**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY**

**COMPUTER ENGINEERING**

Logo, company name

Description automatically generated

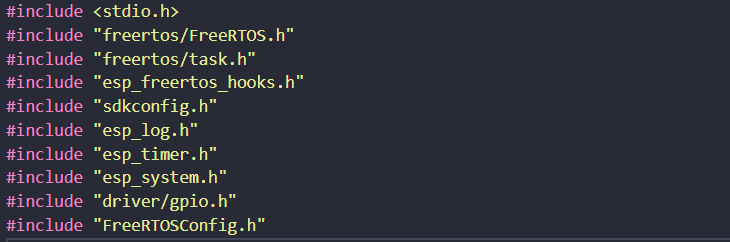
**EMBEDDED SYSTEM**

**LAB 3**

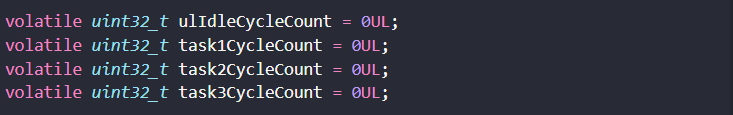
**NHÓM 10**

|  |  |  |
| --- | --- | --- |
| Họ & tên sinh viên | MSSV | Đóng góp |
| Tô Đình Lưu | 2013725 | 100% |
| Nguyễn Tôn Minh Quân | 2014276 | 100% |
| Nguyễn Phước Thịnh | 1915317 | 100% |
| Cao Trung Tín | 1915506 | 100% |

**Thư viện được bao gồm**

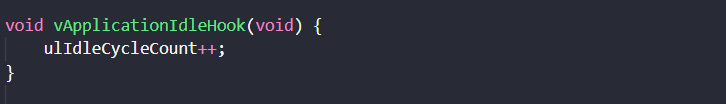
Các thư viện cần thiết cho FreeRTOS, các thành phần ESP-IDF, và các thư viện chuẩn C để sử dụng printf, timers, hàm hệ thống, vv. 

**Biến toàn cục**

ulIdleCycleCount, task1CycleCount, task2CycleCount, task3CycleCount: Các biến này lưu trữ số lần chu kỳ cho từng tác vụ tương ứng và chu kỳ idle (không làm gì). 

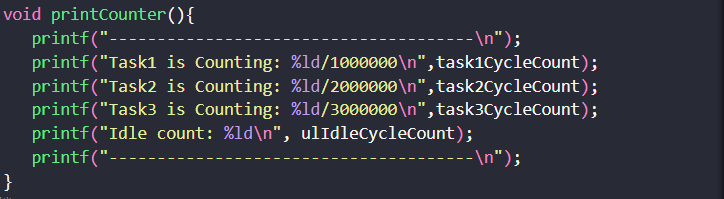
**vApplicationIdleHook():**

Hàm hook được gọi khi tác vụ idle của FreeRTOS đang chạy.

Tăng giá trị của ulIdleCycleCount để theo dõi số chu kỳ idle. 

**printCounter():**

Hàm giúp in ra số chu kỳ của mỗi tác vụ và chu kỳ idle.



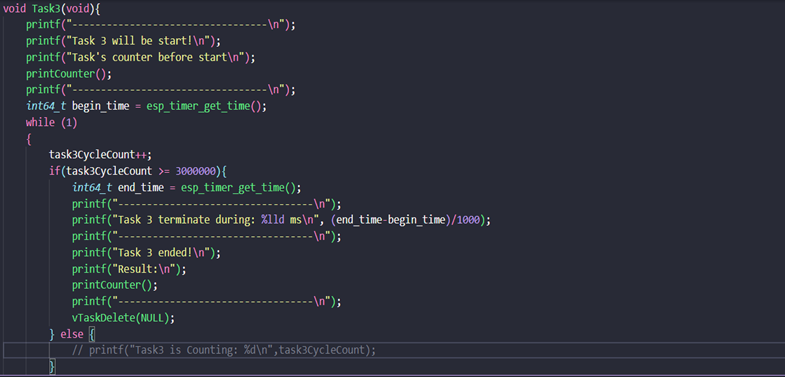
**Task1(), Task2(), Task3():**

·        Đây là ba tác vụ chạy đồng thời.

·        Mỗi tác vụ tăng giá trị của biến taskXCycleCount tương ứng cho đến khi đạt được một giá trị nhất định.

·        Khi tác vụ hoàn thành việc đếm, nó tính thời gian và in ra kết quả cùng với số chu kỳ của tất cả các tác vụ và chu kỳ idle.

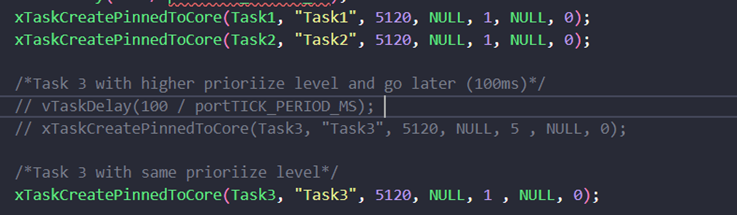
·        Các tác vụ được thiết kế để chạy mãi mãi cho đến khi đạt được giá trị nhất định.



**Idle task hook:**



**Tạo các task1, task2, task3 trong FreeRTOS và ghim chúng vào một core cụ thể trên vi điều khiển ESP32:**



**Cấu hình FreeRTOSConfig.h cho Prioritized Pre-emptive Scheduling with Time Slicing:**

#define configUSE\_PREEMPTION 1

#define configUSE\_TIME\_SLICING 1

**Cấu hình FreeRTOSConfig.h cho Prioritized Pre-emptive Scheduling (without Time Slicing):**

#define configUSE\_PREEMPTION 1

#define configUSE\_TIME\_SLICING 0

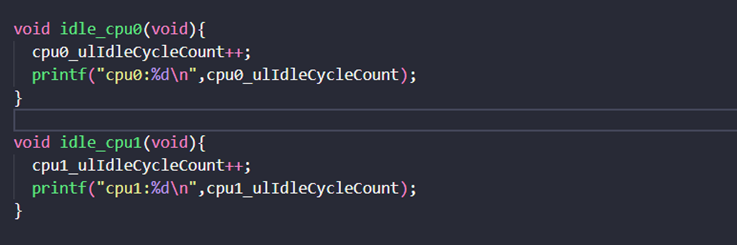
**Cấu hình FreeRTOSConfig.h cho Co-operative Scheduling:**

Đối với Co-operative Scheduling: các tác vụ sẽ chia sẻ CPU một cách tự nguyện, và các tác vụ phải tự quyết định khi nào nhường CPU cho các tác vụ khác:

#define configUSE\_PREEMPTION 0

#define configUSE\_TIME\_SLICING 0

Extra exercise:

Tăng cpu0\_ulIdleCycleCount và cpu1\_ulIdleCycleCount mỗi khi hook idle task được gọi trong CPU:



Sử dụng idle task hook để theo dõi việc sử dụng CPU dual-core.

Link git: [LuuToDinh/embedded-system (github.com)](https://github.com/LuuToDinh/embedded-system)