

실험제목 : 범용 동기식 및 비동기식 직렬 통신 장치 (USART)

실험목적

ATmega328PB의 범용 동기식 및 비동기식 직렬 통신(USART) 장치와 PC에 내장된 가상 직렬 통신 포트(Virtual COM port)를 이용하여 실험용 보드와 PC사이의 직렬 데이터 통신 프로그램을 C언어를 이용하여 작성해 봄으로써 직렬 통신 과정을 이해한다.

실험 준비물

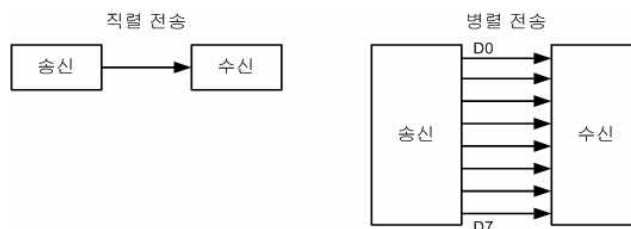
- Microchip Studio 7
- TeraTerm
- Atmega328PB Xplained Mini

실험에 필요한 예비지식

1. 직렬 통신 (serial communication)

1) 정의

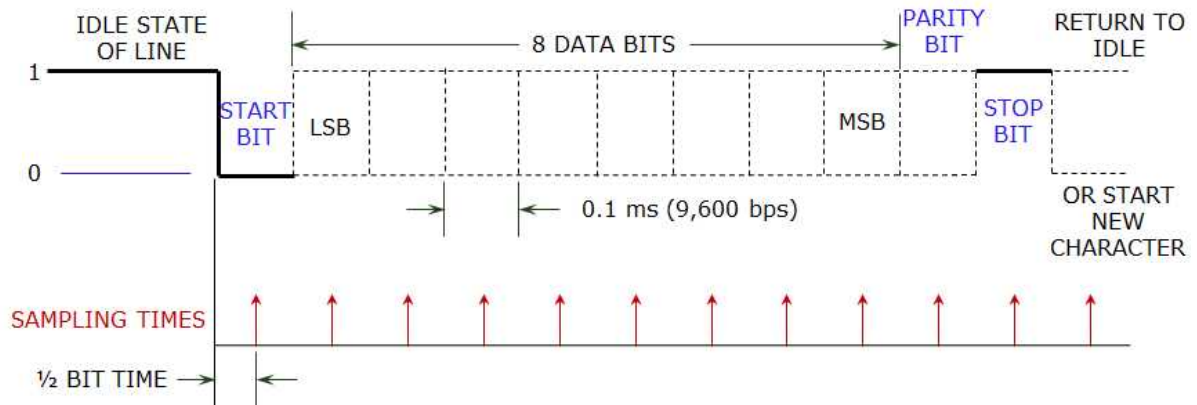
- ① 직렬 통신은 하나의 데이터 선을 이용하여 한 번에 1-비트씩 데이터를 전송
- ② 원거리 통신에 이용



<그림 4-1> 직렬과 병렬 전송

2) USART의 비동기식 직렬 통신시 데이터 프레임의 구성

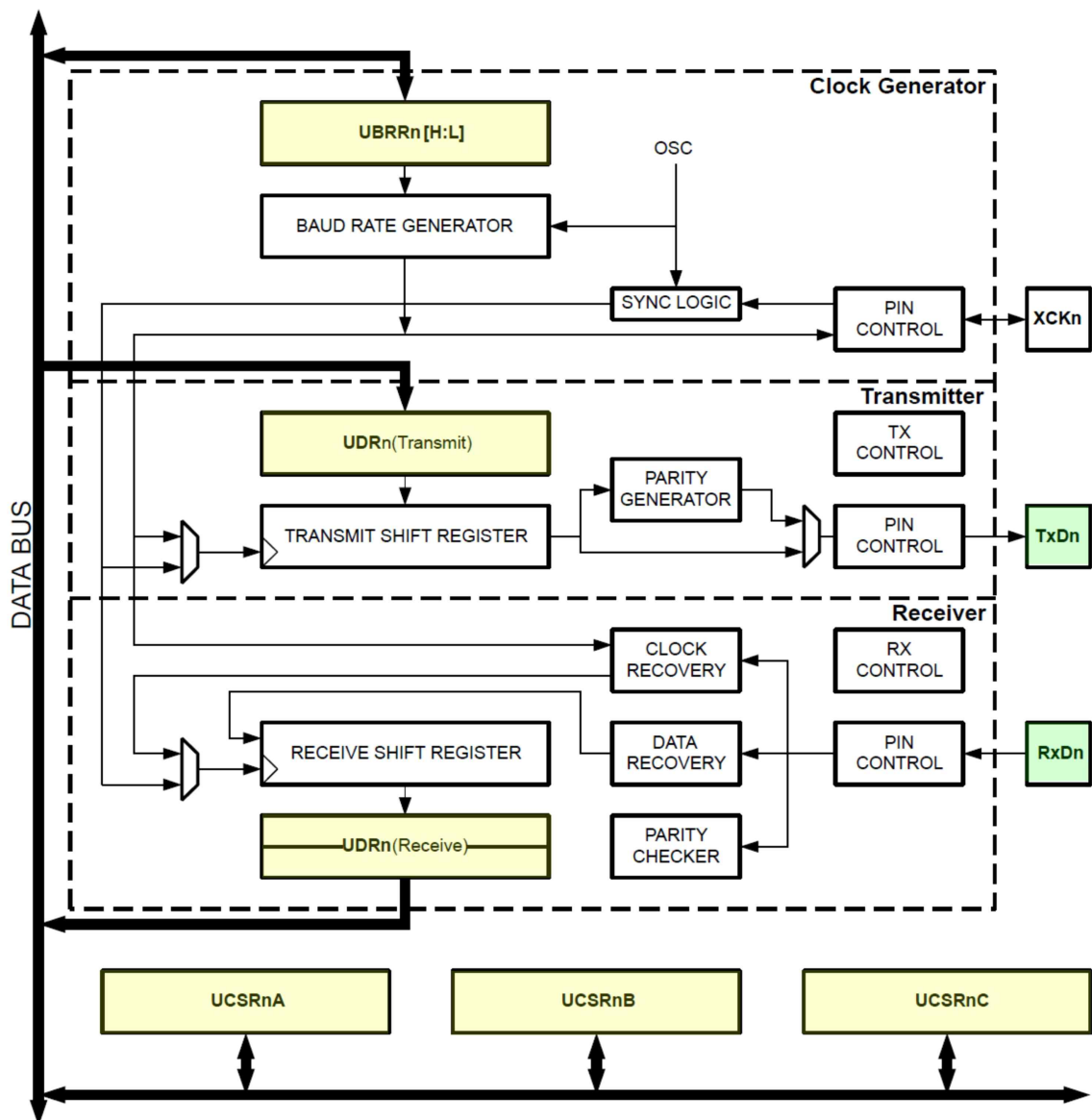
- ① 시작 비트(start bit) : 데이터 전송 시작을 알려줌
- ② 데이터 비트(data bit) : 전달하고자 하는 정보를 담은 비트로 7~9 비트
- ③ 정지 비트(stop bit) : 데이터 전송 끝을 알림
- ④ 패리티 비트(parity bit) : 데이터의 유효성 체크 비트



2. USART

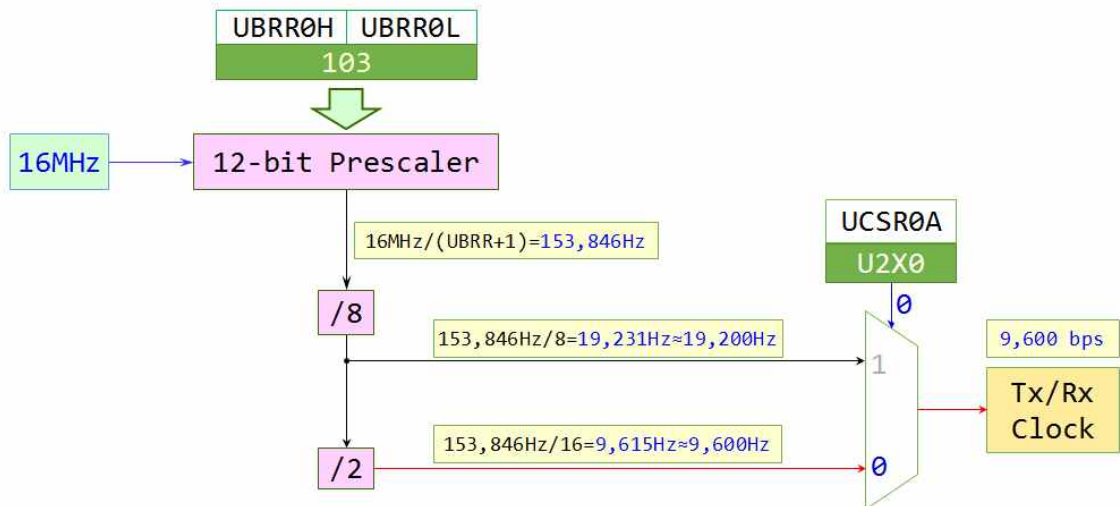
1) UART0와 UART1

Atmega328PB에는 동기식 및 비동기식 직렬 데이터 통신을 위한 2개의 USART 장치(USART0, USART1)가 내장되어 있다. USART를 통해서 전이중방식(수신과 송신이 서로 관련 없이 동시 진행)과 반이중 방식 그리고 멀티프로세서 통신이 가능하다. 다음은 USARTn의 블록다이어그램을 나타낸 것이다.



3. 비동기 직렬 데이터 통신 모드에서의 방식 Baudrate의 계산 예 (9600 bps의 경우)

SYSCLK = 16 MHz
 Baud Rate = $16,000,000 / (103 + 1) / 8 / 2 = 9,615\text{Hz} \approx 9,600\text{Hz}$

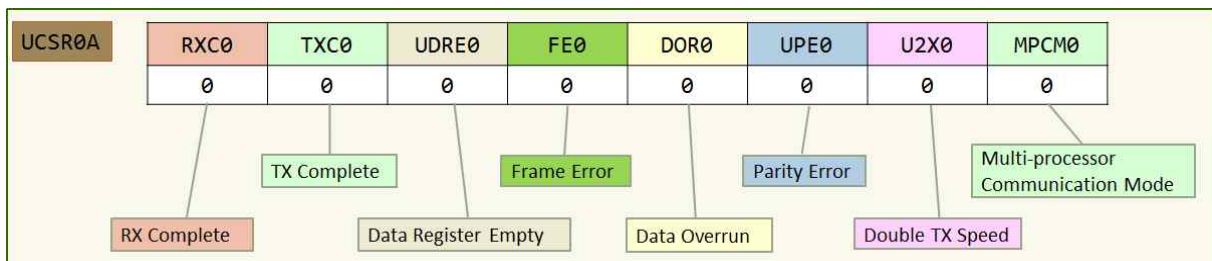


| UBRR0H | UBRR0L |
|--------|--------|
| 0 | 103 |

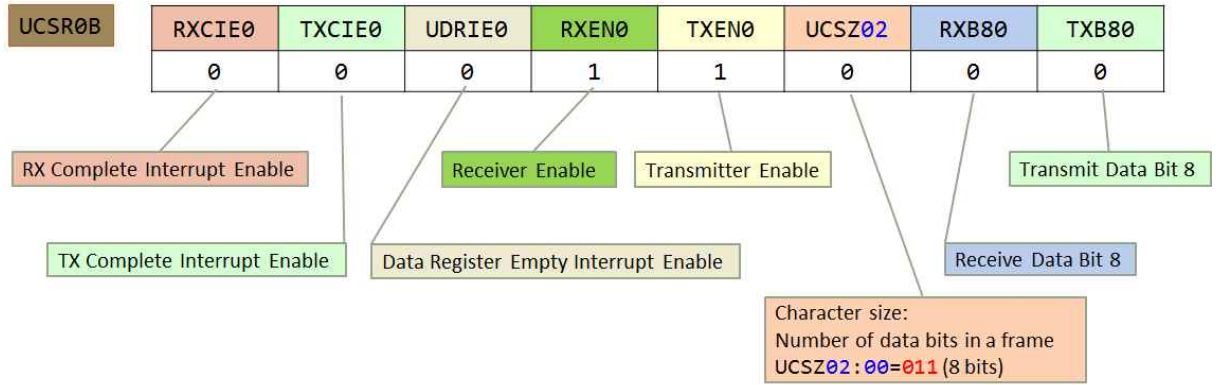
$$UBRR0 = \frac{f_{osc}}{16 \times BAUD} - 1 \quad (\text{when } U2X0=0)$$

4. USART 관련 레지스터

1) UCSRnA Register

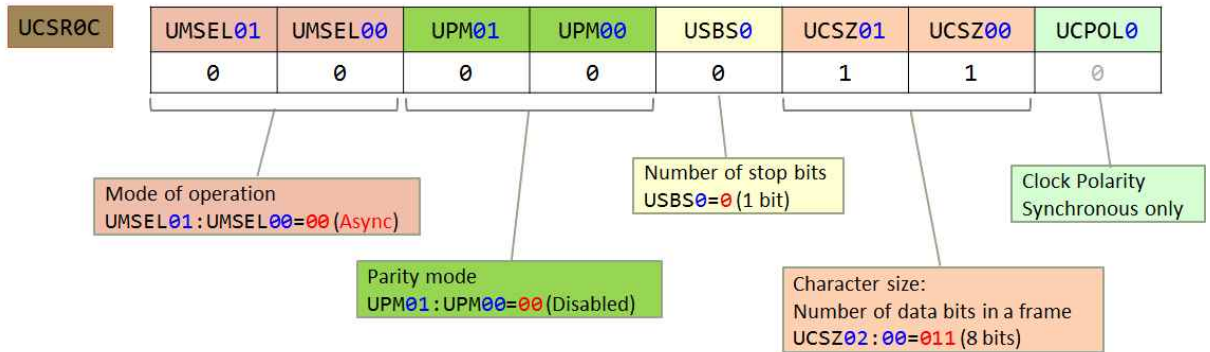


2) UCSRnB Register (Polling 방식의 예)



3) UCSRnC Register

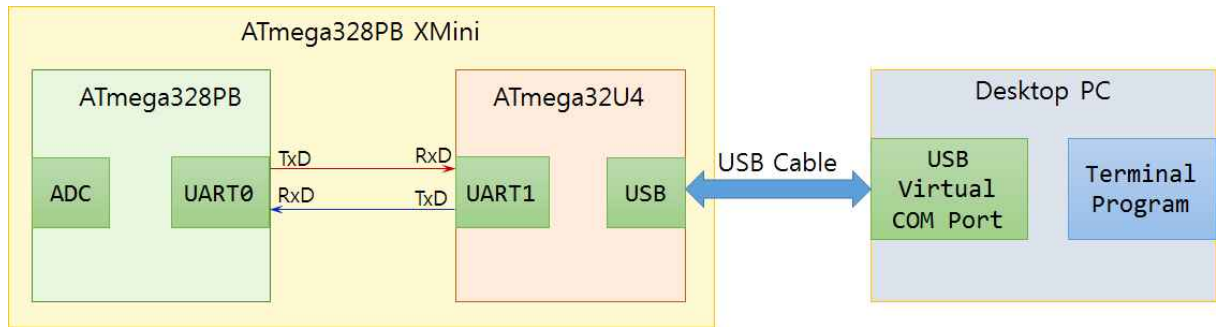
Asynchronous, no parity, 1 stop bit, 8 data bits



5. ATmega328PB XMini Board와 PC와의 연결

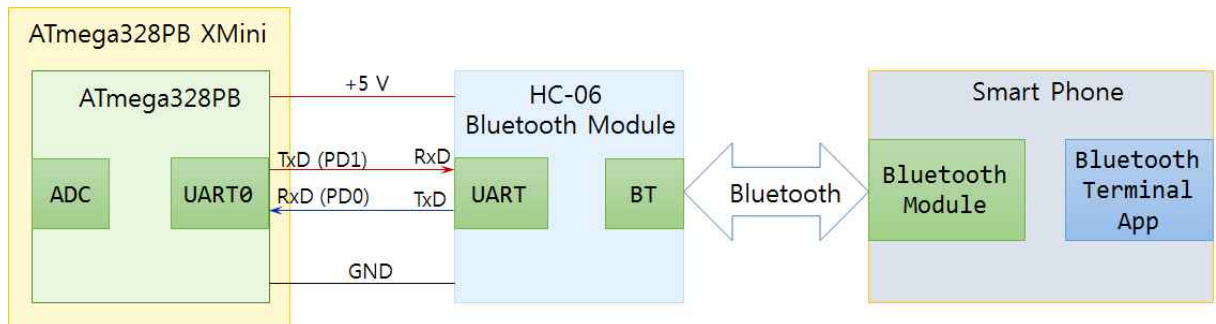
ATmega328PB XMini Board에는 아래의 그림과 같이 ATmega328PB에 내장된 UART0 포트의 TxD 신호와 RxD 신호가 ATmega32U4 마이크로컨트롤러에 의해 USB 신호로 바뀌어 PC측의 USB 포트에 연결된다.

ATmega328PB XMini 보드가 PC의 USB 포트에 연결되면 가상의 직렬통신(Virtual COM) 포트가 구성되며, 이 가상의 포트를 통해 ATmega328PB XMini 보드와 직렬통신을 할 수 있게 해 준다.



6. ATmega328PB XMini Board와 Smart Phone과의 연결

아래의 그림과 같이 ATmega328PB에 내장된 UART0 포트의 TxD 신호와 RxD 신호를 HC-06 Bluetooth Slave 모듈에 연결하면 스마트폰에 내장된 Bluetooth Module을 통해 스마트폰에 설치된 Bluetooth Terminal 응용 프로그램(App.)과 통신할 수 있다.



[참고] HC-06 Bluetooth Slave 모듈 규격은 아래에 링크된 문서를 참조할 것.

<https://www.waveshare.com/w/upload/5/5f/Bluetooth-UART-Board-User-Guide.pdf>

실험내용

1. USART0 비동기식 직렬 데이터 통신 기능과 터미널 프로그램을 이용하여 터미널 화면에 문자 'A'를 무한히 출력하는 프로그램을 작성하시오. 단, USART0의 UDRE0 비트를 polling 하는 방식을 사용하고, baud rate는 **9,600 bps**로 설정하고, USART0를 초기화하는 `uart0_init()` 함수와 한 문자를 출력하는 `uart0_putchar()` 함수를 직접 만들어 사용하시오.

```
#define F_CPU      16000000UL
#include <avr/io.h>

void uart0_init(uint32_t baudrate);
void uart0_putchar(char ch);

int main(void)
{
    _____ // Call uart0_init() function

    while (1)
    {
        _____ // Output 'A' to UART0
    }
}
```

```
void uart0_init(uint32_t baudrate)
{
    UCSR0A = _____; // U2X0=0: No double speed
    UCSR0B = _____; // Enable Tx, 8 Data bits
    UCSR0C = _____ // Async mode
                          // No Parity
                          // 1 Stop bit
                          // 8 Data bits
    UBRR0 = _____; // Baud Rate
}

void uart0_putchar(char ch)
{
    // Wait until Tx Data Register Empty
    while ( _____ );

    // Write a character(ch) to be transmitted
    UDR0 = _____;
}
```

2. USART0 비동기식 직렬 데이터 통신 기능과 터미널 프로그램을 이용하여 터미널 화면에 "Hello, BME!"라는 문자열이 무한히 반복해서 출력되도록 프로그램을 작성하시오. 단, USART0의 UDRE0 비트를 polling 하는 방식을 사용하고, baud rate는 **19,200 bps**로 설정하고, 위 1 에서 만든 두 개의 함수 이외에 문자열을 출력하는 `uart0_puts()` 함수를 직접 만들어 사용하시오. 이 함수 안에서 하나의 문자를 출력할 때는 위 1에서 구현한 `uart0_putchar()` 함수를 사용하시오.

```
void uart0_puts(char* str)
```

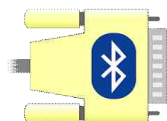
3. USART0 비동기식 직렬 데이터 통신 기능과 터미널 프로그램을 이용하여 PC의 키보드에서 입력한 문자를 수신한 후, 이를 PC로 되돌려 보내 PC의 터미널 화면에 출력되도록 프로그램을 작성하시오. 단, USART0의 UDRE0 비트와 RXC0 비트를 polling 하는 방식을 사용하고, baud rate는 115,200 bps로 설정하시오. 또한, 위 1 에서 만든 두 개의 함수 이외에 한 문자를 수신하여 반환하는 `uart0_getchar()` 함수를 직접 만들어 사용하시오.

```
char uart0_getchar(void)
```

4. PC의 터미널 프로그램에서 보내오는 문자를 수신하여 그 값이 문자 '1'이면 PB5에 연결된 LED를 점등하고 수신한 문자의 값이 '0'이면 LED를 소등하는 동작을 무한히 반복하는 프로그램을 작성하시오. 위 3에서 만든 `uart0_getchar()` 함수를 사용하시오.

이 프로그램이 정상적으로 동작하는 것을 확인한 후에 “실험에 필요한 예비 지식”의 6에서 설명한 그림을 참고하여 PC의 터미널 프로그램 대신 Smart Phone의 Bluetooth Terminal 프로그램을 사용하여 같은 동작이 이루어지는지 확인하시오. 단, Bluetooth Terminal 프로그램을 사용할 때는 UART0의 Baudrate를 9600 bps로 설정하시오.

참고: Play Store에서 Serial Bluetooth Terminal (아래 그림의 아이콘 참조)이라는 App을 다운로드하여 사용.



-
5. ADC0 핀을 통해 입력되는 아날로그 신호를 디지털로 변환한 후, 이 값을 10진수로 변환하여 USART0를 통해 터미널 프로그램으로 출력하는 프로그램을 작성하시오. 단, ADC의 경우 sampling rate는 250 sps (sample per second)로 하고, USART0의 baud rate는 500,000 bps로 설정하며, USART0의 UDRE0 비트를 polling하는 방법을 사용하시오. 단, 시스템에서 제공되는 표준 C 언어의 라이브러리 함수인 printf() 함수를 사용하여 출력하시오. 8주차 ADC 실험 내용 1번 항목과 유사한 내용이나 이번에는 USART0에 관련된 함수를 직접 만들어 사용한다. (다음 쪽의 설명 참조)

실험 내용 4번 항목에 대한 프로그램

```

/* main.c */
#include <stdio.h>
#include <avr/io.h>
void uart0_init(uint32_t baudrate);

int main(void)
{
    /* 이곳에 필요한 변수 선언 */

    uart0_init(500000UL);    // USART0 초기화 함수 호출

    /* 이곳에 초기화에 필요한 프로그램 작성 */

    while (1)
    {
        /* 이곳에 필요한 프로그램을 작성하고,
           필요하다면 printf() 함수 호출하여 출력 */
    }
}

```

```

/* usart0.c */
#define F_CPU      16000000UL
#include <avr/io.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)
{
    /* 이곳에 USART0의 초기화에 필요한 프로그램 작성 */

    stdout = &uart0_dev;
    stdin  = &uart0_dev;
}

int uart_putchar(char ch, FILE *stream)
{
    /* 이곳에 ch 변수값을 USART0로 출력하는 프로그램 작성 */

    return 0;
}

```

속제

위 실험내용 4에서 작성한 `uart0.c` 프로그램을 사용하면 표준 출력 함수인 `printf()` 뿐만 아니라 표준 입력 함수인 `scanf()`도 사용할 수 있다.

- 1) 7주차 실험 내용 1과 같이 회로를 구성하고 Timer/Counter0의 FastPWM 기능을 이용하여 PWM 신호를 만드시오. PWM 신호의 주파수는 244.14 Hz가 되도록 설정하시오.
- 2) TeraTerm에 “**0부터 255 사이의 정수를 입력하시오:**” 라는 안내문을 출력한 후, `scanf()` 함수를 사용하여 사용자로부터 TeraTerm을 통해 이 범위 내의 정수를 입력받은 후, 이를 위 1)에서 구현한 PWM 신호 발생 장치의 Duty Cycle(OCR0A 설정값)로 사용하시오. 범위 밖의 숫자가 입력되면 무시하고, 또한 Duty Cycle은 반드시 Timer/Counter0 Overflow ISR 안에서 변경할 것.
- 3) TeraTerm에서 입력한 값에 따라 PWM 신호의 Duty Cycle이 변화하는지 LED를 통해 확인하시오.
- 4) 위 2)와 3)의 동작을 무한히 반복한다.

주석이 첨부된 source 프로그램을 보고서로 작성하여 제출하시오.