

<통신 조사>

● 용어 정리

- 1) 동기(Synchronous) 통신: 요청과 그에 따른 결과가 동시에 일어나는 통신
- 2) 비동기(Asynchronous) 통신: 요청에 따른 결과가 그 자리에서 동시에 일어나지 않는 통신

동기(Synchronous) 통신	비동기(Asynchronous) 통신
- 요청을 보내면 응답을 얻기 전까지 Block 상태가 됨	- 응답을 기다리지 않기에 Non Block 상태
- 이로 인해 뒤의 요청이 연결을 맺지 못함	- 순서를 보장하지 않음
- But. 순서를 보장해 요청에 대한 결과가 변경되는 사항에서 중요	- 동기에 비해 성능이 좋음
	- But. 처리 결과를 보장받고 처리해야 하는 서비스에는 적합하지 않음

3) 패리티 비트(Parity Bit)

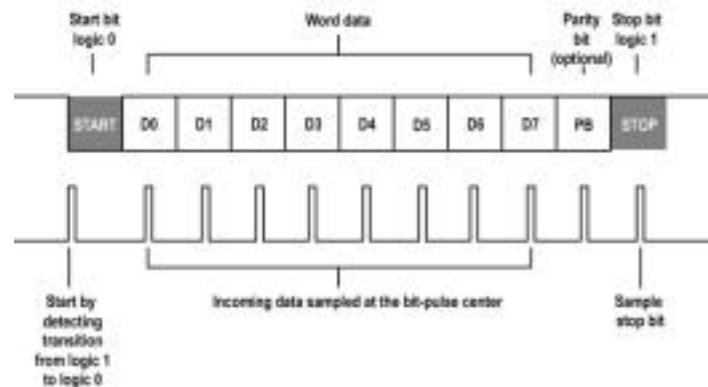
시리얼 통신에서 송수신되는 데이터의 오류를 검출하기 위해 사용되는 비트로 일종의 오류 식별자 역할을 한다. 전송하고자 하는 데이터의 끝에 1비트를 더하여 전송하는 방법으로 2가지 종류의 패리티 비트(홀수, 짝수)가 있다.



● 내용 정리

1. UART(Universal Asynchronous Receiver/Transmitter)

- 병렬 데이터의 형태를 직렬 방식으로 전환하여 데이터를 전환하는 컴퓨터 하드웨어 중 하나로 데이터 전송을 위한 비동기 통신 프로토콜이다.
- 일반적으로 EIA RS-232, RS-422, RS-485와 같은 통신 표준과 함께 사용
- 비동기 통신이기에 동기 신호가 전달되지 않기에 수신에서 동기 신호를 찾아 데이터의 시작과 끝을 시간적으로 알아서 처리할 수 있도록 한다
- TX(송신), RX(수신)으로 구성된 주로 2개의 핀을 사용(RTS/CTS 등의 핀을 사용하기도 함)
- 송신 측에서 데이터를 직렬로 전송하고, 수신 측에서 이를 병렬로 변환해 처리
- Baud Rate(보율): TX와 RX에서 데이터를 보내는 속도 -> 통신 양쪽에서 미리 설정(거의 동일하며 일반적으로 9600bps, 최대 115200bps)
- 바이트 단위 통신을 주로 사용, 시작비트(start bit, '0')와 정지 비트(stop bit, '1')을 이용해 데이터의 시작을 파악하며 이들을 추가한 10비트 데이터를 전송하는 것이 일반적이다.
- UART 통신은 전이중 방식(full duplex) 통신으로 송신과 수신을 동시에 진행할 수 있으며 이를 위해 2개의 입출력 핀이 필요(TX, RX)

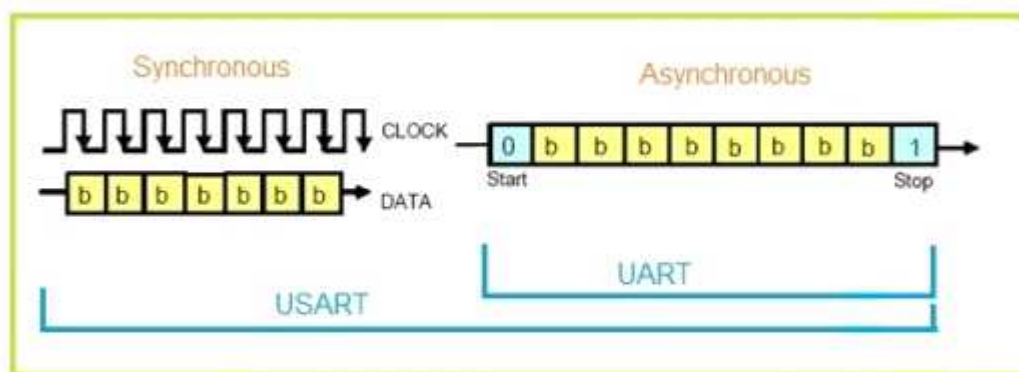


2. USART(Universal Synchronous/Asynchronous Receiver/Transmitter)

- UART의 기능을 확장한 형태로 비동기 통신뿐만 아니라 동기 통신도 지원함
- 동기 통신에서는 송신자와 수신자가 동일한 클럭 신호를 공유하기에 데이터 전송이 더 빠르고 안정적일 수 있다.
- USART 모듈을 UART의 TX, RX 핀 외에도 동기 모드에서 사용하는 클럭신호(SCK) 핀이 추가됨
- 다양한 통신 방식과 전송 속도를 지원하기에 다양한 응용 분야에 사용될 수 있다.

<UART VS USART>

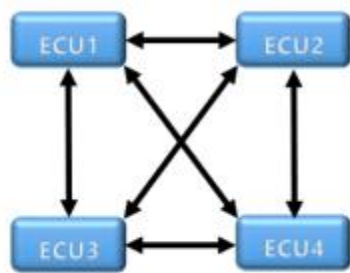
	UART	USART
통신 방식	비동기 통신	비동기 통신 + 동기 통신
전송 속도	클럭 공유가 없어 상대적으로 낮은 전송 속도	클럭을 공유해 더 높은 전송 속도
사용 범위	간단한 직렬 통신에 주로 사용	더 복잡한 통신 요구 사항이 있는 시스템에 사용



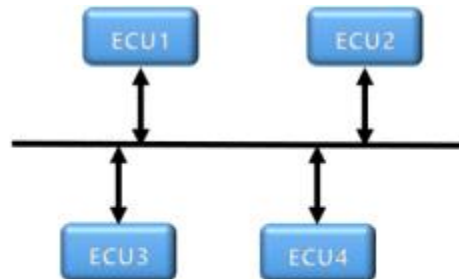
3. CAN(Controller Area Network)

- 차량 내에서 전자 제어 장치들 간에 데이터를 교환하기 위해 개발된 직렬 통신 프로토콜
- 현재 차량의 다양한 컴퓨터 시스템 간의 통신을 위해 사용됨
- 여러 전자 제어 장치들 간에 데이터를 교환하게 해주는 분산 네트워크이다

- 비동기 통신 방식을 사용해 데이터 전송에 대해 송신기와 수신기 간에 별도의 클럭 신호가 필요하지 않는다
- 우선순위 기반 메시지 전송으로 각 메시지는 고유한 식별자(ID)를 가지며 이를 통해 우선순위를 결정한다
- 높은 신뢰성을 요구하는 환경에서 사용되기에 다양한 오류 검출 매커니즘이 존재한다. CRC와 같은 기술을 통해 데이터 전송 과정에서 발생하는 오류를 검출하고 수정 가능



<그림 2> UART 통신 방법



<그림 3> CAN 통신 방법