

Controller Area Network 개요

오늘날, 실질적으로 모든 마이크로컨트롤러 제조업체들은 CAN 인터페이스를 통합한 솔루션을 제공합니다. 또한, 필수요건들과 애플리케이션의 확장된 영역들과 연관하여, 통합된 CAN 컨트롤러의 성능과 구조의 최적화가 계속되고 있으며 기능도 확장되고 있습니다. 최초의 CAN 프로토콜 컨트롤러 칩이 나온 후 20여 년이 지난 지금도, CAN은 그 사용이 계속 증가되고 있습니다. 자동차와 상업용 차량에서의 사용과, 모든 종류의 이동 시스템 (예. 대중 교통 차량, 엘리베이터, 선박, 기차, 특수 목적 차량, 비행기 등등)에서의 사용 외에도, CAN은 산업 자동화 기술과 다양한 애플리케이션 분야에서의 수많은 내장 시스템에서 - 커피 기계에서부터 신장 결석 쇄석기에 이르기까지 - 사용이 늘어나고 있습니다. CANopen, DeviceNet 과 SAE J1939 같은 표준화된 상위 프로토콜들과 프로파일들을 바탕으로, 분산된 애플리케이션의 구현을 위한 일체의 솔루션을 시스템 구현에 이용할 수 있습니다. 또한, 표준화된 통신 체계, 식별자 할당, 네트워크 관리 기능들과 수많은 디바이스와 애플리케이션 프로파일들은 여러 제조업체 디바이스들의 상호운용성과 교환을 가능하게 합니다.



매우 저렴한 시스템 솔루션의 구현 외에도, CAN-기반 네트워크는 다른 시스템 솔루션들과 비교했을 때 일련의 특별한 기능들을 갖고 있습니다. CAN 프로토콜의 주요한 성능 특징들을 다음에 간략히 기술 합니다:

생산자-사용자 원칙에 따른 Controller Area Network 메시지 전송

“노드-지향” 전송과는 달리, 두 개의 특정 노드들 간에 메시지 전송이 발생되는, CAN을 이용한 메시지 전송은 이른바 “producer-consumer principle”을 바탕으로 이루어집니다. 생산자 노드에 의해 전달된 메시지는 모든 다른 사용자 노드들에 의해 수신될 수 있습니다.

이를 위해, 메시지는 목적지 주소 대신 분명한 “메시지 식별자”로 표시됩니다. 네트워크의 모든 노드들에 대한 메시지 전송은 “방송(broadcasting)”으로도 일컬어집니다. CAN 프로토콜에서, 메시지들은 일반적으로 11-비트 길이 식별자 (표준 포맷)를 갖습니다. 이것은 시스템마다 2048개의 서로 다른 메시지들이 정의될 수 있다는 것을 의미합니다. 이 수는 대부분의 애플리케이션들에서는 충분하고도 남습니다. 그러나, 특수한 애플리케이션들의 경우 (예, 대형 차량들, SAE J1939) 29비트 길이 식별자 (확장 포맷) 사용과 따라서 최대 512백만 개의 서로 다른 메시지들 정의도 가능합니다. 노드는 반드시 버스의 메시지 열에서 이것을 위해 관련된 메시지를 여과(filter)해야 합니다. 이렇게 하기 위하여, CAN 프로토콜 컨트롤러 칩은 이른바 수용 필터링 (acceptance filtering)의 형태로 하드웨어에 의해 실행되는 구조를 제공합니다.

CAN의 Multimaster 능력, 사건 지향 메시지 전송

CAN 네트워크의 각 노드는 버스가 자유로워지자마자 메시지 전송을 시작할 수 있습니다. 이것은 하나 이상의 네트워크 노드에서 동시에 메시지 전송이 시작될 수 있기 때문에, 중재 처리가 필수이며, 따라서 실제로는 단 한 개의 노드만이 메시지 전송을 계속할 수 있도록 합니다.

모든 노드가 메시지 전송을 시작할 수 있기 때문에, 네트워크의 모든 노드들 사이에서 직접적인 메시지 전송이 가능합니다. 따라서 요청 시, 즉 관련 사건이 발생했을 때 메시지를 전송만 하는 것이 가능합니다. 주기적 메시지 전송과 비교하면, 요청된 데이터 전송률에서 상당히 낮아진 버스 적재 또는 감소된 결과를 가져옵니다.

손실 없는, bit-wise bus arbitration

CAN 네트워크에서 모든 노드는 버스가 자유로워지면 바로 메시지 전송을 시작할 수 있으므로, 여러 개의 노드들이 동시에 전송처리를 시작하는 것이 가능합니다. 노드가 서로 전송된 데이터를 간섭하는 것을 막기 위해, 동시에 중재되는 모든 메시지를 중 가장 높은 우선 순위를 가진 메시지가 중재 단계에서 결정됩니다. 실제로 가장 높은 우선 순위 메시지를 전송하는 네트워크 노드만이 메시지 전송을 계속할 수 있게 됩니다. 메시지 식별자의 가장 낮은 값을 가진 메시지가 가장 높은 우선 순위를 갖게 됩니다. 아래 그림은 버스 중재에 사용되는 원리를 보여줍니다. 각각의 노드는 중재 단계 동안 버스에서 signal level을 관찰합니다. 이 중재 단계는 메시지 식별자의 전송과 소위 RTR 비트로 불리는 것의 전송으로 이루어져 있습니다. 만약 네트워크 노드가 우성 버스 레벨 (우성 비트)를 발견한다면, 이것은 열성 레벨(열성 비트)로 전환되고, 전송 처리가 즉시 중단되지만, 이 경우 동시에 더 높은 우선 순위를 가진 메시지는 분명히 전송되고 수신 상태로 들어가게 됩니다. 모든 메시지가 버스 중재로 전송되면, 이 과정은 확실한 “손실 없는” 버스 액세스가 됩니다.

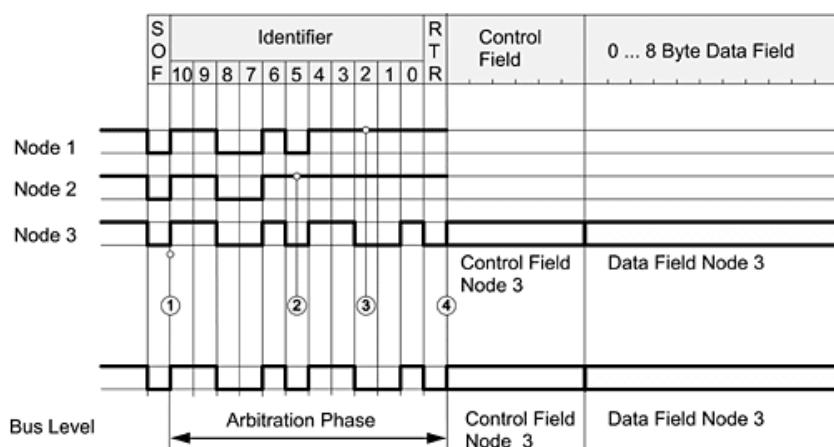


그림: 손실 없는, bit-wise 버스 중재의 원리 - 노드 1, 2 와 3 은 동시에 중재를 시작합니다. 2 지점에서, 노드 2 는 자신의 열성 구동 버스 레벨 대신, 버스가 우성 레벨을 보여주는 것을 탐지하고 따라서 중재를 멈춥니다. 똑같은 일이 3 지점의 노드 1 에서 발생합니다. 중재 구간의 마지막에서는 오직 노드 3 만이 자신의 메시지 전송을 계속합니다.

우선순위 지향 (Priority-oriented) 메시지 전송

위에 설명된 중재 처리는 버스가 자유로워질 경우 언제나 항상 가장 높은 우선 순위의 메시지가 전송된다는 것을 보여줍니다. 메시지의 우선 순위는 메시지 식별자의 값으로 결정됩니다. 이 값이 더 작을수록, 메시지의 우선 순위는 더 높아집니다. 우선 순위 지향 메시지의 원칙은 데이터 전송에 이용할 수 있는 대역폭의 가장 효율적인 사용을 가능하게 합니다. 따라서 낮은 우선 순위의 메시지가 실제로 보다 높은 우선 순위를 가진 메시지를 지연시키는 일 없이 100% 버스를 점령하는 것이 가능합니다. 가장 높은 우선 순위를 가진 메시지의 경우, 1 Mbit/s의 전송 속도에서 최대 약 130 μ s의 대기 시간이 일어납니다. 반면, CAN 시스템을 설계할 때는, 높은 우선순위 메시지가 버스를 계속 점령하지 않는다는 것이 지켜져야 합니다. 이것은, 예를 들면 전송 금지 시간 (CANopen: "Inhibit Time")을 도입함으로써 가능합니다.

CAN 시스템의 비트율(Bitrate) 과 버스 길이

CAN에서 사용되는 bit-wise 중재 원칙은 bit-time 간격 내에서 버스를 걸쳐 분배되는 전체 네트워크 노드들의 local bit level의 비교를 필요로 합니다. 버스를 걸쳐 신호 분배에 요구되는 신호 전파 시간은 버스 길이에 비례하기 때문에 버스 간격의 필요한 지속시간은 버스 길이의 연장에 따라 늘어납니다. 최대 버스 길이(네트워크 확장)와 최대 비트율(bit rate)은 따라서 역으로 비례합니다. 100m 이상의 버스 길이의 경우, 다음의 경험 법칙이 계산에 사용될 수 있습니다:

Baudrate (in Mbit/s) \times bus length (in m) = 60.

메시지 길이와 최대 메시지 비율

최대 8 바이트의 데이터는 한 개의 CAN 메시지로 전송될 수 있습니다. 이용 가능한 데이터 전송 능력은 자동화 시스템 또는 모터 차량 분야의 거의 모든 애플리케이션들이 사용하기에 충분합니다. 보다 큰 데이터 블록들은, 가령 디바이스 구성을 위한, 일련의 연속 CAN 메시지로 전송될 수 있습니다. 짧은 메시지의 가장 큰 장점은 높은 우선 순위 메시지에 맞는 보다 짧은 대기 시간과 높은 결함을 지닌 환경에서도 전송할 수 있는 능력입니다. 짧은 메시지의 경우 (전송 시간), 두 결함 사이에서 방해 받지 않고 전송될 수 있는 가능성은 보다 긴 메시지들의 전송에서보다 훨씬 높습니다.

1 Mbit/s의 비트율로, 평균 데이터 길이 4 바이트에서는 초당 약 10,000 개의 메시지들이 전송될 수 있으며 8 바이트 데이터 길이(표준 포맷)에서는 초당 약 7,200 개의 메시지가 전송될 수 있습니다.

능률적인 오류 검출과 오류 차단

CAN 프로토콜의 가장 눈에 띄는 특징 가운데 하나는 전송 오류를 검출하는데 특히 뛰어난 능력입니다. 따라서 이것은 매우 높은 요구들, 예를 들면 모터 차량에서의 제어 장치들의

네트워킹을 충족시킵니다. 뛰어난 오류 검출 능력은 오류 검출의 서로 다른 측정들의 결합으로 얻어집니다. 이러한 점에서 가장 효과적인 측정들 중 하나는 모든 전역적인 유효 오류들을 검출하는 메시지의 전달자에 의한 버스 레벨의 모니터링입니다. 또한, 모든 메시지 수신자가 정의된 포맷 요소와 CRC 세그먼트에 기초하여 수신된 각 메시지들을 확인합니다. 이러한 방식으로 단지 국소적으로 유효한 오류들이 검출됩니다. 또한 전송 오류 검출 외에도, CAN 프로토콜은 결함 네트워크 노드들을 검출하고 차단하는 체계를 갖고 있습니다. 이는 결함 네트워크 노드들이 계속적으로 메시지 전송을 방해하는 것을 막을 수 있습니다.

메시지 확인응답 (acknowledgement) 대신 오류 신호

노드-지향 전송의 통신 개념과는 대조적으로, 메시지-지향 프로토콜로서의 CAN 은 오류 신호 (Error signaling) 원리를 사용합니다. 각각의 네트워크 노드는 버스에서 전송된 모든 메시지를 오류와 관련하여 검사합니다. 전송 또는 수신 네트워크 노드가 오류를 발견하는 즉시, 오류 메시지 (오류 프레임)를 전송함으로써 모든 다른 노드들에게 이것을 신호로 알립니다. 이것은 일반적으로 우성 비트 순차로, 동일한 극성을 가진 여섯 비트들의 비트 조합을 포함합니다. 모든 네트워크 노드들은 오류 신호를 탐지하고 이미 수신된 메시지의 세그먼트를 취소합니다. 따라서 네트워크 모든 모드들에서 일관된 데이터가 확보됩니다.

전송 노드가 오류 프레임 또는 수신된 것을 전송하자마자, 이것은 즉시 다른 버스 중재 처리로 앞서 전달된 메시지를 다시 전송하려 시도합니다.

오류 신호 체계는 네트워크의 모든 노드를 이용한 메시지 전송의 무-오류(error-free)와 일관성을 확보합니다. 오류 신호는 오류가 검출된 후 즉시 발생하기 때문에, 매우 짧은 오류 복구 시간이 가능합니다.

또한 오류가 발견되었을 때만 버스를 추가적으로 차지하는 것은 추가 버스 적재를 상당히 낮추는 장점을 갖습니다.

CAN 메시지의 형식

아래 그림은 일반적인 CAN 메시지 형식을 보여줍니다. 메시지의 시작은 우성 비트가 이끌고 이어서 11 비트 긴 메시지 식별자가 뒤따르고 데이터 프레임과 데이터 요청 프레임 (remote frame) 사이를 구별하는 추가 비트가 이어집니다.

Remote 프레임으로, 네트워크 노드는 시스템의 다른 노드에 의한 특정 메시지의 전송을 요청할 수 있습니다. 제어 영역은 메시지의 전송 포맷 (표준/확장)과 다음의 데이터 바이트의 수를 지정합니다.

CAN 메시지의 데이터 영역은 0 과 8 데이터 바이트 사이에서 수용할 수 있습니다. 데이터 영역 다음에 15 비트 CRC 세그먼트가 이어집니다. 이 영역은 수신된 메시지를 확인하기 위해 수신기(receiver)에 의해 사용됩니다. 확인응답 영역(acknowledge field)에서는 메시지 송신기가 적어도 한 개의 수신 네트워크 노드로부터 전송된 메시지의 무-오류 수신을 알려주는 확인응답을 기대합니다. 이 확인응답은 오류가 없는 메시지를 수신한 네트워크의 모든 노드들에 의한

확인응답 슬롯에서의 우성 비트 전송으로 주어집니다. 이 확인응답은 전적으로 전송 부분에서 결합 있는 전송 노드 차단에 사용됩니다. 끝으로, 프레임의 마지막 부분은 CAN 메시지의 완전한 무-오류 전송을 나타냅니다.

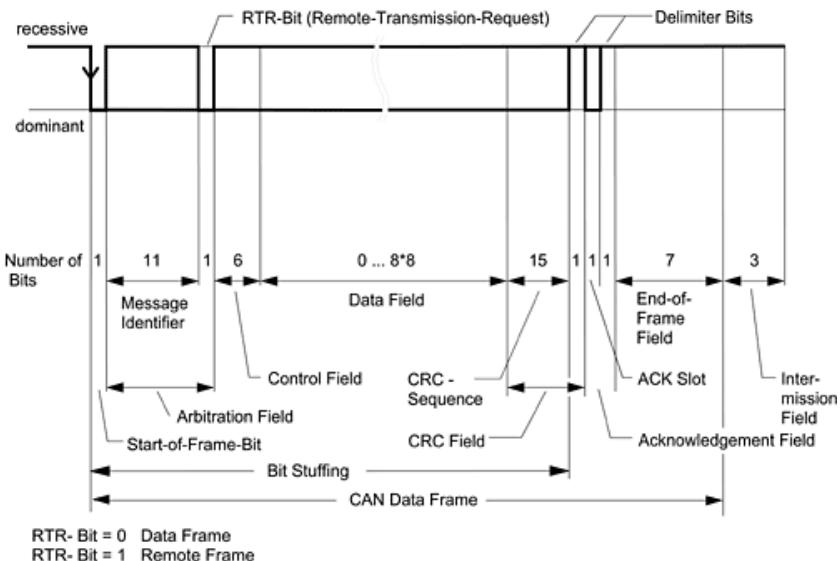


그림: 표준 CAN 메시지 형식 (데이터 프레임). 요청 메시지 (remote frame) 형식은 RTR bit 값과 Empty data 영역에서 차이가 있습니다.

CAN을 위한 Profile과 표준화된 상위 프로토콜들의 이용 가능성

위에서 설명한 ISO 11898에서 표준화된 CAN 프로토콜은 OSI 데이터 통신 모델의 개념에서는 layer 1/2 프로토콜에 해당합니다. 그러나 네트워크의 구현을 위해서는, 더욱 상세한 기능들이 필요합니다. 임베디드 시스템과 산업 자동화의 애플리케이션들을 위해 두 개의 표준들을 이용할 수 있습니다.

CANopen과 DeviceNet. CANopen은 임베디드 네트워크에서의 애플리케이션들을 위한 표준이며, DeviceNet은 특히 Rockwell Automation 환경에서의 산업 자동화에 사용되고 있습니다. 트럭에서의 네트워크에 대해서는 SAE J1939 또한 주요 표준입니다.