

CAN 과 RS485 비교

개요

CAN (Controller Area Network)과 RS485 는 필드버스시스템의 대표적인 표준들입니다. RS485 는 단지 계층 1(물리적계층) 만을 어드레스하고 CAN 은 OSI 모델에서 계층 2(데이터연결계층)도 추가하기 때문에, 두 가지 표준들을 비교하는 것은 어려운 일입니다. 그럼에도 불구하고, 이 글은 사용자의 시스템에 CAN 을 도입 할 것인가 말 것인가 하는 것을 결정하는데 도움을 주기 위하여 이 두 가지 표준들을 비교하고자 합니다.

RS485

최소 하나의 UART 와 더불어 장치들의 multi-drop 통신기능을 제공하는 RS485 표준은 1983 년에 만들어졌습니다. RS485 버스에 연결된 장치는 반이중(half-duplex) 모드에서 다른 디바이스와 통신이 가능합니다. 비록 반이중(half-duplex)이 양방향 통신이 가능하다 할지라도, 단 한 개의 장치만 한번에 전송할 수 있으며 다른 장치들은 수신만 해야 합니다.

여기서, RS485 는 프로토콜이 아닙니다. 이것은 단지 multi-drop 버스를 이용하여 직렬 메시지들의 전송을 가능하게 하며, 데이터 교환을 위한 물리적 연결과 기본 원칙만을 제공하고, 메시지 내용은 전적으로 사용자가 정의합니다.

이것은 또한, 통신프레임구조, 다른 노드들을 어드레스 하는 구조, 데이터 충돌을 피하기 위한 체계와 기타 작업들이 프로토콜 소프트웨어 형태로 개발자에 의해 구현 되어야만 합니다.

CAN

Controller Area Network (CAN)은 1980 년대에 독일 Robert Bosch 사에 의해 만들어졌습니다. 초창기에 CAN 은 주로 자동차 분야를 대상으로 한 것 이었으나, 현재는 산업용자동차, 의료, 교통 등 다양한 애플리케이션들에서 사용되고 있습니다. RS485 와 비교하여, CAN 은 통신을 위한 물리적 매체를 제공하는 것뿐만 아니라, 데이터패킷(메시지들)들의 어드레스, 데이터 충돌을 방지, 전송된 데이터에서의 결함 발견, 분산된 메시지들의 자동반복과 한 네트워크의 전체 노드들에 대한 데이터 일관성 확보를 위해 필요한 모든 기타 체계들을 제공합니다. 또한, CAN 은 메시지 식별자, 데이터와 제어 바이트들로 데이터 프레임의 구조를 지정합니다.

RS485 처럼, CAN 노드는 반이중(half-duplex) 모드에서 다른 노드와 통신할 수 있습니다. 메시지는 원래 식별자로 구성 되는데, 이는 충돌 방지 측면에서 메시지의 우선순위를 나타내는 것으로, 최대 8 데이터바이트입니다. 이것은 버스를 통해 연속적으로 전송됩니다. 이 신호패턴은 non-return-to-zero (NRZ)로 암호화 되며 모든 노드에 의해 판독됩니다.

버스가 자유롭다면, 어떤 노드라도 전송을 시작할 수 있습니다. 메시지 전송을 시작하는 노드가 동시에 두 개 이상이라면, 보다 우선적인 ID (우성 인비트를 갖는, 즉, zero)에 의해 정의된, 높은

우선순위를 갖는 메시지가 다른 노드들의 하위 ID 들을 overwrite 합니다. 이것은 확실하게 오직 한 개의 우성 메시지가 전송 되어 다른 모든 노드들에 의해 수신될 수 있게 합니다. 이러한 체계는 우선순위-원칙버스중재 (priority-based bus arbitration)로 일컬어집니다. 숫자적으로 보다 적은 값의 ID 를 가진 메시지가 더 높은 우선순위를 갖게 되며, 두 개의 메시지가 동시에 전송 되는 경우에는 먼저 전송됩니다.

비교

RS485 는 원래는 UART 모듈을 사용하기 위한 것입니다. 따라서 통신은 8 비트 레지스터를 통해 읽거나 쓰여지는 개별적인 character 를 바탕으로 이루어집니다. 그러므로 통신의 구성은 소프트웨어에서 실행 되어야만 합니다. 반면, 고정된 사전-설정 통신 형태를 지닌 CAN 기능들은, 0 부터 8 데이터바이트를 이용하는 것뿐만 아니라, 어드레싱과 데이터 일관성 (CRC 검사)을 위한 제어정보도 포함하고 있습니다. 사용자는 그저 메시지 식별자와 Payload 만을 전송하면 되고, 나머지는 이른바 CAN 컨트롤러에 의해 추가됩니다. CAN 컨트롤러는 외부칩이 될 수도 있고 또는 마이크로컨트롤러 안의 내부 모듈이 될 수도 있습니다.

각각의 CAN 메시지는 메시지 식별자에서 파생된 고유의 우선순위를 갖고 있습니다. 높은 우선순위를 가진 CAN 메시지들은 CAN 컨트롤러칩에 의해 우선적으로 처리될 수 있으며, 따라서 낮은 순위의 메시지들 보다 먼저 전송됩니다. 또한 CAN 버스의 노드들 사이에서, 가장 높은 우선순위를 가진 메시지가 먼저 전송됩니다.

결과적으로, CAN 은 실시간 솔루션이 되었습니다. 메시지의 우선순위에 따라 이것의 ID 덕분에, 최대 대기시간이 각각의 CAN 메시지에 대해 버스로드 혹은 다른 노드의 실행과는 무관하게 예측 가능해졌습니다. 이와는 대조적으로 RS485 는 메시지 충돌을 트리거할 수 없으며, 따라서 애플리케이션 소프트웨어에서 충돌 방지가 보장 되어야 합니다. 일반적으로 RS485 네트워크에서의 충돌 방지는 마스터가 모든 슬레이브들을 순차적으로 Polling 하는 master/slave 관계로 수행됩니다. 대기시간은 모든 노드들의 수와 반응시간에 따라 도출 되기 때문에 훨씬 더 길어집니다.

Message-wise 중재로 인하여, CAN 은 추가적인 사전 대책 없이도 멀티-마스터 작동이 가능합니다. RS485 의 경우에는, 이것은 오직 특정 프로토콜로만 가능한데, 소프트웨어에서 구현 되어야만 하는 token ring 같은 것입니다.

CAN 은 매우 발전된 오류관리 방법을 갖고 있습니다. 메시지가 노드에 의해 부정확하게 수신된다면 (CRC 또는 포맷오류), 통신은 오류프레임을 통해 수신자에 의해 무시되고, 따라서 모든 노드들에 대해 무효(invalid)한 것으로 표기됩니다. 이러한 행동은 전송처리를 반복하도록 CAN 컨트롤러에서 자동반응을 시작합니다. 이러한 처리과정은 모든 노드들이 오직 유효한 메시지를 수신할 수 있게 해줍니다. 부정확한 데이터를 반복적으로 전송하는 노드들 또는 처음에 수신된 데이터가 결함이 있는 것으로 인식된 노드들은 자동적으로 CAN 버스에서 참여가

불가능하게 됩니다. 이는 결함이 있는 CAN 컨트롤러 또는 버스연결들이 데이터트래픽을 영구적으로 두절 시키는 것을 방지합니다.

이렇게 축적된 방식들 (간략한 메시지, 서로 다른 전송, 오류검색과 문제해결, 결함 노드들의 추출)은 CAN 을 매우 안전하며 신뢰성 있는 네트워크로 만들었으며, 이것이 차량, 선박, 엘리베이터, 의료장비들, 항공기와 산업용 생산 설비들의 많은 중요한 안전 관련 애플리케이션들에서 CAN 이 채택 되고 있는 이유입니다.

오늘날 거의 모든 마이크로컨트롤러 계열들은 CAN 컨트롤러가 통합된 버전을 사실상 무료로 제공하고 있습니다. 일반적으로 내부 UART 와 작동하는 RS485 역시 무료입니다. CAN 용 트랜시버 모듈의 비용은 RS485 보다 매우 저렴합니다.

하드웨어에서 CAN 프로토콜의 완벽한 구현으로 인해, 마이크로컨트롤러 시스템에서의 로드는 감소 되었습니다 (메시지당 단 한 개의 인터럽트). RS485 를 이용하면, 소프트웨어에서 프로토콜의 처리 (데이터 일관성체계 포함)외에, 전형적으로 수신된 character 당 한 개의 인터럽트가 있으며, 이는 RS485 의 상당한 CPU 로드를 야기할 수 있습니다. 애플리케이션의 필수 요건과 더불어 프로토콜과 데이터 전송 속도에 따라 RS485 는 더욱 강력하고, 따라서 더욱 값비싼 CPU 를 필요로 합니다. 훨씬 낮은 전송속도는 CPU 의 성능과 그 결과로 비용에도 상당한 영향을 미칠 수 있습니다.

CAN 컨트롤러는 메시지 식별자를 기초로 수신된 메시지들을 여과할 수 있습니다. 따라서 CPU 로드는 훨씬 감소되는데, 노드와 관련된 메시지들만을 수신하기 때문입니다.

CAN 을 선택하는 이유

CAN 을 사용함으로써, 마이크로컨트롤러의 UART 인터페이스는 자유롭게 유지되어 디버깅이나 다른용도(printf)에 사용될 수 있습니다.

CAN 의 수준 높은 표준화로 인하여, 소프트웨어 드라이버를 포함한 PC/CAN 인터페이스 (USB, PCIe...)들을 저렴한 비용으로 이용할 수 있게 되어 PC 애플리케이션들을 위한 CAN 의 간편한 액세스가 가능합니다.

많은 CAN 분석용 소프트웨어, 시뮬레이션, 그리고 기타 도구들을 PC 용으로 이용할 수 있으며, 이러한 것들은 계층 1 부터 계층 7 까지 액세스를 지원하여 개발과 문제해결이 간단해졌습니다.

CANopen, DeviceNet 과 J1939 같은 다양한 계층 7 프로토콜들을 CAN 의 많은 공급업체들로부터 이용할 수 있습니다. 거의 모든 마이크로컨트롤러들에 대해 검증된 계층 7 소프트웨어를 사용할 수 있게 됨에 따라, 설계자들은 위험을 줄이고, 개발된 제품의 시장 출시를 더욱 앞당기며, 개발비용을 더욱 낮출 수 있습니다.

CANopen 과 DeviceNet layer 7 프로토콜에 대한 I/O 노드들과 Sensor, 드라이버들을 시중에서 이용할 수 있습니다. 또한 이러한 프로토콜들은 표준화 되어 있기 때문에, CAN 버스에서 작동하는 모든 노드들이 함께 통신하는 것을 보증하는 디바이스 표준들과 이용할 수 있습니다. RS485 에서 CAN 표준으로 업그레이드 할 때 중요한 것은 기존의 RS485 wiring 과 cable harness 가 CAN 표준의 한계를 초과하지 않고 너무 길지 않다면 이것들을 이용하는 것 입니다. 그러나 한계를 초과하는 경우에도, 저렴한 CAN repeater 또는 bridge 가 CAN 네트워크의 길이를 연장하는데 사용될 수 있습니다.

요약

Feature	RS485	CAN
Required interface	UART	CAN controller
Supported ISO model layers	Physical layer	Physical layer and data link layer
Detection of data collisions	Not implemented	Yes, CSMA/CR
Maximum transmission rate	10 Mbit/s (up to 12 m)	1 Mbit/s (up to 50 m)
Maximum bus length	1200 m (at 100 kbit/s)	1600 m (at 50 kbit/s)
Supported Bus arbitration principles	Master/Slave or Token Ring	Multimaster and all principles, which can be derived from that like Master/Slave or Token Ring
Maximum data amount per frame	Unlimited	8 bytes
Bus transceiver costs	~ 70 ¢	~ 35 ¢
Examples of popular protocols	Modbus RS485, Profibus	CANopen, DeviceNet, J1939

새로운 개발을 위해, 실시간 동작, 통신 안정성, 검증된 계층 7 통신 프로토콜 스택과 톨의 성능측면에서, CAN 표준은 독보적입니다. 또한 현재 CAN 의 구현은 RS485 보다 비용이 저렴합니다.

위의 사항들이 그 역할을 하지 못하는, 순수 master/slave 시스템의 경우, 매우 많은 수의 노드들을 어드레스 하거나 또는 대량의 데이터를 전송해야 하지만 그것이 CAN 통신에 의해 제한될 수 있다면, RS485 구현이 장점이 될 수도 있습니다.