

실험제목 : 아날로그-디지털 신호 변환기(Analog to Digital Converter)

실험목적

물리계에 존재하는 아날로그의 형태의 신호를 디지털 컴퓨터로 처리하기 위한 선행 작업인 아날로그-디지털 신호 변환기(Analog-to-Digital Converter, ADC)의 동작을 이해하고, 아날로그 신호를 디지털 신호로 변환하는 방법 및 이를 제어하기 위한 프로그램을 작성하여 본다.

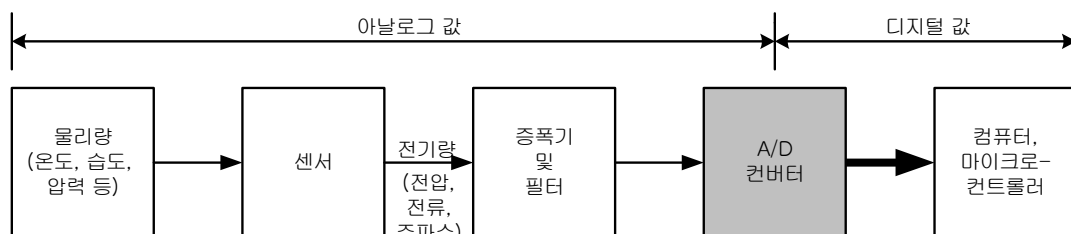
실험 준비물

- Microchip Studio 7
- RealTerm / TeraTerm
- Atmega328PB Xplained Mini
- Potentiometer (100kΩ)
- Pulse Sensor
- adc_monitor 프로그램

실험에 필요한 예비지식

1. A/D 변환기(ADC)

전압, 전류, 온도, 습도, 압력, 유량, 속도, 가속도 등과 같은 아날로그 물리량을 측정하여 이를 컴퓨터로 제어 또는 분석하려면 이들 신호를 디지털 값으로 변환하여 읽어 들여야 하는데 이러한 장치를 DAS(data acquisition system)라고 한다. DAS는 센서, A/D 변환기 및 컴퓨터 등으로 구성된다. 센서는 측정하려는 물리량을 전압, 전류 또는 주파수와 같은 전기량으로 변환하는 소자이며, A/D 변환기는 컴퓨터가 읽을 수 있는 병렬 또는 직렬 데이터로 변환하여 주는 장치이다. 대부분의 경우 센서에서 출력되는 전기량은 매우 미세하므로 이를 적절한 크기로 바꾸기 위한 증폭기가 사용되며, 센서와 A/D 컨버터의 사이에 잡음을 제거하고 필요한 신호만을 추출하기 위한 필터를 사용하는 경우도 많다.



<그림 9-1> Data acquisition system의 기본 구성

2. A/D 변환기 선택시 유의사항

1) 변환 시간(Conversion time) 또는 샘플링 속도(Sampling rate)

한번 A/D 변환을 수행하는데 필요한 시간을 변환시간(conversion time)이라고 하며, A/D 변환을 수행하기 위한 아날로그 값을 1초당 입력받는 회수를 샘플링 속도(sampling rate)라고 한다. 샘플링 속도는 주파수 단위(Hz)로 나타내며, 샘플링 속도가 100Hz라는 것은 초당 100개의 아날로그 값을 입력받아 변환한다는 의미이다. A/D 변환기는 변환시간에 따라 가격이 크게 달라지므로 응용목적에 적절한 변환 시간을 갖는 모델을 선정하는 것은 매우 중요하다. 예를 들어 디지털로 음성이나 영상신호를 처리하는 시스템에서는 고속형이 사용되어야 하지만, 온도나 수위측정에 사용되는 것은 변환시간이 긴 저속형 모델을 사용하여도 무방하다.

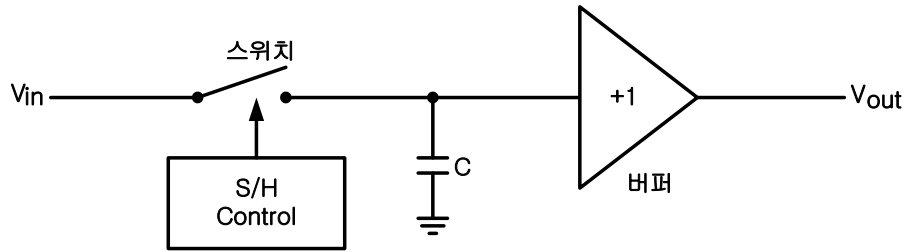
2) 분해능(Resolution)

A/D 변환기에서 분해능(resolution)이란 디지털 출력을 한 등급만큼 변화시키기 위한 아날로그 입력의 최소변화를 말한다. 이것은 A/D 변환기가 표현할 수 있는 최소 아날로그 값을 나타내는데, n-bit A/D 변환기의 경우 디지털 출력 데이터 범위는 $0 \sim 2^n - 1$ 이 되며, 이때 분해능은 Full scale을 2^n 으로 나눈 값이 된다. 예를 들어 $0 \sim 10V$ 의 Full scale을 갖는 전압을 4-비트의 A/D 변환기를 이용하여 A/D 변환할 경우 분해능은 $10/2^4 V = 0.625V$ 로 표시할 수 있다. 따라서 분해능의 값은 A/D 변환기의 비트 수가 커질수록 작아진다.

3) Sample/Hold 또는 Track/Hold

A/D 변환기가 아날로그 신호를 디지털 데이터로 변환하는 동안에 입력전압이 변동하면 변환된 디지털 출력값이 정확하지 않을 수 있다. 이를 방지하기 위하여 A/D컨버터의 입력단에 샘플/홀드(S/H : sample/hold) 또는 트랙/홀드(T/H : track/hold)회로를 사용한다. 특히, 축차 비교형(successive approximation) A/D 변환기는 변환기간 중에 아날로그 입력전압이 일정하게 유지되어야 하므로 샘플/홀드 기능이 꼭 필요하다. A/D 변환기에는 이러한 샘플/홀드 회로를 내장하고 있는 모델도 있고 외부에 별도로 연결해 주어야 하는 모델도 있다. 그러나 이러한 샘플링 동작 때문에 전체 변환시간이 길어지며, 멀티 채널형 A/D 변환기의 경우에는 일반적으로 아날로그 멀티플렉서(analog multiplexer)가 사용되므로 이것의 동작 및 안정시간으로 인하여 전체적인 변환시간은 훨씬 증가한다.

샘플/홀드 회로는 아날로그 스위치, 콘덴서, 버퍼 등으로 구성된다. 스위치는 샘플링 시간동안 닫혀 있어서 커패시터 양단의 전압이 입력전압과 동일하게 되도록 충전되고, 이후에는 스위치가 열려서 일정한 전압이 유지되면서 이 전압이 버퍼를 통하여 비교기로 입력되어 변환이 수행된다.



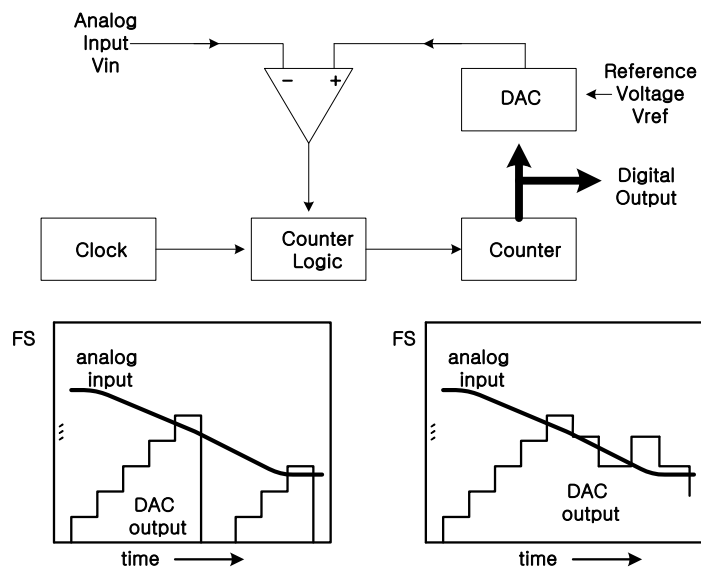
<그림 9-2> Sample/Hold 회로

3. A/D 변환기의 종류

1) 계수 비교형 ADC

계수형(counter-type) A/D 변환기는 내부에서 D/A 변환기로 발생시킨 전압이 아날로그 입력 전압보다 커질 때까지 비교하는 방식으로, D/A 변환기의 출력을 만들기 위한 계수기를 사용하며, 이 계수기는 변환을 시작할 때 리셋되고 각 클럭 사이클마다 1씩 증가되는 구조를 갖는다. 비교기는 D/A 변환기의 출력이 아날로그 입력전압을 초과하는 순간 계수기의 동작을 정지 시키며, 이 마지막의 계수 값이 디지털로 변환된 출력 값이 된다.

이 방식은 회로가 비교적 간단한 장점이 있지만, 변환시간이 길고 더구나 이것이 입력 신호의 크기에 따라 달라지는 단점을 가진다. 즉, n-bit 변환기의 경우 입력신호가 Full scale이면 변환시간이 가장 길어져서 2^n 클럭 주기가 필요하게 된다.

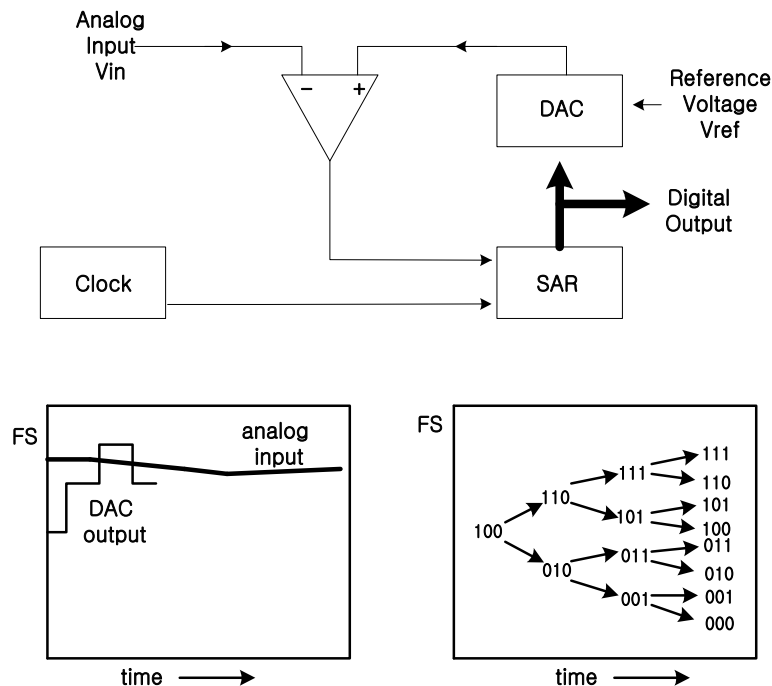


<그림 9-3> 계수형 및 추적 비교형 A/D 변환기

2) 축차 비교형 ADC

축차 비교형 또는 축차 근사형 A/D 변환기(successive approximation ADC)는 계수형과 유사해 보이나 계수기 대신에 SAR(successive approximation register)를 사용하여 이를 순차적으로 증가시키는 것이 아니라 최상위 비트부터 순서대로 하위 비트쪽으로 수정하여 가는 방법으로 DAC의 출력을 훨씬 빨리 아날로그 입력전압에 근사시킨다. <그림 8-8>의 예에서는 100→110→111→110의 과정으로 최종적인 디지털 변환 출력 110을 얻는다.

따라서 변환시간이 빠르고 회로도 간단하여 현재 저가형이면서 분해능이 높은 범용 A/D 변환기에 가장 널리 사용되고 있다. 그러나 변환 중에 아날로그 입력 전압이 일정하게 유지되어야 하므로 샘플/홀드 회로가 반드시 필요하다.



<그림 9-4> 축차 비교형 A/D 컨버터

3) 이중 적분형 ADC

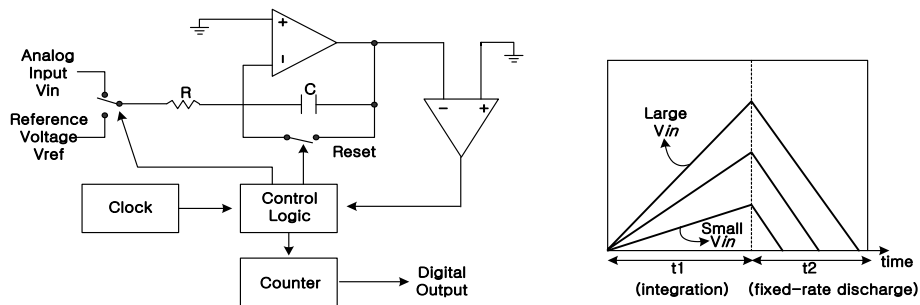
이중 적분형 A/D 컨버터(dual-slope integrating ADC)는 일정한 시간 t_1 동안 아날로그 입력신호 V_{in} 을 적분하고 나서 계수기를 리셋한 후에 다시 기준전압 V_{ref} 를 적분기의 출력이 0이 될 때까지 적분하여 그 시간 t_2 를 측정한다. 그러면 앞의 적분시간 동안의 충전 전하량과 뒤의 적분시간 동안의 방전 전하량은 같아야 하므로

$$t_1 V_{in} = t_2 V_{ref}$$

이 되고, 이로부터 입력전압은 다음과 같이 계산된다.

$$V_{in} = \frac{t_2}{t_1} V_{ref}$$

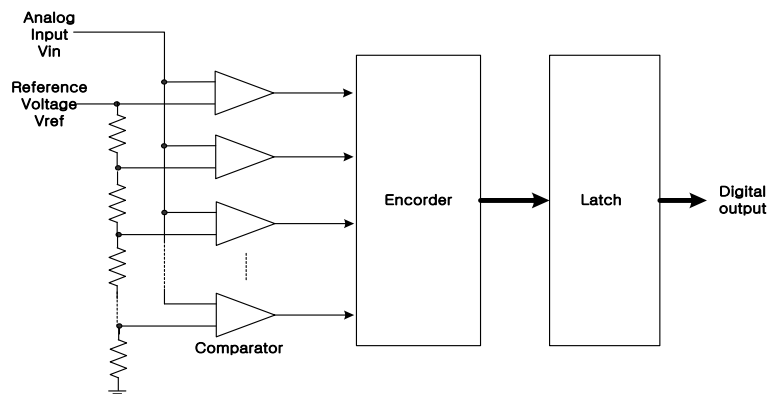
이것은 t_1 동안 아날로그 입력신호를 적분하므로 입력신호의 잡음에 대하여도 안정된 변환특성을 가지나, 두 번의 적분시간 때문에 변환시간이 늦어지는 것이 단점이다. 따라서 이 방식은 디지털 멀티미터(DMM, digital multi-meter)나 디지털 온도계와 같이 매우 저속으로 동작하는 시스템에 많이 사용된다.



<그림 9-5> 이중 적분형 A/D 변환기

4) 병렬비교형 ADC

병렬비교형(parallel ADC)또는 플래쉬형(flash ADC) A/D 변환기는 아날로그 입력신호를 여러 개의 사다리형태의 저항배치로 기준전압을 배분한 값과 각각의 비교기로 비교하는 방식을 사용하므로 한 단계로 비교가 완료되어 매우 빠른 변환시간을 갖는다. 그러나 높은 분해능을 갖게 하려면 정밀저항 회로와 비교기의 수가 많아져 회로가 복잡해지고 가격이 비싸지는 것이 단점이다. 이 방식은 가격이 높더라도 매우 빠른 처리속도가 요구되는 영상신호처리, 디지털 메모리 오실로스코프, 레이더 분야에 사용된다.

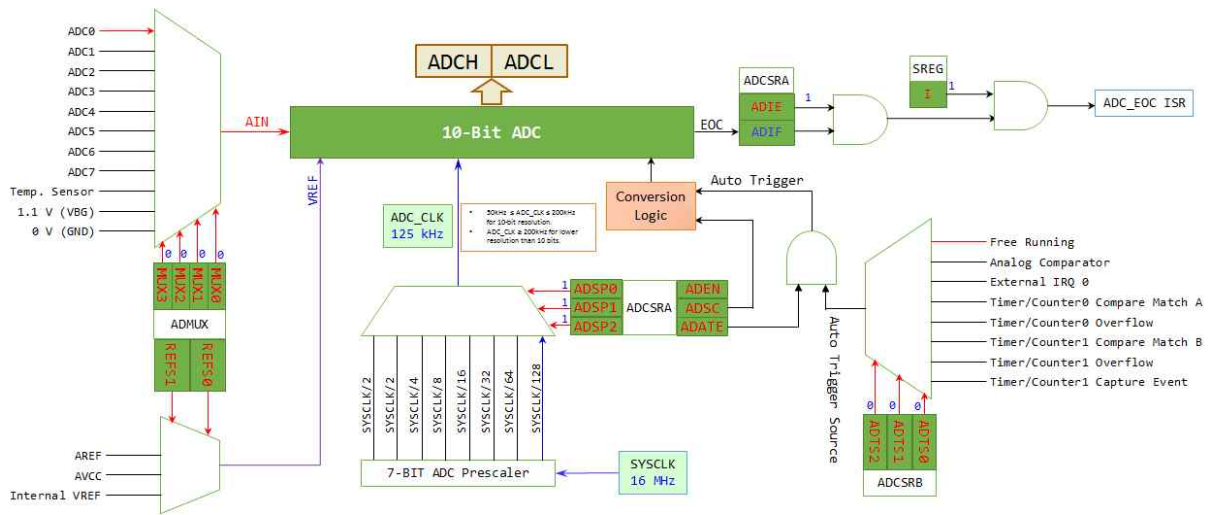


<그림 9-6> 병렬비교형 A/D 변환기

4. ATmega328PB의 ADC

ATmega328PB에는 10-bit ADC가 내장되어 있다.

1) ADC Block Diagram



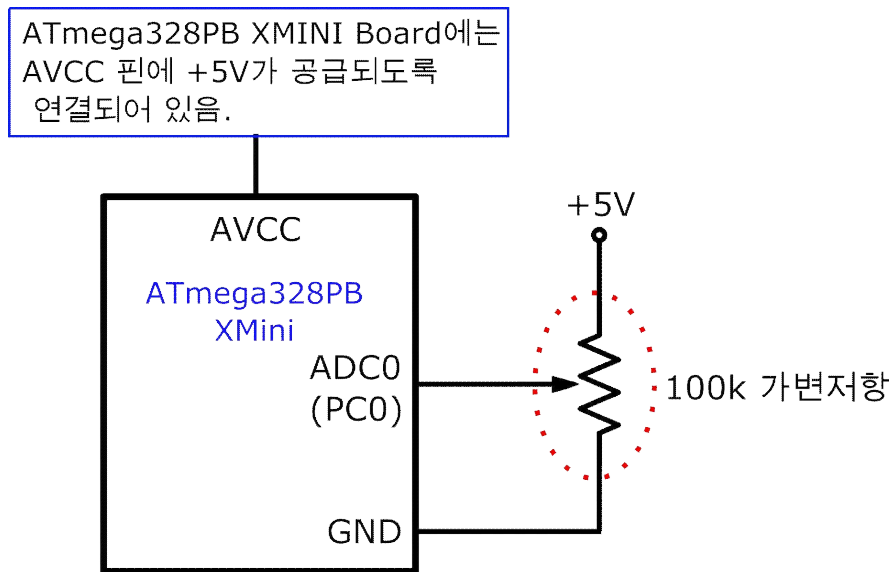
<그림 9-7> ATmega328PB ADC Block Diagram

- 10-비트의 분해능을 가지는 ADC이며, 8개 채널의 외부 아날로그 입력 신호와 3개의 내부 아날로그 입력 신호를 Analog Multiplexer를 통해 하나의 신호를 선택하여 변환할 수 있도록 설계되어 있다.
- 내부 아날로그 입력에는 내부의 온도 센서 출력과 내부에서 발생하는 1.1V의 전압 및 0V의 전압이 있다.
- ADC에 필요한 기준 전압(V_{REF})으로는 AREF 핀을 통해 공급되는 외부 전압, 내부에서 발생하는 1.1V의 전압, 혹은 AVCC 전압 가운데 하나를 선택할 수 있다.

실험 내용

1. 아래의 그림과 같이 ATmega328PB의 ADC0(PC0) 핀으로 입력되는 아날로그 전압을 연속해서 A/D 변환하여 그 결과를 UART0를 통해 PC의 Terminal 프로그램에 출력하는 프로그램을 작성하시오. 단, Terminal 프로그램을 38400bps의 전송 속도, 8 data bits, no parity, one stop bit로 설정하시오.

- (1) A/D 변환은 Polling 방식 사용
- (2) 기준 전압(V_{REF})으로는 AVCC를 사용
- (3) ADC Clock 주파수는 125 kHz를 사용

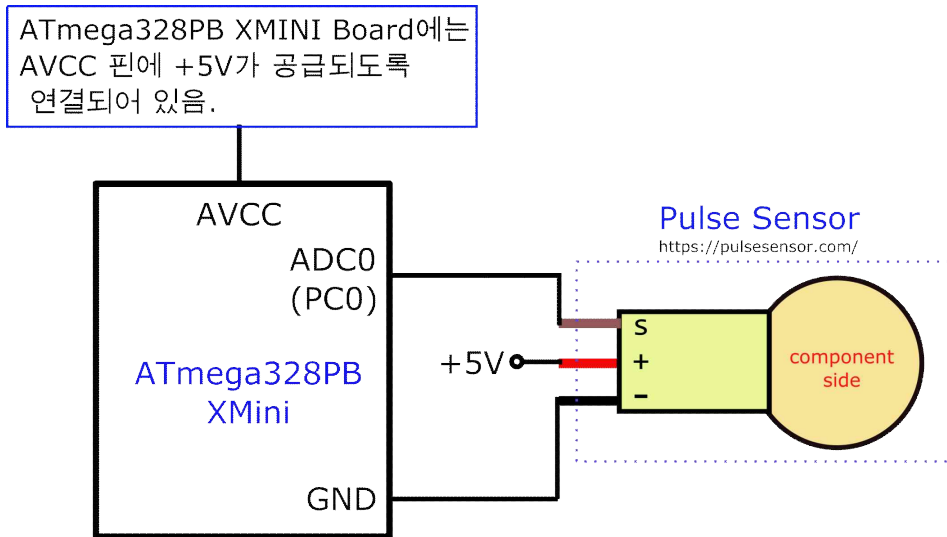


<그림 9-8> ADC 실험 결선도

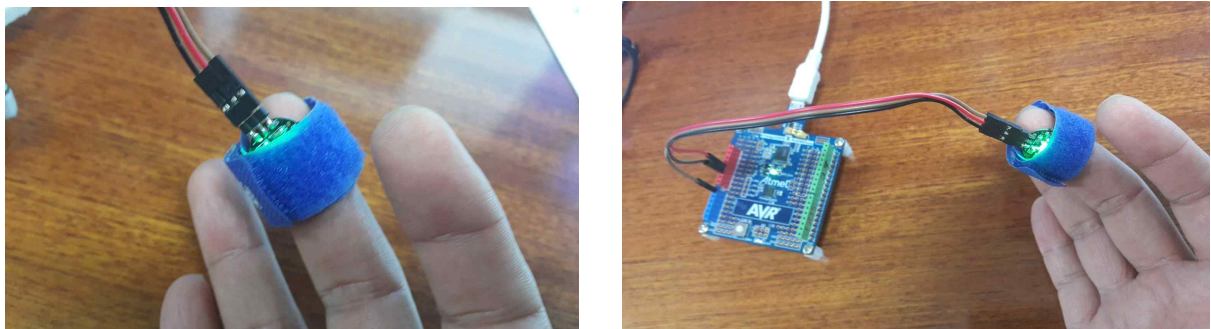
2. 위 1의 실험의 회로 구성에서 A/D 변환을 Polling 방식에서 Interrupt 방식 사용으로 변경한 후, 그 결과를 UART0를 통해 PC의 Terminal에 출력하시오.

- (1) A/D 변환은 Interrupt 방식 사용
- (2) 기준 전압(V_{REF})으로는 AVCC를 사용
- (3) ADC Clock 주파수는 125 kHz를 사용
- (4) Sampling Rate는 250 SPS (samples per second)가 되도록 설정.
- (5) ADC의 Auto Trigger Source로는 Timer/Counter0 Compare Match A를 사용.

3. 아래의 그림과 같이 위 1의 실험의 회로 구성에서 가변저항을 Pulse Sensor로 교체한 후, 위 2의 실험을 반복하시오. 단, A/D 변환 결과를 adc_monitor 프로그램을 사용하여 관찰하시오. 반드시 Pulse Sensor의 흰색 하트가 그려진 면이 피검자의 손가락에 닿도록 부착할 것. Pulse Sensor 참고 자료: <https://pulsesensor.com/>



<그림 9-9> Pulse Sensor 실험 결선도



<그림 9-10> Pulse Sensor 부착 예 및 연결도

ATmega328PB XMINI Board에서 printf() 함수를 사용하기 위한 준비 사항.

1. 강의 파일을 제공하는 웹 사이트를 통해 usart0.c 파일을 내려 받아 main.c와 동일한 폴더에 저장.
2. Microchip Studio7에서 현재의 프로젝트에 이 usart0.c 파일을 포함시킴.

```
/* usart0.c */

#define F_CPU 16000000UL

#include <avr/io.h>
#include <stdio.h>

int uart0_putchar(char ch, FILE *stream);
int uart0_getchar(FILE *stream);

FILE uart0_dev = FDEV_SETUP_STREAM(uart0_putchar, uart0_getchar, _FDEV_SETUP_RW);

void uart0_init(uint32_t baudrate)
{
    UCSRA = 0b00000000; // U2X0=0: No double speed
    UCSRB = 0b00011000; // Enable Rx and Tx
    UCSRC = 0b0000110; // Async mode, No Parity,
                        // 1 Stop bit, 8 Data bits
    UBRR0 = F_CPU/(baudrate*16UL)-1; // Baud Rate

    stdout = &uart0_dev;
    stdin = &uart0_dev;
}

int uart0_putchar(char ch, FILE *stream)
{
    while (!(UCSRA & (1 << UDRE0))); // Wait for empty transmit buffer
    UDR0 = ch;
    return 0; // sends the data
}

int uart0_getchar(FILE *stream)
{
    // Wait until data is received (RXC0 bit)
    while ((UCSRA & (1 << 7)) == 0);

    return UDR0; // Read received data
}
```

3. main.c 파일에 다음 내용 추가

```
/* main.c */

#include <stdio.h>           // 추가
void uart0_init(uint32_t baudrate); // 추가

void main(void)
{
    uart0_init(38400UL);     // 38,400 bps      // 추가
    while(1)
    {
        printf("ADC value = %d\n", adc_value); // Terminal 프로그램 사용시
        //printf("%d\n", adc_value);          // adc_monitor 사용시
    }
}
```

4. PC에 Terminal Program 설치

RealTerm이나 TeraTerm 설치

5. PC에 ATmega328PB XMINI Board를 연결한 후, 장치관리자에서 ATmega328PB XMINI Board에 할당된 COM port 번호 확인

6. 위 4에서 설치한 Terminal Program 실행

- 1) 위 5에서 확인한 COM port를 open.
- 2) 해당 COM port의 통신 규격을 아래와 같이 설정
38400 bps, 8 data bits, no parity, 1 stop bit

7. ATmega328PB XMINI Board에 있는 프로그램 실행.

속제

(이번 학기에는 아래의 1, 2번만 수행할 것)

1. 인터럽트 방식에 의한 ADC 제어 및 변환 결과를 SSD에 출력
 위 실험 2에서와 같은 인터럽트 방식으로 A/D 변환한 결과를 [6주차 실험 3. 시분할 방식에 의한 4-Digit SSD 표시 제어]에서 사용한 네 자리 SSD에 출력하시오. 단, SSD board에 부착된 가변 저항은 PC4에 연결되어 있다. 단, Timer/Counter0가 ADC의 auto-trigger 신호 발생을 위해 사용되고 있으므로 다른 Timer/Counter를 사용하여 SSD 표시를 제어하시오. 지정되지 않은 설정값은 자유롭게 선택하시오.

2. 가변 저항을 LED 밝기 제어
 위 실험 2와 같이 인터럽트 방식으로 A/D 변환 후, 그 변환 결과로 7 주차 실험 3에서 사용한 PWM 발생 장치의 duty cycle을 변화시키는 PWM 발생 장치를 프로그램을 작성하시오. 단, Timer/Counter0가 ADC의 auto-trigger 신호 발생을 위해 사용되고 있으므로 다른 Timer/Counter를 사용하여 PWM 신호를 발생시키고, 이 신호로 LED를 구동하여 가변 저항의 조절값에 따라 LED의 밝기가 변화하는 것을 확인하시오. 지정되지 않은 설정값은 자유롭게 선택하시오.

3. 주위의 밝기에 따른 조명 제어
 - 1) 입사되는 빛의 양에 따라 저항값이 달라지는 Photoresistor (Photocell 혹은 Light-Dependent Resistor, LDR) 소자인 CdS의 특성을 조사하여 보고서에 기술하시오.
 - 2) 위 실험 2에서 사용한 회로를 참조하여 CdS 소자를 이용하여 주위의 밝기를 디지털 값으로 변환하여 UART0를 통해 PC의 터미널로 출력하는 한편, 7 주차 실험 3에서 사용한 PWM 제어 회로를 참조하여 주변이 어두워지면 LED의 밝기를 증가시키도록 회로를 구성하고 프로그램을 작성하시오. 단, Timer/Counter0가 ADC의 auto-trigger 신호 발생을 위해 사용되고 있으므로 다른 Timer/Counter를 사용하여 PWM 신호를 발생시키시오.

4. 주위 온도에 따른 온도 제어
 - 1) 온도에 따라 저항값이 달라지는 Thermistor의 특성을 조사하여 보고서에 기술하시오.
 - 2) 위 실험 2에서 사용한 회로를 참조하여 Thermistor 소자를 이용하여 주위의 온도를 디지털 값으로 변환하여 UART0를 통해 PC의 터미널로 출력하는 한편, 7 주차 실험 3에서 사용한 PWM 제어 회로를 참조하여 주변의 온도에 따라 Heater에 공급되는 전류를 조절하여 일정한 온도를 유지하도록 회로를 구성하고 프로그램을 작성하시오. 단, Timer/Counter0가 ADC의 auto-trigger 신호 발생을 위해 사용되고 있으므로 다른 Timer/Counter를 사용하여 PWM 신호를 발생시키시오. (Heater 대신에 LED 사용)