

Bài 8 : Khảo sát khối chuyển đổi A-D

Nội dung:

Nđ1. Khảo sát hoạt động khối chuyển đổi A-D.

Nđ2. Khảo sát các thanh ghi điều khiển hoạt động khối chuyển đổi A-D.

Yêu cầu:

Yc1. Viết chương trình (không dùng ngắt quãng) đọc và hiển thị liên tục giá trị điện áp thay đổi bởi biến trở ra LCD.

Yc2. Làm lại yêu cầu 1 nhưng dùng ngắt quãng AD để đọc kết quả.

Yc3. Làm lại yêu cầu 2 nhưng dùng thêm ngắt quãng Timer để kiểm soát thời gian AD một giây một lần.

8.1 Các bước hiện thực yêu cầu 1.

Bước 1. Tạo dự án mới Tn08, tập tin nguồn Tn08_ADC.asm.

Bước 2. Khởi động LCD, nút nhấn RA5.

Bước 3. Tạo module Adc.c và viết hàm **Adc_init()** thực hiện cấu hình các thanh ghi:

- **ADCON0**: Chọn kênh AD (ANi), cho phép AD (ADON), bắt đầu AD (cho GO/DONE=1) và kết thúc AD (kiểm tra GO/DONE=0).

REGISTER 21-1: ADCON0: A/D CONTROL REGISTER

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3 ⁽¹⁾	CHS2 ⁽¹⁾	CHS1 ⁽¹⁾	CHS0 ⁽¹⁾	GO/DONE	ADON
bit 7							bit 0

- **ADCON1**: Cho phép ngõ nhập ANi là Analog.

REGISTER 21-2: ADCON1: A/D CONTROL REGISTER 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

- **ADCON2**: ADFM chọn chỉnh biên kết quả 16 bit (trái=10 bit cao / phải=10 bit thấp); ADCSi chọn nguồn xung lấy mẫu.

REGISTER 21-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

- Cấu hình ngõ nhập cho kênh được chọn AN0.

```
#define XUAT      0
#define NHAP      1
#define TINHIEU   PORTAbits.RA0
#define TINHIEU_IO TRISAbits.RA0
#define AD_CFG0 0b00000001 //kênh AN0, GO/nDONE=0, ADON=1
#define AD_CFG1 0x0E      // su dung kênh AN0
#define AD_CFG2 0b10000001; //right justified, ACQT=0TAD, FOSC/8
void Adc_init()
{
    TINHIEU_IO=NHAP;    // ngo AN0 nhap
    ADCON0=AD_CFG0;
    ADCON1=AD_CFG0;
    ADCON2=AD_CFG2;
}
```

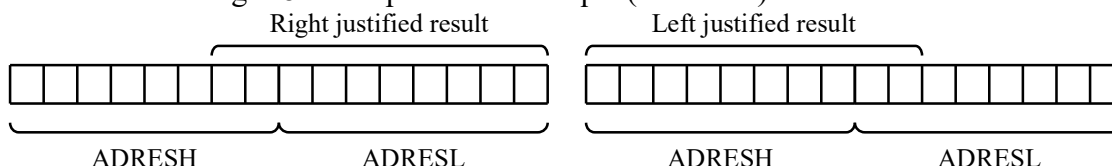
Bước 4. Viết hàm **Adc_update()** thực hiện AD một lần gồm các bước sau:

- Để bắt đầu quá trình chuyển đổi AD, ta phải ra lệnh bằng cách cho bit GO/DONE=1.

- Sau đó, chờ cho đến khi bộ ADC chuyển đổi xong (kiểm tra bit GO/DONE=0).
- Đọc kết quả trong 2 thanh ghi ADRESH-ADRESL cất vào biến **unsigned adres**.
- Gọi hàm **Adc_process()** để xuất kết quả ra LCD.

```
void Adc_update()
{
    ADCON0bits.GO=1;
    while(ADCON0bits.DONE);
    adres=ADRESH*256+ADRESL;
    Adc_process();
}
```

Bước 5. Do ta chọn chỉnh biên phải kết quả nên ADRESH giữ 2 bit cao của kết quả và ADRESL giữ 8 bit thấp nhất của kết quả (xem hình).



Bước 6. Trong **Adc_process()**, thực hiện xuất kết quả AD ra LCD tương tự như hình sau:



8.2 Các bước hiện thực yêu cầu 2.

Bước 7. Thêm phần khởi động ngắt quãng AD vào cuối hàm **Adc_init()** như sau:

```
RCONbits.IPEN=1; //Khởi động ưu tiên ngắt
IPR1bits.ADIP=0; //Ngắt AD ưu tiên thấp
PIR1bits.ADIF=0; //Xóa cờ ngắt
PIE1bits.ADIE=1; //Cho phép ngắt
INTCONbits.GIEH=1;
INTCONbits.GIEL=1;
ADCON0bits.GO=1; // Bắt đầu AD luôn
```

Bước 8. Viết thêm hàm **Adc_isr()** thực hiện:

- Xác nhận có ngắt AD.
- Xóa cờ ngắt.
- Đọc kết quả AD vào biến **adres**.
- Gọi hàm **Adc_process()** để xử lý xuất kết quả ra LCD.
- Cuối cùng, bật bit GO lên 1 cho phép AD tiếp để lấy kết quả liên tục.

```
void interrupt low_priority Adc_isr()
{
    if ((PIE1bits.ADIE) && (PIR1bits.ADIF))
    {
        PIR1bits.ADIF=0;
        adres=ADRESH*256+ADRESL;
        Adc_process();
        ADCON0bits.GO=1;
    }
}
```

Bước 9. Chạy thử chương trình.

8.3 Các bước hiện thực yêu cầu 3.

Bước 10. Thêm module Timer vào, cấu hình ngắt Timer ưu tiên cao, trong hàm `timer_process()` định thì đủ 1s thì bật bit GO lên 1. Như vậy, ta sẽ **không** bật bit GO lên 1 trong hàm `Adc_init()` và `Adc_isr()` nữa.

Bước 11. Chạy thử và quan sát sự thay đổi giá trị kết quả trên LCD.

8.4 Bài tập

- Thực hiện AD trên kênh AN5 (port RF0). Nối dây trên connector mở rộng giữa chân RA0 với chân RF0 để có thể sử dụng biến trở RA0 cho kênh RF0.
- Viết chương trình dùng biến trở RA0 để điều khiển LED PORTD sáng từ 1 LED đến 8 LED tùy theo vị trí biến trở.
- Sử dụng module ADC của Pic để đo nhiệt độ trong phòng với cảm biến MCP9701A có sẵn (xem mạch), dùng LCD để hiển thị giá trị nhiệt độ (tham khảo MCP9701A data sheet để tính nhiệt độ từ điện áp đọc được).

