

화학공정 자동화를 위한 실시간대 다기능 소프트웨어의 개발

서인식\*, 김상우\*, 남성우\*, 백은환\*, 엄태원\*, 김원철\*, 김태운\*, 김흥식\*, 이광준\*\*  
\* 두산 연구소, \*\* 서강 대학교 화학공학과

Development of real time versatile software for automation of chemical processes

I.S. Seo\*, S.W. Kim\*, S.W. Nam\*, U.H. Paek\*, T.W. Eom\*\*, W.C. Kim\*\*, T.Y. Kim\*\*, H.S. Kim\*\*, K.S. I  
\* Doosan Research Lab. \*\* Sogang Univ., Chemical Eng.

ABSTRACT

In this work, we developed a real-time versatile advanced control and supervisory software for a personal computer control. This software, basically, has background and foreground tasks which are performed in parallel at real time. First, background tasks are composed of controls of various kinds, reports and input-output of signals etc, which are performed every sampling time. Second, foreground tasks are observation of operation conditions, data search, regulation of controllers and graphical design and display of processes, which are performed by user's request. Additionally, this software has the functions of transporting data and composing distributed control systems, and all background tasks are composed of combination of unit function blocks.

1. 서론

최근 많은 화학 공장들이 제어용 전산 시스템을 도입하여 운영하고 있다. 이러한 추세에 맞추어 소형 컴퓨터를 이용한 실시간대 다기능 제어용 소프트웨어를 개발하였다. 본 소프트웨어는 기본적으로, 정해진 시간마다 반복적으로 수행되어야 하는 background 작업과 조업자의 요구에의

해 수행되는 foreground 작업을 실시간대에서 병렬적으로 수행할 수 있도록 하였다. 또한 LAN system을 이용하여 분산 제어계의 구성을 가능하도록 하였으며, 각종 작업을 단위 블록으로 구성하여 여러가지 기능을 쉽게 조합할 수 있도록 하였다.

2. Software 구성

(1) 개요

본 software는 크게 조업자의 요구에 의해 수시로 행해지는 foreground tasks와 정해진 시간에 반복적으로 수행되는 background tasks의 두 부분으로 나눌 수 있다. 여기서 정기적으로 수행되는 작업들은 외부 timer에 의해 발생하는 interrupt에 의해 수행되도록 하였다. foreground tasks와 background tasks와의 자료 전달은 memory상의 절대번지를 이용하였다.

(2) Background tasks

Background tasks는 매 1초마다 외부에서 공급되는 interrupt에 의해 Interrupt Service Routine (ISR) 을 수행함으로써 이루어진다. ISR에는 그림 1 에서 보는 바와같이 수행해야 할 일들 - 신호의 입출력 및 변환, 각종 제어등 - 이 순차적으로 해당 시간마다 수행할 수 있도록 알고리즘을 작성하였다. 즉 ISR의 첫머리에서 시간을 계수하여 각 작업의 수행시간이 되었는가를 비교하여, 수행시간이 된 작업을 수행하고 돌아가도록 하며, 만약 수행시간

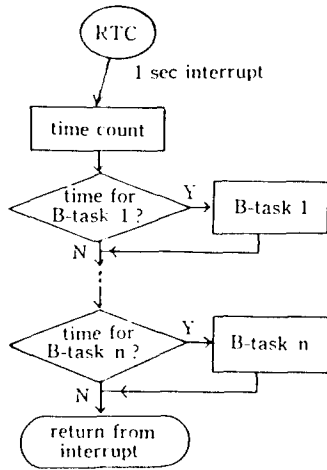


그림 1. Flowchart of background tasks

이 동일한 작업들은 flow chart 구성상 우선 순위가 높은 작업부터 수행하도록 하였다.

(3) Foreground tasks

Foreground tasks은 다른 computer와 자료교환을 제외하고는 모든 작업들이 keyboard의 입력이 있을 때마다 수행된다. foreground는 모든 작업을 그림 2 와같이 menu-style로 제공하며, 유사한 기능을 가진 작업들은 하나의 하위 menu를 갖도록 하였다. 다른 computer와 의 자료교환은 사실상 background에의해 수행 되어야 하나, ISR이 수행되는 중에는 DOS interrupt를 사용하지 못하므로, foreground에

서 정해진 시간마다 수행하도록 하였다.

Foreground에서 시간을 계수하는 방법은 foreground의 모든 일이 keyboard로부터 입력을 받아 수행하므로, 수시로 확인하는 그림 3 과 같은 keyboard interrupt service routine에서 약속된 절대변지상의 시간계수 변수의 값을 읽어 행한다.

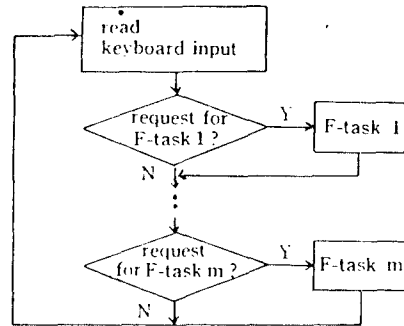


그림 3. Flowchart of foreground tasks

(4) graphic 기능

graphic 기능은 foreground에서 수행되며 크게 세 부분으로 나누어진다.

1) control state display ; 여기서는 제어상황을 실시간 대에서 bargraph로 볼 수 있고, tuning window를 이용하여 제어변수를 수정할 수 있어 on-line tuning이 자유롭다.

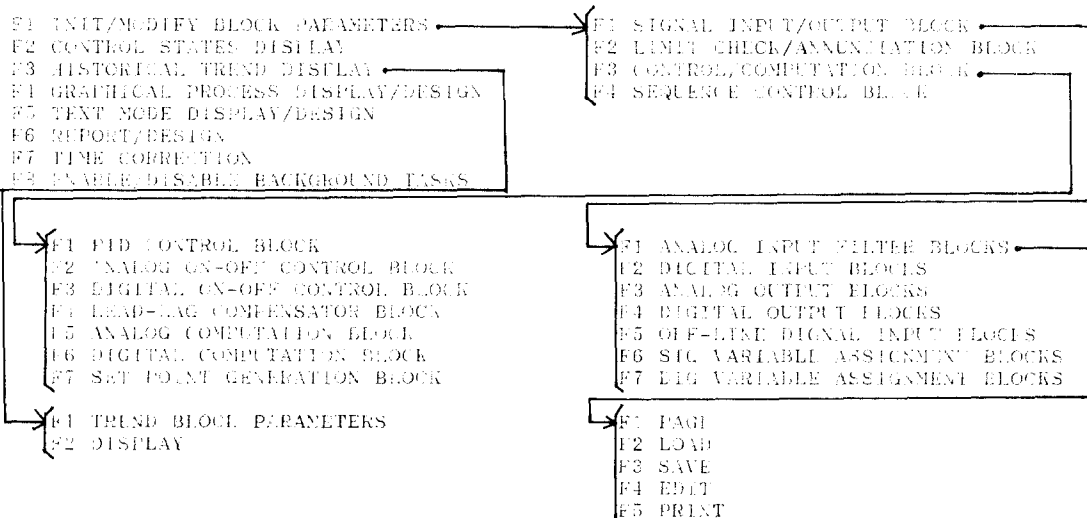


그림 2. 메뉴의 순위도 489

2) historical trend plot display; 모든 signal들의 과거치 200개를 graph로 볼 수 있으며 한 화면에서 4개 signal의 trend를 보면서 control state를 올리거나 tuning window를 이용하여 제어변수를 수정할 수 있다.

3) turtle graphics; 벨브, 펌프등 이미 준비된 그림을 이용하여 전체 공정도를 그릴 수 있고, 실시간대에서 변수값을 bar graph로 볼 수 있으며, 공정도를 보면서 control state를 보거나 tuning을 할 수 있다.

이상과 같은 multiple window 조절기능을 통하여 효율적인 공정감시가 가능하다.

(5) 메모리 관리

본 software를 소형 computer에서도 수행이 가능하도록 하기위해 전체 프로그램을 background tasks와 약간의 foreground tasks 및 program configuration 등이 포함된 주프로그램과 graphic 기능을 가진 보조 프로그램으로 나누었다. 수행방법은 주프로그램이 작동하다가 조업자의 요구에의해 예를 들어, turtle graphics을 나타내는 key가 눌러지면, 그림 4에서 보듯이 이 기능을 가진 execution file 이 load되어 수행한 후 원래 주프로그램으로 돌아오도록 구성하였다. 이때에도 주프로그램과 보조 프로그램사이의 자료교환은 메모리상의 약속된 부분을 서로 참조함으로써 가능하다.

(6) Block의 이용 방법

Background 작업들은 각종 단위 block들로 구성되어 있어 주어진 block들을 임의로 연결하여 새로운 기능을 갖도록 하였다. 예를 들어 computation/compention module의 block들과 control block들을 조합하면 cascade, feed-forward, selective, override 등의 제어계로 구성할 수 있다. 간단한 몇가지 기능 block은 표 1 에 보였다.

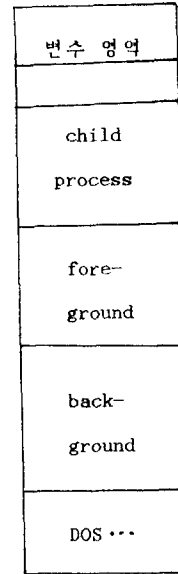


그림 4. 메인메모리의 구성

(7) 사용 언어

Background에서 화면에 날자와 시각을 나타내는 일을 포함한 약간의 작업을 제외하고는 모두 C 언어를 이용하였다.

표 1. 기능 블록의 종류

블록 이름	기능
ANALOG INPUT/FILTER	아나로그 신호를 디지털로 변환 신호의 filtering
ANALOG OUTPUT	디지털 신호를 아나로그로 변환
DIGITAL IN/OUT	디지털 신호의 입출력
PID CONTROL	PID 제어, Smith-predictor등의 기능 선택
ANALOG COMPUTATION	아나로그 신호들의 계산
DIGITAL COMPUTAION	디지털 신호들의 논리 연산
SET POINT GENERATION	설정치를 프로그램
LIMIT CHECK/ANN	상/하한치 비교및 경고
SEQUENCE CONTROL	Batch 작업등의 시작과 종료 시간 설정
CONTROL STATE DISPLAY	제어기들의 상황을 bar graph로 나타내며 on-line tuning
GRAPHICAL PROCESS DISPLAY/DESIGN	공정도를 저장 혹은 수정하고 제어 변수 조정
SIG VARIABLE ASSIGNMENT	신호들의 실제값을 지정하고 LAN을 통해 보낼 신호를 선택
TIME CORRECTION	시간 보정

### 3. Hardware system의 구성

본 소프트웨어를 이용하여 수행할 수 있는 간단한 제어계의 구성을 그림 5.에 보였다. 그림에서 상위 컴퓨터는 host computer로서 주로 보고 및 graphic 기능을 담당하고 중간의 두 컴퓨터는 주 컴퓨터와 보조 컴퓨터로 똑같은 일을 수행하나 보조 컴퓨터는 고장시를 대비하기 위한 back-up용으로 이용한다. 여기서 WDT는 watch dog timer로서 두 컴퓨터 상호 간의 고장 유무를 판별하는 기능을 갖는다. 정상적인 경우에는 I/O interface를 통하여 신호들이 두 컴퓨터로 공급되고 계산된 출력은 주 컴퓨터에서 나오는 것만 긍정으로 연결되나, 만약 WDT에 의해 이상이 발견되면 정상인 컴퓨터에 의해서만 신호의 입출력이 이루어진다. 또한 모든 컴퓨터는 LAN으로 연결되어 있어 서로 자료를 교환할 수 있어 본산 제어계의 구성이 가능하다.

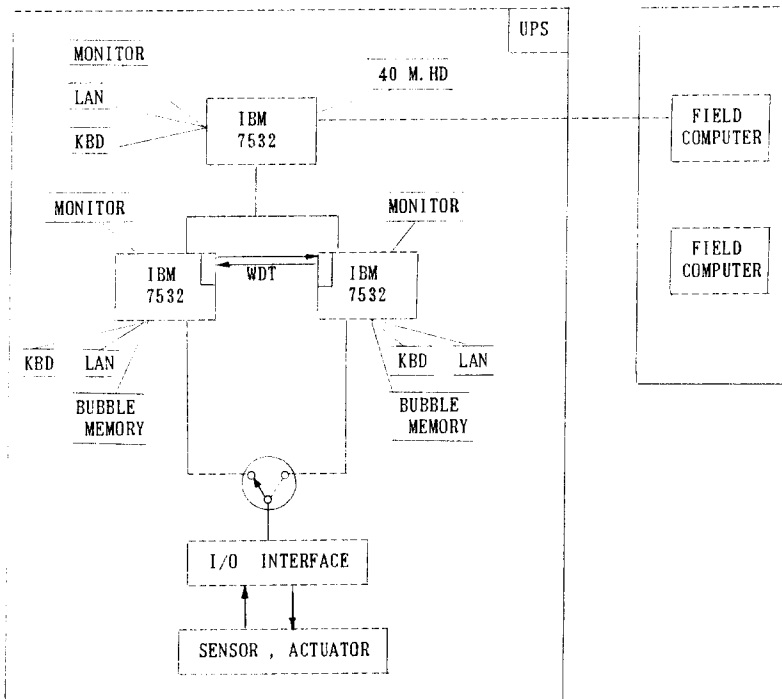


그림 5. Hardware system의 구성도