2강. USART

(Universal Sync/Asynchronous Receiver Transmitter)

박 원 엽
010.5451.0113

목 차

2강 USART (Universal Sync/Asynchronous Receiver Transmitter)

- **USART(Universal Sync/Asynchronous Receiver Transmitter)란?**

  - 입출력 기기 혹은 데이터 회선과 처리 장치와의 사이에 있으며, 전2중 및 비트 직렬 방식으로 데이터 전송을 다루는 기능 단위. 보통 단일의 패키지로서 구성되는 LSI 회로이며, 타이프라이터와 같은 저속도의 입출력 기기에는 비동기식으로, CRT 단말과 같은 고속도의 입출력 기기에 대해서는 동기식으로 데이터의 송수신 제어나 그에 부수하는 기능을 수행한다.

  출처: [네이버 지식백과] 범용 동기/비동기형 송수신기 [USART] (전자용어사전, 1995.3.1, 성안당)

  대부분의 USART는 비동기식으로 쓰이며, UART라고 함.

- **USART(Universal Sync/Asynchronous Receiver Transmitter)란?**

  - 동기식 통신이란? 통신을 할 때 데이터 전송라인과 더불어 클럭전송라인을 이용하여 이 클럭에 동기화 되어 데이터의 전송이 이루어지는 방식.
  - 비동기식 통신이란? 데이터 전송라인만을 이용하여 데이터를 전송하는 방식. 이때 수신부, 송신부에 전송률(Baud rate[bps])을 맞춰주어야 하며 이를 맞춰주지 않으면 데이터 전송 시, 오류가 발생한다.

  - 직렬 통신이란? 데이터 전송 시, 1개의 전송라인으로 데이터를 전송하는 방식.
  - 병렬 통신이란? 데이터 전송 시, 여러 개의 전송라인(흔히 버스라는 표현을 씀)으로 데이터를 전송하는 방식

- **RS-232, CMOS, TTL레벨이란?**

  - TTL레벨: 0V~5V 의 전압레벨. MCU간 통신, 단거리 칩간 통신시 사용됨.

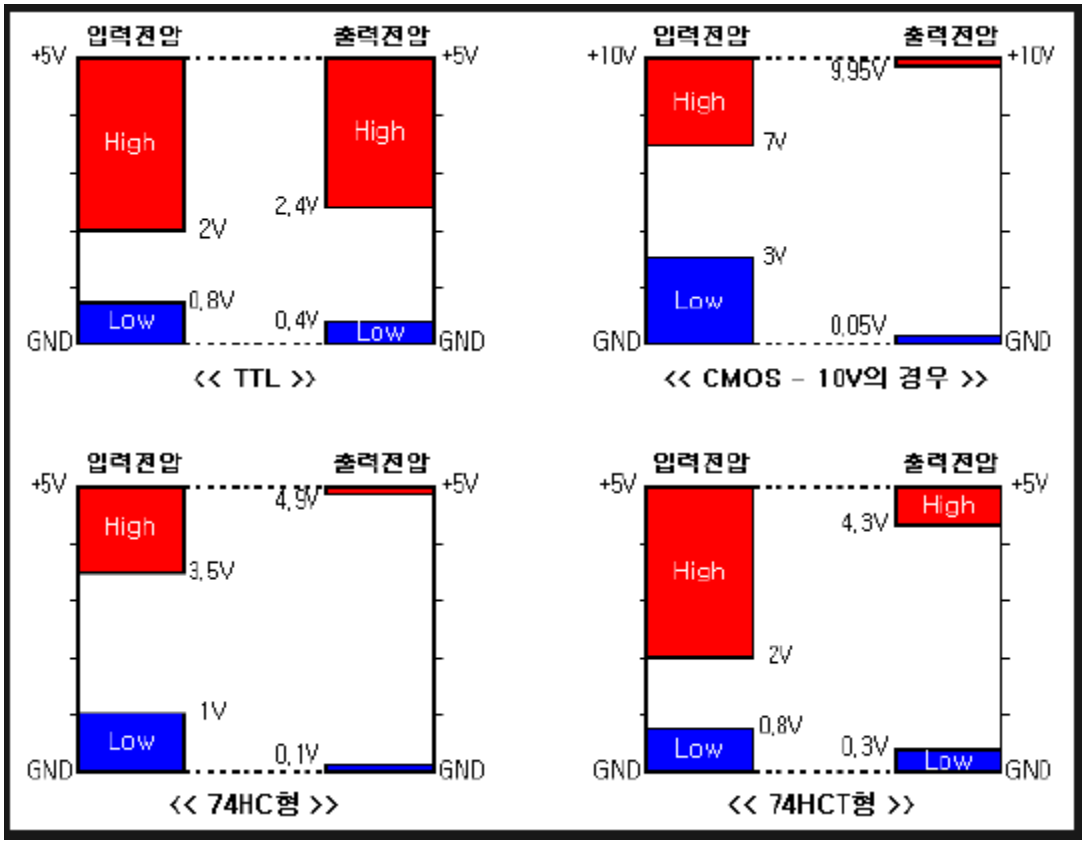
  - CMOS레벨: 0V~3.3V 혹은 0V~10V이상의 전압레벨.

http://blog.naver.com/yashinzone/40037499075 참고.

항목	TTL	CMOS
전원전압	4.75 ~ 5.25V	종래형 : 3 ~ 18V 고속형 : 2 ~ 6V
Threshold Level	1.2 ~ 1.4V	전원전압의 약 1/2
입출력간 전달지연시간	LS형 -- 10ns AS, AL형 -- 3 ~ 3.5ns	종래형 : 100ns 고속형 : 8ns
최고 응답 주파수	LS형 -- 45MHz ALS형 -- 100MHz	종래형 : 2MHz 고속형 : 45MHz
소비전류	LS형 -- 3.2mA(H레벨출력) 1.6mA(50%듀티)	0.0005 ~ 0.0003 micro A
품종	매우 풍부(500종 이상)	실용상 충분한 품종
사용온도 범위	섭씨 0 ~ 70도	섭씨 - 40 ~ 85도
장점	전달 지연 시간이 짧다	구조가 간단하여 집적화가 쉽다
단점	노이즈 마진이 작다 선로 임피던스에 영향받기 쉽다 소비전력이 크다	정전 파괴가 쉽다 고온에 약하다

출처: http://ecima.blog.me/80014445817

- **노이즈 마진이란?**



- **RS-232, CMOS, TTL레벨이란?**
  - RS-232레벨: -12V~+12V의 전압레벨을 이용한 통신 방식. 컴퓨터가 외부와 자료를 주고 받기 위하여 국제적으로 표준화한 데이터 통신규격의 하나이다. 데이터를 직렬 전송 방식으로 전송할 때 통신회선에서 사용하는 전기적인 신호의 특성과 연결장치의 형상 등 물리적인 규격을 정하고 있다.
  이것은 20K baud 이하의 속도로 15m 이내에서 직렬로 자료를 전송할 수 있다. 직렬 전송 방식으로 데이터 통신을 하는데 필요한 또 다른 규격으로는 1976년에 제정한 RS-422와 RS-485가 있다.
  출처: [네이버 지식백과] RS-232C [recommended standard-232C] (두산백과)

- **RS-232, CMOS, TTL레벨이란?**
  - RS-422레벨: 컴퓨터와 주변 장치 또는 데이터 단말 장치(DTE)와 데이터 회선 종단 장치(DCE)를 상호 접속하기 위한 인터페이스 표준. 미국전자공업협회(EIA) 권고 표준의 하나이다. RS-422는 평형 직렬 인터페이스의 표준 규격으로, 이 인터페이스에 사용되는 전기적, 기계적 특성을 규정하고 있다. ITU-T 권고 V.11과 호환성이 있다. RS-422는 케이블 접속기의 규격을 규정하고 있지 않기 때문에 이 표준을 적용하는 제조업체는 다양한 종류의 비표준 핀 구성 접속기를 사용하고 있다. 예를 들면, 매킨토시의 직렬 접속구는 RS-422 접속구이다. 데이터의 최대 전송 속도는 10Mbps이며, DTE와 DCE 간의 거리 1.2km에서 접속이 가능하다.
  출처: [네이버 지식백과] RS-422 [Recommended Standard-422] (IT용어사전, 한국정보통신기술협회)

- **RS-232, CMOS, TTL레벨이란?**
  - RS-485레벨: RS 232, RS 422의 확장 버전으로, 홈 네트워크를 지원하는 일종의 시리얼 통신 프로토콜 표준 규격. 전송 속도가 늦고 전송 거리가 짧은 RS 232를 보완하기 위해 RS 422 통신 방식을 채택하였다. RS 485는 모든 장치들이 같은 라인에서 데이터 전송 및 수신을 할 수 있다. 반이중 방식과 전 이중 통신 방식을 모두 지원한다. 또한 RS 485는 최대 드라이버·리시버 수가 각각 32개에 이르고, 최대 속도 10Mbps에 최장 거리 1.2km까지 네트워크 구축이 가능하다.
  출처: [네이버 지식백과] RS485 (IT용어사전, 한국정보통신기술협회)

2강 USART (Universal Sync/Asynchronous Receiver Transmitter)

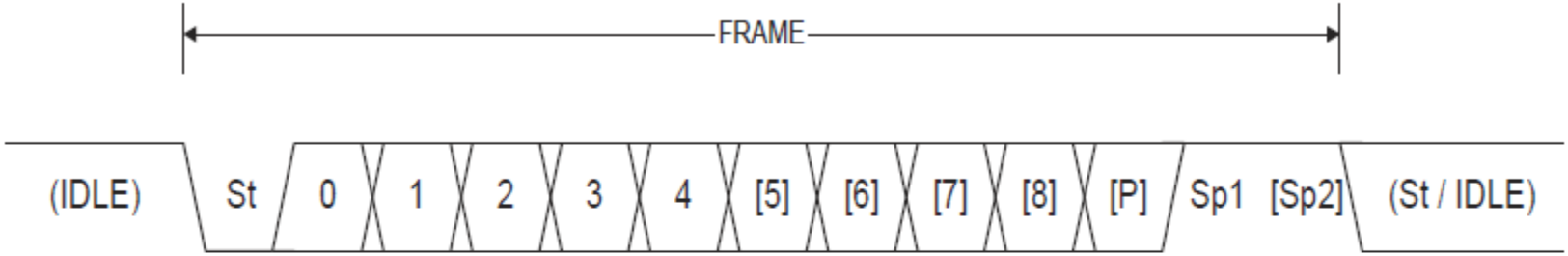
- **RS232C, RS422, RS423, RS485 비교 표**

Specification	RS232C	RS423	RS422	RS485
동작 모드	Single-Ended	Single-Ended	Differential	Differential
최대 Driver/ Receiver 수	1 Driver 1 Receiver	1 Driver 10 Receivers	1 Driver 32 Receivers/256	32 Drivers 32 Receivers/256
최대 통달거리	약 15 m	약 1.2 km	약 1.2 km	약 1.2 km
최고 통신속도	20 Kb/s	100 Kb/s	10 Mb/s	10 Mb/s
지원 전송방식	Full Duplex	Full Duplex	Full Duplex	Half Duplex
최대 출력전압	±25V	±6V	−0.25V to +6V	−7V to +12V
최대 입력전압	±15V	±12V	−7V to +7V	−7V to +12V

2강 USART (Universal Sync/Asynchronous Receiver Transmitter)

- 데이터시트 176페이지. UART 데이터 프레임 포맷

Figure 82. Frame Formats



St	Start bit, always low.
(n)	Data bits (0 to 8). 보통 8비트
P	Parity bit. Can be odd or even. 보통 사용하지 않음
Sp	Stop bit, always high. 보통 1 스탑비트
IDLE	No transfers on the communication line (RxD or TxD). An IDLE line must be high.

2강 USART (Universal Sync/Asynchronous Receiver Transmitter)

- 데이터시트 189페이지

USARTn I/O Data Register – UDRn

Bit	7	6	5	4	3	2	1	0	
	RXBn[7:0]								UDRn (Read)
	TXBn[7:0]								UDRn (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

UDRn (n=0~1) 레지스터는 UART Data Register로써, UART 송수신시 실제 데이터가 저장되는 레지스터.
n은 채널을 의미하며 ATmega128에는 0, 1 두개의 채널이 있음.
송수신 모두 UDRn을 사용하며, 8비트로 구성됨. 읽기/쓰기 가능

2강 USART (Universal Sync/Asynchronous Receiver Transmitter)

- 데이터시트 189페이지

USART Control and Status Register A – UCSRnA

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREn	FEn	DORn	UPEn	U2Xn	MPCMn	UCSRnA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

USARTn Control and Status Register B – UCSRnB

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	UCSZn2	RXB8n	TXB8n	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

USART Control and Status Register C – UCSRnC

Bit	7	6	5	4	3	2	1	0	
	–	UMSELn	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

ATmega128에서 UART 기능을 사용하기 위해서는 위의 3가지 레지스터들을 설정해주어야 한다.
n은 역시 채널을 의미(앞으로 나오는 모든 레지스터중 n은 채널 번호를 의미)

- **UCSRnA: USART Control and Status Register A, n=0~1**

  - Bit7 – 수신완료 플래그 비트
  - Bit6 – 송신완료 플래그 비트
  - Bit5 – 송신시, 버퍼(UDRn) 레지스터가 비어있는지 나타내는 플래그 비트
  - Bit4 – 프레임 에러(Frame Error) 플래그 비트
  - Bit3 – 데이터 오버런(Data OverRun) 플래그 비트
  - Bit2 – 패리티 에러(Parity Error) 플래그 비트
  - Bit1 – 더블 송신 스피드 설정 비트
  - Bit0 – 멀티 프로세서 통신 모드 설정 비트

- **UCSRnB: USART Control and Status Register B, n=0~1**
  - Bit7 – 수신 인터럽트 설정 비트
  - Bit6 – 송신 인터럽트 설정 비트
  - Bit5 – 데이터 레지스터(UDRn) Empty 인터럽트 설정 비트
  - Bit4 – 수신 Enable 설정 비트
  - Bit3 – 송신 Enable 설정 비트
  - Bit2 – 데이터 크기 설정 비트
  - Bit1 – 수신 데이터의 8번 데이터 비트(데이터 크기가 9일때)
  - Bit0 – 송신 데이터의 8번 데이터 비트(데이터 크기가 9일때)

- **UCSRnC: USART Control and Status Register C, n=0~1**
  - Bit7 – Reserved
  - Bit6 – USART 모드 선택 비트
  - Bit5 – 패리티 모드 설정 비트 1
  - Bit4 – 패리티 모드 설정 비트 0
  - Bit3 – 스탑 비트 선택 비트
  - Bit2 – 데이터 사이즈 설정 비트 1
  - Bit1 – 데이터 사이즈 설정 비트 0
  - Bit0 – 클럭 극성 설정 비트(동기식 통신일때만 적용)

2강 USART (Universal Sync/Asynchronous Receiver Transmitter)

- 데이터시트 192페이지

USART Baud Rate Registers – UBRRnL and UBRRnH

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	UBRRn[11:8]				UBRRnH
	UBRRn[7:0]								UBRRnL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

비동기식 통신을 위해서 통신속도(Baud Rate, 보레이트)를 설정해주어야 함. 흔히 9600bps, 57600bps, 115200bps등의 숫자로 정의되며 1초에 몇 비트의 데이터를 전송하는지를 나타내는 수치. 당연히 숫자가 높을수록 통신속도가 빨라짐.

비동기 통신을 위해서 송수신 양단의 통신속도를 일치하게 설정해주어야 함. 왜 그런지 이유에 대해 생각해보자.

- **UBRRnH: USART Baud Rate Register, n=0~1**
  - Bit7 – x
  - Bit6 – x
  - Bit5 – x
  - Bit4 – x
  - Bit3 – UBRR 11번 비트
  - Bit2 – UBRR 10번 비트
  - Bit1 – UBRR 9번 비트
  - Bit0 – UBRR 8번 비트
- **UBRRnL: USART Baud Rate Register, n=0~1**
  - Bit7 – UBRR 7번 비트
  - Bit6 – UBRR 6번 비트
  - Bit5 – UBRR 5번 비트
  - Bit4 – UBRR 4번 비트
  - Bit3 – UBRR 3번 비트
  - Bit2 – UBRR 2번 비트
  - Bit1 – UBRR 1번 비트
  - Bit0 – UBRR 0번 비트

2강 USART (Universal Sync/Asynchronous Receiver Transmitter)

- 데이터시트 174페이지

Table 74. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{OSC}}{16(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{OSC}}{8(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

; 통신속도와 UBRR 레지스터와의 관계식

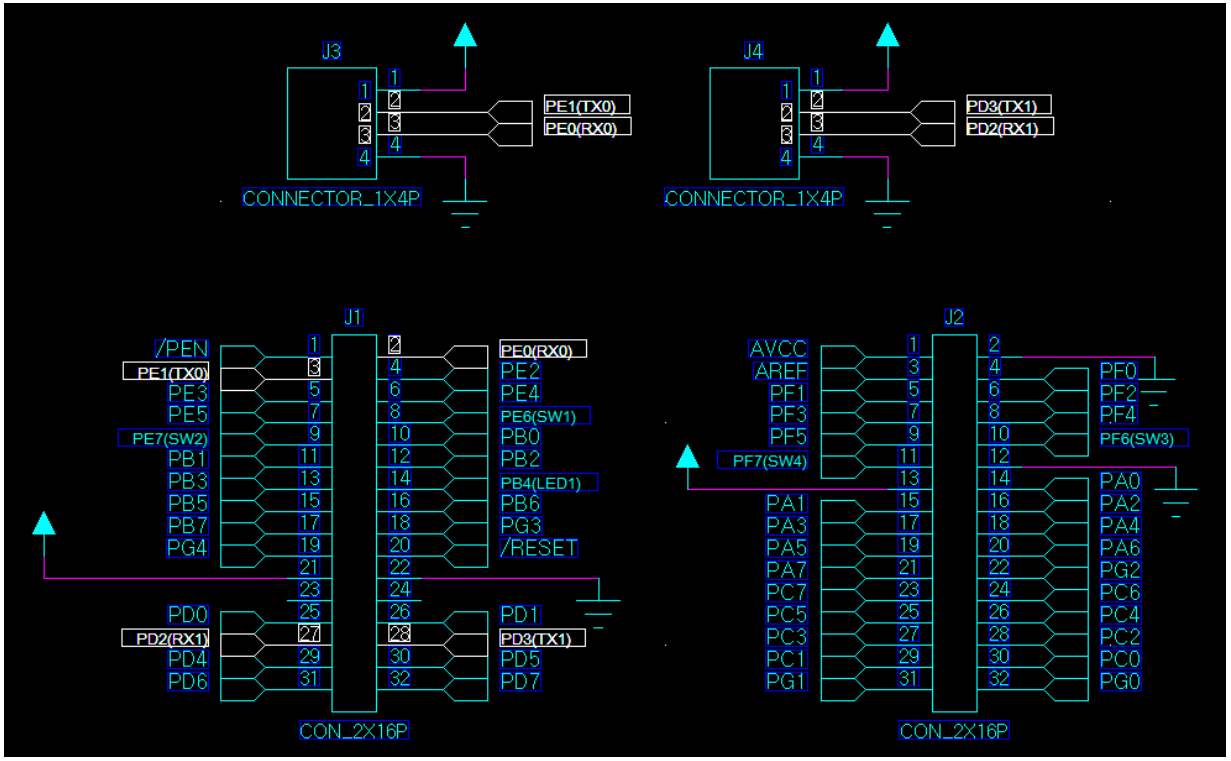
- 데이터시트 197페이지

Table 85. Examples of UBRR Settings for Commonly Used Oscillator Frequencies

Baud Rate (bps)	f_{osc} = 16.0000 MHz			
	U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error
2400	416	-0.1%	832	0.0%
4800	207	0.2%	416	-0.1%
9600	103	0.2%	207	0.2%
14.4k	68	0.6%	138	-0.1%
19.2k	51	0.2%	103	0.2%
28.8k	34	-0.8%	68	0.6%
38.4k	25	0.2%	51	0.2%
57.6k	16	2.1%	34	-0.8%
76.8k	12	0.2%	25	0.2%
115.2k	8	-3.5%	16	2.1%
230.4k	3	8.5%	8	-3.5%
250k	3	0.0%	7	0.0%
0.5M	1	0.0%	3	0.0%
1M	0	0.0%	1	0.0%
Max[1]	1 Mbps		2 Mbps	

; 통신속도와 UBRR 과의 관계를 나타낸 표

USART 연결 회로



USART 연결 회로는 같은 레벨로 통신할 경우 부가적인 회로
(저항, 커패시터, 레벨쉬프터IC 등)가 필요없다.
다른 레벨로 통신할 경우 추가 회로가 반드시 필요하다.
PC와 통신할 경우, USB to UART(IC)을 이용하여 통신하는 것이 가능.

- **USB to UART이란?**
  - 과거 PC는 외부 장치와의 직렬통신을 위해 9 pin D-sub 단자를 사용했다. 하지만 USB가 표준으로 자리잡으면서 USB를 이용한 직렬통신 방식이 이를 대체하게 되었다.
  USB는 D+, D-의 통신라인을 이용한 비동기/동기식 통신 방식이며 이를 UART(TTL)로 변환해주어야 USB로 PC와 Atmega간 통신이 가능해진다.
  이렇게 변환해주는 장치로는 PL2303, CP210x 등의 칩이 있으며 드라이버를 설치해주면 가상의 COM포트로 인식되어 Atmega와 통신이 가능해진다.

2강 USART (Universal Sync/Asynchronous Receiver Transmitter)

다음과 같은 코드를 작성(인터럽트x)

```
1  #include <mega128.h>
2  #include <delay.h>
3
4  main()
5  {
6        DDRB = 0x10; //PB4에 LED 연결
7        PORTB = 0x00; //B포트 출력 초기화
8        DDRE = 0x00; //PE6, PE7에 스위치 연결
9
10       /* UART 관련 레지스터 설정 */
11       UCSR1A=0x00;
12       UCSR1B=0x18;
13       UCSR1C=0x06;
14       UBRR1H=0x00;
15       UBRR1L=0x08;
16       /*****************************/
17
18       while(1)
19       {
20       }
21 }
```

위의 레지스터 설정의 의미를 한번 생각해보자.
위와 같이 설정하면 UART1의 Tx, Rx를 사용할 수 있게 된다!

2강 USART (Universal Sync/Asynchronous Receiver Transmitter)

송수신 함수 코드 작성(인터럽트x)

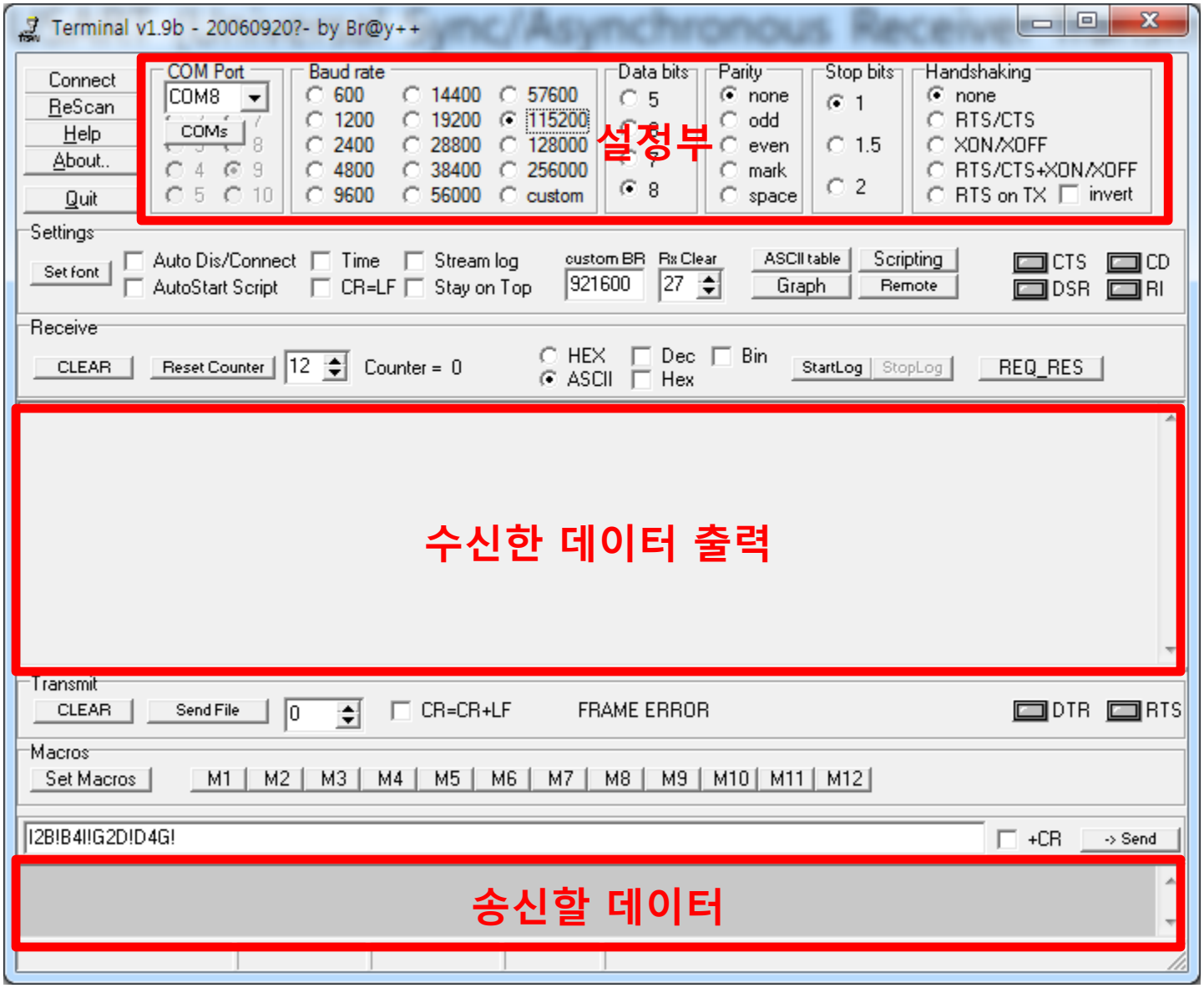
```
4  #define RXB8 1
5  #define TXB8 0
6  #define UPE 2
7  #define OVR 3
8  #define FE 4
9  #define UDRE 5
10 #define RXC 7
11
12 #define FRAMING_ERROR (1<<FE)
13 #define PARITY_ERROR (1<<UPE)
14 #define DATA_OVERRUN (1<<OVR)
15 #define DATA_REGISTER_EMPTY (1<<UDRE)
16 #define RX_COMPLETE (1<<RXC)
17
18 char getchar1(void)
19 {
20      char status, data;
21      while ((UCSR1A) & RX_COMPLETE)==0);
22      data=UDR1;
23
24      return data;
25 }
26
27 void putchar1(char c)
28 {
29      while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
30      UDR1=c;
31 }
```

송신함수 void putchar1(char c) 와 수신함수 char getchar1(void)의 작성 예

- **PC와 통신하기**
  - ATmega128의 모든 코드를 작성하였다면 PC와 통신해본다. PC와 통신하기 위해서는 USB to UART 모듈을 PC에 꽂은 후 드라이버를 설치한다.
  그 후, terminal1.9b.exe를 실행하여 Atmega와 설정을 맞춰준다.

2강 USART (Universal Sync/Asynchronous Receiver Transmitter)

ChrisP의 M World
blog.naver.com/lbiith



2강 USART (Universal Sync/Asynchronous Receiver Transmitter)

모든 설정을 맞춰준 후 PC와의 데이터 통신을 실습해본다.

문1) PC에서 1개의 아무 데이터를 입력받으면 LED를 켜고 끄는 프로그램 작성

문2) PC에서 키보드의 'I'를 입력받으면 LED를 켜고, 'O'를 입력받으면 LED를 끄는 프로그램 작성

문3) PC에서 입력받은 데이터를 다시 PC로 송신하는 프로그램 작성

2강 USART (Universal Sync/Asynchronous Receiver Transmitter)

ATmega128간의 데이터 통신을 실습해본다. (인터럽트에서만 가능)

문1) 1번 스위치를 누르면 'I' 송신, 2번 스위치를 누르면 'O' 송신.
'I'를 수신받으면 LED를 켜고, 'O'를 수신받으면 끄는 프로그램 작성