

복수시스템 제어를 위한 멀티플렉싱 기법 개발

오 동진\*, 윤 상진\*\*, 조 용석\*\*\*, 박 기현\*\*  
 \*현대정공주식회사, \*\*성균관대학교, \*\*\*건양대학교

Development of a Multiplexing Method for Multi-System Control

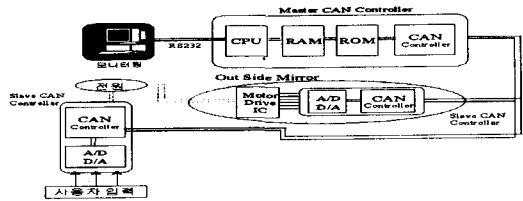
Dongjin OH\*, Sangjin Yoon\*\*, Yongseok Cho\*\*\*, Kiheon Park\*\*  
 \*HYUNDAI PRECISION&IND.CO.LTD, \*\*Sung-Kyun-Kwan Univ, \*\*\*Konyang Univ

**Abstract** - In this paper, CAN(Controller Area Network)is used to solve the wire harness problem in the outside mirror of automobile which has a lot of functions in narrow space. If the number of wires is reduced, it has benefits of lower product cost and maintenance.

CAN was originally developed by the German company Robert Bosch for use in the car industry to provide a cost-effective communications bus for in car electronics and as alternative to expensive and cumbersome wiring looms.

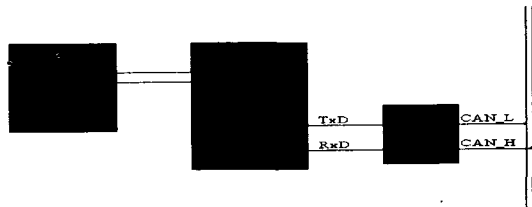
CAN controller is a serial communication protocol which efficiently supports distributed real-time control with a very high level of security.

The communication between master CAN controller and slave CAN controller is realized and controller's performance is tested by experiment.



[그림 2.1] CAN의 블록도

토콜 컨트롤러를 거쳐 수신 버퍼에 저장되는 방식으로 데이터 처리가 된다. 따라서 마스터 컨트롤러의 CPU는 PCA82C200의 수신버퍼에 저장되어 있는 데이터를 읽어 들여 분석 하여야 한다.



[그림 2.2] 마스터 컨트롤러의 구성 블록도

1. 서 론

컴퓨터와 전자 기술의 발전은 생산 장비들과 산업 현장의 공정들을 자동화시켜 공정의 생산성을 높이고 생산 비용을 줄여왔으며 각종 시스템간의 데이터를 공유하여 생산을 담당하는 기기들 간의 생산 정보를 실시간으로 전달시켜 효과 적인 시스템의 감시 및 제어를 수행하여 시스템의 성능을 최적화 하려는 경향이 증대하고 있다. 여기서, 자동차의 전동 거울을 운용하고 통제하는데

CAN프로토콜을 이용하였는데, 기존의 복잡한 선들을 없애고 두 가닥의 통신선 만으로 많은 장치들을 효율적으로 제어, 관리할 수 있었다. 따라서, 본 논문의 적용 대상은 최소 11가닥의 전선수로 구성된 자동차의 아웃사이드 미러를 CAN통신선과 신호선 즉, 4가닥의 전선으로 서 원활히 동작시키는 것에 초점을 맞추었다.

2. 본 론

2.1 시스템 구성

실험 대상인 자동차의 아웃사이드 미러는 거울을 상·하, 좌·우로 조정할 수 있는 2개의 모터와 메모리 기능을 위한 위치센서, 그리고 아웃사이드 미러의 개폐를 위한 1개의 모터와 열선장치의 설치에 총 11가닥의 전선이 사용되었다. CAN을 사용하여 2가닥의 전원선과 2가닥의 신호선을 사용하여 총 4가닥의 전선을 사용하여 기존의 기능을 수행하는 것이 본 연구의 목적이다. CAN의 구성은 마스터 컨트롤러(Master Controller)와 슬레이브 컨트롤러(Slave Controller)로 구성된다. [그림 3.1.]은 구성할 하드웨어의 블록도를 나타내었다.

2.1.1 마스터 컨트롤러

[그림 2.2.]은 마스터 컨트롤러의 구성도를 나타내었다. CAN은 칩내부에 프로토콜 컨트롤러가 내장되어 있어, 사용자가 별도로 프로토콜을 만들 필요는 없지만, PCA82C200의 수신버퍼에 들어오는 메시지를 분석하여야만 한다. CAN버스를 통하여 수신되는 데이터는 프로

2.1.2 확인자 필터링

[표 2.1.]은 확인자 비트(Identifier Bit)의 포맷을 나타 내었다. 이 확인자의 설정은 각각의 SLIO의 16개 포트중 P0~P3포트를 확인자 프로그램 입력 포트로 사용하여 SLIO자신의 번호를 결정 한다. 입력 포트 P0~P3의 값은 각각 ID.8, ID.5, ID.4, ID.3번의 값을 결정한다. 이렇게 결정된 확인자가 마스터 컨트롤러의 수신버퍼에 저장 되면 CPU는 이 수신 버퍼를 읽어 들여 분석하여 어느 SLIO에서 온 데이터인지를 알 수 있고, 같은 방식으로 확인자를 설정하여 원하는 SLIO에 데이터를 전송할 수 있다. DIR 비트는 방향을 결정하는 비트로서 DIR이 '1'로 설정되면 SLIO에서 마스터로 데이터가 전송되고, DIR 비트가 '0'으로 설정되면 마스터에서 SLIO로 데이터를 전송 한다. RTR비트가 '1'로 설정되면 전송 요청 프레임(Remote Request Frame)이된다.

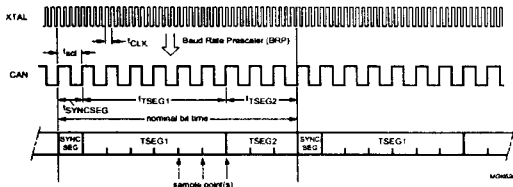
[표 2.1] 확인자 설정 포트

IDENTIFIER												
ID.	ID.9	ID.8	ID.7	ID.6	ID.5	ID.4	ID.3	ID.2	ID.1	ID.0		
10	0	1	P3	1	0	P2	P1	P0	1	0	DIR	RTR

2.1.3 버스타이밍의 정의

CAN의 특징중 하나는 샘플 포인트와 샘플링의 갯수를 사용자가 프로그램을 할 수 있다는 것이다. PCA82C200

은 2개의 레지스터를 사용하여 CAN버스의 속도를 결정한다. 우선 CAN버스의 속도를 결정 하기전에 CAN의 비트의 구조를 살펴보기로 하겠다.



[그림 2.3] BIT의 구조

CAN의 비트 구조는 [그림 2.3]과 같으며 아래와 같이 구성되어 있다.

- 동기화 세그먼트(Synchronization Segment)
- 타임 세그먼트 1(Time Segment)
- 타임 세그먼트 2(Time Segment)

동기화 세그먼트는 항상 1개의 시간 단위량(Time Quantum)의 길이를 가지며, 항상 클럭의 동기화에 사용되고, 시간 세그먼트1과 시간 세그먼트2의 경계에서 샘플링이 이루어진다. PCA82C200의 비트 타이밍을 정하는 레지스터는 BTR0(Bit Timing Register 0 : Address 6), BTR1(Bit Timing Register 1 : Address 7)이며 설정값은 [표 2.2]에 나타내었다.

[표 2.2] BTR0와 BTR1의 설정값

	7	6	5	4	3	2	1	0
BTR0	SJW2	SJW2	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
BTR1	SAM	TSEG2	TSEG2	TSEG2	TSEG2	TSEG2	TSEG2	TSEG2
		22	21	20	13	12	11	10

CAN버스의 속도를 결정하기 위해 BRP(Baud Rate Prescaler)로부터  $T_{SCL}$ (Period of The System Clock)을 식(3.1)로 구한다.

$$T_{SCL} = 2T_{CLK}(32BRP5 + 16BRP4 + 8BRP3 + 4BRP2 + 2BRP1 + BRP0 + 1) \quad (3.1)$$

$T_{CLK}$  : Time Period of The PCA82C200 Oscillator

BTR1을 이용하여 한 비트당 주기와 샘플링 포인트를 결정한다. [표 2.2]에서의 SAM(Sampling)값이 '0'일 때는 샘플링을 한번하고, '1'일 때는 샘플링을 3번한다. 샘플링 포인트의 위치와 한 비트 주기당 클럭 사이클을 결정하는 Time Segment 1과 Time Segment 2의 값들을 각각 식(3.2)와 식(3.3)을 이용하여 결정한다.

$$T_{TSEG1} = T_{SCL}(8T_{SEG13} + 4T_{SEG12} + 2T_{SEG11} + T_{SEG10} + 1) \quad (3.2)$$

$$T_{TSEG2} = T_{SCL}(4T_{SEG22} + 2T_{SEG21} + T_{SEG20} + 1) \quad (3.3)$$

구해진  $T_{SCL}$ 과  $T_{TSEG1}$ ,  $T_{TSEG2}$ 로부터 버스 타이밍(Bus Timing)T를 식(3.4)로 구한다.

$$T = T_{SCL} + T_{TSEG1} + T_{TSEG2} \quad (3.4)$$

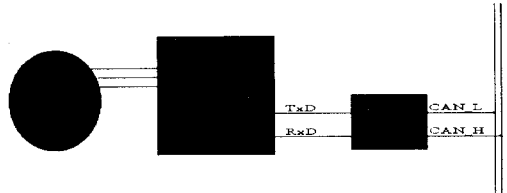
#### 2.1.4 슬레이브 컨트롤러

슬레이브 컨트롤러는 마스터에서 받은 데이터를 처리하고, 명령에 해당하는 데이터를 입/출력 포트를 통하여 명령을 수행한다. 본 시험에서 사용된 슬레이브 컨트롤러

는 크게 P82C150과 PCA82C250으로 구성되어 있다. PCA82C150은 16개의 아날로그/디지털 입력/출력포트가 내장되어 있어 자동차의 아웃사이드 미러의 기능(상·하, 좌·우, 전동접이 기능, 메모리 기능)을 수행할 수 있다. 각각의 SLIO(Serial Linked I/O Device with Digital and Analog Port Function)는 전원이 인가되거나 리셋이 되었을 때 먼저 자신의 확인자를 읽어들이고, 이에 확인자에 따른 명령을 수행할 수 있다. 확인자를 부여하는 것은 하드웨어적으로 설계하여야 한다.

#### 2.1.5 SLIO의 확인자 설정 방법

전원이 인가되거나 리셋이 되었을 때 SLIO(P82C150)는 먼저 POP1P2P3포트를 확인자 프로그램 입력 포트(Identifier program Input Port)로 사용하여[표 2.1]에 나와 있는 값으로 자신의 확인자를 결정한다. SLIO는 4비트를 확인자 입력포트로 사용하므로 최대 16개까지 마스터 컨트롤러에 연결 될 수 있다. 먼저, 입출력 포트가 여분 없이 POP1P2P3포트를 사용하지 않는 경우에는 쉽게 확인자를 인가할 수 있으나, POP1P2P3포트까지 입/출력 포트로 사용 하여야 하는 경우에는 전원이 인가되거나, 리셋되었을 때 POP1P2P3를 확인자 입력포트로 먼저 사용하고, 다시 이 포트들을 입/출력 포트가 되돌려 줄 수 있도록 하드웨어를 설계하여야 한다.



[그림 2.4] 슬레이브 컨트롤러의 구성 블록도

#### 2.1.6 SLIO의 포트 환경설정

SLIO의 포트 설정은 각각의 특성에 맞게 입출력 포트의 기능을 정의 해주어야 한다.

본 실험에서는 SLIO 1번은 사용자의 입력만을 받아들일므로 10개의 포트를 입력포트로 설정하였고, SLIO 3번은 아웃사이드 미러의 거울의 구동과 전동 접이 그리고 열선 기능을 구현하기 위하여 7개의 포트를 출력 포트로 설정하고, 메모리 기능을 위하여 입력포트 2개, ADC(Analog To Digital Conversion)을 위하여 2개의 포트를 사용하였다. 마스터 컨트롤러와 슬레이브 컨트롤러의 초기화 설정이 끝나면, 마스터 컨트롤러인 PCA82C200의 수신 인터럽트(Receive Interrupt)를 활성화 시켜 인터럽트가 걸리면 데이터 필드를 분석하여 정해진 기능을 수행하도록 한다.

#### 2.2 프로그램

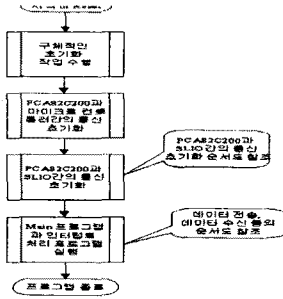
본 실험에서 사용된 소프트웨어의 구조는

1. 마이크로 컨트롤러의 초기화
2. PCA82C200의 환경설정
3. P82C150(SLIO)의 환경설정
4. 인터럽트 처리 수행

으로 4가지로 나눌 수 있다.

이 과정을 간단히 정리하면 먼저 전원이 인가되거나 리셋되면 CPU인 80C32를 초기화하여 모니터링을 위한 RS232C 직렬포트의 속도 설정, 타이머 설정등의 작업을 한다.

그리고, 전체 프로그램을 흐름도로서 나타내면 [그림 2.6]과 같다.

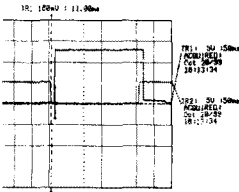


[그림2.5] 프로그램의 전체 흐름도

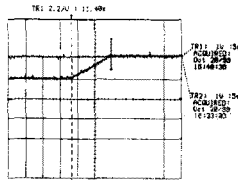
## 2.3 결과

### 2.3.1 실험 결과

본 실험에서는 마스터 CAN컨트롤러에 사용된 인텔사의 80C32 CPU와 IBM호환 기종의 컴퓨터를 RS232직렬 포트를 이용하여 마스터와 슬레이브간의 데이터의 흐름을 모니터링하였으며, 작성된 프로그램은 Micro Series의 icc8051 컴파일러를 사용하여 컴파일 하였다. [그림2.6]의 결과 파형은 사용자가 입력을 주어, SLIO 1번에서 마스터 컨트롤러로 전송하고 마스터 컨트롤러는 이 데이터를 분석하여 SLIO 3번으로 데이터를 전송하여 SLIO 3번에서 모터를 구동 할 때까지의 시간, 즉 입력을 한 시점부터 출력이 나오는 시점까지의 지연시간이다.

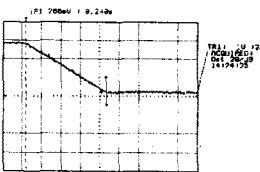


[그림2.6] 입력에서 출력시 까지의 지연시간



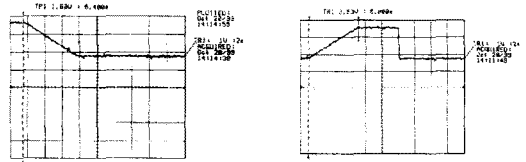
[그림2.7] 기억된 위치를 찾아가는 파형

[그림2.7]은 거울의 위치를 저장한 후 임의의 위치에서 저장된 위치를 찾아 가는 파형이다. 센싱되는 최대값은 상·하 구동시 0xc1~0xc38의 값을 가지며, 좌·우 구동의 센싱 범위는 0xbb~0xc1의 값을 가진다. 이때 0xc1의 전압값은 3.813V 이고, 0xc38의 값은 1.237V가 인가되었다. A/D컨버터의 분해능을 8비트만 사용하여도 약 20mV의 계단값을 가지게 된다. [그림2.8]의 파형은 기준 전압 13.5V에서의 거울의 상하구동에 있어 뒷점에서 맨 아래의 위치 까지의 이동시간을 나타내고 있고, [그림2.9]는 아래의 위치에서 제일 뒷점 까지의 이동 시간을 나타내는 파형이다. 측정 파형은 모터에 부착되어 있는 위치 센서의 값이다. [그림2.10]은 좌측에서 우측으로 거



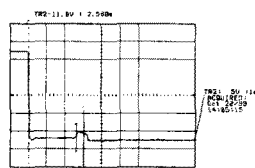
[그림2.8]위에서 아래로 구동 [그림2.9]아래에서 위로 구동시 울을 이동 할 때의 파형이고, [그림2.11]은 반대로 우측에서 좌측으로의 이동 파형이다. 거울을 상·하로 구동하는 시간이 좌·우로 구동하는 시간보다는 평균 2초정

도 더 걸리는데, 이는 아웃사이드 미러의 기구적 특성으로 상하로 움직이는 각이 좌우로 움직이는 각보다 커 이러한 시간 차이가 발생한다.

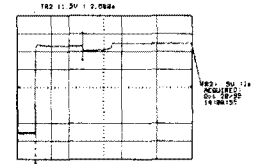


[그림2.10] 우측에서 좌로 구동 [그림2.11]좌에서 우로 구동

[그림2.12]는 아웃사이드 미러를 접을 때의 파형이고, [그림2.13]은 아웃사이드 미러를 펼칠 때의 파형이다. 측정된 전압값은 모터에 인가된 전압을 측정하였다.



[그림2.12] 미러를 접을 때



[그림2.13] 미러를 펼칠 때

## 3. 결론

본 연구에서는 CAN을 사용하여 현재 사용되고 있는 자동차의 아웃사이드 미러에 적용하여 전선의 가닥수를 11가닥에서 4가닥으로 줄여 기존의 기능을 수행 할 수 있는 시스템을 구성하였다. 기존의 시스템에 비하여 멀티 플렉싱 기법을 적용하면 스위치만을 사용하던 기존의 시스템에 비하여 제작 비용이 많이 상승한다. 하지만 기존의 시스템이 가지고 있던 문제점(전선의 수가 증가함으로써 제한된 거울의 하우징(틀)내에 수용하는데 구조적인 어려움이 따른다.)을 해결할 수 있었으며 앞으로 아웃사이드 미러에 새로운 기능을 추가 하더라도 기구적 변경이나 전선 수의 증가 없이 소프트웨어적인 처리 만으로 새로운 기능을 구현할 수 있는 장점을 가지고 있다.

앞으로 CAN을 이용한 자동화 네트워크 구축을 더욱 많은 곳에 적용하기 위해서는 어떠한 경우라도 우선 순위가 낮은 데이터들이 기존 시간내에 전송이 보장될 수 있는 CAN의 응용계층 설계가 앞으로 더 많이 연구 되어야 할 것이다.

### (참 고 문 헌)

- [1] Ken Tindell, Alan Burns, "Guaranteed Message Latencies for Distributed Safety-Critical Hard Real-Time Control Networks", report YCS229, Department of Computer Science, University of York, May 1994
- [2] K.Tindell, A.Burns, and A. Wellings, "Calculating Controller Area Network(CAN) Message Response Times", IFAC woukshop on Distributed Computer Control Systems (DCCS), September 1994
- [3] 하승범, "자동차내의 복잡한 배선문제 해결을 위한 CAN BUS응용에 관한 연구", 서울시립대학교 공학석사 학위 논문 1996
- [4] 조남현, "CAN(Controller Area Network)통신 프로토콜의 계층 분석에 관한 연구", 명지대학교 공학석사 학위 논문 1997
- [5] Gianluca Cena and Adriano Valenzano, "An Improved CAN Fieldbus for Industrial Applications", IEEE Transaction on Industrial Electronics, VOL. 44, NO. 4, pp553-564, August 1997