


Giảng viên ra đề: (Chữ ký và Họ tên)	(Ngày ra đề)	Người phê duyệt: (Chữ ký, Chức vụ và Họ tên)	(Ngày duyệt đề)
--	--------------	--	-----------------

 TRƯỜNG ĐH BÁCH KHOA – ĐHQG-HCM KHOA ĐIỆN ĐIỆN TỬ	THI CUỐI KỲ		Học kỳ/năm học		2	2022-2023
			Ngày thi		15/06/2023	
	Môn học	Vi điều khiển và Hệ thống nhúng				
	Mã môn học	45173				
	Thời lượng	65 phút	Mã đề			
Ghi chú: - Được sử dụng tài liệu						

Câu 1: (3,5đ)

Thiết kế mạch giải mã địa chỉ và kết nối dữ liệu cho hệ thống (8-bit địa chỉ A7-A0; 16-bit dữ liệu D15-D0; RD, WR trên 2 chân riêng biệt) kết nối với các module bên dưới. Ghi rõ kết nối chân CS, WR, RD tới các module và địa chỉ bắt đầu, địa chỉ kết thúc của các module (không giải mã địa chỉ các kênh):

- | | |
|--|--|
| 1/ Module 1: 16 kênh ADC 16-bit | 5/ Module 5: 8 kênh I2C 16-bit |
| 2/ Module 2: 16 kênh Encoder 32-bit | 6/ Module 6: 8 kênh SPI 16-bit |
| 3/ Module 3: 8 kênh PWM 10-bit | 7/ Module 7: 8 kênh Digital Input 8-bit |
| 4/ Module 4: 4 kênh Input Capture 16-bit | 8/ Module 8: 4 kênh Digital Output 8-bit |

Câu 2: (3đ)

Trong yêu cầu thiết kế hệ thống dùng STM32F4 ta cần sử dụng Port {PD0, PD1} cho CAN, {PF0, PF1} cho I2C, {PC10, PC11} cho UART và 2 kênh DAC. Hãy cấu hình các thanh ghi hệ thống để sử dụng 3 chức năng này và I2C, DAC có khả năng hoạt động DMA với yêu cầu sau:

- Chỉ rõ chân nào sử dụng 2 kênh DAC
- Cấu hình thanh ghi cho phép xung clock để hoạt động các chức năng trên
- Cấu hình thanh ghi chân IO để lựa chọn chức năng phù hợp

Câu 3: (3,5đ)

Giả sử dùng STM32F4 giải mã địa chỉ 16-bit dữ liệu để đọc 2 kênh encoder 32-bit ở Câu 1, biết địa chỉ bắt đầu của CS_ENC là 0x6000_0020. Hàm xuất dữ liệu ra UART được cho trước như sau:

- Hàm gửi N byte từ bộ đệm *txbuff* ra UART: void ***send_data*** (char *txbuff, int N). Trong đó, dữ liệu *txbuff* dạng binary với tổng số byte truyền là 11 và format chuỗi dữ liệu truyền:

0x55 (1byte)	0xAA (1byte)	ENC1 (4bytes)	ENC2 (4bytes)	CRC (1byte)
-----------------	-----------------	------------------	------------------	----------------

Viết chương trình đọc 2 kênh encoder 32-bit, tính CRC và truyền dữ liệu ra UART theo frame ở trên, biết rằng CRC bằng xor 10 byte dữ liệu trong frame từ byte thứ 1 (0x55) đến byte thứ 10 (ENC2).

(Hết)

ĐÁP ÁN

Câu 1: (3,5đ)

- Vẽ đúng sơ đồ giải mã 74138, chú ý 16-bit data D[15:0] (1đ)
- Ghi đúng tầm địa chỉ cho 6 module (1đ)
- Vẽ đúng sơ đồ kết nối 6 module (1.5đ)

Câu 2: (3đ)

- Chỉ rõ chân kết nối: (0.5đ)
 - PD0 – PD1: CAN1 – AF9
 - PF0 – PF1: I2C2 – AF4, DMA1
 - PC10, PC11: UART3 – AF7
 - PA4 – PA5: DAC, DMA1
- Cấu hình thanh ghi xung clock (1đ)
 - RCC_AHB1ENR |= 1 | (1<<2) | (1<<3) | (1<<5) | (1<<21); //clock cho PA,PC,PD,PF, DMA1
 - RCC_APB1ENR |= (1<<18) | (1<<22) | (1<<25) | (1<<29); //clock cho UART3, I2C2, CAN1, //DAC
- Cấu hình mode chức năng (1.5đ)
 - GPIOA_MODER |= (3<<8) | (3<<10);
 - GPIOC_MODER |= (2<<20) | (2<<22);
 - GPIOD_MODER |= (2<<0) | (2<<2);
 - GPIOF_MODER |= (2<<0) | (2<<2);
 - GPIOC_AFRH |= (7<<8) | (7<<12);
 - GPIOD_AFRH |= (9<<0) | (9<<4);
 - GPIOF_AFRH |= (4<<0) | (4<<4);
 - DMA_S2CR |= (7<<25); //DMA1, Stream2, Channel7, I2C2_RX
 - DMA_S7CR |= (7<<25); //DMA1, Stream7, Channel7, I2C2_TX
 - DMA_S5CR |= (7<<25); //DMA1, Stream5, Channel7, DAC1
 - DMA_S6CR |= (7<<25); //DMA1, Stream6, Channel7, DAC2

Câu 3: (3,5đ)

```
char txbuff[11];
uint16_t *p, temp, i, crc;
main() {
    while(1) {
        p = 0x60000020;
        txbuff[0] = 0x55;        txbuff[1] = 0xAA;
        for(i=0; i<4; i++){
            temp = *p++;
            txbuff[2*i+2] = temp>>8;
            txbuff[2*i+3] = temp & 0xff;
        }
        crc = 0;
        for(i=0; i<10; i++)    crc ^= txbuff[i];
        txbuff[10] = crc;
        send_data(txbuff, 11);
    }
}
```