'92 韓國自動制御學術會議論文集 1992. 10. 19~21

공정 감시를 위한 Software의 개발

가 민 호*, 임 준 홍*, 조 영 조**, 김 광 배**

*한양대학교 전자공학과 ,**한국과학기술연구원 제어시스템연구실

Design of Color Graphic Monitoring System

M. H. Ka*, J. H. Lim*, Y. J. Cho**, K. B. Kim**

*Dept. of Electronics Eng.,Hanyang Univ. , **Control System Lab.,KIST

요 약

본 논문에서는 연속공정자동화를 위한 시스템 제어 장치 중 공정의 제반운영 및 제어상태를 감시하기 위하여 공정상태를 나타내는 여러가지의 data를 그래픽 모니터상에 graphics로 표시해 주는 User Interface System인 Operator Terminal의 설계 및 구현을 다룬다. 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 상위 computer의 HDD를 이용한다. 그래픽 편집 및 모니터링을 위한 software는 한글 MS Windows 환경에서 구현한다.

1. 서 론

미래 산업 설비의 궁극적인 목표인 무인화 공장을 위하여 점차 가속화되고 있는 공장 자동화에 대한 자체 연구 개발은 필수 불가결하다. 이러한 공장자동화를 위한 필요 기술은 생산 라인의 필수 구동 장치인 전동기의 디지탈 제어 시스템 개발, PLC (programmable logic controller) 및 이송 장치 개발, 각종 센서를 이용한 부품 검색및 처리 기술, 전 공정의 감시 제어 시스템 개발및 효과적인 Man-Machine interface 개발등의 High Technology 가 필요하다. 특히 연속 공정 자동화 시스템에서는 단위 요소 기술 개발은 물론 Distributed Control System 및 전동기 Group Controller 등의 시스템 종합 기술이 요구된다. 즉, 연속공정 자동화 시스템은 다수 전동기들의 연동을 조절하는 전동기 Group controller와, 기계 가공 공정이나 자동차 조립 및 용접공정 등에서 사용되던 PLC(Programmable Logic Controller) 등 지금까지 응답특성이 비교적 느린 공정에 적용되어 왔으나 MMI(Man-Machine Interface)가 우수한 DCS(Distributed Control System)등이 총체적으로 결합된 종합제어 시스템이다. 이러한 기술중에서 본 논문에서는 공정의 운전 상태에 대한 효과적인 감시및 제어 방식으로 가장 효율적인 것으로 알려져 있는 Man-Machine interface를 위하여 공정 제어 그래픽 시스템 (graphic console system) 을 위한 공정도 편집기와 모니터링 시스템을 한글 MS Windows를 이용하여 개발하였다.

공정 제어 그래픽 시스템은 공정 운전자가 공정을 감시하거나 제어할 수 있도록 공정이나 data base로부터 온 dynamic data를 CRT screen 상에 그림으로 효과적으로 나타낼수 있도록 하는 hardware 및 software의 총합으로 정의될 수 있다. 또한 공정의 운영상태를 monitoring하는 기능외에 공정의 data를 계산,처리하거나 조작자의 명령을 계층 구조상 말단에 해당하는 loop제어기의 PLC등에 하달하는 기능을 가짐으로써 중앙제어 console로서의 역할을 아울러 담당하고 있다. 이러한 시스템을 통칭하여 Graphic Console System(GCS)이라 한다. 특히 GCS는 많은 양의 data를 그림으로 쉽게 표현해주고 공정의 schmetic diagram등으로 부터 쉽게 명령을 전달할 수 있으므로 효과적인 Man-Machine Interface라고 할 수 있다. 그러나 기존의 공정제어 그래픽 시스템을 공정의 data를 취하거나 명령을 전달하는 등 하부 구조와의 정보교환을 위하여 고유의 통신방식을 사용하기 때문에 하부의 공정제어기도 반드시 그 방식을 따르는

comnunication interface장치및 관련 software를 갖추어야 한다. 따라서 실제 적용시 정보교환 기능을 하는 복잡한 computer통신 system이 공정 제어 그래픽 시스템과 공정제어 중간에 요구되어, 비용이 많이 소요될 뿐만아니라 computer와 제어및 통신의 복합적인 시스템기술이 필요함으로 hardware와 software의 구조가 복잡해진다. 또한 국내에서 사용되는 공정제어 시스템은 외국으로부터 도입되어 운영되는 실정이다. 따라서 국내여건에 잘 맞지 않아 사용하기 매우 복잡하고 보수, 유지가 쉽지 않다. 그러므로 기존의 공정제어 그래픽 시스템과 비교하여 간단한 구조와 적은 비용으로 공정의 운영상태를 실시간으로 감시, 조절할 수 있으며, 확장성이 있고, 국내여건에 적합하며, 운전자가 쉽게 사용할 수 있는 GCS의 개발이 필요하다. 본 논문에서는 연속공정자동화를 위한 시스템 제어장치 중 공정의 제반운영 및 제어상태를 감시하기 위하여 공정상태를 나타내는 여러가지의 data를 그래픽 모니터상에 표시해 주는 Graphic console system에서 칼라 그래픽 모니터링 시스템의 설계 및 구현을 다룬다. 본 논문에서의 연속공정 시스템 제어 장치의 주 연산부는 MC68030 CPU를 기초로 하고 4MB의 main memory 및 Disk interface 장치를 내장한 Single Board Computer CPU-30이 담당한다. System Console은 processor들이 global bus를 공유하며 network Interface, Line Control, Supervisory Control & Operator Interface (SC & OI)를 담당한다. 칼라 그래픽 모니터링 시스템은 Operator Terminal 로서 SC & OI processor와 접속하여 운영되도록 설계한다.

2. Operator Terminal의 기능 및 구조

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 data 와 matching 시켜 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

본 연구에서는 이러한 operator terminal을 한글 MS Windows를 이용하여 구현했는데 Windows를 사용하므로써 다음과 같은 장점을 가지게 된다.

(1) 그래픽 사용자 인터페이스

Windows의 GUI(Graphic User Interface) 환경은 user 가 동시에 여러 응용 프로그램에 접근할 수 있도록 해줌으로써 현재 수행 중인 작업을 일목요연하게 파악할 수 있도록 해주며, 또 거의 모든 응용 프로그램들이 유사한 그래픽 사용자 인터페이스를 가지고 있기 때문에 처음 대하는 응용프로그램이라도 사용자가 금방 익숙할 수 있도록 해준다.

(2) 큐화된 입력

DOS환경 에서와는 달리 Windows 환경에서는 입력을 하면 이를 응용 프로그램 큐에 저장하고 나중에 각 응용 프로그램들은 큐에 저장된 입력을 수행한다. 이런 입력처리방식은 다른 응용프로그램이 수행될 때에도 입력을 받아 들일 수 있어 멀티태스킹이 가능하게 된다.

(3) Hardware에 의존하지 않는 디바이스 드라이버

Windows가 독립적 디바이스 드라이버를 가진다는 점은 응용프로그램을 작성하는 데 가장 큰 도움을 주는 요인으로, 응용프로그램을 작성할때기종에따른 디바이스 드라이버를 따로이 만들 필요가 없다.

Windows는 한가지 디바이스에 대해서는 디바이스 드라이버의 제작자가 이를 작성 하도록 함으로써 사용자가 이를 작성할 필요가 없도록 해준다.
예를들어 Windows는 표준 마우스로 버튼이 세 개인 마우스를 지정해놓고 이에 대한 입력만을 처리한다.

(4) 멀티태스킹

Windows는 DOS의 단점인 멀티태스킹, 다중작업을 지원하는 운영체계이다. 다시말해, 한 번에 여러개의 응용 프로그램을 실행시키거나 하나의 응용프로그램을 여러 번 수행시킬 수 있는 장점이 있다.

2.1 Hardware의 구조

Operator terminal의 hardware 구성은 다음과 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.

2.2 Software의 구조

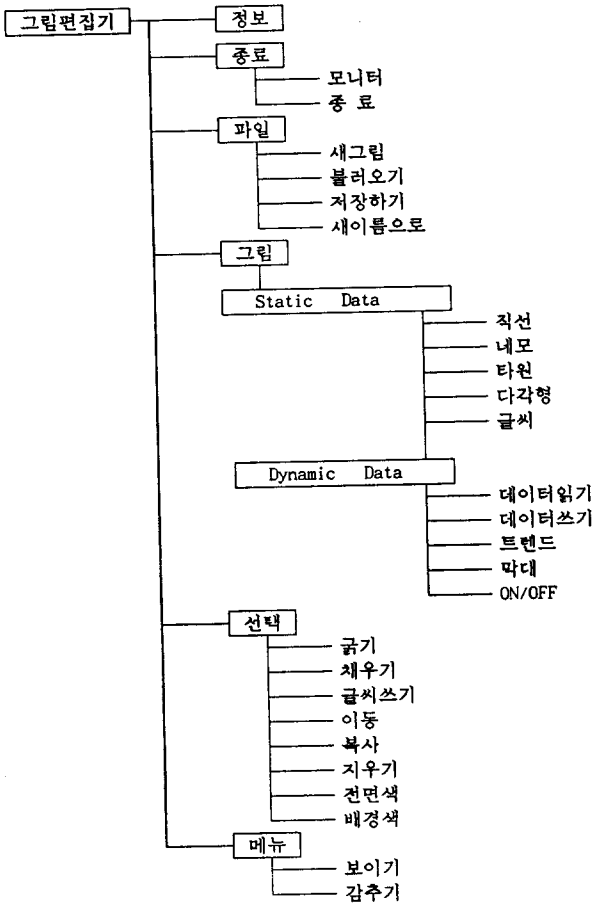
Operator terminal의 software는 File관리, 그림 편집기, 모니터링의 부분으로 구성된다. File관리에서는 모니터링과 그림 편집을 위한 graphic file 들의 이름을 선택, 추가할 수 있도록 되어 있다. Graphic file 이름들은 각각 Area와 Section number를 가지며, 이것으로 모니터링 시에 어느 계층의 file인가를 알 수 있게 한다.

그림 편집기는 file 불러오기에서 선택된 file을 편집하여 저장하는 기능이다. 모니터링은 선택된 file을 graph로 보여준다. Static화면 정보 뿐만 아니라 필요한 dynamic data를 일정시간에 SC & OI processor 와의 통신을 통하여 취득하여 화면에 update 시켜 준다.

3. 그림 편집기

3.1 Menu 구조

그림 편집기는 다음과 같은 Pull-down Menu와 Pop-up Menu로 구성되어 있다.



3.2 자료 구조

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 (A)

```
struct _ResLine {
        int left;       /*****
        int top;          좌
        int right;        표
        int bottom;     *****/
        int width;      /* 굵 기 */
        int color;      /* 전면색 */
}
```

(2) 다각형(B)

```
struct _ResPoly {
        int witdth;      /* 굵 기 */
        int count;       /* 직선의 수 */
        BOOL fill;       /* 채우기 */
        int color;       /* 전면색 */
        int bcolor;      /* 배경색 */
        Px;              /***
        Py;               좌 표 ***/
      }
```

(3) 네모(C), 타원(D)

```
struct _ResRect/ResElip {
        int left;        /******
        int top;            좌
        int right;          표
        int bottom;      *******/
        int width;       /* 굵 기 */
        BOOL fill;       /* 채우기 */
        int color;       /* 전면색 */
        int bcolor;      /* 배경색 */
       }
```

(4) 글씨(E)

```
struct _ResStr {
        Px;              /**
        Py;                 좌 표 **/
        int length;      /* 길 이 */
        char str;        /* 입력된 문자 */
        int color;       /* 전면색 */
        int font;        /* 글자크기 */
        }
```

(5) 막대(F), 트랜드(G)

```
struct _ResBar/_ResTrend {
   char id;          /* Label 이름 */
   int left;         /*******
   int top;             좌
   int right;           표
   int bottom;      ********/
   int color;        /* 전면색 */
   int bcolor;       /* 배경색 */
   }
```

(6) ON/OFF(H)

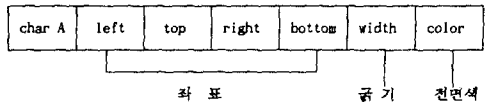
```
struct _ResOnOff {
   char id;          /* Label 이름 */
   Px;               /*****
   Py;                좌 표 ****/
   int onoffcheck;   /* ON/OFF checking */
   int color;        /* 전면색 */
   int bcolor;       /* 배경색 */
   }
```

(7) 데이터읽기(I), 데이터쓰기(J)

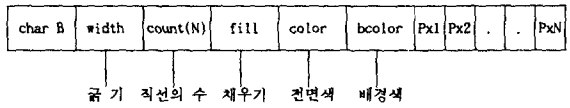
```
struct _DataR/W
   char id;          /* Label 이름 */
   Px;               /********
   Py;               좌 표 ***/
   int color;        /* 전면색 */
   int bcolor;       /* 배경색 */
   }
```

또한 저장된 file에는 다음과 같은 모양으로 저장된다.

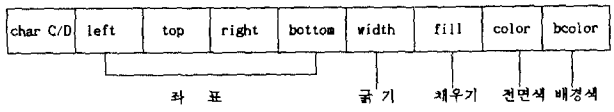
(1) 직선



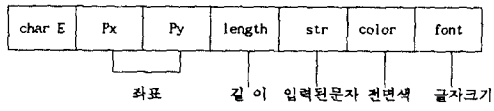
(2) 다각형



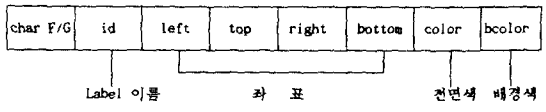
(3) 네모, 타원



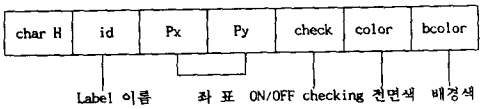
(4) 글씨



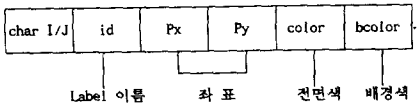
(5) 막대, 트랜드



(6) ON/OFF

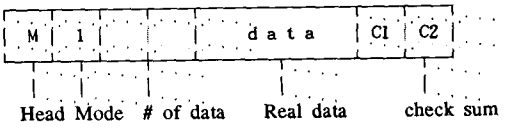


(7) DataRead, DataWrite



(1) 통신 Protocol

- 통신 Packet



.Head : 'M '

.Mode 1 : One line data

2 : data with next data

3 : end of mode 2

4 : data for file transmition

5 : end of file data

4. 모니터링

4.1 메뉴 구조

이 menu는 RS232C를 통하여 전송되는 data를 받아서 해당되는 data의 변화되는 모습을 그림을 통하여 볼 수 있게 해 주는 기능을 한다. 여기에는 실행, 종료 그리고 5개의 Area와 section을 선택해서 display된 그림들을 monitoring 할 수 있는 data file 불러오기 기능이 있다. 모니터링에서는 현재 file을 load하고 load된 data file을 모니터링하게 된다.

모니터링 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 Label list를 string으로 만든 후 만들어 놓은 label string을 전송한다. Label list의 전송이 끝나면 실행 menu를 선택하여 실제 data값을 받기 위해 "DS"라는 명령을 보낸 후, 이때 받은 data의 갯수와 label갯수가 같을때 까지 계속 data를 받는다. Data갯수가 같으면 화면을 update 한 후, 다시 "DS"를 보내고 data를 받는 과정을 종료를 누를 때까지 Sampleling time에 따라 계속하게 된다. 종료가 눌러지면 Dynamic data의 전송을 중지하고 화면 update를 중지하며 맨처음 화면으로 다시 되돌아 가고 file 불러오기를 선택하면 다른 원하는 file을 load한 후 다시 data를 받아서 화면을 update하는 과정을 다시 하게 된다.

4.2 통신 방식

(2) 통신에 사용되는 명령어

1. Data File 입,출력
   - FS [*filename*] : Operator terminal에서 편집된 화면 data를 CPU30에 저장하는 명령이다.
   - FL [*filename*] : 이미 편집되어 있는 공정화면을 CPU30으로 부터 operator terminal로 불러오는 명령이다.
   - FD [*directory name*] : Operator terminal에서 Directory list를 요구하는 명령이다.

2. Dynamic Data 입,출력
   - DS : Operator terminal에서 dynamic data를 요구하는 명령.
   - DL < C > : Label list를 operator terminal에서 요구하는 명령.
     < C > : Label의 첫 문자를 나타낸다.
   - DD : 공정 화면 중의 label list를 O.T.에서 CPU30으로 보낸다.
   - DP [*label name*][*data*] : Set point
   - DR : 실제 data의 minimum, maximun값을 요구하는 명령이다.

4.3 File 관리

File 불러오기에서 display되는 dialogbox에서 file이 속한 Area와 Section을 선택할 수 있다. Area에는 General line[0], Entry[1], Process[2], Delivery[3], Shear line[4] 들이 있으며 각각의 Area는 Section[0 ~ 5]을 가지게 되며 다음과 같은 구조로 되어 있다.

```
struct FileName  {
        char    filename ;          /* data filename */
        int     i ;                 /* # of Area */
        int     j ;                 /* # of Section */
       }
```

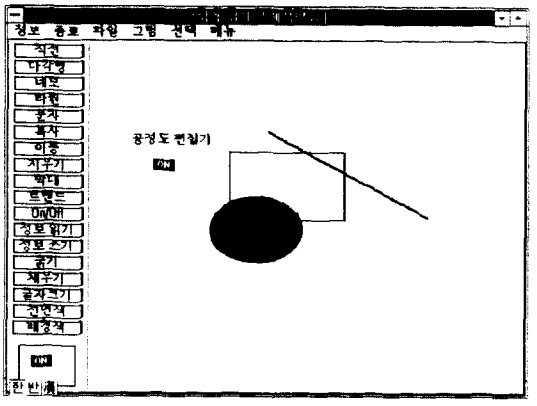


그림1. 그림편집기 화면

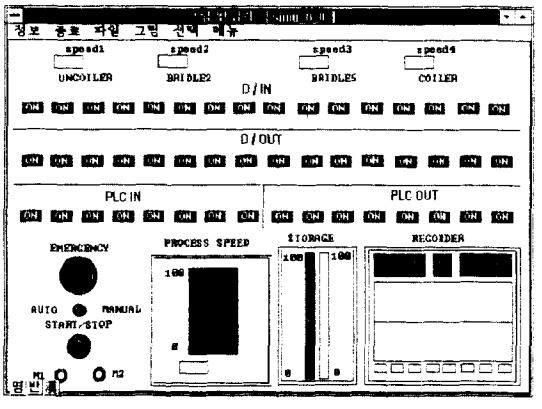


그림2. Monitoring 화면

5.구현결과 및 결론

본 연구에서 설계된 Operator Terminal의 구현 결과 중 일부는 그림 1,2에 보인다. 그림 1은 그림 편집기에서 graphic menu들이 display된 화면을 보여준다. 그림 2는 monitoring 화면으로 그림편집기로 편집한 simulator 화면을 monitoring 하는 그림이다. 전체 제어 시스템 장치에 연결되어 SC & OI process와의 data 교류는 48개의 실제 공정 data를 취하며 화면을 update 하는 시간을 1초 간격으로 하여 구현하였을때 오류없이 실제 data를 monitoring 함을 확인하였다. 화면을 update 하는 시간을 더 빠르게 하더라도 RS232C 통신에 필요한 시간과 화면 update 시간 보다만 크게 하면 실시간 monitoring이 가능함을 확인하였다.

참고 문헌

1. C.R.Berg, "Computer graphics displays : Window for process control", IEEE CG & A, pp. 43-55, 1985
2. A.J.Labuzinsky, "Pixel-based software: ease pain of real time color graphics developement", Control Engineering , pp. 96-97, March, 1985
3. 이창구, 김성준, "컴퓨터 공정제어", 전기 학회지, 제 35권, 제 12호, pp. 776-781, 1986
4. 한국 전기통신 연구소, "공정제어 그래픽 시스템의개발에 관한 연구", 과기처 특정 연구최종 보고서, 1985
5. 조영조, 임준홍 외, "연속 공정 자동화를 위한 전동기 그룹제어 시스템의 개발", '90, 한국 자동제어 학술 회의 논문집, vol.1, pp. 218-224, 1990
6. T.J.Miller, "ease of use, simplicity, flexibilityare key requirement for process control graphics", Control Engineering, pp 63-66, June, 1983