

1. 번호 뒷자리 4개 STM32 – UART 오실로스코프 파형 분석

Chr	4	6	0	1
Hex	0x34	0x36	0x30	0x31
Bin	0011 0100	0011 0110	0011 0000	0011 0001
Bin(LSB)	0010 1100	0110 1100	0000 1100	1000 1100

표1. "4601" 예상 파형 분석

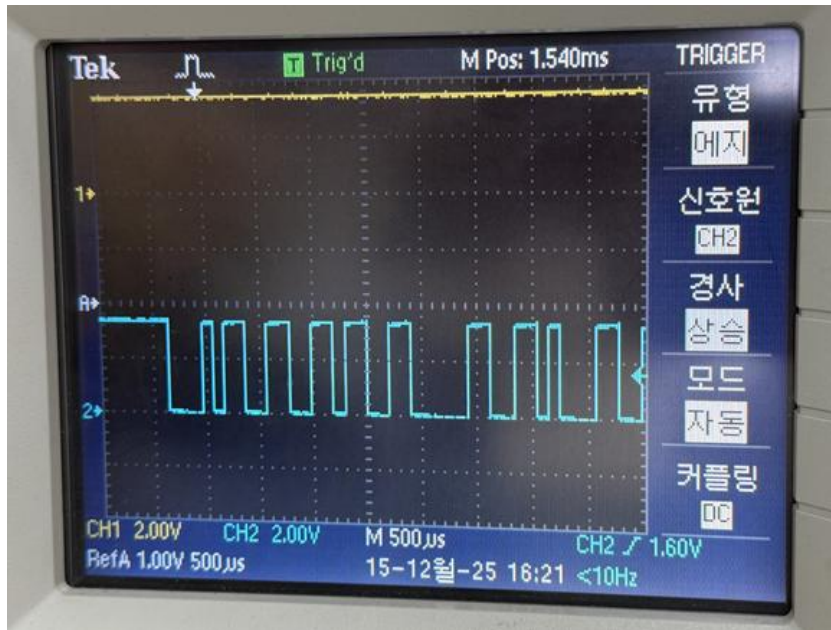


그림1. 오실로스코프 측정 파형

Baudrate	9600 bps
Start bit	1 bit
Data bits	8 bit
Stop bit	1 bit
Idle State	HIGH(3.3V)

표2. STM32 UART 설정

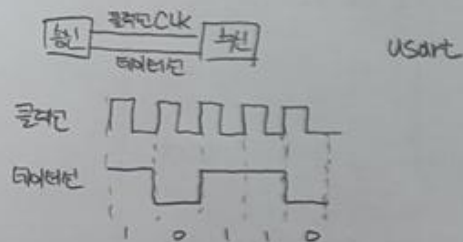
2. 동기식 & 비동기식 차이점

* 동기 (Synchronous) 방식

- 한편의 단위가 아니라 미리 정해진 주기를 타이밍을 한 목적으로 만들어서 한 번에 전송한다.

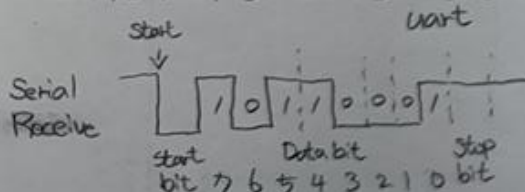
전송 단위를 프레임이라고 한다.

- 데이터와는 별도로 송신측과 수신측이 하나의 기준 clock으로 동기신호를 맞춰 동작한다.



* 비동기 (Asynchronous) 방식

- 송신측과 수신측이 clock에 관계없이 시간을 조율해서 서로 송신 속도를 Baud rate를 정해서 맞춘다.
- 데이터 줄이, 시작, 정지 비트를 포함하여 보내는 방식.



	가장/최소	크기	데이터 전송 단위
* 동기	↑/확정	0	블록
비동기	↓/변화	x	string

3. 빅 엔디언 & 리틀 엔디언 차이점

* 빅 엔디언 (Big Endian)

- 가장 큰 자리 바이트 (MSB)를 낮은 주소에 저장
- 상위 바이트, 큰쪽을 먼저 저장

* 리틀 엔디언 (Little Endian)

- 가장 작은 자리 바이트 (LSB)를 낮은 주소에 저장
- 하위 바이트, 작은쪽을 먼저 저장
- x86, ARM

Input	→ 빅 엔디언	0x01	0x02	0x03	0x04
0x01020304	→ 리틀 엔디언	0x04	0x03	0x02	0x01
Address		A1	A2	A3	A4

4. 인터럽트 & 폴링 차이점

* 인터럽트 (Interrupt)

- 외부 인터럽트 신호 (이벤트)가 발생하면 CPU가 작업 일을 멈추고 ISR을 실행

장점: CPU 효율 ↑, 응답 속도 ↑

단점: 설계가 복잡

* 폴링 (Polling)

- CPU가 주기적으로 상태를 확인해서 이벤트가 발생했는지 판단

장점: 구현 단순, 직관성

단점: CPU 낭비, 응답 속도 ↓

인터럽트: 하드웨어가 CPU를

폴링: CPU가 계속 감지