

KHÓA HỌC LẬP TRÌNH VI ĐIỀU KHIỂN

Giảng viên NGUYỄN HUỲNH NHẬT THƯƠNG LỊCH HỌC:

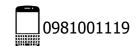
Tại Đà Nẵng: 19h30 - 22h00 thứ 2 và T6

ĐỊA ĐIỂM:

Online qua nền tảng Zoom

MODULE 12

- DMA
- WDG





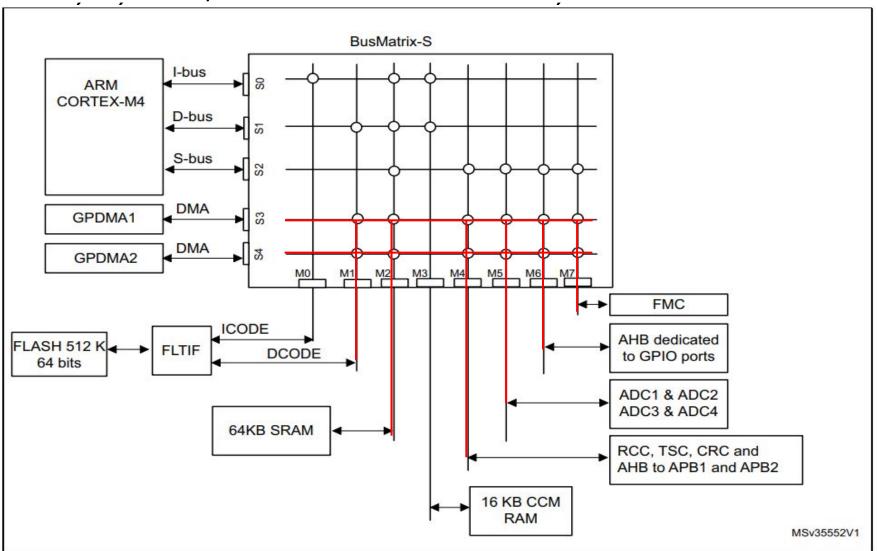








Direct Memory Access





DMA

 Các thông số: Địa chỉ nguồn, địa chỉ đích Kiểu dữ liệu của mỗi lần truyền (data width): byte, half-word, word Lượng dữ liệu truyền (transfer size) Kiểu truyền dữ liệu (transfer types): Normal mode hoặc Circular mode (ring buffer) Tự động tăng địa chỉ nguồn/địa chỉ đích hoặc cả 2 sau mỗi lần truyền dữ liệu Yêu cầu ngắt: Half-transfer hoặc Transfer complete (double buffering)



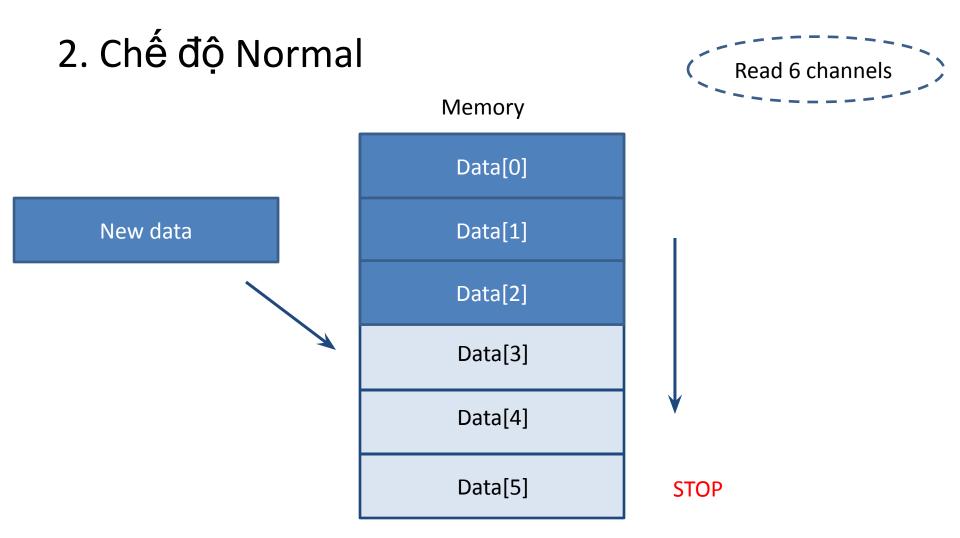
DMA

- Ưu điểm
 - DMA truyền dữ liệu nhanh hơn CPU
 - Nâng cao hiệu suất của vi điều khiển
 - DMA có khả năng tự động tăng địa chỉ lưu trữ dữ liêu
 - Tiết kiệm năng lượng tiêu thụ với các ứng dụng truyền nhận một lượng lớn dữ liệu
- Nhược điểm
 - □ Phần cứng VĐK phải hỗ trợ DMA (STM32 all)
 - Sử dụng DMA không phù hợp thì có khả năng tăng năng lượng tiêu thụ

https://www.st.com/resource/en/application_note/dm00046011-using-the-stm32f2-stm32f4-and-stm32f7-series-dma-controller-stmicroelectronics.pdf



CÁC CHẾ ĐỘ HOẠT ĐỘNG CỦA DMA

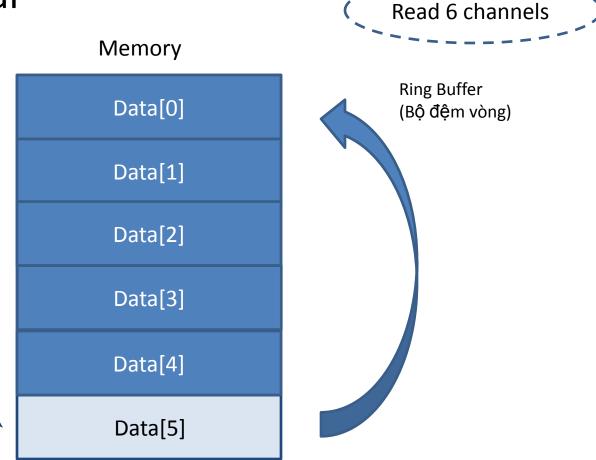


Khi DMA dừng hoạt động, cần gọi lại lệnh bắt đầu DMA để tiếp tục truyền dữ liệu lại từ đầu



CÁC CHẾ ĐỘ HOẠT ĐỘNG CỦA DMA

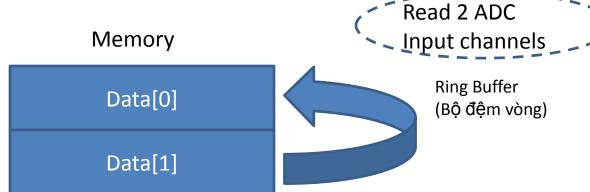
1. Chế độ Circular

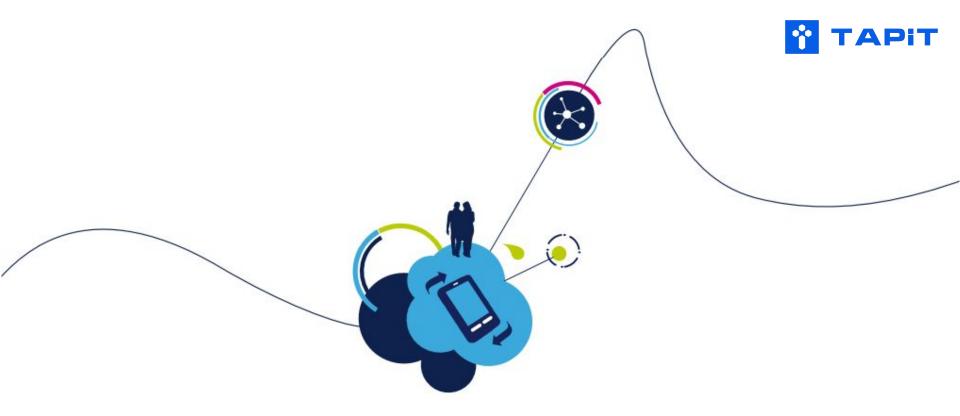




CÁC CHẾ ĐỘ HOẠT ĐỘNG CỦA DMA

1. Chế độ Circular





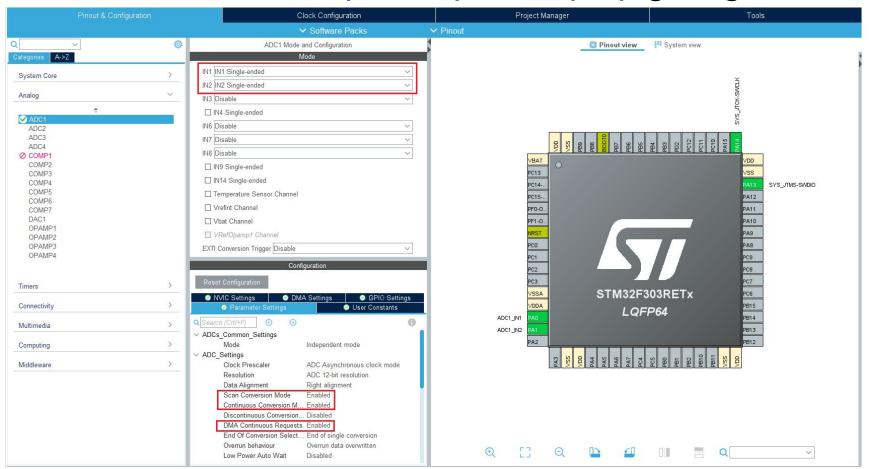
ADC DMA Poll lab

How to setup ADC DMA in CubeMX and Generate Code How to Generate Code in CubeMX and use HAL functions



CẤU HÌNH DMA QUA GIAO DIỆN CUBEMX

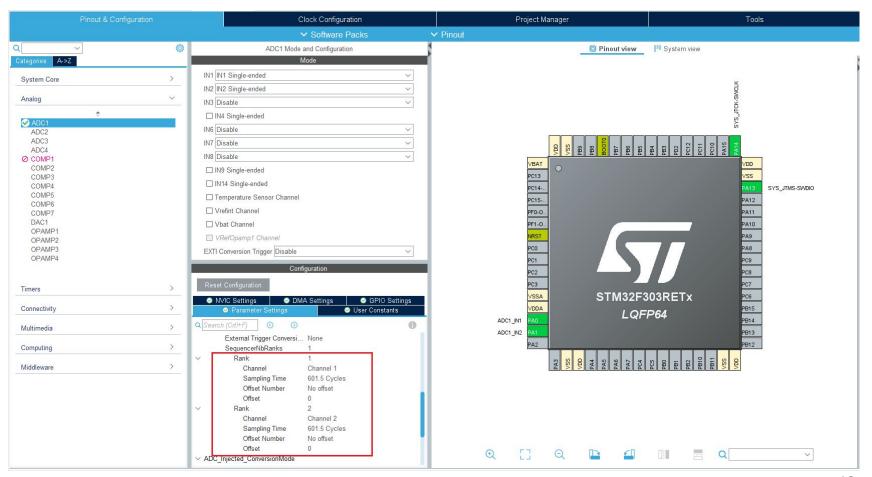
- ADC: Multichannel (2 channel), continous mode Lưu mẫu mới nhất vào địa chỉ một cách tự động bằng DMA





CẤU HÌNH DMA QUA GIAO DIỆN CUBEMX

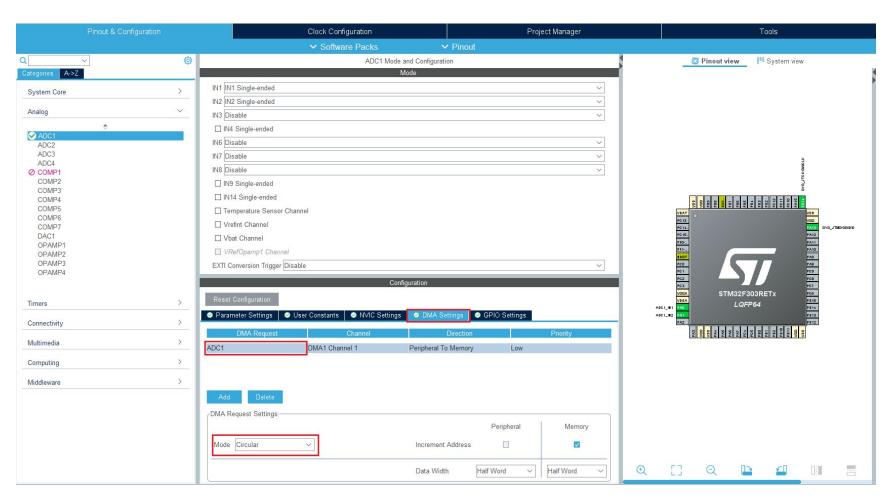
- ADC: Rank number, Sampling Time





CẤU HÌNH DMA QUA GIAO DIỆN CUBEMX

- DMA: Mode Circular, Data width: Half-word



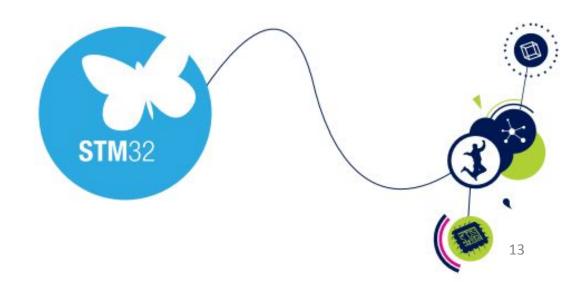


Use ADC in DMA mode

```
HAL ADC Start DMA(ADC HandleTypeDef* hadc, uint32 t* pData, uint32 t Length)
HAL_ADC_Stop_DMA(ADC_HandleTypeDef* hadc)
/* USER CODE BEGIN PV */
    uint16 t adc value[2] = \{0\};
/* USER CODE END PV */
 MX DMA Init();
 MX ADC1 Init();
/* USER CODE BEGIN 2 */
HAL_ADC_Start_DMA(&hadc1, (uint32_t*)adc_value, 2);
 /* Infinite loop */
while (1)
/* USER CODE END 3 */
```



MODULE 12 - UART DMA



Ngắt UART IDLE DMA

TAPIT

- Truyền 1 lượng lớn dữ liệu:
- Truyền file từ máy tính xuống
- Tải 1 file từ internet về (âm thanh, firmware, file cấu hình)

https://tapit.vn/huong-dan-su-dung-chuc-nang-uart-idle-dma/

DMA:

- Một master trên hệ thống bus

Giao tiếp với 1 thiết bị B.

Thiết bị B định kì 2s gửi ra 1 chuỗi dữ liệu không cố định kích cỡ, và cũng không có kí tự kết thúc.

Làm sao để biết là A đã nhận được 1 chuỗi hoàn chỉnh từ B?

- Không dùng ngắt IDLE.
- Không có kí tự kết thúc
- Không cố định?
- Dùng thêm 1 cái timer.
- Mỗi lần vào ngắt uart thì reset 1 biến thời gian.
- Và timer định kì kiểm tra biến thời gian.
- Nếu sau 1 khoản thời gian hơn thời gian nhận 1 byte mà biến k bị reset thì chứng tỏ không nhận thêm dữ liệu, chứng tỏ đã nhận 1 chuỗi hoàn chỉnh

14



Microphone

Thu âm thanh:

Tần số lấy mẫu âm thanh: 16KHz. -> 16.000 mẫu / 1s. 16ksps

Hệ thống yêu cầu khả năng realtime, vừa lấy âm thanh vừa xử lý DSP.

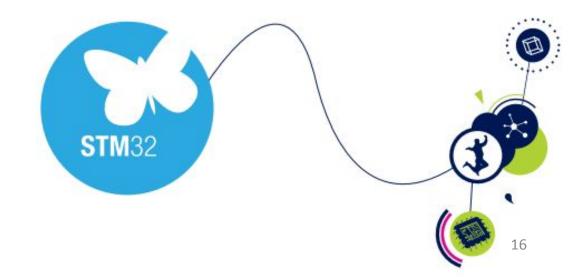
Nếu dùng vi xử lý để lấy mẫu thì vi xử lý không thể đi xử lý âm thanh được.

Dẫn đến cần dùng DMA để lấy mẫu âm thanh từ sensor micro và chuyển vào bộ nhớ. VI xử lý sẽ phối hợp khi đủ lượng mẫu là xử lý ngay. (Ví dụ 16.000 mẫu xử lý 1 lần)

Dùng DMA. Khai báo 1 kích cỡ dữ liệu là 32.000 mẫu. Và dùng ngắt Half transfer và ngắt full transfer.:



MODULE 12 - WATCHDOG TIMER



INDEPENDENT WATCHDOG (IWDG)



- The Independent watchdog peripheral detects and solves malfunctions due to software failure, and triggers system reset when the counter reaches a given timeout value
- The independent watchdog (IWDG) is clocked by its own dedicated low-speed clock (LSI) and thus stays active even if the main clock fails
- Free-running downcounter
- Clocked from an independent RC oscillator (can operate in Standby and Stop modes)
- Conditional Reset
 - ☐ Reset (if watchdog activated) when the downcounter value becomes lower than 0x000
 - Reset (if watchdog activated) if the downcounter is reloaded outside the window

timeout =
$$\frac{\text{(prescaler divider)*(reload value)}}{f_{LSI}}$$
 (s)

16



Sự cố phần mềm bị treo là gì? Là vi xử lý không thực thi được tới một lệnh nào đó mong muốn.

Lỗi phần mềm:

Một số lỗi phần mềm có thể làm cho vi xử lý thực thi đến 1 lệnh nào đơ mong muốn chậm hơn dự kiến Hoặc nhanh hơn dự kiến

Luôn có giới hạn để thực thi xong một khối lệnh nào đó.

Nguyên nhân:

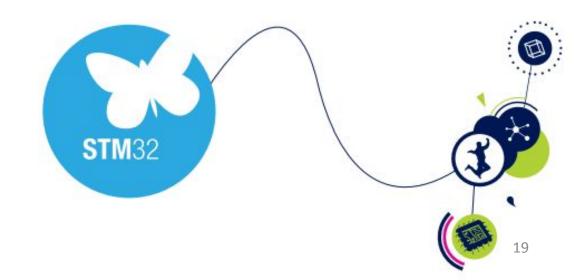
- Nhiễu điện từ
 - Do code sai
 - Do hỏng phần cứng và code k đủ tốt để xử lý
 - Lưu ý với các vòng lặp vô hạn

IDG để kích hoạt reset vi xử lý khi có sự cố treo (không thực thi được đến câu lệnh nào đó trong khoảng thời gian kì vọng) phần mềm



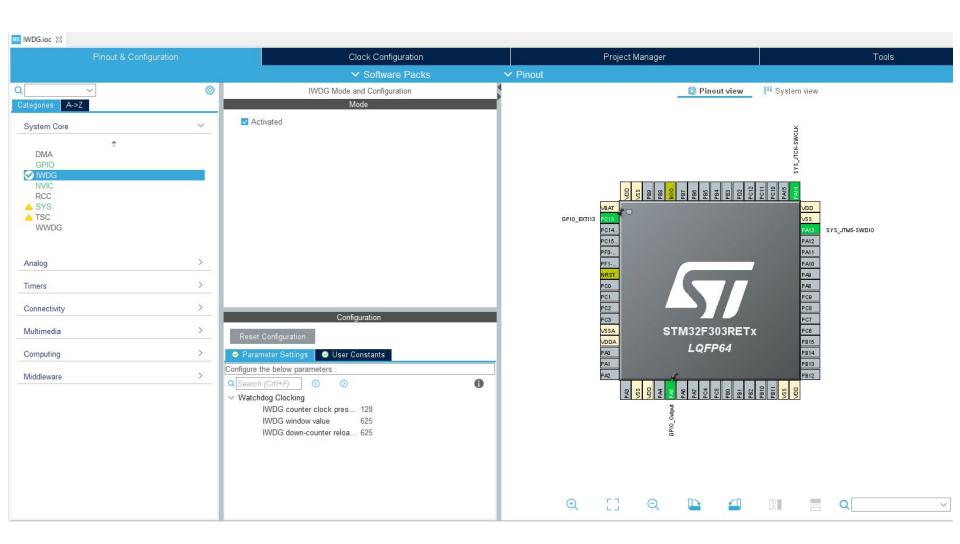


MODULE 12 - THỰC HÀNH IWDG



GIAO DIỆN CẦU HÌNH









```
/* USER CODE BEGIN 0 */
void HAL GPIO EXTI Callback(uint16 t GPIO Pin)
    if(GPIO Pin == GPIO PIN 13)
        while(1);
/* USER CODE BEGIN WHILE */
 while (1)
     HAL GPIO TogglePin(GPIOA, GPIO PIN 5);
     HAL_Delay(500);
     HAL IWDG Refresh(&hiwdg);
  /* USER CODE END WHILE */
```

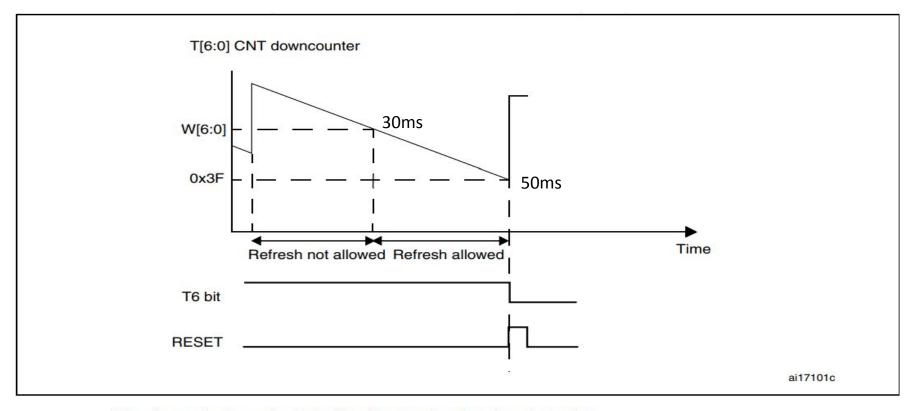




- The system window watchdog (WWDG) is used to detect the occurrence of a software fault
- The watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the contents of the downcounter before the T6 bit becomes cleared
- An MCU reset is also generated if the 7-bit downcounter value (in the control register) is refreshed before the downcounter has reached the window register value
- Conditional Reset
 - ☐ Reset (if watchdog activated) when the downcounter value becomes less than 0x40
 - ☐ Reset (if watchdog activated) if the downcounter is reloaded outside the window
- Early wakeup interrupt (EWI): triggered (if enabled and the watchdog activated) when the downcounter is equal to 0x40







The formula to calculate the timeout value is given by:

$$t_{WWDG} = t_{PCLK1} \times 4096 \times 2^{WDGTB[1:0]} \times (T[5:0] + 1)$$
 (ms)

where:

t_{WWDG}: WWDG timeout

t_{PCLK}: APB1 clock period measured in ms

4096: value corresponding to internal divider







Instructor

Eng. Nguyen Huynh Nhat Thuong

