

# 충돌검지 다중접속(CSMA) 프로토콜에 의한 RS232C 직렬통신의 구현

곽희수, 한경호  
단국대학교 전기공학과

## Implementation of RS232C Serial Communication by CSMA protocol

heesoo Kwak, kyongho Han  
Dept. of Electrical Engineering, Dankook University

### Abstract

In this paper, we implemented RS232C serial communication by the 2 wires(Data,GND), and multiple access, by the applying Collision Sensing Multiple Access(CSMA) Protocol. Multiple access is implemented by assigning a unique ID to each controller. The multiple access control operation starts by sending a command packet from a host to another host and the command packet is composed of ID bytes of source and target host computer, data bytes and the check sum, byte. In host computer, after sending command packet, the collision from loop back data. If collision is detected, it means a command packet was collided with another command packet for another host. The packet communication of the controller enables the multiple access of the controller through the common serial data link. The application of this serial communication through CSMA protocol and the usage of two wires, have an effect on saving the wires and convenient of layout work.

의해 액세스가 가능하다. 또한 이 방식으로 매체에서 각각의 호스트에서 보낸 두 개 또는 그 이상의 데이터 패킷이 충돌할 때 각 호스트에서는 CSMA 프로토콜로부터 충돌을 검지하고 적당한 시간 후에 데이터 패킷을 재전송하는 즉, 충돌검출기능을 갖는 반송파감지 다중 액세스 방법이다.

이러한 CSMA의 전송매체로는 동축케이블(coaxial cable pairs,CP) 또는 꼬인 회전쌍(twisted pairs, TP)를 주로 사용한다.

(그림 1)은 CSMA 방법의 액세스를 위한 버스/트리 구조의 매체액세스제어 기법을 나타낸 것이다.

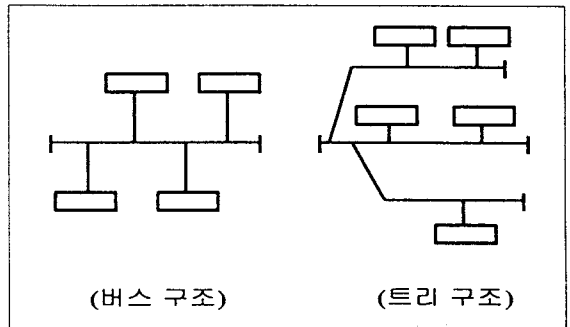


그림 1 버스 및 트리 구조

## 1. 서 론

RS232C는 최근 퍼스널 컴퓨터 및 소형 마이크로 컴퓨터의 입출력 장치로 대부분의 컴퓨터에 장치되어 근거리 통신용으로 많이 활용되고 있다. 하지만 기존의 시스템 구성은 단일 제어 시스템에서 입,출력 제어 시스템까지 통신 선로를 1대 1로 설치하여 통신을 구현하는 방법으로 다수의 제어기와 통신을 구현하기 위해서는 통신 선로 수가 많이 필요하게 된다. 이에 본 논문은 통신 선로 수가 많이 필요한 이러한 단점을 보완하기 위해 기존의 세 선(Rx,Tx,GND)으로 구현 되었던 RS-232-C 직렬 통신 방식을 두 선(Data,GND)으로 구현 되도록 하였으며, 이를 위해 충돌검지 다중접속(Collision Sensing Multiple Access)프로토콜을 적용하였다.

CSMA 프로토콜 방식에 의한 데이터 패킷의 구조는 ANSI/IEEE 802.3에서 규정된 구조를 따르며, 일반적으로 패킷의 첫 부분을 나타내는 Preamble, 데이터 패킷을 수신할 대상인 호스트의 주소(Destination Address), 데이터 패킷을 보내는 송신부의 주소(Source Address), 패킷에 있는 데이터의 Type을 명시한 Type Field, 전송하는 데이터 패킷 중 실제 데이터 부분인 Data Field, CRC(Cycle Redundancy Check) 연산의[2] Source가 되는 오류검사부호(Checksum) 등으로 구성되어 있다.

(그림 2)는 일반적인 CSMA 프로토콜의 패킷 구조를 나타낸 것이다.

이러한 충돌검지 다중접속(Collision Sensing Multiple Access) 프로토콜에 의한 직렬통신 시스템은 서로 다른 다수의 호스트를 공통 통신 채널에 다중 접속함으로써 통신 선로 수를 훨씬 줄일 수 있으며 선로의 중설 및 변경이 용이하고 전체 시스템의 비용 절감에도 큰 기여를 한다.

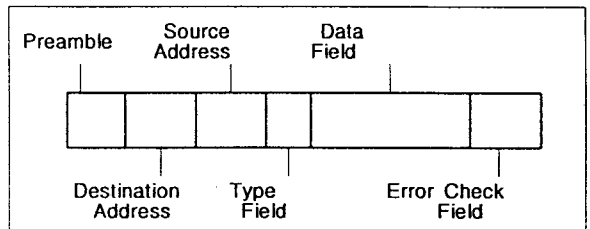


그림 2 CSMA 프로토콜의 패킷 구조

## 2. 본 론

### 2.1 충돌검지 다중접속(CSMA) 프로토콜

CSMA/CD(Carrer Sense Multiple Access with Collision Detection)는 ANSI/IEEE 802.3[1]에서 규정된 다중접속시 가장 많이 사용되는 버스/트리 구조의 매체액세스제어 기법으로, 이것은 두 개 또는 그 이상의 호스트가 하나의 공통 전송매체를 공유하는 수단에

이러한 충돌검지 다중접속(CSMA) 프로토콜은 하나의 통신 선로로 다수의 호스트 컴퓨터를 접속, 상호간의 통신을 가능케 하므로 선로의 수를 크게 줄이며 또한 다

른 입출력 설비의 증설 및 변경시 기존의 통신 선로를 그대로 사용할 수 있으므로 설치비 및 유지, 보수비를 크게 줄일 수 있다.

### 2.2 CSMA에 의한 직렬통신 제어방법

충돌검지 다중접속(CSMA) 프로토콜에 의한 RS232C 직렬 통신을 구현하기 위해서 우선 기존의 세 선이던 RS232C 직렬 통신 선로(Rx,Tx,GND)를 두 선으로 변형하여 구현하도록 하였다. 즉 (그림 3)과 같이 기존의 Rx,Tx 두 선로를 하나의 Data선으로 바꿈으로 총 두 선(Data,GND)으로 직렬 통신 선로를 구성하였다.

각 호스트 컴퓨터에서는 프로그램에 의해 제작된 데이터 패킷은 RS232C의 Tx단자를 통해 공통 통신선으로 전송되며 동시에 Rx단자로 전송된 패킷이 Loop back 된다. 이에 송신 호스트에서는 Rx단자로 Loop back된 패킷이 Tx 단자로 전송한 패킷과 같은가를 비교함으로써 매체(통신 선로)에서의 충돌을 검사하게 된다. 이처럼 직렬 통신 선로로 전송한 데이터 패킷에 충돌이 없음을 판별하면 다음 데이터 패킷을 위와 같은 방법으로 전송하게 된다. 만약 Rx 단자로 받은 패킷이 Tx 단자로 전송한 패킷과 다른 경우 매체에서 다른 호스트 컴퓨터에서 보낸 데이터 패킷과 충돌이 있는 경우이므로 이런 때는 일정시간 후에 충돌된 데이터 패킷을 재전송하게 된다. 이렇게 함으로써 송신 시 통신 선로에서의 충돌을 검지하게 된다. (그림 3)은 두 선(Data,GND)에 의한 다중접속 직렬 통신을 구현하기 위한 각 컴퓨터의 입출력 장치에 연결한 장치의 그림이다.

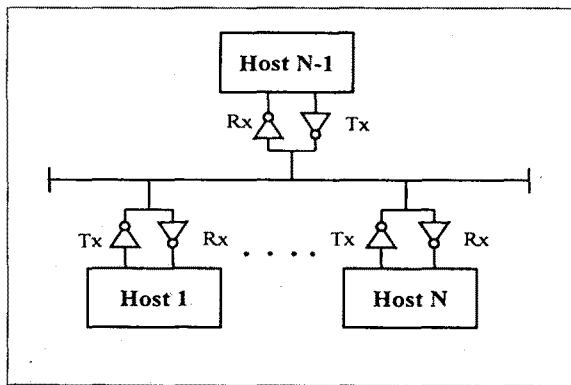


그림 3. 공통 통신 선로 연결도

(그림 3) 처럼 Tx단자로 전송된 신호는 공통 통신 선로를 통해 각 호스트 컴퓨터로 연결되며 또한 송신 컴퓨터의 Rx단자로 직접 연결됨으로 충돌검지를 위한 Loop back 신호를 제공하게 된다.

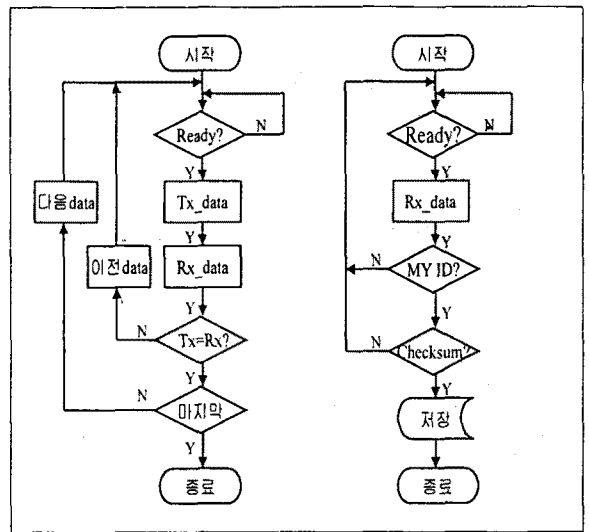
(그림 4)는 통신 선로에서의 충돌검지를 구현하기 위한 내부 프로그램의 일부를 나타내고 있다.

```

int Compare( unsigned char A[BULENGTH],
             unsigned char B[BULENGTH] )
{
    int i,Er=0;
    for(i=0;i<BULENGTH;i++)
    {
        if(A[i]!=B[i])
            Er++;
    }
    return(Er); // Er : Total Collided byte(s)
}
    
```

그림 4. 충돌검지 프로그램

위 함수에서 A[BULENGTH]는 Tx단자로 전송되는 데이터 패킷의 Array를 나타내며 B[BULENGTH]는 Rx단자



(a) 송신부

(b) 수신부

그림 5 충돌검지를 위한 송신부 및 수신부의 흐름도

로 Loop back되는 데이터 패킷을 나타낸다. 위 함수를 통해 전송된 데이터 패킷과 Loop back된 데이터 패킷은 한 Byte씩 비교된다. 만약 도중에 두 Byte(A[i],B[i])가 다를 경우 'Er'이라는 변수를 1씩 가산하여 Error의 개수를 표시하게 된다. 이처럼 데이터 패킷의 총 길이 만큼 각 패킷의 Array를 비교하고 나면 Error 개수를 위 함수에 return 시켜줌으로 통신선로에서의 충돌을 알 수 있게 해 준다. 이와 같이 두 패킷을 비교한 값이 '0'인 경우 공통 통신 선로를 통한 데이터 패킷 전송이 완료되었음을 나타내고, 두 패킷을 비교한 값이 '0'이 아닌 경우는 공통 통신 선로에서 다른 데이터 패킷과 충돌이 있음을 알려 줌으로 다음 번 전송시기에 데이터 패킷을 재전송할 것을 알려주게 된다.

(그림 5)는 충돌검지를 위한 각 호스트에서의 송수신 흐름도를 나타내고 있다. 송신부에서는 호스트와 연결된 공통 선로가 대기 상태인가를 check하여 다른 호스트에서 패킷을 전송하지 않을 때, 전송하고자하는 패킷을 Tx 단자로 전송하고 Rx 단자로 들어온 Loop back 데이터를 비교한다. 두 패킷이 동일하면 위와 같은 방법으로 다음 데이터 패킷 전송을 시도한다. 만약 충돌에 의해 두 패킷이 같지 않을 때에는 이전 데이터 패킷을 다음 번 전송 시기에서 재 전송하도록 한다. 수신부에서는 공통 통신 선로로부터 데이터 패킷의 전송이 있을 시 Rx 단자로 데이터를 받고 자신에게 온 패킷인지를 검사한다. 이렇게 ID를 검색하여 자신의 ID와 같은 경우 패킷에서의 충돌 유무를 가리기 위해 수신된 패킷 데이터로부터 오류검사(checksum)를 실시한다. 충돌이 없음을 판별하면 수신된 데이터 및 명령어를 저장 또는 수행한다. 만약 데이터 패킷에 충돌에 의해 오류가 있음을 판별되면 수신된 데이터는 무시하고 다음 번에 데이터를 수신하도록 한다.

이렇게 기존의 세 선이던 직렬 통신 선로를 두 선으로 줄이고 충돌검지 다중접속(CSMA) 프로토콜을 이용함으로써 직렬통신을 제어할 수 있게 된다.

### 2.3 CSMA에 의한 직렬통신 패킷 전송 구현

공통 통신 선로에 여러개의 제어기를 연결하여 서로 통신을 구현하기 위해서는 CSMA 프로토콜에 의해 각 호스트에서는 일정한 규약을 갖는 데이터 패킷으로 송수신을 구현하여야 한다. 이에 본 논문에서는 (그림 6)와 같은 패킷 구조로 CSMA에 의한 직렬 통신을 구현하였다.

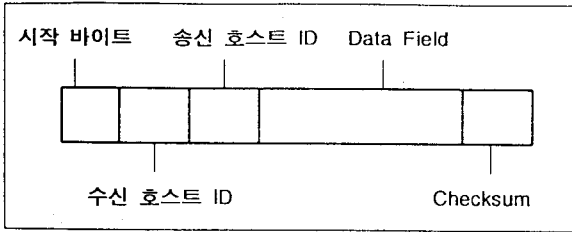


그림 6. 데이터 패킷 구조

호스트 컴퓨터에서 해당 제어기로 전송하는 데이터 패킷은 (그림 6)와 같이 패킷의 시작을 나타내는 시작 바이트, 수신 호스트 ID(Destination Address) 바이트, 송신 호스트 ID(Source Address) 바이트, 데이터 바이트, 오류검사부호(checksum) 바이트로 구성된다.

시작 바이트는 최상위 비트(MSB)가 '1'로 1 바이트 크기를 가지며 데이터 패킷의 시작을 나타낸다. 수신 및 송신 호스트의 ID는 최상위 비트(MSB)가 '0'로 2바이트로 구성되며 16,384개의 고유 주소를 갖고 있다. 이러한 고유 주소는 각 호스트 주소, 공통 주소 및 그룹 주소 등으로 분할하여 사용한다. 데이터 바이트는 전송하고자 하는 명령어 및 정보를 싣는 부분으로 24 바이트의 크기를 갖고 있다. 오류검사부호(checksum) 바이트는 앞의 바이트, 즉, 시작 바이트, 송신 호스트 ID 바이트, 데이터 바이트들의 합과 같으며 전송 시 충돌을 검지하는데 사용된다.

패킷을 전송하고자 하는 호스트에서는 자신의 주소(송신 호스트 ID), 전송하고자 하는 상대 호스트의 주소(수신 호스트 ID), 전송하고자 하는 데이터 및 오류검사부호(checksum)를 위와 같은 패킷 구조로 작성한 후, 호스트 컴퓨터의 RS232C Tx 단자를 통해 통신 선로로 데이터 패킷을 전송한다. 이 때 패킷을 전송하는 호스트 컴퓨터의 RS232C Rx 단자에는 Tx 단자를 통해 전송된 데이터 패킷이 Loop back 된다. 이렇게 Loop back된 데이터 패킷은 Tx 단자로 전송된 데이터 패킷과 같은지를 검사하게 된다. 만약 전송된 데이터 패킷과 Loop back된 데이터 패킷이 다르다면 통신 채널에서 다른 호스트에서 전송한 패킷과 충돌이 일어난 것으로 이전 데이터를 다음 번 전송 시기에 재전송 하도록 한다.

호스트 컴퓨터가 위와 같이 데이터 패킷을 통신 선로를 통해 전송하면 각 호스트에서는 전송된 패킷에서 수신 호스트 ID 바이트를 검색하여 자신에게 해당된 주소(자신의 호스트 주소, 공통 주소, 그룹 주소)인지를 판단하고 해당된 주소이면 다음 바이트들을 받아들인다. 이처럼 자신에게 전송된 패킷을 모두 받으면 오류검사부호(checksum) 바이트를 제외한 나머지 바이트를 모두 더한 후, 하위 8비트 데이터만을 취해 오류검사부호(checksum) 바이트와 같은지를 판별한다. 이렇게 전송된 패킷에 오류가 있는지를 검사하여 전송된 패킷에 오류가 없으면 전송된 데이터를 수신하고, 오류가 있음이 판단하면 전송된 패킷의 데이터를 무시하고 다음 번에 수신하도록 한다.

(그림 7)은 이러한 충돌검지 다중접속(CSMA) 프로토콜에 의한 패킷 전송을 구현한 결과 화면으로 공통 통신 선로로 데이터 패킷을 전송 후 Loop back check(위 그림에서 'Checksum=>' 값이 '1'이면 에러가 없음을 나타낸다)를 통해 충돌여부를 검사하고, 통신 선로에서 패킷이 충돌이 있을 시 다음 전송 시기에 이전 데이터를 재전송하고 있음을 볼 수 있다. 또한 자신에게 해당된 주소를 실은 패킷을 수신할 때('C\_RX\_ID=>' 값이 주소 바이트의 길이가 될 때 : 본 프로그램에서는 주소 바이트를 2바이트로 설정)는 오류검사(checksum)를 통해 데이터의 저장 및 명령어 수행 여부를 검색하고, 전송된 패킷이 다른 호스트로의 전송 패킷일 경우에는

호스트 내부에서의 데이터 저장 및 명령어 수행 작업을 하지 않음으로 공통 통신 선로를 통한 다중접속이 구현됨을 볼 수 있다.

```

[시작]
Loop back Check[Tx=Rx ?]=> Okay
Another Computers are Communicating Now! 'SM' host -> 'FQ' host
Check_RX_ID=>0 Checksum=>1
Another Computers are Communicating Now! 'TC' host -> 'FQ' host
Check_RX_ID=>0 Checksum=>1
Another Computers are Communicating Now! 'TC' host -> 'FQ' host
Check_RX_ID=>0 Checksum=>1
Received DATA from 'GW' Computer
C_RX_ID=>2 D_Checksum=>1
Test Times= 82 Tx_DATA=>[ 75] Error= 9
Loop back Check[Tx=Rx ?]=> Okay
Received DATA from 'GW' Computer
C_RX_ID=>2 D_Checksum=>1
Test Times= 83 Tx_DATA=>[ 75] Error= 9
Loop back Check[Tx=Rx ?]=> Okay
Received DATA from 'GW' Computer
C_RX_ID=>2 D_Checksum=>1
Test Times= 84 Tx_DATA=>[ 74] Error= 9
Loop back Check[Tx=Rx ?]=> Okay
Error DATA=> R [*****]*****
Test Times= 85 Tx_DATA=>[ 75] Error= 10
Test Times= 86 Tx_DATA=>[ 75] Error= 10
Loop back Check[Tx=Rx ?]=> Okay
Test Times= 87 Tx_DATA=>[ 76] Error= 10
Loop back Check[Tx=Rx ?]=> Okay
  
```

그림 7 다중접속에 의한 직렬통신 구현 화면

## 결론

기존의 세 선(Tx,Rx,GND)이던 직렬통신 선로를 두 선(Data,GND)으로 줄이고, 충돌검지 다중접속 프로토콜을 이용함으로써 공통 통신 선로를 이용한 직렬통신을 구현하였다. 공통 통신 선로에 여러 개의 제어기를 연결하고 각 제어기에는 서로 다른 ID를 부여함으로써 패킷 전송 시 해당된 제어기에서만 패킷을 받도록 함으로 공통 통신 선로를 이용한 다중 접속을 구현 하였고, 기존의 직렬 통신 선로로 쓰이던 Tx,Rx,GND 세 선을 Loop back check를 통한 충돌검지 시스템을 첨가함으로써 Data,GND 두 선 만으로도 RS232C 직렬 통신이 가능하도록 구현하였다.

이와 같은 직렬통신 방법은 통신 선로의 수를 줄일 수 있으며 공통통신 선로를 이용하여 다수의 제어기를 연결할 수 있으므로 설치 및 증설시 간편함은 물론 이에 따른 비용절감의 효과를 가져온다.

## [참고 문헌]

- [1] ANSI/IEEE,ANSI/IEEE 802.3, Information technology Local and metropolitan area network,Part3 :Carrier Sense Multiplu Access wiyh Collision Detection(CSMA/CD) access method and physical layer specifications,1993.
- [2] 이우양, "C 프로그래머를 위한 시리얼 커뮤니케이션",가남사,1990.