

간이 다중 시리얼 분배기 설계

오세호*, 류형선, 최현영, 이성진, 박정균, 김양모
충남대학교 전기공학과

A Study on design of simple serial distributor for multiport

Oh Se-Ho*, Ryu Hyoung-sun, Choi Hyun-young, Lee Sung-Jin, Park Jung-Gyun, Kim Yang-Mo
Dept. of Electrical Engineering, Chungnam national University

Abstract - In this paper, we designed simple serial distributor for multiport which is imposed the primitive ranking with a periodic time of received multipath data

The received data are transmitted from manifold multi-device system to decoder and on data in the multiplexer which is to transmit to PC is selected from binary counter.

1. 서 론

마이크로프로세서는 주변장치를 통해 외부와 정보를 교환할 수 있는데 일반적으로 병렬통신과 직렬통신에 의해 이루어진다.

컴퓨터내의 장치와 정보교환을 할 때는 통상적으로 고속의 통신속도를 필요로 하여 한꺼번에 많은 정보를 처리할 수 있는 병렬통신 방식을 주로 쓴다. 이는 대량의 정보를 빠른 시간에 한꺼번에 처리함으로써 컴퓨터의 성능을 향상시킬 수 있기 때문이다.

그렇지만 병렬 통신 방식은 어떤 대상에든 다 사용할 수 있는 것은 아니다. 그것은 통신거리의 제한성, 구현상의 기술적인 어려움과 비용이 너무 비싸다는 데 있다. 또한 어플리케이션 자체가 고속의 통신속도를 필요로 하지 않을 경우도 많다. 그러므로 컴퓨터가 외부와 통신할 때는 직렬통신 방식을 많이 사용한다. 그런데 직렬통신은 1:1 통신을 기본으로 하기 때문에 1:다수 통신을 하기 위해서는 특별한 장치가 필요하다. 그것이 바로 다수의 사용자를 연결할 수 있는 다중 통신 접속 장치로 대표적인 것이 비동기식 시리얼 통신포트와 LAN이다.

이 중 비동기식 통신포트를 사용하여 호스트와 원격지에 있는 다수의 사용자를 연결시켜주는 장치가 멀티포트이다. 멀티디바이스 시스템은 주로 자동화 시스템에서 이용하는데 중앙에 설치한 한 대의 컴퓨터에 여러 대의 비동기식 장치를 연결하여 장치별로 발생하는 데이터를 실시간으로 수집/분배하거나, 원격지에 있는 각종 장치를 제어한다.

본 논문은 이와 같이 다수의 비동기식 장치에서 발생하는 데이터를 실시간으로 수집하는 멀티 디바이스가 여러 개가 있을 때 이것을 다시 모아서 PC쪽으로 보내는 간이 시리얼 분배기를 설계하고자 한다.

멀티 디바이스에서 수집하는 데이터가 다수의 종류일때 그 데이터의 성질이 다르므로 데이터를 받아들이는 주기 또한 다르게 생각을 해야 한다. 빠른 변화를 보이는 데이터에 대해서는 데이터 취득 주기도 짧아야 하고 느린 변화를 보이는 데이터에 대해서는 큰 주기를 가지고 데이터를 받아 들여도 문제가 되지 않을 것이다. 그래서 본 논문에서는 분배기 사용자의 의지에 따라 한주기 동안 각각의 멀티디바이스에 대한 데이터 수집 횟수를 정할 수 있도록 하여 시스템의 효율을 높이고자 한다.

2. 시스템 설계

본 논문에서 설계하게 될 다중 시리얼 분배기는 여러 멀티 디바이스에서 오는 데이터를 하나의 PC로 전달하기 위한 것으로서 데이터 취득의 우선 순위를 두어 자주 취해야 하는 것과 일정한 간격을 두고 취해도 상관없이 없는 장치가 있을 경우를 고려하여 설계 했다.

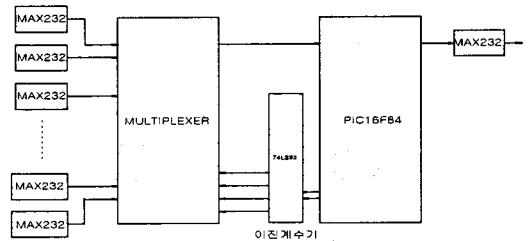


그림 1 다중 시리얼 분배기 전체 구성도

그림 1에서 보는 시스템은 본 논문에서 구성한 전체 시스템의 블록도이다. 다수의 멀티 디바이스 시스템에서 들어오는 데이터를 MAX232로 받아들여 Multiplexer를 사용해서 어떤 데이터를 받아 들일지 정한 후 이것을 다시 MAX232를 사용해서 PC로 보내는 과정을 보이고 있다.

2.1 RS-232C의 콘넥터와 신호선의 접속

RS-232C에 표준으로 사용되는 콘넥터는 DB-25와 DB-9가 있다. DB-25는 25핀 콘넥터로 사실상 사용하지 않는 핀이 많고 부피가 커서 불편하다. 그래서 간편한 9핀 D형 콘넥터를 사용했다.

신호선의 접속도 No Handshaking, Pseudo Handshaking, Handshaking 방법등이 있는데 본 논문에서 설계하게 될 분배기는 하드웨어나 소프트웨어적으로 회선의 상태를 체크하지 않고 단순하게 데이터의 송수신기능만을 수행하기 때문에 No Handshaking 방법을 쓴다. No Handshaking 방법은 Tx D Rx D GND 세선의 연결만으로도 문제없이 통신이 수행되는 방법으로 대부분의 제어용 컴퓨터에서는 시스템을 간단히 하기 위하여 이 방법을 주로 사용한다. 그림 2는 No Handshaking 방법을 보인 것으로 수신측과 송신측의 Rx D와 Tx D가 서로 엇갈려서 연결되고 GND가 서로 연결이 된 것을 볼 수 있다.

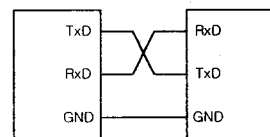


그림 2 No Handshaking 방법

RS-232C 신호 레벨을 보면 먼저 송신측에서는 TTL 신호를 '0'(space) 상태인 경우는 +15 ~ +3 '1'(mark) 상태는 -3 ~ -15V로 표현한다. 이에 비하여 수신측은 입력신호의 감도가 -3[V] 이하이면 '1'(mark) 상태로 인식되고 +3V 이상이면 '0'(space) 상태로 인식해서 다시 TTL 신호로 나오게 된다.[4]

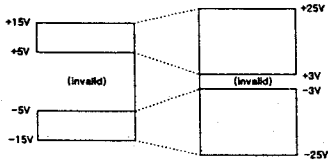


그림 3. RS-232C 신호레벨

그림 3은 RS-232C 통신 방식에서의 전압레벨 및 잡음여유를 보여주고 있다. 이와 같은 전압레벨 변환을 위해서 1개의 소자만으로도 RS-232C 직렬 통신 신호레벨 변환회로를 구성할 수 있는 MAXIM사의 MAX232 소자를 썼다. MAX232는 송수신 변환회로가 각각 2개씩 내장 되어 있어서 위에서 말한 No Handshaking 방식으로 결선을 할 경우 통신채널을 2개까지 사용할 수 있다. 본 시스템에서는 최대 16개의 포트를 지원하므로 8개와 PC 쪽으로 보내는 1개를 포함 9개의 MAX232가 필요하겠다.

2.2 아날로그 멀티플렉서 설계

본 논문의 다중 시리얼 분배기는 다수의 멀티 디바이스에서 들어오는 데이터를 가중치를 두어 순차적으로 받아들이는 것으로써 많은 시리얼 분배기가 그렇듯이 이것 역시 멀티플렉서를 이용했다. 멀티플렉서는 16채널이 지원되는 소자를 이용했다. 멀티플렉서의 운용은 4bit의 컨트롤 신호로 가능한데 이를 위해서 본 시스템에서는 4bit binary counter를 이용했다. [2]

그림 4는 Multiplexer의 실제 회로를 보이고 있다.

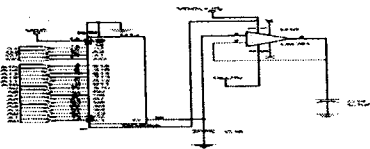


그림 4. multiplexer 회로도

2.3 4-bit binary counter

Multiplexer를 제어하기 위해서 4-bit binary counter를 사용했다. 4-bit binary counter는 하나의 클럭에 1 count가 증가한다. 처음 0000부터 1111까지 카운트가 되어 Multiplexer를 운영한다. 카운터는 클럭의 'H'에서 'L'로의 천이(하강 에지)에 의해 레벨이 변화한다.

그림 5는 4-bit binary counter 74LS93의 실제 회로를 보이고 있다. 74LS93은 2입력 게이트의 마스터 리세트(클리어)가 있는데 이것을 이용해서 Multiplexer를 Reset시킨다.

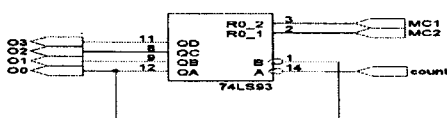


그림 5. 4-bit binary counter (74LS93)

2.4 컨트롤러 회로의 설계

분배기의 수가 적고 그 데이터 량이 적었다면 컨트롤러를 쓰지 않았겠지만 본 논문의 경우는 여러 곳에서 오는 다수의 데이터를 받아야 하기 때문에 컨트롤러를 썼다.

컨트롤러는 미국 마이크로칩테크놀로지사의 PIC시리즈 중에서 PIC16F84를 이용했는데 이 컨트롤러는 소형이면서 주위에 전원과 크리스탈 발진자만 접속하면 LED를 직접 구동할 수 있는 입출력 포트를 개별로 프로그램 제어할 수 있다. 또한 EEPROM을 가지고 있어서 다시 프로그램을 고칠 수 있다. 본 논문에서 소개하고 있는 간이 다중 시리얼 분배기는 사용자의 의지에 따라 멀티 디바이스의 데이터 취득 횟수를 변경할 수 있어야 하기 때문에 EEPROM을 가지고 있는 컨트롤러를 선택했다.

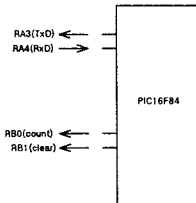


그림 6. PIC16F84 제어신호

그림 6에는 PIC16F84에서 제어신호로 사용하는 모든 핀을 보이고 있다. RA4는 Multiplexer에서 오는 신호를 수신하는 RxD핀이다. 수신된 데이터는 제어 알고리즘에 의해 RA3 TxD핀으로 송신이 된다. PIC16F84에서는 시리얼 통신의 경우 언제나 송신 핀으로 RA3를 쓴다. 그리고 RB0은 4-bit binary counter에 클럭을 주는 핀이다. 이 신호로 인해서 counter는 1이 증가하게 되고 그러면 Multiplexer에 연결된 다음 멀티 디바이스의 신호가 들어오게 된다. 그리고 RB1은 4-bit binary counter의 clear 핀으로 쓴다. clear 신호를 주면 counter는 '0000'으로 되고 Multiplexer는 처음 A0의 신호를 들여보내게 된다.

2.5 제어 알고리즘

다중 시리얼 분배기는 멀티 디바이스가 여러개 있을 때 가중치를 주어 데이터를 받는 순서를 결정하는 역할을 한다. 데이터에는 자주 받아 들여야 하는 것이 있고 가끔씩 받아 들여도 상관없는 것도 있다. 다시 말하면 데이터의 종류에 따라 어떤 장소의 데이터를 취득하느냐에 따라 그 샘플링 시간이 정해지게 된다. 가끔 받아 들여도 상관없는 것까지 순차적으로 받아들이기 필요한 것은 아니다. 그것을 받아들이는 시간에 데이터 변화율이 큰 것을 받아 시스템의 효율을 높일 수 있다. 센서를 예로 들면 거리 센서와 같은 경우는 변화하는율이 크기 때문에 자주 받아들여야 하는 데이터이고 온도센서의 경우 온도는 거리처럼 그 변화율이 크지 않기 때문에 가끔만 확인을 해 주면 되는 것이다. 여기서 가끔이라는 말을 썼지만 그것은 1-2초 정도의 주기이다. 자연 현상에서 온도가 1-2초 사이에 급작스럽게 변하는 경우는 극히 드물 것이다.

처음 시작 부분에서 4-bit binary counter에 의해 멀티플렉서에 0000의 에드레싱 신호가 들어가면서 첫 번째 데이터가(A0 controller) 들어온다. 데이터는 앞에서 말한 것처럼 스타트 비트 1bit와 데이터비트 8bit 그리고 정지 비트 1bit로 되어있다. 데이터를 받았으면 이 데이터를 MAX232를 통해서 PC로 보내게 된다. 그런 다음 count1이 '0'인지를 판단해서 A의 데이터가 다 들어 왔는지를 확인한다. 만약 '0'이 아니라면 다시 data receive 로 가서 다음 data를 받는다. 다시 data를 보낸 후에는 counter1이 '0'인지를 판단한다. 만약

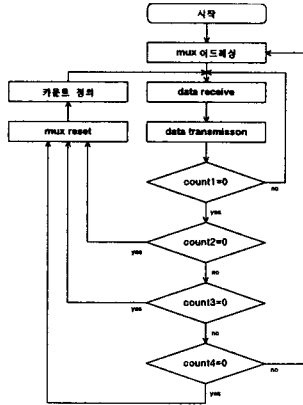


그림 7. 제어 알고리즘

counter1이 '0' 이라면 이때 부터는 Multiplexer를 운영하여 다음 데이터를 받아 들여야 한다. Multiplexer의 운영은 순차적으로 다음 신호를 받아들이는 것과 RESET을 시키는 두 가지로 볼 수 있다. 순차적인 경우는 4-bit binary counter에 하나의 클럭을 주어 다음으로 넘어 갈 수 있다. Reset을 시키는 경우는 4-bit binary Counter의 Clear 신호를 이용하여 Multiplexer에 '0000' 신호를 넣어 줌으로써 다시 처음으로 돌아가는 것이다.

Counter2에서부터 그 뒤에 오는 나머지 Counter는 임의의 숫자를 넣을 수 있는 버퍼 개념으로 자신이 어떠한 주기로 데이터를 받을지를 결정하는 것이 되겠다.

만약 A에서 E까지 5개의 멀티 디바이스에서 오는 데이터를 받아 들여야 한다고 가정을 하자. 그리고 한 주기 동안 데이터를 A와 B는 3번 C와 D는 2번 마지막 E는 1번을 받아들이고 싶다면 처음에 Count 정의부에서 counter4에 5를 넣고 나머지(Counter1은 제외) Counter에는 5보다 큰 수를 넣는다. Counter2와 Counter3에 6을 넣었다고 생각할 하자. 그리고 각 Counter를 체크하기 전에 이 수를 1씩 감산을 한다.

처음 시작을 하면 A 데이터가 들어온다. Counter1이 '0'이면 다음에는 Counter2를 1감산하고 0인지를 확인한다. 0이 아니면 다음에는 Counter3을 1감산하고 0인지를 확인한다. 마찬가지로 0이 아니면 마지막으로 Counter4를 1감산하고 0인지를 판단한다. 만약 0이 아니라면 4-bit binary Counter에 1클럭을 보내어 다음 B 데이터를 받아들이게 된다. 이런 과정을 5번을 거치면 A~E까지 받아들이게 된다. 그러면 counter4는 0이 되고 그러면 4-bit binary Counter에 Clear 신호를 주어 Multiplexer를 Reset 시킨다. 그리고 다시 Count 정의부에서 다음 Counter를 위한 정의를 한다. Counter3은 C와 D를 두 번 체크하기 위한 것으로써 전에 E를 한번체크 하기 위한 과정에서 벌써 1번을 받아들인 상태이다. 그러면 1번만 더 받아들이면 되므로 Counter3에는 4가 들어가게 된다. 그리고 나머지 Counter에는 4보다 큰 수가 들어가면 된다. Counter2와 4에 5를 넣었다고 하면 처음 Multiplexer가 Reset된 상태이므로 A 데이터부터 들어올 것이다. 그리고 위의 과정을 되풀이하면 제일 먼저 Counter3이 0이 될 것이다. 그러면 다시 Multiplexer를 Reset 시키고 다음 A와 B를 위한 Count 정의부에 들어간다. 이 경우는 위에 말한 것과 같은 방법으로 Counter2에는 2를 넣고 나머지는 이것보다 큰 수를 넣는다. 그러면 Counter2가 제일 먼저 0이 될 것이고 그러면 Multiplexer를 Reset 시킨 다음 E를 위한 Count 정의부로 가서 Counter4에 5를 넣고 나머지는 이보다

큰 수를 넣고 data receive로 간다. 이러한 과정을 되풀이하면 한 주기 동안 A와 B는 3번을 받아들인 것이 되고 C와 D는 2번 E는 1번을 받아들인 결과를 낸다.

2.6 실험 결과

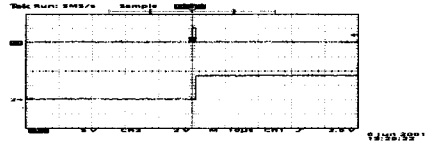


그림8. PIC16F84의 count 신호

그림 8은 PIC16F84의 RB0에서 나오는 count 신호와 4bit binary counter의 O0의 신호이다. 이다. 그림에서와 같이 2μs의 count 신호로써 H에서 L로 천이가 될 때 counter의 신호가 바뀌는 것을 볼 수 있다.

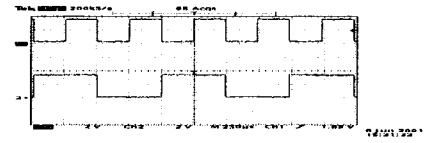


그림 9. Multiplexer 디코딩 신호

그림 9는 4bit binary counter O0, O1에서 나오는 multiplexer의 디코딩 신호로써 위의 count 신호에 의해서 변화가 된다. 그림에서 보듯 0000-1111까지 변화에서 하위 bit 두자리 변화를 볼 수 있다.

3. 결 론

본 논문에서는 여러 비동기식 장치가 있는 멀티 디바이스가 여러개 있을 때 이것을 모아서 하나의 PC로 보낼 수 있는 간이 다중 시리얼 분배기를 설계해 보았다. 또 데이터의 종류에 따라 취득 주기를 달리함으로써 시스템의 효율을 향상시킬 수 있음을 보였는데 이것은 사용자에 따라 유연한 시스템 적용이 가능하다는 장점이 있는 것이다. 하지만 많은 수의 멀티 디바이스가 있을 경우 데이터 취득 속도의 문제 그리고 데이터 전송 거리가 길어질 경우를 생각한다면 무선 모뎀을 이용하거나 아니면 다른 직렬통신을 이용하는 방안을 모색해야 할 것이다.

(참 고 문 헌)

- [1] 황희용, "CROSSTALK XVI 및 데이터 통신 기초와 RS-232C 해설", 교학사, pp.223-458, 1988
- [2] 류형선 김현 이시연 양경록 김양모, "μ컨트롤러를 이용한 범용 아날로그 데이터 취득 시스템 설계", 대한전기학회, 추계 학술대회 논문집, pp.732-734, 2000
- [3] Intel, "80196 User Manual",
- [4] 윤덕용, "80C196KC마스터(1)", Ohm사, pp.293-322 753-758, 2000
- [5] 차영배, "PIC16C84/71", 다다미디어, 1998
- [6] 신철호, "PIC 마이컴 길라잡이", 청안당, 1998