

논문 2005-42CI-4-1

# PLC 모니터링을 위한 임베디드 HMI 시스템의 개발에 관한 연구

(Study on Development of Embedded HMI System for PLC Monitoring)

선 복 근\*, 한 광 록\*\*, 임 기 욱\*\*\*

(Bok Keun Sun, Kwang Rok Han, and Kee Wook Rim)

## 요 약

최근 PLC들은 산업 자동화 분야에 필요한 자동화장비에서 널리 사용되고 있다. HMI는 다수의 PLC들을 효과적으로 제어하기 위해서 필수적인 시스템이다. 초기 HMI시스템은 단순한 아날로그 계기들을 사용하여 구성되었으나 반도체 및 디스플레이 기술의 발달로 최근에는 각종 디지털 부품들로 구성된 임베디드 시스템으로 구성되어 디스플레이 화면을 통해서 PLC를 제어 또는 감시할 수 있게 되었다. HMI시스템은 이러한 HMI 장치와 HMI 장치를 구동시키기 위한 프로그램 그리고 HMI 화면을 편집 하는 프로그램 세 가지 구성요소로 이루어져 있으며 화면편집 프로그램은 실제 HMI 장치에서 표현되는 각각의 화면들을 사용자가 편집할 수 있도록 다양한 화면 구성요소들을 제공해야 한다. 본 논문에서는 소형 HMI 장치용 화면편집 프로그램을 객체지향 언어를 통해 설계 및 구현하며, 임베디드 HMI 시스템의 프로토타입을 제안한다.

## Abstract

Recently, most of PLC has been using widely in automation equipment that is needed for field of industry automation. HMI is essential system for effective control of many numbers of PLC. Even though early HMI system was consists of simple analog devices, however, HMI system could control or supervise PLC through display screen since it embeds various digital parts by development of technology in these days. HMI system consists of three parts as HMI hardware, an operating program for HMI equipment and a HMI screen editor. Among these elements, a program for editing screen of HMI should provide various screen elements that helps users to edit each screens displayed on HMI. In this study, we design and implement screen edit program by object-oriented method for small HMI equipment and propose prototype of embedded HMI system.

**Keywords** : PLC, HMI, 임베디드 시스템, 화면편집 프로그램

## I. 서 론

현재 제조업 분야에서는 높은 인건비를 해결하고 제품의 불량률을 낮추고 또한 높은 생산 수율을 위하여 많은 노력과 자본을 투자하고 있다. 그 중에서도 제품의 생산에 필요한 공정을 사람이 아닌 기계를 통하여

자동화하는 방법이 넓게 사용되고 있는데 이러한 생산의 자동화를 위한 공장 자동화 시스템을 위해서 다양한 종류의 자동화 장비 및 부품들이 사용되고 있다. 이러한 공장 자동화 분야에서 대표적으로 사용되는 기기 중 하나가 PLC이다.

PLC는 프로그래밍 될 수 있는 순차 논리에 의해 ON/OFF 제어만이 수행되는 일종의 소형 컴퓨터로 ON/OFF 신호를 받아 정해진 ON/OFF 제어를 하므로 신호등 제어, 자동화 기계 제어 등을 할 수 있고, 공장 자동화 및 컴퓨터통합생산(CIM)에 필수적으로 사용되는 핵심장치이다<sup>[2][9]</sup>. PLC는 1978년 NEMA (National Electric Manufacturing Association)에서 PLC

\* 정회원, \*\* 중신회원, 호서대학교 컴퓨터공학과  
(Dept. of Computer Engineering, Hoseo University)

\*\*\* 중신회원, 선문대학교 지식정보산업공학과  
(Dept. of Knowledge and Industrial Engineering,  
SunMoon University)

※ 본 연구는 산업자원부의 지역혁신 인력양성사업의 연구결과로 수행되었음.

접수일자: 2005년3월21일, 수정완료일: 2005년6월22일

(Programmable Logic Controller)로 명명하였으며 "디지털 또는 아날로그 입출력 모듈을 통하여 로직기능, 시퀀싱, 타이밍, 카운팅, 연산과 같은 특수한 기능을 수행하기 위하여 프로그램 가능한 메모리를 사용하고 여러 종류의 기계나 프로세서를 제어하는 디지털 전자 장치" 라고 정의하고 있다.

이와 같은 PLC가 이용되는 자동화 장비의 사용분야가 다양해지고, 요구되는 기능이 복잡해짐으로써 PLC들로 구성된 자동화 장비를 운영자가 효과적으로 제어 및 감시하기 위한 사용자용 인터페이스 장치가 필요하게 되었다<sup>[5]</sup>. 자동화 장비의 제어 및 감시를 위한 사용자용 인터페이스 장치를 MMI(Man Machine Interface) 장치 또는 HMI(Human Machine Interface)장치라고 하는데 초기의 HMI는 반도체의 집적도가 현재보다 상대적으로 낮고, 또한 디지털화가 미비하여 장치의 크기가 크고, 이동성이나 설치도 용이하지 못하였기 때문에 새로운 자동화장비를 위한 HMI 장치의 개발비용이 추가로 들어야 한다는 단점이 있었다<sup>[6]</sup>.

HMI 장치는 사용목적과 기능에 따라 입출력장치, 연산장치, 저장장치를 가지는 임베디드 시스템(Embedded System)으로 구성된다. 또한 사용되는 다양한 하드웨어 환경에 맞는 프로그래밍 언어를 사용하여 프로그램을 개발 및 내장하여 장치를 구동시킨다. 이와 같이 HMI 장치를 통해서 PLC의 감시나 제어를 위한 프로그램을 개발하기 위해서는 하드웨어 지식과 소프트웨어 지식을 갖춘 고급의 개발인력과 많은 개발시간이 필요하고 특히 시스템의 변경 시 수정작업이 용이하지 않다는 단점이 있다<sup>[1][3]</sup>.

반도체 집적도의 향상 및 설계기술의 발달로 각종 VLSI, 디스플레이 모듈, SoC, 메모리 칩과 같은 디지털 핵심 부품등의 가격이 하락되고 성능이 증가하면서 고성능 부품이 내장된 HMI 장치를 값 싸게 구성할 수 있게 되었다<sup>[4]</sup>.

최근에 흑백 또는 칼라 그래픽용 LCD를 사용하는 HMI 장치들이 널리 사용되고 있고, 사용자 인터페이스가 GUI(Graphical User Interface)화 되어지고 있는 상황에서 HMI용 GUI를 유연하게 편집하기 위한 저작 프로그램이 필요하게 되었다<sup>[6]</sup>. 임베디드 시스템이 가지고 있는 하드웨어적인 자원과 마우스나 키보드와 같은 편집하기 위해서 필요한 각종 장치들의 제한으로 인하여 HMI의 GUI의 화면을 편집하기 위한 장치로서 PC가 사용되고 있다.

사용자는 PC기반의 저작 프로그램을 통하여 HMI 장

치용 GUI를 작성하고, 작성된 GUI의 데이터를 HMI 장치로 전송 및 저장시켜서 HMI 장치의 구동 프로그램이 PLC 감시 및 제어용 GUI 프로그램을 동작시키는 구조를 가진다.

본 논문에서는 시리얼 통신을 통하여 PLC의 데이터 제어 및 감시를 수행하기 위한 HMI 장비의 GUI를 작성할 수 있는 PC 기반의 HMI 화면편집용 저작 프로그램을 설계 및 구현하고, 임베디드 HMI 시스템의 프로토타입을 제안한다.

논문의 구성은 제II장에서 PLC 와 HMI 시스템의 개요를 소개하고, 제III장에서는 객체지향 프로그래밍 기법을 도입하여 HMI 장치용 GUI를 편집하는 PC 기반 저작 프로그램을 설계 및 구현한다. 제IV장에서는 임베디드 HMI 시스템의 프로토타입을 제안하며, 제V장에서 연구결과 및 향후 연구과제에 대하여 논한다.

## II. PLC 와 HMI 시스템 개요

### 1. PLC의 종류 및 구조

PLC는 크기에 따라 다양한 종류를 가지고 있다. 크기 및 처리하는 데이터에 따라 PLC를 소형, 중형, 대형의 3가지로 분류하며, 각각 독특한 동작 특성을 가진다. 소형의 범위는 128 개의 입.출력점과 2K word 메모리 까지를 말한다. 이들 PLC는 간단한 고수준 기기 제어를 가능하게 해준다. 중형 PLC는 2048 개의 입.출력점과 32K word 메모리까지를 말한다. 특수 입출력 모듈은 중형 PLC가 온도, 압력, 유출, 무게, 위치 또는 보통 공정제어 응용에서 만나게 되는 모든 형태의 아날로그 기능을 수행할 수 있도록 해준다. 물론 대형 PLC는 PLC 종류 가운데 가장 복잡한 유니트이다. 대형 PLC는 16,000 개의 입.출력점과 2M word 메모리를 가지며 응용범위가 매우 넓다. 또한 개별 생산 공정 또는 공장 전체를 제어할 수 있다<sup>[9]</sup>. PLC의 크기를 선택하는데 있어서 중요한 점은 PLC 시스템이 현재의 조건을 충족시키면서 미래의 응용 조건도 충족시키는 적절한 크기인가 검토하는 것이다.

PLC는 기계장치를 운전, 제어하기 위하여 사용함으로써 열악한 생산현장에서도 견딜 수 있도록 온도, 습도나 전기적 노이즈에 강하고 취급이 용이한 구조로 되어 있다. PLC의 구성은 업무용 PC의 구성과 거의 같다. PLC의 기본 구성은 그림 1과 같이 사람의 두뇌에 해당하는 CPU, 센서나 스위치로부터 신호를 입력 받는 입력부, 제어대상에 제어신호를 출력하는 출력부, 시퀀스

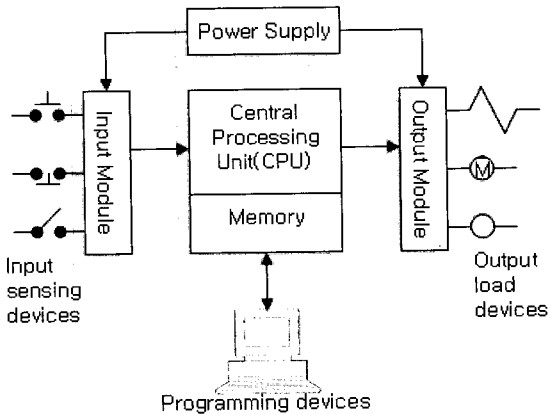


그림 1. PLC의 구조  
Fig. 1. Structure of PLC.

프로그램을 저장하는 기억부, PLC의 각부분에 전원을 공급하는 전원부로 되어 있다. 기본장치와 더불어 프로그램의 작성을 위하여 핸드 로더라는 프로그램 콘솔이나 컴퓨터에서 직접 그래픽으로 프로그램을 작성할 수 있는 소프트웨어와 PC, 작성한 프로그램을 EPROM에 보관하기 위한 EPROM writer, 프로그램을 인쇄하기 위한 프린터 등의 주변장치가 필요하다<sup>[9]</sup>.

2. HMI 시스템 구성 요소

HMI 장치가 올바르게 작동하기 위해서 필요한 시스템 구성 요소는 각각 담당하는 기능별로 임베디드 시스템으로 구성된 HMI 장치, HMI 구동 프로그램, HMI 화면편집 프로그램 3가지로 분류할 수 있다.

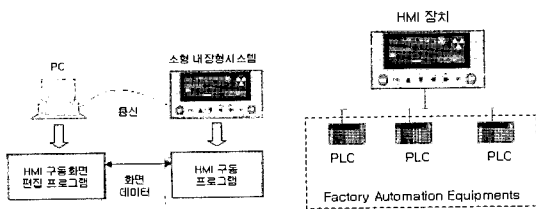


그림 2. HMI 시스템 구성 요소  
Fig. 2. Construction Elements of HMI System.

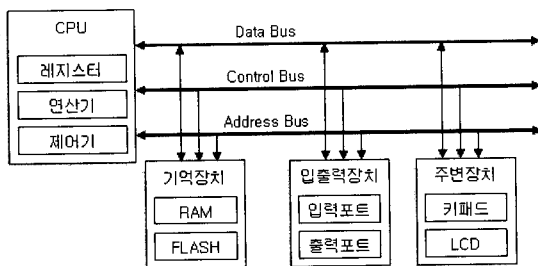


그림 3. HMI 장치 구조  
Fig. 3. Structure of HMI Equipment.

가. 임베디드 시스템으로 구성되는 HMI 장치

HMI 장치는 필요한 최소한의 하드웨어 자원을 사용하여 그림 3과 같은 임베디드 시스템으로 구성되며 각각의 하드웨어 부품은 HMI 장치의 기능 명세와 함께 결정이 된다.

나. HMI 장치의 구동 프로그램

HMI 장치에서 구동되는 프로그램은 해당 연산 장치의 기계적인 명령어가 다르기 때문에 프로그램 개발을 위해서 CPU에 의존적인 프로그램 도구를 사용해야 한다. 각 임베디드 시스템을 위한 CPU는 크로스 컴파일러와 같은 개발도구를 함께 제공하기 때문에 이를 사용하여 프로그램을 작성한다.

임베디드 시스템은 최소한의 프로그램 코드만으로 구동되기 때문에 하드디스크와 같은 보조 기억장치를 가지고 있지 않고, 필요한 코드를 비휘발성 메모리인 플래시 메모리에 저장하여 전원이 들어옴과 동시에 해당하는 플래시 메모리번지로 이동하여 명령어를 실행하도록 구성한다. 그러므로 플래시 메모리에 저장되는 프로그램이 갱신되는 경우 별도의 하드웨어 장치를 통해서 플래시 메모리에 다시 프로그램 코드를 써넣어야만 한다. 이와 같은 형태의 프로그램을 펌웨어라고 하는데 각종 장치와 관련된 하드웨어 지식이 없이는 프로그램을 작성할 수 없다.

다. HMI 화면편집 프로그램

일반적으로 PC를 기반으로 한 저작 프로그램을 의미하며, 실제 HMI 장치에서 구동되는 모습과 동일한 형태를 화면(Screen)에 나타내어 편집하도록 GUI를 사용한 편집기능을 제공한다. 또한 여러 가지의 PLC나 HMI 장치가 사용되기 때문에 하나의 편집 프로그램이 여러 가지 장치 사용의 경우를 고려하여 지원해야 한다.

각 화면에 배치되는 화면요소 객체는 마우스나 키보드로 사용자가 조작할 수 있어야 하며 편집된 화면들이 HMI 장치로 로딩 된 후에 PLC와 통신하면서 정보를 주거나 받은 후 편집 시 배치된 위치에서 그 값을 표현해주는 역할을 하는 태그 객체와 각종 선, 사각형, 다각형, 원 등과 같이 단순히 화면을 꾸며주는 역할을 하는 도형 객체로 나눌 수 있다. 본 논문에서는 특별한 도형 객체를 제공하지 않고 태그 객체만을 통해서 구현하는 것을 목표로 하였다.

편집 프로그램으로 작성된 데이터들은 실제 HMI 장치에 저장될 수 있도록 장치와의 데이터 송수신 및 데

이터의 변환 기능이 필요하며 편집된 화면들과 화면요소들로 구성된 데이터와 함께 각종 파라미터, 공용자원, PLC 통신 정보 등이 포함되어야 한다.

### III. 임베디드 HMI 장치용 GUI 편집기 설계 및 구현

본 장에서는 소형 임베디드 시스템으로 구성된 HMI 장치와 HMI 구동 프로그램의 구조를 이해하여 HMI 장치의 GUI를 편집하는 PC기반의 HMI 화면편집 프로그램을 설계 및 구현한다.

소형 HMI 화면편집 프로그램은 향후에 보다 많은 하드웨어적인 자원을 가지고 있어서 다양한 기능들을 추가할 수 있는 중형 HMI 장치를 위한 구동프로그램 및 구동화면편집 프로그램의 설계 및 구현의 기초 연구 자료를 제공한다. 또한 구현환경은 현재 가장 많이 사용되고 있는 운영체제인 Microsoft Windows 운영체제를 기반으로 하고 개발언어는 Microsoft의 객체지향 개발언어인 Visual C++을 사용하였다<sup>[11][14]</sup>.

소형 HMI 장치에서 동작하는 구동 프로그램은 매우 열악한 메모리 자원을 가지고 있기 때문에 메모리를 효과적으로 사용하기 위하여 고정된 사이즈의 메모리 맵을 가지고 있고, 소형 HMI 장치의 경우는 흑백의 디스플레이만 가능하며, HMI 장치와 통신하는 PLC의 제조사 및 종류가 매우 다양하고 그 통신 프로토콜이 다르기 때문에 이러한 모든 프로토콜들을 저장해 놓을 메모리 공간도 부족하다<sup>[7]</sup>.

또한 HMI 화면편집 프로그램은 그 사용자가 현재 편집하고 있는 화면이 바로 HMI 장치에서 동작하는 모양과 동일하게 느낄 수 있도록 직관적인 인터페이스를 제공해야 하며, HMI 장치가 향후에 다양하게 제공될 수 있기 때문에 추가되는 HMI 장치들을 위한 화면편집 프로그램의 갱신이 용이하도록 확장성을 가져야 한다.

따라서 이와 같은 다양한 요구사항들을 만족시킬 수 있도록 화면편집 프로그램의 설계 시 이러한 요구사항들을 충분히 고려해야 한다.

1. 소형 HMI용 화면편집 프로그램의 구성요소 설계  
편집 프로그램의 외형 및 데이터관리자의 개념도는 그림 4에 나타낸다.

그림 4는 구현할 HMI 화면편집 프로그램의 GUI와 데이터관리자를 도식화한 것이다. 화면은 실제 HMI 장치의 디스플레이 모듈의 화면을 표현하는 부분으로 이

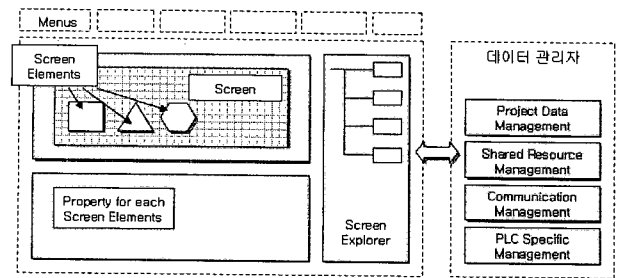


그림 4. HMI 화면편집 프로그램 외형 및 데이터관리 모듈

Fig. 4. HMI screen editor layout and data management module.

화면의 유효한 영역에 사용자가 필요한 그래픽 객체(태그 객체 또는 도형 객체)를 선택하여 배치시킬 수 있어야 한다. 아래에 프로그램의 기능별로 구분하여 설명한다.

- 태그 객체: HMI 장치는 PLC의 특정 메모리 영역에 접근하여 값을 쓰거나 읽어서 제어 또는 감시를 수행하는데 이와 같이 HMI 장치가 접근해야 하는 PLC의 위치, 읽거나 쓸 값, 타입, 화면표현 방식 등 다양한 속성들을 내포하고 있는 객체를 태그 객체로 정의한다.

- 도형 객체: HMI 장치의 화면으로 태그뿐 아니라 화면의 가독성을 높이고, 해당 태그의 값을 표현하기 위해서 추가로 도형이나 그림 등이 추가로 배치되는 것이 필요한데 이러한 객체를 도형 객체로 정의한다. 이러한 도형 객체는 단지 화면의 특정 위치에서 그려지지만 할 뿐 PLC와의 통신은 필요하지 않다.

- 구성요소 속성: 현재 편집중인 화면, 태그, 도형 등에는 필요한 속성들을 사용자가 입력할 수 있어야 하기 때문에 다이얼로그 형태의 속성윈도우가 필요하다.

- 화면탐색: HMI는 사용자가 조작하는 화면이 다수가 존재하고 이를 자유롭게 이동하면서 조작을 수행하는데 편집 프로그램 역시 여러 개의 화면을 편집할 수 있고 편집된 화면들을 탐색하고 복사, 붙이기, 삭제, 생성 등의 기능이 되어야 한다. 이와 같은 기능을 수행하기 위해서 트리 또는 리스트형태의 윈도우가 제공되어야 한다.

- 프로젝트 데이터 관리(Project Data Management): HMI 화면편집 프로그램은 하나의 프로젝트 단위로 관리하는 객체로 사용자가 편집한 모든 화면 데이터를 저장하고 불러오는 기능을 수행한다.

- 공유 자원 관리(Shared Resource Management): HMI 화면편집 프로그램에서 특정 태그 객체는 이미지

나 문자열을 화면에 표시하기 위해서 문자열이나 이미지 데이터를 HMI 장치로 전송해 주고 이를 참조하여 HMI 장치의 화면에 표시할 수가 있다. 소형의 HMI의 경우 화면편집 프로그램에 의해서 편집된 데이터를 저장할 수 있는 공간이 풍부하지 못하기 때문에 이러한 관리자를 통해서 같은 내용을 가지는 이미지나 문자열을 공유할 수 있도록 한다.

- 통신 관리(Communication Management): HMI 화면편집 프로그램이 편집한 데이터는 프로젝트 관리자에 의해서 알맞은 형태로 저장이 이루어진 후에 이를 HMI 장치로 전송하여야 한다. HMI 장치와 PC는 시리얼 통신을 통해서 편집된 데이터가 저장되어야 하는 공간에 적재시켜야 하는데 HMI 장치와 데이터의 송수신을 담당하는 객체가 있어야 한다.

- PLC 규격 관리(PLC Specific Management): HMI 장치는 다양한 종류의 PLC와 통신하며 제어 및 감시를 수행해야 한다. PLC들은 고유의 통신 프로토콜과 어드레스 영역이 있기 때문에 이에 대한 정보를 모두 HMI 장치에 저장할 수는 없으므로, HMI 화면편집 프로그램으로부터 해당하는 정보를 전송받아 사용해야 한다.

## 2. 소형 HMI용 화면편집 프로그램의 클래스 설계

표 1의 클래스 테이블은 구현에 필요한 클래스들을 나열한 것이다. 클래스들은 프로그램 초기화시점에 한 번 생성되고 공유되는 클래스들과 런타임(Runtime)시 생성되는 클래스들로 구분할 수 있다.

표 1. HMI 화면편집 프로그램의 클래스  
Table 1. Classes of HMI screen edit program.

메인	CHMIApp
전체 뷰	CHMIView
프로젝트관리	CHMIDoc
프레임(툴바,메뉴)	CMainFrame
편집화면 뷰	CScreen
편집화면 데이터	CScreenData
화면 구성요소 추상클래스	CTag
화면 구성요소 클래스 (추상 클래스 상속)	CTextTag, CMessageTag, CLampTag, CImageTag, CLineGraphTag, CDigitTag, CButtonTag..
각 화면 구성요소 속성 관리(다이얼로그 박스)	CTextTagDlg, CMessageTagDlg, CLampTagDlg..
공용자원관리	CTextResMgr, CImgResMgr
HMI 장치와 데이터통신	CCommThread
PLC정보 관리	CPlcInfo, CPointInfo

### 가. 초기화 클래스

HMI 화면편집 프로그램은 Microsoft에서 제공하는 클래스 라이브러리에서 제안하는 기본 프레임워크를 기반으로 프로그램의 틀이 형성되는데 프로그램객체, 뷰 객체, 도큐먼트객체, 프레임객체 4개를 통해서 기본적인 하나의 프로그램을 이루게 된다. 프로그램객체가 생성되는 시점에 초기화 루틴이 호출되며, 도큐먼트, 뷰, 프레임객체가 생성되면서 각 객체들 사이의 참조가 가능하도록 연결고리가 형성된다<sup>[10]</sup>.

- 메인 클래스 (CHMIApp): 이 클래스가 인스턴스화되는 시점에서 CHMIDocument, CHMIView, CMainFrame 클래스 객체와 함께 CPlcInfo, CImageResMgr, CTextResMgr 클래스의 객체를 생성 및 초기화한다.

- 도큐먼트 클래스(CHMIDoc): 화면편집 프로젝트 전체 데이터를 포함하고 있다. 사용자가 하나 또는 여러 개의 화면편집을 가능하게 하기 위해서 편집화면의 데이터 모델을 담당하는 ScreenData객체(CScreenData의 인스턴스)가 런타임에 생성, 소멸, 복제 등이 될 때 생성된 ScreenData객체들을 리스트로 관리한다. 도큐먼트 클래스는 화면편집 데이터를 HMI 장치의 메모리맵에 저장할 수 있도록 변환하여 파일에 저장되고 저장된 파일을 역변환하여 화면편집 데이터를 복원한다. 또한 HMI 장치와의 통신을 관리하는 CCommThread객체를 사용하여 데이터를 송신 또는 HMI 장치로부터 HMI 화면 데이터를 수신하여 역변환을 수행한다.

- 뷰 클래스(CHMIView): 도큐먼트 객체가 가지고 있는 데이터를 화면에 디스플레이하는 주요 클래스이다. 편집화면인 CScreen클래스의 인스턴스를 포함하고 있으며 그 이외의 화면탐색을 위한 CScreenList, 각종 태그 객체의 속성을 편집하기 위한 속성 다이얼로그 객체들을 특정위치에 배치한다.

- 프레임 클래스(CMainFrame): 이 클래스는 응용프로그램이 필요한 각종 메뉴 및 툴바를 포함한다.

- 편집화면을 표시하기 위한 뷰 클래스(CScreen): 실제 HMI 장치의 디스플레이 모듈에서 표시되는 화면을 편집 프로그램에서 나타내기 위한 클래스이다. 사용자가 태그 객체를 편집하는 주요 영역이 되는 클래스이며 실제 편집한 데이터의 내용부분을 CScreenData에 저장하도록 하여 MVC(Model-View-Controller)패턴을 형성한다<sup>[13]</sup>. 사용자가 편집하고자 하는 화면을 추가하는 경우 CScreen객체를 추가로 생성하지 않고 CScreen의 데이터모델인 CScreenData객체만을 생성함으로써 객체 생성으로 인한 메모리 낭비를 줄이도록 하였다<sup>[12]</sup>.

CScreen의 주요 기능은 CScreenData를 통해서 현재 편집하고자 하는 화면을 나타냄과 동시에 메뉴, 마우스, 키보드 이벤트를 통해서 발생한 사용자의 요청을 처리하기 위해서 현재 의 화면과 연관되는 CScreenData의 함수를 호출하여 화면정보를 갱신하고 CScreen객체가 이 정보를 리드로우(Redraw)하도록 요청한다.

- 공용 이미지관리자 클래스(CImageResMgr): 이 클래스 객체가 관리하는 아이템은 각각 고유의 ID를 가지는 이미지객체이다.

- 공용 문자열관리자 클래스(CTextResMgr): 이 클래스 객체가 관리하는 아이템은 각각 자신의 고유의 ID를 가지는 스트링객체이다.

- PLC정보들을 관리하는 클래스(CPlcInfo): HMI 화면편집 프로그램은 HMI 장치가 제어하고자 하는 PLC와 연결하기 위해서 통신이 가능한 PLC의 종류들과 통신을 수행하는 어드레스 영역과 프로토콜 정보를 저장하고 있어야 한다. 이러한 정보들은 특정 형식을 갖춘 파일에 저장하여 CPlcInfo라는 클래스를 통해서 응용프로그램의 시작 시점에 파일로부터 메모리에 적재한다. 따라서 지원 가능한 PLC의 개수만큼 CPlcInfo객체들이 생성될 수 있으며, 전역으로 그 리스트들이 관리되면서 화면편집 시에 필요한 경우 유효한 어드레스 영역정보를 알려주거나 프로젝트 데이터의 변환 시에 통신 프로토콜에 맞도록 변환할 때 해당 정보를 알려준다.

- 편집화면 탐색을 위한 리스트 컨트롤(CScreenList) : GUI컨트롤의 하나로 화면에 특정 데이터 모델의 리스트를 나타내는 클래스이다. CHMIView(응용프로그램 뷰 클래스)의 영역에 도큐먼트 객체가 가지고 있는 CScreenData들의 리스트를 화면에 표시한다.

- 속성다이얼로그(CTextTagDlg, CMessageTagDlg): 각 태그들의 속성을 변경할 때 사용되는 클래스들로서 편집하는 모든 화면과 그 화면 구성요소들은 각각의 설정되어야 하는 속성이 다르기 때문에 각 타입별로 하나의 속성 다이얼로그 박스가 필요하다.

나. 런타임 클래스

- 편집화면 데이터 클래스(CScreenData) : MVC패턴의 뷰-모델의 관계에서 CScreen클래스는 모델의 데이터를 화면에 나타내는 View, 사용자와 상호작용을 통하여 모델데이터를 갱신시키는 Controller에 해당하고 CScreenData는 Model에 해당 클래스로서 편집화면의 번호, 이름 그리고 화면을 구성하는 요소인 태그 객체

들의 리스트 등을 내부데이터로 가지고 있다.

- 화면 구성요소 추상클래스 및 상속클래스 : HMI의 화면을 구성하는 요소인 태그 객체를 정의한 클래스로서 실행시간에 사용자가 화면편집을 위해서 화면에 배치시키는 구성요소를 정의하는 클래스로서 그림 5와 표 2에서는 추상클래스인 CTag클래스를 상속한 클래스들의 상속관계와 추상클래스의 정의부분을 묘사하고 있다.

HMI 화면편집 프로그램과 같은 그래픽 편집기들은 그 사용자에게 친숙한 비주얼한 객체들을 표현해주고 그것을 직접적으로 사용자가 조작할 수 있도록 해준다<sup>[8]</sup>. 이러한 편집기들은 그 편집기의 특별한 요구사항들이 있기 때문에 일반적인 사용자 인터페이스를 만드는 것이 쉽지 않다. 예를 들어, 버튼이나 스크롤바와 같은 사용자 인터페이스 도구들을 조합하여 특정 인터페이스를 만들 수 있으나 이러한 편집기의 구현을 보다 쉽게 하기 위해서는 높은 수준의 추상화가 요구된다.

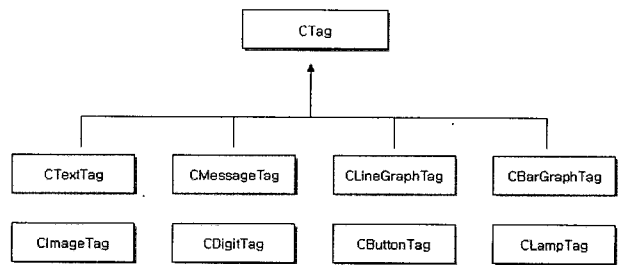


그림 5. 태그 객체 클래스 상속도  
Fig. 5. Class hierarchy of tag object.

표 2. 태그 객체를 위한 추상클래스인 CTag 클래스  
Table 2. CTag - Abstract class for tag objects.

<p><b>CTag 클래스</b></p> <pre> class CTag : public CRect{     unsigned int    m_posX, m_posY, m_type, m_id;     BOOL            m_bSelected;     CWnd*           m_pParent;     CTag(CWnd* pParent, int type, int id);     ~CTag();     SetPosition();     DrawTag();     Serialize(); }                 </pre>
<p><b>변수 및 함수 설명</b></p> <ul style="list-style-type: none"> <li>• m_posX, m_posY: 태그 객체의 left-top 좌표</li> <li>• m_type: 태그의 타입</li> <li>• m_id: 태그 생성시 부여되는 ID</li> <li>• m_bSelected: 태그가 마우스에 의해서 선택되었을 때 설정되는 플래그</li> <li>• SetPosition(): 태그의 위치를 변경하는 함수</li> <li>• DrawTag(): 화면 뷰 객체에 태그를 그려주는 함수</li> <li>• Serialize(): 각 태그를 특정스트림으로 직렬화하는 함수</li> </ul>

### 3. HMI 장치용 화면편집 프로그램의 구현

#### 가. 화면편집부분

그림 6은 화면편집에 사용되는 클래스들의 명칭과 설계 모델을 설명하고 있다. 또한 그림 7은 실행 시에 사용자가 태그를 추가하거나 이동 또는 그 속성을 변경할 때에 CScreen과 CScreenData 객체 사이에서 이루어지는 동작을 예를 들어서 나타낸 것이다.

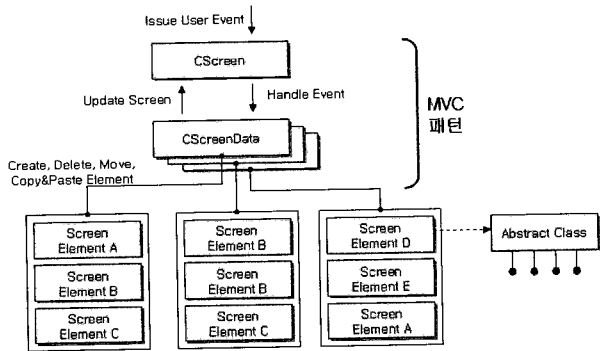


그림 6. 화면편집 부분의 클래스 관계도  
Fig. 6. Class relation of screen edit module.

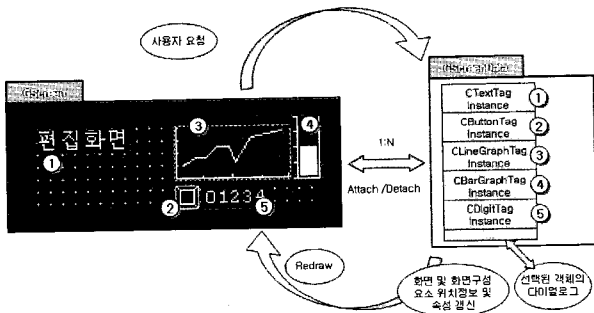


그림 7. 화면편집용 객체들의 동작 예  
Fig. 7. Operation example of screen edit objects.

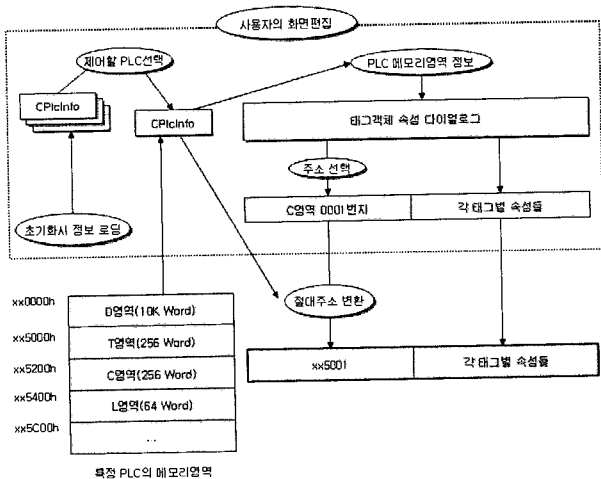


그림 8. PLC 정보 관리 객체의 동작  
Fig. 8. Operation of PLC information management object.

#### 나. 데이터관리 구현

그림 8은 PLC 정보관리를 담당하는 클래스가 저장하는 정보 및 어떻게 사용되는지를 나타내고 있는데, 그림 8에서와 같이 각 종류별로 다르게 정의되어 있는 PLC의 정보들은 프로그램 초기화 시에 파일로딩을 통해서 얻어지고, 화면편집을 수행하면서 태그의 메모리 주소를 입력할 때 사용되어지고, 프로젝트가 저장될 때 참조되는 것을 알 수 있다.

그림 9는 프로젝트의 저장형태를 나타내고 있다. 프로젝트는 HMI 장치의 메모리에 적재되기 위해서 로컬에서 먼저 파일형태로 생성되고 통신관리자인

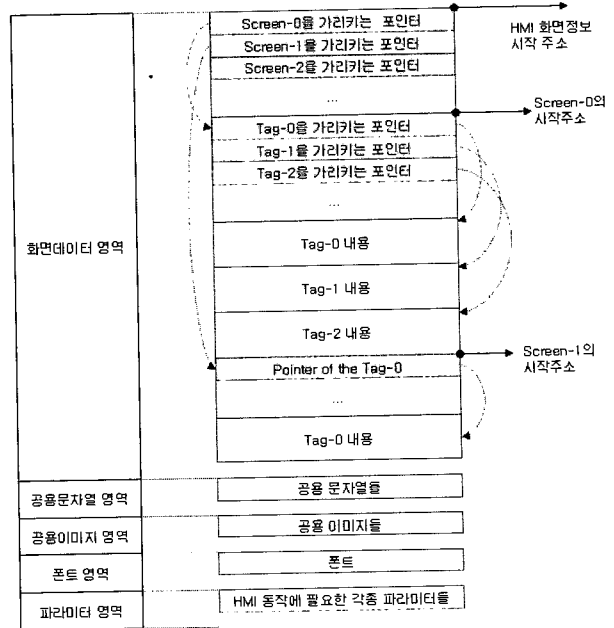


그림 9. 편집된 HMI 화면 데이터의 저장형태  
Fig. 9. Layout of edited HMI screen data.

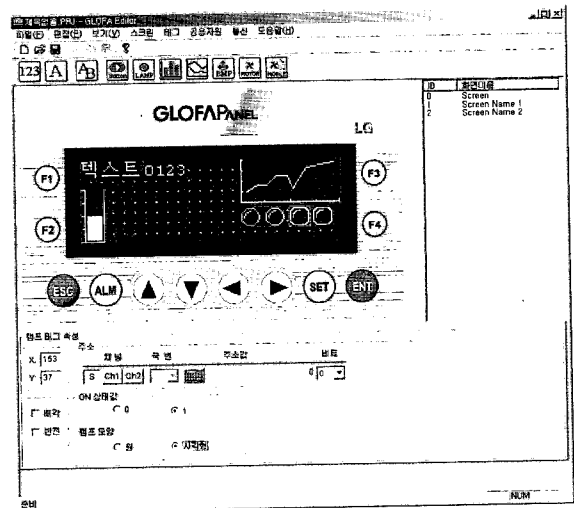


그림 10. 소형 HMI 장치용 화면편집 프로그램 실행 결과  
Fig. 10. Execution of screen editor for mini HMI equipment.

CCommThread객체를 사용하여 HMI로 전송하게 된다. 그림 10은 최종 구현된 소형 HMI 장치용 화면편집 프로그램의 결과를 나타내고 있다.

#### IV. 임베디드 HMI 시스템 프로토타입

본 장에서는 III장의 HMI 장치에 필요한 화면편집 프로그램의 설계 및 구현을 바탕으로 임베디드 HMI 장치용 프로그램과 화면편집 프로그램으로 구성된 HMI 시스템의 프로토타입을 제안한다. HMI 장치 하드웨어의 경우 PDA와 비슷한 하드웨어 사양을 갖춘 임베디드 시스템으로 가정하였다.

##### 1. 임베디드 HMI 장치 운영체제 및 개발환경

HMI 화면편집 프로그램의 개발환경은 제3장에서 설명한 환경과 동일하고, 고급사양을 가지는 임베디드 HMI 장치를 위해서 임베디드 운영체제를 기반으로 한 개발환경을 사용하여 PC와 임베디드 HMI 장치의 개발환경을 동일하게 구성하였으며, 개발환경은 그림 11에 나타난다.

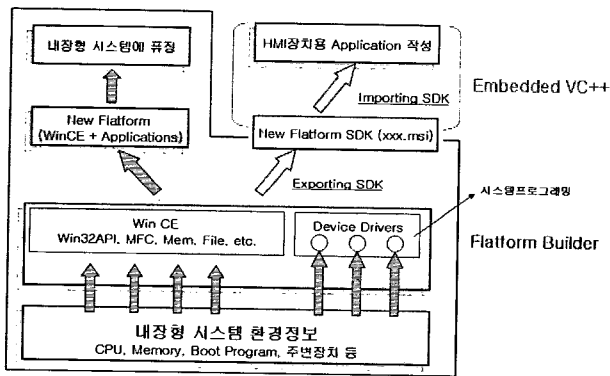


그림 11. 임베디드 HMI 장치용 프로그램 개발환경  
Fig. 11. Development platform for embedded HMI application.

##### 2. 임베디드 HMI 화면편집 프로그램

제III장에서 이미 구현된 화면편집 프로그램의 경우 화면에 태그 객체를 생성 및 배치하고 그 속성을 변경시켜 주는 단순한 기능에 불과했으나, 새로운 HMI 화면 편집 프로그램은 그보다 복잡하고 다양한 기능이 요구되기 때문에 그에 맞는 클래스 설계가 필요하다.

클래스는 화면 및 화면 구성요소를 나타내는 클래스(CDrawObj & CDrawTool Class)와 데이터 관리를 위한 클래스로 나뉜다.

화면부분을 나타내는 화면 객체는 제3장의 뷰 객체

를 활용하고, 화면의 데이터를 저장하는 객체는 제3장의 도큐먼트 객체를 활용한다. CDrawObj는 화면 구성요소인 태그 객체와 그림객체가 구현해야 할 공통의 인터페이스를 정의한 클래스이고, CDrawTool은 각 객체들을 선택, 크기조절, 이동 등을 수행하는 별도의 컨트롤러 객체이다. 제III장에서 구현된 HMI 편집 프로그램의 경우는 컨트롤러의 역할을 화면 뷰 객체에서 수행하였으나 다중선택, 객체의 크기조절과 같은 다양한 편집 기능을 별도의 컨트롤러 클래스에서 수행하도록 하였다.

데이터 관리를 위한 클래스는 제III장에서 구현된 데이터 관리 모듈을 그대로 적용한다.

##### 3. PLC 제어 및 감시용 HMI 장치용 프로그램

제III장에서 설계 및 구현된 HMI 화면편집 프로그램은 HMI 장치의 특정 메모리에 적재되기 위해서 정의된 구조체 집합을 사용하여 파일로 저장하고 저장된 파일을 HMI 장치로 전송하여 적재시키는 방법을 사용하였다. 그러나 고급사양의 하드웨어와 임베디드 운영체제 기반의 HMI 장치는 운영체제에서 제공되는 서비스를 통하여 파일전송이 가능하기 때문에 별도의 파일 송수신을 위한 프로그램 모듈은 필요하지 않다. 또한 동일한 운영체제 기반으로 동일한 프로그램 작성을 위한 라이브러리(MFC)를 기반으로 하기 때문에 새로운 구조체 집합을 정의할 필요가 없다. 따라서 HMI 장치 프로그램의 주요 기능은 PC에서 전송받은 편집된 화면 데이터를 프로그램 실행 시에 메모리에 적재한 이후에 주기적인 PLC통신을 통해서 적재되어 있는 화면들의 화면 구성요소들의 값을 갱신시키는 것이다. 본 장에서는 기존의 펌웨어로 작성된 HMI 장치용 프로그램의 구조를 바탕으로 임베디드 HMI 장치용 프로그램의 프로

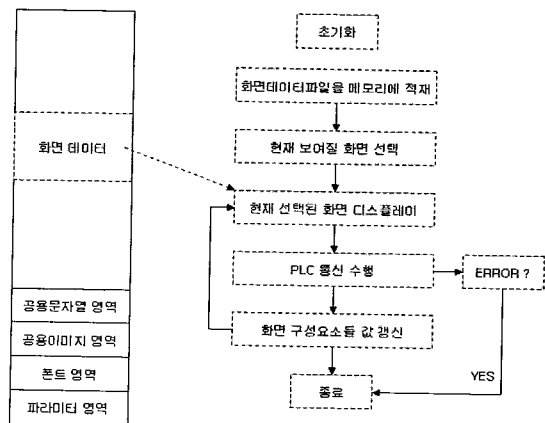


그림 12. 임베디드 HMI 장치용 프로그램 구조도  
Fig. 12. Program structure of embedded HMI equipment.



토타입을 제안하여 그림 12와 같은 구조로 동작하도록 하였다.

4. 임베디드 HMI용 프로그램 프로토타입 구현결과

그림 13에서 볼 수 있듯이 기존에 작성된 화면 구성 요소의 편집을 위해서 발전된 사용자 환경을 제공하도록 프로토타입을 설계하였으며, HMI 장치와 HMI 화면 편집 프로그램이 동작하는 운영체제 환경이 동일하기 때문에 그림 13에서 편집되는 TEXT 및 DIGIT이라는 객체의 클래스는 그대로 HMI 장치용 프로그램의 작성에 활용할 수 있다.

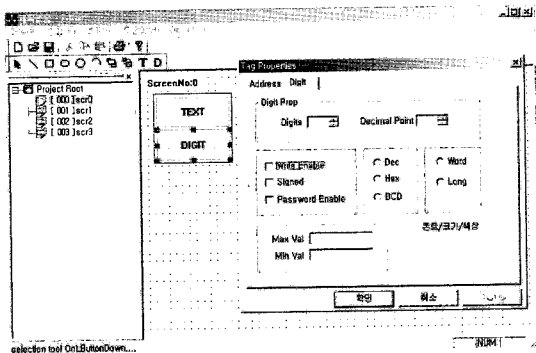


그림 13. 화면편집 도구 프로토타입의 실행결과  
Fig. 13. Execution of screen editor prototype.

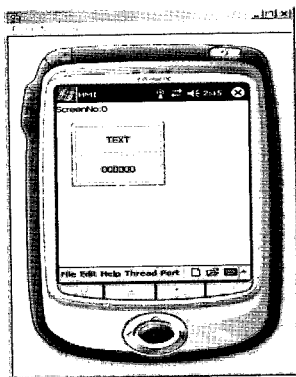


그림 14. HMI 장치용 프로그램 프로토타입 실행 결과  
Fig. 14. Execution of HMI equipment prototype.

본 논문에서 제안하는 임베디드 HMI 장치의 환경은 현재 널리 사용되고 있는 PDA와 비슷한 하드웨어 사양을 가지고 있기 때문에 HMI 장치용 프로그램의 테스트를 PDA를 통해서 수행하여 그림 14과 같은 실행결과를 볼 수 있었다.

V. 결론 및 향후 연구

본 논문은 소형 HMI시스템용 화면편집 프로그램의

구현을 기반으로 임베디드 HMI 시스템의 구현을 위한 프로토타입을 제안하였다. 소형 HMI 장치 하드웨어는 매우 제한적인 시스템 자원을 가지고 있기 때문에 화면 편집 프로그램 역시 제한된 기능만을 제공하도록 설계되었으나 HMI 시스템이 자동화 장비를 제어 및 감시하는 방법과 그에 필요한 프로그램의 구조를 파악할 수 있었으며 이를 기반으로 발전된 형태의 HMI 시스템용 프로그램의 구조 및 클래스 설계를 제안하였다.

HMI 화면편집 프로그램의 경우 다양한 HMI 장치에서 해당 프로그램이 활용될 수 있도록 하기 위해서는 제조사별로 다양하게 존재하는 PLC들과의 통신프로토콜 및 PLC장치에 의존적인 각종 설정 정보에 대하여 공통적인 인터페이스를 설계하고 DLL(Dynamic Linking Library)이나 COM(Component Object Model)을 활용하여 프로그램의 실행시간에 동적으로 적용이 가능하도록 추가적인 설계가 되어야 한다.

향후 화면에 배치되는 화면 구성요소 객체들의 추가가 용이하도록 단순한 클래스 정의가 필요하며, 객체의 인터페이스를 COM 모델을 통해서 정의하고 이를 런타임에 사용하는 방법 등 효과적인 프로그램 설계 방법론을 적용하여 성능을 개선해야 한다.

참고 문헌

- [1] 김남희, "산업용 HMI/SCADA 제품과 적용사례", 계장기술(www.procon.co.kr), 2002.
- [2] 박원일, "비호환 기종 PLC들의 통합제어를 위한 PLC 매니저 설계", 한양대학교 석사학위논문, 1993.
- [3] 박지환, "HMI 시스템의 기술동향", 계장기술(www.procon.co.kr), 1998.
- [4] 임기욱, 김홍남, "임베디드 소프트웨어분야의 기술발전 동향", 한국정보처리학회지 11권 6호, pp.25~32, 2004.
- [5] 조항신, "미래 지향의 자동화 솔루션", 계장기술(www.procon.co.kr), 2005.
- [6] 한광록, 문세호, "반도체 전공정 제어용 온도 제어기 설계 및 구현", 한국과학재단 연구보고서, 2000.
- [7] 한광록, 황미자, "모니터링 시스템을 위한 GUI 설계 및 구현", 한국정보처리학회 춘계학술발표논문집 제4권 1호, pp1177~1180, 1997.
- [8] 한광록, 황미자, "Wet Station 모니터링을 위한 제어모듈의 설계 및 구현" 한국정보처리학회 논문집 5권 7호, pp.1813-1828, 1998.
- [9] LG산전교육부, "LG programmable logic controller 프로그래밍 Master K 시리즈", LG산전, 2002.
- [10] Herbert Schildt, "MFC Programming from the

Ground Up”, McGraw-Hill, pp. 48~88, 1998.

[11] Jeffrey Richter, “Advanced Windows 3th Edition”, Microsoft Press, pp.320~363, 1997.

[12] John M. Vlissides, Mark A. Linton, “Applying Object-Oriented Design to Structured Graphics”, USENIX C++ Conference, Denver, Colorado, pp.81~94, 1998.

[13] Mahemoff M, Johnston, “Handling Multiple Momain Objects with Model-View-Controller” Technology of Object-Oriented Languages and Systems 32, Melbourne, Australia, p28-39. 1999.

[14] Robert Lafore, “Object-Oriented Programming In C++”, Waite Group Press, pp.73~88, 1995.

저 자 소 개



**선 복 근**(정회원)  
 1999년 호서대학교 컴퓨터공학과 졸업(학사)  
 2001년 호서대학교 벤처전문대학원 컴퓨터응용기술학과 졸업(공학석사)  
 2003년~현재 호서대학교 반도체 제조장비국산화연구센터 전임연구원  
 2005년 현재 호서대학교 대학원 컴퓨터공학과 박사과정 재학  
 <주관심분야: 정보검색, 에이전트시스템, 멀티미디어 동기화, HCI>



**한 광 록**(중신회원)  
 1984년 인하대학교 전자공학과 졸업(공학사)  
 1986년 인하대학교 대학원 정보공학 전공(공학석사)  
 1989년 인하대학교 대학원 정보공학 전공(공학박사)  
 1989년~1991년 한국체육과학원 선임연구원  
 1991년~현재 호서대학교 컴퓨터공학부 교수  
 2001년~2002년 ISI University of South California 방문연구원  
 <주관심분야: 멀티미디어, 정보검색, 자연어처리, 기계번역, HCI, 지능형에이전트>



**임 기 욱**(중신회원)  
 1977년 인하대학교 공과대학 전자공학과 졸업  
 1987년 한양대학교 전자계산학 석사  
 1994년 인하대학교 전자계산학 박사  
 1977년~1988년 한국전자통신연구소 시스템 소프트웨어 연구실장  
 1988년~1989년 미 캘리포니아 주립대학(Irvine) 방문연구원  
 1989년~1996년 한국전자통신연구원 시스템연구부장, 주전산기(타이컴) III, IV개발 사업 책임자  
 1997년~1999년 정보통신연구진흥원 정보기술 전문위원  
 2001년~2003년2월 한국전자통신연구원 컴퓨터소프트웨어 연구소장  
 2000년~현재 선문대학교 지식정보산업공학과 교수, ITRC(임베디드 S/W 개발환경 연구센터) 소장  
 <주관심분야: 실시간데이터베이스시스템, 운영체제, 시스템구조>