

CAN의 분산 선행대기 열 기법을 이용한 선박 엔진 모니터링 시스템

Marine Engine State Monitoring System using DPQ in CAN Network

이 현, 이 준 석, 이 장 명*
(Hyun Lee¹, Jun-Seok Lee¹, and Jangmyung Lee¹)
¹Pusan National University

Abstract: This paper proposes a marine engine state monitoring system using a DPQ (Distributed Precedence Queue) mechanism which collects the state of bearings, temperature and pressure of engine through the CAN network. The CAN is developed by Bosch Corp. in the early 1980' for automobile network. The data from various sensors attached in the marine engine are converted to digital by the analog to digital converter and formatted to fit the CAN protocol at the CAN module. All the CAN modules are connected to the SPU (Signal Processing Unit) module for the efficient communication and processing. This design reduces the cost for wiring and improves the data transmission reliability by recognizing the sensor errors and data transmission errors. The DPQ mechanism is newly developed for the performance improvement of the marine engine system, which is demonstrated through the experiments.

Keywords: CAN, ESMS (Engine State Monitoring System), DPQ

I. 서론

선박에서 엔진의 동작상태에 대한 감시와 경보에 빠르게 대처하고 효율적인 관리를 위해서 각 부분의 상태를 점검하는 것이 중요하다. 하지만 현실적으로 엔진 속의 실린더, 베어링, 각각의 밸브 및 밸브 속의 가스 상태, 냉각오일 등의 상태 점검을 위해 엔진을 분해 하기에는 많은 비용과 시간이 소요되며 그런 점검을 하기 위해서 엔지니어가 직접적으로 노출되기 때문에 안전사고와 직접적으로 연결이 된다. 이와 같은 이유로 엔진의 베어링, 온도 등의 상태를 진단하고 실시간으로 모니터링 하여 선박의 운항에 위험요소를 사전에 방지하는 중요한 시스템이므로 시스템 오동작 및 시스템다운 등에 대한 자체 진단과 보호회로 등을 고려하고 신뢰성이 확보가 되어야 한다. 이런 외부 환경요소가 미치는 영향을 최소화 하기 위한 구조적인 기술 분야 및 시스템의 정확한 동작에 영향을 미칠 수 있는 외부 요소들에 대해 보호 될 수 있는 전기, 전자의 회로 및 프로그램 설계에 대한 기술 분야가 요구 된다. 그리고 선박통합화 네트워크 구성, 무선통신을 통한 전기배선의 간단화, 시스템의 원격감시, 선박 엔진 제어 및 통신의 통합화 등이 절실하게 되고 있다[1,2].

Bearing State Monitoring System의 기본 원리는 거대한 2행정 디젤 엔진의 크로스헤더와 거리센서 표면 사이 거리를 지속적으로 측정 함으로서 거리의 변화에 따른 베어링 마모 정도를 감지 할 수 있다. 그리고 엔진 내부의 실린더 배출 가스 온도, 냉각 청수(cooling fresh water) 배출구 온도, 메인 베어링 온도 및 선박 엔진용 피스톤 냉각오일 흐름감지 스위치(PCO non-flow switch) 신호 등을 CAN 통신을 통하여 실시간 모니

터링 할 수 있다. 이와 같이 엔진 베어링(crosshead bearing, crank bearing, main bearing)들의 마모 정도와 온도 및 냉각오일 흐름 등을 감지하여 엔진의 상태를 실시간으로 감시 하여 예견치 못한 피해를 사전에 예방 할 수 있어서 선박의 중요 시스템으로서 요구 되고 있다[3-5].

본 논문은 선박의 Engine Room과 Engine Control Room의 모니터링 시스템과 연동할 수 있는 선박 엔진상태 모니터링 시스템의 개발을 목적으로 하였다.

II. CAN (Controller Area Network) 분석

1. CAN의 특징

CAN은 차량 내에서 컨트롤러, 센서, 액츄에이터, ABS, 오디오 시스템, 엔진 제어 시스템 등을 연결하기 위해 1986년 Bosch사에서 개발한 시리얼 통신 프로토콜이며 Human-Machine Interface 간의 통신을 관리하는 합리적인 방법이다. CAN은 높은 데이터 전송률과 안정성을 제공하며 거리에 따라 다르지만 최고 1Mbps의 높은 데이터 전송률과 충돌한 메시지의 자동 재전송과 15bit Cycle Redundancy Check (CRC)를 이용한 오류 검출등과 같은 안정성을 제공하고 있어서 다수의 전자 제어 장치를 상호 연결하여 실시간 제어를 효율적으로 지원할 수 있는 분산 제어 네트워크이다.

임베디드 시스템(또는 마이크로 컨트롤러)에서 일반적으로 사용되는 CAN 버스는 마이크로 컨트롤러 사이에서 통신망을 형성하며, 2가닥의 꼬임선(Twist Pair Wire)으로 연결되어 반이중 통신(Half Duplex) 방식으로 짧은 메시지를 사용하는 고속 응용 시스템에 적합하다. 더불어 외부의 요인(노이즈 등)에 강인성을 가져 통신 에러율을 최소화 하여 높은 신뢰성을 가지고 있다. 이론적으로는 2032개의 서로 다른 디바이스(임베디드 컨트롤러)를 하나의 네트워크상에 연결하여 통신을 수행할 수 있으나 CAN 트랜시버의 한계로 인하여 110개까지의 노드를 연결하여 사용할 수 있다.

* 책임저자(Corresponding Author)

논문접수: 2010. 11. 02., 수정: 2010. 12. 29., 채택확정: 2011. 11. 15.

이현, 이준석, 이장명: 부산대학교 전자전기공학과

(hleec@pusan.ac.kr/sukidda@pusan.ac.kr/jmlee@pusan.ac.kr)

※ 이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음.

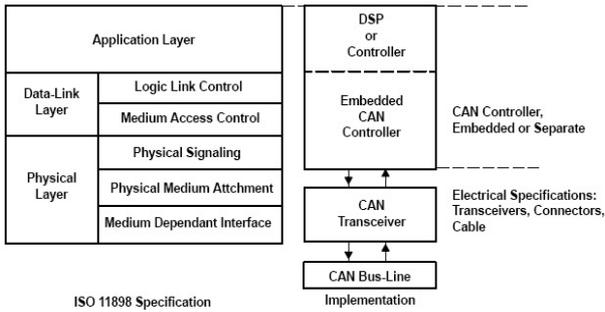


그림 1. ISO 11898 표준 구조.
Fig. 1. ISO 11898 Standard structure.

CAN 규약은 ISO 11898 (그림 1)에 의해 표준화 되었으며 CiA (CAN in Automation) 협회에서는 CAN을 기반으로 하여 ISO/OSI 계층모델에서 응용 계층에 해당하는 통신망 규약을 개발하였다(CANopen). CAN에 의해 데이터가 교환될 때 어떠한 노드도 주소화 되지 않고 통신망에서 유일하게 존재하는 메시지 식별자(ID: Identifier)에 의해서 메시지의 내용(예를 들면 엔진 온도 등)과 같은 데이터가 교환된다. 즉, 데이터 혹은 메시지는 노드의 물리적 주소로 그 데이터의 수신지를 정하지 않고 메시지 ID를 이용하여 그 데이터의 사용처를 정한다.

2. CAN의 특징

그림 2는 표준 CAN 메시지 구조이다. 표준 CAN 메시지의 구조는 메시지의 처음 시작을 표시하며 메시지 프레임의 최우선에 위치하는 SOF (Start of Frame)와 11비트의 식별자와 원격 전송 요구(RTR) 비트를 가지며 디폴트 “0”을 가지는 RTR 비트는 비트값이 “0”일 때 CAN 메시지가 데이터 프레임이라는 것을 가리킨다. 역으로 RTR 비트 값이 “1”이면 CAN 메시지가 원격전송요청(RTR: Remote Transmission Request)을 의미한다. CAN 메시지가 데이터 프레임이 아닌 원격프레임(remote frame) 상태임을 나타내는 중재 필드(arbitration field) 그리고 6비트로 구성되며 향후에 사용되기 위해 예약된 두개의 “0”의 값을 가지는 R0, R1과 데이터 필드의 byte 수를 가리키는 4비트의 데이터 길이 코드(DLC: Data Length Code)로 구성된 데이터의 길이를 의미 하는 제어 필드(control field)와 한 노드로부터 다른 노드로 전하고자 하는 8byte의 데이터 필드(data field)로 구성된다.

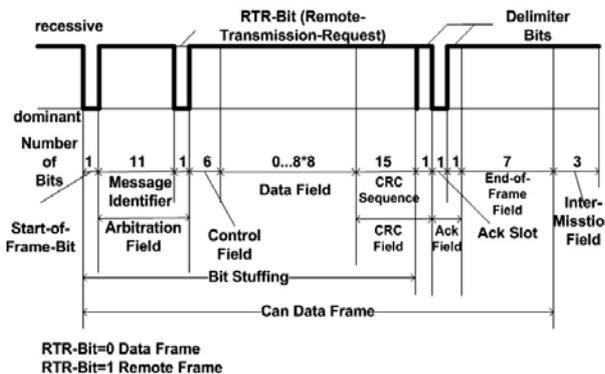


그림 2. 표준 CAN 메시지 구조.
Fig. 2. Standard CAN Message structure.

마지막으로는 에러 검출을 위한 CRC (Cyclic Redundancy Check)필드와 종료점을 알리는 EOF (End of Frame Field)로 구성된다.

CAN은 ISO 11898 규격에 따라 5Kbps에서 1Mbps까지 다양한 전송속도를 제공하며, 토폴로지는 버스형(bus structure)과 스타형(star structure)을 지원한다. 이것은 스테이션에서 고장이 발생하더라도 시스템의 일부는 사용할 수 있어 고장의 파급효과를 줄일 수 있다.

3. 기본 CAN의 프로토콜

CAN은 CSMA/CD 매체접근 기술을 기반으로 개발되었으며, 네트워크 상에서 충돌을 해결하기 위해서 송수신되는 메시지의 우선순위를 ID로 지정하는 기법을 사용하고 있다. CAN 프로토콜은 OSI 기준모델을 기초로 하여 계층(layer) 형태로 구성된다. 그러나 OSI의 전 계층으로 구성되어 있지 않고 공장 자동화 환경과 관련된 부분만을 채택하여 다음의 3 계층으로 이루어져 있다.

1. 응용 계층(application layer): 사용자가 네트워크에 접근할 수 있도록 지원.
2. 데이터 연결 계층(data link layer): 물리적인 주소를 상-하위 계층으로 연결.
3. 물리 계층(physical layer): 물리적인 매체를 통해 비트 흐름을 전송.

본 논문에서는 데이터 연결 계층을 활용하여 전송시간 지연 문제를 해결하고자 하며, 이 데이터 연결 계층을 이루는 MAC (Medium Access Control) 및 LLC (Logical Link Control) 중 LLC 부-계층(sub-layer)만을 활용한다.

4. CAN의 물리 계층

CAN 네트워크는 버스 토폴로지의 형태로 구성된다. 그리고 버스는 두 개의 상호 보완적인 Dominant와 Recessive의 값으로 구성된다.

Dominant 값은 논리적으로 0의 값을 갖고 Recessive 값은 1에 상응하는 값을 갖는다. 동시에 몇 개의 개체가 Dominant와 Recessive 값을 전송 한다면 버스상의 결과 값은 0의 값인 Dominant 값 만이 전송된다. 다시 말하면 CAN은 전자회로에서 open-collector와 유사한 방법으로 구성된다.

5. LLC 부-계층

LLC 부-계층은 데이터 연결 계층의 상위 부분으로 프레임 필터링 및 복구 기능을 담당한다. LLC 부-계층 하부의 MAC 부-계층은 수신된 프레임의 Identifier field에서 어떤 값을 체크하지는 않고 간단히 수신된 값만을 읽는다. CAN 프로토콜에서 ID는 메시지의 목적지 주소를 나타내는 것이 아니고 메

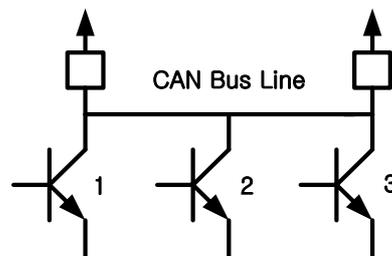


그림 3. CAN의 물리 계층.
Fig. 3. CAN physical layer.

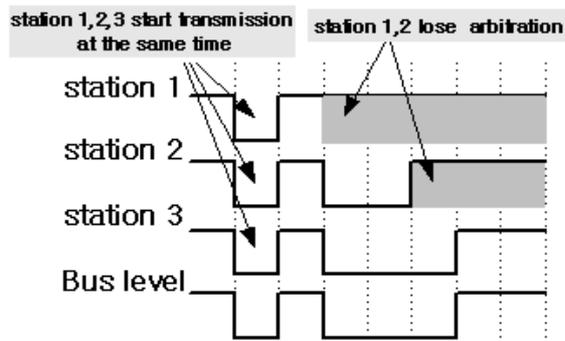


그림 4. CAN 조정자 위상.
Fig. 4. CAN arbitration phase.

시지의 종류를 나타낸다는 것을 주의해야 할 것이다. LLC 부-계층은 수신된 프레임의 수송할 지 아니면 수송하지 않을 지를 결정하는 기능을 가지고 있다. 또한 LLC는 충돌이 발생하여 전송자격을 잃었는지 아니면 에러가 발생해서 프레임을 재전송 해야 할 지를 조정하는 역할을 담당한다. 그리고 사용자는 메시지가 성공적으로 전송될 때까지 전송 결과를 알지 못한다.

III. 분산된 선행 대기 열 기법(DPQ)

CAN은 네트워크 상에서 각각의 메시지의 우선순위를 ID로 절대적으로 할당한다. 그러나 이러한 기법은 몇 개의 개체가 동시에 전송을 시작할 때 일어나는 충돌을 해결할 수는 있으나 각각의 개체에 대해서 최대 허용 지연시간 범위를 보장할 수는 없다. 만약 네트워크 상에 매우 많은 개체가 연결되면 우선순위가 낮은 개체들은 전송 기회를 계속적으로 잃어버릴 수도 있다. 즉, 우선 순위가 높은 개체가 지속적으로 전송을 하면 우선 순위가 낮은 개체는 상대적으로 급하지는 않지만 중요한 메시지를 오랫동안 전송하지 못하여 결국 위험한 상황을 야기할 수도 있다.

따라서 전송되는 ID의 우선순위를 절대적으로 할당해 두면서도 부하의 상황에 따라 우선순위가 낮은 개체를 배려하는 상대적인 우선순위 값을 사용하는 기법이 필요할 것이다. 한 예로 서로 다른 개체들 사이에서 round-robin 방법은 절대적인 우선 순위를 가진 개체 순서를 통신의 기회를 주되 낮은 순위의 개체도 일정 시간 내의 전송은 보장해줄 수 있는 것이다.

본 논문에서는 LLC의 프레임 수용 필터 링(frame acceptance filtering)을 변형하여 낮은 우선 순위의 개체들의 전송을 보장해 줄 수 있는 방법을 제시한다. 이를 위하여, 메시지의 우선 순위에 따라 지정되어 있는 ID를 수정하여 재정의 하여 하나의 대기 열에 넣어 줌으로 하여 결과적으로 네트워크 상의 메시지 전송을 원하는 개체들을 round-robin 방식으로 서비스하게 한다. 이 때 메시지 전송의 우선순위를 결정하기 위하여, 표준 CAN에서 사용하는 ID 및 DPQ 모드 ID를 함께 사용한다. MAC은 전혀 변화가 없으며 기본적인 CAN 프로토콜과도 아무런 충돌 없이 호환이 가능하다.

1. DPQ (Distributed Precedence Queue) 원리

본 논문에서 제시되는 CAN을 기반으로 하는 공정한 제어 기법의 기본 원리는 공유된 네트워크 상에서 전송하기를 원

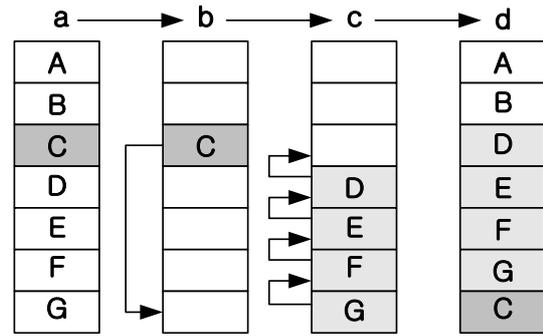


그림 5. DPQ 기법에서의 대기 열 생성.
Fig. 5. Generation of a precedence queue in DPQ mechanism.

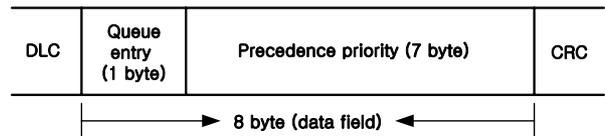


그림 6. DPQ를 위한 데이터 영역의 구성.
Fig. 6. Structure of a data field for DPQ.

하는 모든 개체나 비슷한 목적을 가진 각각의 개체를 하나의 대기 열에 입력시켜 순차적으로 전송되도록 하는 것이다. 이러한 분산 대기 열 protocol은 네트워크 상의 모든 개체들에게 대기 열을 부여하여 동작된다. 그리고 두 개의 우선순위를 할당하는데 각각 서로 독립적으로 동작하며 본 연구에서는 두 개의 대기 열을 사용하는 것으로 한다. 각각의 새로운 대기 열 ID는 각각의 개체에 현재 자기 자신의 대기 열 순서를 지시하게 된다. 하나의 개체들은 자신의 대기 열을 하나씩 증가시켜 가장 높은 대기 열에 도착하면 전송을 하게 된다. 이러한 round-robin 방식을 사용하여 개체 접근 조정자 역할을 수행 할 수 있다.

그림 5와 같이 A에서 G까지 레벨을 가지고 있는 네트워크를 생각해 보자. 그림 5에서 보듯이 전송이 지속적으로 지연되는 하나의 개체, C를 기준으로 그 개체에 함께 전송되어야 하는 개체들을 묶어서 대기 열을 생성한다. 따라서, 다수 개의 대기 열이 분산적으로 생성될 수 있다.

만약 개체 C가 대기 열을 형성시키고자 하면 데이터 프레임에 대기 열에 들어갈 개체들의 ID를 그림 6의 하위 7 바이트에 의하여 7개까지 지정하여 전송해 줄 수 있다. 이 때 상위 바이트부터 높은 선행순위(precedence priority)를 가지게 된다. 그러면 각각의 개체들은 자기가 대기 열에 들어가는 지를 filtering하고 자기 자신의 대기 열을 할당 받게 되고 개체 C는 실제 전송하기를 원하는 메시지를 전송하고 난 후 자기 자신의 대기 열의 마지막으로 가게 된다. 나머지 개체들은 하나씩 대기 열에서 위로 올라간다.

이러한 분산대기 열 프로토콜은 네트워크상의 모든 개체들에게 대기 열을 생성할 수 있는 기회를 제공하고 있다. 그리고 다수 개의 대기 열이 존재하는 경우에는 대기 열 각각에 우선순위를 할당하여 각각 서로 독립적으로 동작할 수 있도록 한다.

분산된 선행대기 열 기법은 11 비트(bit)의 Identifier field를 사용하여 선행순위를 설정하며 상세한 비트지정은 그림 7에



그림 7. CAN 확장 프레임의 헤더 부분의 구성형태.
Fig. 7. Format of the header of extended CAN frames.

서 보여진다. 즉, 표준 CAN에서 사용되는 ID는 18 비트의 EID로 옮겨지고 11 비트 Identifier field는 분산된 선행순위 대기 열 기법에서 새롭게 동적으로 지정하는 선행순위 값을 저장하게 된다.

그림 7에 11 비트 표준 ID 부분에 저장된 DPQ 모드 ID는 각각의 개체의 현재 자기 자신의 대기 열 순서를 가리키게 된다. 하나의 개체가 전송을 완료할 때 마다 자기 자신은 대기 열의 마지막으로 들어가고 다른 개체들은 하나씩 위로 이동하여 대기 열의 맨 위에 도달하면 전송된다. 이러한 round-robin 방식을 사용하여 CSMA/CD에서 행하는 메시지간의 충돌을 조종하는 역할을 수행하게 된다.

이러한 대기 열은 물리적으로 어떤 특정한 위치에 할당되지 아니하고, 네트워크 상에 전반적으로 분산되어서 할당된다. 그리고 각각의 개체는 그들 자신의 대기 열을 저장하고 update를 해야 하는 의무를 가지고 있다.

즉, 자신의 허용 최대 지연시간이 다가오면 선행 대기 열을 생성하여 관련된 개체들과 더불어 우선적 전송이 이루어질 수 있도록 동적으로 우선 순위를 바꾸어 주어야 하고 위급한 상황이 끝나면 대기 열을 해제하여야 한다. 그리고, 전송을 해야 하는 나머지 개체들을 모두 전송 하고 난 후 대기 열을 해제할 것 인지 유지할 것인지는 그림 6의 상위 1 바이트를 사용하여 지정해 준다.

여기서 우선순위와 선행순위와의 차이점을 주목해야 한다. 먼저 네트워크상에 메시지를 전송할 때 우선순위는 각 전송 개체(메시지)에 할당되고 대기 열이 생성되는 경우 대기 열 내에서의 전송순위를 선행순위로 지정한다. 즉, 우선 순위는 메시지의 중요도 및 긴급성에 따라 고정적으로 할당하여 사용하는 것이며 선행순위는 대기 열 내에서의 전송순서를 나타낸다. 두 순위는 모두 충돌을 해결하기 위해서 사용되며 우선순위는 정적인 값인 반면에 선행순위는 사용자가 확인할 수 없으며 아주 동적인 값을 가진다.

2. DPQ 실현 방법

메시지의 우선순위와 선행순위는 각각의 전송 프레임에서 18비트 ID 및 11비트 ID의 위치에 할당 받게 되고 충돌이 발생했을 때 효과적으로 해결을 할 수 있는 기법이다.

분산된 선행대기 열 기법은 기본적인 CAN 프레임의 틀을 변화시키지 않고 실현할 수 있다. 선행대기 열을 지정하기 위해서 Identifier field를 사용한다. 그러나 표준 CAN의 Identifier field의 길이가 너무 짧기 때문에 확장 CAN의 Identifier field를 사용하여 분산된 선행순위 대기 열 기법을 실현한다.

그림 8에서 처음의 두 개의 비트 (t0, t1)는 논리적으로 0의 값으로 설정하여 표준 CAN 통신과 분산 선행대기 열 기법에 의한 프로토콜을 구분한다. 따라서, 분산 선행대기 열 기법은 기본적인 CAN 기법보다 항상 높은 우선순위를 가지게 되고 서로 공존할 수가 있다. P는 두 개의 선행순위 대기 열

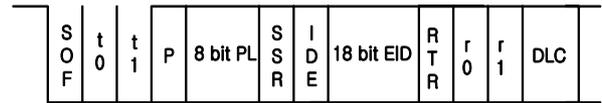


그림 8. DPQ 프레임의 헤더 부분 구성형태.
Fig. 8. Format of the header of DPQ frames.

을 사용할 때 높은 우선순위를 가지는 대기 열에 0의 값을 낮은 우선순위를 가지는 대기 열에 1의 값으로 할당한다. 실제적으로 t1과 P를 활용하면 최대 대기 열을 4개까지 확장하여 우선 순위를 할당할 수 있다. 나머지 8 비트는 선행순위를 나타낸다. Queued mode를 사용해서 전송되는 프레임의 선행순위는 각각의 개체가 프레임을 필터링 해서 자기의 전송 대기 순서를 할당하는 중요한 값이 된다. 그 값은 분산된 대기 열에서 가장 첫 번째 전송대기 순서는 1의 값을 가지게 되고 가장 나중의 대기 열 값은 254가 된다. 그리고 0의 값은 over-queued mode를 사용할 때 사용되고 255는 under-mode를 사용할 때 전송된다. 여기서 over-queued는 항상 다른 mode보다 높은 선행순서를 가지고 있으면 under-mode는 다른 mode들 보다 항상 낮은 선행순위를 자기게 된다. 즉, 이 8 비트가 전송대기 순서를 나타내는 것이다. 본 실험에서 사용한 분산된 선행순위 대기 열 기법은 t0, t1을 사용해서 표준 CAN 기법과 구별하였으며 P를 사용해서 각각의 대기 열의 우선순위를 설정하며 8비트 PL를 사용하여 분산된 대기 열 내의 선행순위를 결정하였다.

IV. 시스템 구성 및 실험

선박에서 엔진 가동의 효율적인 관리 및 엔진 점검은 중요하다. 엔진이 지속적으로 동작을 하면 베어링(crosshead bearing, crank bearing, main bearing)이 마모가 되며 엔진 내부의 실린더 배출 가스 온도, 냉각 청수(cooling fresh water) 배출구 온도, Main bearing 온도가 변화하게 된다. 그리고 Main bearing의 회전 운동을 원활하게 하는 피스톤 냉각오일의 흐름이 변화하게 된다. 베어링의 마모와 엔진내부의 각 부분 온도 변

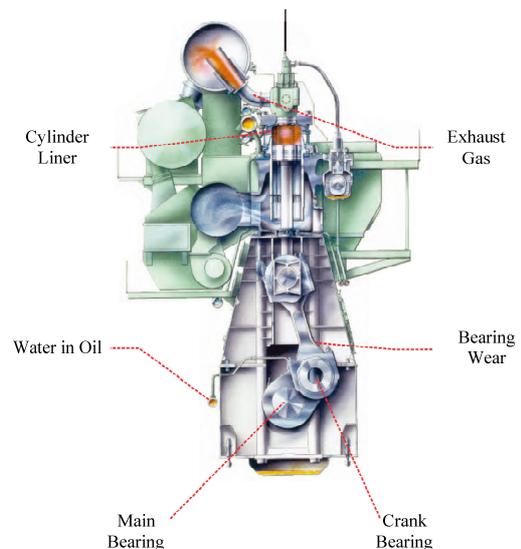


그림 9. 선박 엔진.
Fig. 9. Marine engine.

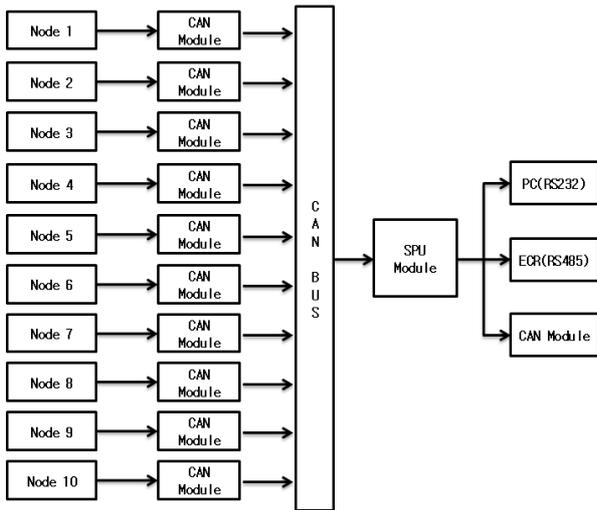


그림 10. ESMS 구조.
Fig. 10. ESMS architecture.

화에 따라서 엔진의 중단 및 가동여부를 판단하게 된다. 특히 베어링의 마모 정도를 점검하기 위해서는 엔진 동작을 중단 되어야 하며, 엔지니어가 엔진을 분해하여 점검을 해야 하므로 많은 비용과 시간이 소요되며 점검시 직접 노출이 되어 안전사고와 직접적으로 연관이 된다.

본 논문에서 제시된 기법의 효용성을 검증하기 위해 실제 선박 엔진 상태 시스템에 사용되고 있는 센서들을 사용함으로써 CAN 통신 모듈 10개와 SPU 모듈 1개로 총 11개의 개체로 시스템을 구성하여 실험 하였다.

그림 10은 본 논문에서 제안하는 ESMS (Engine State Monitoring System) 선박엔진 상태 모니터링 시스템의 구조이다. 엔진 내부의 상태 및 베어링 마모 정도를 감지하는 시스템의 기본원리는 거대한 Two-Stroke 디젤 엔진의 각 실린더마다 crosshead header와 엔진 내부의 실린더 배출 가스 온도, 냉각 청수(Cooling Fresh Water) 배출구 온도, 메인 베어링 온도 및 선박 엔진용 피스톤 냉각오일 온도 등의 상태를 감지하는 온도센서를 CAN 모듈을 통해 온도센서의 아날로그 값을 디지털로 변환하고 CAN 통신을 통해 SPU (Signal Processing Unit) Module로 전송 한다. 전송된 데이터를 통하여 엔진의 온도 값을 전송하고 RS232통신으로 PC로 전송하며, 선박에서 사용하는 RS485통신으로 ECR (Engine Control Room)의 Indication unit 등으로 전송하여 실시간 모니터링 할 수 있도록 제안하였다.

그림 11은 본 논문에서 실제 실험에 사용한 온도센서이다. 온도센서가 측정 할 수 있는 온도의 범위는 0℃~600℃까지이다. 본 논문에서는 가장 일반적으로 산업현장에서 사용되고 있는 Pt100 모델을 사용하였으며, Pt100은 0℃에서 100 ohm의 저항 값을 가지고 100℃에서 138.5 ohm의 저항 값을 가진다.

그림 12는 CAN 모듈 하드웨어 구조이다. 각각의 개체는 CAN 모듈이 내장된 dsPIC30F4012를 사용하였으며 CAN Interface Chip으로는 MCP2551을 사용하였다.

그림 13은 SPU (Signal Processing Unit) 모듈 하드웨어 구조이다. CAN BUS를 통하여 SPU 모듈로 전송되는 데이터를 통



그림 11. 온도센서.
Fig. 11. Resistance bulb.

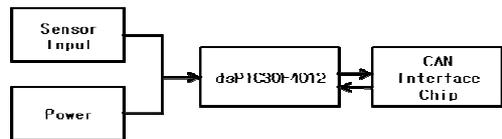


그림 12. CAN 모듈 하드웨어 구조.
Fig. 12. CAN Module Hardware structure.



그림 13. SPU 모듈 하드웨어 구조.
Fig. 13. SPU Module Hardware structure.

합적으로 처리 하여 각각의 센서의 동작 여부를 확인 할 수 있도록 설계하였으며 SPU 모듈은 CAN Core가 내장된 TI사의 TMS320F28335를 사용하였으며 CAN Interface Chip으로는 SN65HVD230을 사용하였다.

각각의 Node의 전송주기는 모두 1ms로 설정하여 충돌이 자주 발생하도록 하였다. Node 1의 전송 message가 가장 높은 Priority로 주어지며 점차적으로 Priority가 낮아진다. 통신속도는 1Mbps로 설정하였다.

엔진 상태를 체크 하기 위한 온도센서들의 데이터를 통합적으로 관리하며 SPU에서 실시간으로 처리된 데이터는 RS232통신을 통해 PC에서 확인 할 수 있도록 하였으며, RS485통신을 통해 ECR (Engine Control Room)에 Indicator에 표시하도록 하였다.

뿐만 아니라 엔진의 각 중요한 부분에 장착되는 온도센서들을 CAN 통신을 통하여 통합적으로 전송하여 전기배선을 줄여 비용 절감과 통합적인 데이터 처리를 통해 각 센서에서 전송된 데이터 에러 여부 확인 및 센서의 고장 여부를 판단 할 수 있어 신뢰성 확보와 수리 경비절감을 할 수 있도록 제안하였다.

V. 실험 및 결과

본 논문에서 실질적인 ESMS (Engine state Monitoring system) 의 전체 실험 시스템 구성은 그림 14와 같다.

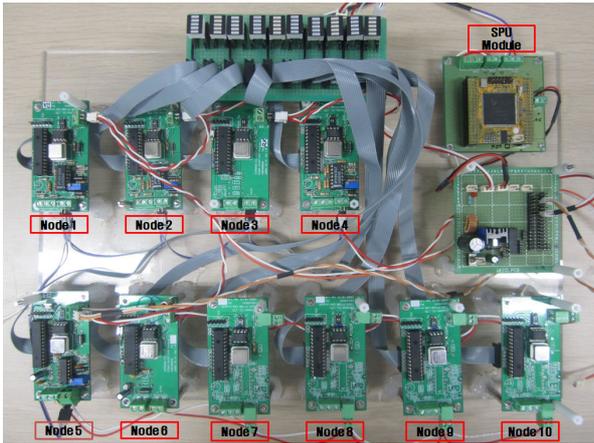


그림 14. 전체 시스템 구성도.
Fig. 14. Total system organization.

그림 14와 같이 오른쪽 상위에 SPU (Signal Processing Unit) 모듈이며 아래는 Power와 CAN 커넥터이고, 나머지는 CAN 모듈이다. 온도센서(PT100)에서 출력되는 아날로그 데이터들은 각각의 CAN 모듈로 연결하여 데이터를 디지털로 변환하여 CAN 통신으로 SPU (Signal Processing Unit) 모듈로 전송하도록 하였다. 여기서 CAN 모듈에서 나오는 신호를 통합 처리하는 SPU를 Node0로 하였고, 고온과 고압이 발생하는 Exhaust gas outlet에 장착 되는 온도센서와 CAN 모듈을 priority 1, 2로 설정하여 전송하고자 하는 데이터를 최우선적으로 전송할 수 있도록 하였고, 냉각 청수(cooling fresh water) 배출구 온도를 측정하는 온도센서와 CAN 모듈을 priority 3, 4로 설정하며, Crosshead bearing의 온도를 측정하는 온도센서와 CAN 모듈을 priority 5, 6으로 설정하며 Crosshead bearing과 Main bearing을 연결하는 Crankpin bearing의 온도를 측정하는 온도센서와 CAN 모듈을 priority 7, 8로 설정하고 Main bearing의 온도를 측정하는 온도센서와 CAN 모듈을 Priority 9, 10으로 설정하였다.

선박에서 엔진 가동의 효율적인 관리 및 엔진 점검이 아주 중요하다. 엔진이 지속적으로 동작을 하면 베어링(crosshead bearing crankpin bearing, main bearing)이 마모가 되며 엔진 내부의 실린더 배출 가스 온도, 냉각 청수(cooling fresh water) 배출구 온도를 측정하여야 한다. 선박 엔진 내부의 각 부분 온도 변화에 따라서 엔진의 중단 및 가동여부를 판단해야 한다. 그럼으로 본 논문에서는 실질적으로 선박 엔진 내부에 사용되는 온도센서를 이용하여 각 요소들의 상태를 실시간으로 감시 하고 모니터링 할 수 있는 시스템을 제시하고, DPQ 알고리즘을 통하여 실시간으로 선박 엔진의 상태를 모니터링 할 수 있도록 제안하였다.

표 1은 Extended CAN Message와 DPQ Mechanism에 대한 ID를 설정한 것이며 실험에 대한 ID는 총 29 비트로 구성된다.

그림 15는 CAN mode(2.0B)로 적용하여 실험한 결과이다. Node 전송 주기는 1ms로 Node간의 충돌이 빈번하게 일어나도록 설정하였다. 그림에서 알 수 있듯이 높은 priority를 가지는 Node의 데이터는 수신이 잘되는 반면(5개의 Node만 수신) 상대적으로 낮은 priority를 가지는 Node의 데이터는 수신되지 않는다는 것을 알 수 있다.

표 1. 실험에 대한 ID 정의.

Table 1. Identification (ID) Definition in experience.

	Extended CAN(Identifier ID, 29bit)										
	SID					EID					
	DPQ	PL				Reserved			Node ID(0x00~0xFF)		
	3	8				11			8		
node 1	1 1 0	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 0 0 1						
node 2	1 1 0	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 0 0 1 0						
node 3	1 1 0	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 0 0 1 1						
node 4	1 1 0	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 0 1 0 0						
node 5	1 1 0	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 0 1 0 1						
node 6	1 1 0	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 0 1 1 0						
node 7	1 1 0	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 0 1 1 1						
node 8	1 1 0	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 1 0 0 0						
node 9	1 1 0	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 1 0 0 1						
node 10	1 1 0	0 0 0 0 0 0 0 0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 1 0 1 0						
DPQ											
node 1	1 0 0	0 0 0 0 0 0 0 0	1	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 0 0 0 1					
node 2	1 0 0	0 0 0 0 0 0 0 1	0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 0 0 1 0					
node 3	1 0 0	0 0 0 0 0 0 1 1	0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 0 1 1					
node 4	1 0 0	0 0 0 0 0 1 0 0	0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 1 0 0					
node 5	1 0 0	0 0 0 0 0 1 0 1	0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 1 0 1					
node 6	1 0 0	0 0 0 0 0 1 1 0	0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 1 1 0					
node 7	1 0 0	0 0 0 0 0 1 1 1	0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 0 1 1 1					
node 8	1 0 0	0 0 0 0 1 0 0 0	0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 1 0 0 0					
node 9	1 0 0	0 0 0 0 1 0 0 1	0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 1 0 0 1					
node 10	1 0 0	0 0 0 0 1 0 1 0	0	0 0 0	0 0 0 0 1 1 1	0 0 0 0 1 0 1 0					

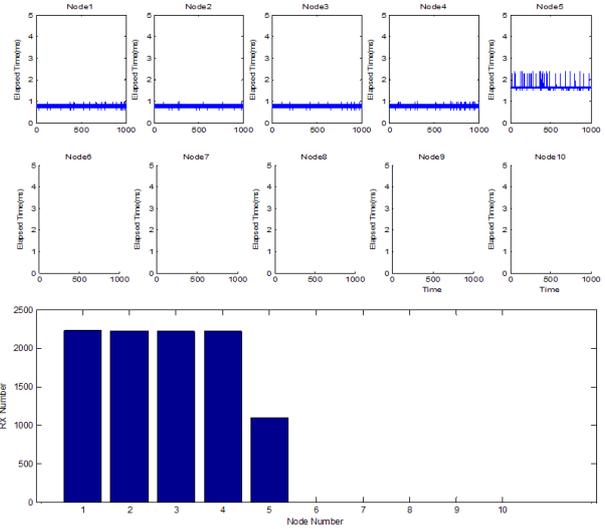


그림 15. 실험의 평균 CAN 전송지연시간.
Fig. 15. Average CAN transmission delay time in experiment.

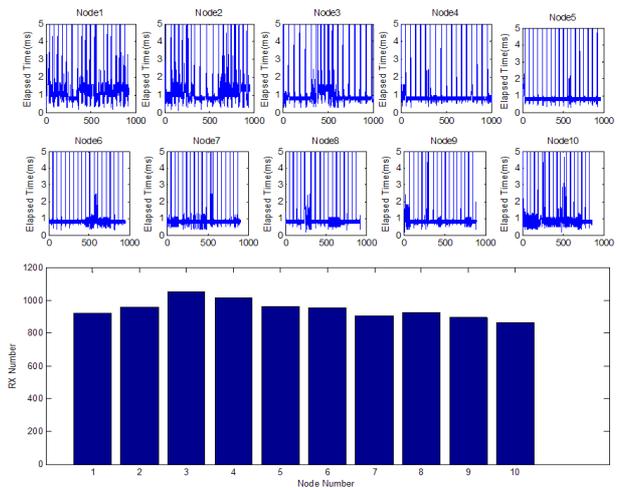


그림 16. 실험의 DPQ mode시 평균 전송지연시간.
Fig. 16. Average mechanism transmission delay time of DPQ mode in experiment.

그림 16은 DPQ Mechanism을 이용한 실험 결과이다. 높은 priority를 가지는 Node와 낮은 priority를 가지는 Node의 데이터가 끌고루 수신된다는 것을 확인 할 수 있다.

VI. 결론

본 논문에서는 실시간 및 분산처리에 적합한 표준 CAN 프로토콜을 이용하여 네트워크 상에서 일어나는 충돌을 조정하기 위한 Priority의 비효율성을 개선하기 위해 DPQ기법을 적용함으로써 선박 엔진 상태 모니터링 시스템을 구현하였다. DPQ 기법을 통해 Priority가 낮은 Node의 전송기회가 보장이 되었으며 그에 따른 지연 시간도 향상되는 것을 실험으로 확인하였으며 데이터를 수신할 수 있도록 하였다.

ESMS (Engine State Monitoring System) 선박 엔진 상태 모니터링 시스템의 특징은 다음과 같다.

1. CAN을 이용하여 실제 선박엔진에 사용되는 온도센서를 이용하여 CAN 모듈을 구성하였으며, ID 부여에 의한 CAN 모듈을 구별하였고, 모듈을 추가 하기에 용이하도록 설계하였다.
2. CAN 모듈에서 전송되는 데이터를 통합처리 하기 위해 SPU 모듈을 구성함으로써 Exhaust gas outlet, 냉각 청수 (cooling fresh water) 배출구 온도, Crosshead bearing, Crankpin bearing, Main bearing 상태를 확인할 수 있도록 하였다.
3. 기존 선박엔진 시스템과 호환을 위한 RS232, RS485 통신 기능을 제공하여 선박의 ER (Engine Room), ECR (Engine Control Room)에서 선박의 상태를 확인 할 수 있도록 하였다.
4. 전송주기를 1ms로 함으로 Node간의 충돌이 빈번하게 일어난다도록 설정하였으며 DPQ기법을 이용함으로써 높은 priority를 가지는 Node와 낮은 priority를 가지는 Node의 데이터가 끌고루 수신이 되고, 낮은 priority를 가지는 Node의 전송기회가 보장이 되며 전송지연 시간이 짧아지는 현상을 실험을 통해 확인 할 수 있었다.

그러나 우선순위가 높은 개체에 DPQ 기법을 적용했을 경우 기본적인 CAN을 적용했을 때의 전송시간 보다 높은 priority를 가지는 Node의 데이터 전송시간 보다 효율이 떨어지는 것을 확인 할 수가 있었다. 향후 과제는 모든 개체에 걸쳐 DPQ 기법을 적용 할 때 각각의 Node의 데이터 전송시간을 보장 할 수 있는지에 대한 연구가 필요하다.

참고문헌

- [1] M. H. Kim, J. G. Lee, S. Lee, and K. C. Lee, "A study on distributed message allocation method of CAN system with dual communication channels," *Journal of Institute of Control, Robotics and System*, pp. 1018-1023, Oct. 2010.
- [2] J. W. Jung and D. S. Kim, "Real-time synchronization algorithm for industrial hybrid networks: CAN and sensor network," *Journal of Institute of Control, Robotics and System*, vol. 16, no. 2, Feb. 2010.
- [3] <http://www.doosanengine.com/>
- [4] J. G. Lee, M. H. Kim, J. H. Park, S. Lee, and K. C. Lee, "Implementation of IEEE 1451 based dual CAN module for fault tolerance of in-vehicle networking system," *Journal of*

Institute of Control, Robotics and System, vol. 15, no. 7, Jul. 2010.

- [5] H. Cao, J. Ma, G. Zhang, J. Zhang, and G. RenMarine, "Marine main engine remote control system with redundancy CAN Bus based on distributed processing technology," *International Conference on Intelligent Control and Information Processing*, Aug. 2010.
- [6] Bosch, "CAN Specification Version 2.0.," Robert Bosch GmbH. Stuttgart, 1991.
- [7] International Standard Organization, "Road - vehicles Interchange of digital information - Controller area network for high-speed communication," ISO 11898, Nov. 1993.
- [8] G. Cena and A. Valenzano, "An improved CAN fieldbus for industrial application," *IEEE Trans. on Industrial Electronics*, vol. 44, no. 4, Aug. 1997.
- [9] H. Hilmer, H. D. Kochs, and E. Dittmar, "A fault-tolerant communication architecture for real-time control system," *IEEE International Workshop on Factory Communication System*, pp. 111-118, 1997.
- [10] S. Corriganm, *Intriduction to the Controller Area Network*, TI, Application Report. Texas, 2002.
- [11] M. Farsi, K. Ratcliff, and M. Barbosa, "An overview of controller area network," *Computing and Control Engineering Journal*, vol. 10, no. 3, pp. 113-120, 1999.
- [12] C. Guerreo, "Hardware support for fault tolerance in triple redundant CAN controllers," *Electronics, Circuits and System, 9th International Conference*, vol. 2, pp. 457-460, 2002.
- [13] C. Guerreo, "Design and implementation of a redundancy manager for triple redundant CAN controllers," *IEEE 28th Annual Conference of the Industrial Electronics Society*, vol. 3, pp. 2294-2299, Nov. 2002.
- [14] R. I. Davis and A. Burns, "Robust priority assignment for message on CAN (Controller Area Network)," 26 November 2008. Springer Science+Business Media, LLC 2008.
- [15] E. Tovar, F. Pacheco, and N. Cruz, "A scalable and efficient approach for obtaining measurements in CAN-based control system," *IEEE Transactions on Industrial Information*, vol. 4, no. 2, May 2008.
- [16] J. M. Giron-Sierra, C. Insaurralde, M. Seminario, J. F. Jimenez, and P. Klose, "CANbus-based distributed fuel system with smart components," *IEEE Transactions on Aerospace and Electronic System*, vol. 44, no. 3 July 2008.
- [17] Jadsonlee Da Silva Sa, Jaidilson Jo Da Silva, Miguel Goncalves Wanzeller, and Jose Sergio Da Rocha Neto, "Monitoring of temperature using smart sensors based on CAN architecture," *Proc. of the 15th International Conference on Electronics*, 2005.
- [18] <http://www.eskorea.net/>



이 현

2007년 동아대학교 전기전자컴퓨터공학부 전자공학과(공학사). 2010년 부산대학교 대학원 전자전기공학과(공학석사). 2010년~현재 부산대학교 전자전기공학과 박사과정. 관심분야는 지능제어시스템, CAN Network, 마이크로 프로세서 응용 시스템 설계.



이 준 석

2009년 부경대학교 전자정보통신공학 전공(공학사). 2011년 부산대학교 전자전기공학과(공학석사). 관심분야는 CAN, Fieldbus.



이 장 명

1980년 서울대학교 전자공학과(공학사). 1982년 서울대학교 전자공학과(공학석사). 1990년 USC(공학박사). 1992년~현재 부산대학교 전자전기공학부 교수. 관심 분야는 지능로봇제어, 마이크로 프로세서 응용 시스템 설계, 특수환경 Localization 및 Navigation.