



# Embedded System Software

(Programming on an embedded system 3 – 실습)

---

<http://jcnet.co.kr/> 의 자료실의 강의노트를 참조함



# Outline

---

- 개관
- 실습 키트와 개발 환경
- LED
- FND
- 스위치 (인터럽트)
- 버저 (Buzzer)
- 광센서
- 온도 센서

## EX1 – 광센서 (1)

```
1  #include <avr/io.h>
2  #define CDS_VALUE 871
3
4  void init_adc();
5  unsigned short read_adc();
6  void show_adc(unsigned short value);
7
8  int main() {
9      unsigned short value;
10     DDRA = 0xff;
11     init_adc();
12
13     while(1) {
14         value = read_adc();
15         show_adc(value);
16     }
17 }
```

# EX1 – 광센서 (2)

```

19 void init_adc() {
20     ADMUX = 0x00; /* REFS(1:0) = "00": AREF(+5V) 기준전압 사용
21                    * ADLAR = '0': 디폴트 오른쪽 정렬
22                    * MUX(4:0) = "00000": ADC0 사용, 단극 입력
23                    */
24     ADCSRA = 0x87; /* ADEN = '1': ADC Enable
25                    * ADFR = '0': single conversion 모드
26                    * ADPS(2:0) = "111": 프리스케일러 128분주
27                    */
28 }

```

실습 슬라이드 p. 7 – 14

7	6	5	4	3	2	1	0
REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
0	0	0	0	0	0	0	0 = 0x00

7	6	5	4	3	2	1	0
ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0
1	0/1	0	0	0	1	1	1 = 0x87

# EX1 – 광센서 (3)

```
30 unsigned short read_adc(){
31     unsigned char adc_low, adc_high;
32     unsigned short value;
33     ADCSRA |= 0x40; /* ADSC = '1': ADC start conversion */
34     while((ADCSRA & 0x10) != 0x10); /* ADC 변환 완료 검사 */
35     adc_low = ADCL; /* 변환된 low와 high값 읽어오기 */
36     adc_high = ADCH;
37     value = (adc_high << 8) | adc_low; /* 16비트 값으로 연결 */
38
39     return value;
40 }
```

실습 슬라이드 p. 12, 13

## ■ ADCSRA(ADC Control and Status Register A)

- 비트 6 : ADSC(ADC Start Conversion)
  - A/D 컨버터 변환 시작
  - 이 비트에 “1”을 설정하면 ADC 변환이 시작
    - ADEN이 1로 설정되고 난 후 첫 번째 변환에 25개의 ADC 클럭 주기가 필요
    - 다음 변환부터는 13 클럭이 요구
    - AD 변환이 종료되고 난 후 자동적으로 0으로 변환

## ■ ADCSRA(ADC Control and Status Register A)

- 비트 4 : ADIF(ADC Interrupt Flag)
  - A/D 컨버터 인터럽트 플래그
  - A/D 변환의 완료를 알리는 플래그
    - AD변환이 완료되어 ADC Data Register 값이 업데이트 되고 나면 이 비트가 “1”로 세트되면서 AD 변환 완료 인터럽트를 요청
    - 이때 ADIE=1로 설정되고, SREG 레지스터의 I 비트가 1로 설정되어 있으면 이 인터럽트가 발생되어 처리됨

## EX2 – 온도 센서 (1)

```
1  #define F_CPU 16000000UL // CPU 클럭 값 = 16 Mhz
2  #define F_SCK 40000UL // SCK 클럭 값 = 40 Khz
3  #include <avr/io.h>
4  #include <util/delay.h>
5  #define ATS75_ADDR 0x98 // 0b10011000, 7비트를 1비트 left shift
6  #define ATS75_CONFIG_REG 1
7  #define ATS75_TEMP_REG 0
8  void init_twi_port();
9  void write_twi_lbyte_nopreset(char reg, char data);
10 int read_twi_2byte_nopreset(char reg);
11 void display_FND(int value);
12
13 int main() {
14     int temperature;
15     init_twi_port(); // TWI 및 포트 초기화
16     write_twi_lbyte_nopreset(ATS75_CONFIG_REG, 0x00); // 9비트, Normal
17     _delay_ms(100); // 다음 사이클을 위하여 잠시 기다림
18     while (1) { // 온도값 읽어 FND 디스플레이
19         temperature = read_twi_2byte_nopreset(ATS75_TEMP_REG);
20         display_FND(temperature);
21     }
22 }
```

# EX2 – 온도 센서 (2)

```

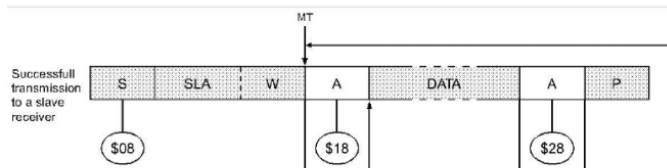
23 void init_twi_port() {
24     DDRC = 0xff;
25     DDRG = 0xff; // FND 출력 세팅
26     PORTD = 3; // For Internal pull-up for SCL & SCK
27     SFIOR &= ~(1 << PUD); // PUD = 0 : Pull Up Disable
28     TWBR = (F_CPU / F_SCK - 16) / 2; // 공식 참조, bit trans rate 설정
29     TWSR = TWSR & 0xfc; // Prescaler 값 = 00 (1백)
30 }
31 void write_twi_lbyte_nopreset(char reg, char data) {
32     TWCR = (1 << TWINT) | (1 << TWSTA) | (1 << TWEN); // START 전송
33     while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x08)
34     TWDR = AT575_ADDR | 0; // SLA+W 준비, W=0
35     TWCR = (1 << TWINT) | (1 << TWEN); // SLA+W 전송
36     while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x18),
37     TWDR = reg; // aTS75 Reg 값 준비
38     TWCR = (1 << TWINT) | (1 << TWEN); // aTS75 Reg 값 전송
39     while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x28),
40     TWDR = data; // DATA 준비
41     TWCR = (1 << TWINT) | (1 << TWEN); // DATA 전송
42     while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x28),
43     TWCR = (1 << TWINT) | (1 << TWSTO) | (1 << TWEN); // STOP 전송
44 }

```



## ■ TWI 동작

### • Master 송신(Write)



1. START 전송 : TWCR → TWINT=1, TWSTA=1, TWEN=1
2. 상태 체크 : TWSR → TWSR & 0xF8 = 0x08
3. TWDR에 SLA+W 세트 : TWDR → TWDR = SLA + W
4. SLA+W 전송 : TWCR → TWINT=1, TWEN=1
5. Ack 체크 : TWSR → TWSR & 0xF8 = 0x18
6. TWDR에 데이터값 세트 : TWDR → TWDR = DATA
7. 데이터 전송 : TWCR → TWINT=1, TWEN=1
7. Ack 체크 : TWSR → TWSR & 0xF8 = 0x28
8. STOP 전송 : TWCR → TWINT=1, TWSTO=1, TWEN=1

실습 슬라이드 p. 39

# EX2 – 온도 센서 (3)

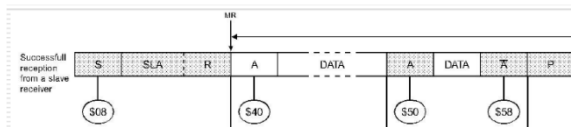
```

45 int read_twi_2byte_nopreset(char reg) {
46     char high_byte, low_byte;
47     TWCR = (1 << TWINT) | (1 << TWSTA) | (1 << TWEN); // START 전송
48     while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x08);
49     TWDR = AT575_ADDR | 0; // SLA+W 준비, W=0
50     TWCR = (1 << TWINT) | (1 << TWEN); // SLA+W 전송
51     while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x18);
52     TWDR = reg; // aTS75 Reg 값 준비
53     TWCR = (1 << TWINT) | (1 << TWEN); // aTS75 Reg 값 전송
54     while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x28);
55     TWCR = (1 << TWINT) | (1 << TWSTA) | (1 << TWEN); // RESTART 전송
56     while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x10);
57     TWDR = AT575_ADDR | 1; // SLA+R 준비, R=1
58     TWCR = (1 << TWINT) | (1 << TWEN); // SLA+R 전송
59     while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x40);
60     TWCR = (1 << TWINT) | (1 << TWEN | 1 << TWEA); // 1st DATA 준비
61     while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x50);
62     high_byte = TWDR; // 1st DATA 수신
63     TWCR = (1 << TWINT) | (1 << TWEN); // 2nd DATA 준비
64     while (((TWCR & (1 << TWINT)) == 0x00) || (TWSR & 0xf8) != 0x58);
65     low_byte = TWDR; // 2nd DATA 수신
66     TWCR = (1 << TWINT) | (1 << TWSTO) | (1 << TWEN); // STOP 전송
67     return((high_byte << 8) | low_byte); // 수신 DATA 리턴
68 }

```

## ■ TWI 동작

### • Master 수신(Read)



1. START 전송 : TWCR → TWINT=1, TWSTA=1, TWEN=1
2. 상태 체크 : TWSR → TWSR & 0xF8 = 0x08
3. TWDR에 SLA+W 세트 : TWDR → TWDR = SLA + R
4. SLA+W 전송 : TWCR → TWINT=1, TWEN=1
5. Ack 체크 : TWSR → TWSR & 0xF8 = 0x40
6. 데이터 수신 : TWCR → TWINT=1, TWEA=1, TWEN=1
7. 상태 체크 : TWSR → TWSR & 0xFC = 0x50 (다음 데이터 6, 7 반복)
8. STOP 전송 : TWCR → TWINT=1, TWSTA=0, TWSTO=1, TWEN=1

실습 슬라이드 p. 39



## EX2 – 온도 센서 (4)

```
70 void display_FND(int value) {
71     unsigned char digit[12] = { 0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7c, 0x07, 0x7f, 0x67, 0x40, 0x00 };
72     unsigned char fnd_sel[4] = { 0x01, 0x02, 0x04, 0x08 };
73     char value_int, value_deci, num[4];
74     int i;
75     if ((value & 0x8000) != 0x8000) // Sign 비트 체크
76     {
77         num[3] = 11; else {
78             num[3] = 10;
79             value = (~value) - 1; // 2's Compliment
80         }
81
82     value_int = (char)((value & 0x7f00) >> 8);
83     value_deci = (char)(value & 0x00ff);
84     num[2] = (value_int / 10) % 10; num[1] = value_int % 10;
85     num[0] = ((value_deci & 0x80) == 0x80) * 5;
86     for (i = 0; i < 4; i++) {
87         PORTC = digit[num[i]];
88         PORTG = fnd_sel[i];
89         if (i == 1)
90             PORTC |= 0x80;
91         _delay_ms(2);
92     }
93 }
```



# Assignments

---

- I-Class에 다음 슬라이드의 과제 제출
- 제출 파일
  - 소스 코드(2 mandatory)
  - 보고서(.pdf)
  - 사진 파일 첨부
- 파일명
  - 12주차\_학번\_이름(압축하여 하나의 파일로 제출)
- 제출 기한
  - 다음주 화요일 자정까지



# Implement 1 (Mandatory)

---

- EX1.c 코드 응용
- 이론 슬라이드 참조하여 **ADMUX** 값 변경
  - **ADCH, ADCL** 값이 좌정렬되도록
- 8단계의 밝기값에 따라 켜지는 **LED**의 수 다르게 할 것
- **FND**에 읽어온 **ADC** 값 출력



# Implement 1 (Mandatory)

---

- Utilize EX1.c
- Change bits of ADMUX referring the slide of Monday class
  - In order to align the value of ADCH, ADCL to the left
- Differentiate the number of LED turned on according to brightness
- Display the value of ADC on FND



## Implement 2 (Mandatory)

---

- EX2.c 코드 응용
- FND에 화씨 온도 출력
- (Optional) 스위치 입력에 따라 섭씨, 화씨 변경 가능하도록



## Implement 2 (Mandatory)

---

- Utilize EX2.c
- Display in Fahrenheit temperature on FND
- (Optional) When there's an input from a switch, switch Celsius to Fahrenheit and vice versa

- 실습 후 Erase device 해주세요.
- Before you go out, please click “Erase device” button.

