



Trường ĐH Sư Phạm Kỹ Thuật TP. Hồ Chí Minh
Khoa Cơ khí chế tạo máy

Môn học

TT Vi xử lý

Giảng viên: Huỳnh Quang Duy

Email: duyhq.ute@gmail.com

VI ĐIỀU KHIỂN STM32

Tuần 1

Thiết lập chung cho STM32

Đệ Lập Trình được cho STM32 cần đảm bảo được các bước sau.

Step 1: Cấp xung clock cho bộ cho thành phần ngoại vi (GPIO, Timer, USART,..)

Step 2. Thiết lập chức năng và các thông số cần thiết cho các thành phần ngoại vi tương ứng

Step 3: Tiến hành lập trình bằng ngôn ngữ C và các thanh ghi liên quan.

Tương tác với thanh ghi

UTE

Ví dụ: dòng lệnh cơ bản tương tác với 1 thanh ghi.

```
GPIOB -> ODR |= (1<<0) ;
```

Module cần truy cập/tương tác

Thanh ghi thuộc phạm vi quản lý/hoạt động của module

Phép toán OR bit trên thanh ghi

Ghi giá trị 1 vào thanh ghi và dịch qua trái 0 bit

Các bước thiết lập GPIO.

Step 1: Cấp xung clock cho bộ GPIOx (RCC register)

Step 2. Thiết lập chức năng của chân cần sử dụng (input/output/...)

Step 3: Đọc/Ghi với chân IO đã cấu hình

Step 1. Cấp xung hoạt động cho module GPIOx (x: A, B, C, D,...)

Được quản lý bởi module RCC (Reset and Clock Control). Mỗi một module GPIO nếu muốn hoạt động được thì cần phải được cấp xung (clock) từ nguồn xung của hệ thống. Tùy thuộc dòng vi điều khiển STM32 mà ta có thanh ghi quản tương ứng thuộc RCC.

❖ F1 Series: APB2ENR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Reserved																
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Res.	USART 1EN	Res.	SPI1 EN	TIM1 EN	ADC2 EN	ADC1 EN	Reserved			IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.	AFIO EN
	rw		rw	rw	rw	rw				rw	rw	rw	rw	rw		

❖ F2,3,4 Series: AHB2ENR

Ví dụ:

RCC

->

AHB1ENR

|=

(1<<0)

;

Step 2. Thiết lập chức năng input/output

Cấu hình input/output chân GPIO cần sử dụng. Tùy thuộc vào dòng chip mà ta có cách cấu hình khác nhau

❖ F1 Series: $GPIOx_CRL$ or $GPIOx_CRH$

$GPIOx_CRL$: cho chân IO từ 0 đến 8 ----- $GPIOx_CRH$: cho chân IO từ 8 đến 16

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]		MODE7[1:0]		CNF6[1:0]		MODE6[1:0]		CNF5[1:0]		MODE5[1:0]		CNF4[1:0]		MODE4[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3[1:0]		MODE3[1:0]		CNF2[1:0]		MODE2[1:0]		CNF1[1:0]		MODE1[1:0]		CNF0[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

❖ CNF_x: Lựa chọn chế độ hoạt động cho chân IO. Input hoặc Output

❖ MODE_x: Lựa chọn chế độ hoạt động tương ứng với mode input or output

Step 2. Thiết lập chức năng input/output

❖ F1 Series: *GPIOx_CRL or GPIOx_CRH*

MODEx

- 00: Input mode (reset state)
- 01: Output mode, max speed 10 MHz.
- 10: Output mode, max speed 2 MHz.
- 11: Output mode, max speed 50 MHz.

CNFx

In input mode (**MODE[1:0]=00**):

- 00: Analog mode
- 01: Floating input (reset state)
- 10: Input with pull-up / pull-down
- 11: Reserved

In output mode (**MODE[1:0] > 00**):

- 00: General purpose output push-pull
- 01: General purpose output Open-drain
- 10: Alternate function output Push-pull
- 11: Alternate function output Open-drain

Ví dụ:

GPIOA	->	CRL	 =	(1<<0)	;	// Output 2MHz at IO pin GPIOA_0
GPIOB	->	CRH	 =	(0b0001<<0)	;	// Output 2MHz at IO pin GPIOB_8
GPIOC	->	CRL	 =	(8<<0)	;	// Input 2MHz at IO pin GPIOC_0

Step 2. Thiết lập chức năng input/output

❖ *F2 Series: Theo trình tự: GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEED, (GPIOx_PUPDR)*

GPIOx_MODER: Thanh ghi cho phép thiết lập input/output/... trên các chân IO được đánh số từ 0 đến 15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

00: Input (Reset state)

01: General purpose output mode

10: Alternate function mode

11: Analog mode

Ví dụ:

```

GPIOA -> MODER |= (1<<0) ; // Output at IO pin GPIOA_0
GPIOB -> MODER |= (0b01<<28) ; // Output at IO pin GPIOB_14
    
```


Step 2. Thiết lập chức năng input/output

❖ *F2 Series: Theo trình tự: GPIOx_MODER, (GPIOx_OTYPER), GPIOx_OSPEED, (GPIOx_PUPDR)*

GPIOx_OTYPER: Thanh ghi cho phép thiết lập các chế độ hoạt động tương ứng với Output mode... trên các chân IO được đánh số từ 0 đến 15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

0: Output push-pull (Reset state)

1: Output open-drain

Ví dụ:

```

GPIOA -> OTYPER &= ~(1<<0) ; // Push-pull at IO pin GPIOA_0
GPIOB -> OTYPER |= 1<<14) ; // Open-drain at IO pin GPIOB_14
    
```

Step 2. Thiết lập chức năng input/output

❖ *F2 Series: Theo trình tự: GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEED, (GPIOx_PUPDR)*

GPIOx_OSPEEDR: Thanh ghi cho phép thiết lập các tốc độ hoạt động trên các chân IO được đánh số từ 0 đến 15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]		OSPEEDR14 [1:0]		OSPEEDR13 [1:0]		OSPEEDR12 [1:0]		OSPEEDR11 [1:0]		OSPEEDR10 [1:0]		OSPEEDR9 [1:0]		OSPEEDR8 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1 [1:0]		OSPEEDR0 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

00: Low speed

01: Medium speed

10: High speed

11: Very high speed

Ví dụ:

```

GPIOA -> OSPEEDR |= (1<<0) ; //Medium speed at IO pin GPIOA_0
GPIOB -> OSPEEDR |= (3<<2*14) ; // High speed at IO pin GPIOB_14
    
```

Step 2. Thiết lập chức năng input/output

❖ *F2 Series: Theo trình tự: GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEED, GPIOx_PUPDR*

GPIOx_PUPDR: Thanh ghi cho phép thiết lập chế độ treo điện trở cao/thấp trên các chân Input được đánh số từ 0 đến 15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]	PUPDR14[1:0]	PUPDR13[1:0]	PUPDR12[1:0]	PUPDR11[1:0]	PUPDR10[1:0]	PUPDR9[1:0]	PUPDR8[1:0]	PUPDR7[1:0]	PUPDR6[1:0]	PUPDR5[1:0]	PUPDR4[1:0]	PUPDR3[1:0]	PUPDR2[1:0]	PUPDR1[1:0]	PUPDR0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR15[1:0]	PUPDR14[1:0]	PUPDR13[1:0]	PUPDR12[1:0]	PUPDR11[1:0]	PUPDR10[1:0]	PUPDR9[1:0]	PUPDR8[1:0]	PUPDR7[1:0]	PUPDR6[1:0]	PUPDR5[1:0]	PUPDR4[1:0]	PUPDR3[1:0]	PUPDR2[1:0]	PUPDR1[1:0]	PUPDR0[1:0]
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

00: No pull

01: Pull-up

10: Pull-down

11: Reserved

Ví dụ:

```

GPIOA -> PUPDR |= (1<<0) ; //Pull-up at IO pin GPIOA_0
GPIOB -> PUPDR |= (2<<28) ; // Pull-down at IO pin GPIOB_14
    
```

Step 3. Đọc tín hiệu

❖ F1,2,3,4 Series: GPIOx_IDR

GPIOx_IDR: Thanh ghi cho phép đọc tín hiệu logic trên các chân Input được đánh số từ 0 đến 15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Ví dụ:

```
if ((( GPIOA -> IDR ) & (1<<0) )) == 0 )
```

//Check input logic at Pin GPIOA_0 is Low(0) ???

Step 3. Ghi tín hiệu

❖ *F1,2,3,4 Series: GPIOx_ODR, GPIOx_BSRR*

GPIOx_ODR: Thanh ghi cho phép ghi tín hiệu logic low (0) hoặc high (1) trên các chân Input được đánh số từ 0 đến 15

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Ví dụ:

```

GPIOB -> ODR |= (1<<14) ; //Write HIGH (1) at pin GPIOB_14
GPIOB -> ODR &= ~(1<<14) ; // Write LOW (0) at pin GPIO_14
    
```

Step 3. Ghi tín hiệu

❖ *F1,2,3,4 Series: GPIOx_ODR, GPIOx_BSRR*

GPIOx_BSRR: Thanh ghi cho phép ghi tín hiệu logic reset (0) hoặc hoặc set (1) trên các chân Output từ 0 đến 15 theo thứ tự các bit tương ứng: Từ 0 đến 15 (set) và từ 16 đến 31 (reset).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

→ *Set logic at PINx to HIGH*

→ *Reset logic at PINx to LOW*

Ví dụ: **GPIOB** -> **BSRR** |= **(1<<0)** ; //Write HIGH (1) at pin GPIOB_0

GPIOB -> **BSRR** |= **(1<<28)** ; // Write LOW (0) at pin GPIO_14

Mô hình băng tải – STM32F205VCT6

- Bài tập 1** Viết chương trình chớp tắt led đơn ở chân GPIOB0
- Bài tập 2** Viết chương trình đọc nút nhấn P1 và chớp tắt led đơn
- Bài tập 3** Viết chương trình nhấn nút thực hiện yêu cầu sau
- Nhấn nút P1 (không giữ) băng tải quay
 - Nhấn nút P2 (không giữ) băng tải dừng
- Bài tập 4** Viết chương trình xử lý tín hiệu trả về từ cảm biến S1, S2 theo yêu cầu sau
- Nếu S1 bị ngừng tác động, Băng tải quay
 - Nếu S2 bị ngừng tác động, Băng tải dừng

Mô hình xy-lanh, khí nén– STM32F205VCT6

Bài tập 1 Viết chương trình thực hiện theo yêu cầu sau

Nhấn nút Start (Màu xanh)

(Xy-lanh 1 ở vị trí S3, tay kẹp đóng lại, xy-lanh 2 ở vị trí phía trên)

Khi có sản phẩm được đưa vào khay thì:

Tay kẹp mở ra => Xy-lanh 1 tiến ra vị trí S4 => Xy-lanh 2 đi xuống vị trí dưới => Tay kẹp đóng lại (kẹp vật) => Xy-lanh 2 đi lên => Xy-lanh 1 tiến về vị trí S3 => Xy-lanh 2 đi xuống => Tay kẹp mở ra (thả vật) => Xy-lanh 2 lên => Tay kẹp đóng lại => Tiếp tục chờ khi có sản phẩm mới.

Mô hình LED, Nút nhấn, LCD – STM32F103VCT6

- Bài tập 1** Viết chương trình hiển thị số bất kỳ lên LED 7 đoạn
- Bài tập 2** Viết chương trình hiển thị số lần nhấn nút (từ 0 đến 99()) lên LED 7 đoạn