

## 충돌검지 다중접속 프로토콜에 의한 2선식 RS232C 직렬통신 인터페이스 구현

(Implementation of two wire RS232C Serial Communication Interface using CSMA protocol)

한경호\* · 최천원

(Kyong - Ho Han · Cheon - Won Choi)

### 요 약

본 논문에서는 충돌검지 다중 접속 프로토콜을 적용하여 기존의 3선식 RS232C 직렬 통신 포트를 2 선식 공통 통신채널에 접속하여 다중접속을 가능하게 하는 인터페이스의 구현을 다루었다. 기존의 3선식 RS232C 직렬 통신 포트의 하드웨어의 변경 없이 송신데이터와 수신데이터를 공통데이터 선으로 결합하는 인터페이스회로에 의하여 2 선으로 구성하였다. 2선식 연결방법에 의하여 다수의 호스트가 공통선로에 연결되는 경우 2개 이상의 호스트가 동시에 데이터를 전송할 경우 데이터의 충돌이 발생한다. 이를 감지하기 위하여 전송 데이터를 궤환하여 비교하고 전송데이터의 왜곡 여부로 데이터 충돌을 검지한다. 데이터의 재전송시 충돌 확률을 낮추기 위하여 여러 가지 재 전송 모델링을 제시하였으며 각 방법에 대한 실험결과를 통하여 성능을 비교하였다. 본 연구의 결과는 기존의 RS232C 기기의 회로를 거의 변경하지 않고 다중접속이 가능하게 함으로 기존의 각종기기 및 설비를 단일 통신 선로에 의한 네트워크에 연결하는 데 응용될 수 있다.

### Abstract

In this paper, we implemented 2-wire (data, gnd) RS232C serial communication interface by applying the Collision Sensing Multiple Access(CSMA) Protocol. The transmitting and receiving wires of RS232C port are connected together by the interface circuit forming data wire without hardware modification. On two-wired common channel connection, simultaneous transmission from more than two hosts causes data collision. The collision is detected by loop-back test of transmission data comparing with the fed back data to detect the data distortion. Various models are adopted to reduce the probability of retransmitted data collision and experiments show the performance of each cases. Due to allowing multiple hosts to be accessed through the common communication channel with minimum circuit addition, the result of this paper can be easily applied for conventional RS232C instruments and machines to connect to the single data communication line.

**Key Words:** RS232C, CSMA, Common channel communications, Loop-back test

\* 주저자 : 단국대학교 전기제어시스템공학부 교수  
Tel : 02-709-2831, Fax : 02-793-1387  
E-mail : kyongghan@dku.edu

접수일자 : 2002년 12월13일  
1차심사 : 2002년 12월26일  
심사완료 : 2003년 1월21일

## 1. 서 론

RS232C는 중대형 컴퓨터 및 마이크로프로세서의 입출력 장치로 대부분의 컴퓨터, 그리고 대부분의 자동화 장치 및 측정 기기에도 장치되어 근거리 통신용으로 많이 활용되고 있다. 기존의 시스템 구성은 제어 호스트에서 입, 출력 제어장치까지 연결방법이 1 대 1 구성으로 다수의 제어 장치와 통신을 구현하기 위해서는 제어 장치 갯수의 통신 선로 수가 필요하다. 또한 통신 전송을 위하여 제어 호스트가 먼저 전송을 시작하고 제어장치가 이에 응답하는 폴링(Polling) 방식만을 허용함으로 호스트는 모든 제어장치를 주기적으로 폴링 함으로 통신채널의 효율성이 낮게되는 경우도 있다. 이에 본 논문은 RS232C 포트를 갖는 다수의 제어 장치를 단일 공통 선로에 접속하기 위하여 기존의 세 선(송신, 수신, 접지)으로 구현되었던 RS232C 직렬포트를 두 선(데이터, 접지)으로 변환하는 인터페이스를 구현하였으며 호스트 시스템과 장치간에 폴링방식과 함께, 제어 장치가 먼저 호스트에 전송할 수 있도록 하기 위한 전송제어 방식을 충돌검지 다중접속(Collision Sensing Multiple Access: CSMA) 프로토콜을 적용하여 연구하였다. 이러한 충돌검지 프로토콜에 의한 다중접속 직렬통신 시스템은 기존의 RS232C포트를 가지는 각종 제어 장치 와 호스트의 하드웨어를 변경하지 않고 간단한 인터페이스만 연결하여 단일 공통 통신 채널에 다중 접속하도록 함으로 통신 선로수를 훨씬 줄일 수 있으며 통신망의 구성에서 호스트 및 제어 장치의 추가, 제거 및 위치 변경 등이 용이하도록 할 수 있어 호스트 간의 트래픽이 많지 않은 경우 빌딩 및 공장 등의 자동화 영역에 응용될 수 있다.

## 2. 본 론

### 2.1 충돌검지 다중접속(CSMA) 프로토콜

충돌검지 다중접속 프로토콜은 ANSI/IEEE 802.3에서 규정된 Ethernet 접속 방법 등, 다중접속시 가장 많이 사용되는 매체 액세스 제어 기법으로, 두 개 또는 그 이상의 호스트가 하나의 공통 전송매체를

공유하는 수단에 의해 액세스가 가능하도록 하는 기법이다[1][2]. 또한 이 방식으로 매체에서 각각의 호스트에서 보낸 두 개 또는 그 이상의 데이터 패킷이 충돌할 때 각 호스트는 데이터 패킷의 충돌을 검지하고 적당한 시간 후에 데이터 패킷을 재 전송하여 데이터 충돌의 가능성을 낮추는 충돌검출기능을 갖는 반송파감지 다중 액세스 방법이다. 이때 전송 매체로는 동축케이블(coaxial cable pairs, CP) 또는 꼬인 회전쌍(twisted pairs, TP)을 주로 사용한다[3]. 이러한 프로토콜은 다중 구조를 가지며 이를 기존의 직렬 통신 시스템에 구현하기 위하여 하드웨어 및 소프트웨어를 대폭 변경해야 한다.

### 2.2 충돌검지에 의한 2선식 직렬통신 다중 접속

기존의 RS232C 직렬통신에서는 송신, 수신, 접지의 세 개의 선로를 사용한다. 이를 데이터와 접지의 두 개의 선로를 사용하기 위하여 직렬포트를 변경하여야하나 기존의 각종 자동화 장비의 하드웨어는 변경이 극히 곤란하므로 기존의 하드웨어를 직접 변경하지 않고 그림 1 과 같이 간단한 접속장치를 구성하여 3선식 RS232C 포트를 2선식 선로에 연결하였다. 그림 2 에서는 각 포트에 연결되는 장치의 회로를 보인다.

기존의 송신, 수신 신호는 공통으로 연결하여 하나의 데이터 신호선로로 접속하였다[4].

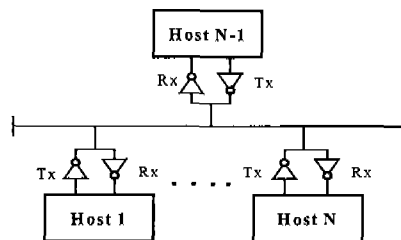


그림 1. RS232C포트의 공통채널 접속  
Fig. 1. Common channel interface for RS232C

이 경우, 단일 통신채널에 다수의 호스트가 전송할 경우, 주 호스트가 지정되고 나머지 호스트는 종속적 호스트가 되며 항상 주 호스트가 종속적 호스

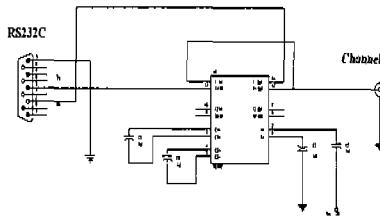


그림 2. RS232C포트의 접속 회로  
Fig. 2. Interface circuit for RS232C

트를 지정하여 전송을 시작하고 종속적 호스트들은 수신기능만 수행한다. 주 호스트의 전송이 완료되면 지정된 종속 호스트는 데이터 전송을 수행함으로 단일 통신 채널에 의한 호스트간의 데이터 전송이 종료된다. 이는 해당 통신망에 연결되어 있는 호스트 가운데 주 호스트가 항상 전송을 시작하는 방식으로 폴링(polling) 방식이다. 각종 자동화 네트워크에서 장치들의 데이터 및 이벤트의 처리를 하나의 호스트 컴퓨터가 제어하게 되면 호스트 컴퓨터의 폴링횟수가 늘어나며 폴링 횟수대미 실시 데이터 처리 비율이 낮아지며 이는 해당 네트워크의 효율성을 떨어뜨리는 결과를 초래한다. 또한 주 호스트에 장애가 발생할 경우 네트워크의 중단을 초래하게된다.

본 논문에서는 이를 보완하여 2선식 공통 채널에 접속된 모든 호스트가 동등한 우선권을 가지며 데이터 전송을 시작할 수 있도록 하기 위하여 각 호스트는 데이터 패킷을 RS232C 포트의 송신단자를 통해 공통 통신선로로 전송되며 동시에 수신단자로 전송된 패킷이 궤환 된다. 이에 송신 호스트에서는 수신 단자로 궤환된 패킷이 전송한 패킷과 같은지 비교함으로써 통신선로에서 충돌여부를 검사한다. 이와 같이 직렬 통신 선로로 전송한 데이터 패킷에 충돌이 없음을 판별하면 다음 데이터 패킷을 위와 같은 방법으로 전송하게 된다. 이때, 궤환 되어 수신 단자로 받은 패킷이 전송한 패킷과 다른 경우 매체에서 다른 호스트 컴퓨터에서 보낸 데이터 패킷과 충돌이 있는 경우이므로 이런 때는 충돌을 회피하기 위하여 임의의 시간만큼 지연한 다음, 충돌된 데이터 패킷을 다시 전송하게 된다.

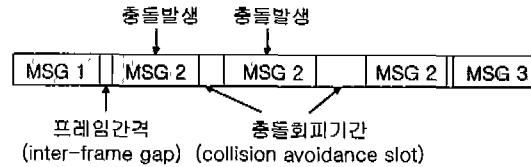


그림 3. 전송데이터 충돌발생과 재전송  
Fig. 3. Transmission data collision and retransmission

데이터 패킷이 충돌한 경우 패킷과 다음 전송 패킷사이의 최소한의 시간 간격(Inter-frame gap: IFG)을 조정함으로써 재전송시 다른 호스트의 전송 시작 시점이 다르게 될 가능성이 크게되면 재전송 패킷이 다시 충돌할 가능성은 적게된다. 충돌발생 후 재전송 때까지의 지연시간을 결정하는 재전송모드, 데이터 패킷 길이 등 여러 경우들의 조건을 달리 하여 패킷 전송속도(단위시간당 전송한 데이터 패킷 양), 패킷 전송 성공률 및 패킷 수신 오류율 등을 실험하였다.

위에서 기술한 재전송모드는 데이터 패킷 전송 중 충돌발생 시 이전 데이터 패킷을 재전송하기 위해 지연시간을 결정하는 모드로 본 논문에서는 표 1과 같이 세 가지의 재전송 모드를 설정하였다.

재전송모드1은 충돌 발생 시 초기지연시간을 2배씩 증가하면서 증가된 범위 내에서 난수를 발생하여 재전송 시간을 결정한다. 재전송모드2는 재전송모드1에서 사용한 IFG 값을 사용하여 발생한 난수를 이용한 지연시간을 추가하였다. 이렇게 난수 범위를 늘림으로 재전송 시 충돌 확률을 줄이도록 하였다. 마지막으로 재전송모드3은 재전송횟수와 무관하게 동일한 초기지연시간을 시료로 하여 난수 값을 결정하였다.

단일 통신채널을 통하여 다수의 호스트를 접속할 경우 각 호스트는 고유번호(ID)에 의하여 자신의 데이터 패킷을 수신하게 된다. 또한 데이터 패킷의 시작과 끝을 HEADER와 TAIL 바이트로 나타낸다[5]. HEADER와 TAIL도 역시 바이트 단위로 구성되며 데이터 코드 중에는 HEADER 및 TAIL 과 같은 패턴을 가질 수도 있다. 이 경우 HEADER 및 TAIL과 같은 패턴의 데이터는 동일한 바이트를 추가하여

충돌검지 다중접속 프로토콜에 의한 2선식 RS232C 직렬통신 인터페이스 구현

HEADER 및 TAIL과 구분한다. 수신시에는 이를 역순으로 하여 데이터중 HEADER와 TAIL과 같은 패턴을 갖는 데이터가 중복될 경우 하나를 삭제하는 방식으로 원 데이터를 복구한다.

표 1. 모드별 재전송 지연시간  
Table 1. Retransmission delay modes

모드	재전송지연시간(ms)
1	random (초기지연시간×2×재전송횟수)
2	random(IFG)+random (초기지연시간×2×재전송횟수)
3	random (초기지연시간)

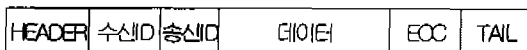


그림 4. 데이터 패킷의 구성  
Fig. 4. Data packet structure

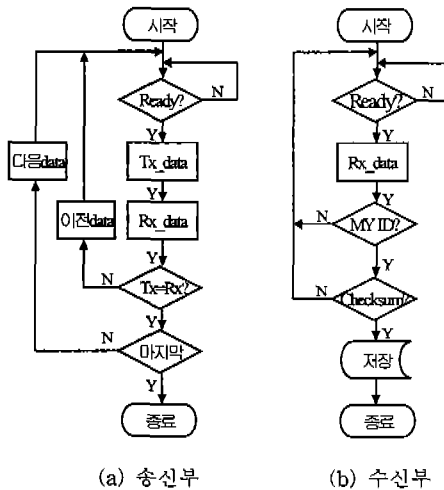


그림 5. 데이터 전송 제어프로그램 흐름도  
Fig. 5. Data transfer control flowchart

이는 Ethernet의 비트추가(Bit Stuffing)와 유사한 방법인 바이트 추가(Byte Stuffing) 방법을 사용하였다. 따라서 HEADER 및 TAIL과 같은 패턴의 바이트가 단독일 경우 그 바이트는 HEADER 및 TAIL를 의미한다. ECC부분은 오류검출부호(Error

Checking Code)로 CRC-16 코드가 사용되었다[5].

이와 같이 하여 충돌검지 방식에 의한 2선식 직렬 포트의 다중접속이 이루어지며 기존의 각 호스트의 RS232C 포트제어 프로그램에 위의 제어프로그램을 추가함으로써 구현될 수 있다[6]. 그림 5에서 패킷의 전송과 수신 각 기능에 대한 제어 프로그램에 흐름도를 보인다.

### 3. 시스템의 접속성능평가

#### 3.1 평가 시스템 구성

본 논문에서 제안한 접속 방식의 접속 성능 평가를 위하여 그림 6 과 같이 여러 가지 성능의 PC의 직렬 포트에 접속인터페이스 회로를 연결하여 동축 단일 선로에 접속하여 구성하고 단위시간 (1분)동안 여러 가지 전송 조건에 대하여 세 가지 재전송 모드에 대한 접속 성능을 실험으로 확인하였다[7]. 접속 호스트의 수는 여러 가지로 하여 실험하였다. 그림 7 과 그림 8 은 실험용으로 제작된 4개의 직렬포트를 지원하는 회로 및 이를 직렬포트에 연결한 모습을 보인다. 접속 성능 평가 요소는 패킷 전송속도, 패킷 전송 성공률, 패킷 수신 오류율을 중심으로 평가하였다. 패킷 전송 속도는 단위 시간동안 전송한 총 패킷의 양을 나타내며 전송도중 충돌한 패킷도 포함되어 있다.

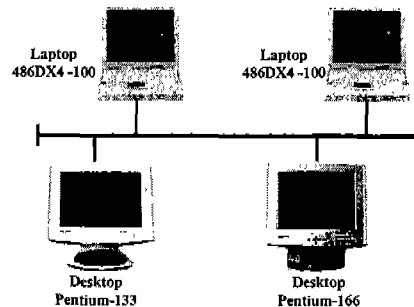


그림 6. 성능 평가 시스템 구성  
Fig. 6. Performance evaluation system

패킷 전송 성공률은 전송한 총 패킷 중 충돌 없이 전송이 완료된 패킷의 백분을 값이며, 패킷 수신 오

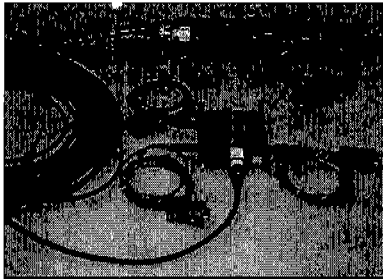


그림 7. 다중포트 실험 시작품  
Fig 7. Multiport experimental kit

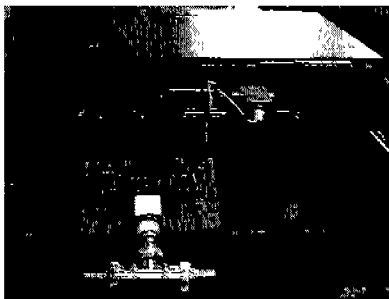


그림 8. 직렬포트에 연결한 모습  
Fig 8. Conenction to serial port

류율은 수신 호스트에서 데이터 패킷을 패킷 순서대로 받는 도중 누락된 데이터 패킷이 발생 시 증가하는 누락 데이터의 양을 총 수신된 패킷 양의 백분율로 표현하였다.

표 2에서 여러 가지의 전송속도 및 세 가지 재 전송 모드에 따른 접속 성능 평가 요소를 보이고 있다.

표 2. 속도 및 재전송모드에 따른 성능평가요소  
Table 2. Performance evaluation factors for speeds and retransmission modes

비교 항목	모드별 전송 속도	4800 bps	9600 bps	38400 bps	57600 bps	115200 bps
전송속도 [frame/min]	모드1	280.47	433.79	736.97	745.59	749.86
	모드2	266.06	416.52	587.56	592.75	596.78
	모드3	272.26	372.20	459.57	463.15	461.37
전송 성공률 [%]	모드1	71.560	77.139	98.086	98.657	98.938
	모드2	68.916	74.895	97.362	99.247	99.475
	모드3	57.009	78.952	90.847	96.189	97.667
수신 에러율 [%]	모드1	4.711	2.466	0.227	0.125	0.079
	모드2	4.442	3.550	0.167	0.082	0.035
	모드3	5.404	2.037	1.046	0.306	0.197

### 3.2 전송속도에 대한 성능평가

표 2의 결과를 그림으로 표시하면 그림 9와 같이 된다. 표 2에서 볼 수 있듯이 전송속도가 증가할수록 동일한 길이의 패킷을 송신하는데 걸리는 시간이 단축되므로 단위시간당 보낼 수 있는 패킷 양이 많아지게 된다. 이에 패킷 전송 속도가 증가함을 알 수 있다. 이 때 전송 패킷 양은 전송이 성공된 패킷 및 전송 중 충돌된 패킷 모두를 포함하고 있다.

패킷 전송 성공률은 패킷 전송 속도가 빨라질수록 전송 도중 충돌이 발생할 확률도 낮아지므로 패킷 전송 성공률이 높아짐을 그림 9를 통해 보인다.

패킷 수신 에러는 송신 호스트에서 데이터 패킷을 전송 후 송신측 호스트에서는 케환 검사로 전송 상태가 이상이 없음을 인식하였지만 수신측 호스트에서는 수신된 데이터 패킷에 이상이 있다고 식별될 때 발생하는 에러로 통신 채널에서의 노이즈 및 잦은 충돌로 인한 패킷 수신 작업에서의 오동작에 의해 발생된다. 수신 오류율은 패킷 전송성공률과 반비례 관계로 나타나는데, 이는 충돌횟수가 많을수록 수신 오류율도 증가함을 볼 수 있다.

재전송지연시간이 짧아지면 잦은 충돌발생으로 재전송 대기시간 동안 대기하는 횟수가 줄어들므로 패킷전송 속도가 가장 느리게 나타남을 볼 수 있다. 이에 재전송지연시간은 재전송횟수와 비례하여 증가하는 것이 더 높은 전송 성공률을 가짐을 알 수 있다.

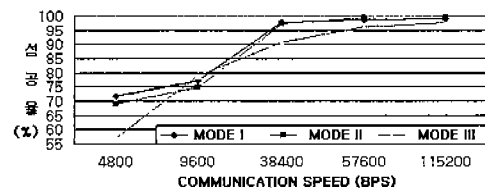


그림 9. 전송속도와 성능  
Fig. 9. Transmission speed and performance

전송속도가 57600[bps] 이상의 경우에는 패킷 전송 성공률 및 수신 오류율이 재전송모드에 별 영향을 받지 않음을 볼 수 있다. 이에 57600[bps]이상의 전송속도에서는 재전송 지연시간을 표 2의 재전송을 모드 1과 같이 하여 패킷 전송 속도를 높이는

방법이 더 효율적임을 알 수 있다.

### 3.3 패킷 지연시간에 대한 성능평가

그림 10 은 패킷간 지연시간 변화에 따른 각 재전송 모드별 패킷 전송 성공률을 나타내고 있다.

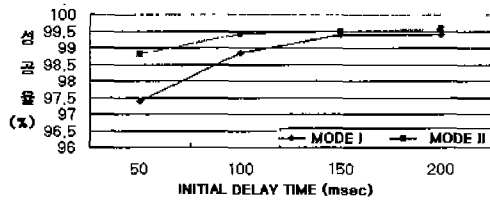


그림 10. 패킷간 지연시간(IFG)과 성능  
Fig. 10. IFG and performance

패킷간 지연시간은 송신 호스트에서 패킷 전송 후 다음 패킷 전송 때까지의 지연시간 및 충돌 후 재전송지연시간을 결정하는 기본 값으로 설정된다. 그림 8에서 볼 수 있듯이 패킷간 지연시간이 증가할수록 패킷 전송 속도는 느려지며 단위 시간당 보낼 수 있는 패킷 양이 적어진다. 반면 패킷간 지연시간이 증가할수록 충돌 발생률 및 패킷 수신 오류율은 감소하였다.

### 3.4 패킷길이에 대한 성능평가

그림11 은 데이터 패킷 길이 변화에 따른 재전송 모드별 패킷 전송 성공률을 보이고 있다. 패킷의 데이터 크기가 같은 조건에서 재전송 때까지의 지연시간이 길어질수록 패킷 전송 속도가 느려짐을 보였다.

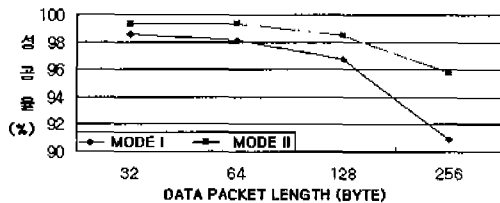


그림 11. 데이터 패킷 길이와 성능  
Fig. 11. Packet length and performance

또한 패킷의 길이가 늘어날수록 패킷 전송 기간 중 충돌이 일어날 확률이 증가하므로 패킷 전송 성

공률이 감소하며 같은 데이터 패킷 길이 조건에서는 재전송 지연시간이 길어질수록 전송 성공률이 증가함을 볼 수 있었다. 따라서 그림 11 에서는 패킷의 길이가 늘어날수록 충돌 발생률이 증가하므로 수신 오류율도 함께 증가하며 재전송지연시간이 짧을수록 수신 오류율이 증가하여 결국 전송 성공률은 감소함을 보였다.

### 3.5 접속호스트 수에 대한 성능평가

마지막으로 제시한 방법으로 공통선 채널에 접속한 호스트의 수가 증가할 경우의 전송 성공률을 그림 12 에서 보이고 있다.

이 경우에 채널에 PC를 3대부터 7대까지 증설하면서 각 호스트의 전송 성공률을 보였다. 실험에 사용된 PC 기종으로는 P-166, 486DX4-100, 486DX2-33, 486-25, P-MMX266, 386DX-40, P-133을 각각 1대씩 사용하였다[8][9]. 그림 10에서 볼 수 있듯이 연결한 호스트의 수가 많아질수록 공통 채널에서의 패킷 충돌횟수가 증가함으로 패킷 전송 성공률이 저하되며 수신 오류율은 증가되어 패킷의 재전송이 증가함을 볼 수 있었다.

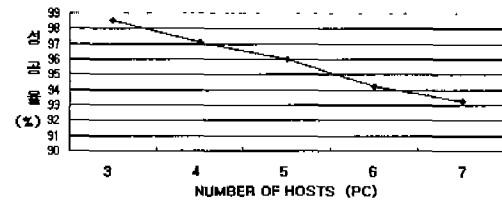


그림 12. 접속 호스트 수와 성능  
Fig. 12. Number of hosts and performance

이와 같이 접속 성능 평가를 통하여 전송 속도가 높을수록, 초기 지연시간이 길수록, 데이터 패킷의 길이가 짧을수록 그리고 연결된 호스트의 수가 작을수록 전송 패킷의 충돌 확률이 낮아지므로 전송 성공률이 높아짐을 알 수 있었으며 다양한 기종 및 여러 대의 호스트를 공통 통신채널에 연결하여 다중접속을 구현함으로써 간단한 하드웨어 접속정치와 제어 프로그램의 추가로 직렬통신을 통한 다중접속 구현이 가능함을 보였다.

#### 4. 결 론

본 논문에서는 기존의 RS232C 프로토콜을 이용하여 1 대 1 접속만을 허용하던 시스템에 시스템 하드웨어를 변경하지 않고, 다수의 호스트를 공통 통신 채널에 접속할 수 있는 다중접속 구현 방법을 제안하였다. 이를 위하여 기존의 3 선 식 직렬통신 선로를 2 선으로 연결하는 하드웨어 회로와 함께, 충돌검지 다중접속 프로토콜 제어 방법을 제시하여 공통 통신선로를 이용한 직렬통신을 구현하였다. 공통 통신선로에 연결되는 다수의 호스트는 데이터 패킷에 고유 ID를 사용하여 패킷을 구분한다. 본 논문에서 제안된 접속 방법의 성능 평가를 전송속도, 패킷간 지연시간, 패킷의 길이 그리고 연결된 호스트의 수에 따라 단위시간당 전송 성공률을 측정함으로써 접속 성능을 확인하였다.

본 연구의 결과는 기존의 RS232C를 사용하는 시스템 및 장비에 응용하면 공통 통신 선을 이용할 수 있으므로 선로의 수가 크게 감소되며 특히 트래픽이 크지 않은 네트워크에 제어 장치 및 호스트의 증설 및 이동 설치 시 전송 케이블의 변경이 필요 없는 유연성 있으며 Ethernet 등의 프로토콜에 비하여 훨씬 간단하고 저 비용의 접속 장치에 응용 될 수 있다.

앞으로는 통신 중 발생하는 패킷 수신 오류율을 개선하고 누락된 패킷 검출 시 자동반복요구 기능을 추가해 누락된 패킷을 재 전송함으로써 보다 안정적인 통신이 구현될 수 있을 것으로 기대된다.

#### References

- [1] ANSI/IEEE, "ANSI/IEEE 802.3, Information technology Local and metropolitan area network, Part3 : Carrier Sense Multiple Access with Collision Detection(CSMA/CD) access method and physical layer specifications", IEEE, 1993.
- [2] Heinz-Gerd Hegering and Alfred Lapple, "Ethernet Building Communications Infrastructure", Addison Wesley, 1993.
- [3] Green, "Data Communications", Longman Scientific & Technical, 1992.
- [4] William Stallings, "Data and computer communications", 회중당, 1987.
- [5] Stephen B. Wicker, "Error Control Systems for Digital Communication and Storage", Printce Hall, 1991.

- [6] 이육양, "C 프로그래머를 위한 시리얼 커뮤니케이션", 가남사, 1990.
- [7] 김석주, "C로 하드웨어 주무르기", 가메출판사, 1997.
- [8] 편집부, "RS232C 인터페이스 응용사례집", 가남사, 1991.
- [9] 이소베도시오, "C언어와 RS-232C/CP-1B", 다다미디어, 1993

#### ◇ 저자소개 ◇

##### 안경호(韓敬浩)

1959년 6월 25일생. 1982년 서울대학교 공과대학 전자공학과 졸업. 1984년 서울대학교 대학원 전자공학과 졸업(석사). 1992년 미국 Texas A&M University 졸업(박사). 1993년 ~ 현재 단국대학교 교수.

##### 최천원(崔天源)

1964년 2월 23일생. 1986년 서울대학교 공과대학 전자공학과 졸업. 1988년 서울대학교 대학원 전자공학과 졸업(석사). 1996년 미국 University of California at Los Angeles 졸업(박사). 1997년 ~ 현재 단국대학교 교수.