STM32F103C–FREERTOS-PLATFORM

TUT1: FREERTOS, TASK,

Driver UART, TIMEBASE

Address: 666/49 3 Thang 2, Ward 14, District 10, Ho Chi Minh City, Vietnam

Phone: +84986035797 | Email: [sonhai26988@gmail.com](mailto:sonhai26988@gmail.com)

# Công việc:

* Port FREERTOS v10.0.0.0 cho core ARM STM32F103C8T6
* Viết driver cho UART1: debug, tập lệnh giao tiếp
* Viết driver timebase, xung clock nền dùng cho định thì các function sẽ thực hiện

# Ý tưởng:

* Port FreeRTOS: port full function, nhưng chạy demo theo từng option từng bài học cho dễ tiếp cận
* Driver UART1: dựa trên ngắt nhận và ngắt thu, đẩy data vào RINGBUF đã được phân hoạch buffer size từ đầu chương trình. UART1 đang sử dụng mặc định để debug realtime qua terminal, update OTA hoặc hỗ trợ các AT command: info sản phẩm, cấu hình config các thông số sản phẩm, test config các kiểu…
* Driver timebase: sử dụng timer 2 vs xung clock cao, 100us ngắt 1 lần để lấy timebase tần số cao, dùng cho các function hoặc ứng dụng định thì: delay, wait...

# Thực hiện, các hàm,và cách sử dụng:

## UART1:

### Các file sử dụng:

USART1\_Function.c: tầng driver cho các hàm sử dụng: dùng init, truyền nhận data…

USART1\_Function.h:

USART1\_AppCall\_Function.c: móc pointer hàm cho các function

USART1\_AppCall\_Function.h:

PointerFunctionDefine.h:

USART1\_Function\_Profile.h: Các cấu hình cho UART, buffer size, cách thức truyền nhận

### Cách thực hiện:

Data từ UART1 đưa vào buffer qua ringbuf trong ngắt thu và ngắt nhận: USART1\_RXRingBuff và USART1\_TXRingBuff.

1 Function ngoài như UART\_Comm của chương trình bootloader sẽ process để giao tiếp vs phần mềm máy tính hay config.

## TIMEBASE

### Các file sử dụng:

Time\_Manage\_Function.c: các hàm quản lý định thì dựa trên các đơn vị: us, ms, s, rtc(s)

Time\_Manage\_Function.h:

Timer\_Function.c: móc pointer hàm cho các function

Timer\_Function.h:

Timer\_Function\_Profile.h:

### Cách thực hiện:

Sử dụng timer 2 làm timebase, chu kì ngắt 100us, tương ứng tần số 10khz

Biến sử dụng uTimeHighResolution (100us tăng 1 đơn vị), uTimeMsCounter 1 ms tăng 1 đơn vị

Trong Time\_Manage\_Function.c:

timer\_set: khởi tạo 1 chu kì cần định thì

timer\_reset: reset chu kì

timer\_restart: restart lại chu kì

timer\_expired: check timer expired

timer\_remaining: check thời gian còn lại

## TASK

### Demo 2 task led sử dụng UART1 và timer expired để hiểu rõ cấu trúc:

Task 1 tác động lên LED1

Task 1 tác động lên LED2

### Trong ví dụ này sử dụng LED1 phối hợp vs các hàm UART1 để in data debug và để định thì chớp tắt LED1:

LED1\_Task được tạo trong file main.c cho dễ ví dụ:

* LED1\_Task( void \*pvParameters );
* Chu kì của task #define LED1\_TASK\_FREQUENCY 10 /\* 10 ms \*/
* Vì 1 tick tương đương 1ms, nên 10 tick là 10ms, Task led 10ms sẽ được call 1 lần
* timer\_set(&pLED1Test,500,CLOCK\_TYPE\_MS): init chu kì timer là 500ms sẽ toggle led 1 lần và in ra UART1

## BOOTLOADER:

### Địa chỉ bootloader FW\_FLASH\_ADDR: 0x08005000

### Địa chỉ chứa thông tin sản phẩm: USER\_INFO\_FLASH\_ADDR: 0x08004800