

EPx 자동생화학분석기와 컴퓨터간의 양방향 통신 시스템 구축

계명대학교 공과대학 전자계산학과*, 의과대학 임상병리학교실

박기현* · 전효진

서 론

병원에서의 임상병리과 역할은 아주 중요하므로 병원 전산화를 위해서는 임상병리 정보시스템 (laboratory information system, LIS)의 구축은 필수적이다. 따라서 LIS에 대한 연구가 최근 활발해지고 있다¹⁻⁴⁾. 그런데 임상병리과에서 사용하는 대부분의 검사기기들은 자체적으로는 자동화되어 있으나, 출력 장치들로는 고작 자체의 프린터나 화면 등이 있을 뿐이다. 따라서 검사결과를 보고서에 옮겨적거나 컴퓨터에 입력하는 수작업이 전체 작업의 상당 부분을 차지하고 있으므로 검사결과 배달이 지연된다. 이런 원인으로 인하여 환자의 진료 대기시간은 늘어나고 있다.

그러므로 검사기기들로 부터 검사결과가 직접 컴퓨터에 입력될 수 있게 하는 통신 시스템이 필요하다. 또한 컴퓨터가 검사기기들의 검사항위를 제어할 수 있게 한다면 더욱 빠르고 정확한 검사항위가 이루어질 것이다. 그런데 검사기기와 컴퓨터간의 양방향 통신의 중요성에 비해서 이에 관한 연구는 많지 않다.

본 연구의 목적은 임상병리과의 ABBOTT SPE CTRUM EPx 자동생화학분석기 관련 작업을 전산화하기 위한 양방향 통신을 구축하는데 있다. 즉, EPx에 컴퓨터(PC)를 연결하고 이들간의 양방향 통신을 가능케 하므로써 기존의 수작업에서 야기될 수 있는 오류를 줄이고 시간과 인력을 절감하고자 한다.

재료 및 방법

본 연구에서는 그림 1과 같이 EPx와 컴퓨터(PC)를 RS232-C port를 통하여 연결시키고 양방향 통신 소프트웨어를 제작했다. 그림 안쪽 정면에 EPx가 놓여있고 오른쪽 책상에 PC가 있다. 양방향 통신

소프트웨어를 위한 모든 함수는 C 언어로 제작되었고 serial 통신을 위해 __bios_serialcom 함수를 사용하였다. 이 함수는 Borland C에서도 지원되고 MS C에서도 지원하는데 본 연구에서는 Borland C 3.0을 사용해서 제작하였다. 화면 입출력을 위해 한라프로 2.4를 사용했고 한라프로용 입출력 보조 프로그램 (PP2_WIN2.C) 및 PP2_WIN2.H 파일도 사용했다.

결 과

1. 시스템 개요

EPx의 직렬 포트를 나온 전송 차료는 RS-232C를 통해서 시스템의 통신 엔진에 도달하게 된다. 통신 엔진은 직렬 송수신을 담당하고 메인 프로그램은 통신 인터페이스를 통하여 통신 엔진을 제어한다. 이외에도, 마우스나 키보드의 동작을 받아들이는 입력장치 관리 루틴, GUI 화면을 제공하는 화면 관리 루틴, 그리고 원하는 형태대로 분석결과를 프린터에 출력하게 하는 프린터 지원 루틴이 있다. 출력 포맷은 프로그램 수정 없이도 가능한데, 포맷을 지정해주는 파일은 PRINT.FRM이다. 이 파일은 텍스트형 파일이므로, 출력 포맷을 바꾸려면 기존의 에디터로 이 파일을 수정하면 된다.

이 시스템의 실행 파일은 EPX.EXE이다. 이 파일을 실행시키면 로고가 나타나고 여기서 아무런 키나 누르면 작업 메뉴들을 볼 수 있다. 작업 메뉴는 크게 'sample', '결과 찾기', '지우기', 그리고 '끝내기'로 나누어진다. 작업은 키보드의 화살표 키와 ENTER 키를 이용하여 선택할 수도 있고, 마우스의 오른쪽 버튼을 클릭하여 선택할 수도 있다. 선택 취소 혹은 화면 빠져나가기 기능은 키보드의 ESC 버튼 혹은 마우스의 원쪽 버튼을 누름으로써 행해진다.

'Sample'은 분석 schedule에 사용되고 '결과 찾

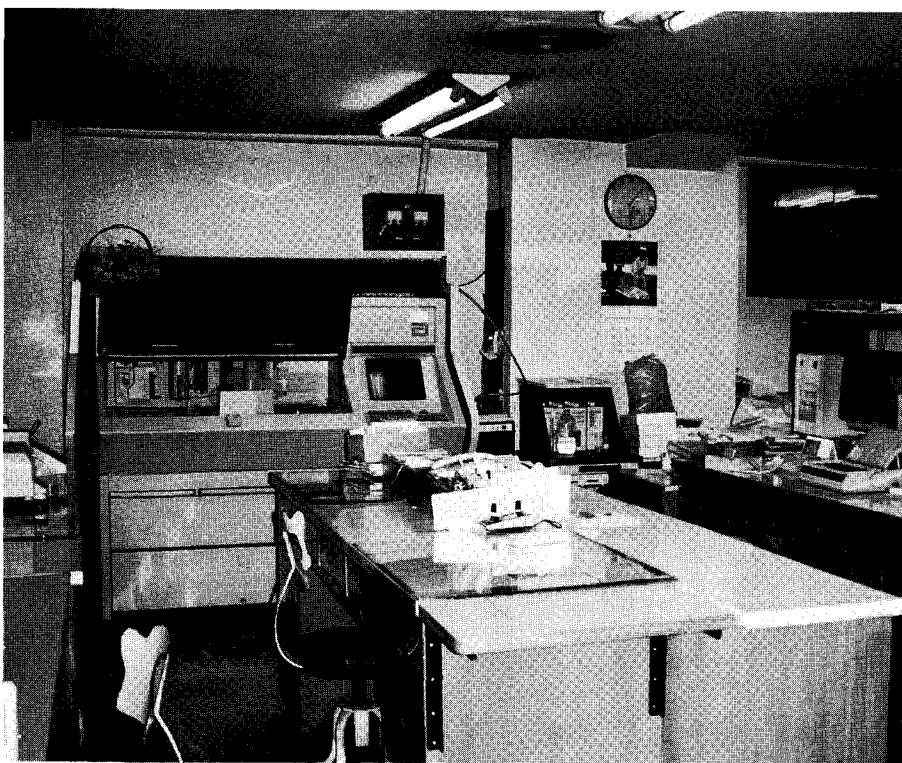


Fig 1. EPx와 PC와의 연결

기'는 분석이 끝난 결과를 처리할 때 사용된다. '지우기'는 분석결과를 데이터베이스에서 지울 때 사용된다. '끝내기'는 이 시스템을 빠져나갈 때에 사용된다. '결과 찾기'에서는 결과를 carousel 번호나 SID 번호로 찾는다. 'carousel 찾기'를 선택하면 carousel 번호를 묻게 되고, 원하는 carousel 번호를 입력하면 해당 carousel의 분석 결과를 데이터베이스에서 가져와서 보여준다. '지우기'는 분석 결과를 데이터베이스에서 지운다. carousel 단위나 SID 단위로 선택하여 결과를 지울 수 있다.

2. 통신 시스템

2. 1 통신 Cable

본 연구에서는 EPx와 PC를 연결하여 양방향 통신을 가능하게 하였다. EPx는 hardware handshake를 지원하지 않는다^{5,6)}. 그러므로 RS-232C interface의 pin들 중에서 입출력을 위한 data line들과 ground line만을 필요로 한다. PC의 interface가 DTE로 사용되므로 그림 2와 같이 cable을 연결하여 사용하였다. 25 pin female connector를 사용하였다.

양방향 통신을 위한 소프트웨어의 실행흐름도는

| ABBOTT EPx | HOST Computer (PC) |
|------------------------|--------------------|
| DCE | DTE |
| 1 GND (CHASSIS GROUND) | 1 GND |
| 2 TXD (TRANSMIT DATA) | 2 TXD |
| 3 RXD (RECEIVE DATA) | 3 RXD |
| 7 GND (SIGNAL GROUND) | 7 GND |

Fig 2. 양방향 통신 cable 연결

그림 3과 같다. 이 소프트웨어를 위하여 제작된 함수들중에서 중요한 것들은 appendix에 설명되어 있다.

고 찰

ABBOTT SPECTRUM EPx 자동생화학분석기는 환자의 혈액을 검사하여 albumin, calcium, uric acid 등의 성분 함량을 분석하는 기기이다. 이 기기의 분석작업 자체는 자동화되어 있으나, 출력장치로는 자체 스크린과 프린터가 있을 뿐이다. 따라서 스크린이나 프린터에 출력되는 결과를 다시 보고서에

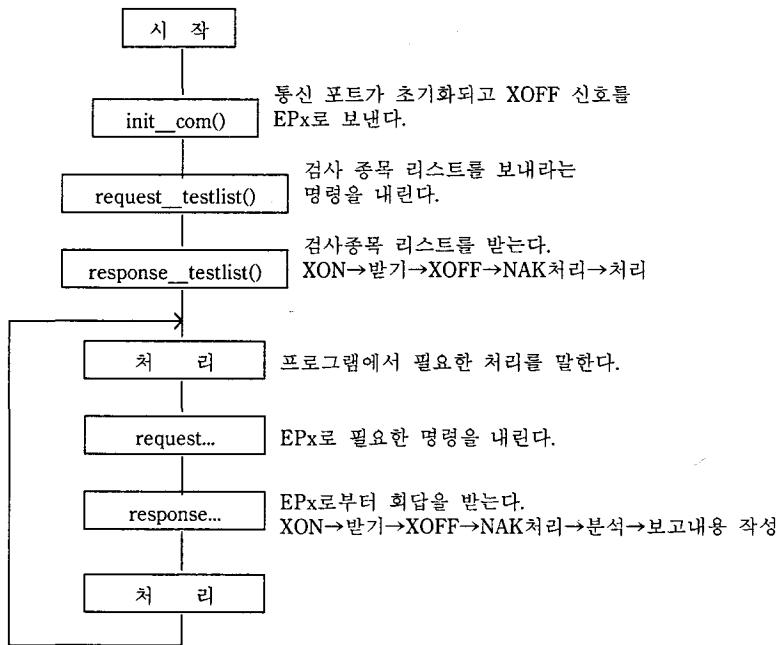


Fig 3. 양방향 통신 소프트웨어 실행흐름도

옮겨 적는 수작업이 오차와 병목현상의 원인이 되고 있다. 본 연구는 이러한 단점을 없애기 위하여 EPx와 PC 사이의 양방향 통신을 가능하게 하는 통신 시스템을 제작하였다.

그런데 EPx가 hardware handshaking을 지원하지 않으므로 XON과 XOFF의 software handshaking을 통해 입출력의 흐름을 제어하였다. Host (PC)가 데이터를 받을 수 없는 상황일 경우, EPx에 XOFF를 보내면 EPx는 데이터를 더이상 보내지 않고 기다리게 된다. 다시 데이터를 계속 받기 원하면 XON 신호를 보내준다. 본 연구에서는 interface 초기화로서, baud rate는 1200, parity는 none, start bits는 1, 그리고 stop bits는 1로 정했다.

EPx는 데이터의 입출력을 동시에 할 수 없다. 그러므로 EPx가 데이터를 보내고 있는 동안에는 host가 데이터를 보내서는 안된다. EPx가 데이터 입출력 작업을 끝낸 다음에야 host가 명령을 보낼 수 있다. EPx가 명령을 제대로 받지 못할 경우 NAK report를 보낸다.

EPx는 schedule과 report를 위해 number 모드와 name모드의 두가지 모드로 작동하도록 되어있다⁵⁾. number 모드는 검사 종목을 번호로 표시하는 모

드이고 name 모드는 검사 종목을 이름으로 표시하는 모드이다. name 모드를 사용하면 사용자들이 결과를 바로 알아보기 편리한 장점이 있다. 본 연구에서는 두 모드를 모두 지원하도록 소프트웨어를 제작하였으나, 여기서는 name 모드를 기준으로 설명하였다. 이 두 모드를 사용하기 전에 먼저 EPx로 send test list request 명령을 보내야 한다. 이 명령은 검사 종목들을 보내라는 명령도 되지만 두 모드 중 하나로 선택하는 역할도 한다.

시스템이 작동되면, 우선 PC는 EPx로 부터 test list를 받아들인다. Testlist에는 가능한 검사종목이 들어있다. 이 작업이 성공적으로 끝나면, PC는 필요에 따라, auto send, schedule, delete 혹은 recall 요구들을 EPx에 보낼 수 있다. PC는 이러한 요구들을 이용하여 EPx의 작동을 제어할 수 있다. Appendix에는 본 연구를 위하여 제작된 함수들을 설명하고 있는데, 위에서 언급한 요구들에 대한 설명도 같이 포함되어 있다. 한편, EPx는 초기의 testlist 전송이후에 PC로 부터 오는 요구들을 처리하여 그 결과를 PC로 전송한다. EPx로 부터의 전송된 검사 결과는 자동으로 PC에 저장되고 보고서 양식에 맞게 출력할 수 있도록 하였다.

요 약

본 연구에서는 EPx와 PC를 RS-232C port를 통하여 연결하고 양방향 통신 소프트웨어를 설계하고 제작하였다. 이를 통하여 EPx의 검사행위를 PC가 자동으로 제어할 수 있게 되었을 뿐만 아니라, EPx의 검사결과가 PC에 자동으로 저장되고 slip에 출력될 수 있게 되었다. 본 논문의 EPx를 위한 전산화는 임상병리과 업무 및 병원업무 전체의 전산화를 위하여 사용되어 질 수 있을 것이다.

앞으로의 연구과제로서, 정도검사 등의 기능을 추가한 기능적 보강이외에 사용자들의 보다 편리한 사용환경 제공 등이 있다. 또한, 타 검사기기들과 PC간의 양방향 통신을 위한 소프트웨어 제작도 계획하고 있다.

참 고 문 헌

- 민원기, 지현숙, 배직현 : 서울중앙병원 임상병리과 정보관리시스템(I)-바코드를 이용한 검체접수 전산시스템. 임상병리와 정도관리 1989; 11(2): 235-240.
- 민원기 : 서울중앙병원 임상병리과 정보관리시스템(II)-Down-sized computer를 이용한 일반화학 검사실의 전산화. 대한임상병리학회지 1993; 13 (1): 27-34.
- Blume, P: Design of a Clinical Laboratory Computer System, in Aller RD, Elevitch FR(eds): *Clinics in Laboratory Medicine: Laboratory and Hospital Information Systems*, ed 1. W. B. Saunders Co., 1991, pp. 83-104.
- Weilert M, Ashwood ER, Skinner M: The Ins and Outs of Laboratory Information Systems: Combining the Hospital and Reference Laboratory, in Aller RD, Elevitch FR: *Clinics in Laboratory Medicine: Laboratory and Hospital Information Systems*, ed 1. W. B. Saunders Co., 1991, pp. 153-170.
- Abbott Laboratories: Abbott Spectrum® EPx™ Uni-directional and Bi-directional Host Interfaces Specification for Software Versions 5.2 and Higher, ECOS2-1195, Abbott Diagnostics Division, 1989.
- Abbott Laboratories: Operator's Manual, Abbott Diagnostics Division, 1989.

Appendix (제작된 중요 함수들)

본 연구를 위하여 제작된 중요 함수를 살펴보면 다음과 같다:

- void init_com(void): Serial port를 초기화 하는 함수이다. EPx에 맞게 설정을 바꾼다. 가장 먼저 이 함수를 한번 호출해야 한다.
- void x_on(void): XON 신호를 보낸다.
- void x_on(void): XOFF 신호를 보낸다.
- int com_receive(const char *file_name, int time_out_second): EPx로 부터 데이터를 받는다. file_name은 받을 파일의 이름이고 time_out_second는 기다리는 시간이다. 이 함수를 직접 사용해 데이터를 받기 보다는 다음에 설명 할 reponse... 함수들을 사용해 대답을 받아야 한다. 호출되면 화면에 창이 생기고 현재 상황을 알려준다. <ESC>를 누르면 취소된다. 성공하면 0보다 큰 값이 return된다. 실패하거나 <ESC>를 누르면 0이 return된다.
- int com_send(const char *str): EPx로 데이터를 보낸다. str은 보낼 문자열을 가리킨다. 문자열은 '\0'으로 끝나야 한다. 이 함수들을 직접 사용하기 보다는 request... 함수들을 사용해 명령을 보내야 한다. 호출되면 화면에 창이 생기고 현재 상황을 알려준다. <ESC>를 누르면 취소된다. 성공하면 0보다 큰 값이 return된다. 실패하거나 <ESC>를 누르면 0이 return된다.
- int request_testlist(void): Name 모드로 검사 종목의 리스트를 보내는 명령을 내린다. com_send()를 사용한다. Return값은 com_send()와 같다. 명령을 주거나 대답을 받기 전에 한번은 이 함수를 호출해야 한다.
- int request_schedule(schedule_t *sch, int test_cnt, char *testlist, char *buffer): 검사 schedule을 잡는 명령을 만들어 보낸다. Sample의 상태를 지정한다. 이 함수가 사용된 다음에는 반드시 대답을 받아 보아야 한다. schedule_t 형은 SID, stat flag, barcode flag, carousel no, position no 등으로 정의되어 있다. test_cnt는 검사 종목의 갯수를 나타내고 testlist는 검사 종목(8자)의 연속된 나열을 가리키고 있다. buffer에는 이 명령을 EPx의 명령 품의 문자열 형태로 만들어 넘겨준다. 실제로 EPx로 보내진 문자열이다. 받을만큼 충분히 확보되어야 한다. malloc으로 한 1000바이트 정도를 미리 확보해 두고 다음에 설명하는 명령들과 같이 사용하면 좋다. 성공하면 0보다 큰 값을 아니면 0을 return한다.

- int request_delete_carousel(const char *car_n, char *buffer): 지정된 carousel의 내용을 모두 지우는 명령을 보낸다. 이 함수를 호출한 후 꼭 결과를 받는 함수를 호출해서 대답을 체크해야 한다. car_n은 carousel 번호를 문자열 형태로 담고 있는 배열이다. buffer는 함수 requestschedule()에서의 buffer와 같은 역할을 한다. 이 함수를 호출하기 전에 carousel의 상태를 따로 받아두는 것이 좋다. 성공하면 0보다 큰 값을, 아니면 0을 return한다.
- int requestdeletesid(const char *sid, char *buffer): 지정된 sid의 내용을 지우는 명령을 보낸다. 단위가 sid라는 점을 제외하면 함수 requestdeletecarousel()과 같다.
- int requestdeletrepid(const char *pid, char *buffer): 지정된 pid의 내용을 지우는 명령을 보낸다. 단위가 pid라는 점을 제외하면 함수 requestdeletecarousel()과 같다.
- int requestresultcarousel(const char *carn, char *buffer): 지정된 carousel의 검사 결과를 보내라는 명령을 내린다. 이 함수가 호출된 후 꼭 결과를 받는 함수가 호출되어야 한다. carn은 carousel 번호를 문자열 형태로 기억하고 있는 배열이다. 성공하면 0보다 큰 값을 아니면, 0을 return한다.
- int requestresultsid(const char *sid, char *buffer): 지정된 sid의 검사 결과를 보내라는 명령을 내린다. 단위가 sid라는 점을 제외하면 함수 requestresultcarousel()과 같다.
- int requestresultpid(const char *pid, char *buffer): 지정된 pid의 검사 결과를 보내라는 명령을 내린다. 단위가 pid라는 점만 제외하면 함수 requestresultcarousel()과 같다.
- int requestautosend(const char *function_code, char *buffer): 자동으로 결과를 보내게 한다. functioncode는 작동을 위한 기능코드이다. 성공하면 0보다 큰 값을 아니면 0을 return한다.
- int findcommand(FILE *file, const char *command): 받은 대답들 중에서 특정한 command(명령이라기보다는 대답의 종류코드)를 찾아낸다. Open된 파일에서 command를 찾아 서화일 포인터를 대답 코드가 위치한 곳으로 옮긴다. 다음부터 읽으면 원하는 대답을 완전히 찾을 수 있다. 직접 사용할 일은 드물 것이다. 아래에 제시된 대답 받기 함수에서 사용 되어

진다. 성공하면 0보다 큰 값을 리턴하고, 찾지 못하면 0을 return한다.

- int checknak(const char *filename): 대답으로 받은 파일에서 NAK가 있는지 찾는다. NAK가 발견되면 0을, NAK가 없으면 0보다 큰 값을 리턴한다.
- int response_testlist(char testlist[][9]): 검사할 수 있는 종목들의 리스트를 받는다. 함수 request_testlist()의 대답을 받는다. testlist[][9]] 배열에 종목이름(8자 + '\0')들을 넣어 주고, 총 갯수를 return한다.
- int response_schedule(schedule_t *sch): 함수 request_testlist()의 대답을 받는다. 결과는 temp.dat에 기록된다. 그리고 sch에 정보를 수정하여 보내준다. 결과보다는 특정한 sid의 상태(pid, posititon number, flag들)을 알아내기 위해 사용되어 진다. 성공하면 0보다 큰 값을 실패하거나 NAK가 발견되면 0을 return한다.
- int response_delete(schedule_t *sch): request_delete...%4/4의 대답을 받는 함수이다. 지워진 sample의 정보를 sch에 담아 보내고 completion code를 return한다. 'D'가 return되면 지워졌음을 나타내고, 'E'가 return되면 예러가 났음을 나타내고, 'N'이 return되면 원한 sample을 찾지 못했음을 나타낸다. 0이 return되면 받는데 실패했거나 NAK가 발견되었음을 나타낸다.
- int response_result(const char *file_name): 검사 결과를 받는다. 함수 request_result...()의 대답을 받는 함수이다. file_name은 결과를 받을 파일이름이다. 이 파일은 알아보기 좋은 형태의 텍스트 파일이다. 이 파일을 바로 보거나 바로 프린터로 출력할 수 있다. 또한 다른 프로그램에서 사용할 수 있도록 format된 형태의 ("result.tmp") 파일도 만들어준다. 성공하면 0보다 큰 값을 리턴하고 아니면 0을 리턴한다.
- int response_result_sid(schedule_t *sch, int MAX, char select[], char testlist[])[받은 결과에서 sch.sid에 일치하는 내용을 찾아서 select[]를 설정해 준다. 원하는 sid의 상태를 알아내고 현재 선택된 검사 종목을 확인하기 위해 사용된다. 성공하면 0보다 큰 값을 return하고, 아니면 0을 return한다.]

=Abstract=

Constructing a Bi-directional Communication System between an EPx Auto-chemical Analyzer and a Computer

Kee Hyun Park, Ph.D.* and Hyo Jin Chun, M.D.

Department of Computer Science, Keimyung University College of Engineering,*

*Department of Clinical Pathology, Keimyung University School of Medicine,
Taegu, Korea.*

It has been understood that a laboratory information system(LIS) plays an important role in a hospital information system(HIS). However, few HIS has been equipped with a fully automated LIS. This is partly because, in order to achieve a fully automated LIS, there are many testing equipments which should communicate with computers in both directions. Furthermore each equipment has its own communication protocol.

In this paper, a bi-directional communication system between an ABBOTT SPECTRUM EPx auto-chemical analyzer and a PC is constructed. With the communication system, it is possible that a PC controls an EPx's operations and that the PC receives test results from the EPx automatically. RS-232C serial ports are used in order to make bi-directional communication possible between them.

Keywords : Laboratory Information System, bi-directional communication, RS-232C.