전자회로실험2

Lab2.
USART

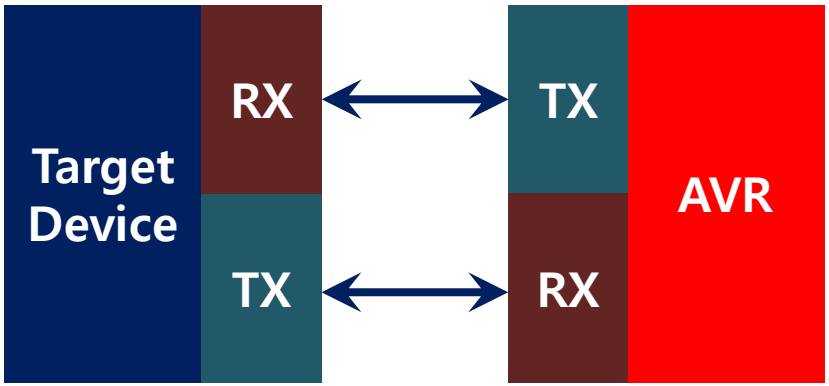
USART

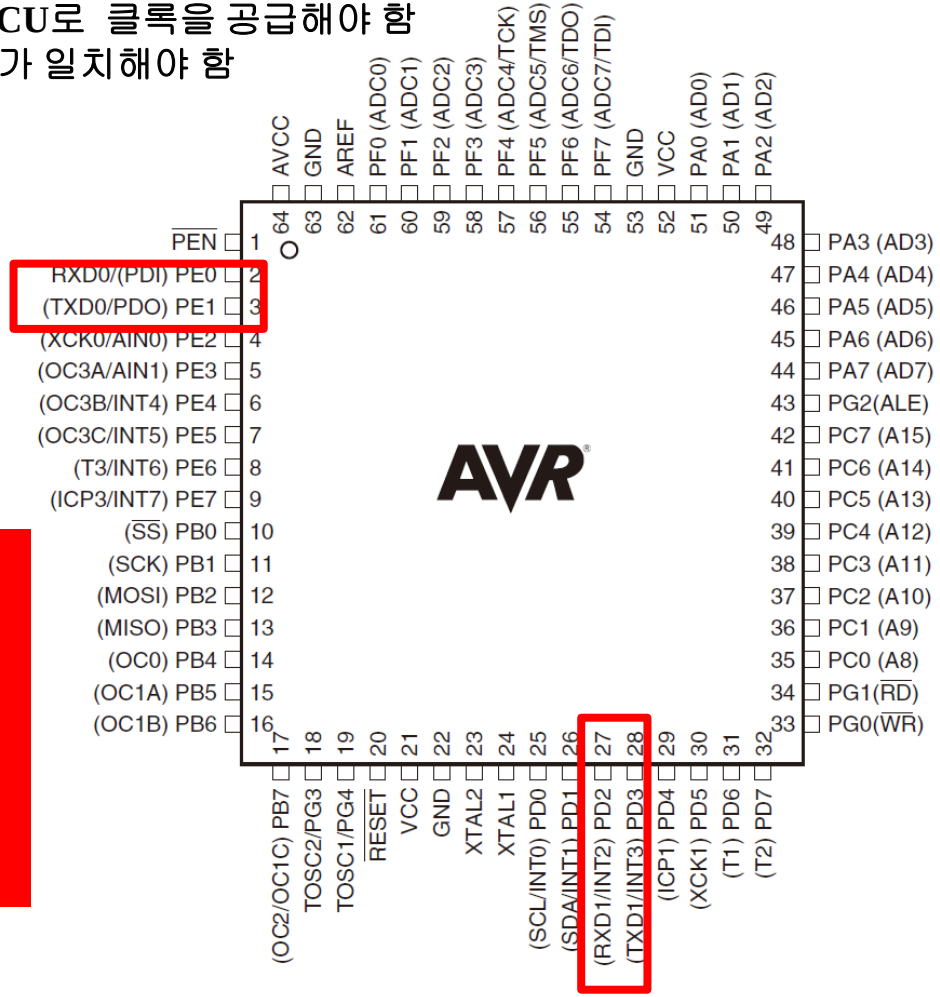
Universal Synchronous and Asynchronous serial Receiver and Transmitter

- 범용 동기/비동기 수신기 및 송신기

- **SPI통신과 같이 마스터 MCU가 슬레이브 MCU로 클록을 공급해야 함**
- 올바른 통신을 위해 프레임 구조와 전송속도가 일치해야 함

- **Full Duplex**
- **Asynchronous or Synchronous**
- **5~9 Data Bits, 1~2 Stop Bits**
- **Odd or Even Parity**
- **Baud Rate (bps)**

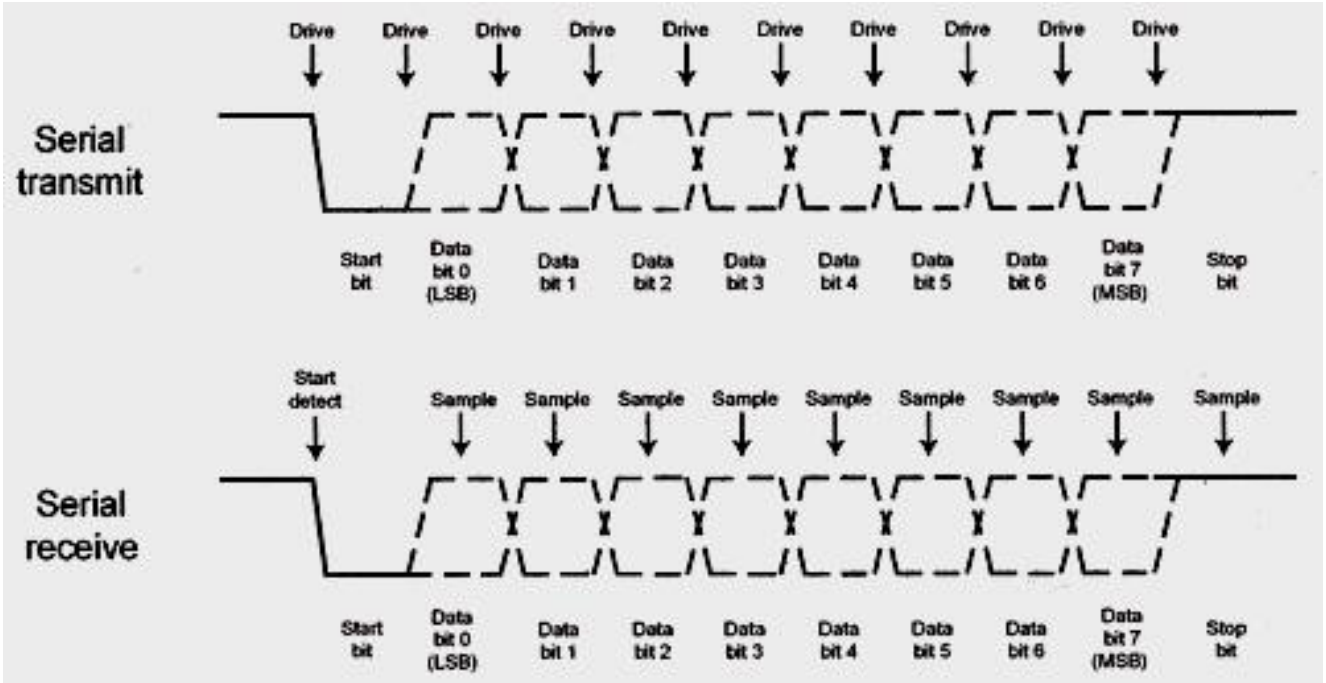




USART

Universal Synchronous and Asynchronous serial Receiver and Transmitter

- 구성 및 순서: **START + DATA + PARITY + STOP**
- **START** 비트 길이 = **1** 비트 레벨 = **0 (low)**
- **DATA** 비트 길이 = **5~ 9** 비트 레벨 = **X** (사용자 정의)
- **PARITY** 비트 길이 = **1** 또는 **None** 레벨 = 설정**(even, odd, none)**
- **STOP** 비트 길이 = **1~2** 비트 레벨 = **1 (high)**



USART

- **Baud Rate (bps)**

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	$BAUD = \frac{f_{OSC}}{2(UBRR+1)}$	$UBRR = \frac{f_{OSC}}{2BAUD} - 1$

$ex)\ f_{OSC} = 16MHz, BAUD = 9600bps, Async \qquad UBRR = \frac{f_{osc}}{16(BAUD)} - 1 = 103$

USART

- **Baud Rate (bps)**

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

Baud Rate (bps)	f_{osc} = 16.0000MHz			
	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)	1Mbps		2Mbps	

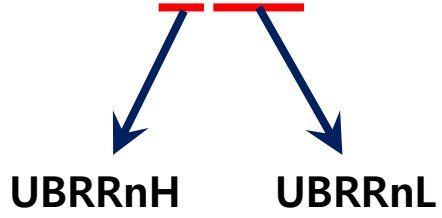
USART

- **Baud Rate (bps)**

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	

ex) 103 = 0x067



USART

- **UDRn(USART I/O Data Register)**

  - UDRn(USART I/O Data Register)

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	

- 송신 및 수신 데이터를 읽거나 쓸 수 있는 버퍼
- 읽기동작에서 수신 버퍼**(RXBn[7:0]**에 접근
- **ex) ch = UDR0; // 수신 버퍼에서 값을 읽고, ch 변수에 저장**
- 쓰기 동작 에서 송신 버퍼(TXBn[7:0])에 접근
- ex) UDR0 = dat; // dat 변수 값을 읽어, 송신 버퍼에서 쓰기

USART

- **Control Register**

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	

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	

- **Bit 7 – RXCn: USART Receive Complete**
- **Bit 6 – TXCn: USART Transmit Complete**
- **Bit 5 – UDREn: USART Data Register Empty**
- **Bit 4 – FEn: Frame Error**
- **Bit 3 – DORn: Data OverRun**
- **Bit 2 – UPEn: Parity Error**
- **Bit 1 – U2Xn: Double the USART Transmission Speed**
- **Bit 0 – MPCMn: Multi-Processor Communication Mode**

USART

- **Control Register**

UCSZn[2:0] [2 1 0]	문자 길이
0 0 0	5비트
0 0 1	6비트
0 1 0	7비트
0 1 1	**8비트 (기본값)**
1 0 0	-
1 0 1	-
1 1 0	-
1 1 1	9비트

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	

- **Bit 7 – RXCIEn: RX Complete Interrupt Enable**
- **Bit 6 – TXCIEn: TX Complete Interrupt Enable**
- **Bit 5 – UDRIEn: USART Data Register Empty Interrupt Enable**
- **Bit 4 – RXENn: Receiver Enable**
- **Bit 3 – TXENn: Transmitter Enable**
- **Bit 2 – UCSZn2: Character Size**
- **Bit 1 – RXB8n: Receive Data Bit 8**
- **Bit 0 – TXB8n: Transmit Data Bit 8**

USART

- **Control Register**

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	

- **Bit 6 – UMSELn: USART Mode Select**
- **Bit 5:4 – UPMn1:0: Parity Mode**

Table 77. UMSELn Bit Settings

UMSELn	Mode
0	Asynchronous Operation
1	Synchronous Operation

Table 78. UPMn Bits Settings

UPMn1	UPMn0	Parity Mode
0	0	Disabled
0	1	(Reserved)
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

USART

- **Control Register**

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	

- **Bit 3 – USBSn: Stop Bit Select**
- **Bit 2:1 – UCSZn1:0: Character Size**
- **Bit 0 – UCPOLn: Clock Polarity**

Table 79. USBSn Bit Settings

USBSn	Stop Bit(s)
0	1-bit
1	2-bits

Table 80. UCSZn Bits Settings

UCSZn2	UCSZn1	UCSZn0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

USART

```c
#define F_CPU 16000000UL
#define BAUD 9600
#define MYUBRR F_CPU/16/BAUD-1

#include <avr/io.h>
#include <util/delay.h>

void USART_Init(unsigned int ubrr){
    /*
     *  Baud Rate 9600, Stop Bit 1
     *  Character Size: 8-Bit
     *  No Parity
     */
    UBRR0H = (unsigned char)(ubrr >> 8);
    UBRR0L = (unsigned char)ubrr;
    UCSR0B = (1<<RXEN0) | (1<<TXEN0);
    UCSR0C = (3 << UCSZ0);
}

void USART_Transmit(char data){
    while(!((UCSR0A) & (1<<UDRE0)));     //Data Register Not Empty: Wait
    UDR0 = data;                         //Data Register Empty: Transmit
}

char USART_Receive(){
    while(!(UCSR0A & (1<<RXC0)));    //Receive Not complete: Wait
    return UDR0;                     //Receive Complete : Receive
}

int main(void)
{
    USART_Init(MYUBRR);

    while (1)
    {
        USART_Transmit(USART_Receive());  //Echo
        _delay_ms(100);
    }

}
```

Baud Rate: 9600bps
Stop Bit: 1-bit
Character Size: 8-bit
No Parity

Data Register Empty: Transmit
Not Empty: Wait

Receive Complete: Receive
Not Complete: Wait

Echo

USART

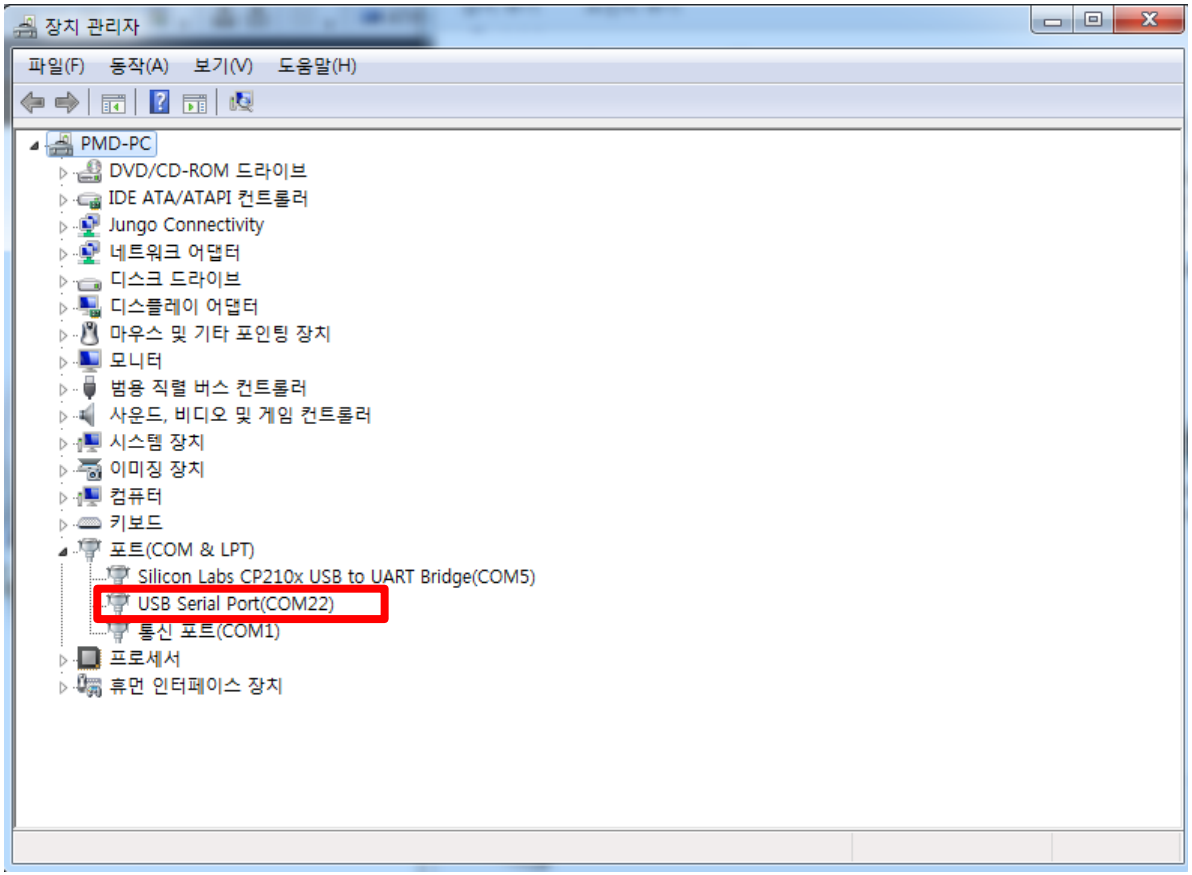
USB to USB mini

USB Serial Driver

RS232 to 3pin

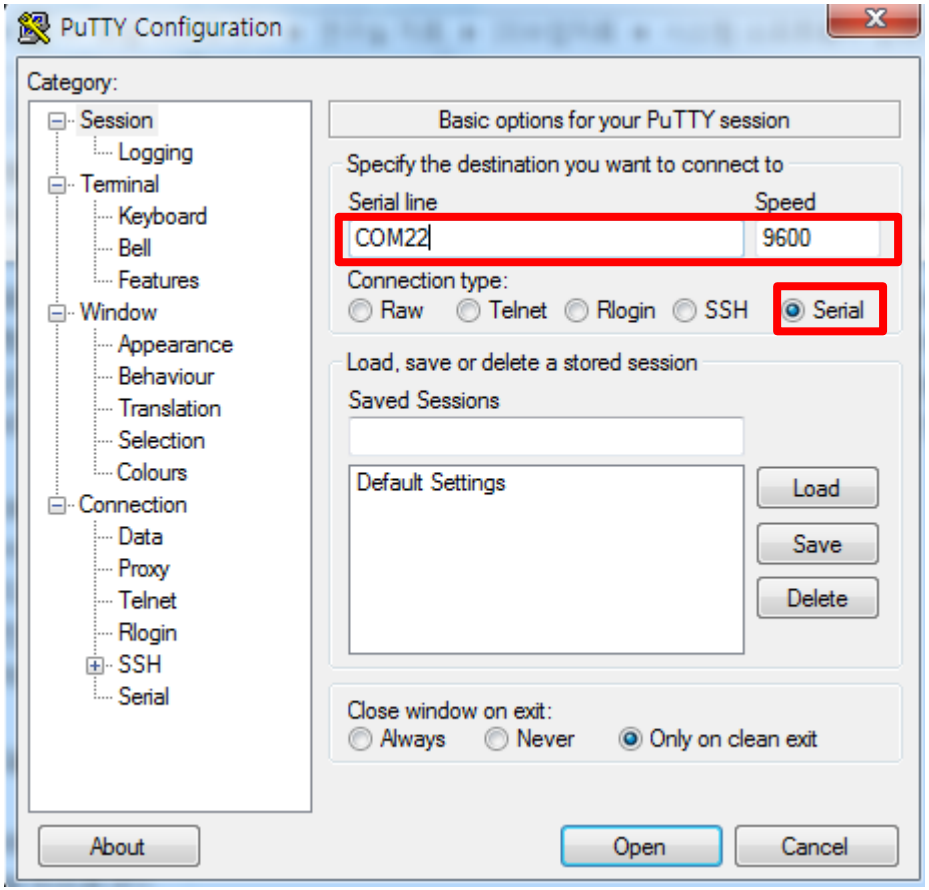
USB Serial Driver에 각각 RS232와 USB mini 를 연결
RS232 to 3pin의 3pin 연결을 보드의 3pin USART 0 에 연결

USART



장치 관리자에서 USB Serial Port 연결 확인

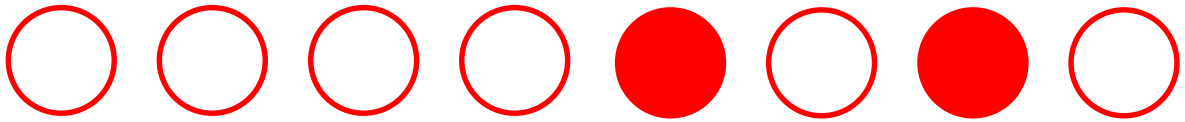
USART



설치된 Putty에서 Serial 통신, Speed 9600, Serial Port를 설정해준뒤 Open 이후에 터미널에 타이핑하여 결과를 확인

과제

1. 터미널에서 숫자를 입력받아 2진수로 변한한 뒤, 변환된 숫자를 LED로 표현하기 (EX: 10을 입력하면 LED가 다음과 같이 출력)



2. 스위치를 누르면 어떤 스위치가 눌렸는지 확인하고, 터미널에 해당 스위치 번호를 띄우고, LED 켜기

3. UART 0 가 아닌 USART 1을 사용하여 USART 통신하기