

착자 자동화 시스템 및 통계분석 툴 개발

이 천 희[†] · 하 기 종^{††}

요 약

현재 CRT(Cathode-Ray Tube)의 마그네트(Magnet) 생산업체에서 색상제어에 사용하고 있는 착자기 및 착자 검사기는 수동이므로 작업 시간이 오래 걸리고 오동작시킬 우려가 있다. 본 연구에서는 착자와 검사 공정까지를 한 시스템으로 묶은 시스템 장치를 개발하여 전공정을 자동화시켰다. 따라서 본 시스템은 매회 작업 상태를 검사하고 품질의 산포를 정확하게 분석하여 착자 및 착자 상태 검사의 신뢰도를 향상시켜 불량율을 없애도록 하였다.

Development of Statistical Analyzing Tool and System of Automatic Magnetizer

Cheon-Hee Yi[†] · Ki-Jong Hah^{††}

ABSTRACT

The magnetizer that is magnetizing the magnet which is used for RGB(Red Green Blue) control of CRT(Cathode-Ray Tube) and the magnet inspection unit which is used for test the state of magnetizing have been imported by magnet manufactures up to now. They are operating by manual now, so that they are needed lots of time and have an increase in probability of malfunction by operator.

In this study, we have developed a united system from the magnet production process to inspection work to automated a entire progress. Therefore, as we are testing the status of every work and analyzing exactly the distribution in quality with this system, we have known that the reliability of magnetization and magnetized status test has not only increased, but the rate of inferior quality almost not generated.

1. 서 론

현시대는 빠르게 변화하는 기술을 따라가기 위하여 점점 자동화 되어지는 추세이다. TV와 컴퓨터는 필요 불가결한 제품이 아닐 수 없고, TV 브라운관과 컴퓨터 모니터에 사용되는 마그네트(magnet)의 생산

량이 크게 증가하고 있다. 따라서 증가하는 생산량을 유지하려면 착자 공정을 자동화 하지 않을 수 없게 되었다.

현재까지는 착자공정 검사장치가 부분적으로 자동화가 되어있으나 장착부터 착자하여 검사까지의 전공정을 자동으로 하고, 결과 분석까지 시도하는 연구는 외국에서도 개발되어 있지 않으므로 PLC를 사용한 공장자동화 시스템과 착자와 검사 공정까지를 한 시스템으로 묶은 시스템장치를 개발하여 생산성을 높이고 불량율을 크게 감소시켰다.

※이 논문은 청주대학교 연구비 지원에 의해 연구되었음.

† 종신회원: 청주대학교 전자공학과 교수

†† 정 회 원: 영동전문대학 전자통신과 전임강사

논문접수: 1996년 1월 6일, 심사완료: 1996년 4월 22일

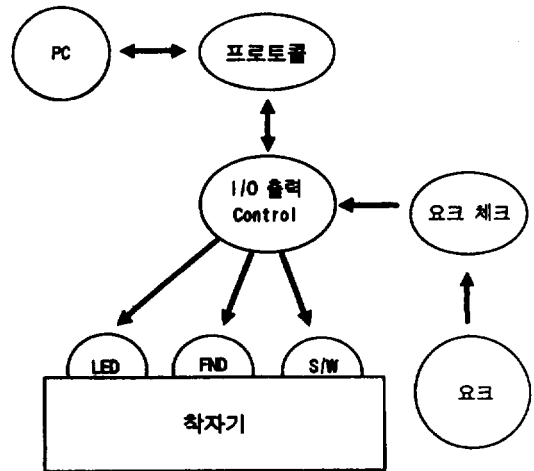
본 자동 착자 및 검사 시스템은 마그네트의 품질을 좌우하는 착자 및 검사의 신뢰성을 향상시키기 위하여 착자와 성능검사의 자동화가 구현되어 불량이 혼입될 수 있는 소지를 최소화 하였고, 매회 작업상태를 검사함으로써 불량제품의 생산을 조기에 발견하여 불량제품의 생산을 막아주며, 또한 기존의 수동적인 검사 방법으로는 데이터를 기록하지 못하고 불량여부만 판정하므로 정확한 품질의 산포를 알 수 없었으나 본 시스템을 사용하면 제품의 가우스(gauss) 값을 판독 및 유지하여 전체 공정능력지수(CP) 및 품질의 산포를 정확하게 분석하여 생산 데이터 관리가 가능하게 하였다.

이 자동 착자 및 검사 시스템은 몇가지 부분으로 나뉘어 개발되었는데 먼저 제어 시스템 부분과 프로토콜 모듈 부분과 착자기를 운영하는 소프트웨어 부분으로 나뉘어 개발되었다. 제어 시스템 부분은 8051 계열의 원칩을 이용하여 하드웨어의 크기를 최소화 구성하였고 이것은 두개의 보드로 나뉘어 지는데 하나는 원칩과 메모리 소자를 갖고 있는 메인보드 부분이고 다른 하나는 착자기와 연결을 하기 위한 입출력 제어 보드로 구성된다[1, 2]. 제어 시스템을 컴퓨터에서 제어하기 위해 직렬 통신을 이용하기 위한 프로토콜을 개발하여 제어 시스템을 제어하였으며 직렬 통신을 이용함으로써 여러대의 제어 시스템을 제어가 가능하여 차후 다수의 착자기 제어도 할 수 있다. 개발한 프로토콜은 기본적인 엑스모뎀/씨알씨(XMODEM/CRC) 프로토콜 규약을 따라 개발된 것이며 제어 보드를 제어하기 위해 기본적인 통신 기능에 몇가지를 더 추가 시켰다. 그리고 본 시스템을 구동하기 위한 운영 소프트웨어에는 작업 설정 모드와 작업 시작 모드 그리고 통계분석 모드의 세가지 프로그램으로 구성된다. 검사 작업시 가우스 미터로 마그네트(magnet)의 가우스 값을 읽어 컴퓨터에서는 A/D 컨버터를 이용해서 읽어 들이며 가우스 값을 잘못 읽어 들일 수 있는 문제점을 보완하기 위해 가우스 값을 보정하도록 프로그램 되었다. 끝으로 통계분석은 작업 모드에서 생성된 작업지도서와 작업 시작모드에서 생성된 정보를 데이터베이스화 하여 분석할 수 있도록 하였고, 출력 모드에는 데이터 분석에서 생성된 데이터를 화면과 프린터로 출력하도록 처리하였다. 이렇게 해서 모든 착자 작업이 수행되며 작업자가 작업

설정만 해주면 자동적으로 모든 작업이 수행된다. 운영 소프트웨어를 개발하기 위해 Borland C++ v3.1을 사용였고 프로토콜 프로그램은 Borland C++ v3.1과 C51 컴파일러를 이용하여 C++와 어셈블리로 개발되었다.

2. 전체 시스템의 구성

현재 작업공정의 순서는 작업자가 요크에다 마그네트를 20개씩 올려 놓은 후에 착자기의 출력 전압을 맞춘후 착자지시 스위치를 눌러 착자시키고 그중 한개를 작업자가 가우스 메탈을 사용하여 자화값이 기준치와 맞는지 비교하여 합격 여부를 판정하였다. 그러나 본 시스템은 PLC 사용한 공장자동화 시스템과 착자와 검사 공정까지를 한 시스템으로 묶는 시스템 장치를 개발하여 전 공정이 자동으로 이뤄지게 하였다. 본 연구에서 개발한 착자기 제어 시스템을 구성하는 부분은 크게 두가지 부분으로 나뉜다. 첫째는 하드웨어 부분이고 둘째는 그 하드웨어와 컴퓨터 간의 통신을 하는 프로토콜과 명령을 수행하는 부분이다. 그림 1은 컴퓨터에서 착자기와의 관계를 보여준다.



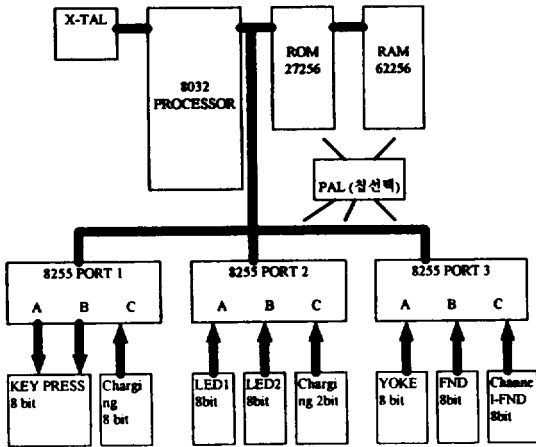
(그림 1) 제어 시스템 신호 흐름도
(Fig. 1) Signal flow-chart of control system

하드웨어는 PC와의 통신을 하고 또 제어보드로 부터 데이터를 입력 받아 처리하기도 하고 출력하기도

하는 메인 보드와 착자기 제어를 위한 제어 보드, 그리고 각종 입출력 신호를 처리하기 위한 보드로 구성된다.

2.1 메인 보드

제어를 담당하는 메인 보드는 컴퓨터와 시리얼 통신을 통한 제어코드를 입력받아 작업 지도서에 설정되어있는 값으로 착자기의 전압과 전류값등을 설정하고 착자에 필요한 착자기의 작동명령을 제어보드로 보내고, 반면에 제어보드를 통해 착자기에서 발생하는 모든 착자기의 변화(LED, FND, KEY 등)를 읽어 PC로 전달해 주는 보드로서, 프로세서(8032)와 ROM(27256), RAM(62256), PAL(16L8) 등을 사용하여 제작 하였다.[3, 4, 5]



(그림 2) 제어 시스템 구성도
(Fig. 2) Configuration of control system

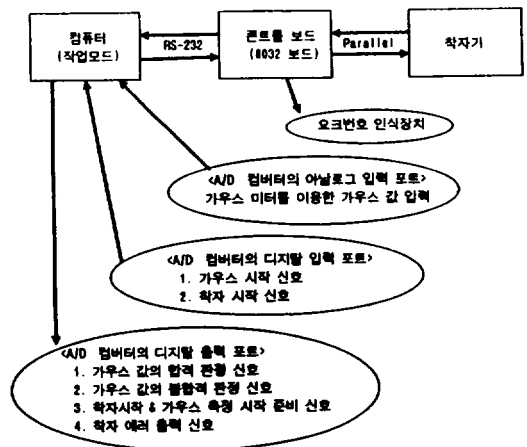
2.2 제어 보드

제어 보드는 착자기 제어를 위하여 메인 보드의 신호를 착자기로 출력하고, 착자기의 변화 신호를 입력받아 메인 보드로 전송하는 일종의 데이터 입출력 보드로서 3개의 병렬 인터페이스용 PPI(8255A)를 사용했다. 제어보드에서는 착자기 판넬의 제어를 할 수 있도록 내부 보드에 연결하여 제어한다. 이 제어보드가 메인 보드와 분리된 이유는 하드웨어 상에서 문제가 생겼을 경우에 대비해서 쉽게 보드를 교체 가능하게 하기 위해 분리해서 제작 했다. 이 제어 보드에

는 많은 저항과 콘덴서가 들어가는데 이는 착자기에서 발생하는 노이즈를 없애기 위해 하나 하나의 전송 선로에 연결된다.

2.3 각종 입출력 신호 처리 보드

입출력 신호에는 가우스값과 그밖에 여러가지 제어 신호들이 있는데, 먼저 가우스 값을 입력받기 위해 가우스 입력 장치로 부터 가우스 미터를 통해 측정된 가우스 값은 A/D 컨버터를 사용하여 데이터를 읽어 들이며, 이 데이터는 파일로 저장된다.



(그림 3) 각종 입출력 신호
(Fig. 3) Input/Output signal for various kinds

가우스의 정밀한 아날로그 신호값을 입력받기 위해 고분해능력을 갖는 20비트 이상의 A/D 컨버터를 이용하였고 가우스의 변동을 줄이기 위한 방법으로 측정시 수십개의 데이터를 읽어 적당한 값으로 보정하여 입력 받는다.

또한, 착자 시작 제어 신호와 가우스 측정 시작 제어 신호를 입력 받기 위해 같은 A/D 컨버터를 이용하여 디지털 데이터를 입력받아 컴퓨터에서 처리한다. 그리고 출력 제어 신호에는 가우스 값에 대한 합격/불합격 판정 제어 신호, 착자 시작 준비 제어 신호와 가우스 측정 준비 제어 신호, 착자 에러 제어 신호가 있는데 이 제어 신호들을 출력하기 위해 같은 A/D 컨버터의 디지털 출력 포트를 이용하였다.[6, 7, 8]

여기서 사용한 A/D 컨버터의 기종은 Model PCL-

812 Enhanced Multi-Lab Card이다. 기본 A/D 컨버터의 기본 어드레스는 0×220(Base Address)이다. 프로그램은 A/D 컨버터에서 제공하는 라이브러리를 이용하여 사용하였다.

3. 프로토콜

제어 보드와 컴퓨터간의 통신을 위해 전용 프로토콜을 개발하였으며, 정확한 통신을 하기 위해서는 데이터의 송/수신에 대한 규약을 정의한 프로토콜이 필요하다. 프로토콜에는 X-MODEM/CRC, XMODEM-1K, XMODEM-1K/G, X-MODEM 등이 있는데 이 중에서 XMODEM/CRC를 사용하였다. 프로토콜은 NULL 모뎀 방식으로 처리 하였고, 따라서 RS-232C의 3선만을 사용하며, 데이터 에러 검출은 CRC-16기법으로 처리하였다.

3.1 프로토콜을 위한 기본 구성들

프로토콜을 위한 기본 구성에는 먼저 통신 상태에 대해서 알아보고 통신을 위한 부호들에 대해 알아보고 패킷구조와 본 시스템에서 사용되는 명령들은 다음과 같다.

직렬 통신 상태

전송율은 초당 9,600 비트이며 데이터 비트는 8비트이며 스탑비트(Stop Bit)는 1이고 패리티 비트(Parity bit)는 없다.

사용되는 부호들

시작을 알리는 SOH→01, 문자의 시작을 알리는 STX→02, 문자의 끝을 알리는 ETX→03, 전송을 다 했다고 알리는 EOT→04, 송신이 잘못됐다고 즉 깨진 패킷이라고 알리는 NAK→05, 송신이 잘됐다고 알리는 ACK→06 부호들이 있다.

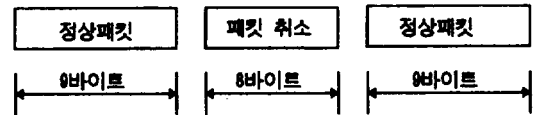
패킷 구조

본 프로토콜은 패킷 전송의 방식을 따르고 있으며 하나의 패킷은 전체 9바이트의 패킷으로 구성된다.

SOH (1 byte)	패킷번호 (1 byte)	STX (1 byte)	DATA (3byte)	ETX (1byte)	CRC16 (2byte)
-----------------	------------------	-----------------	-----------------	----------------	------------------

- SOH : 한 패킷의 시작.
- 패킷번호 : 패킷의 중복을 막기 위한 ID.
- STX : 데이터의 시작.
- DATA : 실제 data.
- ETX : 데이터의 끝
- CRC16 : CRC-16 코드. 이 CRC 코드는 2 byte로 구성되며 앞코드는 low 코드이며 뒤코드는 high코드로 되어있다.

패킷 전송시 패킷의 처음을 나타내는 SOH 부분과 CRC-16코드가 같을 수 있는데 프로토콜에서 이 부분의 문제점을 해결하기 위해 패킷의 길이를 가변이 아닌 고정(9바이트)으로 처리하여 정확한 패킷의 전송 및 복구를 할 수 있게 했다. 즉 SOH가 시작되면 그로부터 9바이트는 하나의 패킷으로 처리하고 그 다음에 SOH가 존재할 때까지 데이터는 무시된다.



(그림 4) 정상 패킷 선택
(Fig. 4) Selection of normal packet

패킷의 프로그램 구조는 다음과 같다.

```

struct PacketSt {
    unsigned char StartOfHead;
    unsigned char ID;
    unsigned char StartOfText;
    unsigned char Data[3];
    unsigned char EndOfText;
    union crc {int i; char ch[2];} CRC;
};
    
```

프로토콜을 위한 기본 명령구조

명령문은 3바이트로 구성되며 다음과 같은 기본 명령으로 착자기가 제어 된다.

키 작동

착자기의 판넬에 있는 키를 제어하기 위해 다음과 같은 구조의 데이터를 패킷에 실어 전송을 한다. 명

령 처리 내용은 다음과 같은 값을 사용한다.

명령어 (KEY_PRESS)	명령 처리내용 (CH_KEY)	데이터 (사용없함)
--------------------	---------------------	---------------

상태값 요구

이 명령은 현재 상태의 값을 요구하는 것으로 명령 처리 내용은 다음과 같다.

명령어 (GET_STATUS)	명령 처리내용 (LED_EXT)	데이터 (0 또는 1)
---------------------	----------------------	-----------------

상태의 변화

이 명령은 변화된 상태의 데이터를 보내는 명령이다.

명령어 (VARIED_STATUS)	명령 처리내용 (LED_EXT)	데이터 (0 또는 1)
------------------------	----------------------	-----------------

변화된 값 전달 유무 설정

이 명령은 변화된 상태를 보내줄 것인지 아닌지에 대한 플래그 값을 설정한다. 데이터 처리 내용에는 상태의 변화를 전송하는 명령에서 사용하는 명령 처리 내용과 같다.

명령어 (SET_VARIED_STATUS)	명령 처리내용 (LED_EXT)	데이터 (0 또는 1)
----------------------------	----------------------	-----------------

보통 다음의 사항들에 대한 변화만을 주고 받을 필요가 있으므로 초기설정에 다음에 대해서만 플래그를 설정해주면 된다. 여기에서 ALL_STATUS를 이용하면 TROUBLE LED와 UP LED만 제외하고는 모든 입력을 SET/RESET 할 수 있다.

ALL_STATUS 0×FF

1. 각 FND들. 모두 17개이다.
2. 각종 LED들. 모두 8개이다.

LED_EXT, LED_OVER, LED_UP,
LED_CR, LED_SHOT, LED_TROUBLE,
LED_CHARG_LEVEL_HI, LED_CHARG_LEVEL_LO,

명령어 (ERROR_OCCURED)	명령 처리내용 (발생된 에러 번호)	데이터 (에러 값)
------------------------	------------------------	---------------

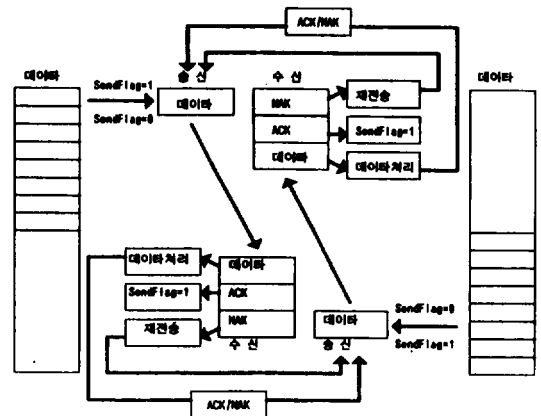
에러 발생

- 1:수행 에러, 즉 어떤명령을 받아 수행시에 기계가 동작을 안할때. 또는 LED에 불이 켜지지 않을경우, →메세지를 화면에 표시.
- 2:RESET에러, 즉 RESET키를 눌렀는데 FND등이 초기치로 되지 않을경우 →메세지를 화면에 표시한다. "RESET이 되지 않았음."

3.2 프로토콜 기법

1) 전체 프로토콜의 구조

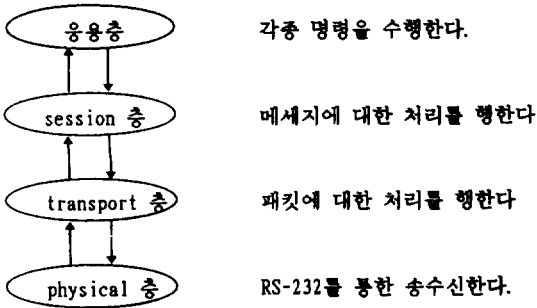
본 자동화 시스템에서는 제어 시스템을 제어하기 위해 전용 프로토콜을 개발하였는데 그림 5에서 보여주는 구조로 되어있다. 하나의 데이터를 가져와 전송을 하고 수신한 곳에서 데이터를 확인한후 그에 맞는 처리를 하고 그에 해당하는 신호(ACK 또는 NAK)와 ACK 신호를 받으면 Sendflag를 1로 세팅하고 NAK 이면 재전송을 요구하는 데이터를 송신했던 곳으로 데이터를 보낸다.



(그림 5) 프로토콜 처리 구조
(Fig. 5) Configuration of protocol process

2) 프로토콜 각 층별 구조

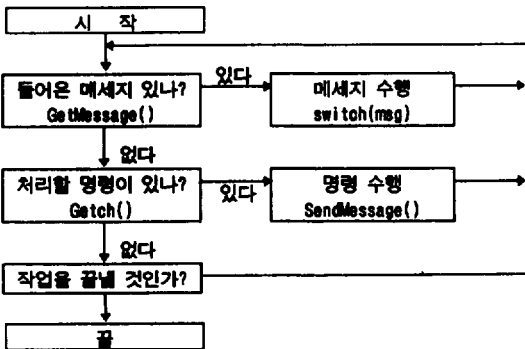
프로토콜은 각 네개의 층으로 구분하여 개발되었다.



3) 명령 처리 구조

프로토콜에서 가장 위층인 응용층에 해당하는 부분으로 메시지 처리에 대한 기본 프로그램 구조를 나타낸다.

새로 들어온 메시지가 있난 확인하고 있으면 메시지에서 요구하는 일을 수행하고 메시지가 없으면 처리해야 할 명령이 있나 확인을 하고 있으면 명령을 수행하고 없으면 작업을 끝낼 것인가를 확인한 후 끝낼거면 끝을 내고 그렇지 않으면 새로 들어온 메시지가 있는 확인을 하고 이런 식으로 끝내기 전까지 계속 반복한다.

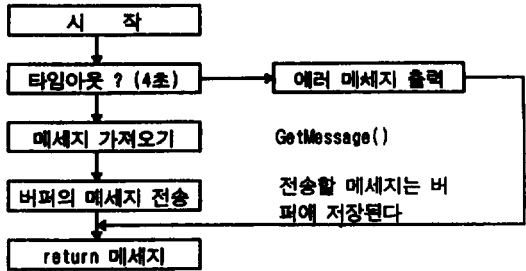


(그림 6) 명령 처리 구조
 (Fig. 6) Configuration of command process

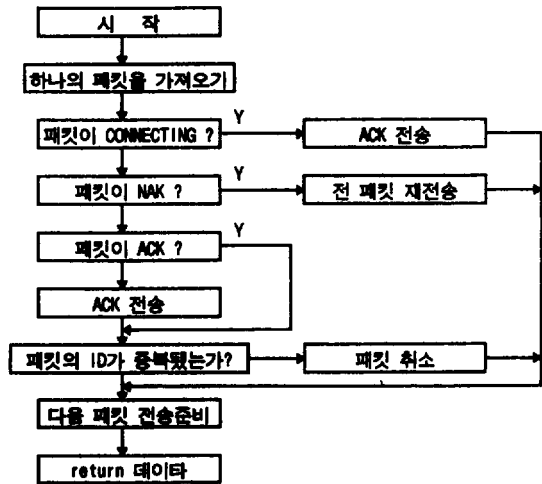
4) Session 층

Transport층으로 부터 데이터를 받아 들어온 메시지가 있으면 그 메시지를 위 응용층으로 되돌려 주며 전송할 메시지가 있으면 transport 층으로 전송한다.

5) Transport 층



(그림 7) Session 층 흐름도
 (Fig. 7) Flow-chart of session layer



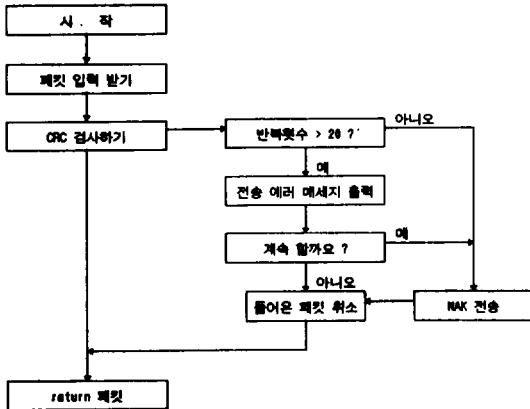
(그림 8) Transport 층 흐름도
 (Fig. 8) Flow-chart of transport layer

하나의 패킷을 가져와 이 패킷이 깨졌으면 재전송을 요구하고 온전한 패킷이면 이 패킷이 전에 왔던 패킷과 중복이 되는지 확인을 한후 다음 패킷 전송 준비를 하고 순수 데이터만을 session 층으로 되돌려 준다.

Transport 층에서는 physical 층으로 부터 가져온 하나의 패킷이 명령에 해당하는 데이터이면 위 session 층으로 보내고 그밖에 처리를 요구하는 데이터이면 각각에 대한 처리를 한다. 즉 들어온 패킷이 NAK이면 패킷을 재전송을 하고 명령 데이터이면 다음 층으로 되돌려 준다.

6) Physical 층

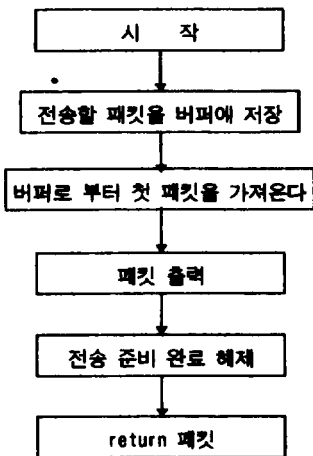
이 physical 층에서는 하드웨어 관련된 처리를 행하는 부분으로 RS-232를 통하여 데이터를 송수신한다. 여기서 데이터는 하나의 패킷에 해당하며 받은 패킷에 대해 이상이 없는가를 검사하는 것이다.



(그림 9) Physical 층 흐름도
(Fig. 9) Flow-chart of physical layer

7) 메시지 전송

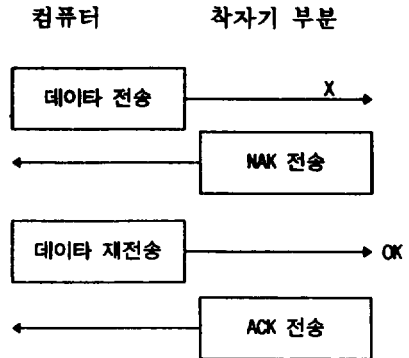
메시지를 보내는 것은 하나의 패킷을 전송하기 위해 다음 그림 3-8과 같은 흐름을 따른다. 전송하고자 하는 패킷은 항상 버퍼를 통해 저장되었다가 전송 준비가 되면 전송한다.



(그림 10) 메시지 전송
(Fig. 10) Message transportation

8) 통신상에서 데이터를 못받았을 경우

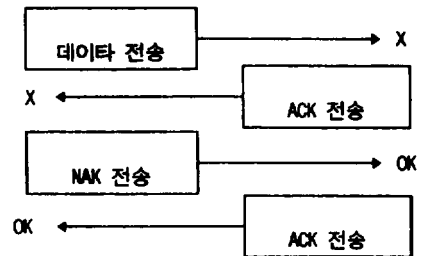
통신에서 데이터를 전송했는데 상대방에서 받지 못했을 경우에는 NAK 메시지를 전송하게 되며 상대방에서는 데이터를 재전송한다.



(그림 11) 데이터를 못 받은 경우의 프로토콜
(Fig. 11) Protocol of data missing

9) 통신상에서 ACK 메시지를 못받았을 경우

통신상에서 ACK 메시지를 못받았을 경우에는 ACK 메시지를 받을때 까지 계속해서 NAK 메시지가 전송된다.



(그림 12) ACK 메시지를 못받은 경우의 프로토콜
(Fig. 12) Protocol of ACK message missing

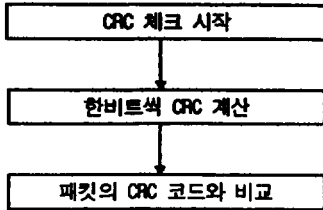
3.3 CRC 체크 기법

본 프로토콜에서 사용한 CRC-16 기법은 기존의 공개되어 있는 기법을 사용하였다.

1) CRC 코드 검사

CRC를 검사하기 위해 계산 공식에 따라 하나의 패킷을 계산하여 CRC코드를 만들어 그 패킷이 갖고 있

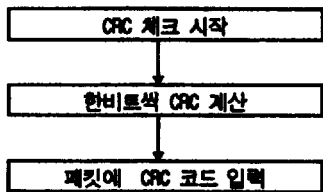
는 CRC 코드와 비교한다.



(그림 13) CRC코드 검사 흐름도
(Fig. 13) Flow-chart of CRC code check

2) CRC 코드 생성

CRC 코드를 만들기 위해서는 계산 공식에 의해 만들어지며 그 CRC 코드는 그 패킷의 CRC 데이터로 저장된다.



(그림 14) CRC 코드 생성 흐름도
(Fig. 14) Flow-chart of CRC code generation

3) 계산 공식

계산 공식은 속도를 빠르게 하기 위해 변환 테이블을 미리 만들어 계산한다. 하나의 데이터가 있는데 다음과 같이 accum이라는 CRC코드를 생성하는 계산 식으로 된다.

```

comb_val = (* accum >> 8) ^ data;
accum = (* accum << 8) ^ crctab[comb_val];
    
```

3.4 직렬 패킷 송수신

RS-232를 통한 기본적인 프로그램 기법을 살펴보자. 먼저 현재 인터럽트로 받아들인 데이터가 메모리에 들어 있는가를 검사한다. 그리고 하나의 패킷을 전송하고 또 하나의 패킷을 입력 받는 기법이다.

1) 들어온 패킷 유무 확인

메모리의 어드레스를 조사한다.

한 바이트씩 전송을 하는데 있어 이것을 인터럽트를 사용하며 데이터 메모리에 한 바이트씩 저장을 하면서 메모리 번지도 기록을 한다.

```

char rsready()
{
    if (*rb_e >= rb_s) return(*rb_e - rb_s > 8 ? 1:0);
    else return (*rb_e + RECV_BUFLEN - rb_s > 8 ? 1:0);
}
    
```

2) 패킷 전송

데이터를 전송 시키려면 다음과 같이 하나의 문자를 전송하면 된다. 다음 함수는 하나의 패킷을 전송하는 것이다.

```

void SendPacket(char * packet)
{
    char a;
    for(a=0; a < PACKLENG; a++) rsout(packet[a]);
}
    
```

3) 패킷 입력

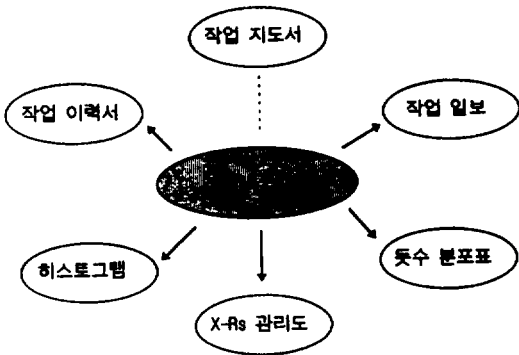
다음은 하나의 패킷을 입력받는 기법을 나타낸다. 즉 RS-232를 통해 들어온 데이터는 일정한 버퍼에 기록이 되며 그 버퍼로부터 하나의 패킷 단위로 가져오는 기법이다.

```

char GetPacket(char * packet)
{
    char i;
    if (!rsready()) return 0;
    for(i=0; i < PACKLENG; i++) {
        *(packet++) = read_XDATA(rb_s);
        if (++rb_s >= RECV_MEM_ADDR + RECV_BUFLEN)
            rb_s = RECV_MEM_ADDR;
    }
    return 1;
}
    
```

4. 통계 분석 및 데이터 처리 기법

통계 분석은 작업 설정 모드에서 생성된 작업 지도서와 작업 시작 모드에서 생성된 데이터를 가지고 작업 이력서, 작업 일보, 돛수 분포표, 히스토그램, X-Rs 관리도 등을 그림 15과 같이 분할하여 분석할 수 있도록 하였다.



(그림 15) 통계 분석도
(Fig. 15) Chart of statistical analyzing

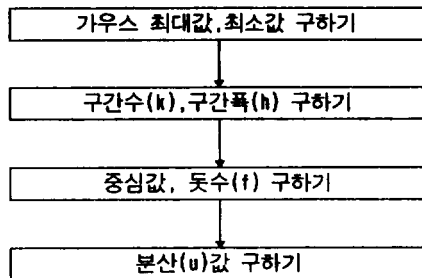
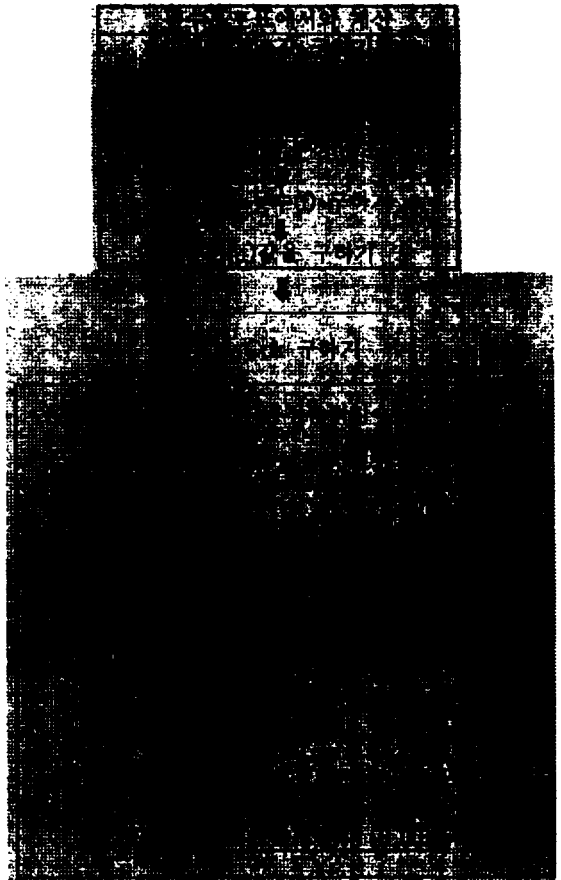
4.1 돛수 분포표

하루 생성된 데이터중 한모델에 대한 가우스값의 산포도를 보여준다. 산포도를 보면 생산한 하루의 제품중에서 어느 정도의 가우스 값을 갖는 것이 많은지 또는 적은지 쉽게 알 수 있다. 이 도표를 봄으로써 정확하게 숫자로 파악할 수 있으며, 히스토그램으로 대략적인 그림도 알 수 있다. 여기서 제품의 산포는 작을 수록 좋으며 계산수식을 살펴보면 다음과 같다.

f는 빈도수, u는 분산, n은 착자수량이며, Xmin은 가우스의 최소값이며, Xmax는 가우스의 최대값이며, R은 Xmax - Xmin이며, 구간수 k = 1 + 3.3 log n이며, h는 구간폭으로 R/k 또는 (Xmax - Xmin)/(1 + 3.3 log n)이며, 각각의 구간들은 Xmin - 최소측정단위(0.001)/2 + h로 되며, fu = f * u, u^2 = f * u * u 이라 하면 다음과 같은 순서로 계산을 한다.

4.2 히스토그램

하루 생성된 데이터중 한모델에 대한 가우스값의 산포도를 그래프로 보여줌과 동시에 공정능력지수(Cp)를 보여준다. 히스토그램을 출력하기 위한 계산 과정은 돛수 분포표와 같으며 여기에 공정능력 지수를 계산하는 과정이 더해진다. 공정능력 지수 계산



(그림 16) 돛수 분포표 계산과정
(Fig. 16) Calculation process for table of frequency distribution

과정은 다음과 같다.

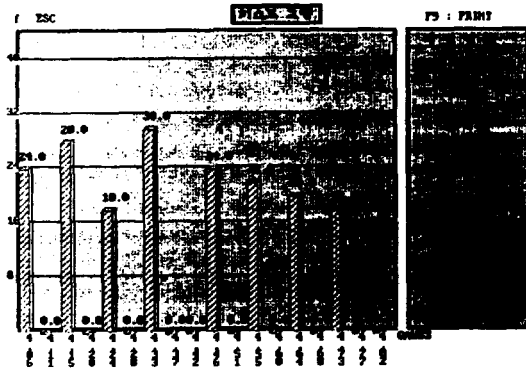
$$Cp = (Su - S1) / 6 * V^{(0.5)}$$

Su와 S1은 가우스값의 상한값과 하한값이며,

뫏수 분포표						
No.	GradeTerm	합산값	f	u	fu	fu ²
1	-0.001	-0.106	0.053	0	-13	-70
2	0.106	-0.212	0.159	0	-12	0
3	0.212	-0.319	0.266	0	-11	0
4	0.319	-0.425	0.372	0	-10	0
5	0.425	-0.532	0.478	0	-9	0
6	0.532	-0.638	0.585	0	-8	0
7	0.638	-0.745	0.691	0	-7	0
8	0.745	-0.851	0.798	0	-6	0
9	0.851	-0.957	0.904	0	-5	0
10	0.957	-1.064	1.011	0	-4	0
11	1.064	-1.170	1.117	0	-3	0
12	1.170	-1.277	1.223	0	-2	0
13	1.277	-1.383	1.330	3	-1	3
14	1.383	-1.490	1.436	50	0	0
계			50		-61	1819

(그림 17) 뫏수 분포표 출력

(Fig. 17) Output for table of frequency distribution



(그림 18) 히스토그램 출력

(Fig. 18) Output of histogram

$$V(0.5) = h * ((\text{SUM}(fu^2) - \text{SUM}(fu)^2 / \text{SUM}(f)) / (\text{SUM}(f) - 1))$$

이다.

이렇게 해서 구해진 공정능력지수는 다음과 같은 등급결정을 한다.

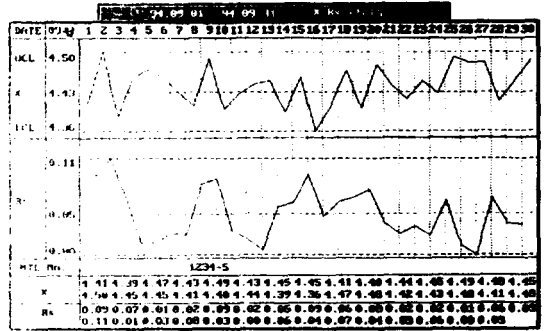
- 1 등급: Cp >= 1.33 공정능력 매우 양호
- 2 등급: 1.33 > Cp >= 1.00 공정능력 양호
- 3 등급: 1.00 > Cp >= 0.67 공정능력 부족
- 4 등급: 0.67 > Cp 공정능력 매우부족

4.3 X-Rs 관리도

일정기간(30일 이내) 동안의 한 모델에 대한 가우스 평균의 변화량을 보여준다. 가우스 평균의 변화량은

절대값으로 표시하며 작업을 하지 않은 날은 빼고 처리한다. 계산과정은 간단하지만 30일치의 데이터를 분석해야 하므로 시간이 많이 걸린다. X의 계산은 하루의 가우스평균을 계산한 것이고, Rs라는 것은 전날 X와 다음날 X의 차이를 계산한 것이다. (Rs = Xn - Xn-1)

그림 19은 한 모델에 대한 하루의 가우스 평균(X)을 날짜별로 값들을 찍은선 그래프로 그렸으며 한 달 중 평균이 가장 높은 값과 낮은 값을 HCL과 LCL로 표시하였으며, 전날의 가우스 평균값과 다음날의 가우스 평균값의 차이(Rs)를 계산하여 그 값들을 찍은선 그래프로 그리고 가장 높은 값과 낮은 값을 표시하였다.



(그림 19) X-Rs 관리도 출력

(Fig. 19) Output of X-Rs management table

4.4 데이터 베이스 시스템 구성

데이터 분석을 하기위해 자체적으로 데이터 베이스 시스템을 구성한다. 간단한 방법으로 데이터를 저장하거나 불러오게 되며 처리 속도를 빠르게 하기 보다는 많은 데이터를 처리하는데 중점을 두었다.

1) DataBase system 구조

InformationStruct

ChakJaGiNo[MAX_CHAKJAGINO +1]	작자기번호[5 +1]
Worker[MAX_WORKER +1]	적업자명[10 +1]

InformationStruct라는 것은 하루 작업에 대한 기본 레코드가 되는데 여기에는 작업자, 작자기 번호가 들어간다. 이 데이터는 생성화일의 첫부분에 있으며 한

번만 기입한다.

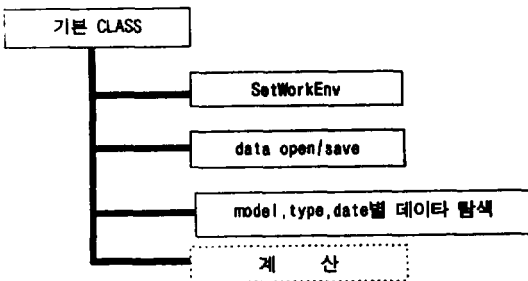
OutDataStruct

V[$\text{MAX_V} + 1$]	착자전압[4 + 1]
I[$\text{MAX_I} + 1$]	착자전류[5 + 1]
G[$\text{MAX_G} + 1$]	가우스[7 + 1]
ModelName[$\text{MAX_MODELNAME} + 1$]	모델이름[13 + 1]
Type[$\text{MAX_TYPE} + 1$]	모델타입[2 + 1]
Worktime[$\text{MAX_WORKTIME} + 1$]	작업시간[4 + 1]

OutDataStruct는 착자수량만큼의 레코드가 생성되며 모델, 타입이 바뀔 경우는 변화된 레코드만이 바뀌어 저장 된다. 파일명은 yy_mm_dd.o00 식으로 저장 된다. 확장자 뒤에는 숫자가 붙는데 이것은 각 컴퓨터 마다 고유 ID에 해당하며 여러대의 컴퓨터를 가지고 작업시 분류하기 위한 코드이다.

2) DataBase System 구축 기법

기본클래스는 작업 이력서, 작업 일보, 돛수 분포표, 히스토그램, X-Rs 관리도 클래스에 상속되며 계산함수는 가상함수으로써, 각각의 툴들은 기본 클래스를 상속받아 각각 해당한 계산을 행하고, 분석 및 출력을 행한다.



(그림 20) 데이터 처리 기본 클래스 구조
(Fig. 20) Configuration of basic class for data processing

5. 결 론

자동 착자 및 검사 자동화 시스템의 개발로 착자 작업의 자동화를 이룩하여 수작업에 의한 착자 작업

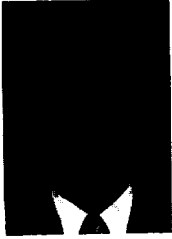
보다 정확하고 빠른 작업을 구현하였고, 가우스 값 합/불 판정의 자동화를 실현하여 품질과 생산성을 향상시켰으며 작업시 발생하는 데이터를 저장, 관리하여 제품의 품질 관리와 출하된 제품의 관리가 가능하게 되었다.

자동 착자 시스템의 적용 효과는 다음과 같다. 첫째, 작업 이력서의 파일 관리로 매 작업시 시면화되는 불편함을 줄이게 되었다. 둘째, 수작업이 적어지므로 사람의 오판등에 의한 실수가 줄어들어 제품의 불량률이 감소 되었으므로 품질이 향상되었다. 셋째, 착자와 합/불판정의 자동화로 생산성이 증가하고 착자 및 검사의 신뢰성이 향상되었다. 넷째, 정확한 데이터의 생성으로 세밀한 분석이 가능하였다.

앞으로의 연구과제는 LAN 개념을 도입하여 각각의 착자자동화 및 검사 시스템과 작업 데이터를 통합적으로 관리할 수 있는 다중작업에 대한 연구가 필요하다.

참 고 문 헌

- [1] Intel, "Micro Processors and Peripheral handbook," vol.2, Peripheral, Intel Corporation, pp. 4-34~4-37, 1988.
- [2] National Semiconductor, "Programmable Logic Devices" Databook and design guide, pp.2-171~2-174, 1992.
- [3] Mohamed Rafiguzzaman, "Micro Processors and Micro computer based system design," CRC Press, pp.75~80, 1990.
- [4]Girslin, G. "16bit modern micro computers," The Intel 8086 family, Prentice-Hall, pp.43~60, 1985.
- [5] Motorola, "MC68040 user's manual," pp.53~73, 1993.
- [6] Carroll, John M. "Designing interaction psychology at the human-computer interface," Cambridge university press, NewYork, pp.171~192, 1991.
- [7] G. Boothroyd and P. Dewhurst, Handbook design for assembly, University of Massachusetts, Ancherst, pp.158~175, 1984.
- [8] Joseph J. Carr, "Micro Computer Interfacing Handbook: A/D & D/A," TAB Press, pp.235~257, 1991.



이 천 희

- 1971년 한양대학교 전자공학과 졸업(학사)
- 1975년 성균관대학교 대학원 전자자료처리과 졸업(석사)
- 1981년 한양대학교 대학원 전자공학과 졸업(석사)

1987년 성균관대학교 대학원 전자공학과 졸업(박사)
 1971년 2월~1972년 8월 한국 마벨(전자업체)
 1972년 9월~1977년 2월 수송 전기 공업고등학교 교사
 1977년 3월~1979년 2월 동양공업전문대학 전자과 전임강사
 1979년 3월~현재 청주대학교 전자공학과 교수
 1983년 8월~1985년 8월 미국 산호세 캘리포니아 주립대학교 전산과 객원교수

관심분야: CAD & VLSI



하 기 종

- 1981년 광운대학교 응용전자공학과 졸업(공학사)
- 1986년 중앙대학교 대학원 전자공학과 졸업(공학석사)
- 1995년 청주대학교 대학원 전자공학과 박사과정 수료
- 1986년~1994년 한국통신 연구개발본부 전임연구원

1994년~현재 영동전문대학 전자통신과 전임강사
 관심분야: VLSI & CAD, 실시간 정보처리, 컴퓨터 네트워크