

홈 네트워크 시스템 환경하에 상호 운용성을 가지는 닥내 제어 네트워크의 통합 방안

김충환*, 황중선**, 송의성**

*고려대학교 컴퓨터과학기술 대학원, **고려대학교 컴퓨터학과
e-mail : drako@korea.ac.kr, {hwang, ussong}@disys.korea.ac.kr

Integration method of inside home control network for interoperability among the control devices using in home network system.

Chung-Whan Kim*, Chong-Sun Hwang**, Ui-Sung Song**

*Graduate School of Computer Science and Technology, Korea University

** Dept. of Computer Science & Engineering, Korea University

요 약

닥내의 제어 네트워크는 냉장고, 세탁기, 전자레인지 같은 가전제품과 전등, 가스 경보기, 보일러 등에서 제품 고유의 기능을 수행하기 위하여 일반적으로 8Bit 이하의 비교적 저 기능의 마이컴을 채용하고 있으며 각각의 통신을 위한 각 기기 고유의 통신 프로토콜을 가지고 있다. 이렇게 각각 다른 통신 방식으로 인하여 실제적으로 하나의 통합된 닥내 제어 네트워크 구성은 매우 어렵다. 이에 기존 제어기기 각각의 프로토콜을 유지한 채 모든 제어기기의 프로토콜을 수용할 수 있는 방법은 새로운 프로토콜을 만들어 기존 기기의 프로토콜을 Wrapping 하는 방법이다. 본 논문에서는 기존 제어기기들의 통신 프로토콜을 Wrapping 할 수 있는 통합 프로토콜을 구현하여 하나의 통일된 제어 네트워크를 구성할 수 있도록 하였다.

1. 서론

홈 네트워크는 가정 내의 정보가전기기를 네트워크로 연결하여 사용자가 기기, 시간, 장소에 구애 받지 않고 서비스를 제공 받을 수 있도록 해준다.

즉, 홈 네트워크란 TV, 냉장고, 에어컨 등 집안의 가전제품과 안방, 부엌, 거실, 현관 등 집안의 각 공간을 네트워크를 통해 연결, 정보를 전달해 휴대전화 등을 통해서도 작동이 가능토록 하는 미래형 가전시스템을 말한다[1].

홈 네트워크를 구성하기 위해서는 정보를 송수신할 수 있는 정보가전이 우선 구축되어야 한다. 최근에는 서로 다른 홈 네트워크 기술과 정보화 기술들을 접목한 다양한 종류의 닥내에 설치될 수 있는 기기들이 선보이고 있다. 하지만 각기 독자적인 네트워크 기술과 정보화 기술을 적용하여 홈 네트워크의 구성이 통일 되지 못하고 있는 것이 현재의 상황이다. 정보가전 기기 업체를 포함한 몇몇 메이저 업체에서는 이러한 통일되지 못한 정보화 기기를 하나로 통합하기 위해 여러 기술들을 선보이고 있지만 각 업체들의 이해관계와 경제성, 성능, 신뢰성의 문제로 통합이 이루어지지 못하고 있다. 그러므로, 이 기종 정보화

기기간의 통합을 위해서는 독립된 모든 정보화 기술을 통합할 수 있는 새로운 기술이 필요하다. 홈 네트워크 구성은 크게 두 가지 형태로 구분된다. 하나는 외부망을 연결하기 위한 닥외 네트워크이고 또 하나는 닥내의 정보기기를 운용하기 위한 닥내 네트워크이다. 특히 홈 네트워크의 핵심인 닥내의 제어 네트워크 통합은 매우 중요하다. 닥내의 제어 네트워크를 구성하기 위해서는 각 제품의 통신 특성에 대한 이해가 필요하다. PC 와 그 주변기기들로 구성되는 Network 나 DTV(Digital TV)를 비롯한 Multi Media 제품은 기본기능을 동작시키기 위하여 고기능의 H/W(Hard Ware) 사양이 채용되며, 대량의 데이터 처리와 고속 통신을 지원하기 위한 규약이 필요하다. 반면에 냉장고, 세탁기, 전자레인지 같은 가전제품과 전등, 가스 경보기, 보일러 등에서는 제품 고유의 기능을 수행하기 위하여 일반적으로 8Bit 이하의 비교적 저 기능의 마이컴을 채용하고 있고, 이러한 제품들로 구성된 Network 에서는 원격 제어나 동작상태의 모니터링 등 소규모의 데이터 전송이 이루어지므로 PC 나 Multi Media 계열의 제품들로 구성되는 Network 와는 달리 최소한의 마이컴 자원을 이용하면서 통신

할 수 있는 규약이 필요하다. 본 논문에서는 PC 나 Multi Media 계열의 Network 이 아닌 맥내에서 사용되는 가전제품 및 맥내 제어기기(조명, 가스경보기 등)의 이 기종 기기간의 제어를 위한 네트워크의 통합 방안을 제안 하고자 한다.

본 논문의 구성은 다음과 같다. 2 장은 관련 기술에 대하여 기술하고 3 장은 맥내 제어네트워크의 통합 방안을 위한 핵심 요소에 대해 기술하고, 4 장은 맥내 제어 네트워크 통합을 위한 구현 그리고 5 장에서는 결론과 향후 과제에 대해 간략하게 기술한다.

2. 관련 기술

국내외적으로 홈 네트워크상에서의 맥내 제어 네트워크를 단일화 하기 위한 여러 가지 방안이 제시되고 있다. 대표적인 국외 기술에는 Object 교환 방식인 LonTalk[2]과 CEBus[3]가 있으며, 국내 기술로는 패킷 교환방식인 HnCP(LnCP)[4][5]와 S-Cube 등이 있다.

3. 맥내 제어 네트워크의 통합을 위한 핵심 요소

3.1 제어 네트워크 통합을 위한 프로토콜 개발

맥내에 들어가는 정보가전의 경우 대개 각각의 통신 프로토콜을 가지고 있다. 하지만 하나의 통일된 제어 네트워크를 위해서는 통합된 통신 프로토콜을 하나의 규격화된 방법으로 개발할 필요가 있다. 하나의 통합된 통신 프로토콜을 구현 하는 데는 크게 두 가지 방법이 있다. 하나는 기존의 모든 제어기기의 통신 프로토콜을 무시하고 새로운 프로토콜을 만들어 모든 제어기기들의 통신 방식이나 프로토콜을 수정하는 방법이고, 또 하나는 기존 제어기기 각각의 프로토콜을 유지한 채 모든 제어기기의 프로토콜을 수용할 수 있는 새로운 프로토콜을 만들어 기존 기기의 프로토콜을 Wrapping 하는 방법이다. 본 논문에서는 기존 제어기기들의 통신 프로토콜을 Wrapping 할 수 있는 통합 프로토콜을 설계하였다.

3.2 제어 네트워크의 통신 구조 설계

제어 네트워크를 구성하기 위해서는 기본적으로 통신을 하기 위한 기기가 수반되어야 한다. 이때 가장 중요한 것이 통신의 구조이다. 기존 제어기기의 통신 프로토콜을 Wrapping 하는 통합 프로토콜을 사용하려면 Wrapping 되어 있는 제어기기의 통신 패킷을 제어기기 바로 전단에서 Unwrapping 하여야만 실제로 제어기기를 제어 할 수 있게 된다. 이를 위하여 제어 네트워크의 통신 구조를 설계하여야만 한다

3.3 제어 네트워크의 통신을 위한 모뎀 구현

실제적인 제어 네트워크를 구현하기 위해서는 물리적 구성이 이루어져야만 한다. 3.1 절과 3.2 절의 사항을 만족할 수 있는 통신 모뎀의 구현이 있어야 실제적인 제어 네트워크를 구현 할 수 있다

4. 맥내 제어 네트워크의 구현

4.1 제어관련 통합 통신 데이터 프레임(CNIDF: Control Network Integration Data Frame)의 구조

CNIDF 는 홈 네트워크 G/W(Gateway)(그림 2 의 홈

네트워크 G/W)와 제어 네트워크 통신 단말기(그림 2 의 Comm Master) 사이의 통신을 위한 데이터 프레임이며 그림 1 과 같은 구조를 가진다.

STX	D_ID	S_ID	CMD	FC	P_No	R_ID	D_Len	DT	CS
-----	------	------	-----	----	------	------	-------	----	----

그림 1. CNIDF 의 구조

- 가. STX : Start Byte ('S') → 1byte
 - 패킷의 Start byte 를 뜻하며, 항상 'S' 값이다.
- 나. D_ID : Destination ID → 1byte
 - Destination 의 Address 를 의미한다.
- 다. S_ID : Source ID → 1byte
 - Source 의 Address 를 의미한다.
- 라. CMD : Command → 1byte
 - 제어 명령이나 상태값 요구명령 등, 제어기기에 대한 명령의 종류를 나타낸다.
- 마. FC : Frame Count → 1byte
 - 다른 Frame 과의 중복방지를 위해 사용된다.
 - 처음 시작은 0x00 에서부터 시작되며 frame 이 바뀔 때 마다 1 씩 증가한다. 증가한 값이 최대 값 255(0xff)보다 클 경우에는 다시 처음(0x00) 부터 시작한다.
 - Master(홈 네트워크 Gateway)에서 제어나 상태값을 요구하지 않았는데도 불구하고, Slave 통신 모뎀의 하부기기에서 이벤트가 발생하였을 경우에 CMD 는 이벤트에 해당하는 값이 되며, FC 는 위와 같이 처음시작은 0x00 에서부터 시작되며 이벤트가 발생할 때마다 1 씩 증가한다. 또한 증가한 값이 최대값 255(0xff)보다 클 경우에는 다시 처음(0x00)부터 시작한다.
 - 패킷의 Number 가 바뀐다고 해서 반드시 FC 도 같이 바뀌지는 않는다. 하나의 Frame 에 패킷의 개수가 복수 개라면 FC 의 값은 변하지 않는다. 즉 패킷의 개수가 아닌 frame 이 변화할 때만 카운트가 증가한다.
- 바. P_No : Packet Number → 1byte
 - 몇 번째 패킷인가를 나타낸다.
 - P_No 는 FC 와는 반대로 값이 감소하며, 그 값이 0x00 이 되었을 때가 마지막 패킷을 의미한다. 예를 들어 하나의 Frame 이 3 개의 패킷으로 구성되었을 경우 FC 의 값은 변화가 없고, P_No 의 값은 2→1→0 로 감소를 한다.
- 사. R_ID : Resource ID → 1byte
 - 통신하는 하부기기(예: 세탁기, 가스차단기)와의 통신방식(RS-232 등) 및 I/O port 의 구분을 나타낸다.
- 아. D_Len : Data Length → 1byte
 - Data field 에서 사용되는 유효한 data 를 나타낸다. 예를 들면, 하부기기로 전송이 되어야 할 데이터의 총 길이가 45 bytes 일 경우 패킷의 개수는 3 개가 된다. Data field 의 Length 는 20byte 로 정해져 있기 때문에 D_Len 은 패킷의 번호에 따라 변하지 않고 하나의 Frame 에서 유효한 총 data 의 길이를 나타낸다. 즉 첫번째, 두번째, 세번째 패킷 에 표현되어야 할 D_Len 은 항상 45(0x2d)

가 된다.

자. DT : Data → 20bytes

- slave 모델에서 하부기기로 전달되어야 할 하부기기의 프로토콜등을 위한 field 이다. 즉 각각의 제어기기 고유의 통신 프로토콜이 저장된다.
- 유효한 data 의 길이가 45 bytes 라고 하면 마지막 3 번째 패킷의 data field 중 나머지 15 bytes 는 0x30 으로 채운다.

차. CS : Checksum → 1byte

- 각 패킷의 에러 검출용이며, D_ID 에서 DT 의 마지막 byte 까지 Exclusive OR 를 한 값이다.

4.2 제어 네트워크의 통신 계층 구조

홈 네트워크 시스템을 위해 본 논문에서 사용되는 제어 네트워크의 통신구조는 그림 2 와 같다. 단 Comm Master 와 Comm Slave 사이의 통신을 위한 Topology 는 사용되는 모델의 종류에 따라 변경될 수 있다.

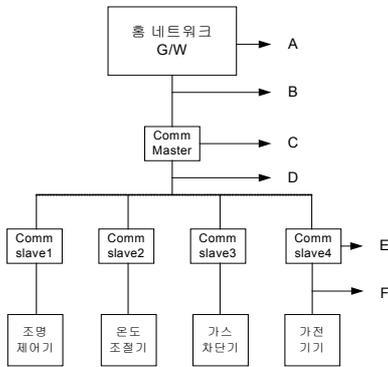


그림 2. 제어 네트워크의 통신 구조

그림 2 의 A 부터 F 에서 사용된 각 장비와 패킷의 특성 및 구조는 표 1 과 같다.

통신 계층	패킷의 특성 및 구조							
A	특성 : 홈 네트워크 G/W에서는 제어기기 각각의 통신 데이터 패킷을 수용할 수 있는 CNIDF를 구현하며, 이 때 CNIDF의 DT field에 응용에 따른 각각의 제어기기 통신 데이터 패킷을 채운다.							
B	특성 : 본 논문에서 사용한 홈 네트워크 G/W와 Comm Master 사이의 통신 규격은 RS-232C, No parity, 8Data bits, 1Stop bit, 19200bps 이며, 통신 패킷은 CNIDF가 된다. 구조 : <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>STX</td><td>D_ID</td><td>S_ID</td><td>.....</td><td>D_Len</td><td>제어기기 각각의 고유 데이터 패킷을 채우기 위한 data field</td><td>CS</td></tr></table> CNIDF	STX	D_ID	S_ID	D_Len	제어기기 각각의 고유 데이터 패킷을 채우기 위한 data field	CS
STX	D_ID	S_ID	D_Len	제어기기 각각의 고유 데이터 패킷을 채우기 위한 data field	CS		
C	특성 : 각각의 전송매체 고유의 통신 패킷의 data field에 CNIDF를 채운다. 예를 들어 전송매체를 LonTalk이나 CEBus의 PLC 모델을 사용한다면 LonTalk이나 CEBus의 User Define 패킷의 data field에 CNIDF를 채운다. 또한 아래의 D계층에서 Date의 충돌을 피할 수 있는 data link계층의 MAC(Medium Access Control) Protocol이 구현되어야 한다. LonTalk 및 CEBus와 같은 별도의 통신 패킷을 사용하지 않을 경우에는 MAC Protocol만을 구현한 Dummy Modem과 같은 형태가 된다.							
D	특성 : 각각의 전송매체별 통신 데이터 패킷을 사용한다. 본 논문에서는 CEBus PLC모델과 RS-485 통신 모델을 사용하였다. 구조 : <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>.....</td><td>.....</td><td>.....</td><td>각 전송매체의 통신 패킷의 data field에 CNIDF를 채운다.</td><td>.....</td></tr></table> CEBus의 User Defined 패킷	각 전송매체의 통신 패킷의 data field에 CNIDF를 채운다.		
.....	각 전송매체의 통신 패킷의 data field에 CNIDF를 채운다.				
E	특성 : 전송매체별 통신 데이터 패킷의 data field에 채워진 CNIDF만을 추출하여 자신에게 해당되는 패킷인지 구별하고, CNIDF의 CMD field의 값에 해당하는 작업을 한다. 또한 최종단 제어기기(세탁기, 조명제어기 등)의 통신 규격에 맞는 데이터를 송수신한다.							
F	특성 : 최종단 제어기기(세탁기, 조명 제어기 등)의 통신 데이터 패킷을 각 기기에 맞는 통신 규격으로 전송된다. 구조 : <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>제어기기 각각의 고유 데이터 패킷</td></tr></table> CNIDF의 제어기기 각각의 고유 데이터 패킷을 채우기 위한 data field 값	제어기기 각각의 고유 데이터 패킷						
제어기기 각각의 고유 데이터 패킷								

표 1. 제어 네트워크의 단계별 통신 패킷 구조

4.3 제어 네트워크의 모델 구현

제어 네트워크를 위한 통신 모델은 홈 네트워크 G/W 와 통신을 주고 받을 수 있는 Master 모델과 하부기기와 데이터를 주고 받을 수 있는 Slave 모델로 구성된다.

4.3.1 CEBus 의 PLC 모델을 이용한 구현

국내에서 생산되고 있는 CEBus 용 PLC 모델을 이용하고, CEBus 의 통신 패킷 중에 User Defined 패킷을 사용하여 그림 2 와 같은 구조 구현이 가능하도록 모델의 Firmware 를 변경하였다.



그림 3. CNIDF 를 위한 CEBus PLC 모델

4.3.2 RS-485 방식을 사용한 통신 모델의 구현

Atmel 사의 89C52rd2 라는 마이컴과 RS-485 통신 Chip 을 사용하여 그림 2 의 D 부분에 해당되는 통신 패킷의 Header 와 Tail 을 제외하고 순수한 CNIDF 패킷을 위한 통신 모델을 구현하였다. 통신시 일어날 수 있는 데이터의 충돌은 CSMA/CA 방식을 사용하여 데이터의 충돌을 방지하였다.

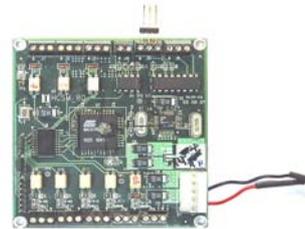


그림 4. CNIDF 를 위한 RS-485 통신 모델

4.4 제어 네트워크의 구현 시험

4.4.1 제어 네트워크 구현 시험을 위한 환경

가. 홈 네트워크 G/W

- OS : Linux
- Kernel Version : 2.4.15

나. 제어기기 : 조명 컨트롤러, 정보가전(세탁기, 에어컨), 가스차단기.

다. 제어 네트워크용 통신 모델

- CEBus PLC 모델
- RS-485 통신 모델

4.4.2 제어 네트워크 구현 시험을 위한 홈 G/W 프로그램 구조.

제안한 통신 프로토콜의 구현 시험을 위하여 홈 네트워크 G/W 용 프로그램을 개발하였다. G/W 프로그램의 구조는 UI 프로세스, 시그널 처리 프로세스, 제어 프로세스로 구분되며, 프로세스간 Interface(IPC : Inter Process Communication) 를 위해서 Signal 과 공유메모리를 사용하였으며 그 구조와 동작 방식은 다음과 같다.

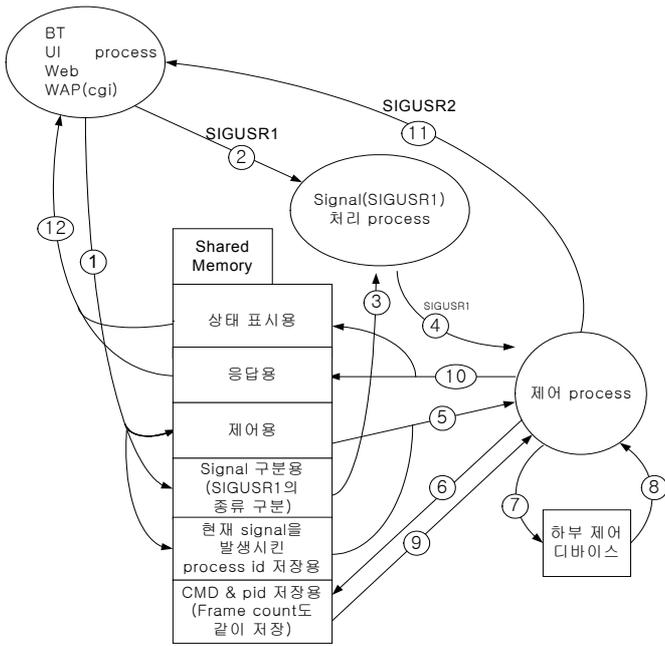


그림 5. 홈네트워크 G/W 프로그램의 구조

- ① 제어용 공유메모리에 제어명령에 해당되는 값을 저장하고, Signal 구분용 공유메모리에는 제어명령 수행 요구를 나타내는 값을 저장하고, Signal(SIGUSR1)을 발생시킬 process의 pid (Process ID) 값은 현재 signal을 발생시킨 pid 저장용 공유메모리에 BT/UI/Web process의 값을 저장한다.
- ② 명령 수행을 요구하는 Process(상위 프로그램:GUI 등)가 Signal 처리 process에 signal(SIGUSR1)을 발생시킨다.
- ③ 상위 프로그램으로부터 발생된 SIGUSR1에 의해 Signal 처리 process가 wake up 되어지고, Signal 구분용 공유메모리의 값을 읽는다.
- ④ Signal 구분용 공유메모리의 값에 따라 해당되는 process 에 SIGUSR1 을 발생시키거나, 해당 함수를 호출하여 일을 수행한다.
- ⑤ Signal 처리 process로부터 SIGUSR1을 받은 제어 process가 제어용 공유메모리에서 수행해야 될 명령값과 명령을 내린 pid 읽어온다.
- ⑥ 5번 과정에서 읽어 온 명령에 따라 frame을 만들고 frame을 생성할 때 생긴 frame count를 CMD, pid 와 함께 CMD & pid 저장용 shm(Shared Memory) 에 저장한다.
- ⑦ 제어용 공유메모리에서 읽은 명령값대로 명령을 수행한다.
- ⑧ 하부 제어 디바이스로부터 제어명령에 대한 응답값 또는 이벤트 값을 받는다.
- ⑨ 8번 과정에서 들어온 Frame count에 해당하는 CMD 와 pid를 CMD & pid shm에서 읽어온다.
- ⑩ 9번 과정에서 읽은 값을 응답용 shm에 저장을 하는 것과 동시에, 상태 표시용 shm에도 명령에 의해 처리되어진 값들을 저장한다.
- ⑪ 9번 과정에서 읽은 pid에 SIGUSR2를 발생시킨다.
- ⑫ SIGUSR2를 받은 process는 응답용 shm에서 이전에 자신이 내린 명령의 내용을 확인하고 상태 표시용 shm의 값을 읽은 후 다음 작업을 수행한다.

4.4.3 제어 프로세스의 구조

제어 프로세스는 통신 패킷 처리 과정, 제어 명령 처리 과정, 제어 패킷 생성 과정으로 구성된다. 그 중 제어 명령 처리 과정 알고리즘은 다음과 같다.

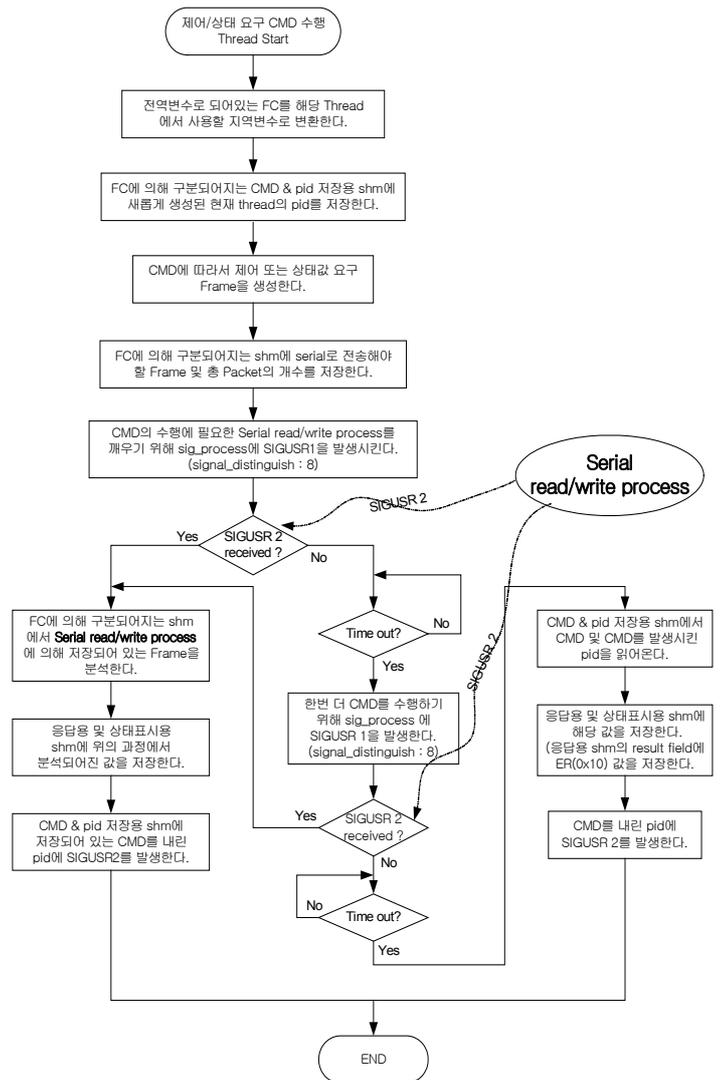


그림 6. 제어 명령처리 과정 알고리즘

5. 결론

제안된 맥내 제어 네트워크의 구조는 상호 운용성에 중점을 둔 통합 통신 프로토콜을 구현함으로써 맥내에서 사용되는 제어기기들의 각각 독립된 통신 프로토콜을 사용하면서 발생하는 통신상의 문제점을 해결할 수 있었다.

위와 같은 방식으로 제어 네트워크를 구성하였을 경우의 단점은 PnP(Plug and Play) 기능을 구현하는데 제한적이라는 것이다. 모든 제어기기들이 동일한 구조의 통신 프로토콜을 가지고 있지 않기 때문에 PnP 를 위한 기능 구현이 최종단(제어기기)에서는 불가능하다. 하지만 최종단에 부착될 통신 모델을 이용한다면 완벽하지는 않지만 PnP 기능의 일부를 수용할 수 있다. 향후에는 PnP 기능을 구현하여 제어 네트워크의 운용상 편리함과 안정성을 확대시켜야 한다.

참고문헌

- [1] <http://kin.naver.com/open100/entry.php?docid=178468>
- [2] LonTalk Protocol Specification - EIA/CEA 709.1-B-2002
- [3] ANSI/EIA-600 CEBus Standard Specification, EIA,1998
- [4] http://icat2.snu.ac.kr/ilab/paper/dist_file/1042002-08-ljm.pdf
- [5] LnCP Protocol Specification Version 1.15. LG Electronics, 2002