

선박 주기관 원격제어시스템을 위한 통신방식에 관한 연구

정 경 열* · 류 길 수**

A Study on the Communication Method for a Ship Main Engine Remote Control System

K. Y. Chung · K. S. Rhyu

Key words : Main Engine Remote Control System(주기관 원격제어시스템), Serial Communication(시리얼 통신), Real-time System(실시간 시스템)

Abstract

In this paper, a communication method is proposed for the development of a main engine remote control system. The main engine control system comprises three subsystems such as RCS (Remote Control System), BCS (Bridge Control System), and SS (Safety System). Thus, it is required to exchange data each other among these subsystems. The communication method has simplified hardware through the minimization of communication components where the interrupt method are employed for receiving and the polling method for transmitting. We discuss a methodology of using a ring buffer for data storage physically which has two buffers virtually for the effective use of memory. This communication method presents a good performance in the system which has rather small numbers of communication data.

1. 서 론

실시간 제어시스템의 필드측은 센서, 액츄에이터, 입출력 기기 및 다른 제어시스템들로 구성되어 있으며, 이들 사이에서는 데이터들이 실시간으로 교환되고 있다. 대부분의 시스템에서는 서로 다른 주기를 가진 데이터의 교환이 함께 이루어지고 있는데, 이러한 데이터들을 확실하게 교환해 주고,

주변 노이즈 등으로부터 전기적 내력과 유지보수의 유효성을 향상시키기 위해서는 전용 통신망이 필요하다.^{[1][4]}

선박 주기관 원격제어시스템은 위치적으로 살펴보면 브릿지와 기관실에 있어야 하며, 기능적으로 살펴보면 제어 기능과 안전관리 기능이 있어야 한다. 그러므로 이 시스템을 구축하는 경우 3개의 서브시스템, 즉 BCS(Bridge Control System),

* 한국기계연구원(원고접수일 : 98년 10월)

** 한국해양대학교

RCS(Remote Control System), SS(Safety System)로 구성하는 것이 바람직하다.^{[5],[6]} 실제로 상품화되어 있는 제품들을 살펴보면, NABCO사의 경우에는^[7] 소형선박을 대상으로 하여 RCS의 기능만을 포함하고 있기 때문에 시스템은 통신기능이 필요없는 단순한 구조로 이루어져 있다. 반면에 NORCON사의 경우에는^[8] 대형선박을 대상으로 하고 있기 때문에, 시스템은 3개의 서브시스템으로 구성되어 있으며, 하드웨어도 성능향상을 위해 16비트의 8088MPU를 기반으로 하고 있다. 이 서브시스템들은 데이터의 송수신을 위해 인터럽트에 의한 통신 방식을 채택하고 있어서 하드웨어 구성시 별도의 제어신호선을 필요로 한다. 따라서 하드웨어가 복잡하게 될 뿐만 아니라 소프트웨어 작성시에도 많은 코드를 요구한다. 실제로 NORCON사의 제품의 경우 8259인터럽트 컨트롤러의 IRQ 2개를 이용하고 있으며, 타이머 컨트롤러도 3개나 이용하고 있다.

본 논문에서는 대형선박의 주기관 원격제어시스템을 구축하는 것을 목적으로 한다. 먼저, NORCON사의 단점을 보완하기 위해 인터럽트에 의한 수신 방식과 폴링형태의 송신 방식을 도입함으로써 인터럽트 컨트롤러 및 타이머 컨트롤러를 각 1개씩만 이용할 수 있도록 하여 하드웨어를 간결하게 한다. 이러한 방식을 도입하는 경우 송신측 시스템의 타이밍과 수신측 시스템의 타이밍을 동기화시키기가 대단히 어려우므로 송신을 완료한 후 즉시 수신대기상태로 유지해야 한다. 이 수신대기상태 동안에 수신되고 있는 데이터 중에서 유효한 것을 수집하여 저장하는 처리가 필요하지만 이 처리에는 상당한 시간이 요구된다. 따라서 본 논문에서는 수신 알고리즘을 인터럽트 루틴과 별도의 처리 루틴으로 나누어 실행시키도록 하고 있다. 즉 전자에서는 데이터를 수신하여 버퍼에 저장하도록 하고, 후자에서는 버퍼내의 유효한 데이터를 수집하도록 한다. 또한 원격제어 알고리즘이 대단히 많은 메모리를 차지할 경우를 대비하여 메모리를 절약하는 측면에서 가상 링버퍼를 사용하는 방법을 제안하고 이때에 발생할 수 있는 비정상적인 상태를 수정하기 위한 방법도 제시한다.

2. 원격제어시스템의 구성

본 논문에서 이용하고 있는 원격제어시스템의 하드웨어^[9]는 80186MPU를 기반으로 하여 Fig. 1과 같이 3개의 서브시스템, 즉 BCS, RCS, SS로 구성되어 있다. BCS에서는 회전속도 명령(RPM Command), 제어위치선택, 각종 스위치 조작이 가능하고 이 값들을 RCS에 전송해 주며, 주기관 상태 및 안전관련 값들을 전송받아 패널에 디스플레이 해주고 있다. RCS에서는 주기관의 원격제어기능과 함께, 회전속도 검출, 제어위치, 주기관 상태 표시, 알람표시 등을 해주며, 이 값들을 BCS와 SS에 전송해 준다. SS에서는 주기관을 감시보호하기 위해 회전속도 검출, 과속 검출, 비상정지(Shutdown), 비상감속(Slowdown) 등의 기능을 수행해 주고, 이 값들을 RCS에 전송해 준다.

이와 같이 3개의 서브시스템은 서로 유기적으로 데이터를 주고받음으로써 자신의 기능을 원활히 수행할 수 있다. 각 서브시스템들은 서로 통신을 수행하기 위해 인터럽트용 8259소자와 타이머용 8253소자 각 1개, 시리얼 통신용 8251소자 2개를 이용하여 RS-422A방식^{[10],[11]}으로 연결되어 있다.

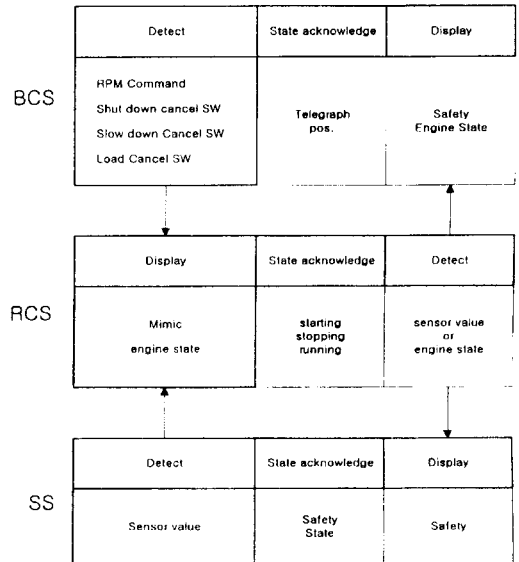


Fig. 1 Function between sub systems of a main engine remote control system

3. 비동기식 시리얼 통신과 프로토콜

본 논문에서는 RCS를 마스터로 하여 BCS와 SS에 교대로 송수신하도록 하고 있다. Fig. 2는 이들 간의 송수신 타임차트를 나타낸 것으로 RCS에서 송신한 후 약간의 시간지연을 가지고 BCS에서 처리해 주고 있음을 알 수 있다. 이것은 수신버퍼에 저장된 데이터 중에서 무효 데이터를 삭제하고 유효 데이터를 추출하는 처리 루틴에 상당한 시간이 요구되고 있음을 의미한다.

3.1 수신용 링버퍼

Fig. 3은 수신용 입력데이터를 저장하고 처리하기 위한 링버퍼를 나타내고 있으며 각각 다음과 같은 의미를 가지고 있다.

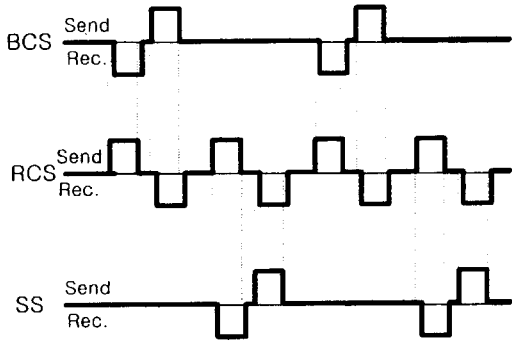


Fig. 2 Time chart of communication between sub systems

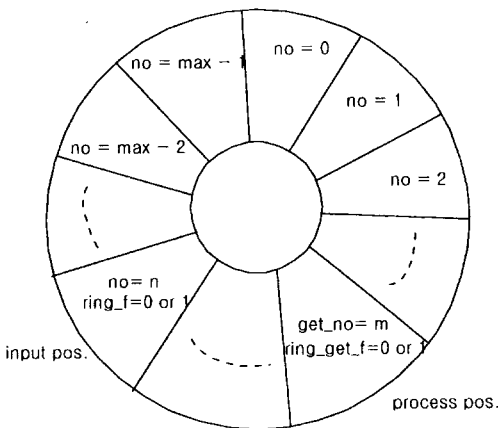


Fig. 3 Ring buffer for receiving data

no : 현재 수신된 데이터를 저장하기 위한 배열 번호를 의미한다.

ring_f : 수신데이터를 저장하기 위해 어느 가상 링을 사용중인가를 나타내며, 0 또는 1의 값을 가지도록 한다.

get_no : 수신된 데이터를 처리하기 위한 장소로서 배열번호를 가진다.

ring_get_f : 수신된 데이터를 처리하기 위해 어느 가상 링을 사용중인가를 나타내며, 0 또는 1의 값을 가진다.

실제 메모리상에는 링이 하나뿐이지만, ring_f 및 ring_get_f 플래그를 이용함으로써 시간상으로는 두 개의 링을 이용하는 것처럼 하였다. 이것을 시간상의 가상링이라고 부른다. 따라서 데이터가 수신될 때마다 no를 1씩 증가시켜 링의 마지막에 도달하게 되면 다음에 수신된 데이터를 no=0, ring_f=1로 하여 저장함으로써 실제로는 동일 링에 저장되더라도 다른 링을 사용하고 있는 것처럼 가상화하였다.

3.2 통신 패킷의 정의

실시간 데이터들은 종류와 환경에 따라 변화하는 주기가 서로 다르다. 예를 들어 주기관 시동시의 회전속도 명령값은 변화가 심하지만, 해수온도의 변화는 거의 없다고 볼 수 있다. 그러므로 상대 시스템에 데이터를 전송하는 경우에 있어서 필요한 데이터만을 송신할 수 있도록 하는 것이 효율적이다. 이를 위해서는 데이터값만을 송신하기보다는 몇 번째 데이터인지를 파악할 수 있도록 하여야 한다.

본 논문에서는 통신 패킷을 정의하는데 이상의 점을 고려하여 Fig. 4와 같은 패킷형식을 이용하고

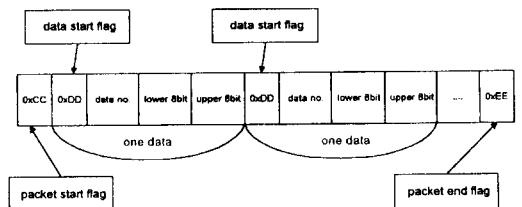


Fig. 4 Format of communication packet

있다. 여기에서 0xCC, 0xDD, 0xEE는 각각 패킷의 시작플래그, 데이터의 시작플래그, 패킷의 마지막 플래그를 의미하는 기호이다. 또한, 각 데이터들은 정수형으로 표현되어 있기 때문에 데이터의 번호, 하위 8비트, 상위 8비트 순으로 송수신되고 있다.

4. 통신 알고리즘의 구현

시스템의 통신 관련 소자로는 8259, 8253, 8251이 사용되고 있으며 이들 소자의 초기화가 필요하다. 이후 RCS측에서 먼저 BCS측에 송신을 시작하고 BCS측에서는 이 데이터를 인터럽트방식으로 수신하여 링버퍼에 저장함과 동시에 멀티태스킹 루틴에서 이 버퍼의 내용을 처리하도록 한다. BCS에서는 수신이 완료되면 RCS측에 송신을 시작하고, RCS에서는 BCS와 같은 방법으로 수신을 시작한다. 이렇게 해서 RCS와 BCS의 송수신이 완료되면, 이번에는 RCS와 SS간에 송수신을 수행한다.

통신을 수행할 때 어느 한쪽이 고장 등의 원인으로 송수신이 불가능한 경우 다른 쪽에서는 무한정

으로 대기할 수 없기 때문에 Fig. 5의 알고리즘에 의해 상대 시스템의 통신 불능상태를 확인하고 있다. 즉, 통신 플래그(comm_f)가 수신상태인 경우 0.1초 간격으로 count 변수를 1씩 증가시키고, 그림의 ①에서와 같이 0xDD가 수신되면 count 변수 및 miss 변수를 리셋시킨다. 만일 count가 10보다 크면 count 변수를 리셋시키고, miss 변수를 1 증가시킨다. 또한 miss 변수가 3보다 크면 3초동안 상대 시스템으로부터 송신데이터가 전송되지 않은 것을 의미하므로 상대 시스템을 고장으로 간주한다.

4.1 송신 알고리즘

Fig. 6의 comm_f=0, 즉 송신인 경우에는 우선 새롭게 수집된 송신해야 할 데이터값(SENDING[i])과 직전에 송신한 데이터값(SENDED[i])을 비교하여 서로 다른 것만을 송신하고 있다. 즉, ①의 송신해야 할 SENDING 배열 값 중에서 SENDING[i]=SENDED[i]인 배열값들을 제외하 나머지로써 ②와 같이 NEW_SEND 배열을 만든

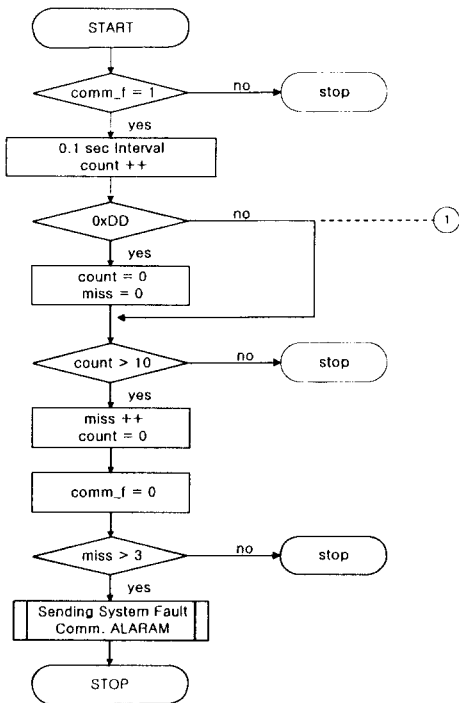


Fig. 5 Flow chart of communication failure

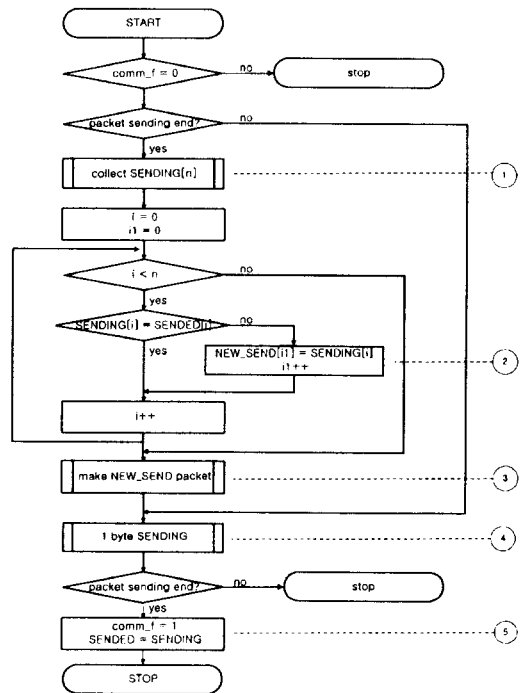


Fig. 6 Flow chart of transmission algorithm

다. ③에서는 NEW_SEND 배열값을 Fig. 4의 통신패킷 형식에 맞추어 패킷화 하며, ④에서는 바이트 단위로 송신한다. 8251소자의 경우 아래와 같이 상태레지스터(C8251)의 TxRDY값(0비트 번째)을 읽어 이 값이 1 이면 송신가능한 것으로 판단하여 1바이트의 데이터(ch)를 송신한다.

```
while ((inportb(C8251) & 0x01) != 0x01) {
    outportb(D8251, ch);
}
```

송신이 끝나면 ⑤에서와 같이 comm_f=1로 하여 자신을 수신상태로 하고, SENDING 배열값을 SENDEED 배열값에 복사한다.

4.2 인터럽트에 의한 수신 알고리즘

comm_f=1 이면 수신상태이므로 이때는 Fig. 7의 인터럽트 알고리즘에 의해 8251소자의 데이터 라인으로부터 값을 읽어 링버퍼에 저장한다. 여기에서 ①은 수신된 데이터 값을 링버퍼의 현재 장소에 저장하는 것을 의미하며, ②는 링의 마지막에 저장한 경우 다음 저장 장소를 새로운 가상 링으로 지정하고 no=0으로 하는 것을 의미한다.

4.3 링버퍼내의 수신데이터 처리알고리즘

링버퍼내의 수신된 데이터를 저장하고 처리하기 위해 4개의 변수가 이용되고 있다. 이 4개의 변수의 조합으로 부터 링버퍼내의 저장 장소와 처리 장소 사이에는 다음과 같은 4가지 경우가 존재한다.

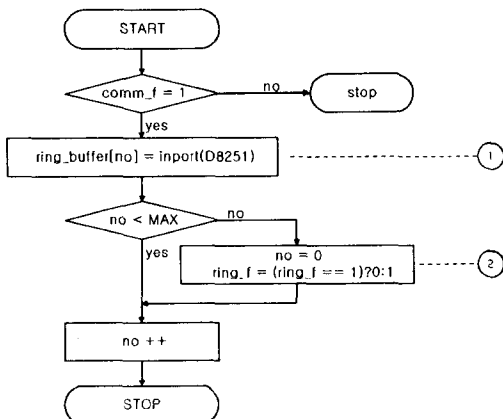


Fig. 7 Flow chart of receiving algorithm by interrupt

- ① (ring_f == ring_get_f) && (no < get_no)
- ② (ring_f == ring_get_f) && (no ≥ get_no)
- ③ (ring_f != ring_get_f) && (no ≥ get_no)
- ④ (ring_f != ring_get_f) && (no < get_no)

일반적으로 처리보다는 저장이 먼저 수행되어야 하기 때문에 ⑥와 ④는 정상적인 상태이지만, ③과 ②는 비정상적인 상태라고 할 수 있다. 여기에서 ③은 존재할 수 없는 상태로서 대상에서 제외하였으며, ②는 저장이 처리보다 아주 빠르게 진행되어 링 버퍼의 한 바퀴를 추월해버린 상황을 나타내는 비정상적인 상태의 경우이다. 이것을 정상적인 상태로 수정하기 위해서는 일부 수신된 데이터를 포기하여 인위적으로 ⑥ 또는 ④의 형태로 바꾼 후 처리하도록 할 필요가 있다.

본 논문에서는 ④와 ⑥를 동일 링으로 ③과 ②를 다른 링으로 분류하여, 동일 링의 경우에는 Fig. 8(a)와 같이, 다른 링의 경우에는 Fig. 8(b)와 같이 수신 데이터를 처리하도록 하고 있다.

Fig. 8(a)에서 ①은 위의 ④의 경우로서 곧바로 수신처리를 정지하기 위하여 수신 인터럽트를 불능상태로 하고 comm_f=0으로 하여 자신을 송신상태로 한다. ①의 아래쪽은 ⑥의 경우로서 정상적인 수신처리를 수행하는 부분이다. 여기에서 0xCC

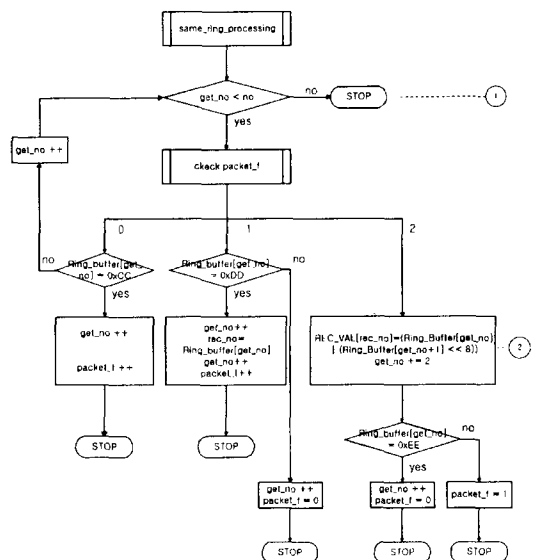


Fig. 8(a) Data process of same ring

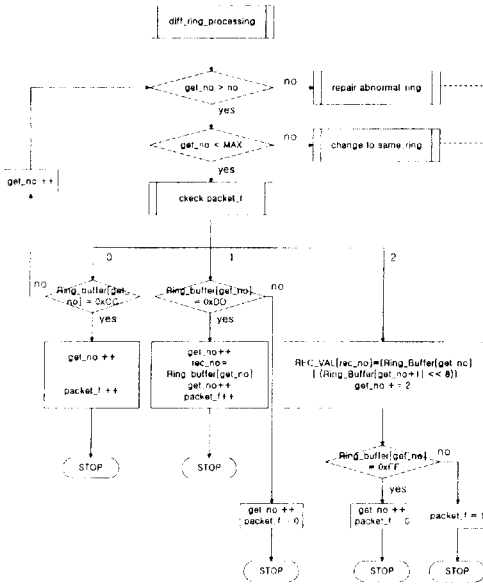


Fig. 8(b) Data Process of Different Ring

는 패킷 시작 플래그를 나타내는 값이므로 이 값이 나타날 때까지 get_no를 1씩 증가시켜 이 값이 나타나면 이번에는 0xDD 즉, 데이터 시작 플래그를 확인하고 ②에서와 같이 REC_VAL[rec_no]=(Ring_Buffer[get_no+1] | (Ring_Buffer[get_no+2]<<8))로 대입한다. 즉 Ring_Buffer[get_no+1]를 하위 8비트로, Ring_Buffer[get_no+2]을 상위 8비트로 하여 REC_VAL[rec_no]에 대입한다. 여기에서 REC_VAL 배열은 제어 알고리즘에서 내부적으로 이용하기 위한 데이터 변수이다.

Fig. 8(b)는 서로 다른 링을 처리하는 경우로서 ①은 앞의 ③의 경우에 해당하므로 링의 일부를 삭제하여야 한다.

본 논문에서는 수신된 데이터를 효율적으로 일부분만 삭제하기 위하여 Fig. 9에서와 같이 no값을 두 가지 경우로 분류하였다. 즉, 수신 장소가 링버퍼의 중간보다 작으면 ($no < (MAX \gg 1)$), 처리장소를 링버퍼의 중간에서부터 시작하여 0xCC가 처음 발견되는 장소로 옮기고, 수신 장소가 중간보다 큰 경우에는 처리 장소를 링버퍼의 시작점로부터 탐색하여 0xCC인 장소로 수정하였다. 이 경우에는 서로 다른 가상 링을 처리하다가 동일 링으로 처리 장소를 전환한 경우이므로 그림에서 ②의 경

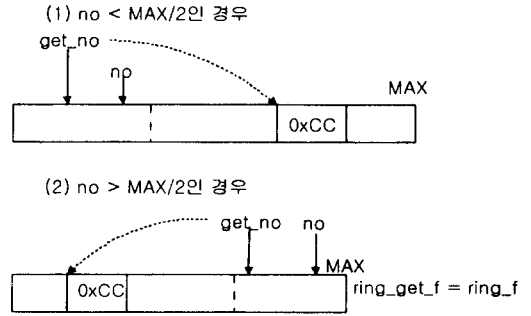


Fig. 9 Modification of abnormal ring

우에 해당하며 ring_get_f를 다음과 같이 수정해 줄 필요가 있다.

```
ring_get_f=(ring_get_f== 0) ? 1 : 0 ;
```

②이하의 앞의 ④의 경우로서 정상적인 수신처리를 수행하는 부분이며 이 경우는 Fig. 8(a)의 경우와 동일하다.

5. 검토 및 결론

본 논문에서는 선박 주기관 원격제어시스템을 3개의 서브 시스템으로 구성하고 이들 간의 통신방식에 대하여 논하였다. 서브 시스템들은 기능상으로 시리얼 통신외에도 정지, 시동 및 운전의 제어와 감시, 입력센서로부터의 계측, 계측된 값과 제어량들의 디스플레이, 각종 파라미터의 변경 기능 등을 제공하고 있다.^[12] 이 중에서 파라미터의 변경 기능은 필요할 때에만 실행되는 기능으로서 파라미터값을 EEPROM에 기억시키므로써 전원 차단시에도 보존되도록 하기 위한 것이지만, EEPROM의 하드웨어 동작 특성상 많은 액세스시간이 요구된다.

한편 링버퍼의 배열을 128개, 송수신데이터를 10개로 하고, 통신에러 검출시에 7 세스먼트에 출력하도록 하여 시스템을 실행시켜 본 결과, 파라미터 변경기능을 수행시키지 않았을 때에는 링의 비정상적인 상태에 한번도 도달하지 않았다. 파라미터 변경기능을 수행시켜 보았을 때에도 정상이었지만 일부러 EEPROM의 쓰기시간을 늘리기 위해 연속적으로 엔터키를 10회 이상 눌렀을 때 비정상

적인 상태에 1회 도달하여 링버퍼를 수정하였다. 그러나 일반적으로 이런 극한 상황은 거의 존재하지 않는다고 볼 때 시스템의 실시간 실행에 문제가 없을 것으로 판단되며, 필요에 따라서는 배열의 갯수를 더 늘리면 전혀 문제가 없을 것이다.

또한 각 서브시스템에는 동일한 기능을 수행하는 스위치들이 존재한다. 예를들어 SS와 BCS에는 Shut Down Cancel 스위치, Slow Down Cancel 스위치 등이 있으며, RCS와 BCS에는 제어위치 절환 스위치가 존재한다. 이들은 상대방의 스위치 값들을 통신을 경유하여 항시 감시하고 현재 상태를 유지시켜 준다. 본 논문에서는 통신속도를 9600, 4800, 2400, 1200, 600BPS로 변경시키면서 이 스위치값들을 계속적으로 변경시켜 본 결과가 값들이 통신속도에 큰 영향을 받지 않고 정상적으로 수정됨을 확인하였다.

일반적으로 많은 데이터의 전송시에는 CRC 에러검출 기법¹³⁾을 이용함으로써 데이터의 신뢰성을 확보하는 것이 보통이다. 그러나 이런 기법은 데이터의 변환과 부가적인 데이터의 전송을 수반하므로 그만큼 통신시간을 많이 소비하게 된다. 따라서 통신데이터 수가 적은 경우에는 비효율적이라고 할 수 있다. 본 논문에서는 이런 기법 대신에 통신패킷의 형식에 맞추어 0xCC, 0xDD, 0xEE값이 예상되는 수신 장소를 조사하여 이 값과 동일하면 통신데이터에 에러가 없다고 판단하고 그렇지 않을 경우에는 해당 패킷에 에러가 있다고 판단하는 방법을 도입하였다. 그러나 이 방법의 경우에는 에러가 발생하였을 때에 새롭게 찾아낸 패킷전까지의 데이터가 소멸된다.

어느 방법이 우수한가에 대한 명확한 비교는 어렵지만, 통신데이터 수가 적은 경우에는 일부 데이터가 소멸되어도 시스템의 실시간 성능에는 전혀 문제가 없기 때문에 본 논문의 방법이 우수할 것이며, 데이터의 수가 많아지면 패킷내의 하나의 데이터 에러 때문에 상당량의 데이터를 포기해야 하므로 실시간 성능에 문제가 발생할 수 있을 것이다. 따라서 많은 실험을 통하여 데이터 수에 따라 에러 검출을 선택적으로 수행하는 방법을 도입하는 것이 바람직할 것이다.

참고문헌

- [1] 박종우, 필드버스 기술동향, 1996. 6, 계측정보(제어현장의 새얼굴 필드버스 특집호).
- [2] 麥谷富浩, FA/組み組み用途における通信とは, インターフェース, CQ出版社, 1995. 10.
- [3] 笹嶋久, 制御系のライトサイジングを支えるネットワーク技術, 計測と制御, vol.34, No.11, pp.874-879, 1995.
- [4] 洪承鎬, 센서레벨네트워크 필드버스 기술 개요, 電子工學會誌, 第21卷, 第4號, pp.87-95, 1994.
- [5] 中川司, 船用コンピュータシステムと國際動向, 日本船舶機關學會誌, 第32卷, 第2號, pp.79-85, 1997.
- [6] 細川成通, 主機關の自動化の現状と將來動向, 日韓學術交流講演會, pp.2-11, 1993.
- [7] M-800B Bridge Maneuvering System Manual, NABCO.
- [8] Auto-chief 4 Main Engine Remote Control System Manual, NORCON.
- [9] 선박용 증소형 엔진 자동제어 시스템 개발, 정경열 외 9인, 통산산업부연구보고서, 1996
- [10] 宮崎誠一ら, 標準シリアルインターフェース, トランジスタ技術 Special No. 8, CQ出版社, 1992. 10.
- [11] 石川裕次, シリアル轉送用LSIの使い方, トランジスタ技術 Special No. 8, CQ出版社, 1992.10.
- [12] 주기관제어용 소프트웨어개발, 류길수 외 4인, 상공부위탁연구보고서, 1996. 7.
- [13] Gould Modbus Protocol Reference Guide, GOULD Electronics.

저 자 소 개



정경열 (鄭慶烈)

1960년 1월생, 1982년 한국해양대학교 기관학과 졸업, 1984년 동대학교 대학원 졸업(석사), 1996년 동대학원 박사과정수료, 1982년 - 1986년 기관사 승선근무, 1987년 - 현재 한국기계연구원 환경설비연구부 선임연구원, 당학회 중신회원



류길수 (柳吉洙)

1953년 5월생, 1976년 한국해양대학교 기관학과 졸업, 1979년 동대학교 대학원 졸업(석사), 1978년 - 1982년 기관사 승선근무, 1982년 - 현재 한국해양대학교 자동화·정보공학부 교수, 1984년 - 1986년 일본 동경공업대학 대학원 졸업(석사), 1986년 - 1989년 동 대학원 졸업(박사), 당학회 중신회원