

Serial(직렬) 통신

1, 종류

- Morse부호를 사용하는 통신
- RS-232, RS-422, RS-485
- USB, CAN, TCP/IP
- I2C, SPI, UART/USART

2, 특징

- 통신 거리가 멀어도 사용하기 용이하다
- 구현이(병렬 통신에 비해) 비교적 쉽다
- 고주파수의 클럭의 사용이 가능
→ 크로스토크(Crosstalk) 등의 전기적 특징이 유리함

※ 크로스토크(Crosstalk) : 선간의 결합에 의한 신호
및 노이즈 전파

3, 중요 요소

- **Baud Rate** : 송신과 수신 사이의 초당 전송 **Bits** 수를 의미.
일반적으로 **9600**을 사용하지만, **1200,2400,4800, 57600, 115200** 등 선택해서 쓸 수 있고 높을 수록 빠르다.
송신과 수신 **Baud Rate**를 동일하게 설정해야 함
- **Framing** : 데이터 프레임
일반적으로 **8bits**를 사용하곤 하지만, **5,6,7bits**도 모두 사용이 가능하다
- **Error Control** : 시리얼 통신에는 노이즈에 영향을 주는 요소들이 많기 때문에, 에러 컨트롤이 필요하다. 그래서 이러한 오류들을 제거하기 위해 **Parity Bits**를 검사

4, 사용 형태

- 단방향(Simplex) : 데이터가 한 방향($A \rightarrow B$)으로만 전송되는 것(라디오 방송, TV)
- 반이중(Half-Duplex) : 데이터가 양 방향($A \rightarrow B$, $B \rightarrow A$)으로 가능하나 동시에 양 방향 통신불가(위키토키, RS-485)
- 전이중(Full-Duplex) : 데이터를 동시에 양 방향($A \leftrightarrow B$)으로 전송이 가능(전화, RS-232C)

5, 통신 프로토콜

- 동기식 : USB, CAN, SPI, I2C, Microwire, USART
- 비동기식 : RS232, RS422, RS485, UART/USART

UART/USART

1, UART통신

- 데이터 신호만 필요 따라서 데이터를 고정 속도로 전송할 필요가 없다
- 일반적으로 한 번에 한 바이트(보통 8bit)씩 전송
- 특정값으로 전송 : 4800, 9600, 38400, 115200(BPS)
- 데이터만 보내기 때문에 속도가 빠르지만 데이터가 어디서 시작(=시작 비트(Start Bits) 어디서 끝인지(=정비 비트))를 구분
 - 장점
 - 간단한 구현
 - 범용성
 - 비동기 통신 : 데이터 전송에 정확한 타이밍이 필요하지 않다
 - 저렴한 비용
 - 단점
 - 속도 제한 : 상대적으로 느린 전송 속도를 가진다.
 - 전송 거리 제한: 짧은 거리 통신에 적합
 - 노이즈 민감성
 - 다중 장치 연결의 어려움
 - 전압 호환성

2, USART통신

- 데이터 동기화를 위해 별도의 클럭 신호를 전송
- 데이터를 구분할 필요없이 클럭의 유무만 체크하면 되므로 데이터 송수신 효율이 높다는 장점
- 그러나 클럭 핀(Clock Pin)을 추가로 요구하여 구조적 어려움
- 다른 모든 요소가 일정하게 유지되는 경우 비동기 모드보다 높은 DTR(데이터 전송 속도)를 가진다.

○ 장점

- 동기(Synchronous), 비동기(Asynchronous) 통신 모드를 모두 지원
- 고속 통신
- 다중 프로토콜 지원
- 하드웨어 플로 컨트롤
- 다중 장치 연결

○ 단점

- 복잡한 구성
- 전원 및 접지 문제
- 전송 거리 제한 : 근거리 통신
- 별도의 하드웨어 필요

3, UART와 USART의 차이점

- Clock(클럭) : 타이밍 신호/동기 신호, 외부 신호로 처리 내부적으로 펄스 발생

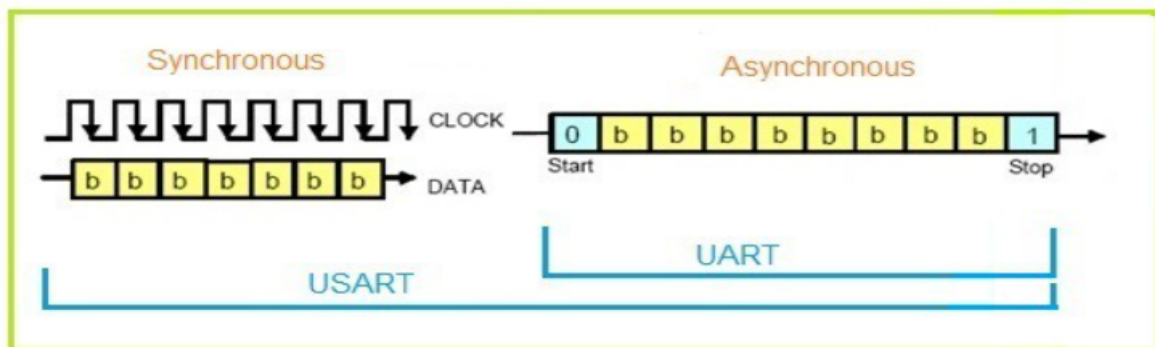
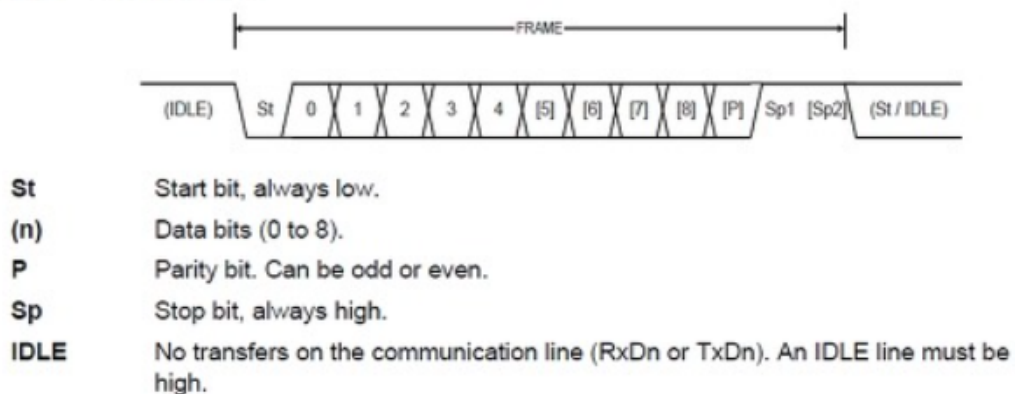
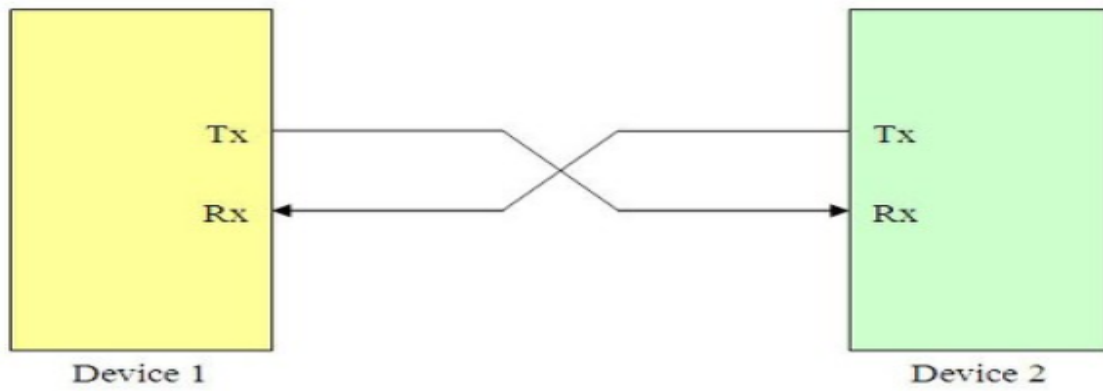


Figure 20-4. Frame Formats



4, 연결 방법



- TX(송신)에는 RX(수신)가, RX에는 TX가 연결되어야 한다.
- 마찬가지로 GND도 서로 연결해주어야 같은 전압 레벨을 갖게 된다.