

임베디드 리눅스 기반 산업용 무선 HMI 소프트웨어 모듈 설계 및 구현

The Design and Implementation of Embedded Linux-Based Industrial Wireless HMI Software Module

최속영* · 문승진**

Suk-Young Choi* and Seung-Jin Moon**

수원대학교 컴퓨터학과

요 약

산업용 HMI(Human Machine Interface) 시스템은 공장 자동화의 주요 구성요소 중 하나로서 PLC와 연결되어 자동화 설비 또는 장치의 운전 상태를 감시하고 제어하는데 사용된다. 이러한 HMI는 주로 제조업체별로 특정한 시스템을 사용하고, 근거리에서 위치하여 쓰이기 때문에 시스템 개발 시 많은 부하를 주게 되고 시스템 확장이 어려운 단점이 있다. 이에 본 논문에서는 오픈 소스인 임베디드 리눅스 기반에 멀티 플랫폼을 지원하는 Qt/Embedded와 무선 통신 모듈을 사용하여 터치 패널형 산업용 HMI 소프트웨어 모듈을 설계 및 구현하였다. 이 모듈은 Qt가 지원되는 시스템이면 소스 수정 없이 사용할 수 있으며 무선 랜 모듈을 이용하여 시스템의 이동성 및 네트워크 구축 및 시스템 확장을 보다 유동성 있게 설계할 수 있다. 이에, 리눅스 기반의 무선통신이 가능한 HMI 소프트웨어 모듈 구현으로 이동성 확보 및 범용 운영체제의 사용으로 인한 시스템 개발 시 부하 감소와 가격 경쟁력의 향상을 이루게 되었다.

키워드 : 휴먼 인터페이스, 무선 휴먼 인터페이스, 공장 자동화

Abstract

Industrial HMI(Human Machine Interface) system is the main element among the factory automation processes and have been used to monitor and control operation and status of machine in factory with PLC. This HMI often brings heavy loads to the system development and difficult decreasing the system because it tends to use a specific system per each manufacturer. Therefore, in this thesis, we have developed an embedded linux-based embedded industrial HMI software modules which can be used for touch panel embedded system to solve these problem. In this module, we have used the Qt/Embedded software component because it can be used by all systems which support C++ compiler without modifying the existing codes. We can design more flexible system and network configuration because we have used the wireless communication module. In this thesis, we implement linux-based HMI software modules which are capable of wireless communication as well as bringing the mobility to the overall system and finally decreasing the system development loads by using the general purpose OS with competitive price.

Key Words : Human Machine Interface, Wireless Human Machine Interface, Factory Automation

1. 서 론

산업용 HMI(Human Machine Interface) 시스템은 산업 자동화의 발전으로 등장하여 공장 자동화 시스템의 구성 요소로서, 프로그래밍이 가능한 제어 시스템인 PLC(Programmable Logic Controller)와 연결되어 공장 기기를 모니터링하고 제어하기 위해 사용되고 있다.[1]

터치패널 임베디드 HMI는 제조업체별 전용 하드웨어와 소프트웨어를 사용하기 때문에 시스템의 성능을 최대한 이끌어내는 장점이 있으나 다른 시스템으로의 이식과 프로그램의 유지, 보수와 인터페이스의 확장과 의사 결정 시스템이나 전사적 자원관리 시스템 같은 대규모 관리 시스템의 일부분으

로 사용될 필요성이 생겨나게 되었으나 전용 시스템으로는 이러한 요구를 충족하기 어렵다. 또한, 시스템이 복잡성을 요구함에 따라 PLC에 한 대의 마스터 컨트롤러와 여러 대의 슬레이브 컨트롤러를 연결 하게 되었다. 이러한 시스템은 초기 기기들의 정해진 위치에서 공정을 실행 하게 되는데, 이는 유동성이 있는 기기의 모니터링 및 제어를 불가능하게 하였다.

이에 본 논문에서는 오픈 소스 기반의 임베디드 운영체제를 이식하여 트롤테크사의 크로스 플랫폼 툴킷인 Qt(cute)/Embedded(이하 Qt)와 무선통신 모듈을 이용하여 기존 시스템의 단점들을 해결하였다.

본 논문에서 설계한 리눅스 기반의 무선통신 HMI 소프트웨어 모듈은 공장자동화 및 기타 제어대상 기기의 제어를 사용자가 손쉽게 구성할 수 있고, 무선통신모듈을 이용하여 원 거리에 위치한 slave PLC의 제어 및 모니터링이 가능하게 되었다.

접수일자 : 2006년 6월 26일

완료일자 : 2007년 3월 26일

2. 관련연구

2.1 PLC와 HMI

PLC(Programmable Logic Controller)는 On/Off 제어만으로 수행되는 일종의 소형 컴퓨터로 각종 센서들로부터 입력을 받아 저장된 프로그램에 의해 계산하고 기계나 프로세스들을 제어하는 공장 자동화에 필수적으로 사용되는 핵심장치이다. 이러한 PLC는 제조사와 기종이 독점적이고 폐쇄적 이어서 하드웨어와 소프트웨어 활용 기술이 서로 다르기 때문에 각각을 별도로 학습해야 하는 단점이 있다.[2]

HMI(Human Machine Interface)는 광범위하게 인간의 모든 감각기관을 이용한 정보의 표출과 외부 정보에 대한 인지 행위를 통해 기계와 통신하는 모든 인터페이스를 칭한다.[3,4] 공장자동화 분야에서 HMI는 주로 HMI 소프트웨어를 지칭하지만 본 논문에서의 HMI는 터치패널 임베디드 HMI를 가리키며, 그래픽 화면을 통한 PLC와 연결된 기기나 공정 상태를 모니터링하고 터치 동작으로 제어를 한다.

2.2 임베디드 리눅스

임베디드 리눅스 운영체제는 특정 어플리케이션에 맞도록 리눅스 커널을 최적화 한 것이다. 임베디드 리눅스의 커널은 오픈 소스이기 때문에 소스 코드 자체가 공개되어 있고 이의 적용에 따른 로열티 자체가 없으며, 리눅스 커널 자체가 태생적으로 모듈 형식의 코드로 이루어져 있어 손쉽게 최소, 최적화가 가능하다. 또한, 네트워킹 어플리케이션에 필수인 TCP/IP, PPP, X.25, HDLC, FDDI 등의 네트워크 코드를 가지며, 현존하는 거의 대부분의 프로세스에 디바이스가 포팅되어 있으며, 실시간(real time)을 지원하여 공장 자동화 기기에는 필수적인 요소이다. 이러한 이유로 임베디드 리눅스는 임베디드 시스템에서 동작하는 리눅스 커널로서 임베디드 시장에서 주목받고 있다.

2.3 무선 랜

무선 랜은 유선 망 없이 무선주파수를 이용하여 데이터를 주고받을 수 있으며, 전파를 송수신하는 AP(Access Point)를 사용하여 반경 내에 있는 무선 단말기들을 무선으로 연계시키고, 단말기를 유선망과 연결시킬 수 있다. 따라서 무선 랜은 회선이 필요 없기 때문에 네트워크의 설치나 재배치 및 유지보수가 간편하고 빠르게 이루어질 수 있다. 또한 임시 네트워크 구축 및 케이블 설치가 어려운 곳에서의 구축이 용이하여 이동성, 네트워크 재구성의 용이성, 비용 절감효과로 인해 사용영역이 점차 확대되고 있다.

3. HMI 소프트웨어 설계

본 논문에서는 기존의 윈도우 CE기반의 HMI 모듈 기술과 무선 통신 기술을 추가하여 리눅스 기반의 HMI 무선통신 시스템을 설계하였다. HMI 소프트웨어 모듈은 TOP 디자이너를 이용하여 slave PLC의 종류, 모니터링 조건, 삽입된 태그, 각종 도형이나 그림 정보 등을 포함시킨다. 이렇게 추가된 HMI 소프트웨어 모듈을 제어 노드에 탑재 시킨 후 무선 랜을 통하여 데이터 변환기에서 무선 신호를 변조하여 slave PLC에서 실제 공정에 사용할 수 있도록 한다.

3.1 시스템의 구성

임베디드 리눅스 기반의 Qt를 이용하여 사용자 정의에 맞게 설계된 무선 HMI 시스템 모듈 시나리오를 다음과 같이 구성하였다. 시스템 시나리오의 전체적인 흐름은 그림 1과 같은 방법으로 실행된다.

- ① TOP 디자이너를 이용하여 slave PLC에 맞도록 모니터링 및 제어 환경 디자인을 한다.
- ② 오브젝트 파일의 환경 변수를 Qt와 연결하여 모니터링 및 제어할 수 있도록 보정한다.
- ③ HMI 시스템 모듈 모니터링 또는 제어신호를 무선통신 패킷으로 생성하여 전송한다.
 - Packet Header
 - HMI Information
 - Control PLC Info
 - Monitoring PLC Info
 - Control_Commend_unit
 - Monitoring_Commend_unit
 - Packet Information
 - Checksum
- ④ master PLC는 HMI 모듈로부터 수신된 패킷을 참조하여 해당 slave PLC로 무선 패킷을 송신한다.
- ⑤ slave PLC는 master PLC로부터 수신된 패킷을 데이터로 변환한다.
- ⑥ 변환된 데이터를 참조하여 실제 공정 제어실행 및 모니터링을 수행한다.
- ⑦ 모니터링 정보를 ③항의 무선통신 패킷으로 생성하여 master PLC로 전송한다.
- ⑧ master PLC는 수신된 모니터링 패킷을 HMI 모듈로 송신한다.
- ⑨ HMI 모듈 viewer는 TOP 디자이너의 형식에 맞게 환경변수와 연결, 보정한다.

무선 HMI 시스템은 TOP 디자이너로부터 분석된 각종 태그들과 오브젝트들을 매칭시키는 모듈과, HMI 기기 모니터링 및 제어 모듈, slave PLC로 무선 데이터를 송신하기 위한 패킷생성과 분석 모듈 그리고 실제 기계제어 모듈로 구성되어 있다. 이것은 하나의 임베디드 시스템에서의 시스템 부하와 추후 유지 보수를 위하여 모듈별로 구성하였다.

3.2 시스템의 설계

HMI 모듈은 본 논문에서 제안하는 가장 중요한 부분으로 크게 세 가지의 기능을 가진다. 첫째, M2I 코퍼레이션의 터치패널제품 화면구성 작화 프로그램인 TOP-Designer에서 만들어진 작화 파일을 사용하여 GUI화면을 구성하고 slave PLC로부터 받은 데이터에 따라 화면을 다시 그려 모니터링 상황을 표시 하고 둘째, 사용자의 터치를 처리하고 셋째, 무선 랜 통신처리를 위한 통신 모듈을 통해 주고 받는 데이터를 처리하며 세 번째 기능은 무한 반복되기 때문에 쓰레드로 구성되어 있다.

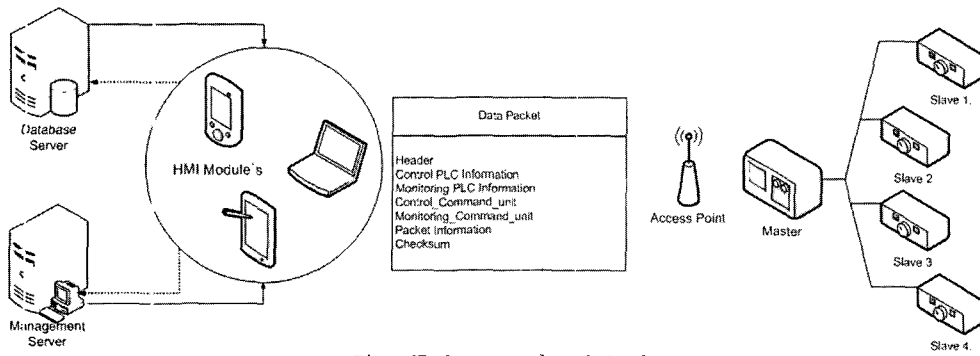


그림 1 무선 HMI 시스템 구성도
Fig 1. wireless HMI system design

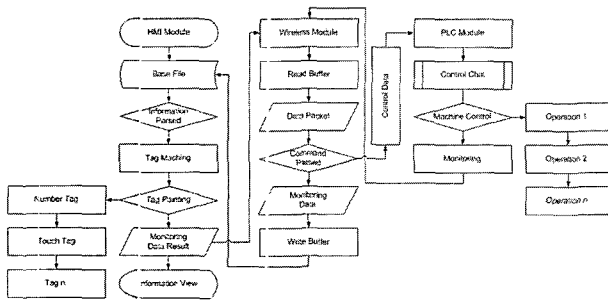


그림 2. 실행 제어 모듈
Fig 2. flow control module

HMI 시스템에서 화면에 표시하여 제어하거나 모니터링하기 위한 표현 방법으로 태그(tag)를 이용한다. 태그는 slave PLC 또는 HMI 본체의 데이터를 조건 또는 시간주기, 실시간에 의해 HMI 시스템 화면에 나타내거나 데이터를 외부 시스템으로 전달할 수 있도록 하는 매개 역할을 한다. 본 논문에서 사용된 태그는 숫자, 터치, 램프, 키 표시, 경고 태그를 구현 및 설계 하였다[5]. HMI 시스템은 작화 파일을 분석하여 첫 번째 윈도우의 화면을 표현 하면서 시작하게 되며, 이는 BaseFile 구조체안이 시스템 시작에 필요한 정보들을

표 1 basefile 구조체
Table 1. basefile structure

```

struct BaseFile {
char cColor;
char cScrGnd;
char *desc;
long tag_ptr;
long draw_ptr;
long device_ptr;
short tag_nura;
short draw_nam;
short device_num;
short sLampSum;
short sTouchSum;
short sAlarmSum;
short sLeyviewSum;
long lTouchPtr;
long lAlarmPtr;
long lKeyviewPtr;
...
}
    
```

이용한다.

표 1에서와 같이 BaseFile 구조체에는 윈도우의 색상과 관련된 정보와 이미지 데이터 포인터, 디바이스, 상수 테이블의 포인터 정보들과 함께 각 태그들과 오브젝트들의 포인터와 개수 정보가 저장된다. 이러한 태그들의 정보와 오브젝트들은 디바이스 입력, 글자 또는 오브젝트(선, 원, 사각, 타원) 등록 시 배경 색, 선 색, 채움 색 지정과 같은 공통적으로 사용되는 기능이 있고 각 태그별 특성에 따라 설정된다.

표 2 각 태그의 공통 사용기능
Table 2. common function of tag

기능	설명
설명	태그에 대한 간단한 설명 영문자 기준, 최대 20자 입력.
디바이스	참조, 제어하는 디바이스 주소 워드단위, 비트단위 선택
시스템 버퍼	시스템 버퍼입력 시스템의 내부 영역 0~1023까지의 워드공간
상수	피연산자로만 사용 상수의 크기와 타입에 따라 범위가 다름
폰트	한글 : 16 * 16, 영문 : 8 * 16 한글 : 32 * 32, 영문 : 16 * 32
배각	표시 숫자의 크기표현 가로, 세로 각각 1~8 배각까지 설정
글자색, 배경색	글자는 글자색으로, 배경은 배경색으로 표시
정렬	지정된 전체 자릿수의 데이터 표시방법 왼쪽, 오른쪽, '0' 채움 방법.

HMI와 slave PLC의 통신 방법은 무선 패킷 통신 방법으로, 이 방법은 패킷을 생성하여 단순 브로드 캐스팅하여 송·수신하게 된다. 이에 패킷을 생성해야 하는데 패킷내의 내용으로는 그림 3과 같다.

이렇게 생성된 패킷은 하나의 master PLC로 데이터를 송신하고 master PLC는 수신된 패킷의 데이터들 중에서 "Control PLC Info"의 정보를 참조하여 해당 slave PLC로 모든 패킷을 전송하게 된다. 이렇게 하나의 master PLC와 여러 대의 slave PLC로 나누어 처리하는 것은 공정 모니터링과 공정제어에 사용되는 slave PLC는 각기 다른 공정의 기기에 연결되어 있기 때문이며, master PLC는 다시 여러 대의 HMI 모듈로부터 제어데이터를 수신하거나, slave PLC로 부터의 모니터링 정보를 해당 HMI 모듈로 송신하여 일련의 게이트웨이역할을 함으로 다중의 HMI 모듈과 slave

PLC의 데이터 충돌을 방지할 수 있다.

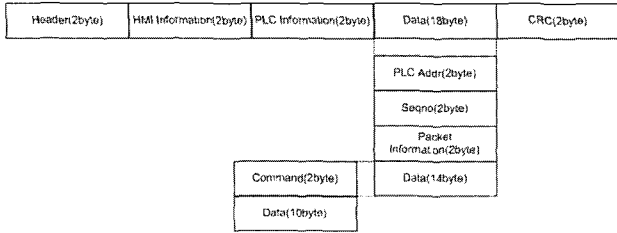


그림 3. 무선 HMI와 PLC 데이터 구조
Fig 3. wireless HMI and PLC data structure

4. HMI 소프트웨어 구현

본 논문에서 구현한 무선 HMI 소프트웨어는 리눅스 기반의 플랫폼에서 Top-Designer 작화파일의 HMI 구성정보를 자유롭게 이용할 수 있도록 Qt를 이용하여 구현하였다. 이와 함께 기존의 시리얼 통신만으로 slave PLC를 제어하던 것과는 달리 무선네트워크 기능을 추가하여 장소의 제약이 없이 원거리에서도 하나의 HMI로 여러 대의 slave PLC의 상태를 모니터링 하거나 제어할 수 있다.

4.1 태그의 구현

4.1.1 숫자 태그

숫자 태그는 외부 slave PLC의 특정 주소의 값을 화면에 실시간으로 표시하며, 숫자 태그가 위치할 영역 정보, 데이터가 표현될 형태의 대한 정보들, 숫자들이 표현되기 위해 참조할 외부 디바이스 기기 주소 정보, 각종 색상 정보와 크기 정보들로 구성되어 있다.



그림 4. 숫자 태그
Fig 4. number tag

4.1.2 터치 태그

터치 태그는 화면에 터치영역을 설정하고, 그 터치입력에 의해 비트나 워드의 조작을 실행하는 태그이며, 사용자가 설정한 조건들에 대한 동작이 이루어져야 한다. HMI 시스템에서 모든 기기의 제어나 비트의 조작 등 행동이 수반되는 거의 모든 조작에는 터치가 관련되어 있을 정도로 중요하다.



그림 5. 터치 태그
Fig 5. touch tag

터치 태그는 비트를 조작할 경우에 터치할 때 On 스위치, Off 스위치, 반전스위치로 사용할 수 있으며 워드 데이터를 저장 디바이스에 쓰기를 수행하는 스위치로도 사용할 수 있고, 키 기능은 Ten Key의 숫자 키 또는 문자열을 입력하기 위한 문자키 및 특수기능 키로도 사용할 수 있다.

4.1.3 램프 태그

램프 태그는 원이나 사각으로 둘러싸인 영역을 설정된 조건에 따라 변경하여 사용자에게 시각적인 효과를 준다. 버퍼에서 램프 태그와 관련된 외부 기기의 주소와 관련된 내용을 읽어 들어 이루어지며 크게 비트단위 동작과 워드단위 동작으로 구분된다.



그림 6. 램프 태그
Fig 6. lamp tag

비트단위의 동작은 표시 주소의 비트 값에 따라 디스플레이에 설정에서 지정한 On, Off 시의 색으로 나타나며, 워드단위 동작은 설정한 주소 값이 범위 설정에서 설정한 범위에 해당할 때, 각 해당 범위에서 설정한 색으로 나타난다.

4.1.4 키 표시 태그

키 표시 태그는 평상시 숫자 태그처럼 동작하며 키 표시 선택조건에 의해 키를 표시하는 패드로 쓰인다. 키 표시 태그를 사용하기 위해서는 태그를 터치하여 활성화 하고 이때 태그의 영역이 반전되어 표시된다. 또한, 키 표시 태그는 단독으로 쓰이지 않고 그림과 같이 키 표시 태그에 입력을 할 수 있는 터치 태그와 함께 사용된다.

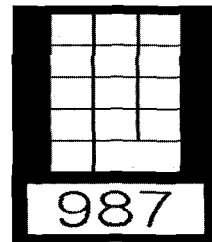


그림 7. 키 표시 태그
Fig 7. keyInfo tag

4.1.5 경보 태그

경보 태그는 사용자가 지정한 조건에 따라 경보가 발생했을 때 지정된 메시지를 일정 영역에 표시한다. 경보 태그는 다른 태그들과는 달리 별도로 태그등록 파일로 작성하여야 하며, 별도로 저장되어 있는 경보 조건과 정보들을 읽어 내어 사용하며 경보 태그에서 지정된 외부 디바이스 주소의 값들을 감시하여 이들 조건과 일치하게 되면 경보를 출력한다.

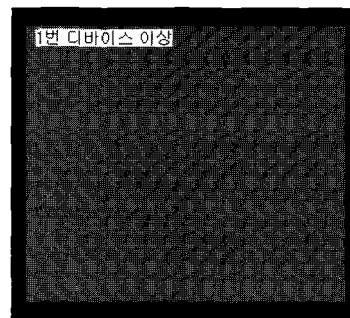


그림 8. 경보 태그
Fig 8. alarm tag

이러한 태그들은 각 태그정보를 저장하기 위한 태그 별 구조체가 존재하고, BaseFile 구조체에서 사용하려는 태그의 작화 파일 내 태그정보 저장 위치를 참조하여, 그 위치부터 해당되는 태그의 구조체에 저장을 한 후 모듈에서 사용한다.

4.2 터치 패드의 구현

HMI 사용자의 터치 동작은 모든 제어와 관련이 있다. 이에 터치 동작이 일어나게 되면 우선 터치 영역내의 동작인지 검사한 후 터치 영역내의 터치이면 그것이 비트 터치, 터치 태그, 워드 터치 태그, 기 표시 태그인지에 따라 처리를 하고 키 표시 태그인 경우 데이터를 입력하는 도중이나 clear키를 눌렀는지 또는 enter키를 눌렀는지에 따라 처리를 다르게 하게 된다.

표 3 터치 명령

Table 3. touch command

```
r.setCoords(Touch Pad x, y coordinates values);
if(r.contains(me->pos())){
switch(oper[0]){
case ' ':
Write_Tag=e_TouchBit;
break;
case '1':
Write_Tag=e_TouchWord;
break;
case '2':
if(oper_tmp==0xF1){
touch command location
}
else {
if(oper_tmp!=0x2F && oper_tmp!=0xF0){
touch operation
}
if(oper_tmp==0xF0){
touch Enter / Clear
}
}
break;
}
}
```

4.3 PLC 시스템 구현

slave PLC 구현 부에서 실행되는 모듈은 무선모듈을 통해 HMI 모듈로부터 모니터링 정보를 요구 받으면 각, slave PLC로부터 master PLC로 모니터링 패킷을 생성하여 무선으로 HMI로 전송하게 된다. 또한, HMI로부터 특정 PLC로 제어 정보를 보낸다면 HMI의 제어 정보를 패킷화 하여 master PLC로 보내게 되고 다시 master PLC는 패킷을 참조하여 해당 제어 slave PLC로 송신하여 기기 제어를 한다. 이에 하나의 패킷으로 모니터링정보 및 제어 정보를 판별하기 위하여 패킷의 마지막에 구분 정보를 넣어 "1"이면 제어, "0"이면 모니터링 정보 요구로 받을 수 있도록 하였다.

표 4 데이터 변환

Table 4. data converting

```
while(1) {
recv(socket_fc,tmp,9,0);
write(serial_fc,tmp,8);
usleep(5000);
if(tmp[8]read(serial_fd,tmp,8); //제어적용
else { //모니터링
read(serial_fd,tmp,8);
send(socket_fc,tmp,8,0);
}
}
```

5. 실험 및 결과분석

임베디드 리눅스 기반의 Qt를 이용한 HMI 시스템 모듈 알고리즘의 유용성을 검증하기 위하여 TOP 디자이너로 디자인한 작화파일을 이용하였다. 기존 Windows CE 기반의 시스템은 시리얼(RS-232C)로 HMI 모듈에 하나의 PLC가 1:1로 연결되어 있으나, 본 논문에서 제안한 무선 HMI 시스템은 HMI 모듈에 하나의 master PLC가 여러대의 slave PLC로 무선으로 데이터를 주고받을 수 있도록 구성되어 있다. 이에 그림 9의 공통 작화파일로 HMI 시스템과 slave PLC의 제어 및 모니터링 데이터를 송수신하는 시간(%)과 그래픽 처리 시간, HMI 시스템의 이동성 확보를 위한 원격 리 이동 테스트, 다중 slave PLC의 제어와 모니터링 등을 실험 하였다.

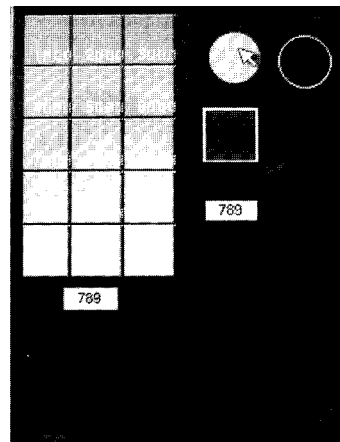


그림 9. HMI 시스템 작화파일
Fig 9. HMI system draw file

그림 10과 그림 11은 위에서 설명한 조건으로 실험 하고 제안하는 시스템은 HMI와 slave PLC는 무선으로 3m거리에서 하나의 태그를 30회씩 실험하여 평균값을 이용하였다.

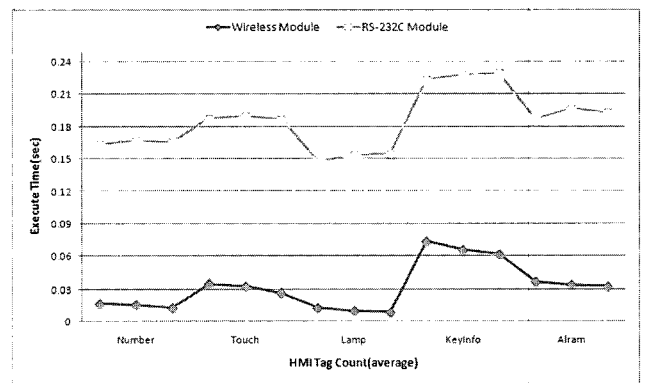


그림 10. HMI 데이터 처리 시간
Fig 10. HMI data processing time

그림 10과 같이 각 태그의 실행 시간은 차이를 보였다. 태그별 실행 시간차는 제안하는 wireless module과 RS-232C module이 서로 비슷한 태그에서 시간차이가 있으나 제안하는 모듈과 기존 모듈은 약 0.10초 정도 차이를 보이고 있다. 이는 기존 모듈의 무거운 알고리즘 수행과 embedded VC++

의 라이브러리 참조에 문제가 있음을 반증하며, 제안한 모듈에서 약 0.10초의 빠른 실행 속도는 기존의 알고리즘을 가볍게 수행할 수 있도록 수정하여 1회 수행에 필요한 리소스를 최대한 확보하여 수행하도록 되어 있음을 증명한다.

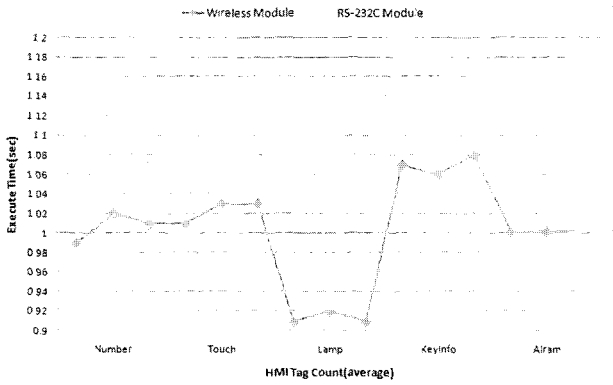


그림 11. HMI 이미지 처리 시간
Fig 11. HMI image processing time

그림 11의 기존 모듈과 제안하는 모듈의 화면처리 시간은 기존 모듈의 구조상 그래픽 처리와 데이터 처리를 동시에 번갈아 가면서 처리하므로 속도가 느리다. 하지만 제안하는 모듈은 세 번의 그래픽 처리(layer 3) 후 한 번에 데이터 처리를 하여 데이터 처리 실행 후에 그래픽 처리가 중복되지 않는다. 이에 기존의 모듈은 HMI 모듈에서 확인할 경우 화면 반짝임이 발생하게 되지만, 제안하는 모듈은 초기에 생성된 이미지 데이터를 3 layer로 분류하여 초기 세 번만 화면에 그리게 되므로 화면의 깜박거림 없이 데이터 처리를 원활하게 수행할 수 있다.

제안하는 HMI 시스템 모듈의 이동성 확보 실험은 그림 12와 그림 13과 같다. 기존의 시스템 모듈은 유선이며 통신 속도는 20kbps이고 최대 통신 거리는 15m이하 이므로 제안하는 시스템 모듈만 실험한다.

제안하는 시스템 모듈은 802.11b,g를 지원하며 최대 통신 속도는 2Mb에서 최대 54Mb이고 통신거리는 최대 수십 Km까지 가능하다. 따라서 HMI 모듈과 slave PLC와의 거리는 최소 50cm에서 최대 10m의 범위에서 실험한다.

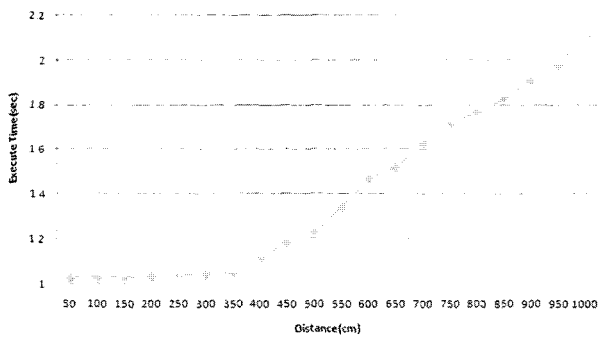


그림 12. 거리에 따른 실행 속도
Fig 12. distance for execute time

그림 12에서와 같이 50cm에서 350cm에서 평균 1.14초의 실행 속도를 유지 하였으나 400cm에서 느려지는데 이는

HMI와 slave PLC에 장애물로 인하여 데이터의 송수신 시간이 지연되는 문제로 측정된다. 이는 그림 13에서도 동일한 문제로 인하여 50cm에서 350cm의 이동구간에서는 큰 실행지연이 없으나 400cm의 이동구간에서 큰 실행지연이 측정된다.

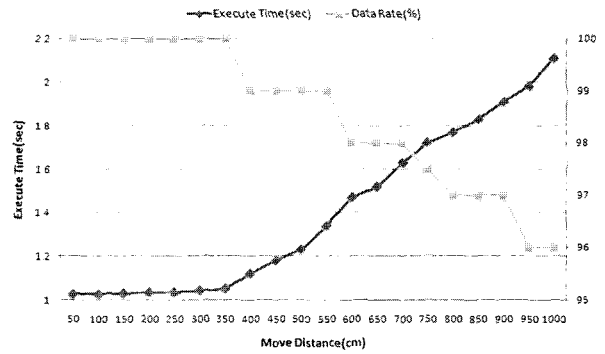


그림 13. HMI 이동에 따른 실행시간 및 데이터 전송률
Fig 13. execute time and data rate according to moving HMI

데이터 전송률 또한 50cm에서 300cm의 구간에서는 100%의 전송률을 보이며 300cm 이후의 구간에서는 최대 4%의 전송률 손실이 있다.

다중 slave PLC의 제어를 실험은 실행시간과 이미지 처리 시간의 평균을 데이터로 사용 하였다.

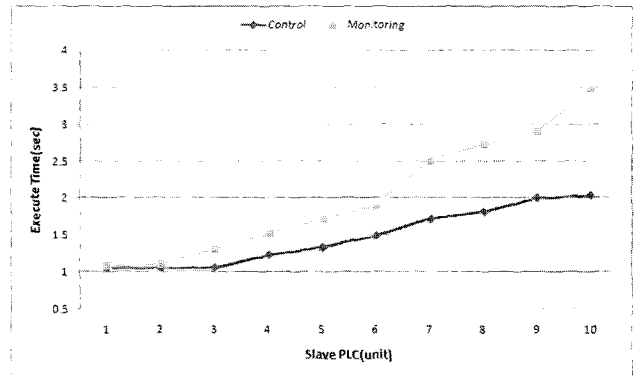


그림 14. slave PLC 증가에 따른 실행시간
Fig 14. multiple slave PLC for execute time

하나의 HMI 모듈에서 많은 slave PLC의 제어 및 모니터링은 그림 14와 같으며 slave PLC가 1~2개는 실행속도에 큰 차이를 보이지 않으나 3개부터 차이를 보인다. 이는 slave PLC는 데이터의 송수신을 하나의 패킷으로 처리하도록 설계되어 있는데 HMI 모듈에서 동시에 여러 slave PLC로 실행 패킷을 전송하게 되면 slave PLC는 패킷을 확인하여 처리해야 하기 때문에 실행지연이 있다.

6. 결론 및 향후 연구과제

본 논문에서는 제조업체별 전용의 시스템을 사용하는 HMI 시스템의 개발 시 발생하는 오버헤드 및 유지보수의

어려움을 줄이기 위해 임베디드 리눅스를 기반으로 하는 터치 패널형 산업용 무선 HMI 소프트웨어 모듈을 구현하였다. 본 논문에서 구현한 모듈은 산업용 HMI에 사용되는 다섯 가지 태그 모듈과(터치 태그, 램프 태그, 키 표시 태그, 경보 태그) 무선 통신 모듈(데이터 패킷 생성과 분할 모듈)을 이용하여 무선으로 원거리에 있는 slave PLC의 기본적인 모니터링 및 제어를 할 수 있다. 이 모듈은 리눅스라는 범용 운영 체제를 사용하였기 때문에 기능 추가 및 확장이 용이하고 멀티 플랫폼을 지원하는 Qt를 사용하여 구현하였기 때문에 기존의 소스 수정 없이 Qt가 지원되는 모든 시스템에서 사용할 수 있다. 또한 무선 통신 기능이 있기 때문에 무선 랜을 지원하는 장치 상에서 사용하면 이동하면서 모니터링 및 제어를 할 수 있다는 장점이 있다.

향후 본 논문에서 구현한 기본적인 태그를 이용한 정밀한 동작 설정과 더 다양한 태그를 구현할 필요가 있으며, 무선 랜을 사용하여 이동성이 지원되어 편리하지만 보안에 약하다는 문제는 앞으로 해결해야 할 것이다.

참 고 문 헌

[1] Siemens Automation and Drives, "Human machine interface -- how they will evolve", <http://manufacturingtalk.com>

[2] 디이시스 시스템 사업부, "PC 기반 제어 기술", 제어계측, 2001

[3] 김인수, "User Interface Design", <http://ui.korea.ac.kr>

[4] 장효석, 임성락, "반도체 웨이퍼 세정 장비 모니터링 시스템을 위한 기본 요소의 분석 및 설계", 한국정보처리학회 논문지, 2000

[5] (주)M2I 코퍼레이션 "TOP User's Manual", 2002

[6] 교봉석, "최신 MMI의 발전과 개요" 계장기술, 2004.7

[7] 이민석, "모바일 기기를 위한 임베디드 리눅스", 한국정보처리학회, 제9권 제1호, 2002

[8] 이순렬, "전사적 개념의 통합 HMI 솔루션", 21C F.A Vision, 2003.9

[9] 문영성, "비균일 트래픽 환경하에서 다단상호연결네트워킹의 해석적 성능 모델링 및 평가", 인터넷정보학회논문지, Vol. 4, No. 5, 2003.10

[10] 이성열, "A Case Study of On-Line PLC Laboratories Using Distance Learning", 산업공학 논문지, 18권 4호

[11] O. GomisBellmunt, "A Distance PLC Programming Course Employing a Remote Laboratory Based on a Flexible Manufacturing Cell", IEEE Trans on Education, Vol. 49 No. 02pp. 2006.5

[12] O. Moriwaki, "Nevel PLC-based optical correlator for multiple phase-modulated labels", IEEE PTL, Vol. 17No. 02pp. 2005.2

[13] Jong Moo Lee, "PLC platform for bidirectional transceiver with wide multimode output waveguide to receiver", IEEE PTL, VOL. 17 No. 01pp, 2005.1

[14] 선복근, 한광록, 임기욱, "PLC 모니터링을 위한 임베디드 HMI 시스템의 개발에 관한 연구", 전자공학회 논문지 CI 제42권 4호, 2005.7

[15] 홍상현, "차별화된 PLC 프로그램과 HMI 장비를 이용한 차체 라인의 운용에 관한 연구", 울산대학교 석사학위 논문, 2002.

저 자 소 개



최숙영(Choi Suk Young)

2002년 : 수원대학교 컴퓨터학 학사
 2005년 : 수원대학교 컴퓨터학 석사
 2006년~현재 : (주)아이트로닉스

Phone : 010-3326-2122
 E-mail : wind_csy@daum.net



문승진(Moon Seung Jin)

1986년 : 미텍사스 주립대 컴퓨터학 학사
 1991년 : 미플로리다 주립대 컴퓨터학 석사
 1997년 : 미플로리다 주립대 컴퓨터학 박사
 1997~현재 : 수원대학교 IT대학
 컴퓨터학과 부교수

Phone : 011-9961-2496
 E-mail : sjmoon@suwon.ac.kr