Part. 3 Local Interconnect Network

자동차에서 간단하고 경제적인 데이터 교환 (2)



ⓒ Vector Informatik

LIN 버스는 매우 짧은 기간에 자동차에서 간단하고 경제적인 데이터 교환을 위한 기술로 자리매김했다. 오늘날 많은 자동차 OEM들은 LIN을 통해 바디/편의 영역에서 비결정적인(non-critical) 신호를 전송하고 있다. 자동차에서 LIN 사양이 성공하게 된 이유를 구체적으로 밝히고, 기반 기술을 소개한다.

자료제공 | Vector Informatik

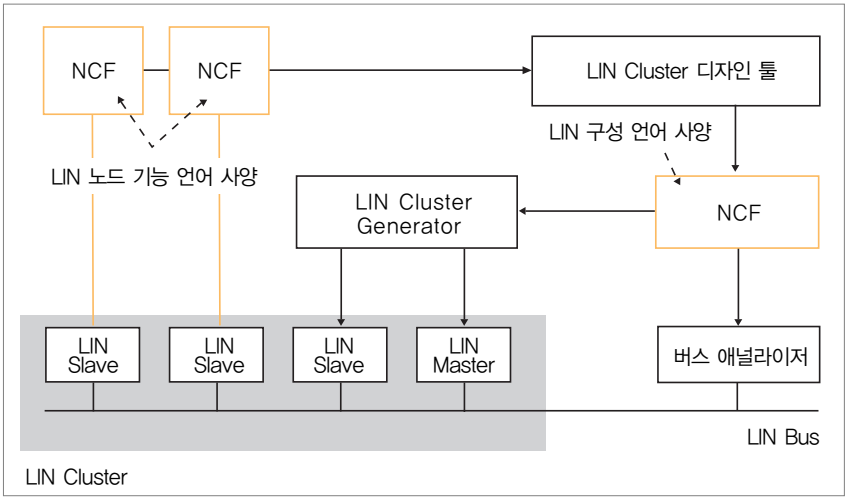
또 다른 데이터 버스가 필요한 이유

운전의 편리성에 대한 자동차 유저들의 수요가 증가하면서 이 영역의 자동차 기술도 폭넓게 전자화되고 있다. 이는 다수의 전자부품이 편의 영역으로 이동하는데 영향을 미치고 있다. 오랫동안 그 수가 계속해서 늘어나고 있는 센서, 액추에이터, 스테핑 모터 및 DC모터를 중앙 제어 모듈에 직접 연결하는 것이 일반적인 관행이었다. 이러한 흐름은 점차 비판에 직면하게 되었다. 즉, 이로 인해 배선 경비가 크게 증가하였고 공간도 더 필요해졌으며 하중도 증가하였고 고장 가능성도 크게 높아졌다. 또한 증가하는 개별화는 많은 다른 와이어 하니스와 커넥터 변종을 요구하며 이로 인해 생산, 설치, 정비 역시 더욱 어려워졌다.

개발자들은 버스 시스템을 통한 부품들의 네트워킹이 자동차 애플리케이션 영역에서는 이상적인 솔루션이라는 것을 빠르게 인식하게 되었다. 그러나 CAN 버스가 비용에 민감한 센서/액추에이터 영역에서는 활용할 수 있는 후보가 아니었기 때문에 많은 자동차 OEM들과 서플라이어들은 1990년대 중

반에 이미 자신들의 고유한 센서/액추에이터 버스 시스템을 개발하기 시작했다.

이에 따라 많은 경제적이고 간단한 고유의 센서/액추에이터 버스들이 개발되었다. 그 중 2000년에 LIN이 센서/액추에이터 영역을 위한 시리얼 버스 시스템으로서 네트워킹 시장에 소개되었다. 이 기술은 광범위하게 보급되었으며 현재 거의 모든 차량에서 공조, 시트 조정, 도어 제어 및 미러 조정과 같은 편의장치에 일반적으로 사용되고 있다.



【그림 1】 LIN Work Flow: LIN Cluster로 가는 표준화된 신속한 경로

LIN 컨소시엄의 지원

LIN이 빠르게 정착된 중요한 이유는 LIN 컨소시엄의 설립이었다. 컨소시엄에는 반도체와 툴 메이커들은 물론 유명한 자동차 OEM 및 서플라이어들이 참여했다. 컨소시엄의 목표는 센서/액추에이터 영역을 위한 OEM 간 통신 표준을 만드는 것이었다. 간단하고 간결한 통신 프로토콜은 물론 ISO 9141 표준에 기반한 간단하고 경제적인 물리계층의 정의로 LIN 컨소시엄은 성공을 위한 토대를 마련했다. 즉 간단하고 경제적인 버스 노드의 구현을 위한 무대가 마련된 것이다.

LIN 컨소시엄은 LIN 통신 자체에 초점을 두었음은 물론 개발 방법론(LIN Work Flow)까지 제공하였는데, 이것은 자동차 산업에 버스 시스템의 승인을 가속화시켰다. LIN Work Flow는 시간과 경비를 절약하면서도 LIN 네트워크(LIN Cluster)의 개발을 자동화할 수 있게 해준다. 개발 방법론의 기본은 전체 LIN Cluster와 개별 LIN 노드를 일관되게 기술하는데 사용하는 2가지 데이터 교환 포맷이다(**그림 1**).

전체 LIN Cluster를 기술하는 데에 기여하는 것은 균일한 신택스(LIN 구성 언어)와 표준화된 LDF(LIN Description File)이다. 모든 LIN Cluster의 특성들, 특히 통신 관계는 LDF에 정의되어 있다. 생성기 툴들은 LDF를 이용하여 LIN 통신에 필요한 소프트웨어 구성요소들을 생성한다. 더욱이 LDF는 분석, 측정, 테스팅 툴 및 rest-of-bus 에뮬레이터에 대해 필요한 정보를 제공한다.

마찬가지로, 개별 LIN 노드들(LIN Slaves)의 기술은 Node Capability Language와 표준화된 Node Capability Files(NCF)의 균일한 신택스에 의해 주어진다. NCF는 LIN Slave(프레임과 신호 정의, 비트 속도, 진단을 포함)의 성능 특성을 기술하며 시스템 설계의 프레임워크에서는 LDF의 자동 생성을 위한 기초를 기술한다.

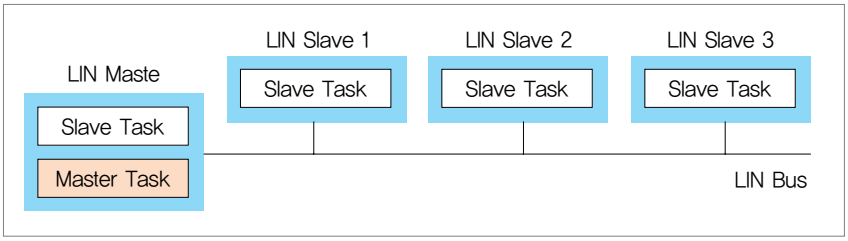
이 2가지 데이터 교환 포맷과 LIN 사양에서 정의된 환경 설정 프로세스는 사용자에게 LIN Slave 타입(예를 들면 스테핑 모터)을 LIN Cluster에 수차례 구현하게 하거나 LIN Slave를 다른 LIN Cluster들(LIN Cluster들의 재사용성)에 사용하게 해준다.

LIN의 성공에 똑같이 중요하게 기여한 것은 사양의 상세한 문서화다. 2006년 11월부터 공개된 LIN 사양 2.1은 물리층, 통신 프로토콜, LIN Work Flow, LIN API, LIN 노드들의 진단과 배치를 정의한다.

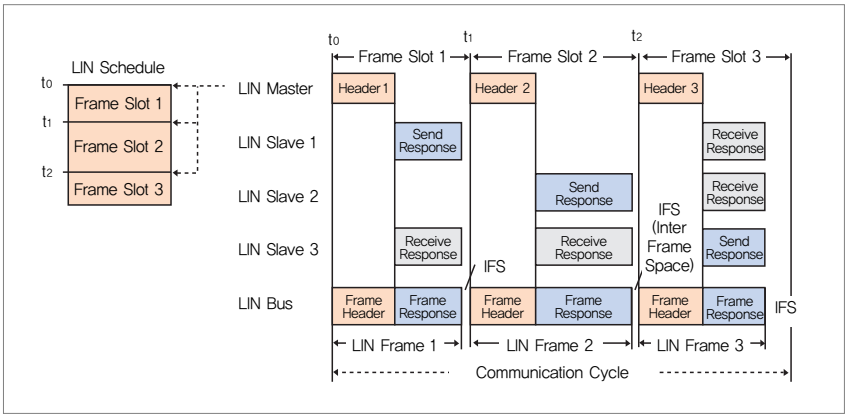
최대 20 kbps의 단선 통신

안전이 결정적이지는 않은 센서/액추에이터 영역에서 시리얼 데이터 교환을 위한 저렴한 통신 프로토콜을 만드는 목표는 주로 물리계층의 설계에 영향을 미쳤다. LIN Cluster에서 물리적인 신호 전송은 CAN에서 익숙한 차동 신호(differential signal) 전송을 포함하지 않으며 기존의 단일 선이 사용된다. 이 방법에도 불구하고 노이즈 내성을 충분히 보장하기 위해 버스 레벨을 위한 기준 전압으로 ECU의 공급전압과 그라운드가 사용된다. LIN 트랜시버는 물리적 버스 인터페이스로 사용된다. 공급전압보다 적어도 40% 낮은 레벨은 리시버에 의해 논리 "0"으로 해석된다. 리시버들은 적어도 공급전압보다 60% 높은 레벨을 논리 "1"로 해석한다.

최대 데이터 전송속도는 노이즈 방사를 한계 이내로 유지하기 위해 20 kbps로 한정된다. 전선의 길이 최대 40 m까지는 최대 추천 노드 수가 16개이다. 이것은 LIN 사양에서 규정한 LIN Cluster의 최대 허용 시상수를 포함한 노드와 라인의 커패시턴스를 고려하게 된다.



【그림 2】 Master-Slave 통신 아키텍처: 모든 LIN 노드들은 LIN Cluster에서 통신에 참여하기 위한 Slave 태스크를 갖는다. 하나의 LIN 노드는 역시 LIN 통신을 제어하기 위해 하나의 Master 태스크를 갖는다.



【그림 3】 중앙 메시지 분산 시스템: LIN Master는 Master Task와 LIN 스케줄에 의해 모든 LIN 프레임들의 송신과 수신을 제어한다. 프레임 슬롯은 연관된 LIN 프레임을 전송할 수 있을 만큼 충분히 길어야 한다. IFS의 길이는 주로 Master 태스크의 실행 사이클(시 기반, time base)에 의해 정해진다.

반도체 기술의 관점에서 LIN Cluster는 Open Collector 회로에 해당한다. pull-up 저항은 모든 LIN 노드의 Tx 트랜지스터들이 억제되는 동안 버스 레벨이 거의 공급전압 레벨(High 레벨)로 유지되는 것을 보장한다. 버스 레벨은 Tx 트랜지스터들 중 하나가 인에이블 되는 순간에 거의 그라운드 레벨(Low 레벨)까지 끌어내려진다. 따라서 Low 상태는 도미넌트(dominant) 레벨, High 상태는 리세시브(recessive) 레벨이라 한다.

Master-Slave 통신

LIN Cluster의 통신은 Master-Slave 아키텍처를 기반으로 한다. 클러스터는 Master 노드(LIN Master)와 적어도 하나의 Slave 노드(LIN Slave)로 구성된다. 경제적인 이유에서 명시적 통신 컨트롤러는 사용되지 않는다. 대신에 LIN 통신은 각각의 노드에서 소프트웨어 태스크들, 즉 Slave 태스크들로 구현된다. LIN Master는 또한 Master 태스크를 가지고 있는데, 이것은 클러스터 통신을 조정하기 위해 사용된다(**그림 2**).

조정은 프레임 슬롯에 조직되어 있는 LIN 스케줄의 주기적인 실행으로 이루어진다(**그림 3**).

각 프레임 슬롯의 초입에서 Master 태스크는 프레임 식별자(Identifier, ID)를 갖는 프레임 헤더를 전송하는데, 모든 LIN Slave들은 이것을 Slave 태스크에서 평가한다. 프레임 헤더에 바로 후속하여 LIN Slave는 ID와 연관된 프레임 리스펀스(frame response)를 전송한다. LIN 프레임은 프레임 헤더와 프레임 리스펀스로 구성되는데, 이것은 ID 기반의 메시지 어드레싱 덕분에 각각의 LIN 노드(방송)에 의해 수신될 수 있다.

LIN 프레임에 의한 데이터 전송

통신 컨트롤러의 부족으로, LIN Cluster에서 시리얼 데이터 전송은 마이크로컨트롤러의 시리얼 인터페이스(Serial Communication Interface, SCI)를 통해 처리되며 바이트 단위로 수행된다. SCI는 먼저 LSB(Least Significant Bit)를 필두로 각 바이트를 전송하며, 바이트는 스타트 비트와 스톱 비트로 구성된다(SCI 프레임). 즉 LIN 프레임은 프레임 헤더와 프레임 리스펀스 사이에 분산되어 있는 수많은 SCI 프레임의 조합으로 되어 있다(**그림 4**).

프레임 헤더를 전송함에 있어서 LIN Master는 2가지 핵심 통신 태스크를 수행한다. 즉 LIN Slave들을 동기하고 sender와 하나 이상의 리시버를 프레임 리스펀스에 할당함으로써 통신을 위임한다.

비용에 민감한 문제로 인해 LIN Slave들은 주파수 허용오차가 최대 14%에 달하는 온칩 레조네이터를 사용할 수 있다. 따라서 LIN Master는 모든 LIN Slave들이 LIN 프레임 전송이 시작되고 있음을 알 수 있게 Sync Break를 먼저 전송한다. Sync Break는 적어도 13개의 연속된 도미넌트 비트들로 구성되어 있으며, 모든 LIN Slave들로부터의 SCI 에러를 도출해낸다. 이것은 Sync Break Delimiter(적어도 하나의 리세시브 비트)에 의해 종료된다. LIN Master는 통신 클록 펄스와 함께 이어지는 Sync Byte(0x55 값을 갖는 SCI 프레임)를 전송한다.

ID는 통신을 위임하는 데에 기여한다: 즉

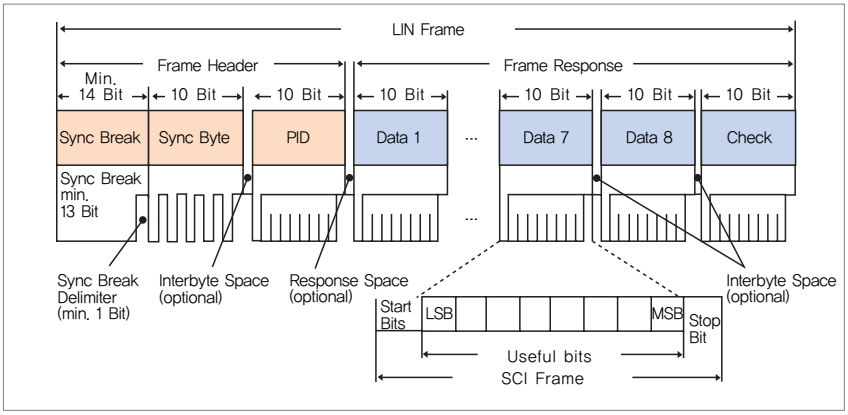
길이는 6비트이며 짝수 패리티와 홀수 패리티 등 2가지 패리티 비트로 보호된다. ID와 2개의 패리티 비트는 PID(Protected Identifier)로 불린다. 처음 60 ID는 유용한 데이터 통신에 이용된다. ID 60에서 63까지 마지막 4개의 식별자들은 진단 목적으로 사용된다.

프레임 리스펀스는 최대 8개의 데이터 바이트와 에러 검출을 위한 체크섬(checksum)으로 구성된다. 구분은 클래식과 확장된 체크섬으로 가능하다. 클래식 체크섬은 모든 데이터 바이트들의 인버트된 모듈로 256 섬이다. 모듈로 256 섬을 도착하는 데이터 바이트들에 추가한 결과가 "0xFF"와 일치하지 않으면 전송 에러가 있게 된다. 연장된 체크섬은 인버트된 모듈로 256 섬을 형성함에 있어서 역시 PID를 고려한다.

LIN Slave들이 통상 매우 간단하고 저렴한 마이크로컨트롤러로 구성되기 시작한 이래, 프레임 리스펀스의 전송을 지연하는 것(Response Space)이 허용됨은 물론, SCI 프레임의 전송 사이에서 센딩 펄스(Interbyte Spaces)를 삽입할 수도 있다. 전체적으로 이것은 프레임 리스펀스를 40%까지 늘릴 수 있다. 이것은 프레임 헤더에도 똑같이 적용되는데, 이유는 Sync Break를 생성하는 방법에는 다른 방법들이 있기 때문이다. 이것은 LIN 스케줄을 설계함에 있어서 40%의 시간 여유를 결정하는 데 필수적이다.

산발적, 이벤트 트리거형, 진단 프레임들

LIN 사양은 LIN 스케줄에 의해 엄격하게 규정된 통신 사이클에 유연성과 경제성을 제공하는 규정들을 가지고 있다. 이 규정들은 통신 사이클들을 더욱 유연하고 경제적이게 해 준다. Sporadic(산발적) 프레임과 Event Triggered(이벤트 트리거형) 프레임



【그림 4】 각각의 LIN 프레임은 프레임 헤더와 프레임 리스펀스로 이루어진다. 프레임 헤더는 항상 LIN Master에 의해 보내진다. 프레임 리스펀스는 하나의 LIN Slave나 LIN Master에 의해 보내질 수 있다.

이라는 2가지 프레임 타입은 이런 목적으로 제공된다. 이들 부수적인 프레임 타입들을 도입함에 있어서 종래의 LIN 프레임을 Unconditional(무조건적) 프레임이라고 명명하는 것이 상식이 되었다.

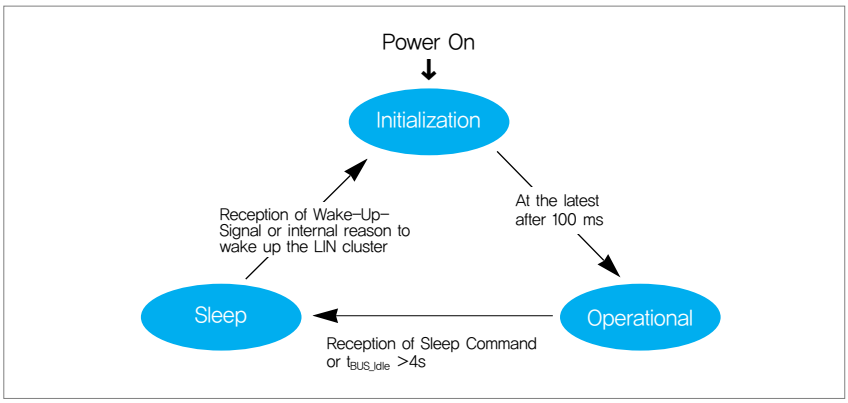
Sporadic 프레임은 다른 Unconditional 프레임들과 프레임 슬롯을 공유하는 Unconditional 프레임으로 이해된다. Sporadic 프레임들은 필요에 따라 전적으로 LIN Master에 의해 전송되므로 충돌이 있을 수 없다. LIN Master가 어떤 프레임도 필요하지 않으면 연관된 프레임 슬롯은 비어 있게 된다.

Event Triggered Frame은 LIN Slave 쪽의 산발적인 변화나 이벤트들을 통신하기 위해 도입되었다. 이것은 필연적으로 Unconditional Frame에 상당하지만 다른 LIN Slave들로부터의 수많은 프레임 응답이 프레임 헤더에 할당되는 것과는 다르다. Event Triggered 프레임 헤더를 완성하는 데에 사용되는 프레임 응답은 관련된 LIN Slave들의 필요에 따른다. 이 필요는 만일 전송되어야 할 새로운 데이터가 있을 때에 있게 된다. Event Triggered Frame은 첫 번째 바이트에 있는 연관된 Unconditional Frame의 PID에 의해 인식된다.

Sporadic Frame에 대조적으로 충돌은 Event Triggered Frame에서는 배제될 수 없다. 충돌의 경우에 LIN Master는 모든 Event Triggered Frame에 할당된 Unconditional Frame들의 전송에 책임이 있다. 이것은 "Collision Resolving Schedule"을 활성화하고 처리함으로써 행해진다.

종래의 Unconditional Frames와 특별한 Diagnostic Frames는 LIN Slave들을 진단하는 데에 적합하다. Unconditional Frames는 간단한 신호 기반의 진단에 사용되는데 반해 Diagnostic Frames는 유저가 규정한 진단이나 표준화된 전송 프로토콜과 균일한 진단 서비스들을 위해 사용된다.

LIN 사양은 "Master Request Frame"과 "Slave Response Frame"이라는 2가지 Diagnostic Frames를 규정한다. Master Request Frame (ID=0x60)은 Diagnostic Request를 나타낸다. 이 경우에 LIN Master는 프레임 헤더와 프레임 리스펀스를 모두 전송한다. 예를 들어 CAN을 경유하여 Diagnostic Request가 있으면 Master Request Frame이 전송된다. 이 경우에 LIN Master는 헤더를 전송하며, 특정 LIN Slave는 리스펀스를 전송한다.



【그림 5】 LIN Slave의 통신 상태 모델

Management 기능

LIN 사양은 Status Management와 Network Management를 정의한다. Status Management는 LIN Slaves가 LIN Master에게 패리티나 체크섬 에러들 같은 검출된 송신에러들을 알린다. 이것은 Unconditional Frame에 있는 "Response Error Signal"에 의해 행해진다("Status Frame"). 그러나 이 프레임은 에러의 유형에 대한 더 이상의 정보를 갖지는 않는다. LIN 사양은 에러 처리를 규정하지 않으며, 오히려 이 태스크를 유저에게 남긴다.

LIN Network Management의 첫 번째 태스크는 LIN Cluster에 있는 모든 Slave들을 정상 통신 상태(Operational)에서 Sleep 상태 및 다른 방향으로의 변환을 조정하는 것이다(그림 5). LIN Slaves가 4초간 어떠한 버스 활성도 감지하지 못하면 이들은 Operational 상태에서 Sleep 상태로 전환된다. 같은 조건이 LIN Master로부터 Sleep 명령을 끌어내는데, 이것은 확실히 특수한 Master Request Frame이다.

바꿔 말하면 LIN Slave가 유효(valid) 헤더가 따르는 "Wake Up Signal"을 감지하면, LIN Slave는 Sleep 상태에서 초기화(Initialization) 상태를 거쳐 Operational 상태로 전환된다. Wake Up 신호는 길이가 최소 250 μs와 최대 5 ms의 dominant 펄스로 구성되며, 어떤 LIN 노드에 의해서든 보내질 수 있다. LIN 사양은 100 ms의 최대 초기화 단계를 규정한다. 즉 LIN Master는 이 시간 경과 후 LIN Schedule의 실행을 개시해야 한다. 만일 패시브로 유지하면 상당하는 LIN Slave는 또 다른 Wake Up 신호를 보낸다. 반복 횟수와 반복시간 및 시간 종료 간격은 규격에 정의되어 있다.

참고문헌

[1] www.lin-subbus.org
[2] LIN Specification Package Revision 2.1
[3] Road vehicles-Diagnostics on Controller Area Network (CAN)-Part 2: Network layer services, International Standard ISO 15765-2.4, Issue 4, 2002-06-21
[4] Road vehicles-Diagnostics on controller area network (CAN)-Part 3: Implementation of diagnostic services, International Standard ISO 15765-3.5, Issue 5, 2002-12-12
[5] Road vehicles-Diagnostic systems-Part1:Diagnostic services, International Standard ISO 14229-1.6, Issue 6, 2001-02-22
[6] www.vector-informatik.com
[7] www.vector-academy.com

www.AUTOELECTRONICS.co.kr

회원 가입하고
매거진 받고!!



회원으로 가입하신 분에게는
잡지 증정본을 보내드립니다.
지금 바로 신청하세요.

회원 혜택

1. www.AUTOELECTRONICS.co.kr의 회원이 되시면, 게시된 기사를 보실 수 있습니다.
2. www.AUTOELECTRONICS.co.kr의 회원이 되시면, aeNews를 이메일을 통해 받아보실 수 있습니다.
3. www.AUTOELECTRONICS.co.kr의 회원이 되시면, 자동차 전기전자 행사를 가장 먼저 확인하실 수 있습니다.
4. www.AUTOELECTRONICS.co.kr의 회원이 되시면, 가입하신 달의 잡지를 무료로 받아보실 수 있습니다.(1회)

※ aeNews는 국내외 자동차 전기전자 기술 및 산업계 소식을 정리한 보고서로, 2주에 한 번씩 월요일에 발행됩니다.