

18장. RS232, CAN 신호 측정 및 분석



1. 학습개요

목표

1. RS232 및 CAN 시리얼 통신의 기본 구조와 신호 특성을 이해한다.
2. 오실로스코프를 이용해 RS232, CAN 신호를 안정적으로 트리거링하는 방법을 익힌다.
3. 디코딩(Decode) 기능을 활용해 실제 데이터를 해석하고 통신 오류를 분석한다.
4. RS232와 CAN 통신에서 자주 발생하는 오류 유형과 디버깅 포인트를 이해한다.

2. 이론

오실로스코프는 단순히 전압 파형을 확인하는 장비를 넘어, 디지털 통신 신호를 시간 축에서 직접 관찰하고 해석할 수 있는 강력한 디버깅 도구이다. RS232, CAN과 같은 시리얼 통신 신호는 파형만 놓고 보면 단순한 디지털 펄스처럼 보이지만, 실제로는 프레임 구조, 타이밍, 오류 검출 규칙이 모두 포함된 신호이다. 범용 오실로스코프의 트리거와 디코딩 기능을 활용하면, 논리 분석기 없이도 이러한 통신 신호의 동작 상태와 오류 원인을 충분히 분석할 수 있다.

2.1 RS232 (시리얼 통신) 신호의 이해



그림 18-1. RS232 신호의 예시

A. RS232 개요

RS232는 컴퓨터와 외부 장치 간의 통신에 사용되는 대표적인 비동기식 시리얼 통신 규격이다. 클럭 신호 없이 데이터 자체의 타이밍만으로 통신을 수행하며, 산업용 장비, 계측기, 자동화 설비 등에서 지금도 널리 사용되고 있다. 비동기 통신 방식이기 때문에, 송신 측과 수신 측은 정해진 전송 속도(Baud Rate)와 프레임 구조를 사전에 공유해야 정상적인 통신이 가능하다.

B. 전기적 특성

- 전압 레벨: -15V ~ +15V
- 논리 레벨: Logic 1은 Low, 음전압을 의미하고 Logic 0은 High, 양전압을 의미한다.

이는 일반적인 TTL/CMOS(UART)와 논리 정의가 반대이므로, 측정 시 기준 전압과 극성을 반드시 확인해야 한다.

C. 프레임 구조

RS232는 클럭이 없는 비동기 통신이며, 데이터는 프레임 단위로 전송된다.

예시) 9600 bps, 8N1 -> Start (1) + Data (8) + Stop (1)

구성 요소	설명
Start Bit	통신 시작 알림 (Low 레벨)
Data Bit	일반적으로 7~8bit
Parity	오류 검출용 (None / Even / Odd)
Stop Bit	프레임 종료 (High 레벨)

이 중 Start Bits는 프레임의 기준점 역할을 하며, 수신 측은 이 비트를 기준으로 전체 데이터 타이밍을 판단한다.

2.2 RS232 신호 측정 및 디코딩 원리

A. 왜 RS232 트리거가 중요한가?

RS232 신호는 주기적으로 반복되는 파형이 아니라, 비주기적(Non-periodic) 데이터가 연속적으로 흘러가는 구조를 가진다. 이러한 신호를 단순한 에지 트리거로 측정할 경우, 데이터 비트 중간의 에지에서도 트리거가 발생해 프레임이 흔들리거나 끊겨 보이는 현상이 발생할 수 있다. 따라서 RS232 신호를 안정적으로 관찰하기 위해서는 Start Bit를 기준으로 트리거링하고, 사전에 설정된 Baud Rate에 맞춰 타이밍을 해석하는 RS232 전용 트리거 설정이 필요하다. 이 과정을 통해 오실로스코프는 임의의 에지가 아닌 의미 있는 데이터 프레임의 시작점에 동기화된다.

B. 디코딩의 역할

디코딩 기능을 사용하면, 단순한 전압 파형을 비트->바이트->문자 데이터로 변환하여 확인할 수 있다. 이를 통해 다음과 같은 사항을 직관적으로 분석할 수 있다.

- Start Bit 누락 여부
- 프레임 길이 오류
- 패리티 오류 발생 여부
- 실제 전송된 문자열 (ASCII 데이터)

즉, 디코딩 기능은 보이는 파형과 의미 있는 데이터를 동시에 분석할 수 있게 해주며, 통신 디버깅 과정에서 매우 강력한 도구가 된다.

2.3 CAN(Controller Area Network) 신호의 이해



그림 18-2. CAN 신호의 예시

A. CAN 개요

CAN은 자동차 및 산업용 제어 시스템에서 널리 사용되는 고신뢰성 멀티마스터 통신 프로토콜이다. CAN 통신의 주요 특징은 다음과 같다.

- 다수의 ECU가 하나의 버스를 공유
- 주소 기반이 아닌 메시지 ID 기반 통신
- 충돌 및 오류를 고려한 강력한 오류 검출 구조 (CRC, ACK)

이러한 구조 덕분에, CAN은 노이즈가 많은 환경에서도 안정적인 통신이 가능하다.

B. 물리적 특성

CAN 신호는 차동 신호(Differential Signal) 방식으로 전송된다. 차동 방식은 외부 노이즈에 강하며, 두 신호 간의 전압 차이를 기준으로 데이터를 판단한다.

C. CAN 프레임 구조

필드	설명
SOF	Start of Frame
ID	메시지 식별자
DLC	데이터 길이
Data	실제 데이터
CRC	오류 검출
ACK	수신 확인
EOF	End of Frame

3. 기본 실습

준비물: 오실로스코프, DS6000보드, 패시브 프로브

3.1 RS232 신호 측정 및 디코딩 실습



그림 18-3. DS6000 데모보드에서의 RS232 신호

- ① 데모보드에서 나오는 RS232_TX 신호를 CH1에 연결합니다.
- ② 프로브의 GND를 데모보드의 GND 핀에 연결합니다.
- ③ AUTO를 눌러 오실로스코프에 정상적인 파형이 들어오는지 확인합니다.
- ④ 이제 Trigger 버튼을 눌러 RS232 트리거를 설정합니다. 트리거 설정값은 Trigger Type: RS232, Source:CH1, Level: 50%, Polarity: Negative, Baud Rate:9.6kbps Data Bits:8, Stop Bit: 1, Parity: None, When: Start로 설정해 줍니다. 이 내용은 반드시 실제 시스템의 통신 조건과 동일하게 설정해야 합니다.
- ⑤ 이제 Decode 버튼을 눌러 Decode 값을 설정합니다. Bus Type은 RS232로, Bus Status: On, Copy Trig:On (Copy Trigger를 누르면 트리거에 설정한 내용을 복사하여 불러와집니다.), Format: ASCII, Event Table:On을 누릅니다.
- ⑥ 화면 상단에 문자열이 표시되고 Event Table에서 데이터 흐름이 어떻게 되는지 확인할 수 있습니다.

3.2 RS232 신호 측정 및 디코딩 실습

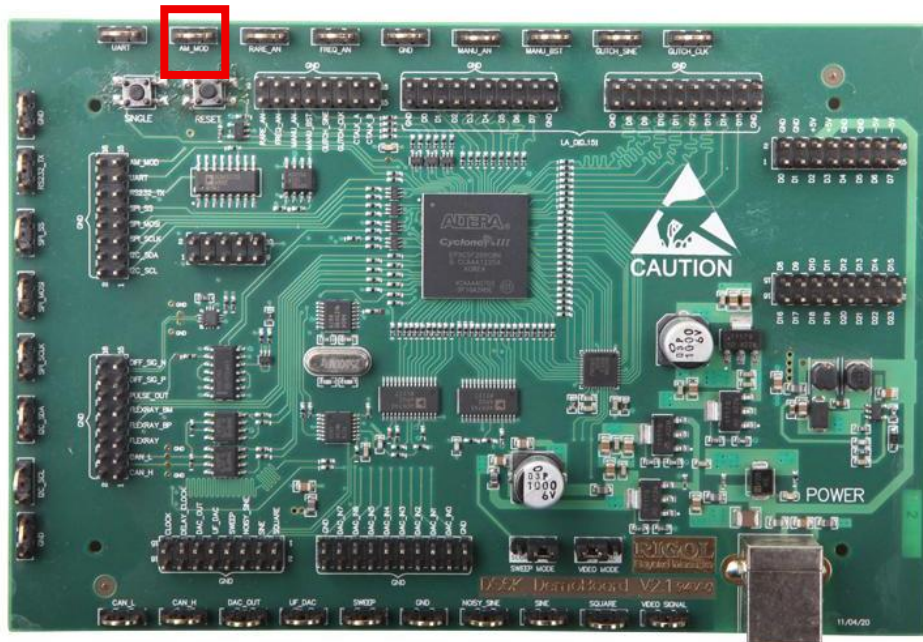


그림 18-4. DS6000 데모보드에서의 CAN_H 신호

- ① 데모보드에서 나오는 CAN_H 신호를 CH1에 연결합니다. (단일 채널 측정 기준으로 측정합니다.)
- ② 프로브의 GND를 데모보드의 GND 핀에 연결합니다.
- ③ AUTO를 눌러 오실로스코프에 정상적인 파형이 들어오는지 확인합니다.
- ④ 이제 Trigger 버튼을 눌러 CAN 트리거를 설정합니다. 트리거 설정값은 Trigger Type: CAN, Source: CH1, Signal Type: CAN_H, Condition: SOF, Baud Rate: 1 Mb/s, Mode는 Auto로 설정합니다.
- ⑤ 이제 Decode 버튼을 눌러 Decode 값을 설정합니다. Bus Type은 CAN, Bus Status: ON, Copy Trig: ON, Data Format: ASCII, Event Table: ON을 설정합니다.
- ⑥ 화면 상단에 RIGOL이 표시되고 Event Table에서 데이터 흐름이 어떻게 되는지 확인할 수 있습니다.

4. 결과 확인

- RS232의 캡처 파형과 디코딩 문자열에 대한 이벤트 테이블을 캡처합니다.
- CAN의 디코딩된 ID/DATA를 확인하고 이벤트 테이블이 포함되도록 캡처합니다.

5. 확인 문제

1. RS232에서 Start Bit의 역할은 무엇인가요?

2. CRC 통신에서 CRC 오류가 발생하면 어떤 문제가 의심되나요?

6. 핵심요약

- RS232는 비동기 시리얼 통신, CAN은 차동 기반 고신뢰 통신이다.
- 전용 트리거 설정이 안정적인 디코딩의 핵심이다.
- 디코딩 기능은 단순 파형 관찰을 넘어 실제 데이터 검증 도구로 사용된다.
- 오실로스코프 하나로 신호 무결성과 프로토콜 분석을 동시에 수행할 수 있다.

7. 참고 및 부록

용어집:

- Baud Rate: 초당 전송 비트 수
- Start Bit: 프레임 시작 알림
- CRC: 오류 검출 코드
- ACK: 데이터 수신 확인