

'96 춘계학술발표회 논문집
한국원자력학회

자동기동시스템과 시험검증설비간의 통신프로그램 개발

김 정수, 정 철환, 함 창식

한국원자력연구소

정 일영

충남대학교

요 약

본 논문은 저온정지에서 2%까지 원전 자동기동시스템에 필요한 발전소 데이터를 시험검증설비로부터 얻기 위해 공유메모리와 TCP/IP를 사용하여 통신프로그램을 개발하였다. 자동기동시스템은 foxboro에서 제공하는 API(Application Program Interface)를 이용하여 데이터베이스에서 제어기에 필요한 데이터를 공유메모리에 올려놓고, 통신프로그램이 읽고 쓸 수 있도록 했으며, 시험검증설비에서는 HP Workstation에서 사용되는 내부 프로세스 통신방법을 이용하여 시험검증설비에서 나온 데이터를 공유메모리에 넣을 수 있도록 설계하였다.

1. 서론

일반적으로 workstation과 workstation 간에 통신은 개방형구조인 TCP/IP(Transmission Control Protocol/Internet Protocol)프로토콜[1]을 이용하여 통신한다. TCP/IP 통신이란 소켓 인터페이스를 이용하여 통신하는 것을 말한다. 즉, OSI(Open Systems Interconnections)의 7개 계층(계층1은 physical layer, 계층2는 data link layer, 계층 3은 network layer, 계층 4는 transport layer, 계층 5는 session layer, 계층 6은 presentation layer, 계층 7은 application layer)중 TCP/IP가 사용하는 3번째 계층인 network 계층과 4번째 계층인 transport 계층을 제외한 나머지 상위계층에 소켓인터페이스를 이용하여 통신프로그램을 설계한다[2].

시험검증설비는 원자력발전소를 모사할 수 있는 코드가 내장되어있어, 자동기동시스템과 같이 개발될 시스템에 연결하여 발전소의 데이터를 제공할 뿐만 아니라 이 시스템이 발전소에 설치할 경우 제대로 성능을 발휘하는 지를 검사하기 위한 설비이다.

그리고 자동기동시스템은 기동 운전모드(저온정지에서 고온정지까지의 운전모드, 고온정지에서 고온대기까지의 운전모드, 2차계통 가열 및 기동모드, 고온대기에서 2%까지의 운전모드)를 자동화하는 시스템이다. 이들 기동 운전모드들은 현재 발전소의 경우 자동화가 이루어져 있지 않아 운전원이 수동으로 운전하고 있다. 본 논문에서는 각 운전모드에 필요한 데이터를 시험검증설비로부터 얻고, 제어기로부터 시험검증설비에게 데이터를 줄 수 있는 통신 프로그램을 개발하였다. 이 프로그램에서는 각시스템에서 필요한 데이터를 공유메모리에 넣어주기 위해, 자동기동시스템인 foxboro는 제어기에서 생성된 데이터 또는 제어기가 필요로 하는 데이터를 공유메모리에 넣어주는 작업이 선행되어야한다. 이를 위해 foxboro에서 제공하는 API[4] 소프트웨어를 이용하였고, 시험검증설비는 내부 프로세스통신[3]에서 사용되는 함수를 이용하여, 시험검증설비에서 생성된 데이터를 공유메모리에 넣었다. 또한 각 시스템의 공유메모리에 넣어져있는 데이터에 접근하기 위해서도 foxboro의 API 및 HP의 내부 프로세스에서 사용되는 함수를 이용하여 구성하였다. 이 통신 프로그램은 송/수신을 위해 서버시스템과 클라이언트시스템에 각각 두 개의 프로그램으로 설계하였다. 그림 1은 시험검증설비와 자동기동시스템간에 데이터 흐름을 보여주는 구성도이다.

2. TCP/IP를 이용한 통신프로그래밍[1-2]

일반적으로 데이터전송시에는 연결설정, 데이터 전송 및 연결 끊기의 세과정이 필요하다. 이러한 데이터전송 방법에는 connection-oriented 프로토콜과 connectionless 프로토콜이 사용된다. Connection-oriented 프로토콜은 전화처럼 연결이 확보된 상태에서 이루어지므로 데이터를 전송하기에 앞서 두 응용계층을 연결하는 과정이 필요하다. Connection-oriented 프로토콜을 이용한 TCP/IP의 프로그래밍 구조는 그림 2와 같다. 여기서 프로토콜에 대한 세부함수들은 다음과 같다. ① socket 함수: 서버나 클라이언트 모두가 사용하며, 이 함수의 역할은 통신의 범위(internet domain or unix domain), 통신의 형태(tcp or udp) 등을 결정한다. ② bind 함수: 서버쪽에서만 사용되는 함수로서 소켓의 주소 및 그 크기 등을 결정한다. 이 함수의 호출결과로는 서버주소와 서버포트에 관한 정보를 알 수 있다. ③ listen 함수: 서버쪽에서만 사용되는 함수로서 클라이언트 쪽에서 connect() 함수가 올 때까지 기다리는 역할을 한다. 즉, 클라이언트로부터 연결설정을 기다리겠다고 클라이언트에게 알리는 역할이다. ④ accept 함수: 서버쪽에서만 사용되는 함수로서 connect()에서 넘어온 클라이언트의 주소 및 그 크기를 알 수 있으며, 이 함수의 호출결과로는 소켓 descriptor가 반환되고 server는 다음 클라이언트의 연결을 요청할 때까지 기다린다. ⑤ connect 함수: client 쪽에서만 사용되는 함수로서 서버쪽 listen()함수와 연결되어 있으며, 함수의 호출결과로는 서버쪽 주소와 크기를 가진다.

3. 자동기동시스템과 시험검증설비간에 통신프로그램 개발

3.1 시험검증설비 (server) 통신프로그램

3.1.1 시험검증설비에서 공유메모리 접근방법

통신을 위해서는 먼저 공유메모리를 접근하여 공유메모리에 있는 데이터를 가져오는 기능이 필요하다. 시험검증설비에서 사용되는 HP workstation에서, 공유메모리를 사용하기 위하여 HP에서 제공하는 내부 프로세스통신을 이용하였다. 이 내부 프로세스통신은 Message Queue, Semaphores 및 Shared Memory로 구성된다. 두 프로세스가 정보를 교환하기 위해서는 kernel을 이용하는 방법과 공유메모리를 이용하는 방법이 있다. 일반적으로 사용되는 kernel을 이용하는 방법은 데이터간의 통신시간이 많이 걸리지만, 공유기억장소를 이용하면 kernel을 통하지 않고서도 정보를 직접 주고받을 수 있다. 이렇게 하면 workstation 내부에서의 통신로드를 줄일 수 있다. 공유메모리와 관련된 내부 통신함수로는 shmget(), shmat(), shmdt(), shmctl()등을 사용하였다. 이 공유메모리를 사용한 이유는 시험검증설비의 경우 시험검증설비에서 운영되는 데이터를 변화시키지 않고 공유메모리에서 필요한 데이터만을 넣어주므로해서, 시험검증설비의 데이터와 자동기동에 이용되는 데이터는 분리할 수 있고, 통신프로그램과 시험검증설비의 모사 프로그램을 분리 운영할 수 있는 장점이 있다. 그림 3은 workstation 내부에서 kernel을 이용한 방법과 공유메모리를 이용한 방법을 비교한 그림이다.

3.1.2 서버 통신 프로그램 구성

프로그램의 중심역할을 하는 master server 밑에 수신담당과 송신담당 프로세스를 분리하기 위하여 두개의 slave server를 두었고, 이들은 master server로부터 프로세스 복제(fork())에 의해 생성된다. Master server가 프로세스 복제를 하는 시점은 client로부터 연결요청이 온 이후에 수행된다. 일반적인 TCP/IP 프로토콜에서 server-client의 통신은 서버의 요청에 의해 데이터를 보내는 것인데 이러한 방식을 본 시스템에 적용하면 양쪽 시스템의 scan rate가 상이하야(서버의 scan rate 0.2초, 클라이언트의 scan rate 0.5초) 문제가 발생한다. 즉, 서버의 요청에 의해 클라이언트는 데이터가 변하지 않았는데도 데이터를 보내는 문제가 발생하게 된다. 이를 방지하기 위해 클라이언트가 scan rate에 의해 변하는 시점에 서버의 요청 없이도 무조건 보내도록 설계하였다. 이렇게 함으로써 서버는 클라이언트가 변한 시점에 바로 데이터를 받아볼 수 있는 장점이 있다.

그림 4는 server 프로그램의 전체적인 구성을 나타내고 있다. Slave server1은 송신을 담당하고 slave server 2는 수신을 담당하는 프로세스이다. 통신프로그램은 다음과 같은 절차로 수행된다. 즉 server는 연결시작, 프로세스복제에 의한 fork() 수행, 연결인식, 데이터수신/저장, 데이터취득/송신 등으로 수행된다. 연결시작은 클라이언트에서 서버로의 연결 요구신호로부터 시작된다. 서버는 이러한 요구신호를 취득하고, 자기자신의 프로세스를 복제한다. 서버에 의한 fork()수행은 프로세스 복제가 이루어지면 두 개의 프로세스가 생성되는데, 하나는 원래의 프로세스이고, 다른 하나는 새로이 생성된 프로세스이다. 연결인식은 앞의 fork()에 의해 생성된 두 개의 server는 각각 다른 port를 연다. 따라서 slave server들은 클라이언트의 연결요구시에 생성된 포트를 통해 데이터를 주고 받으며, 클라이언트는 이 포트에 대한 정보를 알아야한다. 데이터 수신/저장, 취득/송신을 위해 slave server들은 시험검증설비의 공유메모리와 공유하고있으며, 일정주기로 내부 프로세스 통신을 이용하여 필요한 데이터를 취득하여 클라이언트에게 송/수신한다.

3.2 자동기동시스템(client) 통신프로그램

3.2.1 제어기에서 공유메모리 접근방법

제어기에서 공유메모리 접근방법은 foxboro에서 제공하는 API 소프트웨어 도구를 이용하여 공유메모리에 데이터를 쓰거나 읽어오도록했다. 이 API 소프트웨어는 제어기에서 생성된 데이터를 공유 메모리로 넣어주는 역할을 담당한다. 또한 API는 제어기의 데이터와 공유메모리에 있는 데이터를 연결해주는 역할을 한다. Foxboro의 경우 제어기에서 생성되는 모든 데이터는 OM(Object Manager)[4]라는 데이터베이스가 관리하고 있다. 이 데이터베이스로부터 원하는 데이터를 공유 메모리에 넣어주는 역할이 바로 API 소프트웨어이다. API 소프트웨어를 사용하는 절차는 다음과 같다. 먼저, 원하는 데이터의 변수이름(예를 들면 steam generator level 값: IN_DATA:Steam_Gen.meas)을 데이터 화일로 만든다. 다음으로 API 환경을 구축한다. 즉, 공유메모리를 할당하고, 앞의 데이터를 OM으로부터 공유메모리에 넣어주기위해 데이터 화일을 연다. 그리고 API 절차를 따라 데이터를 열면 OM에 있는 데이터들이 공유 메모리에 놓이지며 데이터 간의 통신할 준비가 완료된다. 이러한 방법으로 제어기의 공유메모리를 접근할 수 있다. 통신프로그램과의 연결은 foxboro에서 제공되는 read, write함수(readval(), wrtval())를 이용하여 프로그램하고 library와 링크 하여 compiler하면 제어기의 공유메모리 접근 및 통신 준비작업은 완료된다.

3.2.2 클라이언트 통신 프로그램 구성

클라이언트의 구성은 그림 5와 같이 두개의 slave server들에 대응하는 두개의 클라이언트로 구성하였다. Slave server 1에 대응하는 client는 송신을 담당하고, slave server 2에 대응되는 client는 수신을 담당한다. 이 그림에서 클라이언트는 server처럼 프로세스복제를 하는 것이 아니라 두 개의 독립된 프로세스로 각각 서버와 연결되어 있고 각 클라이언트는 제어기의 공유메모리를 공유하고 있다. 이것은 client 시스템의 특성상 서버에서와 같이 하나의 통신프로그램에서 처리하지 못하고 송/수신프로그램이 따로 설계해야 하기 때문이다. 클라이언트 프로그램은 다음과 같은 절차로 수행된다. 즉 클라이언트는 연결시작, 데이터수신/저장, 데이터취득/송신 등으로 수행된다. 연결시작에서 클라이언트는 데이터 송/수신을 위해 서버에게 연결시작신호를 보낸다. 연결이 이루어지면 클라이언트는 자신이 어떤 종류(송신/수신)의 통신을 원하는지를 알려주는 신호를 server에게 알려준다. server는 이 신호를 확인하여 어떤 클라이언트가 연결되었는지를 알게된다. 데이터 수신/저장, 취득/송신을 위해 클라이언트들은 제어기의 공유메모리와 공유하고있으며, 일정주기로 필요한 데이터를 취득하여 서버에게 송/수신한다.

4. 실험

자동기동시스템의 제어기는 저온정지에서 2%까지 자동화를 위해 발전소에서 사용될 주요변수를 정상운전절차서로부터 분석한 결과 약 160개의 변수가 발전소로부터 제어기에 송신되어야하고, 제어기로부터 발전소에 송신해야할 변수는 약 40 개 정도였다. 이러한 데이터를 얻기 위해

먼저 시험검증설비에서 사용되는 모든 데이터 중에서 자동기동시스템에 필요한 데이터를 선정하여 HP의 공유메모리에 넣었다.

그리고 제어기는 송/수신 하고자하는 변수목록을 만들고 API 소프트웨어를 이용하여 제어기의 공유메모리에 변수목록을 넣어준 후 본 논문에서 개발한 통신프로그램을 이용하여 HP의 공유메모리에 올려져있는 데이터를 취득하여 제어기로 송신하였다. 먼저 실험을 위해 40개의 데이터를 선정하였고, 이를 analog 변수와 digital변수로 분리하여 제어기로 데이터를 송신하였고, 시험검증설비가 수신하는 데이터도 역시 분리하여 수신하였다. 이러한 값들을 통신프로그램에서는 HP 공유메모리에 있는 변수값을 하나씩 차례대로 읽어 제어기에 보냈다. 제어기는 차례대로은 데이터를 제어기의 변수목록에 맞게 넣어준다. 한편 제어기에서 시험검증설비로 보내는 변수값은 제어기에서 변환값(예를 들면 제어기의 setpoint 값 또는 output 값)을 통신프로그램에서 차례대로 읽어 시험검증설비로 송신한다. 그림 6은 시험검증설비가 실행되는 상태를 나타내고 있다. 이 그림에서 약 160개 이상의 데이터가 시험검증설비에서 실행되고 있지만 통신프로그램에 의해서 제어기에서 필요한 160개의 데이터만 보내도록 하였다. 그림 7은 자동기동시스템에서 개발될 화면중 일부화면인 2차측 화면을 나타내고 있으며, 이 그림은 시험검증설비에서 보내온 데이터와 연결되어 발전소가 100 %일 때의 값을 보여주고 있다. 이 그림에서 디지털 값은 펄프의 경우 on 상태이면 빨간 색으로 표시하고, off 상태이면 파란색을 표시하고 있으며, 아날로그 값인 경우 예를들면 증기발생기 수위값을 숫자로 표시하고 있다. 그림 8은 제어기에서 API 소프트웨어를 이용하여 제어기에 원하는 데이터를 공유메모리에 올리는 것을 보여주는 것으로, 원하는 데이터 값이 제어기와 연결되는 상태 및 그때 값 등을 표시하고 있다. 물론 이 값들은 foxboro의 OM database와 연결되어 있어 이 값이 변하면 그림 7에서 해당되는 표시 값들이 즉시 변화된다. 이 그림에서 제어기와 연결상태는 status에 표시되며, 연결이 되지 않은 경우 no response, 연결된 경우는 Act_on_으로 나타나 연결 전에 제어기와 공유메모리 데이터가 맞게 연결되었는지를 직접 확인하여 제어기와 공유메모리사이의 데이터 연결상태를 바로 잡을 수 있다. 그리고 error나 status에 의한 bit표시는(예를 들면 0623x)등은 이 데이터 값이 정수 혹은 실수 값인지를 알려준다. 본 논문에서 개발한 통신프로그램으로 약 40개의 데이터를 실험한 결과 정수/실수값이 구분되어 시험검증설비와 제어기사이 데이터교환이 설정된 시간내에 데이터가 송/수신됨으로써 통신시간에 문제가 없음을 확인하였다. 이 실험으로 부터 앞으로 개발될 자동기동 시스템에 본 논문에서 개발한 통신프로그램을 사용하면 연결이 용이하고 데이터검사가 편리하여 필요한 데이터를 송/수신하는데에 문제가 없으리라 예상된다.

5. 결론

자동기동시스템에서 제어기를 설계하기 위해서는 각모드에 맞는 발전소의 데이터가 필요하며 이를 시험검증설비로부터 얻어야한다. 본 논문에서는 데이터베이스에 접근이 용이하고 통신시간을 줄이기 위해 공유메모리를 이용하였고, TCP/IP(Transmission Control Protocol/Internet Protocol) 프로토콜을 이용한 시험검증설비와 자동기동시스템간에 통신프로그램을 개발하였다. 각 시스템의 공유메모리에 데이터를 넣어주는 방법은 제어기의 경우 foxboro에서 제공하는 API 소프트웨어를 이용하여 제어기에서 필요한 데이터를 공유메모리에 넣어 데이터를 읽고 쓸 수 있게 했으며, 시험검증설비에서는 HP workstation에서 사용되는 내부 프로세스통신을 이용하여 시험검증설비에서 나온 데이터중 제어기가 필요로 하는 데이터를 공유메모리에 넣을 수 있도록 구성하였다. 그러므로 이를 타 시스템(예를 들면 이기종의 DCS와 W/S간에)에 적용할 경우 본 논문에서 개발한 공유메모리 접근방법만 변경하면 쉽게 적용할 수있을것으로 예상된다. 또한 앞으로 개발될 자동기동 시스템의 경우에도 본 논문에서 개발한 통신프로그램을 사용하면 연결이 용이하여 필요한 데이터를 송/수신하는데에 문제가 없으리라 예상된다.

참 고 문 헌

1. 유상렬, TCP/IP인터넷(이론, 프로그래밍, 사용법), 성안당, 1994.
2. Stevens, UNIX Networking programming, Prentice-Hall, 1991.
3. HP-UX Manual, HP-UX Reference Volume 2, Hewlett-Packard, 1991.
4. Foxboro Manual, Application Interface Software Programmer's Guide for 50 Series System Foxboro, 1993.

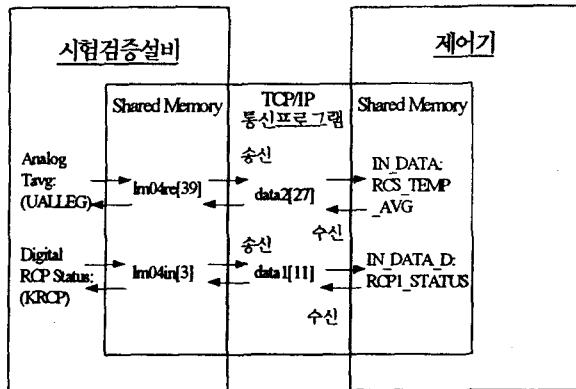


그림 1. 시험검증설비와 자동기동시스템간의 데이터 흐름도

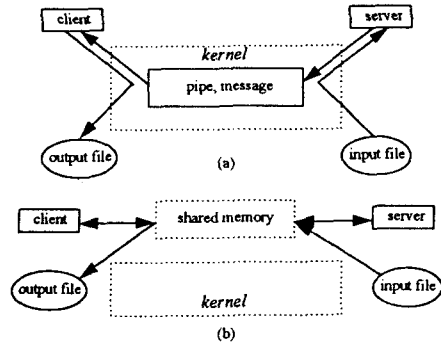


그림 3. (a) kernel 을 이용한 데이터교환과 (b) 공유기억장소를 이용한 데이터 교환

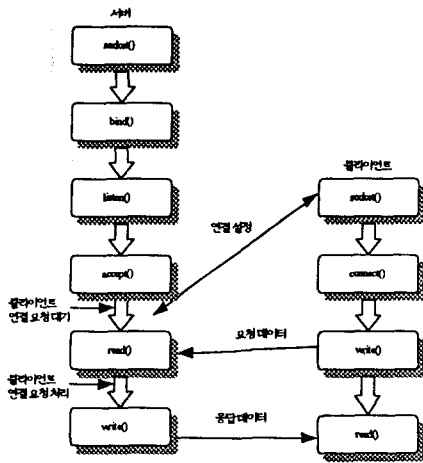


그림 2. 일반적인 TCP/IP 프로그래밍 구조

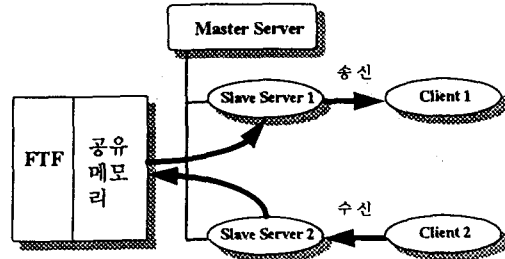


그림 4. Server 프로그램의 구성도

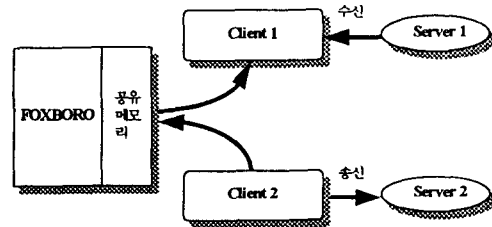


그림 5. Client 프로그램의 구성도

KAERI I&C Functional Test Facility (제측제어 시험설비) Variables Display

```

aRUN
6
9 tVariable      Value      Comment      No Variable  Value      Unit
1 PRZ            1.318732e+07 PRZ pressure (Mpa) 31 QNET      -2.052683e+04 T
2 ZPRZNO        4.471579e-01 PRZ nor. level(%) 32 QLOAD     -2.067153e-05 N
3 URZ           3.447764e+02 PRZ temperature(C) 33 KLDW      1.000000e+00 L
4 UAVLECH       4.502768e+02 Average temp. (deg. 34 OLDEY      0.000000e+00 T
5 UDHCO         3.163418e+02 Delta temp.        35 OLDET      0.000000e+00 T
6 USCRCP        -1.901863e+03 RCP-R/C flow      36 QLRATE    -1.327862e-10 T
7 PSC           8.564952e+03 S/G press. (Mpa)  37 OLSET      0.000000e+00 T
8 ZSCHOR        4.867343e-01 S/G Nor. level(%) 38 OLZER      0.000000e+00 T
9 WFLW          1.819332e+01 Fw line flow(kg/sec 39 CBORW     4.800000e-04 B
10 VSTN          1.601212e-01 Steam line flow    40 ZREAC     6.471757e+00 V
11 UTIN          2.822453e+02 Steam brd temp.   41 MLETWD    5.188974e+00 W
12 VTIN          0.000000e+00 S/G Steam flow     42 MCRWD     8.853121e+00 C
13 PTIN          2.645662e+02 S/G Steam press    43 WNETCH    9.786662e+00 M
14 BTIN          2.789091e+06 Steam onthlpy      44 MSPRAY    3.327794e+00 S
15 PRPTIN        9.800000e+04 RP pressure        45 ERUS      3.000000e+00 H
16 BEVTON        1.000000e+00 RS isolation v/v   46 ERUST     0.000000e+00 H
17 QPROMEL       7.749113e-10 Relative power     47 PFVP      0.000000e+00 F
18 QPROMOR       4.784305e-10 Normal QPROMEL    48 BFV478    9.933113e-01 F
19 XPTIN         -1.100000e+01 Intermediate       49 BFV479    7.365590e-02
                -5.549863e-02                2.898252e+00
                -3.936842e-03                1.282525e+01
                4.854200e+02                0.000000e+00
                2.000000e-03                0.000000e+00
                0.000000e+00                0.000000e+00
                1.230416e+05                0.000000e+00
                3.513888e+01                0.000000e+00
                1.000000e-06                3.736195e+01
                1.000000e+07                5.555146e+01
                1.000000e+00                1.000000e+00
                -3.612533e-03                -1.000000e+00
                6.685000e+03

```

그림 6. 시험검증설비 수행상태

ASICS SECONDARY SYSTEM

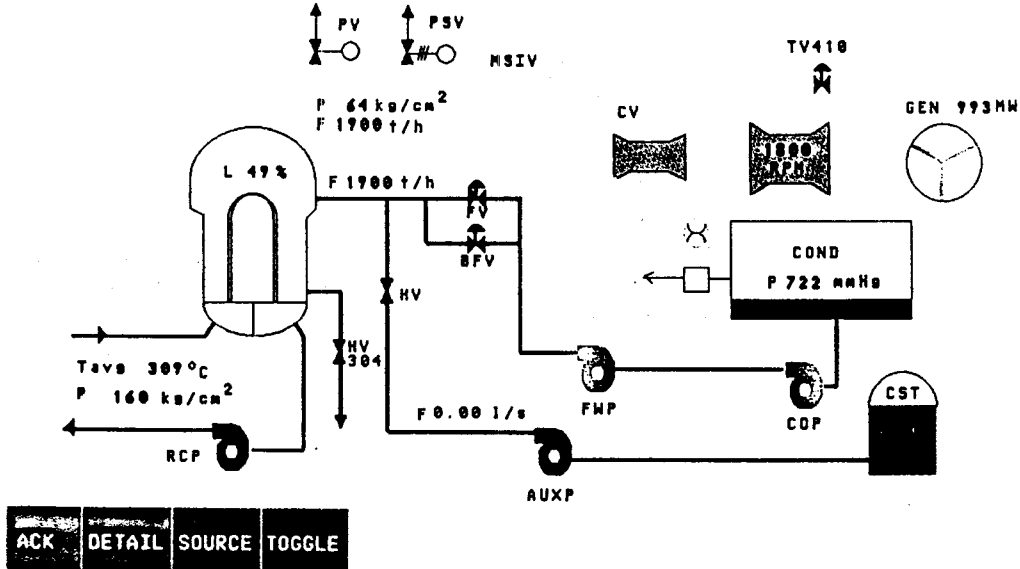


그림 7. 자동기동시스템중 2차측화면

```

Central task delays between states are:
State 1 = 0
State 2 = 1000
State 3 = 0
State 4 = 0
State 5 = 1000
State 6 = 1000
State 7 = 1000
State 8 = 0
State 9 = 0
State 10 = 0
State 11 = 0
State 12 = 0

The Open/close CTL/C message is displayed.
Logging to terminal is enabled.
Packing of read lists is turned off.
Packing of write lists is turned off.
Packing of open lists is turned off.
Entry Pt #: rslid=APC101, lat_bug=APC101, entry_pt=/. rw_access=R/W(1)
Remounting of remote file system is turned off.
Tue Apr 9 14:01:38 1986 alslog (rombox AIS package).
CTR Tue Apr 9 14:01:34 1986 f131 ctnit: central task initialization successful.
CTRL C disabling.
Open done, start bread to initialize taval and lastat
retarr = 26
dset = 1
set name = my_test1.dat

entry name          error          value status
-----
1 IN_DATA:PV_RANGE_DET.MEAS 0623x          26.08 0623x Act_OnSec_OkFlt
2 IN_DATA:RCS_TEMP_AVG.MEAS 0623x          26.11 0623x Act_OnSec_OkFlt
3 IN_DATA:RCS_PRESS.MEAS     0623x          26.10 0623x Act_OnSec_OkFlt
4 IN_DATA:PRZ_LEVEL.MEAS     0623x          26.07 0623x Act_OnSec_OkFlt
5 IN_DATA:PRZ_VV_POS.MEAS    0623x          25.94 0623x Act_OnSec_OkFlt
6 IN_DATA:RHR_BP_FLW.MEAS    0623x          0.00 0623x Act_OnSec_OkFlt
7 IN_DATA:FK_L1_FLW.MEAS     0623x          26.15 0623x Act_OnSec_OkFlt
8 IN_DATA:RCS_PRESS_W.MEAS   0623x          26.10 0623x Act_OnSec_OkFlt
9 IN_DATA:SG1_LEVEL_MO.MEAS  0623x          26.07 0623x Act_OnSec_OkFlt
10 IN_DATA:SG1_LEVEL_MA.MEAS 0623x          26.10 0623x Act_OnSec_OkFlt
11 IN_DATA:BOI_FLW_DAT.MEAS   0623x          26.34 0623x Act_OnSec_OkFlt
12 IN_DATA:SG1_PORV_101.MEAS 0623x          0.00 0623x Act_OnSec_OkFlt
13 IN_DATA:AUX_FW_313.MEAS    0623x          25.94 0623x Act_OnSec_OkFlt
14 IN_DATA:TB_LOAD.MEAS      0623x          26.14 0623x Act_OnSec_OkFlt
15 IN_DATA:HV142_VV_POS.MEAS 0623x          0.00 0623x Act_OnSec_OkFlt
16 IN_DATA:HV603_VV_POS.MEAS 0623x          26.09 0623x Act_OnSec_OkFlt
17 IN_DATA:BHV108_VV_PO.MEAS  0623x          26.09 0623x Act_OnSec_OkFlt
18 IN_DATA:HST_L_PRES.MEAS    0623x          26.08 0623x Act_OnSec_OkFlt
19 IN_DATA:TB_SPEED.MEAS     0623x          26.15 0623x Act_OnSec_OkFlt
20 IN_DATA:VCC_PRES.MEAS     0623x          26.13 0623x Act_OnSec_OkFlt
21 IN_DATA:CHD_LEVEL.MEAS    0623x          26.03 0623x Act_OnSec_OkFlt
22 IN_DATA:TB8_VV_POS.MEAS   0623x          0.00 0623x Act_OnSec_OkFlt
23 IN_DATA:FW_VV_POS.MEAS    0623x          25.94 0623x Act_OnSec_OkFlt
24 IN_DATA:FK_VV_POS.MEAS    0623x          26.09 0623x Act_OnSec_OkFlt
25 IN_DATA:HPT_CVV_POS.MEAS  0623x          26.08 0623x Act_OnSec_OkFlt
26 IN_DATA:AUX_FW_TFLW.MEAS  0623x          0.00 0623x Act_OnSec_OkFlt
27 IN_DATA:LD_NET_FLW.MEAS   0623x          26.05 0623x Act_OnSec_OkFlt
28 IN_DATA:CC_MO_FLW.MEAS    0623x          26.01 0623x Act_OnSec_OkFlt
29 IN_DATA:HST_L1_FLW.MEAS   0623x          0.00 0623x Act_OnSec_OkFlt
30 IN_DATA:SR_RANGE_DET.MEAS 0623x          0.00 0623x Act_OnSec_OkFlt
31 IN_DATA:IN_RANGE_DET.MEAS 0623x          0.00 0623x Act_OnSec_OkFlt
32 IN_DATA:REACTIVITY.MEAS   0623x          0.00 0623x Act_OnSec_OkFlt
33 IN_DATA:HOT_LEG1_TEN.MEAS 0623x          0.00 0623x Act_OnSec_OkFlt
34 IN_DATA:HOT_LEG2_TEN.MEAS 0623x          0.00 0623x Act_OnSec_OkFlt
    
```

그림 8. API를 이용하여 공유메모리에 올린경우