

## 최적화된 큐 기반의 NMEA 프로토콜 멀티플렉서 시뮬레이션에 관한 연구

박시형<sup>\*</sup>, 정성훈<sup>\*</sup>, 김창수<sup>\*</sup>, 임창묵<sup>\*</sup>, 임재홍<sup>\*\*</sup>

<sup>\*</sup>한국해양대학교 대학원, <sup>\*\*</sup>한국해양대학교

### A Study of NMEA Protocol Multiplexer Simulation on the based optimizing Queue

Si-Hyoung Park<sup>\*</sup>, Sung-Hun Jung<sup>\*</sup>, Chang-Soo Kim<sup>\*</sup>, Chang-Mook Yim<sup>\*</sup>, Jae-Hong Yim<sup>\*\*</sup>

<sup>\*</sup>Graduate school of National Korea Maritime University, <sup>\*\*</sup>National Korea Maritime University

e-mail:goodclick@hotmail.com

#### 요 약

현재 국내에서는 NMEA 프로토콜 멀티플렉서를 개발한 사례 및 제품이 없어, 외국의 고비용 장비를 수입하여 사용하거나, 멀티 포트를 이용하여 NMEA 프로토콜을 전송하는 프로그램을 소프트웨어적으로 구현하여 사용하고 있다. 이러한 방법은 고비용이 지출되거나, 각 응용프로그램마다 별도의 처리 부분을 작성해야하는 문제점이 있다. 또한, NMEA 프로토콜을 출력하는 각각의 장비들은 제조 회사 및 플랫폼이 다르므로 이중의 자원낭비 및 손실 등을 초래할 수 있다. 본 논문에서는 이를 보완하기 위하여 신뢰성 있는 처리방법과 고성능의 단일 하드웨어 모듈로서 독립적으로 동작할 수 있고, 최적화된 큐의 설계를 이용하여 모듈의 메모리 효율을 높이며, 주요 입력 센서 장비인 자이로콤파스, 에코 사운드, GPS 등의 장비들과 실시간 통신의 높은 신뢰성을 유지할 수 있는 NMEA 프로토콜 멀티플렉서를 구현에 대하여 제안하였다.

#### ABSTRACT

Domestic use, or embody program that transmit NMEA protocol using multi port as software and is using because there is no fee and product that develop NMEA protocol Multiplexer, import mounting for foreign climax present. These method is paid or there is problem that must make out special processing part in each application program. Also, each mountings that display NMEA protocol can cause double resources waste and damage etc. because manufacturing firm and platform are different. Can act separately as single hardware module of reliable processing method and high efficiency to supplement this in this treatise, and because using design of optimized cue, heighten memory efficiency of module, and proposed NMEA protocol Multiplexer that can keep high trustability of Come on, deviation compass, echo sound, mountings of GPS and so on and real time communication that is main input sensor equipment about embodiment.

#### 키워드

NMEA0183, MULTIPLEXER, PROTOCOL, QUEUE, MULTI PORT, GPS, RS-232

## 1. 서 론

순항하는 선박과 항공기가 일정 항로에 따라 목표지점까지 가기 위해서는 여러 가지 제반 사항이 따른다. 그중에서 운송물을 빠르고 안전하게 이동하기 위해 현재의 위치 정보를 알려주는 GPS 수신기와 같은 장치가 필요할 것이며, 또한 자이로컴파스(Gyrocompass), 풍향풍속계(Anemometer), 스피드 로그(Speed Log), 타각지시기(Rudder) 등 해상 전자 장치들도 필요하다.

이러한 해상 전자 장치들 간의 통신을 위한 인터페이스의 표준을 정하는 목적으로 미국해상전자통신협회(NMEA : National Marine Electronics Association)라는 위원회가 발족하였다. NMEA는 제조업자들, 배급업자들, 도매업자들 그리고 교육기관들로 구성된 비영리적인 제휴이다. 해상의 기기 장치 사이에 통신을 위하여 전기적 신호에 관한 인터페이스와 데이터 프로토콜로 정의할 수 있으며, 해상전자 장치의 인터페이스에 널리 사용되고 있는 프로토콜이다.

NMEA 프로토콜을 사용하는 장치로부터 출력되는 신호를 처리하기 위해서 장비의 기본 포트를 이용하며, 부족할 경우 멀티포트를 사용하여 처리하고 있다. 그러나 수많은 신호에 대한 멀티포트의 적용과 전용장치로서의 출력의 제약 사항들을 해결하기 위하여 NMEA 관련 신호를 멀티플렉싱하여 제공할 수 있는 모듈 개발이 필요하다.

현재 국내에서는 NMEA 프로토콜 멀티플렉서를 개발한 사례 및 제품이 없어, 외국의 고비용 장비를 수입하여 사용하거나, 멀티 포트를 이용하여 NMEA 프로토콜을 전송하는 프로그램을 소프트웨어적으로 구현하여 사용하고 있다. 이러한 방법은 고비용이 지출되거나, 각 응용프로그램마다 별도의 처리 부분을 작성해야하는 문제점이 있다. 또한, NMEA 프로토콜을 출력하는 각각의 장비들은 제조회사 및 플랫폼이 다르므로 이종의 자원낭비 및 손실 등을 초래할 수 있다. 본 논문에서는 이를 보완하기 위하여 신뢰성 있는 처리방법과 고성능의 단일 하드웨어 모듈로서 독립적으로 동작할 수 있고, 최적화된 큐의 설계를 이용하여 모듈의 메모리 효율을 높이며, 중요 입력 센서 장비인 자이로컴파스, 에코 사운드, GPS 등의 장비들과 실시간 통신의 높은 신뢰성을 유지할 수 있는 NMEA 프로토콜 멀티플렉서를 구현에 대하여 제안하였다.

## II. 관련 연구

### 2.1. NMEA 신호종류

NMEA 신호는 NMEA-0180, 0182, 0183 등의 종류가 있으며, 현재 국제 표준으로 쓰이는 NMEA-0183을 중심으로 알아 볼 것이다. 우선

NMEA-0180과 NMEA-0182의 차이점은 단순 데이터 포맷인가 아님 복합 데이터 포맷인가 하는 차이점뿐이다.

단순 데이터 포맷은 7비트로 비트 0에서 비트 5까지는 크로스 트랙 에러를 나타내며, 비트 6에서의 값이 1이면은 데이터가 유효함을 나타내는 것이고, 비트 7에서의 값 0은 단순 데이터 포맷을 나타내는 것이다.

복합 데이터 포맷은 아스키코드로 나타내어진 크로스 트랙 에러와 방위각, 현재의 위도/경도 및 2진 상태 바이트로 구성된 37바이트의 데이터 블록으로 이루어져 있다. 복합 데이터 포맷에서의 모든 바이트는 비트 7의 값이 1로써 단순 데이터 포맷형식과 구별이 되어진다. 그리고 단순 데이터와 복합 데이터를 동시에 전송이 가능하며, 복합 데이터 블록 중간에 단순 데이터를 삽입도 가능하다.

그리고, NMEA-0183은 데이터 전송을 ANSI 표준에 따라서 직렬 비동기 방식으로 전송을 한다. 아래 그림에 나타난 것처럼 첫 번째 비트는 Start Bit이며, 다음에 따라 오는 비트는 Data Bit 이며, 마지막 비트는 Stop Bit로 이루어져 있다.

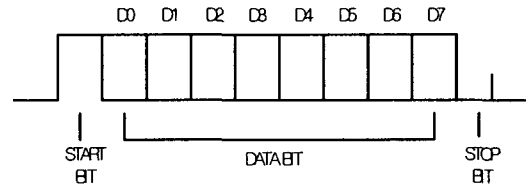


그림1. NMEA-0183 데이터 전송

### 2.2. NMEA 신호형식

NMEA-0183은 ASCII 코드의 모든 문자를 사용하며, 데이터는 문장형태로 전송이 되어진다. 각 문장은 "\$"로 시작을 하며, 송신자 ID를 나타내는 두 개의 문자와 문자 3개로 표현이 되는 데이터 ID, 콤마로 구분된 데이터 필드가 따라오고 추가적으로 체크섬과 캐리어 리턴/라인피드로 종결이 된다. 그리고,, 한 문장은 \$와 CR/LF를 포함해서 82개의 문자로 구성되어 있다.

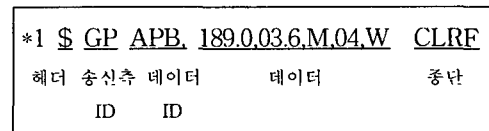


그림2. NMEA-0183 출력 형식

만약에 필드의 데이터가 유효하지 않으면 해당 필드를 생략하거나 콤마로 범위를 정하여 전송을 하며, 이때 문자 사이에 빈 공간은 없어야 한다. 체크섬은

“\*”와 두 자의 16진수로 이뤄지는데, “\$”와 “\*”를 포함하지 않는 모든 글자의 벤티적-or의 결과이다.

다음 그림2는 일반적인 NMEA-0183 출력의 예이다.

### III. NMEA 멀티플렉서 설계 및 구현

#### 3.1. 시스템의 구성 및 기능 정의

그림4의 시스템 구성도는 GPS 및 각종 Node장치들로부터 NMEA 데이터를 받아들여 Laptop Computer에서 데이터를 확인 하는 것을 보여주고 있다.

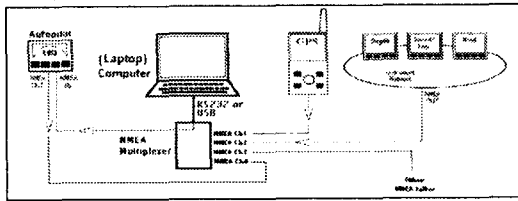


그림3. 실제 시스템 구성도

이때 Node를 구성하는 장비들은 수심 측정기, 알파 레이더, 전자해도표시시스템(ECDIS)들이 되고 위의 그림5 에서는 이들 장비들의 데이터를 가진 PC들로 대신 하기로 한다. 또한 위 그림5의 점선에 해당하는 부분이 이 논문에서 핵심에 해당하는 NMEA 멀티플렉서를 가상적으로 멀티포트와 PC로 구성한 것이다. 이때 모니터링 PC에서는 각 Node들로부터 올라오는 데이터가 정상적으로 Laptop Computer에 전송되는지 확인할 필요가 있다.

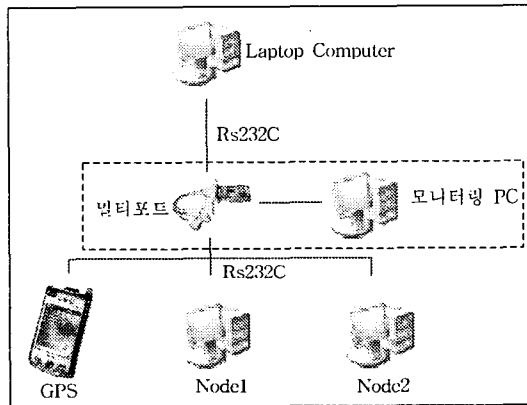


그림4. 가상 시스템 구성도

#### 3.2. 프레임 처리 기법

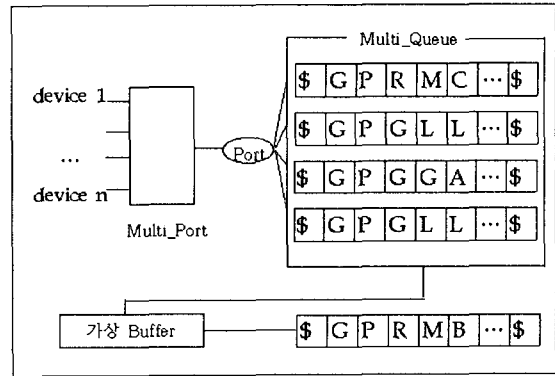


그림6. 멀티 대기큐의 구조

Multi-Queue 구조는 선입선출(FIFO)방식으로 운영된다. 신호가 Buffer로 입력 될 때, 효율적인 메모리 관리를 위하여 버퍼의 길이는 가변적인 크기를 가지게 된다. NMEA 멀티플렉서의 특성상 device로부터 오는 데이터 중 최근의 정보가 가장 중요하므로, 최근의 정보가 계속 갱신 되도록 구성되어 있으며, '\$'구분자 수를 판별하여 정상적인 데이터인지 잘못된 데이터 인지를 판별하도록 구성되어 있다.

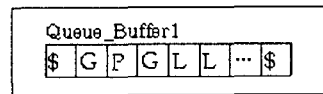


그림7. 큐 버퍼

#### 3.3. NMEA 멀티플렉서 구현

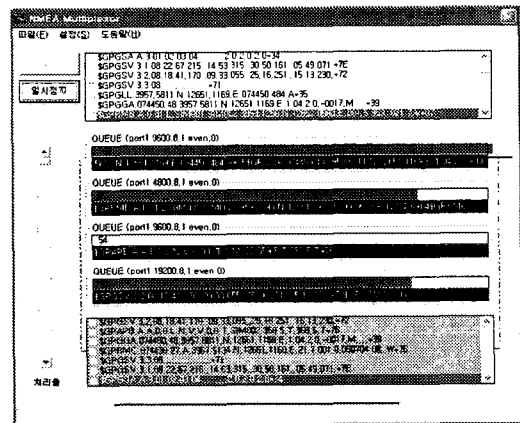


그림8. 모니터링 PC의 메인 폼

메인 폼의 화면 구성은 그림8과 같다. 각각의 장치 Computer들이 보내온 데이터들을 에러 없이 정상적으로 처리하고 있는지, 들어온 데이터들

이 어떤 내용인지를 확인하고 가장 효율적인 처리를 할 수 있는 처리율을 찾아보도록 설계되어 있는 폼구조이다.

본 실험에서는 4개의 대기큐를 사용하여 배치하였다. 대기큐의 처리량에서 대기큐 버퍼로부터 정상 프레임이 발견되었을 때, 파서에 의해서 파싱된 문자열을 버퍼에 연속적으로 추가하여 나타낸 모습이다. 밑의 텍스트 박스는 실제로 채워지는 텍스트 스트림이다. 또한 그림8의 빨강 실선은 특정 대기큐로부터 프레임이 검출 되었을 때를 표현하고 있다. 이 실험에서는 처리량에 대한 최적화 처리를 프로그래스바로 버퍼 사이즈 변화량을 확인할 수 있다.

대기큐에 버퍼링되는 스트림이 정상 처리 안 될 때, 정상프레임을 파싱할 수 있는 상태, 버퍼의 크기보다 프레임이 커서 트렁케이션 현상(잘림 현상)을 나타내고 있다. 정상 처리 안 될 때는 Shape가 노랑색으로, 정상처리 될 경우는 Shape가 녹색으로, 트렁케이션 현상이 발생했을 때는 Shape가 빨강색으로 표현되어 있다.

#### IV. 대기 큐 알고리즘의 구현

##### 4.1. 큐 설계

각각의 장치들로부터 들어온 데이터를 입력포트로 받아들이며 대기큐를 활성화 시키고, 그 대기큐에서 프레임을 얻어온 후 각 포트에 해당하는 버퍼에 버퍼링 시켜주는 과정을 그림10에서 블록 다이어그램으로 보여주고 있다.



그림10. 큐 블록 다이어그램

그림11은 대기큐에서 프레임을 얻어오는 과정을 흐름도(DFD)로 보여주고 있다.

우선 입력포트에서 대기큐로 전송 활성화하는 소스를 보면 NMEA 멀티포터에서 입력된 데이터 중 리스트 박스에서 한 줄 얻어온다. 그런 다음 GPS, Echo Sounder, Gyro Compass, Radar등과 같은 데이터 프레임들을 종류별로 버퍼링한다. 각각의 Case문 0번부터 3번까지는 각 포트들로부터 입력될 큐4개를 나타낸다. 이 Case문에서는 데이터 프레임들이 한번에 전송될 때, 전송한 프레임

의 양(버퍼 크기)을 대기큐에 뿌려주는 양을 체크하기 위해 코딩된 부분이다. Case문을 빠져 나오면 마지막으로 활성화 큐의 버퍼 양을 체크하고 시뮬레이션 한다.

대기큐에서 프레임을 얻어오는 소스를 살펴보면, 우선 각각의 프레임 구분자인 '\$'의 개수를 정의하는 'EA'배열과 조사의 위치를 위한 변수를 정의한다.

그 다음으로 들어온 데이터 프레임 중 완전한 프레임이 있는지 조사하기 위하여 For문을 사용하고 각 대기큐별 '\$'의 개수를 모두 조사한다. 유효한 프레임이 있는지 조사하여 각 버퍼의 색상을 결정하여준 뒤 유효한 프레임을 파싱하여 출력하고 출력 리스트박스 하단에 보여주는 소스구조로 이루어져 있다.

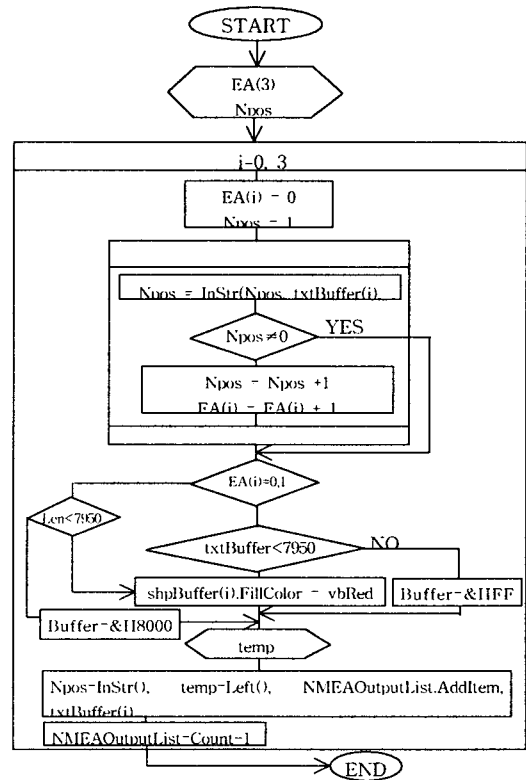


그림11. 대기큐로부터 프레임을 얻는 흐름도

대기큐에서 프레임을 얻어오는 소스를 살펴보면, 우선 각각의 프레임 구분자인 '\$'의 개수를 정의하는 'EA'배열과 조사의 위치를 위한 변수를 정의한다. 그 다음으로 들어온 데이터 프레임 중 완전한 프레임이 있는지 조사하기 위하여 For문을 사용하고 각 대기큐별 '\$'의 개수를 모두 조사한다. 유효한 프레임이 있는지 조사하여 각 버퍼의 색상을 결정하여준 뒤 유효한 프레임을 파싱하여 출력하고 출력 리스트박스 하단에 보여주는 소스구조로 이루어져 있다.

#### 4.2 결과 및 고찰

아래 그림13은 본 논문에서 제안한 큐 설계 기반으로 처리율, 프레임 손실률 및 전송시간과의 관계를 보여주는 그래프이다. 처리율을 높여준다는 말은 큐상의 버퍼 크기를 늘려준다는 말과 같다. 버퍼의 크기를 크게 즉, 처리율을 높여 줄수록 프레임 손실률 없이 들어오는 데이터 신뢰성은 높지만, 전송시간이 길어진다는 단점이 있다.

또한 장치들로부터 들어오는 데이터프레임의 크기보다 큐상의 버퍼를 크게 한다면 낭비되어지는 공간이 발생하고 이는 많은 메모리를 낭비한다는 것을 의미한다.

그렇다고 무작정 처리율을 줄여 준다면 이는 버퍼의 크기가 줄어들게 되기 때문에 버퍼의 크기보다 큰 데이터 프레임들은 다 버려지는 문제가 발생한다.

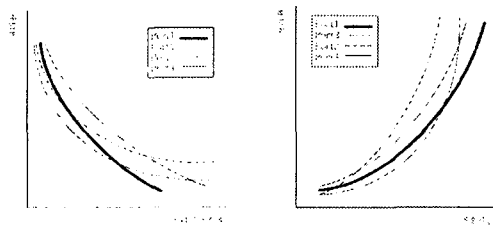


그림13. 처리율 변화에 따른 프레임 손실률과 전송시간

#### IV. 결론 및 향후 연구 방향

해상에서 작업하는 장비들이 다양해지고 더욱 많은 센서 설비들로부터 입력되는 신호들을 처리하기 위하여 수용하는 장비들의 경우 많은 수의 포트를 지원해야 한다. 그러나 여건상 제한적인 포트를 지원할 경우 수많은 장비들을 연결하기 위하여 별도의 멀티 포트의 기능을 수행하는 장치가 필요하다.

본 연구에서는 이러한 상황에서 입력되는 신호의 대역폭에 비례하여 수용하는 장치의 병목현상을 방지하고 최적의 신호처리를 수행할 수 있도록 하기 위한 대기-큐 기법의 연구와 버퍼링의 방법론들을 연구하고, 입력되는 다수 개의 포트로부터 신호를 처리하여 단일 또는 다수개의 포트에 출력시켜줄 수 있는 시스템에 대한 시뮬레이션을 통하여 최적의 요건을 만족하는 경우 이상적인 출력을 기대할 수가 있었다.

NMEA 멀티플렉서가 임베디드 환경에서 개발될 경우 제한적인 메모리의 양에 따라 대기큐의 운영 기법이 최적화 될 수 있도록 하기 위해서는 가변적인 큐의 운영 방식이 필수적이며, 처리량의 분산 기법을 통하여 최적의 메모리 운영 계획을 세울 수 있었다.

본 연구에서는 4개의 멀티 큐와 가상 큐를 설계하였고, 큐의 운영 알고리즘의 기법 적용과 가변 크기의 메모리를 동적으로 할당하여 적은 자원 환경에서도 다수개의 포트로부터 입력되는 신호들을 최소의 손실을 유지하고 단일의 포트에 신호를 통합하여 출력하여 주는 시스템의 시뮬레이션을 연구하였다.

향후 계획으로 소프트웨어적인 설계를 기반으로 하는 지금의 연구에서 임베디드 보드를 설계하여 실제 현장에서 사용될 수 있도록 모듈을 구현하고, 본 실험에서 얻어진 최적의 메모리를 활용한 설계와 큐 운영 기법의 적용으로 저비용, 고효율의 NMEA 멀티플렉서 장비를 개발해보고자 한다.

#### 참고문헌

- [1] 김익수, "디지털 공학", 대림 출판사, 1998.
- [2] 대한전자공학회, "디지털회로 및 시스템 실험", 청문원 출판사, 1998.
- [3] 이석호, "C로 쓴 자료구조론", 회중당, 1996.
- [4] 오용철, "자료구조", 이프레스, 2003.
- [5] NMEA0183 Version 2.00, "STANDARD FOR INTERFACING MARINE ELECTRONIC DEVICES".
- [6] 가남사 편집부, "RSC232C 인터페이스 사용법", 가남사, 1990.
- [7] SiRF Technology, Inc. "NMEA Reference Manual", pp.1-5, 2002.