「經營情報硏究」第2號
大韓經營情報學會
1998年 6月


# CAN을 이용한 자동차용 Network 구현

허 화 라*

〈목 차〉



## Ⅰ. 서 론

정보통신 기술의 급진적인 발달로 오늘날을 정보화 사회라 하며, 정보화를 구축하기 위한 다양한 연구 부분들이 발전되고 있으며 개발된 정보화 기술이 인간의 생활에 편의성을 제공하기 위해 정보를 보다 효율적으로 처리하기 위한 통신 시스템도 그 종류와 수가 빠른 속도로 증가하고 있어 멀지 않은 미래에 가상도시에서나 접할 수 있는 환경이 실세계(real world)로 전개될 것이며, 그 중의 하나가 미래의 자동차를 위한 통신기술이다.

초기의 자동차산업은 단순히 동력을 발생시켜 이것을 운동으로 변환하여 구동장치에 전달하는 기계적 기술에서 머물렀다. 이후, 1950년대 말과 1960년대 초에 이르러 상용차에 처음으로 기계적 메카니즘에 전자적인 기술이 적용되었으나, 1980년대 초반까지는 추가비용에 따른 경제적 이유로 수요자들에게 큰 호응을 얻지 못하여 기술 개발의 속도가 느렸다. 1980년대에 들어서 세계적인 경기회복에 힘입어 자동차산업이 산업의 중추

*동명대학 경영정보과 겸임전임강사

적 역할을 담당하면서 자동차에 관련된 많은 부분의 기술들이 개발되어 기계적인 제어 시스템들이 전자화 되었고, 최근에 이르러 수요자의 경향이 자동차의 안정성과 편의성에 관심을 가짐으로 인하여 기술 개발의 목표가 시스템의 오동작에 대한 진단 및 예방 기능, 주행시의 안정성 확보, 실시간 제어 기술 등으로 발전되어 자동차 생산국의 기술 경쟁이 가속화되면서 차량 내의 통신시스템 개발이 연구과제로 대두되었다.

미국의 컨설팅회사인 쿠퍼스&리브랜드사는 세계 자동차산업의 연간 생산대수가 1997년에 7,030만대에서 2002년에는 7,930만대로 전망하였으며, 한국의 경우 1998년 2월말 자동차 등록대수가 1,041만대, 1999년의 생산 예상량이 466만대로 발표되었다. 이러한 측면에서 볼 때 향후 자동차산업은 지속적으로 발전할 것이며, 더불어 자동차와 관련된 기술도 급속도로 발달하여 내부의 제어 시스템들이 network화된 항공기 시스템을 모델로 하는 저가의 고기능 자동차 시스템이 개발될 것이다. 따라서, 각 시스템들을 효율적으로 운용하기 위하여 network를 통한 실시간 제어시스템이 필요하고, 이를 위해서는 차량내의 모든 시스템을 network화하는 통신시스템이 요구된다.

본 연구에서는 이러한 차량용 통신시스템을 위한 최적의 통신 protocol인 CAN의 성능평가를 하고 CAN이 내장된 Intel사의 Micro-controller(87C196CA)를 사용하여 모듈을 설계하였다.

## Ⅱ. CAN(Controller Area Network) Protocol

CAN Protocol은 일정 지역내의 각종 신호들을 디지털로 변환한 형태에 신호의 종류나 신호를 필요로 하는 기구에 따른 식별코드를 부착하여 여러 기구가 공유하는 하나의 전송매체를 통하여 전송하는 방식이다. 즉, 차량에 지역 통신망(Local Area Network, LAN)과 유사한 통신망을 설치하고 차량의 주요 부분에 통신 스테이션(station) 또는 모듈(module)을 설치하여 여러 전장시스템을 연결한다. 각 스테이션은 접속된 전장시스템의 통신기능을 대행하며 여러 종류의 센서, 액츄에이터, 제어기뿐만 아니라 운전자가 조작할 수 있는 스위치와 차량 여러 곳에 설치된 램프와 같은 전기적인 부하들도 연결되어 차량의 상태에 대한 모든 정보를 어디에서라도 획득할 수 있기 때문에 보다 진보한 기능을 제공할 수 있는 장점이 있다.

## 1. CAN의 구조

CAN은 5 Kbps(kilobits per second)에서 1 Mbps까지 다양한 전송속도를 제공하며, 토폴로지로는 버스형(bus structure)과 스타형(star structure)을 지원한다. 링형 토폴로지는 한 스테이션의 고장이 전체 시스템에 영향을 주어 신뢰성이 떨어지는 반면, 버스형이나 스타형 토폴로지는 한 스테이션에서 고장이 발생하더라도 시스템의 일부는 사용할 수 있기 때문에 고장의 파급효과를 최소로 한정시킬 수 있다.

CAN의 계층 구조는 물리(physical) 계층, 트랜스퍼(transfer) 계층, 오브젝트(object) 계층의 3계층으로 나뉘어진다. 물리 계층은 신호들이 실제적으로 어떻게 전송되는지를 정의하는 층으로서 신호 레벨과 비트(bit) 표현방법, 전송매체 등을 정의하게 된다. 트랜스퍼 계층은 CAN 프로토콜의 핵심 부분으로서 받은 메시지를 오브젝트 계층에 보내고 오브젝트 계층으로부터 전송되는 메시지를 받아들이는 일을 수행한다. 그 세부 사항으로서는 고장 제한(fault confinement) 기능, 에러 감지 및 시그널링, 메시지 확인, 메시지를 올바르게 수신했을 때 보내는 억크날리지먼트(acknowledgement)기능, 메시지 충돌 중재 기능, 메시지 프레임화, 전송률과 타이밍 정의 등이다. 오브젝트 계층은 어떤 메시지가 전송되는지를 찾고, 트랜스퍼 계층으로부터 받은 메시지가 실제로 어떻게 사용되는지를 결정하며, 하드웨어와 관련된 응용 계층에 인터페이스를 제공하는 일을 하게 된다.

## 2. Content-based addressing과 bitwise arbitration

CAN은 송신 스테이션과 수신 스테이션의 어드레스 대신 메시지의 내용에 따라 ID(identifier)를 부여하여 모든 메시지를 구별하고 메시지의 우선권을 정하는 내용에 의한 주소지정(content-based addressing)을 한다. 즉, 어느 한 스테이션이 메시지를 전송하기 시작하면 나머지 스테이션들은 수신 상태가 되어 메시지를 받게 된다. 이때, 수신 상태의 스테이션들은 수신된 메시지의 ID를 판독하여 수신된 데이터가 그들과 관련이 있는지 없는지를 확인하여 관련이 있으면 받아들이고 관련이 없으면 무시하는 메카니즘이다.

또한, 둘 이상의 스테이션에서 거의 동시에 전송을 시도했을 경우, 충돌한 메시지의 ID를 한 비트씩 비교하여 수치적으로 가장 낮은 ID 값을 가진 메시지(가장 높은 우선순위를 가진 메시지)만 전송을 계속하며, 나머지 메시지는 즉시 전송을 중단하게 된다. 이

같은 ID에 의한 비트 비교 방식(identifier-based bitwise arbitration)은 정보와 시간의 손실이 발생하는 것을 방지하기 때문에 비파괴적 버스 액세스(non-destructive bus access)라고 불리며 파괴적 버스 액세스의 대표적인 예인 CSMA/CD(Carrier Sense Multiple Access with Collision Detection)와는 달리 네트워크의 부하가 상당히 증가하여도 효율적인 통신이 가능하다.

### 3. 주소지정(addressing) 방식과 메시지 우선순위(message priorities)

내용에 의한 주소지정 방식은 어떤 한 스테이션에서 메시지를 보내고자 할 때, 데이터(또는 전송요청)와 ID를 함께 보내게 되는데, 이때 다른 모든 스테이션들은 수신 상태에서 대기하다가 메시지를 수신하면, ID로써 그 내용이 자신과 관련성이 있는가를 체크하여 관련성이 있으면 받아들여서 저장하고 그렇지 않으면 무시(message filtering)하는 기법이다. 내용에 의한 주소지정 방식은 현재의 시스템에 새로운 스테이션이 첨가되거나, 삭제되더라도 소프트웨어나 하드웨어 상의 재구성이 필요하지 않으므로 유연성있는 시스템 구축을 가능하게 한다. 또한, 버스를 할당함에 있어서도 ID를 사용하여 메시지 내용에 따라 기능적으로 중요한 메시지에 높은 우선순위를 부여한다.

어느 한 스테이션이 전송할 메시지가 있을 때, 먼저 버스의 상태를 살핀 후 버스가 'idle'하면 메시지를 전송하게 되고, 버스의 상태가 'busy'하면 수신모드로 들어가게 된다. 메시지를 전송하는 도중 충돌이 발생하게 되면 데이터 프레임의 중재 필드에서 ID영역을 비교하여 우선순위가 높은 메시지는 계속 전송하고, 우선순위가 낮은 메시지는 전송을 중단하고 수신모드로 들어갔다가 재전송을 시도하게 된다. 또, 수신모드에 있는 스테이션은 전송되는 메시지의 ID를 체크하여 관련이 있으면 받아들이고 관련이 없으면 무시하게 된다.

### 4. 에러 처리 방법

CAN은 전송 에러를 인식하고 재전송에 의해 자동적으로 에러를 수정해 준다. 에러 타입에는 비트 에러, 스터프(stuff) 에러, CRC 에러, 형식 에러, 그리고 억크날리지먼트 에러의 5가지가 있다

에러 감지 방법은 순환중복검사, 모니터링(monitoring), 비트 스터핑, 메시지 프레임 체크 등이 있다. 순환중복검사 방법은 전송시 그 전송값을 생성 다항식으로 나누고, 그 값을 메시지 뒤에 붙여서 함께 전송한다. 그러면 수신 측에서는 받은 데이터를 동일한 생성 다항식으로 나누어 나머지가 '0'이면 에러가 없는 것으로 가정하는 방법이다. 모니터링 방법은 비트를 전송하는 스테이션이 동시에 버스 레벨(bus level)을 체크하여 전송한 비트와 버스 상에서 감지된 비트를 비교하는 방법이다. 이때, 스테이션에서 전송한 비트와 다른 비트가 버스 상에서 감지되면 에러 플래그를 보내게 되고 이러한 방법으로 비트 에러를 감지한다. 스터프 에러는 SOF와 CRC 필드의 끝 사이에서 같은 극성을 가진 비트가 6개 이상 연속으로 나타나면 비트 스터핑이 적절히 수행되지 못한 것으로 간주하고, 스터프 에러 상태가 된다. 메시지 프레임 체크 방법은 CAN 프로토콜에 포함되어 있는 고정 포맷 비트 필드를 체크하여 메시지 프레임 형식에 어긋나는가의 여부를 알아내는 방법으로, 형식에러를 체크한다.

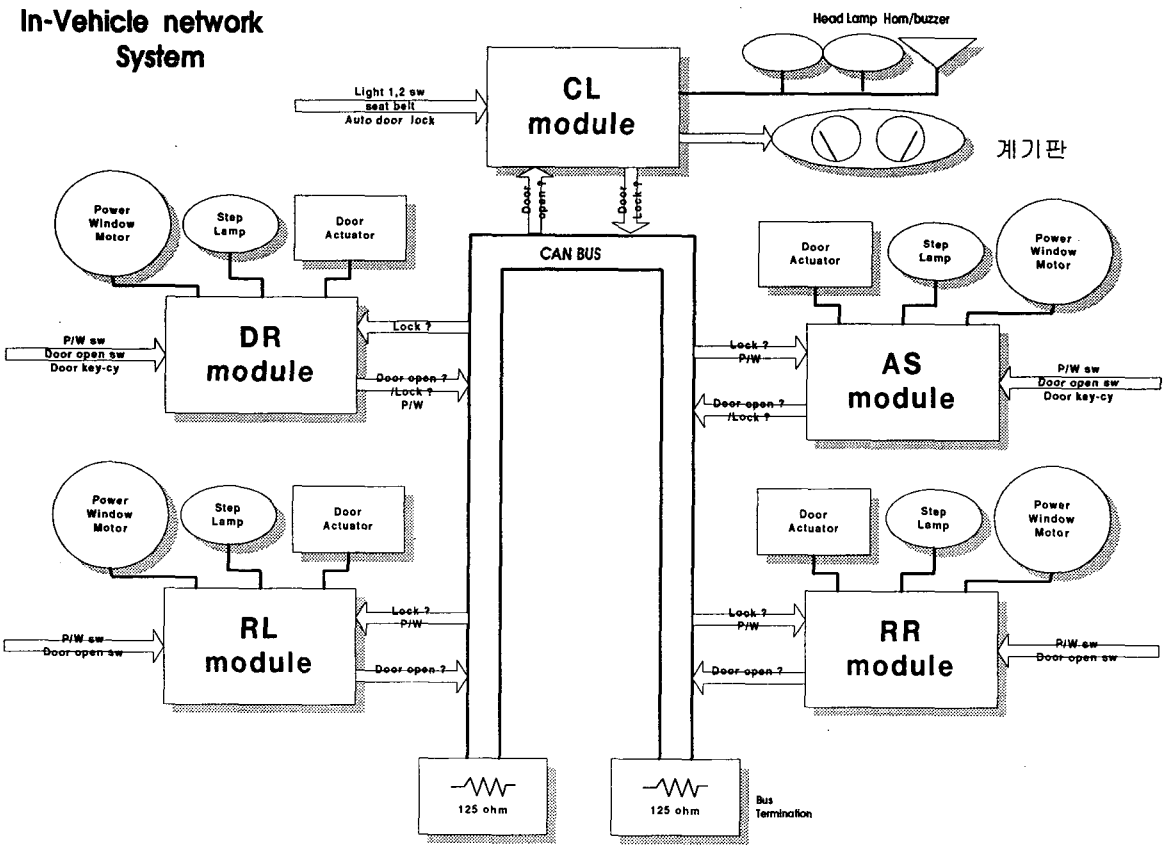
에러의 발생을 알리는 방법으로 에러를 감지한 스테이션에서 에러 플래그를 전송하게 되는데, 비트 에러, 스터프 에러, 형식 에러, ACK 에러의 경우에는 에러를 감지한 비트의 바로 다음에 전송하고, CRC 에러의 경우에는 ACK 딜리미터 바로 다음에 에러 플래그를 전송한다.

이 에러 플래그는 비트 스터핑 법칙이나 프레임의 고정된 형식을 깨뜨리게 되고, 이것이 다시 다른 스테이션으로 하여금 에러 플래그를 전송하게 한다. 예를 들어, 임의의 스테이션 A가 데이터 프레임을 수신하다가 에러를 감지하면 A는 6개의 'd' 비트(에러 플래그)와 8개의 'r' 비트(에러 딜리미터)로 구성된 에러 프레임(액티브 에러 플레그)을 전송하게 된다. 이 에러 프레임을 받은 다른 스테이션들은 6개의 'd' 비트를 수신하여 에러가 발생했음을 인식한다. 그래서 이들 스테이션들도 또 6개의 'd' 비트(에러 플래그)와 8개의 'r' 비트(에러 딜리미터)로 구성된 에러 프레임을 전송하게 되고, A가 전송한 에러 프레임의 에러 딜리미터 영역에 6개의 'd' 비트를 오버라이트(overwrite)하여 에러 플래그의 중첩 영역(6~12 bits)을 만들게 된다. 또, CAN에는 전송 에러 카운터와 수신 에러 카운터가 있는데, 메시지가 올바로 송수신된 경우에는 에러 카운터가 감소하고 에러가 발생한 경우에는 에러 카운터가 증가한다. 만일 이 두 에러 카운터가 127 에러 포인트를 넘지 않으면 에러 액티브(error active) 상태로 동작하고, 두 에러 카운터 중 어느 하나가 127 에러 포인트를 넘으면 에러 패시브(error passive) 상태로 바뀌게 된다. 한

편, 두 255 에러 포인트를 초과한 전송 에러는 시스템을 'bus off' 상태로 바뀌게 한다.

## Ⅲ. 모듈의 설계

본 연구에서는 다섯 개의 모듈을 구성하고 이들 모듈을 Network으로 통합하였다. 다섯 개의 모듈을 CL(cluster) DR(dirver), AS(assistant), RR(rear right), RL(rear. left)로 정의하고, 〈그림 1〉에서 나타난 바와 같이 CL 모듈은 계기판에 나머지 네 모듈들은 네 개의 도어에 각각 설치하였으며, 각 역할에 따라 입·출력 신호들을 구분하고 기능에 맞도록 설계되었다. 모듈의 형태는 기능에 따라 크게 두 가지로 나뉘어지는데 CL 모듈과 나머지 모듈, 즉 DR, AS, RR, RL 모듈들로 구분되어진다.




〈그림 1〉 전체 NetWork System 주요구성도

## 1. 모듈의 설계

각각의 기능을 정의하고 입·출력 포트를 구분하여 정리하였다. Test Bed에서 I/O로 사용 가능한 포트는 PSD 302 Port A와 87C196CA의 Port 1,6 이다. 회로의 설계는 입·출력 신호상 DR 모듈과 CL, AS, RR, RL 모듈로 구분되고, 기능상 CL 모듈과 나머지 모듈로 구분된다. 여기서 나타낸 표기 '.number'은 지정된 포트의 Bit number를 의미한다(예, Port 6.3 87C196CA의 Port6의 3번 bit).

### 1.1 CL(cluster)모듈

〈표 1〉 입력 및 출력신호에 따른 포트할당(CL 모듈)

| 입력 (PSD 302 Port) | | 출력 (87C196CA Port) | |
|---|---|---|---|
| | | Head Lamp | Port 6.0 |
| Light 1단 | PortA .1 | Tail Lamp | Port 6.1 |
| Light 2단 | PortA .0 | Seat belt Lamp | Port 1.3 |
| Seat belt SW | PortA .2 | Horn Relay | Port 1.2 |
| Auto door lock | PortA .3 | Buzzer Relay | Port 1.1 |
| | | Room Lamp | Port 1.0 |

CL 모듈은 전체적인 시스템관리 및 계기판과 관련된 입·출력을 담당하는 모듈이다. CL 모듈의 신호는 주로 운전자의 리모콘 신호와 multi-function switch에 의한 신호이며 출력 신호는 각종 경고등 및 실내외 등을 점등하거나 door lock을 해제하는 신호이다. 각 신호의 입·출력포트의 할당은 〈표 1〉과 같다.

### 1.2 DR(driver) 모듈

DR 모듈은 운전석에 부착되는 모듈로서, DR 모듈에서는 입력신호가 DR 모듈과 관련된 키 입력 외에도 다른 모듈(AS, RR, RL)을 제어하는 키 입력이 있다. 따라서 입력포트가 다른 모듈에 비해 비교적 많이 할당되어 구성되었는데 각각의 키 입력 및 출력에 따른 포트 할당은 〈표 2〉와 같다.

〈표 2〉 입력 및 출력신호에 따른 포트할당 (DR 모듈)

| 입력 | | 출력 (87C196CA Port) | |
|---|---|---|---|
| DR 모듈 창 | Port 1.0(↑) 1.1(↓) | | |
| AS 모듈 창 | PortA 0(↑) 1(↓) | P/W motor | Port 6.4 (↑) |
| RR 모듈 창 | PortA 4(↑) 5(↓) | | Port 6.5 (↓) |
| RL 모듈 창 | PortA 6(↑) 6(↓) | Door Actuator | Port 6.0 |
| 문 열림 검출 | PortA 3 | Step Lamp | Port 6.1 |
| 문 잠금 신호 | PortA 2 | | |

(↑) : P/W (Power Window) motor up (↓) : P/W motor down

DR 모듈에서는 다른 모듈의 창까지도 제어해야 하므로 그에 따르는 입력이 많다. 다른 모듈의 창은 Network을 통하여 명령이 전달되어 제어된다.

### 1.3 AS(assistant), RR(rear right), RL(rear left) 모듈

AS, RR, RL 모듈은 각각 조수석, 오른쪽, 왼쪽 뒷 좌석에 부착되는 모듈로, 이들 모듈에서는 입·출력 신호의 기능이 거의 같으므로 같이 언급한다. 입력신호는 P/W (Power Window) 입력신호 및 문 열림 검출 신호이며, 출력은 P/W Motor 출력 및 Step Lamp신호이며 입력 및 출력에 따른 포트 할당은 〈표 3〉과 같다.

〈표 3〉 입력 및 출력신호에 따른 포트할당 (AS, RR, RL 모듈)

| 입력 | | 출력 (87C196CA Port) | |
|---|---|---|---|
| AS, RR, RL 모듈 창 | PortA 0(↑) 1(↓) | P/W motor | Port 1.0 (↑) |
| 문 열림 검출 | PortA 3 | | Port 1.1 (↓) |
| *문 잠금 신호 | PortA 2 | Door Actuator | Port 1.2 |
| | | Step Lamp | Port 1.3 |

* 문 잠금 신호는 AS 모듈에만 있음

## 2. 패킷의 정의

차량용 Network 시스템에서 다섯 개의 모듈들은 서로 독립적으로 운영되고 있으며 구성된 Network을 통하여 다른 모듈들에게 어떤 정보를 제공하거나 다른 모듈을 제어할 수 있도록 되어야 한다. 효율적인 패킷을 구성하기 위하여 먼저 각각의 모듈에서 발생될 수 있는 입·출력되는 신호들 사이에서 Network을 통하여 전송하는 신호와 전송 받는 신호를 구분하였다.

Network 신호들을 살펴보면 크게 두 가지로 나뉘어지는데 CL 모듈로 전송되는 메시지1과 DR, AS, RR, RL 모듈들로 전송되는 메시지2로 구분된다. 효율적인 통신 패킷을 정의하기 위하여 이 신호들을 하나의 ID를 가진 메시지로 통합하고, 구분해야 할 부분은 메시지 내부에 포함시켰다. 그래서 두 가지 ID의 메시지로 나눌 수 있는데, 메시지의 중요성을 고려하여 문 열림/닫힘 상태는 주행중이거나, 주차중이거나 매우 중요한 정보가 되므로 낮은 ID를 주어 우선권을 높였고, 다른 정보는 높은 ID를 주어 우선권을 낮추었다.

Message 1은 DR, AS, RR, RL 모듈이 CL 모듈로 전송하는 정보로 〈표 4〉에 ID 및 Format을 나타내었다.

〈표 4〉 Message 1의 ID와 Format의 정의

Message 1

ID 00010001001

메시지 내용 DR, AS, RR, RL로부터 CL모듈로 전송되는 문 열림/닫힘 정보

메시지 format

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| × | × | × | × | ST | 0 | MD1 | MD0 |

X:don't care

ST:문열림/닫힘 상태(0:열림 1:닫힘)

| MD1 | MD0 | module |
|---|---|---|
| 0 | 0 | DR |
| 0 | 1 | AS |
| 1 | 0 | RR |
| 1 | 1 | RL |

메시지 전송한 모듈을 나타낸다.

Message 1은 CL 모듈에만 전송되는 메시지이므로 CL 모듈에서의 Acceptance Filter는 Message 1만 받아들이도록 설정하면 된다. CL모듈에서는 새로운 메시지가 입력되면 MD1, MD0 bit를 check하여 보내온 모듈을 확인하고 열림/닫힘 상태를 확인한다.

Message 2는 DR, AS, RR, RL로 전송되는 정보로 〈표 5〉에 ID 및 Format을 나타내었다.

〈표 5〉 Message 2의 ID와 Format의 정의

Message 2

ID 00100010001

메시지내용 CL, DR, AS 모듈에서 DR, AS, RR, RL모듈로 전송되는 복합적 정보

메시지 format

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CL-i | CL-d | CY-i | CY-d | MD1 | MD0 | PW1 | PW0 |

| CL-i | CY-i | 동작 mode |
|---|---|---|
| 0 | 0 | P/W 신호로 MD1, MD0에 모듈정보를 PW1, PW0에 P/W motor 제어신호를 담고있다. |
| 0 | 1 | DR, AS 모듈의 Door-Actuator 제어신호로 CY-d에 on/off제어신호를 담고 있다. |
| 1 | 0 | CL 모듈의 Door-Actuator 제어신호로 CL-d에 도난방지 제어신호를 담고 있다. |
| 1 | 1 | 사용하지 않음 |

| MD1 | MD0 | module |
|---|---|---|
| 0 | 0 | DR |
| 0 | 1 | AS |
| 1 | 0 | RR |
| 1 | 1 | RL |

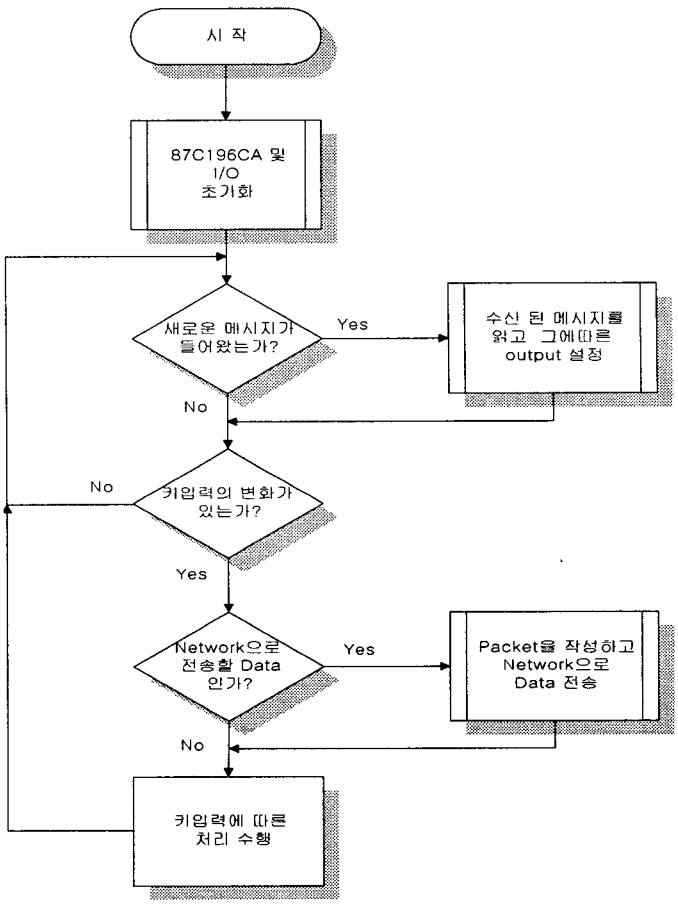
| PW1 | PW0 | motor control |
|---|---|---|
| 0 | 0 | STOP |
| 0 | 1 | CW |
| 1 | 0 | CCW |
| 1 | 1 | not used |

message 2는 DR, AS, RR, RL 모듈들이 공유하는 정보이다. Door Actuator 제어정보는 네 개의 모듈이 일제히 동작하므로 구분지을 필요가 없으며, Power window 제어정보는 모듈마다 다르므로 MD1, MD0에서 지정해 주게 된다. DR, AS, RR, RL 모듈들은 새로운 메시지가 전송되면 CL-i, CY-i를 취하여 어떤 정보인지를 판단하는데, Door

Actuator 신호일 때는 모든 모듈이 같이 동작되고, P/W 신호일 경우 MD1, MD0를 취하여 자신에게 주어진 정보인지를 다시 한번 조사하여 자신의 정보이면 거기에 알맞은 처리를 하게 된다.

### 3. 모듈별 software

모든 모듈에서의 동작은 기본적으로 CAN controller의 상태를 점검하여 새로운 메시지가 들어왔는지를 검색하여 있으면 해당되는 처리를 하고, 자신의 키 상태를 점검하여 키 상태의 변화가 있으면 또한 그에 따르는 처리를 하는 일련의 순환과정을 반복한다 (그림 2).

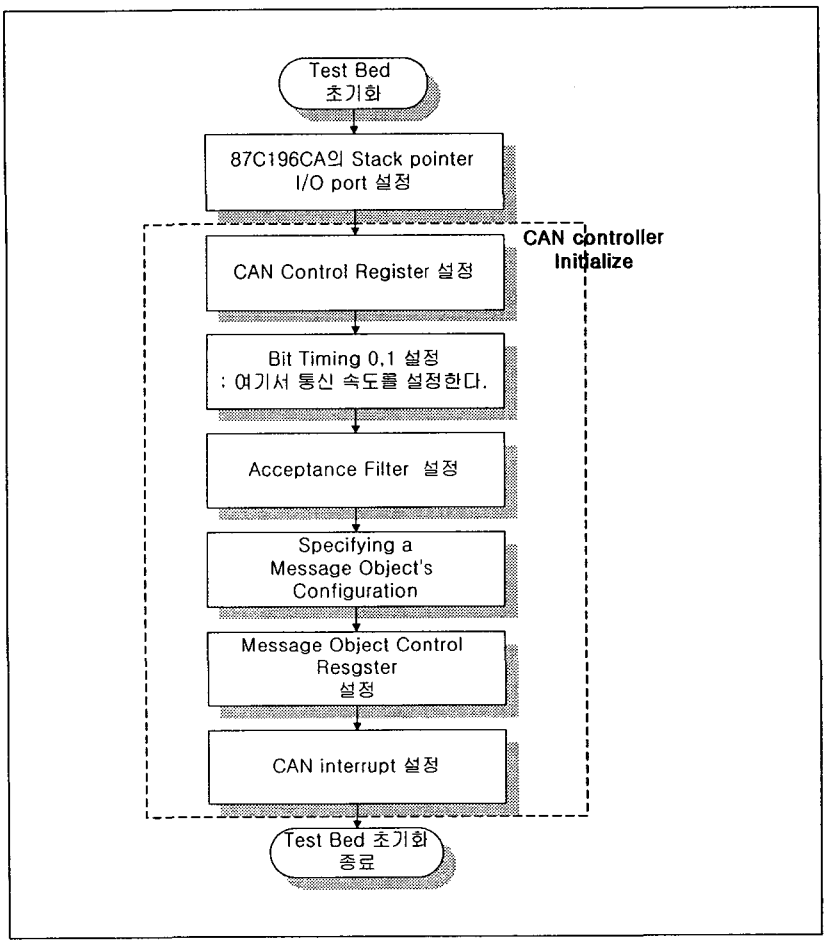



〈그림 2〉 각 모듈의 Main Routine

Test Bed의 software routine은 크게 초기화, 수신확인 및 처리, 키 변화 확인 및 처리의 세 부분으로 나뉘어진다. 초기화는 모든 모듈이 동일하며 다른 부분은 기능과 특성에 따라 조금씩 차이가 나타나는데 CL 모듈, DR 모듈, 그리고 AS, RR, RL 모듈로 구분하였다.

### 3.1 Test Bed의 초기화

Test Bed의 초기화는 모듈에 관계없이 모두 동일하며 각 포트 및 키 상태의 초기화, Network Controller의 초기화로 구성된다(그림 3).




〈그림 3〉 Test Bed의 초기화 과정

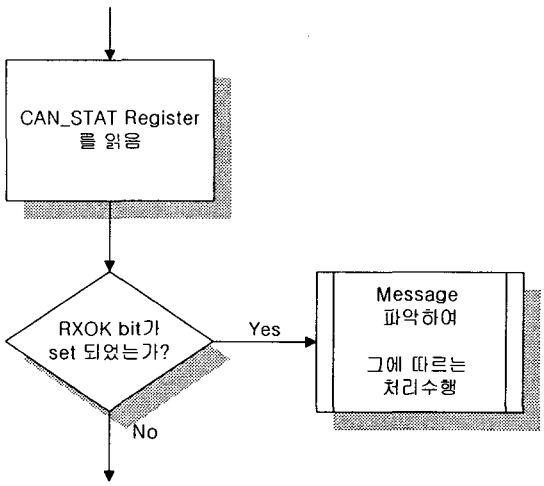
87C196CA 및 PSD302 포트는 Programmable I/O 포트이므로 사용하기전 입·출력을 설정하여야 한다. 또한 각 포트들은 I/O 기능 외에도 특별한 기능을 가지고 있는데, 기능을 쓰기 위해서는 Px-mode Register를 설정한다. Test Bed에서는 특별한 기능은 사용하지 않고 Standard I/O Port mode로 사용하였는데, Wiper Motor control Board에서는 EPA와 Timer/Counter를 Special mode로 사용한다. 〈표 6〉에 각 모듈의 입·출력 초기 설정을 나타내었다. 또한 키 상태의 초기치는 모두 '1'로 설정한다.

〈표 6〉 각 모듈에서의 포트의 초기화 설정

| 구분 | 입력 포트 | 출력 포트 |
|---|---|---|
| DR 모듈 | PSD302 PortA<br>87C196CA Port1 | 87C196CA Port2<br>Port6 |
| CL, AS,<br>RR, RL 모듈 | PSD302 PortA | 87C196CA Port1<br>Port2<br>Port6 |

### 3.2 수신확인 및 처리

수신확인 및 처리루틴은 새로운 메시지가 들어왔는지 검색하고 있으면 그에 따르는 처리를 수행한다. 〈그림 4〉에서 나타난 바와 같이 수신확인은 CAN-STAT Register를




〈그림 4〉 새로운 메시지여부 점검

〈표 7〉 CAN-STAT Register

CAN-STAT(1e01H)

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| BUSOFF | WARN | – | RXOK | TXOK | LEC2 | LEC1 | LECO |

BUSOFF : Bus Off Status >> CAN bus로부터 격리되었을 때 set

WARN : Warning Status >> Error counter가 96에 도달하면 set

RXOK : Reception Successful >> Error 없이 어떤 데이터가 전송되면 set

TXOK : Transmission Successful >> Message의 전송이 완료되면 set

LEC : Last Error Code

0 0 0 no error

0 0 1 stuff error

0 1 0 form error

0 1 1 acknowledgement error

1 0 0 bit 1 error

1 0 1 bit 0 error

1 1 0 CRC error

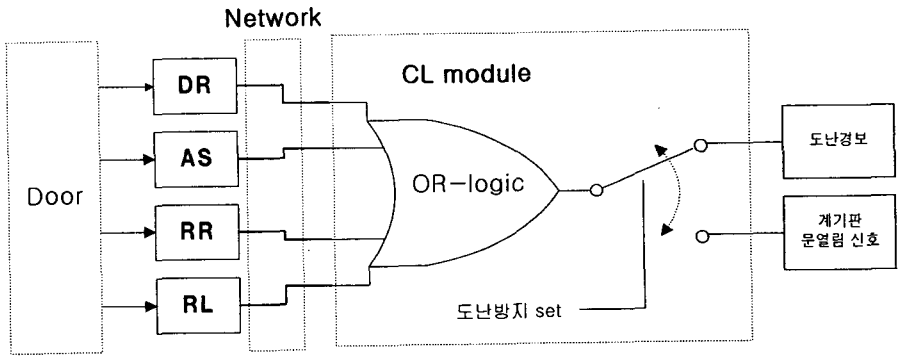
1 1 1 unused

읽어 CAN상태를 점검함으로써 가능하다. CAN-STAT Register의 내용은 〈표 7〉에 나타내었다.

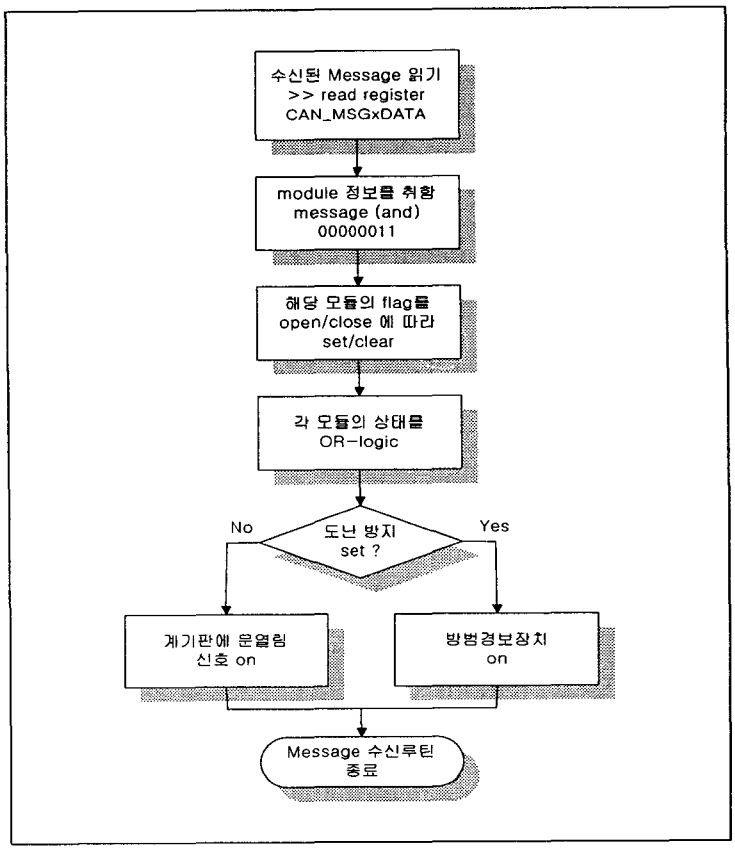
메시지 파악은 패킷에 정의된 형태를 기준으로 분석한다. 본 연구에서 사용된 Test Bed 사이에서 메시지는 다음과 같은 두 가지 형태가 있으며 수신하는 모듈도 서로 다르다.

### 3.2.1 CL 모듈

먼저 Message 1을 수신하는 CL 모듈부터 살펴보면, CL 모듈에서 Network을 통하여 받는 Message 1 정보는 문 열림/닫힘 신호이므로 Message 내에 있는 모듈정보와 열림/닫힘 정보를 찾으면 된다. 모듈정보는 bit 0,1에 있으며 열림/닫힘 정보는 bit 3에 실려 있으므로 Mask를 취해 정보를 얻어내고 〈그림 5〉와 같이 각 문의 상태를 OR logic 연산하여 계기판의 문열림 신호나 도난 경보시스템을 구동한다. 〈그림 6〉에 CL 모듈에서의 수신 Message 처리과정을 나타내었다.



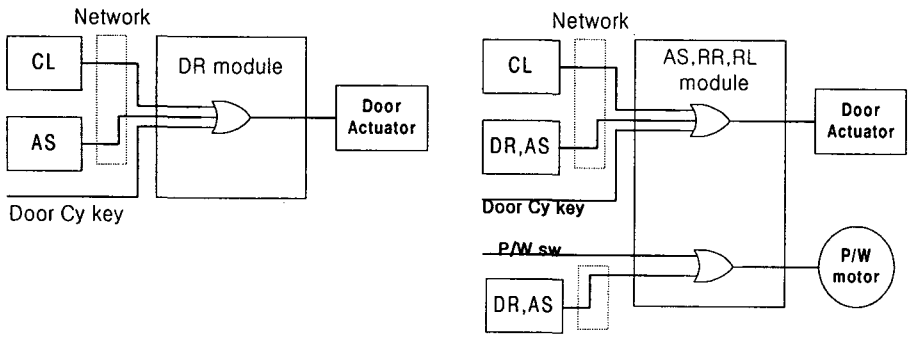

〈그림 5〉 CL 모듈에서의 Network 처리




〈그림 6〉 CL모듈의 수신 메시지

### 3.2.2 DR, AS, RR, RL 모듈

이들 모듈에서는 Message 2를 수신한다. Message는 CL 모듈의 Door-Actuator 신호, 및 DR, AS 모듈의 Door-Actuator의 신호, DR 모듈의 P/W(Power Window) 신호를 담고 있다. 〈그림 7〉과 같이 DR 모듈을 제외한 나머지 모듈에서는 이 신호들을 차례로 검색하여 그에 알맞은 처리를 한다. DR 모듈에서는 P/W 신호는 받지 않으므로 CL, AS 모듈에서 전송하는 Door-Actuator 신호만 검색하면 된다. 〈그림 8〉에 이들 모듈의 수신 메시지 처리과정을 나타내었다.

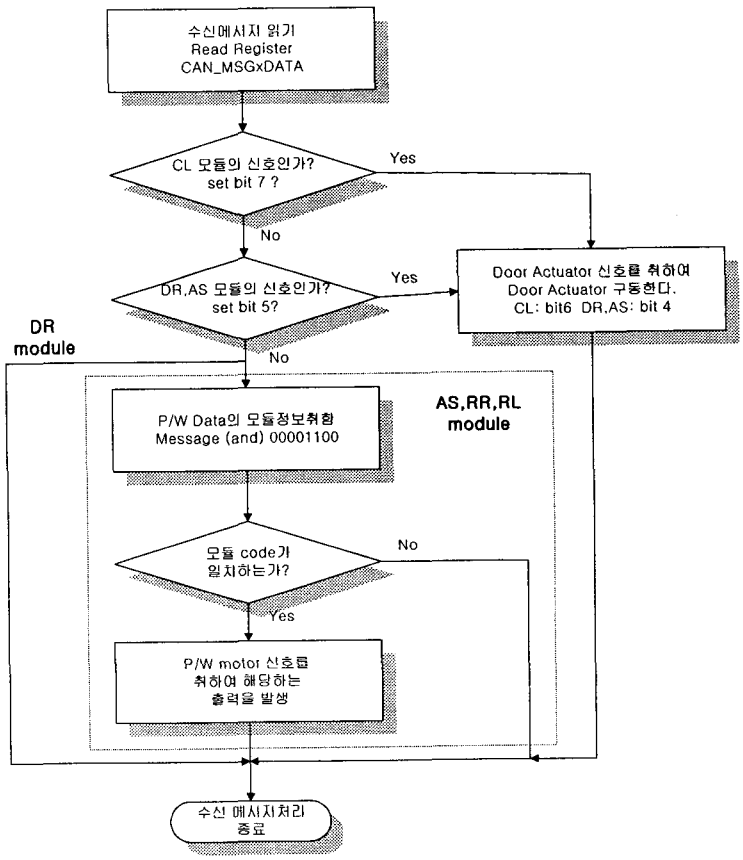



〈그림 7〉 DR, AS, RR, RL 모듈의 Network처리

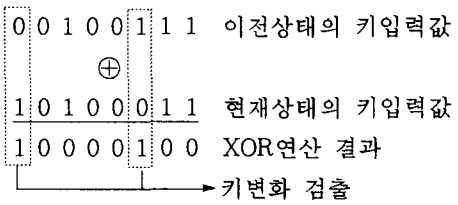
### 3.3 키변화 확인 및 처리

키 상태변화의 검출은 전 상태와 현 상태를 XOR logic 연산을 취하면 가능하다. 〈그림 9〉에서 보듯 XOR 연산을 취하면 변화된 bit는 '1'로 set 되어 있고 변화되지 않은 bit는 '0'이 된다.

변화된 키를 검출하여 먼저 이 신호가 Network으로 전송되어야 할 신호인지, 아니면 모듈 자체적으로 처리되는 신호인지 구분하여 Network 신호이면 데이터 전송후 키 처리를 한다.




〈그림 8〉 DR, AS, RR, RL 모듈의 수신메시지




〈그림 9〉 XOR 연산에 의한 키변화 검출

데이터 전송은 데이터 전송버퍼에 TX-REQ를 설정하면 전송된다. TX-REQ의 설정에 따라 전송 완료시 인터럽트도 발생시킬 수 있으나 실제 Test Bed에서는 CAN-STAT Register의 TXOK bit를 조사하는 Polling 방식에 의해 전송완료를 check 하였다.

## Ⅳ. 결 론

차량용 통신시스템의 개발에 필요한 5개의 모듈의 설계를 완료하였다. 설계된 모듈은 각종 스위치와 센서정보를 입력으로 받아서 경고등, 실내외등, 모터 등을 동작시키는 기능을 수행한다. 모듈의 구성을 위하여 CAN 기능이 내장되어 있는 인텔사의 87C196CA를 사용하였고, 설계된 모듈을 제작하여 각종 스위치, 센서, 램프, 릴레이, 모터 등과 연결하고 모듈의 작동에 필요한 프로그램을 개발하여 우수한 성능 평가를 하였다.

향후 과제로는 자동차의 충돌방지시스템, 위성송수신시스템, 거리정보시스템 등의 보다 많은 시스템들이 network을 통하여 정보를 송·수신할 때 발생 가능한 시간지연에 따르는 안정성 문제의 연구가 요구된다. 즉, 시스템 자체의 제어에 존재하는 시간지연과 통신상의 트래픽 제어에서 발생되는 시간지연을 함께 극복하는 방법에 대한 연구로 항공기나 우주선 등에서 적용하고 있으나 효과에 비하여 고비용이 드는 문제로 개발 단계에 있으며 다음과 같은 문제에 대한 시간지연 극복 기술이 연구되어야 한다.

① 시스템의 구성요소 사이의 동기가 일치되지 않음으로 인한 트래픽의 불균일
② 샘플링 지연
③ 메시지 거부
④ 통신매체의 노이즈 등

## 參 考 文 獻

1. 김종식(1992), 선형 제어시스템 공학, 청문각.
2. 김태윤(1995), 데이타통신과 컴퓨터통신, 집문당.
3. Chan, H. and Ü. Özgüner(1995), "Closed-loop control of systems over a communications network with queues," *Int. Journal of Control*, Vol. 62, No. 3, pp.493-510.
4. Kato, Mitsunori and Yutaka Matsuda(1994), "Evaluation of Latency Time for Class C High Speed LAN Protocols," SAE paper 940363.
5. Keiser, G. E.(1989), *Local Area Networks*, Macmillan.
6. Luen, W. L. and A. Ray(1990), "Integrated Communication and Control Systems: Part 3-Nonidentical Sensor and Controller Sampling," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 112, pp.357-364.
7. Ray, A.(1987), "Performance Evaluation of Medium Access Control Protocols for Distributed Digital Avionics," *Journal of Dynamic Systems, Measurement and Control*, Vol. 109.
8. Ray, A. and Yoram Halevi(1988), "Integrated Communication and Control Systems: Part 2-Design Considerations," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 110, pp.374-381.
9. Stallings, W.(1993), *Local and Metropolitan Area Networks*, Macmillan.

Abstract

Network Implementation for automobiles using CAN

Hur, Hwa-ra

In this study I construct CAN(Controller Area Network) for automobiles similar to LAN(Local Area Network) and build communication modules in the major part of an automobile to link several sub-systems.

Since each station replaces the communication function of sub-systems and has various types of sensor, actuator, controller, and switch, every information about automobile's status is obtained from the network. The manufactured system showed a superior capability.

The following is the contents of study.

1. The definition of communication packet through the analysis of CAN protocol.
2. The Design of modules using micro-controller 80C196CA.
3. The Network configuration.