



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

ISO 19847/19848 기반 선박 기관부 데이터 관리 시스템 및 모니터링 어플리케이션 개발

A Development of ISO 19847/19848 based Data Management System and
Monitoring Application for Shipboard Machinery Equipment



지도교수 이 장 세

2019년 02월

한국해양대학교 대학원

컴퓨터공학과

우 윤 태

본 논문을 우윤태의 공학석사 학위논문으로 인준함

위원장 : 박 휴 찬



위원 : 이 서 정



위원 : 이 장 세



2018년 12월 26일

한국해양대학교 대학원

제 5 장 선박 기관부 데이터 관리 시스템과 모니터링 어플리케이션 구현 및 검증 ...	35
5.1 개발 환경	35
5.2 시스템 구현	36
5.3 시스템 검증	43
 제 6 장 결론 및 향후연구	 51
 참고문헌	 53



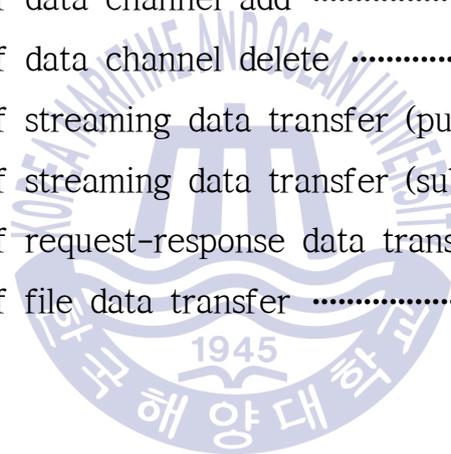
List of Tables

Table 2.1	Main action items in SSAP	7
Table 2.2	MQTT message format	11
Table 2.3	Features of QoS(Quality Of Service)	12
Table 2.4	Features of retain	12
Table 2.5	REST methods	13
Table 3.1	General of requirements in ISO 19847	15
Table 3.2	Performance for shipboard data server	16
Table 3.3	Data I/O and management of requirements in ISO 19847 ...	16
Table 3.4	Categorization of requirements in ISO 19848	18
Table 4.1	Classification of IEC 61162-1 talker identifier mnemonics ..	27
Table 4.2	Column and description of database table	33
Table 5.1	System development environments	35

List of Figures

Fig 2.1 Ship-shore open platform	5
Fig 2.2 Application services	6
Fig 2.3 The concept model of ISO 19847/19848	9
Fig 2.4 MQTT protocol communication process	10
Fig 2.5 Example for MQTT topic structure	12
Fig 3.1 Channel ID structures	19
Fig 3.2 Relationship between data set and time series data	21
Fig 3.3 Logical structure of data model	22
Fig 4.1 Example of equipment identification code (before encoding) ..	23
Fig 4.2 Equipment identification code table	24
Fig 4.3 Example of equipment identification code (after encoding)	24
Fig 4.4 Designed system architecture based on ISO 19847/19848	26
Fig 4.5 IEC 61162-450 message example	27
Fig 4.6 Classification of IEC 61162-1 sentences	28
Fig 4.7 Equipment identification code encoding process	28
Fig 4.8 Class diagrams of IEC 61162-450 message generator	29
Fig 4.9 Sentence generation algorithm for the IEC 61162-1/2 messages	29
Fig 4.10 Class diagrams of data management service module	30
Fig 4.11 Equipment identification code decoding process	31
Fig 4.12 Class diagrams of web service module	32
Fig 4.13 Database table structure	33
Fig 4.14 Class diagrams of monitoring application	34
Fig 5.1 IEC 61162-450 message generator	36

Fig 5.2 Data management service module (main UI)	37
Fig 5.3 Data management service module (channel list UI)	38
Fig 5.4 Data management service module (setting UI)	39
Fig 5.5 Web service module	39
Fig 5.6 Monitoring application (main UI)	40
Fig 5.7 Monitoring application (main UI - MQTT setting)	41
Fig 5.8 Monitoring application (request-response UI)	41
Fig 5.9 Monitoring application (file transfer protocol UI)	42
Fig 5.10 Test scenarios for system verification	43
Fig 5.11 Test result of create data channel list file	44
Fig 5.12 Test result of data channel add	45
Fig 5.13 Test result of data channel delete	46
Fig 5.14 Test result of streaming data transfer (publisher)	47
Fig 5.15 Test result of streaming data transfer (subscriber)	48
Fig 5.16 Test result of request-response data transfer	49
Fig 5.17 Test result of file data transfer	50



ISO 19847/19848 기반 선박 기관부 데이터 관리 시스템 및 모니터링 어플리케이션 개발

우 윤 태

한국해양대학교 대학원
컴퓨터공학과

초록

최근, 선박에 탑재된 장비로부터 수집되는 정보를 분석하여 운항 및 유지보수 지원 목적으로 활용하기 위한 여러 연구가 진행되고 있으나, 인터페이스 방식 및 데이터 프로토콜이 공개되어 있지 않아서 장비의 실시간 작동 상태 확인, 항해장비와의 연관성 분석 등 확장 측면에 여러 제약점을 가진다.

항해통신장비의 인터페이스의 경우 상대적으로 표준화가 잘 이루어져 있으나, 선박 기관부 및 기타 장비부의 인터페이스는 여러 이유로 인해 표준화가 되어있지 않아 데이터 교환 및 관리에 많은 한계가 존재한다.

이러한 문제를 해결하기 위해 2012년 일본에서 SSAP(Smart Ship Application Platform) 프로젝트를 진행하였고, 이에 결과물로 해상 필드 데이터 공유를 위한 선박 데이터 서버에 관한 표준인 ISO 19847과 선박 기계 장비부 데이터 표준인 ISO 19848이 등장하였으며, 2018년 10월에 정식 표준으로 공포되었다.

본 논문에서는 이러한 두 가지 국제표준인 DIS(Draft International Standard) 버전을 기반으로 선박 기관부 데이터 관리 시스템을 개발하고, 유용성 검

증을 위한 모니터링 어플리케이션을 개발하였으며, 이를 기반으로 데이터 교환에 관한 실험을 수행하는 것에 관한 내용을 다룬다.

이를 위해서는 관련 연구 및 ISO 19847/19848의 요구사항을 분석하고, 요구사항을 충족하는 선박 데이터 관리 시스템 아키텍처를 설계하며, 각 구성요소를 개발하여 유용성을 검증한다.

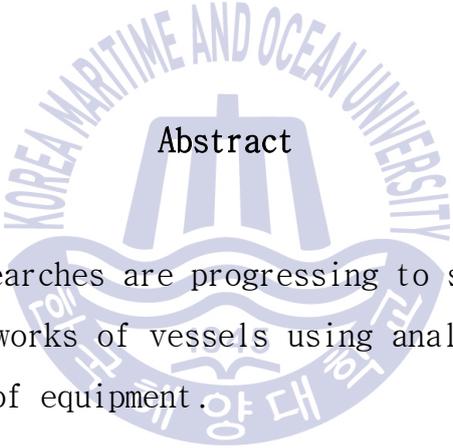
키워드 : ISO 19847, ISO 19848, 선박 기계 장비, 선박 기관부 데이터, 선박 데이터 관리 시스템, 모니터링 어플리케이션



A Development of ISO 19847/19848 based Data Management System and Monitoring Application for Shipboard Machinery Equipment

Woo, Yun Tae

Department of Computer Engineering,
Graduate School of Korea Maritime and Ocean University



Abstract

Recently, many researches are progressing to support the operation and the maintenance works of vessels using analyzed result based on various information of equipment.

However, there are limitations in terms of the expansion such as the realtime check for the operation state of the equipment and the analysis of the navigation equipment because the interface method and the data protocol had not released.

The interfaces of communication equipment are standardized very well, but the interfaces of machinery and other parts are not standardized yet.

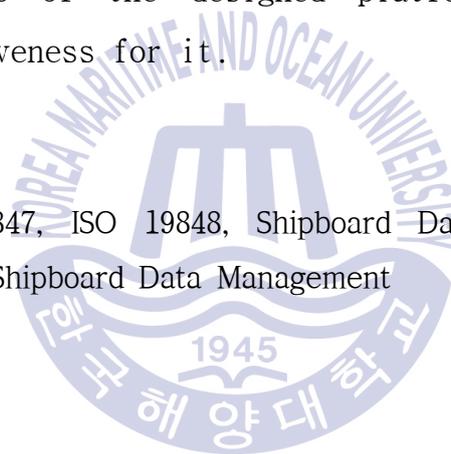
To solve these problem, the SSAP(Smart Ship Application Platform) project was progressed in Japan in 2012, and the ISO 19847 for shipboard data servers to share field data at sea and ISO 19848 for

standard for shipboard machinery and equipment are established as the result for this project.

In this paper, we developed a data management system for shipboard machinery equipment and the monitoring application for the verification of the utility, and we tested the data exchanging using the developed platform based on the standards.

To do this, we analyzed the requirements of the standards and related researches, and designed an architecture of shipboard data management system that satisfied the requirements. Also, we developed components of the designed platform architecture and verified the effectiveness for it.

Key Words: ISO 19847, ISO 19848, Shipboard Data Management System, Monitoring Application, Shipboard Data Management



제 1 장 서 론

운항중인 선박의 탑재 장비 상태의 실시간 모니터링에 관한 연구는 오래전부터 대두되어 왔으며, 최근 일부 해운선사, 항해 장비 제조사 및 주요 항만을 위주로 해상교통관제시스템과 선박 원격 모니터링 시스템을 구축하여 운영하고 있다.

이러한 시스템은 선박 위치정보 확인과 탑재 장비의 관리 목적으로 활용되고 있고, 장비의 상태를 실시간으로 파악하기에는 한계가 존재하며 대부분의 기관부 장비의 경우 선박 원격 모니터링 시스템을 통해 각 제조사에서 공급한 장비에 국한된 관리 기능 및 상태 확인이 가능한 실정이다[1]. 또한, 인터페이스 방식 및 데이터 프로토콜이 공개되어 있지 않아서 선박 관리자 및 선주 등이 요구하는 장비의 실시간 작동 상태 확인, 항해 장비와의 연관성 분석 등 확장 측면에 있어 여러 제약점을 가진다.

항해통신장비의 경우 IEC 61162 시리즈를 기반으로 표준화가 이루어져 있어 상호 데이터 교환을 위한 인터페이스가 가능하지만, 기관부 장비들은 상대적으로 표준화가 잘 이루어져있지 않고, 각각 독립적인 형태로 구성되어 있어 인터페이스 및 통합 관리에 어려움이 존재한다.

이러한 기관 관련 장비의 모니터링, 원격 유지보수 진단 및 고장예측을 위해서는 선박에 탑재되는 장비로부터 획득된 데이터가 확보되어야 하는데 이를 위해서는 선박 내 통합 데이터 관리가 필요하다.

이러한 통합 데이터 관리에 대한 필요성 때문에 유럽, 일본 등은 2012년부터 관련된 프로젝트를 진행하였다. 특히, 일본에서는 SSAP(Smart Ship Application Platform) 프로젝트를 통하여 선박 응용 서비스의 기반이 되는 플랫폼 구축을 진행하였고, 결과물로 “ISO 19847 : 해상 필드 데이터 공유를 위한 데이터 서

버”와 “ISO 19848 : 선박 기계 장비 데이터 표준”에 관한 국제 표준 제정 활동을 진행하였다[2,3]. 2016년 DIS(Draft International Standard)가 공개되었으며, 2018년 10월에 정식 표준으로 공표되었다.

본 논문에서는 국제표준인 ISO 19847/19848 DIS 버전을 분석한다. 분석한 내용을 토대로 선박 기관부 데이터 관리 시스템 아키텍처를 설계한다. 설계한 아키텍처는 IEC 61162 데이터 형식을 이용하여 데이터를 전달하였을 때, 장비를 식별할 수 있는 구체적인 방안이 없어 본 논문에서는 ISO 19848 표준에서 제시한 장비 식별 코드 테이블을 참조하여 데이터의 장비 식별 및 코드화 방안을 제시한다. 또한 설계한 내용을 바탕으로 선박 기관부 데이터 관리 시스템을 구현하였고, 유용성 검증을 위한 모니터링 어플리케이션을 개발하였으며, 모니터링 어플리케이션을 이용하여 기관부 데이터 관리 시스템의 유용성을 검증하였다.

개발한 선박 기관부 데이터 관리 시스템을 통하여 선박 탑재 장비의 데이터를 쉽게 접근할 수 있으므로 데이터에 대한 다양한 분석이 가능하고, 선박 탑재 장비의 효율 향상 및 유지보수 등 여러 목적으로 활용될 것으로 기대된다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 선박 탑재 장비 상태 모니터링 및 데이터 관련 선행 연구 및 관련연구에 대해 설명하고 3장에서는 국제표준인 ISO 19847/19848 요구사항을 분석한다. 또한, 4장에서는 선박 기관부 데이터 관리 시스템을 설계하고, 5장에서는 설계한 내용을 바탕으로 구현 및 검증에 대해 다루며, 6장의 결론 및 향후 연구로 끝을 맺는다.

제 2 장 관련 연구

2.1 선박 탑재 장비 상태 모니터링 및 데이터 관리 관련 선행 연구

선박의 안전하고 효율적인 운항을 지원하고 이상 징후를 파악하는데 있어 선박 내의 각종 장비들의 인터페이스를 통한 상태 모니터링이나 통합 데이터 수집 및 관리의 필요성이 대두되고 있다[4].

과거에는 아날로그 신호 또는 NMEA 0183(IEC 61162-1/2) 인터페이스[5]를 이용하여 직접 연결되는 형태로 구현되었지만 선박 장비들의 상호 인터페이스 증가로 확장 및 통합에는 많은 문제점을 지닌다. 이러한 NMEA 0183 표준의 문제점을 해결하기 위해 2000년 초반에 CAN 통신 기반의 NMEA 2000(IEC 61162-3) 프로토콜[6]이 등장하게 되었으며, 이를 활용하여 선박 장비들의 상태 모니터링 및 통합 데이터 관리에 관한 많은 연구가 진행되어 왔다. 대표적인 연구로는 CAN 통신을 이용한 선박 상태 모니터링 시스템[7], CAN을 기반으로 하는 선박 엔진 상태 모니터링 시스템[8], NMEA 2000 프로토콜을 적용한 선박 모니터링 시스템 설계 및 구현[9], NMEA 2000을 이용한 임베디드 선박 모니터링 시스템의 개발[4], NMEA 2000 기반 USB/이더넷 게이트웨이와 모니터링 시스템 개발[10] 등이 있다. 이들 연구에서는 NMEA 2000 표준을 기반으로 한 선박 상태 모니터링 시스템에 대한 연구 및 개발을 하였으나, NMEA 2000 표준의 경우 중 대형 선박에 적용하기 어려운 제약점이 존재한다.

이에 따라 다수의 장비에서 발생하는 데이터 통합 관리의 목적으로 2011년에 이더넷 기반 인터페이스 표준인 IEC 61162-450이 등장하였고, 이 표준을 이용한 선박 통합 네트워크의 통신 라이브러리 기반 모니터링 어플리케이션의 설계 및 구현[11], IEC 61162-450에 기반한 선박 통합 정보시스템의 설계 및 구현[12]

등 다양한 연구가 진행되었다.

선박 데이터 통합 관리에 관한 연구로는 선박에서 데이터의 통합 관리를 위한 데이터베이스 설계 및 구현[13], 디지털 선박 센서 스트림 데이터의 효율적인 저장 관리[14], 선박 데이터 관리를 위한 Shipdex 기반 CSDB Manager 구현[15] 등에 대한 연구들이 진행되었지만, 대부분의 연구들이 상대적으로 표준화가 잘 이루어져있는 항해통신 장비를 대상으로 하고 있다. 또한, 선박 데이터 모델링 규약인 Shipdex에 관한 연구의 경우, 데이터 교환 시 필요한 정보만을 추출하여 통신비의 지출을 최소화하고, 전송 시간을 상당히 줄일 수 있는 장점이 있으나 구성하고 있는 데이터양이 매우 방대하여 적용하는데 어려움이 있다.

이러한 데이터 통합 및 관리의 문제를 해결하기 위해 해외에서는 다양한 프로젝트를 시도하였으며, 특히 일본에서는 2012년부터 SSAP 프로젝트를 진행하였고, 프로젝트의 결과물로 ISO 19847/19848이 등장하였다.

2.2 SSAP(smart ship application platform) 프로젝트

현재 기상정보 전송, 최적 트림, 성능 모니터링, 엔진 모니터링, 상태 모니터링, 에너지 관리 및 원격 유지보수 등 장비의 데이터를 사용하는 해양 기반 어플리케이션이 보편화되고 있다. 이러한 추세의 반영을 위해 2012년에 일본에서 선박-육상 간 플랫폼 연구에 관한 연구를 진행하였으며, 선박 장비 데이터에 쉽게 접근하여 더 많은 응용 프로그램 서비스 개발을 향상시킴으로 응용 프로그램 서비스를 지원한다는 목표로 진행되었다.

그림 2.1은 SSAP 프로젝트에서 제시한 선박-육상 오픈 플랫폼으로 선박에서는 선박 탑재 장비들의 데이터가 네트워크를 통해 데이터 서버에 저장되고, 육상에서는 데이터 요청을 통해 데이터를 가져올 수 있다는 것을 보여준다.

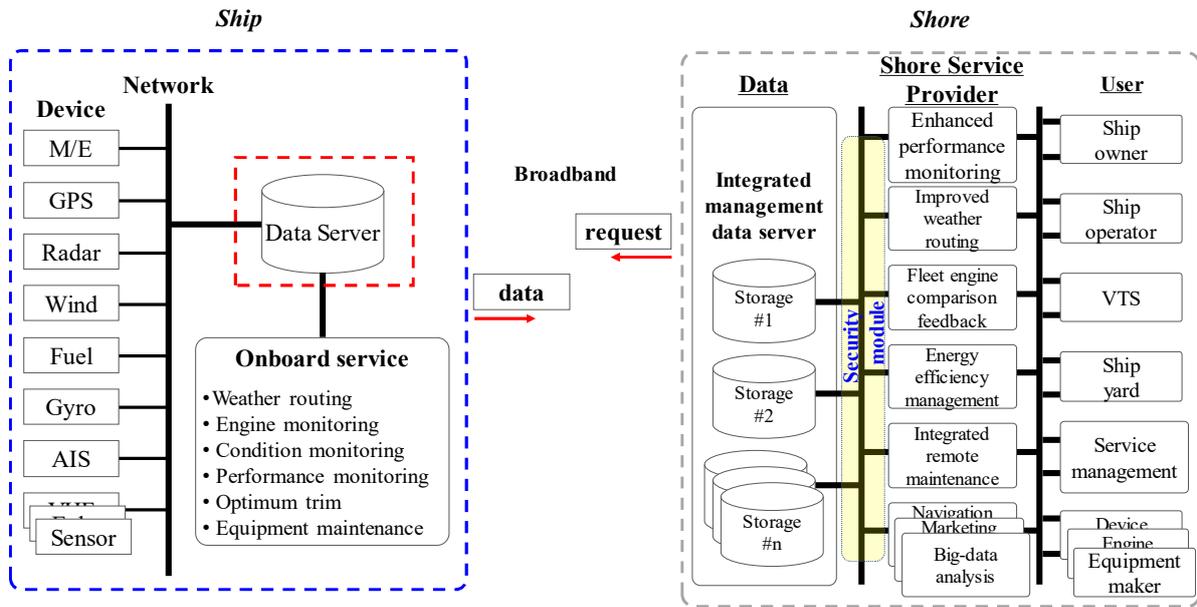
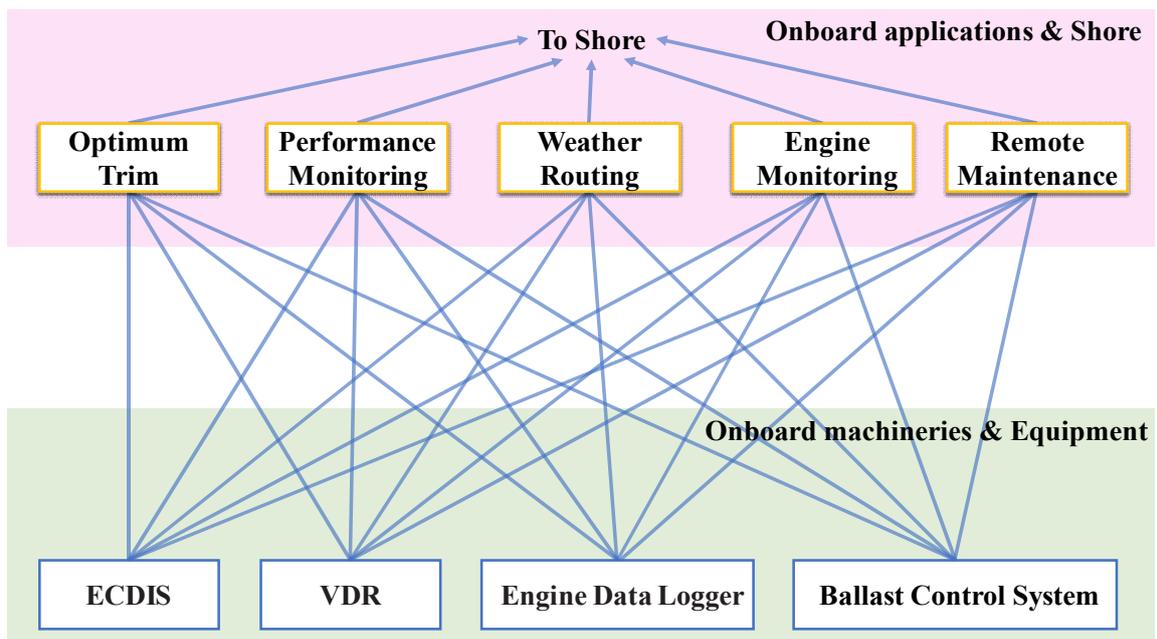


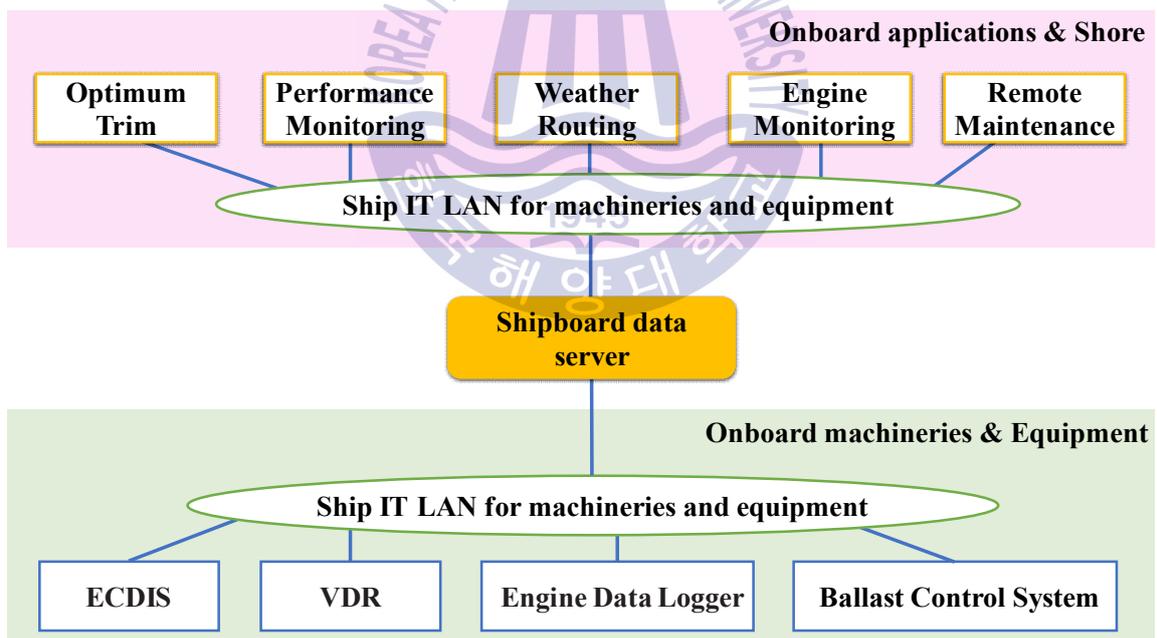
Fig. 2.1 Ship-shore open platform

SSAP 프로젝트의 이점은 첫째, 어플리케이션 제공 업체는 방대한 양의 선박 운영 데이터를 축적한 선박 소유자에게 데이터 분석 서비스를 쉽게 제공할 수 있고, 데이터 수집 플랫폼 공유를 통해 선상 어플리케이션에 대한 선주들의 투자비용을 줄일 수 있다. 둘째, 조선소 및 장비 제조업체는 플랫폼을 통해 데이터에 쉽게 접근할 수 있으며 이를 통해 새로운 서비스를 제공할 수 있다. 셋째, 선박 소유자는 선박과 육상 간의 데이터 전송에 대한 관리 및 제어를 할 수 있다. 마지막으로 표준화된 형식과 프로토콜로 많은 어플리케이션 개발을 향상시킬 수 있다.

그림 2.2는 어플리케이션 서비스 제공에 있어 플랫폼의 이점에 대해 설명한 것으로 그림 2.2 (a)에서는 플랫폼이 없는 경우의 그림으로 어떤 장비에 어플리케이션을 적용할 때 1:1로 연결해야 하는 것을 보여준다. 만약 하나의 장비에만 적용하면 문제가 없지만 하나의 장비를 여러 어플리케이션에 연결할 경우 복잡해진다. 그림 2.2 (b)는 플랫폼을 적용한 결과로 어플리케이션이 장비로부터 데이터를 받을 때 플랫폼을 통해 수신하므로 여러 장비의 데이터에 대한 접근이 용이하다.



(a) Without platform



(b) Adopt platform

Fig. 2.2 Application services

SSAP 프로젝트는 총 2단계로 진행되었고 첫 번째는 2012년 12월에서 2015년 3월까지 진행되었으며 총 36개 기관이 참가하였다. 또한, 주요 성과는 표 2.1과 같이 선상 데이터 서버의 사양설계, 선상 데이터 서버 구현 및 배 2척 시험, 선박-육상 개방형 플랫폼 설계 ISO 19847/19848(NP) 표준의 제안이다. 또한, 두 번째는 2015년 8월에서 2018년까지 진행하였으며, 주요 수행 내용은 SSAP ISO 19847/19848(DIS) 제정활동, SSAP 프로젝트 개념 홍보, 시스템 설계 및 시제품화 등이다.

Table 2.1 Main action items in SSAP

Main Items	Description
Develop specifications of Master Database and its interface	<ul style="list-style-type: none"> • Clarify requirements for Master Database by studying necessary data input of existing application services • Design Master Database and its interface to onboard equipment and applications
Implementation and trial of Master Database	<ul style="list-style-type: none"> • Prototype implementation • Interface between Master DB and onboard equipment • Interface between Master DB and application • Share and onboard trials
Develop specifications for ship-shore information system	<ul style="list-style-type: none"> • Clarify requirements for ship-shore information system • Prototype implementation and shore trial
Standardization	<ul style="list-style-type: none"> • Study association with other international standards • Clarify scope of the ISO proposal • Making WD for ISO proposal

2.3 ISO 19847/19848(DIS) 소개

ISO 19847은 해상 필드 데이터 공유를 위한 선박 데이터 서버에 관한 국제표준으로 선박 탑재 장비에서 발생된 데이터를 저장하고, 데이터를 육상으로 전달하는 데이터 서버에 대한 기능, 성능, 서비스 및 안전 요건에 대해 제시한다. 또한, ISO 19848은 선박 기계류 장비의 데이터 교환에 관한 표준으로 선박에서 데이터를 교환하고 처리하는 것을 용이하도록 하는 목적이며, 탑재된 장비 및 기계를 구분할 수 있는 데이터 규칙 및 식별자를 정의한다.

그림 2.3은 두 표준의 상호 연동 구조를 나타낸다. 그림 2.3 (1)은 ISO 19847 개념 모델로 입력 기능, 출력 기능, 데이터 저장소로 구성되고, 입력 기능은 데이터 스트리밍(Data Streaming)과 데이터 저장(Data Stored)으로 구성된다. 또한 스트리밍 전송 서비스(Streaming Transport Service), 요청-응답 전송 서비스(Request-Response Transport Service), 파일 전송 서비스(File Transfer Service)를 제공한다.

그림 2.3 (2)는 ISO 19848 개념 모델로 입력 데이터는 IEC 61162-1/2 포맷 데이터, ISO 19848에서 정의하는 정형화된 데이터 및 파일을 송신하고, 출력 데이터는 IEC 61162-450 데이터 및 ISO 19848 형식으로 수신하며, 각 표준에서 정의한 형식을 준수하여 데이터를 송수신하도록 정의하고 있다.

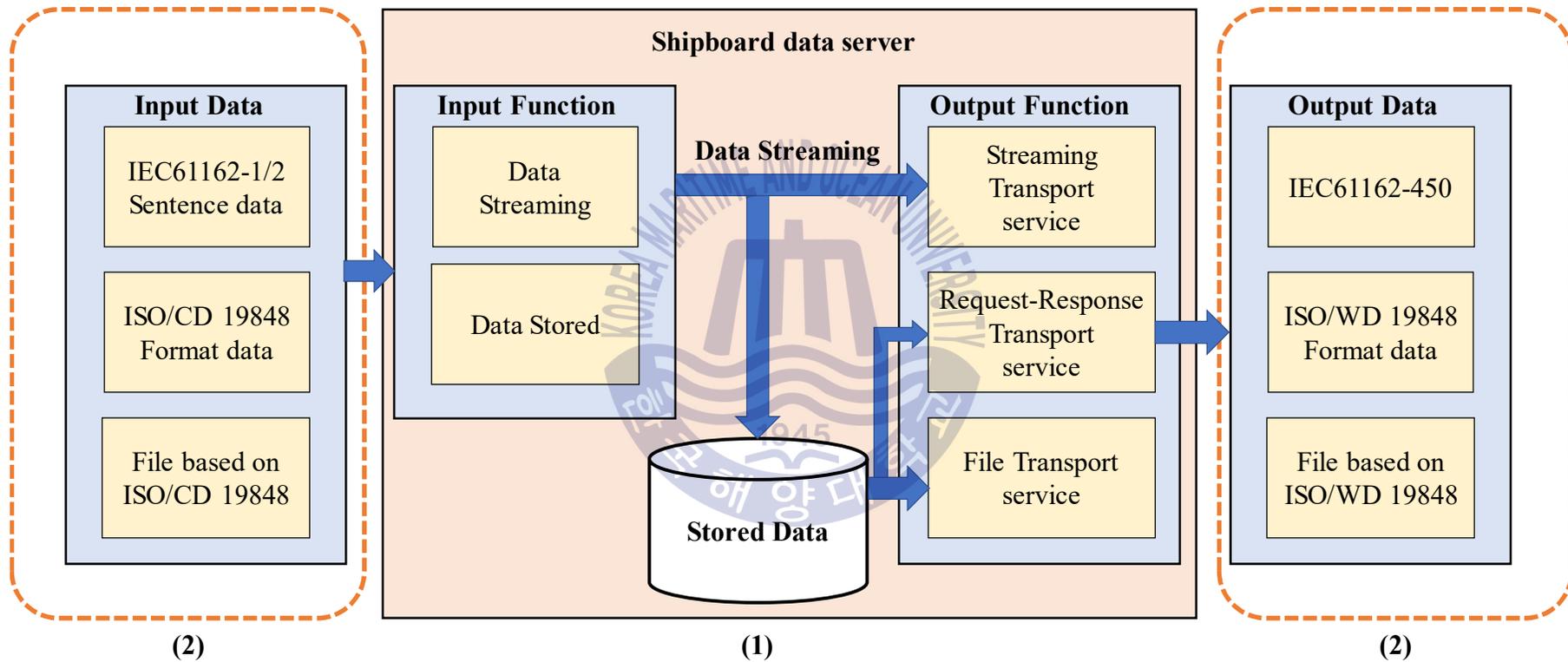


Fig. 2.3 The concept model of ISO 19847/19848

2.4 MQTT(Message Queue Telemetry Transport) 프로토콜

MQTT는 M2M(Machine to Machine), IoT(Internet of Things) 등과 같이 대역폭이 낮은 디바이스들에 적합한 메시징 프로토콜로 품질이 낮고 느린 네트워크에서도 메시지를 안정적으로 전송할 수 있도록 설계되었다. 또한, 개방형 프로토콜로 배포되어, 2013년 OASIS(Organization for the Advancement of Structured Information Standards)에서 IoT를 위한 표준 프로토콜로 발표하였다. 그림 2.4는 MQTT 프로토콜의 통신과정으로 송신자가 메시지 매개자를 통해 특정 메시지를 발행하고 수신자는 메시지를 구독하는 방식을 사용한다.

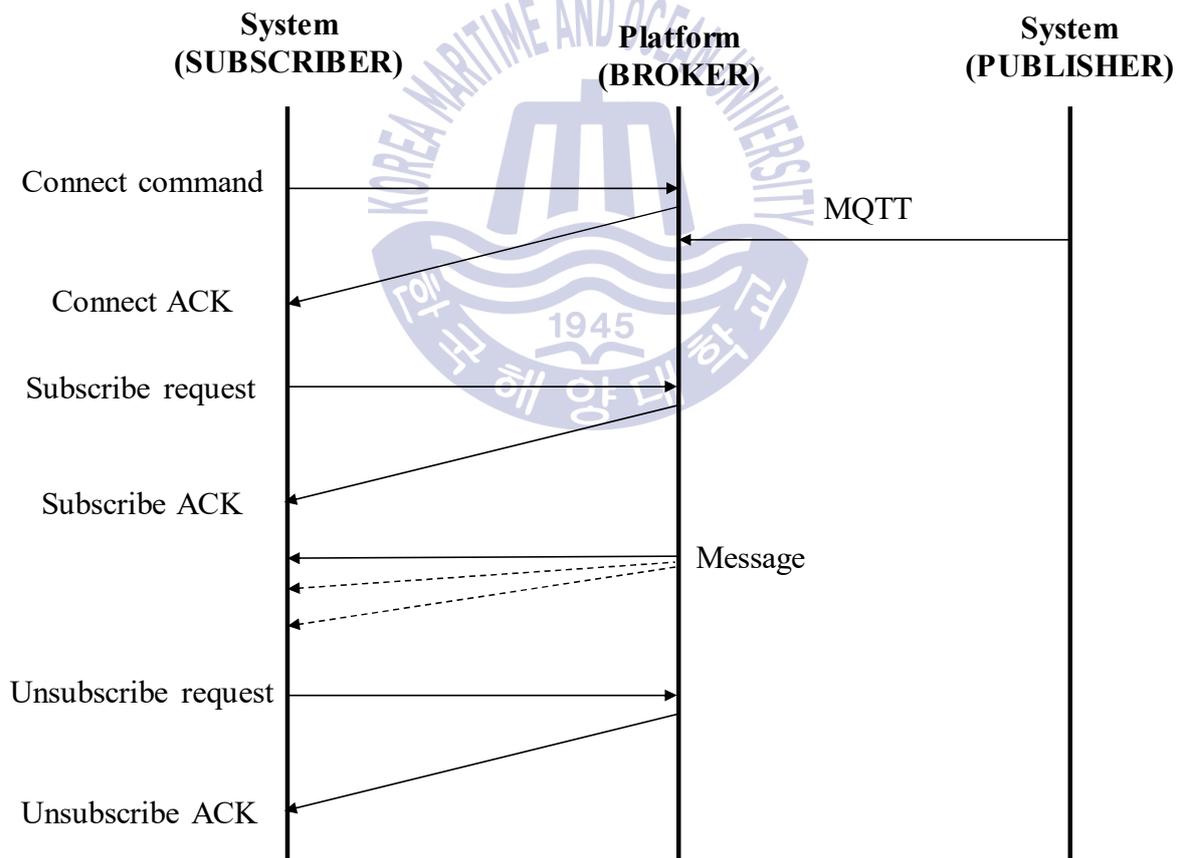


Fig. 2.4 MQTT protocol communication process

MQTT는 단순하고 작은 메시지 구조로 구성되어 있고, 메시지의 최소 크기는 2byte로 메시지 크기에는 제약이 없는 경량화 메시지 프로토콜이다. 표 2.2는 MQTT의 메시지 포맷으로 고정 헤더, 가변 헤더 및 페이로드 세 가지로 구성되고, 고정 헤더에서는 Message Type, DUP(Duplicate delivery) Flag, QoS(Quality of Service) Level, Retain 및 Remaining Length를 정의하고 2개의 바이트로 구성된다.

Table 2.2 MQTT message format

	0	1	2	3	4	5	6	7
Byte 1	Message Type				DUP Flag	QoS Level		Retain
Byte 2	Remaining Length(1~4 bytes)							
Byte 3 ... Byte n	Variable Length Header(Optional)							
Byte n+1 ... Byte m	Variable Length Message Payload(Optional)							

Message Type의 종류는 14가지로 구성되며 4개의 비트를 이용하여 메시지 형식을 정의한다. DUP Flag는 메시지가 중복된 메시지인지, 처음 보낸 메시지인지를 표시한다. QoS는 2개의 비트를 이용하여 표시하고, 표 2.3과 같이 품질 수준에 따라 0~2단계로 나누어지는데, 0단계에서는 하나의 메시지만을 전달하고 전달 여부는 확인하지 않으며, 1단계에서는 하나 이상의 메시지를 전달하나 핸드셰이킹 과정이 없으므로 중복 전송 될 수 있다. 또한, 2단계에서는 하나의 메시지만을 전달하지만 핸드셰이킹 과정이 있기 때문에 품질은 보장되나 메시지 처리 부하가 급격히 늘어난다. Retain의 경우 1개의 비트를 이용하여 표시하고, Retain을 사용하여 데이터 전송 시 메시지 매개자에 메시지가 저장되어 새로운 구독자에게 메시지를 전송해주며, 그 특징을 표 2.4에 정리하였다.

Table 2.3 Features of QoS

QoS	Bit 2	Bit 1	Description
0	0	0	Forward a message only one fire and forget type
1	0	1	Forward a message more than once without handshaking
2	1	0	Forward a message only one with handshaking

Table 2.4 Features of retain

Retain value	Description
False(0)	Don't send the latest message
True(1)	Send the latest message

MQTT의 Publish와 Subscribe는 토픽을 기준으로 동작하고, 토픽은 슬래시(/)로 구분된 계층 구조를 가지며, 토픽들을 파일 시스템 형식으로 나눌 수 있어서 효율적인 관리가 가능하다. 예를 들어 선박 엔진의 온도, 습도 등을 확인한다고 가정할 때, 토픽 구조는 그림 2.5와 같다.

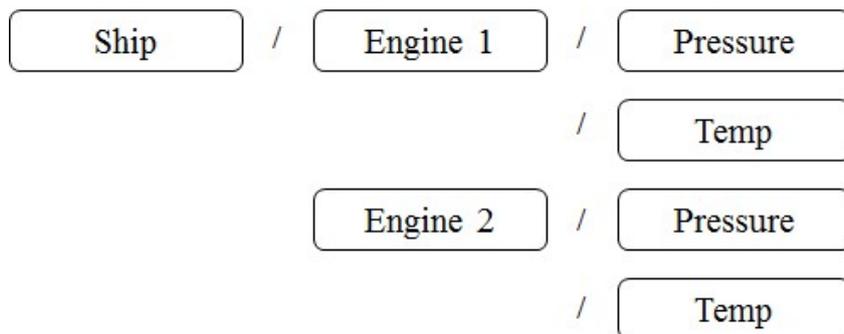


Fig. 2.5 Example for MQTT topic structure

2.5 REST(Representational State Transfer) API

REST는 웹의 창시자 중의 한 사람인 Roy Fielding의 2000년 논문에 의해 처음 소개되었고, 웹의 본래 설계의 우수성을 많이 사용하지 못하고 있다고 판단하여 웹의 장점을 최대한 활용할 수 있는 네트워크 기반의 아키텍처이다. 또한, REST는 자원지향구조로 웹 사이트의 모든 내용을 자원으로 파악하여 각 자원에 고유한 URI(Uniform Resource Identifier)를 부여하고, 해당 자원에 대한 요청을 GET, PUT, POST, DELETE 메서드를 통하여 처리하며, REST 웹 서비스는 자원의 URI를 알고 있으면 클라이언트의 종류에 상관없이 웹 프로토콜만을 이용하여 접근이 가능한 서비스이다.

REST 웹 서비스는 자원 기반의 아키텍처 구성되어 있으며, 자원 중심의 접근 방식, 표현, 전달의 특성을 가지고 있다. 또한 REST 웹 서비스는 동일한 인터페이스, 연결성, 상태 없음 및 주소로 표현 가능함과 같은 특징을 가지고 있으며 각 특징은 다음과 같다.

① 동일한 인터페이스(Homogeneous interface) : HTTP에서 제공하는 기본적인 4가지를 이용하며 REST에서 자원에 대한 동작은 표 2.5와 같이 자원의 조회, 갱신, 추가 및 삭제 기능을 수행한다.

Table 2.5 REST methods

Method	Description
GET	Create resource
PUT	Select resource
POST	Modify resource
DELETE	Delete resource

② 연결성(Connectedness) : 하나의 자원을 주변의 연관된 자원들과 서로 연결하여 표현한다.

③ 상태 없음(Statelessness) : 클라이언트의 상태를 서버에서 관리해서는 안 되며, 모든 요청은 일회성의 성격을 가진다.

④ 주소로 표현 가능함(Addressability) : 모든 자원은 고유 식별자인 URI이 존재하고, 모든 정보를 URI로 표시한다.



제 3 장 ISO 19847/19848 요구사항 분석

본 장에서는 ISO 19847/19848에 기반한 선박 기관부 데이터 교환 및 관리를 위한 아키텍처 설계를 위해 ISO 19847과 ISO 19848의 요구사항에 대해 다룬다.

3.1 ISO 19847 요구사항 분석

ISO 19847은 해상에서 필드 데이터 공유를 위한 선박 데이터 서버에 대해 정의한 국제표준으로 요구사항은 일반적인 요구사항과 데이터 입출력 및 관리 요구사항으로 나뉘어진다.

1) 일반적인 요구사항

일반적인 요구사항은 표 3.1과 같이 크게 세 가지로 구성된다.

첫째, 선상 데이터 서버의 성능 경우, 표 3.2와 같이 선박 데이터 서버의 입출력 및 실시간 데이터 처리 시 필요한 선상 데이터 서버의 성능 요구사항에 대해 정의하고 있고, 기능 요구사항은 저장, 인터페이스, 상태 모니터링, 데이터 백업 및 알람 등을 요구하고 있다.

Table 3.1 General of requirements in ISO 19847

Category	Requirement	Description
General	Performance and function	Processing, Storage, interface, condition monitoring, backup and alert
	Environment	Power supply, vibration-resistant, electromagnetic noise-resistant, temperature/humidity-resistant
	Installation	Environment, network security, etc.

둘째, 환경의 경우, 전원 공급, 진동 감내, 노이즈 방지 등의 물리적 요구사항에 대해 정의하고 있으며, 이는 선박 전자기기 설치에 관한 표준인 IEC 60092를 따르도록 요구하고 있다.

셋째, 설치의 경우에는 장비 제조업체의 매뉴얼에 의해 공간 및 설치 환경 등이 정의되어야 하고, 선상 데이터 서버와 연결되는 네트워크의 보안은 선박 통신 네트워크 설치 가이드라인에 관한 표준인 ISO 16425를 만족하도록 요구하고 있다.

Table 3.2 Performance for shipboard data server

	Performance
Input data	30 data in 1 session / 5 sessions in 1 seconds
Output data	30 data in 1 session / 5 sessions in 5 seconds
Streaming data	1 data in 1 sessions / 1 sessions in 1 seconds

2) 데이터 입력 및 관리 요구사항

데이터 입력 및 관리 요구사항은 표 3.3과 같이 크게 다섯 가지 기능으로 구성된다.

Table 3.3 Data I/O and management of requirements in ISO 19847

Category	Requirement	Description
Data I/O and management	Data management	System clock, data channel list and data source information
	Data I/O	Input, output, request-response, streaming and file
	Alias	Common alias and user-defined alias
	Data calculation	Max, min, average, etc
	Log management	History of access, status changes and management data changes

첫째, 데이터 관리 기능은 크게 세 가지 기능으로 나뉜다.

① 시스템 시간 관리 기능은 데이터에 정확한 타임스탬프를 추가하기 위해 UTC(Universal Time Coordinated)로 동기화되어야하고 시간 편차는 1시간 당 1초 이상 나지 않아야 한다.

② 데이터 채널 리스트 관리 기능은 데이터 입출력 기능에 의해 수정, 삭제 및 등록이 되어야하고 동시에 시간, 날짜 및 바뀐 정보를 기록한다. ③ 데이터 소스 정보 관리 기능은 시간, 날짜 등 바뀐 정보를 기록해야하고, 입력 기능에 의해 수정, 추가 및 삭제가 가능해야한다. 또한, 현재 어떤 데이터 소스가 기록을 위해 현재 할당되었는지 명확하게 나타내야한다.

둘째, 데이터 입출력 기능은 크게 다섯 가지 기능으로 나뉜다.

① 입력 기능은 IEC 61162-1/2 데이터 형식과 ISO 19848에 의해 정의된 파일 형식을 준수하여 데이터를 수신하고 입력 데이터가 타임스탬프를 가지고 있지 않을 경우 각각의 입력 데이터에 시스템 시간의 타임스탬프를 추가하는 기능을 수행한다.

② 출력 기능은 IEC 61162-1/2 및 ISO 19848에 의해 정의된 데이터 혹은 파일 형식을 준수하여 데이터를 전송하는 기능을 제공한다.

③ 요청-응답 데이터 전송 서비스 기능은 시간 순서대로 정렬된 데이터를 처리하는 기능을 제공하고 REST API를 사용하여 요청된 데이터를 교환한다.

④ 스트리밍 데이터 전송 서비스 기능은 하나 이상의 선상 시스템 및 장비에 가장 최근의 데이터를 전송하는 기능을 제공하고 MQTT 프로토콜을 사용한다.

⑤ 파일 전송 서비스 기능은 파일 형식의 데이터 교환 기능을 제공하고, FTP(File Transfer Protocol)를 사용하며 파일 저장 시, PUT 메서드를 사용하여 데이터를 수신한다.

셋째, 별칭 기능은 간략화 된 이름을 사용하여 데이터 접근이 용이하도록 하는 것이 목적이며, 두 가지 형태의 별칭 기능으로 나뉜다.

① 공통 별칭 기능은 공통의 별칭을 정의하며 시스템 및 선내 장비를 공통

데이터 이름으로 데이터를 전송하는 기능이다.

② 사용자 정의 별칭 기능은 특정 데이터를 대체하여 정의하고, 대체 이름으로 데이터를 전송을 위해 사용자 정의 별칭을 만드는 역할을 한다.

넷째, 데이터 연산 기능은 ISO 19848에 명시된 구조에 따라 계산된 데이터를 기록하는 기능을 가진다.

다섯째, 로그 관리 기능은 변경 이력 및 데이터 접근 시간을 기록해야하며, 최소 30일 동안 저장되어야 한다.

3.2 ISO 19848(DIS) 요구사항 분석

항해통신 장비의 경우 IEC 61162 표준 시리즈를 기반으로 데이터 인터페이스가 가능한 반면에 기관부 및 기타 장비부의 데이터 인터페이스는 표준화가 이루어져있지 않아 상호 데이터 교환 및 통합 관리에 대한 어려움이 존재한다. 이러한 이유로 기관부 및 기타 장비부에 대한 식별자 및 데이터 구조의 규칙에 대해 정의한 표준인 ISO 19848이 등장하게 되었다.

ISO 19848 표준에서는 표 3.4와 같이 데이터 교환 모델로 데이터 채널 모델과 시계열 데이터 모델에 대해 정의하고 있다.

Table 3.4 Categorization of requirements in ISO 19848

Category	Requirement	Description
Data Channel	Channel ID	Universal ID(naming entity and ship ID)
		Local ID(naming rule and local data name)
	Data Channel Property	Data channel type, format, range, name, unit and remarks
Time Series Data	Periodically Updated Data	Numeric value from sensors / transmitters and result of calculation
	Irregularly Updated Data	Alarm information, status information and manually input value

1) 데이터 채널 모델

데이터 채널 모델은 기관부 장비 및 자동화 장비 내에 설치된 각 센서들의 데이터 스트림에 대한 정적 정보를 제공하며, 채널ID(ChannelID)와 데이터 채널 속성으로 구성된다. 첫째, 채널ID는 범용ID(UniversalID)와 로컬ID(LocalID)로 나뉘며 이러한 채널ID의 구조를 그림 3.1에 나타내었다.

① 범용ID는 그림 3.1 (a)의 구조로 선박 내 데이터 채널을 식별하기 위한 목적으로 로컬ID를 생성하는 개체를 관리하는 역할을 하는 이름개체(NamingEntity)와 선박 IMO(International Maritime Organization) 식별번호 혹은 국가나 지역에서 제공하는 식별자를 사용하는 선박ID(ShipID)와 로컬ID로 구성된다.

② 로컬ID는 그림 3.1 (b)의 구조로 선박 데이터 채널을 로컬에서 식별하기 위한 목적으로 선박에 탑재된 구성요소 및 시스템에 대해 이름을 붙이는 명명규칙(NamingRule)과 명명규칙에 따라 정해진 데이터 채널의 식별자인 로컬 데이터 이름(LocalDataName)으로 구성된다.

```

UniversalID = [protocol]"/"/NamingEntity ShipID LocalID
NamingEntity = authority
ShipID       = Path-element
Path-element = "/" unreserved
Path-elements = Path-element | Path-element Path-elements
    
```

(a) Universal ID

```

Local ID      = NamingRule LocalDataName
NamingRule   = Path-element
LocalDataName = Path-element
Path-element = "/" unreserved
Path-elements = Path-element | Path-element Path-elements
    
```

(b) Local ID

Fig. 3.1 Channel ID structures

둘째, 데이터 채널 속성은 데이터 채널유형, 형식, 범위, 이름, 단위 및 비고로 구성된다.

① 데이터 채널유형은 숫자 값, 평균 값, 알람 및 상태와 같은 데이터 채널 식별에 사용되며 유형, 업데이트 주기, 계산 주기와 같은 하위 속성을 가진다.

② 형식은 데이터 형식을 설명하는데 사용되며, 유형과 제한으로 구성된다.

③ 범위는 장비가 사용할 수 있는 데이터 범위를 설명하는데 사용되며, High와 Low로 구성된다.

④ 이름은 선박 탑재 장비 및 계측기에 지정된 이름을 설명한다.

⑤ 단위는 측정값에 적용되는 기준을 나타내고 기호, 환산 계수, 수량으로 구성된다.

⑥ 비고는 데이터 채널에 대한 임의의 보완 정보를 설명하고 제조업체, 유형 및 위치 등으로 구성된다.

2) 시계열 데이터

시계열 데이터는 시간 순서대로 기록, 측정 및 정렬된 데이터를 의미한다. 이는 시간 경과에 따라 측정값의 경향을 분석하고 가장 최근의 측정값을 공유하는데 사용되며, 주기적 데이터와 비주기적 데이터로 나누어진다.

① 주기적 데이터는 규칙적으로 제공되는 센서 측정 값이나 계산된 결과 값 등이 포함된다.

② 비주기적 데이터는 알람 정보, 상태 정보 및 수동 입력 값을 포함된다.

시계열 데이터는 하나 혹은 여러 개의 데이터 집합으로 구성된다. 데이터 집합은 특정 시점에 데이터 채널로부터 수집되는 데이터들의 집합을 의미하며, 이에 대한 예시를 그림 3.2에 나타내었다.

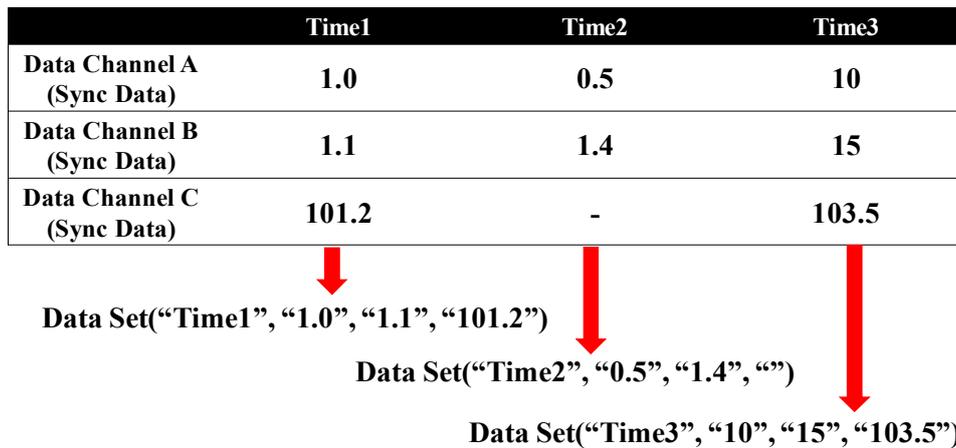


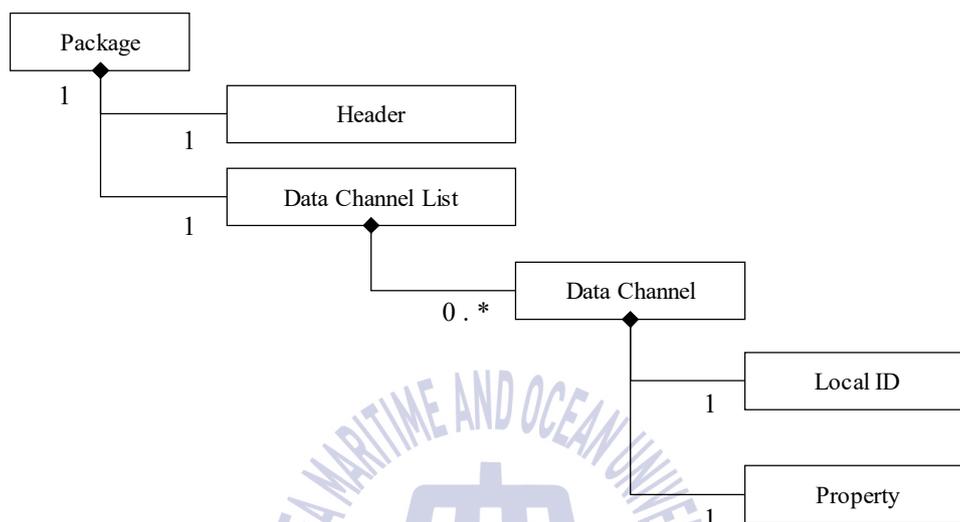
Fig. 3.2 Relationship between data set and time series data

ISO 19848에서는 CSV(Comma Separated Value), XML(eXtensible Markup Language) 및 JSON(Javascript Object Notation)과 같은 언어를 기반으로 데이터 구조를 정의하고 데이터 채널리스트와 시계열 데이터로 구분한다.

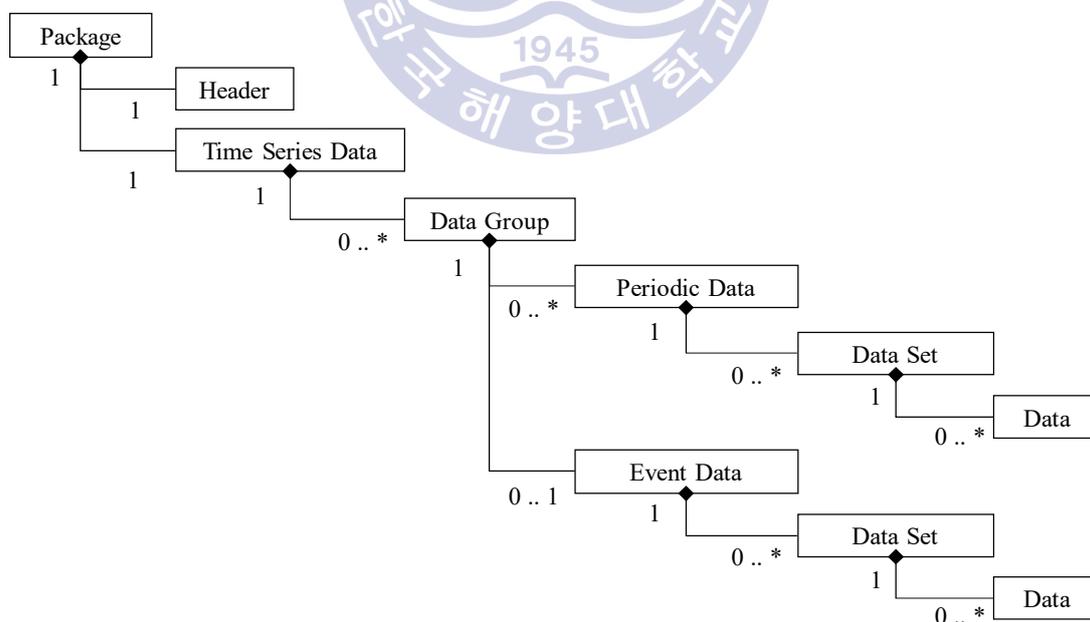
그림 3.3은 데이터 모델의 논리적 구조를 나타낸 것이며, 그림 3.3 (a)는 데이터 채널 리스트 모델의 논리적 구조를 나타낸 것으로 패키지, 헤더, 데이터 채널, 로컬ID 및 속성의 총 5가지 요소로 구성된다. 패키지는 메타 데이터 집합인 헤더와 데이터 본문인 데이터 채널로 구성되고, 헤더는 데이터 채널이 작성되는 시기와 작성자의 정보를 포함하고 있으며, 데이터 채널은 로컬ID와 속성으로 구성된다. 또한 로컬ID는 선박 내 데이터 채널을 식별하기 위한 것이며 속성은 채널의 속성을 정의하고 있다.

그림 3.3 (b)는 시계열 데이터 모델의 논리적 구조를 나타낸 것으로 패키지, 헤더, 시계열 데이터, 데이터 그룹, 주기 데이터, 이벤트 데이터, 데이터 집합 및 데이터의 총 8가지 요소로 구성되어 있다. 패키지는 헤더와 시계열 데이터로 구성되어 있으며, 헤더는 시계열 데이터가 생성되는 시기, 데이터 작성자, 측정기간을 나타내는 메타 데이터를 가지고 있다. 시계열 데이터는 시간 순으로 정렬된 데이터의 목록을 나타내고, 데이터 그룹은 업데이트 주기 등으로 그룹화된 데이터 집합이다. 또한 주기 데이터는 주기적으로 업데이트 된 값의 목

록이며, 이벤트 데이터는 불규칙적으로 업데이트 된 값을 목록이다. 데이터 집합은 동일한 타임스탬프가 있는 데이터 집합을 의미하고, 데이터는 실제 계측된 값을 포함하도록 정의하고 있다.



(a) Data channel list



(b) Time series data

Fig. 3.3 Logical structure of data model

제 4 장 선박 기관부 데이터 관리 시스템 및 모니터링 어플리케이션 설계

본 장에서는 앞서 3장에서 분석한 표준을 기반으로 아키텍처를 설계하고, 구성요소인 인터페이스 장비, 통합 데이터 모듈, 데이터 서버 및 모니터링 어플리케이션의 설계에 대해 다룬다. 또한, IEC 61162-450 데이터에서 장비 식별을 위한 코드화 기법을 제안한다.

4.1 장비 식별 코드화 기법 제안

IEC 61162 데이터 형식을 이용하여 데이터 전달 시, 장비를 식별할 수 있는 구체적인 방안이 없어 본 논문에서는 그림 4.1과 같이 IEC 61162-1/2 센텐스의 5번째 필드인 Transducer ID에 장비를 식별할 수 있는 데이터를 실어서 전달하는 방법을 사용한다. 또한, 그림 4.1과 같이 데이터를 구성할 경우 데이터가 길어질 뿐만 아니라, 장비 식별에 대한 일관성이 떨어지므로, 본 논문에서는 그림 4.2와 같이 ISO 19848 표준에서 제시한 장비식별 코드표를 참조한 장비 식별 코드화 기법을 제안한다.

```
$SERXDR,P,2.5,P,Me/Lubrication/Lo/In/Press*45<CR><LF>
```

Fig. 4.1 Example of equipment identification code (before encoding)

그림 4.2는 장비식별체계를 나타내고 5개의 계층인 System, Component, Content, Position, Item으로 구성되며, 두 개의 ASCII 숫자로 코드화하여 표현한다. System은 기관부 대상 장비인 12개로 구성되고, Component는 11개로 구성된다. 또한 그 외 나머지 하위계층들은 분류로 구성하기에 종류가 많아 고정적으로 구성하기 어려워 필요에 따라 추가하여 사용해야한다.

System		Component		Content		Position		Item	
Index	Name	Index	Name	Index	Name	Index	Name	Index	Name
0	BG(Bilge)	0	Lubrication Systems	0	CSW	0	Aft	0	CloseAngle
1	BF(Boiler Feed Water)	1	Fuel Systems	1	Bilge	1	End	1	Current
2	BT(Ballast Water Treatment)	2	Cooling Systems	2	BoilerWater	2	FishRoom	2	DevTemp
3	BL(Ballast)	3	Air Intake Systems	3	BilgeSeparatedOil	3	Forward	3	Frequency
4	CD(Car Deck)	4	Exhaust Systems	4	CFW	4	ForwardPort	4	FuelCommand
5	FA(Fire Alarm)	5	Valves and Heads	5	Chemical	5	ForwardStarboard	5	FuelIndex
6	FO(Fuel Oil Valve)	6	Emissions Systems	6	CompressedAir	6	In	6	Insulation
7	GR(Generator)	7	Mechanical Systems	7	CondensateWater	7	Lobby	7	Leak
8	LC>Loading Computer)	8	Operating Systems	8	ControlAir	8	Lower	8	Level
9	ME(Main Engine)	9	Battery System	9	CylinderAir	9	MeatFishRoom	9	LoadIndex
10	ML(Miscellaneous)	10	Emergency System	10	CylinderOil	10	MeatRoom	10	L.Press
11	PC(Pump Control)			11	DO	11	Out	11	Position
12	SG(Steering Gear)			12	DrinkWater	12	OutPort	12	Power
				13	Electric	13	PhaseK	13	Press
				14	ExhaustGas	14	PhaseI	14	Temp
				15	FeedWater	15	Port	15	Torq
				16	FO	16	Upper	16	Vaccum
				17	FOSludge			17	Voltage
				18	FW				
				19	HydroOil				
				20	InertGas				
				21	LO				
				22	LSFO				
				23	OilMist				
				24	HT				
				25	SW				
				26	LT				

Equipment identification code Example
 09 00 21 06 13 -> System(ME) Component(Lubrication Systems) Content(LO) Position(In) Item(Press)

Fig. 4.2 Equipment identification code table

그림 4.3은 그림 4.2의 장비식별 코드표를 참조하여 적용한 예이며 장비의 코드가 “0900210613” 인 경우, 그림 4.2의 장비식별코드 예와 같이 메인 엔진에 윤활 시스템의 내부 압력이라는 의미를 가지고, IEC 61162-450 메시지 생성 소프트웨어에서 데이터 전송 시, 데이터 수신 측에서 장비 코드표를 가지고 있으면 어떤 장비의 데이터인지 쉽게 식별할 수 있을 뿐만 아니라, 숫자로 식별하므로 데이터의 용량 또한 최소화 할 수 있는 장점을 가진다.

\$ERXDR,P,2.5,P,0900210613*7E<CR><LF>

Fig. 4.3 Example of equipment Identification Code (after encoding)

4.2 선박 기관부 데이터 관리 아키텍처 설계

앞서 살펴본 ISO 19847/19848(DIS)을 기반으로 설계한 선박 기관부 데이터 관리 시스템 아키텍처를 도식화하면 그림 4.4와 같고, 크게 데이터를 처리하고 전달하는 통합 데이터 관리 모듈(Integrated Data Management Module)과 처리된 데이터를 저장하는 데이터 서버(Data Server)로 구성된다.

통합 데이터 관리 모듈은 실시간 전송, 요청-응답 전송, 파일 전송의 3가지 형태로 어플리케이션에게 데이터를 제공하고, 실시간 전송은 MQTT를 기반으로 어플리케이션에서 구독을 요청한 데이터를 실시간 제공한다. 또한 요청-응답 전송은 어플리케이션에서 REST API를 이용하여 데이터 요청 시 해당 데이터를 제공하고, 파일 전송의 경우, FTP를 통해 파일 형태(CSV)로 데이터를 제공한다.

데이터 입출력 흐름은 기관부 장비에서 발생하는 아날로그 또는 디지털 데이터를 인터페이스 장비(Interface Equipment)에서 IEC 61162-450 형식으로 변환한 후, 통합 데이터 관리 모듈에 송신하고, 통합 데이터 관리 모듈은 데이터를 수신 및 처리하여 데이터 서버에 저장한다. 그 후, 육해상의 특정 어플리케이션에서 데이터 요청이 발생하면 IEC 61162-450, XML, JSON, CSV으로 데이터를 제공한다.

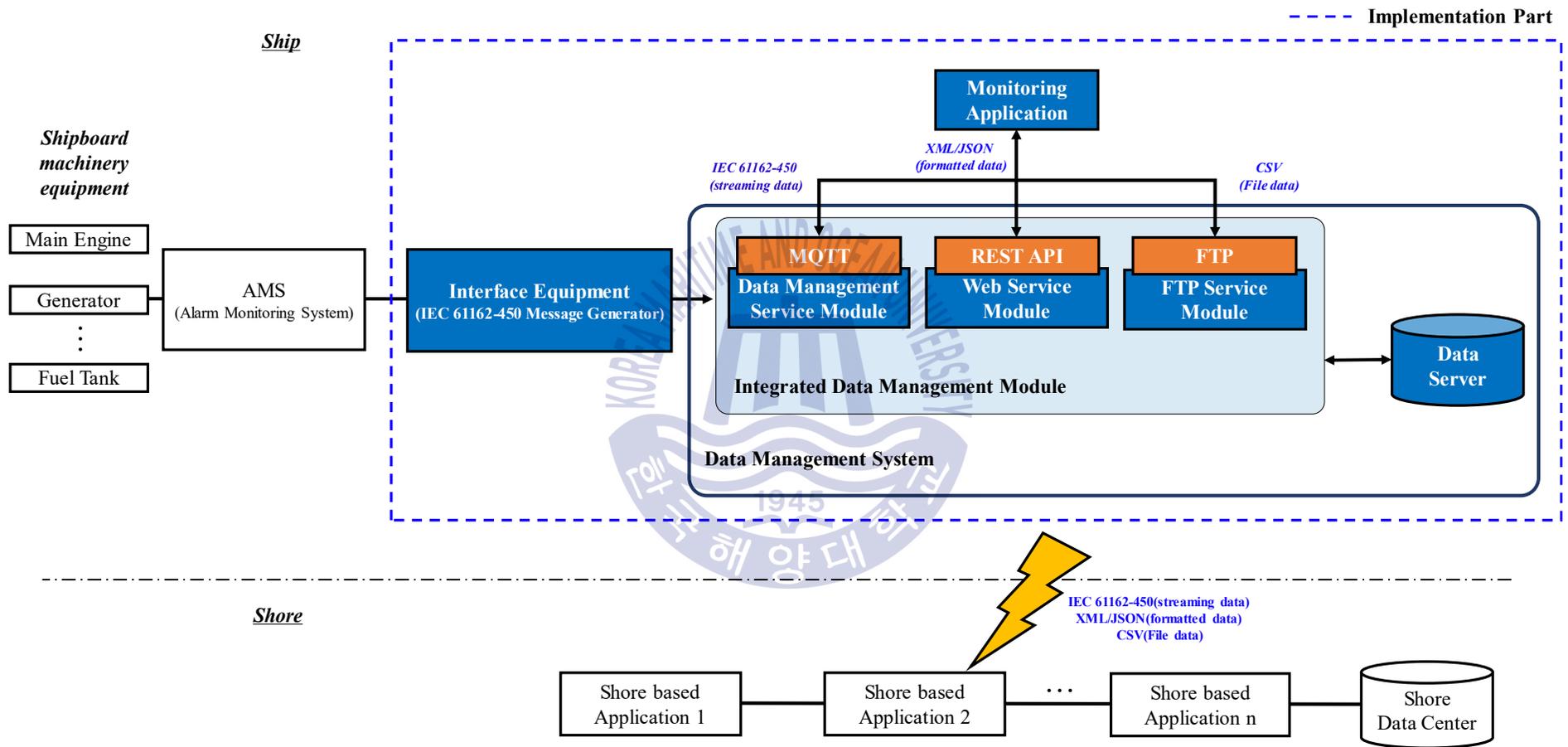


Fig. 4.4 Designed system architecture based on ISO 19847/19848

4.2.1 인터페이스 장비 (Interface Equipment) 설계

실제 인터페이스 장비는 하드웨어장비로 선박 탑재 장비에서 발생하는 데이터를 IEC 61162-450 형식으로 가공하여 전송을 하는 역할을 한다. 이러한 인터페이스 장비의 기능을 소프트웨어로 모사하기 위해 IEC 61162-450 메시지 생성 소프트웨어를 설계하였다.

메시지 설계는 그림 4.5와 같으며 IEC 61162-450 표준에 맞게 TAG/Sentence로 구성되고, “\$”의 앞부분은 TAG(Transmission Annotate and Group) 부분이고, “\$”를 포함한 뒷부분은 Sentence로 실제 데이터를 가지고 있다.

```
UdPbc\0$ERXDR,P,2.5,P,0900210613*7E<CR><LF>
```

Fig. 4.5 IEC 61162-450 message example

Sentence에서 “\$”는 Sentence의 시작이고 “ER”은 표 4.1과 같이 IEC 61162-1에서 정의하는 talker identifier로 엔진룸의 모니터링 시스템이라는 의미를 가지고 있다. “XDR”은 변환기의 측정값을 의미하고, 메시지 구성은 그림 4.6과 같이 구성되어있다. 특히, Transducer ID는 4.1절에서 설명한 장비 식별 코드화 기법을 적용하여 그림 4.7과 같이 장비식별 코드표를 참조하여 코드를 생성한다.

Table 4.1 Classification of IEC 61162-1 talker identifier mnemonics

Type of equipment	Talker identifier	Transmission group
Engine room monitoring system	ER	MISC

XDR(Transducer Measurements)

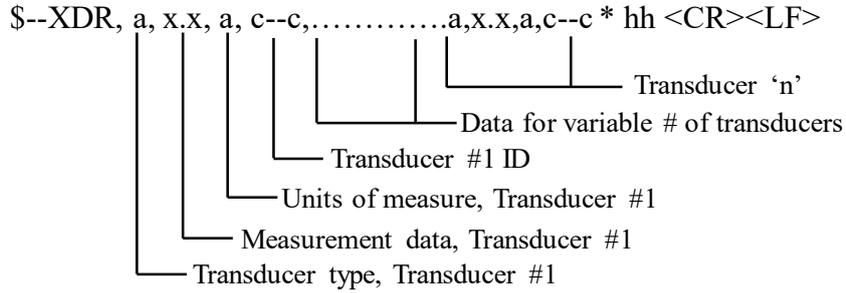


Fig. 4.6 Classification of IEC 61162-1 sentences

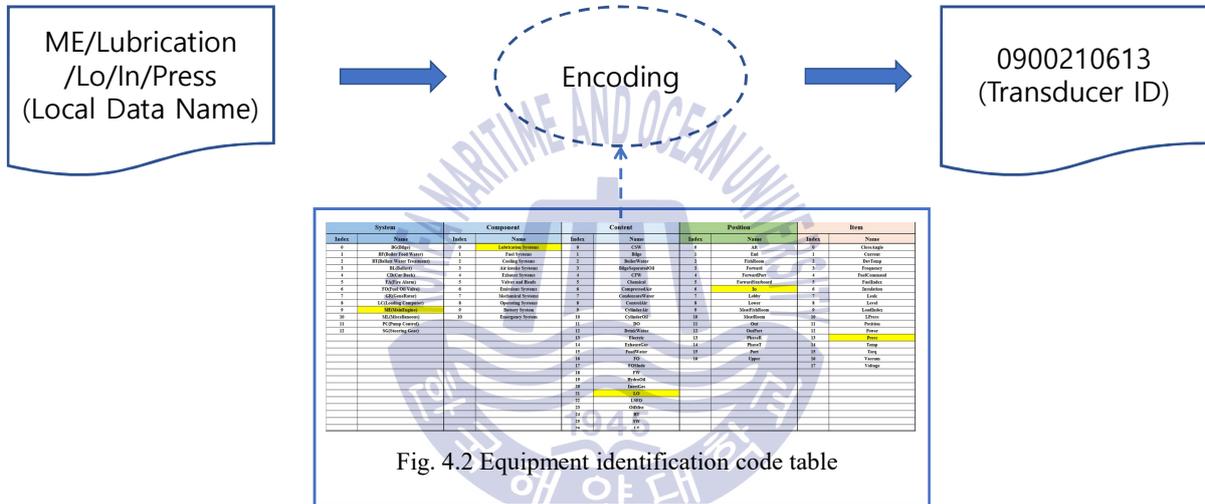


Fig. 4.7 Equipment identification code encoding process

IEC 61162-450 메시지 생성 소프트웨어는 그림 4.8과 같이 4개의 클래스로 구성된다. 첫째, DataGeneratorModule 클래스는 IEC 61162-1/2 데이터의 유효성을 그림 4.9와 같은 과정을 통해 검사를 한 후 생성된 데이터를 버퍼에 저장한다. 그 후 버퍼에 저장된 데이터에 IEC 61162-450 TAG를 추가하여 최종 데이터를 생성하는 역할을 수행한다. 둘째, DataGeneratorForm 클래스는 사용자 인터페이스 구성 및 생성한 IEC 61162-450 데이터를 전송하는 역할을 수행한다. 셋째, Net-workManager 클래스는 IEC 61162-450 데이터 송신을 위한 UDP 연결 및 해제 기능을 수행하고, Udp_Connection과 Udp_DisConnection 메서드로 구성된다. 넷

째, Utility 클래스는 데이터 유효성을 검증하기 위한 체크섬을 생성하는 역할을 하고, CreateChecksum 메서드로 구성된다.

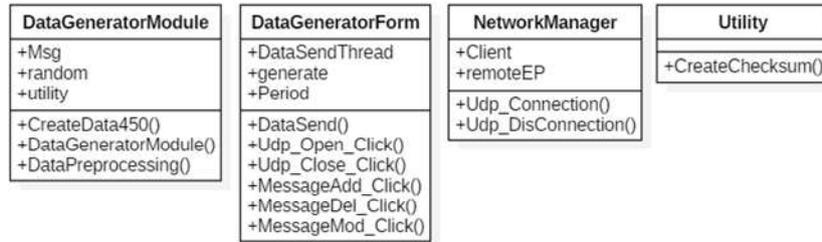


Fig. 4.8 Class diagrams of IEC 61162-450 message generator

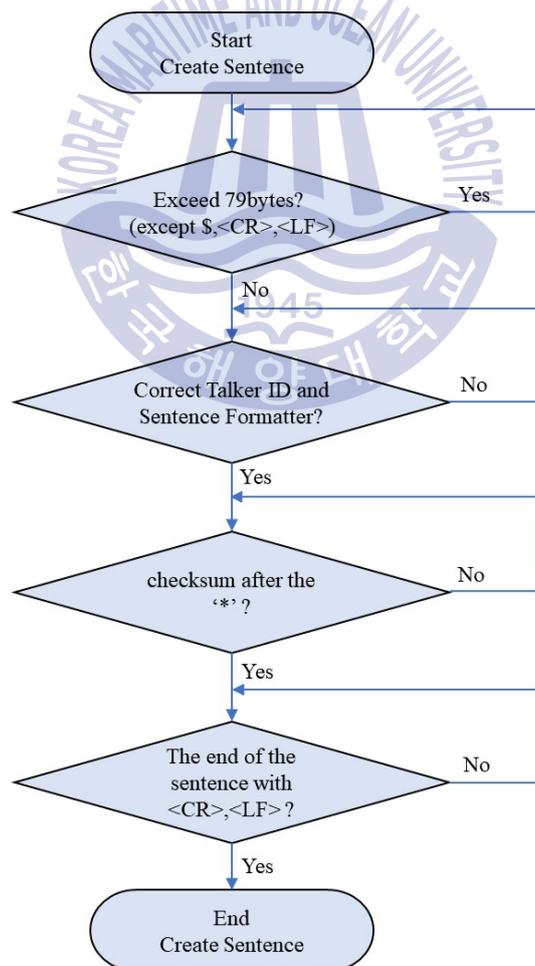


Fig. 4.9 Sentence generation algorithm for the IEC 61162-1/2 messages

4.2.2 통합 데이터 관리 모듈 (Integrated Data Management Module) 설계

통합 데이터 관리 모듈은 데이터 수신, 저장, 데이터 채널리스트 관리 및 실시간 전송 기능을 수행하는 데이터 관리 서비스 모듈, 요청-응답 전송 기능을 수행하는 웹 서비스 모듈 및 파일을 제공하는 FTP 서비스 모듈로 구성된다.

먼저, 데이터 관리 서비스 모듈은 그림 4.10과 같이 20개의 클래스로 구성된다. 그림 4.10 (a)는 주요 기능들을 수행하기 위한 클래스이고, 그림 4.10(b)는 데이터 채널 리스트 모델을 구성요소 각각을 위한 클래스이며, 그림 (c)는 사용자 인터페이스 구성을 위한 클래스이다. 데이터 관리 서비스 모듈에서 주요 기능을 수행하는 클래스는 그림 4.10 (b)로 DBManager, DataChannelListManager, NetworkManager, ExcelProcessing 및 Utility 클래스로 구성된다.

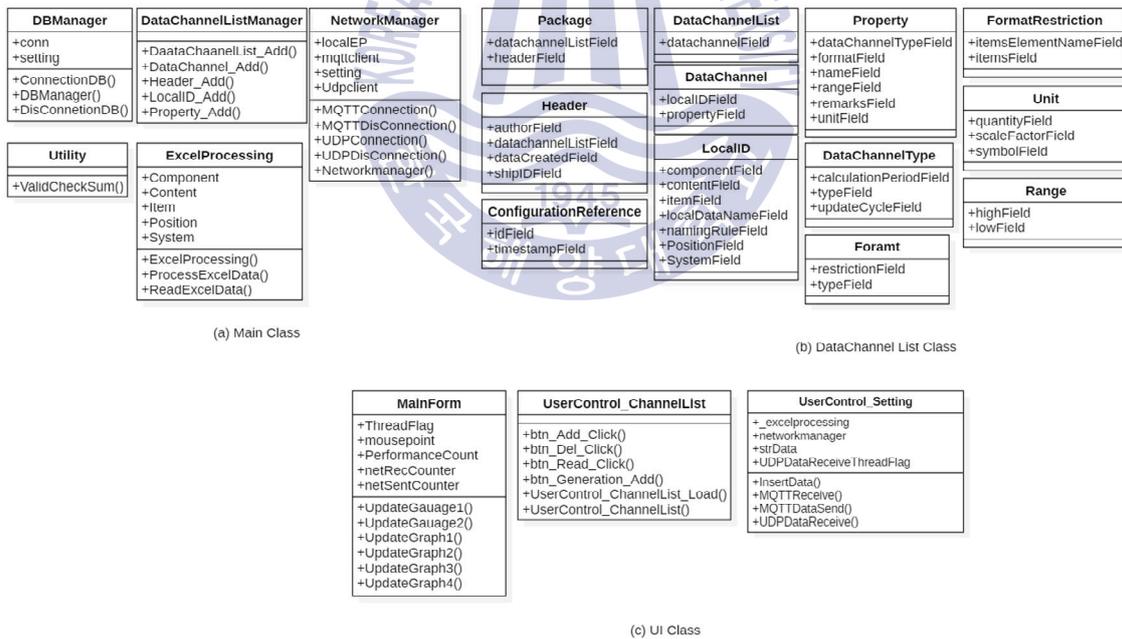


Fig. 4.10 Class diagrams of data management service module

먼저 DBManager 클래스는 데이터베이스와의 연결 및 해제 기능을 수행한다. DataChannelListManager 클래스는 데이터 채널 리스트 관리를 위한 메서드로 구성되고, 그림 4.10(c)의 클래스를 사용하여 데이터 채널리스트의 각각의 구성요소를 생성하는 기능을 수행한다. NetworkManager 클래스는 IEC 61162-450 데이터 수신을 위한 UDP 연결 및 해제 기능을 수행하고, 선내 또는 육상에서 실시간 데이터 요청 시 데이터를 전송하는 MQTT 연결 및 해제 기능을 수행한다. Utility 클래스는 수신한 IEC 61162-450 데이터의 체크섬을 계산하여 데이터의 유효성을 검사한다. ExcelProcessing 클래스는 수신한 IEC 61162-450 데이터의 네 번째 필드인 Transducer ID를 그림 4.3의 장비 식별 코드 테이블을 이용하여 디코딩 과정을 수행하는데 그 과정은 그림 4.11과 같이 표현된다. 이를 위해 장비 식별 코드 테이블을 미리 읽어와 저장하는 ReadExcelData 메서드와 디코딩 과정을 수행하는 ProcessExcelData 메서드로 구성된다.

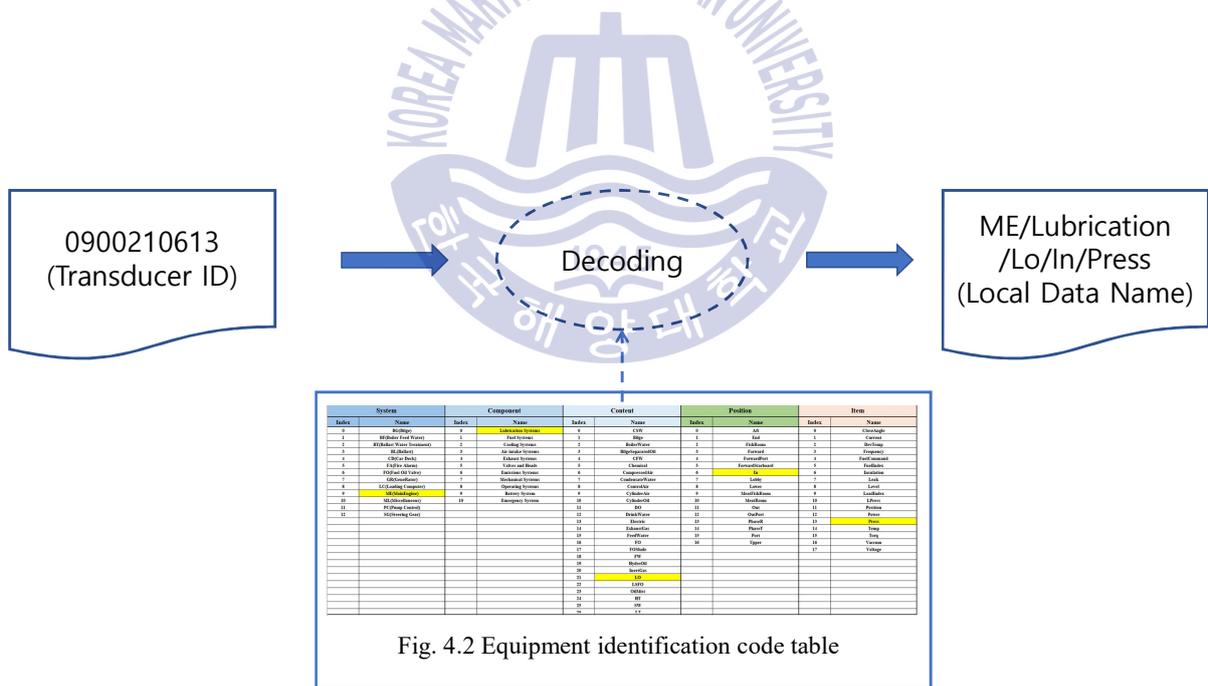


Fig. 4.11 Equipment identification code decoding process

웹 서비스 모듈은 그림 4.12와 같이 15개의 클래스로 구성되며, 그림 4.12 (a)는 주요 기능을 수행하는 클래스이며, 그림 4.12 (b)는 Time Series Data Model의 구성요소의 클래스이다.

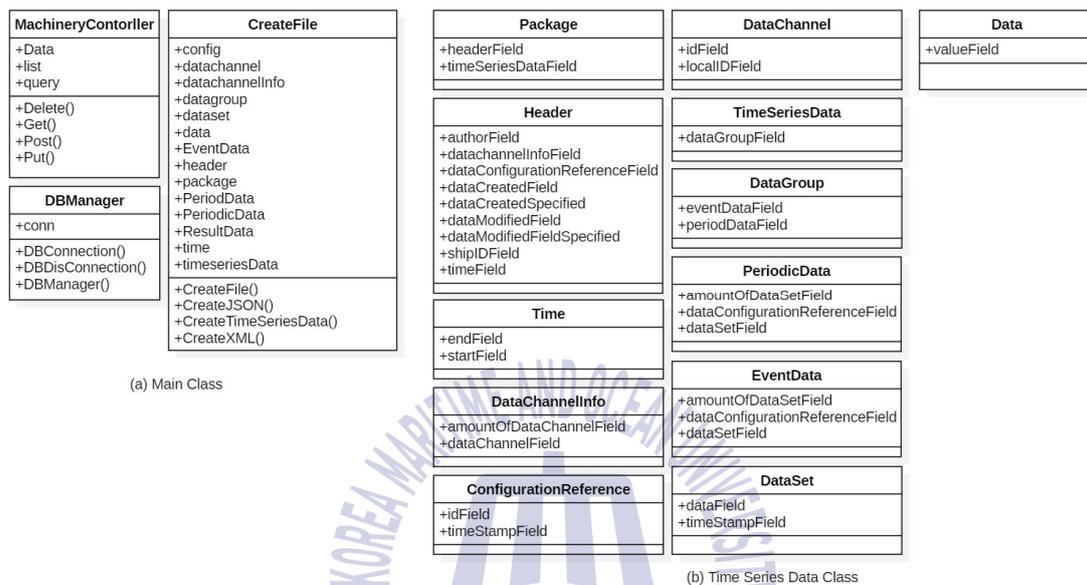


Fig. 4.12 Class diagrams of web service module

웹 서비스 모듈에서 주요 기능을 수행하는 클래스는 그림 4.12의 (a)로 3개의 클래스로 구성된다. MachineryController 클래스는 REST API의 기능인 정보의 조회, 삭제, 생성 및 수정 기능을 수행하고 메서드로는 Delete, Get, Post 및 Put로 구성된다. DBManager는 데이터베이스의 연결 및 해제 기능을 수행하고 DBOpen와 DBClose 메서드로 구성된다. CreateFile은 데이터베이스에서 가져온 데이터를 XML 또는 JSON 형태로 제공해주는 기능을 수행하며, Time Series Data 구조를 생성하는 CreateTime-Series 메서드, JSON 형태로 정보를 제공하는 CreateJSON 메서드 및 XML 형태로 정보를 제공하는 CreateXML 메서드로 구성된다.

4.2.3 데이터 서버(Data Server) 설계

데이터 서버는 인터페이스 장비에서 생성된 데이터를 통합 데이터 관리 모듈의 데이터 관리 소프트웨어에서 수신 및 처리하여 저장한다. 그림 4.13은 데이터베이스 구조로 데이터베이스 이름은 선박명이 되고, 테이블명은 ISO 19848에서 정의한 데이터 채널리스트의 LocalDataName이 된다. 테이블의 각 칼럼은 time, Transducer Type, value 및 unit으로 구성되며, 각 칼럼에 대한 설명은 표 4.2와 같다.

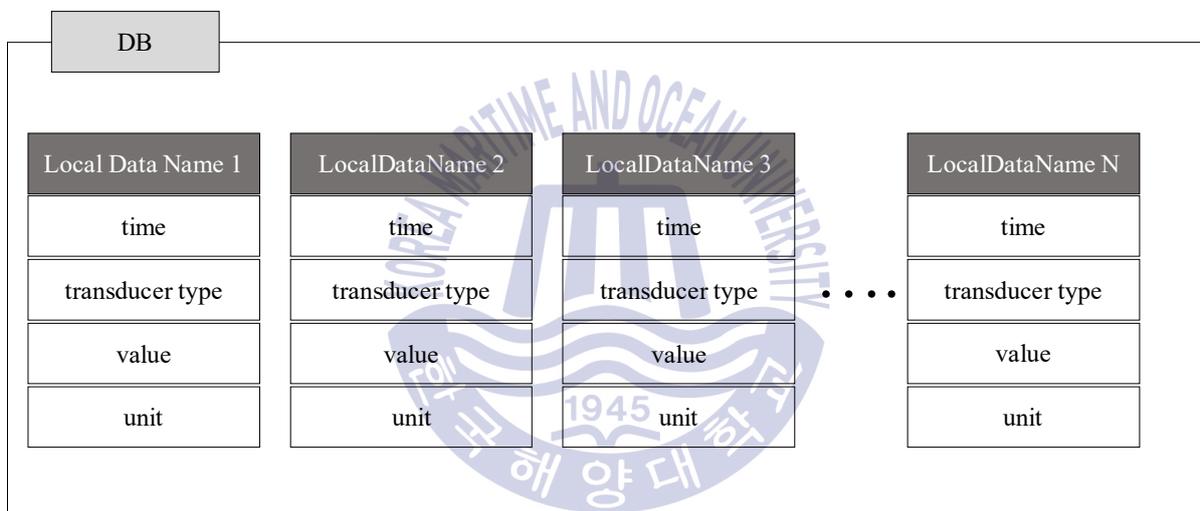


Fig. 4.13 Database table structure

Table 4.2 Column and description of database table

Column name	Description
time	save time
transducer type	sensor type
value	filed value
unit	unit of field value

4.3 모니터링 어플리케이션 설계

모니터링 어플리케이션은 설계된 통합 데이터 관리 모듈의 유용성을 검증하기 위해 설계하였다. 그림 4.14는 클래스 다이어그램으로 8개의 클래스로 구성되어있고, 그림 4.14 (a)는 사용자 인터페이스 구성을 위한 클래스이고, 그림 4.14(b)는 실시간 전송 서비스, 요청-응답 전송 서비스 및 파일 전송 서비스 기능을 수행하기 위한 클래스이다.

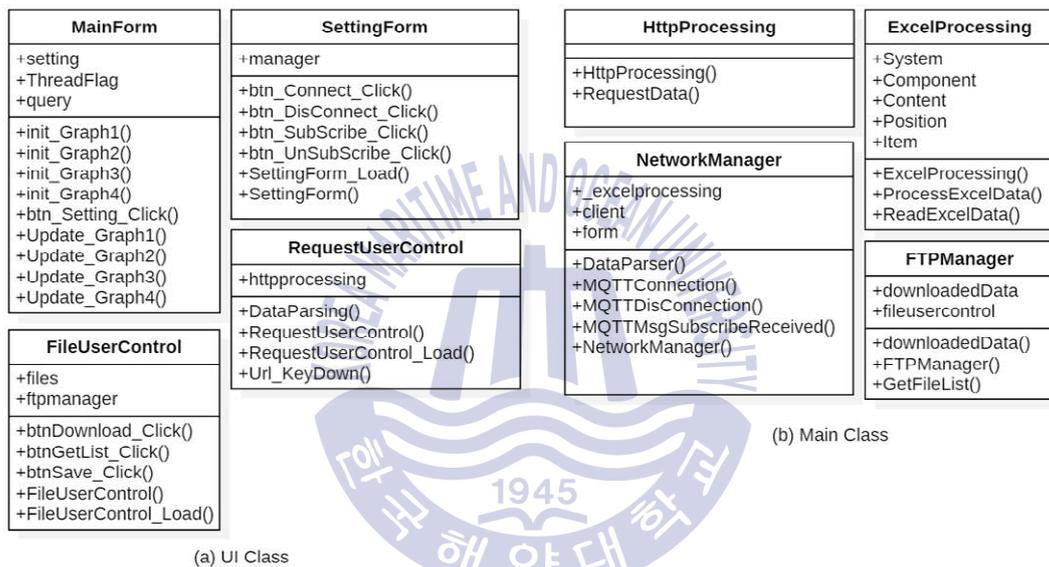


Fig. 4.14 Class diagrams of monitoring application

주요 기능을 수행하는 클래스는 그림 4.14 (b)로 먼저, HTTPProcessing 클래스는 요청-응답 데이터를 수신하는 역할을 수행하고, RequestData 메서드를 통하여 요청-응답 데이터를 수신한다. ExcelProcessing 클래스는 그림 4.11과 같이 IEC 61162-450의 “Transducer ID” 를 디코딩해주는 역할을 하고, 장비 식별 코드표를 읽어와 저장하는 ReadExcelData와 디코딩 과정을 수행하는 ProcessExcelData 메서드로 구성된다. NetworkManager 클래스는 실시간 데이터 수신을 위한 MQTT 연결, 해제 및 수신하는 역할을 수행한다. FTPManager 클래스는 FTP 서버에 접속 및 파일 다운로드의 기능을 수행하고, FTP 서버의 파일 리스트를 가져오는 GetFileList와 파일을 다운로드하는 downloadedData 메서드로 구성된다.

제 5 장 선박 기관부 데이터 관리 시스템과 모니터링 어플리케이션 구현 및 검증

본 장에서는 4장에서 설계한 데이터 관리 시스템 내용을 토대로 시스템을 구현하였고, 모니터링 어플리케이션을 통해 실시간 전송, 요청-응답 전송, 파일 전송 기능에 대해 유용성을 검증한다.

5.1 개발 환경

본 논문에서 다루는 해상 필드 데이터 공유를 위한 선박 기관부 데이터 관리 시스템의 개발 환경은 표 5.1과 같다. 운영체제는 Microsoft Windows 10 Professional 64-bit, 개발 도구는 Microsoft Visual Studio 2015, 개발언어는 .NET Framework 4.5 기반의 C#을 활용하였으며, 데이터 서버의 경우 PostgreSQL 10을 사용하여 구축하였다. 또한 실시간 전송 서비스는 MQTT 프로토콜을 사용하였고, 이를 위해 Mosquitto 라이브러리를 활용하였으며 요청-응답 전송 서비스는 ASP.NET을 활용하여 개발하였다.

Table 5.1 System development environments

Item	Environment
OS	Microsoft Windows 10 Professional 64-bit
Development Tool	Microsoft Visual Studio 2015
Development language	.Net Framework 4.5 based C# and ASP.NET
API/Library	Mosquitto Library
DBMS	PostgreSQL 10

5.2 시스템 구현

4.2절에서 설계한 내용을 바탕으로 시스템의 각 구성요소들을 개발하였으며, 그림 5.1은 인터페이스 장비에서 발생하는 데이터를 모사하여 실험실 수준의 유용성 검증을 수행하기 위해 개발한 IEC 61162-450 메시지 생성 소프트웨어이다. IEC 61162-450 메시지 생성 소프트웨어의 사용자 인터페이스는 ① 소켓 설정 부(Socket Setting), ② 메시지 설정 부(Message Setting) 및 ③ 메시지 출력부(Data View)로 구성되어 있으며, 통신 프로토콜은 UDP(User Datagram Protocol)을 사용하며 IEC 61162-450 메시지를 생성한다.

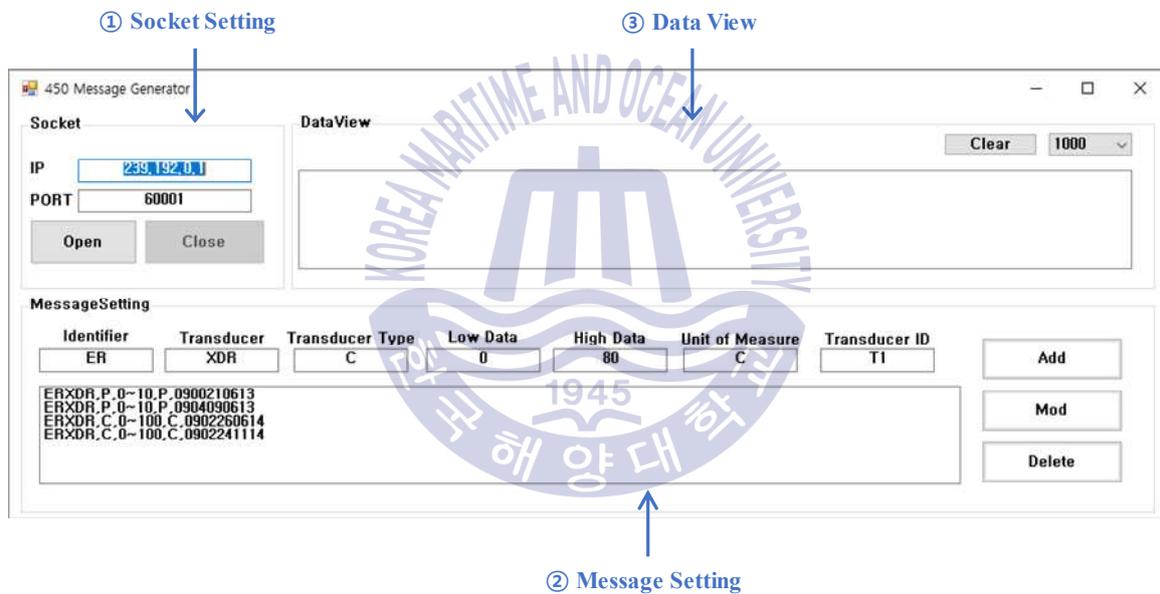


Fig. 5.1 IEC 61162-450 message generator

① 소켓 설정 부는 UDP 소켓을 열거나 닫는 기능이 있으며, IP와 PORT는 IEC 61162-450 표준에 정의된 일반 데이터 그룹인 MISC에 해당하는 239.192.0.1과 60001번을 사용한다. ② 메시지 설정 부는 전송하고자 하는 메시지 설정이 가능하며, IEC 61162-1/2에서 정의하는 데이터 형식에 따라 데이터를 생성하고, 필요에 따라 수정 및 삭제가 가능하다. ③ 데이터 출력부는 현재 전송하는 메시지 데이터를 확인할 수 있으며 필요에 따라 송신 주기의 설정이 가능하다.

그림 5.2는 데이터 관리 서비스 모듈의 메인화면으로 ① CPU 사용량(CPU Usage), ② RAM 사용량(RAM Usage), ③ Ethernet 사용량(Ethernet Usage), ④ 현재 연결된 외부 저장 장치접근 상태(Storage Connection Status), ⑤ 이더넷 링크 상태(Ethernet Link Status) ⑥ UTC(Coordinated Universal Time) 동기화 손실 상태(UTC Sync Loss) 구성되어 있으며, ISO 19847 표준의 요구사항에 따라 선박 데이터 관리 시스템의 상태를 실시간으로 표시한다.

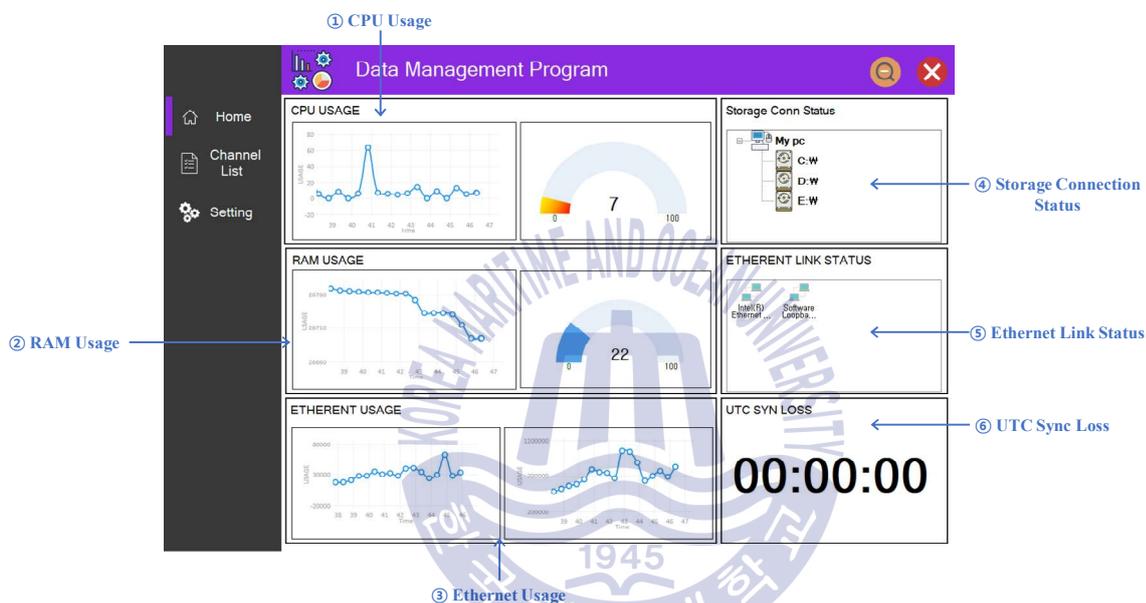


Fig. 5.2 Data management service module (main UI)

그림 5.3은 데이터 관리 서비스 모듈의 데이터 채널 리스트 관리를 위한 UI로 ① Header 설정 부(Header Setting), ② LocalID 설정 부(LocalID Setting), ③ Property 설정 부(Property Setting), ④ 데이터 뷰(Data View)로 구성되어 있다. 그림 3.4 (a)의 데이터 채널 리스트 계층 구조에 따라 입력하고자하는 Header, LocalID 및 Property를 설정한 후에 Create(생성) 버튼을 누르면 XML 형식으로 파일이 생성된다. 또한, ADD(추가) 버튼을 누르면 데이터 채널노드가 추가되고, Delete(삭제) 버튼을 누르면 선택한 데이터 채널 리스트 파일의 가장 최근 데이터 채널이 삭제된다. 마지막으로 Read(파일 읽기) 버튼을 누른 후 확인하고자하는 XML 파일을 선택하면 ④ 데이터 뷰(Data View)에 표시한다.

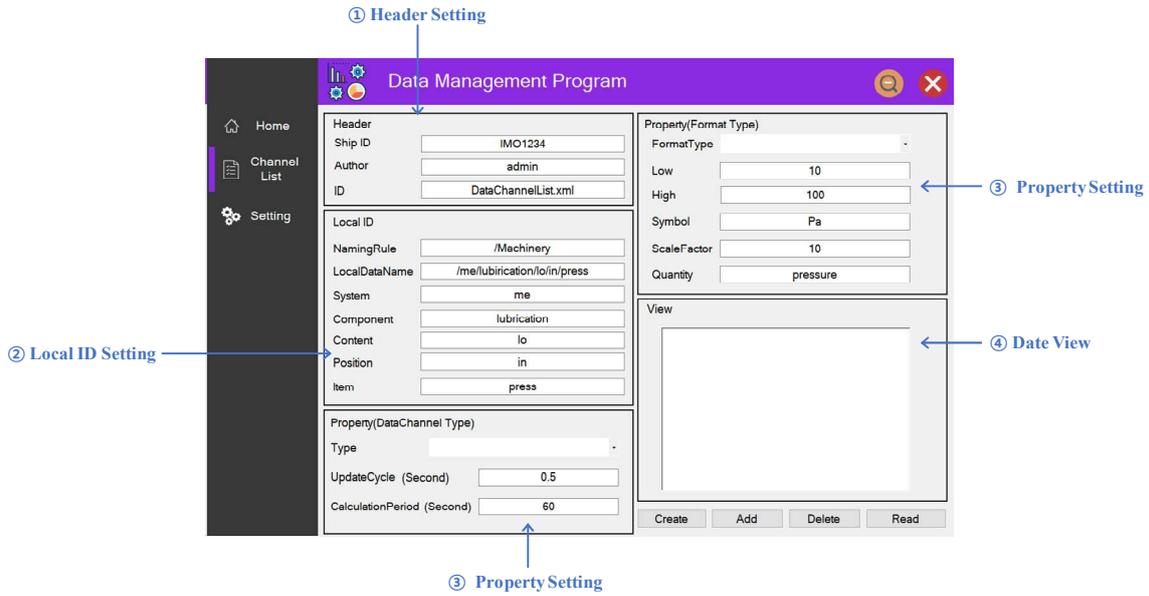


Fig. 5.3 Data management service module (channel list UI)

그림 5.4는 데이터 관리 서비스 모듈의 데이터 저장 및 수신 설정을 위한 설정화면으로 UI는 ① 데이터베이스 설정 부(DataBase Setting), ② 데이터 저장 설정 부(Data Store Setting), ③ 실시간 데이터 설정 부(Streaming Setting), ④ 데이터 뷰(Data View)로 구성된다. ① 데이터베이스 설정 부는 데이터 서버의 연결 및 해제가 가능하며 호스트, 사용자 이름, 비밀번호 및 데이터베이스 이름으로 구성되어 있다. ② 데이터 저장 설정 부는 인터페이스 장치에서 발생하는 데이터 수신을 위한 IP 및 PORT 설정이 가능하고 통신 연결 및 해제가 가능하다. ③ 실시간 데이터 설정 부는 실시간 데이터 송신을 위한 연결 및 해제가 가능하고, IP 및 PORT 설정을 비롯하여 QoS, Retain 및 Topic 설정이 가능하다. ④ 데이터 뷰는 수신한 데이터의 로그를 시간 및 데이터로 표시한다.

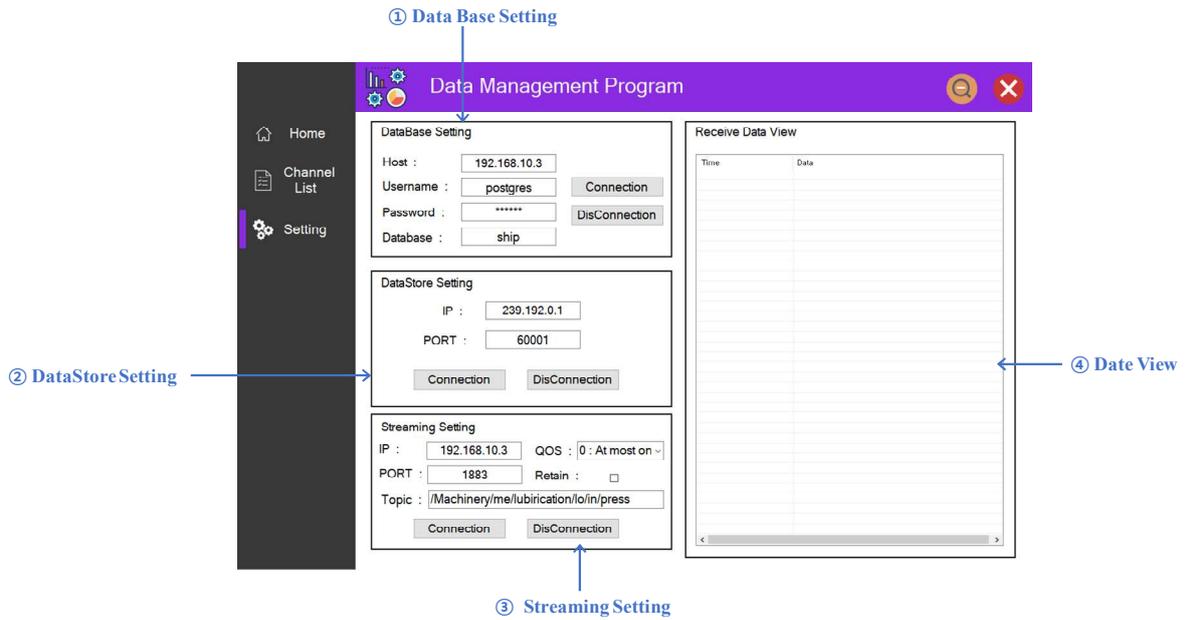


Fig. 5.4 Data management service module (setting UI)

그림 5.5는 요청-응답 데이터를 제공해주는 웹 서비스에 접속한 화면으로 외부 어플리케이션에서 URI로 요청 시 데이터를 제공한다. 특히, 그림 5.5 (a)와 같이 옵션을 주면 XML 형태로 데이터를 제공하고, 그림 5.5 (b)와 같이 옵션을 설정하면 JSON 형태로 데이터를 제공한다.



- (a) Localhost:56914/Machinery/ME/Cooling/LT/IN/TEMP?type=xml
- (b) Localhost:56914/Machinery/ME/Cooling/LT/IN/TEMP?type=json

Fig. 5.5 web service module

그림 5.6은 모니터링 어플리케이션의 메인 사용자 인터페이스로 ① 실시간 그래프 뷰(Realtime Graph View)와 ② MQTT 설정 부(MQTT Setting)로 구성된다. ① 실시간 그래프 뷰는 MQTT 프로토콜을 통해 수신하는 실시간 데이터를 그래프로 표시하는 역할을 하고 ② MQTT 설정 부는 그림 5.7과 같이 실시간 데이터 수신을 위한 ① 연결 설정 부(Connection Setting)와 ② 토픽 설정 부(Topic Setting)로 구성된다. ① 연결 설정 부에서는 실시간 데이터 수신을 위한 IP 및 PORT 설정을 비롯하여 QoS 설정이 가능하다. ② 토픽 설정 부에서는 수신하고자 하는 장비의 토픽이 설정이 가능하다.

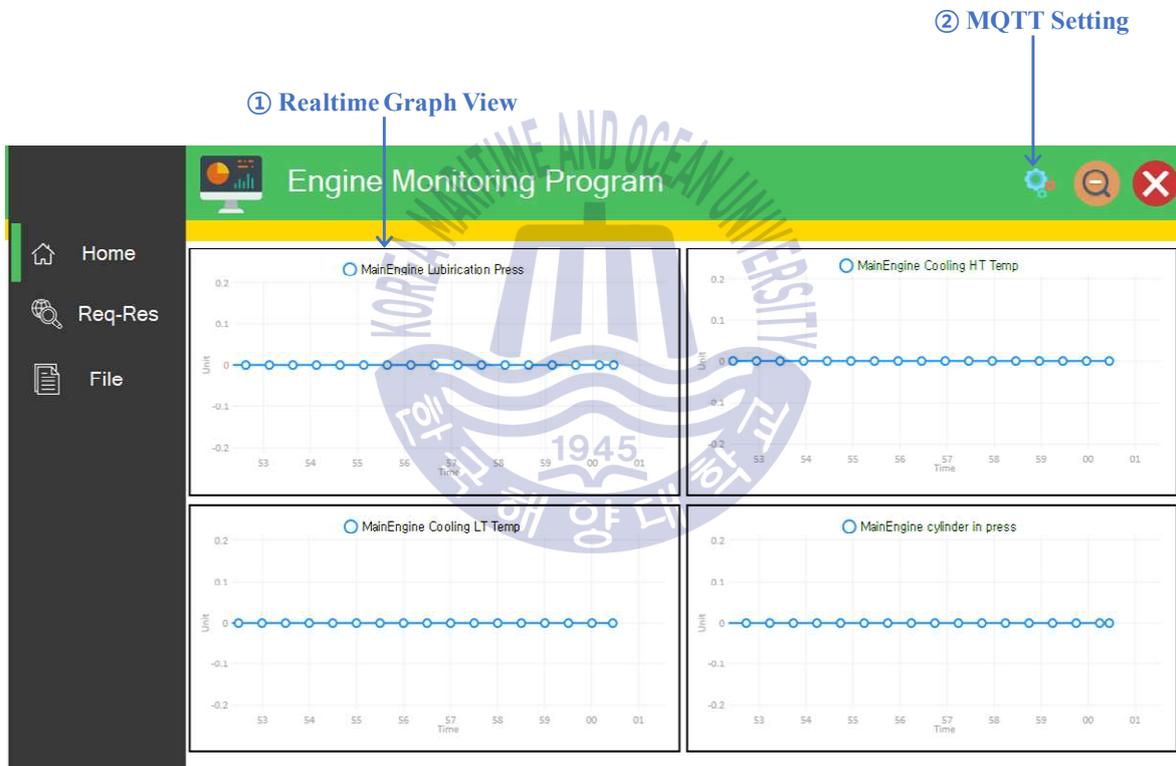


Fig. 5.6 Monitoring application (main UI)

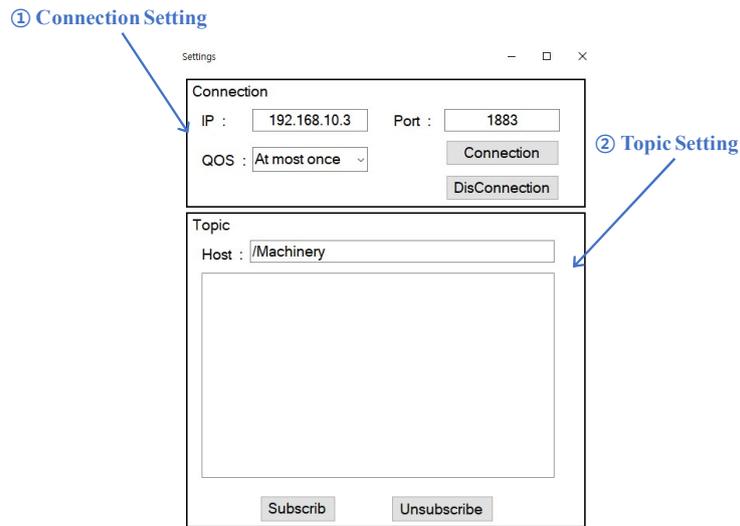


Fig. 5.7 Monitoring application (main UI- MQTT setting)

그림 5.8은 모니터링 어플리케이션의 데이터 요청-응답을 위한 사용자 인터페이스로 ① URI 설정 부(Uri Setting)와 ② 정적 그래프 뷰(Static Graph View)로 구성된다. ① URI 설정 부는 요청하고자하는 장비의 URI를 입력하여 데이터를 요청하는 곳이다. ② 정적 그래프 뷰는 URI를 통해 요청한 데이터를 처리하여 그래프로 표시하는 역할을 한다.

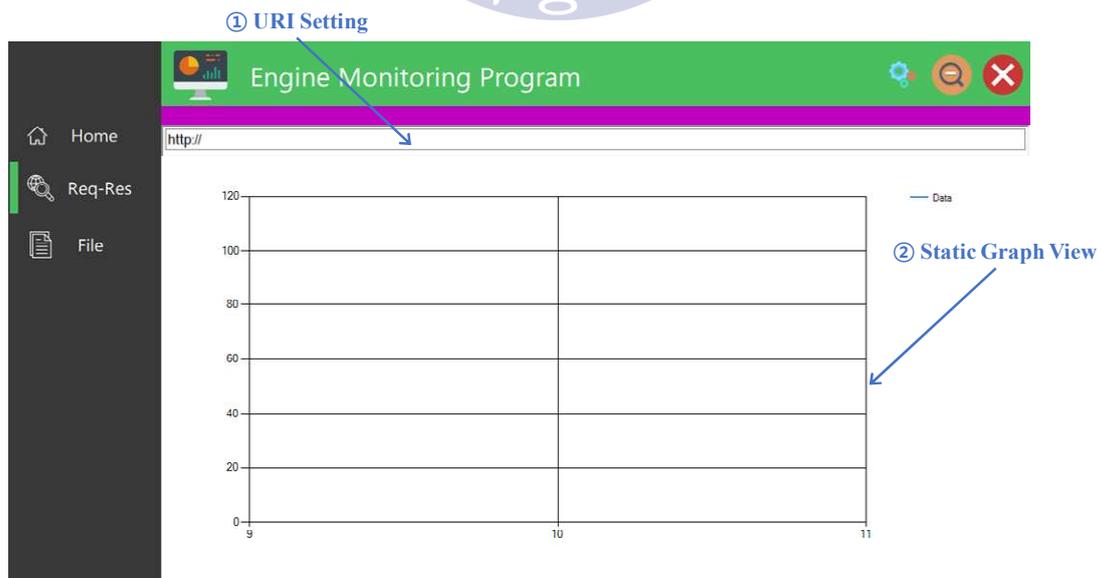


Fig. 5.8 Monitoring application (request-response UI)

그림 5.9은 모니터링 어플리케이션의 FTP를 이용한 파일을 수신을 위한 사용자 인터페이스로 ① 연결 설정 부(Connection Setting), ② 파일 목록 뷰(File List View) 및 ③ 파일 다운로드 설정 부(File DownLoad Setting)로 구성된다. ① 연결 설정 부는 접속하고자 하는 FTP 서버 이름, 사용자 이름 및 패드워드 설정이 가능하다. ② 파일 목록 뷰는 현재 접속한 FTP서버의 파일 목록을 보여주는 역할을 한다. ③ 파일 다운로드 설정 부는 파일 목록 뷰에 파일을 선택했을 때, 파일 이름 및 크기를 보여주고 다운로드를 할 수 있다.

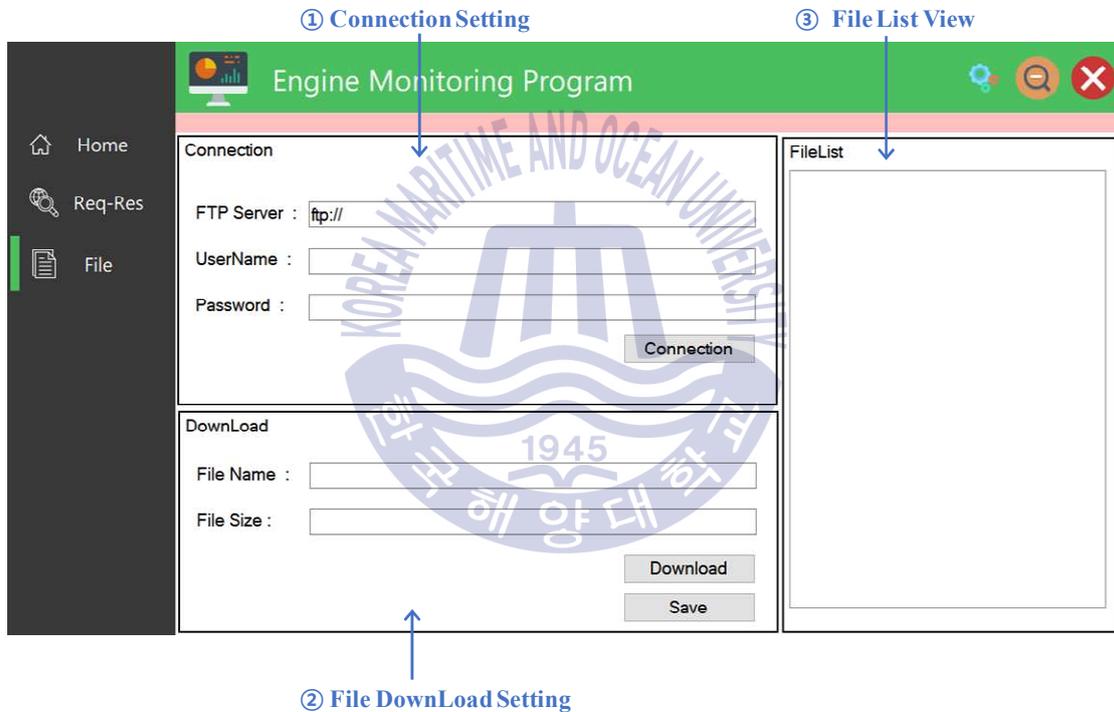


Fig. 5.9 Monitoring application (file transfer protocol UI)

5.3 시스템 검증

개발한 시스템의 유용성에 대해 검증하기 위해서 그림 5.10과 같이 테스트 시나리오를 작성하여 결과를 확인하였다. 메시지 생성기에서 데이터를 생성하고 데이터 관리 시스템에서 처리하여 모니터링 어플리케이션에서 요청하였을 때, 정상적으로 동작하는지 검증하였다. 주요 테스트 항목은 데이터 채널 리스트 파일 관리, 실시간 데이터 요청, 요청-응답 기능, 파일 전송 기능이며, 데이터 관리 모듈의 기능적인 부분이 제대로 동작하는지 확인하였다.

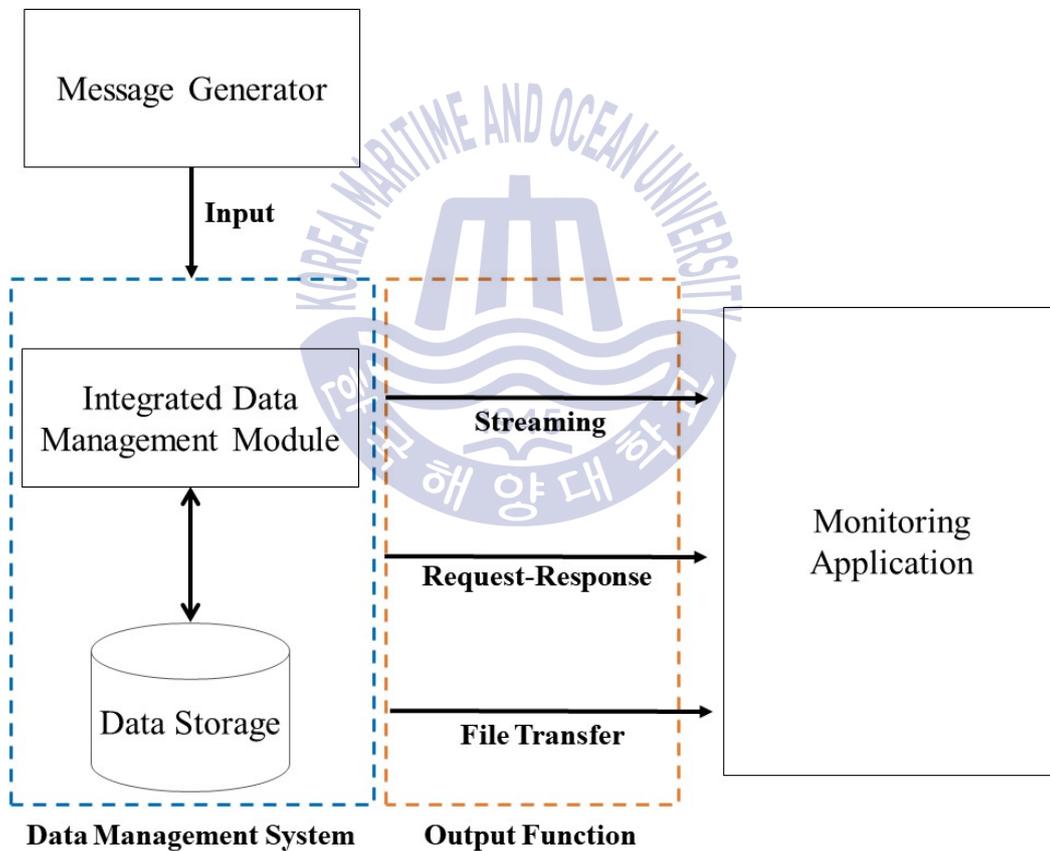
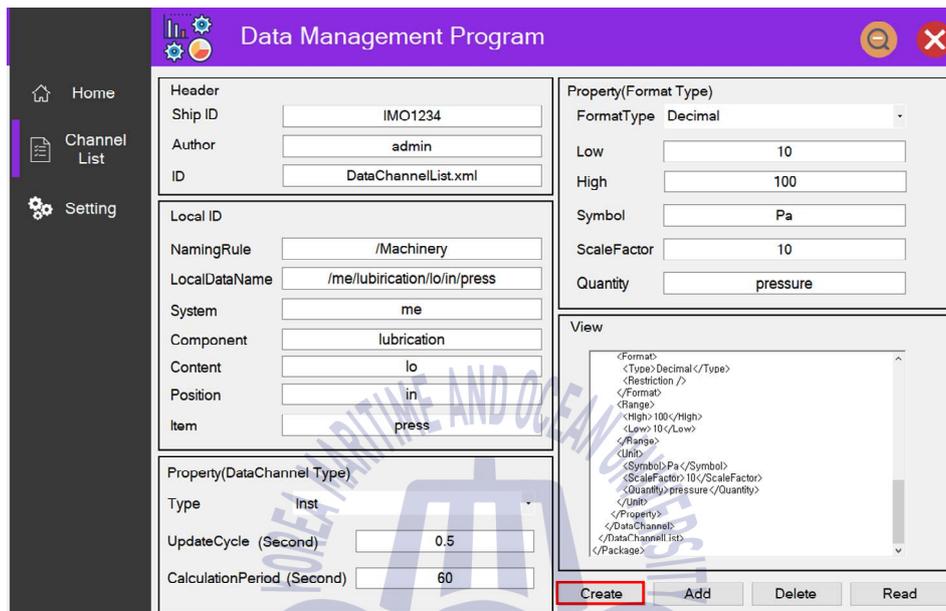


Fig. 5.10 Test scenarios for system verification

그림 5.11은 데이터 채널 리스트 파일 생성의 결과로서 데이터 채널 리스트의 구성요소인 Header, Local ID, Property를 입력한 후에 생성 버튼을 누르면 데이터 채널 리스트 모델 구조로 XML 파일이 생성됨을 확인할 수 있다.



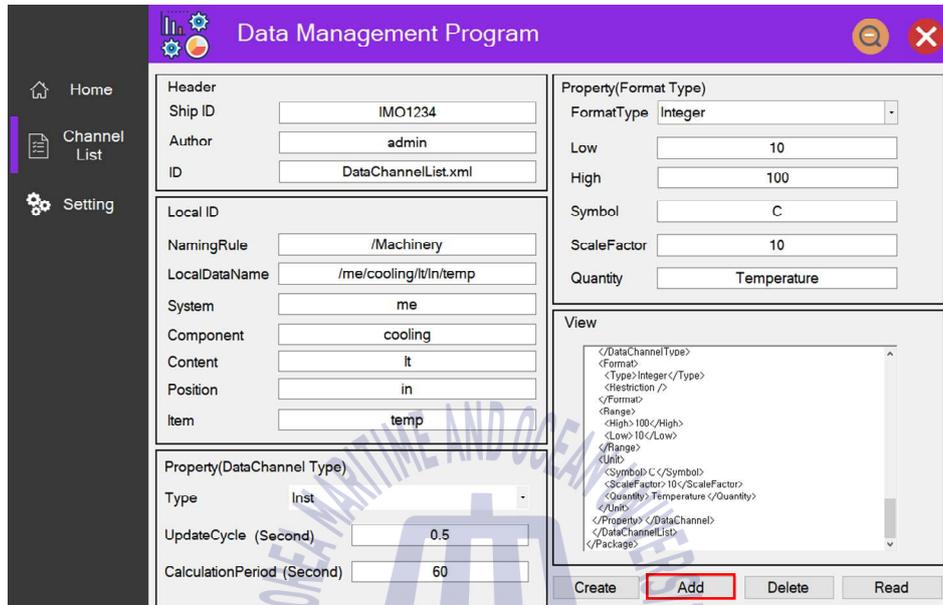
```

<?xml version="1.0" encoding="UTF-8"?>
- <Package xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <Header>
    - <ShipID>IMO1234</ShipID>
    - <DataChannelListID>
      - <ID>DataChannelList.xml</ID>
      - <TimeStamp>2018-12-17T21:59:27.1905102+09:00</TimeStamp>
    - </DataChannelListID>
    - <DataCreated>2018-12-17T21:59:27.1905102+09:00</DataCreated>
    - <Author>admin</Author>
  - </Header>
  - <DataChannelList>
    - <DataChannel>
      - <LocalID>
        - <NamingRule>/Machinery</NamingRule>
        - <LocalDataName>/me/lubrication/lo/in/press</LocalDataName>
        - <System>me</System>
        - <Component>lubrication</Component>
        - <Content>lo</Content>
        - <Position>in</Position>
        - <Item>press</Item>
      - </LocalID>
      - <Property>
        - <DataChannelType>
          - <Type>Inst</Type>
          - <UpdateCycle>0.5</UpdateCycle>
          - <CalculationPeriod>60</CalculationPeriod>
        - </DataChannelType>
        - <Format>
          - <Type>Decimal</Type>
          - <Restriction/>
        - </Format>
        - <Range>
          - <High>100</High>
          - <Low>10</Low>
        - </Range>
        - <Unit>
          - <Symbol>Pa</Symbol>
          - <ScaleFactor>10</ScaleFactor>
          - <Quantity>pressure</Quantity>
        - </Unit>
        - </Property>
      - </DataChannel>
    - </DataChannelList>
  - </Package>

```

Fig. 5.11 Test result of create data channel list file

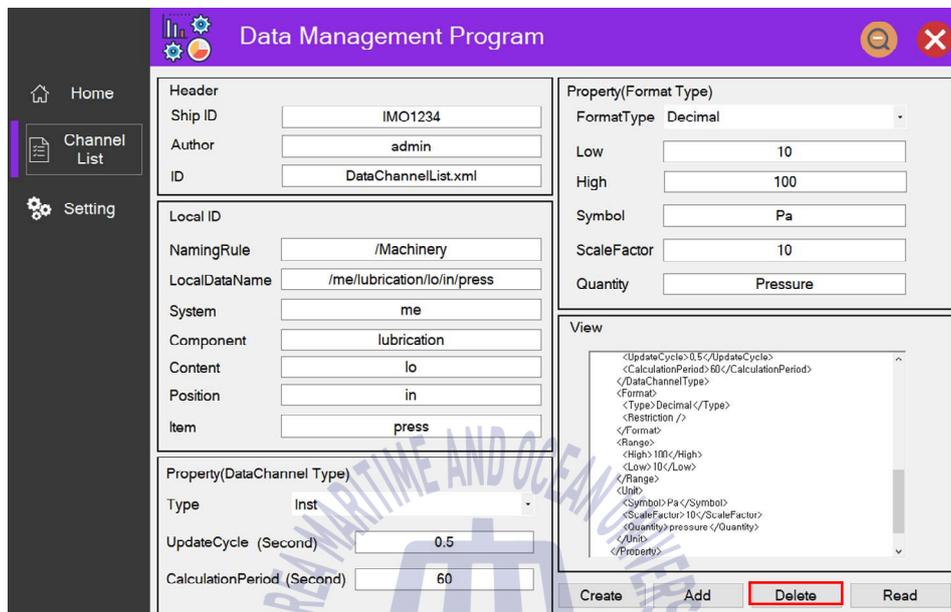
그림 5.12는 기존 데이터 채널 리스트의 데이터 채널을 추가한 결과로서 추가하고자 하는 데이터 채널의 정보를 입력한 후 추가 버튼을 누르면 데이터 채널이 추가됨을 확인할 수 있다.



```
<?xml version="1.0" encoding="UTF-8"?>
- <Package xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <Header>
    <ShipID>IMO1234</ShipID>
    - <DataChannelListID>
      <ID>DataChannelList.xml</ID>
      <TimeStamp>2018-12-17T21:59:27.1905102+09:00</Time
    </DataChannelListID>
    <DateCreated>2018-12-17T21:59:27.1905102+09:00</DateCr
    <Author>admin</Author>
  </Header>
  - <DataChannelList>
    - <DataChannel>
      - <LocalID>
        <NamingRule>/Machinery</NamingRule>
        <LocalDataName>/me/lubrication/lo/in/press</LocalID
        <System>me</System>
        <Component>lubrication</Component>
        <Content>lo</Content>
        <Position>in</Position>
        <Item>press</Item>
      </LocalID>
      - <Property>
        - <DataChannelType>
          <Type>Inst</Type>
          <UpdateCycle>0.5</UpdateCycle>
          <CalculationPeriod>60</CalculationPeriod>
        </DataChannelType>
        - <Format>
          <Type>Decimal</Type>
          <Restriction/>
        </Format>
        - <Range>
          <High>100</High>
          <Low>10</Low>
        </Range>
        - <Unit>
          <Symbol>Pa</Symbol>
          <ScaleFactor>10</ScaleFactor>
          <Quantity>pressure</Quantity>
        </Unit>
      </Property>
    </DataChannel>
    - <DataChannel>
      - <LocalID>
        <NamingRule>/Machinery</NamingRule>
        <LocalDataName>/me/cooling/lt/in/temp</LocalDataName>
        <System>me</System>
        <Component>cooling</Component>
        <Content>lt</Content>
        <Position>in</Position>
        <Item>temp</Item>
      </LocalID>
      - <Property>
        - <DataChannelType>
          <Type>Inst</Type>
          <UpdateCycle>0.5</UpdateCycle>
          <CalculationPeriod>60</CalculationPeriod>
        </DataChannelType>
        - <Format>
          <Type>Integer</Type>
          <Restriction/>
        </Format>
        - <Range>
          <High>100</High>
          <Low>10</Low>
        </Range>
        - <Unit>
          <Symbol>C</Symbol>
          <ScaleFactor>10</ScaleFactor>
          <Quantity>Temperature</Quantity>
        </Unit>
      </Property>
    </DataChannel>
  </DataChannelList>
</Package>
```

Fig. 5.12 Test result of data channel add

그림 5.13은 기존 데이터 채널 리스트의 데이터 채널을 삭제한 결과로서 삭제 버튼을 누르면 가장 최근에 생성한 데이터 채널이 삭제됨을 확인할 수 있다.



```

<?xml version="1.0" encoding="UTF-8"?>
- <Package xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <Header>
    - <ShipID>IMO1234</ShipID>
    - <DataChannelListID>
      - <ID>DataChannelList.xml</ID>
      - <TimeStamp>2018-12-17T21:59:27.1905102+09:00</TimeStamp>
      - </DataChannelListID>
      - <DateCreated>2018-12-17T21:59:27.1905102+09:00</DateCreated>
      - <Author>admin</Author>
    - </Header>
  - <DataChannelList>
    - <DataChannel>
      - <LocalID>
        - <NamingRule>/Machinery</NamingRule>
        - <LocalDataName>/me/lubrication/lo/in/press</LocalDataName>
        - <System>me</System>
        - <Component>lubrication</Component>
        - <Content>lo</Content>
        - <Position>in</Position>
        - <Item>press</Item>
      - </LocalID>
      - <Property>
        - <DataChannelType>
          - <Type>Inst</Type>
          - <UpdateCycle>0.5</UpdateCycle>
          - <CalculationPeriod>60</CalculationPeriod>
        - </DataChannelType>
        - <Format>
          - <Type>Decimal</Type>
          - <Restriction/>
        - </Format>
        - <Range>
          - <High>100</High>
          - <Low>10</Low>
        - </Range>
        - <Unit>
          - <Symbol>Pa</Symbol>
          - <ScaleFactor>10</ScaleFactor>
          - <Quantity>pressure</Quantity>
        - </Unit>
      - </Property>
    - </DataChannel>
  - </DataChannelList>
</Package>

```

Fig. 5.13 Test result of data channel delete

그림 5.14은 MQTT 프로토콜을 이용한 실시간 데이터 송신 테스트 결과로서 MQTT 프로토콜을 위한 IP, PORT 및 사용자가 수신하고자 하는 장비의 Topic을 설정한 후에 연결 버튼을 누르면 MQTT 프로토콜의 중간 역할을 하는 mosquitto 라이브러리에 데이터를 전송하고 전송되는 데이터는 mosquitto 라이브러리 프로그램을 통해 어떤 토픽에 데이터가 얼마나 전송되는지 확인 할 수 있다.

The figure shows the 'Data Management Program' interface. It has a sidebar with 'Home', 'Channel List', and 'Setting'. The main area is divided into three sections: 'DataBase Setting', 'DateStore Setting', and 'Streaming Setting'. The 'Streaming Setting' section is highlighted with a red box and contains the following values: IP: 192.168.10.3, QOS: 0: At most on, PORT: 1883, Retain: unchecked, and Topic: /Machinery. A blue arrow points from this section to a terminal window below. The terminal window shows a series of MQTT messages being sent to the topic '/Machinery', with a table summarizing the data:

Topic	Message Length
/Machinery	37 bytes

The terminal output shows multiple 'Sending PUBLISH' and 'Received PUBLISH' messages, all for the topic '/Machinery' with a message length of 37 bytes. A 'Sending PINGREQ' message is also visible.

Fig. 5.14 Test result of streaming data transfer (publisher)

그림 5.15는 모니터링 어플리케이션에서 실시간 데이터를 수신한 결과로 접근하고자 하는 mosquitto 라이브러리의 IP, PORT 및 Topic을 등록한 후에 연결 버튼을 누르면 실시간으로 수신한 데이터를 그래프를 통해 시각화해주는 것을 확인할 수 있다.

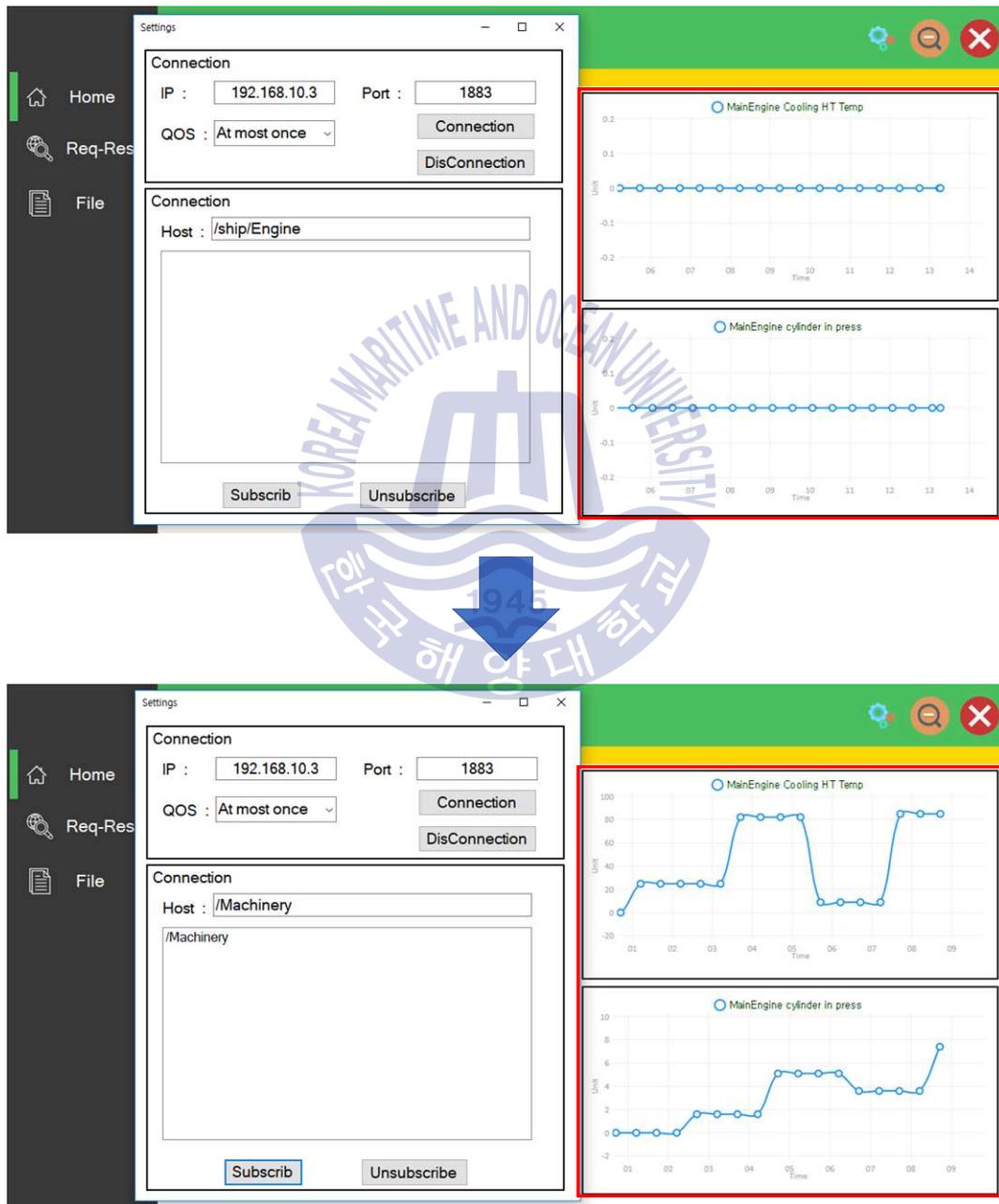


Fig. 5.15 Test result of streaming data transfer (subscriber)

그림 5.16은 요청-응답 데이터 전송 테스트 결과로서 상단의 URL 입력창에 조회하고자 하는 URI와 함께 시작일 2018년 11월 10일(20181110), 종료일 2018년 11월 10일(20181110)로 설정하여 데이터를 요청하게 되면 해당 기간 동안 축적된 데이터를 응답받아 그래프 형태로 시각화해주는 것을 확인 할 수 있다.

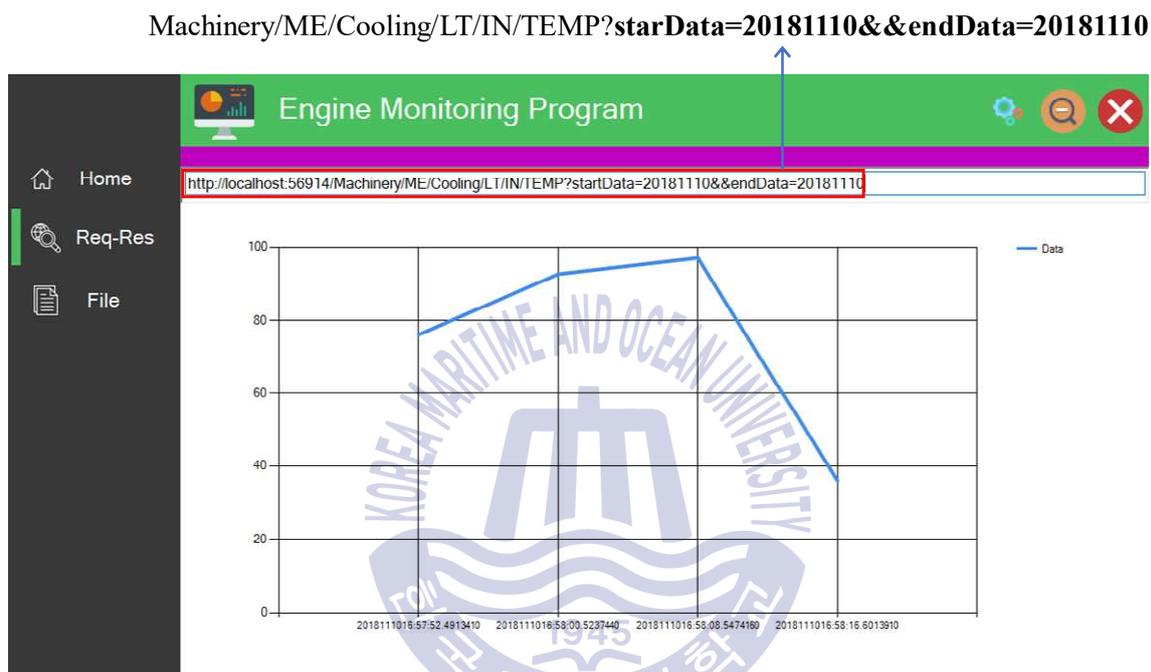
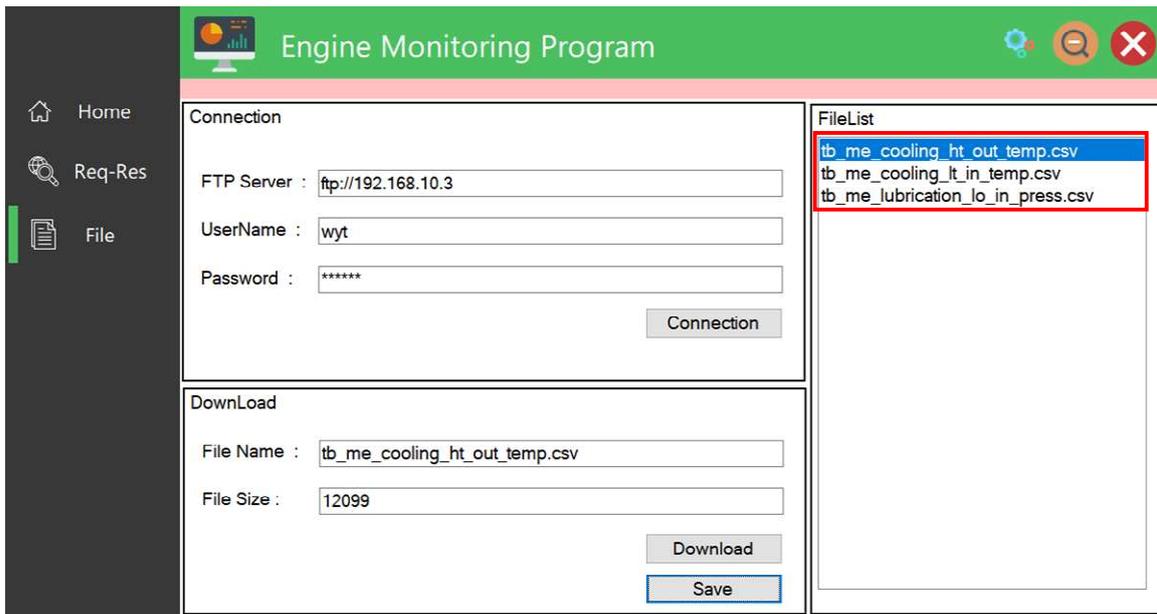


Fig. 5.16 Test result of request-response data transfer

그림 5.17은 파일 데이터 전송 결과로서 그림 5.10 (a)는 데이터 관리 시스템의 FTP 서비스에 접속하였을 때, 현재 저장하고 있는 센서에 대한 CSV 파일 목록을 나타낸다. 또한, 그림 5.10 (b)는 그 중 하나의 파일을 다운로드하여 데이터 확인해본 결과로서, Time은 데이터가 저장된 시간을 의미하고, Transducer Type은 센서의 유형을 의미한다. 또한, unit은 해당 센서 값의 단위를 의미한다.



(a) Result of file download

time	transducertype	value	unit
2018-10-16 09:39:09.30192+09	C	13.1	C
2018-10-16 09:39:15.314881+09	C	33.6	C
2018-10-16 09:39:21.325042+09	C	38.6	C
2018-10-16 09:39:27.334292+09	C	96.6	C
2018-10-16 09:39:33.339285+09	C	22.3	C
2018-10-16 09:39:39.34756+09	C	23.3	C
2018-10-16 09:39:45.356389+09	C	69.4	C
2018-10-16 09:39:51.367178+09	C	2.4	C
2018-10-16 09:39:57.377886+09	C	19.5	C
2018-10-16 09:40:03.384765+09	C	30.3	C
2018-10-16 09:40:09.395958+09	C	13.7	C
2018-10-16 09:40:15.407059+09	C	51.9	C
2018-10-16 09:40:21.41688+09	C	32.1	C
2018-10-16 09:40:27.427121+09	C	87.1	C
2018-10-16 09:40:33.436018+09	C	69.4	C
2018-10-16 09:40:39.444528+09	C	23.9	C

(b) Result of file contents

Fig. 5.17 Test result of file data transfer

제 6 장 결론 및 향후 연구

항해통신장비의 경우 IEC 61162 시리즈를 기반으로 표준화가 이루어져 있어 상호 데이터 교환을 위한 인터페이스가 가능하지만, 기관부 장비들은 상대적으로 표준화가 잘 이루어져있지 않고, 각각 독립적인 형태로 구성되어 있어 인터페이스 및 통합 관리에 어려움이 있다. 이러한 통합 데이터 관리에 대한 필요성 때문에 유럽, 일본 등은 2012년부터 관련된 프로젝트를 진행하였고, 특히 일본에서는 SSAP 프로젝트를 통하여 선박 응용 서비스의 기반이 되는 플랫폼을 구축을 진행하였으며, 그 결과물로 ISO 19847/19848 표준을 제정하였다.

본 논문에서는 해상 필드 데이터 공유를 위한 선박 데이터 서버에 관한 표준인 ISO 19847과 선박 기계 장비부 데이터 표준인 ISO 19848을 분석하였다. 또한, 분석된 내용으로 바탕으로 선박 기관부 데이터 관리 시스템을 설계 및 구현하였고, 선박 기관부 데이터 관리 시스템의 유용성 검증을 위해 모니터링 어플리케이션을 설계 및 구현하였다.

개발한 선박 기관부 데이터 관리 시스템은 통합 데이터 관리 모듈과 데이터 서버로 구성되어 있고, 통합 데이터 관리 모듈은 인터페이스 장비에서 발생하는 IEC 61162-450 형식의 데이터를 수신 및 처리한다. 또한, 사용자의 요청에 따라 실시간 전송, 요청-응답 전송 및 파일 전송 기능을 수행하고, 데이터 서버는 통합 데이터 관리 모듈에서 처리한 데이터를 저장하는 역할을 한다. 또한, 설계한 아키텍처에서는 IEC 61162 데이터 형식을 이용하여 데이터를 전달하는데 장비를 식별할 수 있는 구체적인 방안이 없어 본 논문에서는 ISO 19848 표준에서 제시한 장비 식별 코드표를 참조하여 데이터의 장비 식별 및 코드화 방안을 제시하였다. 마지막으로 선박 기관부 데이터 관리 시스템에 대한 테스트 시나리오를 작성하여 개발한 시스템의 유용성에 대해 검증하였다.

개발한 시스템을 통하여 장비의 제조사마다 상이하던 데이터 관리 및 교환 방법에 대해 일관성이 높아질 것이라 판단되며 데이터에 대한 접근이 용이하여 데이터를 이용한 다양한 활용 및 분석이 기대된다.

향후에는 본 논문에서 구현한 데이터 관리 시스템의 성능적인 요구사항에 대한 시험 연구 등을 진행할 예정이다.



참고문헌

- [1] 황훈규, 김배성, 신일식, 이장세, 2017. 소형 선박 기관부 탑재 장비의 경량화 이더넷 인터페이스를 위한 게이트웨이 시스템의 개발. 한국정보통신학회논문지, 21(1), pp.115-122
- [2] ISO Std. 19847, Ship and marine technology-Shipboard data servers to share field data at sea , ISO, 2013.
- [3] ISO Std. 19848, Ship and marine technology-Standard data for shipboard machinery and equipment , ISO, 2013.
- [4] 이창의, 김달용, 유영호, 신옥근, 2009. NMEA2000을 이용한 임베디드 선박 모니터링 시스템의 개발. 한국마린엔지니어링학회지, 33(5), pp.746-755
- [5] NMEA Std. 0183, Standard for Interfacing Marine Electronic Devices, Version 4.00, NMEA, 2008.
- [6] NMEA Std. 2000, Standard for Serial-Data Networking of Marine Electronic Devices, Edition 3.101, NMEA, 2008.
- [7] 이승준, 김철환, 노경식, 최수영, 강성인, 추영열, 2008. CAN 통신을 이용한 선박상태 모니터링 시스템. 2008년 한국멀티미디어학회 추계학술대회발표논문집, pp.59-63
- [8] 이현, 이동규, 이장명, 2010. CAN을 기반으로 하는 선박 엔진 상태 모니터링 시스템. 제어로봇시스템학회논문지, 19(2), pp.705-709
- [9] 김창영, 이입건, 2015. NMEA2000 프로토콜을 적용한 선박 모니터링 시스템 설계 및 구현. 한국정보통신학회논문지, 19(2), pp.317-322

- [10] 문성미, 장원석, 유영호, 손주영, 2012. NMEA2000 기반 USB/Ethernet 게이트 웨이와 모니터링 시스템 개발. 한국마린엔지니어링학회지, 36(4), pp.407-413
- [11] 김태중, 황훈규, 이성대, 이장세, 장길웅, 박휴찬, 2011. 선박 통합 네트워크의 통신 라이브러리를 이용한 모니터링 어플리케이션의 설계 및 구현. 한국마린엔지니어링학회지, 35(8), pp.1085-1091
- [12] 황훈규, 이성대, 이장세, 장길웅, 박휴찬, 2013. IEC 61162-450에 기반한 선박 통합 정보시스템의 설계 및 구현. 한국정보통신학회논문지, 17(1), pp.101-109
- [13] 서정민, 황훈규, 윤진식, 이성대, 박휴찬, 이장세, 장길웅, 2010. 선박에서 데이터의 통합 관리를 위한 데이터베이스 설계 및 구현. 한국마린엔지니어링학회지, 34(8), pp.1188-1194
- [14] 정승오, 송병호, 류상진, 이성로, 임양섭, 신준우, 2010. 디지털 선박 센서 스트림 데이터의 효율적인 저장 관리. 2010 한국통신학회 종합학술발표회 논문집(하계), 2010(6), pp.115-116
- [15] 박정호, 최유희, 장병태, 2013. 선박 데이터 관리를 위한 Shipdex기반 CSDB Manager 구현. 한국통신학회논문지, 2013(6), pp.1055-1056
- [16] IEC Std. 61162-1, Maritime navigation and radio communication equipment and systems-Digital Interface-Part 1: Single talker and multiple listeners, IEC, 2016.