

Lab8. UART

실습 목표

- 주어진 코드를 바탕으로 UART의 동작을 이해한다.
- Asynchronous 통신의 동작을 이해한다.

실습 내용과 결과물

PC에서 terminal emulation program으로부터 입력된 character를 UART의 RX 모듈을 통해 수신하고, 이 character를 UART TX 모듈을 통하여 시리얼로 PC에 전송한다. 또한, 수신한 character를 7 Segments로 표시한다.

실습 준비물

vivado 2016.x 버전 (64bit 운영체제 필요)

NEXYS4 DDR Board

터미널 에뮬레이션 프로그램 (ex. teraterm)

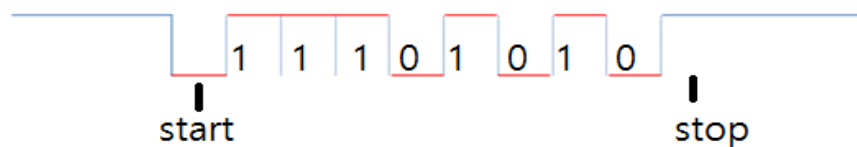
● 선수지식

➤ 시리얼 통신 프로토콜 - UART

UART는 Universal asynchronous receiver/transmitter의 약어이다. UART는 시리얼 통신 혹은 직렬 통신의 하나로, 이는 병렬 통신과 반대되는 개념이다. 시리얼 통신은 한 비트씩 순차적으로 데이터를 보내는 방식이고 병렬 통신은 여러 비트를(대개 수 바이트) 한번에 보내는 방식이다. 시리얼 통신은 병렬 통신보다 전송이 느린 대신 구조가 간단하며, 가격이 저렴하고 만들기가 쉽다. 또한 통신 거리가 멀다. 시리얼 통신의 대표적인 프로토콜은 UART, I2Q(아이 스퀘어 씨), SPI 등이 있다. UART는 직렬 통신 중에서도 비 동기식이며, TTS 신호 level을 사용한다. 이는 노이즈에 취약하기 때문에 RS-232, RS-422, RS-485와 같은 통신 표준과 함께 사용한다. UART 통신 프로토콜은 Stop비트의 길이, parity 비트, 흐름제어, 전송률(baud rate)을 사전에 송/수신자 간 약속해야 한다.

➤ UART 신호

UART는 주로 1 character(1 byte)단위로 데이터를 전송한다.



(그림 8.1) UART 신호의 예

(그림 8.1)에서 1-1-1-0-1-0-1-0은 전송하는 데이터이다. 한 비트씩 순차적으로 한 바이트를 전송하는 것을 확인할 수 있다. 이같이 한 단위의 데이터를 보내는 과정을 한 프레임이라고 한다. 한 프레임의 구성은 다음과 같다.

- 1) Start bit – 신호가 High to Low로 떨어짐
- 2) Data - 8비트의 데이터가 LSB부터 1비트씩 순차적으로 나타남
- 3) parity bit – (그림 8.1)에는 나와있지 않지만, error를 체크하기 위해 parity bit를 사용하는 경우가 있음. parity bit은 전송하는 데이터 중 'High' bit의 개수가 홀수인지 짝수인지 표시함.
- 4) Stop bit – 한 프레임의 전송이 끝남을 알려주는 것으로, 1~2 bit width를 가진다. 전송자와 수신자 사이에 사전에 약속된 길이.

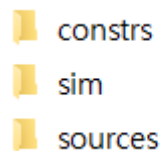
➤ Baud rate

Baud rate는 프랑스 한 전신 기술자의 이름을 딴 것인데, 일반적으로 bit율(BPS, bits/sec)과 같다. 데이터 전송 속도를 의미한다. baud rate가 9600이라면, 1초에 9600비트를 전송한다는 것이다.

● 실습 따라하기

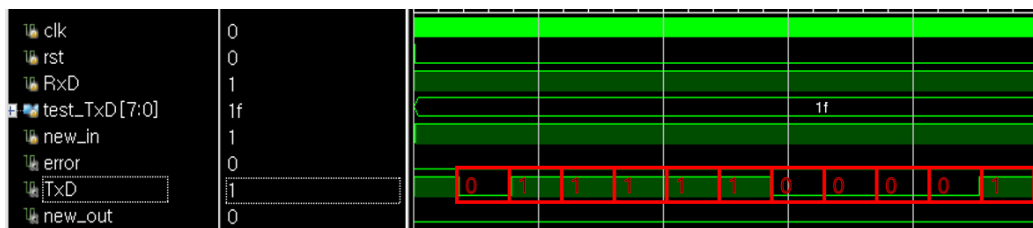
➤ UART_top

프로젝트를 생성하고 주어진 소스파일을 추가한다.



(그림 8.2) Given Files

소스파일을 추가한 뒤, <tb_UART.v> 파일을 이용해 시뮬레이션을 진행한다.

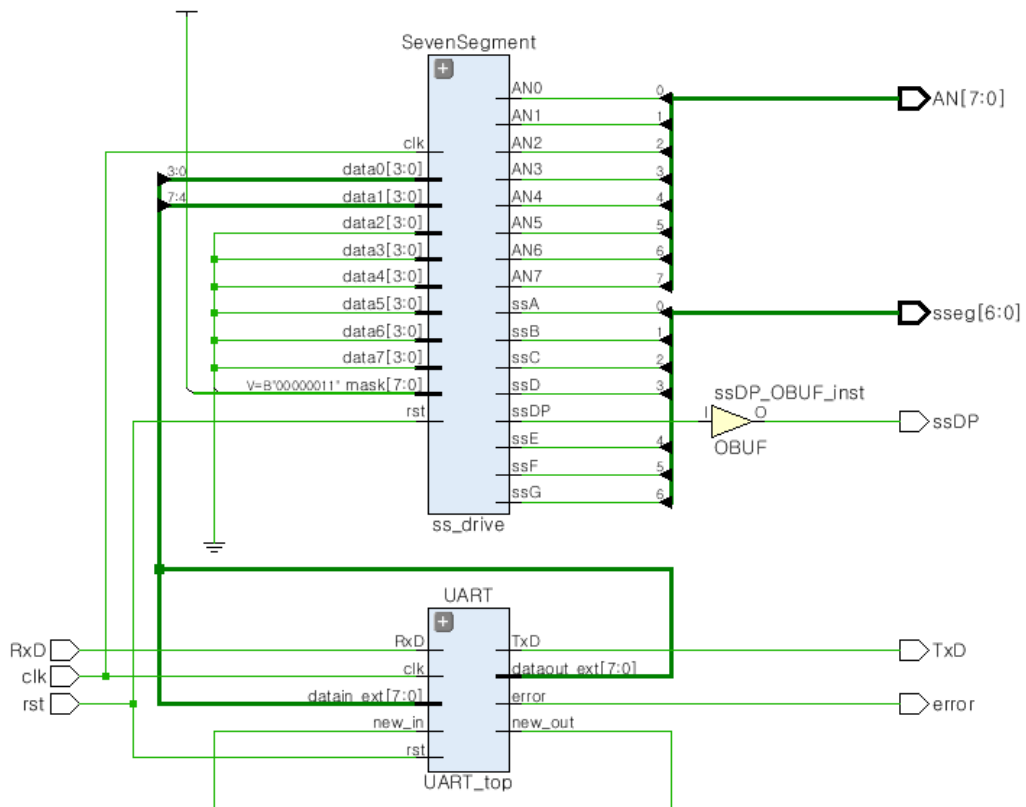


(그림 8.3) UART Wave Form

(그림 8.3)에서, 처음 1비트는 start bit로, High에서 Low로 처음 떨어진 신호이다. 10진수 31은 16진수로 0x1F에 해당하는데, UART는 LSB를 먼저 전송하기 때문에, 전송하는 데이터의 sequence는 1 - 1 - 1 - 1 - 1 - 0 - 0 - 0 이어야 한다. (그림 8.3)에서, start bit이후로 위의 sequence가 정상적으로 나타나는 것을 확인할 수 있다. 마지막 2 비트는 parity bit와 stop bit이다. Parity bit는 위 design의 경우 내부적으로 odd parity로 되어있다. 전송하는 데이터 'd31'에는 High가 총 5개 있다. 모듈 내에서 정한 Parity는 홀수로 High이므로, parity를 포함하면 총 6개의 High가 XOR된다. High가 짝수 만큼 있기 때문에 parity bit는 Low이다. Stop bit는 1bit이며, High이다.

➤ UART_example

UART_example은 UART모듈을 이용해 PC에서 전송받은 character를 7 Segments에 표시하고, 다시 PC로 전송(echo)하는 모듈이다. 아래 Schematic View를 참고해 UART_example 모듈을 기술한다.

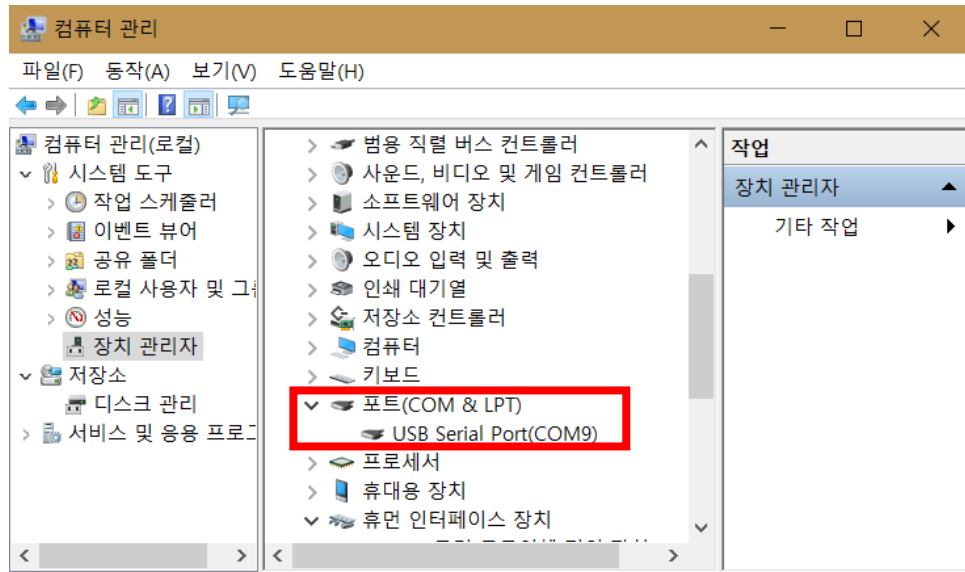


(그림 8.4) UART_example Schematic View

(그림 8.4)의 UART 모듈의 입/출력을 살펴보면, RxD 입력을 통해 수신한 데이터를 `dataout_ext` 출력으로 내보내고, 그 데이터를 다시 `datain_ext`로 되먹이는 것을 알 수 있다. 이는 echo의 기능을 구현한 것이다.

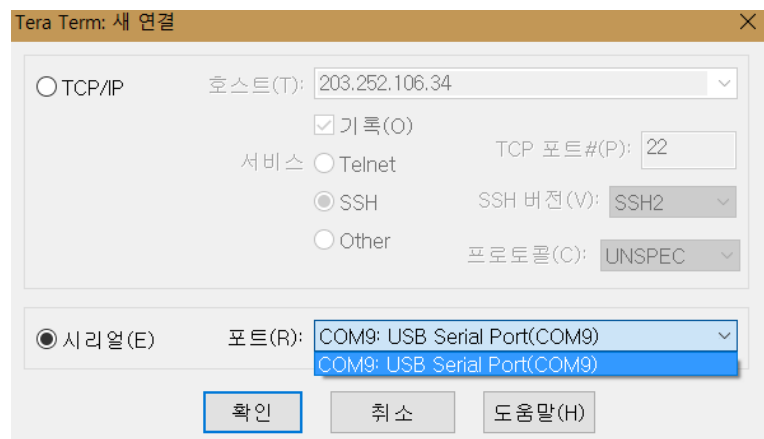
➤ 동작 확인

UART_example 기술을 완료한 뒤, <Synthesis>, <Run Implementation>, <Generate Bitstream> 과정을 거쳐 bit 파일을 생성한다. 생성한 bit 파일을 <Hardware Manager>를 통해 Nexys4 DDR 보드에 프로그램한 뒤 컴퓨터의 <장치관리자>를 실행한다.



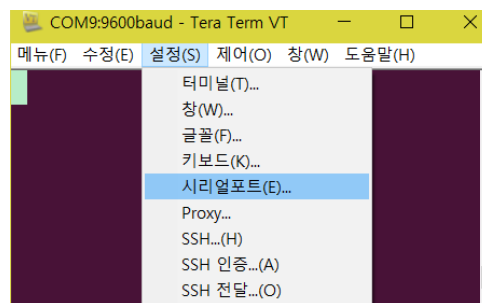
(그림 8.5) 장치관리자

(그림 8.5)에서 USB Serial Port가 COM9번으로 잡힌 것을 확인했다. 다음으로, PC에서 terminal emulator program을 실행한다. 본 실습서에서는 <tera term>을 사용한다.



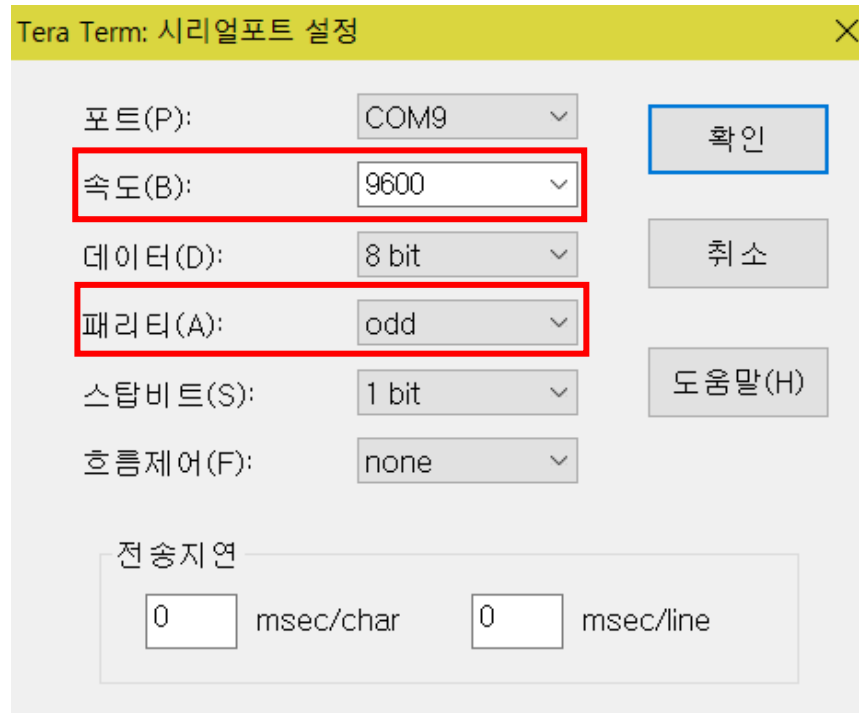
(그림 8.6) Tera Term 시작화면

(그림 8.6)에서, 시리얼 포트 연결을 시도한다. 포트 list에서 COM9번 포트를 선택한다.



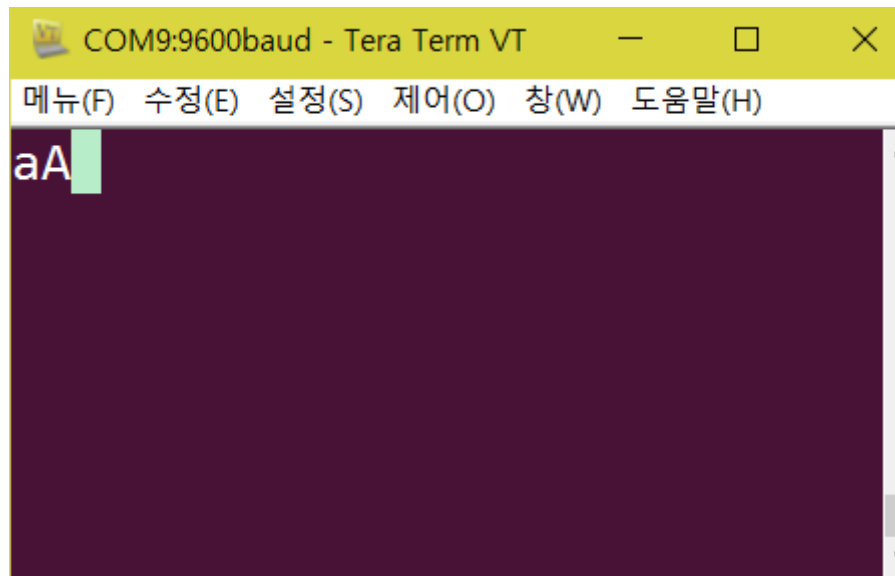
(그림 8.7) Tera Term 설정

(그림 8.7)과 같이 <설정>, <시리얼포트> 를 눌러 시리얼포트 설정 창을 연다.



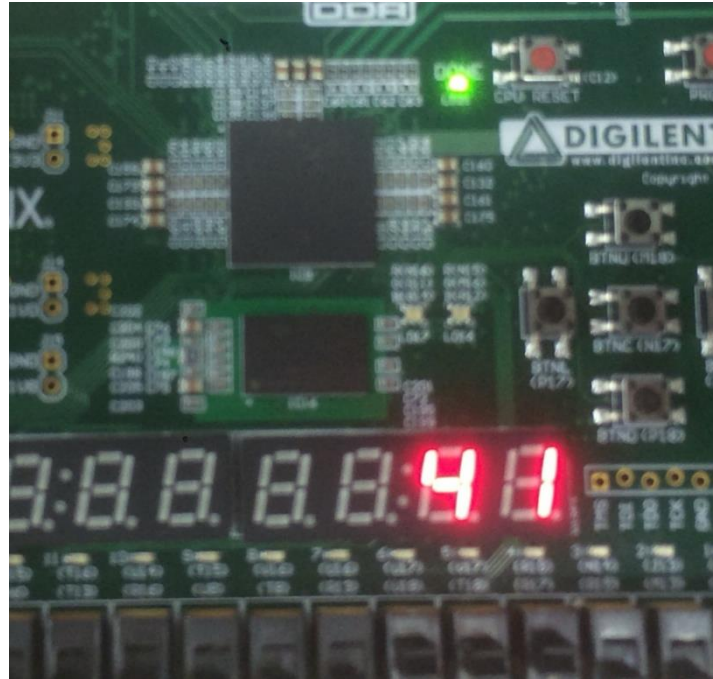
(그림 8.8) 시리얼포트 설정

(그림 8.8)과 같이 설정한다. 전송속도는 9600 bit/s이며, 한 프레임에 8비트의 데이터를 전송한다. 패리티 비트는 odd이며, stop 비트는 1bit으로 한다. 흐름제어는 하지 않는다.



(그림 8.9) 데이터 전송

(그림 8.9)은 tera term 창에 'a'와 'A'를 차례로 입력한 것이다.



(그림 8.10) 7 Segments

(그림 8.10)는 'A' 를 수신해 7 Segments에 표시한 것이다.