



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0007118
(43) 공개일자 2012년01월20일

(51) Int. Cl.

H04L 12/28 (2006.01) H04L 12/56 (2006.01)

H04L 29/06 (2006.01)

(21) 출원번호 10-2010-0067690

(22) 출원일자 2010년07월14일

심사청구일자 2010년07월14일

(71) 출원인

재단법인대구경북과학기술원

대구광역시 달성군 현풍면 상리 50-1

(72) 발명자

조희섭

대구광역시 달서구 성서공단로11길 62, 대구TP벤처 1공장 301호 (호산동)

김정면

대구광역시 달서구 성서공단로11길 62, 대구TP벤처 1공장 301호 (호산동)

(74) 대리인

송해모, 김은구

전체 청구항 수 : 총 12 항

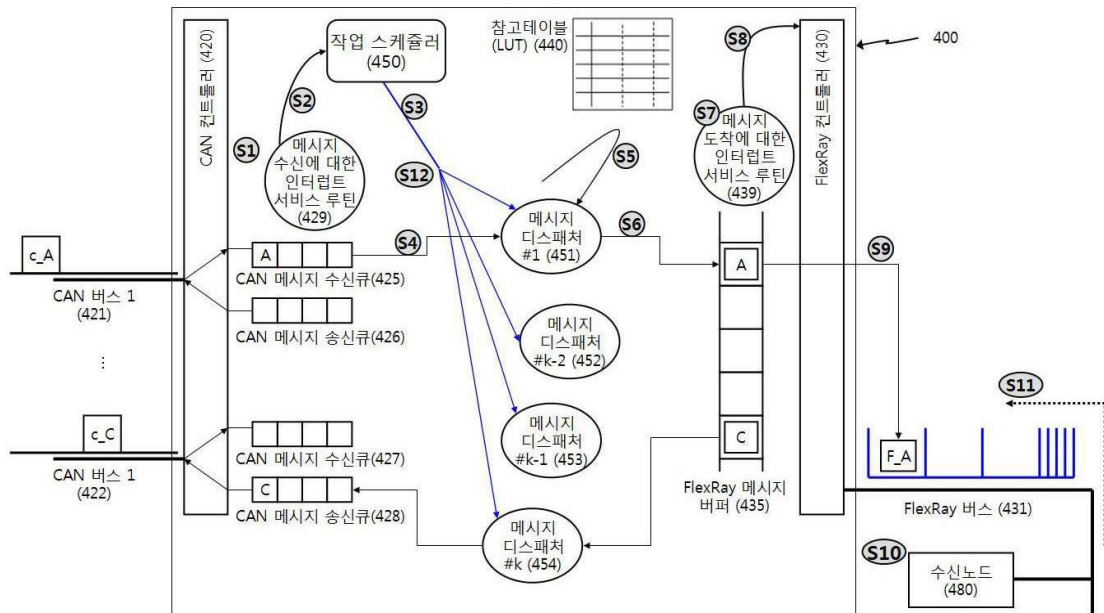
(54) 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 방법 및 장치

(57) 요약

본 발명은 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 방법 및 장치를 개시하고 있다.

본 발명의 일 실시 예에 따른 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 방법은 이종의 프로토콜을 사용하는 노드들 간에 메시지를 교환함에 있어서, 제 1 프로토콜을 사용하는 제 1 노드로부터 제 1 메시지를 수신하여 상기 메시지 수신 시간을 저장하는 단계, 제 2 프로토콜의 제 2 노드로 메시지 송신을 담당하는 메시지 디스패처에게 상기 제 1 메시지 및 상기 메시지 수신 시간을 제공하는 단계, 및 상기 메시지 디스패처는 우선 순위에 따라 상기 제 1 메시지를 송신하는 단계를 포함하며, 상기 우선 순위는 상기 메시지 디스패처가 처리하게 되는 노드의 메시지 전송 지연 발생에 따라 산출되는 것을 특징으로 한다.

대표도



특허청구의 범위

청구항 1

이종의 프로토콜을 사용하는 노드들 간에 메시지를 교환함에 있어서,

제 1 프로토콜을 사용하는 제 1 노드로부터 제 1 메시지를 수신하여 상기 메시지 수신 시간을 저장하는 단계;

제 2 프로토콜의 제 2 노드로 메시지 송신을 담당하는 메시지 디스패처에게 상기 제 1 메시지 및 상기 메시지 수신 시간을 제공하는 단계; 및

상기 메시지 디스패처는 우선 순위에 따라 상기 제 1 메시지를 송신하는 단계를 포함하며,

상기 우선 순위는 상기 메시지 디스패처가 처리하게 되는 노드의 메시지 전송 지연 발생에 따라 산출되는 것을 특징으로 하는, 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 방법.

청구항 2

제 1항에 있어서,

상기 제 1 메시지를 송신하는 단계 이후에,

상기 제 2 노드로부터 메시지 전송 지연을 알리는 항의 메시지를 수신하는 단계; 및

상기 수신한 항의 메시지를 이용하여 상기 메시지 디스패처의 우선 순위 조절 여부를 판단하는 단계를 더 포함하는, 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 방법.

청구항 3

제 1항에 있어서,

상기 제 1 프로토콜을 사용하는 제 1 노드로부터 제 1 메시지를 수신한 후, 상기 제 1 메시지의 수신을 인터럽트를 통하여 작업 스케줄러에게 알리는 단계; 및

상기 작업 스케줄러가 상기 메시지 디스패처에게 상기 제 1 메시지의 송신 작업을 할당하는 단계를 더 포함하는, 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 방법.

청구항 4

제 1항에 있어서,

상기 제 1 프로토콜을 사용하는 제 1 노드로부터 제 1 메시지를 수신한 후, 메시지 수신을 정기적으로 모니터링한 프로세스가 상기 제 1 메시지의 수신을 상기 작업 스케줄러에게 알리는 단계; 및

상기 작업 스케줄러가 상기 메시지 디스패처에게 상기 제 1 메시지의 송신 작업을 할당하는 단계를 더 포함하는, 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 방법.

청구항 5

제 1항에 있어서,

상기 제 1 프로토콜과 상기 제 2 프로토콜은 CAN 프로토콜 또는 FlexRay 프로토콜 중 하나인 것을 특징으로 하는, 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 방법.

청구항 6

제 1항에 있어서,

상기 메시지 디스패처가 우선 순위에 따라 상기 제 1 메시지를 송신하는 단계는

상기 제 1 메시지를 상기 제 2 프로토콜의 제 2 노드가 사용하는 메모리 버퍼에 상기 제 1 메시지를 저장하는 단계; 및

상기 제 1 메시지의 저장을 상기 제 2 프로토콜을 제어하는 컨트롤러에게 알리는 단계를 더 포함하는, 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 방법.

청구항 7

이종의 프로토콜을 사용하는 노드들 간에 메시지를 교환함에 있어서,

제 1 프로토콜을 사용하는 노드의 메시지 송수신을 제어하는 제 1 컨트롤러;

제 2 프로토콜을 사용하는 노드의 메시지 송수신을 제어하는 제 2 컨트롤러;

상기 제 1 프로토콜을 사용하는 제 1 노드로부터 제 1 메시지를 수신하여 상기 메시지 수신 시간을 저장하며, 메시지 디스패처를 선택하는 작업 스케줄러; 및

상기 제 1 메시지를 상기 제 2 컨트롤러가 제어하는 메모리 버퍼에 저장하는 메시지 디스패처를 포함하며,

상기 메시지 디스패처는 우선 순위에 따라 상기 제 1 메시지를 송신하고, 상기 우선 순위는 상기 메시지 디스패처가 처리하게 되는 노드의 메시지 전송 지연 발생에 따라 산출되는 것을 특징으로 하는, 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 장치.

청구항 8

제 7항에 있어서,

상기 작업 스케줄러는 상기 제 2 노드로부터 메시지 전송 지연을 알리는 항의 메시지를 수신한 후, 상기 수신한 항의 메시지를 이용하여 상기 메시지 디스패처의 우선 순위 조절 여부를 판단하는, 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 장치.

청구항 9

제 7항에 있어서,

상기 제 1 프로토콜을 사용하는 제 1 노드로부터 제 1 메시지를 수신한 후, 상기 제 1 메시지의 수신을 작업 스케줄러에게 알리는 인터럽트를 처리하는 인터럽트 서비스 루틴을 더 포함하며,

상기 작업 스케줄러는 상기 인터럽트에 의하여 상기 메시지 디스패처에게 상기 제 1 메시지의 송신 작업을 할당하는, 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 장치.

청구항 10

제 7항에 있어서,

메시지 수신을 정기적으로 모니터링 하는 프로세스를 더 포함하며,

상기 프로세스는 상기 제 1 프로토콜을 사용하는 제 1 노드로부터 상기 제 1 메시지의 수신을 작업 스케줄러에게 알리며,

상기 작업 스케줄러는 상기 메시지 디스패처에게 상기 제 1 메시지의 송신 작업을 할당하는, 이종 프로토콜 통

신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 장치.

청구항 11

제 7항에 있어서,

상기 제 1 프로토콜과 상기 제 2 프로토콜은 CAN 프로토콜 또는 FlexRay 프로토콜 중 하나인 것을 특징으로 하는, 이중 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 장치.

청구항 12

제 7항에 있어서,

상기 메시지 디스패처는 상기 제 1 메시지를 상기 제 2 프로토콜의 제 2 노드가 사용하는 메모리 버퍼에 상기 제 1 메시지를 저장하고, 상기 제 1 메시지의 저장을 상기 제 2 프로토콜을 제어하는 컨트롤러에게 알리는, 이중 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 장치.

명세서

기술분야

[0001] 본 발명은 이중 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 방법 및 장치를 개시하고 있다.

배경기술

[0002] 현대의 차량 내에는 마이크로프로세서, 센서, 그리고 액추에이터 등의 다양한 전자기기들이 사용되고 있으며 이들은 차량의 안전한 주행과 운전자의 편의를 위해 상호 간의 신뢰성 있는 정보 교환을 필요로 한다.

[0003] 차량 내부의 통신 방식으로 가장 널리 사용되고 있는 것은 CAN (Control Area Network)으로서 최고 1 Mbps의 전송률로 데이터를 전송할 수 있다. 그러나 이벤트 기반 (Event-triggered) 통신 방식의 한계와 비교적 낮은 전송률로 인해 x-by-wire 시스템과 같이 고도의 신뢰성과 안전성이 요구되는 시스템에 적용하기에는 적합하지 않은 방식이다. 이의 대안으로 확정적 통신 (deterministic communication)의 속성을 지니며 적은 지터(jitter)로 주기적인 데이터 교환이 가능한 FlexRay가 개발되었다. FlexRay는 시간 동기 (Time-triggered)에 의해 확정성을 제공하는 TDMA 방식과 이벤트 기반 통신 방식을 혼용하여 적용할 수 있는 유연성을 제공하도록 설계되었다. 그러나, 시간 동기에 의해 운용되는 FlexRay는 데이터를 전송하는 과정에서 시간상의 지연이 발생할 경우, 수신측이 지연된 데이터를 수신하게 되며, 이러한 지연이 다른 데이터를 전송하는 데 있어 영향을 미칠 수 있다는 문제가 있다.

[0004] '차량 내 네트워크'는 증가하는 네트워크 도메인의 수와 여기에서 기인하게 되는 도메인 간 통신에 있어서의 복잡도의 증가, 그리고 전송 지연의 문제점에 직면하게 되었다. 특히, 이중 도메인 간 (CAN ? LIN, CAN ? FlexRay 등)의 통신을 위해서는 '게이트웨이 (Gateway) 기능'이 필요한데, 이 게이트웨이 기능을 제공함에 있어서는 해당 데이터가 적절한 시점에 전송이 완료될 수 있는 신뢰성 있는 통신을 보장하는 것이 필요하다.

발명의 내용

해결하려는 과제

[0005] 본 발명은 FlexRay 통신에서 메시지를 송수신하는 과정에서 발생하는 지연을 해결하기 위해 메시지 전송을 제어하는 프로세스의 우선 순위를 조절하는 기술을 제시하고자 한다.

[0006] 본 발명은 게이트웨이 내에서의 전송 지연을 정량적으로 측정하여, 전송 지연 시간이 시스템의 허용 한계를 넘어서는 경우에 이를 극복할 수 있는 방법을 제시하고자 한다.

[0007] 특히, 본 발명은 CAN - FlexRay 이중의 메시지 전송에서 발생하는 지연을 메시지 전송을 제어하는 메시지 디스

패치의 처리 우선 순위를 제어하여, 메시지 자체의 우선 순위를 변경시키지 않도록 하여, 복합적인 전송 지연을 정량화하고 이를 보상할 수 있으며, 차량 내 게이트웨이에 적용될 경우 차량 내 네트워크의 신뢰성을 극대화 시킬 수 있다.

과제의 해결 수단

[0008] 전술한 과제를 달성하기 위해, 본 발명의 일 실시 예에 따른 이중 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 방법은 이중의 프로토콜을 사용하는 노드들 간에 메시지를 교환함에 있어서, 제 1 프로토콜을 사용하는 제 1 노드로부터 제 1 메시지를 수신하여 상기 메시지 수신 시간을 저장하는 단계, 제 2 프로토콜의 제 2 노드로 메시지 송신을 담당하는 메시지 디스패처에게 상기 제 1 메시지 및 상기 메시지 수신 시간을 제공하는 단계, 및 상기 메시지 디스패처는 우선 순위에 따라 상기 제 1 메시지를 송신하는 단계를 포함하며, 상기 우선 순위는 상기 메시지 디스패처가 처리하게 되는 노드의 메시지 전송 지연 발생에 따라 산출되는 것을 특징으로 한다.

[0009] 본 발명의 일 실시 예에 따른 이중 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 장치는 이중의 프로토콜을 사용하는 노드들 간에 메시지를 교환함에 있어서, 제 1 프로토콜을 사용하는 노드의 메시지 송수신을 제어하는 제 1 컨트롤러, 제 2 프로토콜을 사용하는 노드의 메시지 송수신을 제어하는 제 2 컨트롤러, 상기 제 1 프로토콜을 사용하는 제 1 노드로부터 제 1 메시지를 수신하여 상기 메시지 수신 시간을 저장하며, 메시지 디스패처를 선택하는 작업 스케줄러, 및 상기 제 1 메시지를 상기 제 2 컨트롤러가 제어하는 메모리 버퍼에 저장하는 메시지 디스패처를 포함하며, 상기 메시지 디스패처는 우선 순위에 따라 상기 제 1 메시지를 송신하고, 상기 우선 순위는 상기 메시지 디스패처가 처리하게 되는 노드의 메시지 전송 지연 발생에 따라 산출되는 것을 특징으로 한다.

발명의 효과

[0010] 본 발명은 게이트웨이 내에서의 전송 지연을 정량적으로 측정하여, 메시지 전송의 지연을 막고, 전체 네트워크의 메시지 흐름의 변화에 탄력적으로 적응할 수 있다. 특히, 본 발명은 메시지 전송을 담당하는 프로세스의 우선 순위를 조절하여, 메시지 지연에 효과적으로 대응할 수 있도록 한다.

도면의 간단한 설명

[0011] 도 1은 본 명세서의 일 실시예가 적용되는 CAN-FlexRay 게이트웨이의 구조를 보여주는 도면이다.
 도 2는 본 명세서에서 언급되는 게이트웨이의 일반적인 구조를 보여주는 도면이다.
 도 3은 본 명세서의 일 실시예에 의한 게이트웨이의 소프트웨어적, 또는 기능적 구조를 보여주는 도면이다.
 도 4는 본 명세서의 일 실시예에 의한 게이트웨이의 구성 및 각각의 구성에서 이루어지는 메시지 전달 과정을 보여주는 도면이다.
 도 5는 본 명세서의 일 실시예에 의한 CAN 메시지가 FlexRay 노드에 전송되는 과정에서 발생하는 메시지 지연을 처리하는 과정을 보여주는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0012] 이하, 본 발명의 일부 실시 예들을 예시적인 도면을 통해 상세하게 설명한다. 각 도면의 구성요소들에 참조부호를 부가함에 있어서, 동일한 구성요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의해야 한다. 또한, 본 발명을 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 발명의 요지를 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명은 생략한다.

[0013] 또한, 본 발명의 구성 요소를 설명하는 데 있어서, 제 1, 제 2, A, B, (a), (b) 등의 용어를 사용할 수 있다. 이러한 용어는 그 구성 요소를 다른 구성 요소와 구별하기 위한 것일 뿐, 그 용어에 의해 해당 구성 요소의 본질이나 차례 또는 순서 등이 한정되지 않는다. 어떤 구성 요소가 다른 구성요소에 "연결", "결합" 또는 "접속"

된다고 기재된 경우, 그 구성 요소는 그 다른 구성요소에 직접적으로 연결되거나 접속될 수 있지만, 각 구성 요소 사이에 또 다른 구성 요소가 "연결", "결합" 또는 "접속"될 수도 있다고 이해되어야 할 것이다.

[0014] CAN과 FlexRay 버스 간에 메시지를 교환할 수 있는 방법으로 제안된 내용은 CAN 메시지의 ID와 FlexRay 슬롯을 매칭하는 방법 또는 메시지를 변환 방법이 있다. 한편, CAN과 FlexRay 버스 간에 메시지를 교환하기 위한 매핑 테이블을 제안하고 이에 필요한 필터링 기법과 메시지를 저장하기 위한 메시지 큐를 적용할 수 있다. 그러나 이러한 방법을 적용할 경우, CAN과 FlexRay와 같이 서로 다른(이종의) 버스 사이에 위치한 게이트웨이 내의 메시지 전달 과정에서 전송 지연이 발생할 수 있다. 특히, 한쪽 버스의 노드에서 전송한 메시지가 상대 측 버스에 접속된 노드에 도달할 때까지의 시간에 더해진 전송 지연을 정량적으로 측정하지 않아, 이러한 전송 지연이 전체 네트워크 시스템의 허용 한계를 넘어설 경우, 네트워크의 효율이 급격히 저하될 수 있다. 이하, 게이트웨이 내에서의 전송 지연을 측정하여 전송 지연에 의한 네트워크 효율의 저하를 방지하는 방법 및 장치에 대해 살펴보고자 한다.

[0015] 또한, 본 명세서에서는 게이트웨이 내에서의 전송 지연을 정량적으로 측정하고, 전송 지연 시간이 시스템의 허용 한계를 넘어서는 경우에 이를 극복할 수 있는 과정 및 장치를 살펴보고자 한다.

[0016] 도 1은 본 명세서의 일 실시예가 적용되는 CAN-FlexRay 게이트웨이의 구조를 보여주는 도면이다. 도 1의 CAN-FlexRay 게이트웨이(150)는 다수의 CAN 버스(1, ..., N)와 한 개 이상의 FlexRay 버스가 접속된다. 다수의 CAN 버스(1, ..., N)는 다수의 CAN 노드들(111, 112, 113, 114)로부터 수신한 다수의 메시지를 전달하며, 동시에 전달되는 수많은 CAN 메시지들이 전송 지연되며 FlexRay 버스로 전달될 수도 있다. 즉, CAN 버스는 다수이므로 동시에 메시지를 송신할 수 있으나 게이트웨이의 기능은 이들을 동시에 처리할 수 없기 때문에 CAN 노드들(111, 112, 113, 114)이 모두 메시지들을 생성하여 송신할 경우, FlexRay 노드들(121, 122, 123, 124)에게 지연되어 전송될 수 있다.

[0017] 도 2는 본 명세서에서 언급되는 게이트웨이의 일반적인 구조를 보여주는 도면이다.

[0018] CAN-FlexRay 게이트웨이(200)는 다수의 CAN 버스(221, 229)와 한 개 이상의 FlexRay 버스(231)가 접속되는 경우가 일반적이다. 이러한 구성 하에서는 다수의 CAN 메시지가 복수의 CAN 버스를 통해 게이트웨이에 전달되는 상황이 발생할 수 있으며, 극단적인 경우에는 거의 동시에 전달된 수많은 CAN 메시지들을 최소한의 전송지연을 부가하여 FlexRay 버스로 전달하는 것이 요구될 수 있다.

[0019] 게이트웨이에 접속된 각각의 CAN 버스마다 이를 통해 수신된 CAN 메시지를 임시로 저장할 수 있는 'CAN 메시지 수신 큐 (Queue)'가 할당되어 있으며 또한, CAN 버스를 통해 전송될 메시지를 임시로 저장할 수 있는 'CAN 메시지 송신 큐'가 할당된다. 두 가지 큐를 'CAN 메시지 큐'로 통칭하기로 한다. FlexRay 메시지 버퍼(235)는 FlexRay 데이터를 송, 수신 할 때 임시로 데이터를 저장하는 데에 사용되며 다수의 메시지 버퍼들은 전송용과 수신용으로 미리 구분되어 사용된다. 도 2에서 CAN 버스 1의 메시지 'c_A'가 CAN 메시지 수신큐에 데이터인 A만 추출되어 수신된다. 이는 FlexRay 버퍼(235)에 A로 저장된 후, FlexRay 버스(231)에 'F_A'로 변환되어 프레임에 포함되어 전송되는 것을 알 수 있다. 이러한 이종의 버스들 간의 메시지 교환은 메인 CPU 코어(210)에서 각각의 CAN 컨트롤러(220)와 FlexRay 컨트롤러(230)를 제어하는 것을 통해 이루어진다.

[0020] 본 명세서에서 제안한 방법은 다양한 버스 접속 상황에 적용 가능하지만 도 1에서와 같은 일반적인 상황을 상정하여 기술하도록 한다.

[0021] 도 3은 본 명세서의 일 실시예에 의한 게이트웨이의 소프트웨어적, 또는 기능적 구조를 보여주는 도면이다.

[0022] CAN 컨트롤러(320)는 CAN 버스(321)를 통해 메시지(A)를 수신하게 되면 인터럽트를 발생시키고 이에 대한 인터럽트 서비스 루틴(Interrupt Service Routine, ISR)을 통해 CAN 메시지에서 데이터(Payload를 지칭)를 추출하여 CAN 메시지 수신 큐(325)에 저장하게 된다. CAN 메시지 수신 큐(325)에 새로운 데이터가 저장되면 인터럽트가 발생되며 이의 인터럽트 서비스 루틴에서 작업스케줄러(Job Scheduler, 작업 관리자, 350)를 호출하며 제어권을 넘겨주게 된다. 또는 도 2에서 살펴보았던 메인 CPU 코어(210)에 의해 동작하는 주기적 태스크가 CAN 컨트롤러(320)를 통해 CAN 버스(321)를 주기적으로 체크 (Polling)하며 메시지의 수신을 확인 할 수도 있으며 이때에는 수신을 확인한 태스크가 CAN 메시지에서 데이터를 추출하여 CAN 메시지 수신 큐에 저장하게 되며 이후에 작업 스케줄러(350)를 호출하여 제어권을 넘겨주게 된다.

- [0023] 이와는 반대로, FlexRay 측에서 전달된 데이터가 CAN 메시지 송신 큐(328)에 저장되면 인터럽트가 발생하게 되고 이에 대한 인터럽트 서비스 루틴에서는 CAN 컨트롤러(320)로 하여금 큐(328)에 저장된 메시지를 CAN 버스(321)를 통해 전송하도록 요청하게 된다. 또는 도 2에서 살펴보았던 메인 CPU 코어(210)에 의해 동작하는 주기적 태스크가 CAN 메시지 송신 큐(328)를 주기적으로 체크하며 메시지의 도착을 확인할 수도 있는데, 이 경우, 주기적 태스크가 CAN 컨트롤러(320)에게 큐에 저장된 메시지를 CAN 버스를 통해 전송하도록 요청하게 된다.
- [0024] 본 명세서의 일 실시예에서 제공하는 게이트웨이의 기능 중에서 이종 버스 간에 메시지 전달을 담당하는 것은 메시지 디스패처(Message Dispatcher)(351, 355)로서 CAN 메시지 큐와 FlexRay 메시지 버퍼 간에 데이터를 전송하는 역할을 담당하는 태스크이다. 메시지 디스패처는 CAN 메시지의 ID, CAN 메시지 큐의 주소, FlexRay 메시지 버퍼의 주소, FlexRay 슬롯의 번호 간의 관계가 정의된 참고 테이블(Look-up Table, LUT)(340)을 참고하여 데이터의 송, 수신 방향과 목적지를 정확하게 파악할 수 있게 된다. 복수의 메시지가 게이트웨이 내에서 다양한 버스 쪽으로 전달되므로 메시지 디스패처(351, 355)는 다수 개가 구동될 수 있다. 이들 태스크들은 작업 스케줄러(Job Scheduler, 350)에 의해 우선순위를 부여 받고 실행 순서가 결정된다. 작업 스케줄러(350)는 메시지 디스패처(351, 355) 외에도 인터럽트 서비스 루틴이나 주기적 태스크의 우선순위를 조절할 수 있는 자격을 가지고 있다.
- [0025] 게이트웨이의 메시지 디스패처(351, 355)를 통해 CAN 버스 측의 데이터가 FlexRay 메시지 버퍼(335)에 저장되면 인터럽트가 발생되며, 발생한 인터럽트에 대한 서비스 루틴(ISR)에서는 FlexRay 컨트롤러(330)로 하여금 메시지 버퍼(335)에 저장된 데이터를 FlexRay 버스(331)를 통해 전송하도록 요청하게 된다. 또한 도 2에서 살펴보았던 메인 CPU 코어(210)에 의해 동작하는 주기적 태스크가 FlexRay 컨트롤러(330)를 통해 FlexRay 메시지 버퍼(335)를 주기적으로 체크하며 메시지의 도착을 확인할 수도 있으며 이때에는 도착을 확인한 태스크가 FlexRay 컨트롤러(330)에게 메시지 버퍼에 저장된 메시지를 FlexRay 버스(331)를 통해 전송하도록 요청하게 된다.
- [0026] 역으로, FlexRay 컨트롤러(330)는 FlexRay 버스(331)를 통해 메시지를 수신하게 되면 인터럽트를 발생시키고 발생한 인터럽트를 처리하는 ISR을 통해 FlexRay 프레임(F_C)에서 데이터(C)를 추출하여 FlexRay 메시지 버퍼에 저장하게 된다. 메시지 버퍼에 데이터가 저장되면 인터럽트가 발생하게 되고 이의 인터럽트 서비스 루틴에서는 작업 스케줄러를 호출하며 제어권을 넘겨주게 된다. 또는 메인 CPU 코어에 의해 동작하는 주기적 태스크가 FlexRay 컨트롤러를 통해 FlexRay 버스를 주기적으로 체크하며 메시지의 수신을 확인할 수도 있으며 이때에는 수신을 확인한 태스크가 FlexRay 프레임에서 데이터를 추출하여 FlexRay 메시지 버퍼(335)에 저장하게 되며 이후에 작업 스케줄러(350)를 호출하여 제어권을 넘겨주게 된다.
- [0027] 도 4는 본 명세서의 일 실시예에 의한 게이트웨이의 구성 및 각각의 구성에서 이루어지는 메시지 전달 과정을 보여주는 도면이다.
- [0028] 전체 구성은 앞서 도 2, 3에서 살펴본 구조에, 작업 스케줄러(작업 관리자, 450)이 추가되며, 이들 작업 스케줄러(450)가 메시지 디스패처(451, 452, 453, 454)를 이용하여 CAN 컨트롤러와 FlexRay 컨트롤러 사이에 메시지를 교환하는 작업을 수행한다. 또한 작업 스케줄러(450)는 메시지가 지연되는 경우, 메시지 디스패처(451, 452, 453, 454)의 태스크 우선 순위를 조절하여 메시지 지연을 보상한다. 보다 상세히 살펴보면 다음과 같다. 설명의 편의를 위하여 메시지 및 작업의 흐름을 (S1, S2, ..., S12)로 표시하며 각각의 단계에 대해 S1, S2와 같이 설명하고자 하며, 이들 흐름 및 순서는 발명을 적용함에 있어 일부 그 우선 순위가 바뀔 수 있으며, 새로운 과정이 추가되거나, 또는 일부 과정이 생략하여 진행될 수도 있다. 작업 스케줄러는 작업 관리 기능을 포함할 수 있다.
- [0029] CAN 버스(421)를 통해 수신되는 메시지(c_A)에 대해 CAN 컨트롤러(420)가 인터럽트를 발생시킨다(S1). 또는 인터럽트 기능 대신에 도면에 미도시 되었으나, 특정 태스크(수신 처리 태스크)에 의한 주기적인 폴링을 수행하여 CAN 메시지가 수신되었는지 확인할 수 있다. 해당 인터럽트 서비스 루틴(429)이나 수신 처리 태스크에서 알리는 CAN 메시지의 도착시간을 별도로 저장한 후에 수신된 메시지에서 데이터(A)를 추출해 CAN 메시지 수신큐(425)에 CAN 노드의 식별자 정보(ID)와 함께 저장한 후 작업 스케줄러에게 제어권을 넘긴다(S2). 이후, 작업 스케줄러는 작업 관리 기능을 이용하여 도착시간과 제어권을 넘길 메시지 디스패처를 선택한다(S3). 즉, 작업 스케줄러(450)는 해당 데이터의 전송을 위해 메시지 디스패처 #1(451)에게 작업을 할당하고 우선 순위를 부여하고, 메시지의 도착시간을 해당 메시지 디스패처 #1(451)에게 전달한다. 이를 수신한 메시지 디스패처 #1(451)는 CAN 메시지 수신큐(425)의 메시지 ID를 추출하여(S4), 추출한 메시지 ID를 참고 테이블(LUT, 440)와 대조한다(S5). 그리고, 대응되는 FlexRay 사이클(Cycle) 내의 슬롯 번호와 이와 연관된 FlexRay 메시지 버퍼

(435)의 주소를 찾아내어 데이터와 도착시간을 FlexRay 메시지 버퍼에 써 넣게 된다(S6).

- [0030] FlexRay 메시지 버퍼에 새로운 데이터가 기록될 때 인터럽트가 발생될 수 있다(S7). 이후, 메시지 도착에 대한 인터럽트의 서비스 루틴 (439)이 발생하여 FlexRay 컨트롤러에 메시지 도착에 대한 작업 처리를 요청할 수 있다. 또한, 앞서 CAN 메시지 수신 과정에서 살펴본 바와 같이, 특정 태스크에 의한 주기적인 폴링에 의해 메시지 버퍼(435)에 데이터의 도착했음을 확인할 수 있다. 즉, 해당 인터럽트 서비스 루틴(439) 또는 특정 태스크에서 메시지 버퍼 내의 데이터를 전송하기 위해 FlexRay 컨트롤러에 전송 요청하게 된다(S8).
- [0031] FlexRay 컨트롤러는 메시지 버퍼(435)에 저장된 메시지 내의 데이터(A)와 CAN 버스 측에서 보내온 메시지 도착 시간(타임스탬프의 의미를 부여)을 FlexRay 프레임의 형태(F_A)로 가공하여 해당 슬롯에 넣어서 FlexRay 버스(431)를 이용하여 전송을 수행한다(S9).
- [0032] FlexRay 버스(431) 상에 CAN 버스 측에서 보내온 메시지를 수신하는 노드(480)에서 FlexRay 프레임으로부터 데이터와 타임스탬프를 추출하며(S10), 수신 FlexRay 노드(480)에서 FlexRay 프레임 도착시간과 타임스탬프의 차이를 계산하고 만일 시간 차이가 허용 한계치를 넘어선 경우 게이트웨이에 향의 메시지를 전송한다(S11). FlexRay 노드(480)로부터 향의 메시지를 수신한 게이트웨이(400)는 작업 스케줄러(450)에게 이를 통보하며, 작업 스케줄러(450)는 향의 메시지를 전달 받은 후에 메시지 지연을 평가하는 작업(메시지 우선순위 상승 인자, Message Priority Raising Factor, MPRF)를 계산하며, MPRF의 계산 결과가 사전에 미리 결정된 허용 한계를 넘어선다고 판단될 경우 작업 스케줄러(450)는 해당 메시지를 전송한 CAN 버스의 특정 노드가 현 시점 이후에 보내는 메시지를 처리하게 될 메시지 디스패처에 대해 태스크 우선순위를 한 단계 상향 조정할 것을 지시한다(S12). 이후, 태스크 우선 순위가 상향 조정된 메시지 디스패처(예를 들어 메시지 디스패처 #1, 451)는 작업 스케줄러(450)가 지시한 사항에 따라, CAN 버스의 특정 노드가 이후 보내는 메시지를 처리함에 있어, 상향 조정된 태스크 우선 순위에 따라 작업을 진행할 수 있다. 그 결과, 작업 스케줄러(450)는 해당 메시지 디스패처 #1(451)를 우선 처리할 수 있다.
- [0033] 한편, 작업 스케줄러(450)는 앞서 S1-S12의 과정을 반복하며 MPRF를 계산하여 허용 한계치 이하가 되는지 관찰하며 만일 허용 한계치 이하로 내려가면 메시지 디스패처의 태스크 우선순위를 현 상태로 고정하도록 하며 그렇지 않을 경우에는 태스크 우선순위를 단계적으로 상향 조정하며 추이를 관찰할 수 있다. 또한, 본 명세서의 다른 실시예에 의하여, MPRF가 일정 수준보다 낮게 내려갈 경우, 해당 메시지 디스패처의 태스크 우선 순위를 최초의 상태로 낮추도록 할 수도 있다.
- [0034] 한편, 메시지 디스패처의 우선순위를 최대한 높였음에도 불구하고 해당 메시지의 MPRF가 허용치 아래로 내려가지 않을 경우에는 차량 진단 모듈에 이러한 사실을 통보하여 추후에 게이트웨이 운용 정책을 수정할 수 있는 확인 자료를 저장할 수 있는 기회를 제공한다.
- [0035] 도 4에서는 CAN 버스 측에서 FlexRay 버스 측으로의 메시지 전송에 관한 것만을 나타내고 있으나 역방향으로의 메시지 전송에서도 유사한 기능을 하게 된다. 즉, 역방향으로의 메시지 전송에서도 유사한 순서로 동작하게 되는데, 작업 스케줄러는 메시지 디스패처를 제어함에 있어서, FlexRay 메시지 버퍼(435)에 저장된 메시지를 CAN 메시지 송신 큐(426, 428)에 송신하여 이를 CAN 노드에 전달하는 과정에서 FlexRay 컨트롤러(430)가 기록한 메시지 도착 시간 또는 FlexRay 메시지 버퍼(435)에 메시지가 기록된 도착 시간과 CAN 노드들로부터 발생한 향의 메시지를 이용하여 MPRF를 계산하고, 특정 FlexRay 노드를 통해 송신되는 메시지가 허용 기준 이상 지연되는 경우, 이를 해결하기 위해 메시지 디스패처의 우선 순위를 상향 조정시킬 수 있다.
- [0036] 도 4에서 제시된 구성에 의하여, 게이트웨이 내에서의 전송 지연을 정량적으로 측정할 수 있으며, 전송 지연 시간이 시스템의 허용 한계를 넘어서는 경우에 이를 극복할 수 있다. 도 4에서는 CAN-FlexRay 게이트웨이에서 특히 전송 지연이 문제가 되는 CAN 버스에서 FlexRay 버스로 메시지가 전달될 경우의 구성에 대해 설명하고 있으며, 그 결과, CAN 버스에서 FlexRay 버스로의 메시지 전달 과정에 중점을 두고 있으나, 이는 역방향으로의 메시지 전달과정에도 도 4의 구성을 그대로 활용할 수 있다.
- [0037] 즉, 도 4의 구성에서 작업 스케줄러(450) 또는 메시지 디스패처(451, 452, 453, 454) 등은 게이트웨이 내에서 메시지 변환과 전달 과정에 발생하는 전송지연을 정량적으로 측정하기 위해 CAN 메시지가 게이트웨이에 도착할 때의 시간을 파악해 두었다가 FlexRay 프레임을 생성할 때 이를 타임스탬프 (Time Stamp)로 삽입하여 전송하도록 한다. 이러한 프레임을 수신한 FlexRay 노드(480)에서는 타임스탬프 값과 현재의 시간을 비교하여 허용 한계 내에 도착하였는지를 판단하도록 한다. 만일 허용 한계를 넘어설 경우에는 해당 노드(480)에서 게이트웨이에 향

의 메시지(Complaint Message)를 전송하도록 한다.

[0038] 항의 메시지를 전달 받은 게이트웨이는 지연되어 전송된 CAN 메시지를 전송하는 CAN 버스의 노드가 전송하는 메시지를 처리할 때 높은 우선순위를 부여하여 다른 메시지들 보다 우선적으로 처리되도록 한다. 게이트웨이 내에서 특정 메시지를 우선적으로 처리하는 방법은 해당 메시지를 처리하는 특정 태스크와 인터럽트 서비스 루틴(ISR)에 더 높은 우선순위를 부여하는 것을 통해 구현될 수 있다. 즉, 앞서 살펴본 바와 같이 지연되어 전송된 CAN 메시지를 전송하는 CAN 버스의 노드에서 이후 전송하게 되는 메시지를 처리할 메시지 디스패처의 우선 순위를 높게 하여, 다른 메시지 디스패처 보다 우선적으로 작업을 진행할 수 있도록 할 수 있다.

[0039] 도 4에 미도시 되었으나, 인터럽트 서비스 루틴에서도 해당 CAN 버스의 노드에서 메시지를 전송할 경우, 인터럽트의 우선순위를 상향 조정하여 보다 빨리 처리되도록 할 수 있다.

[0040] 메시지의 전송과 관련하여 메시지 전송이 지연되었는지 확인하는 기준인 허용 한계치에 대해 살펴보면 다음과 같다.

[0041] 게이트웨이의 기본적인 기능에 따라 타임스탬프를 포함하는 FlexRay 프레임이 목적지 노드에 도착하게 되고, 목적지 노드에서는 타임스탬프와 현재 시각을 비교하여 시간차를 계산하게 되며 이를 허용 한계치와 비교하게 된다. 허용 한계치는 시스템 설계 단계에서 미리 결정되어야 하며 수학적 식 1과 같은 범위를 가진다.

[0042] [수학적 식 1]

$$\text{게이트웨이 Latency} \leq \text{허용 한계치} < (\text{게이트웨이 Latency} + \text{FlexRay Cycle Time})$$

[0043] 허용 한계치는 게이트웨이 통과에 의해 발생하는 지연 시간(게이트웨이 Latency)과 FlexRay Cycle의 주기(FlexRay Cycle Time)를 합한 시간 보다 적어야 하며 이를 넘어서게 될 경우 항의 메시지를 게이트웨이 측으로 전송하게 된다.

[0045] 항의 메시지를 수신한 게이트웨이는 작업 스케줄러(450)에게 이를 통보하며 작업 스케줄러(450)는 자신이 관리하고 있는 메시지 디스패처 태스크(451, 452, 453, 454)들과 인터럽트 서비스 루틴(429)의 실행 우선순위를 조절하기 위해 메시지 우선순위 상승 인자(Message Priority Raising Factor, MPRF)를 계산하게 된다.

[0046] MPRF는 CAN 메시지의 ID에 의해 결정되는 CAN 메시지 고유의 메시지 우선순위를 포함하고 이를 처리하는 태스크의 우선순위를 의미하며 다음의 수학적 식 2에 의해 계산된다.

[0047] [수학적 식 2]

$$\text{MPRF} = \begin{cases} \text{Complaints Counter} + \text{Message Priority}, & \text{when Complaints Counter} > 0 \\ 0, & \text{otherwise} \end{cases}$$

[0048] 위의 수식에서 Complaints Counter 는 작업 스케줄러가 받은 항의 메시지의 수를 의미하며 이것이 하나 이상일 때 CAN 메시지의 우선순위를 더하여 MPRF를 산출하게 된다.

[0050] CAN 버스에서 보내온 메시지의 우선순위는 자신이 가지고 있는 ID에 따라 결정 되지만, 이들이 게이트웨이의 동작에 의해 지연되어 전달될 경우에 우선순위가 높은 메시지의 지연이 우선순위가 낮은 메시지의 지연보다 시스템에 보다 큰 영향을 끼치게 되므로 이를 반영하기 위해 MPRF의 계산에서 CAN 메시지 본연의 우선순위를 더하게 된다.

[0051] MPRF의 한계치는 시스템 설계 단계에서 미리 결정되어야 하며 항의 메시지가 없으면 0으로 처리된다. 메시지 디스패처 태스크(451, 452, 453, 454)들과 주기적 태스크, 인터럽트 서비스 루틴(429)의 우선순위가 가변될 수 있는 범위 역시 시스템 설계 단계에서 결정되어야 하며 MPRF에 영향을 받지 않고 최고의 우선순위를 유지해야 하는 ISR과 메시지 디스패처도 미리 결정해야 한다.

[0052] 게이트웨이 내에서는 한 쪽 버스에서 수신된 다수의 메시지들을 상대측 버스에 전달하기 위해 메시지를 변환하

는 과정과 이를 메시지 저장소 간에 이동시키는 과정들이 다수의 메시지 디스패처 태스크(451, 452, 453, 454)들과 인터럽트 서비스 루틴(429) 또는 주기적 태스크의 실행에 의해 구현되고 있다. 이들 메시지 디스패처 태스크(451, 452, 453, 454)들과 인터럽트 서비스 루틴(429) 또는 주기적 태스크들은 운영체제와 작업 스케줄러(450)에서 부여하는 우선순위에 의해 상호간에 순서가 정해져서 실행되며 우선순위가 높은 메시지 디스패처 태스크나 인터럽트 서비스 루틴 또는 주기적 태스크는 다른 메시지 디스패처 태스크나 인터럽트 서비스 루틴 또는 주기적 태스크 보다 먼저 실행됨으로써 이를 통해 전달되는 메시지가 다른 메시지들 보다 빨리 전송되는 효과를 발생시킨다. 작업 스케줄러의 관점에서 본다면, 작업 스케줄러는 메인 코어 CPU 상에서 실행되며 CAN 컨트롤러나 FlexRay 컨트롤러가 단독으로 제어하는 기능을 제외한 모든 메시지 디스패처 태스크나 인터럽트 서비스 루틴 또는 주기적 태스크의 우선순위를 변경할 수 있는 자격을 가지고 있으며 이것은 프로세서 내에서 구동되고 있는 운영체제에 우선순위 변경을 요청함으로써 구현된다.

- [0053] 본 명세서의 일 실시예에 의한 작업 스케줄러(450)는 CAN-FlexRay 간의 메시지 전송에서 지연이 발생할 경우, 메시지를 전송하는 태스크의 우선순위를 변경할 수 있으며, 이 과정에서 CAN 메시지 자체의 우선순위를 바꾸지는 않는다. 또한, 앞서 도 2, 3, 4에서 본 바와 같이 복수개의 CAN 버스가 게이트웨이에 접속되어 있고 각각의 CAN 버스에 수많은 CAN 노드들이 접속된 상황에서는 각 CAN 버스의 우선순위 체계를 따르는 다수의 CAN 메시지들이 동시에 게이트웨이에 입력될 수 있기 때문에 CAN 메시지의 우선순위에 따라 이를 처리하는 태스크의 우선순위를 바꾸는 것은 먼저 도착한 메시지가 먼저 처리되어야 전송 지연이 최소화 된다는 원칙에 위배될 수 있다.
- [0054] 그러므로 본 명세서의 일 실시예에서는 각각의 메시지들의 도착 순서에 따라 우선적으로 처리해야 한다는 원칙을 따르며 다만, 지나치게 지연되어 도착하는 메시지들에 대해 제한적으로 이를 처리하는 메시지 디스패처 태스크나 인터럽트 서비스 루틴 또는 주기적 태스크의 우선순위를 상향 조정함으로써 시스템의 동작에 최소한의 영향을 끼치면서 메시지 전송 지연을 보상하고자 한다. 즉, 작업 스케줄러는 기본적으로 CAN 버스를 통해 메시지가 수신되는 순서대로 인터럽트 서비스 루틴이나 주기적 태스크 또는 메시지 디스패처가 실행되도록 내버려 두지만 MPRF가 허용 한계치를 넘어서는 메시지를 보내는 CAN 노드에 대해서는 실행 우선순위를 바꿔주게 된다. FlexRay 버스 측에서 CAN 버스 측으로의 메시지 전달에서도 동일한 동작을 하게 된다.
- [0055] 작업 스케줄러는 MPRF가 허용 한계치를 넘어서는 메시지를 보내고 있는 CAN 노드가 현 시점 이후에 보내는 메시지에 대해 이것을 처리할 메시지 디스패처 및 인터럽트 서비스 루틴 또는 주기적 태스크의 실행 우선순위를 다른 태스크나 인터럽트 서비스 루틴보다 한 단계 높여주게 되며 그 결과를 MPRF를 재계산하며 확인하게 된다.
- [0056] 실행 우선순위를 높여주는 과정은 아래의 a), b)를 반복하여 진행할 수 있다.
- [0057] a) 태스크의 우선순위를 높이며 MPRF를 계산
- [0058] b) a)에 의해 MPRF의 값이 허용 한계치보다 작아지지 않을 경우 관련된 인터럽트와 이의 인터럽트 서비스 루틴에 대한 우선순위를 높이며 MPRF를 계산
- [0059] a), b)의 순서를 지정하는 이유는 인터럽트와 인터럽트 서비스 루틴의 우선순위를 변경하는 것보다 일반 태스크의 우선순위를 변경하는 것이 전체 시스템의 운용에 미치는 영향이 작다는 것에 기인한다.
- [0060] 만일 태스크와 인터럽트 서비스 루틴의 우선순위를 한 단계 높였을 때 MPRF 허용 한계치 이하로 떨어지면 현 상태에서 우선순위를 고정하게 되지만 그렇지 않을 경우에는 허용 한계치 이하로 떨어질 때까지 우선순위를 반복적으로 높여주게 된다.
- [0061] 태스크와 인터럽트 서비스 루틴의 우선순위가 이들이 가질 수 있는 최고 우선순위에 이르도록 MPRF가 개선되지 않으면 메시지의 전달에 있어서 메시지 디스패처를 거치지 않도록 하여 전송 시간을 단축시키도록 한다.
- [0062] 앞서 수차례 CAN 메시지 큐와 FlexRay 메시지 버퍼 간에 데이터 교환이 있었으므로 그 동안에 메시지 디스패처는 인터럽트 서비스 루틴에서 활용할 수 있는 전역 변수나 포인터에 상대 측 메시지 버퍼의 주소를 미리 저장해둠으로써 메시지 디스패처가 개입하지 않더라도 직접적인 데이터 전송이 가능하도록 한다. 그러나, 인터럽트 서비스 루틴에 의한 데이터의 직접 전송은 이러한 비상 상황에서만 활성화 되도록 하며 일상적인 상황에서는 비활성화 됨으로써 인터럽트 처리가 빠르게 진행되도록 한다. 이를 위해 작업 스케줄러가 인터럽트 서비스 루틴으로 하여금 비상 상황임을 알아챌 수 있도록 특정 플래그(플래그, Flag)를 설정하도록 하며, 인터럽트 서비스 루틴에서는 해당 플래그가 설정되었음을 확인한 후에 이에 따라 직접 전송 루틴을 실행시키도록 한다. 이상에서 기술한

내용이 실행되는 순서는 다음과 같다.

- [0063] i) 작업 스케줄러가 비상 상황임을 알리는 플래그를 설정
- [0064] ii) CAN 메시지가 수신되면 인터럽트가 발생하고 이의 ISR에서 데이터를 메시지 큐에 저장
- [0065] iii) ii)의 결과로 인터럽트가 발생함.
- [0066] iv) iii)에서 발생한 인터럽트의 ISR에서 플래그를 확인
- [0067] v) 만일 플래그가 설정되어 있다면 상대측 메시지 버퍼의 주소가 저장된 전역 변수를 확인한 후에 메시지 큐에 저장된 데이터를 직접적으로 목적지 메시지 버퍼에 전달
- [0068] vi) v)의 결과로 메시지 버퍼에서 인터럽트가 발생
- [0069] vii) 메시지 버퍼 ISR에서는 FlexRay 컨트롤러로 하여금 즉시 프레임 전송할 수 있도록 요청

- [0070] 상기 i)~vii) 과정을 거치는 동안 MPRF 값이 허용 한계치 이하로 내려가지 않으면 진단 모듈에 이러한 상황을 통보하여 추후에 개발자가 이를 개선할 수 있는 여지를 마련해주도록 한다. 반대로, MPRF가 허용 한계치 이하로 내려가면 태스크의 우선순위를 한 단계씩 내리며 MPRF를 측정하고 MPRF가 허용 한계치를 넘지 않는 제일 낮은 우선순위를 태스크에 부여하도록 한다.

- [0071] 실행 우선순위를 낮추는 과정은 a-i) 먼저 ISR의 우선순위를 낮추며 MPRF를 계산한 후, a-ii) 계산된 MPRF의 값이 허용 한계치보다 작아질 경우 관련된 메시지 디스패처에 대한 우선순위를 낮추며 MPRF를 계산한다. 그리고 이 과정(a-i, a-ii)을 반복하여 진행할 수 있다. 이러한 순서를 지정하는 이유는 인터럽트와 ISR의 우선순위를 변경하는 것이 일반 태스크의 우선순위를 변경하는 것보다 전체 시스템의 운용에 미치는 영향이 크다는 것에 기인한다.

- [0072] 도 5는 본 명세서의 일 실시예에 의한 CAN 메시지가 FlexRay 노드에 전송되는 과정에서 발생하는 메시지 지연을 처리하는 과정을 보여주는 도면이다. 각각의 과정을 살펴보면 먼저, CAN 버스로부터 메시지를 수신하고, CAN 컨트롤러에 의해 인터럽트가 발생된다(S502). 그리고 발생한 인터럽트를 처리하기 위한 인터럽트 서비스 루틴에서 CAN 메시지지에서 데이터를 추출한 후, 이를 CAN 메시지 큐에 저장하고, 메시지 도착 시간을 별도로 저장한다(S510). 저장한 후, 인터럽트 서비스 루틴은 작업 스케줄러를 호출하고 복귀한다(S512). 이후 인터럽트 서비스 루틴은 다른 인터럽트를 처리하기 위해 대기할 수 있다. 그리고 작업 스케줄러는 수신한 메시지를 전송하기 위해, 특정 메시지 디스패처에 작업을 할당하고, 메시지의 도착 시간(타임 스탬프)를 전달한다(S520). 메시지 디스패처는 LUT를 참고하여 FlexRay 메시지 버퍼의 주소를 확인한다(S522). 확인한 후, 메시지 디스패처가 해당 주소의 FlexRay 메시지 버퍼에 데이터와 도착 시간을 기입한다(S524). 이후, FlexRay 메시지 버퍼에서 인터럽트를 발생시킨다(S530). 그리고 발생한 인터럽트를 처리하기 위한 인터럽트 서비스 루틴에서 FlexRay 컨트롤러에 버퍼 내 데이터 전송을 요청한다(S532). 요청을 수신한 FlexRay 컨트롤러는 데이터와 도착 시간을 프레임으로 가공한 후, 의도된, 즉 해당 메시지를 FlexRay 노드가 수신할 수 있도록, 데이터와 도착 시간을 프레임으로 가공한 후, 의도된 슬롯을 통해 전송한다(S534). 그리고 수신 노드에서 프레임 도착 시간과 타임 스탬프를 비교하고(S540), 그 결과 시간 차이가 허용 한계보다 큰지 혹은 작은지 확인한다(S550). 만약 허용 한계보다 작다면, 메시지가 지연되지 않았으므로 다음 프로세스를 진행하게 된다.

- [0073] 한편, 시간 차이가 허용 한계보다 큰 경우에는 수신 노드는 게이트웨이에 향의 메시지를 전송한다(S560). 게이트웨이는 작업 스케줄러에게 향의 메시지를 전달하고 MPRF를 계산한다(S562). 그리고 MPRF가 허용 한계보다 클 경우(S570) 메시지 디스패처의 우선 순위를 조절하게 되는데, 해당 메시지의 디스패처의 우선순위가 최상인지 확인한다(S580). 그리고 최상이 아니면, 해당 메시지 디스패처의 우선 순위를 높여준다(S582). 이후 해당 메시지 디스패처는 해당 노드를 통해 메시지를 수신할 경우 보다 빨리 처리할 수 있다. 한편, S570에서 MPRF가 허용 한계보다 작은 경우, 메시지가 지연되었으나, 디스패처의 우선 순위를 상승시킬 수준은 아니므로 다음 프로세스를 진행한다. 이후 지연이 누적될 경우, 다시 S570 단계에서 디스패처의 상승 조정 여부를 확인할 수 있다. 한편, S580에서 메시지 디스패처의 순위가 최상인 경우, 더 이상 조절할 수 없기 때문에, 이를 진단 모듈에 통보한다(S590). 이는 차후, 시스템 설계시 혹은 시스템 변경시 적용할 수 있도록 하기 위함이다.

[0074] 종래에는 단순히 CAN 메시지의 ID와 FlexRay 슬롯 간의 메시지를 교환하는 과정에서 이종의 버스 사이에 위치한 게이트웨이 내의 메시지 전달 과정에 발생하는 전송 지연을 고려하고 있지 않다. 특히, 한쪽 버스의 노드에서 전송한 메시지가 상대 측 버스에 접속된 노드에 도달할 때까지의 시간에 더해진 전송 지연을 정량적으로 측정하는 방법을 고려하고 있지 않으며 더구나 전송 지연이 전체 네트워크 시스템의 허용 한계를 넘어서는 경우에 대한 대책을 제시하지 않고 있다.

[0075] 본 명세서의 일 실시예에 의한 이종 프로토콜 통신에서 프로세스의 우선 순위를 조절하여 지연을 보상하는 방법 및 장치에서는 종래의 방법들이 고려하지 않은 게이트웨이 내에서의 전송 지연을 정량적으로 측정하는 방법과 전송 지연 시간이 시스템의 허용 한계를 넘어서는 경우에 이를 극복할 수 있는 방법을 제시한다. 그러므로 제안한 방법이 차량 내 게이트웨이에 적용될 경우 차량 내 네트워크의 신뢰성을 극대화 시킬 수 있게 된다.

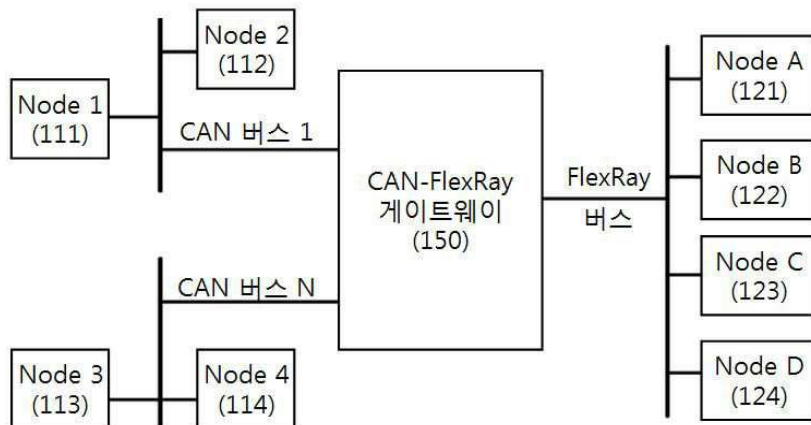
[0076] 이상의 설명은 본 발명의 기술 사상을 예시적으로 설명한 것에 불과한 것으로서, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것이다. 따라서, 본 발명에 개시된 실시 예들은 본 발명의 기술 사상을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시 예에 의하여 본 발명의 기술 사상의 범위가 한정되는 것은 아니다. 본 발명의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

부호의 설명

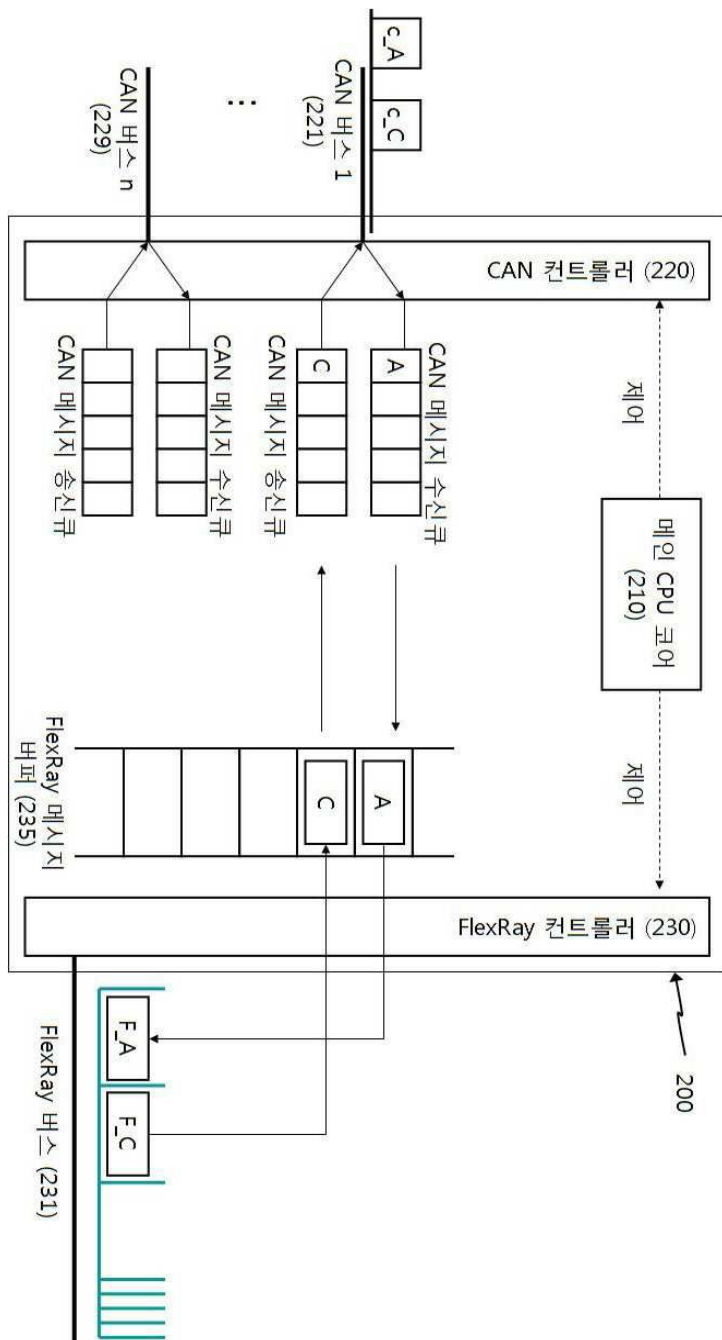
- [0077] 300, 400: CAN-FlexRay 게이트웨이
- 420: CAN 컨트롤러
- 430: FlexRay 컨트롤러

도면

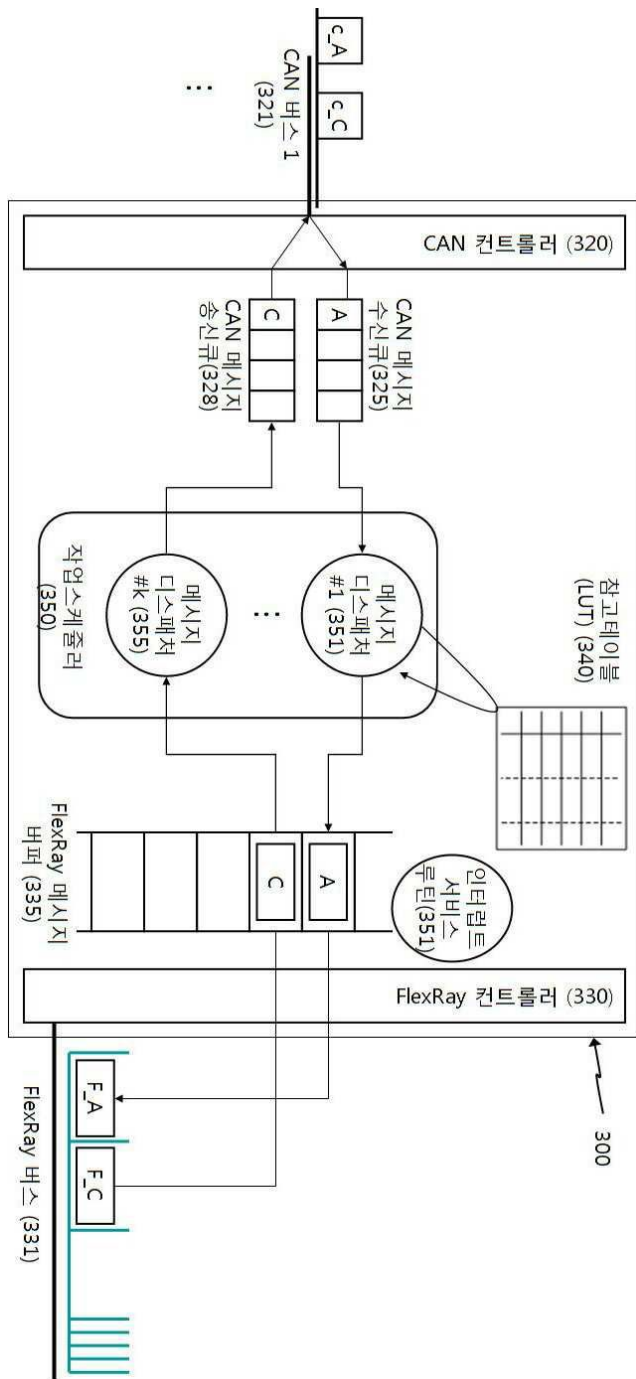
도면1



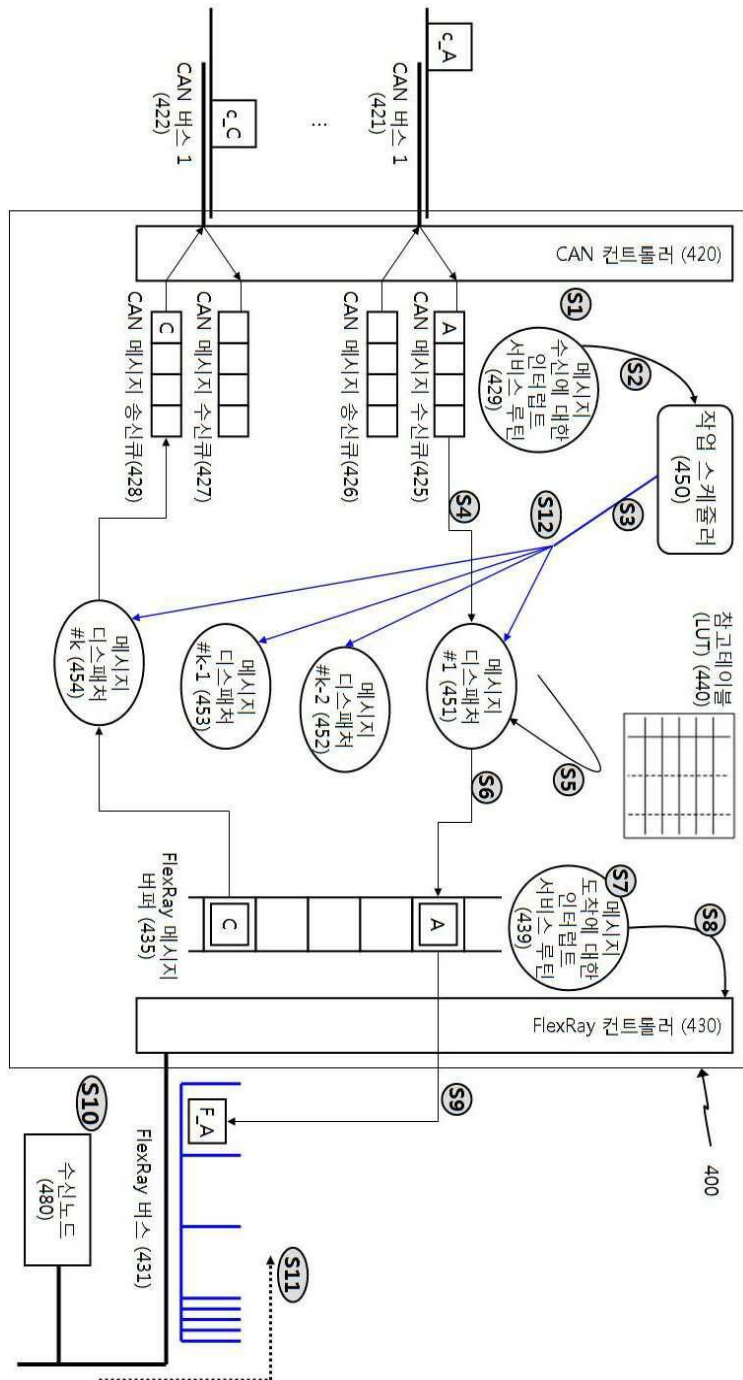
도면2



도면3



도면4



도면5

