01 code:

```
#include <stdio.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_system.h"
void hello_task(void *pvParameter)
    printf("Hello world!\n");
    for (int i = 10; i >= 0; i--) {
        printf("Restarting in %d seconds...\n", i);
        vTaskDelay(1000 / portTICK_RATE_MS);
    printf("Restarting now.\n");
    fflush(stdout);
    esp_restart();
}
void app_main()
    xTaskCreate(&hello_task, "hello_task", 2048, NULL, 5, NULL);
}
```

ESP-L01 result:

```
■ workspace - neilo_world/main/main.c - Espressit-IUE
ille Edit Source Refactor Navigate Search Project Run Espressif Window Help

□ □ □ □ Run □ □ hello_world □ ⇔ on: ⊕ esp3:
 ≪ O Run
                                                                                                                                                                                                                                                    ∨ 🌼 on: 🏵 esp32
                                                                                           Project Explorer ×

$\sigma$ hello_world [esp-idf-v4.4.2 v4.4.2 lb16ef6]
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   □ □ Build Targ
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       stdio.h
freertos/FreeRTOS.h
freertos/task.h
esp_system.h
hello_task(void*):vi
app_main():void
        > 👺 build
      build
c+ esp_idf_components
c+ > main
c+ CMakeLists.txt
c- Kconfig.projbuild
                                                                                                                                                                       6@ void hello_task(void *pvParameter)
                                                                                                                                                                    7 {
8 printf("Hello world!\n");
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         0 M (0
                                                                                                                                                                  ♠ CMakeLists.txt
                                                                                                                                                                □ COMS ×

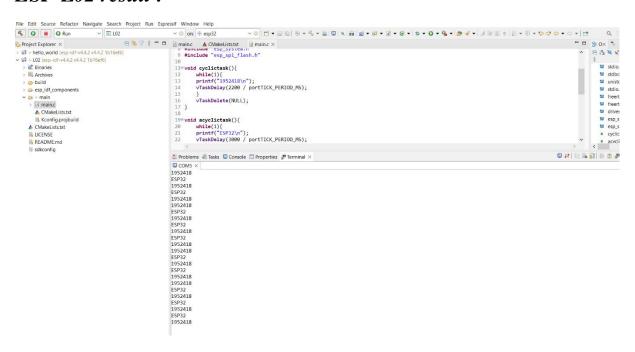
I (188) cpu_start: App cpu up.
I (215) cpu_start: Pro cpu start user code
I (216) cpu_start: cpr req: 160000000
I (216) cpu_start: cpr req: 1600000000
I (216) cpu_start: Application information:
I (220) cpu_start: Application information:
I (220) cpu_start: Application information:
I (212) cpu_start: Capplie time: Cot 28 2022 13:38:17
I (237) cpu_start: EpF file SHA256: 6c86se0929df998f...
I (243) cpu_start: EpF file SHA256: 6c86se0929df998f.
I (243) cpu_start: EpF file SHA256: 6c86se0929df998f.
I (243) cpu_start: EpF file SHA256: 0c86se0929df998f.
I (243) cpu_start: EpF file SHA256: 0c86se0929df998f.
I (243) cpu_start: EpF file SHA256: 0c86se0929df998f.
I (243) heap_init: At 3FF8A258 len 00001920 (6c18): DRAM
I (255) heap_init: At 3FF8A258 len 00001920 (8c18): DRAM
I (268) heap_init: At 3FF8A258 len 00001808 (111 KiB): D/IRAM
I (274) heap_init: At 3FF8A258 len 0001808 (111 KiB): D/IRAM
I (274) heap_init: At 3FF8A358 len 0001808 (111 KiB): D/IRAM
I (274) heap_init: At 3FF8A358 len 0001808 (111 KiB): D/IRAM
I (278) spi_flash: detected chip: generic
I (291) spi_flash: Detected size(A0906k) larger than the size in the bim
I (309) cpu_start: Starting scheduler on RP0 CPU.
I (0) cpu_start: Starting scheduler on RP0 CPU.
Restarting in 10 seconds...
Restarting in 10 seconds...
Restarting in 5 seconds...
Restarting in 4 seconds...
Restarting in 5 seconds...
Restarting in 5 seconds...
Restarting in 5 seconds...
Restarting in 3 seconds...
Restarting in 5 seconds...
Restarting in 5 seconds...
Restarting in 5 seconds...
                                                                                                                                                                  □ COM5 ×
             IR LICENSE
             README.md
```

ESP-L02 code:

```
#include <stdio.h>
#include <stdbool.h>
#include <unistd.h>
#include <stdio.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "driver/gpio.h"
#include "esp_system.h"
#include "esp_spi_flash.h"
void cyclictask(){
      while(1){
      printf("1952418\n");
      vTaskDelay(2200 / portTICK_PERIOD_MS);
      vTaskDelete(NULL);
}
void acyclictask(){
      while(1){
      printf("ESP32\n");
      vTaskDelay(3000 / portTICK_PERIOD_MS);
      vTaskDelete(NULL);
}
void app_main(void)
```

```
xTaskCreate(
                    cyclictask,
                    "cyclic ",
                    1024*2 ,
                    NULL ,
                    1,
                    NULL
             );
      xTaskCreate(
                           acyclictask,
                           "acyclic ",
                           1024*2 ,
                           NULL,
                           1,
                           NULL
                    );
}
```

ESP-L02 result:



Does the ESP-IDF need the vTaskStartScheduler() routine?

ESP-IDF FreeRTOS does not require users to call vTaskStartScheduler() to start the scheduler. The startup flow of an ESP-IDF application will already call this automatically.