

HTTP를 이용한 원거리 가스누출 감시 시스템에 관한 연구

A Study of Gas Leakage Monitoring System on a Long Distance Using the Hypertext Transfer Protocol

이 광 희*

Lee, Kwang Hee

안 형 일*

Ahn, Hyung Il

김 응 식**

Kim, Eung Sik

Abstract

In this paper, I present an architecture and techniques to monitor and identify the hazards of plants anywhere in the whole country using the HTTP(Hypertext Transfer Protocol) based on RFC1945 and PLC(Programmable Logic Controller) protocol. I constructed the upward network and downward network for intercommunication between the PLC and computer around the internet. I also constructed WWW(World-Wide Web) server in the personal computer

The result of this research constructed monitoring system to monitor and identify the hazards through WWW browser on the internet.

Key Words : Monitoring System, Gas Leakage, HTTP, Web

국문요약

본 논문은 RFC1945의 바탕을 둔 HTTP와 PLC전용프로토콜을 사용하여 시간과 장소에 구애받지 않고 위험상황을 감시할 수 있는 원거리 가스누출 감시 시스템에 관해 연구하였다. 이를 위하여 인터넷상에 컴퓨터와 PLC의 접속을 위해 상위네트워크와 하위네트워크를 구성하여 인터넷상의 Web을 통해 위험요소를 감시할 수 있는 원거리 가스누출 감시 시스템을 구축하였다.

* LG소프트 시스템사업팀 연구원
* 호서대학교 공과대학 안전공학부
** 호서대학교 공과대학 안전공학부 교수

1. 서 론

근래의 화재 사고는 인명 및 재산피해가 극대화될 가능성이 높으므로 시간 및 장소 등 환경에 따라 미묘하게 달라지는 재해 발생에 효과적으로 대처하기 위한 진단 및 판단, 선택 능력을 가진 CIM(Computer Integrated Manufacturing)을 설치하여 운용하여야 한다. 이러한 시스템은 위급상황시 비전문가가 판단하기 어려웠던 상황을 실시간에 처리할 수 있으며, 인간을 전적으로 배제할 수 없었던 수정, 보완 등의 작업도 모니터링시스템을 통해 운용될 수 있다. 그러나 전국 또는 국가간에 흩어져 있는 대규모의 작업장을 대상으로, 각 작업장의 화재, 폭발요소를 감시하기 위해서는 기존에 시판되고 있는 방재 시스템의 RS232-C통신으로는 통신거리의 한계를 극복할 수 없어 원거리에서의 감시 및 제어가 불가능하다. 때문에 기존의 설비를 그대로 유지하면서 시간과 공간에 구애받지 않고 전역적인 감시 및 제어를 위한 중앙 집중적 가스누출 감시 시스템에 관해 연구가 필요하다.

본 논문은 기존의 감시 시스템을 원거리에서 중앙집중적으로 감시할 수 있는 통신환경에 관하여 연구하였다. 이를 위해 자동화에서 필수적으로 사용하고 있는 PLC(Programmable Logic Controller)에 가스센서를 연결하여 센싱된 센서의 데이터를 RS232-C 통신으로 컴퓨터에 전송한 뒤, 인터넷(Internet)의 WWW(World Wide Web)의 프로토콜인 HTTP[1](Hypertext Transfer Protocol)을 이용해 인터넷상에서 감시 및 제어할 수 있게 하였다.

인터넷은 전세계에 걸쳐 연결되어진 거대한 네트워크로 인터넷에 컴퓨터를 연결하면 전세계에 연결된 컴퓨터와 통신을 할 수 있기 때문에 국한된 목적만을 위한 전용의 네트워크를 따로 구축하지 않아도 원거리 통신이 가능하여 경제적이고 효과적인 가스누출 감시 시스템의 구축이 기대된다.

2. 통신 시스템 설계 및 환경

2.1 전체 시스템 구성

방재 시스템은 화재나 폭발과 같은 위험 요소 발견시, 재해를 최대한 예방할 수 있도록 선(先)처리 후(後)관리할 수 있게 설계되어야 한다. 때문에 순차적 제어회로를 구성하여 위험요소 감지에 따른 적절한 조치가 이루어질 수 있게 설계해야 한다.

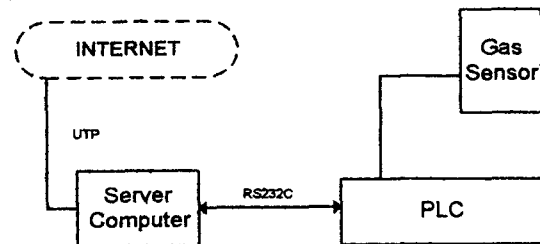


그림 1. 시스템 구성.

본 연구에서는 그림 1.과 같이 순차적인 제어를 위해 PLC를 사용하여 반도체식 가스센서를 PLC의 입/출력 유니트에 연결하여 가스누출을 감시할 수 있도록 구성했다. 그리고 설비의 이력관리와 통합적인 관리를 위해 컴퓨터와 PLC 간에 데이터링크했으며, 원거리에서 가스누출 상황을 감시하고 PLC를 제어하고자 인터넷에 연결된 컴퓨터에 웹서버(Web server)를 구축했다.

2.2 시스템 구현 환경

자동화공정에서 필수적으로 사용하고 있는 PLC는 기능별로 여러 종류의 유니트(unit)가 있다. 그 종류를 살펴보면, 제어하고자 하는 각종 기기의 입/출력을 담당하는 입력유니트와 출력유니트, 아날로그 신호나 디지털 신호를 변환시켜주는 A/D유니트, D/A유니트가 있으며 외부 기기와 통신하기 위한 통신유니트 등이 있다. 연구에서 사용된 PLC는 LG산전에서 시판하고 있는 Gold Sec-G5 PLC를 사용하여 PLC 입력유니트인 GX41에 4개의 가스센서를 부착하여 가스누출 여부를 감지할 수 있게 했다.

표 1. 서버 컴퓨터의 구성.

server specification	PC
Operating System	MS - Windows95
IP	134.75.122.244
Daemon	HTTP1.0
HTML	HTML2.0
Cable	UTP
Ethernet	NE2000 Compatible
PLC	GoldSec-G5

사용한 가스센서는 반도체식 가스센서인 FIGARO사의 TGS813 가연성 가스센서를 사용했다. 이 센서는 회로 전압(Circuit voltage) AC, DC 최대 24V까지 가능하고 히터 전압(Heater voltage) AC/DC 5V를 사용할 수 있다. 감지 대상으로는 프로판(propane)과 메탄(methane)을 500~ 10,000ppm을 감지할 수 있다. 그리고 한글 Windows95를 운영체제로 하는 486DX4-100Mhz IBM호환용 컴퓨터를 사용하여, 컴퓨터의 RS232-C포트와 PLC통신유닛(G5C24)의 RS232-C포트간에 서로 연결했다. 케이블은 RS232-C표준을 준수하여 배선하였으며 거리는 8m로 하였다.

인터넷의 접속을 위하여 컴퓨터에 NE2000 호환 이더넷카드(Ethernet Card)를 장착하여 KREONET(연구전산망)의 T1급 전용회선과 연결하였다. 사용한 IP주소는 134.75.122.244를 부여하였고, 이때 사용한 케이블은 Twisted Pair를 사용하였다.

3. 메시지 교환과 데이터 프레임(Data Frame)구조

개인용 컴퓨터로 통신을 하는 경우, 데이터 끝에 단지 종료부호만을 두어 통신하는 경우와 일정한 통신규약에 의해 실행하는 경우가 있다. 통신규약의 대표적인 것으로는 아래의 내용과 같다.

- ① 회선이 접속되었는지의 확인을 위한 회선 접속과 상대방의 확인.
- ② 데이터가 올바르게 전송되었는지를 확인.

③ 데이터의 송신이 종료되었는지의 확인 등이 있다.

이 같은 데이터 전송에 대한 제어 절차를 전송 제어(Transfer control)라고 한다. 이 전송제어를 사용하여 컴퓨터와 외부기간에 메시지 교환이 이루어지는데 그림.2는 본 연구에서 사용한 메시지 교환에서 전송제어 캐릭터에 대한 것이다.

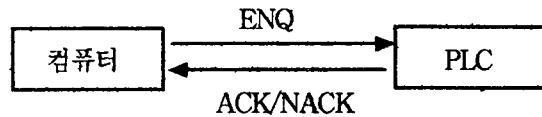


그림 2 데이터 전송제어.

ENQ(Enquiry)는 ASCII 코드 5번에 해당되는 것으로 전송제어시 상대 기기의 응답을 구하기 위해 사용되는 부호이다. ACK(Acknowledge)는 수신측에서 송신측으로 반송하는 부호인데 만약 접속이 정상적이면 ACK를 반송하고 비정상일 때는 NACK(Negative Acknowledge)를 반송하게 된다.

본 논문에서는 그림 2.와 같은 원리로서 PLC의 G5C24 통신 유닛과 컴퓨터간에 데이터통신이 이루어진다. 사용한 프로토콜은 PLC전용 프로토콜(protocol)을 사용하였으며, 이때 사용한 데이터 프레임구조는 PLC에서 데이터를 Read나 Write하기 위한 요청프레임(Request frame)과 PLC에서 컴퓨터로 전송한 응답프레임(Response frame)이 있다.

3.1 요청프레임(Request Frame)

그림 3.은 컴퓨터에서 PLC에 전송하기 위한 데이터 프레임의 구조이다. 이 구조의 "명령어"란에 어떠한 캐릭터를 기입하는냐에 따라 PLC 접점의 상태를 Read할 것인지 Write할 것인지 등을 결정하게 된다.

각각의 캐릭터는 ASCII코드로 구성되어 있고, 본 논문에서는 Read를 위한 요청프레임을 사용하여 가스센서의 상태를 읽어들인다.

E	국	명	Character	갯	E
N		령			O
Q	번	어		수	T

그림 3. READ/ WRITE^[3] 요청프레임 구조.

그림 3.에서 보는 것과 같이 ENQ를 시작으로, EOT까지의 캐릭터 순서로 데이터 프레임을 PLC에 전송한다.

각각의 전송제어 캐릭터는 다음과 같다.

① ENQ는 PLC의 응답을 구하기 위한 캐릭터이다.

② 국번은 PLC의 국번으로, 여러 대의 PLC가 서로 연결(Link)되었을 경우 PLC간의 구분을 위한 것이다. 논문에서 사용한 국번은 0번으로 설정하였다.

③ 명령어는 PLC의 데이터를 Read할 것인지 Write할 것인지를 결정하는 것으로 Read를 위해 명령어 RB를 사용하였다. RB는 byte단위로 데이터를 읽어 들이겠다는 의미이다.

④ Character는 읽고자 하는 PLC접점의 어드레스(address)이다.

⑤ 갯수는 읽고자하는 데이터의 갯수이다.

⑥ EOT는 전송을 종료하고 송신권을 PLC에게 인도하겠다는 캐릭터다.

본 논문에서는 입력유닛 X0016~X0019접점에 4개의 가스센서를 연결하여 가스누출상태를 감시하였다. 감시를 위해 사용한 요청프레임은 그림 4.의 구조를 사용하였고 PLC의 제어를 위해 그림 5, 그림 6.의 요청프레임을 사용하였다.

그림 4.는 가스누출상태 감시에 X0016, X0017, X0018, X0019의 4개 접점상태를 Read하고자 사용된 13byte크기의 데이터 프레임이다.

E					E
N	00	BR	X0016	04	O
Q					T

그림 4. READ^[3] 요청프레임 구조.

E			E
N	00	RR	O
Q			T

그림 5. Remote-Run^[3] 요청프레임 구조.

E			E
N	00	RS	O
Q			T

그림 6. Remote-Stop^[3] 요청프레임 구조.

그림 5.는 PLC CPU의 Remote Run을 제어하기 위해 사용한 요청프레임이며 그림 6.은 PLC CPU의 Remote Stop을 제어하기 위해 사용한 요청프레임이다.

3.2 응답프레임 (Response Frame)

그림 4.의 요청프레임을 PLC에 전송하면 PLC는 그림 7.과 같은 구조를 가지고 가스센서의 상태를 컴퓨터에게 응답한다.

응답프레임의 "데이터"를 바탕으로 현재 가스센서의 상태를 점검하게 되며, 1회 송신에 1회 수신으로 한 사이클이 완성된다.

S	국	명	데이터	E
T		령		T
X	번	어		X

그림 7. READ^[3] 응답프레임 구조.

그림 7.의 응답프레임에 대한 내용은 아래와 같다.

① STX는 컴퓨터에 응답하기 위한 캐릭터이다.

② 국번은 PLC의 국번이다.

③ 명령어는 요청프레임에 의해 수행된 명령어로, 요청프레임에서 명령한 명령과 동일하다.

④ 데이터는 PLC의 접점상태이다.

⑤ ETX는 프레임 끝을 나타낸다.

S	국		E
T		RR	T
X	번		X

그림 8. Remote-Stop^[3] 응답프레임 구조.

S	국		E
T		RS	T
X	번		X

그림 9. Remote-Stop^[3] 응답프레임 구조.

그림 8.은 PLC CPU의 Remote Run을 제어 하기 위해 사용된 응답프레임이며, 그림 9.는 PLC CPU의 Remote Stop을 제어하기 위해 사용된 응답프레임이다.

4. Web을 통한 감시 및 제어

대부분의 PLC는 이더넷 통신유니트를 가지고 있지 않고 TCP/IP를 지원하지 않고 있다. 때문에 TCP/IP를 사용하는 인터넷와는 하드웨어적으로 연결이 불가능하다. 본 연구에서는 이러한 문제를 소프트웨어적으로 해결하고자 그림 10.과 같이 컴퓨터-PLC간의 하위네트워크와 컴퓨터-Internet간의 상위네트워크를 구현하였다. 하위네트워크에서는 컴퓨터와 PLC간의 데이터전송을 위해, 3.1절과 3.2절에서 보인 것과 같이 PLC전용 프로토콜을 사용하여 데이터를 송수신하고 상위네트워크인 컴퓨터와 인터넷은 Windows95에서 내장된 TCP/IP로 통신한다. 이렇게 구성된 네트워크는 컴퓨터를 기점으로 상/하위네트워크간의 서로 다른 프로토콜을 연결시켜 준다.

본 논문에서는 PLC로 입력된 접점의 상태를 원거리에서 감시 및 제어를 위해 컴퓨터에 웹서버를 구축하여 서로 다른 프로토콜의 데이터전송을 그림 10.과 같이 구현했다.

웹(Web)은 HTTP^[1]라는 프로토콜을 사용해 동화상, 정지화상, 음성, 가상현실, 텍스트(Text)등을 지원할 수 있는 멀티미디어환경의

프로토콜이다. IETF(Internet Engineering Task Force)라는 인터넷 그룹에 의해서 발표된 개념이며 IETF HTTP Working Group에 의해 추가/수정되어지고 있다. 이 프로토콜은 분산 하퍼미디어 인포메이션 시스템에 적합하고 빠른 응용레벨(Application Level)의 프로토콜로서 요청/응답 페러다임을 기본으로 한다.

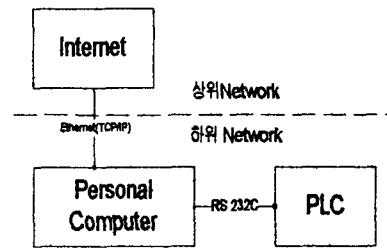


그림 10. 상/ 하위 네트워크의 구조.

논문에서 사용한 HTTP는 IETF의 RFC1945를 기반으로 하는 HTTP 버전1.0을 사용하여 웹서버를 구축하였다. 웹서버는 그림.11에서 보는 것과 같이 브라우저에서 서버에 접속하여 데이터를 요청하면 서버는 요청한 데이터에 대한 응답으로 HTTP의 헤드(head)와 HTML^[4]문서를 전송한다. 그런데 HTML문서는 클라이언트에게 정(靜)적인 문서만을 출력할 뿐이다. 그래서 외부 프로그램을 가동하고, 동적인 문서를 얻을 수 있는 CGI프로그램을 제작하여 모니터링했다.

그 과정(그림 11.)을 살펴보면

- ① 가스센서 상태를 감지하기 위해 브라우저는 서버에 접속하여 데이터를 요청한다.
- ② 서버는 CGI를 실행한다.
- ③ PLC에 연결된 가스센서의 접점상태를 파악하기 위해 컴퓨터와 PLC간에 데이터 송/수신이 이루어진다.
- ④ PLC로부터 수신된 데이터 프레임을 바탕으로 브라우저에 응답한다.

4.1 CGI 제작

CGI는 웹하이퍼미디어시스템과 외부 응용 프로그램이 서로 연계되기 위한 인코딩(Encoding), 디코딩(Decoding) 방법과 데이터 교환의

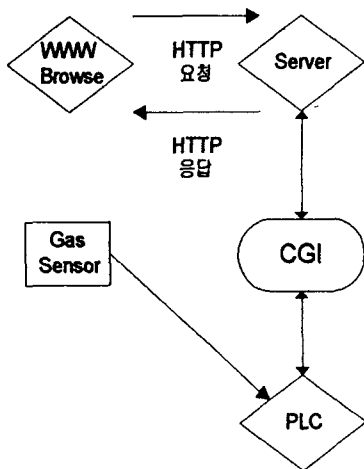


그림 11. 웹서버의 감시/ 제어 구조.

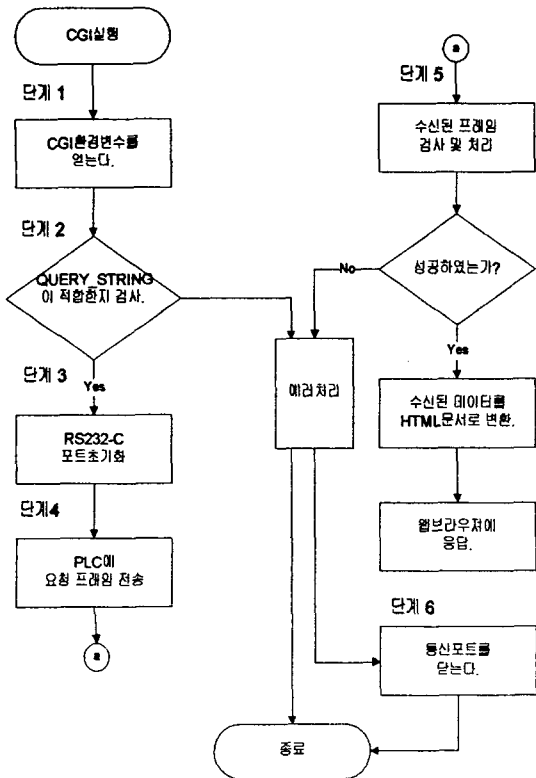


그림 12. CGI프로그램 flow chart.

방법을 정의한 표준이다. 이를 이용하여 PLC의 감시 및 모니터링을 위한 그림 12.의 CGI프로그램을 작성하였다. 사용한 컴파일러는 Borland C++4.5이고, Windows 32bit 플랫폼에 메모리

모델은 GUI로 하였다.

단계별로 살펴보면 아래와 같다.

단계 1. 그림 11.에서와 같이 웹브라우저가 서버에게 URL를 요청하면 HTTP는 이를 해석하고, 그것이 CGI인 경우에는 환경변수를 통해 CGI에 전달한다. 즉, 검색하고자 하는 가스센서의 입력을 QUERY-STRING(그림 13.)이란 인자로 받아 CGI가 처리한다는 의미이다.

그림 13.은 HTTP에서 CGI를 위해 사용하는 환경변수이다. Query-String의 변수는 CGI에 입력된 내용을 받아들이는 변수로서 아래 표 2.과 같이 인자를 받아들일 수 있게 CGI프로그램을 제작했다.

```

struct Environment (
    char *Server_Software;
    char *Server_Name;
    char *Gateway_Interface;
    char *Server_Protocol;
    char *Server_Port;
    char *Server_Admin;
    char *Request_Method;
    char *HTTP_Accept;
    char *Path_Info;
    char *Path_Translated;
    char *Script_Name;
    char *Query_String;
    char *Remote_Host;
    char *Remote_Addr;
    char *Auth_Type;
    char *Remote_User;
    char *Remote_Ident;
    char *Content_Type;
    char *Content_Length;
    char *HTTP_User_Agent;
    char *HTTP_REFERER;
    char *Auth_User;
    char *Auth_Password;
);
    
```

그림 13. CGI프로그램 변수.

표 2. 인자의 상태값.

인 자	기 능
gas1	센서 1번의 상태를 읽어 들인다.
gas2	센서 2번의 상태를 읽어 들인다.
gas3	센서 3번의 상태를 읽어 들인다.
gas4	센서 4번의 상태를 읽어 들인다.
all	모든 센서의 상태를 읽어 들인다.
start	CPU를 가동 시킨다
stop	CPU를 정지 시킨다

단계 2. 단계 1에서 얻어진 인자가 인자로서 적합한지 아닌지를 검사하고 적합하지 않을 경우, 에러처리를 웹브라우저에게 응답하고 종료한다.

단계 3. 적합한 인자가 얻어지면 컴퓨터와 PLC간의 데이터 송수신을 위해 RS232-C포트를 초기화한다. 본 논문에서는 CGI프로그램 작성을 위해 WIN32 AP 함수를 사용하였다.

단계 4. 초기화가 성공적으로 수행되면 그림 4, 그림 5.나 그림 6.의 요청프레임 구조로 PLC에 전송한다. 그림 14.의 ①, ②, ③은 본 논문에서 사용한 요청프레임이다.

```

char signal_buffer[13] = {'\5', '0', '0', 'B', 'R', 'X', '0', '0', '1',
                          '6', '0', '6', '\4'};           ①

char signal_buffer[6] = {'\5', '0', '0', 'R', 'R', '\4'};           ②

char signal_buffer[6] = {'\5', '0', '0', 'R', 'S', '\4'};           ③
    
```

그림 14. 요청프레임의 선언.

- ① 4개의 가스센서를 읽어들이기 위한 요청프레임(그림 4.)이다.
- ② PLC CPU를 Remote Run하기 위한 요청프레임(그림 5.)이다.
- ③ PLC CPU를 Remote Stop하기 위한 요청프레임(그림 6.)이다.

단계 5. 단계 4가 성공하면, PLC는 컴퓨터의 수신버퍼에 그림 7.이나 그림 8, 그림 9.와 같은 프레임이 수신되었는지를 검사한다. 검사된 결과가 오류로 판단되면 에러처리 메시지를 HTTP를 통해 출력하고 통신포트를 닫은 후 종료한다. 반대로 검사된 결과가 성공적으로 수신되었으면 버퍼의 프레임을 읽어 들인다. 그리고 이 응답프레임의 "데이터" 부분을 배열에 저장하여 각각의 센서상태를 점검한다. 만약 0이 수신되어 있으면 가스누출이 감지된 것을 의미하고 1이 수신되어 있으면 가스누출이 없다는 것을 의미한다. 다음 예는 수신된 데이터 프레임의 예를 보인 것이다.

예) 만약, X0016, X0017의 접점에 연결된 가스센서가 가스누출을 감지하고 X0018와 X0019의 접점에 연결된 가스센서는 가스누출이 감지되지 않았다면 다음과 같은 데이터 프레임이 수신된다.

```

char Inbuffer[10] = { '\2', '0', '0', 'B', 'R',
                      '1', '1', '0', '0', '\3' };
    
```

이와 같이 수신된 프레임의 "데이터"부분을 검사하여 가스누출상황을 식별한 후 HTML 규격으로 변환하여 응답하게 했다.

단계 6. 위의 단계들이 모두 처리되면 통신포트를 닫고 CGI프로그램을 종료 한다.

4.2 감시 및 제어를 위한 CGI동작

PLC의 접점상태를 감시하고 제어하기 위한 CGI프로그램은 그림 15.와 같은 방법으로 동작한다.

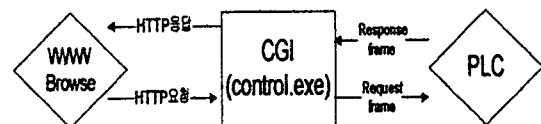


그림 15. CGI프로그램의 동작.

그림 15.의 과정을 살펴보면 아래와 같다.

- ① 웹브라우저가 웹서버의 CGI프로그램을 요청한다.
- ② CGI가 가동되면서 PLC에게 요청프레임을 전송한다.
- ③ PLC는 전송된 프레임에 대한 응답으로 응답프레임을 송신한다.
- ④ CGI는 응답프레임을 분석하고 Character 부분의 코드를 추출한다.
- ⑤ 추출한 데이터를 HTML문법에 맞게 재 구성하고 요청한 웹브라우저에 응답한다.

이와 같은 동작을 수행하기 위하여 제작한 CGI프로그램은 각각의 가스센서 상태를 개별적으로 감시하거나 한번에 모든 센서상태를 감시할 수 있도록 했고, 필요한 경우 PLC CUP를 정지하거나 가동할 수 있게 했다. 이를 위해서는 그림 12.에서의 단계1에서 얻어진 인자로 처리한다.

4.3 HTML처리

웹페이지는 HTML문서를 이용한다. 이 문서는 지시자(tag)와 이와 함께 사용하는 문장으로 이뤄지는데, 특정한 문장을 만드는데 있어 문장의 형태나 구성요소를 지시하는 지시자를 이용해 구분하게 되며, 본 논문에서는 CGI프로그램을 실행시키기 위해서는 HTML문서에 <FORM>이란 지시자(tag)를 사용하여 HTML문서를 작성했다.

그림 16.에서, ACTION은 입력된 내용을 전달하기 위한 프로그램 이름이 오고 METHOD에는 해당 입력 내용이 어떤 형식으로 프로그램에 전달되는가를 나타내는 것이다.

```
<FORM METHOD="get" ACTION="/cgi-bin/control.exe">
stop OR start <INPUT TYPE="text" NAME="control"><P>
</FORM>

<FORM METHOD="get" ACTION="/cgi-bin/control.exe">
gas1, gas2, gas3, gas4: <INPUT TYPE="text"
NAME="monitoring"><P>
</FORM>
```

그림 16. <FORM> tag^[4].

이 방법에는 두 가지가 사용될 수 있는데 post와 get이다. get의 경우 입력될 내용이 많을 경우 스택오버플로(stack overflow)가 일어날 수 있는 단점이 있긴 하지만 입력될 내용이 많지 않은 관계로 get을 사용하여 입력된 내용을 프로그램의 인자로서 넘겨주게 했다.

그림 16.은 명령어방식을 사용하여 입력하는 것에 반해 아래의 그림 17.은 영상을 보면서 모니터링하고자 하는 부분을 마우스로 클릭하여 감시할 수 있도록 제작한 HTML문서의 Image-map 지시자이다.

```
<p><a href="/thesis/map/gas.map">
<IMG SRC="/img/map/gas.jpg" ISMAP WIDTH=420
HEIGHT=300
BORDER=4 ALT="GAS"></a><P>
```

그림 17. Imagemap tag^[4].

5. 결 과

그림 18.은 가스센서의 가스누출 여부를 감지하기 위해 웹브라우저가 서버에 접속한 홈페이지의 모습이다.

가스배관(그림 18.의 ①)에 부착되어 있는 가스센서를 마우스로 클릭하면 그림 19.와 같이 가스누출 여부를 확인할 수 있다. 또한 Edit Box(그림 18.의 ②③)에 텍스트로 입력하여도 그림 19.와 동일한 결과를 얻을 수 있는 명령어 방식도 지원할 수 있다.

6. 결 론

기존의 원거리 모니터링시스템은 모뎀을 사용하거나 전용네트워크를 구축하여 모니터링하고 있는 실정이다. 그러나 모뎀은 유도 전송매체가 두 디바이스간의 직접링크를 제공하여 오직 디바이스들만의 접속에 의한 점대점(point-to-point)전송을 사용함으로써 여러 사용자들이 동시에 모니터링할 수 없는 단점이 있고, 아날로그 전송방식에서 중계전송로에 발생하는 잡음이나

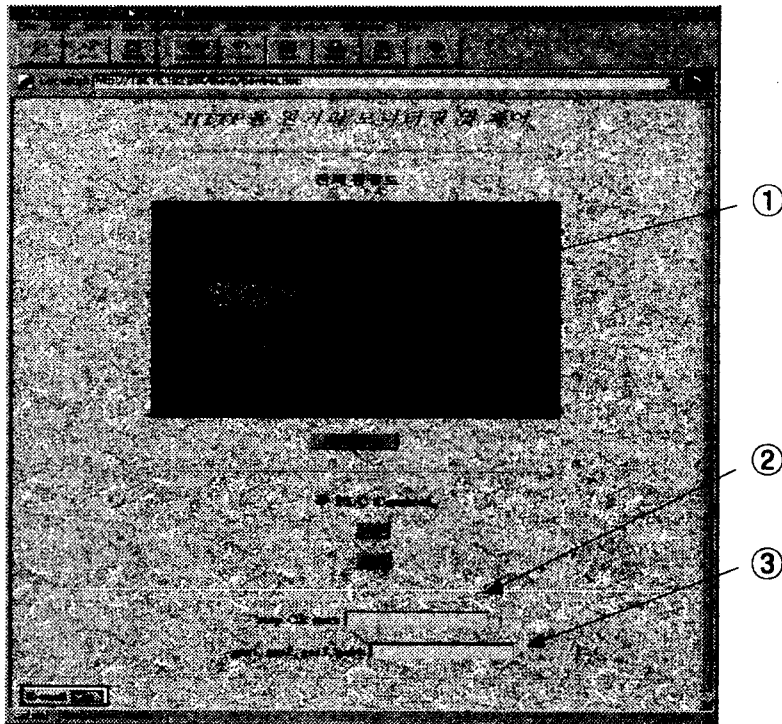


그림 18. 가스센서 모니터링 화면.

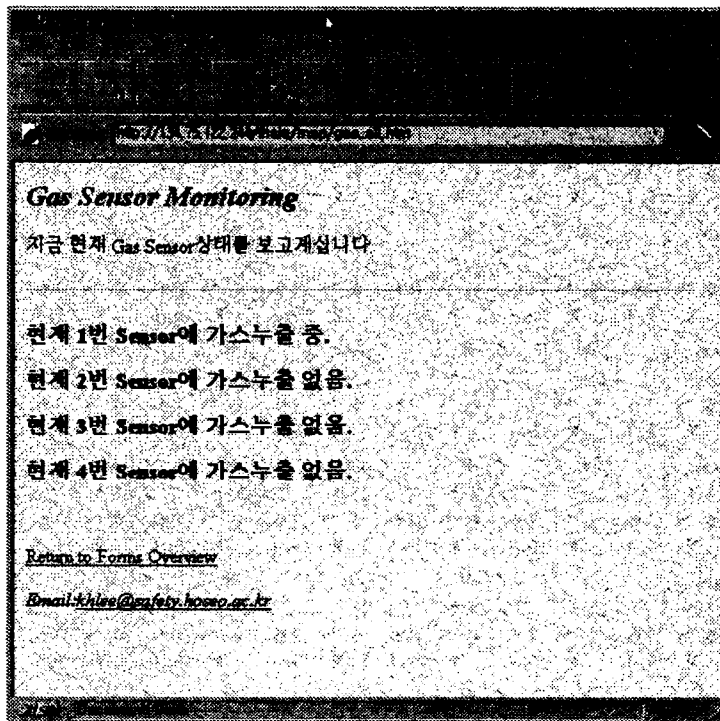


그림 19. 가스센서 모니터링 결과.

왜곡의 누적으로 인한 데이터 송수신의 문제는 장거리 전송에 적합하지 않다. 그리고 전용네트워크를 구축하여 사용하는 원거리 모니터링시스템은 고가의 설치비용과 네트워크의 유지, 보수 의 어려움으로 소규모의 사업장에는 적합하지 않다. 때문에 대용량의 데이터교환을 요구하는 멀티미디어환경과 다중사용자 환경에서 모니터링을 위해 인터넷상의 웹프로토콜인 HTTP 와 PLC의 전용프로토콜을 사용하여 장소와 시간에 구애받지 않고 위험요소를 감지할 수 있는 원거리 모니터링시스템을 구축하였다.

본 논문에서는 기존의 RS232-C 통신포트를 지원할 수 있는 R형 수신기나 PLC 등을 장소와 시간에 구애받지 않고 모니터링할 수 있도록 인터넷과 연계 방안을 제시하였고, 여러 제작사들의 독자적인 모델을 지향함으로 인한, 소프트웨어의 호환성 문제를 여러 플랫폼에서 포팅되어 사용할 수 있는 웹브라우저를 사용하기 때문에 투명한 환경에서도 사용할 수 있었다. 그리고 웹은 멀티미디어를 지원하고 CGI를 이용해 외부 프로그램을 가동시킬 수 있음은 물론 암호화(security)기능까지 가지고 있어 전역적인 멀티미디어 통신환경의 모니터링시스템에서도 사용할 수 있다는 것을 보였다. 또한 RS232-C통신이 가능한 어떠한 시스템과도 상호 호환할 수 있어 설비의 재투자 없이 인터넷 통신환경의 중앙집중적 감시시스템을 구축할 수 있었다.

참 고 문 헌

1. IETF, "IETF Hypertext Transfer Protocol (HTTP) Working", 1994. World-Wide-Web, URL=//http://www.ics.uci.edu/pub/ietf/http/
2. Andrew S. Tanenbaum, "Computer Network", Prentice-Hall, Inc, pp.681-723, 1996.
3. 금성기술센터, "G5/G2 시리즈 DATA LINK 과정 매뉴얼", 금성산전, pp.9-36, 1994.
4. Dave Raggett, "Hypertext markup language specification version3.0", HTML-WG of IETF, pp.19-20, 28th, March, 1995.
5. James L. Conger, "Windows API Bible II", Wait Group, pp.719-748, 1992
6. U. Minoni, G.Sansoni, and N. Scara-bottolo, "A Fault Tolerant Microcomputer Ring for Data Acquisition in Industrial Environments," IEEE Trans. on Instrumentation and Measurement, Vol.38, No.1., pp.32-36, Feb. 1989, .
7. 박용렬, "분산제어 시스템에서 개방화 구조 설계에 관한 연구", KITE 하계종합학술대회 논문집, Vol.19, 1996.
8. 박종태, "OSI망관리와 인터넷 관리의통합구조", 한국정보과학논문지, Vol.21, NO.9, pp. 1722-1729, 1994.
9. 박성원, "정보고속도로의 구축시 통신망 연동 방향 및 통합 연동 모델", 전자공학회지, Vol. 23, pp.423-431, 1996.
10. 장광규, "PLC를 이용한 공정제어 모니터링 시스템의 개발에 관한 연구", 한양대 석사학위 논문, 1994.
11. Addison-Wesley, "A. Valenzano, C. Demartini, and L. Ciminiera MAP and TOP Communications" Standards and Applications, 1992.
12. ED Krol, "THE WHOLE INTERNET", O'Reilly & ASSOCIATES INC., pp.19-30, 227 -242, 1992.
13. K. Obraczka, et. al., "Internet Discovery Service", IEEE Computer, pp.8-22., Sep. 1993.
14. Kilnam Chon, "Global Network from Asia-Pacific Perspectives", Proceeding of INET '93, Aug.1993.

본 연구는 호서대학교 공업연구소 연구과제에 의해 수행되었으며 이에 감사드립니다.