
마이크로컨트롤러 기능

제 12장 A/D 컨버터



ICAT

Integrated Circuits for Advanced Technology Lab.

A/D 컨버터

1. ATmega128A의 A/D 컨버터 기능
2. A/D 컨버터로 광센서 읽기
3. 직렬 D/A 변환기로 LED 밝기 제어하기



ATmega128A의 A/D 컨버터 기능

□ ATmega128A의 A/D 컨버터 특징

- ❖ 10비트 분해능
- ❖ 0.5 LSB Integral Non-linearity(적분 비선형성)
- ❖ ± 2 LSB 정확도
- ❖ 13~260 μ sec 변환시간(50KHz~200KHz), 15kSPS의 최대 분해능,
- ❖ 8채널의 멀티플렉스된 단일 입력(A/D 컨버터는 한 개이며 채널을 바꿔가며 아날로그 신호를 입력 받음)
- ❖ 8채널의 단일입력, 10배 또는 200배의 증폭률을 가진 2채널의 차동입력, ADC 결과 값의 좌 정렬.
- ❖ 0~V_{cc} ADC 입력 전압 범위, 선택 가능한 2.56V의 ADC 레퍼런스 전압. 안정된 동작을 위한 MCU의 디지털 전원과 별도의 아날로그 회로 전원단자 AVCC를 가지며, A/D 변환에 필요한 기준전압 AREF 단자 지원
- ❖ Free running 또는 Single Conversion Mode, ADC변환 완료 인터럽트, Sleep Mode Noise Canceler



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128A의 A/D 컨버터 기능

□ ATMega128A 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



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128A의 A/D 컨버터 기능

□ADMUX (ADC Multiplexer Selection Register)

❖ A/D 컨버터 멀티플렉서 선택 레지스터

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

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



ATMega128A의 A/D 컨버터 기능

□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



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128A의 A/D 컨버터 기능

□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	
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128A의 A/D 컨버터 기능

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	
11110		1.22V(VBG)	N/A		
11111		0V(GND)			



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128A의 A/D 컨버터 기능

□ 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.



ATMega128A의 A/D 컨버터 기능

□ 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



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128A의 A/D 컨버터 기능

□ 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.



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128A의 A/D 컨버터 기능

□ ADCSRA(ADC Control and Status Register A)

- ❖ 비트 2~0 : ADPS2~0(ADC Prescaler Select Bit)
 - A/D 컨버터 프리스케일러 선택
 - ADC 모듈에 인가되는 클록의 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



ATMega128A의 A/D 컨버터 기능

□ADCH, ADCL

❖ A/D 컨버터 데이터 레지스터

- A/D 컨버터의 결과를 저장하는 레지스터
- 단일입력 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.



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128A의 A/D 컨버터 기능

□ADCH, ADCL

ADMUX 레지스터의 ADLAR = 0 인 경우 : 우정렬

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



ATMega128A의 A/D 컨버터 기능

□ADCH, ADCL

ADMUX 레지스터의 ADLAR = 1 인 경우 : 좌정렬

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	-	-	-	-	-	-



ATMega128A의 A/D 컨버터 기능

□ ATMega128A A/D 컨버터의 동작 설정

- ❖ ADMUX로 A/D 신호를 입력받을 채널 선택
- ❖ ADCSR로 컨버전 프리스케일러 설정
- ❖ 인터럽트 사용시, 전역 인터럽트 플래그를 'SET'하여 인터럽트 활성화하고, 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



ATMega128A의 A/D 컨버터 기능

□ ATMega128A의 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



실습 18 : A/D 컨버터로 광센서 읽기

□ 실습 개요

- ❖ ATmega128A의 A/D 컨버터 기능을 이용하여 광 센서(CdS)로부터 밝기 정보를 읽어내어 7-Segment FND에 출력
- ❖ CdS : 빛의 세기에 따라 내부 저항이 변하는데 밝은 곳에서는 내부 저항이 작아지는 광 가변 저항기

□ 실습 목표

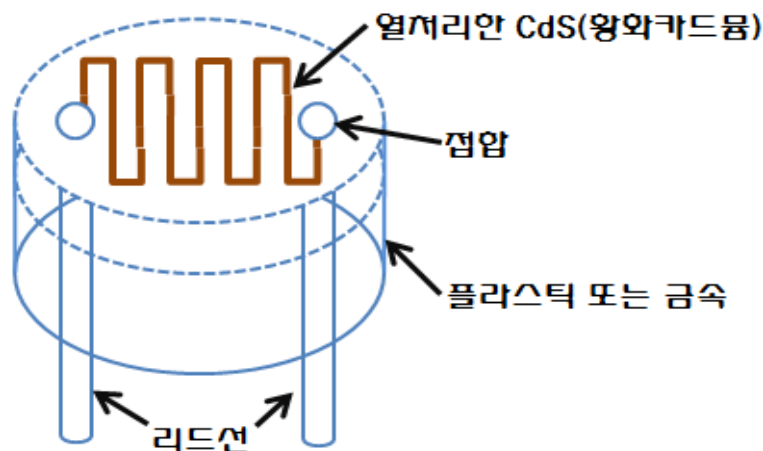
- ❖ ATmega128A A/D 컨버터의 동작 원리 이해
- ❖ A/D 컨버터 제어 방법 습득(레지스터 설정)
- ❖ CdS 동작 원리 이해



실습 18 : A/D 컨버터로 광센서 읽기

□CdS

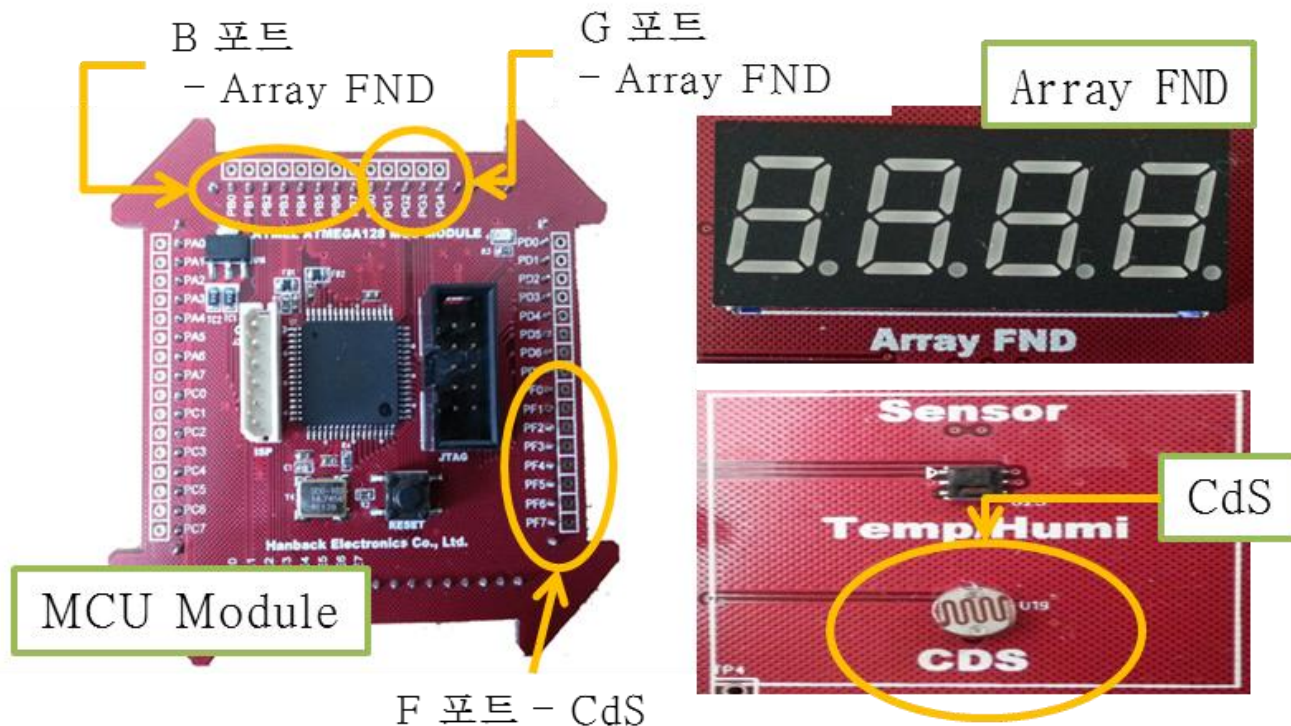
- ❖ 빛의 세기에 따라 내부 저항이 변하는데 밝은 곳에서는 내부 저항이 작아지는 광 가변 저항
- ❖ 저항은 옴의 법칙에 따라 전압에 비례하고 전류에 반비례($R = V/I$)한다.
- ❖ 밝은 곳에서는 저항이 작아져 전류가 증가하고 어두운 곳에서는 전류가 감소.



실습 18 : A/D 컨버터로 광센서 읽기

□ 사용 모듈

- ❖ MCU 모듈, Array FND 모듈, CdS 모듈

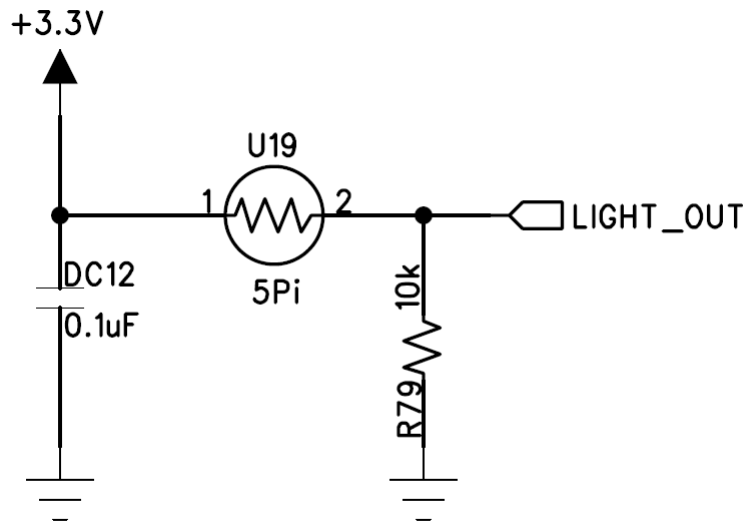


실습 18 : A/D 컨버터로 광센서 읽기

□ 사용 모듈

❖ 센서 모듈의 광센서 회로

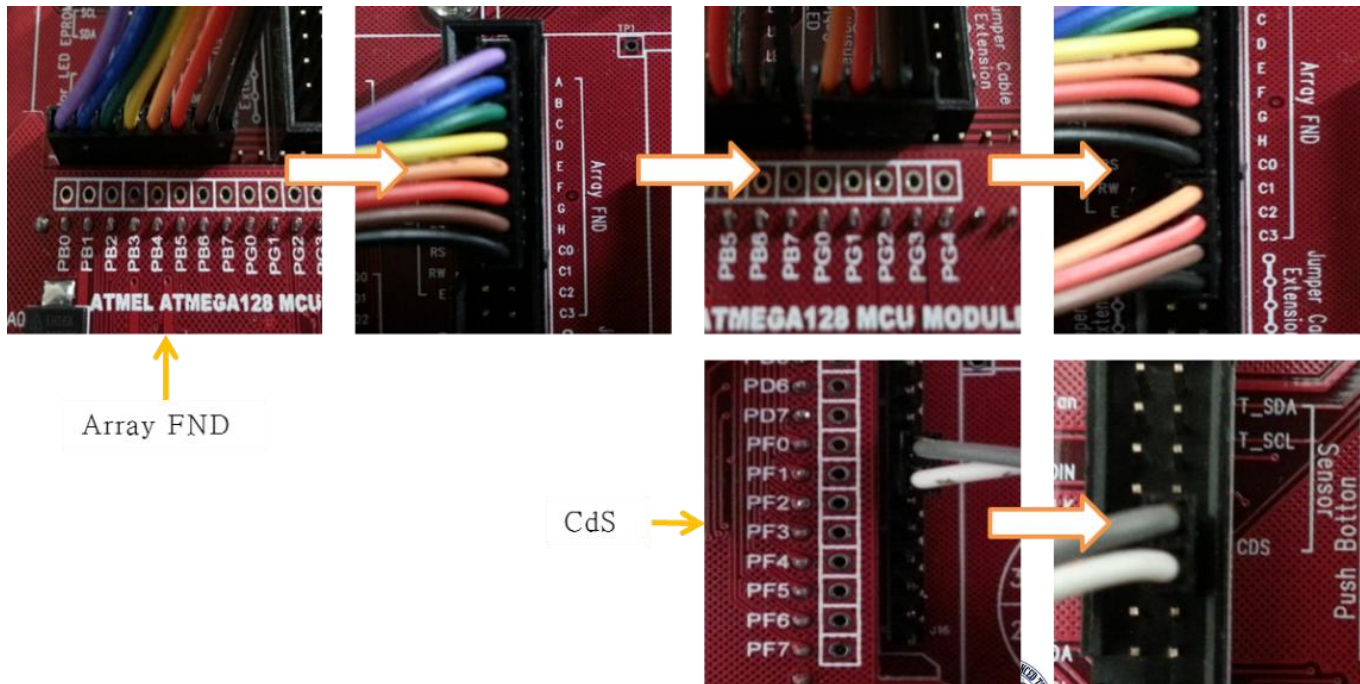
- LIGHT_OUT 신호가 Sensor 신호선 포트의 CdS에 매핑
- 빛이 들어오면 세기에 따라 저항 값이 변화



실습 18 : A/D 컨버터로 광센서 읽기

□ 모듈 결선 방법

- ❖ MCU 모듈 포트 E의 PE0~PE7을 Array FND 모듈의 A~H까지 연결
- ❖ MCU 모듈 포트 G의 PG0~PG3을 Array FND 모듈의 C0~C3핀에 연결
- ❖ MCU 모듈 포트 F의 PF0을 CdS 모듈의 CDS핀에 연결.



실습 18 : A/D 컨버터로 광센서 읽기

- ❖ 구동 프로그램 : 사전 지식
- ❖ A/D 컨버터 기능을 이용하여 CdS에서 출력되는 빛의 세기에 대한 아날로그 신호를 받아 디지털로 변환한 뒤, 이를 Array-FND에 표시
- ❖ 타이머/카운터를 이용하여 1초간격으로 데이터를 읽어들이도록 한다.
- ❖ A/D 컨버터의 설정 방법
 - A/D 컨버터의 입력채널 결정 : 0번 채널 사용
 - 입력 데이터의 정렬 방법과 기준전압 선택 : 데이터 정렬은 디폴트인 우정렬로 하고, 기준 전압을 내부 2.56V
 - 프리스케일러와 인터럽트 결정 : 프리스케일러는 64분주를 사용하고, 인터럽트는 사용하지 않음.
 - 인터럽트를 사용하지 않으므로 설정 끝
- ❖ A/D 컨버팅 제어
 - 설정을 마친 후, ADCSR의 6번 비트를 세트하여 A/D 변환을 시작
 - A/D 변환 도중에는 변환 완료 플래그를 주기적으로 점검하면서 A/D 데이터 레지스터로부터 데이터를 읽어 들이면 된다.
 - 이때, 반드시 ADCL(하위 데이터)를 먼저 읽어서 저장한 다음에 ADCH(상위 데이터)를 읽어 내야 한다.



실습 18 : A/D 컨버터로 광센서 읽기

□ 구동 프로그램 : 소스 분석

❖ ADC.c

1)	<pre>#include<avr/io.h> #include<util/delay.h> #include<avr/interrupt.h> enum{C0,C1,C2,C3}; unsigned char FND_DATA_TBL [] = {0x3F, 0X06, 0X5B, 0X4F, 0X66, 0X6D, 0X7C, 0X07,0X7F, 0X67, 0X77, 0X7C, 0X39, 0X5E, 0X79, 0X71, 0X08, 0X80};</pre>
2)	<pre>unsigned char lowValue=0; unsigned char highValue=0; unsigned int adcValue=0; unsigned char timer0Cnt=0; void print_FND(char selCx, unsigned char data); SIGNAL(SIG_OVERFLOW0); //Timer0 Overflow0 ISP int main(){</pre>



ICAT

7	6	5	4	3	2	1	0
REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
7	6	5	4	3	2	1	0
ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0

3)	<pre> DDR_B = 0xFF; //data port PORT_B = 0x00; DDR_G = 0x0F; //control port 0-3 PIN PORT_G = 0x0F; </pre>
4)	<pre> TCCR0 = 0x07; TCNT0 = 112; //256-112 = 144 => 0.01s TIMSK = 1 << TOIE0; //오버플로우 인터럽트 TIFR = 1 << TOV0; </pre>
5)	<pre> print_FND(C0, adcValue%10); _delay_ms(3); print_FND(C1, (adcValue/10)%10); _delay_ms(3); print_FND(C2, adcValue/100); _delay_ms(3); } } </pre>

```

ADMUX=0xC0;
ADCSR=0xA6;
ADCSR |= 0x40; //ADC start conversion

sei();

while(1){

```



실습 18 : A/D 컨버터로 광센서 읽기

6)	<pre>SIGNAL(SIG_OVERFLOW0){ cli(); TCNT0 = 112; timer0Cnt++; if(timer0Cnt ==100){ //0.01s * 100 = 1s timer0Cnt = 0; while((ADCSR&0x10)==0x00); //ADC 변환이 끝날 때 까지 대기 lowValue = ADCL; highValue= ADCH ; adcValue = (highValue << 8) lowValue ; } sei(); }</pre>
7)	<pre>void print_FND(char selCx, unsigned char data){ switch(selCx){ case C3: //4번째 FND (좌측기준) PORTG &= 0xF0; PORTG = 0x0E; break;</pre>



실습 18 : A/D 컨버터로 광센서 읽기

```
7) case C2: //3번째 FND
    PORTG &= 0xF0;
    PORTG |= 0x0D;
    break;
case C1: //2번째 FND
    PORTG &= 0xF0;
    PORTG |= 0x0B;
    break;
case C0: //1번째 FND
    PORTG &= 0xF0;
    PORTG |= 0x07;
    break;
default:
    PORTG &= 0xF0;
    PORTG |= 0x0F;
    break;
}
PORTB = FND_DATA_TBL[data];
}
```



실습 18 : A/D 컨버터로 광센서 읽기

□ 실행 결과



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 19 : 직렬 D/A변환기로 LED 밝기 제어하기

□실습 개요

- ❖ DAC 모듈의 직렬 D/A 변환기를 이용하여 LED의 밝기를 조절하도록 한다.
- ❖ DAC로 디지털 데이터를 출력하여 DAC에서 변환된 아날로그 신호가 LED에 입력되어 그 값의 크기에 따라 LED의 밝기가 제어된다.

□실습 목표

- ❖ D/A 변환기의 동작 원리를 이해
- ❖ 직렬 D/A 변환기의 아날로그 출력 특성을 이해
- ❖ ATmega128A의 SPI 기능에 대한 프로그램 방법 습득



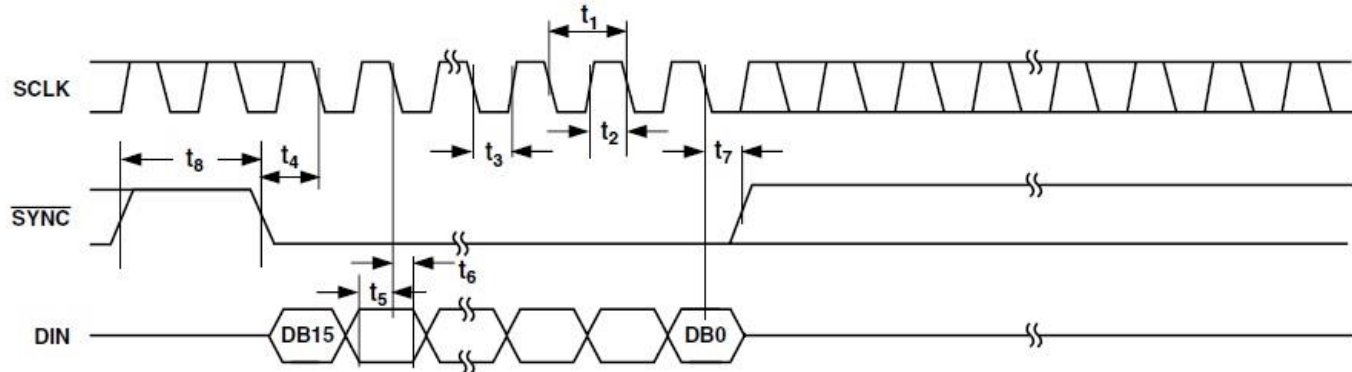
ICAT

Integrated Circuits for Advanced Technology Lab.

실습 19 : 직렬 D/A변환기로 LED 밝기 제어하기

❖ AD5317ARU

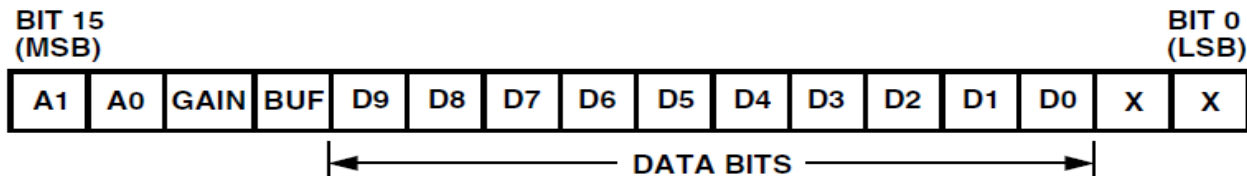
❖ AD5317ARU의 SPI 타이밍도



❖ AD5317ARU는 SCLK의 폴링 엣지에서 데이터를 읽음

- ATmega128A의 SPI mode 2(CPOL = 1,CPHA = 0) 와 동일

❖ AD5317ARU의 입력 데이터 정의



실습 19 : 직렬 D/A변환기로 LED 밝기 제어하기

❖ AD5317ARU

❖ AD5317ARU의 입력 데이터 정의

- 비트 15~14 : Address bits

A1(Bits15)	A0(Bits14)	DAC Addressed
0	0	DAC A
0	1	DAC B
1	0	DAC C
1	1	DAC D

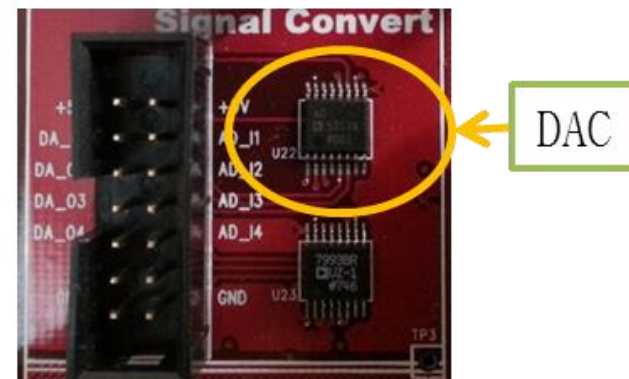
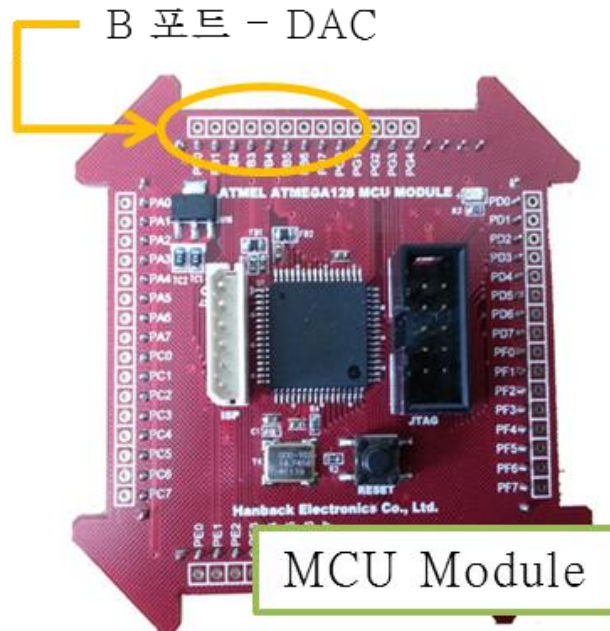
- 비트 13 (Gain) : DAC 출력의 범위를 설정한다.
- 비트 12 (BUF) : DAC의 기준 입력 전압을 선택한다.
- 비트 11~2 (D) : 10 bit DAC 레지스터 값이다.
- 비트 1~0 : 사용하지 않는 비트이다.



실습 19 : 직렬 D/A변환기로 LED 밝기 제어하기

□ 사용 모듈

- ❖ MCU 모듈, LED 모듈, DAC 모듈

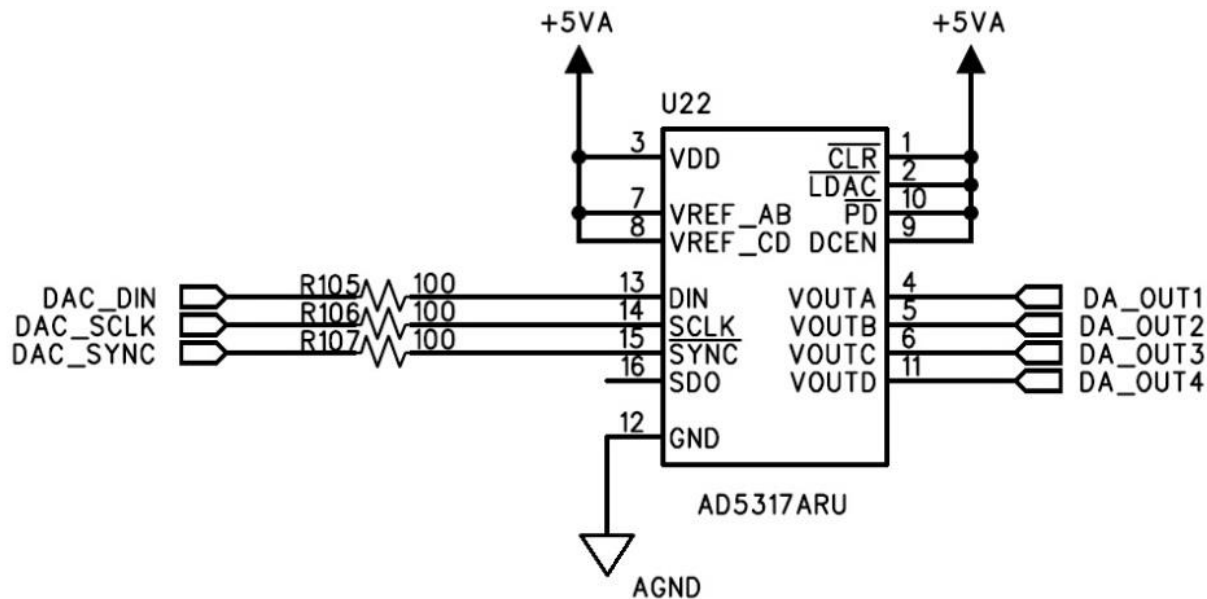


실습 19 : 직렬 D/A변환기로 LED 밝기 제어하기

□사용 모듈

❖ Serial D/A 컨버터 회로

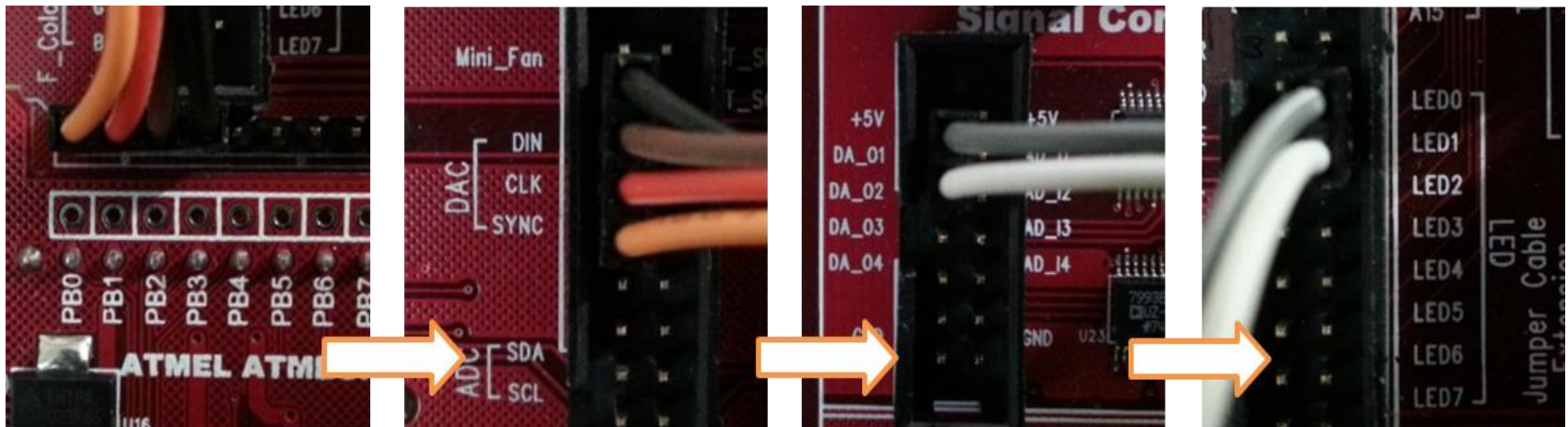
- AD5317ARU D/A 컨버터는 SPI로 제어되는 D/A 컨버터
- AD5317ARU는 최대 30MHz 클럭 속도에서 동작이 가능
- 4채널의 10bit D/A 컨버팅이 가능



실습 19 : 직렬 D/A변환기로 LED 밝기 제어하기

□ 모듈 결선 방법

- ❖ MCU 모듈 포트 B의 PB0(/SS)을 Serial Converter 모듈의 DAC_SYNC에, PB1(SCK)은 DAC_CLK에, PB2(MOSI)는 DAC_DIN에 연결
- ❖ Signal Convert 모듈의 DA_01(DAC 출력단자)은 LED 모듈의 LED0에 연결



실습 19 : 직렬 D/A변환기로 LED 밝기 제어하기

- ❖ 구동 프로그램 : 사전 지식(SPI 초기화)
- ❖ SPI 신호로 사용할 핀들을 설정하고, 입출력을 결정하여 SPI를 초기화하여야 한다.
- ❖ ATmega128A가 마스터로 사용될 경우, MISO는 입력으로, 나머지 MOSI, SCK, /SS는 출력으로 설정한다.
- ❖ 먼저 SPCR레지스터의 MSTR비트를 세팅하여, Master 모드를 사용
- ❖ AD5317ARU에 적용하기 위해, MSB 먼저 보내도록 설정
- ❖ CPOL과 CPHA 비트를 세팅하여, 클럭의 극성(Polarity)와 위상(Phase)을 SPI mode 2(CPOL = 1, CPHA = 0)로 결정
- ❖ 클럭의 주파수를 결정을 위해, SPCR의 SPR(1:0)과 SPSR의 SPI2X를 세팅
- ❖ SPI2X='0', SPR(1:0)="00"으로 하여, 4분주($14.7456\text{MHz}/4 = 3.6864\text{MHz}$)로 설정
- ❖ SPCR의 SPE 비트를 '1'로 세팅하여 SPI를 Enable



		7	6	5	4	3	2	1	0
	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
<hr/>									
		7	6	5	4	3	2	1	0
	SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X

❖ Serial_DAC.c

1)	<pre>#include <avr/io.h> #include <util/delay.h></pre>
2)	<pre>void Convert_sDAC(unsigned char ch,unsigned int tmp); int main(){ unsigned int VoIC = 0x00; DDRB = 0x07; PORTB = 0x01; SPCR = 0x54; SPSR = 0x00;</pre>
3)	<pre>while(1){ Convert_sDAC(0,VoiC); VoiC+=4; if(VoiC == 1020) VoiC = 0; _delay_ms(100);</pre>



실습 19 : 직렬 D/A변환기로 LED 밝기 제어하기

3)	<pre>} return 0; }</pre>
4)	<pre>void Convert_sDAC(unsigned char ch,unsigned int tmp){ unsigned int data=0; data = ch<<14; data = tmp<<2; // xx mode tmp xxxx // write sequence PORTB &= ~0x01; SPDR = (unsigned char)(data>>8); while((SPSR & 0x80)==0x00); //데이터 전송완료 될 때까지 대기 SPDR = (unsigned char)data; while((SPSR & 0x80)==0x00); //데이터 전송완료 될 때까지 대기 PORTB = 0x01; }</pre>



실습 19 : 직렬 D/A변환기로 LED 밝기 제어하기

□ 실행 결과

