

선박 기관실 실시간 모니터링 시스템의 HMI 구현에 관한 연구

정 경 열* · 최 욱 현** · 류 길 수***

A Study on the HMI Development for Real-time Monitoring System of Ship
Engine Room

K. Y. Chung · W. H. Choi · K. S. Rhyu

〈 목 차 〉

Abstract

- | | |
|----------------------------------|------------------------|
| 1. 서론 | 3.1 수집한 데이터의 처리 |
| 2. 선박기관실 모니터링 시스템 | 3.2 실시간 데이터의 표시 |
| 2.1 시스템의 전체구성 | 3.3 실시간 표시를 위한 타이머의 사용 |
| 2.2 마스터 컴퓨터의 구성요소와 동작 | 4. 결론 |
| 3. 선박기관실 모니터링 시스템의
HMI 설계와 구현 | * 참고 문헌 |

Abstract

In this paper, the design and implementation of HMI for a real-time monitoring system of ship engine room is presented. The monitoring system gets sensor data of real-time from distributed local units after processing data, displays at CRT and saves in the database. The proposed system consists of four modules, the data management module, the alarm management module,

* 한국기계연구원
** 한국해양대학교 대학원
*** 한국해양대학교 교수

the user interaction module, and the display module. The data management module transfers raw data from a device to engineering data. The alarm management module checks the alarm status of data received from local units. If the alarm occurs, data are displayed at CRT and saved at database.

Key Words : Monitoring System, HMI(Human Machine Interface), Ship Engine Room

1. 서 론

선박기관실 모니터링 시스템이란 선박기관실내의 모든 기기들에 대한 온도, 압력, 회전 수, 속도, 유속 등과 같은 프로세스들의 실시간 데이터를 검출하여 사람이 인식할 수 있도록 표시하고, 고장진단 또는 선박관리 등에 필요한 데이터를 체계적으로 저장, 관리하는 것을 목표로 하는 시스템이다. 이러한 시스템을 개발하기 위해서는 시스템의 고기능화와 사용자 편리성을 추구하여야 하지만, 이를 위해서는 하드웨어가 충분한 처리능력이 갖추어야 할뿐만 아니라 좋은 HMI가 제공되어야 한다.^{[1]-[3]}

최근, 하드웨어로는 마이크로프로세서의 성능향상이 이루어져서 처리속도가 현격하게 증가하고 있으며, 소프트웨어로는 실시간 시스템 구축에 필수적인 멀티태스킹 기법을 지원하는 OS인 "Window 95", "Window NT" 등이 널리 보급되어 이용되고 있기 때문에, 절차형 방식으로 대형 실시간 시스템을 설계할 때 생기는 문제점들을 해결할 만한 환경이 갖추어졌다고 볼 수 있다.^{[4]-[6]} 그리고 방대한 소프트웨어 설계시의 복잡성 문제를 해결하기 위해 80년대 후반부터 제기된 소프트웨어 개발방법론 중의 하나인 객체지향기법을 이용하면 대형의 실시간 시스템의 설계가 용이할 뿐 만 아니라, 시스템 구축 후에도 유지보수의 비용이 절감되는 효과를 얻을 수 있다.^{[7][8]}

본 논문에서는 선박기관실 실시간 모니터링 시스템의 HMI를 구축하는 것을 목표로 하고 있다. 이를 위해 로컬에 산재해 있는 입력 데이터들의 처리 방법, 화면상의 표시 방법 등에 대하여 논한다. 특히 윈도우 시스템에서 실시간 처리를 구현하기 위한 타이머의 사용법에 대하여 기술한다.

2. 선박기관실 모니터링 시스템

2.1 시스템의 전체구성

선박기관실 모니터링 시스템은 Fig. 1과 같이 구성할 수 있다. Master Computer는

Local Unit들(Local Unit 1, Local Unit 2, ..., Local Unit n)과의 통신을 위한 통신 모듈, 획득한데이터의 저장 및 관리를 위한 자료처리 모듈, 데이터의 디스플레이를 위한 실시간 디스플레이 모듈, 다른 Display Unit들(Display Unit 1, Display Unit 2, ..., Display Unit n)과의 통신을 위한 LAN모듈로 구성되어 있다. Local Unit들은 Master Computer와의 통신을 위한 통신 모듈과 제어를 위한 컨트롤 모듈로 구성되어 있다. Display Unit들간의 통신에는 LAN을 이용한 TCP/IP 프로토콜을 이용하고, Master Computer와 Local Unit들과의 통신에는 표준 RS-485를 이용한 마스터/슬레이브 형태의 통신 방식을 이용하도록 하였다. 또한 Master Computer 측에는 윈도우 환경에서 Local Unit 측의 데이터를 획득하기 위하여 VxD 형태의 윈도우용 통신 드라이버를 구현하여 이용하도록 설계하였다.

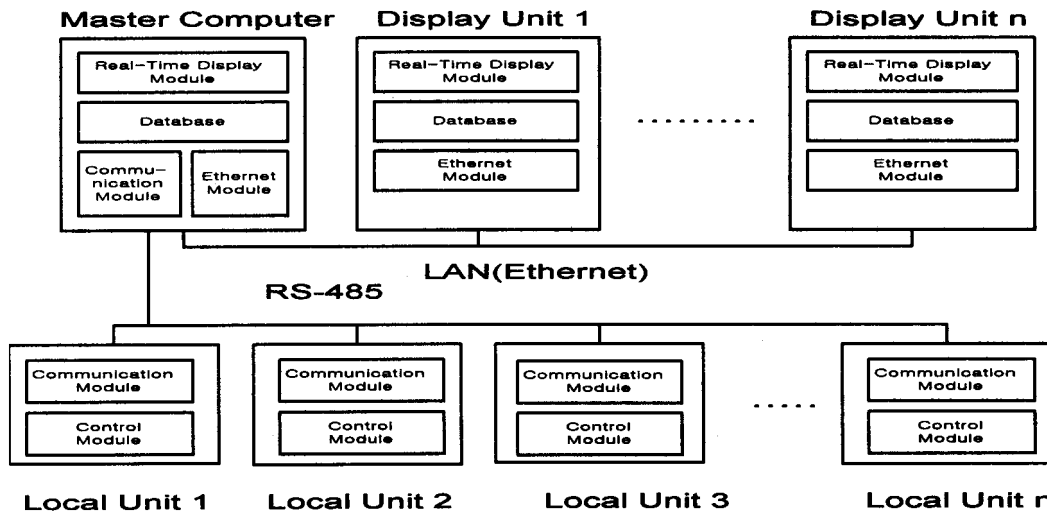


Fig. 1 구현된 선박 기관실 모니터링 시스템의 전체 개략도

2.2 마스터 컴퓨터의 구성요소와 동작

각 모듈과 입출력 동작들을 Fig. 2와 같은 독립된 프로그램인 스레드로 구현하였다. 프린터 출력은 별도의 스레드로 두지 않고 Real-time Display Thread에서 호출되도록 하였다.

각 스레드 간의 우선 순위는 다음과 같이 정하여 두었으며, 이중에서 Data Acquisition thread와 LAN thread는 배타제어가 되도록 구성하였다.

- ① Data Acquisition thread, LAN thread
- ② User Interaction thread
- ③ Real-time Display thread

④ Database thread

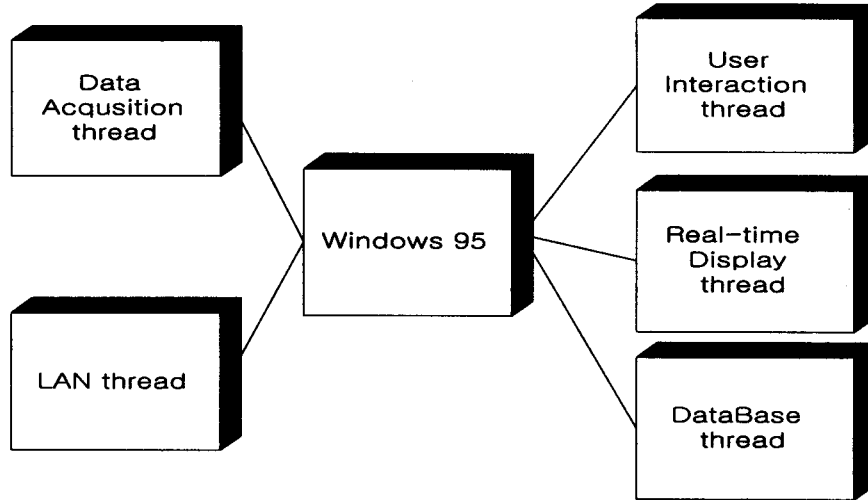


Fig. 2 마스터 컴퓨터의 구성요소

전체 시스템은 Fig. 3과 같은 C++의 클래스 계층도로 구성되어 있으며, 각 스레드는 다음과 같이 동작한다.

(1) Data Acquisition Thread

이 스레드는 RS485 통신카드를 구동시키는 드라이브 프로그램과 프로토콜을 내장한 통신 프로그램으로 구성되며, RS485 통신라인을 통해서 전달된 데이터를 입력받아 디코딩한 후 공유메모리에 저장하는 동작을 반복한다.

(2) Database Thread

이 스레드는 시스템의 모든 초기화 정보를 저장하는 참조 테이블과 시스템의 운영에 대한 이력을 저장하기 위한 이력 테이블로 구성된다.

(3) User Interaction Thread

컴퓨터에서는 입력장치로서 마우스나 키보드가 이용되고 있지만 현장에서는 이들 장치의 빈번한 사용으로 고장율이 많기 때문에 별도의 OCP(Operating Control Panel)를 두어 이것을 입출력장치로 사용하고 있다. 따라서 User Interaction Thread에는 마우스, 키보드와 같은 표준입력장치에 따라 메시지를 처리해 주는 방법과 시리얼 통신을 통하여 OCP로부터의 입력을 받아 메시지를 처리해 주는 두 가지 방법이 모두 구현되어 있다.

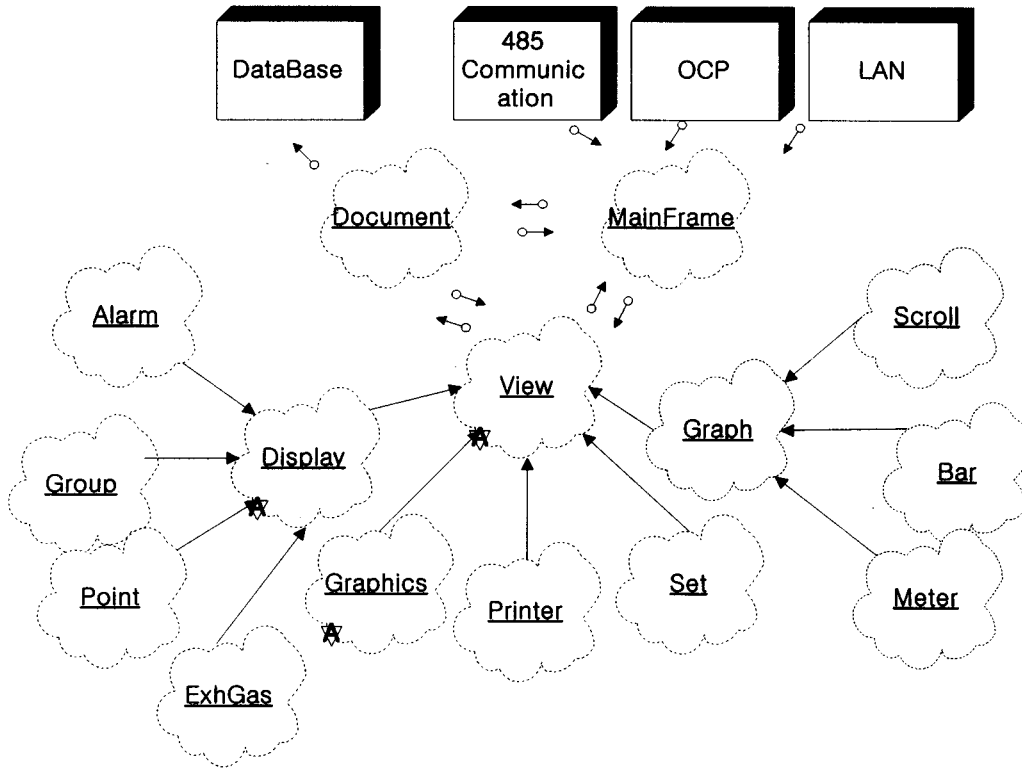


Fig. 3 HMI 시스템을 구성하는 전체 Class의 구성도

(4) Real-Time Display Thread

사용자가 이해하기 쉬운 여러 가지 형태로 화면에 실시간으로 표시해주고, 데이터를 쉽게 선택하여 볼 수 있도록 다섯 가지 형태의 표시기능을 가진다.

3. 선박기관실 모니터링 시스템의 HMI 설계와 구현

본 논문에서 설계한 실시간 모니터링 시스템을 위한 HMI의 기능은 아래와 같다.

- 데이터 수집 및 저장기능 : Local Unit들로부터 전송 받은 데이터를 표현하기에 적당한 형태로 저장한다.
- 데이터 표시기능 : 데이터를 사용자가 이해하기 쉬운 형태로 모니터에 표시한다.
- 사용자 인터페이스 기능 : 사용자의 입력을 받아들여 그것을 적절하게 처리한다.
- 프린팅 기능 : 사용자가 원하는 내용을 프린터로 출력할 수 있도록 한다.
- 설정기능 : 시스템의 작동을 설정하는 기능을 지원한다.
- 보안기능 : 시스템의 안정성을 위해 패스워드를 이용한 보안기능을 지원한다.

사용자로부터 입력을 받아들이는 기능은 User Interaction Thread에 구현하였고, 이를 제외한 모든 기능은 Real-Time Display Thread에 구현하였다.

3.1 수집한 데이터의 처리

(1) 데이터의 저장

I/O 드라이브가 취득한 데이터들은 기관실의 각 부분에 배치된 Local Unit들로부터 수집되기 때문에 HMI의 각 부분에서 곧바로 이용하기에는 부적합하다. 그래서, Table 1과 같이 기관실을 구성하는 주요 구성 요소별로 그룹화 하여 저장하고 있다. Fig. 4는 이러한 데이터를 수집하는 과정을 보여주고 있다. 여기에서 Local Unit에서 수집된 데이터는 물리적 데이터(physical data), 그룹으로 분류된 데이터는 논리적 데이터(logical data)라고 부른다.

(2) 설계한 데이터 구조체의 종류

- Reference Data

데이터가 가질 수 있는 모든 상세 내용(이름, 번호, 종류, 상세한 설명, 알람 발생의 최대 한계와 최소한계 등.....)에 대한 기본 값을 가지는 구조체로서 시스템의 초기화 과정에서 데이터베이스로부터 참조 테이블의 내용을 읽어 저장한다. 이 구조체는 데이터들의 셋팅 환경을 항상 메인 메모리에 상주시켜 놓고 사용함으로써 데이터의 상태를 비교하는데 소요되는 시간을 최소화하고 있다.

- Current Data

샘플링한 데이터 값을 저장하는 구조체로서 일정한 시간 간격으로 최신의 데이터로 갱신된다.

- Alarm Data

현재 발생한 경보 데이터들을 저장하는 구조체이다. 알람의 종류(low, high, low_low, high_high), 인식 유무, 경보 발생 시각을 저장하는 요소 등으로 구성되어 있다.

- Configuration Display

각 디스플레이 방식의 기본 설정 값과 시스템의 운영 중에 사용자가 바꾼 설정 값을 저장하는 구조체이다.

Table 1 선박 기관실 모니터링 시스템의 구성요소별 분류

	M/E CONTROL & SAFETY SYSTEM	G/E COMMON & EM'CY GENERATOR
	M/E L.O. SYSTEM	F.W & S.W PIPING SYSTEM
	M/E COOL WATER SYSTEM	BILGE SYSTEM
	M/E PISTON COOL. SYSTEM	PURIFIER SYSTEM
	M/E FUEL OIL SYSTEM	L.O & F.O PIPING SYSTEM
	M/E EXH. GAS SYSTEM	AIR PIPING SYSTEM
	M/E AIR SYSTEM	AIR COND. & REF. SYSTEM
	SHAFT SYSTEM	CARGO & STEER GEAR SYSTEM
	BOILER SYSTEM	MISCELLANEOUS
	NO.1 GEN SYSTEM	V/V & P/P CONTROL SYSTEM
	NO.2 GEN SYSTEM	
	NO.3 GEN SYSTEM	

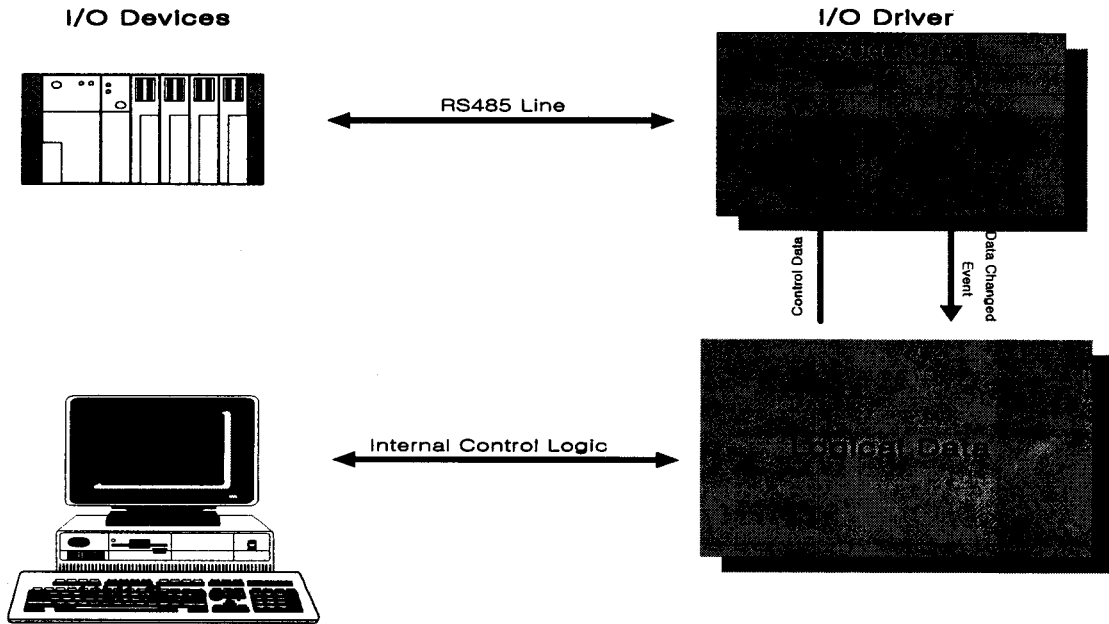


Fig. 4 데이터의 수집 및 정리

3.2 실시간 데이터의 표시

(1) Group Data Display

각 센서의 데이터를 Table 1과 같이 그룹별로 묶어서 표시한다. 그리고 정보 리스트도 각 그룹별로 표시한다.

(2) Alarm Data Display

각 센서의 데이터를 읽어 들여 기준치를 벗어나는 데이터에 대해 경고 메시지와 음을 발생시킨다. 최근 발생한 경고 데이터가 가장 상단에 표시되도록 하고 있다.

경보의 처리는 Fig. 5와 같이 경보가 발생한 점에서 프린터에 출력함과 동시에 경고음을 발생시키고 화면상에는 플리커링 상태로 표시되도록 하며, 사용자로부터 확인 신호가 입력되면 고정상태로 표시 되도록 한다. 이후 경보가 해제되면 프린터에 검정색으로 출력시키고 화면상에서 지운다.

(3) Point Display

사용자의 직책에 따라 표시하고자 하는 데이터의 종류가 다르므로 사용자가 표시하고자 하는 데이터를 설정할 수 있도록 한다.

(4) Exhaust Gas Display

배기가스용 데이터만을 별도로 모아서 디스플레이 하도록 하고 있다. 선박에 따라 메인 엔진의 실린더 수가 다르므로 표시 갯수를 6~8개까지 변경하여 설정 할 수 있도록 하고 있다.

(5) Graph Display

데이터의 변화를 시각적으로 쉽게 관찰할 수 있는 기능을 지원한다. 표시하는 형식도 각 센서의 특징에 맞추어 스크롤, 바, 메타 형태를 지원한다.

① Scroll Graph Display

선택된 데이터를 스크롤 그래프 형태로 표시 해 주도록 하고 있으며, 한 화면에 두 개의 그래프를 표시 할 수 있다. 또한 데이터를 임의로 변경 할 수 있도록 하고 있으며, 이때는 참조 테이블로부터 최대, 최소값을 읽어와 이 값으로부터 스케일을 자동으로 설정해 준다.

② Bar Graph Display

선택된 데이터를 바 그래프 형태로 표시 해준다.

③ Meter Graph Display

선택된 데이터를 바 그래프 형태로 표시 해준다.

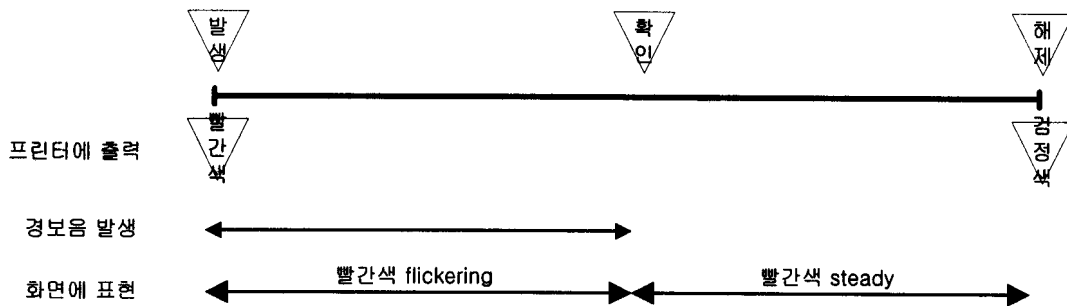


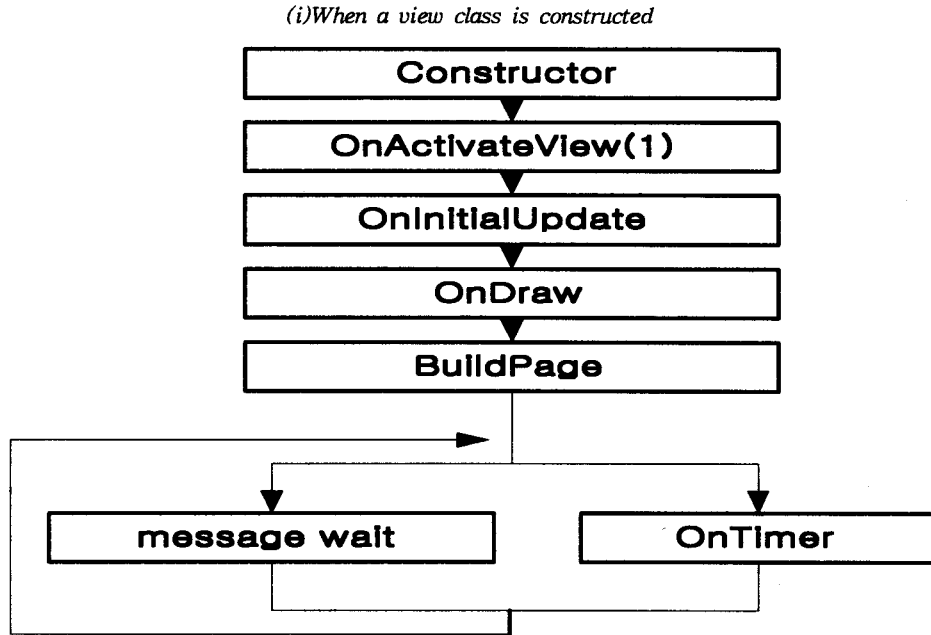
Fig.5 경보의 처리

3.3 실시간 표시를 위한 타이머의 사용

Windows 95나 Windows NT 운영체제에서 실시간 멀티태스킹을 처리하기 위해서는 동시에 여러 개의 타이머를 구동시켜 일정한 시간마다 메모리의 데이터를 갱신하고 이것을 표시해 주어야 한다. 이런 경우 실행되고 있는 디스플레이용 윈도우 창이 많아질 경우에는 각각의 타이머 사용 때문에 심각한 시간 지연이 발생하게 된다. 이것을 방지하기 위해서 본 논문에서는 실제 사용자가 감시하고 있는 활성화된 윈도우 창을 한 개로 제한하여 해당 윈도우 창에서 동작하고 있는 타이머만을 구동하도록 설계하였다. 이때 가동중이면서 화면상에는 표시되고 있지 않은 윈도우 창들은 윈도우 핸들을 저장해 두었다가 해당 윈도우가 활성화 될 때 타이머를 먼저 기동시키도록 해야 한다. 이를 위한 알고리즘은 Fig. 6 과 같다.

여기에서 새로운 윈도우 창이 생성되면 그림 (i)와 같이 이 윈도우 창에 해당하는 클래스가

생성되어 OnActivateView(1)을 실행하게 되는데, 이때 새로운 윈도우 창의 핸들을 Document 클래스에 저장해 두고 SetTimer 함수를 실행하여 실시간 데이터 처리가 되도록 한다. 또한 기존의 윈도우 창에 대해서는 그림(ii)와 같이 해당클래스의 OnActivateView(0)을 실행하여 기존 윈도우 창에 해당하는 타이머를 정지시킨 후 클래스를 소멸시킨다.



(ii) When a view class is destructed

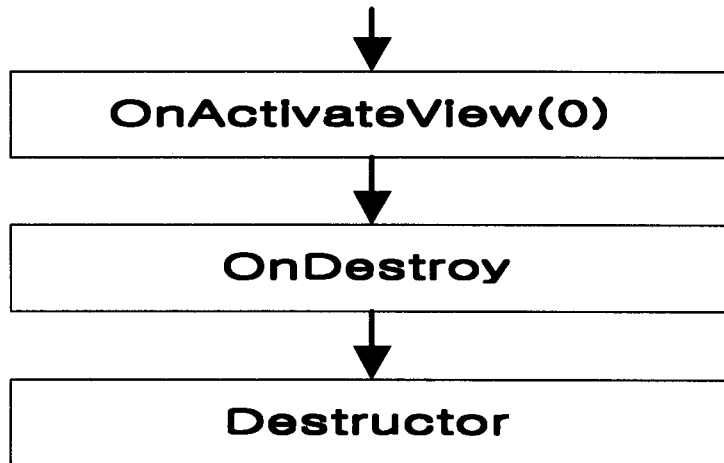


Fig. 6 Algorithm when a view class is constructed or destructed

4. 결론

본 논문에서는 선박기관실 실시간 모니터링시스템을 위한 HMI 구현에 관하여 논하였다. 로컬 유니트로 부터의 데이터 수집은 RS-485 통신기법[9]을 이용하였고, 여기에서는 마스터컴퓨터의 실시간 HMI 설계에 중점을 두었다. 시스템이 대형화되면 시스템의 구현이 복잡하게 되므로 이를 극복하기 위해 객체지향기법을 이용했고, 실시간 시스템의 처리속도를 향상시키기 위해 Windows 95 상에서 멀티태스킹 기법을 적용하여 각 기능들을 구현하였다

현 단계에서는 분 또는 시 데이터를 데이터베이스에 저장하도록 하고 있어서 S/B상태와 같이 동시에 많은 이벤트가 발생하는 경우에는 태스크의 처리속도가 떨어지게 된다. 이와 같은 경우 인공지능 기법을 적용하여 각 상황에 따라 동적으로 태스크의 중요도를 변경시켜 급한 태스크를 중점적으로 실행시키는 기법이 제공될 필요가 있을 것이다.

참 고 문 헌

1. 김병덕, 윤지근, “선박용 모니터링 기술에 관하여”, 한국박용기관학회지, 제13권, 제4호, pp.271-278, 1989.
2. 中川 司, “舶用コンピュータシステムと國際動向”, 日本舶用機關學會誌 第32卷 第2号, pp79 - 85, 1997.
3. SYSTEM MANUAL Volume 1, “NORCONTROL DataChief-7 midi Alarm and Monitoring System”
4. Hassan Gomaa, “Software Design Methods for Concurrent and Real-Time System”, ADDISON WESLEY
5. Jeffrey J.P. Tsai and Steve J.H. Yang, “Monitoring and Debugging of Distributed Real-time Systems”, IEEE Computer Society Press, pp.9
6. Damatic XD, <http://www.valmet.com/automat/business/damatic/dama3.htm>
7. 이광용, 류성열, 정기원, “객체지향 실시간 시스템 개발 방법론”, 정보과학회지, 제11권, 제2호, pp.59 - 74, 1993.
8. 정기원, 류성열, “실시간 소프트웨어의 객체중심설계”, 정보과학회지 제 8권 제5호, pp.61 - 69, 1990.
9. 최대석, “선박기관실 모니터링 시스템을 위한 통신방식에 관한 연구”, 한국해양대학교 대학원 석사학위논문, 1997.

