UART

마이크로프로세서

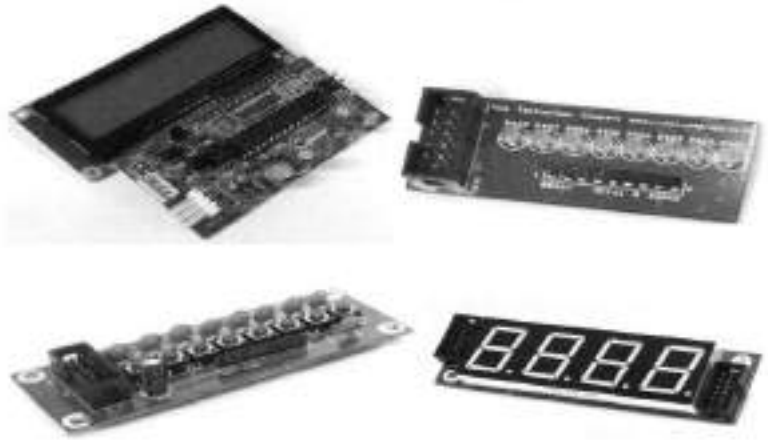
HRI 연구실

김동한

Human-Robot Interaction Laboratory

KYUNG HEE UNIVERSITY

Contents



KYUNG HEE UNIVERSITY

8.1 통신의 개요

8.1.1 통신이란?

- 사람과 사람 사이의 의사소통이나 정보를 교환하는 것

- 데이터 통신
  - 전송한 데이터가 목적지에 정확하게 전달되도록 정해진 순서나 규칙에 따라 데이터를 보내고 받는 과정
  - 전보, 전화 등 전달하고자 하는 목적물을 오류 없이 통신회선을 통하여 2진수로 표시된 디지털 형태의 정보로 교신하는 것

- 데이터 통신 시스템(Data communication system)
  - 데이터 전송 링크를 연결하는 하드웨어와 소프트웨어가 결합된 형태
  - 정보의 이동을 담당하는 전송시스템과 정보의 가공, 처리 및 보관을 위한 데이터 처리시스템(컴퓨터)으로 나눔

8.1 통신의 개요

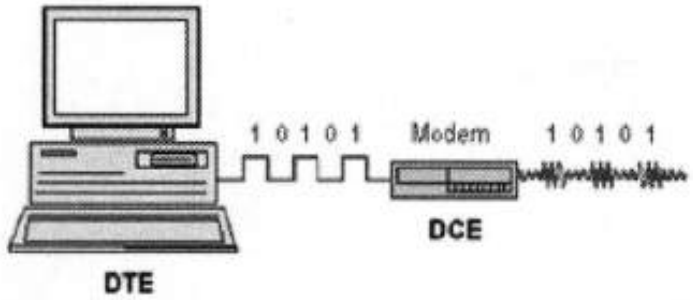
8.1.1 통신이란?

- 데이터 전송(data transmission)이란 정보통신 기기 간에 처리할 데이터를 네트워크 등의 전송매체를 사용하여 한 지점에서 다른 지점으로 보내는 것

- DTE(Data Terminal Equipment) : PC나 단말 등
- DCE(Data Communication Equipment) : 모뎀 등 통신 장치

8.1 통신의 개요

8.1.1 통신이란?



- 단말장치(DTE : Data Terminal Equipment)
  - 데이터 통신망에 연결된 사용자 장치의 일반적인 이름
  - 데이터의 전송 시스템에서 최종적으로 데이터를 송수신 하는 기능과 아울러 입출력 기능, 전송 제어 기능, 기억 기능을 갖고 있음
  - 대표적인 단말장치로는 컴퓨터 시스템이 있음

- DCE(Data Communication Equipment : 데이터 통신 기기)
  - 통신회선에 적합하도록 변환하는 기능과 통신 회선을 통해 수신된 데이터를 원래의 정보로 변환하는 기능을 갖는 장치
  - 일반적으로 컴퓨터가 모뎀이나 기타, 다른 직렬장치를 이용하여 데이터를 교환하기 위한 RS-232C 인터페이스

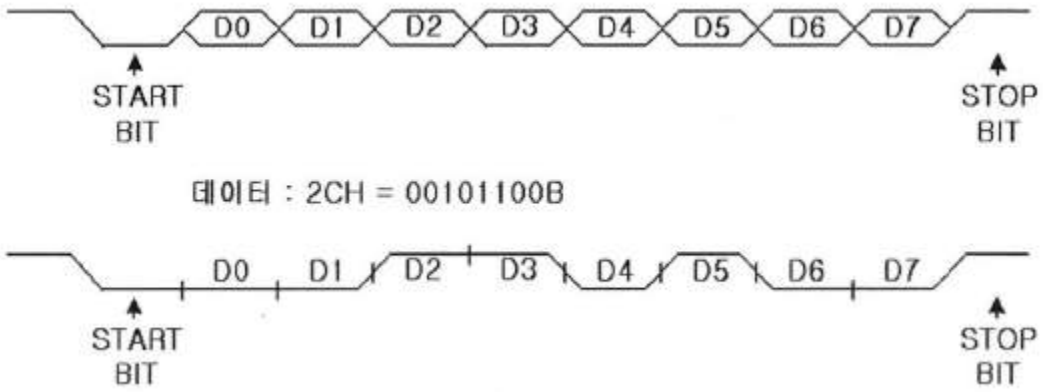
8.1 통신의 개요

8.1.2 통신의 분류

- **UART(Universal Asynchronous Receiver and Transmitter)**
  - 범용 비동기식 시리얼 통신 컨트롤러

8.1 통신의 개요

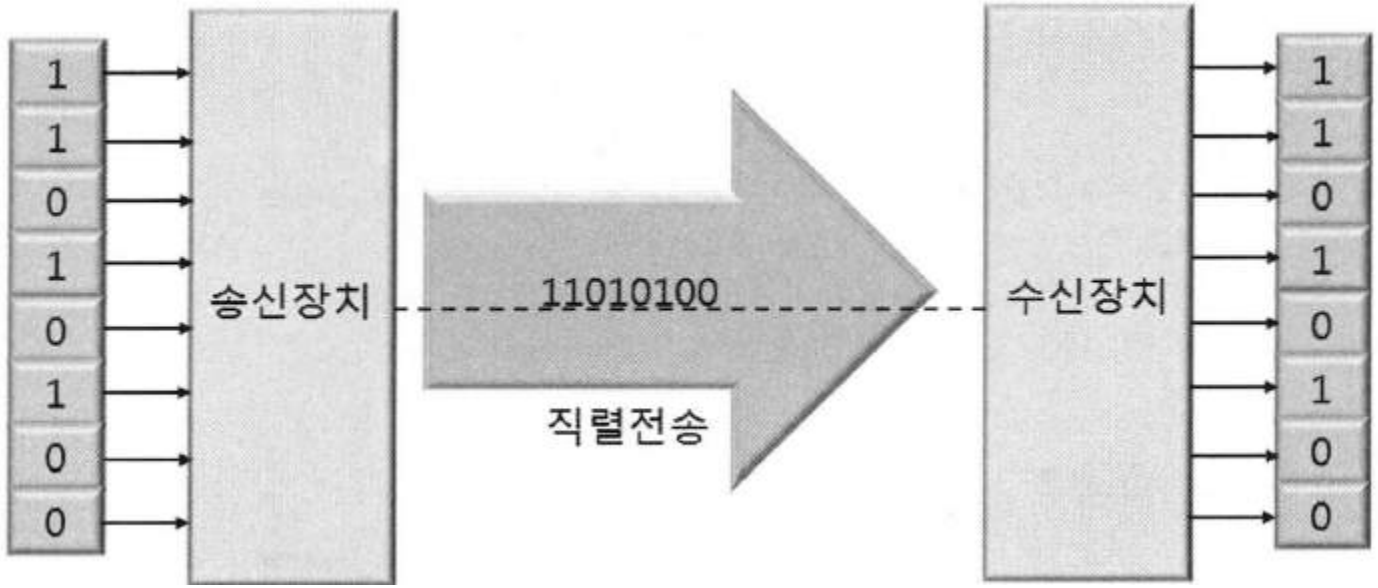
8.1.3 직렬통신과 병렬통신



- 데이터는 LSB부터 MSB 순으로 데이터 전송

8.1 통신의 개요

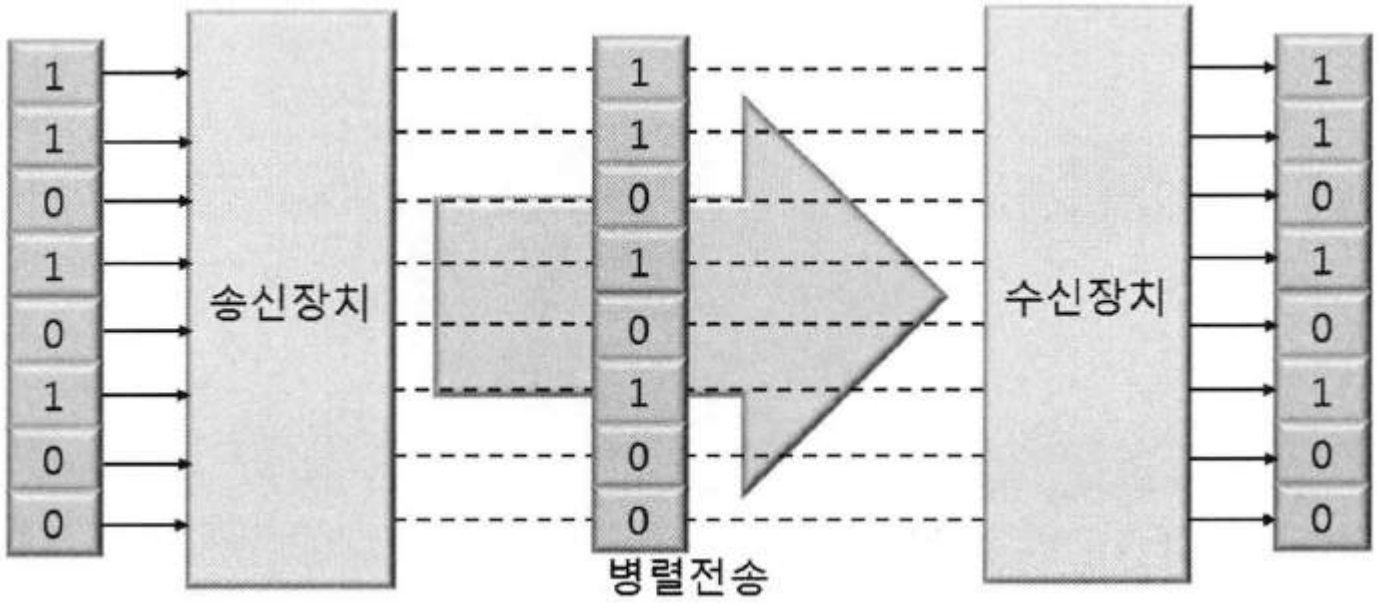
8.1.3 직렬통신과 병렬통신



- **직렬 통신**
  - 컴퓨터 간에 또는 컴퓨터와 주변 장치 간에 한 번에 한 비트(bit)씩 전송하는 통신방식

8.1 통신의 개요

8.1.3 직렬통신과 병렬통신

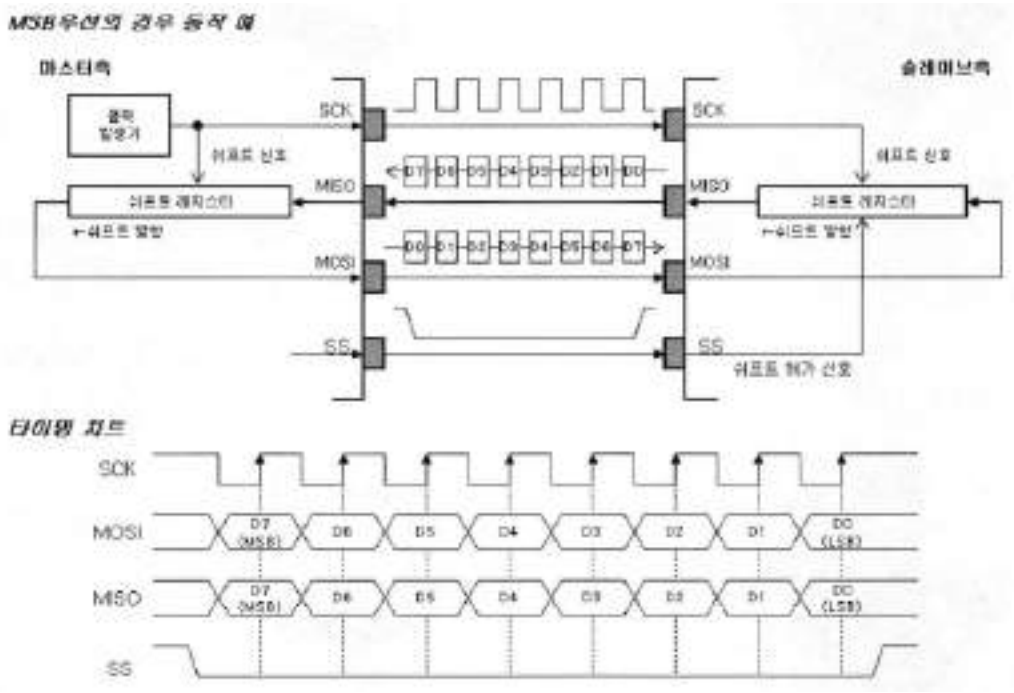


- **병렬 통신**
  - 컴퓨터 간에 보내고자 하는 신호를 몇 개의 선으로 나누어 여러 개의 데이터 비트(data bit)를 동시에 전송하는 방식

8.1 통신의 개요

8.1.4 동기통신과 비동기통신

- **동기(synchronous) 통신**
  - 데이터와는 별도로 송신 측과 수신 즉이 하나의 기준 클럭 (clock)으로 동기신호를 맞추어 동작
  - 이 때, 기준 클럭을 발생시키는 쪽을 마스터라고 하며, 받아서 동작하는 쪽을 슬레이브라고 함
  - 대표적인 동기 통신으로는 12C(혹은 TWI, IIC), SPI 통신 등이 있음
  - 실제 데이터가 전송되기 전에 동기 유지를 위한 문자(동기문자: sync or SCK)를 전송하여 수신 측에서는 이 동기문자를 인식하고 동기를 취함
  - 클럭에 의해 비트를 구별하게 되므로 동기식 전송을 위해서는 데이터와 클럭을 위한 2회선 이상이 필요

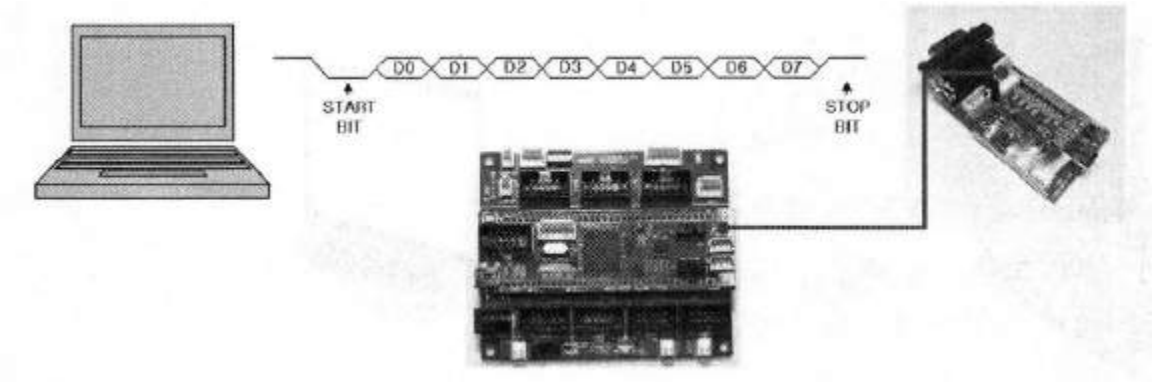


KYUNG HEE UNIVERSITY

8.1 통신의 개요

8.1.4 동기통신과 비동기통신

- **비동기(synchronous) 통신**
  - 송신 컴퓨터는 데이터의 시작을 나타내는 시작 비트(start bit)를 전송하고 데이터 블록이 라는 연속된 비트를 전송함
  - 각 블록의 끝은 하나 이상의 정지 비트(stop bit)를 전송
  - 비동기식 전송은 스타트 비트와 스톱 비트 사이의 간격이 가변적이므로 불규칙적인 전송 에 적합



8.1 통신의 개요

8.1.5 단방향 통신과 양방향 통신

- **단방향 통신(simplex communication)**
  - 오직 한 방향으로만 신호의 전송을 위한 통로가 설정되어 있는 통신 모드로 한 방향으로만 통신이 가능
  - 라디오 송신기, MUX 통신과 PWM 방식

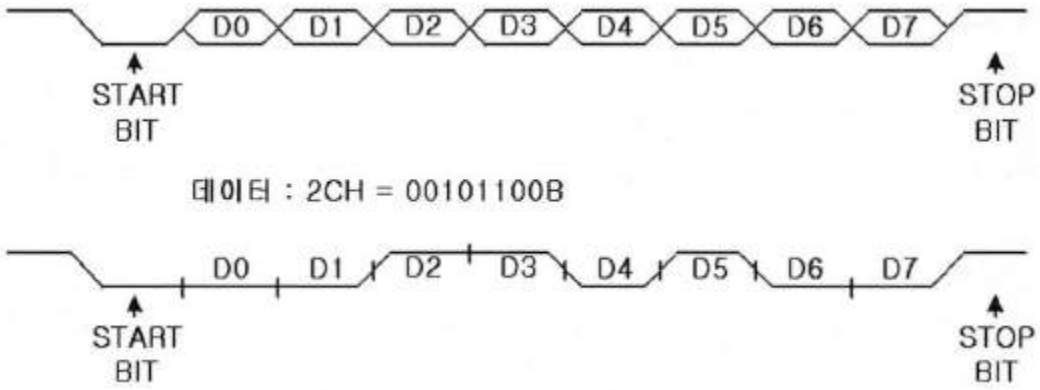
- **반 양방향 통신(Half-duplex communication)**
  - 반이중 모드라고도 하며 전송 통로는 하나이나 방향 선택을 할 수 있는 장치를 부착하여 양방향으로 통신이 가능한 통신 모드로 메시지를 보내고 받을 수 있지만 동시에 할 수는 없음
  - 무전기는 반 양방향 전송의 예, 무전기는 "talk" 버튼을 누르지 않으면 수신 모드에서 대기(CAN 통신)

- **완전 양방향 통신(Full-duplex communication)**
  - 전이중 모드라고도 하며 전송 통로를 통하여 메시지를 동시에 보내고 받는 것이 동시에 가능한 통신 모드
  - 전화는 완전 양방향 통신의 예이고, 대부분의 컴퓨터 통신에서 이용

8.1 통신의 개요

8.1.6 USART 통신

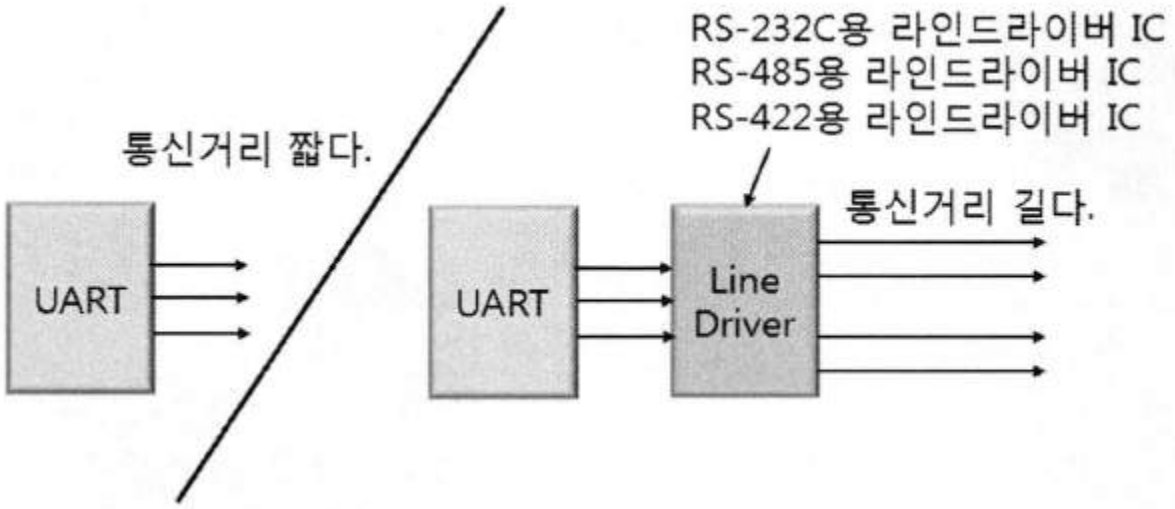
- 비트단위 데이터 전송
  - 송신단 : 1바이트를 8비트로 분리하여 한 번에 1비트씩 통신 선로로 전송
  - 수신단 : 통신 선로를 통해 수신한 비트들을 조립하여 1바이트를 만들어 냄
  - 1바이트 식별을 위해 'Start Bit'와 'Stop Bit'를 사용
  - ERROR 체크를 위해 Parity Bit를 제공하여 에러 검출
  - Parity Bit : 직렬통신시 송수신되는 데이터의 에러 유무를 알려주는 비트(Even parity, Odd parity, Mark & Space Parity)



8.1 통신의 개요

8.1.6 USART 통신

- **Line Transmitter & Receiver**
  - 직렬 출력 신호는 보통 TTL 신호레벨
  - TTL 신호는 노이즈에 약하고 통신거리에 제약
  - TTL 신호를 입력 받아 노이즈에 강하고 멀리 갈 수 있게 해주는 인터페이스 IC인 LINE DRIVER/RECEIVER를 사용
  - 대표적인 Line Driver/Receiver는 RS-232C, RS-422 및 RS-485가 있음



8.1 통신의 개요

8.1.7 RS232C 통신

- EIA(Electronic Industries Association)에 의해 규정
- 데이터단말기(DTE)와 데이터통신기(DCE) 사이의 인터페이스에 대한 규정
- 전기적인 인수, 컨트롤 핸드쉐이킹, 전송속도, 신호 대기시간, 임피던스 인수 등을 정의
- 전송되는 데이터의 포맷과 내용은 지정하지 않음
- DTE간의 인터페이스에 대한 내용도 포함하지 않음
- CCITI(Consultative Committee for International Telegraph and Telephony) 에서도 CCITI V.24에서 DTE와 DCE간의 상호 접속회로의 정의, 핀 번호와 회로의 의미 규정
- GND를 기준점으로 신호선의 크기를 정보로 사용
- 노이즈에 약하여 장거리 전송에 부적합
- 전이중(full duplex, 양방향)방식으로 직렬 접속
- 단지 3 개의 선으로 통신을 하며 Xon/Xoff라 불리는 소프트웨어 적인 방법으로 제어(핸드쉐이크)
- 추가로 선을 사용하여 하드웨어적으로 제어할 수 있음
- Baud Rate: 변조율이나 1초간 통신선의 신호 변경 회수를 가리키는 단어로서 2개의 시리얼 디바이스를 접속한 경우에는 Baud와 BPS는 사실상 똑같음

8.2 PC와 MCU간의 통신제어

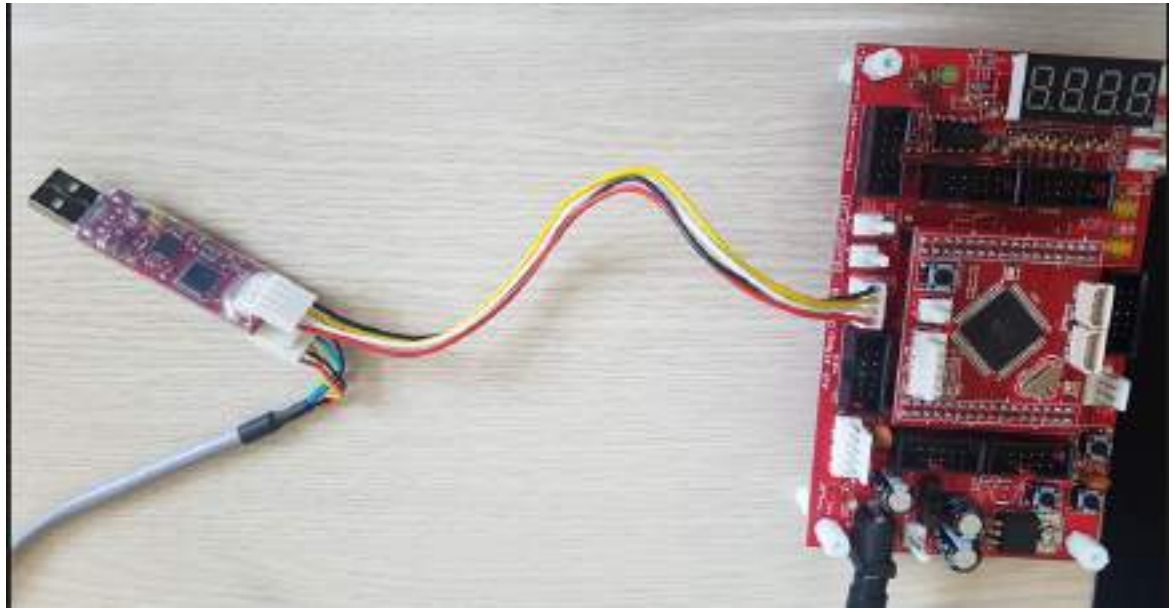
8.2.1 무선통신을 위한 PC와 ATmega128 하드웨어 연결하기

- PC USB 포트와 연결하여 RS232 serial 통신으로 ATmega128의 UART 포트와의 통신제어 시스템을 구성
- PC에서 데이터를 전송(Tx)하면 ATmega128에서 수신(Rx)하여 원하는 제어를 수행
- 또한 ATmega128에서 Tx하면 PC에서 RX하여 모니터 상에 표현할 수 있음
- 실제 모든 데이터를 PC와 ATmega128간에 주고받을 수 있음

8.2 PC와 MCU간의 통신제어

8.2.2 PC와 ATmega128의 포트 연결하기

- **PC와 MCU간의 통신제어를 위한 통신 케이블 연결은 다음과 같이 결선함**
  1) **ISP케이블의 UART 핀과 보드의 UART0를 연결함**
  2) **여기서 주의할 점은 UART 포트 연결 시 Rx와 Tx는 서로 교차되게 연결해야 함 (4pin커넥터를 바로연결해서 사용하면 됨)**



- **USB-UART로 사용하실 경우 타겟보드에 ISP 커넥터와 동시에 UART 커넥터를 연결하시면 안됩니다. 데이터 수신부가 합선되어 문제가 생길 수 있습니다. 꽂는다고 해서 바로 잘 망가지지는 않지만, 심할 경우 망가질 수 있으며, 동시에는 가능한 연결하지 마세요.**

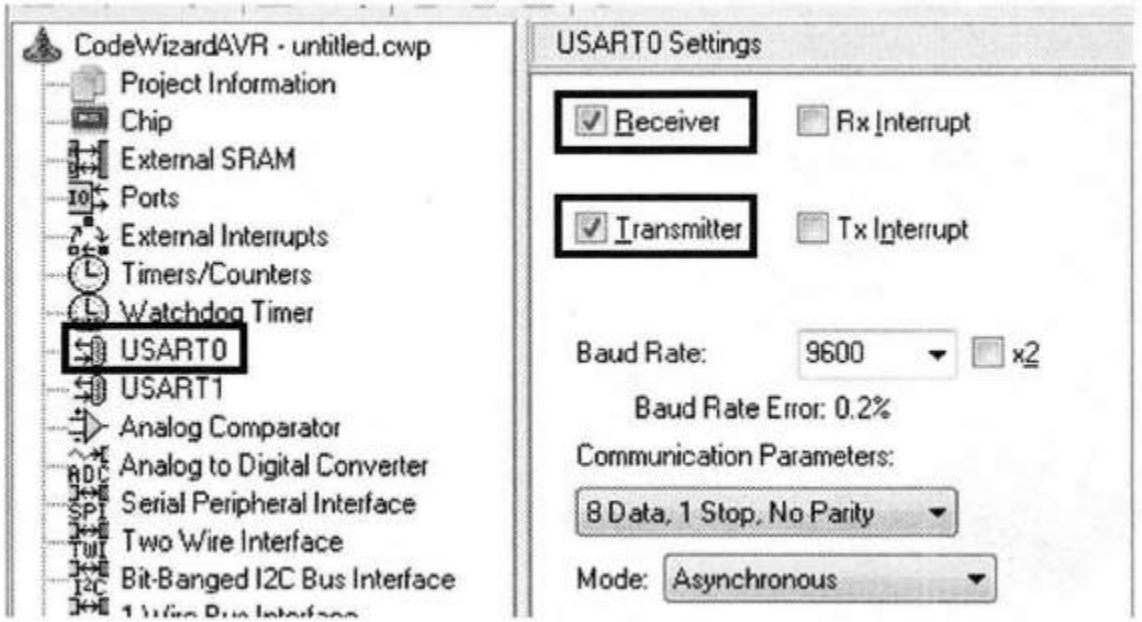
1	VCC
2	RXD (PC가 데이터를 수신하는 핀)
3	TXD (PC가 데이터를 송신하는 핀)
4	GND

* RXD, TXD 신호는 PC를 기준으로 한 것입니다.

8.2 PC와 MCU간의 통신제어

실습1. PC와 MCU간의 통신 제어 예제

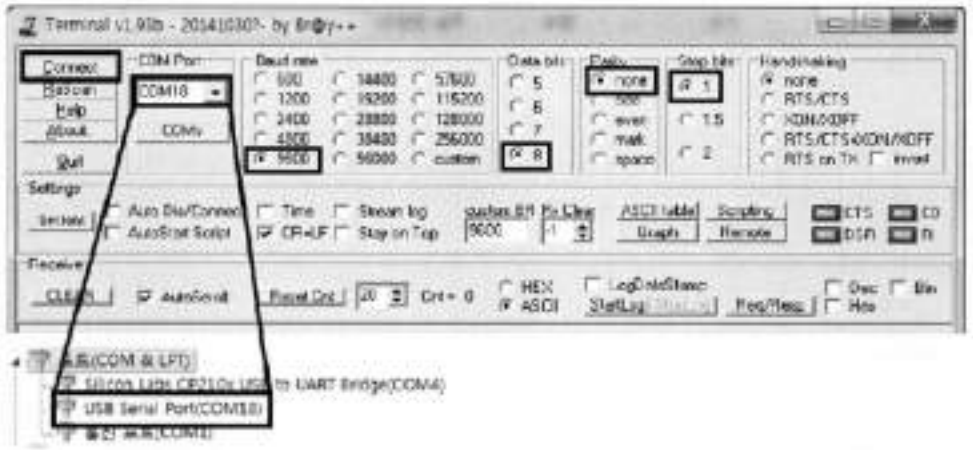
- **ATmega128에 통신을 위한 프로그램을 작성함**
- **우선 CodeWizardAVR에서 아래와 같이 UART0을 Tx, Rx가 될 수 있도록 클릭**
- **C- 언어 코드 결과는 rx_char, tx_char로 나누어 서브루틴 프로그램이 만들어 짐**
- **최종적으로 무한루프 안에서 아래와 같이 키인하여 compile하면 ATmega128에서 결과를 볼 수 있음**



8.2 PC와 MCU간의 통신제어

실습1. PC와 MCU간의 통신 제어 예제

- 컴퓨터 장치관리자에서 USB Serial Port가 COM 몇 번과 연결되어 있는지 확인
- Naver에서 "Terminal V1.93b" 를 검색하여 다운로드 받은 후에 실행
- 실행한 Terminal 프로그램 상에서 COM X로 변경하고 Baud rate, Data bit, parity, Stop mode를 체크
- 마지막으로 Connect 하기



KYUNG HEE UNIVERSITY

8.2 PC와 MCU간의 통신제어

실습1. PC와 MCU간의 통신 제어 예제

- UART0번에 y데이터가 들어오면, "_OK"라는 문자열을 돌려줌
- 8.2.4 PC프로그램 다운로드 및 Setup을 따라해 터미널 프로그램을 설치하여 실행해보기

```
#include <io.h>
#include <mega128.h>
#include <stdio.h>

unsigned char indata;

void putch(char data)
{
    while(!(UCSR0A & 0x20));
    UDR0 = data;
}

char getch(void)
{
    while(!(UCSR0A & 0x80));
    return UDR0;
}
void main(void)
{
    UCSR0A=(0<<RXC0) | (0<<TXC0) | (0<<UDRE0) |
           (0<<FE0) | (0<<DOR0) | (0<<UPE0) |
           (0<<U2X0) | (0<<MPCM0); /* 0x00 */
    UCSR0B=(0<<RXCIE0) | (0<<TXCIE0) | (0<<UDRIE0) |
           (1<<RXEN0) | (1<<TXEN0) | (0<<UCSZ02) |
           (0<<RXB80) | (0<<TXB80); /* 0x18 */
    UCSR0C=(0<<UMSEL0) | (0<<UPM01) | (0<<UPM00) |
           (0<<USBS0) | (1<<UCSZ01) | (1<<UCSZ00) |
           (0<<UCPOL0); /* 0x06 */
    UBRR0H=0x00;
    UBRR0L=0x67;
    while (1)
    {
    // Place your code here
        indata = getch();
        if(indata == 'y')
        {
            putch('_');
            putch('O');
            putch('K');
            putch('\n');
        }
    }
}
```

송신완료 flag 비트가 1이 되면 정지

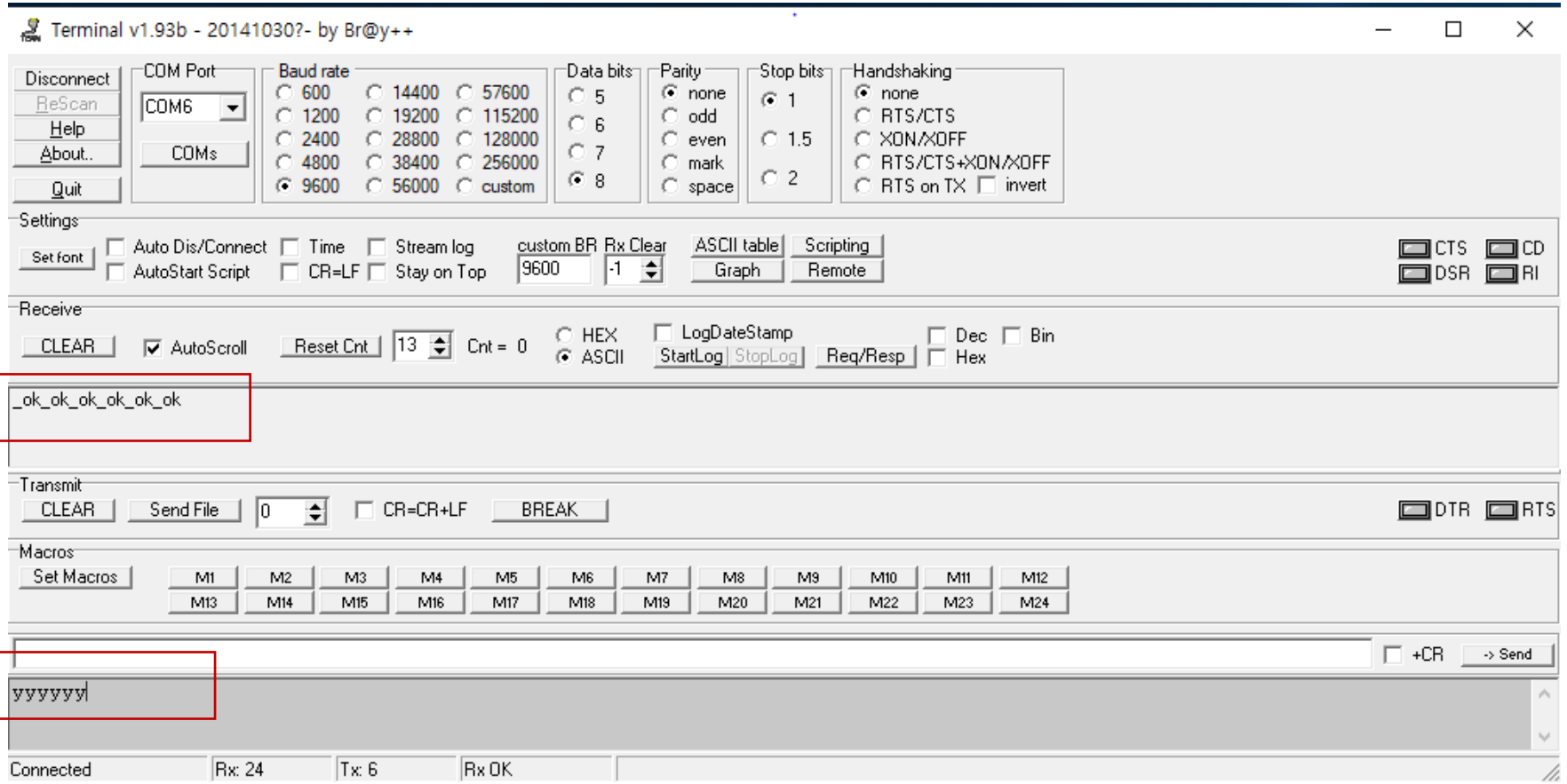
수신완료 flag 비트가 1이 되면 정지

KYUNG HEE UNIVERSITY

8.2 PC와 MCU간의 통신제어

실습1. PC와 MCU간의 통신 제어 예제

- 결과



8.2 PC와 MCU간의 통신제어

실습 2. PC에서 ATmega128에 제어신호 보내기

- 하드웨어 구성은 **ATmega128 PORTC**에 **LED**를 연결하고 **PC**에서 데이터를 **Tx(송신)**할 경우에 **ATmega12801 Rx(수신)**하여 **LED**를 제어할 수 있는 프로그램을 작성하시오.
  - 문자열 "1" 을 받아서 **ATmega128**의 **PORTC**의 1 번 **LED**가 **ON** 될 수 있도록 제어
  - 문자열 " a "를 받아서 **ATmega128**의 **PORTC**의 2 번 **LED**가 **ON** 될 수 있도록 제어

- 실습1에서 쓴 프로그램을 밑의 코드와 같이 약간 수정하여 사용

```
DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) |
     (0<<DDC4) | (0<<DDC3) | (0<<DDC2) |
     (1<<DDC1) | (1<<DDC0); /* 0x03 */

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) |
      (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) |
      (0<<PORTC1) | (0<<PORTC0); /* 0x00 */
 while (1)
 {
 // Place your code here
     indata = getch();
     switch (indata)
     {
         case '1': PORTC = 0x01;
         break;

         case 'a': PORTC = 0x02;
         break;
     }
 }
```

8.2 PC와 MCU간의 통신제어

8.2.5 정보를 받기 위한 서브루틴 함수 getch 이해하기

```
char getch(void)
{
  unsigned char  data;  // 정보를 받는 데이터
  while(! (UCSR0A & 0x80) );
   // 1) RXC0=1 – 수신 버퍼에 읽혀지지 않은 수신문자가 들어 있다.  그러면 while을 빠져 나와서 data변수에
수신 버퍼에 있는 값을 대입한다.
  // 2)RXC0=0  -  수신 버퍼가 비었다. 그러면 while을 계속 돈다.
  data=UDR0;     // data변수에 수신 버퍼에 있는 값을 대입한다.  보내는 순간엔 받을 수 없다.
  UCSR0A |= 0x80;    // UCSR0A 레지스터 중에서 7번째 비트(RXCn)를 세트하여 RXCn를 클리어 함.
  return(data);
}
```

while((UCSR0A & 0x80) == 0);

UXSR0A 레지스터

7	6	5	4	3	2	1	0
RXCn	TXCn	UDREn	FEn	DORn	PEn	U2Xn	MPCMn

*) RXCn(USARTn Receiver Complete) → 수신 버퍼의 상태 플래그,
"1" → 수신 버퍼에 수신문자 있음,
"0" → 수신 버퍼 비어있음.

8.2 PC와 MCU간의 통신제어

8.2.6 정보를 받기 위한 서브루틴 함수 putch 이해하기

```
void putch(unsigned char data)
{
    while(!(UCSR0A & 0x20));
     // 1) UDRE0=1 - 송신버퍼에 새로운 송신데이터 받을 준비 되있다. 넣고 비웠음
                     // 그러면 while을 빠져나와서 송신버퍼에 data값을 대입한다.
     // 2) UDRE0=0 - 송신버퍼에 새로운 송신데이터를 받을 준비가 되지 않았다.
                     // 그러면 while을 계속 돈다.
    UDR0=data;     // 송신버퍼에 데이터변수의 값을 대입한다.
    UCSR0A |= 0x20;  // UCSR0A 레지스터 중에서 5번째 비트(UDREn)를 세트하여 UDRE0를 클리어 함.
```

UXSR0A 레지스터

7	6	5	4	3	2	1	0
RXCn	TXCn	UDREn	FEn	DORn	PEn	U2Xn	MPCMn

*) UDREn(USARTn Data Register Empty) → 새로운 송신 데이터를 받기 위한 상태 플래그, UDRn의 송신 버퍼에 새로운 송신 데이터를 받을 준비가 되어있으면 "1"로 세트

8.2 PC와 MCU간의 통신제어

실습 3. ATmega128에서 PC로 신호 보내기

- 하드웨어 구성은 ATmega128의 Tx에서 데이터가 전송되면 PC에서 Rx로 데이터를 받아서 출력하는 프로그램을 작성하시오.
  - PORTC의 푸쉬버튼 I 번을 On하면 숫자 "1"을 PC로 보낸다.
  - PORTC의 푸쉬버튼 2 번을 On하면 숫자 "a " 를 PC로 보낸다.
  - PORTC의 푸쉬버튼 3 번을 On하면 숫자 "Hello World !"를 PC로 보낸다.

- 8.23 ATmega128의 프로그램 작성하기에서 쓴 프로그램을 밑의 코드와 같이 약간 수정하여 사용한다.

```
#include <delay.h>
DDRC=(0<<DDC7) | (0<<DDC6) | (0<<DDC5) |
     (0<<DDC4) | (0<<DDC3) | (0<<DDC2) |
     (0<<DDC1) | (0<<DDC0); /* 0x00 */

PORTC=(0<<PORTC7) | (0<<PORTC6) | (0<<PORTC5) |
      (0<<PORTC4) | (0<<PORTC3) | (0<<PORTC2) |
      (0<<PORTC1) | (0<<PORTC0); /* 0x00 */

while (1)
{
// Place your code here
    unsigned char text[16] = "Hello World! \r\n";
    unsigned char i=0;
    indata = PINC;
    switch(indata)
    {
        case 0x01: putch('1');
        break;
        case 0x02: putch('a');
        break;

        case 0x04:
            while(text[i] != '\0'){
                putch(text[i++]);
            };
        break;
    }
    delay_ms(100);
}
```

8.2 PC와 MCU간의 통신제어

실습 4. ATmega128과 PC로 통신신호 주고받기

- ATmega128과 PC간의 통신제어 신호 데이터를 주고받는 프로그램을 작성하시오.
  - Master(PC)에서 Keyboard를 통하여 데이터를 Tx(Transmitter) 한다.
  - Slave 인 ATmega128 에서 데이터를 받아서 LCD 액정에 디스플레이한다.

```
28  char getch(void)
29  {
30      while(!(UCSR0A & 0x80));
31      return UDR0;
32  }
33
34  void putch(char data)
35  {
36      while(!(UCSR1A & 0x20));
37      UDR1 = data;
38  }
40  void main(void)
41  {
42      //USART0 initialization
43      UCSR0A=0x00;
44      UCSR0B=0x10;
45      UCSR0C=0x06;
```

```
46      UBRR0H=0x00;
47      UBRR0L=0x67;
48
49      //USART1 initialization
50      UCSR1A=0x00;
51      UCSR1B=0x08;
52      UCSR1C=0x06;
53      UBRR1H=0x00;
54      UBRR1L=0x67;
55
56      while (1)
57      {
58          putch(getch());
59      }
60  }
```

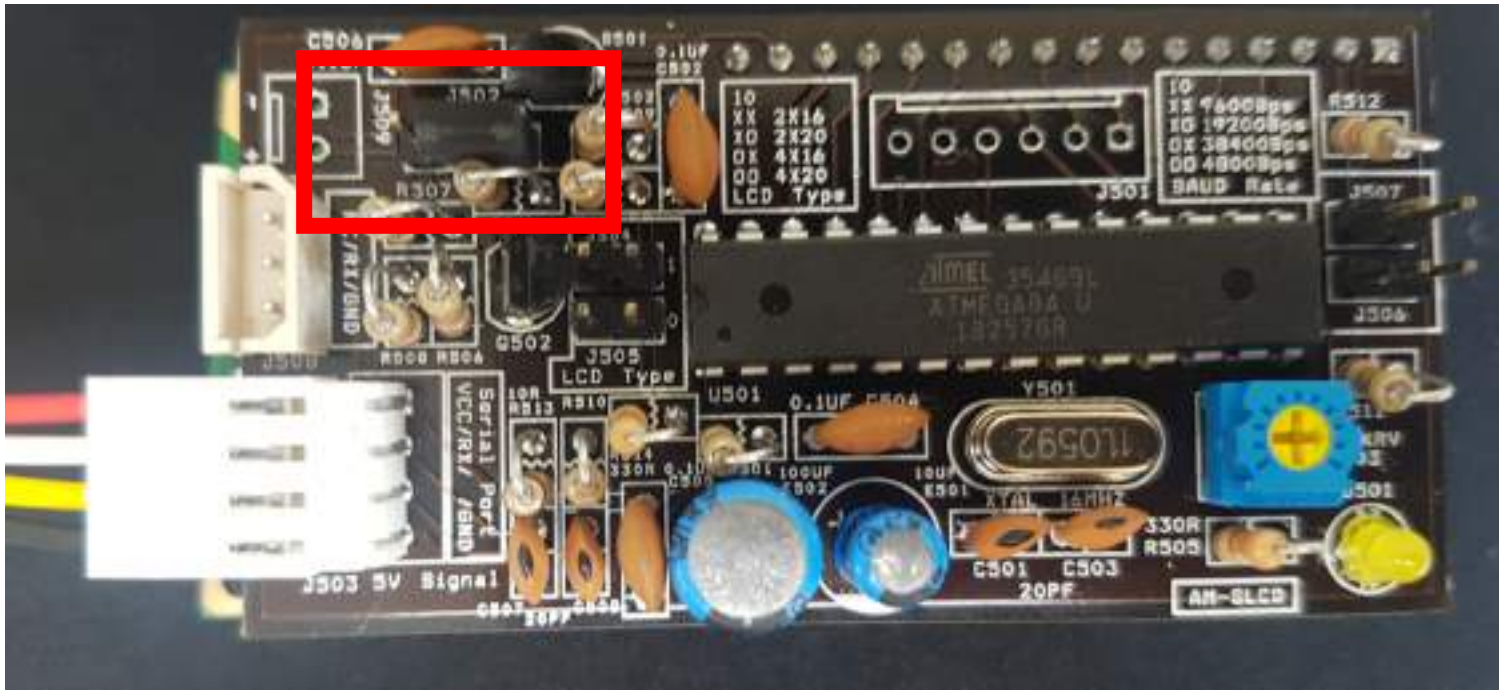
- UART0는 마스터 인 PC에 연결하고, UART1은 LCD에 연결한다.
- LCD는 모드를 커맨드 모드가 아닌 터미널 모드로 변경하여 사용하면, 시리얼 데이터를 그대로 받아 표시하게 된다.

8.2 PC와 MCU간의 통신제어

실습 4. ATmega128과 PC로 통신신호 주고받기

- **UART0는 마스터 인 PC에 연결하고, UART1은 LCD에 연결한다.**
- **LCD는 모드를 커맨드 모드가 아닌 터미널 모드로 변경하여 사용하면, 시리얼 데이터를 그대로 받아 표시하게 된다.**

CM: 커맨드 모드, TM : 터미널 모드



KYUNG HEE UNIVERSITY

8.2 PC와 MCU간의 통신제어

[과제 10] 전송받은 알파벳이 소문자면 대문자로, 대문자면 소문자로 변환

- 터미널 창에 전송받은 알파벳이 소문자면 대문자로, 대문자면 소문자로 변환
- 알파벳이 아닌 경우에는 “?”를 출력하는 프로그램을 작성해 보시오.

Thank you

DONHAN KIM

HRI Lab

Human-Robot Interaction Laboratory

KYUNG HEE UNIVERSITY