

Real-time performance Cortex-M with F _{max} 168 MHz/210 DMIPS	Outstanding power efficiency < 1 µA V _{DD} , RTC, Ultra low dynamic consumption, 1.7 to 3.6 V V _{DD}	Superior and innovative peripherals Faster peripherals, 2 full duplex PS, RTC with sub second accuracy...	Extensive tools and software CMSIS DSP library, Matlab support, various IDE starter kits, RTOS and stacks
--	--	---	---



STM32F103xC, STM32F103xD, STM32F103xE

High-density performance line Arm®-based 32-bit MCU with 256 to 512KB Flash, USB, CAN, 11 timers, 3 ADCs, 13 communication interfaces

DAY 1 – Input Output Port

Website: elec2pcb.com - Email: elec2pcb@gmail.com - Mobile: 0905 912 019

Chuẩn bị trước khi tham gia khóa học

1. Phần mềm Visual Studio Code ([tải tại đây](#)), hướng dẫn cài đặt ([tải tại đây](#))
2. Phần mềm nạp STM32 ST-Link Utility ([tải tại đây](#))
3. Driver cho mạch nạp ST-Link ([tải tại đây](#))
4. Phần mềm nạp Bootloader ([tải tại đây](#))
5. Driver cho module USB-UART chip CP2102 ([tải tại đây](#))
6. Tài liệu đọc của khóa học ([tải tại đây](#))
7. Tài liệu video của khóa học ([tải tại đây](#))
8. Diễn đàn trao đổi trong và sau khi học xong ([Link đến](#))



-
-
-
-
-
-



➤ Power supply scheme

- 5



➤ Power supply overview



1. Cấp nguồn cho STM32

➤ Low-power modes

By default, the microcontroller is in Run mode after a system or a power Reset. Several low-power modes are available to save power when the CPU does not need to be kept running, for example when waiting for an external event.

The STM32F10xxx devices feature three low-power modes:

- Sleep mode (CPU clock off, all peripherals including Cortex®-M3 core peripherals like NVIC, SysTick, etc. are kept running)
- Stop mode (all clocks are stopped)
- Standby mode (1.8V domain powered-off)

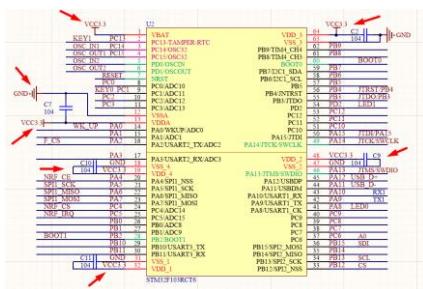
The RTC can be used to wakeup the MCU from low-power mode without depending on an external interrupt (Auto-wakeup mode)

www.POL.com

7

1. Cấp nguồn cho STM32

➤ Phân bố chân nguồn trên chip STM32F103



www.POL.com

8

2. RESET cho STM32

➤ System reset

A system reset is generated when one of the following events occurs:

1. A low level on the NRST pin (external reset)
2. Window watchdog end of count condition (WWDG reset)
3. Independent watchdog end of count condition (IWDG reset)
4. A software reset (SW reset)
5. Low-power management reset

www.POL.com

9

2. RESET cho STM32

➤ Power reset

A power reset is generated when one of the following events occurs:

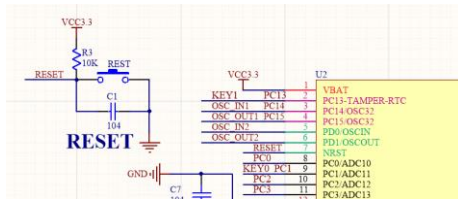
1. Power-on/power-down reset (POR/PDR reset)
2. When exiting Standby mode



10

2. RESET cho STM32

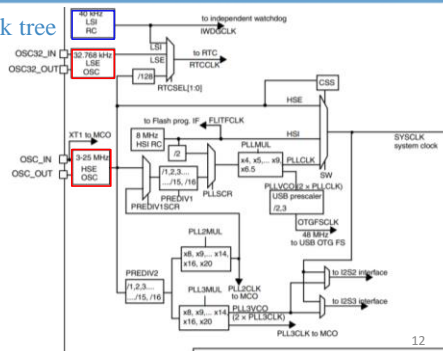
➤ MẠCH RESET



11

3. Các bộ dao động

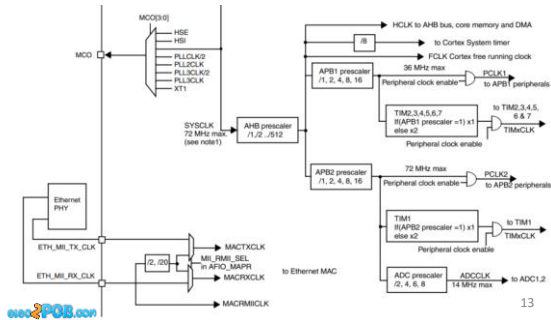
➤ Clock tree



12

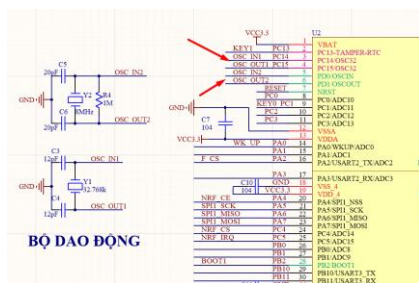
3. Các bộ dao động

➤ Clock tree

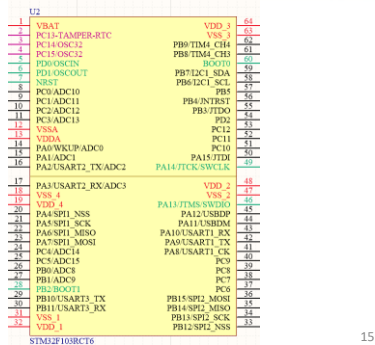


3. Các bộ dao động

➤ Mạch dao động cho STM32F103

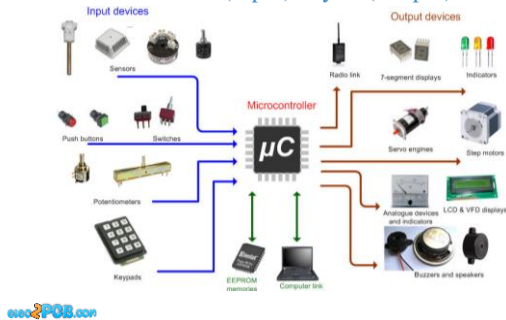


4. Vào ra đa chức năng GPIO



4. Vào ra đa chức năng GPIO

➤ Khi nào thì vào (Input) hay ra (Output) ?



16

4. Vào ra đa chức năng GPIO

- Each of the general-purpose I/O ports has:

- two 32-bit configuration registers (GPIOx_CRL, GPIOx_CRH)
- two 32-bit data registers (GPIOx_IDR, GPIOx_ODR)
- a 32-bit set/reset register (GPIOx_BSRR)
- a 16-bit reset register (GPIOx_BRR)
- a 32-bit locking register (GPIOx_LCKR)

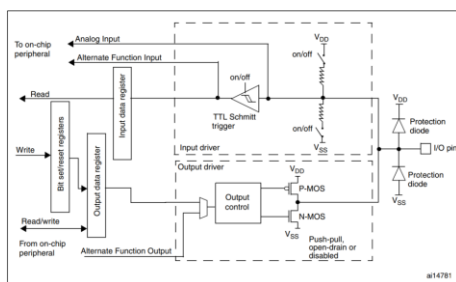
➤ Main several modes:

- Input floating
- Input pull-up
- Input-pull-down
- Analog
- Output open-drain

17

4. Vào ra đa chức năng GPIO

- Basic structure of a standard I/O port bit



18

4. Vào ra đa chức năng GPIO

➤ Basic structure of a standard I/O port bit

Table 20. Port bit configuration table

Configuration mode	CNF1	CNF0	MODE1	MODE0	PxODR register
General purpose output	0	0	01	0	0 or 1
Open-drain		1		10	0 or 1
Alternate Function output	1	0	11	0	Don't care
Open-drain		1		1	Don't care
Analog	0	0	00	0	Don't care
Input floating	0	1		0	Don't care
Input pull-down	1	0		0	0
Input pull-up	1	0	0	1	1

Table 21. Output MODE bits

MODE[1:0]	Meaning
00	Reserved
01	Maximum output speed 10 MHz
10	Maximum output speed 2 MHz
11	Maximum output speed 50 MHz

oioo3POB.com

19

4. Vào ra đa chức năng GPIO

➤ Các hàm trong Mbed

Cú pháp

DigitalOut (PinName pin)

Ví dụ

Khai báo:	DigitalOut Led(PD_2);
Sử dụng:	Led =0; wait_ms(100); Led =1;

oioo3POB.com

20

4. Vào ra đa chức năng GPIO

➤ Các hàm trong Mbed

Cú pháp

BusOut (PinName p0, PinName p1=NC, PinName p2=NC, PinName p3=NC, PinName p4=NC, PinName p5=NC, PinName p6=NC, PinName p7=NC, PinName p8=NC, PinName p9=NC, PinName p10=NC, PinName p11=NC, PinName p12=NC, PinName p13=NC, PinName p14=NC, PinName p15=NC)

Ví dụ

```
// 7seg LED - a, b, c, d, e, f, g, DP
BusOut LED7seg(PB_0, PA_2, PA_7, PB_1, PB_11, PC_4, PA_5, PC_5);
BusOut EN_LED(PA_3, PA_4, PA_6, PB_10);
```

```
EN_LED = 0b0001;
LED7seg = 7segcode[5];
EN_LED = 0b0010;
LED7seg = 7segcode[7];
```

oioo3POB.com

21

4. Vào ra đa chức năng GPIO

➤ Các hàm trong Mbed

Cú pháp

DigitalIn (PinName pin)

Ví dụ

Khai báo:	DigitalIn KEY1(PC_13);
Set mode:	PullUp/PullDown/PullNone/OpenDrain
Mode:	KEY1.mode(PullNone);
Đọc:	Value = KEY1.read();

oOo3POB.com

22

4. Vào ra đa chức năng GPIO

➤ Các hàm trong Mbed

Cú pháp

BusIn (PinName p0, PinName p1=NC, PinName p2=NC, PinName p3=NC, PinName p4=NC, PinName p5=NC, PinName p6=NC, PinName p7=NC, PinName p8=NC, PinName p9=NC, PinName p10=NC, PinName p11=NC, PinName p12=NC, PinName p13=NC, PinName p14=NC, PinName p15=NC)

Ví dụ

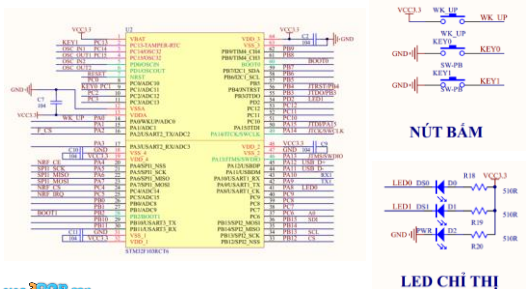
```
BusIn Button(PA_2, PA_7, PB_1, PB_11, PC_4, PA_5, PC_5);
Button.mode(PullNone); // Optional: set mode as PullUp/PullDown/PullNone/OpenDrain
while(1) {
    switch(Button & Button.mask()) { // read the bus and mask out bits not being used
        case 0x0: printf("0b0000, D3,D2,D1,D0 are low \n");break;
        case 0x1: printf("0b0001, D0 is high \n");break;
        case 0x2: printf("0b0010, D1 is high \n");break;
        // ...
    }
    wait(1);
}
```

oOo3POB.com

23

5. Một số ví dụ viết trên Visual Studio Code

➤ Trích mạch nguyên lý KIT STM32F103



oOo3POB.com

5. Một số ví dụ viết trên Visual Studio Code

➤ Ví dụ 1: Nhấp nháy LED đơn

Mbed

```

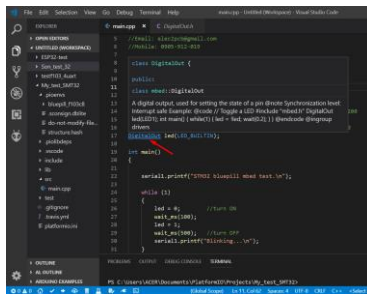
1 //Khai báo tập trình STM32
2 //Bk1: Nhấp nháy LED đơn
3 //MCU: STM32F103R8T6
4 //ipson @ ElecPCB.com
5 //Email: elecpcb@gmail.com
6 //Mobile: 0905-912-019
7
8 #include <mbed.h>
9
10 #define LED_1_PD_2 //định nghĩa LED_1 tại chân PD_2
11
12 //Khởi báo gán chân LED_1/PD_2 như là ngõ ra và đặt tên là led
13 DigitalOut led(LED_1);
14
15 int main()
16 {
17     while (1)
18     {
19         led = 0; //turn ON
20         wait_ms(500);
21         led = 1;
22         wait_ms(500); //turn OFF
23     }
24 }

```

25

5. Một số ví dụ viết trên Visual Studio Code

Mẹo: Trong VSC Bấm **Ctrl+chuột trái** để đi đến hàm C



```

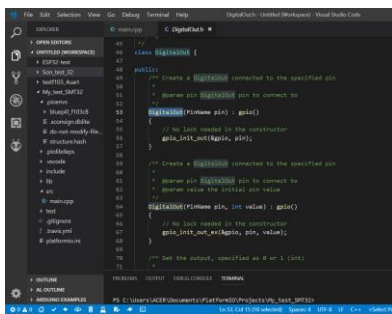
1 //Khai báo tập trình STM32
2 //Bk1: Nhấp nháy LED đơn
3 //MCU: STM32F103R8T6
4 //ipson @ ElecPCB.com
5 //Email: elecpcb@gmail.com
6 //Mobile: 0905-912-019
7
8 #include <mbed.h>
9
10 #define LED_1_PD_2 //định nghĩa LED_1 tại chân PD_2
11
12 //Khởi báo gán chân LED_1/PD_2 như là ngõ ra và đặt tên là led
13 DigitalOut led(LED_1);
14
15 int main()
16 {
17     while (1)
18     {
19         led = 0; //turn ON
20         wait_ms(500);
21         led = 1;
22         wait_ms(500); //turn OFF
23     }
24 }

```

26

5. Một số ví dụ viết trên Visual Studio Code

Mẹo: Trong VSC Bấm **Ctrl+chuột trái** để đi đến hàm C



```

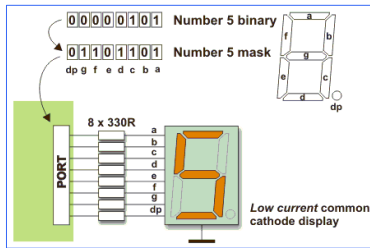
1 //Khai báo tập trình STM32
2 //Bk1: Nhấp nháy LED đơn
3 //MCU: STM32F103R8T6
4 //ipson @ ElecPCB.com
5 //Email: elecpcb@gmail.com
6 //Mobile: 0905-912-019
7
8 #include <mbed.h>
9
10 #define LED_1_PD_2 //định nghĩa LED_1 tại chân PD_2
11
12 //Khởi báo gán chân LED_1/PD_2 như là ngõ ra và đặt tên là led
13 DigitalOut led(LED_1);
14
15 int main()
16 {
17     while (1)
18     {
19         led = 0; //turn ON
20         wait_ms(500);
21         led = 1;
22         wait_ms(500); //turn OFF
23     }
24 }

```

27

5. Một số ví dụ viết trên Visual Studio Code

➤ Ví dụ 2: Đếm lên trên 1 LED 7 đoạn

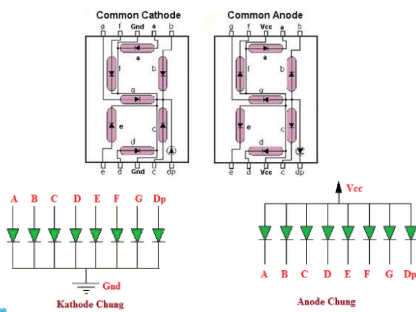


oioo3POB.com

28

5. Một số ví dụ viết trên Visual Studio Code

• Có 2 loại LED 7 đoạn

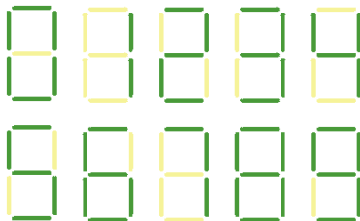


oioo3POB.com

29

5. Một số ví dụ viết trên Visual Studio Code

• Hình dạng những con số



oioo3POB.com

30

5. Một số ví dụ viết trên Visual Studio Code

- Tạo bảng mã cho LED 7 đoạn

Loại Anode chung

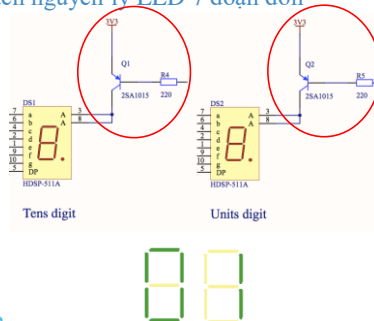
	dot	g	f	e	d	c	b	a	Nhị phân	Hexa
0										
1	1	1	1	1	1	0	0	1		0xF9
2										
3										
4										
5										
6										
7										
8										
9										

oioo3POB.com

31

5. Một số ví dụ viết trên Visual Studio Code

- Mạch nguyên lý LED 7 đoạn đơn

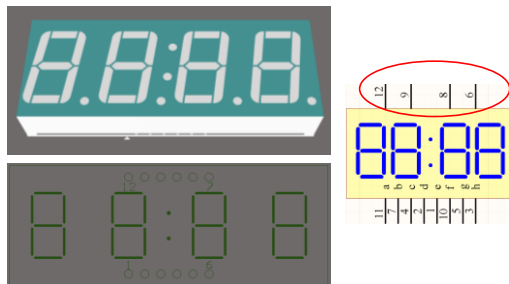


oioo3POB.com

32

5. Một số ví dụ viết trên Visual Studio Code

- Mạch nguyên lý Module 4 LED đoạn



oioo3POB.com

33

5. Một số ví dụ viết trên Visual Studio Code

► Đáp án ví dụ 3: Đếm lên trên 2 LED 7 đoạn

```

24 int main() {
25     // Ví dụ thay đổi các LED 2,3,4 sáng
26     EN_LED = 0x0001;
27     // EN_LED = 0x0010;
28     // EN_LED = 0x0100;
29     // EN_LED = 0x1000;
30     // port init
31     LED7seg = 0;
32     // Tạo LED bit pattern - Dãy dãy
33     const int n0 = 0x3F; // 0011 1111
34     const int n1 = 0x06; // 0000 0110
35     const int n2 = 0x5B; // 0101 1011
36     const int n3 = 0x4F; // 0100 1111
37     const int n4 = 0x66; // 0110 0110
38     const int n5 = 0x6D; // 0110 1101
39     const int n6 = 0x7D; // 0111 1101
40     const int n7 = 0x6F; // 0000 1111
41     const int n8 = 0x77; // 0111 1111
42     const int n9 = 0x69; // 0110 1111
43     // convert: count -> 7seg number
44     const int seg[] = {n0, n1, n2, n3, n4, n5, n6, n7, n8, n9};
45     int nCntX = 0; // 7seg LED counter
46     // port init
47     LED7seg = 0;
48     while(1) {
49         for(int i=0; i<10; i++){
50             EN_LED = 0x0001;
51             LED7seg = ~seg[nCntX/10]; // Hàng chục
52             for(int j=0; j<10; j++){
53                 EN_LED = 0x0001;
54                 LED7seg = ~seg[nCntX%10]; // Hàng đơn vị
55             }
56             nCntX++;
57             // 7seg LED limit count 99
58             if (100 == nCntX) {
59                 nCntX = 0;
60             }
61         }
62     }
63 }

```

37

5. Một số ví dụ viết trên Visual Studio Code

► Ví dụ 3: Đếm lên trên 4 LED 7 đoạn

38

5. Một số ví dụ viết trên Visual Studio Code

► Ví dụ 4: Kiểm tra phím bấm

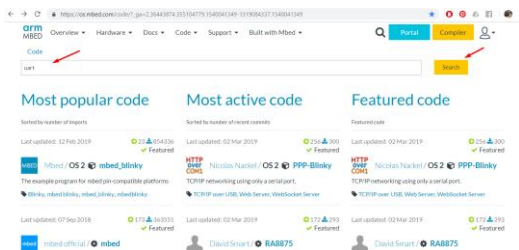
```

1 //Khai học lập trình STM32
2 //BÀI: Kiểm tra phím bấm
3 //MCU: STM32F103R8T6
4 //Ipsos @ Elecpoe.com
5 //Email: elecpoe@gmail.com
6 //Mobile: 0905-912-919
7
8 #include "mbed.h"
9
10 DigitalIn KEY1(PC_13); // Khai báo chân PC_13 là ngõ vào số, gán tên là KEY1
11 DigitalOut LED01(PD_2); // Khai báo chân PD_2 là ngõ ra số, gán tên là LED01
12
13 int main()
14 {
15     // Bỏ chọn: cấu mode là PullUp/PullDown/PullNone/OpenDrain
16     KEY1.mode(PullUp);
17
18     // Bấm phím và xem trạng thái LED thay đổi
19     while(1) {
20         LED01 = KEY1.read(); // Trạng thái LED thay đổi theo mức logic tại chân KEY1
21         wait(0.25); // Delay 0.25s
22     }
23 }

```

39

Trang công đồng Mbed



https://os.mbed.com/code/?_ga=2.36443874.355104779.1540041349-1319084337.1540041349

43

Trang công đồng ST



Welcome to the ST Community!



<https://community.st.com/s/>

44

Page Group



➤ Được giải đáp thắc mắc trong và sau khi học tại diễn đàn này

<https://www.facebook.com/groups/1952065898399986/>

45



<https://www.youtube.com/channel/UC7b10JGt5MIQKRLBMHR3eA/featured>

46



<https://www.facebook.com/elec2pcb/>

47



Website: elec2pcb.com - Email: elec2pcb@gmail.com - Mobile: 0905 912 019

48



thank you!