

---

# ADC

마이크로프로세서

HRI 연구실

김동한



Human-Robot Interaction  
Laboratory



KYUNG HEE  
UNIVERSITY

# 중요. ADC (Analog to Digital Converter)

- A/D 컨버터(Analog-to-Digital Converter)
  - 아날로그 신호를 컴퓨터가 읽을 수 있는 병렬 또는 직렬의 디지털 데이터로 변환하여 주는 장치
  - 측정하려는 아날로그 물리량의 범위 및 시스템의 응용목적에 따라 분해능이나 정밀도가 적합한 것을 사용.
  - 변환시간(conversion time)
    - A/D변환을 수행하는데 필요한 시간
    - 소당 샘플링 속도(sampling rate)로 나타냄
  - 분해능(resolution)
    - 디지털 출력값을 한 등급만큼 변화시키기 위한 아날로그 입력의 최소변화(A/D 컨버터가 표현할 수 있는 최소 아날로그량)
    - n 비트 A/D컨버터의 경우 출력의 데이터 범위는  $1/2^n$

# 중요. ADC (Analog to Digital Converter)

## □ ATmega128의 A/D 컨버터 특징

- 10비트 분해능
- 0.5 LBS Integral Non-linearity(적분 비선형성)
- $\pm 2$  LBS 정확도
- 13~260usec 변환시간(50KHz~200KHz), 15kSPS의 최대 분해능,
- 8채널의 멀티플렉스된 단일 입력(A/D 컨버터는 한 개이며 채널을 바꿔가며 아날로그 신호를 입력 받음)
- 7채널의 차동입력, 10배 또는 200배의 증폭률을 가진 2채널의 차동입력, ADC 결과 값의 짝 정렬.
- 0~Vcc ADC 입력 전압 범위, 선택 가능한 2.56V의 ADC 레퍼런스 전압. 안정된 동작을 위한 MCU의 디지털 전원과 별도의 아날로그 외로 전원단자 A VCC를 가지며, A/D 변환에 필요한 기준전압 AREF 단자 지원
- Free running 또는 Single Conversion Mode, ADC변환 완료 인터럽트, Sleep Mode Noise Canceler



# 중요. ADC (Analog to Digital Converter)

## □ ATmega128 A/D 컨버터 레지스터

### □ ADMUX (ADC Multiplexer Selection Register)

- A/D 컨버터 멀티플렉서 선택 레지스터

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

- A/D 컨버터 제어 및 상태 레지스터 A

### □ ADCH, ADCL

- A/D 컨버터 데이터 레지스터

## □ ADMUX (ADC Multiplexer Selection Register)

### □ A/D 컨버터 멀티플렉서 선택 레지스터

- ADC모듈의 아날로그 입력 채널 선택
- ADC모듈의 기준 전압 선택
- 변환 결과 레지스터의 데이터 저장영역 지정

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

# 중요. ADC (Analog to Digital Converter)

## □ ADMUX (ADC Multiplexer Selection Register)

### □ 비트 7,6 : REFS1,0 (Reference Selection Bit)

- ADC모듈에서 사용하는 기준전압을 선택하는 비트

ADC에 대한 Voltage Reference 선택 표

| REFS1 | REFS0 | Voltage Reference  |
|-------|-------|--|
| 0     | 0     | AREF, Internal Vref turned off                                       |
| 0     | 1     | AVCC with external capacitor at AREF pin                             |
| 1     | 0     | 예약   |
| 1     | 1     | Internal 2.56V Voltage Reference with external capacitor at AREF pin |

## 중요. ADC (Analog to Digital Converter)

- ADMUX (ADC Multiplexer Selection Register)
  - 비트 5 : ADLAR (ADC Left Adjust Result)
    - 이 비트를 1로 설정하면 변환결과가 ADC 데이터 레지스터에 저장될 때 ADC Data Register의 좌측으로 끝을 맞추어 저장
  - 비트 4~0 : MUX4~0 (Analog Channel and Gain Selection Bit)
    - ADC 모듈의 아날로그 입력채널을 선택하는 비트

MUX 비트에 의한 아날로그 입력 채널 선택표

| MUX4~0 | Single Ended Input | Positive Differential Input | Negative Differential Input | Gain |
|--------|--------------------|-----------------------------|-----------------------------|------|
| 00000  | ADC0               | N/A                         | N/A                         |      |
| 00001  | ADC1               |                             |                             |      |
| 00010  | ADC2               |                             |                             |      |
| 00011  | ADC3               |                             |                             |      |
| 00100  | ADC4               |                             |                             |      |
| 00101  | ADC5               |                             |                             |      |
| 00110  | ADC6               |                             |                             |      |
| 00111  | ADC7               |                             |                             |      |

## 중요. ADC (Analog to Digital Converter)

MUX 비트에 의한 아날로그 입력 채널 선택표

| MUX4-0 | Single Ended Input | Positive Differential Input | Negative Differential Input | Gain |
|--------|--------------------|-----------------------------|-----------------------------|------|
| 01000  | N/A                | ADC0                        | ADC0                        | 10x  |
| 01001  |                    | ADC1                        | ADC0                        | 10x  |
| 01010  |                    | ADC0                        | ADC0                        | 200x |
| 01011  |                    | ADC1                        | ADC0                        | 200x |
| 01100  |                    | ADC2                        | ADC2                        | 10x  |
| 01101  |                    | ADC3                        | ADC2                        | 10x  |
| 01110  |                    | ADC2                        | ADC2                        | 200x |
| 01111  |                    | ADC3                        | ADC2                        | 200x |
| 10000  |                    | ADC0                        | ADC1                        | 1x   |
| 10001  |                    | ADC1                        | ADC1                        | 1x   |
| 10010  |                    | ADC2                        | ADC1                        | 1x   |
| 10011  |                    | ADC3                        | ADC1                        | 1x   |
| 10100  |                    | ADC4                        | ADC1                        | 1x   |
| 10101  |                    | ADC5                        | ADC1                        | 1x   |
| 10110  |                    | ADC6                        | ADC1                        | 1x   |
| 10111  |                    | ADC7                        | ADC1                        | 1x   |
| 11000  |                    | ADC0                        | ADC2                        | 1x   |
| 11001  |                    | ADC1                        | ADC2                        | 1x   |
| 11010  |                    | ADC2                        | ADC2                        | 1x   |
| 11011  |                    | ADC3                        | ADC2                        | 1x   |
| 11100  |                    | ADC4                        | ADC2                        | 1x   |
| 11101  |                    | ADC5                        | ADC2                        | 1x   |

# 중요. ADC (Analog to Digital Converter)

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

### □ A/D 컨버터 제어 및 상태 레지스터 A

- ADC 모듈의 동작 설정
- ADC 모듈의 동작 상태 표시

| 7    | 6    | 5    | 4    | 3    | 2     | 1     | 0     |
|------|------|------|------|------|-------|-------|-------|
| ADEN | ADSC | ADFR | ADIF | ADIE | ADPS2 | ADPS1 | ADPS0 |

### □ 비트 7 : ADEN (ADC Enable)

- A/D 컨버터 작동유무 지정
  - 1로 설정하면 ADC 모듈 enable
  - 0으로 설정하면 ADC 모듈 disable.



# 중요. ADC (Analog to Digital Converter)

## □ 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

### □ 비트 5 : ADFR(ADC Free Running Select)

- 프리런닝 모드 설정
- 1 : Free running 모드로 설정
  - 자동으로 계속해서 AD 변환 실행
- 0 : 단일 변환 모드 (Single conversion mode)로 설정
  - 사용자가 시작하면 한번만 AD 변환을 실행

# 중요. ADC (Analog to Digital Converter)

## □ 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 I비트가 1로 설정되어 있으면 이 인터럽트가 발생되어 처리된다.

### □ 비트 3 : ADIE(ADC Interrupt Enable)

- A/D 변환완료 인터럽트 허용
  - AD변환 완료 인터럽트를 개별적으로 설정
  - SREG 레지스터의 I비트가 1로 설정되어 있어야 한다.

# 중요. ADC (Analog to Digital Converter)

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

### □ 비트 2~0 : ADPS2~0(ADC Prescaler Select Bit)

#### ■ A/D 컨버터 프리스케일러 선택

- ADC 모듈에 인가되는 클럭의 분주비를 선택한다.

#### ADPS에 의한 ADC Prescaler 설정표

| ADPS2 | ADPS1 | ADPS0 | 분주비 |
|-------|-------|-------|-----|
| 0     | 0     | 0     | 2   |
| 0     | 0     | 1     | 2   |
| 0     | 1     | 0     | 4   |
| 0     | 1     | 1     | 8   |
| 1     | 0     | 0     | 16  |
| 1     | 0     | 1     | 32  |
| 1     | 1     | 0     | 64  |
| 1     | 1     | 1     | 128 |

# 중요. ADC (Analog to Digital Converter)

## □ ADCH, ADCL

### □ A/D 컨버터 데이터 레지스터

- A/D 컨버터의 결과를 저장하는 레지스터
- 단극성 입력 사용시 (Single Ended Input)
  - ADMUX레지스터의 MUX(4:0)가 "00000" ~ "00111" 혹은 "11110" ~ "11111"
  - 변환결과가 10비트 양의 정수로 표시된다(0~1023)
- 차동입력을 사용시(Differential Input)
  - ADMUX레지스터의 MUX(4:0)가 "01000" ~ "11101"
  - 변환 결과가 10비트 2의 보수로 표현(-512~+511)한다.
- 반드시 ADCL(하위 데이터)를 먼저 읽어서 저장한 다음에 ADCH(상위 데이터)를 저장한다.



# 중요. ADC (Analog to Digital Converter)

## □ ADCH, ADCL

ADMUX 레지스터의 ADLAR = 0 인 경우 : 우생렬

|      |    |    |    |    |    |    |      |      |
|------|----|----|----|----|----|----|------|------|
| ADCH | 15 | 14 | 13 | 12 | 11 | 10 | 9    | 8    |
|      | -  | -  | -  | -  | -  | -  | ADC9 | ADC8 |

|      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|
| ADCL | 7    | 6    | 5    | 4    | 3    | 2    | 1    | 0    |
|      | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 | ADC1 | ADC0 |

ADMUX 레지스터의 ADLAR = 1 인 경우 : 좌생렬

|      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|
| ADCH | 15   | 14   | 13   | 12   | 11   | 10   | 9    | 8    |
|      | ADC9 | ADC8 | ADC7 | ADC6 | ADC5 | ADC4 | ADC3 | ADC2 |

|      |      |      |   |   |   |   |   |   |
|------|------|------|---|---|---|---|---|---|
| ADCL | 7    | 6    | 5 | 4 | 3 | 2 | 1 | 0 |
|      | ADC1 | ADC0 | - | - | - | - | - | - |

# 중요. ADC (Analog to Digital Converter)

## □ ATmega128 A/D 컨버터의 동작 설정

- ADMUX로 A/D 신호를 입력받을 채널 선택
- ADCSR로 컨버전 프리스케일러 설정
- 인터럽트 사용시, 전역 인터럽트 플래그를 'SET' 하여 인터럽트 발생시키고, A/D 변환 완료 인터럽트 처리 루틴을 구성
- ADCSR의 6번 비트를 세트하여 A/D 변환을 시작
- 변환 완료 플래그를 주기적으로 점검하거나 아날로그 디지털 컨버전 완료 인터럽트를 이용하여 A/D 컨버전 이후의 데이터 처리 루틴을 구성
- 반드시 ADCL(하위 데이터)를 먼저 읽어서 저장한 다음에 ADCH(상위 데이터)를 저장



## 중요. ADC (Analog to Digital Converter)

- **ATMega128의 A/D 컨버터 설정시 클럭의 선택**
  - 10비트 분해능으로 정상적인 동작을 위해서는 50kHz~200kHz 범위의 클럭 사용
  - ADCSRA 레지스터의 ADPS2~0 비트에 의하여 2,4,8,16,32,64,128 중의 1가지로 선택
  - 프리스케일러는 ADCSRA 레지스터에서 ADEN=1로 설정한 경우에만 동작

# 중요. ADC (Analog to Digital Converter)

## A/D 컨버터 스타트

- ADC 시작신호 (ADSC) 1로 주면 됨

## A/D 변환

- ADC 변환 결과를 int형 변수 value에 저장하는 예
- `value = (ADCL)&0x00ff;`
- `value |= (ADCH << 8);`

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

[단극성 입력]

$$ADC = \frac{(V_{POS} - V_{NEG}) \cdot GAIN \cdot 512}{V_{REF}}$$

[차등입력]



# 중요. ADC (Analog to Digital Converter)

## A/D 변환 완료

- ADC 변환이 완료되면 변환값을 ADCL/H 에 저장하게 된다.

## A/D 변환 완료 인터럽트 요청

- ADC Conversion Complete Interrupt 요청

## ADC 상태플래그 셋

- AIDF = 1

## ADC 다음 동작 결정

- 단일/연속동작인지 구분하여 다음 동작 수행을 결정한다.  
(ADFR)

# 중요. ADC (Analog to Digital Converter)

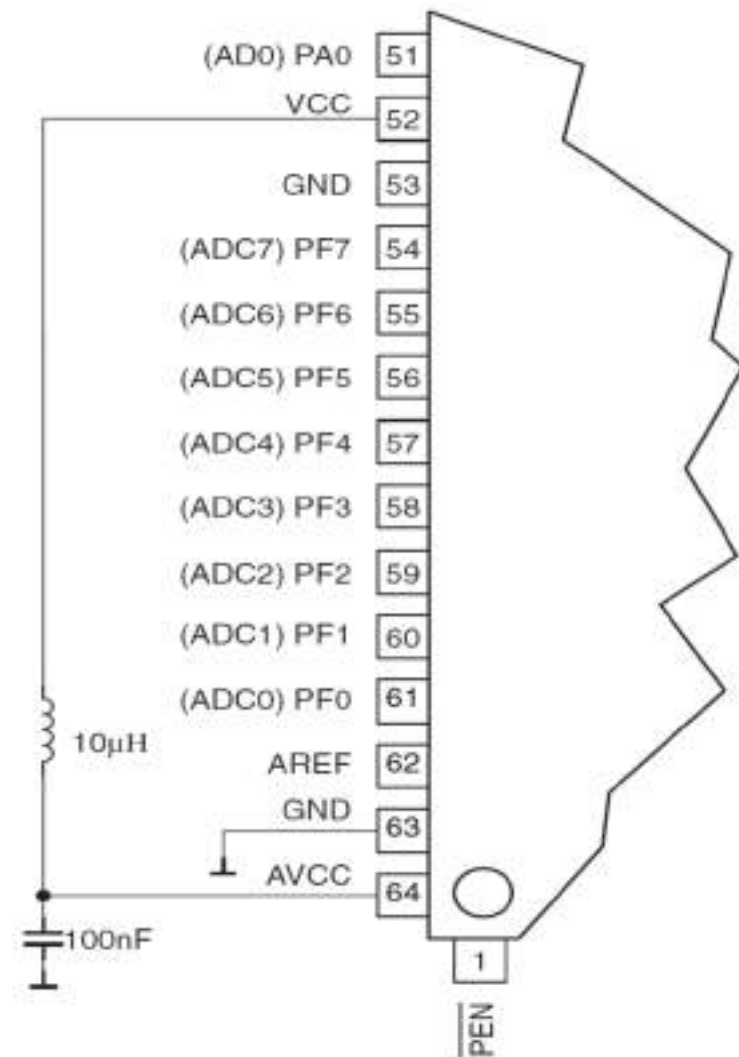
## A/D Convertor 잡음 제거 방법

- A/D Convertor의 경우에는 노이즈에 매우 민감하기 때문에 ATmega128 내에서도 AVCC, AREF, AGND와 같은 ADC 전원 구성도 따로 하였으며 사용자 또한 몇 가지 사항을 주의하여 사용해야 한다.
  - AVCC = 독립적인 아날로그 회로 전원 단자
  - AREF = 기준 전원 입력 단자
  - AGND = 아날로그 회로 접지 단자

# 중요. ADC (Analog to Digital Converter)

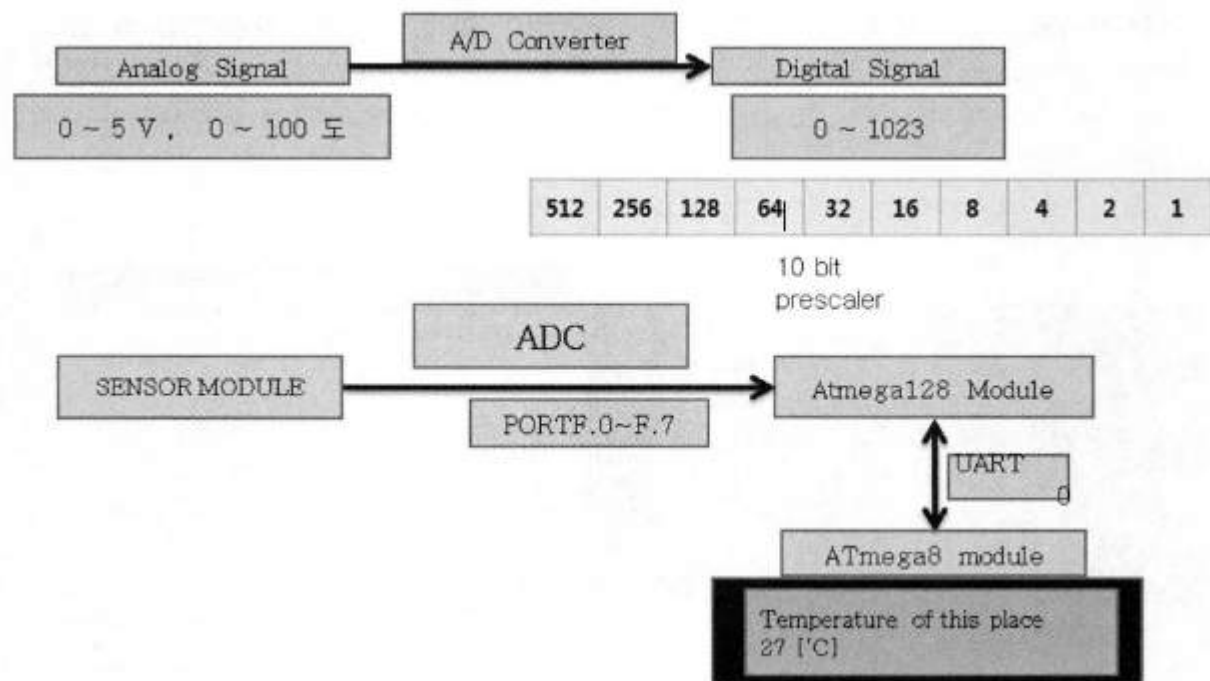
## A/D Converter 잡음 제거 방법

- ① 아날로그 입력선은 최소한으로 짧게 하고 잡음의 영향이 없도록 회로를 구성한다.
- ② 아날로그 전원단자 AVCC에 VCC를 인가할 때는 LC필터를 거쳐 안정화 시킨다.
- ③ 아날로그 회로의 모든 접지는 AGND에 접지하여 한 포인트에서만 GND와 접속한다.
- ④ ADC 동작중에는 병렬 I/O 포트의 논리상태를 스위칭하지 않는다.
- ⑤ 잡음에 민감한 아날로그 소자의 ADC의 경우에는 Adc Noise Reduction mode를 사용한다.
- ⑥ 잡음이 심하여 결과 값의 변동이 심하면 디지털 필터를 사용하거나 평균치를 구하여 사용한다.



## 6.3 LCD 제어의 이론 및 실습

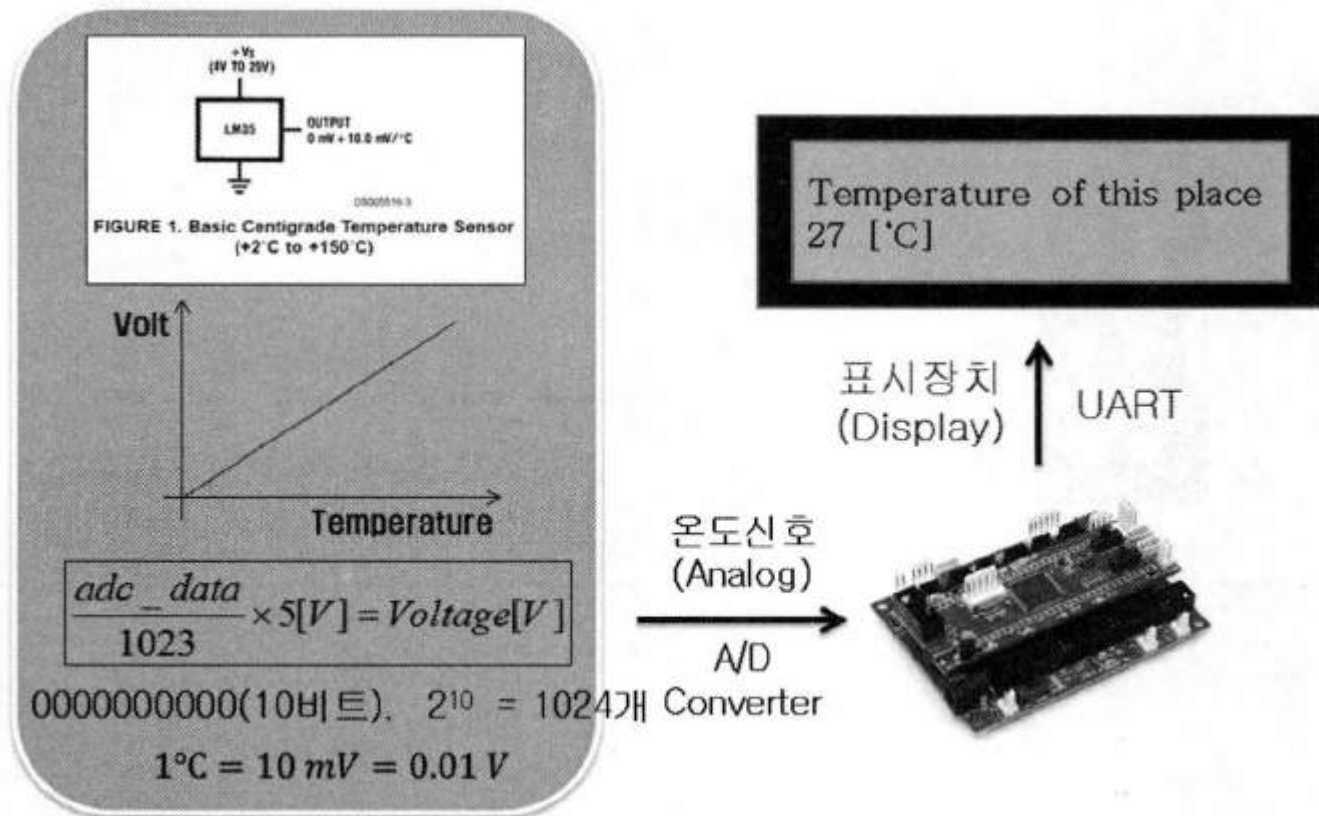
ATmega128의 ADC포트를 이용한 센서인터페이스 하기





## 6.3 LCD 제어의 이론 및 실습

### 온도센서 표시장치 개념도



## 6.3 LCD 제어의 이론 및 실습

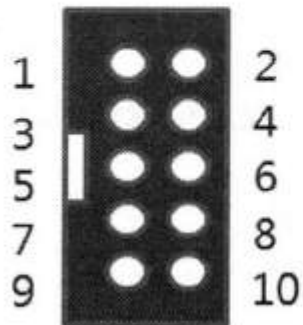
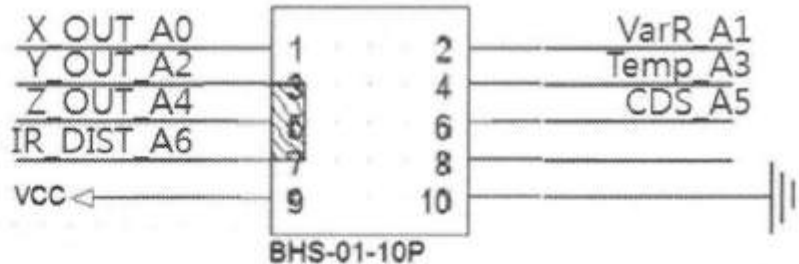
### Sensor module 이해하기

- 각종 센서의 값을 확인하기 위하여 read\_adc(n)에서 n의번호만 변경하면 읽을 수 있다.
- Sensor module은 PORTF.1에 가변 저항기가 연결되어 0에서 1023 까지 숫자로 변경된다.
- 이와 같이 PORTF.5에 조도센서(CDS), PORTF.6에 적외선 센서(IR), 그리고 PORTF.3 에 온도센서가 연결되어 있다.



| SENSOR MODULE |                             |
|---------------|-----------------------------|
| PORTF.0       | Acceleration Sensor, AXIS X |
| PORTF.1       | Variable Register           |
| PORTF.2       | Acceleration Sensor, AXIS Y |
| PORTF.3       | Temperature Sensor          |
| PORTF.4       | Acceleration Sensor, AXIS Z |
| PORTF.5       | CDS                         |
| PORTF.6       | IR Sensor                   |
| PORTF.7       | X                           |

TOP VIEW  
( ADC-DEV-P)



|            |   |    |         |
|------------|---|----|---------|
| X_OUT_A0   | 1 | 2  | VarR_A1 |
| Y_OUT_A2   | 3 | 4  | Temp_A3 |
| Z_OUT_A4   | 5 | 6  | CDS_A5  |
| IR_DIST_A6 | 7 | 8  |         |
| VCC(DC5V)  | 9 | 10 | GND     |

## 6.3 LCD 제어의 이론 및 실습

실습.8 온도센서를 이용한 LCD에 모니터 할 수 있는 프로그램

- ATmega128의 Sensor modul을 연결하여 PORTF.3에 온도센서를 연결
- 마이크로프로세서에 UART0 데이터 포트에 LCD를 연결
- 온도센서의 변화를 계산하여 온도 값으로 LCD에 표현

### 포트 설정

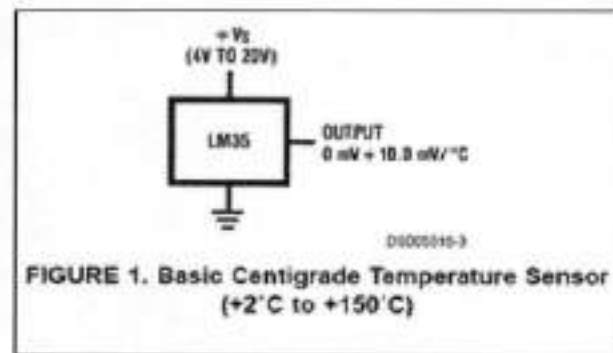
| Port B         | Port C | Port D | Port E | Port F |
|----------------|--------|--------|--------|--------|
| Data Direction |        |        |        |        |
| Bit 0 In       |        | Bit 0  |        |        |
| Bit 1 In       |        | Bit 1  |        |        |
| Bit 2 In       |        | Bit 2  |        |        |
| Bit 3 In       |        | Bit 3  |        |        |
| Bit 4 In       |        | Bit 4  |        |        |
| Bit 5 In       |        | Bit 5  |        |        |
| Bit 6 In       |        | Bit 6  |        |        |
| Bit 7 In       |        | Bit 7  |        |        |

### ADC설정

|   |                                     |
|---|-------------------------------------|
| ADC Settings                                    |                                     |
| <input checked="" type="checkbox"/> ADC Enabled | <input type="checkbox"/> Use 8 bits |
| <input type="checkbox"/> Interrupt              |                                     |
| Volt. Ref:                                      | AREF pin                            |
| Clock:  | 500.000 kHz                         |

LM35는 온도 1도에 0.01V의 변화로 체크되는 센서이다.

실제 0에서 100도가 0에서 5V로 변화하고 이것이 0~1023의 디지털 값으로 변환한다.  
그래서 온도로 변화하기 위해서는 아래와 같은 계산 값에 의하여 온도를 계산한다.

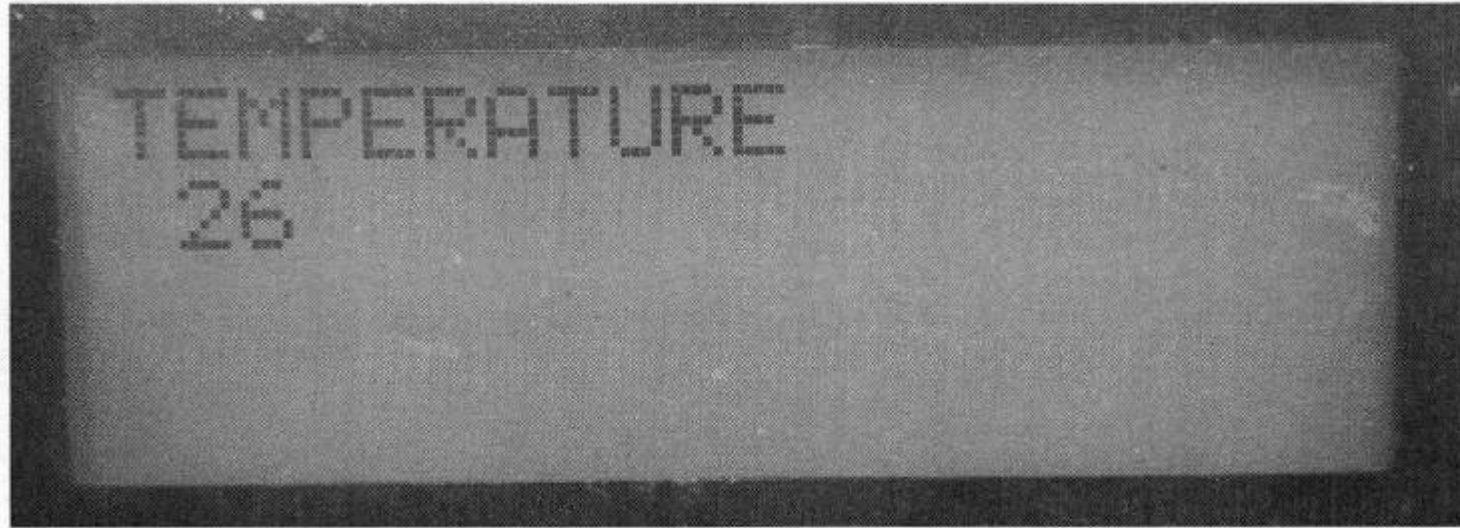


$$1^{\circ}\text{C} = 10\text{mV} = 0.01\text{V}$$

$$\frac{adc\_data}{1023} \times 5[V] = Voltage[V]$$

## 6.3 LCD 제어의 이론 및 실습

실습.8 온도센서를 이용한 LCD에 모니터 할 수 있는 프로그램 - 결과



```
include <stdio.h>
#include <mega128.h>
#include <delay.h>
// Voltage Reference: AREF pin
#define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) | (0<<ADLAR))
```



## 6.3 LCD 제어의 이론 및 실습

실습.8 온도센서를 이용한 LCD에 모니터 할 수 있는 프로그램 - 결과

```
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | ADC_VREF_TYPE;
    // Delay needed for the stabilization of the ADC input voltage
    delay_us(10);
    // Start the AD conversion
    ADCSRA|=(1<<ADSC);
    // Wait for the AD conversion to complete
    while ((ADCSRA & (1<<ADIF))==0);
    ADCSRA|=(1<<ADIF);
    return ADCW;
}
```

## 6.3 LCD 제어의 이론 및 실습

실습.8 온도센서를 이용한 LCD에 모니터 할 수 있는 프로그램 - 결과

```
void main(void)
{

    int buffer = 0;
    float adc_data = 0;

    // Port F initialization
    // Function: Bit7=In Bit6=In Bit5=In Bit4=In Bit3=In Bit2=In Bit1=In Bit0=In
    DDRF=(0<<DDF7) | (0<<DDF6) | (0<<DDF5) | (0<<DDF4) | (0<<DDF3) | (0<<DDF2) |
    (0<<DDF1) | (0<<DDF0);
    // State: Bit7=T Bit6=T Bit5=T Bit4=T Bit3=T Bit2=T Bit1=T Bit0=T
    PORTF=(0<<PORTF7) | (0<<PORTF6) | (0<<PORTF5) | (0<<PORTF4) | (0<<PORTF3) |
    (0<<PORTF2) | (0<<PORTF1) | (0<<PORTF0);
```

## 6.3 LCD 제어의 이론 및 실습

실습.8 온도센서를 이용한 LCD에 모니터 할 수 있는 프로그램 - 결과

```
// USART0 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART0 Receiver: Off
// USART0 Transmitter: On
// USART0 Mode: Asynchronous
// USART0 Baud Rate: 9600
UCSR0A=(0<<RXC0) | (0<<TXC0) | (0<<UDRE0) | (0<<FE0) | (0<<DOR0) | (0<<UPE0) | (0<<
U2X0) | (0<<MPCM0);
UCSR0B=(0<<RXCIE0) | (0<<TXCIE0) | (0<<UDRIE0) | (0<<RXEN0) | (1<<TXEN0) | (0<<UCSZ
02) | (0<<RXB80) | (0<<TXB80);
UCSR0C=(0<<UMSEL0) | (0<<UPM01) | (0<<UPM00) | (0<<USBS0) | (1<<UCSZ01) | (1<<UCSZ0
0) | (0<<UCPOL0);
UBRR0H=0x00;
UBRR0L=0x67;
```

## 6.3 LCD 제어의 이론 및 실습

실습.8 온도센서를 이용한 LCD에 모니터 할 수 있는 프로그램 - 결과

```
// ADC initialization
// ADC Clock frequency: 500.000 kHz
// ADC Voltage Reference: AREF pin
ADMUX=ADC_VREF_TYPE;
ADCSRA=(1<<ADEN) | (0<<ADSC) | (0<<ADFR) | (0<<ADIF) | (0<<ADIE) | (1<<ADPS2)
| (0<<ADPS1) | (1<<ADPS0);
SFIOR=(0<<ACME);

printf("$I\r");
```

## 6.3 LCD 제어의 이론 및 실습

실습.8 온도센서를 이용한 LCD에 모니터 할 수 있는 프로그램 - 결과

```
while (1)
{
// Please write your application code here
adc_data = read_adc(3);
buffer = (((float)adc_data/1023)*(5/0.01));
printf("$G, 1, 1\r");
printf("$T,TEMPERATURE \r");
printf("$G, 2, 1\r");
printf("$T, %2d\r",buffer);
}
}
```



## 6.3 LCD 제어의 이론 및 실습

[과제7] 가변저항기의 변화를 LCD 액정에 모니터 할 수 있는 프로그램

- ATmega128의 센서모듈을 연결하여 PORTF.1에 가변저항을 연결
- 마이크로프로세서에 UART0 데이터 포트에 LCD를 연결
- 가변저항기의 변화에 따른 0~1023의 값을 LCD에 표현

