REPORT LAB

EMBEDDED SYSTEM - CO3054

Group: CC02

Student: Dương Gia An – 1952163

1. INTRODUCTION TO ESP32 AND ESP-IDF
2. After install ESP-IDF extension on VS code, I create a project with HelloWorld Template.

A screenshot of a computer

Description automatically generated

1. Build the Project and connect ESP32 (COM3).

Text

Description automatically generated

Finish Build

Text

Description automatically generated

1. Flash (UART) project code to ESP32 by Press BOOT button on ESP32 while Flashing.

Text

Description automatically generated

1. Press Monitor Device to see ESP execute flashed code.

Text

Description automatically generated

1. **ESP32 GPIO AND FREERTOS TASK**
2. #include <stdio.h>
3. #include "sdkconfig.h"
4. #include "freertos/FreeRTOS.h"
5. #include "freertos/task.h"
6. #include "esp\_system.h"
7. #include "esp\_spi\_flash.h"

10. **void** print\_id(**void** \*pvParameter){
11. **while**(1){
12. **printf**("DUONG GIA AN : %d \n",1952163);
13. vTaskDelay(1000/portTICK\_PERIOD\_MS);
15. }
16. vTaskDelete(NULL);
18. }
20. **void** blinky(**void** \*pvParameter){
21. **while**(1){
22. **printf**("Press Button\n");
23. **int**  rd = **rand**() % (5000 + 1 - 0) + 0;
24. vTaskDelay(rd /portTICK\_PERIOD\_MS);
25. }
26. vTaskDelete(NULL);
27. }

30. **void** app\_main(){
31. xTaskCreate(&print\_id, "print\_id", 2048, NULL, 0, NULL);
32. xTaskCreate(&blinky, "blinky", 2048,NULL,0,NULL );
34. **for** (**int** i = 20; i >= 0; i--) {
35. **printf**("Remaing %d seconds...\n", i);
36. vTaskDelay(1000 / portTICK\_PERIOD\_MS);
37. }
38. **printf**("Restarting now.\n");
39. vTaskDelay(5000 / portTICK\_PERIOD\_MS);
40. **fflush**(stdout);
41. esp\_restart();
42. }

Link Github:

***Explain:***

* + Cyclic task: void **print\_id()** is task that print my student ID every 2 seconds.
  + Acylic task: void **Blinky()** is alternated for button in GPIO in ESP32. I change to a random time to press button from 0 – 5000ms.
  + **app\_main()** will print time stamp every 1 second and restart ESP after 20 seconds.

The *priority* and *usStackDepth* of task cyclic/acylic is the same as 0 and 2048 (mean 2048\*4 bytes will be allocated for these tasks).

***Result:*** Graphical user interface, text

Description automatically generated