공정 감시용 칼라 그래픽 모니터링 시스템의 개발

채영석*, 임준홍*, 조영조**, 오상록**, 김광배**
*한양 대학교 전자공학과, **한국과학기술연구원 제어시스템연구실

Design of Color Graphics Monitoring System

Y.S.Chae*, J.H.Lim*, Y.J.Cho**, S.R.Oh**, K.B.Kim**
*Dept. of Electronic Eng., Hanyang University, **Control System Lab, KIST.

1. 서론

본 논문에서는 연속공정자동화를 위한 시스템 제어 장치 중 공정의 제반운영 및 제어상태를 감시하기 위하여 공정상태를 나타내는 여러가지의 data를 그래픽 모니터상에 표시해 주는 user Interface System 으로 칼라 그래픽 모니터링 시스템의 설계 및 구현을 다룬다. 본 연구에서의 연속공정 시스템 제어 장치는 그림 1과 같이 3개의 MC68030 CPU를 기초로 하는 processor 들이 global bus를 공유하며 network Interface, Line Control, Supervisory Control & Operator Interface (SC & OI)를 담당한다. 본 논문에서의 칼라 그래픽 모니터링 시스템은 Operator Terminal 로서 SC & OI processor와 접속하여 운영되도록 설계한다.

Operator terminal은 공정의 제반 상태를 감시하기 위하여 공정상태를 나타내는 여러가지 data를 monitor 상에 graphics로 표시해 주는 User Interface System이다. 이를 위하여 operator terminal은 출력장치로 VGA board와 color monitor, 입력장치로는 mouse와 function keyboard, 통신을 위한 RS232C serial port를 갖는 IBM-PC AT급의 mother board로 구성된다. 또한 program 저장은 HDD를 이용한다. Monitoring을 위한 operator terminal은 다음의 기능을 갖는다.

(1) User와의 interface를 위한 메뉴화면의 제공
(2) Operator 조작이 용이한 입력 장치들의 구동
(3) 새로운 화면을 graphic symbol을 이용하여 편집할 수 있는 graphic 화면 편집 기능
(4) 편집된 graphic symbol을 취득한 공정 data와 matching시켜 data를 graphic 화면을 통하여 monitoring 하는 기능
(5) Data의 요구, 취득을 위한 host와의 interface 기능
(6) Monitoring 하는 graphic 화면들이 계층적 구조를 갖게하여 용이한 선택이 이루어지게 하는 관리 기능

2. Operator Terminal의 구조

가. Hardware의 구조

Operator terminal의 hardware 구성은 그림 2와 같다. IBM PC AT급의 mother board를 주 CPU로 하여 color monitor driving을 위한 VGA card와 2개의 RS232C 직렬 port가 있다. Serial port COM1은 SC & OI processor와의 통신에 이용되고, COM2는 입력장치인 mouse 구동에 사용된다. ROM card는 power on 시 system booting, initialize, program 과 file data loading 등의 routine등이 저장된다.

나. Software의 구조

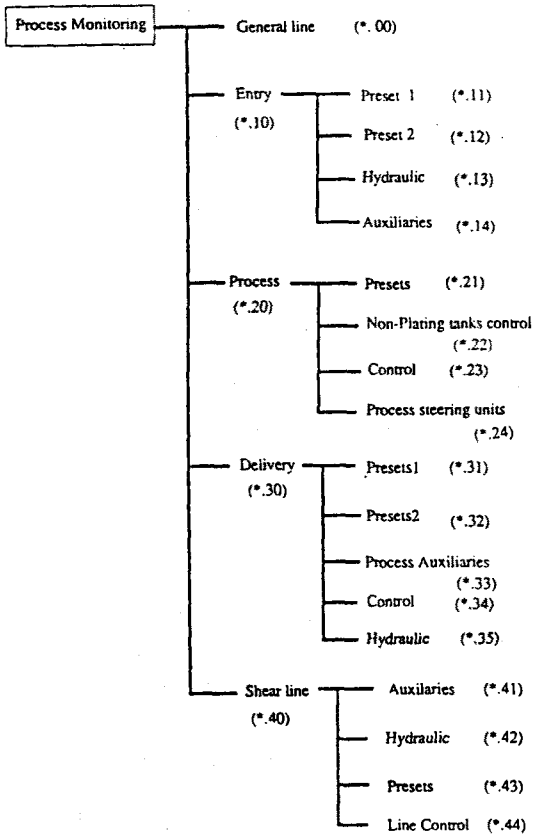
Operator terminal의 software는 그림 2와 같이 file관리, graphic edit, process monitoring의 세부분으로 구성된다. File관리에서는 monitoring과 editing을 위한 graphic file 들의 이름을 선택, 추가할 수 있도록 되어 있다. Graphic file 이름들은 extension을 가지며, 이 extention으로 monitoring시에 어느 계층의 file인가를 알 수 있게 한다.

Graphic edit는 file관리에서 선택된 file을 편집하여 저장하는 기능이다. Process monitoring은 선택된 file을 graph로 보여준다. Static화면 정보 뿐만 아니라 필요한 dynamic data를 일정시간에 SC & OI processor 와의 통신을 통하여 취득하여 화면에 update 시켜 준다.

3. File 관리

이 menu는 전체 그림을 가지는 file 들을 보여주는 menu이다. 여기에는 선택, 추가, 지움, 종료의 네 부분으로 구성되어 있어 원하는 file을 선택, list에 추가, list에서 삭제할 수 있다.

그림 file들은 계층적 구조를 가지며 이의 전체 모습은 다음과 같다.



file menu에서 display되는 filename에서 이 계층적 구조를 보여주기 위해서 일정한 규칙을 가지게 filename의 extension을 정하도록 되어 있다. 즉, filename.XY에서 filename은 임의로 정할 수 있고, 최대 8자까지 가능하도록 되어있고, 확장자는 다음과 같이 정해지도록 한다.

Filename.XY

X = '0' : General Line Control

'1' : Entry

'2' : Process

'3' : Delivery

'4' : Shear Line

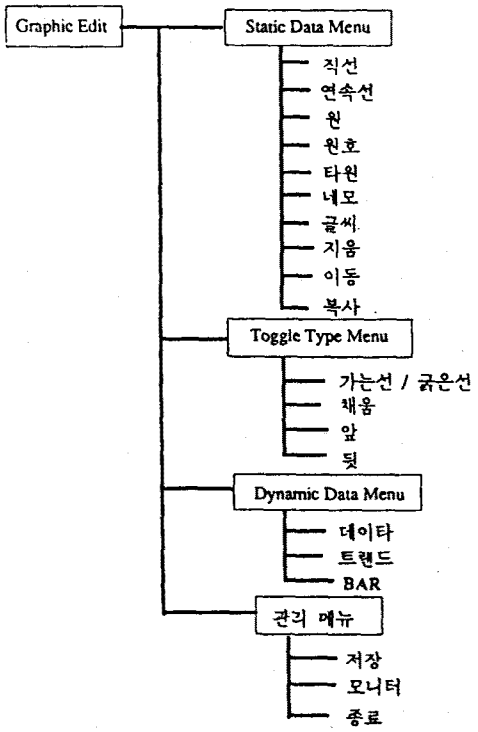
Y = '0' : 각각의 Section에서 main 화면

'1' :
'2' :
'3' :
'4' :
'5' :
'6' :
— 각각의 Section에 속하는 그이하 화면

이와 같이 filename의 확장자를 의미있게 정해주므로 해서 전체 file에서 어떤 file이 어디에 속해 있는가를 한 눈에 알 수있게 해준다.

4. Graphic Edit

이 menu의 전체 구조는 다음과 같다.



Static Data Menu는 monitoring 시에 정지되어있는 정적인 graph data를 편집하며, Toggle Type Menu는 각 그림의 attribute와 뒷 배경및 앞 화면의 색을 지정해준다. Dynamic Data Menu는 Monitoring 해야할 Data, On/Off, BAR와 Trend의 화면을 편집하는데 사용된다. 그림을 저장할 때 다음과 같은 structure로 구성되어 있다.

(1) 직선, 연속선, 원, 타원, 네모

```
Struct Sdb_graph {
        char    func;       /* function name */
        int     color;      /* 색깔 */
        int     width;      /* 가는선 / 굵은선 */
        int     X1, y1;     /* 첫번째 좌표 */
        int     X2, y2;     /* 두번째 좌표 */
        int     fill;       /* 채움 ? */
};
```

(2) 원호

```
Struct arc_graph {
        char    func;       /* function name */
        int     color;      /* 색깔 */
        int     width;      /* 가는선 / 굵은선 */
        int     X1, y1;     /* 첫번째 좌표 */
        int     X2, y2;     /* 두번째 좌표 */
        int     X3, y3;     /* 세번째 좌표 */
```

(3) 글씨

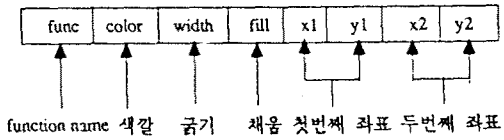
```
Struct udb_data {
        char    func ;          /* function name */
        int     f col ;         /* 글씨 색깔 */
        int     b col ;         /* 글씨 배경 색깔 */
        int     x, y ;          /* 글씨 기준 좌표 */
        int     Str[MAX_STR] ;  /* 두번째 좌표 */
} ;
```

(4) 데이타, 트랜드, BAR, ON/OFF

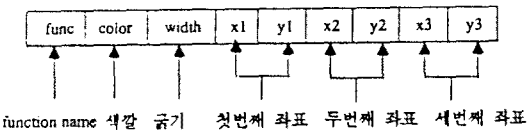
```
Struct dynamic {
        char    func ;          /* function name */
        int     f color ;       /* foreground color */
        int     b color ;       /* back ground color */
        int     x1, y1 ;        /* 첫번째 좌표 */
        int     X2, y2 ;        /* 두번째 좌표 */
        int     Str[MAX_STR] ;  /* label name */
} ;
```

또한 저장된 file에는 다음과 같은 모양으로 저장된다.

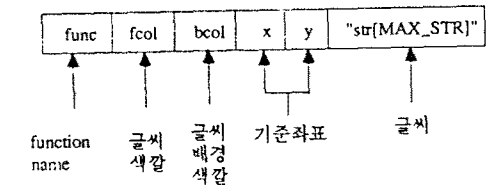
(1) 직선, 연속선, 원, 타원, 네모



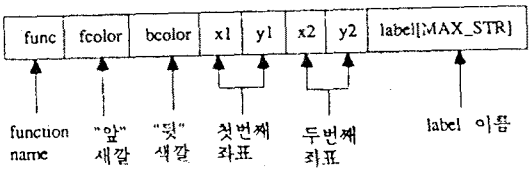
(2) 원호



(3) 글씨



(4) 데이타, 트랜드, BAR, ON/OFF



5. Process Monitoring

이 menu는 RS232C를 통하여 전송되는 data를 받아서 해당되는 data의 변화되는 모습을 그림을 통하여 볼 수 있게 해 주는 기능을 한다. 여기에는 편집,종료 그리고 5개의 section으로 들어가서 다른 그림들을 monitoring 할 수 있는 menu 선택 기능이 있다. monitoring mode로 들어가면 현재 file을 load하고 화면 아래 section에서 현재 file이 속해 있는 section이 빨간색으로 칠해진다.

Monitoring menu에서는 SC & OI processor에서 RS232C를 통해서 보내 오는 data를 handshaking 방식을 통하여 화면을 update한다. 여기서 현재 file을 load 한 후 dynamic data가 있는가를 판단한 후 있으면 dynamic data의 label을 string으로 만든 후 "FL"을 RS232C로 보낸다. 그리고 "OK!!"를 받으면 이미 만들어 놓은 label string을 전송하고, 다시 "OK!!"가 오기를 기다린다. "OK!!"가 오면 data를 받기 위해 "FS!!"를 보낸 후, 이때 받은 data의 갯수와 label갯수가 같을때 까지 계속 data를 받는다. Data갯수가 같으면 화면을 update한 후, 다시 "FS!!"를 보내고 data를 받는 과정을 종료를 누를 때까지 계속한다. 종료가 눌러지면 "FE!!"를 보내고 "OK!!"를 받으면 화면 update를 중지하고 맨처음 화면으로 다시 되돌아 가고 section을 선택하면 다른 원하는 file을 load한 후 다시 data를 받아서 화면을 update하는 과정을 다시 하게 된다. 편집을 누르면 RS232C를 받아버리고 편집 file을 편집할 수 있는 graphic edit mode간다.

6. 구현결과 및 결론

본 연구에서 설계된 Operator Terminal의 구현 결과 중 일부는 사진 1,2,3에 보인다. 사진 1은 graph editor에서 general line control 화면을 편집한 것을 보여준다. 사진 2,3은 monitoring 화면으로 사진 2는 monitoring을 시작하는 화면이고, 사진 3은 monitoring하는 file을 선택하는 화면이다. 전체 제어시스템 장치에 연결되어 SC & OI processor와의 data 교류는 48개의 실제 공정 data를 취하며 화면을 update 하는 시간을 1초 간격으로 하여 구현하였을때 오류없이 실제 data를 monitoring 함을 확인하였다. 편의상 update 하는 시간을 1초로 하였고 그 보다 빠르게 하더라도 RS232C 통신에 필요한 시간과 화면 update 시간 보다만 크게 하면 실시간 monitoring이 가능함을 확인되었다.

앞으로의 연구 과제는 graphic editor의 기능 강화와 animation을 위한 editing 및 monitoring 방법의 연구 등이 있다.

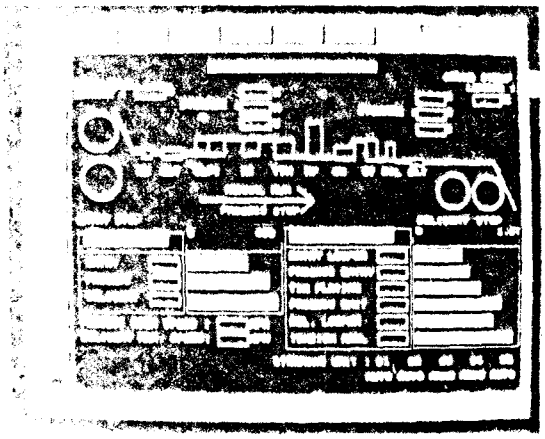


사진 1. Editor 화면

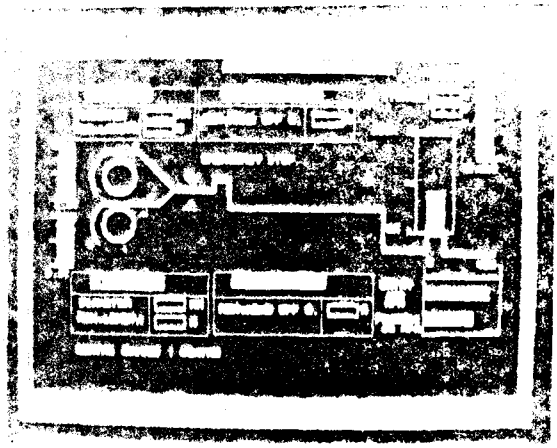


사진 2. Monitoring 화면

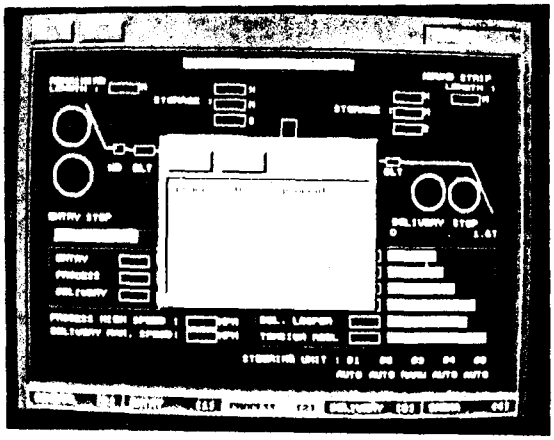


사진 3. Section 선택 화면

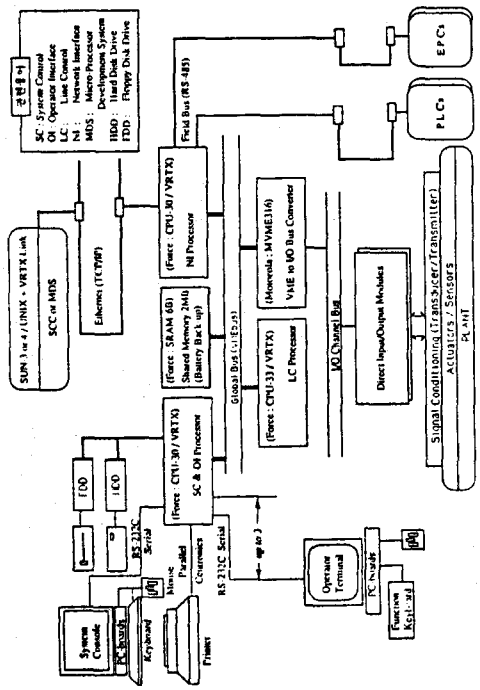


그림 1. 연속공정시스템 제어장치

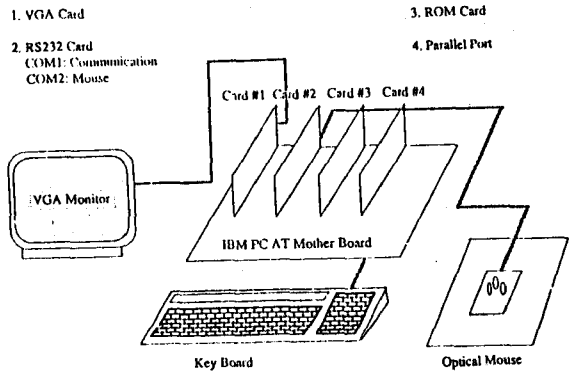


그림 2. H/W 구성

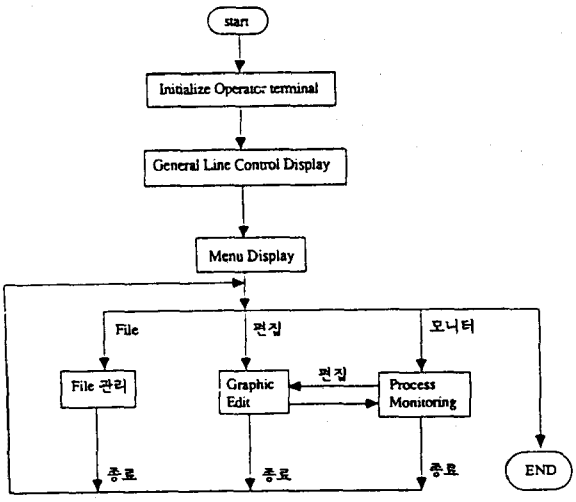


그림 3. 전체 S/W 구조

참 고 문 헌

1. 이창구, 김성준, "컴퓨터 공정제어", 전기 학회지, 제 35권, 제 12호, pp 776-781, 1986
2. 한국 전기통신 연구소, "공정제어 그래픽 시스템의개발에 관한 연구", 과기처 특정 연구최종 보고서, 1985
3. T.J.Miller, "ease of use, simplicity, flexiblityare key requirement for process control graphics", Control Engineering, pp 63-66, June, 1983
4. A.J.Labuzinsky,"Pixel-based software: ease pain of real time color graphics developement", Control Engineering , pp96 - 97, March, 1985
5. C.R.Berg, "Computer graphics displays : Window for process control", IEEE CG & A, pp 43-55, 1985
6. 조영조, 임준홍 외, "연속 공정 자동화를 위한 전동기 그룹제어 시스템의 개발", '90, 한국 자동제어 학술 회의 논문집, vol.1, pp. 218-224, 1990