

## Controller Area Network을 이용한 자동차 전동거울 시스템 구현

윤상진\*, 조웅석\*\*, 이종성\*\*\*, 박 기현\*  
 \*성균관대학교, \*\*건양대학교, \*\*\*부천대학

### Realization of automobile electromotive mirror system using Controller Area Network(CAN Protocol)

Sangjin Yoon\*, Yongseok Cho\*\*, Jongsung Lee\*\*\*, Kiheon Park\*  
 \*Sung-Kyun-Kwan Univ, \*\*Konyang Univ, \*\*\*Bucheon College

**Abstract** - In this paper, realized an automation system that applies the automobile electromotive mirror using CAN(Controller Area Network : ISO 11898). CAN is being used mainly in ECUs (Electronic Control Units) connection of control system or automobile inside. And it has high reliability in the various network protocol.

To be realized position system, Automobile Electro mirror, has a strong point that establishment e decrease and ease of maintenance it compare PPP(point-to-point)method of existed. The realization composed of three portions. One Input Slave Con which accept a user's input, another Output Slave Co which dreve it makes the motor of electromotive mirr other Master Controller which interfacing the two Controller. Automobile electromotive mirror realized time system that will be able to minize the delay t point of time user's input until output point of time mirror.

### 1. 서 론

최근 자동차내의 전자 부품의 증가와 전자 시스템의 기능 향상 등에 의해 하니스(Harness, 작업설비)는 보다 복잡화되어 중량의 증대와 전선다발의 비대화를 초래하였다. 이는  $10^{-3} [m^2]$ 의 공간과  $10^{-3} [Kg]$ 의 중량까지 설계사양으로 고려하는 자동차 설계에 있어서 심각한 문제로 대두된다. 따라서, 전자 시스템의 제어가 보다 고도화되어 복수의 시스템이 집적화, 복잡화 될수록 자동차의 전장시스템에 대한 토탈 네트워크(Total Network) 기능이 필요하게 되었다.

본 연구에서는 다양한 필드버스 프로토콜 중에서 국제표준으로 선정되어 시장성이 뛰어나고 데이터 전송시 어려움이 매우 낮아 제어응용에 가장 적합한 CAN 프로토콜을 실시간 데이터 전송이 중요하고 많은 입출력 장치들을 동시에 제어해야 하는 차량내 전자장치 제어부 중의 하나인 사이드미러에 적용함으로써 CAN을 이용한 멀티플렉싱 시스템을 구현하고 이를 분석하였다.

### 2. 본 론

#### 2.1 CAN(Controller Area Network)

분산형 제어시스템에서는 마이크로프로세서 사이의 통신기능이 필수적이며, 이러한 목적을 위하여 개발된 네트워크 시스템이 필드버스이다. CAN(Controller Area Network)은 1986년 독일의 Bosch GmbH에 의하여 개발되었으며, 최초의 CAN 칩은 Intel에 의하여 1987년에 만들어졌고, 이 후 Philips, NEC등에서 CAN 칩이 제작되었다. CAN의 특징은 국제표준으로 선정되어 시장성이 뛰어나고(ISO11898, 1993), 높은 전송률과 안정성을 제공하고

있어서 다수의 ECU(Electronic Control Unit)를 상호연결하는 분산시스템의 실시간 제어를 효율적으로 지원할 수 있다(2),(3).

#### 2.2 CAN의 구조

(그림 1)은 CAN의 계층구조를 나타내었다. CAN은 OSI 7계층 모델중 물리계층과 데이터링크 계층만으로 구성되며, Master와 Slave구조를 가지고 있어 Network상에서의 제어를 가능하게 한다. 데이터링크 계층은 객체계층(Object Layer)과 전달계층(Trans Layer)으로 나뉘어지는데, 객체계층은 어떤 메시지가 전송되는지를 찾고, 전달계층으로부터 받은 메시지를 필터링하는 기능을 수행한다(Message Filtering). 즉, 하드웨어와 관련된 응용계층 인터페이스를 제공하는 역할을 수행하는 부분이 객체계층이며, 이에 반해 전달계층은 CAN프로토콜의 핵심부분으로서, 수신한 메시지를 객체계층에 보내고, 객체계층으로부터 송신해야 할 메시지를 받아들이는 과정을 수행한다.

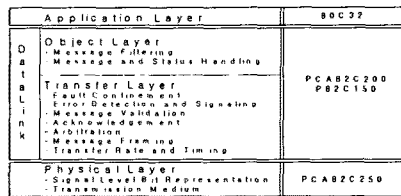


그림 1. CAN의 계층구조

#### 2.3 CAN 프로토콜

##### 2.3.1 메시지 종류

CAN 프로토콜은 데이터프레임, 리모트프레임, 에러프레임, 오버로드 프레임의 4종류의 메시지 프레임을 사용하여 데이터를 전송한다.

· 데이터 프레임(Data Frame): 데이터를 전송하는 프레임으로 (그림 2)와 같이 7개의 필드형식으로 구성되어 있으며, 최대 8바이트의 데이터를 전송할 수 있다. CAN에서는 11비트의 확인자(Identifier)를 통해서 데이터 전송을 위한 버스 액세스 우선순위(Bus access priority)를 조정하게 되는데 확인자값이 작은쪽이 우선순위가 높아서 버스를 액세스할 수 있는 CSMA/AB (Carrier Sense Multiple Access with Arbitrary Bitwise) 방법을 사용하고 있다.

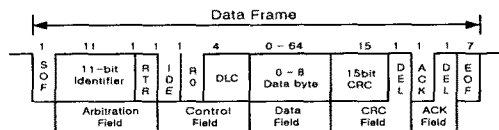


그림 2. Data Frame 구조

· 전송요청프레임(The Remote Frame): 한 노드에서 다른

노드로 데이터를 요구할 때 전송하는 프레임으로 <그림 3>과 같이 6개의 필드로 구성된다. 데이터 필드가 존재하지 않으며, RTR(Remote Transmission Request)비트가 recessive('r') 값을 가지며, 데이터프레임과 동일한 확인자를 갖고 있다.

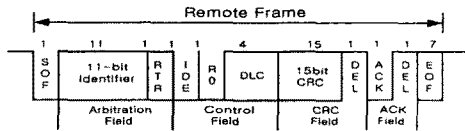


그림 3. Remote Frame 구조

· 에러프레임(Error Frame): 한 노드에서 수신한 메시지 중에서 에러를 발견할 때 전송하는 프레임으로 <그림 4>와 같이 두 개의 필드로 이루어진다. 여기서 에러플래그(Error Flag)에는 능동 에러플래그(Active error flag)와 수동에러플래그(Passive error flag)가 있는데, 수동에러플래그는 6개의 연속적인 dominant('d')로, 수동에러플래그는 6개의 연속적인 recessive('r')비트로 구성된다.

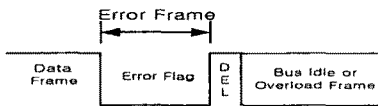


그림 4. Error Frame 구조

· 오버로드프레임(The Overload Frame): 현재 수행중인 프레임의 다음에 오는 데이터 프레임과 리모트 프레임을 지연시키기 위해 사용된다.

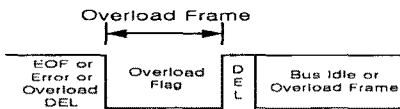


그림 5. Overload Frame 구조

· 인터프레임 스페이스(Interframe Space): 데이터프레임과 리모트프레임은 3비트의 인터프레임스페이스에 의해 프레임과 분리된다. 반면 오버로드 프레임과 에러 프레임 앞에는 인터프레임 스페이스가 존재하지 않으며 또, 여러개의 오버로드 프레임은 인터프레임스페이스에 의해 분리되지 않는다.

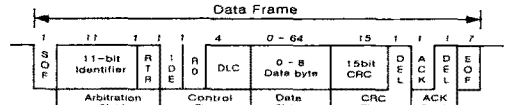
### 2.3.2 주소지정 방식과 메시지 우선순위

메시지 필터링(Message Filtering)에 의한 주소지정 방식은 한 노드에서 메시지를 보내고자할 때 데이터와 확인자(Identifier)를 함께 보내는데, 수신대기상태에 있는 다른 노드들이 메시지를 수신하면, 확인자를 필터링하여 그 데이터가 자신과 관련성이 있는지 없는지를 판별-각 노드의 하드웨어적인 구성과 응용 계층의 설정에 의해 판별한다-하여 관련이 있으면 데이터를 받아들여 저장하고, 없으면 무시하게 된다. 이와 같은 메시지 전달방식은 현재 구성된 CAN 시스템에 새로운 CAN node 등의 스테이션이 추가되더라도 소프트웨어나 하드웨어의 재구성이 필요없는 유연성있는 시스템구축을 가능하게 한다.[2]-[4]

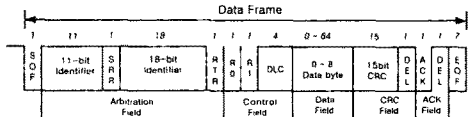
### 2.3.3 CAN메시지 포맷

CAN의 메시지 포맷은 버전2.0A의 표준포맷과 2.0B의 확장 포맷이 있다. 이에따라 CAN Controller가 버전2.0A만 지원하는 경우에는 버전2.0B형식의 메시지는 처리할 수 없지만 2.0B버전을 지원하는 CAN Controller제어기는 두가지 포맷의 메시지를 모두 처리할 수 있다.이 두 버전의 구별은 IDE(Identifier Extension)비트에 의해 레벨 '1'이면 버전 2.0B

포맷을 레벨 '0'이면 버전 2.0A포맷을 나타낸다. SRR비트는 같은 BasicID를 갖는 2.0A포맷의 메시지와 2.0B포맷의 메시지가 있을 때 2.0A포맷의 메시지가 더 높은 우선순위를 갖도록 하는 역할을 하는 레벨 '1'의 비트이다.



(a) 표준 메시지 포맷(Standard Message Format)



(b) 확장 메시지 포맷(Extension Message Format)

그림 6. CAN의 메시지 포맷

## 2.4 전체시스템 구성

본 연구에서 적용한 자동차의 전동거울 시스템은 상하좌우를 동작 시키는데 필요한 2개의 직류전동기와 미러의 위치를 저장하고 로드하기 위한 위치센서, 그리고 미러전체를 개·폐시키는데 필요한 전동기와 미러의 열선으로 구성되었다. 이를 제어하기 위해 <그림 7>에서 보논바와 같이 입력부와 출력부 스테이션인 슬레이브컨트롤러(Slave Controller) 2개와 마스터 CAN컨트롤러(Master CAN Controller)를 사용하여 중앙의 CAN Bus를 통해 멀티플렉싱 시스템을 구현하였다.

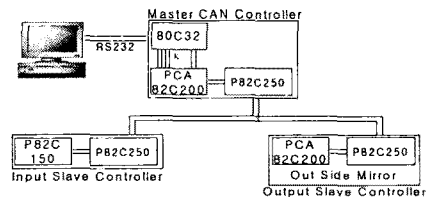


그림 7. 구성된 전체 시스템의 블록도

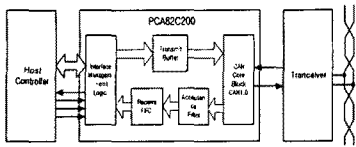
### 2.4.1 마스터 컨트롤러(Master Controller)

<그림 8>의 (a)와(b)는 마스터 컨트롤러 구성의 블록도와 응용이다. 본 논문에서는80C32(Intel)와 PCA82C200(Philips) CAN Controller를 사용하여 마스터 CAN 제어부를 구성하고, P82C250을 CAN 프로토콜 전송기(Transceiver)로 사용하였다.

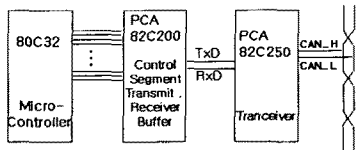
여기서 CAN제어기(PCA82C200)는 물리계층(Physical Layer)과 응용계층(Application Layer)사이의 중계역할을 한다. 칩내부에 프로토콜이 내장되어 있어 사용자가 별도의 프로토콜 생성 알고리즘을 응용계층에서 만들 필요가 없으며, CAN버스를 통하여 PCA82C200의 수신버퍼에 들어오는 메시지를 8032의 수신 버퍼에 저장하는 방식으로, 메시지를 입력받고 이와 반대되는 과정으로 메시지를 출력하는 방식으로 데이터를 처리한다. 즉, CPU인 82C32는 CAN제어기(PCA82C200)의 수신버퍼에 저장 되는 데이터를 할당된 번지에 받아들여 그 값을 기억하고, 필요한 시기마다 그 해당번지에서 데이터를 읽어오게 한다.

### 2.4.2 슬레이브 컨트롤러(Slave Controller)

<그림 9>은 입력부와 출력부의 슬레이브 컨트롤러(Slave Controller)의 구성에 대한 블록도이다. 슬레이브컨트롤러를 구성



(a) PCA82C200의 구성 블록도



(b) PCA82C200의 응용

그림 8. Master Controller의 블록도

하는데는 입출력 제어칩인 P82C150과 PCA82C250(CAN 기)를 사용하였다. P82C150은 16개의 아날로그/디지털 입출력 포트가 내장되어 있어 본 연구의 적용시스템인 자동차의 아웃사이드미러의 상하좌우, 전동접이, 메모리기능을 수행하기 위해 입력 슬레이브컨트롤러(Input Slave Controller: SLIO1)에 10개의 포트를 설정하였고, 출력 슬레이브 컨트롤러(Output Slave Controller:SLIO2)에는 아웃사이드 미러의 구동과 전동접이 그리고 열선기능을 구현하기 위해 7개의 포트를 출력포트로 이용하였으며, 기준위치저장/로드 및 ADC(Analog to Digital Conversion)를 위하여 4개의 포트를 사용하여 해당 기능을 처리토록 하였다.

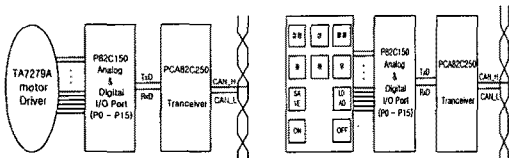


그림 9. 입력부, 출력부 Slave Controller의 블록도

### 2.4.3 프로그램

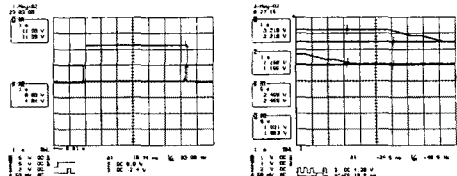
본 실험에서는 C/C-51컴파일러를 사용하여 Slave Controller 1(SLIO 1)에서 받아들인 사용자 입력에 따라 Slave Controller 2(SLIO 2)의 사이드미러를 제어할 수 있는 프로그램을 작성 사용된 프로그램의 구조를 간단히 정리해보면,

- step1. 80C32/PCA82C200의 초기화(Master Controller)
- step2. P82C150의 초기화
- step3. 외부 인터럽트 대기
- step4. 외부 인터럽트 실행
- step5. 메시지 전송
- step6. 버퍼에 입력 데이터 저장
- step7. 정해진 인터럽트의 기능수행

### 2.4.4 실험결과

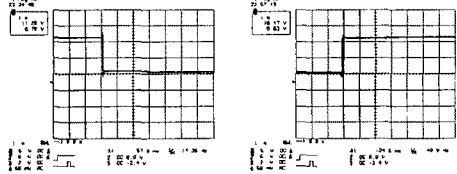
(그림 10)의 (a),(b),(c),(d)는 전동거울 구동시의 출력파형을 나타내었다. (a)는 SLIO1에서 받아들인 사용자 기준입력값을 Master로 보내고 해당 기능의 메시지를 SLIO3로 보내어 전동거울이 동작할때까지의 파형을 나타내었으며 입력과 출력파형 사이에 나타나는 발생지연시간, 행렬대기시간(Queueing Delay), 메시지지연시간(Transmission Delay), 그리고 전달지연시간(Delivery Delay)에 의한 약 15ms의 시간지연이 나타남을 알 수 있으며, 이것은 데이터필드(Data Field)의 크기와 노드의 증가/감소, 데이터경로의 길이등에 의해 달라질 수 있으며, 설계되는

시스템의 특성에 맞춰 최적의 파라미터값을 찾아 설계해야 할 것이다. 또 (b)에서의 기준위치를 저장하고 다시 찾아가는 파형을 나타내었다. 기준위치로의 복귀시 한번의



(a) 입력에서 출력(상, 하, 좌, 우)

(b) 기준위치로의 복귀



(c) 전동거울을 펼칠 때

(d) 전동거울을 닫을 때

그림 10. 전동거울 구동시의 출력파형

깊이가 있음을 발견할 수 있는데 기준위치와 현재위치의 차이에 의해 한정된 만큼의 구동 후 이동한 위치를 다시 저장된 위치와 비교하여 ADC(Analog to Digital Conversion)과정을 다시 수행해야 하는 칩셋 특성이 있기 때문이다. 파형(c)와 (d)는 전동거울을 펼칠 때와 닫을 때의 파형을 나타내었다.

## 3. 결 론

본 연구에서는 자동차의 전동거울을 제어하기 위해 CAN을 이용한 멀티플렉싱 시스템을 구현하였으며, 이를 통하여 차량내 각종 전장시스템과 ECU(Electronic Control Unit)들을 하는데 있어서 배선수를 감소시켜 고장발생요인을 줄이는 한편, 협소한 차체 영역에서 효과적인 실장공간을 제공할 수 있다는 등의 장점을 얻을 수 있었다. 또한, 구현한 자동차의 전동거울 시스템은 기존에 장착되어진 하드웨어나 소프트웨어를 교체 할 필요없이 새로운 기능을 수행하는 다른 노드를 연결할 수 있는 우수한 확장성 및 유연성을 가지고 있었다. CAN의 시스템 응용시 시스템의 특성에 맞는 상위 레이어의 개발을 필두로 한다면 어떤 시스템특성에도 부합할 수 있는 즉, 하위계층부터 상위계층 모두를 표준화 시킬수 있을 것이며, 특정 시스템만의 응용이 아닌 모든 시스템에 응용 될 수 있는 프로토콜로 정착될 것이다.

### (참 고 문 헌)

- [1] J. Rufino, P. Verissimo, "A Study on the Inaccessibility Characteristics of the CAN", 2nd International CAN Conference, 1995
- [2] BOSCH, CAN Specification, Part A,B 1991.
- [3] Ken Tindell, Alan Burns, "Guaranteed Message Latencies for Distributed Safety-Critical Hard Real Time Control Network", report YCS229, Department of computer Science, University of York, May 1994.
- [4] CiA, CAN Layer, www.can-cia.de, 1997.
- [5] 윤상진 외, "CAN(Controller Area Network)을 이용한 멀티플렉싱 기술개발", 대한전기학회 하계학술대회, pp2206~2208, 7. 2001
- [6] SJA1000 Stand-alone CAN controller, AN97076, Philips Semiconductors, 1997.