**TÀI LIỆU THƯ VIỆN SYSTICK – STM32F103C8T6 (STD LIB v3.5)**

**1. Mục đích**

Thư viện systick.c/h cung cấp các hàm:

* delay\_ms(), delay\_us() – delay dạng blocking
* millis(), micros() – đo thời gian thực
* delay\_ms\_non\_blocking() – delay không chặn CPU
* is\_delay\_done() – kiểm tra delay không chặn đã kết thúc

Ứng dụng:

* Chớp LED, đo thời gian xử lý, debounce nút nhấn, PWM thủ công, giao tiếp UART tự chế, xử lý đa nhiệm đơn giản…

**2. Khởi tạo SysTick – SysTick\_Init()**

void SysTick\_Init(uint32\_t ms);

Sử dụng macro SysTick\_Config() để:

* Tự động tính toán và nạp giá trị LOAD
* Tạo ngắt định kỳ mỗi ms (thường là 1ms)

**Ví dụ:**

SysTick\_Init(1); // ngắt mỗi 1ms

**3. Hàm delay\_ms()**

void delay\_ms(uint32\_t ms);

* Dùng biến sysTickCounter tăng mỗi 1ms trong SysTick\_Handler
* Dạng **blocking** – CPU chờ đến khi hết thời gian

**Ví dụ:**

GPIO\_SetBits(GPIOC, GPIO\_Pin\_13);

delay\_ms(1000);

GPIO\_ResetBits(GPIOC, GPIO\_Pin\_13);

**4. Hàm delay\_us()**

void delay\_us(uint32\_t us);

* Dùng vòng lặp \_\_NOP() hoặc nội suy dựa trên tần số CPU
* Tạo độ trễ ngắn chính xác tương đối

**Ví dụ:**

delay\_us(500); // delay 500 micro giây

**5. Hàm millis()**

uint32\_t millis(void);

* Trả về số mili giây đã trôi qua kể từ khi SysTick được bật

**Ví dụ:**

uint32\_t t0 = millis();

// làm gì đó

uint32\_t t1 = millis();

uint32\_t elapsed = t1 - t0;

**6. Hàm micros()**

uint32\_t micros(void);

* Trả về số micro giây đã trôi qua
* Kết hợp:
  + sysTickCounter \* 1000
  + (LOAD - VAL) / (ticks\_per\_us)
* Chống lỗi sai số khi SysTick->VAL tràn

**Ví dụ:**

uint32\_t start = micros();

while (micros() - start < 200); // delay khoảng 200us

**7. Delay không chặn – delay\_ms\_non\_blocking() + is\_delay\_done()**

void delay\_ms\_non\_blocking(uint32\_t ms);

uint8\_t is\_delay\_done(void);

* Cho phép **thực hiện công việc khác trong lúc delay**
* Không giữ CPU chờ

**Ví dụ:**

delay\_ms\_non\_blocking(1000);

while (1) {

if (is\_delay\_done()) {

// delay đã xong, làm gì đó

}

// vẫn xử lý công việc khác trong khi chờ

}

**8. Ngắt SysTick – SysTick\_Handler**

void SysTick\_Handler(void) {

sysTickCounter++;

}

* Phải đặt trong stm32f10x\_it.c
* Cần thiết cho:
  + delay\_ms()
  + millis(), micros()
  + delay\_ms\_non\_blocking()

**9. Tổng hợp API**

| **Hàm** |  | **Mô tả** |
| --- | --- | --- |
| SysTick\_Init(ms) |  | Khởi tạo systick |
| delay\_ms(ms) |  | Delay dạng blocking |
| delay\_us(us) |  | Delay micro giây |
| millis() / micros() |  | Đo thời gian thực |
| delay\_ms\_non\_blocking(ms) |  | Delay không chặn |
| is\_delay\_done() |  | Kiểm tra delay đã xong |

**10. Ứng dụng thực tế**

* Blink LED chính xác
* Đo thời gian xử lý
* Chống dội nút nhấn
* PWM phần mềm
* Thực hiện đa nhiệm đơn giản không cần RTOS

**Phụ lục: Thiết lập clock**

Đảm bảo bạn đã gọi:

SystemInit(); // Thiết lập clock = 72MHz

SysTick\_Init(1); // Bắt đầu ngắt 1ms