

제14강

C51 프로그래밍 III

함수 및 인터럽트 핸들러 함수

외부 인터럽트

타이머/카운터 인터럽트

시리얼인터럽트

실습과제

Ref.) Chapter 12, 13, [C51_03.zip](#) 다운로드

함수

* 함수 유형

: 일반 함수 / 인터럽트 핸들러 함수

* 일반 함수 정의

: 표준 C에서의 함수 정의와 유사

```
[return_type] <func_name>([args]) [{small|compact|large}]  
    [reentrant] [using y] {  
        :  
    }  
  
    [{small|compact|large}] // 메모리모델(디폴트 small)  
    [reentrant]             // 재진입함수(재귀함수)  
    [using y]               // 레지스터뱅크(y:뱅크번호) 선택
```

인터럽트 핸들러 함수

* 인터럽트 핸들러 함수 정의

: 인터럽트 요청에 대한 서비스를 수행하는 함수

```
void <func_name>(void) interrupt <벡터번호>
    [using <뱅크번호>] { ..... }
```

벡터번호	기 능	주 소
0	EXTERNAL INT 0	0003H
1	TIMER/COUNTER 0	000BH
2	EXTERNAL INT 1	0013H
3	TIMER/COUNTER 1	001BH
4	SERIAL	0023H

뱅크번호	레지스터뱅크
0	RB0
1	RB1
2	RB2
3	RB3

예) 타이머/카운터 0의 인터럽트 핸들러 함수 정의

```
// Timer/Counter 0 ISR
void TCO_ISR(void) interrupt 1 using 2 { // T/C0, RB2
    : // 함수명은 스타트업코드 파일에서 정의한 이름으로...
    :
}
```

인터럽트 핸들러 함수(계속)

* IHF(ISR)의 어셈블리 변환

C 소스	변환된 Assembly 소스
	ASSEMBLY LISTING OF GENERATED OBJECT CODE ; FUNCTION falarm (BEGIN)
	0000 C0E0 PUSH ACC
	0002 C0D0 PUSH PSW
	; SOURCE LINE # 5
	MOV PSW,#00011000B ;???
	; SOURCE LINE # 6
1 extern bit alarm;	0004 E5?? MOV A,alarm_count+01H
2 int alarm_count;	0006 25E0 ADD A,ACC
3	0008 F5?? MOV alarm_count+01H,A
4 // ISR	000A E5?? MOV A,alarm_count
5 void falarm (void)	000C 33 RLC A
interrupt 1 using 3 {	000D F5?? MOV alarm_count,A
6 1 alarm_count *= 2;	; SOURCE LINE # 7
7 1 alarm = 1;	000F D2?? SETB alarm
8 1 }	; SOURCE LINE # 8
	0011 D0D0 POP PSW
	0013 D0E0 POP ACC
	0015 32 RETI
	; FUNCTION falarm (END)

주) 밑줄(??)부분은 내부데이터메모리의 직접주소로 미상

스타트업파일 설정

* 인터럽트 취급시, 스타트업코드 파일 취급

```

;=====
; File : startup.a51
;=====
IDATALEN      EQU 80H
XDATASTART    EQU 0H
XDATALEN      EQU 0H
PDATASTART    EQU 0H
PDATALEN      EQU 0H

        NAME ?C_STARTUP
?C_C51STARTUP SEGMENT    CODE
?STACK      SEGMENT IDATA
        RSEG ?STACK
STACK:      DS 1

;      EXTRN      CODE (?C_START)
;      EXTRN      CODE (INT0_ISR)
;      EXTRN      CODE (TC0_ISR)
;      EXTRN      CODE (INT1_ISR)
;      EXTRN      CODE (TC1_ISR)
;      EXTRN      CODE (SERIAL_ISR)
    
```

```

PUBLIC ?C_STARTUP

        CSEG AT 8000H           ; start address of user program, KUT51보드
?C_STARTUP: LJMP STARTUP1

;      CSEG      AT 8003H
;      LJMP INTO_ISR

;      CSEG      AT 800BH
;      LJMP TCO_ISR

;      CSEG      AT 8013H
;      LJMP INT1_ISR

;      CSEG      AT 801BH
;      LJMP TC1_ISR

;      CSEG      AT 8023H
;      LJMP SERIAL_ISR

RSEG ?C_C51STARTUP
STARTUP1:

IF IDATALEN <> 0                ; idata clear
    MOV R0,#IDATALEN-1
    :

```

외부 인터럽트(C51_03/ExINT 폴더)

* 스타트업코드 파일(STARTUP.A51)

- 1) **인터럽트방식의 경우**, 외부인터럽트 관련부분 활성화
: 함수명 일치시킬 것!!

```
      :  
      EXTRN      CODE      (INT0_ISR)  
      :  
  
      CSEG      AT      8000H      ; start address of user program  
  
?C_STARTUP:LJMP      STARTUP1  
  
      CSEG      AT      8003H  
      LJMP      INTO_ISR  
      :
```

- 2) **폴링방식의 경우**, 주석화

예제 실습

[실습1] 외부인터럽트 (인터럽트방식)

```
//=====
// INT0_01.C
//      interrupt driven method
//      normal : pattern shift, INT0 : blink
//      P1_L : LED module
//=====
#include <reg51.h>
#include "delay.h"

// external INT0 ISR
void INT0_ISR(void) interrupt 0 {
    unsigned char blink = 0x00;

    P1 = blink;
    delay(50000);
    P1 = ~blink;
    delay(50000);
}

void main(void) {
    unsigned char pat;
```



```
ITO = 1;           // edge trigger
EXO = 1;           // INT0 enable
EA = 1;            // global enable

pat = 0xFE;
while(1) {
    P1 = pat ;
    delay(50000);
    pat = (pat << 1) | 1;
    if(!(pat & 0x10))
        pat = 0xFE;
}
}
```

예제 실습(계속)

[실습2] 외부인터럽트 (폴링방식)

```
//=====
// INT0_02.C
//      polling method
//      normal : pattern shift, INT0 : blink
//      P1_L : LED module
//=====
#include <reg51.h>
#include "delay.h"

void main(void) {
    unsigned char blink = 0x00;
    unsigned char pat;

    IT0 = 1;                // edge trigger
    IE0 = 0;                // flag clear

    pat = 0xFE;
    while(1) {
        P1 = pat ;

        delay(50000);
        pat = (pat << 1) | 1;
    }
}
```

```
    if(!(pat & 0x10))
        pat = 0xFE;

    if(IE0) {                                // interrupt service code
        P1 = blink;
        delay(50000);
        P1 = ~blink;
        delay(50000);

        IE0 = 0;                            // flag clear
    }
}
```

실습과제

[과제1] LED 상태 횡수만큼 점멸

: P1하위니블에 LED 모듈연결

: 노말... 최하위LED부터 차례로 ON

인터럽트시...LED값만큼 전체를 점멸

[과제2] FND 모듈을 이용한 자모음패턴 출력

: 노말... FND 세그먼트 하나씩 ON

인터럽트시...본인의 이름 자모음패턴을 출력

타이머/카운터 인터럽트(C51_03/TC 폴더)

* STARTUP.A51 소스

: 허용인터럽트 관련항목 활성화

```
;=====
;  STARTUP.A51
;=====
```

```
      :
EXTRN : CODE (TC0_ISR)
      :
```

```
PUBLIC ?C_STARTUP
```

```
CSEG AT 8000H ; *
```

```
?C_STARTUP:LJMP STARTUP1
```

```
CSEG AT 800BH
LJMP TC0_ISR
      :
```

예제 실습

[실습1] 타이머 모드 (인터럽트방식)

```
//=====
// TCNT_01.C
//      T/C0, Timer, mode 1, interrupt driven
//      P1_L : LED module
//=====
#include <reg51.h>

unsigned char pat;

void TCO_ISR(void) interrupt 1 {
    TR0 = 0;                // T/C0 stop

    pat = (pat << 1) | 0x01; // move left 1 bit
    if((pat & 0x10) == 0)
        pat = 0xFE;
    P1 = pat;                // output the pat to P1

    TH0 = 0;                 // timer constant
    TL0 = 0;

    TR0 = 1;                // T/C0 run
}
```

```
void main(void) {  
    TMOD = 0x01;           // T/C0, timer, mode 1  
  
    TH0 = 0;               // timer constant  
    TLO = 0;  
  
    PTO = 0;               // T/C0 priority  
    ETO = 1;               // enable T/C0 interrupt  
    EA = 1;                // enable all interrupt  
  
    pat = 0xFE;             // initial pat  
    P1 = pat;  
  
    TR0 = 1;               // T/C0 run  
  
    while(1) ;             // loop, Normal state  
}
```

예제 실습

[실습2] 카운터 모드 (폴링방식)

```
//=====
// TCNT_02.C
//      T/C0, Counter, mode 2, polling
//      P1_L : LED module
//=====
#include <reg51.h>

void main(void) {
    unsigned char pat;

    TMOD = 0x06;           // T/C0, counter mode, mode 2
    TH0 = 255;             // timer constant
    TL0 = 255;

    pat = 0xFE;            // initial pattern
    while(1) {
        P1 = pat;          // output

        TR0 = 1;           // T/C0 run
        TF0 = 0;           // overflow clear
        while(!TF0) ;      // wait until overflow, polling
    }
}
```



```
// following is interrupt service code
TR0 = 0; // overflow clear
pat = (pat << 1) | 0x01; // shift left and OR 01
if((pat & 0x10) == 0)
    pat = 0xFE; // initial pattern
}
```

실습과제

[과제1] 게이트 모드

: 게이트 모드로의 제어

[과제2] 타이머 모드 응용

: FND 모듈 활용

: 자모음 패턴 출력

[과제3] 카운터 모드 응용(교통량측정)

: 10대의 차량이동시 인터럽트 요청

시리얼 인터럽트(C51_03/SERIAL 폴더)

* 시리얼인터럽트 관련 모듈

```
//=====
//  serial.h
//=====
#ifndef __SERIAL_H__
#define __SERIAL_H__

#define      CR      0x0D          // carriage return, '\r'
#define      LF      0x0A          // line feed, '\n'

// for polling method
void initSerial(void);
void txRTN(unsigned char txbuf);
unsigned char rxRTN(void);
void txStr(unsigned char *str);

// for interrupt driven method
void initISerial(void);
void txIRTN(unsigned char txbuf);
unsigned char rxIRTN(void);

#endif
```

```
//=====
//  serial.c
//=====
#include <reg51.h>
#include "serial.h"

// serial initialize, 19200bps
void initSerial(void) {
    TMOD = 0x20;           // T/C1, mode 2
    TH1 = 0xFD;            // 9600bps
    PCON = 0x80;           // x 2 = 19200bps
    TR1 = 1;              // T/C1 run

    SCON = 0x52;           // Mode=1, REN=1, TI=1, RI=0

    // for interrupt driven method
    //PS = 1;              // enhance serial priority
    //ES = 1;              // enable serial interrupt
    //EA = 1;              // enable all interrupt
}

// transmit a character
void txRTN(unsigned char txbuf) {
    while(!TI) ;           // wait until TI == 1
    TI = 0;                // clear TI
    SBUF = txbuf;          // write transmitting data
}
```

```
// receive a character
unsigned char rxRTN(void) {
    unsigned char rxbuf;

    while(!RI) ;           // wait until RI == 1
    RI = 0;                 // clear RI
    rxbuf = SBUF;           // read received data

    return rxbuf;
}

// transmit a string
void txStr(unsigned char *str) {
    while(*str)
        txRTN(*str++);
}

// Interrupt... serial initialize, 19200bps
void initSerial(void) {
    TMOD = 0x20;           // T/C1, mode 2
    TH1 = 0xFD;            // 9600bps
    PCON = 0x80;           // x 2 = 19200bps
    TR1 = 1;               // T/C1 run

    SCON = 0x52;           // Mode=1, REN=1, TI=1, RI=0

    // for interrupt driven method
```

```
    PS = 1;                // enhance serial priority
    ES = 1;                // enable serial interrupt
    EA = 1;                // enable all interrupt
}

// transmit a character
void txlRTN(unsigned char txbuf) {
    TI = 0;                // clear TI
    SBUF = txbuf;          // write transmitting data
}

// receive a character
unsigned char rxlRTN(void) {
    unsigned char rxbuf;

    RI = 0;                // clear RI
    rxbuf = SBUF;           // read received data

    return rxbuf;
}
```

예제 실습

* SERIALAPP.Uv2 더블클릭

[실습1] 일정 시간지연후 문자전송

```
//=====
// SERIAL_01.C
//      transmit ALPHABET to PC
//      using delay
//=====
#include <reg51.h>
#include "serial.h"
#include "delay.h"

void main(void) {
    unsigned char i;

    initSerial();

    SBUF = CR;                // line home
    delay(100);
    SBUF = LF;                // next line
    delay(100);
    SBUF = CR;                // line home
    delay(100);
```

```
do {  
    for(i=0; i<26; i++) { // 'A'~'Z'  
        SBUF = 'A' + i;  
        delay(10000);    // delay  
    }  
  
    SBUF = LF;           // next line  
    delay(100);  
    SBUF = CR;           // line home  
    delay(100);  
} while(1);  
}
```


예제실습(계속)

[실습2] 폴링에 의한 통신

```
//=====
// SERIAL_02.C
//      transmit ALPHABET to PC
//      using polling method
//=====
#include <reg51.h>
#include "serial.h"

void main(void) {
    unsigned char i;

    initSerial();

    txRTN(CR);           // line home
    txRTN(LF);           // next line
    txRTN(CR);           // line home

    while(1) {
        for(i=0; i<26; i++) // 'a'~'z'
            txRTN('a' + i);
        txRTN(LF);         // next line
        txRTN(CR);         // line home
    }
}
```

```
}  
}
```

예제실습(계속)

[실습3] 인터럽트에 의한 통신

```
//=====
// SERIAL_03.C
//      transmit ALPHABET to PC
//      using interrupt driven method
///=====
#include <reg51.h>
#include "serial.h"
#include "delay.h"

unsigned char txbuf;
unsigned char flagTx;

void SERIAL_ISR(void) interrupt 4 {
    if(TI) {
        if(flagTx==1) {
            TI = 0;
            SBUF = txbuf;
            flagTx = 0;
        }
    }
}
```

```
void main(void) {
    unsigned char i;

    flagTx = 0;

    initISerial();

    txbuf = CR;                // line home
    flagTx = 1;
    txbuf = LF;                // next line

    flagTx = 1;
    while(1) {
        for(i=0; i<26; i++) {  // 'A'~'Z'
            txbuf = 'A' + i;
            flagTx = 1;
            msec_delay(10);    // delay
        }
        txbuf = LF;           // next line
        flagTx = 1;
    }
}
```

예제 실습(계속)

[실습4] 문자반송(인터럽트방식)

```
//=====
// SERIAL_04.C
//      receive a char. from PC and retransmit it to PC
//      using interrupt driven method
//      P1_L : LED module
//=====
#include <reg51.h>
#include "serial.h"
#include "delay.h"

unsigned char buf;
unsigned char flagRx;

void SERIAL_ISR(void) interrupt 4 {
    if(RI) {
        buf = rxIRTN();
        flagRx = 1;
    }
    if(TI) {
        if(flagRx == 1) {
            txIRTN(buf);
            flagRx = 0;
        }
    }
}
```

```
    }  
  }  
}  
  
void main(void) {  
    unsigned char pat;  
    flagRx = 0;  
  
    initIISerial();  
  
    pat = 0x03;  
    while(1) {                                // normal  
        P1 = pat;  
        delay(1000);  
        pat = ~pat;  
    }  
}
```

예제실습(계속)

[실습5] 문자반송 (폴링방식)

```
//=====
// SERIAL_05.C
//      receive a char. from PC and retransmit it to PC
//      include carriage return char handling
//      using polling method, P1_L : LED module
//      serial.h, serial.c
//=====
#include <reg51.h>
#include "serial.h"

void main(void) {
    unsigned char rxbuf;

    initSerial();

    txStr("input any char..... \r\n");
    while(1) {
        rxbuf = rxRTN();

        txRTN(rxbuf);           // retransmit to PC
        if(rxbuf==CR) {        // next line
            txRTN(LF);
        }
    }
}
```

```
        txRTN(CR);           // line home
    }
    P1 = ~rxbuf;              // out the char to P1
}
}
```


예제실습(계속)

[실습6] 보드간 시리얼통신

: 두 보드간 연결은 **크로스 시리얼케이블**로..

* 전송측 보드용

```
//=====
//  SERIAL_06_Tx.C
//      for sending board
//=====
#include <reg51.h>
#include <stdio.h>
#include "serial.h"
#include "clcd8.h"
#include "delay.h"

int main(void) {
    char msg[16] ;
    unsigned char tx;

    initLCD_8();
    initSerial();
```

```
tx = 0;
while(1) {
    txRTN(tx);                // tx : ASCII code

    sprintf(msg, "SendingNO.: %3d", (unsigned)tx);
    writeCommand_8(0x02);
    putStr_8(msg);

    tx++;
    msec_delay(500);
}
}
```

* 수신측 보드용

```
//=====
//  SERIAL_06_Rx.C
//      for receiving board
//=====
#include <reg51.h>
#include <stdio.h>
#include "serial.h"
#include "clcd8.h"
#include "delay.h"

int main(void) {
    char msg[16];
    unsigned char rx;

    initLCD_8();
    initSerial();

    writeCommand_8(0x02);
    putStr_8("Waiting.....");

    while(1) {
        rx = rxRTN();
        txRTN(rx);                                // retransmit

        sprintf(msg, "RxData : %3d %c", (unsigned)rx, rx);
        writeCommand_8(0x02);
    }
}
```

```

putStr_8(msg);

writeCommand_8(0x40);           // line 2
putStr_8("Type : ");

if(rx >= 'A' && rx <= 'Z')       // 대문자
    putStr_8("ALPHABET");
else if(rx >= 'a' && rx <= 'z')  // 소문자
    putStr_8("alphabet");
else if(rx >= '0' && rx <= '9')  // 숫자
    putStr_8("Number ");
else                             // 기타
    putStr_8("other...");
    }
}

```

: 실습방법) 수신측 보드의 프로그램이 실행중인 상태에서 시리얼 케이블을 크로스 시리얼케이블로 교체하고, 송신측 보드의 프로그램을 실행중인 상태에서 수신측보드에 연결된 크로스 시리얼케이블의 다른 한쪽을 송신측 보드에 연결(전원라인은 계속 연결된 상태로)

실습과제

[과제1] 문자열 반송

: 문자열 반송하는 프로그램

: 인터럽트방식 및 폴링방식으로 구현

[과제2] 원격디바이스 제어