Doan Viet Tu – 1952521

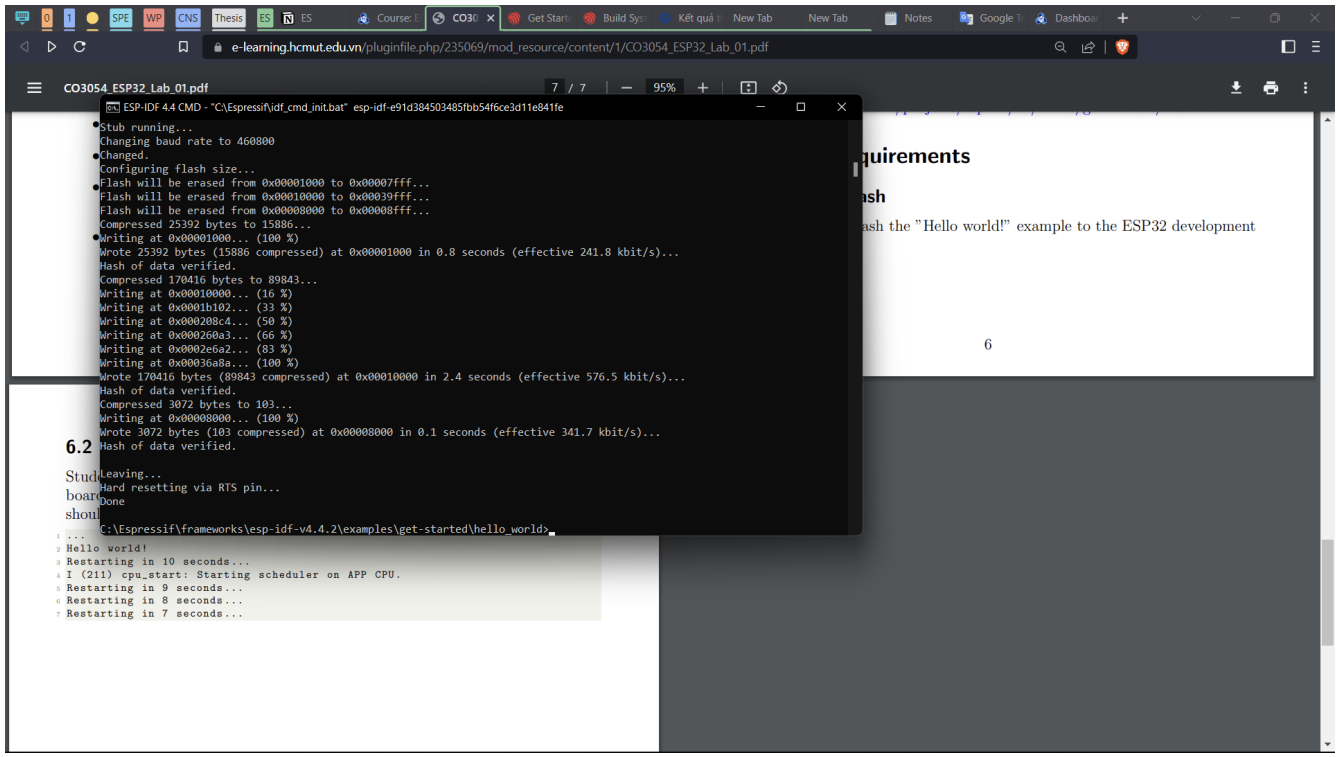
EMBEDDED SYSTEMS – CC01
LAB 1
ESP

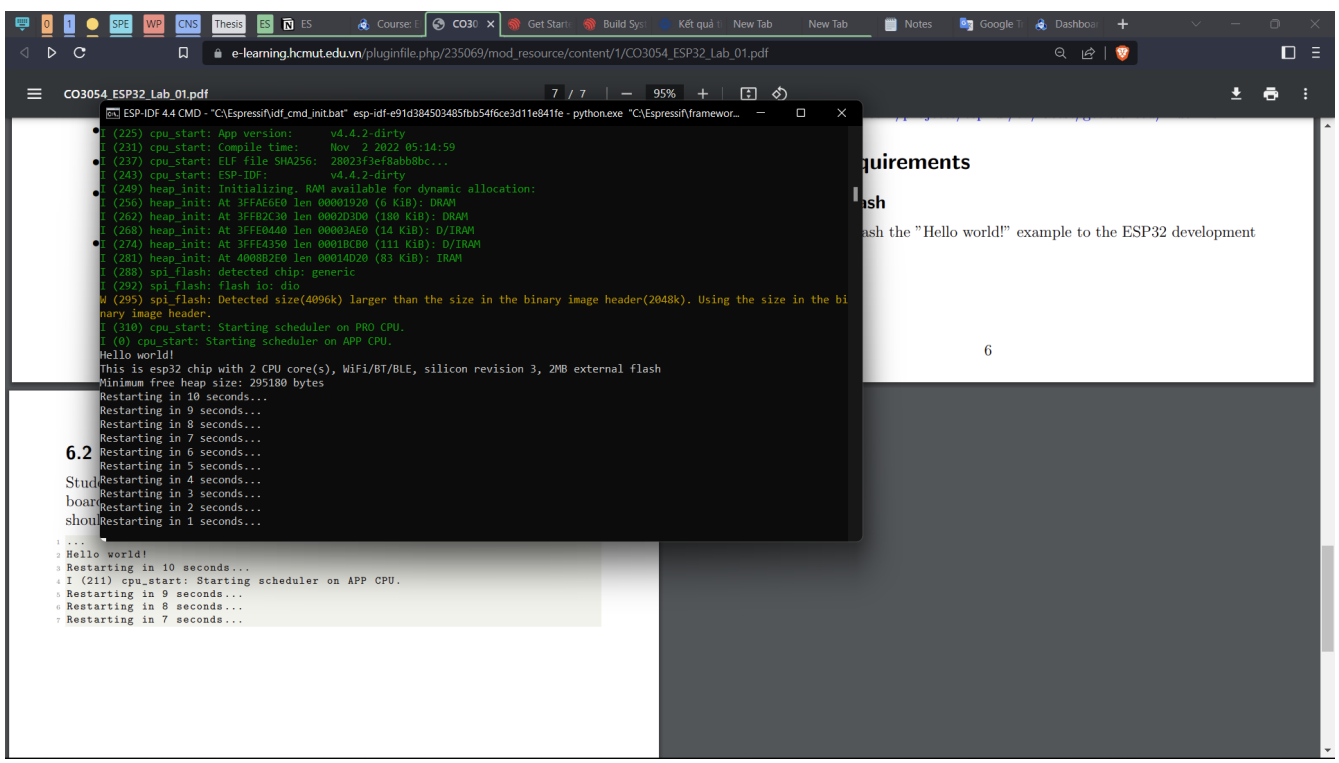
ESP-L01

Source code:

```c
*/
#include <stdio.h>
#include "sdkconfig.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_system.h"
#include "esp_spi_flash.h"

void app_main(void)
{
    printf("Hello world!\n");

    /* Print chip information */
    esp_chip_info_t chip_info;
    esp_chip_info(&chip_info);
    printf("This is %s chip with %d CPU core(s), WiFi%s%s, ",
            CONFIG_IDF_TARGET,
            chip_info.cores,
            (chip_info.features & CHIP_FEATURE_BT) ? "/BT" : "",
            (chip_info.features & CHIP_FEATURE_BLE) ? "/BLE" : "");

    printf("silicon revision %d, ", chip_info.revision);

    printf("%dMB %s flash\n", spi_flash_get_chip_size() / (1024 * 1024),
            (chip_info.features & CHIP_FEATURE_EMB_FLASH) ? "embedded" : "external");

    printf("Minimum free heap size: %d bytes\n", esp_get_minimum_free_heap_size());

    for (int i = 10; i >= 0; i--) {
        printf("Restarting in %d seconds...\n", i);
        vTaskDelay(1000 / portTICK_PERIOD_MS);
    }
    printf("Restarting now.\n");
    fflush(stdout);
    esp_restart();
}
```

Results:


e-learning.hcmut.edu.vn/pluginfile.php/235069/mod_resource/content/1/CO3054_ESP32_Lab_01.pdf
CO3054_ESP32_Lab_01.pdf
ESP-IDF 4.4 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-e91d384503485fbb54f6ce3d11e841fe
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Flash will be erased from 0x00001000 to 0x00007fff...
Flash will be erased from 0x00010000 to 0x00039fff...
Flash will be erased from 0x00008000 to 0x00008fff...
Compressed 25392 bytes to 15886...
Writing at 0x00001000... (100 %)
Wrote 25392 bytes (15886 compressed) at 0x00001000 in 0.8 seconds (effective 241.8 kbit/s)...
Hash of data verified.
Compressed 170416 bytes to 89843...
Writing at 0x00010000... (16 %)
Writing at 0x0001b102... (33 %)
Writing at 0x000208c4... (50 %)
Writing at 0x000260a3... (66 %)
Writing at 0x0002e6a2... (83 %)
Writing at 0x00036a8a... (100 %)
Wrote 170416 bytes (89843 compressed) at 0x00010000 in 2.4 seconds (effective 576.5 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 103...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.1 seconds (effective 341.7 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
Done
C:\Espressif\frameworks\esp-idf-v4.4.2\examples\get-started\hello_world>
Hello world!
Restarting in 10 seconds...
I (211) cpu_start: Starting scheduler on APP CPU.
Restarting in 9 seconds...
Restarting in 8 seconds...
Restarting in 7 seconds...
6

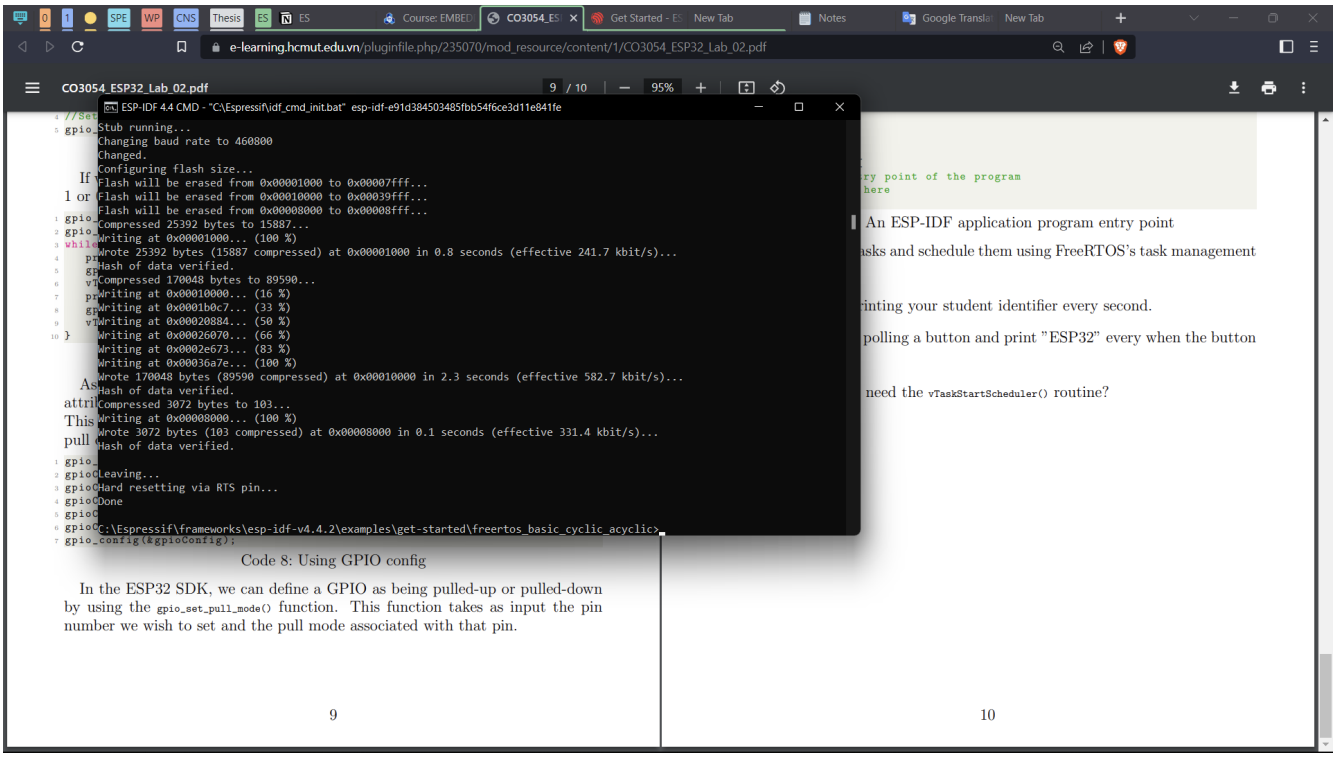

e-learning.hcmut.edu.vn/pluginfile.php/235069/mod_resource/content/1/CO3054_ESP32_Lab_01.pdf
CO3054_ESP32_Lab_01.pdf
ESP-IDF 4.4 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-e91d384503485fbb54f6ce3d11e841fe - python.exe "C:\Espressif\framewor...
I (225) cpu_start: App version: v4.4.2-dirty
I (231) cpu_start: Compile time: Nov 2 2022 05:14:59
I (237) cpu_start: ELF file SHA256: 28023f3ef8abb8bc...
I (243) cpu_start: ESP-IDF: v4.4.2-dirty
I (249) heap_init: Initializing. RAM available for dynamic allocation:
I (256) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (262) heap_init: At 3FFB2C30 len 0002D3D0 (180 KiB): DRAM
I (268) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (274) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (281) heap_init: At 4008B2E0 len 00014D20 (83 KiB): IRAM
I (288) spi_flash: detected chip: generic
I (292) spi_flash: flash io: dio
W (295) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using the size in the bi
nary image header.
I (310) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
Hello world!
This is esp32 chip with 2 CPU core(s), WiFi/BT/BLE, silicon revision 3, 2MB external flash
Minimum free heap size: 295180 bytes
Restarting in 10 seconds...
Restarting in 9 seconds...
Restarting in 8 seconds...
Restarting in 7 seconds...
Restarting in 6 seconds...
Restarting in 5 seconds...
Restarting in 4 seconds...
Restarting in 3 seconds...
Restarting in 2 seconds...
Restarting in 1 seconds...
Hello world!
Restarting in 10 seconds...
I (211) cpu_start: Starting scheduler on APP CPU.
Restarting in 9 seconds...
Restarting in 8 seconds...
Restarting in 7 seconds...
6

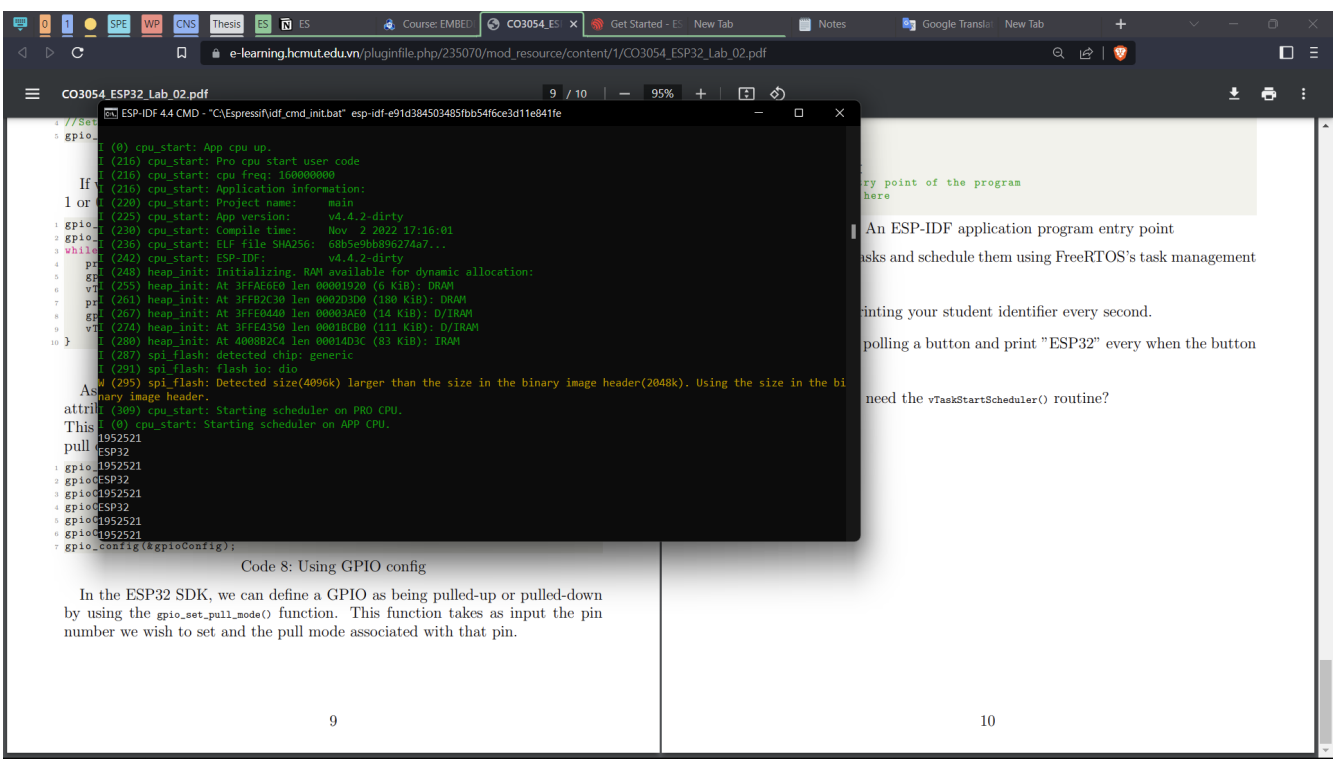
ESP-L02

Source code:

```c
#include <stdio.h>
#include <stdbool.h>
#include <unistd.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("1952521\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);
}
```

Results:





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

→ No, because the startup flow of an ESP-IDF application will call this automatically.