
마이크로컨트롤러 기능

제 9장 UART 통신



ICAT

Integrated Circuits for Advanced Technology Lab.

UART

1. UART와 RS232 개요
2. ATmega128A의 USART 포트
3. UART로 Hello 보내기
4. UART로 PC와 데이터 주고받기



UART와 RS232 개요

□UART(Universal Asynchronous Receiver/Transmitter)

- ❖ 시리얼 기반의 통신 방식으로 일반적으로 RS232 프로토콜을 통해 원격지와 통신을 지원하는 방식
- ❖ UART는 컴퓨터에게 RS-232C DTE 인터페이스를 제공함으로써, 모뎀이나 기타 다른 직렬장치들과 통신하거나 데이터를 주고받을 수 있게 한다.
- ❖ UART의 동작
 - 병렬 데이터를 하나의 단일 직렬 비트 스트림으로 변환
 - 직렬 비트 스트림을 컴퓨터가 처리할 수 있도록 바이트로 변환
 - 패리티 비트 처리
 - 시작 비트와 정지 비트 처리
 - 키보드나 마우스로부터 들어오는 인터럽트 처리



ATMega128A의 USART 포트

□ ATMega128A의 직렬통신 포트

- ❖ 직렬통신포트 USART(Universal Synchronous and Asynchronous Receive and Transmitter) 2개 내장
 - USART0
 - USART1
- ❖ 완전 이중방식(Full-Duplex) – 송신과 수신을 동시에 할 수 있음
- ❖ 동기 및 비동기 전송 가능
- ❖ 멀티 프로세서 통신 모드로 동작 가능
- ❖ 높은 정밀도의 보레이트 발생기 내장
- ❖ 인터럽트
 - 송신 완료(TX Complete)
 - 송신 데이터 레지스터 준비완료(TX Data Register Empty)
 - 수신완료(RX Complete)



ICAT

Integrated Circuits for Advanced Technology Lab.

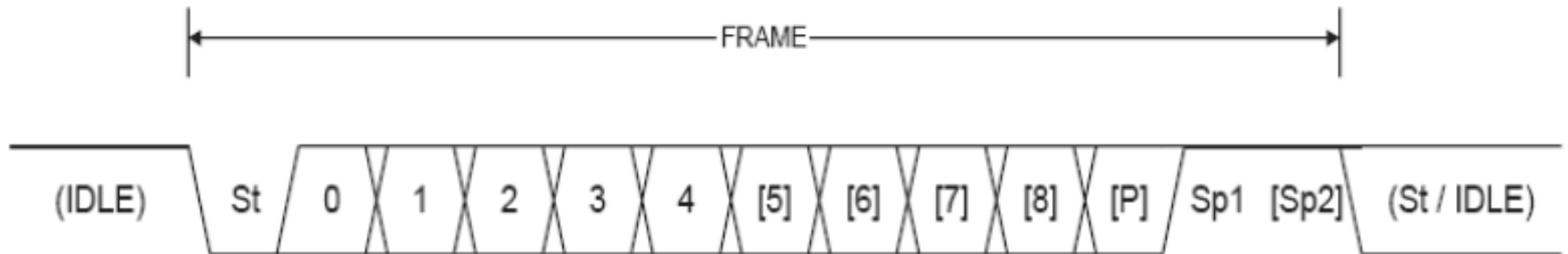
ATMega128A의 USART 포트

□ ATMega128A USART 데이터 프레임 포맷

❖ 최소 7비트 최대 13비트로 구성

- (1 비트의 스타트 비트) + (5,6,7,8,9 비트의 데이터 비트) + (0, 1 비트의 패리티비트) + (1,2 비트의 스탑비트) 프레임

USART 통신의 데이터 프레임



ATMega128A의 USART 포트

□ ATMega128A USART 데이터 프레임 포맷

❖ 스타트 비트

- 1비트로 이루어 졌으며 항상 0레벨이다. 송신시에 자동적으로 생성된다.

❖ 데이터 비트

- 5, 6, 7, 8, 9비트가 가능하다.

❖ 패리티 비트

- 패리티를 사용하지 않을 수도 있고 사용하는 경우 홀수 혹은 짝수 패리티 1비트를 사용한다.

❖ 스톱 비트

- 1, 2개의 비트가 가능하며 항상 1레벨이다. 송신시에 자동적으로 생성된다.



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128A의 USART 포트

□ ATMega128A USART 레지스터

- ❖ UDRn(Usart i/o Data Register n)
 - USART I/O 데이터 레지스터 (UDR0, UDR1)
- ❖ UCSRnA(Usart Control and Status Register n A)
 - USART 제어 및 상태 레지스터 A
- ❖ UCSRnB(Usart Control and Status Register n B)
 - USART 제어 및 상태 레지스터 B
- ❖ UCSRnC(Usart Control and Status Register n C)
 - USART 제어 및 상태 레지스터 C
- ❖ UBRRnH/L (USART BAUD RATE REGISTER)
 - USART baud Rate 레지스터



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128A의 USART 포트

- ❖ UDRn(Usart i/o Data Register n)
- ❖ USART I/O 데이터 레지스터 (UDR0, UDR1)
- ❖ USARTn 모듈의 송수신 데이터 버퍼의 기능을 수행하는 8비트 레지스터(n= 0, 1)
 - 송신 데이터를 UDRn에 write하면, 송신 데이터 버퍼 TXB에 저장
 - 수신 데이터를 UDRn에서 읽으면 수신 데이터 버퍼 RXB에 수신되어 있는 값이 읽힘

7	6	5	4	3	2	1	0
RXB7	RXB6	RXB5	RXB4	RXB3	RXB2	RXB1	RXB0
TXB7	TXB6	TXB5	TXB4	TXB3	TXB2	TXB1	TXB0



ATMega128A의 USART 포트

- ❖ UCSRnA(Usart Control and Status Register nA)
- ❖ USART 제어 및 상태 레지스터 A
- ❖ USARTn모듈의 송수신 동작을 제어하거나 송수신 상태를 저장하는 기능을 수행하는 8비트 레지스터

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

- ❖ 비트 7 : RXCn (USARTn Receiver Complete)
 - 수신버퍼에 수신문자가 있으면 “1”로 세트
 - 수신 버퍼가 비어있는 상태라면 “0”으로 클리어



ATMega128A의 USART 포트

- ❖ UCSRnA(Usart Control and Status Register nA)
- ❖ 비트 6 : TXCn (USARTn Transmit Complete)
 - 송신 데이터가 모두 송신되고 UDRn의 송신 버퍼에 아직 새로운 데이터가 저장되지 않은 상태이면 “1”로 세트
- ❖ 비트 5 : UDREn (USARTn Data Register Empty)
 - UDRn의 송신 버퍼에 새로운 송신 데이터를 받을 준비가 되어 있으면 “1”로 세트
- ❖ 비트 2 : UPEn (USARTn Parity Error)
 - UDR의 수신버퍼에 현재 저장되어 있는 데이터를 수신하는 동안에 패리티 에러가 발생하였음을 나타내는 상태 플러그
- ❖ 비트 1 : U2Xn (Double the USARTn Transmission Speed)
 - 비동기 모드에서만 사용가능,
 - 클록의 n분주비를 16에서 8로 1/2만큼 낮추어 전송속도를 2배 높이는 기능



ATMega128A의 USART 포트

- ❖ UCSRnB(Usart Control and Status Register nB)
- ❖ USART 제어 및 상태 레지스터 B
- ❖ USART 모듈의 송수신 동작 제어/송수신 상태 저장
- ❖ USART0, USART1 포트의 송수신 동작제어, 전송 데이터의 9번째 비트값 저장

7	6	5	4	3	2	1	0
RXCIE _n	TXCIE _n	UDRIE _n	RXEN _n	TXEN _n	UCSZ _{n2}	RXB8 _n	TXB8 _n

- ❖ 비트 7 : RXCIE_n (USART_n RX Complete Interrupt Enable)
 - 수신완료 인터럽트를 개별적으로 enable
 - 이 비트를 "1"로 설정하고 SREG 레지스터의 I비트가 "1"이고, UCSR_{nA} 레지스터의 RXC_n비트가 "1"로 설정되어 있으면 수신완료 인터럽트가 발생



ATMega128A의 USART 포트

- ❖ UCSRnB(Usart Control and Status Register n B)
- ❖ 비트 6 : TXCIEn (USARTn TX Complete Interrupt Enable)
 - 송신완료 인터럽트를 개별적으로 enable
 - 이 비트를 “1”로 설정하고 SREG레지스터의 I비트가 “1”이고, UCSRnA 레지스터의 TXCn비트가 “1”로 설정되어 있으면 송신완료 인터럽트가 발생
- ❖ 비트 5 : UDRIEn (USARTn Data Register Empty Interrupt Enable)
 - 송신 데이터 레지스터 준비완료 인터럽트(Data Register Empty)를 개별적으로 Enable
 - “1”로 설정하고 SREG레지스터의 I비트가 “1”이고, UCSRnA 레지스터의 UDR En비트가 “1”로 되면, USARTn Data Register Empty 인터럽트가 발생
- ❖ 비트 4 : RXENn (USARTN Receiver Enable)
 - USARTn 모듈의 수신부가 동작하도록 enable
 - RXDn 핀이 병렬 I/O포트가 아니라 직렬 데이터 수신단자로 동작하도록 설정



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128A의 USART 포트

- ❖ UCSRnB(Usart Control and Status Register n B)
- ❖ 비트 3 : TXENn (USARTn Transmitter Enable)
 - USARTn 모듈의 송신부가 동작하도록 enable
 - TXDn 핀이 병렬 I/O포트가 아니라 직렬 데이터 송신단자로 동작하도록 설정
- ❖ 비트 2 : UCSZn2 (USARTn Character Size)
 - UCSRnC레지스터의 UCSZn1~0비트와 함께 전송문자의 데이터 비트수를 설정



ICAT

Integrated Circuits for Advanced Technology Lab.

ATMega128A의 USART 포트

- ❖ UCSRnC(Usart Control and Status Register nC)
- ❖ USART 제어 및 상태 레지스터 C
- ❖ USARTn 모듈의 송수신 동작을 제어하거나 송수신 상태를 저장

7	6	5	4	3	2	1	0
-	UMSELn	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn

- ❖ 비트 6 : UMSELn(USARTn Mode Select)
 - USART 모드 설정
 - “1”이면 USARTn 모듈을 동기 전송모드로 설정하고, “0”이면 비동기 전송모드로 설정



ATMega128A의 USART 포트

- ❖ UCSRnC(Usart Control and Status Register nC)
- ❖ 비트 5, 4 : UPMn1, 0 (USARTn Parity Mode)
 - 패리티 모드 설정
 - UPMn1비트를 “1”로 설정하면 패리티를 발생
 - 오류가 발생하면 UCSRnA 레지스터의 PE플래그가 “1”로 세트

패리티 모드 설정표

UPMn1	UPMn0	Parity모드
0	0	Disable
0	1	예약
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity



ATMega128A의 USART 포트

- ❖ UCSRnC(Usart Control and Status Register n C)
- ❖ 비트 2, 1 : UCSZn1, 0(USARTn Character Size)
 - UCSRnB 레지스터의 UCSZn2 비트와 함께 전송문자의 데이터 비트수를 설정

UCSZn에 의한 character Size 설정표

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	예약
1	0	1	예약
1	1	0	예약
1	1	1	9-bit



ATMega128A의 USART 포트

- ❖ UCSRnC (Usart Control and Status Register n C)
- ❖ 비트 3 : USBSn (Usartn Stop Bit Select)
 - '0' 이면 스톱비트 1개, "1"이면 스톱비트를 2개로 설정
- ❖ 비트 0 : UCPOLn (Usart Clock POLarity n)
 - 동기 전송 모드의 슬레이브 동작에서만 유효
 - '1'로 설정하면 송신 데이터는 클럭의 하강에지에서 새로운 XCKn 값이 출력되고, 수신 문자는 XCKn의 상승에지에서 얻어진다.
 - '0'로 설정하면 반대로 된다.



ATMega128A의 USART 포트

- ❖ UBRRnH/L (USART BAUD RATE REGISTER)
- ❖ USART baud Rate 레지스터
- ❖ USARTn 모듈의 송수신 속도를 설정
- ❖ 16비트중에서 12비트만 사용

15	14	13	12	11	10	9	8
-	-	-	-	UBRRn	UBRRn	UBRRn	UBRRn
7	6	5	4	3	2	1	0
UBRRn7	UBRRn6	UBRRn5	UBRRn4	UBRRn3	UBRRn2	UBRRn1	UBRRn0

- ❖ 비트 11~0 : UBRRn11~0
 - 12비트를 이용하여 USARTn의 Baud Rate을 결정
 - UBRRnH의 4비트와 UBRRnL의 8비트가 조합을 이룸.



ATMega128A의 USART 포트

❖ UBRRnH/L (USART BAUD RATE REGISTER)

UBRR에 의한 Baud Rate 설정표

Baud Rate (bps)/14.7456MHz	비동기 일반모드 (U2Xn = 0)		비동기 2배속 모드 (U2Xn=1)	
	UBRRn	Error	UBRRn	Error
2400	383	0.0%	767	0.0%
4800	191	0.0%	383	0.0%
9600	95	0.0%	191	0.0%
14,400	63	0.0%	127	0.0%
19,200	47	0.0%	95	0.0%
28,800	31	0.0%	63	0.0%
38,400	23	0.0%	47	0.0%
57,600	15	0.0%	31	0.0%
76,800	11	0.0%	23	0.0%
115,200	7	0.0%	15	0.0%
230,400	3	0.0%	7	0.0%
250,000	3	-7.8%	6	5.3%
500,000	1	-7.8%	3	-7.8%
1,000,000	0	-7.8%	1	-7.8%



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 13 : UART로 Hello 보내기

□ 실습 개요

- ❖ UART를 이용하여 미리 작성된 문장(“Hello World”)을 PC로 전송하는 실습
- ❖ ATmega128A의 USART 포트를 입력과 출력으로 선언하고 이 포트를 UART 모듈에 연결.
- ❖ USB 케이블을 이용하여 PC와 연결.

□ 실습 목표

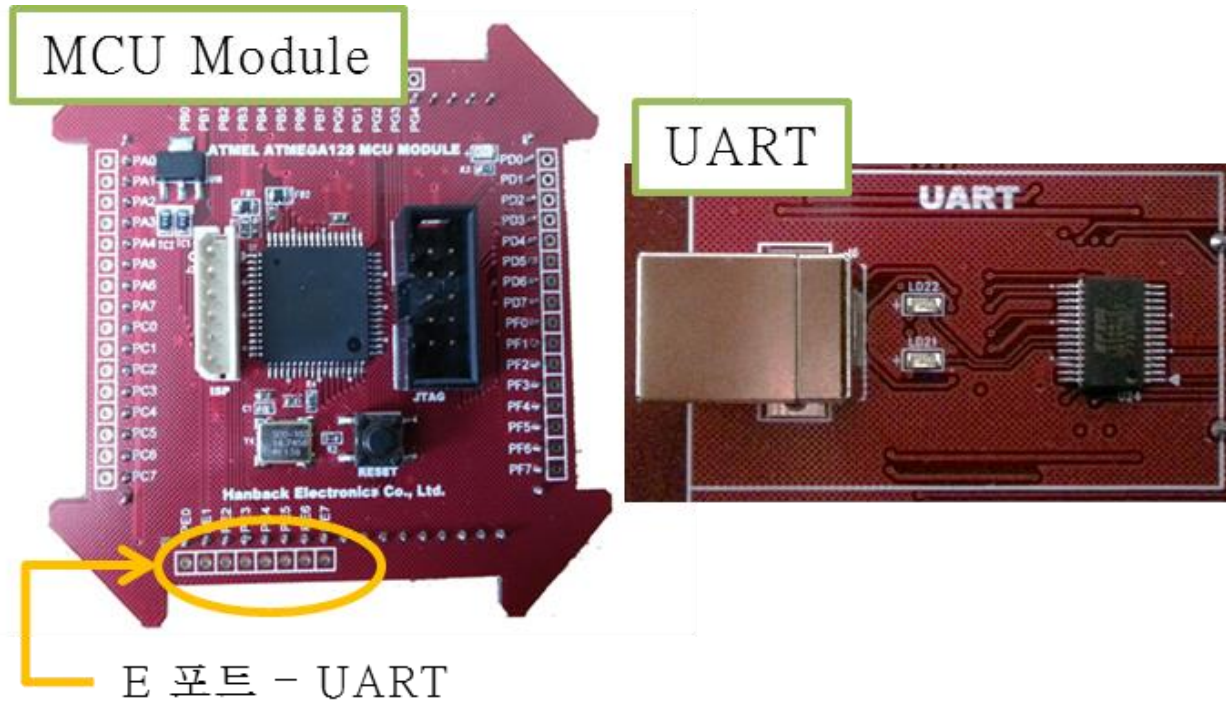
- ❖ UART 기능 동작원리 이해
- ❖ ATmega128A의 USART 제어 방법의 습득(관련 레지스터 이해)
- ❖ UART를 통해 PC와 통신하는 방법 습득



실습 13 : UART로 Hello 보내기

□ 사용 모듈

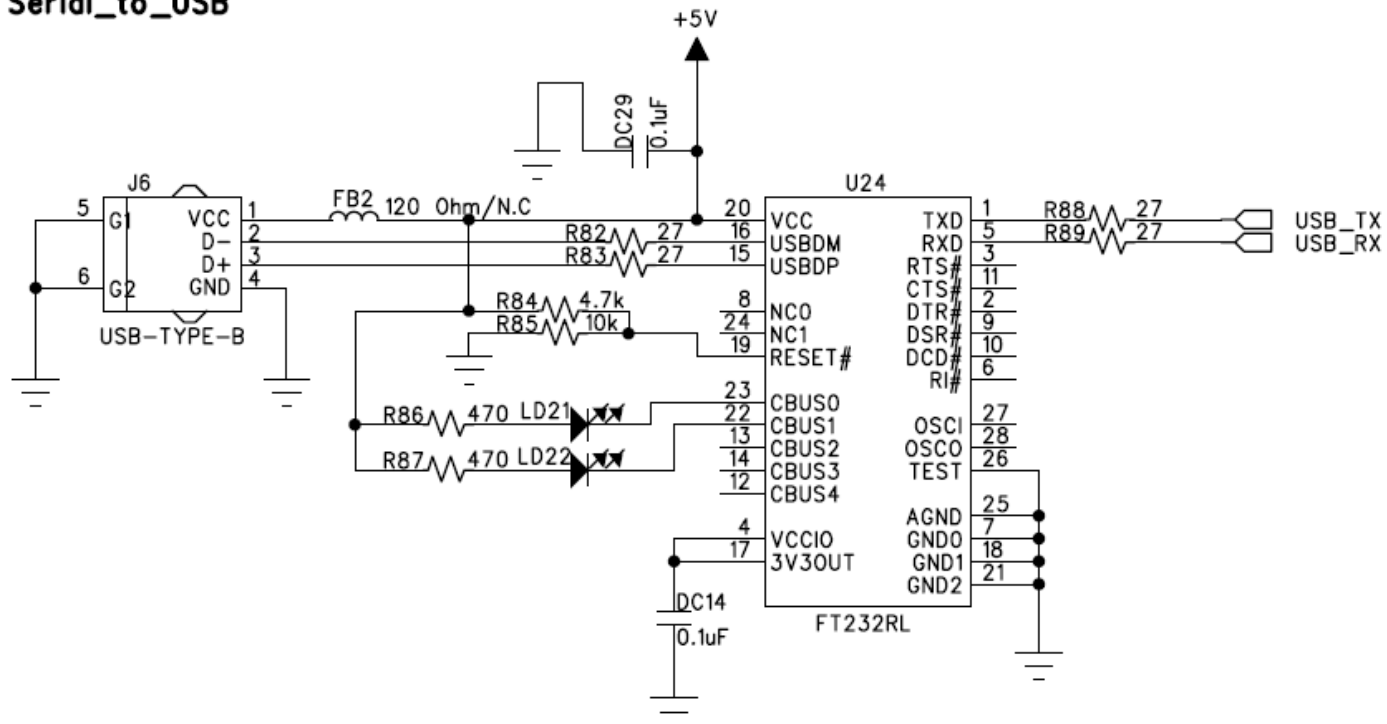
- ❖ MCU 모듈, UART 모듈



실습 13 : UART로 Hello 보내기

□ 사용 모듈 : UART 모듈 회로

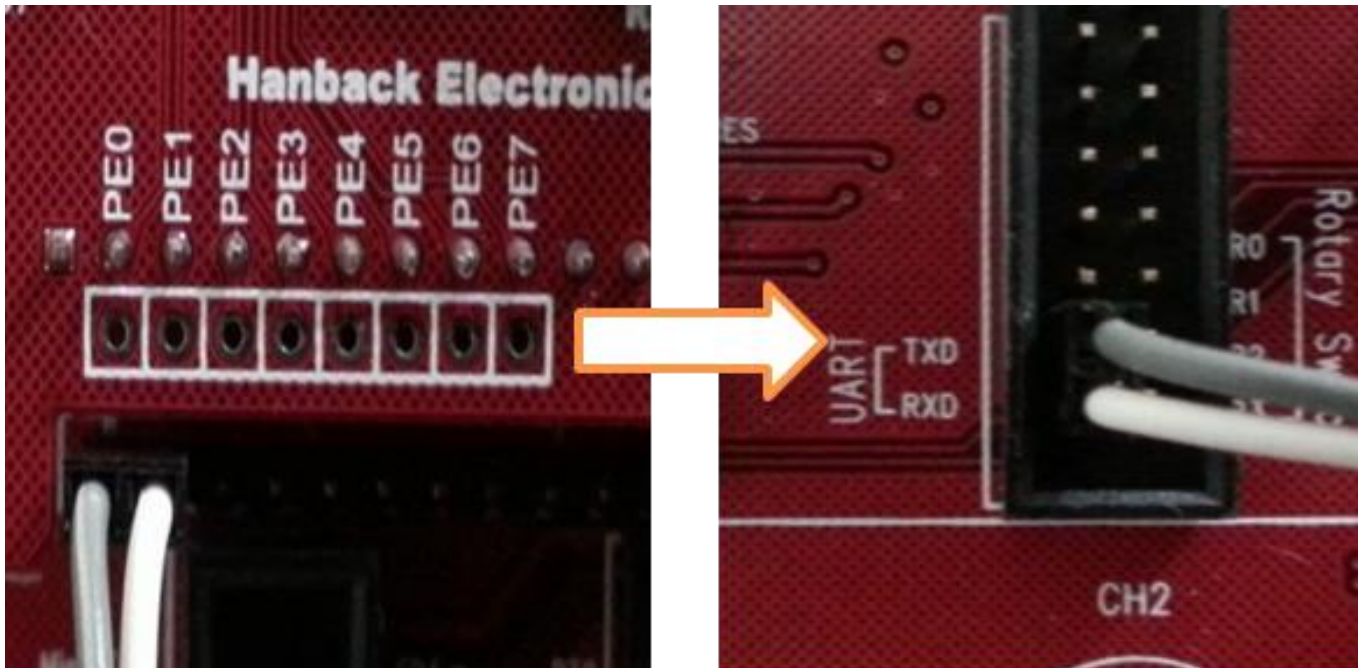
Serial_to_USB



실습 13 : UART로 Hello 보내기

□ 모듈 결선 방법

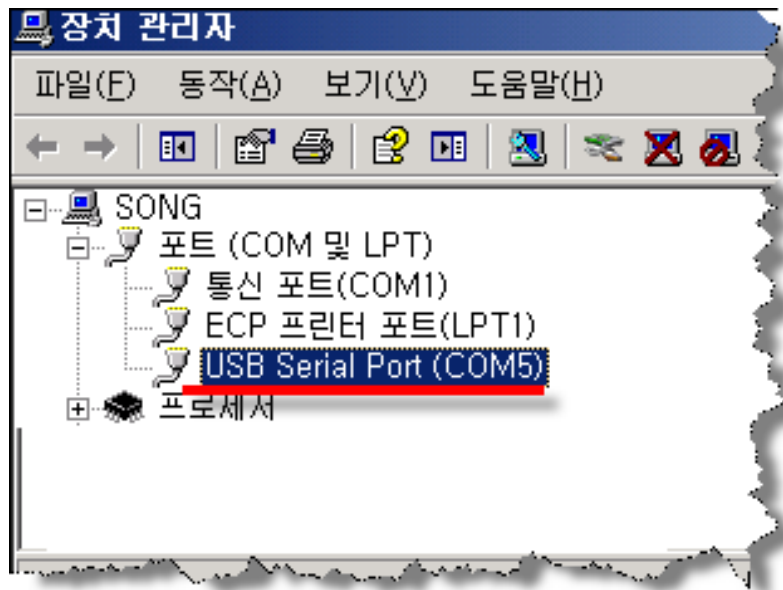
- ❖ MCU 모듈 포트 E의 PE0는 UART 모듈의 TXD핀에 연결
- ❖ MCU 모듈 포트 E의 PE1은 UART 모듈의 RXD핀에 연결
- ❖ UART 모듈의 USB 포트를 통해 PC의 USB 포트와 연결



실습 13 : UART로 Hello 보내기

□ 모듈 결선 방법

- ❖ UART모듈을 PC와 연결하면 “USB Serial Port” 로 인식
- ❖ UART모듈의 COM포트를 확인 연결



실습 13 : UART로 Hello 보내기

- ❖ 구동 프로그램 : 사전 지식
- ❖ 미리 정해 주어야 하는 통신 규약을 결정

Baud Rate	115200 (UBRR0H=0x00, UBRR0L=0x07)
패리티	No Parity (UCSR0C=00 <u>0</u> 0110)
Stop Bit	1 (UCSR0C=0000 <u>0</u> 110)
전송문자 데이터 비트수	8 (UCSR0B=00011 <u>0</u> 00, UCSR0C=00000 <u>1</u> 10)
흐름제어	없음

- ❖ UART 제어 레지스터 세팅
 - 비동기 전송 모드 (UCSR0C=0x06, 00000110)
 - 멀티 프로세서 통신 모드 (UCSR0A=0x00, 00000000)
 - UART의 RX와 TX를 Enable (UCSR0B=0x18, 00011000)
- ❖ UCSR0A 레지스터의 플래그 를 보면서 데이터를 보낼 수 있는 상태를 기다렸다가 UDR0 레지스터에 데이터를 넣어주면 UART로 데이터가 출력됨.



실습 13 : UART로 Hello 보내기

□ 구동 프로그램 : 소스 분석

❖ UART_Hello.c

1)	<pre>#include<avr/io.h></pre>
2)	<pre>void putch(unsigned char data){ while((UCSR0A & 0x20) == 0); //전송준비가 될 때까지 대기 UDR0 = data; UCSR0A = 0x20; /* UCSR0A 레지스터 중에서 5번째 비트(UDREN) 를 세트하여 UDRE0를 클리어 함. */ } int main(){</pre>
3)	<pre>unsigned char text []="Hello! Wolrd!! WrWn"; unsigned char i=0;</pre>



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 13 : UART로 Hello 보내기

4)	<pre>DDRE = 0xfd; //Rx(입력 0), Tx(출력, 1) UCSR0A = 0x00; /* 0번째 비트, 즉 MPCMn 를 0으로 세트 (USARTn을 멀티 프로세서 통신모드로 설정) */ UCSR0B = 0x18; //Rx, Tx enable /* 3번째, 4번째 비트 세트 즉, TXENn (USARTn모듈의 송신부 동작 enable) RXENn (USARTn모듈의 수신부 동작 enable) 2번 비트 UCSZ02 = 0으로 세트 */ UCSR0C = 0x06; //비동기 방식, No Parity bit, 1 Stop bit /* 1번째, 2번째 비트 세트 즉, UCSZ0(1:0) = "11"로 세트 데이터 비트수를 8비트로 결정 */ UBRR0H = 0x00; UBRR0L = 0x07; //14.7456 MHz -> 115200 bps /* USARTn모듈의 송수신 속도를 설정 */</pre>
5)	<pre>while(text[i] != '\0') putchar(text[i++]); return 0; }</pre>



실습 13 : UART로 Hello 보내기

□ 실행 결과

- ❖ 하이퍼터미널을 다음과 같이 설정한 다음 MCU를 리셋하여 문자열을 확인한다.

Baud Rate	115200
패리티	No Parity
Stop Bit	1
전송문자 데이터 비트수	8
흐름제어	없음

프로그램
다운로드시에는
PC의 하이퍼
터미널을
연결해제상태로...

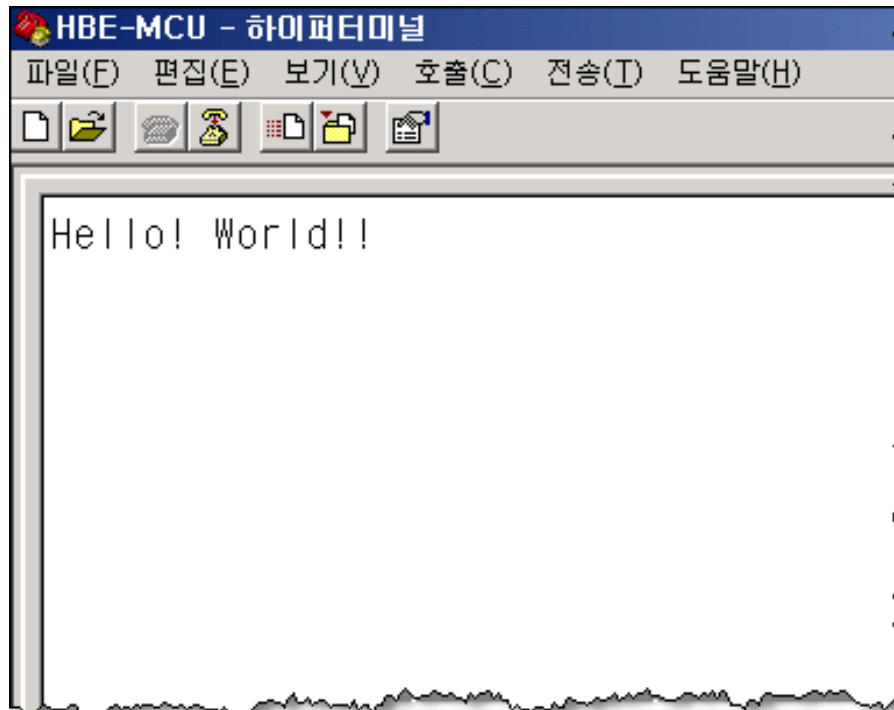


ICAT

Integrated Circuits for Advanced Technology Lab.

실습 13 : UART로 Hello 보내기

□ 실행 결과



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 14 : UART로 PC와 데이터 주고받기

□실습 개요

- ❖ PC로부터 전송되는 문자열을 받아 다시 PC로 되돌려 전송하도록 함.
- ❖ 설정은 앞의 예제와 유사
- ❖ 프로그램의 작성된 문장을 PC 화면에 뿌려주는 역할과 키보드를 통해 입력 받은 ASCII값을 화면상에 보여주는 기능을 함.

□실습 목표

- ❖ UART 기능 동작원리 이해
- ❖ ATmega128A의 USART 제어 방법의 습득(관련 레지스터 이해)
- ❖ UART를 통해 PC와 통신하는 방법 습득



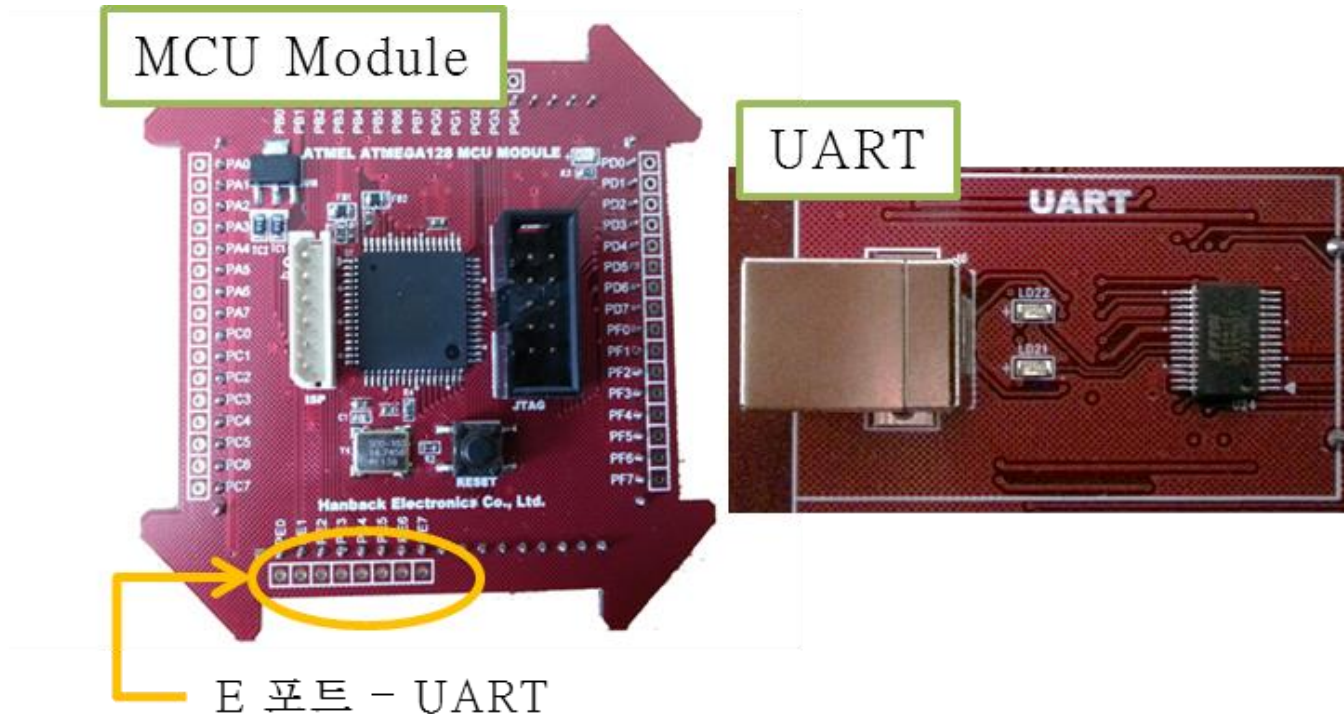
ICAT

Integrated Circuits for Advanced Technology Lab.

실습 14 : UART로 PC와 데이터 주고받기

□ 사용 모듈

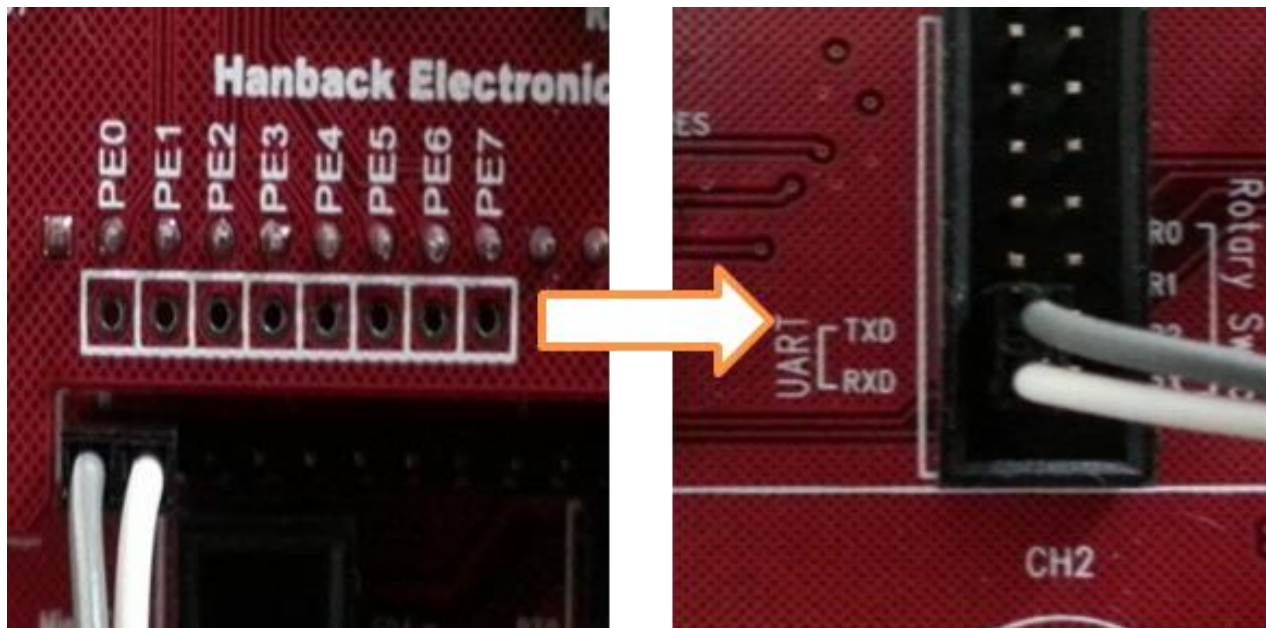
- ❖ MCU 모듈, UART 모듈



실습 14 : UART로 PC와 데이터 주고받기

□ 모듈 결선 방법(앞 예제와 동일)

- ❖ MCU 모듈 포트 E의 PE0는 UART 모듈의 TXD핀에 연결
- ❖ MCU 모듈 포트 E의 PE1은 UART 모듈의 RXD핀에 연결
- ❖ UART 모듈의 USB 포트를 통해 PC의 USB 포트와 연결



실습 14 : UART로 PC와 데이터 주고받기

□ 구동 프로그램 : 사전 지식

- ❖ ATmega128A의 USART 포트를 통해서 PC와 UART 통신을 연결하고, PC로 받은 문자열을 그대로 되돌려 전송.
- ❖ **설정** : 기본적인 UART 설정은 이전 예제와 동일하게 설정
- ❖ **데이터 수신** : 모든 설정이 끝나면 UCSR0A 레지스터의 플래그를 보면서 PC로부터 데이터가 도착했는지 살펴보고 있다가 데이터가 도착하면 UDR0 레지스터로부터 데이터를 가져오면 된다.
- ❖ **데이터 송신** : UCSR0A 레지스터의 플래그를 보면서 데이터를 보낼 수 있는 상태를 기다렸다가 UDR0 레지스터에 데이터를 넣어주면 UART로 데이터가 출력될 것이다.



ICAT

Integrated Circuits for Advanced Technology Lab.

실습 14 : UART로 PC와 데이터 주고받기

□ 구동 프로그램 : 소스 분석

❖ UART_Echo.c

1)	<pre>#include<avr/io.h> void putch(unsigned char data){ while((UCSR0A & 0x20) == 0); //전송준비가 될 때까지 대기 UDR0 = data; UCSR0A = 0x20; /* UCSR0A 레지스터 중에서 5번째 비트(UDREn) 를 세트하여 UDRE0를 클리어 함. */ }</pre>
2)	<pre>unsigned char getch(){ unsigned char data; while((UCSR0A & 0x80) == 0); //데이터를 받을 때까지 대기 data = UDR0; UCSR0A = 0x80; /* UCSR0A 레지스터 중에서 7번째 비트를 세트하여 RXC0를 클리어 함. */ return data; } int main(){</pre>



ICAT

Integrated Circuits for Advanced Technology Lab.

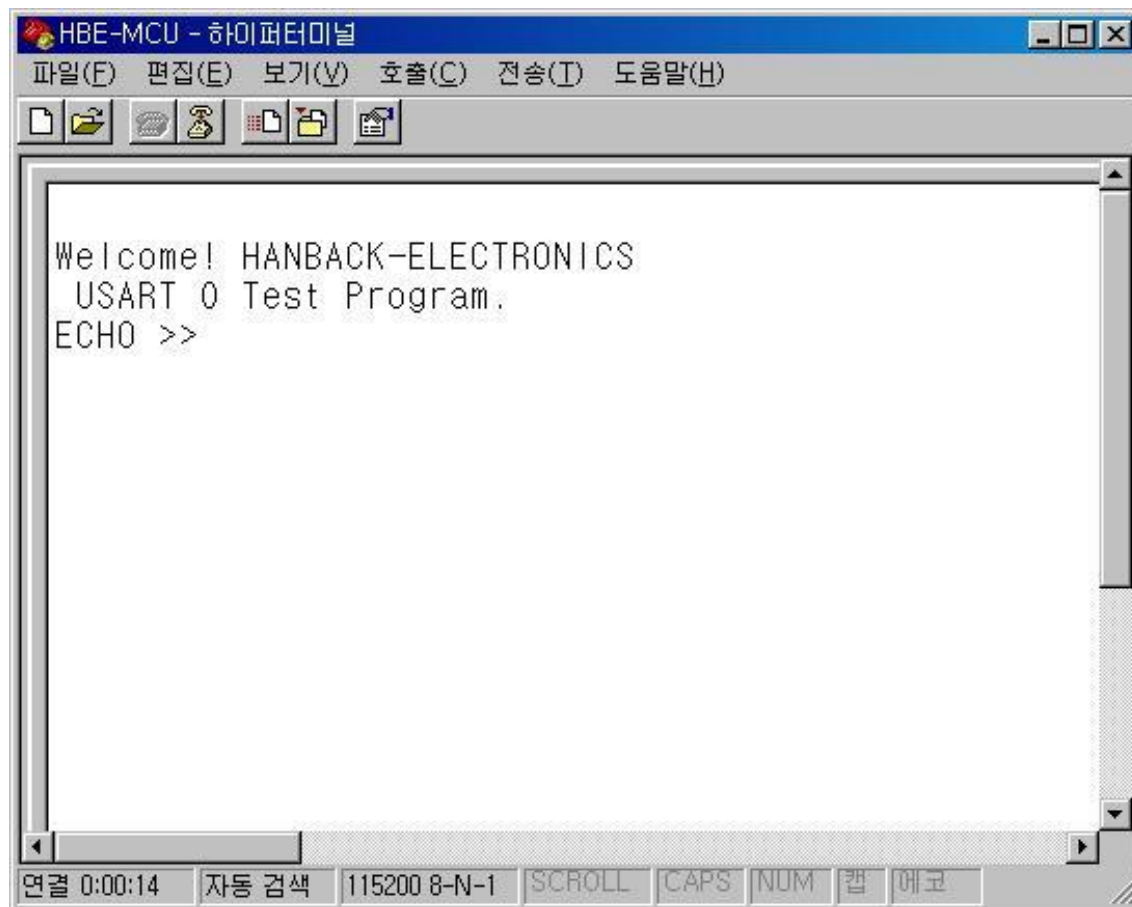
실습 14 : UART로 PC와 데이터 주고받기

3)	<pre>unsigned char text[]="WrWnWelcome! Chungbuk National UniWrWn USART 0 Test Program.WrWn"; unsigned char echo[]="ECHO >> "; unsigned char i=0;</pre>
4)	<pre>DDRE = 0xfd; //Rx(입력 0), Tx(출력, 1) UCSR0A = 0x00; UCSR0B = 0x18; //Rx, Tx enable UCSR0C = 0x06; //비동기 방식, No Parity bit, 1 Stop bit UBRR0H = 0x00; UBRR0L = 0x07; //14.7456 MHz 에서 115200 bps</pre>
5)	<pre>while(text[i] != '\0') putchar(text[i++]); i=0; while(echo[i] != '\0') putchar(echo[i++]);</pre>
6)	<pre>while(1) putchar(getch()); return 0; }</pre>



실습 14 : UART로 PC와 데이터 주고받기

□ 실행 결과



```
HBE-MCU - 하이퍼터미널
파일(E) 편집(E) 보기(V) 호출(C) 전송(T) 도움말(H)
Welcome! HANBACK-ELECTRONICS
USART 0 Test Program.
ECHO >>
```

연결 0:00:14 자동 검색 115200 8-N-1 SCROLL CAPS NUM 캡 메코